Gowin PicoRV32 Software Download

Reference Manual

IPUG913-1.8E, 09/26/2025

Disclaimer

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

Revision History

Date	Version	Description
01/16/2020	1.0E	Initial version published.
03/06/2020	1.1E	● MCU supports GPIO of Wishbone bus interface. ● MCU supports extension AHB bus interface. ● MCU supports off-chip SPI-Flash download and startup. ● MCU supports the read, write and erasure SPI-Flash. ● MCU supports Hardware Stack Protection and Trap Stack Overflow.
06/01/2020	1.2E	● MCU on-line debug function supported. ● MCU core interrupt handler function enhanced. ● MCU core instruction optimized. ● Mergebin tool updated supports GowinSynthesis to parse the rules of naming.
07/16/2021	1.3E	● The synthesis tool SynplifyPro deleted. ● FPGA software version updated. ● Supported devices updated.
02/11/2022	1.4E	The reference design of makehex and mergebin added.
08/18/2023	1.5E	Arora V FPGA products supported.
03/29/2024	1.6E	mergebin tool updated to support GW2AN-18X/9X series of FPGA products.
06/14/2024	1.7E	mergebin tool updated to support Arora V FPGA products.
09/26/2025	1.8E	The version descriptions of Gowin Software and GMD software updated.

Contents

Contents i
List of Figures iii
List of Tables iv
1 Download Methods 1
2 Software Programming Output Used as ITCM Initialization Value 3
2.1 Software Tools 3
2.2 Command Parameters 3
2.3 Hardware Configuration 3
2.4 Software Configuration 4
2.4.1 Boot Mode Configuration 4
2.4.2 Flash Linker Script File Configuration 4
2.4.3 Output File Format Configuration 5
2.5 Design Flow 5
2.6 Devices Supported 6
2.7 Reference Design 6
3 Merge Results of Software Design and Hardware Design 7
3.1 Software Tools 7
3.2 Command Parameters 7
3.3 Hardware Configuration 7
3.3.1 ITCM Configuration 8
3.3.2 Post-Place File Configuration 8
3.4 Software Configuration 9
3.4.1 Boot Mode Configuration 9
3.4.2 Flash Linker Script File Configuration 9
3.4.3 Output File Format Configuration 10
3.5 Design Flow 10
3.5.1 Merge 10
3.5.2 Download 11
3.6 Devices Supported 11
3.7 Reference Design 11

4 Off-chip SPI-Flash Memory Download 12

4.1 Software Configuration 12

4.1.1 Boot Mode Configuration 12

4.1.2 Flash Linker Script File Configuration 13

4.2 Hardware Configuration 15

4.2.1 ITCM Configuration 15

4.2.2 Dual-Purpose Pin Configuration 16

4.3 Design Flow 16

4.4 Download 17

4.4.1 Download Bitstream Files in Hardware Design 17

4.4.2 Download Software Executable File 19

4.5 Devices Supported 21

List of Figures

Figure 2-1 Boot Mode and ITCM Initial Value Configuration 4
Figure 2-2 Boot Mode Configuration 4
Figure 2-3 Flash Linker Script File Configuration 5
Figure 3-1 ITCM Configuration 8
Figure 3-2 Post-Place File Configuration 9
Figure 3-3 Boot Mode Configuration 9
Figure 3-4 Flash Linker Script File Configuration 10
Figure 3-5 Merge the Outputs of Software Design and Hardware Design 11
Figure 4-1 Boot MODE "BUILD_BURN" Configuration 12
Figure 4-2 Boot MODE "BUILD_XIP" Configuration 13
Figure 4-3 Flash Linker Script File Configuration 14
Figure 4-4 Flash Linker Script File Configuration 15
Figure 4-5 ITCM Configuration 16
Figure 4-6 Dual-Purpose Pin Configuration 16
Figure 4-7 Device configuration for Arora Family 18
Figure 4-8 Device Configuration for Arora V 19
Figure 4-9 Device configuration for Arora Family 20
Figure 4-10 Device configuration for Arora V 21

List of Tables

Table 3-1 Command Parameter 7

1 Download Methods

Gowin_PicoRV32 provides three download methods of hardware design and software programming design:

1. Use executable program file generated by software programming design as the initial value of ITCM, the instruction memory, in hardware design.
   a) Gowin_PicoRV32 software programming design:
      - Define the macro definition of config.h as #define BUILD_MODE BUILD_LOAD
      - Select sections.lds as FLASH linker script file
      - Build to generate software executable file
   b) Use makehex tool to convert the software executable files to ram32.hex.
   c) Configure ITCM, the instruction memory, where IP Core Generator generates Gowin_PicoRV32 design.
      - Select "MCU boot and run in ITCM" in Boot Mode.
      - ITCM Initialiaztion File import ram32.hex file in b) as the initial value of ITCM.
   d) Synthesis, place & route to generate the bitstream files in hardware design including software programming design and hardware design.
   e) Use Programmer to download bitstream files in hardware design.
   f) After each update of the software programming design, rerun a) ~ e).
2. Merge the software executable files generated by software programming design and the bitstream files generated by hardware design.
   a) Generate the Gowin_PicoRV32 bitstream files and Post-Place File in hardware design according to the application requirements in accordance with Method 1. If there is no hardware update requirement, the fixed bitstream files in hardware design will not be

updated.

b) Update Gowin_PicoRV32 software programming design:

- Define the macro definition of config.h as #define BUILD_MODE BUILD_LOAD
- Configure sections.lds as FLASH linker script file
- Update user application design according to the application requirements
- Build to generate software executable files

c) Use merge-bit tool to merge the software executable files .bin and bitstream files in hardware design in a).

d) Generate new bitstream files in hardware design after merging the software design and the hardware design.

e) Use Programmer to download the new bitstream files in hardware design after merging.

f) After each update of the software programming design, rerun b) ~ e).

3. Use off-chip SPI-Flash memory to download the executable files generated by software programming design.

a) Configure ITCM, the instruction memory, where IP Core Generator generates Gowin_PicoRV32 design.

- Select "MCU boot and run in external Flash" or "MCU boot from external Flash and run in ITCM" in Boot Mode

b) Gowin_PicoRV32 hardware design generates bitstream files in hardware design with the function of off-chip SPI-Flash downloading and startup.

c) Use Programmer to download bitstream files in hardware design.

d) Gowin_PicoRV32 software programming design:

- MCU boot and run in external Flash
  Define the macro definition of config.h as #define BUILD_MODE BUILD_XIP
  Select sections_xip.lds as FLASH linker script file
- MCU boot from external Flash and run in ITCM
  Define the macro definition of config.h as #define BUILD_MODE BUILD_BURN
  Select sections.lds as FLASH linker script file
- Build to generate software executable files.

e) Use Programmer to download the executable files generated by software programming design.

2 Software Programming Output Used as ITCM Initialization Value

2.1 Software Tools

…\tool\makehex\bin\makehex32.exe

…\tool\makehex\bin\makehex08.exe

Access the above software tools through the following link:

2.2 Command Parameters

Software tool command and parameter: make_hex.exe bin-file

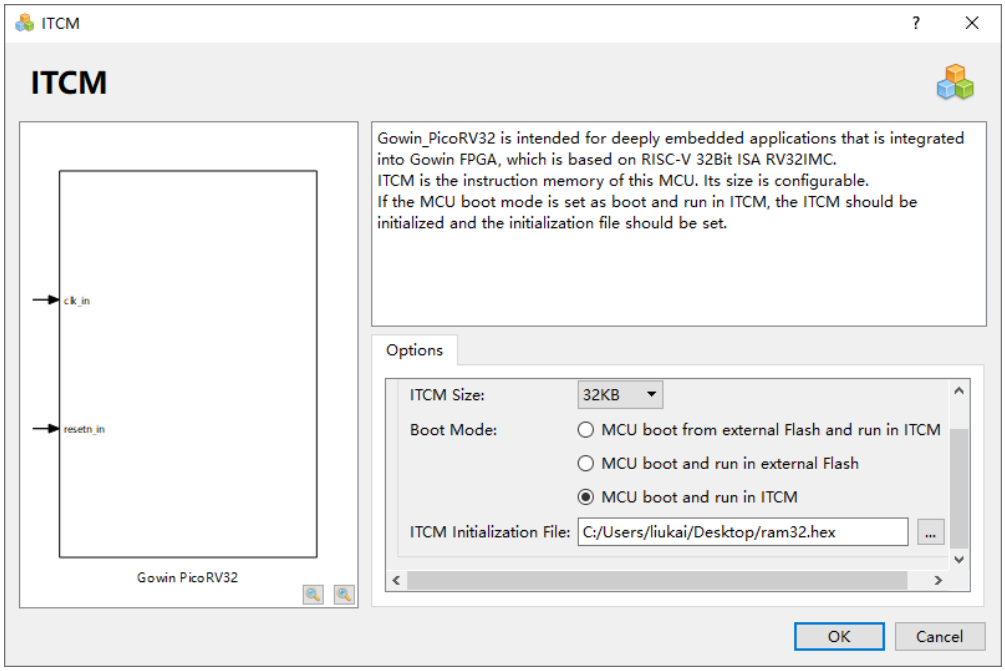
2.3 Hardware Configuration

Double click to open ITCM configuration options when configuring Gowin_PicoRV32 in IP Core Generator tool integrated in Gowin Software:

Select "ITCM > Boot Mode > MCU boot and run in ITCM" option.

Import ram32.hex file generated by make_hex tool as the initial value of ITCM in "ITCM Initialization File", as shown in Figure 2-1.

Figure 2-1 Boot Mode and ITCM Initial Value Configuration



2.4 Software Configuration

2.4.1 Boot Mode Configuration

Define the macro definition of config.h as #define BUILD_MODE BUILD_LOAD, as shown in Figure 2-2.

Figure 2-2 Boot Mode Configuration

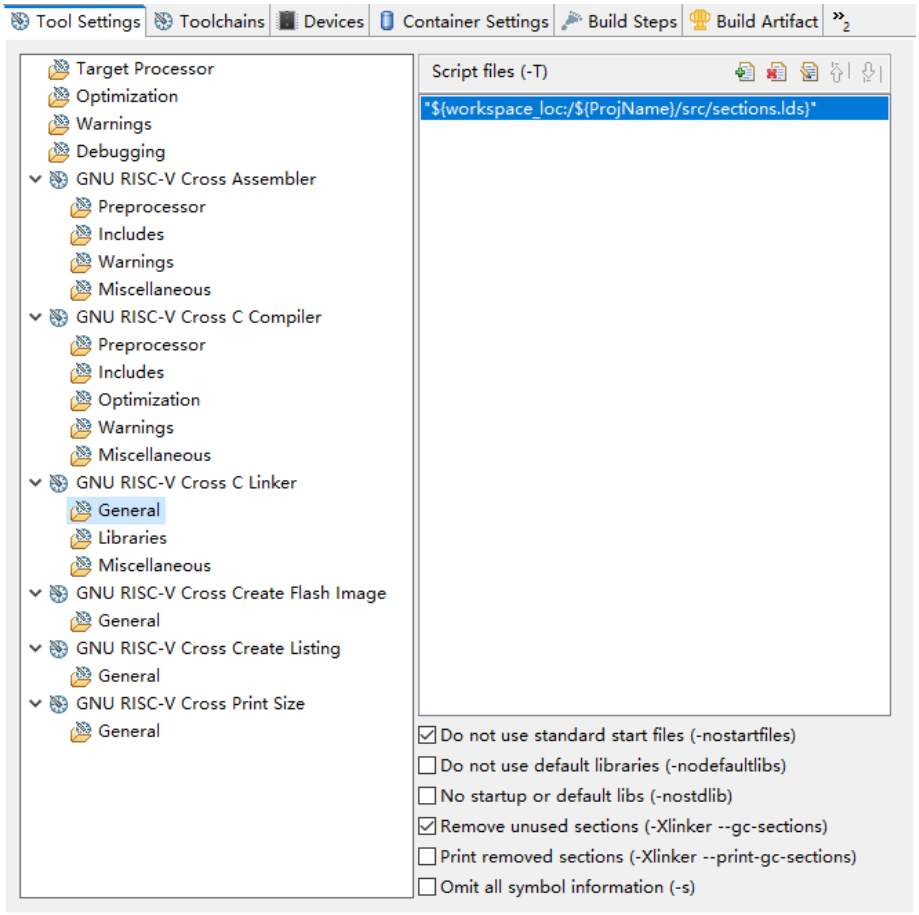
config.h

```
/*
 ******************************************************************************
 * @file        config.h
 * @author      GowinSemicoductor
 * @device      Gowin_PicoRV32
 * @brief       Configurations.
 ******************************************************************************
 */

#ifndef CONFIG_H_
#define CONFIG_H_

//User configures MCU boot mode
/*
 * BUILD_LOAD : MCU boot from and run in ITCM
 *              sections.lds
 * BUILD_BURN : MCU boot from external flash and run in ITCM
 *              sections.lds
 * BUILD_XIP  : MCU boot from and run in external flash
 *              sections_xip.lds
 *
 * */
#define BUILD_LOAD 0  // MCU boot from and run in ITCM, must use sections.lds as linker script!
#define BUILD_BURN 1  // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
#define BUILD_XIP  2  // MCU boot from and run in external flash, must use sections_xip.lds as linker script!

#define BUILD_MODE BUILD_LOAD   //must match with hardware IP

/*
 * sections_debug.lds is for debug.
```

2.4.2 Flash Linker Script File Configuration

In the "Project Explorer" view of GMD software, select the current project, right-click and select "Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General", select "sections.lds" as the Flash linker script file, as shown in Figure 2-3.

Figure 2-3 Flash Linker Script File Configuration



2.4.3 Output File Format Configuration

Compile Gowin_PicoRV32 software programming design and generate software executable files.

Run make_hex.exe bin-file command to generate ram32.hex.

2.5 Design Flow

1. Software programming design flow is as follows:
   - The definition of config.h: #define BUILD_MODE BUILD_LOAD
   - Select section.lds as FLASH linker script file
   - Build to generate software executable files
   - Run make_hex.exe to generate ram32.hex as the initial value of ITCM in Gowin_PicoRV32 IP design
2. Hardware design flow is as follows:
   - Select "ITCM > Boot Mode > MCU boot and run in ITCM" option
   - Import ram32.hex file as the initial value of ITCM in "ITCM Initialization File"
3. Generate Gowin_PicoRV32 IP design, instantiate Gowin_PicoRV32 Top Module, and connect user design

4. Add physical and timing Constraints
5. Use GowinSynthesis to synthesize and generate the netlist file
6. Run Place & Route tool to generate the bitstream files containing software programming design
7. Use Programmer to download
8. After each update of the software programming design, rerun 1~7

2.6 Devices Supported

- LittleBee Family FPGA products
- Arora Family FPGA products
- Arora V FPGA products

2.7 Reference Design

You can get the following reference design through the link:

…\tool\make_hex\ref_design\FPGA_RefDesign\gowin_picorv32

…\tool\make_hex\ref_design\MCU_RefDesign\picorv32_load_demo

3 Merge Results of Software Design and Hardware Design

3.1 Software Tools

…\tool\merge_bit\bin\merge_bit.exe

Access the above software tools through the following link:

3.2 Command Parameters

Software tool command parameter: merge_bit.exe bin-file fs-file itcm_size posp-file.

The descriptions of command parameters are as shown in Table 3-1.

Table 3-1 Command Parameter

Parameter	Description
posp-file	Post-Place File
itcm-size	ITCM Size (KB) For example, if ITCM Size is set as 64K Byte, the parameter is 64.
bin-file	Software executable file
fs-file	Bitstream files in hardware design

Merge the software executable files generated by software programming design and the bitstream files in hardware design generated by hardware design.

When merge_bit.bat is in use, you can modify the parameters, such as posp-file, itcm-size, bin-file, and fs-file according to your requirements.

3.3 Hardware Configuration

Generate the Gowin_PicoRV32 IP design according to Method 1 described in Chapter 2 Software Programming Output Used as ITCM Initialization Value, synthesize, place and route to generate the bitstream files and Post-Place File in hardware design. If there is no hardware update requirement, the fixed bitstream files in hardware design will not be

updated.

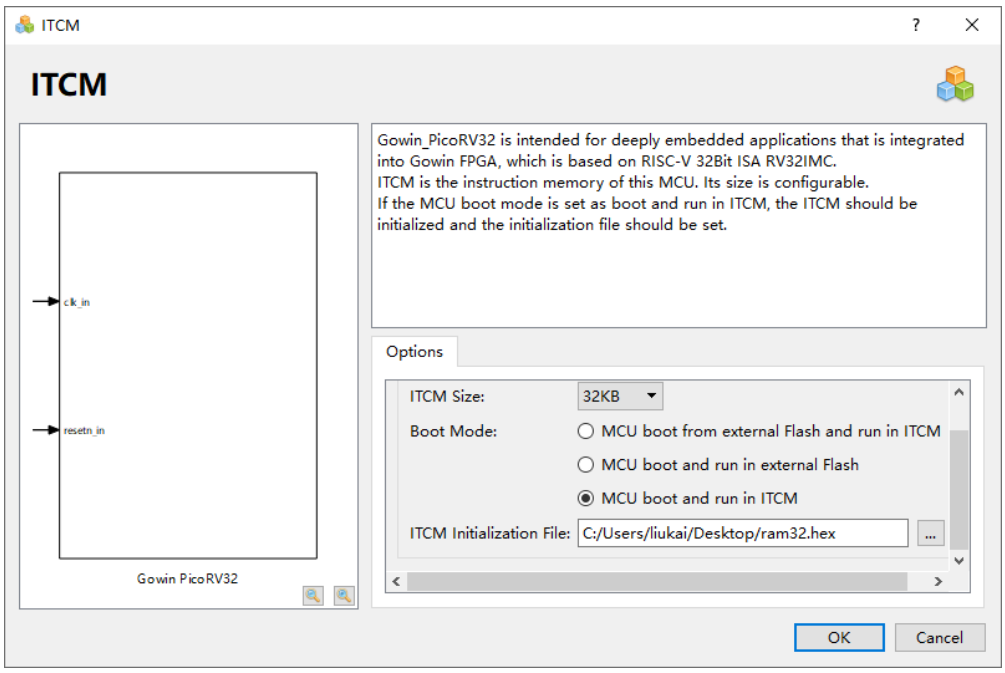
After each update of the software programming design, simply use the merge_bit tool each time to merge the bitstream files in hardware design mentioned above with the software executable files for each update.

3.3.1 ITCM Configuration

Double click to open ITCM configuration options when configuring Gowin_PicoRV32 in IP Core Generator tool integrated in Gowin Software:

- Select "ITCM > Boot Mode > MCU boot and run in ITCM" option
- Import ram32.hex file as the initial value of ITCM in "ITCM Initialization File", as shown in Figure 3-1.

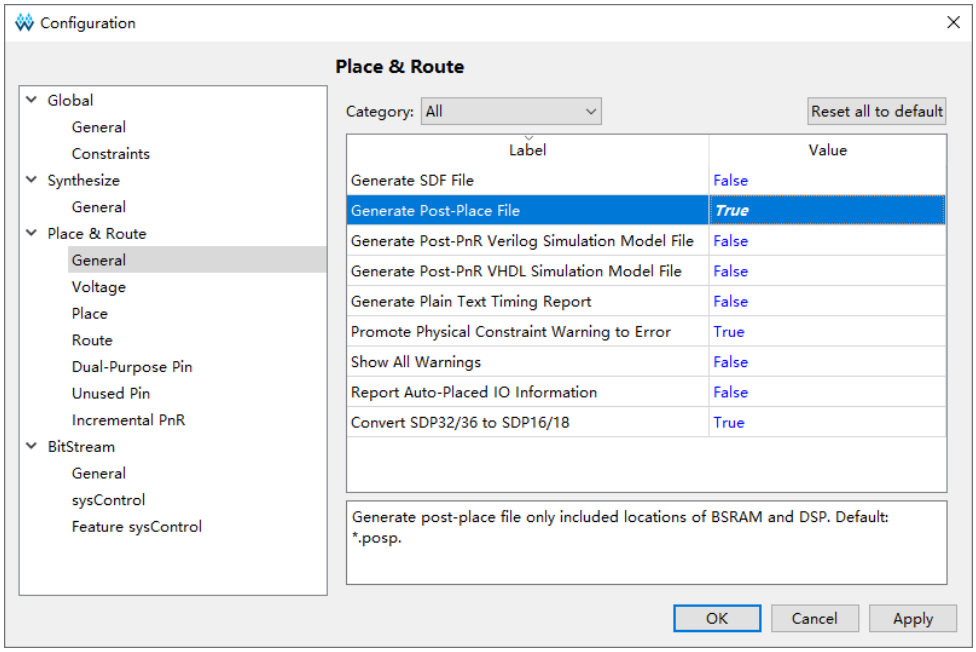
Figure 3-1 ITCM Configuration



3.3.2 Post-Place File Configuration

A Post-Place file, as the posp-file, will be generated when the value in "Place & Route > General > Generate Post-Place File" is set to "True", as shown in Figure 3-2.

Figure 3-2 Post-Place File Configuration

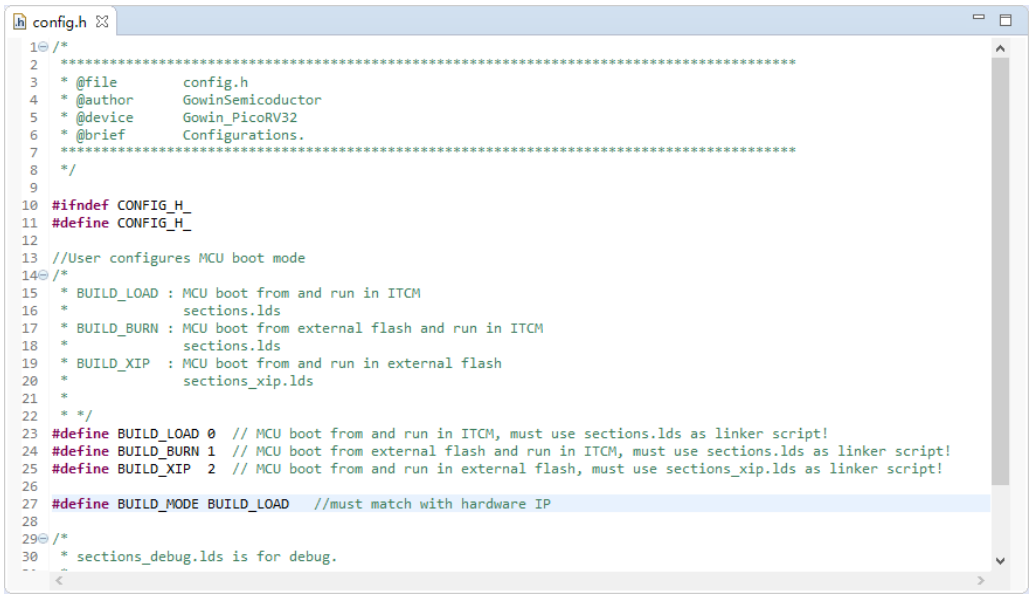


3.4 Software Configuration

3.4.1 Boot Mode Configuration

Configure the macro definition of config.h as #define BUILD_MODE BUILD_LOAD, as shown in Figure 3-3.

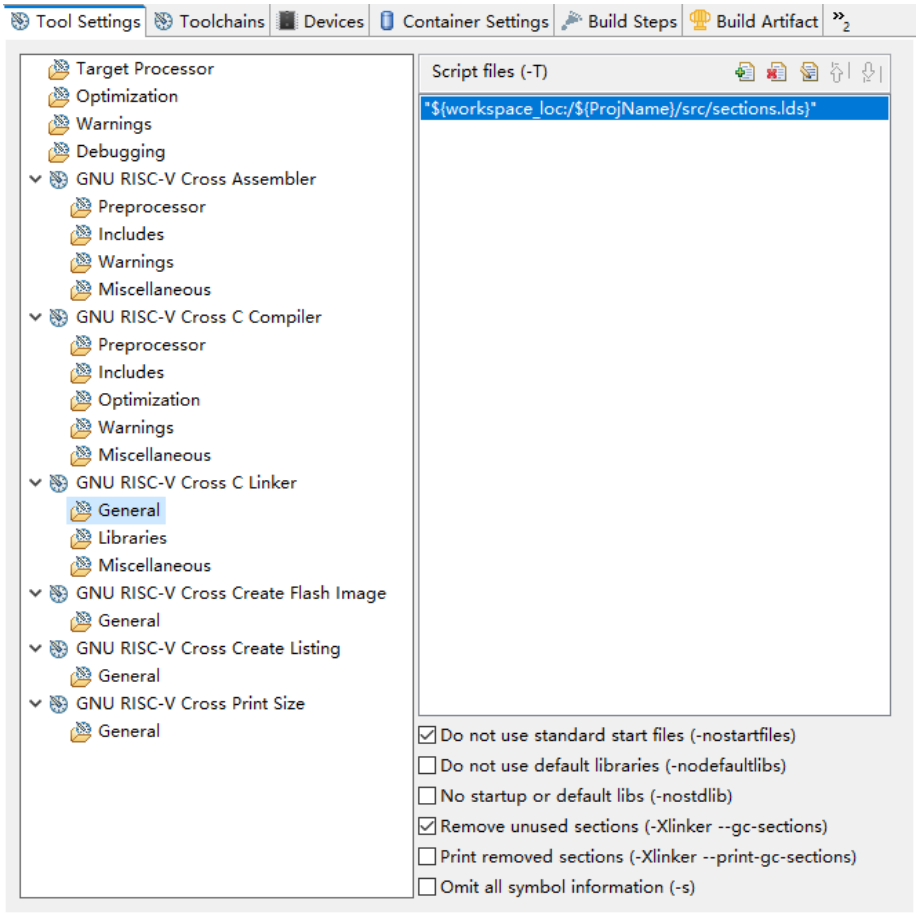
Figure 3-3 Boot Mode Configuration



3.4.2 Flash Linker Script File Configuration

In the "Project Explorer" view of GMD software, select the current project, right-click and select "Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General", select sections.lds as the Flash linker script file, as shown in Figure 3-4.

Figure 3-4 Flash Linker Script File Configuration



3.4.3 Output File Format Configuration

Compile Gowin_PicoRV32 software programming design and generate software executable files.

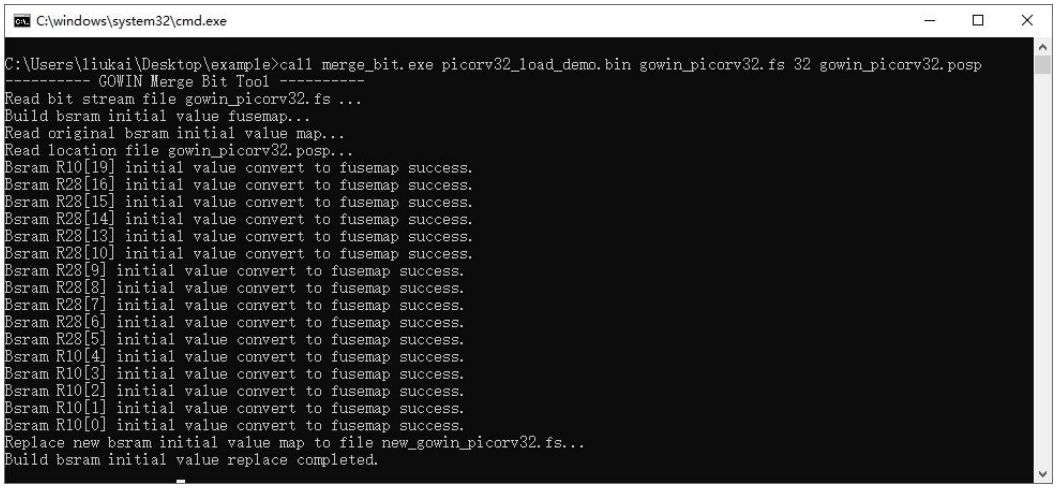
3.5 Design Flow

3.5.1 Merge

1. Generate the Gowin_PicoRV32 bitstream files and Post-Place File in hardware design according to the application requirements in accordance with Method 1 described in Chapter 2 Software Programming Output Used as ITCM Initialization Value. If there is no hardware update requirement, the fixed bitstream files in hardware design will not be updated.
2. Update Gowin_PicoRV32 software programming design:
   - Define the macro definition of config.h as #define BUILD_MODE BUILD_LOAD
   - Select sections.lds as FLASH linker script file
   - Update user application design according to the application requirements
   - Build to generate software executable files

3. Modify merge_bit.bat according to the actual application, and perform merge_bit.bat, merge the bitstream files in hardware design generated by hardware design and the software executable files generated by software programming design to generate new bitstream files, as shown in Figure 3-5;
4. After each update of the software programming design, re-run step 3~4.

Figure 3-5 Merge the Outputs of Software Design and Hardware Design



3.5.2 Download

After merging, use Programmer, the download tool, to download the new bitstream files in hardware design.

For the usage of Gowin Programmer, please see *SUG502, Gowin Programmer User Guide*.

3.6 Devices Supported

- LittleBee Family FPGA products
- Arora Family FPGA products
- Arora V FPGA products

3.7 Reference Design

You can get the following reference design through the link:

…\tool\merge_bit\ref_design\FPGA_RefDesign\gowin_picorv32

…\tool\merge_bit\ref_design\MCU_RefDesign\picorv32_load_demo

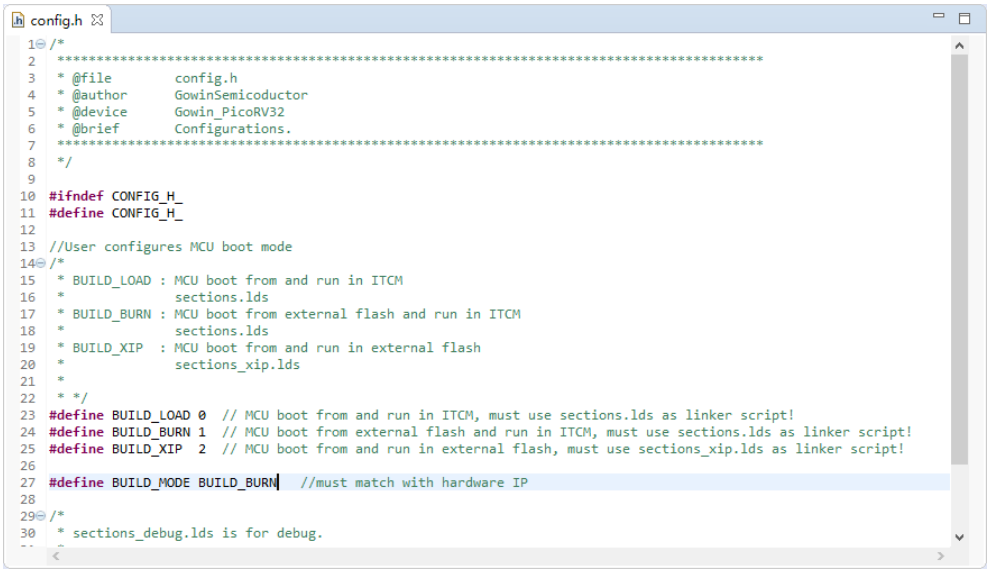
4 Off-chip SPI-Flash Memory Download

4.1 Software Configuration

4.1.1 Boot Mode Configuration

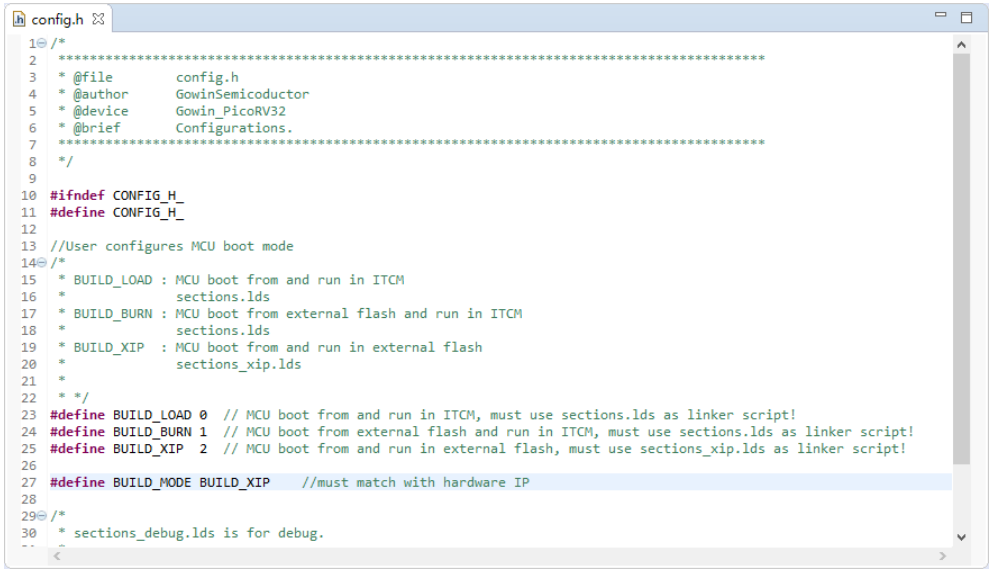
If "ITCM > Boot Mode > MCU boot from external Flash and run in ITCM" is selected in Gowin_PicoRV32 IP design, then config.h is defined as #define BUILD_MODE BUILD_BURN in Gowin_PicoRV32 software programming design config.h, as shown in Figure 4-1.

Figure 4-1 Boot MODE "BUILD_BURN" Configuration



If the "ITCM > Boot Mode > MCU boot and run in external Flash" is selected in Gowin_PicoRV32 IP design, then config.h is defined as #define BUILD_MODE BUILD_XIP in Gowin_PicoRV32 software programming design config.h, as shown in Figure 4-2.

Figure 4-2 Boot MODE "BUILD_XIP" Configuration

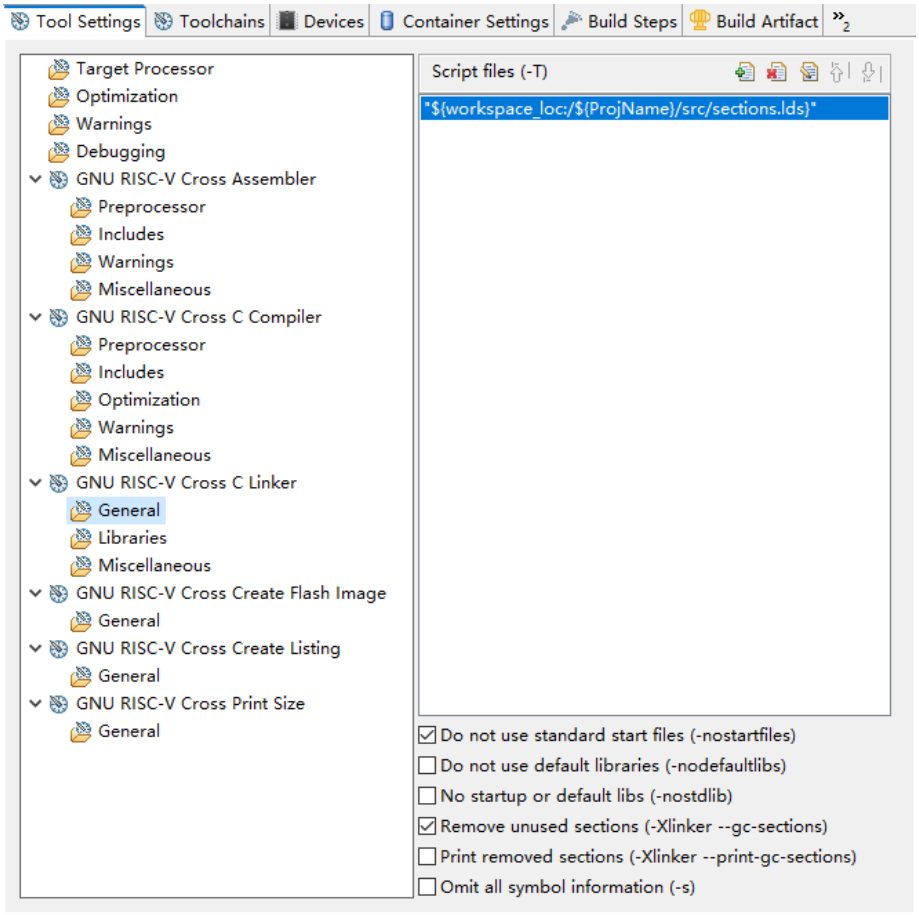


4.1.2 Flash Linker Script File Configuration

If "ITCM > Boot Mode > MCU boot from external Flash and run in ITCM" is selected in the Gowin_PicoRV32 IP design, then the current project is selected in the "Project Explorer" view of the GMD software; right-click and select "Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General", then select "Selections.lds" as the Flash linker script file, as shown in Figure 4-3.

For example, "${workspace_loc:/${ProjName}/src/sections.lds}".

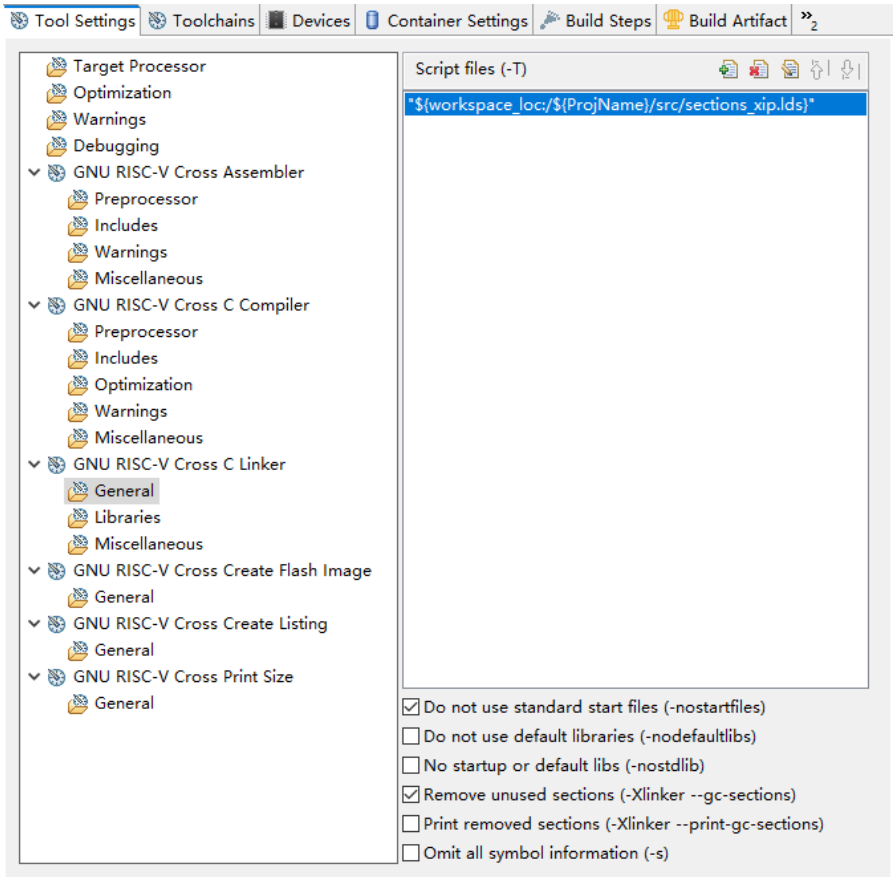
Figure 4-3 Flash Linker Script File Configuration



If "ITCM > Boot Mode > MCU boot and run in external Flash" is selected in the Gowin_PicoRV32 IP design, then the current project is selected in the "Project Explorer" view of the GMD software; right-click and select "Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General", then select "sections_xip.lds" as the Flash linker script file, as shown in Figure 4-4.

For example, "${workspace_loc:/${ProjName}/src/sections_xip.lds}".

Figure 4-4 Flash Linker Script File Configuration

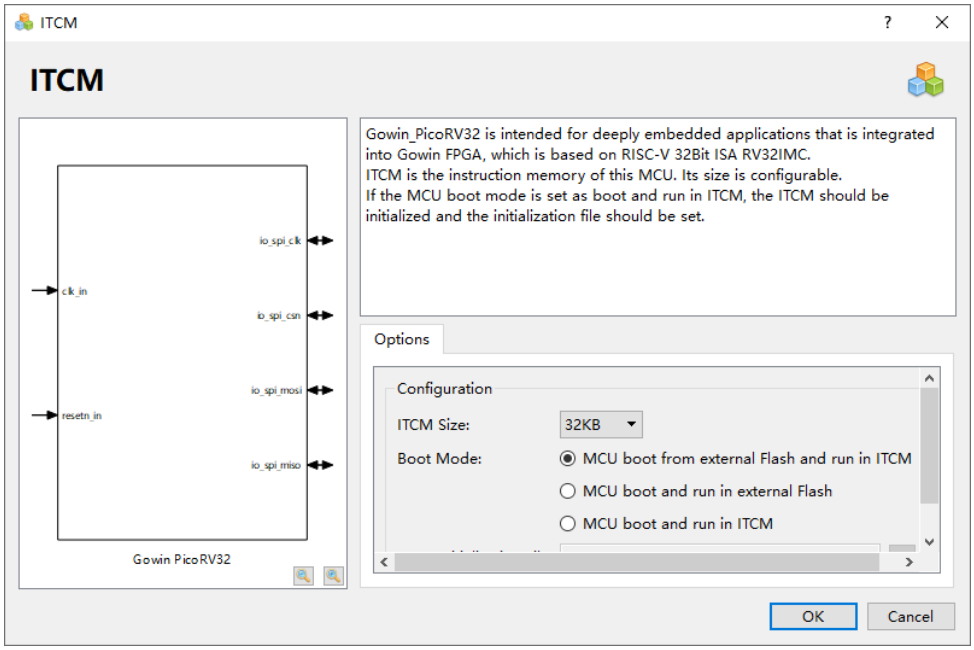


4.2 Hardware Configuration

4.2.1 ITCM Configuration

Select the "ITCM > Boot Mode > MCU Boots from External Flash and Runs in ITCM" or "MCU Boots from External Flash and Runs in External Flash" option in the IP Core Generator tool of Gowin Software, as shown in Figure 4-5.

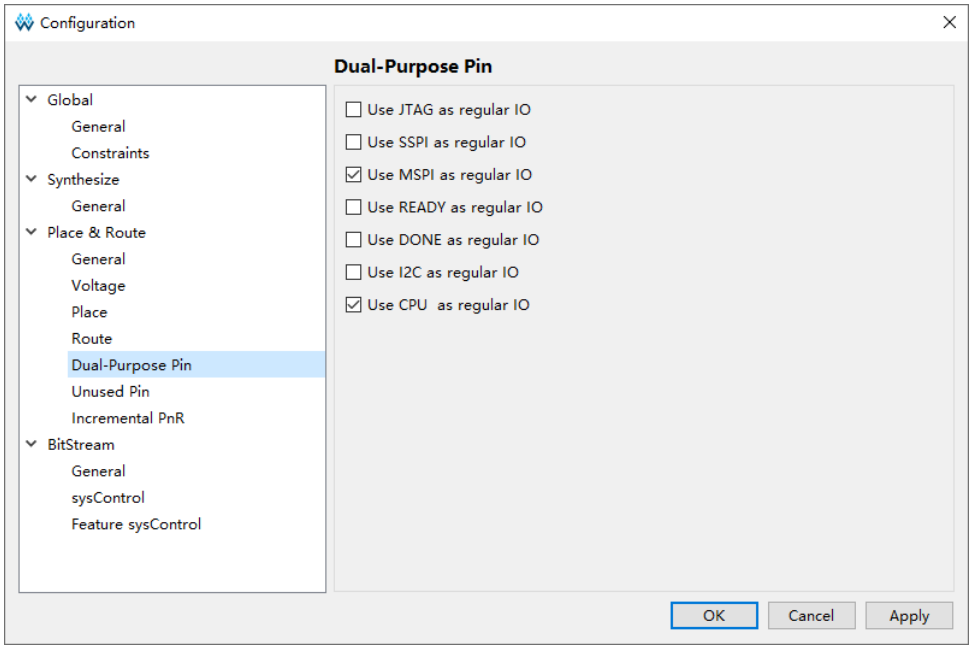
Figure 4-5 ITCM Configuration



4.2.2 Dual-Purpose Pin Configuration

Configure dedicated IO as regular IO under "Place & Route > Dual-Purpose Pin" option, such as MSPI and CPU, as shown in Figure 4-6.

Figure 4-6 Dual-Purpose Pin Configuration



4.3 Design Flow

1. Hardware Design Flow:
   - Select "Boot Mode > MCU boot from external Flash and run in ITCM" or "MCU boot and run in external Flash"
   - Generate Gowin_PicoRV32 IP design

- Synthesis, place & route to generate the bitstream files with off-chip SPI-Flash memory download function

2. Configure Device configuration with Programmer to download the bitstream files in hardware design.
3. Gowin_PicoRV32 software programming design:
   - MCU boot and run in external Flash:

     Define the macro definition of config.h as #define BUILD_MODE BUILD_XIP

     Select sections_xip.lds as FLASH linker script file
   - MCU boot from external Flash and run in ITCM

     Define the macro definition of config.h as #define BUILD_MODE BUILD_BURN

     Select sections.lds as FLASH linker script file
   - Build to generate software programming design BIN files
4. Configure Device configuration with Programmer to download the software executable files.

4.4 Download

For the usage of Gowin Programmer, please see *SUG502, Gowin Programmer User Guide*.

4.4.1 Download Bitstream Files in Hardware Design

Gowin_PicoRV32 hardware design generates bitstream files with the function of off-chip SPI-Flash downloading and startup. Use the download tool Programmer to download the hardware design bitstream file.

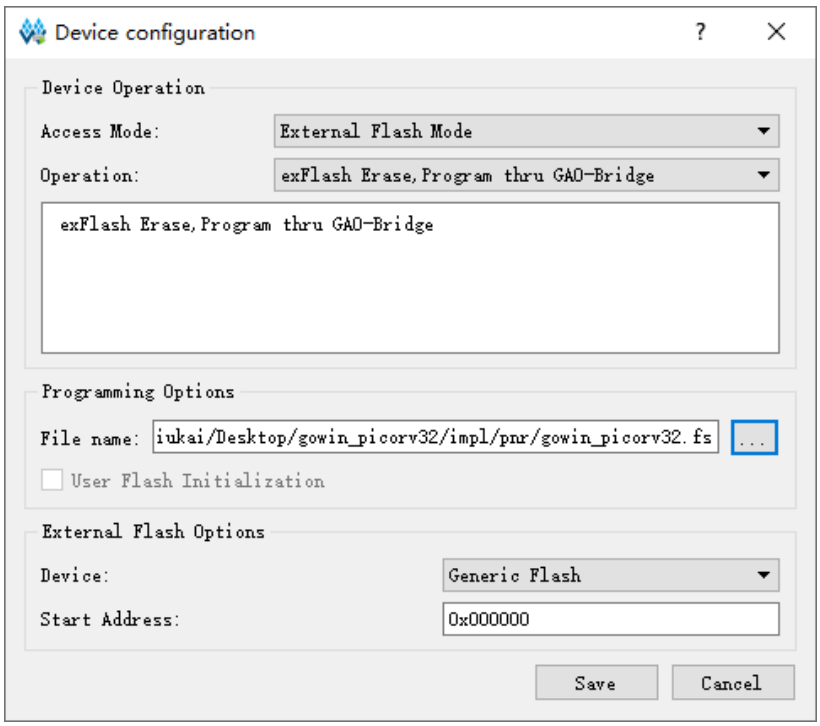
Click "Tools > Programmer" on the menu bar or "Programmer" () on the tool bar to open the "Device configuration".

If the development board has a Arora Family FPGA product on board, the download configuration option is as shown in Figure 4-7.

- Select "External Flash Mode" from the "Access Mode" drop-down list.
- Select "exFlash Erase, Program thru GAO-Bridge" or "exFlash Erase, Program, Verify thru GAO-Bridge" from the "Operation" drop-down list.
- Import the hardware design bitstream file to be downloaded using "Programming Options > File name" option.
- Select "Generic Flash" from "External Flash Options > Device" option.
- Set "0x000000" in "External Flash Options > Start Address" option.

Click "Save" as shown in Figure 4-7.

Figure 4-7 Device configuration for Arora Family

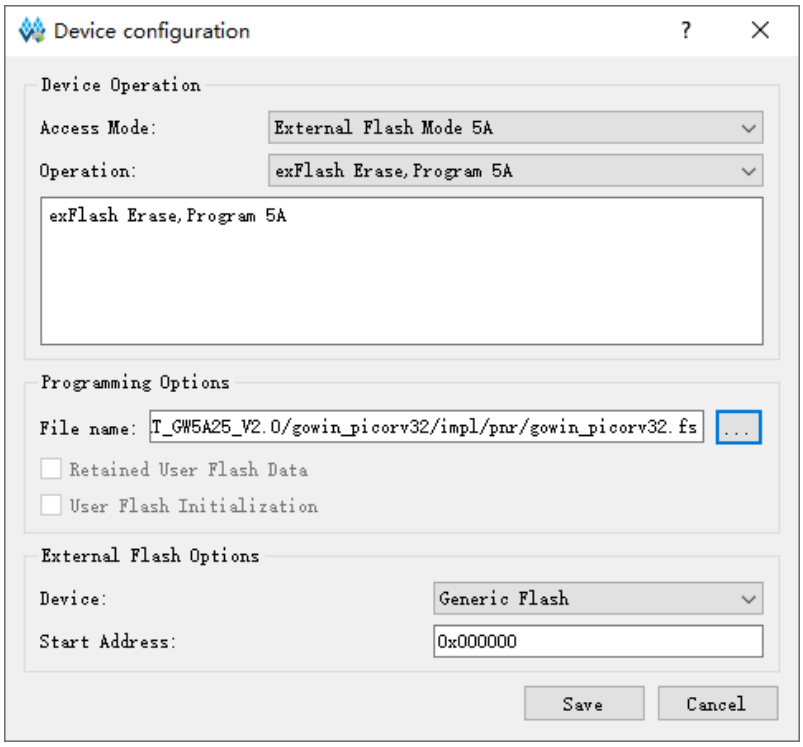


If the development board has a Arora V FPGA product on board, the download configuration option is as shown in Figure 4-8.

- Select "External Flash Mode 5A" in "Access Mode" drop-down list.
- Select "exFlash Erase, Program 5A" or "exFlash Erase, Program, Verify 5A" from the "Operation" drop-down list.
- Import the hardware design bitstream file to be downloaded using "Programming Options > File name" option.
- Select "Generic Flash" from "External Flash Options > Device" option.
- Set "0x000000" in "External Flash Options > Start Address" option.

Click "Save" to complete the configuration, as shown in Figure 4-8.

Figure 4-8 Device Configuration for Arora V



After device configuration, click "Program/Configure" ( ) in the Programmer toolbar to complete the downloading of bitstream files in hardware design.

4.4.2 Download Software Executable File

After Gowin_PicoRV32 software programming design, generate software executable files, and use Programmer to download Gowin_PicoRV32 software executable files.

In GMD software, select "Gowin > Programmer" on the menu bar or "Programmer" ( ) on the tool bar to open Programmer.

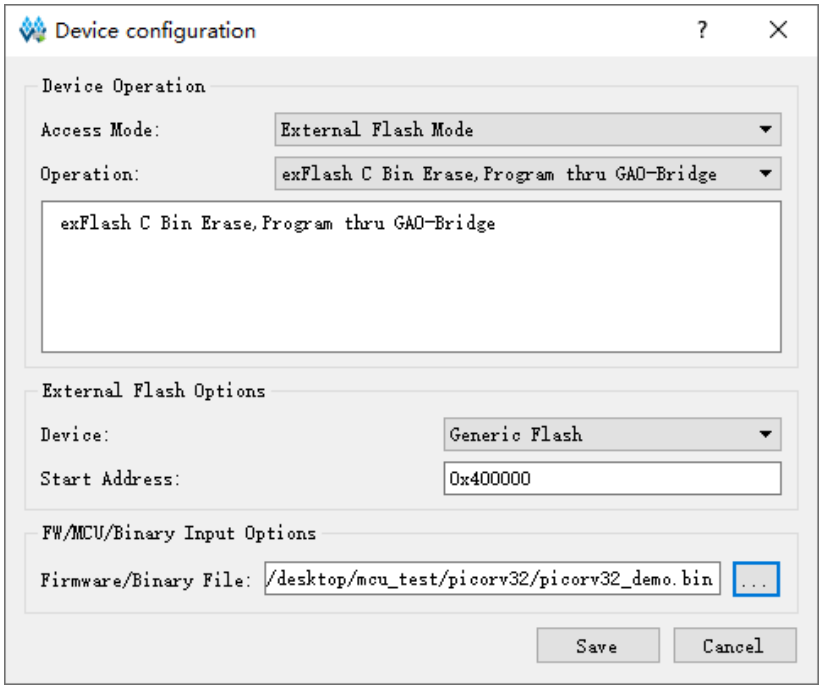
Select "Edit > Configure Device" on the menu bar or "Configure Device" ( ) on the tool bar to open the "Device configuration".

If the development board has a Arora Family FPGA product on board, the download configuration option is as shown in Figure 4-9.

- Select "External Flash Mode" in "Access Mode" drop-down list.
- Select "exFlash C Bin Erase, Program thru GAO-Bridge" or "exFlash C Bin Erase, Program, Verify thru GAO-Bridge" from the "Operation" drop-down list.
- Select "FW/MCU Input Options > Firmware/Binary File" to import the software executable files to be downloaded.
- Select "Generic Flash" from "External Flash Options > Device" option.
- Set "0x400000" in "External Flash Options > Start Address" option.

Click "Save" to complete the configuration, as shown in Figure 4-9.

Figure 4-9 Device configuration for Arora Family

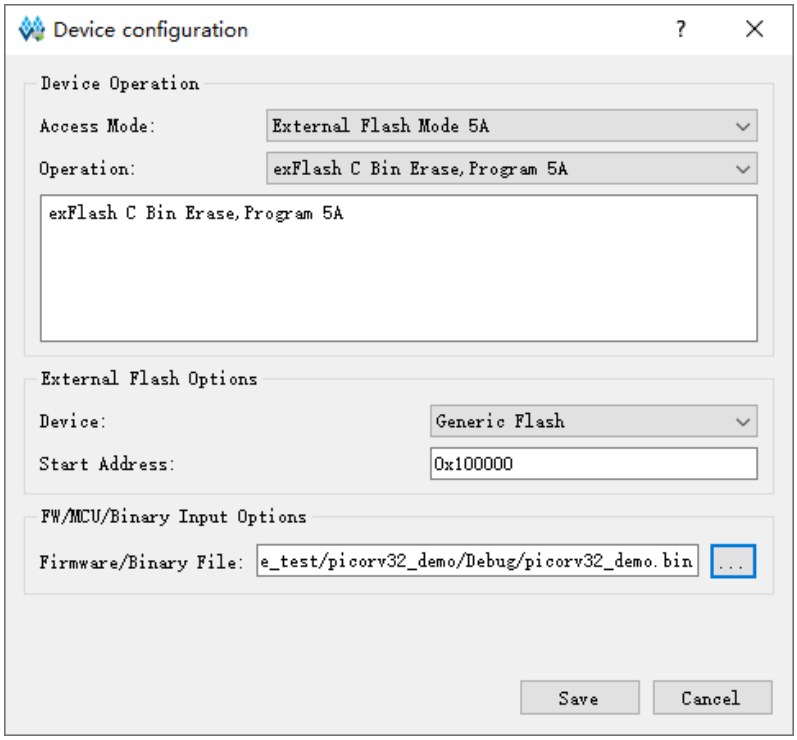


If the development board has a Arora V FPGA product on board, the download configuration option is as shown in Figure 4-10.

- Select "External Flash Mode 5A" in "Access Mode" drop-down list.
- Select "exFlash C Bin Erase, Program 5A" or "exFlash Erase, Program, Verify 5A" in "Operation" drop-down list.
- Import the software executable file to be downloaded using "FW/MCU/Binary Input Options > Firmware/Binary File" option.
- Select "Generic Flash" from "External Flash Options > Device" option.
- In "External Flash Options > Start Address" option, according to different Arora V FPGA products, you can set the address to "0x600000" (138K, 75K products), "0x400000" (60K products), "0x100000" (25K and 15K products).

Click "Save" to complete the configuration, as shown in Figure 4-10.

Figure 4-10 Device configuration for Arora V



After device configuration, click "Program/Configure" ( ) on the Programmer tool bar to complete the downloading of software executable files.

4.5 Devices Supported

- Arora Family FPGA products
- Arora V FPGA products


GOWIN
PROGRAMMING FOR THE FUTURE