



# Gowin\_EMPU\_M1 ダウンロード リファレンスマニュアル

IPUG532-1.8J, 2021-07-16

## 著作権について (2021)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

**GOWIN**、Gowin、及びGOWINSEMIは、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

## 免責事項

当社は、**GOWINSEMI Terms and Conditions of Sale (GOWINSEMI取引条件)** に規定されている内容を除き、(明示的か又は黙示的に拘わらず) いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。本文書における全ての情報は、予備的情報として取り扱われなければなりません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

## バージョン履歴

日付	バージョン	説明
2019/02/19	1.0J	初版。
2019/07/18	1.1J	<ul style="list-style-type: none"> <li>● MCU はハードウェア設計およびソフトウェア設計のための自動マージツールをサポート。</li> <li>● MCU はオフチップ SPI-Flash によるダウンロードと起動をサポート。</li> </ul>
2019/08/18	1.2J	<ul style="list-style-type: none"> <li>● MCU ハードウェア設計とソフトウェア設計では、DDR3 Memory の拡張をサポート。</li> <li>● 既知の ITCM、DTCM Size および IDE の問題を修正。</li> </ul>
2019/09/27	1.3J	ソフトウェア構成の説明を更新。
2020/01/16	1.4J	<ul style="list-style-type: none"> <li>● MCU ハードウェア設計とソフトウェア設計では、周辺機器 PSRAM の拡張をサポート。</li> <li>● MCU コンパイラ GMD V1.0 を更新。</li> <li>● RTOS リファレンスデザインを更新。</li> </ul>
2020/03/03	1.5J	<ul style="list-style-type: none"> <li>● 既知の bootload size の問題を修正。</li> <li>● デバイス GW2A-18C/GW2AR-18C/GW2A-55C へのサポートを追加。</li> </ul>
2020/06/12	1.6J	<ul style="list-style-type: none"> <li>● MCU が外部命令メモリをサポート。</li> <li>● MCU が外部データメモリをサポート。</li> <li>● AHB バスインターフェースを 6 つ拡張。</li> <li>● APB バスインターフェースを 16 個拡張。</li> <li>● GPIO が複数のインターフェースタイプをサポート。</li> <li>● I2C が複数のインターフェースタイプをサポート。</li> <li>● merge_bit ツールが合成ツール GowinSynthesis の命名方法の分析をサポート。</li> </ul>
2021/01/25	1.7J	<ul style="list-style-type: none"> <li>● C バージョンの GW1N-9C、GW2A-18C、GW2A-55C のリファレンスデザインを更新。</li> <li>● リファレンスデザインで使用される Gowin ソフトウェアのバージョンを更新。</li> </ul>
2021/07/16	1.8J	<ul style="list-style-type: none"> <li>● merge_bit ツールを更新。</li> <li>● Synplify Pro の関連内容を削除。</li> <li>● FPGA および MCU ソフトウェアのバージョン情報を更新。</li> </ul>

# 目次

目次.....	i
図一覧.....	iii
表一覧.....	iv
<b>1 ダウンロード方法.....</b>	<b>1</b>
<b>2 ソフトウェア設計によって出力されたイメージファイルをハードウェア ITCM 初期値として使用.....</b>	<b>3</b>
2.1 ツール.....	3
2.2 パラメータ.....	3
2.3 ソフトウェアの構成.....	3
2.4 ハードウェアの構成.....	4
2.5 設計フロー.....	5
2.6 サポートされるデバイス.....	5
2.7 リファレンスデザイン.....	6
<b>3 ソフトウェア設計とハードウェア設計をマージ.....</b>	<b>7</b>
3.1 ツール.....	7
3.2 パラメータ.....	7
3.3 ハードウェアの構成.....	8
3.4 設計フロー.....	8
3.4.1 マージ.....	8
3.4.2 ダウンロード.....	9
3.5 サポートされるデバイス.....	9
3.6 サポートされる設計.....	9
3.7 リファレンスデザイン.....	9
<b>4 オフチップ SPI-Flash によるダウンロード方法.....</b>	<b>10</b>
4.1 ソフトウェアの構成.....	10
4.2 ハードウェアの構成.....	11
4.2.1 ITCM Initialization の構成.....	11
4.2.2 Dual-Purpose Pin の構成.....	12

---

4.3 設計フロー.....	12
4.4 ダウンロード.....	13
4.4.1 ハードウェア設計のビットストリームファイル.....	13
4.4.2 ソフトウェア設計の <b>BIN</b> ファイルのダウンロード .....	14
4.5 サポートされるデバイス.....	15
4.6 リファレンスデザイン.....	15

## 図一覧

図 2-1 外部ツールの構成.....	4
図 2-2 ITCM Initialization の構成.....	5
表 3-1 merge_bit のコマンドとパラメータ.....	7
図 3-1 Generate Post-Place File オプションの構成.....	8
図 3-2 ソフトウェア設計とハードウェア設計のマージ.....	9
図 4-1 ROM の開始アドレスと容量の構成.....	10
図 4-2 ITCM Initialization Path オプション.....	11
表 4-1 ITCM Size と bootload の対応関係.....	11
図 4-3 Dual-Purpose Pin オプションの構成.....	12
図 4-4 Device configuration(ハードウェア設計).....	14
図 4-5 Device configuration (ソフトウェア設計).....	15

## 表一覧

表 3-1 merge_bit のコマンドとパラメータ .....	7
表 4-1 ITCM Size と bootload の対応関係 .....	11

# 1 ダウンロード方法

Gowin\_EMPU\_M1 は、3 つのハードウェア設計およびソフトウェア設計のダウンロード方法をサポートしています。

1. ソフトウェア設計によって生成されたイメージファイルを、ハードウェア設計の ITCM 初期値として使用します。
  - a) Gowin\_EMPU\_M1 ソフトウェア設計で BIN ファイルを生成します。
  - b) 「make\_hex」ツールを使用して BIN ファイルを 4 つの 16 進数のイメージファイル (itcm0、itcm1、itcm2、および itcm3) に変換します。
  - c) itcm0、itcm1、itcm2、および itcm3 を ITCM 初期値として読み込みます。
  - d) 合成、配置配線し、ソフトウェア設計およびハードウェア設計のビットストリームファイルを生成します。
  - e) ダウンロードツールである **Programmer** を使用してビットストリームファイルをダウンロードします。
2. ソフトウェア設計によって生成された BIN ファイルとハードウェア設計によって生成されたビットストリームファイルをマージします。
  - a) Gowin\_EMPU\_M1 ハードウェア設計でビットストリームファイルを生成します。
  - b) Gowin\_EMPU\_M1 ソフトウェア設計で BIN ファイルを生成します。
  - c) merge\_bit ツールを使用してソフトウェア設計の BIN ファイルとハードウェア設計のビットストリームファイルをマージします。
  - d) ソフトウェア設計とハードウェア設計をマージした後の新しいハードウェア設計のビットストリームファイルを生成します。
  - e) ダウンロードツールである **Programmer** を使用して新しいビットストリームファイルをダウンロードします。
3. オフチップ SPI-Flash によってソフトウェア設計が生成した BIN ファイルをダウンロードします。
  - a) ハードウェア設計において構成された ITCM Size に応じて



bootload を ITCM 初期値として選択します。

- b) Gowin\_EMPU\_M1 ハードウェア設計でオフチップ **SPI-Flash** によるダウンロード機能を備えたビットストリームファイルを生成します。
- c) ダウンロードツールである **Programmer** を使用してハードウェア設計のビットストリームファイルをダウンロードします；
- d) Gowin\_EMPU\_M1 ソフトウェア設計で **BIN** ファイルを生成します。
- e) ダウンロードツールである **Programmer** を使用してソフトウェア設計の **BIN** ファイルをダウンロードします。

# 2 ソフトウェア設計によって出力されたイメージファイルをハードウェア ITCM 初期値として使用

## 2.1 ツール

Gowin\_EMPU\_M1¥tool¥make\_hex¥bin¥make\_hex.exe

ダウンロード URL : [http://cdn.gowinsemi.com.cn/Gowin\\_EMPU\\_M1.zip](http://cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip)

## 2.2 パラメータ

make\_hex.exe bin-file

## 2.3 ソフトウェアの構成

Gowin\_EMPU\_M1 ソフトウェア設計で BIN ファイルを生成します。

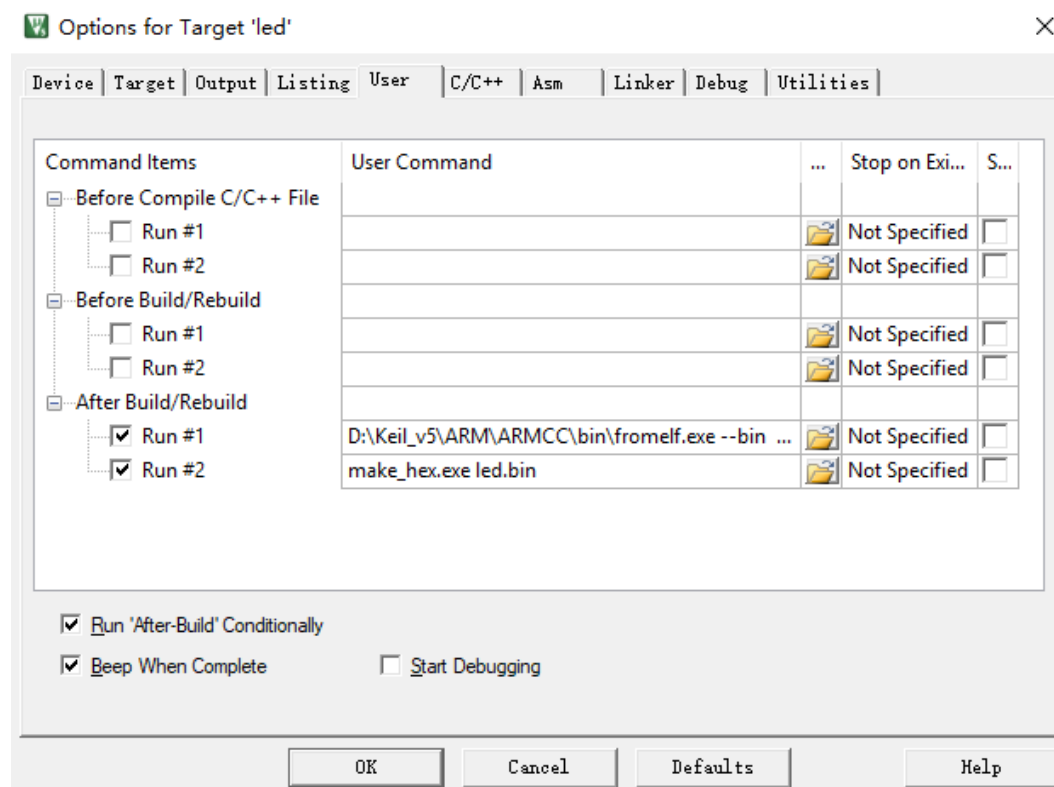
「make\_hex」ツールを使用して BIN ファイルを 4 つの 16 進数のイメージファイル (itcm0、itcm1、itcm2、および itcm3) に変換します。

ARM Keil MDK (V5.26 以降) では、make\_hex.exe を外部ツールとして構成します (図 2-1)。

- Run #1 : fromelf.exe --bin -o bin-file axf-file
- Run #2 : make\_hex.exe bin-file

ソフトウェアがコンパイルされると、make\_hex.exe ツールが自動的に呼び出され、ソフトウェア設計の BIN ファイルと 4 つの 16 進数のイメージファイルが生成されます。

図 2-1 外部ツールの構成



## 2.4 ハードウェアの構成

Gowin ソフトウェアの IP Core Generator で次のように操作します：

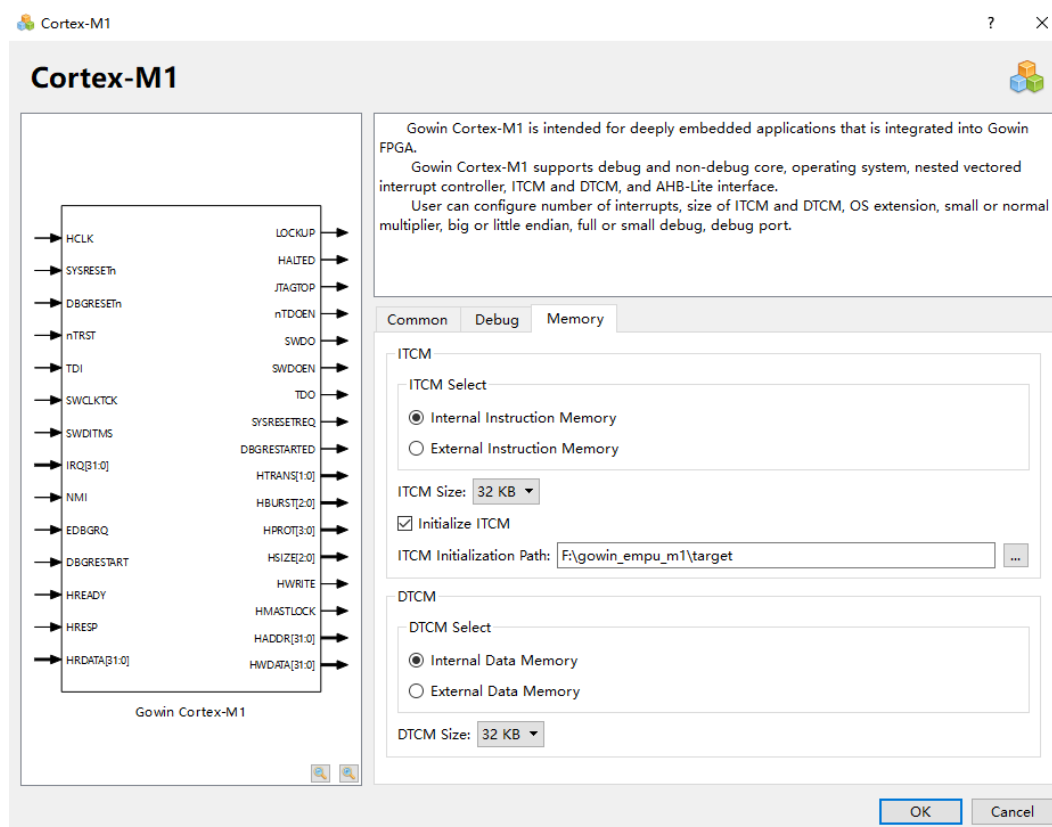
“Cortex-M1 > Memory > ITCM > ITCM Select > Internal Instruction Memory” をチェックします。

“Cortex-M1 > Memory > ITCM > Initialize ITCM” をチェックします。

“Cortex-M1 > Memory > ITCM > ITCM Initialization Path” で、itcm0、itcm1、itcm2、itcm3 の 4 つの 16 進数ファイルが所在するパスを ITCM の初期値のパスとしてインポートします（図 2-2）。

ITCM の初期値として itcm0、itcm1、itcm2、itcm3 をインポートし、IP Core Generator で他の Cortex-M1 および AHB/APB 外部デバイスの構成を完了すると、ソフトウェア設計を含む Gowin\_EMPU\_M1 ハードウェア設計が生成されます。

図 2-2 ITCM Initialization の構成



## 2.5 設計フロー

1. ARM Keil MDK (V5.26 以降) または GOWIN MCU Designer (V1.1) を使用してソフトウェア設計を実行し、コンパイルして4つの16進数のイメージファイル (itcm0、itcm1、itcm2、および itcm3) を生成します。
2. Gowin ソフトウェアの IP Core Generator で Gowin\_EMPU\_M1 ハードウェア設計を構成・生成し、itcm0、itcm1、itcm2 および itcm3 をハードウェア設計における ITCM の初期値として使用します。
3. Gowin\_EMPU\_M1 の Top Module をインスタンス化し、それをユーザーデザインに接続します。
4. 物理制約とタイミング制約を与えます。
5. GowinSynthesis を使用して合成します。
6. 合成が完了したら、配置配線ツールの Place & Route を実行し、ソフトウェア設計を含むビットストリームファイルを生成します。
7. ダウンロードツールである Programmer を使用してビットストリームファイルをダウンロードします。

## 2.6 サポートされるデバイス

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

AN-18X

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

## 2.7 リファレンスデザイン

ダウンロード URL :

Gowin\_EMPU\_M1¥tool¥make\_hex¥example

# 3 ソフトウェア設計とハードウェア設計をマージ

## 3.1 ツール

Gowin\_EMPU\_M1¥tool¥merge\_bit¥bin¥merge\_bit.bat

ダウンロード URL : [http://cdn.gowinsemi.com.cn/Gowin\\_EMPU\\_M1.zip](http://cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip)

## 3.2 パラメータ

ソフトウェアツールのコマンドとパラメータを説明します。

```
call make_loc.exe -i posp-file -s itcm_size [-d] -t synthesis_tool
```

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

コマンドとパラメータの説明を表 3-1 に示します。

表 3-1 merge\_bit のコマンドとパラメータ

パラメータ	説明
make_loc.exe	posp-file を入力して、ITCM の配置情報を含む itcm.loc ファイルを生成します
-i	Gowin ソフトウェアで “Place & Route > General > Generate Post-Place File” を構成して生成された Post-Place File。
-s	Gowin_EMPU_M1 ハードウェア設計で構成された ITCM Size に従って設定します。
-d	オプション Enable Debug を構成する場合、-d を有効にします。Disable Debug を構成する場合、-d を無効にします。
-t	合成ツールの選択 : gowin_syn。
merge_bit.exe	Gowin_EMPU_M1 ハードウェア設計およびソフトウェア設計をマージします。
bin-file	Gowin_EMPU_M1 ソフトウェア設計で生成された BIN ファイル。
itcm.loc	make_loc.exe によって生成された ITCM の配置情報を含む itcm.loc ファイル
fs-file	Gowin_EMPU_M1 ハードウェア設計で生成されたビットストリームファイル。

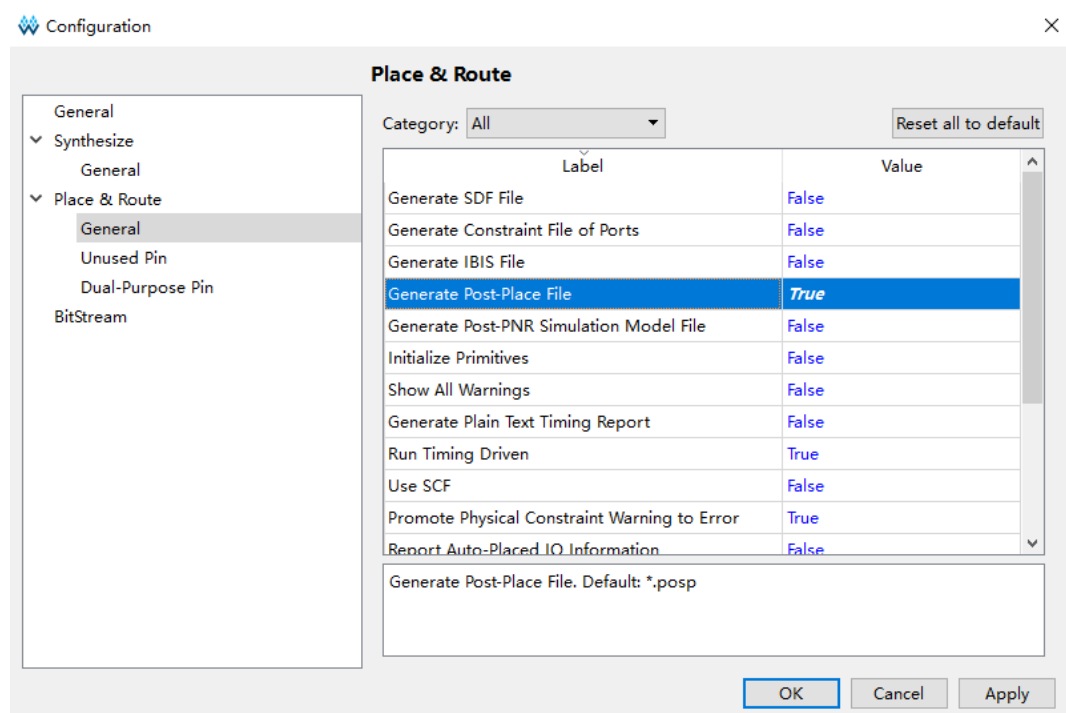
ソフトウェア設計の BIN ファイルとハードウェア設計のビットストリームファイルをマージして生成された新しいハードウェア設計のビットストリームファイル。

merge\_bit.bat を使用するとき、実際の状況に応じてパラメータ -i posp-file、-s itcm\_size、-d、bin-file、fs-file を変更します。

## 3.3 ハードウェアの構成

Gowin ソフトウェアで“Place & Route > General > Generate Post-Place File” オプションを“True” に設定し、make\_loc.exe -i パラメータの posp 入力ファイルとしての Post-Place File を生成します (図 3-1)。

図 3-1 Generate Post-Place File オプションの構成



## 3.4 設計フロー

### 3.4.1 マージ

1. Gowin\_EMPU\_M1 ハードウェア設計でビットストリームファイルと Generate Post-Place File を生成します。
2. Gowin\_EMPU\_M1 ソフトウェア設計で BIN ファイルを生成します。
3. merge\_bit.bat を実行して、ハードウェア設計のストリームファイルとソフトウェア設計の BIN ファイルをマージし、新しいハードウェア設計のストリームファイルを出力します (図 3-2)。

図 3-2 ソフトウェア設計とハードウェア設計のマージ

```
----- GOWIN Merge Tool -----  
Read bit stream file gowin_empu_m1.fs ...  
Build bsram init value fusemap...  
Reading original bsram init value map...  
Location file itcm.loc reading...  
Bsram R46[0] init value convert to fusemap success.  
Bsram R46[1] init value convert to fusemap success.  
Bsram R46[2] init value convert to fusemap success.  
Bsram R46[3] init value convert to fusemap success.  
Bsram R28[5] init value convert to fusemap success.  
Bsram R46[4] init value convert to fusemap success.  
Bsram R28[6] init value convert to fusemap success.  
Bsram R28[7] init value convert to fusemap success.  
Bsram R28[8] init value convert to fusemap success.  
Bsram R28[9] init value convert to fusemap success.  
Bsram R10[11] init value convert to fusemap success.  
Bsram R28[10] init value convert to fusemap success.  
Bsram R10[12] init value convert to fusemap success.  
Bsram R10[13] init value convert to fusemap success.  
Bsram R10[14] init value convert to fusemap success.  
Bsram R10[15] 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 ダウンロード

マージしたあと、Programmer を使用して新しいハードウェア設計のビットストリームファイルをダウンロードします。

Programmer の使用方法については、『Gowin Programmer ユーザーガイド (SUG502)』を参照してください。

## 3.5 サポートされるデバイス

- GW2AN-9X
- GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C/GW2AN-18X
- GW2A-55/GW2A-55C/GW2AN-55C

## 3.6 サポートされるソフトウェア

Gowin\_V1.9.8 Beta 以降。

## 3.7 リファレンスデザイン

Gowin\_EMPU\_M1¥tool¥merge\_bit¥example



# 4 オフチップ SPI-Flash によるダウンロード方法

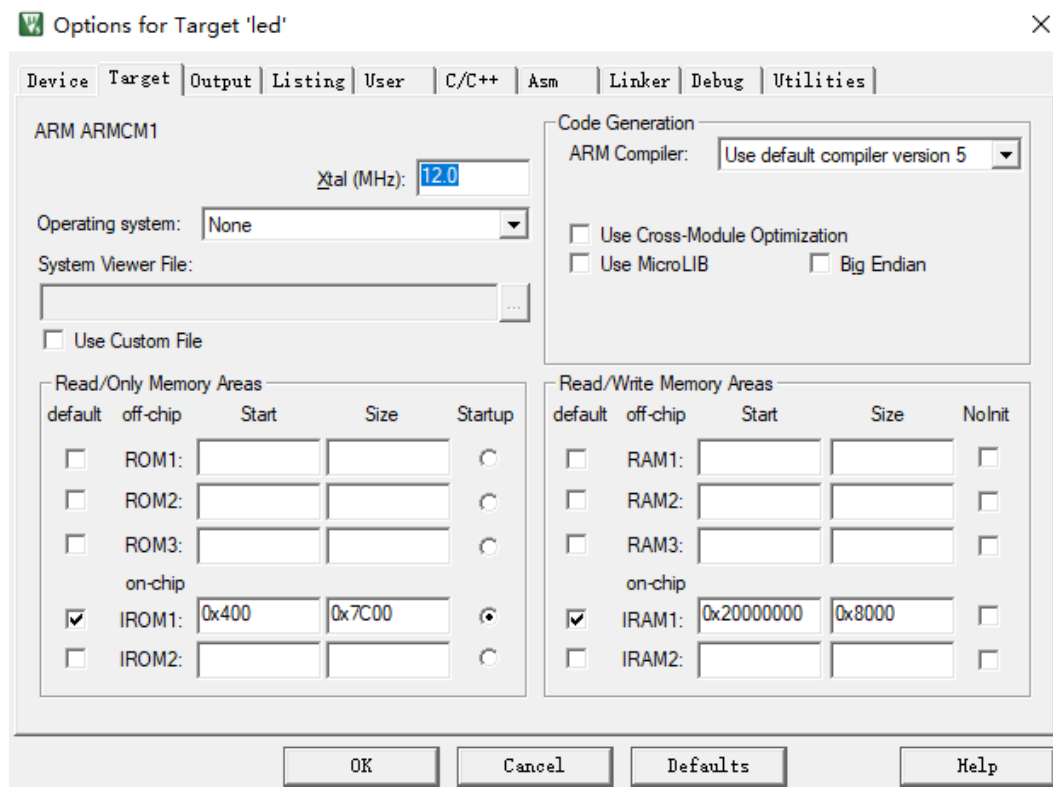
## 4.1 ソフトウェアの構成

Gowin\_EMPU\_M1 ソフトウェア設計：

ARM Keil MDK (V5.26 以降) を使用する場合、IROM1 の開始アドレスを 0x400 に設定し、IROM1 Size は実際の ITCM Size に従って設定します。

DK-START-GW2A18 V2.0 開発ボードのリファレンスデザインを例に説明します。IROM1 は 0x7C00 に設定されています (図 4-1)。

図 4-1 ROM の開始アドレスと容量の構成



GOWIN MCU Designer (V1.1 以降) を使用する場合、Flash のリンカー GOWIN\_M1\_flash.Id の Flash 開始アドレス” FLASH ORIGIN” を 0x00000400 に変更します。

## 4.2 ハードウェアの構成

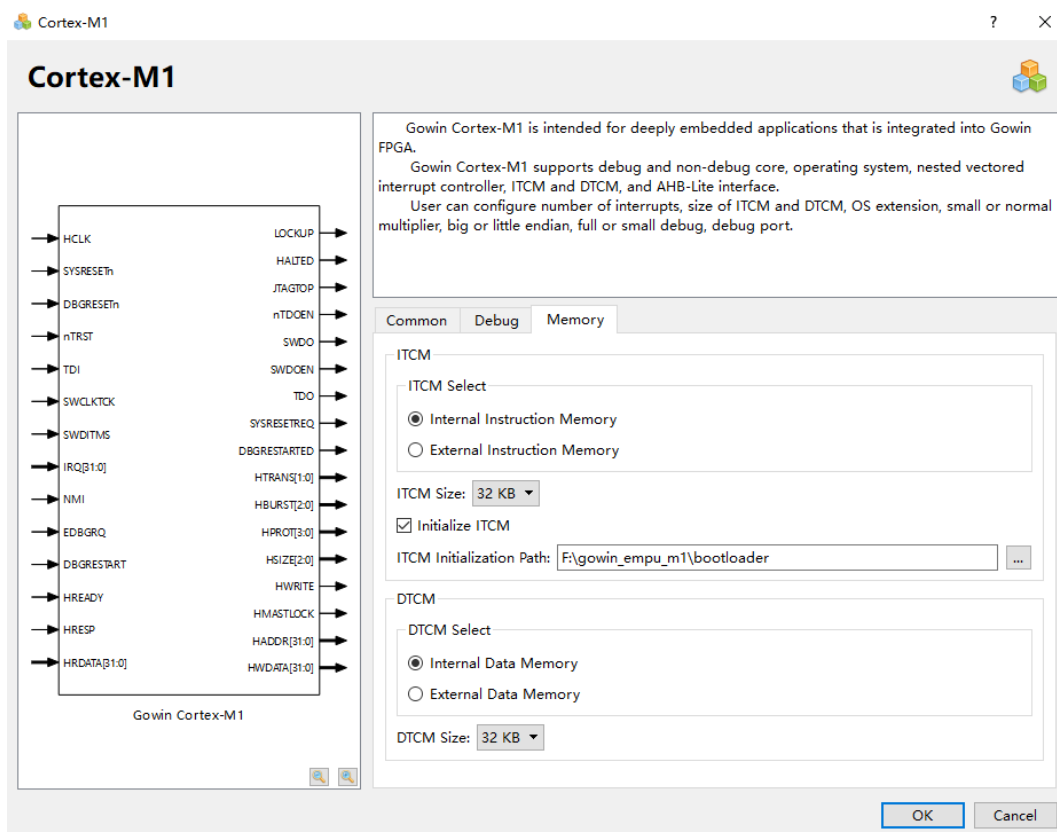
### 4.2.1 ITCM Initialization の構成

Gowin c ソフトウェアの IP Core Generator ツールを使用して Gowin\_EMPU\_M1 ハードウェア設計を構成するプロセスで、次のように操作します。

- Internal Instruction Memory を Gowin\_EMPU\_M1 の命令メモリとして選択します。
- ITCM Size を構成します。
- Initialized ITCM を選択します。
- ITCM Size に応じて、ITCM 初期値として bootload を選択し、ITCM Initialization Path フィールドにブートロードパスをインポートします。

ITCM Initialization の構成を図 4-2 に示します。

図 4-2 ITCM Initialization Path オプション



ITCM Size と bootload の対応関係を表 4-1 に示します。

表 4-1 ITCM Size と bootload の対応関係

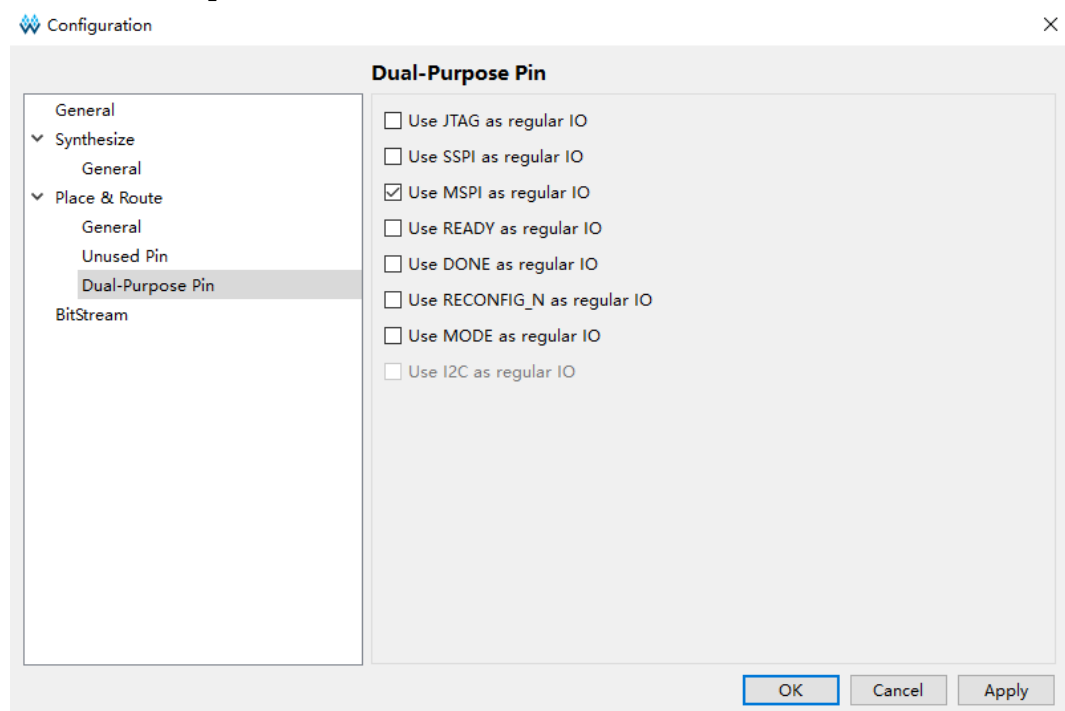
ITCM Size (KByte)	bootload
-------------------	----------

ITCM Size (KByte)	bootload
2	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_2KB
4	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_4KB
8	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_8KB
16	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_16KB
32	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_32KB
64	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_64KB
128	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_128KB
256	Gowin_EMPU_M1¥bootload¥boot¥ITCM_Size_256KB

## 4.2.2 Dual-Purpose Pin の構成

Gowin ソフトウェアの “Place & Route > Dual-Purpose Pin” で MSPI を通常の IO として構成します (図 4-3)。

図 4-3 Dual-Purpose Pin オプションの構成



## 4.3 設計フロー

1. Gowin\_EMPU\_M1 のハードウェア設計では :
  - Internal Instruction Memory を選択します。
  - ITCM Size を構成します。
  - Initialized ITCM を選択します。
  - ITCM Size に応じて bootload を ITCM 初期値として選択します。
2. Gowin\_EMPU\_M1 ハードウェア設計を生成します。
3. 合成、配置配線を経てオフチップ SPI-Flash によるダウンロード機能を備えたビットストリームファイルを生成します。
4. ダウンロードツールである Programmer を使用してハードウェア設計のビットストリームファイルをダウンロードします。

5. Gowin\_EMPU\_M1 ソフトウェア設計で BIN ファイルを生成します。
6. Programmer を使用してソフトウェア設計の BIN ファイルをダウンロードします。


## 4.4 ダウンロード

Programmer の使用方法については、『Gowin Programmer ユーザーガイド ([SUG502](#))』を参照して下さい。

### 4.4.1 ハードウェア設計のビットストリームファイル

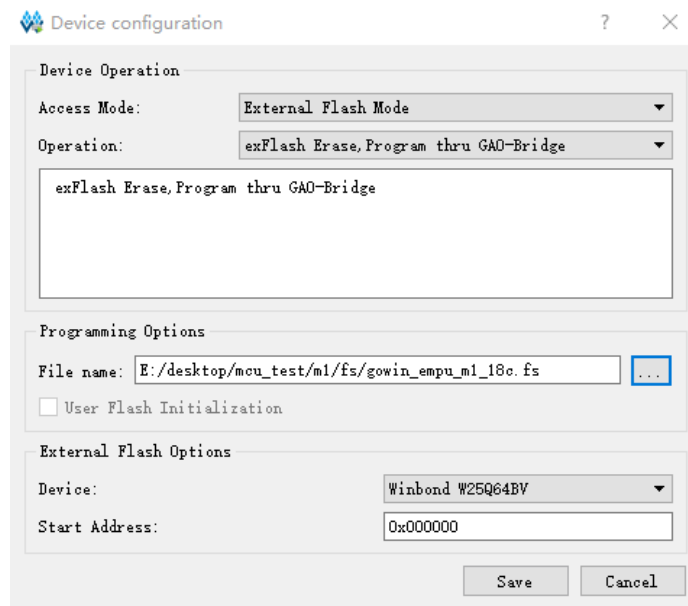
Gowin\_EMPU\_M1 ハードウェア設計では、bootload を ITCM 初期値として使用し、オフチップ SPI-Flash によるダウンロード機能を備えたビットストリームファイルを生成します。Programmer を使用してビットストリームファイルをダウンロードできます。


メニューバーの “Tools > Programmer” またはツールバー “Programmer ” をクリックして Programmer を開きます。

“Programmer > Edit > Configure Device” またはツールバーの “Configure Device ” をクリックして、“Device configuration” ダイアログボックスを開きます。

- “Access Mode” のプルダウンリストから “External Flash Mode” を選択します。
- “Operation” のプルダウンリストから “exFlash Erase, Program thru GAO-Bridge” または “exFlash Erase, Program, Verify thru GAO-Bridge” を選択します。
- Programming Options > File name を選択してダウンロードされるビットストリームファイルをインポートします。
- オンボード Flash チップのタイプに応じて “External Flash Options” の “Device” を選択します。例えば、DK-START-GW2A18 V2.0 の場合、Winbond W25Q64BV を選択します。
- “External Flash Options > Start Address” は “0x000000” に構成します。
- 最後に “Save” をクリックします (図 4-4)。

図 4-4 Device configuration(ハードウェア設計)




Device configuration が完了したら、Programmer のツールバーの Program/Configure “” をクリックしてビットストリームファイルをダウンロードします。

#### 4.4.2 ソフトウェア設計の BIN ファイルのダウンロード

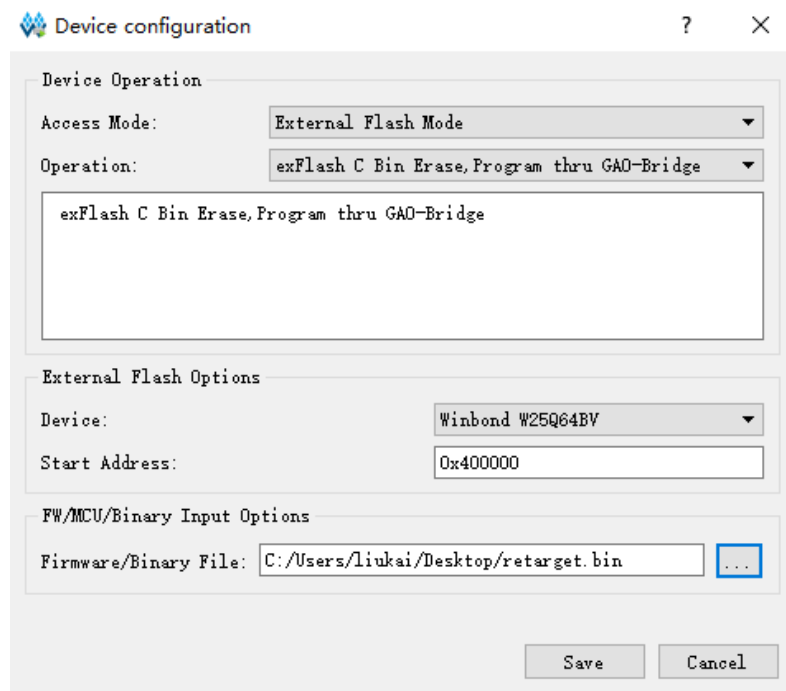
Gowin\_EMPU\_M1 ハードウェア設計からのビットストリームファイルのダウンロードが完了したら、Programmer を使用してソフトウェア設計の BIN ファイルをダウンロードします。


Gowin ソフトウェアから Programmer を開きます。

“Programmer > Edit > Configure Device” またはツールバーの “Configure Device ” をクリックして、“Device configuration” ダイアログボックスを開きます。

- “Access Mode” のプルダウンリストから “External Flash Mode” を選択します。
- “Operation” のプルダウンリストから “exFlash C Bin Erase, Program thru GAO-Bridge” または “exFlash C Bin Erase, Program, Verify thru GAO-Bridge” を選択します。
- ” FW/MCU/Binary Input Options > Firmware/Binary File” を選択して BIN ファイルをダウンロードします。
- オンボード Flash チップのタイプに応じて “External Flash Options” の “Device” を選択します。例えば、DK-START-GW2A18 V2.0 開発ボードの場合、Winbond W25Q64BV を選択します。
- “External Flash Options > Start Address” は “0x400000” に構成します。
- 最後に “Save” をクリックします (図 4-5)。

図 4-5 Device configuration (ソフトウェア設計)



Device configuration が完了したら、Programmer のツールバーの “Program/Configure” (  ) をクリックして BIN ファイルをダウンロードします。

## 4.5 サポートされるデバイス

- GW2AN-9X
- GW2A-18/GW2A-18C/GW2AR-18/GW2AR-18C/GW2ANR-18C/GW2AN-18X
- GW2A-55/GW2A-55C/GW2AN-55C

## 4.6 リファレンスデザイン

[リファレンスデザイン](#) :

Gowin\_EMPU\_M1¥bootload¥example

