



Gowin PicoRV32 Software Download **Reference Manual**

IPUG913-1.2E,06/01/2020

Copyright© 2020 Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

Disclaimer

GOWINSEMI[®], LittleBee[®], Arora, and the GOWINSEMI logos are trademarks of GOWINSEMI 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, as described at www.gowinsemi.com. 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 extended 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 Stac Overflow.
06/01/2020	1.2E	<ul style="list-style-type: none">● MCU software online debugging function supported;● MCU core interrupt handling function enhanced;● MCU core instructions optimized;● Mergebin tool updated and naming convention resolution for the synthesis tool of "GowinSynthesis" supported.

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 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	5
2.5 Design Flow.....	5
2.6 Devices Supported	6
3 Merge Results of Software Design and Hhardware Design	7
3.1 Tools.....	7
3.2 Command Parameters.....	7
3.3 Hardware Configuration	8
3.3.1 Boot Mode Configuration	8
3.3.2 Post-Place File Configuration.....	10
3.4 Software Configuration	10
3.4.1 Boot Mode Configuration	10
3.4.2 Flash linker Configuration	11
3.4.3 Output File Format Configuration	11
3.5 Design Flow.....	11
3.5.1 Merge.....	11
3.5.2 Download	12
3.6 Devices Supported	12
3.7 Application Software	12
4 Off-chip SPI-Flash Download and Startup.....	13

4.1 Software Configuration	13
4.1.1 Boot Mode Configuration	13
4.1.2 Flash linker Configuration	14
4.2 Hardware Configuration	15
4.2.1 Boot Mode Configuration	15
4.2.2 Dual-Purpose Pin Configuration	16
4.3 Design Flow.....	17
4.4 Download	17
4.4.1 Download Bitstream File	17
4.4.2 Download BIN file	18
4.5 Devices Supported	19

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 Configuration	5
Figure 3-1 Boot Mode and ITCM Initial Value Configuration.....	10
Figure 3-2 posp Configuration.....	10
Figure 3-3 Boot Mode Configuration	11
Figure 3-4 FLASH Linker Configuration.....	12
Figure 3-5 Merge the Results of Software Design and Hardware Design.....	12
Figure 4-1 Boot Mode Configuration	13
Figure 4-2 Boot Mode Configuration	14
Figure 4-3 FLASH Linker Configuration.....	14
Figure 4-4 FLASH Linker Configuration.....	15
Figure 4-5 Boot Mode Configuration	16
Figure 4-6 Dual-Purpose Pin Configuration	16
Figure 4-7 Device Configuration.....	18
Figure 4-8 Device Configuration.....	19

List of Tables

Table 3-1 mergebin Commands and Parameters7

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 ITCM initial value in hardware design.

- Gowin_PicoRV32 Software Design:
 - Define config.h macro: #define BUILD_MODE BUILD_LOAD;
 - Configure sections.lds as the Flash linker;
 - Compile and generate software design BIN file.
- Convert the software design BIN file to the Hex mapping file (ram32.hex) using makehex;
- IP Core Generator generates Gowin_PicoRV32 ITCM:
 - Boot Mode select MCU;
 - Boot and run in ITCM Mode, import ram32.hex in list b) as the initial value of ITCM in ITCM Initialiaztion File.
- Synthesize and place&route, and generate the hardware design bitstream files containing software design and hardware design;
- Download the hardware design bitstream file using Gowin Programmer;
- Re-execute list a) ~ e) after update of the software programming design.

2. Merge the software design BIN fille generated by software programming design and the bitstream file output by hardware design.

- Generate Gowin_PicoRV32 hardware design bitstream file and post-place file according to the requirements. If there is no need to update hardware , the generated bitstream file will not be updated;
- Update Gowin_PicoRV32 software programming design:

- Define config.h macro: #define BUILD_MODE BUILD_LOAD;
- Configure sections.lids as the Flash linker;
- Update the user design according to the application;
- Compile and generate software design BIN file.
- Merge the BIN file and the bitstream file in list a) using a mergebin tool;
- Generate the new hardware design bitstream file after merging;
- Download the new hardware design bitstream file using Gowin Programmer;
- Re-execute list b)~ e) after update of the software programming design.

3. Download the software design BIN file generated by software programming design using off-chip SPI-Flash.

- a) IP Core Generator generates Gowin_PicoRV32 ITCM:
 - Boot Mode select MCU boot and run in external Flash or MCU boot from external Flash and run in ITCM.
- b) Gowin_PicoRV32 hardware design generates a hardware design bitstream file with the function of off-chip SPI-Flash downloading and startup;
- c) Download the hardware design bitstream file using Gowin Programmer;
- d) Gowin_PicoRV32 Software Design:
 - MCU boot and run in external Flash;
 - Define config.h macro: #define BUILD_MODE BUILD_XIP;
 - Configure sections_xip.lids as the Flash linker.
 - MCU boot from external Flash and run in ITCM;
 - Define config.h macro: #define BUILD_MODE BUILD_BURN;
 - Configure sections.lids as the Flash linker.
 - Compile and generate software design BIN file.
- e) Download the software design BIN file generated by software programming design using Gowin Programmer.

2 Software Programming Output Used as ITCM Initialization Value

2.1 Tools

- Linux:
Gowin_PicoRV32\tool\linux\makehex32\bin\makehex32
- Windows:
Gowin_PicoRV32\tool\windows\makehex32\bin\makehex32.exe

Note!

The above tools link is given at http://cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

2.2 Command Parameters

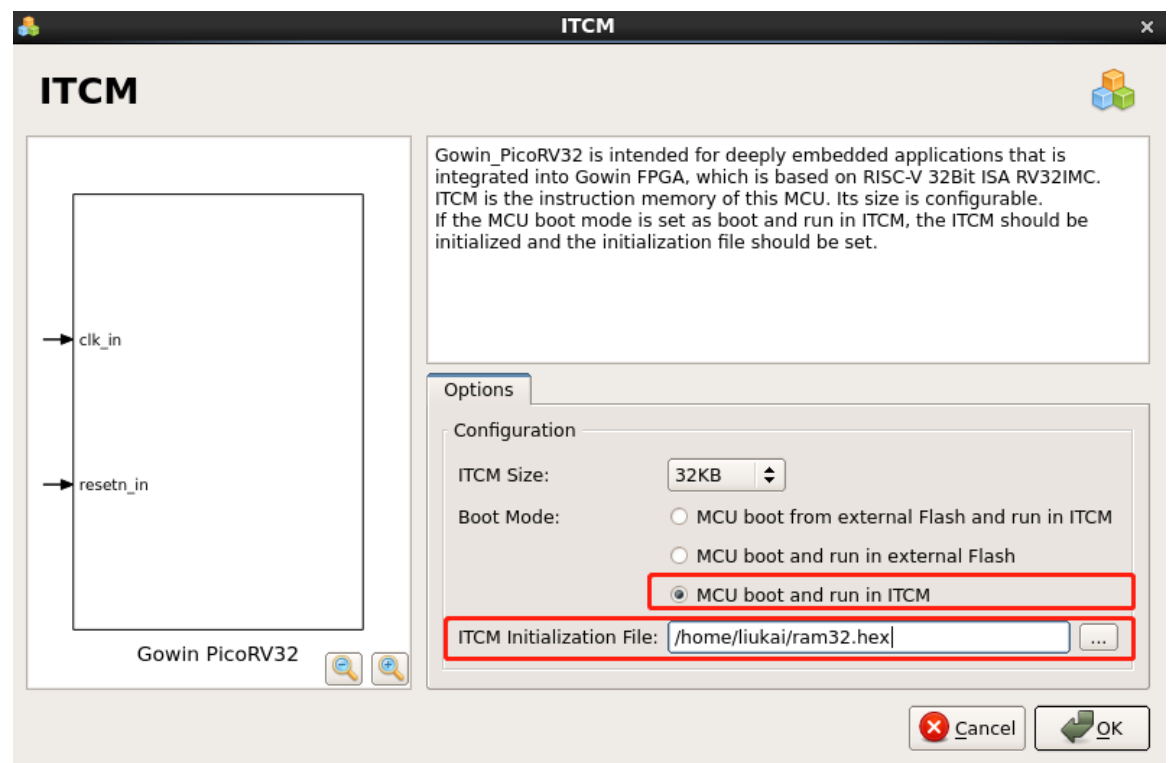
- Linux: makehex32 bin-file
- Windows: makehex32.exe bin-file

2.3 Hardware Configuration

Configuring Gowin_PicoRV32 in IP Core Generator, double click to open the ITCM configuration interface:

- Select "MCU boot and run in ITCM" in "Boot Mode".
- Import the software programming mapping file of ram32.hex in the "ITCM Initialization File" as the ITCM initial value, 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 config.h macro: #define BUILD_MODE BUILD_LOAD as shown in Figure 2-2.

Figure 2-2 Boot Mode Configuration

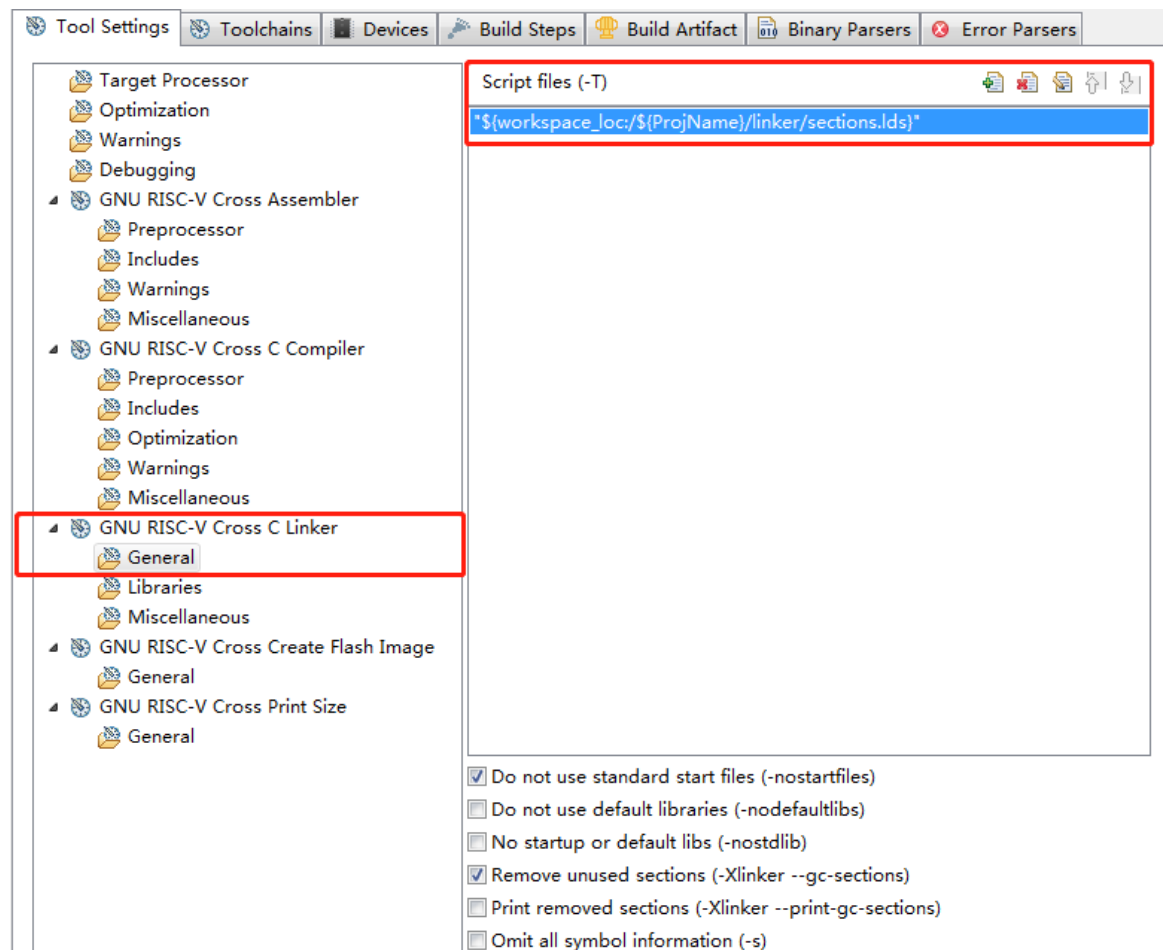
```

*config.h
Copyright (C) 2014-2020 Gowin Semiconductor Technology Co.,Ltd.
13
14 #ifndef CONFIG_H_
15 #define CONFIG_H_
16
17 //User configures MCU boot mode
18 /*
19 * BUILD_LOAD : MCU boot from and run in ITCM
20 *               sections.lds
21 * BUILD_BURN : MCU boot from external flash and run in ITCM
22 *               sections.lds
23 * BUILD_XIP  : MCU boot from and run in external flash
24 *               sections_xip.lds
25 *
26 */
27 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
28 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
29 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
30
31 #define BUILD_MODE BUILD_LOAD
32
33 #endif /* CONFIG_H_ */
34

```

2.4.2 Flash linker Configuration

Select "sections.lds" as the Flash linker in GNU RISC-V Cross C Linker parameter configuration of GOWIN MCU Designer (V1.1 and above), 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 by GOWIN MCU Designer (V1.1 and above) and generate the software design BIN file.

Run makehex32.exe bin-file to output the Hex mapping file (ram32.hex).

2.5 Design Flow

1. Gowin_PicoRV32 Software Design:
 - Define config.h macro: `#define BUILD_MODE BUILD_LOAD;`
 - Configure section.lds as the Flash linker;
 - Compile and generate software design BIN file;
 - Run makehex32.exe to generate the Hex mapping file (ram32.hex) used as the ITCM initial value of Gowin_PicoRV32 hardware design.
2. Gowin_PicoRV32 Hardware Design:
 - Select MCU boot and run in ITCM in Boot Mode;
 - Use ram32.hex as the initial value of ITCM;

3. Generate Gowin_PicoRV32 hardware design and instantiate Gowin_PicoRV32 Top Module for user design;
4. Add physical and timing Constraints;
5. Use Synplify_Pro or GowinSynthesis to synthesize;
6. Use Place & Route to place and route and generate hardware design bitstream file;
7. Download the bitstream file using Gowin Programmer;
8. Re-execute list 1~ 7 after update of the software programming design.

2.6 Devices Supported

- GW1N-9/GW1N-9C/GW1NR-9/GW1NR-9C
- GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C
- GW2A-55/GW2A-55C

3 Merge Results of Software Design and Hhardware Design

3.1 Tools

- Linux: Gowin_PicoRV32\tool\ linux\mergebin\bin\mergebin.sh
- Windows: Gowin_PicoRV32\tool\windows\mergebin\bin\mergebin.bat

Note!

The above tools link is given at http://cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

3.2 Command Parameters

- Linux: mergebin.sh
- Windows: mergebin.bat

Take mergebin.bat for instance, the software commands and parameters are as follows:

```
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 placement and layout, generate the BRAM location information, 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 的 ITCM Size (KB) For example, if set ITCM as 64K Byte, the value is 64.
gwsyn	The naming convention for the Netlist File generated by the comprehensive tool "GowinSynthesis" defaults to the synthesis tool of Synplify Pro.
merge_bit.exe	Merge Gowin_PicoRV32 software design and hardware design

Parameter	Description
bin-file	The software design BIN file generated by Gowin_PicoRV32 software programming
BramLoc.txt	ITCM layout location information file generated by posp_parse.exe
fs-file	The hardware design bitstream file generated by Gowin_PicoRV32 hardware design.

Merge the BIN file output by software programming design and the bitstream file output by hardware design, and then output a new bitstream file.

When mergebin.sh or mergebin.bat is in use, users can modify the parameters, such as posp-file, ilm-size, gwsyn, bin-file, and fs-file according to your requirements.

3.3 Hardware Configuration

Generate Gowin_PicoRV32 hardware design bitstream file and post-place file according to the chapter 2. If there is no need to update hardware, the generated bitstream file will not be updated;

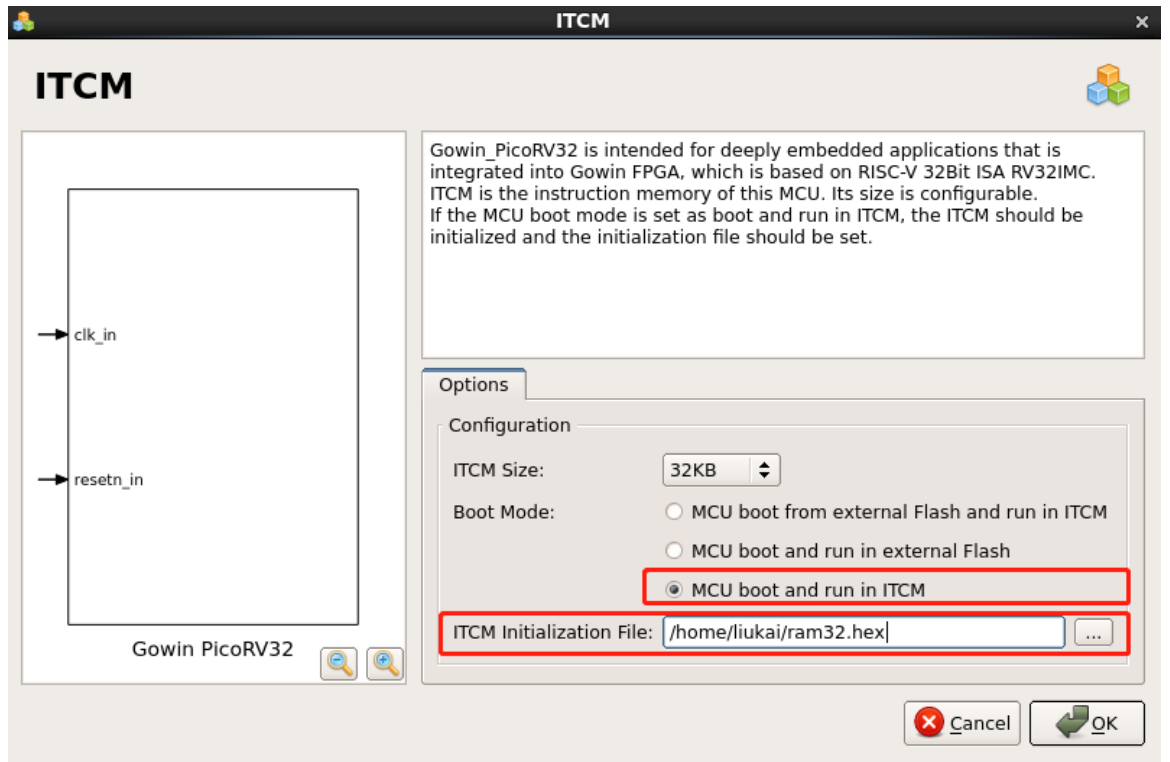
After update of the software programming design, it only needs to use the mergebin tool to merge the bitstream file and the updated software design BIN file.

3.3.1 Boot Mode Configuration

Configuring Gowin_PicoRV32 in IP Core Generator, double click to open the ITCM configuration interface:

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

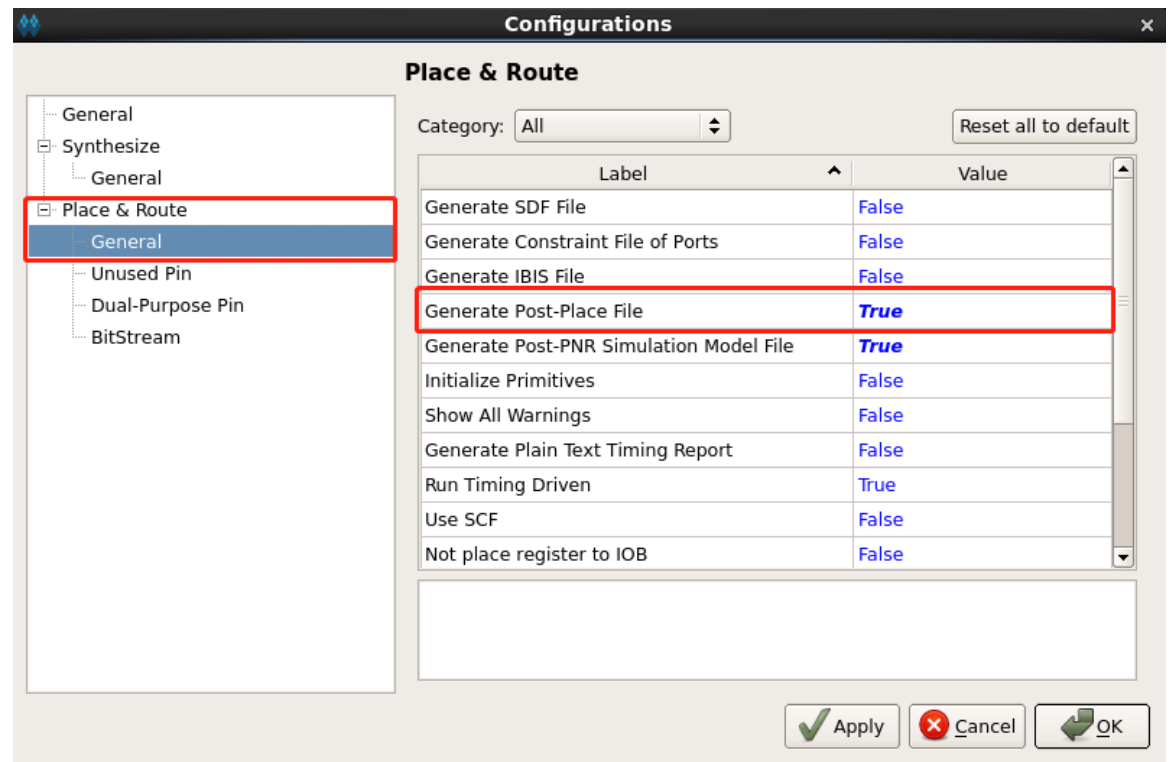
Figure 3-1 Boot Mode and ITCM Initial Value Configuration



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

Define config.h macro: #define BUILD_MODE BUILD_LOAD as shown in Figure 3-3.

Figure 3-3 Boot Mode Configuration

```

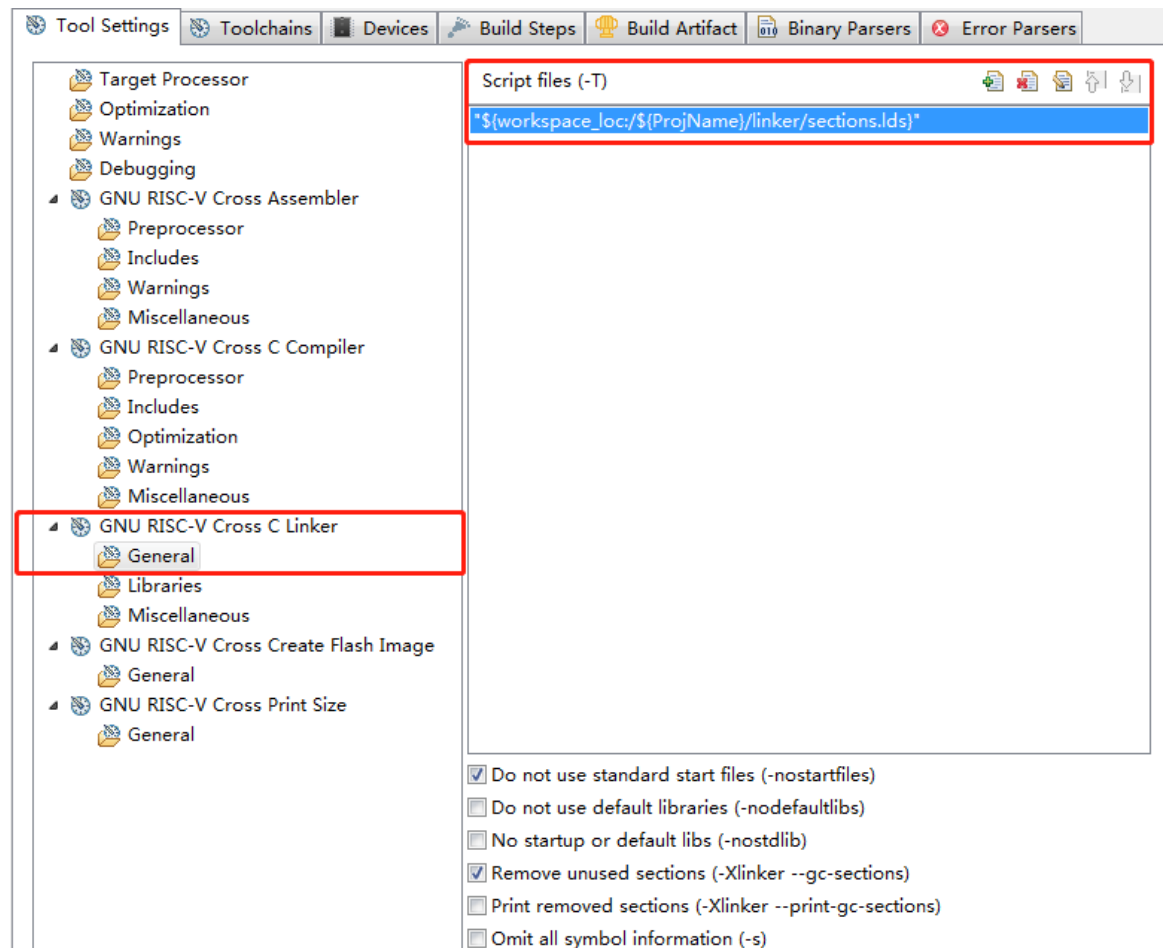
40 * Copyright (C) 2014-2020 Gowin Semiconductor Technology Co.,Ltd.
13
14 #ifndef CONFIG_H_
15 #define CONFIG_H_
16
17 //User configures MCU boot mode
18 /*
19 * BUILD_LOAD : MCU boot from and run in ITCM
20 * sections.lds
21 * BUILD_BURN : MCU boot from external flash and run in ITCM
22 * sections.lds
23 * BUILD_XIP : MCU boot from and run in external flash
24 * sections_xip.lds
25 *
26 */
27 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
28 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
29 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
30
31 #define BUILD_MODE BUILD_LOAD
32
33 #endif /* CONFIG_H_ */
34

```

3.4.2 Flash linker Configuration

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

Figure 3-4 FLASH Linker Configuration



3.4.3 Output File Format Configuration

Compile Gowin_PicoRV32 software programming by GOWIN MCU Designer (V1.1 and above) and output the BIN file.

3.5 Design Flow

3.5.1 Merge

1. Generate Gowin_PicoRV32 hardware design bitstream file and post-place file according to the requirements. If there is no need to update hardware , the generated bitstream file will not be updated;
2. Update Gowin_PicoRV32 software programming design:
 - Define config.h macro: `#define BUILD_MODE BUILD_LOAD`;
 - Configure sections.lds as the Flash linker;
 - Update the user design according to the application;

- Compile and generate hardware design BIN file.
3. Use mergebin.sh in the Linux environment or mergebin.bat in the Windows environment to merge the hardware design bitstream file and the software design BIN file generated by hardware design and then generate a new hardware design bitstream file, as shown in Figure 3-5;
 4. Re-execute list 3~ 4 after update of the software programming design.

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

```

BsrAm R10[16] init value convert to fusemap success.
BsrAm R45[15] init value convert to fusemap success.
BsrAm R54[16] init value convert to fusemap success.
BsrAm R10[15] init value convert to fusemap success.
BsrAm R54[15] init value convert to fusemap success.
BsrAm R10[14] init value convert to fusemap success.
BsrAm R54[14] init value convert to fusemap success.
BsrAm R10[13] init value convert to fusemap success.
BsrAm R45[13] init value convert to fusemap success.
BsrAm R45[12] init value convert to fusemap success.
BsrAm R36[11] init value convert to fusemap success.
BsrAm R45[11] init value convert to fusemap success.
BsrAm R36[10] init value convert to fusemap success.
BsrAm R36[9] init value convert to fusemap success.
BsrAm R36[8] init value convert to fusemap success.
BsrAm R36[7] init value convert to fusemap success.
Replace new bsrAm init value map to file new_PicoSoC_gw2a55_superboard_191106_00.fs...
Build bsrAm init value replace completed.

D:\WORKS\PicoRV32\Demo_Projects\2a55\PicoSoC_gw2a55_superBoard_191106_00\imp1\pnr>pause
请按任意键继续. . .

```

3.5.2 Download

Download the new bitstream file using Gowin Programmer after the hardware design bitstream file and the BIN file are merged.

For the Gowin Programmer usage, please refer to [SUG502](#), Gowin Programmer User Guide.

3.6 Devices Supported

- GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C
- GW2A-55/GW2A-55C

3.7 Application Software

Apply to the Gowin_PicoRV32 hardware design generated by Synplify Pro and GowinSynthesis in IP Core Generator.

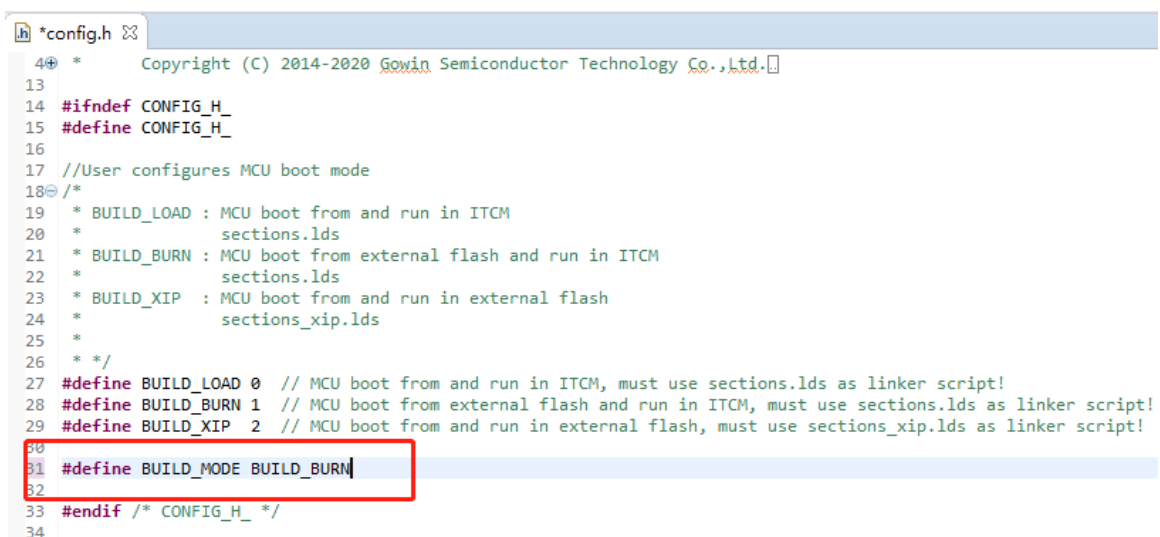
4 Off-chip SPI-Flash Download and Startup

4.1 Software Configuration

4.1.1 Boot Mode Configuration

In Gowin_PicoRV32 hardware design, if Boot Mode is configured to MCU boot from external Flash and run in ITCM, define BUILD_MODE BUILD_BURN in config.h, as shown in Figure 4-1.

Figure 4-1 Boot Mode Configuration



```
*config.h
4+ * Copyright (C) 2014-2020 Gowin Semiconductor Technology Co.,Ltd.
13
14 #ifndef CONFIG_H_
15 #define CONFIG_H_
16
17 //User configures MCU boot mode
18 /*
19 * BUILD_LOAD : MCU boot from and run in ITCM
20 * sections.lds
21 * BUILD_BURN : MCU boot from external flash and run in ITCM
22 * sections.lds
23 * BUILD_XIP : MCU boot from and run in external flash
24 * sections_xip.lds
25 *
26 */
27 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
28 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
29 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
30
31 #define BUILD_MODE BUILD_BURN
32
33 #endif /* CONFIG_H_ */
34
```

In Gowin_PicoRV32 hardware design, if Boot Mode is configured to MCU boot and run in external Flash, define BUILD_MODE BUILD_XIP in config.h, as shown in Figure 4-2.

Figure 4-2 Boot Mode Configuration

```

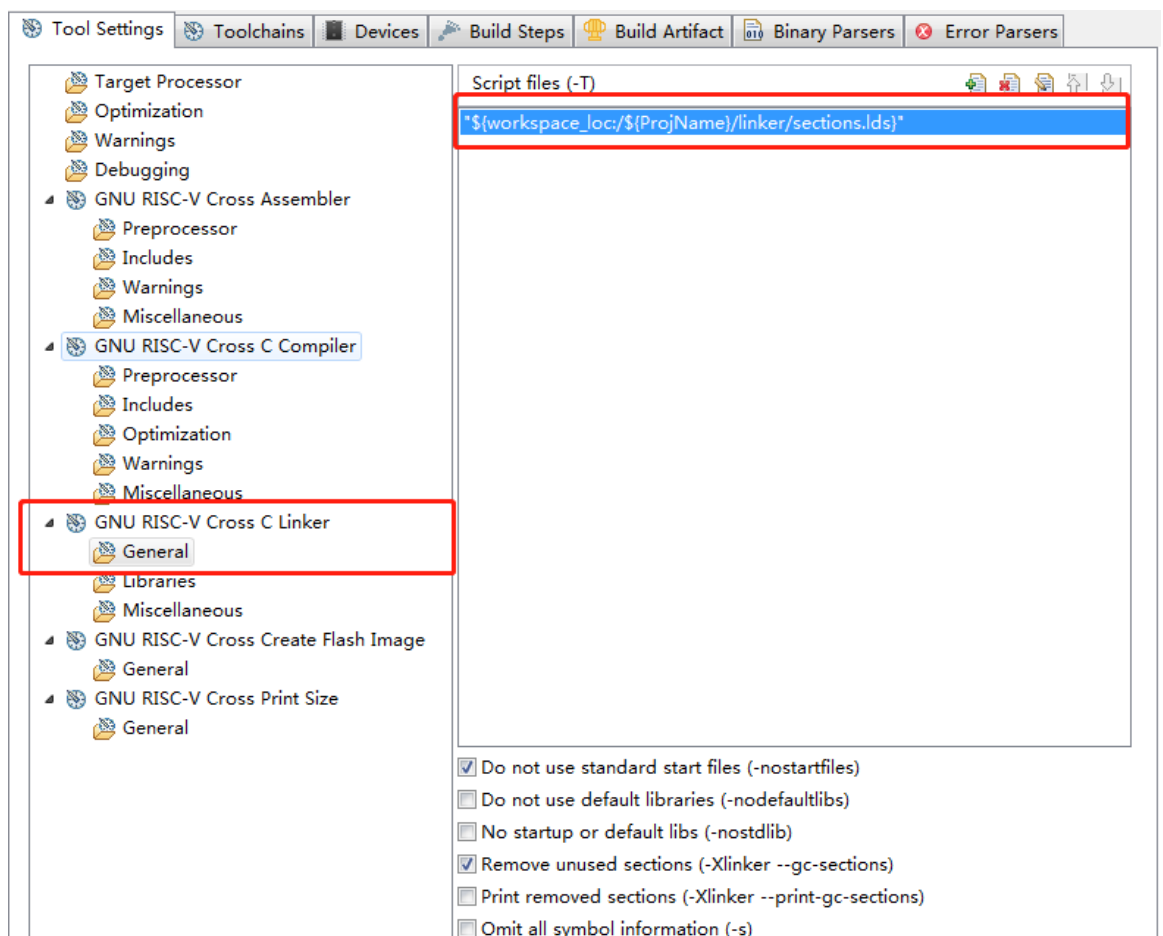
*config.h
40 * Copyright (C) 2014-2020 Gowin Semiconductor Technology Co., Ltd.
13
14 #ifndef CONFIG_H_
15 #define CONFIG_H_
16
17 //User configures MCU boot mode
18 /*
19 * BUILD_LOAD : MCU boot from and run in ITCM
20 * sections.lds
21 * BUILD_BURN : MCU boot from external flash and run in ITCM
22 * sections.lds
23 * BUILD_XIP : MCU boot from and run in external flash
24 * sections_xip.lds
25 *
26 */
27 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
28 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
29 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
30
31 #define BUILD_MODE BUILD_XIP
32
33 #endif /* CONFIG_H_ */
34

```

4.1.2 Flash linker Configuration

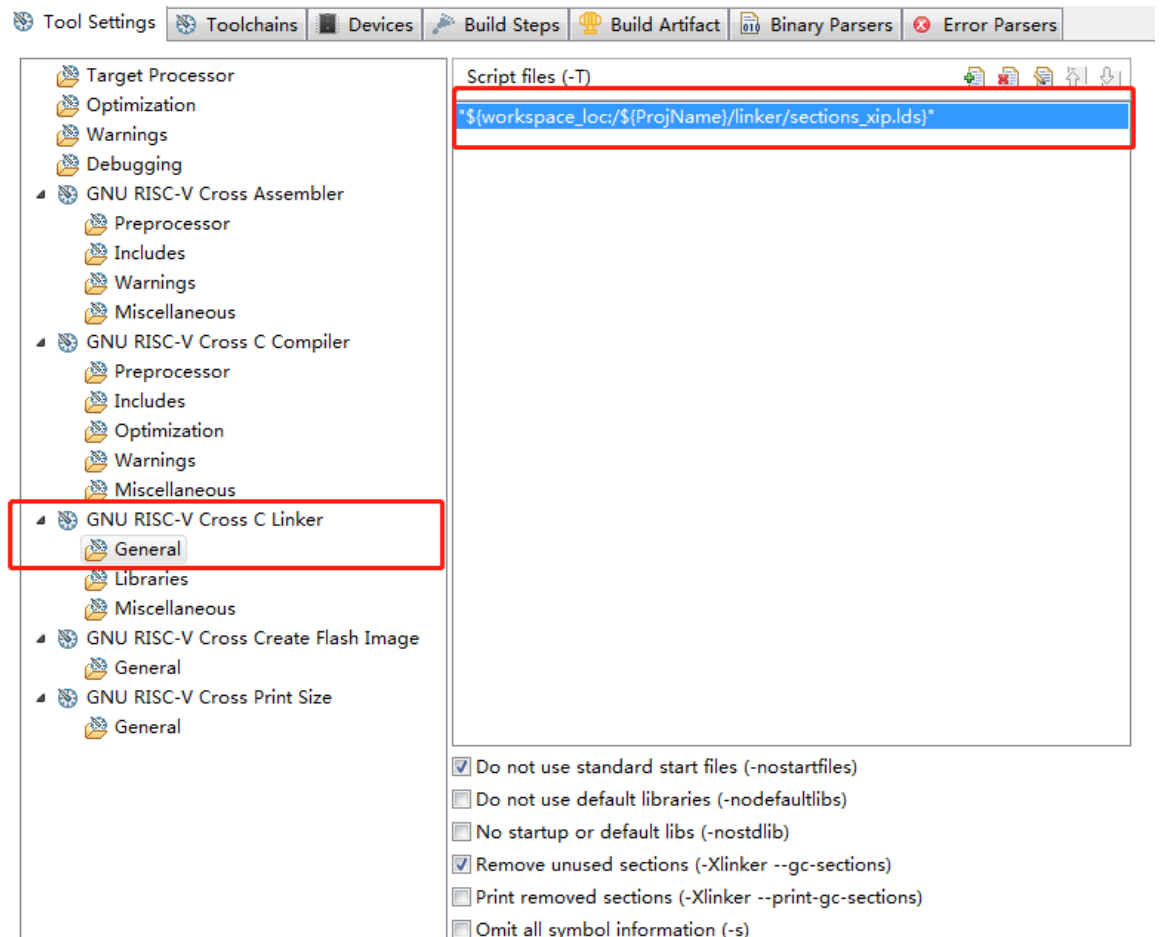
In Gowin_PicoRV32 hardware design, if Boot Mode is configured to MCU boot from external Flash and run in ITCM, configure sections.lds as Flash linker in GNU RISC-V Cross C Linker of GOWIN MCU Designer (V1.1 and above), as shown in Figure 4-6, such as, "\${workspace_loc}/\${ProjName}/linker/sections.lds".

Figure 4-3 FLASH Linker Configuration



In Gowin_PicoRV32 hardware design, if Boot Mode is configured to MCU boot and run in external Flash, configure sections_xip.lds as Flash linker in GNU RISC-V Cross C Linker of GOWIN MCU Designer (V1.1 and above), as shown in Figure 4-4, such as, "\${workspace_loc:\${ProjName}}/linker/sections_xip.lds".

Figure 4-4 FLASH Linker Configuration

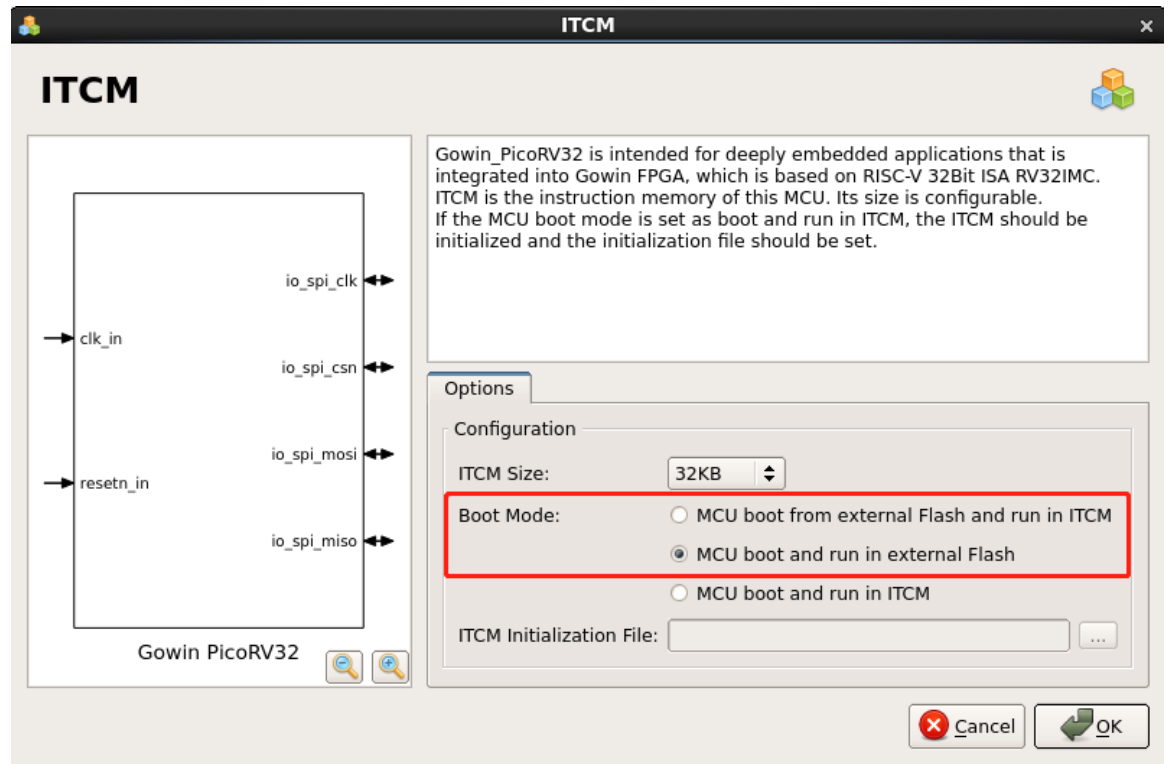


4.2 Hardware Configuration

4.2.1 Boot Mode Configuration

In Gowin_PicoRV32 hardware design, double click the ITCM configuration interface, select "MCU boot from external Flash and run in ITCM" or "MCU boot and run in external Flash", as shown in Figure 4-5.

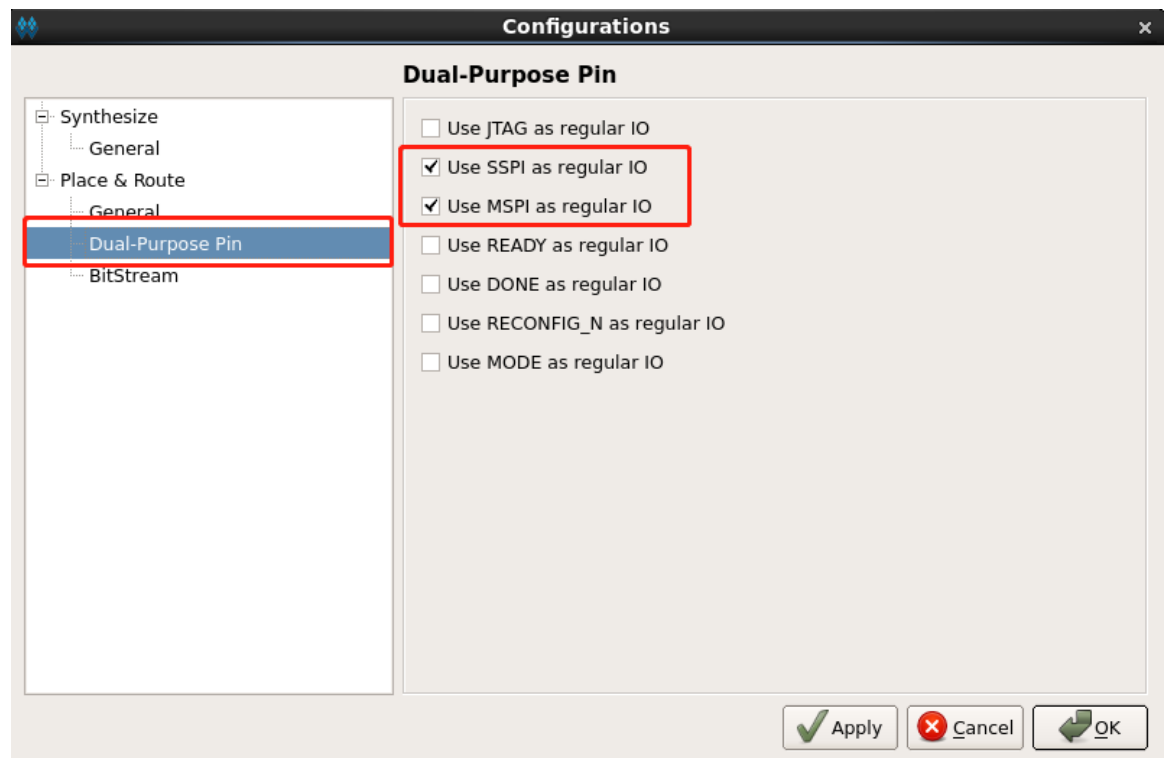
Figure 4-5 Boot Mode Configuration



4.2.2 Dual-Purpose Pin Configuration

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

Figure 4-6 Configure 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 a bitstream file with the function of hardware design off-chip SPI-Flash downloading and startup.
2. Configure Device configuration in Programmer and download the hardware design bitstream file;
3. Gowin_PicoRV32 Software Design:
 - MCU boot and run in external Flash;
Define config.h macro: #define BUILD_MODE BUILD_XIP;
Configure sections_xip.lids as the Flash linker.
 - MCU boot from external Flash and run in ITCM;
Define config.h macro: #define BUILD_MODE BUILD_BURN;
Configure sections.lids as the Flash linker.
 - Compile and generate software design BIN file;
4. Configure Device configuration in Programmer and download the BIN file.

4.4 Download

For the Gowin Programmer usage, please refer to [SUG502](#), Gowin Programmer User Guide.

4.4.1 Download Hardware Design Bitstream File

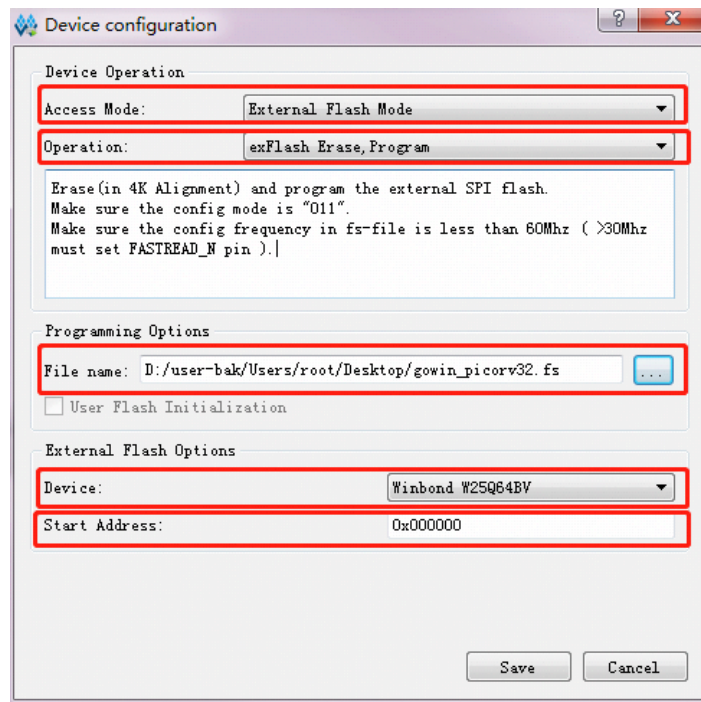
Gowin_PicoRV32 hardware design generates a hardware design bitstream file with the function of off-chip SPI-Flash. Use Programmer to download;


Run Gowin Programmer, click "Edit/Configure Device" in the menu bar or Configure Device " " in the tool bar to open the "Device configuration" dialog box.

- Select "Embedded Flash Mode" from the "Access Mode" drop-down list.
- Select "embFlash Erase, Program" from the "Operation" drop-down list.
- Import the hardware design bit stream file required in "Programming Options > File name".
- Configure the ""External Flash Options > Device" option according to the development board on-board Flash chip type, such as Gowin on-board "Winbond W25Q64BV"Flash of DK-START-GW2A18 (V2.0)Configure "External Flash Options > Start Address" as "0x000000".

- 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 hardware design bit stream files downloading.

4.4.2 Download BIN file

Complete the Gowin_PicoRV32 software programming design and generate software design BIN files. And then use the Programmer to download Gowin_PicoRV32 software design BIN file.

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


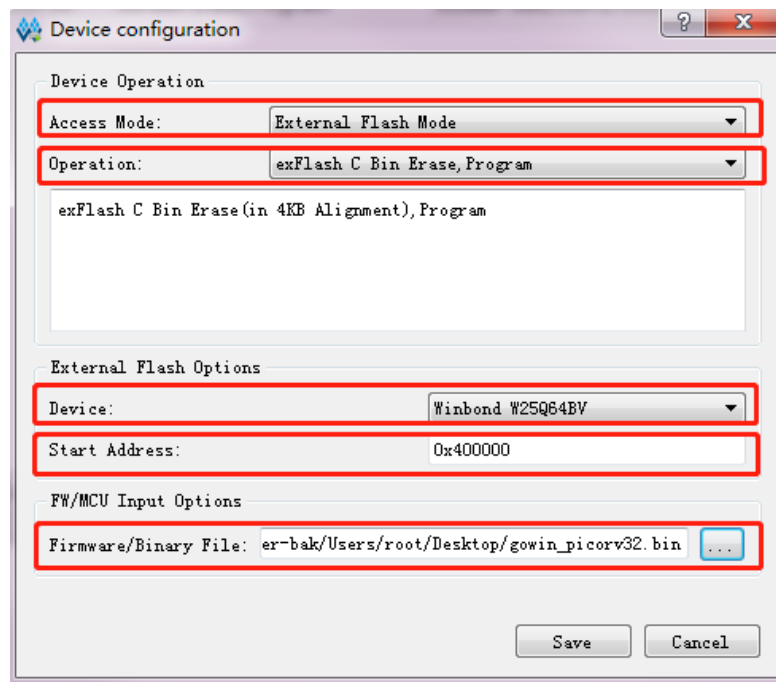

- Click "Edit > Configure Device" in the Programmer menu bar or "Configure Device" () in the toolbar to open the "Device configuration" dialog box.
- Select "External Flash Mode" from the "Access Mode" drop-down list.
- Select "exFlash C Bin Erase, Program" or "exFlash C Bin Erase, Program, Verify" from the "Operation" drop-down list.
- Import the Gowin_PicoRV32 BIN file required in " FW/MCU Input Options > Firmware/Binary File ".
- Select according to the on-board Flash chip type, such as Gowin on-board DK-START-GW2A18 (V2.0) and Winbond W25Q64BV.
- Configure " External Flash Options >Start Address" as "0x000000".
- Click "Save", as shown in Figure 4-8.

Figure 4-8 Device Configuration

After device configuration, click Program/Configure "  " in the Programmer toolbar to complete software design BIN file downloading.

4.5 Devices Supported

- GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C
- GW2A-55/GW2A-55C

