Compa Family CPLDs Development Software Application Guide

(AN03020, V1.0)

(19.04.2022)

Revisions History

Document Revisions

Version	Date of Release	Revisions
V1.0	19.04.2022	Initial release.

About this Manual

Terms and Abbreviations

Terms and Abbreviations	Meaning
JTAG	Joint Test Action Group
SPI	Serial Peripheral Interface
I2C	Inter-Integrated Circuit
SVF	Serial Vector Format
CRAM	Configuration RAM
CCS	Configuration Control System

Related Documentation

The following documentation is related to this manual:

1. ***AN03008_Compa Family Loaded Platform Software Application Guide***
2. ***AN03009_Compa Family CPLDs Device Slave SPI Interface Configuration and Programming Application Guide***
3. ***AN03013_Compa Family CPLDs Device JTAG Interface Configuration and Programming Application Guide***
4. ***AN03014_Compa Family CPLDs Device Slave I2C Interface Configuration and Programming Application Guide***
5. ***UG030004_Compa Family CPLDs Configuration User Guide***

Table of Contents

Chapter 1 Start Software

1.1 Software Requirements

The EDA tool from Pango Microsystems is called Pango Design Suite, hereinafter referred to as PDS, which includes both Linux and Windows versions.

PDS requires RHEL6.6 (64-bit) or higher to run on Linux systems.

PDS requires Windows 7 Ultimate Edition Service Pack 1 (64-bit) or higher to run on Windows systems. This document introduces the use of PDS software based on Compa family CPLD devices.

1.2 Hardware Requirements

The minimum hardware configuration required to run PDS is:

- Processor: 2GHz
- Memory: 8GB
- Hard drive: 10GB

1.3 PDS Installation and Uninstallation

1.3.1 Windows 7 System

To install PDS, double-click the Setup.exe file in the installation package, and then follow the software prompts to complete the installation.

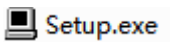


Figure 1-1

After installation, users will be prompted to install drivers vcredist_VS2017.exe, USB Cable Driver, and Parallel Port Driver. It is recommended to select [Yes].

To uninstall PDS, please click [Start > All Programs > Pango > Pango Design Suite XXX > Uninstall]; or open [Control Panel > Programs and Functions] and select the PDS version to be uninstalled.

1.3.2 RHEL6.6 System

To install PDS, right-click the software compressed package and click [Extract here] in the menu item that appears.

Users can also decompress via command, enter in the terminal window according to the actual installation package name (e.g., PDS_2019.3-RHEL6.6):

"tar –zxvf PDS_2019.3-RHEL6.6-x64.tar.gz"

Then, decompress it.

If users need to install the USB Cable Driver, perform the following steps:

- Obtain user permissions (if users already have root access, this step is not necessary), switch to the root account, or directly use the command "sudo -s" (or, if sudo is not available on your system: su);
- Run install.

To uninstall PDS, simply delete the entire software folder (e.g., "/home/PDS_2019.3"), and also delete the configuration files (e.g., "/home/user/.config/pango/2019.3").

To uninstall the USB Cable Driver, just delete all files copied during installation as well as Soft Connect:

"rm –f /usr/local/lib/libftd2xx* "

1.4 Licence and Environment Variables

After installing PDS, users cannot use it directly. users need to apply for a Licence and correctly set environment variables before users can open the PDS software. For specific operations, please refer to the document "Pango_Design_Suite_Windows_Install_Guide", which can be obtained by extracting the PDS compressed package.

Chapter 2 Project Management

2.1 New Project

- Double-click the Pango Design Suite shortcut on the desktop to start PDS;

Figure2-1

- The initial interface of PDS is shown in the figure below. Select [New Project] to open the new project wizard and start creating a new project;

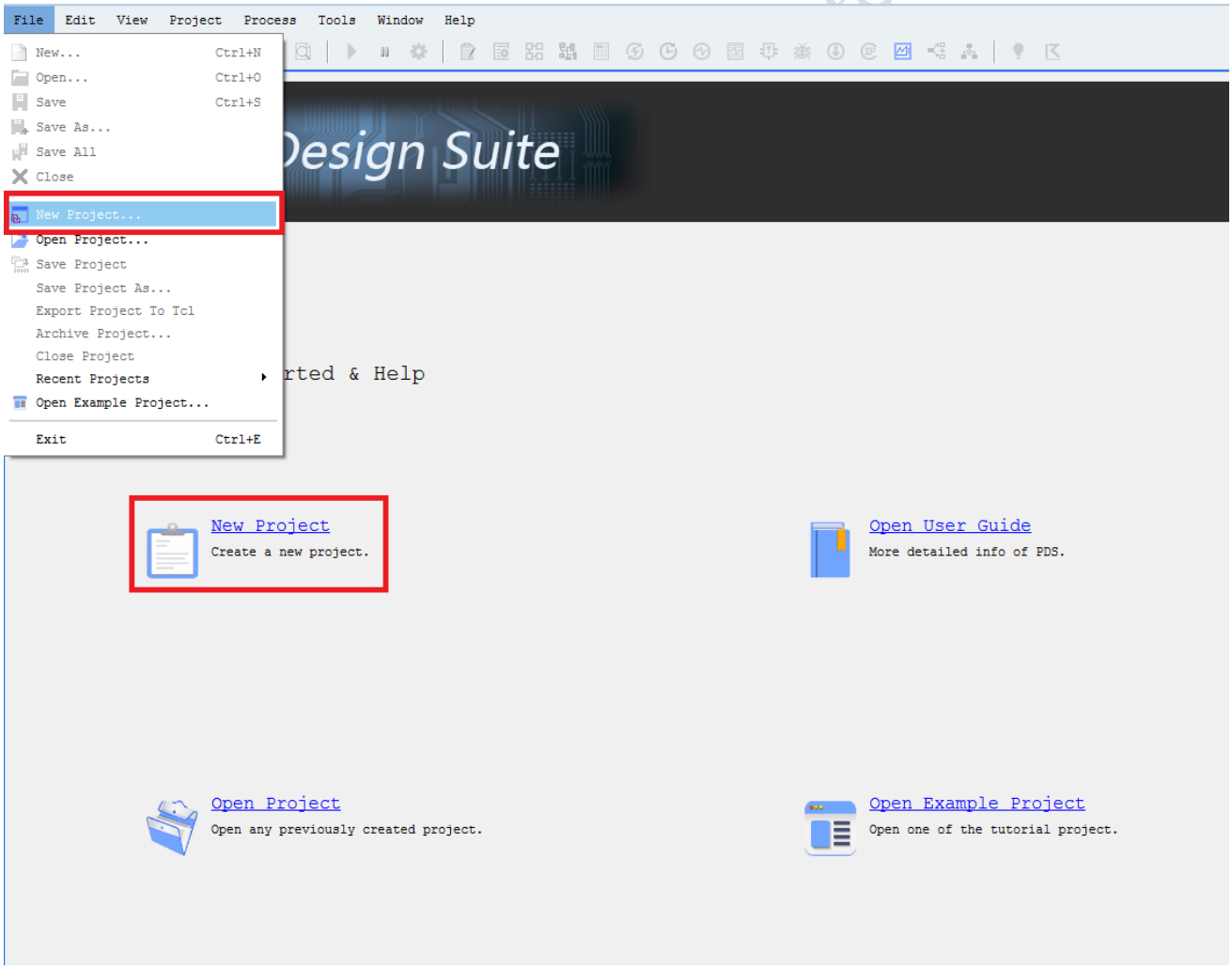


Figure2-2

- After reviewing the introduction, click [Next];

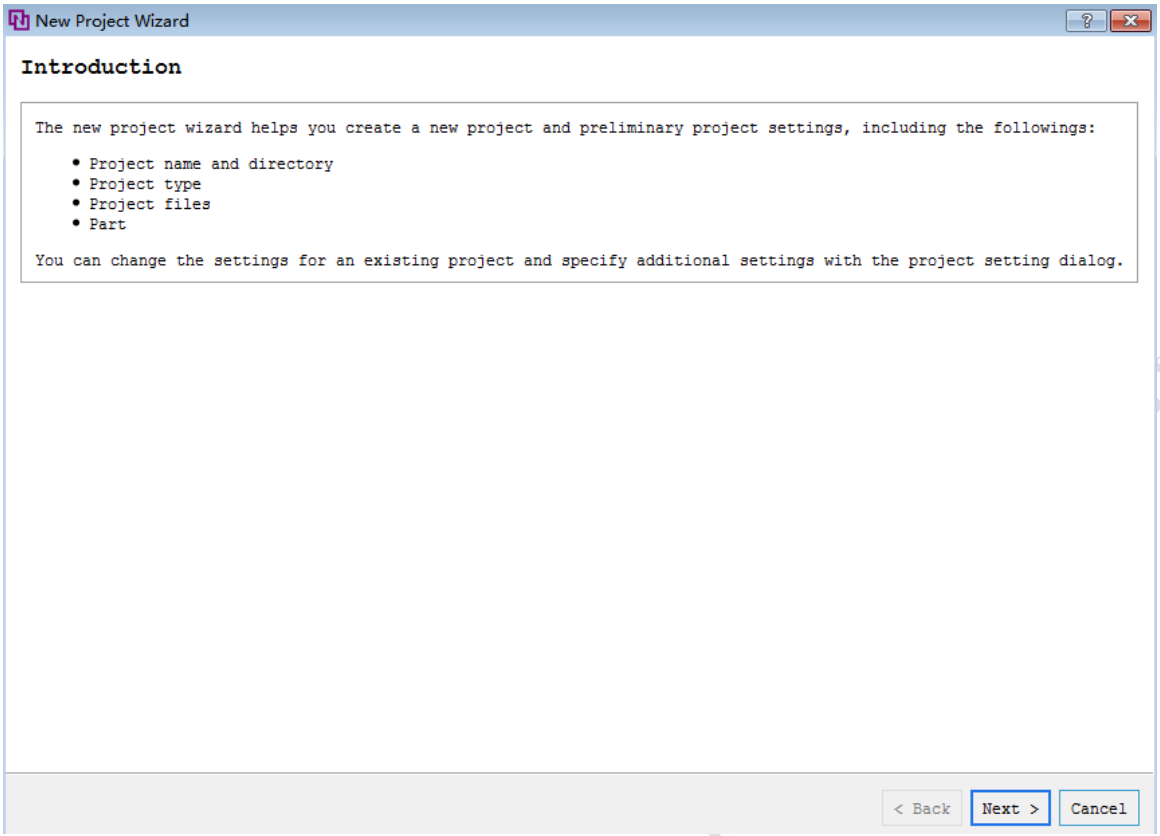


Figure2-3

➢ Enter the project name and project location, and then click [Next];

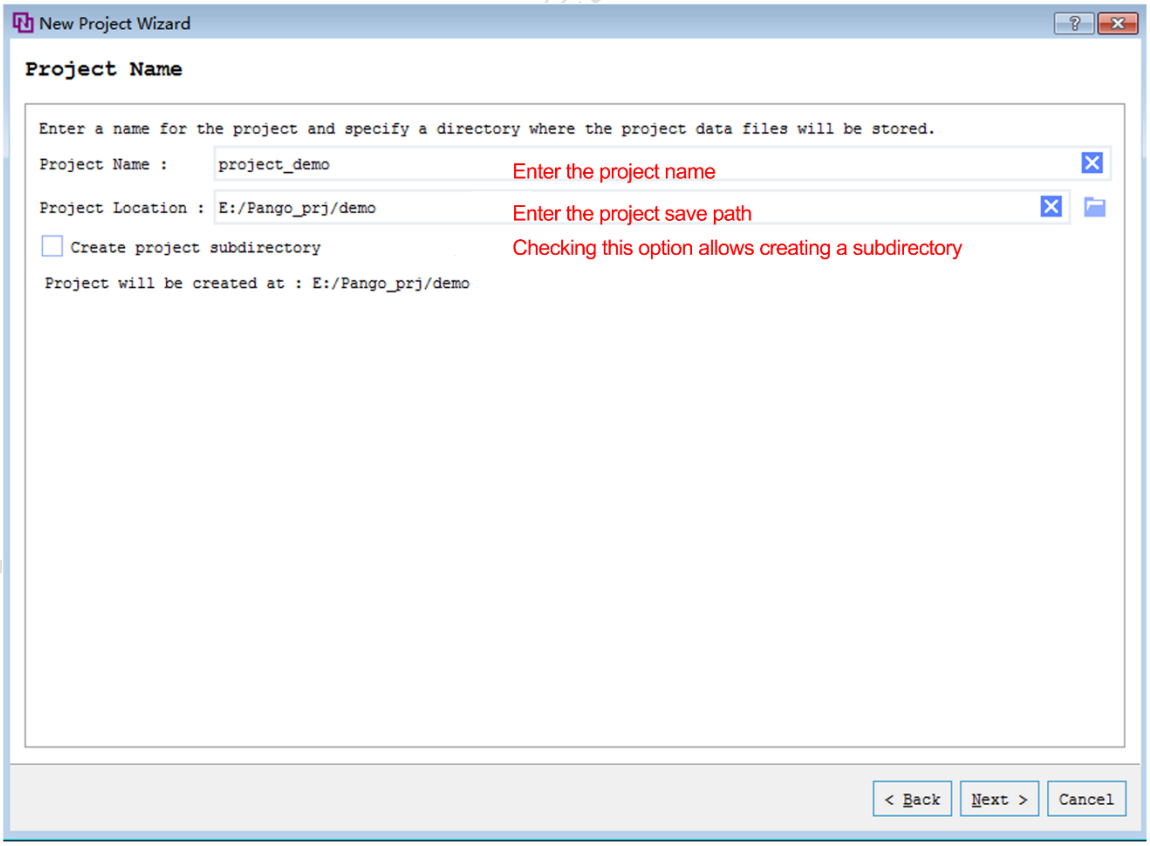


Figure2-4

➢ Select the type of new project;

[RTL project]: The input file is a design source file, with synthesis tools supported;

[Post-Synthesize project]: The input file is synthesized netlist files, with no synthesis required.

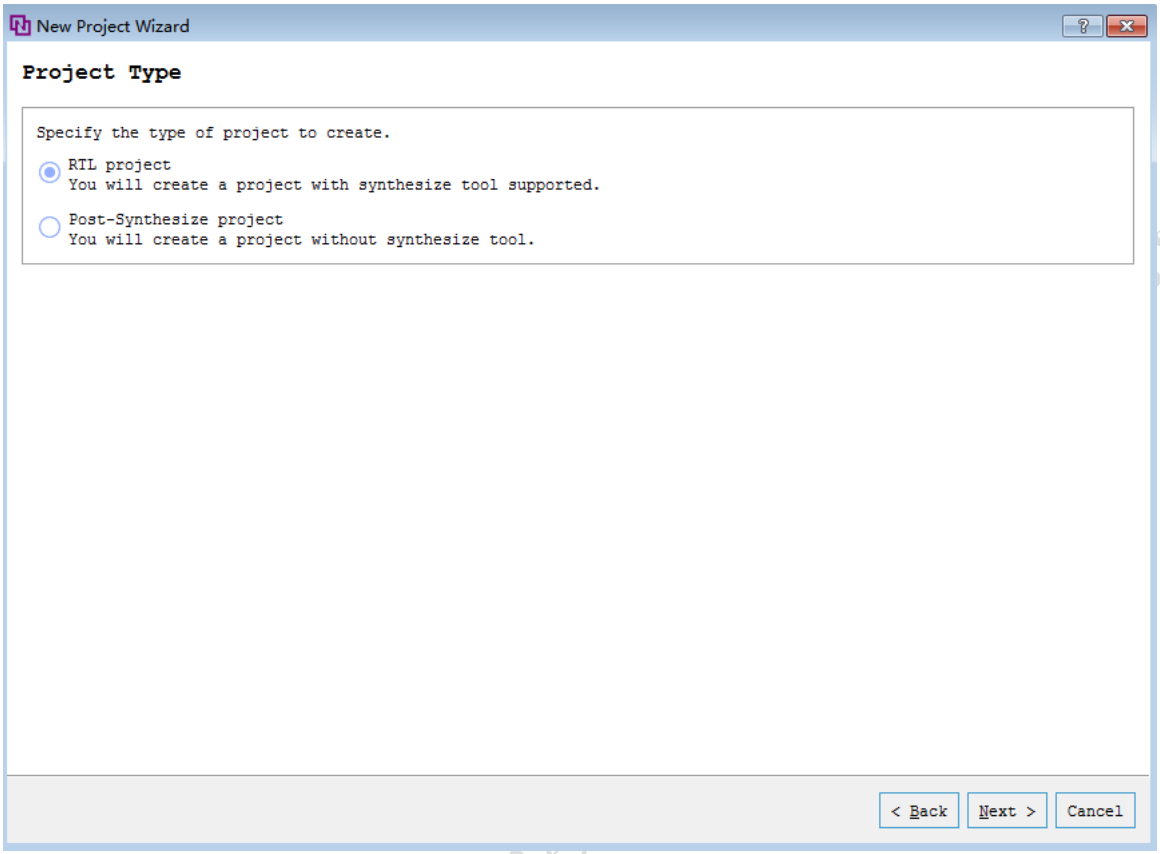


Figure2-5

- Add design source files

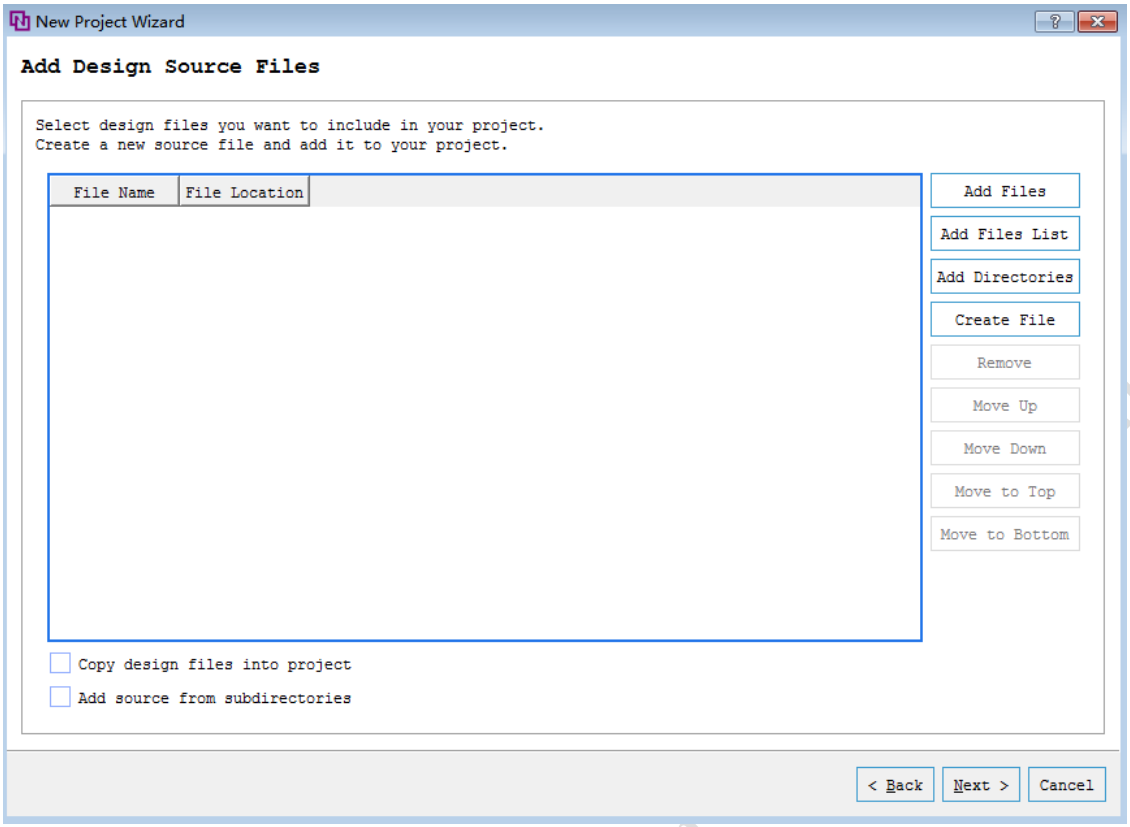


Figure2-6

[Add Files]: Add existing individual or multiple design source files;

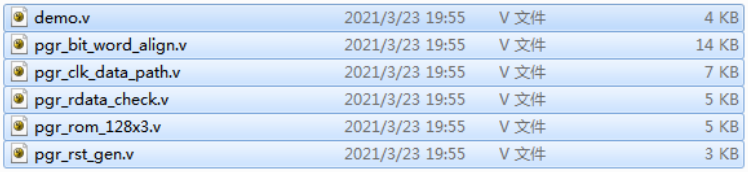


Figure2-7

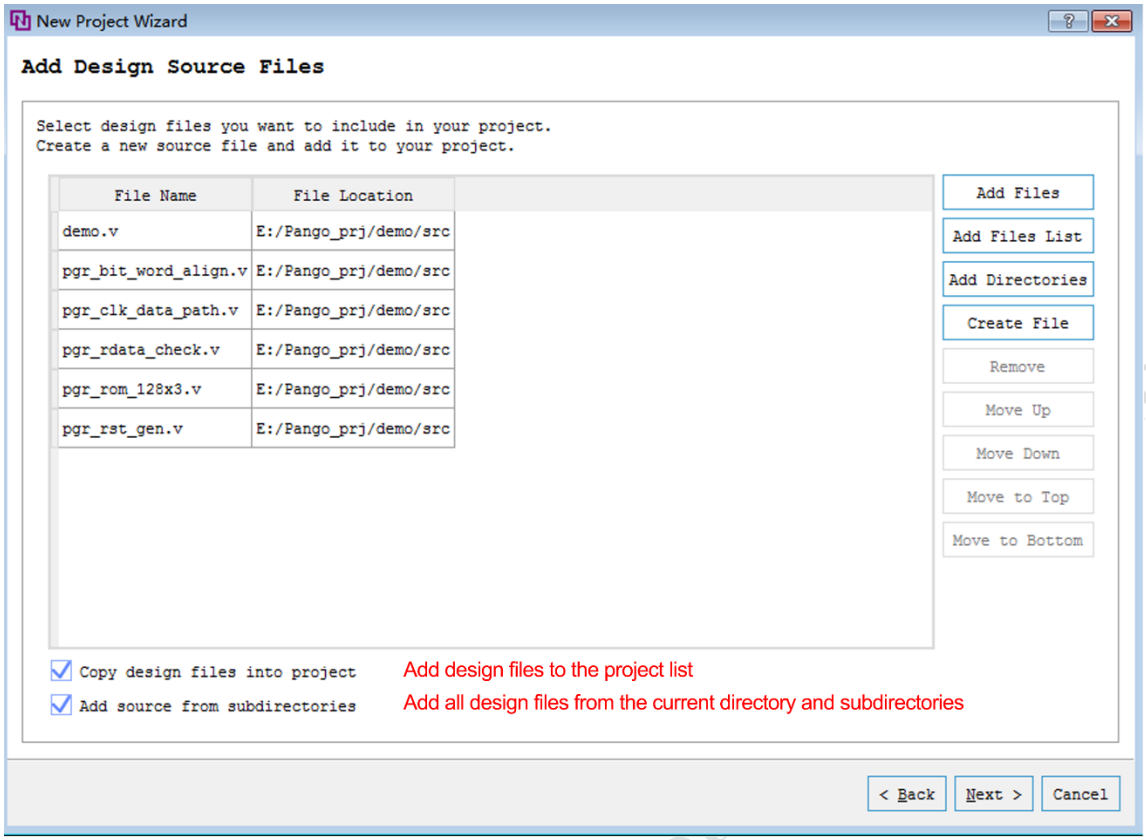


Figure2-8

[Add Files List]: Add a file list, usually a .txt file, which contains the storage paths and names of the design source files, as shown in the figure below;

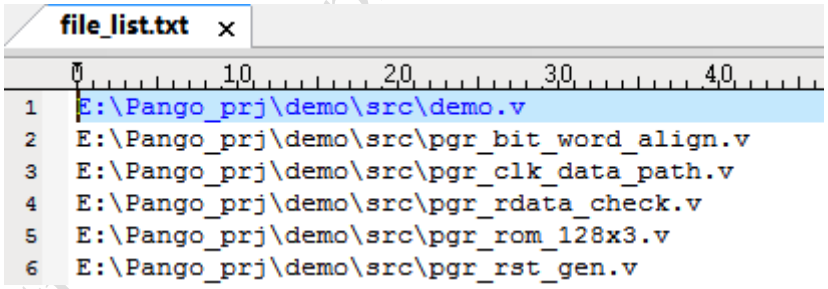


Figure2-9

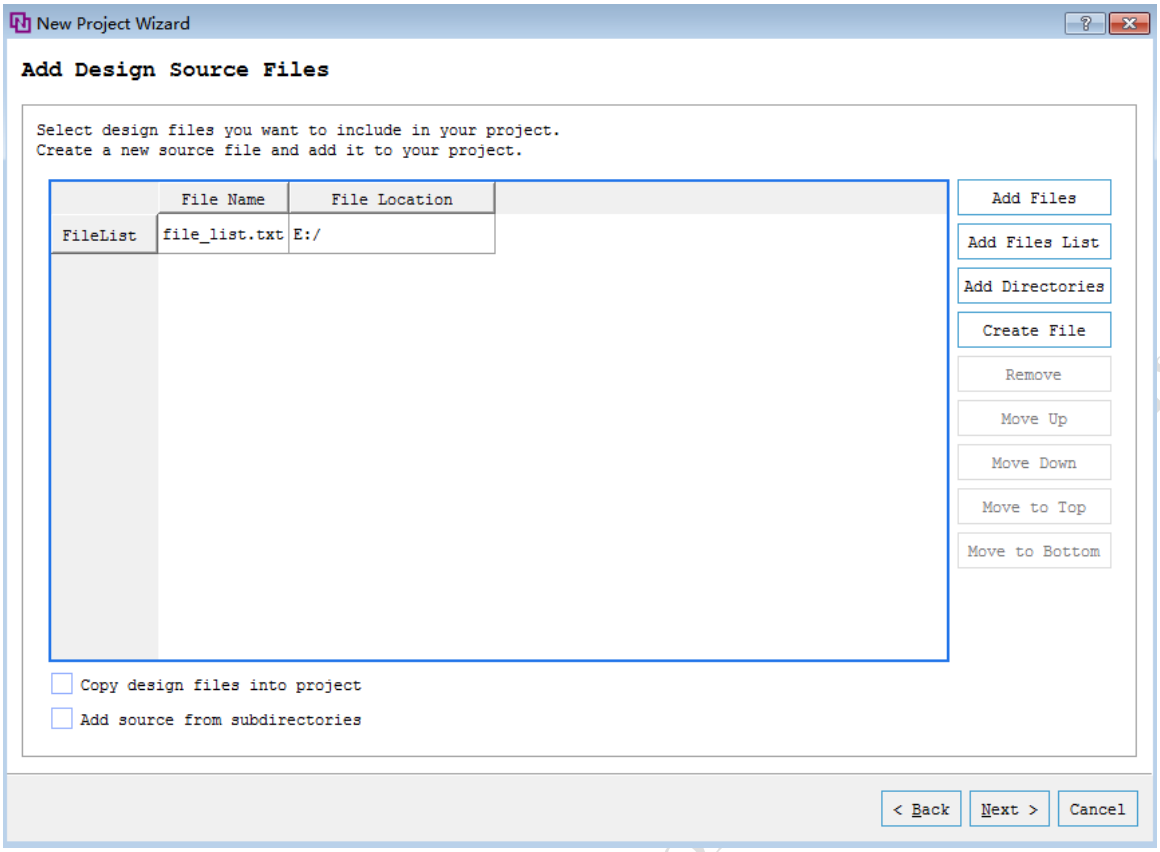


Figure2-10

[Add Directories]: Add directories of design source files;

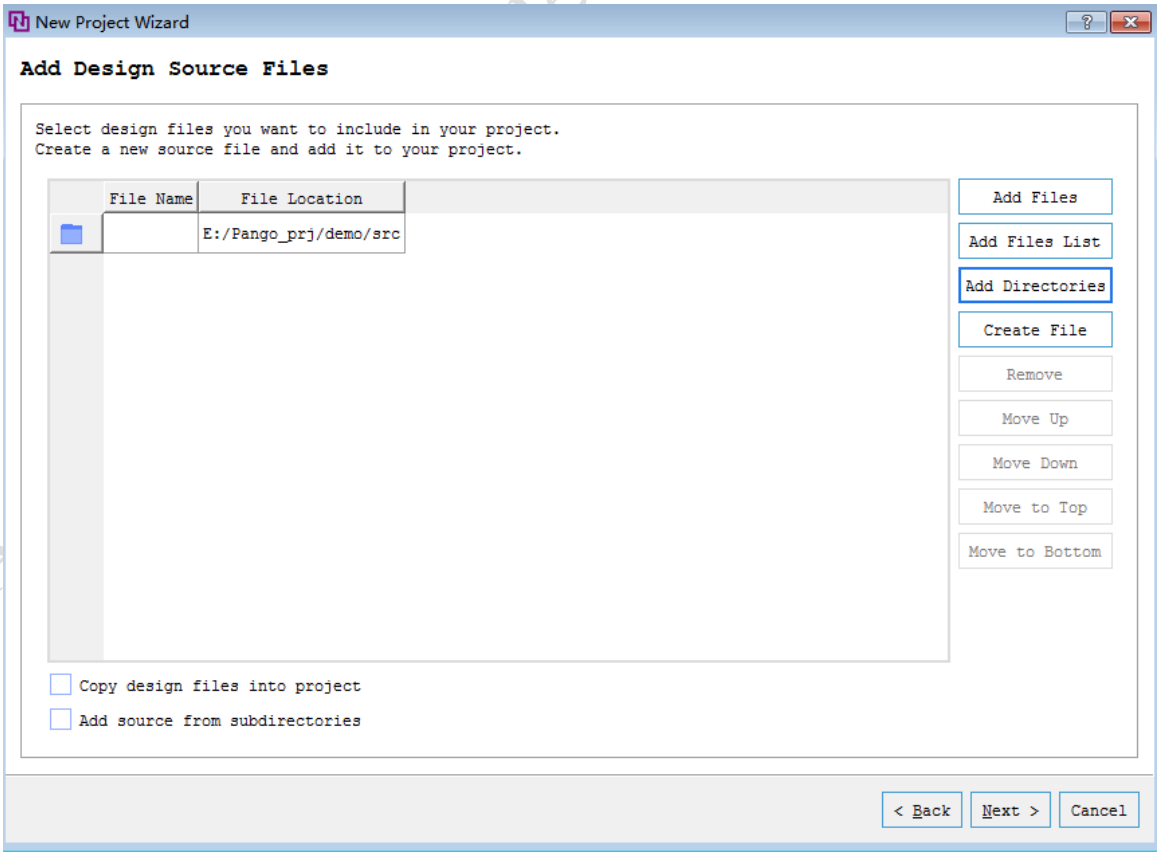


Figure2-11

[Create File]: Create a new design file;

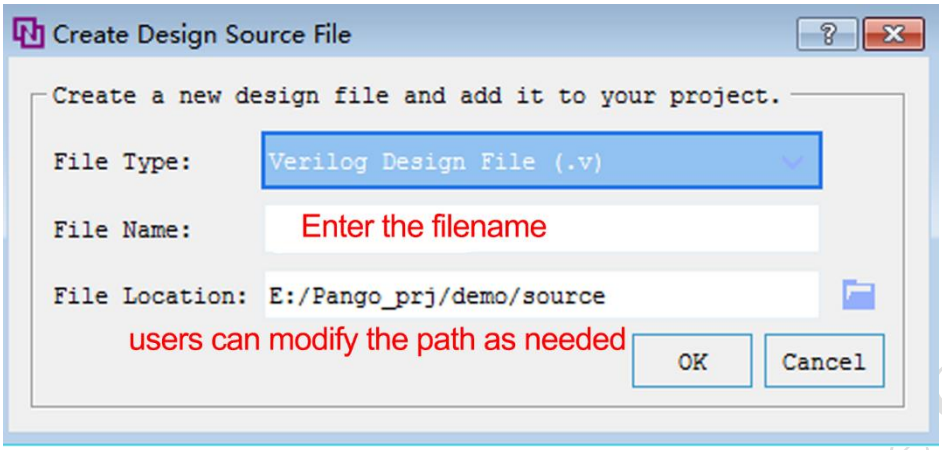


Figure2-12

[Remove]: Remove files;

[Move Up]: Move the target file up one line;

[Move Down]: Move the target file down one line;

[Move to Top]: Move the target file directly to the top of the file list;

[Move to Bottom]: Move the target file directly to the bottom of the file list.

- Add IP files;

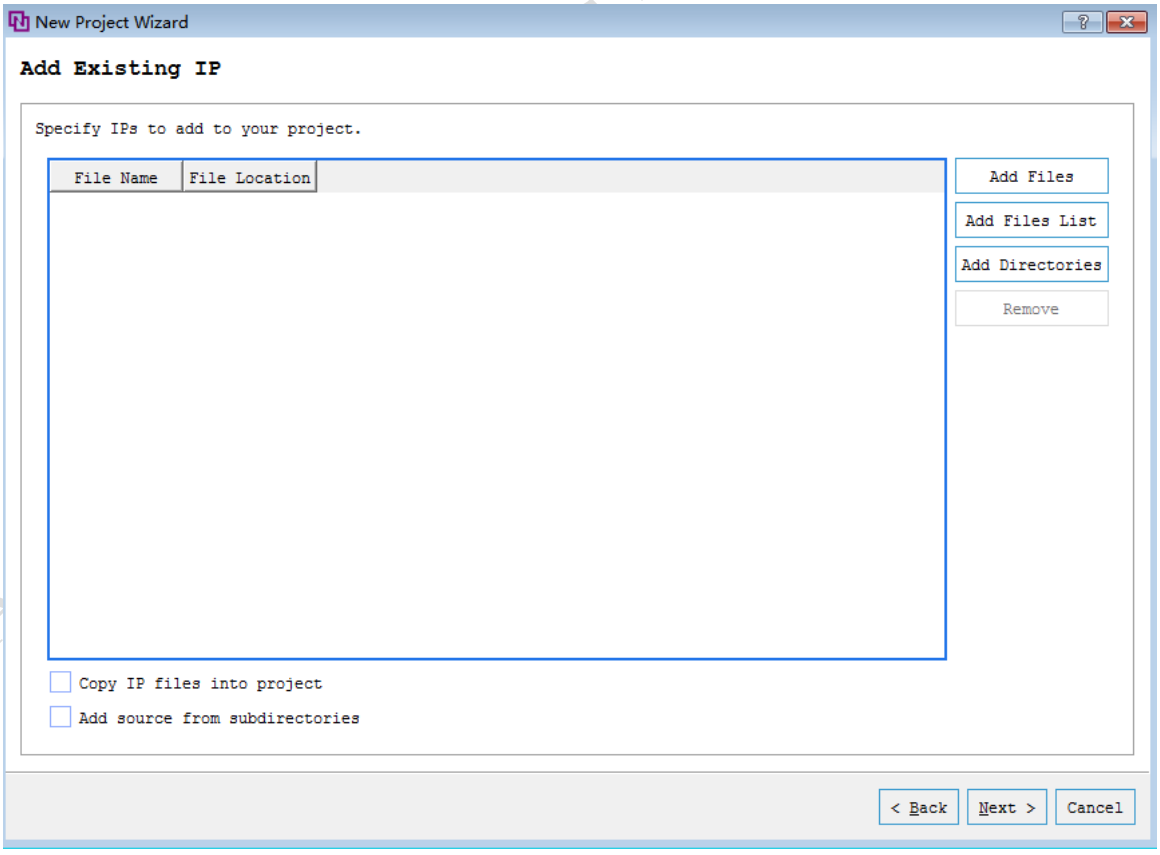


Figure2-13

[Add Files]: Add existing IP project files, i.e., .idf files;

[Add Files List]: Add a list of IP files;

[Add Directories]: Add IP directories.

[Remove]: Remove files.

- Add constraint files;

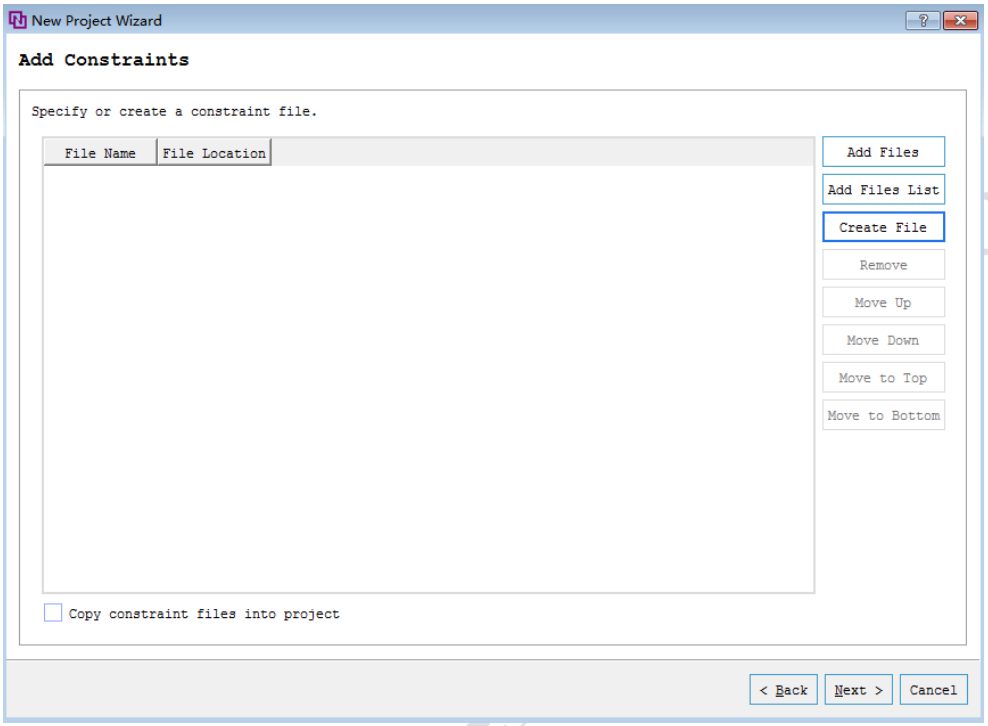


Figure2-14

[Add Files]: Add existing constraint files;

[Add Files List]: Add a list of constraint files;

[Create File]: Add a new constraint file;

[Remove]: Remove files;

[Move Up]: Move the target file up one line;

[Move Down]: Move the target file down one line;

[Move to Top]: Move the target file directly to the top of the file list;

[Move to Bottom]: Move the target file directly to the bottom of the file list.

- Select device;

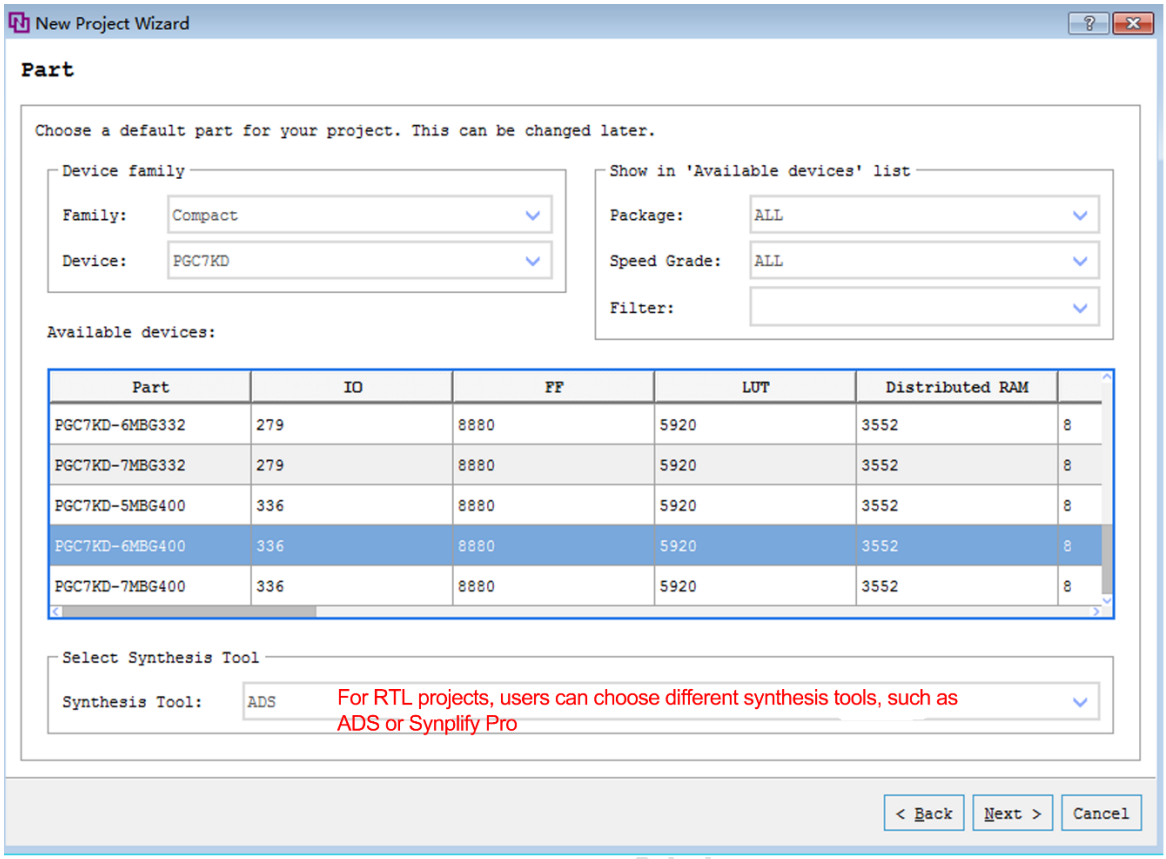


Figure2-15

- Check the [Summary], and click [Finish] to create the new project.

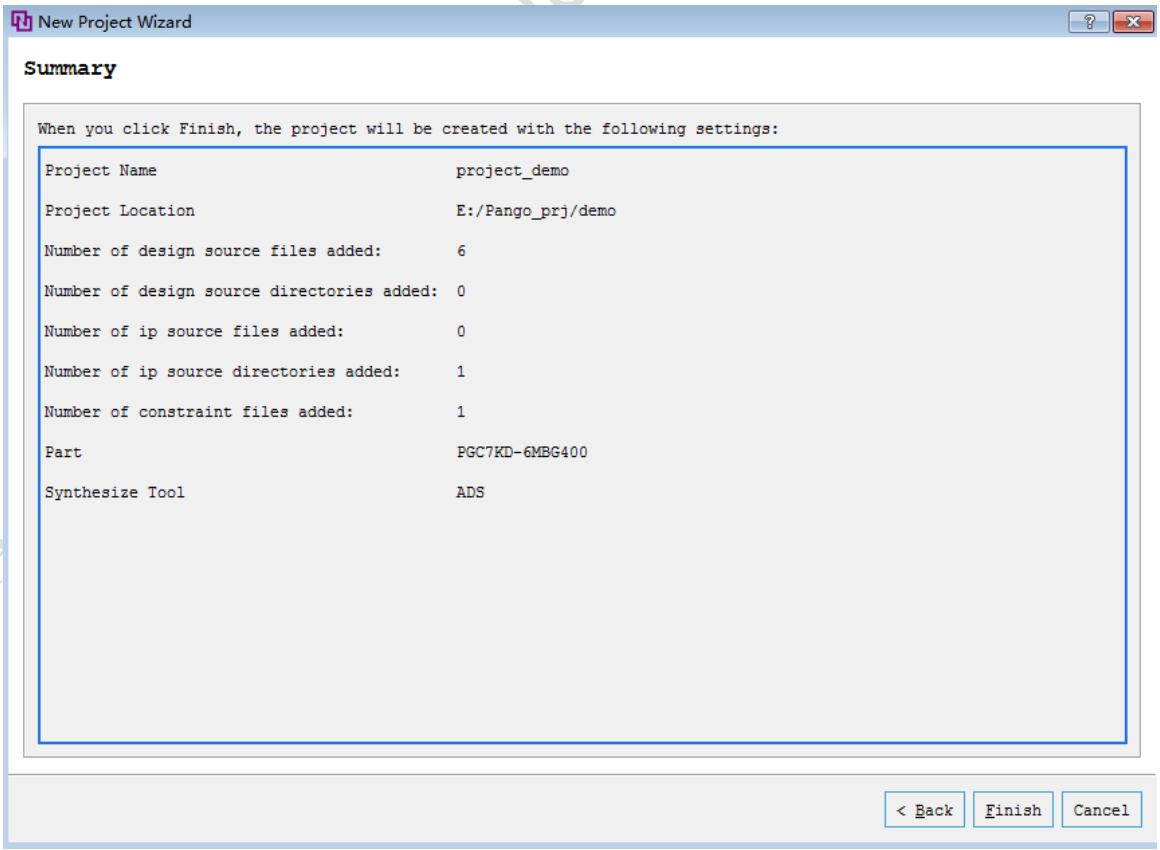


Figure2-16

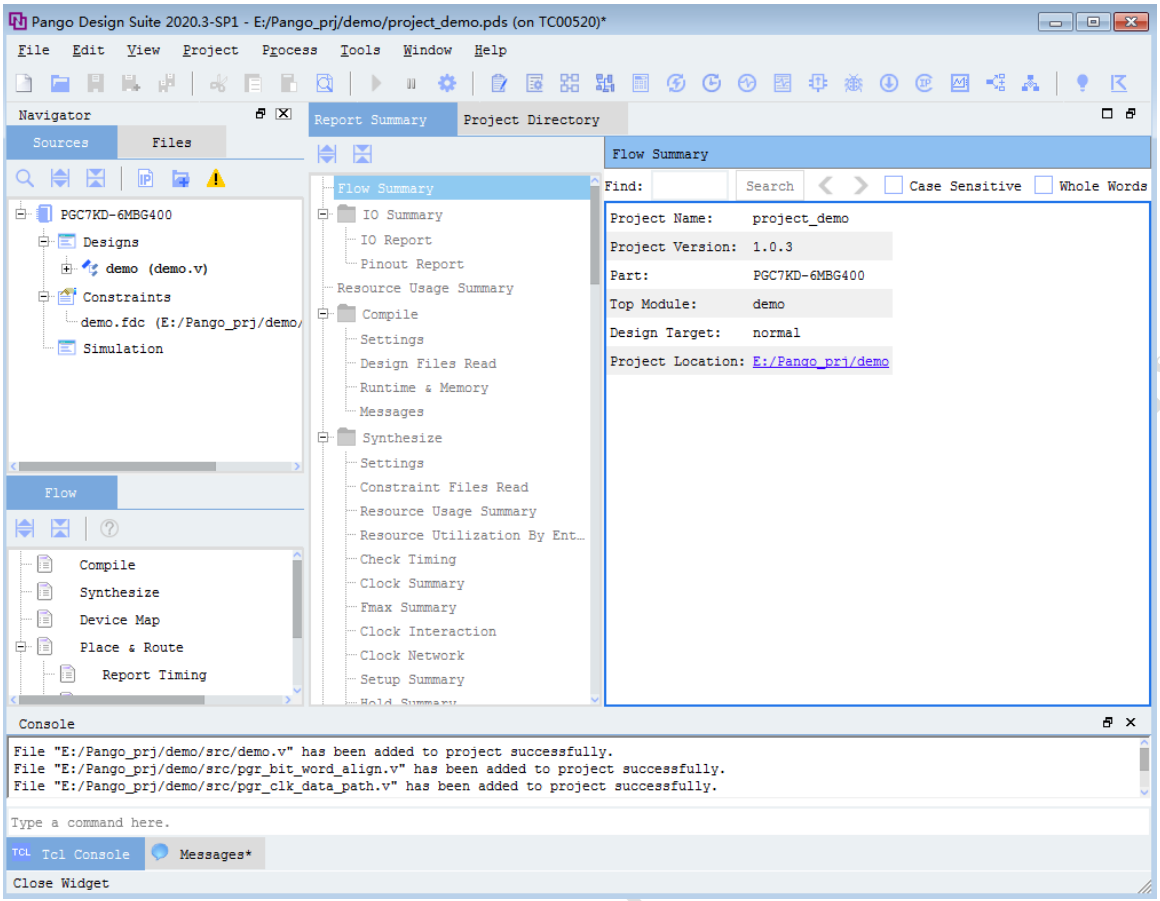


Figure2-17

2.2 Open Project

PDS records the projects that users have opened. Click [File > Recent Projects] to open previously used projects.

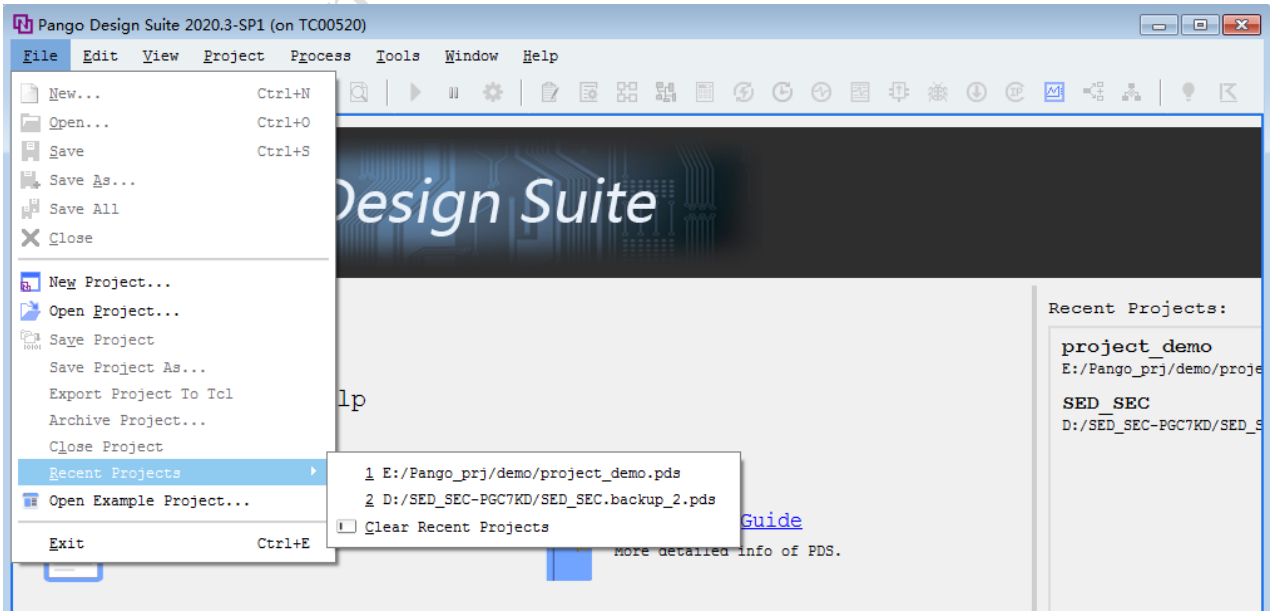


Figure2-18

Users can also select ".pds" or ".prj" files through [Open Project] to open existing projects.

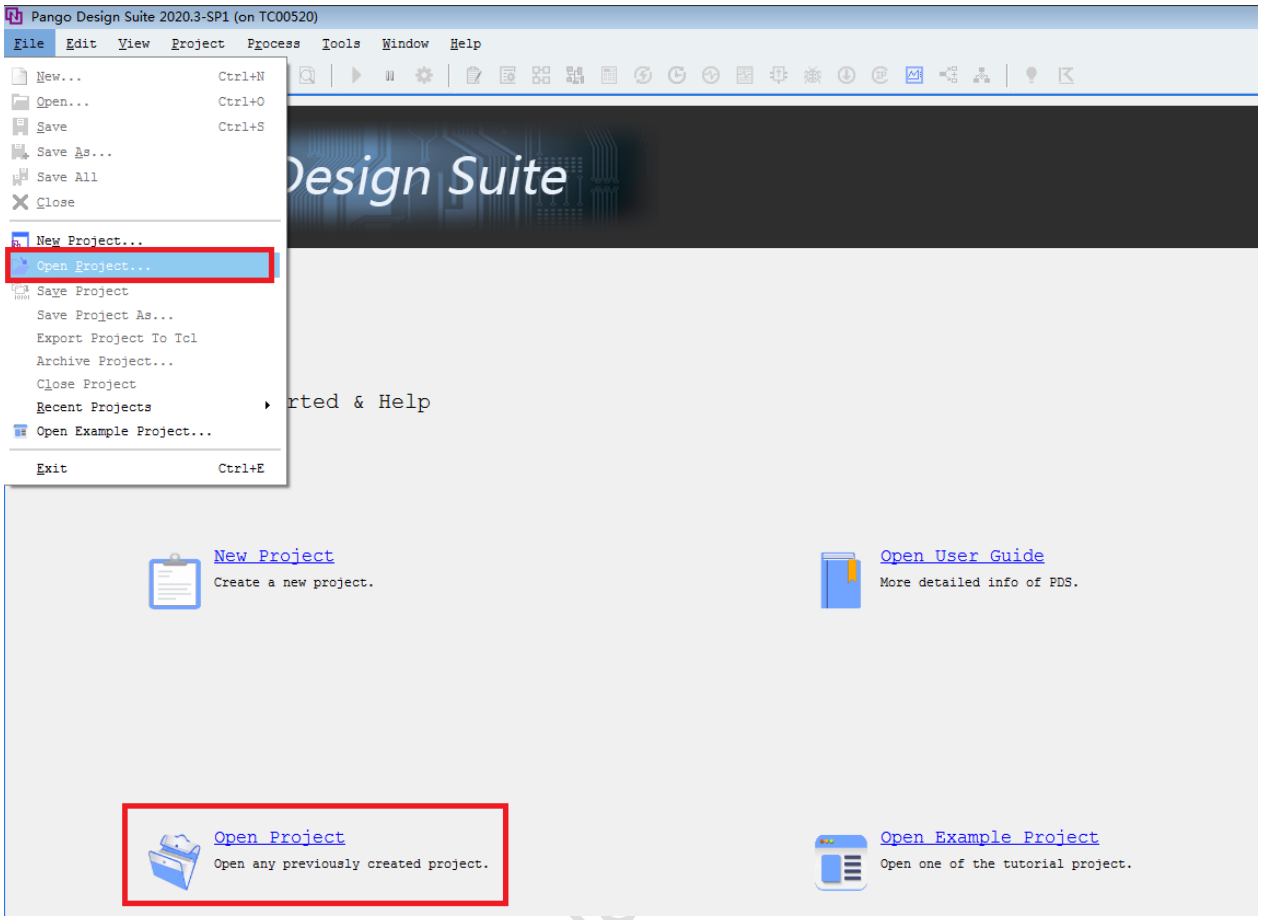


Figure2-19

The default access path is the path where the project was last closed;

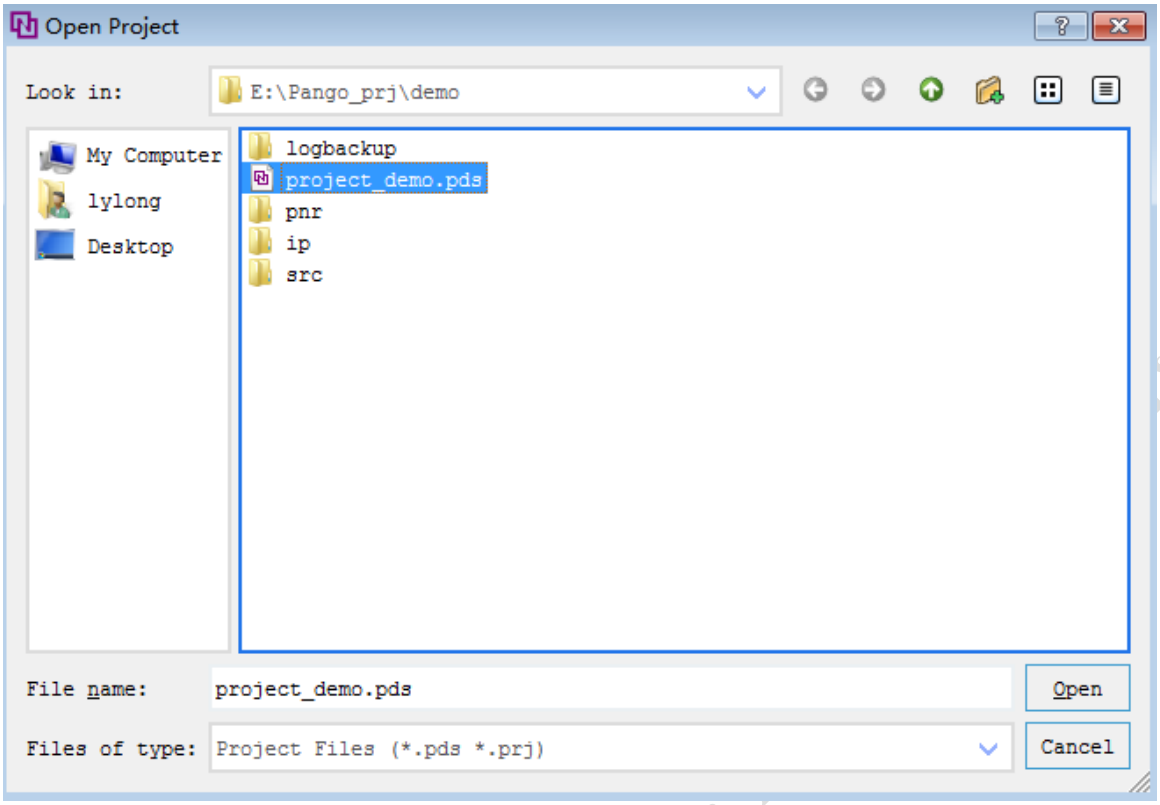


Figure2-20

When users open a project file created with an older version of PDS using a new version, they will be prompted to upgrade the original project file, as shown in the figure below. If "No" is selected, the project will fail to open; if "Yes" is selected, the new version of PDS will first back up the original project file, creating project_demo.backup_1.pds, and then upgrade the original project file project_demo.pds to a file that can be opened by the new version of PDS. Among them, project_demo.backup_1.pds can be opened with the old version of PDS, but project_demo.pds can no longer be opened with the old version of PDS.

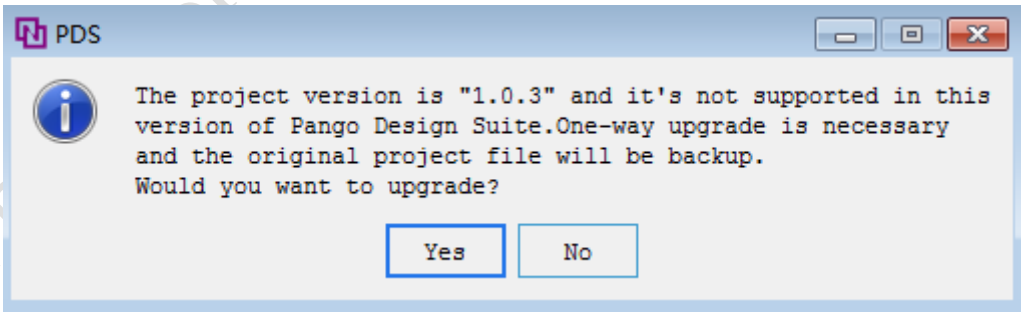


Figure2-21

2.3 Export tcl Script

PDS supports the execution of various processes of the software with tcl scripts, which can reduce user interface operations.

Click [File > Export Project to Tcl], as shown in the figure below.

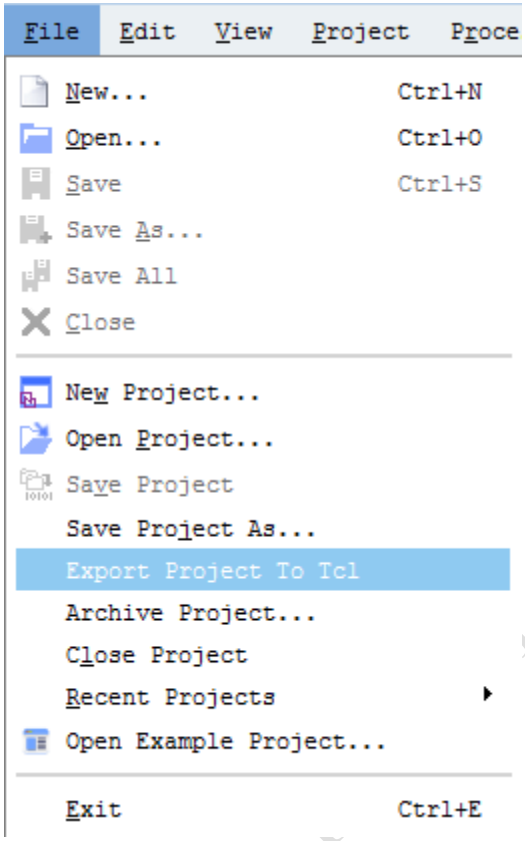


Figure2-22

An export.tcl file is generated in the project directory, which records the commands from the last interface operation.

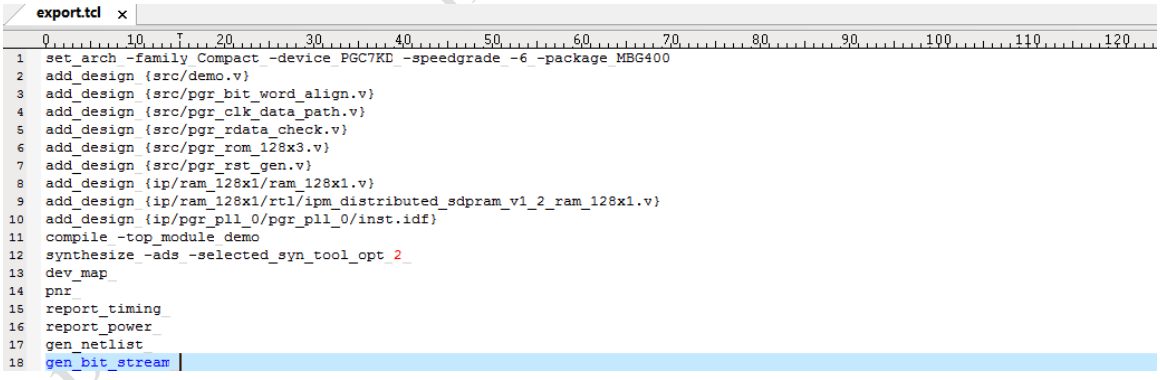


Figure2-23

The steps to execute the export.tcl file are:

1. Right-click in the blank area of the project directory while holding "Shift" to bring up the menu, and select [Open command window here];
2. A cmd window appears as shown in the figure below. Drag and drop pds_shell.exe into the window, and then press "Enter"; pds_shell.exe is located in the "\bin" folder of the PDS installation directory;

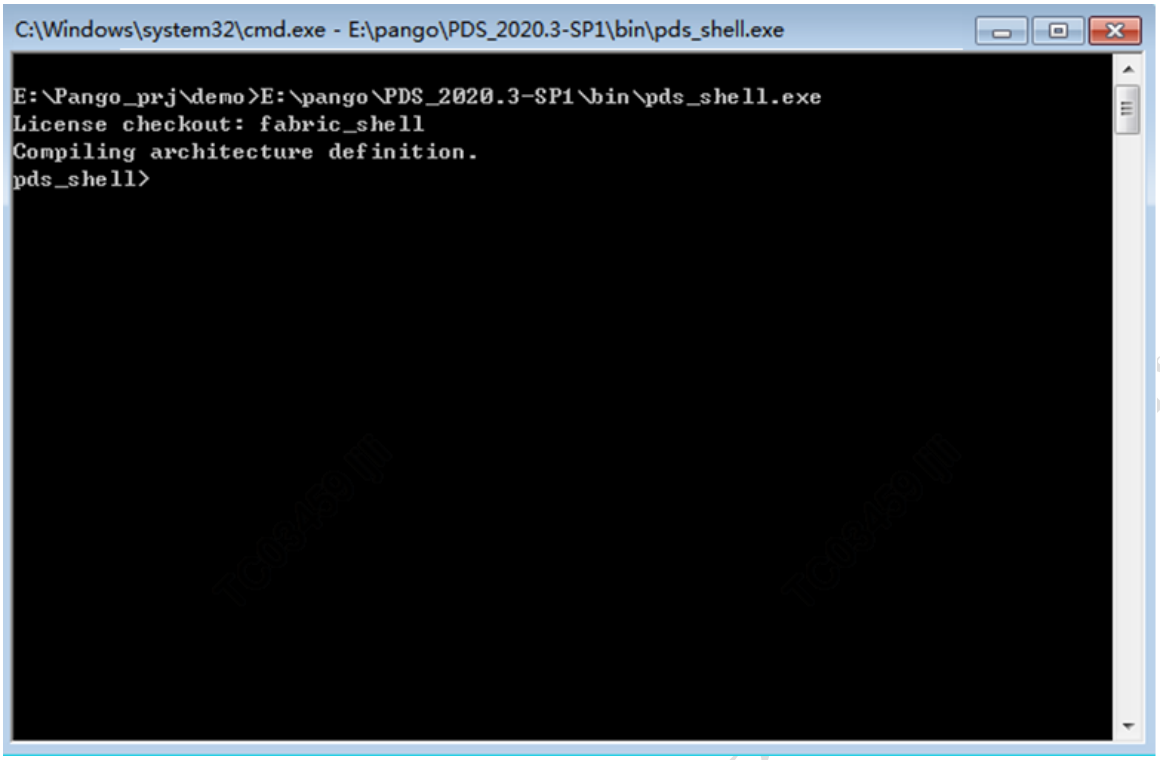


Figure2-24

3. Enter "source export.tcl" in the command line to execute the process in the tcl file.

2.4 Compress project

PDS can compress the project to the maximum extent, click [File > Archive Project], as shown in the figure below.

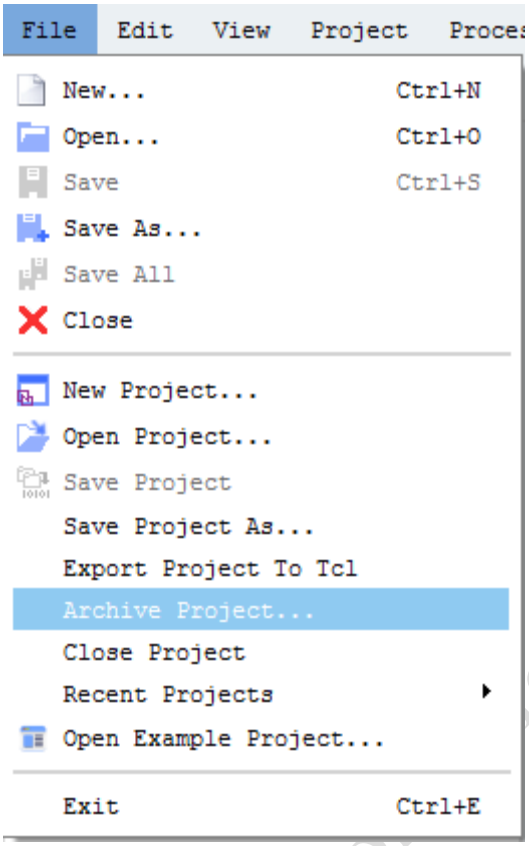


Figure2-25

In the pop-up box, define the name of the compressed file and edit the location; when [Include run results] is checked, it compresses all the result files of the current project run, avoiding repeated operations after decompression; if unchecked, only the minimum project is compressed.

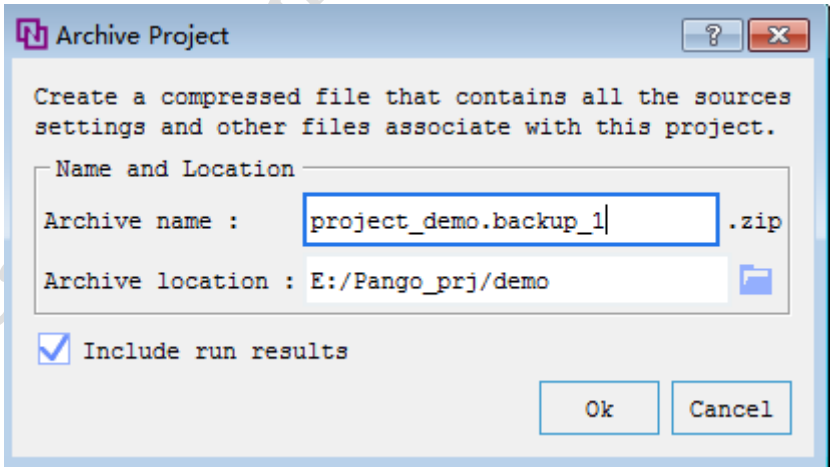


Figure2-26

2.5 File Management

2.5.1 Create and Add Design Files

Click [Project > Add Source];

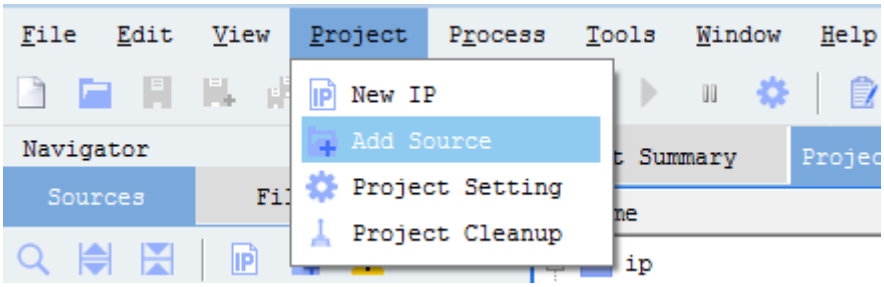


Figure2-27

Or right-click within the Sources or Files area of Navigator and select [Add Source] in the menu;

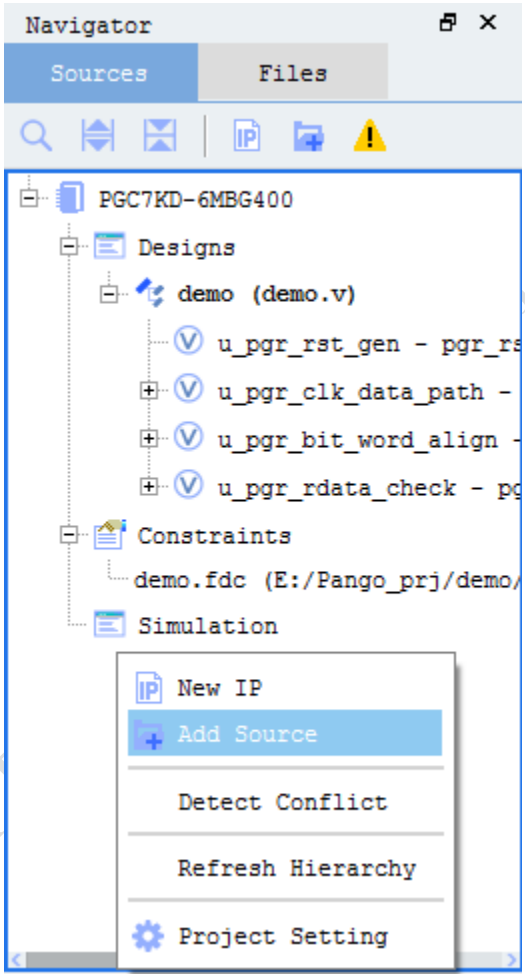


Figure2-28

Or click the shortcut in the Navigator toolbar;

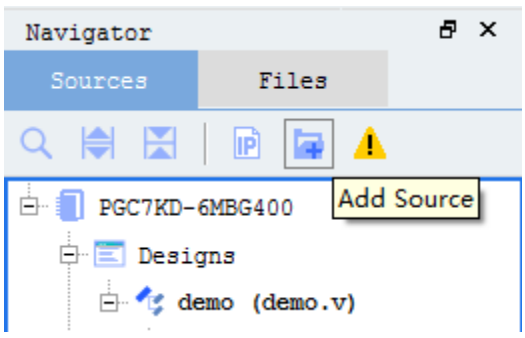


Figure2-29

Select [Add or create design sources], and then click [Next];

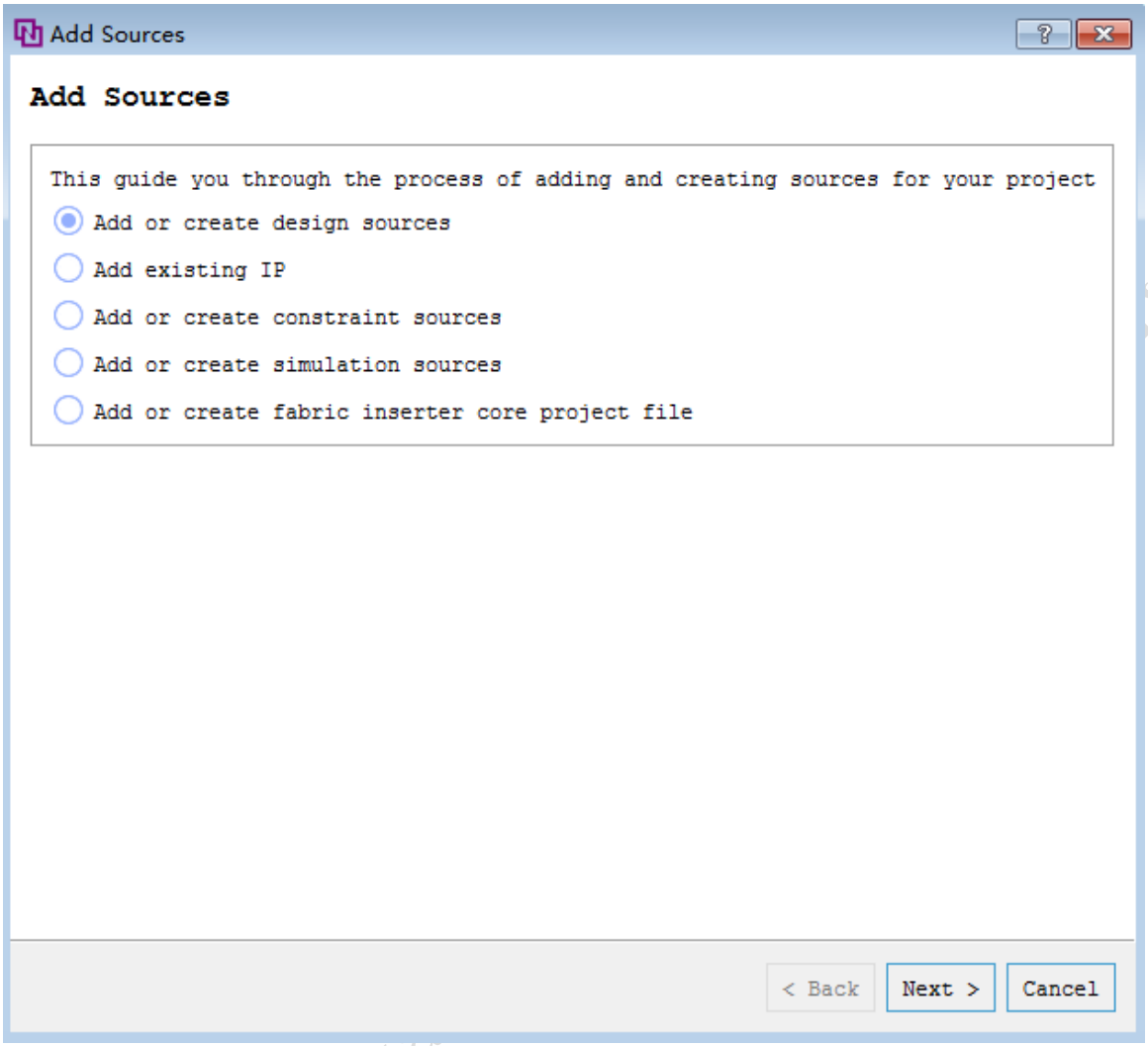


Figure2-30

Click [Add Files] to add existing design files; click [Create File] to create a new file, enter the name and location of the new file in the pop-up box, and then click [OK];

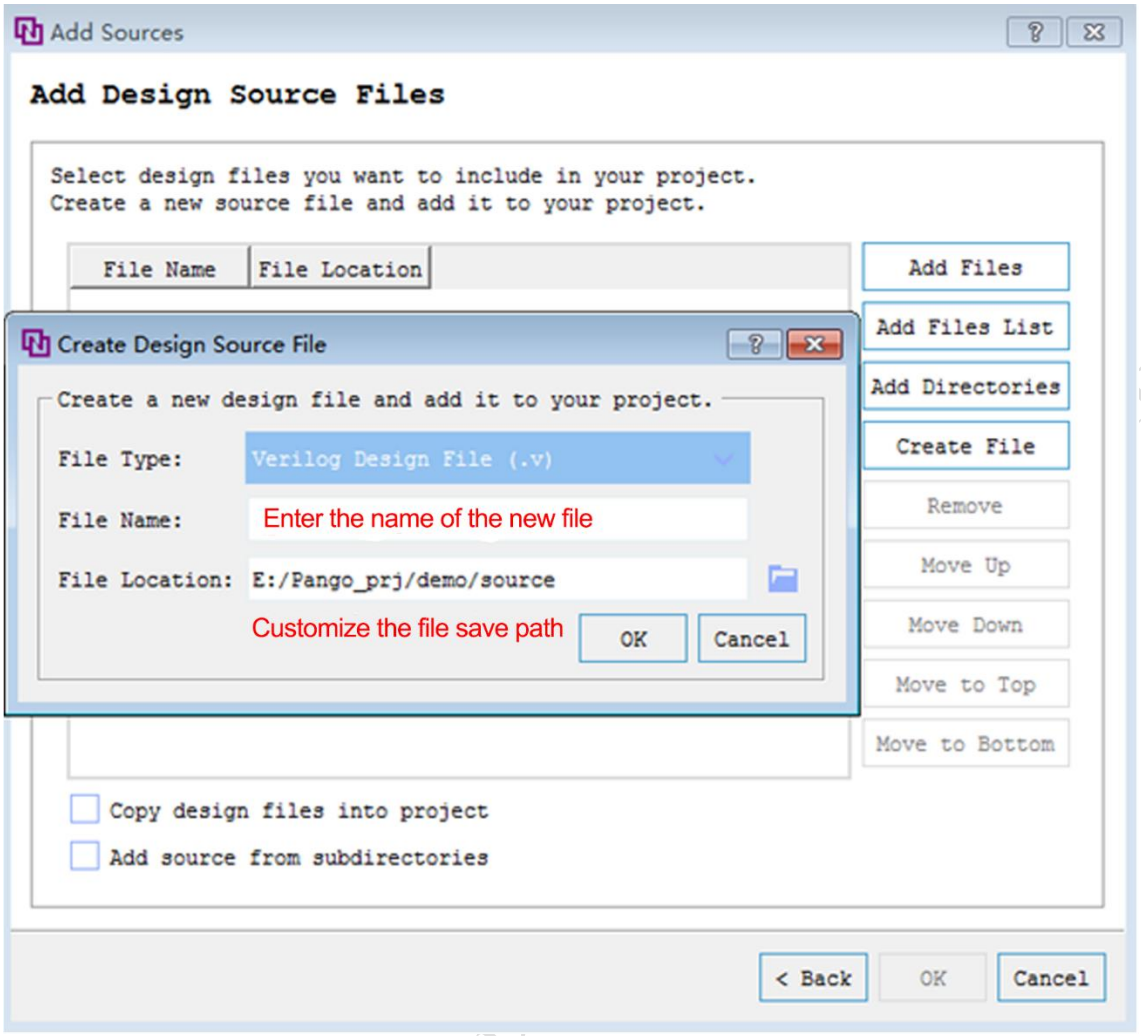


Figure2-31

The following file types are supported:

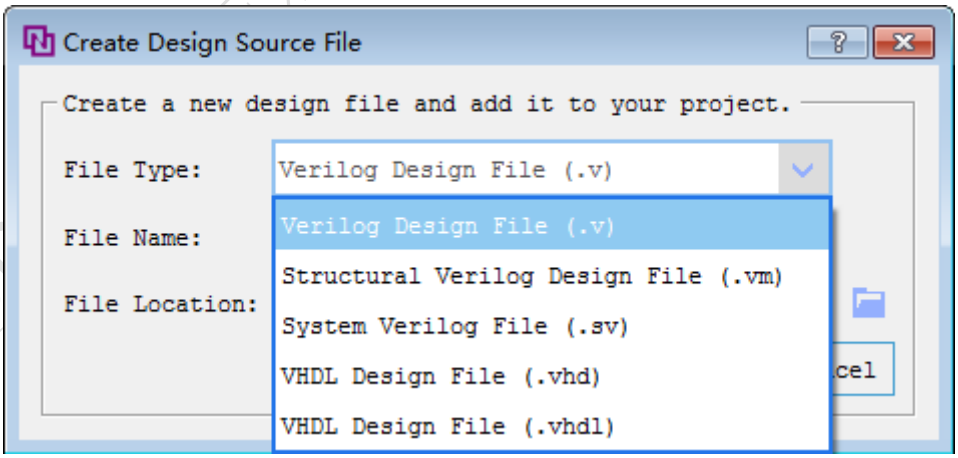


Figure2-32

After creating the design files, click [OK] to define signals, and then click [OK];

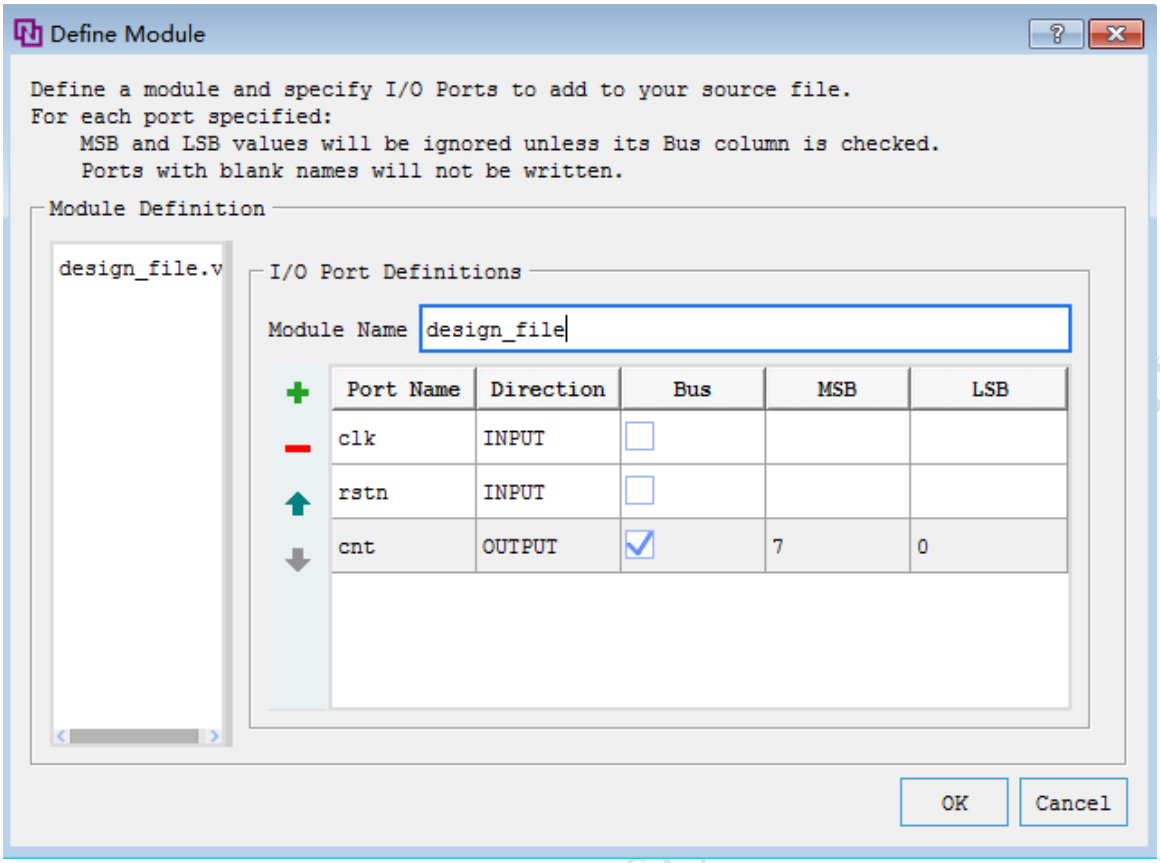


Figure2-33

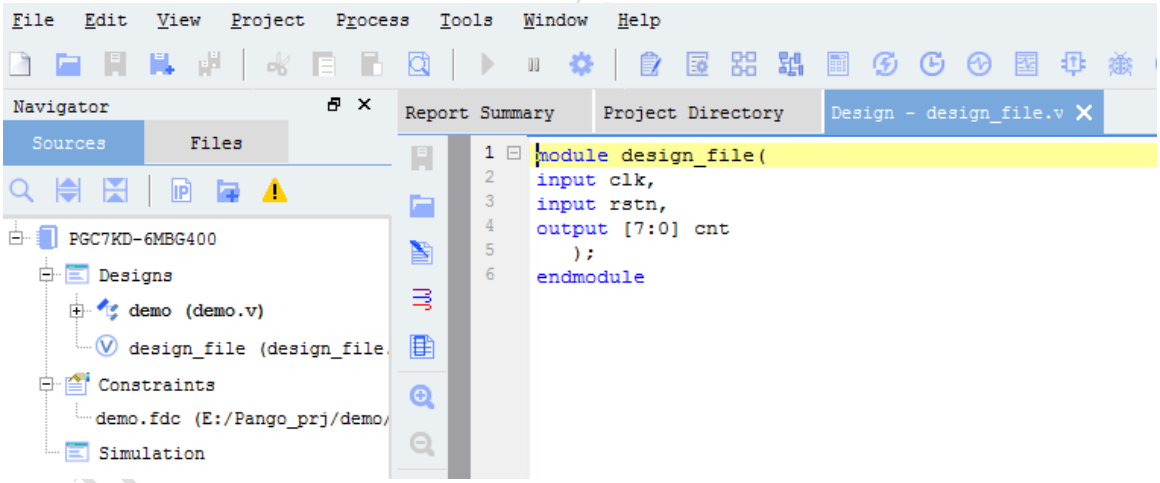


Figure2-34

2.5.2 Remove Files

Right-click on the file to be removed in Navigator, and select [Remove Source] from the menu, as shown in the figure below;

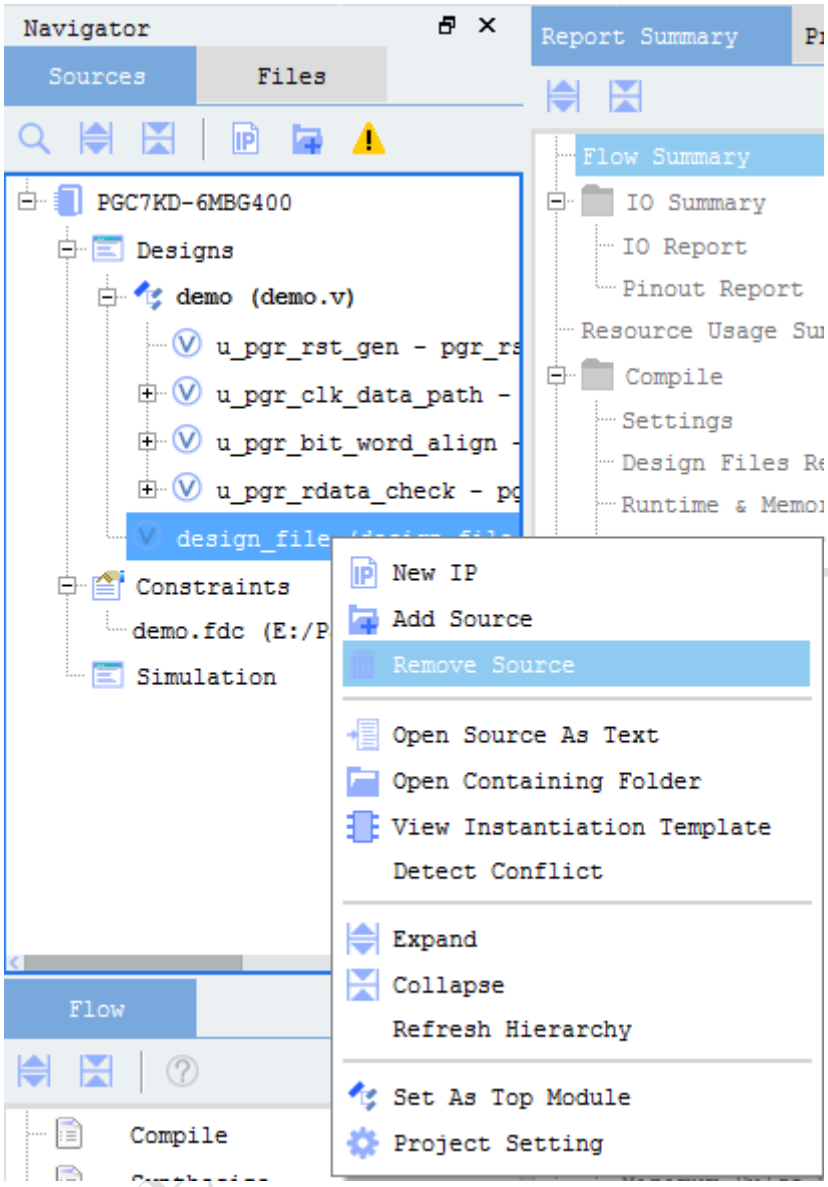


Figure2-35

2.5.3 Edit Files

PDS Editor provides many convenient buttons for editing files. The figure below shows the vertical toolbar of the Editor, located on the left side of the file editing area.

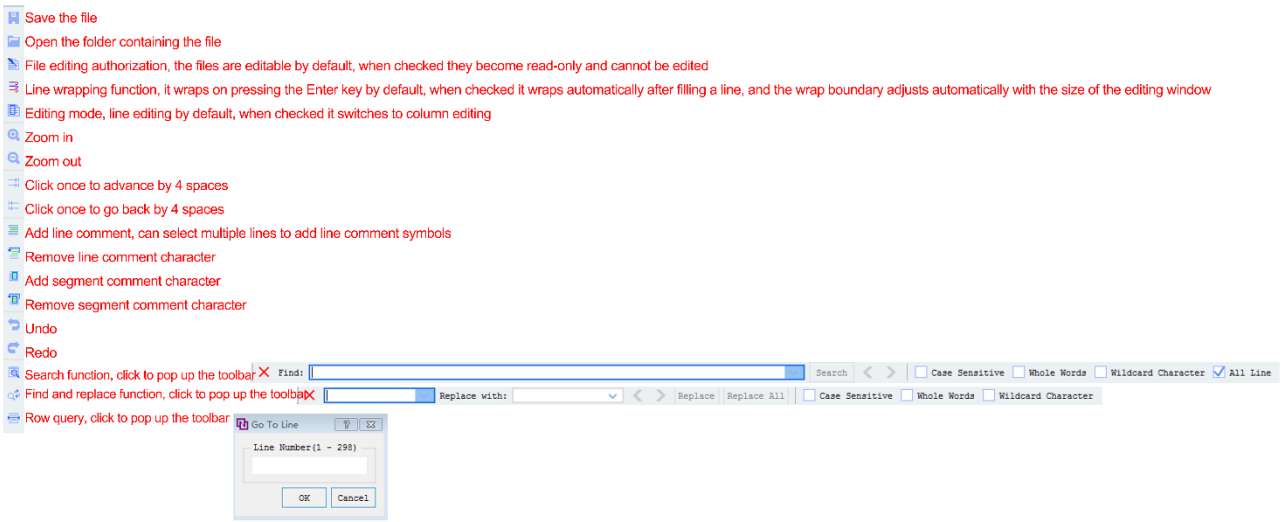


Figure2-36

Chapter 3 IP Complier

IP Complier is a graphical interactive design interface for creating IP cores. Users can configure the required IPs in the IP Complier and automatically generate the corresponding IP modules. The Module IPs currently supported by PGC devices include Distributed RAM, DRM, and PLL.

Click [Tools > IP Compiler] to open the IP Complier;

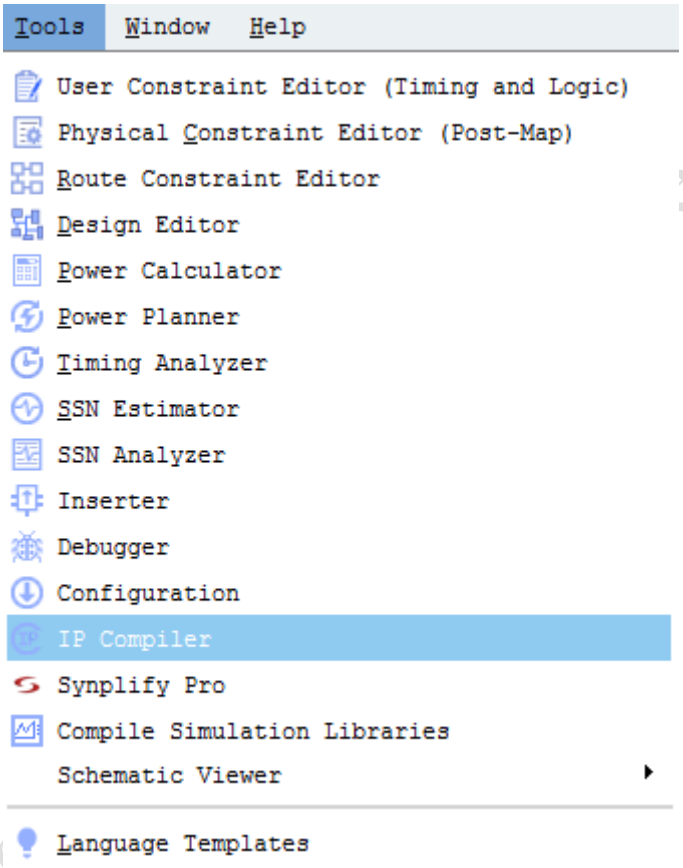


Figure3-1

Or click the IP Compiler icon in the toolbar to open the IP Complier.

Figure3-2

The interface of the IP Complier is as shown in the figure below.

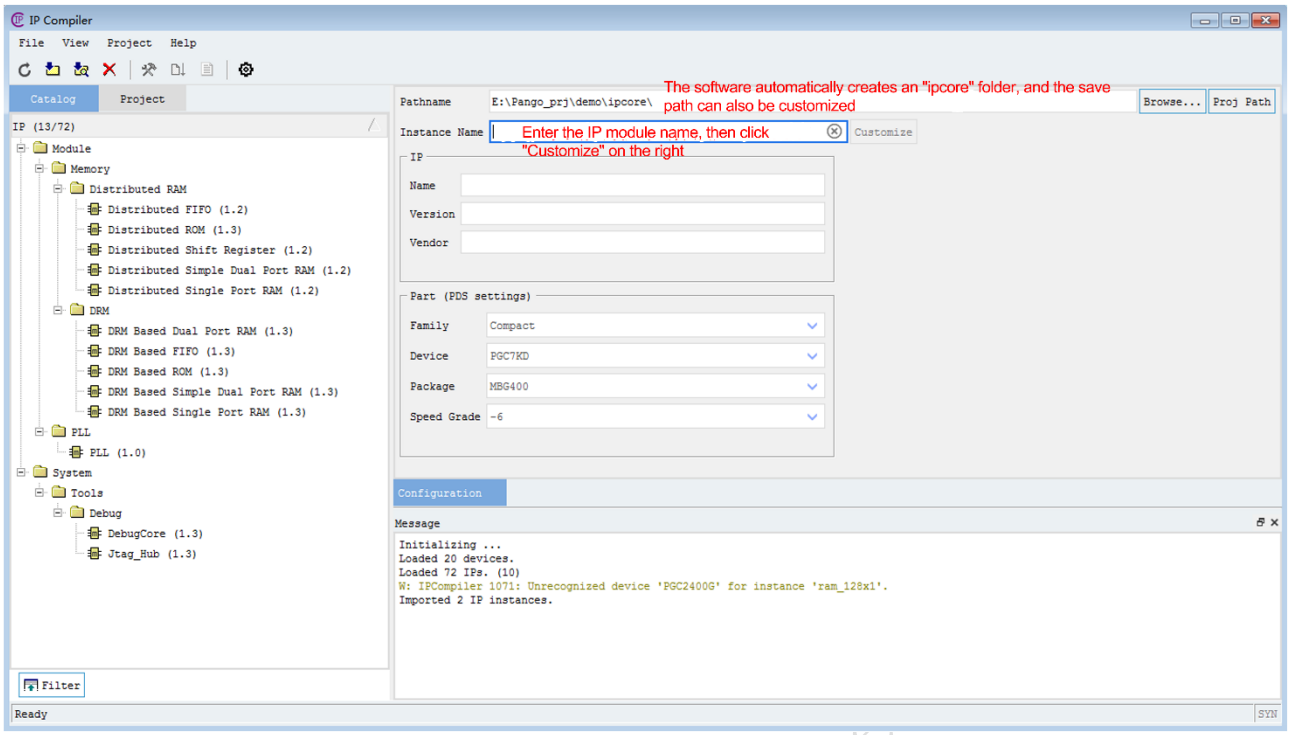


Figure3-3

3.1 Distributed RAM

Distributed RAM is implemented by CLM. CLM is divided into CLMA and CLMS. For Compa family, each CLM contains 4 LUTs and 6 FFs.

Distributed RAM includes Distributed FIFO, Distributed ROM, Distributed Shift Register, Distributed Simple Dual Port RAM, and Distributed Single Port RAM.

3.1.1 Distributed FIFO

Distributed FIFO is implemented by CLMS. The Distributed FIFO has the following features:

1. Synchronous and asynchronous FIFO
2. Synchronous and asynchronous reset
3. Output register
4. Almostfull\Almostempty signals
5. Reading and writing water level

3.1.1.1 Create a Distributed FIFO Module

A new Distributed FIFO module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a Distributed FIFO module.

In the IP Complier interface, select [Distributed FIFO] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

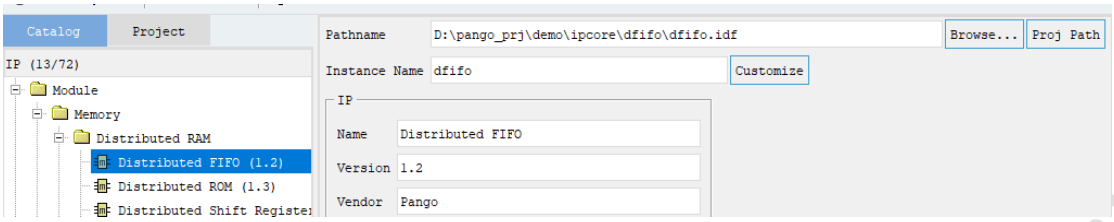


Figure3-4

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

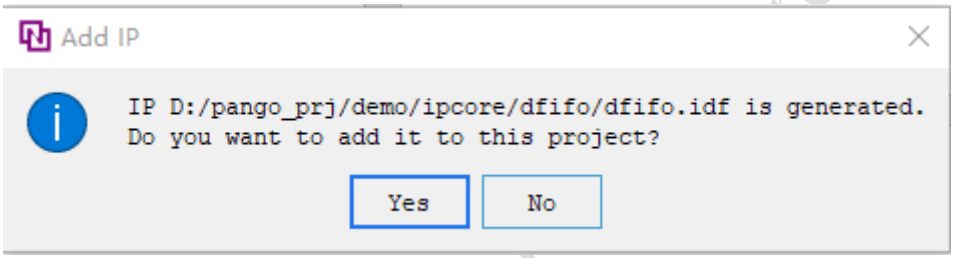


Figure3-5

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the Distributed FIFO configuration interface, as shown in the figure below.

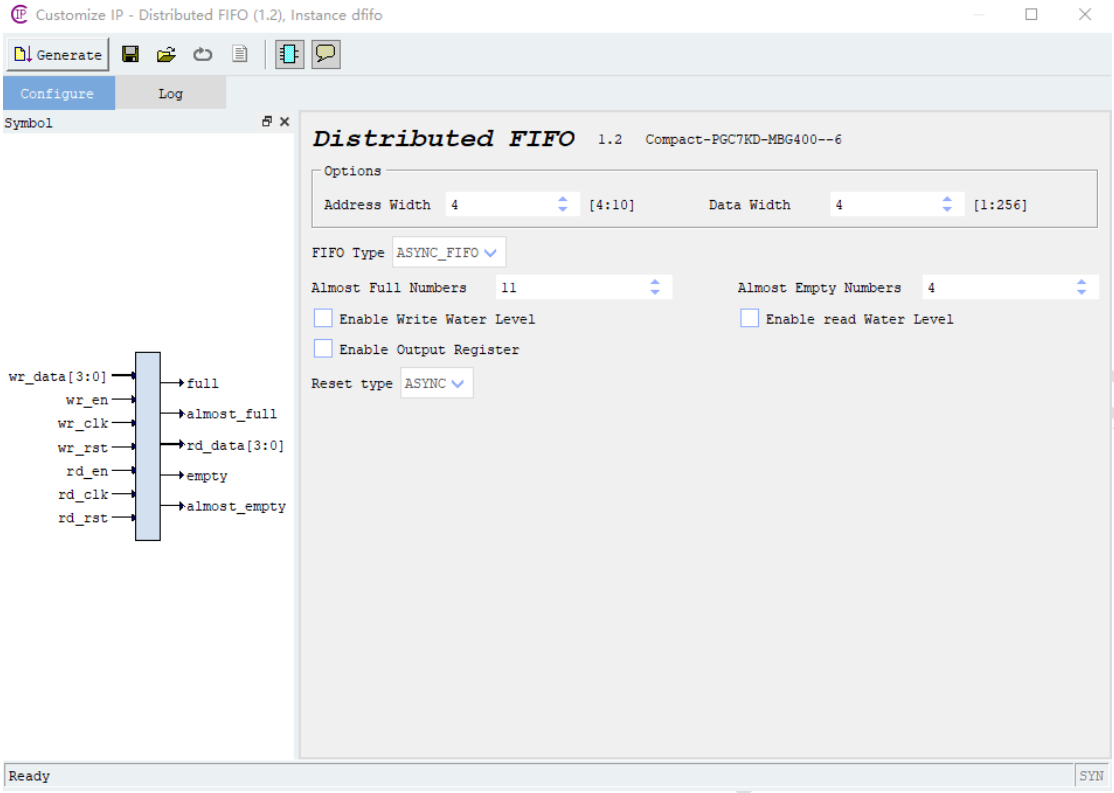


Figure3-6

After completing the settings, click [Generate] in the upper-left corner of the interface to generate the Distributed FIFO module. users can then see that the module has been added to the project, as shown in the figure below.

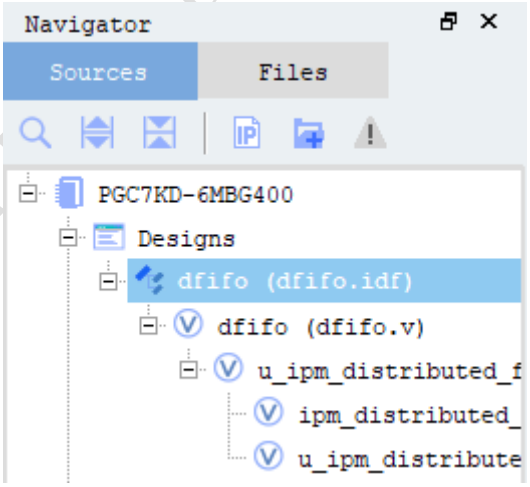
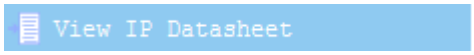
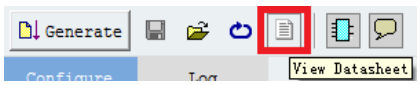


Figure3-7

To reset the parameters of the Distributed FIFO, simply double-click the IP file "dfifo.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Distributed RAM IP User Guide". To open it, right-click "dfifo.idf" and select



from the drop-down menu; or select [View Datasheet]



in the upper-left corner of the IP configuration interface.

3.1.1.2 Instantiate Distributed FIFO Module

Right-click the top-level file of the Distributed FIFO module and select [View Instantiation Template] to see the Verilog instantiation template for the Distributed FIFO module, as shown in the figure below.

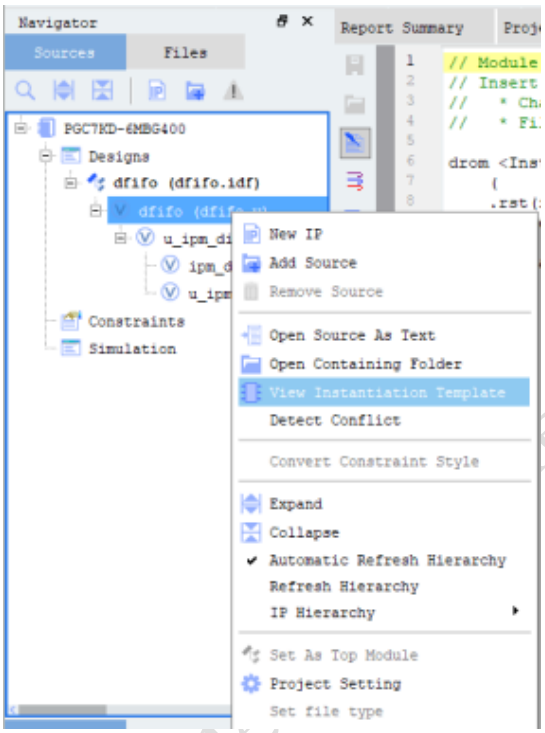


Figure3-8

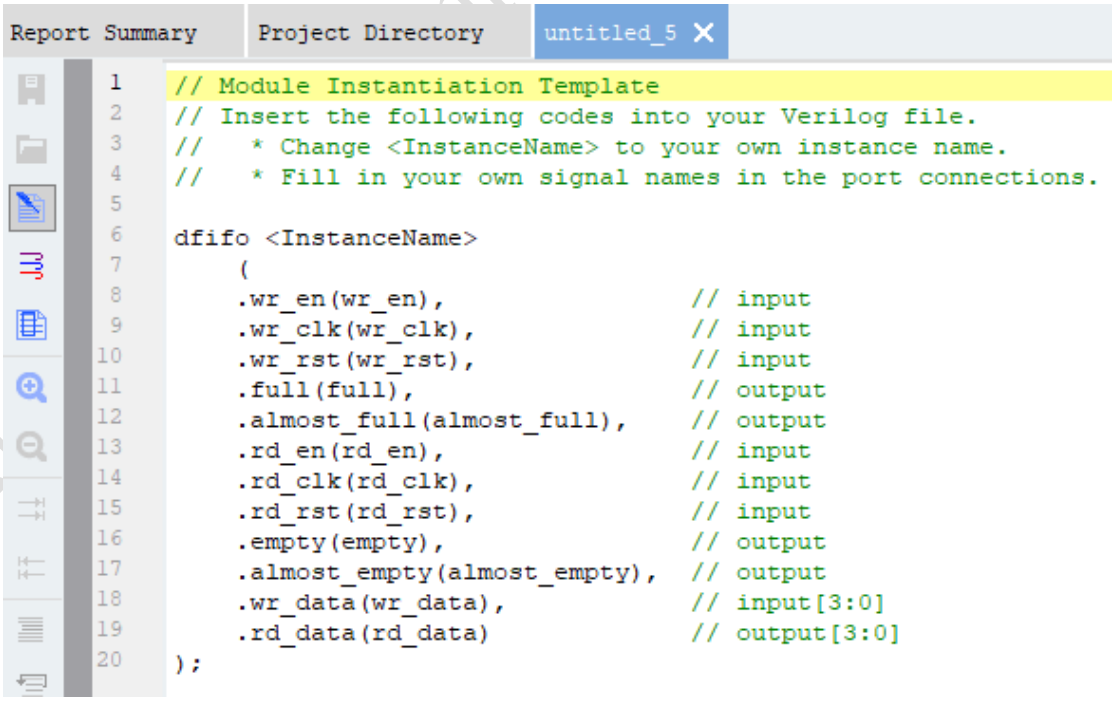


Figure3-9

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.1.2 Distributed ROM

Distributed ROM can be implemented by CLMA or CLMS. The Distributed FIFO has the following features:

Synchronous and asynchronous reset

1. Output register
2. Input initialization files. The initialization file is a .dat file, supporting binary and hexadecimal formats. Each line in the file stores the data for one address, with no spaces allowed after each data entry, and no other symbols within the file. The content of the DAT file is shown in the figure below.

```
1  00000032$
2  FFFD0032$
3  FFFB0032$
4  FFF90032$
5  FFF70032$
6  FFF60033$
7  FFF50033$
8  FFF20034$
9  FFF00034$
10 FFEE0034$
11 FFEC0034$
12 FFEA0036$
13 FFE90037$
14 FFE60037$
15 FFE40038$
16 FFE20038$
```

Figure3-10

3.1.2.1 Create a Distributed ROM Module

A new Distributed ROM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a Distributed ROM module.

In the IP Complier interface, select [Distributed ROM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

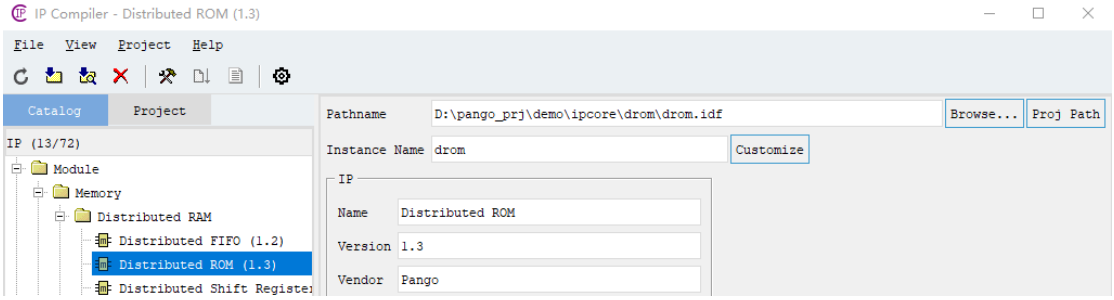


Figure3-11

After completing the input, click [Customize], and a prompt will appear asking if there is an initialization file. If an ROM initialization file already exists, click [Next].

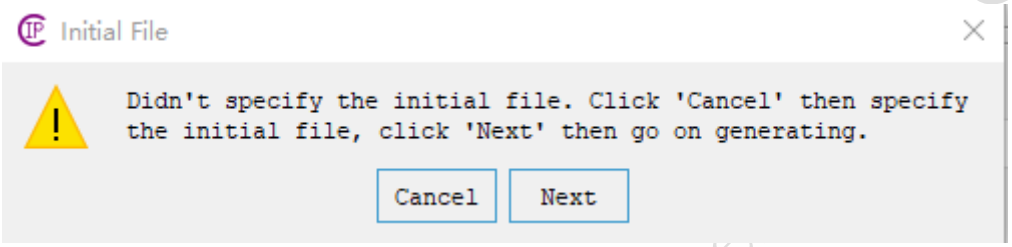


Figure3-12

Then, a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

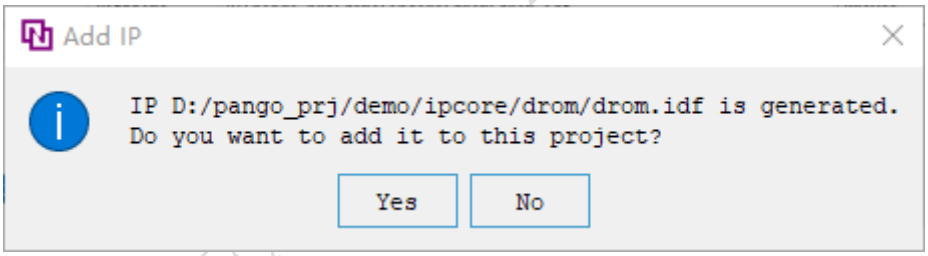


Figure3-13

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the Distributed ROM configuration interface, as shown in the figure below.

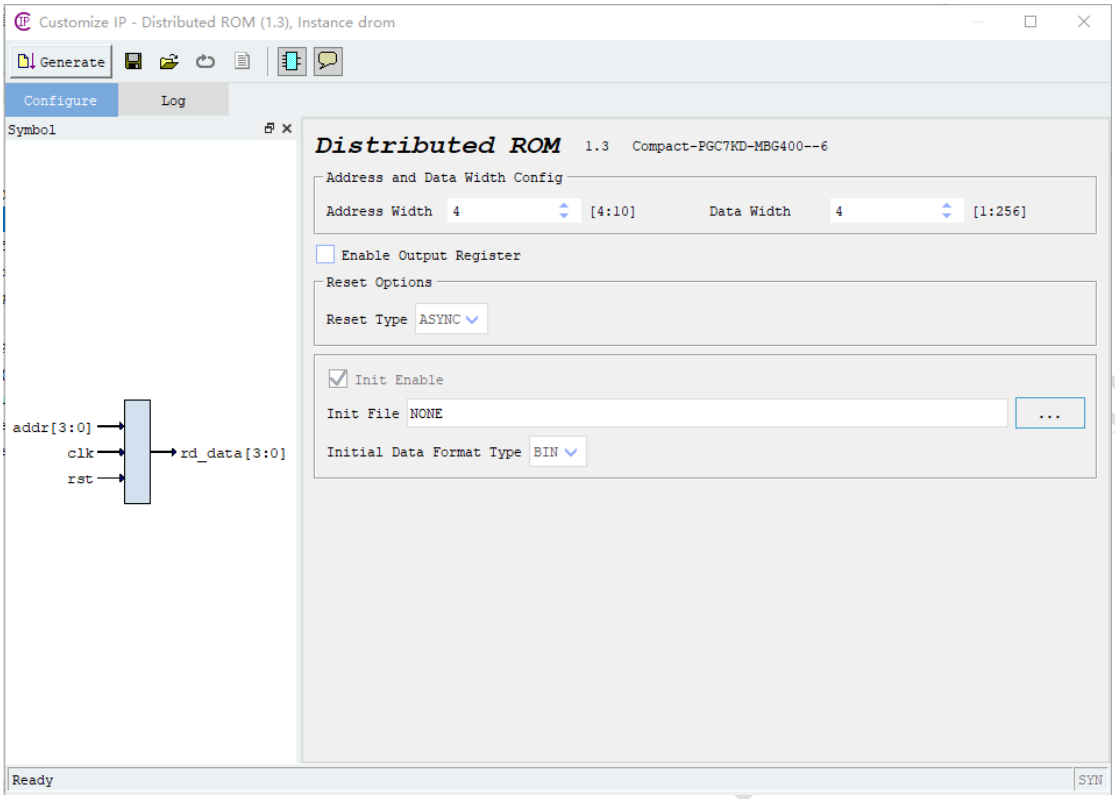


Figure3-14

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the Distributed ROM module, which can then be seen added to the project as shown in the figure below.

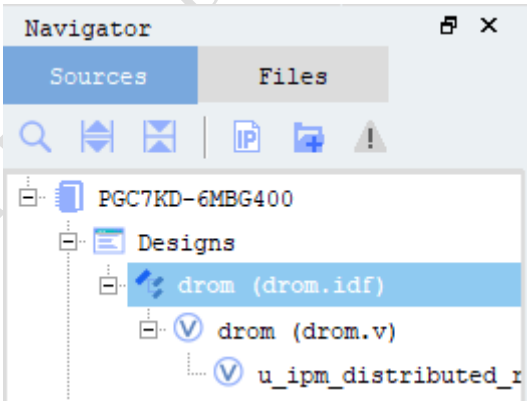
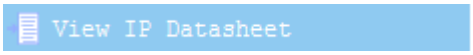
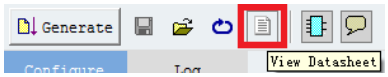


Figure3-15

To reset the parameters of the Distributed ROM, simply double-click the IP file "drom.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Distributed RAM IP User Guide". To open it, right-click "drom.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.1.2.2 Instantiate Distributed ROM Module

Right-click the top-level file of the Distributed ROM module and select [View Instantiation Template] to see the Verilog instantiation template for the Distributed ROM module, as shown in the figure below.

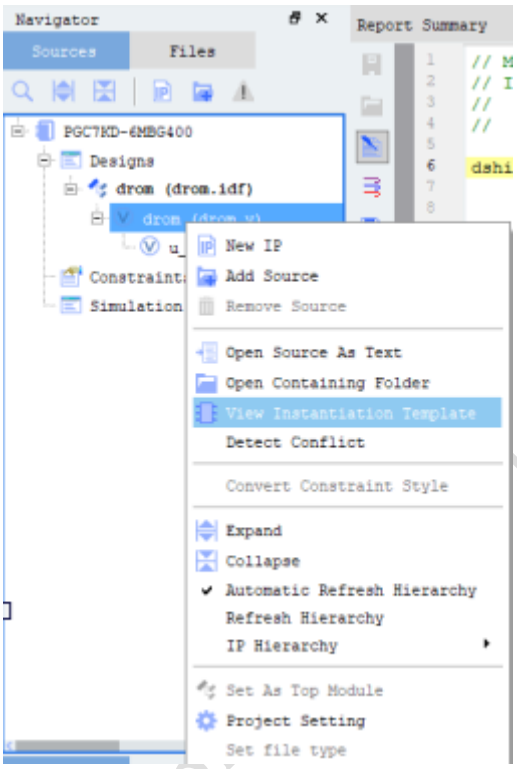


Figure3-16

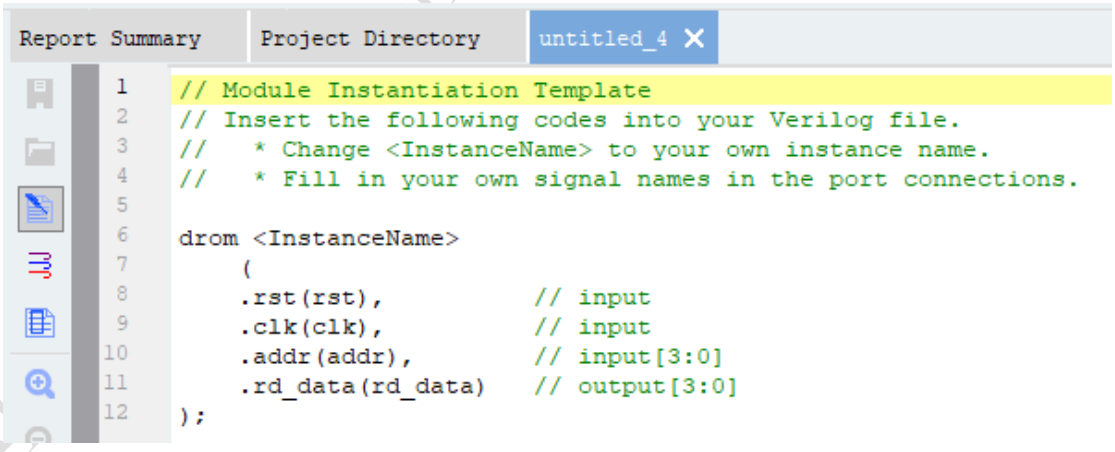


Figure3-17

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.1.3 Distributed Shift Register

Distributed Shift Register is implemented by FFs in CLM. The Distributed FIFO has the following

features:

1. Dynamic and static shifters
2. Synchronous and asynchronous reset

3.1.3.1 Create Distributed Shift Register Module

A new Distributed Shift Register module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a Distributed Shift Register module.

In the IP Complier interface, select [Distributed Shift Register] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

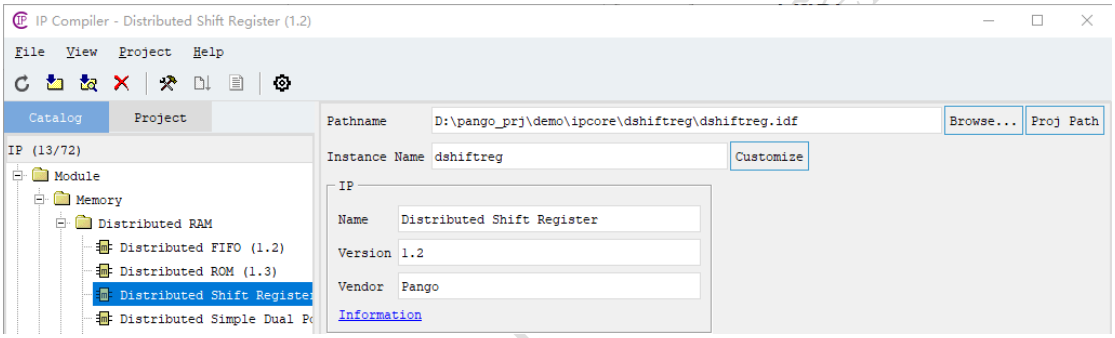


Figure3-18

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

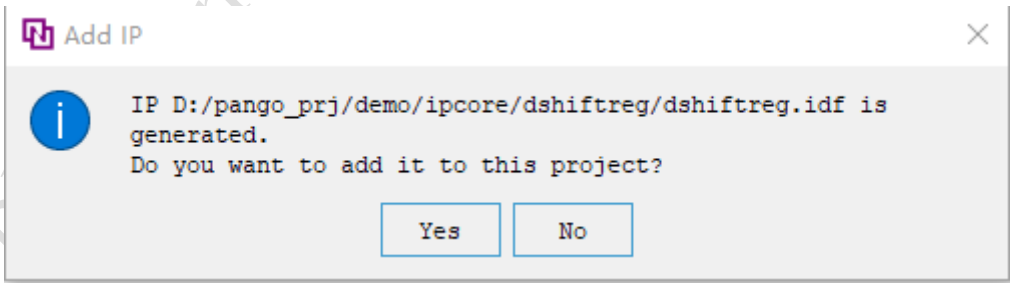


Figure3-19

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then set the corresponding parameters in the configuration interface of the Distributed Shift Register, as shown in the figure below.

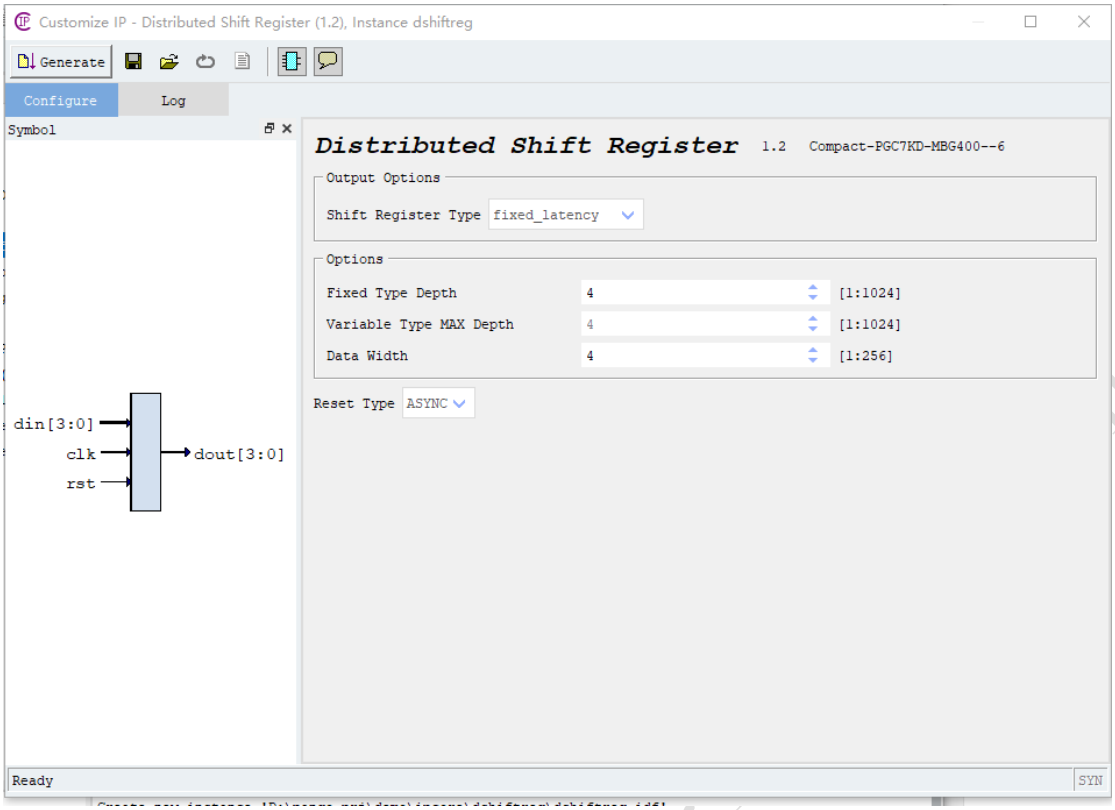


Figure3-20

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the Distributed Shift Register module, which can then be seen added to the project as shown in the figure below.

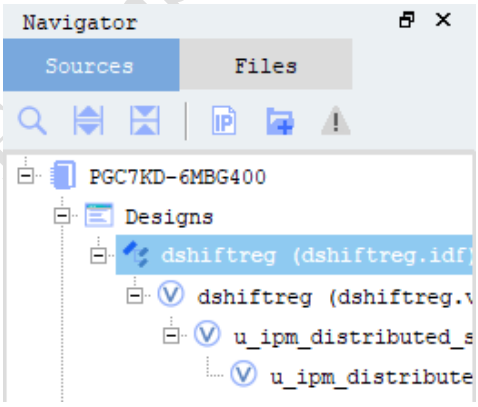
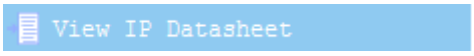
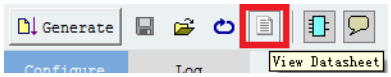


Figure3-21

To reset the parameters of the Distributed Shift Register, simply double-click the IP file "dshiftreg.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Distributed RAM IP User Guide". To open it, right-click "dshiftreg.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.1.3.2 Instantiate Distributed Shift Register Module

Right-click the top-level file of the Distributed Shift Register module and select [View Instantiation Template] to see the Verilog instantiation template for the Distributed Shift Register module, as shown in the figure below.

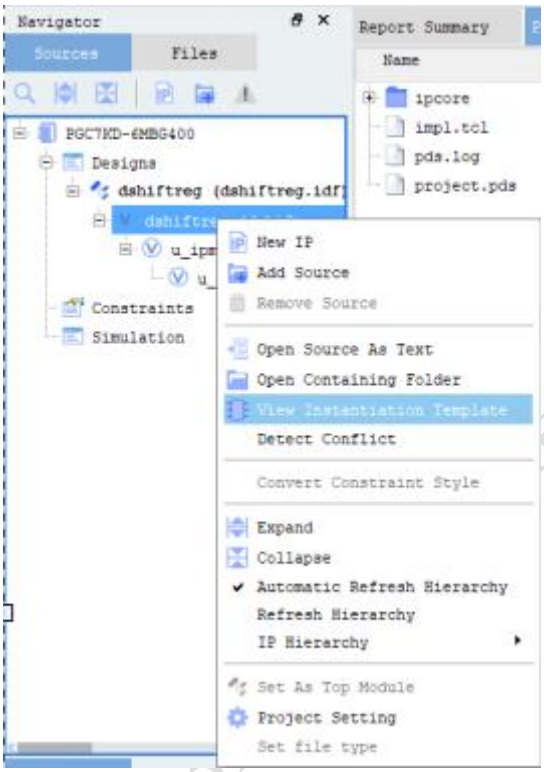


Figure3-22

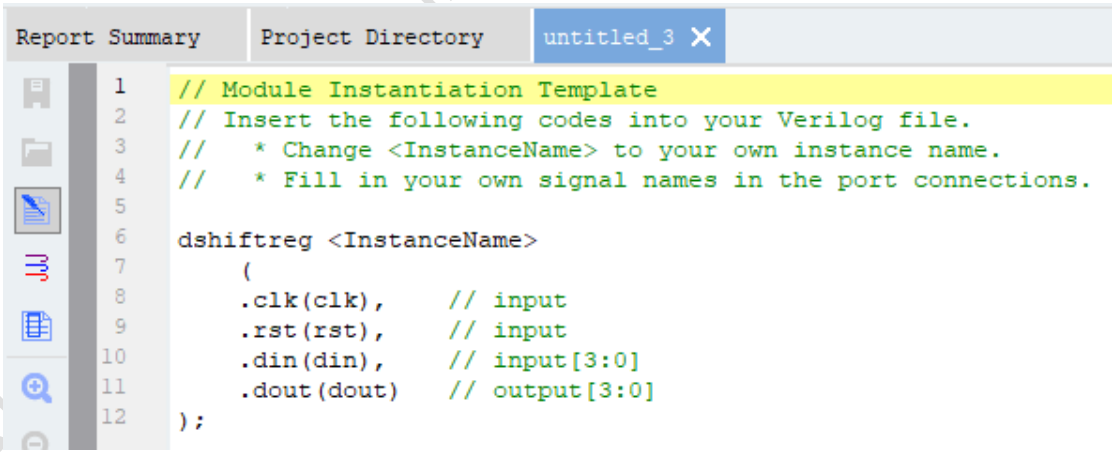


Figure3-23

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.1.4 Distributed Simple Dual Port RAM

Distributed Simple Dual Port RAM is implemented by LUT5S in CLMS. Each LUT5S can implement a 16x1 Simple Dual Port RAM. The Distributed FIFO has the following features:

1. Synchronous and asynchronous reset
2. Read and write ports in different clock domains
3. Output register
4. Initialization using initialization files, which can be in binary or hexadecimal format

3.1.4.1 Create Distributed Simple Dual Port RAM Module

A new Distributed Simple Dual Port RAM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a Distributed Simple Dual Port RAM module.

In the IP Complier interface, select [Distributed Simple Dual Port RAM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

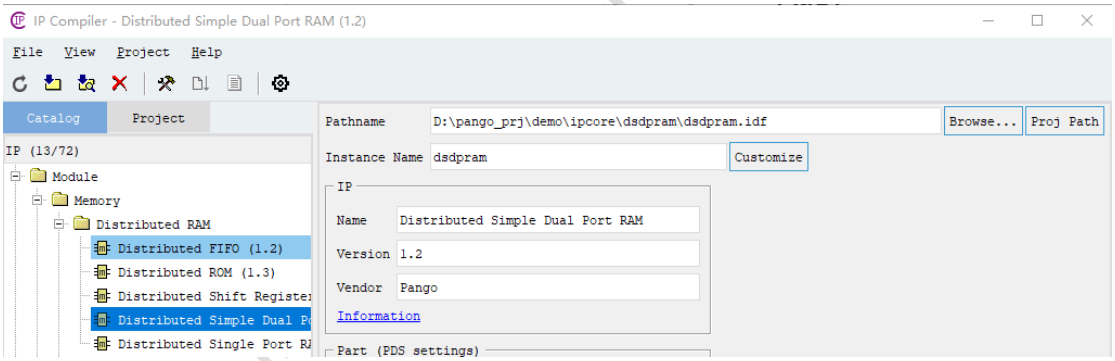


Figure3-24

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

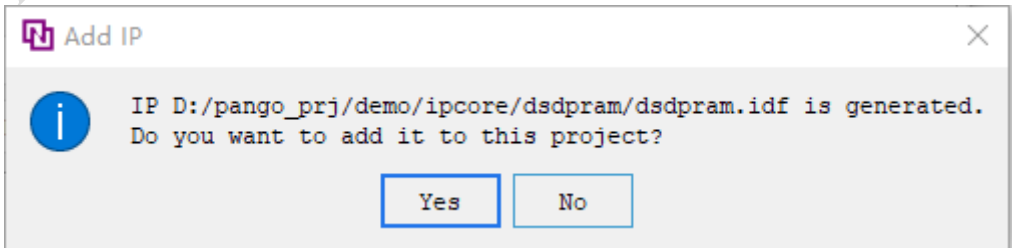


Figure3-25

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the Distributed Simple Dual Port RAM configuration

interface, as shown in the figure below.

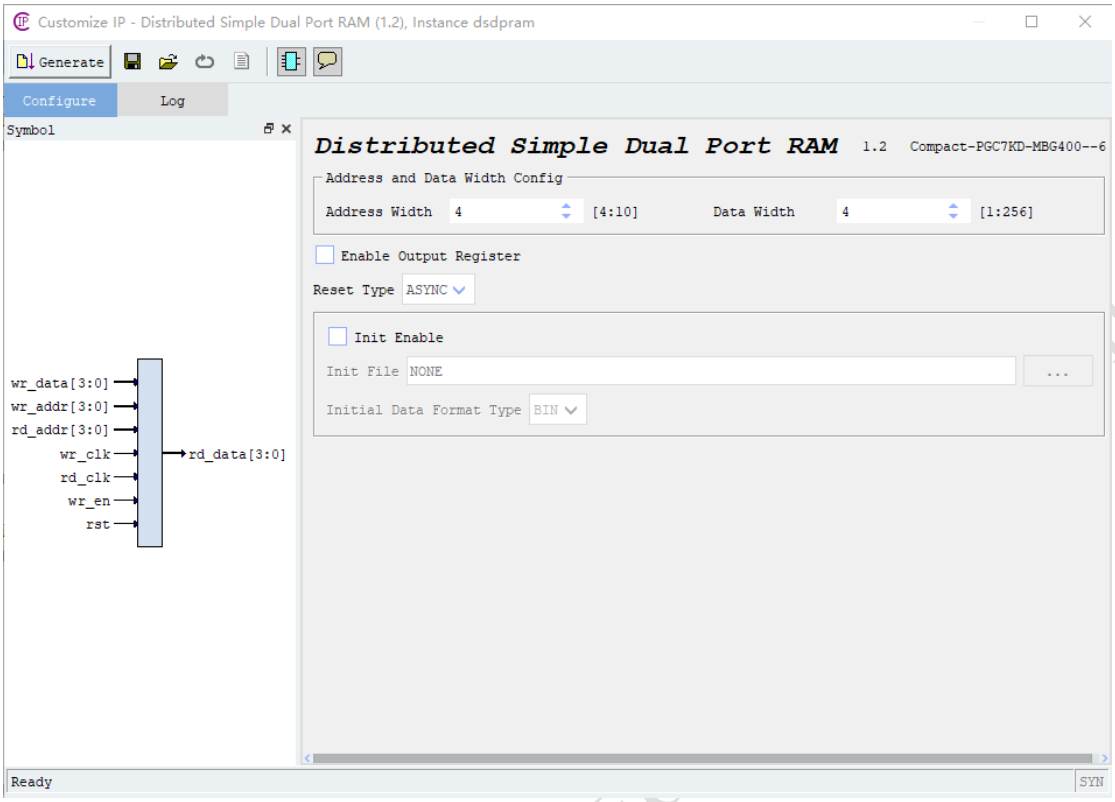


Figure3-26

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the Distributed Simple Dual Port RAM module, which can then be seen added to the project as shown in the figure below.

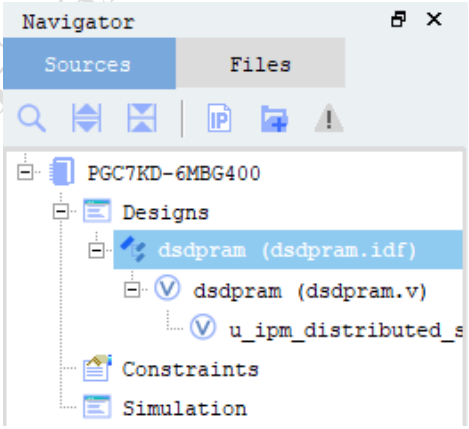
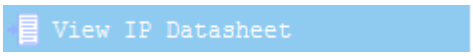


Figure3-27

To reset the parameters of the Distributed Simple Dual Port RAM, simply double-click the IP file "dsdpram.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Distributed RAM IP User Guide". To open it, right-click "dsdpram.idf" and select 

from the drop-down

menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.1.4.2 Instantiate Distributed Simple Dual Port RAM Module

Right-click the top-level file of the Distributed Simple Dual Port RAM module and select [View Instantiation Template] to see the Verilog instantiation template for the Distributed Simple Dual Port RAM module, as shown in the figure below.

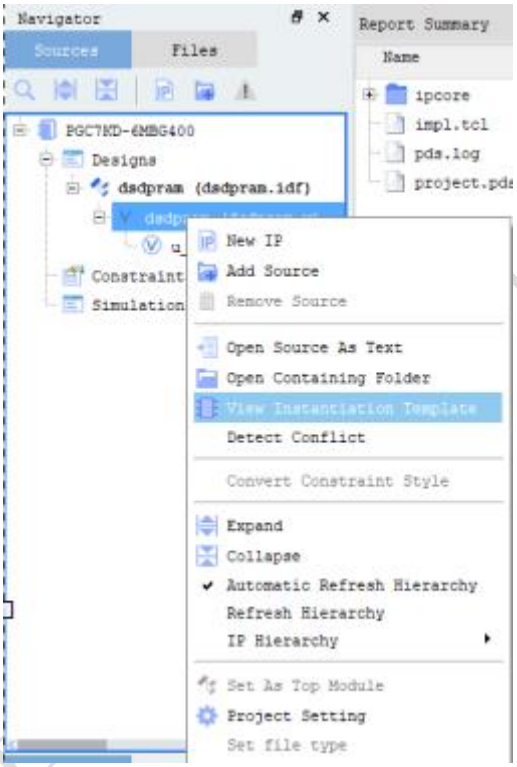


Figure3-28

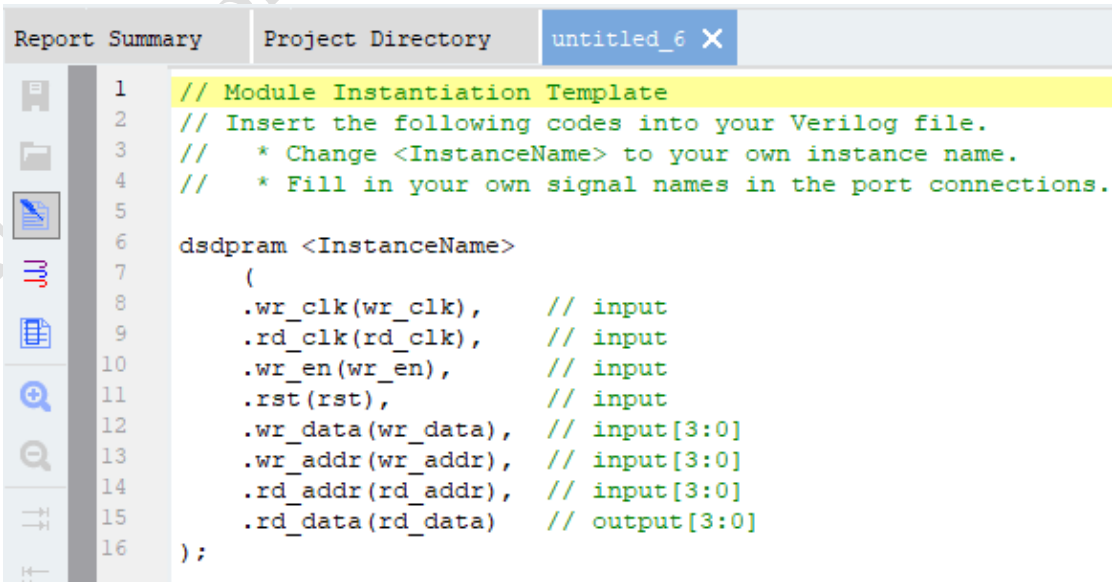


Figure3-29

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.1.5 Distributed Single Port RAM

Distributed Single Port RAM is implemented by LUT5S in CLMS. Each LUT5S can implement a 16x1 Single Port RAM. The Distributed FIFO has the following features:

1. Synchronous and asynchronous reset
2. Output register
3. Initialization using initialization files, which can be in binary or hexadecimal format

3.1.5.1 Create a new Distributed Single Port RAM module

A new Distributed Single Port RAM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a Distributed Single Port RAM module.

In the IP Complier interface, select [Distributed Single Port RAM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

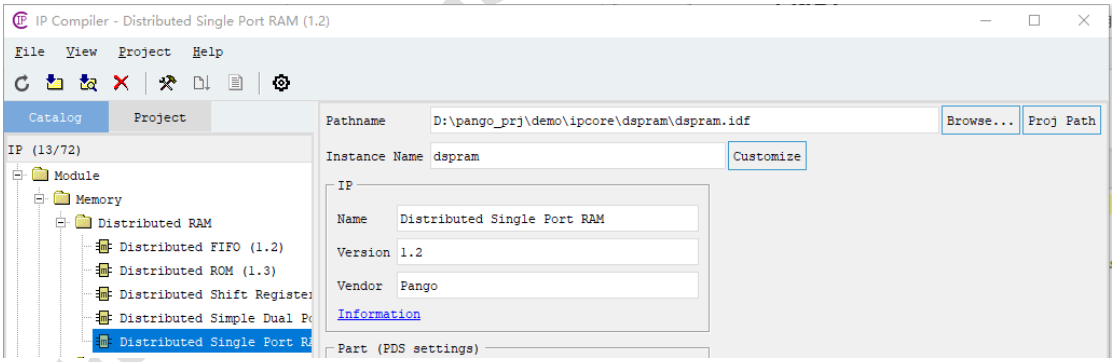


Figure3-30

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

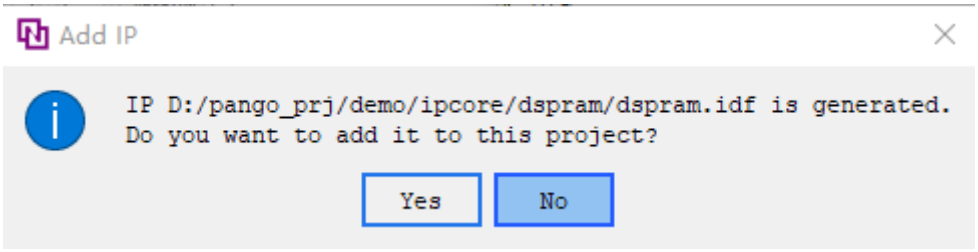


Figure3-31

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the Distributed Single Port RAM configuration interface, as shown in the figure below.

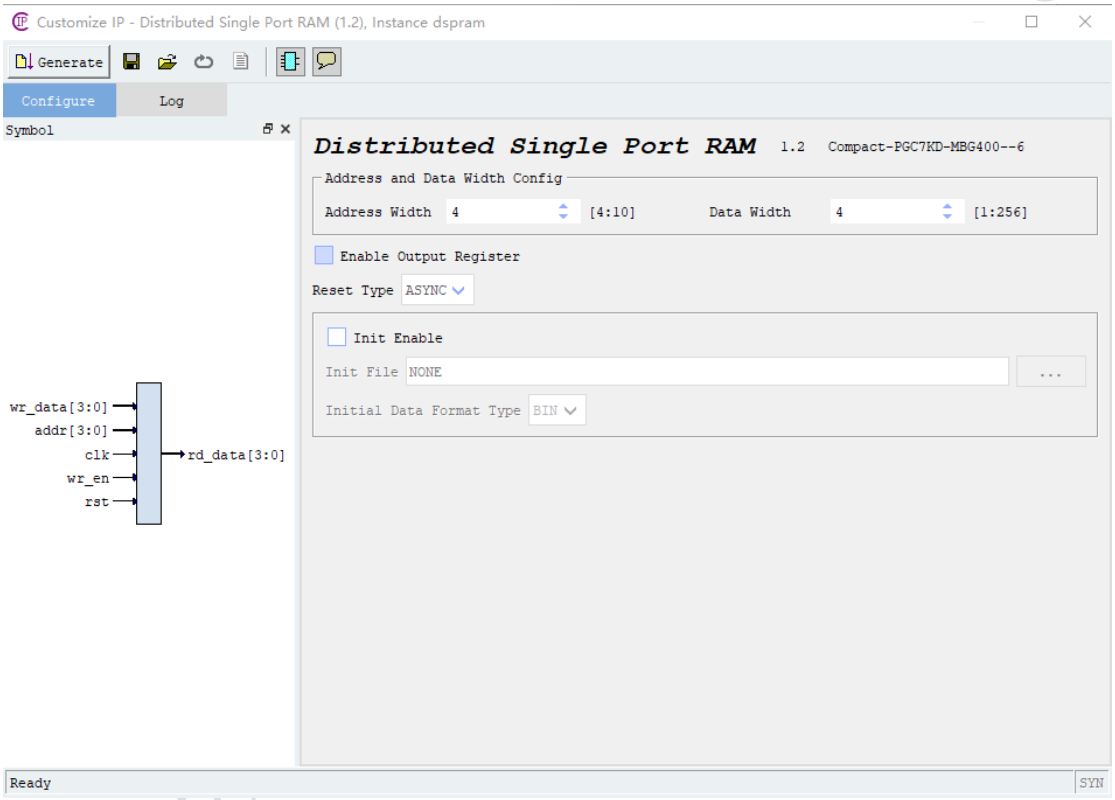


Figure3-32

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the Distributed Single Port RAM module, which can then be seen added to the project as shown in the figure below.

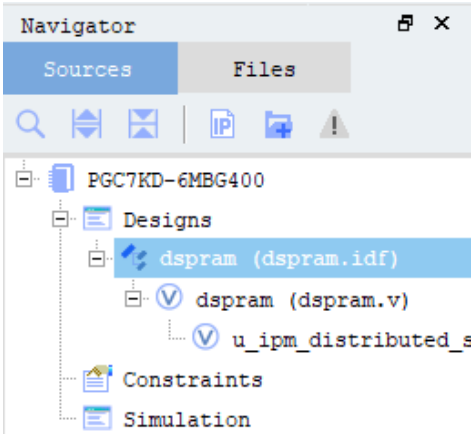
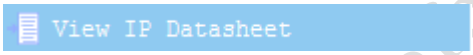
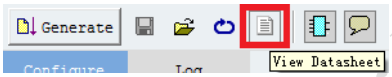


Figure3-33

To reset the parameters of the Distributed Single Port RAM, simply double-click the IP file "dspram.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Distributed RAM IP User Guide". To open it, right-click "dspram.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.1.5.2 Instantiate Distributed Single Port RAM Module

Right-click the top-level file of the Distributed Single Port RAM module and select [View Instantiation Template] to see the Verilog instantiation template for the Distributed Single Port RAM module, as shown in the figure below.

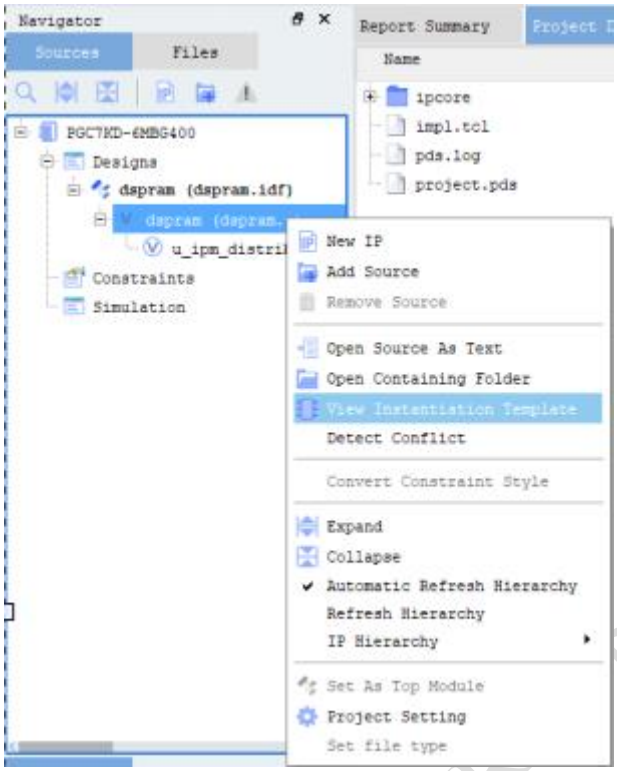


Figure3-34

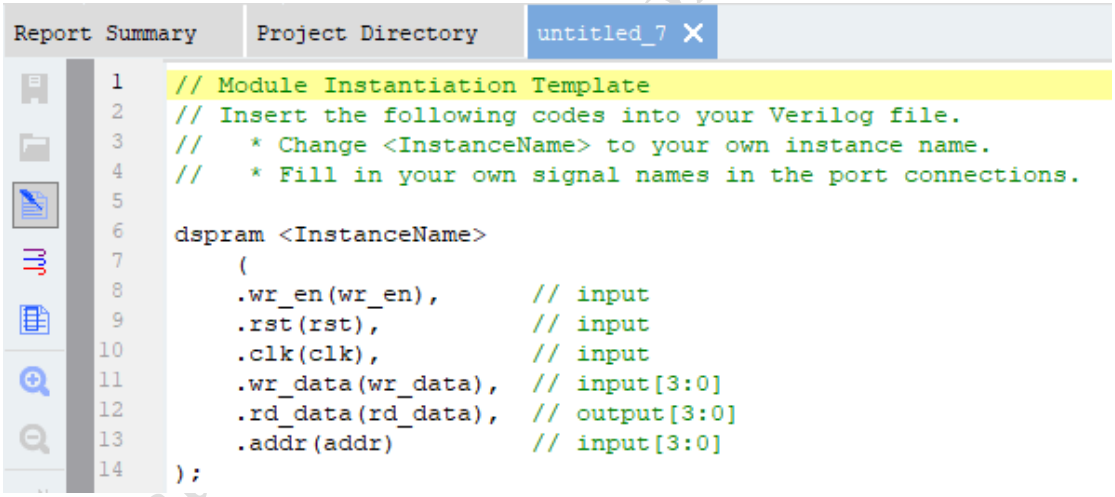


Figure3-35

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.2 DRM

DRM is a dedicated storage module for the Compa family products, with each DRM having a storage capacity of 9K bits. Each DRM can be configured to generate Dual Port RAM, FIFO, ROM, Simple Dual Port RAM, and Single Port RAM.

3.2.1 DRM Based Dual Port RAM

3.2.1.1 Create DRM Based Dual Port RAM Module

A new DRM Based Dual Port RAM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a DRM Based Dual Port RAM module.

In the IP Complier interface, select [DRM Based Dual Port RAM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

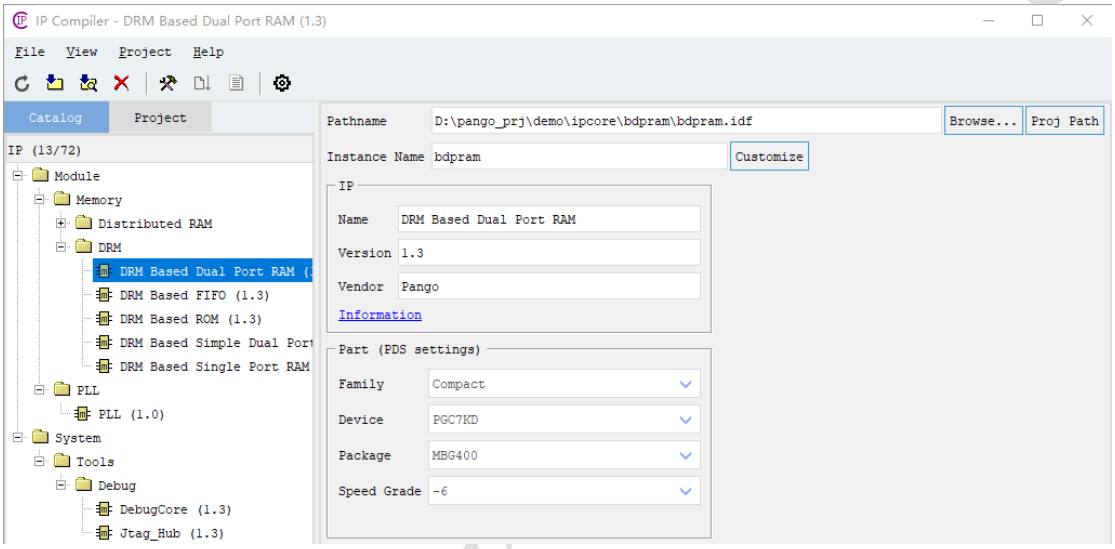


Figure3-36

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

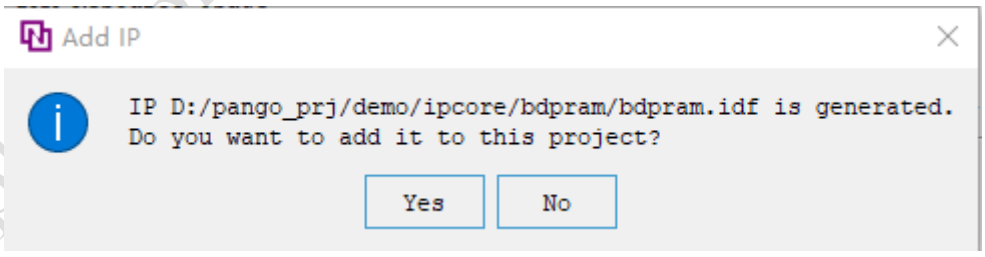


Figure3-37

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the DRM Based Dual Port RAM configuration interface, as shown in the figure below.

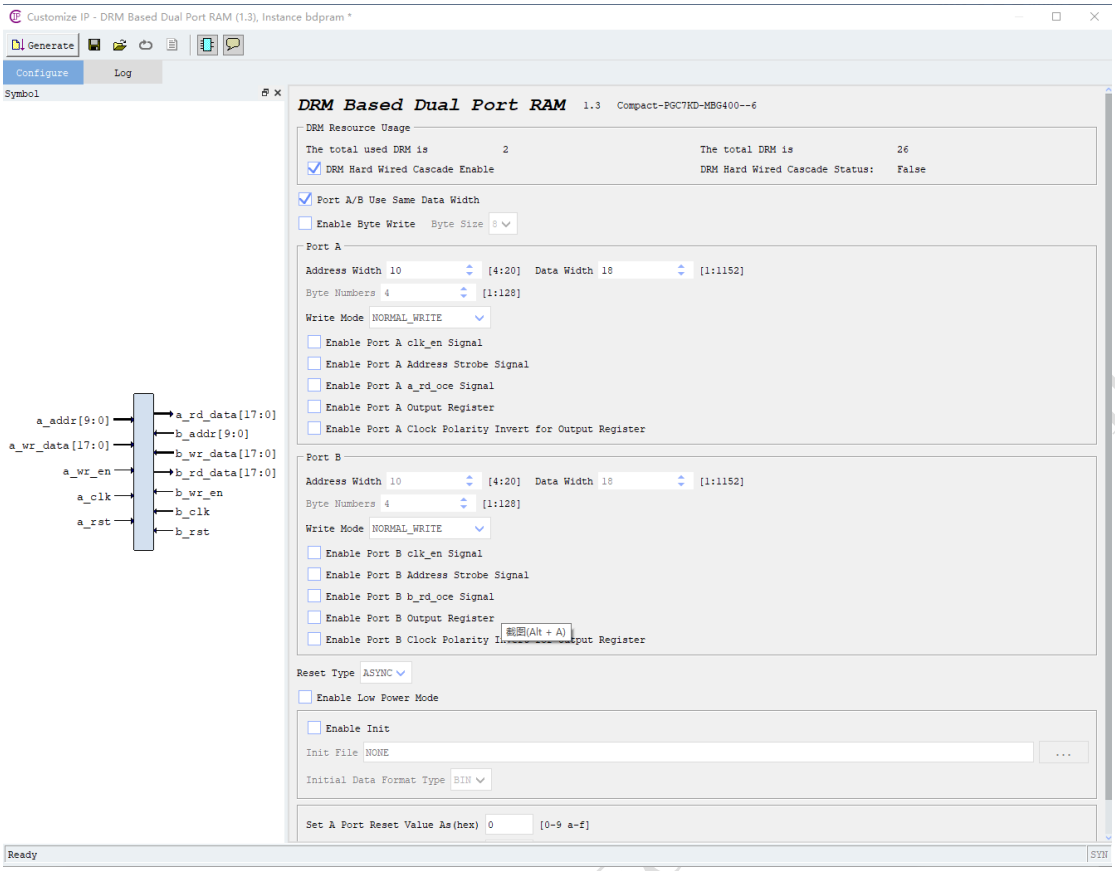


Figure3-38

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the DRM Based Dual Port RAM module, which can then be seen added to the project as shown in the figure below.

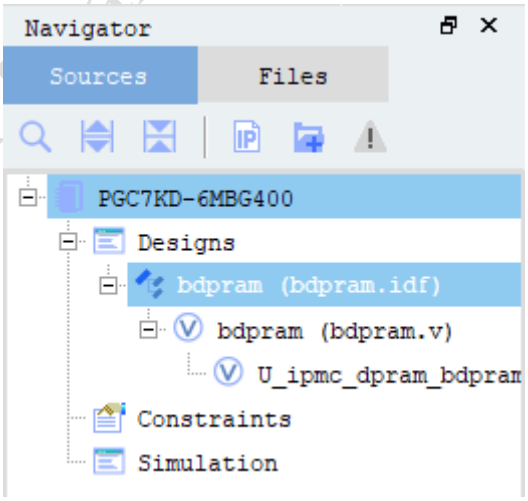
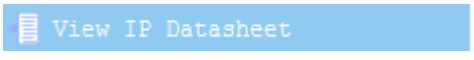
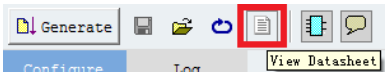


Figure3-39

To reset the parameters of the DRM Based Dual Port RAM, simply double-click the IP file "bdpram.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family DRM RAM/FIFO IP User Guide". To open it, right-click "bdpram.idf" and select  from the

drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.2.1.2 Instantiate DRM Based Dual Port RAM Module

Right-click the top-level file of the DRM Based Dual Port RAM module and select [View Instantiation Template] to see the Verilog instantiation template for the DRM Based Dual Port RAM module, as shown in the figure below.

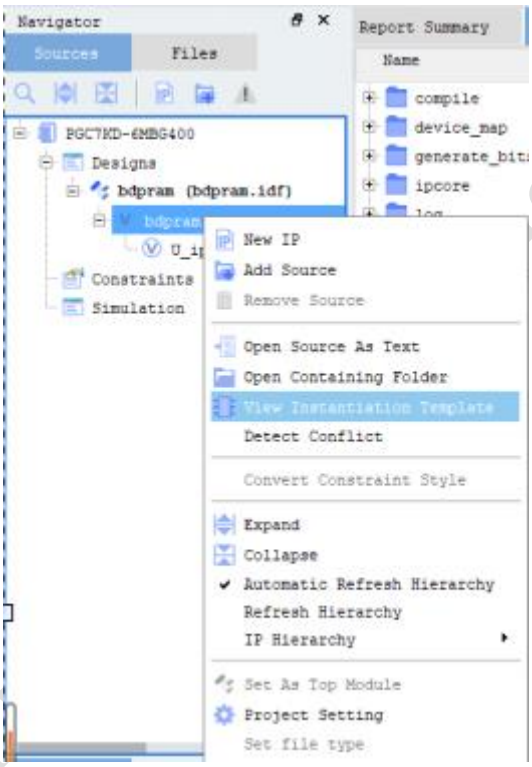


Figure3-40

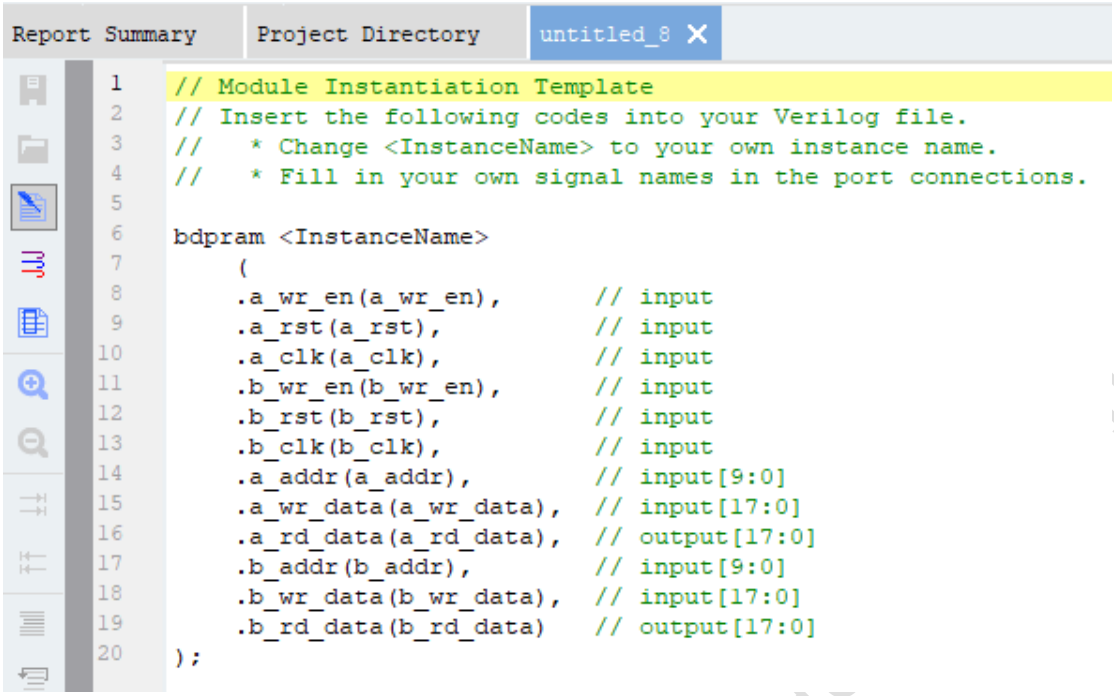


Figure3-41

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.2.2 DRM Based FIFO

3.2.2.1 Create DRM Based FIFO Module

A new DRM Based FIFO module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a DRM Based FIFO module.

In the IP Complier interface, select [DRM Based FIFO] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

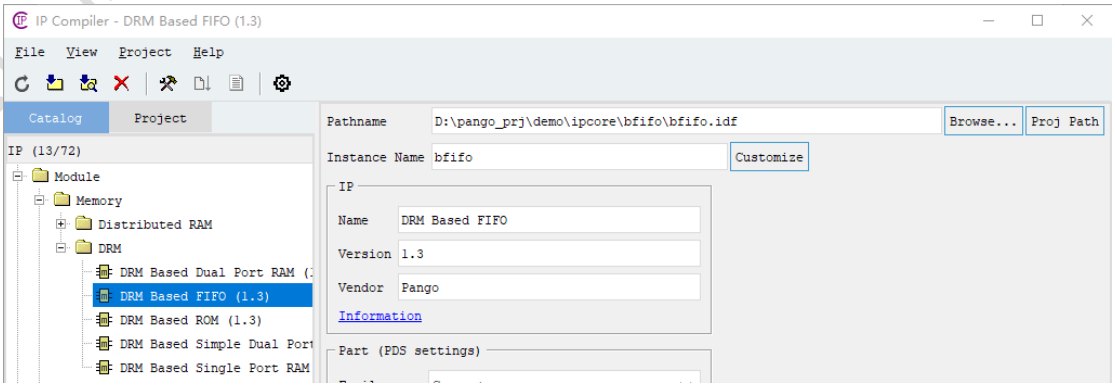


Figure3-42

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

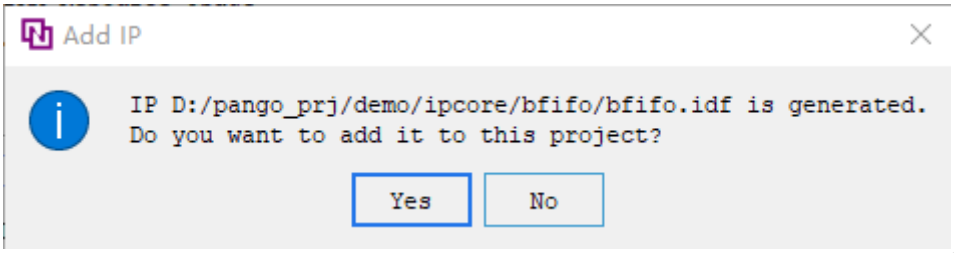


Figure3-43

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the DRM Based FIFO configuration interface, as shown in the figure below.

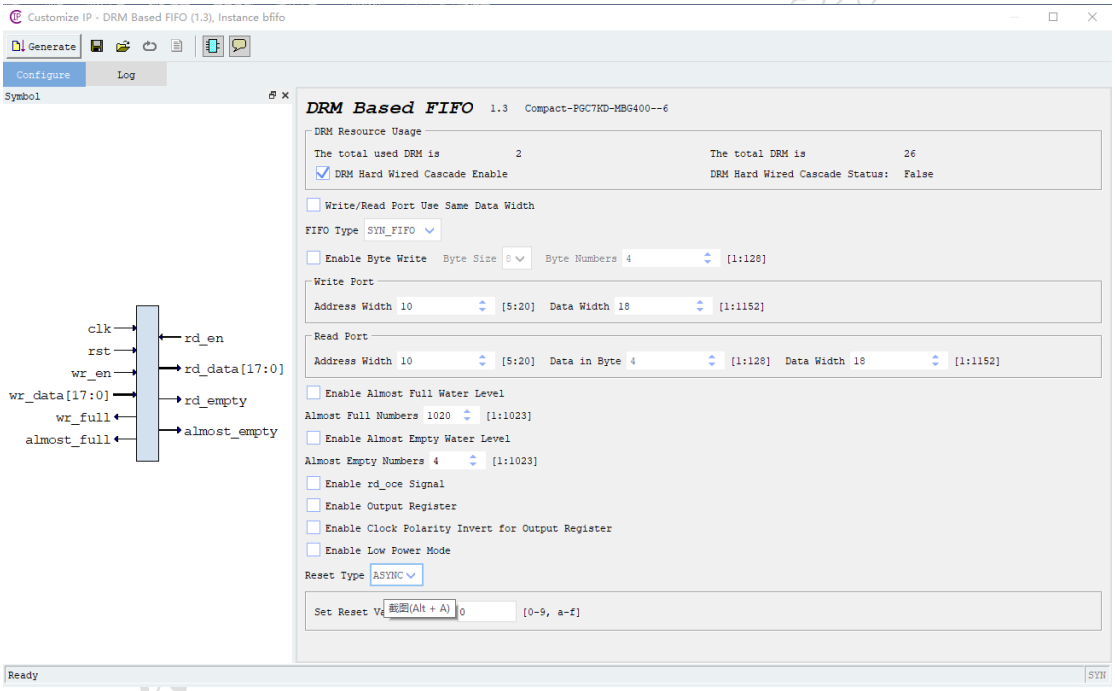


Figure3-44

After completing the settings, click [Generate] in the upper-left corner of the interface to generate the DRM Based FIFO module. users can then see that the module has been added to the project, as shown in the figure below.

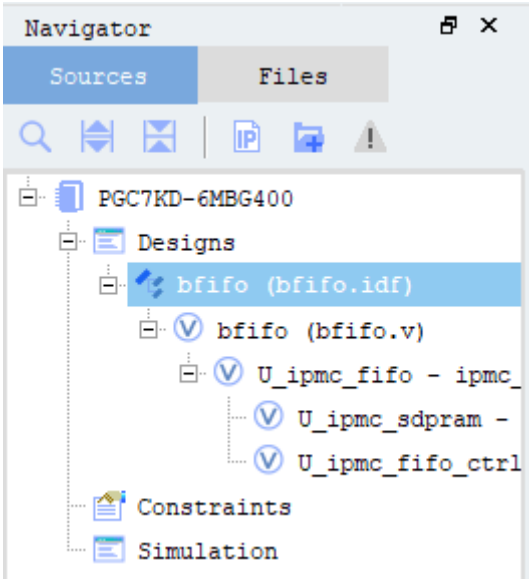
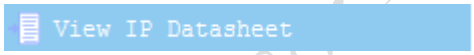
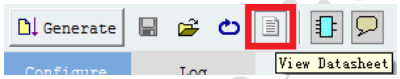


Figure3-45

To reset the parameters of the DRM Based FIFO, simply double-click the IP file "bfifo.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family DRM RAM/FIFO IP User Guide". To open it, right-click "bfifo.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.2.2.2 Instantiate DRM Based FIFO Module

Right-click the top-level file of the DRM Based FIFO module and select [View Instantiation Template] to see the Verilog instantiation template for the DRM Based FIFO module, as shown in the figure below.

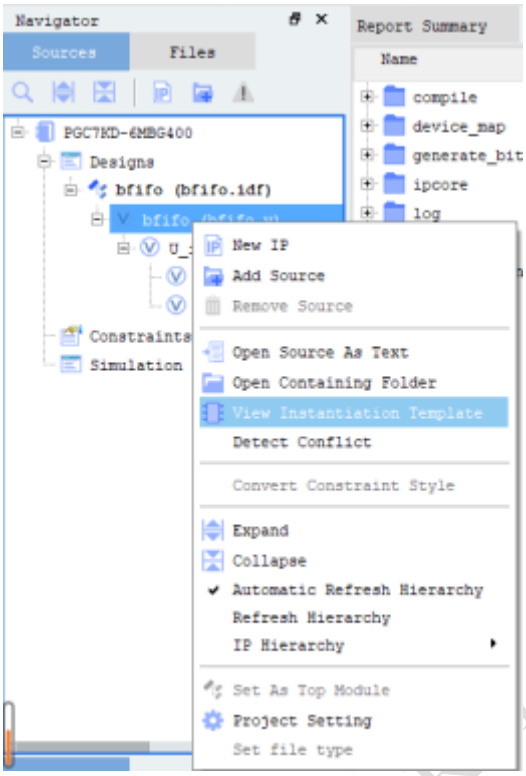


Figure3-46

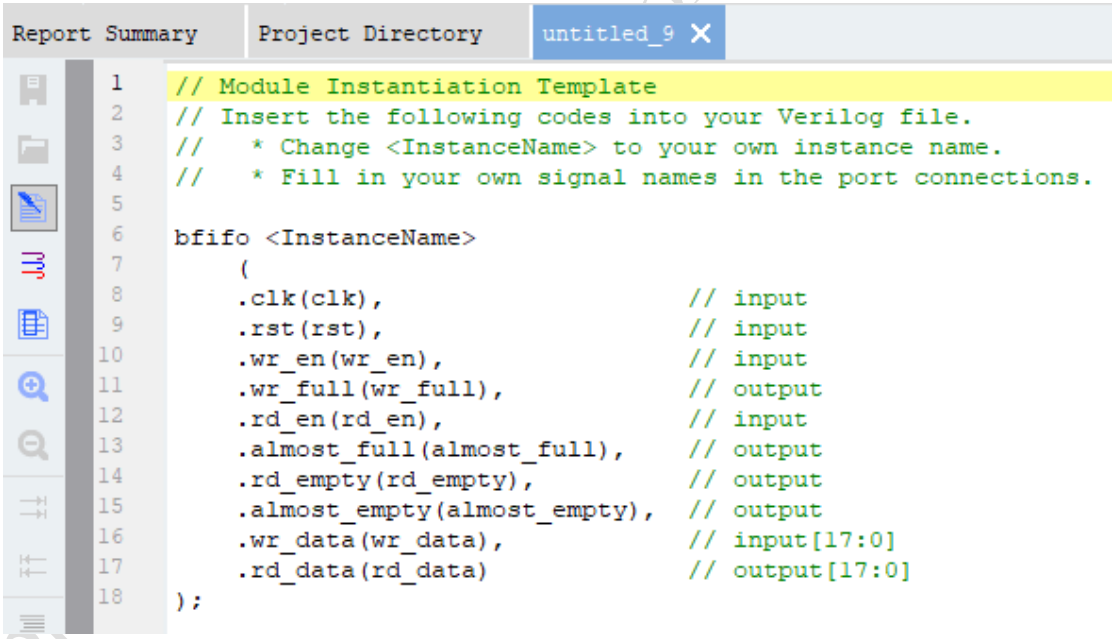


Figure3-47

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.2.3 DRM Based ROM

3.2.3.1 Create DRM Based ROM Module

A new DRM Based ROM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a DRM Based ROM module.

In the IP Complier interface, select [DRM Based ROM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

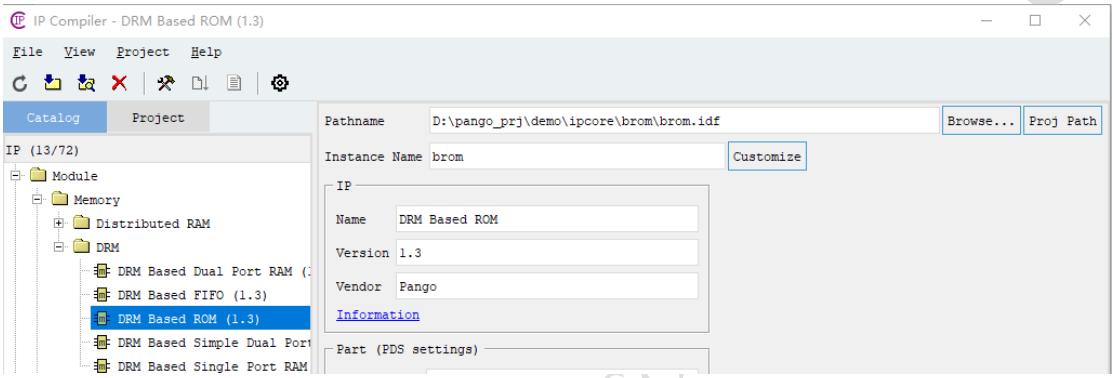


Figure3-48

After completing the input, click [Customize], and a prompt will appear asking if there is an initialization file. If an ROM initialization file already exists, click [Next].

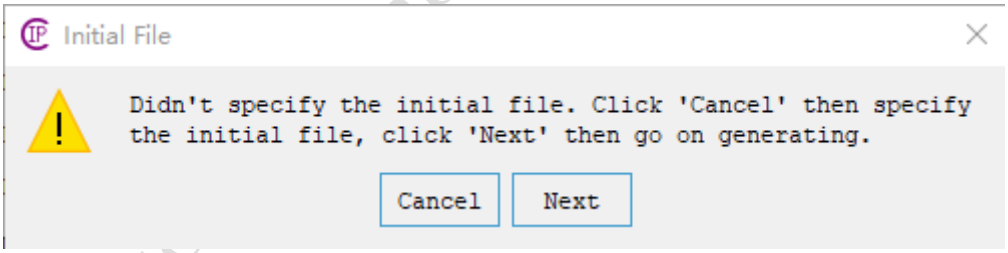


Figure3-49

Then, a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

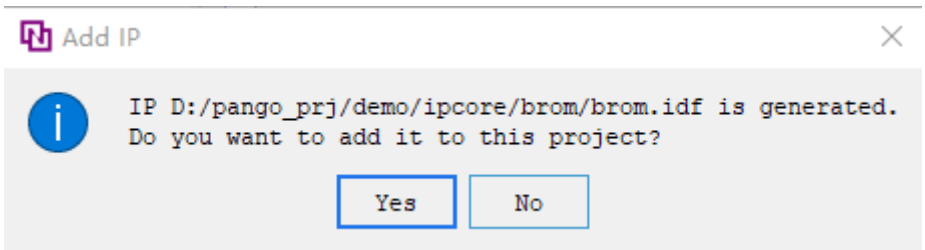


Figure3-50

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the DRM Based ROM configuration interface, as shown

in the figure below.

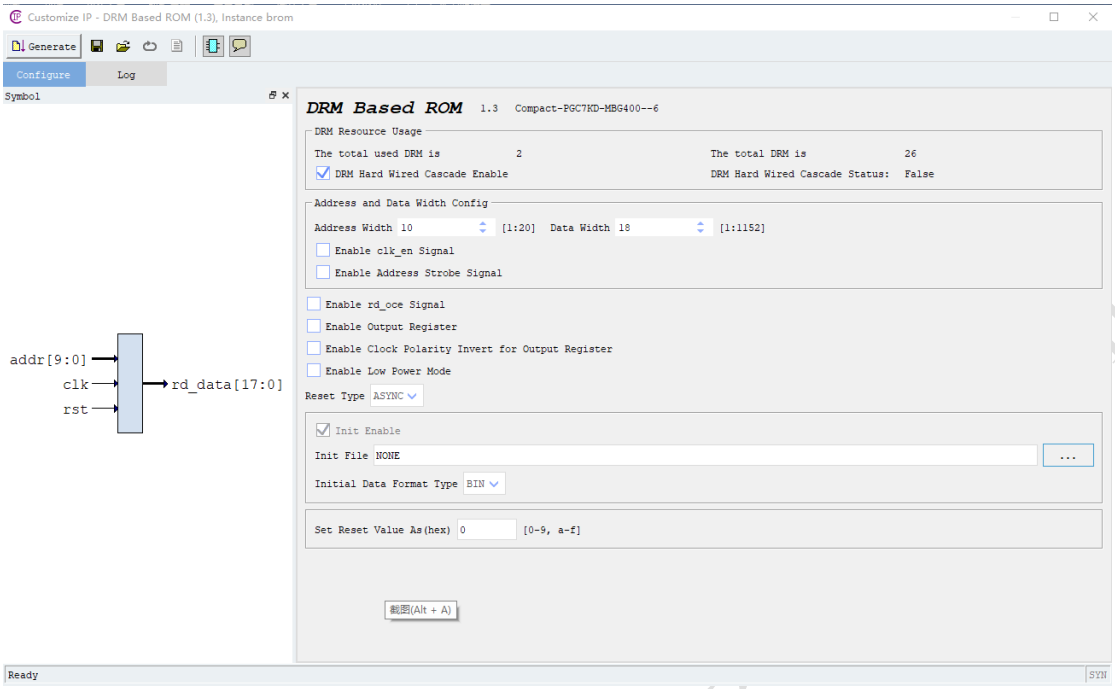


Figure3-51

After completing the settings, click [Generate] in the upper-left corner of the interface to generate the DRM Based ROM module. users can then see that the module has been added to the project, as shown in the figure below.

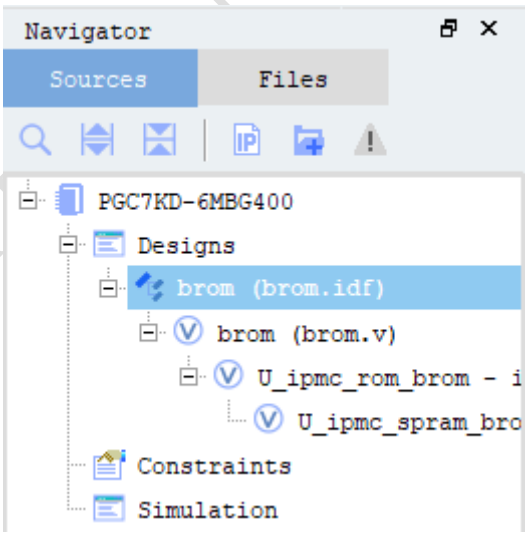
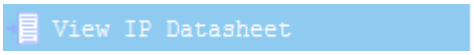
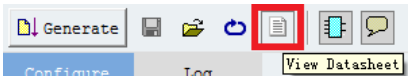


Figure3-52

To reset the parameters of the DRM Based ROM, simply double-click the IP file "brom.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family DRM RAM/FIFO IP User Guide". To open it, right-click "brom.idf" and select 

from the drop-down menu;

or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.2.3.2 Instantiate DRM Based ROM Module

Right-click the top-level file of the DRM Based ROM module and select [View Instantiation Template] to see the Verilog instantiation template for the DRM Based ROM module, as shown in the figure below.

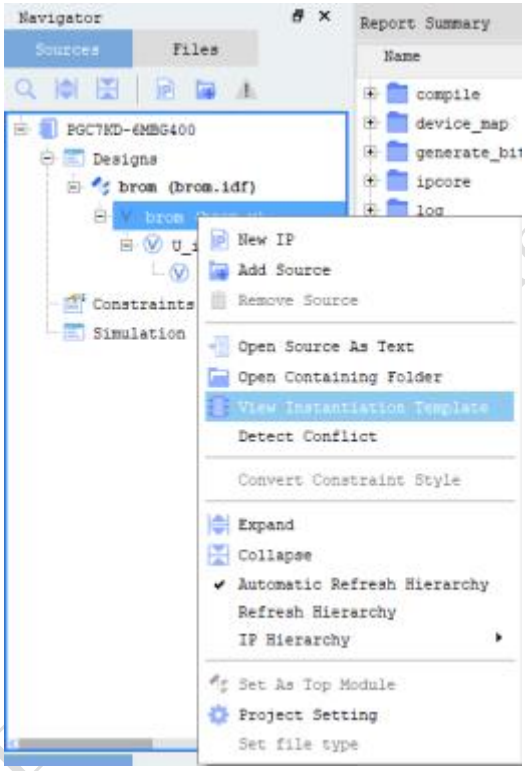


Figure3-53

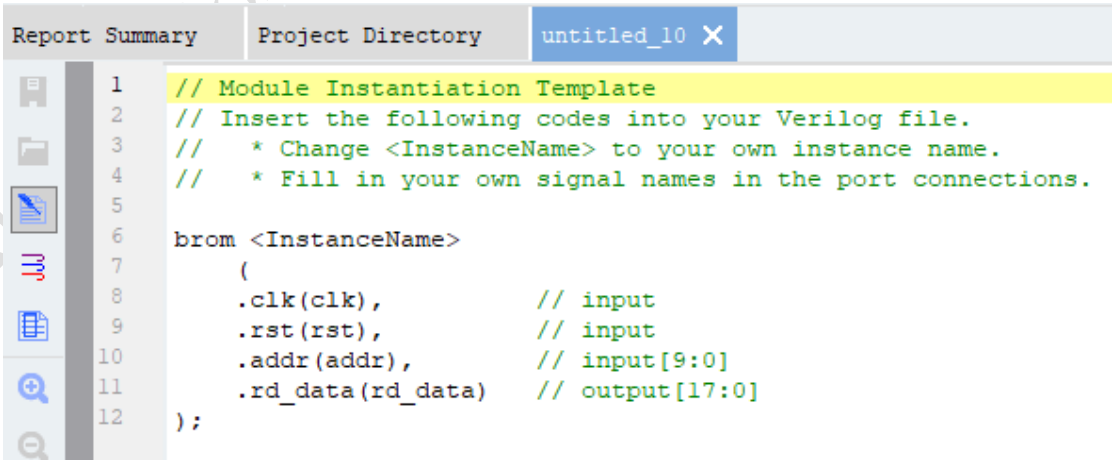


Figure3-54

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.2.4 DRM Based Simple Dual Port RAM

3.2.4.1 Create DRM Based Simple Dual Port RAM Module

A new DRM Based Simple Dual Port RAM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a DRM Based Simple Dual Port RAM module.

In the IP Complier interface, select [DRM Based Simple Dual Port RAM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

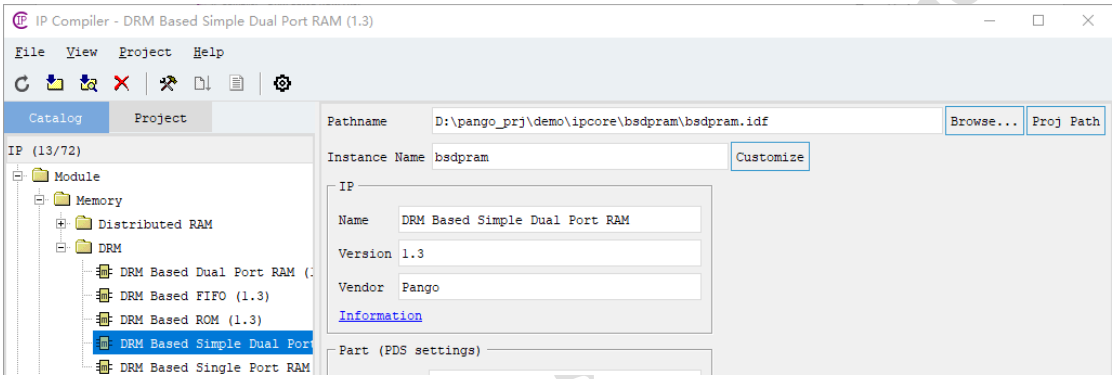


Figure3-55

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

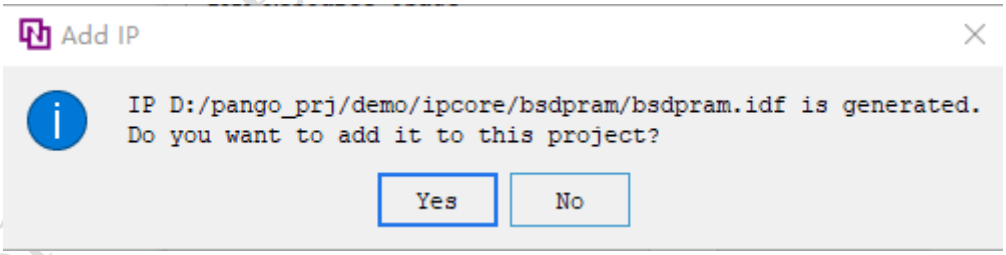


Figure3-56

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the DRM Based Simple Dual Port RAM configuration interface, as shown in the figure below.

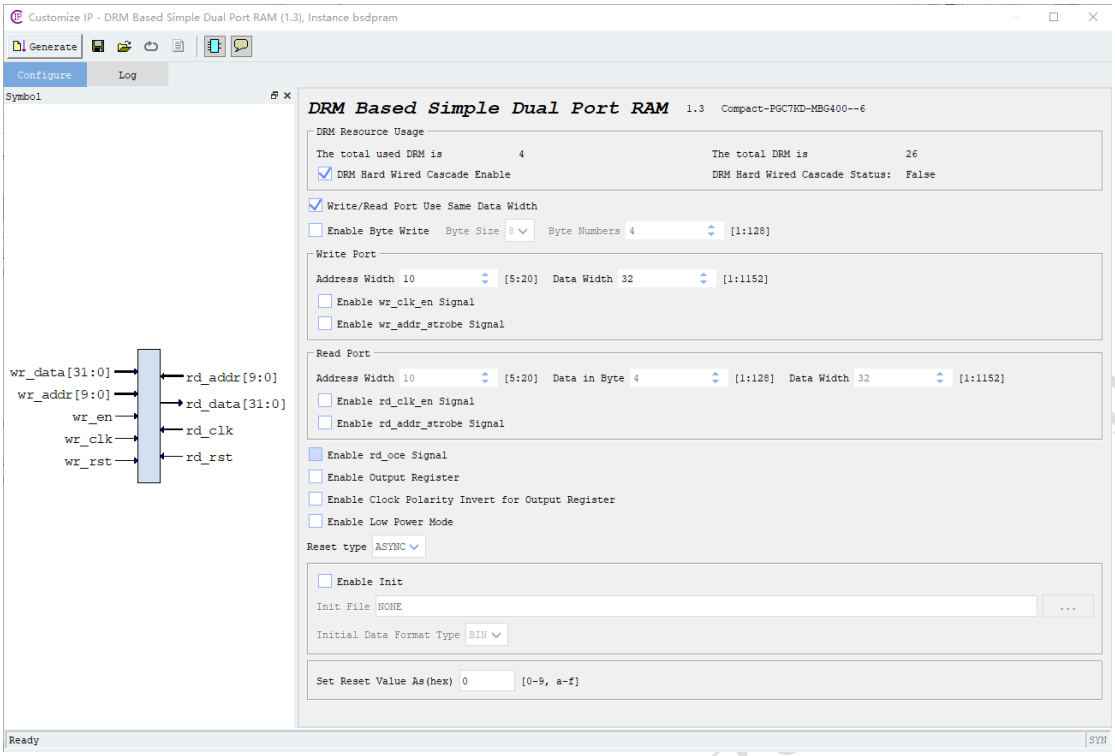


Figure3-57

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the DRM Based Simple Dual Port RAM module, which can then be seen added to the project as shown in the figure below.

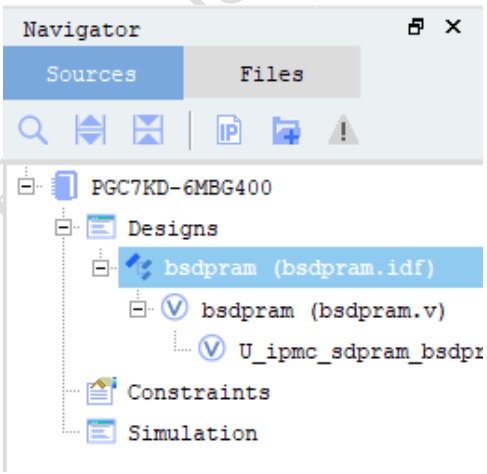
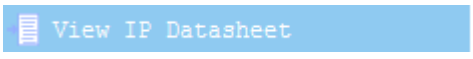
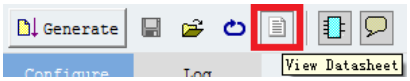


Figure3-58

To reset the parameters of the DRM Based Simple Dual Port RAM, simply double-click the IP file "bsdpram.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family DRM RAM/FIFO IP User Guide". To open it, right-click "bsdpram.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.2.4.2 Instantiate DRM Based Simple Dual Port RAM Module

Right-click the top-level file of the DRM Based Simple Dual Port RAM module and select [View Instantiation Template] to see the Verilog instantiation template for the DRM Based Simple Dual Port RAM module, as shown in the figure below.

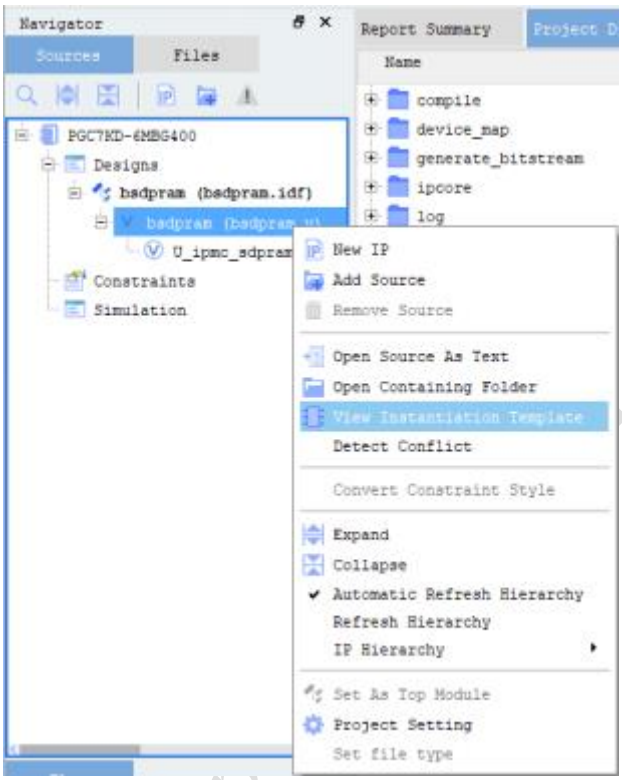


Figure3-59

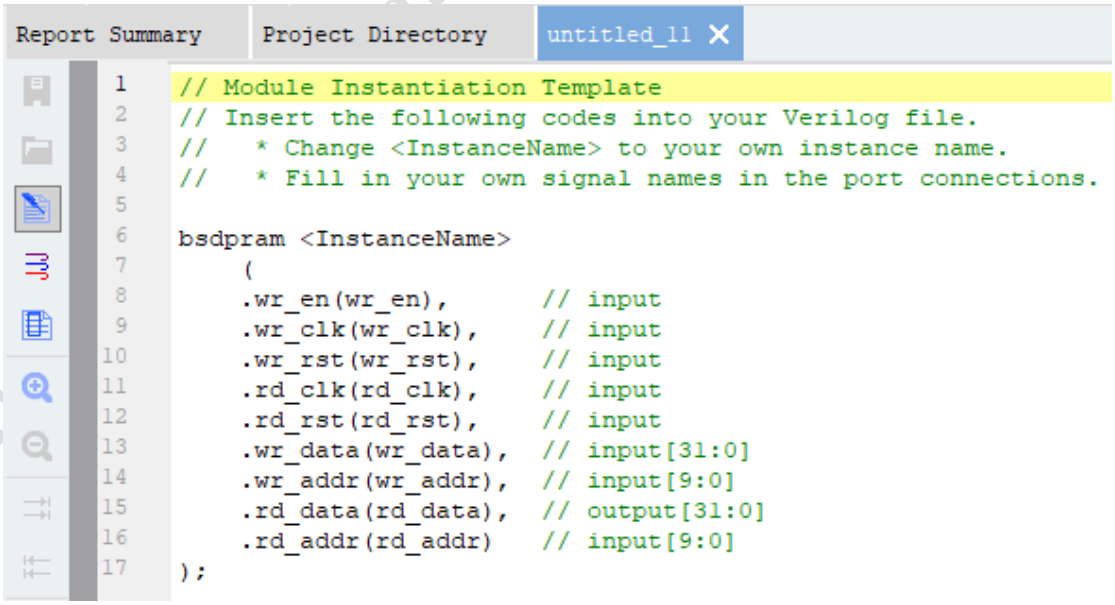


Figure3-60

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.2.5 DRM Based Single Port RAM

3.2.5.1 Create DRM Based Single Port RAM Module

A new DRM Based Single Port RAM module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a DRM Based Single Port RAM module.

In the IP Complier interface, select [DRM Based Single Port RAM] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

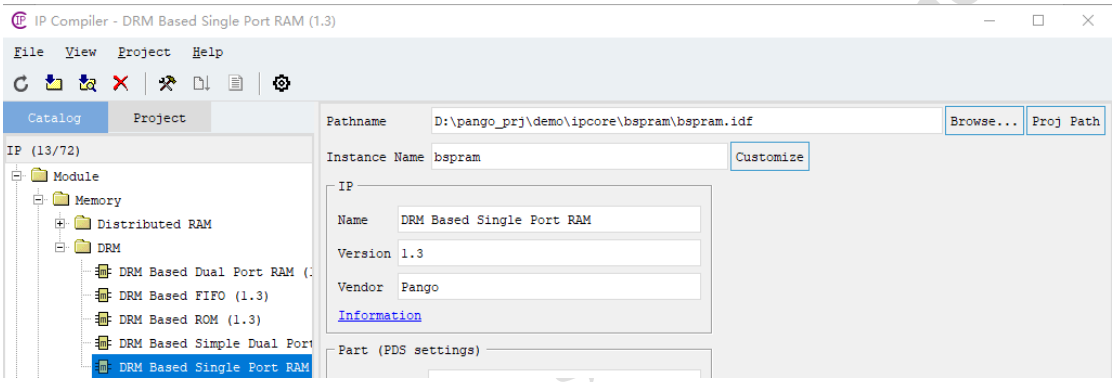


Figure3-61

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

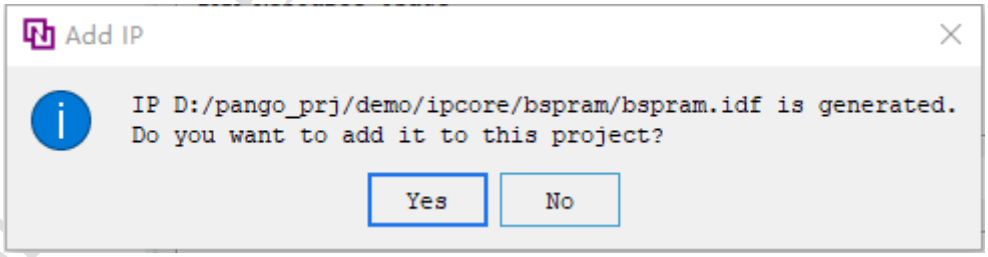


Figure3-62

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the DRM Based Single Port RAM configuration interface, as shown in the figure below.

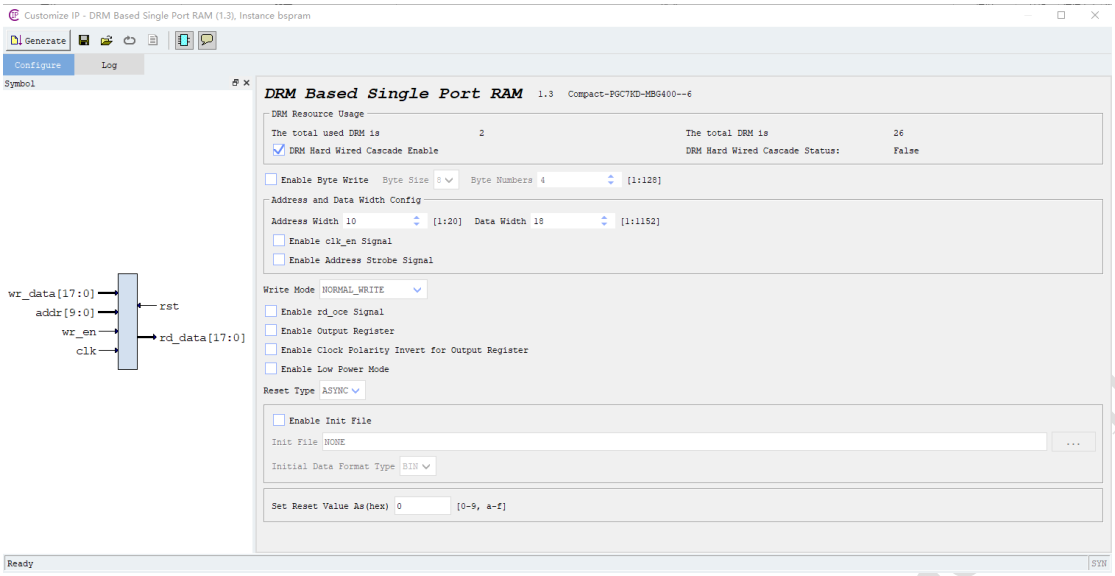


Figure3-63

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the DRM Based Single Port RAM module, which can then be seen added to the project as shown in the figure below.

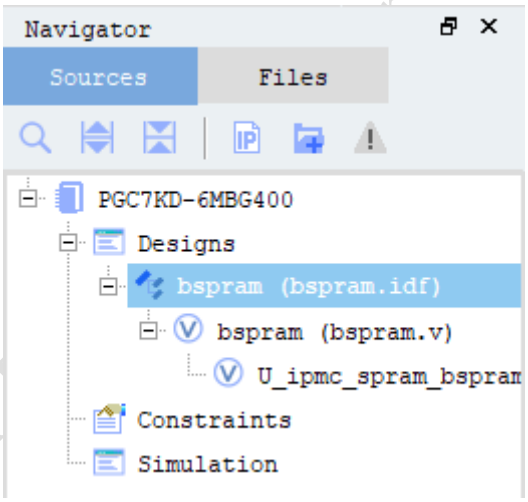
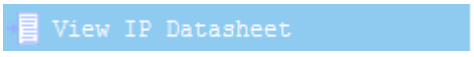
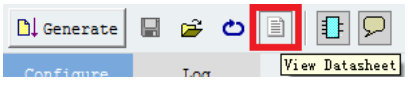


Figure3-64

To reset the parameters of the DRM Based Single Port RAM, simply double-click the IP file "bspram.idf" to reopen its configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family DRM RAM/FIFO IP User Guide". To open it, right-click "bspram.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.2.5.2 Instantiate DRM Based Single Port RAM Module

Right-click the top-level file of the DRM Based Single Port RAM module and select [View Instantiation Template] to see the Verilog instantiation template for the DRM Based Single Port RAM module, as shown in the figure below.

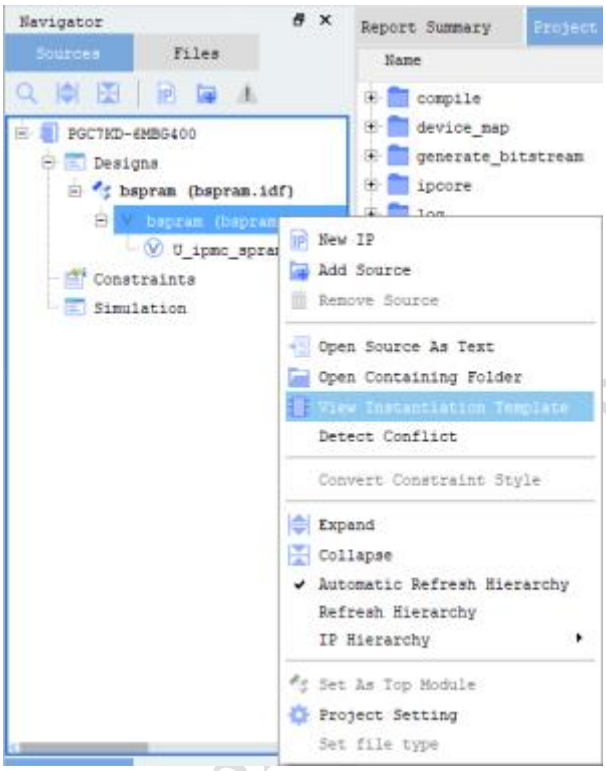


Figure3-65

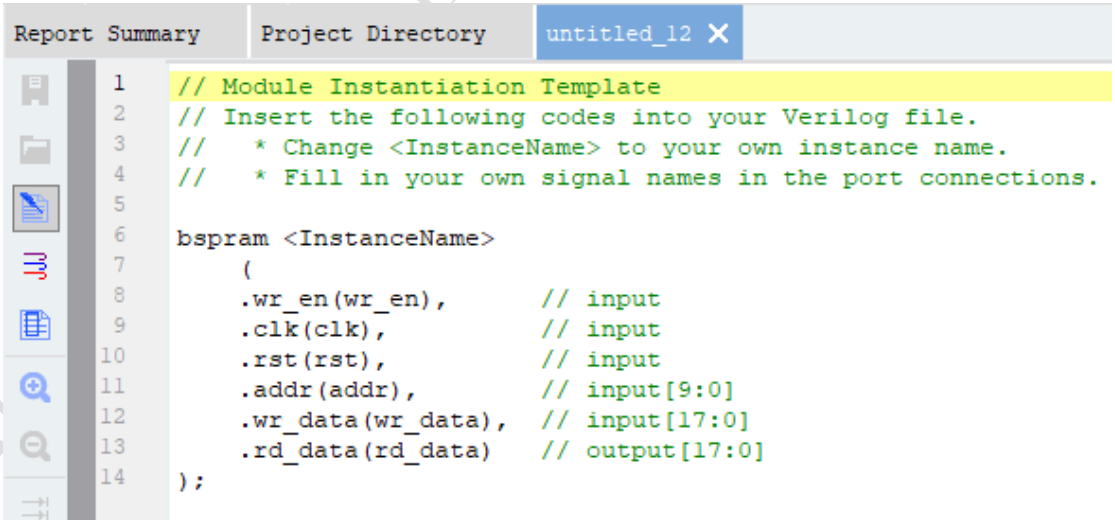


Figure3-66

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

3.3 PLL

PGC devices contain up to 2 PLLs, each with the following features:

1. Dynamic switching of input clocks
2. Power down function
3. Reset function
4. Internal/external feedback
5. Static/dynamic integer division
6. Fractional division
7. Static/dynamic phase offset
8. Output clock gate
9. Bypass output
10. Divider output cascading
11. APB interface

3.3.1 Create PLL Module

A new PLL module can be created separately or within an existing project. This section take an existing project as an example to illustrate how to create a PLL module.

In the IP Complier interface, select [PLL] under [Catalog], enter the IP module name in [Instance Name], and a folder with the same name will be created in the "..\ipcore\" path, as shown in the figure below.

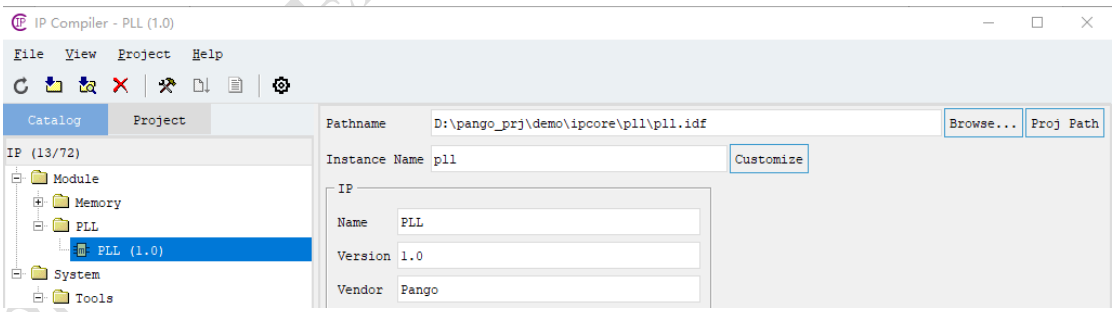


Figure3-67

After completing the input, click [Customize], and a prompt will appear asking if users want to add the newly created IP module to the project. It is recommended to select [Yes], as shown in the figure below.

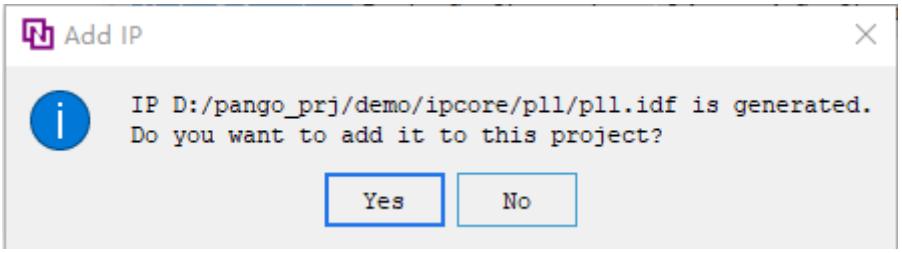


Figure3-68

If [No] is selected, users need to right-click [Navigator] and manually add .idf or .v files.

Then, set the corresponding parameters in the PLL configuration interface, as shown in the figure below.

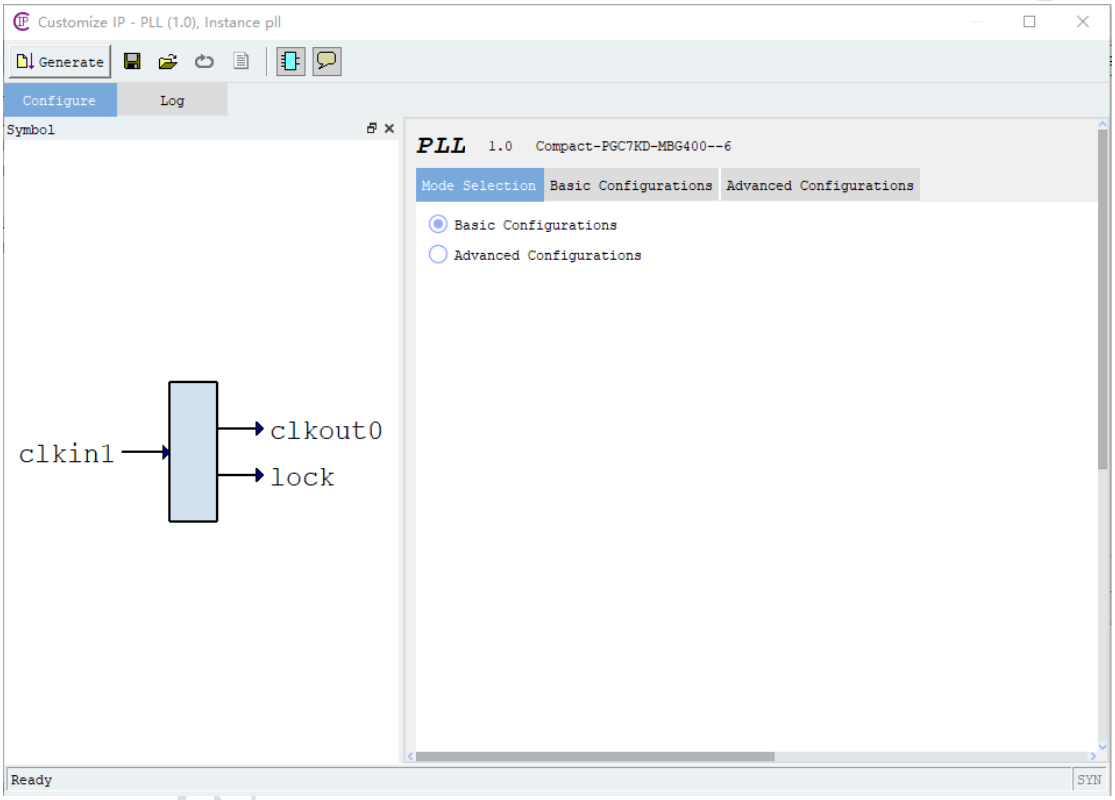


Figure3-69

The PLL IP configuration interface includes a basic mode and an advanced mode.

3.3.1.1 Basic Mode

On the [Mode Selection] page, select [Basic Configuration], and then click on the [Basic Configuration] page to set up the basic mode, as shown in the figure below.

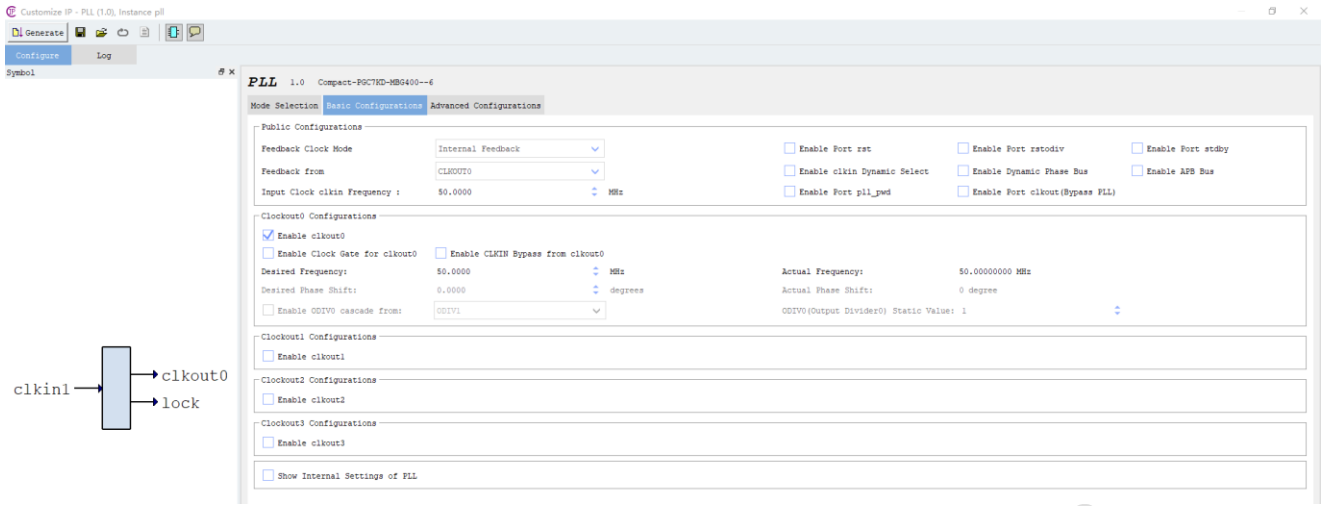


Figure3-70

[Feedback Clock Mode]: users can choose between internal feedback and external feedback, with external feedback typically used for zero delay buffer mode.

[Feedback from]: Select the feedback clock path, users can choose CLKOUT0, CLKOUT1, CLKOUT2, or CLKOUT3. The selected output clock determines the output division ratio, which ultimately affects the frequency of each output clock.

[Input Clock clkin Frequency]: PLL input clock, with an input range of 10-500MHz;

[Enable Port rst]: When checked, it enables the rst pin, which is active-high, to reset the PLL.

[Enable clkin Dynamic Select]: When checked, it enables the clkin_sel pin, which allows for dynamic switching of input clocks.

[Enable Port pll_pwd]: When checked, it enables the pll_pwd pin, which is active-high, and modules other than LDO are turned off.

[Enable Port rstodiv]: When checked, it enables the rstodiv pin, which is active-high, to reset the PLL Core (VCO/PFD/CP/LPF) and all dividers except the input divider.

[Enable Dynamic Phase Bus]: When checked, it enables the pin for dynamic phase adjustment.

[Enable Port clkout (Bypass PLL)]: When checked, it outputs the bypass clock clkout.

[Enable Port stdby]: When checked, it enables the stdby pin, which is active-high, to put the PLL in standby mode.

[Enable APB Bus]: When checked, it enables the APB bus interface, allowing users to access the PLL internal registers through the APB interface.

[Enable clkout0/1/2/3]: When checked, it enables the output clock pins clkout0/1/2/3.

[Enable Clock Gate for clkout0/1/2/3]: When checked, it enables the clkout0/1/2/3_syn pins as the gating signal for the output clock.

[Enable CLKIN Bypass from clkout0/1/2/3]: When checked, clkin is output directly through clkout0/1/2/3 bypass.

[Desired Frequency]: Sets the desired output clock frequency, with an output frequency range of 3.125-600MHz.

[Actual Frequency]: The actual output frequency.

[Desired Phase Shift]: Set the desired output clock phase.

[Actual Phase Shift]: The actual output clock phase.

[Enable ODIV1/2/3 cascade from]: Selects the input for cascading clocks to further divide the output clock of the previous stage; when checked, users can enter the division factor in [ODIV (Output Divider) Static Value] to achieve further division, with a minimum output of 190.73Hz.

[Show Internal Settings of PLL]: When checked, the PLL parameter details are displayed.

3.3.1.2 Advanced Mode

On the [Mode Selection] page, select [Advanced Configuration], and then click on the [Advanced Configuration] page to set up the advanced mode, as shown in the figure below.

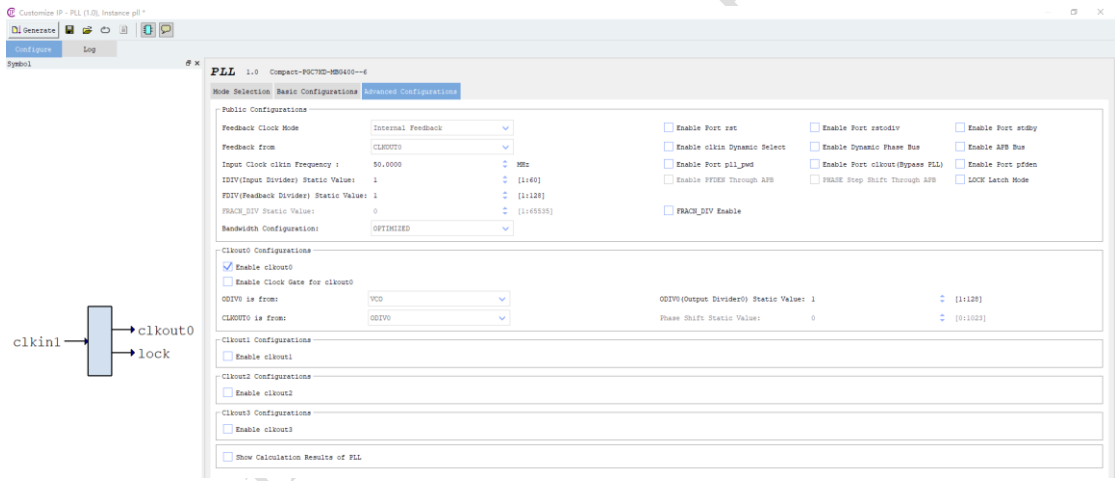


Figure3-71

In addition to all the functions of the basic mode, the advanced mode also supports the following functions:

[Enable Port PFEDN]: When checked, the pfden interface is enabled; pfden is the PFD enable signal, with high level turning PFD on;

[Enable PFDEN Through APB]: When checked, PFDEN is allowed to be configured through the APB interface;

[PHASE Step Shift Through APB]: When checked, the phase is allowed to be dynamically adjusted through the APB interface;

[LOCK Latch Mode]: When not checked, the LOCK signal is in normal lock mode, pulled high within lock precision range, and pulled low when exceeding lock precision. When checked, the LOCK signal is in strict lock mode, pulled high within lock precision range unless the PLL is reset

or powered down.

When setting the output frequency, users need to set the input division ratio IDIV, feedback division ratio FDIV, and output division ratio ODIV, with the calculation formula as shown in the figure below.

F_{PFD} = Fin /M

Fvco = (Fin /M)*V*Neff

Fout = Fvco /V

Neff =N+ Frac_div

Table 3-1

Symbol	Description
F_{in}	Input reference frequency
F_{PFD}	Input reference frequency of PFD
F_{vco}	Output frequency of VCO
F_{out}	Output frequency of CLKOUT0/1/2/3
M	The division ratio of the input divider
V	The division ratio of the output divider
N_{eff}	The division ratio of the feedback divider
N	The integer division ratio of the feedback divider
F_{rac_div}	The fractional division ratio of the feedback divider

Additionally, users can employ fractional division and select [FRACN_DIV Enable], and then enter the fractional division ratio in [FRACN_DIV Static Value].

After completing the configuration, click on the [Generate] button at the upper-left corner of the interface to generate the PLL module, which can then be seen added to the project as shown in the figure below.

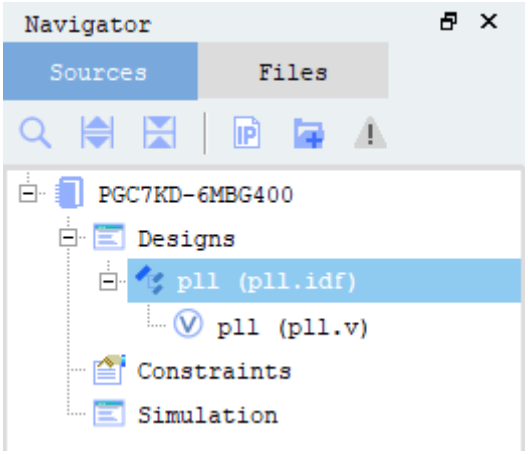
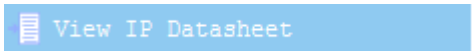
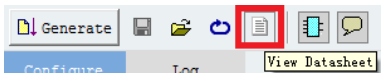


Figure3-72

To reset the parameters of the PLL, simply double-click the IP file "pll.idf" to reopen its

configuration interface. If the user wants to learn more about the IP's interfaces and parameters, please refer to the "Compa Family PLL IP User Guide". To open it, right-click "pll.idf" and select 

from the drop-down menu; or select [View Datasheet] 

in the upper-left corner of the IP configuration interface.

3.3.2 Instantiate PLL Module

Right-click the top-level file of the PLL module and select [View Instantiation Template] to see the Verilog instantiation template for the PLL module, as shown in the figure below.

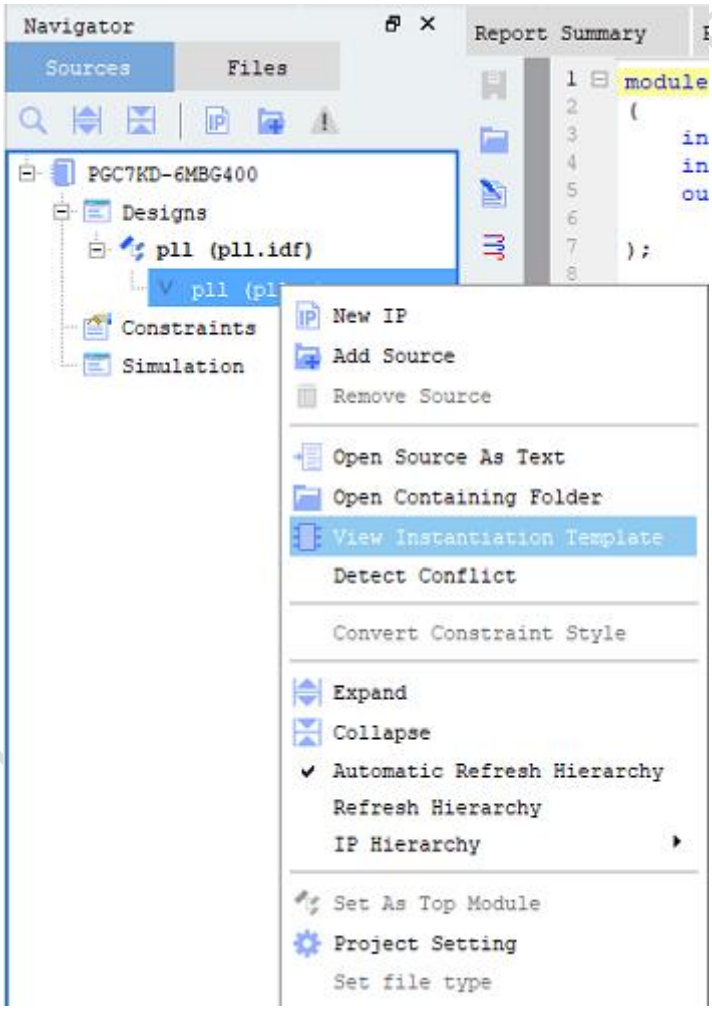


Figure3-73

Report Summary | Project Directory | untitled_1

```
// Module Instantiation Template
// Insert the following codes into your Verilog file.
//   * Change <InstanceName> to your own instance name.
//   * Fill in your own signal names in the port connections.

pll <InstanceName>
    (
    .clkout(clkout),                    // output
    .clkout0(clkout0),                  // output
    .clkout1(clkout1),                  // output
    .clkout2(clkout2),                  // output
    .clkout3(clkout3),                  // output
    .clkin1(clkin1),                    // input
    .clkin2(clkin2),                    // input
    .clkin_sel(clkin_sel),              // input
    .phase_dir(phase_dir),              // input
    .phase_step_n(phase_step_n),        // input
    .load_phase(load_phase),            // input
    .cphase_step_n(cphase_step_n),      // input
    .clkout0_syn(clkout0_syn),          // input
    .clkout1_syn(clkout1_syn),          // input
    .clkout2_syn(clkout2_syn),          // input
    .clkout3_syn(clkout3_syn),          // input
    .stdby(stdby),                      // input
    .pll_pwd(pll_pwd),                  // input
    .rst(rst),                          // input
    .rstodiv(rstodiv),                  // input
    .apb_clk(apb_clk),                  // input
    .apb_rst_n(apb_rst_n),              // input
    .apb_sel(apb_sel),                  // input
    .apb_en(apb_en),                    // input
    .apb_write(apb_write),              // input
    .lock(lock),                        // output
    .phase_sel(phase_sel),              // input[1:0]
    .apb_addr(apb_addr),                // input[4:0]
    .apb_wdata(apb_wdata)               // input[7:0]
);
```

Figure3-74

Modify the <InstanceName> and copy the entire code block to other design files to complete instantiation.

Chapter 4 User Constraint

User constraints include physical constraints and timing constraints. Users can apply these constraints to the design using the User Constraint Editor (hereinafter referred to as UCE). User Constraint Editor can be opened by clicking the icon on the toolbar, as shown in the figure below.

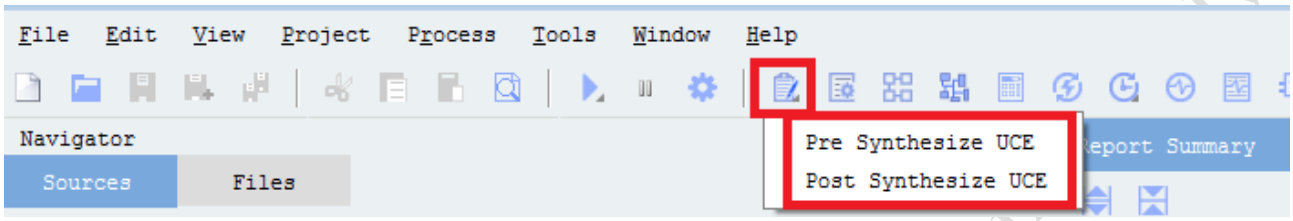


Figure4-1

Or by clicking [Tools] and selecting [User Constraint Editor (Timing and Logic)], as shown in the figure below.

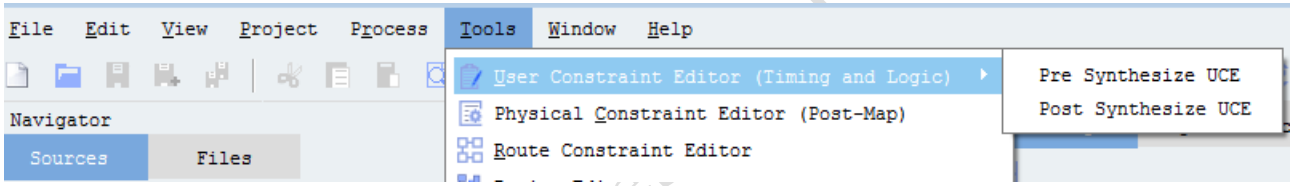


Figure4-2

It can be seen that UCE is divided into two stages: Pre Synthesize UCE and Post Synthesize UCE. Pre Synthesize UCE reads, writes, and edits pre-synthesis constraints, meaning the constraints files read and saved are .fdc (FPGA Design Constraint) files, and the saved .fdc files are used during synthesis. Post Synthesize UCE reads, writes, and edits post-synthesis constraints, meaning the constraints files read and saved are .lcf (Logical Constraint File) files and .scf (Synthesis Constraint File) files, and the saved .lcf and .scf files are used during Device Map. The difference between them is that the Post Synthesize UCE adds constraints and region constraints to the instances.

After opening the UCE, there are three buttons at the top of the UCE interface, as shown in the figure below:

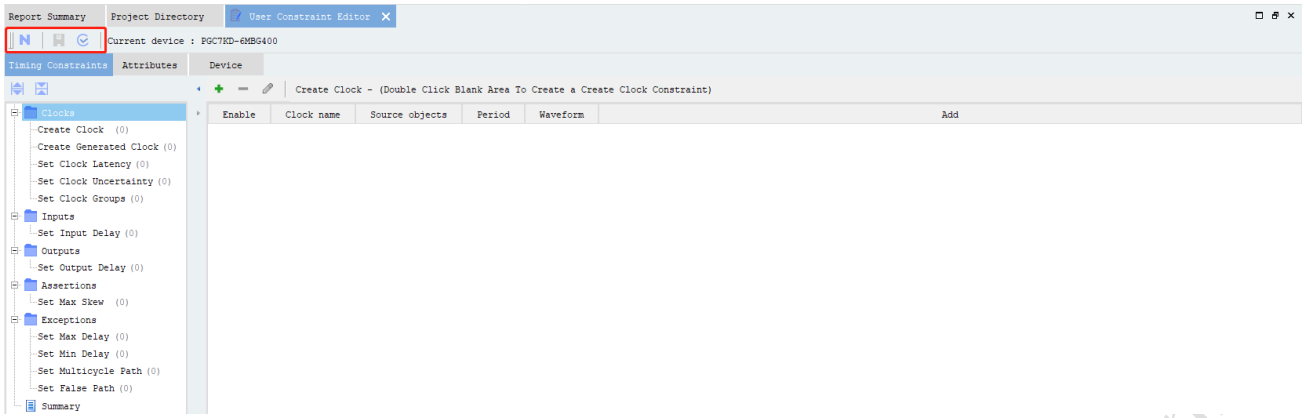


Figure4-3

They are, in order, [Design Browser], [Save], and [Check Constraints].

The [Design Browser] button opens the netlist viewing tool, which allows users to view the structure and content of the design netlist at that stage. The content of the netlist is roughly divided into three categories: Ports, Leaf Cells, and Nets. After clicking the button, a tree structure will be displayed on the left side of the UCE, as shown in the figure below.

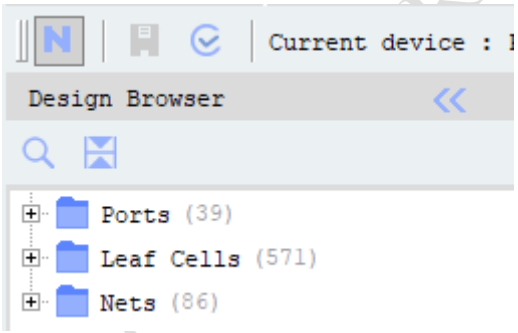


Figure4-4

The [Save] button is used to save all the constraints entered in the current UCE into the constraint file .fdc, without filtering out the incorrect constraints in the check report. Therefore, if users want all the generated constraint files to be correct, please view the check report first.

The [Check Constraints] button is used to check the correctness of the constraints entered in the UCE, and the results will be displayed on the UCE interface in the form of a pop-up check report (where errors will be highlighted in red font).

Constraint information is saved in the .fdc file.

4.1 Physical Constraint

Selecting [Device] in the UCE interface allows for physical constraints, as shown in the figure below.

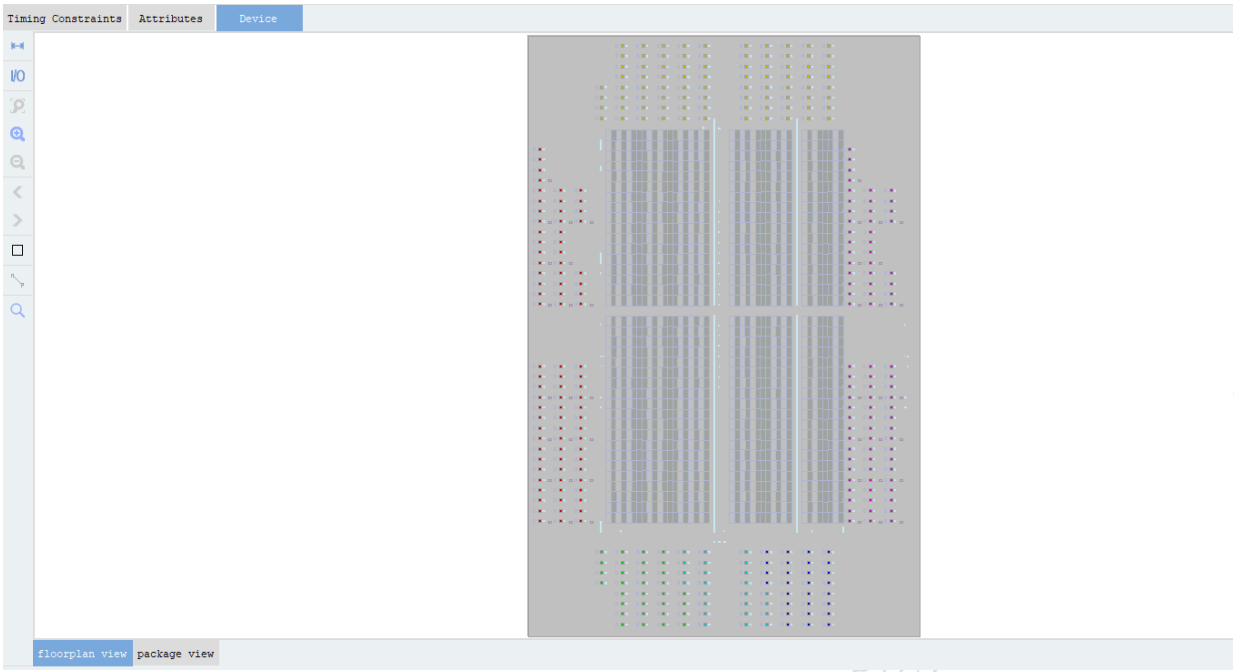


Figure4-5

4.1.1 Toolbar

The functions of the toolbar on the left side of the [Device] interface, from top to bottom, are as follows:

[Device Browser]: Displays design information and Device hierarchy;

[I/O Table]: Constrains IO;

[Zoom]: Includes View All, Zoom In, and Zoom Out buttons, corresponding to the functions of displaying the overall placement of resources, zooming in, and zooming out;

[History]: Displays Previous View and Next View, used to switch the views of historical operations of floorplan view or package view;

[Mode]: Region Mode button. Click to switch to region mode, which can impose region constraints on instances.

[Show Differential IO]: Differential button. Click to display the distribution of differential IO in package view.

[Search Inst]: Search button. Click to search for Device information in the Device interface through the search box.

4.1.2 View

users can select [floorplan view] window and [package view] window on the [Device] interface. The [floorplan view] window is used to display the resource locations of the chip, with IO resources

around and logic resources in the middle. Users can perform interface constraint operations such as drag-and-drop and canceling constraints. Instances dragged into the [floorplan view] window will be highlighted.

The [package view] window is used to display the pin package locations of the chip, and the constrained IO dragged into it will be highlighted. This is shown in the following figure.

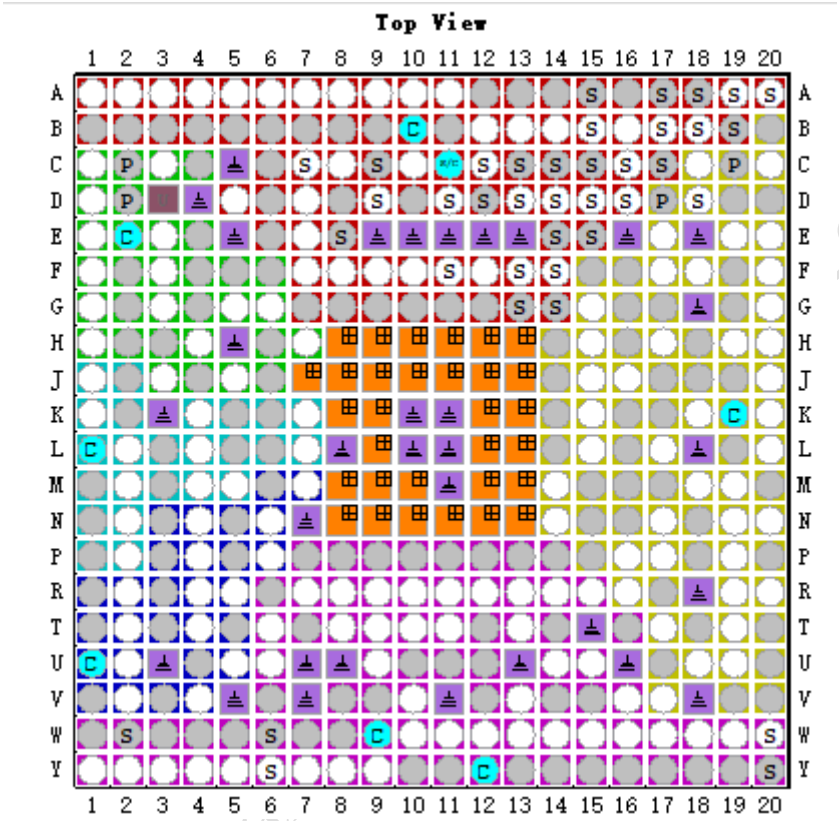


Figure4-6

The [floorplan view] window and the [package view] window can be switched using the buttons below the [Device] interface, and zooming operations can be performed by scrolling the mouse wheel while holding down "Ctrl" or by clicking the [Zoom] button in the toolbar.

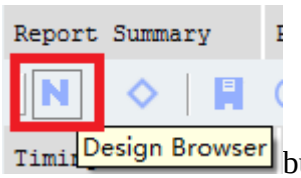
4.1.3 Instance constraint

Instance constraints include device location constraints and location preserve and can only be operated after running Post Synthesize UCE.

4.1.3.1 Location Constraint

Device location constraints refer to the positioning of base cells such as LUTs, FFs, PLLs, DRMs, and IOs within the device. The command format in the constraint file is as follows:

Example: define_attribute {i:instance} {PAP_LOC} {CLMA_5_4:FGA}

Additionally, location constraints support graphical user interface operations. By clicking the [Design Browser] 

button, instances from the Leaf Cells can be dragged onto the corresponding device in the floorplan view on the right. This is shown in the following figure.

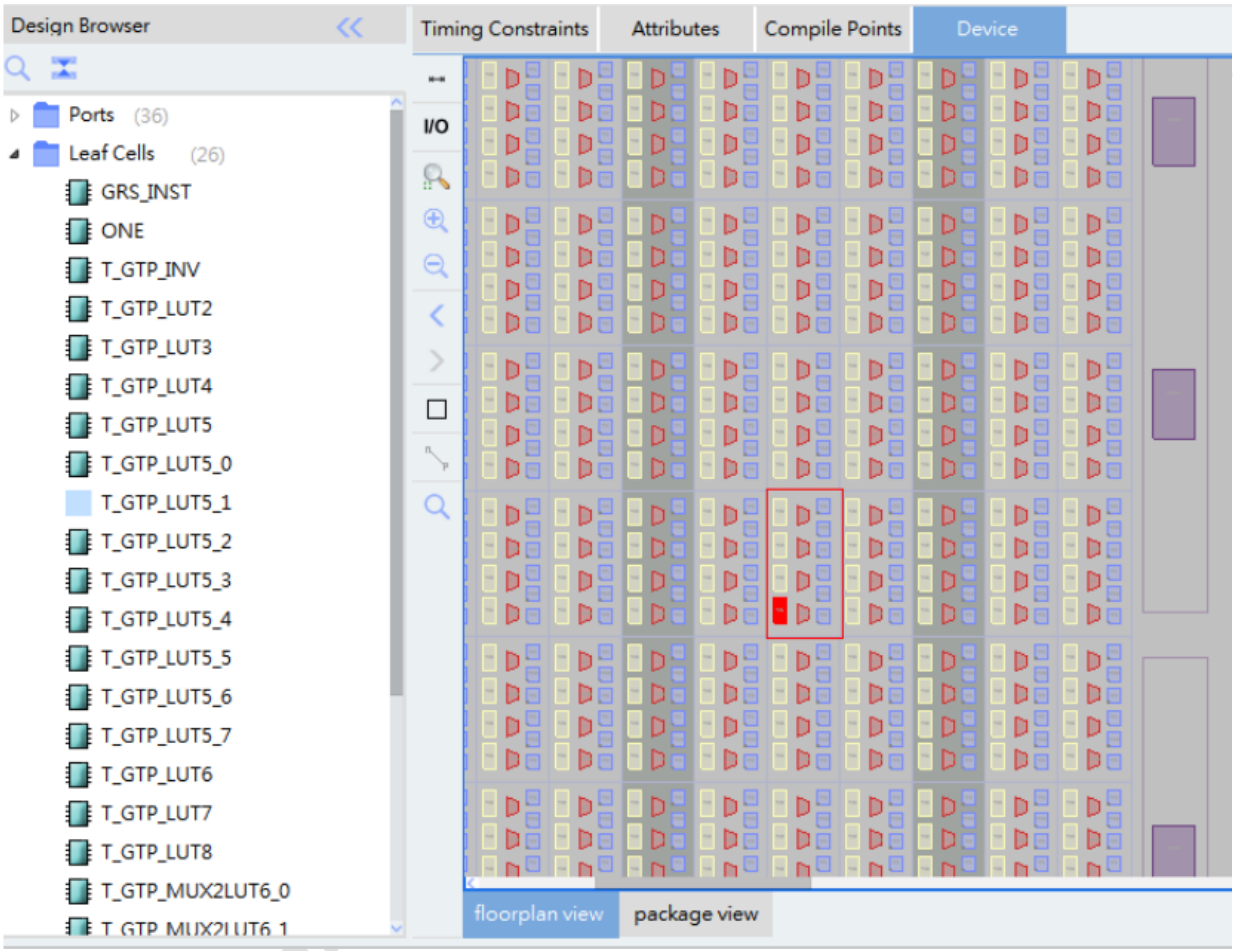


Figure4-7

Figure4-8

Users can also drag signals from "Ports" to the corresponding IO in the package view on the right to constrain the IO locations, as shown in the figure below.

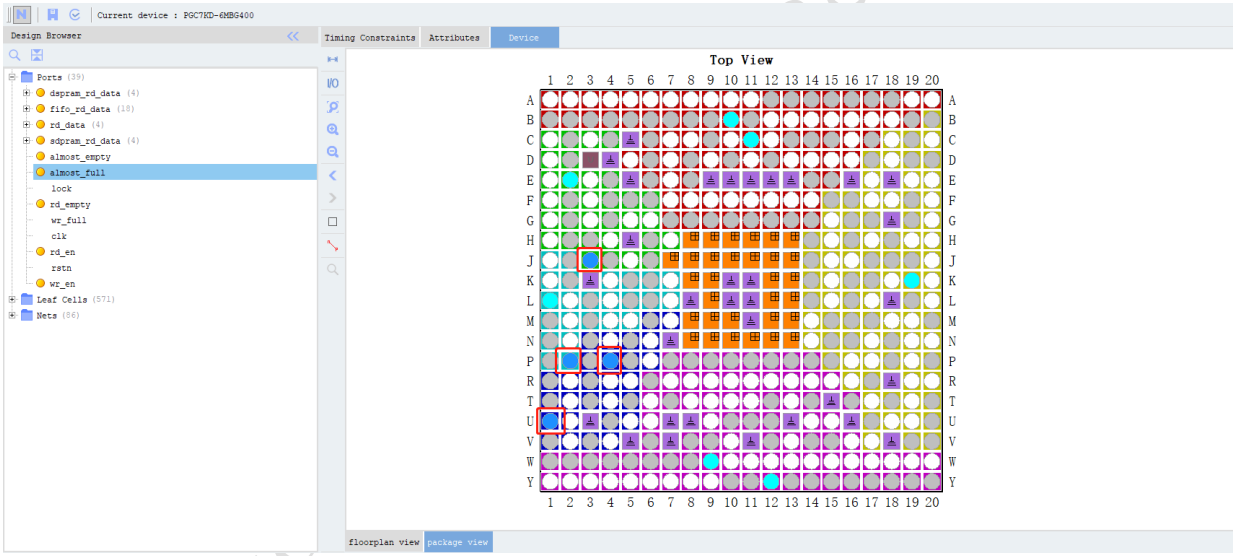


Figure4-9

4.1.3.2 Location Preserve

Device location preserve means preserving the positions of base cells such as CLMs, PLLs, DRMs, and IOs within the device, preventing them from being occupied. The command format in the constraint file is as follows:

Example: define_global_attribute {PAP_SITE_PRESERVE} { CLMA_6_4;CLMA_6_7}

Additionally, location preserve supports graphical user interface operations. Right-click on any CLM, PLL, DRM, or IO and select [Set Preserve] in the drop-down menu to preserve it from being occupied, as shown in the figure below.

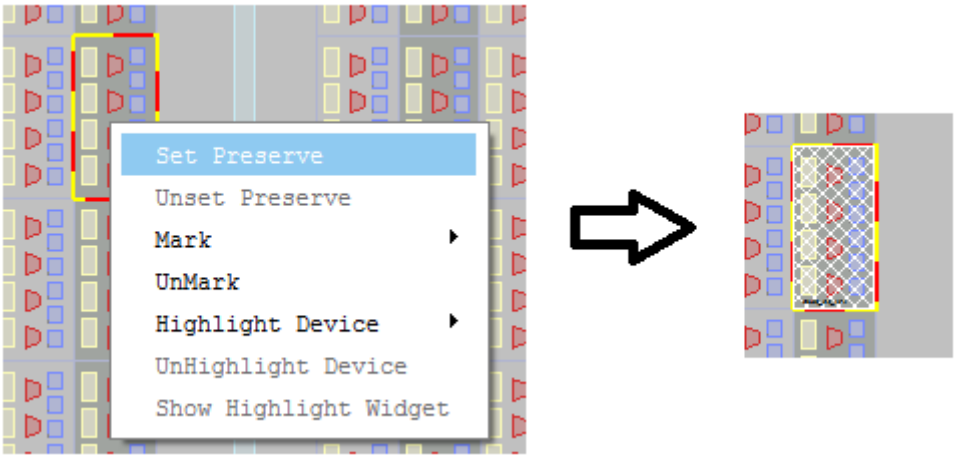


Figure4-10

To release the location preserve of a cell, right-click on the cell and select [Unset Preserve] in the drop-down menu, as shown in the figure below.

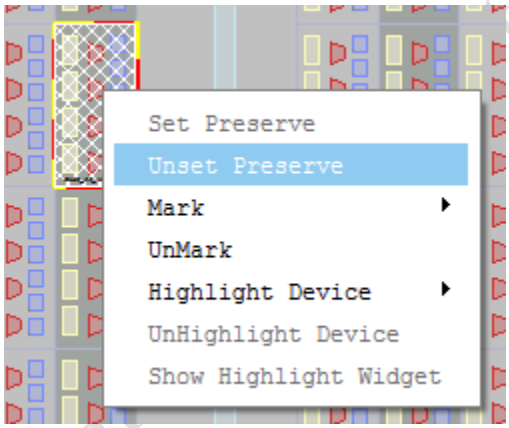


Figure4-11

4.1.4 Region Constraint

For placement purposes, users can constrain certain modules within a specific region before synthesis, which requires the Post Synthesize UCE to be run before operation. The command format in the region constraint is as follows:

Example: define_global_attribute {PAP_REGION} {region1(x0,x1,y0,y1);}

Herein, (x0,y0) corresponds to the coordinates of the bottom-left corner of the region, and (x1,y1) corresponds to the coordinates of the upper-right corner.

The steps for graphical user interface operations are as follows:

1. Click the "Region Mode" button in the toolbar.
2. Right-click in the floorplan view region and select [Draw Region], at which point the mouse cursor icon will change to "+".

3. Press the left mouse button in the floorplan view region to circle the required constraint range and click [OK], as shown in the figure below.

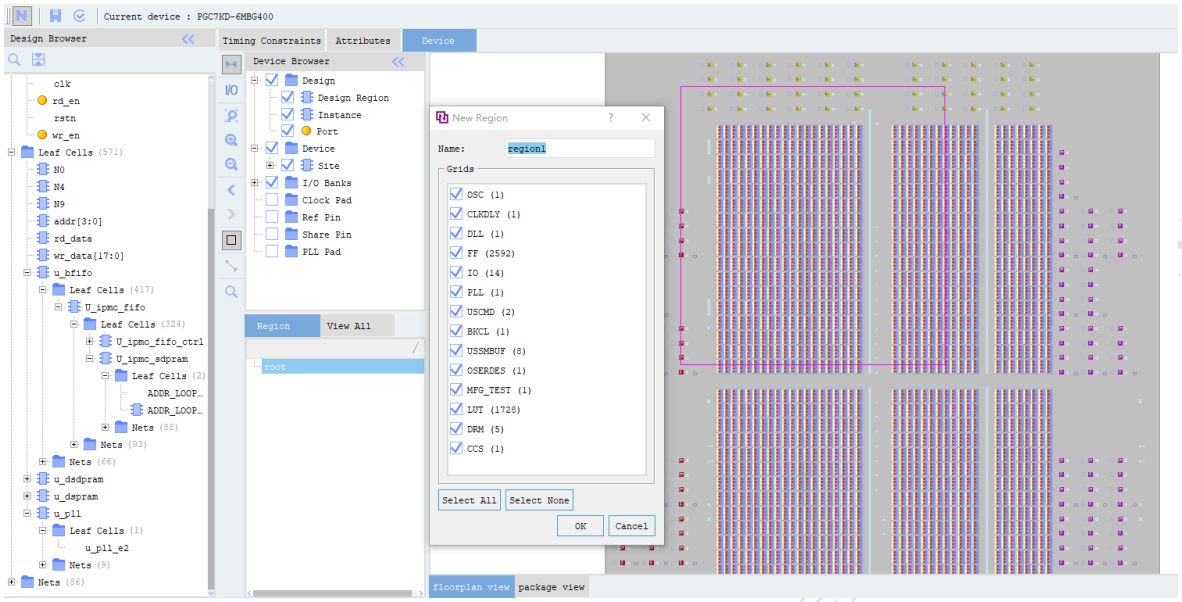


Figure4-12

4. At this point, instances from Leaf Cells in the [Design Browser] can be dragged into the designated region, as shown in the figure below;

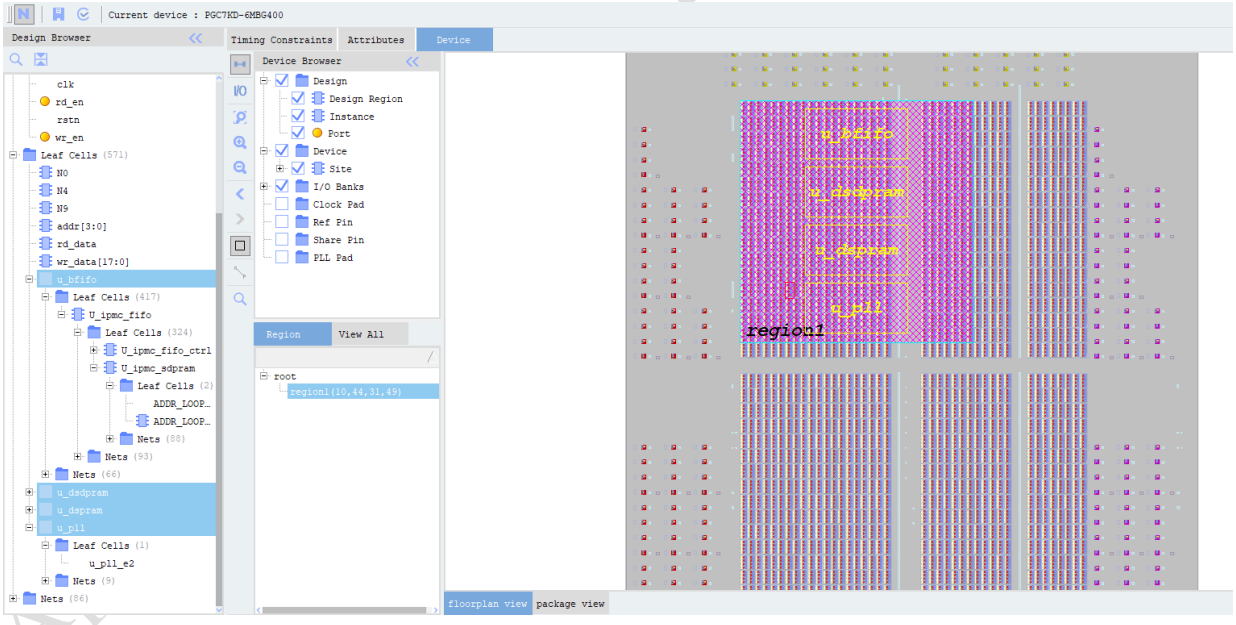


Figure4-13

During the dragging process, if the mouse displays "+", it indicates that dragging is possible. If the mouse displays "Ø", it indicates that dragging is not possible. The possible reason is that the designated region does not include the corresponding device type or the software temporarily cannot recognize the type of instance.

In the [Region], right-click on the region name and select New Region, Edit Region, Delete Region, or Show Region Info, as shown in the figure below.

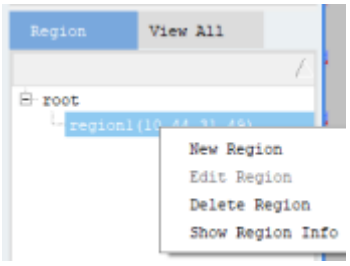


Figure4-14

[New Region]: Create a new region

[Edit Region]: Edit the range of the region. If instances have already been dragged in, it cannot be edited

[Delete Region]: Delete the designated region

[Show Region Info]: Display the resources contained in the region

4.1.5 I/O Constraint

In the [Device] interface, clicking the button 

[I/O Table] in the toolbar will display the I/O Table, and clicking again will hide the I/O Table. users can configure I/Os in the I/O Table, and the constraints are also synced to package view and Design Browser in real-time.

Constraint information of the I/O Table is also represented as attributes in the .fdc file, with the columns described as follows:

I/O NAME: Describes the name of the port, designed and defined by users;

I/O DIRECTION: Displays the I/O type, which can be input, output, or inout;

LOC: Constrains the location of the pin;

BANK: The bank where the pin is located;

VCCIO: I/O input and output voltage;

IOSTANDARD: I/O level standard, related to VCCIO;

DRIVE: Output drive current, related to IOSTANDARD;

BUS_KEEPER: Sets pull-up, pull-down, left open, and hold;

SLEW: Conversion rate, includes fast and slow;

HYS_DRIVE_MODE: Input hysteresis mode;

DIFF_IN_TERM_MODE: Input matching resistance settings for differential inputs, supported only by BANK2;

OPEN_DRAIN: Drive mode selection, whether to use open-drain output;

IN_DELAY: Input delay;

OUT_DELAY: Output delay;

CP_CLAMP: Clamping function, only effective for input ports;

IO_REGISTER: IO internal registers, including input, output, and tri-state registers. When checked, the first-level registers connected to the port in the code can be mapped within the IOL, thus not occupying the CLM's FF;

VIRTUAL_IO: Virtual I/O is employed during the initial stages of user design or project migration to assess resource utilization. Since the quantity and placement constraints of I/O are not considered at these stages, virtual I/O can be set.

4.2 Timing Constraint

In the UCE interface, select [Timing Constraints] to apply timing constraints, as shown in the figure below.

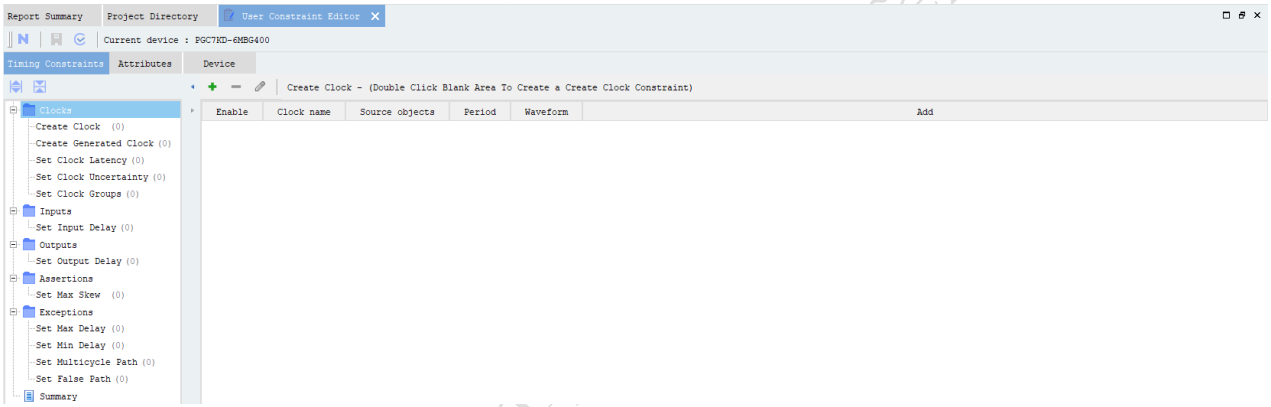


Figure4-15

The [Timing Constraints] interface is divided into two parts. The tree structure on the left displays commands and their quantities. The table on the right displays the content of each command line. Each command corresponds to a row in the table.

The tree structure displays the supported timing commands, which are divided into 5 categories: Clocks, Inputs, Outputs, Assertions, and Exceptions, with a total of 12 timing commands. Clicking [Summary] in the tree structure allows for a preview of the commands.

Each command has its own command creation window interface, within which users can input content for every option of the command. After the operation is completed, click [OK]. The software will automatically check the syntax. If there is a command error in the window interface, an error message will pop up. Only correct commands can be passed from the window interface to the list.

Each timing constraint option is introduced below.

4.2.1 create_clock

The function of "create_clock" is to create a clock for timing analysis. Double-click [Create Clock] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window, as shown in the figure below.

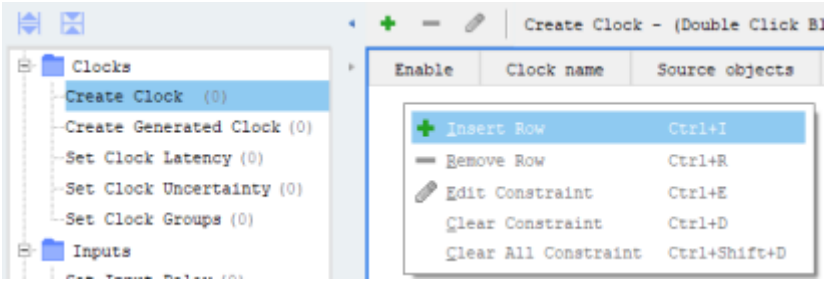


Figure4-16

Create the create_clock command in the Create Clock window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

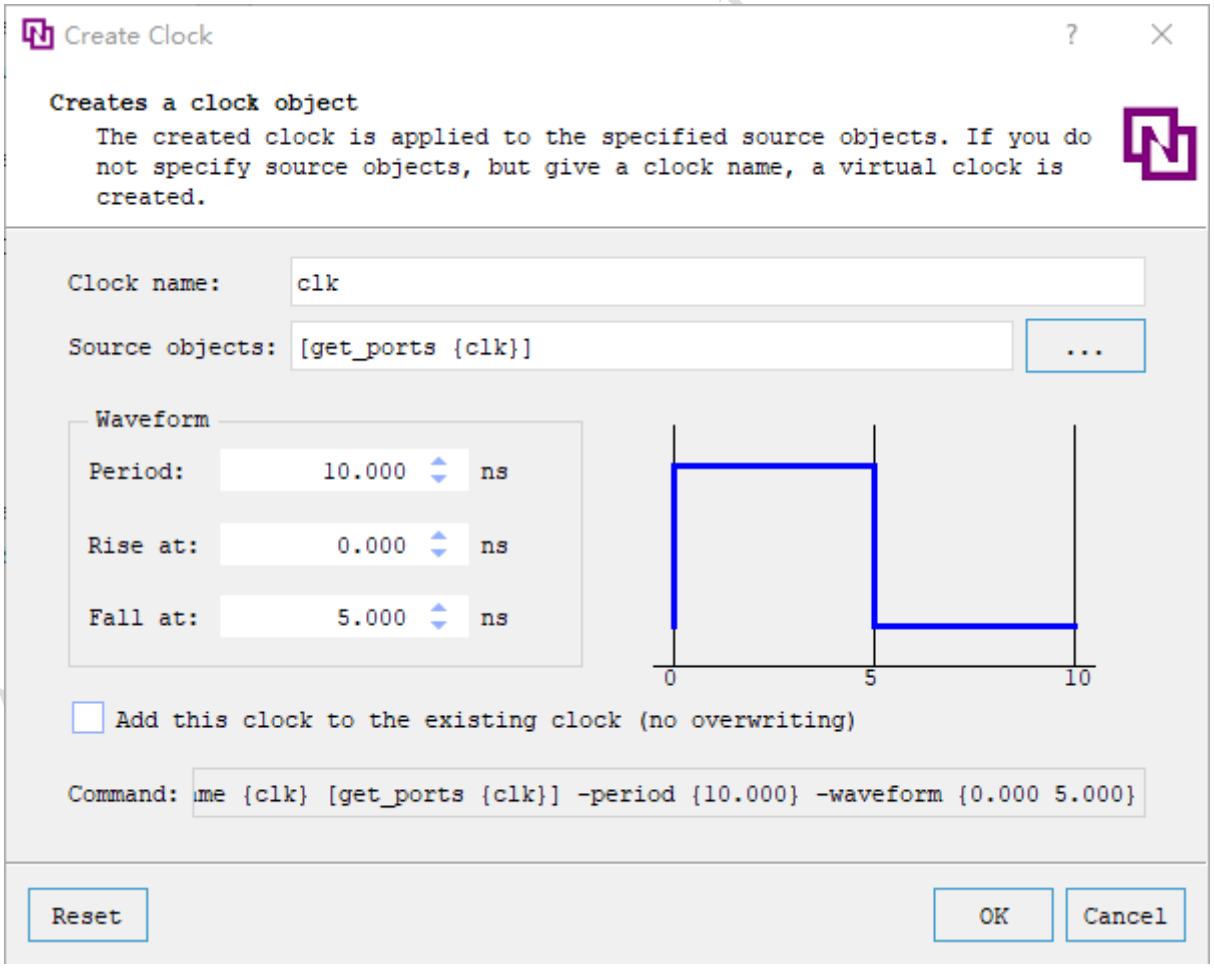


Figure4-17

When setting [Source objects], click the "..." on the right, and in the [Specify Source objects]

window that pops up, users can further specify the object's type and name, as shown in the figure below.

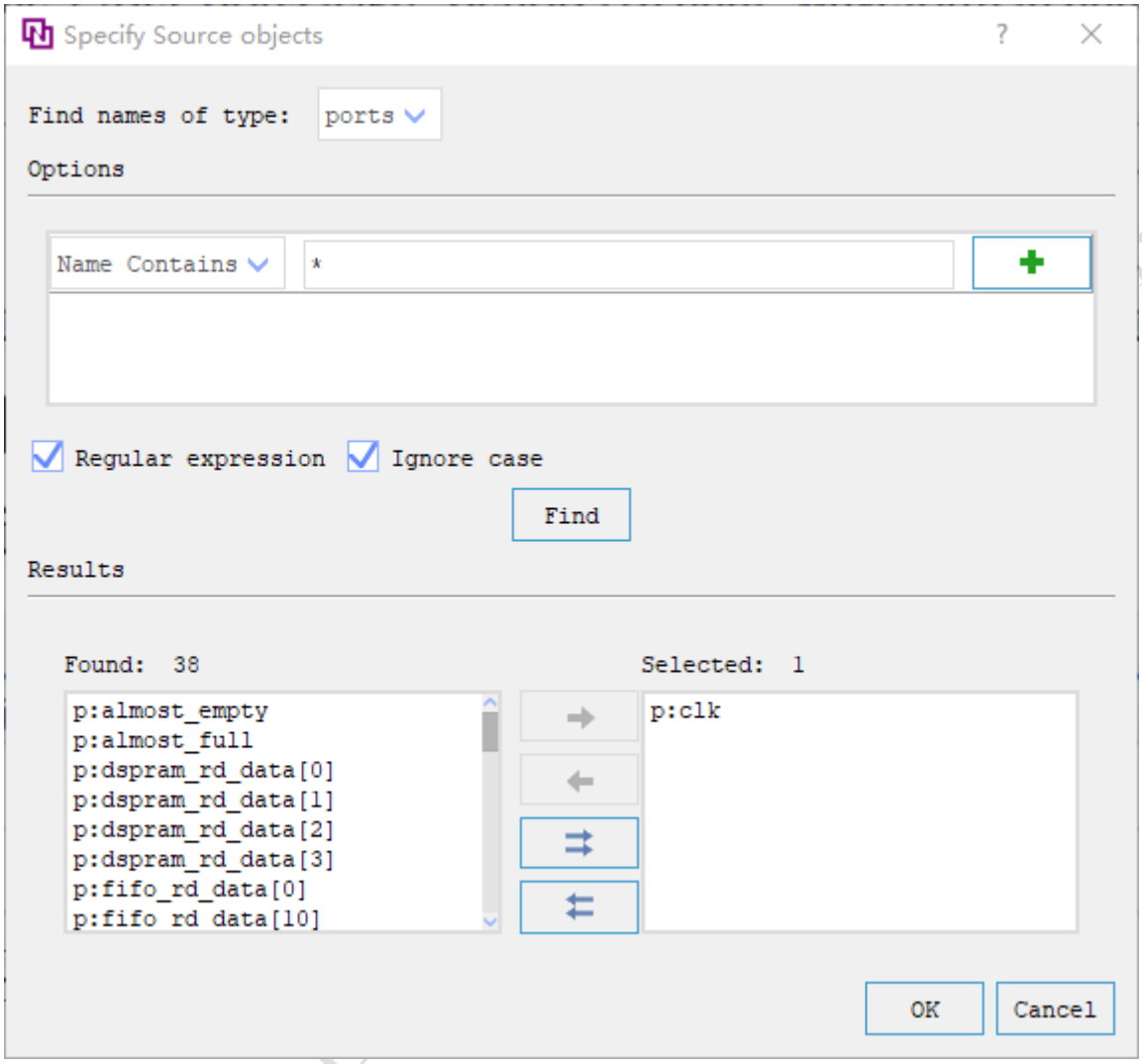


Figure4-18

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

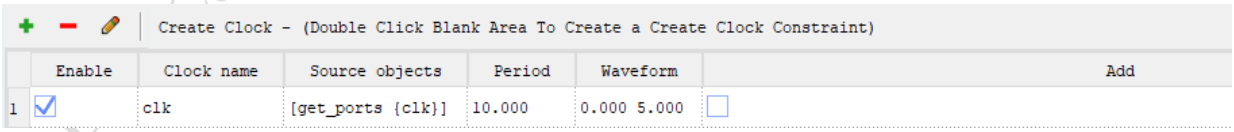


Figure4-19

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

create_clock -name {clk} [get_ports {clk}] -period {10.000} -waveform {0.000 5.000}

The correspondence between the list information and the command line is as follows.

Table 4-1

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Clock name	-name {clock_name}	The name of the clock created. If the user does not define it, the default name is the same as the first source object in source objects. Therefore, if -name is empty, users must specify source objects; if the user has not defined a source object, the user must define -name, in which case the clock is a virtual clock.
Source Objects	[get_pins/ get_ports]	The source of the clock created, which can be a pin or port (note that it must be obtained through get_pins or get_ports). When it is empty, a virtual clock is created.
Period	-period {period_value}	The period of the clock created. This is a required option. If it is not defined, the software will report an error.
Waveform	-waveform {edge_list}	The positions of the clock's rising edge and falling edge, with the rising edge first, followed by the falling edge, existing in pairs. The format is as follows: -waveform {time_rise time_fall time_rise time_fall ...}
Add	-add	Indicates that multiple clocks can be defined on the same pin. If this parameter is not added, the clock defined later will override the previous one. If a clock has already been defined on the source_object previously and the user has not added the -add parameter, indicating that multiple clocks need to be defined on that pin or port, the software will issue a warning (UCE will not check this; a warning will be given when parsing the timing constraint file later).

4.2.2 create_generated_clock

The function of create_generated_clock is to create a clock derived from "Create Clock"; when creating a derivative clock, a Create Clock must first be created.

Double-click [Create Generated Clock] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window, as shown in the figure below.

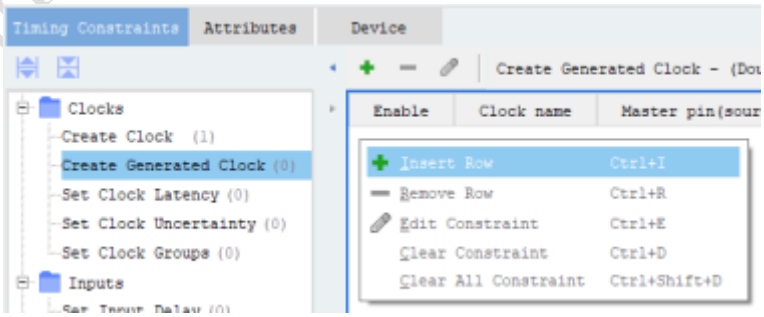


Figure4-20

Create the "create_generated_clock" command in the [Create Generated Clock] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface

is as follows:

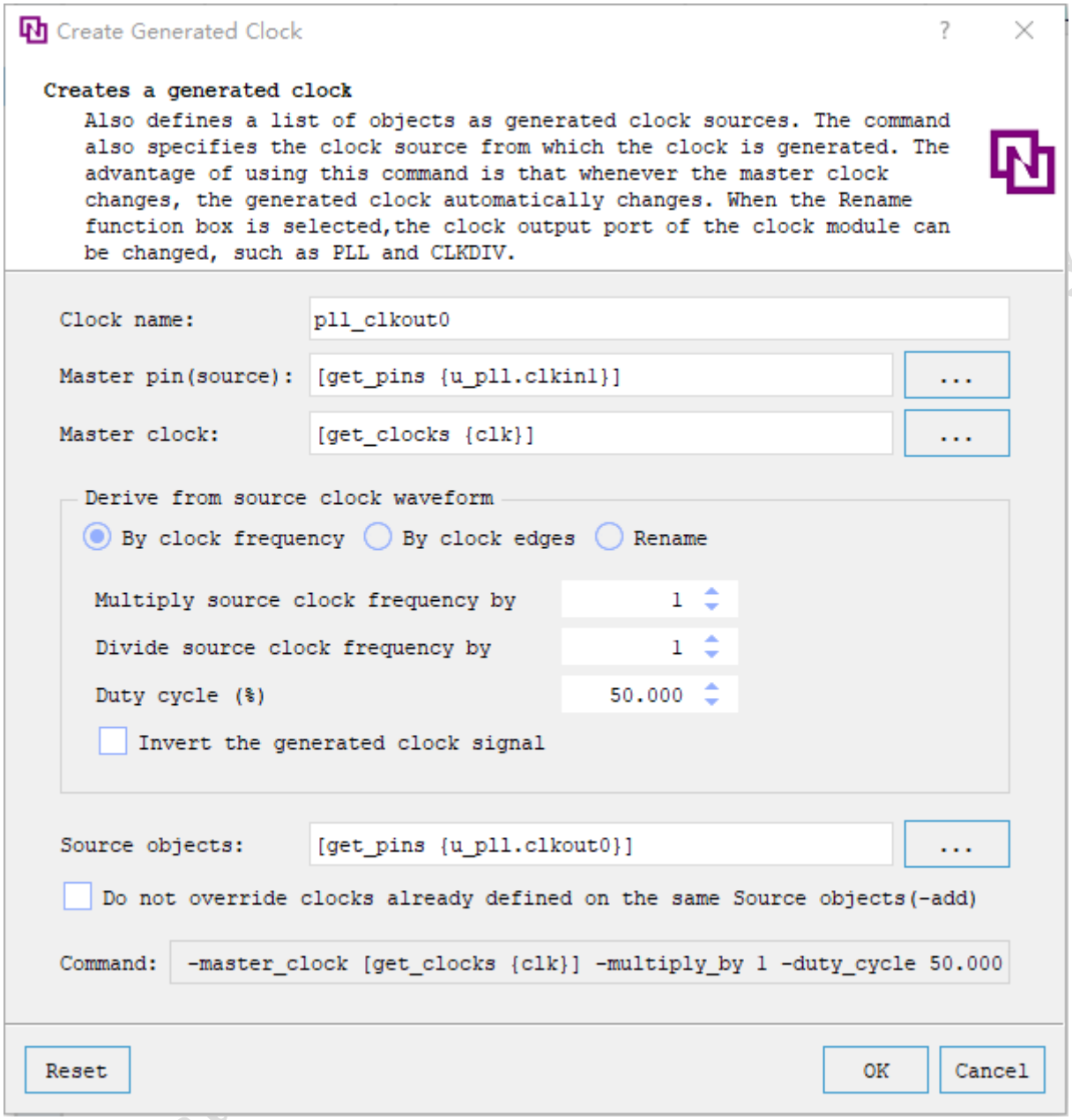


Figure4-21

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

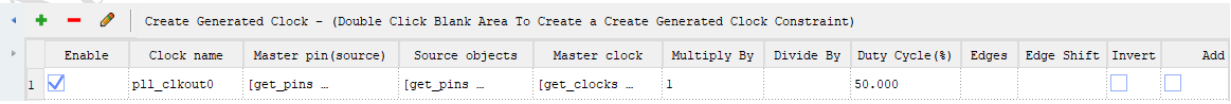


Figure4-22

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

create_generated_clock -name {pll_clkout0} -source [get_pins {u_pll.clkin1}] [get_pins {u_pll.clkout0}] -master_clock [get_clocks {clk}] -multiply_by {1} -duty_cycle {50.000}

The correspondence between the list information and the command line is as follows.

Table 4-2

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Clock name	-name {clock_name}	The name of the generated clock based on the source clock
Master Pin (Source)	-source {master_pin}	Specify the pin or port of the source clock that drives the generated clock
Source Objects	[get_pins/ get_ports]	The pin or port of the generated clock created
Master Clock	-master_clock [get_clocks {clock}]	Source clock. Master pin can define multiple source clocks, but the generated clock only comes from one of those source clocks. When there is only one source clock on the master pin, specifying the master clock option is not necessary. The principle of "declare before use" must be followed, meaning the clock must be created with create_clock first.
Multiply By	-multiply_by {factor}	Multiplier ratio, which is an integer greater than 0 that reduces the clock period by a specified multiple. Multiplier ratio not only changes the period but also proportionally reduces the waveform.
Divide By	-divide_by {factor}	Division factor, which is an integer greater than 0 that increases the clock period by a specified multiple. Division not only causes changes in the period but also changes in the waveform. When the division factor is 2 or an exponential multiple of 2, the duty cycle is set to 50%.
Duty Cycle	-duty_cycle {percent}	The duty cycle is described in percent (%), which is the ratio of the high level to the entire period. The duty cycle can only be used with the -multiply_by option.
Edges	-edges {edge_list}	Consistent with the waveform format in create_clock, with the addition of the start position of the second period. PDS requires that the edge_list must contain 3 elements, including the rising and falling edges of the first period and the start of the rising edge of the second period. PDS can calculate the period of the generated clock from this.
Edge Shift	-edge_shift {shift_list}	A family of floating-point numbers are specified to represent the offset of each edge, with the default unit being nanoseconds (ns), and must be used with -edges. If the generated clock needs to implement phase shift, "-edge_shift" is required, and the number of elements in the "-edge_shift" list must match the number in "-edges".
Invert	-invert	Indicates that the clock signal of the generated clock is inverted relative to the source clock.
Add	-add	If a clock of the same type has already been defined for this pin, adding the "-add" option will include a new clock on top of the previous one. If the user does not add this option, the newly created clock will replace the original one. If multiple clocks of the exact same type are defined on the same source object, and then the clocks are independent of each other.

4.2.3 set_clock_latency

set_clock_latency is commonly used to constrain the delay between the off-chip clock source and the chip pins.

Double-click [Set Clock Latency] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window, as shown in the figure below.

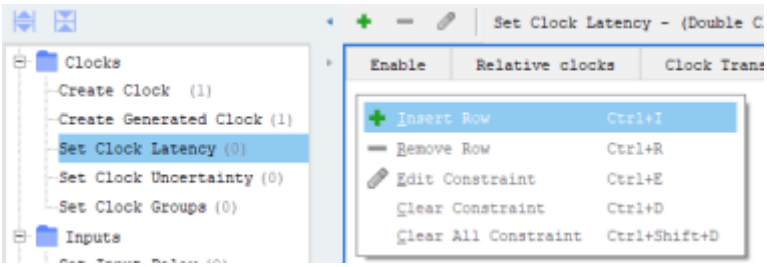


Figure4-23

Create the "set_clock_latency" command In the Set Clock Latency window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

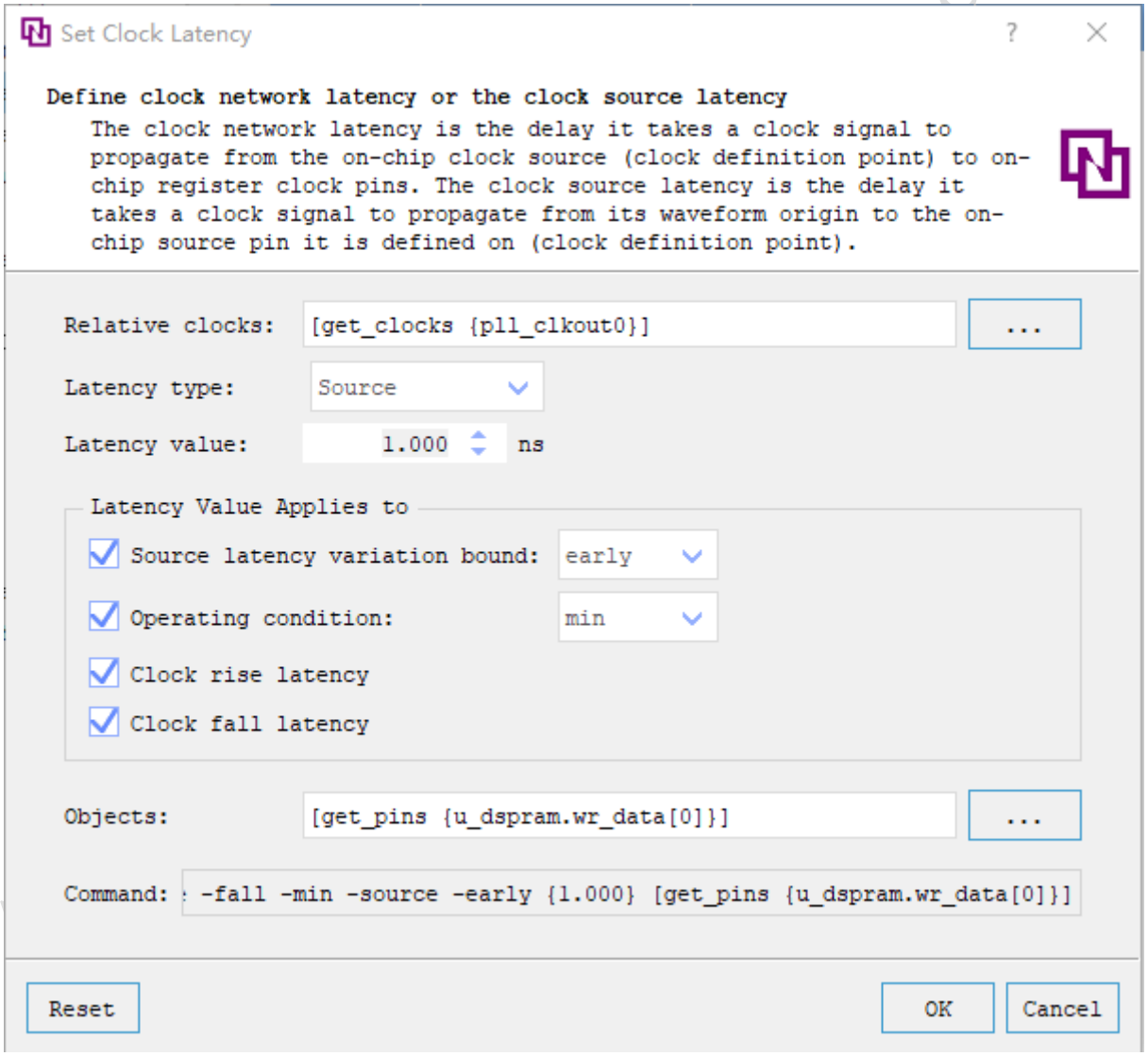
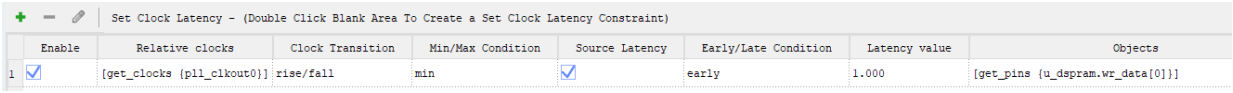


Figure4-24

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.



At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_clock_latency {1.000} -rise -fall -min -source -early -clock [get_clocks {pll_clkout0}] [get_pins {u_dspram.wr_data[0]}]

The correspondence between the list information and the command line is as follows.

Table 4-3

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Relative Clocks	-clock [get_clocks {clock_name}]	The clock affected by this command must follow the principle of "declare before use", meaning the clock must be created with "create_generated_clock" or "create_clock" before use.
Clock Transition	-rise/-fall	Indicates that the latency is applied at the rising or falling edge of the trigger clock
Min/Max Condition	-max/-min	-min indicates fast corner, -max indicates "slow corner"; if -min and -max are not specified, both "fast corner" and "slow corner" commands will take effect by default, but -min and -max cannot be specified at the same time
Source Latency	-source	Only source latency is supported
Early/Late Condition	-early/-late	-early indicates that the latency is the left offset of the clock, -late indicates the right offset of the clock; if -early and -late are not specified, the default is that both left and right offsets of the clock are effective, but -early and -late cannot be specified at the same time
Latency Value	{latency_value}	Latency value, float type, measured in ns
Objects	[get_pins/get_ports]	The clock defined on the corresponding pin/port can be obtained through get_pins or get_ports, or the latency can be directly defined on the clock; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first

4.2.4 set_clock_uncertainty

set_clock_uncertainty is used to constrain the skew of the clock path.

Double-click [Set Clock Uncertainty] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window, as shown in the figure below.

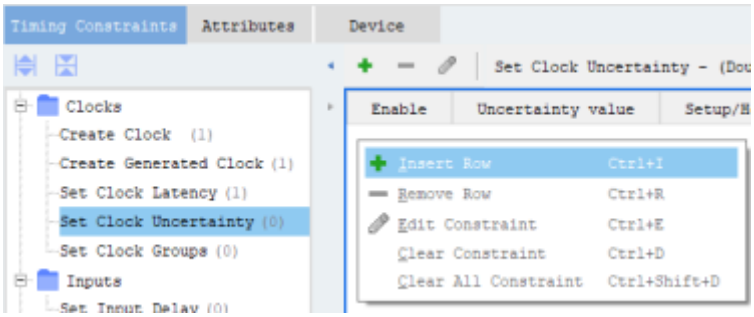


Figure4-25

Create the "set_clock_uncertainty" command in the [Set Clock Uncertainty] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

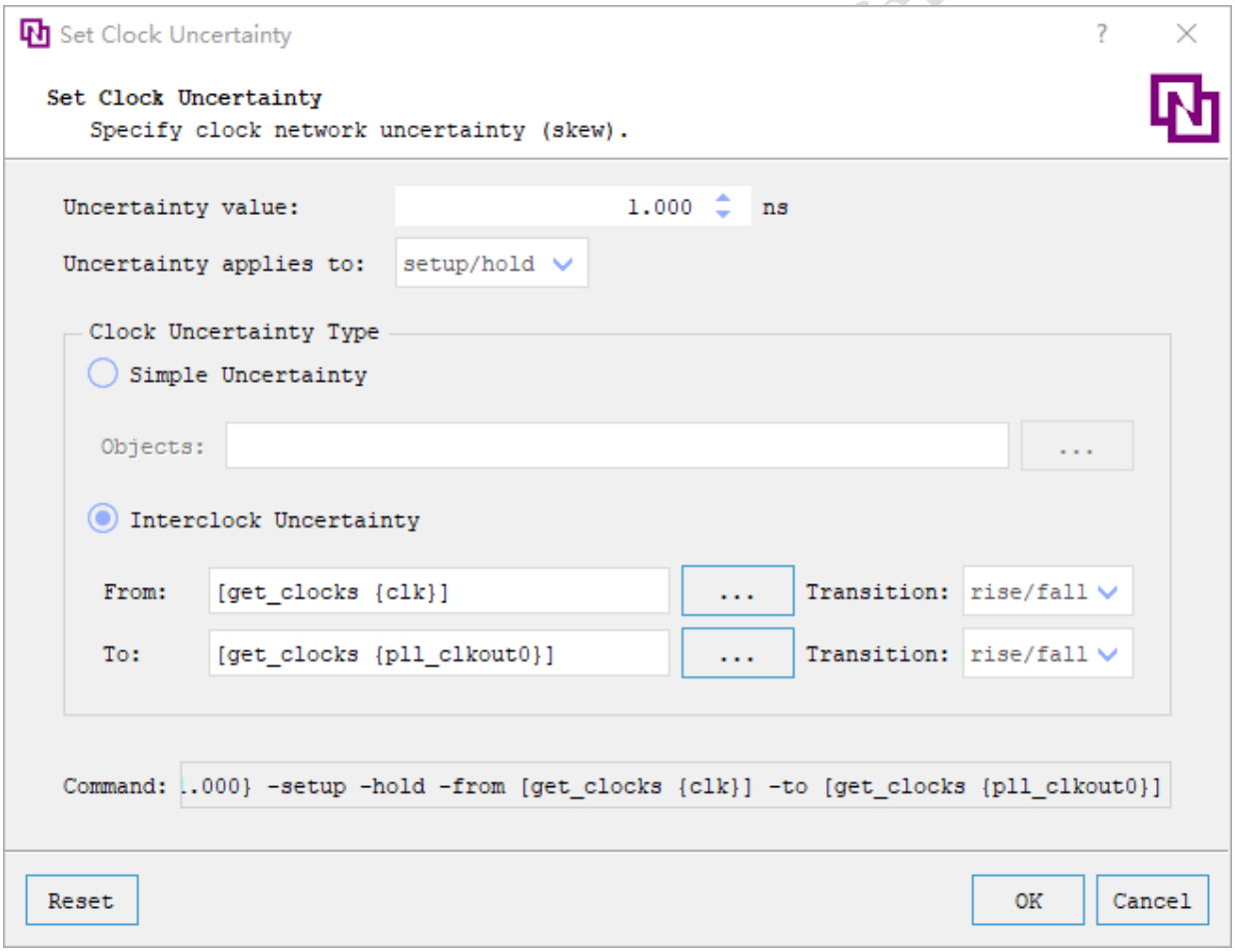


Figure4-26

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

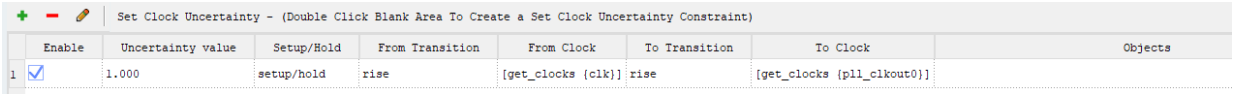


Figure4-27

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_clock_uncertainty {1.000} -rise_from [get_clocks {clk}] -rise_to [get_clocks {pll_clkout0}] -setup -hold

The correspondence between the list information and the command line is as follows.

Table 4-4

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Uncertainty Value	{uncertainty_ value}	Uncertainty value, float type
Setup/Hold	-setup/-hold	-setup indicates that the uncertainty value is used in setup analysis; -hold indicates that the uncertainty value is used in hold analysis
From Transition	-rise_from/ -fall_from	Indicates that the uncertainty is applied at the rising or falling edge of the launch clock
From Clock	[get_clocks {clock}]	Indicates the launch clock, which is obtained through get_clocks; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first
To Transition	-rise_to/ -fall_to	Indicates that the uncertainty is applied at the rising or falling edge of the capture clock
To Clock	[get_clocks {clock}]	Indicates the capture clock, which is obtained through get_clocks; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first
Objects	[get_pins/ get_ports/ get_clocks]	If multiple clocks are defined on the pin or port of the trigger clock and the user only wants to define uncertainty for one of the clocks, and then the object can be a clock defined by the user. If the user wants to define the same uncertainty for all clocks on the pin of the trigger clock, and then the object can be a pin or port; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first

4.2.5 set_clock_groups

set_clock_groups can define multiple clocks in different groups; clocks in different groups are mutually exclusive or asynchronous, and paths of asynchronous clocks are not analyzed during timing analysis.

Double-click [Set Clock Groups] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Creates the "set_clock_groups" command in the [Set Clock Groups] window, with the top part of the window providing an introduction to the command. The middle part is for creating the

command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

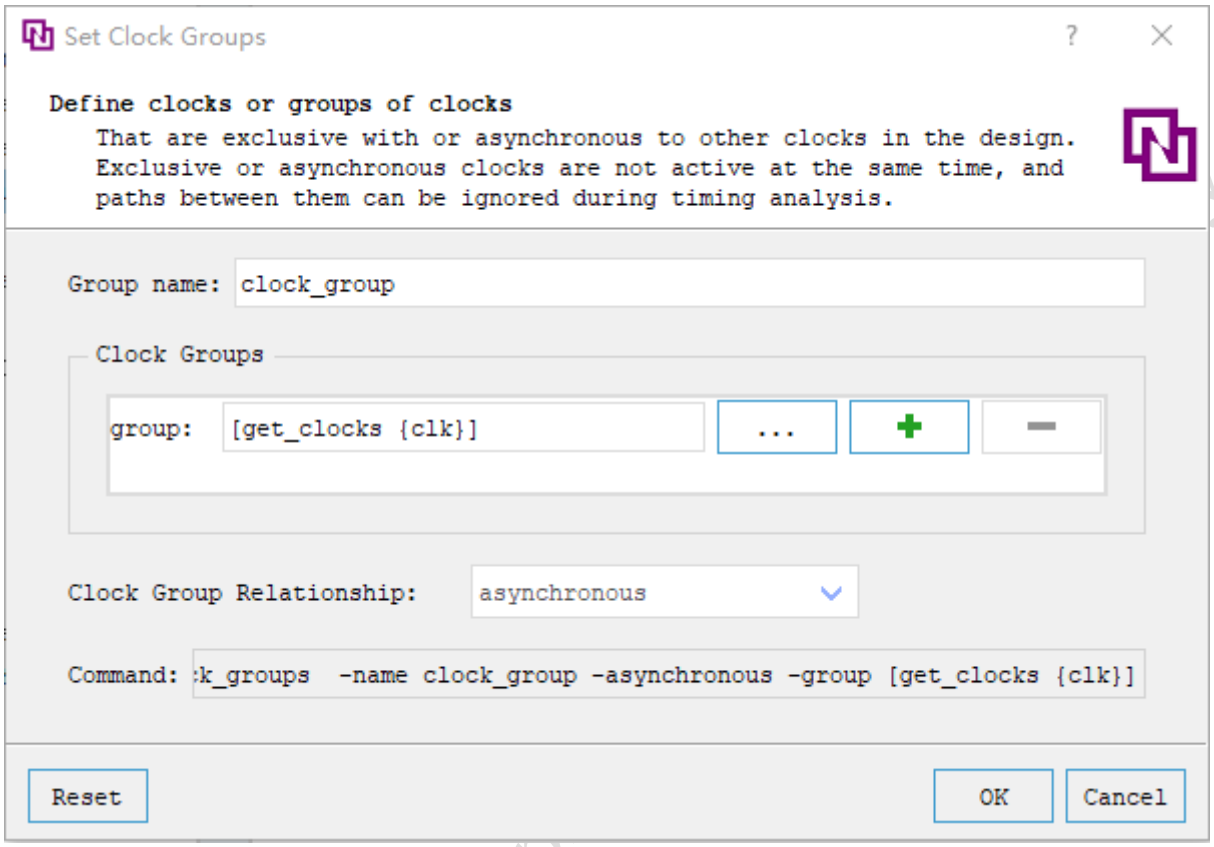


Figure4-28

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

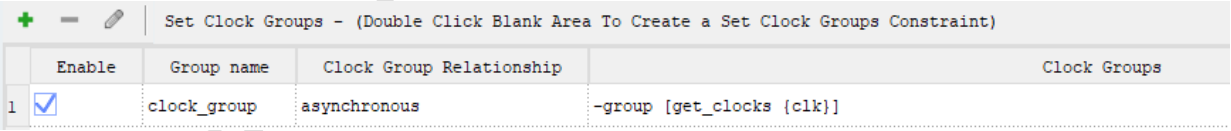


Figure4-29

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_clock_groups -name clock_group -asynchronous -group [get_clocks {clk}]

The correspondence between the list information and the command line is as follows.

Table 4-5

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Group name	-name group_name	Defines the name of the clock group
Clock Group	-asynchronous	Clock group type. Currently, there is only one type: -asynchronous

List Attribute	Command Parameter	Description
Relationship		
Clock Groups	-group [get_clocks {clock}]	Specifies the clock group, obtained through get_clocks

4.2.6 set_input_delay

set_input_delay is used to specify the input delay for an input pin or port relative to a clock edge.

Double-click [Set Input Delay] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_input_delay" command in the [Set Input Delay] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

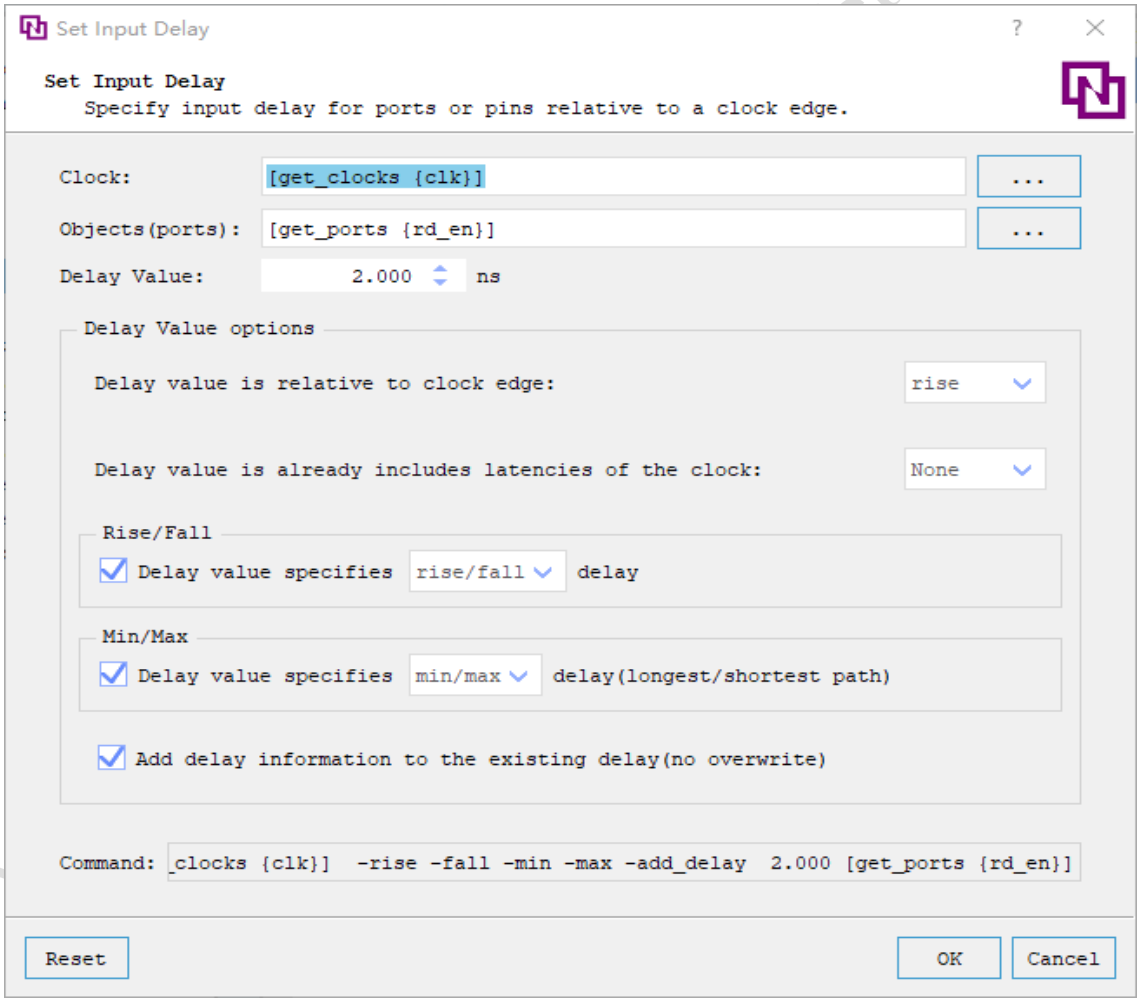


Figure4-30

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

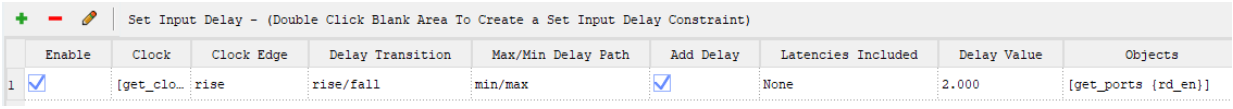


Figure4-31

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_input_delay {2.000} [get_ports {rd_en}] -rise -fall -min -max -add_delay -clock [get_clocks {clk}]

The correspondence between the list information and the command line is as follows.

Table 4-6

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Clock	-clock [get_clocks {clock_name}]	Indicates the clock the delay applies to, which is obtained through get_clocks; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first
Clock Edge	-clock_fall	Indicates that the delay applies to the falling edge of the clock
Delay Transition	-rise/-fall	Indicates that the delay is valid at the rising/falling edge of the input signal
Max/Min Delay Path	-max/-min	-max indicates that the user-defined delay value is the maximum value, and the software only needs to perform setup analysis -min indicates that the user-defined delay value is the minimum value, and the software only needs to perform hold analysis
Add Delay	-add_delay	If the user has already set a set_input_delay constraint on the input port, adding this option means the software will preserve the previously set constraint values. For -max, the delay value will be calculated using the maximum of all values under the same conditions; for -min, the calculation will use the minimum of all values under the same conditions.
Latencies Included	-source_latency_included	Indicates that [Set Clock Latency] has been taken into account
Delay Value	{delay_value}	Delay value, float type
Objects	[get_ports /get_pins]	Indicates the pin or port the delay applies to, obtained through get_pins or get_ports

4.2.7 set_output_delay

set_output_delay is used to specify the output delay for an output pin or port relative to a clock edge.

Double-click [Set Output Delay] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_output_delay" command in the [Set Output Delay] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

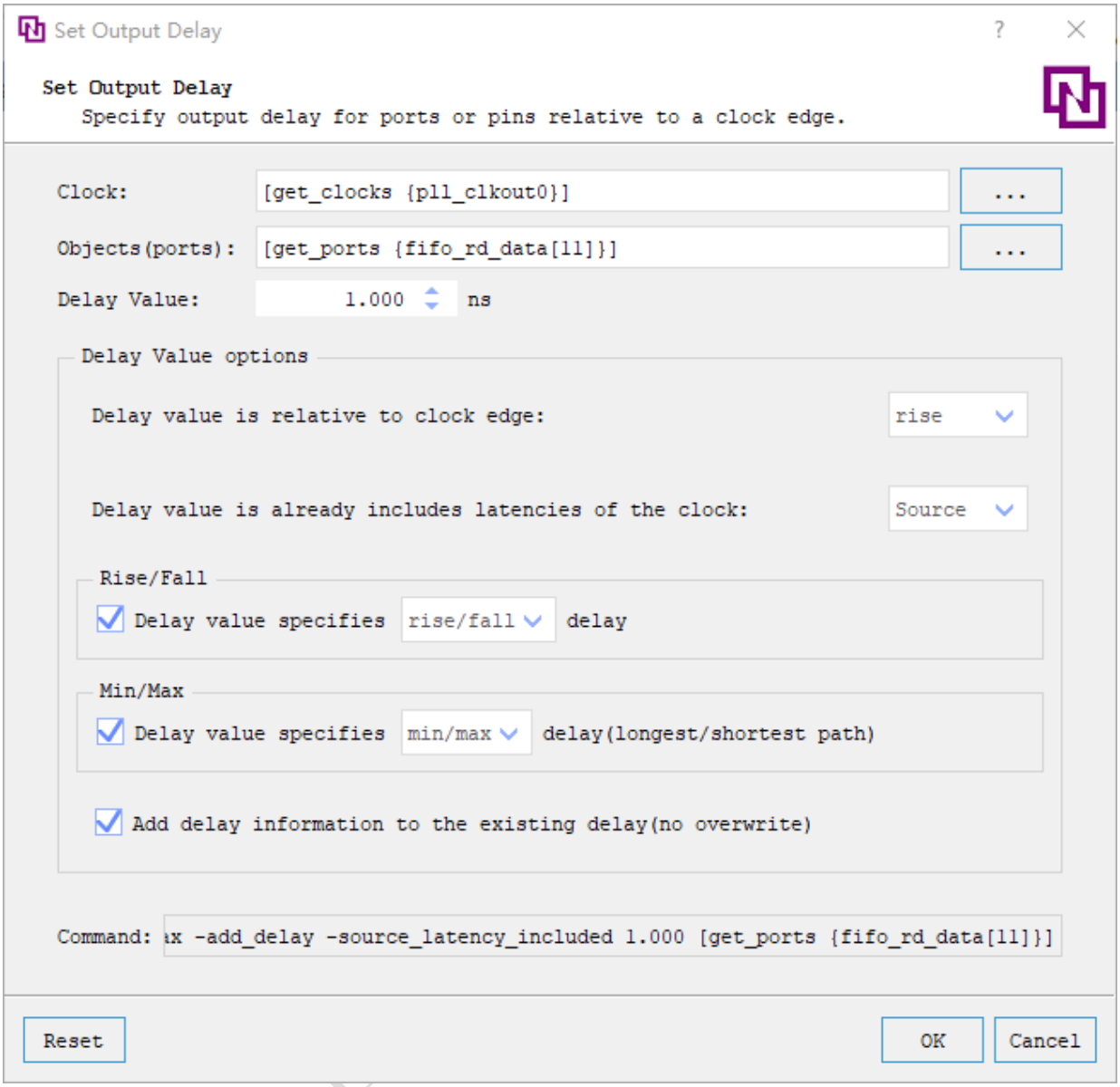


Figure4-32

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

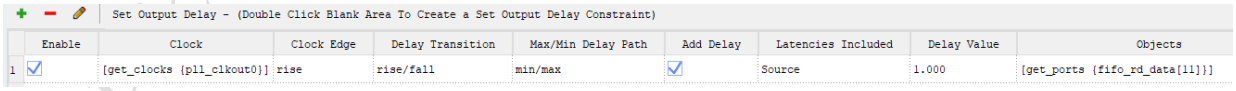


Figure4-33

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_output_delay {1.000} [get_ports {fifo_rd_data[11]}] -rise -fall -min -max -add_delay -clock [get_clocks {pll_clkout0}] -source_latency_included

The correspondence between the list information and the command line is as follows.

Table 4-7

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Clock	-clock [get_clocks {clock_name}]	Indicates the clock the delay applies to, which is obtained through get_clocks; when using the clock, the principle of "declare before use" must be followed, meaning the clock must be created with create_generated_clock or create_clock first
Clock Edge	-clock_fall	Indicates that the delay applies to the falling edge of the clock
Delay Transition	-rise/-fall	Indicates that the delay is valid at the rising/falling edge of the input signal
Max/Min Delay Path	-max/-min	-max indicates that the user-defined delay value is the maximum value, and the software only needs to perform setup analysis -min indicates that the user-defined delay value is the minimum value, and the software only needs to perform hold analysis
Add Delay	-add_delay	If the user has already set a set_input_delay constraint on the input port, adding this option means the software will preserve the previously set constraint values. For -max, the delay value will be calculated using the maximum of all values under the same conditions; for -min, the calculation will use the minimum of all values under the same conditions.
Latencies Included	-source_latency _included	Indicates that [Set Clock Latency] has been taken into account
Delay Value	{delay_value}	Delay value, float type
Objects	[get_ports /get_pins]	Indicates the pin or port the delay applies to, obtained through get_pins or get_ports

4.2.8 set_max_skew

set_max_skew is used to constrain the maximum skew between a set of user-specified paths. The constrained path's start and end points must satisfy the requirements of Start Points and End Points. When set_false_path or set_clock_groups is set, set_max_skew will be ignored; however, after set_max_delay, set_min_delay, or set_multicycle_path is set, set_max_skew remains effective and does not conflict with their priorities.

Double-click [Set Max Skew] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_max_skew" command in the [Set Max Skew] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

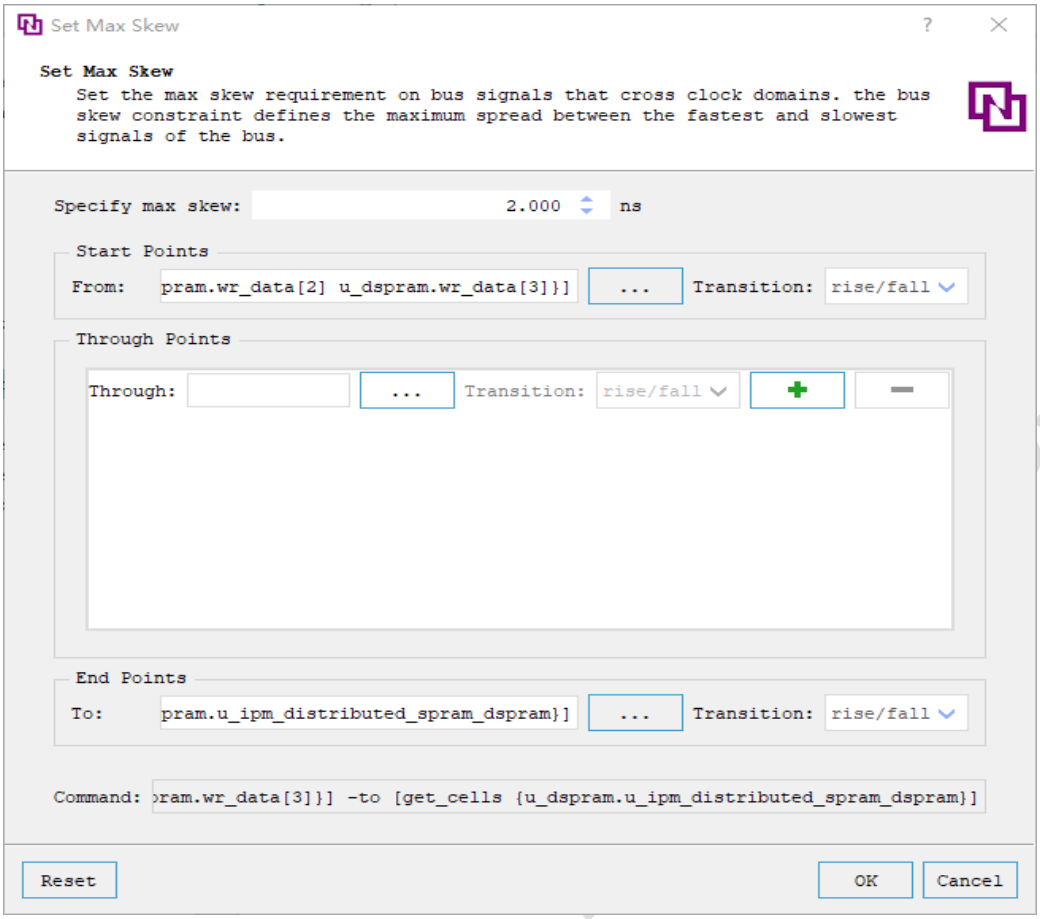


Figure4-34

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

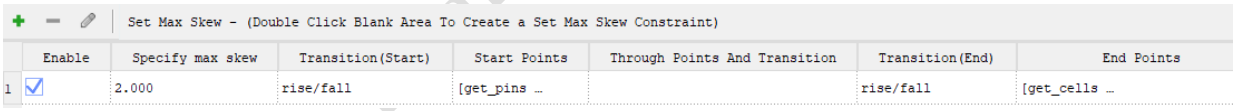


Figure4-35

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_max_skew {2.000} -from [get_pins {u_dspram.wr_data[0] u_dspram.wr_data[1] u_dspram.wr_data[2] u_dspram.wr_data[3]}] -to [get_cells {u_dspram.u_ipm_distributed_spram_dspram}]

The correspondence between the list information and the command line is as follows.

Table 4-8

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Specify max skew	{ skew_value}	Set the skew value, measured in ns
Transiton(Strat)	-rise_from/	-rise_from: This option can only specify clocks, indicating that the start

List Attribute	Command Parameter	Description
	-fall_from	point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point; -fall_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
Start Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the startpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells
Through Points And Transition	[get_nets/get_pins/ get_insts]	Indicates the intermediate points that the path must pass through, obtained through get_nets/get_pins/get_insts
Transiton(End)	-rise_to/ -fall_to	-rise_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point. -fall_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
End Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the endpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells

4.2.9 set_max_delay

set_max_delay is used to specify the maximum delay for a certain path, where the start and end points of the path must satisfy the requirements of Start Points and End Points. Generally, the maximum delay that satisfies the setup check is determined by the delay of the clock path, the clock period, and the constraint value of the setup check. However, if the user specifies set_max_delay, whether the requirements can be met is no longer related to the clock path, clock period, and the constraint value of the setup check; it is sufficient as long as the delay from the Start Points to the End Points is less than the maximum value constrained by the user.

Double-click [Set Max Delay] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_max_delay" command in the [Set Max Delay] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

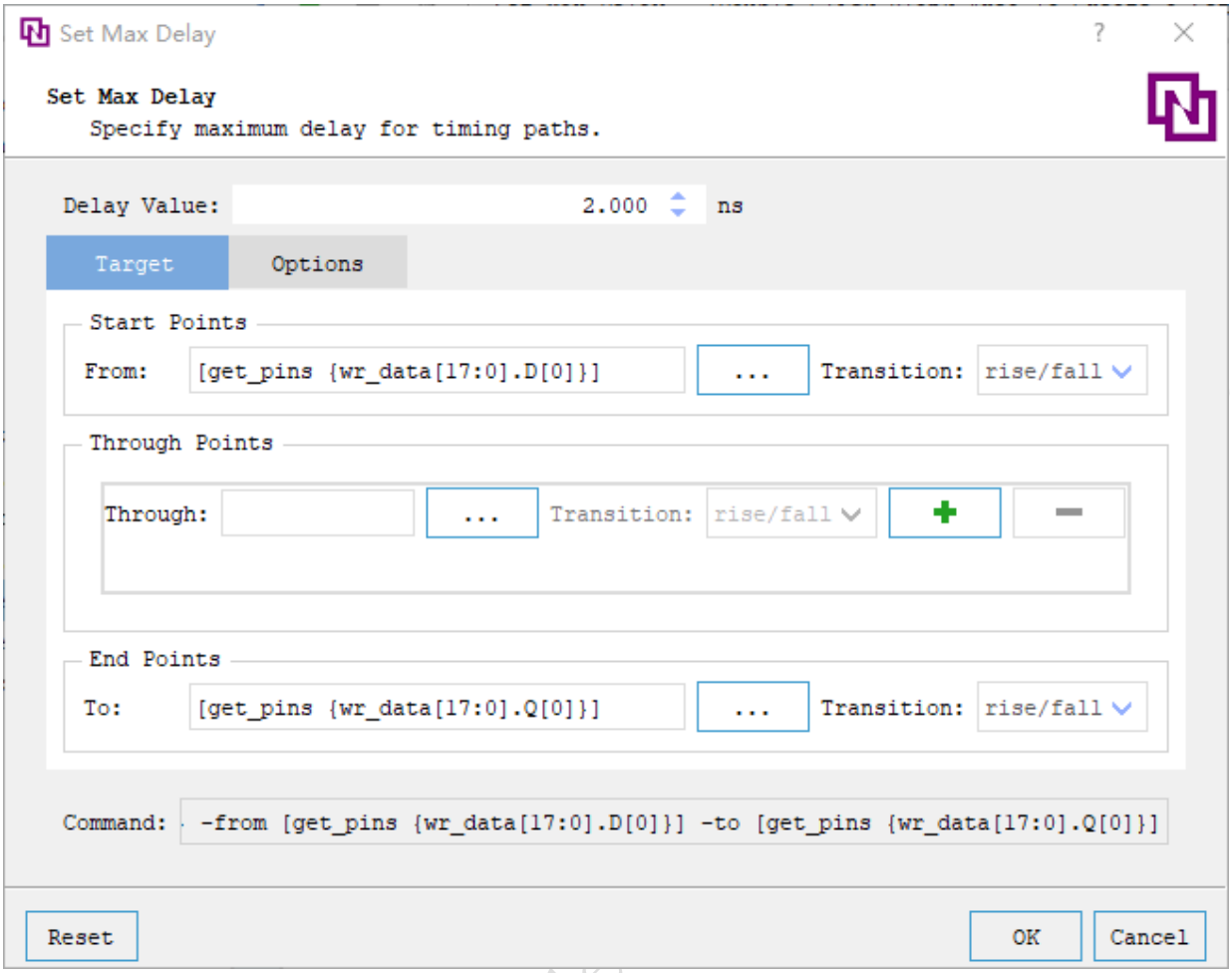


Figure4-36

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

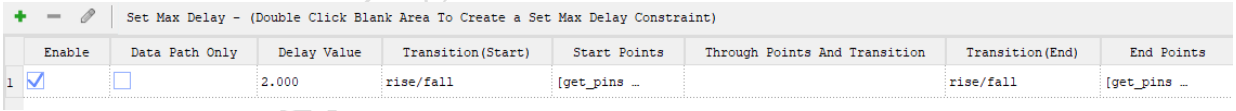


Figure4-37

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_max_delay {2.000} -from [get_pins {wr_data[17:0].D[0]}] -to [get_pins {wr_data[17:0].Q[0]}]

The correspondence between the list information and the command line is as follows.

Table 4-9

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Data Path Only	-datapath_only	Indicates that the specified max delay path does not concern clock skew and jitter, considering only the data path
Delay Value	{delay_value}	Delay value, float type, measured in ns
Transiton(Strat)	-rise_from/	-rise_from: This option can only specify clocks, indicating that the start

List Attribute	Command Parameter	Description
	-fall_from	point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point. -fall_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
Start Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the startpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells
Through Points And Transition	[get_nets/get_pins/ get_insts]	Indicates the intermediate points that the path must pass through, obtained through get_nets/get_pins/get_insts
Transiton(End)	-rise_to/ -fall_to	-rise_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point. -fall_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
End Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the endpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells

4.2.10 set_min_delay

set_min_delay is used to specify the minimum delay for a certain path, where the start and end points of the path must satisfy the requirements of Start Points and End Points. Generally, the minimum delay that satisfies the hold check is determined by the delay of the clock path and the constraints of the hold check. However, if the user specifies set_min_delay, whether the requirements can be met is no longer related to the clock path, clock period, and the constraint value of the setup check; just ensure the delay from the Start Points to the End Points is greater than the minimum value constrained by the user.

Double-click [Set Min Delay] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the set_min_delay command in the [Set Min Delay] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

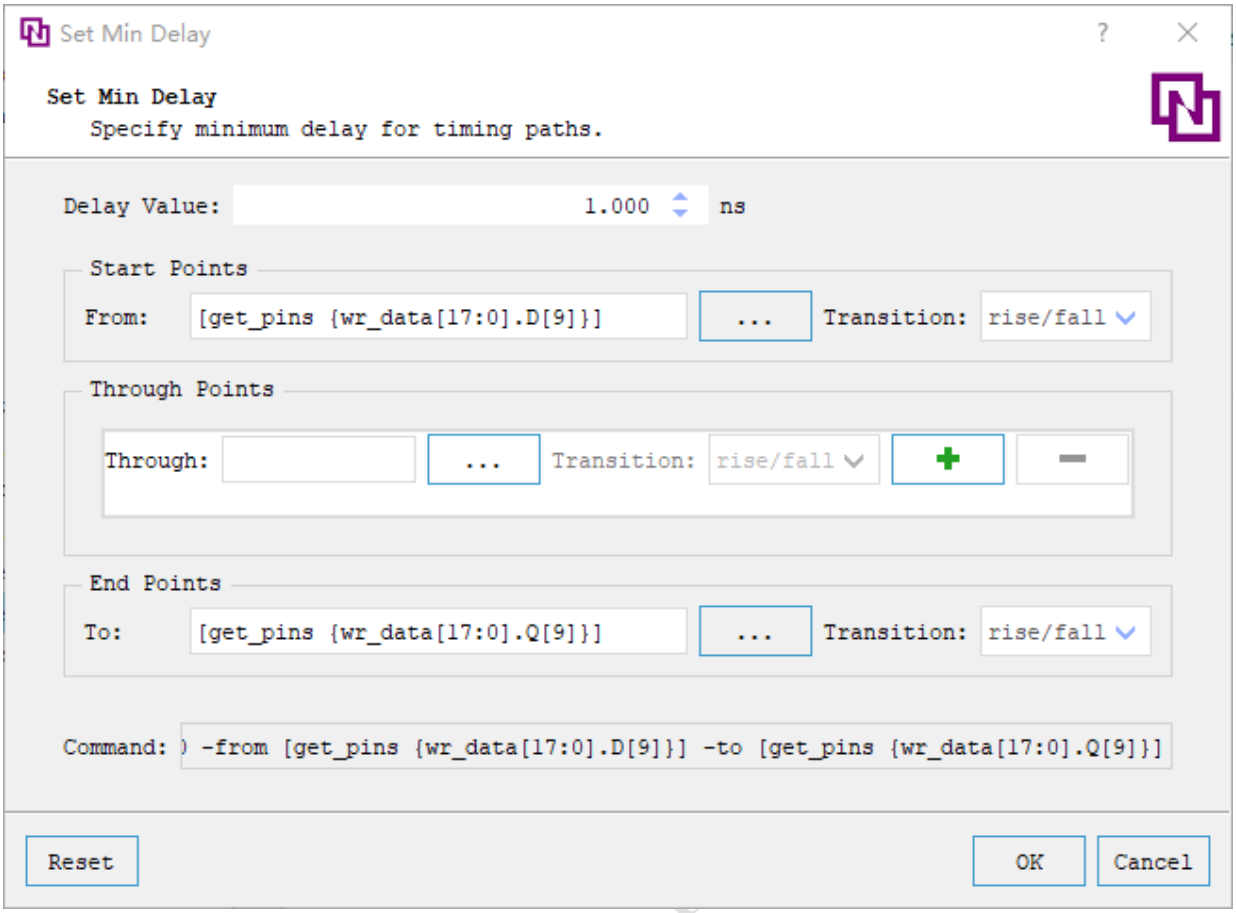


Figure4-38

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

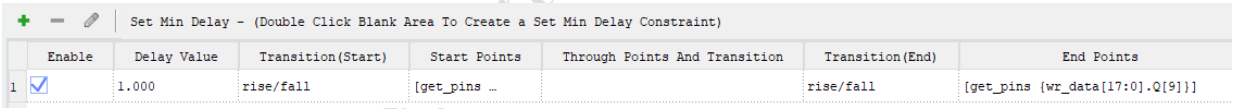


Figure4-39

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_min_delay {1.000} -from [get_pins {wr_data[17:0].D[9]}] -to [get_pins {wr_data[17:0].Q[9]}]

The correspondence between the list information and the command line is as follows.

Table 4-10

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Data Path Only	-datapath_only	Indicates that the specified max delay path does not concern clock skew and jitter, considering only the data path
Delay Value	{delay_value}	Delay value, float type, measured in ns
Transiton(Strat)	-rise_from/ -fall_from	-rise_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point.

List Attribute	Command Parameter	Description
		-fall_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
Start Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the startpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells
Through Points And Transition	[get_nets/get_pins/ get_insts]	Indicates the intermediate points that the path must pass through, obtained through get_nets/get_pins/get_insts
Transiton(End)	-rise_to/ -fall_to	-rise_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point. -fall_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
End Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the endpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells

4.2.11 set_multicycle_path

set_multicycle_path is used to set up multi-cycle paths. Generally, the signal transmission delay between registers under the same clock drive is a single cycle, but sometimes it may exceed one cycle. In such cases, the timing analysis tool must be informed that the transmission along this path requires multiple cycles.

Double-click [Set Multicycle Path] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_multicycle_path" command in the [Set Multicycle Path] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

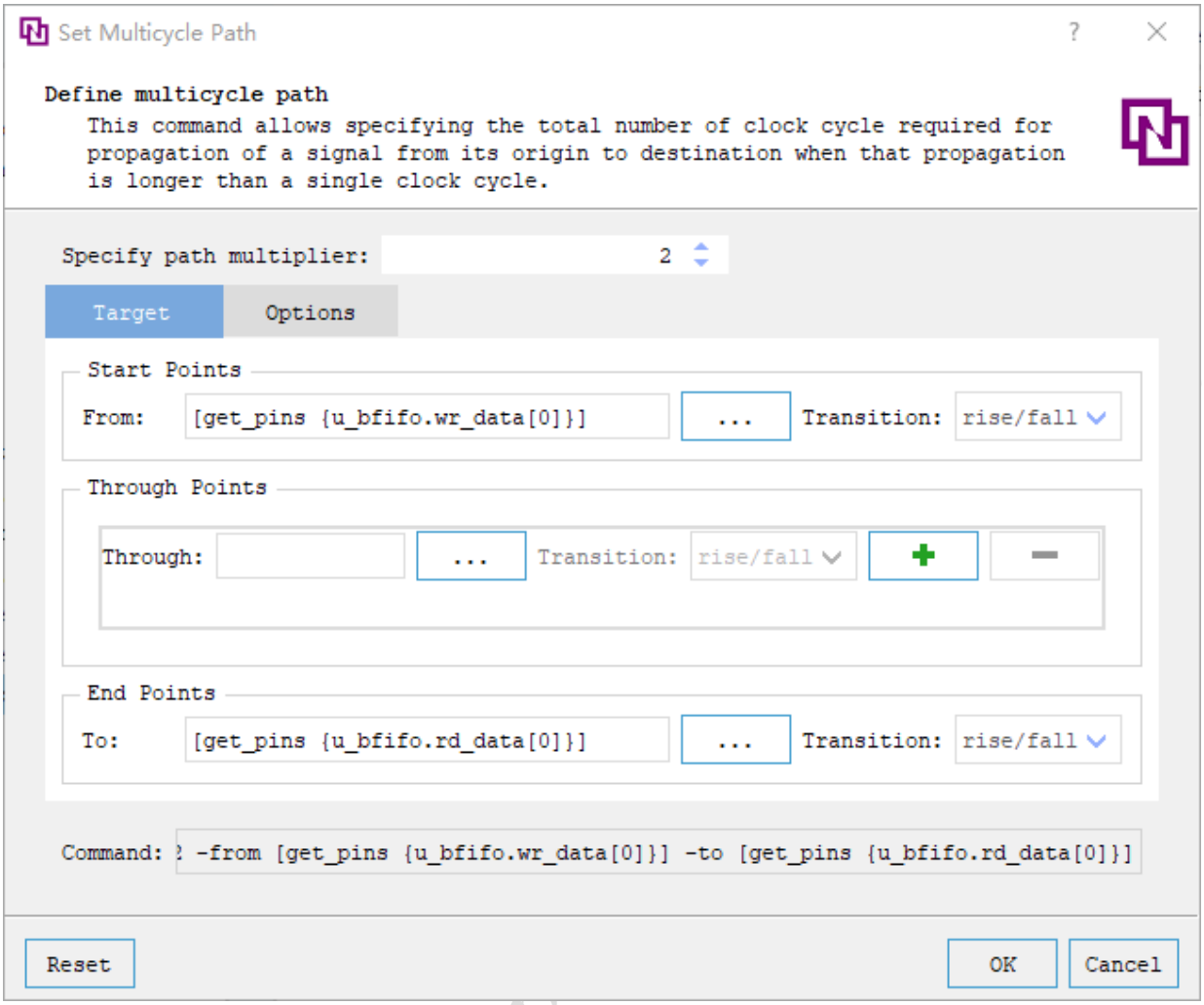


Figure4-40

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

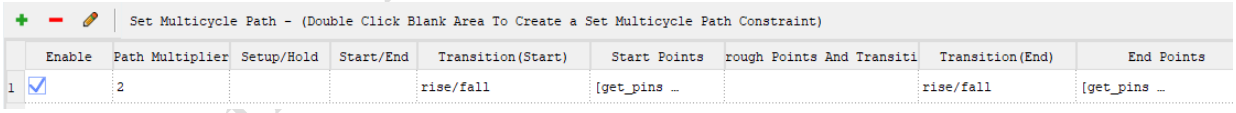


Figure4-41

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_multicycle_path {2} -from [get_pins {u_bfifo.wr_data [0]}] -to [get_pins {u_bfifo.rd_data[0]}]

The correspondence between the list information and the command line is as follows.

Table 4-11

List Attribute	Command Parameter	Description
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Path Multiplier	{path_value}	Indicates the number of cycles set, which is an integer greater than or equal to 0. When set to 0, it indicates that the specified path is a zero cycle path

List Attribute	Command Parameter	Description
Setup/Hold	-setup/-hold	-setup indicates that during setup analysis, the clock periods use a specified multiple. After setup is specified, hold is also affected. By default, setup is used; -hold indicates that during hold analysis, the clock periods use a specified multiple
Start/End	-start/-end	-start indicates that the setting is related to the launch clock. If the number of cycles set is N, and then for setup, the launch clock needs to use the edge of the N-1 cycles before the default analysis edge (the edge for hold analysis also needs to move); for hold, it uses the edge of the N cycles after the launch clock's default analysis edge; -end indicates that the setting is related to the capture clock. If the number of cycles set is N, and then for setup, the capture clock needs to use the edge of the N-1 cycles after the default analysis edge (the edge for hold analysis also needs to move); for hold, it uses the edge of the N cycles before the capture clock's default analysis edge
Transiton(Strat)	-rise_from/ -fall_from	-rise_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point; -fall_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
Start Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the startpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells
Through Points And Transition	[get_nets/get_pins/ get_insts]	Indicates the intermediate points that the path must pass through, obtained through get_nets/get_pins/get_insts
Transiton(End)	-rise_to/ -fall_to	-rise_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point; -fall_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
End Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the endpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells

4.2.12 set_false_path

set_false_path is used to eliminate timing analysis for a certain path, which is generally non-functional or timing is not a concern, such as registers written only once during power-up initialization, reset logic, test logic, signals in asynchronous clock domains, or asynchronous read/write operations of asynchronous distributed RAM.

Double-click [Set False Path] on the left, click "+", or right-click on an empty space and select [Insert Row] in the menu to pop up the command creation window.

Create the "set_false_path" command in the [Set False Path] window, with the top part of the window providing an introduction to the command. The middle part is for creating the command, where the [Command] field can display the form and parameters of the created command. Clicking the [Reset] button at the bottom will reset the interface. The window interface is as follows:

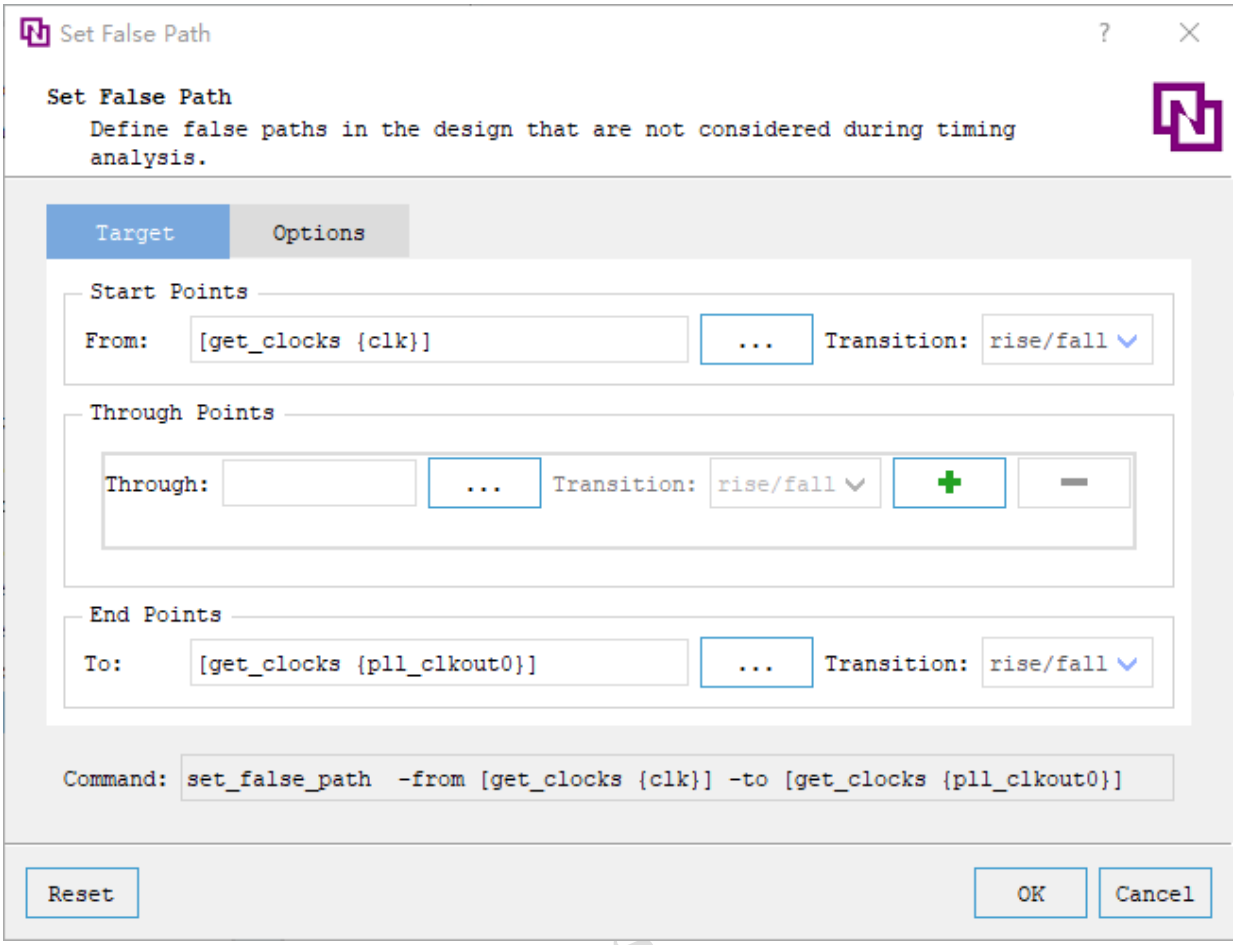


Figure4-42

Click [OK], and the newly created clock constraint information is saved in the list on the right, as shown in the figure below.

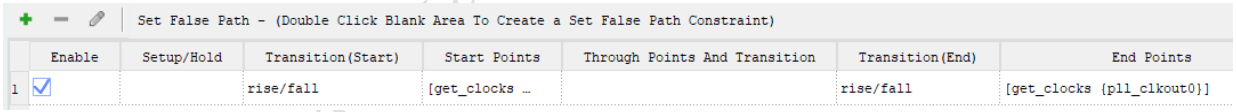


Figure4-43

At the same time, the corresponding command is recorded in the .fdc file, as shown in the example below.

set_false_path -setup -from [get_clocks {clk}] -to [get_clocks {pll_clkout0}]

The correspondence between the list information and the command line is as follows.

Table 4-12

List Attribute	**Command Parameter**	**Description**
Enable	-disable	Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.
Setup/Hold	-setup/-hold	-setup indicates setup analysis; -hold indicates hold analysis
Transiton(Strat)	-rise_from/ -fall_from	-rise_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point;

List Attribute	Command Parameter	Description
		-fall_from: This option can only specify clocks, indicating that the start point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
Start Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the startpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells
Through Points And Transition	[get_nets/get_pins/ get_insts]	Indicates the intermediate points that the path must pass through, obtained through get_nets/get_pins/get_insts
Transiton(End)	-rise_to/ -fall_to	-rise_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the rising edge of the clock, where the rising edge refers to the rising edge at the clock definition point; -fall_to: This option can only specify clocks, indicating that the end point of the timing path is triggered by the falling edge of the clock, where the falling edge refers to the falling edge at the clock definition point
End Points	[get_ports/get_pins/ get_insts/get_cells]	Indicates the endpoint of the path, obtained through get_ports/get_pins/get_insts/get_cells

4.2.13 Summary

Click [Summary] to view the timing constraint information, as shown in the figure below.

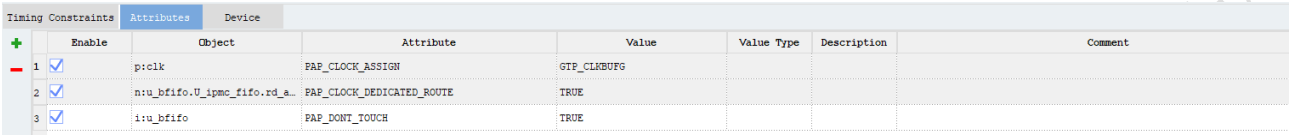
```
###==== BEGIN "create_clock"
create_clock -name {clk} [get_ports {clk}] -period {10.000} -waveform {0.000 5.000}
###==== END "create_clock"

###==== BEGIN "create_generated_clock"
create_generated_clock -name {pll_clkout0} -source [get_pins {u_pll.clkin1}] [get_pins {u_pll.clkout0}] -master_clock [get_clocks {clk}] -multiply_by {1} -duty_cycle {50.000}
###==== END "create_generated_clock"

###==== BEGIN "set_clock_latency"
set_clock_latency {0.000} -rise -fall -min -source -early -clock [get_clocks {pll_clkout0}] [get_pins {u_dspram.wr_data[0]}]
###==== END "set_clock_latency"

###==== BEGIN "set_clock_uncertainty"
set_clock_uncertainty {1.000} -from [get_clocks {clk}] -to [get_clocks {pll_clkout0}] -setup -hold
###==== END "set_clock_uncertainty"

###==== BEGIN "set_clock_groups"
set_clock_groups -name clock_group -asynchronous -group [get_clocks {clk}]
###==== END "set_clock_groups"

###==== BEGIN "set_input_delay"
set_input_delay {2.000} [get_ports {rd_en}] -rise -fall -min -max -add_delay -clock [get_clocks {clk}]
###==== END "set_input_delay"

###==== BEGIN "set_output_delay"
set_output_delay {0.000} [get_ports {fifo_rd_data[11]}] -rise -fall -min -max -add_delay -clock [get_clocks {pll_clkout0}]
###==== END "set_output_delay"

###==== BEGIN "set_max_skew"
set_max_skew {2.000} -from [get_pins {u_dspram.wr_data[0] u_dspram.wr_data[1] u_dspram.wr_data[2] u_dspram.wr_data[3]}] -to [get_cells {u_dspram.u_ipm_distributed_spram_dspram}]
###==== END "set_max_skew"

###==== BEGIN "set_max_delay"
set_max_delay {2.000} -from [get_pins {wr_data[17:0].D[0]}] -to [get_pins {wr_data[17:0].Q[0]}]
###==== END "set_max_delay"

###==== BEGIN "set_min_delay"
set_min_delay {1.000} -from [get_pins {wr_data[17:0].D[9]}] -to [get_pins {wr_data[17:0].Q[9]}]
###==== END "set_min_delay"

###==== BEGIN "set_multicycle_path"
set_multicycle_path {2} -from [get_pins {u_bfifo.wr_data[0]}] -to [get_pins {u_bfifo.rd_data[0]}]
###==== END "set_multicycle_path"

###==== BEGIN "set_false_path"
set_false_path  -setup -from [get_clocks {clk}] -to [get_clocks {pll_clkout0}]
###==== END "set_false_path"

<<< "NEW UNSAVED COMMAND">>>
set_clock_groups -name clock_group0 -asynchronous -group [get_clocks {pll_clkout0}]
<<< "NEW UNSAVED COMMAND">>>
```

Figure4-44

The red "NEW UNSAVED COMMAND" indicates the constraint information that has not yet been saved. After clicking [Save], this constraint information is transferred to the corresponding command region, and the "NEW UNSAVED COMMAND" region is cleared.

4.3 Attributes

In the UCE interface, select [Attributes] to set logical constraints. Currently, logical constraints are represented in the constraint file in the form of define_attribute. When using synthesis tools, attributes are saved in the constraint file (.fdc); without using synthesis tools (backend version), attributes are saved in the constraint file (.lcf). The following describes the attributes of each column in the [Attributes] table.



	Enable	Object	Attribute	Value	Value Type	Description	Comment
1	☑	p:clk	PAP_CLOCK_ASSIGN	GTP_CLKBUFG			
2	☑	n:u_bfifo.U_ipmc_fifo.rd_a…	PAP_CLOCK_DEDICATED_ROUTE	TRUE			
3	☑	i:u_bfifo	PAP_DONT_TOUCH	TRUE			

Figure4-45

[Enable]: Checked by default. If not checked, "-disable" will be added to the end of the constraint when saving the constraint file, indicating the constraint is not effective.

[Object]: Indicates the constraint object, which can be a port, pin, instance, or net; a prefix is added to the object in the table to indicate its type, where the prefix for instance type is "i:", for net type is "n:", for port type is "p:", and for pin type is "t:"; different types of Object support different Attributes;

[Attribute]: Indicates the constrained attribute type;

[Value]: Describes the value of the attribute. The format and requirements of the value itself are related to the specific attribute, and illegal values will result in errors or warnings during constraint checks;

[Value Type/Description/Comment]: Further describes the characteristics related to the attribute value.

The attributes supported by the [Attributes] page are shown in the table below.

Table 4-13

Attribute	Object	Value	Functional Description
PAP_GROUP	instance	Any string that conforms to naming standards	After mapping, the instances in the design are combined into a group, which is displayed separately as a group in the PCE's design browser, for example define_attribute {u_top_module.sub_module} {PAP_GROUP} {group_01}
PAP_CLOCK_DEDICATED_ROUTE	pin,net, or without specifying an object (global attribute, affects all relevant objects, with lower priority than pin and net)	TRUE FALSE	1. If the clock is not constrained to a dedicated clock pin and is connected to the clock network, the PDS process will report an error (E: Place-0084) when running to Place&Route. If the net related to the clock or the pin driven by that net is set to FALSE, the PDS process will downgrade the error to a Critical Warning (C: Place-2028); 2. Used to control the trace of the clock. If this attribute is not set or is set to TRUE, the clock is not allowed to run through the interconnect channels between SRBs; if the clock is desired to run through the interconnect channels between SRBs, it needs to be set to FALSE. For example define _attribute {n:net}

Attribute	Object	Value	Functional Description
			{PAP_CLOCK_DEDICATED_ROUTE} {FALSE}
PAP_CLOCK_ASSIGN	net, pin, or port	GTP_CLKBUFG GTP_BUFGS GENERAL	During Place&Route, the value of this attribute will determine whether to use a global clock, global signal, or ordinary SRB. For example define_attribute {n:net} {PAP_CLOCK_ASSIGN} {GTP_CLKBUFR}
PAP_IO_REGISTER	instance (register)	TRUE FALSE	During mapping, the value of this attribute will determine whether to pack the register with IO together. This attribute does not support -disable and -comment. For example define_attribute {i:register} {PAP_IO_REGISTER} {TRUE}
PAP_PLL_FEEDBACK_DELAY	instance (PLL)	Double type number, measured in ns	Configure the delay of the PLL's feedback end, for example define_attribute {i:pll} {PAP_PLL_FEEDBACK_DELAY} {120ns}
PAP_DONT_TOUCH	instance (module or GTP type instance)	TRUE FALSE	During the Device Map phase, ensure that the instance is not optimized or changed, for example define_attribute {i:GTP_LUT6} {PAP_DONT_TOUCH} {TRUE}
PAP_KEEP_CHAIN	instance(Carry Chain)	TRUE FALSE	Control the Carry Chain to prevent it from being broken, for example define_attribute {i:T_GTP_LUT2} {PAP_KEEP_CHAIN} {TRUE}

Chapter 5 Feature Control Bits

Compa family products have many multi-function pins, which users can configure as either dedicated IO or general IO by setting feature control bits, and can be set differently before and after the chip enters user mode.

Right-click within the Navigator region, select [Project Setting] from the menu, which brings up the configuration interface. Choose [Generate Bitstream > Feature Control], as shown in the image below (Versions PDS2021.2 and later support the new feature control bit configuration interface).

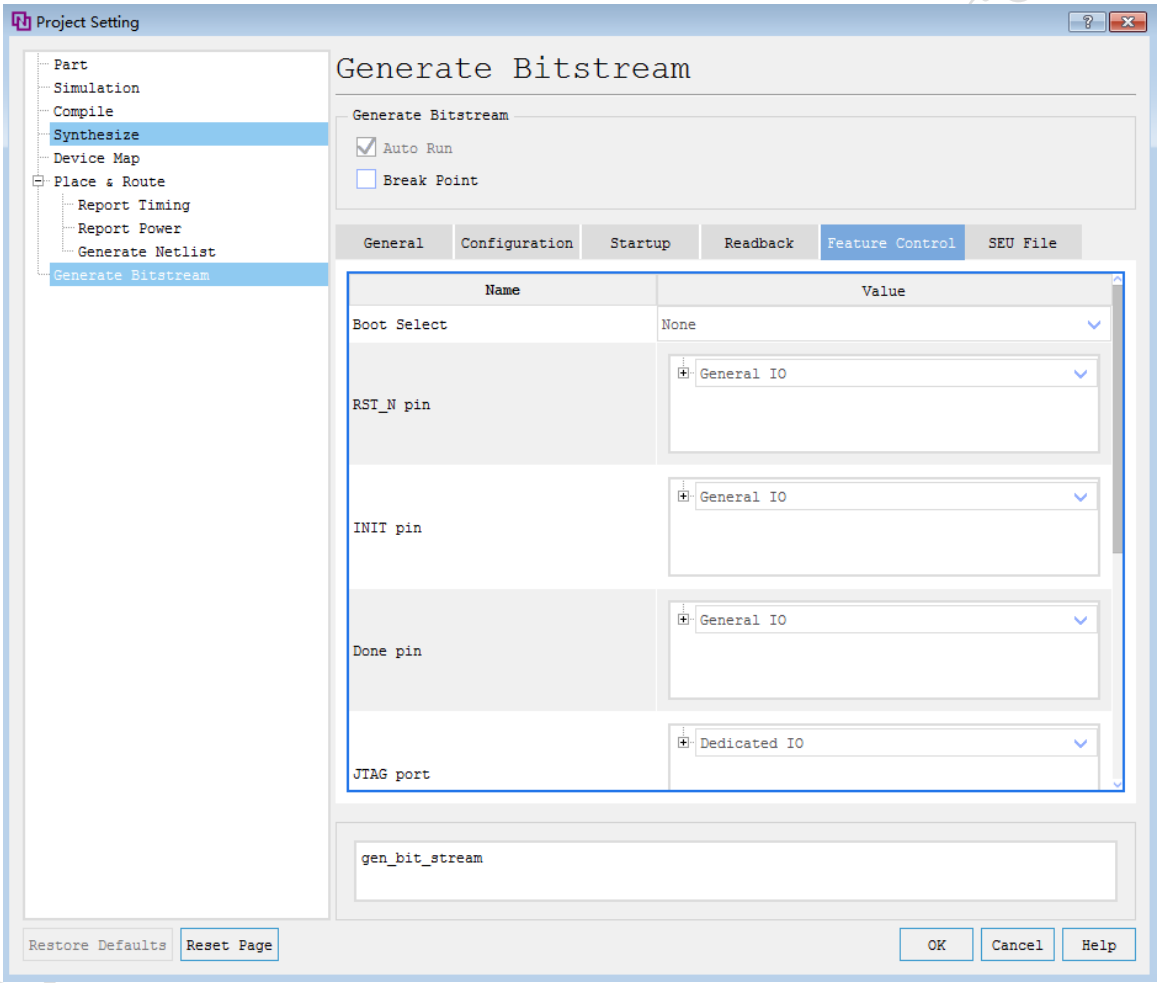


Figure5-1

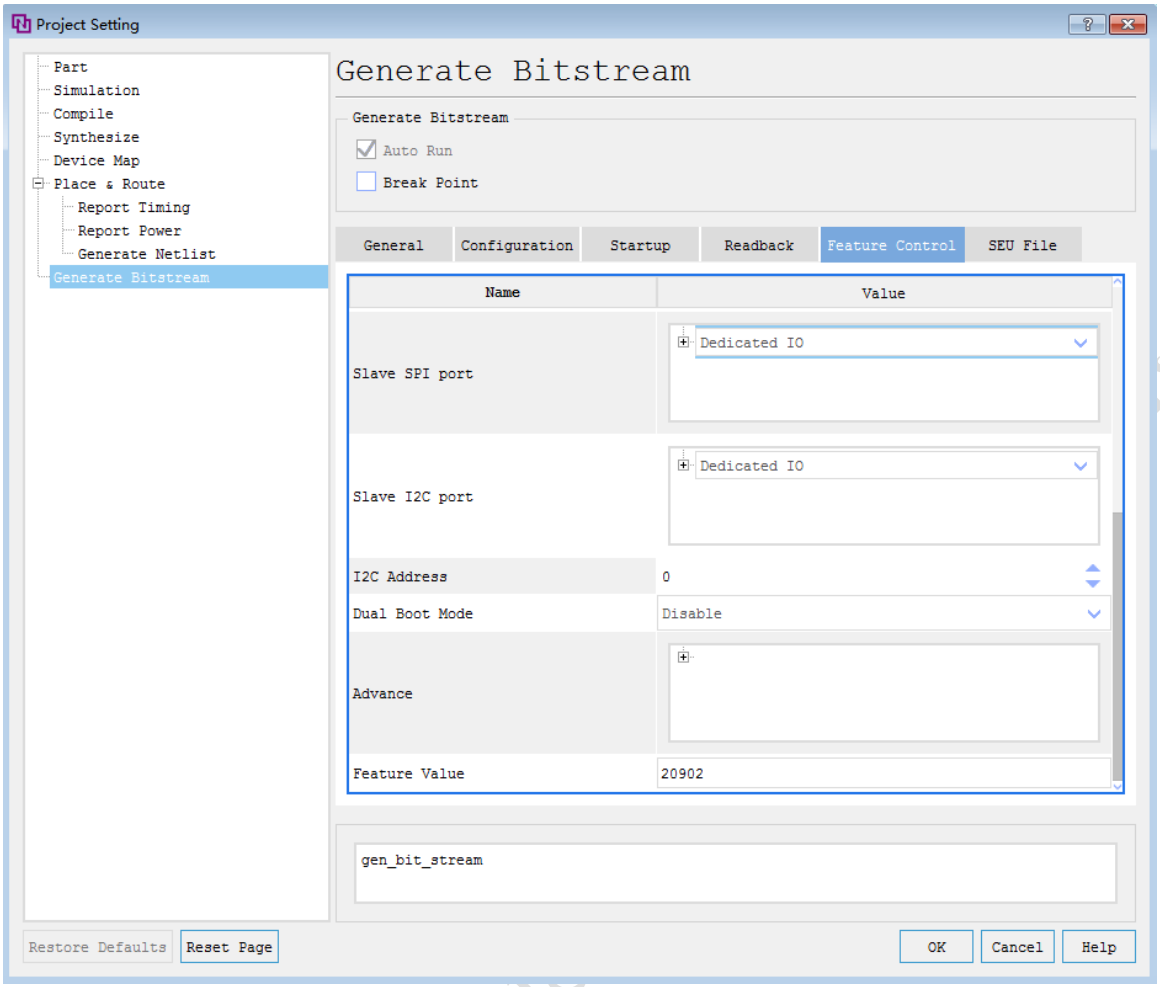


Figure5-2

5.1 Boot Select

For PGC devices, the default boot mode selection is NONE, meaning CRAM will not be loaded actively.

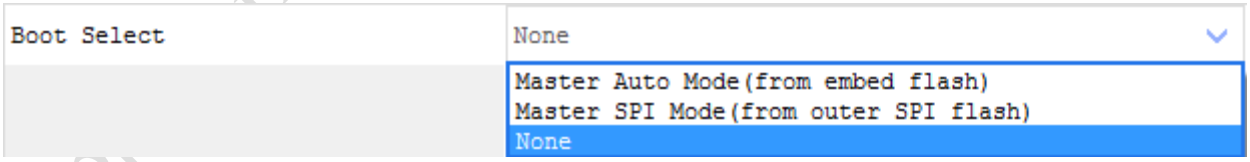


Figure5-3

[Master Auto Mode]: Master Self Configuration. After the chip is powered up and initialization is complete, it actively obtains the bitstream from embedded Flash to download the CRAM;

[Master SPI Mode]: Master SPI configuration. After the chip is powered up and initialization is complete, it actively obtains the bitstream from external SPI Flash to download the CRAM;

[None]: After the chip is powered up and initialization is complete, there are no further actions.

5.2 RST_N pin

RST_N (RSTN) serves as a reset pin when used as a dedicated IO; by default, it is a general IO.

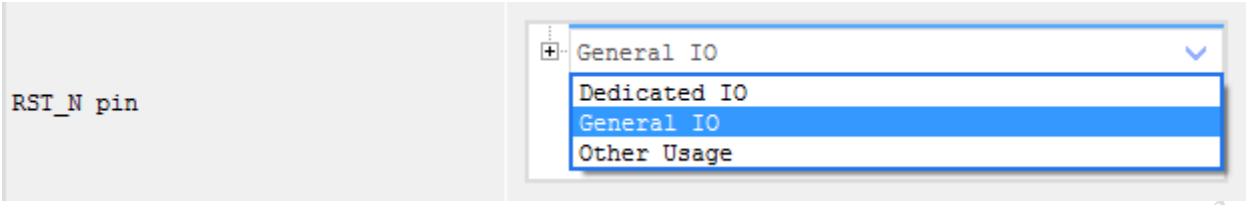


Figure5-4

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, used to reset the chip;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a general IO.

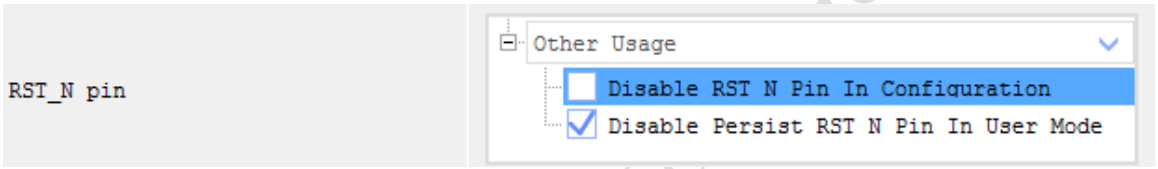


Figure5-5

5.3 INIT pin

INIT (INIT_FLAG_N) indicates the chip's initialization status when used as a dedicated IO; by default, it is a general IO.

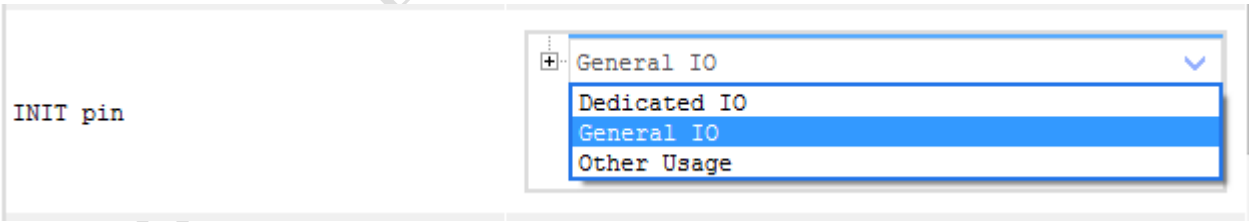


Figure5-6

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, indicating the chip's initialization status;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a dedicated IO.

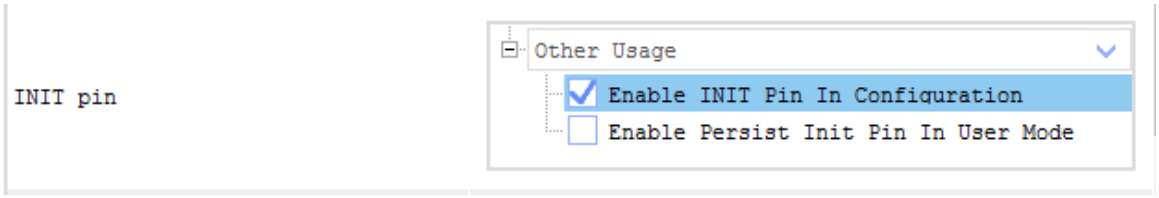


Figure5-7

5.4 DONE pin

DONE (CFG_DONE) indicates the chip's configuration status when used as a dedicated IO; by default, it is a general IO.

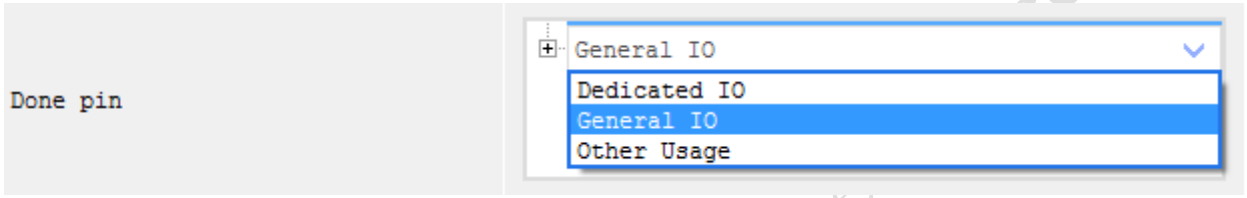


Figure5-8

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, indicating the chip's configuration status;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a dedicated IO.

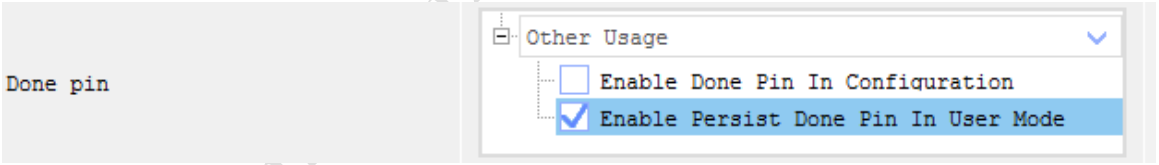


Figure5-9

5.5 JTAG port

JTAG port (TDI, TDO, TCK, TMS) serves as the JTAG configuration interface when used as a dedicated IO; by default, it is a dedicated IO.

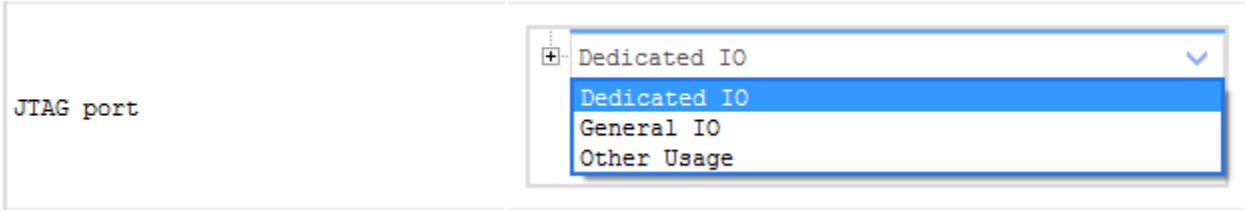


Figure5-10

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, used as the

configuration interface;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a general IO.

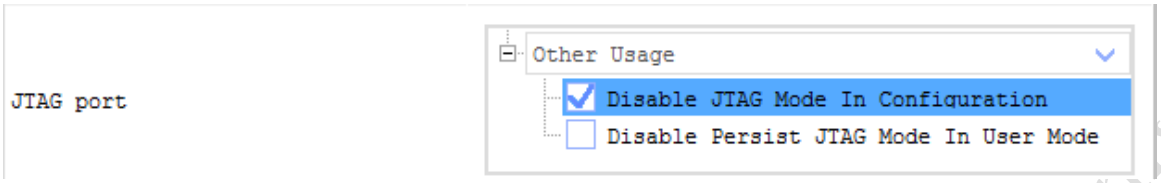


Figure5-11

Note that JTAGEN is also a multi-function pin. When the JTAG port is used as a dedicated IO, JTAGEN serves as a general IO; when the JTAG port is used as a general IO, JTAGEN serves as a dedicated IO to control whether the JTAG port can still be used as a dedicated IO, with JTAGEN set to 1 indicating that the JTAG port is a dedicated IO.

5.6 Slave SPI port

Slave SPI port (CFG_CLK, MISO_SO, MOSI_SI, FCSI_N) serves as the slave SPI configuration interface when used as a dedicated IO; by default, it is a dedicated IO.

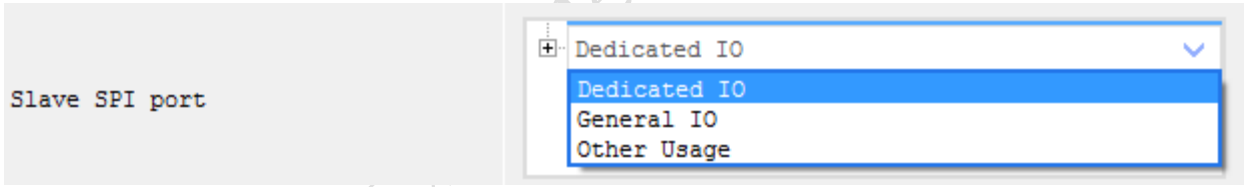


Figure5-12

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, used as the configuration interface;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a general IO.

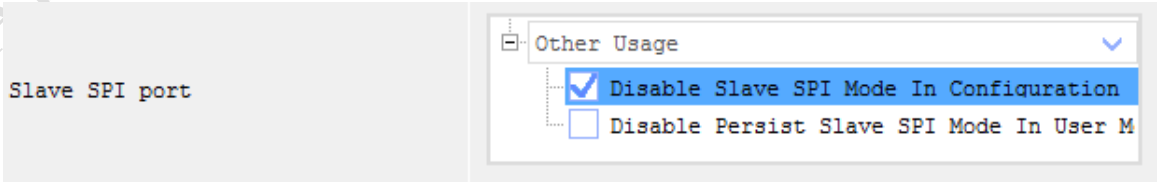


Figure5-13

5.7 Slave I2C port

Slave I2C port (SCL, SDA) serves as the slave I2C configuration interface when used as a dedicated IO; by default, it is a dedicated IO.

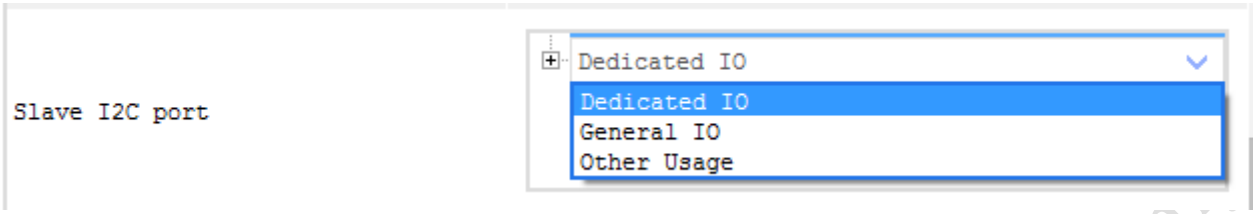


Figure5-14

[Dedicated IO]: Serves as a dedicated IO during configuration and in user mode, used as the configuration interface;

[General IO]: Serves as a general IO during configuration and in user mode;

[Other Usage]: Click "+" to mix settings during configuration and in user mode. A checkmark indicates it serves as a general IO.

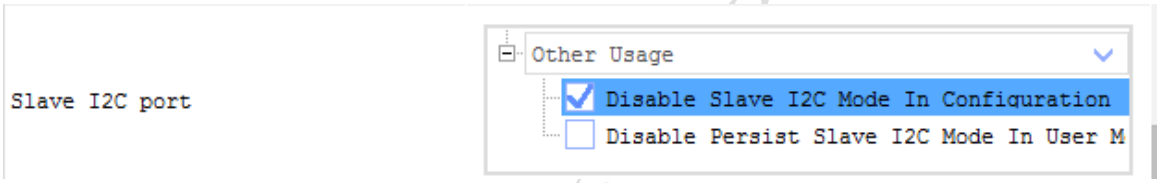


Figure5-15

5.8 I2C Address

I2C address for PGC devices when serving as a slave device. The address is a 10-bit binary number, displayed in hexadecimal, with a default value of 0.

Figure5-16

5.9 Dual Boot Mode

Dual boot mode. The default is Disable, and dual boot is not enabled.

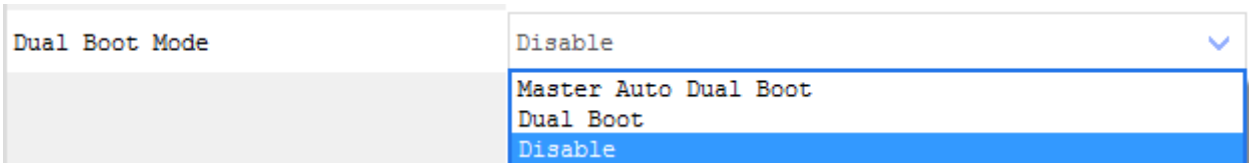


Figure5-17

[Master Auto Dual Boot]: Enables the Master Self Configuration Dual Boot, which stores the

golden bitstream and application bitstream in the embedded Flash. It automatically switches to the golden bitstream if the application bitstream fails to start; Supported by PGC4K/7K/10K.

[Dual Boot]: Enables regular dual boot, where a set of bitstreams is stored in both the embedded Flash and external SPI Flash; Supported by all PGC devices.

[Disable]: Disables the dual boot mode.

5.10 Other Entries

[Advance]: It is recommended to use the default settings.

[Feature Value]: Displays the set feature control bits, a 32-bit hexadecimal number.

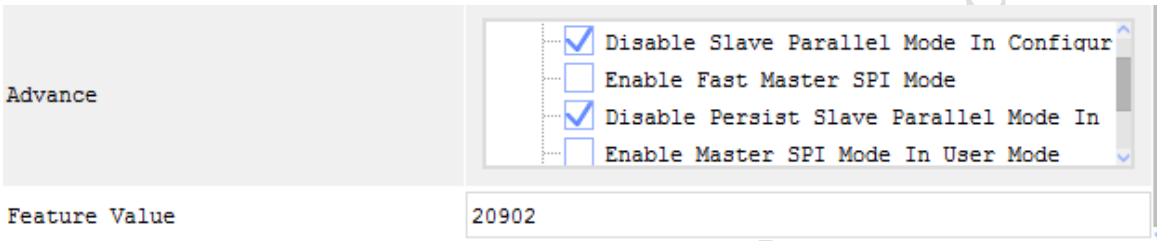


Figure5-18

Chapter 6 Generate Bitstream

After entering the design source files and constraint files, the next step is to generate the bitstream files to implement the design. Generating the bitstream files involves six essential steps]: [Compile], [Synthesize], [Device Map], [Place & Route], [Report Timing], and [Generate Bitstream]. Additionally, there are optional processes]: [Report Power] and [Generate Netlist]. This is shown in the following figure.

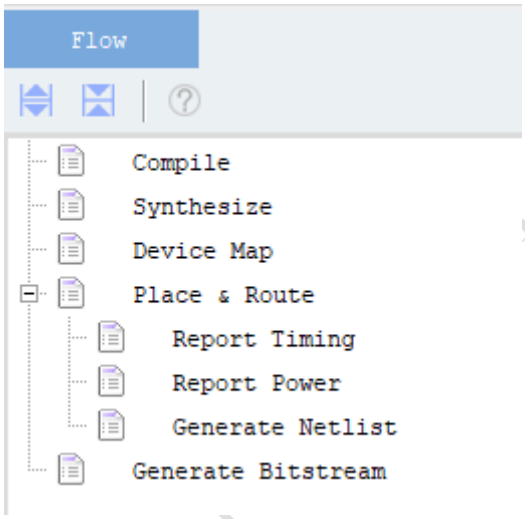


Figure6-1

Typically, double-clicking [Generate Bitstream] will sequentially execute from [Compile] to generate the bitstream files, or users can click to run the complete process.

For each process, project settings can be made to fine-tune and optimize the process. Right-click in the [Flow] region and select [Project Setting] from the menu to set parameters for each process.

6.1 Compile

The [Compile] process is only available when "ADS" is selected as the synthesis tool during project creation. During the [Compile] stage, the design source code is converted into RTL netlist files.

6.1.1 [Verilog] Page

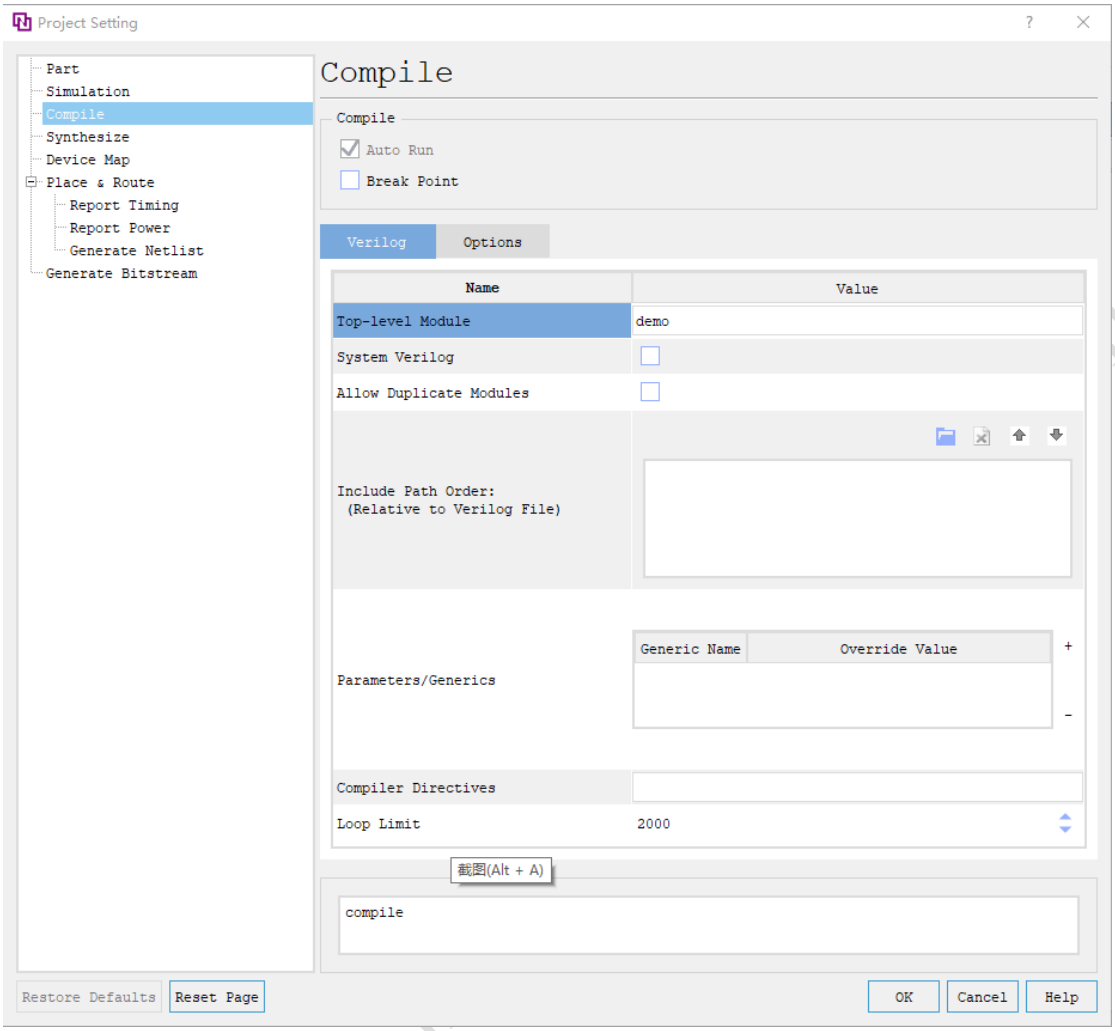


Figure6-2

[Top-level Module]: Sets the top module, with the value being the name of the module. If this is not set, PDS will automatically select a module that has not been instantiated as the top module. If there are multiple uninstantiated modules in the input file, ADS will select the first uninstantiated module in alphabetical order as the top module.

[System Verilog]: Not checked by default. When checked, Verilog is set to System Verilog by default. This option must be checked for successful compilation when System Verilog syntax is present in the design.

[Allow Duplicate Modules]: Not checked by default. If this option is not checked, an error will occur when there are modules with the same name in the design. When this option is checked, modules with the same name can be compiled successfully, and the software will override the previous module with the subsequent one according to the module parsing order.

Note: In scenarios where an IP is instantiated multiple times, if this option is not checked, the software will not check for duplicate module names and will directly override the previous module

with the subsequent one according to the module parsing order.

[Include Path Order]: Sets the project's include path. Click the button  , a [Select Directory] prompt box will appear, and select the target folder. This option will only list the file directory as an include target, not including the files in the directory. For the operations of actual include files to take effect, they must be used with the "include" statement in the RTL files. Include files can take various forms: include absolute path, such as adding `include "/user/dirA/fileA.v" directly in the RTL file; include relative path, such as `include "../fileA.v". At this point, the relative path is based on the directory where the RTL file containing the include statement is located, and it is necessary to ensure the correct relative path between the RTL file containing the statement and fileA.v.

Note: The include file is only a macro definition file, not a Verilog file, hence the include file does not need to be added to the project.

[Parameters/Generics]: The default is empty. Parameter values set in the options will override the top-level parameter values in the design. Generic Name indicates the top-level parameter name, and Override Value indicates the new parameter value. Click "+" to add a new one. Use spaces to separate parameter names and parameter values. String type parameter values need to be quoted "". For example, ADDR_WIDTH 10; ASYNC "TRUE";

[Compiler Directives]: The default is empty. Used to set macro definition parameters. Use "=" between macro names and values, and use spaces to separate multiple macro definition instructions, such as ADDR_WIDTH=8 DATA_WIDTH=4 ASYNC="TRUE";

[Loop Limit]: Sets the upper limit of the number of for statement loops. The default value is 2000;

6.1.2 [Options] Page

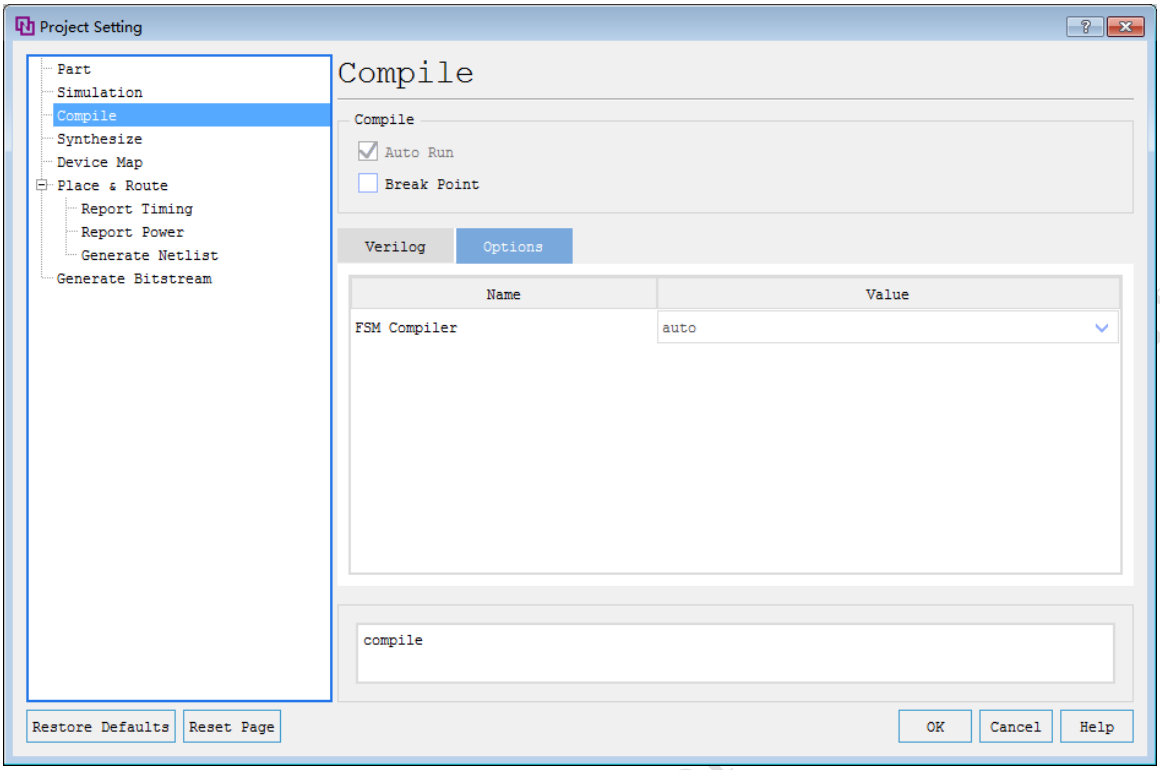


Figure6-3

[FSM Compiler]: A switch designed to select the FSM encoding format. The default is auto. Other optional values are shown in the table below.

Table 6-1

FSM Encoding Method	**Description**
auto	Choose the best encoding method based on software optimization strategies
off	Do not perform FSM infer
one_hot	One-hot encoding, represented by N registers for N states, with only one bit as 1 for each state. It can reduce combinational logic and optimize timing with fastest speed, but uses more registers
gray	Gray code encoding, only one state flips during adjacent state transitions. It uses fewer registers compared to one_hot
sequential	Sequential encoding. It uses the fewest registers under the same conditions
original	Maintains the original encoding in the source files
safe	Prevents the state machine from being locked in an illegal state. Once the FSM enters any illegal state, it can be returned to a legal state immediately through safe logic, enhancing the stability of the design
safe, one_hot	Uses one-hot encoding method, and prevent the state machine from being locked in an illegal state
safe, gray	Uses Gray code encoding method, and prevent the state machine from being locked in an illegal state
safe, sequential	Uses sequential encoding method, and prevent the state machine from being locked in an illegal state
safe, original	Maintains the original encoding in the source files, and prevent the state machine from being locked in an illegal state

6.2 Synthesize

When ADS is selected as the synthesis tool, [Synthesize] generates Technology netlist files (.vm) by performing mapping and optimization based on the DB generated by the [Compile] process. The generated .vm netlist files can be verified using third-party verification tools. The netlist information is transferred between [Compile] and [Synthesize] via DB files, and the outcome of the logic synthesis is conveyed to [Device Map] by creating DB files that encapsulate the synthesized netlist data.

The generated .vm netlist file can be used for functional simulation, with the pre-simulation library directory at "..\arch\vendor\pango\verilog\simulation".

6.2.1 [Option] Page

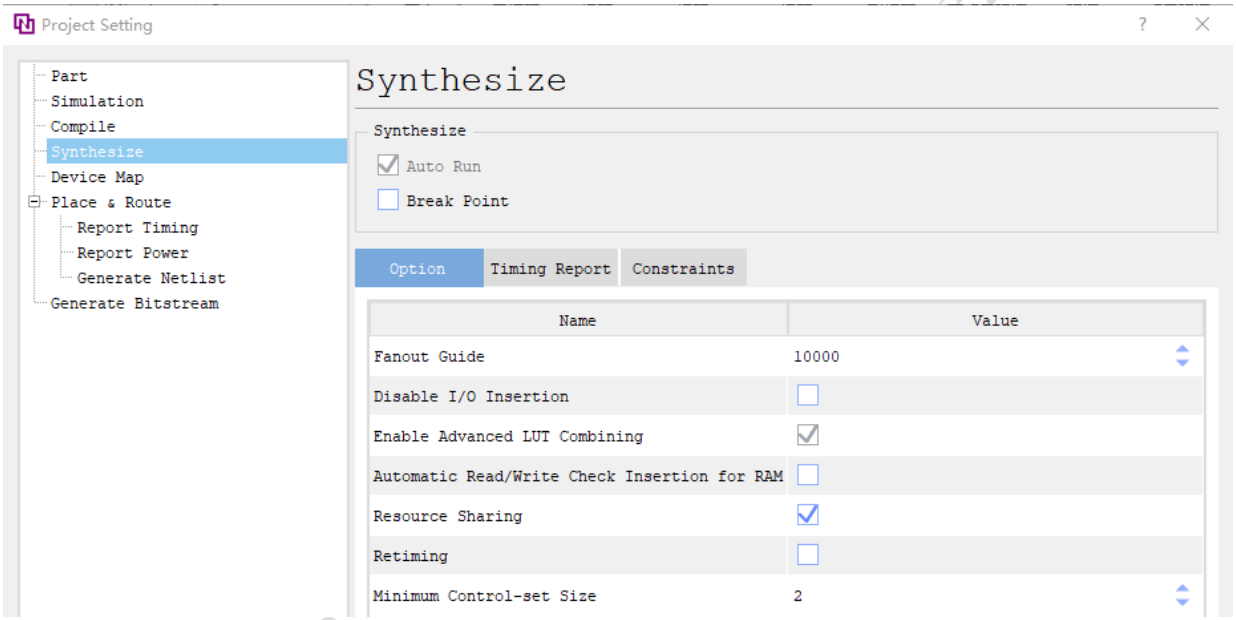


Figure6-4

[Fanout Guide]: Sets the maximum global fanout value. The default value is 10000;

[Disable I/O Insertion]: I/O insert switch. By default, it is not checked, and I/O buffers are inserted into the netlist; if this option is selected, I/O buffers will not be inserted into the netlist;

[Enable Advanced LUT Combining]: A switch that combines two compatible LUTs into a dual-output GTP_LUT6D. Only valid for family with GTP_LUT6D function, such as Logos2/Titan2 family. If set to unselected, the LUT will not be packed. The default is on. For family other than Logos2/Titan2, it is greyed out by default;

[Automatic Read/Write Check Insertion for RAM]: Currently, only simple dual-port RAM using transparent_write mode and selecting this option will insert bypass logic, ensuring that the data read out is the same as the data written in the same clock cycle, avoiding simulation inconsistencies;

[Resource Sharing]: The default is on for arithmetic operations to share resources, mainly for

sharing resources in addition (subtraction) and multiplication operations. Resource merging optimizes area but may reduce timing. If set to unselected, resources will not be shared;

[Retiming]: The default is off; when checked, retiming optimizes the circuit's timing, power, or area by moving existing registers in the circuit. During this process, no extra logic will be inserted, but the number of registers may increase or decrease when moving registers;

[Minimum Control-set Size]: Adjusts the minimum number of registers with the same control port, i.e., the minimum number of registers consistent with CLK, CE, synchronous SET/RESET, etc. The default value is 2, and the optional values are positive integers. When the number of registers with the same control port is greater than or equal to the set value, optimization is not performed; when it is less than the minimum value, some or all of the control pins of the corresponding register are moved to the D input.

6.2.2 [Timing Report] Page

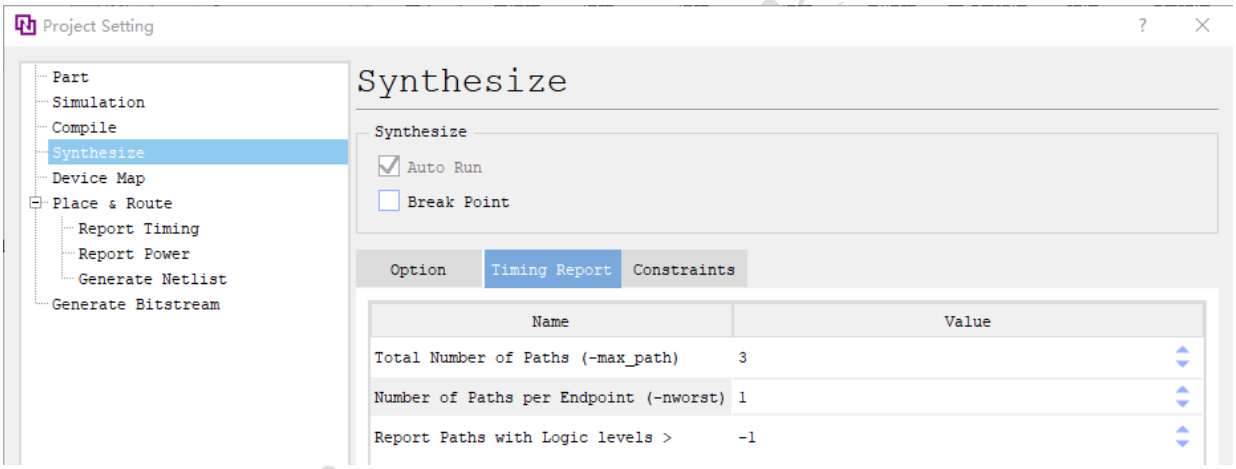


Figure6-5

[Total Number of Paths (-max_path)]: Indicates the number of the worst timing paths reported in the synthesis timing report;

[Number of Paths per Endpoint (-nworst)]: Indicates the maximum number of paths for each endpoint that can be reported during synthesis timing analysis;

[Report Paths with Logic Levels >]: Indicates that the Logic Levels of each setup/recovery timing path reported during synthesis timing must be greater than the set value; paths with Logic Levels less than this value will not be displayed. This setting does not affect the hold/removal analysis timing path;

6.2.3 [Constraints] Page

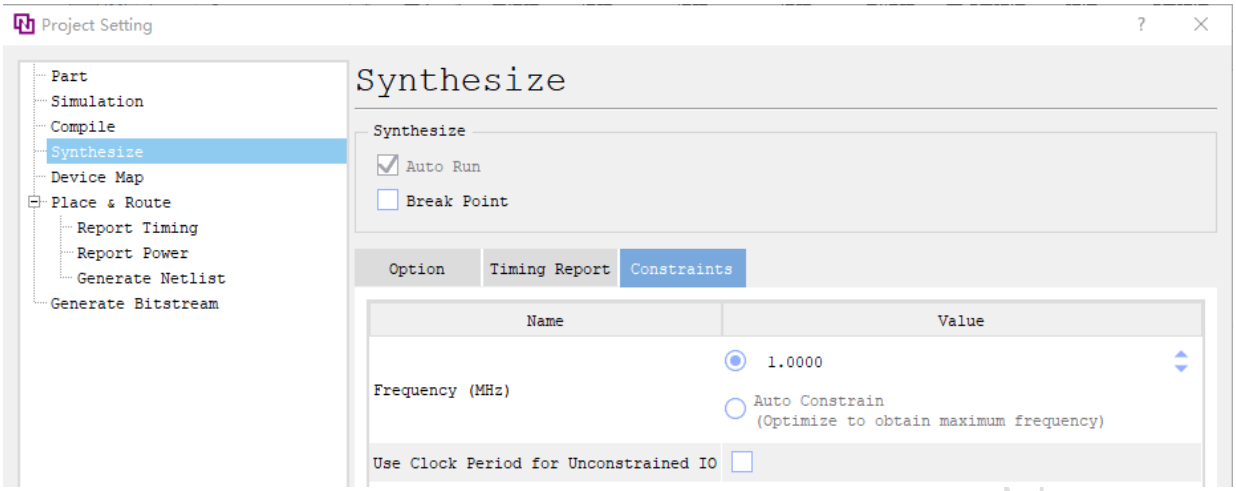


Figure6-6

[Frequency (MHz)]: Used to infer clock. If the correct clock frequency cannot be calculated based on the PLL parameters and reference clock, the setting value of this option will be used to create the clock as the clock frequency. The default is 1.0000. Auto constraint is currently not supported. Auto Constrain is greyed out and cannot be checked;

[Use Clock Period for Unconstrained IO]: This option is to add IO delay constraints for pins without constrained IO delay, to improve the coverage of timing constraints. The default is off.

6.2.4 Synthesis Report

Double-click [Synthesize] in the [Flow] to execute the synthesis process, ultimately generating a synthesis report, which includes resource usage and timing report after synthesis, as shown in the figure below.

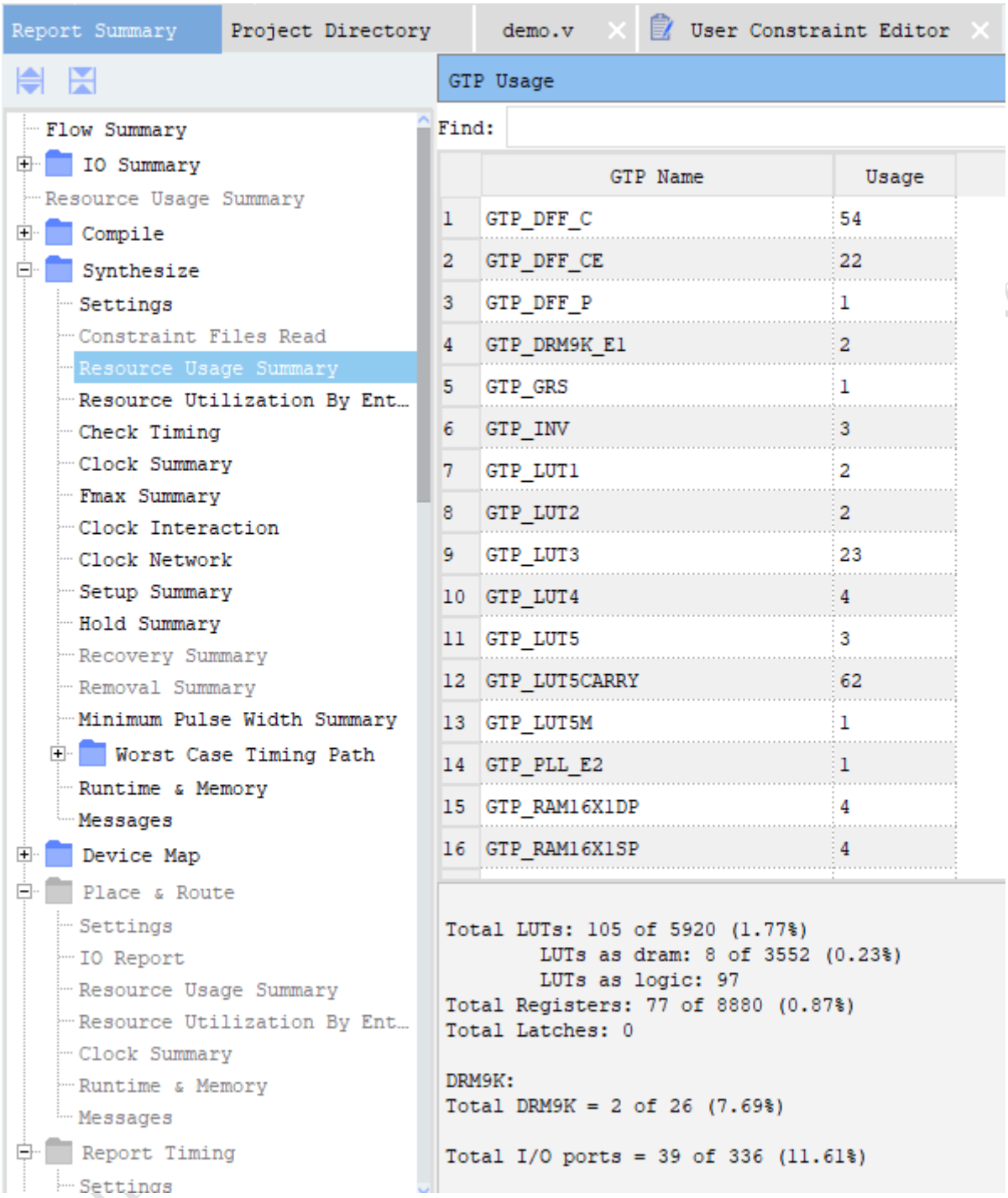


Figure6-7

6.3 Device Map

[Device Map] stage maps the design to specific hardware units, such as LUTs, FFs, and DRMs.

6.3.1 [Mapping] Page

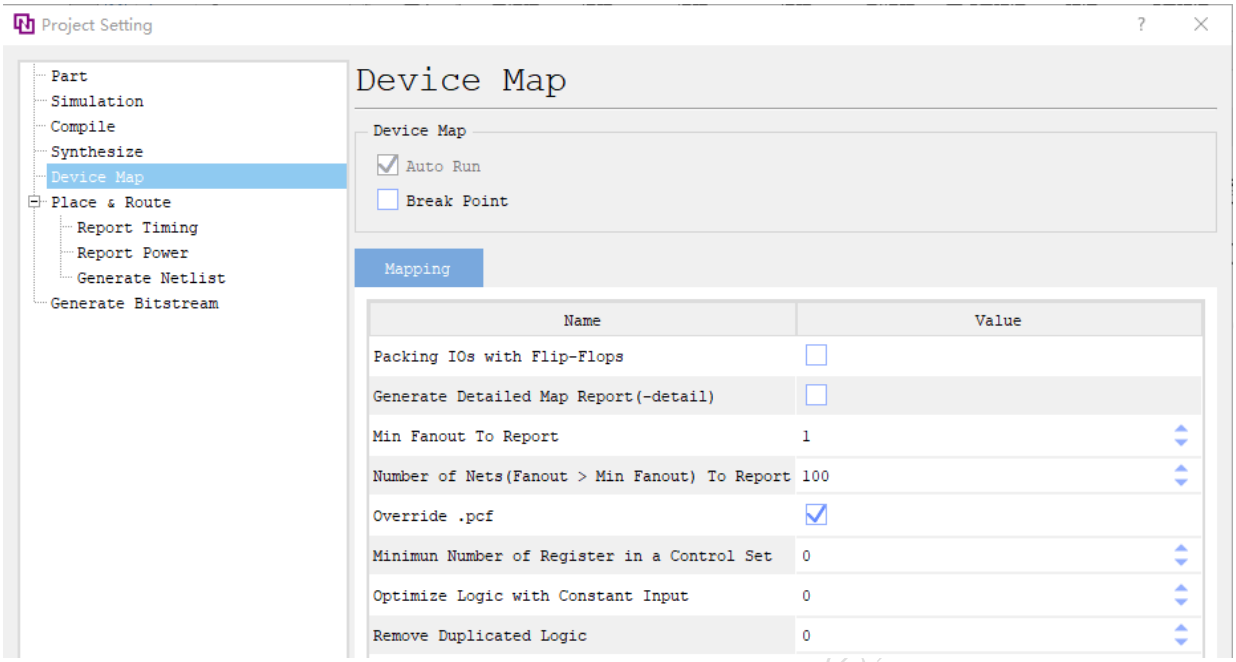


Figure6-8

[Packing IOs with Flip-Flops]: Indicates whether registers directly connected to IO ports are packed into the corresponding IOLs. The input/output mode of this IO port include input, output, or inout. The default is off, meaning the registers are not packed with the IO;

[Generate Detailed Map Report(-detail)]: Determines whether to print the resource utilization for each module instance in the map's report file; The defaut is off, so the resource utilization of each module is not printed;

[Min Fanout To Report]: Sets the minimum number of net fanouts to be printed in the map report;

[Number of Nets(Fanout > Min Fanout) To Report]: Sets the number of nets to be printed in the map report, where the fanout of the printed nets must be greater than the minimum fanout setting value;

[Override .pcf]: Sets whether to override the existing .pcf when the .pcf constraint information generated from the user constraint files (fdc/lcf) is inconsistent with the existing .pcf file constraint information. The defaut is on, and the .pcf will be automatically overwritten when device map is executed;

[Enable timing commands commented by Synplify Pro]: Sets the enable state, which allows timing commands marked as comments by the OEM to take effect in the PDS process; (not supported by ADS synthesis)

[Minimum Number of Register in a Control Set]: Used to control the minimum number of registers under the same Control Set. When the number of registers is below the threshold, the gating signal is implemented as a data path using LUTs. Control Set refers to the clock, enable signal, and RST

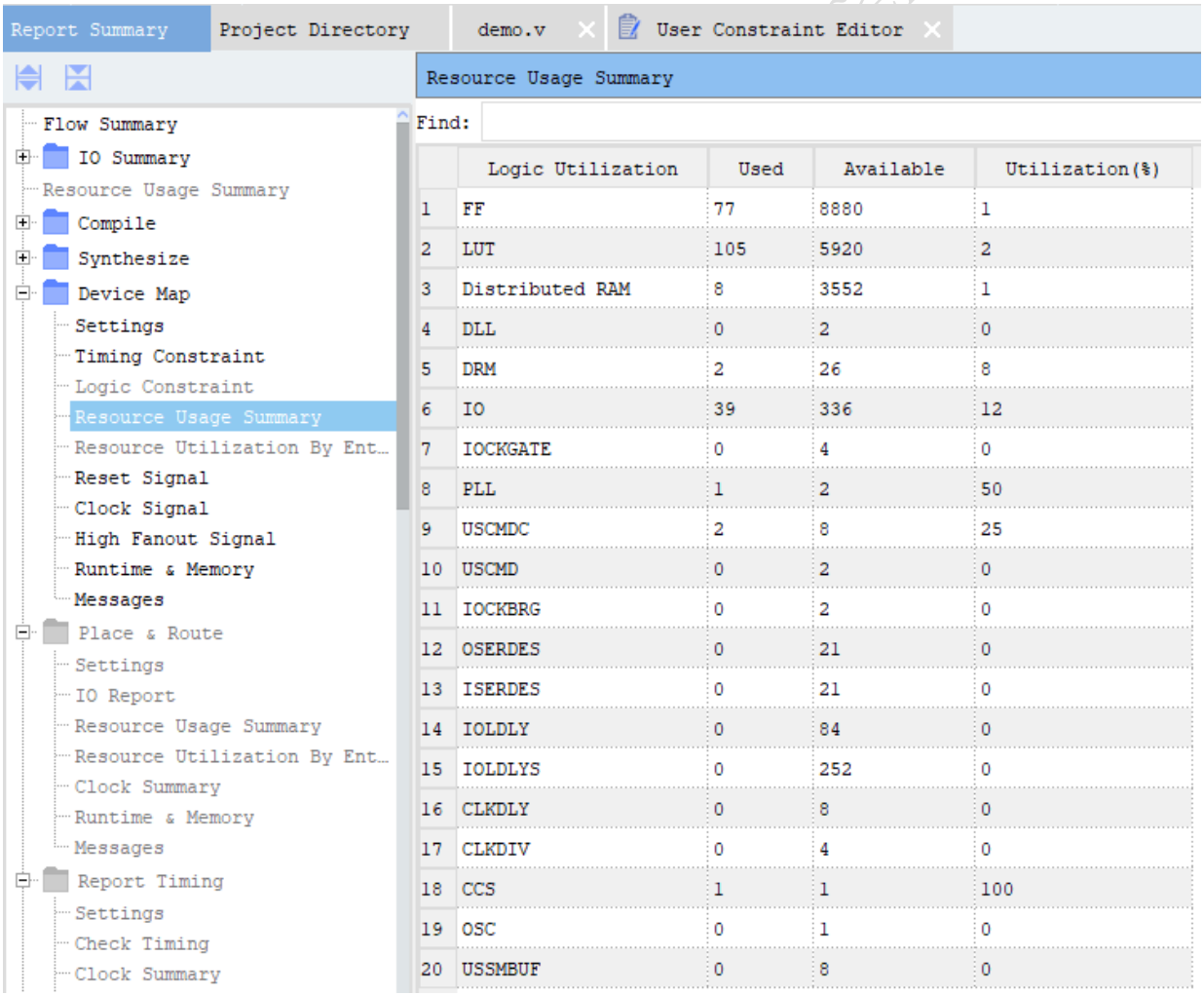
signal of registers. Optimizing this control signal can increase resource utilization;

[Optimize Logic with Constant Input]: Used to optimize the constant logic input. Constant logic mainly refers to VCC and GND logic. This optimization is mainly achieved by leveraging the inherent characteristics of the circuit to disconnect or optimize the original VCC and GND to the best state;

[Remove Duplicated Logic]: Merges partial duplicate logic, which is an option for area optimization. However, this operation will also affect the replication during the synthesis stage, resulting in decreased timing performance.

6.3.2 Map Report

Double-click on [Device Map] in [Flow] to generate a map report, as shown in the figure below.



	Logic Utilization	Used	Available	Utilization(%)
1	FF	77	8880	1
2	LUT	105	5920	2
3	Distributed RAM	8	3552	1
4	DLL	0	2	0
5	DRM	2	26	8
6	IO	39	336	12
7	IOCKGATE	0	4	0
8	PLL	1	2	50
9	USCMDC	2	8	25
10	USCMD	0	2	0
11	IOCKBRG	0	2	0
12	OSERDES	0	21	0
13	ISERDES	0	21	0
14	IOLDLY	0	84	0
15	IOLDLYS	0	252	0
16	CLKDLY	0	8	0
17	CLKDIV	0	4	0
18	CCS	1	1	100
19	OSC	0	1	0
20	USSMBUF	0	8	0

Figure6-9

6.3.3 PCE Tool

PCE stands for Physical Constraint Editor, which can be used to set location constraints and region constraints for instances as well as set I/O attributes for I/Os. Its function is similar to UCE, which is used for constraints during synthesis, whereas PCE is used for constraints after Device Map. The following shows how to open the PCE tool:

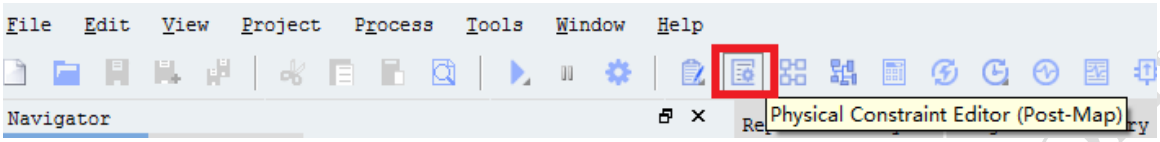


Figure6-10

For detailed information and usage of the PCE tool, please see the "Physical_Constraint_Editor_User_Guide" at ..\pango\PDS_.....\doc\.

6.4 Place&Route

[Place&Route] Performs actual placement and routing of design modules based on physical constraints.

6.4.1 [General] Page

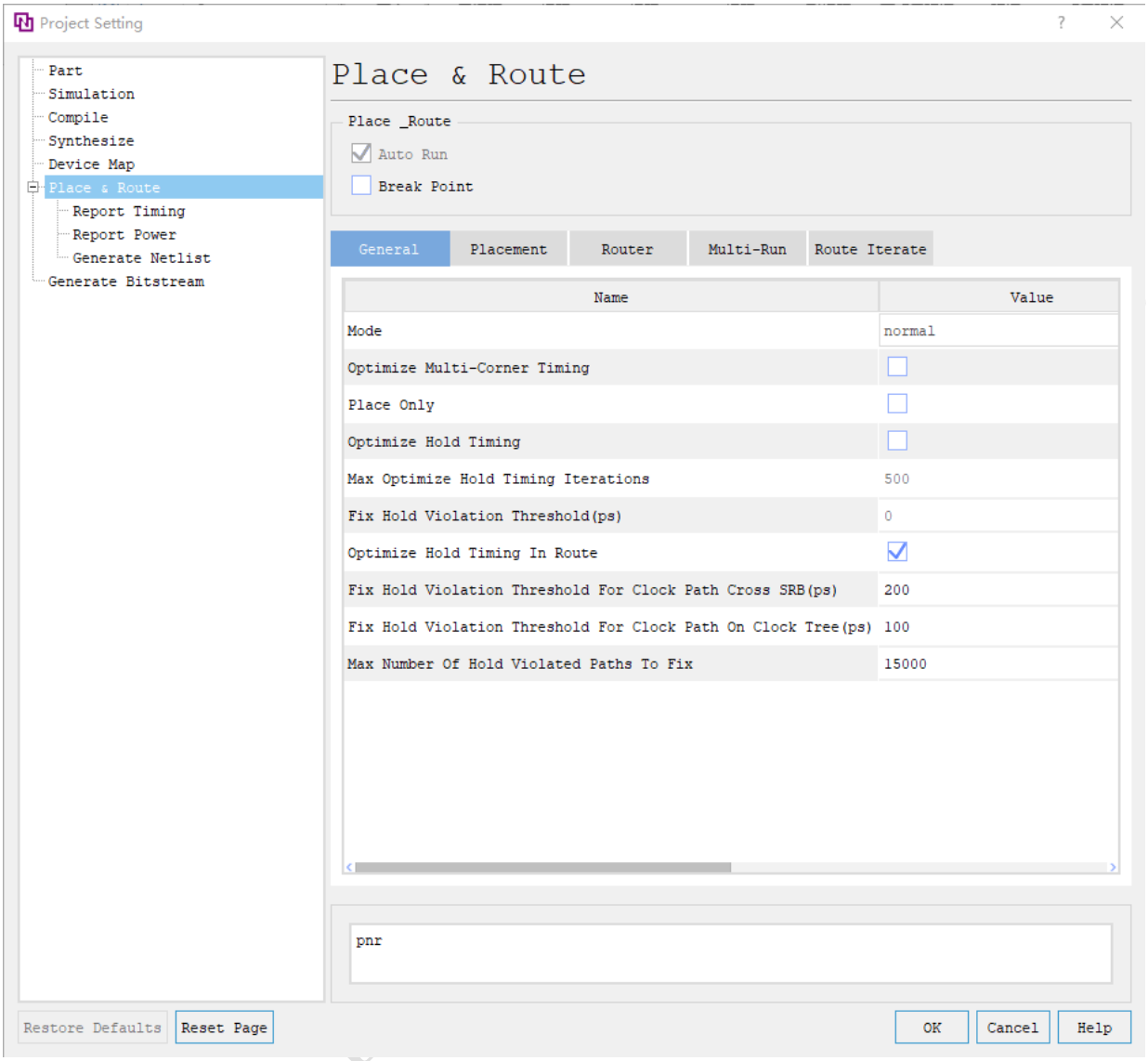


Figure6-11

[Mode]: There are three options]: fast, normal, and performance. Fast mode has the quickest run time but the worst performance; normal mode balances performance and speed, meeting general design requirements; performance mode has the best performance but the longest run time.

[Optimize multi-corner timing]: Whether to enable the multi-corner function when fixing hold violations. When checked, hold fix will fix hold violations for both slow corners and fast corners. If not checked, hold fix will by default fix violations for fast corners. It is only valid when [Optimize Hold Timing] is checked;

[Place Only]: Only placement without routing;

[Optimize Hold Timing]: Whether to fix violations when the design has hold violations. When checked, fix hold violations;

[Max Optimize Hold Timing Iterations]: When [Optimize Hold Timing] is checked, this option

indicates the maximum number of iterations for fixing violations, stopping the fix when the maximum number is reached;

[Fix Hold Violation Threshold(ps)]: When [Optimize Hold Timing] is checked, this option can be used to optimize the hold timing for timing paths with slack less than this value, thus leaving a margin for hold timing (unit]: ps);

[Optimize Hold Timing In Route]: Indicates whether to fix violations during the routing process when the design has hold violations, with the default being checked and fixed;

[Fix Hold Violation Threshold For Clock Path Cross SRB(ps)]: When [Optimize Hold Timing In Route] is checked, this option can be used to optimize the hold timing for timing paths with slack less than this value, thus leaving a certain margin for hold timing (unit: ps). This parameter is only used for those violation paths whose clock paths cross SRB. If this parameter is set too high, there may be a long fixing time due to a large number of violations needing to be fixed; if set too low, there is a risk of hold timing violations. Suggested parameter value setting range is [0, 1800];

[Fix Hold Violation Threshold For Clock Path On Clock Tree (ps)]: When [Optimize Hold Timing In Route] is checked, this option can be used to optimize the hold timing for timing paths with slack less than this value, thus providing a certain margin for hold timing (unit]: ps). This parameter is only used for those violation paths whose clock paths do not cross SRB, with a setting range of [0, 200];

[Max Number Of Hold Violated Paths To Fix]: When [Optimize Hold Timing In Route] is checked, this option can be used to limit the number of paths that can be processed during the hold fix process. If the limit is exceeded, hold violation will not be fixed. The range is set to [0,100000];

6.4.2 [Placement] Page

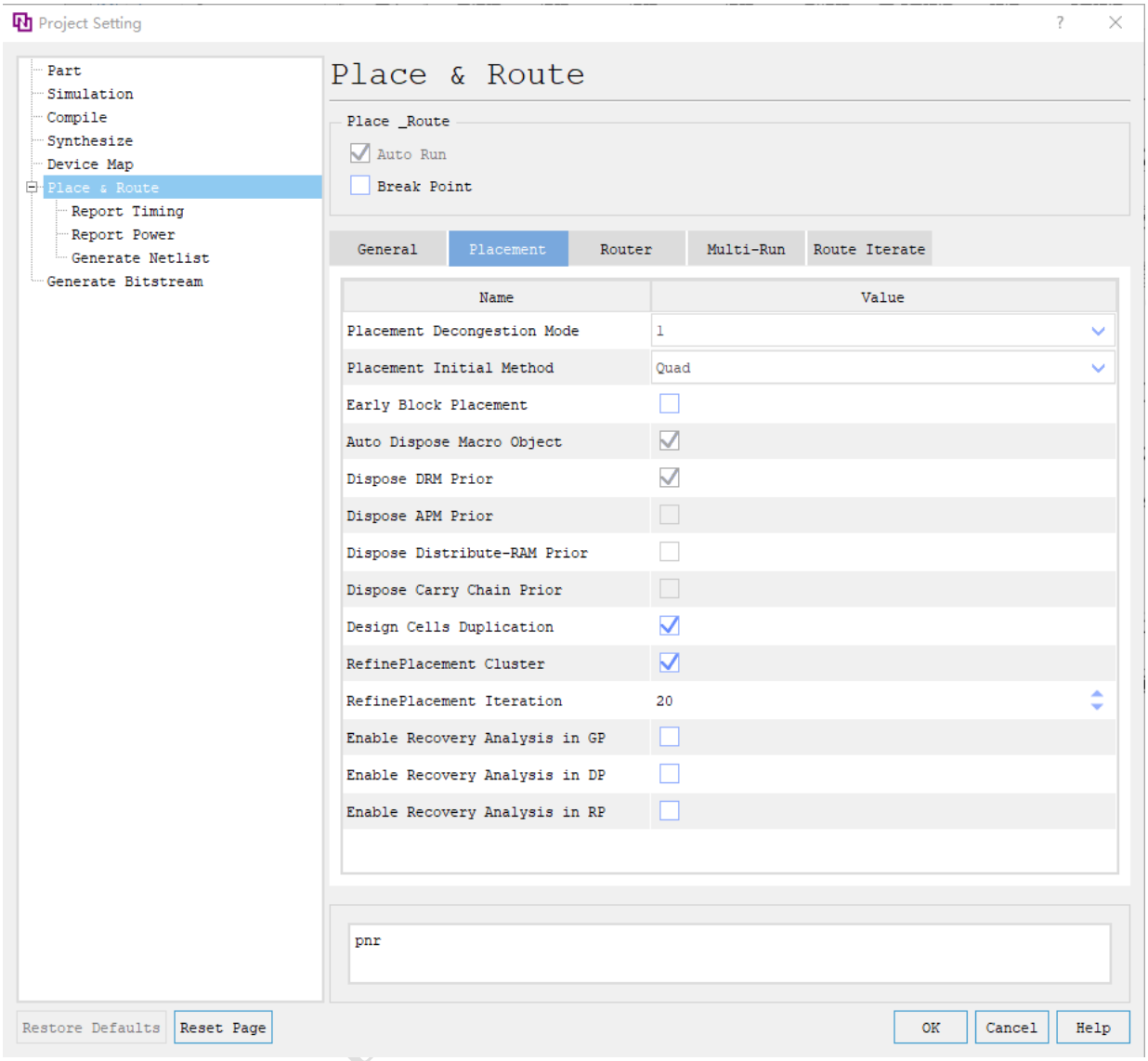


Figure6-12

[Placement Decongestion Mode]: This option sets the area expansion factor based on the maximum proportion of the user's main resources (typically LUT, FF, APM, and DRM). The value can be set to any integer of [0,1,2,3,4,5,6,7,8,9]. Generally, the larger the set mode value, the larger the placement range and the faster the routing speed, but the timing will become worse. When the utilization rate of resources (LUT, FF, APM, and DRM) exceeds 70%, the larger the set mode value, the more uncertain the routing speed will become. It is recommended that the parameter value be set to 0, 1, 2, 3, or 4;

[Placement Initial Method]: This option is used to select the starting position for the global placement. If this parameter is set to Centre, it indicates that the starting position begins from the middle of the chip; if it is set to Quad, it indicates that the placement starting position depends on the positional constraints and IO placement results;

[Early Block Placement]: This option enables the switch for priority placement of Macro Objects. If this option is selected, priority placement of Macro Objects will be allowed, which can speed up the routing process. If this option is not selected, priority placement of Macro Objects will be prohibited. This option must be selected to operate the [Auto Dispose Macro Object] option;

[Auto Dispose Macro Object]: This option is the PDS software's adaptive switch for priority placement of Macro Instances. If this option is not selected, users can choose whether to prioritize placement of DRM, APM, DRAM, and Carry Chain respectively through the [Dispose DRM Prior], [Dispose APM Prior], [Dispose Distribute-RAM Prior], and [Dispose Carry Chain Prior] options;

[Dispose DRM Prior]: If this option is selected, the placement of DRM will be prioritized during placement. This option can be operated when [Auto Dispose Macro Object] is not selected;

[Dispose APM Prior]: If this option is selected, the placement of APM will be prioritized during placement. This option can be operated when [Auto Dispose Macro Object] is not selected;

[Dispose Distribute-RAM Prior]: If this option is selected, the placement of Distribute-RAM will be prioritized during placement. This option can be operated when [Auto Dispose Macro Object] is not selected;

[Dispose Carry Chain Prior]: If this option is selected, the placement of Carry Chain will be prioritized during placement. This option can be operated when [Auto Dispose Macro Object] is not selected;

[Design Cells Duplication]: placement algorithm timing optimization; when it is selected, adaptive insertion and duplication of logic cells algorithm is used to optimize placement timing;

[RefinePlacement Cluster]: If this option is selected, the closure of placement timing performance will be improved;

[RefinePlacement Iteration]: This option is used to set the number of detailed placement closure iterations. When the detailed placement timing has not fully converged, it is recommended to set this parameter value larger, which will result in a longer detailed placement closure time;

[Enable Recovery Analysis in GP]: Enables recovery timing analysis during global placement;

[Enable Recovery Analysis in DP]: Enables recovery timing analysis during detail placement;

[Enable Recovery Analysis in RP[]: Enables recovery timing analysis during replication placement;

6.4.3 [Router] Page

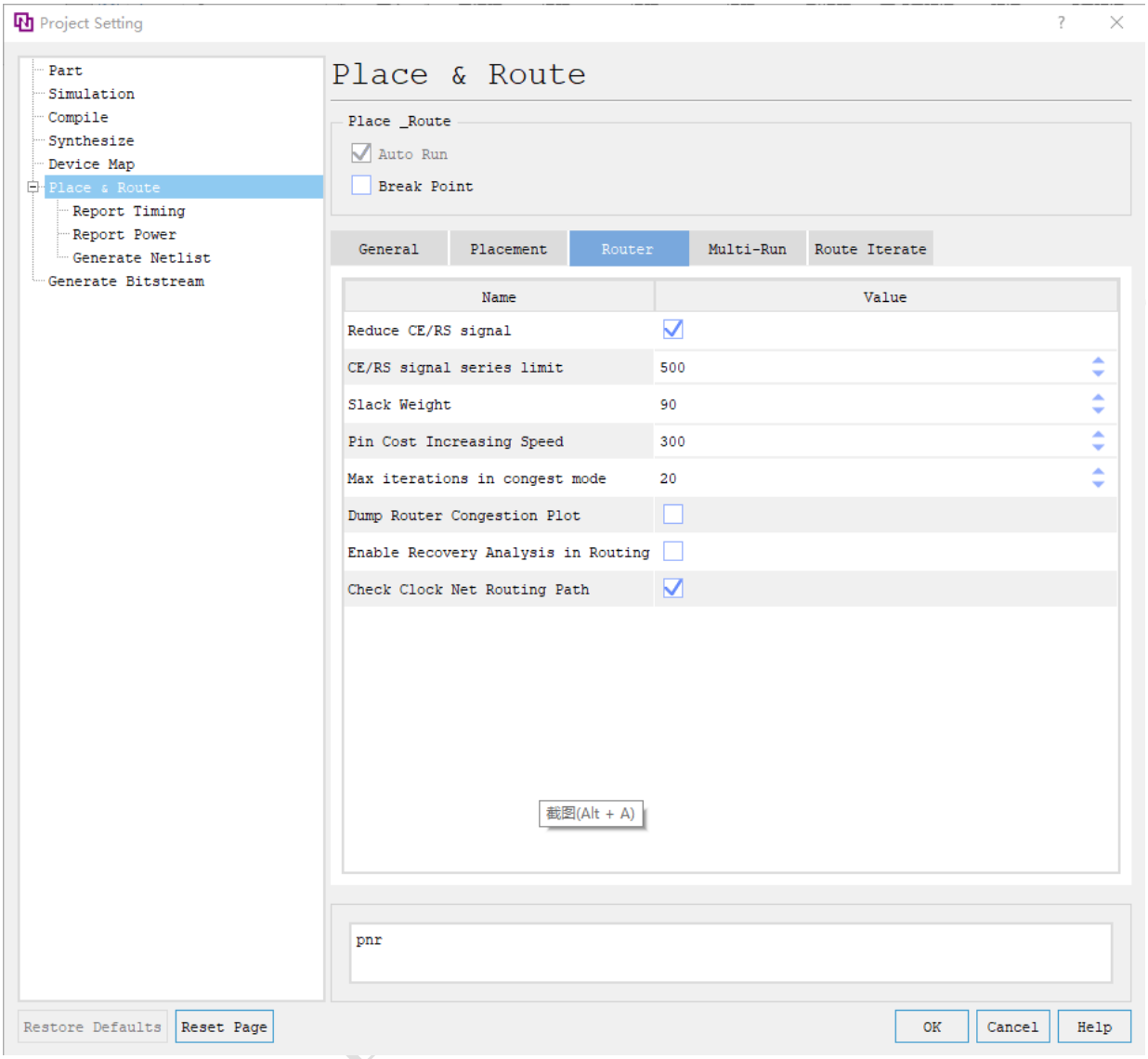


Figure6-13

[Reduce CE/RS Signal]: If this option is selected, Control Chain will be used;

[CE/RS Signal Series Limit]: If this option is selected, the maximum length of the CE/RS signal chain will be controlled;

[Slack Weight]: Controls the weight of timing factors during routing;

[Pin Cost Increasing Speed]: Controls the speed of increasing the cost value for congested ports;

[Max iterations in congest mode]: Sets the maximum number of routing iterations in congestion control mode;

[Dump Router Congestion Plot]: If this option is selected, the congestion node files will be printed. The default is off. After selection, when the routing process is run, "CongestionPlot" and "CongestionNetInfo" folders will be created in the "place_route" directory, recording congestion node files generated in each iteration during the detailed routing process and net information of

routing conflicts, respectively. In PDS, right-click anywhere within [Report Summary > Flow Summary] and select [Open Congestion Plot] to open the congestion map. This will automatically load the congestion node files from the "CongestionPlot" folder, providing a visualization of the congestion situation for each iteration during the detailed routing process.

[Enable Recovery Analysis in Routing]: If this option is selected, recovery timing analysis will be enabled during the routing process;

[Check Clock Net Routing Path]: Checks whether the clock signal is routed using SRB. When this option is selected, if the clock signal uses a SRB, the path will be flagged for a warning, and a Critical Warning message will be printed in the Console:

C: Route-2036;

6.4.4 [Multi-Run] Page

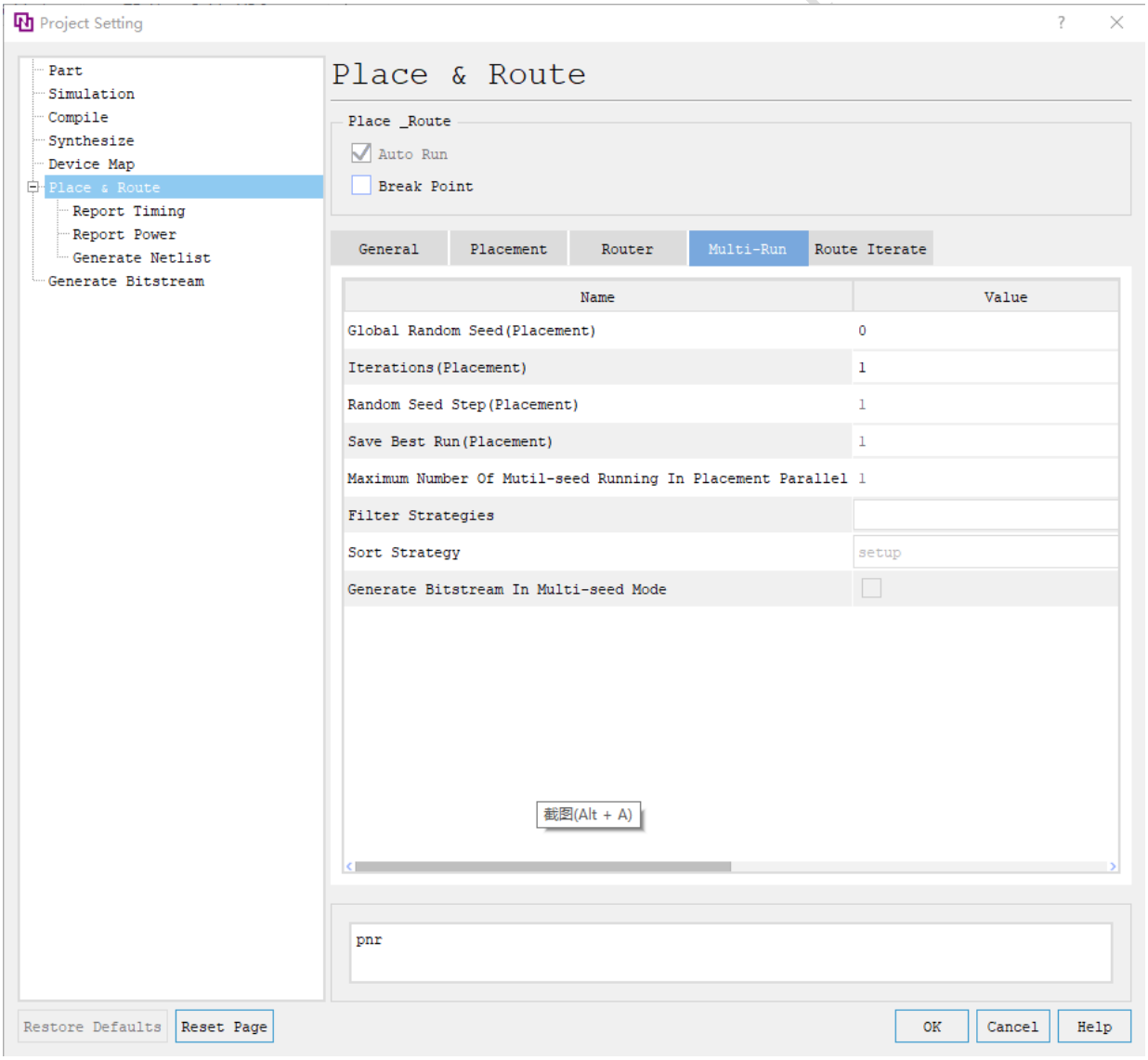


Figure6-14

[Global Placement Random Seed]: The placement algorithm of PDS is a global optimization algorithm, which depends on initial values. Different initial values can lead to different optimization results. The main function of this parameter is to provide random positions for the placement algorithm, enabling users to obtain the best performance placement and routing results;

[Iterations(Placement)]: The number of times to execute multi-seed placement and routing, with a minimum of 1;

[Random Seed Step(Placement)]: The seed step size for executing multi-seed placement and routing. It can only be set when [Iterations(Placement)] is greater than 1;

[Save Best Run (Placement)]: The number of best results to save from multiple-seed placement and routing executions. It can only be set when [Iterations (Placement)] is greater than 1;

[Maximum Number Of Multi-seed Running In Parallel]: The maximum number of seeds that can be executed in parallel. It can only be set when [Iterations (Placement)] is greater than 1;

[Filter Strategies]: Sets the filtering conditions for selecting the best seed, with the highest priority. It can only be set when [Iterations (Placement)] is greater than 1;

[Sort Strategy]: Sets the sorting rules for selecting the best seed, with the second-highest priority. It can only be set when [Iterations (Placement)] is greater than 1;

[Generate Bitstream In Multi-seed Mode]: If this option is selected, bitstreams for multiple seeds will be generated. It can only be selected when [Iterations (Placement)] is greater than 1;

6.4.5 [Route Iterate] Page

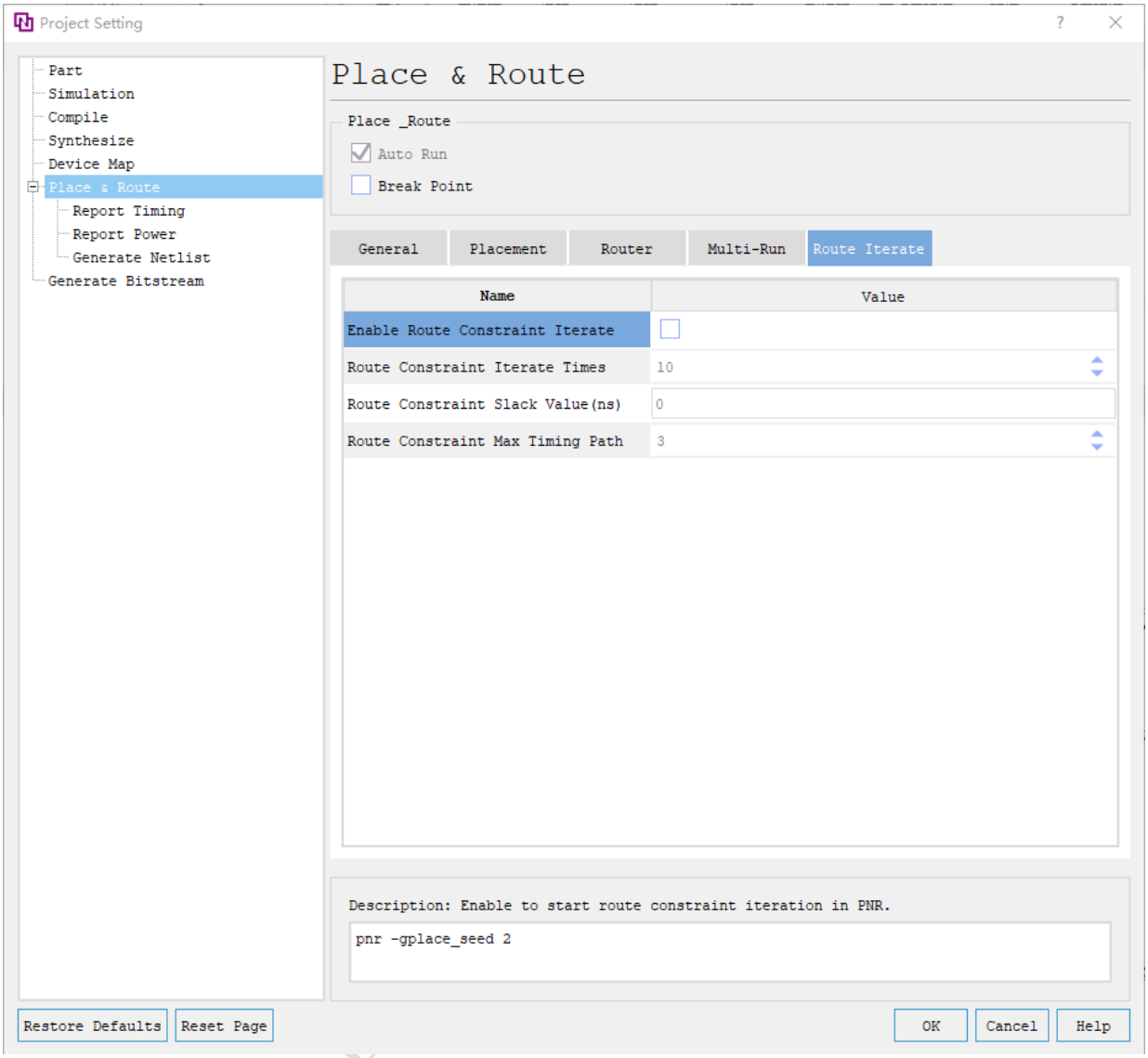


Figure6-15

[Enable Route Constraint Iterate]: The switch for routing iteration, used to determine whether to enable automatic iteration. The default is off, and no routing iteration is performed. If it is selected, automatic routing iterations will be performed based on the specified configurations.

[Route Constraint Iterate Times]: The maximum number of performing automatic routing iterations. When the maximum number of iterations is reached, the iteration will be automatically terminated. The default value is 10 times;

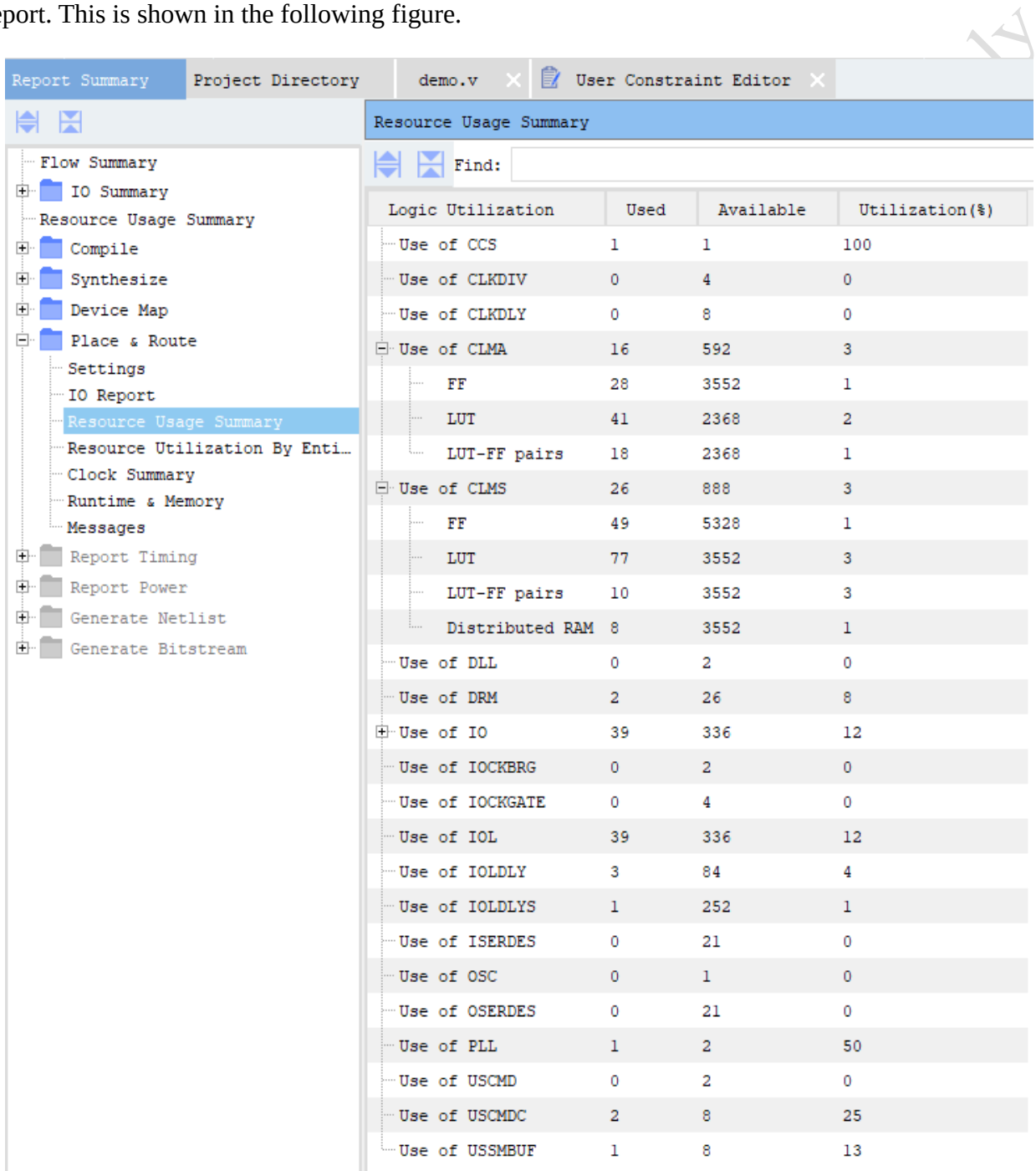
[Route Constraint Slack Value(ns)]: Specifies the setup slack threshold for the nets in the generated .rcf file. If it is not set, the default is 0.

[Route Constraint Max Timing Path]: Specifies the maximum number of timing paths in the timing report for each iteration. If it is not set, the default value is 3. The number of iterations and the setup slack threshold for the nets are also conditions for exiting a routing iteration. If either of the conditions is met during the routing process, automatic iteration will terminate. Note: Automatic

routing iteration cannot be used in conjunction with the multi-seed placement function in Multi-Run.

6.4.6 Place & Route Report

Double-click [Place & Route] in [Flow] to complete the placement and routing and generate a report. This is shown in the following figure.



Logic Utilization	Used	Available	Utilization(%)
Use of CCS	1	1	100
Use of CLKDIV	0	4	0
Use of CLKDLY	0	8	0
Use of CLMA	16	592	3
FF	28	3552	1
LUT	41	2368	2
LUT-FF pairs	18	2368	1
Use of CLMS	26	888	3
FF	49	5328	1
LUT	77	3552	3
LUT-FF pairs	10	3552	3
Distributed RAM	8	3552	1
Use of DLL	0	2	0
Use of DRM	2	26	8
Use of IO	39	336	12
Use of IOCKBRG	0	2	0
Use of IOCKGATE	0	4	0
Use of IOL	39	336	12
Use of IOLDLY	3	84	4
Use of IOLDLYS	1	252	1
Use of ISERDES	0	21	0
Use of OSC	0	1	0
Use of OSERDES	0	21	0
Use of PLL	1	2	50
Use of USCMD	0	2	0
Use of USCMDC	2	8	25
Use of USSMBUF	1	8	13

Figure6-16

6.4.7 DE Tool

DE stands for Design Editor, whose main functions include viewing the chip structure, viewing the placement and routing results, manually modifying the placement and routing results, and entering commands to perform various manual placement and routing operations. The following shows how to open the DE tool:

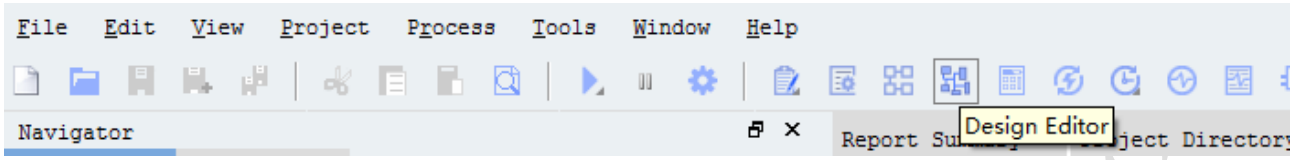


Figure6-17

For detailed information and usage of the DE tool, please see the "Design_Editor_User_Guide" at ..\pango\PDS_.....\doc\.

6.5 Report Timing

In the [Report Timing] process, a timing report is generated.

6.5.1 [Timing] Page

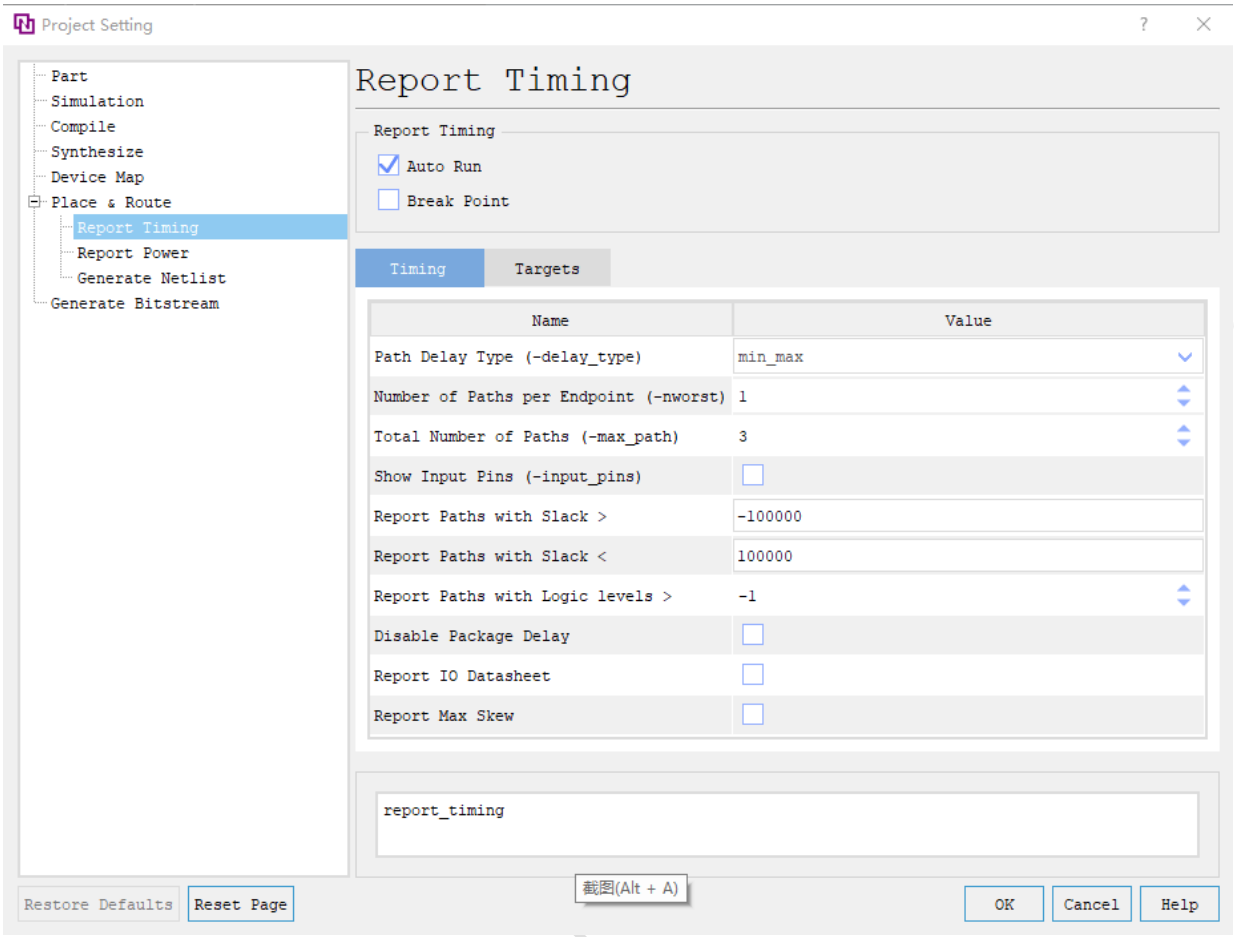


Figure6-18

[Path Delay Type (-delay_type)]: Sets the type of timing analysis, with options including [max], [min], and [min_max]. If "max" is selected, only setup analysis will be conducted on the timing path; if "min" is selected, only hold analysis will be conducted on the timing path; if "min_max" is selected, both setup and hold analyses will be conducted on the timing path;

[Number of Paths per Endpoint (-nworst)]: Sets the maximum number of timing paths to report for each endpoint;

[Total Number of Paths (-max_path)]: Sets the maximum number of paths for the timing report. The timing analysis tool sorts the timing paths by their slack, and the maximum number of paths will not exceed the number specified by this setting;

[Show Input Pins (-input_pins)]: Displays input pins. If this option is selected, the information of the input ports will be displayed in the timing path. If it is not selected, the information of the input ports will not be displayed in the timing path. By default, it is not selected;

[Report Paths with Slack >]: Sets the minimum range of slack for the report. Timing paths with slack less than the set value will not be displayed;

[Report Paths with Slack <]: Sets the maximum range of slack for the report. Timing paths with

slack greater than the set value will not be displayed;

[Report Paths with Logic Levels >]: Sets the minimum range of Logic Levels for the report. For setup/recovery timing analysis, timing paths with logic levels less than or equal to this value will not be displayed. This setting does not affect the timing paths analyzed for hold/removal;

[Disable Package Delay]: Sets whether the timing report includes the delay contributed by the package pins. If this option is selected, the package delay will not be included in the timing report paths;

[Report IO Datasheet]: Sets whether the timing report includes the timing characteristics of IO, including Input Ports Setup/Hold, Output Ports Clock-to-out, Combinational Delays, Setup/Hold times for input bus, Max/Min delays for output bus;

[Report Max Skew]: Sets whether the timing report includes the timing characteristics of the set_max_skew command as defined by user constraints. If this option is not selected, the final timing report will not include the related content, even if there is a "set_max_skew" command constraint;

6.5.2 [Targets] Page

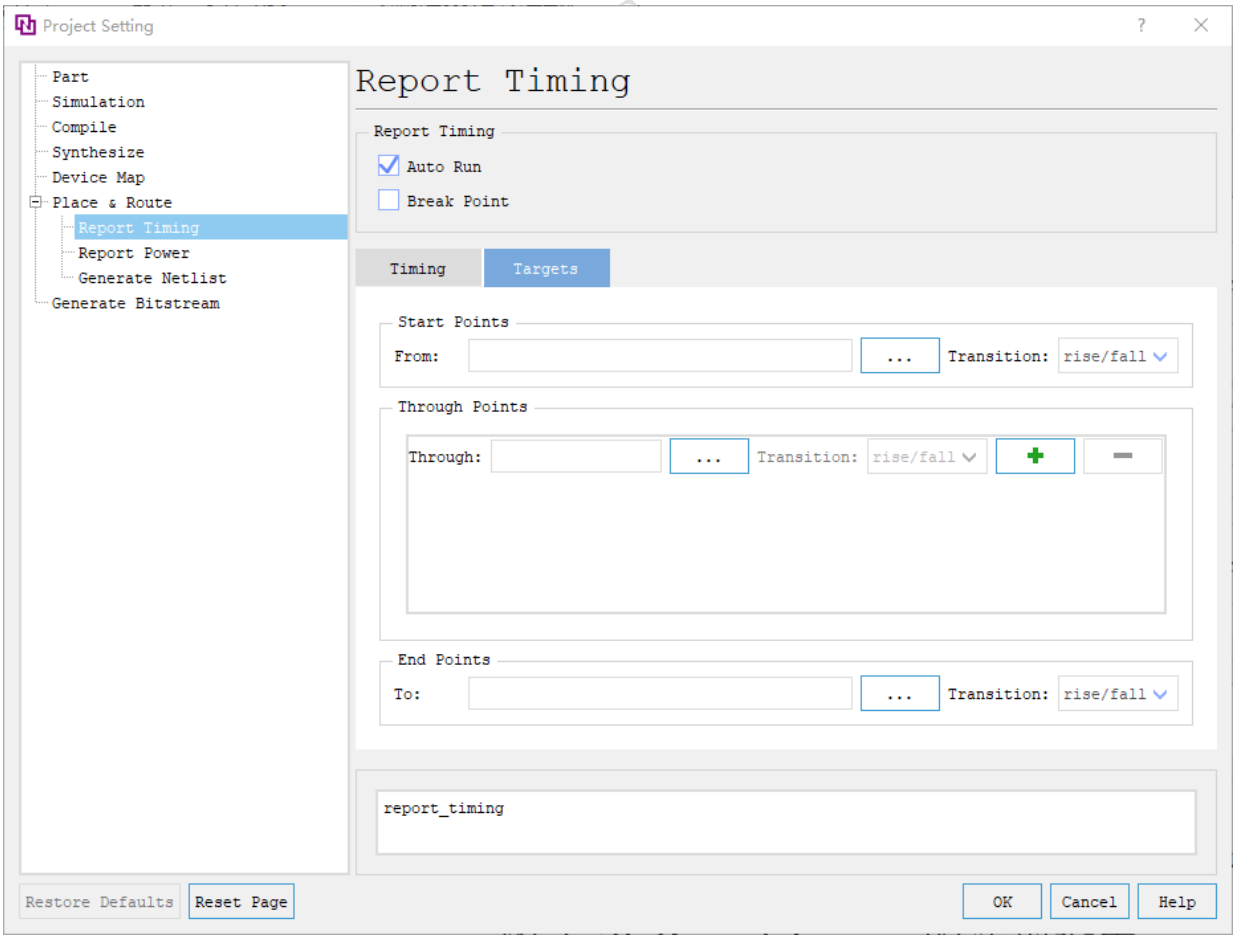


Figure6-19

[From], [Through], and [To] can specify the from/through/to nodes of the timing paths in the timing report.

6.5.3 Timing Report

First, it is necessary to apply appropriate timing constraints to the clocks and timing paths for the timing report to display specific content. Double-click [Report Timing], and the timing report will be generated. The corresponding directory tree is shown in the figure below.

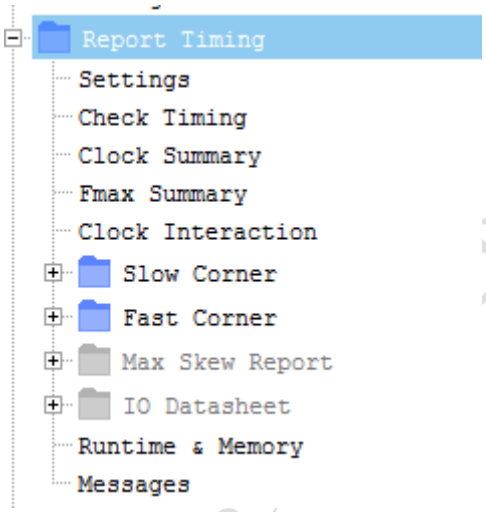


Figure6-20

- Settings

Displays the current settings for the timing analysis.

- Check Timing

Check Timing is used to check for issues with timing constraints in the current design before performing timing analysis, as shown in the figure below.

	Check	Number of Issues
1	no_clock	0
2	virtual_clock	1
3	unexpandable_clocks	0
4	no_input_delay	2
5	no_output_delay	39
6	partial_input_delay	0
7	partial_output_delay	0
8	io_min_max_delay_consistency	0
9	input_delay_assigned_to_clock	0
10	loops	0
11	latchs	0

Figure6-21

The main items and explanations for Check Timing are as follows:

Table 6-2

Check Item	**Description**
no_clock	Checks whether the clock ports of timing devices have clock propagation. If the clock ports of timing devices do not have clock input, timing analysis such as setup and hold time checks cannot be performed for those devices.
virtual_clock	Checks whether a virtual clock is created in the design or set but not used.
unexpandable_clocks	Checks whether a common cycle can be found within 1000 cycles between asynchronous clocks.
no_input_delay	Checks that the input ports do not have set_input_delay constraint.
no_output_delay	Checks that the output ports do not have set_output_delay constraint.
partial_input_delay, partial_output_delay	Checks if partial delay constraints are set for an input port; specifically, the port only has set_input_delay –max or set_input_delay –min constraint. Setting only partial delay constraints for a port can lead to the timing analysis tool being unable to perform a comprehensive timing analysis for that path. If the delay constraints for a port are set without specifying -min or -max, the timing analysis tool will assume that the port has the same min and max delay.
partial_output_delay	Checks if partial delay constraints are set for an output port; specifically, the port only has set_output_delay –max or set_output_delay –min constraint. Setting only partial delay constraints for a port can lead to the timing analysis tool being unable to perform a comprehensive timing analysis for that path. If the delay constraints for a port are set without specifying -min or -max, the timing analysis tool will assume that the port has the same min and max delay.
io_min_max_delay_consistency	Checks the consistency of the IO delay constraints, and the set_input_delay and set_output_delay settings. If the min delay for an IO delay is greater than or equal to the max delay, a warning is reported. For IO delay constraints where neither min nor max is specified, the software defaults to assuming that the min and max delays are equal, and this should also be reported.
input_delay_assigned_to_clock	Checks whether an input delay constraint is set for a clock port; if yes, it will be ignored.
Loops	Checks if there are combinational loops in the design, as they may not have been analyzed correctly.
latches	Checks if there are latches in the design.

- Clock Summary

Displays the basic information of a clock.

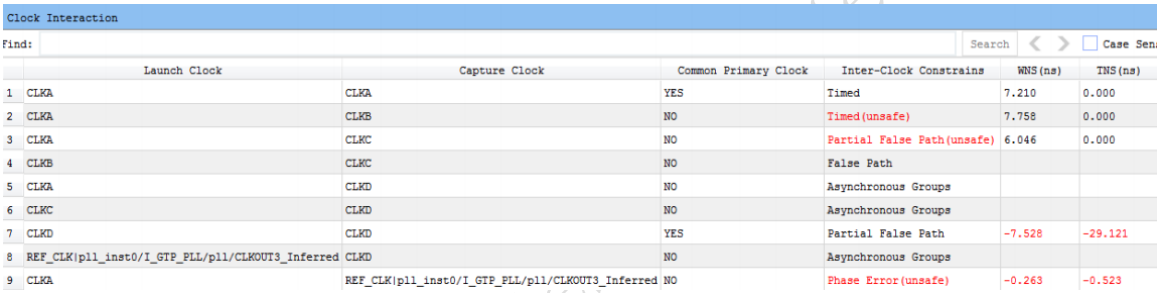
➢ Fmax Summary

Displays the maximum operating frequency of a clock. The Fmax of the clock is calculated based on the timing path with the worst slack within the same clock domain. Timing paths across clock domains do not affect the Fmax result. Since the calculation of Fmax does not consider timing paths across clock domains, this result should only be used as a reference for clock performance and not as a basis for judging timing closure.

If the maximum operating frequency of the clock does not meet the user's constraints, that is, Fmax < Requested Frequency, then the row containing that clock will be highlighted in red.

➢ Clock Interaction

Clock Interaction is used to report the relationships between clocks as well as the timing constraints in the design, as shown in the figure below.



Clock Interaction

Find: Search Case Sen

	Launch Clock	Capture Clock	Common Primary Clock	Inter-Clock Constrains	WNS(ns)	TNS(ns)
1	CLKA	CLKA	YES	Timed	7.210	0.000
2	CLKA	CLKB	NO	Timed(unsafe)	7.758	0.000
3	CLKA	CLKC	NO	Partial False Path(unsafe)	6.046	0.000
4	CLKB	CLKC	NO	False Path		
5	CLKA	CLKD	NO	Asynchronous Groups		
6	CLKC	CLKD	NO	Asynchronous Groups		
7	CLKD	CLKD	YES	Partial False Path	-7.528	-29.121
8	REF_CLK|pll_inst0/I_GTP_PLL/pll/CLKOUT3_Inferred	CLKD	NO	Asynchronous Groups		
9	CLKA	REF_CLK|pll_inst0/I_GTP_PLL/pll/CLKOUT3_Inferred	NO	Phase Error(unsafe)	-0.263	-0.523

Figure6-22

The main contents of the report are as follows:

[Launch Clock]: Indicates the launch clock;

[Capture Clock]: Indicates the capture clock;

[Common Primary Clock]: Indicates whether the primary clock for both the launch and capture clocks is consistent;

[Inter-Clock Constraints]: Indicates the relationship between the launch and capture clocks. The relationships include the following:

Table 6-3

Clock Relationship	**Description**
Timed	Indicates that the launch and capture clocks are synchronous, and the paths between them are safely constrained.
Timed(unsafe)	Indicates that the launch and capture clocks are asynchronous, and the paths between them are unconstrained.
False Path	Indicates that all the paths between tlaunch and capture clocks are defined as false paths by the "set_false_path" command.
Partial False Path	Indicates that the launch and capture clocks are synchronous, and some paths between them are defined as false paths by the user with the "set_false_path" command.
Partial False Path(unsafe)	Indicates that the launch and capture clocks are asynchronous, and some paths between them are defined as false paths by the user with the "set_false_path" command.
Asynchronous Groups	Indicates that the launch and capture clocks are defined as false paths with the "set_clock_groups" command.

Clock Relationship	Description
Phase Error(unsafe)	Indicates that at least one of the launch and capture clocks is a generated clock, and the phase relationship between the generated clock and another clock cannot be determined, indicating that the timing path between these two clocks is unsafe.

[WNS]: Indicates the worst slack for all setup and recovery paths between the launch and capture clocks;

[TNS]: Indicates the sum of all slack values less than 0 for all setup and recovery paths between the launch and capture clocks;

[Failing End Point(TNS)]: Indicates the number of end points with slack less than 0 in all setup and recovery paths between the launch and capture clocks;

[Total End Point(TNS)]: Indicates the total number of end points in all setup and recovery paths between the launch and capture clocks;

[WHS]: Indicates the worst slack for all hold and removal paths between the launch and capture clocks;

[THS]: Indicates the sum of all slack values less than 0 for all hold and removal paths between the launch and capture clocks;

[Failing End Point(THS)]: Indicates the number of end points with slack less than 0 in all hold and removal paths between the launch and capture clocks;

[Total End Point(THS)]: Indicates the total number of end points in all hold and removal paths between the launch and capture clocks.

- Worst Case Timing path

Worst Case timing path includes types such as Setup, Hold, Recovery, Removal, and Minimum Pulse Width, with summary and detailed information listed for each path in the report. Among these, the Setup, Hold, Recovery, and Removal reports are similar. The following shows the paths for Setup and Minimum Pulse Width.

Setup Path Summary

Find:

	Slack	Logic Levels	High Fanout	Start Point	End Point	Exception	Launch Clock	Capture Clock
1	-2.566	4	1	ffd[0]/opit_0/CLK	Q[3]		CLKD	CLKD
2	-2.410	3	1	ffd[0]/opit_0/CLK	Q[1]		CLKD	CLKD
3	-2.251	4	1	ffd[0]/opit_0/CLK	Q[4]		CLKD	CLKD
4	8.209	1	3	ff0[2]/opit_0/CLK	ffc[3]/opit_0_AQ_perm/Cin		CLKA	CLKC
5	8.467	0	3	ff0[2]/opit_0/CLK	ffc[2]/opit_0_A2Q21/I11		CLKA	CLKC
6	8.616	0	3	ff0[2]/opit_0/CLK	ff1[2]/opit_0/D		CLKA	CLKA
7	8.619	1	3	ff0[1]/opit_0/CLK	ffc[3]/opit_0_AQ_perm/Cin		CLKA	CLKC
8	8.851	0	3	ff0[1]/opit_0/CLK	ff1[1]/opit_0/D		CLKA	CLKA
9	8.976	0	3	ff0[3]/opit_0/CLK	ff1[3]/opit_0/D		CLKA	CLKA
10	9.220	0	1	ff1[3]/opit_0/CLK	ffb[3]/opit_0_L5Q/L0		CLKA	CLKB
11	9.272	0	1	ff1[2]/opit_0/CLK	ffb[2]/opit_0_L5Q_perm/L4		CLKA	CLKB
12	9.299	0	1	ff1[0]/opit_0/CLK	ffb[0]/opit_0_L5Q_perm/L4		CLKA	CLKB

Figure6-23

The Setup Path Summary in the figure above includes the following information from left to right: Slack, Logic Levels, High Fanout, Start Point, End Point, Exception, Launch Clock, Capture Clock, Clock Edges, Clock Skew, Launch Clock Delay, Capture Clock Delay, Clock Pessimism Removal, Requirement, Data Delay, Logic Delay, and Route Delay. In the Path Summary, rows with slack less than 0 are highlighted in red, and by default, they are displayed in ascending order of slack. Below the Path Summary is the detailed information for the selected path, including whether the path meets the timing requirements, and the path's data arrival time and data required time.

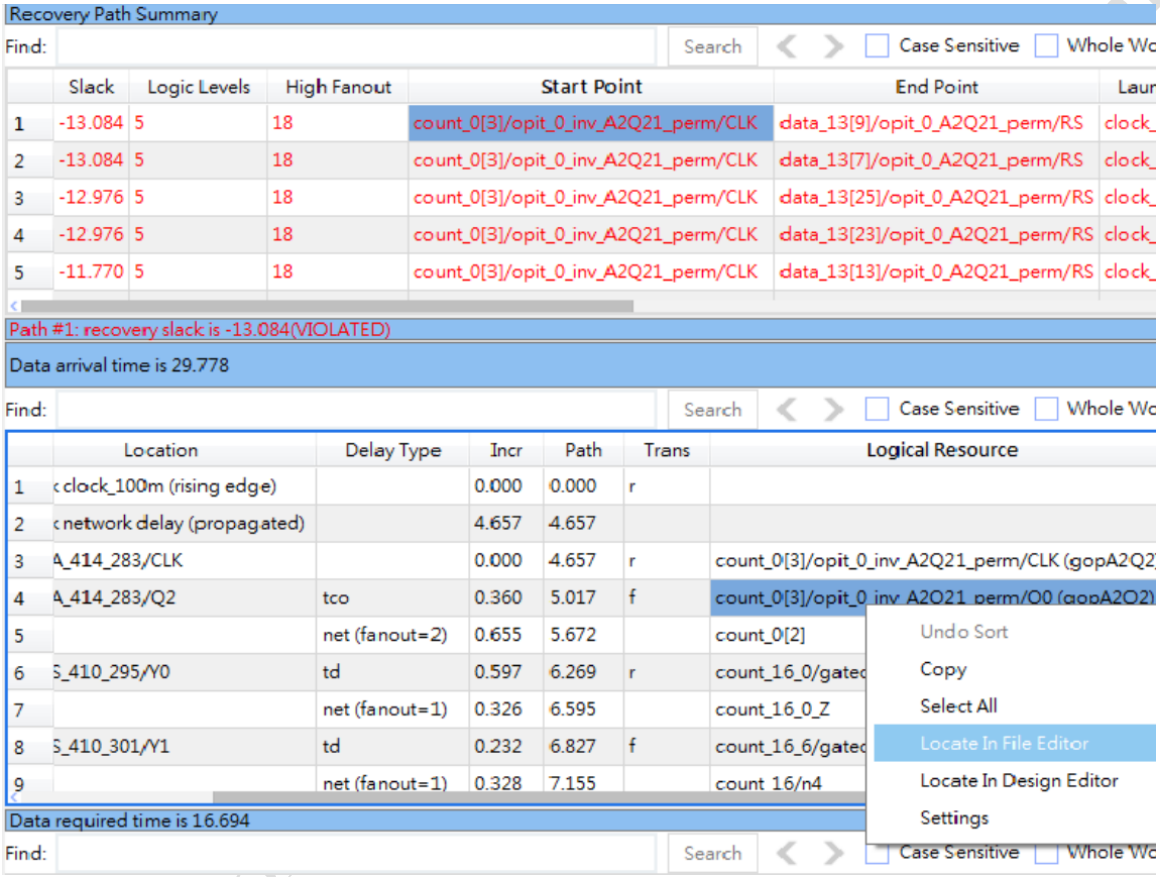


Figure6-24

The Start Point and End Point in the figure above, as well as the logical resources in the detailed path, can be right-clicked to map and jump to the corresponding RTL source files and DE for viewing.

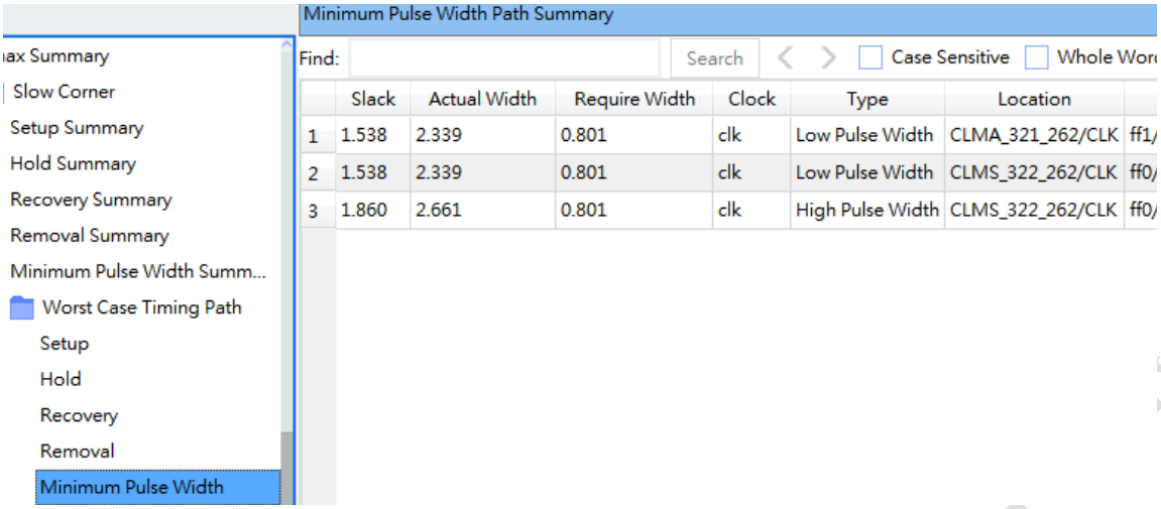


Figure6-25

The Minimum Pulse Width Path Summary is different from the Setup Path Report, as all information is already listed in the summary, eliminating the need for a detailed information list. This information includes Slack, Actual Width, Required Width, Clock, Type, Location, and Pin.

Additionally, in the Flow Summary, users can view information regarding whether timing constraints are violated. If all timing constraints are satisfied, "All Constraints Met" will be displayed; if there are timing violations, the specific unmet constraints will be displayed (highlighted in red). Users can click it to view the detailed report.

6.5.4 TA Tool

TA stands for Timing Analyzer, which is a tool used for timing analysis. Users can conduct timing analysis by importing Design DB (PnR DB or Syn DB) and SDC files. By reviewing the timing results, users can modify the timing constraints based on their needs and export the modified SDC files. Then, with the new SDC files and the timing information reported by the Design DB, users can ultimately obtain a timing constraint file that meets their expectations. Using the TA tool, users can perform timing analysis with the given Design DB and constraint files. Even if the constraint files are modified at this point, timing analysis can be directly restarted, saving a significant amount of time.

The following shows how to open the TA tool:

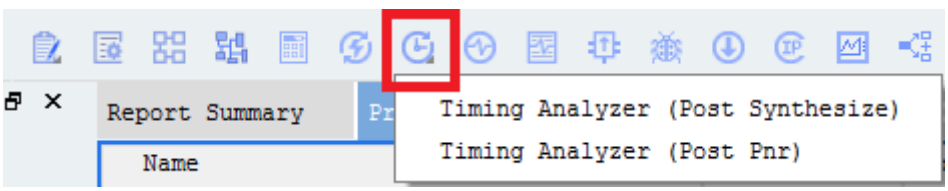


Figure6-26

For detailed information and usage of the TA tool, please see the "Timing_Analyzer_User_Guide"

at ..\pango\PDS_.....\doc\.

6.6 Generate Bitstream

[Generate Bitstream] refers to the process of creating a binary bitstream file, which is the final step in the PDS design process.

6.6.1 [General] Page

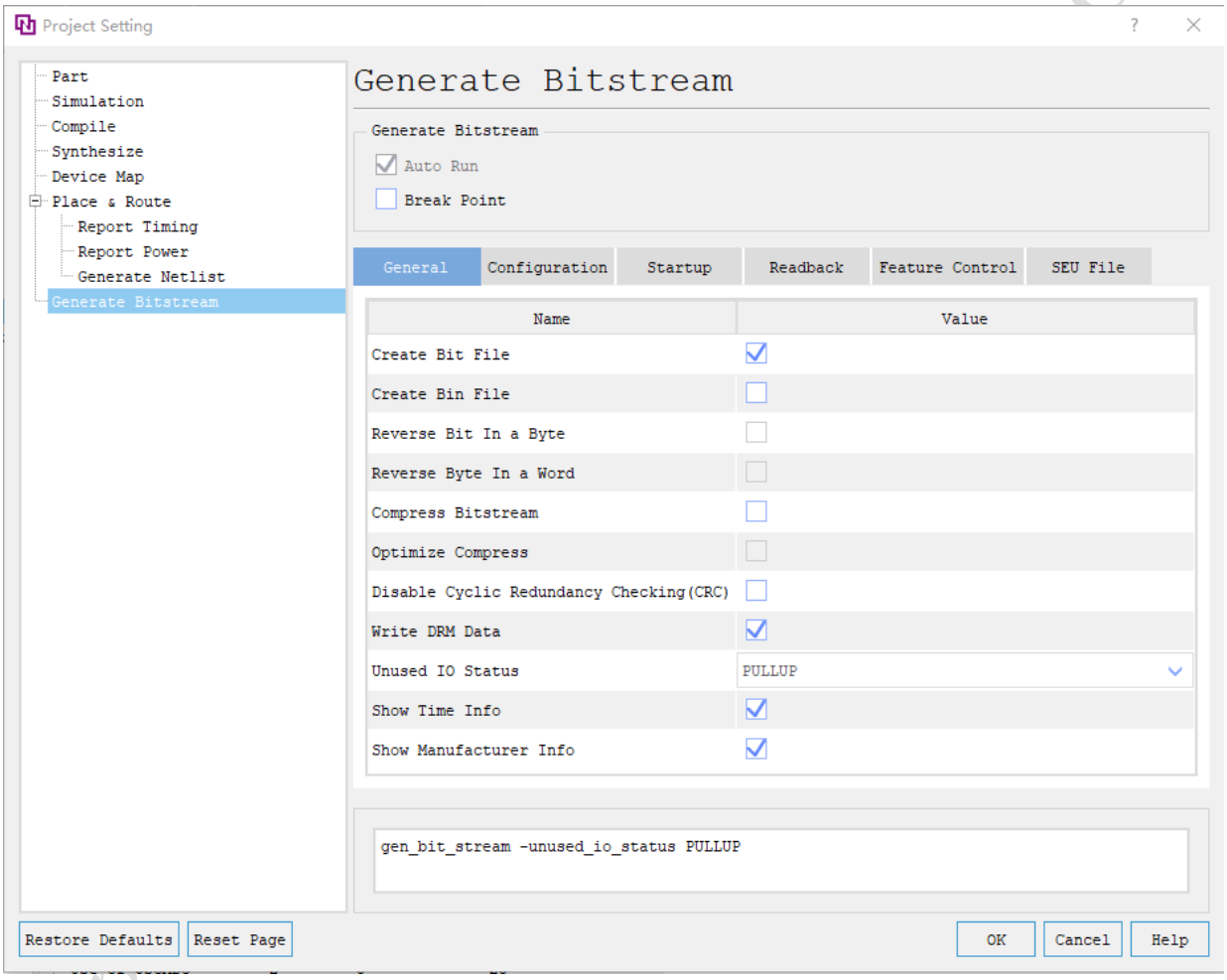


Figure6-27

[Create Bit File]: If this option is selected, a .bit file will be created;

[Create Bin File]: If this option is selected, a .bin file will be created;

[Reverse Bit In a Byte]: If this option is selected, the order of the high and low bits in each byte of the .bin file will be reversed. For example, bit7...bit0 will be reversed to bit0...bit7. The default is off;

[Reverse Byte In a Word]: If this option is selected, the order of the high and low bytes in each word (a word = 4 bytes) of the .bin file will be reversed. For example, byte3byte2...byte0 will be

reversed to byte0...byte2byte3. The default is off;

[Compress Bitstream]: If the [Create bit file] option is selected, selecting this option will create a compressed .sbit file. If this option is not selected, an uncompressed standard .sbit file will be generated. The default is off;

[Optimize Compress]: If the Compress Bitstream option is selected, selecting this option will optimize the compression process. The default value for optimization is device-dependent. Optimized compression means that when the frame data is 0, it will not be written into the compressed bitstream file;

[Disable Cyclic Redundancy Checking (CRC)]: If this option is selected, CRC validation will be ignored. The default is off;

[Write DRM Data]: If this option is selected, DRM initialization data will be written into the bitstream. The default is on. If it is deselected, the size of the bitstream file will be reduced;

[Unused IO Status]: This option is used to select the status for unused IO pins. The options include PULLUP, PULLDOWN, UNUSED, and KEEPER. The default value is PULLDOWN;

[Show Time Info]: If this option is selected, the time and date information in the bitstream header will be displayed. This option is selected by default and does not affect the user's design functionality;

[Show Manufacturer Info]: If this option is selected, the manufacturer information in the bitstream header will be displayed. This option is selected by default and does not affect the user's design functionality;

6.6.2 [Configuration] Page

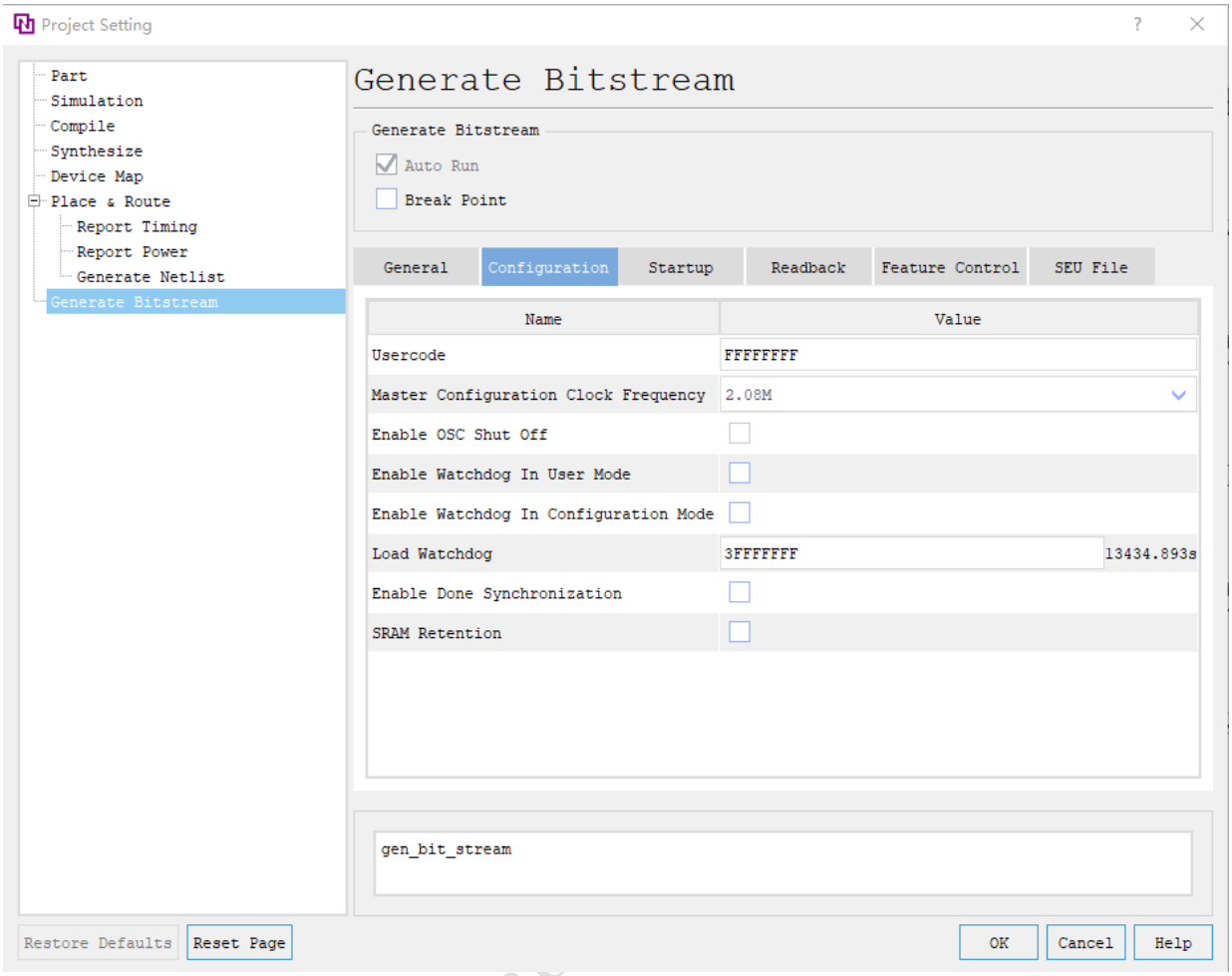


Figure6-28

[Usercode]: Assigns a value to the user identification registers by inputting a 32-bit hexadecimal user code. This value will be written into the bitstream file, with the default value being 0xFFFFFFFF;

[Master Configuration Clock Frequency]: Selects the frequency for the master configuration clock (clk). The default value is 2.08MHz;

[Enable OSC Shut Off]: If this option is selected, user logic will be allowed to shut off the OSC; if it is not selected, the OSC operates continuously.

[Enable Watchdog In User Mode]: If this option is selected, the watchdog timer will be enabled in user mode. The default is off;

[Enable Watchdog In Configuration Mode]: If this option selected, the watchdog timer will be enabled in configuration mode. The default is off;

[Load Watchdog]: Sets the watchdog timer's timeout period. The default is 3FFF_FFFF;

[Enable Done Synchronization]: If this option is selected, the Done signal will be synchronized. The default is off;

[SRAM Retention]: If this option is selected, the SRAM retention testing will be enabled. The default is off;

6.6.3 [Startup] Page

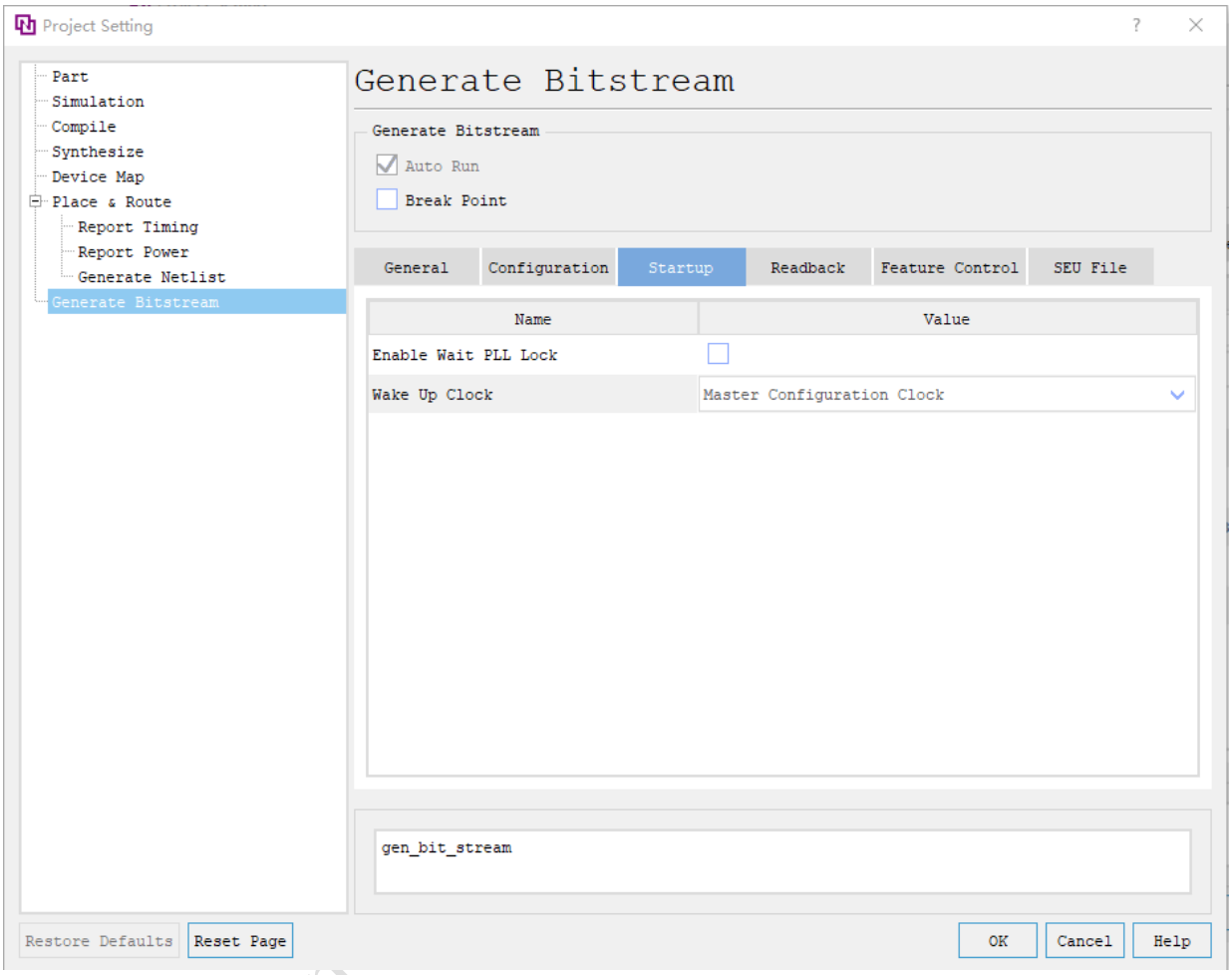


Figure6-29

[Enable Wait PLL Lock]: If this option is selected, the device will wait for the PLL to lock before waking up. The default is off;

[Wake Up Clock]: Selects the wake-up clock. The default is the master configuration clock.

6.6.4 [Readback] Page

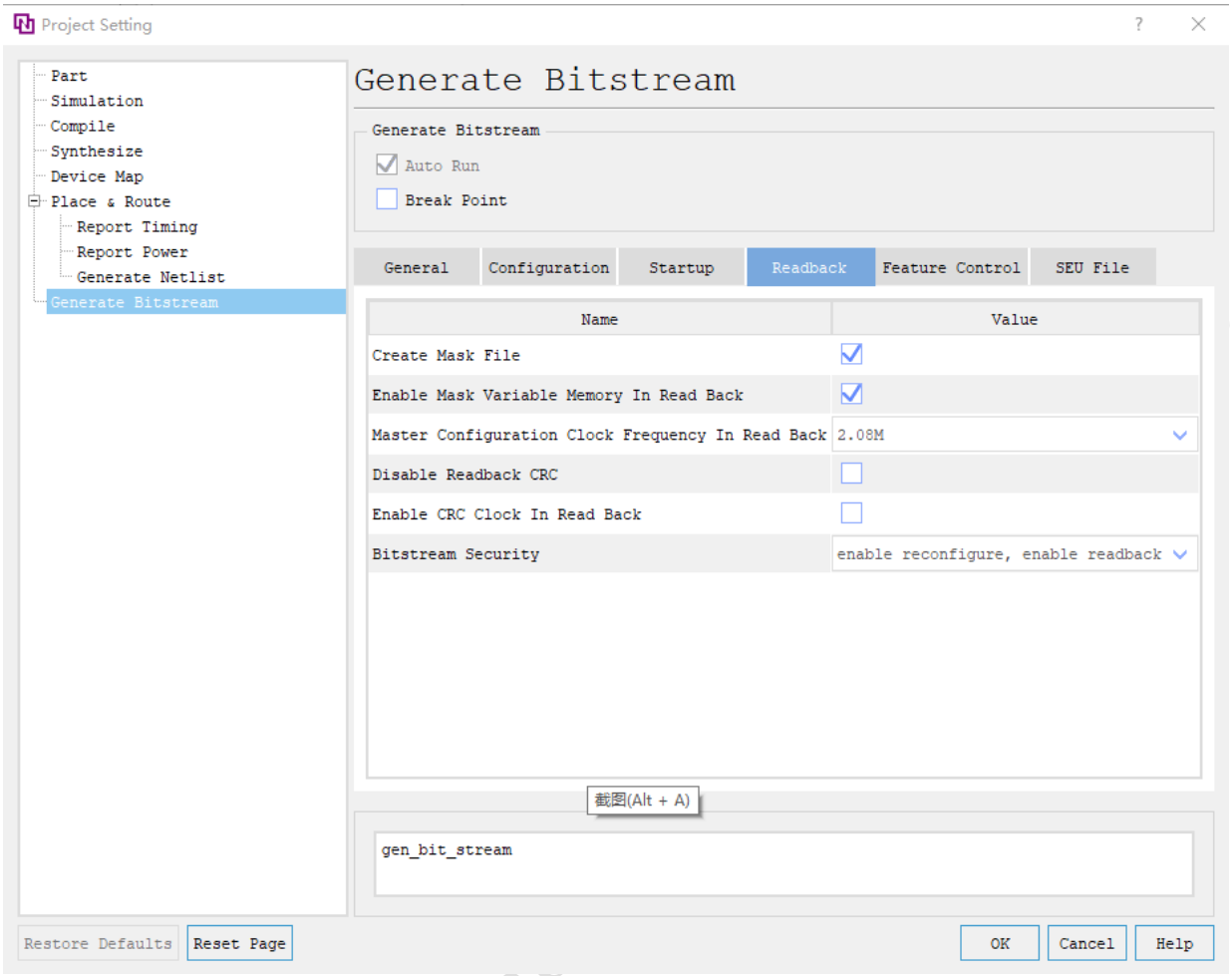


Figure6-30

[Create Mask File]: If this option is selected, a .mask file will be created to determine the comparison rules between the readback file and the original file. The default is on;

[Enable Mask Variable Memory In Read Back]: If this option is selected, the RAM area data will be masked during readback verification so that it will not be involved in the verification.

[Master Configuration Clock Frequency In Read Back]: Selects the readback clock. The default value is 2.08MHz.

[Disable Readback CRC]: If this option is selected, CRC verification will not be performed during the readback process. The default is off;

[Enable CRC Clock In Read Back]: If this option is selected, the clock for CRC will be enabled during the readback process. The default is off;

[Bitstream Security]: Controls the reconfiguration and readback of the configuration memory. By default, both reconfiguration and readback are allowed.

6.6.5 [SEU File] Page

The Soft Error Injection (SEI) functionality can be set on this page.

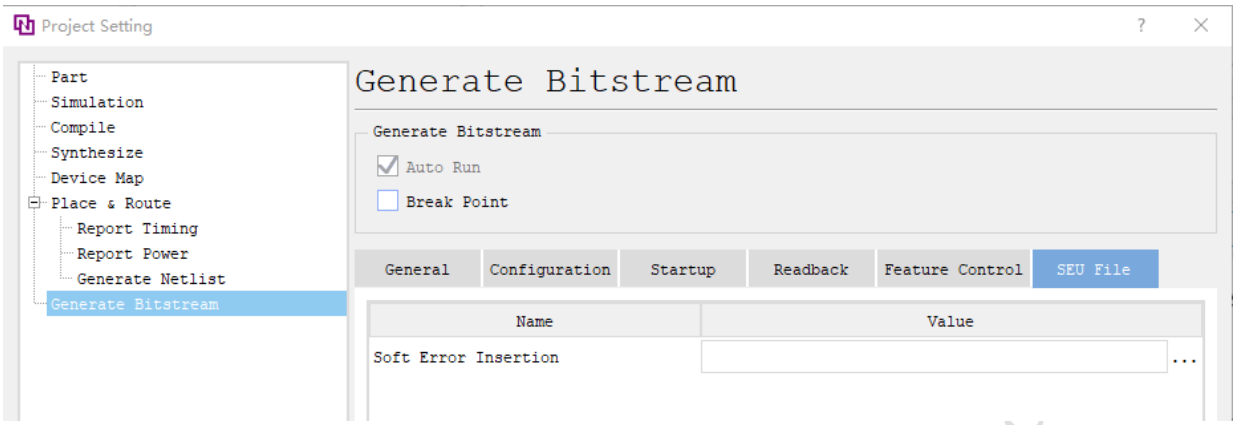


Figure6-31

Click "..." to open "SEI Editor". Click "+" to set up error injection. This is shown in the following figure.

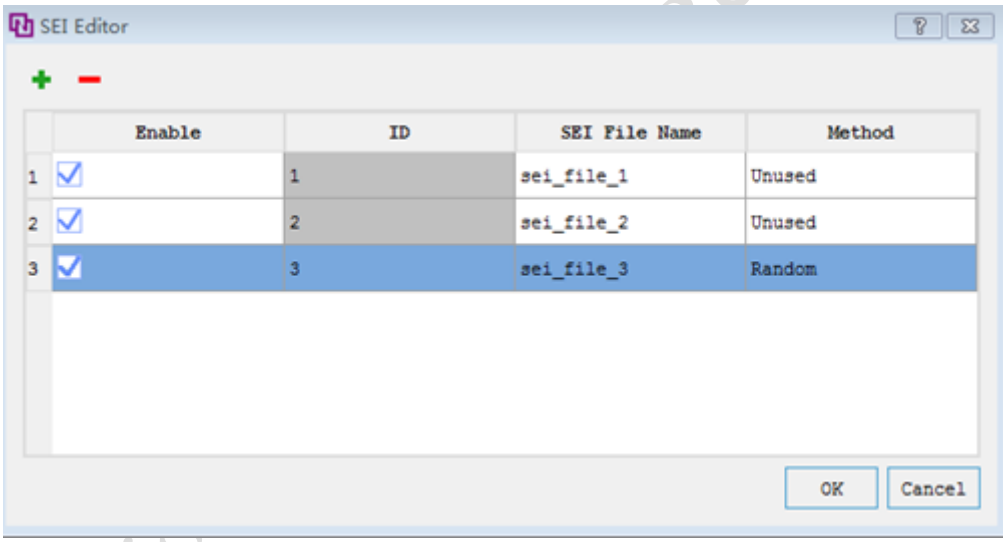


Figure6-32

[Enable]: If this option is selected, error injection will be activated, and an error-injected bitstream will be generated simultaneously when users double-click Generate Bitstream.

[ID]: Error injection sequence number;

[SEI File Name]: Name of the error-injected bitstream, which can be modified by double-clicking.

[Method]: Selects the location for error injection; options include "Unused" and "Random". "Unused" indicates error injection in the configuration space not used by the user's design; "Random" indicates error injection in the space used by the user's design.

After the bitstream is generated, click [Flow Summary > Generate Bitstream], and the error injection information will be displayed, as shown in the following figure.

```
SEI File Status:
The SEI File:sei_file_1.sbit, use method: Unused, which is inserted error in frame 1, bit 47, IOBD_0_24 block.
The SEI File:sei_file_2.sbit, use method: Unused, which is inserted error in frame 1, bit 40, IOBD_0_24 block.
The SEI File:sei_file_3.sbit, use method: Random, which is inserted error in frame 1, bit 1765, IOBS_0_159 block.
Note: frame index is from zero, bit format is MSB.
```

Figure6-33

6.7 Report Power

[Report Power] is used to assess the power consumption in the design. Parameters setting is supported.

6.7.1 [Device] Page

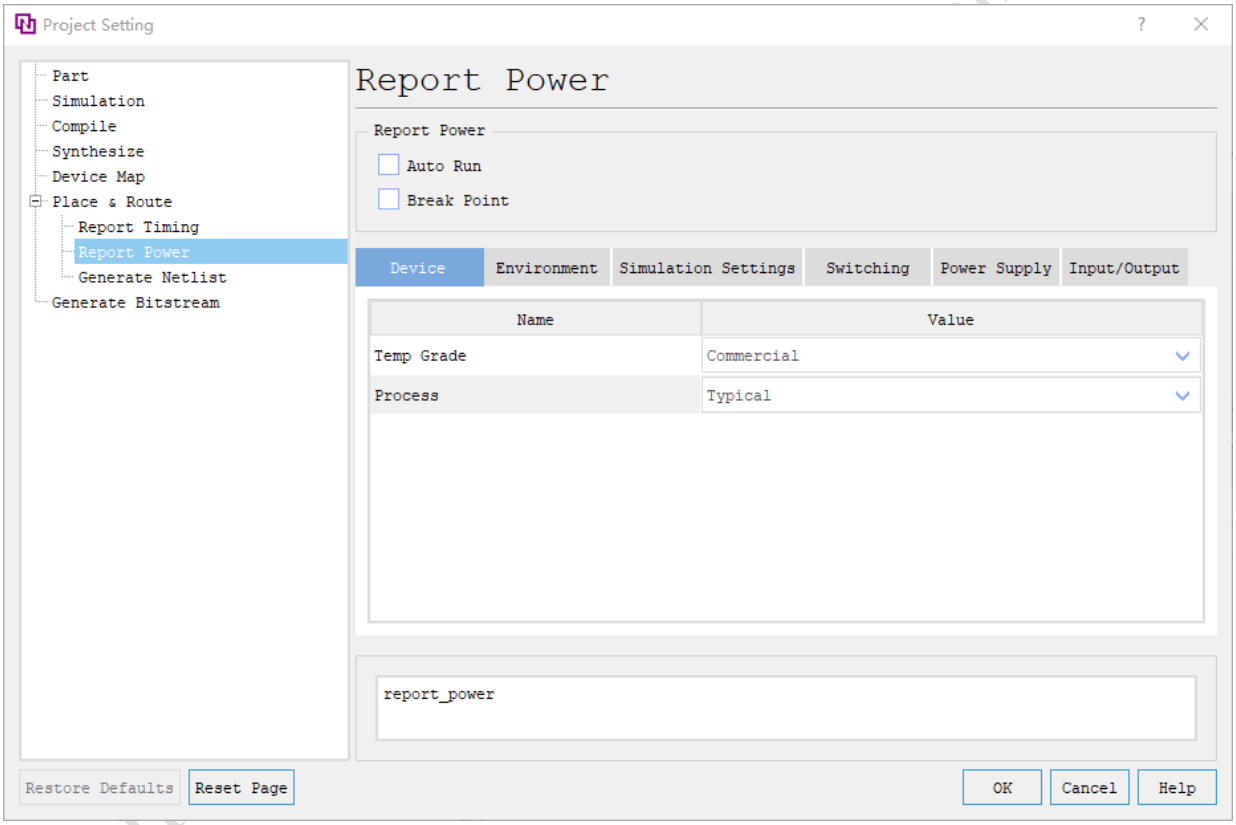


Figure6-34

[Temp Grade]: Sets the temperature grade, including Commercial and Industrial, which mainly affects the range of junction and ambient temperature settings in the Environment;

[Process]: Sets the process variation, including Typical and Worst, which affects the final power consumption assessment result.

6.7.2 [Environment] Page

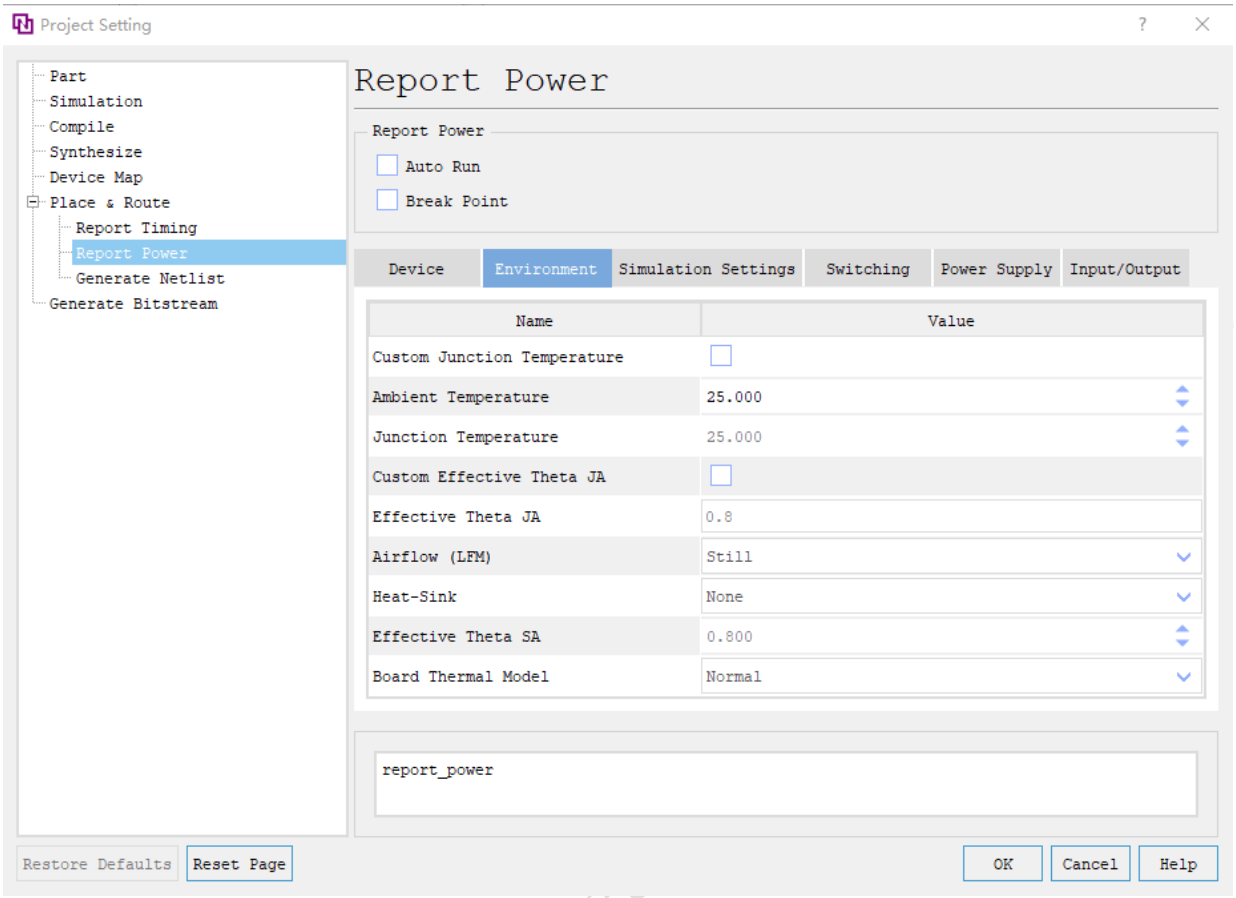


Figure6-35

[Custom Junction Temperature]: If this option is selected, only [Junction Temperature] can be set; all other settings cannot be modified. When it is not selected, the [Junction Temperature] will be the default value of 25.000;

[Ambient Temperature]: Sets the ambient temperature, determined by the [Temp Grade] setting in [Device], specifically Commercial (0~85), Industrial (-40~100);

[Junction Temperature]: Sets the junction temperature, with a default value of 25.000, also determined by the [Temp Grade] setting in [Device], specifically Commercial (0~85), Industrial (-40~100);

[Custom Effective Theta JA]: If this option is selected, users can modify the default thermal coefficient [Effective Theta JA;

[Effective Theta JA]: The equivalent thermal resistance through which the chip's heat dissipates to the surroundings, which is related to the device's process, packaging, and cooling. It can only be set if Custom Effective Theta JA is selected;

[Airflow(LFM)]: Sets the airflow condition, with options of Still, Low, Medium, and High;

[Heat-Sink]: Sets the thermal dissipation condition, with options of None, Custom, Low, Medium, and High;

[Effective Theta SA]: Sets the thermal resistance coefficient for thermal dissipation. It can be modified when [Heat-Sink] is set to "Custom";

[Board Thermal Model]: Sets the thermal dissipation condition of the board, with options of JEDC, Best, Normal, and Worst.

6.7.3 [Simulation Settings] Page

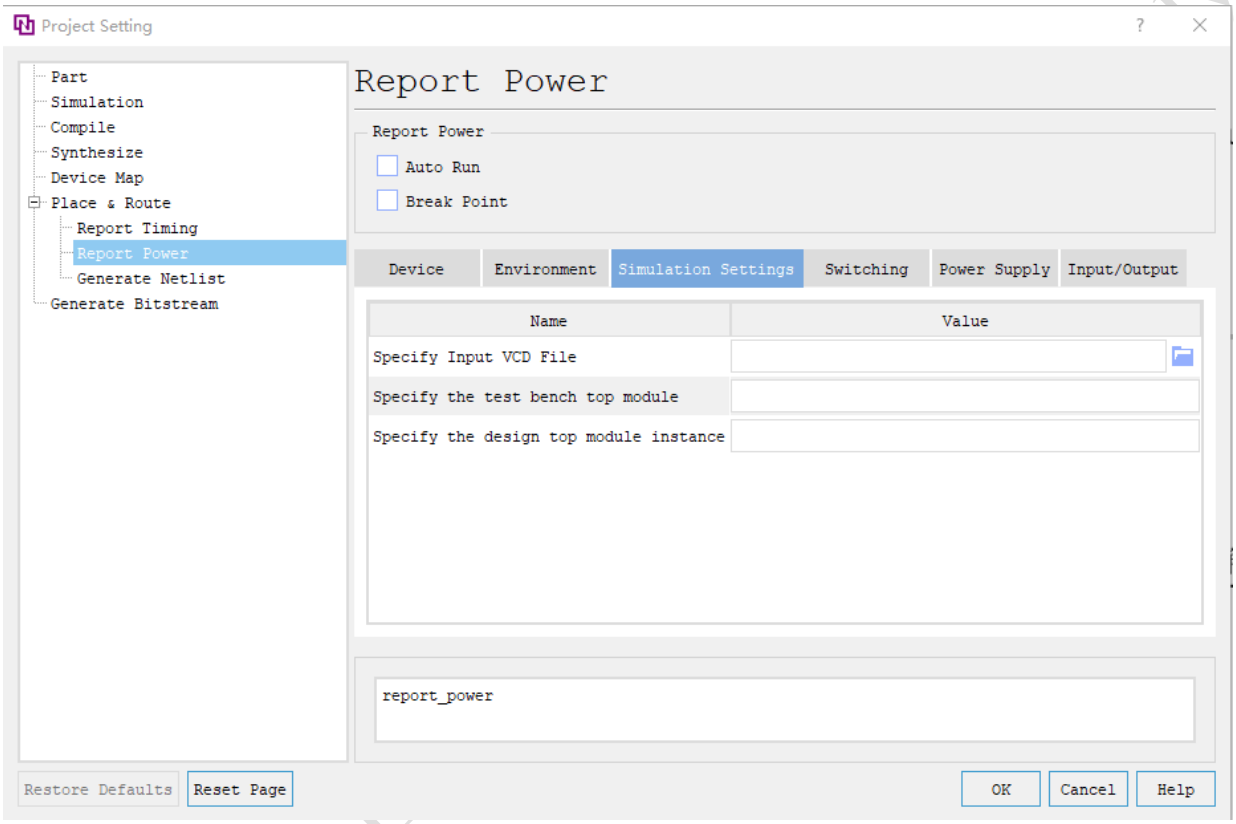


Figure6-36

[Specify Input VCD File]: The post-simulation file specified by the user, which records the simulation waveforms and through which the signal's toggles can be obtained;

[Specify the test bench top module]: The name of the top module in the simulation file specified by the user;

[Specify the design top module instance]: The instantiation name of the design's top module in the test bench, as specified by the user.

6.7.4 [Switching] Page

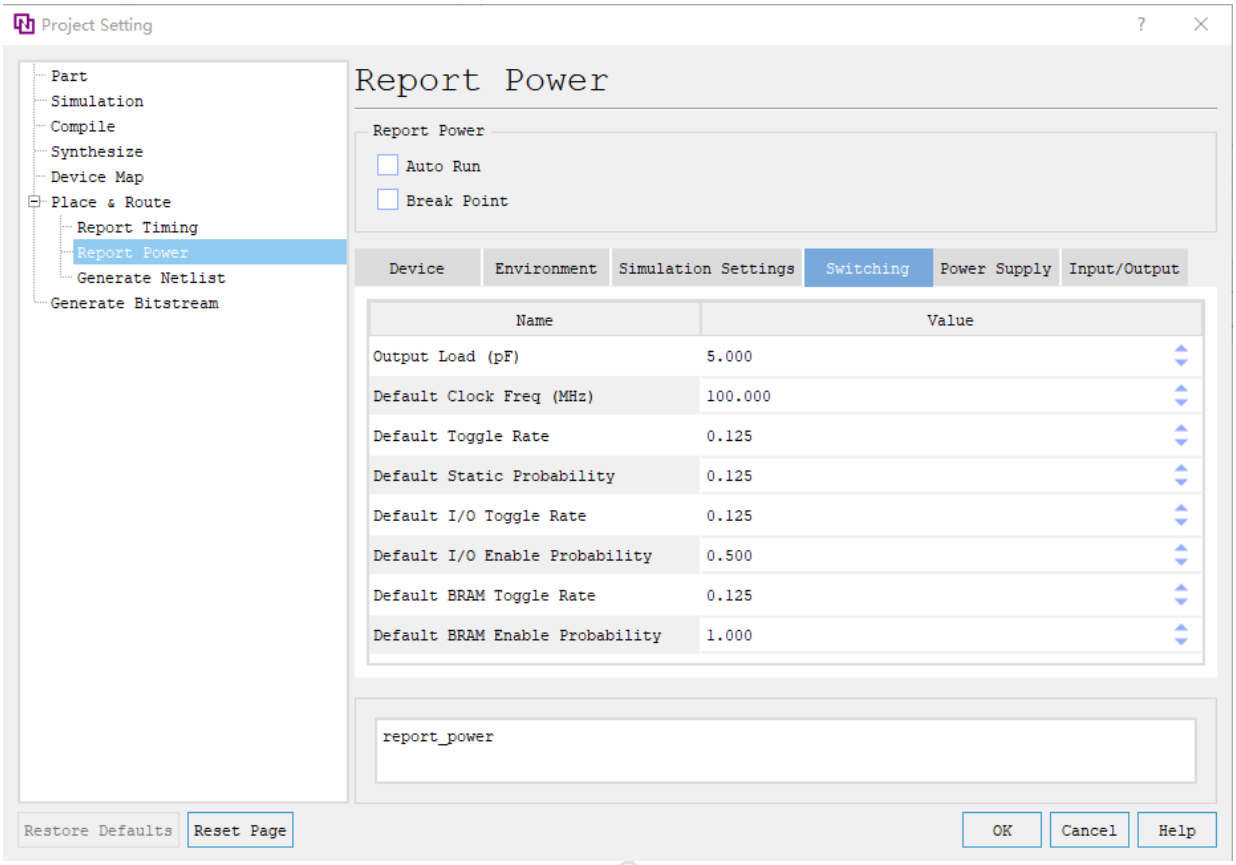


Figure6-37

[Output Load (pF)]: The external load capacitance of IO, in the range of [0~50];

[Default Clock Freq(MHz)]: The system clock frequency, in the range of [0~500];

[Default Toggle Rate]: The logic toggle rate, in the range of [0~1];

[Default Static Probability]: The probability of a static high level, in the range of [0~1];

[Default I/O Toggle Rate]: The toggle rate of I/O, in the range of [0~1];

[Default I/O Enable Probability]: The probability that I/O will be enabled, in the range of [0~1];

[Default BRAM Toggle Rate]: The toggle rate of BRAM, in the range of [0~1];

[Default BRAM Enable Probability]: The probability that BRAM will be enabled, in the range of [0~1];

6.7.5 [Power Supply] Page

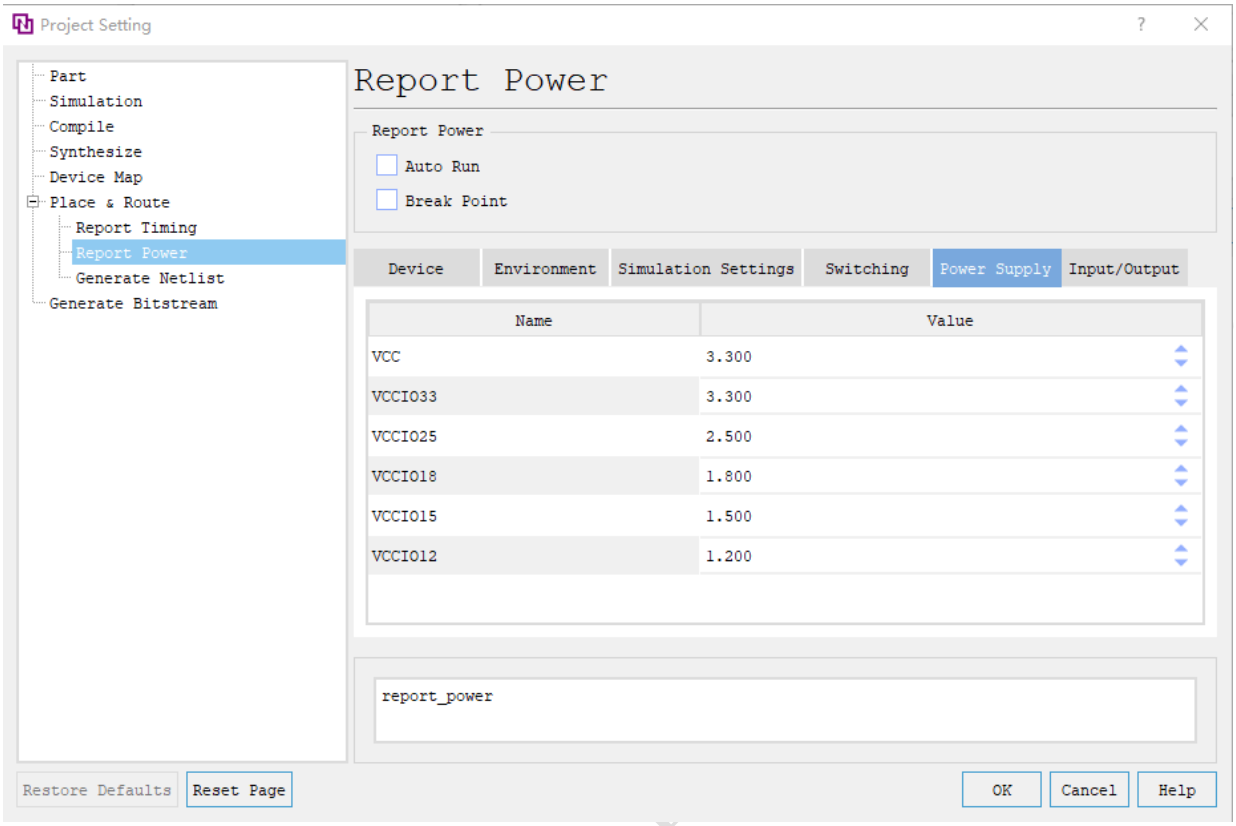


Figure6-38

[VCC]: Core power supply voltage, with a voltage range of [0.99-1.21] for L-type devices and [2.25-3.64] for D/G-type devices;

[VCCIO33]: When the bank power supply is 3.3V, the voltage can be set within the range of [2.97-3.63];

[VCCIO25]: When the bank power supply is 2.5V, the voltage can be set within the range of [2.25-2.75];

[VCCIO18]: When the bank power supply is 1.8V, the voltage can be set within the range of [1.62-1.98];

[VCCIO15]: When the bank power supply is 1.5V, the voltage can be set within the range of [1.35-1.65];

[VCCIO12]: When the bank power supply is 1.2V, the voltage can be set within the range of [1.08-1.32].

6.7.6 [Input/Output] Page

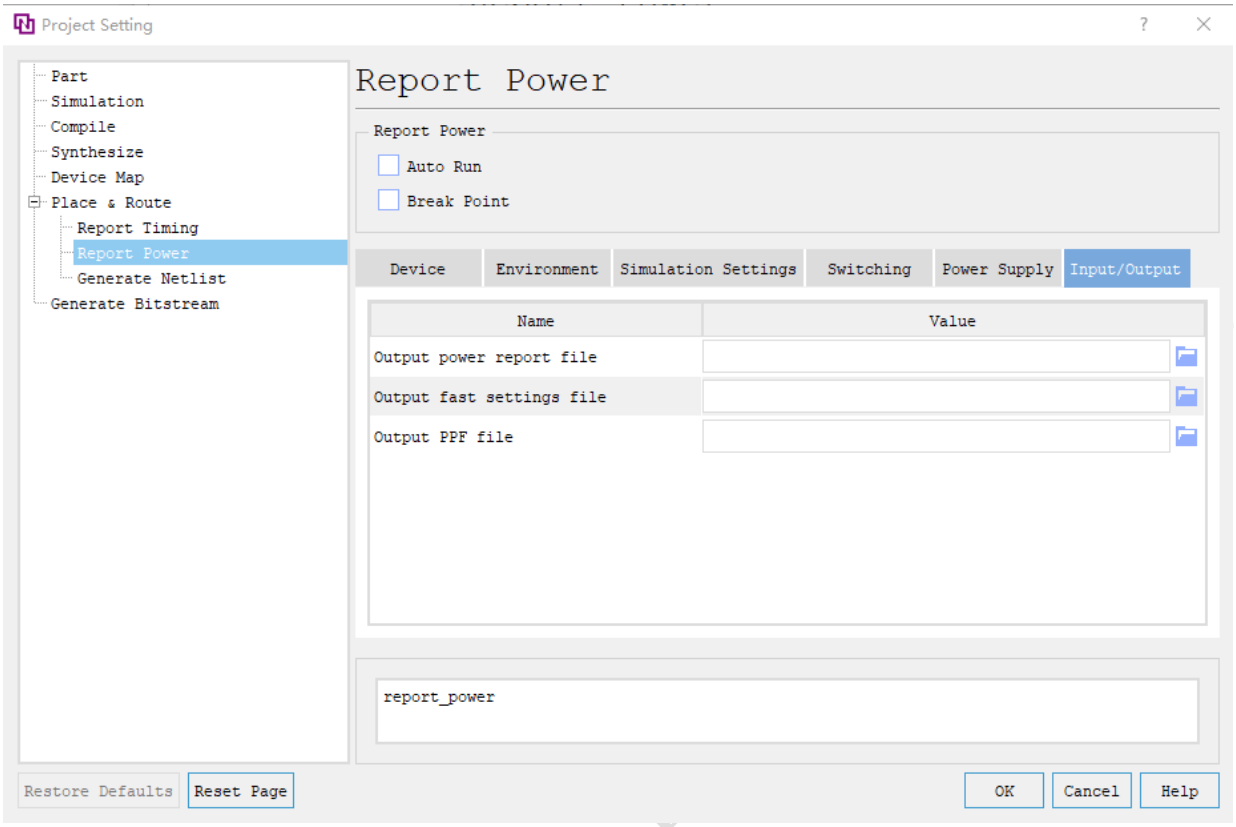


Figure6-39

[Output power report file]: Outputs the power consumption report file;

[Output fast settings file]: Outputs the power consumption report settings file for the project;

[Output PPF file]: Outputs the PPP power consumption evaluation file.

6.7.7 Power Consumption Report

[Double-click Report Power to generate the power consumption report for the design, as shown in the figure below.

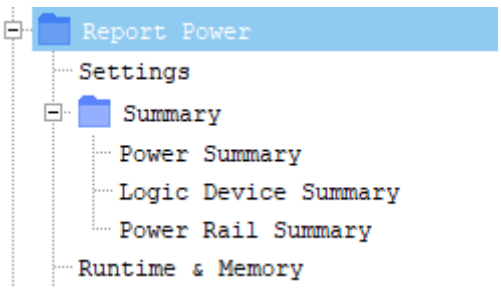


Figure6-40

The power consumption information includes:

1. Power Summary

The Power Summary reports the overall power consumption and junction temperature of the design

under the current settings, mainly including the total on-die power consumption, static power consumption, and chip junction temperature.

2. Logic Device Summary

The Logic Device Summary reports the dynamic power consumption of each main resource according to the classification of resource usage in the design, as well as their percentage in the total power consumption.

3. Power Rail Summary

The Power Rail Summary reports the power consumption for each power supply within the chip, mainly including power supply name, voltage value (in V), current value (in A), external current value (in A), and on-die power consumption (in W).

6.8 Generate Netlist

After Generate Netlist is run, the post-placement and routing simulation netlist file _sim.v and the timing back-annotation file .sdf are produced. The post-simulation library directory is "..\arch\vendor\pango\verilog\bsim".

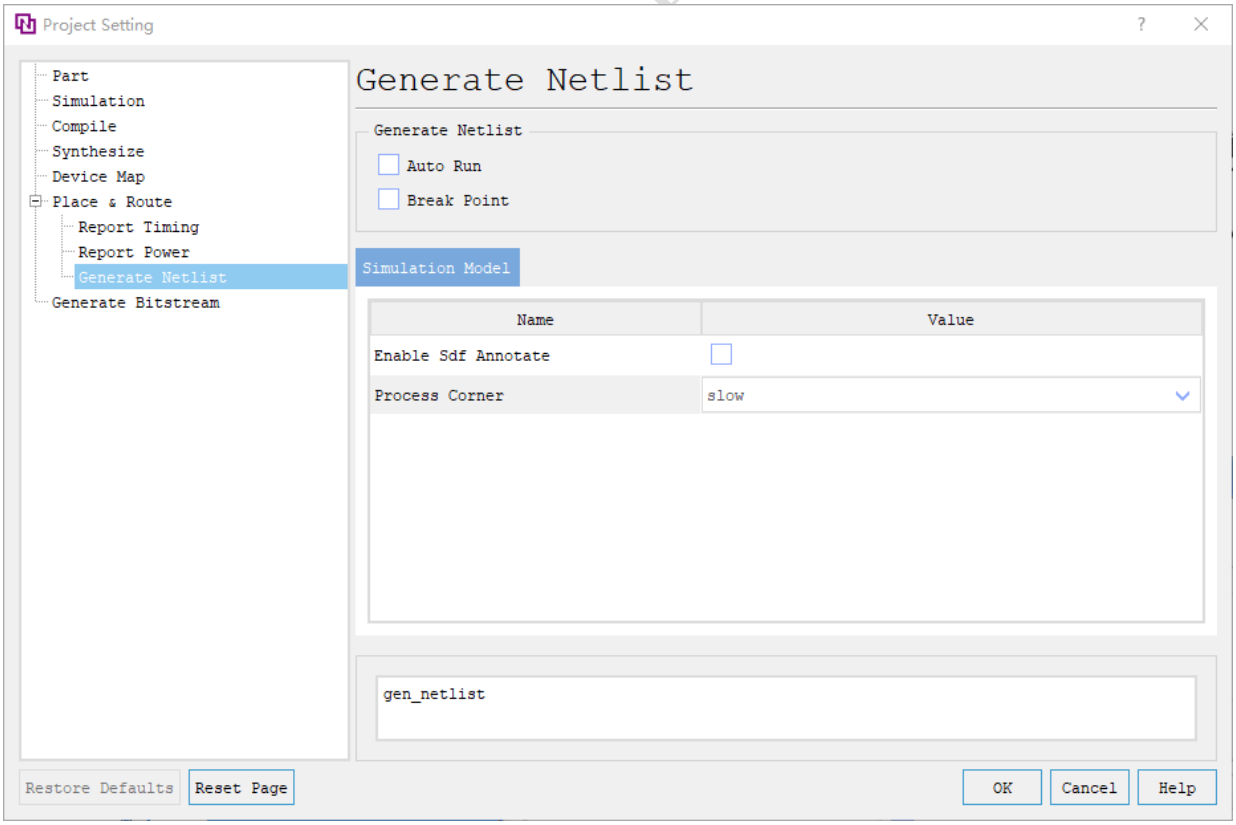


Figure6-41

[Enable Sdf Annotate]: Selects whether to generate an sdf_annotate task declaration in the simulation netlist (sim.v);

[Process Corner]: Specifies the process corner (slow or fast) corresponding to the .sdf delay file.

Chapter 7 Simulation

PDS supports the use of third-party simulation tools such as ModelSim or QuestaSim for simulation, which mainly includes compiling simulation libraries and launching simulation runs. Compiling simulation libraries refers to using ModelSim or QuestaSim to compile libraries for simulation; launching simulation runs refers to using ModelSim or QuestaSim to compile and simulate the user's design. The simulation is divided into different stages such as Run Behavioural Simulation, Run Post-Synthesis Simulation, Run Post-Place and Route (PnR) Functional Simulation, and Run Post-PnR Timing Simulation.

7.1 Simulation Settings

Right-click in the [Navigator] area, and select [Project Setting > Simulation] to open the Simulation Settings interface, as shown in the figure below.

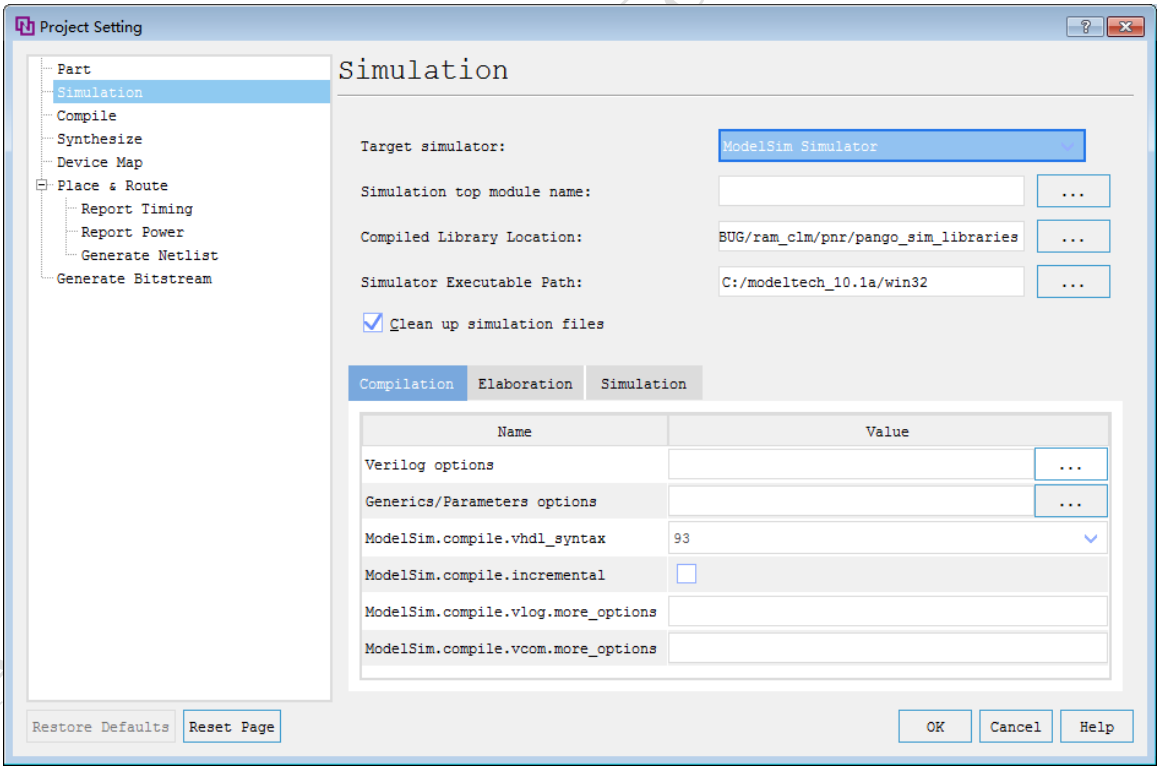


Figure7-1

Target simulator: Currently two third-party simulation tools, ModelSim Simulator and QuestaSim Simulator, are supported;

Simulation top module name: Select the top module for simulation, or right-click the simulation file to select;

Compile Library Location: The location of the compiled simulation library, which will be consistent with the location selected during library compilation;

Simulator Executable Path: Sets the simulation tool path;

Clean up simulation files: Whether to clear the existing files in the simulation folder before running the simulation. If this option is not checked, files with the same name will be replaced during the simulation run.

Additionally, there are 3 sub-pages in the Simulation Settings interface.

7.1.1 [Compilation] Page

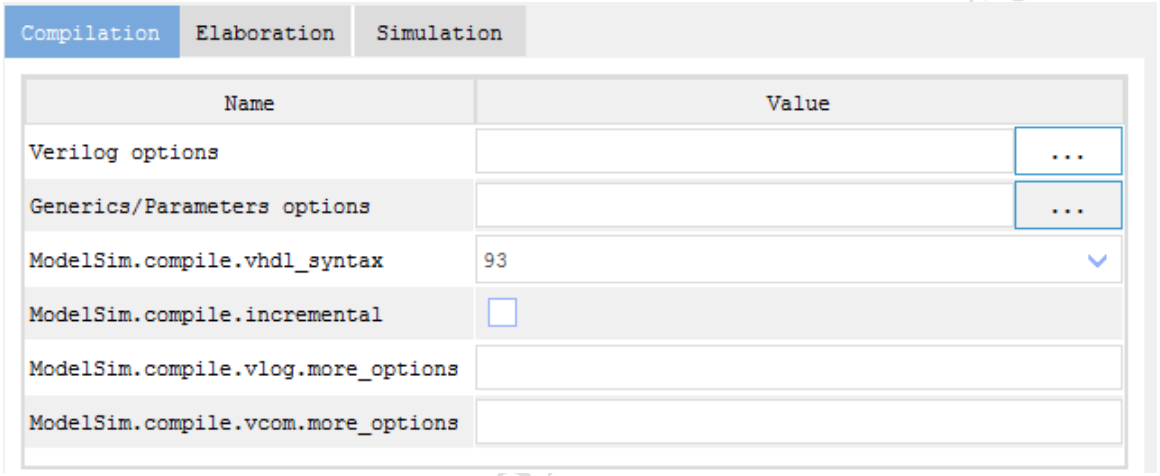


Figure7-2

[Verilog options]: Sets the Verilog include path and defines macros. This option corresponds to the +incdir+ and +define+ options of the vlog command for ModelSim or QuestaSim;

[Generics/Parameters options]: Sets the values for generics/parameters. Corresponds to the -g option of the vsim command;

[ModelSim.compile.vhdl_syntax]: Sets the supported vhdl syntax. Corresponds to the -87|-93|-2002|-2008 options of the vcom command;

[ModelSim.compile.incremental]: Sets whether ModelSim compilation is incremental. Corresponds to the -incr option of the vlog command;

[ModelSim.compile.vlog.more_options]: Whether to add more options to the vlog command, such as -nodebug, -64; multiple options must be separated by spaces;

[ModelSim.compile.vcom.more_options]: Whether to add more options to the vcom command, such as -refresh, -64; multiple options must be separated by spaces.

7.1.2 [Elaboration] Page

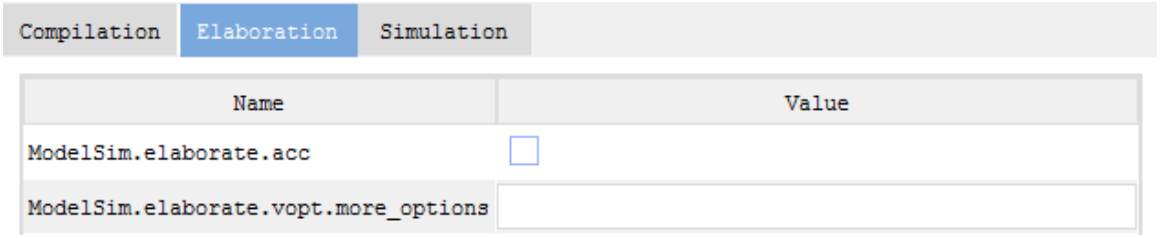


Figure7-3

[ModelSim.elaborate.acc]: If this option is selected, the simulation Objects can be accessed during vopt optimization to prevent them from being optimized. Corresponds to the -voptargs="+acc" option of the vsim command;

[ModelSim.elaborate.vopt.more_options]: More options for the vopt command, such as -64.

7.1.3 [Simulation] Page

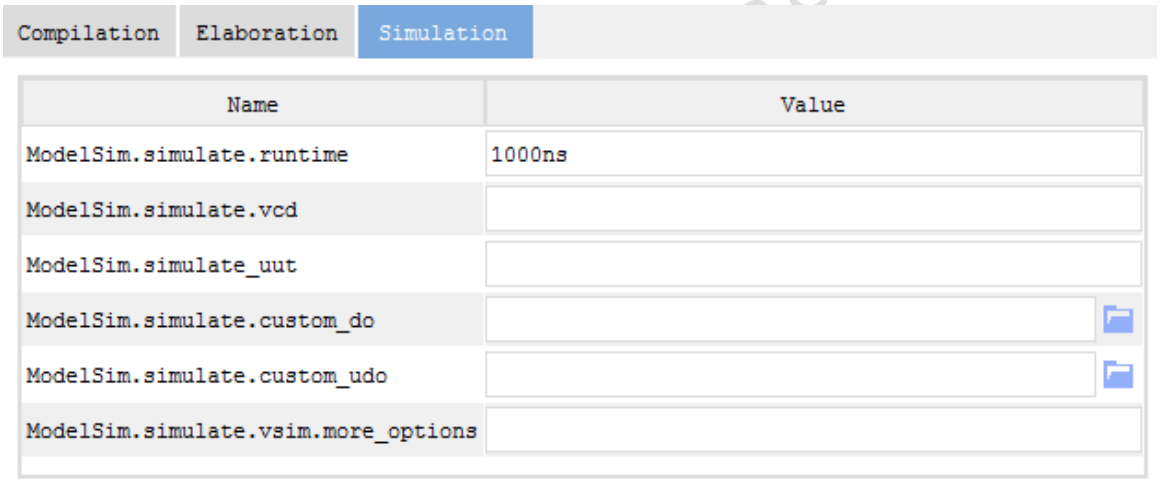


Figure7-4

[ModelSim.simulate.runtime]: Sets the simulation time, corresponding to ModelSim's run xxxns;

[ModelSim.simulate.vcd]: Specifies the name of the vcd file generated for power analysis;

[ModelSim.simulate.uut]: All signal changes under the uut (unit under test) will be recorded in the vcd file;

[ModelSim.simulate.custom_do]: The user's own script is utilized for compilation, which will be executed using ModelSim's do command, and the compile.tcl file will not be generated;

[ModelSim.simulate.custom_udo]: The user's own script is utilized for compilation, which will be executed during simulation through ModelSim's do command;

[ModelSim.simulate.vsim.more_options]: More options for vsim, such as -sdfmax, loading the max value for sdf.

7.2 Compile Library

In the toolbar, select [Tools > Compile Simulation Libraries], as shown in the figure below.

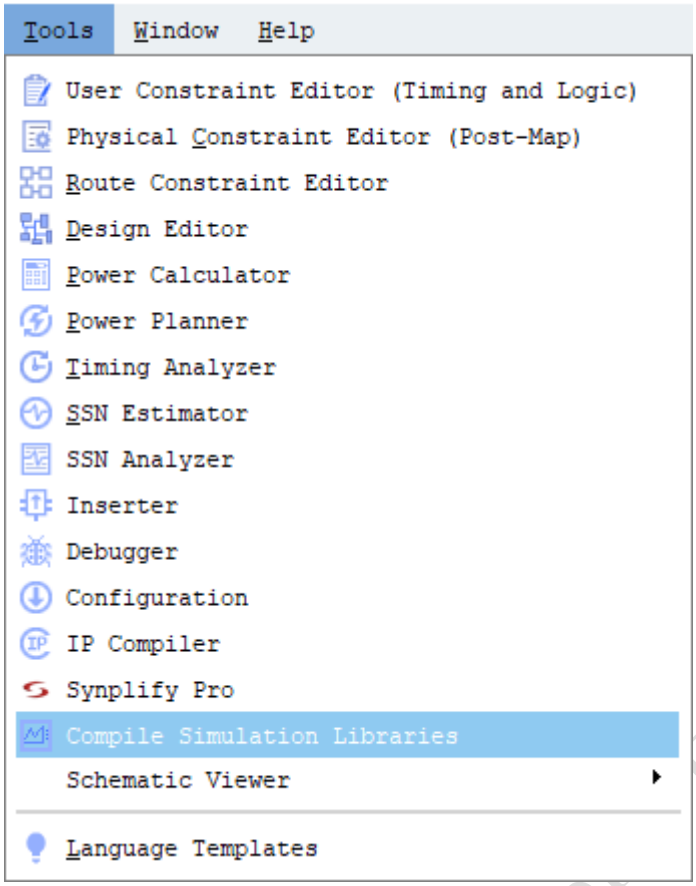


Figure7-5

Enter the path to the executable file of the third-party simulation tool in the [Simulator Executable Path] field, and then click [Compile] to start compiling the library. This is shown in the following figure.

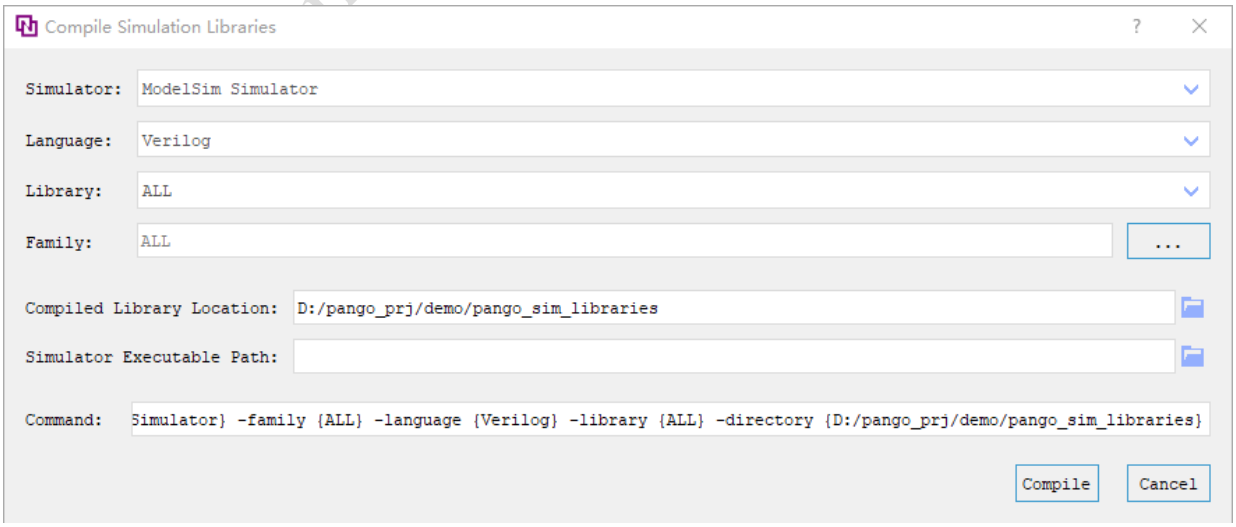


Figure7-6

[Simulator]: Selects the simulation tool. Available options are ModelSim and QuestaSim;

[Language]: Simulation language. The available option is Verilog;

[Library]: Type of simulation library. Available options are USIM library for pre-simulation (before pnr) and VSIM library for post-simulation (after pnr). Selecting "[ALL] will generate both VSIM and USIM libraries;

[Family]: Sets the family of the device.

[Compile Library Location]: The address of the generated simulation library files, which are saved in the current project directory by default;

[Simulator Executable Path]: The path to the executable file of the third-party simulation tool;

[Command]: The automatically generated tcl command.

Upon completion of the library compilation, the message "Compile libraries succeed." will be displayed.

7.3 Start Simulation

First, add the tb file for simulation.

Right-click [Simulation] under [Navigator], select [Add Source] from the context menu, and then follow the prompts to add the tb file. This is shown in the following figure.

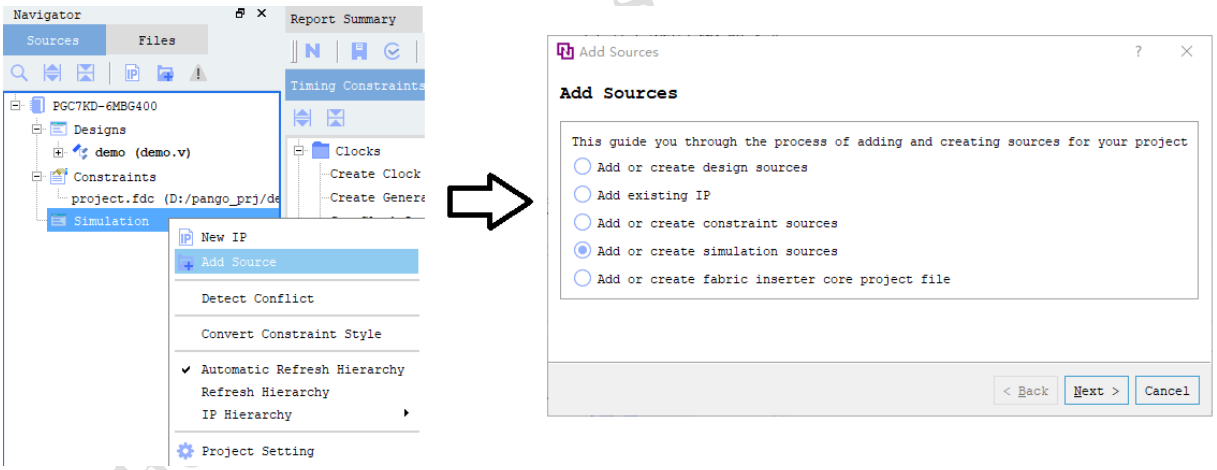


Figure7-7

After adding the tb file, right-click it to execute the simulation. This is shown in the following figure.

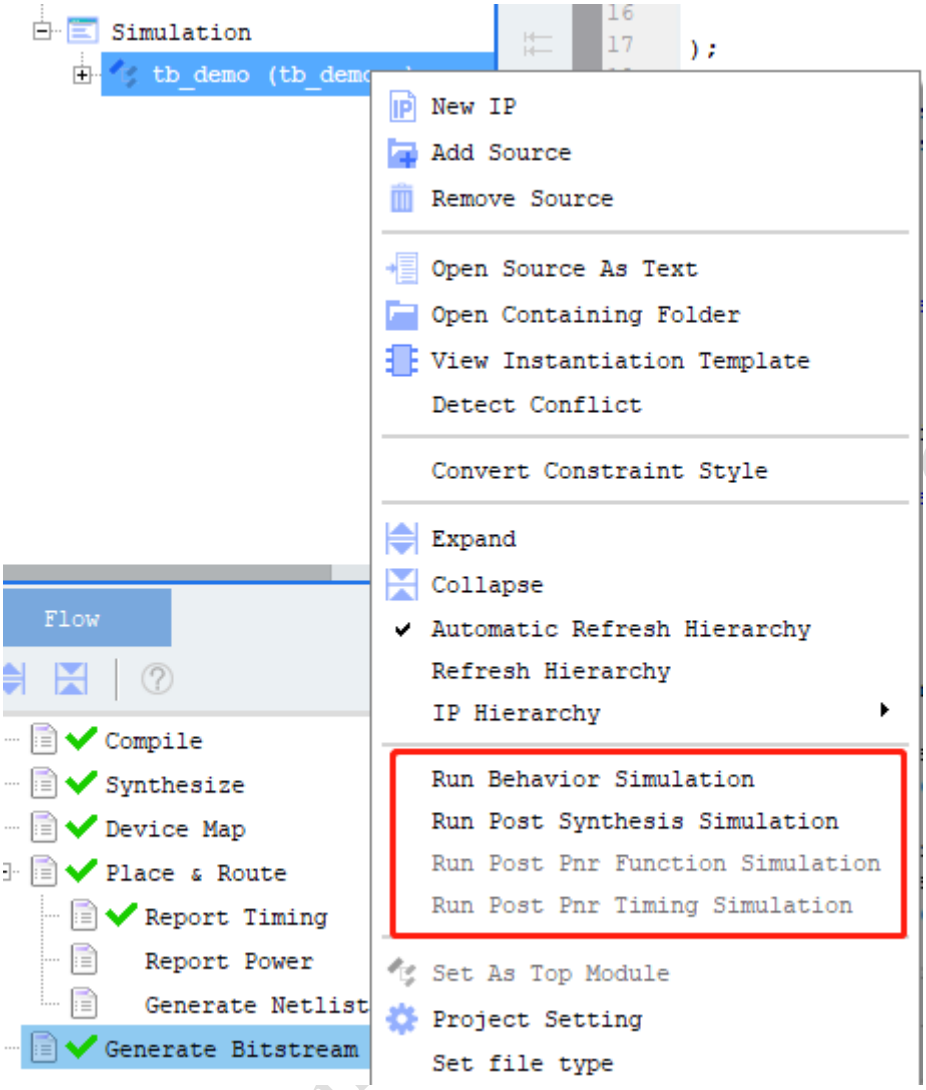


Figure7-8

When the project has not been synthesized, only behavioral simulation can be performed using "Run Behaviour Simulation";

After synthesis is complete, post-synthesis functional simulation can be performed using [Run Post Synthesis Simulation], but at this point [Run Post Pnr Function Simulation] and [Run Post Pnr Timing Simulation] are still grayed out and only after the [Generate Netlist] operation has been completed can these two simulations be run.

Different simulation functions generate different scripts. The generated scripts are saved in the "sim" folder under the project directory. The "sim" directory contains 4 folders corresponding to 4 different stages of simulation, each containing simulation scripts and other files.

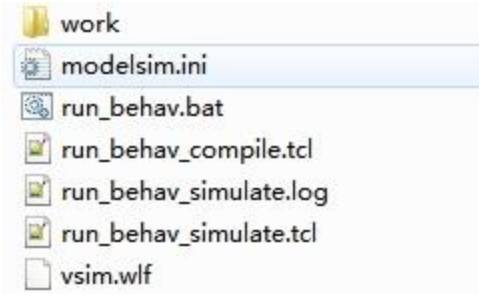


Figure7-9

Among them, *compile.tcl is the compilation command script, and *simulate.tcl is the simulation execution script. Double-clicking the .bat file in the directory can directly launch the simulation.

Chapter 8 Download

8.1 Basic Operation

First, connect one end of the USB Cable to the PC and the other end to the board with the PGC chip, and then power up the board.

After the PDS generates the bitstream file, users can download the file to CRAM or embedded Flash using the Fabric Configuration download tool via USB Cable.

8.1.1 Launch Download Tool

There are mainly two ways to open the Fabric Configuration download tool:

- Method 1:

In the PDS interface, click [Tools > Configuration], as shown in the figure below.

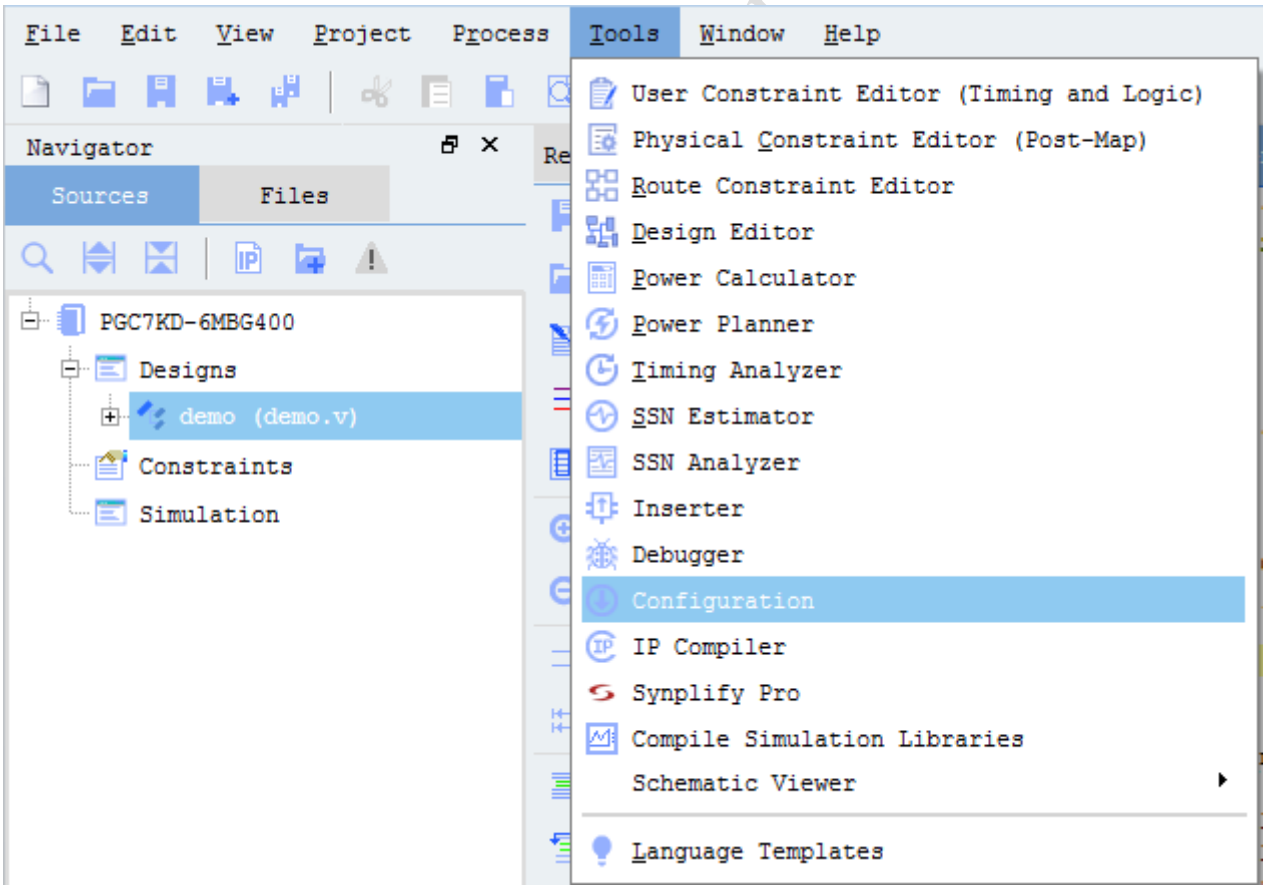


Figure8-1

Or click the icon in the toolbar.

Figure8-2

Note that the Fabric Configuration download tool can be opened through PDS only within an existing project; otherwise, the icon for Fabric Configuration will be grayed out and unclickable.

- Method 2:

Click [Start > All Programs > pango > Pango Design Suite xxxx > Accessories > Configuration].

The Fabric Configuration tool interface is shown in the figure below.

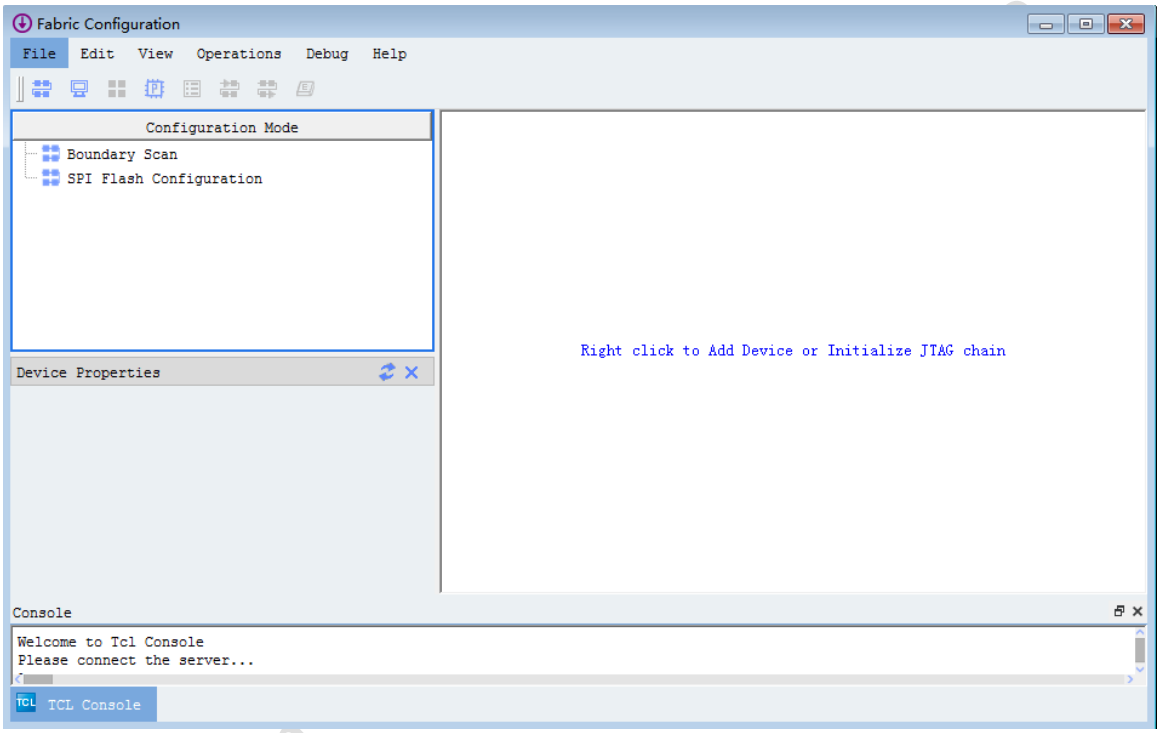


Figure8-3

8.1.2 Connect to Server

The Fabric Configuration tool is used in conjunction with the corresponding Jtag Server software, and their versions must be consistent; otherwise, it will not work properly.

Click in the toolbar, or select [File > Connect To Server] to open the setup wizard, as shown in the figure below.

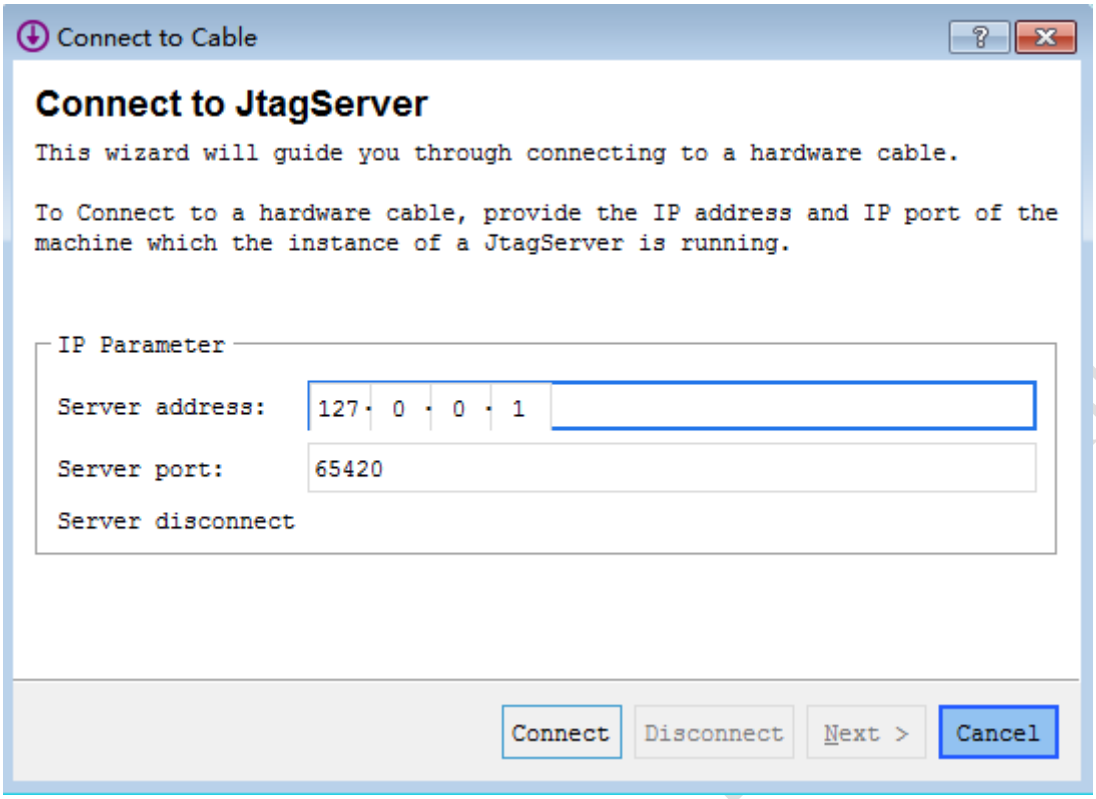


Figure8-4

This software uses the TCP/IP protocol to communicate with the server, so it is necessary to specify an IP address and port. The IP address can be set as the local IP address or a remote IP address. If the local IP address is specified, which is the same as the default address, just click [Connect]. When a remote IP address is specified, find the cdt_js software in the bin directory of the installation path, and then double-click it, so that the Jtag Server will be started remotely.

Click [Connect], and if the download cable has already connected the PC with the board fitted with a PGC chip (and the board is powered on), proceed with the settings, as shown in the figure below.

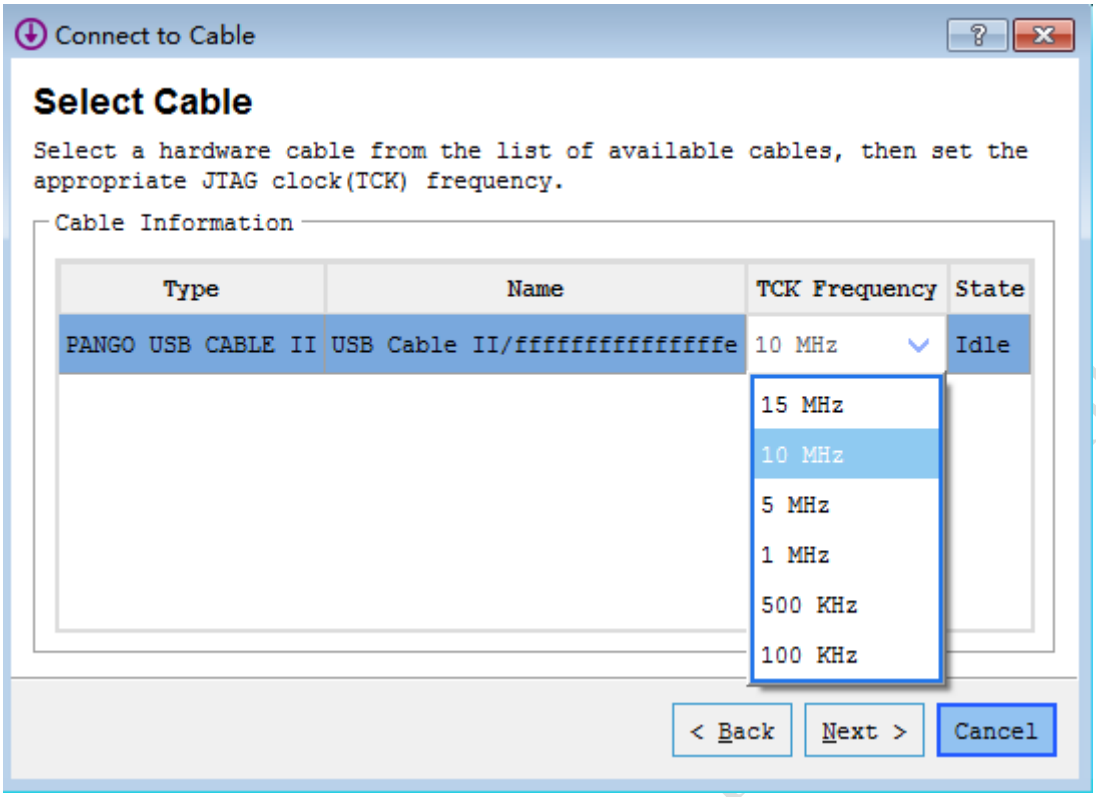


Figure8-5

At this point, users can set the speed of the download cable, i.e., the TCK frequency of the JTAG interface, which defaults to 10MHz and can be set up to 15MHz. Click [Next], and then click [Finish] to complete the settings for server connection.

8.1.3 Scan Device

Right-clicking in the blank area on the right side of the [Fabric Configuration] interface will bring up a context menu, or users can click on the in the toolbar, and the software will automatically connect to the server and identify devices on the JTAG chain. Once devices are successfully identified, the type of devices will be displayed on the right side of the interface, and the [Device Properties] on the left will display information such as the Device ID. A pop-up window [Assign New Configuration File] will appear to load .sbit or .sfc file. This is shown in the following figure.

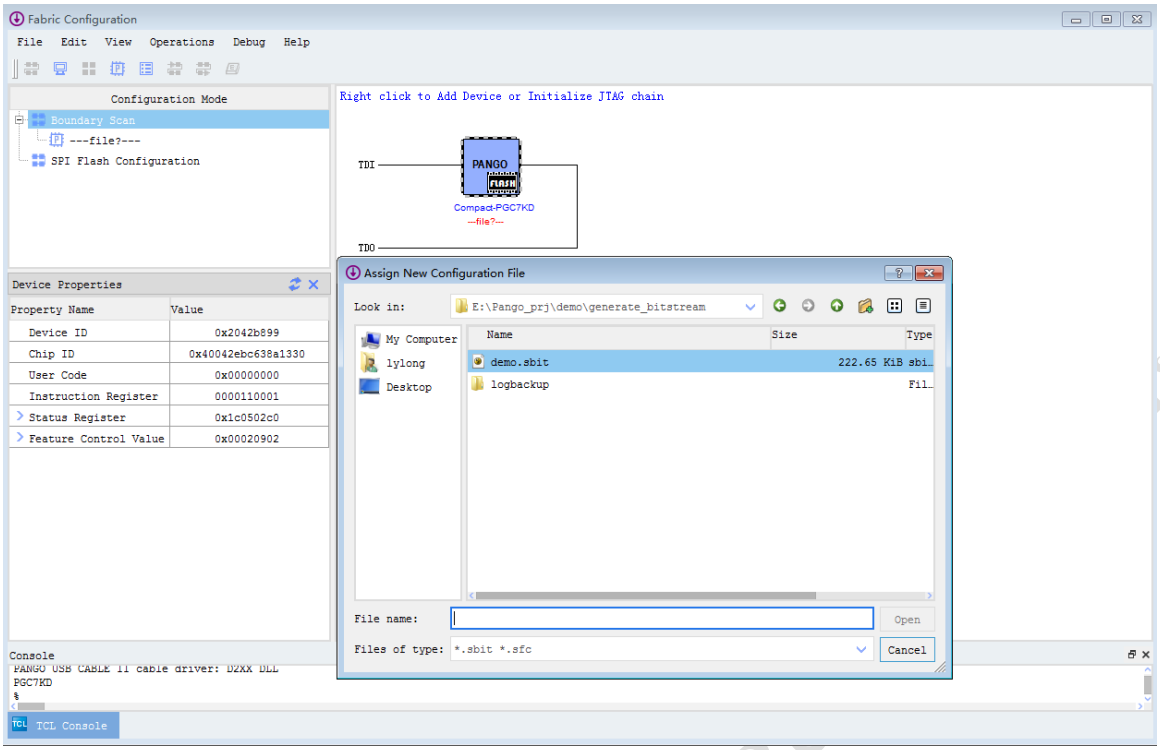


Figure8-6

[Device Properties] description:

[Device ID]: A 32-bit identity code for each type of device;

[Chip ID]: A unique 64-bit identity code for each chip;

[User Code]: A user-defined 32-bit code stored in the bitstream, which is 0x00000000 when the chip is not configured;

[Instruction Register]: The state of the instruction register;

[Status Register]: The status register, indicating the operational state of the device;

[Feature Control Value]: The feature control bits, indicating the current feature control bits of the device.

8.1.4 Download

After assigning the bitstream file, right-click the device icon to perform a family of operations such as "Program...".

- Operations for embedded Flash

Right-click the "FLASH" icon, and a menu will pop up. All the options in the menu are operations for the embedded Flash. This is shown in the following figure.

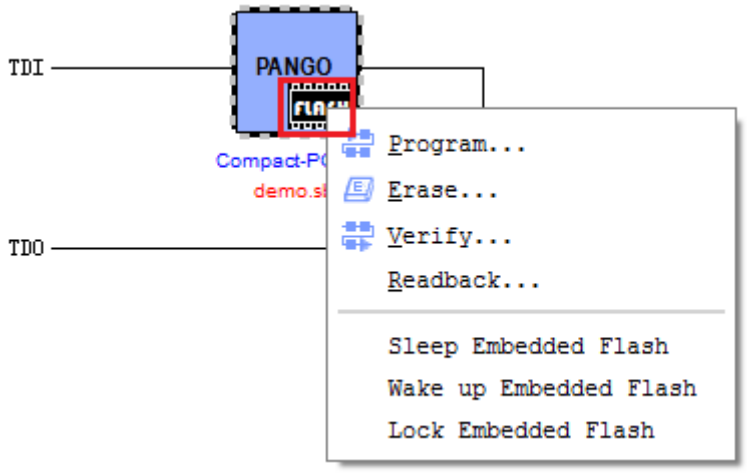


Figure8-7

[Program]: Downloads the bitstream into the embedded Flash. If the user wants the logic functions to take effect immediately after they are downloaded to the embedded Flash, Master Self Configuration must be selected, as described in 5.1 Boot Select;

[Erase]: Erases all the user data on the embedded Flash;

[Verify]: Reads back data from the embedded Flash and compares it with the data in the bitstream file;

[Readback]: Reads back data from the embedded Flash and saves it in a custom .sfc file;

[Sleep Embedded Flash]: Puts the embedded Flash into sleep mode. During sleep mode, operations such as downloading data or reading back data from the Flash cannot be performed;

[Wake up Embedded Flash]: Wakes up the embedded Flash;

[Lock Embedded Flash]: Locks the embedded Flash. Once it is locked, reading back data from Flash is not allowed. To unlock the Flash, users need to perform an [Erase] operation.

- Operations for CRAM

Right-click the area of the device icon other than "FLASH", and a menu for operations on CRAM will pop up, as shown in the figure below.

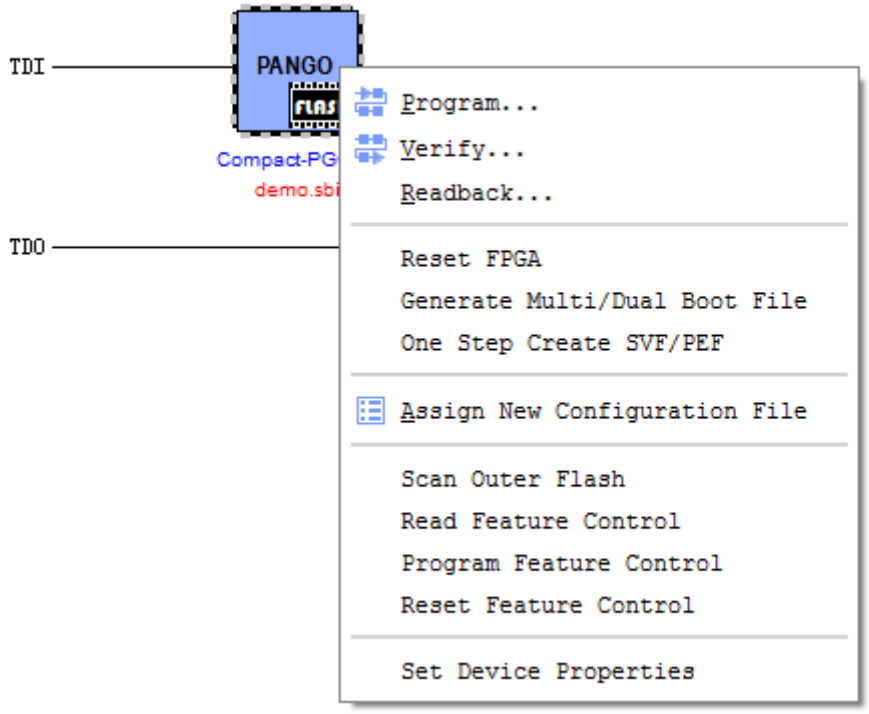


Figure8-8

[Program]: Downloads the bitstream into the chip's CRAM;

[Verify]: Reads back the data from CRAM and compares it with the data in the bitstream file;

[Readback]: Reads back the data from CRAM and saves it in a custom .sbit file;

[Reset FPGA]: Sends a reset command to reset the chip;

[Generate Multi/Dual Boot File]: Generates a multi boot or dual boot bitstream file; PGC devices only support the dual boot bitstream file;

[One Step Create SVF/PEF]: Generates .svf and .pef files with just one click;

[Assign New Configuration File]: Assigns a new bitstream file; .sbit and .sfc files are supported;

[Scan Outer Flash]: Scans external SPI Flash;

[Read Feature Control]: Reads feature control bits;

[Program Feature Control]: Programs feature control bits;

[Reset Feature Control]: Resets feature control bits. After they are reset, the value will be 0;

[Set Device Properties]: Sets download properties.

The following sections will provide a detailed introduction to [Generate Multi/Dual Boot File] and [One Step Create SVF/PEF].

8.2 Bitstream File

The formats of the bitstream file are as follows:

Table 8-1

Configuration File Extension	Description
.sbit	Binary configuration data that contains header information (bitstream name, date, etc.). The download tool will recognize the header information but will not write it into the configuration registers of PGC devices. The Fabric Configuration tool can directly program the .sbit file into PGC devices via USB Cable.
.bin	Binary configuration data without header information
.sfc	The bitstream file written into Flash, which is converted from the .sbit file.
.svf	A universal download file in text format, containing operation commands and data. Suitable for all Pango devices, which can be used for configuration, erasure, and verification of embedded Flash, or configuration of CRAM
.pef	A compressed version of the .svf file, in binary format. Suitable for all Pango devices, which can be used for configuration, erasure, and verification of embedded Flash, or configuration of CRAM
.smsk	A binary file, which is mainly used for the readback verification process. It records data bits that are masked from verification and are not involved in verification during readback. It is used in conjunction with the .sbit file.

8.3 Generate .sfc File

In the Fabric Configuration interface, select [Operations > Convert File], as shown in the figure below.

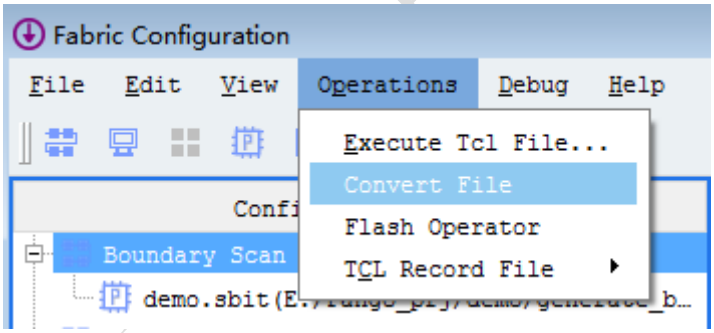


Figure8-9

To generate an .sfc file for downloading to embedded Flash, keep the default settings for the "Device" area in the "Convert File Dialog" interface; the output name of the .sfc file can be customized. To generate an .sfc file for external SPI Flash, select the corresponding "Factory Name" and "Device Name" as well as "Flash Read Mode" according to the specific model of Flash. This is shown in the following figure.

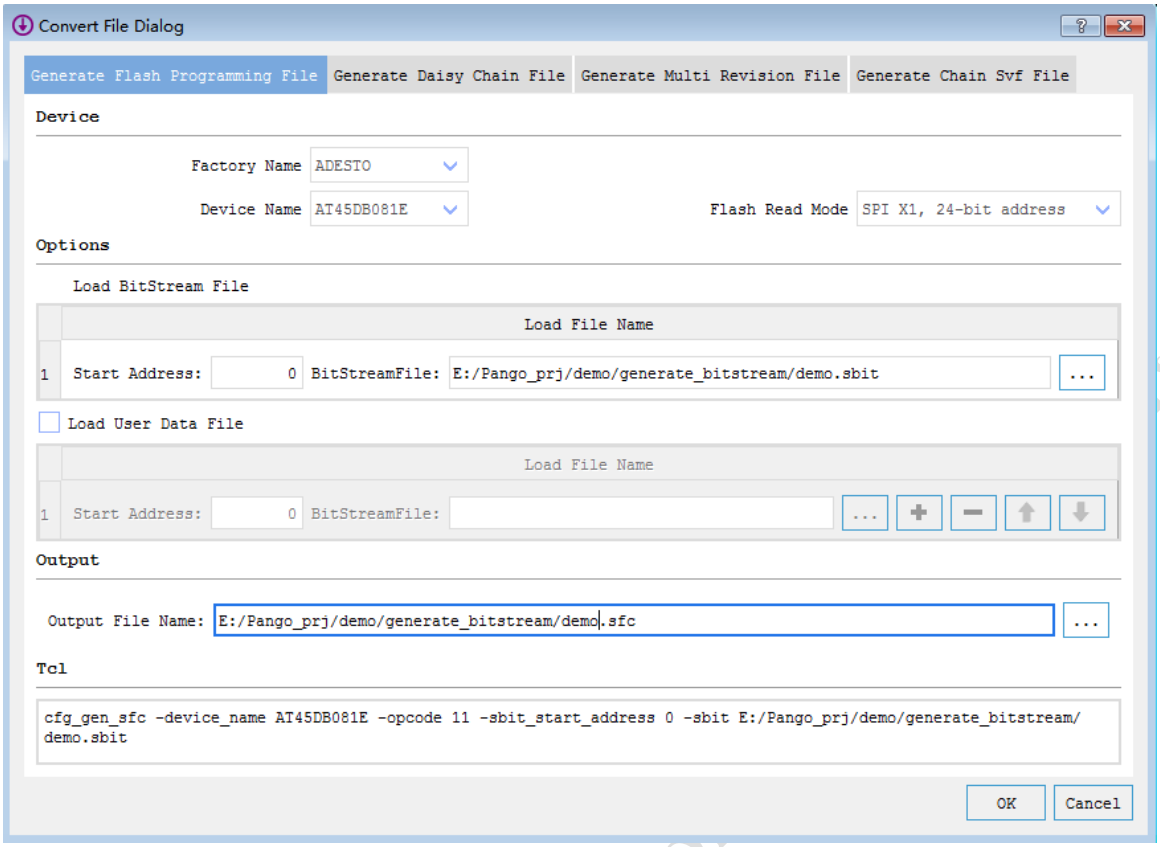


Figure8-10

Click [OK] to generate the .sfc file.

8.4 Download External SPI Flash

First, connect the 4 pins of the main SPI of the USB Cable to the corresponding pins of the PGC chip;

In the [Fabric Configuration] interface, select [SPI Flash Configuration], and then click to scan for external SPI Flash and a window will pop up to allocate the bitstream file. This is shown in the following figure.

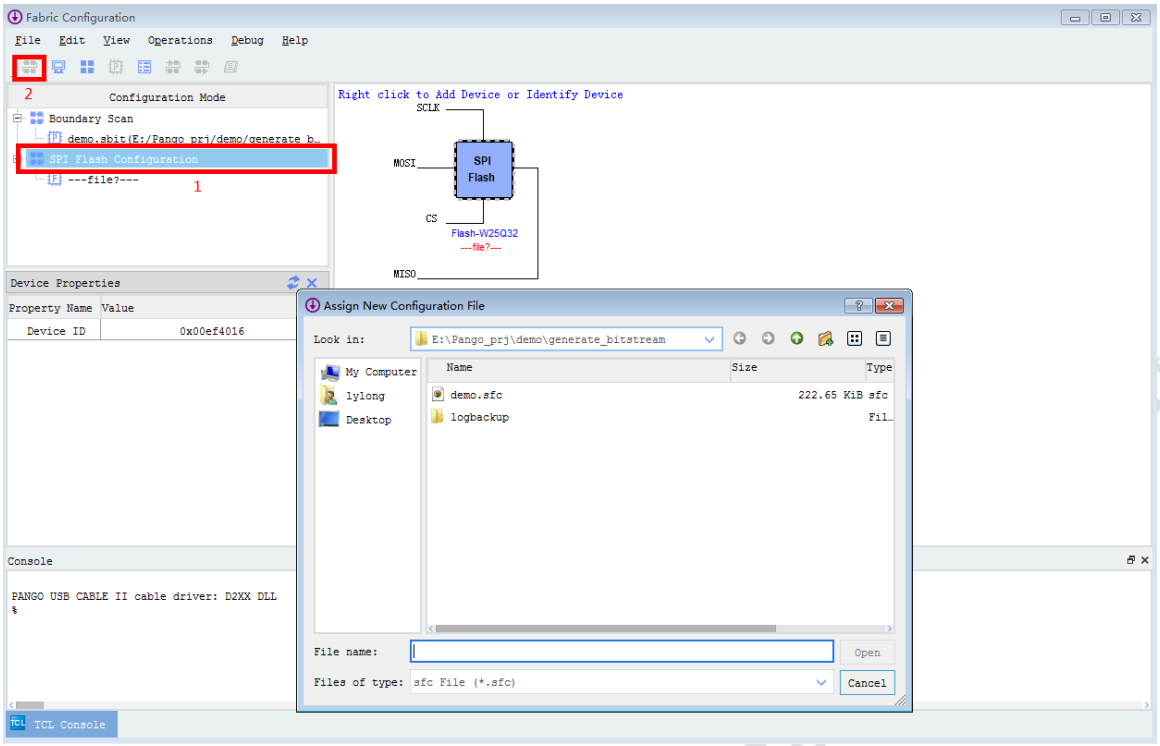


Figure8-11

Right-click the "SPI Flash" icon and select [Program] from the popup menu to complete the download, as shown in the figure below.

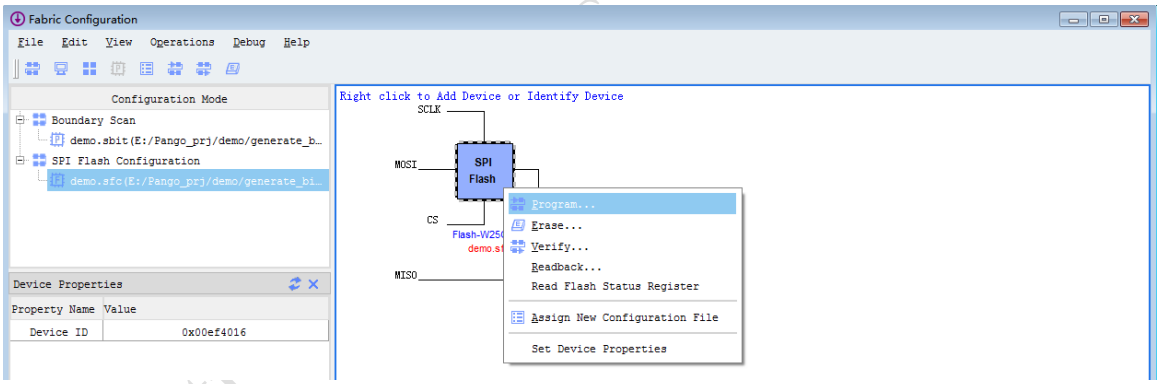


Figure8-12

8.5 Dual Boot

All PGC devices support regular dual boot, while only D-type devices and PGC4KL/4KLS support Master Self Configuration Dual Boot.

8.5.1 Regular Dual Boot

In regular dual boot applications, two bitstreams are stored in the embedded Flash and external SPI Flash, respectively. Based on feature control bit settings, the application bitstream is prioritized to

load from embedded Flash or SPI Flash. If an error occurs during the loading, version fallback will be triggered, and the golden bitstream will be loaded from the alternate storage location.

During device loading, the following errors will trigger version fallback:

- Incorrect device ID
- CRC error
- Watchdog timeout

To determine if the currently loaded bitstream has errors, read the status register's timeout, crc_err, and id_err. After fallback, the error signals in the status register are cleared, and the fallback in the status register indicates whether the current bitstream is the fallback bitstream.

The operation steps for regular dual boot are as follows:

1. Enable the watchdog

If regular dual boot is selected, the watchdog needs to be enabled for the application bitstream (to detect if the configuration process times out. For each 512 system clock cycles consumed, the watchdog count decreases by 1; for the PGC10KD system clock at 15.647MHz, the watchdog count decreases by 1 after 32.7μs; for other devices with a system clock of 20.46MHz, the watchdog count decreases by 1 after about 25μs). The method to enable the watchdog is:

Go to [Project Setting > Generate Bitstream > Configuration > Load Watchdog], set the watchdog count, and select [Enable Watchdog In Configuration Mode], as shown in the figure below:

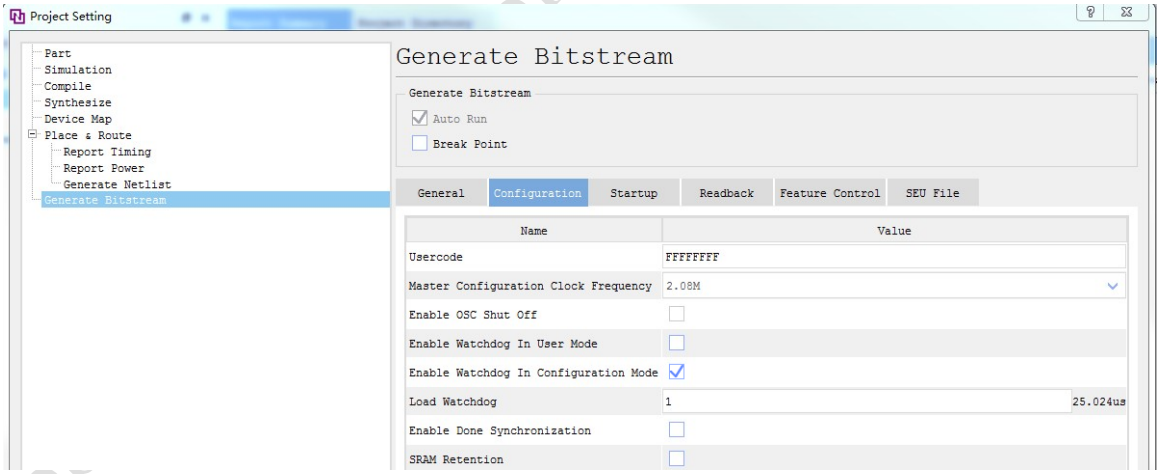


Figure8-13

The watchdog count depends on the time it takes to complete the Master Self Configuration. The watchdog timing should be slightly greater than the Master Self Configuration time. For the Master Self Configuration time of various PGC devices, please refer to the "DS03001_Compa Family CPLDs Datasheet".

2. Setting Feature Control Bits

The feature control bits can be set as follows.

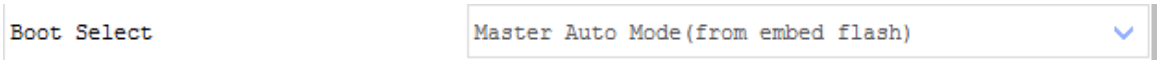


Figure8-14

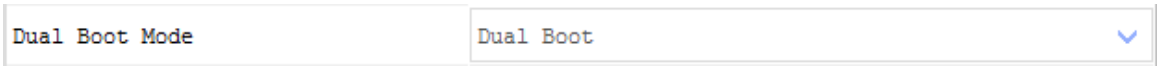


Figure8-15

If [Master SPI Mode] is available for [Boot Select], the golden bitstream should be put in the embedded Flash.

3. Generate Bitstream

The golden bitstream and the application bitstream will be generated separately.

4. Download

Load the bitstream files into the embedded Flash and external SPI Flash, respectively.

8.5.2 Master Self Configuration Dual Boot

In Master Self Configuration Dual Boot applications, both bitstreams are stored in the embedded Flash, with the golden bitstream starting at address 0 and the application bitstream beginning on the next page following the golden bitstream. If an error occurs when the application bitstream is loaded, the device automatically falls back to the golden bitstream at address 0.

The operation steps for Master Self Configuration Dual Boot are as follows:

1. Enable the watchdog

When the Master Self Configuration Dual Boot is used, the watchdog needs to be enabled for both the application bitstream and the golden bitstream. The method to enable the watchdog is:

Go to [Project Setting > Generate Bitstream > Configuration > Load Watchdog], set the watchdog count, and select [Enable Watchdog In Configuration Mode], as shown in the figure below:

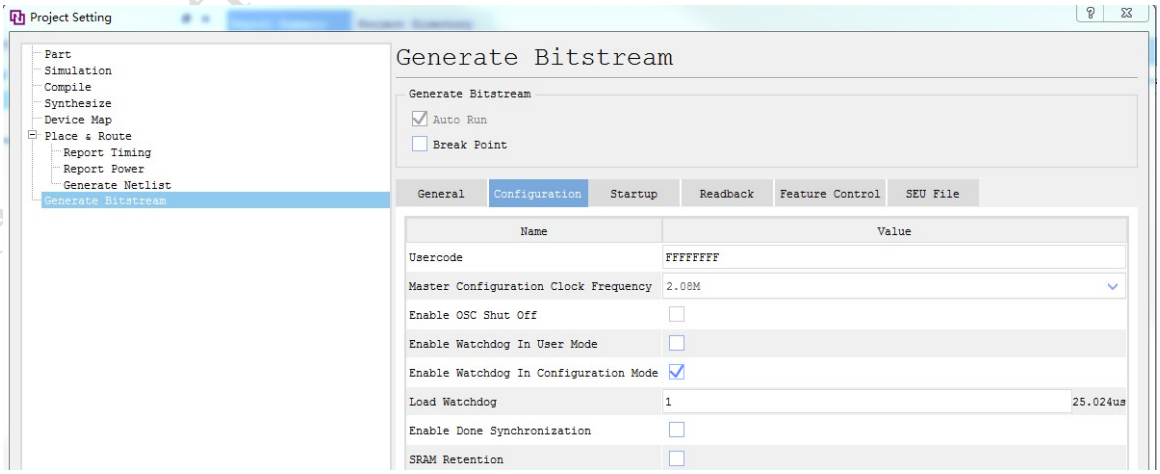


Figure8-16

The watchdog count depends on the time it takes to complete the Master Self Configuration. The

watchdog timing should be slightly greater than the Master Self Configuration time. For the Master Self Configuration time of various PGC devices, please refer to the "DS03001_Compa Family CPLDs Datasheet".

2. Setting Feature Control Bits

The feature control bits can be set as follows.

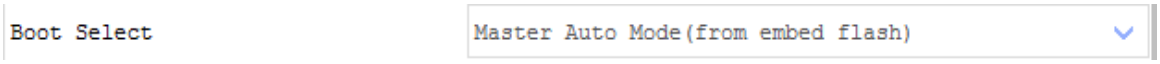


Figure8-17

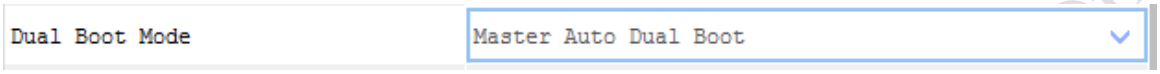


Figure8-18

At this time, [Boot Select] can only be set to [Master Auto Mode].

3. Generate Master Self Configuration Dual Boot bitstreams

Right-click the device icon and select [Generate Multi/Dual Boot File] from the menu, as shown in the figure below.

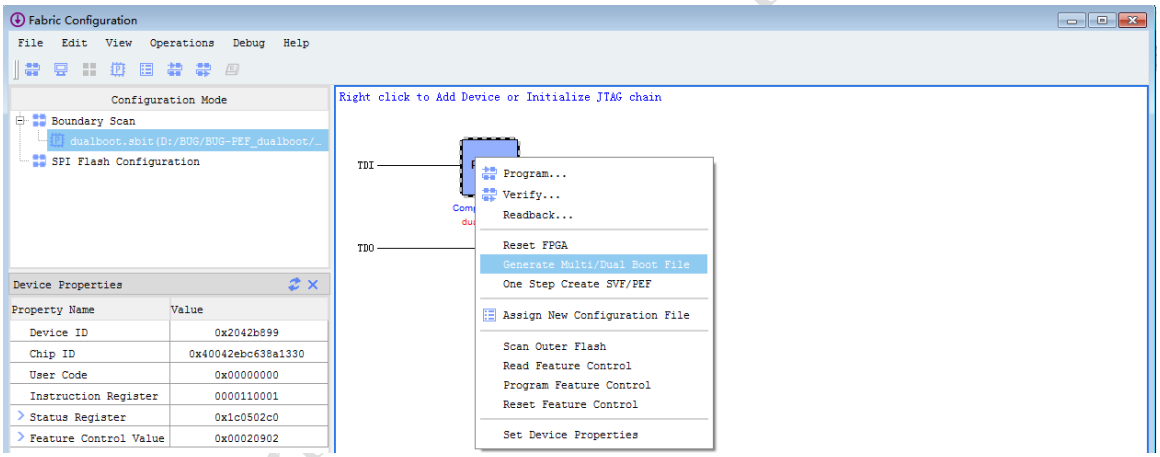


Figure8-19

In the interface for generating Master Self Configuration Dual Boot bitstreams, click [Next];

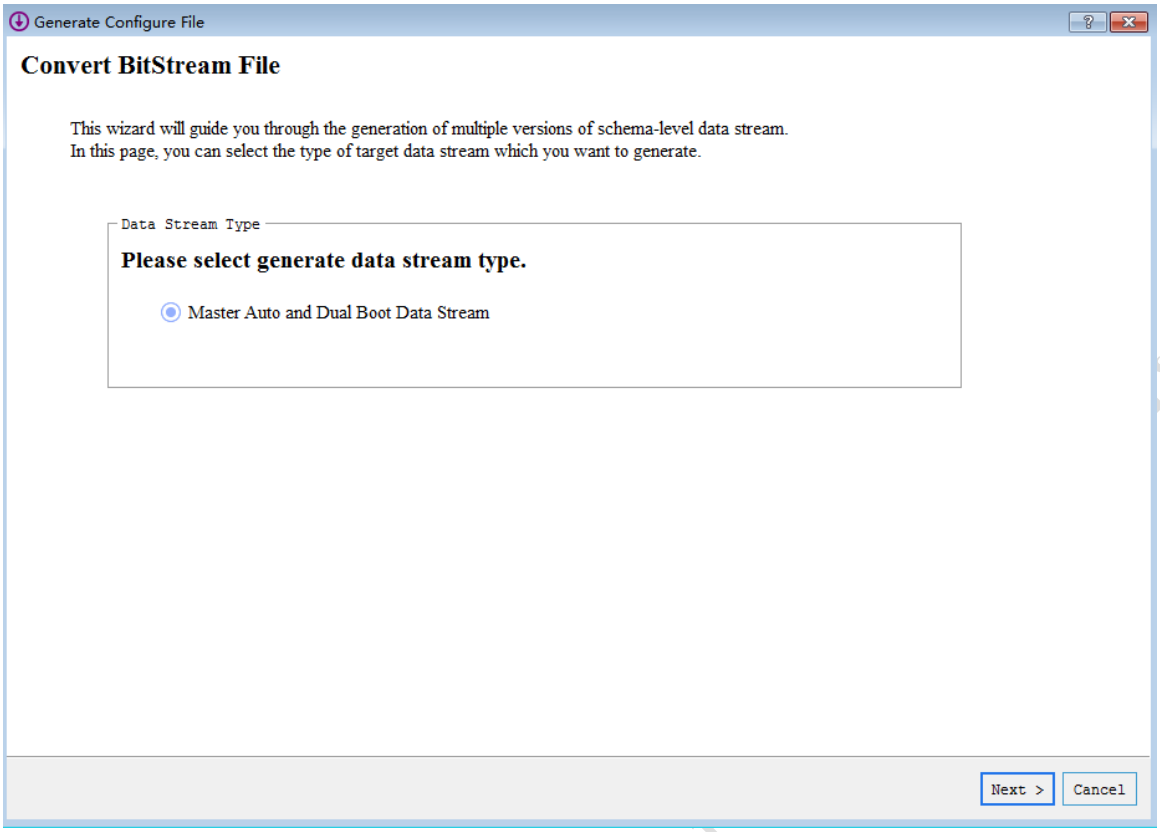


Figure8-20

Specify the golden bitstream file and the application bitstream file under [Golden Bitstream] and [Applied Bitstream], respectively, and the software will automatically calculate their start addresses based on the device size. If [Enable Set Start Address] is selected, users can customize the start address of the bitstream. This is shown in the following figure.

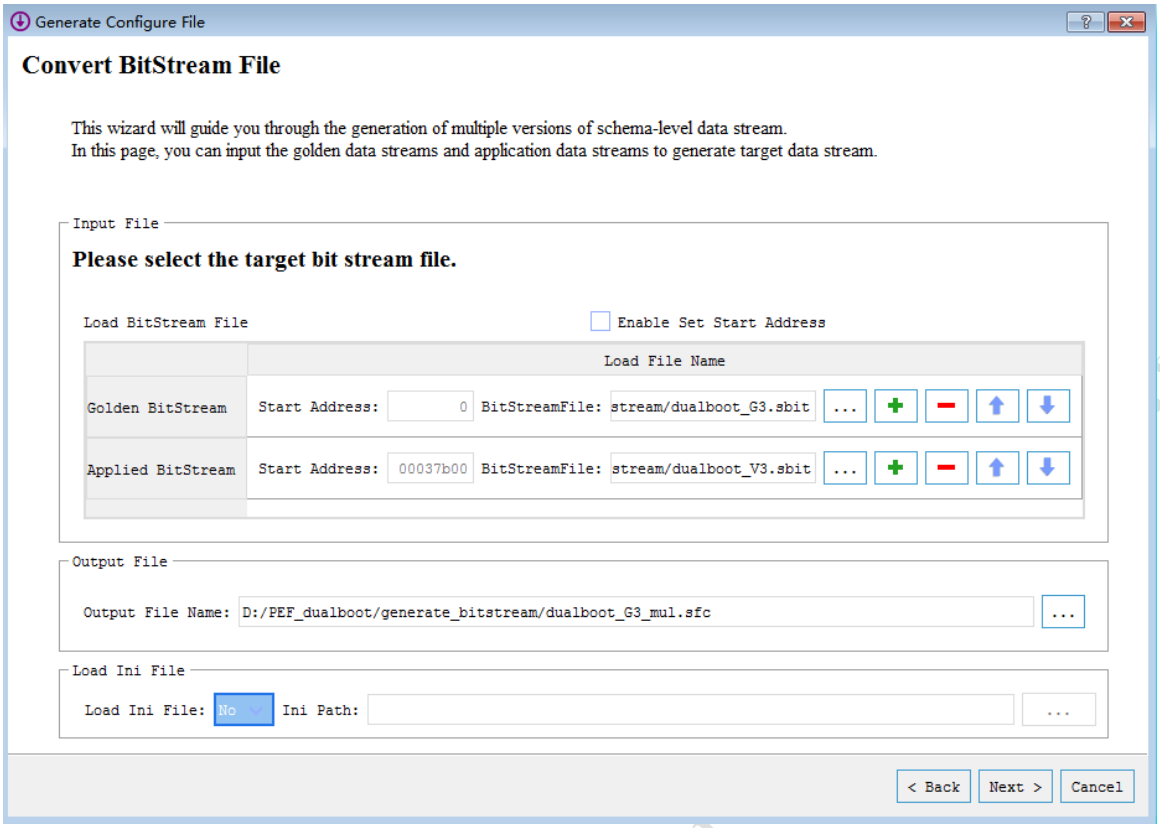


Figure8-21

For the size of the bitstream files, please refer to the table below.

Table 8-2

Device	**Master Self Configuration Dual Boot bitstream size (Kbyte)**	**Golden bitstream page range**	**Application bitstream page range**	**User data page range**
PGC4K	285.5	0~570	571~1141	1142~1279
PGC7K	445.5	0~890	891~1781	1782~1807
PGC10K	638.75	0~1277	1278~2554	2555~2559

The conversion relationship between the [Start Address] and the page is:

Page=Addr*8/2048.

Taking PGC7KD as an example, the default start address for the application bitstream is 0x37b00, which converts to the decimal number 228096, and the starting page is 228096*8/2048=891.

Set the output file, as shown in the figure below.

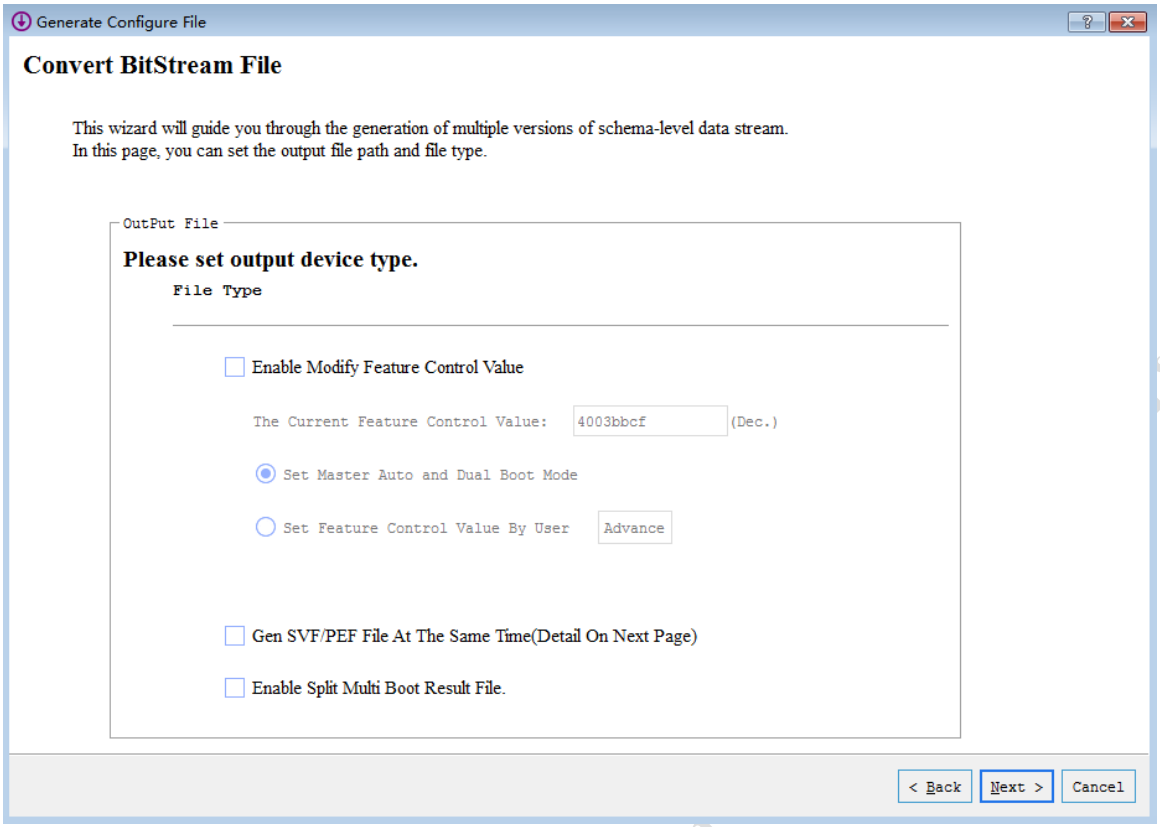


Figure8-22

[Enable Modify Feature Control Value]: If the feature control bits have been set, this option need not be selected; if it is selected, further settings of the feature control bits can be made.

[Set Master Auto and Dual Boot Mode]: Only the configuration bits related to Master Self Configuration Dual Boot are enabled; all other bits remain unchanged;

[Set Feature Control Value By User]: Clicking "Advance" allows for the complete setting of feature control bits.

[Gen SVF/PEF File At The Same Time (Detail On Next Page)]: If this option is selected, the .svf and .pef files will be generated at the same time for the Master Self Configuration Dual Boot bitstreams. Click [Next] to proceed with further settings;

[Enable Split Multi Boot Result File]: If this option is selected, .sfc file will be generated for the golden bitstream and the application bitstream, respectively; if [Gen SVF/PEF File At The Same Time] is also selected, .sfc, .svf, and .pef files will be generated for the golden bitstream and the application bitstream, respectively.

[If Gen SVF/PEF File At The Same Time (Detail On Next Page)] is selected, click [Next] to proceed with the settings, as shown in the figure below.

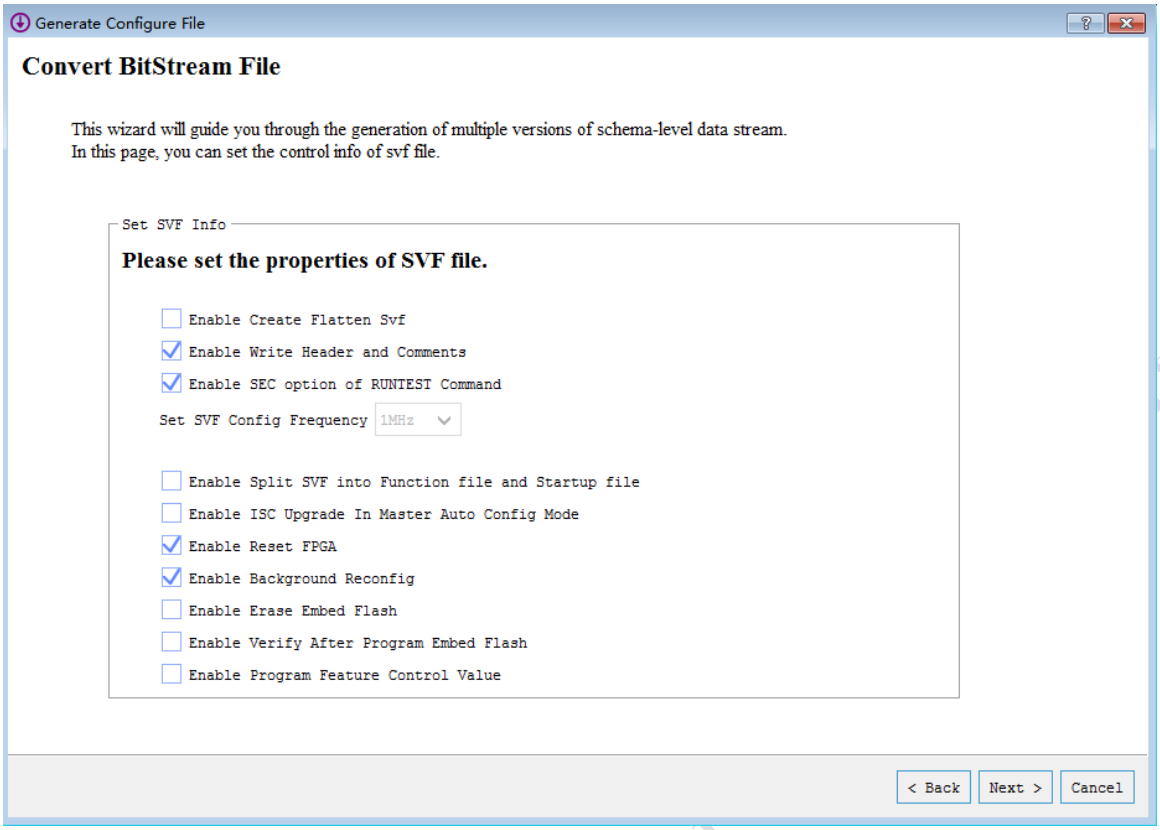


Figure8-23

In this interface, the operational procedures included in the .svf and .pef files can be set. Detailed descriptions for each option are as follows:

[Enable Create Flatten Svf]: If this option is selected, an .svf file that contains SDR/SIR information for populating chains will be generated (Note]: Using this function will not generate a .pef file). The default is off;

[Enable Write Header and Comments]: If this option is selected, file header and comments will be enabled. If it is not selected, the generated .svf file will not include the bitstream header. The default is on;

[Enable SEC option of RUNTEST Command]: The RUNTEST command is used to specify the waiting time the JTAG state machine must maintain when transitioning to a particular state (the specific waiting time is related to the operation option). There are two ways to implement the waiting time]: TCK clock cycles and system delay function. If this option is selected, the waiting time will be implemented through the system delay function; if it is not selected, the waiting time will be implemented through TCK clock cycles. The default is on;

[Set SVF Config Frequency]: This sets the SVF configuration frequency, with the default being 1MHz. If the [Enable SEC option of RUNTEST Command] is selected, the waiting time for the RUNTEST command is independent of the frequency; when it is not selected, the number of TCK clock cycles for the RUNTEST command is directly proportional to the frequency. In this scenario,

the TCK frequency on the user's board must match or be lower than the software-set frequency; if the board's TCK frequency exceeds the software-set frequency, it may lead to configuration failure.

[Enable Split SVF into Function file and Startup file]: If this option is selected, the .svf file will be split into a function file and a startup file. The function file must be configured before the startup file.

[Enable ISC Upgrade In Master Auto Config Mode]: If this option is selected, ISC (In-System Configuration) upgrade will be enabled in the Master Self Configuration mode. In ISC mode, the IO port state will remain unchanged. Please ensure that the chip's JTAGEN pin is kept pulled high during the remote upgrade process.

[Enable Reset FPGA]: If this option is selected, the operation to reset the chip will be added.

[Enable Background Reconfig]: If this option is selected, background reconfiguration will be enabled.

[Enable Erase Embed Flash]: If this option is selected, the operation to erase the embedded Flash will be added. By default, this option is disabled. This operation is not required as the embedded Flash programming process includes a page erase operation, and enabling it will affect the SVF configuration time. It is recommended not to select this option.

[Enable Verify After Program Embed Flash]: If this option is selected, verification will be performed after the embedded Flash is programmed.

[Enable Program Feature Control Value]: If this option is selected, the operation to program the feature control bits will be added.

Then click [Next] until [Finish] to generate the Master Self Configuration Dual Boot bitstream file.

4. Download

Right-click the device icon, select [Assign New Configuration File] from the menu, and then select the generated Master Self Configuration Dual Boot bitstream .sfc file.

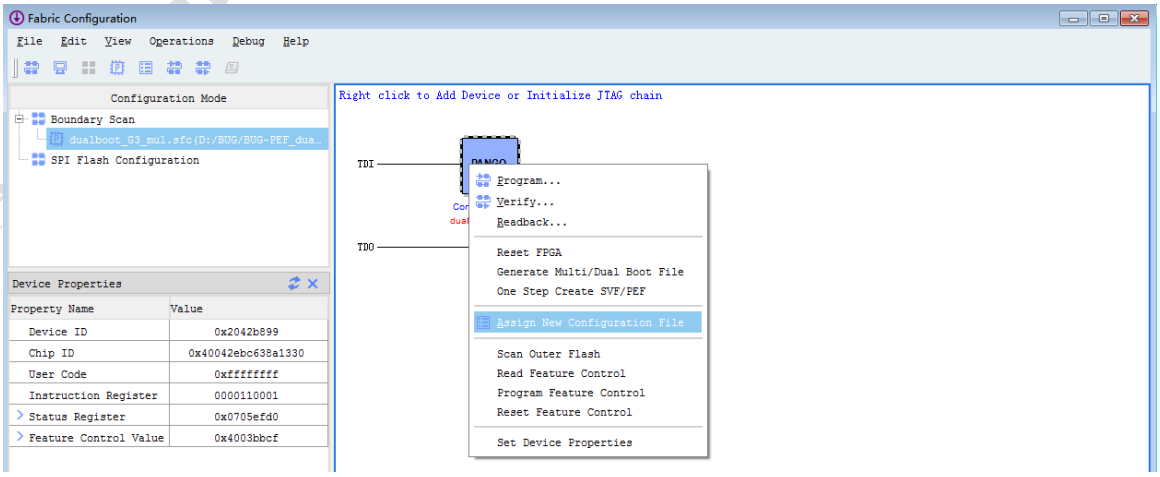


Figure8-24

Then right-click the "FLASH" icon and select [Program] to complete the download.

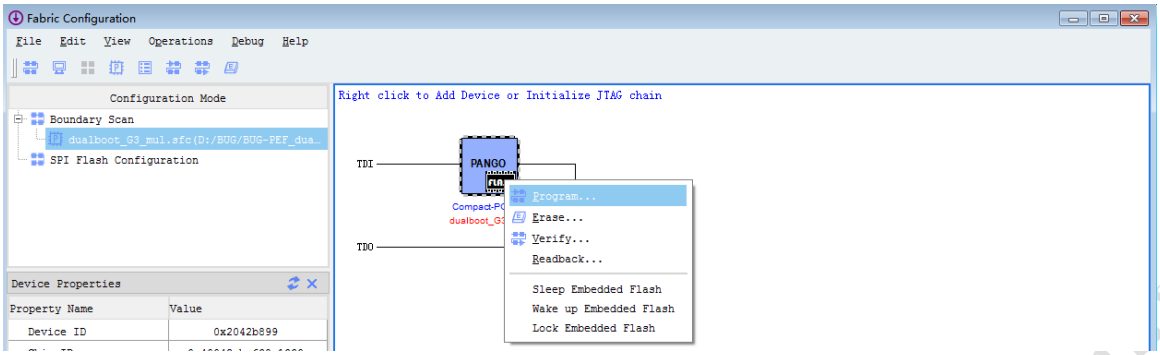


Figure8-25

8.6 Remote Upgrade

Remote upgrade typically refers to the process where a user utilizes, such as a CPU or MCU, to simulate the JTAG interface, reprogramming the embedded Flash of the PGC device through the SPI or I2C interface. After programming, the host computer sends a reset command to restart the PGC device through Master Self Configuration and load a new bitstream into the CRAM, thus achieving the purpose of function upgrading.

The Fabric Configuration of PDS supports remote upgrade debugging through the JTAG interface. The JTAG interface is used to download standard SVF (Serial Vector Format) files. It also supports the download of the .pef file, which is the binary equivalent of the .svf file.

8.6.1 Generate .svf/.pef File

This section primarily explains how to generate the .svf/.pef file required for remote upgrade and the functions of these files.

Right-click the device icon and select [One Step Create SVF/PEF] from the menu, as shown in the figure below.

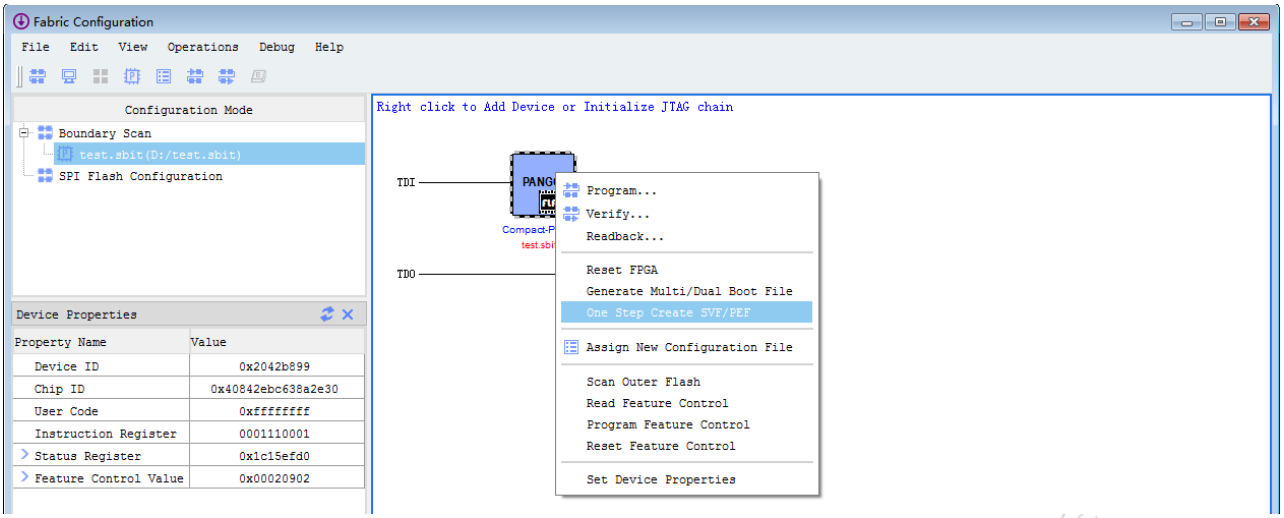


Figure8-26

In the [Select SVF Type] section on the [Generate SVF File] interface, there are 3 options available for selection to implement different functions, namely [Generate For CRAM], [Generate For Embedded Flash], and [Generate For Special Command]. Below is an explanation of their functions.

- If [Generate For Embedded Flash] is selected, the following options will be available:

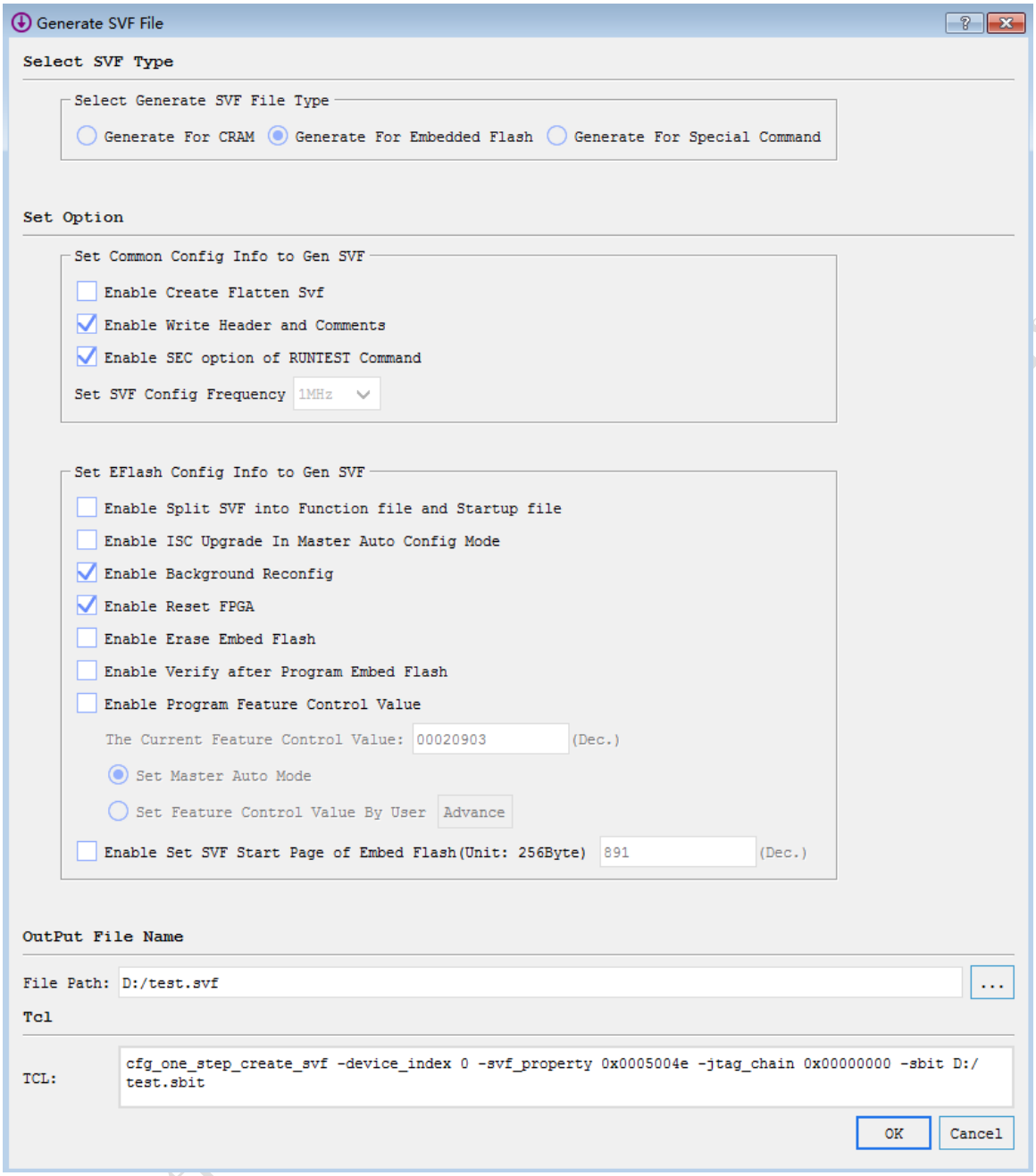


Figure8-27

[Enable Create Flatten Svf]: If this option is selected, an .svf file that contains SDR/SIR information for populating chains will be generated (Note]: Using this function will not generate a .pef file). The default is off;

[Enable Write Header and Comments]: If this option is selected, file header and comments will be enabled. The default is on;

[Enable SEC option of RUNTEST Command]: The RUNTEST command is used to specify the waiting time the JTAG state machine must maintain when transitioning to a particular state (the specific waiting time is related to the operation option). There are two ways to implement the waiting time]: TCK clock cycles and system delay function. If this option is selected, the waiting time will be implemented through the system delay function; if it is not selected, the waiting time

will be implemented through TCK clock cycles. The default is on;

[Set SVF Config Frequency]: This sets the SVF configuration frequency, with the default being 1MHz. If the [Enable SEC option of RUNTEST Command] is selected, the waiting time for the RUNTEST command is independent of the frequency; when it is not selected, the number of TCK clock cycles for the RUNTEST command is directly proportional to the frequency. In this scenario, the TCK frequency on the user's board must match or be lower than the software-set frequency; if the board's TCK frequency exceeds the software-set frequency, it may lead to configuration failure.

[Enable Split SVF into Function file and Startup file]: If this option is selected, the .svf file will be split into a function file and a startup file. The function file must be configured before the startup file.

[Enable ISC Upgrade In Master Auto Config Mode]: If this option is selected, ISC (In-System Configuration) upgrade will be enabled in the Master Self Configuration mode. In ISC, the IO port state will remain unchanged. Please ensure that the chip's JTAGEN pin is kept pulled high during the remote upgrade process.

[Enable Background Reconfig]: If this option is selected, background reconfiguration will be enabled.

[Enable Reset FPGA]: If this option is selected, the operation to reset the chip will be added.

[Enable Erase Embed Flash]: If this option is selected, the operation to erase the embedded Flash will be added. By default, this option is disabled. This operation is not required as the embedded Flash programming process includes a page erase operation, and enabling it will affect the SVF configuration time. It is recommended not to select this option.

[Enable Verify After Program Embed Flash]: If this option is selected, verification will be performed after the embedded Flash is programmed.

[Enable Program Feature Control Value]: If this option is selected, operations for the programming feature control bits will be added.

[Set Master Auto Mode]: If this option is selected, only the configuration bits related to the Master Self Configuration will be set; other bits will remain unchanged.

[Set Feature Control Value By User]: Clicking "Advance" allows for the complete setting of feature control bits.

[Enable Set SVF Start Page of Embed Flash]: If this option is selected, the start address of the download bitstream can be set.

- If [Generate For CRAM] is selected, the following function options will be available:

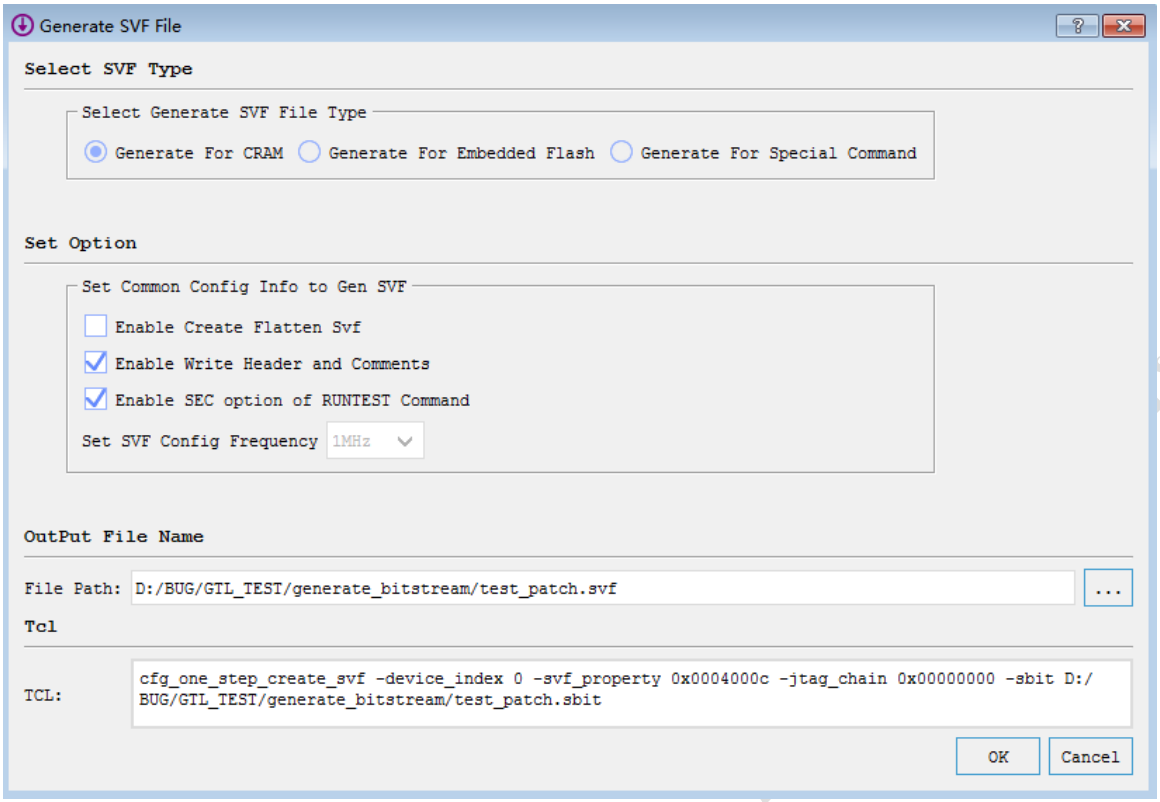


Figure8-28

[Enable Create Flatten Svf]: If this option is selected, an .svf file that contains SDR/SIR information for populating chains will be generated (Note]: Using this function will not generate a .pef file). The default is off;

[Enable Write Header and Comments]: If this option is selected, file header and comments will be enabled. The default is on;

[Enable SEC option of RUNTEST Command]: The RUNTEST command is used to specify the waiting time the JTAG state machine must maintain when transitioning to a particular state (the specific waiting time is related to the operation option). There are two ways to implement the waiting time]: TCK clock cycles and system delay function. If this option is selected, the waiting time will be implemented through the system delay function; if it is not selected, the waiting time will be implemented through TCK clock cycles. The default is on;

[Set SVF Config Frequency]: This sets the SVF configuration frequency, with the default being 1MHz. If the [Enable SEC option of RUNTEST Command] is selected, the waiting time for the RUNTEST command is independent of the frequency; when it is not selected, the number of TCK clock cycles for the RUNTEST command is directly proportional to the frequency. In this scenario, the TCK frequency on the user's board must match or be lower than the software-set frequency; if the board's TCK frequency exceeds the software-set frequency, it may lead to configuration failure.

- If [Generate For Special Command] is selected, the following function options will be available:

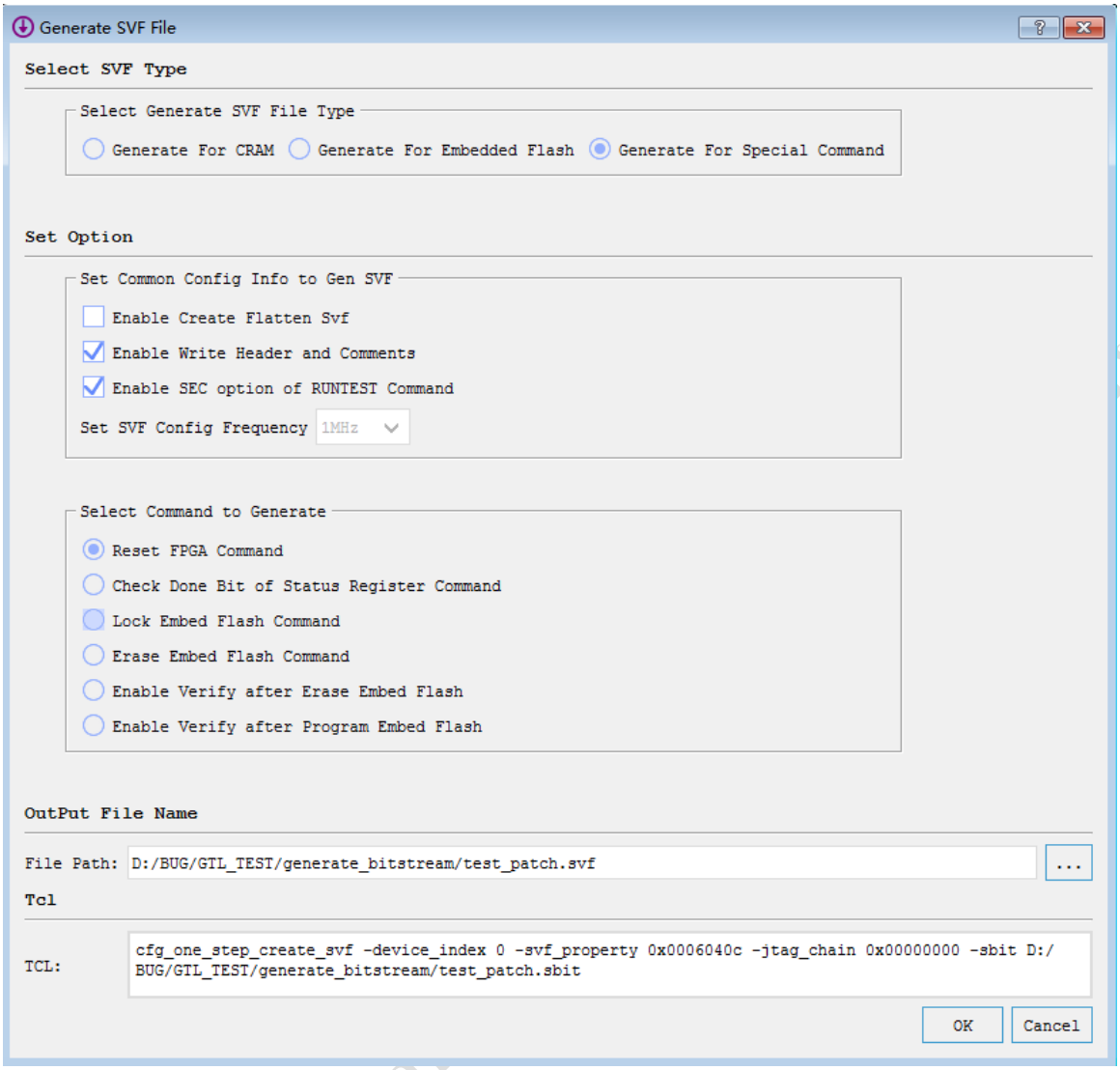


Figure8-29

[Enable Create Flatten Svf]: If this option is selected, an .svf file that contains SDR/SIR information for populating chains will be generated (Note]: Using this function will not generate a .pef file). The default is off;

[Enable Write Header and Comments]: If this option is selected, file header and comments will be enabled. The default is on;

[Enable SEC option of RUNTEST Command]: The RUNTEST command is used to specify the waiting time the JTAG state machine must maintain when transitioning to a particular state (the specific waiting time is related to the operation option). There are two ways to implement the waiting time]: TCK clock cycles and system delay function. If this option is selected, the waiting time will be implemented through the system delay function; if it is not selected, the waiting time will be implemented through TCK clock cycles. The default is on;

[Set SVF Config Frequency]: This sets the SVF configuration frequency, with the default being 1MHz. If the [Enable SEC option of RUNTEST Command] is selected, the waiting time for the

RUNTEST command is independent of the frequency; when it is not selected, the number of TCK clock cycles for the RUNTEST command is directly proportional to the frequency. In this scenario, the TCK frequency on the user's board must match or be lower than the software-set frequency; if the board's TCK frequency exceeds the software-set frequency, it may lead to configuration failure.

[Reset FPGA Command]: Resets the CPLD;

[Check Done Bit of Status Register Command]: Checks the Done indicator status in the status register;

[Lock Embed Flash Command]: Locks the embedded Flash, making it inaccessible for read/write operations;

[Entire Embed Flash Command]: Erases the entire embedded Flash;

[Enable Verify after Erase Embed Flash]: Verifies after the embedded Flash is erased;

[Enable Verify after Program Embed Flash]: Verifies after the embedded Flash is programmed

After completing the settings in the [Generate SVF File] interface, click [OK] to generate the .svf/.pef file.

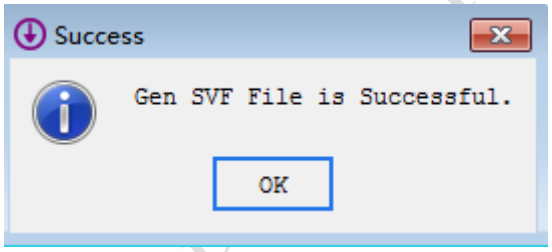


Figure8-30

8.6.2 Download .svf/.pef File

To download the .svf/.pef file, please ensure that the JtagServer is connected but the devices will not be automatically scanned.

Right-click the blank area to the right of Fabric Configuration, select [Add Pango Device] from the pop-up menu to select the .svf/.pef file to download, as shown below.

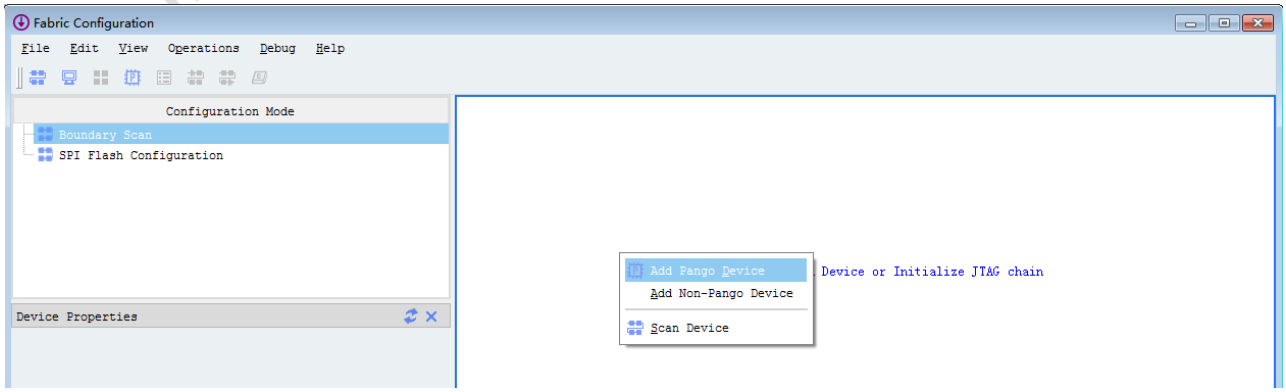


Figure8-31

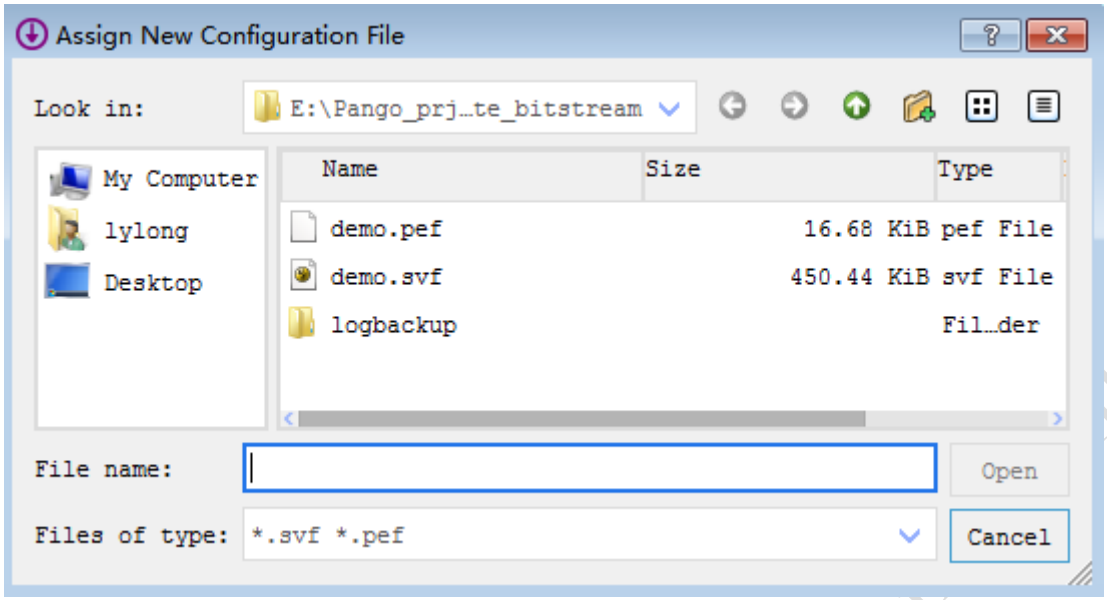


Figure8-32

After adding the file, right-click the icon and select [Execute SVF/PEF File..] to download it.

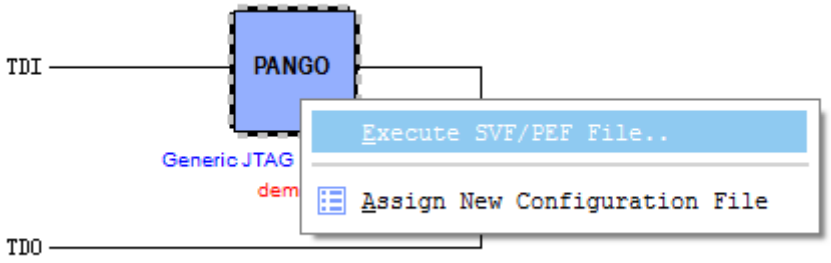


Figure8-33

8.7 Other Download Methods and Reference Documents

Users can use a host computer, such as a CPU or MCU, to simulate JTAG interface, slave SPI interface, or slave I2C interface to program the embedded Flash of PGC devices or directly download CRAM. Users can refer to the following documents to meet their application needs.

Table 8-3

Application Document	Content Description
AN03008_Compa Family Loaded Platform Software Application Guide	Describes how to use an MCU to simulate the JTAG interface for loading the svf/pef file
AN03009_Compa Family CPLDs Device Slave SPI Interface Configuration and Programming Application Guide	Describes how to use an MCU to simulate slave SPI interface to download CRAM and embedded Flash and provides a recommended download process.
AN03013_Compa Family CPLDs Device JTAG Interface Configuration and Programming Application Guide	Describes how to use an MCU to simulate JTAG interface to download CRAM and embedded Flash and provides a recommended download process.
AN03014_Compa Family CPLDs Device Slave I2C Interface Configuration and Programming Application Guide	Describes how to use an MCU to simulate slave I2C interface to download CRAM and embedded Flash and provides a recommended download process.

Application Document	Content Description
UG030004_Compa Family CPLDs Configuration User Guide	Provides detailed information about the download process, download interface, supported commands, timing, etc.

Chapter 9 Capture Waveform

The PDS software provides a method for users to observe the operational status of internal signals in their designs, which mainly includes two steps. First, use Fabric Inserter to automatically insert the information of the signals to be observed into the user's design netlist, thereby generating a new netlist; then generate a bitstream with the new netlist and download the bitstream and .fic file using Fabric Debugger, at which point the inserted signals can be observed with Fabric Debugger. Below are the instructions for using Fabric Inserter and Fabric Debugger.

9.1 Fabric Inserter

The main function of Fabric Inserter is to insert signals that the user wishes to observe into the user's design netlist, creating a new design netlist.

The method to start the Fabric Inserter is as follows.

Click  in the toolbar, or click [Tools > Inserter] to open the Inserter configuration wizard, as shown in the figure below.

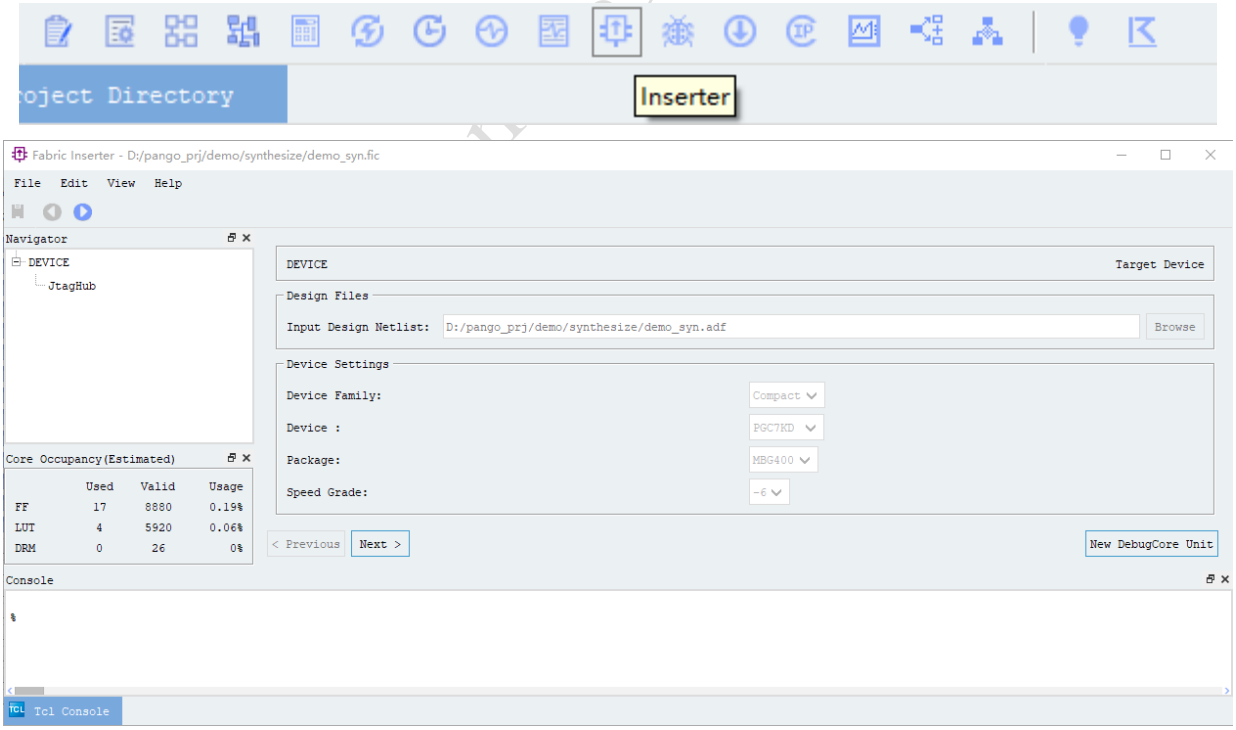


Figure9-1

Click [Next] to enter the DebugCore configuration interface, as shown in the figure below.

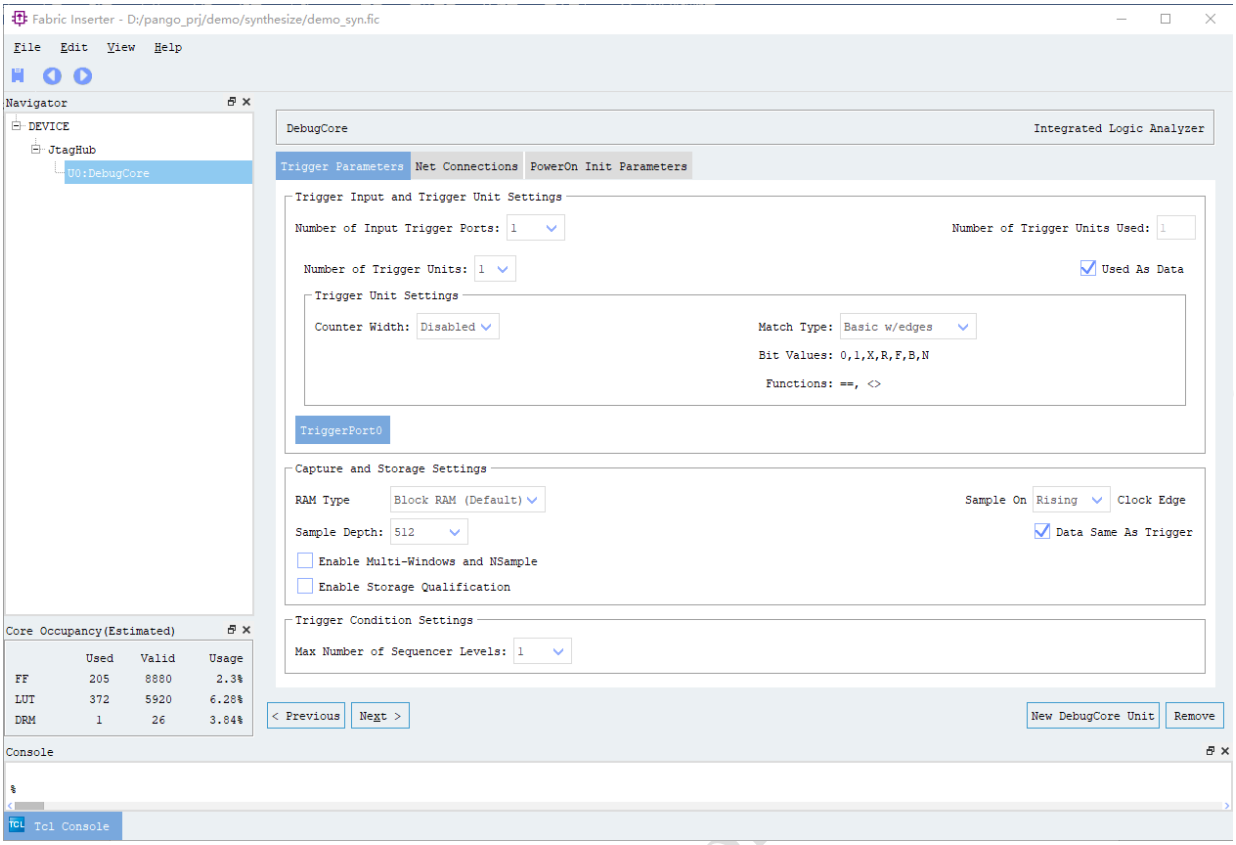


Figure9-2

The DebugCore configuration interface includes three pages: [Trigger Parameters], [Net Connections], and [PowerOn Init Parameters], which need to be configured one by one.

9.1.1 Configure Parameters

Users configure parameters on the [Trigger Parameters] page. The functions of each option are as follows:

[Number of Input Trigger Ports]: The number of input trigger ports for DebugCore, with a maximum value of 16;

[Number Of Trigger Units]: The number of matching units affected by the trigger port, which share this trigger port, with a maximum value of 16;

[Number of Trigger Units Used]: The total number of matching units in the current DebugCore;

[Used As Data]: Sets whether the input signal of the current trigger port is also used as the input signal for the data capture port (only effective when [Data Same As Trigger] is enabled);

[Counter Width]: The width of the counter for the matching units, which can be used to count the number of times the match conditions for the matching units are met;

[Match Type]: Sets the trigger condition, with options including [Basic], [Basic w/edges], [Extended], [Extended w/edges], [Range], and [Range w/edges]. This option is associated with [Bit

Values] and [Functions];

[RAM Type]: Sets the implementation method for DebugCore storage resources. [Block RAM (Default)] indicates the implementation using DRM; [Distribute RAM] indicates the implementation using distributed RAM;

[Sample On]: Sets the sampling clock edge, with Rising for the clock's rising edge and Falling for the falling edge;

[Sample Depth]: Sets the sampling depth, which ranges from 64 to 131072;

[Enable Multi-windows and NSample]: Sets whether to enable multi-window triggering and the Nsample function; by default, it is not enabled. Enabling this function will have an impact on the timing performance of DebugCore. It is not recommended to enable this function if DebugCore is required to operate at higher frequencies;

[Enable Storage Qualification]: Sets whether to enable storage conditions. If this option is selected, specific conditions can be set to filter data to be captured, storing only data that meets the conditions and displaying it in the Debugger;

[Data Same As Trigger]: Sets whether the signals connected to the trigger ports are also connected to the data capture port. If this option is enabled, all signals connected to trigger ports with the Used As Data option selected will also be connected to the data capture port, and no additional connections to the data capture port can be made;

[Max Number of Sequence Levels]: Used to select the maximum number of trigger condition levels n, not exceeding 16.

9.1.2 Signal Connection

Click [Next] to enter the [Net Connections] page, where users can connect the DebugCore's input signals to the nets in their design. Signals that are not yet connected are displayed in red. This is shown in the following figure.

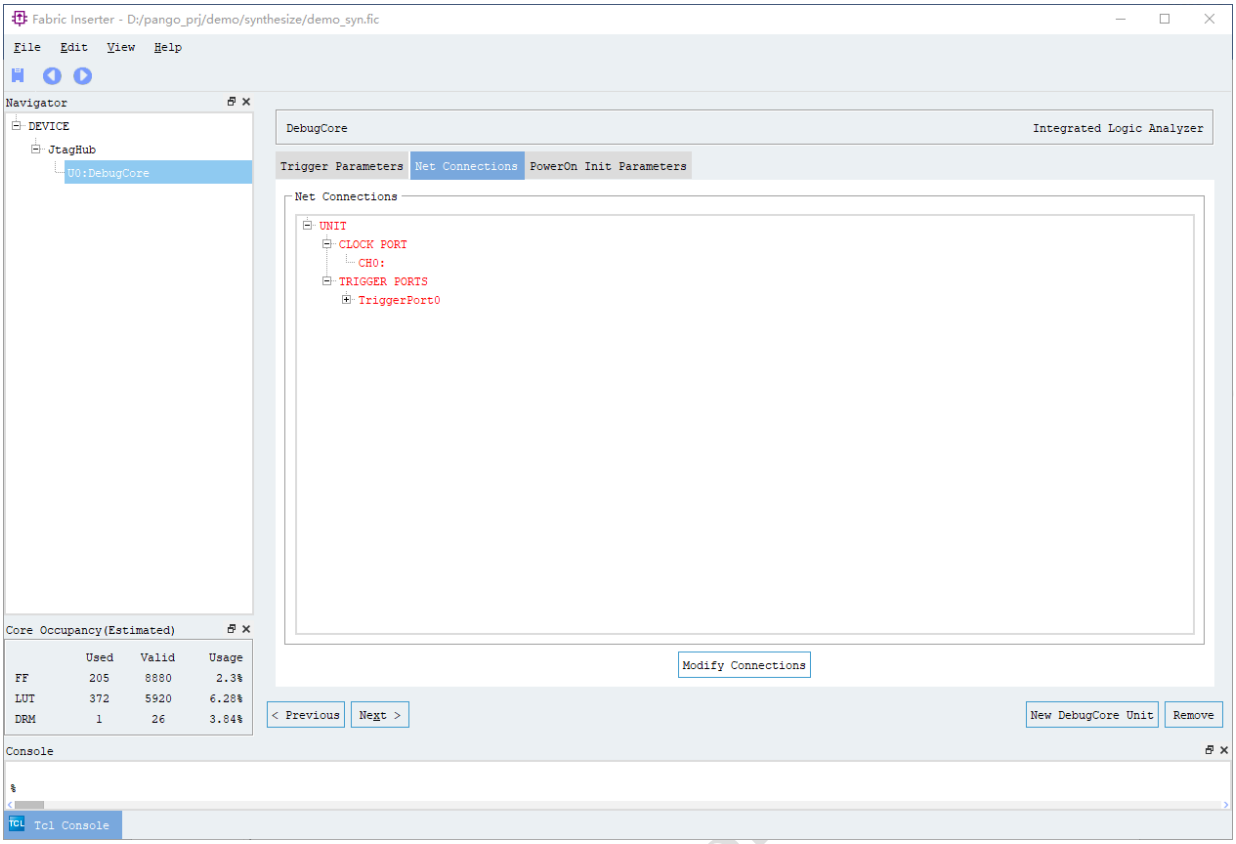


Figure9-3

Click [Modify Connection], and the [Select Net] configuration interface will appear, as shown in the figure below. The Select Net interface is mainly divided into 4 areas: Module View, Filter ToolBar, Net View, and Channel View.

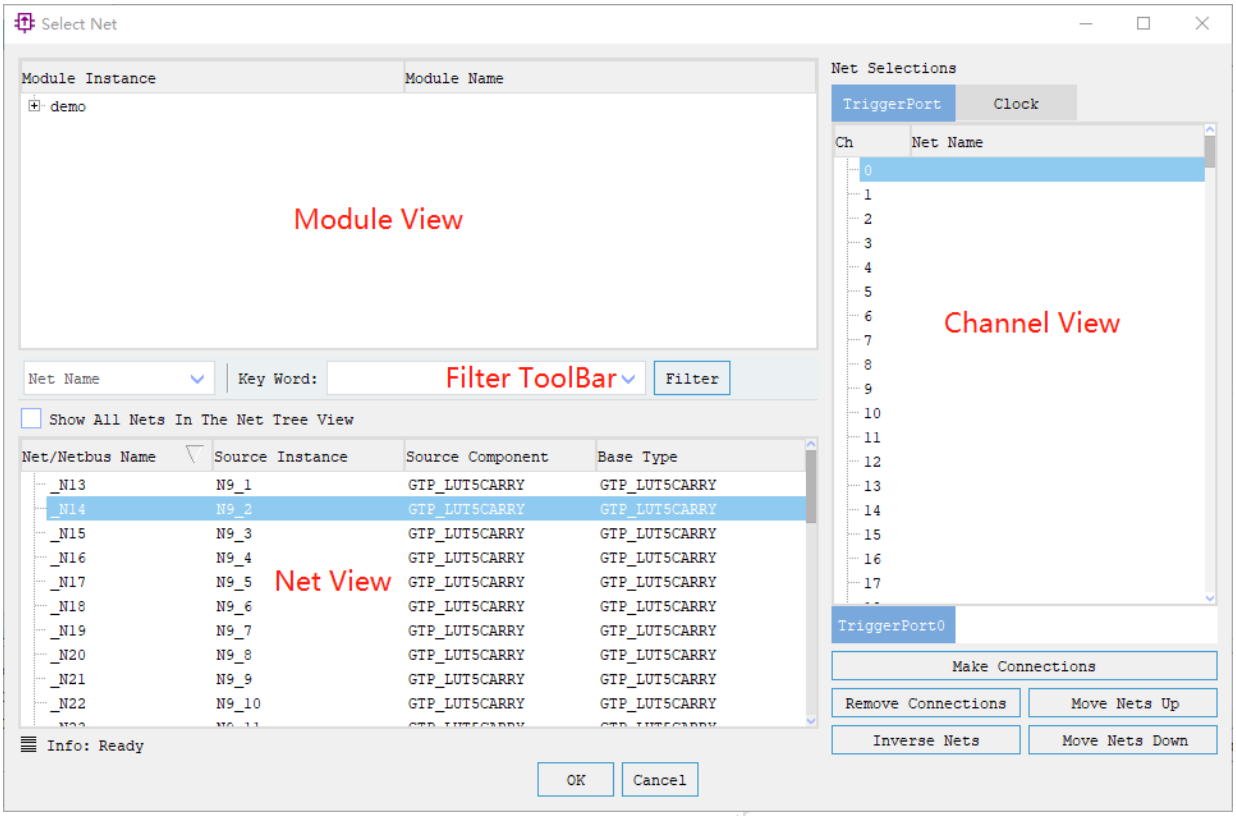


Figure9-4

Module View displays the hierarchical structure of the user's design, while Net View displays the signals in the user's design. Click different hierarchical modules in [Module View], and the signals contained in them will be displayed in [Net View]. If [Show All Nets In The Net Tree View] is selected, the [Module View] area will be grayed out, and [Net View] will display all signals in the user's design.

The Filter ToolBar is mainly used for filtering signals in Module View and Net View. The drop-down box on the left allows for the selection of filtering modes, while the Key Word field on the right is for entering search keywords.

There are mainly the following 7 filtering modes:

[Module Instance]: Used to filter modules in [Module View], focusing only on whether the information in the [Module Instance] column (the first column of [Module View]) meets the criteria;

[Module Name]: Used to filter modules in [Module View], focusing only on whether the information in the [Module Name] column (the second column of [Module View]) meets the criteria;

[Net -All]: Used to filter signals and buses in [Net View], focusing on all four columns of information. If any type of information in a signal meets the criteria, this signal is considered qualified;

[Net Name]: Used to filter signals and buses in [Net View], focusing only on whether the

information in the [Net/Netbus Name] column (the first column) meets the criteria;

[Source Instance]: Used to filter signals and buses in [Net View], focusing only on whether the information in the [Source Instance] column (the second column) meets the criteria;

[Source Component]: Used to filter signals and buses in Net View, focusing only on whether the information in the [Source Component] column (the third column) meets the criteria;

[Base Type]: Used to filter signals and buses in [Net View], focusing only on whether the information in the [Base Type] column (the fourth column) meets the criteria.

The Channel View area displays all the channels that can be connected in the current DebugCore. This window contains multiple tabs, each representing a port (such as TriggerPort or Clock). The maximum number of channels for each port is different. For example, the TriggerPort has 256 channels, indicating that it can connect up to 256 signals; the Data Port has 4096 channels, indicating that it can connect up to 4096 signals (and will not be displayed in this window when Data Same As Trigger is enabled); the Clock Port and the Reset Port each have only one channel, and must be connected to the corresponding clock signal and reset signal, respectively. Incorrect connections to the Clock Port and Reset Port will cause the DebugCore to malfunction. In addition, the trigger port and the data capture port must be connected to at least one signal.

Channel View contains multiple buttons. Their functions are as follows:

[Make Connections]: Connects the signal or bus selected in [Net View] to the channel selected in [Channel View], with the shortcut key "Ctrl+M";

[Remove Connections]: Removes the connection of one or more channels selected in [Channel View], with the shortcut key "Ctrl+R";

[Inverse Nets]: Inverts the connection order of multiple channels selected in [Channel View] and reconnects them, with the shortcut key "Ctrl+I";

[Move Nets Up]: Moves one or more channels selected in [Channel View] upwards, with the shortcut key "Ctrl+Up";

[Move Nets Down]: Moves one or more channels selected in [Channel View] downwards, with the shortcut key "Ctrl+Down";

[OK]: Saves the current connections and exits the configuration interface, with the shortcut keys "Ctrl+S" (save current connections only) and "ESC" (exit window);

[Cancel]: Exits the interface without saving changes, with the shortcut key "ESC".

For signal connection, in addition to the above buttons, the software also supports the drag-and-drop operation. Users simply need to drag the target signal from the Net View on the left to the corresponding position in the Channel View on the right.

In addition, users can create a new bus, split an existing bus, or rename a bus by right-clicking the

Channel View area and selecting from the menu.

By selecting multiple consecutive and connected channels in Channel View, right-clicking them, and selecting [Make Bus] from the pop-up menu, a bus can be created. The newly created bus is automatically named "CustomBus", and users can rename it by right-clicking the bus and selecting [Rename Bus] from the pop-up menu, or by double-clicking the bus with the left mouse button. To split a bus, select the bus to be split, right-click it, and select [Split Bus] from the pop-up menu to split it into separate nets.

9.1.3 PowerOn Init Parameters Settings

Click [Next] to enter the [PowerOn Init Parameters] page, as shown in the figure below. This step can be skipped when there is no need to observe the initial state of the signals.

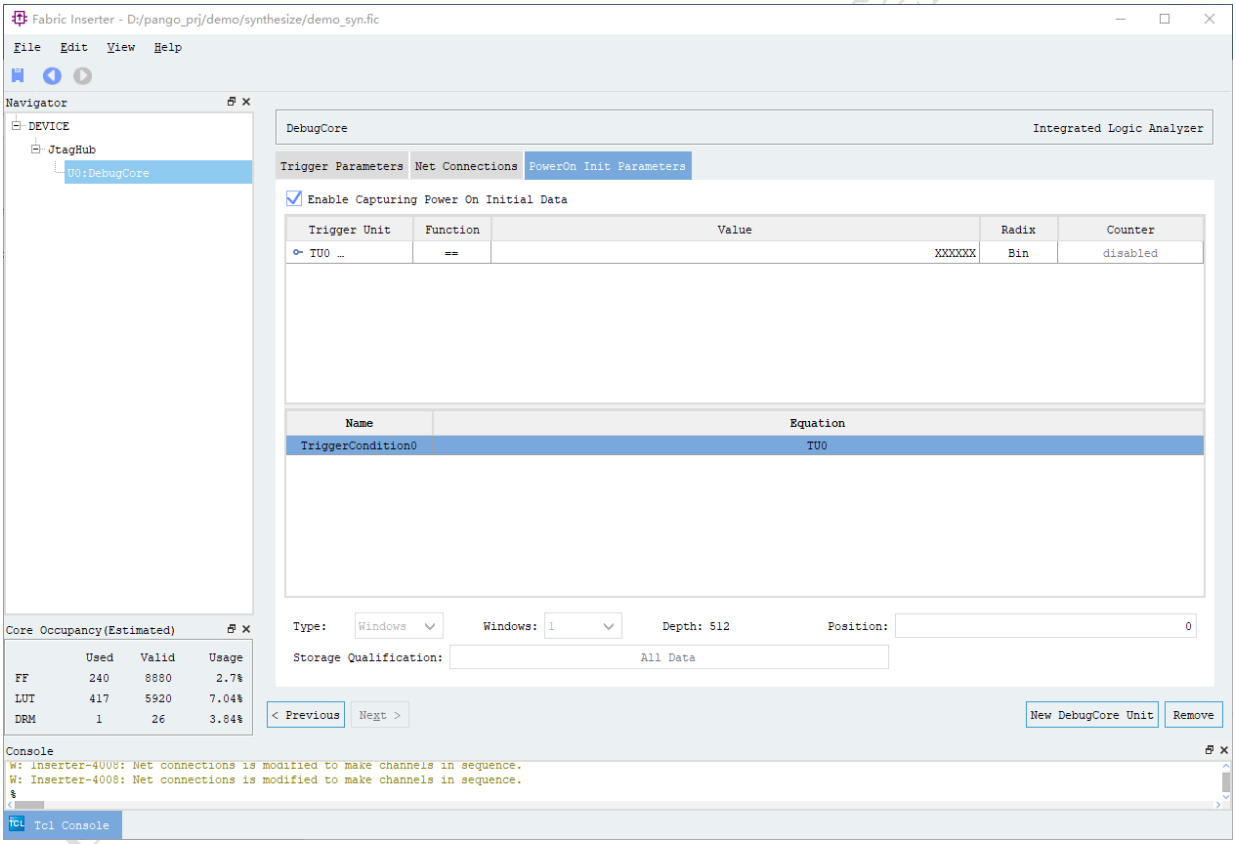


Figure9-5

This page is used to set the signal capture conditions during power-up initialization, as well as the status and stability of signals at the start of power-up initialization. By default, capturing Power on initial data is not enabled. To enable this function, click [Enable Capturing Power On Initial Data]. Below is a description of the parameters on this page:

[Trigger Unit]: Lists all Trigger Units in the current DebugCore, with the default name TU0. Click an item in the Trigger Unit column, and the channels within the Trigger Unit will be expanded.

Users can directly edit the value of a single signal based on its name;

[Function]: Used to select the conditions for signal triggering;

Value]: Used to input the value that meets the trigger condition, which is displayed according to the selected Radix column;

[Radix]: Used to select in which number system the Trigger Unit value is displayed. Options include Hex, Octal, Bin, and Unsigned:

Hex: X, ?, 0-9, and A-F. X indicates the corresponding 4-bit value is don't-care, and can be any value. "?" indicates the corresponding 4-bit value includes a combination of 0-9, A-F, X, R, F, B, and N; this symbol cannot be input;

Octal: X, ?, 0-7. "?" indicates the corresponding 3-bit value includes a combination of 0-7, X, R, F, B, and N; this symbol cannot be input;

[Binary]: X (don't-care value), 0, 1, R (rising edge), F (falling edge), B (any transition), N (no transition); R, F, B, and N can be input when [Match Type] is set to [Basic w/edges], [Extended w/edges], or [Range w/edges];

Unsigned: 0 to 2 to the power of n minus one, where n is the number of signals in the corresponding Trigger Port for the Trigger Unit;

[Counter]: Used to select how many times the Function of the Trigger Unit must be met before the expression for the Trigger Unit is considered satisfied. If the Counter in the Trigger Unit is available, the column will be in black text; if not available, it will be in grey. Click this column, and the Match Function configuration dialogue will pop up, as shown in the figure below.

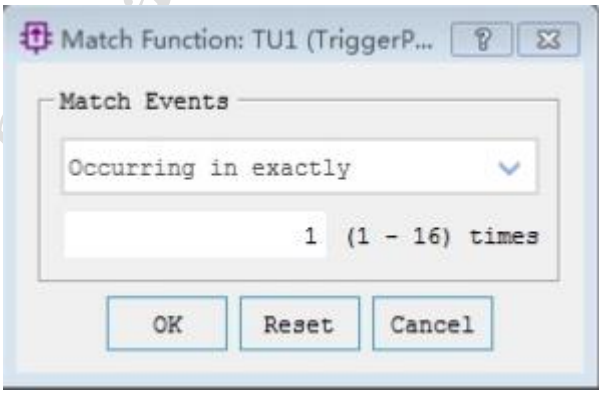


Figure9-6

The Match Events drop-down menu contains 4 options:

[counter disable]: Do not use the counter;

[Occurring in exactly]: Matches the conditions set by the Trigger Unit exactly N times (N is the value entered in the text box);

[Occurring in at least]: Matches the conditions set by the Trigger Unit at least N times (N is the value entered in the text box);

[Lasting for at least]: Matches the conditions set by the Trigger Unit continuously for at least N times (N is the value entered in the text box).

Name	Equation
TriggerCondition0	TU0

The Trigger Condition is a Boolean expression or sequence composed of one or more Trigger Units, and is used to guide the capture of DebugCore data.

[Name]: Used to edit the name of the Trigger Condition, with the default name being "TriggerCondition0";

[Equation]: Displays the Boolean expression or sequence composed of Trigger Units. The sequence can be modified by clicking the column to pop up the [rigger Condition] configuration table.

The [Trigger Condition] configuration table includes a [Boolean] tab and a [Sequencer] tab.

The [Boolean] tab includes a table containing all Trigger Units, with each Trigger Unit occupying one row of the table. The "Enable" column is used to select whether to include the Trigger Unit as part of the expression; the "Negate" column determines whether to logically negate the Trigger Unit. All enabled Trigger Units can be combined using the logical AND or OR operation, which can be selected by clicking the radio button [And Equation] or [Or Equation]. "Negate Whole Equation" is used to negate the entire expression. The selected expression is displayed at the bottom of the dialogue, as shown in the figure below.

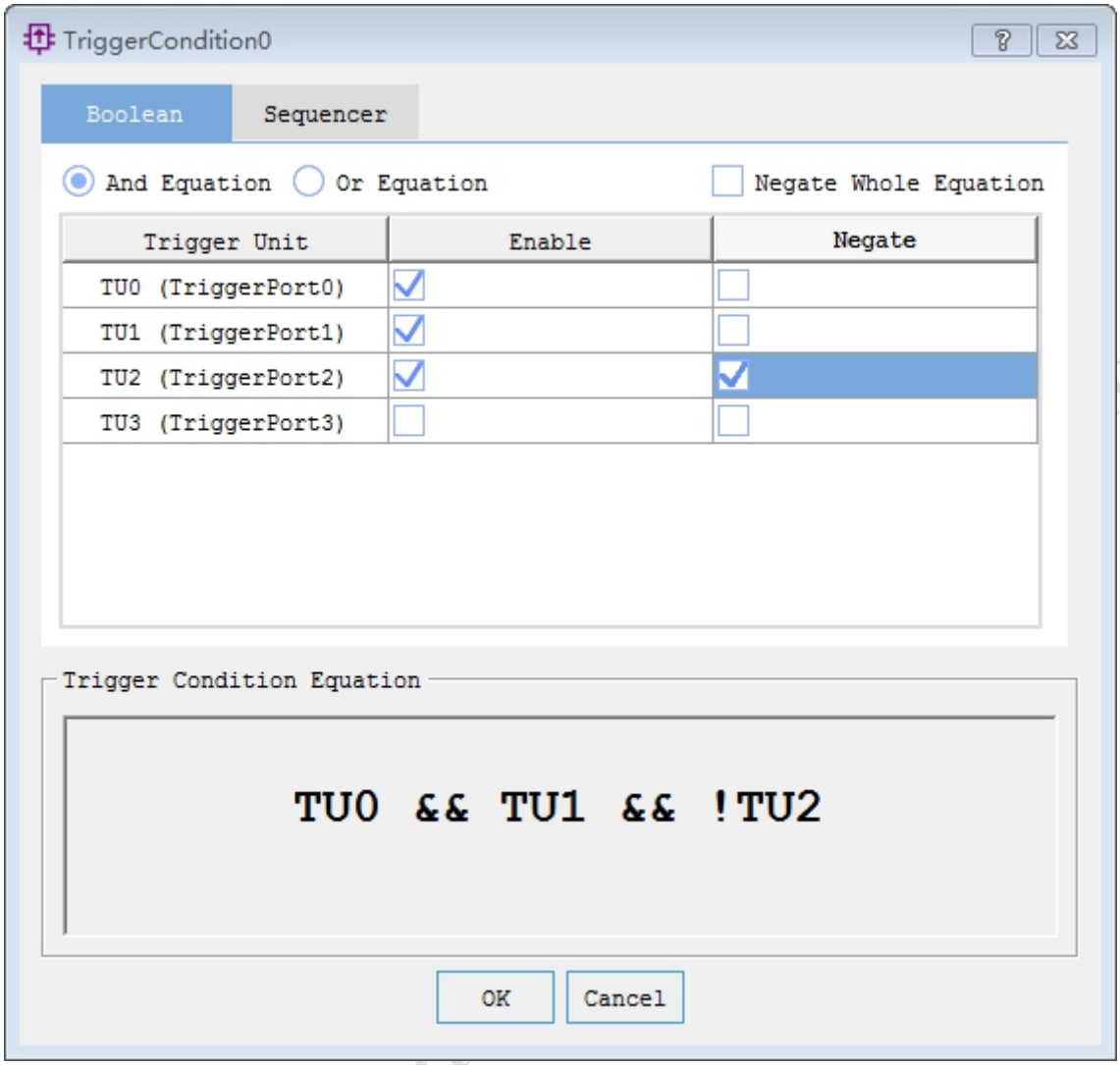


Figure9-7

The [Sequencer] tab includes a [Number of Levels] drop-down list for selecting the number of levels n for trigger conditions, which is one of the current DebugCore parameters and does not exceed 16 levels, and a list with n rows for selecting the Trigger Unit for each level. Once clicked, the items in the [Trigger Unit] column will become a drop-down list showing all Trigger Units available in the current DebugCore, allowing any to be selected. The [Negate] column is used to negate the corresponding Trigger Unit. The selected Trigger Units are queued by Level for triggering, with sequential triggering starting from Level 1 and proceeding to the next level only after the previous one is satisfied, and so on. All triggers can be either continuous or non-continuous, which can be selected via "Use Continuous Match Event Only". The trigger condition expression is displayed at the bottom of the dialogue, as shown in the figure below.

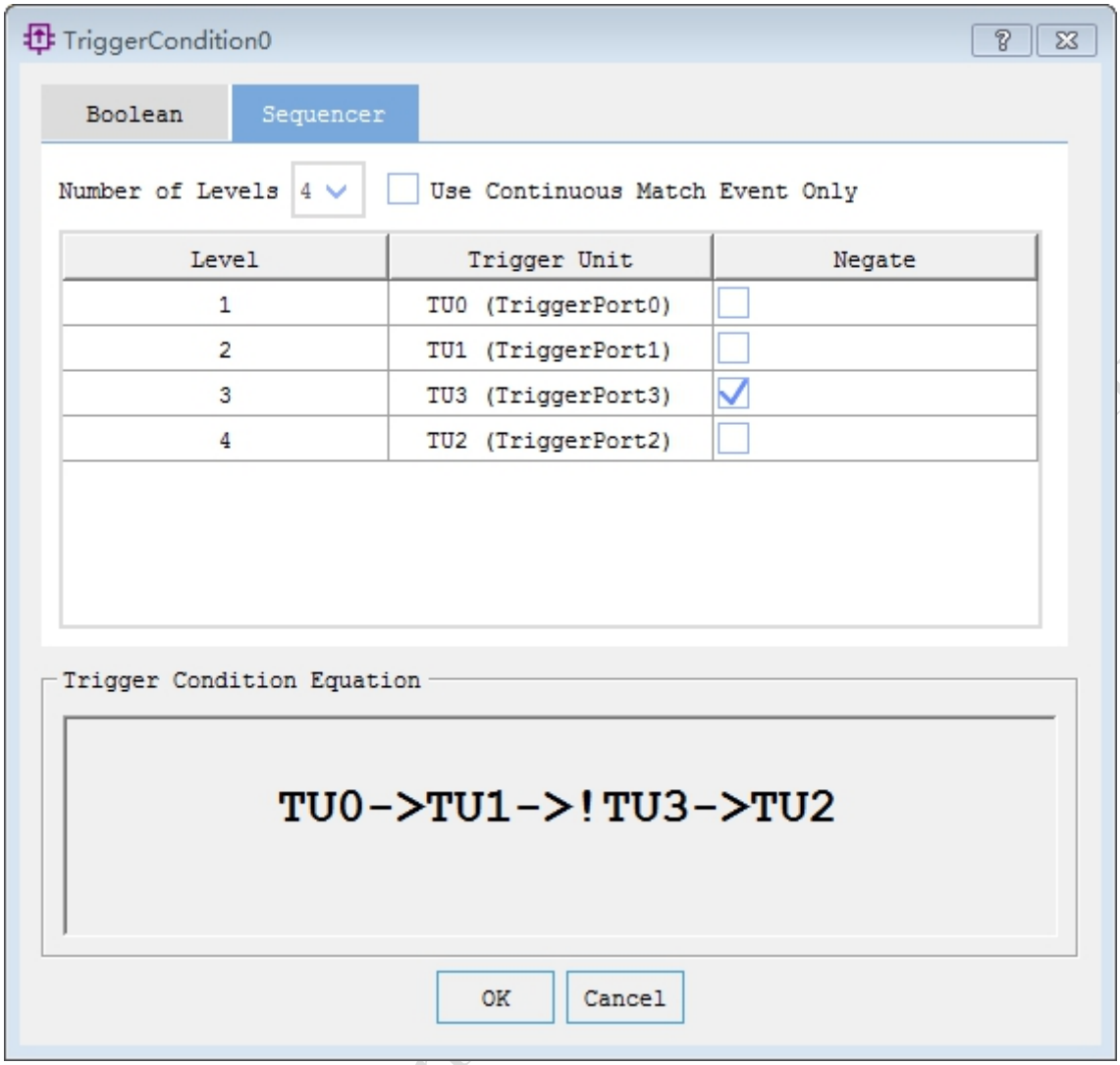


Figure9-8

After completing the operations on the DebugCore configuration interface, click in the top left corner, or select [File > Save Project] from the toolbar to save the configuration. All information is recorded in the .fic file, which is also automatically added to the project list.

9.1.4 Methods to Prevent Signal Optimization

When Fabric Inserter is used to add signals for observation, some signals may be optimized out and therefore cannot be added. In such cases, the following methods can be used.

For reg type signals, add syn_preserve after the signal declaration, for example,

reg [3:0] curstate /* synthesis syn_preserve = 1 */;

For wire type signals, add syn_keep after the signal declaration, for example,

wire signal1 /* synthesis syn_keep=1 */;

If the above methods do not work, PAP_MARK_DEBUG can be used, for example,

wire signal2 /*synthesis PAP_MARK_DEBUG=1*/; supported by ADS only.

For modules that need to be preserved, syn_noprune can be added during module instantiation, for example,

my_design my_design1(out,in,clk_in) /* synthesis syn_noprune=1 */;

Regardless of the method used, it is necessary to ensure that RTL code is written in compliance with the specifications: signals between submodules are effectively connected, critical signals are not left floating, signal bit widths between submodules match, and output signals of submodules are brought up to the top-level signal list.

9.2 Fabric Debugger

Fabric Debugger is an interface-based FPGA/CPLD chip debugging tool that allows for internal debugging of a chip. This software interacts directly with JtagHub and DebugCore, enabling real-time configuration of FPGA/CPLD, setting of trigger conditions, and observation of results.

Before using Fabric Debugger, please ensure that DebugCore has been inserted into the user design and the corresponding .sbit file has been generated.

The method to launch the Fabric Debugger is as follows:

Click  in the PDS toolbar, as shown in the figure below.

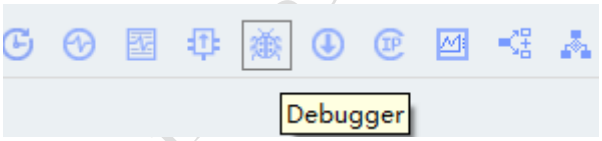


Figure9-9

Or click [Tools > Debugger], and the Fabric Debugger's initial interface will pop up, as shown in the figure below.

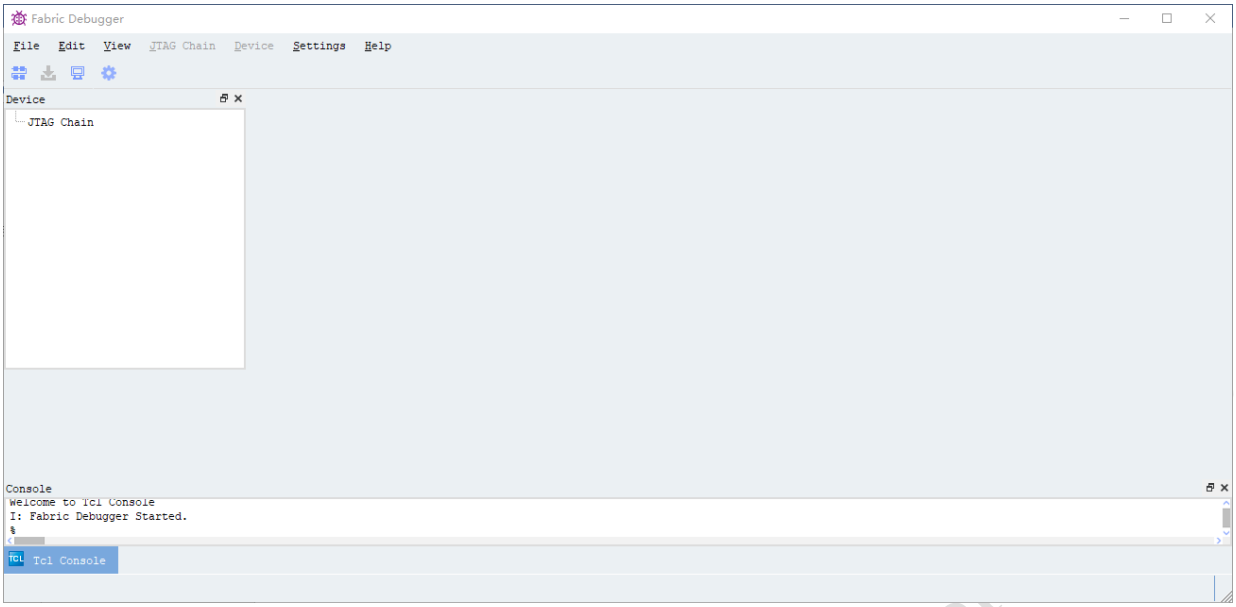


Figure9-10

9.2.1 Connect to Server

Click the button on the toolbar to pop up the Connect to JtagServer wizard, and then click [Connect] to connect to the server. This software uses the TCP/IP protocol to communicate with the server, so it is necessary to specify an IP address and port. The IP address can be set as the local machine's IP address or a remote IP address. If the local IP address is specified, just click [Connect]; if a remote IP address is specified, first start the server remotely using the command line. To do this, switch to the directory where the executable program is located, launch the operating system's shell terminal, and enter the command "cdt_js –port 65420". For more parameters, enter the command "vcdt_js –help".

Figure9-11

9.2.2 Set Cable Parameters

Upon successful connection to the server, click [Next], and set the TCK frequency through the drop-down menu, with the default being 10MHz.

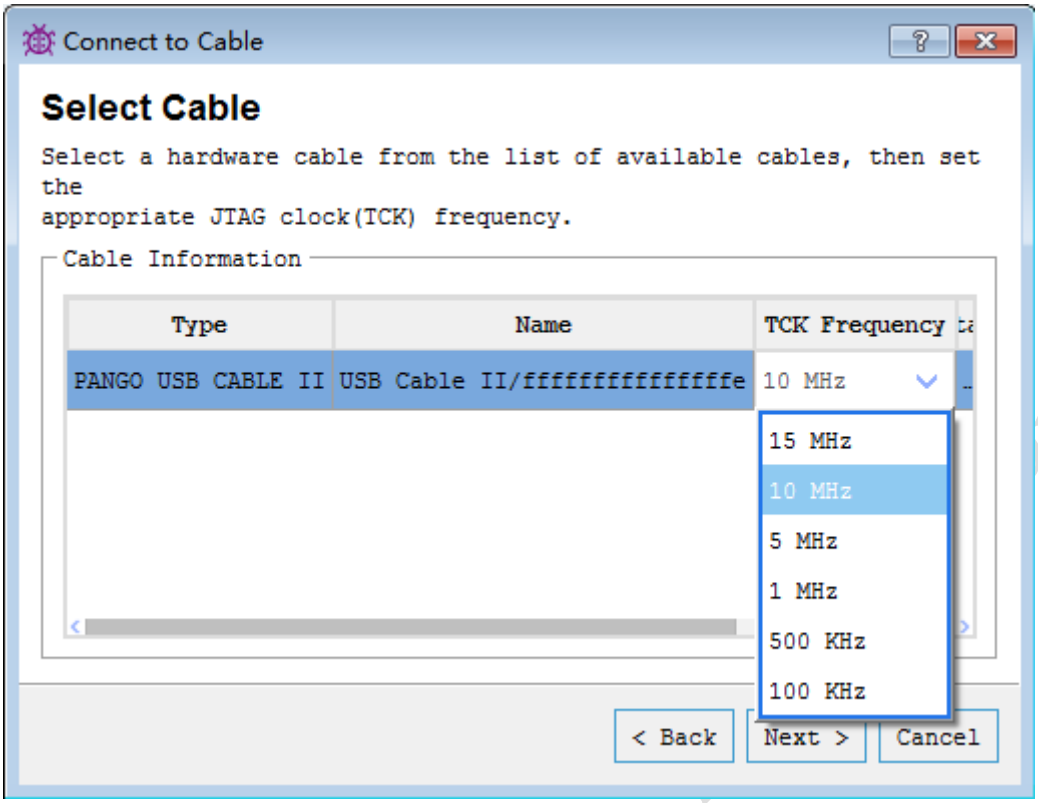


Figure9-12

9.2.3 Scan Device

Click the button on the toolbar to scan devices. The software supports automatically connecting to the server for device scanning, and the scanned devices are shown as in the figure below.

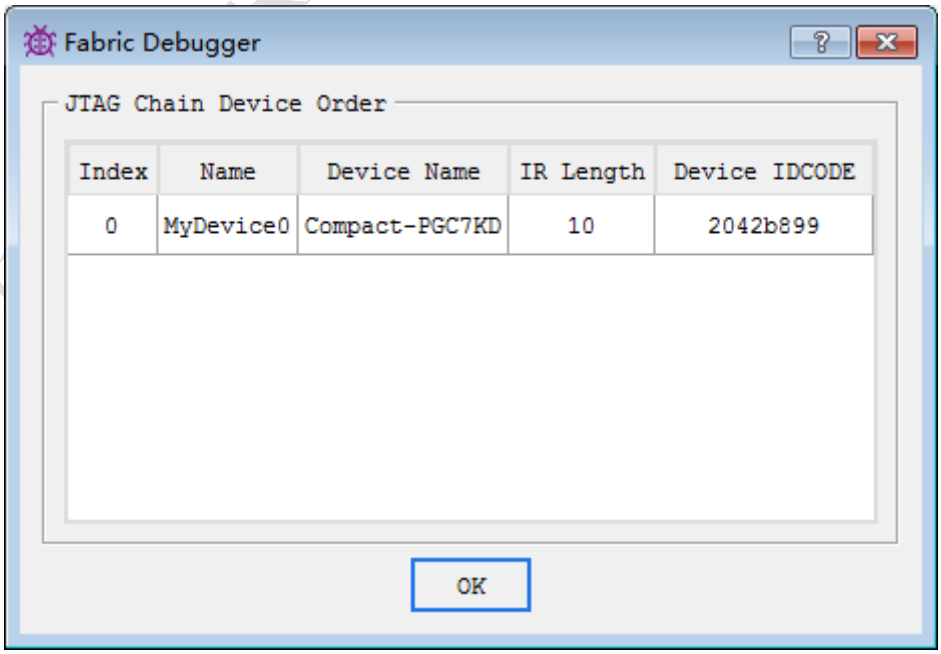


Figure9-13

9.2.4 Download Bitstream File

After device scanning is completed, click the button on the toolbar to pop up the interface for assigning the bitstream file, as shown in the figure below.

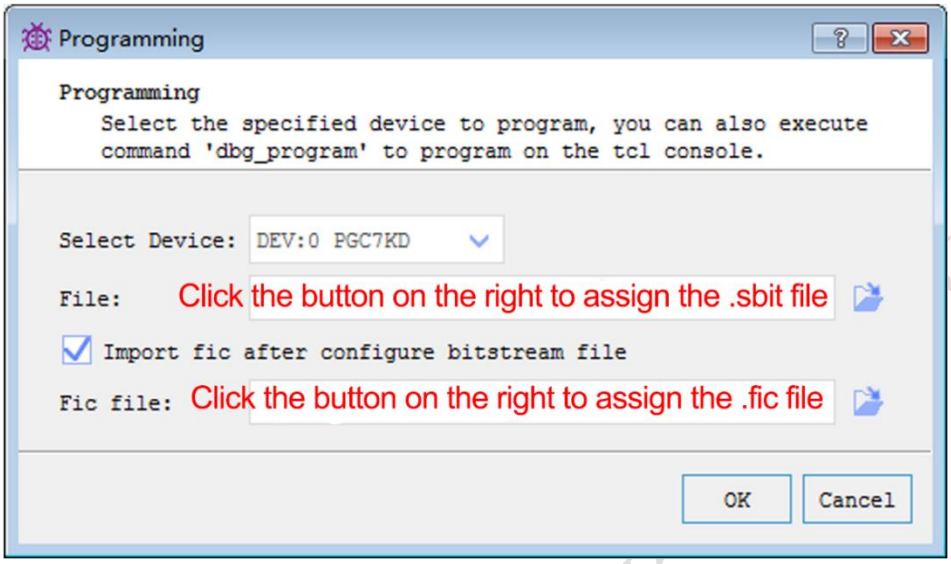


Figure9-14

Select [Import fic after configure bitstream file], and the associated .fic file will be automatically selected when the .sbit file is assigned. Click [OK].

9.2.5 Capture Waveforms and Debug

The main interface of the Fabric Debugger is shown in the figure below.

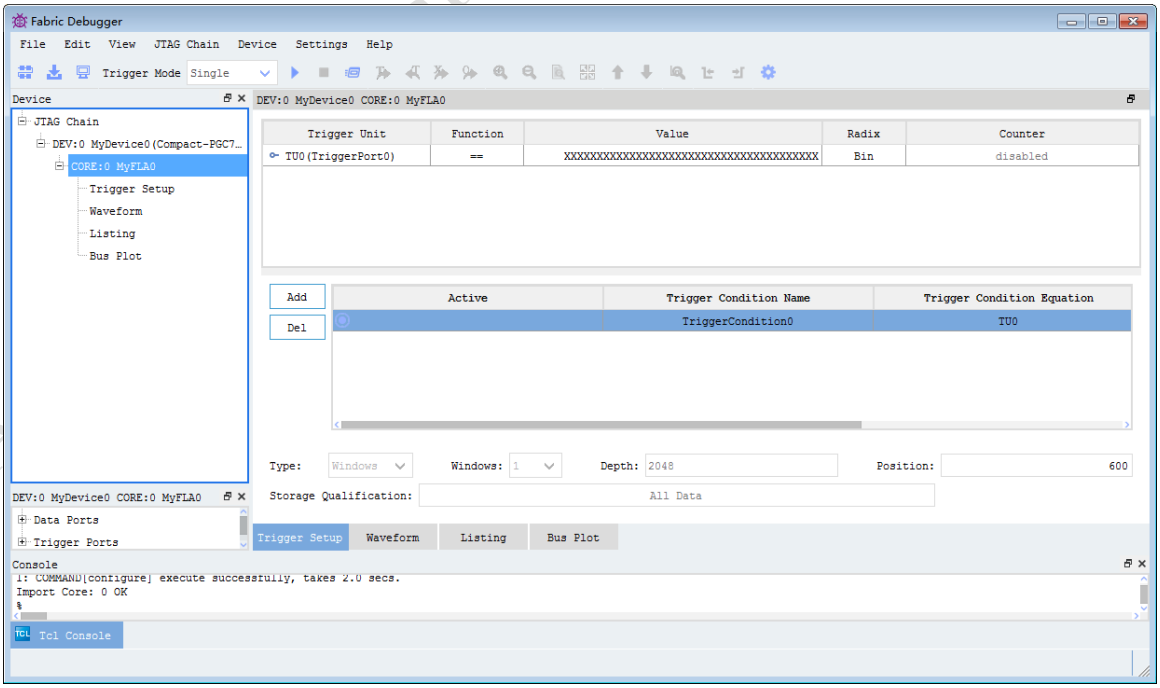


Figure9-15

Users can manipulate the waveforms through the buttons on the toolbar as needed. The toolbar is

shown below.

Figure9-16

Below is a description of the toolbar buttons (from left to right):

[Search JTAG Chain]: Automatically scans the JTAG chain and query the devices on the chain;

[Configure Bitstream File]: Imports the configuration file;

[Connect to Server]: Connects to the server;

[Trigger Mode]: Sets the trigger mode: single trigger or continuous trigger;

[Run]: Captures waveforms according to DebugCore configuration conditions;

[Stop]: Stops capturing waveforms;

[Trigger Immediately]: Captures waveforms directly without reading DebugCore configurations;

[Go To X Cursor]: Positions the interface to the X cursor;

Go To O Cursor]: Positions the interface to the O cursor;

[Go To Next Trigger]: Positions the interface to the next trigger point;

[Go To Previous Trigger]: Positions the interface to the previous trigger point;

[Zoom In]: Enlarges the interface;

[Zoom Out]: Shrinks the interface;

[Zoom Fit]: Displays waveforms at an appropriate scale;

[Zoom Full]: Displays all waveforms;

[Move Up]: Moves the selected bus and signals up in the Waveform;

[Move Down]: Moves the selected bus and signals down in the Waveform;

[Analyse Wave]: Analyzes statistically waveform data;

[Previous Transition]: Jumps to the previous value different from that at the current position;

[Next Transition]: Jumps to the next value different from that at the current position;

[System Settings]: System settings menu.

The captured waveforms are shown in the figure below.

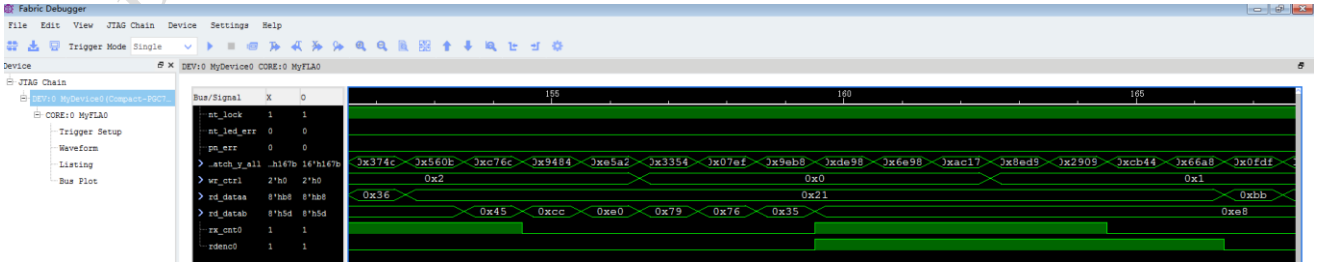


Figure9-17

Additionally, Fabric Debugger supports several convenient operations:

- Left-clicking and dragging over a certain area will zoom in on that area's waveform.

- Using "Ctrl"+mouse wheel will zoom in or out on the waveform centred around the mouse cursor.

Chapter 10 Power Consumption Evaluation

Pango Power Planner (abbreviated as PPP) is a power consumption evaluation software designed for evaluating the power consumption of users' CPLD designs, typically used before the completion of a design or the implementation of a project. PPP can assist users in evaluating the overall architecture of their design, selecting devices, designing power supplies, and creating cooling solutions.

PPP estimates power consumption based on the usage of resources in the user's design, the toggle rate, the load on I/O, and other factors, in conjunction with the power model of the target device.

Click  on the toolbar, or select [Tools > Power Planner] to launch PPP. The interface is shown in the figure below.

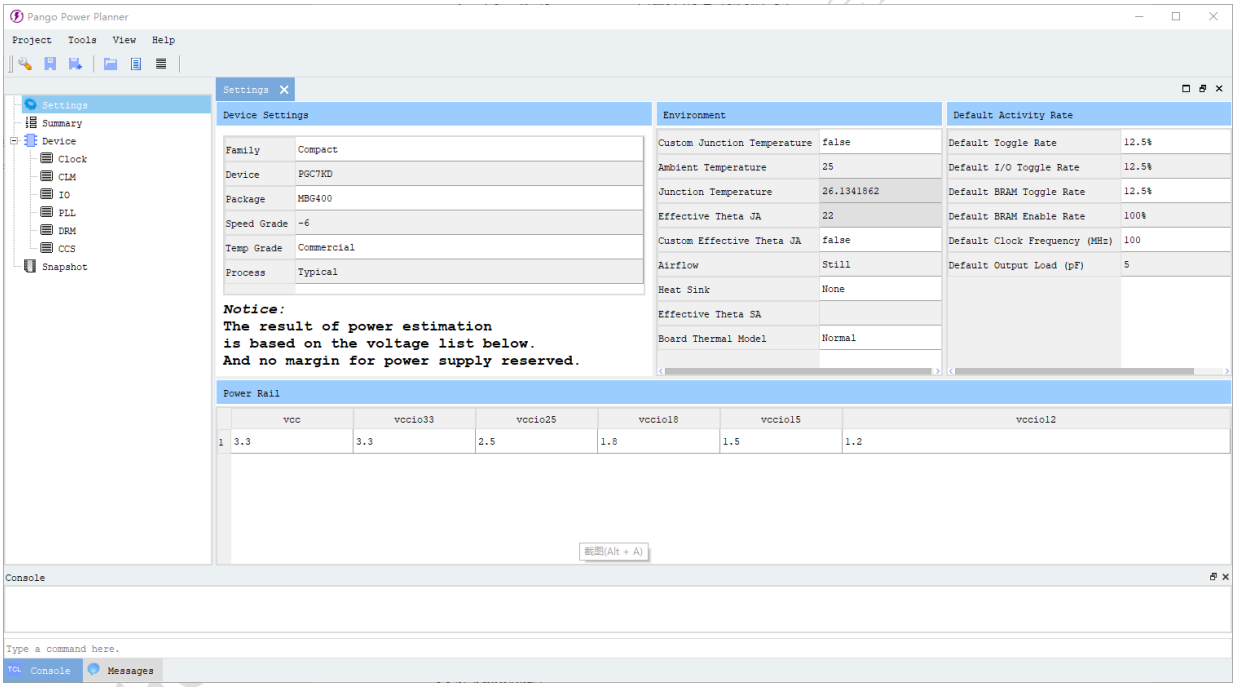


Figure10-1

10.1 New Project

Click the [New Project] option under the [Project] menu, or click the corresponding button on the toolbar to create a power consumption evaluation project. If a project is already open, the software will prompt whether to save the current project before proceeding to the [Device Setting] window. This is shown in the following figure.

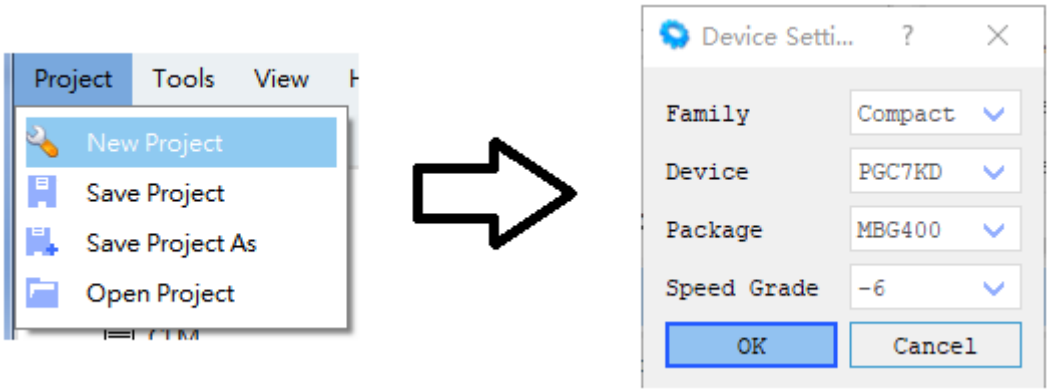


Figure10-2

Configure the [Family], [Device], [Package], and [Speed Grade], and then click [OK]. At this point, the software will clear all previous configuration information and reload the interface.

10.2 Save Project File

Click the [Save Project] option under the [Project] menu, or click the corresponding button on the toolbar, and the current settings and user-input data will be saved to the project file. This is shown in the following figure.

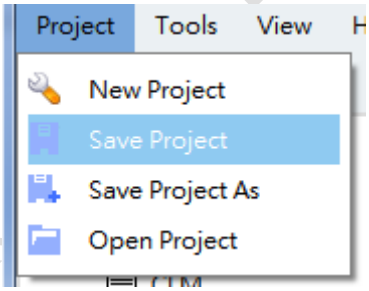


Figure10-3

10.3 Save as Project File

Click the [Save Project As] option under the [Project] menu, or click the corresponding button on the toolbar to open the window for file saving. Enter the file name and click Save, and all current data will be saved to the project file. This is shown in the following figure.

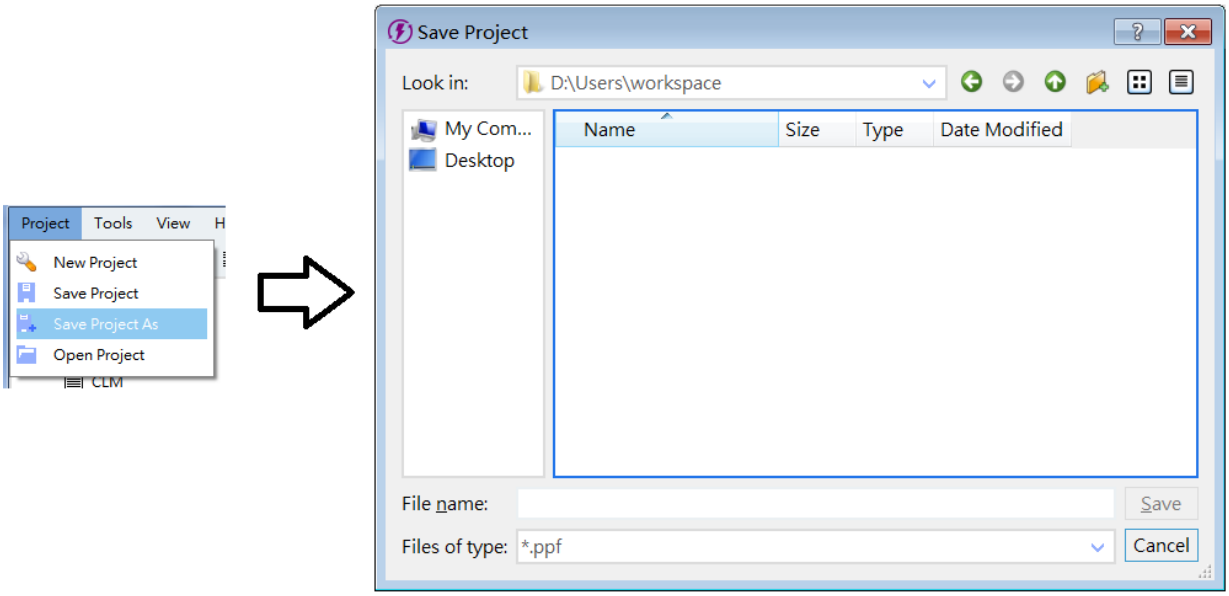


Figure10-4

10.4 Open Project

Click the [Open Project] option under the [Project] menu, or click the corresponding button on the toolbar to pop up the dialogue for file opening. Select the .ppf project file and click Open, and the data saved in the file will be loaded into PPP. This is shown in the following figure.

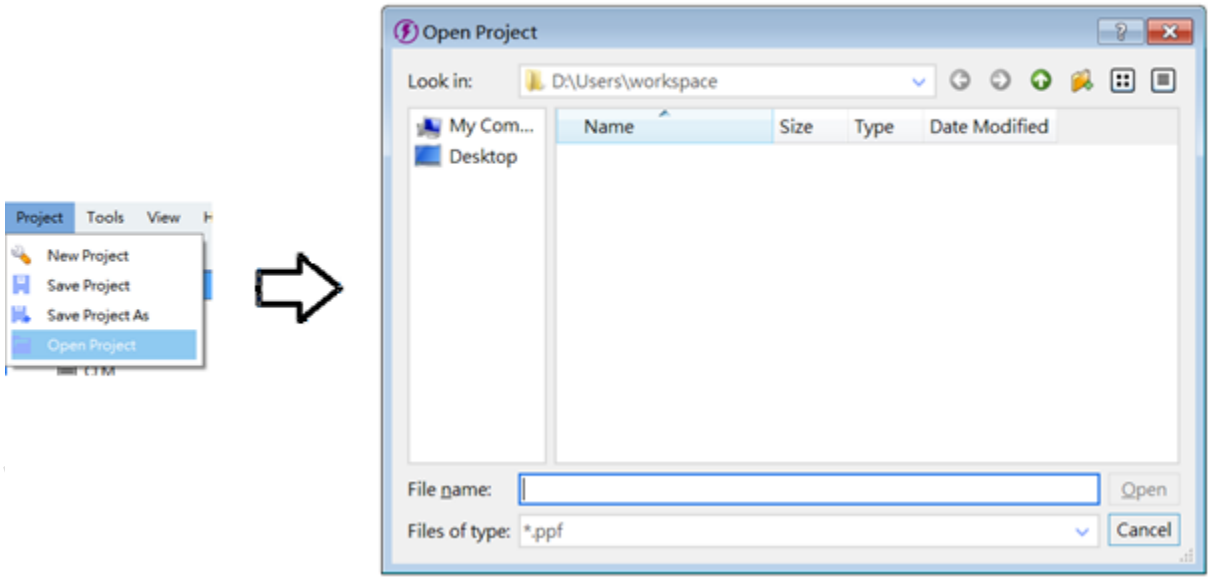


Figure10-5

10.5 Export Parameters Settings

Click the [Export Settings] option under the [Tools] menu, or click the corresponding button on the

toolbar to open the window for file saving. Enter the file name and click [Save], and the device, environment, and voltage parameters related to PPP will be saved to the file. This is shown in the following figure.

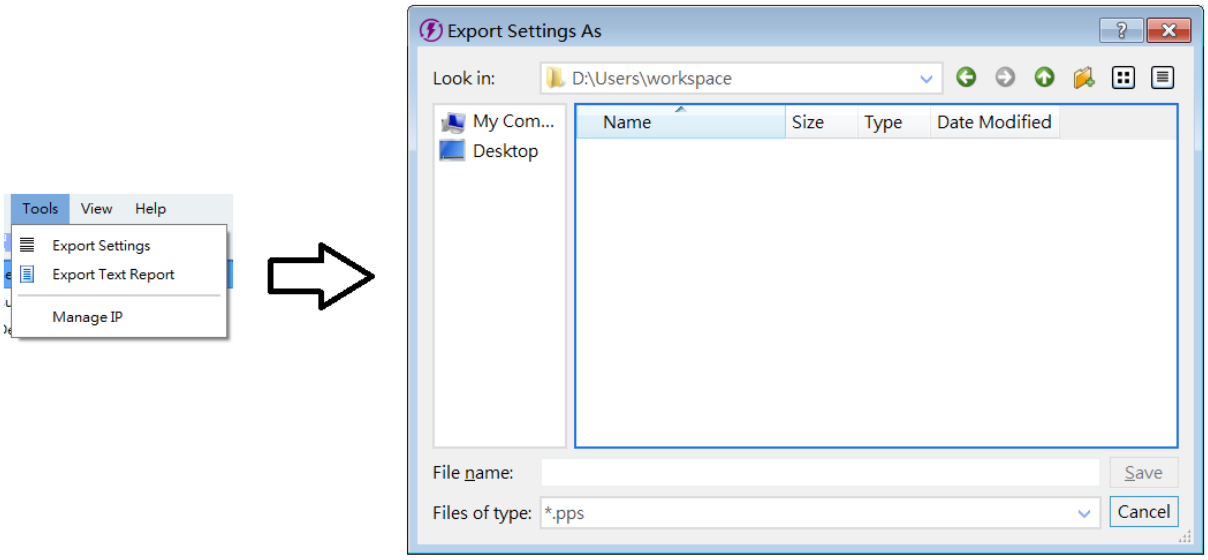


Figure10-6

10.6 Export Power Consumption Report

Click the [Export Text Report] option under the [Tools] menu, or click the corresponding button on the toolbar to open the dialogue for file saving. Entering the file name and click [Save], and the power consumption data and related configuration information in PPP will be exported to the report file. This is shown in the following figure.

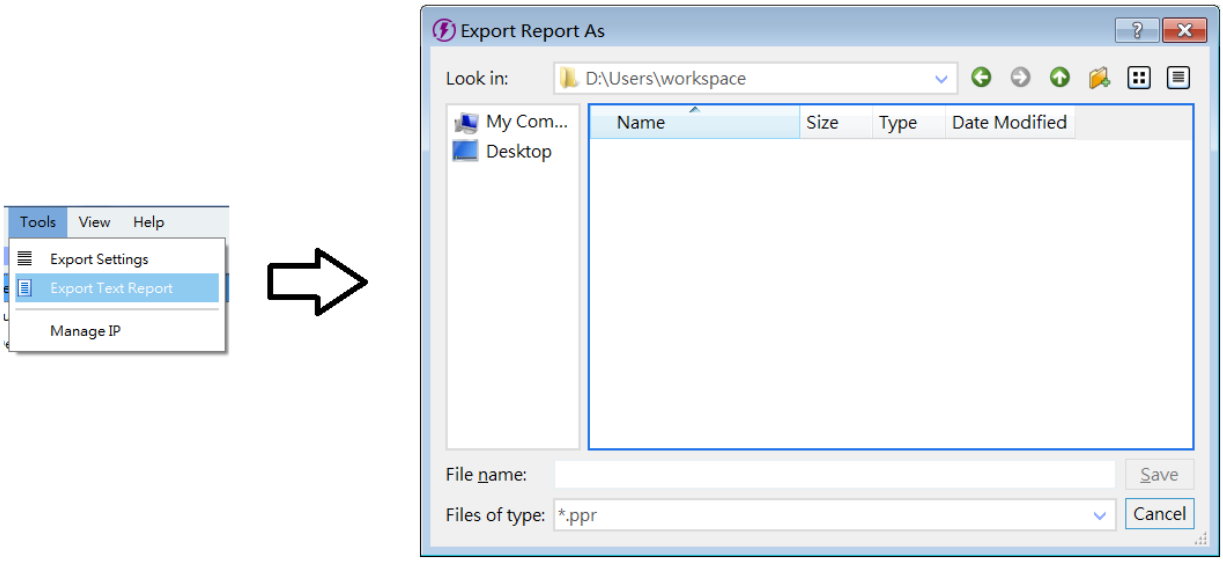


Figure10-7

10.7 Environment Parameters Settings

On the left of PPP's configuration interface, click [Setting], and the [Settings] page will appear on the right, where users can configure environmental parameters. [Settings] page includes four areas: [Device Settings], [Environment], [Default Activity Rate], and [Power Rail]. This is shown in the following figure.

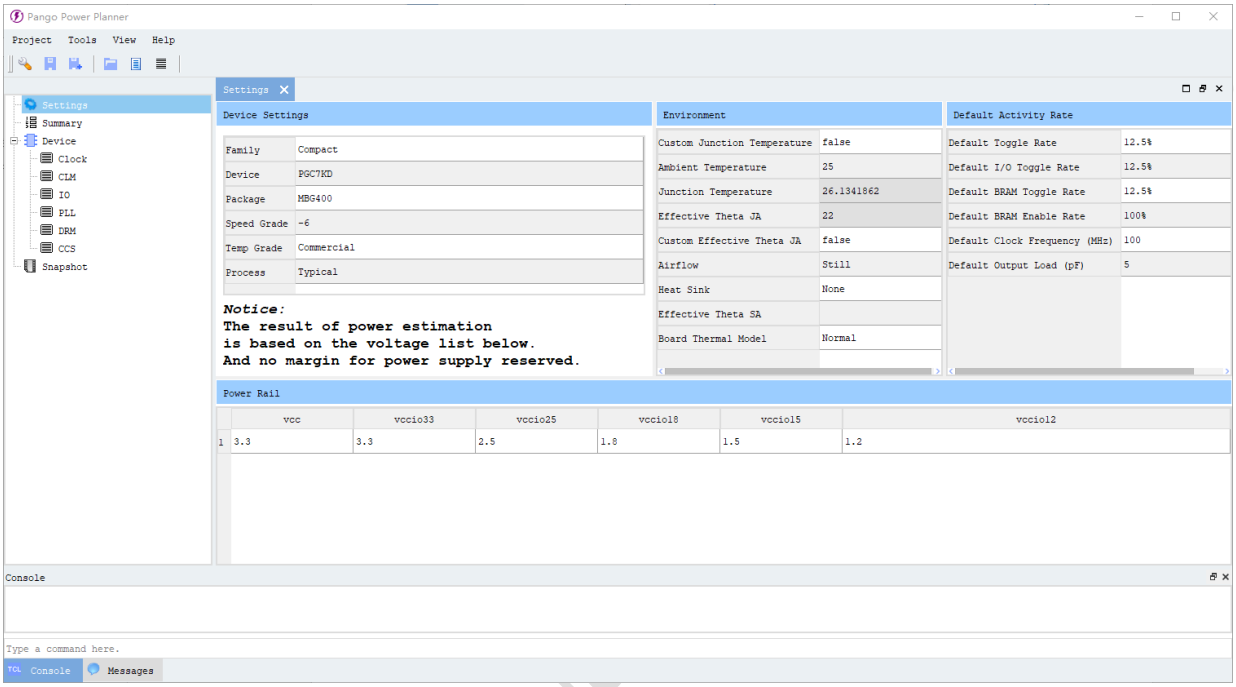


Figure10-8

[Device Settings] area displays detailed information of the current device, which can be edited as needed;

The [Environment] area displays parameters related to the device's operating environment, where junction temperature, ambient temperature, airflow rate, type of heat sink, and other parameters can be set. Among them:

1. [Custom Junction Temperature]: Specifies whether to set the junction temperature. When true is selected, users can specify the [Junction Temperature] and other options cannot be modified; when false is selected, the default value is 25.0;
2. [Ambient Temperature]: Enters the ambient temperature, which is determined by the [Temp Grade] setting in the [Device], with a range of (0~85) for Commercial and (-40~100) for [Industrial];
3. [Junction Temperature]: The default value is 25.0, determined by the [Temp Grade] setting in the [Device], with a range of (0~85) for [Commercial] and (-40~100) for [Industrial];
4. [Effective Theta JA]: The effective thermal resistance to air, i.e., the thermal resistance coefficient for heat dissipation through air;

5. [Custom Effective Theta JA]: Specifies whether to set the heat dissipation coefficient. If false is selected, users can modify the default thermal coefficient;
6. [Airflow]: The status of airflow, with four options available: [Still], [Custom], [Low], [Medium], and [High];
7. [Heat-Sink]: The status of the heat sink, with 5 options available: [None], [Custom], [Low], [Medium], and [High];
8. [Effective Theta SA]: The thermal resistance coefficient for heat dissipation through the heat sink, which can only be modified when Heat-Sink is set to [Custom];
9. [Board Thermal Model]: The board's heat dissipation condition, with four options: JEDC, Best, Normal, and Worst;

[Default Activity Rate] area allows setting the toggle rate and operating clock frequency for various resources;

[Power Rail] area allows configuring the voltage for different power supplies, and PPP estimates the chip's power consumption under various power supply voltages based on these settings.

When setting these parameters, hovering the mouse over the editing window will pop up a tooltip, which displays the valid input range for that data.

10.8 Power Consumption Data Input

[Device] contains multiple modules. Clicking any of them will open a separate configuration interface. Each interface provides a right-click context menu, which offers options to add a row of data, add multiple rows, remove a row, and remove multiple rows.

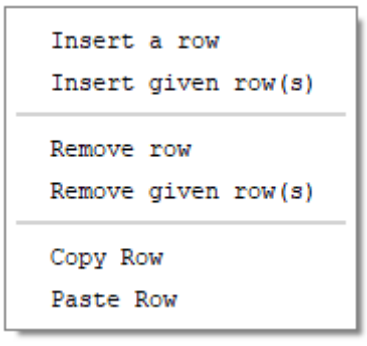


Figure10-9

10.8.1 Clock Configuration

Click [Clock] under [Device] to pop up the Clock configuration interface.

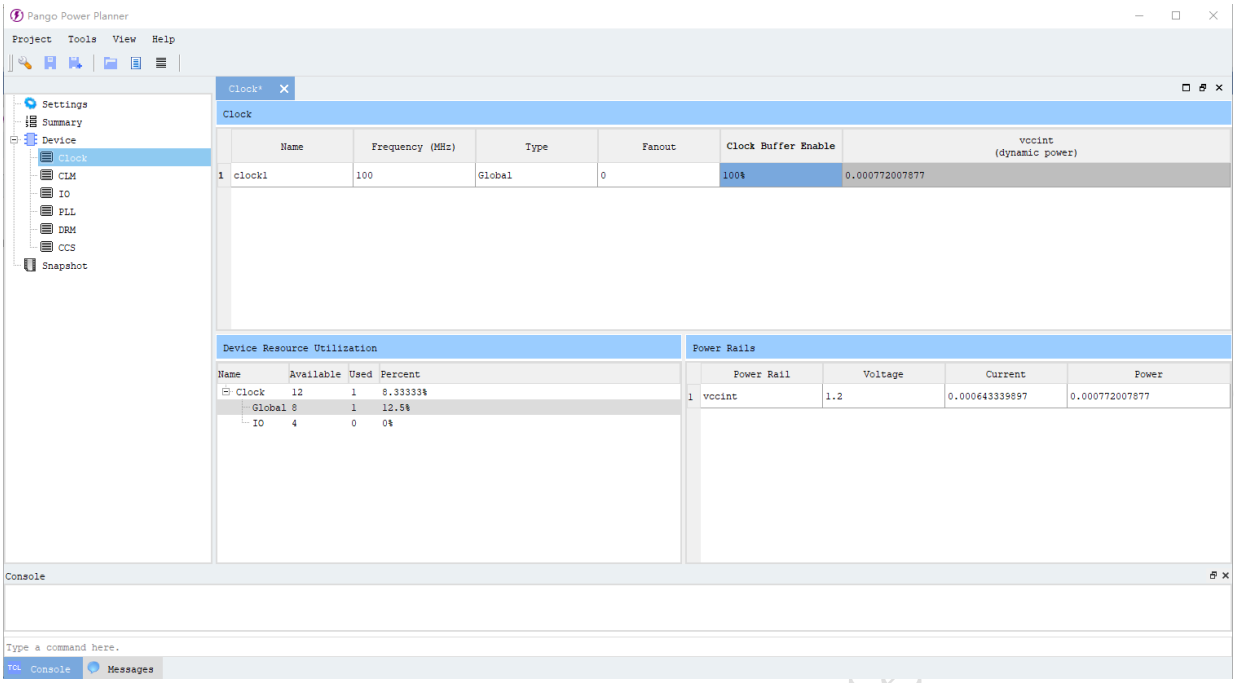


Figure10-10

In the [Clock] area, users can configure various clock parameters, and the dynamic power consumption is displayed in real time. Typically, the higher the frequency of the Clock, the greater the dynamic performance of the circuit and the higher the power consumption. The larger the Fanout (average fan-out), the more loads driven, the larger the load capacity, and the higher the power consumption.

[Clock Type] is divided into [Global Clock] and [I/O Clock]. Selecting different types of clocks will result in different calculated power consumptions. [Clock Buffer Enable] indicates the probability that the clock buffer is in an enabled state. Only when the clock type is [Global] will [Clock Buffer Enable] affect power consumption; the higher the enable rate of [Clock Buffer Enable], the higher the power consumption.

[Device Resource Utilization] area displays the usage of Clock resources; [Power Rails] area displays the dynamic power consumption categorized by power supply.

10.8.2 CLM Configuration

Click CLM under [Device] to pop up the CLM configuration interface.

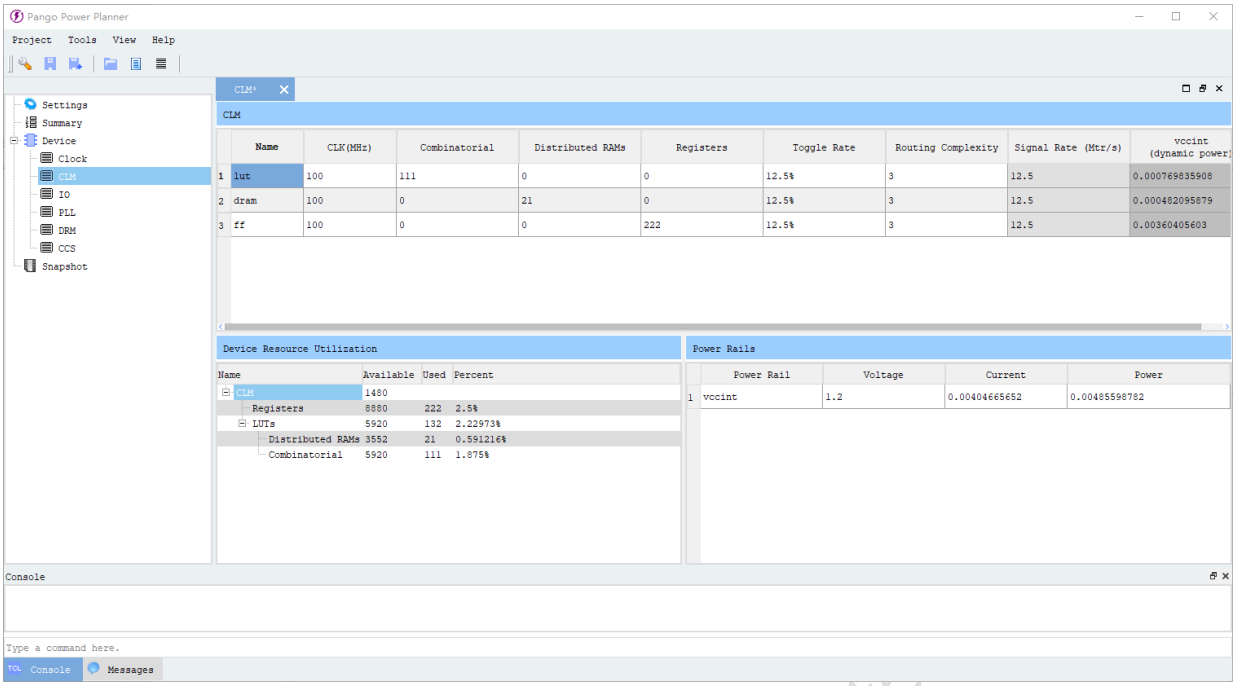


Figure10-11

Here, the power consumption of LUTs, FFs, and Distributed RAMs is primarily assessed. In the [CLM] area, users need to enter the resource usage, operating clock frequency (CLK), toggle rate, and routing complexity.

[Routing Complexity] refers to the average number of routing resources for each logic cell, which is quantified by factors such as routing length, number of loads, and the level of routing congestion. The load here refers to the direct fanout and indirect fanout of routing (the number of SRBs passed by the Routing driver). Specifically, the greater the number of routing resources required by a logic cell and the higher level of the congestion, the greater the value of Routing Complexity.

[Toggle Rate] indicates the clock's toggle rate. The faster the clock toggles, the greater the dynamic power consumption.

Please note that in PPP, routing is estimated, and the most accurate routing information can only be obtained after placement and routing. When conducting an evaluation, users can increase the routing complexity appropriately based on the scale and complexity of the design. Moreover, as an evaluation tool, PPP's routing estimation should have a margin, making it difficult to be very precise. Therefore, users can generally take the default value.

[Signal Rate] defines the signal toggle rate, which is calculated through the Toggle rate and Clock frequency. The formula is as follows:

Signal Rate (Mtr/s) = Clock Frequency (Mhz) * Toggle rate (%)

The larger the Signal Rate, the higher the dynamic power consumption.

10.8.3 IO Configuration

Click [IO] under [Device] to pop up the IO configuration interface.

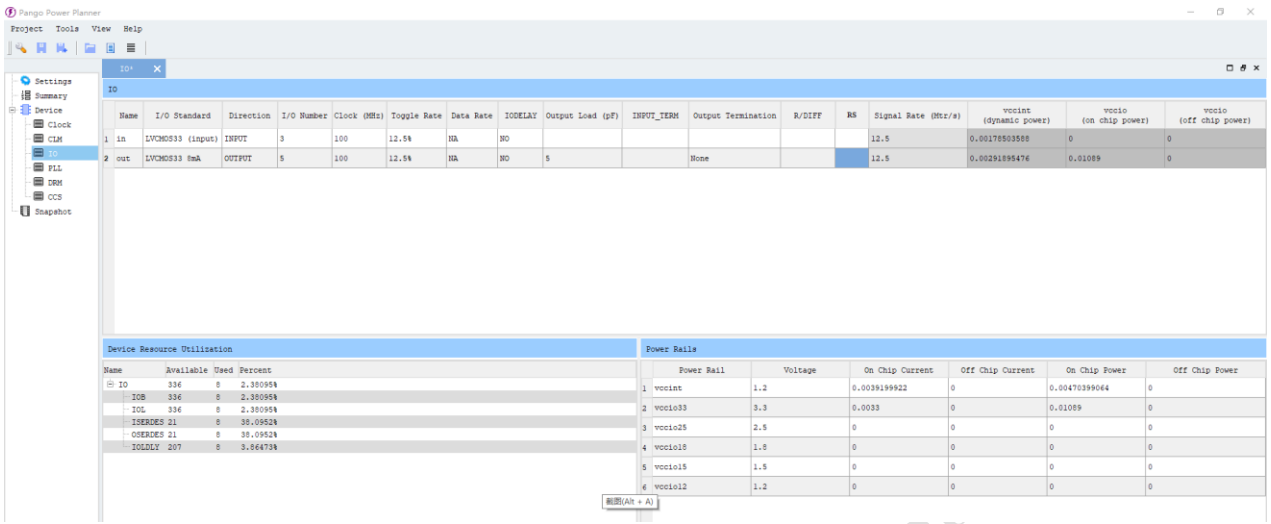


Figure10-12

The IO configuration interface estimates not only the power consumption of the IO itself but also the power consumption of the external circuits driven by the IO. IO supports multiple voltage standards: 1.2V, 1.5V, 1.8V, 2.5V, and 3.3V.

Please note that when the IODELAY option is set to [YES], the dynamic power consumption of the vccint power supply increases. Output Load only affects the size of the vccio power consumption; when the other parameters are fixed, the larger the Output Load, the greater the vccio power consumption. The Output Termination, RS, and R/DIFF parameters are related to external power consumption (off-chip power). Under different connection modes, the enabled resistance differs. For example, in series, the external power consumption decreases as RS increases; in parallel, it decreases as R increases; in series-parallel, external power consumption decreases as RS and R increase. Only some protocols have external power consumption (e.g., HSTL and SSTL).

10.8.4 PLL Configuration

Click [PLL] under [Device] to pop up the PLL configuration interface.

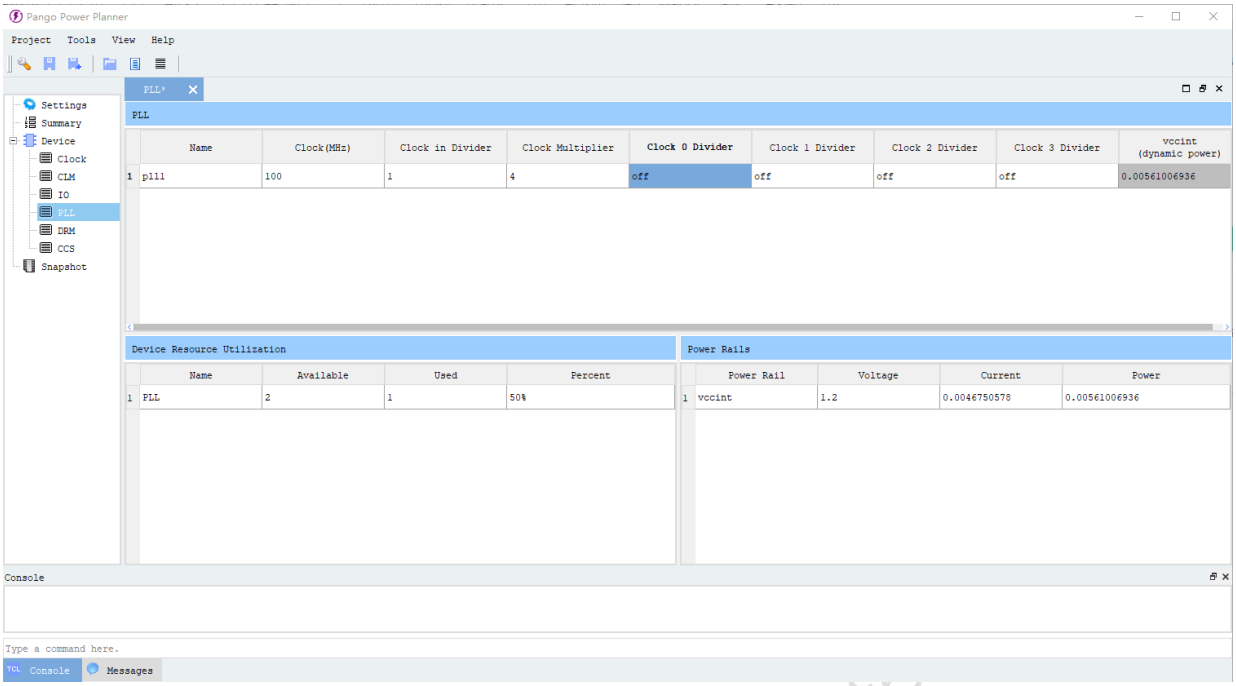


Figure10-13

Clock is the input reference clock frequency, Clock in Divider is the division ratio of the input divider, and Clock Multiplier is an abstract parameter obtained by multiplying the feedback division ratio (i.e., FDIV) and the ODIV on the feedback channel, which needs to be calculated and filled in. Since PPP does not support the selection of feedback channels and ODIV cascading is not considered, the feedback channel is defaulted to internal feedback for evaluation. At this time, the VCO frequency Fvco is calculated as Input Clock frequency / Clock in Divider * Clock Multiplier. Clock0/1/2/3 Divider is for selecting the output channels and the output division ratios (ODIV0/1/2/3), with each channel's output frequency Fout0/1/2/3 = Fvco/ODIV0/1/2/3.

10.8.5 DRM Configuration

Click [DRM] under [Device] to pop up the DRM configuration interface, where the power consumption of DRM resources is evaluated. It includes Port A and Port B, whose parameters can be configured separately. This is shown in the following figure.

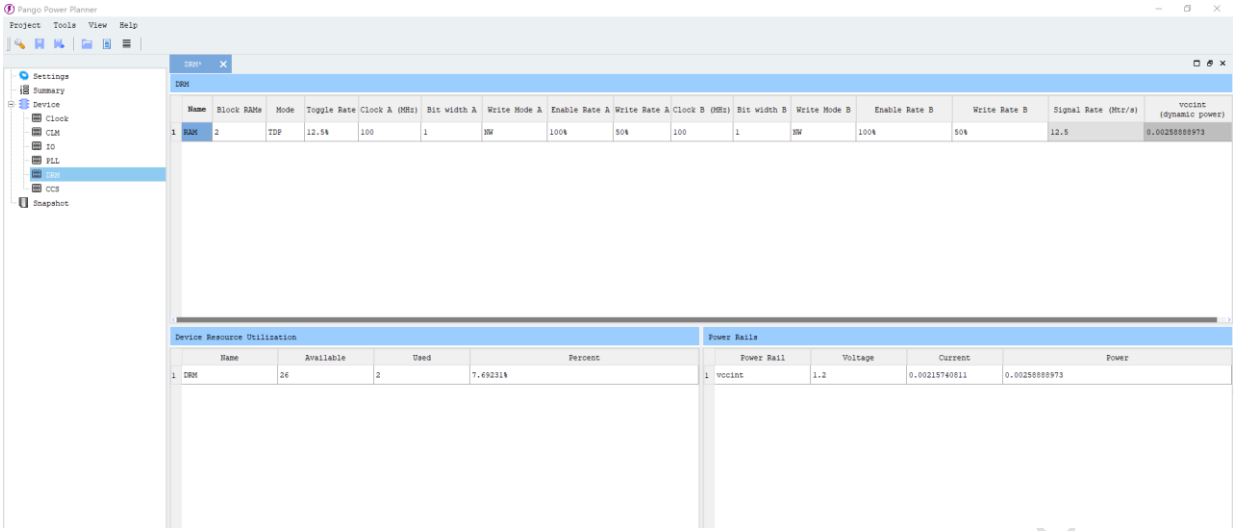


Figure10-14

[Block RAMs] indicate the number of DRMs used, which must be entered for power consumption estimation.

[Mode] indicates the configuration mode for the RAM, with five modes available: TDP, SDP, SP, ROM, and FIFO. Different modes offer different selectable Bit widths for Port A and Port B. Additionally, users need to configure the clock frequency (Clock), bit width (Bit Width), data write mode (Write Mode), port enable rate (Enable Rate), and data write enable rate (Write Rate) for Port A and Port B separately.

[Signal Rate] is read-only and is calculated based on the Toggle rate and Clock frequency. The formula is as follows:

When only Port A is configured, the formula is: Signal Rate = Clock A * Toggle Rate;

When only Port B is configured, the formula is: Signal Rate = Clock B * Toggle Rate;

When both Port A and Port B are configured, the formula is:

Signal Rate = (Clock A * Toggle Rate + Clock B * Toggle Rate) / 2;

In the same mode, an increase in the number of Block RAMs leads to higher power consumption. Likewise, higher clock frequencies (Clock) and Toggle rates result in increased dynamic power consumption. Wider bit widths also contribute to greater dynamic power usage, as do higher port enable rates (Enable Rate) and data write enable rates (Write Rate).

Write modes include NW (Normal Write mode), TW (Transparent Write), and RBW (Read before Write). With other parameters fixed, NW has the least dynamic power consumption, while RBW has the most.

10.8.6 CCS Configuration

Click [CCS] under [Device] to pop up the CCS configuration interface.

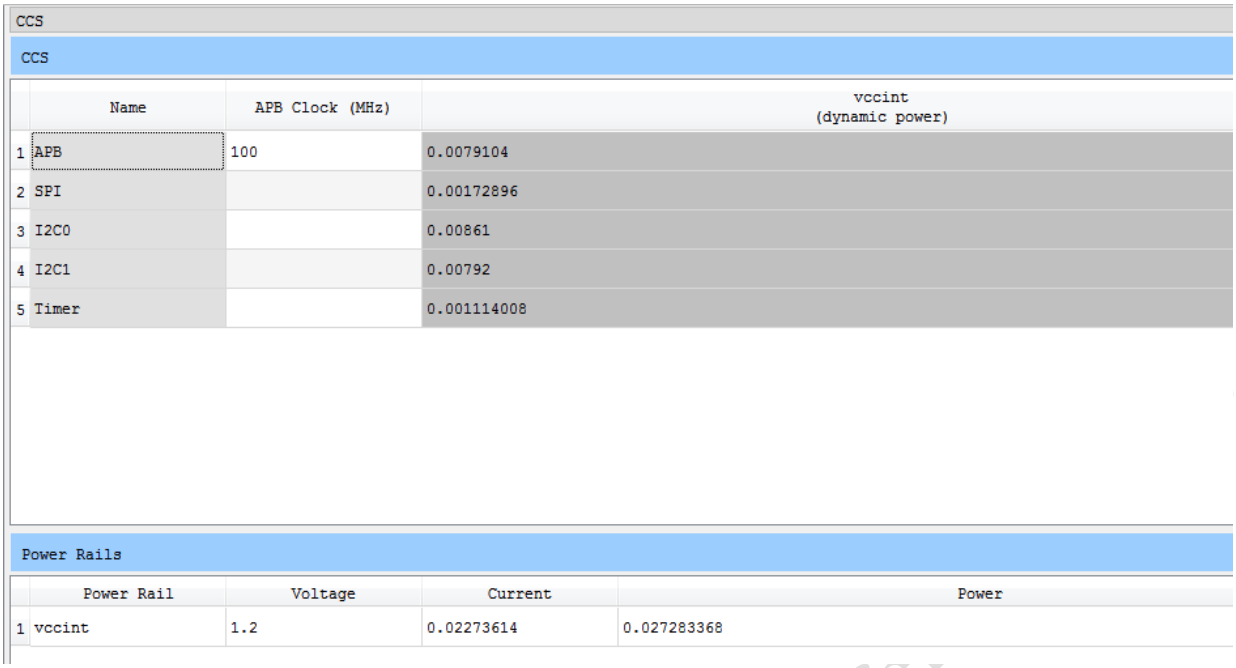


Figure10-15

The CCS configuration interface assesses the power consumption of the PGC embedded hard cores, including I2C, SPI, APB, and Timer.

10.8.7 Summary

Click [Summary] to pop up the power consumption evaluation report interface, as shown in the figure below.

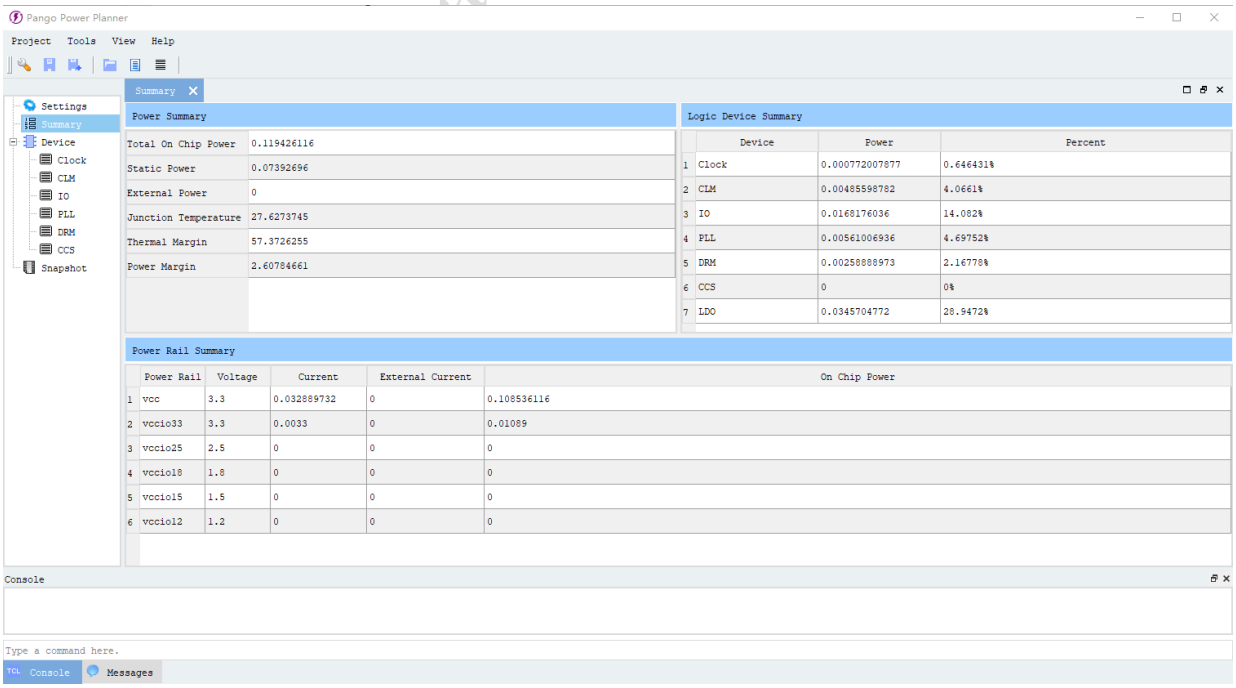


Figure10-16

Please note that [External Current] is the current supplied by the chip power through the I/O port to

external circuits; the larger this current value, the higher the resulting external power consumption (Off-Chip Power). The value is 0 by default.

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides information in stages and may be updated at any time based on the actual situation of the products without further notice. The Company assumes no legal responsibility for any direct or indirect losses caused by improper use of this document.
2. This document is provided "as is" without any warranties, including but not limited to warranties of merchantability, fitness for a particular purpose, non-infringement, or any other warranties mentioned in proposals, specifications, or samples. This document does not grant any explicit or implied intellectual property usage licence, whether by estoppel or otherwise.
3. The Company reserves the right to modify any documents related to its family products at any time without prior notice.
4. The information contained in this document is intended to assist users in resolving application-related issues. While we strive for accuracy, we cannot guarantee that the document is entirely free from flaws. Should any functional abnormalities and performance degradation arise due to deviation from the prescribed procedures outlined herein, our company will neither be held liable nor concede that such issues stem from product deficiencies. The solutions presented in this document are just one of the feasible options and cannot cover all application scenarios. Consequently, if users encounter functional abnormalities or performance degradation despite adhering to the prescribed procedures outlined herein, we cannot assure that such issues are indicative of product deficiencies.