



Gowin_EMPU_M1 下载 参考手册

IPUG532-1.4,2020-01-16

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
2019/02/19	1.0	初始版本。
2019/07/18	1.1	<ul style="list-style-type: none">● MCU 支持硬件设计与软件编程设计自动化合并工具；● MCU 支持片外 SPI-Flash 下载启动方式。
2019/08/18	1.2	<ul style="list-style-type: none">● MCU 硬件设计与软件编程设计支持扩展外部设备 DDR3 Memory；● 修复已知 ITCM、DTCM Size 和 IDE 问题。
2019/09/27	1.3	完善软件配置描述。
2020/01/16	1.4	<ul style="list-style-type: none">● MCU 硬件设计与软件编程设计支持外部设备 PSRAM；● 更新 MCU 编译软件 GMD V1.0；● 更新 RTOS 参考设计；● 增加 AHB2 和 APB2 扩展总线接口硬件和软件参考设计。

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	2
2.1 软件工具	2
2.2 命令参数	2
2.3 软件配置	2
2.4 硬件配置	3
2.5 设计流程	4
2.6 适用器件	4
2.7 参考设计	4
3 合并软件编程设计和硬件设计	6
3.1 软件工具	6
3.2 命令参数	6
3.3 硬件配置	7
3.4 设计流程	7
3.4.1 合并	7
3.4.2 下载	8
3.5 适用器件	8
3.6 适用软件	8
3.7 参考设计	8
4 片外 SPI-Flash 下载方法	9
4.1 软件配置	9
4.2 硬件配置	10
4.2.1 ITCM Initialization 配置	10
4.2.2 Dual-Purpose Pin 配置	10

4.3 设计流程.....	11
4.4 下载.....	11
4.4.1 下载硬件设计码流文件.....	11
4.4.2 下载软件设计 BIN 文件.....	12
4.5 适用器件.....	13
4.6 参考设计.....	13

图目录

图 2-1 外部脚本调用	3
图 2-2 配置 ITCM Initialization	4
图 3-1 配置布局布线 posp 选项	7
图 3-2 合并软件编程设计输出和硬件设计输出	8
图 4-1 ROM 起始地址和容量配置	9
图 4-2 配置 ITCM Initialization Path	10
图 4-3 配置 Dual-Purpose Pin	11
图 4-4 配置 Programmer 码流下载模式	12
图 4-5 配置 Programmer C-Bin 下载模式	13

表目录

表 3-1 merge_bit 命令及参数	6
-----------------------------	---

1 下载方法

Gowin_EMPU_M1 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 ITCM 初始值。
 - a). Gowin_EMPU_M1 软件编程设计输出二进制 BIN 文件；
 - b). 使用 make_hex 工具将二进制 BIN 文件转换为四个十六进制格式映像文件 itcm0、itcm1、itcm2 和 itcm3；
 - c). itcm0、itcm1、itcm2 和 itcm3 作为硬件设计 ITCM 的初始值文件读入；
 - 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。
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). Gowin_EMPU_M1 硬件设计输出码流文件；
 - b). Gowin_EMPU_M1 软件编程设计输出二进制 BIN 文件；
 - c). 使用 merge_bit 工具合并二进制 BIN 文件和码流文件；
 - 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。
3. 片外 SPI-Flash 下载软件编程设计输出的二进制 BIN 文件。
 - a). bootload 的 itcm0、itcm1、itcm2 和 itcm3 作为硬件设计中 ITCM 初始值读入；
 - b). Gowin_EMPU_M1 硬件设计输出具有片外 SPI-Flash 下载功能的码流文件；
 - c). Programmer 下载硬件设计输出的码流文件；
 - d). Gowin_EMPU_M1 软件编程设计输出二进制 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_EMPU_M1\script\make_hex_script\linux\make_hex\bin\make_hex
- Windows:
Gowin_EMPU_M1\script\make_hex_script\windows\make_hex\bin\make_hex.exe

2.2 命令参数

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

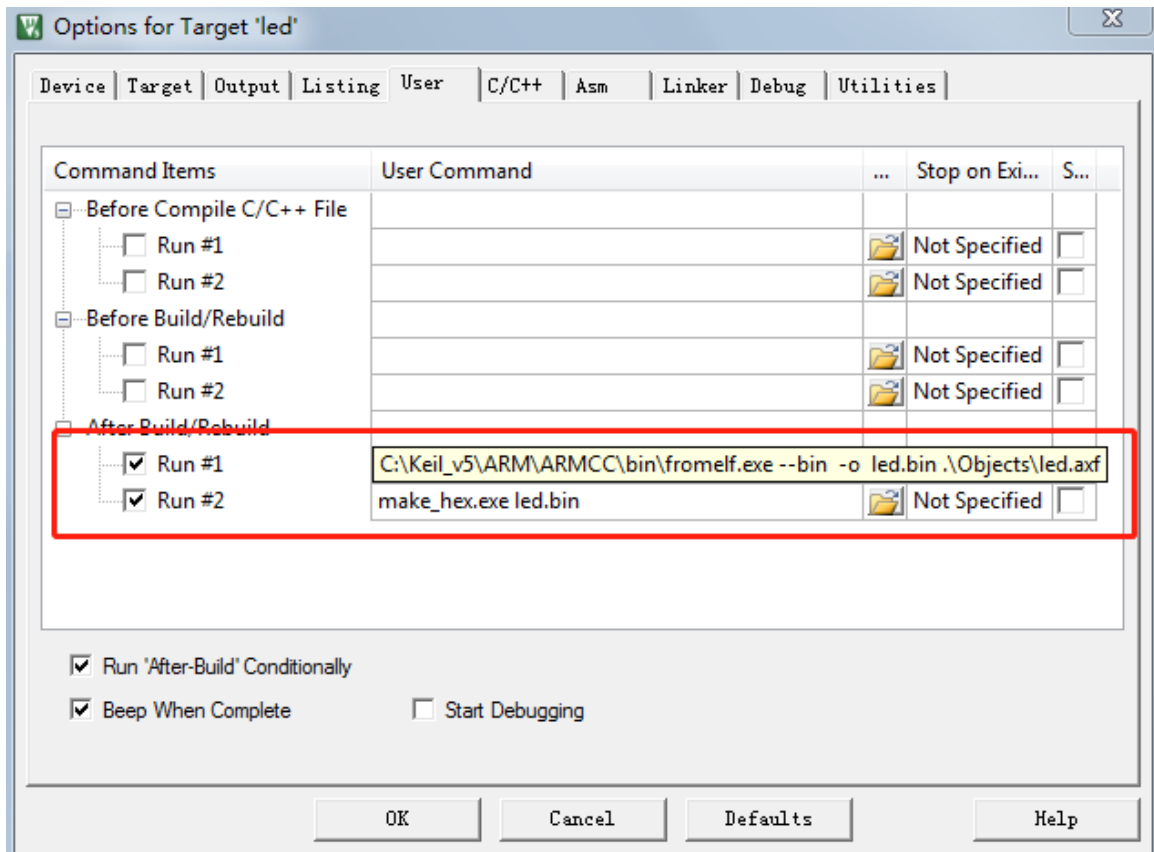
2.3 软件配置

软件编程设计输出二进制 BIN 文件，使用 make_hex 工具将二进制 BIN 文件转换为四个十六进制格式的映像文件 itcm0、itcm1、itcm2 和 itcm3。

ARM Keil Microcontroller Tool 中可以配置外部脚本，在编译时自动调用 make_hex.exe 产生十六进制格式的映像文件，如图 2-1 所示。

- Run #1
fromelf.exe --bin -o bin-file axf-file
- Run #2
make_hex.exe bin-file

图 2-1 外部脚本调用

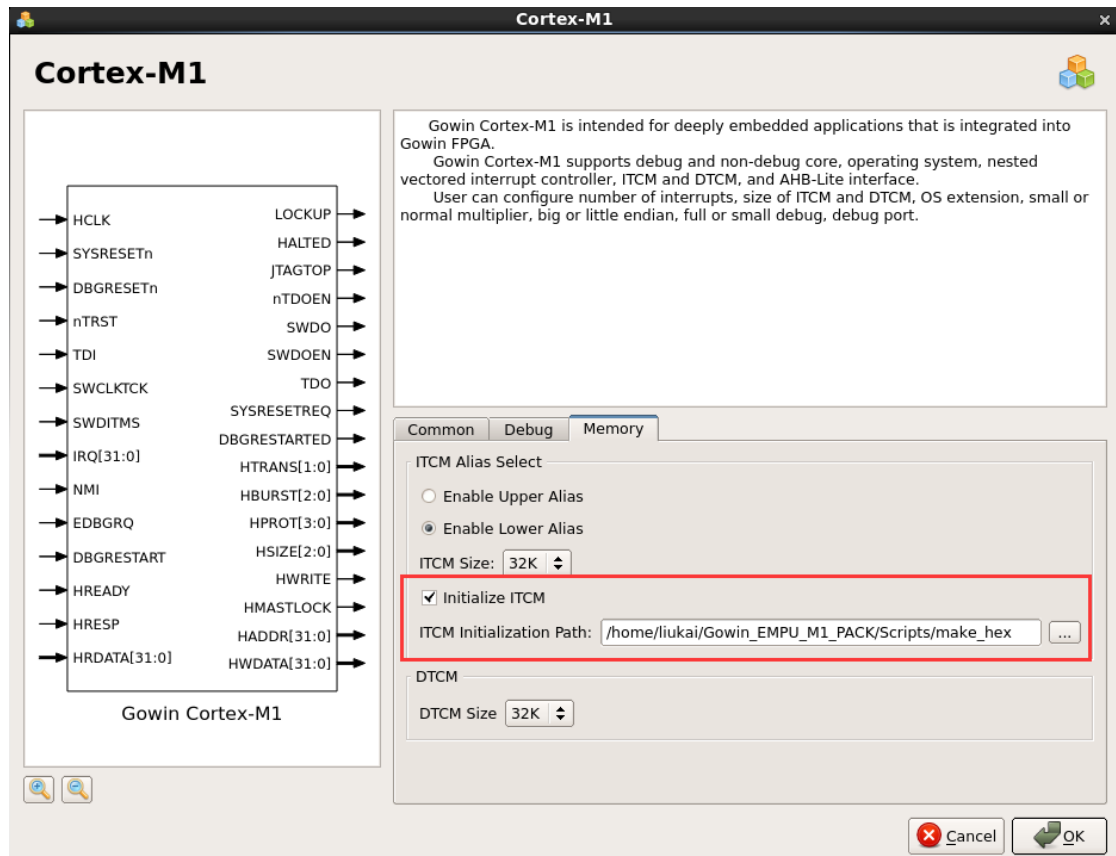


2.4 硬件配置

IP Core Generator 中配置 Cortex-M1 时，选择“Initialize ITCM”，在“ITCM Initialization Path”选项中，导入 itcm0、itcm1、itcm2、itcm3 四个十六进制映像文件所在的路径，作为 ITCM 初始值的路径，如图 2-2 所示。

导入 itcm0、itcm1、itcm2、itcm3 作为 ITCM 初始值，以及 IP Core Generator 中完成其他 Cortex-M1 和 AHB/APB 外部设备选项配置后，产生的 Gowin_EMPU_M1 硬件设计即可包含软件编程设计输出。

图 2-2 配置 ITCM Initialization



2.5 设计流程

1. ARM Keil MDK 或 GOWIN MCU Designer 软件编程设计，编译输出十六进制映像文件 itcm0、itcm1、itcm2 和 itcm3；
2. Gowin 云源软件 IP Core Generator 配置产生 Gowin_EMPU_M1 硬件设计，itcm0、itcm1、itcm2 和 itcm3 作为硬件设计中 ITCM 的初始值；
3. 实例化 Gowin_EMPU_M1，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。

2.6 适用器件

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

2.7 参考设计

- Linux:
Gowin_EMPU_M1\script\make_hex_script\linux\make_hex\example

- Windows:
Gowin_EMPU_M1\script\make_hex_script\windows\make_hex\example

3 合并软件编程设计和硬件设计

3.1 软件工具

- Linux:
Gowin_EMPU_M1\script\merge_bit_script\linux\merge_bit\bin\merge_bit.sh
- Windows:
Gowin_EMPU_M1\script\merge_bit_script\windows\merge_bit\bin\merge_bit.bat

3.2 命令参数

- Linux: bash merge_bit.sh
- Windows: merge_bit.bat

以 merge_bit.bat 为例，描述软件工具命令及参数。

```
call make_loc.exe -i posp-file -s itcm's size [-d]
```

```
call merge_bit.exe bin-file itcm.loc fs-file
```

其中命令及参数描述如表 3-1 所示。

表 3-1 merge_bit 命令及参数

参数	描述
make_loc.exe	输入 posp-file，产生 ITCM 布局信息 itcm.loc 文件
-i	Gowin_EMPU_M1 硬件设计配置 Place&Route 选项“Generate Post-Place File”产生的 posp 文件
-s	根据 Gowin_EMPU_M1 硬件设计中配置的 ITCM Size 设定
-d	可选项 如果配置 Enable Debug，则使能-d；如果 Disable Debug，则禁用 -d
merge_bit.exe	合并 Gowin_EMPU_M1 硬件设计和软件编程设计
bin-file	Gowin_EMPU_M1 软件编程设计输出的二进制 BIN 文件
itcm.loc	make_loc.exe 产生的 ITCM 布局位置信息 itcm.loc 文件
fs-file	Gowin_EMPU_M1 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，

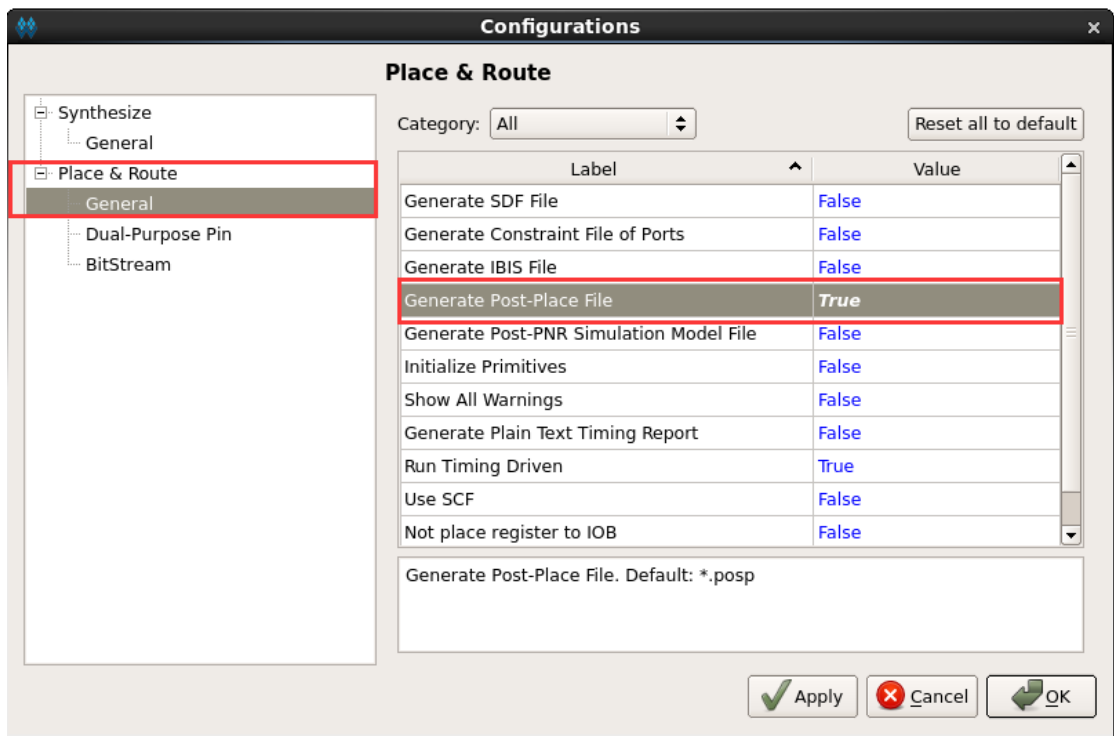
输出新的码流文件。

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

3.3 硬件配置

Gowin_EMPU_M1 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 `make_loc.exe -i` 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 设计流程

3.4.1 合并

1. Gowin_EMPU_M1 硬件设计输出码流文件和 `posp` 文件；
2. Gowin_EMPU_M1 软件编程设计输出二进制 `BIN` 文件；
3. Linux 环境执行 `merge_bit.sh` 或 Windows 环境执行 `merge_bit.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 3-2 所示。

图 3-2 合并软件编程设计输出和硬件设计输出

```

----- GOWIN Bin2FS -----
Read bit stream file gowin_empu_m1.fs ...
Build bsram init value fusemap...
Location file location.txt reading...
Bsram R28[9] init value convert to fusemap success.
Bsram R28[8] init value convert to fusemap success.
Bsram R28[7] init value convert to fusemap success.
Bsram R28[6] init value convert to fusemap success.
Bsram R46[4] 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 R46[3] 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 R28[2] 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_empu_m1.fs...
Build bsram init value replace completed.

```

3.4.2 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。

3.5 适用器件

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

3.6 适用软件

适用于 IP Core Generator 中使用 Synplify Pro 产生的 Gowin_EMPU_M1 硬件设计。

3.7 参考设计

- Linux:
Gowin_EMPU_M1\script\merge_bit_script\linux\merge_bit\example
- Windows:
Gowin_EMPU_M1\script\merge_bit_script\windows\merge_bit\example

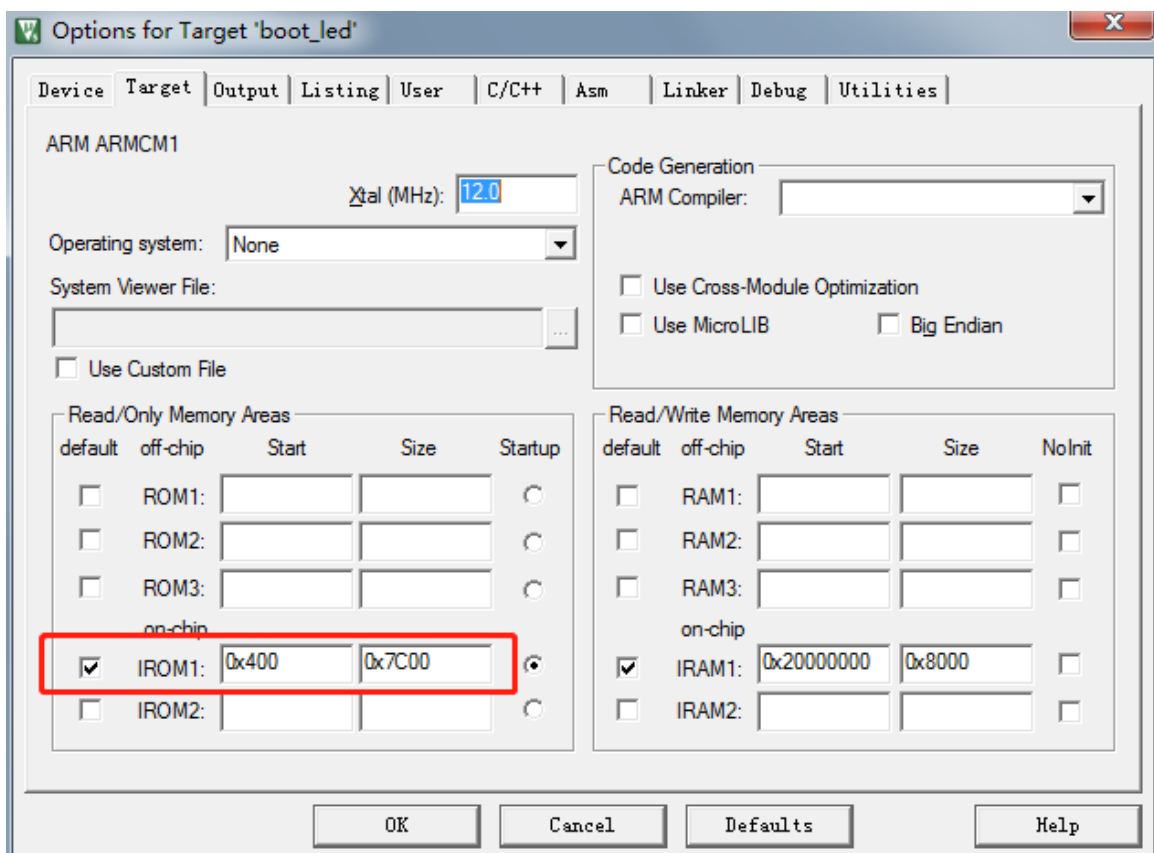
4 片外 SPI-Flash 下载方法

4.1 软件配置

Gowin_EMPU_M1 软件编程设计中：

如果使用 ARM Keil MDK(V5.24.2.0 及以上版本)软件开发环境，IROM1 起始地址设为 0x400，IROM1 容量根据 ITCM 硬件实际配置设定，以 DK-START-GW2A18 V2.0 开发板参考设计为例，设为 0x7C00，如图 4-1 所示。

图 4-1 ROM 起始地址和容量配置



如果使用 GOWIN MCU Designer 软件开发环境，修改 Flash 链接脚本

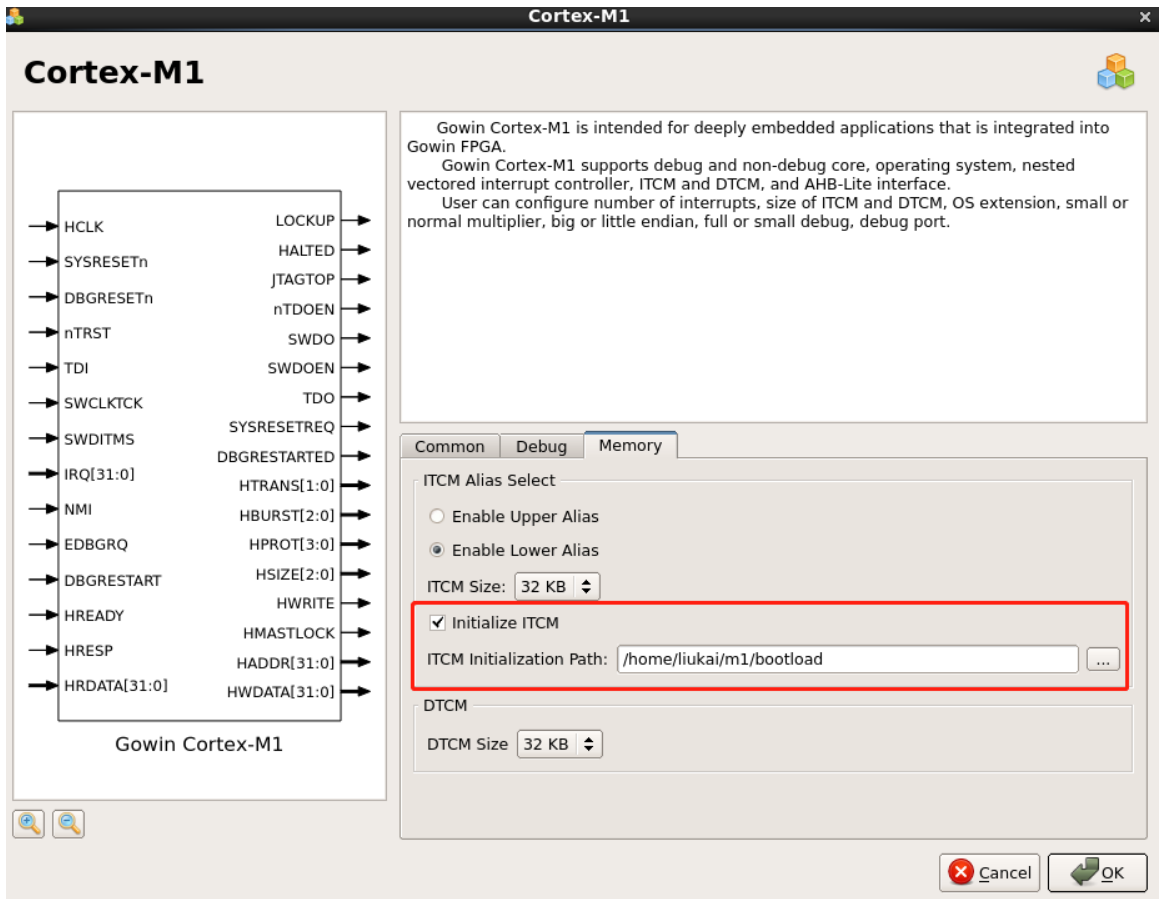
GOWIN_M1_flash.Id 中的 Flash 起始地址"FLASH ORIGIN"为 0x00000400。

4.2 硬件配置

4.2.1 ITCM Initialization 配置

IP Core Generator 配置产生 Gowin_EMPU_M1 硬件设计过程中，配置 Cortex-M1 的 ITCM Initialization Path 为 bootload 路径，读入 bootload 的 itcm0、itcm1、itcm2 和 itcm3 作为 ITCM 初始值，如图 4-2 所示。

图 4-2 配置 ITCM Initialization Path



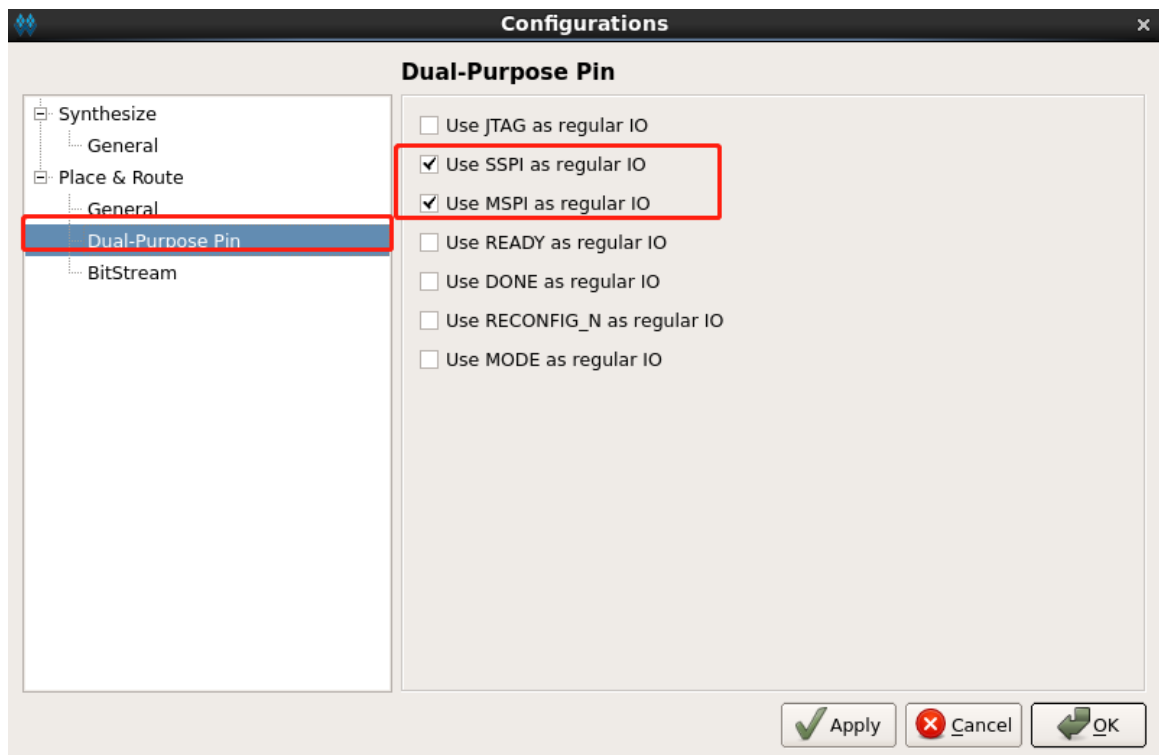
Gowin_EMPU_M1 提供片外 SPI-Flash 下载启动 bootload:

bootload\boot\itcm0、itcm1、itcm2、itcm3

4.2.2 Dual-Purpose Pin 配置

Gowin_EMPU_M1 硬件设计中，配置页面 Place & Route 标签下 Dual-Purpose Pin 配置 SSPI、MSPI 为通用端口，如图 4-3 所示。

图 4-3 配置 Dual-Purpose Pin



4.3 设计流程

1. Gowin_EMPU_M1 硬件设计配置 bootload 的 itcm0、itcm1、itcm2 和 itcm3 作为 ITCM 初始值；
2. Gowin_EMPU_M1 硬件设计输出具有片外 SPI-Flash 下载功能的码流文件；
3. Programmer 配置 Device configuration，下载码流文件；
4. Gowin_EMPU_M1 软件编程设计输出二进制 BIN 文件；
5. Programmer 配置 Device configuration，下载二进制 BIN 文件。

4.4 下载

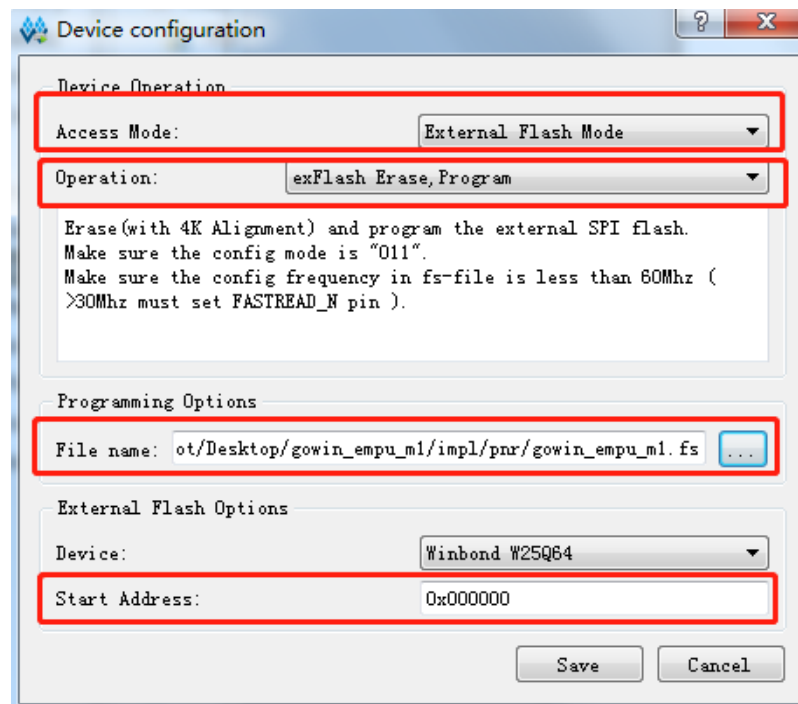
Programmer 使用方法请参考 [SUG502](#)，Gowin Programmer 用户指南。


4.4.1 下载硬件设计码流文件

Gowin_EMPU_M1 硬件设计输出以 bootload 作为 ITCM 初始值、具有片外 SPI-Flash 下载功能的码流文件，使用 Programmer 下载。

选择 Gowin 云源软件菜单栏“Tools > Programmer”或工具栏 Programmer“”，打开 Programmer。选择“Programmer > Edit > Configure Device”或工具栏“Configure Device ”，“Access Mode”下拉列表中选择“External Flash Mode”，“Operation”下拉列表中选择“exFlash Erase, Program”，“File name”中导入需要下载的 Gowin_EMPU_M1 硬件设计码流文件，“External Flash Options”中“Device”根据板载 Flash 芯片类型选择，“Start Address”设置为 0x000000，单击“Save”，如图 4-4 所示。

图 4-4 配置 Programmer 码流下载模式



完成 Device configuration 后，单击 Programmer 工具栏 Program/Configure “”，完成 Gowin_EMPU_M1 硬件设计码流文件。

4.4.2 下载软件设计 BIN 文件

完成 Gowin_EMPU_M1 硬件设计码流文件下载后，使用 Programmer 下载 Gowin_EMPU_M1 软件编程设计二进制 BIN 文件。


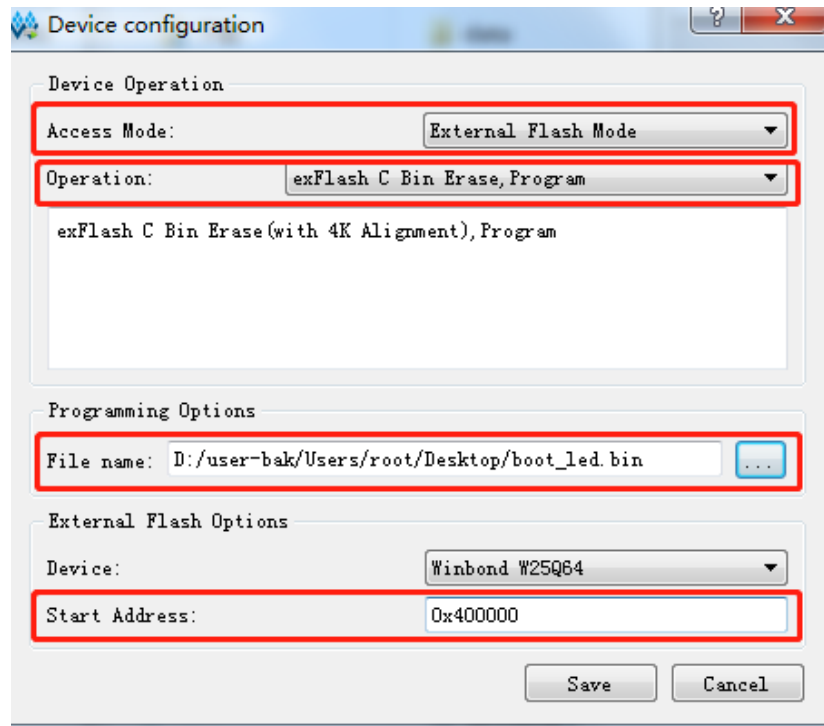

由 Gowin 云源软件中或软件安装路径中打开 Programmer，单击“Programmer > Edit > Configure Device”或工具栏 Configure Device “”，打开“Device configuration”对话框，“Access Mode”下拉列表选择“External Flash Mode”，“Operation”下拉列表选择“exFlash C Bin Erase, Program”，“Firmware/Binary File”中导入需要下载的 Gowin_EMPU_M1 二进制 BIN 文件，“External Flash Options”中“Device”根据板载 Flash 芯片类型选择，“Start Address”设置为“0x400000”，单击“Save”，如图 4-5 所示。

图 4-5 配置 Programmer C-Bin 下载模式



完成 Device configuration 后，单击 Programmer 工具栏 Program/Configure “”，完成 Gowin_EMPU_M1 软件编程设计二进制 BIN 文件下载。

4.5 适用器件

- GW2A-18
- GW2AR-18
- GW2A-55

4.6 参考设计

Gowin_EMPU_M1\bootload\example

