Gowin Design Timing Constraints

User Guide

SUG940-2.0.3E, 03/09/2026

Copyright © 2026 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN and LittleBee are trademarks of Guangdong Gowin Semiconductor Corporation and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

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. 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 descriptions of wildcards added. ● Some figures updated.
11/02/2021	1.3E	● Software version updated. ● Some figures and corresponding descriptions updated. ● Chapter 5 Syntax Specification updated.
05/20/2022	1.3.1E	Some language descriptions updated.
07/28/2022	1.4E	● -Add option added in Create Clock. ● The description of temperature and voltage of delay data model updated. ● The descriptions of tUnc and tSu added. ● The function of wildcard character cross-hierarchy matching added.
12/13/2022	1.4.1E	The description of Add delay in 3.7.2 I/O Constraints updated.
03/31/2023	1.4.2E	● The descriptions of set_input_delay and set_output_delay in 3.7.2 I/O Constraints updated. ● The description of 4.1.4 Total Negative Slack Summary updated.
04/20/2023	1.4.3E	Figure 3-8 New Constraint, Figure 3-9 Open Timing Constraint, and the related descriptions added.
05/25/2023	1.5E	● The descriptions of exception constraints updated. ● The descriptions of the virtual clock DEFAULT_CLK updated.
08/18/2023	1.5.1E	● set_operation_conditions modified to set_operating_conditions. ● The constraint descriptions of reporting max. clock frequency updated.
11/30/2023	1.6E	● Descriptions of conversion between clock period and frequency removed. ● Process window screenshot and related description removed. ● Descriptions of set_output_delay updated.
02/02/2024	1.7E	derive_clocks constraint added.
06/28/2024	1.8E	Use case for DCS in -add option added in create_clock.
08/09/2024	1.8.1E	Figure 4-1 Static Timing Analysis Report and Figure 4-2 Timing Summaries updated.
10/25/2024	1.8.2E	Descriptions of the default conventions for Clock Name in section 4.1.2 Clock Summary enhanced.
12/31/2024	1.8.3E	● Figure 3-5 Timing Constraint Editor Interface updated.

Date	Version	Description
		● Examples in Chapter 5 Syntax Specification updated.
04/30/2025	1.9E	● Descriptions of Chapter 5 Syntax Specification updated. ● Some screenshots and their corresponding descriptions updated.
06/27/2025	2.0E	● Descriptions for rules of creating differential BUF clock added. ● Formula for minimum pulse width calculation added.
2025/08/29	2.0.1E	The NODE information print in the tUnc row of the STA report removed, and the related figures updated: Figure 4-10 Hold Analysis ReportFigure 4-11 Recovery Analysis Report, Figure 4-12 Removal Analysis Report, Figure 4-18 Timing Exceptions Report, Figure 4-20 report_exception Report.
12/31/2025	2.0.2E	● Updated group constraint examples: Example 5.16 and Example 5.17. ● Added rules for specifying the Master Clock when creating generated clocks. ● Added software analysis rule “-add_delay” in 5.3 I/O Constraints. ● Updated Figure 3-27 Create Operating Conditions Constraints. ● Added voltage and temperature specifications for Slow and Fast models in different device series in 3.7.4 Operating Conditions Constraints. ● Added priority rules for create_clock, derive_clocks, and "Configuration > Global > Frequency (MHz)".
03/09/2026	2.0.3E	● Updated the default timing models used for Setup and Hold analysis in section 4.1.1 STA Tool Run Summary. ● Added the descriptions of "active clock edge time" and the printing of special paths such as PLL-driven DCE to section 4.2.3 Timing Report By Analysis Type.

Contents

Contents i
List of Figures iv
List of Tables vi
1 About This Guide 1
1.1 Purpose 1
1.2 Related Documents 1
1.3 Terminology and Abbreviations 1
1.4 Support and Feedback 2
2 Introduction 3
2.1 Constraint Support 3
2.2 Report Printing 3
2.3 Timing Paths 4
3 Timing Constraints Editor 5
3.1 Overview 5
3.2 Start Timing Constraints Editor 5
3.3 Create/Open/Add Constraints File 5
3.3.1 Create Constraints File 5
3.3.2 Open Constraints File 7
3.3.3 Add Constraints File 8
3.4 Timing Constraints Editor Interface 8
3.5 Open Timing Constraints Window 11
3.6 Edit SDC File 12
3.7 Create Timing Constraints 12
3.7.1 Clock Constraints 13
3.7.2 I/O Constraints 21
3.7.3 Exceptions Constraints 23
3.7.4 Operating Conditions Constraints 25
3.7.5 Reports Constraints 26
3.7.6 Other Constraints 35

3.7.7 Save 37
3.8 Priority of Timing Constraints 37
4 Timing Report 38
4.1 Timing Summaries 38
4.1.1 STA Tool Run Summary 39
4.1.2 Clock Summary 40
4.1.3 Max Frequency Summary 41
4.1.4 Total Negative Slack Summary 41
4.2 Timing Details 41
4.2.1 Path Slacks Table 41
4.2.2 Minimum Pulse Width Table 43
4.2.3 Timing Report By Analysis Type 44
4.2.4 Minimum Pulse Width Report 49
4.2.5 High Fanout Nets Report 50
4.2.6 Route Congestions Report 50
4.2.7 Timing Exceptions Report 51
4.2.8 Timing Constraints Report 54
5 Syntax Specification 55
5.1 Object Collections 55
5.2 Clock Constraints 55
5.2.1 create_clock 55
5.2.2 create_generated_clock 57
5.2.3 set_clock_latency 60
5.2.4 set_clock_uncertainty 62
5.2.5 set_clock_groups 64
5.3 I/O Constraints 65
5.4 Exception Constraints 69
5.4.1 set_false_path 69
5.4.2 set_max_delay/set_min_delay 70
5.4.3 set_multicycle_path 71
5.5 Report Constraints 73
5.5.1 report_timing 73
5.5.2 report_high_fanout_nets 76
5.5.3 report_route_congestion 77
5.5.4 report_min_pulse_width 78
5.5.5 report_max_frequency 79
5.5.6 report_exceptions 80

5.6 Operating Conditions Constraints 82
5.7 Other Constraints 83
5.7.1 derive_clocks 83

List of Figures

Figure 3-1 New Dialog Box 6
Figure 3-2 New Timing Constraints File 6
Figure 3-3 Open Timing Constraints File 7
Figure 3-4 Add Constraints File 8
Figure 3-5 Timing Constraint Editor Interface 9
Figure 3-6 Netlist Tree View 9
Figure 3-7 Constraints Editing Window 10
Figure 3-8 New Constraint 10
Figure 3-9 Open Timing Constraint 11
Figure 3-10 Open Timing Constraints Window via Menu Bar 11
Figure 3-11 Right-click to Open Timing Constraints Window 12
Figure 3-12 Edit SDC File 12
Figure 3-13 Create Clock 13
Figure 3-14 Select Objects 14
Figure 3-15 Add Clock 15
Figure 3-16 Clock List 15
Figure 3-17 Right-click Menu for the Clock 15
Figure 3-18 Create Generated Clock Constraints 17
Figure 3-19 Select Create Generated Clock 18
Figure 3-20 Set Clock Latency 19
Figure 3-21 Set Clock Uncertainty 20
Figure 3-22 Set Clock Groups 21
Figure 3-23 Create I/O Delay Constraints 22
Figure 3-24 Create False Path Constraints 23
Figure 3-25 Create Max/Min Delay Constraints 24
Figure 3-26 Create Multicycle Path Constraints 25
Figure 3-27 Create Operating Conditions Constraints 26
Figure 3-28 Create Report Timing 27
Figure 3-29 Report Timing Dialog Box 28
Figure 3-30 Create Report High Fanout Nets 29

Figure 3-31 Report High Fanout Nets Dialog Box 29
Figure 3-32 Create Report Route Congestion 30
Figure 3-33 Report Route Congestion Dialog Box 31
Figure 3-34 Create Report Min Pulse Width 32
Figure 3-35 Report Min Pulse Width Dialog Box 32
Figure 3-36 Create Report Max Frequency 33
Figure 3-37 Report Max Frequency Dialog Box 33
Figure 3-38 Create Report Exception 34
Figure 3-39 Report Exception Dialog Box 35
Figure 3-40 Create Derive Clocks 35
Figure 3-41 Select Create Derive Clocks 36
Figure 3-42 Derive Clocks List 36
Figure 4-1 Static Timing Analysis Report 38
Figure 4-2 Timing Summaries 39
Figure 4-3 Path & Endpoints 40
Figure 4-4 Path Slacks Table 43
Figure 4-5 Minimum Pulse Width Table 44
Figure 4-6 Path Summary 45
Figure 4-7 Data Arrival Path 45
Figure 4-8 Path Required by Data 46
Figure 4-9 Path Statistics 46
Figure 4-10 Hold Analysis Report 47
Figure 4-11 Recovery Analysis Report 48
Figure 4-12 Removal Analysis Report 49
Figure 4-13 Minimum Pulse Width 50
Figure 4-14 High Fanout Nets Report 50
Figure 4-15 Route Congestions Report 51
Figure 4-16 Test Case 51
Figure 4-17 Timing Exceptions Constraints 51
Figure 4-18 Timing Exceptions Report 52
Figure 4-19 report_exception Statement 53
Figure 4-20 report_exception Report 53
Figure 4-21 Timing Constraints Report 54

List of Tables

Table 1-1 Terminology and Abbreviations 1
Table 5-1 Collection Description 55
Table 5-2 create_clock Parameters 56
Table 5-3 create_generated_clock Parameters 57
Table 5-4 set_clock_latency Parameters 60
Table 5-5 set_clock_uncertainty Parameters 62
Table 5-6 set_clock_groups Parameters 64
Table 5-7 Analysis Results 65
Table 5-8 Analysis Results 65
Table 5-9 I/O Constraints Parameters 66
Table 5-10 set_false_path Parameters 69
Table 5-11 set_max_delay/set_min_delay Parameters 70
Table 5-12 set_multicycle_path Parameters 72
Table 5-13 report_timing Parameters 74
Table 5-14 report_high_fanout_nets Parameters 76
Table 5-15 report_route_congestion Parameters 77
Table 5-16 report_min_pulse_width Parameters 78
Table 5-17 report_max_frequency Parameters 80
Table 5-18 report_exceptions Parameters 81
Table 5-19 Operating Conditions Constraints Parameters 82
Table 5-20 derive_clocks Parameter 83

1 About This Guide

1.1 Purpose

This manual describes Gowin Timing Constraint Editor, timing reports, and syntax specifications.

1.2 Related Documents

The latest user guides are available on GOWINSEMI Website: www.gowinsemi.com.You can find the related document 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
FF	Flip-Flop
GUI	Graphical User Interface
I/O	Input and Output
LSR	Clear, Set, Reset, and Preset
NET	Net
OSC	Oscillator
PIN	Input and Output Pins of Instantiated Module
PORT	Input and Output Ports of Top-level Module
PnR	Place and Route
SDC	Synopsys Design Constraint

Terminology and Abbreviations	Meaning
STA	Static Timing Analysis

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

Gowin Static Timing Analysis (STA) performs a comprehensive analysis on timing paths within the circuit netlists. It calculates the latency of these timing paths and determines whether they meet the specified requirements. Designers only need to provide constraint stimuli, and the analysis and calculation process is automatically completed by Gowin Software. It delivers detailed timing reports, with significant advantages in time verification and coverage.

2.1 Constraint Support

- Supports clock constraints
- Supports I/O latency constraints
- Supports exception constraints
- Supports report constraints
- Supports operating condition constraints
- Supports derive clock constraints
- Supports editing constraints through Timing Constraint Editor GUI;
- Timing constraint syntax conforms to the Synopsys Design Constraint (SDC) specification.

2.2 Report Printing

- Supports printing of timing latency models
- Supports printing of maximum clock frequency
- Supports printing of setup and hold path analysis
- Supports printing of recovery and removal path analysis
- Supports printing of minimum pulse width analysis
- Supports printing of high fanout nets
- Supports printing of routing congestion
- Supports printing of exception constraint reports

- Supports printing in both HTML and TXT file formats

2.3 Timing Paths

Static Timing Analysis analyzes four types of timing paths, which are classified based on their starting and ending points:

- Paths from input ports to internal timing components.
- Paths from timing components to timing components.
- Paths from internal timing components to output ports.
- Paths from input ports to output ports.

3 Timing Constraints Editor

3.1 Overview

Users can add, delete, modify, and check timing constraints through the Timing Constraints Editor.

3.2 Start Timing Constraints Editor

Timing Constraints Editor can be launched independently or used after a successful synthesis.

To launch it independently, click "Tools > Timing Constraints Editor". To launch it within a project, first run "Synthesize" successfully in Gowin Software "Process" window, then double-click "Process > Timing Constraints Editor". The netlist file and the timing constraint file from the project will be automatically loaded into the editor. If no timing constraint file exists in the project, one will be created automatically.

3.3 Create/Open/Add Constraints File

3.3.1 Create Constraints File

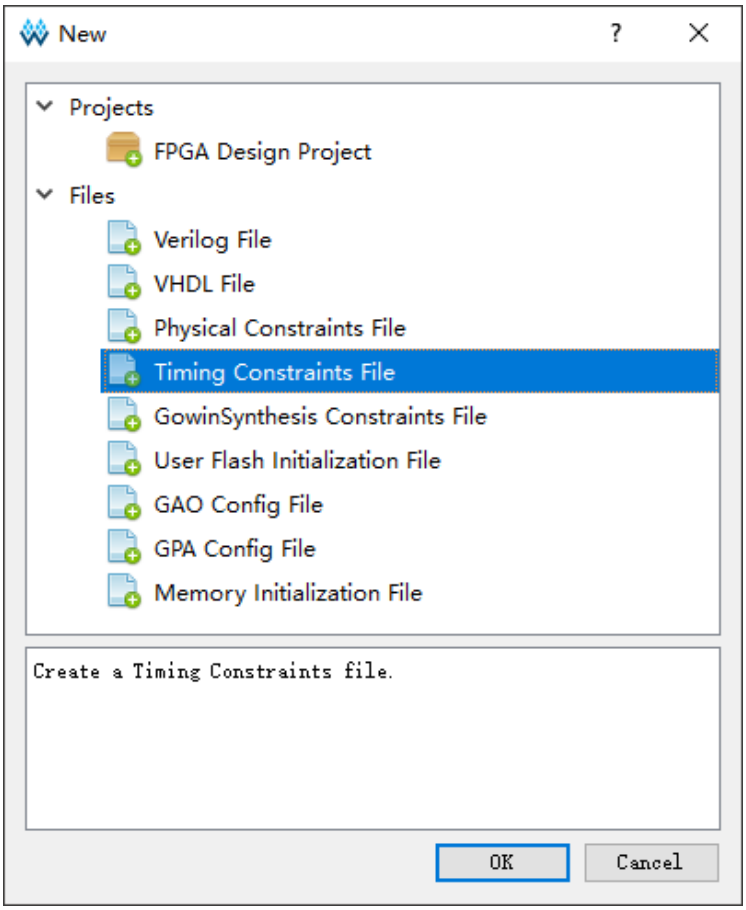
The steps of creating constraint files are as follows.

1. Click "File > New" and "New" dialog box pops up.
2. Select "Timing Constraints File" option, as shown in Figure 3-1.

You can also create constraints file in the following methods.

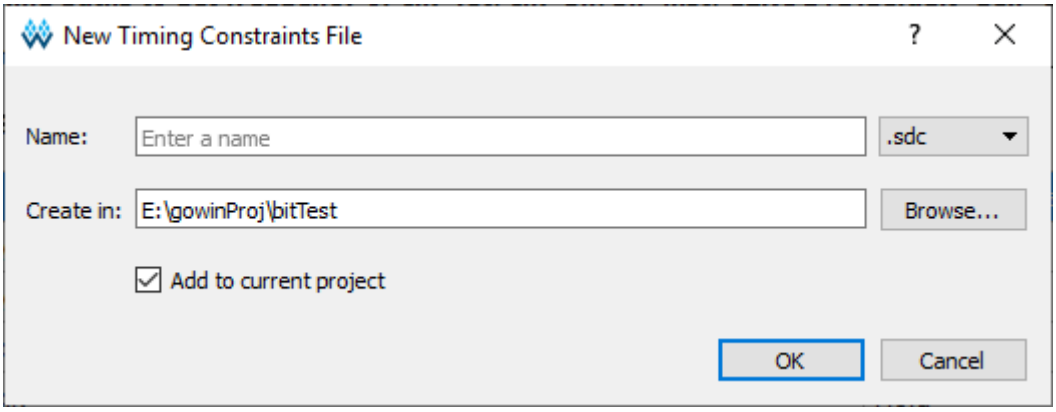
- Click the "New" icon in the toolbar.
- Use shortcut "Ctrl+N".

Figure 3-1 New Dialog Box



3. Click "OK" and "New Timing Constraints File" pops up, as shown in Figure 3-2.

Figure 3-2 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: Used to select the file storage path by clicking "Browse" and the directory must already exist. The default path is the 'src' folder under the project directory.
- Add to current project: Used to add the constraints file to the project automatically when this option is checked.

3.3.2 Open Constraints File

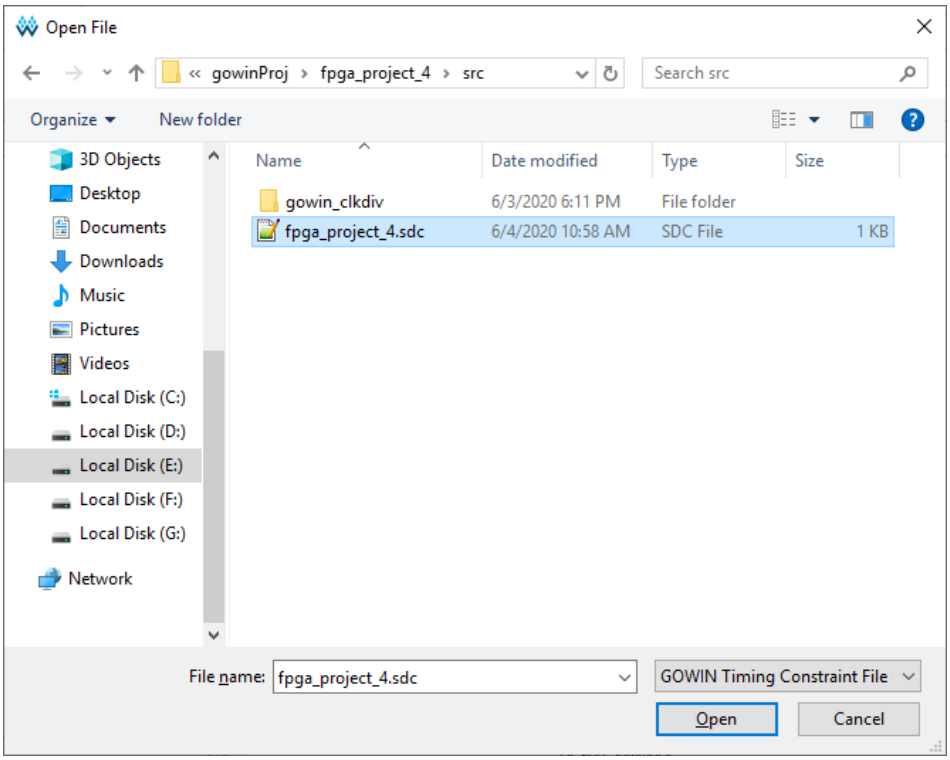
The steps are as follows.

1. On IDE interface, click "File > Open".
2. "Open File" dialog box pops up, as shown in Figure 3-3.

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

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

Figure 3-3 Open Timing Constraints File



3. Select the directory where the timing constraint file is located. After selecting the file, click "Open" to open it. SDC file types are supported.

3.3.3 Add Constraints File

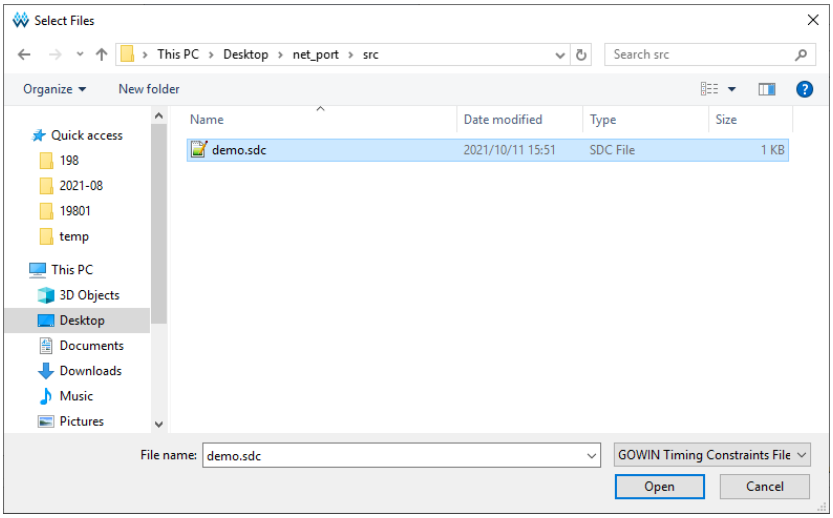
The steps are as follows.

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 3-4.
3. Click "Open" to add one or more selected files to the project.

Note!

When multiple files are added, the last one added takes effect by default.

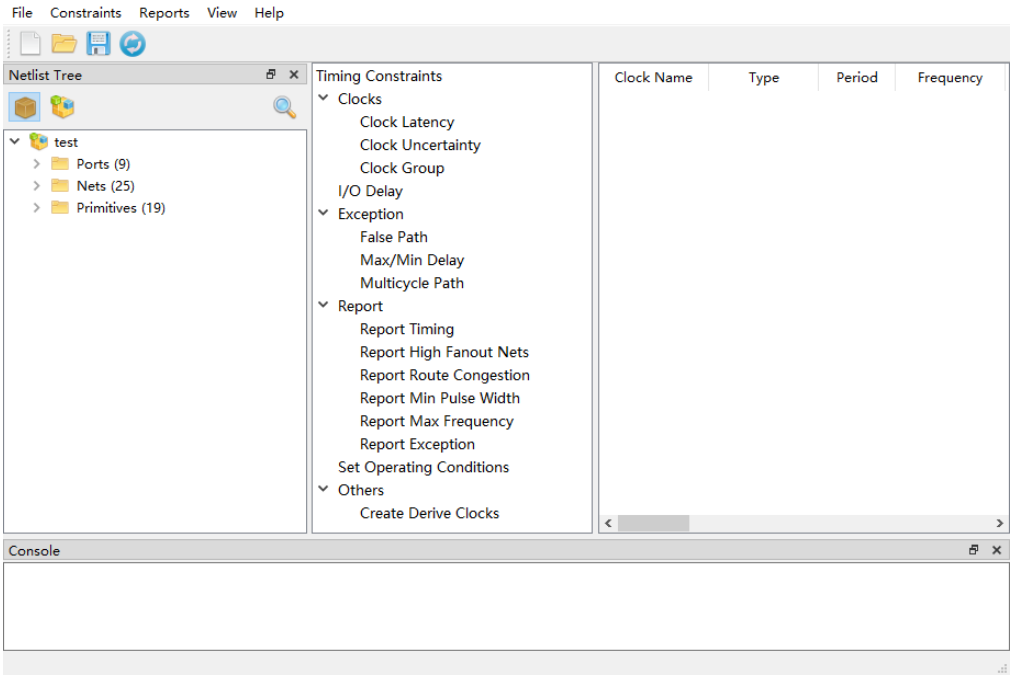
Figure 3-4 Add Constraints File



3.4 Timing Constraints Editor Interface

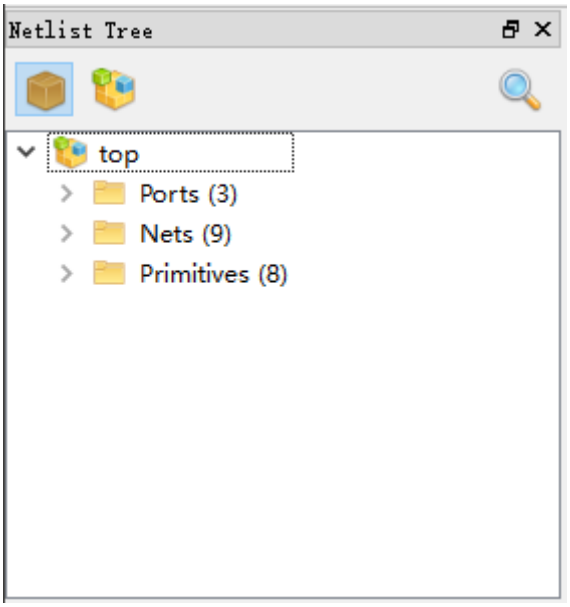
The editor interface is shown in Figure 3-5.

Figure 3-5 Timing Constraint Editor Interface



The "Netlist Tree" view is as shown in Figure 3-6.

Figure 3-6 Netlist Tree View



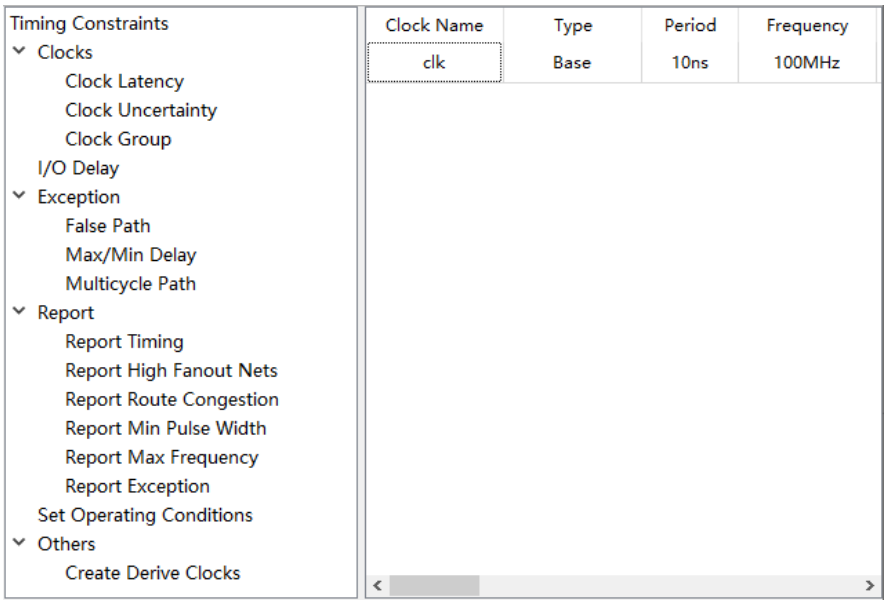
The Netlist Tree includes Top Module, I/O Ports, Nets, and Primitives.

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

The constraints editing area is as shown in Figure 3-7. The left is the

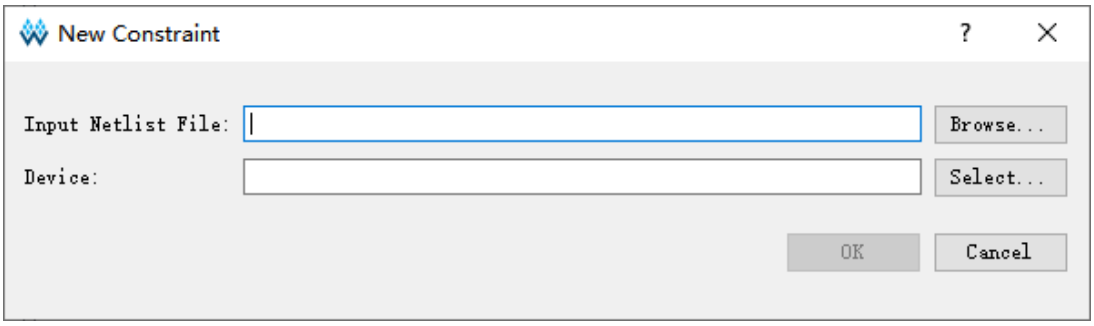
timing constraints type and the right is the table editing area. Click a constraints type, and the constraints editing list will be displayed in editing area.

Figure 3-7 Constraints Editing Window



The toolbar includes New " ", Open " ", Save " ", and Reload " ". In the "New Constraint" dialog box, you can configure "Input Netlist File" and "Device", as shown in Figure 3-8.

Figure 3-8 New Constraint



Click " " to open "Open Timing Constraint" dialog box, you can configure "Input Netlist File", "Constraint File", and "Device", as shown in Figure 3-9.

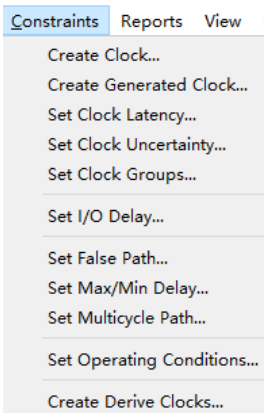
Figure 3-9 Open Timing Constraint

3.5 Open Timing Constraints Window

There are two methods to open timing constraints window.

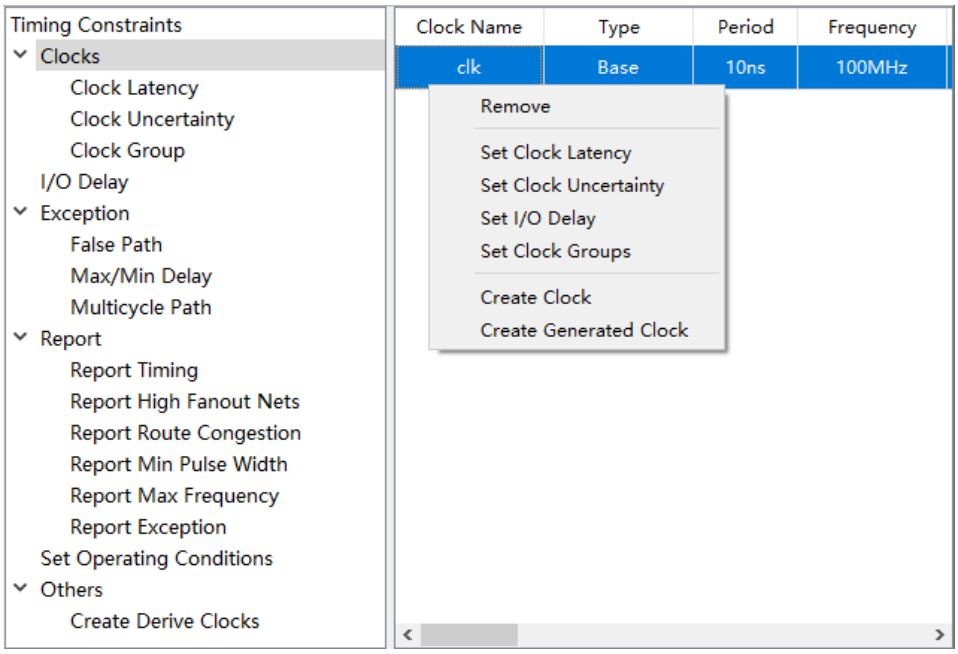
1. In the menu bar, click "Constraints", then select a timing constraint command from the drop-down menu. The corresponding timing constraint window will open based on the selected command, as shown in Figure 3-10.

Figure 3-10 Open Timing Constraints Window via Menu Bar



2. On the right side of the Timing Constraints Editor window, right-click and select different timing constraint commands from the menu. The corresponding constraint window will open based on the selected command, as shown in Figure 3-11.

Figure 3-11 Right-click to Open Timing Constraints Window



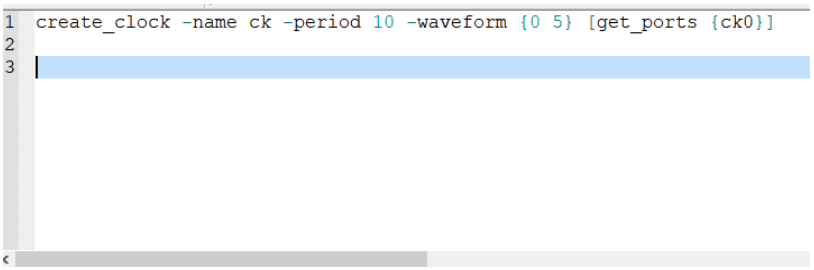
3.6 Edit SDC File

Gowin Software supports reading SDC file and you can manually modify the SDC file in the timing constraints editor, as shown in Figure 3-12.

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 3-12 Edit SDC File



3.7 Create Timing Constraints

This section introduces how to create timing constraints using the Timing Constraints Editor. The created constraints will be printed in the Console window, and can be saved to an SDC file by clicking "Save".

3.7.1 Clock Constraints

Create Clock

Create a clock.

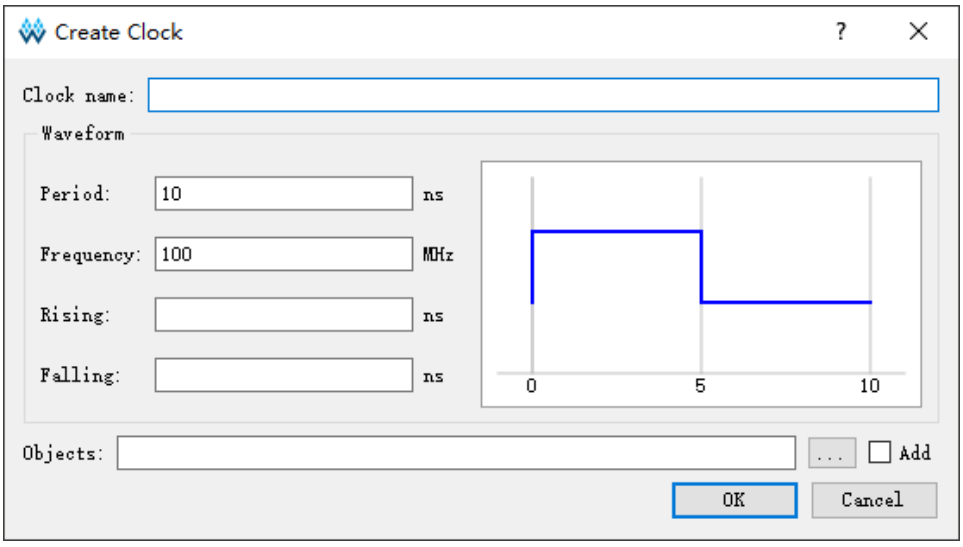
You can configure parameters, such as name, period, frequency, rising edge, falling edge, etc. Gowin Software can create multiple clocks which form multiple clock domains, and support clock domain crossing analysis.

You can add clock constraints in the following two methods.

1. Add Clock constraint via Constraints option.

   a) Select "Constraints > Create Clock... ", and the "Create Clock" dialog box pops up, as shown in Figure 3-13.

Figure 3-13 Create Clock



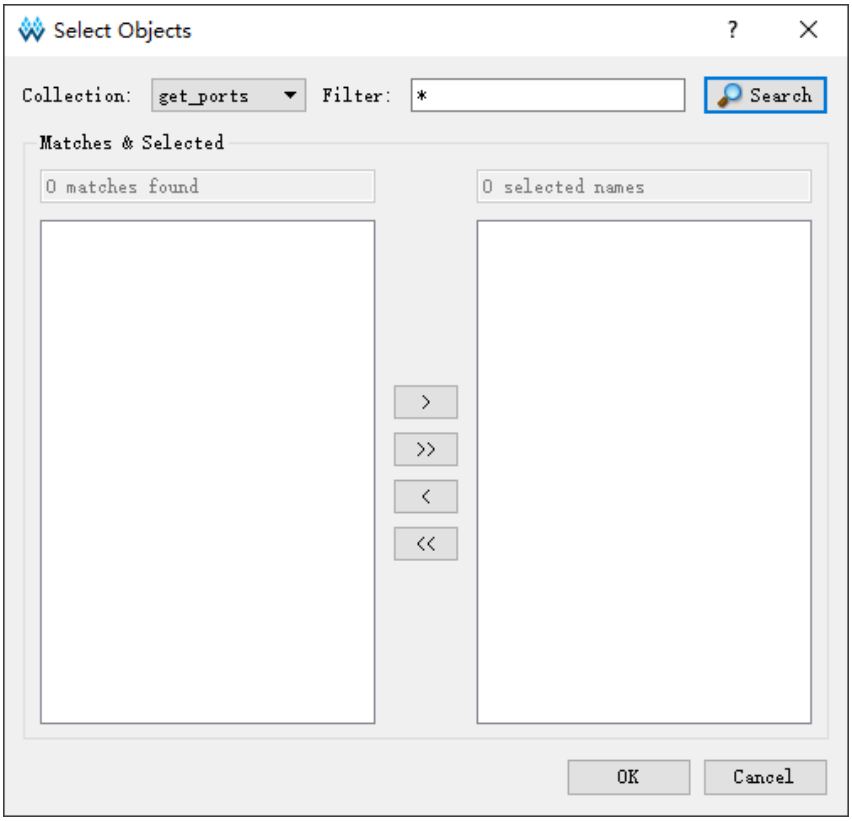
   b) Configure the clock

   - Clock Name: Supports identifiers beginning with letters or underscores.
   - Period: Floating point greater than 0, accurate to three decimal places, unit in ns, and 10 by default.
   - Frequency: Floating point greater than 0, accurate to three decimal places, unit in MHz, and 100 by default.
   - Rising: Floating point greater than or equal to 0, accurate to three decimal places, unit in ns.
   - Falling: Floating point greater than 0, accurate to three decimal places, unit in ns.
   - Objects: Specifies the target; click "...." to select the object.
   - Add: Checks it when adding multiple clocks to the same source.

   c) Click "...." on the right side of "Objects" and "Select Objects" pops

up, as shown in Figure 3-14.

Figure 3-14 Select Objects



d) As shown in Figure 3-14, "Collection" is used to specify 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. ">" is used to add the selected items from the left list to the right list. "> >" is used to add all the selected items from the left list to the right list. "<" is used to remove the selected item in the right list. "< <" is used to remove all the selected items in the right list.

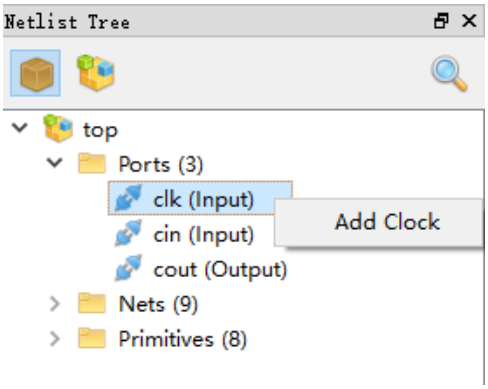
e) Click "OK" to add Objects.

2. Add clock constraints via "Netlist Tree".

a) Select Port or Net in Netlist Tree.

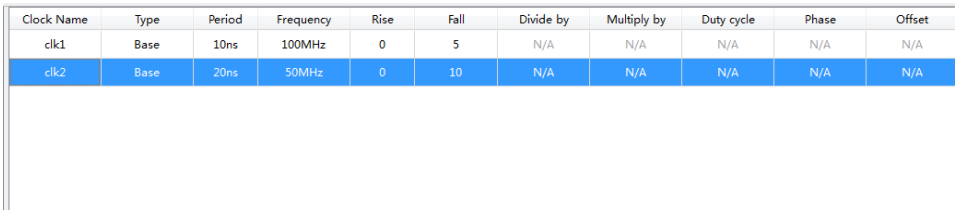
b) Right-click and select "Add Clock" to add a clock, as shown in Figure 3-15.

Figure 3-15 Add Clock



After finishing, the constraints will be added in clock list, as shown in Figure 3-16.

Figure 3-16 Clock List

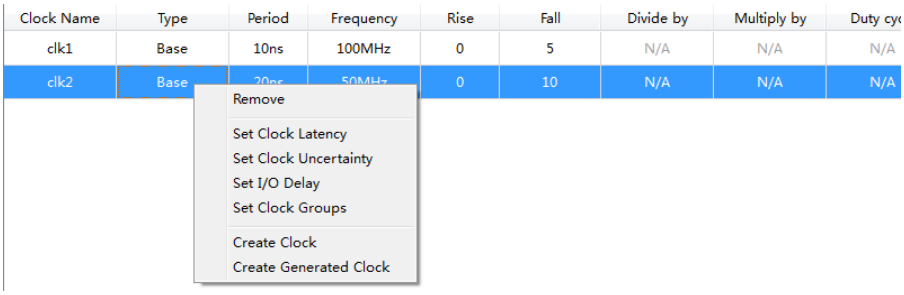


Clock Name	Type	Period	Frequency	Rise	Fall	Divide by	Multiply by	Duty cycle	Phase	Offset
clk1	Base	10ns	100MHz	0	5	N/A	N/A	N/A	N/A	N/A
clk2	Base	20ns	50MHz	0	10	N/A	N/A	N/A	N/A	N/A

In this list, the following operations can be performed:

- Edit Clock: Double-click the corresponding constraint in the "Clocks" list to open the Clock editing dialog box, where you can modify the Clock information.
- Delete Clock: Select the Clock entry in the list, right-click, and choose "Remove".
- Right-click Clock: Quickly set "Clock Latency", "Clock Uncertainty", or "I/O Delay" for the selected Clock, as shown in Figure 3-17.

Figure 3-17 Right-click Menu for the Clock



Note!

- When a constraint conflicts with the PLL configuration, the constraint created by "Create Clock" takes precedence, and a warning will be issued during PnR.
- "Create Clock" does not support the creation of virtual clocks.
- Creating clocks on the negative (N) end of a differential buffer, such as IOB, is not

supported.

- Gowin Software supports timing analysis based on different clock duty cycles.

Create Generated Clock

Create a generated clock.

This constraint allows for operations such as frequency division, frequency multiplication, phase shifting, and duty cycle adjustment based on the master clock, thereby creating the generated clock. The generated clock is automatically linked to the master clock, and the generated clock is automatically corrected to adapt to the master clock when the attributes of the master clock change.

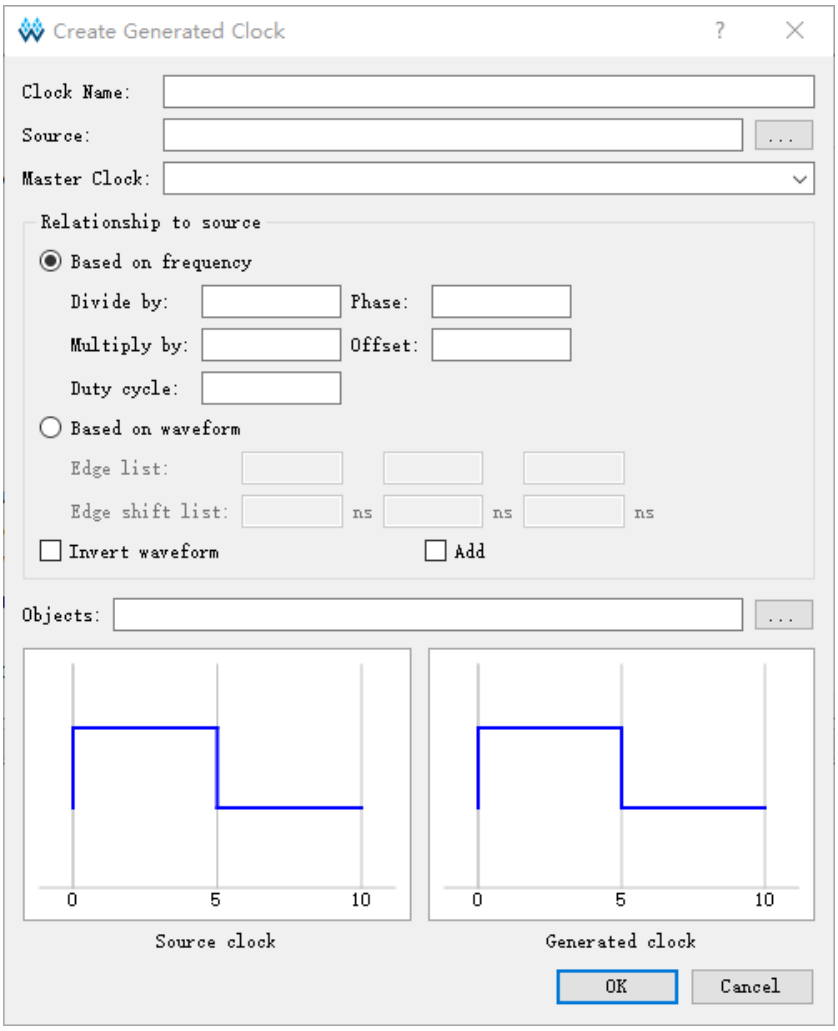
You can create the generated clock in the following two methods:

1. Create via constraints

   a) Select "Create Generated Clock" in "Constraints" menu and "Create Generated Clock" dialogue box pops up, as shown in Figure 3-18.

   - Clock Name: Supports identifiers beginning with letters or underscores.
   - Source: The source of generated clock; click "..." on the right to select one.
   - Master Clock: Click "⌄" to select one.

Figure 3-18 Create Generated Clock Constraints



b) Click the icon "[...]" to the right of Source to select the clock source. The clocks associated with Source are automatically added to the "Master Clock" list. Select one from the "Master Clock" list; when multiple clocks are present in the "Master Clock" list, only one can be selected. All clocks selected as the “Master Clock” must belong to the same group.

c) In "Relationship to source" pane, you can configure frequency division/multiplication, offset, duty cycle and phase based on frequency; and you can also adjust edge based on waveform using an Edge list in combination with an Edge shift list.

- Divide by: Division factor, a positive integer.
- Phase: Floating point value accurate to three decimal places, with negative values indicating left shift and positive values indicating right shift.
- Multiply by: Multiplication factor, a positive integer.
- Offset: Floating point value accurate to three decimal places, with negative values indicating left shift and positive values

indicating right shift.

- Duty cycle: Floating point value accurate to three decimal places.
- Edge list: Positive integer increased sequentially.
- Edge shift list: Floating point value accurate to three decimal places.

d) You can invert clocks by clicking "Invert waveform" and add clocks by clicking "Add", and both are effective during STA analysis.

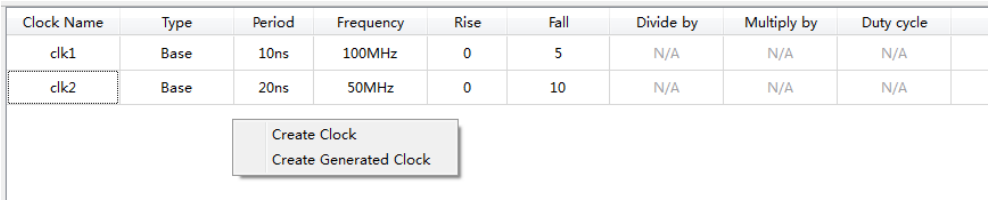
e) In "Objects", click " " and "Select Objects" dialogue box pops up to select the object for the clock.

Note!

When the constraint conflicts with the PLL configuration and the constraint created by Create Clock takes precedence, a warning message will be displayed during PnR.

2. Create generated clock from clocks list. Right-click and select "Create Generated Clock" to create generated clock, as shown in Figure 3-19.

Figure 3-19 Select Create Generated Clock



The new constraints will be added in table editing area.

You can perform the following operations.

- Double-click the corresponding constraint in the "Clocks" list to open the "Generated Clock" editing dialog box, where you can modify the Generated Clock information.
- Select the Generated Clock in the table editing area and right-click to select "Remove" to remove the clock.

Set Clock Latency

It is used to set the clock latency. Clock latency includes two types: network latency and source latency.

- Network latency refers to the latency between the clock and the timing components.
- Source latency refers to the latency between the clock and the source.

Gowin Software will calculate clock network latency automatically, so you only need to set source latency.

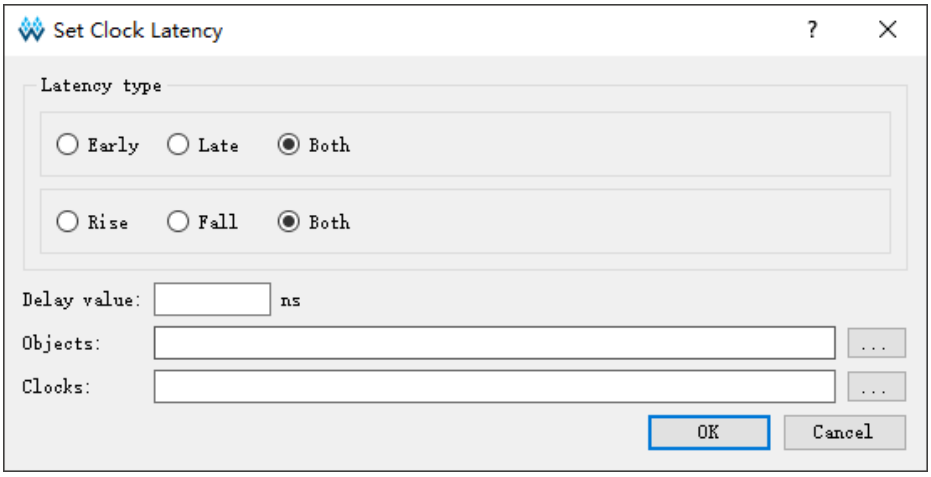
You can create clock latency constraints in the following two methods.

1. Select "Set Clock Latency" in "Constraints" menu, and "Set Clock Latency" dialog box pops up, as shown in Figure 3-20. Fill in the

Latency information, and click “OK” to save the constraint.

- Early and Late: Indicates the earliest and latest latency values for the set clock. "Both" indicates Rise and Fall both are valid.
- Rise and Fall: Valid on the rising edge and falling edge; "Both" indicates Rise and Fall both are valid.
- Delay value: Clock delay value; floating point value accurate to three decimal places, in unit of nanoseconds (ns).
- Objects: Specifies the object; select one by clicking "...".
- Clocks: Specifies a clock; select one by clicking "...".

Figure 3-20 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.

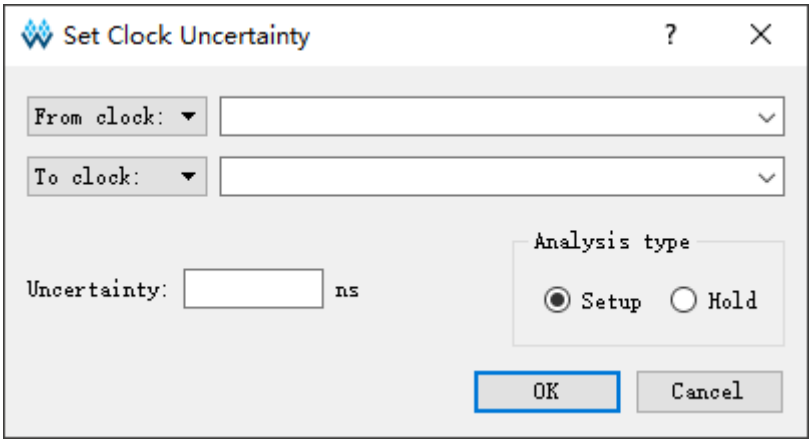
Set Clock Uncertainty

It is used to set clock uncertainty.

The steps to create clock uncertainty are as follows.

1. In "Constraints", select "Set Clock Uncertainty" and "Set Clock Uncertainty" dialogue box pops up, as shown in Figure 3-21.
   - 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: Floating point value accurate to three decimal places, unit in ns.
   - Analysis type: Indicates the type of analysis.

Figure 3-21 Set Clock Uncertainty



2. Use the dropdown menus on the left to select the type of "From" (From clock, Rise from, Fall from) and the type of "To" (To clock, Rise to, Fall to). Then use the dropdown menus on the right to select the target clock from all currently created clocks.
3. Click "OK" to save the constraint and add the uncertainty.

Set Clock Group

It is used to set clock group.

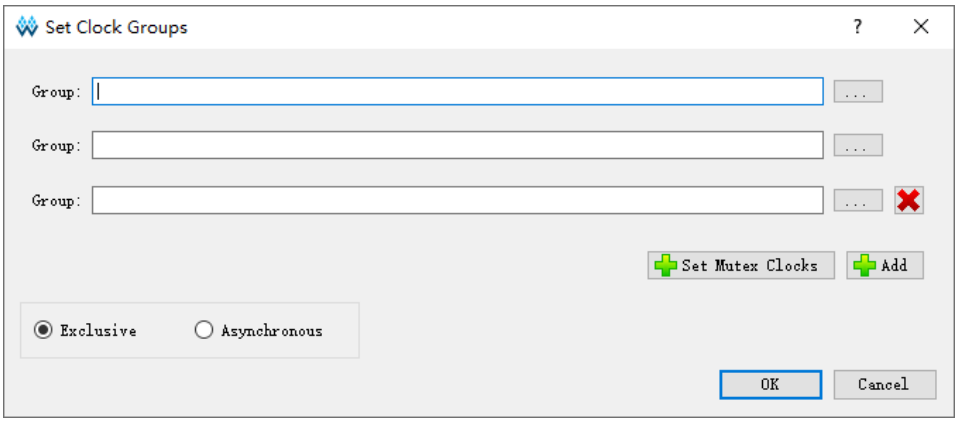
Gowin Software defaults to the fact that all clocks in the design belong to a group and are related.

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

You can create clock group in the following methods.

1. In "Constraints", select "Set Clock Groups" and "Set Clock Groups" dialog box pops up, as shown in Figure 3-22.
   - Group: Specifies at least one clock in the group, selected by clicking "..." on the right.
   - Set Mutex Clocks: Sets multiple clock groups at one time.
   - Add: Add a group.
   - Exclusive: Indicates clocks are exclusive. For example, Clock0 and Clock1 pass through a MUX2 and outputs Clock3 acting on a timing model, and Clock3 can only take Clock0 and Clock1 at the same time; then you can check this option.
   - 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 3-22 Set Clock Groups



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

Note!

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

3.7.2 I/O Constraints

set_input_delay

Set the delay value of data input. Users need to set a reasonable input delay value, and Gowin Software will analyze the timing based on the given delay value. The input delay type in the timing report is "tIn".

set_output_delay

Set the delay value of data output. Users need to set a reasonable output delay value, and Gowin Software will analyze the timing based on the given delay value. The output delay type in the timing report is "tOut".

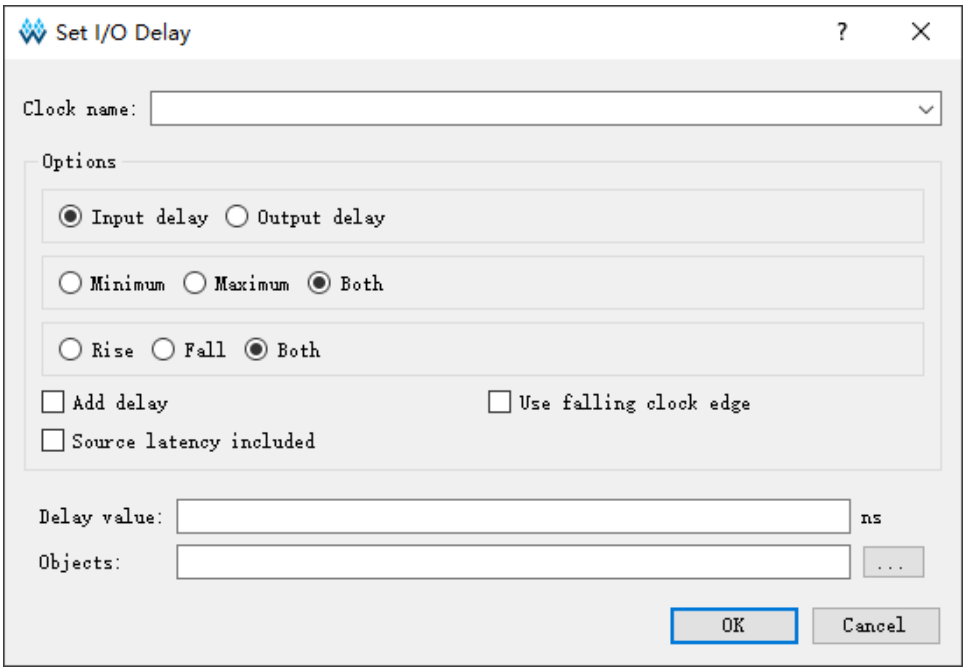
You can create I/O delay constraints as follows.

1. In "Constraints" menu, select "Set I/O Delay" and "Set I/O Delay" dialog box pops up, as shown in Figure 3-23.
   - 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 type, 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: Used to set the delay value for I/O, a positive or negative floating point value accurate to three decimal places. When it is a negative number, it indicates an advance arrival, and when it is a positive number, it indicates a delayed arrival, measured in ns.
- Objects: Specifies the input and output ports, and click "..." on the right to select one.
- Add delay: Used to add a delay value to the same port; when there are multiple delay values on the same port, Gowin Software will select the largest one for Setup analysis and the smallest one for Hold analysis; if this option is not specified, the same constraint on the same port will be overwritten.
- Use falling clock edge: If it is checked, the falling edge of the associated clock is related; the default is rising edge related.
- Source Latency include: If it is checked, it indicates that the set delay value includes the delay value of the clock; if it is not checked, Gowin Software will add the clock delay during calculation.

Figure 3-23 Create I/O Delay Constraints



2. Click "OK" to save the constraints.

3.7.3 Exceptions Constraints

Timing exceptions allow users to modify the default static timing analysis rules for a specific path. The timing exception constraint commands includes four types: set_false_path, set_max_delay, set_min_delay, set_multicycle_path.

Set False Path

It is used to set a false path. Timing analysis will not be performed on the specified timing paths.

You can create False Path constraints as follows:

1. Select "Constraints > Set False Path", then "Set False Path" dialogue box pops up, as shown in Figure 3-24.
   - Analysis type: Specifies the type not to be analyzed; "Both" means neither type will be analyzed.
   - 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 3-24 Create False Path Constraints

2. Click "...." on the right to select objects corresponding to From, To, and Through, as shown in Figure 3-14. Click "OK" to save.

Set Max/Min Delay

It is used to specify the maximum and minimum delay values. The maximum delay is reported in Setup, and the minimum delay is reported in Hold analysis report.

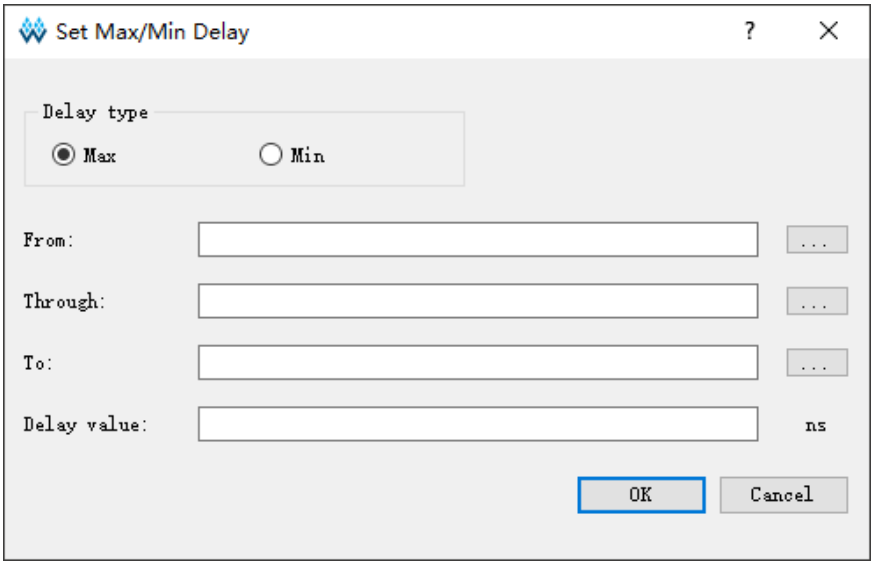
You can create Max/Min Delay constraints as follows:

1. Select "Constraints > Set Max/Min Delay", then "Set Max/Min Delay" dialogue box pops up, as shown in Figure 3-25.
   - From: Indicates the start of the path; click "..." on the right to select one.
   - To: Indicates the end of the path; click "..." on the right to select one.
   - Through: Indicates the pins or nets through the path; click "..." on the right to select one.
   - Delay value: Sets delay value; Floating point value accurate to three decimal places.
   - Delay value is the user-specified delay value, a floating point value accurate to three decimal places, measured in ns.

Note!

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

Figure 3-25 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

It is used to set a multicycle path.

By default, Gowin Software performs single-cycle clock analysis.

You can create multicycle path constraints as follows:

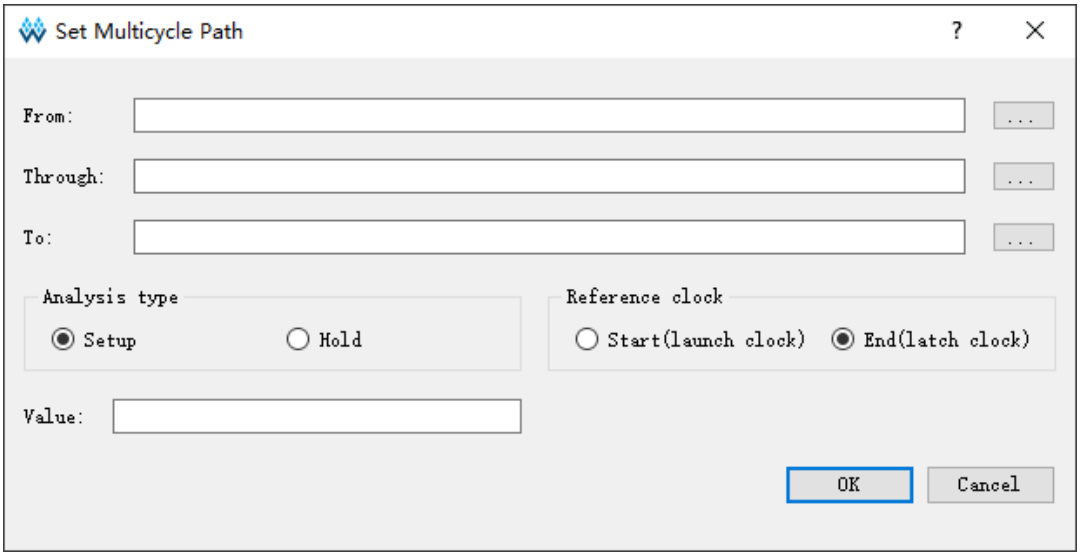
1. Select "Constraints > Set Multicycle Path", then "Set Multicycle Path" dialogue box pops up, as shown in Figure 3-26.

- Reference clock: Indicates whether the reference clock is a launch or a latch clock.
- Analysis type: Specifies constraints on Setup or Hold check.
- From: Indicates the start of the path; click "..." on the right to select one.
- Through: Specifies the NODE that the path passes through; click "..." on the right to select one.
- To: Indicates the end of the path; click "..." on the right 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 3-26 Create Multicycle Path Constraints



2. Click "OK" to save the constraints.

3.7.4 Operating Conditions Constraints

It is used to set operating conditions constraints.

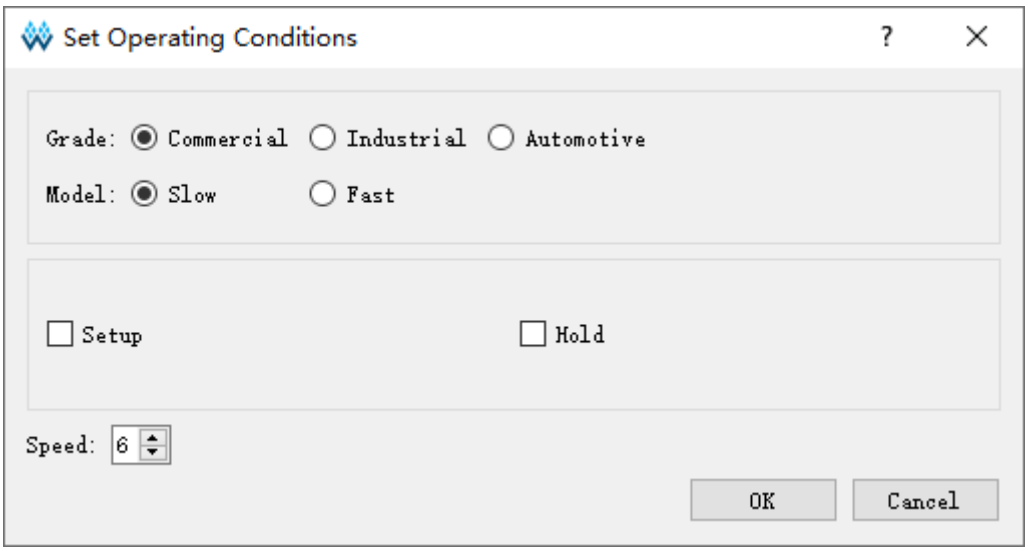
Gowin Software uses Slow Model for Setup analysis and Fast Model for Hold analysis by default. You can also customize the timing model. You can check the delay model in STA Tool Run Summary.

Select "Constraints > Set Operating Conditions", then "Set Operating Conditions" dialogue box pops up, as shown in Figure 3-27.

- Grade: Commercial, industrial and automotive.

- Model: Slow and fast.
- Hold and Min are valid for hold and removal.
- Setup and Max are valid for setup and recovery.
- Max-Min is valid for setup, hold, recovery, and removal.

Figure 3-27 Create Operating Conditions Constraints



Note!

- When the set grade and speed 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.
- If only the setup is configured for Grade and Speed, Hold is analyzed according to the Grade and Speed set by Setup.
- If only the hold is configured for Grade and Speed, Setup is analyzed according to the Grade and Speed set by Hold.
- The engineering sample (ES) uses the slowest speed to analyze the timing by default, and you can set the speed of engineering sample as required.
- For GW2A and GW1N series, the Slow model corresponds to high temperature and low voltage, while the Fast model corresponds to low temperature and high voltage.
- For GW5A series, the Fast model corresponds to high temperature and high voltage, while the Slow model corresponds to low temperature and low voltage.

3.7.5 Reports Constraints

Report Timing

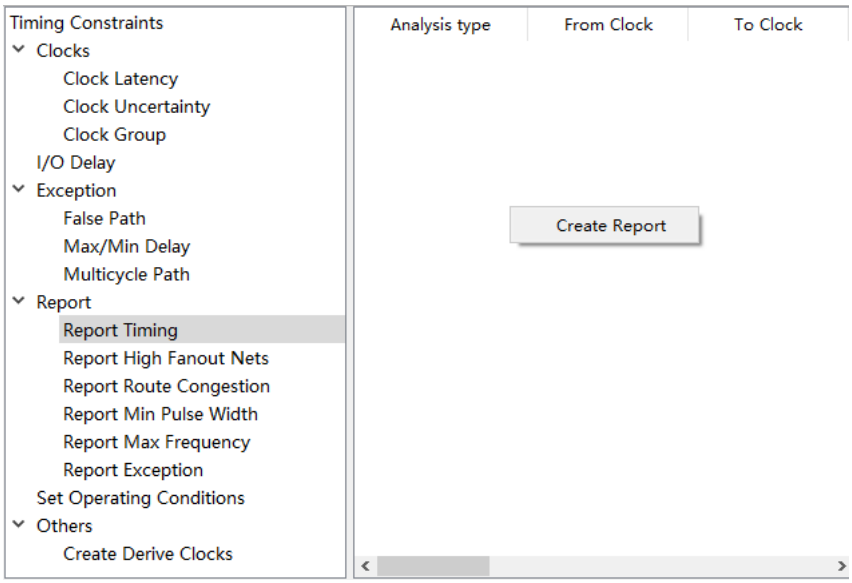
It is used to report timing.

According to the set parameters, Gowin Software can provide reports with more details. The results can be found in the Setup and Hold reports.

The steps are as follows:

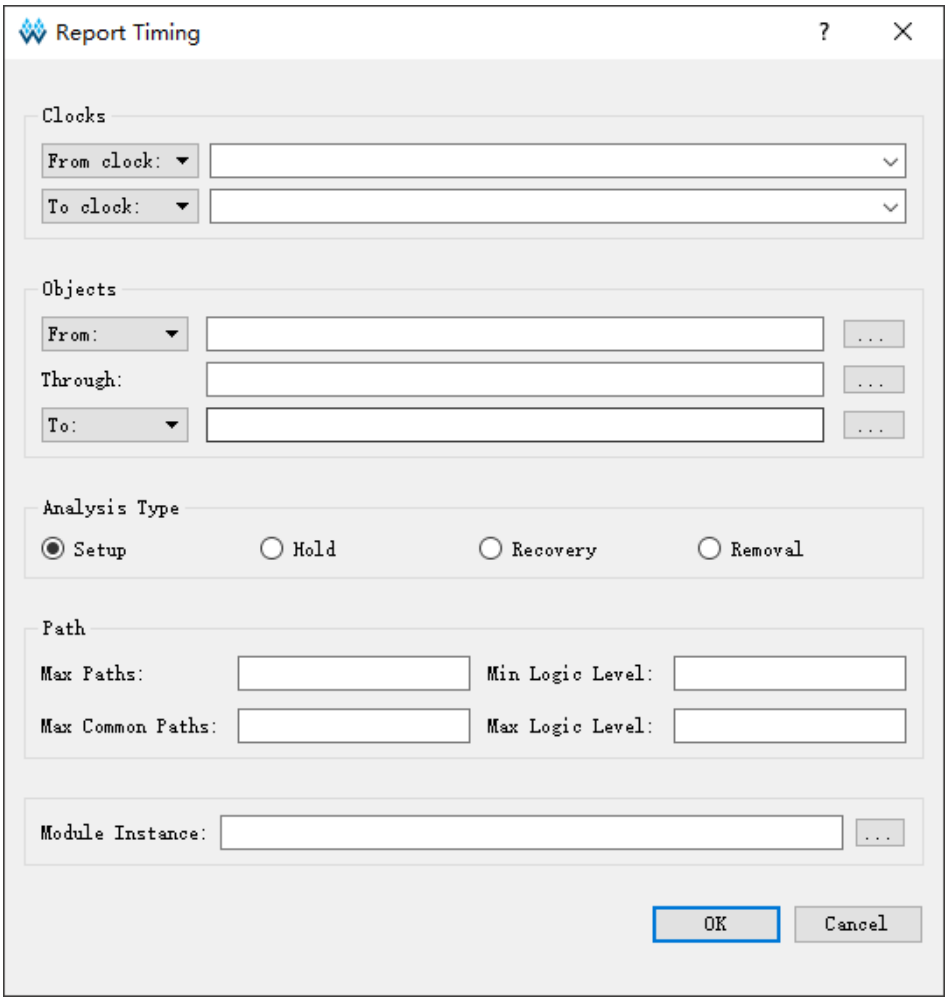
1. Select "Timing Constraints > Report Timing" and right-click on the blank area, then "Create Report" pops up, as shown in Figure 3-28.

Figure 3-28 Create Report Timing



2. Select "Create Report" and Figure 3-29 pops up.
   - Path: Specifies the max. paths, the max. common paths, the max./min. logic level.
   - Clocks: Specifies the associated clock of a path. From/To Clock indicates the launch clock and latch 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 3-29 Report Timing Dialog Box



3. Click "OK" to save the timing report settings.

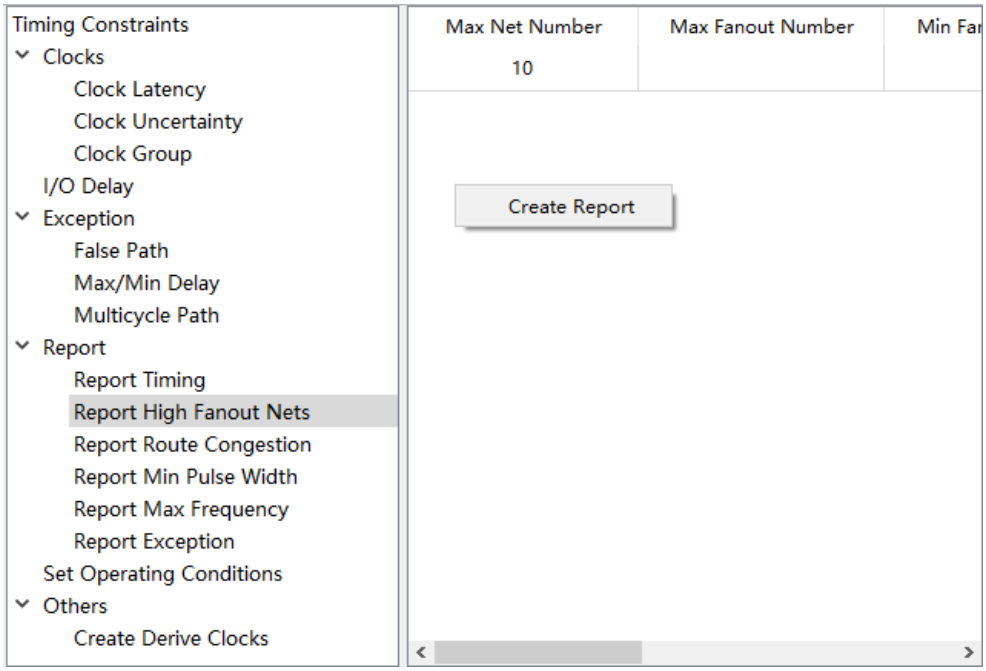
Report High Fanout Nets

It reports high fanout nets, and reports 10 of the largest by default. The results can be viewed in High Fanout Nets Report.

The steps are as follows:

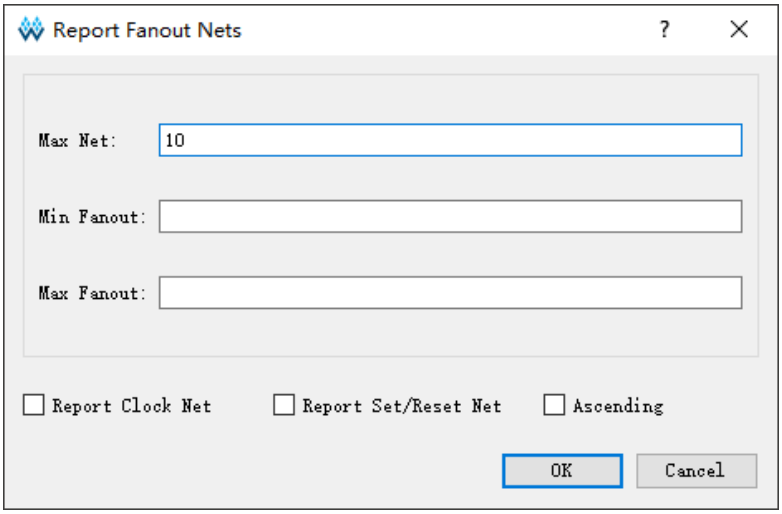
1. Select "Timing Constraints > Report High Fanout Nets".
2. Right-click on the blank area, then "Create Report" pops up, and Create Report dialog box pops up, as shown in Figure 3-30.

Figure 3-30 Create Report High Fanout Nets



3. Select "Create Report" and Figure 3-31 pops up.
   - Max Net: Specifies the maximum of nets, a positive integer.
   - Min and Max Fanout: Specifies the min. and max. fanout, a positive integer.
   - Report Clock Net: Reports the net connected to the clock input.
   - Report Set/Reset Net: Reports the net connected to the LSR input.
   - Ascending: Specifies the net order, ascending by default.

Figure 3-31 Report High Fanout Nets Dialog Box



4. Click "OK" to save the timing report settings.

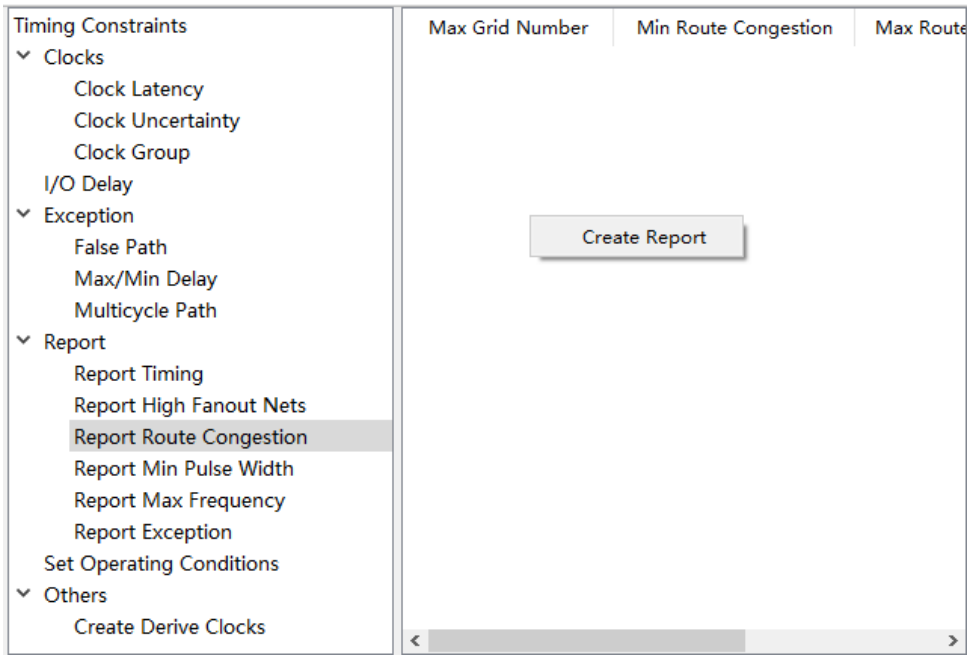
Report Route Congestion

It reports the route congestion, and reports 10 of the worst grid by default. It usually reports the congestion on a specific grid. The results can be seen in Route Congestions Report.

The steps are as follows:

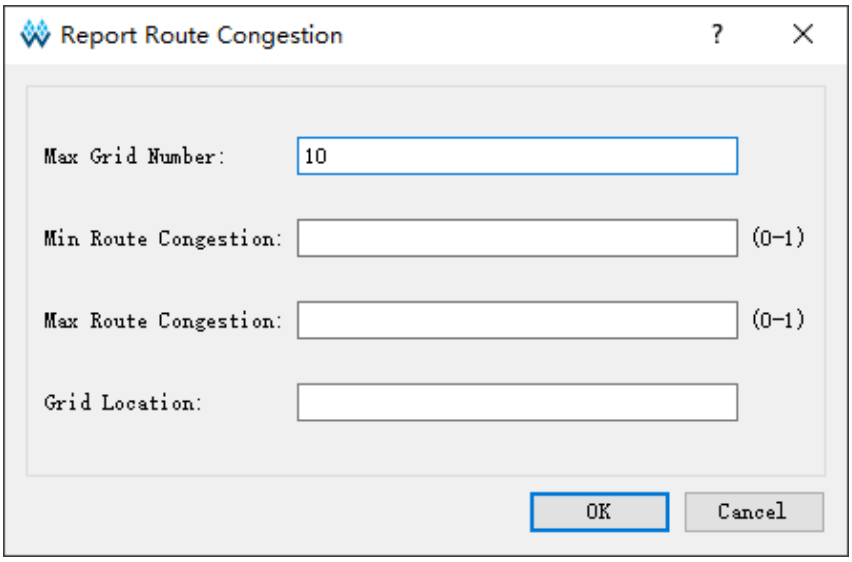
1. Select "Timing Constraints > Report Route Congestion".
2. Right-click on the blank area, then "Create Report" pops up, as shown in Figure 3-32.

Figure 3-32 Create Report Route Congestion



3. Select "Create Report" and Figure 3-33 pops up.
   - Max Grid Number: Specifies max. number of grid.
   - Min and Max Route Congestion: Specifies the min. and max. route congestion; Floating point value accurate to three decimal places.
   - Grid Location: Specifies the grid, such as R4C4.

Figure 3-33 Report Route Congestion Dialog Box



4. Click "OK" to save the timing report settings.

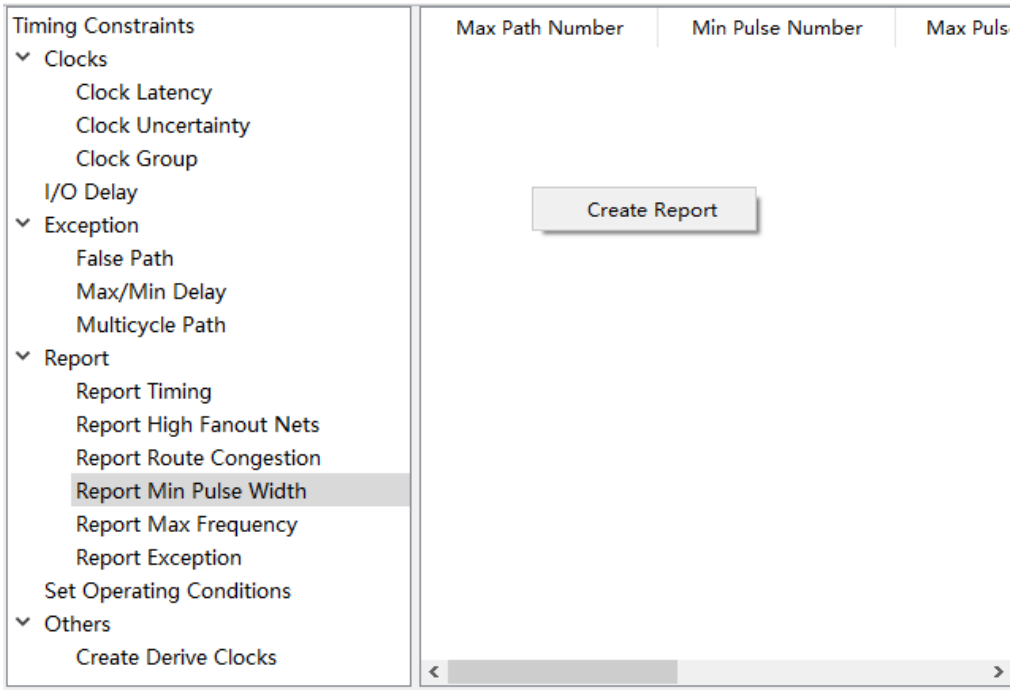
Report Min Pulse Width

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. The report can be viewed in Minimum Pulse Width Report.

The steps are as follows:

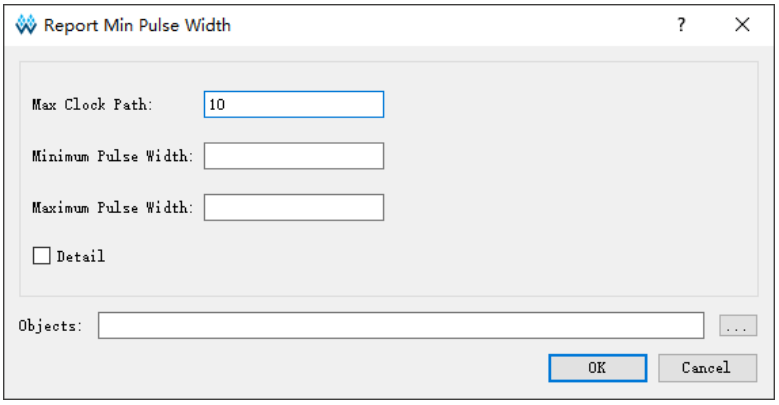
1. Select "Timing Constraints > Report Min Pulse Width".
2. Right-click on the blank area, then "Create Report" pops up, as shown in Figure 3-34.

Figure 3-34 Create Report Min Pulse Width



3. Select "Create Report" and Figure 3-35 pops up.
   - Max Clock Path: Specifies the maximum of paths, a positive integer.
   - Minimum and Maximum Pulse Width: Specifies the min. and max. pulse width; floating point value accurate to three decimal places.
   - 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 "..." on the right to select one.

Figure 3-35 Report Min Pulse Width Dialog Box



4. Click "OK" to save the timing report settings.

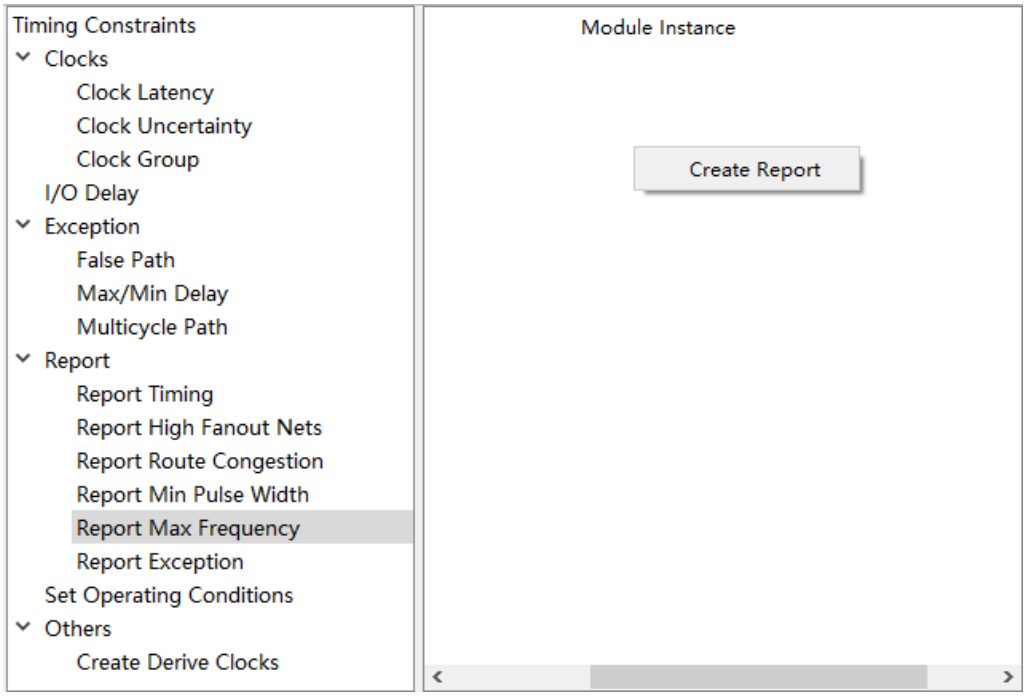
Report Max Frequency

It reports the max. operating frequency and Gowin Software reports the max. frequency of the top by default. Users can specify the max. operating clock frequency for a specific module whose max. operating clock frequency timing critical path is not limited to within this module itself but is related to related to the module's timing endpoints.

The steps are as follows:

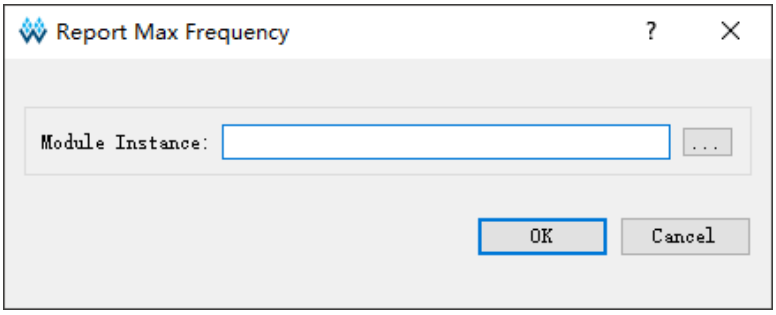
1. Select "Timing Constraints > Report > Report Max Frequency".
2. Right-click in the blank space on the right, and "Create Report" pops up, as shown in Figure 3-36.

Figure 3-36 Create Report Max Frequency



3. Select "Create Report" and Figure 3-37 pops up. Type the instance name in "Module Instance" and click "..." on the right to select one.

Figure 3-37 Report Max Frequency Dialog Box



4. Click "OK" to save.

Report Exception

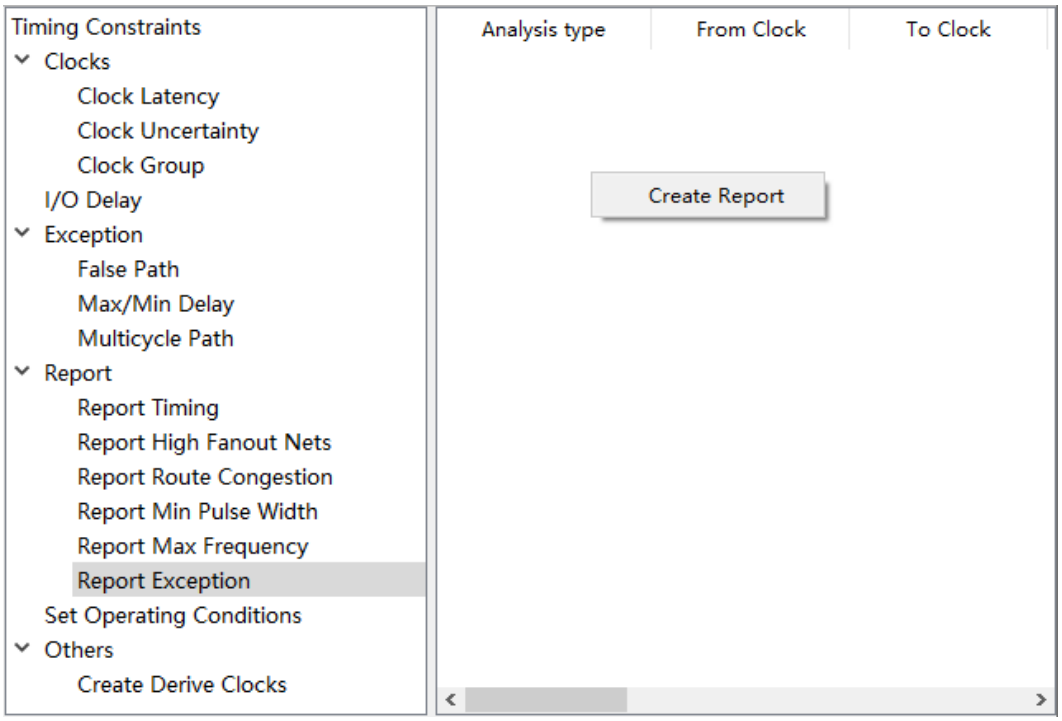
It is used to report exceptions.

Report Exception is used to constrain the timing paths acted upon by the exception constraint statement to report the timing paths that the user cares about.

The steps are as follows.

1. In main interface, select "Timing Constraints > Report > Report Exception".
2. Right-click on the blank area and "Create Report" pops up, as shown in Figure 3-38.

Figure 3-38 Create Report Exception

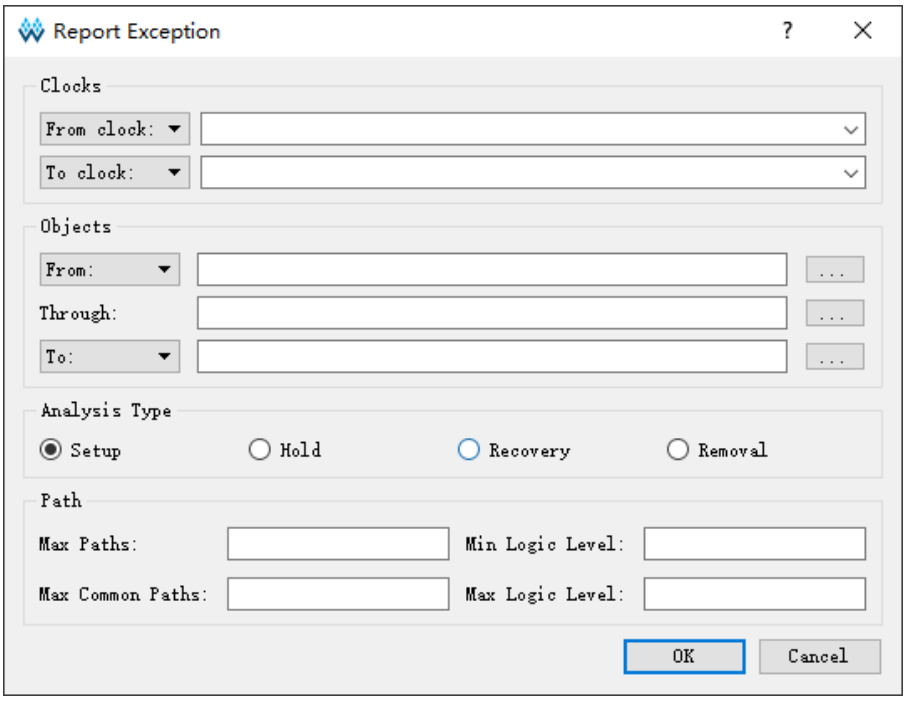


3. Select "Create Report" and Figure 3-39 pops up.

Note!

For the introduction of the options, see Report Timing.

Figure 3-39 Report Exception Dialog Box



4. Click "OK" to save the timing report settings.

3.7.6 Other Constraints

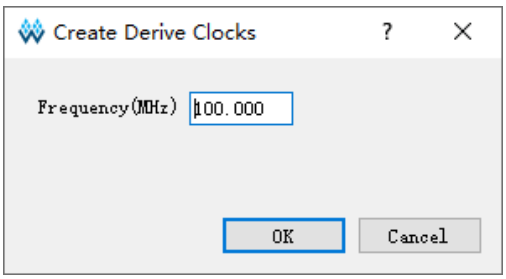
Create Derive Clocks

A global target clock frequency can be created for the design. The frequency can be set, with a maximum of 1200 MHz.

You can add Derive Clocks constraints in the following two methods:

1. Create Derive Clocks through the Constraints menu
   a) Select "Constraints > Create Derive Clocks..." to open the "Create Derive Clocks" dialogue box, as shown in Figure 3-40.

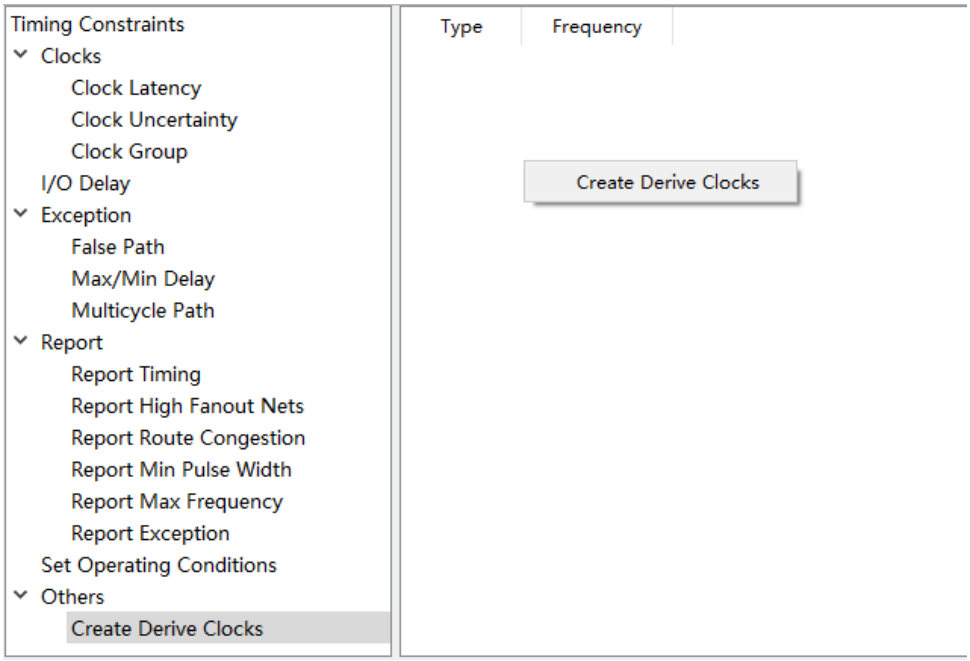
Figure 3-40 Create Derive Clocks



   b) Frequency (MHz): Global target frequency, a positive floating point value less than or equal to 1200, accurate to three decimal places.
2. Click "Others > Create Derive Clocks" to create Derive Clocks.

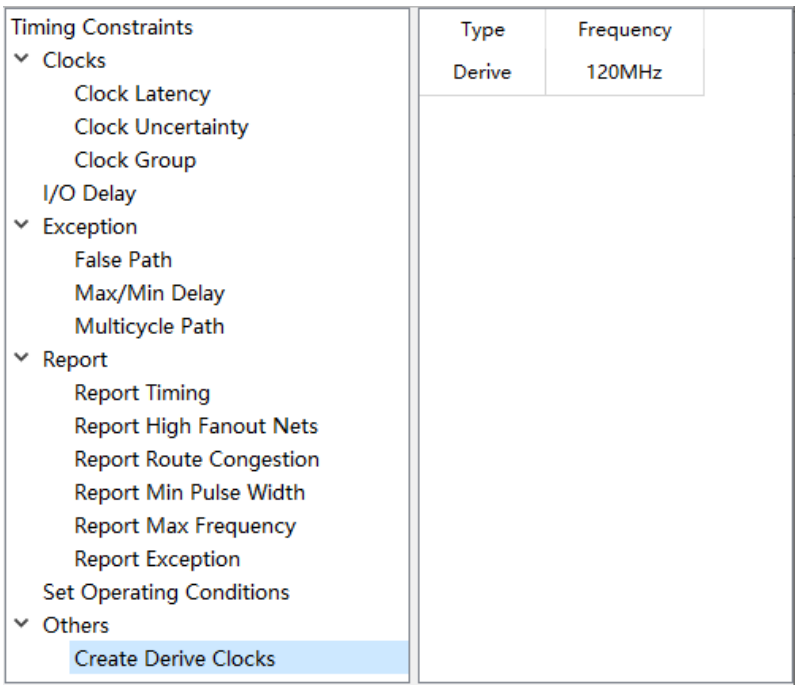
Right-click in the blank space, choose "Create Derive Clocks" to create Derive Clocks, as shown in Figure 3-41.

Figure 3-41 Select Create Derive Clocks



Once the clock is created, the Derive Clocks list will display the corresponding constraints, as shown in Figure 3-42.

Figure 3-42 Derive Clocks List



3.7.7 Save

After editing all constraints, click "File > Save" or "File > Save As" to save the constraints in .sdc file.

3.8 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. derive_clocks
3. set_multicycle_path
4. set_max_delay and set_min_delay
5. set_false_path
6. set_clock_groups

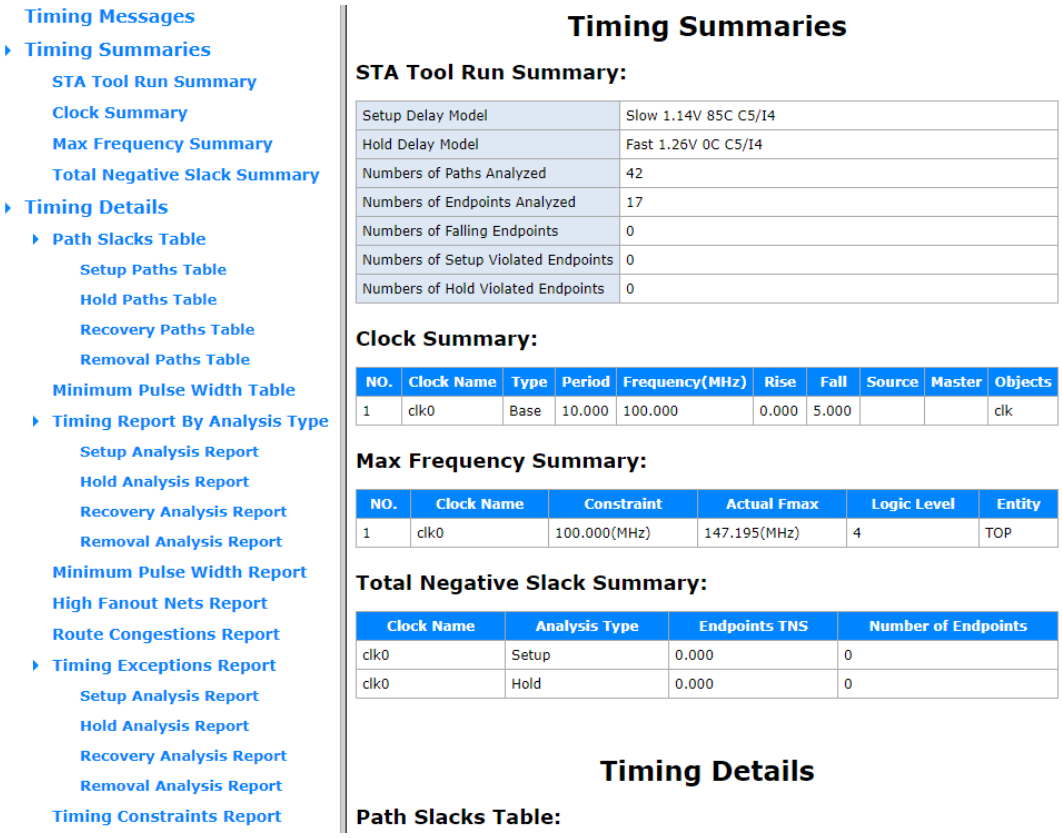
Note!

Only timing constraints that may cause conflicts on the same timing path are sorted; other unspecified constraints do not result in competition between different types of constraints.

4 Timing Report

This chapter will describe Gowin STA to help you learn the timing report. As shown in Figure 4-1, the report includes the navigation bar and the content bar, and the title in the navigation bar will be highlighted in red When there are timing analysis violations.

Figure 4-1 Static Timing Analysis Report



4.1 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 4-2.

Figure 4-2 Timing Summaries

Timing Summaries

STA Tool Run Summary:

Setup Delay Model	Slow 1.14V 85C C5/I4
Hold Delay Model	Fast 1.26V 0C C5/I4
Numbers of Paths Analyzed	42
Numbers of Endpoints Analyzed	17
Numbers of Falling Endpoints	0
Numbers of Setup Violated Endpoints	0
Numbers of Hold Violated Endpoints	0

Clock Summary:

NO.	Clock Name	Type	Period	Frequency(MHz)	Rise	Fall	Source	Master	Objects
1	clk0	Base	10.000	100.000	0.000	5.000			clk

Max Frequency Summary:

NO.	Clock Name	Constraint	Actual Fmax	Logic Level	Entity
1	clk0	100.000(MHz)	147.195(MHz)	4	TOP

Total Negative Slack Summary:

Clock Name	Analysis Type	Endpoints TNS	Number of Endpoints
clk0	Setup	0.000	0
clk0	Hold	0.000	0

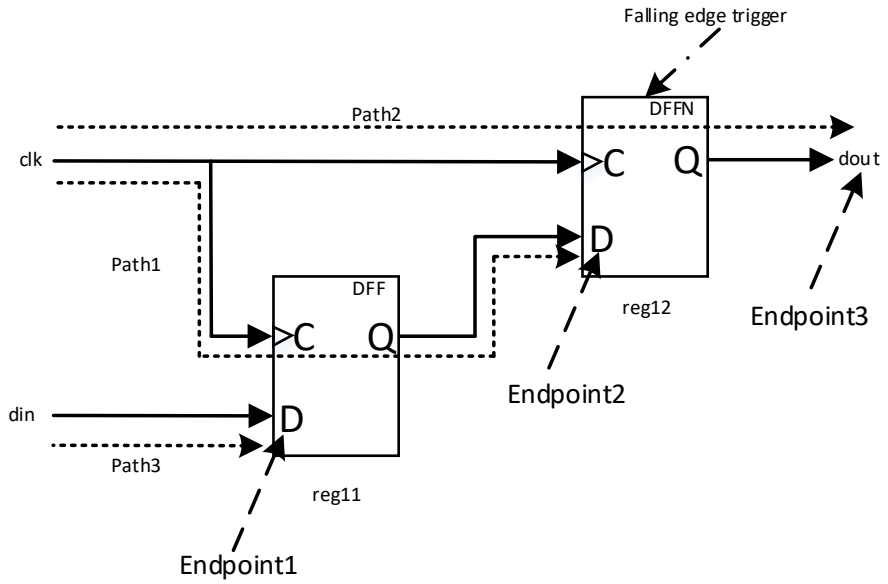
4.1.1 STA Tool Run Summary

- Setup Delay Model: The delay model used for setup time analysis. By default, the Slow model in the device Speed grade is used. For devices supporting both commercial and industrial grades, the commercial Slow model is used by default.
- Hold Delay Model: The delay data model used for hold time analysis. By default, the Fast model in the device Speed grade is used. For devices supporting both commercial and industrial grades, the commercial Fast model is used by default.
- Numbers of Paths Analyzed: The number of static timing analysis paths. As shown in Figure 4-3, three timing paths, Path1, Path2, and Path3, are analyzed.
- Numbers of Endpoints Analyzed: The endpoint of the analysis timing path. As shown in Figure 4-3, 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 4-3, if the reg12 is a Flip-Flop 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 4-3 Path & Endpoints



4.1.2 Clock Summary

It reports all clocks in the user design. If the clock in the design is not constrained, the software will automatically create a clock by default. The clock for Arora family is 100MHz, and the clock for LittleBee family is 50MHz.

- NO.: Number
- Clock Name: The name of the clock.
- Type: Base and Generated. Base represents the base clock and Generated represents the generated clock
- Period: Clock period in nanoseconds (ns)
- Frequency (MHz): Clock frequency
- Rise: Clock rise time
- Fall: Clock fall time
- Source: Clock source from PORT, PIN, NET, REG
- Master: The master clock
- Objects: Clock objects such as PORT, PIN, NET, REG

Note!

- If a clock with the same name already exists, Gowin Software will automatically

append the "_gowin" suffix when creating a new clock by default.

- For PLL, OSC, and CLKDIV types, Gowin Software will automatically append the ".default_gen_clk" suffix to the clock name by default.
- Gowin Software automatically analyzes the relationships among the PLL clock output ports. For example, CLKOUTP is a phase-adjusted and duty cycle-adjusted clock output signal derived from CLKOUT.

4.1.3 Max Frequency Summary

- NO.: 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 obtained by Gowin Software after PnR.
- Logic Level: Logical level.
- Entity: The max. frequency of modules. The default is TOP.

Note!

- When there is no drive timing model after PnR, "No timing paths to get frequency of *" is displayed.
- The maximum clock frequency is reported only for synchronous clocks.
- It is recommended to add complete timing constraints to the design to enable Gowin Software to analyze them more accurately.

4.1.4 Total Negative Slack Summary

- Clock Name: The name of the clock
- Analysis Type: Setup or Hold
- Endpoints TNS: The time of the endpoints with Total Negative Slack (TNS) in timing path driven by clocks (Corresponding ClockName); only the worst paths are counted for common endpoints.
- Number of Endpoints: The number of endpoints with Total Negative Slack (TNS) in the timing path driven by clocks (Corresponding ClockName); only the worst paths are counted for common endpoints.

4.2 Timing Details

4.2.1 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 4-4.

- Path Number: Up to 25 by default.
- Path Slack: It is equal to the time of required by data 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 launch clock and launch edge type 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 type of the next level timing component.
- Relation: The time relationship between the launch clock and the latch clock.
- Clock Skew: The time difference between the arrival of the launch clock (at the source timing component) and the latch clock (at the destination timing component).
- Data Delay: Data delay on the path. The data delay time in the data arrival path, which is a portion of the total delay time for the entire data arrival path.

Note!

- It reports "Nothing to report!" when no timing path is available for analysis.
- The worst 25 paths are printed 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 printed by default. If users do not care about clock domain crossing analysis, you can configure through set_clock_groups or set_false_path.

Figure 4-4 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

4.2.2 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 4-5. The table header is described as follows:

- Number: Ascending order, 10 paths by default.
- Slack: The slack value of the minimum pulse width recognizable by the component.
- 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 "Nothing to report!" when there is no minimum pulse width analysis report.

Figure 4-5 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

4.2.3 Timing Report By Analysis Type

This section includes Setup Analysis Report, Hold Analysis Report, Recovery Analysis Report, and Removal Analysis Report.

Setup Analysis Report

The setup analysis report is used to analyze the time to stabilize 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, time required by data, latch 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 4-6, 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 data arrival at a path.
   - Data Required Time: The time required by data.
   - From: The previous level timing component.
   - To: The next level timing component.
   - Launch Clock: The launch clock that provides edges, including R (Rise) and F (Fall).
   - Latch Clock: The latch clock that provides edges, including R (Rise) and F (Fall).

Figure 4-6 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 4-7, is the path of data arrival, 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.
   - RF: The signal transfer type of the component currently being analyzed.
   - FANOUT: Fanout.
   - LOC: The physical position of the currently analyzed component in the device, and UNPLACE means no location.
   - NODE: Node on the static timing analysis path, including instance name, port, clock, and active clock edge time.

   Note!

   The meanings of some TYPE and NODE entries in Figure 4-7 are as follows:
   - tCL: Time of clock latency; clock source latency.
   - tINS: Time of module instance; component delay.
   - tNET: Time of net; the delay of net.
   - tIn: time of input; data input delay.
   - tC2Q: Time of clock to output; the delay from the clock to the output.
   - active clock edge time: The time when the clock edge becomes active, calculated based on the clock period and the clock model.

Figure 4-7 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 path required by data is the path through

which the clock reaches the clock port from the active edge, as shown in Figure 4-8.

Note!

In Figure 4-8, the meaning of type is as followings.

- tUnc: Time of clock uncertainty
- tOut: Delay time of data output
- tSu: Time of setup

Figure 4-8 Path Required by Data

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				
12.556	-0.480	tSu		1	R5C9[1][A]	reg12_Z

Note!

In the data arrival path and the path required by data, the printing of connection relationships for certain DCE and DQCE is supported: PLL class, OSC class, DLLDLY class, CLKDIV, and DQS-driven DCE and DQCE.

4. Path Statistics is the statistics of the path, as shown in Figure 4-9.
   - Clock Skew: Clock skew.
   - Setup Relationship: The time relationship between the previous level timing component launch data and the next level timing component latch data.
   - Logic Level: Logic levels.
   - Arrival Clock Path Delay: The clock delay time on the Data Arrival Path. Cell indicates the logical delay time; Route indicates the route delay time, and tC2Q indicates the internal delay time.
   - Arrival Data Path Delay: The data delay on the Data Arrival Path.
   - Required Clock Path Delay: The clock delay time on the Data Required Path.

Figure 4-9 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 4-10 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 insufficient, the data will not be stably transmitted to the timing component. Gowin Software calculates and analyzes the arrival time, time required by data, latch clock, and launch 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 4-10 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				
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 4-11 is the recovery time report, which analyzes the shortest stability time for asynchronous clear/set/reset signals before the timing component is on the active clock edge. If the time is not met, the timing components 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 4-11 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				
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 4-12 is the removal time report, which analyzes the shortest stability time for asynchronous clear/set/reset signals after the timing component is on the positive edge. If the time is not met, the timing components may not operate. The analysis and calculation for removal time is the same as that for 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 4-12 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%

4.2.4 Minimum Pulse Width Report

The minimum pulse width report analyzes all the minimum pulse width on the timing analysis path. As shown in Figure 4-13, 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, that is, the minimum time for pulse signal to hold. If it is less than that width, the level pulse will not be recognized.
- Slack: Pulse width slack, defined as the actual pulse width minus the required pulse width. The calculation formula is: $Slack = (T_{Early_clock_Path} - T_{Late_clock_Path}) - T_{\text{Required Width}}$.
- 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.

- Late Clock Path: The path where the clock arrives the latest.
- Early Clock Path: The path where the clock arrives the earliest.

Figure 4-13 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

4.2.5 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 time. The default value is 10. As shown in Figure 4-14, the description is as follows:

- FANOUT: The fanout.
- 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 time on the net.

Figure 4-14 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

4.2.6 Route Congestions Report

As shown in Figure 4-15, 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 based on the ROUTE CONGESTIONS value.

Figure 4-15 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

4.2.7 Timing Exceptions Report

In the following, the Timing Exceptions Report is illustrated by a case.

For the case in Figure 4-16, design a specific SDC file, as shown in Figure 4-17.

Figure 4-16 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 4-17 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
```

In Figure 4-17, the timing exception constraint statement set_max_delay is used to set the maximum absolute delay values on

timing paths influenced by sysclk1 and sysclk2 to 5 ns and 4 ns, respectively. The set_max_delay constraint affects setup analysis, and the affected paths will, by default, be displayed in the timing exception report. The default report generated is shown in Figure 4-18.

Figure 4-18 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				
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 4-17, as shown in Figure 4-19, 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 4-19 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 4-19, the timing exceptions report is as shown in Figure 4-20.

Figure 4-20 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				
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%

4.2.8 Timing Constraints Report

As shown in Figure 4-21, the description is as follows:

- SDC Command Type: Type of static timing constraint command.
- State: Includes 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.

Figure 4-21 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}]

5 Syntax Specification

Gowin syntax specification supports the wildcard character "?" and "*"; "?" matches a single character, and "*" matches zero or more characters and supports cross-hierarchy matching. It also supports timing constraints in multiple lines.

5.1 Object Collections

Table 5-1 Collection Description

Collection	Description
get_regs	Only retrieves flip-flop (FF) objects.
get_pins	Retrieves pin objects.
get_ports	Retrieves port objects.
get_clocks	Retrieves clock objects.
get_nets	Retrieves net objects.
all_clocks	Retrieves all clock objects.
all_inputs	Retrieves all input ports.
all_outputs	Retrieves all output ports.

5.2 Clock Constraints

5.2.1 create_clock

It is used to create clocks. The create_clock has higher priority than derive_clocks.

Syntax

create_clock

-name <*clock_name*>

-period <*period_value*>

[-waveform <*edge_list*>]

<objects>

[-add]

Parameters

Table 5-2 create_clock Parameters

-name <clock_name>	Specifies the clock name, which serves as the unique identifier for the clock. If multiple clocks with the same name are created, the later one will overwrite the previous one.
-period <period_value>	Specifies the clock period in nanoseconds. The value must be a positive real number.
-waveform <edge_list>	Specifies the rising and falling edges of the clock waveform (in nanoseconds) within a full clock period. The two time points must be non-negative, increasing values, and their difference must be less than one clock period. If this option is omitted, the default waveform is used, where the rising edge is at 0 ns and the falling edge is at *<period_value>*/2 ns. Example: Example 5.1
<objects>	Specifies the target objects. Supported options include get_ports, get_nets, get_regs, get_pins, all_inputs, and all_outputs.
-add	When creating multiple clocks on the same <objects>, the -add option must be used starting from the second create_clock command onward. If this option is omitted, any additional clocks created on the same <objects> will be ignored. Example: Example 5.2

Examples

Example 5.1

// Use wildcard “?” to create a clock named user_clk on port clk, with a period of 10 ns, rising edge at 0 ns, and falling edge at 5 ns

create_clock -name user_clk -period 10 -waveform {0 5} [get_ports {c?k}]

Example 5.2

// Create two clocks on the same port clk. The second clock must use the -add.

create_clock -name user_clk0 -period 10 -waveform {0 5} [get_ports {clk}]

create_clock -name user_clk1 -period 20 -waveform {0 10} [get_ports {clk}] -add

5.2.2 create_generated_clock

It is used to create a generated clock.

Syntax

create_generated_clock

-name <*clock_name*>

-source <*clock_source*>

[-master_clock <*clock*>]

[-edges <*edge_list*>]

[-edge_shift <*shift_list*>]

[-divide_by <*factor*>]

[-multiply_by <*factor*>]

[-duty_cycle <*percent*>]

[-phase <*degrees*>]

[-offset <*time*>]

[-invert]

[-add]

<*objects*>

Parameters

Table 5-3 create_generated_clock Parameters

-name <clock_name>	Specifies the name of the generated clock, which serves as the unique identifier for the clock. If multiple clocks with the same name are created, the later one will overwrite the previous one. If the generated clock name is the same as the master clock name, the generated clock creation will be ignored.
-source <clock_source>	Specifies the source of the generated clock, supports get_ports, get_nets, get_regs, and get_pins. If multiple clocks exist on the source, the -master_clock option must be used to specify which clock to use.
-master_clock <clock>	Specifies the master clock of the generated clock. Example: Example 5.3

-edges <edge_list>	A list of three increasing positive integers that define the generated clock's first rising edge, first falling edge, and second rising edge. The integers in the list represent edges extracted from the source clock in an alternating "rising–falling–rising" order; 1 refers to the first rising edge; 2 refers to the following falling edge; 3 refers to the next rising edge, and so on. -edges cannot be used together with -divide_by, -multiply_by, -duty_cycle, -phase, or –offset. Example: Example 5.4
-edge_shift <shift_list>	A list of three real numbers represent the offset (in ns) of the three clock edges defined in -edges. This option must be used together with -edges. The shifts must not cause edge overlap or misalignment with adjacent edges. -edge_shift cannot be used with -divide_by, -multiply_by, -duty_cycle, -phase, or -offset. Example: Example 5.4
-divide_by <factor>	Specifies a divide factor. It must be a positive integer. Default is 1. Example: Example 5.3, Example 5.6, Example 5.8
-multiply_by <factor>	Specifies a multiply factor. It must be a positive integer. Default is 1. Example: Example 5.5 and Example 5.8
-duty_cycle <percent>	Specifies the duty cycle as a percentage in the range (0, 100). Default is 50%. Example: Example 5.5
-phase <degrees>	Specifies the clock phase in degrees. Default is 0°. A positive value shifts the generated clock to the right relative to the master clock, while a negative value shifts it to the left relative to the master clock. Example: Example 5.8
-offset <time>	Specifies the offset in nanoseconds. Default is 0 ns. A positive value shifts the generated clock to the right by the specified amount relative to the master clock, while a negative value shifts it to the left by the specified amount relative to the master clock. Example: Example 5.7
-invert	Inverts the clock. Example: Example 5.7 and Example 5.8
-add	When creating two or more different clocks on the same object, the -add option must be specified; otherwise, the

	subsequently created clocks will be ignored. Example: Example 5.8
<objects>	Supports get_ports, get_nets, get_regs, get_pins, all_inputs, and all_outputs.

Examples

Example 5.3

//Create two master clocks, user_clk0 and user_clk1, on port clk0, and create a divided-by-2 generated clock user_clk2 on port clk1, with user_clk1 as the master clock.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 20 [get_ports {clk0}] -add
create_generated_clock -name user_clk2 -source [get_ports {clk0}] -master_clock user_clk1 -divide_by 2 [get_ports {clk1}]
```

Example 5.4

//Create a generated clock equivalent to a divide-by-2 clock using the -edges option, with each edge offset by 0.2 ns

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_generated_clock -name user_clk1 -source [get_ports {clk0}] -edges {1 3 5} -edge_shift {0.2 0.2 0.2} [get_ports {clk1}]
```

Example 5.5

//Create a generated clock with 3x frequency and 40% duty cycle

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_generated_clock -name user_clk1 -source [get_ports {clk0}] -multiply_by 3 -duty_cycle 40 [get_ports {clk}]
```

Example 5.6

//Create a generated clock with divide-by-2 and 90-degree phase shift

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_generated_clock -name user_clk1 -source [get_ports {clk0}] -divide_by 2 -phase 90 [get_ports {clk1}]
```

Example 5.7

//Create a generated clock that is inverted and shifted right by 4 ns relative to the master clock

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
```

create_generated_clock -name user_clk1 -source [get_ports {clk0}] -offset 4 -invert [get_ports {clk1}]

Example 5.8

//Create a second generated clock on port clk0 with divide-by-2, 3x multiply, and inverted output using -add

create_clock -name user_clk0 -period 10 [get_ports {clk0}]

create_generated_clock -name user_clk1 -source [get_ports {clk0}] -divide_by 2 -phase 90 [get_ports {clk1}]

create_generated_clock -name user_clk2 -source [get_ports {clk0}] -divide_by 2 -multiply_by 3 -invert -add [get_ports {clk1}]

5.2.3 set_clock_latency

It is used to set clock latency.

Syntax

set_clock_latency

-source

[-rise | -fall]

[-late | -early]

<*delay*>

<*objects*>

[-clock <*clock_list*>]

Parameters

Table 5-4 set_clock_latency Parameters

-source	Specifies the analysis type as clock source latency.
-rise | -fall	Defines the latency for the rising or falling edge of the clock. If both are omitted, the value applies to both -rise and -fall. If only -rise is set, -fall will use the same value, and vice versa. Example: Example 5.9
-late | -early	Specifies the latest or earliest clock latency. For setup and recovery analysis: The latest clock latency is applied on the data arrival path. The earliest clock latency is applied on the data required path. For hold and removal analysis: The earliest clock latency is applied on the data arrival path. The latest clock latency is

	applied on the data required path. If only -late is set, -early takes the same value, and vice versa. If the value of -early is greater than -late, -early will use the -late value. If both are omitted, the value applies to both -late and -early. Example: Example 5.10
<delay>	A real number in nanoseconds (ns).
<objects>	Supports get_clocks, get_ports, get_nets, get_regs, get_pins and all_clocks。
-clock <clock_list>	Specifies the clock(s). If this option is omitted, it applies to all clocks associated with the <objects>. Example: Example 5.11

Examples

Example 5.9

// user_clk0 and user_clk1 are the transmit and sample clocks, respectively. Set a 0.22 ns latency on user_clk0 (used in Data Arrival Path), and 0.33 ns on user_clk1 (used in Data Required Path).

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 10 [get_ports {clk1}]
set_clock_latency -source 0.22 -rise [get_clocks { user_clk0}]
set_clock_latency -source 0.33 -fall [get_clocks { user_clk1}]
```

Example 5.10

// Set the latest clock latency to 0.33 ns and the earliest clock latency to 0.11 ns for the clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_clock_latency -source -late 0.33 [get_clocks {user_clk}]
set_clock_latency -source -early 0.11 [get_clocks {user_clk}]
```

Example 5.11

// Set a 0.5 ns latency for user_clk1 on port clk, applied to both rising/falling edges and earliest/latest times

```
create_clock -name user_clk0 -period 10 [get_ports {clk}]
create_clock -name user_clk1 -period 20 [get_ports {clk}] -add
set_clock_latency -source 0.5 [get_ports {clk}] -clock [get_clocks
```

{user_clk1}]

5.2.4 set_clock_uncertainty

It is used to set clock uncertainty.

Syntax

set_clock_uncertainty

<uncertainty>

-setup | -hold

[-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>*]

Parameters

Table 5-5 set_clock_uncertainty Parameters

<uncertainty>	A real number in nanoseconds (ns).
-setup | -hold	-setup: Specifies that the uncertainty analysis applies to setup and recovery. -hold: Specifies that the uncertainty analysis applies to hold and removal. If both options are omitted, the uncertainty is applied to setup, hold, recovery, and removal analyses. Example: Example 5.12 and Example 5.13
-from <from_clock>	Applies uncertainty to the launch clock; Supports get_clocks. Example: Example 5.12
-rise_from <rise_from_clock>	Applies uncertainty to the rising edge of the launch clock; Supports get_clocks. Example: Example 5.13
-fall_from <fall_from_clock>	Applies uncertainty to the falling edge of the launch clock; Supports get_clocks. Example: Example 5.14
-to <from_clock>	Applies uncertainty to the latch clock; Supports get_clocks. For the same timing path, if the latency specified by -to differs from -from, -rise_from, or -fall_from, the latency specified by -to will be used. Example: Example 5.15
-rise_to <rise_to_clock>	Applies uncertainty to the rising edge of latch clock; Supports get_clocks. For the same timing path, if the latency specified by -rise_to differs from -from, -rise_from, or -fall_from, the latency

	specified by -to will be used. Example: Example 5.13
-fall_to <fall_to_clock>	Applies uncertainty to the falling edge of the latch clock. Supports get_clocks. For the same timing path, if the latency specified by -fall_to differs from -from, -rise_from, or -fall_from, the latency specified by -to will be used. At least one of the following options must be specified: -from, -rise_from, -fall_from, -to, -rise_to, -fall_to. Example: Example 5.14

Examples

Example 5.12

// For setup and recovery analysis, set an uncertainty value of 0.2 ns for the launch clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_clock_uncertainty 0.2 -setup -from [get_clocks {user_clk}]
```

Example 5.13

// For hold and removal analysis, set an uncertainty of 0.111 ns on the rising edge of the launch clock user_clk0, and 0.222 ns on the rising edge of the latch clock user_clk1.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 10 [get_ports {clk1}]
set_clock_uncertainty 0.111 -hold -rise_from [get_clocks {user_clk0}]
set_clock_uncertainty 0.222 -hold -rise_to [get_clocks {user_clk1}]
```

Example 5.14

// For hold and removal analysis, set an uncertainty of 0.111 ns on the falling edge of the launch clock user_clk0, and 0.222 ns on the falling edge of the latch clock user_clk1.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 20 [get_ports {clk1}]
set_clock_uncertainty -hold 0.111 -fall_from [get_clocks {user_clk0}]
set_clock_uncertainty -hold 0.222 -fall_to [get_clocks {user_clk1}]
```

Example 5.15

// For setup and recovery analysis, specify 0.2 ns uncertainty for

launch clock user_clk0 and 0.3 ns for latch clock user_clk1. Gowin Software will use the value of 0.3 ns.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 10 [get_ports {clk1}]
set_clock_uncertainty -setup 0.2 -from [get_clocks {user_clk0}]
set_clock_uncertainty -setup 0.3 -to [get_clocks {user_clk1}]
```

5.2.5 set_clock_groups

It is used to set clock groups.

Syntax

```
set_clock_groups
    -asynchronous | -exclusive
    [-group <clocks>]
    [-group <clocks>]
    …
```

Parameters

Table 5-6 set_clock_groups Parameters

-asynchronous | -exclusive	Specifies the relationship between clock groups as asynchronous or exclusive. Example: Example 5.16 and Example 5.17
-group *<clocks>*	Defines a clock group. Clocks can be specified using get_clocks.

Examples

Example 5.16

// Clocks clk1 and clk2 have timing paths between each other. Clock groups and an asynchronous relationship are defined for clk1 and clk2, respectively.

```
create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk0}]
create_clock -name clk2 -period 10 -waveform {0 5} [get_ports {clk1}]
set_clock_groups -asynchronous -group [get_clocks {clk1}] -group [get_clocks {clk2}]
```

The analysis results are shown in the below table.

Table 5-7 Analysis Results

Latch Clock \ Launch Clock	clk1	clk2
clk1	Analyze	Not Analyze
clk2	Not Analyze	Analyze

Example 5.17

// There are timing paths among clocks clk1, clk2, and clk3; clk1 and clk2 are grouped, and clk3 is placed in a separate group.

create_clock -name clk1 -period 10 -waveform {0 5} [get_ports {clk0}]

create_clock -name clk2 -period 10 -waveform {0 5} [get_ports {clk1}]

create_clock -name clk3 -period 10 -waveform {0 5} [get_ports {clk2}]

set_clock_groups -exclusive -group [get_clocks {clk1 clk2}] -group [get_clocks {clk3}]

The analysis results are shown in the below table.

Table 5-8 Analysis Results

Latch Clock \ Launch Clock	clk1	clk2	clk3
clk1	Analyze	Analyze	Not Analyze
clk2	Analyze	Analyze	Not Analyze
clk3	Not Analyze	Not Analyze	Analyze

5.3 I/O Constraints

It is used to set the delay of data input and output.

Syntax

set_input_delay / set_output_delay

-clock *<name>*

<delay_value>

[-clock_fall]

[-max | -min]

[-rise | -fall]

[-add_delay]

[-source_latency_included]

<objects>

Parameters

Table 5-9 I/O Constraints Parameters

-clock <name>	Specifies the clock, which must be a previously created clock.
<delay_value>	A real number in nanoseconds (ns).
-clock_fall	Specifies that the delay is relative to the falling edge of the clock. If omitted, the input and output delay is relative to the rising edge by default. Example: Example 5.18 and Example 5.23
-max | -min	-max: Sets the maximum input and output delay value, used for setup and recovery analysis. -min: Sets the minimum input and output delay value, used for hold and removal analysis. If only -max is specified, -min will take the same value, and vice versa. If both are omitted, max and min are set equally. Example: Example 5.19 and Example 5.24
-rise | -fall	Specifies delay values for rising or falling edges of the input/output port. If only one is specified, the other inherits the same value. If both are omitted, the delay values for rising and falling edges are considered the same. Example: Example 5.20 and Example 5.24
-add_delay	Allows multiple delay constraints to apply simultaneously. Gowin Software performs setup (recovery) and hold (removal) analysis based on the configured values. Examples: Example 5.21 and Example 5.26
-source_latency_included	Indicates that the input and output delay includes the source clock latency. If omitted, the delay does not include the associated clock's source latency by default. Example: Example 5.22 and Example 5.27
<objects>	The input or output port objects. Supports get_ports, all_inputs, or all_outputs.

Note!

set_input_delay does not support clock input ports.

Examples

Example 5.18

// Set an input delay of 0.2 ns for port d, relative to the falling edge of clock user_clk

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_input_delay -clock user_clk 0.2 -clock_fall [get_ports {d}]
```

Example 5.19

// Set the maximum input delay to 0.2 ns and the minimum input delay to 0.1 ns for input port d, relative to the associated clock.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_input_delay -clock user_clk 0.2 -max [get_ports {d}]
set_input_delay -clock user_clk 0.1 -min [get_ports {d}]
```

Example 5.20

// Set the input delay for port d to 0.2 ns on the rising edge and 0.1 ns on the falling edge, relative to the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_input_delay -clock user_clk 0.2 -rise [get_ports {d}]
set_input_delay -clock user_clk 0.1 -fall [get_ports {d}]
```

Example 5.21

// Set two input delays of 0.2 ns and 0.4 ns for port d, relative to the associated clock user_clk. For setup and recovery analysis, tIn is taken as 0.4 ns. For hold and removal analysis, tIn is taken as 0.2 ns.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_input_delay -clock user_clk 0.2 [get_ports {d}]
set_input_delay -clock user_clk 0.4 -add_delay [get_ports {d}]
```

Example 5.22

// Set an input delay of 0.2 ns for port d, including the source latency of 0.15 ns from the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_clock_latency -source 0.15 [get_ports {clk}] -clock [get_clocks {user_clk}]
set_input_delay -clock user_clk 0.2 -source_latency_included [get_ports {d}]
```

Example 5.23

// Set an output delay of 0.2 ns for port q, relative to the falling edge of the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_output_delay -clock user_clk 0.2 -clock_fall [get_ports {q}]
```

Example 5.24

// Set the maximum output delay to 0.2 ns and the minimum output delay to 0.1 ns for port q, relative to the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_output_delay -clock user_clk 0.2 -max [get_ports {q}]
set_output_delay -clock user_clk 0.1 -min [get_ports {q}]
```

Example 5.25

// Set the output delay for port q to 0.2 ns on the rising edge and 0.1 ns on the falling edge, relative to the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_output_delay -clock user_clk 0.2 -rise [get_ports {q}]
set_output_delay -clock user_clk 0.1 -fall [get_ports {q}]
```

Example 5.26

// Set two output delays of 0.2 ns and 0.4 ns for port q, relative to the associated clock user_clk. For setup and recovery analysis, tOut is -0.2 ns. For hold and removal analysis, tOut is -0.4 ns.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_output_delay -clock user_clk 0.2 [get_ports {q}]
set_output_delay -clock user_clk 0.4 -add_delay [get_ports {q}]
```

Example 5.27

// Set an output delay of 0.2 ns for port q, including the source latency of 0.15 ns from the associated clock user_clk.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_clock_latency -source 0.15 [get_ports {clk}] -clock [get_clocks {user_clk}]
set_output_delay -clock user_clk 0.2 -source_latency_included [get_ports {q}
```

5.4 Exception Constraints

5.4.1 set_false_path

It is used to set a false path. Timing analysis will no longer be performed on the specified timing path.

Syntax

set_false_path

[-from <*from_list*>]

[-through <*through_list*>]

[-to <*to_list*>]

[-setup | -hold]

Parameters

Table 5-10 set_false_path Parameters

-from <from_list>	Start point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. Example: Example 5.28 and Example 5.29
-through <through_list>	Through points; supports get_nets and get_pins. Example: Example 5.30
-to <to_list>	End point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. At least one of -from, -through, or -to must be specified. Example: Example 5.28
-setup | -hold	-setup is valid for setup and recovery analysis. -hold is valid for hold and removal analysis. If omitted, it defaults to being valid for setup, hold, recovery, and removal analyses. Example: Example 5.29 and Example 5.30

Examples

Example 5.28

// Set all timing paths from launch clock user_clk0 to latch clock user_clk1 as false paths; valid for setup, hold, recovery, and removal analysis.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 5 [get_ports {clk1}]
```

set_false_path -from [get_clocks {user_clk0}] -to [get_clocks {user_clk1}]

Example 5.29

// Set all timing paths starting from port d as false paths; valid only for setup and recovery analysis.

set_false_path -from [get_ports {d}] -setup

Example 5.30

// Set all timing paths passing through the D input of flip-flop reg0 as false paths; valid only for hold and removal analysis.

set_false_path -through [get_pins {reg0_s0/D}] -hold

5.4.2 set_max_delay/set_min_delay

It is used to set the maximum or minimum delay value.

Syntax

set_max_delay/set_min_delay

[-from <*from_list*>]

[-through <*through_list*>]

[-to <*to_list*>]

<*delay_value*>

Parameters

Table 5-11 set_max_delay/set_min_delay Parameters

-from <from_list>	Start point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. Example: Example 5.31
-through <through_list>	Through points; supports get_nets and get_pins. Example: Example 5.32
-to <to_list>	End point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. At least one of -from, -through, or -to must be specified. Example: Example 5.31 and Example 5.33
<delay_value>	Real number in nanoseconds. Maximum delay is valid for setup and recovery analysis. Minimum delay is valid for hold and removal analysis.

Examples

Example 5.31

// Set max delay to 5 ns and min delay to 2 ns for paths from launch clock user_clk0 to latch clock user_clk1. 5 ns applies to setup/recovery; 2 ns to hold/removal.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 5 [get_ports {clk1}]
set_max_delay -from [get_clocks {user_clk0}] 5 -to [get_clocks {user_clk1}]
set_min_delay -from [get_clocks {user_clk0}] 2 -to [get_clocks {user_clk1}]
```

Example 5.32

// Set a minimum delay of 0.2 ns for all timing paths passing through the D input of flip-flop reg0.

```
set_min_delay -through [get_pins {reg0_s0/D}] 0.2
```

Example 5.33

// Set a max delay of 6 ns and min delay of 0.5 ns for all timing paths ending at flip-flop reg0.

```
set_max_delay -to [get_regs {reg0_s0}] 6
set_min_delay -to [get_regs {reg0_s0}] 0.5
```

5.4.3 set_multicycle_path

It is used to set multicycle path.

Syntax

```
set_multicycle_path
    [-from <from_list>]
    [-through <through_list>]
    [-to <to_list>]
    -setup | -hold
    -start | -end
    <multiple_value>
```

Parameter

Table 5-12 set_multicycle_path Parameters

-from <from_list>	Start point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. Example: Example 5.34
-through <through_list>	Through points; supports get_nets and get_pins. Example: Example 5.35
-to <to_list>	End point; supports get_clocks, get_ports, get_regs, get_pins, all_clocks, all_inputs, and all_outputs. At least one of -from, -through, or -to must be specified. Example: Example 5.36
-setup | -hold	-setup applies multi-cycle constraints to setup and recovery analyses. -hold applies multi-cycle constraints to hold and removal analyses. Example: Example 5.34 and Example 5.36
-start | -end	Specifies multi-cycle settings relative to the launch clock or the latch clock. For setup analysis: Specifying -start shifts the launch clock to the left by the specified number of cycles. Specifying -end shifts the latch clock to the right by the specified number of cycles. For hold analysis: Specifying -start shifts the launch clock to the right by the specified number of cycles. Specifying -end shifts the latch clock to the left by the specified number of cycles. Example: Example 5.34 and Example 5.36
<multiple_value>	Integer values

Examples

Example 5.34

//For all timing paths launched by clock user_clk, apply 3-cycle setup/recovery analysis relative to the latch clock, and 2-cycle hold/removal analysis relative to the launch clock.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_multicycle_path -from [get_clocks {user_clk}] -setup -end 3
set_multicycle_path -from [get_clocks {user_clk}] -hold -start 2
```

Example 5.35

//Apply 5-cycle setup/recovery analysis to all paths passing through the Q output of flip-flop reg0.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
set_multicycle_path -through [get_pins {reg0_s0/Q}] 5 -setup
```

Example 5.36

// For timing paths from launch clock user_clk0 to latch clock user_clk1, set the multicycle path values to 2 and 3 respectively. which are valid for setup and recovery analyses. Gowin Software sets multicycle path value to 2.

```
create_clock -name user_clk0 -period 10 [get_ports {clk0}]
create_clock -name user_clk1 -period 20 [get_ports {clk1}]
set_multicycle_path -from [get_clocks { user_clk0}]   -setup -end 2
set_multicycle_path -to   [get_clocks { user_clk1}]   -setup -end 3
```

5.5 Report Constraints

5.5.1 report_timing

It is used to report timing.

Syntax

report_timing

-setup | -hold | -recovery | -removal

[-from_clock <*clock_name*> | -rise_from_clock <*clock_name*> | -fall_from_clock <*clock_name*>]

[-to_clock <*clock_name*> | -rise_to <*clock_name*> | -fall_to <*clock_name*>]

[-from <*from_list*> | -rise_from <*rise_from_list*> | -fall_from <*fall_from_list*>]

[-through <*through_list*>]

[-to <*to_list*> | -rise_to <*rise_to_list*> | -fall_to <*fall_to_list*>]

[-max_paths <*value*>]

[-max_common_paths <*value*>]

[-max_logic_level <*value*>]

[-min_logic_level <*value*>]

[-mod_ins {mod_ins1 mod_ins2 ...}]

Parameters

Table 5-13 report_timing Parameters

-setup | -hold | -recovery | -removal	-setup, -hold, -recovery, or -removal are respectively valid for setup, hold, recovery, or removal analysis. Example: Example 5.37, Example 5.38, Example 5.39, Example 5.40
-from_clock <clock_name>	Launch clock; supports get_clocks. Example: Example 5.37
-rise_from_clock <clock_name>	Launch clock with rising edge active; supports get_clocks.
-fall_from_clock <clock_name>	Launch clock with falling edge active; supports get_clocks.
-to_clock <clock_name>	latch clock; supports get_clocks. Example: Example 5.37
-rise_to_clock <clock_name>	Latch clock with rising edge active; supports get_clocks.
-fall_to_clock <clock_name>	Latch clock with falling edge active; supports get_clocks.
-from <from_list>	Start points; supports get_clocks, get_ports, get_regs, and get_pins. Example 5.38
-rise_from <rise_from_list>	Specifies the launch clock active on the rising edge for the object; supports get_clocks.
-fall_from <fall_from_list>	Specifies the launch clock active on the falling edge for the object; supports get_clocks.
-through <through_list>	Through points; supports get_nets and get_pins. Example 5.39
-to <to_list>	End points; supports get_clocks, get_ports, get_regs, and get_pins. Example 5.38
-rise_to <rise_to_list>	Specifies the latch clock active on the rising edge for the object; supports get_clocks and all_clocks.
-fall_to <fall_to_list>	Specifies the latch clock active on the falling edge for the object; supports get_clocks and all_clocks.

-max_paths <value>	Maximum number of timing paths to be reported. It must be a positive integer. Default is 25 if not specified. Example 5.38
-max_common_paths <value>	Maximum number of common paths to be reported. It must be a positive integer. The default is 1. A path is considered common only if both the clock and the object are the same. Example 5.38
-max_logic_level <value>	Reports timing paths with logic levels less than the specified value. It must be a positive integer. Example 5.37
-min_logic_level <value>	Reports timing paths with logic levels greater than the specified value. It must be a positive integer. Example 5.39
-mod_ins {mod_ins1 mod_ins2 …}	Reports timing paths related to specified sub-module instances in the design. Example 5.40

Examples

Example 5.37

//For the setup, constrain the report to timing paths, where both the launch and latch clocks are user_clk, and the logic level is less than 4.

```
create_clock -name user_clk -period 10 [get_ports {clk}]
report_timing -setup -from_clock [get_clocks {user_clk}] -to_clock [get_clocks {user_clk}] -max_logic_level 4
```

Example 5.38

//For the hold, constrain the report to timing paths between flip-flops reg0 and reg1, with a maximum of 50 paths, and up to 5 common paths are reported.

```
report_timing -hold -from [get_regs {reg0_s0}] -to [get_regs {reg1_s0}] -max_paths 50 -max_common_paths 5
```

Example 5.39

//For the recovery, constrain the report to timing paths that pass through the D pin of flip-flop reg0, with a logic level greater than 3.

```
report_timing -recovery -through [get_pins {reg0_s0/D}]
```

-min_logic_level 3

Example 5.40

//For the removal, constrain the report to timing paths related to the module instance sub_inst in the design.

report_timing -removal -mod_ins {sub_inst}

5.5.2 report_high_fanout_nets

It is used to report high fanout nets.

Syntax

report_high_fanout_nets

-max_nets <*max_nets_value*>

[-min_fanout <*min_fanout_value*>]

[-max_fanout <*max_fanout_value*>]

[-clock_regions]

[-slr]

[-ascending]

Parameters

Table 5-14 report_high_fanout_nets Parameters

-max_nets <max_nets_value>	Maximum number of nets to report. It must be a positive integer. Default is 10.
-min_fanout <min_fanout_value>	Report nets with fanout greater than or equal to the specified value. It must be a positive integer and be less than or equal to the value of -max_fanout. Example: Example 5.41
-max_fanout <max_fanout_value>	Report nets with fanout less than or equal to the specified value. It must be a positive integer, and be greater than or equal to the value of -min_fanout. Example: Example 5.42
-clock_regions	Report nets connected to clock pins. Example: Example 5.42
-slr	Report nets connected to LSR ports (LSR: Clear, Set, Reset, and Preset). Example: Example 5.43

-ascending	Sort the report in ascending order of fanout. If omitted, the report is sorted in descending order by default. Example: Example 5.43

Examples

Example 5.41

```
//Report up to 25 nets with fanout greater than or equal to 2.
report_high_fanout_nets -max_nets 25 -min_fanout 2
```

Example 5.42

```
// Report nets related to the clk signal with fanout not greater than 3.
report_high_fanout_nets -max_fanout 3 -clock_regions
```

Example 5.43

```
//Report nets related to set and reset, in ascending fanout order.
report_high_fanout_nets -slr -ascending
```

5.5.3 report_route_congestion

It is used to report route congestion.

Syntax

report_route_congestion

-max_grids <*max_grids_value*>

[-min_route_congestion <*min_route_congestion_value*>]

[-max_route_congestion <*max_route_congestion_value*>]

[-LOC <*grid_position*>]

Parameters

Table 5-15 report_route_congestion Parameters

-max_grids <max_grids_value>	Maximum number of grids to be reported. It must be a positive integer. Default is 10. Example: Example 5.44
-min_route_congestion <min_ route_congestion _value>	Report grids with routing congestion not less than the specified value. It must be a real number in the range [0, 1], less than or equal to -max_route_congestion. Example: Example 5.45

-max_route_congestion <max_ route_congestion _value>	Report grids with routing congestion not greater than the specified value. It must be a real number in the range [0, 1], and greater than or equal to -min_route_congestion. Example: Example 5.45
-LOC <grid_position>	Print the grid within the range of the constraint location. Example: Example 5.44

Examples

Example 5.44

//Report up to 30 grids located between rows 3 to 5 and columns 4 to 6.

report_route_congestion -max_grids 30 -LOC R[3:5]C[4:6]

Example 5.45

//Report up to 30 grids with congestion between 0.5 and 1.0.

report_route_congestion -max_grids 30 -min_route_congestion 0.5 -max_route_congestion 1

5.5.4 report_min_pulse_width

It is used to report minimum pulse width.

Syntax

report_min_pulse_width

-nworst <nworst_value>

[-min_pulse_width <*min_pulse_width_value*>]

[-max_pulse_width <*max_pulse_width_value*>]

[-detail]

<*objects*>

Parameters

Table 5-16 report_min_pulse_width Parameters

-nworst <nworst_value>	Print maximum number of worst MPW flip-flops. It must be a positive integer. Default is 10. Example: Example 5.46, Example 5.47, and Example 5.48

-min_pulse_width <min_pulse_width_value>	Print flip-flops with actual MPW greater than or equal to the specified value. It must be a positive real number, and less than or equal to -max_pulse_width. Example: Example 5.46 and Example 5.47
-max_pulse_width <max_pulse_width_value>	Print flip-flops with actual MPW less than or equal to the specified value. It must be a positive real number, and greater than or equal to -min_pulse_width. Example: Example 5.47
-detail	Print detailed information of MPW for FFs, including clock path analysis in addition to the summary table. If omitted, only the summary table will be printed. Example: Example 5.46
<objects>	Specify to print the MPW for specified FFs only; Supports get_regs. Example: Example 5.48

Examples

Example 5.46

//Report up to 30 flip-flops with MPW⩾2.5, including clock path analysis.

```
report_min_pulse_width -nworst 30 -min_pulse_width 2.5 -detail
```

Example 5.47

//Report up to 15 flip-flops with MPW between 2 and 4.

```
report_min_pulse_width -nworst 15 -min_pulse_width 2
-max_pulse_width 4
```

Example 5.48

//Specify to print the MPW for only reg0 and reg1, with a maximum of 10 entries.

```
report_min_pulse_width -nworst 10 [get_regs {reg0_s0 reg1_s0}]
```

5.5.5 report_max_frequency

It is used to report maximum operating frequency of submodules.

It prints the maximum operating frequency of synchronous clocks associated with submodules in the timing report. The maximum frequency is calculated based on the worst-case synchronous timing path whose endpoint is inside the submodule.

Syntax

report_max_frequency

-mod_ins {mod_ins0 mod_ins1 ...}

Parameters

Table 5-17 report_max_frequency Parameters

-mod_ins {mod_ins0 mod_ins1 …}	Specifies the maximum operating frequency of clocks associated with submodules in the design. Example: Example 5.49

Example

Example 5.49

//Report the maximum operating frequency of synchronous clocks in the submodule mod_inst0.

report_max_frequency -mod_ins {mod_inst0}

5.5.6 report_exceptions

It is used to report exception constraints.

The syntax options are similar to those of report_timing.

Syntax

report_exceptions

-setup | -hold | -recovery | -removal

[-from_clock <*clock_name*> | -rise_from_clock <*clock_name*> | -fall_from_clock <*name*>]

[-to_clock <*clock_name*> | -rise_to <*clock_name*> | -fall_to <*clock_name*>]

[-from <*from_list*> | -rise_from <*rise_from_list*> | -fall_from <*fall_from_list*>]

[-through <*through_list*>]

[-to <*to_list*> | -rise_to <*rise_to_list*> | -fall_to <*fall_to_list*>]

[-max_paths <*value*>]

[-max_common_paths <*value*>]

[-max_logic_level <*value*>]

[-min_logic_level <*value*>]

Parameters

Table 5-18 report_exceptions Parameters

-setup | -hold | -recovery | -removal	-setup, -hold, -recovery, or -removal are respectively valid for setup, hold, recovery, or removal analysis
-from_clock <name>	Launch clock; supports get_clocks.
-rise_from_clock <name>	Launch clock, valid on the rising edge; supports get_clocks.
-fall_from_clock <name>	Launch clock, valid on the falling edge; supports get_clocks.
-to_clock <name>	Latch clock; supports get_clocks.
-rise_to_clock <name>	Latch clock, valid on the rising edge; supports get_clocks.
-fall_to_clock <name>	Latch clock, valid on the falling edge; supports get_clocks.
-from <from_list>	Start points; supports get_clocks, get_ports, get_regs, and get_pins.
-rise_from <rise_from_list>	The object's launch clock with rising edge valid; supports get_clocks.
-fall_from <fall_from_list>	The object's launch clock with falling edge valid; supports get_clocks.
-through <through_list>	Through points; supports get_nets and get_pins.
-to <to_list>	End points; supports get_clocks, get_ports, get_regs, and get_pins.
-rise_to <rise_to_list>	The object's latch clock with rising edge valid; supports get_clocks and all_clocks.
-fall_to <fall_to_list>	The object's latch clock with falling edge valid; supports get_clocks and all_clocks.

-max_paths <value>	Print maximum number of timing paths, a positive integer. Default is 25.
-max_common_paths <value>	Print maximum number of common paths, a positive integer. Default is 1. Only paths with the same clock and objects are considered common end point.
-max_logic_level <value>	Print paths with logic levels less than the specified value, a positive integer.
-min_logic_level <value>	Print paths with logic levels greater than the specified value, a positive integer.

5.6 Operating Conditions Constraints

It is used to set the operating conditions.

Syntax

set_operating_conditions

-grade <*c* | *i* | *a*>

-model <*slow* | *fast*>

-speed <*speed_value*>

[-setup]

[-hold]

Parameters

Table 5-19 Operating Conditions Constraints Parameters

-grade <c | i | a>	Specifies the speed grade of the device, including c (commercial), i (industrial), and a (automotive). Example: Example 5.50, Example 5.51, and Example 5.52
-model <slow | fast>	Specifies the delay model used for timing analysis. Example: Example 5.50, Example 5.51, Example 5.52
-speed <speed_value>	Specifies the speed grade of the device. The value must be a positive integer supported by the device.
-setup	Specifies the delay model to be applied for setup and recovery analysis. If omitted, the default delay model is slow. Example: Example 5.50 and Example 5.52
-hold	Specifies the delay model to be applied for hold and removal analysis. If omitted, the default delay model is fast. Example: Example 5.51 and Example 5.52

Examples

Example 5.50

//Set the device to commercial grade with speed grade 7, using the fast delay model for setup and recovery analysis.

set_operating_conditions -grade c -model fast -speed 7 -setup

Example 5.51

//Set the device to industrial grade with speed grade 6, using the slow delay model for hold and removal analysis.

set_operating_conditions -grade i -model slow -speed 6 -hold

Example 5.52

//Set the device to automotive grade with speed grade 6, using the fast delay model for setup, hold, recovery, and removal analysis.

set_operating_conditions -grade a -model fast -speed 6 -setup -hold

5.7 Other Constraints

5.7.1 derive_clocks

It is used to create a derived clock frequency. The derive_clocks has higher priority than the frequency set in "Configuration > Global > Frequency (MHz)".

Syntax

derive_clocks

-freq <*freq_value*>

Parameter

Table 5-20 derive_clocks Parameter

-freq <*freq_value*>	Specifies the derived clock frequency in MHz. The value must be a real number within the range (0, 1200]. Default is 100. Example: Example 5.53

Example

Example 5.53

//Create a derived clock with frequency of 90 MHz.

derive_clocks -freq 90

GOWIN

PROGRAMMING FOR THE FUTURE