

Gowin PicoRV32 ソフトウェアダウンロード リファレンスマニュアル

IPUG913-1.8J, 2025-09-26

著作権について(2025)

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高云、、Gowin、及び LittleBee 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かに拘わらず)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。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
2020/01/16	1.0J	初版。
2020/03/06	1.1J	● MCU が Wishbone インターフェースの周辺機器 GPIO をサポート。 ● MCU が拡張 AHB バスインターフェースをサポート。 ● MCU がオフチップ SPI-Flash memory によるダウンロードをサポート。 ● MCU がオフチップ SPI-Flash memory の読み出し、書き込み、および消去をサポート。 ● MCU が Hardware Stack Protection および Trap Stack Overflow 機能をサポート。
2020/06/01	1.2J	● MCU のオンラインデバッグをサポート。 ● MCU コアの割り込み処理機能を強化。 ● MCU コアの命令を最適化。 ● mergebin ツールが合成ツール GowinSynthesis の命名方法の分析をサポートするように更新。
2021/07/16	1.3J	● 合成ツールの Synplify Pro を削除。 ● FPGA ソフトウェアのバージョン情報を更新。 ● サポートされるデバイスの情報を更新。
2022/02/11	1.4J	makehex および mergebin 方法を使用したリファレンス・デザインを追加。
2023/08/18	1.5J	Arora V FPGA 製品のサポートを追加。
2024/03/29	1.6J	mergebin ツールに GW2AN-18X/9X FPGA のサポートを追加。
2024/06/14	1.7J	merge_bit ツールに Arora V FPGA のサポートを追加。
2025/09/26	1.8J	関連ソフトウェアのバージョン情報を更新。

目次

目次	i
図一覧	iii
表一覧	iv
1 ダウンロード方法	1
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	4
2.1 ツール	4
2.2 コマンドとパラメータ	4
2.3 ハードウェアの構成	4
2.4 ソフトウェアの構成	5
2.4.1 起動モードの構成	5
2.4.2 Flash リンカー・スクリプト・ファイルの構成	5
2.4.3 出力されるファイルの形式の構成	6
2.5 設計フロー	6
2.6 サポートされるデバイス	7
2.7 リファレンス・デザイン	7
3 ソフトウェア設計とハードウェア設計をマージ	8
3.1 ツール	8
3.2 コマンドとパラメータ	8
3.3 ハードウェアの構成	8
3.3.1 ITCM の構成	9
3.3.2 Post-Place File の構成	9
3.4 ソフトウェアの構成	10
3.4.1 起動モードの構成	10
3.4.2 Flash リンカー・スクリプト・ファイルの構成	10
3.4.3 出力されるファイルの形式の構成	11
3.5 設計フロー	11
3.5.1 マージ	11

3.5.2 ダウンロード	12
3.6 サポートされるデバイス	12
3.7 リファレンス・デザイン	12
4 オフチップ SPI-Flash Memory のダウンロード方法	13
4.1 ソフトウェアの構成	13
4.1.1 起動モードの構成	13
4.1.2 Flash リンカー・スクリプト・ファイルの構成	14
4.2 ハードウェアの構成	16
4.2.1 ITCM の構成	16
4.2.2 Dual-Purpose Pin の構成	17
4.3 設計フロー	17
4.4 ダウンロード	18
4.4.1 ハードウェア設計のビットストリーム・ファイルのダウンロード	18
4.4.2 実行ファイルのダウンロード	20
4.5 サポートされるデバイス	22

図一覧

図 2-1 ITCM の構成	5
図 2-2 起動モードの構成	5
図 2-3 Flash リンカー・スクリプト・ファイルの構成	6
図 3-1 ITCM の構成	9
図 3-2 Generate Post-Place File オプションの構成	10
図 3-3 起動モードの構成	10
図 3-4 Flash リンカー・スクリプト・ファイルの構成	11
図 3-5 ソフトウェア設計の出力とハードウェア設計の出力のマージ	12
図 4-1 起動モードの構成「BUILD_BURN」	13
図 4-2 起動モードの構成「BUILD_XIP」	14
図 4-3 Flash リンカー・スクリプト・ファイルの構成	15
図 4-4 Flash リンカー・スクリプト・ファイルの構成	16
図 4-5 ITCM の構成	17
図 4-6 Dual-Purpose Pin の構成	17
図 4-7 Arora ファミリーの Device configuration	19
図 4-8 Arora V の Device Configuration	20
図 4-9 Arora ファミリーの Device configuration	21
図 4-10 Arora V の Device configuration	22

表一覧

表 3-1 パラメータ	8
-------------------	---

1 ダウンロード方法

Gowin_PicoRV32 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出力された実行ファイルを、ハードウェア設計の命令レジスタ ITCM の初期値として使用します。
 - a) Gowin_PicoRV32 ソフトウェア設計：
 - config.h でマクロ定義します：`#define BUILD_MODE BUILD_LOAD`
 - sections.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - コンパイルして実行ファイルを生成します
 - b) make_hex ツールを使用して実行ファイル.bin を ram32.hex に変換します。
 - c) IP Core Generator を使用して Gowin_PicoRV32 IP を生成する際に命令メモリ ITCM を構成します：
 - Boot Mode として MCU boot and run in ITCM を選択します
 - ITCM Initialiaztion File では b)の ram32.hex ファイルを ITCM の初期値としてインポートします
 - 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。
 - f) ソフトウェア設計を更新した後、a)～e)を再実行します。
2. ソフトウェア設計によって生成された実行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) 必要に応じて、上記の方法 1 で Gowin_PicoRV32 ハードウェア設計のビットストリーム・ファイルおよび Post-Place

File を生成します。

- b) Gowin_PicoRV32 ソフトウェア設計を更新します :
 - config.h でマクロ定義します : #define BUILD_MODE BUILD_LOAD
 - sections.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - 必要に応じてユーザーのソフトウェア設計を更新します
 - コンパイルして実行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の実行ファイル.bin と a) の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。
 - f) ソフトウェア設計を更新した後、b)~e)を再実行します。
3. オフチップ SPI-Flash Memory を使用して、ソフトウェア設計で生成された実行ファイルをダウンロードします。
- a) IP Core Generator を使用して Gowin_PicoRV32 IP を生成する際に命令メモリ ITCM を構成します :
 - Boot Mode として MCU boot and run in external Flash または MCU boot from external Flash and run in ITCM を選択します。
 - b) Gowin_PicoRV32 ハードウェア設計でオフチップ SPI-Flash Memory によるダウンロード用のビットストリーム・ファイルを生成します。
 - c) ダウンロードツールである **Programmer** を使用してビットストリーム・ファイルをダウンロードします。
 - d) Gowin_PicoRV32 ソフトウェア設計 :
 - MCU boot and run in external Flash
config.h でマクロ定義します : #define BUILD_MODE BUILD_XIP
sections_xip.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - MCU boot from external Flash and run in ITCM
定義 config.h でマクロ定義します : #define BUILD_MODE BUILD_BURN
sections.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - コンパイルして実行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実行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

…¥tool¥make_hex¥bin¥make_hex.exe

上記ツールのダウンロード URL :

cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

2.2 コマンドとパラメータ

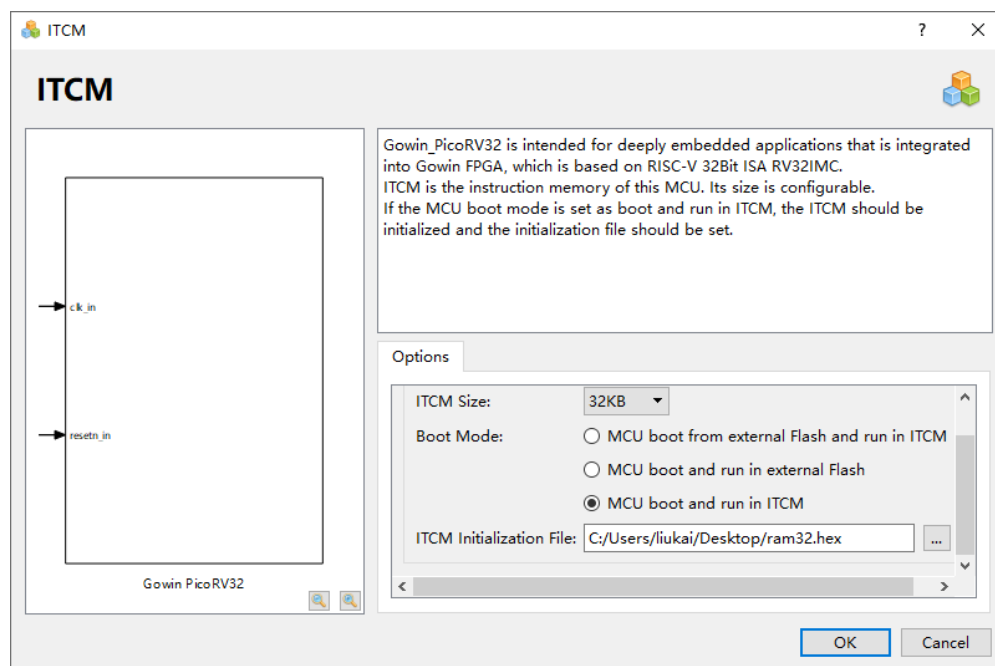
ツールのコマンドとパラメータ : make_hex.exe bin-file

2.3 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 :

- ITCM > Boot Mode > MCU boot and run in ITCM を選択します。
- ITCM > ITCM Initialization File で、make_hex ツールによって生成される ram32.hex を ITCM の初期値としてインポートします(図 2-1)。

図 2-1 ITCM の構成

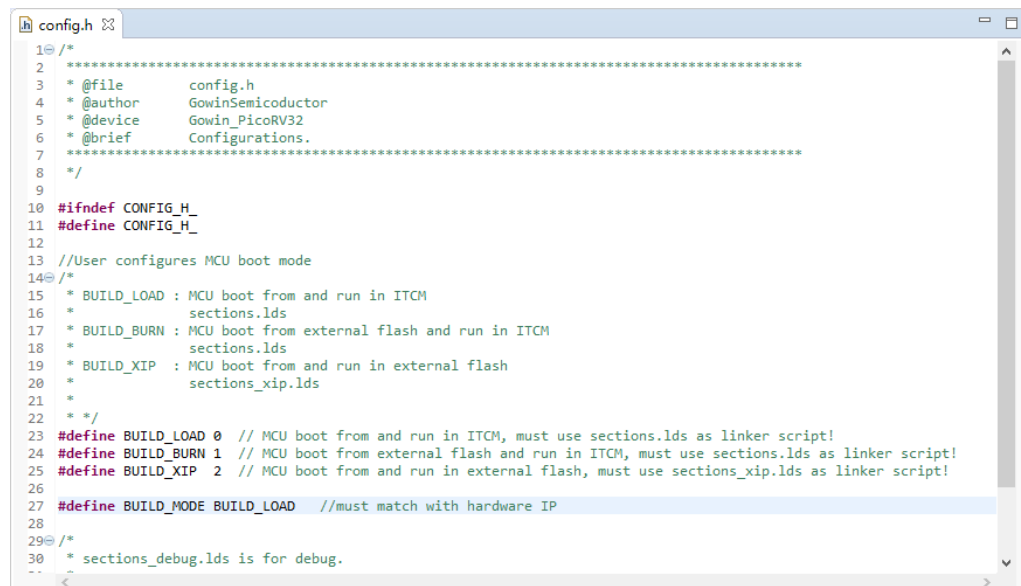


2.4 ソフトウェアの構成

2.4.1 起動モードの構成

config.h でマクロ定義します : #define BUILD_MODE BUILD_LOAD。
図 2-2 に示すとおりです。

図 2-2 起動モードの構成

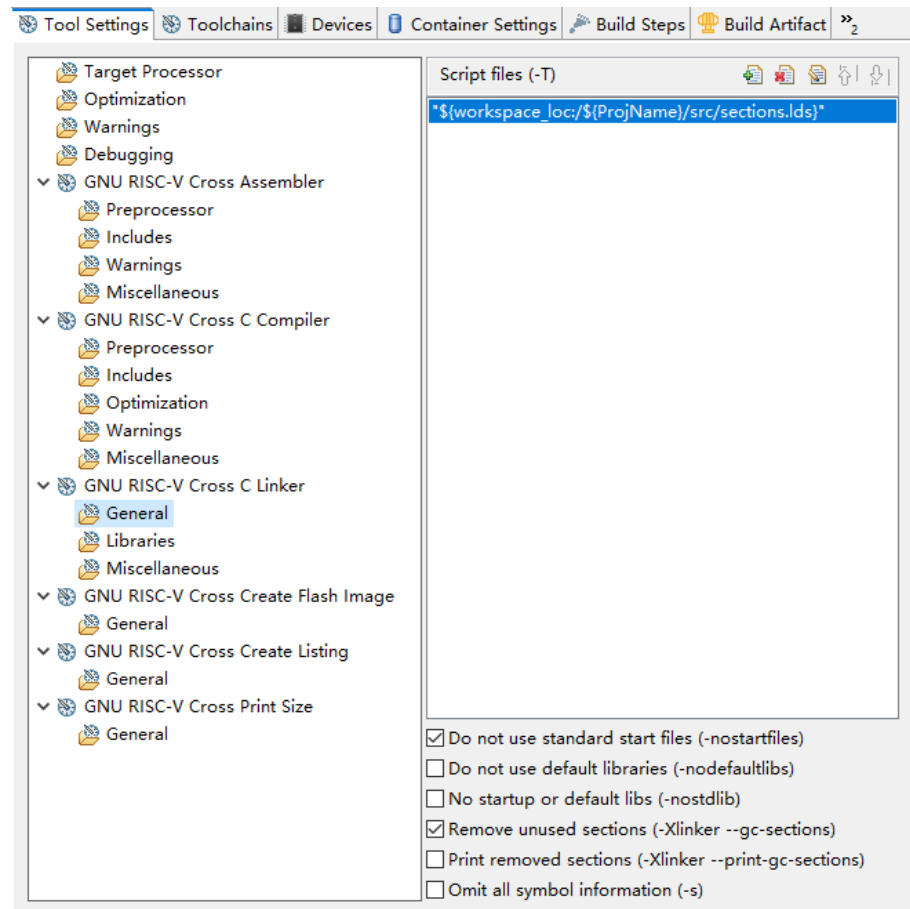


2.4.2 Flash リンカー・スクリプト・ファイルの構成

GMD ソフトウェアの Project Explorer ビューで、現在のプロジェクト
を右クリックして、Properties > C/C++ Build > Settings > Tool Settings >
GNU RISC-V Cross C Linker > General で sections.lds を Flash リンカー・

スクリプト・ファイルとして構成します(図 2-3)。

図 2-3 Flash リンカー・スクリプト・ファイルの構成



2.4.3 出力されるファイルの形式の構成

Gowin_PicoRV32 ソフトウェア設計をコンパイルし、ソフトウェア設計の実行ファイルを生成します。

`make_hex.exe bin-file` コマンドを実行し、`ram32.hex` ファイルを生成します。

2.5 設計フロー

1. ソフトウェア設計のフロー：
 - `config.h` でマクロ定義します：`#define BUILD_MODE BUILD_LOAD`。
 - `section.lds` を Flash リンカー・スクリプト・ファイルとして構成します。
 - コンパイルして実行ファイルを生成します。
 - `make_hex.exe` を実行して `ram32.hex` ファイルを Gowin_PicoRV32 IP 設計の ITCM の初期値として生成します。
2. ハードウェア設計のフロー：

- ITCM > Boot Mode > MCU boot and run in ITCM を選択します。
 - ITCM > ITCM Initialization File で ram32.hex ファイルを ITCM の初期値としてインポートします。
3. Gowin_PicoRV32 IP 設計を生成します。Gowin_PicoRV32 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. 合成ツールである GowinSynthesis で合成してネットリスト・ファイルを生成します。
 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。
 8. ソフトウェア設計を更新した後、上記のステップ 1~7 を再実行します。

2.6 サポートされるデバイス

- LittleBee ファミリーFPGA 製品
- Arora ファミリーFPGA 製品
- Arora V FPGA 製品

2.7 リファレンス・デザイン

[リファレンス・デザイン](#)のダウンロード URL :

…¥tool¥make_hex¥ref_design¥FPGA_RefDesign¥gowin_picorv32

...

¥tool¥make_hex¥ref_design¥MCU_RefDesign¥picorv32_load_demo

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

…¥tool¥merge_bit¥bin¥merge_bit.exe

上記ツールのダウンロード URL :

cdn.gowinsemi.com.cn/Gowin_PicoRV32.zip

3.2 コマンドとパラメータ

ツールのコマンドとパラメータ : merge_bit.exe bin-file fs-file itcm_size posp-file。

パラメータの説明を表 3-1 に示します。

表 3-1 パラメータ

パラメータ	説明
posp-file	Post-Place File
itcm_size	ITCM Size (KB) 例:ITCM Size が 64K バイトに設定されている場合、itcm-size は 64 です。
bin-file	実行ファイル。
fs-file	ハードウェア設計のビットストリーム・ファイル。

ソフトウェア設計の実行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。

merge_bit.bat を使用する際、実際の状況に応じてパラメータ posp-file、itcm_size、bin-file、fs-file を変更します。

3.3 ハードウェアの構成

上記第 2 章の方法 1 で Gowin_PicoRV32 ハードウェア設計のビットストリーム・ファイルおよび Post-Place File を生成することができます。

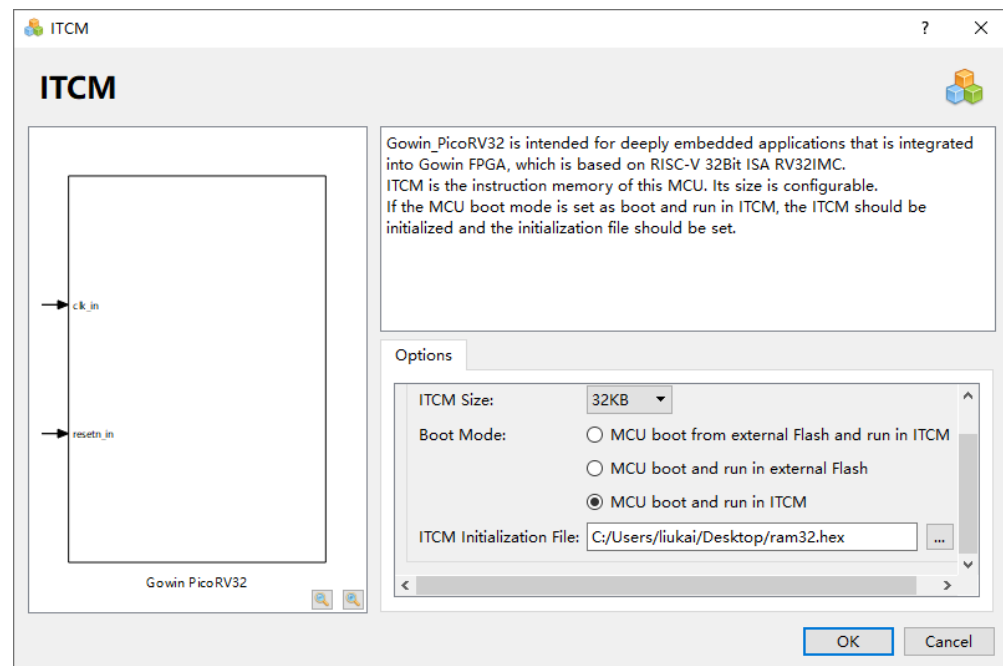
ソフトウェア設計を更新した後、merge_bit ツールを使用して上記のハードウェア設計のビットストリーム・ファイルと更新された各ソフトウェア実行ファイルをマージするだけです。

3.3.1 ITCM 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：

- ITCM > Boot Mode > MCU boot and run in ITCM を選択します。
- ITCM > ITCM Initialization File で ram32.hex ファイルを ITCM の初期値としてインポートします(図 3-1)。

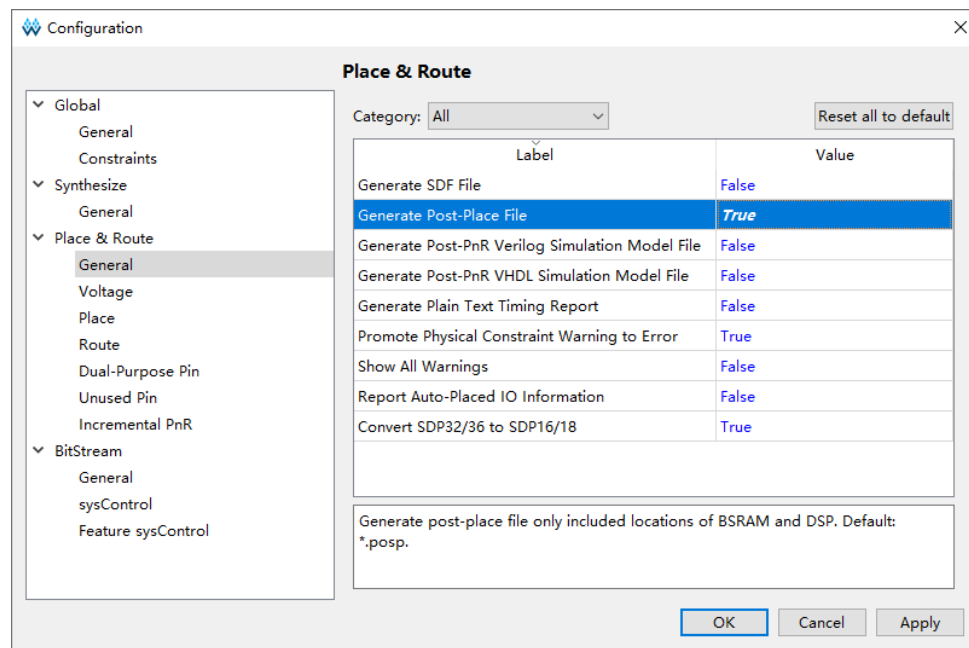
図 3-1 ITCM の構成



3.3.2 Post-Place File 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、posp-file としての Post-Place File を生成します(図 3-2)。

図 3-2 Generate Post-Place File オプションの構成

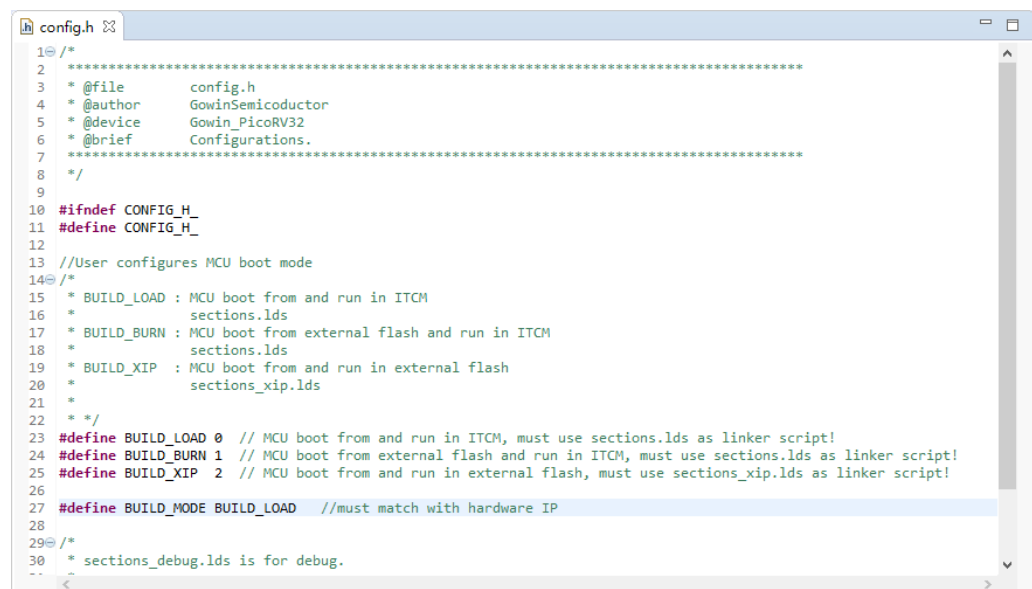


3.4 ソフトウェアの構成

3.4.1 起動モードの構成

config.h でマクロ定義します : #define BUILD_MODE BUILD_LOAD。
図 3-3 に示すとおりです。

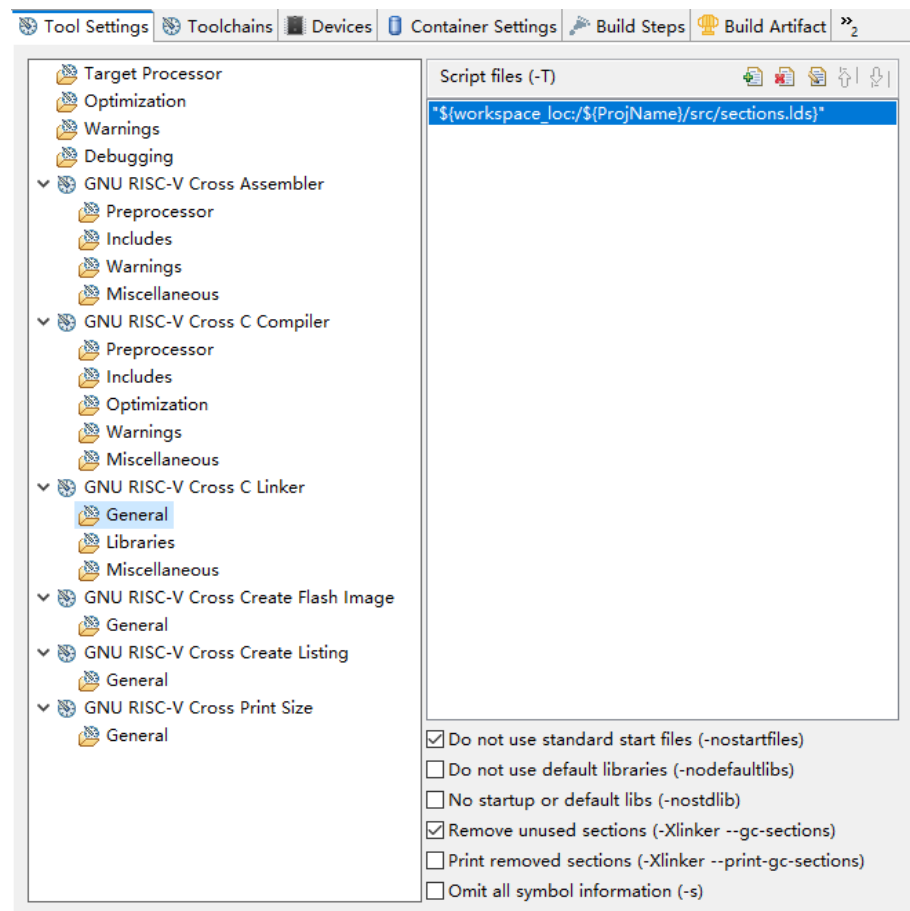
図 3-3 起動モードの構成



3.4.2 Flash リンカー・スクリプト・ファイルの構成

GMD ソフトウェアの Project Explorer ビューで、現在のプロジェクトを右クリックして、Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General で sections.lds を Flash リンカー・スクリプト・ファイルとして構成します(図 3-4)。

図 3-4 Flash リンカー・スクリプト・ファイルの構成



3.4.3 出力されるファイルの形式の構成

Gowin_PicoRV32 ソフトウェア設計をコンパイルし、ソフトウェア設計の実行ファイルを生成します。

3.5 設計フロー

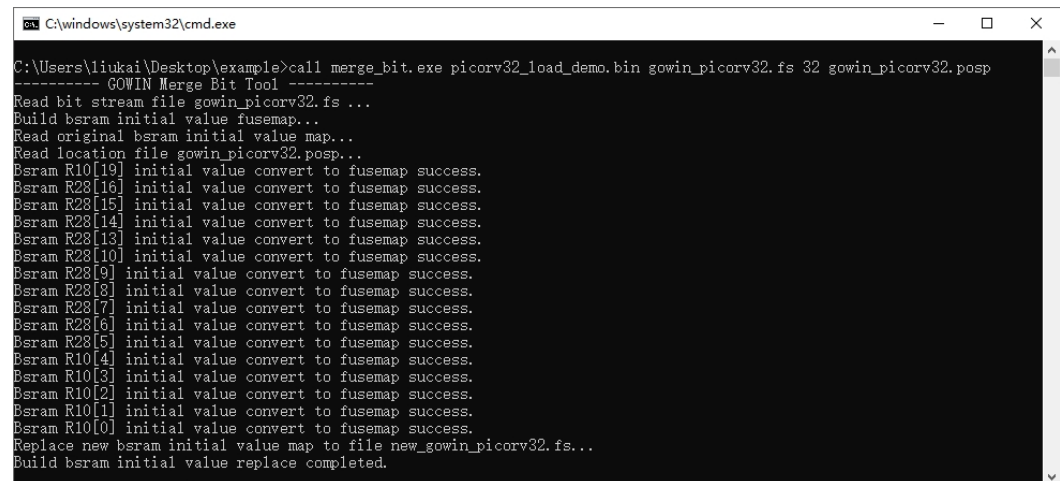
3.5.1 マージ

1. 必要に応じて、上記第 2 章の方法 1 で Gowin_PicoRV32 ハードウェア設計のビットストリーム・ファイルおよび Post-Place File を生成します。
2. Gowin_PicoRV32 ソフトウェア設計を更新します：
 - config.h でマクロ定義します：`#define BUILD_MODE BUILD_LOAD`
 - sections.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - 必要に応じてユーザーのソフトウェア設計を更新します
 - コンパイルして実行ファイルを生成します
3. 実際のアプリケーションに応じて merge_bit.bat を変更します。

merge_bit.bat を実行して、ハードウェア設計のビットストリーム・ファイルとソフトウェア設計の実行ファイルをマージし、新しいハードウェア設計のビットストリーム・ファイルを出力します(図 3-5)。

4. ソフトウェア設計を更新した後、ステップ 3~4 を再実行します。

図 3-5 ソフトウェア設計の出力とハードウェア設計の出力のマージ



```
C:\windows\system32\cmd.exe
C:\Users\liukai\Desktop\example>call merge_bit.exe picorv32_load_demo.bin gowin_picorv32.fs 32 gowin_picorv32.posp
----- GOWIN Merge Bit Tool -----
Read bit stream file gowin_picorv32.fs ...
Build bsrain initial value fusemap...
Read original bsrain initial value map...
Read location file gowin_picorv32.posp...
Bsrain R10[19] initial value convert to fusemap success.
Bsrain R28[16] initial value convert to fusemap success.
Bsrain R28[15] initial value convert to fusemap success.
Bsrain R28[14] initial value convert to fusemap success.
Bsrain R28[13] initial value convert to fusemap success.
Bsrain R28[10] initial value convert to fusemap success.
Bsrain R28[9] initial value convert to fusemap success.
Bsrain R28[8] initial value convert to fusemap success.
Bsrain R28[7] initial value convert to fusemap success.
Bsrain R28[6] initial value convert to fusemap success.
Bsrain R28[5] initial value convert to fusemap success.
Bsrain R10[4] initial value convert to fusemap success.
Bsrain R10[3] initial value convert to fusemap success.
Bsrain R10[2] initial value convert to fusemap success.
Bsrain R10[1] initial value convert to fusemap success.
Bsrain R10[0] initial value convert to fusemap success.
Replace new bsrain initial value map to file new_gowin_picorv32.fs...
Build bsrain initial value replace completed.
```

3.5.2 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。

3.6 サポートされるデバイス

- LittleBee ファミリーFPGA 製品
- Arora ファミリーFPGA 製品
- Arora V FPGA 製品

3.7 リファレンス・デザイン

リファレンス・デザインのダウンロード URL :

…¥tool¥merge_bit¥ref_design¥FPGA_RefDesign¥gowin_picorv32

…

¥tool¥merge_bit¥ref_design¥MCU_RefDesign¥picorv32_load_demo

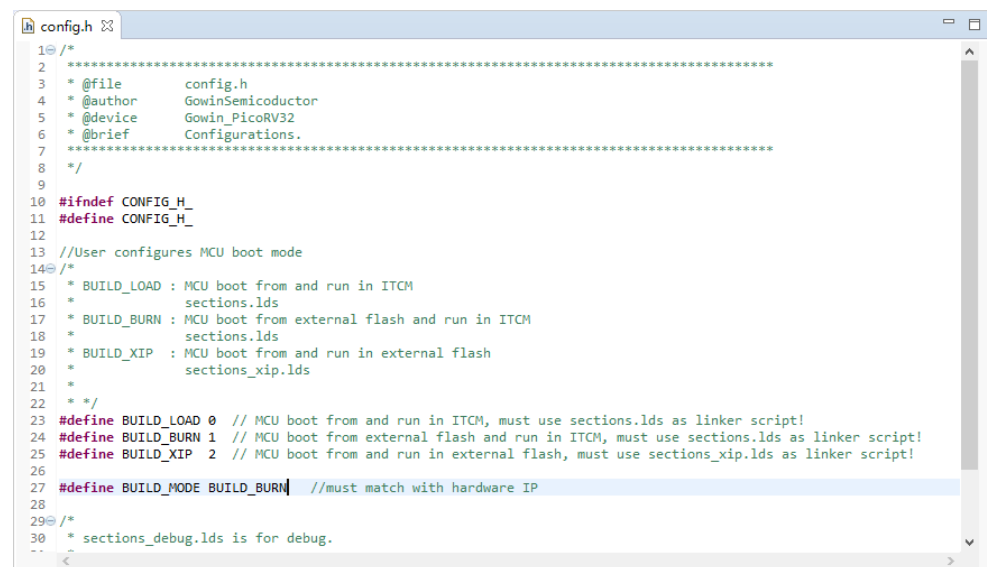
4 オフチップ SPI-Flash Memory のダウンロード方法

4.1 ソフトウェアの構成

4.1.1 起動モードの構成

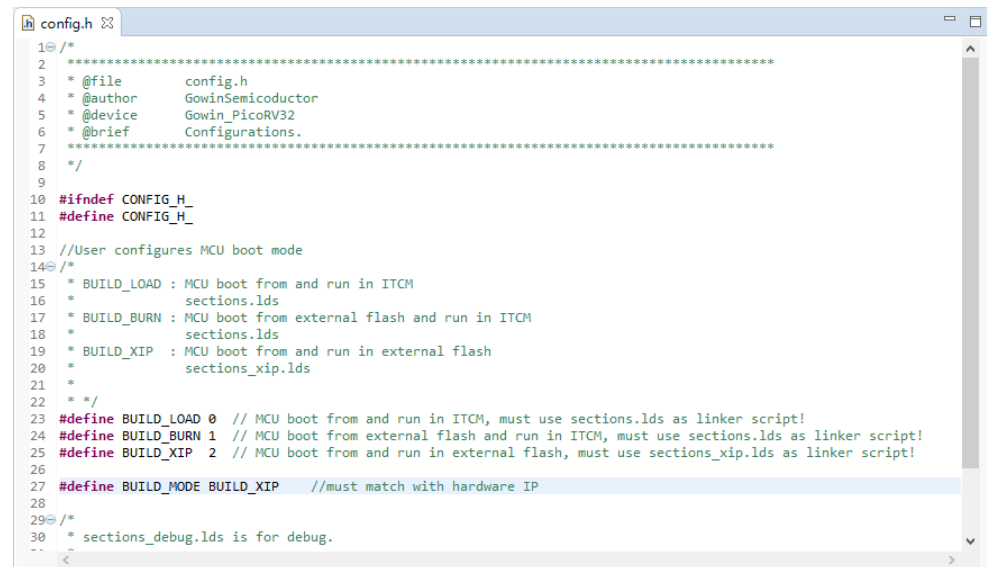
Gowin_PicoRV32 IP 設計において ITCM > Boot Mode > MCU boot from external Flash and run in ITCM が選択された場合、Gowin_PicoRV32 ソフトウェア設計における config.h では、以下のように定義します: #define BUILD_MODE BUILD_BURN。図 4-1 に示すとおりです。

図 4-1 起動モードの構成「BUILD_BURN」



Gowin_PicoRV32 IP 設計において ITCM > Boot Mode > MCU boot and run in external Flash が選択された場合、Gowin_PicoRV32 ソフトウェア設計における config.h では、以下のように定義します: #define BUILD_MODE BUILD_XIP。図 4-2 に示すとおりです。

図 4-2 起動モードの構成「BUILD_XIP」

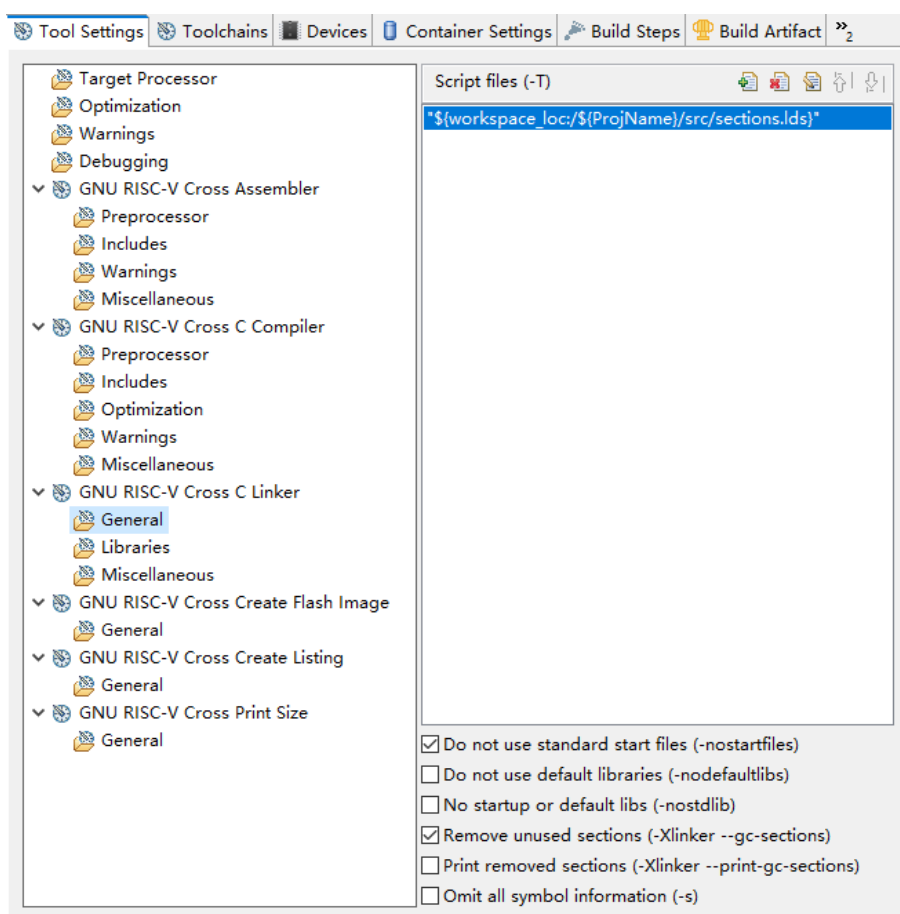


4.1.2 Flash リンカー・スクリプト・ファイルの構成

Gowin_PicoRV32 IP 設計において ITCM > Boot Mode > MCU boot from external Flash and run in ITCM が選択された場合、GMD ソフトウェアの Project Explorer ビューで、現在のプロジェクトを右クリックして、Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General で sections.lds を Flash リンカー・スクリプト・ファイルとして構成します(図 4-3)。

例えば、「\${workspace_loc}/\${ProjName}/src/sections.lds}」。

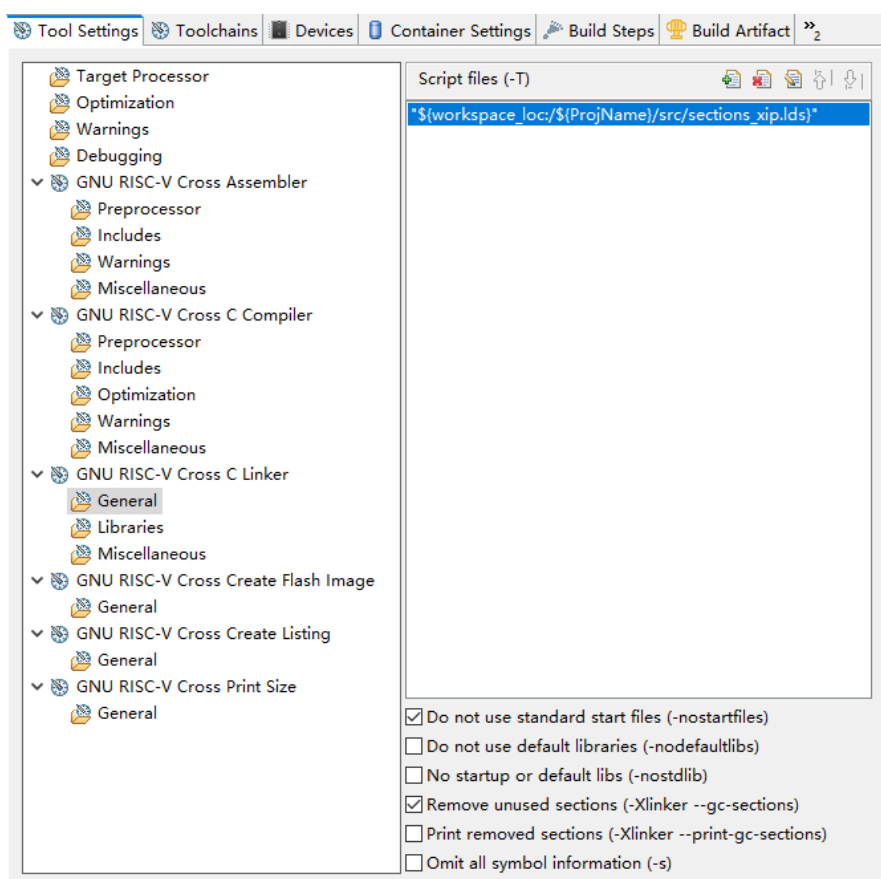
図 4-3 Flash リンカー・スクリプト・ファイルの構成



Gowin_PicoRV32 IP 設計において ITCM > Boot Mode > MCU boot and run in external Flash が選択された場合、GMD ソフトウェアの Project Explorer ビューで、現在のプロジェクトを右クリックして、Properties > C/C++ Build > Settings > Tool Settings > GNU RISC-V Cross C Linker > General で sections_xip.lds を Flash リンカー・スクリプト・ファイルとして構成します(図 4-4)。

例えば、「`${workspace_loc}/${ProjName}/src/sections_xip.lds`」。

図 4-4 Flash リンカー・スクリプト・ファイルの構成

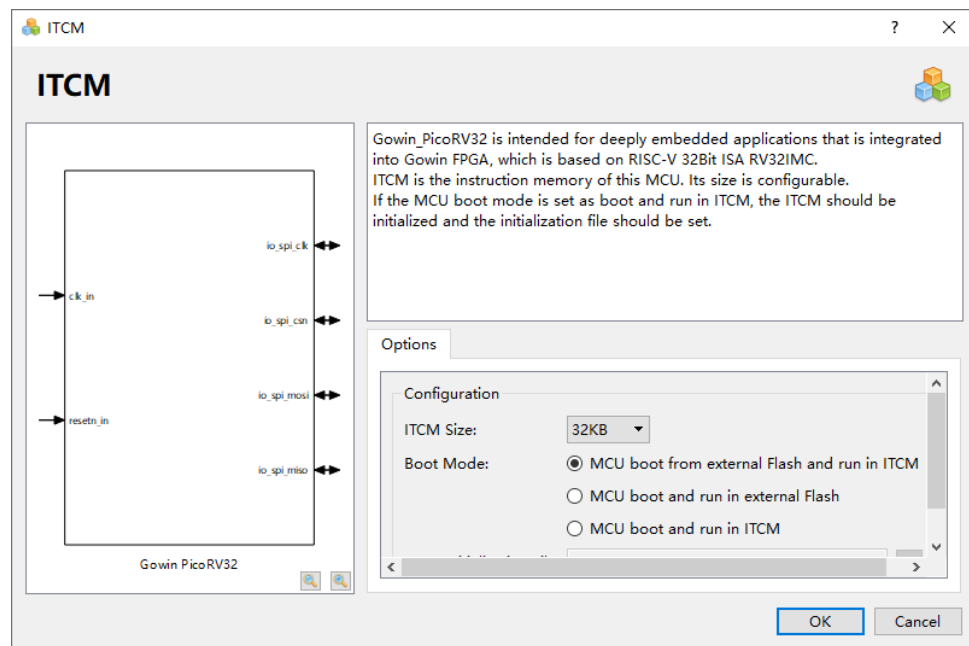


4.2 ハードウェアの構成

4.2.1 ITCM の構成

Gowin ソフトウェアの IP Core Generator で ITCM > Boot Mode > MCU boot from external Flash and run in ITCM または MCU boot and run in external Flash を選択します(図 4-5)。

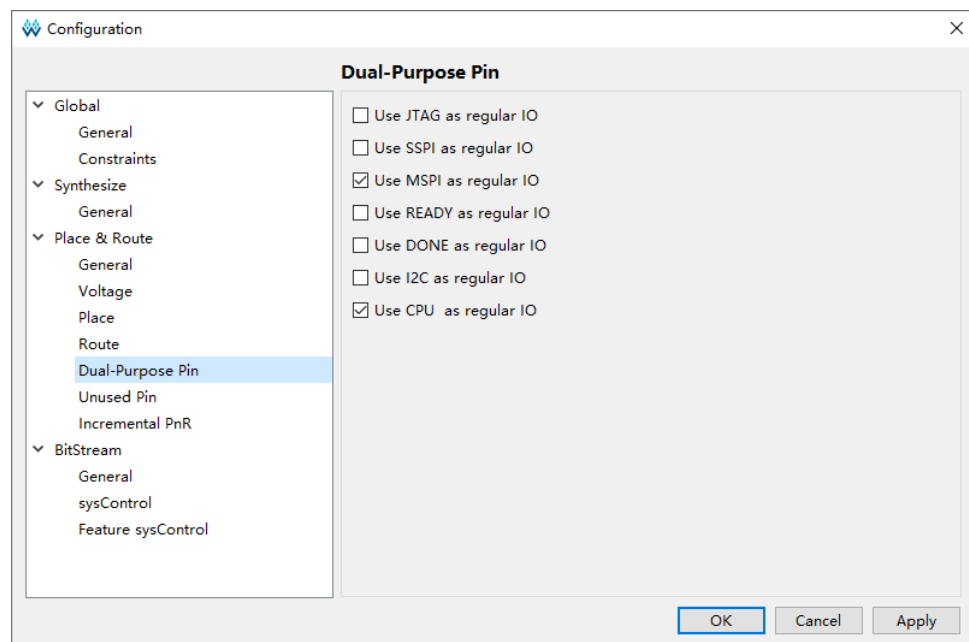
図 4-5 ITCM の構成



4.2.2 Dual-Purpose Pin の構成

Gowin ソフトウェアの「Place & Route > Dual-Purpose Pin」オプションで専用ピン(例えば、MSPI、CPU など)を通常の IO として構成します(図 4-6)。

図 4-6 Dual-Purpose Pin の構成



4.3 設計フロー

1. ハードウェア設計のフロー：

- Boot Mode > MCU boot from external Flash and run in ITCM または MCU boot and run in external Flash を選択します

- Gowin_PicoRV32 IP 設計を生成します
 - 合成、配置配線を経てオフチップ SPI-Flash Memory 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。
2. 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。
 3. Gowin_PicoRV32 ソフトウェア設計 :
 - MCU boot and run in external Flash :
config.h でマクロ定義します : #define BUILD_MODE BUILD_XIP
sections_xip.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - MCU boot from external Flash and run in ITCM
config.h でマクロ定義します : #define BUILD_MODE BUILD_BURN
sections.lds を Flash リンカー・スクリプト・ファイルとして構成します
 - コンパイルして実行ファイルを生成します
 4. **Programmer** を使用してソフトウェア設計の実行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](#))』を参照してください。

4.4.1 ハードウェア設計のビットストリーム・ファイルのダウンロード

Gowin_PicoRV32 ハードウェア設計でオフチップ SPI-Flash Memory によるダウンロード用のビットストリーム・ファイルを生成します。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 

を開きます。

開発ボード上のデバイスが **Arora** ファミリーFPGA 製品の場合、ダウンロード・オプションの構成は図 4-7 に示すとおりです。

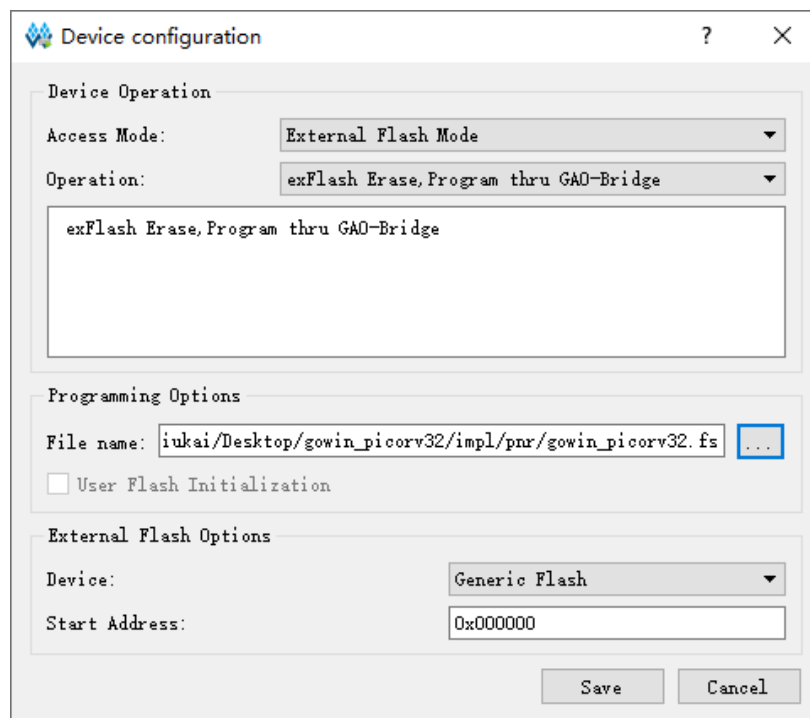
- 「Access Mode」のドロップダウン・リストから「External Flash Mode」を選択します。
- Operation のドロップダウン・リストから「exFlash Erase, Program thru

GAO-Bridge」または「exFlash Erase, Program, Verify thru GAO-Bridge」を選択します。

- 「Programming Options > File name」で、ダウンロードする必要のあるビットストリーム・ファイルをインポートします。
- 「External Flash Options > Device」オプションで「Generic Flash」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 4-7 Arora ファミリーの Device configuration



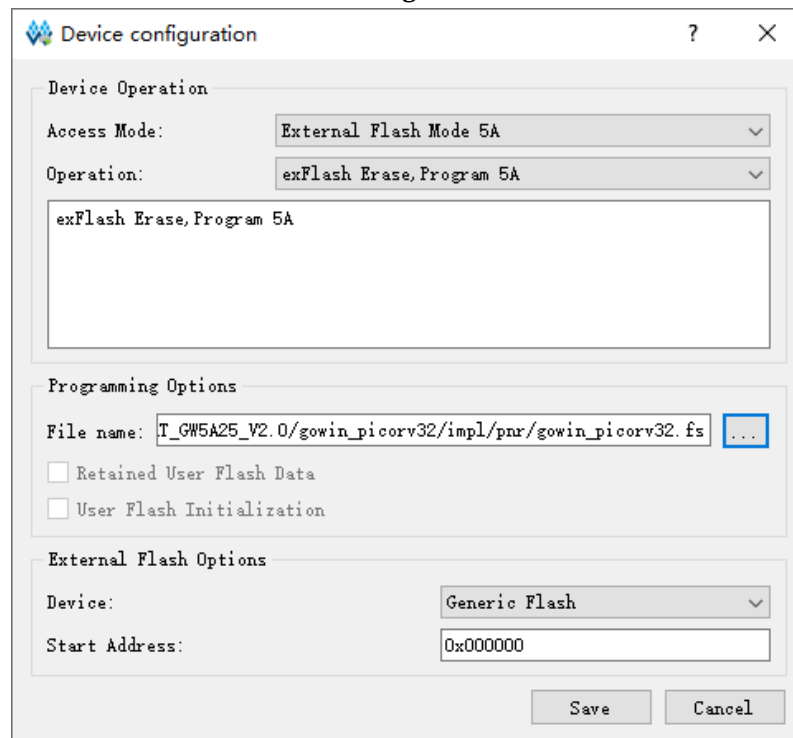
開発ボード上のデバイスが Arora V FPGA 製品の場合、ダウンロード・オプションの構成は図 4-8 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「External Flash Mode 5A」を選択します。
- 「Operation」のドロップダウン・リストから「exFlash Erase, Program 5A」または「exFlash Erase, Program, Verify 5A」を選択します。
- 「Programming Options > File name」で、ダウンロードする必要のあるビットストリーム・ファイルをインポートします。
- 「External Flash Options > Device」オプションで「Generic Flash」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 4-8 Arora V の Device Configuration



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 実行ファイルのダウンロード

Gowin_PicoRV32 ソフトウェア設計が完了したら、Programmer を使用してソフトウェア設計の実行ファイルをダウンロードします。

GMD ソフトウェアのメニューバー上の「Gowin>Programmer」またはツールバー上の「Programmer」()を選択して Programmer を開きます。

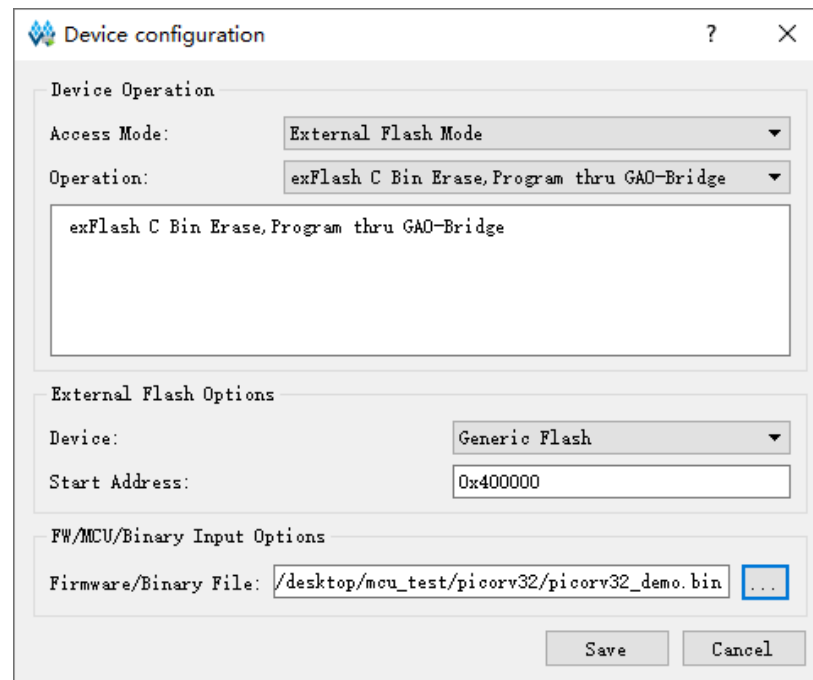
Programmer のメニューバー上の「Edit > Configure Device」またはツールバー上の「Configure Device」()をクリックして、「Device configuration」ダイアログボックスを開きます。

開発ボード上のデバイスが Arora ファミリーFPGA 製品の場合、ダウンロード・オプションの構成は図 4-9 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「External Flash Mode」を選択します。
- Operation のドロップダウン・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」でソフトウェア設計の実行ファイルを選択します。
 - 「External Flash Options > Device」オプションで「Generic Flash」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 4-9 Arora ファミリーの Device configuration

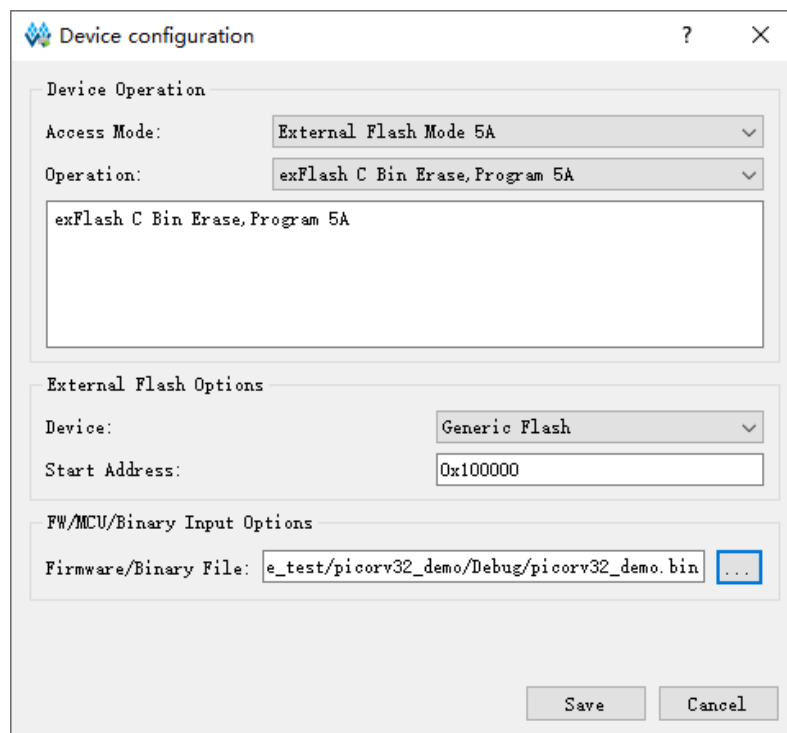


開発ボード上のデバイスが Arora V FPGA 製品の場合、ダウンロード・オプションの構成は図 4-10 に示すとおりです。

- 「Access Mode」のドロップダウン・リストから「External Flash Mode 5A」を選択します。
- 「Operation」のドロップダウン・リストから「exFlash C Bin Erase, Program 5A」または「exFlash C Bin Erase, Program, Verify 5A」を選択します。
- 「FW/MCU/Binary Input Options > Firmware/Binary File」でソフトウェア設計の実行ファイルを選択します。
- 「External Flash Options > Device」オプションで「Generic Flash」を選択します。
- 「External Flash Options > Start Address」は、「0x600000」（138K、75K デバイス）、「0x400000」（60K デバイス）、「0x100000」（25K、15K デバイス）に構成します。

最後に「Save」をクリックします。

図 4-10 Arora V の Device configuration



Device configuration が完了したら、Programmer のツールバー上の Program/Configure 「」をクリックしてソフトウェア設計の実行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

- Arora ファミリーFPGA 製品
- Arora V FPGA 製品

