



Gowin PicoRV32 软件下载 软件下载参考手册

IPUG913-1.5,2023-08-18

版权所有 © 2023 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin 小蜜蜂、LittleBee、晨熙以及高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、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

本文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。高云半导体对文档中包含的文字、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文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文档进行适时的更新。

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 memory 下载及运行;● MCU 支持外部设备 SPI-Flash memory 读、写和擦除功能;● MCU 支持 Hardware Stack Protection 和 Trap Stack Overflow 功能。
2020/06/01	1.2	<ul style="list-style-type: none">● 支持 MCU 软件在线调试功能;● 增强 MCU 内核中断处理功能;● 优化 MCU 内核指令;● 更新 mergebin 工具, 支持综合工具 GowinSynthesis®命名方式的解析。
2021/07/16	1.3	<ul style="list-style-type: none">● 删除综合工具 SynplifyPro;● 更新 FPGA 软件版本;● 更新支持器件。
2022/02/11	1.4	增加 makehex 和 mergebin 方法的参考设计。
2023/08/18	1.5	支持 Arora V FPGA 产品。

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 配置 BUILD_MODE	4
2.4.2 配置 Flash 链接器	4
2.4.3 配置输出文件格式	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 配置 ITCM	8
3.3.2 配置 Post-Place File	8
3.4 软件配置	9
3.4.1 配置 BUILD_MODE	9
3.4.2 配置 Flash 链接器	9
3.4.3 配置输出文件格式	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

4 片外 SPI-Flash Memory 下载方法.....	12
4.1 软件配置.....	12
4.1.1 配置 BUILD_MODE	12
4.1.2 配置 Flash 链接器	13
4.2 硬件配置.....	15
4.2.1 配置 ITCM.....	15
4.2.2 配置 Dual-Purpose Pin.....	16
4.3 设计流程.....	16
4.4 下载.....	17
4.4.1 下载硬件设计码流文件.....	17
4.4.2 下载软件编程设计 Binary 文件.....	19
4.5 适用器件.....	21

图目录

图 2-1 配置 ITCM.....	4
图 2-2 配置 BUILD_MODE	4
图 2-3 配置 Flash 链接器.....	5
图 3-1 配置 ITCM.....	8
图 3-2 配置布局布线 Post-Place File 选项	9
图 3-3 配置 BUILD_MODE	9
图 3-4 配置 Flash 链接器.....	10
图 3-5 合并软件编程设计输出和硬件设计输出	11
图 4-1 配置 BUILD_MODE“BUILD_BURN”.....	12
图 4-2 配置 BUILD_MODE“BUILD_XIP”	13
图 4-3 配置 Flash 链接器.....	14
图 4-4 配置 Flash 链接器.....	15
图 4-5 配置 ITCM.....	16
图 4-6 配置 Dual-Purpose Pin	16
图 4-7 晨熙家族 Device configuration	18
图 4-8 Arora V Device Configuration.....	19
图 4-9 晨熙家族 Device configuration	20
图 4-10 Arora V Device configuration	21

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. 软件编程设计产生映像文件，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`
 - 选择 sections.lds 作为 Flash 链接器
 - 编译产生软件编程设计 Binary 文件
 - b) 使用 makehex 工具，将软件编程设计 Binary 文件转换为十六进制映像文件 ram32.hex；
 - c) IP Core Generator 工具产生 Gowin_PicoRV32 IP 设计时，配置指令存储器 ITCM：
 - Boot Mode 选择 MCU boot and run in ITCM 方式
 - ITCM Initialiaztion File 引入 b).中 ram32.hex 文件作为 ITCM 的初始值
 - d) 综合、布局布线，产生包括软件编程设计和硬件设计的硬件设计码流文件；
 - e) 下载工具 Programmer，下载硬件设计码流文件；
 - f) 每次更新软件编程设计后，重新执行 a).~ e).。
2. 合并软件编程设计产生的软件编程设计 Binary 文件和硬件设计产生的硬件设计码流文件。
 - a) 参照应用场景需求，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`
 - 选择 sections.lds 作为 Flash 链接器

- 参照应用场景需求，更新用户软件应用设计
 - 编译产生软件编程设计 **Binary** 文件
 - c) 使用 **mergebin** 工具，合并软件编程设计 **Binary** 文件和 a) 中硬件设计码流文件；
 - d) 产生合并软件编程设计和硬件设计后的新的硬件设计码流文件；
 - e) 下载工具 **Programmer**，下载合并后的新的硬件设计码流文件；
 - f) 每次更新软件编程设计后，重新执行 b)~ e)。
3. 片外 **SPI-Flash Memory** 下载软件编程设计产生的软件编程设计 **Binary** 文件。
- a) **IPCore Generator** 工具产生 **Gowin_PicoRV32 IP** 设计时，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 IP** 设计产生具有片外 **SPI-Flash Memory** 下载启动功能的硬件设计码流文件；
 - c) 下载工具 **Programmer**，下载硬件设计码流文件；
 - 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**
选择 **sections_xip.lids** 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**
选择 **sections.lids** 作为 **Flash** 链接器
 - 编译产生软件编程设计 **Binary** 文件
 - e) 下载工具 **Programmer**，下载软件编程设计产生的软件编程设计 **Binary** 文件。

2 软件编程输出作为硬件 ITCM 初始值

2.1 软件工具

```
...\tool\makehex\bin\makehex32.exe  
...\tool\makehex\bin\makehex08.exe
```

注！

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

软件工具命令及参数：

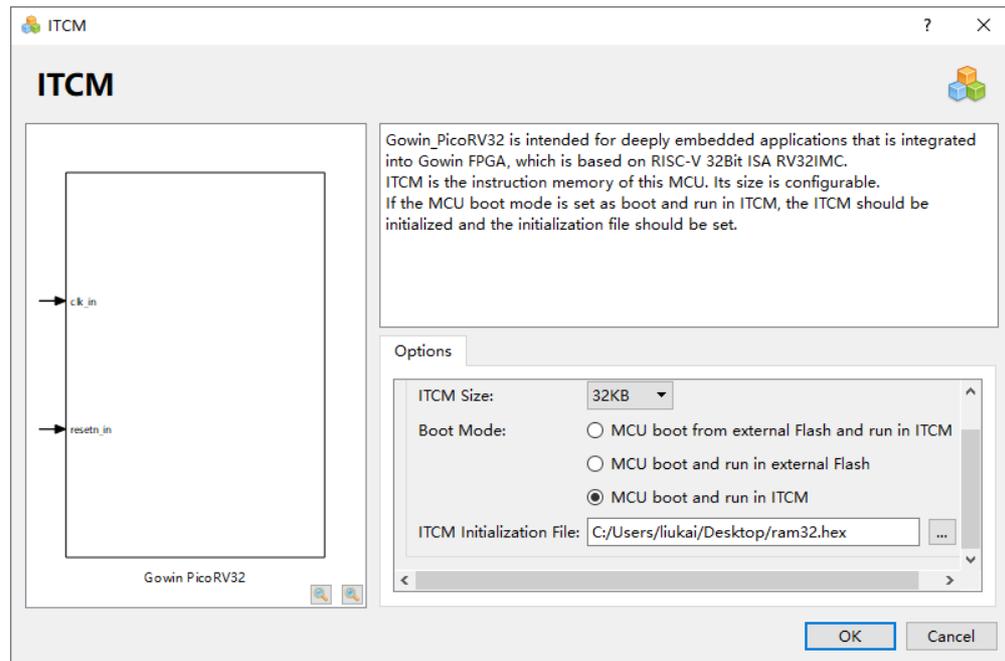
- makehex32.exe bin-file
- makehex08.exe bin-file

2.3 硬件配置

高云半导体云源®软件的 IP Core Generator 工具中：

- 选择“ITCM > Boot Mode > MCU boot and run in ITCM”选项；
- 在“ITCM > ITCM Initialization File”中引入软件编程设计映像文件“ram32.hex”，作为 ITCM 的初始值，如图 2-1 所示。

图 2-1 配置 ITCM

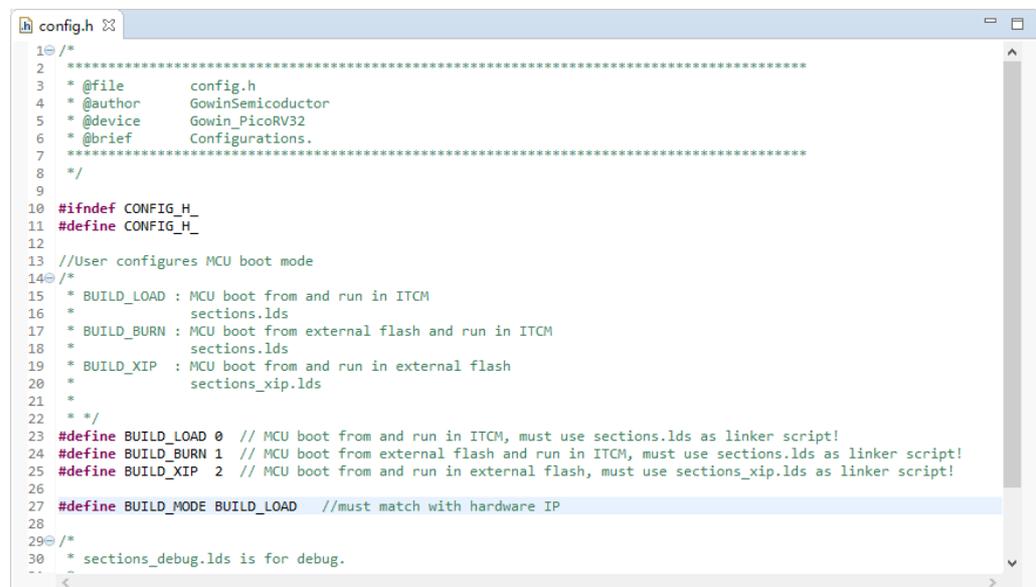


2.4 软件配置

2.4.1 配置 BUILD_MODE

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

图 2-2 配置 BUILD_MODE

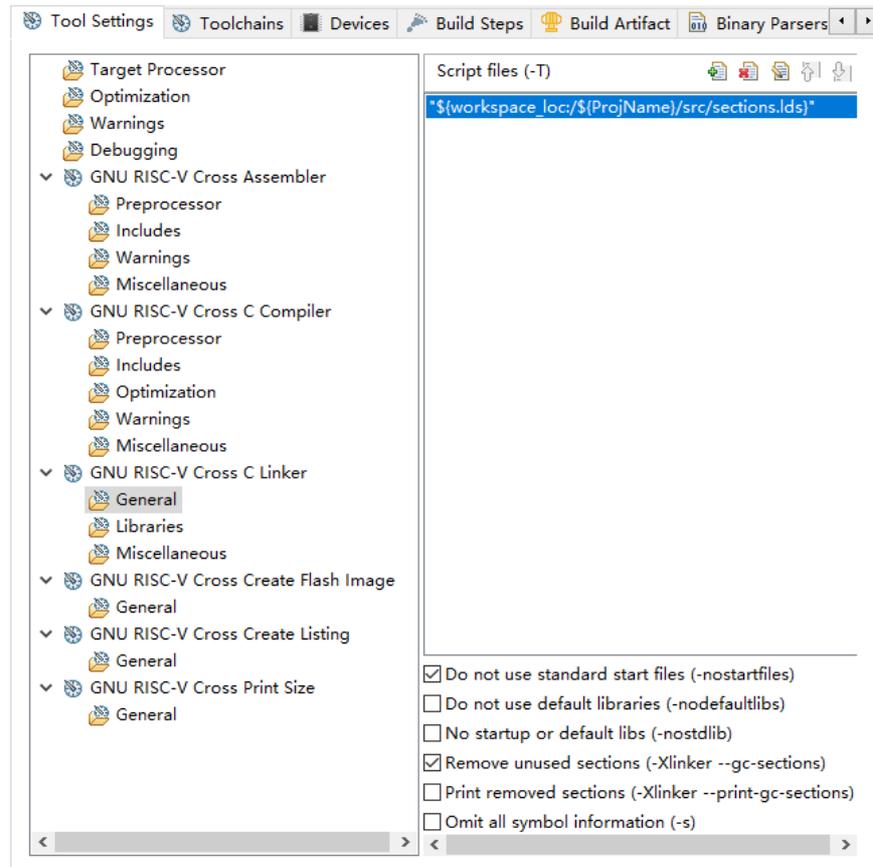


2.4.2 配置 Flash 链接器

GMD 软件“Project Explorer”视图中，选定当前软件工程，右击选择“Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General”，选择“sections.lds”作为 Flash 链接器，如图

2-3 所示。

图 2-3 配置 Flash 链接器



2.4.3 配置输出文件格式

编译 Gowin_PicoRV32 软件编程设计，产生软件编程设计 Binary 文件。

运行 `makehex32.exe bin-file` 命令，产生十六进制映像文件 `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`;
 - 选择 `section.lds` 作为 Flash 链接器；
 - 编译产生软件编程设计 Binary 文件；
 - 运行 `makehex32.exe` 工具，产生十六进制映像文件 `ram32.hex`，作为 Gowin_PicoRV32 IP 设计中 ITCM 的初始值。
2. Gowin_PicoRV32 硬件设计：
 - 选择 “ITCM > Boot Mode > MCU boot and run in ITCM” 选项；
 - 在 “ITCM > ITCM Initialization File” 中引入 `ram32.hex` 作为 ITCM 的初始值。
3. 产生 Gowin_PicoRV32 IP 设计，实例化 Gowin_PicoRV32 Top Module，

连接用户设计；

4. 物理约束和时序约束；
5. 使用综合工具 **GowinSynthesis**®综合，产生网表文件；
6. 使用布局布线工具 **Place & Route** 布局布线，产生包含软件编程设计的硬件设计码流文件；
7. 使用下载工具 **Programmer**，下载硬件设计码流文件；
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

- 小蜜蜂®（LittleBee®）家族 FPGA 产品
- 晨熙®（Arora）家族 FPGA 产品
- Arora V FPGA 产品

3 合并软件编程设计和硬件设计

3.1 软件工具

...tool\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

- 软件工具：mergebin.bat
- 软件工具命令及参数：


```
call posp_parse.exe posp-file itcm-size gwsyn
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 BSRAM 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。
gwsyn	指定综合工具 GowinSynthesis®。
merge_bit.exe	合并 Gowin_PicoRV32 硬件设计和软件编程设计。
bin-file	Gowin_PicoRV32 软件编程设计产生的软件编程设计 Binary 文件。
BramLoc.txt	posp_parse.exe 产生的 ITCM 布局位置信息文件。
fs-file	Gowin_PicoRV32 硬件设计产生的硬件设计码流文件。

合并软件编程设计产生的软件编程设计 Binary 文件与硬件设计产生的硬件设计码流文件，产生新的硬件设计码流文件。

mergebin.bat 在使用过程中,参照实际应用场景需求修改参数 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 IP 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。

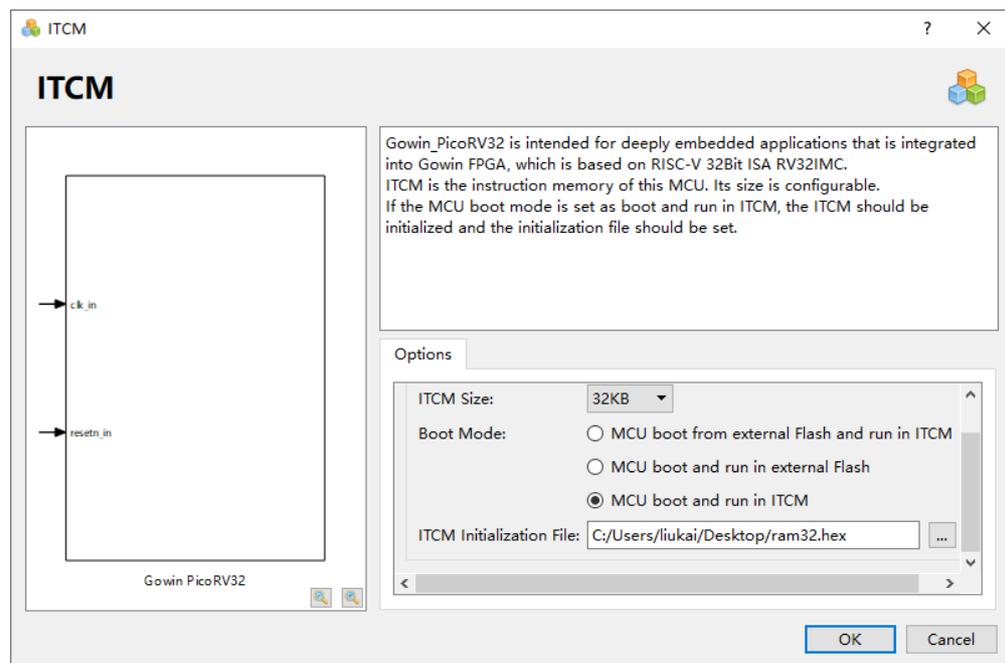
每次更新软件编程设计后,只需要每次使用 mergebin 工具合并上述硬件设计码流文件和每次更新的软件编程设计 Binary 文件。

3.3.1 配置 ITCM

云源软件的 IP Core Generator 工具中:

- 选择“ITCM > Boot Mode > MCU boot and run in ITCM”选项;
- 在“ITCM > ITCM Initialization File”中引入软件编程设计产生的映像文件“ram32.hex”,作为 ITCM 的初始值,如图 3-1 所示。

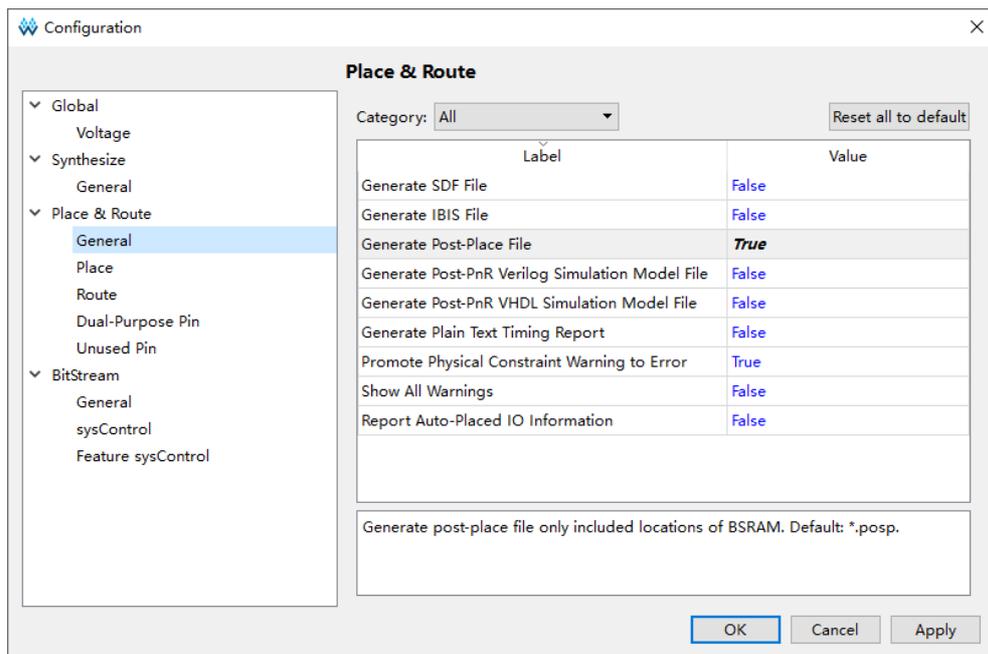
图 3-1 配置 ITCM



3.3.2 配置 Post-Place File

云源软件配置选项中,“Place & Route > General > Generate Post-Place File”选项设置为“True”,产生 Post-Place File,作为 posp_parse.exe 参数的 posp 输入文件,如图 3-2 所示。

图 3-2 配置布局布线 Post-Place File 选项

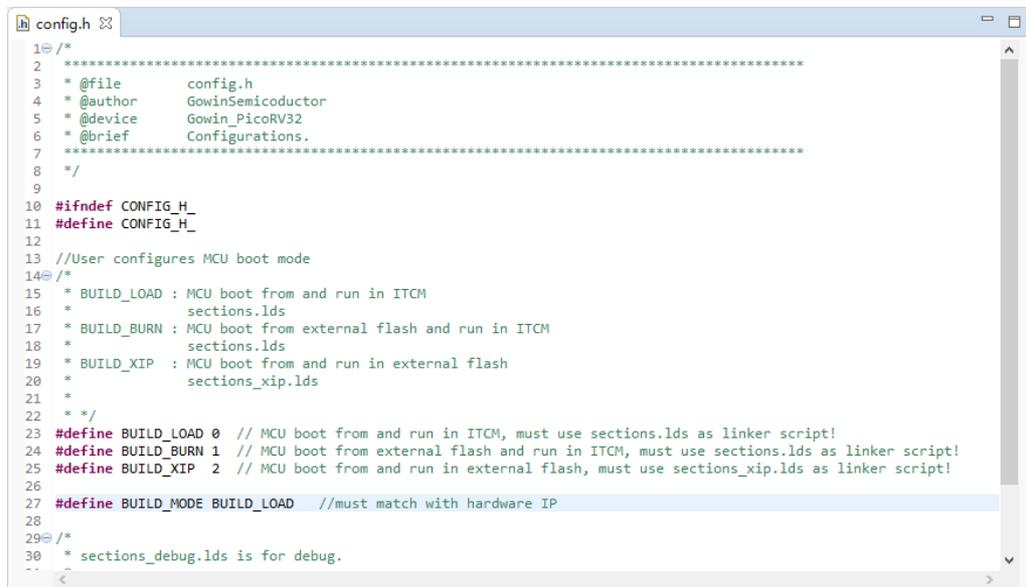


3.4 软件配置

3.4.1 配置 BUILD_MODE

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

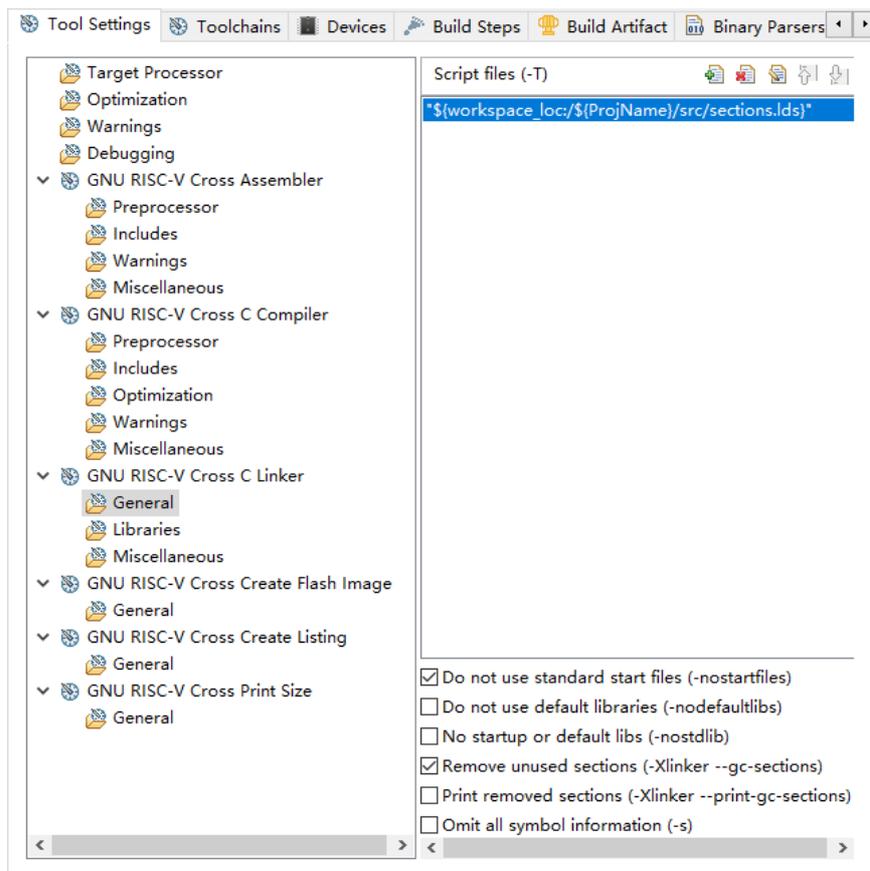
图 3-3 配置 BUILD_MODE



3.4.2 配置 Flash 链接器

GMD 软件 “Project Explorer” 视图中, 选定当前软件工程, 右击选择 “Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General”, 选择 sections.lds 作为 Flash 链接器, 如图 3-4 所示。

图 3-4 配置 Flash 链接器



3.4.3 配置输出文件格式

编译 Gowin_PicoRV32 软件编程设计, 产生软件编程设计 Binary 文件。

3.5 设计流程

3.5.1 合并

1. 参照应用场景需求, 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
 - 选择 sections.lds 作为 Flash 链接器
 - 参照应用场景需求, 更新用户软件应用设计
 - 编译产生软件编程设计 Binary 文件
3. 参照实际应用场景需求, 修改 mergebin.bat, 执行 mergebin.bat, 合并硬件设计产生的硬件设计码流文件和软件编程设计产生的软件编程设计 Binary 文件, 产生新的硬件设计码流文件, 如图 3-5 所示;
4. 每次更新软件编程设计后, 重新执行 3~4。

图 3-5 合并软件编程设计输出和硬件设计输出

```
----- 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 下载

完成合并后, 使用下载工具 Programmer, 下载新的硬件设计码流文件。

下载工具 Programmer 的使用方法, 请参考 [SUG502, Gowin Programmer 用户指南](#)。

3.6 适用器件

晨熙® (Arora) 家族 FPGA 产品

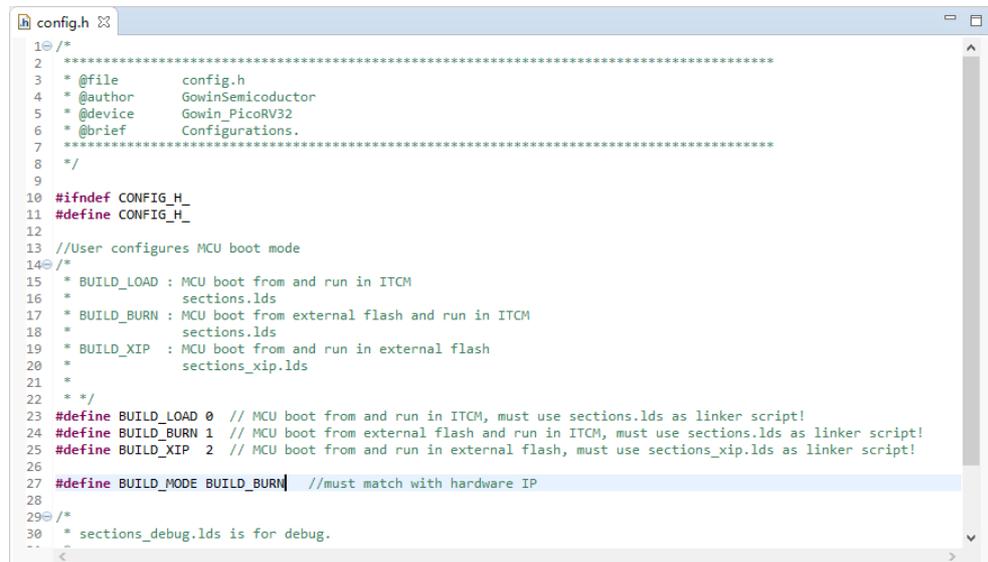
4 片外 SPI-Flash Memory 下载方法

4.1 软件配置

4.1.1 配置 BUILD_MODE

如果 Gowin_PicoRV32 IP 设计中，选定“ITCM > Boot Mode > MCU boot from external Flash and run in ITCM”选项，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。

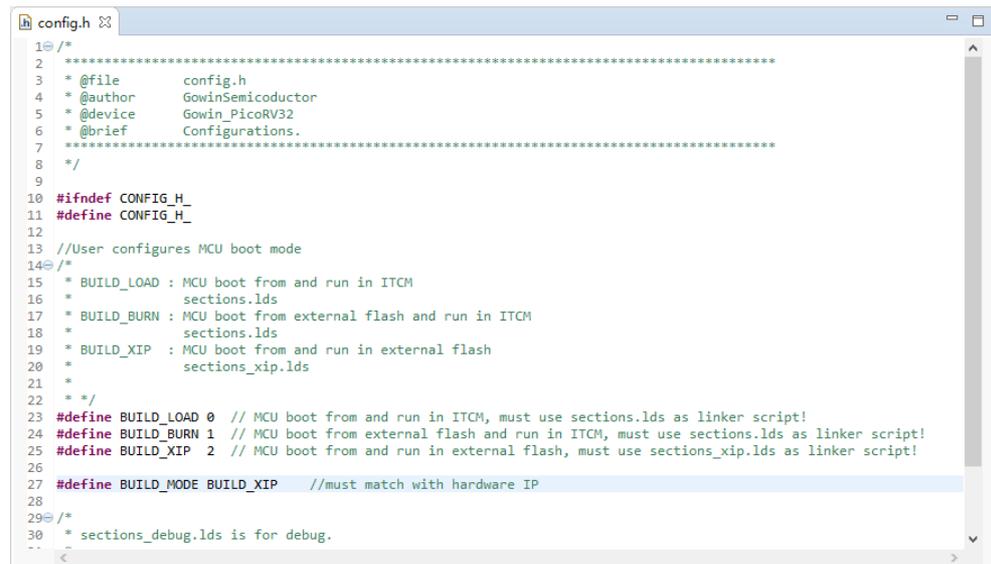
图 4-1 配置 BUILD_MODE “BUILD_BURN”



```
1  /*
2  *****
3  * @file      config.h
4  * @author    GowinSemiconductor
5  * @device    Gowin_PicoRV32
6  * @brief     Configurations.
7  *****
8  */
9
10 #ifndef CONFIG_H_
11 #define CONFIG_H_
12
13 //User configures MCU boot mode
14 /*
15 * BUILD_LOAD : MCU boot from and run in ITCM
16 *               sections.lds
17 * BUILD_BURN : MCU boot from external flash and run in ITCM
18 *               sections.lds
19 * BUILD_XIP  : MCU boot from and run in external flash
20 *               sections_xip.lds
21 *
22 */
23 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
24 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
25 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
26
27 #define BUILD_MODE BUILD_BURN //must match with hardware IP
28
29 /*
30 * sections_debug.lds is for debug.
31 */
```

如果 Gowin_PicoRV32 IP 设计中，选定“ITCM > Boot Mode > MCU boot and run in external Flash”选项，则 Gowin_PicoRV32 软件编程设计 config.h 中，定义#define BUILD_MODE BUILD_XIP，如图 4-2 所示。

图 4-2 配置 BUILD_MODE “BUILD_XIP”



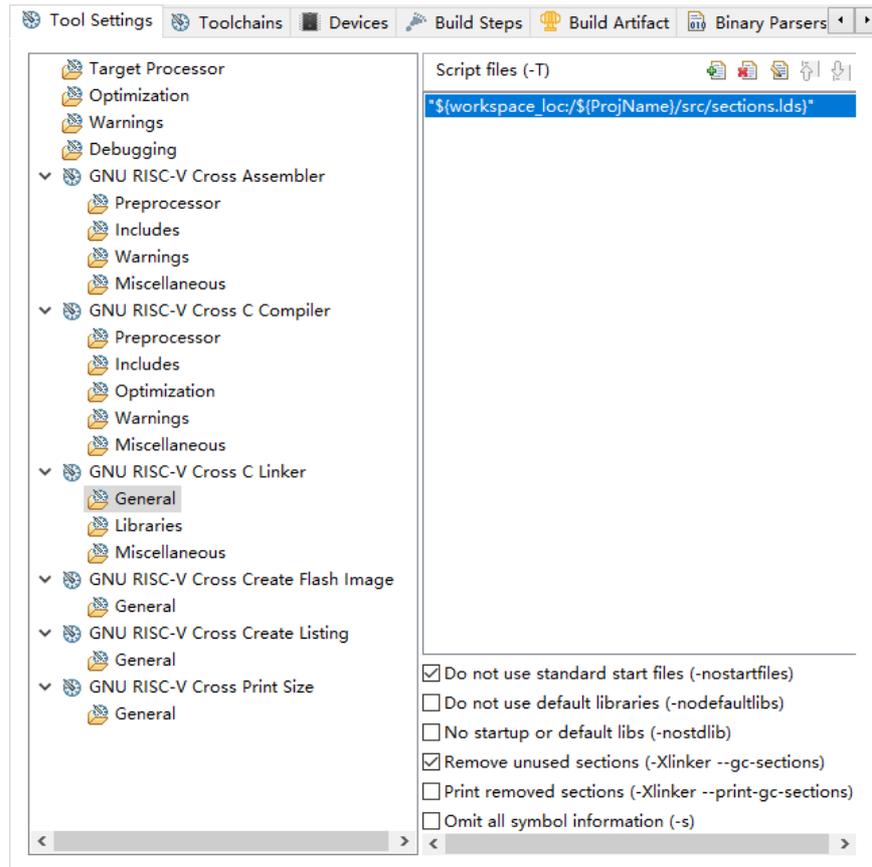
```
1  /*
2  *
3  * @file      config.h
4  * @author    GowinSemiconductor
5  * @device    Gowin_PicoRV32
6  * @brief     Configurations.
7  *
8  */
9
10 #ifndef CONFIG_H_
11 #define CONFIG_H_
12
13 //User configures MCU boot mode
14 /*
15 * BUILD_LOAD : MCU boot from and run in ITCM
16 *               sections.lds
17 * BUILD_BURN : MCU boot from external flash and run in ITCM
18 *               sections.lds
19 * BUILD_XIP  : MCU boot from and run in external flash
20 *               sections_xip.lds
21 *
22 */
23 #define BUILD_LOAD 0 // MCU boot from and run in ITCM, must use sections.lds as linker script!
24 #define BUILD_BURN 1 // MCU boot from external flash and run in ITCM, must use sections.lds as linker script!
25 #define BUILD_XIP 2 // MCU boot from and run in external flash, must use sections_xip.lds as linker script!
26
27 #define BUILD_MODE BUILD_XIP //must match with hardware IP
28
29 /*
30 * sections_debug.lds is for debug.
31 */
```

4.1.2 配置 Flash 链接器

如果 Gowin_PicoRV32 IP 设计中，选定“ITCM > Boot Mode > MCU boot from external Flash and run in ITCM”选项，则 GMD 软件“Project Explorer”视图中，选定当前软件工程，右击选择“Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General”，选择“sections.lds”作为 Flash 链接器，如图 4-3 所示。

例如，“\${workspace_loc}/\${ProjName}/src/sections.lds”。

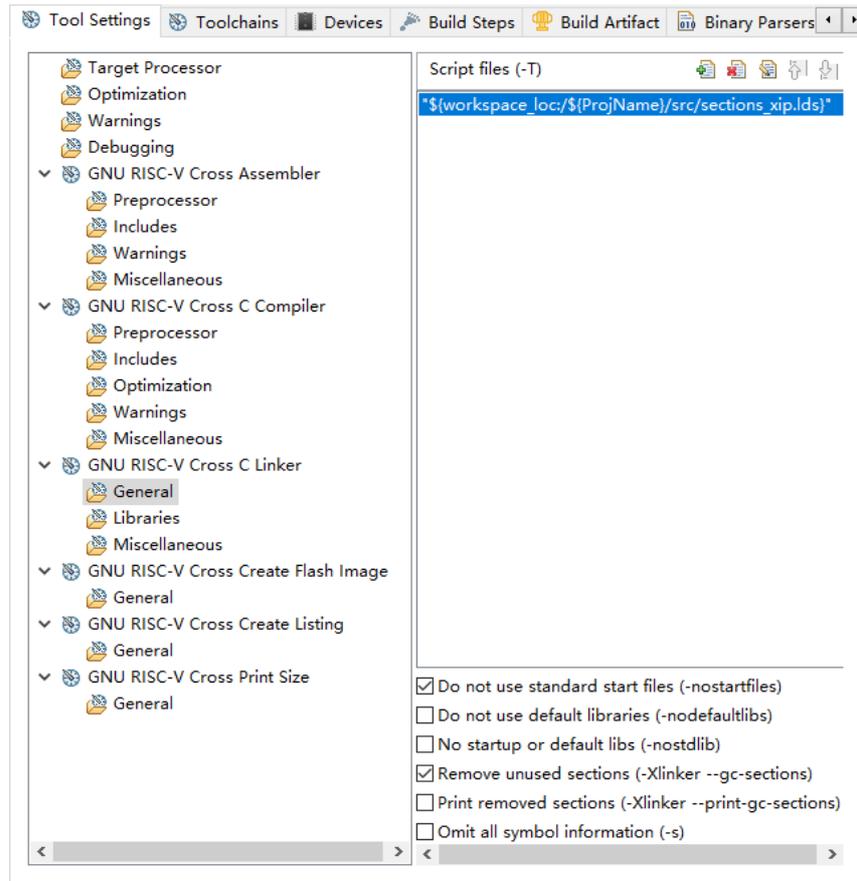
图 4-3 配置 Flash 链接器



如果 Gowin_PicoRV32 IP 设计中，选定“ITCM > Boot Mode > MCU boot and run in external Flash”选项，则 GMD 软件“Project Explorer”视图中，选定当前软件工程，右击选择“Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General”，选择“sections_xip.lds”作为 Flash 链接器，如图 4-4 所示。

例如，“`*${workspace_loc}/${ProjName}/src/sections_xip.lds*`”。

图 4-4 配置 Flash 链接器

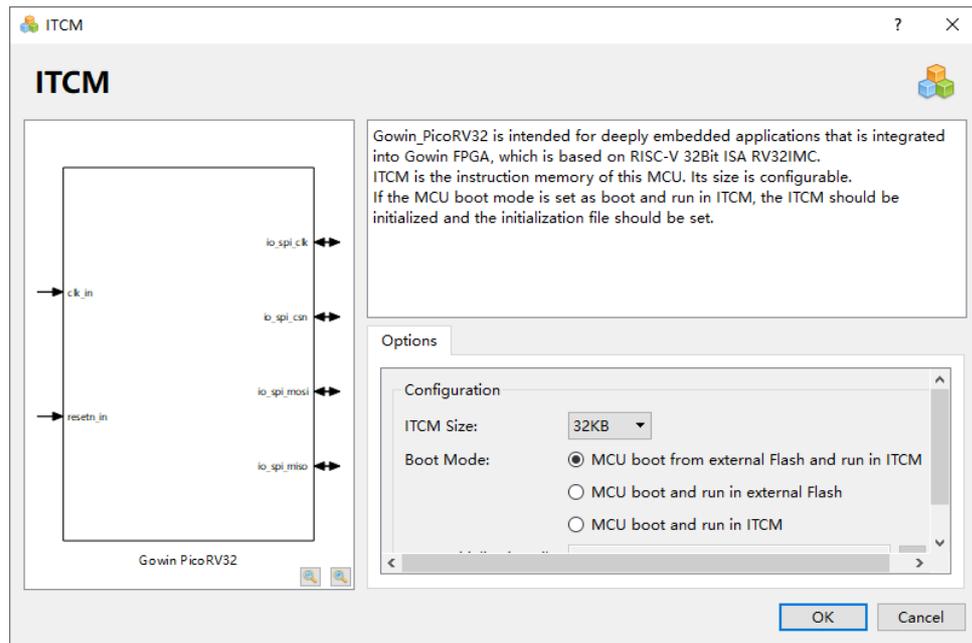


4.2 硬件配置

4.2.1 配置 ITCM

云源软件的 IP Core Generator 工具中选择“ITCM > Boot Mode > MCU boot from external Flash and run in ITCM”或“MCU boot and run in external Flash”选项，如图 4-5 所示。

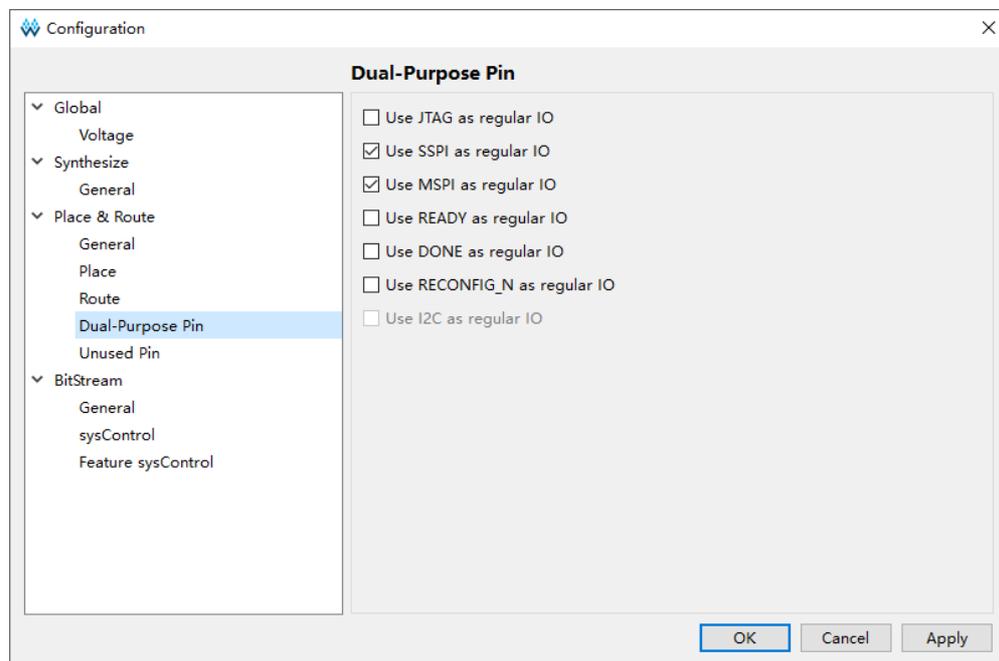
图 4-5 配置 ITCM



4.2.2 配置 Dual-Purpose Pin

在云源软件配置选项 “Place & Route > Dual-Purpose Pin” 中，配置 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：
 - 选择“Boot Mode > MCU boot from external Flash and run in ITCM”或“MCU boot and run in external Flash”选项

- 产生 Gowin_PicoRV32 IP 设计
 - 综合、布局布线，产生具有片外 SPI-Flash Memory 下载功能的硬件设计码流文件；
2. 下载工具 Programmer，配置 Device configuration，下载硬件设计码流文件；
 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
选择 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
选择 sections.lds 作为 Flash 链接器
 - 编译产生软件编程设计 Binary 文件
 4. 下载工具 Programmer，配置 Device configuration，下载软件编程设计 Binary 文件。

4.4 下载

下载工具 Programmer 的使用方法，请参考 [SUG502, Gowin Programmer 用户指南](#)。

4.4.1 下载硬件设计码流文件

Gowin_PicoRV32 硬件设计产生具有片外 SPI-Flash Memory 下载功能的硬件设计码流文件，使用下载工具 Programmer 下载硬件设计码流文件。

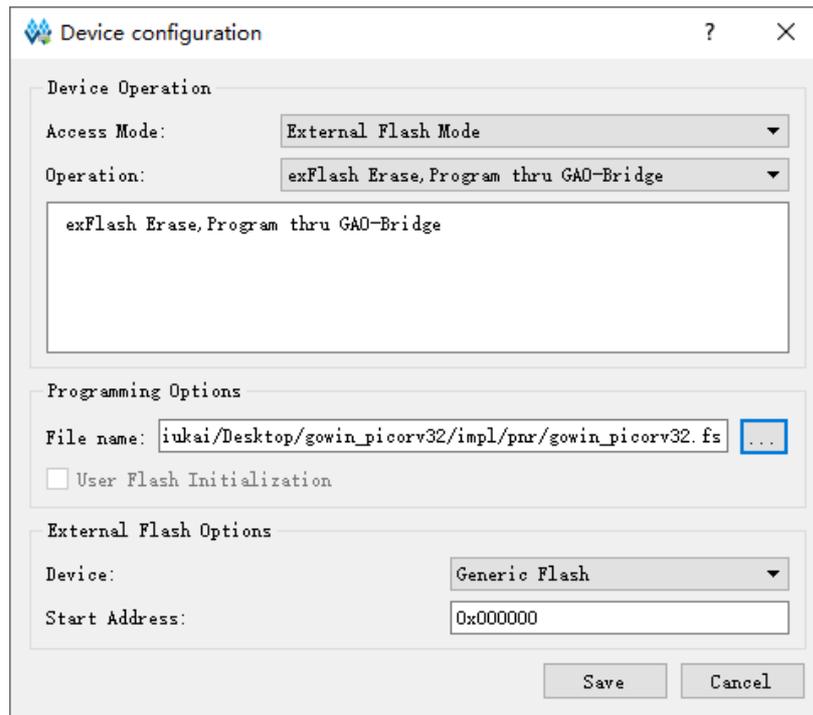
选择云源软件菜单栏“Tools > Programmer”或工具栏“Programmer” ()，打开下载工具 Programmer。

选择 Programmer 菜单栏“Edit > Configure Device”或工具栏“Configure Device” ()，打开 Device configuration。

如果开发板板载晨曦家族 FPGA 产品，下载选项配置，如图 4-7 所示。

- “Access Mode”下拉列表，选择“External Flash Mode”选项；
 - “Operation”下拉列表，选择“exFlash Erase, Program thru GAO-Bridge”或“exFlash Erase, Program, Verify thru GAO-Bridge”选项；
 - “Programming Options > File name”选项，引入需要下载的硬件设计码流文件；
 - “External Flash Options > Device”选项，选择“Generic Flash”；
 - “External Flash Options > Start Address”选项，设置为“0x000000”；
- 单击“Save”，完成硬件设计码流文件下载选项配置。

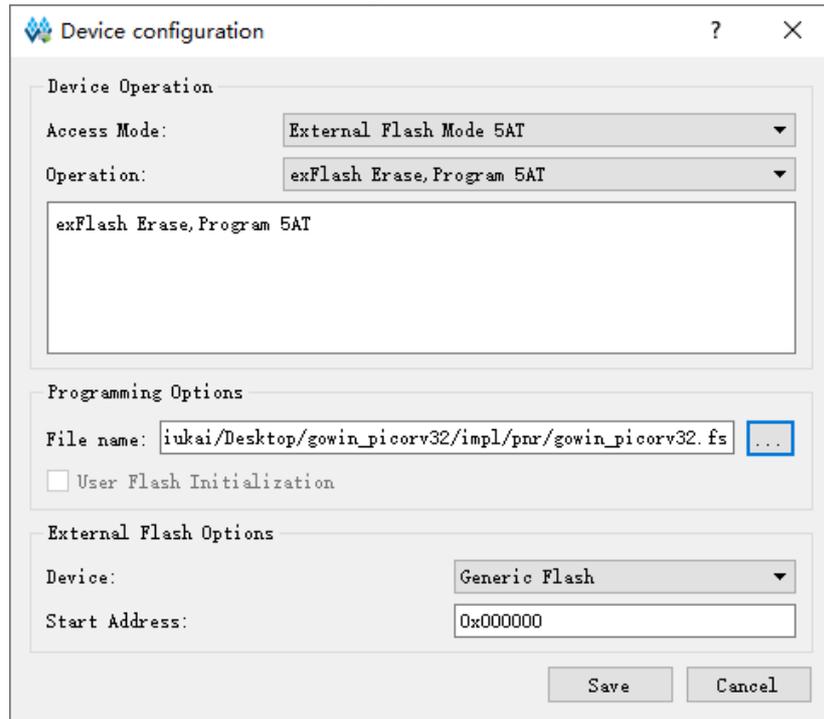
图 4-7 晨熙家族 Device configuration



如果开发板板载 Arora V FPGA 产品，下载选项配置如图 4-8 所示。

- “Access Mode” 下拉列表，选择 “External Flash Mode 5AT” 选项；
 - “Operation” 下拉列表，选择 “exFlash Erase, Program 5AT” 或 “exFlash Erase, Program, Verify 5AT” 选项；
 - “Programming Options > File name” 选项，引入需要下载的硬件设计码流文件；
 - “External Flash Options > Device” 选项，选择 “Generic Flash”；
 - “External Flash Options > Start Address” 选项，设置为 0x000000；
- 单击 “Save”，完成硬件设计码流文件下载选项配置。

图 4-8 Arora V Device Configuration



完成 Device configuration 后，单击 Programmer 工具栏“Program/Configure”（），下载硬件设计码流文件。

4.4.2 下载软件编程设计 Binary 文件

完成 Gowin_PicoRV32 软件编程设计，产生软件编程设计 Binary 文件，使用下载工具 Programmer，下载软件编程设计 Binary 文件。

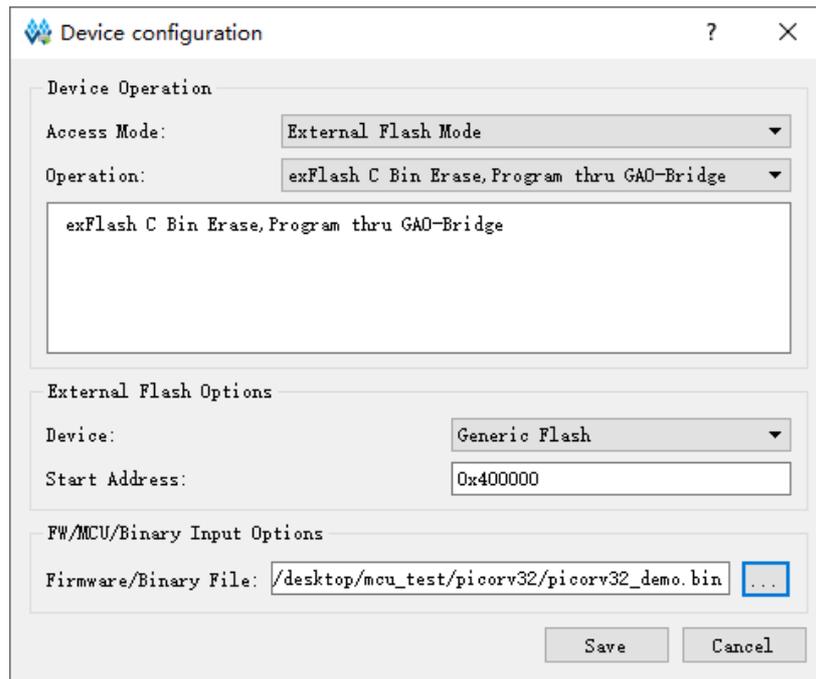
GMD 软件中，单击菜单栏“Run > Programmer”或工具栏“Programmer”（），打开 Programmer。

单击 Programmer 菜单栏“Edit > Configure Device”或工具栏“Configure Device”（），打开 Device configuration。

如果开发板板载晨曦家族 FPGA 产品，下载选项配置，如图 4-9 所示。

- “Access Mode” 下拉列表，选择“External Flash Mode”选项；
 - “Operation” 下拉列表，选择“exFlash C Bin Erase, Program thru GAO-Bridge”或“exFlash C Bin Erase, Program, Verify thru GAO-Bridge”选项；
 - “FW/MCU/Binary Input Options > Firmware/Binary File”选项，引入需要下载的软件编程设计 Binary 文件；
 - “External Flash Options > Device”选项，选择“Generic Flash”；
 - “External Flash Options > Start Address”选项，设置为“0x400000”；
- 单击“Save”，完成软件编程设计 Binary 文件下载选项配置。

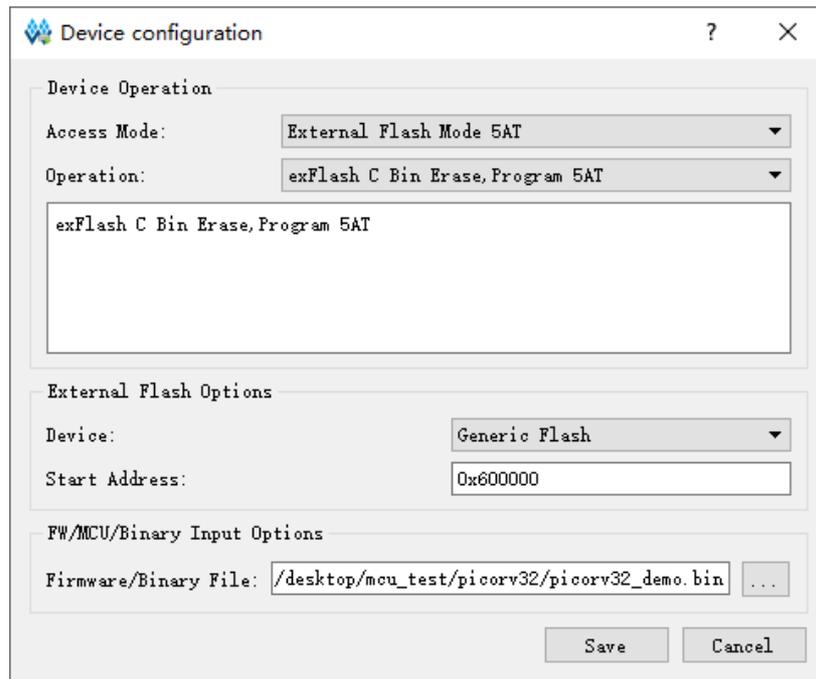
图 4-9 晨熙家族 Device configuration



如果开发板板载 Arora V FPGA 产品，下载选项配置，如图 4-10 所示。

- “Access Mode” 下拉列表，选择 “External Flash Mode 5AT” 选项；
 - “Operation” 下拉列表，选择 “exFlash C Bin Erase, Program 5AT” 或 “exFlash C Bin Erase, Program, Verify 5AT” 选项；
 - “FW/MCU/Binary Input Options > Firmware/Binary File” 选项，引入需要下载的软件编程设计 Binary 文件；
 - “External Flash Options > Device” 选项，选择 “Generic Flash”；
 - “External Flash Options > Start Address” 选项，设置为 “0x600000”；
- 单击 “Save”，完成软件编程设计 Binary 文件下载选项配置。

图 4-10 Arora V Device configuration



完成 Device configuration 后，单击 Programmer 工具栏
“Program/Configure” ()，下载软件编程设计 Binary 文件。

4.5 适用器件

- 晨熙® (Arora) 家族 FPGA 产品
- Arora V FPGA 产品

