



Gowin PicoRV32 软件下载参考手册

IPUG913-1.1,2020-03-12

版权所有© 2020 广东高云半导体科技股份有限公司

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

免责声明

本档并未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任，高云半导体保留修改档中任何内容的权利，恕不另行通知。高云半导体不承诺对这些档进行适时的更新。

版本信息

日期	版本	说明
2020/01/16	1.0	初始版本。
2020/03/06	1.1	<ul style="list-style-type: none">● MCU 支持 Wishbone 总线接口的外部设备 GPIO;● MCU 支持扩展 AHB 总线接口;● MCU 支持片外 SPI-Flash 下载及运行;● MCU 支持外部设备 SPI-Flash 读、写和擦除功能;● MCU 支持 Hardware Stack Protection 和 Trap Stack Overflow 功能。

目录

目录	i
图目录	iii
表目录	iv
1 下载方法	1
2 软件编程输出作为硬件 ITCM 初始值	3
2.1 软件工具	3
2.2 命令参数	3
2.3 硬件配置	3
2.4 软件配置	4
2.4.1 Boot Mode 配置	4
2.4.2 Flash 链接器配置	4
2.4.3 输出文件格式配置	5
2.5 设计流程	5
2.6 适用器件	6
3 合并软件编程设计和硬件设计	7
3.1 软件工具	7
3.2 命令参数	7
3.3 硬件配置	8
3.3.1 Boot Mode 配置	8
3.3.2 Post-Place File 配置	8
3.4 软件配置	9
3.4.1 Boot Mode 配置	9
3.4.2 Flash 链接器配置	9
3.4.3 输出文件格式配置	10
3.5 设计流程	10
3.5.1 合并	10
3.5.2 下载	11
3.6 适用器件	11

3.7 适用软件.....	11
4 片外 SPI-Flash 下载启动方法.....	12
4.1 软件配置.....	12
4.1.1 Boot Mode 配置.....	12
4.1.2 Flash 链接器配置.....	13
4.2 硬件配置.....	14
4.2.1 Boot Mode 配置.....	14
4.2.2 Dual-Purpose Pin 配置.....	15
4.3 设计流程.....	16
4.4 下载.....	16
4.4.1 下载码流文件.....	16
4.4.2 下载软件设计 BIN 文件.....	17
4.5 适用器件.....	18

图目录

图 2-1 Boot Mode 以及 ITCM 初始值配置	4
图 2-2 Boot Mode 配置	4
图 2-3 Flash 链接器脚本配置	5
图 3-1 配置布局布线 posp 选项	9
图 3-2 Boot Mode 配置	9
图 3-3 Flash 链接器配置	10
图 3-4 合并软件编程设计输出和硬件设计输出	11
图 4-1 Boot Mode 配置	12
图 4-2 Boot Mode 配置	13
图 4-3 Flash 链接器配置	13
图 4-4 Flash 链接器配置	14
图 4-5 Boot Mode 配置	15
图 4-6 配置 Dual-Purpose Pin	15
图 4-7 GW1N 系列 Configure Device	16
图 4-8 GW2A 系列 Configure Device	17
图 4-9 Device configuration	18

表目录

表 3-1 mergebin 命令及参数 7

1 下载方法

Gowin_PicoRV32 支持三种硬件设计和软件编程设计下载方法：

1. 软件编程设计输出映像文件，作为硬件设计中指令存储器 ITCM 的初始值。
 - a) Gowin_PicoRV32 软件编程设计：
 - 定义 config.h 宏定义 #define BUILD_MODE BUILD_LOAD
 - 配置 sections.lds 作为 Flash 链接器
 - 编译输出二进制 BIN 文件
 - b) 使用 makehex 工具将二进制 BIN 文件转换为十六进制映像文件 ram32.hex;
 - c) IP Core Generator 生成 Gowin_PicoRV32 硬件设计中配置指令存储器 ITCM，Boot Mode 选择 MCU boot and run in ITCM 方式，ITCM Initialiaztion File 导入 b).中 ram32.hex 文件作为 ITCM 的初始值;
 - d) 综合、布局布线，输出包括软件编程设计和硬件设计的码流文件;
 - e) Programmer 下载码流文件;
 - f) 每次更新软件编程设计后，重新执行 a)~ e)。
2. 合并软件编程设计输出的二进制 BIN 文件和硬件设计输出的码流文件。
 - a) 根据应用需求，按照方法 1.产生 Gowin_PicoRV32 硬件设计码流文件及 Post-Place File，如果没有硬件更新需求，则固定硬件设计码流文件不再更新;
 - b) 更新 Gowin_PicoRV32 软件编程设计：
 - 定义 config.h 宏定义 #define BUILD_MODE BUILD_LOAD
 - 配置 sections.lds 作为 Flash 链接器
 - 根据应用需求，更新用户应用设计
 - 编译输出二进制 BIN 文件
 - c) 使用 mergebin 工具合并二进制 BIN 文件和 a)中硬件设计码流文件;
 - d) 输出合并软件编程设计和硬件设计后的新的码流文件;
 - e) Programmer 下载合并后的新的码流文件;
 - f) 每次更新软件编程设计后，重新执行 b)~ e)。
3. 片外 SPI-Flash 下载软件编程设计输出的二进制 BIN 文件。
 - a) IPCore Generator 生成 Gowin_PicoRV32 硬件设计中配置指令存储器 ITCM，Boot Mode 选择 MCU boot and run in external Flash 或 MCU boot from external Flash and run in ITCM 方式;

- b) Gowin_PicoRV32 硬件设计输出具有片外 SPI-Flash 下载启动功能的码流文件；
- c) Programmer 下载硬件设计输出的码流文件；
- d) Gowin_PicoRV32 软件编程设计：
 - MCU boot and run in external Flash
定义 config.h 宏定义 `#define BUILD_MODE BUILD_XIP`
配置 sections_xip.lds 作为 Flash 链接器
 - MCU boot from external Flash and run in ITCM
定义 config.h 宏定义 `#define BUILD_MODE BUILD_BURN`
配置 sections.lds 作为 Flash 链接器
 - 编译输出二进制 BIN 文件
- e) Programmer 下载软件编程设计输出的二进制 BIN 文件。

2 软件编程输出作为硬件 ITCM 初始值

2.1 软件工具

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

通过此链接获取上述软件工具:

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

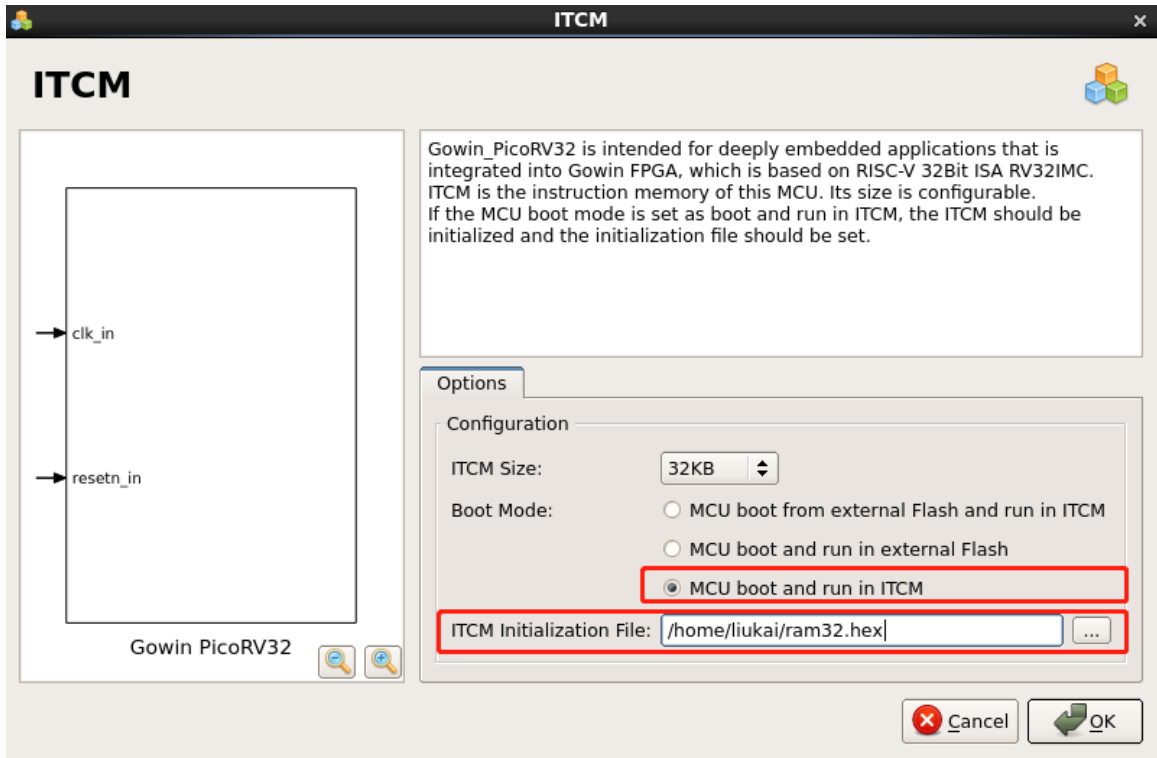
2.2 命令参数

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

2.3 硬件配置

IP Core Generator 中配置 Gowin_PicoRV32 时，双击打开 ITCM 配置页面，“Boot Mode”选择“MCU boot and run in ITCM”方式，“ITCM Initialization File”中导入软件编程设计映像文件“ram32.hex”文件作为 ITCM 的初始值，如图 2-1 所示。

图 2-1 Boot Mode 以及 ITCM 初始值配置

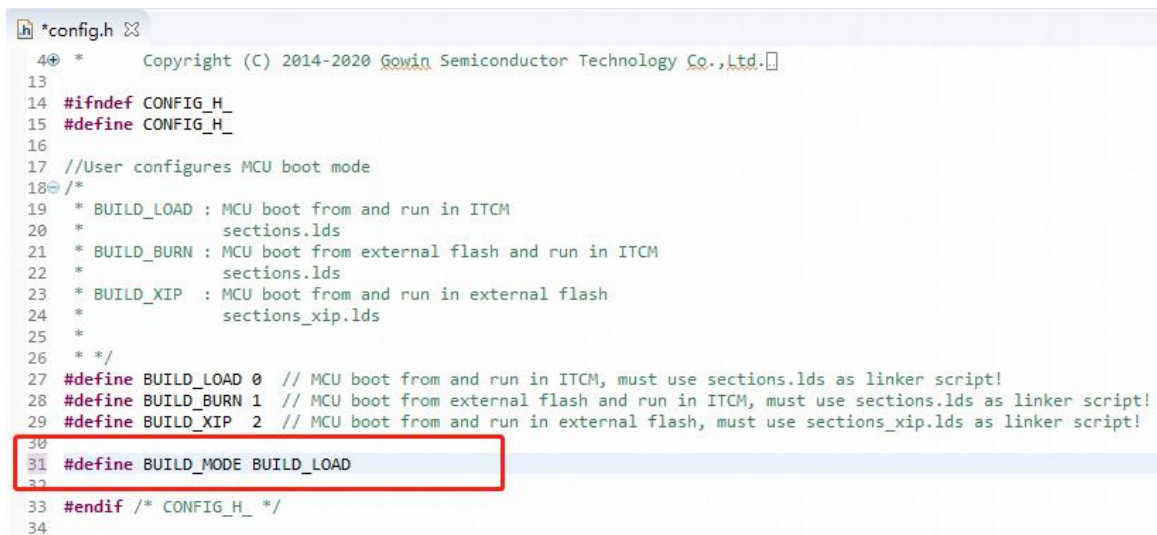


2.4 软件配置

2.4.1 Boot Mode 配置

定义 config.h 宏定义：#define BUILD_MODE BUILD_LOAD，如图 2-2 所示。

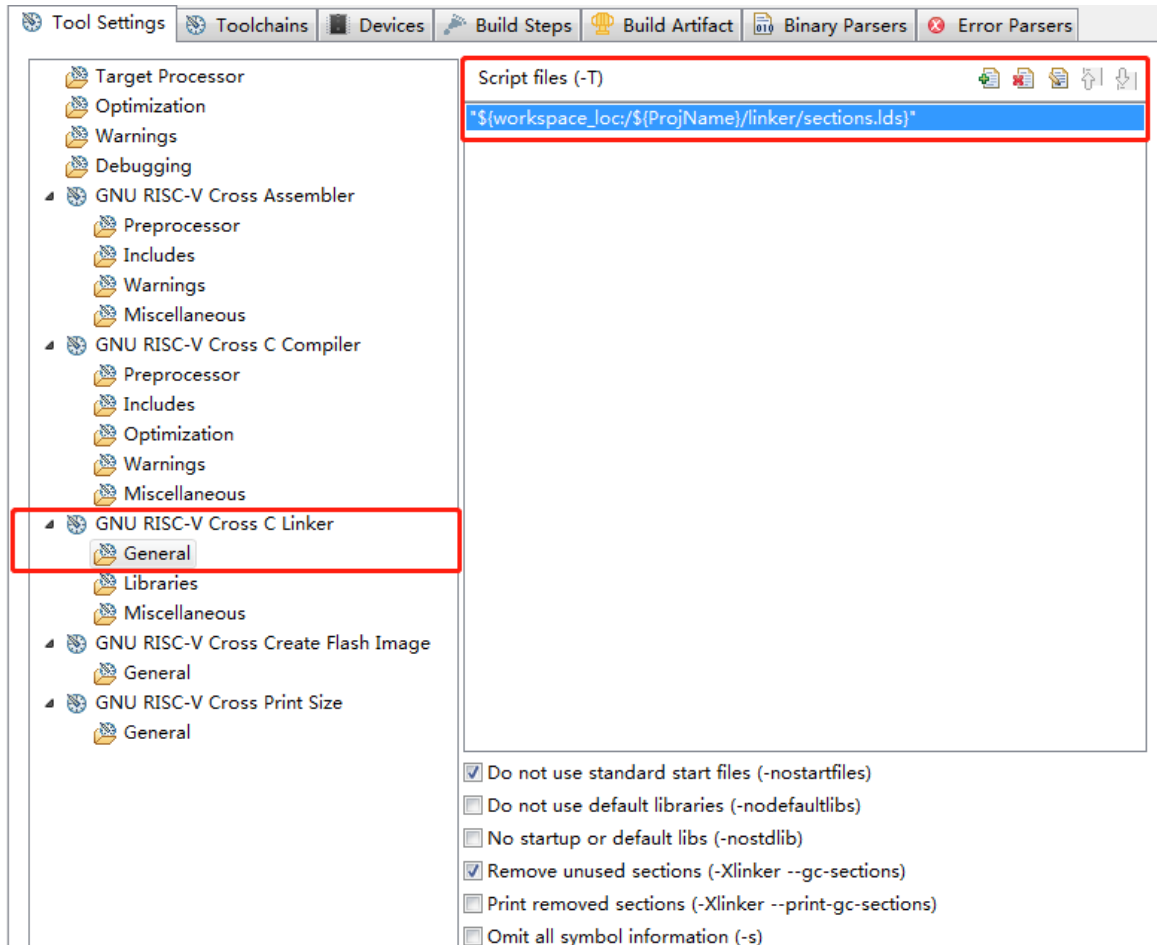
图 2-2 Boot Mode 配置



2.4.2 Flash 链接器配置

GOWIN MCU Designer V1.0 及以上版本 GNU RISC-V Cross C Linker 参数配置中，选择“sections.lds”作为 Flash 链接器，如图 2-3 所示。

图 2-3 Flash 链接器脚本配置



2.4.3 输出文件格式配置

GOWIN MCU Designer V1.0 及以上版本编译 Gowin_PicoRV32 软件编程设计，输出二进制 BIN 文件。

运行 makehex32.exe bin-file，输出十六进制映像文件 ram32.hex。

2.5 设计流程

1. Gowin_PicoRV32 软件编程设计：
 - config.h 定义 #define BUILD_MODE BUILD_LOAD
 - 配置 section.lds 作为 Flash 链接器
 - 编译输出二进制 BIN 文件
 - 运行 makehex32.exe，输出十六进制映像文件 ram32.hex，作为 Gowin_PicoRV32 硬件设计中 ITCM 的初始值；
2. Gowin_PicoRV32 硬件设计：
 - Boot Mode 配置为 MCU boot and run in ITCM
 - ram32.hex 作为 ITCM 的初始值
3. 实例化 Gowin_PicoRV32，连接用户设计；
4. 物理约束和时序约束；
5. 使用 Synplify_Pro 或 GowinSynthesis 综合；
6. 使用 Place & Route 布局布线，输出码流文件；

7. 使用 Programmer 下载;
8. 每次更新软件编程设计后, 重新执行上述 1~7。

2.6 适用器件

- GW1N-9 系列
- GW1NR-9 系列
- GW2A-18 系列
- GW2A-18C 系列
- GW2AR-18 系列
- GW2AR-18C 系列
- GW2A-55 系列
- GW2A-55C 系列

3 合并软件编程设计和硬件设计

3.1 软件工具

- Linux: Gowin_PicoRV32\tool\linux\mergebin\bin\mergebin.sh
- Windows: Gowin_PicoRV32\tool\windows\mergebin\bin\mergebin.bat
- 通过此链接获取上述软件工具:
http://cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

3.2 命令参数

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

以 mergebin.bat 为例，描述软件工具命令及参数。

```
call posp_parse.exe posp-file itcm-size
```

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

其中命令及参数描述如表 3-1 所示。

表 3-1 mergebin 命令及参数

参数	描述
posp_parse.exe	解析布局布线产生的 posp 文件，生成 ITCM 中 BRAM 的位置信息，并输出到文件 BramLoc.txt 中。
posp-file	posp 文件名，文件名与工程名一致，扩展名为 posp。
itcm-size	Gowin_PicoRV32 的 ITCM Size (KB) 如：若 ITCM 设置为 64K Byte，则该参数为 64。
merge_bit.exe	合并 Gowin_PicoRV32 硬件设计和软件编程设计
bin-file	Gowin_PicoRV32 软件编程设计输出的二进制 BIN 文件
BramLoc.txt	posp_parse.exe 产生的 ITCM 布局位置信息文件
fs-file	Gowin_PicoRV32 硬件设计输出的码流文件

合并软件编程设计输出的二进制 BIN 文件与硬件设计输出的码流文件，输出新的码流文件。

mergebin.sh 或 mergebin.bat 在使用过程中，根据实际需求修改参数 posp-file、itcm-size、bin-file、fs-file。

3.3 硬件配置

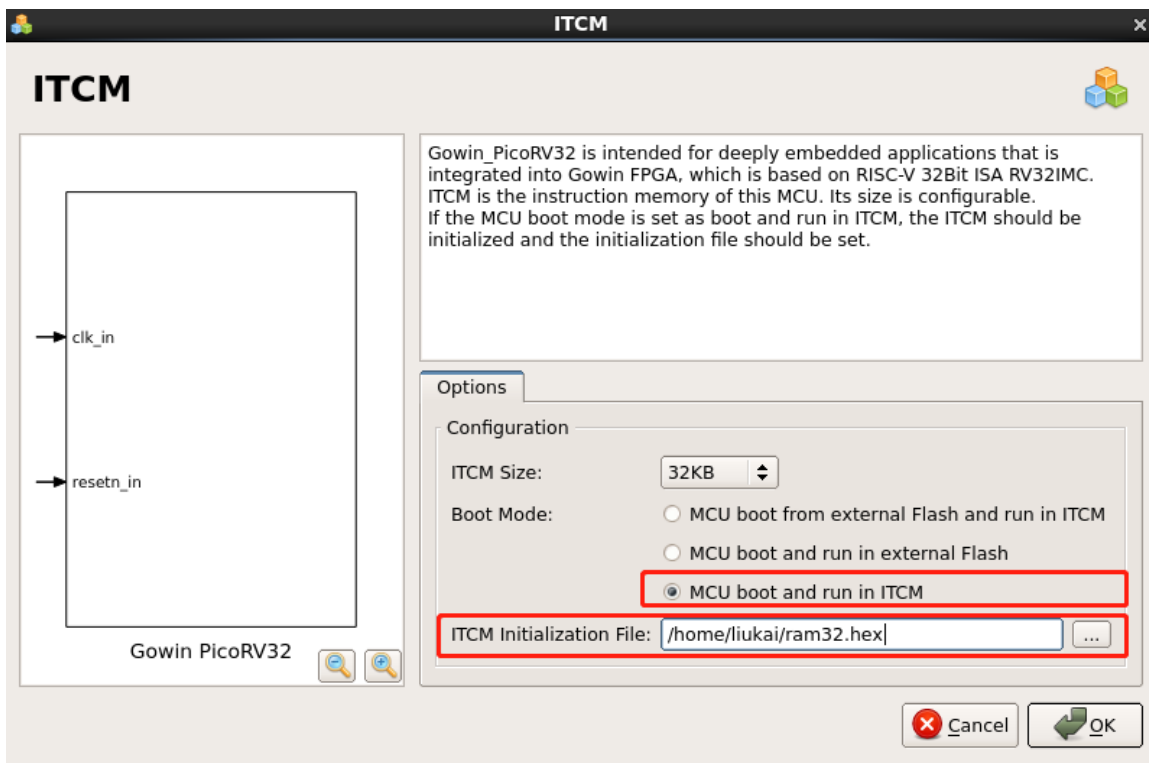
按照第 2 章方法 1 产生 Gowin_PicoRV32 的硬件设计,综合、布局布线,产生硬件设计码流文件及 Post-Place File, 如果没有硬件更新需求, 则固定硬件设计码流文件不再更新。

每次更新软件编程设计后, 只需要每次使用 mergebin 工具合并上述硬件设计码流文件和每次更新的软件编程设计 BIN 文件。

3.3.1 Boot Mode 配置

IP Core Generator 中配置 Gowin_PicoRV32 时, 双击打开 ITCM 配置页面, “Boot Mode” 选择 “MCU boot and run in ITCM” 方式, “ITCM Initialization File” 中导入软件编程设计产生的映像文件 “ram32.hex” 文件作为 ITCM 的初始值, 如图 3-1 所示。

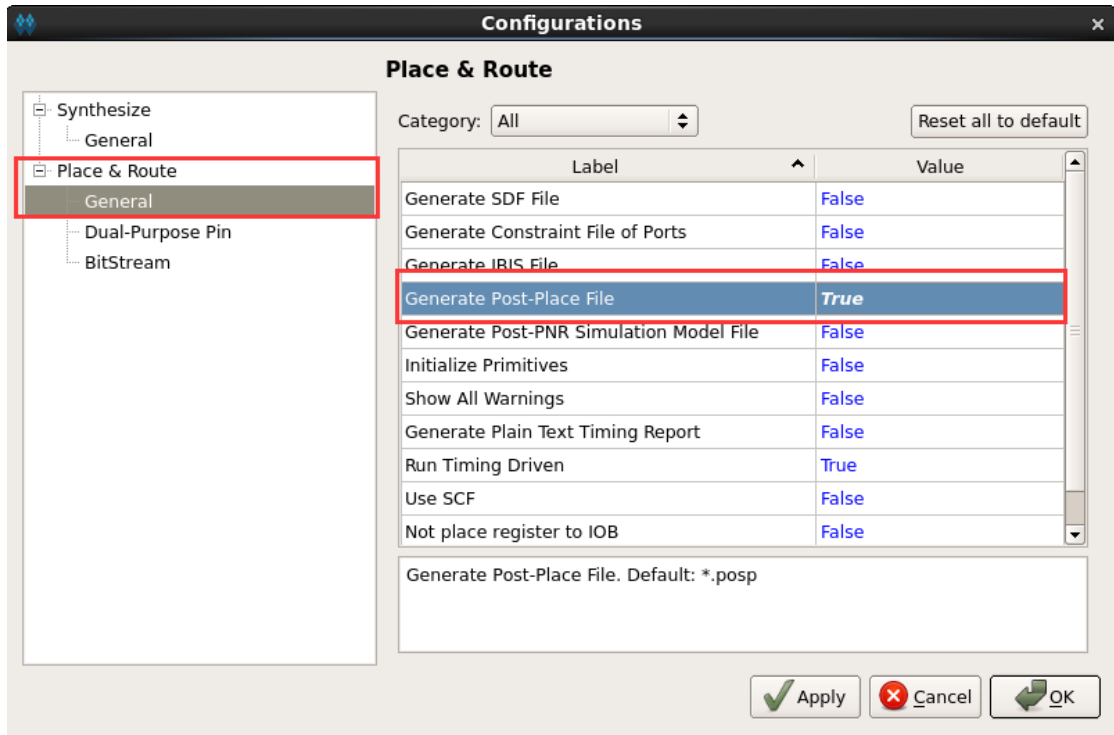
图 3-1 Boot Mode 以及 ITCM 初始值配置



3.3.2 Post-Place File 配置

Gowin_PicoRV32 硬件设计中, 配置页面 Place & Route 标签下的 Generate Post-Place File 的值设置为 True, 产生 posp 文件, 作为 posp_parse.exe 参数的 posp 输入文件, 如图 3-1 所示。

图 3-1 配置布局布线 posp 选项



3.4 软件配置

3.4.1 Boot Mode 配置

配置 config.h 宏定义: #define BUILD_MODE BUILD_LOAD, 如图 3-2 所示。

图 3-2 Boot Mode 配置

```

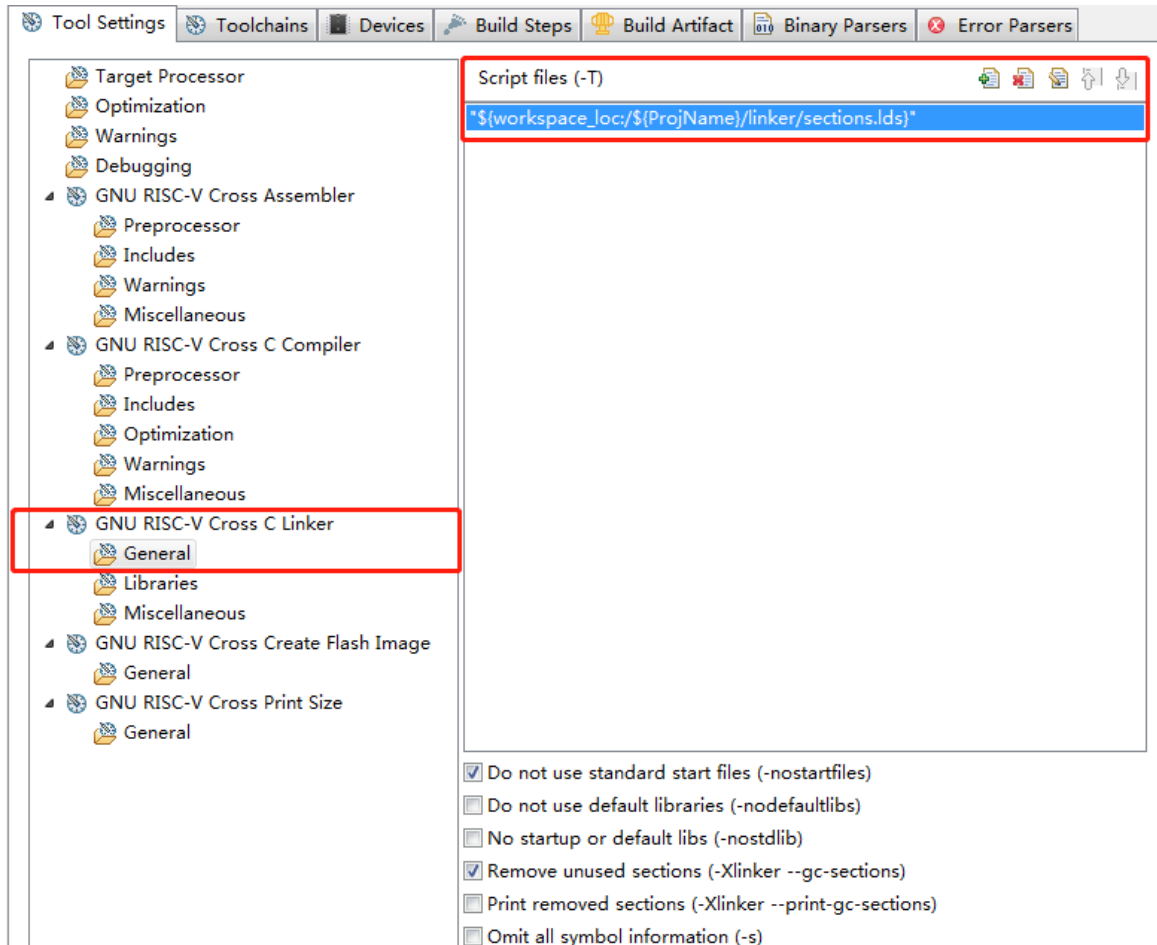
13
14 #ifndef CONFIG_H_
15 #define CONFIG_H_
16
17 //User configures MCU boot mode
18 /*
19 * BUILD_LOAD : MCU boot from and run in ITCM
20 *               sections.lds
21 * BUILD_BURN : MCU boot from external flash and run in ITCM
22 *               sections.lds
23 * BUILD_XIP  : MCU boot from and run 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 链接器配置

GOWIN MCU Designer V1.0 及以上版本 GNU RISC-V Cross C Linker 参数配置中, 选择 sections.lds 作为 Flash 链接器, 如图 3-3 所示。

图 3-3 Flash 链接器配置



3.4.3 输出文件格式配置

GOWIN MCU Designer V1.0 及以上版本编译 Gowin_PicoRV32 软件编程设计，输出二进制 BIN 文件。

3.5 设计流程

3.5.1 合并

1. 根据应用需求，按照第 2 章的方法 1 产生 Gowin_PicoRV32 硬件设计码流文件及 Post-Place File，如果没有硬件更新需求，则固定硬件设计码流文件不再更新；
2. 更新 Gowin_PicoRV32 软件编程设计：
 - 定义 config.h 宏定义 `#define BUILD_MODE BUILD_LOAD`
 - 配置 sections.lds 作为 Flash 链接器
 - 根据应用需求，更新用户应用设计
 - 编译输出二进制 BIN 文件
3. Linux 环境执行 mergebin.sh 或 Windows 环境执行 mergebin.bat，合并硬件设计输出的码流文件和软件编程设计输出的二进制 BIN 文件，输出新的码流文件，如图 3-4 所示；
4. 每次更新软件编程设计后，重新执行 3~4。

图 3-4 合并软件编程设计输出和硬件设计输出

```
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 下载

完成合并后，使用 Programmer 下载新的码流文件。

Programmer 使用方法请参考 [SUG502](#), Gowin Programmer 用户指南。

3.6 适用器件

- GW2A-18 系列
- GW2A-18C 系列
- GW2AR-18 系列
- GW2AR-18C 系列
- GW2A-55 系列
- GW2A-55C 系列

3.7 适用软件

适用于 IP Core Generator 中使用综合工具 Synplify Pro 产生的 Gowin_PicoRV32 硬件设计。

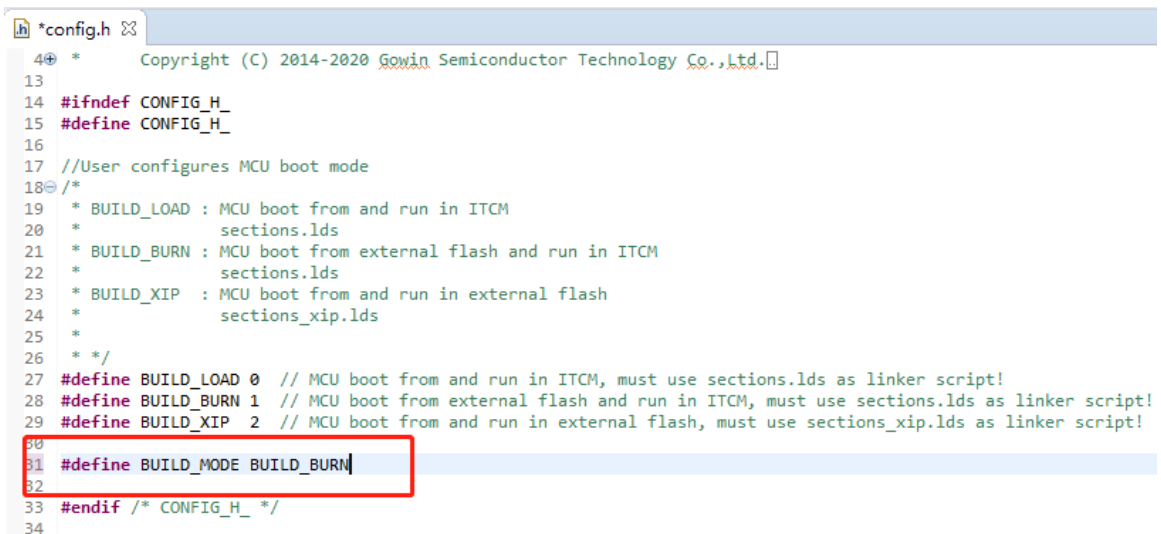
4 片外 SPI-Flash 下载启动方法

4.1 软件配置

4.1.1 Boot Mode 配置

如果 Gowin_PicoRV32 硬件设计中，Boot Mode 配置为 MCU boot from external Flash and run in ITCM 方式，则 Gowin_PicoRV32 软件编程设计 config.h 中，定义#define BUILD_MODE BUILD_BURN，如图 4-1 所示。

图 4-1 Boot Mode 配置



```
*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_BURN
32
33 #endif /* CONFIG_H_ */
34
```

如果 Gowin_PicoRV32 硬件设计中，Boot Mode 配置为 MCU boot and run in external Flash 方式，则 Gowin_PicoRV32 软件编程设计 config.h 中，定义#define BUILD_MODE BUILD_XIP，如图 4-5 所示。

图 4-2 Boot Mode 配置

```

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

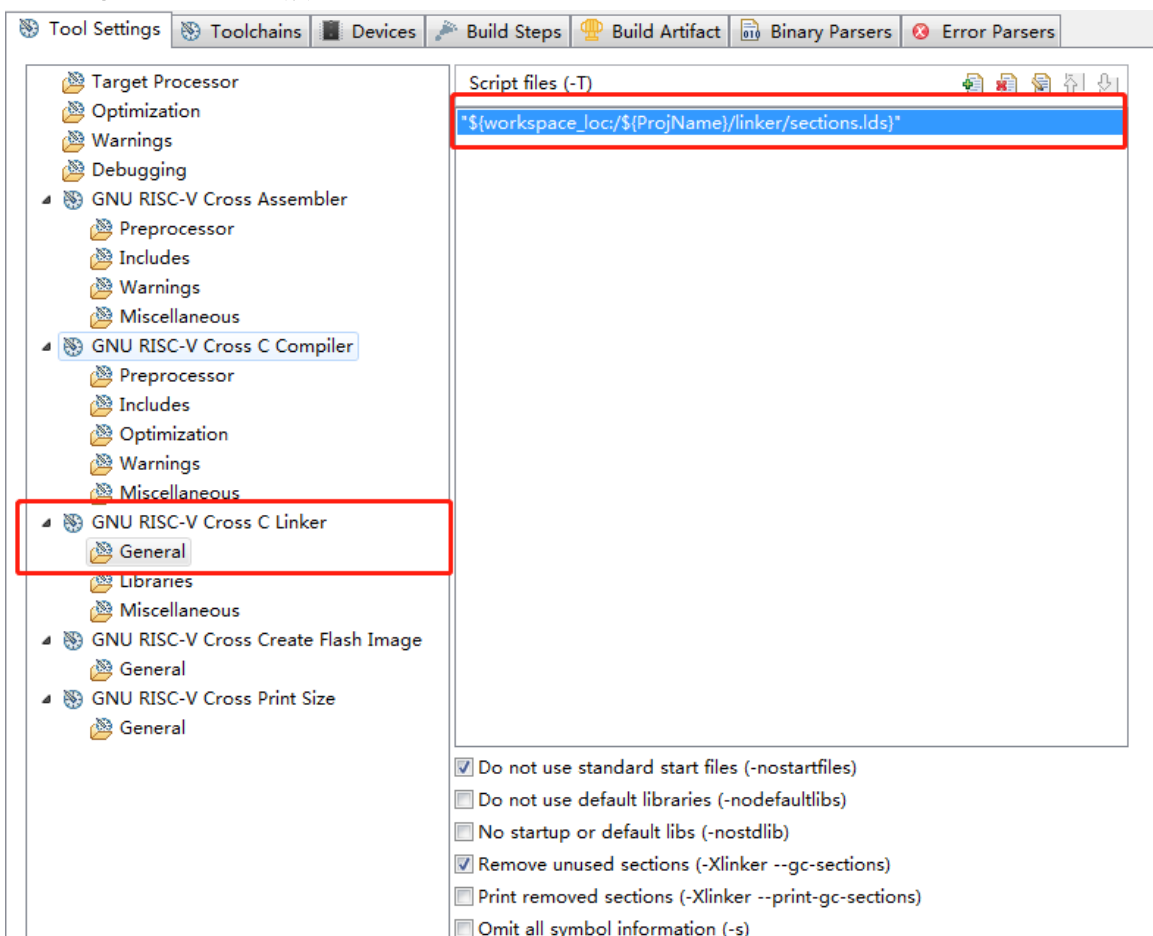
```

4.1.2 Flash 链接器配置

如果 Gowin_PicoRV32 硬件设计中，Boot Mode 配置为 MCU boot from external Flash and run in ITCM 方式，则 GOWIN MCU Designer V1.0 及以上版本中 GNU RISC-V Cross C Linker 选项配置 sections.lds 作为 Flash 链接器，如图 4-6 所示。

如：“\${workspace_loc:\${ProjName}}/linker/sections.lds”。

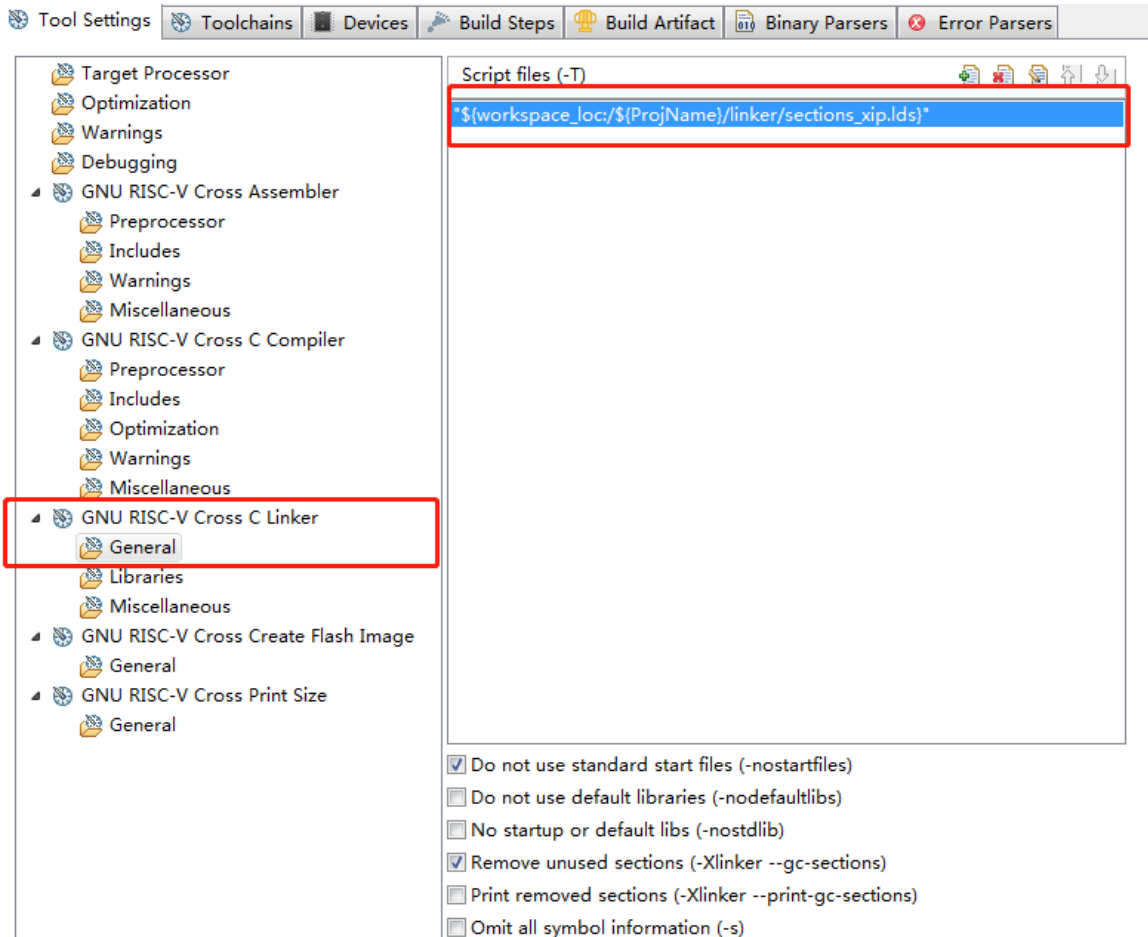
图 4-3 Flash 链接器配置



如果 Gowin_PicoRV32 硬件设计中，Boot Mode 配置为 MCU boot and run in external Flash 方式，则 GOWIN MCU Designer V1.0 及以上版本中 GNU RISC-V Cross C Linker 选项配置 sections_xip.lds 作为 Flash 链接器，如图 4-4 所示。

如：“\${workspace_loc}/\${ProjName}/linker/sections_xip.lds”。

图 4-4 Flash 链接器配置

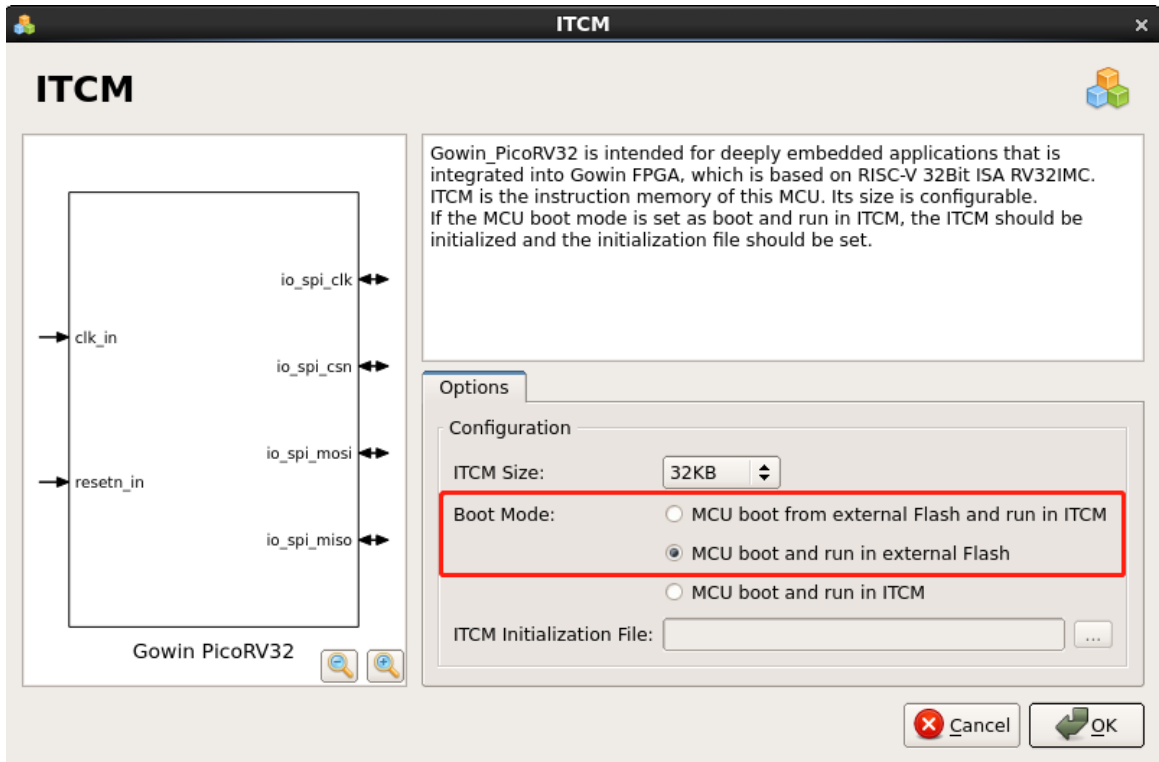


4.2 硬件配置

4.2.1 Boot Mode 配置

IP Core Generator 配置产生 Gowin_PicoRV32 硬件设计过程中，双击打开 ITCM 配置页面，“Boot Mode”选择“MCU boot from external Flash and run in ITCM”或“MCU boot and run in external Flash”方式，如图 4-5 所示。

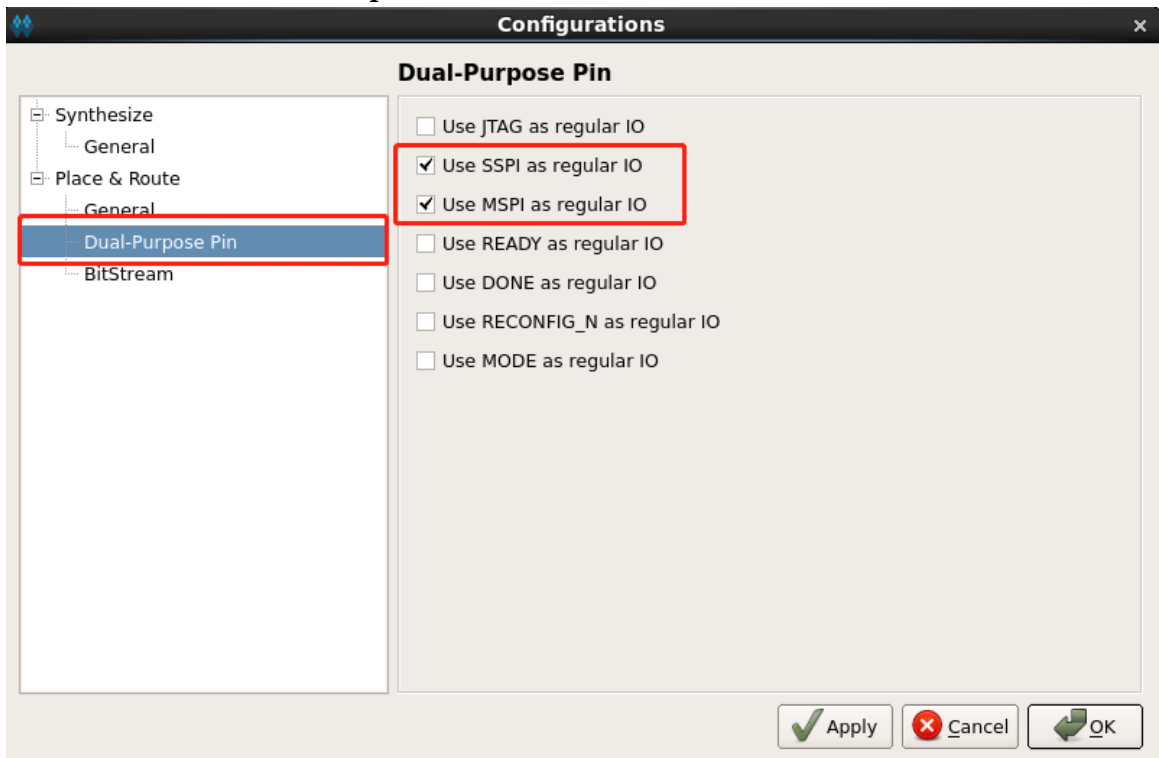
图 4-5 Boot Mode 配置



4.2.2 Dual-Purpose Pin 配置

Gowin_PicoRV32 硬件设计中，配置页面“Place & Route > Dual-Purpose Pin”配置 SSPI、MSPI 为通用端口，如图 4-6 所示。

图 4-6 配置 Dual-Purpose Pin



4.3 设计流程


1. Gowin_PicoRV32 硬件设计：
 - Boot Mode 选择 MCU boot from external Flash and run in ITCM 或 MCU boot and run in external Flash 方式
 - 输出具有片外 SPI-Flash 下载启动功能的码流文件；
2. Programmer 配置 Device configuration，下载码流文件；
3. Gowin_PicoRV32 软件编程设计：
 - MCU boot and run in external Flash:
定义 config.h 宏定义 `#define BUILD_MODE BUILD_XIP`
配置 sections_xip.lds 作为 Flash 链接器
 - MCU boot from external Flash and run in ITCM
定义 config.h 宏定义 `#define BUILD_MODE BUILD_BURN`
配置 sections.lds 作为 Flash 链接器
 - 编译输出二进制 BIN 文件
4. Programmer 配置 Device configuration，下载二进制 BIN 文件。

4.4 下载

Programmer 使用方法请参考 [SUG502](#)，Gowin Programmer 用户指南。

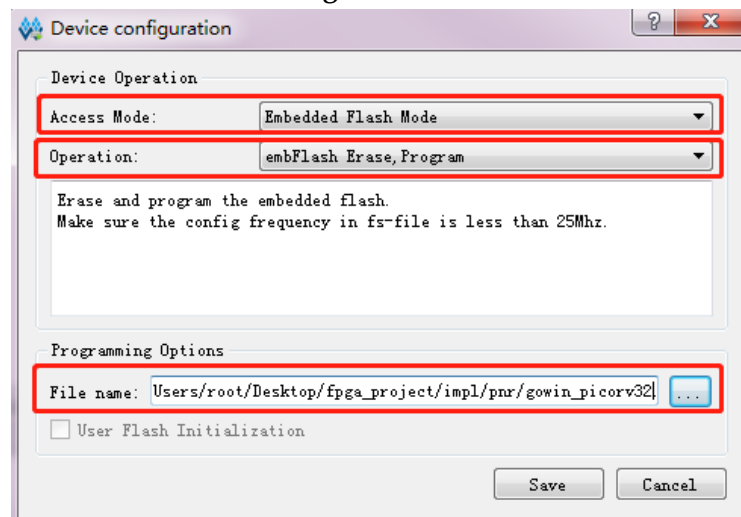
4.4.1 下载码流文件

Gowin_PicoRV32 硬件设计输出具有片外 SPI-Flash 下载启动功能的码流文件，使用 Programmer 下载。

运行 Gowin 云源软件的下载工具 Programmer，单击 Programmer 菜单栏“Edit > Configure Device”或工具栏 Configure Device “”，打开 Device configuration 对话框。

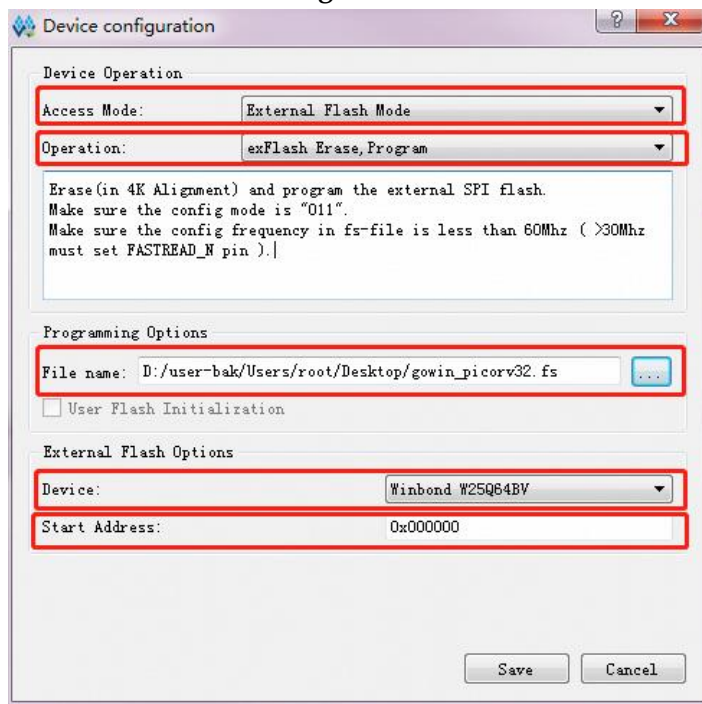
如果板载器件 GW1N-9/GW1NR-9 系列，“Access Mode”下拉列表中选择“Embedded Flash Mode”，“Operation”下拉列表中选择“embFlash Erase, Program”，“Programming Options 中”导入需要下载的码流文件，单击“Save”，如图 4-7 所示。


图 4-7 GW1N 系列 Configure Device



如果板载器件 GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C /GW2A-55/GW2A-55C, “Access Mode” 下拉列表中选择 “External Flash Mode”, “Operation” 下拉列表中选择 “exFlash Erase, Program”, “Programming Options” 中导入需要下载的码流文件, “External Flash Options” 中 “Device” 根据板载 Flash 芯片选择, 如 Winbond W25Q64BV, “Start Address” 选择 “0x000000”, 单击 “Save”, 如图 4-8 所示。


图 4-8 GW2A 系列 Configure Device



完成 Device configuration 后, 单击 Programmer 工具栏 Program/Configure “”, 完成码流下载。

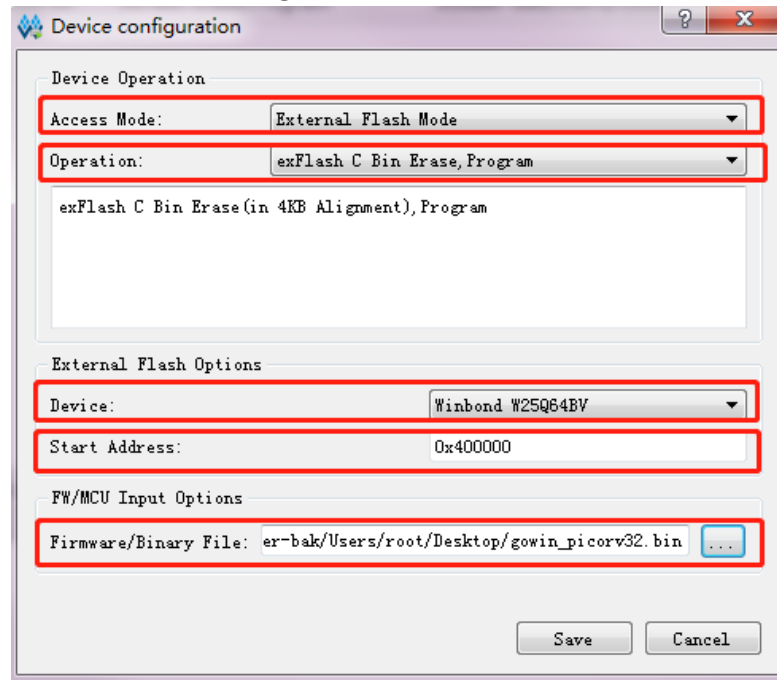
4.4.2 下载软件设计 BIN 文件

完成 Gowin_PicoRV32 软件编程设计输出 BIN 文件, 使用 Programmer 下载 Gowin_PicoRV32 软件编程设计二进制 BIN 文件。

GOWIN MCU Designer V1.0 及以上版本中, 单击菜单栏 “Run > Programmer” 或工具栏 Programmer “”, 打开 Programmer。

单击 Programmer 菜单栏 “Edit > Configure Device” 或工具栏 Configure Device “”, 打开 Device configuration 对话框, “Access Mode” 下拉列表选择 “External Flash Mode”, “Operation” 下拉列表选择 “exFlash C Bin Erase, Program”, “Firmware/Binary File” 中导入需要下载的 Gowin_PicoRV32 二进制 BIN 文件, “External Flash Options” 中 “Device” 根据板载 Flash 芯片类型选择, 如 Winbond W25Q64BV, “Start Address” 设置为 “0x400000”, 单击 “Save”, 如图 4-9 所示。

图 4-9 Device configuration



完成 Device configuration 后，单击 Programmer 工具栏 Program/Configure “”，完成 Gowin_PicoRV32 二进制 BIN 文件下载。

4.5 适用器件

- GW2A-18 系列
- GW2A-18C 系列
- GW2AR-18 系列
- GW2AR-18C 系列
- GW2A-55 系列
- GW2A-55C 系列

