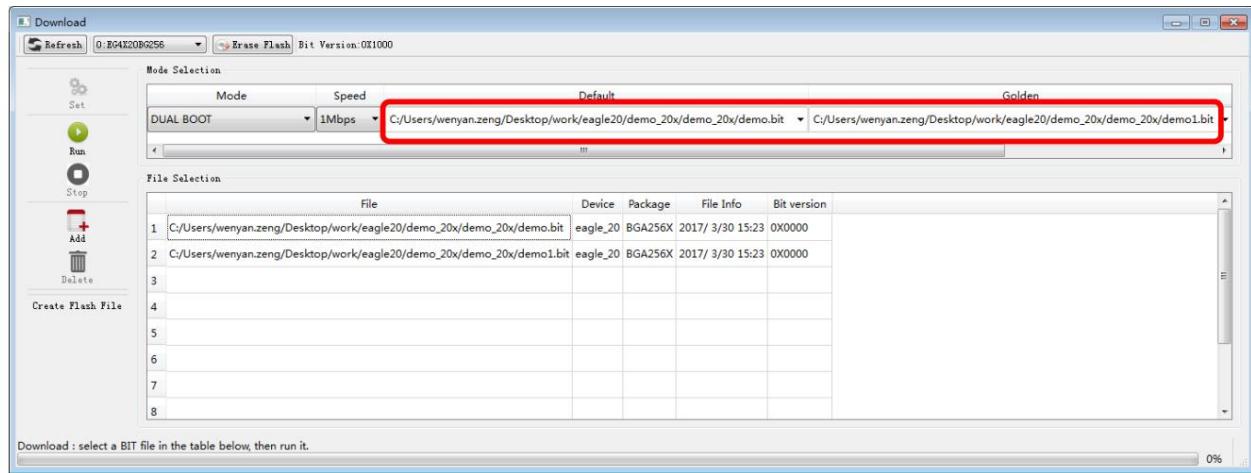
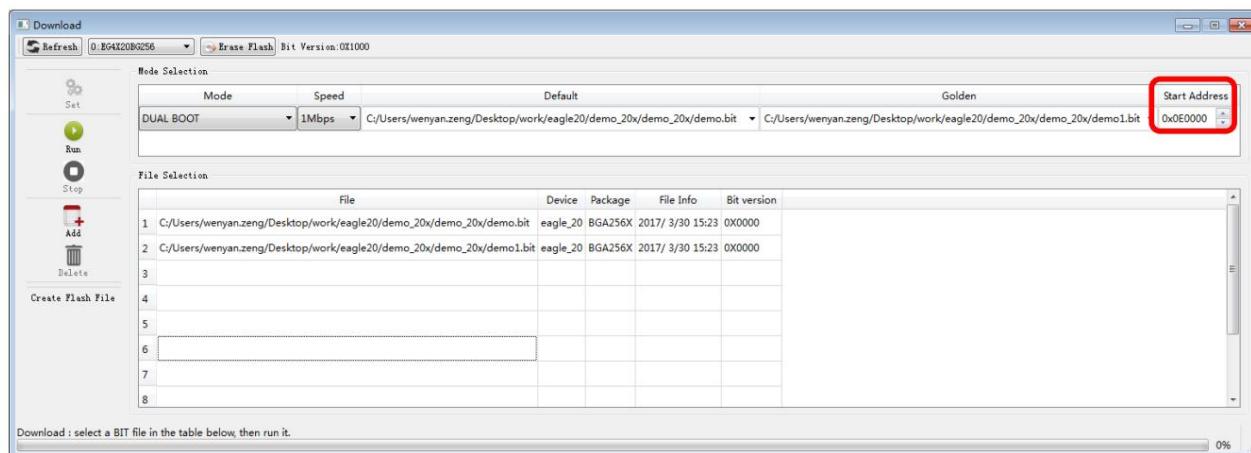


3. Set the **Default** bit stream (Primary address segment) and the **Golden** bit stream (Golden address segment);



4. Set the starting address where the Golden bit stream is stored. Note that the Golden storage address can only be greater than 0X0D0000,

The default is: 0X0E0000;



5. Click **Run** to download the bitstream.

7.3.2 Multi Boot

The multi-boot mode means that the user can store two or more sets of FPGA bit streams in the SPI FLASH.

The FPGA first loads the Primary bitstream, and then in the FPGA code of the Primary bitstream, the FPGA can be controlled from

Load the bitstream at the specified address. In multi-start mode, the data space distribution is shown in the following figure:



Before using multiboot mode, the user needs to call the following IP units in the code:

```
EG_LOGIC_MBOOT #("DYNAMIC",8'h00) mboot(rebootn, dynamic_addr);
```

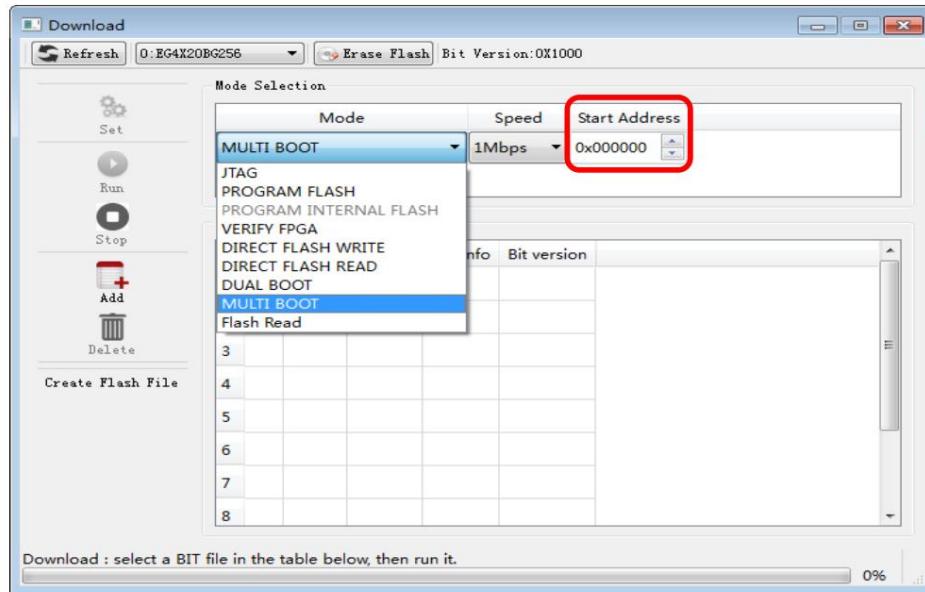
Among them, dynamic_addr is the 8-bit FLASH address, which is the upper 8 bits of the 24-bit FLASH address.

After the address is set, by giving rebootn a low pulse, it can be reloaded from the specified address of dynamic_addr

FPGA program.

In the TD Download interface, you need to set the download mode to **MULTI BOOT**, and specify the bit used for jumping

The **Start Address** of the file , which must be the same as dynamic_addr.



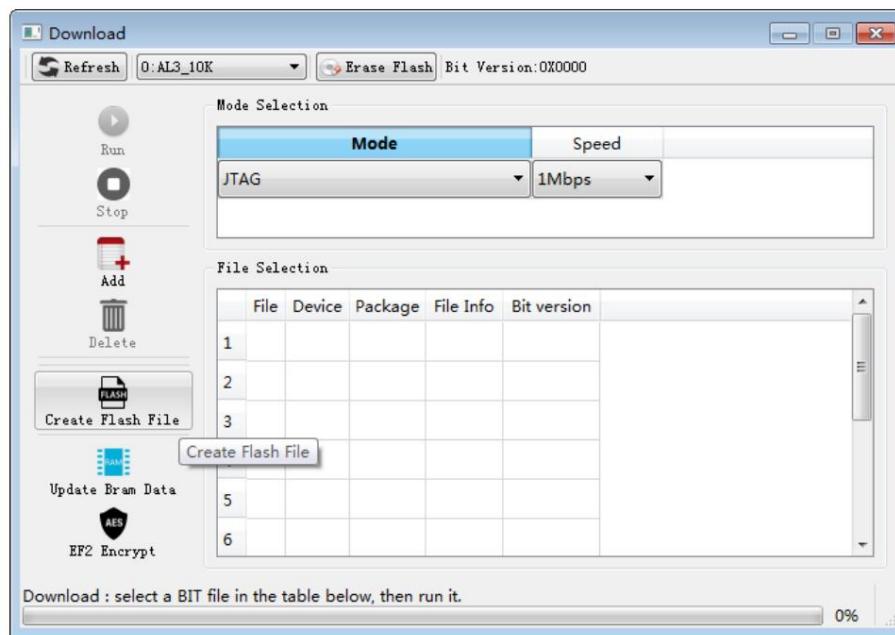
7.4 Extended functions

7.4.1 Create Flash File

TD software provides the Create Flash File function, which is convenient for users to add custom content in the target bitstream file.

Extending the functionality of existing bitstream files requires re-modifying the source code, saving a lot of time. The specific operations are as follows:

1. Open the **Download interface** and click "Create Flash File";

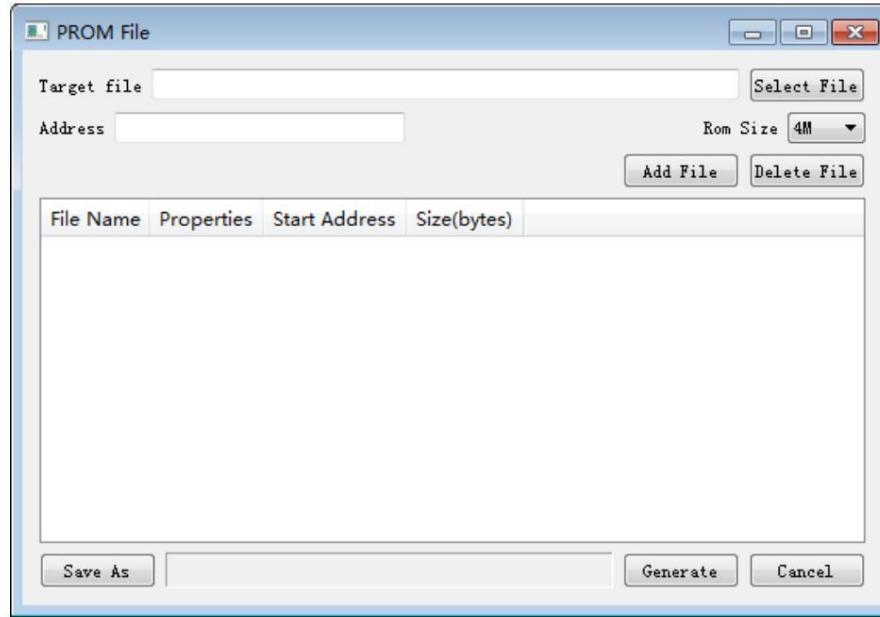


2. Click "Select File" to add Target File. The target file can be either a bit file or a bin file.

After adding the target file, the size of the file will be displayed accordingly, that is, **Address**.

Rom Size refers to the capacity of the target Flash, that is, the final generated file cannot exceed this capacity, otherwise

Generate will fail.



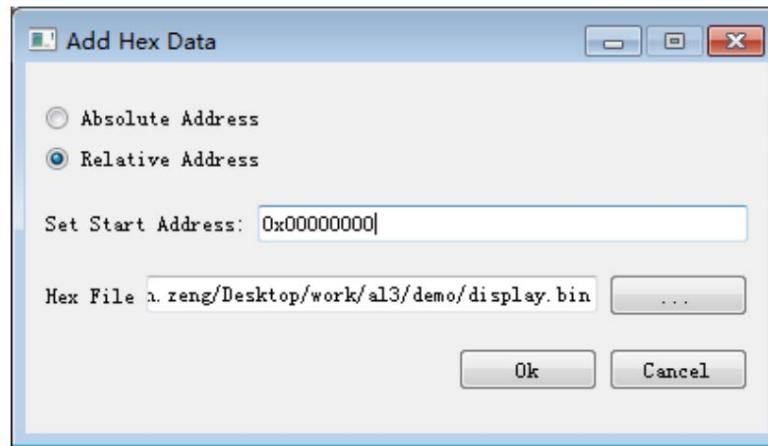
3. Click "Add File" to add a file, call the file a merged file, and the merged file can be a bin file, hex file

file or bit file. If you choose **Absolute** Address, you need to set the starting address to be greater than Target File

Otherwise, after adding the file, the content of the Target File will be overwritten; if you select Relative Address,

Refers to the size of the Target File, which is added after it. If the starting address is set to 0x00, it is followed by

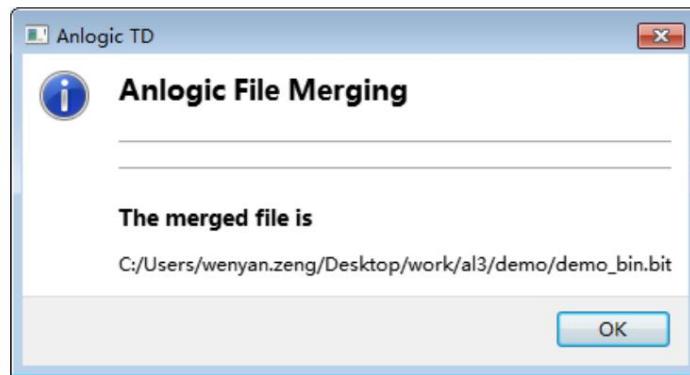
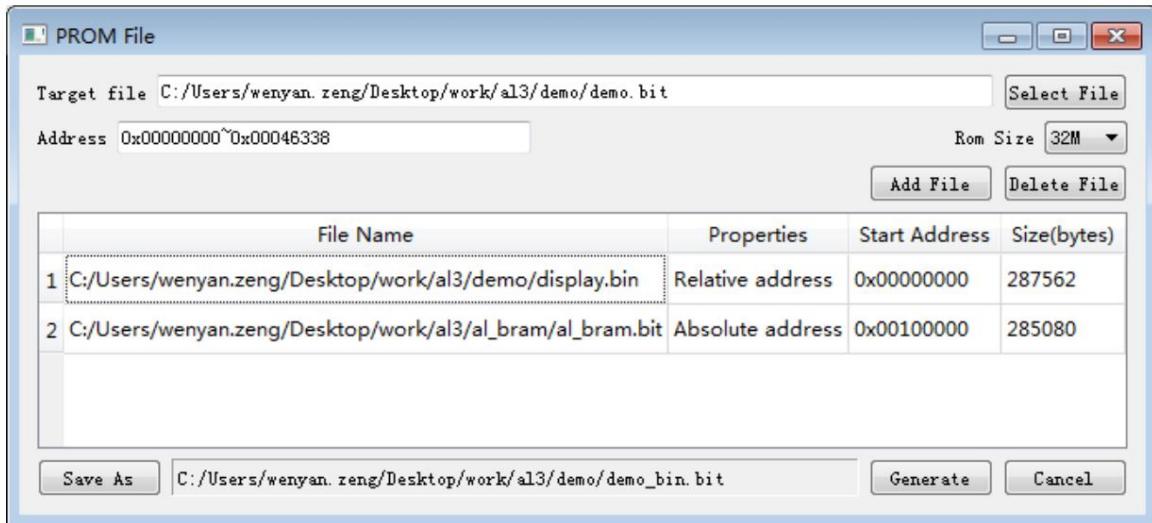
Add the file after Target File.



4. Multiple merged files can be added at the same time, just set the starting address reasonably according to the size of each file.

5. Select the path to generate the file, click Generate, the following prompt will be given, otherwise a corresponding error will be given.

wrong prompt.



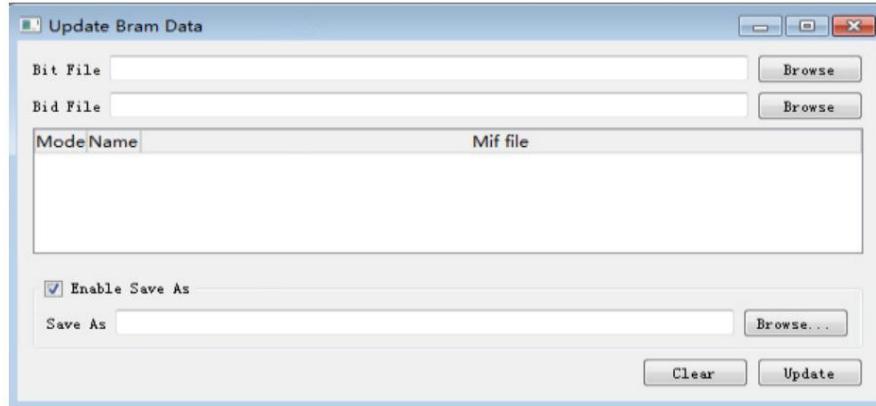
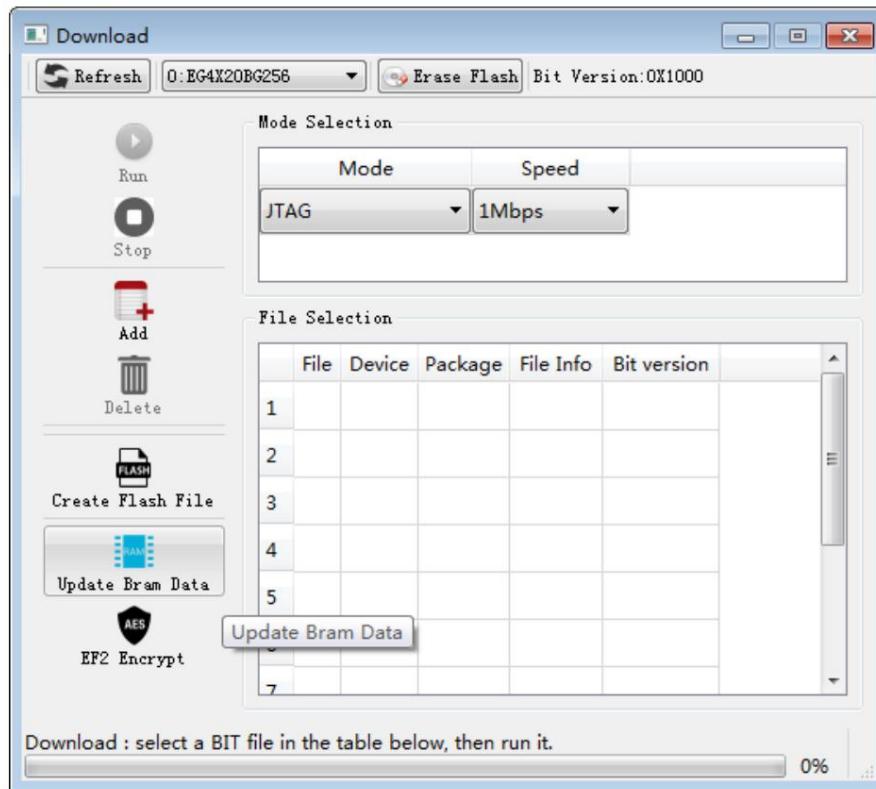
7.4.2 Update BRAM Data

TD provides the Update BRAM Data tool, if you only need to update the initial value of the BRAM in the design, there is no need to re-

For a new compilation project, directly modify the data segment of the BRAM in the bitstream file to generate a new bitstream file, saving a lot of money

amount of time. The specific operations are as follows:

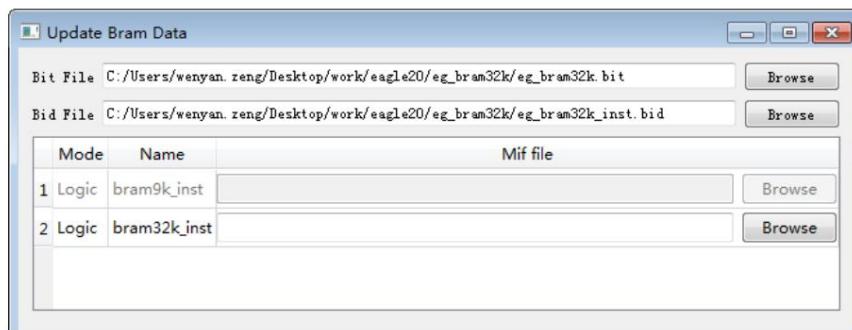
1. Open the Download interface and click "Update Bram Data";



2. Select the **Bit File** that needs to be updated and the **Bid File** describing the BRAM, and the bit stream will be displayed accordingly.

Logic BRAM. This can only be done here for BRAMs where initialization files have been added early in the design

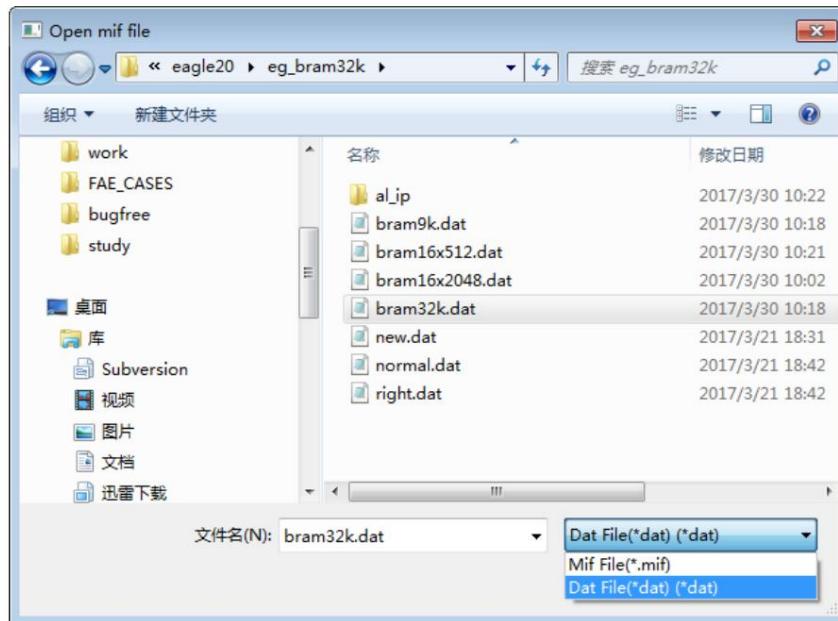
Update; the BRAM without the initialization file is displayed in gray, as shown in the following figure, bram9k_inst;



3. Click "Browse" to add new BRAM Data, the files that can be added are .dat files and .mif files.

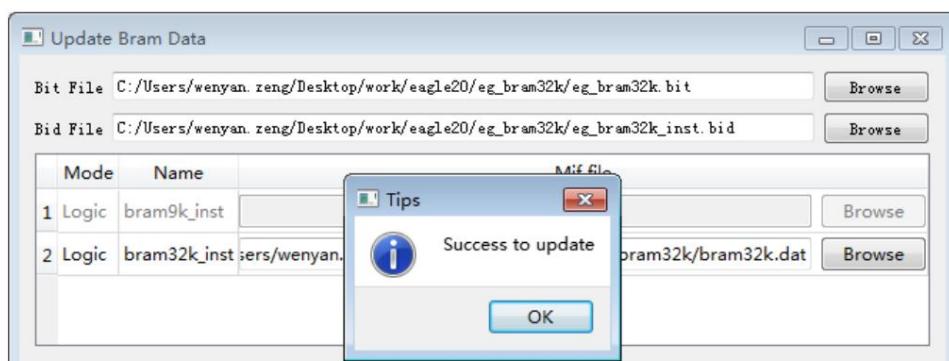
Here, the size of the new BRAM Data must be the same as the size of the BRAM in the design, otherwise an alarm will be given.

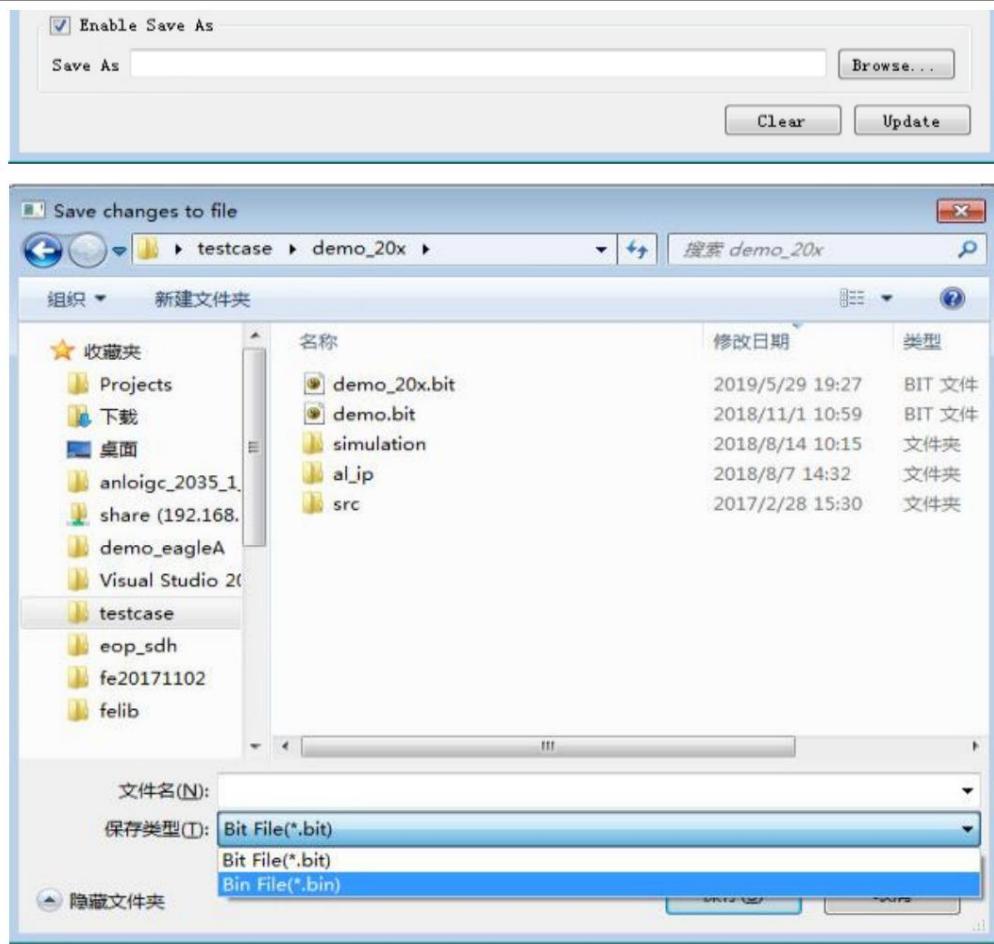
report, and cannot achieve the expected function;



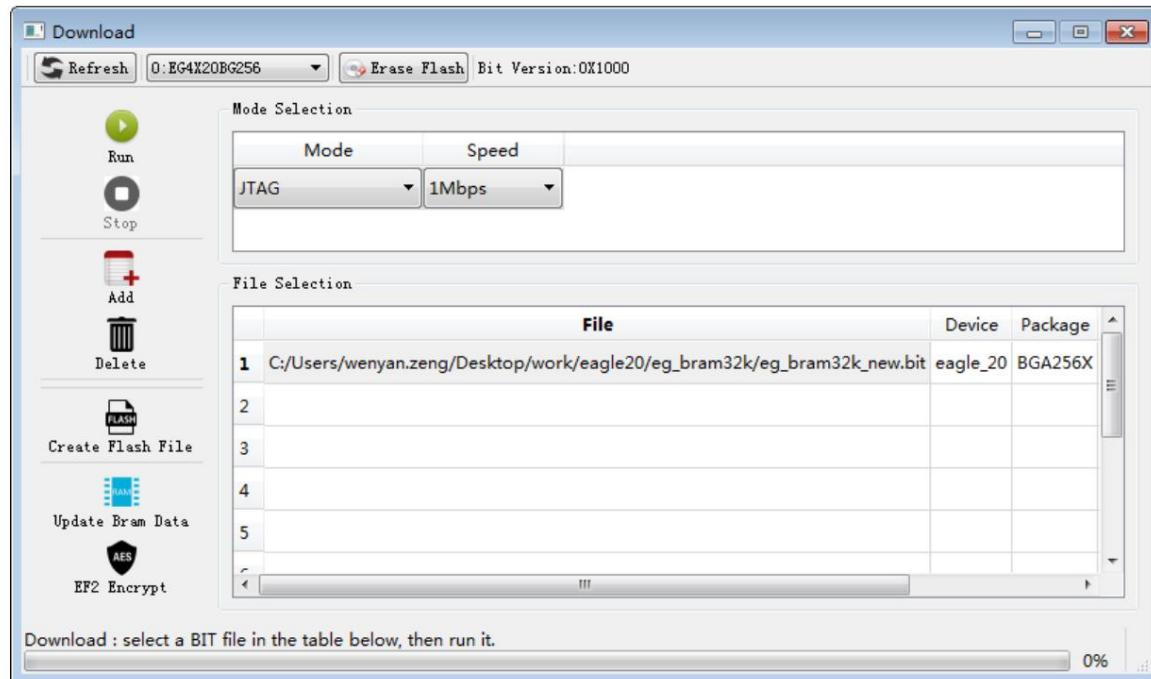
4. You can click **Update** to directly update the current bit file, or you can check the **Enable Save As below**,

Click **Browse** to generate a new bit file. You can also choose to generate a new bin file through **Browse** ;





5. Click OK to complete the update and add a new bit file to download on the **Download** interface.



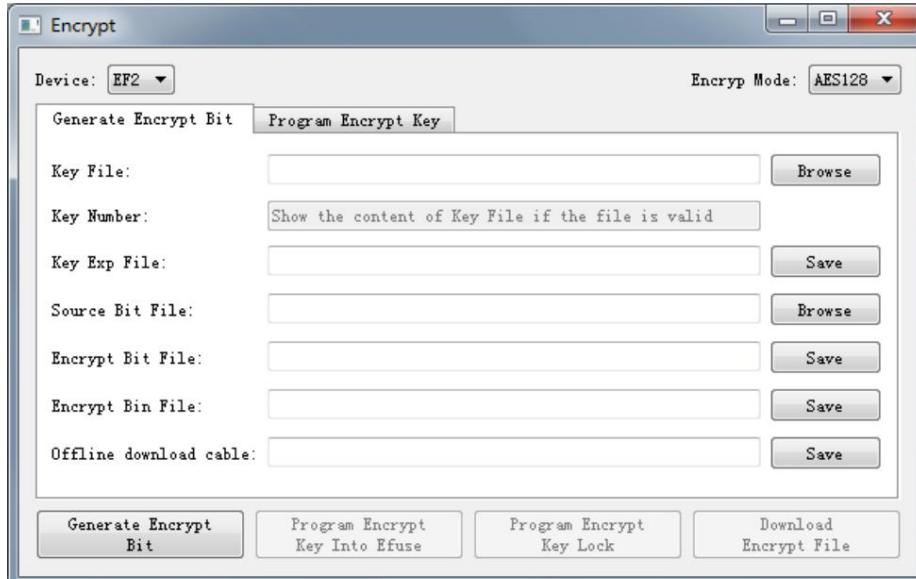
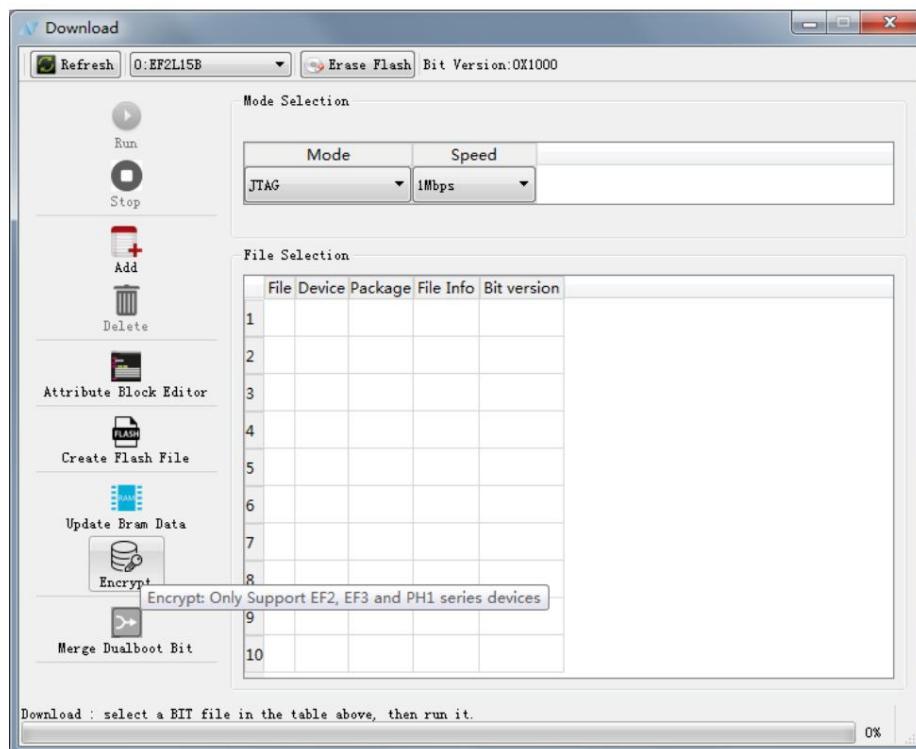
7.4.3 Encrypt

For EF2/EF3 series devices, TD software supports 128Bit AES encryption of bitstream files; for PH1

Series devices, TD software supports 256Bit AES/128Bit SM4 two encryption algorithms. Encrypted files must be

The key provided by the user can be decrypted.

Open the Download interface, click the **Encrypt** button, and the following interface will appear:



When encrypting, the user first uses the plaintext key to encrypt the cipher stream file. The decryption key corresponding to the plaintext key is then communicated

It is written into the chip by programming. The FPGA configuration module will read the decryption key to decrypt the encrypted stream, and the decrypted

The data is then transferred to the configuration module data bus. Taking the EF2 series as an example here, the encryption steps are described in detail as follows:

1. Select the user key file in Key File. The user key is 32 hexadecimal numbers. After selecting the key, the

Displayed in Key Number.

2. In Key Exp File, select the location where the encrypted key file is saved.

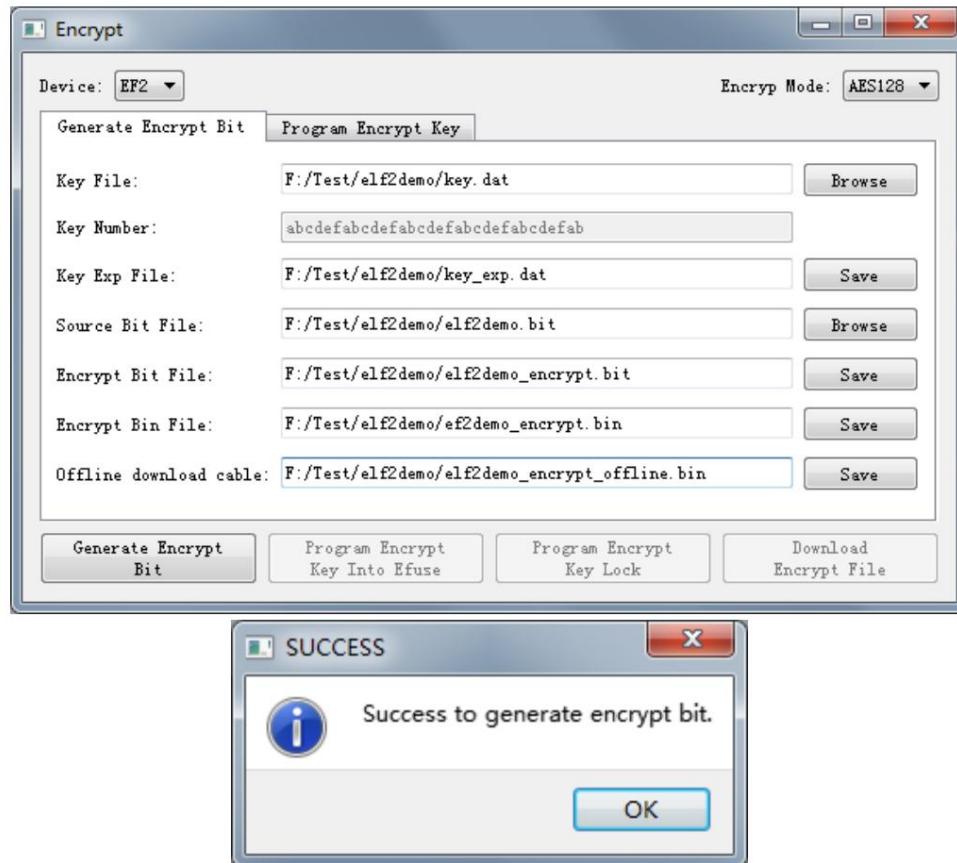
3. Select the EF2 bit stream configuration file (*.bit file) to be encrypted in Source Bit File.

4. In Encrypt Bit File/Encrypt Bin File, select the storage location of the encrypted bit stream configuration file.

Offline download cable generates bin files for offline downloader.

5. Click the Generate Encrypt Bit button to generate the encrypted file. If the encryption is successful, a prompt will be given, otherwise

Encryption failure will give a corresponding error on the TD interface.



After the encrypted file is generated, the content of the encrypted key file needs to be programmed into the EF2 chip. The programming process is as follows

Down:

1. In the Encrypt screen, select the Program Encrypt Key screen.
2. Select the newly generated encryption key file key_exp.dat in Key Exp File; in Encrypt Bit File

Select the encrypted stream file you just generated.

3. Click the Program Encrypt Key Into Efuse button to write the key into Efuse through the programmer

Department.

4. After programming the encryption key, click the Program Encrypt Key Lock button to set the key Lock bit information.

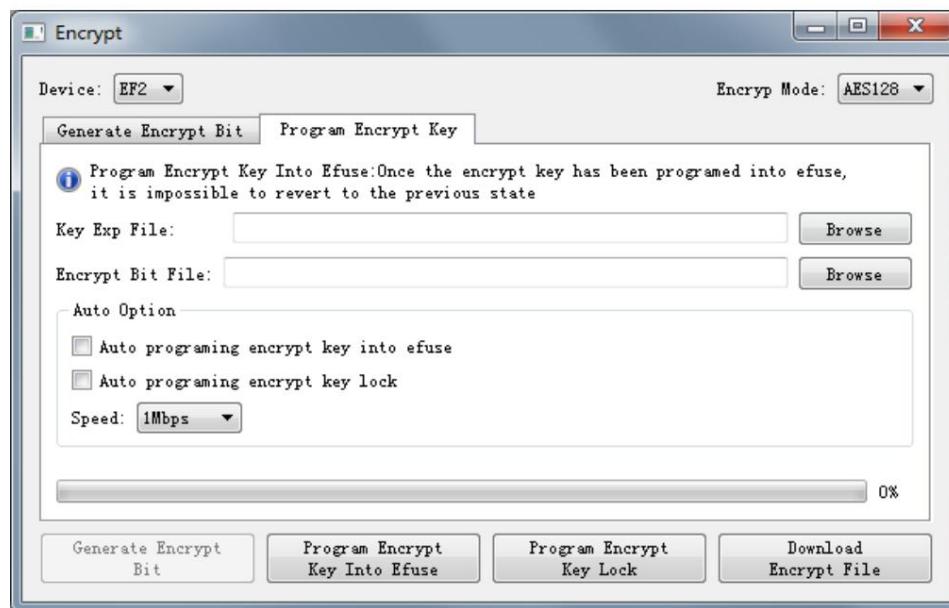
Write to the inside of Efuse through the programmer.

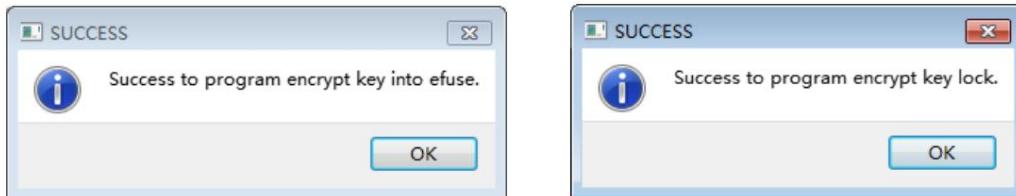
5. Click Download Encrypt File in the lower right corner to download the encrypted stream. If Auto programming is checked

encrypt key into efuse, the software will automatically write the key into Efuse before downloading the encrypted stream

If you check Auto programing encrypt key lock, the software will automatically

Write the key Lock information into Efuse.





7.5 Offline Downloader

7.5.1 Introduction of Offline Downloader

The offline downloader also supports online JTAG program download (compatible with traditional downloaders), and online FLASH direct reading

Write, offline FLASH program programming in three modes. Among them, the offline FLASH program programming mode is divided into the following three

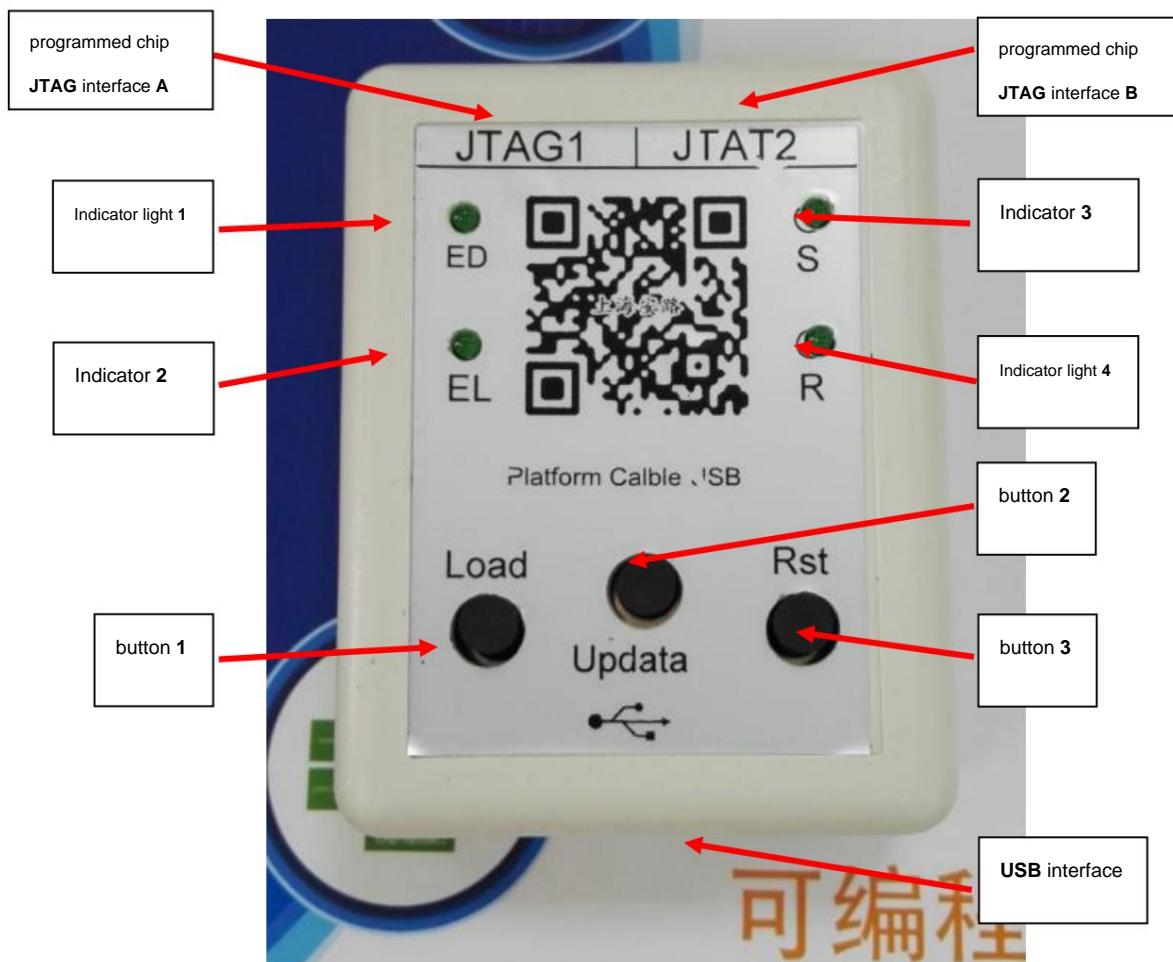
model:

First, support programming SPI FLASH through JTAG

Second, support direct burning of SPI FLASH

Third, support the programming of encrypted bit stream and encrypted KEY of ELF2 series devices

The hardware introduction of the offline downloader is as follows:



The working status of the offline downloader is described as follows:

After power-on, the offline downloader is in the online JTAG download mode. At this time, the downloader is like an ordinary downloader.

The JTAG interface of the programmed chip can be connected to the JTAG of the FPGA on the target board to realize the FPGA on the target board.

program download, including online JTAG debugging and target board FLASH online download. In this state, indicator light 1,

Indicator light 2 and indicator light 3 are all off, and indicator light 4 is flashing.

When the button 2 is pressed, the offline downloader enters the FLASH direct reading and writing mode, and the PC software can read and write at this time.

The content of the source FLASH used in offline mode on the offline downloader. In this mode indicator 3 flashes, indicating

Indicator 4 remains flashing.

When pressing button 1, the offline downloader will enter the offline download mode, and will no longer be controlled by the PC software.

The downloader will automatically read the 16-byte header of the source FLASH, and then enter the three offline download modes according to the byte header.

One of them, and then make a FLASH copy. In this mode, indicator 4 is always on, and indicator 1 is on to indicate detection

The ID of the target FLASH is wrong, and the target FLASH may not be detected; the indicator light 2 is on to indicate that the copy is detected

The comparison of FLASH data is wrong; the indicator light 3 is on, it means that a FLASH copy is completed correctly, and it can be exchanged at this time.

For the next chip, press button 1 again to continue the next programming.

In any mode, by pressing button 3, the offline downloader can return to the power-on initial mode, that is, the online JTAG

Download mode.

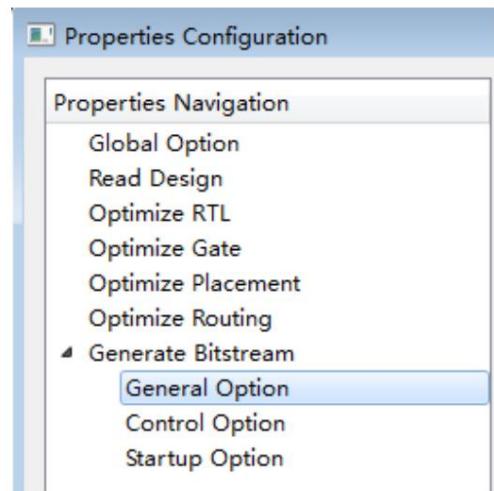
7.5.2 Use steps of offline downloader

1. Generate a BIN file that can be downloaded to the source FLASH

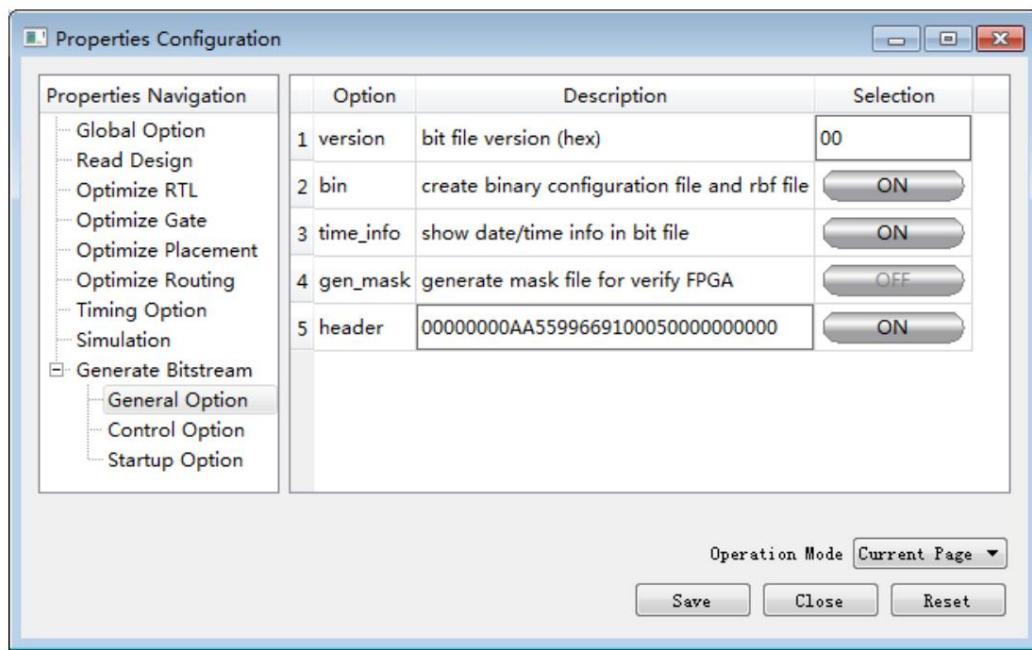
The BIN file is characterized by adding a 16-byte header based on the bit file running on the target FPGA.

These 16-byte headers can be added by setting the software before compiling the project. The specific steps are as follows:

Step 1, open the software property setting window **Properties**→**General Option**



Step 2: Set the value of the **bin** option to ON, and add a 16-byte header to the **header** column.



The header format is defined in the following table:

Byte	Example (hexadecimal)	Meaning
0~3	0000_0000	padding bits
4~7	AA55_9966	Identifier
8	11	<p>Command word:</p> <p>Bit7~6 Select the erase command:</p> <ul style="list-style-type: none"> 01: sector erase 4K ~50ms 10: block erase 64K~0.2s 11: chip erase Bit5~4 full chip ~1.5s <p>Select whether to read back the verification:</p> <ul style="list-style-type: none"> 01: read back 10: no readback <p>Bit3 selection is automatic connection test:</p> <ul style="list-style-type: none"> 0: do not test automatically 1: Automatic test connection in idle state <p>Bit2:0 Select programming mode:</p> <ul style="list-style-type: none"> 001: Program SPI FLASH through JTAG (except ELF1 outside) 010: Directly burn SPI FLASH 011: Burn SPI FLASH through JTAG (for ELF1 device)
9~10	800	Burning length, number of pages, minimum 1 page (256 bytes), low byte first
11~12	400	Burning start address, page number, 0 address corresponds to 0, low byte first
13	A5	Whether to program the encryption KEY indication bit, if it is A5 or AA needs to burn the encryption KEY, which is only applicable to ELF2 device pieces. Any value other than AA/A5 will not be burned KEY.
14~15	0000	reserved padding bytes

After filling in the header according to this form, save the settings, and recompile the target project.

Set the BIN download file of the header.

2. Download the generated BIN file to the source FLASH of the offline downloader

Extract the 16-byte file header of the BIN file generated in the previous step and divide it into two separate files

PROJ_HEAD.bin (stores the file header), PROJ_BIN.bin (stores the file body bit stream).

```

PROJ_BIN.bin x
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 00 00 00 00 AA 55 99 66 91 00 05 00 00 00 00 00 ; ....烽櫻?.....
00000010h: FF DD 66 BB 22 FF FF FF 82 FF FF FF FF FF FF FF ; 烽?
00000020h: CC 88 88 33 CF F1 20 4E 87 A9 8E 61 18 69 5D 45 ; 烟?像 N噪龜.i]E
00000030h: A3 1A EE 05 CF F1 20 4E 87 A9 8E 61 18 69 5D 45 ; ??像 N噪龜.i]E
00000040h: A3 1A EE 05 E7 E6 7E 1C E8 51 3A C9 28 AF 44 45 ; ??珂~.鏤?:?蒼E
00000050h: 30 A0 4C 97 25 50 1E 98 F7 89 3A 18 3D 4B E4 38 ; 0燥?P.僚?.=K?
00000060h: 9A C6 95 9C FD 11 34 F0 E7 52 45 60 5B E5 29 32 ; 鬼魄?4瘡RE`[?2
00000070h: 4E A2 EE D5 89 20 BA E6 F4 9A 22 77 9E 23 26 68 ; N(+)諫 烘飴"w?&h
00000080h: E3 76 B4 EA B0 46 7A 52 AA B4 64 50 18 4C F6 B2 ; 試搓癮zR dP.L龜
00000090h: 70 FB 19 0F 90 92 69 6F D6 9A 0B DA EC 53 C4 01 ; p?.悞io謙.陟S?
000000a0h: 1E 58 8A 3D D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; .X?...f |-?聳
000000b0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳
000000c0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳
000000d0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳
000000e0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳

```

Complete BIN file (bit stream + 16 -byte file header)

```

PROJ_HEAD.bin x
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: 00 00 00 00 AA 55 99 66 91 00 05 00 00 A5 00 00 ; ....烽櫻?.....

```

The BIN file that stores the 16 -byte file header after splitting

```

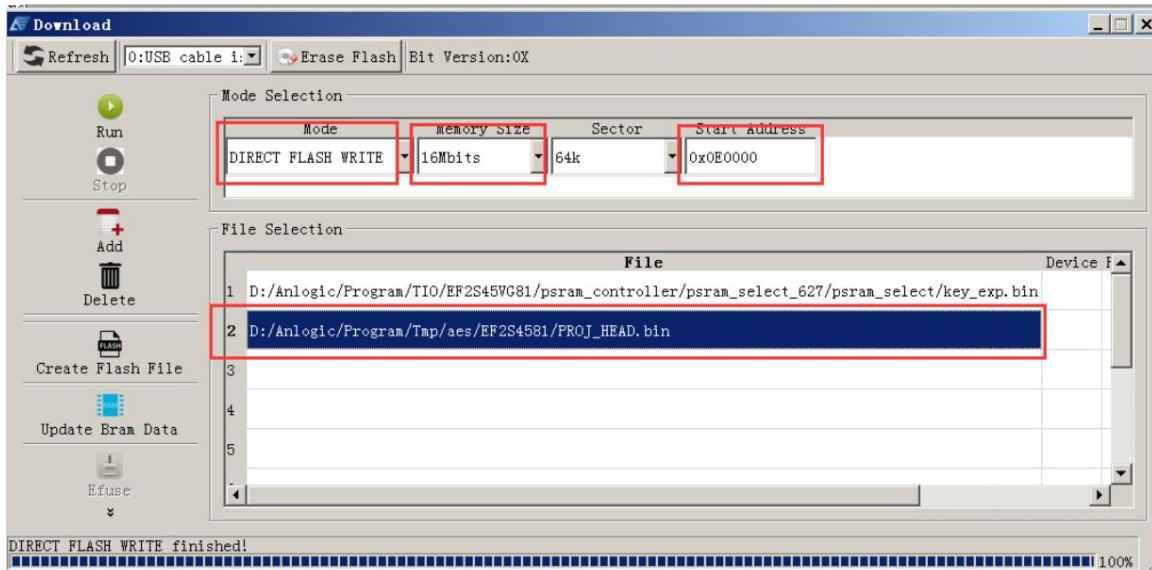
PROJ_HEAD.bin x PROJ_BIN.bin x
0 1 2 3 4 5 6 7 8 9 a b c d e f
00000000h: FF DD 66 BB 22 FF FF FF 82 FF FF FF FF FF FF FF ; 烽? ?
00000010h: CC 88 88 33 CF F1 20 4E 87 A9 8E 61 18 69 5D 45 ; 烟?像 N噪龜.i]E
00000020h: A3 1A EE 05 CF F1 20 4E 87 A9 8E 61 18 69 5D 45 ; ??像 N噪龜.i]E
00000030h: A3 1A EE 05 E7 E6 7E 1C E8 51 3A C9 28 AF 44 45 ; ??珂~.鏤?:?蒼E
00000040h: 30 A0 4C 97 25 50 1E 98 F7 89 3A 18 3D 4B E4 38 ; 0燥?P.僚?.=K?
00000050h: 9A C6 95 9C FD 11 34 F0 E7 52 45 60 5B E5 29 32 ; 鬼魄?4瘡RE`[?2
00000060h: 4E A2 EE D5 89 20 BA E6 F4 9A 22 77 9E 23 26 68 ; N(+)諫 烘飴"w?&h
00000070h: E3 76 B4 EA B0 46 7A 52 AA B4 64 50 18 4C F6 B2 ; 試搓癮zR dP.L龜
00000080h: 70 FB 19 0F 90 92 69 6F D6 9A 0B DA EC 53 C4 01 ; p?.悞io謙.陟S?
00000090h: 1E 58 8A 3D D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; .X?...f |-?聳
000000a0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳
000000b0h: 70 15 45 02 D7 28 1D 08 1A 66 A9 C4 E3 10 EC A2 ; p.E.?...f |-?聳

```

The BIN file that stores the bit stream after splitting

3. After splitting, store the BIN file of the encrypted bit stream file header into the offline downloader, StartAddress

is 0xE0000.



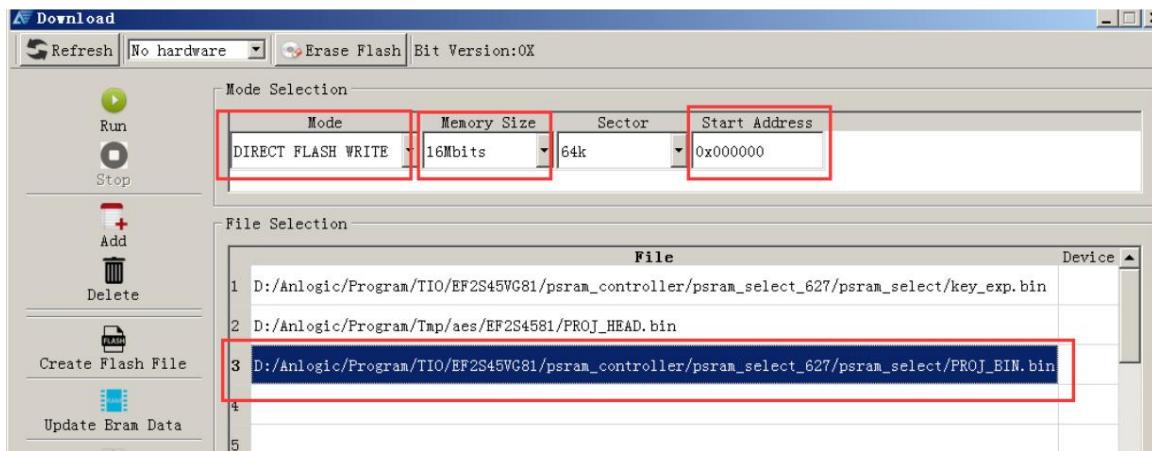
Download file headers to offline downloader

As shown in the figure, select DIRECT FLASH WRITE for the download mode, and fill in the download start address as 0x0E000.

Then select the PROJ_HEAD.bin file generated in the previous step, click button 2 (middle button) of the offline downloader,

Then click the run of the software.

4. Put the split BIN file that stores the bitstream into the offline downloader.



Download bitstream to offline downloader

As shown in the figure, select DIRECT FLASH WRITE for the download mode, and fill in the download start address as 0x00000. Of course

Then select the PROJ_BIN.bin file generated in the previous step, click button 2 (middle button) of the offline downloader, and then

Click the run of the software.

5. Use the offline downloader to download the bitstream to the target board.

Power on the target board, connect it with the JTAG1 port of the offline downloader, and then press the button 1 of the offline downloader

(left key), that is, download the bit stream to the target board. If you hear a sound of about 2S from the buzzer, the indicator lights 3 and

The indicator light 4 is on, it means the loading is successful.

7.5.3 Offline Downloader Indication Status

1 Indicator	4 blinks rapidly	Online download status, the downloader can be used as an online downloader with TD software. The internal FLASH update status of the offline downloader. At this time, the
2 Indicator	light 3 and indicator light 4 flash rapidly at the same time 3 Indicator light 4 is always on	header,, source and other information in the offline downloader can be updated through the TD software. The offline downloader is in the process of burning. The longest bright
		time will not exceed 20S, otherwise it will prove that the downloader is faulty. There is an error in offline download, which usually occurs within 2S after pressing offline download. It
4 Indicator	light 2 and indicator light 4 are always on, and the buzzer sounds intermittently	is more common when the normal header is not found or the target device is not found. Offline download success indication. Usually download about 7S.
6 Indicator	lights 3 and 4 are always on, and the buzzer stops after 1S	

7.6 Device Chain

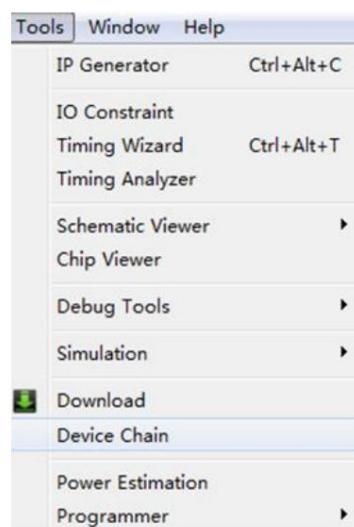
In order to meet users' needs for JTAG cascade loading of multiple CPLDs or FPGAs, TD provides Device

The Chain function is used to generate and download SVF files. One chip can be selected for download, or multiple SVF files can be downloaded.

After the files are merged, download them at one time to avoid users from performing download operations multiple times.

The specific operations are as follows:

1. Open Device Chain: Tools → Device Chain;



The Device Chain interface as shown in the figure below appears. The left side of the interface is the toolbar, and the Files Option area is

File list, Device Chain area is the chip cascade display area.

Program : JTAG load button;

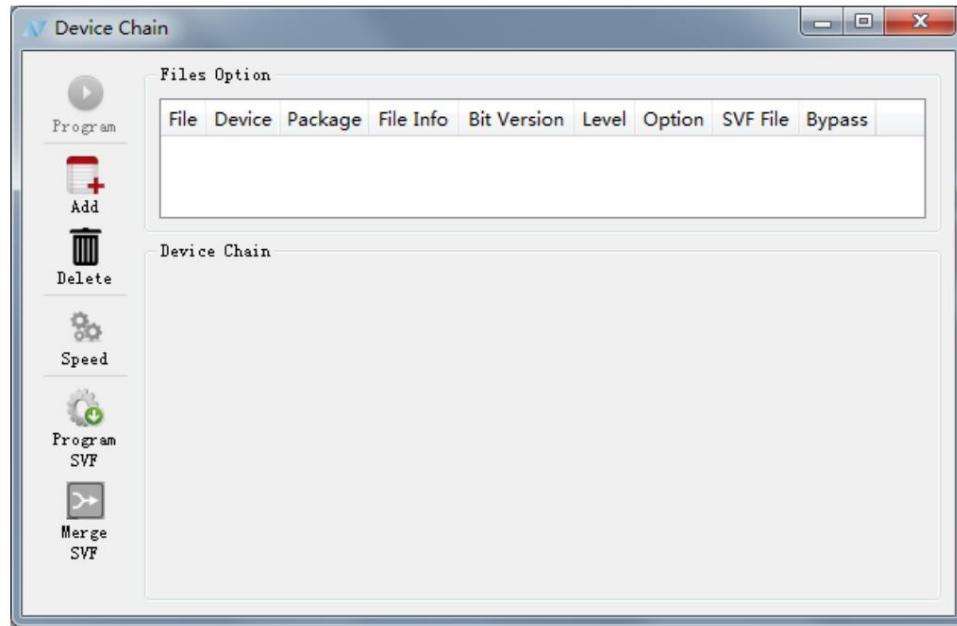
Add : Add file button;

Delete : delete file button;

Speed : Speed selection button when JTAG is loaded;

Program SVF : Load the merged SVF file;

Merge SVF: Merge multiple SVF files.



2. Add files and devices

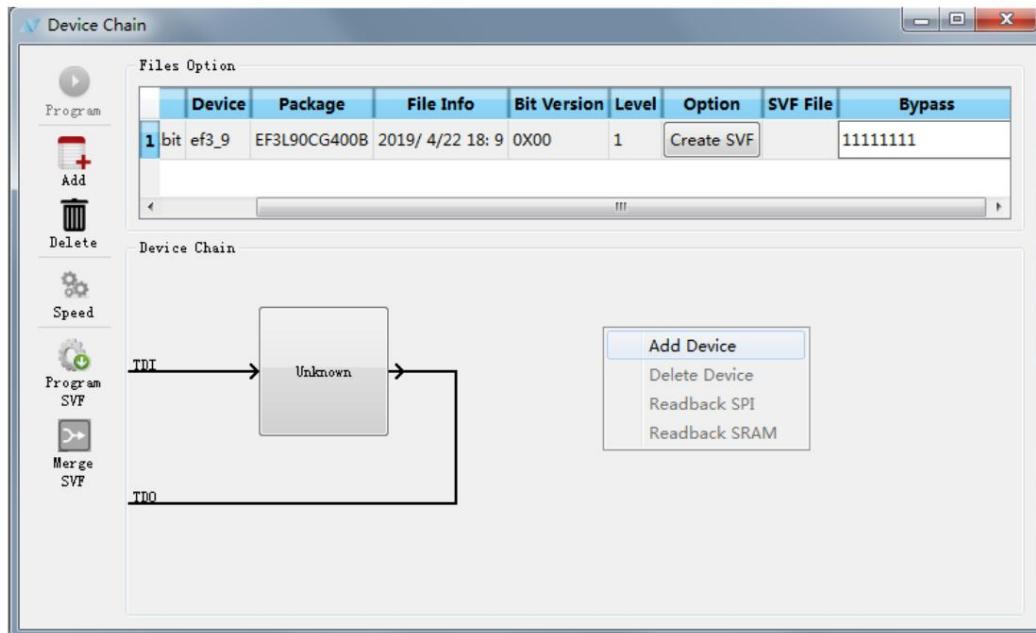
Click the "Add" icon on the toolbar on the left side of the interface to add a bit file or bin file. If it is a three-level cascade,

Three files need to be added, and the first added file is Level1 by default. Correspondingly, each time you add a file

A Device cascade will be added.

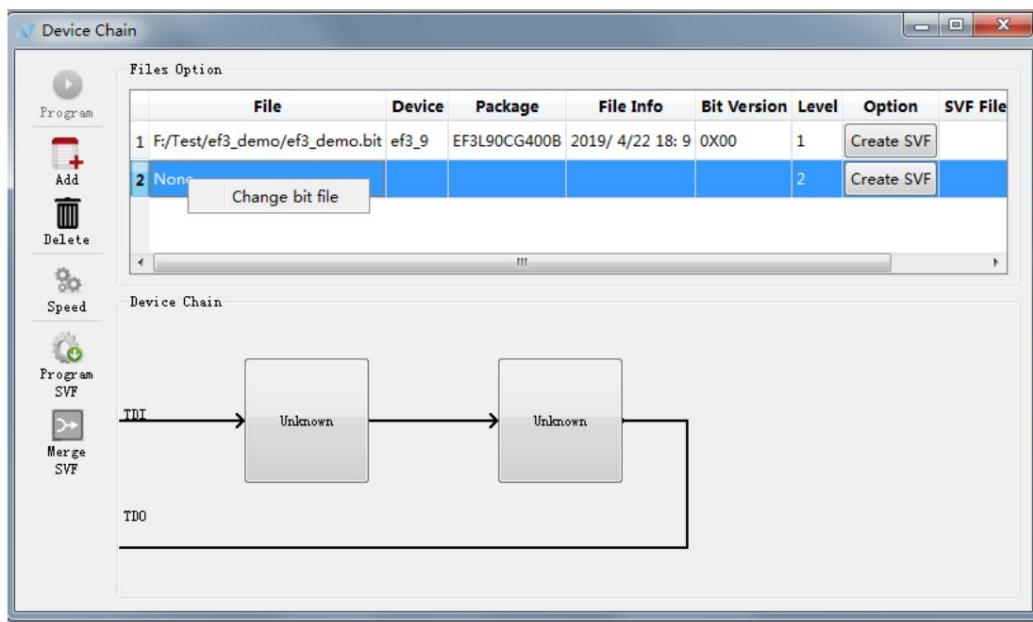
Similarly, you can first in the Device Chain area, right-click, and select "Add Device" to add. First

The added Device is Level1.



If you need to change the file of a certain level, first select the line corresponding to the level in the file list, and then select "Change"

bit file" to perform file change operations.



To delete a certain level, you can select the file of this level and click the "Delete" button on the left menu bar.

In the Device corresponding to this level, right-click and select "Delete Device".

The last column of the file list is the Bypass value corresponding to each level of chip. Currently, the default value is 11111111.

The value of Bypass of different manufacturers may be different.

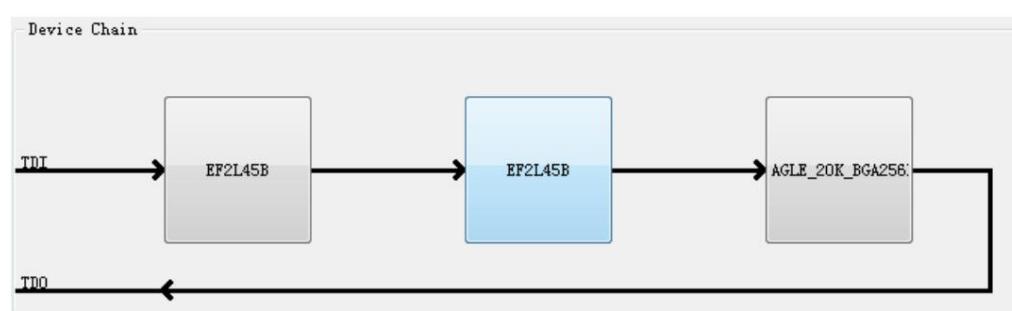
3. Chip ID identification

After adding files or devices, all devices are displayed as Unknown, and the connected devices need to be identified manually.

device. If the Device is connected correctly, just click on the Unkown icon and it will display on the corresponding button

The specific device type is shown, as shown in the figure below, two EF2 devices and one Eagle_20 device are cascaded

pieces.



4. Generate SVF file

Click the "Create SVF" button in the file list, and set the parameters in the pop-up dialog box to generate the corresponding

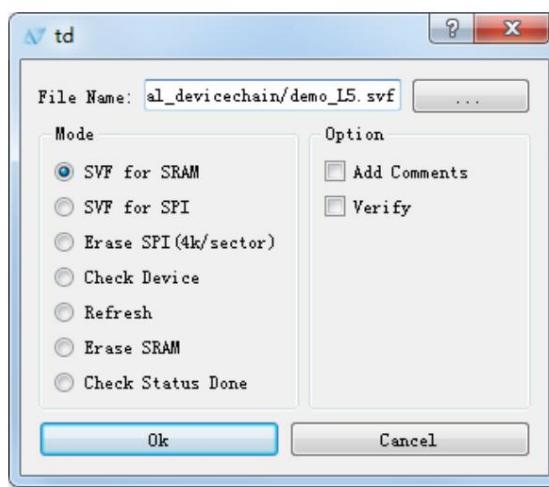
Hierarchical SVF file. There are 7 modes when generating SVF files:

1) SVF for SRAM : default mode, can support n-level SRAM download;

ÿ Add Comments option is to add comment information in SVF file;

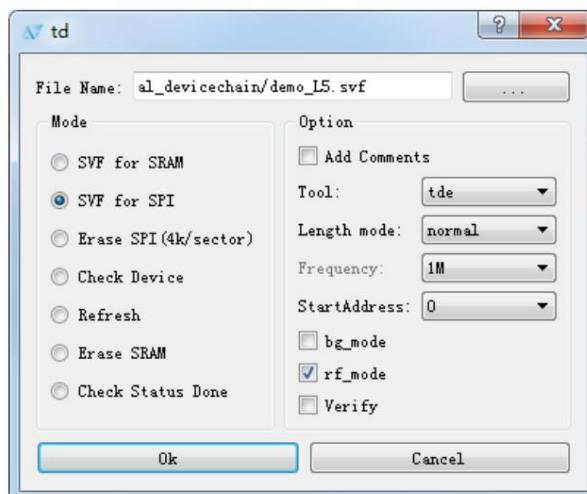
ÿ The Verify option is used to compare the configuration bit information in the FPGA chip with the bit currently selected by the user

Whether the information in the file is consistent.



2) SVF for SPI : support 1, 2, 3 level write (EF3L90CG400B and EF3L40CG332B two

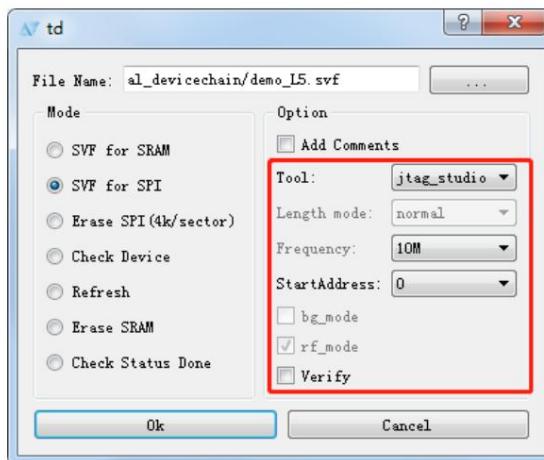
A variety of device types support arbitrary multi-level write), n-level readback verification (verify);



ÿ tool : optional tde and jtag_studio, the default is tde. When selected as jtag_studio,

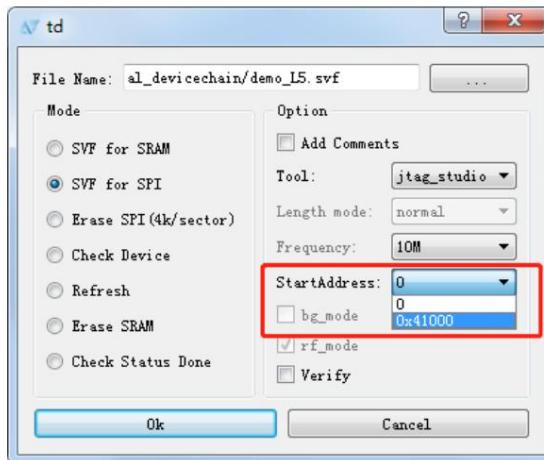
length_mode, bg_mode and rf_mode are not optional, they are all default options; Frequency

becomes optional.



ÿ length mode: normal and short are optional, the default is normal;

ÿ StartAddress: The starting address of the download can be specified;



ÿ bg_mode : background mode, unchecked by default;

ÿ rf_mode : Checked by default, automatically refresh after program spi;

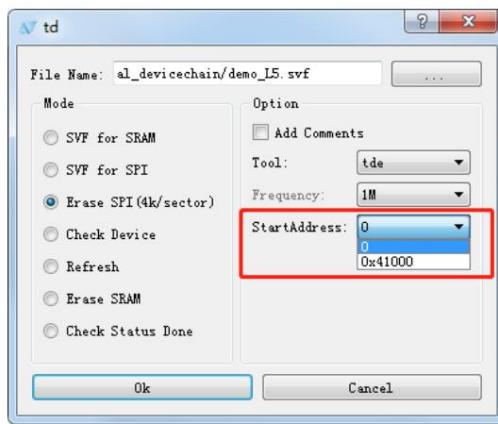
ÿ Verify : The verification function of SVF. After the bit stream programming is completed, read out the target area information

Compare with the file to determine whether there is a bit write error.

3) Erase SPI (4K/sector) : SPI erase;

ÿ tool can be tde and jtag studio, the default is tde;

ÿ StartAddress can specify the start address of erasing.



4) **Check Device** : Check whether the device id corresponding to the bit file is consistent with the device id;

5) **Refresh** : refresh the chip status;

6) **Erase SRAM** : Erase SRAM;

7) **Check Status Done** : Read the status of done in the status register through JTAG, check

Whether program spi is successful.

5. SVF merge

In order to avoid cascading, the user performs multiple download operations, and the SVF file merging function is added.

SVF files are combined into one file for one-time download. As shown in the figure, select where the SVF to be merged is located

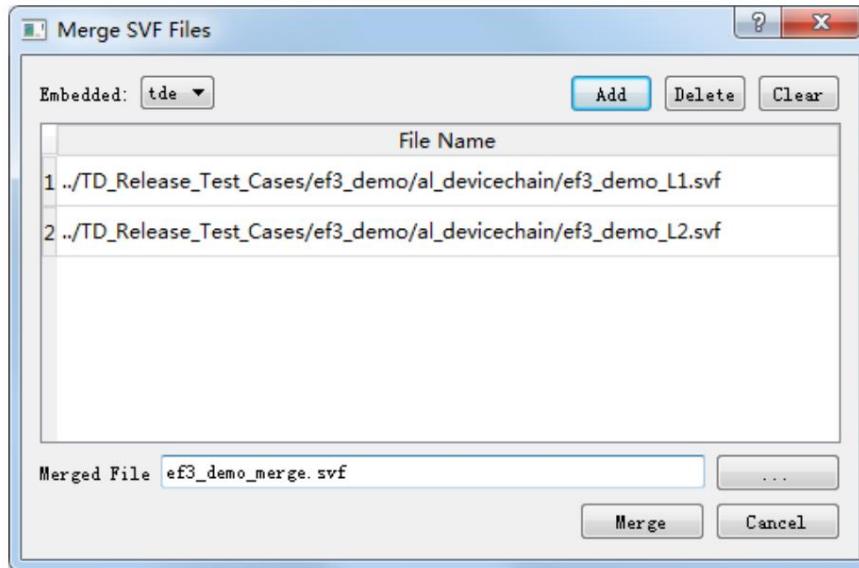
, right-click and select "Merge SVF files".

	Device	Package	File Info	Bit Version	Level	Option	SVF File	Bypass
1 ELF2_DOMO_2/project/domo/domo.bit	ef2_4	EF2L45LG144B	2018/11/ 1 11:35	0X11	1	Create SVF	al_devicechain/test_osc_l1.svf	11111111
2 ELF2_DOMO_1/project/domo/domo.bit	ef2_4	EF2L45LG144B	2018/ 8/17 18:13	0X00	2	Create SVF	al_devicechain/test_osc_l2.svf	11111111
3 domo.bit	eagle_20	BGA256X	2017/11/ 3 11: 2	0X00	3	Create SVF	al_devicechain/test_osc_l3.svf	11111111

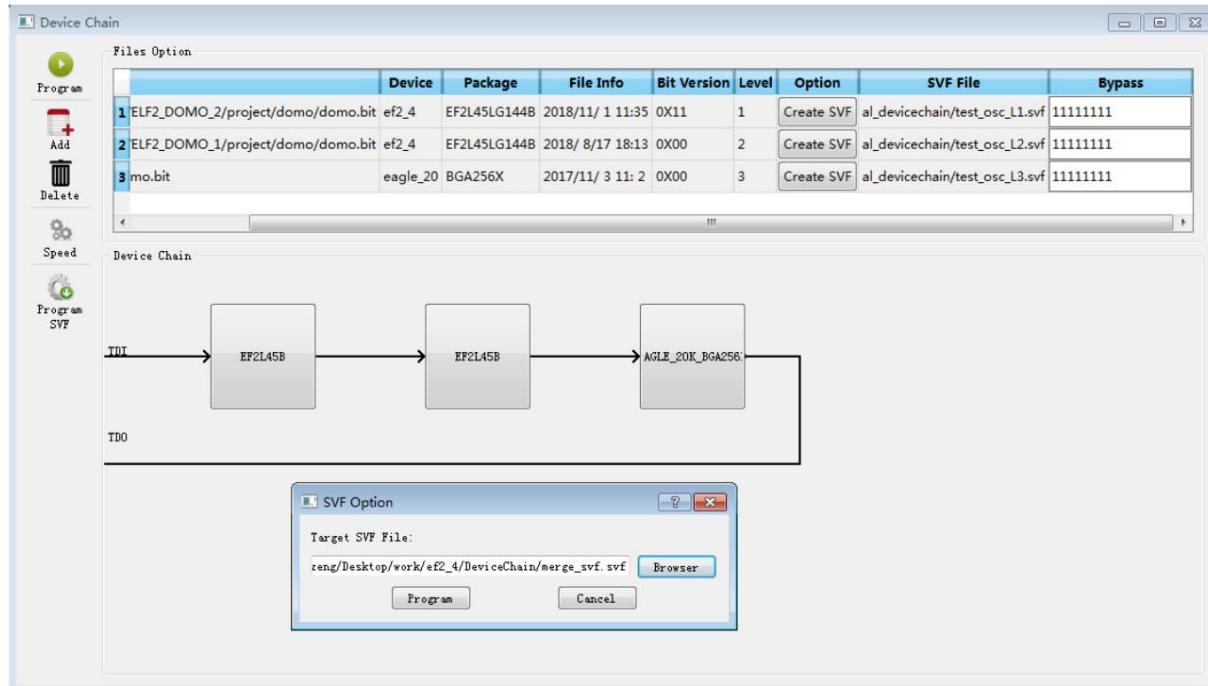
Merge svf files

Or open the Merge SVF menu on the left, Add/Delete/Clear can add/delete/clear SVF respectively

file, click Merge/Cancel in the lower right corner to Merge SVF/Cancel Merge respectively.



The merged SVF file is downloaded using Program SVF.



6. Program

Select the row of the SVF to be downloaded, and click the Program button on the left toolbar to download. SVF for

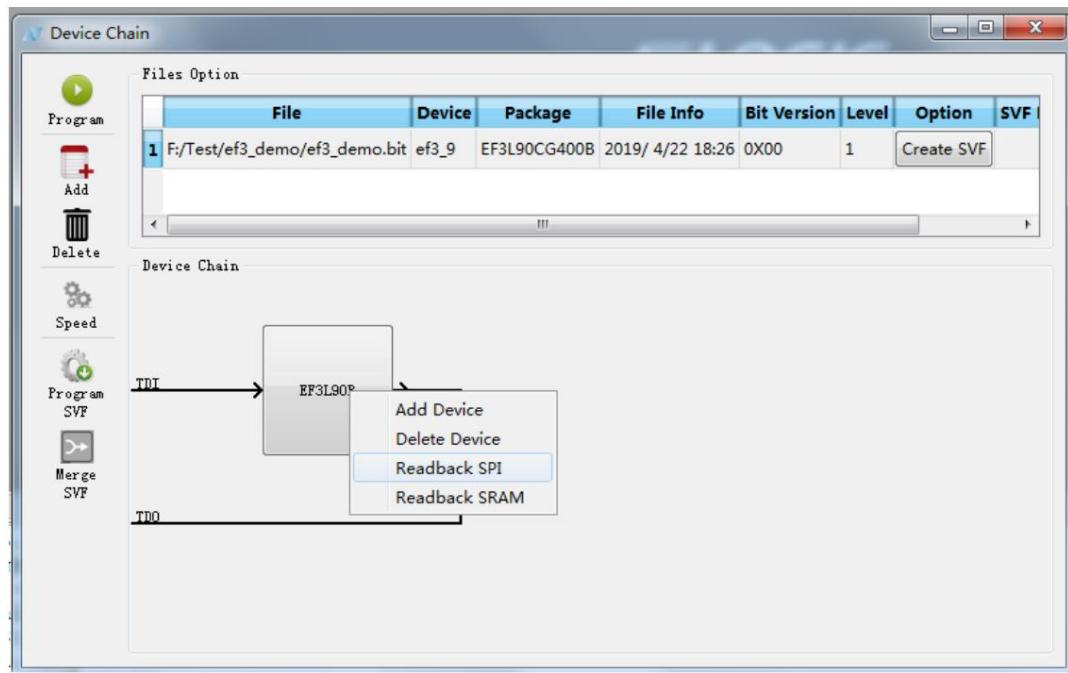
SRAM, SVF for SPI and Check Device mode all check the device id and

The device id is consistent, if it is inconsistent, an error will be given in the TD Console window.

Before downloading, you can also click the Speed button on the toolbar to adjust the download speed, the default value is 6

(1Mbps).

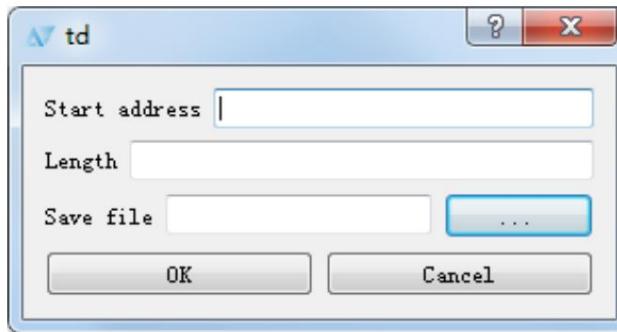
7. Read SPI & Read SRAM



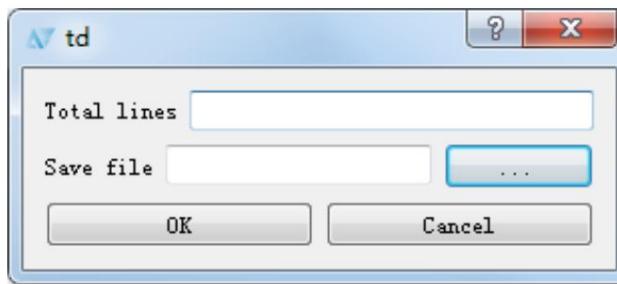
Read SPI: To read the content in Flash, the user needs to specify the starting address and content length. specific operation

Action: In the Device Chain area, right-click to select Readback SPI (note: please make sure the mouse position is on the core

within the range of the chip icon and the chip has been correctly identified);



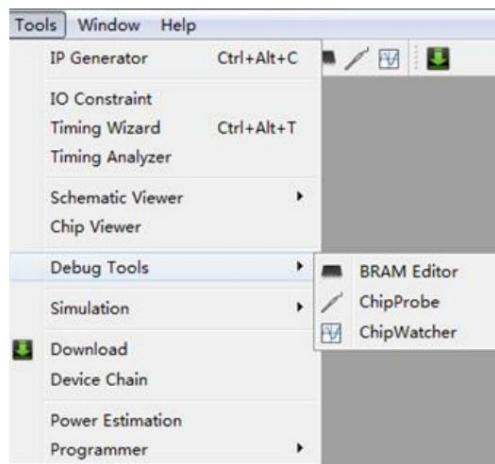
Read Sram: Read the content of Sram, the specific operation is the same as above, select Readback Sram when right-clicking.



8 Tool Sets

There are many tools in TD software to help users better analyze projects, mainly including: Schematic **Viewer**,

ChipViewer, and three tools in the debugging toolset: **BramEditor**, **ChipProbe**, **ChipWatcher**.



Schematic Viewer generates corresponding logic circuit diagrams for each intermediate process of synthesis optimization, improving the design

The interactivity in the process helps designers understand the circuit and shortens the development cycle of circuit design. **Chip Viewer** can

Provides post-physical implementation details, including resource usage, detailed placement, global routing, and critical timing paths.

Without changing the design, ChipWatcher can see the internal signal

It is an online debugging tool to monitor the changes under certain conditions. **ChipProbe**, on the other hand, takes the information that needs to be monitored internally by

The number is assigned to an idle IO port, and can be viewed by external tools such as an oscilloscope. **Bram Editor** from the chip

RAM reads data, and can modify these data. After modification, write it into the chip, and you can see the effect of the modification.

8.1 Schematic Viewer

After completing each process in the TD software, the corresponding logic circuit can be viewed in the **Schematic Viewer**

picture. After "Read Design" and "Optimize RTL" are completed, the viewed logic circuit diagram is a syntax analysis, logic

The effect after serial optimization, the circuit structure at this time is independent from the engineering device, what you see in the circuit diagram is the adder,

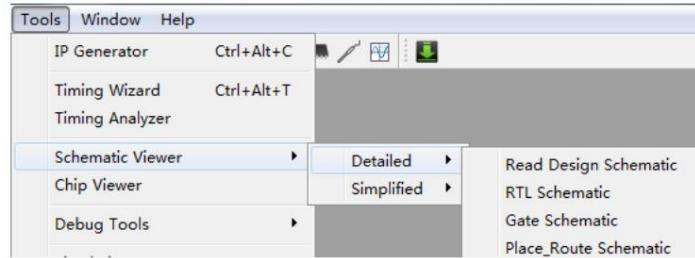
Components such as multipliers, comparators, AND gates, or gates. And in "Optimize Gate", "Optimize Placement",

After "Optimize Routing" is completed, the viewed logic circuit is optimized based on the structure of the FPGA device selected in the project.

The effect of the circuit diagram is the look-up table, register, BRAM, PLL and other components.

Expand **Tools** > **Schematic** Viewer, you can see that there are two modes to choose from: Detailed and

Simplified.



When HDL2Bit Flow runs to the Read Design step, the Read Design Schematic can be executed.

When HDL2Bit Flow runs to the step of Optimize RTL, RTL Schematic can be executed

When HDL2Bit Flow runs to the Optimize Gate step, Gate Schematic can be executed.

When HDL2Bit Flow runs to the Optimize Routing step, the Place_Route Schematic can be executed.

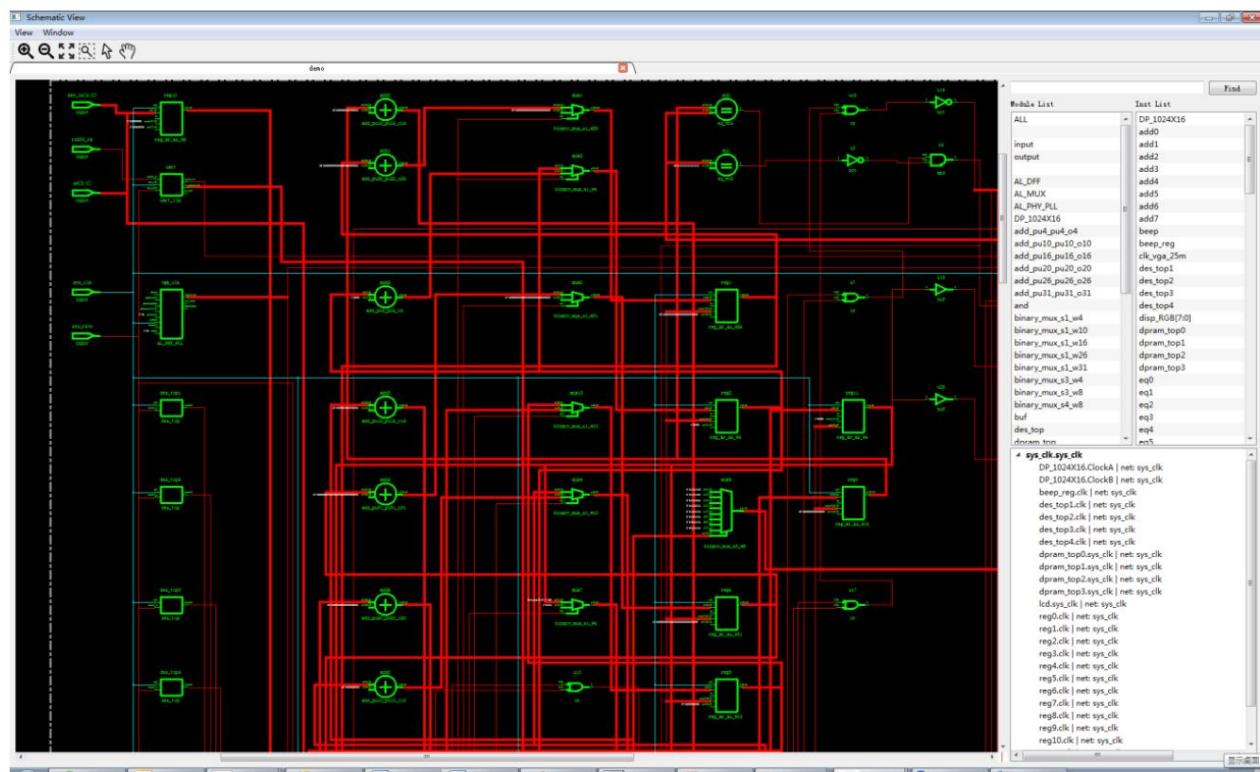
In Simplified mode, the displayed circuit diagram will be more compact, without complicated circuit connections, all the lines

All are transmitted through the bus. This mode is suitable for projects with more resource utilization, as shown in the following figure.



In Detailed mode, you can see detailed circuit information after TD synthesis optimization. This mode is suitable for capital

The source utilizes fewer circuits and can better understand the circuit design, as shown in the figure below.



How to operate in the graphical interface:

1. Highlight: mouse click on the connection, circuit element or by double-clicking Inst, Net;

2. Zoom: Click the button in the toolbar to zoom in, click the button to zoom out, or

Ctrl+wheel to zoom;

3. Overall view: Click the button in the toolbar to display the entire circuit diagram in the main window;

4. Drag: Click the button in the toolbar to drag the main window up, down, left and right;

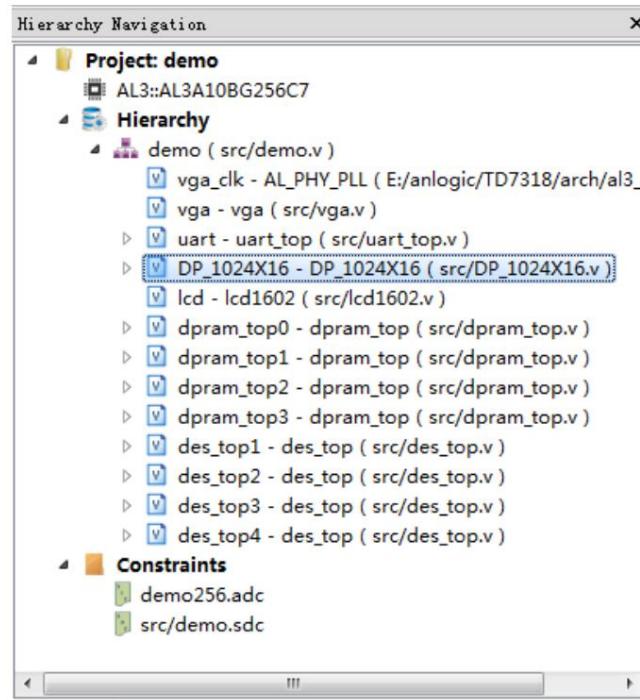
5. Partial zoom in/out: Click the button in the toolbar, press the left mouse button in the main window, and drag down to the right

Move to zoom in on the part, and drag it to the upper left to zoom out;

6. Selection mode: Click the button in the toolbar to switch back to normal mouse functions.

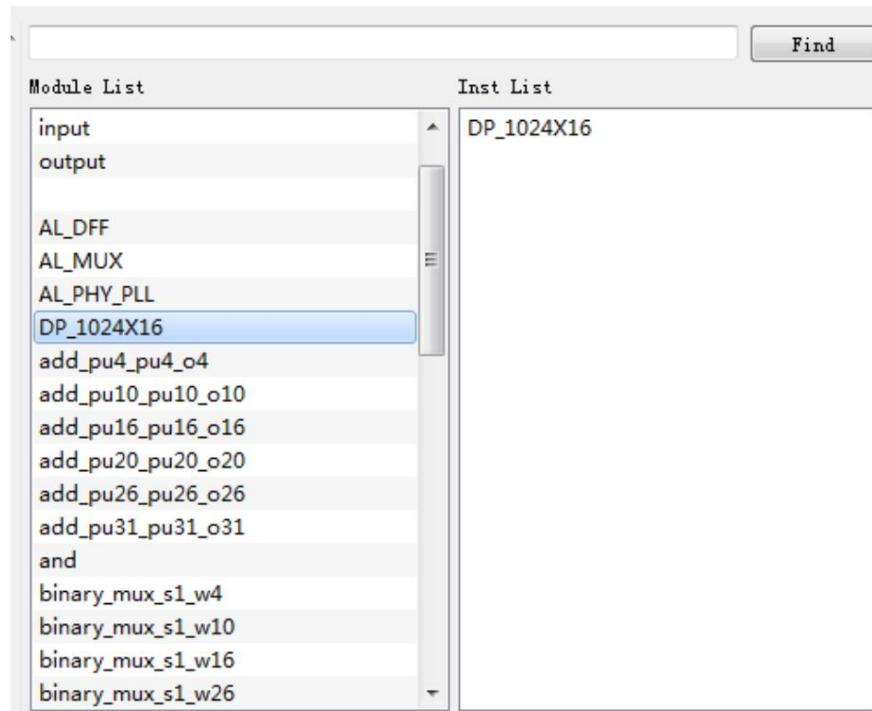
After "Read Design", the user design hierarchy is also preserved, so in the Schematic **Viewer**

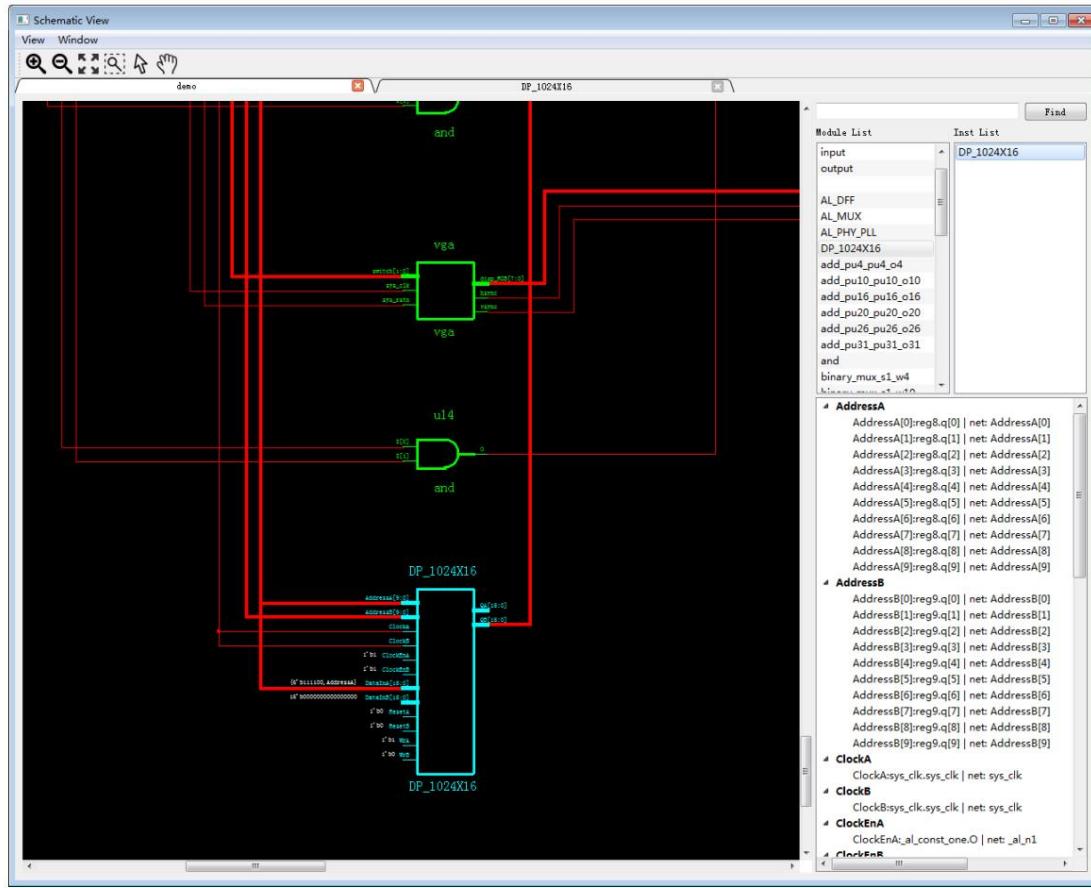
To double click to view the submodules in the design, such as the submodule DP_1024x16 in the following figure.



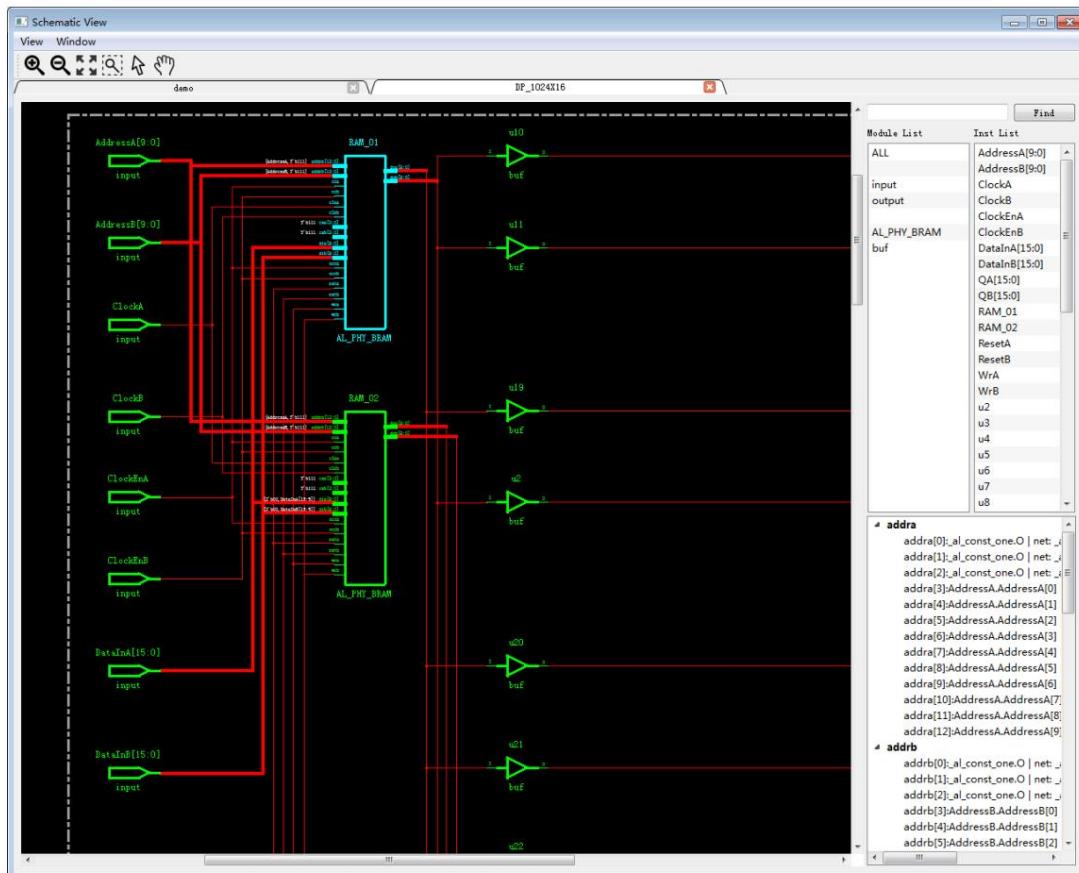
Find DP_1024x16 in the **Module List** and double-click, the corresponding module will be listed in the **Inst List**

Instance, double-click the Instance to jump to its location in the main window.





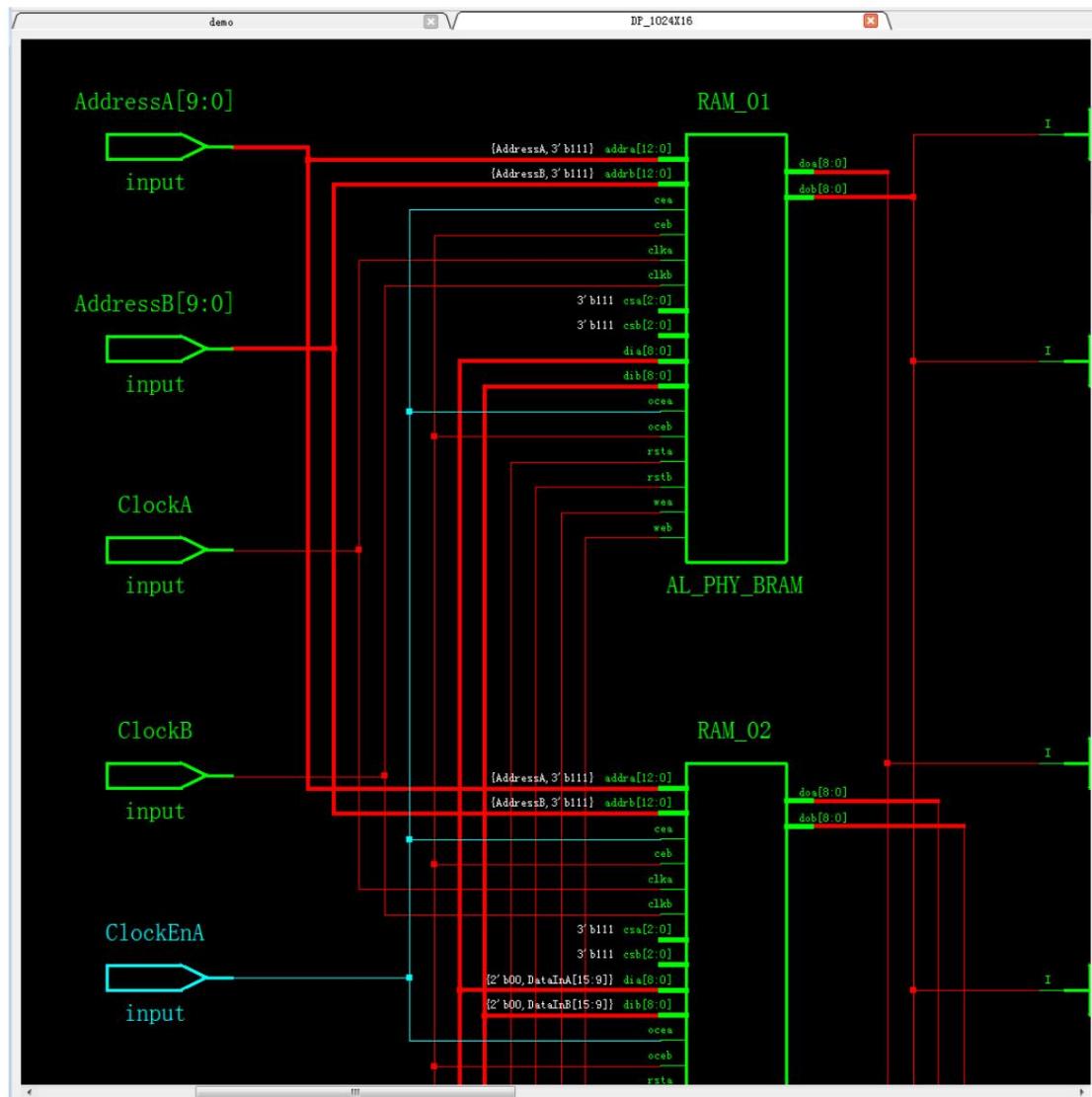
Double-click the DP_1024x16 graphic in the main window to open the corresponding circuit diagram of the module.



Selecting a module in the main view will display all its related nets in the lower right corner of the window, double-click a net

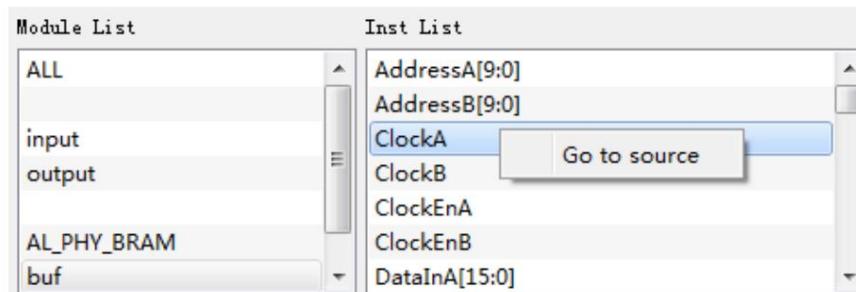
All connections of the net can be highlighted. In the main window, the bold connection is bus, and the non-bold connection is net.

An unwired port is a constant, and the value of that port can be viewed.



In the Inst List, you can also right-click an Instance and select "Go to Souce" to jump to the Instance

location in the source code.



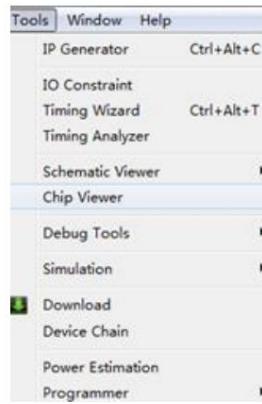
8.2 Chip Viewer

8.2.1 Introduction to Chip Viewer

In TD software , Chip Viewer can be opened after completing "Optimize Gate". In "Optimize

Gate", "Optimize Placement", and "Optimize Routing" are completed, you can see that the physical implementation is gradually completed.

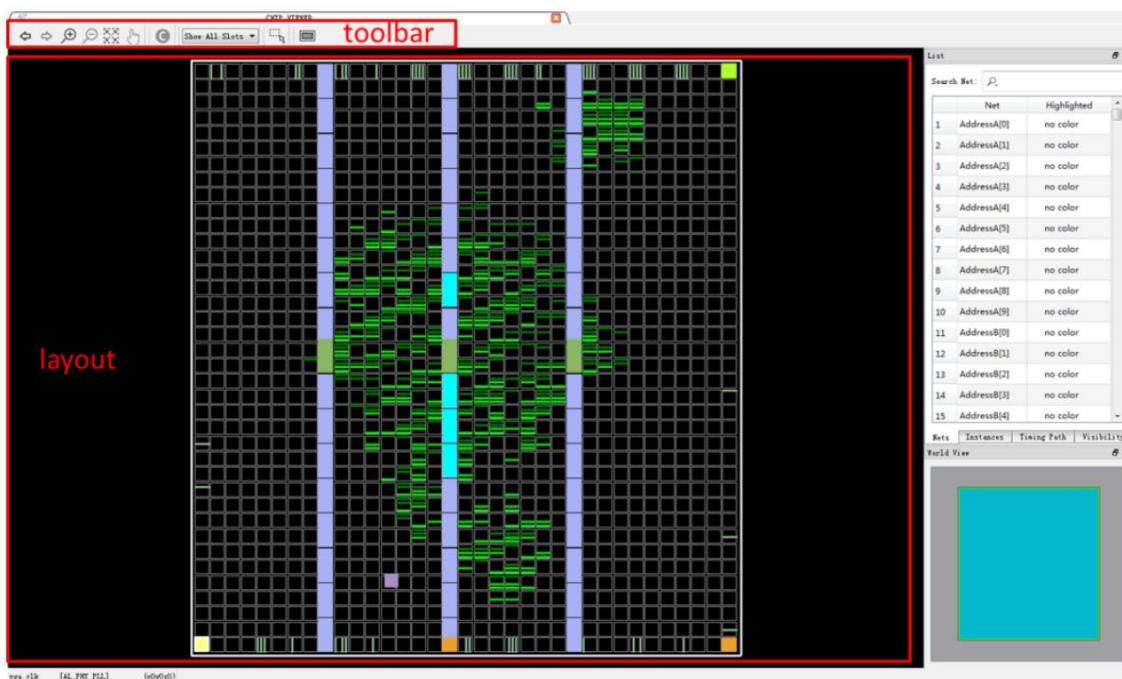
into the details. Click Tools → Chip Viewer to open,



The Chip Viewer interface consists of toolbar, layout, list on the upper right and world view on the lower right. layout

Displays the positions and sizes of all cells in the chip. When the chip viewer is opened, the entire layout is displayed by default.

chip. After zooming and panning, any cell in the chip can be displayed.



The toolbar functions are as follows:



go back to previous view



go back to the next view



Zoom in on current view



Zoom out the current view



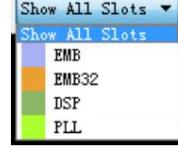
Go back to the default view, which is the view when the Chip Viewer was just opened



Drag the view to pan when pressed



Press to enter the zone constraint management mode



Displays options for a specific type of device



After pressing, you can drag the mouse to select a group of cells/devices



Maximize/Restore Chip Viewer window size

Layout has three view modes:

SELECT (select, default mode), AREA_SELECTION (box selection), PAN (pan).

a) SELECT mode

mouse click

-- Select the instance or slot of the current position, as shown in the upper left figure

double click

-- Select the cell at the current position, multiple selections can be made when CTRL is pressed, such as the lower left

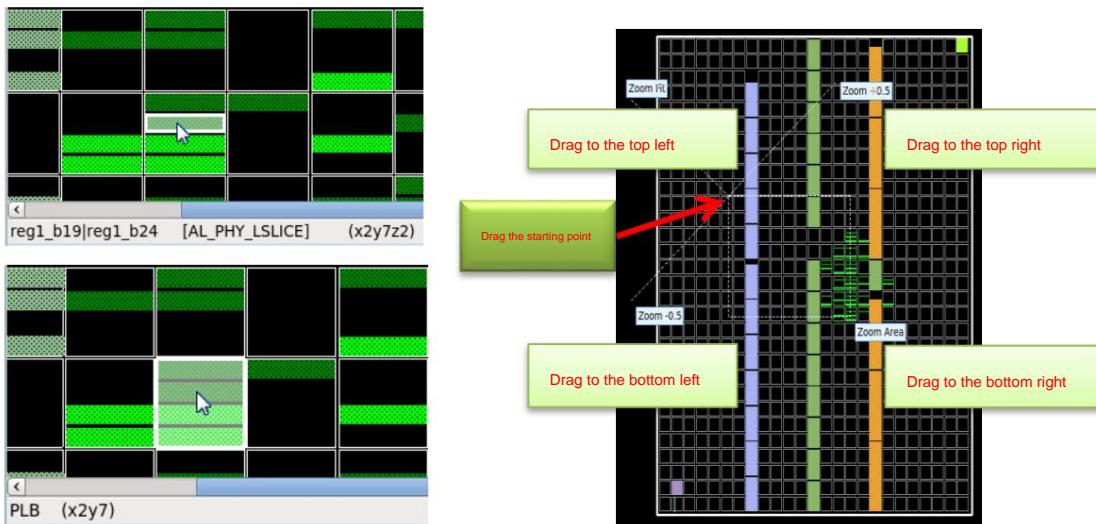
picture

Mouse drag (with CTRL) -- pan the current view, equivalent to PAN mode

mouse drag

-- area zoom (zoom area), fixed ratio zoom (such as zoom +0.5),

Fixed scale reduction (such as zoom -0.5), return to the default view (zoom fit), as shown in the lower right figure

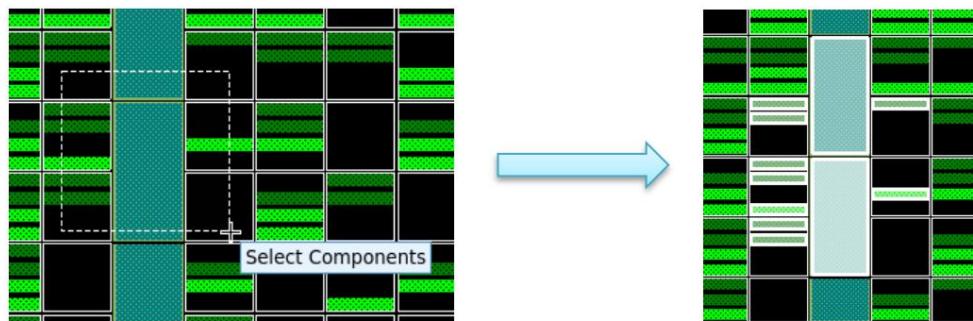


b) AREA_SELECTION mode:

mouse drag

-- Select the cell in the wireframe, when CTRL is pressed, multiple selections can be made; when entering

In regional constraint management mode, a new bound can be created based on the cells selected by the wireframe



c) PAN mode

mouse drag

-- pan the current view

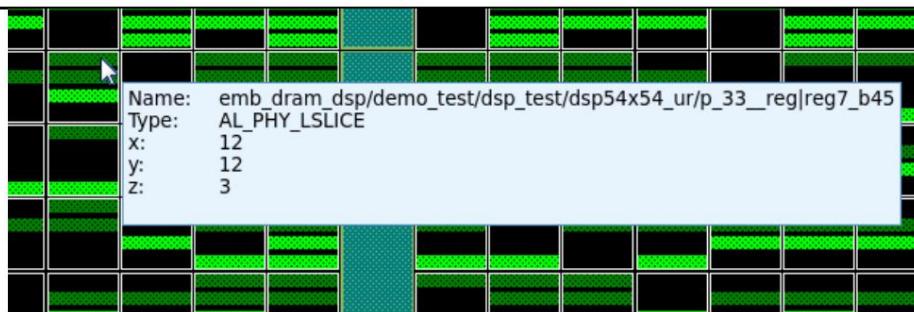
When the mouse wheel is used, scrolling up and down corresponds to vertical scrolling of the view; when the CTRL key is held down, the mouse wheel

Scrolling up and down corresponds to the zoom in/out of the view; while holding down the SHIFT key, scrolling the mouse wheel up and down corresponds to the horizontal view

direction scrolling. Hover over the instance or slot to display the tooltip.

TangDynasty

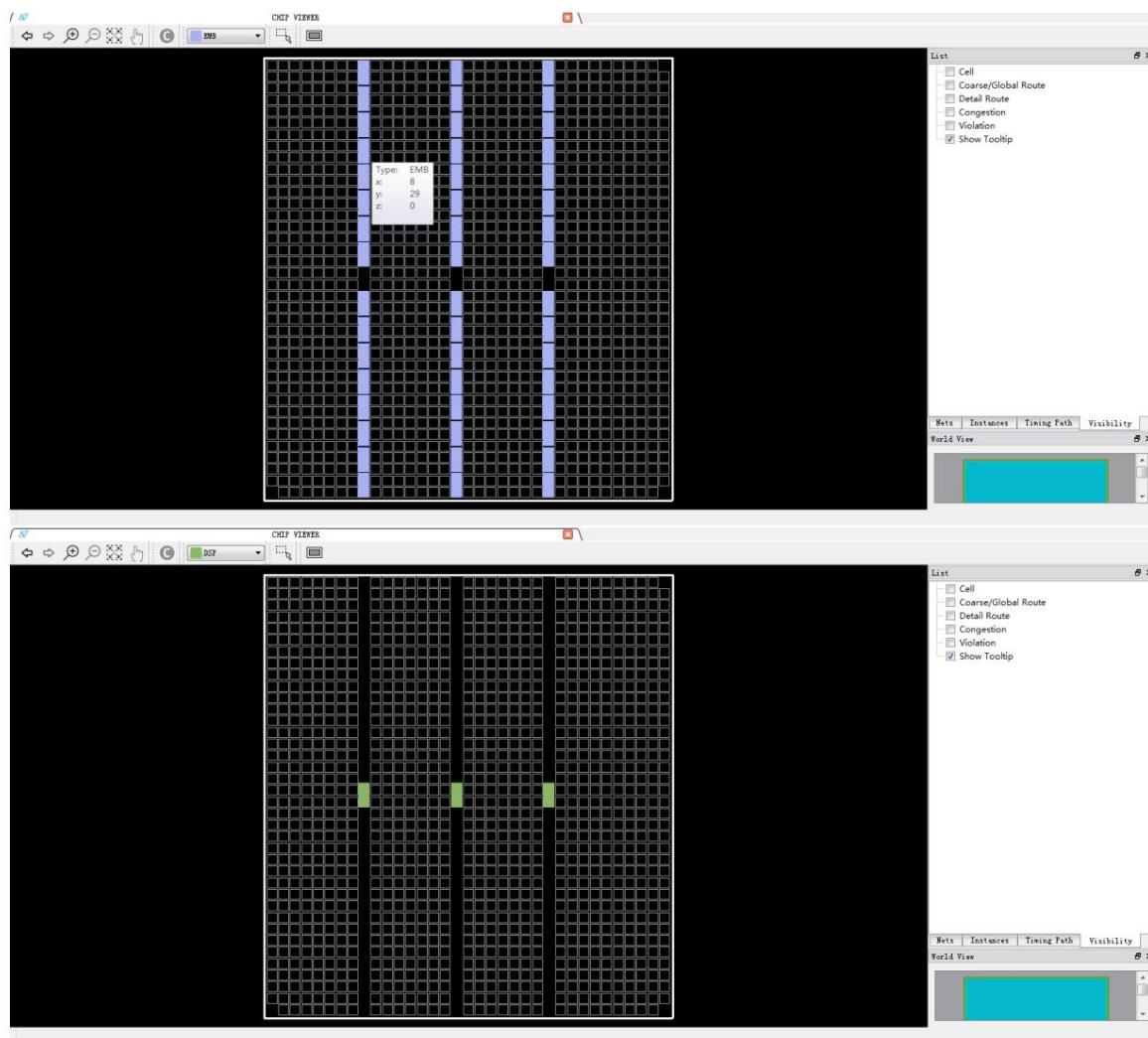
Software Manual



In the menu bar of the Chip Viewer, you can choose to display the location of all physical resources in the device, such as:

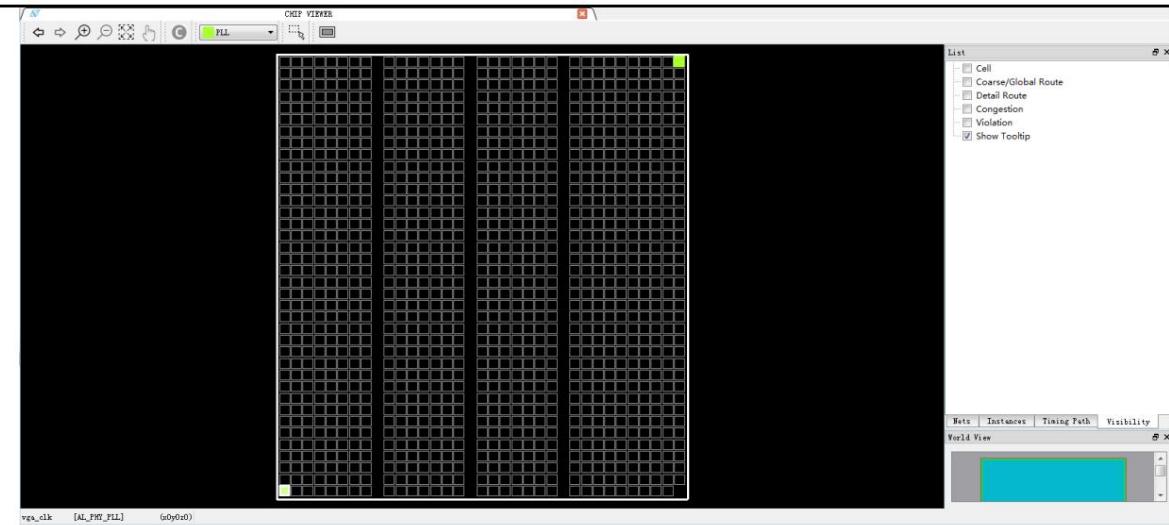
PLL, EMB, EMB32K, DSP. Click a module to display its physical coordinates in the lower left corner or hover the mouse

When you stop to a module, a floating window will display its physical coordinates.



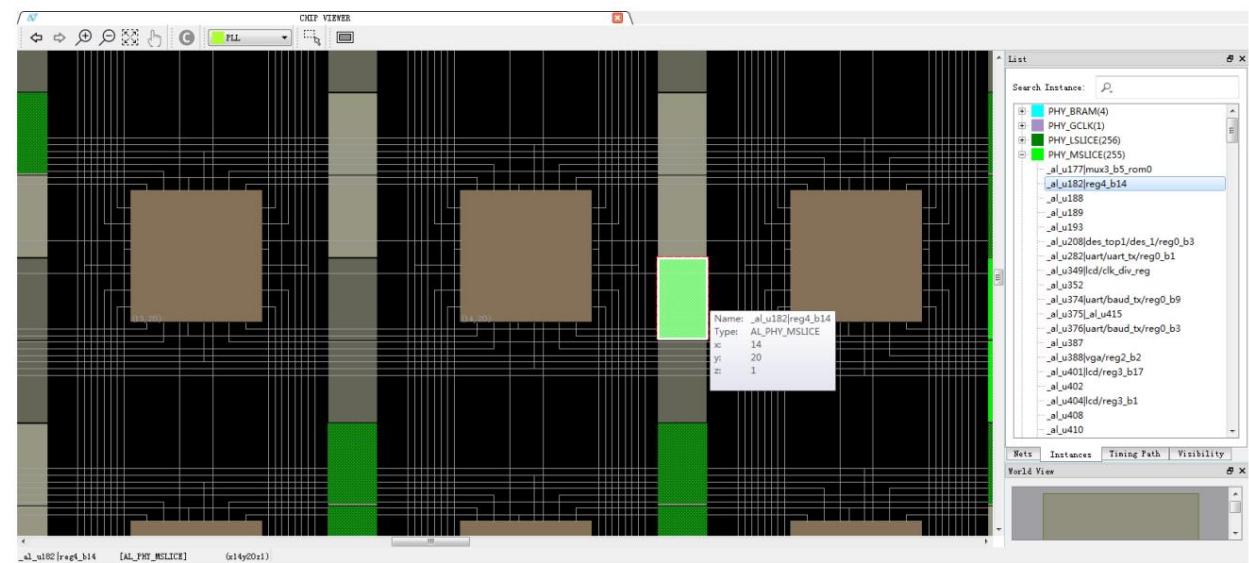
TangDynasty

Software Manual



A plb contains 4 slices, the z-coordinate of mslice is 0,1; the z-coordinate of lslice is 2,3. as shown below

The coordinates of the mslice shown are (x14y20z1).

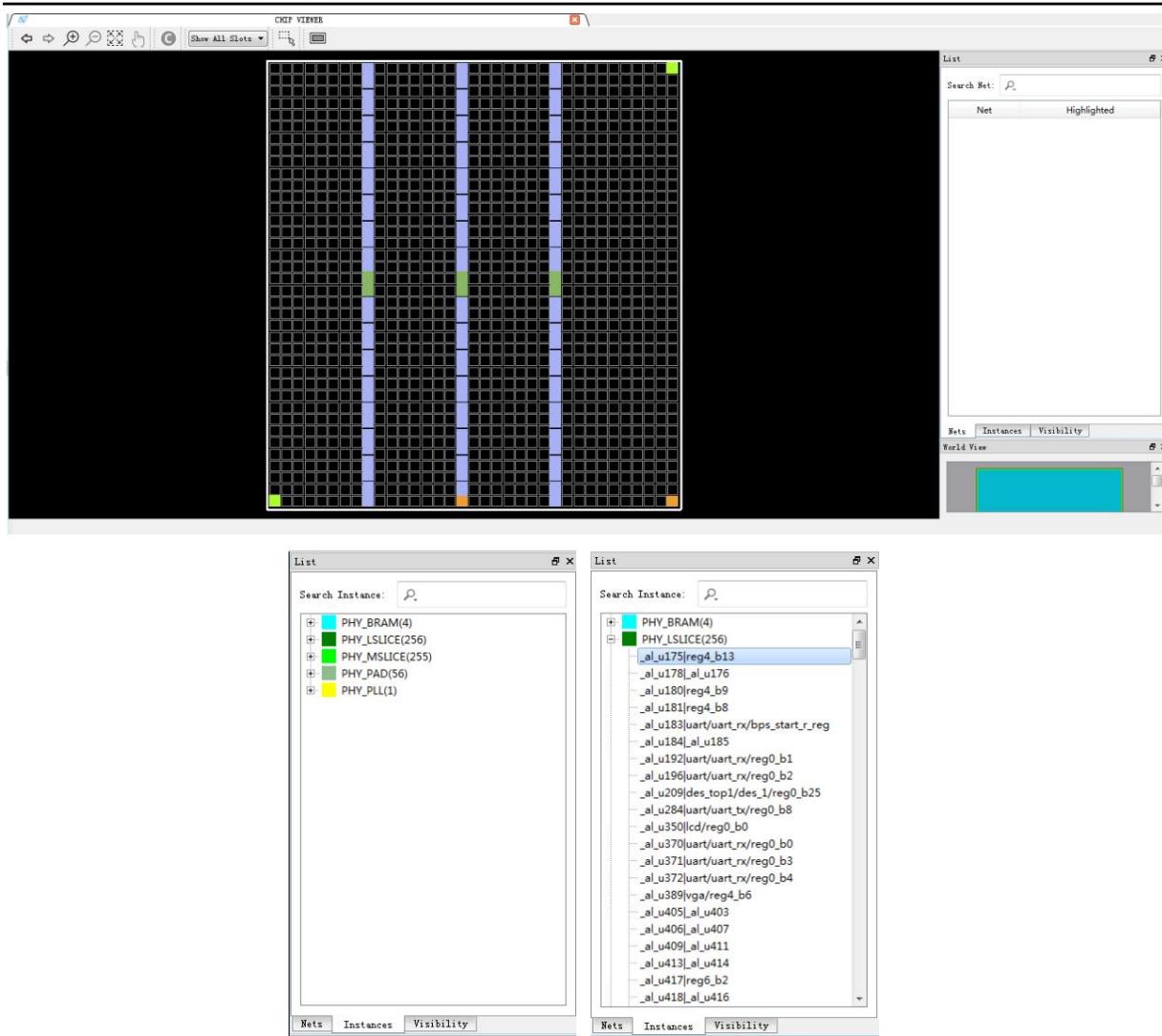


When Flow finishes running **Optimize Gate**, open the **Chip Viewer** to see the information of instances and

And the location of all physical resources of the selected device, such as: PLL, EMB, EMB32K, DSP.

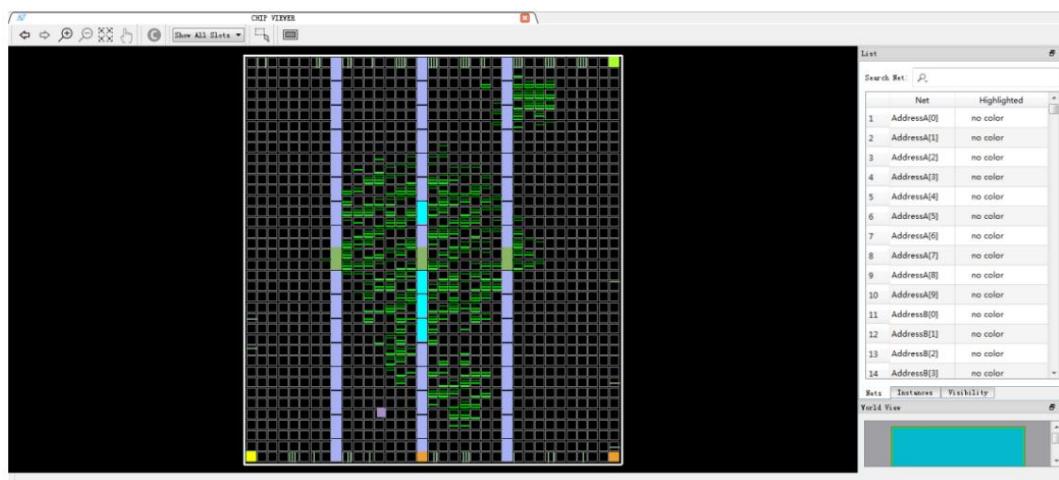
TangDynasty

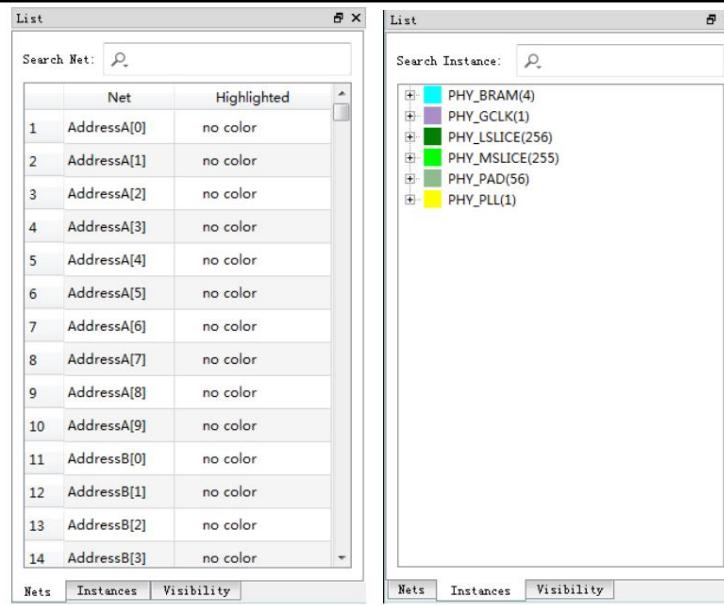
Software Manual



After Flow runs **Optimize Placement**, in addition to the information that can be seen by **Optimize Gate**, it also

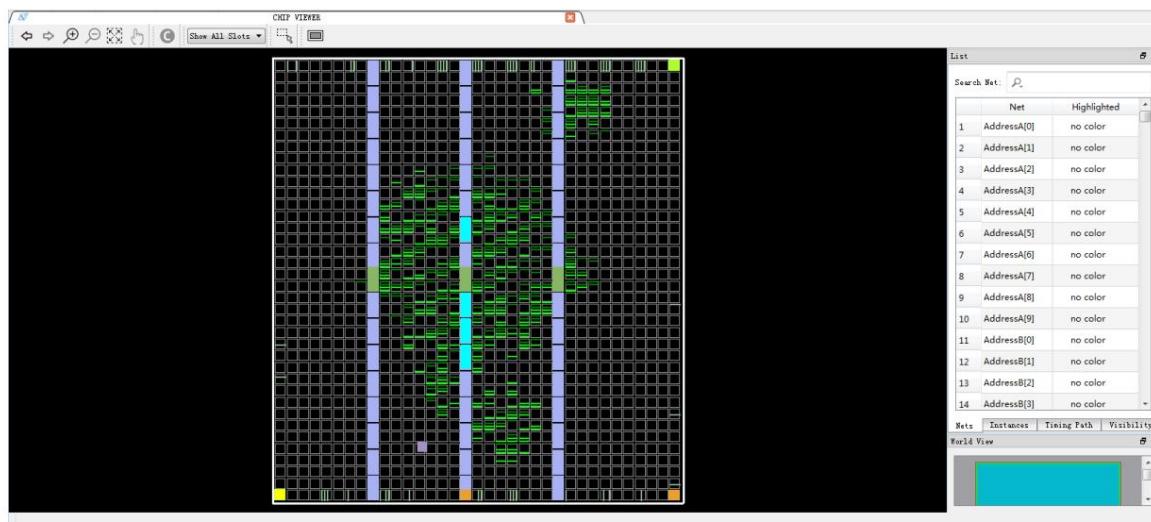
You can see the information of All Nets.

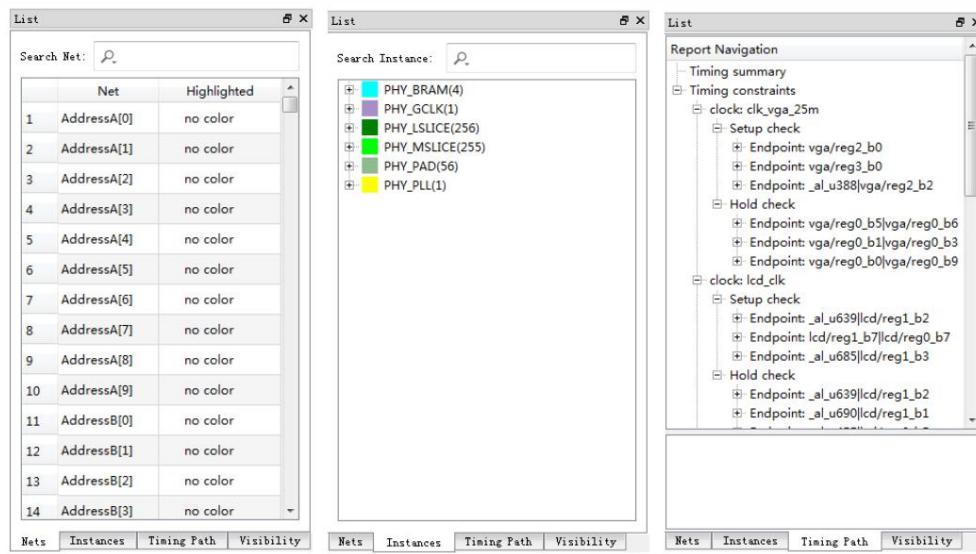




After the **Optimize Routing** is completed, the user can view the detailed physical implementation through **ChipViewer**

Information: Includes Cells Utilization, Global Routing, Detail Routing, All Nets, Timing Path.

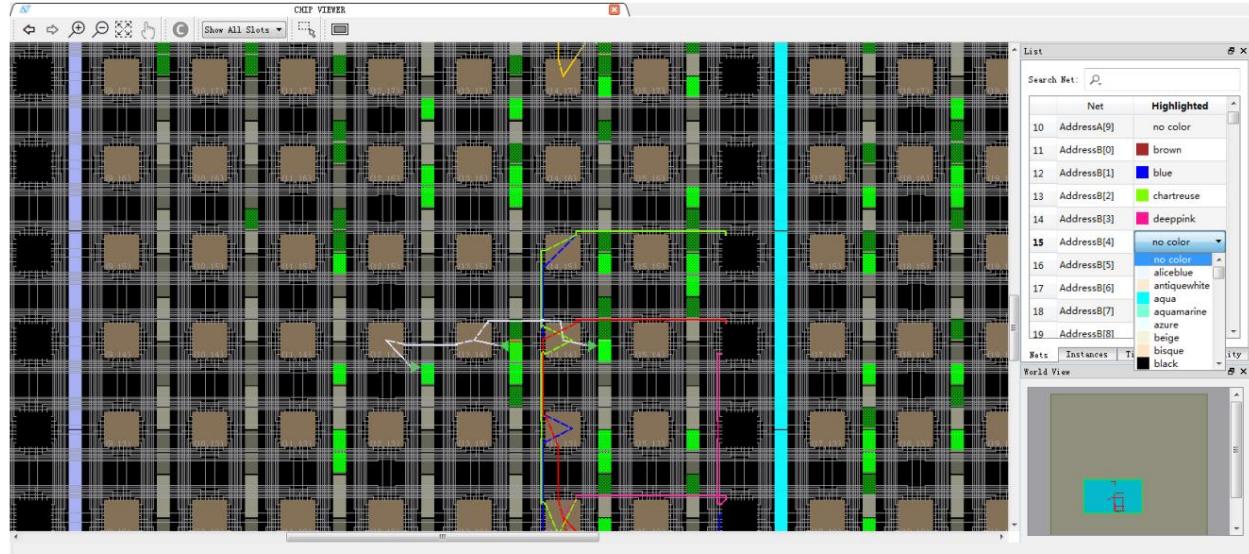




In the **Nets** column, you can view all the nets in the design. Double-click the net to display the specific nets in the left window.

In the case of wiring, you can choose a different color in the **Highlighted** column to keep the net highlighted. for a

net, the triangle arrow facing outward from plb is the source end, and the triangle arrow facing plb is the sink end.



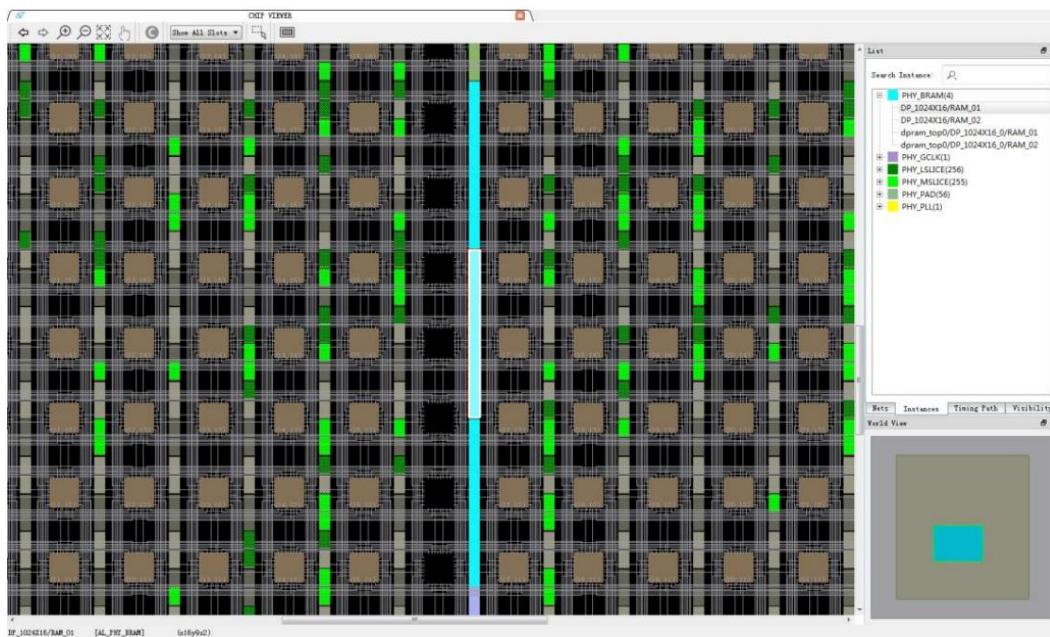
In the Find box the net can be searched.

Net	Highlighted
1 AddressA[0]	no color
2 AddressA[1]	no color
3 AddressA[2]	no color
4 AddressA[3]	no color
5 AddressA[4]	no color

Nets Instances Timing Path Visibility

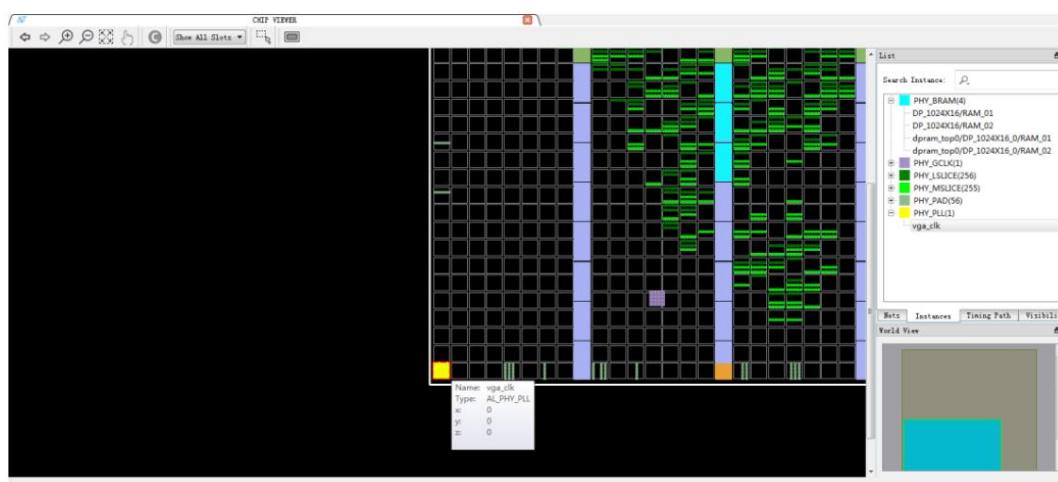
In the **Instance** column, you can view all physical modules in the design. Double-click the instance to display it in the left window.

its layout position. Clicking on the module will display its physical coordinates in the lower left corner.



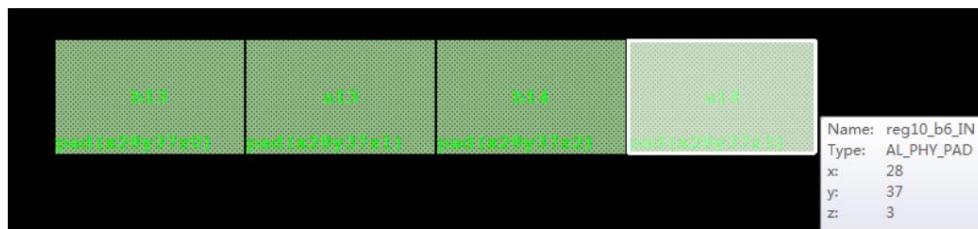
When the mouse hovers over a module that has been used in design, a floating window will display the information of the module.

information, including name, type, and physical coordinates.



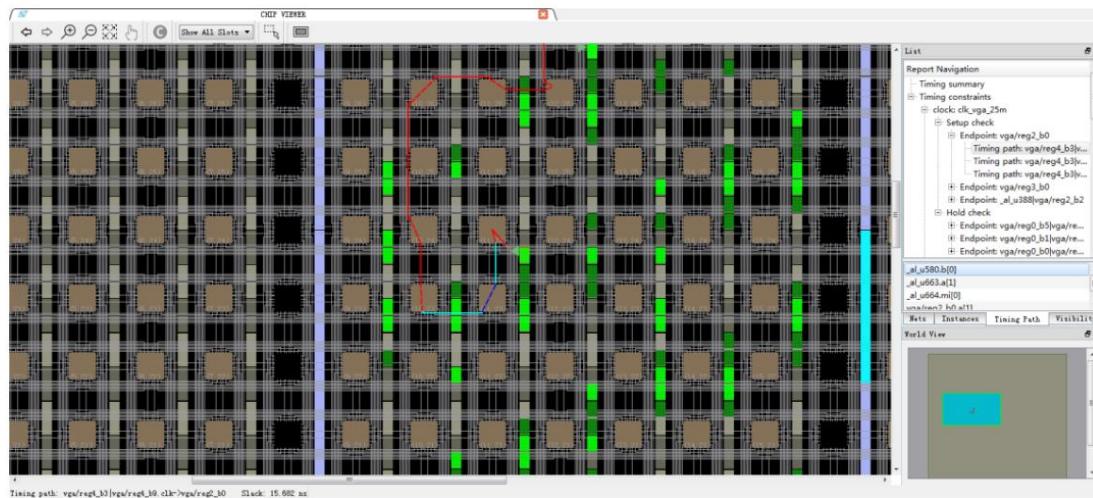
Enlarging the view, you can see that each pad corresponds to the location inside the package, which is convenient for users to constrain,

As shown in the figure below, the corresponding locations are: B13, A13, B14, A14.



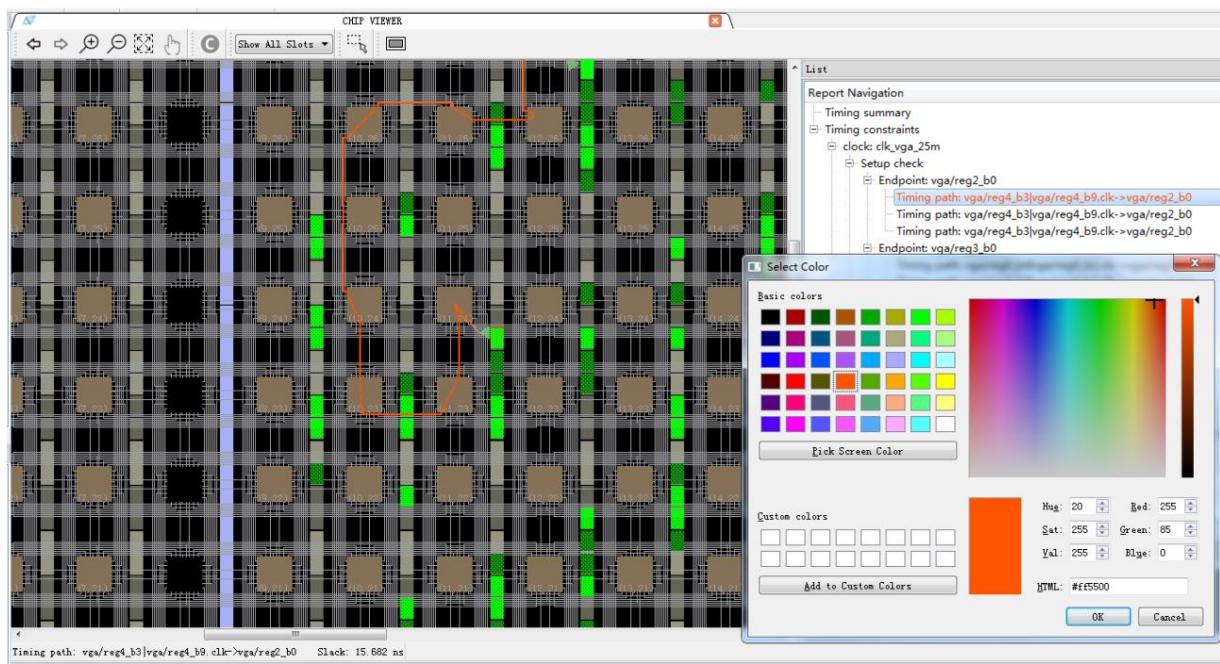
In the **Timing Path** column, you can view all the timing paths listed in the timing report, double-click the timing path can display its detailed routing in the left window. Double-click a net in the path to highlight it.

The lower left corner will display the path of the timing path (start point & endpoint) and the slack of the path.

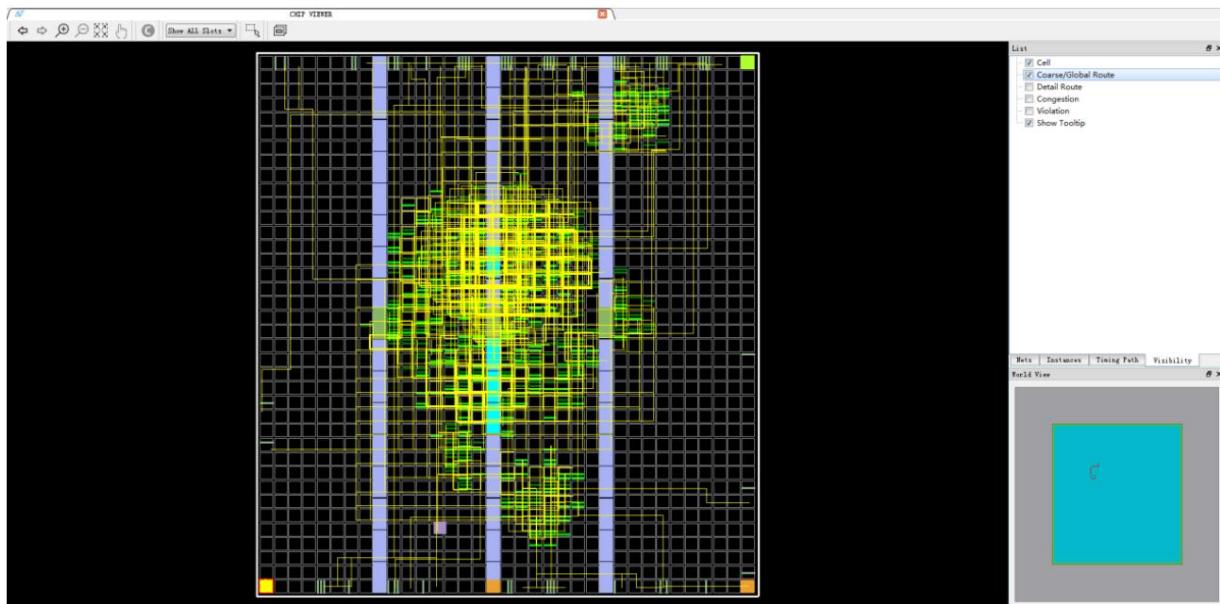


Similarly, select a timing path, right-click, select keep, and select a color to add

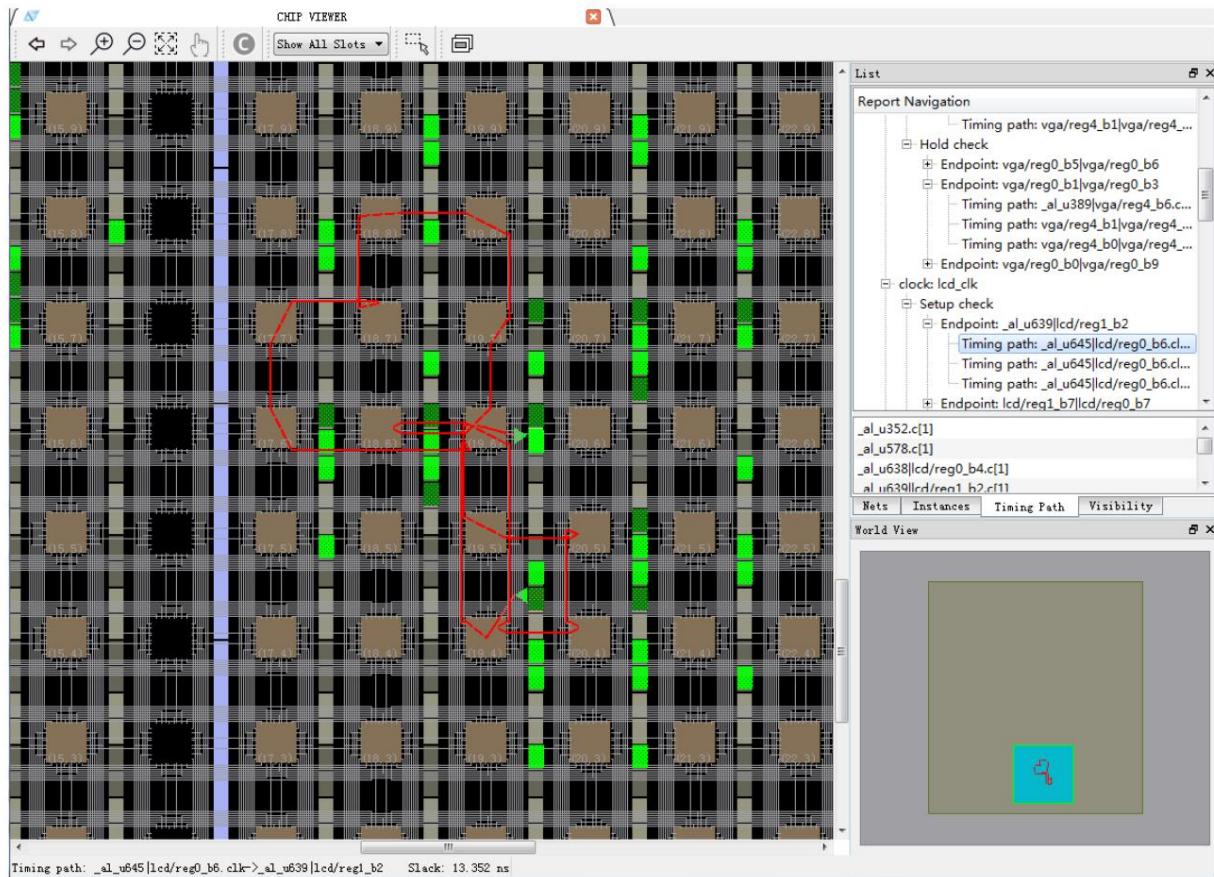
The line is highlighted and the path is kept displayed, that is, multiple timings can be displayed at the same time in the Chip Viewer path or net.



In the Visibility column, you can choose to display basic elements, global routing or detailed routing.



Shown in the lower right corner is a bird's-eye view in the Chip Viewer.



8.2.2 Region Constraint

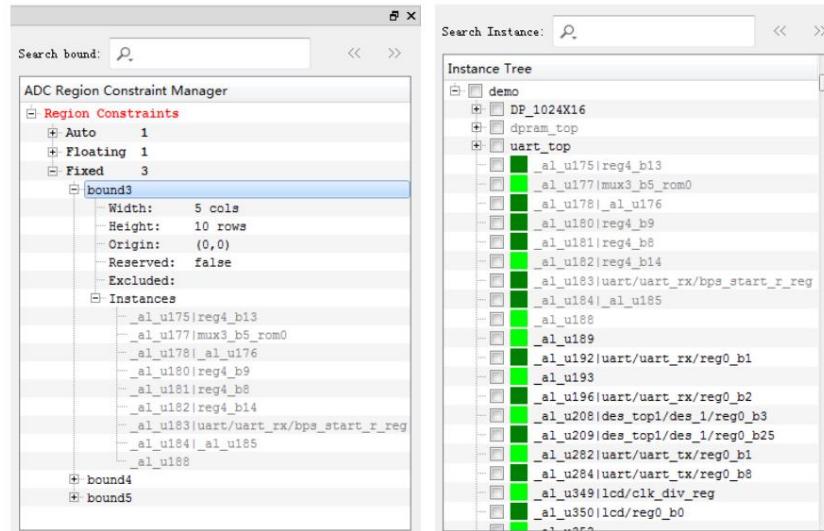
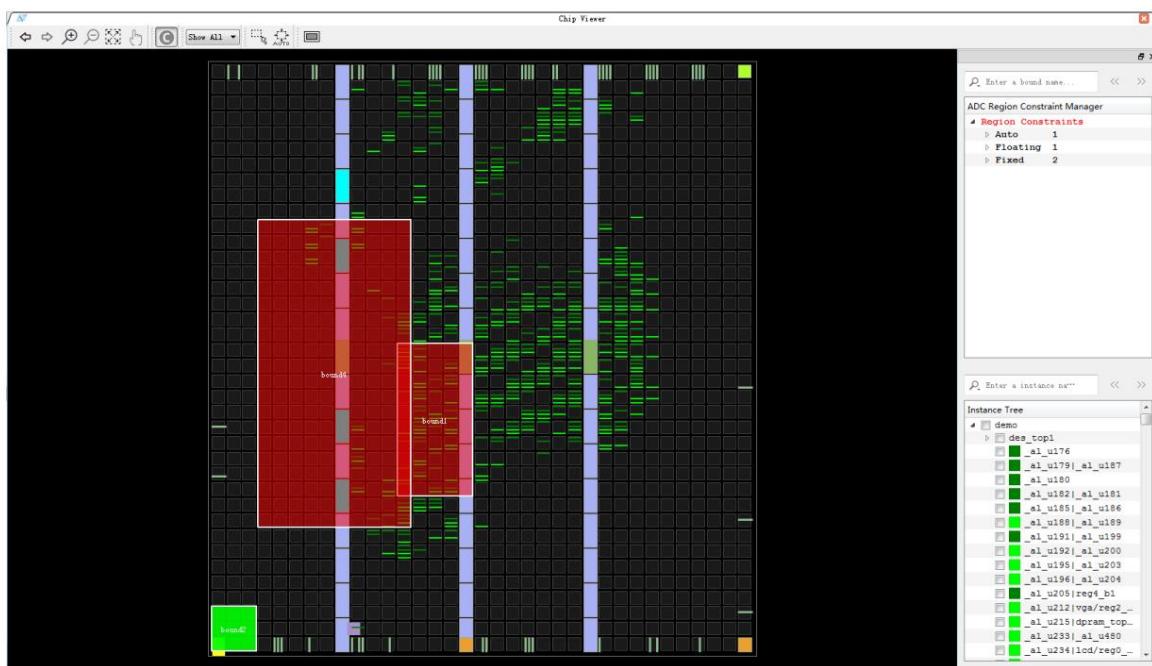
After pressing the **button in the toolbar**, the Chip Viewer enters the regional constraint management mode.

The constraint manager will display all bounds defined in the ADC file by category, and the instance tree at the bottom right will be sorted by layer

The level displays the logical resources occupied by each unit, and you can select a group of instances to add to the new bound or the existing ones.

bound. The main function of the constraint manager is to add, delete, and modify bounds.

The top-level interface of the Region Constraint Constraint Manager is displayed as shown in the following figure:



Bound mode is mainly divided into three categories: auto, floating, fixed. Among them, auto mode will automatically be

Bound allocation position; floating mode will fix bound width and height, floating allocation position; fixed

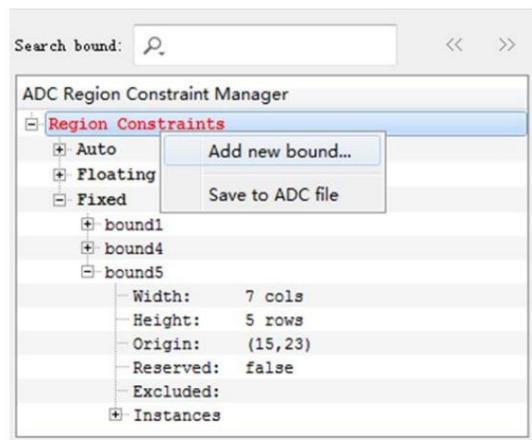
mode is to fix the bound at a given size and position. Expanding bound will show all attributes and add

All instances that come in. All bounds (red) whose mode is Fixed will be displayed on the Layout and highlighted

Displays the currently selected bound (bright red); all modes whose mode is Floating (green) and highlight the current

The bound (bright green) of the previous selection.

If you want to create a new bound, you can do it through the right-click menu on the constraint manager.



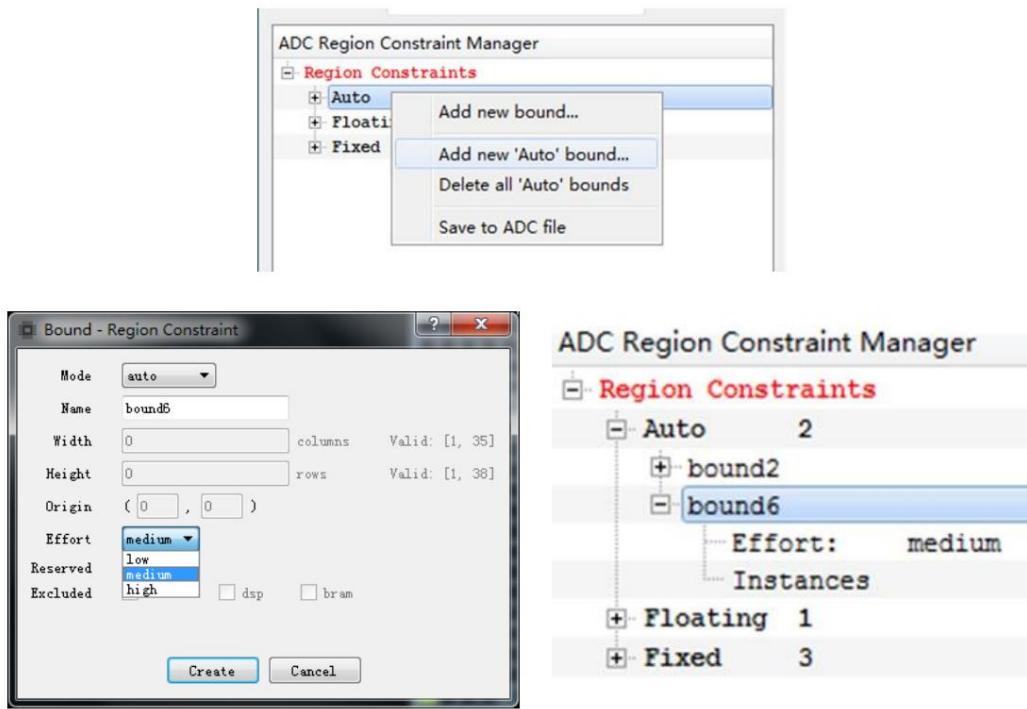
Add new bound..., the default mode is auto, the bound name will be given automatically, or the user can do it himself

defined, but the name is unique. The optional properties of bound are different in different modes.

When mode is auto, all other attributes except effort cannot be changed, and effort can be low,

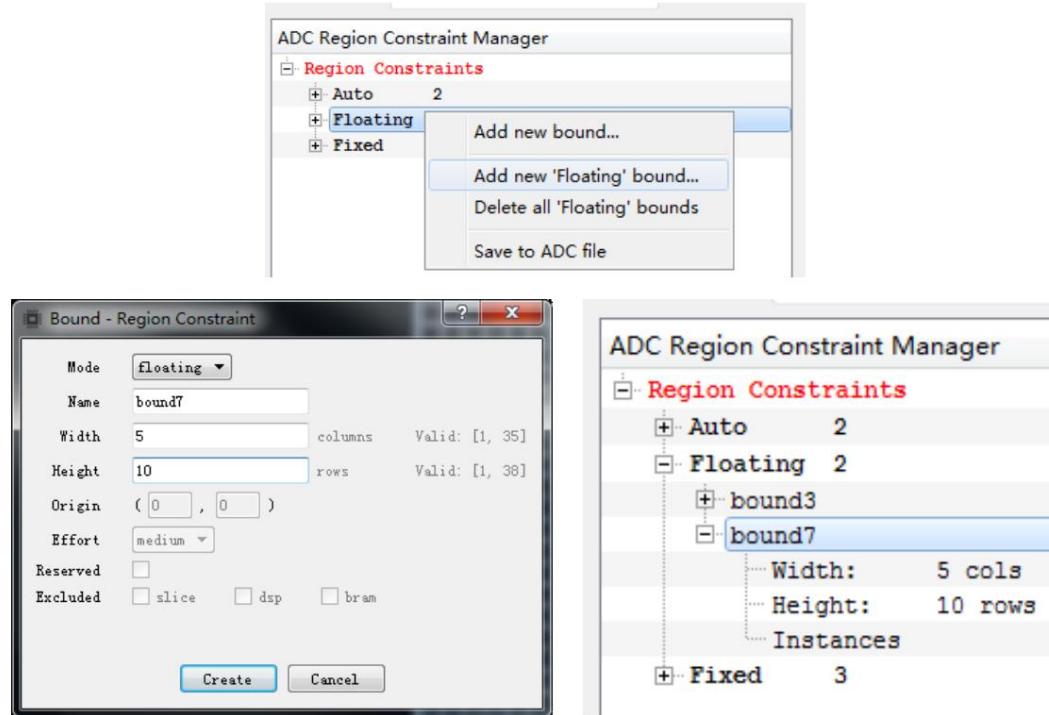
medium, high, the default is medium. Click create to see the newly created under the corresponding mode

bound.



When mode is floating, other properties except width and height cannot be changed, width and height

The range of height depends on the selected device family and package. Click create to see the new creation under the corresponding mode bound.



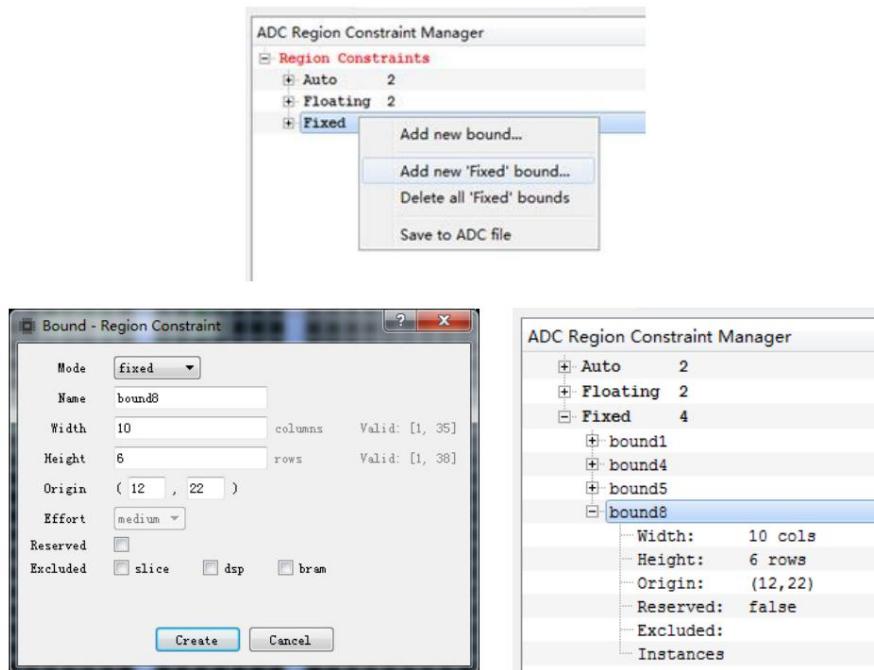
When mode is fixed, all other attributes can be filled in as needed except for the attribute of effort, which cannot be changed.

The range of and height depends on the selected device series and package, origin is the starting position coordinate (left to the bound area).

lower corner). When executing area constraints, if reserved is checked, the area will be reserved, and excluded is checked

slice, dsp, bram will exclude the tick option in the selected instances. Click create to be in the corresponding mode

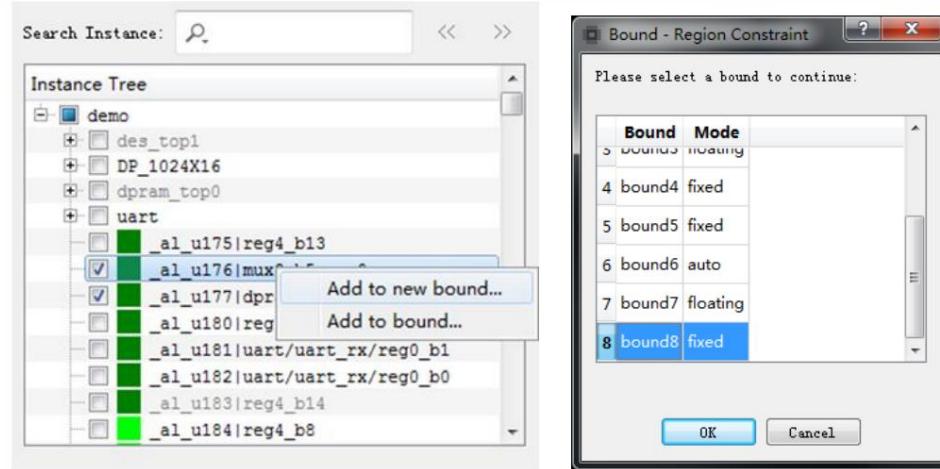
See the newly created bound.



After completing the new bound, you can check the required instances in the instance tree, and right-click to select add

to new bound, create a new bound and add the currently selected instances; or right-click and select add to

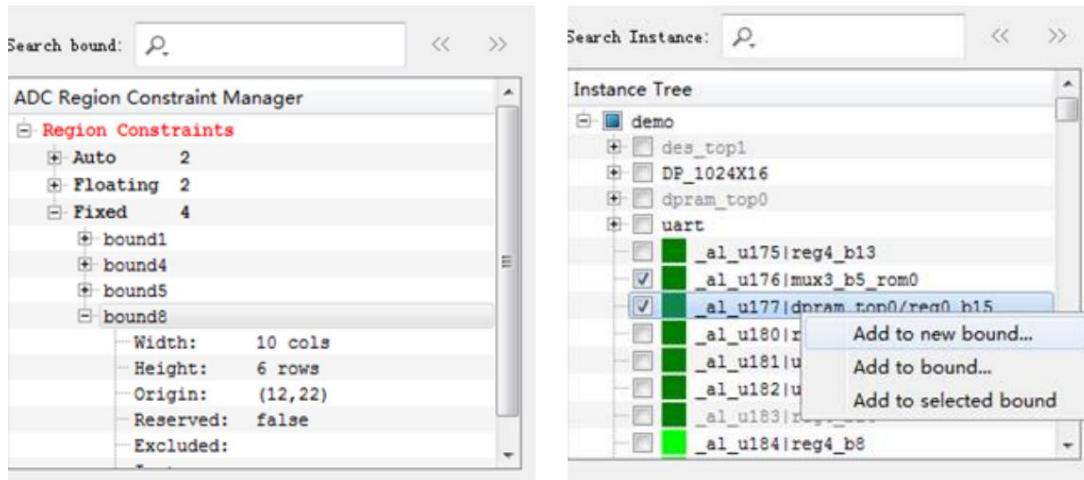
bound, selects an existing bound and adds the currently selected instances.



For bound in fixed mode, the number of instances of various types cannot exceed the number specified by

The number of the area defined by width, height, origin can hold. Right-click menu when several instances are selected

add to selected bound. Appears only if the manager's current selection is bound.



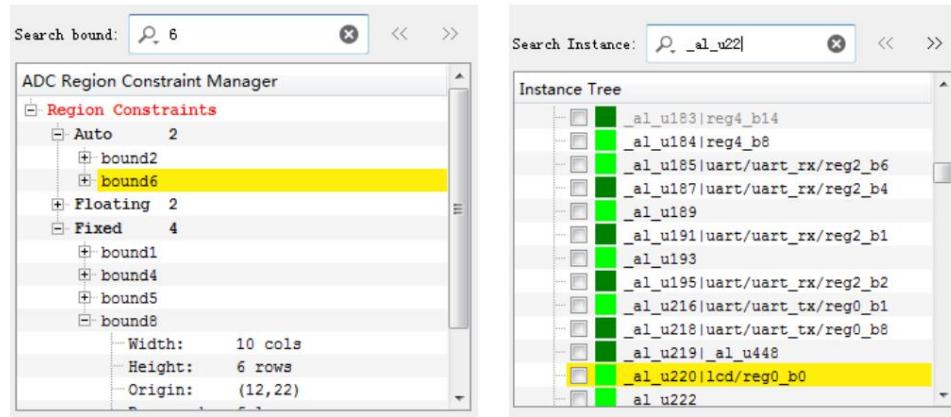
There is a search box above the ADC Region Constraint Manager/Instance tree, which supports searching by keywords

way to list the required bound/instances. The search criteria can be set by yourself. The default search criteria are as follows:



After typing in the search box, it will jump to the first bound/ instance that matches the search criteria and the

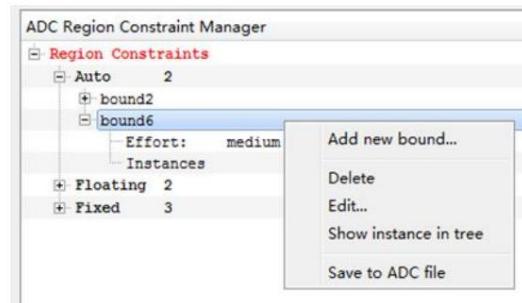
bound/instance will be marked in yellow.



Selecting a bound right-click can perform the following operations: Delete to delete the current bound; Edit... to edit the current

bound property; Show instance in tree displays the instance setting of the current bound in the instance tree

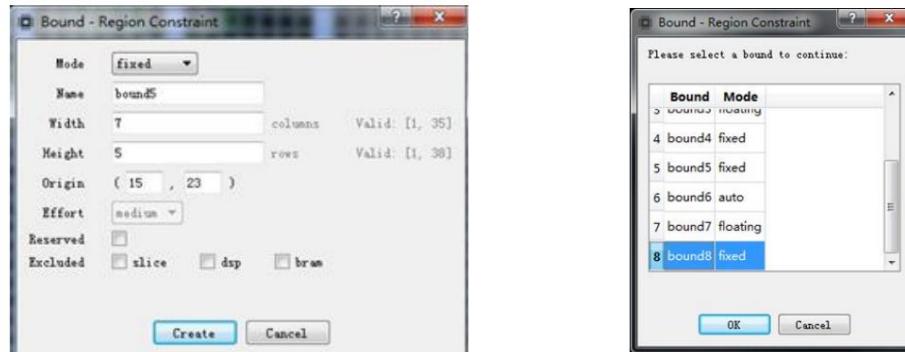
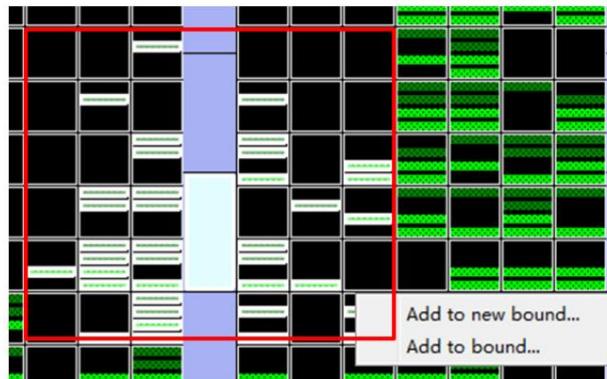
set.



In addition to the right-click menu on the constraint manager, the new bound can also be created in the AREA_SELECTION mode.

It is carried out by box selection under the mode, and the default mode is Fixed. After the box is selected, a menu bar will appear in the lower right corner and select New

bound or added to an existing bound.



After completing the region constraints, you can select any column in the ADC Region Constraint Manager, right-click

Select save to ADC file from the menu to write the area constraint settings into the adc file.

```
demo256.adc
49 set_pin_assignment {lcd12864_data[6]} { LOCATION = K9; }
50 set_pin_assignment {lcd12864_data[5]} { LOCATION = P9; }
51 set_pin_assignment {lcd12864_data[4]} { LOCATION = M10; }
52 set_pin_assignment {lcd12864_data[3]} { LOCATION = R16; }
53 set_pin_assignment {lcd12864_data[2]} { LOCATION = L9; }
54 set_pin_assignment {lcd12864_data[1]} { LOCATION = T15; }
55 set_pin_assignment {lcd12864_data[0]} { LOCATION = N9; }
56 set_pin_assignment {lcd12864_rs} { LOCATION = R6; }
57 set_pin_assignment {lcd12864_rw} { LOCATION = T6; }
58 set_pin_assignment {lcd12864_en} { LOCATION = R5; }
59 #set_pin_assignment {test_out} {location=T5;}
60 set_pin_assignment {clk_vga_25m} { LOCATION = T4; }
61 #set_pin_assignment {pllclk_div} {location=R4;}
62
63 #Region Constraint
64 create_bound bound1 -mode fixed -width 10 -height 10 -origin {20 10}
65 add_cells_to_bound -bound bound1 -cells des_top1
66 create_bound bound2 -mode auto -auto_effort medium
67 add_cells_to_bound -bound bound2 -cells uart_rx
68 create_bound bound3 -mode floating -width 5 -height 14
69 add_cells_to_bound -bound bound3 -cells dpram_top0
70 create_bound bound4 -mode fixed -width 7 -height 6 -origin {22 30}
71 add_cells_to_bound -bound bound4 -cells {beep_reg reg0_b10|reg0_b8 reg
72 add_cells_to_bound -bound bound4 -cells {reg0_b1|reg0_b2 add0|u7 al_u6
73 add_cells_to_bound -bound bound4 -cells {al_u572|al_u548 al_u573_a
74 add_cells_to_bound -bound bound4 -cells {al_u552|al_u553 al_u559_a
75 add_cells_to_bound -bound bound4 -cells {al_u531|al_u532 al_u540|_a
76 create_bound bounds5 -mode fixed -width 7 -height 5 -origin {15 23}
77 add_cells_to_bound -bound bound5 -cells {al_u183|reg4_b14 uart/baud_r
78 add_cells_to_bound -bound bound5 -cells {vga/reg0_b0|vga/reg0_b9 vga/r
79 add_cells_to_bound -bound bound5 -cells {al_u292|al_u293 al_u439|_a
80 add_cells_to_bound -bound bound5 -cells {al_u490|al_u258|vga/reg2_b2
81 create_bound bound6 -mode auto -auto_effort medium
82 create_bound bound7 -mode floating -width 5 -height 10
83 create_bound bound8 -mode fixed -width 10 -height 6 -origin {12 22}
84
```

If you close Chip Viewer directly, if the area constraints are not saved, there will be a pop-up prompt to save to the specified adc

document. If the Chip Viewer is open and the region constraints are not saved, running the flow directly will close the Chip

Viewer, and save the area constraint settings directly to the specified adc file, no more pop-up prompts.

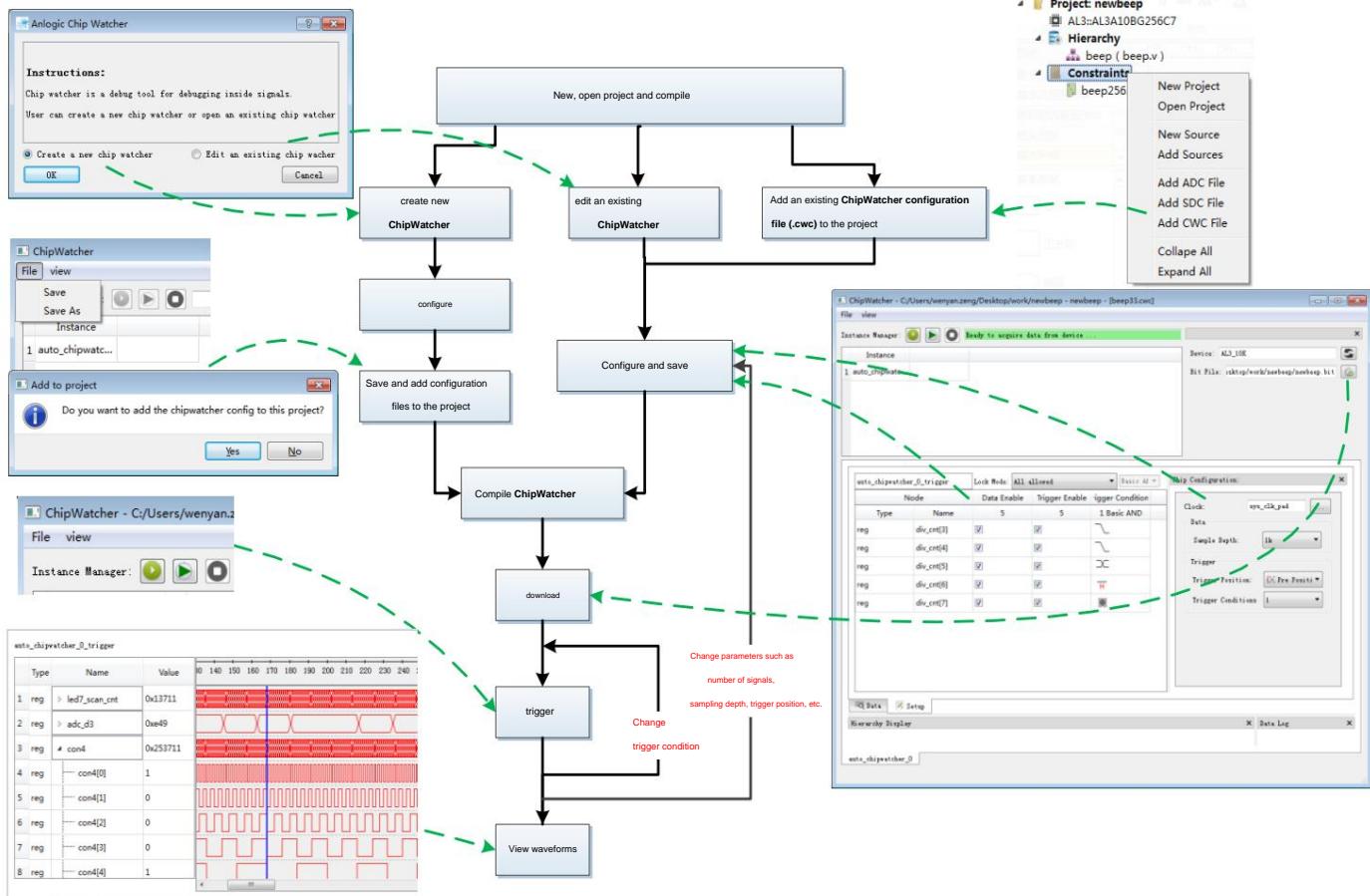
8.3 ChipWatcher

With ChipWatcher, users can monitor the changes of the signal inside the circuit online without the need of external equipment.

In **ChipWatcher**, users can add multiple signals at the same time, after setting the sampling clock, sampling depth, touch

After sending the conditions and trigger positions, after recompiling, downloading and triggering, you can view the signal changes under the specified conditions

condition. The workflow of **ChipWatcher** is shown in the figure below.

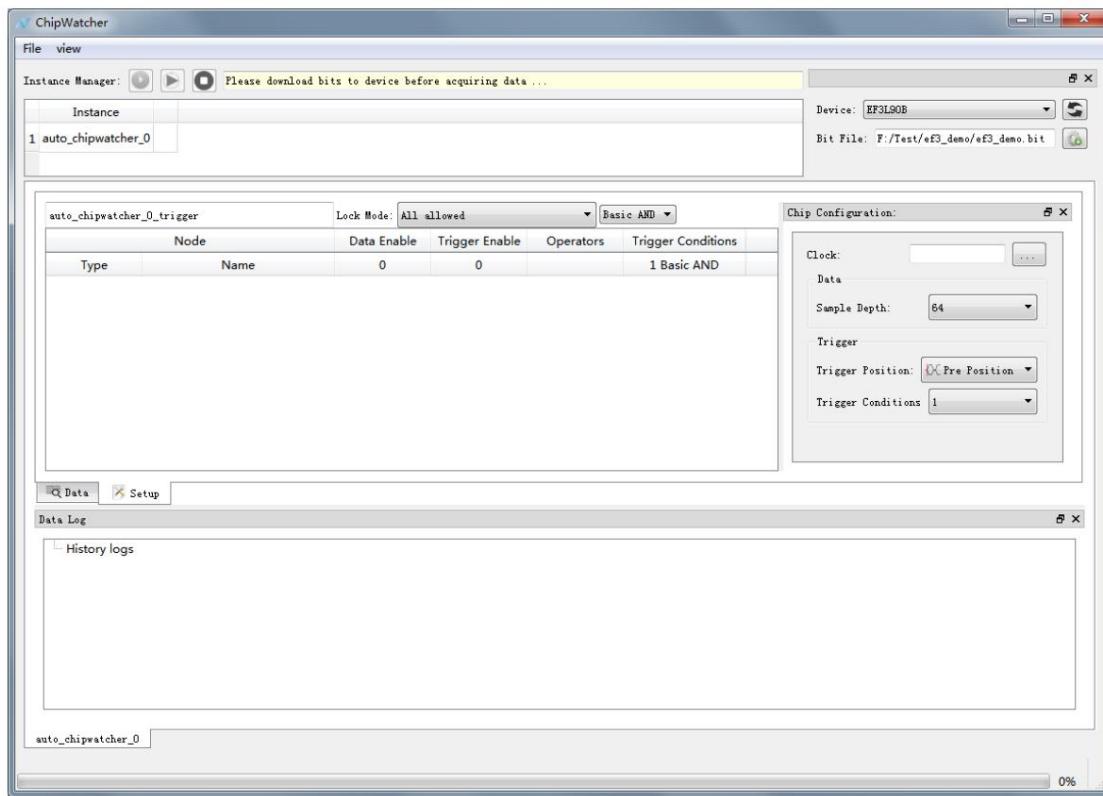


1. After running the HDL2Bit process, there are three ways to start ChipWatcher:

- 1. Create a new ChipWatcher

Expand **Tools** → **Debug Tools**, select **ChipWatcher**, and select "Create a new" according to the prompt

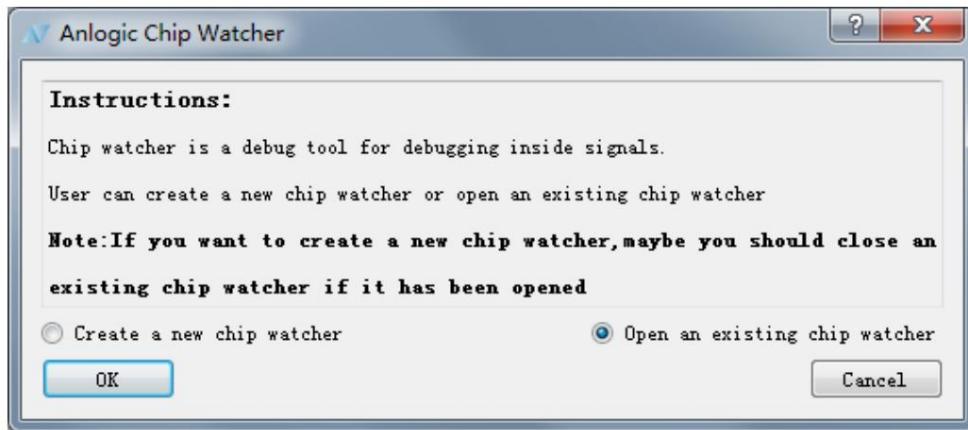
chip watcher", click "OK", the following main interface of ChipWatcher appears:



•2. Edit an existing ChipWatcher

Expand **Tools** → **Debug Tools**, select **ChipWatcher**, and select "Open an

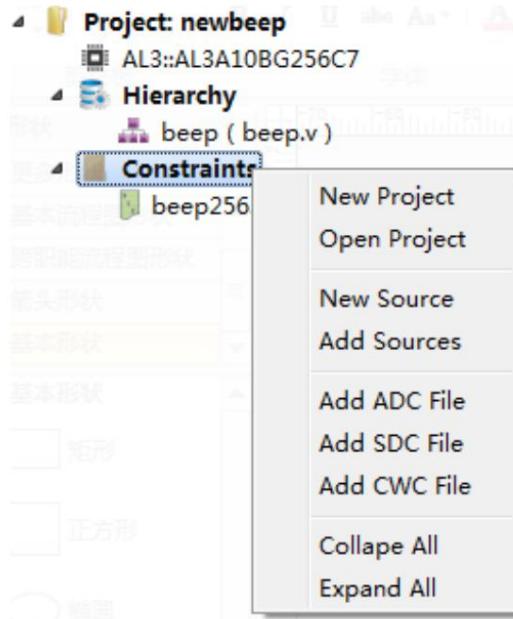
existing chip watcher", click "OK" to enter the main interface of ChipWatcher.



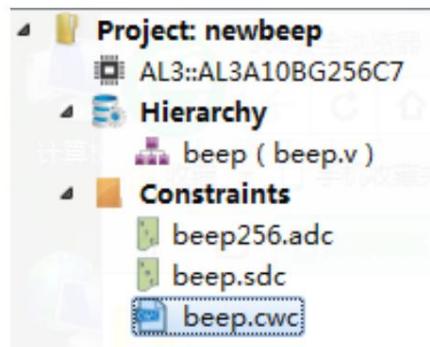
•3. Add an existing ChipWatcher configuration file (.cwc) to the project and double-click to open it.

In the **Project** column, right-click **Constraints**, select "Add CWC File", and add to the project

Generated configuration file (.cwc).



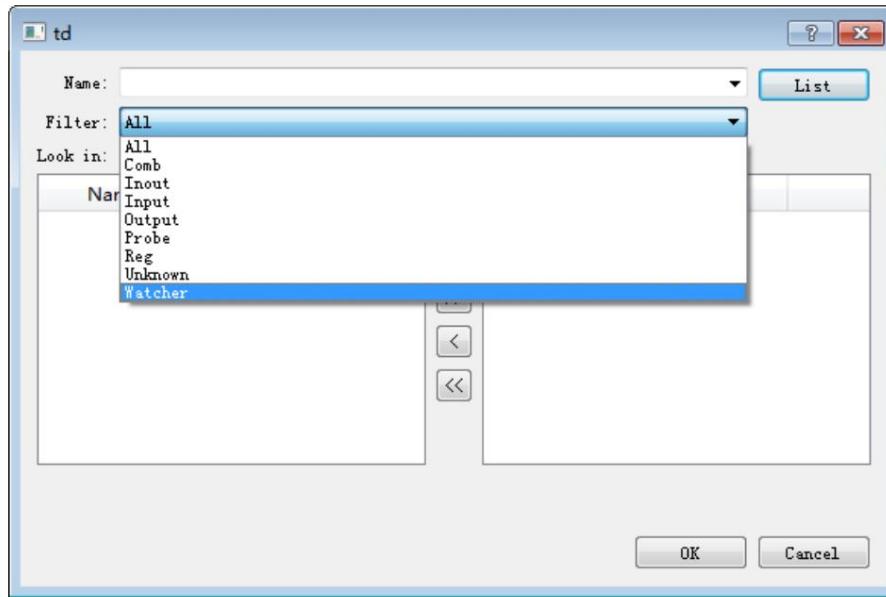
Double-click the file to open the main interface of ChipWatcher.



2. Right-click in the blank space of the **setup** interface, select "Add Node...", the following Node Filter interface will appear

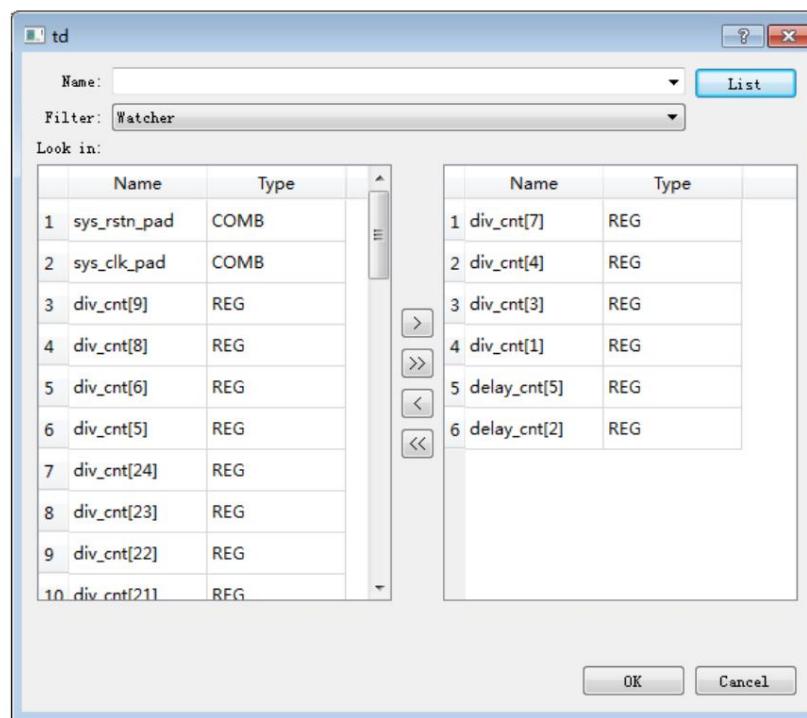
surface, expand the **Filter** drop-down menu, the default type is **Watcher**. If the user selects the filter type as

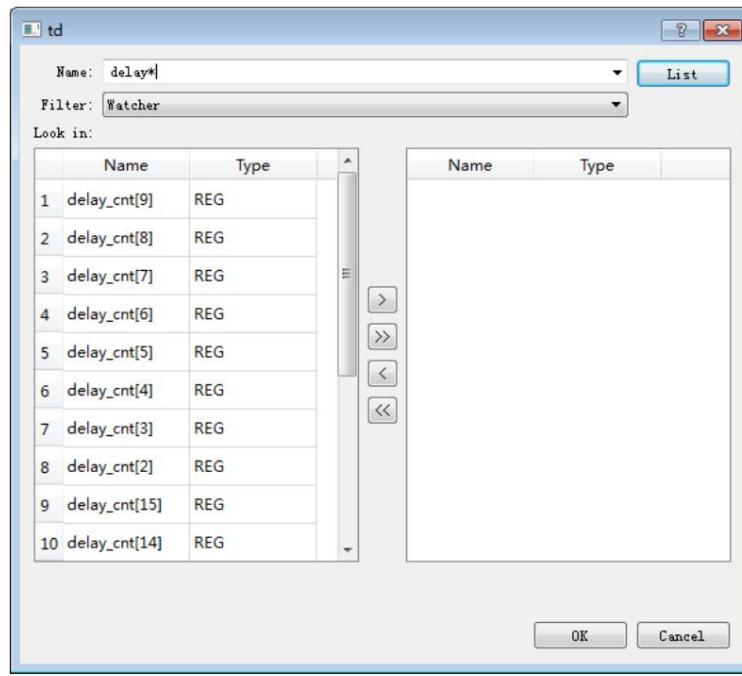
Signals outside the Watcher will not be guaranteed to be sampled correctly;



3. Use the left and right arrows to select or delete signals, and you can also search for signals in the **Name** column (supports wildcards)

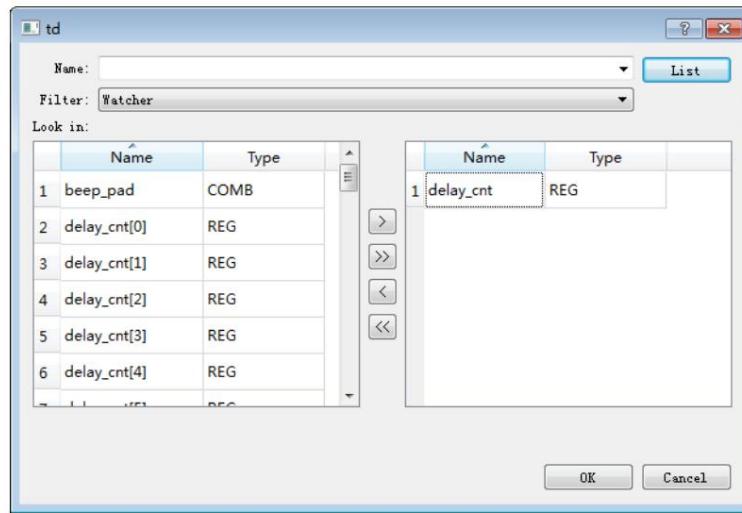
symbol*);





You can add net or bus signal as needed, click the OK button to complete the addition of the signal

*Note: The total number of buses cannot exceed 8, and the width of a single bus cannot exceed 64.



4. For the added net or bus, right-click the signal name to perform **Delete/group/ungroup** operations .

operation; a single signal or a whole group of buses can be directly deleted;

Type	Name
reg	delay_cnt
reg	div_cnt[0]
reg	div_cnt[1]
reg	div_cnt[2]
reg	div_cnt[3]
reg	div_cnt[4]

At the same time, in order to facilitate the observation of signals, **group** operations can be performed on multiple network signals ;

Type	Name
reg	delay_cnt
reg	div_cnt[0]
reg	div_cnt[1]
reg	div_cnt[2]
reg	div_cnt[3]
reg	div_cnt[4]

Type	Name
reg	delay_cnt
gp	div_cnt[0]_group
reg	div_cnt[1]
reg	div_cnt[3]

Upgroup operation can be performed for bus or group . At the same time, the mouse can select net, bus or group to

Drag and drop it to adjust the order of the current Node.

Node		Data E
Type	Name	
reg	delay_cnt	
gp	div_cnt[0]_group	
reg	div_cnt[1]	
reg	div_cnt[3]	

auto_chipwatcher_0_trigger		Lock Mode:	All allowed	Basic AND	
Node		Data Enable	Trigger Enable	Operators	Trigger Conditions
Type	Name	35	35		1 Basic AND
reg	emb_dram_dsp/demo_test/dp8...	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
gp	cnt[25]_group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	==	0000000000000000...
reg	cnt[25]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
reg	cnt[24]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
reg	cnt[23]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		

Data Setup

5. **Data Enable** means to select the signal that needs to acquire and display the waveform, tick the check box to enable it

The signal; Trigger Enable refers to a certain state of the signal as a trigger condition; Trigger

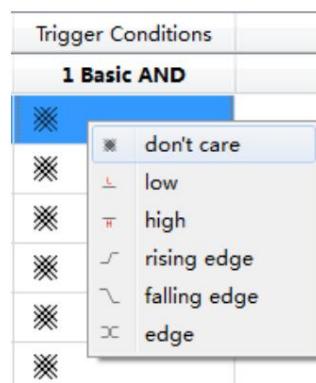
Conditions means that the signal can be triggered only when the condition is met; **Basic AND** means that the signal needs to be triggered at the same time .

The signal can be triggered only when all of the following trigger conditions are met; **Basic OR** means that as long as the following conditions are met:

A trigger condition can trigger on a signal. Right-click the trigger condition column to change the trigger condition, such as

As shown below, for net, the trigger conditions are: any position, low level, high level, rising edge,

Falling and double edges (rising or falling);



For bus or group, there are logical operators **Operators** to choose from, including a total of 7 conditions:

Equal to (= =), Not equal to (!=), Greater than or equal to (>=), Less than or equal to (<=), Greater than (>), Less than (<),

Any (don't care). At the same time, the trigger conditions of each bit in the bus and group are

The optional values are limited to 0 and 1. The default value is 0. After inputting, you can see that the trigger condition of the corresponding net has changed.

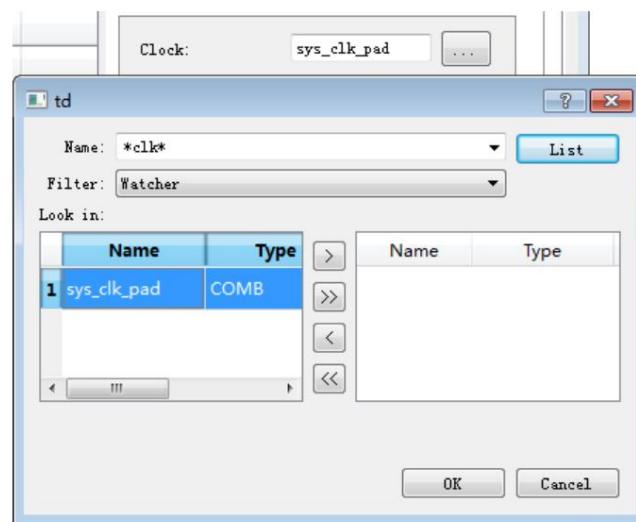
change.

gp	cnt[0].group	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="=="/> 0110101b
reg	cnt[0]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="!="/> L
reg	cnt[1]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value=">="/> H
reg	cnt[2]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="<="/> H
reg	cnt[4]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value=">"/> H
reg	cnt[5]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="<"/> L
reg	cnt[6]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="button" value="H"/> H
reg	cnt[7]	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

6. Click the button after the **Clock** column to add a sampling clock, which will be used as the entire ChipWatcher

The working clock of the module needs to ensure that a valid and appropriate clock signal is selected, otherwise it may cause failure to

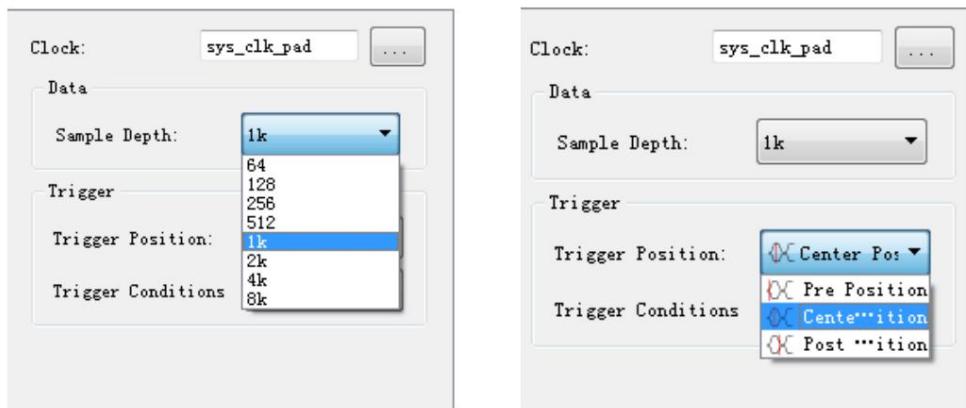
Triggered correctly or the sampling accuracy is deviated;



7. Select the sampling depth and set the trigger position. Pre Position means that the trigger position will be at the entire number of samples

The first third of the data; Center Position indicates that the trigger position will be in half of the entire sampled data

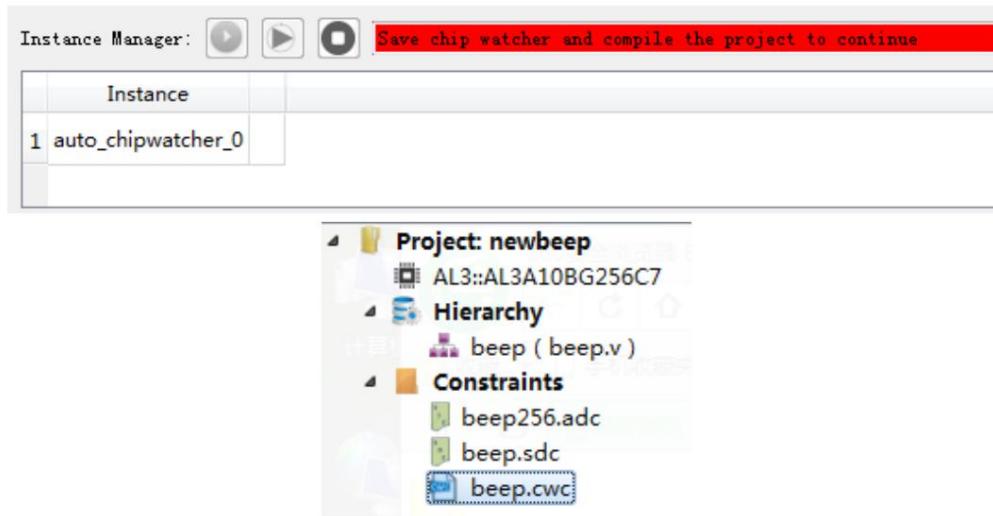
One place; Post Position means that the trigger position will be in the last third of the entire sampled data;



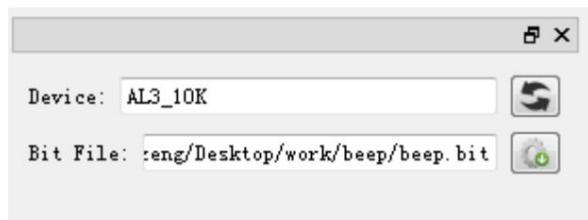
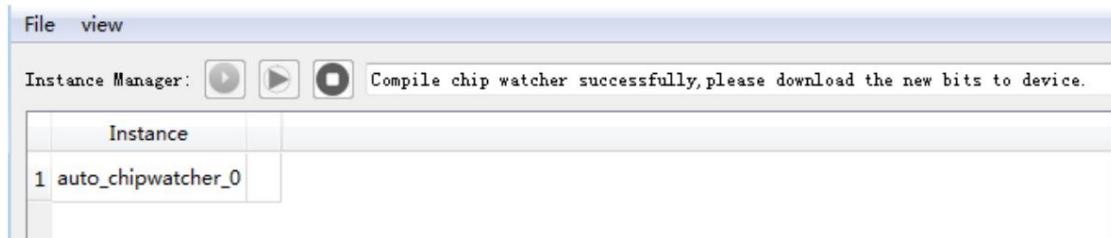
8. After setting all parameters, click **File** → **Save** in the upper left corner according to the prompts on the interface, or use

Shortcut operation **ctrl+s**, save the configuration file (.cwc) of ChipWatcher, add the configuration file to the project,

and recompile the project;



9. After compiling, download the newly generated bit file according to the prompts;



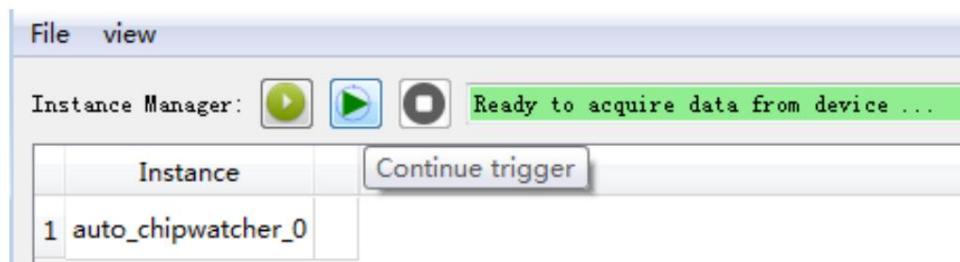
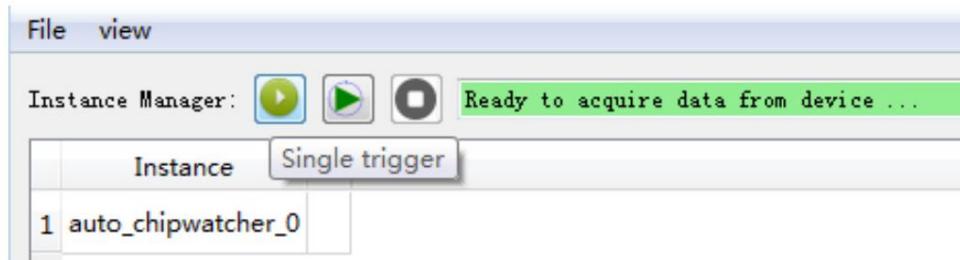
10. After downloading, the trigger button in the upper left corner of ChipWatcher will light up, and the following prompt will be given. At this time, click

Trigger button, ChipWatcher will start monitoring the specified signal, once the preset trigger conditions are met, the

will return the data in the chip. Among them, Single trigger is a single trigger, that is, to obtain the current moment in the chip

The data under the current condition; Continue trigger is a continuous trigger, which can obtain real-time data under this condition in the chip

data;

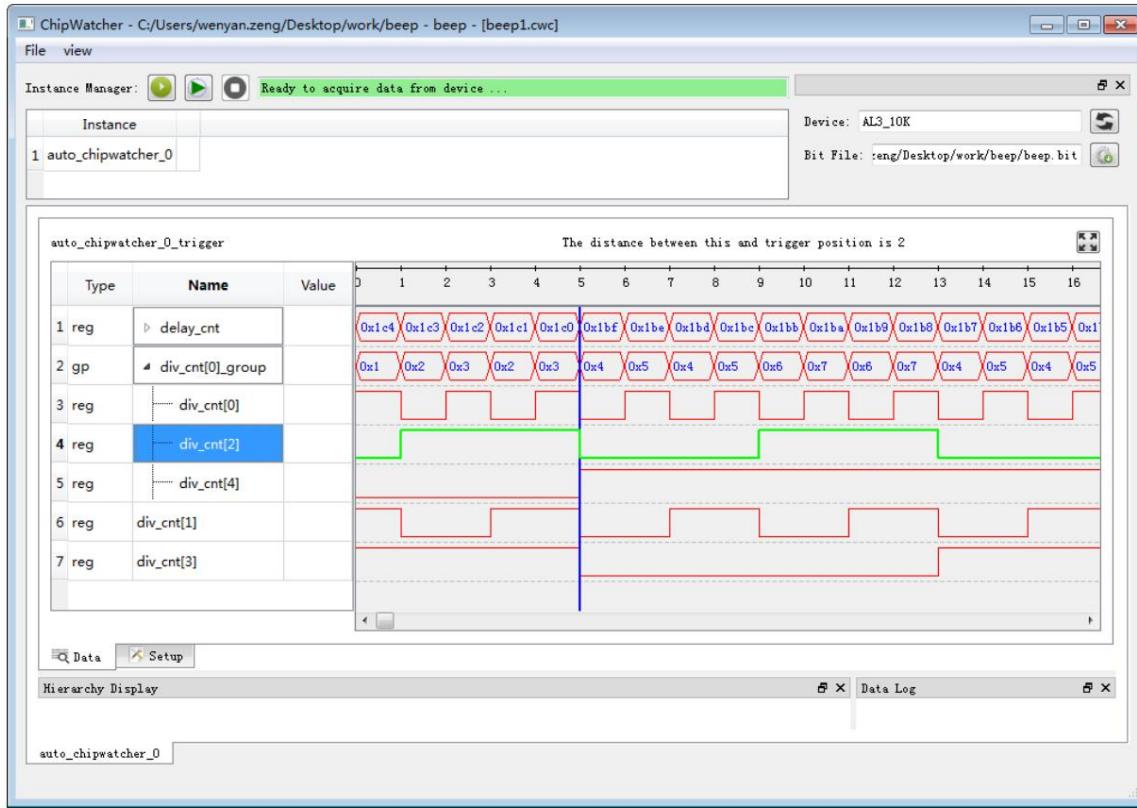


If the downloaded bit file does not match the current ChipWatcher object, the following prompt will be given.



11. Once the signal is triggered, the ChipWatcher page will switch from the Setup interface to the Data page and display the read

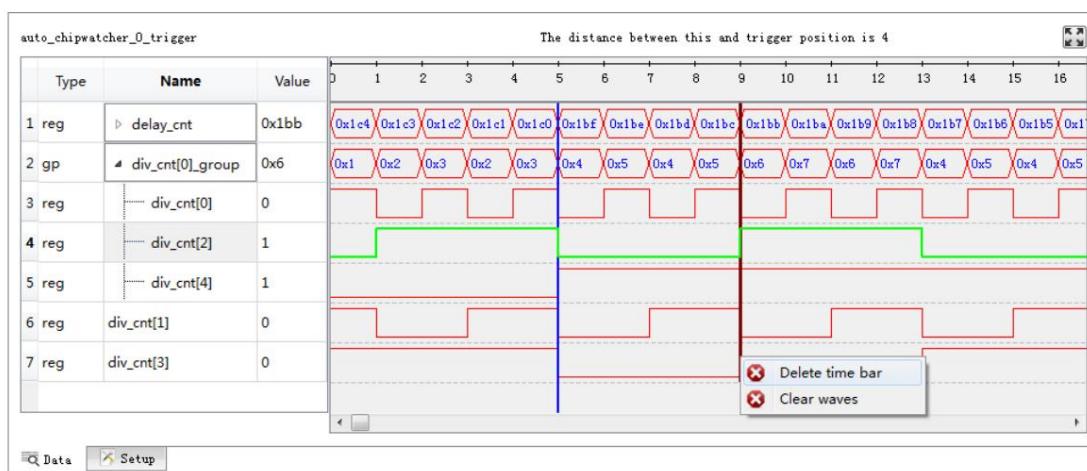
waveform of the return signal. The blue vertical line represents the trigger position and cannot be deleted;



The user can double-click a position on the waveform to add a **time bar**, as shown in the dark red line in the figure below, the **time bar** can be displayed by the mouse to move the cursor, or right-click to delete. When moving the **time bar**, the **Value** column on the left will

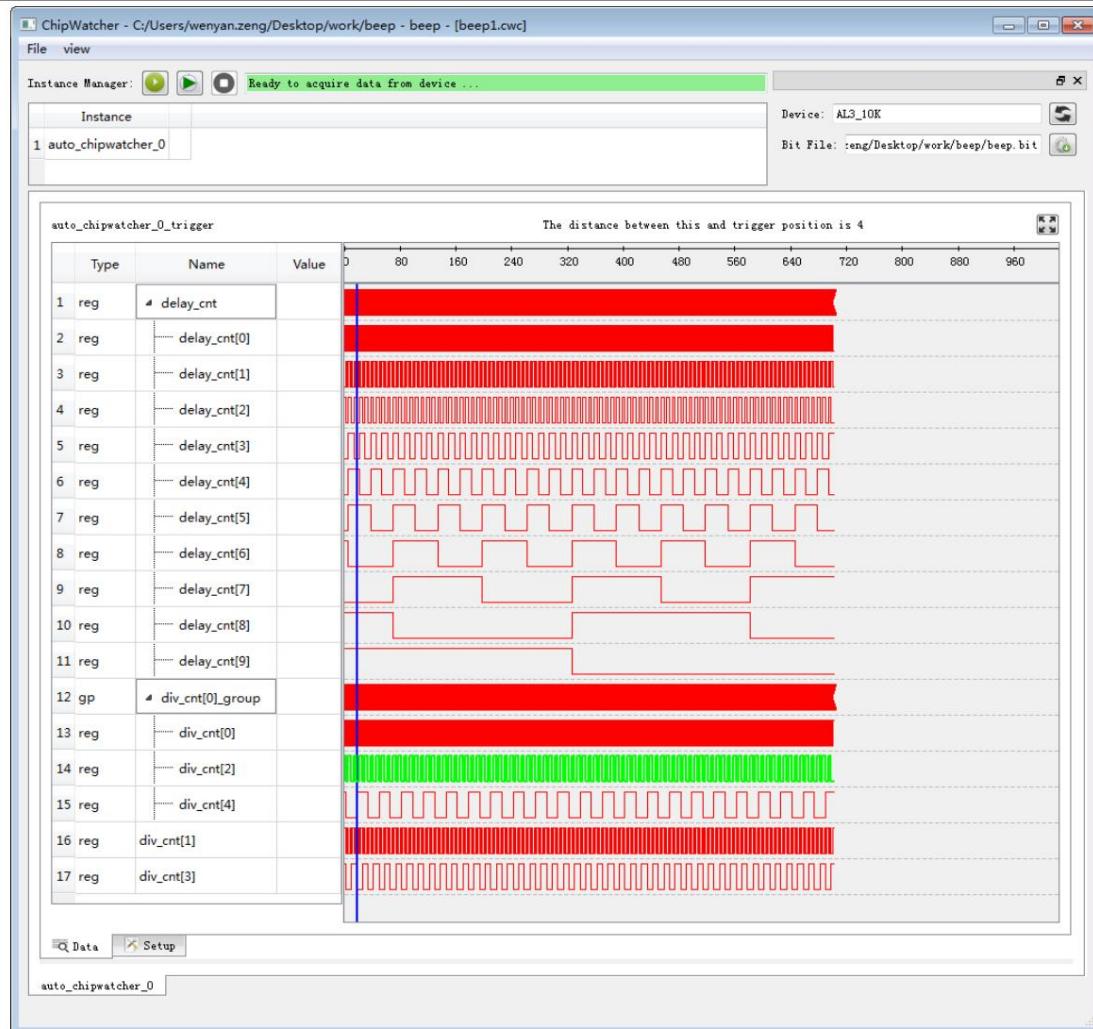
Displays the value of each signal at that location and gives the distance from that location to the trigger location. You can also right-click the waveform and select

Select “Clear waves” to clear the current waveform and re-trigger.



12. When the mouse is in the waveform area, the mouse wheel up means zooming in on the waveform, and the scroll wheel down means zooming out.

, you can also click "Fit for view" in the upper right corner to display the entire waveform status;

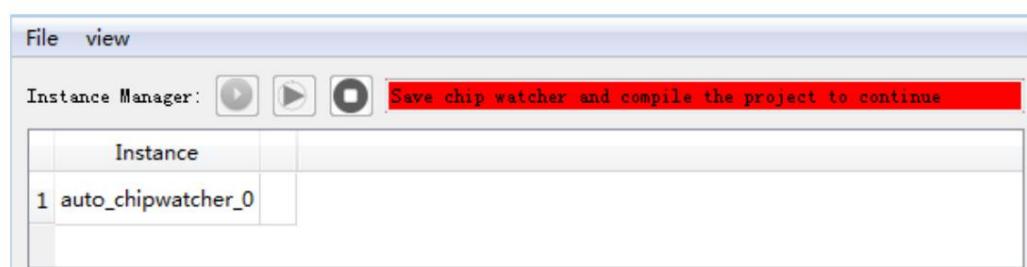


13. If the user only changes the trigger condition (such as: changing the rising edge to the falling edge, changing the low level to the high level, etc.),

Retriggering can be done directly without recompiling. And when the user changes the number of signals, sampling depth, sampling

If the sample location and other conditions are met, you need to re-save, compile and download before triggering again. At this time,

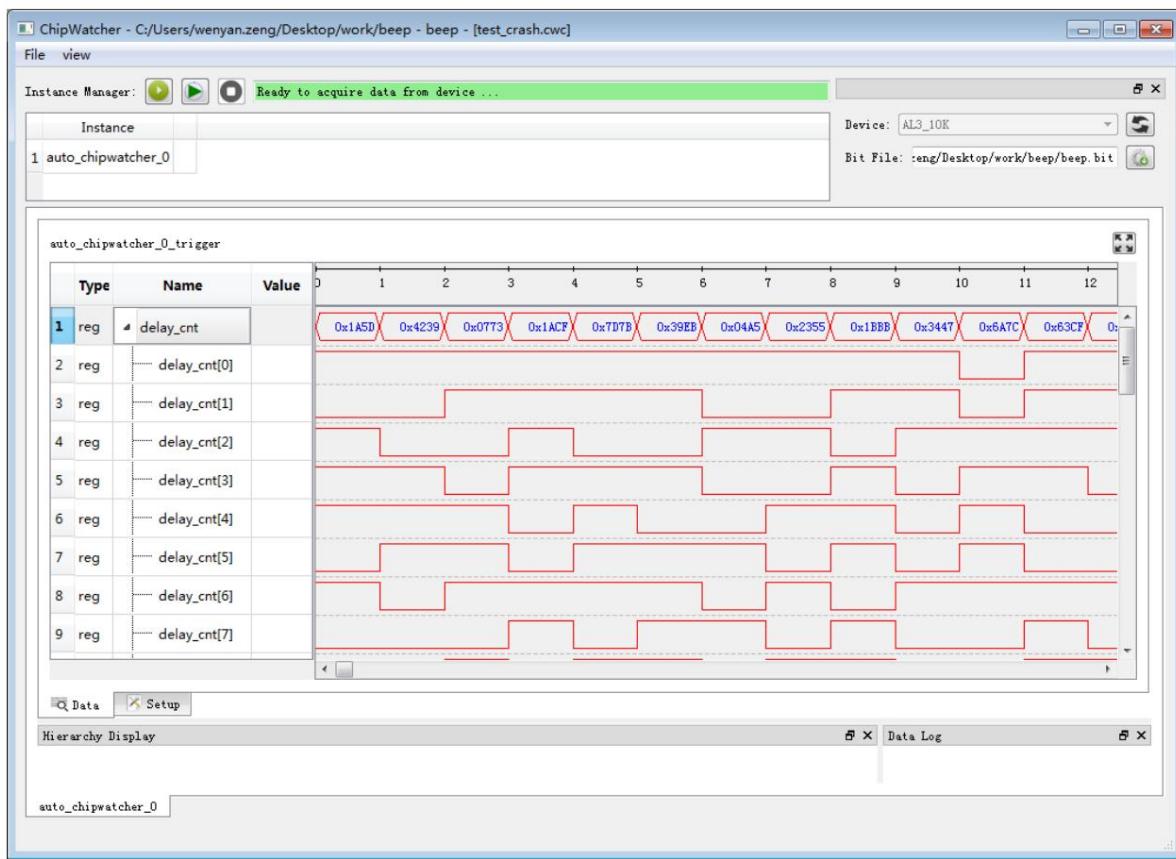
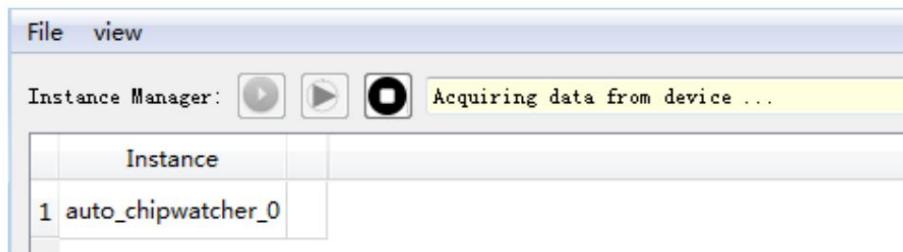
Chipwatcher will also give corresponding prompts:



14. If the state is always as follows after clicking the trigger, it means that the data in the chip cannot meet the trigger condition, please change it

Trigger condition, re-trigger; at this time, click the stop button, the data interface will export the current signal of interest

waveform.



To use **ChipWatcher** without engineering :

If there is no project, you can still use ChipWatcher to open the existing cwc file for waveform

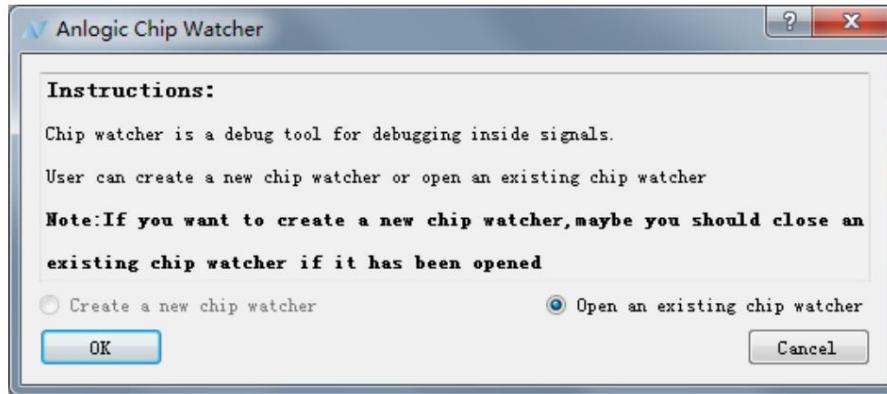
Check. The following issues need attention:

1. It is necessary to ensure that the files .cwc, .bid and .bit required by ChipWatcher are in the same folder;
2. Signals cannot be added, deleted or changed from the ChipWatcher interface of the project;

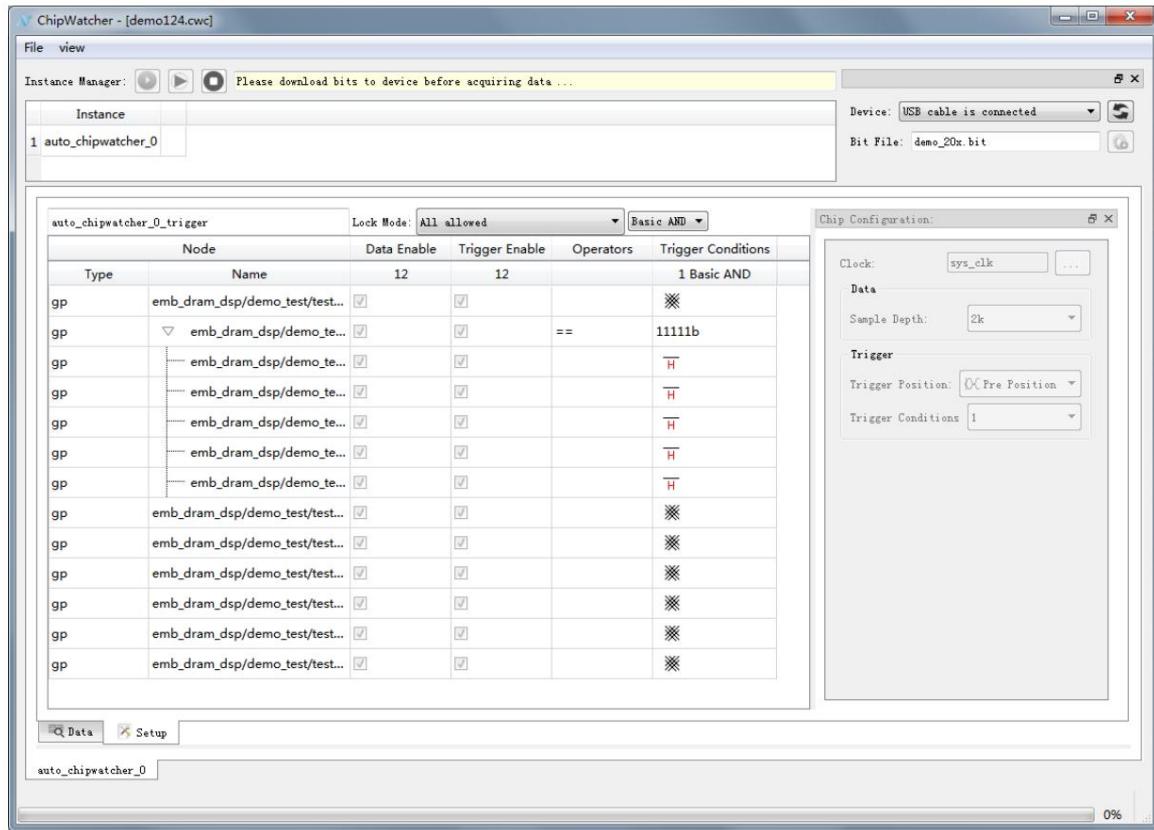
3. The sampling depth and trigger position cannot be changed from the ChipWatcher interface of the project.

The specific usage is as follows:

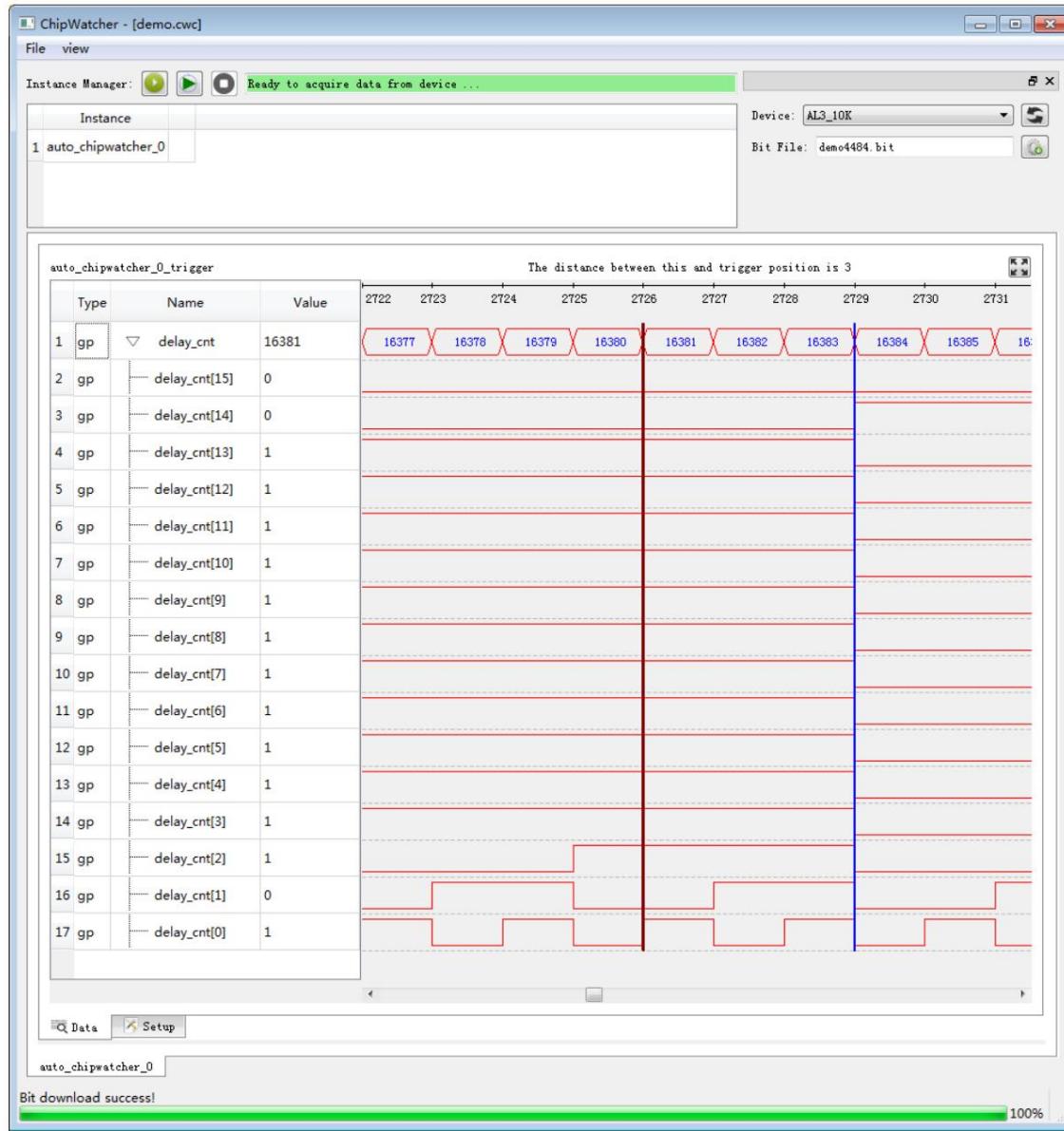
1. Open the ChipWatcher interface, only select "Open an existing chip watcher", and click "OK";



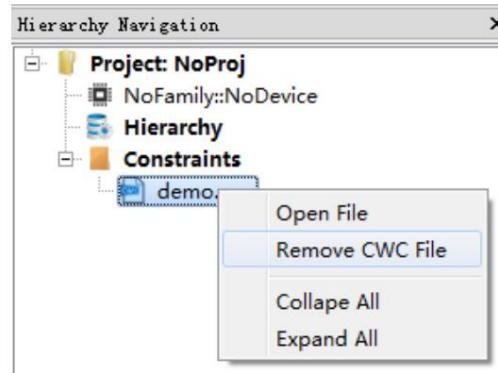
2. Open an existing .cwc file;



3. The trigger conditions can be modified to download and trigger;



4. When you need to open another .cwc file, you need to close and remove the current file first



8.4 BramEditor

Users can use **BramEditor** to read data from RAM in the chip, and can modify these data,

After the modification is written into the chip, the effect of the modification can be seen.

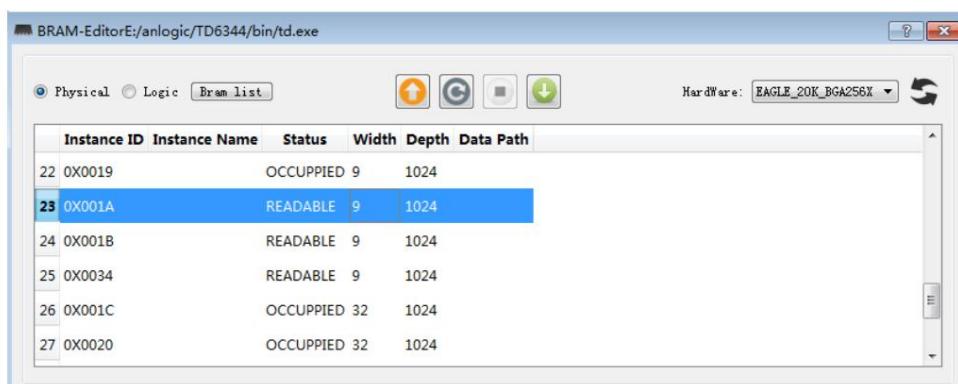
1. Expand **Tools** → **Debug Tools** and select **BramEditor**;

2. If the **Hardware** column displays "No Hardware", please check whether the hardware ports are connected correctly, and

Whether the chip is powered on, and finally click the refresh button next to it to refresh. In the pop-up **BramEditor** dialog box, select

Select an Instance, and then read and write the information of the Bram. Only **Status** is **READABLE** _

Instance can perform read and write operations;



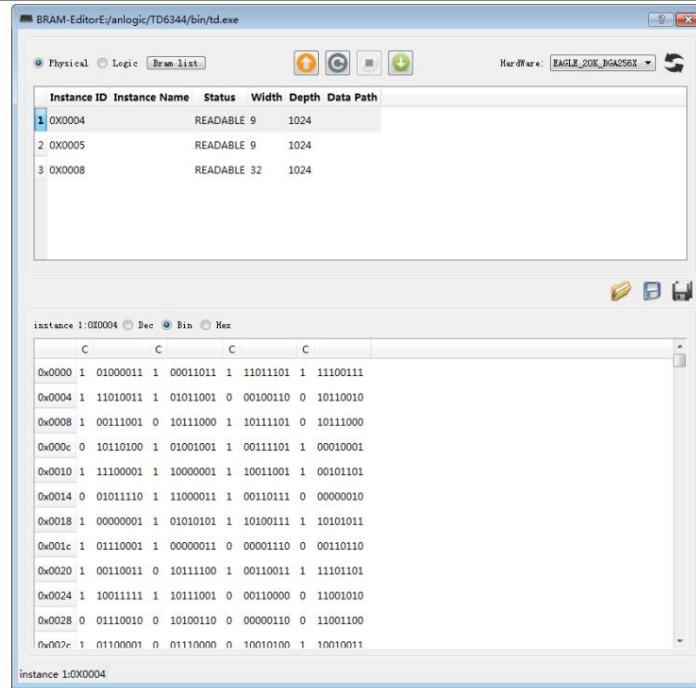
3. Click the button to **read** data from the chip, the user can choose decimal (Dec), binary (Bin) or hexadecimal

Control (Hex) to display the read back data, the default is binary. For Physical BRAM9K, depth is 1024, width

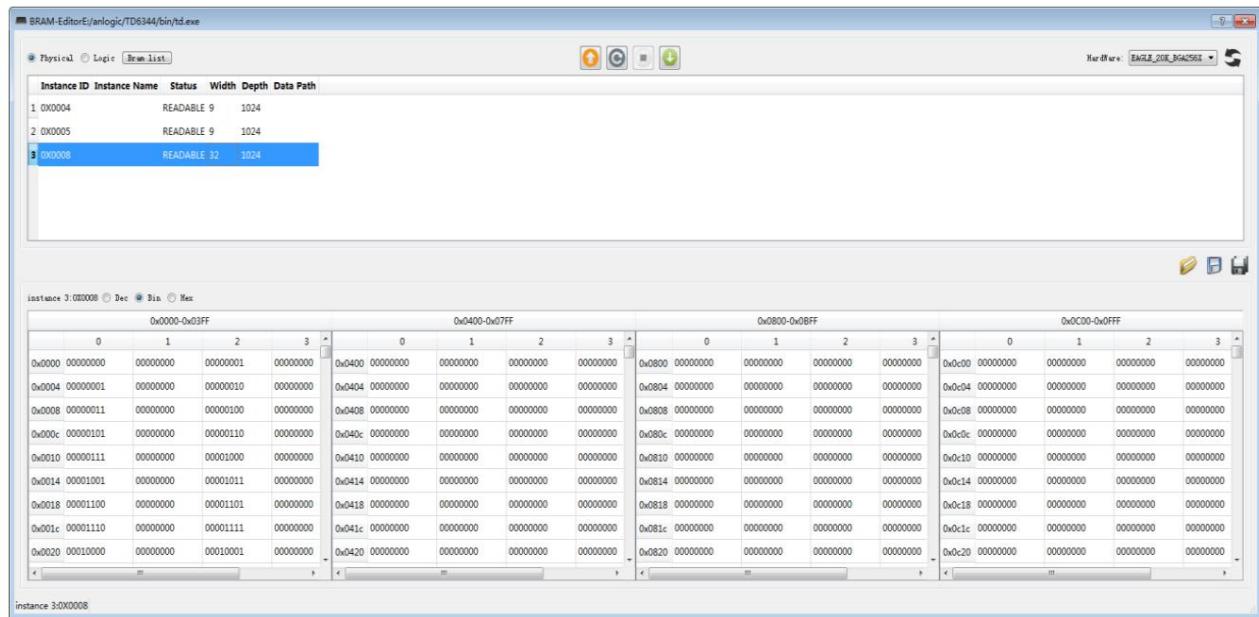
The degree is 9 bits, and the highest bit is the parity bit (the ninth bit); for Physical BRAM32K, the depth is 1024, and the width is

32 bits, no check digit. For Logic BRAM, the depth and width are the same as those designed by the user;

The data of BRAM9K is displayed as follows:



The data of BRAM32K is displayed as follows:

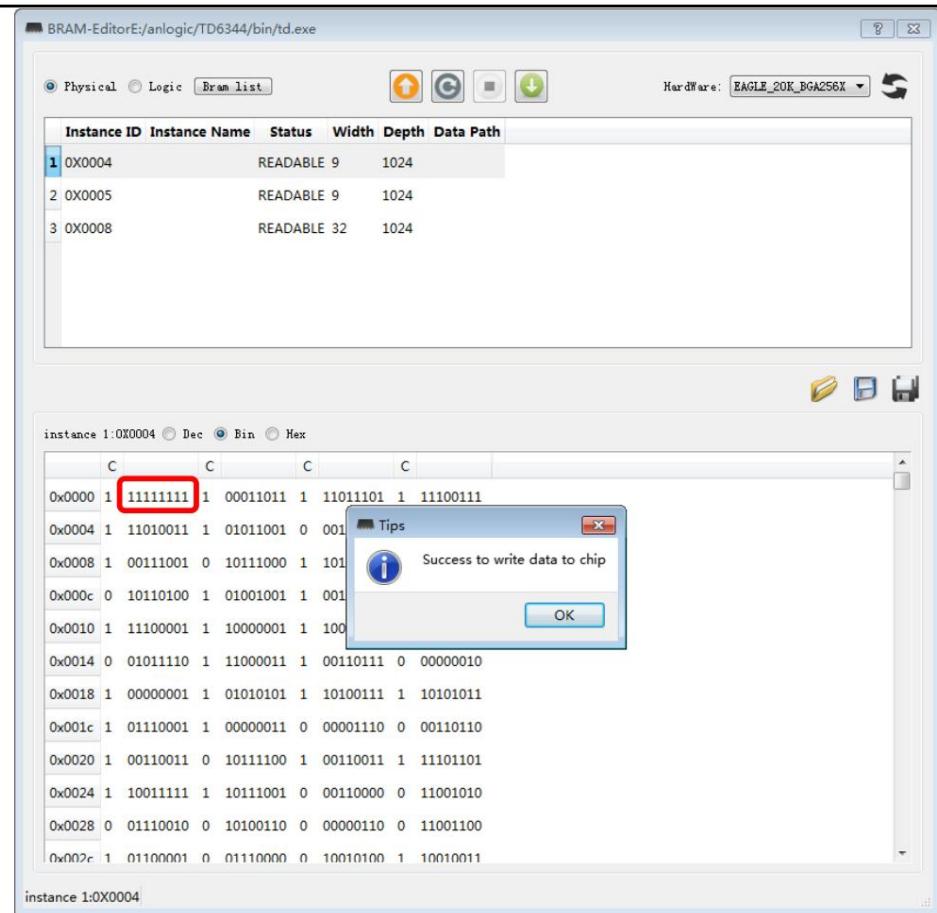


6. Double-click a data to modify it. After modification, click the button to write the data back to the chip. button available



Read the data in a loop, the button can stop the loop;



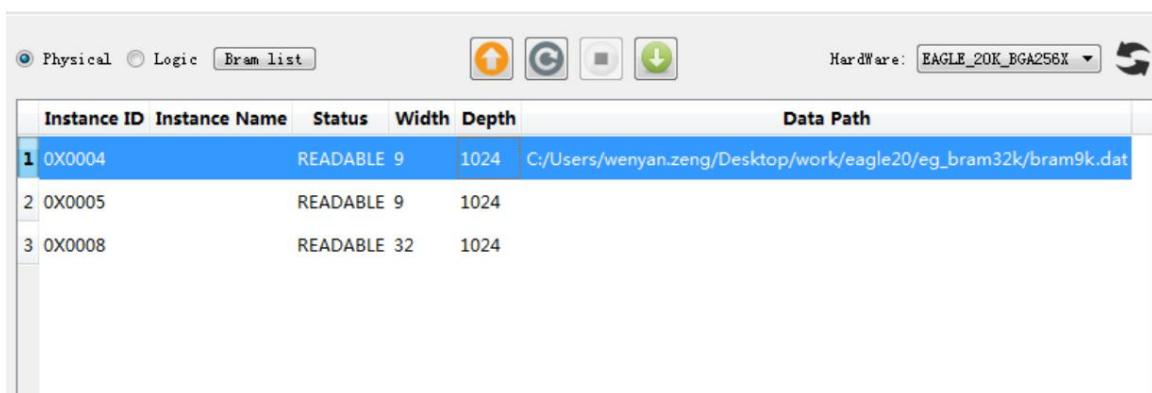


Data in RAM can also be written in batches.

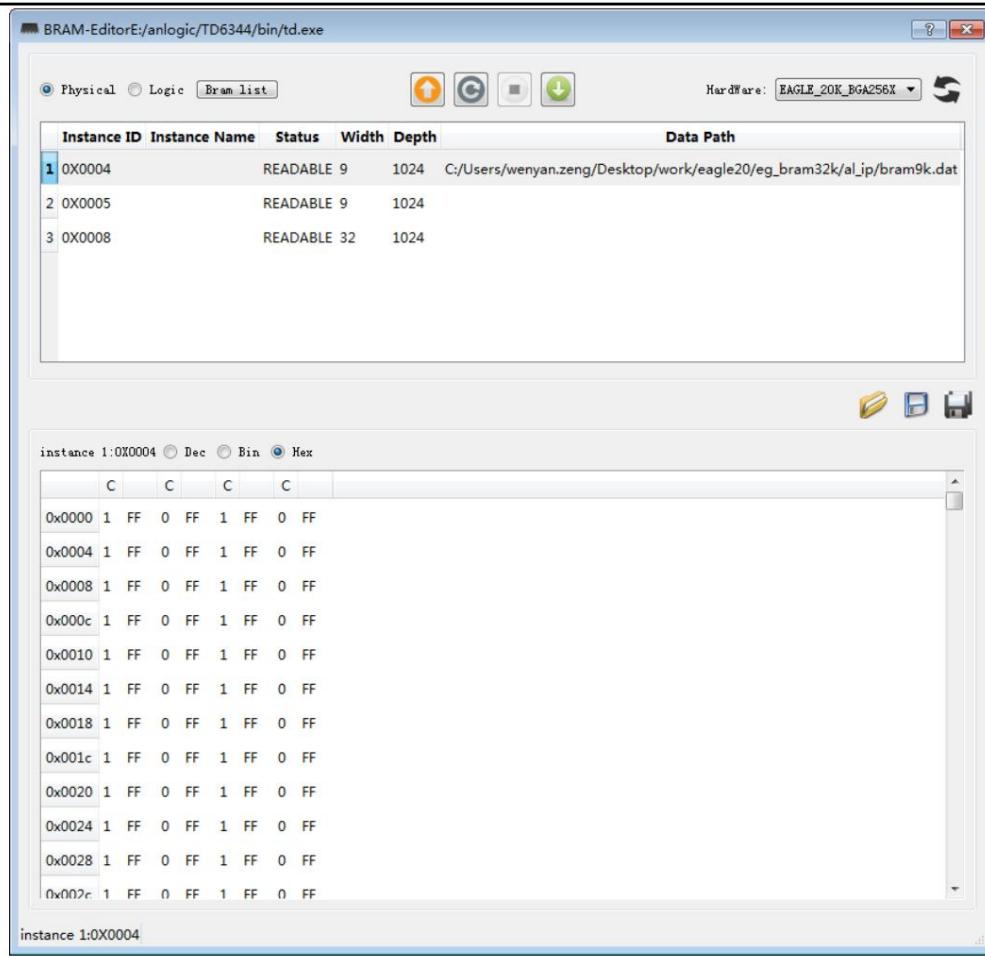
Select an Instance, click to open a .dat file to be written, click to write data to the chip,

It will prompt that the data was written successfully. If the data written does not match the size of the RAM, a warning will be given. last point

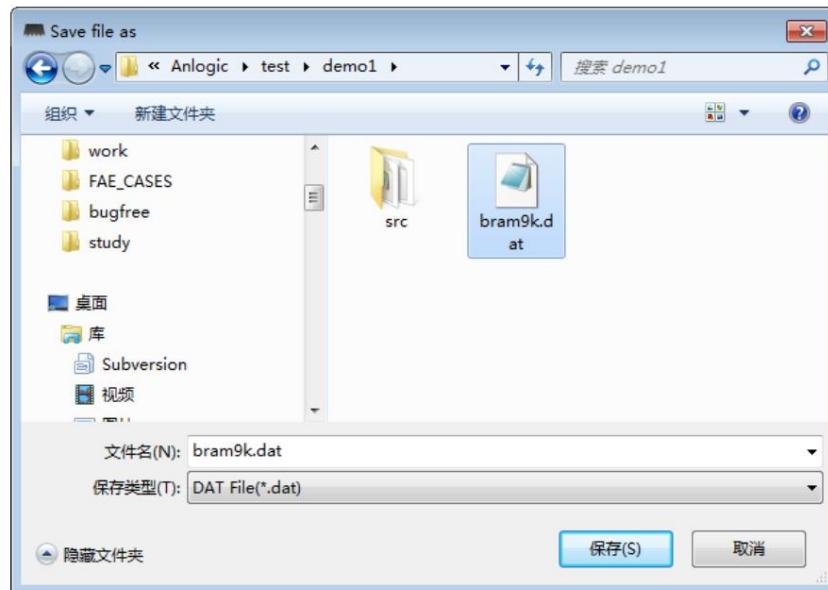
Click to view the written data.



After writing to RAM, the data read back is as follows



7. The read data can be saved as a dat file by clicking the button.



8. During the running process of Flow, a .bid file containing the BRAM Instance Name will be generated.

In BramEditor, you can click the **Bram list** button in the upper left corner of the interface to match the internal Instance Name with the

Instance ID corresponds, which is convenient for users to debug.

Instance ID	Instance Name	Status	Width	Depth	Data Path
1	inst2_inst_512x16_sub_000000_000	READABLE	9	1024	C:/Users/wenyan.zeng/Desktop/work/eagle20/eg_bram:
2	inst2_inst_512x16_sub_000000_009	READABLE	9	1024	
3	inst1_inst_2048x16_sub_000000_000	READABLE	32	1024	

9. Click the **Logic** button to switch to the interface displayed by the Logic BRAM. Other operations are the same as those of Physical

BRAM consistent.

Logic Name	Status	Width	Depth	Data Path
1 inst2_inst	READABLE	16	512	
2 inst1_inst	READABLE	16	2048	

instance 1:inst2_inst		Dec	Bin	Hex
0x0000~0x0080	0x0081~0x0101	0x0102~0x0182	0x0183~0x01ff	
15~8 7~0	15~8 7~0	15~8 7~0	15~8 7~0	
0x0000 01000011	01000011	0x0081 00100101	00100101	
0x0001 00011011	00011011	0x0082 00100101	00100101	
0x0002 11011101	11011101	0x0083 00111101	00111101	
0x0003 11100111	11100111	0x0084 01111010	01111010	
0x0004 11010011	11010011	0x0085 00011111	00011111	
0x0005 01011001	01011001	0x0086 11110011	11110011	
0x0006 00100010	00100010	0x0087 10000000	10000000	
0x0007 10110010	10110010	0x0088 01000100	01000100	
0x0008 00111001	00111001	0x0089 01011000	01011000	
0x0009 10111000	10111000	0x008a 01010101	01010101	
		0x0102 10010101	10010101	
		0x0103 01101101	01101101	
		0x0104 11101100	11101100	
		0x0105 01000001	01000001	
		0x0106 10010100	10010100	
		0x0107 11010100	11010100	
		0x0108 11001010	11001010	
		0x0109 10111100	10111100	
		0x010a 10010111	10010111	
		0x010b 11001101	11001101	
		0x010c 10010010	10010010	

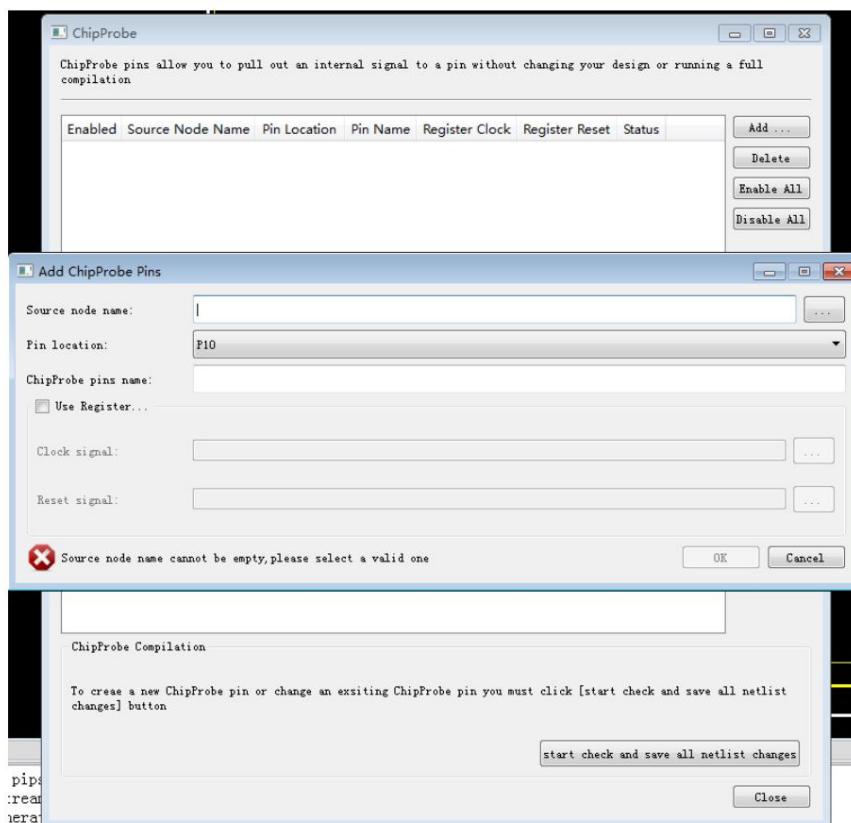
8.5 ChipProbe

Using ChipProbe, users can lead some internal signals to the IO terminal without changing the design

This allows users to check the changes of internal signals in real time with external devices.

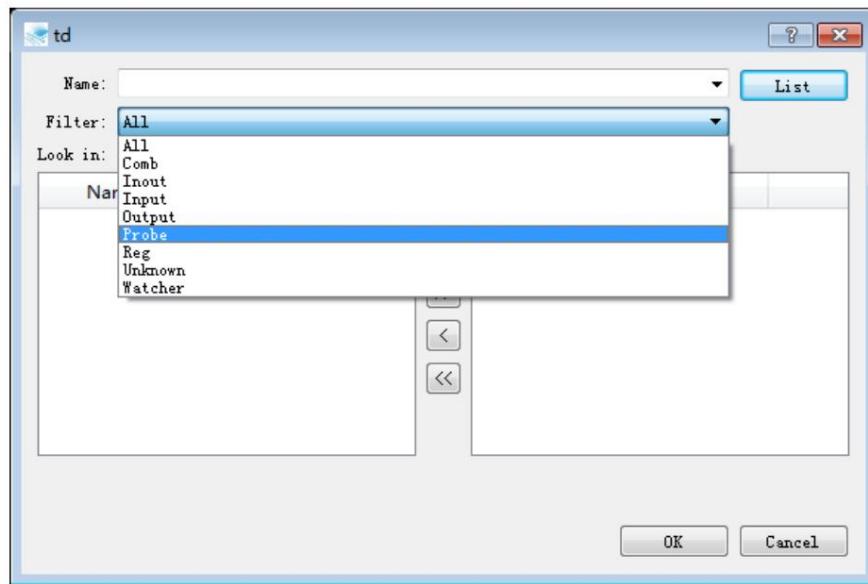
1. After running the HDL2Bit process, expand **Tools** → **Debug Tools** and select **ChipProbe**

2. In the **ChipProbe** dialog box that pops up, click **Add** to add the internal variables you want to view.

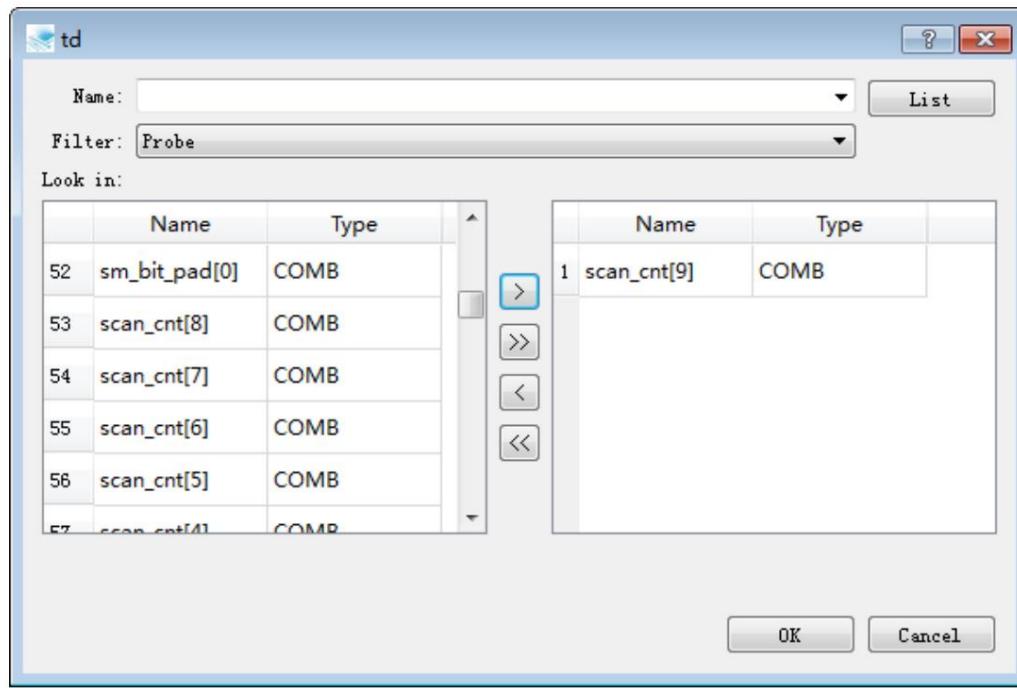


3. The user can manually enter the **Source node name**, or click to add it, in the new pop-up dialog box, select the filter type as **Probe**. When using ChipProbe, only

When the filter type of the number is Probe, it can be extracted for debugging.



4. Click to select an internal variable, click to add, and click OK after adding.



5. After adding, select an output pin for the internal signal. If you need to use register to latch the internal

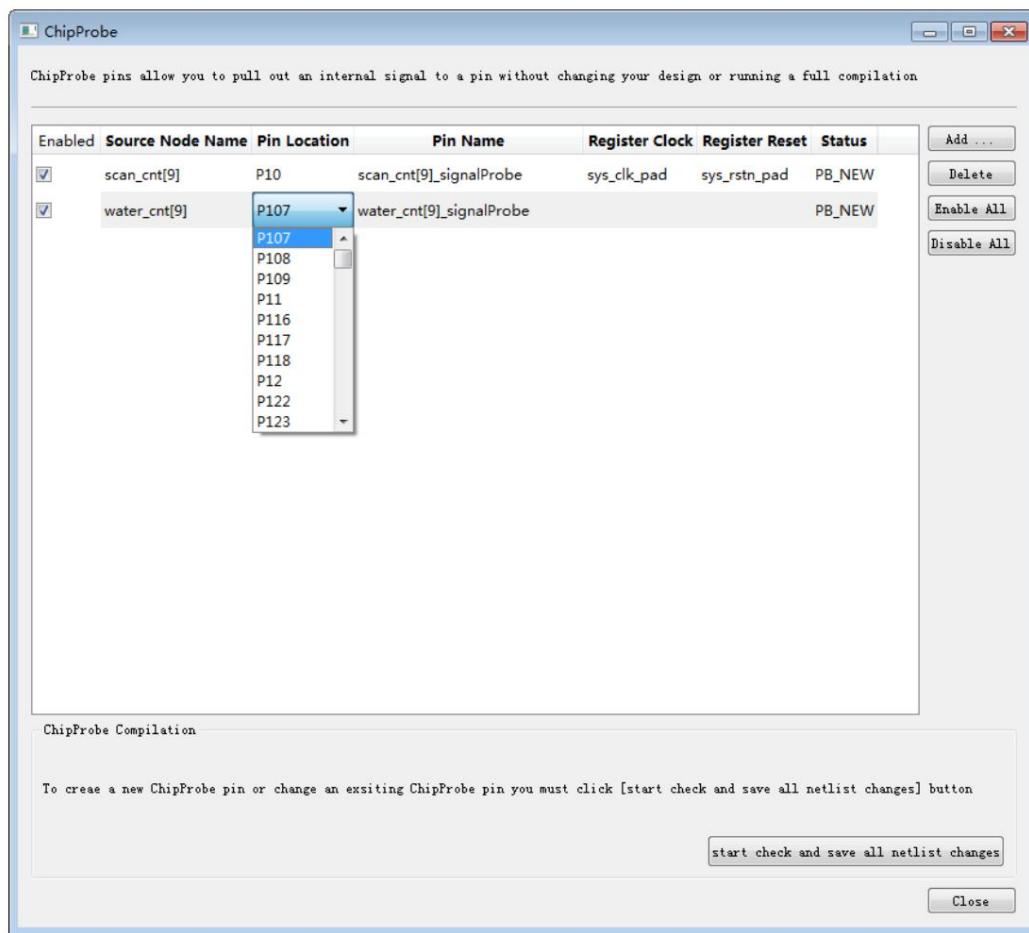
If you need to select a **Clock** signal, you can also add a **Reset signal** to it, as shown in the figure below

shown. If you do not need to use register, just click **OK**.



6. You can continue to add through Add, or delete through Delete . Users can also right-click on the Pin

Location changes the output pin.



7. You can check the small box under **Enabled** to activate a certain internal signal, or you can activate all the signals through **Enable all** .

There is a signal, after activation, select **start check and save all netlist changes** in the lower right corner, and reprogram

After translating and generating a bit stream file, after downloading it to the chip, connect the external device to the pre-selected output pin for checking.

Test debugging.

8.6 Power Estimator

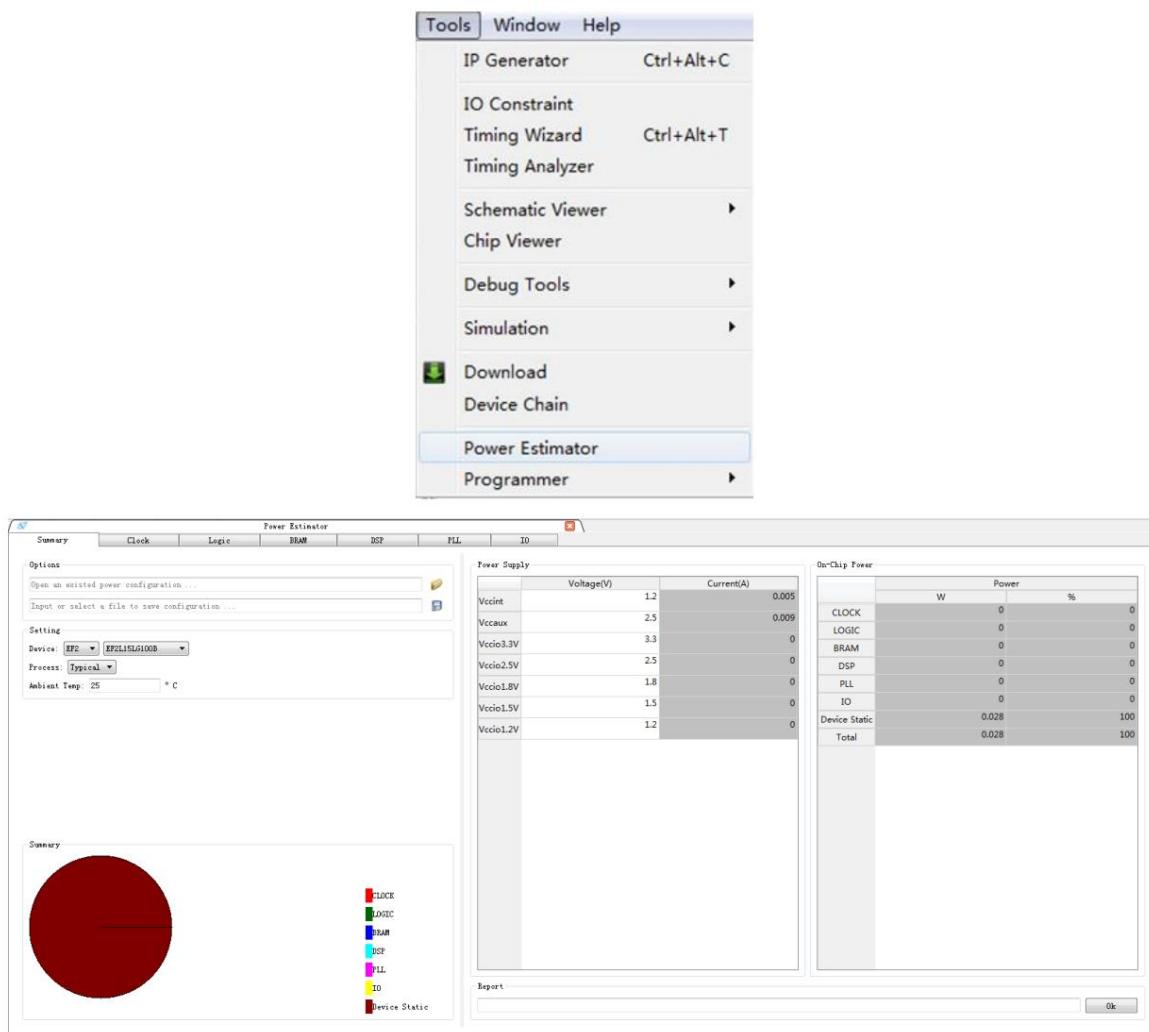
Power Estimator is a simple visualization tool for evaluating and counting the overall power consumption data of the entire chip

Tool. Support out of the project, no design data is required. The user only needs to input various resource configuration and total amount, clock

The frequency and the average toggle rate of the signal can be evaluated. All power dissipation (W) and current (A) are accurate to decimal point

Last 3 digits, all percentages (%) are accurate to 1 decimal place.

It can be opened through Tools → Power Estimator. After Power Estimator is opened, the interface is displayed as follows:



Power Estimator according to Summary, Clock, Logic, BRAM, DSP, PLL, IO these major modules

The block pagination displays power consumption information. When Power Estimator opens, it defaults to the Summary page.

1. Summary

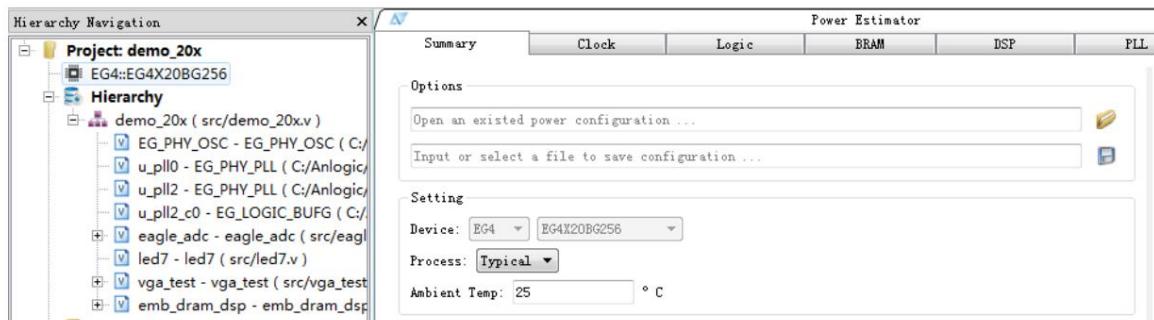
Currently Power Estimator supports EF2, EF3, EG4 and other series of chips. If the TD has opened a

project, then after Power Estimator is turned on, the device is the same as the project by default and cannot be changed; if the TD is not

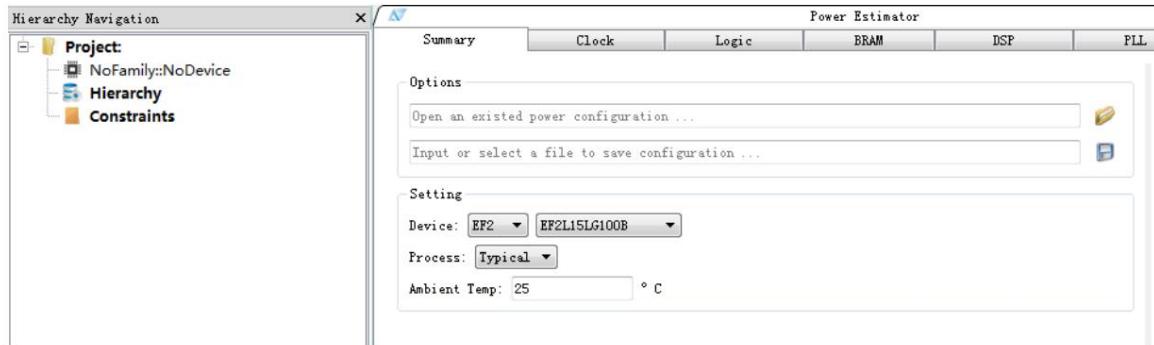
If any project is opened, then after opening the Power Estimator, the device defaults to EF2L15LG100B, which supports

Users can switch devices by themselves.

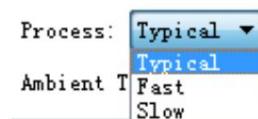
TD open with a project:



TD with no project:



Specify the evaluation target process angle, currently only supports Typical/Slow/Fast



Specify the ambient temperature of the chip, the recommended range is -40°~125°, and the default is 25°. Transforming the temperature can cause leakage of the associated modules

Electrical functions and static power consumption vary.

Ambient Temp: 25 ° C

The Voltage Domains table lists the configuration voltages and supply currents for all voltage sources. The default value of the configuration voltage is the reference voltage

voltage, the user can click to fine-tune (it can be up or down by 5%, and there are two kinds of Vccaux reference voltages). Current value

The size varies with factors such as resource usage, clock frequency, and average toggle rate. Adjust any voltage value, by this

The power consumption of a module driven by a voltage source varies.

Power Supply

	Voltage(V)	Current(A)
Vccint	1.2	0.005
Vccaux	2.5	0.009
Vccio3.3V	3.3	0
Vccio2.5V	2.5	0
Vccio1.8V	1.8	0
Vccio1.5V	1.5	0
Vccio1.2V	1.2	0
	1.2	
	1.14	
	1.26	

On-chip power consumption data is divided into leakage power consumption, static power consumption, and dynamic power consumption. Device in the power consumption decomposition table

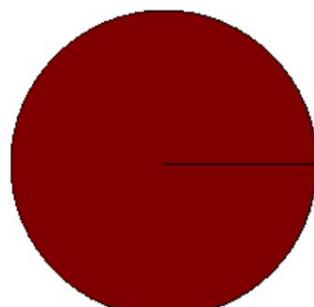
Static includes all leakage power consumption and static power consumption of the entire chip, and the dynamic power consumption sub-circuit module is decomposed and displayed. same

, the lower left corner of the Summary page displays the power consumption ratio of each module in the form of a pie chart.

On-Chip Power

	Power	
	W	%
CLOCK	0	0
LOGIC	0	0
BRAM	0	0
DSP	0	0
PLL	0	0
IO	0	0
Device Static	0.028	100
Total	0.028	100

Summary



- █ CLOCK
- █ LOGIC
- █ BRAM
- █ DSP
- █ PLL
- █ IO
- █ Device Static

Save and import the configuration information entered by the user in the format of .xml file



Export evaluation report: detailed power consumption and current data, proportion and sub-module printing user-set data. available

to compare the results of power consumption evaluations on different devices.



```
Device           ef2_1
Package        EF2L15LG100B
=====
```

Power Estimator Report

Contents

- 1. Summary
 - 1.1 Power Breakdown
 - 1.2 Power Supply
- 2. Detailed Report
 - 2.0 Regulators
 - 2.0.1 Regulator Power Summary
 - 2.1 Clock
 - 2.1.1 Clock Power Summary
 - 2.1.2 Clock Settings
 - 2.2 Logic
 - 2.2.1 Logic Power Summary
 - 2.2.2 Logic Settings
 - 2.3 Block RAM
 - 2.3.1 Block RAM Power Summary
 - 2.3.2 Block RAM Settings
 - 2.4 DSP
 - 2.4.1 DSP Power Summary
 - 2.4.2 DSP Settings
 - 2.5 PLL
 - 2.5.1 PLL Power Summary
 - 2.5.2 PLL Settings
 - 2.6 IO
 - 2.6.1 IO Power Summary
 - 2.6.2 IO Settings

1. Summary

Process Corner	Typical
Environment Temp (C)	25
Total on-chip power (W)	0.048
Dynamic (W)	0.02
Device static (W)	0.028

1.1 Power Breakdown

Except for the Summary page, the upper left corner of other pages is the power assembly table, and the upper right corner is the resource usage rate

sheet. Among them, the power consumption percentage in the power assembly table represents the total power consumption of the current module in the total power consumption of the entire chip

In principle, the resource usage rate does not exceed 100%, if it exceeds 100%, it will be highlighted in red. Tool only negative

It is responsible for evaluating the power consumption based on the information input by the user, and does not evaluate the resource usage, clock frequency and signal toggle rate.

Legality checks or constraints.

Power Supply		
	Voltage(V)	Current(A)
Vccint	1.2	0.005
Vccaux	2.5	0.009
Vccio3.3V	3.3	0
Vccio2.5V	2.5	0
Vccio1.8V	1.8	0
Vccio1.5V	1.5	0
Vccio1.2V	1.2	0

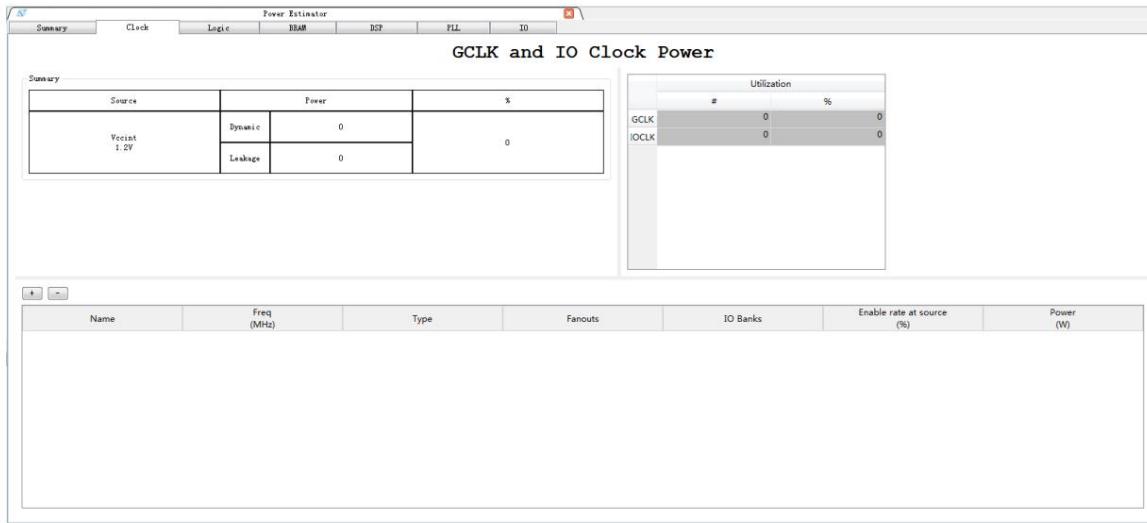
Voltage values on the Summary page

Summary	
Source	Supply
Vccint 1.2V	Vccint
Vccaux 2.5V	Vccio1.2V Vccio1.5V Vccio1.8V Vccio2.5V Vccio3.3V

Voltage values on non-summary pages

2. Clock

The power consumption interface of the clock resource module is as follows:



The power consumption data of the clock resource module, including the voltage source name and voltage value, dynamic power consumption and leakage power consumption, total

The power consumption is the percentage of the overall power consumption.

Summary			
Source	Power		%
Vccint 1.2V	Dynamic	0	0
	Leakage	0	

Global clock resource usage and IO clock resource usage (each IO bank holds two clocks). Complete

The dynamic power consumption of the office clock is the power consumption of the entire clock tree, and the dynamic power consumption of the IO clock is the calculation of the entire IO in a bank.

clock Clock tree rollover power.

Utilization		
	#	%
GCLK	0	0
IOCLK	0	0

Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Not repeatable; input the expected clock frequency at Freq; Type is the clock resource type, supporting Global and IO Clock;

Fanout is the total fanout of the global clock, which can be filled only when the clock type is Global; IO Banks is IO

The number of IO banks that the clock spans, which can only be filled in when the clock type is IO Clock; Enable rate at

source is the clock source enable, the default value is 100%, which can be modified within the range of 0~100%.

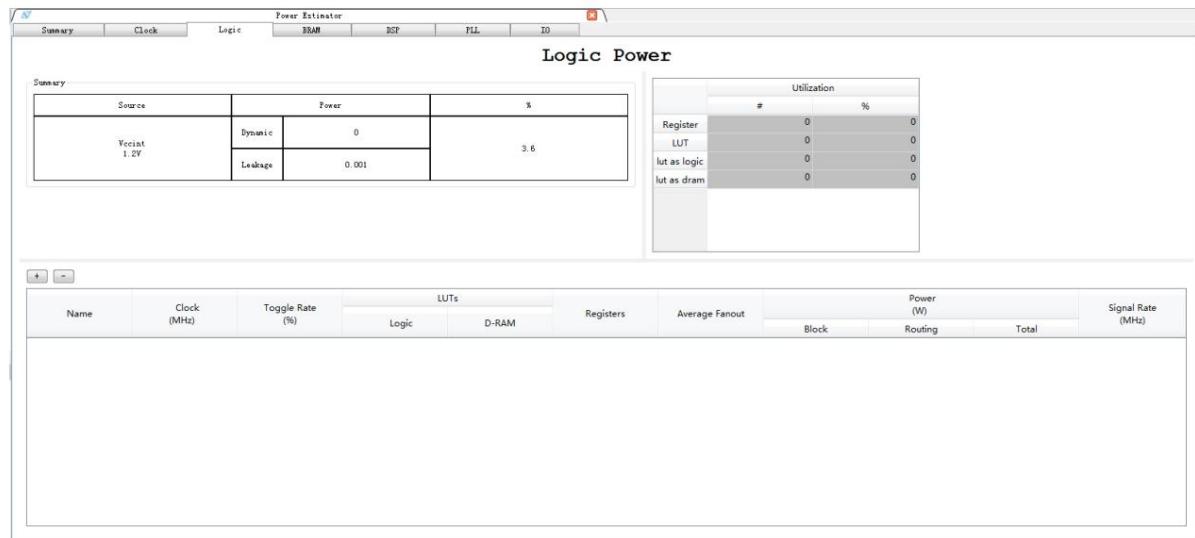
After filling in the required data, you can see the calculation results for this configuration. If the process of summary page

Angle, temperature, and voltage values change, and here will change as well.

	Name	Freq (MHz)	Type	Fanouts	IO Banks	Enable rate at source (%)	Power (W)
1	clock1	125	GLOBAL	1000		100	0.005
2	clock2	125	IOCLOCK		4	100	0
3	clock3	150	GLOBAL	50		100	0.006

3. Logic

The power consumption interface of the logic resource module is as follows:



The power consumption data of the logic resource module includes the voltage source name and voltage value, dynamic power consumption and leakage power consumption,

Overall power consumption as a percentage of overall power consumption.

Summary			
Source	Power		%
Vccint 1.2V	Dynamic	0	3.6
	Leakage	0.001	

The usage rate of logical resources, including the usage rate of LUT and register. LUT as normal logic and dram respectively

The quantity is the corresponding proportion of the total available resources.

	Utilization	
	#	%
Register	0	0
LUT	0	0
lut as logic	0	0
lut as dram	0	0

Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Not repeatable; Input the expected clock frequency at Clock, which must be filled in; Toggle Rate defaults to 12.5%,

You can also modify it yourself, but the range of the flip rate needs to be 0~100 or the flip rate is 200, where the flip rate is 200.

The clock signal, that is, the current logic is used as a clock resource; the estimated number of resources can be filled in the LUTs and Registers respectively,

LUTs are divided into Logic and D-RAM. When one of the three columns of data is not empty, the calculation can be triggered; Average

The default value of Fanout is 2.87, which can be modified to any decimal greater than or equal to 1.

After filling in the required data, you can see the calculation results for this configuration.

Calculated, divided into logic (including LUTs and Registers), wiring (including LUTs and Registers) and this configuration

total dynamic power consumption. If the process angle, temperature and voltage values of the summary page are changed, this will also change

change. Signal toggle density value: clock frequency x average toggle rate.

Name	Clock (MHz)	Toggle Rate (%)	LUTs			Registers	Average Fanout	Power (W)			Signal Rate (MHz)
			Logic	D-RAM				Block	Routing	Total	
logic1	135	12.5	1250	210	1255	2.87	0.012	0.026	0.038	16.9	
logic2	120	12.5	30	10	220	2.87	0.001	0.001	0.002	15	

4. BRAM

The power consumption interface of the BRAM module is as follows:

Source	Power		%
Vccint 1.2V	Dynamic	0	33.3
	Leakage	0.006	

#	%
BRAM9K	0 0
BRAM32K	0 0

Power consumption data of BRAM module, including voltage source name and voltage value, dynamic power consumption and leakage power consumption, assembly

Power consumption as a percentage of overall power consumption.

Summary			
Source	Power		%
Vccint 1.2V	Dynamic	0	2.6
	Leakage	0.001	

The utilization rate of BRAM resources lists the various types of BRAM resources supported by the device in turn.

	Utilization	
	#	%
BRAM9K	0	0
BRAM32K	0	0
BRAM128K	0	0
BRAM256K	0	0

Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Non-repeatable; select BRAM type at Type, EG4 series supports 9k and 32k, EF2 series except 9k and 32k

In addition to supporting 128k and 256k, the EF3 series only supports 9k; Num is the number of BRAM used in this type; Mode is

BRAM configuration mode; Toggle Rate default is 12.5%, users can modify it within the range of 0~100%;

BRAM port Port A/B, with independent clock, clock enable, write mode, write enable and bit width, specific by BRAM

The type and configuration mode determine the options.

BRAM Configuration Mode: The Mode drop-down menu options need to be adjusted with the BRAM type. As shown in the picture of BRAM9k

All modes: ROM, SPM (single port, bit width options are 1/2/4/8/9/16/18), SDPM_16 (simple dual port, bit width

The options are 1/2/4/8/16), SDPM_18 (simple dual ports, the bit width options are 9/18), DPM (dual ports, the bit width options are available in 1/2/4/8), DPM_9 (dual port, only 9 bit width options), FIFO (bit width options are 1/2/4/8/16), FIFO_9 (bit

Wide selection has 9/18).

Name	Type	Num	Mode
bram1	BRAM9K	2	ROM
bram2	BRAM9K	5	ROM SPM SDPM_16 SDPM_18 DPM DPM_9 FIFO FIFO_9

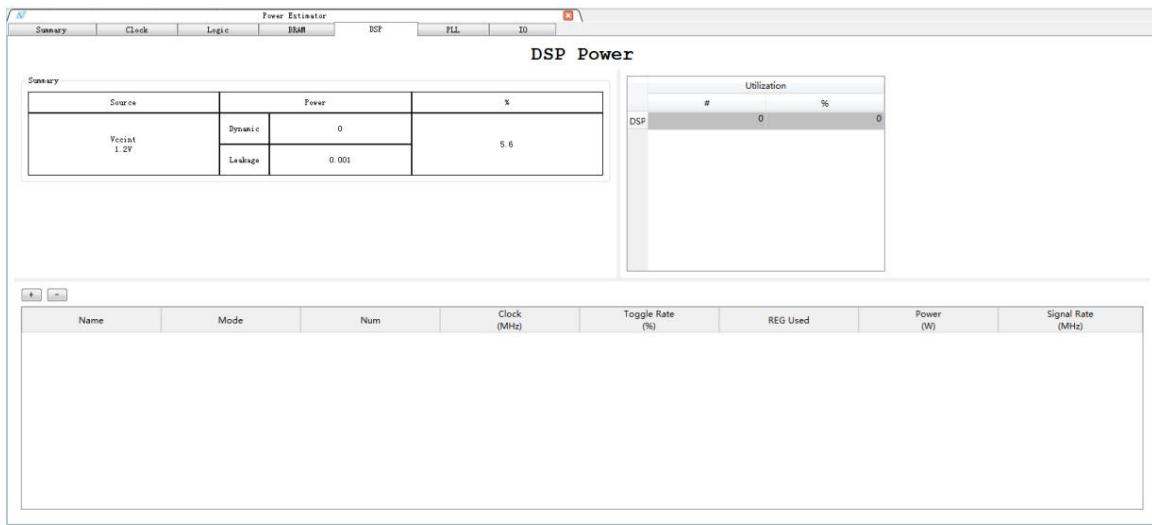
After filling in the required data, you can see the calculation results for this configuration. If the process of summary page

Angle, temperature, and voltage values change, and here will change as well.

Name	Type	Num	Mode	Toggle Rate (%)	Port A				Port B				Power (W)		
					Clock (MHz)	Enable Rate (%)	Write Mode	Write Rate (%)	Bit Width	Clock (MHz)	Enable Rate (%)	Write Mode	Write Rate (%)		
bram1	BRAM9K	12	DPM	12.5	125	100	Read First	100	2	125	100	Write First	100	4	0.02
bram2	BRAM32K	2	SPM	12.5	110	100	No Change	100	16						0.004

5. DSP

The power consumption interface of the DSP module is as follows:



Power consumption summary data of DSP resource module, including voltage source name and voltage value, dynamic power consumption and leakage power consumption, total power consumption

The power consumption is the percentage of the overall power consumption.

Summary			
Source	Power		%
Vccint 1.2V	Dynamic	0	2.4
	Leakage	0.001	

Statistics on the usage rate of DSP resources.



Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Non-repeatable; Mode is the DSP configuration mode, divided into 9x9 and 18x18; Num is the usage number of this type of DSP

Clock is the clock frequency; Toggle Rate defaults to 12.5%, the user can set it within the range of 0~100%

Modified; REG Used is an option for output register, and true/false correspond to whether or not to use output register.

After filling in the required data, you can see the calculation results for this configuration. If the process of summary page

Angle, temperature, and voltage values change, and here will change as well.

Name	Mode	Num	Clock (MHz)	Toggle Rate (%)	REG Used	Power (W)	Signal Rate (MHz)
1 dsp1	MULT9x9	5	125	12.5	false	0.002	15.6
2 dsp2	MULT18x18	2	100	12.5	false	0.001	12.5

6. PLL

The power consumption interface of the PLL module is as follows:

Source	Power	%
Vccint 1.2V	Dynamic: 0 Leakage: 0 Static: 0	0
Vccaux 2.5V	Dynamic: 0 Leakage: 0 Static: 0	0

Name	Input Clock (MHz)	Divide Counter	Multiply Counter	Clock0 Divide	Clock1 Divide	Clock2 Divide	Clock3 Divide	Clock4 Divide	Power (W)
Vccint									Vccint
Vccaux									Vccaux

Source	Power	%
Vccint 1.2V	Dynamic: 0 Leakage: 0 Static: 0	0
Vccaux 2.5V	Dynamic: 0 Leakage: 0 Static: 0	0

Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Not repeatable; Input Clock is the clock frequency; Divide Counter defaults to 1, the user can select the

Select within the range of 1~128; Multiply Counter defaults to 5, the user can select within the range of 1~128 through the drop-down menu

Select; Clock Divide is off by default, the user can select from 1~128 through the drop-down menu.

After filling in the required data, you can see the calculation results for this configuration. If the process of summary page

Angle, temperature, and voltage values change, and here will change as well.

Name	Input Clock (MHz)	Divide Counter	Multiply Counter	Clock0 Divide	Clock1 Divide	Clock2 Divide	Clock3 Divide	Clock4 Divide	Clock5 Divide	Clock6 Divide	Power (W)	Vccint	Vccaux
PLL1	125	3	6	2	off	off	off	off	off	off	0.002	0.003	
PLL2	100	5	5	off	4	off	6	off	off	off	0.001	0.001	

7. IO

The power consumption interface of the IO module is as follows:

I/O Power					
Source		Power			
Supply	Volt	Leakage	Static	Dynamic	Total
Vccint	1.2	0	0	0	0
Vccio1.2V	1.2	0	0	0	0
Vccio1.5V	1.5	0	0	0	0
Vccio1.8V	1.8	0	0	0	0
Vccio2.5V	2.5	0	0	0	0
Vccio3.3V	3.3	0	0	0	0

I/O Setting															
Name	Standard	DriverStrength	Direction	Num of Pin	IDelay	ODelay	Clock (MHz)	Toggle Rate (%)	Data Rate	Output Enable (%)	Output Load (pF)	Signal Rate (MHz)	Vccint	Vccaux	Vccio

In IO, the IOL is powered by the core voltage, and there are two types of IOB voltages: some circuits are powered by the core, and other circuits are powered by the core voltage.

Powered by the IO power supply. IO has a variety of level standards, and the corresponding power consumption is listed separately.

Source		Power			
Supply	Volt	Leakage	Static	Dynamic	Total
Vccint	1.2	0	0	0	0
Vccio1.2V	1.2	0	0	0	0
Vccio1.5V	1.5	0	0	0	0
Vccio1.8V	1.8	0	0	0	0
Vccio2.5V	2.5	0	0	0	0
Vccio3.3V	3.3	0	0	0	0

The power consumption generated in the core voltage (Vccint) domain is called logic power consumption, and in the IO voltage domain (contributed by a variety of levels)

The total power consumption generated is called buffer power consumption.

Power	%
Logic	0
Buffer	0

Pin Used	%
0	0

Click "+" to add a new configuration, and double-click under the corresponding column to enter the content. Name is the configuration name,

Not repeatable; the Standard menu indicates the supported level standards, and the DriveStrength menu indicates the currently selected level

The drive strength supported by the standard; Direction has INPUT/OUTPUT/INOUT; Num of Pin is the type of Pin

Quantity; select the series of idelay/odelay in IOL, the maximum series is selected from the series supported by IOLE, when Pin is

ODelay is not selectable when INPUT, and IDelay is not selectable when Pin is OUTPUT; Clock is the clock frequency;

Toggle Rate The default toggle rate is 12.5%, and the user can modify it within the range of 0~100%; Data Rate is divided into

BYPASS/SDR/DDR/DDR2; Output enable range is 0~100%, and cannot be edited when Pin is INPUT;

The default value of Output Load data is 0, which is optional. The calculation can be triggered without data. When Pin is INPUT, Output

Load is not editable.

After filling in the required data, you can see the calculation results for this configuration. If the process of summary page

Angle, temperature, and voltage values change, and here will change as well.

Name	I/O Setting						Activity				Power				
	Standard	DriverStrength	Direction	Num of Pin	IDelay	ODelay	Clock (MHz)	Toggle Rate (%)	Data Rate	Output Enable (%)	Output Load (pF)	Signal Rate (MHz)	Vccint	Vccaux	Vccio
io1	LVDS 2.5V		INPUT	24	4		125	12.5	SDR	0		15.625	0.004	0	0.026

Among them, the Vccint, Vccaux, and Vccio values calculated by power are their respective static and dynamic total values.

And, the summary in the upper left corner lists the power consumption of static and dynamic according to different level standards.

Power		
Vccint	Vccaux	Vccio
0.004	0	0.026

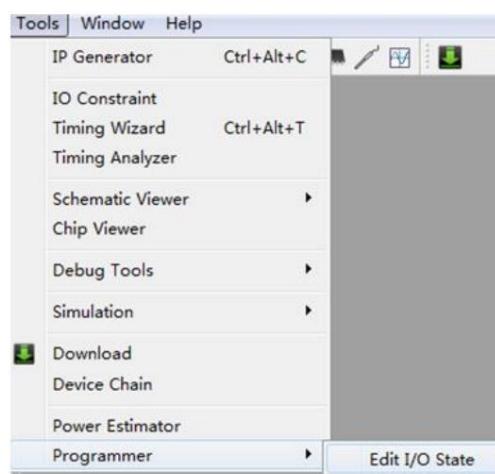
Source		Power				
Supply	Volt	Leakage	Static	Dynamic	Total	
Vccint	1.2	0	0	0.004	0.004	0.004
Vccio1.2V	1.2	0	0	0	0	0
Vccio1.5V	1.5	0	0	0	0	0
Vccio1.8V	1.8	0	0	0	0	0
Vccio2.5V	2.5	0	0.025	0.001	0.026	0.026
Vccio3.3V	3.3	0	0	0	0	0

8.7 I/O State Editor

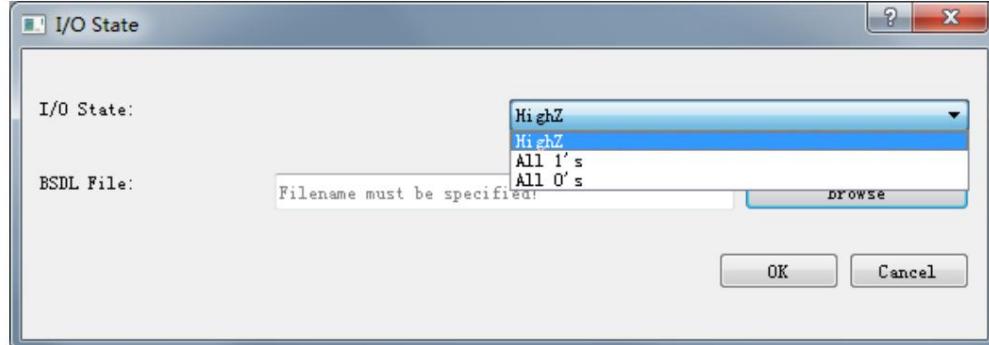
The I/O State Editor tool can specify specific output values of IO through JTAG boundary scan instructions, providing high power

Flat, low level and tri-state output, and after selecting a specific value to output, the chip can be refreshed by adding the refresh command.

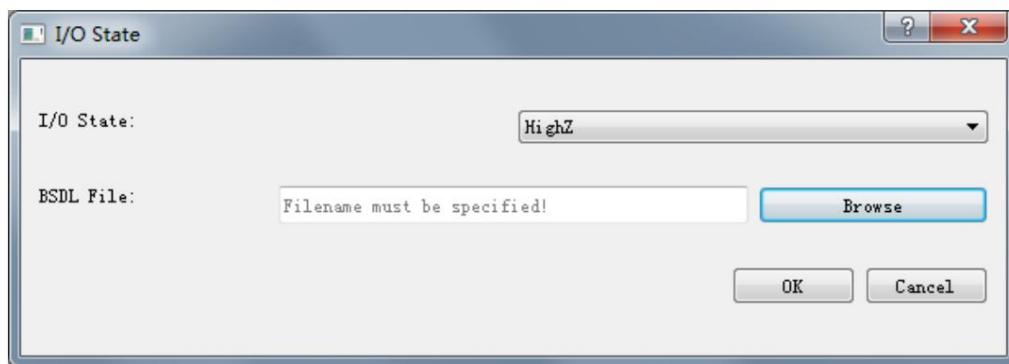
The interface can be opened via Tools -> Programmer -> Edit I/O State.



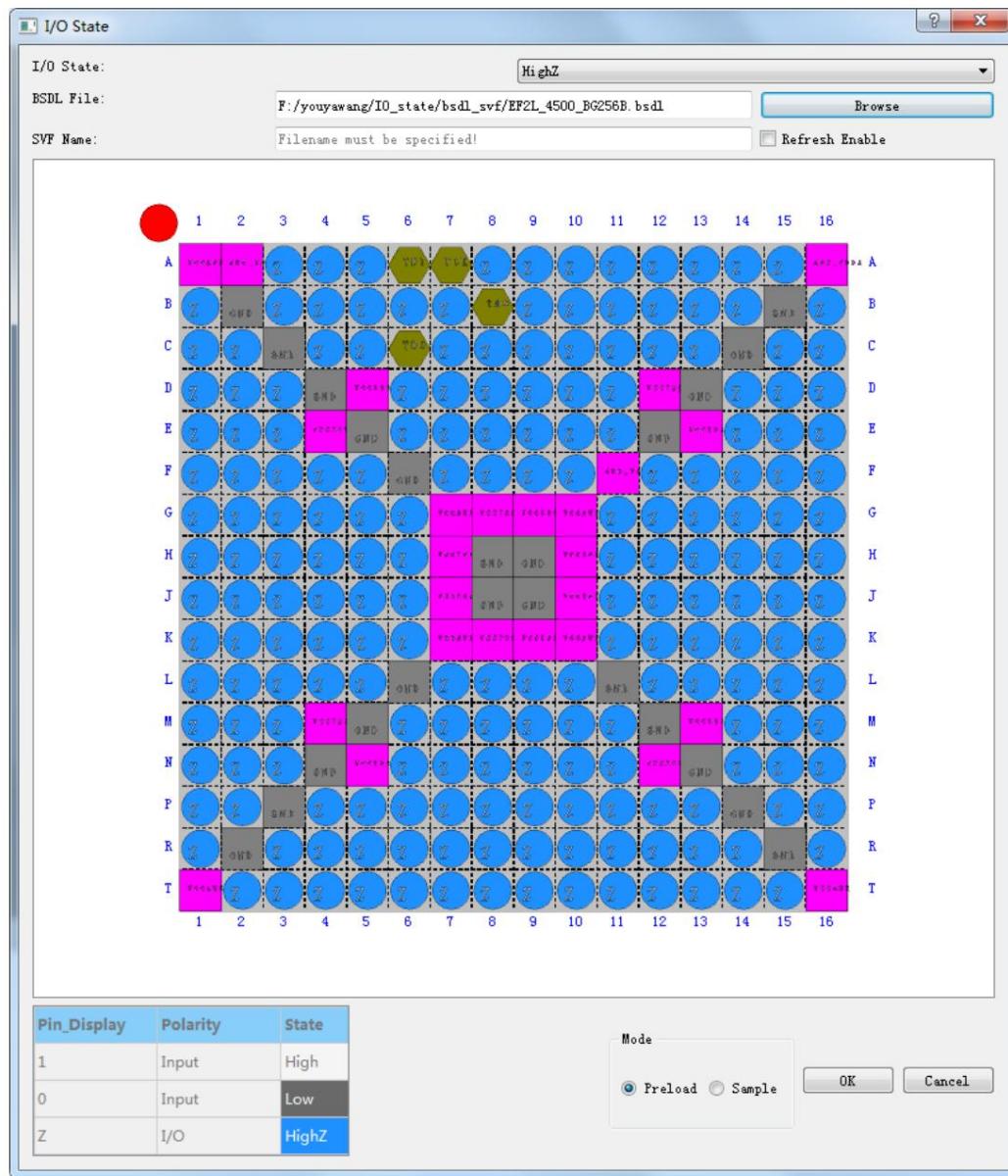
I/O state: Set the initial value of all IO, the optional value is HighZ/All 1's/All 0's



Browse: select *.bsdl file



The main interface of the I/O State Editor is as follows:

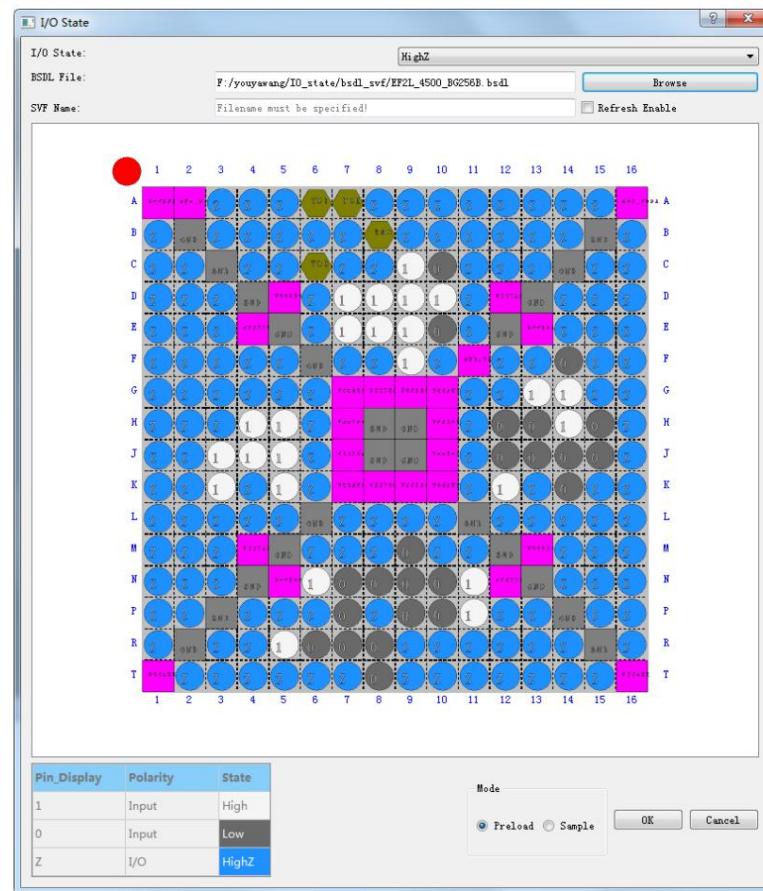


1. Left-click any IO pin to set and specify its output value or right-click to select the IO value to modify it at one time.

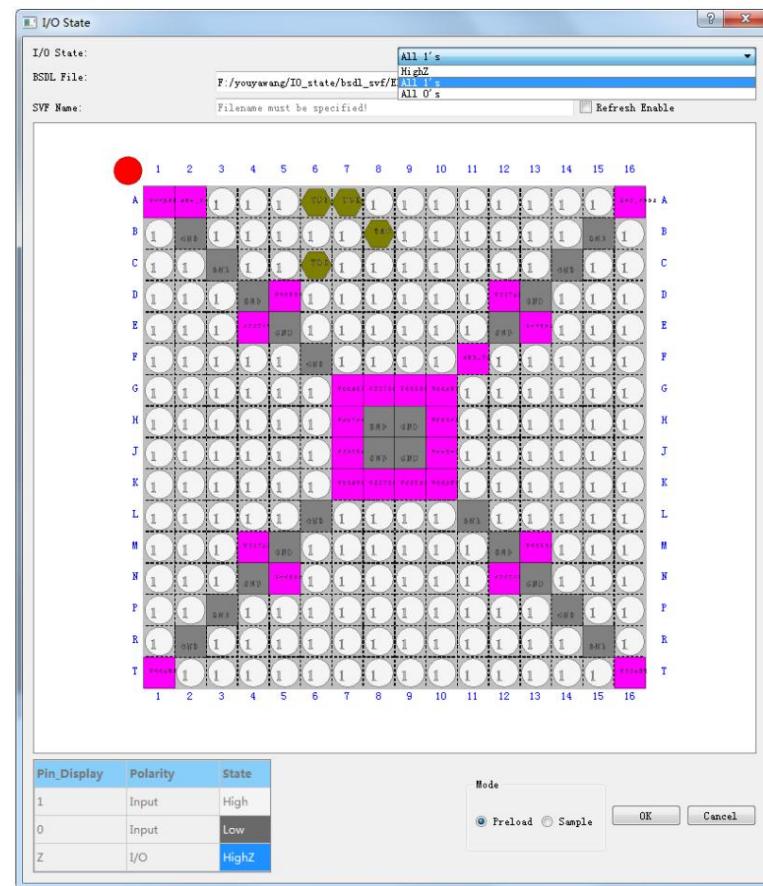
Change the IO value of all pins.

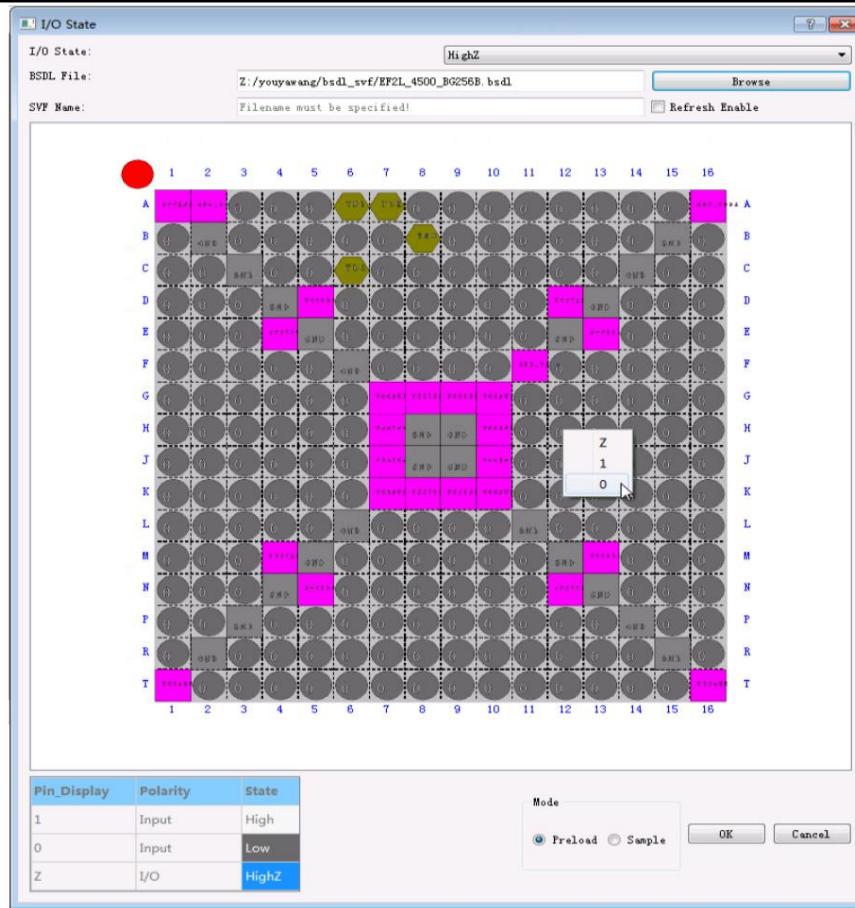
*Left-click on the IO pin to set the IO value of the specified pin:

The IO value changes cyclically in the order of Z -> 1 -> 0 -> Z -> 1 -> 0



Right-click to select or modify all IO values in the drop-down menu at I/O State:





2. SVF Name: Specify the name of the generated SVF file.
3. Refresh Enable: Enable the "refresh" function, the refresh command refreshes the chip.
4. Mode: Default Preload, optional Sample. Sample is mainly sampling verification.
5. After editing, click "OK" to generate the SVF file.

9 Appendix

9.1 ADC Constraint Description

The ADC (Anlogic Design Constraint) file is used as the user physical constraint file of the TD software, including the

User-specified constraints related to various types of pins and unit physical information.

1. Pin characteristic constraints

Defined as follows:

```
set_pin_assignment {pin_name} {attributes}
```

Pin_name: the pin name of the circuit

Attributes: Various physics-related attributes, currently supported constraints include BANK, LOCATION,

IOSTANDARD, DRIVESTRENGTH, PULLTYPE, PREEMPHASIS,

10 kinds of SLEWRATE, DIFFRESISTOR, PCICLAMP, PACKREG, etc. in,

For location constraints (LOCATION), it will check whether the target location is legal, and whether there are multiple

pins are assigned to the same location, once the ADC file is added, it is required to lock all pins

fixed location.

Note: PreEmphasis pre-emphasis technology is a kind of pre-emphasis on the high-frequency components of the input signal at the transmitting end.

The signal processing method of line compensation can effectively improve the output signal-to-noise ratio. Only required for LVDS output

To be added, LVDS_E is an analog LVDS output with no pre-emphasis settings.

Format example:

```
> set_pin_assignment {sys_clk} {LOCATION=P23; IOSTANDARD=LVCMS18;
DRIVESTRENGTH=4; PULLTYPE=PULLUP; SLEWRATE=MED; PCICLAMP=ON;
PACKREG=ON; OUTDEL=1;}
```

Table 9-1 IO characteristic setting parameters

PAMERATER	VALUE
BANK	BANK1, BANK 2, BANK 3, BANK 4, BANK 5, BANK 6, BANK 7, BANK 8
LOCATION	The chip package is different, the number of pins and the name of the pins are different
IOSTANDARD	LVCMOS12,LVCMOS15,LVCMOS18,LVCMOS25,LVCMOS33,LVDS25, LVDS25_E, LVDS33, LVDS33_E, PCI33, LVTTL33, LVPECL33, LVPECL33_E
DRIVESTRENGTH	2, 4, 8, 12, 16, 20, NA
PULLTYPE	PULLUP, PULLDOWN, NONE, KEEPER
SLEWRATE	FAST, MED, SLOW
DIFFRESISTOR	100, NONE
PCICLAMP	ON, OFF
PACKREG	AUTO, ON, OFF

Note: The specific support of IOSTANDARD is subject to the data sheet of each series of chips.

For users who have differential pair output VCM and VOD setting requirements, they need to be manually set in the ADC file

The specific values of VCM and VOD, the format examples are as follows:

```
> set_pin_assignment {sys_clk} { LOCATION = A3; IOSTANDARD = LVDS33; VCM = 0.8; VOD = 350m }
```

VCM and VOD settings are only for LVDS25 and LVDS33, other levels are not supported. different devices for

The settings of VCM and VOD are different, and the specific parameter values that can be set for each device are shown in Table 9-2.

Table 9-2 Parameter range table of each device

Device	VOD(mV)	VCM(V)
EG4	150, 200, 250, 350*, 480	0.8, 0.9, 1.2*
EF2	150, 200, 250, 350*, 480	not support
EF3	150, 200, 250, 350*, 480	0.8, 0.9, 1.2*
AL3	150, 200, 250, 350*	0.8, 0.9, 1.2*

Note: (1) It is not recommended to set VOD to 480mV. (2) * means software default value, VOD default value is 350mV, VCM default value

The default value is 1.2V. (3) The VCM and VOD settings should be set according to the specific values given in the above table, and those beyond the range of the above table are not supported.

2. Element characteristic constraints

Defined as follows:

```
set_inst_assignment {inst_name} {attributes}
```

Inst_name: The unit instance name of the circuit, the name of the instance can be found in the generated netlist file (prj_gate_sim.v)

Find in.

Attributes: Currently only supports the location constraint Location, the location can be found in the Chip Viewer. Chip

For the detailed user manual of the Viewer, please refer to the 8.2 Chip Viewer chapter of this document.

Format example:

```
> set_inst_assignment {PLL_INST1} {location = x34y37z0;}  
> set_inst_assignment {PLL_INST2} {location = x0y0z0;}  
> set_inst_assignment { _al_u451||cd/lcd_rs_reg } {location = x21y17z1;}
```

9.2 SDC Constraint Description

The Timer module of TD supports a subset of common commands in the SDC standard format for the timing constraints input by the user.

The following lists the SDC commands supported by the TD and the associated parameter options.

1. Referencing object related:

a) **all_clocks / all_inputs / all_outputs / all_registers**: refer to all objects of this category;

b) **get_cells / get_pins / get_nets** [-hierarchical] [-nocase] [-nowarn] <filter>

Returns a collection of cells/pins/nets in the design, -hierarchical means hierarchical search, will

Match the corresponding name in the next module, -nocase means case insensitive, -nowarn means no match

It also doesn't give a warning on success.

c) **get_clocks / get_ports** [-nocase] [-nowarn] <filter>

Returns the set of clock/input and output ports, the parameters have the same meaning as above.

2. Clock setting related (clock information is the most basic timing constraint):

a) **create_clock** -add -name <string> -period <double> -waveform <string> target

Define a clock: target specifies the clock source, which can be nets or ports, pins, if target is

Empty means that a virtual clock is defined; -name specifies the clock name, if the item is empty, the clock name is

The first item in the target list; -period is the period, this option must be specified, and the value must be greater than 0; -

waveform specifies the time point of the first rising edge and the first falling edge, temporarily only supports per-cycle packets

In the case of two clock edges; -add indicates that when the clock has been defined on the target pin, add the new

The clock is added to this pin, otherwise the existing clock is overwritten, mainly used in the case of clock multiplexer.

Format example:

```
> create_clock -name sys_clk -period 20 -waveform {0 10} [get_ports sys_clk]
```

b) **create_generated_clock** -add -name <string> -source <list> -divide_by <double> -
 multiply_by <double> -edges <string> -duty_cycle <double> -invert -edge_shift <string> -master_clock <string>
 -phase <double> target

Define a derived clock that belongs to the clock domain where the master clock resides and will also have and

The same start point as the master clock.

-source specifies the source point where its master clock is located, which can be nets or ports, a list of pins, -

master_clock specifies the name of the master clock; -name specifies the clock name, if this item is empty

Then the clock name is the first item in the source list; target is the source point of the currently generated clock. -add instructions are in

If the clock has been defined on the target pin, add a new clock to the pin, otherwise the current generation

The clock is ignored, unlike when master clock is defined.

The cycle and waveform of the generated clock are adjusted by the master clock, -divide_by / -multiply_by refers to the frequency

rate divided/multiplied by the specified multiple, -invert is used in conjunction with these two options to invert the clock waveform, while

duty_cycle is used with the -multiply_by option to adjust the duty cycle; the -phase option is used to adjust

Phase shift, it is recommended to input parameters within (0, 360), and modulo processing will be performed if it exceeds 360.

The -edges option contains three integers specifying the first rising edge of the newly generated clock, the first falling edge

edge, the second rising edge corresponds to the first edge of the source clock; the -edge_shift option will specify the -edges option

The offset of the three clock edges in the

bits (default ns).

Format example:

```
> create_generated_clock -divide_by 1.25 -source [get_ports clk] -name sys_clk [get_ports sys_clk]
```

c) **derive_pll_clocks [-gen_basic_clock]**

Automatically generate clock constraints on all used PLL clk[x] ports, the frequency and phase of the generated clock will be

Strictly follow the parameter settings inside the PLL. -gen_basic_clock will be defined on the corresponding PLL refclk

The base clock of the FIN frequency, otherwise it will automatically search for the refclk pin and the clock defined on the connected net.

The clock generated by this command will only take effect when the flow is running, so in the subsequent settings of the timing wizard

Can't cite yet.

```
> derive_pll_clocks -gen_basic_clock
```

d) **derive_clocks** -period <double> -waveform <string>

Specify a default clock on the clock pins of each undefined clock, -period is the period, the option

The item must be specified, and the value must be greater than 0; -waveform specifies the first rising edge and the first falling edge

At the time point, only the case where each cycle contains two clock edges is temporarily supported.

Format example:

```
> derive_clocks -period 10 -waveform {0 5}
```

e) **set_clock_latency** -clock <list> -max -min -source delay

Set the delay of the clock, delay is the value of the delay; -clock specifies the list of clocks;

The -max/ -min option specifies the maximum/minimum delay specified by this command, and the default is the same for both;

The -source option specifies that the source latency is specified, otherwise it is the network latency.

The network latency is replaced by the actual interconnect latency in post-routing timing analysis.

```
> set_clock_latency -clock [get_clocks {clk_vga_25m}] -source 2
```

f) **remove_clock_latency** -clock <list> -source target

Cancel the previously set clock delay, the meaning of each option is the same as above.

Format example:

```
> remove_clock_latency -source [get_clocks clk_vga_25m]
```

g) **set_clock_uncertainty** -setup -hold uncertainty target

Set the uncertainty value of the clock, currently only supports setting for a single clock itself and does not support spanning time

Uncertainty definition of clock domain; target is a list of clocks, uncertainty is a numerical item; -setup/-

The hold option indicates that this command corresponds to the value corresponding to the maximum/minimum path timing analysis, if

If this option is not specified, both checks take effect at the same time.

Format example:

```
> set_clock_uncertainty --setup/-hold 1.00 [get_clocks clk]
```

h) **remove_clock_uncertainty** -setup -hold target

Remove the previously set clock uncertainty value, the meaning of each option is the same as above.

Format example:

```
> remove_clock_uncertainty --setup/-hold [get_clocks clk]
```

3. Timing path constraint setting:

a) **set_input_delay / set_output_delay** -clock <list> -clock_fall -max -min delay target

Set the delay of the input/output port, the delay item is the delay value, and the target can only be the object column of ports

Table; -clock specifies the clock information corresponding to the delay; -clock_fall specifies the reference clock below

Falling edge sampling; the -max/-min option specifies the maximum/minimum delay value set by this command, and the default is

same.

Format example:

```
> set_input_delay -clock sys_clk 1.0 [all_inputs]
```

```
> set_output_delay -clock lcd_clk 3 [get_ports {disp_RGB} {led} {sm_bit}]
```

b) **remove_input_delay / remove_output_delay -max -min target**

Remove the input/output delay set before, the meaning of each parameter is the same as above.

Format example:

```
> remove_input_delay -clock sys_clk 2 [get_ports {data[0]}]
> remove_output_delay -clock sys_clk -0.55 [all_outputs]
```

c) **set_max_delay / set_min_delay -from <list> -to <list> -through <list> delay**

Set the allowable maximum/minimum delay of the timing path, the delay item is the delay value; -from must be the timing path

The starting point of the trail, that is, the input list, or regs, or clocks, or the pins corresponding to the position defined by clock;

-to must be the end point of the timing path, that is, a list of outputs, or regs, or clocks, or clock definitions

The pins corresponding to the position of

The intermediate point that must be passed. When there are multiple through options, the target timing path must pass through each one in turn.

middle point.

Format example:

```
> set_max_delay -from [get_ports {sw}] -to [get_ports {sm_bit}] -through [get_nets {lcd/delay_cnt[2]}] 1
```

d) **set_false_path -setup -hold -from <list> -to <list> -through <list>**

Sets the timing path as a false path, so it is not subject to timing analysis; the -setup/-hold option specifies

Target paths are not analyzed during setup/hold check. -from specifies the starting point of the constraint, which can be clocks or

A list of inputs, regs, pins, etc., -to specifies the endpoint of the constraint, which can be clocks or outputs

A list of regs, pins, etc., -through specifies the intermediate point that the timing path must pass through, but is

List of pins or nets. After specifying, the system will automatically trace back to the start and end points of the timing path.

Format example:

```
> set_false_path -from [get_clocks clk] -to [get_clocks sys_clk]
```

```
> set_false_path -from [get_clocks {clk_vga_25m}] -to [get_ports {sm_bit}] -through [get_nets {dpram_top0/lfsr_done}]
```

e) **set_multicycle_path** -setup -hold -start -end -from <list> -to <list> -through <list>

multiplier

Set the timing path that allows multiple clock cycles to delay, the multiplier item is the number of clock cycles; -setup/-hold

The option sets the type of timing path constrained by this command. When only the -setup option is used, setup check will allow

Allow timing paths to use up to N clock cycles, but at the same time hold check becomes

Short after N-1 clock cycles. When only the -hold option is used, the hold check constraint is set to N and

Constraints that do not affect setup check.

In normal usage scenarios, in order to allow the setup check to use multiple clock cycles without affecting the hold check,

Two commands are required to be used together, that is, to set a setup constraint of N cycles, and then cooperate with a N-1

Period hold constraint.

-start/-end is the option used when crossing the clock domain, and the specified delay is the week corresponding to the launch/capture clock

Period, by default, the setup check uses the capture clock and the hold check uses the launch clock.

-from specifies the starting point of the constraint, which can be a list of clocks or inputs, regs, pins, etc., -to specifies

The endpoint of the constraint, which can be a list of clocks or outputs, regs, pins, etc., -through specified

The intermediate point that this timing path must pass through, but is a list of pins or nets. After specifying, the system will automatically

Trace back to the beginning and end of the timing path.

Format example:

```
> set_multicycle_path -setup -end -from [get_clocks {clk_vga_25m}] -through [get_nets {uart uart_tx/num[3]}] 2
```

f) **set_clock_group** -exclusive -asynchronous -group <list>

Set groups for clock domains and do not analyze timing paths across clock groups. In general, the -exclusive option table

indicates that these groups of clocks do not logically appear at the same time, while -asynchronous indicates completely unrelated clocks,

However, in terms of specific implementation and effect, these two options are the same, and this command will be displayed in all -group columns

A set of false path constraints is defined between outgoing clocks.

```
> set_clock_groups -exclusive -group [get_clocks {clk_vga_25m}] -group [get_clocks {sys_clk}]
```

g) **set_clock_route** <net_name>

Used to specify that a specific clock net does not go through a dedicated clock interconnect. For the number of clocks in design exceeds TD

When the limit is set, you can manually specify which clock nets do not follow the clock nets. If the specified clock does not follow the clock network

When the network is connected, the layout and routing will fail, and the TD will give an error in time.

```
> set_clock_route lcd12864_en_pad
```

9.3 Description of main warning messages of TD software

ID	illustrate	suggestion
(USR) User input warning/error message		
USR-6010	ADC position is not assigned to a pin	Add ADC information in IO Constraint
USR-6011	Clock pin assignment is not optimal	Group GCLKIOs as suggested
USR-6101	Unable to match clock object specified in target list Check netlist and sdc file, correct sdc error	
USR-6102	The sdc command does not conform to the syntax or usage requirements. Check the sdc documentation and specify constraints as required	
USR-6103	set_clock_group did not specify enough as requested clock	Check the sdc documentation and specify constraints as required
USR-6104	sdc has a master clock defined by the same name	Check the sdc file to eliminate duplicate definitions
USR-6105	Match to generate clock derived from multiple masters clock	Check the sdc file for errors in the clock constraints error or wildcard problem
USR-6106	sdc has a duplicate definition of generate clock Check the sdc file to eliminate duplicate definitions	
USR-6107	PLL without clock output	Check netlist or replace other sdc commands
	The input reference frequency of the USR-6108 PLL is 0 Check the PLL module configuration	
USR-6109	PLL's target output frequency is 0 Check PLL block configuration	
USR-6111	The clock option conflicts with the target object of this command Remove what was specified by the -clock option	
USR-8002	Unable to open a file to check whether the file name and path are correct	
USR-8011	Command option error correction input	
USR-8012	Command option value error correction input	
USR-8019	Illegal package using correct package name	
USR-8020	The number of ADC analog-to-digital conversion modules in the user project exceeds total	Instantiate a reasonable number of ADCs based on chip capacity
USR-8021	has up to 1 ADC with temperature sensor function	Instantiate a reasonable number of temperature sensor ADCs based on chip capacity target
USR-8050	The correct format for pin_assignment is {pin_name} {attributes}	Please use the correct format
USR-8054	inout pins cannot be used as virtual IO Please correct the pin constraint settings	
USR-8055	pin_assignment ADC pin location is repeatedly defined, please check ADC definition	
USR-8056	pin_assignment pin location information is missing Add pin location definition	
USR-8057	pin_assignment IOSTANDARD cannot be set Set to LVDS25_E	Please select another IOSTANDARD
USR-8058	pin_assignment IOSTANDARD cannot be set Set to LVDS25	Please select another IOSTANDARD
USR-8059	pin_assignment set an illegal value inst_assignment	Please correct the settings
USR-8060	instruction correct format is {inst_name} {attribute=value}	Use the correct format
USR-8062	inst_assignment command format is incorrect use correct format	
USR-8068	pin_assignment GCLKIOP_5 and GCLKIOP_8 cannot be used at the same time	Please use other GCLKIO combinations

TangDynasty

Software Manual

USR-8069	pin_assignment GCLKIOP_4 and GCLKIOP_9 cannot be used at the same time	Please use other GCLKIO combinations
USR-8070	pin_assignment GCLKIOP_0 and GCLKIOP_10 cannot use both inst_assignment	Please use other GCLKIO combinations
USR-8071	command format incorrect use correct format	
USR-8072	Unable to find Instance	Please check if the instance name is correct
USR-8073	The pin_assignment command position is not set correctly, please check and set the correct position	set the correct position
USR-8074	pin_assignment command a pin position setting is not	Please check and set the correct location
USR-8075	Correctly each IOBank has more than one IOSTANDARD	Please check the IOSTANDARD setting for this bank
USR-8101	Design not imported	Before making timing constraints, make sure to import the design first
USR-8102	The clock period value is invalid	Check sdc statement, period value should be of float or integer class type
USR-8103	clock period value overflow sdc syntax	Specify a legal value
USR-8104	error: append master clock with -add lack of name	When appending the master clock with -add, the clock name must be specified
USR-8105	sdc syntax error: clock constraint lacks name and target	When creating a clock constraint, the clock name must be specified. missing If the target is lost, a virtual clock will be created.
USR-8106	Unable to match specified in create_clock statement clock starting point	The name of the specified start pin may not be the same as the actual netlist. or has been optimized. Compare sdc and netlist, make set correctly
USR-8107	design sdc not imported Syntax	Before making timing constraints, make sure to import the design first
USR-8108	error: must be created to create generate clock sdc syntax error specifying	Add target to this clock in the constraint statement
USR-8109	this clock origin: append generate clock with -add lack of name or master clock name	Set a name for this clock in the constraint statement or specify all the name of the masterclock
USR-8110	sdc syntax error: no pointer to generate clock Determine the starting pin of the corresponding master clock	Specify the corresponding master clock for generate clock start pin
USR-8111	Can't match in create_generate_clock statement The specified master clock starting point	The name of the specified start pin may not be the same as the actual netlist. or has been optimized. Compare sdc and netlist, make set correctly
USR-8112	Can't match master when creating generate clock clock	Check that the name of -master_clock is correct. another The possibility is that the corresponding master clock has been overwritten.
USR-8113	Numerical overflow on clock edge	Check that the constraint values are entered correctly
USR-8114	Clock period value overflow after phase shift	Check that the constraint values are entered correctly
USR-8115	Lack of correct times when creating generate clock Frequency, divide or phase shift information	Check constraints, correct errors
USR-8116	Numerical overflow of clock frequency	Check constraint settings for generate clock
USR-8117	Phase shift difference is more than 360° or less than 0° When	Check that the constraint values are entered correctly
USR-8119	using derive_pll_clocks, the pll configuration is lacking corresponding information	Check the PLL block configuration

USR-8121	uses derive_pll_clocks, the pin information of PLL refclk cannot be obtained	It is recommended to specify the -gen_basic_clock option
USR-8122	design not imported	Before making timing constraints, make sure to import the design first
USR-8123	Clock latency value overflow	Check that the constraint values are entered correctly
USR-8124	When using set_clock_latency, no phase is specified should be the starting point of the clock	The value of latency is imposed on the starting point of the clock and must be The corresponding clock pin must be specified
USR-8125	When using set_clock_latency, it is not possible to match the index the starting point of the corresponding clock	The name of the specified start pin may not be the same as the actual netlist. or has been optimized. Compare sdc and netlist, make set correctly
USR-8126	Clock uncertainty value overflow	Check that the constraint values are entered correctly
USR-8129	Error reading sdc file	View the latest specific error msg in the log and sdc sentence to be modified accordingly
USR-8130	Could not match clock name specified in get_clocks say	Compare the netlist and the sdc file, and correct the errors.
USR-8131	The cell name specified in get_cells cannot be matched against the netlist and sdc file, and the error is corrected.	
USR-8132	Unable to match net names specified in get_nets	Compare the netlist and the sdc file, and correct the errors. or correspondingly net has been optimized away.
USR-8133	Could not match reg names specified in get_regs (input output net name)	Compare the netlist and the sdc file, and correct the errors.
USR-8134	can not match the pin name specified in get_pins, compare the netlist and sdc file, and modify the error.	
USR-8135	The port name specified in get_ports cannot be matched against the netlist and sdc file, and the error is corrected.	
USR-8136	set_input_delay cannot match for input pins reference clock	Check the netlist and sdc file to ensure that the specified reference clock exists and the name is correct
USR-8137	set_input_delay is not specified for input pins reference clock	Specify reference clock for input pin
USR-8138	set_input_delay cannot be used for input pins reference clock matches reference pin	Check the netlist and sdc file to ensure that the specified reference pin exists and the name is correct
USR-8139	set_input_delay cannot match input pins	Check the netlist and sdc file to ensure the specified input pins exists with no errors in the name
USR-8142	sdc syntax error: set_multicycle_path cannot Specify both -setup and -hold	Read the user guide to understand the use of -setup/-hold respectively and the behavior when not in use and according to actual requirements choose
USR-8143	sdc syntax error: set_multicycle_path cannot Specify both -start and -end	Read the user guide to understand the use of -start/-end respectively to and the behavior when not in use and according to actual requirements choose
USR-8144	Unrecognized filename	Timing constraints file must be a .sdc file
(HDL) Verilog or vhdl warning/error message		
HDL-5007	HDL source code issues	Modify the HDL source code according to the prompts
HDL-8007	HDL source code error	Modify the HDL source code according to the prompts

(RUN) Flow warning/error message		
RUN-6006	Clock register is not placed in IO PAD	Please update the design file so that the clock registers can absorb the IO PAD in
RUN-6007	Data register is not placed in IO PAD	Please update the design file so that the data registers can absorb the IO PAD in
RUN-6011	The input frequency of PLL cannot be empty	correct input
RUN-8002	The number of threads cannot be set to 0	Please set the number of threads correctly
RUN-8416	PLL is missing the specified parameter or the parameter value is incorrect	correction input
RUN-8419	The parameter value 'EXTERNAL' for the feedback path is no longer To be	Iteratively generate PLL blocks using IPGen
RUN-8420	used in PLL source synchronous mode, the data register must be Use the same IO standard	Please adjust the constraints of the data register IO
RUN-9990	Number of threads set to non-integer	Please use integer
(SYN) Synthesis warning/error message		
	Please modify the design file for SYN-5011 instance pins that are not driven	
SYN-5012	Please modify the design file for undriven module pins	
SYN-5013	Please modify the design file for the undriven net	
SYN-5014	is driven by the wire net in Warning-5013, please modify the design file	
There is a Latch in the	SYN-5015 module, please modify the design file	
SYN-5020	Nets using "synthesis keep" are not driving any pin, it will be removed	Please modify the design file
SYN-5021	does not support the realization of this module type, please modify the design file	
SYN-5023	The minimum GSRN fanout number is set to a non-integer, please correct the setting	
SYN-5024	The minimum GSRN fanout number is set to a non-positive integer, please correct the setting	
SYN-5025	Use 0, 1 or x to drive those undriven leads foot and wire mesh	Please modify the design file
SYN-5026	Internal network does not belong to instance	Please modify the design file
SYN-5027	Worst timing path not found	Please modify the constraints file
SYN-5031	The required time is less than 0. The	Please modify the constraints file
SYN-5032	number of module ports is incorrect or the bit width does not match, can not be realized	Please check the port of this module
SYN-6001	PAD pin connection error	Modify the source code
SYN-6002	PAD pin connection error	Modify the source code
SYN-6003	PAD pin connection error	Modify the source code
SYN-6005	PAD pin connection error	Modify the source code
SYN-6006	PAD location setting error	Modify the source code
SYN-6007	PAD attribute setting error	Modify the source code
SYN-6008	PAD attribute setting error	Modify the source code
SYN-6009	syn-ip flow appears inout	Check Code Confirmation

SYN-6010 IO related	IP appears in syn-ip flow	Check Code Confirmation
SYN-6012 sram floating pin		Enter as needed
SYN-6013 sram floating pin		Enter as needed
SYN-6014 IO floating		give logical connection
SYN-6511	User calls phy-bram must be given or not given at the same time RID/WID	Modify the source code
SYN-6521 bram init	file is wrong to modify init-file	
SYN-6522 bram init	file is wrong, modify init-file	
SYN-6531 LUT mapping error check source code combinational logic		
SYN-8101 The parameter is not set, please correct the setting		
SYN-8102 The value of the parameter is incorrect, please correct the setting		
SYN-8103 The value of the parameter is incorrect, please correct the setting		
SYN-8105 net is driven by multiple pins, please modify the design file		
SYN-8106 drives the input pins of the net in Error-8105, please modify the design file		
SYN-8107	Drive the input and output leads of the net in Error-8105 foot	Please modify the design file
SYN-8108	In BRAM single port mode, A port and B port inconsistency	Please check the information of the A and B ports of the BRAM
The data bit width of	SYN-8109 BRAM is 0	Please correct the BRAM data width
SYN-8110	Data bit width of A port and B port of BRAM inconsistent	Please correct the BRAM data width
SYN-8111	Address width of A port and B port of BRAM inconsistent	Please correct the BRAM bit width
SYN-8112	Size of A port or B port of BRAM is 0	Please correct the BRAM settings
SYN-8113	BRAM's A port and B port have different capacities To	Please correct the BRAM settings
SYN-8114 The driver of CLKBUF or CLKMUX is incorrect, please modify the design file		
SYN-8115 Find the undriven net, please modify the design file		
SYN-8116 Find the net driven by multiple pins, please modify the design file		
SYN-8117	The wire net can only drive the input and output pins of the module, not the I/O pins of the driver instance	Please modify the design file
SYN-8119 Tri-state cache can only exist in IO		Please modify the design file
SYN-8120	The instance is neither a primitive instance nor a submodule Piece	Please check the design file
SYN-8121 Excessive memory usage		Confirm hardware configuration
SYN-8201 The number of IPs used exceeds the limit		Modify the design
SYN-8202 The number of IPs used exceeds the limit		Modify the design
SYN-8203 ADC is wrong		Modify the design
SYN-8204 ADC is wrong		Modify the design
SYN-8205 map io error		Modify pad related logic
SYN-8206 map io failed		Modify pad related logic

SYN-8207 IO related IP is not mapped		Please connect to the io pin correctly.
SYN-8212 AST IP setting error		Modify the design
SYN-8213 AST IP setting error		Modify the design
SYN-8214 MIPI IP setting error		Modify the design
SYN-8215 MIPI IP setting error		Modify the design
SYN-8216 MIPI IP setting error		Modify the design
SYN-8217 MIPI IP setting error		Modify the design
SYN-8218 MIPI IP setting error		Modify the design
SYN-8219 The number of IPs used exceeds the limit		Modify the design
SYN-8220 IDDR IP setting error		Modify the design
SYN-8402 The number of IPs used exceeds the limit		Modify the design
SYN-8412 PHY-BRAM setting error		Modify the design
SYN-8413 PHY-BRAM setting error		Modify the design
SYN-8414 PHY-BRAM setting error		Modify the design
SYN-8416 LOGIC-BRAM setting error		Modify the design
SYN-8417 LOGIC-BRAM setting error		Modify the design
SYN-8422 LOGIC-BRAM setting error		Modify the design
SYN-8423 LOGIC-BRAM setting error		Modify the design
SYN-8431 LOGIC-BRAM setting error		Modify the design
SYN-8432 LOGIC-BRAM setting error		Modify the design
SYN-8433 LOGIC-BRAM setting error		Modify the design
SYN-8434 LOGIC-BRAM setting error		Modify the design
SYN-8441 BRAM init file cannot be opened		check path
SYN-8442 LOGIC-BRAM setting error		Modify the design
SYN-8443 BRAM init file mismatch		modify mif
SYN-8444 BRAM init file mismatch		modify mif
SYN-8447 Command line invocation error		Modify command
SYN-8448 Command line call error		Modify command
SYN-8451 Combination logic loop found		Modify the design
SYN-8452 LUT mapping error		Check combinatorial logic
SYN-8453 LUT mapping error		Check combinatorial logic
SYN-8461 DSP is not enough		Use IMPLEMENT:DSP sparingly
SYN-8609 Combinatorial logic error found, additional info		Modify the design
SYN-8610 Combinatorial logic error found, additional information		Modify the design
SYN-8611 Combinatorial logic error found, additional information		Modify the design
SYN-8702 SanityCheck finds errors		Modify the design
SYN-8703 does not support setting this net as route net		Cancel settings
SYN-8704 clock net resources are not enough		set more nets as route nets
(PHY) Physical warning/error message		
PHY-7001 instantiated OSC module output driver error		OSC block can only drive PLL or OSC_DIV

TangDynasty

Please use a

PHY-8004 License check failed		valid license for the software manual
PHY-9008 project does not contain any instance		Please check input design
PHY-9009	The project contains more mslice than chip mslice capacity quantity	Please choose the device reasonably
PHY-9010	Project contains more lslice than chip lslice capacity quantity	Please choose the device reasonably
The PHY-9013 project	contains the number of embs that exceeds the emb capacity of the chip, please	choose the device reasonably
The PHY-9014 project	contains more dsp than the chip dsp capacity, please choose the	device reasonably
PHY-9015	Project contains more emb32 than chip emb32 capacity	Please choose the device reasonably
The PHY-9016 project	contains more pads than the chip pad capacity, please choose the	device reasonably
The PHY-9017 project	contains more plls than the chip pll capacity, please choose the	device reasonably
PHY-9018	The number of configs contained in the project exceeds the chip config capacity quantity	Please choose the device reasonably
The PHY-9019 project	contains more ADCs than the chip ADC capacity, please choose	the device reasonably
The PHY-9020 project	contains more osc than the chip osc capacity, please choose the	device reasonably
PHY-9021	The number of oscdvs contained in the project exceeds the capacity of the chip oscdvs quantity	Please choose the device reasonably
The PHY-9022 project	contains more MCUs than the chip MCU capacity, please choose	the device reasonably
PHY-9023	Project contains more pwrmt than chip pwrmt The	Please choose the device reasonably
PHY-9024	capacity item contains more bram128K than chips bram128K capacity	Please choose the device reasonably
PHY-9025	project contains more bram256K chips bram256K capacity	Please choose the device reasonably
PHY-9026	project contains more bram18K chips bram18K capacity	Please choose the device reasonably
PHY-9123	engineering An IO Bank contains more than one IOCLK_DIV	Each IO bank contains at most one IOCLK_DIV
PHY-9124 instantiated	IOCLK module can only drive illegal	An instantiated IOCLK module can only drive Pad or CLKDIV
The PHY-9128 project	contains a temperature sensor module. Please use the temperature	sensor according to the chip resources.
PHY-9129 temperature	sensor cannot be used alone, please configure ADC module for	temperature sensor
PHY-9130 temperature	sensor cannot be used alone, please configure ADC module for	temperature sensor
PHY-9131 project	contains AST module. Please use AST according to chip resources.	
PHY-9132 AST module	setting is wrong, please set AST correctly	
PHY-9133 AST module	setting error, wrong pin assignment Please assign correct AST pin	
PHY-9134 AST module	setting error, missing pin assignment please assign correct AST pin	
PHY-9135 MIPI IO	setting error, invalid pin Please set MIPI IO correctly	
PHY-9136	MIPI IO setting error, pin is not Positive LVDS location	Please set the MIPI IO pin assignments correctly

PHY-9137	MIPI IO setting error, pin is not Positive LVDS location	Please set the MIPI IO pin assignments correctly
PHY-9138	MIPI IO setting is wrong, the pin assignment is missing, please set the MIPI	O pin assignment correctly
PHY-9144	IOCLK drives IO of different IOBANKs	The IO driven by IOCLK should be the same as the Bank
(TMR) Timing warning/error message		
TMR-5001	Missing Timing Constraints	Add timing constraints
TMR-5009	Some or all clock signals lack constraints	Check the circuit netlist and timing constraints, for when constraints are lacking Bell add constraints
TMR-5504	Match to generate clock derived from multiple masters clock	Check the sdc file for errors in the clock constraints error or wildcard problem
TMR-5506	cannot find the master clock of generate clock	Check the sdc file for errors in the clock constraints error or wildcard problem
TMR-5508	The source of the derived clock was searched while building the clock tree point	If it is confirmed that a derived clock is defined on this node, then This message can be ignored
TMR-6004	The detailed wiring of the wire net is not completed and the detailed wiring cannot be obtained. The timing information of	Rerun the flow to fix the wiring issue first.
TMR-6019	the routing The routing timing information	Rerun the flow to fix the wiring issue first.
TMR-6503	master clock constraint is repeated, and the coverage defined later is first. The defined	Check the sdc file and cancel the duplicate definition of master clock
TMR-6504	generate clock constraint is repeated, and the later defined is ignored slightly	Check the sdc file and cancel the duplicate definition of generate clock
TMR-6505	Could not match specified in set_input_delay constraint input pin	Check the netlist and sdc file to ensure the specified input pins exists with no errors in the name
TMR-6506	Could not match set_output_delay constraints specified in the output pin of	Check the netlist and sdc file to ensure the specified output pins exists with no errors in the name
TMR-6510	Unable to match clock specified by get_clocks	Check netlist and sdc file, correct sdc error
TMR-6511	Unable to match clock specified by get_pins	Check netlist and sdc file, correct sdc error
TMR-6512	Unable to match clock specified by get_ports	Check netlist and sdc file, correct sdc error
TMR-6513	Unable to search for a valid timing path for constraint checking	netlist and sdc file, correct sdc error
TMR-6515	Unable to search for timing path between specified clocks The object	Check netlist and sdc files, correct sdc errors
TMR-6523	specified by '-from'/'to' in the constraint sdc command is not valid startpoint/endpoint	Check netlist and sdc files, correct sdc writing errors
TMR-7006	Timing Path Start Lack of Clock Constraints	Check the netlist and sdc file and add the corresponding clock constraints
TMR-7008	Lack of clock constraints at end of timing path	Check the netlist and sdc file and add the corresponding clock constraints
TMR-8001	Timer initialization failed	Please check the netlist and rerun Flow
TMR-8102	cannot match the source pin of the clock	The name of the specified start pin may not be the same as the actual netlist. or has been optimized. Compare sdc and netlist, make set correctly

TMR-8103	Unable to match the source pin of the generate clock corresponding to the master clock	The name of the specified start pin may not be the same as the actual netlist or has been optimized. Compare sdc and netlist, make set correctly
TMR-8104 Clock period value overflow		Check the sdc file to see if the specified values are entered correctly
TMR-8105 Clock edge value overflow		Check the sdc file to see if the specified values are entered correctly
TMR-8106 Clock period value overflow after phase shift		Check the sdc file to see if the specified values are entered correctly
TMR-9001 Error in initializing clock constraint		Check the clock constraints of the sdc file
TMR-9002	Could not match specified in set_input_delay constraint reference clock	Check the netlist and sdc file to ensure that the specified reference clock exists and the name is correct
TMR-9003	Could not match set_output_delay constraints specified in the reference clock	Check the netlist and sdc file to ensure that the specified reference clock exists and the name is correct
(BIT) Bitgen warning/error message		
BIT-5701	bin file is not specified when using MCU	When using MCU, you need to assign the bin file to a specific RAM to initialize
BIT-5702	Bitgen set option name wrong delete btc file and reset bitgen property	
BIT-5703	Bitgen set option value wrong delete btc file and reset bitgen property	
BIT-5705	BRAM 9k initialization data length error check the specified initialization file	
BIT-5706	BRAM 32k initialization data length error check specified initialization file	
BIT-5710	BRAM 128k initialization data length error check the specified initialization file	
BIT-5711	BRAM 256k initialization data length error check specified initialization file	
BIT-5712	Corresponding BRAM not found when updating BRAM data information	Check if bid file matches
BIT-8203	The parameter setting of the unit is wrong	Check the generated IP unit properties
BIT-8401	Bitgen settings file is missing	Check and resave the bitgen property page
BIT-8404	Failed to create bit file	Check if the disk space is full or if you have write permission
BIT-8406	Failed to create svf file	Check if the disk space is full or if you have write permission
BIT-8407	Failed to create bitstream mask file	Check if the disk space is full or if you have write permission
BIT-8409	Cannot open specified RAM file	Make sure the file path is correct
BIT-8410	The specified RAM file exceeds the BRAM 128k capacity quantity	Make sure the file size is appropriate
BIT-8411	The specified RAM file exceeds the BRAM 256k capacity quantity	Make sure the file size is appropriate
BIT-8412	Failed to create svf file for spi download check if disk space is full or have write permission	
(PRG) Program warning/error message		
PRG-5006	View Generate Option in Properties settings Whether the gen_mask option is On	gen_mask is set to on, regenerate the bit file
PRG-8509	The chip encryption key is locked and can no longer be repaired change	

PRG-8511 Key format is incorrect		Confirm the key file format, whether it is a 32-bit hexadecimal number composition
PRG-9002 The flash only supports 64k erasing		flash needs to support 4k erasing
PRG-9004 Failed to read device information from bit file		Whether the format of the bit file is correct, and whether the package class is specified type
PRG-9009 Error reading device information from bit file		Whether the format of the bit file that generates svf is correct, and whether it is specified package type
PRG-9023	The package corresponding to the two bit files of dualboot is not consistent	2 bit files are required for dualboot operation The package is consistent
PRG-9500 Failed to read data		Check the chip and cable for problems
PRG-9501 Failed to write data		Check the chip and cable for problems
PRG-9505 cable failed to successfully identify data		cable reconnect
PRG-9507	write error, read data and reference data are not	Try slowing down and reprogramming
PRG-9508	Consistent data write error, read data and reference data are not Consistent (stm32 cable)	Try slowing down and reprogramming
PRG-9510 The chip failed to be recognized successfully. Check whether there is any problem with the chip connection		
PRG-9511 Failed to open device	Check chip connection for problems	
PRG-9516 Flash ID read failed, try to slow down and reprogram		
PRG-9519 Flash data write error try to slow down and reprogram		
PRG-9520 cannot perform program operation on the chip at the same time, wait for the current program to end before performing the operation		
PRG-9525 Chip identification failed	Check whether there is any problem with the chip connection	

(KIT) Tools/chipwatcher warning/error message

KIT-5003	Specifies that the length of the dat file written to BRAM 9k is not symbol	Check dat file content
KIT-5004	Specifies that the length of the dat file written to BRAM 32k is not symbol	Check dat file content
KIT-5601	Clock net specified in ChipWatcher not found	Check circuit netlist and update cwc file
KIT-5602	Data bus not found in ChipWatcher	Check circuit netlist and update cwc file
KIT-5603	Data net not found in ChipWatcher	Check circuit netlist and update cwc file
KIT-5609	An error occurred while importing cwc files	cwc file does not match current circuit netlist, regenerate cwc file
KIT-5610	Replaces the specified clock net with an equivalent signal	It is recommended to directly specify the equivalent signal as the clock net
KIT-8001	Failed to obtain BRAM unit information in AL chip, confirm chip connection	and power-on status
KIT-8002	Failed to obtain BRAM unit information in EG chip, confirm chip connection	and power-on status
KIT-8003	Failed to obtain BRAM unit information in EF2 chip, confirm chip connection	and power-on status
KIT-8005	Unable to map all of Logic BRAM Physical BRAM	Check if the bid file matches the bit file
KIT-8008	Unrecognized chip type	BRAM Editor does not currently support this device type
KIT-8011	Failed to create dat file	Check if the disk space is full or if you have write permission

TangDynasty

Software

KIT-8016	Unable to read the specified dat file	manual check dat file path and read permission
KIT-8203	This pin position is already occupied	Select ChipProbe Pin position
KIT-8207	The Probe Pin from this signal already exists, no There is a format	It is recommended not to create repeatedly
KIT-8406	error in the cwc file created repeatedly	Reconfigure ChipWatcher and generate cwc file
KIT-8408	Cannot open cwc file	Check file path and read permission
KIT-8411	Error reading data back	Check chip connection status
KIT-8429	Specified signal does not exist	Check the netlist and regenerate the cwc file
KIT-8430	The specified signal group does not exist	Check the netlist and regenerate the cwc file
KIT-8431	The name of the specified signal group is invalid	Reassign signal group
KIT-8432	Cannot add target signal to new group	Nets that already belong to a signal group cannot be added directly to new signal group
KIT-8433	Duplicate names for newly created signal groups	Rename
KIT-8435	Can't move signal group inside itself	Select target location
KIT-8439	ChipWatcher status register length is incorrect, read data error,	recompile or download
KIT-8440	Failed to read Ucode in chip	Check chip connection status
KIT-8441	The identification code in the chip matches the current ChipWatcher image mismatch	Confirm whether the downloaded bit file matches the cwc file, and whether the download was successful
KIT-8445	The readback data length is insufficient	Error reading data, recompile or download
KIT-8446	cwc file check fails the current	Check that the netlist is consistent with the current cwc configuration
KIT-8448	CWC file version is too old to leave the item use	Recompile the original project with the latest version of TD software, and save cwc file

(GUI) GUI warning/error message

GUI-5002	Error when plugging and unplugging USB cable	Please check if the USB cable is in good condition
GUI-5003	Unable to read data from chip	Please check the connection status of the chip
GUI-5004	The signal is not set to input/output pin	Please set the input/output pins
GUI-6002	Unable to open the file The db file	Please check if the file exists
GUI-6003	cannot be generated during flow running piece	Please check whether the read and write permissions of the folder where it is located are allowed
GUI-8002	Cannot open the file Source files are	Please check that the file exists and is in the correct format
GUI-8003	missing, mainly including Verilog, VHDL, Header files, ADC, SDC, CWC files, etc.	Please complete relevant documents
GUI-8004	There is no source file in the project. The	Please add relevant source files
GUI-8005	corresponding file cannot be created during the flow running. file directory, mainly the simulation folder	Please check whether the read and write permissions of the folder where it is located are allowed
GUI-8006	The file is not a complete xml file	Please check whether the file has been manipulated by human or third-party software tampering
GUI-8013	Erase flash failed	Please check the connection status of the chip
GUI-8014	There is an error on a line in the file	Please check that the file is in the correct format
GUI-8015	The project file is empty	Please create a project

TangDynasty

Software

GUI-8016	There is an error in a certain line and a certain column of the file	Please check that the file is in the correct format
GUI-8017	Failed to check a chip	Please check if the chip is connected
GUI-8102	The device cannot be found	Please verify that the device is connected and that the USB cable is intact
GUI-8103	Unable to load circuit database	Please verify that the circuit database exists and that the circuit Is the road database corrupted?
GUI-8104	Failed to load CWC file	Please verify that the CWC file exists and the format is correct
GUI-8201	Unable to find hardware device	Please verify that the device is connected and that the USB cable is intact
GUI-8202	Cannot read data from the chip, please check the connection status of the chip	
GUI-8203	Please select an instance first	
GUI-8204	The content of the bid file is empty, please verify the content of the bid file	
GUI-8205	The content of the bid file matches the hardware equipment department, please verify the content of the bid file	
GUI-8206	The bid file is malformed	Please verify that the bid file is in the correct format
GUI-8211	Unable to get uicode from this device	Please check the device connection status
GUI-8301	Error reading this SDC file	Please check that the file exists and is in the correct format
GUI-8302	Work model is empty	Please check if the project file is normal
GUI-8303	An error occurred while restoring the last running state. Please recompile the project	
GUI-8304	The bit file is missing during the compilation of the chip problem defeat	Please check whether the read and write permissions of the folder where it is located are allowed
GUI-8306	Failed to compile chip watcher	Please check the cwc file for errors
GUI-8350	Failed to convert the project	Please check the pin and Ann of the device of the third-party project Whether the corresponding device matches
GUI-8380	No project file specified	Please specify project file
GUI-8381	Unable to find project file	Please verify that the project file exists
GUI-8383	Unknown tag	Please enter the correct tag
GUI-8384	Failed to parse file	Please check the file for syntax errors
GUI-8401	Syntax error somewhere	Please check the relevant syntax
GUI-8402	There is an error somewhere	Please check the ADC file logic is correct
GUI-8403	A pin in this file cannot be used as a common pin, please set another pin, this pin has other purposes	
GUI-8404	This signal is input/output and cannot be set as a differential pair, please set it as a non-differential pair	
GUI-8406	This pin cannot be occupied by multiple signals, please select another pin	
GUI-8407	This pin cannot be set as a differential pair, please select another pin	
GUI-8408	Illegal pins, please select legal pins	
GUI-8409	Cannot set PreEmphasis property for non-differential pair, please set it to differential pair or remove this property	
GUI-8410	Cannot set Diffr_Dyn property for non-differential pair, please set it to differential pair or remove this property	
GUI-8601	No net information found, please check if the design is correct	
GUI-8701	can't present bit file	Please stop the chip watcher to collect data first, and then enter line download bit
GUI-8702	Failed to download file	Please check the connection status of the device and whether the download mode is correct
GUI-8703	Erase flash failed	Please check the connection status of the chip

TangDynasty

Software manual ,

GUI-8705	The content of the selected file is incorrect	please check whether the content and format of the file are correct
GUI-8706	The content of the file is incorrect	Please check whether the content and format of the file are correct
GUI-8801	project is missing adc file	Please add ADC file for this project
GUI-8901	Failed to create device	Please check the device connection status

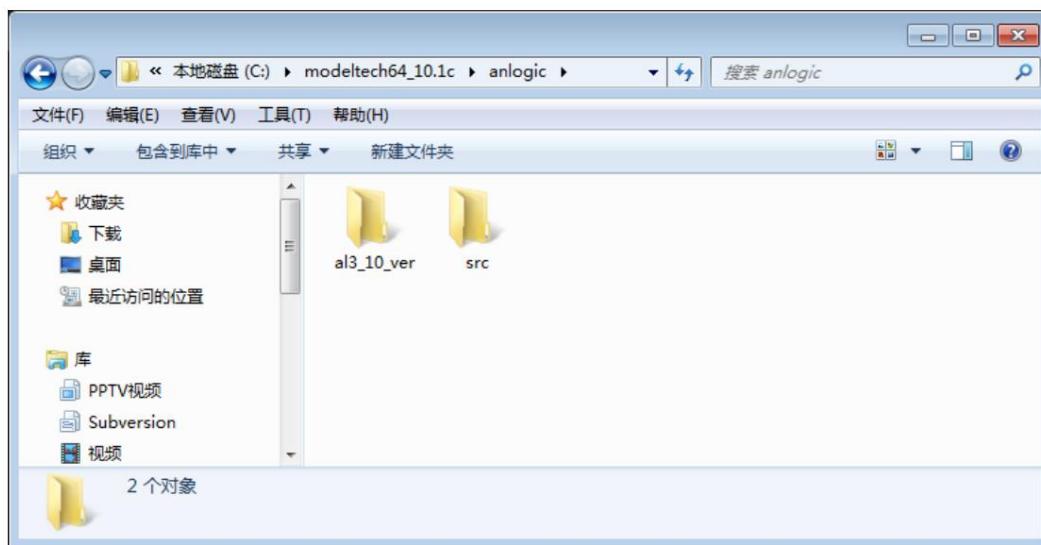
9.4 ModelSim Simulation Process

9.4.1 Add simulation library

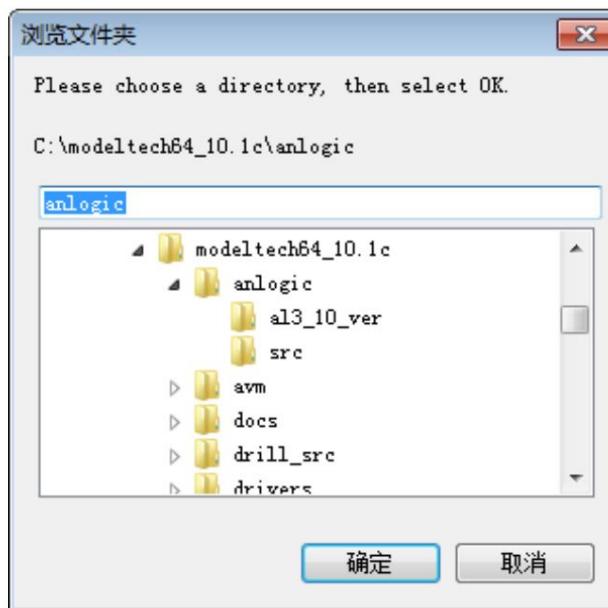
Taking the AL3_10 device as an example, the TD software has its own simulation model and can be compiled in modelsim.

as follows:

1. In the installation directory of modelsim, create a new folder, such as: Anlogic,



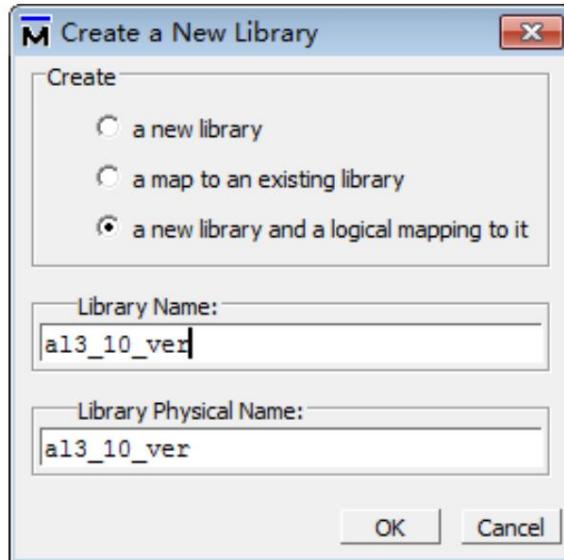
2. Start modelsim, select file → change directory and change the path to the anlogic folder



3. Create a new folder under the anlogic folder, such as src, to store the TD simulation model source file, and store the

Copy all the files in the sim directory under the TD installation path.

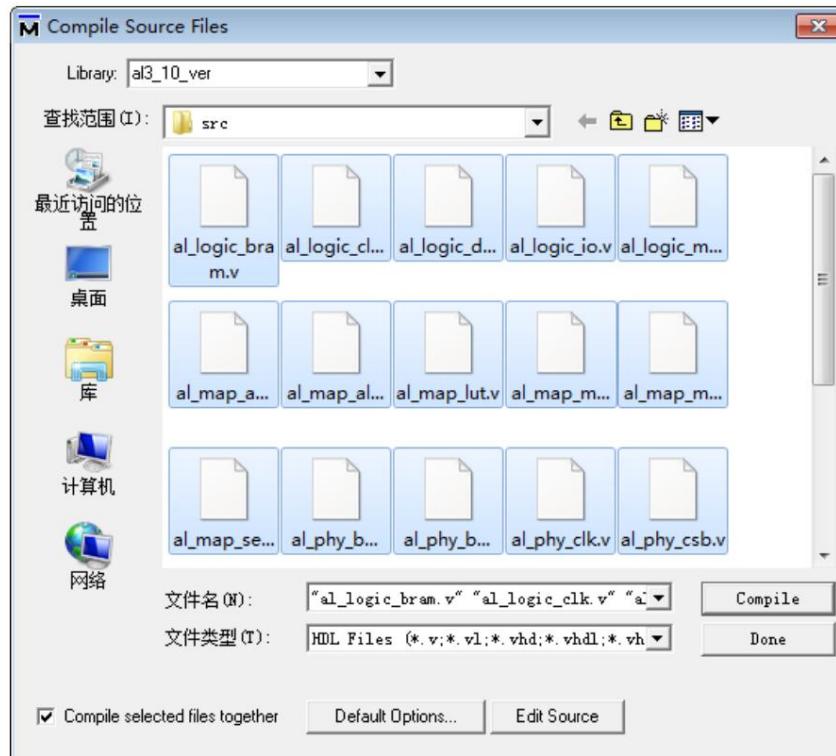
4. Create a new library named al3_10_ver under file → new → library in modelsim



5. Open compile → compile, and the compile source files dialog box will pop up. In the library, select just created.

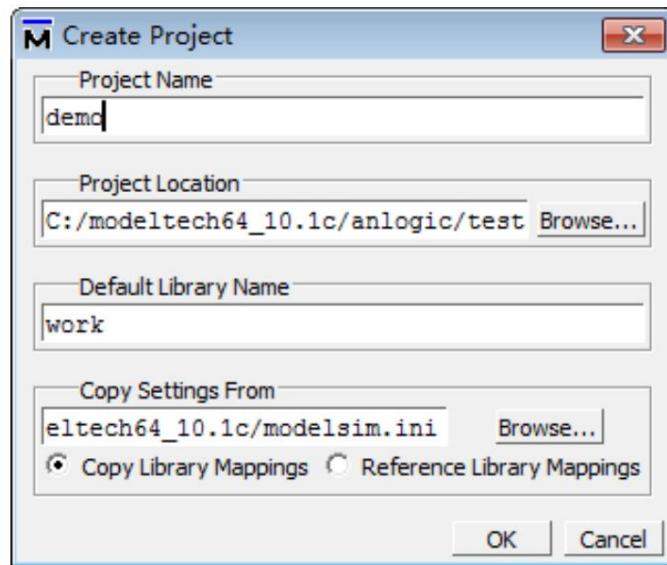
al3_10_ver, select all files under src in the search range, check compile selected files together,

execute compile command

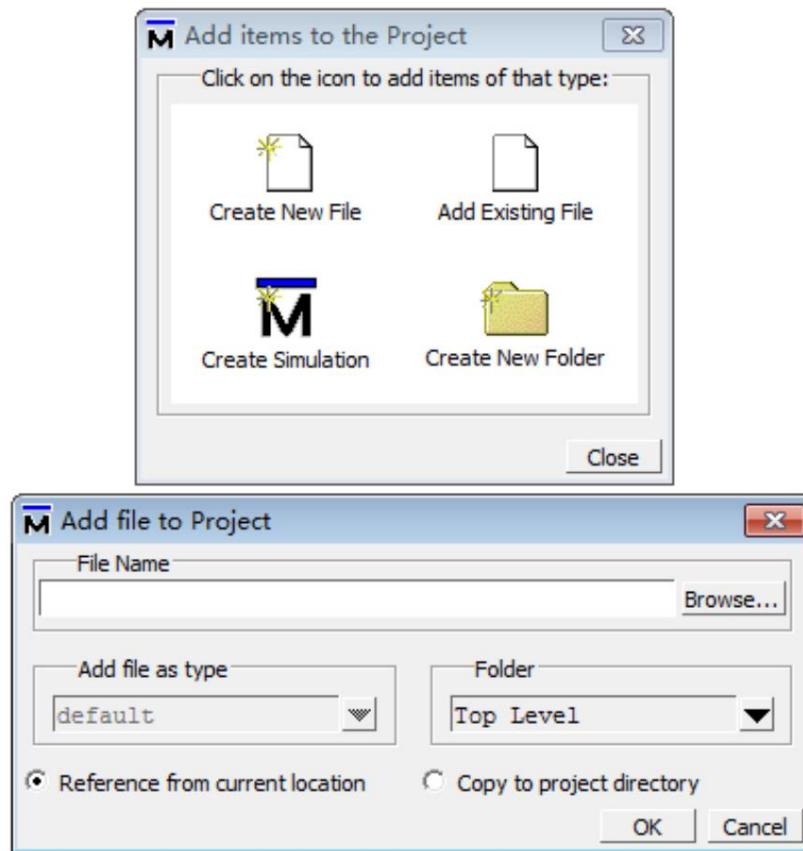


9.4.2 Simulation

1. In modelsim, click file → new → project to create a new project, such as: demo



2. Click add existing file to add a design file, or click Create New File to create a new design file and add it to the project.

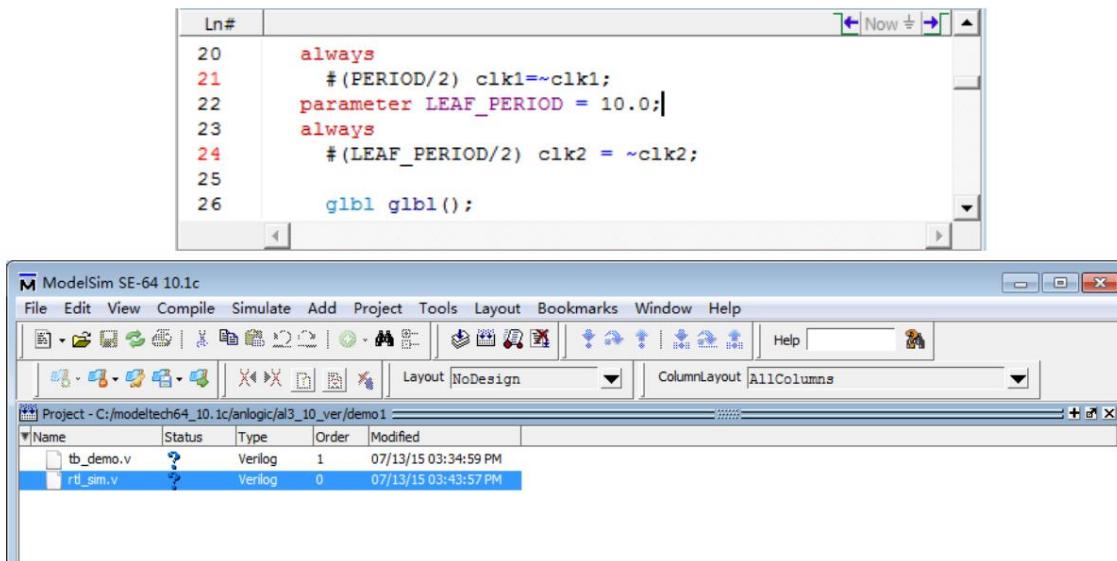


3. Select an existing design source file and its testbench file. Here, the RTL-level electrical program generated by TD is used.

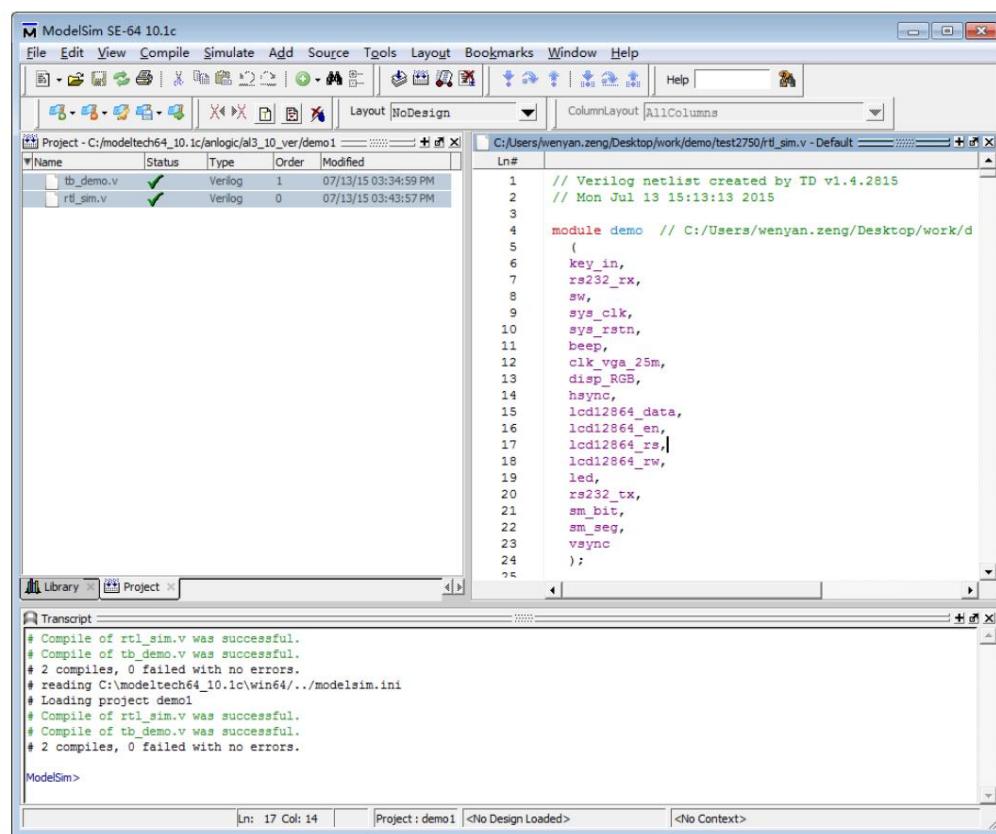
Take the road simulation model `rtl_sim.v` as an example to simulate. If you encounter problems with `glbl` during simulation, please refer to the

The `glbl` module of Anlogic is referenced in the testbench (for PH1, PH1_PHY_GSR needs to be referenced),

As follows:



4. Click to . After the compilation is successful, the status of the source file will change from "?" to " "

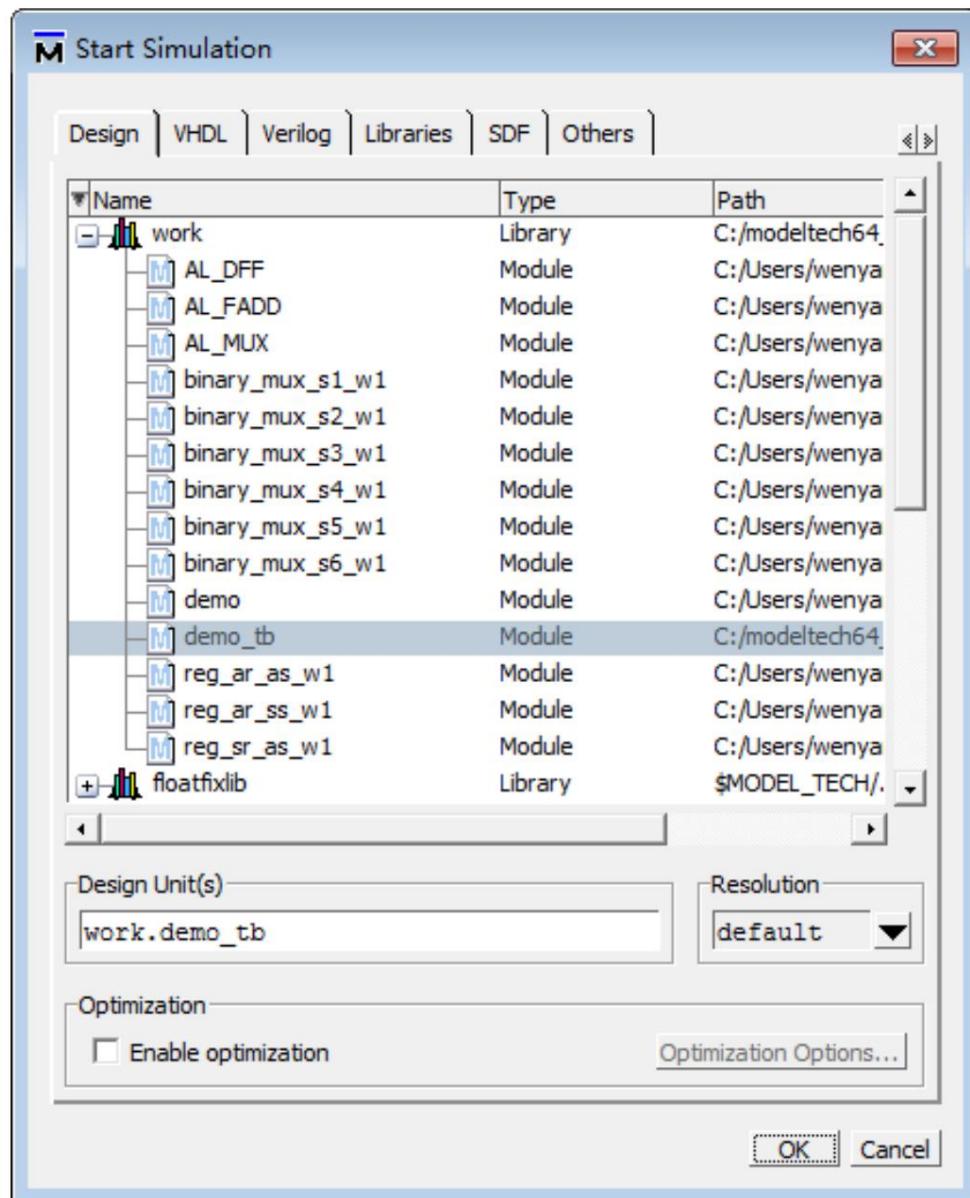


5. Click simulate → start simulate, select the testbench file in the work Library to simulate,

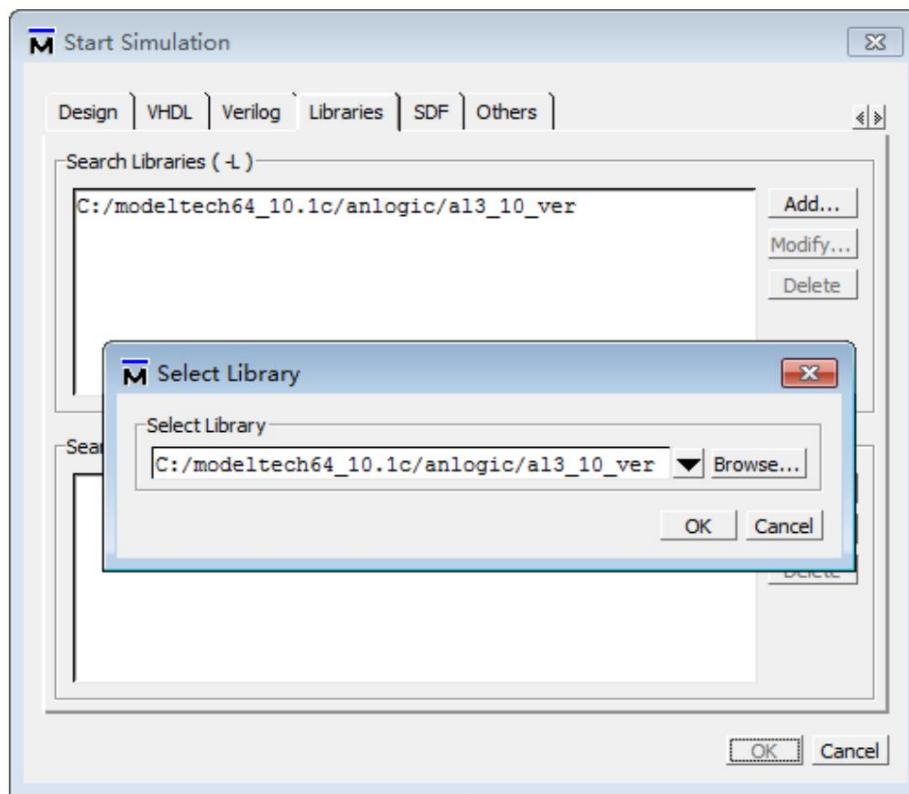
If you want to view the changes of each signal parameter or waveform in the module list after the simulation, you can set the "Enable

Remove the check mark in front of "optimization", otherwise, Modelsim will optimize the signal parameters, resulting in the signal column

Table is empty.

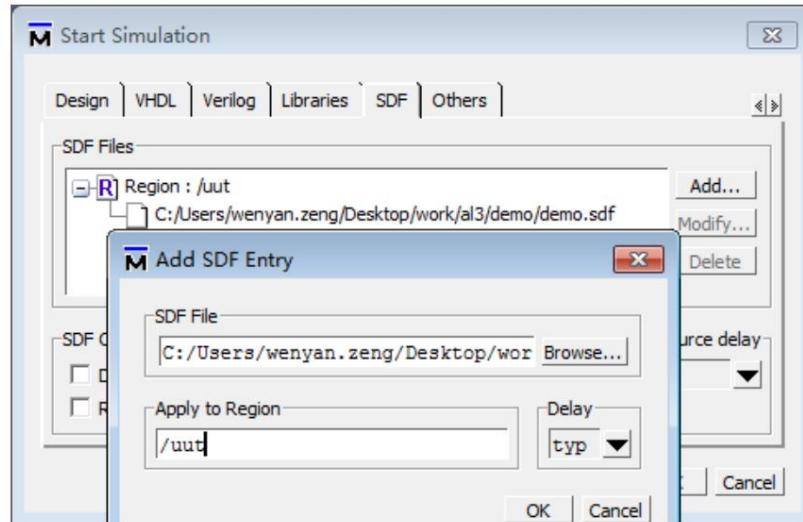


6. Then select libraries and click add, select al3_10_ver, and click OK to simulate.



Note: If it is a timing simulation, you need to specify the sdf file of the project during simulation, where Apply to Region refers to

The instantiated Instance name in the tb file. The netlist file for timing simulation is: prj_name_phy_sim.v



7. After the simulation is over, you can view the list of signals under Objects. You can right-click a signal and select "Add

wave", you can view the waveform changes after running.