




Gowin PicoRV32 Software Download **Reference Manual**

IPUG913-1.3E, 07/16/2021

Copyright © 2021 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

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

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. 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	<ul style="list-style-type: none">● 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	<ul style="list-style-type: none">● 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	<ul style="list-style-type: none">● The synthesis tool, SynplifyPro, deleted;● FPGA software version updated;● Supported devices updated.

Contents

Contents	i
List of Figures	iii
List of Tables	iv
1 Download Methods	1
2 Software Programming Output as ITCM Initialization Value	3
2.1 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 Configuration	4
2.4.3 Output File Format Configuration	4
2.5 Design Flow	5
2.6 Devices Supported	5
3 Merge Outputs of Software Design and Hardware Design	6
3.1 Tools.....	6
3.2 Command Parameters.....	6
3.3 Hardware Configuration.....	7
3.3.1 Boot Mode Configuration	7
3.3.2 Post-Place File Configuration	8
3.4 Software Configuration	8
3.4.1 Boot Mode Configuration	8
3.4.2 Flash linker Configuration	8
3.4.3 Output File Format Configuration	9
3.5 Design Flow	9
3.5.1 Merge.....	9
3.5.2 Download	10
3.6 Devices Supported	10
3.7 Application Software	10
4 Off-chip SPI-Flash Download and Startup	11

4.1 Software Configuration	11
4.1.1 Boot Mode Configuration	11
4.1.2 Flash linker Configuration	12
4.2 Hardware Configuration.....	13
4.2.1 Boot Mode Configuration	13
4.2.2 Dual-Purpose Pin Configuration	14
4.3 Design Flow	14
4.4 Download	15
4.4.1 Download Bitstream Files in Hardware Design	15
4.4.2 Download BIN Files in a Binary Format in Software Design	16
4.5 Devices Supported	17

List of Figures

Figure 2-1 Boot Mode and ITCM Initial Value Configuration	3
Figure 2-2 Boot Mode Configuration.....	4
Figure 2-3 Flash Linker Script Configuration	4
Figure 3-1 Boot Mode and ITCM Initial Value Configuration	7
Figure 3-2 posp Configuration	8
Figure 3-3 Boot Mode Configuration.....	8
Figure 3-4 Flash Linker Configuration	9
Figure 3-5 Merge the Outputs of Software Design and Hardware Design	10
Figure 4-1 Boot Mode Configuration.....	11
Figure 4-2 Boot Mode Configuration.....	12
Figure 4-3 Flash Linker Configuration	12
Figure 4-4 Flash Linker Configuration	13
Figure 4-5 Boot Mode Configuration.....	13
Figure 4-6 Dual-Purpose Pin Configuration.....	14
Figure 4-7 Device configuration	15
Figure 4-8 Device configuration	16

List of Tables

Table 3-1 mergebin Commands and Parameters	6
--	---

1 Download Methods

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

1. Use the mapping file generated by software programming design as the initial value of ITCM, the instruction memory, in hardware design.
 - a) Gowin_PicoRV32 Software Design:
 - Define the macro definition of config.h as `#define BUILD_MODE BUILD_LOAD`
 - Configure sections.lds as FLASH linker
 - Build to generate BIN files in a binary format in software design.
 - b) Use makehex tool to convert the BIN files in a binary format in software design to ram32.hex, the mapping file in a hexadecimal format;
 - c) Configure ITCM, the instruction memory, in the hardware design where IP Core Generator generates Gowin_PicoRV32;
 - Select “MCU boot and run in ITCM” in Boot Mode.
 - ITCM Initialization 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 BIN files in a binary format in software design generated by software programming design and the BIN files in a binary format in hardware design 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.lids as FLASH linker
 - Update user application design according to the application requirements
 - Build to generate BIN files in a binary format in software design.
 - c) Use mergebin tool to merge the BIN files in a binary format in software design 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 to download the BIN files in a binary format generated by software programming design.
- a) Configure ITCM, the instruction memory, in the hardware design where IP Core Generator generates Gowin_PicoRV32.
 - 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`
 - Configure sections_xip.lids as FLASH linker
 - MCU boot from external Flash and run in ITCM
 - Define the macro definition of config.h as `#define BUILD_MODE BUILD_BURN`
 - Configure sections.lids as FLASH linker
 - Build to generate BIN files in a binary format in software design.
 - e) Use Programmer to download the BIN files in a binary format generated by software programming design.

2 Software Programming Outputs as ITCM Initialization Value

2.1 Tools

Windows:

Gowin_PicoRV32\tool\makehex\bin\makehex32.exe

Note!

Access the above software tools via this link:

http://cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

2.2 Command Parameters

Windows: makehex32.exe bin-file

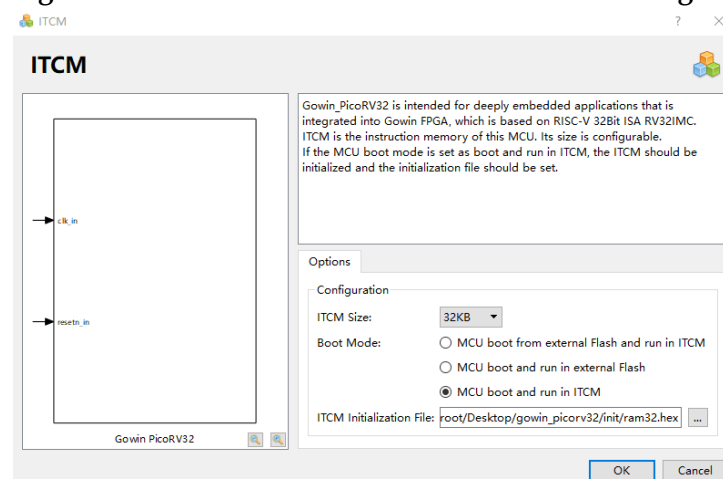
2.3 Hardware Configuration

Double click to open ITCM configuration options when configuring Gowin_PicoRV32 in IP Core Generator:

Select “MCU boot and run in ITCM” in Boot Mode

Import ram32.hex, the mapping file in a hexadecimal format, 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

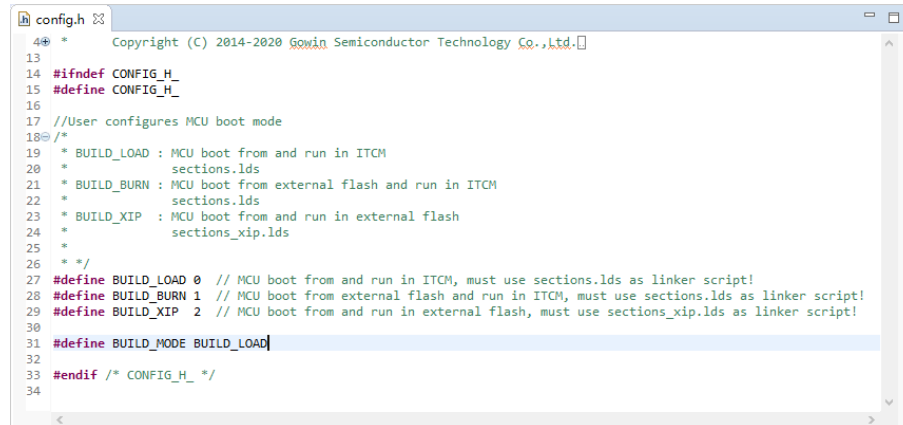


2.4 Software Configuration

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.

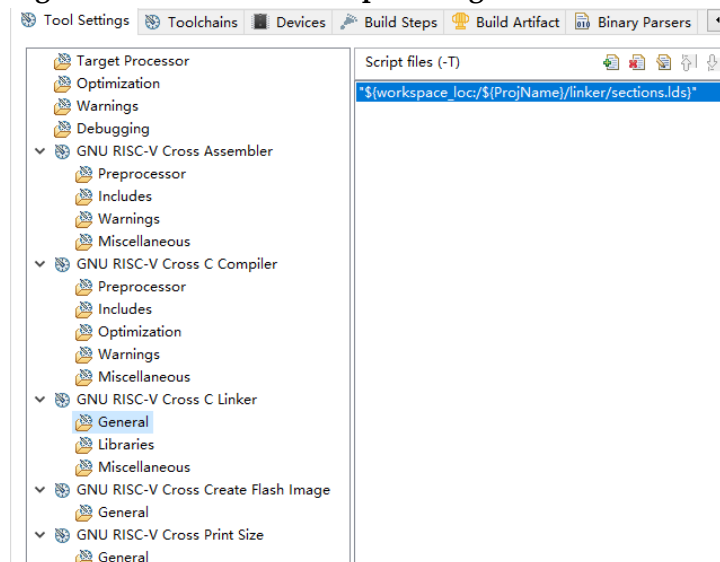
Figure 2-2 Boot Mode Configuration



2.4.2 Flash linker Configuration

In the parameters configuration of GNU RISC-V Cross C Linker in GOWIN MCU Designer (V1.1 and above), select “sections.lds” as Flash linker, as shown in Figure 2-3.

Figure 2-3 Flash Linker Script Configuration



2.4.3 Output File Format Configuration

Compile Gowin_PicoRV32 software programming design by GOWIN MCU Designer (V1.1 and above) and generate BIN file in a binary format in software design.

Run makehex32.exe bin-file to generate ram32.hex, the mapping file in a hexadecimal format.

2.5 Design Flow

1. Gowin_PicoRV32 software programming design:
 - The definition of config.h: #define BUILD_MODE BUILD_LOAD;
 - Configure section.lids as FLASH linker;
 - Build to generate BIN files in a binary format in software design;
 - Run makehex32.exe to generate ram32.hex, the mapping file in a hexadecimal format, as the initial value of ITCM in Gowin_PicoRV32 hardware design.
2. Gowin_PicoRV32 Hardware Design:
 - Configure “MCU boot and run in ITCM” in Boot Mode.
 - Use ram32.hex as the initial value of ITCM.
3. Generate Gowin_PicoRV32 hardware design, instantiate Gowin_PicoRV32 Top Module, and connect user design;
4. Add physical and timing Constraints;
5. Use GowinSynthesis® to synthesis and generate the netlist file;
6. Run Place & Route tool to generate the bitstream files in hardware design;
7. Use Programmer to download;
8. After each update of the software programming design, rerun 1~7.

2.6 Devices Supported

GW1N-9/GW1N-9C/GW1NR-9/GW1NR-9C/GW2AN-9X
GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C/GW2
AN-18X
GW2A-55/ GW2A-55C/ GW2AN-55C

3 Merge Outputs of Software Design and Hardware Design

3.1 Tools

Windows:

Gowin_PicoRV32\tool\mergebin\bin\mergebin.bat

Access the above software tools via this link:

http://cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

3.2 Command Parameters

Windows: mergebin.bat

mergebin software commands and parameters:

call posp_parse.exe posp-file itcm-size gwsyn

call merge_bit.exe bin-file BramLoc.txt fs-file

For the description of commands and parameters, please refer to Table 3-1.

Table 3-1 mergebin Commands and Parameters

Parameter	Description
posp_parse.exe	Parse the posp file generated by place & layout, generate the location information of BRAM in ITCM, and output to the BramLoc.txt file.
posp-file	posp file name, same with the project name, with an extension of posp.
itcm-size	Gowin_PicoRV32 ITCM Size (KB) For example, if set ITCM as 64K Byte, the value is 64.
gwsyn	Specify GowinSynthesis® as the synthesis tool.
merge_bit.exe	Merge Gowin_PicoRV32 software design and hardware design
bin-file	Bin files in a binary format in software design generated by Gowin_PicoRV32 software programming design
BramLoc.txt	ITCM layout location information file generated by posp_parse.exe

Parameter	Description
fs-file	Bitstream files in hardware design generated by Gowin_PicoRV32 hardware design

Merge the BIN files in a binary format in software design generated by software programming design and the bitstream files in hardware design generated by hardware design.

When mergebin.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 hardware design of Gowin_PicoRV32 according to Method 1 in Chapter 2, synthesize, place and layout 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 mergebin tool each time to merge the bitstream files in hardware design mentioned above with the BIN files in a binary format in software design for each update.

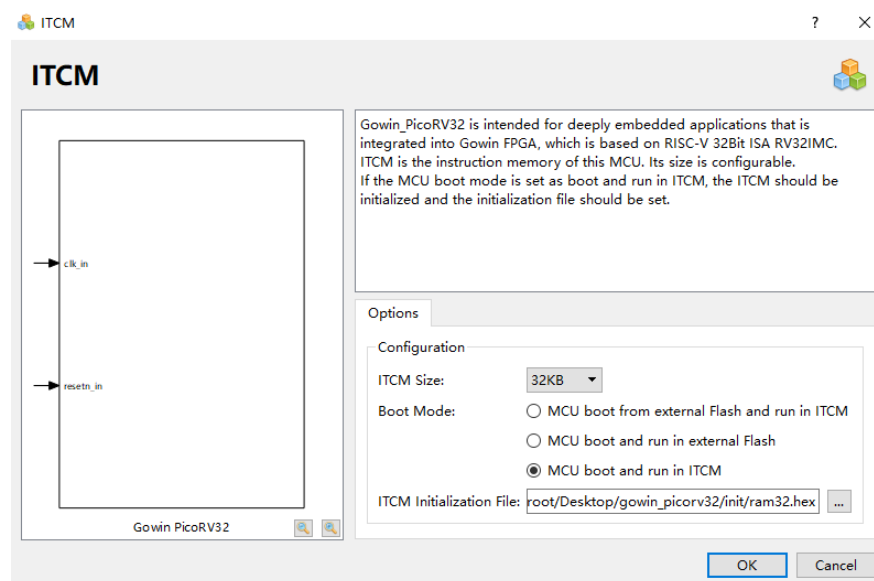
3.3.1 Boot Mode Configuration

Double click to open ITCM configuration options when configuring Gowin_PicoRV32 in IP Core Generator:

Select “MCU boot and run in ITCM” in Boot Mode

Import ram32.hex, the mapping file in a hexadecimal format, as the initial value of ITCM in “ITCM Initialization File”, as shown in Figure 3-1.

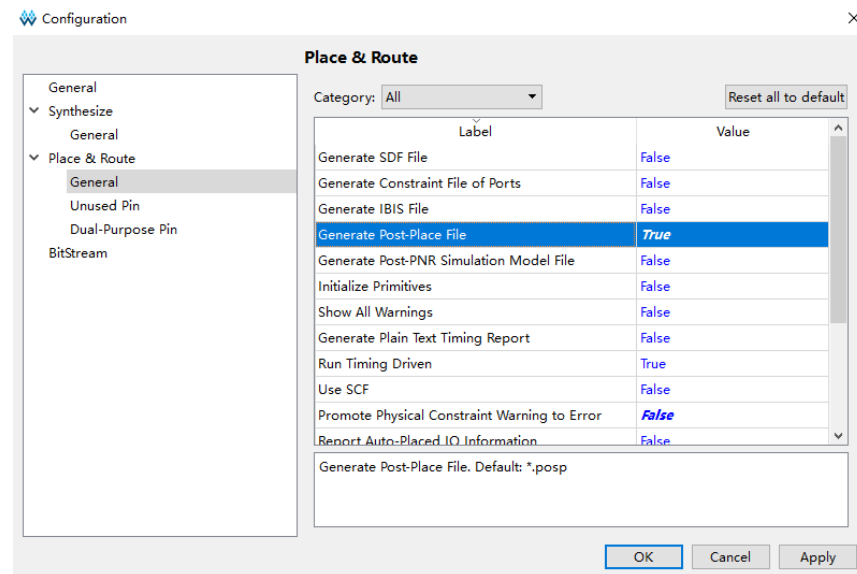
Figure 3-1 Boot Mode and ITCM Initial Value Configuration



3.3.2 Post-Place File Configuration

A posp file, as the posp input file of posp_parse.exe, will be generated when the value of "Generate Post-Place File" under "Place & Route" is True in Gowin_PicoRV32 hardware design, as shown in Figure 3-2.

Figure 3-2 posp Configuration

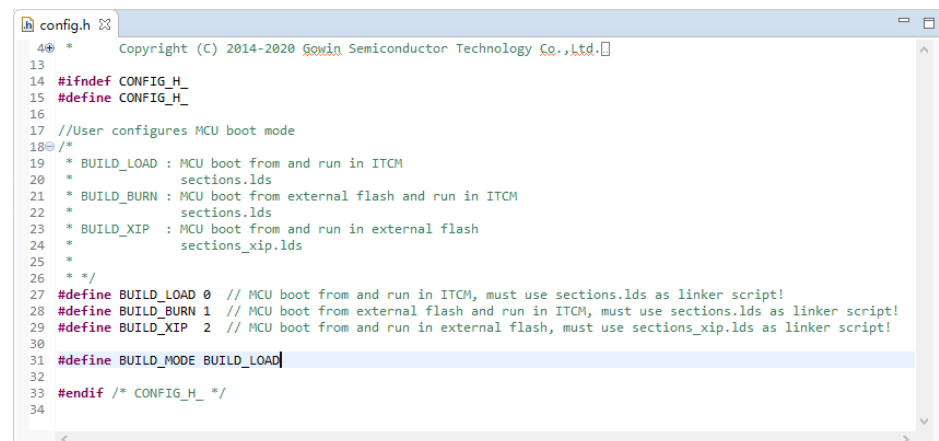


3.4 Software Configuration

3.4.1 Boot Mode Configuration

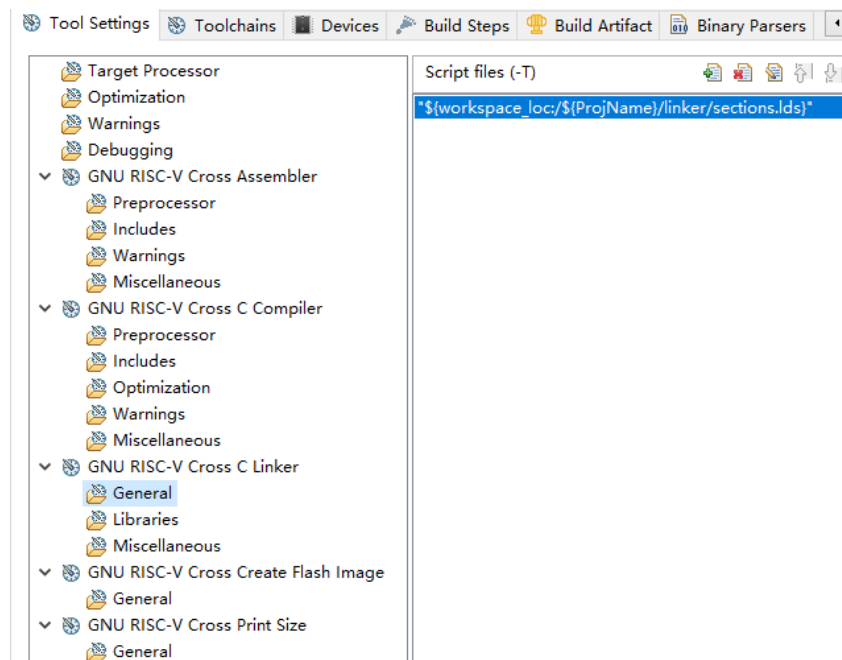
Configure the macro definition of config.h as #define BUILD_MODE BUILD_LOAD, as shown in Figure 3-3.

Figure 3-3 Boot Mode Configuration



3.4.2 Flash linker Configuration

In the parameters configuration of GNU RISC-V Cross C Linker in GOWIN MCU Designer (V1.1 and above), select sections.lds as Flash linker, as shown in Figure 3-4.

Figure 3-4 Flash Linker Configuration

3.4.3 Output File Format Configuration

Compile Gowin_PicoRV32 software programming design by GOWIN MCU Designer (V1.1 and above) and generate BIN file in a binary format in software design.

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 in Chapter 2. 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
 - Configure sections.lds as FLASH linker
 - Update user application design according to the application requirements
 - Build to generate BIN files in a binary format in software design.
3. Modify mergebin.bat according to the actual application, perform mergebin.bat, merge the bitstream files in hardware design generated by hardware design and the BIN files in a binary format in software design generated by software programming design to generate new bitstream files, as shown in Figure Figure 3-5;
4. After each update of the software programming design, rerun 3~4.

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

```

----- GOWIN Merge Tool -----
Read bit stream file gowin_picorv32.fs ...
Build bsram init value fusemap...
Reading original bsram init value map...
Location file BramLoc.txt reading...
Bsram R28[7] init value convert to fusemap success.
Bsram R28[6] init value convert to fusemap success.
Bsram R10[4] init value convert to fusemap success.
Bsram R28[5] init value convert to fusemap success.
Bsram R10[3] init value convert to fusemap success.
Bsram R28[4] init value convert to fusemap success.
Bsram R46[2] init value convert to fusemap success.
Bsram R10[2] init value convert to fusemap success.
Bsram R28[3] init value convert to fusemap success.
Bsram R46[1] init value convert to fusemap success.
Bsram R10[1] init value convert to fusemap success.
Bsram R28[2] init value convert to fusemap success.
Bsram R46[0] init value convert to fusemap success.
Bsram R10[0] init value convert to fusemap success.
Bsram R28[1] init value convert to fusemap success.
Bsram R28[0] init value convert to fusemap success.
Replace new bsram init value map to file new_gowin_picorv32.fs...
Build bsram init value replace completed.

```

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

GW2AN-9X

GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C/GW2AN-18X

GW2A-55/GW2A-55C/GW2AN-55C

3.7 Application Software

Gowin_V1.9.8 Beta and above

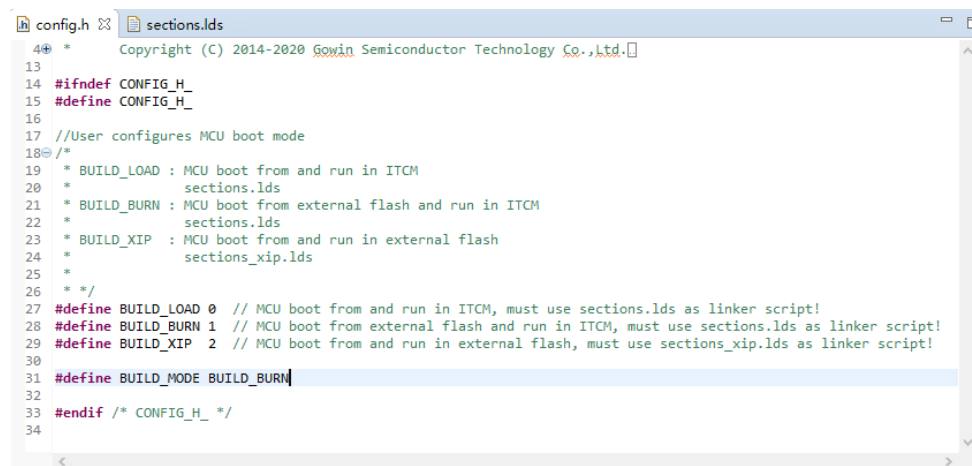
4 Off-chip SPI-Flash Download and Startup

4.1 Software Configuration

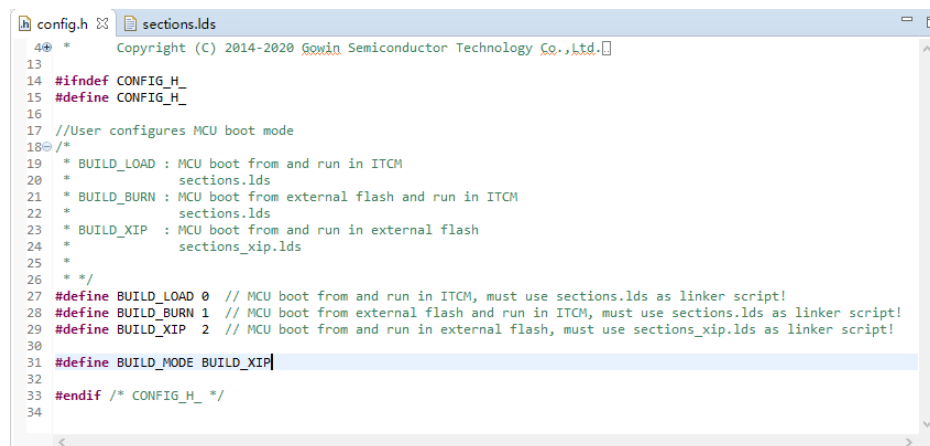
4.1.1 Boot Mode Configuration

If Boot Mode is configured as “MCU boot from external Flash and run in ITCM” in Gowin_PicoRV32 hardware design, then config.h is defined as `#define BUILD_MODE BUILD_BURN` in Gowin_PicoRV32 software programming design, as shown in Figure 4-1.

Figure 4-1 Boot Mode Configuration

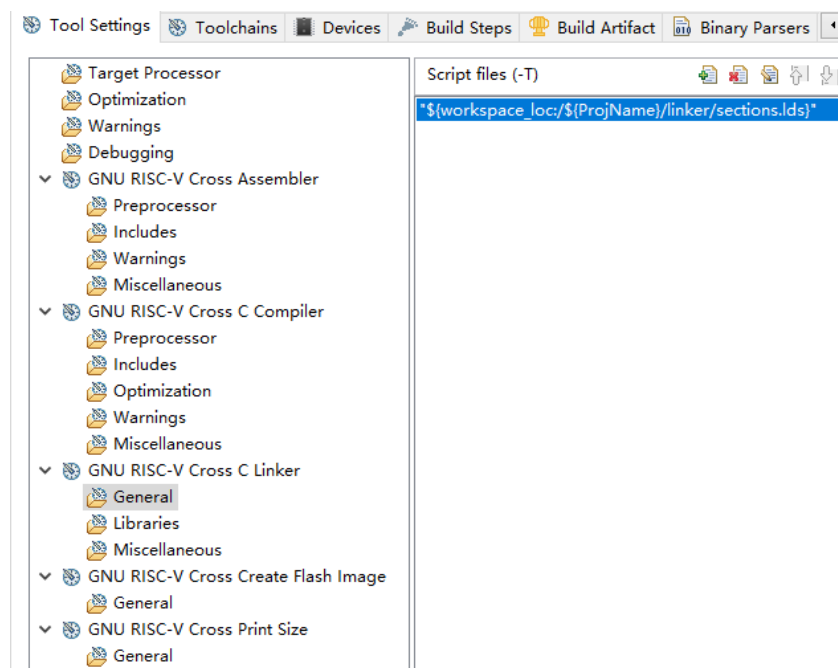


If Boot Mode is configured as “MCU boot and run in external Flash” in Gowin_PicoRV32 hardware design, then config.h is defined as `#define BUILD_MODE BUILD_XIP` in Gowin_PicoRV32 software programming design, as shown in Figure 4-2.

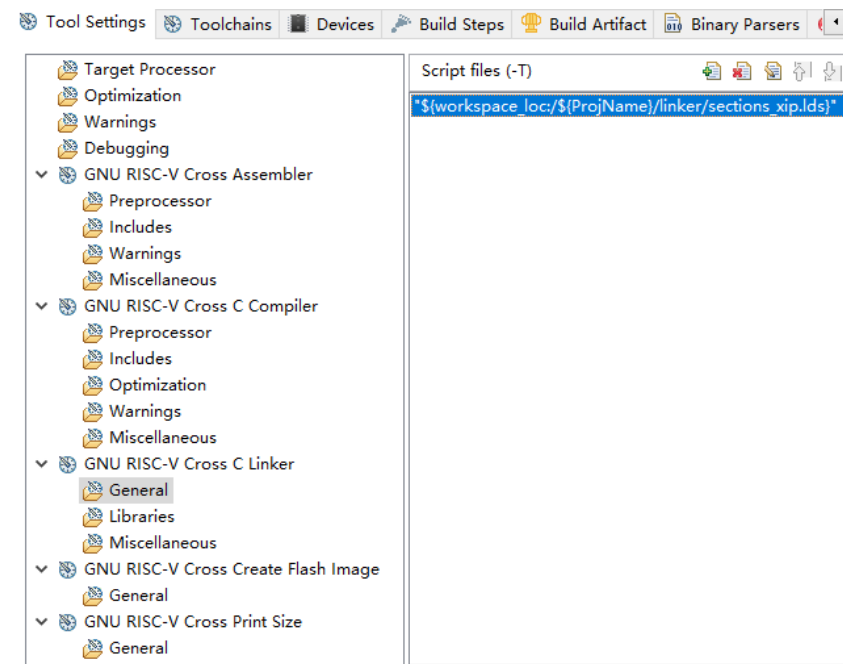
Figure 4-2 Boot Mode Configuration

4.1.2 Flash linker Configuration

If Boot Mode is configured as “MCU boot from external Flash and run in ITCM” in Gowin_PicoRV32 hardware design, then configure sections.lds as Flash linker in the configuration of GNU RISC-V Cross C Linker in GOWIN MCU Designer (V1.1 and above) , as shown in Figure 4-3, “`${workspace_loc}/${ProjName}/linker/sections.lds`”.

Figure 4-3 Flash Linker Configuration

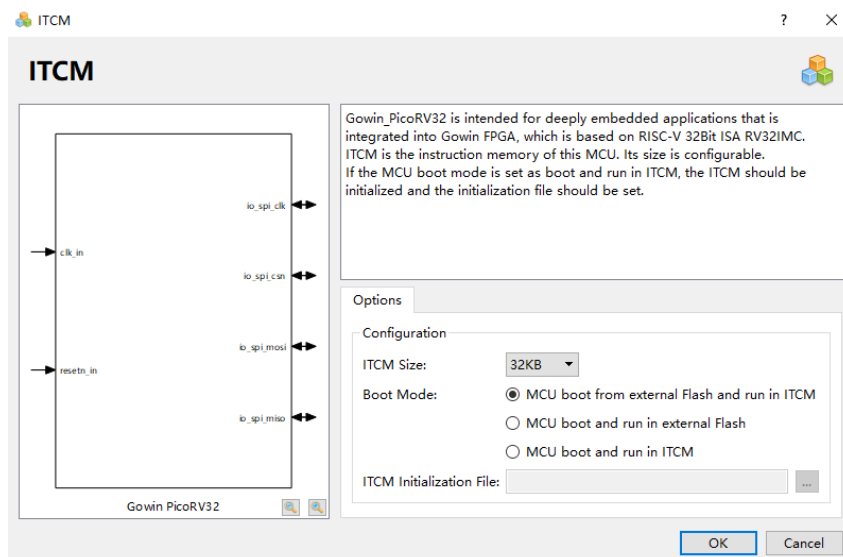
If Boot Mode is configured as “MCU boot and run in external Flash” in Gowin_PicoRV32 hardware design, then configure sections_xip.lds as Flash linker in the configuration of GNU RISC-V Cross C Linker in GOWIN MCU Designer (V1.1 and above) , as shown in Figure 4-4, “`${workspace_loc}/${ProjName}/linker/sections_xip.lds`”.

Figure 4-4 Flash Linker Configuration

4.2 Hardware Configuration

4.2.1 Boot Mode Configuration

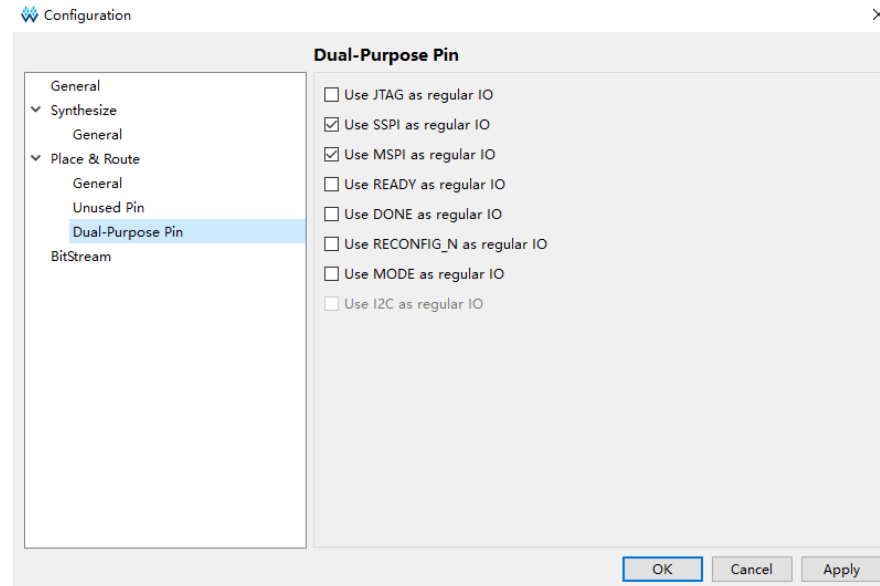
When configuring to generate the Gowin_PicoRV32 hardware design in IP Core Generator, double click to open the ITCM configuration page, and select "MCU boot from external Flash and run in ITCM" or "MCU boot and run in external Flash" in Boot Mode, as shown in Figure 4-5.

Figure 4-5 Boot Mode Configuration

4.2.2 Dual-Purpose Pin Configuration

In Gowin_PicoRV32 hardware design, configure MSPI as regular IO under “Place & Route > Dual-Purpose Pin”, as shown in Figure 4-6.

Figure 4-6 Dual-Purpose Pin Configuration



4.3 Design Flow


1. Gowin_PicoRV32 Hardware Design:
 - Select “MCU boot from external Flash and run in ITCM” or “MCU boot and run in external Flash” in Boot Mode
 - Generate bitstream files in hardware design with the functions of off-chip SPI-Flash downloading and startup.
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`
 - Configure sections_xip.lids as FLASH linker
 - MCU boot from external Flash and run in ITCM
 - Define the macro definition of config.h as `#define BUILD_MODE BUILD_BURN`
 - Configure sections.lids as FLASH linker
 - Build to generate BIN files in a binary format in software design.
4. Configure Device configuration with Programmer to download the BIN files in a binary format in software design.

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 in hardware design with the functions of off-chip SPI-Flash downloading and startup.

Run Gowin Programmer, click "Edit > Configure Device" in the menu bar or "Configure Device" () in the tool bar to open the "Device configuration".

Select "External Flash Mode" in "Access Mode" drop-down list.

Select "exFlash Erase, Program thru GAO-Bridge" or "exFlash Erase, Program, Verify thru GAO-Bridge" in "Operation" drop-down list.

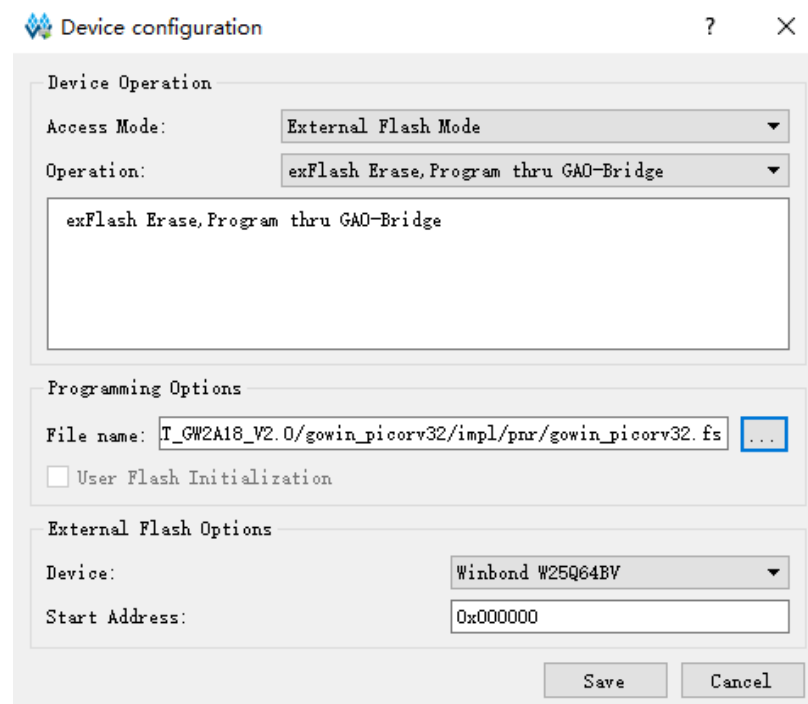
Import the hardware design bitstream file required to download in "Programming Options > File name" option.

Select "External Flash Options > Device" based on the on-board Flash chip types (such as on-board Winbond W25Q64BV of DK-START-GW2A18 V2.0).

Configure the start address as "0x000000" in "External Flash Options > Start Address" option.

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

Figure 4-7 Device Configuration



After device configuration, click "Program/Configure" () in the

Programmer toolbar to complete the downloading of bitstream files in hardware design.

4.4.2 Download BIN Files in a Binary Format in Software Design

After Gowin_PicoRV32 software programming design, generate Bin files in a binary format in software design, and use Programmer to download Gowin_PicoRV32 BIN files in a binary format in software design.

In GOWIN MCU Designer (V1.1 and above), Click "Run > Programmer" in the menu bar or "Programmer" (📁) in the tool bar to open Programmer.

Click "Edit > Configure Device" in the menu bar or "Configure Device" (🔧) in the tool bar to open Device configuration.

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" in "Operation" drop-down list.

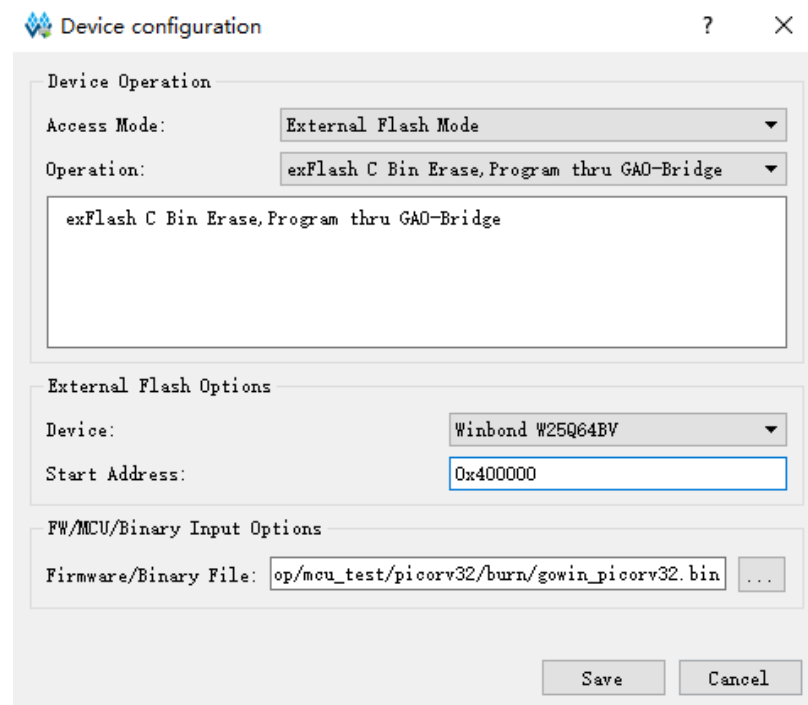
Select in "FW/MCU Input Options > Firmware/Binary File" to import the BIN files in binary format in software design to download.


Select "External Flash Options > Device" based on the on-board Flash chip types (such as on-board Winbond W25Q64BV of DK-START-GW2A18 V2.0).

Configure the start address as "0x400000" in "External Flash Options > Start Address" option.

Click "Save", as shown in Figure 4-8.

Figure 4-8 Device Configuration



After device configuration, click "Program/Configure" () in the Programmer tool bar to complete the downloading of BIN files in a binary format in software design.

4.5 Devices Supported

GW2AN-9X

GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C/GW2AN-18X

GW2A-55/GW2A-55C/GW2AN-55C

