Gowin Design Timing Constraints

User Guide

SUG940-1.3E, 11/02/2021

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
06/09/2020	1.0E	Initial version published.
09/01/2020	1.1E	● Automotive grade added in operating condition; ● The link between base clock and generated clock added.
06/16/2021	1.2E	● The dscriptions of wildcards added. ● Some figures updated.
11/02/2021	1.3E	● Software version updated. ● Some figures and corresponding descriptions updated. ● Appedix A Timing Constraints Syntax Specification updated.

Contents

List of Figures

List of Table

1 About This Guide

1.1 Purpose

This manual describes the timing constraints, including timing constraints editor usage, syntax definition and static timing analysis report (hereinafter referred to as timing report). It aims to help you realize timing constraints and how to read STA reports. The software screenshots in this manual are based on V1.9.8.01. As the software is subject to change without notice, some information may not remain relevant and may need to be adjusted according to the software that is in use.

1.2 Related Documents

The latest user guides are available on GOWINSEMI Website: www.gowinsemi.com .You can find the related documents SUG918, Gowin Software Quick Start Guide.

1.3 Terminology and Abbreviations

Table 1-1 shows the abbreviations and terminology that are used in this manual.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
MPW	Minimum Pulse Width
STA	Static Timing Analysis
GUI	Graphical User Interface
REG	Register
SDC	Synopsys Design Constraint
OSC	Oscillator
PnR	Placement and Route

1.4 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly using the information provided below.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

2 Introduction

This manual includes three parts: STA, Timing Constraint Editor and Timing Report.

STA introduces the basic concepts, which is intended to help you understand the basic principles of timing analysis, learn timing constraints editor usage and read timing reports.

The timing constraints editor is GUI tool that can create and modify sdc file, and a timing report is generated according to your configuration after Place and Route.

Function

- Supports clock constraints, such as base clock, generated clock constraints, source delay and uncertainty constraints and group constraints.
- Supports data input and output delay constraints.
- Supports exception constraints, such as multi-cycle, maximum and minimum path delay constraints, and false path constraints.
- Supports the report constraints, such as the max. frequency of the module and route congestion of grid.
- Supports device operating conditions constraints.
- It provides efficient netlist lookup and expression matching.
- The GUI is flattened, simple and clear.

Feature

- The report strictly follows the W3C XHTML 1.0 specification.
- The report can open with an external browser.
- It supports navigation bar and quick positioning.
- It reports all constraints generated by timing constraints editor.
- The report is easy to read with clear hierarchy.

3 STA Overview

3.1 Overview

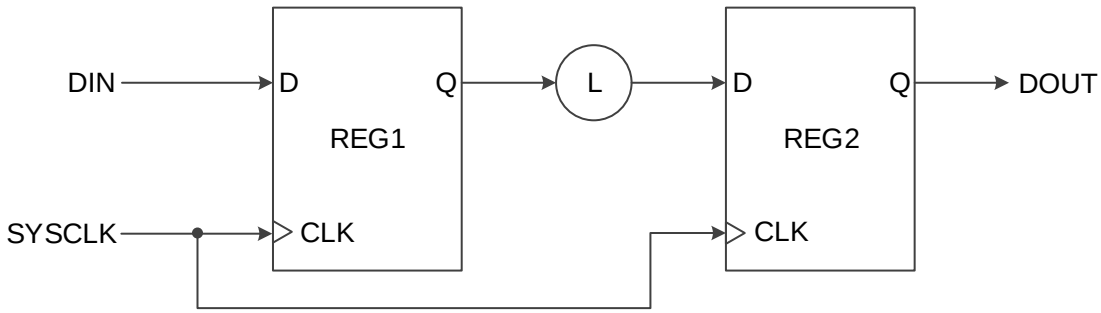
STA comprehensively analyzes the timing model in netlist, calculates the timing delay and determines whether it meets the requirements. The designer needs to provide constraint incentives, and Gowin Software completes the analysis automatically. Compared with the traditional analysis method, it features short verification time and high coverage.

The basic models, terms and concepts involved in STA are described below.

3.2 Basic Model

STA is a timing analysis model that starts from and ends with the timing component. The basic model diagram is shown in Figure 3-1. The REG1 triggers data from D to Q at the active edge. The data arrives at REG2 via logic circuit. Then the REG2 captures the data transmitting from REG1 at the active edge. STA is employed to verify whether REG2 can capture the data from the REG1 correctly.

Figure 3-1 Basic Model Diagram



The active edge of REG1 is called launch edge, and the active edge of REG2 is called latch edge. If you do not take into account the effect of the

path constraints, the interval of the two edges is usually one clock period or a half.

3.3 Terms

The basic timing units involved in the timing model are as follows:

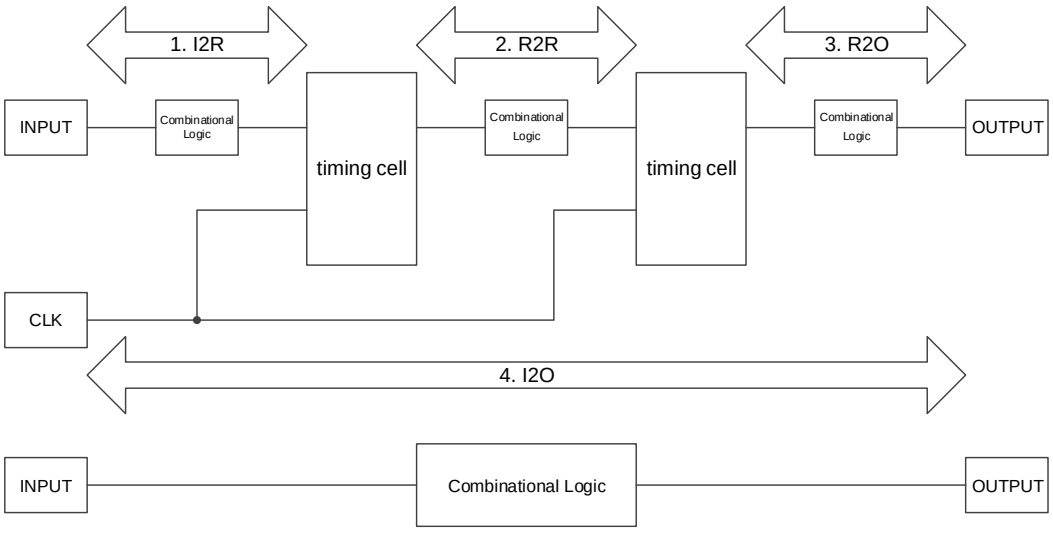
- Cells: Basic logic such as LUT, DFF, MUX, etc.
- Pins: I/O port of cells
- Ports: I/O ports of top-level module
- Nets: net
- Clocks: The clock in timing constraints

3.4 Path

STA usually analyzes four types of paths and classifies them according to different starts and ends, as shown in Figure 3-2.

- I2R: From input to timing cell
- R2R: From timing cell to timing cell
- R2O: From timing cell to output
- I2O: From input to output.

Figure 3-2 Four Types of Timing Paths



Gowin Software calculates the data arrival time and data required time of each path.

The data arrival time refers to the time from the start to the end of the timing path. The data required time refers to the time when the data arrives. When calculating data arrival time, clock path has a clock skew which

refers to the time difference of the clock arriving at the clock port of different timing components.

3.5 Common Timing Checks

STA usually check the following three types of timing and provides suggestions on PnR to better meet your requirements for timing.

3.5.1 Setup Time and Hold Time Check

- Setup time: The shortest time for data stability before the active edge. If the time is not met, the subordinate flip flop cannot capture data.
- Hold time: The shortest time for data stability after clock effective edge, if the time is not met, the data will be overwritten by the new data transmitted by the superior flip flop.

3.5.2 Recovery Time and Removal Time Check

- Recovery time: Before active edge, the shortest stable time for removing asynchronous set/reset signal. If the time is not met, the flip flop may not operate.
- Removal time: After active edge, the shortest stable time for removing asynchronous set/reset signal, if the time is not met, the flip flop may not operate.

3.5.3 MPW Check

MPW: Min. width of high and low level recognized by flip flop, such as DFF. The clock will not be recognized if the width is lower than MPW.

4 Timing Constraints Editor

4.1 Overview

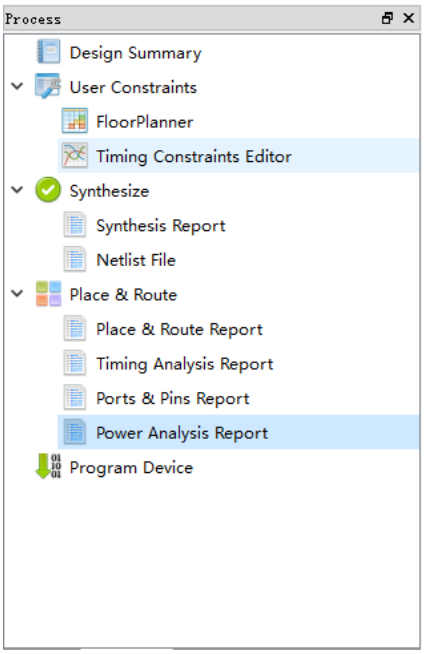
Gowin Timing Constraints Editor supports timing commands, including clock, input/output, path constraints, and clock report. You can add timing constraints via GUI. For a simple example, see SUG918, Gowin Software Quick Start User Guide.

4.2 Start Timing Constraints Editor

You can use the Timing Constraints Editor alone or start it after synthesis.

Click "Tools > Timing Constraints Editor" to start. After running Synthesize in Process window, click "Process > Timing Constraints Editor" to start the timing constraints editor, and the netlist file will be loaded automatically as shown in Figure 4-1. The netlist file and the timing constraints file in the project are automatically loaded into the timing constraint editor, or created automatically if there is no timing constraint file in the project.

Figure 4-1 Process Window



4.3 Create/Open/Add Constraints File

4.3.1 Create Constraints File

The steps of creating constraint files are as follows.

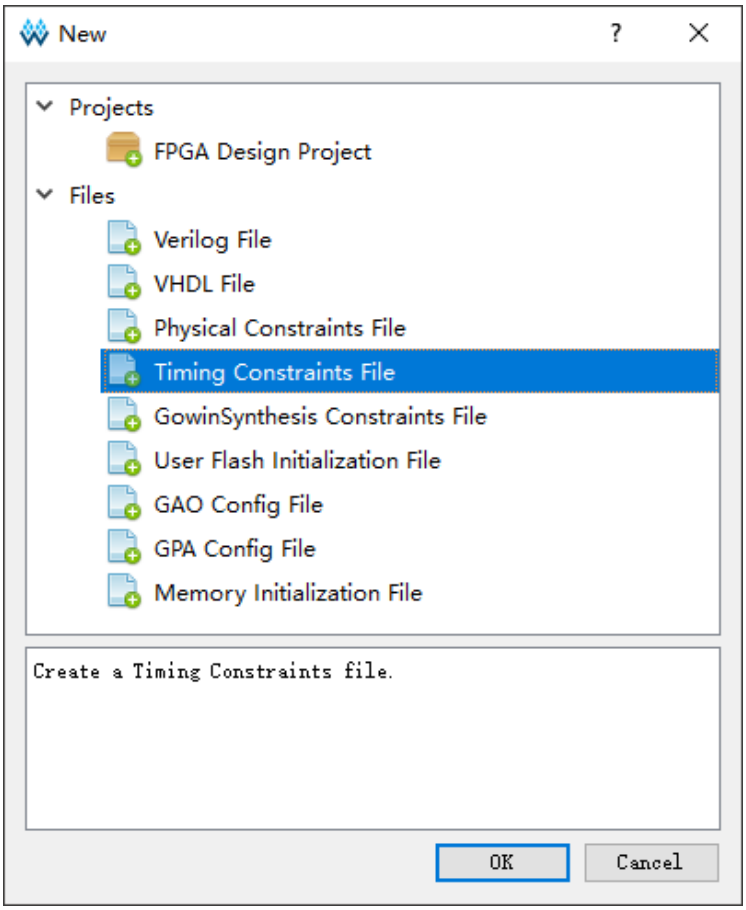
1. Click "File > New" and "New" dialog box pops up.
2. Select "Timing Constraints File", as shown in Figure 4-2.

Note!

You can also create constraints file in the following ways.

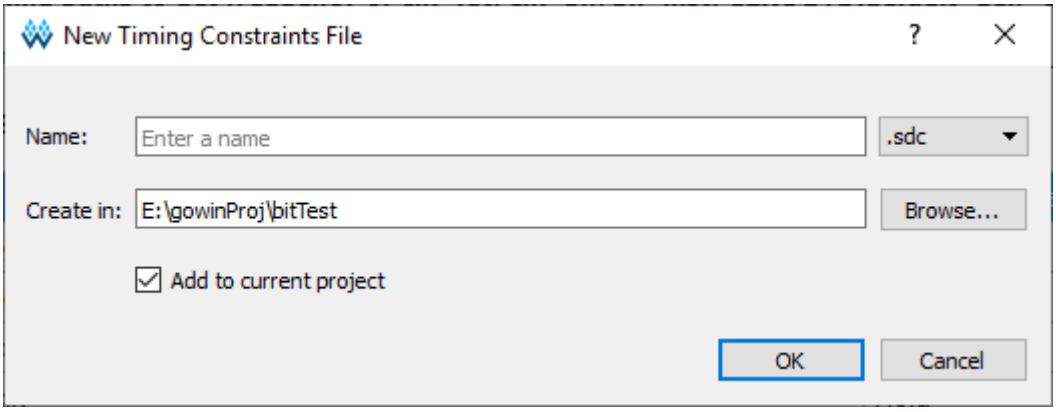
- Click the "New" icon in the toolbar.
- Use shortcut "Ctrl+N".

Figure 4-2 New Dialog Box



3. Click "OK" and "New Timing Constraints File" pops up, as shown in Figure 4-3.

Figure 4-3 New Timing Constraints File



4. Type the file name and select a path, then click "OK", and the created constraints file is automatically loaded into the project.
   - Name: The new file name with .sdc extension; and the name is recommended to use identifiers with engineering-related meaning,

beginning with letters or underscores.

- Create in: Select the path by clicking "Browse" and it is stored in src folder in project by default.
- Add to current project: Add the constraints file to the project automatically.

4.3.2 Open Constraints File

The steps are as follows.

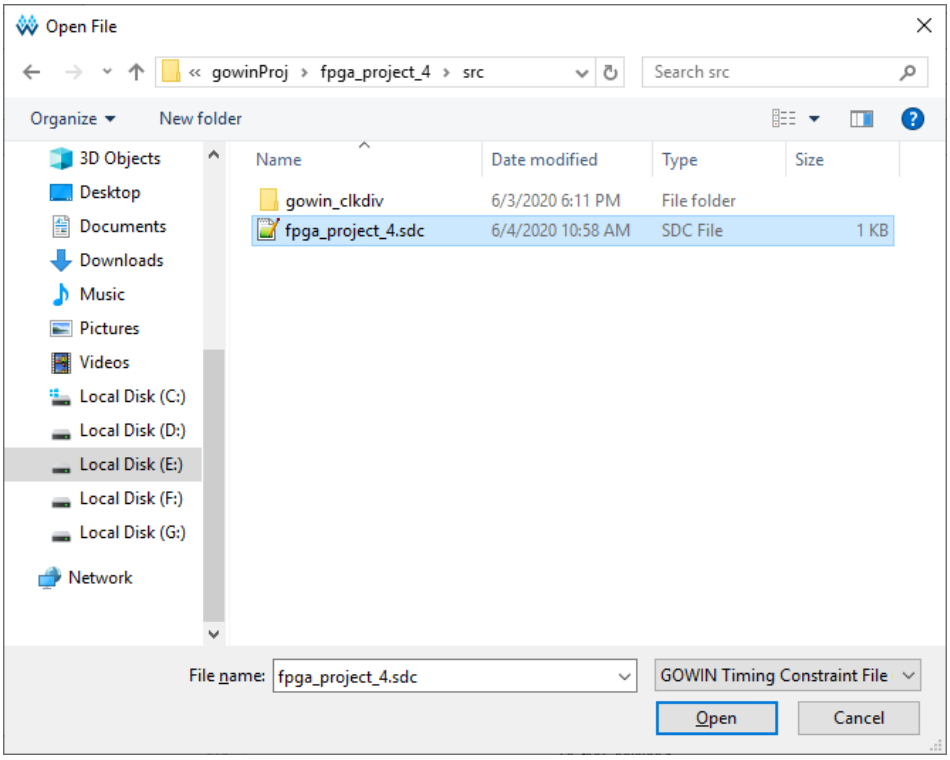
1. On IDE interface, click "File > Open".
2. "Open File" dialog box pops up, as shown in Figure 4-4.

Note!

You can also open timing constraints file in the following ways:

- Click "Open" icon in the toolbar;
- Use shortcut "Ctrl + O".

Figure 4-4 Open Timing Constraints File



3. Select and open the file with.sdc extension.

Note!

The file open operation does not automatically load the file into the project.

4.3.3 Add Constraints File

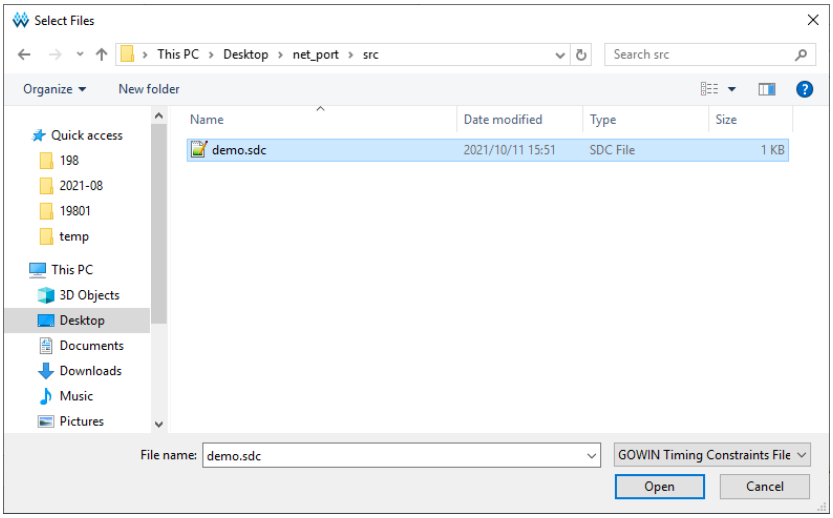
The steps are as follows.

1. Right-click and select "Add Files" in Design window.
2. "Select Files" dialog box pops up, and select the file with .sdc extension, as shown in Figure 4-5.
3. Click "Open" to add one or more selected files to the project.

Note!

Only one file is valid when multiple files added.

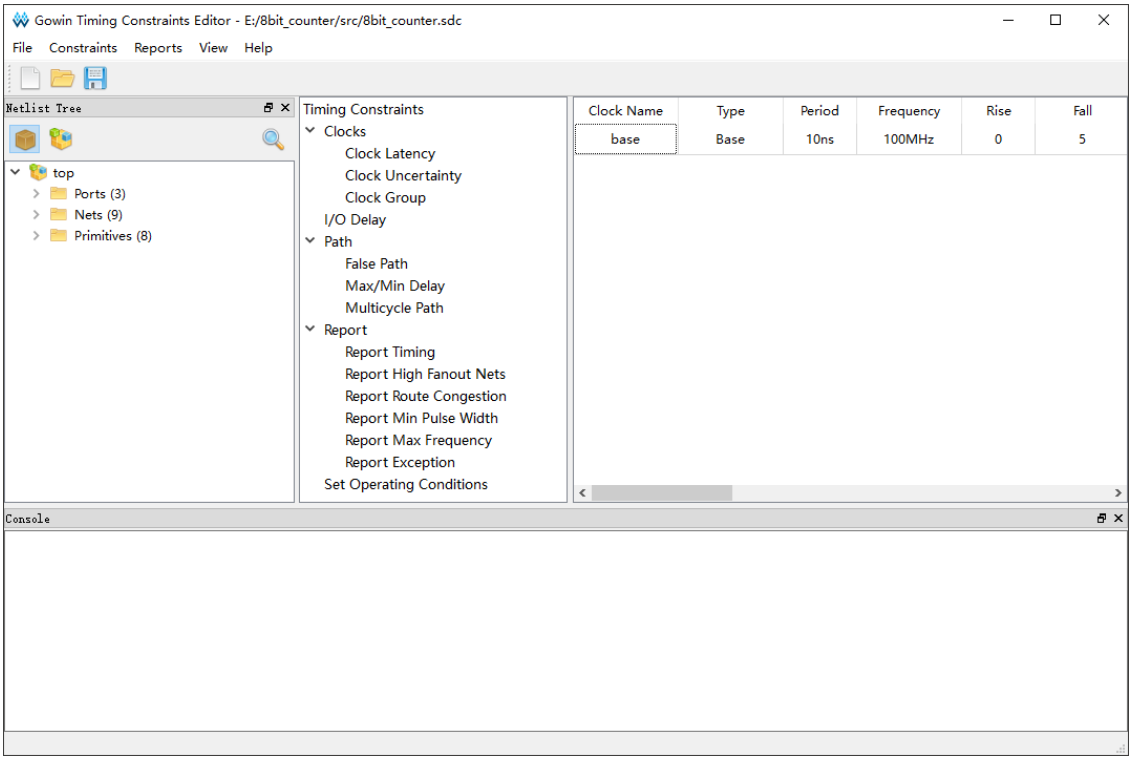
Figure 4-5 Add Constraints File



4.4 Timing Constraints Editor Interface

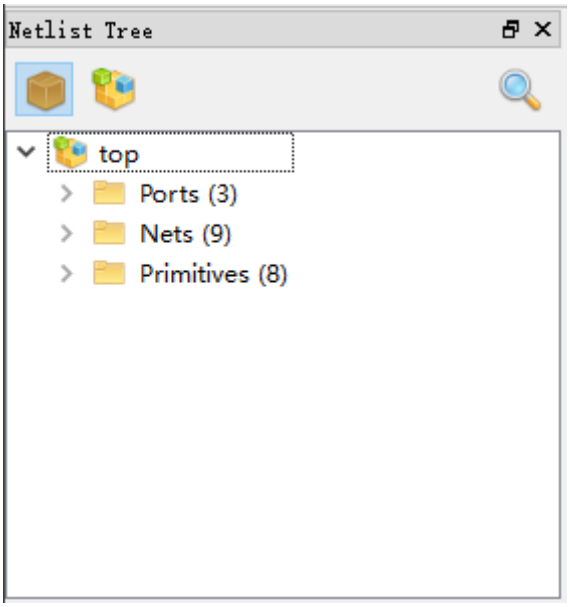
The editor interface is shown in Figure 4-6.

Figure 4-6 Timing Constraint Editor Interface



The Netlist Tree view is as shown in Figure 4-7.

Figure 4-7 Netlist Tree View

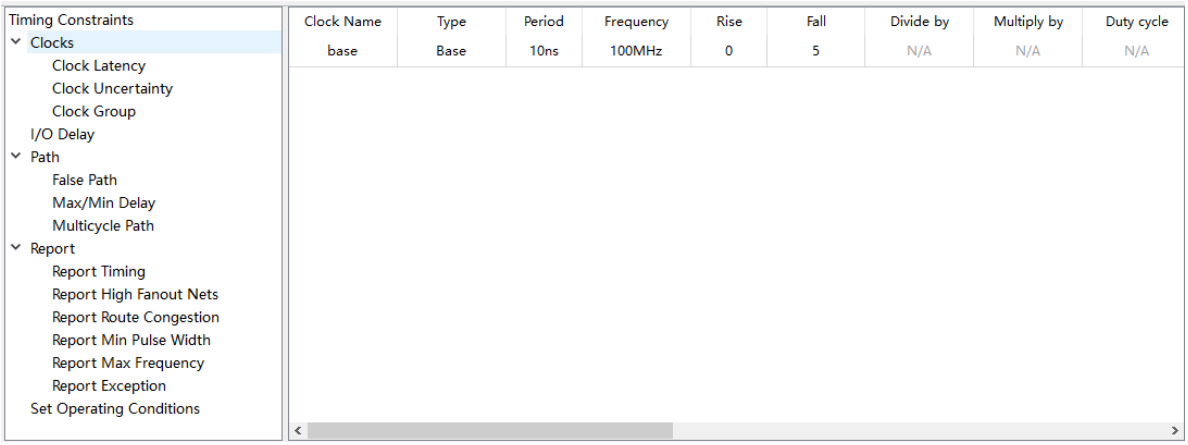


The Netlist Tree includes Top Module, I/O Ports, Nets, and Primitives.

- "  " : Flatten display
- "  ": Hierarchy display

The constraints editing area is as shown in Figure 4-8. The left is the timing constraints type and the right is the editing area. Click a constraints type, and the constraints editing list will be displayed in editing area.

Figure 4-8 Constraints Editing Window

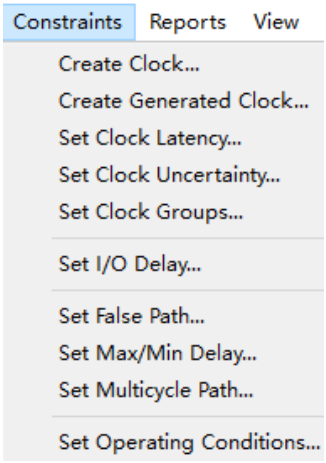


4.5 Open Timing Constraints Window

There are two ways to open.

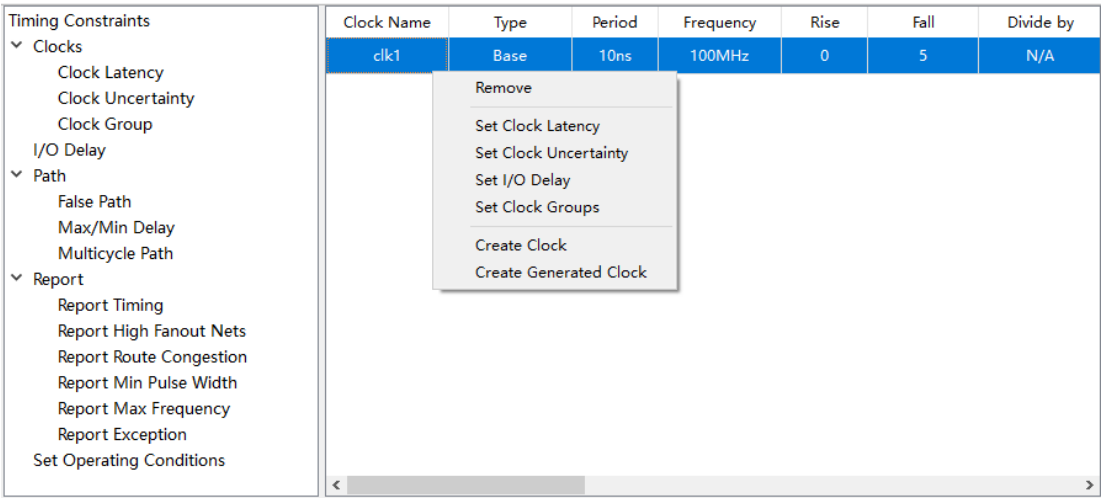
1. Click "Constraints" in menu. Select timing constraints command to open the window, as shown in Figure 4-9.

Figure 4-9 Open Timing Constraints Window via Menu Bar



2. Right click to select different timing constraints commands, as shown in Figure 4-10.

Figure 4-10 Right-click to Open Timing Constraints Window



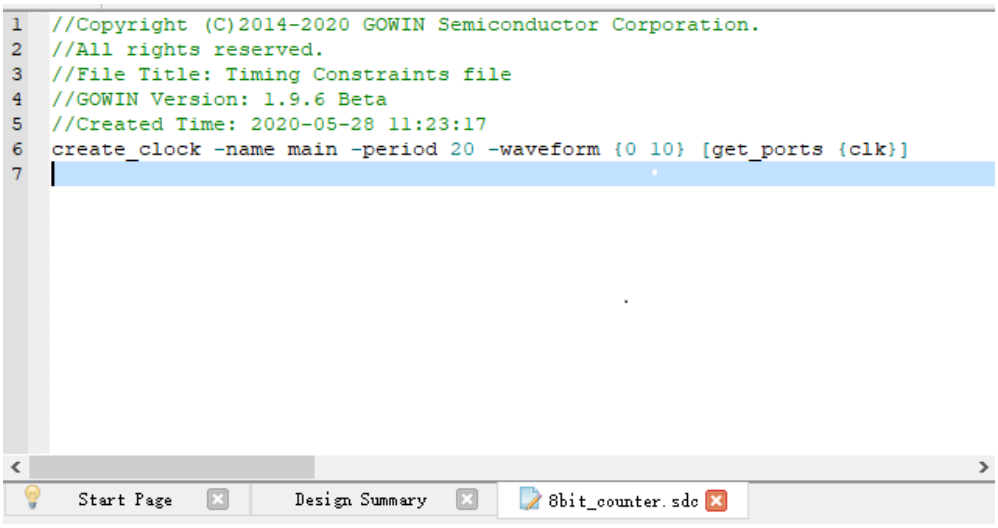
4.6 Edit SDC File

Gowin Software supports to read SDC file and you can manually modify the SDC file in the timing constraints editor, as shown in Figure 4-11.

The parsing of SDC files supports wildcard; currently two wildcard characters "*" and "?" are supported. "*" matches zero or multiple characters, while "?" matches a single character.

SDC files support single line and multi-line comments. Single line comments use "//" or "#", and multi-line comments use "/* */".

Figure 4-11 Edit SDC File



4.7 Create Timing Constraints

This section introduces how to create timing constraints using editor. The created timing constraints will be written to the SDC file in the project. You can see Appendix A for details.

4.7.1 Clock Constraints

Create Clock

- You can configure parameters, such as name, period, frequency, rising edge, falling edge, etc.
- Gowin Software can create multiple clocks which form multiple clock domains, and support clock domain crossing analysis.

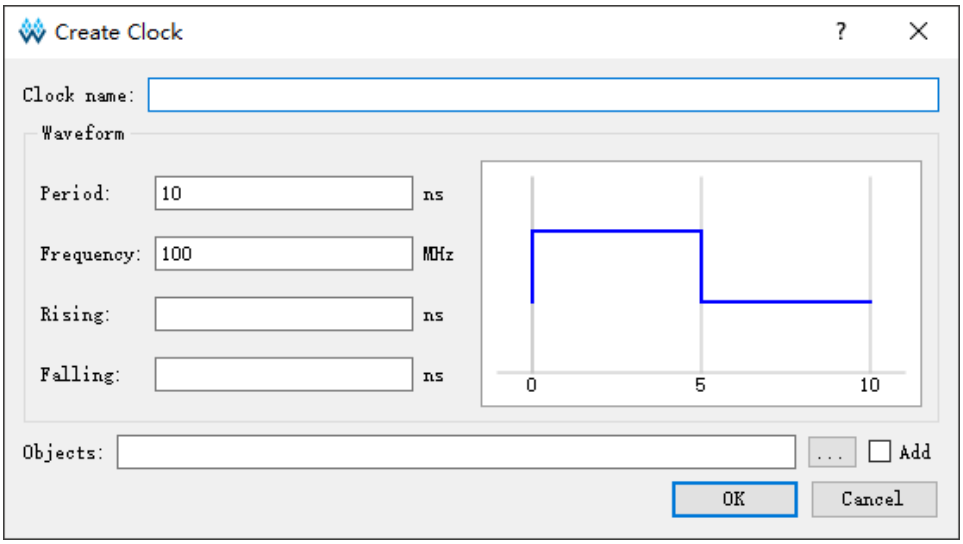
Create_clock can create a base clock for user design. For example, Gowin Software creates the 100MHz clock by default, while the external OSC used is 50MHz, and you can create a base clock to resolve the frequency mismatch.

You can add clock constraints in the following two ways.

1. Add Clock constraint via Constraints

   a). Select "Constraints > Create Clock... ", and the "Create Clock" dialog box pops up, as shown in Figure 4-12.

Figure 4-12 Create Clock



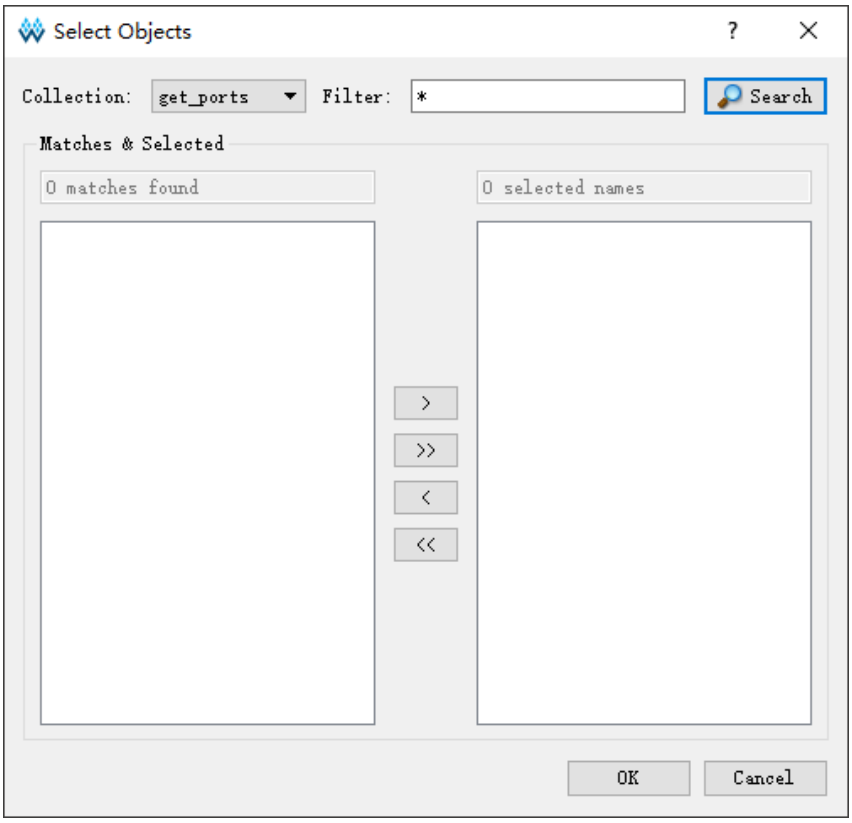
   b). Configure "Clock Name", "Waveform", and "Objects".

   - Clock Name: Supports identifiers beginning with letters or underscores.
   - Period: Accurate to thousandth, floating; 10 by default.

- Frequency: Accurate to thousandth, floating; 100 by default.
- Rising: Accurate to thousandth, floating.
- Falling: Accurate to thousandth, floating.
- Objects: Specifies the target; click "..." to select the object.

c). Click "..." and " Select Objects" pops up, as shown in Figure 4-13.

Figure 4-13 Select Objects



d). As shown in Figure 4-13, "Collection" specifies the object type. "Filter" is a filter. After clicking "Search", the objects are displayed on the left, and the selected will be displayed on the right. ">" adds the selected from the left list to the right list. "> >" adds all the selected from the left list to the right list. "<" removes the selected in the right list. "< <" removes all the selected in the right list.

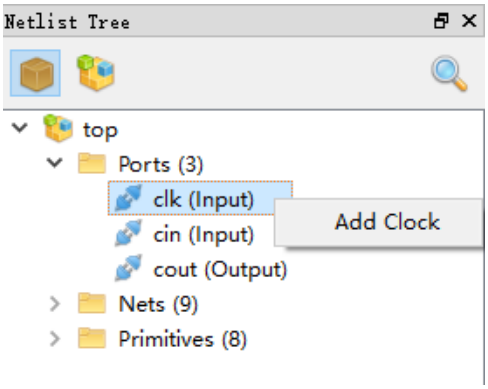
e). Click "OK" to add Objects.

2. Add clock constraints via Netlist Tree.

a). Select I/O Port or Net in Netlist Tree.

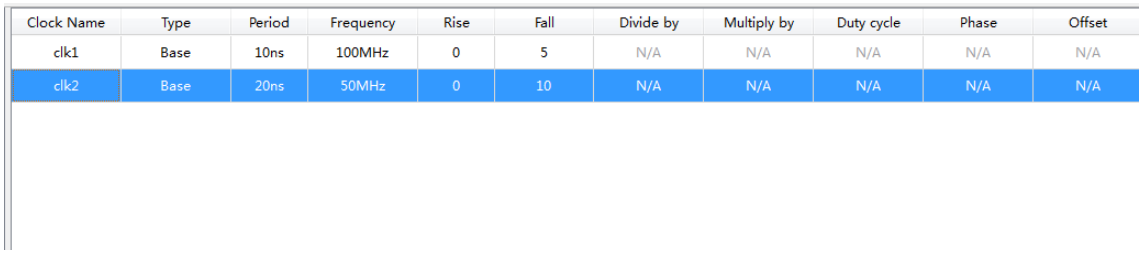
b). Right-click and select "Add Clock" to add a clock, as shown in Figure 4-14.

Figure 4-14 Add Clock



After finishing, the constraints will be added in clock list, as shown in Figure 4-15.

Figure 4-15 Clock List

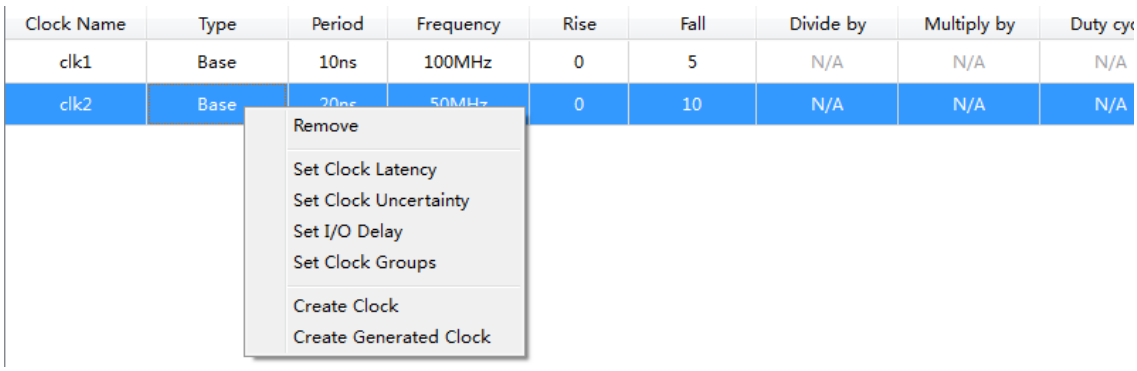


Clock Name	Type	Period	Frequency	Rise	Fall	Divide by	Multiply by	Duty cycle	Phase	Offset
clk1	Base	10ns	100MHz	0	5	N/A	N/A	N/A	N/A	N/A
clk2	Base	20ns	50MHz	0	10	N/A	N/A	N/A	N/A	N/A

You can perform the following operations.

- Double-click the constraints to edit.
- Right-click and select "Remove" to remove the clock.
- Right-click to set Clock Latency, Clock Uncertainty, or I/O Delay, as shown in Figure 4-16.

Figure 4-16 Right-click



Note!

- When a constraint is inconsistent with the PLL configuration, you should take the one created by Create Clock as the standard. A prompt will pop up in PnR.
- Create Clock does not support virtual clock.

Create Generated Clock

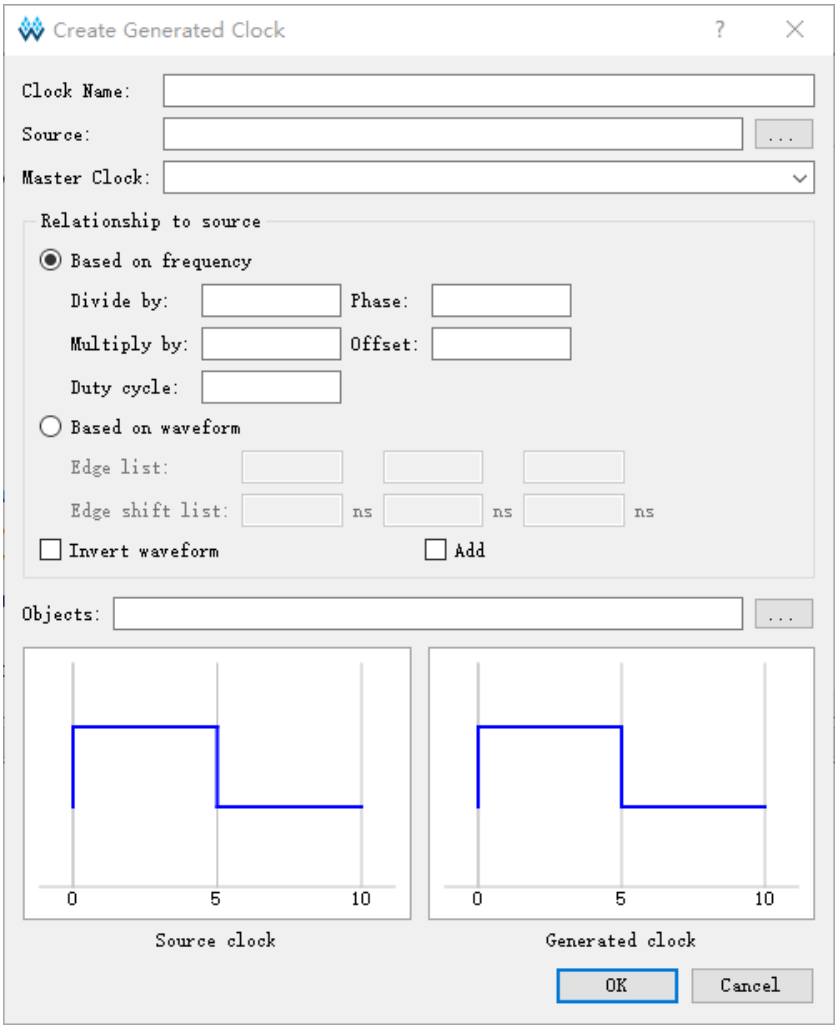
- Create generated clock according to the base clock;
- You can create generated clock based on the frequency division, frequency multiplication, phase and duty cycle of the base clock.

The generated clock must be based on the base clock and you can create at any node in the user design. They are usually applied to the output ports of PLL, CLKDIV and other hard cores. If you use PLL in the design, after the base clock is created, the generated clock with Objects as PLL. CLKOUT and Source as the base clock can be created. The generated clock is automatically linked to the base clock, and the generated clock is automatically corrected to adapt to the base clock when the attributes of the base clock change.

You can create the generated clock in the following two ways:

1. Create via Constraints
   a). Select "Create Generated Clock" and "Create Generated Clock" pops up, as shown in Figure 4-17.
   - Clock Name: Supports identifiers beginning with letters or underscores.
   - Source: The source of generated clock; click "..." to select one.
   - Master Clock: The clock acting on Source; click " " to select one.

Figure 4-17 Create Generated Clock Constraints



b). You can automatically add the clock associated with Source to "Master Clock" by clicking "..." on the right side of "Source"; when Master Clock has multiple clocks, only one of them is supported.

c). In "Relationship to source" pane, you can configure frequency division/multiplication, offset, duty cycle and phase in "Based on frequency"; and you can also adjust edge in "Based on waveform".

- Divide by: Division, positive integer.
- Phase: Accurate to thousandth, floating; negative shifted left, positive shifted right.
- Multiply by: Multiplication, positive integer.
- Offset: Accurate to thousandth, floating; negative shifted left, positive shifted right.
- Duty cycle: Accurate to thousandth, floating; value not greater than 100.
- Edge list: Positive integer increased sequentially.

- Edge shift list: Accurate to thousandth, floating.

d). You can invert clocks by clicking "Invert waveform" and add clocks by clicking "Add".

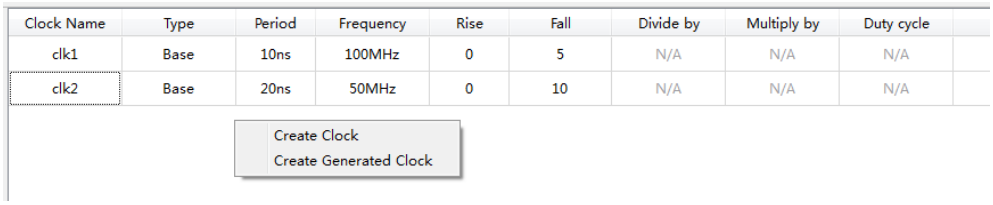
e). In "Objects", click " ... " and "Select Objects" pops up to select the object.

Note!

- If there is no clock in Source, Master Clock has no option, and you need to select Source again.
- When a constraint is inconsistent with the PLL configuration, you should take the one created by Create Generated Clock as the standard. A prompt will pop up in PnR.

2. Create generated clock from clocks list. Right-click and select "Create Generated Clock" to create generated clock, as shown in Figure 4-18.

Figure 4-18 Select Create Generated Clock



The new constraints will be added.

You can perform the following operations.

- Double-click the constraints to edit.
- Select the clock and right-click to select "Remove" to remove the clock.

Set Clock Latency

- It is used to set the latency before clock signal reaching device port. You can configure the max./min. latency respectively for rising /falling edges.
- Clock latency includes two types: network latency and source latency.
  - Network latency is internal clock path delay.
  - Source latency is external clock path delay.
- Gowin Software will calculate clock network latency automatically, so you only need to set source latency.

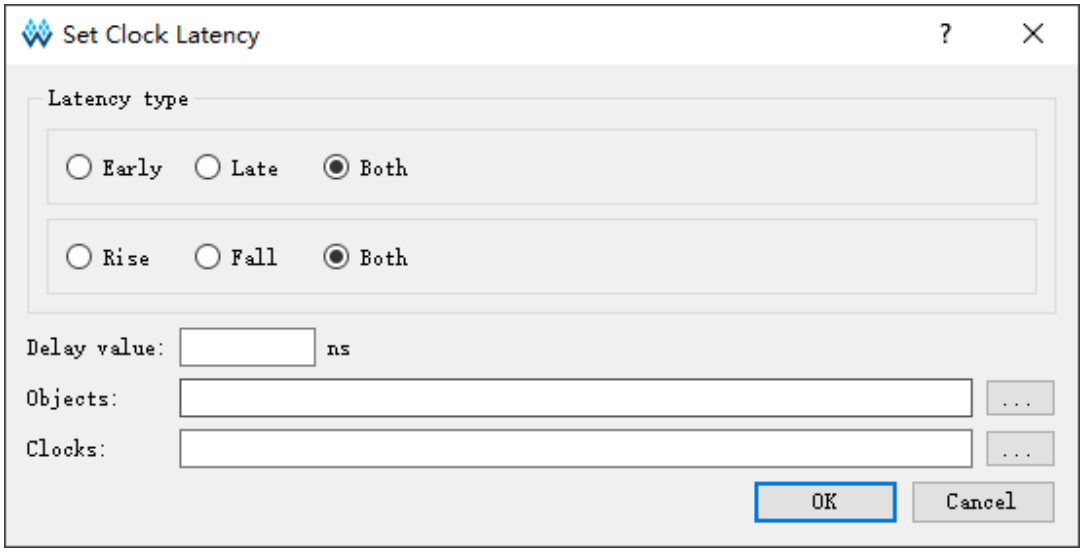
The latency of the clock signal from the clock source(external crystal oscillator) to the FPGA clock port is called the source latency, which is not automatically known by Gowin Software. The default value is 0ns. If you know the source latency 2ns, the Delay Value can be configured as of 2ns, and Gowin Software will added 2ns value automatically in timing analysis.

The results can be found in Tcl of the Setup and Hold reports.

You can create clock latency constraints in the following two ways.

1. Select "Set Clock Latency" in "Constraints" menu, and Set Clock Latency dialog box pops up, as shown in Figure 4-19.
   - Early and Late: Indicates whether the setting is minimum or maximum latency. Late is used for Setup analysis and Early is used for Hold analysis. Both is used for Setup and Early analysis.
   - Rise and Fall are valid for rising edge and falling edge; Both indicates Rise and Fall both are valid.
   - Objects: Specifies the clock input port or clock; select one by clicking "...".
   - Clocks: Specifies a clock; select one by clicking "...".

Figure 4-19 Set Clock Latency



2. Create Clock Latency constraints in clocks list. Select a clock in clock list; right-click and select Set Clock Latency; this clock will be automatically selected in Objects.

Set Clock Uncertainty

- Set Clock Uncertainty is used to set clock uncertainty or offset to analyze clock transmission.
- It can set uncertainty for setup and hold, or clock rising edge and falling edge.
- You can inform Gowin Software of clock jitter, pessimistic, etc. via the constraints to affect timing.

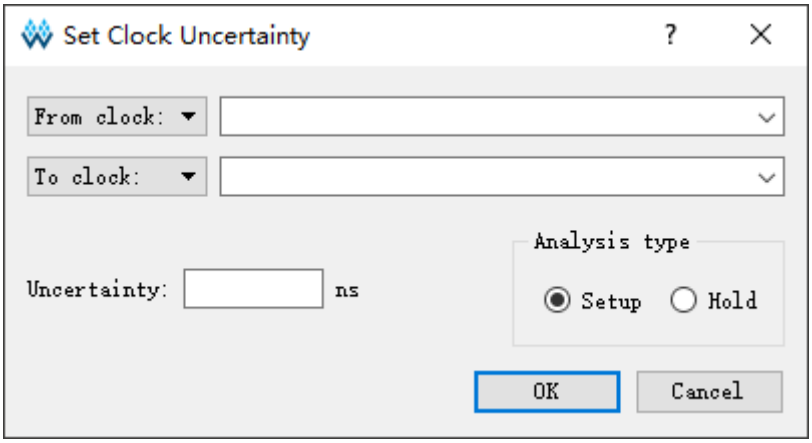
An ideal clock signal does not generate uncertainty. However, the clock uncertainty usually exists, and Gowin Software will calculate the

uncertainties by default. You can also provide a more reasonable uncertainties according to the actual hardware environment to Gowin Software for analysis. Assuming that the device works in a strong magnetic environment and the uncertain value is 0.2ns, then Uncertainty can be set as 0.2ns. The results can be found in tUnc of the Setup and Hold reports.

The steps to create clock uncertainty are as follows.

1. In "Constraints", select "Set Clock Uncertainty" and "Set Clock Uncertainty" pops up, as shown in Figure 4-20.
   - From clock: Specifies the start clock, selected by clicking "˅" on the right.
   - To clock: Specifies the end clock, selected by clicking "˅" on the right.
   - Uncertainty: Accurate to thousandth, floating.
   - Analysis type: Indicates the type of analysis.

Figure 4-20 Set Clock Uncertainty



2. Select in From type and To type on the left, select clocks on the right;
3. Click "OK" to add uncertainty.

Set Clock Group

- It used to specify the relationship between different clocks.
- Gowin Software provides the relationship between the group members by default, and there is no correlation between groups.
- Gowin Software defaults to the fact that all clocks in the deisgn belong to a group and are related.

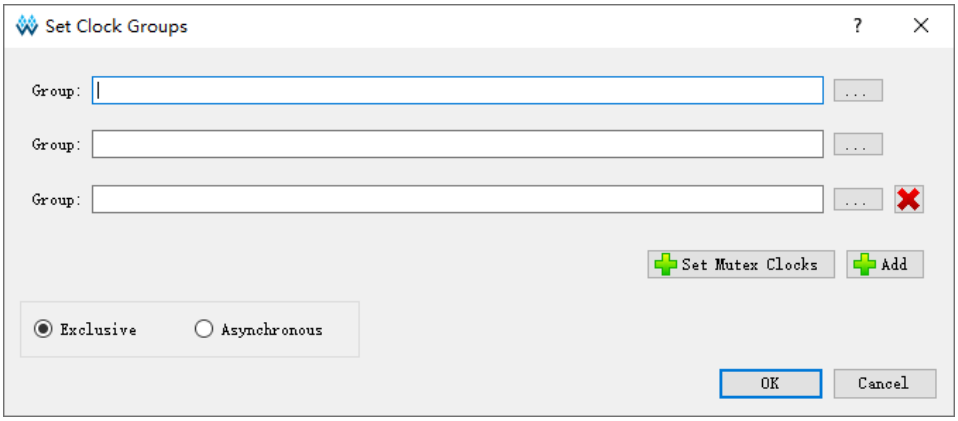
The constraint is usually used for mutual exclusive or asynchronous clocks. For example, there are CLK1 and CLK2 with different frequencies in the design, and only one clock is valid at the same time via a multiple-selection.

It is recommended that you specify the relationship between clocks and create clock group constraints for asynchronous or mutually exclusive clocks.

You can create clock group in the following ways.

1. In "Constraints", select "Set Clock Groups" and "Set Clock Groups" dialog box pops up, as shown in Figure 4-21.
   - Group: Specifies the clock, at least one clock, selected by clicking "..." on the right.
   - Add: Add a group.
   - Set Mutex Clocks: Sets multiple groups at one time.
   - Exclusive: Indicates clocks are exclusive. For example, Clock0 and Clock1 pass through a MUX2 and outputs Clock3 acting on a timing model, and the same time Clock3 can not be both Clock0 and Clock1.
   - Asynchronous: Indicates that clock asynchronization is not related, and the clock has different clock sources. For example, a timing model is transmitted and sampled by Clock0 and Clock1, which can be from different external ports.

Figure 4-21 Set Clock Groups



2. Click "..." to select Clock for group; and click "✖" to remove the clock.
3. Click "OK" to save.

Note!

The options "Exclusive" and "Asynchronous" achieve the same result.

4.7.2 I/O Delay Constraints

set_input_delay

You can adjust the data arrival and clock arrival relationship by set_input_delay; the too early or too late input data of device will cause the internal clock to latch the wrong data; then you can set a suitable input delay value to make the clock have enough time to latch the correct data.

Note!

The input delay type in the timing report is "tIn".

set_output_delay

You can adjust the data output and clock output relationship by set_output_delay; the too early or too late output data of device will cause the external clock to latch the wrong data; then you can set a suitable output delay value to make the clock have enough time to latch the correct data.

Note!

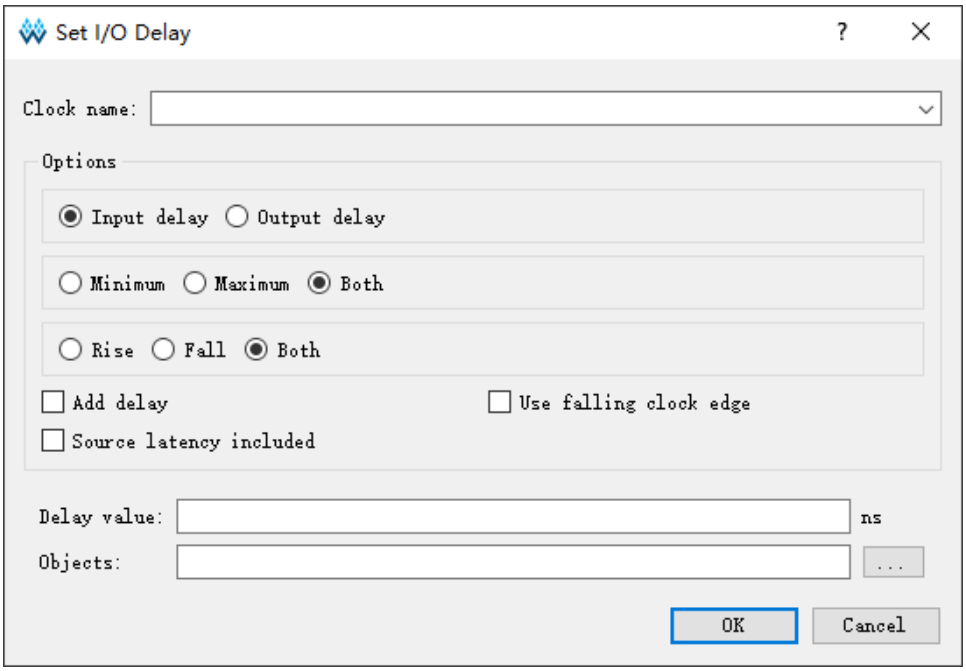
The output delay type in the timing report is "tOut".

You can create I/O delay constraints as follows.

1. In "Constraints" menu, select "Set I/O Delay" and "Set I/O Delay" dialog box pops up, as shown in Figure 4-22.
   - Clock name: Indicates the clock associated with I/O; the clock must exist, and click " " to select it.
   - Options: Configure delay type, maximum and minimum delay, clock edge, etc.
   - Input delay/Output delay: Specifies the input/output delay, and they are exclusive.
   - Minimum/Maximum: Specifies the minimum or maximum delay value of I/O; both indicate that both delay values are the same.
   - Rise/Fall: Indicates that rising or falling edge is valid; both indicate that they are both valid at the same time.
   - Delay value: Sets the delay value of I/O, positive or negative floating, accurate to thousandth; when it is negative, it means arrive early; when it is positive, it means arrive late.
   - Objects: Specifies the input and output ports, and click " " on to select one.
   - Add delay: Used to add a delay value to the same port; if not specified, the same constraint on the same port will be overwritten.

- Use falling clock edge: If checked, the falling edge of the associated clock is related; the default is rising edge correled.
- Source Latency include: If checked, it indicates that the delay value set includes the delay value of the clock; if not checked, Gowin Software will count in the delay value of the clock.

Figure 4-22 Create I/O Delay Constraints



2. Click "OK" to save the constraints.

4.7.3 Timing Path Constraints

Set False Path

Gowin Software will analyze all timing paths. Set False Path specifies the paths in the design that do not need to be analyzed. It is recommended for you to specify paths that need not be analyzed.

There are usually two types of timing paths that do not require analysis.

- The timing circuit unrelated to operating, such as the test circuit.
- The path across the asynchronous clock domain. Assuming that there are register A and register B, A outputs data to B, and A and B are respectively driven by asynchronous clocks CLK1 and CLK2. Then From can be configured as CLK1 and To as CLK2, and the path from CLK1 launch to CLK2 latch will not be analyzed.

You can create Flase Path constraints as follows:

1. Select "Constraints > Set False Path", then "Set False Path" pops up, as shown in Figure 4-23.
   - Analysis type: Checks Setup or Hold; Both indicate both are checked.
   - From: Indicates the start of the path.
   - To: Indicates the end of the path.
   - Through: Indicates the pins or nets through the path.

Note!

From, To and Through can be used individually or together with each other.

Figure 4-23 Create False Path Constraints

2. Click "..." to select objects, as shown in Figure 4-13. Click "OK" to save.

Set Max/Min Delay

Specify the maximum and minimum delay values on a path.

It is usually used in pin-to-pin delay analysis. If the input port A is output to port B after combinational logic, Gowin software does not analyze and report the path from port A to port B by default. You can use this constraint to specify a delay value from port A to port B. Gowin Software automatically calculates, analyzes and reports the path specified by you. The maximum delay is reported in Setup, and the minimum delay is reported in Hold.

You can create Max/Min Delay constraints as follows:

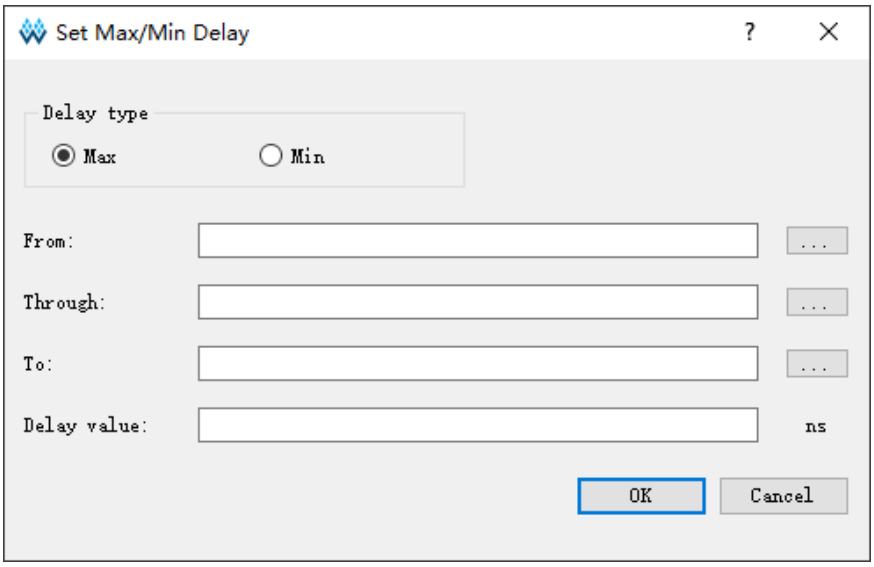
1. Select "Constraints > Set Max/Min Delay", then "Set Max/Min Delay" pops up, as shown in Figure 4-24.
   - From: Indicates the start of the path; click "..." to select one.
   - To: Indicates the end of the path; click "..." to select one.

- Through: Indicates the pins or nets through the path; click "..." to select one.
- Delay value: Sets delay value; accurate to thousandth, floating.

Note!

From, To and Through can be used individually or together with each other.

Figure 4-24 Create Max/Min Delay Constraints



2. Select type (Max or Min) in Delay Type and select Objects in From/To. Click "OK" to save.

Set MultiCycle Path

By default, Gowin Softwre performs single-cycle clock analysis, that is, the check of setup time is on the active clock edge of the next clock cycle at the edge of the source clock, but this method does not apply to certain timing paths. Logic design circuit is the most typical example. More than one clock cycle data shall be needed to stabilize if a logic circuit calculates more complex or longer path.

If the data on the timing Path_A in the design needs two cycles to stabilize, and Gowin Software defaults to the one cycle analysis. You need to set Value to 2, and Gowin Software can analyze according to the value. The results can be found in the Setup and Hold reports.

Note!

- Setting the multicycle path command will affect the setup time and the hold time. If the -setup or -hold option is not specified, Gowin Software defaults to -setup. If -setup value is set, hold value will not be affected.
- Gowin Software provides the function to automatically repair Hold by default. If you specify a hold value, Gowin Software will prioritize user setting.

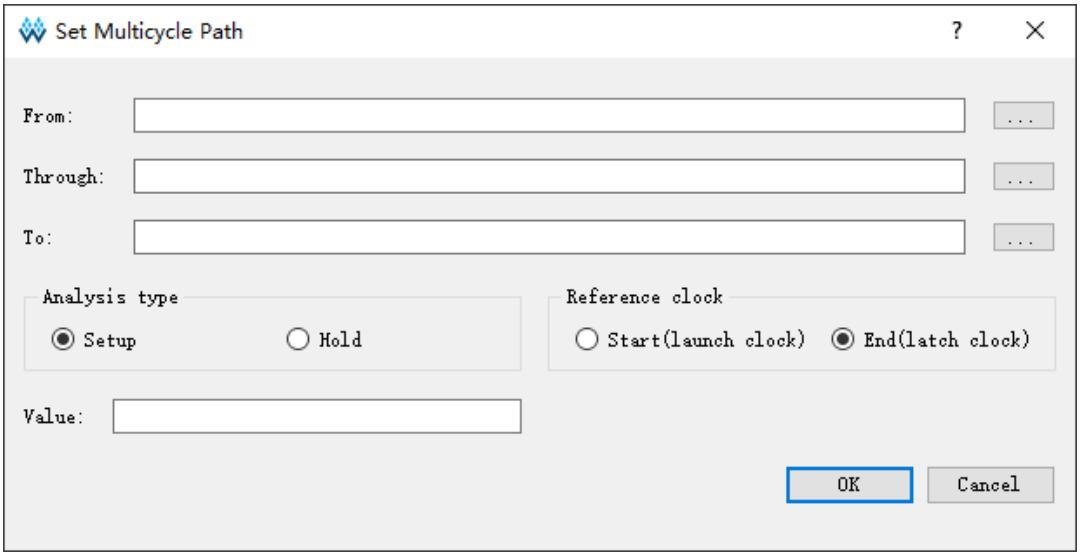
You can create multicycle path constraints as follows:

1. Select "Constraints > Set Multicycle Path", then "Set Multicycle Path" pops up, as shown in Figure 4-25.
   - Reference clock: Indicates whether the reference clock is launch or latch clock.
   - Analysis type: Specifies constraints on Setup or Hold check.
   - From: Indicates the start of the path; click "..." to select one.
   - Through: Indicates the pins or nets through the path; click "..." to select one.
   - To: Indicates the end of the path; click "..." to select one.
   - Value: Specifies the number of multi-cycle periods, positive or negative integer; negative means advance, and positive means delay.

Note!

From, To and Through can be used individually or together with each other.

Figure 4-25 Create Multicycle Path Constraints



2. Click "OK" to save the constraints.

4.7.4 Operating Conditions Constraints

The delay model used in timing constraints analysis can specify the speed level, model type and so on. Gowin Software uses Slow Model for Setup analysis and Fast Model for Hold analysis by default.

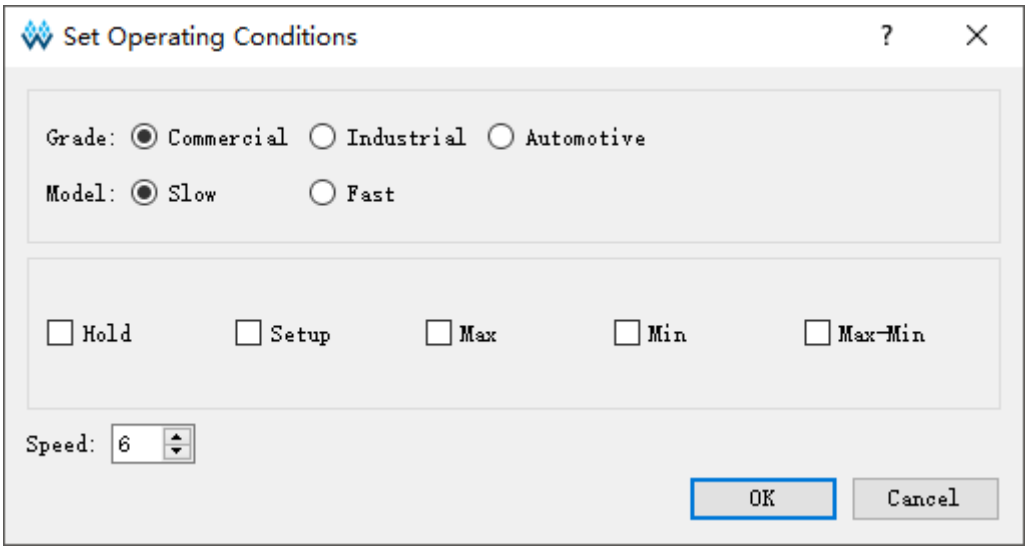
You can also customize the timing model. For example, in the case of hot and unstable power supply, the slow delay model can be specified. You can check the delay model in STA Tool Run Summary.

Select "Constraints > Set Operating Conditions", then "Set Operating

Conditions" dialog box pops up, as shown in Figure 4-26.

- Grade: Commercial, industrial and automotive.
- Model: Slow and fast. Slow applies to high temperature and low pressure, and fast applies to low temperature and high pressure.
- Hold and Setup: Indicates hold time and setup time valid.
- Max function is the same as Setup, and Min function is the same as Hold.
- Max-Min is equivalent to selecting both Max and Min.

Figure 4-26 Create Operating Conditions Constraints



Note!

- When the grade and speed set do not match the chip part number, the actual constraint shall prevail.
- If the grade and speed of the actual constraints do not support the current project, Gowin Software will report a warning message.
- The engineering sample (ES) uses the slowest speed to analyze the timing sequence by default, and you can set the speed of engineering sample as required.

4.7.5 Reports

Report Timing

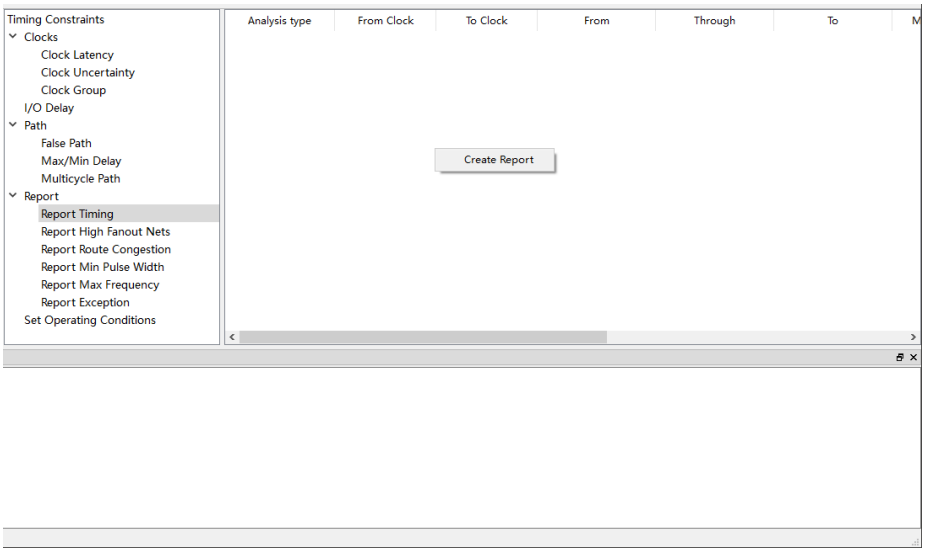
According to the set parameters, Gowin Software can provide reports with more details.

For example, Gowin Software reports 25 Setup analysis paths by default. When you need to view the analysis of 35 worst Setup paths, enter 35 in "Max Paths", as shown in Figure 4-28. The results can be found in the Setup and Hold reports.

The steps are as follows:

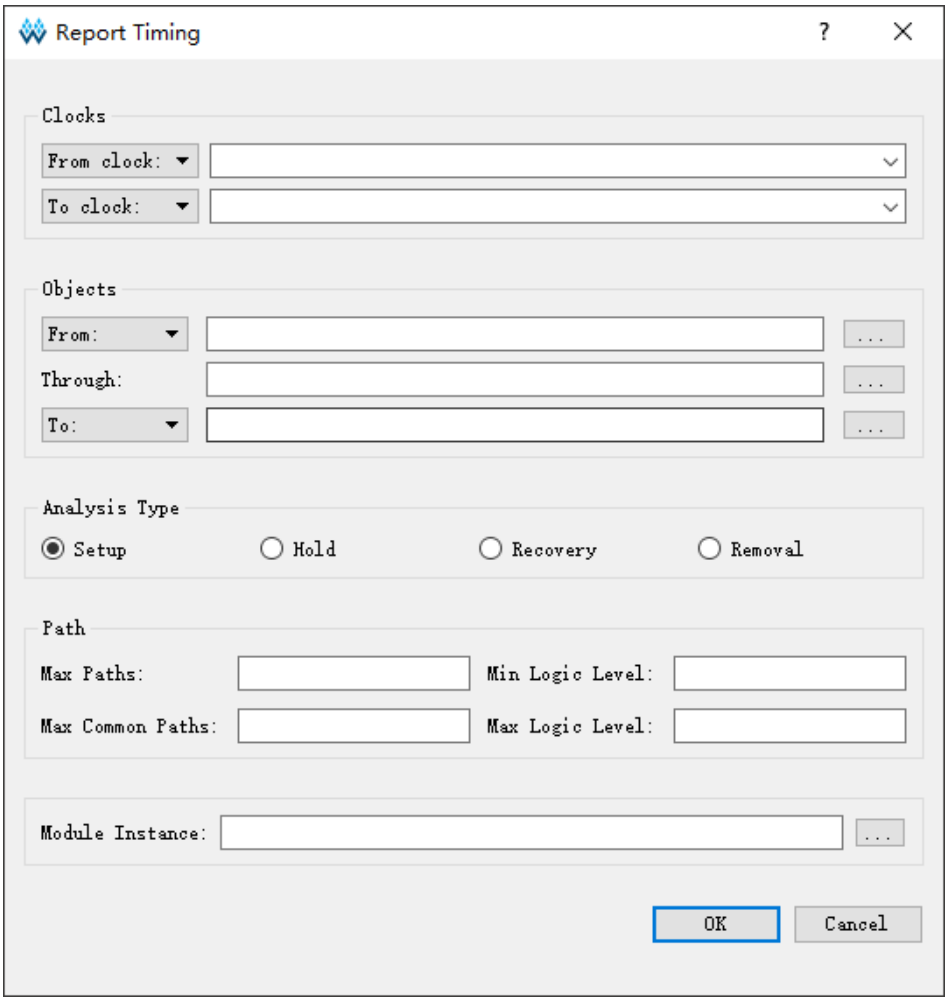
1. Select "Timing Constraints > Report Timing" and right-click to select "Create Report", as shown in Figure 4-27.

Figure 4-27 Create Report Timing



2. Select "Create Report" and Figure 4-28 pops up.
   - Path: Specifies the max. paths, the max. common paths, the max./min. logic level, positive integers
   - Clocks: Specifies the associated clock of a path. From/To Clock indicates the transmitted clock and sample clock respectively; click "⌄" on the right to select one.
   - Objects: Specifies the start and end objects of the analysis; click "⌄" on the right to select one.
   - Analysis Type: Specifies Setup, Hold, Recovery and Removal.
   - Module Instance: Specifies the instance name; click "⌄" on the right to select one.

Figure 4-28 Report Timing Dialog Box



3. Click "OK" to save.

Report High Fanout Nets

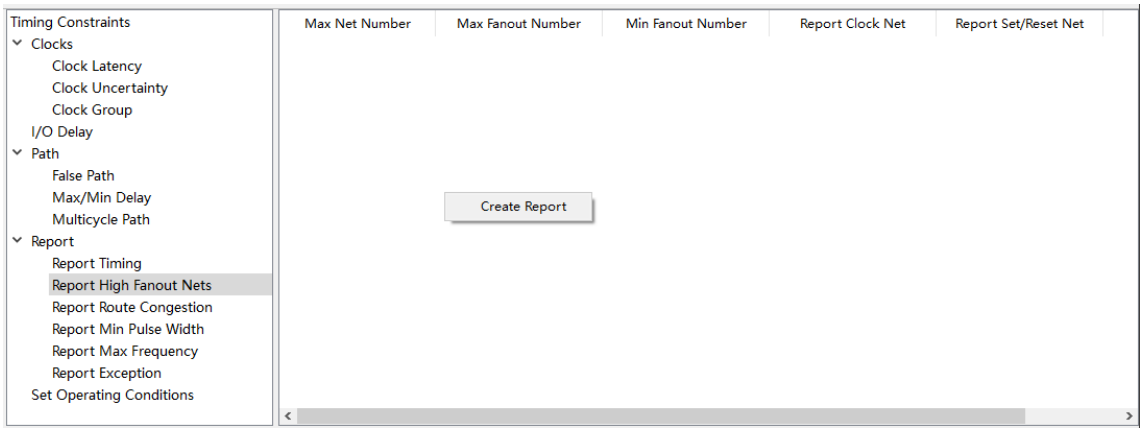
It reports the number of fans for Net, 10 of the largest by default.

If you need to view the net between 5 and 7, you can specify Min Fanout as 5 and Max Fanout as 7. The results can be viewed in High Fanout Nets Report.

The steps are as follows:

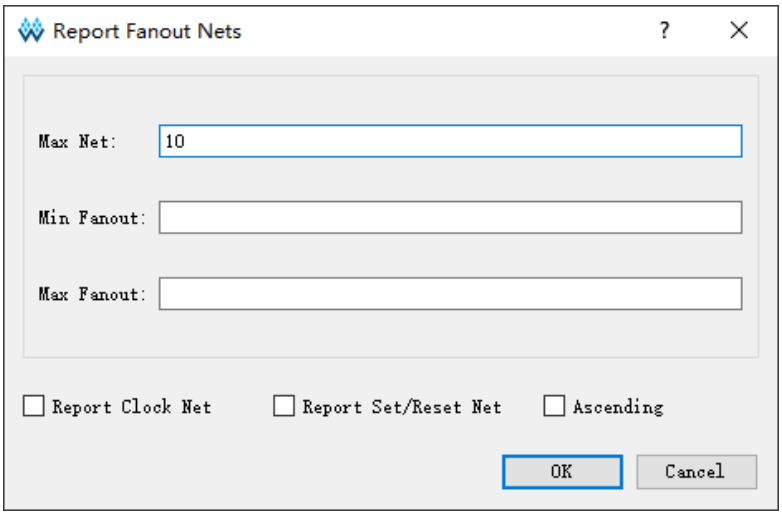
1. Select "Timing Constraints > Report High Fanout Nets".
2. Right-click to select "Create Report", and Create Report dialog box pops up, as shown in Figure 4-29.

Figure 4-29 Create Report High Fanout Nets



3. Select "Create Report" and Figure 4-30 pops up.
   - Max Net: Specifies the maximum, positive integer.
   - Min and Max Fanout: Specifies the min. and max. fanout, positive integer.
   - Report Clock Net: Reports the net connected to the clock input of the timing component.
   - Report Set/Reset Net: Reports the net connected to the reset input of the timing component.
   - Ascending: Specifies the net order, ascending by default.

Figure 4-30 Report High Fanout Nets Dialog Box



4. Click "OK" to save.

Report Route Congestion

It reports the route congestion, 10 of the worst grid by default.

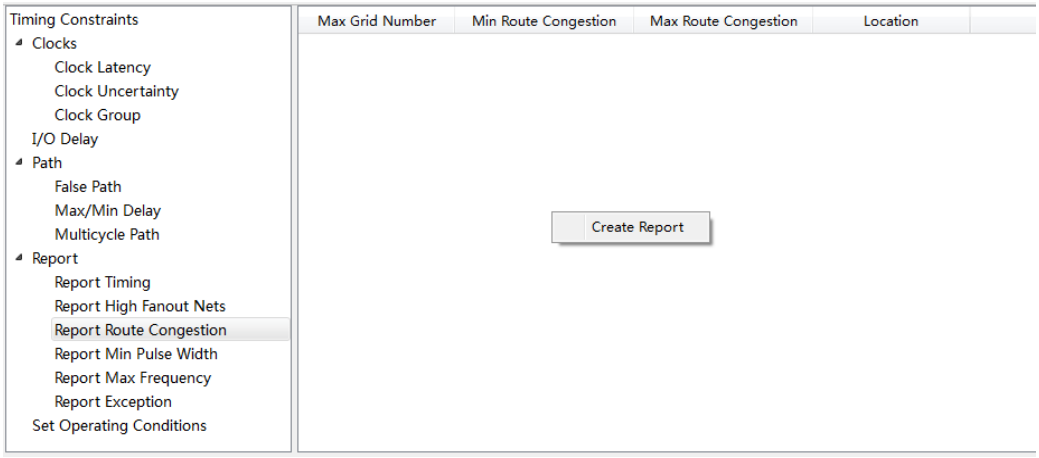
It usually reports the congestion on a specific grid, such as the Grid

R4C4. The results can be seen in Route Congestions Report.

The steps are as follows:

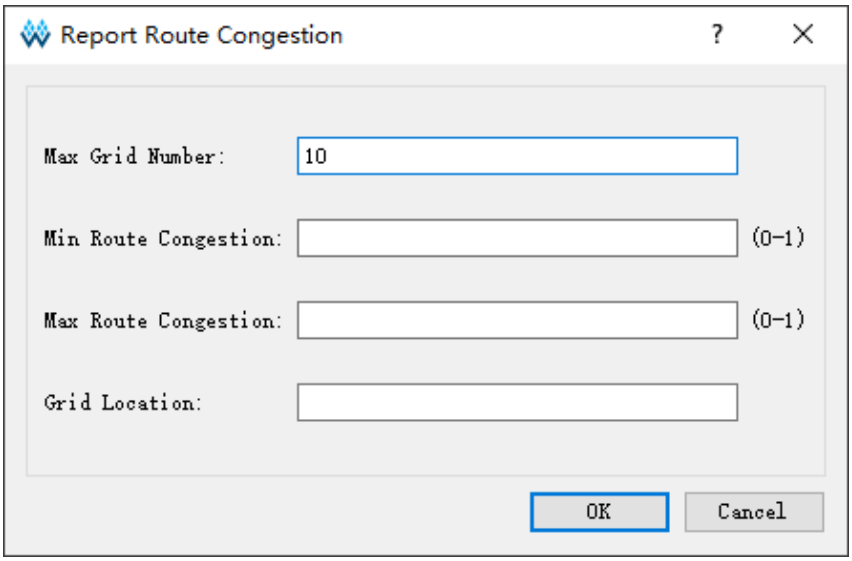
1. Select "Timing Constraints > Report Route Congestion".
2. Right-click and select "Create Report", as shown in Figure 4-31.

Figure 4-31 Report Route Congestion Interface



3. Select "Create Report" and Figure 4-32 pops up.
   - Max Grid Number: Specifies max. number of grid.
   - Min and Max Route Congestion: Specifies the min. and route congestion; accurate to thousandth, floating.
   - Grid Location: Specifies the grid, such as R4C4.

Figure 4-32 Report Route Congestion Dialog Box



4. Click "OK" to save.

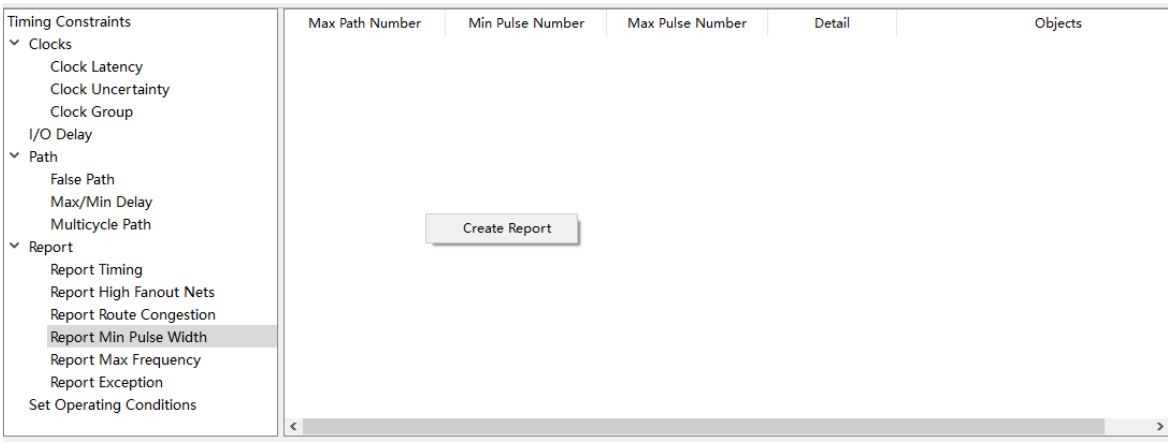
Report Min Pulse Width

It reports the minimum pulse width, 10 by default. You can use this constraint to report a pulse width in a specific range or on a specific object. If a Reg11_Z exists in the design, you can specify Reg11_Z. The results can be viewed in Minimum Pulse Width Report.

The steps are as follows:

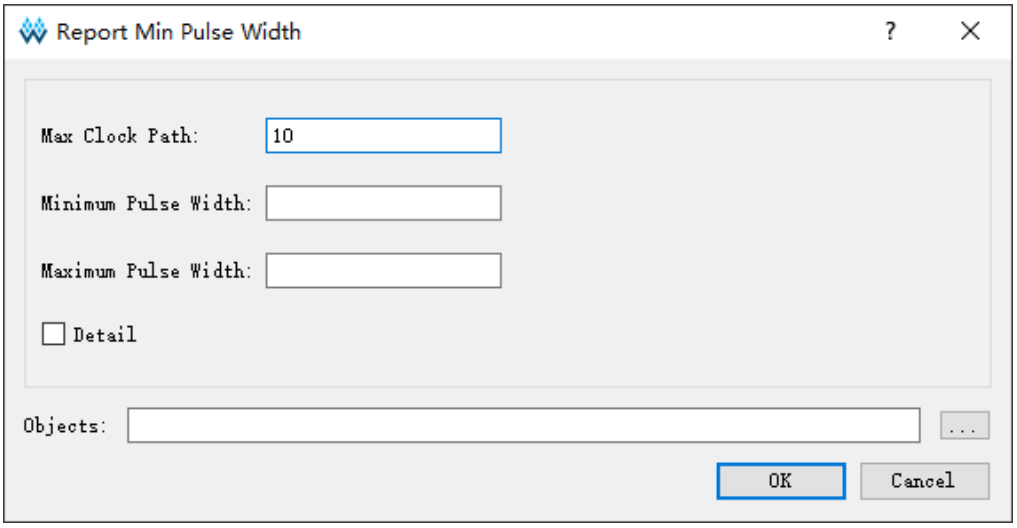
1. Select "Timing Constraints > Report Min Pulse Width".
2. Right-click and select "Create Report", as shown in Figure 4-33.

Figure 4-33 Create Report Min Pulse Width



3. Select "Create Report" and Figure 4-34 pops up.
   - Max Clock Path: Specifies the maximum, positive integer.
   - Minimum and Maximum Pulse Width: Specifies the min. and max. pulse width; accurate to thousandth, floating.
   - Detail: Whether a detailed path is reported.
   - Objects: Specifies the timing component that needs to be reported, only flip flop supported, such as DFF;click "..." to select one.

Figure 4-34 Report Min Pulse Width Dialog Box



4. Click "OK" to save.

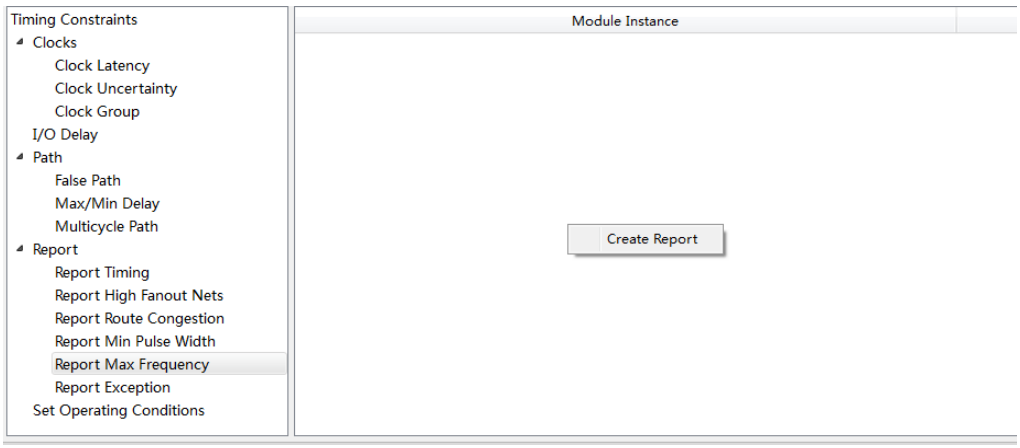
Report Max Frequency

It reports the max. frequency and Gowin Software reports the max. frequency of the top by default. It can report the max. frequency of a specific module, such as the sp_inst in the design, and the module instance can be set to sp_inst. Gowin Software automatically analyzes and reports the max. frequency of a given module, and the results can be viewed in Max Frequency Summary.

The steps are as follows:

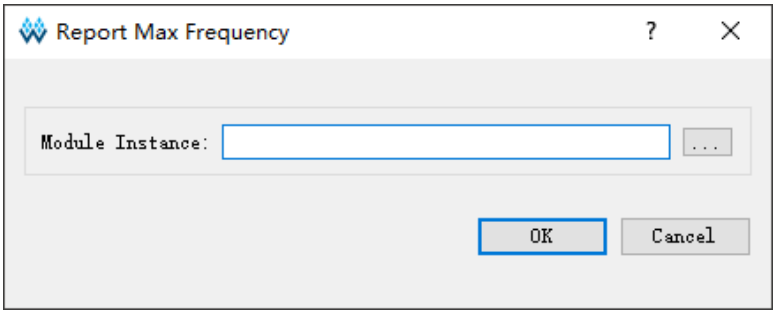
1. Select "Timing Constraints > Report > Report Max Frequency".
2. Right-click and select "Create Report", as shown in Figure 4-35.

Figure 4-35 Create Report Exception



3. Select "Create Report" and Figure 4-36 pops up. Type the instance name in "Module Instance" and click "..." to select one.

Figure 4-36 Report Max Frequency Dialog Box



4. Click "OK" to save.

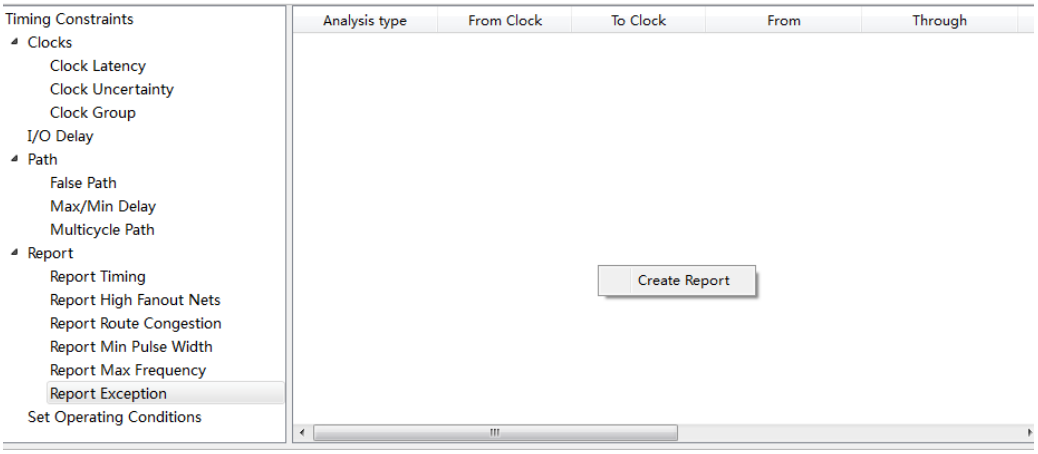
Report Exception

For the report exception, see Timing Exceptions Report.

The steps are as follows.

1. In main interface, select "Timing Constraints > Report > Report Exception".
2. Right-click on the right in the blank and pop up "Create Report", as shown in Figure 4-37.

Figure 4-37 Create Report Exception

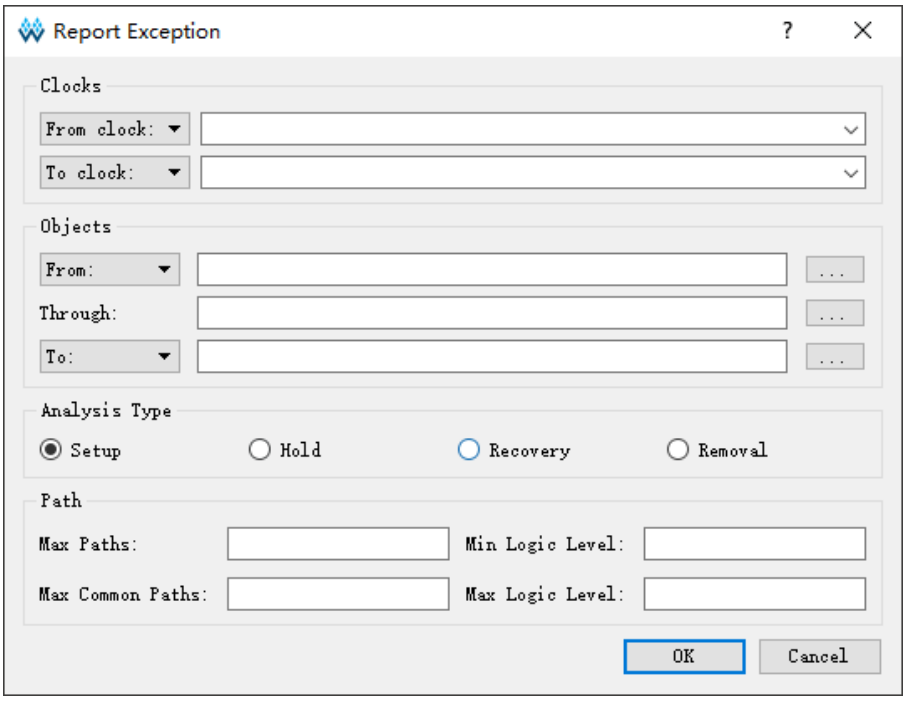


3. Select "Create Report" and Figure 4-38 pops up.

Note!

For the introduction of the options, see Report Timing.

Figure 4-38 Report Exception Dialog Box



4. Click "OK" to save.

4.7.6 Save

After editing all constraints, click "File > Save" or "File > Save As" to save the constraints in .sdc file, see Appedix A Timing Constraints Syntax for details.

4.8 Priority of Timing Constraints

Gowin Software provides multiple types of timing constraints. The following priority is from low to high.

1. create_clock and create_generated_clock
2. set_multicycle_path
3. set_max_delay and set_min_delay
4. set_false_path
5. set_clock_groups

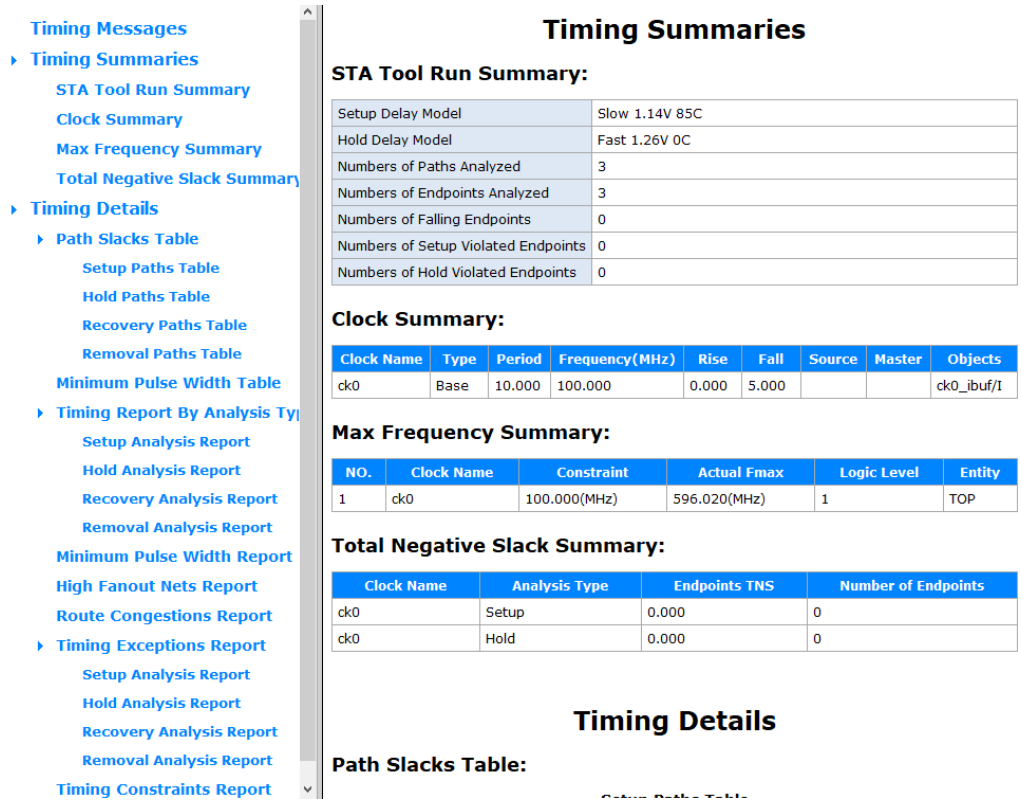
Note!

Only the timing constraints which may produce competition on the same path can be sorted.

5 Timing Report

This chapter will describe Gowin STA to help you learn the timing report. As shown in Figure 5-1, the report includes the navigation bar and the content bar, and the title in the navigation bar will be highlighted in red when there is no analysis.

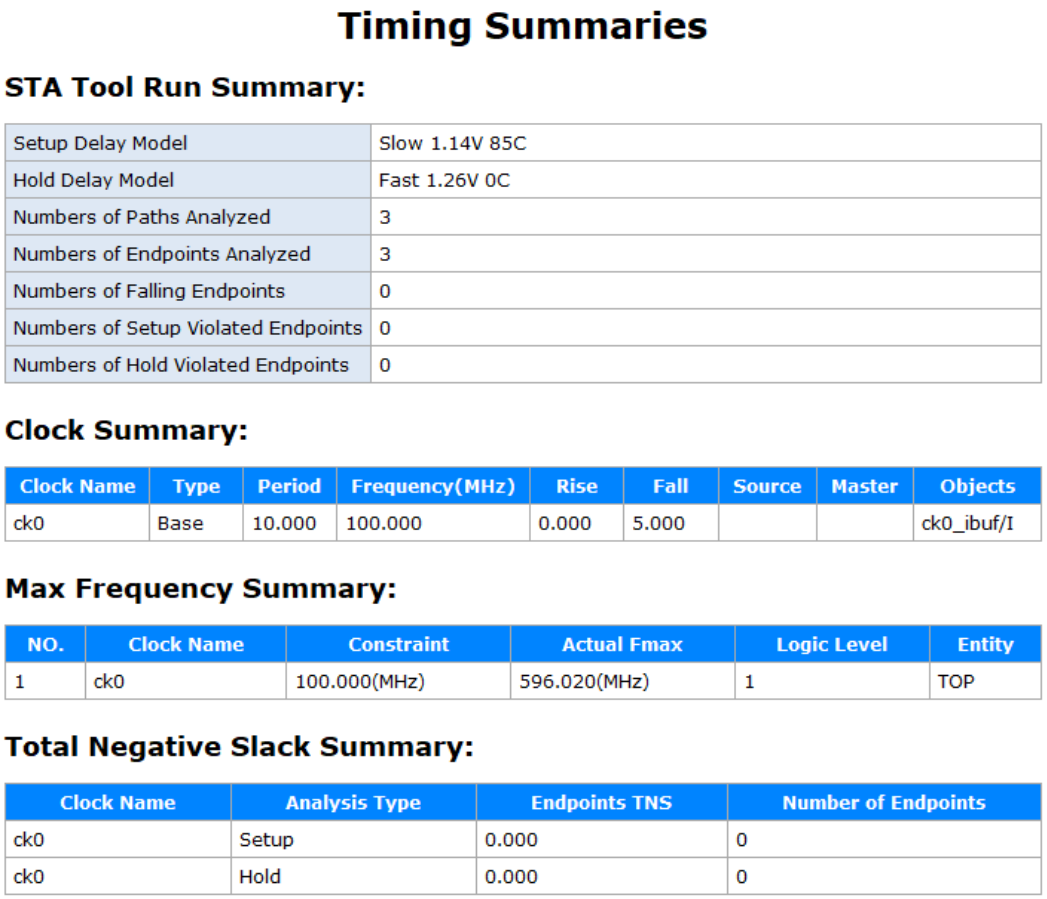
Figure 5-1 Static Timing Analysis Report



5.1 Timing Summaries

Timing Summaries include four parts: STA Tool Run Summary, Clock Summary, Max Frequency Summary and Total Negative Slack Summary, as shown in Figure 5-2.

Figure 5-2 Timing Summaries



Timing Summaries

STA Tool Run Summary:

Setup Delay Model	Slow 1.14V 85C
Hold Delay Model	Fast 1.26V 0C
Numbers of Paths Analyzed	3
Numbers of Endpoints Analyzed	3
Numbers of Falling Endpoints	0
Numbers of Setup Violated Endpoints	0
Numbers of Hold Violated Endpoints	0

Clock Summary:

Clock Name	Type	Period	Frequency(MHz)	Rise	Fall	Source	Master	Objects
ck0	Base	10.000	100.000	0.000	5.000			ck0_ibuf/I

Max Frequency Summary:

NO.	Clock Name	Constraint	Actual Fmax	Logic Level	Entity
1	ck0	100.000(MHz)	596.020(MHz)	1	TOP

Total Negative Slack Summary:

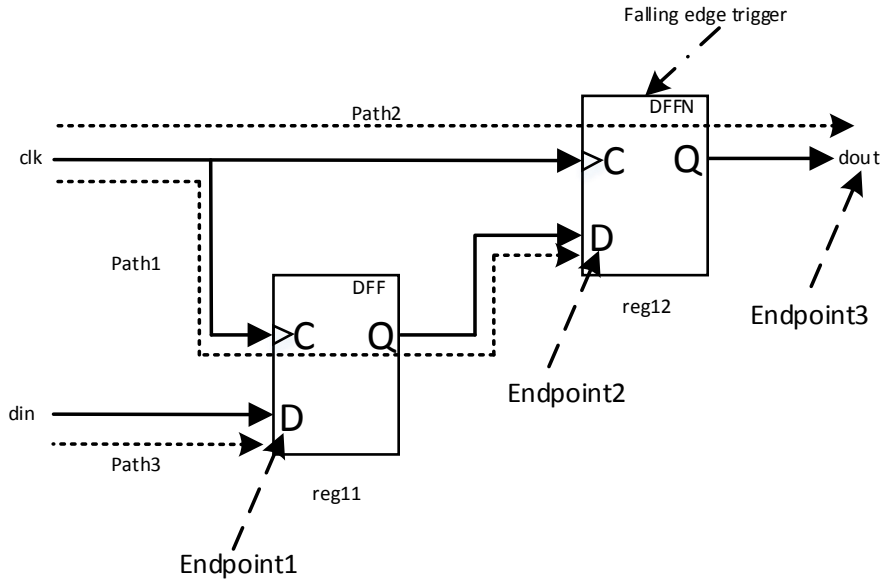
Clock Name	Analysis Type	Endpoints TNS	Number of Endpoints
ck0	Setup	0.000	0
ck0	Hold	0.000	0

5.1.1 STA Tool Run Summary

- Setup Delay Model: Data model for setup analysis used by Gowin Software.
- Hold Delay Model: Data model for hold analysis used by Gowin Software.
- Numbers of Paths Analyzed: The number of static timing analysis paths. As shown in Figure 5-3, three timing paths, Path1, Path2 and Path3, are analyzed.
- Numbers of Endpoints Analyzed: The endpoint of the analysis timing path. As shown in Figure 5-3, three endpoints are analyzed, labeled as Endpoint1, Endpoint2 and Endpoint3.
- Numbers of Falling Endpoints: The number of falling edges triggered of endpoints analysis. As shown in Figure 5-3, if the reg12 is DFFN and the trigger mode is the falling edge, D is the endpoint of falling edge.
- Numbers of Setup Violated Endpoints: The number of the endpoints that can not meet setup after timing analysis.
- Numbers of Hold Violated Endpoints: The number of the endpoints that

can not meet hold after timing analysis.

Figure 5-3 Path & Endpoints



5.1.2 Clock Summary

It reports all clocks in the user design, including the generated clocks.

- Clock Name: The name of the clock.
- Type: Base and Generated. Base represents the base clock and Generated represents the generated clock.
- Period: The clock period.
- Frequency (MHz): Clock frequency, Frequency=1/Period; by default, a clock of Arora family is 100MHz, and a clock of LittleBee® family is 50MHz; if GAO included in the design, TCK clock frequency is 20MHz.
- Rise: Clock rise time.
- Fall: Clock fall time.
- Source: Clock source from PORT, PIN, NET, REG.
- Master: The generated clcok is the master clock.
- Objects: Clock objects such as PORT, PIN, NET, REG.

5.1.3 Max Frequency Summary

- NO.: Item number
- Clock Name: Clock name that drives the timing model.
- Constraint: Clock frequency of SDC or the default clock frequency when there is no SDC constraint.
- Actual Fmax: The max. actual frequency after PnR.

- Logic Level: Logical level of the worst timing paths driven by the clock.
- Entity: The max. frequency of modules. The default is TOP.

Note!

- When there is no drive timing model after PnR, it is "No timing paths to get frequency of *".
- The max. clock frequency only reports the clock on the timing model (including generated clock) driven by the same clock;
- It is recommended to add complete timing constraints to the design.

5.1.4 Total Negative Slack Summary

- Clock Name: The name of the clock.
- Analysis Type: Setup or Hold.
- Endpoints TNS: The time of the endpoints with TNS in timing path driven by clock (ClockName).
- Number of Endpoints: The number of endpoints with TNS in the timing path driven by the clock (ClockName).

5.2 Timing Details

5.2.1 Path Slacks Table

The path slacks table includes Setup Paths Table, Hold Paths Table, Recovery Paths Table, and Removal Paths Table, and the details are shown in Figure 5-4.

- Path Number: Up to 25 by default.
- Path Slack: It is equal to the time of data request minus the time of data arrival, and the timing is not satisfied when it is negative.
- From Node: The start node for timing analysis of the previous level timing component.
- To Node: The end node for timing analysis of the next level timing component.
- From Clock: The clock and edge of the previous level timing component, and the edge type refers to the rising or falling edge.
- To Clock: The latch clock and latch edge of the next level timing component.
- Relation: The time relationship between the transmitting clock and the sampling clock.
- Clock Skew: The time difference between the transmitting clock and the sampling clock to arrive at the previous level and next level timing

components.

- Data Delay: Data delay on the path.

Note!

- It reports "Nothing to report!" when no timing path is available for analysis.
- The worst 25 paths are analyzed by default. If the path you need to check is not within these 25 paths, the SDC constraint command report_timing can be used to report. For the details, see Report Timing.
- Timing path of clock domain crossing is analyzed by default. If you do not care about clock domain crossing analysis, you can configure through set_clk_group or set_false_path. For the details, see Set Clock Group or Set False Path.

Figure 5-4 Path Slacks Table

Path Slacks Table:

Setup Paths Table

Report Command:report_timing -setup -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	8.806	synS_r_s0/Q	synE_r_s0/D	ck0:[R]	ck0:[R]	10.000	0.000	0.794

Hold Paths Table

Report Command:report_timing -hold -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	0.570	synS_r_s0/Q	synE_r_s0/D	ck0:[R]	ck0:[R]	0.000	0.000	0.570

Recovery Paths Table

Report Command:report_timing -recovery -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	8.649	rstSrc_r_s0/Q	rstObj_r_s0/CLEAR	ck0:[R]	ck1:[R]	10.000	0.000	1.278

Removal Paths Table

Report Command:report_timing -removal -max_paths 25 -max_common_paths 1

Path Number	Path Slack	From Node	To Node	From Clock	To Clock	Relation	Clock Skew	Data Delay
1	0.788	rstSrc_r_s0/Q	rstObj_r_s0/CLEAR	ck0:[R]	ck1:[R]	0.000	0.000	0.833

5.2.2 Minimum Pulse Width Table

It is the minimum pulse width table which can be recognized by timing component. Pulse width is the duration of active high/low level signals. The worst 10 paths are reported by default, as shown in Figure 5-5. The table header is described as follows:

- Number: Ascending order, 10 paths by default.
- Slack: The slack value of the minimum pulse width.
- Actual Width: The actual pulse width that the component can recognize in STA after PnR.
- Required Width: The minimum pulse width required by the component.
- Type: Low Pulse Width and High Pulse Width.

- Clock: Clock for minimum pulse width analysis.
- Objects: Instance object of timing component for minimum pulse width analysis.

Note!

It reports as "Nothing to report!" when there is no minimum pulse width analysis report.

Figure 5-5 Minimum Pulse Width Table

Minimum Pulse Width Table:

Report Command:report_min_pulse_width -nworst 10 -detail

Number	Slack	Actual Width	Required Width	Type	Clock	Objects
1	2.738	4.238	1.500	Low Pulse Width	DEFAULT_CLK	reg12
2	2.738	4.238	1.500	Low Pulse Width	DEFAULT_CLK	reg11_Z
3	2.813	4.313	1.500	High Pulse Width	DEFAULT_CLK	reg12
4	2.813	4.313	1.500	High Pulse Width	DEFAULT_CLK	reg11_Z

5.2.3 Timing Report By Analysis Type

This section includes Setup Analysis Report, Hold Analysis Report, Recovery Analysis Report and Removal Analysis Report, where Setup Analysis Report includs Recovery Analysis Report and Hold Analysis Report includes Removal Analysis Report. The analysis methods are consistent. The four types of analysis are described below.

Setup Analysis Report

The setup analysis report is used to analyze the time to stablize data before the clock rising edge arrives. If the time is not enough, the data will not be stably transmitted to the timing component at the clock rising edge.

Gowin Software calculates, analyzes and prints the arrival time, request time, sampling clock and transmitting clock on the path for reference.

The report is generated by the command report_timing -setup. Gowin Software analyzes and reports the 25 timing paths with the worst slack by default, including Path Summary, Data Arrival Path and Path Statistics.

1. Path Summary, as shown in Figure 5-6, is a summary of the path:
   - Slack: The latest time of arrival minus the actual time of arrival of the data. Positive value indicates timing closure, and negative value indicates timing non-closure.
   - Data Arrival Time: The time of launch edge to arrive at the data port of the next level timing component.
   - Data Required Time: The time of latch edge to arrive at the clock port of the next level timing component.
   - From: The previous level timing component.

- To: The next level timing component.
- Launch Clock: The clock that provides the launch edge and the edge tpye. The edge includes R (Rise) and F (Fall).
- Latch Clock: The clock that provides the latch edge and the edge tpye. The edge includes R and F.

Figure 5-6 Path Summary

Path Summary:

Slack	5.789
Data Arrival Time	6.767
Data Required Time	12.556
From	reg11_Z
To	reg12_Z
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

2. Data Arrival Path, as shown in Figure 5-7, is the path of data arriva, and the header is described below:
   - At: A time node on the timing path.
   - DELAY: A delay value meaning a time interval.
   - TYPE: The type of node on the timing path, which is not available when null.

   Note!

   In Figure 5-7, TYPE descriptions are as follows:
   - tCL: Time of clock latency; clock source latency.
   - tINS: Time of module instance; instantiated component delay.
   - tNET: Time of net; the delay of net.
   - tC2Q: Time of clock to quit; the internal delay of timing component.
   - RF: The inverse type. RR indicates the positive pulse and not inverse; FF indicates the negative pulse and not inverse; F indicates that the positive pulse is inversed to the negative pulse, and FR indicates that the negative pulse is inversed to the positive pulse.
   - FANOUT: Fanout.
   - LOC: The physical position of the currently analyzed component in the device, and UNPLACE means no location, such as DHCEN.
   - NODE: Node on the static timing analysis path, including instance name, port, clock, and active clock edge time.

Figure 5-7 Data Arrival Path

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
0.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
3.236	2.293	tNET	RR	1	IOL2[B]	reg11_Z/CLK
3.786	0.550	tC2Q	RF	1	IOL2[B]	**reg11_Z/Q**
6.767	2.981	tNET	FF	1	R5C9[1][A]	**reg12_Z/D**

3. Data Required Path: The data required path is the path through which the clock reaches the clock port from the active edge. As shown in Figure 5-8, the header matches the data arrival path.

Figure 5-8 Data Required Path

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysclk1
10.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
10.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
13.236	2.293	tNET	RR	1	R5C9[1][A]	reg12_Z/CLK
13.036	-0.200	tUnc				reg12_Z
12.556	-0.480	tSu		1	R5C9[1][A]	reg12_Z

4. Path Statistics is the statistics of the path, as shown in Figure 5-9.
   - Clock Skew: Clock skew.
   - Setup Relationship: The time relationship between the previous level timing component transmitting data and the next level timing component latching data.
   - Logic Level: The number of logic levels; 0 refers to a direct connection.
   - Arrival Clock Path Delay: The clock delay on the Data Arrival Path. Cell indicates the logical delay; Route indicates the route delay, and tC2Q indicates the internal delay.
   - Arrival Data Path Delay: The data delay on the Data Arrival Path.
   - Required Clock Path Delay: The clock delay on the Data Required Path.

Figure 5-9 Path Statistics

Path Statistics:

Clock Skew	0.000
Setup Relationship	10.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 2.981, 84.423%; tC2Q: 0.550, 15.577%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Hold Analysis Report

Figure 5-10 is hold analysis report, which is used to analyze the time to stablize data after the the clock rising edge arrives. If the time is not enough, the data will not be stably transmitted to the timing component. Gowin Software calculates and analyzes the arrival time, request time, sampling clock and transmitting clock on the path. The report is generated by the command report_timing - hold. It reports the 25 timing paths with the worst slack by default. For the report header, see Setup Analysis Report.

Figure 5-10 Hold Analysis Report

Hold Analysis Report

Report Command:report_timing -hold -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	1.003
Data Arrival Time	3.554
Data Required Time	2.551
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
2.933	0.400	tC2Q	RR	1	R2C9[0][A]	**reg11_s0/Q**
3.554	0.621	tNET	RR	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
2.533	0.000	tUnc				reg12_s0
2.551	0.018	tHld		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Hold Relationship	0.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.621, 60.818%; tC2Q: 0.400, 39.182%
Required Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%

Recovery Analysis Report

Figure 5-11 is the recovery time report, which analyzes the shortest time to keep stable for signal to remove asynchronous reset before clock active edge. If the time is not met, the flip flop may not operate. The report is generated by the command report_timing -recovery. It reports the 25 timing paths with the worst slack by default. For the table header, see Setup Analysis Report.

Figure 5-11 Recovery Analysis Report

Recovery Analysis Report

Report Command:report_timing -recovery -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	8.355
Data Arrival Time	4.629
Data Required Time	12.984
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.943	0.943	tINS	RR	2	IOL11[A]	clk_ibuf/O
3.236	2.293	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
3.786	0.550	tC2Q	RF	1	R2C9[0][A]	**reg11_s0/Q**
4.629	0.843	tNET	FF	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
10.000	10.000					active clock edge time
10.000	0.000					sysyclk
10.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
10.943	0.943	tINS	RR	2	IOL11[A]	clk_ibuf/O
13.236	2.293	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
13.036	-0.200	tUnc				reg12_s0
12.984	-0.052	tSu		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Setup Relationship	10.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.843, 60.531%; tC2Q: 0.550, 39.469%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Removal Analysis Report

Figure 5-12 is removal time report, which analyzes the shortest hold time to remove asynchronous reset signal after the timing component is on the positive edge. If the time is not met, the flip flop may not operate. The analysis and calculation is the same as the hold time. The report is generated by the command report_timing - removal. It reports the 25 timing paths with the worst slack by default. For the table header, see Hold Analysis Report.

Figure 5-12 Removal Analysis Report

Removal Analysis Report

Report Command:report_timing -removal -max_paths 25 -max_common_paths 1

Path1

Path Summary:

Slack	1.003
Data Arrival Time	3.554
Data Required Time	2.551
From	reg11_s0
To	reg12_s0
Launch Clk	sysyclk:[R]
Latch Clk	sysyclk:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[0][A]	reg11_s0/CLK
2.933	0.400	tC2Q	RR	1	R2C9[0][A]	**reg11_s0/Q**
3.554	0.621	tNET	RR	1	R2C9[1][A]	**reg12_s0/CLEAR**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysyclk
0.000	0.000	tCL	RR	1	IOL11[A]	clk_ibuf/I
0.811	0.811	tINS	RR	2	IOL11[A]	clk_ibuf/O
2.533	1.723	tNET	RR	1	R2C9[1][A]	reg12_s0/CLK
2.533	0.000	tUnc				reg12_s0
2.551	0.018	tHld		1	R2C9[1][A]	reg12_s0

Path Statistics:

Clock Skew	0.000
Hold Relationship	0.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 0.621, 60.818%; tC2Q: 0.400, 39.182%
Required Clock Path Delay	cell: 0.811, 31.998%; route: 1.723, 68.002%

5.2.4 Minimum Pulse Width Report

The minimum pulse width report analyzes all the minimum pulse width on the timing analysis path. As shown in Figure 5-13, the description is as follows.

- Actual Width: The actual pulse width, whose value is Early clock Path minus Late clock Path.
- Required Width: The minimum width required by the component. If it is less than that width, the low level pulse will not be recognized.
- Slack: The actual pulse width minus the required pulse width.
- Type: The pulse type. Low Pulse Width and High Pulse Width.
- Clock: Clock for STA.
- Objects: The current objects.
- Late clock Path: For the high pulse width, it is the path from which the logic high signal starts. For the low pulse width, it is the path from which the logic low signal starts.

- Early clock Path: For the high pulse width, it is the path from which the logic high signal ends. For the low pulse width, it is the path from which the logic low signal ends.

Figure 5-13 Minimum Pulse Width

MPW Summary:

Slack:	2.738
Actual Width:	4.238
Required Width:	1.500
Type:	Low Pulse Width
Clock:	sysclk1
Objects:	reg12_Z

Late clock Path:

AT	DELAY	TYPE	RF	NODE
5.000	0.000			active clock edge time
5.000	0.000			sysclk1
5.000	0.000	tCL	FF	clk1_ibuf/I
5.945	0.945	tINS	FF	clk1_ibuf/O
8.295	2.350	tNET	FF	reg12_Z/CLK

Early clock Path:

AT	DELAY	TYPE	RF	NODE
10.000	0.000			active clock edge time
10.000	0.000			sysclk1
10.000	0.000	tCL	RR	clk1_ibuf/I
10.811	0.811	tINS	RR	clk1_ibuf/O
12.533	1.723	tNET	RR	reg12_Z/CLK

5.2.5 High Fanout Nets Report

High fanout nets report analyzes all the net fanout on the timing analysis path, and also the worst slack and max. delay. The default value is 10. As shown in Figure 5-14, the description is as follows:

- FANOUT: The fanout of the net.
- NET NAME: The net name.
- WORST SLACK: The worst slack on the net and more than one slack may on one net.
- MAX DELAY: The max. delay on the net.

Figure 5-14 High Fanout Nets Report

High Fanout Nets Report:

Report Command:report_high_fanout_nets -max_nets 10

FANOUT	NET NAME	WORST SLACK	MAX DELAY
2	clk1_c	5.789	2.350
2	clk2_c	17.616	2.350
1	reg21_i	17.616	0.000
1	reg11	5.789	2.981
1	reg21	17.616	0.403

5.2.6 Route Congestions Report

As shown in Figure 5-15, the description is as follows:

- GRID LOC: The location of grid.
- ROUTE CONGESTIONS: The route congestion on the grid, such as 0.056, indicating that the route congestion is 5.6%.
- It reports 10 of the worst by default, in descending order.

Figure 5-15 Route Congestions Report

Route Congestions Report:

Report Command:report_route_congestion -max_grids 10

GRID LOC	ROUTE CONGESTIONS
R5C9	0.056
R2C1	0.028
R3C1	0.028
R3C9	0.028
R1C1	0.014
R5C1	0.014

5.2.7 Timing Exceptions Report

The timing exception allows you to modify the static timing analysis of a specific path. The timing exception constraints commands include set_false_path, set_multicycle_path, set_max_delay and set_min_delay. This is illustrated by a case.

For the case in Figure 5-16, design a specific SDC file, as shown in Figure 5-17.

Figure 5-16 Test Case

```
module timing(
output dout,
input din, clk1, clk2
);

reg reg11, reg12;
reg reg21, reg22;


always @(posedge clk1)
begin
    reg11 <= din;
    reg12 <= reg11;
end

always @(posedge clk2)
begin
    reg21 <= din;
    reg22 <= ~reg21;
end

assign dout = reg22 & reg12;

endmodule
```

Figure 5-17 Timing Exceptions Constraints

```
create_clock -name sysclk1 -period 10 -waveform {0 5} [get_ports {clk1}]
create_clock -name sysclk2 -period 10 -waveform {0 5} [get_ports {clk2}]
set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5
set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4
```

The set_max_delay in Figure 5-17 is to set sysclk1 and sysclk2 to 5ns and 4ns respectively. The set_max_delay affects the setup analysis and the affected path is displayed by default in the timing exceptions report, and the generated report is shown in Figure 5-18.

Figure 5-18 Timing Exceptions Report

Timing Exceptions Report:

Setup Analysis Report

Report Command:report_exceptions -setup -max_paths 5 -max_common_paths 1

Timing Path Constraint[1]: set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5

Path1

Path Summary:

Slack	0.789
Data Arrival Time	6.767
Data Required Time	7.556
From	reg11_Z
To	reg12_Z
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
0.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
3.236	2.293	tNET	RR	1	IOL2[B]	reg11_Z/CLK
3.786	0.550	tC2Q	RF	1	IOL2[B]	**reg11_Z/Q**
6.767	2.981	tNET	FF	1	R5C9[1][A]	**reg12_Z/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
5.000	5.000					active clock edge time
5.000	0.000					sysclk1
5.000	0.000	tCL	RR	1	IOL7[A]	clk1_ibuf/I
5.943	0.943	tINS	RR	2	IOL7[A]	clk1_ibuf/O
8.236	2.293	tNET	RR	1	R5C9[1][A]	reg12_Z/CLK
8.036	-0.200	tUnc				reg12_Z
7.556	-0.480	tSu		1	R5C9[1][A]	reg12_Z

Path Statistics:

Clock Skew	0.000
Setup Relationship	5.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 2.981, 84.423%; tC2Q: 0.550, 15.577%
Required Clock Path Delay	cell: 0.943, 29.131%; route: 2.293, 70.869%

Timing Path Constraint[14]: set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4

Path1

Path Summary:

Slack	1.616
Data Arrival Time	4.940
Data Required Time	6.556
From	reg21_Z
To	reg22_Z
Launch Clk	sysclk2:[R]
Latch Clk	sysclk2:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk2
0.000	0.000	tCL	RR	1	IOL5[A]	clk2_ibuf/I
0.943	0.943	tINS	RR	2	IOL5[A]	clk2_ibuf/O
3.236	2.293	tNET	RR	1	R5C9[0][B]	reg21_Z/CLK
3.786	0.550	tC2Q	RR	1	R5C9[0][B]	**reg21_Z/Q**
4.189	0.403	tNET	RR	1	R5C9[0][A]	reg21_i_cZ/I0
4.940	0.751	tINS	RF	1	R5C9[0][A]	reg21_i_cZ/F
4.940	0.000	tNET	FF	1	R5C9[0][A]	**reg22_Z/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
4.000	4.000					active clock edge time
4.000	0.000					sysclk2
4.000	0.000	tCL	RR	1	IOL5[A]	clk2_ibuf/I
4.943	0.943	tINS	RR	2	IOL5[A]	clk2_ibuf/O
7.236	2.293	tNET	RR	1	R5C9[0][A]	reg22_Z/CLK

The timing exceptions report defaults to report all exceptions paths,

and Gowin Software provides the report_exception constraints command, which allows you to configure and display some of the contents that are concerned and filter the paths that are not concerned.

Add the report_exception statement on the basis of Figure 5-17, as shown in Figure 5-19, the first line in the red box indicates that the path affected by sysclk1 reports a setup analysis, and the second line indicates that the path affected by sysclk2 does not report a setup analysis.

Figure 5-19 report_exception Statement

```
create_clock -name sysclk1 -period 10 -waveform {0 5} [get_ports {clk1}]
create_clock -name sysclk2 -period 10 -waveform {0 5} [get_ports {clk2}]
set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5
set_max_delay -from [get_clocks {sysclk2}] -to [get_clocks {sysclk2}] 4
report_exceptions -setup -from_clock [get_clocks {sysclk1}] -to_clock [get_clocks {sysclk1}] -max_paths 1 -max_common_paths 1
report_exceptions -setup -from_clock [get_clocks {sysclk2}] -to_clock [get_clocks {sysclk2}] -max_paths 0 -max_common_paths 0
```

After constraints as shown in Figure 5-19, the timing exceptions report is as shown in Figure 5-20.

Figure 5-20 report_exception Report

Timing Exceptions Report:

Setup Analysis Report

Setup Analysis Report[1]:

Report Command:report_exceptions -setup -from_clock [get_clocks {sysclk1}] -to_clock [get_clocks {sysclk1}] -max_paths 1 -max_common_paths 1

Timing Path Constraint[1]: set_max_delay -from [get_clocks {sysclk1}] -to [get_clocks {sysclk1}] 5

Path1

Path Summary:

Slack	-0.654
Data Arrival Time	7.947
Data Required Time	7.293
From	reg11_ins23
To	reg12_ins20
Launch Clk	sysclk1:[R]
Latch Clk	sysclk1:[R]

Data Arrival Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
0.000	0.000					active clock edge time
0.000	0.000					sysclk1
0.000	0.000	tCL	RR	1	IOL15[A]	clk1_ibuf13/I
0.982	0.982	tINS	RR	2	IOL15[A]	clk1_ibuf13/O
2.893	1.911	tNET	RR	1	IOL2[B]	reg11_ins23/CLK
3.351	0.458	tC2Q	RF	1	IOL2[B]	**reg11_ins23/Q**
7.947	4.596	tNET	FF	1	R15C23[1][A]	**reg12_ins20/D**

Data Required Path:

AT	DELAY	TYPE	RF	FANOUT	LOC	NODE
5.000	5.000					active clock edge time
5.000	0.000					sysclk1
5.000	0.000	tCL	RR	1	IOL15[A]	clk1_ibuf13/I
5.982	0.982	tINS	RR	2	IOL15[A]	clk1_ibuf13/O
7.893	1.911	tNET	RR	1	R15C23[1][A]	reg12_ins20/CLK
7.693	-0.200	tUnc				reg12_ins20
7.293	-0.400	tSu		1	R15C23[1][A]	reg12_ins20

Path Statistics:

Clock Skew	0.000
Setup Relationship	5.000
Logic Level	1
Arrival Clock Path Delay	cell: 0.982, 33.942%; route: 1.911, 66.058%
Arrival Data Path Delay	cell: 0.000, 0.000%; route: 4.596, 90.932%; tC2Q: 0.458, 9.068%
Required Clock Path Delay	cell: 0.982, 33.942%; route: 1.911, 66.058%

5.2.8 Timing Constraints Report

As shown in Figure 5-21, the description is as follows:

- SDC Command Type includes TC_CLOCK, TC_GENERATED_CLOCK, TC_INPUT_DELAY, TC_CLOCK_LATENCY, TC_CLOCK_UNCERTAINTY, TC_FALSE_PATH, TC_MULTICYCLE, TC_MAX_DELAY, TC_CLOCK_GROUP. When the value is null, it means it is not available.
- State: Invalid and Actived. Actived indicates that the command takes effect, and Invalid indicates that the command is invalid.
- Detail Command: The value is equal to the corresponding timing constraints statement in the SDC file.

Note!

Invalid SDC command statements are not counted in Timing Constraints Report.

Figure 5-21 Timing Constraints Report

Timing Constraints Report:

SDC Command Type	State	Detail Command
TC_CLOCK	Actived	create_clock -name main -period 18.182 -waveform {0 9.091} [get_ports {clk}]
TC_GENERATED_CLOCK	Actived	create_generated_clock -name main_gen -source [get_ports {clk}] -master_clock main -divide_by 5 -duty_cycle 40 -phase 22 -offset 50 [get_ports {in}]
TC_INPUT_DELAY	Actived	set_input_delay -clock main_gen 0.2 -clock_fall -add_delay -source_latency_included [get_ports {in}]
TC_CLOCK_LATENCY	Actived	set_clock_latency -source 1.2 [get_clocks {main}]
TC_CLOCK_UNCERTAINTY	Actived	set_clock_uncertainty 2.3 -setup -from [get_clocks {main}] -to [get_clocks {main}]
TC_FALSE_PATH	Actived	set_false_path -from [get_clocks {main_gen}] -to [get_clocks {main_gen}]
TC_MULTICYCLE	Actived	set_multicycle_path -from [get_clocks {main_gen}] -to [get_clocks {main_gen}] -setup -end 3
TC_MAX_DELAY	Actived	set_max_delay -from [get_clocks {main}] -to [get_clocks {main}] 1.11
TC_CLOCK_GROUP	Actived	set_clock_groups -exclusive -group [get_clocks {main}] -group [get_clocks {main_gen}]
	Actived	report_timing -setup -from_clock [get_clocks {main}] -to_clock [get_clocks {main}]
	Actived	report_exceptions -setup -from_clock [get_clocks {main}] -to_clock [get_clocks {main}]

Appedix A Timing Constraints Syntax Specification

Gowin timing constraints syntax specification references the standard SDC (Synopsys Design Constraint), which provides efficient timing constraints on designs to meet specific timing requirements. At the same time, it supports the wildcard character "?" and "*"; "?" matches a single character, and "*" matches zero or more characters.

Note!

- Wildcards are special characters, use the character "\" to escape if there is a syntax error.
- In the syntax descriptions, [] means it is optional.

A.1 Clock Constraints

A.1.1 create_clock

Syntax

Command: create_clock

Parameter: -period <period_value>

[-name <clock_name>]

[-waveform <edge_list>]

<objects>

[-add]

-period: Specify the period of clock. The parameter value should be greater than 0, and the period unit is ns.

-name: Specify the clock name. The clock name must be unique. If the

new created clock has the same name with already created clock, the already created clock will be overwritten with new created clock. If the name is not specified, the name of first source objects will be regarded as clock name.

-waveform: Specify the time of the rising edge and falling edge of clock. The difference time between rising edge and falling edge should be less than one period. In general, if the rising edge arrives first, both the rising edge time and the falling edge time should be less than one period. For example, "{0 5}" means the clock rising edge arrives at 0ns, and the clock falling edge arrives at 5ns; if the clock falling edge arrives, set the clock rising edge time to less than one period, and the falling edge time to equal to or greater than one period. If the period is set to 10ns, "-waveform {5 10}" means the clock falling edge arrives at 0ns, and the clock falling edge arrives at 5ns.

-add: Use -add option to add multiple clocks to the same source object, or the new created clock with different clock name on the same source object will be ignored when the source object already has one created clock.

<objects> Specify the object of the clock, such as get_ports, get_pins, get_nets, and get_regs,etc. When the source object already has one created clock, you can create new clock with -add command. If source object is not specified when you create clock with create_clock command, Gowin Software will ignore this command.

Examples

Wildcard "?" matches a character such as clk, cck.

create_clock -name ck -period 100 -waveform {0 50} [get_ports {c?k}]

Wildcard "*" matches zero or more single characters such as clk, clock.

create_clock -name ck -period 100 -waveform {0 50} [get_ports {c*k}

Wildcard "*" matches uut/rpll_inst/CLKOUT.

create_clock -name cck0 -period 25 -waveform {0 12.5} [get_pins {uut/r*_inst/CLKOUT}

Use escaped characters to avoid syntax errors.

create_clock -name cck -period 25 -waveform {0 12.5} [get_pins {uut/*ll_inst/CLKOUT}]

create_clock -name cck -period 25 -waveform {0 12.5} [get_pins {\?ut/rpll_inst/CLKOUT}]

create a clock which period is 10 ns, and the falling edge arrives first; the default name is clk.

create_clock -period 10.000 -waveform {5 10} [get_ports {clk}]

create a clock with the duty cycle of 40%

create_clock -name clk -period 10.000 -waveform {6 10} [get_ports {clk}]

create two clocks to one input port

- create_clock -period 10 -name clk [get_ports {clk}] # create clk successfully
- create_clock -period 10 -name clk1 [get_ports {clk}] # commands is ignored and no clk can be created because -add is not used.

create_clock -period 20 -name clk1 -add [get_ports {clk}]# create clk1 successfully

A.1.2 create_generated_clock

Syntax

Command: create_generated_clock

Prameter: [-name <clock name>]

-source <master pin>

[-edges <edge list>]

[-edge_shift <shift list>]

[-divide_by <factor>]

[-multiply_by<factor>]

[-duty_cycle <percent>]

[-add]

[-invert]

[-master_clock <clock>]

[-phase <phase>]

[-offset <offset>]

<objects>

-name: Specify the name of the generated clock. If –name is not specified, the name of first source object will be regarded as clock name, the clock name must be unique. If the new created clock has the same name with the already created clock, the already created clock will be overwritten with new created clock.

-source: Specify the source where generated clock is from; if there are more than one clocks, master_clock option must be used to specify the master clock; and it supports get_ports, get_pins, get_nets and get_regs.

-master_clock: Specify the master clock of the generated clock.

-edges: Specify the edge list of generated clock; this option specifies a three positive ascending order integer parameter list, which indicates the relationship between the first rising edge of generated clock, the first falling edge of generated clock, the second rising edge of generated clock and the master clock edge. For instance, we mark the first rising edge of master clock as 1, the next falling edge is 2, the next rising edge is 3 …, the marker numbers of master clock edges are elements of edge list; we can create a 2 divider generated clock with "-edges {1 3 5}".

-edge_shift: Specify the edge shift of each edge, should be used together with "-edges" option. It can be specified as any number, but the edge can not go beyond the adjacent boerder.

Note!

"-edge" and "-edge_shift" can not be used together with the other parameters used for waveform adjustment, except "-invert".

-divide_by: Specify the divider value.

-multiply_by: Specify the multiplier value.

-duty_cycle: Specify the duty cycle of generated clock.

-add: When specify this option, this clock can coexist with existing clock.

-invert: Specify if invert the waveform of the generated clock.

-phase: Specify the phase shift of clock edges.

-offset: Specify the offset of clock edges**.**

<objects>: Specify the input port of clock; it supports the set of get_ports, get_pins, get_nets and get_regs.

Examples

Create a 2*divider generated clock to port a by using "-divide_by".

create_clock -period 10 [get_ports clk]

create_generated_clock -name genClk -source [get_ports {clk}] -divide_by 2 [get_ports {a}]

Create a 2*divider generated clock to port a by using "-edges".

create_generated_clock -name genClk -source [get_ports {clk}] -edges {1 3 5} [get_ports {a}]

Create a 2*multiplier clock with a 40% duty cycle

create_generated_clock -name genClk0 -source [get_ports {clk}] -multiply_by 2 -duty_cycle 40 [get_pins {pll_out}]

Create an inverted clock 2*divider relative to the output of the source clock

create_generated_clock -name genClk1 -source [get_ports {clk}] -divide_by 2 -invert [get_pins {pll_out}]

Create a clock 2*multiplier with a 90-degree phase shift

create_generated_clock -name genClk2 -source [get_ports{clk}] -multiply_by 2 -phase 90[get_pins {pll_out}]

Create a 2*divider generated clock

create_generated_clock -name genClk3 -source [get_ports {clk}] -edges {2 4 6}[get_pins {pll_out}]

#Create two clocks to an input port that are switched externally

create_clock -period 10 -name clk [get_ports {clk}]

create_clock -period 20 -name clk1 -add [get_ports {clk}]

create_generated_clock -name genClk -source [get_ports {clk}] -divide_by 2 -master_clock clk -add [get_pins {pll_out}]

create_generated_clock -name genClk1 -source [get_ports {clk}] -master_clock clk1 -divide_by 2 -add [get_pins {pll_out}]

A.1.3 set_clock_latency

Syntax

Command: set_clock_latency

Parameter: -source [-rise | -fall]

[-late | -early]

<delay>

[-clock <clock list>]

<object list>

-source: Indicate the clock source delay, mandatory.

-rise | -fall: Specify the rising | falling clock latency. -rise| -fall can not be specified at one statement. If both are not specified, the clock latency is applied to all conditions.

-late | -early: Specify the max.or min. latency; For setup analysis, late acts on the launch clock and early acts on the latch clock, while for hold analysis, it is the opposite of setup.

<delay>: Specify the clock source latency value, and the default is 0.

Note!

The value of late should be greater than or equal to the value of early, otherwise late will be calculated according to the value of early.

-clock: You can specify the clock which source latency affected on when more than one clocks are on one source object, if this parameter is not configured, all clock have same delay; and it supports the set of get_clocks.

<source objects>: Specify the delay source objects; and it supports get_clocks, get_ports, get_pins, get_nets and get_regs.

Examples

```
create_clock -period 10 -name clk [get_ports {clk}]
create_clock -period 10 -name clk0 [get_ports {clk}] -add
# Specify 2ns clock latency for clk
set_clock_latency -source 2 [get_clocks {clk}]
# Specify 2ns clock latency for clk0
set_clock_latency -source 2 -clock [get_clocks {clk0}] [get_ports {clk}]
# Set the rising edge clock source delay of clock cck on clk0, and the latest and earliest values are 0.111, 0.011 respectively.
set_clock_latency -source -rise -late 0.111 [get_ports {clk0}]  -clock [get_clocks {cck}]
set_clock_latency -source -rise -early 0.011 [get_ports {clk0}]  -clock [get_clocks {cck}]
# Set the falling edge clock source delay of clock cck on clk0, and the latest and earliest values are 0.222, 0.022 respectively.
set_clock_latency -source -fall -late 0.222 [get_ports {clk0}]  -clock [get_clocks {cck}]
set_clock_latency -source -fall -early 0.022 [get_ports {clk0}]  -clock [get_clocks {cck}]
# Wildcard matches uut/rpll_inst/CLKOUT
set_clock_latency -source 0.123 [get_pins {u?t/r*_inst/\*OUT}] -clock [get_clocks {cck0}]
```

A.1.4 set_clock_uncertainty

Syntax

Command: set_clock_uncertainty

Parameter: [-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>]

[-setup | -hold]

<uncertainty value>

-from/-rise_from/-fall_from: Specify the start clock of uncertainty; "-rise_from" and "-fall_from" specify the valid start clock edge of uncertainty, supporting the set of get_clocks.

-to/-rise_to/-fall_to: Specify the end clock of uncertainty; "-rise_to" and "-fall_to" specify the valid clock edge of the end of uncertainty, supporting the set of get_clocks.

-setup/-hold: Specify whether the uncertainty has an effect on the setup time or hold time; it is exclusive for the same constraint statement; if neither is specified, both checks are valid.

<uncertainty value>: Uncertainty value

Note!

At least one launch clock or latch clock must be specified, otherwise the constraints are invalid.

Examples

```
# Set the clock uncertainty setup time from clk to clk to 0.5.
set_clock_uncertainty -setup -from clk -to clk 0.5
# Set the clock uncertainty hold time from clk0 to clk to 0.0.
set_clock_uncertainty -hold -from clk0 -to clk 0.0
#Set hold and setup uncertainty of clk0 launch to 0.111, 0.222.
set_clock_uncertainty 0.222 -setup -from [get_clocks {clk0}]
set_clock_uncertainty 0.111 -hold -from [get_clocks {clk0}]
# Set hold and setup uncertainty of clk1 launch to 0.111, 0.222.
```

set_clock_uncertainty 0.222 -setup -to [get_clocks {clk1}]

set_clock_uncertainty 0.111 -hold -to [get_clocks {clk1}]

Set hold and setup uncertainty of clk launch to 0.111.

set_clock_uncertainty 0.111 -from [get_clocks {clk}]

A.1.5 set_clock_groups

Syntax

Command: set_clock_groups

Parameter: [-asynchronous | -Exclusive]

[-group <clock name>].

-asynchronous | -Exclusive: Specify the clocks, asynchronous or exclusive.

-group: Specify the clocks as the same group, supporting the collection of one or more clocks using the set of get_clocks

Examples

Set the relationship between clk and clk0 as exclusive.

set_clock_groups -exclusive -group [get_clocks {clk}] -group [get_clocks {clk0}]

A.2 I/O Delay Constraints

A.2.1 set_input_delay

Syntax

Command: set_input_delay

Parameter: -clock clock_name

[-clock_fall]

[-rise]

[-fall]

[-max]

[-min]

[-add_delay]

[-source_latency_included]

<delay_value>

<port_list>

-clock: Specify which clock is associated with this input port.

-clock_fall: Specify that the input delay is relative to falling clock edge.

If there is no this parameter, the input delay is relative to rising clock edge by default.

-rise/-fall: Specify rise/fall input delay; If only one of them is specified, the specified input delay is applied to the other.

-max/-min: Specify max./min. input delay, affecting setup, hold, respectively; If only one of them is specified, the specified input delay is applied to the other.

-add_delay: When this option is specified, already input constraints will not be overwritten.

-source_latency_included: specify that the input delay has already included source latency.

If not specified, the external clock delay is not included in the input delay.

<delay_value>: Specify input delay value, and the default is 0ns.

<port_list>: Specify input port for the constraints, and it supports the set of get_ports.

Examples

```
# set the input delay based on clk rising edge for port a as 0.8ns
set_input_delay -clock clk 0.8 [get_ports {a}]
# set the input delay based on clk rising edge for all input ports as 0.8ns
set_input_delay -clock clk 0.8 [all_inputs]
# set the input delay based on clk falling edge for port a as 0.8ns
set_input_delay -clock clk -clock_fall 0.8 [get_ports {a}]
# Create Input delays for different min/max and rise/fall combinations
set_input_delay -clock clk -max -rise 1.4 [get_ports {a}]
set_input_delay -clock clk -max -fall 1.5 [get_ports {a}]
set_input_delay -clock clk -min -rise 0.7 [get_ports {a}]
set_input_delay -clock clk -min -fall 0.8 [get_ports {a}]
#Override by -add_delay
set_input_delay -clock clk1 -max 1.5 [get_ports {a}]
set_input_delay -clock clk1 -max 2.5 -add_delay [get_ports {a}]
#Wildcard matches d0, d1, etc.
set_input_delay -clock cck0 -max 1.4 [get_ports {d*}]
```

A.2.2 set_output_delay

Syntax

Command: set_output_delay

Parameter: -clock clock_name

[-clock_fall]

[-rise]

[-fall]

[-max]

[-min]

[-add_delay]

[-source_latency_included]

<delay_value>

<port_list>

-clock: Specify the clock related with output delay.

-clock_fall: Specify that the output delay is related to the falling edge of the clock; if not specified, it is related to the rising edge by default.

-rise/-fall: Specify rise | fall input delay; If only one of them is specified, the specified input delay is applied to the other;

-max/-min: Specify max | min input delay, affecting set and hold respectively; If only one of them is specified, the specified input delay is applied to the other;

-add_delay: When this option is specified, already input constraints will not be overwritten;

-source_latency_included: Specify that the input delay has already included source latency;

<delay_value>: Specify output delay value, and the default is 0ns.

<port_list>: Specify ports for this constraints; and it supports the set of get_ports.

Examples

```
# set the output delay of port b as 0.5ns
set_output_delay -clock clk 0.5 [get_ports {b}]
# set the output delay of all ports as 0.5ns
set_output_delay -clock clk 0.5 [all_outputs]
# set the output delay based on falling edge for all ports as 0.5ns
```

```
set_output_delay -clock clk -clock_fall 0.5 [get_ports {b}]
# set the output delay based on rising edge for all port b
set_output_delay -clock clk -max -rise 0.3 [get_ports {b}]
set_output_delay -clock clk -max -fall 0.5 [get_ports {b}]
set_output_delay -clock clk -min -rise 0.8 [get_ports {b}]
set_output_delay -clock clk -min -fall 0.7 [get_ports {b}]
# Create several input delays related with more than one clocks
set_output_delay -clock clk0 -min 0.5 [get_ports {b}]
set_output_delay -clock clk0 -max 0.6 [get_ports {b}]
set_output_delay -clock clk0 -clock_fall 0.7 -add_delay [get_ports {b}]
set_output_delay -clock clk1 -min 0.8 -add_delay [get_ports {b}]
set_output_delay -clock clk1 -max 0.9 -add_delay [get_ports {b}]
```

A.3 Timing Path Constraints

A.3.1 set_max_delay/ set_min_delay

Syntax

Command: set_max_delay

Parameter: [-from <from list>]

[-to <to list>]

[-through <through_list>]

<delay value>

Command: set_min_delay

Parameter: [-from <from list>]

[-to <to list>]

[-through <through_list>]

<delay value>

-from: Used to specify the start of the path; and it supports the sets of get_clocks, get_ports, get_regs, get_pins.

-to: Used to specify the end of the path; and it supports the sets of get_clocks, get_ports, get_regs, get_pins.

-through: Specify the pins or nets through the path; it supports the sets of get_pins, get_nets, and the pins can only be the ones of non-timing

component. Only one "-through can be used in one statement at one time.

<delay value>: Specify output delay value.

Note!

- The set_max_delay constraint affects the setup clocks and set_min_delay affects the hold clocks.
- The three parameters above can be used together or alone. If the basic objects of the three parameters are not on the same path, Gowin Software will ignore this constraints, and timing analysis will not be affected.

Examples

Set the maximum delay of the timing path from the clock0-driven component to the clock1-driven component to 5ns.

set_max_delay -from [get_clocks {clk0}] -to [get_clocks {clk1}] 5

Wildcard example, e.g. the clock relationship from port d00, d10 to flip flop r0, r1 is 2ns.

set_max_delay -from [get_ports {d*}] -to [get_regs {r?}] 2

Input port to pin constraint affects setup analysis, and pin to output constraint port affects hold analysis.

set_max_delay -from [all_inputs] -to [get_pins {r*/D}] 1.234

set_max_delay -from [get_pins {r?_s0/CLK}] -to [all_outputs] 0.989

Set the maximum delay for all clock-driven timing components to 5ns.

set_max_delay -from [all_clocks] 5 -to [get_ports {out*}]

Set the maximum delay from port a to port b to 2ns.

set_max_delay -from [get_ports {a}] -to [get_ports {b}] 2

Set the maximum delay of the timing component from reg0 to clk falling edge to 2ns.

set_max_delay -from [get_regs {reg0}] -to [get_clocks {clk}] 2

Set the minimum delay of the timing path from the clock-driven component to the clock-driven component to 0.5ns.

set_min_delay -from [get_clocks {clk}] -to [get_clocks {clk}] 0.5

Set the minimum delay from port a to reg0 to 0.5ns.

set_min_delay -from [get_ports {a}] -to [get_regs {reg0}] 0.5

Set the minimum delay from reg0 to port b to 0.5.

set_min_delay -from [get_regs {reg0}] -to [get_ports {b}] 0.5

#Set the minimum delay from port a to port b to 0.5ns.

set_min_delay -from [get_ports {a}] -to [get_ports {b}] 0.5

Set the setup delay relationship from port a to clk and from all data ports to the relevant clocks.

set_max_delay -from [get_ports {a}] -to [get_clocks {clk}] 0.5

set_max_delay -from [all_inputs] -to [all_clocks] 0.111

A.3.2 set_false_path

Syntax

Command: set_false_path

Parameter: [-from <from list>]

[-to <to list>]

[-through <through list>]

[-setup]

[-hold]

-setup/-hold: Specify constraints for setup time or hold time. The two parameters are mutually exclusive. If not specified, it is valid for both setup and hold by default.

-from: Specify the start of the path; it can be collected through the sets of get_ports, get_regs, ger_pins or get_clocks, which can be used separately, and Gowin software automatically gets the end.

-to: Specify the end of the path; it can be collected through the sets of get_ports, get_regs, ger_pins, get_clocks, which can be used separately, and Gowin software automatically gets the start.

-through: Specify the pins or nets through the path, and it supports the sets of get_pins, get_nets. In the parameter list, you can specify multiple pins (PIN) or multiple nets (NET), which can be on the same path or on different paths; multiple "-through" parameters cannot be used in the same constraint.

Note!

If the set of get_pins is used, the value of -from must be a clock pin; the value of -to must be a non-clock pin; the value of -through must be an output pin such as DFF.Q or a rx pin such as DFF.D or DFF.CE on the path.

Examples

#Set_clock_clk0 and clock clk1 excitation path without timing analysis.

set_false_path -from [get_clocks {clk0}] -to [get_clocks {clk1}]

Set the path from reg0 to reg1 without timing analysis.

```
set_false_path -from [get_regs {reg0}] -to [get_regs {reg1}]
# Set the path from the rising edge of clk to the falling edge of clk1 without timing analysis.
set_false_path -from [get_clocks {clk}] -to [get_clocks {clk1}]
# Specify the path from port a to port b without timing analysis.
set_false_path -from [get_ports {a}] to [get_ports {b}]
#Use -from alone, and it is valid for setup and hold.
set_false_path -from [get_pins {reg0_s0/CLK}]
set_false_path -from [get_regs {reg0_s0}]
set_false_path -from [get_clocks {cck}]
# Use -to alone, and it is valid for setup.
set_false_path -from [get_regs {reg0_s0}] -setup
# Use -to alone, and it is valid for hold.
set_false_path -from [get_regs {reg0_s0}] -hold
#use -through alone, the timing path through reg0_s0.Q is not analyzed
set_false_path - through [get_pins {reg0_s0/Q}]
# Use -through alone, the timing path through reg0_c is not analyzed.
set_false_path - through [get_nets {reg0_c}]
#"*" matches multiple characters such as mi/reg0.
set_false_path -from [get_regs {mi/r*0}] -to [get_regs {spi/R*}]
#"?" matches a character such as reg0, reg1.
set_false_path -from [get_pins {mi/r?g0/CLK}] -to [get_pins {spi/DI}]
```

A.3.3 set_multicycle_path

Syntax

Command: set_multicycle_path

Parameter: [-setup|-hold]

[-start|-end]

[-from <from_list>]

[-to <to list>]

[-through <through_list>]

<path multiplier>

-start/-end: Specify whether the constraint reference clock is a launch

clock or a latch clock; the reference clock specified by the parameter "-start" is a launch clock; the reference clock specified by the parameter "-end" is a latch clock. The default is a latch clock.

-setup/-hold: Specify whether the current constraint affects setup check or hold check; these two parameters are mutually exclusive. The default is to affect the setup check.

-from: The start of the path, and it can collect the start by sets of get_pins, get_ports, get_regs, get_clocks.

-to: The end of the path, and it can collect the end by sets of get_pins, get_ports, get_regs, get_clocks.

-through: Specify the pins or nets through the path, and it supports the sets of get_pins, get_nets. In the parameter list, you can specify multiple pins (PIN) or multiple nets (NET), which can be on the same path or on different paths; multiple "-through" parameters cannot be used in the same constraint.

<path multiplier>: Specify the number of the cycles.

Note!

"-from", "-to", and "-through" can be used together or alone. If the specified objects of the three parameters are not on the same path, Gowin Software will ignore this constraints, and timing analysis will not be affected.

Examples

```
create_clock -name clk -period 10 [get_ports {clk}]
create_generated_clock -name genClk -multiply_by 2 -source [get_ports {clk}] [get_pins {pll_out}]
# Set multi-cycle path: reference clock is genClk, which has an impact on the setup check.
set_multicycle_path -end -setup -from [get_clocks {clk}] -to [get_clocks {genClk}] 2
#Set multi-cycle path: the reference clock is the clock of the reg0, which has an effect on the setup and hold checks.
set_multicycle_path -start -setup -from [get_regs {reg0}] -to [get_regs {reg1}] 3
set_multicycle_path -start -hold -from [get_regs {reg0}] -to [get_regs {reg1}] 1
# Set multi-cycle path: the reference clock is clk0, which only affects the path where the source clock is the rising edge of clk to the falling edge of clk0.
```

set_multicycle_path -end -setup -from [get_clocks {clk}] -to [get_clocks {clk0}] 3

Wildcard "?" and "*" example; "?" can match addr0, addra, etc. "*" can match Data_s0, D0_s0, etc.

set_multicycle_path -from [get_regs {SD/addr? }] -to [get_regs {RSG/D*_s0}

A.4 Operating Conditions Constraints

Syntax

Command: set_operation_conditions

Parameter: [-grade < c|i|a >]

[-model <slow|fast>]

[-speed <speed>]

[-setup]

[-hold]

[-max]

[-min]

[-max_min]

- -grade: Specify the device temperature grade, supporting commercial, industrial and automotive grade.
- -model: Specify the timing analysis model.
- -speed: Specify the device speed grade.
- -setup: Setup time check under current process corner; and it has the same function as –max.
- -hold: Hold time check under current process corner; and it has the same function as –min.
- -max: Setup time check under current process corner; and it has the same function as –setup.
- -min: Hold time check under current process corner; and it has the same function as –hold.
- -max_min: Setup time and hold time check under current process corner; it has the same function as -setup and -hold.

Examples

Industry speed level 6 and fast model have an effect on setup and

hold analysis.

set_operating_conditions -grade i -model fast -speed 6 -setup –hold

Industry speed level 7 and slow model have an effect on setup and hold analysis.

set_operating_conditions -grade c -model slow -speed 7 -max_min

A.5 Timing Report Constraints

A.5.1 report_timing

Syntax

Command:: report_timing

Parameter:[-setup|-hold|-recovery|-removal]

[-max_paths <value>]

[-max_common_paths < value >]

[-rise_from <rise_from_list>]

[-fall_from <fall_from_list>]

[-to <to list>]

[-rise_to <rise_to_list>]

[-fall_to <fall_to_list>]

[-through <through list>]

[-from_clock<from clok>]

[-fall_from_clock <from clok>]

[-rise_from_clock <from clok>]

[-to_clock <to clok>]

[-rise_to_clock <to clok>]

[-fall_to_clock <to clok>]

[-min_logic_level]

[-max_logic_level]

[-mod_ins {mod_ins1 mod_ins2 ...}]

-setup|-hold|-recovery|-removal: Specify the type of timing check, exclusive.

-max_paths: Specify the maximum number of paths of timing report.

-max_common_paths: Specify the maximum number of paths sharing the common end of timing report.

-from/-rise_from/-fall_from: Specify the start of the timing report paths;

-rise/fall_from needs to be clocks and supports the set of get_clocks; when used alone, Gowin Software automatically gets the start.

-to /-rise_to /-fall_to: Specify the end of the timing report path; -rise/fall_to needs to be a clock and supports the set of get_clocks; when used alone, Gowin Software automatically gets the end.

-through: Specify the pins or nets through the path, and it suppotst the set get_nets, get_pins.

-from_clock /-fall_from_clock /-rise_from_clock: Specify the clock associated with the start of the timing report path and it supports the set of get_clocks; when used alone, Gowin Software automatically gets the end.

-to_ clock /-rise_to_ clock /-fall_to_clock: Specify the clock associated with the end of the timing report path and it supports the set of get_clocks; when used alone, Gowin Software automatically gets the start.

-min_logic_level/-max_logic_level: Limit the logic level of the reporting paths.

-mod_ins {mod_ins1 mod_ins2 ...}: Specify multiple instantiated module instances, separated by spaces; if this parameter is not added, the timings in the whole design are reported by default.

Examples

Specify to report setup time checks with 100 paths.

report_timing -setup -max_paths 100 -max_common_paths 5

#The start launch of the report is ck; the end uses wildcards to match r0, r1, etc.

report_timing -hold -rise_from [get_clocks {ck}] -to [get_pins {r*/D }]

report_timing -setup -fall_from [get_clocks {ck }] -to [get_regs {r*}]

Specify the number of logical levels on the path as 2; and report a maximum of 2 paths and a maximum of 1 path with common end path.

report_timing -recovery -from_clock [get_clocks {cck0}] -to_clock [get_clocks {cck1}] -max_paths 2 -max_common_paths 1 -max_logic_level 2 -min_logic_level 2

#Hold analysis reports only the timing of the instantiated uut module.

report_timing -hold -mod_ins {uut}

A.5.2 report_high_fanout_nets

Syntax

Command: report_high_fanout_nets

Parameter: [-clock_regions]

[-slr]

[-ascending]

[-max_nets <max_net_value>]

[-min_fanout <min_fanout_value>]

[-max_fanout <max_fanout_value>]

-clock_regions: Optional; when specified, report the clock net connecting to the clock input of timing component only.

-slr: Optional; when specified, report the clock net connecting to the reset/set port (synchronous or asynchronous) of timing component only.

-ascending: Optional; the report nets fanout is arranged in descending order; if this parameter is not specified, the report nets fanout is arranged in ascending order by default.

-max_net: Optional; this parameter specifies the maximum number of nets to report. When this parameter is not specified, the default maximum number of nets reported is 10.

-min_fanout: Optional; this parameter specifies only the reported net fanout whose number is not less than this parameter value.

-max_fanout: Optional; this parameter specifies only the reported net fanout whose number is not greater than this parameter value.

Examples

#NET of the reset/set input of the timing component, range of [1,15]; up to 10 nets are reported.

report_high_fanout_Nets -slr -max_nets 10 -min_fanout 1 -max_fanout 15

#Of all NETs, report NET fanouts, up to 10.

report_high_fanout_Nets -max_nets 10

A.5.3 report_route_congestion

Syntax

Command: report_route_congestion

Parameter: [-max_grids <max grids value>]

[-min_route_congestion <min route congestion value>]

[-max_route_congestion <max route congestion>]

[-LOC <position>]

-max_grids: Optional; specify the maximum number of grids to report;

if it is not specified, it reports the congestion of ten grids by default.

-min_route_congestion: Optional; specify the minimum route congestion of grid to report; if it is not specified, the default value is 0.

-max_route_congestion: Optional; specify the maximum route congestion of grid to report; if it is not specified, the default value is 1. Its value should not be less than parameter value min_route_congestion; or the warning is reported, this statement is ignored.

-LOC: Optional; specify the physical location of grids to report. It can be be a single location, such as R1C3, which means the grid of first row and the third column; it also can be a location range, such as R1C[1:3], which means the grid of column 1~3, row 3; R[1:3]C1, which means the grid of column 1, row 1~3; or R[1:3]C[1:3], which means the grid of column 1~3, row 1~3.

Examples

Report the route congestion of grids locating on row 1 to 5, column 1 to 5 whose route congestion is between 0 and 0.5; and only the five with the highest congestion are reported.

report_route_congestion -max_grids 5 -min_route_congestion 0 -max_route_congestion 0.5 -LOC R[1:5]C[1:5]

A.5.4 report_min_pulse_width

Syntax

Command: report_min_pulse_width

Parameter: [-nworst <nworst value>]

[-min_pulse_width <min pulse width value>]

[-max_pulse_width <max pulse width value>]

[-detail]

[get_regs {regIns name}]

- -nworst: Specify the number of the worst paths to report.
- -min_pulse_width: Specify the minimum pulse width of timing component to report.
- -max_pulse_width: Specify the maximum pulse width of timing component to report.
- -detail: The report will be detailed if this parameter is specified, otherwise the report will be brief.
- get_regs {regIns name}: Specify reg, one or more regs can be specified.

All flip flops pulse width timing analysis will be reported by default.

Examples

#Report the worst 3 clock paths that have pulse width between 0.1 and 4 in detail.

report_min_pulse_width -nworst 3 -min_pulse_width 0.1 -max_pulse_width 4 –detail

#Report the worst 20 clock paths that have pulse width between 0.001 and 4 in brief:

report_min_pulse_width -nworst 20 -min_pulse_width 0.001 -max_pulse_width 4

A.5.5 report_max_frequency

Syntax

Command: report_max_frequency

Parameter: -mod_ins {mod_ins1 mod_ins2 ...}

-mod_ins {mod_ins1 mod_ins2 ...}: specify multiple module instances, separated by a space; The whole design maxminum frquency will be reported by default no matter this parameter is specified or not.

Examples

Report the max. frequency of bsram0.

report_max_frequency -mod_ins {bsram0}

A.5.6 report_exceptions

Syntax

Command: report_exceptions

Parameter: -setup|-hold | -recovery | removal

[-max_paths<number>]

[-max_common_paths< number >]

[-max_logic_level <number>]

[-min_logic_level <number>]

[-rise_from <rise_from_list>]

[-fall_from <fall_from_list>]

[-to <to list>]

[-rise_to <rise_to_list>]

[-fall_to <fall_to_list>]

[-through <through list>]

[-rise_through <rise_through_list>]

[-fall_through <fall_through_list>]

[-from_clock<from clock>]

[-fall_from_clock<from clock>]

[-rise_from_clock<from clock>]

[-to_clock<to clock>]

[-rise_to_clock<to clock>]

[-fall_to_clock<to clock>]

The name, meaning, and use of its keywords are the same as those of report_timing; and it reports the path generated by the exception constraint.

Example

Display a path reported by recovery.

create_clock -name mm -period 10 -waveform {0 5} [get_ports {clk}]

set_max_delay -from [get_clocks {mm}] -to [get_clocks {mm}] 0.22

report_exceptions -recovery -from_clock [get_clocks {mm}] -to_clock [get_clocks {mm}] -max_paths 1 -max_common_paths 1


GOWIN
PROGRAMMING FOR THE FUTURE