# Gowin Software Tcl Commands

# User Guide

**SUG1220-1.0E, 08/29/2025**

## 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 |
|---|---|---|
| 08/29/2025 | 1.0E | Initial version published. |

# Contents

Contents ........ i

List of Tables ........ iii

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 ........ 1

2 Tcl Overview ........ 2

2.1 Introduction ........ 2

2.2 Launching Command Line Mode ........ 2

3 Tcl Commands ........ 3

3.1 Command ........ 3

3.1.1 Command Type ........ 3

3.1.2 Command List ........ 3

3.2 Command Description ........ 4

3.2.1 add_file ........ 4

3.2.2 create_ipc ........ 5

3.2.3 create_project ........ 6

3.2.4 generate_target ........ 7

3.2.5 get_ips ........ 7

3.2.6 import_files ........ 8

3.2.7 list_property ........ 9

3.2.8 open_project ........ 10

3.2.9 read_ipc ........ 11

3.2.10 report_property ........ 11

3.2.11 rm_file ........ 12

3.2.12 run ........ 12

3.2.13 run close ........ 13

3.2.14 saveto ........ 13

3.2.15 set_device ........ 14

3.2.16 set_file_enable ........ 15
3.2.17 set_file_prop ........ 15
3.2.18 set_csr ........ 16
3.2.19 set_option ........ 17
3.2.20 set_property ........ 57
3.2.21 source ........ 58
3.2.22 write_ip_tcl ........ 59

# List of Tables

Table 1-1 Terminology and Abbreviations ............................................................................................ 1

# 1 About This Guide

## 1.1 Purpose

This manual provides a detailed description of the command-line mode and related Tcl command usage for Gowin Software.

## 1.2 Related Documents

The latest user guides are available on GOWINSEMI Website. You can find the related documents at www.gowinsemi.com: SUG100, Gowin Software User Guide.

## 1.3 Terminology and Abbreviations

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

**Table 1-1 Terminology and Abbreviations**

| Terminology and Abbreviations | Meaning |
|---|---|
| CRC | Cyclic Redundancy Check |
| FPGA | Field Programmable Gate Array |
| GowinSynthesis | GowinSynthesis |
| GPA | Gowin Power Analyzer |
| IP Core | Intellectual Property Core |
| PnR | Place & Route |
| SEU Handler | Single-Event Upsets Handler |
| Tcl | Tool Command Language |

## 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 by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

# 2 Tcl Overview

## 2.1 Introduction

Tool Command Language (Tcl) is a scripting language integrated into Gowin Software. Gowin Software supports command-line mode operation, allowing users to interact with the tool and manage projects through Tcl commands.

## 2.2 Launching Command Line Mode

Gowin Software supports operation in command-line mode, which can be accessed via gw_sh.exe.

### Syntax

Command: (Windows example)

Launch gw_sh.exe from the installation directory:

\x.x\IDE\bin\gw_sh.exe.

Parameter: [script file]

- If no parameter is provided: Enters the command line console mode directly.
- script file: Executes a specified script file, optional.

### Application Example

# Launch Command-line mode

gw_sh.exe

#Execute a script file; for the examples, you can see the section Tcl Quick Start in *SUG918, Gowin Software Quick Start Guide*.

gw_sh.exe script_file

# 3 Tcl Commands

Gowin Software supports command line mode. In the following description, the content in <> must be specified, and the content in [] is optional. When creating file names, special characters such as /, *, - and space are not supported. Additionally, when specifying an IP module_name, names starting with a digit are prohibited.

## 3.1 Command

### 3.1.1 Command Type

IPFlow

Project

### 3.1.2 Command List

**IPFlow**

create_ipc

generate_target

get_ips

list_property

read_ipc

report_property

set_property

source

write_ip_tcl

**Project**

add_file

create_project

import_files

open_project

rm_file

run

run close

saveto

set_device

set_file_enable

set_file_prop

set_option

source

# 3.2 Command Description

## 3.2.1 add_file

Adds design files to the project.

### Syntax

```
add_file [-type] [-disable] [-h/--help] <file>
```

### Usage

| Name | Description |
|---|---|
| [-type] | Specifies the type of design file to add |
| [-disable] | Disables the file |
| [-h/--help] | Displays help information |
| <file> | File(s) to add |

### Type

Project

### Description

When adding design files, two types of file path separators / or \\ are supported on Windows and Linux systems. Both relative and absolute paths are supported. In Gowin Software, the relative path is relative to the current project directory, and in command line mode, it is relative to the path from which gw_sh is launched.

### Parameter

- <file>: The design file(s) to be added. Multiple files can be specified, separated by spaces.
- [-type <type>]: The add_file command will automatically determine the file type based on the file extension. Alternatively, this option can be used to explicitly specify the file type. Supported file types include verilog, vhdl, sv, vg, cst, sdc, gao, gpa, gsc, etc.
- [-disable]: Disables the added files. The disabled files are only added to the file list and are excluded from implementation processes. Related command: set_file_enable.

- [-h/--help]: Displays help information.

### Example

```
add_file abc.v
add_file -type vhdl 1.vhd 2.vhdl 3.vhd
add_file D:/gowin_project/top.v
add_file D:\\gowin_project\\top.v
```

## 3.2.2 create_ipc

Creates IPC file with default configuration.

### Syntax

```
create_ipc -name <ipName> -module_name <moduleName>
[-language <arg>] [-file_name <fileName>] [-dir <path>] [-force]
```

### Usage

| Name | Description |
|---|---|
| -name | IP name |
| -module_name | The module name of the IP to be created |
| [-language] | Language for the IP file, template file, and simulation file |
| [-file_name] | IP file name |
| [-dir] | The path to generate the IP |
| [-force] | Overwrites existing files |

### Type

IPFlow

### Description

This command is used to create an IPC file for the IP core in the IP Core Generator.

### Parameter

- -name <ipName>: Specifies the name of the IP, and the name can be obtained from the IP Core Generator.
- -module_name <moduleName>: Specifies the module name of the IP to be created.
- [-language <arg>]: Specifies the language of the generated IP files, template files, and simulation files (Verilog/VHDL). If not specified, the default is Verilog.
- [-file_name <fileName>]: Specifies the name of the generated IP file. If not specified, it will be the same as the name specified by the module_name.
- [-dir <path>]: Specifies the directory path to generate the IP. If not specified, it will be generated into the current project's src folder.
- [-force]: Overwrite existing files.

### Example

In the following example, the IP core specified by -name is created in the current project's src folder, and the module name, language, and file name is specified.

create_ip -name fifo -module_name FIFO_Top -language Verilog -file_name fifo

### See also

generate_target

## 3.2.3 create_project

Creates a project.

### Syntax

create_project [-name <prjName>] [-dir <path>] [-pn <pnName>] [-device_version <arg>] [-force] [-h/--help]

### Usage

| Name | Description |
|---|---|
| [-name] | The name of the project to be created |
| [-dir] | The path to generate the project |
| [-pn] | The part number for the project to be created |
| [-device_version] | The device version for the project to be created |
| [-force] | Overwrites existing files |
| [-h/--help] | Displays help information |

### Type

Project

### Description

Creates a project. For the file path format, see 3.2.1 add_file.

### Parameter

- [-name <prjName>]: Specifies the name of the project to be created.
- [-dir <path>]: Specifies the path where the project will be created. If the specified path does not exist, a new path will be created. If this option is not specified, the default will be the current working path or the path where the TCL script is located.
- [-pn <pnName>]: Specifies the part number for the project to be created.
- [-device_version <arg>]: Specifies the device version for the part number. For devices with only an initial version, the value of the device version will be NA.
- [-force]: Overwrites existing projects.
- [-h/--help]: Displays help information.

**Example**

create_project -name prj0 -dir D:/tclprj -pn GW1N-UV4LQ144C6/15 -device_version B

create_project -name prjlest -pn GW1N-UV4LQ144C6/15 -

device_version B -force

## 3.2.4 generate_target

Generates target files for specified objects.

**Syntax**

generate_target <objects> [-force]

**Usage**

| Name | Description |
|---|---|
| <objects> | Specifies the object to generate the target file |
| [-force] | Overwrite existing files |

**Type**

IPFlow

**Description**

This command is used to generate target files for the specified IP objects (get_ips) and adds the IP core design files to the current project.

**Parameter**

- <objects>: Specifies one or more IP objects to generate design files. An object is specified using [get_ips module_name]. Multiple objects are specified using [get_ips module_name0 module_name1 …].
- [-force]: Overwrites existing files.

**Example**

In the following example, design files and template files for the specified IP are generated, and load them into the current project.

generate_target [get_ips FIFO_Top]

**See also**

- generate_target
- create_ipc
- read_ipc

## 3.2.5 get_ips

Specifies IP objects.

**Syntax**

get_ips <module_name>

### Usage

| Name | Description |
|---|---|
| <module_name> | Specifies the module name of the IP |

### Type

IPFlow

### Description

Specifies the IP objects in the current project.

### Parameter

<module_name>: Specifies the module name of the IP object, which can be one or multiple.

### Example

In the following example, a single IP object in the current project is specified.

```
get_ips FIFO_Top
```

In the following example, multiple IP objects in the current project are specified.

```
get_ips FIFO_Top FIFO_Top_1
```

### See also

- generate_target
- list_property
- report_property
- set_property

## 3.2.6 import_files

Copies files or directories to the current project.

### Syntax

import_files [-file <file>] [-dir <path>] [-fileList <fileList>] [-force] [-h/--help]

### Usage

| Name | Description |
|---|---|
| [-file] | Specifies the file(s) to copy |
| [-dir] | Specifies the path to copy from |
| [-fileList] | Specifies a file list. |
| [-force] | Overwrites existing files with the same name |
| [-h/--help] | Displays help information |

### Type

Project

### Description

Copy files or directories to the current project path /src. The values for -file, -dir, and -fileList can be absolute or relative paths. If a tcl script is executed via gwsh.exe, the relative path is based on the tcl script's path. If the import_files command is executed via the console window in Gowin Software, the relative path is based on the current working directory (pwd). If no options are specified after import_files, it defaults to copying all files specified by the add_file command to the project path /src.

### Parameter

- [-file <file>]: Adds one or more files to the project path /src.
- [-dir <path>]: Adds all files and subfolders in the path to the project path /src.
- [-fileList <fileList>]: Specifies a file list where each line represents a file to be added to the project. Examples are as follows:
  - D:/test1.v
  - D:/test2.v
  - This option allows adding all the files specified in each line of the file to the project path /src.
- [-force]: Overwrites files with the same name in the project path /src.
- [-h/--help]: Displays help information.

### Example

```
import_files -file D/test1 .v -force
import_files -file D:/test1 .v D:/test2.v -force
import_files -dir D:/sourceFile
import_files -fileList log, and the file contents are as follows:
  D:/Test1.v
  D:/Test2.v
```

## 3.2.7 list_property

Lists the properties of an object.

### Syntax

```
list_property <object>
```

### Usage

| Name | Description |
|---|---|
| <object> | The object that queries properties. |

### Type

IPFlow

### Description

Obtains a list of all options for the specified IP object.

### Parameter

<object>: Specifies the IP object that queries option. Use [get_ips module_name] to specify.

### Example

In the following example, all properties of the specified IP object are listed.

list_property [get_ips FIFO_Top]

### See also

- report_property
- set_property

## 3.2.8 open_project

Opens a project.

### Syntax

open_project <file>] [-pn] [-device_version] [-h/--help]

### Usage

| Name | Description |
|---|---|
| <file> | Specifies the project file |
| [-pn] | Specifies part number |
| [-device_version] | Specifies device version |
| [-h/--help] | Displays help information |

### Type

Project

### Description

This command is used to open a project, and allows specifying a new part number for the opened project. For the file path format, see 3.2.1 add_file.

- <file>: Specifies the name of the project file to be opened.
- [-pn]: Specifies the part number for the project.
- [-device_version]: Specifies the device version for the project.
- [-h/--help]: Displays help information.

### Example

```
open_project D:\test.gprj
```

### 3.2.9 read_ipc

Reads ipc files.

**Syntax**

read_ipc <file>

**Usage**

| Name | Description |
|---|---|
| <file> | IPC files |

**Type**

IPFlow

**Description**

Read the specified IPC file and configure options through the IPC file. For the file path format, see 3.2.1 add_file.

**Parameter**

<file>: Specifies the name of the IPC file to be opened.

**Example**

In the following example, an IPC file is read from the specified path.

```
read_ipc D:/gowin_project/src/fifo/fifio.ipc
```

**See also**

generate_target

### 3.2.10 report_property

Reports properties of an object.

**Syntax**

report_property <object>

**Usage**

| Name | Description |
|---|---|
| <object> | The object that queries properties. |

**Type**

IPFlow

**Description**

This command is used to obtain the option names, types, and values for the specified IP object.

**Parameter**

<object>: Specifies the IP object that queries option. Use [get_ips module_name] to specify.

### Example

In the following example, all the properties of a specified IP object are listed.

report_property [get_ips FIFO_Top]

### See also

- set_property
- list_property

## 3.2.11 rm_file

Removes design files.

### Syntax

rm_file [-h/--help ] <files>

### Usage

| Name | Description |
|---|---|
| <-files> | Design files to be removed |
| [-h/--help] | Displays help information |

### Type

Project

### Description

Removes design files; and for the file path format, see 3.2.1 add_file.

### Parameter

- <-files> Specifies the design files to be removed. Multiple files can be specified, separated by spaces.
- [-h/--help]: Displays help information.

### Example

rm_file a.v

rm_file a.v b.v c.v

rm_file D:/gowin_project/top.v

rm_file D:\\gowin_project\\top.v

## 3.2.12 run

Runs flows.

### Syntax

run [-h/--help] <syn/pnr/all>

### Usage

| Name | Description |
|---|---|
| <syn/pnr/all> | Specifies the name of the flow to be run |

| Name | Description |
|---|---|
| [-h/--help] | Displays help information |

**Type**

Project

**Description**

Runs a specified flow or all flows.

**Parameter**

- <syn/pnr/all>: Specifies the name of the flow to be run. The available flow names are "syn" (synthesis), "pnr" (place and route), or you can specify all to run all flows.
- [-h/--help]: Displays help information

**Example**

run pnr

run all

### 3.2.13 run close

Closes a project.

**Syntax**

run close

**Type**

Project

**Description**

Closes the current project.

**Example**

run close

### 3.2.14 saveto

Saves project data to a tcl script.

**Syntax**

saveto [-all_options] [-h/--help] <file>

**Usage**

| Name | Description |
|---|---|
| [-all_options] | Saves all option information |
| [-h/--help] | Displays help information |
| <file> | The file to be saved |

**Type**

Project

### Description

Saves the current project design data to a tcl script. For the file path format, see 3.2.1 add_file.

### Parameter

- [-all_options]: By default, the saveto command only saves the modified options, i.e., those different from the default values. By using -all_options, all option information will be saved.
- [-h/--help]: Displays help information.
- <file>: The file to be saved.

### Example

```
saveto project.tcl
saveto -all_options project.tcl
saveto -all_options D:/gowin_project/project.tcl
saveto -all_options D:\\gowin_project\\project.tcl
```

## 3.2.15 set_device

Sets device part number.

### Syntax

```
set_device [-device_version <value>] [-h/--help ] <part number>
```

### Usage

| Name | Description |
|---|---|
| [-device_version <value>] | Sets device version |
| [-h/--help] | Displays help information |
| <part number> | Sets device part number |

### Type

Project

### Description

Sets device part number.

### Parameter

- <part number>: Specifies the part number of the target device, such as GW1N-UV4LQ144C6/I5.
- [-device_version <value>]: Specifies the device version. Supported values are NA|B|C|D.
- [-h/--help]: Displays help information.

### Example

```
set_device GW1N-LV1CS30C6/I5
set_device –device_version C GW1N-UV4LQ144C6/I5
```

### 3.2.16 set_file_enable

Sets file enable properties.

**Syntax**

set_file_enable <file> <true|false> [-h/--help]

**Usage**

| Name | Description |
|---|---|
| <file> | Specifies the design file to be set |
| <true\|false> | Whether the file can be used or not |
| [-h/--help] | Displays help information |

**Type**

Project

**Description**

Sets whether the file can be used. For the file path format, see 3.2.1 add_file.

**Parameter**

- <file>: Specifies the file to be set.
- <true|false>: true means the file can be used, and false means it cannot.
- [-h/--help]: Displays help information.

**Example**

```
set_file_enable top.v false
set_file_enable D:/gowin_project/top.v
set_file_enable D:\\gowin_project\\top.v
```

### 3.2.17 set_file_prop

Sets file properties.

**Syntax**

set_file_prop <file> [-lib <name>] [-h/--help]

**Usage**

| Name | Description |
|---|---|
| <file> | Specifies the design files to be set |
| [-lib <name>] | Sets the library name for the file |
| [-h/--help] | Displays help information |

**Type**

Project

### Description

Sets properties for the specified files. For the file path format, see 3.2.1 add_file.

### Parameter

- <file>: Specifies the file(s) to be set. Multiple files can be listed, separated by spaces.
- [-lib <name>]: Sets the library name for the file. This option is only valid for VHDL files.
- [-h/--help]: Displays help information.

### Example

```
set_file_prop -lib work top .vhd
set_file_prop -lib work D:/gowin_project/top.vhd
set_file_prop -lib work D:\\gowin_project\\top.vhd
```

## 3.2.18 set_csr

Specifies the CSR file.

### Syntax

```
set_csr [-h/--help ] <file>
```

### Usage

| Name | Description |
|---|---|
| <file> | The CSR file to be specified. |
| [-h/--help] | Displays help information |

### Type

Project

### Description

Specifies the CSR file. For the file path format, see 3.2.1 add_file.

### Parameter

- <files>: Specifies the CSR file.
- [-h/--help]: Displays help information.

### Example

```
set_csr a.csr
set_csr D:/gowin_project/a.csr
set_csr D:\\gowin_project\\a.csr
```

### 3.2.19 set_option

Sets project-related property configurations and flow options.

**Syntax**

set_option [options] [-h/--help]

**Usage**

| Name | Description |
|---|---|
| [options] | Specifies configurations or flow options |
| [-h/--help] | Displays help information |

**Type**

Project

**Description**

Sets project-related property configurations and flow options.

**Parameter**

- [options]: Specifies configurations or flow options.
- [-h/--help]: Displays help information.

**Global Property Configuration**

**-output_base_name**

Specify the output file name.

**Syntax**

-output_base_name <name>

**Usage**

| Name | Description |
|---|---|
| <name> | Specifies the output file name |

**Type**

Project

**Description**

Specifies the name of the output file. This option is only used to set the base name, and different flows will append appropriate extensions based on the output file type. For example, if -output_base_name abc is set, the netlist file generated by Gowin Synthesis will be named abc.vg.

**Parameter**

<name>: Specifies the output file name.

**Example**

set_option -output_base_name abc

**-global_freq**

Specify frequency value

**Syntax**

-global_freq <default|value>

**Usage**

| Name | Description |
|---|---|
| <default\|value> | Specifies frequency value |

**Type**

Project

**Description**

Specifies the frequency value. The default setting is: 50 MHz for the LittleBee family, 100 MHz for Arora family.

**Parameter**

<default|value>frequency value

**Example**

set_option -global_freq 80

### Synthesis Property Configuration

#### -synthesis_tool

Specifies Synthesis tool.

**Syntax**

-synthesis_tool <tool>

**Usage**

| Name | Description |
|---|---|
| <tool> | Specifies Synthesis tool |

**Type**

Project

**Description**

Specifies Synthesis tool GowinSynthesis.

**Parameter**

<tool>: Specifies Synthesis tool GowinSynthesis.

**Example**

set_option -synthesis_tool GowinSynthesis

#### -top_module

Specifies Top Module/Entity.

**Syntax**

-top_module <name>

**Usage**

| Name | Description |
|---|---|
| <name> | Specifies top module |

**Type**

Project

**Description**

Specifies top module.

**Parameter**

<name>: Specifies top module.

**Example**

set_option -top_module test

### -include_path

Specifies Include Path.

**Syntax**

-include_path <path or path list>

**Usage**

| Name | Description |
|---|---|
| <path or path list> | Specifies Include Path |

**Type**

Project

**Description**

Specifies the include path. When specifying multiple include paths, separate them with semicolons (;) and enclose all paths within curly braces {}. For example: -include_path {/path1;/path2;/path3}. Both relative and absolute paths are supported. A relative path is interpreted relative to the program's current working directory.

**Parameter**

<path or path list>: Specifies Include Path.

**Example**

set_option -include_path D:/project

### -verilog_std

Specifies Verilog language standard.

**Syntax**

-verilog_std<v1995|v2001|sysv2017>

**Usage**

| Name | Description |
|---|---|
| <v1995\|v2001\|sysv2017> | Specifies Verilog language standard |

### Type

Project

### Description

Specifies Verilog language standard: Verilog 95/Verilog 2001/System Verilog 2017, and the default is Verilog 2001.

### Parameter

<v1995|v2001|sysv2017>: Specifies Verilog language standard.

### Example

set_option -verilog_std v1995

## -vhdl_std

Specifies VHDL language standard.

### Syntax

-vhdl_std <vhd1993|vhd2008|vhd2019>

### Usage

| Name | Description |
|---|---|
| < vhd1993\|vhd2008\|vhd2019> | Specifies VHDL language standard |

### Type

Project

### Description

Specifies VHDL language standard: VHDL 1993/VHDL 2008/VHDL 2019, and the default is VHDL1993.

### Parameter

<vhd1993|vhd2008|vhd2019>: Specifies VHDL language standard.

### Example

set_option -vhdl_std vhd2008

## -print_all_synthesis_warning <0|1>

Specifies whether to print all synthesis warning messages. The default is 0.

### Syntax

-print_all_synthesis_warning <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not print all warning messages.<br>1: Print all warning messages. |

### Type

Project

**Description**

Specifies whether to print all synthesis warning messages. The default is 0.

**Parameter**

<0|1>: Specifies whether to print all synthesis warning messages.

**Example**

set_option -print_all_synthesis_warning 1

### -disable_io_insertion

Enables or disables I/O insertion.

**Syntax**

-disable_io_insertion <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Enables I/O insertion<br>1: Disables I/O insertion |

**Type**

Project

**Description**

Enables or disables I/O insertion. The default is 0.

**Parameter**

<0|1>: Enables or disables I/O insertion.

**Example**

set_option -disable_io_insertion 1

### -looplimit <value>

The loop limit value of default compiler in RTL.

**Syntax**

-looplimit <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Looplimit value |

**Type**

Project

**Description**

The loop limit value of default compiler in RTL, with the default value being 2000.

**Parameter**

<value>: The loop limit value of default compiler in RTL.

**Example**

set_option -looplimit 1000

**-maxfan <value>**

Set fanout value.

**Syntax**

-maxfan <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Maxfan value |

**Type**

Project

**Description**

Set the fanout value for input ports, nets, or register output ports, with the default value being 10000.

**Parameter**

<value>: The fanout value for input ports, nets, or register output ports.

**Example**

set_option -maxfan 5000

**-rw_check_on_ram**

Insert bypass logic around RAM.

**Syntax**

-rw_check_on_ram <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disable<br>1: Enable |

**Type**

Project

**Description**

If there are read or write conflicts in RAM, enabling this option will insert bypass logic around the RAM to prevent simulation mismatches, with the default being 0.

**Parameter**

<0|1>: Enables or disables the insertion of bypass logic around RAM.

**Example**

set_option -rw_check_on_ram 1

### Place & Route Property Configuration

#### -vccx

Specifies vccx value.

**Syntax**

-vccx <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Specifies vccx value |

**Type**

Project

**Description**

Specifies vccx value.

**Parameter**

<value>: Specifies vccx value.

**Example**

set_option -vccx 3.3

#### -vcc

**Syntax**

-vcc <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Specifies vcc value |

**Type**

Project

**Description**

Specifies vccx value.

**Parameter**

<value>: Specifies vccx value.

**Example**

set_option -vcc 3.3

#### -gen_sdf

Whether to generate the SDF file.

**Syntax**

-gen_sdf <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate the SDF file.<br>1: Generates the SDF file. |

### Type

Project

### Description

Specifies whether the Place & Route generates an SDF, and the default is 0.

### Parameter

<0|1>: Enable control for generating the SDF file.

### Example

```
set_option -gen_sdf 1
```

## -gen_io_cst

Whether to generate the physical constraint file for the port, named *.io.cst.

### Syntax

```
-gen_io_cst <0|1>
```

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Does not generate the *.io.cst file.<br>1: Generates the *.io.cst file. |

### Type

Project

### Description

Specifies whether the Place & Route generates the physical constraint file for the port, named *.io.cst. The default is 0.

### Parameter

<0|1>: Generates physical constraints file enable control for port.

### Example

```
set_option -gen_io_cst 1
```

## -gen_ibis

Specifies whether to generate the input/output buffer information file named *.ibs.

### Syntax

```
-gen_ibis <0|1>
```

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate *.ibs file.<br>1: Generates *.ibs file. |

**Type**

Project

**Description**

Specifies whether Place & Route generates the input/output buffer information file named *.ibs.

**Parameter**

<0|1>: Generates the input/output buffer information file enable control.

**Example**

set_option -gen_ibis 1

**-gen_posp**

Whether to generate a device placement file.

**Syntax**

-gen_posp <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate *.posp file.<br>1: Generates *.posp file. |

**Type**

Project

**Description**

Specifies whether Place & Route generates a device placement file named *.posp, which contains only BSRAM placement information. The default value is 0.

**Parameter**

<0|1>: Generates device placement file enable control.

**Example**

set_option -gen_posp 1

**-gen_text_timing_rpt**

Generate a timing report in text format.

**Syntax**

-gen_text_timing_rpt <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate *.tr file. |

| Name | Description |
|---|---|
|  | 1: Generates *.tr file. |

**Type**

Project

**Description**

Specifies whether Place & Route generates a timing report in text format with the filename *.tr. The default value is 0.

**Parameter**

<0|1>: Generates a text-format timing report enable control.

**Example**

set_option -gen_text_timing_rpt 1

### -gen_verilog_sim_netlist

Specifies whether to generate a Verilog timing simulation model file.

**Syntax**

-gen_verilog_sim_netlist <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate *.vo file.<br>1: Generates *.vo file. |

**Type**

Project

**Description**

Specifies whether Place & Route generates a Verilog timing simulation model file named *.vo, and the default is 0.

**Parameter**

<0|1>: Generates enable control for the Verilog timing simulation model file.

**Example**

set_option -gen_verilog_sim_netlist 1

### -gen_vhdl_sim_netlist

Generates VHDL timing simulation model file or not.

**Syntax**

-gen_vhdl_sim_netlist <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not generate *.vho file.<br>1: Generates *.vho file. |

**Type**

Project

**Description**

Specifies whether Place & Route generates a VHDL timing simulation model file named *.vho, and the default is 0.

**Parameter**

<0|1>: Generates enable control for the VHDL timing simulation model file.

**Example**

set_option -gen_vhdl_sim_netlist 1

**-show_init_in_vo**

Adds default initial values to the instances in the timing simulation model file.

**Syntax**

-show_init_in_vo <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not add default initial values to the instances in the timing simulation model file.<br>1: Adds default initial values to the instances in the timing simulation model file. |

**Type**

Project

**Description**

Adds default initial values to the instances in the generated Place & Route timing simulation model file, and the default is 0.

**Parameter**

<0|1>: Enable control for adding default initial values to the instances in the generated Place & Route timing simulation model file.

**Example**

set_option -show_init_in_vo 1

**-show_all_warn**

Whether to show all warnings.

**Syntax**

-show_all_warn<0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not output all warning messages during |

| Name | Description |
|---|---|
| | Placement & Routing.<br>1: Outputs all warning messages during Placment & Routing. |

### Type

Project

### Description

Outputs all warning messages when running Place & Route, and the default is 0.

### Parameter

<0|1>: Enable control for outputting all warning messages when running Place & Route.

### Example

set_option -show_all_warn 1

## -timing_driven

Whether optimizes timing driven for Place & Route.

### Syntax

-timing_driven <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Disables Place & Route timing driven.<br>1: Enables Place & Route timing driven. |

### Type

Project

### Description

Timing-driven optimization is performed when running Place & Route. The default value is 1.

### Parameter

<0|1>: Enable control for timing-driven optimization when running Place & Route.

### Example

set_option -timing_driven 1

## -cst_warn_to_error

Elevates physical constraint warnings to errors.

### Syntax

-cst_warn_to_error <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Physical constraint warnings will not be elevated to errors.<br>1: Physical constraint warnings will be elevated to errors. |

**Type**

Project

**Description**

Elevates physical constraint warnings to errors when running Place & Route. The default value is 1.

**Parameter**

<0|1>: Enable control for elevating physical constraint warnings to errors when running Place & Route.

**Example**

set_option -cst_warn_to_error 1

### -rpt_auto_place_io_info

Reports auto-placed IO location information.

**Syntax**

-rpt_auto_place_io_info <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Auto-placed IO location information will not be reported.<br>1: Auto-placed IO location information will be reported. |

**Type**

Project

**Description**

Reports auto-placed IO location information when running Place & Route. The default value is 0.

**Parameter**

<0|1>: Enable control for reporting auto-plzaced IO location information.

**Example**

set_option -cst_warn_to_error 1

### -place_option

Placement algorithm option.

**Syntax**

-place_option <0|1|2|3|4>

**Usage**

| Name | Description |
|---|---|
| <0\|1\|2\|3\|4> | 0: Uses the default placement algorithm.<br>1: Uses placement algorithm 1.<br>2: Uses placement algorithm 2.<br>3: Uses placement algorithm 3.<br>4: Uses placement algorithm 4. |

**Type**

Project

**Description**

Placement algorithm option. The default value is 0.

**Parameter**

<0|1|2|3|4>: Placement algorithm option。

**Example**

set_option -place_option 1

**-route_option**

Routing algorithm option.

**Syntax**

-route_option <0|1|2>

**Usage**

| Name | Description |
|---|---|
| <0\|1\|2> | 0: Uses the default routing algorithm.<br>1: Uses the default routing algorithm 1.<br>2: Uses the default routing algorithm 2. |

**Type**

Project

**Description**

Routing algorithm option. The default value is 0.

**Parameter**

<0|1|2>: Routing algorithm option.

**Example**

set_option -route_option 1

**-ireg_in_iob**

Place registers connected to input buffers to IOB.

**Syntax**

-ireg_in_iob <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not place registers connected to input buffers to IOB.<br>1: Place registers connected to input buffers to IOB. |

### Type

Project

### Description

Enabling this option will cause Place & Route to place registers connected to input buffers to IOB. The default value is 1.

### Parameter

<0|1>: Enable control for placing registers connected to input buffers to IOB.

### Example

set_option -ireg_in_iob 1

## -oreg_in_iob

Place registers connected to output/tristate buffers to IOB.

### Syntax

-oreg_in_iob <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not place registers connected to output/tristate buffers to IOB.<br>1: Place registers connected to output/tristate buffers to IOB. |

### Type

Project

### Description

Enabling this option will cause Place & Route to place registers connected to output/tristate buffers to IOB. The default value is 1.

### Parameter

<0|1>: Enable control for place registers connected to output/tristate buffers to IOB.

### Example

set_option -oreg_in_iob 1

## -ioreg_in_iob

Place registers connected to bidirectional buffers to IOB.

### Syntax

-ioreg_in_iob <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not place registers connected to bidirectional buffers to IOB.<br>1: Place registers connected to bidirectional buffers to IOB. |

### Type

Project

### Description

Enabling this option will cause Place & Route to place registers connected to bidirectional buffers in IOB. The default value is 1.

### Parameter

<0|1>: Enable control for placing registers connected to bidirectional buffers to IOB.

### Example

set_option -ioreg_in_iob 1

## -replicate_resources

Replicate resources for high fanout to reduce fanout and improve timing.

### Syntax

-replicate_resources <0|1>

### Usage

| Name | Description |
|---|---|
| <0\|1> | 0: Do not replicate resources for high fanout to reduce fanout.<br>1: Replicate resources for high fanout to reduce fanout. |

### Type

Project

### Description

Enabling this option will cause Place & Route to replicate resources for high fanout to reduce fanout and improve timing results. The default value is 0.

### Parameter

<0|1>: Enable control for replicate resources for high fanout to reduce fanout.

### Example

set_option -replicate_resources 1

## -clock_route_order

Specifies the routing order for clock wires other than those generated by Clock primitives. The optional values are 0 and 1, with the default value being 0.

**Syntax**

-clock_route_order <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Route clock wires in the order of net fanout, from highest to lowest.<br>1: Route clock wires in the order of frequency, from highest to lowest. |

**Type**

Project

**Description**

Specifies the routing order for clock wires other than those generated by clock primitives.

**Parameter**

<0|1>: Specifies the routing order for clock wires other than those generated by clock primitives.

**Example**

set_option -clock_route_order 1

### -route_maxfan

Sets maximum fanout for routing.

**Syntax**

-route_maxfan <value>

**Usage**

| Name | Description |
|---|---|
| < value> | Sets maximum fanout for routing. |

**Type**

Project

**Description**

Based on routing optimization, this option is used to set the maximum fanout for routing. The value should be an integer greater than 0 and less than or equal to 100. For GW1NZ-1/GW1N-2/GW1NR-2/GW1N-1P5 devices, the default value of this option is 10, while for other devices, the default value is 23.

**Parameter**

<value>: Sets maximum fanout for routing.

**Example**

set_option -route_maxfan 60

### -correct_hold_violation

Automatic hold violation fixing for routing.

**Syntax**

-correct_hold_violation <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not automatically fix hold violations during routing.<br>1: Automatically fix hold violations during routing. |

**Type**

Project

**Description**

Enabling this option allows automatically fixing hold timing violations during routing. The default value is 1.

**Parameter**

<0|1>: Enable control for automatically fixing hold timing violations during routing

**Example**

set_option -correct_hold_violation 1

### -convert_sdp32_36_to_sdp16_18

Converts 32/36-bit wide SDPB/SDPX9B into two 16/18-bit wide SDPB/SDPX9B. This command is only supported by Arora V devices.

**Syntax**

-convert_sdp32_36_to_sdp16_18 <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables conversion<br>1: Converts 32/36-bit wide SDPB/SDPX9B into two 16/18-bit wide SDPB/SDPX9B |

**Type**

Project

**Description**

Enabling this option will convert 32/36-bit wide SDPB/SDPX9B into two 16/18-bit wide SDPB/SDPX9B. The default value is 0.

**Parameter**

<0|1> Enable control for converting 32/36-bit wide SDPB/SDPX9B into two 16/18-bit wide SDPB/SDPX9B.

**Example**

set_option -convert_sdp32_36_to_sdp16_18 1

**-inc_place <0|auto|file>**

Incremental placement.

**Syntax**

-inc_place <0|auto|file>

**Usage**

| Name | Description |
|---|---|
| <0\|auto\|file > | 0: Disables incremental placement.<br>auto: Automatic incremental placement.<br>file: Specifies a *.p file for incremental placement. |

**Type**

Project

**Description**

Enabling this option allows incremental placement. The default value is 0.

**Parameter**

<0|auto|file >: Enable control for incremental placement.

**Example**

set_option -inc_place auto

- -inc_pnr <0|auto|file>

Incremental place & route.

**Syntax**

-inc_pnr <0|auto|file>

**Usage**

| Name | Description |
|---|---|
| <0\|auto\|file > | 0: Disables incremental place & route.<br>auto: Automatic incremental place & route.<br>file: Specifies a *.p file for incremental place & route. |

**Type**

Project

**Description**

Enabling this option allows incremental place & route. The default value is 0.

**Parameter**

<0|auto|file >: Enable control for incremental place & route.

**Example**

set_option -inc_pnr auto

**Note!**

For detailed usage of Place & Route-related running, see Place & Route section in *SUG100, Gowin Software User Guide*.

### Pin Multiplexing Property Configuration

#### -use_jtag_as_gpio

Multiplexes JTAG-related pins as general IO pins.

**Syntax**

-use_jtag_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as JTAG-dedicated pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes JTAG-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing JTAG-related pins as general IO pins.

**Example**

set_option -use_jtag_as_gpio 1

#### -use_sspi_as_gpio

Multiplexes SSPI-related pins as general IO pins.

**Syntax**

-use_sspi_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as SSPI-dedicated pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes SSPI-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing SSPI-related pins as general IO pins.

**Example**

set_option -use_sspi _as_gpio 1

### -use_mspi_as_gpio

Multiplexes MSPI-related pins as general IO pins.

**Syntax**

-use_mspi_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as MSPI-dedicated pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes MSPI-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing MSPI-related pins as general IO pins.

**Example**

set_option -use_mspi _as_gpio 1

### -use_ready_as_gpio

Multiplexes READY-related pins as general IO pins.

**Syntax**

-use_ready_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as READY-dedicated pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes READY-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing READY-related pins as general IO pins.

**Example**

set_option -use_ready_as_gpio 1

### -use_done_as_gpio

Multiplexes DONE-related pins as general IO pins.

**Syntax**

-use_done_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as DONE-dedicated pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes DONE-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing DONE-related pins as general IO pins.

**Example**

set_option -use_ done_as_gpio 1

**-use_reconfign_as_gpio**

Multiplexes RECONFIG_N-related pins as general IO pins.

**Syntax**

-use_ reconfign_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as RECONFIG_N-related pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes RECONFIG_N-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing RECONFIG_N-related pins as general IO pins.

**Example**

set_option -use_reconfign_as_gpio 1

**-use_i2c_as_gpio**

Multiplexes I2C-related pins as general IO pins.

**Syntax**

-use_i2c_as_gpio <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Uses as I2C-related pins.<br>1: Multiplexes as general IO pins. |

**Type**

Project

**Description**

Multiplexes I2C-related pins as general IO pins, and the default is 0.

**Parameter**

<0|1>: Enable control for multiplexing I2C-related pins as general IO pins.

**Example**

set_option -use_i2c_as_gpio 1

### BitStream Property Configuration

#### -bit_format

Specifies the format of the generated bitstream file content.

**Syntax**

-bit_format <txt|bin>

**Usage**

| Name | Description |
|---|---|
| <txt\|bin> | The format of the bitstream file content. |

**Type**

Project

**Description**

Used to specify the format of the generated bitstream file content.

**Parameter**

<txt|bin>: the format of the generated bitstream file content.

**Example**

set_option -bit_format txt

#### -bit_crc_check

Cyclic redundancy check.

**Syntax**

-bit_crc_check <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables cyclic redundancy check.<br>1: Enables cyclic redundancy check. |

**Type**

Project

**Description**

Enables CRC for the bitstream file. The default is 1.

**Parameter**

<0|1>: Enables CRC for the bitstream file.

**Example**

set_option -bit_crc_check 1

**-bit_compress**

Compresses the bitstream file.

**Syntax**

-bit_compress <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not compress the bitstream file.<br>1: Compresses the bitstream file. |

**Type**

Project

**Description**

Compresses the generated bitstream file. The default value is 1.

**Parameter**

<0|1>: Compresses the bitstream file.

**Example**

set_option -bit_compress 1

**-bit_encrypt**

Encrypts the bitstream file.

**Syntax**

-bit_encrypt <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not encrypt the bitstream file.<br>1: Encrypts the bitstream file. |

**Type**

Project

**Description**

Encrypts the bitstream file. This feature is only supported for Arora family devices. The default value is 0.

**Parameter**

<0|1>: Encrypts the bitstream file.

**Example**

set_option -bit_encrypt 1

**-bit_encrypt_key**

Defines the encrypted key.

**Syntax**

-bit_encrypt_key <key>

**Usage**

| Name | Description |
|---|---|
| <key> | The encrypted key. |

**Type**

Project

**Description**

This option works in conjunction with the -bit_encrypt option, allowing users to define a custom encrypted key. The default value is all 0.

**Parameter**

<key>: The encrypted key.

**Example**

set_option -bit_encrypt_key 00000000000000000000000000001101

**-bit_security**

Security bit enable control.

**Syntax**

-bit_security <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables security bit.<br>1: Enables security bit. |

**Type**

Project

**Description**

Security bit enable control, and the default is 1.

**Parameter**

<0|1>: Security bit enable control.

**Example**

set_option -bit_security 1

**-bit_incl_bsram_init**

Prints initial values of BSRAM to the Bitstream file.

**Syntax**

-bit_incl_bsram_init <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not print initial values of BSRAM to the Bitstream file；<br>1: Prints initial values of BSRAM to the Bitstream file. |

**Type**

Project

**Description**

This option is used to control whether the initial values of BSRAM are printed into the bitstream file. The default value is 1. For GW1N and GW2A devices, when this option is set to 1, all positions of the BSRAM will have their initial values printed in the bitstream file, and the initial values of unoccupied BSRAM positions are printed as 0. For GW5A(N)(S)(R)(T) devices, when this option is set to 1, all initial values of BSRAM in the same column as the occupied BSRAM will be printed to the bitstream file, and the initial values of unoccupied BSRAM positions in that column are printed as 0.

**Parameter**

<0|1>: Enable control for printing initial values of BSRAM to the Bitstream file.

**Example**

set_option -bit_incl_bsram_init 1

**-bg_programming**

Background programming function.

**Syntax**

-bg_programming <off | jtag | i2c | goconfig | userlogic | i2c_jtag_sspi_qsspi | jtag_sspi_qsspi>

**Usage**

| Name | Description |
|---|---|
| <off\|jtag\|i2c\|goconfig\|userlogic\|<br>i2c_jtag_sspi_qsspi\|jtag_sspi_qsspi> | Off: Do not use background programming function. |

| Name | Description |
|---|---|
| | jtag: Background programming via JTAG mode.<br>i2c: Background programming via I2C mode.<br>goconfig: Background programming using goConfig IP.<br>userlogic: Background programming using FPGA internal logic.<br>i2c_jtag_sspi_qsspi: Background programming using I2C/JTAG/SSPI/QSSPI mode.<br>jtag_sspi_qsspi: Background programming using JTAG/SSPI/QSSPI mode. |

**Type**

Project

**Description**

It allows programming of Flash without disrupting the FPGA chip's current operations. The default setting is off.

**Parameter**

< off | jtag | i2c | goconfig | userlogic | i2c_jtag_sspi_qsspi | jtag_sspi_qsspi>: Background programming method.

**Example**

```
set_option -bg_programming userlogic
```

**-hotboot**

Hotboot mode.

**Syntax**

-hotboot <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables hotboot mode.<br>1: Enables hotboot mode. |

**Type**

Project

**Description**

Hotboot mode enable control. The default is 0.

**Parameter**

<0|1>: Hotboot mode enable control.

**Example**

```
set_option -hotboot 1
```

### -i2c_slave_addr

Sets I2C device address.

**Syntax**

-i2c_slave_addr <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Sets I2C device address. |

**Type**

Project

**Description**

Sets the I2C device address, and the configurable range is from 00 to 7F. The default is 00.

**Parameter**

<value>: I2C device address.

**Example**

set_option -i2c_slave_addr 2F

### -secure_mode

Enables secure mode.

**Syntax**

-secure_mode <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables secure mode.<br>1: Enables secure mode. |

**Type**

Project

**Description**

Enables secure mode, and at this time the JTAG pins are used as GPIOs; the bitstream file can only be programmed into the device once. The default is 0.

**Parameter**

<0|1>: Enables control for secure mode.

**Example**

set_option -secure_mode 1

### -loading_rate

The loading speed of bitstream data from Flash to SRAM in AutoBoot and MSPI configuration modes.

**Syntax**

-loading_rate <value>

**Usage**

| Name | Description |
|---|---|
| <value> | The loading speed of bitstream data from Flash to SRAM in AutoBoot and MSPI configuration modes. |

**Type**

Project

**Description**

The loading speed of bitstream data from Flash to SRAM in AutoBoot and MSPI configuration modes, and the default value is 2.500MHz.

**Parameter**

<value>: The loading speed of bitstream data from Flash to SRAM.

**Example**

set_option -loading_rate 21.000MHz

**-seu_handler**

Enable Single Event Upset (SEU) handler.

**Syntax**

-seu_handler <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables SEU handler.<br>1: Enables SEU handler. |

**Type**

Project

**Description**

Enables SEU handler, and the default is 0.

**Parameter**

<0|1>: Enable control for SEU handler.

**Example**

set_option -seu_handler 1

**-seu_handler_mode**

Selects the mode for booting or stopping the SEU Handler.

**Syntax**

-seu_handler_mode <auto|userlogic>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | auto: Automatically enables the SEU handler after the chip wakes up.<br>userlogic: Uses logic to enable or stop the SEU handler. |

**Type**

Project

**Description**

Selects the mode for booting or stopping the SEU Handler, and the default is auto.

**Parameter**

< auto|userlogic>: The mode for booting or stopping the SEU Handler.

**Example**

set_option -seu_handler_mode userlogic

**-seu_handler_checksum**

Enables SEU processing, checking, calculation, and comparison.

**Syntax**

-seu_handler_checksum <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables SEU processing, checking, calculation, and comparison.<br>1: Enables SEU processing, checking, calculation, and comparison. |

**Type**

Project

**Description**

Enables SEU processing, checking, calculation, and comparison, and the default is 0.

**Parameter**

<0|1>: Enables control for SEU processing, checking, calculation, and comparison.

**Example**

set_option -seu_handler_checksum 1

**-error_detection**

Enables error detection only.

**Syntax**

-error_detection <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables error detection.<br>1: Enables error detection only. |

**Type**

Project

**Description**

Enables error detection only, and the default is 0.

**Parameter**

<0|1>: Enables control for error detection only.

**Example**

set_option -error_detection 1

### -error_detection_correction

Enables error detection and correction.

**Syntax**

-error_detection_correction <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables error detection and correction.<br>1: Enables error detection and correction. |

**Type**

Project

**Description**

Enables error detection and correction, and the default is 0.

**Parameter**

<0|1>: Enables control for error detection and correction.

**Example**

set_option -error_detection_correction 1

### -stop_seu_handler

Stops SEU Handler.

**Syntax**

-stop_seu_handler <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not stop the SEU Handler when an uncorrectable ECC error or CRC checksum mismatch is detected.<br>1: Stops the SEU Handler when an uncorrectable ECC error or CRC checksum mismatch is detected. |

**Type**

Project

**Description**

Stops the SEU Handler when an uncorrectable ECC error or CRC checksum mismatch is detected. The default is 0.

**Parameter**

<0|1>: Enable control for stopping SEU Handler.

**Example**

set_option -stop_seu_handler 1

**-osc_div**

Sets the frequency division ratio for the extended control register.

**Syntax**

-osc_div <4|8|16|32>

**Usage**

| Name | Description |
|---|---|
| <4\|8\|16\|32> | 4: Sets the frequency division ratio of the extended control register to 4.<br>8: Sets the frequency division ratio of the extended control register to 8.<br>16: Sets the frequency division ratio of the extended control register to 16.<br>32: Sets the frequency division ratio of the extended control register to 32. |

**Type**

Project

**Description**

Sets the frequency division ratio for the extended control register, and the default is 8.

**Parameter**

<4|8|16|32>: The frequency division ratio for the extended control register.

**Example**

set_option -osc_div 8

**-error_injection**

Enables error injection.

**Syntax**

-error_injection <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables error injection.<br>1: Enables error injection. |

**Type**

Project

**Description**

Enables error injection, and the default is 0

**Parameter**

<0|1>: Enables control for error injection.

**Example**

set_option -error_injection 1

**-ext_cclk**

Enables external primary configuration clock.

**Syntax**

-ext_cclk <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables external primary configuration clock.<br>1: Enables external primary configuration clock. |

**Type**

Project

**Description**

Enables external primary configuration clock. The default is 0.

**Parameter**

<0|1>: Enable control for external primary configuration clock.

**Example**

set_option -ext_cclk 1

**-ext_cclk_div**

Sets the frequency divider parameter.

**Syntax**

-ext_cclk_div <value>

**Usage**

| Name | Description |
|---|---|
| <value> | Frequency divider Parameter |

**Type**

Project

**Description**

Sets the frequency divider parameter.

**Parameter**

<value>: Sets the frequency divider parameter.

**Example**

set_option -ext_cclk_div 4

**-multi_boot**

Multi Boot enable control.

**Syntax**

-multi_boot <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables Multi Boot.<br>1: Enables Multi Boot. |

**Type**

Project

**Description**

Multi Boot enable control, and the default is 0.

**Parameter**

<0|1>: Multi Boot enable control.

**Example**

set_option -multi_boot 1

**-multiboot_address_width**

Configures SPI Flash address width.

**Syntax**

-multiboot_address_width<24|32>

**Usage**

| Name | Description |
|---|---|
| <24\|32> | 24: Sets the SPI Flash address width to 24.<br>32: Sets the SPI Flash address width to 32. |

**Type**

Project

**Description**

Configures SPI Flash address width, and the default is 24.

**Parameter**

<24|32>: SPI Flash address width.

**Example**

set_option -multiboot_address_width 32

**-multiboot_spi_flash_address**

Specifies SPI Flash address.

**Syntax**

-multiboot_spi_flash_address <value>

**Usage**

| Name | Description |
|---|---|
| <value> | SPI Flash address. |

**Type**

Project

**Description**

Specifies the SPI Flash address. The SPI Flash address refers to the start address of the bitstream file to be loaded during the next multiboot, with the default value of 000000.

**Parameter**

< value>: SPI Flash address.

**Example**

set_option -multiboot_spi_flash_address 000110

**-multiboot_mode**

Configures SPI Flash address access mode.

**Syntax**

-multiboot_mode <single | fast | dual | quad>

**Usage**

| Name | Description |
|---|---|
| < single\|fast\|dual\|quad > | single: Uses single mode<br>fast: Uses fast mode<br>dual: Uses gedual mode<br>quad: Uses gequad mode |

**Type**

Project

**Description**

Configures SPI Flash address access mode, and the default is single.

**Parameter**

<single | fast | dual | quad>: SPI Flash address access mode.

**Example**

set_option -multiboot_mode single

### -mspi_jump

MSPI JUMP enable control.

**Syntax**

-mspi_jump<0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables MSPI JUMP.<br>1: Enables MSPI JUMP. |

**Type**

Project

**Description**

MSPI JUMP enable control, and the default is 0.

**Parameter**

<0|1>: MSPI JUMP enable control.

**Example**

set_option -mspi_jump 1

### -merge_jumpbit

Merges the MSPI JUMP bitstream file into the general bitstream file.

**Syntax**

-merge_jumpbit <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Do not merge the bitstream file.<br>1: Merges the bitstream file. |

**Type**

Project

**Description**

Merges the MSPI JUMP bitstream file into the general bitstream file, and the default is 0.

**Parameter**

<0|1>: Enables control for merging the MSPI JUMP bitstream file into the general bitstream file.

**Example**

set_option -merge_jumpbit 1

### -mspijump_address_width

Configures SPI Flash address width.

**Syntax**

-mspijump_address_width <24|32>

**Usage**

| Name | Description |
|---|---|
| <24\|32> | 24: Sets the SPI Flash address width to 24.<br>32: Sets the SPI Flash address width to 32. |

**Type**

Project

**Description**

Configures SPI Flash address width, and the default is 24.

**Parameter**

<24|32>: SPI Flash address width.

**Example**

set_option - mspijump_address_width 32

**-mspijump_spi_flash_address**

Specifies SPI Flash address.

**Syntax**

-mspijump_spi_flash_address <value>

**Usage**

| Name | Description |
|---|---|
| <value> | SPI Flash address |

**Type**

Project

**Description**

Specifies SPI Flash address, and the default is 000000.

**Parameter**

<value>: SPI Flash address

**Example**

set_option - mspijump_spi_flash_address 000110

**-mspijump_mode<single | fast | dual | quad>**

Configures SPI Flash address access mode.

**Syntax**

-mspijump_mode <single | fast | dual | quad>

**Usage**

| Name | Description |
|---|---|
| < single\|fast\|dual\|quad > | single: Usagesingle mode<br>fast: Usagefast mode<br>dual: Usagedual mode<br>quad: Usagequad mode |

**Type**

Project

**Description**

Configures SPI Flash address access mode, and the default is single.

**Parameter**

< single | fast | dual | quad >: SPI Flash address access mode.

**Example**

set_option -mspijump_mode single

**-program_done_bypass**

Forward new bitstream.

**Syntax**

-program_done_bypass <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables this function.<br>1: Enables this function. |

**Type**

Project

**Description**

After configuring this option, when the internal signal of Done Final is activated, the external Done Pin will remain low, allowing the new bitstream to be forwarded after the current bitstream is loaded. The default value is 0.

**Parameter**

<0|1>: Enable control for forwarding new bitstream.

**Example**

set_option -program_done_bypass 1

**-power_on_reset_monitor <0|1>**

Power on reset.

**Syntax**

-power_on_reset_monitor <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables power on reset.<br>1: Enables power on reset. |

**Type**

Project

**Description**

Enable control for power on reset. The default is 0.

**Parameter**

<0|1>: Enable control for power on reset.

**Example**

set_option -power_on_reset_monitor 1

**-turn_off_bg**

Bandgap function.

**Syntax**

-turn_off_bg <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Enables Bandgap.<br>1: Disables Bandgap. |

**Type**

Project

**Description**

Enable control for Bandgap. The default is 0.

**Parameter**

<0|1>: Enable control for Bandgap.

**Example**

set_option -turn_off_bg 1

**-wakeup_mode**

Wake Up Mode enable control.

**Syntax**

-wakeup_mode <0|1>

**Usage**

| Name | Description |
|---|---|
| <0\|1> | 0: Disables Wake Up Mode.<br>1: Enables Wake Up Mode. |

**Type**

Project

**Description**

Enable control for Wake Up Mode. The default is 0.

**Parameter**

<0|1>: Enable control for Wake Up Mode.

**Example**

set_option -wakeup_mode 1

**-user_code**

Custom User Code.

**Syntax**

-user_code <default|value>

**Usage**

| Name | Description |
|---|---|
| <default\|value> | Custom User Cod value. |

**Type**

Project

**Description**

Users can customize the User Code. The default value is default (00000000).

**Parameter**

<default|value>: Custom User Code.

**Example**

set_option -user_code 00000010

**Note!**

For a detailed explanation of the BitStream-related options, see BitStream section in *SUG100, Gowin Software User Guide*.

### Unused Pin Property Configuration

**-unused_pin**

Sets different IO properties for unused GPIOs.

**Syntax**

-unused_pin <default|open_drain>

**Usage**

| Name | Description |
|---|---|
| <default\|open_drain> | Default: All unused GPIO pins will be configured as input tri-state with weak pull-up.<br>open_drain：open_drain: All unused GPIO pins will be configured as output with setting OPEN DRAIN as ON. |

**Type**

Project

**Description**

Sets different IO properties for unused GPIOs.

**Parameter**

<default|open_drain>: Sets different IO properties for unused GPIOs.

**Example**

set_option -unused_pin open_drain

**Note!**

For a detailed explanation of Unused Pin-related options, see Unused Pin in *SUG100, Gowin Software User Guide*.

### 3.2.20 set_property

Sets the properties of an object.

**Syntax**

set_property [-dict <args>] <name> <value> <objects>

**Usage**

| Name | Description |
|---|---|
| [-dict] | A property list of (name/value) pairs to be set. |
| <name> | The name of the property to be set. Not applicable when using -dict. |
| <value> | The value of the property to be set. Not applicable when using -dict. |
| <objects> | The object of the property to be set. |

**Type**

IPFlow

**Description**

Configure options and their corresponding values for specified IP object.

**Parameter**

- [-dict]: Specifies a dictionary containing multiple pairs of options and their corresponding values. Each pair is defined as (<name> <value>), and multiple pairs are separated by spaces. The dictionary should be enclosed in curly braces {}.
- <name>: Specifies the name of the option to be configured. The description is in the form of CONFIG.property, and the property refers to the option name.
- <value>: Specifies the value for the corresponding option. The value needs to be valid according to the property type. If the option value is a string, it should follow the original form.

- <objects>: Specifies one or more IP objects for which the option is to be configured. A single object is specified using [get_ips module_name]. To specify multiple objects, use [get_ips module_name0 module_name1 …].

### Example

The -dict option is used to specify multiple properties in the current design at once:

set_property -dict {CONFIG.Data_Width 16 CONFIG.Write_Depth 1024 CONFIG.Read_Depth 1024} [get_ips FIFO_Top]

Use name, value, objects to specify a single property in the current design:

set_property CONFIG.Data_Width {16} [get_ips FIFO_Top]

This example shows how to set a property value that contains a hyphen ("-") or spaces:

set_property {CONFIG.Almost_Full_Type} {Full-Single Threshold Constant Parameter} [get_ips FIFO_Top]

**Note!**

In some cases, the option value may contain special characters, such as a hyphen ("-") or spaces, which could cause the value to be incorrectly parsed. In such cases, it is necessary to enclose the option value in curly braces {}.

### See also

- list_property
- report_property

## 3.2.21 source

In tcl command editor window of Gowin Software, or after launching command-line mode, this command can be used to execute a tcl script. For file path formats, see 3.2.1 add_file.

### Syntax

source <file>

### Usage

| Name | Description |
|---|---|
| <file> | The tcl script to be executed. |

### Type

IPFlow, Project

### Description

Sets different IO properties for unused GPIOs.

### Parameter

<file>: The tcl script to be executed.

### Example

```
source project.tcl
source D:/gowin_project/project.tcl
source D:\\gowin_project\\project.tcl
```

## 3.2.22 write_ip_tcl

Exports a tcl script that can regenerate the given IP.

### Syntax

write_ip_tcl [-ip_name <newModuleName>] [-multiple_files] [-force] [<tcl_filename>] <objects>

### Usage

| Name | Description |
|---|---|
| [-ip_name] | Sets the IP module name |
| [-multiple_files] | Creates a separate .tcl file for each IP object |
| [-force] | Overwrites existing files |
| [<tcl_filename>] | Exported tcl file |
| <objects> | IP objects that export tcl files |

### Type

IPFlow

### Description

This command is used to export tcl script files for specified IP objects, enabling regeneration of the IPs via tcl scripts. For file path formats, see 3.2.1 add_file.

### Parameter

- [<tcl_filename>]: Name of the generated tcl script file. If not specified, the current project name is used.
- <objects>: Specifies one or more IP objects to generate the tcl script. A single object is specified using [get_ips module_name], and multiple objects are specified using [get_ips module_name0 module_name1 …].
- [-ip_name <newModuleName>]: Renames the IP module name in the generated tcl script. Only one object can be specified.
- [-multiple_files]: Generates tcl script files for all specified IPs, with file names based on the respective module_name. This option is mutually exclusive with -ip_name and <tcl_filename>.
- [-force] Overwrites existing tcl files with the same name.

### Example

In this example, a tcl file is created, and this file is used to specify the IP object FIFO_Top, but when using the source command, it is created with the new name FIFO_Top_new:

```
write_ip_tcl -ip_name FIFO_Top_new [get_ips FIFO_Top]
```

In this example, a separate tcl file is generated for each specified IP module in the project.

```
write_ip_tcl -multiple_files [get_ips FIFO_Top FIFO_Top_1]
```

In this example, multiple specified IP modules from the project are written into a single Tcl file:

```
write_ip_tcl [get_ips FIFO_To
p FIFO_Top_1] my_fifo.tcl
```

GOWIN

PROGRAMMING FOR THE FUTURE