Gowin_EMPU_M1 Download
Reference Design

IPUG532-2.4E, 01/17/2025

Disclaimer

GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
02/19/2019	1.0E	Initial version published.
07/18/2019	1.1E	● Supports the automated merging tool used for MCU hardware design and software programming design. ● MCU supports off-chip SPI-Flash downloading and startup.
08/18/2019	1.2E	● MCU hardware design and software programming design support extended peripheral: DD3 Memory. ● Fixed known issues of ITCM, DTCM Size and IDE.
09/27/2019	1.3E	The description of software configuration updated.
01/16/2020	1.4E	● MCU hardware design and software programming design supports PSRAM. ● Updated MCU compiling software GMD V1.0. ● RTOS reference design updated. ● Hardware and software reference design of AHB2 and APB2 extension bus interface added.
03/03/2020	1.5E	● Known issues of ITCM, DTCM Size and IDE fixed. ● The FPGA devices of GW2A-18C/GW2AR-18C/GW2A-55C supported.
06/12/2020	1.6E	● MCU supports External instruction memory. ● MCU supports External data memory. ● 6 AHB bus interfaces extended. ● 16 APB bus interfaces extended. ● GPIO supports multiple interface types. ● I^2C supports multiple interface types. ● Merge_bit tool supports GowinSynthesis to parse the rules of naming.
01/25/2021	1.7E	● The reference design of GW1N-9C, GW2A-18C, GW2A-55C (Version C) updated. ● The download auxiliary tools called Merge_bit and make_hex updated.
07/21/2021	1.8E	● GW1N-9C/GW1NR-9C supports embedded UserFlash as instruction memory. ● Merge_bit tool updated. ● The SynplifyPro deleted. ● The version of FPGA and MCU software updated.
10/12/2021	1.9E	GW2AN-9X/GW2AN-18X merge_bit download deleted.
05/11/2023	2.0E	Arora V FPGA products supported.
07/21/2023	2.1E	● Tested software version updated ● The reference design of Off-chip SPI-Flash Memory Download Method added.
03/07/2024	2.2E	● bootload updated to support multiple Flash start addresses. ● merge_bit tool updated to support GW2AN-18X/9X series of FPGA products.
06/28/2024	2.3E	merge_bit tool updated to support Arora V FPGA products.
01/17/2025	2.4E	Tested software version updated.

Contents

List of Figures

List of Tables

1 Download Methods

Gowin_EMPU_M1 provides four download methods of hardware design and software programming design:

1. Use the executable program files generated by software programming design as the ITCM initial value of hardware design.
   a) Gowin_EMPU_M1 software programming design generates software programming design BIN files.
   b) Use make_hex tool to convert the BIN files to four files in hexadecimal format: itcm0, itcm1, itcm2, and itcm3.
   c) Use itcm0, itcm1, itcm2, and itcm3 as the ITCM initial value files in hardware design to read.
   d) Synthesize, place & route to generate the bitstream files in hardware design including software programming design and hardware design.
   e) Use Programmer, the download tool, to download the bitstream files in hardware design.
2. Merge the BIN files generated in software design and the bitstream files generated in hardware design.
   a) Gowin_EMPU_M1 hardware design generates bitstream files in hardware design.
   b) Gowin_EMPU_M1 software programming design generates BIN files in software design.
   c) Use merge_bit tool to merge the BIN files in software design and bitstream files in hardware design.
   d) Generate new bitstream files in hardware design after merging the software design and the hardware design.
   e) Use Programmer, the download tool, to download the bitstream files in hardware design after merging.
3. Use off-chip SPI-Flash memory to download the BIN files generated by software programming design.
   a) In Gowin_EMPU_M1 hardware design, configure ITCM Size and

select different bootload as the initial value of ITCM according to ITCM Size.

b) Gowin_EMPU_M1 hardware design generates bitstream files with the off-chip SPI-Flash memory that provides download functions.

c) Use Programmer, the download tool, to download the bitstream files in hardware design.

d) Gowin_EMPU_M1 software programming design generates BIN files in software design.

e) Use Programmer, the download tool, to download the BIN files generated by software programming design.

4. Use embedded UserFlash to download the BIN files generated by software programming design.

   a) In Gowin_EMPU_M1 hardware design, ITCM Select is configured as External Instruction Memory.

   b) Instantiate UserFlash Controller (GW1N/R-9 FLASH608K) with Memory Map as the instruction memory of Gowin_EMPU_M1.

   c) Synthesize, place & route Gowin_EMPU_M1 hardware design to generate bitstream files in hardware design.

   d) Build and link Gowin_EMPU_M1 software programming design to generate BIN files in software design.

   e) Use Programmer, the download tool, to download the bitstream files in hardware design and BIN files in software design.

2 Software Programming Output as ITCM Initialization Value

2.1 Tool

…\tool\make_hex\bin\make_hex.exe

Click this link to access the above software tools: cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip

2.2 Command Parameter

make_hex.exe bin-file, and the bin-file represents the binary file of the software programming design.

2.3 Software Configuration

Use software programming design to generate BIN files.

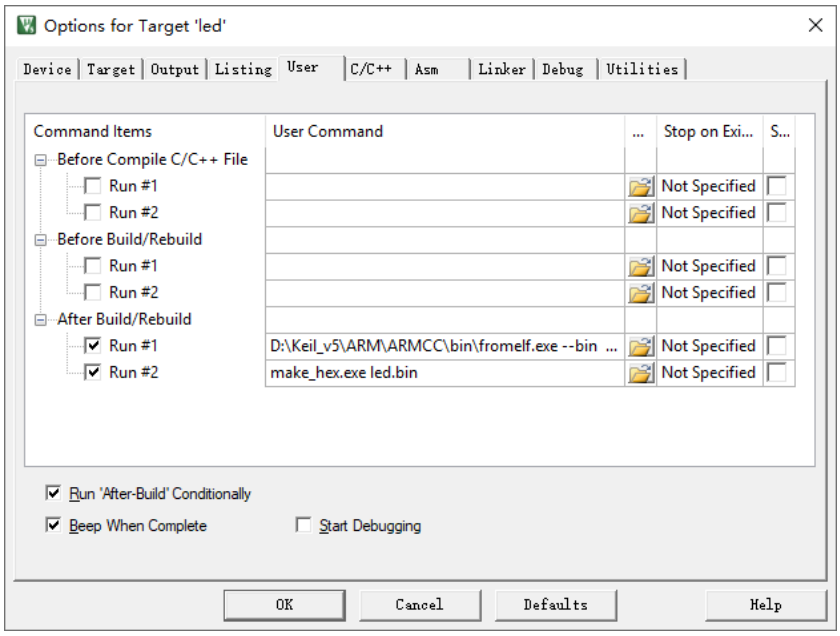
Use make_hex tool to convert the BIN files to four files in a hexadecimal format: itcm0, itcm1, itcm2, and itcm3.

In ARM Keil MDK (tested software version V5.26), configure make_hex.exe as external tool, as shown in Figure 2-1.

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

Automatically call make_hex.exe tool to generate the BIN files and four files in a hexadecimal format when the software is building.

Figure 2-1 Configure External Tool

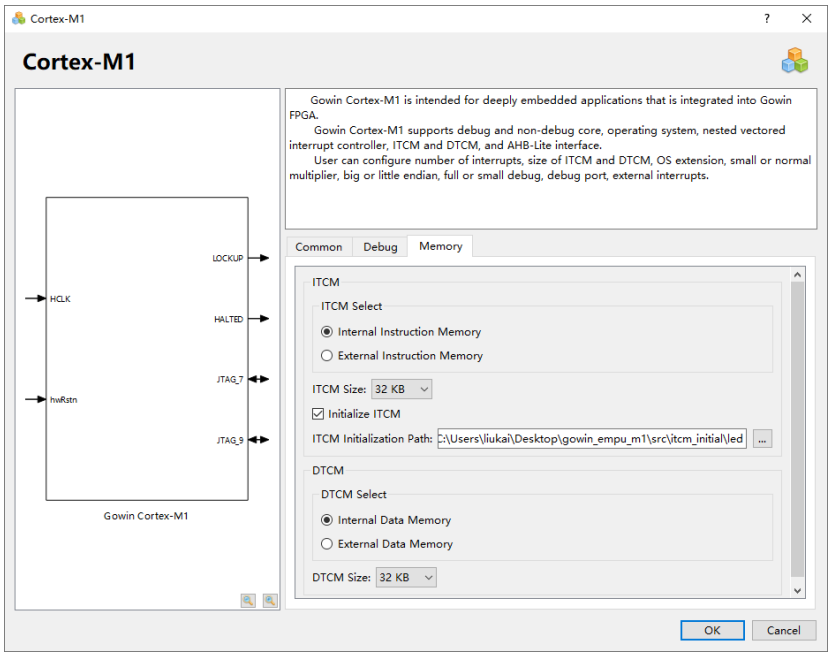


2.4 Hardware Configuration

In IP Core Generator integrated in Gowin Software:

- Select "Cortex-M1 > Memory > ITCM > ITCM Select > Internal Instruction Memory".
- Select "Cortex-M1 > Memory > ITCM > Initialize ITCM".
- In "Cortex-M1 > Memory > ITCM > ITCM Initialization Path", import the path of the four files itcm0, itcm1, itcm2, and itcm3 in hexadecimal format as the initial value of ITCM, as shown in Figure 2-2.
- Import itcm0、itcm1、itcm2、itcm3 as the initial value of ITCM, as well as Gowin_EMPU_M1 hardware design generated after the external tool configuration of Cortex-M1 and AHB/APB in IP Core Generator. The hardware design includes software programming design.
- Gowin_EMPU_M1 hardware design, which contains the software programming design, is generated after importing itcm0, itcm1, itcm2, and itcm3 as the initial ITCM values and completing the configuration of Cortex-M1 core system and bus peripheral system options in IP Core Generator.

Figure 2-2 Configure ITCM Initialization



2.5 Design Flow

1. Build the software programming design in ARM Keil MDK (tested software version V5.26) and GMD ((tested software version V1.2) to generate four files itcm0, itcm1, itcm2, and itcm3 in hexadecimal format.
2. Use IP Core Generator in Gowin Software to generate Gowin_EMPU_M1 hardware design. and itcm0, itcm1, itcm2, and itcm3 generated by software programming design are used as the initial value of ITCM in hardware design.
3. Instantiate Gowin_EMPU_M1 Top Module, and connect user design.
4. Add physical and timing Constraints.
5. Use GowinSynthesis, the synthesis tool, to synthesize.
6. Use Place & Route tool to generate the bitstream files containing software programming design.
7. Use Programmer, the download tool, to download the bitstream files in hardware design.

2.6 Devices Supported

- GW1N/R-9
- GW2AN-9X/18X, GW2A/R/NR-18, GW2A/N-55
- GW5A/T/ST/S-138, GW5AT-75, GW5A/R/S-25, GW5A/T-60, GW5AT/RT-15

2.7 Reference Design

Click this link to access the reference design:

…\tool\make_hex\ref_design\FPGA_RefDesign\gowin_empu_m1

…\tool\make_hex\ref_design\MCU_RefDesign\cm1_demo\project\led

…\tool\make_hex\ref_design\MCU_RefDesign\cm1_demo\project\printf

3 Merge Software Design and Hardware Design

3.1 Tool

…\tool\merge_bit\bin\merge_bit.exe

Click this link to access the above software tools: cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip

3.2 Command Parameters

Software tool command parameter: merge_bit.exe bin-file fs-file itcm_size posp-file.

For the descriptions of command parameters, see Table 3-1.

Table 3-1 Command Parameters

Parameter	Description
bin-file	Software programming Binary file
itcm_size	ITCM Size (KB) For example, if ITCM Size is set to 64KB, this parameter is 64.
fs-file	Hardware design bitstream file
posp-file	Post-Place File

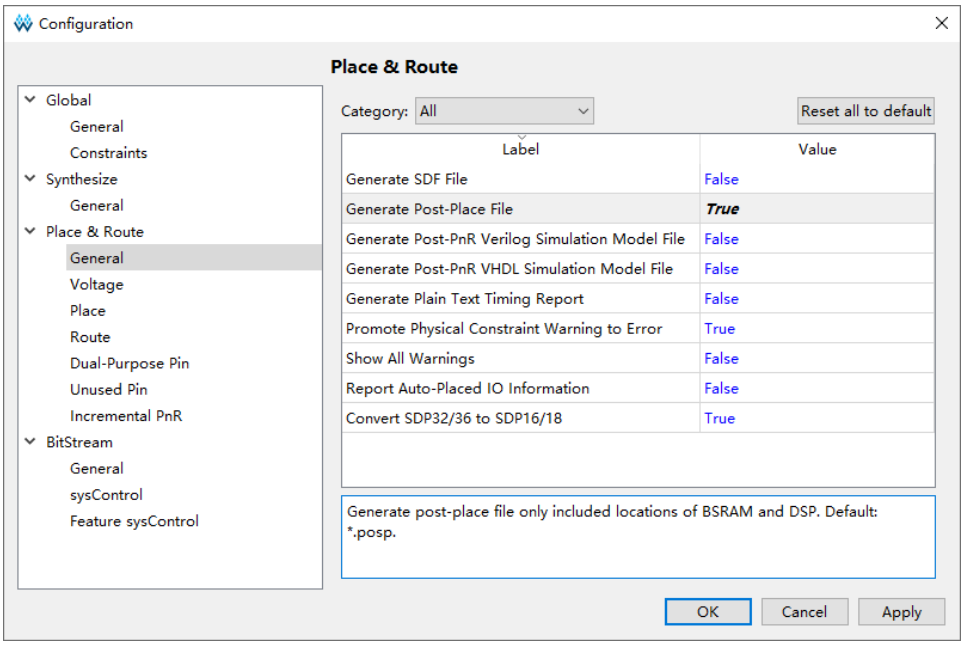
Merge the BIN files in software design and the bitstream files in hardware design.

During the use of merge_bit.bat, you can modify the parameters, such as posp-file, itcm_size, bin-file, fs-file, according to your requirements.

3.3 Hardware Configuration

In "Place & Route > General > Generate Post-Place File", set "Generate Post-Place File" to "True" to generate Post-Place File as posp-file, as shown in Figure 3-1.

Figure 3-1 Post-Place File Configuration

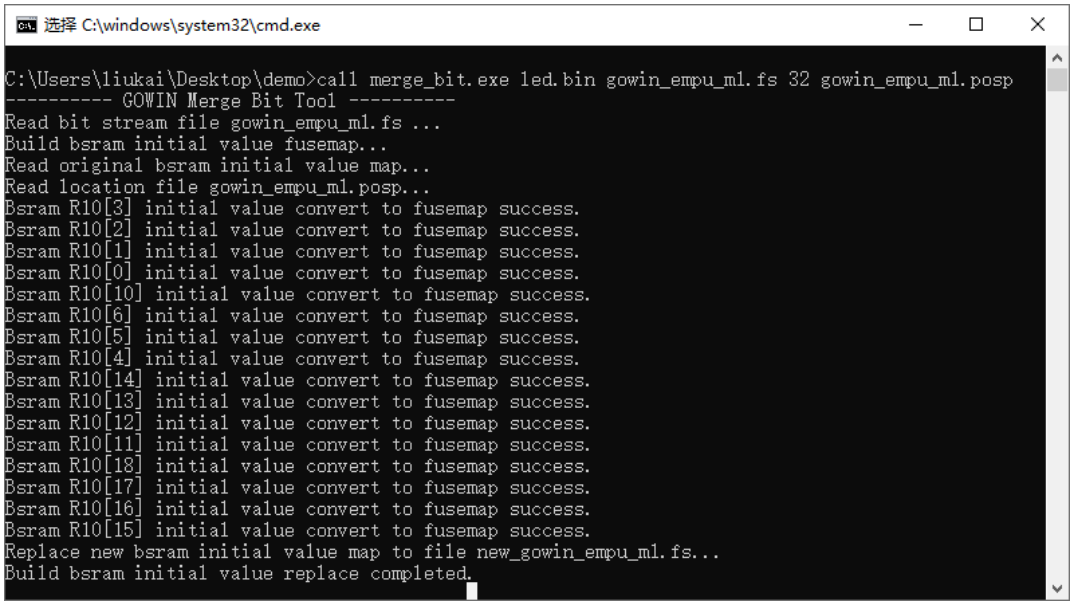


3.4 Design Flow

3.4.1 Merge

1. Gowin_EMPU_M1 hardware design generates bitstream files in hardware design and Post-Place File.
2. Gowin_EMPU_M1 software programming design generates BIN files in software design.
3. Run merge_bit.bat, merge the bitstream files in hardware design and the BIN files in software design to generate new bitstream files, as shown in Figure 3-2.

Figure 3-2 Merge Software Programming Design and Hardware Design



3.4.2 Download

After merging, use Programmer to download the new bitstream files in hardware design.

For the use of Gowin Programmer, please see *SUG502, Gowin Programmer User Guide*.

3.5 Devices Supported

- GW1N/R-9
- GW2AN-9X/18X, GW2A/R/NR-18, GW2A/N-55
- GW5A/T/ST/S-138, GW5AT-75, GW5A/R/S-25, GW5A/T-60, GW5AT/RT-15

3.6 Reference Design

Click this link to access the reference design:

…\tool\merge_bit\ref_design\FPGA_RefDesign\gowin_empu_m1

…\tool\merge_bit\ref_design\MCU_RefDesign\cm1_demo\project\led

…\tool\merge_bit\ref_design\MCU_RefDesign\cm1_demo\project\printf

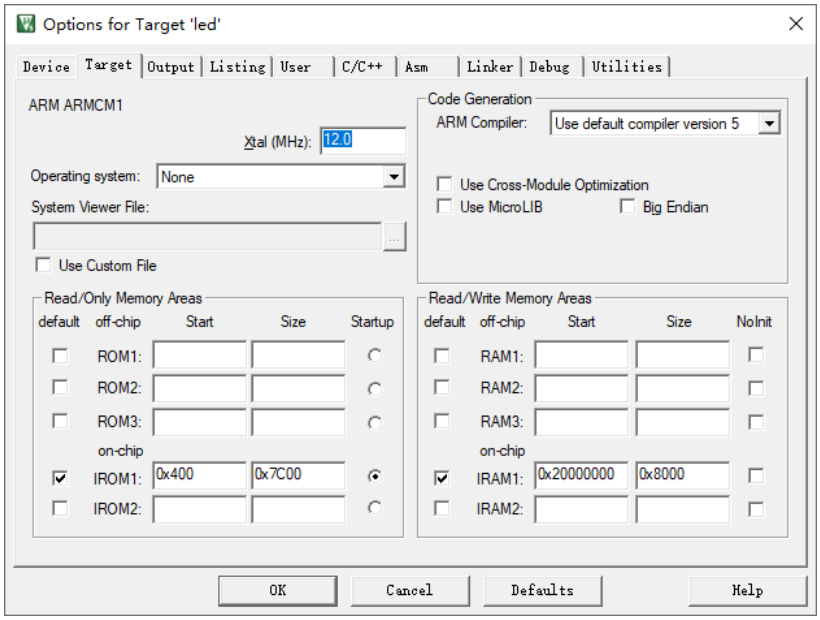
4 Off-chip SPI-Flash Memory Download Method

4.1 Software Configuration

In Gowin_EMPU_M1 Software Programming Design:

If you use ARM Keil MDK (tested software version V5.26), set the IROM1 start address to 0x400 and set the IROM1 Size according to the actual ITCM Size hardware configuration. For example, ITCM Size 32KB, then set IROM1 to 0x7C00, as shown in Figure 4-1.

Figure 4-1 ROM Start Address and Size Configuration



If you use GMD (tested software version V1.2), select the Flash linker GOWIN_M1_flash_burn.ld, and set the Flash start address "FLASH ORIGIN" to 0x00000400.

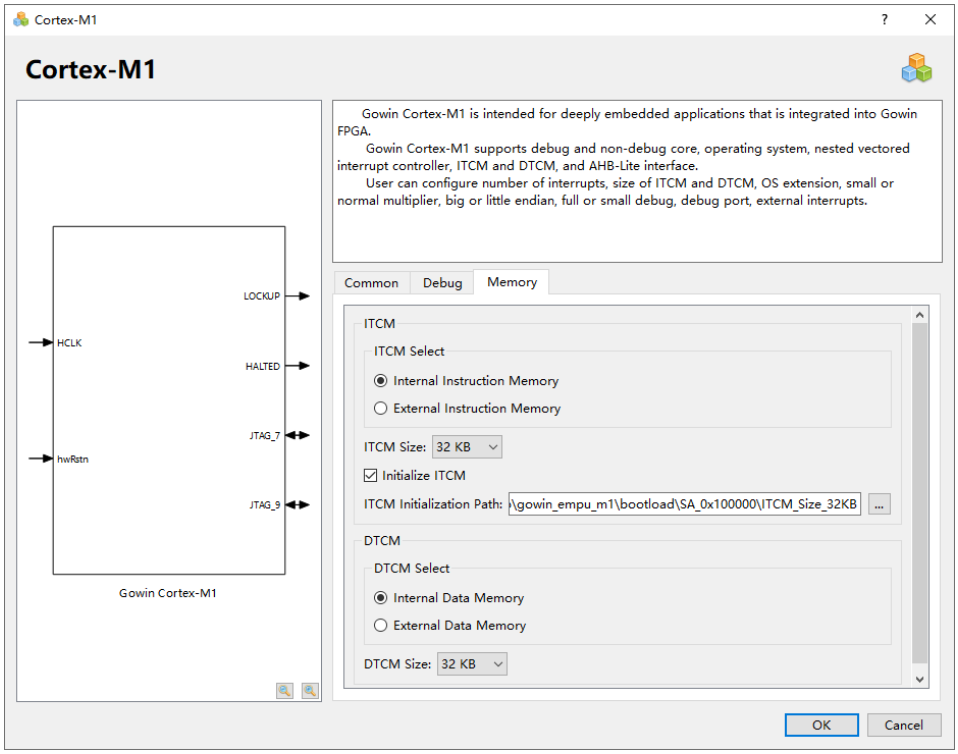
4.2 Hardware Configuration

Use IP Core Generator tool in Gowin Software to configure and generate Gowin_EMPU_M1 hardware design:

- Select "Internal Instruction Memory" as the instruction memory of Gowin_EMPU_M1
- Select "ITCM Size"
- Select "Initialize ITCM"
- Select a bootload as the initial value of ITCM based on different Flash start addresses and various ITCM sizes. Import bootload path to ITCM Initialization Path.

Configure ITCM as shown in Figure 4-2.

Figure 4-2 Configure ITCM



The bootload corresponding to different Flash start addresses and ITCM Sizes are as show in Table 4-1.

The recommendations are as follows:

- For GW5AT/RT-15, choose 0x100000 as the Flash start address.
- For GW5A/R/S-25, choose 0x100000 as the Flash start address.
- For GW5A/T-60, choose 0x400000 as the Flash start address.
- For GW5A/T/ST/S-138 and GW5AT-75, choose 0x600000 as the Flash start address.
- For GW2AN-9X/18X, GW2A/R/NR-18, and GW2A/N-55, choose 0x400000 as the Flash start address

Table 4-1 Bootload

Flash Start Address	ITCM Size (KByte)	bootload
0x100000	2	…\bootload\SA_0x100000\ITCM_Size_2KB
	4	…\bootload\SA_0x100000\ITCM_Size_4KB
	8	…\bootload\SA_0x100000\ITCM_Size_8KB
	16	…\bootload\SA_0x100000\ITCM_Size_16KB
	32	…\bootload\SA_0x100000\ITCM_Size_32KB
	64	…\bootload\SA_0x100000\ITCM_Size_64KB
	128	…\bootload\SA_0x100000\ITCM_Size_128KB
	256	…\bootload\SA_0x100000\ITCM_Size_256KB
0x200000	2	…\bootload\SA_0x200000\ITCM_Size_2KB
	4	…\bootload\SA_0x200000\ITCM_Size_4KB
	8	…\bootload\SA_0x200000\ITCM_Size_8KB
	16	…\bootload\SA_0x200000\ITCM_Size_16KB
	32	…\bootload\SA_0x200000\ITCM_Size_32KB
	64	…\bootload\SA_0x200000\ITCM_Size_64KB
	128	…\bootload\SA_0x200000\ITCM_Size_128KB
	256	…\bootload\SA_0x200000\ITCM_Size_256KB
0x300000	2	…\bootload\SA_0x300000\ITCM_Size_2KB
	4	…\bootload\SA_0x300000\ITCM_Size_4KB
	8	…\bootload\SA_0x300000\ITCM_Size_8KB
	16	…\bootload\SA_0x300000\ITCM_Size_16KB
	32	…\bootload\SA_0x300000\ITCM_Size_32KB
	64	…\bootload\SA_0x300000\ITCM_Size_64KB
	128	…\bootload\SA_0x300000\ITCM_Size_128KB
	256	…\bootload\SA_0x300000\ITCM_Size_256KB
0x400000	2	…\bootload\SA_0x400000\ITCM_Size_2KB
	4	…\bootload\SA_0x400000\ITCM_Size_4KB
	8	…\bootload\SA_0x400000\ITCM_Size_8KB
	16	…\bootload\SA_0x400000\ITCM_Size_16KB
	32	…\bootload\SA_0x400000\ITCM_Size_32KB
	64	…\bootload\SA_0x400000\ITCM_Size_64KB
	128	…\bootload\SA_0x400000\ITCM_Size_128KB
	256	…\bootload\SA_0x400000\ITCM_Size_256KB
0x500000	2	…\bootload\SA_0x500000\ITCM_Size_2KB
	4	…\bootload\SA_0x500000\ITCM_Size_4KB
	8	…\bootload\SA_0x500000\ITCM_Size_8KB
	16	…\bootload\SA_0x500000\ITCM_Size_16KB
	32	…\bootload\SA_0x500000\ITCM_Size_32KB
	64	…\bootload\SA_0x500000\ITCM_Size_64KB
	128	…\bootload\SA_0x500000\ITCM_Size_128KB
	256	…\bootload\SA_0x500000\ITCM_Size_256KB

Flash Start Address	ITCM Size (KByte)	bootload
	512	…\bootload\SA_0x500000\ITCM_Size_512KB
0x600000	2	…\bootload\SA_0x600000\ITCM_Size_2KB
	4	…\bootload\SA_0x600000\ITCM_Size_4KB
	8	…\bootload\SA_0x600000\ITCM_Size_8KB
	16	…\bootload\SA_0x600000\ITCM_Size_16KB
	32	…\bootload\SA_0x600000\ITCM_Size_32KB
	64	…\bootload\SA_0x600000\ITCM_Size_64KB
	128	…\bootload\SA_0x600000\ITCM_Size_128KB
	256	…\bootload\SA_0x600000\ITCM_Size_256KB
	512	…\bootload\SA_0x600000\ITCM_Size_512KB

4.3 Design Flow

1. Gowin_EMPU_M1 hardware design configuration:
   - Select "Internal Instruction Memory"
   - Select "ITCM Size"
   - Select "Initialize ITCM"
   - Select a bootload as the initial value of ITCM according to different Flash start addresses and ITCM Sizes
2. Generate Gowin_EMPU_M1 hardware design
3. Synthesize, place & route to generate bitstream files in hardware design with the off-chip SPI-Flash memory that provides download functions.
4. Configure Device configuration in Programmer to download the bitstream files in hardware design.
5. Gowin_EMPU_M1 software programming design generates BIN files in software design.
6. Configure Device configuration in Programmer to download the BIN file in software design.

4.4 Download

For the use of Gowin Programmer, please see *SUG502, Gowin Programmer User Guide*.

4.4.1 Download Bitstream Files in Hardware Design

Gowin_EMPU_M1 hardware design generates bootload as the initial value of ITCM and bitstream files with the off-chip SPI-Flash memory that provides download functions. Use Programmer, the download tool, to download the bitstream files in hardware design.

Select "Tools > Programmer" in the menu bar or "Programmer" () in the tool bar in Gowin Software to open Programmer, the download tool.

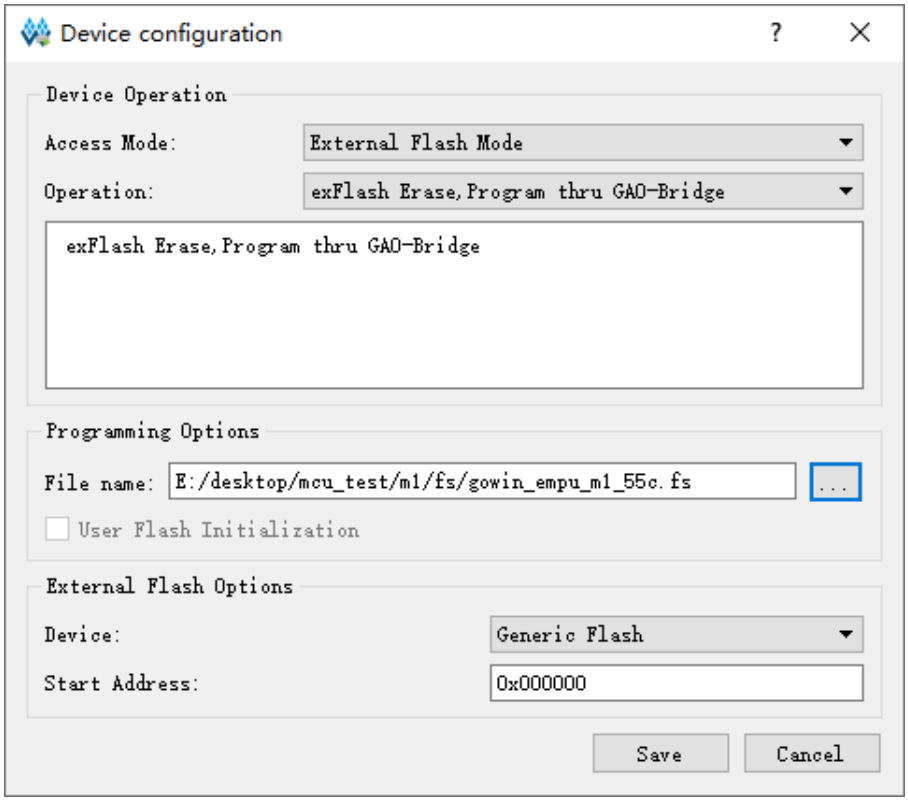
Select "Edit > Configure Device" in the menu bar or "Configure

Device" ( ) in the tool bar to open the "Device configuration".

For GW2AN-9X/18X, GW2A/R/NR-18, and GW2A/N-55, the option configuration is as shown in Figure 4-4.

- Select "External Flash Mode" in "Access Mode" drop-down list.
- Select "exFlash Erase, Program thru GAO-Bridge" or "exFlash Erase, Program, Verify thru GAO-Bridge" in "Operation" drop-down list.
- Import the hardware design bitstream file to download in "Programming Options > File name" option.
- In "External Flash Options > Device", select " Generic Flash".
- In "External Flash Options > Start Address", set the start address as 0x000000.
- Click "Save" to complete the configuration, as shown in Figure 4-3.

Figure 4-3 Device Configuration for GW2AN-9X/18X, GW2A/R/NR-18, and GW2A/N-55

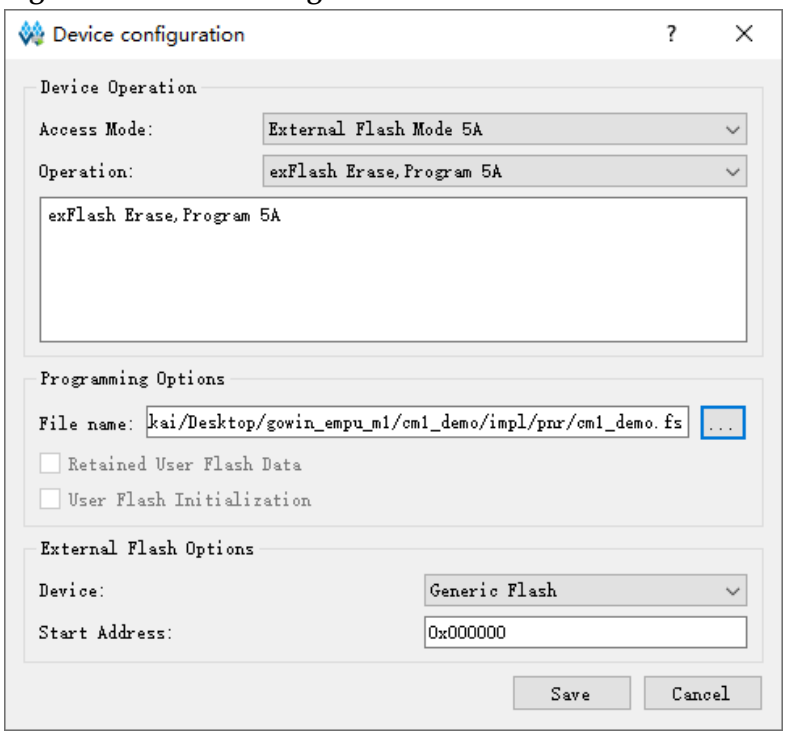


For Arora V, the option configuration is as shown in Figure 4-5.

- Select "External Flash Mode 5A" in "Access Mode" drop-down list.
- Select "exFlash Erase, Program 5A" or "exFlash Erase, Program, Verify 5A" in "Operation" drop-down list.
- Import the hardware design bitstream file to download in "Programming Options > File name" option.
- In "External Flash Options > Device", select " Generic Flash".

- In "External Flash Options > Start Address", set the start address as 0x000000.
- Click "Save" to complete the configuration, as shown in Figure 4-5.

Figure 4-4 Device Configuration for Arora V



After device configuration, click "Program/Configure" ( ) in the Programmer tool bar to download bitstream files in hardaware design.

4.4.2 Download BIN File in Software Programing Design

After downloading the bitstream files in Gowin_EMPU_M1 hardware design, download the BIN files in software design with Programmer.

Open Programmer, the download tool, in Gowin Software or under the software installation path.

Click "Edit > Configure Device" in the menu bar or "Configure Device" ( ) in the tool bar to open the "Device configuration".

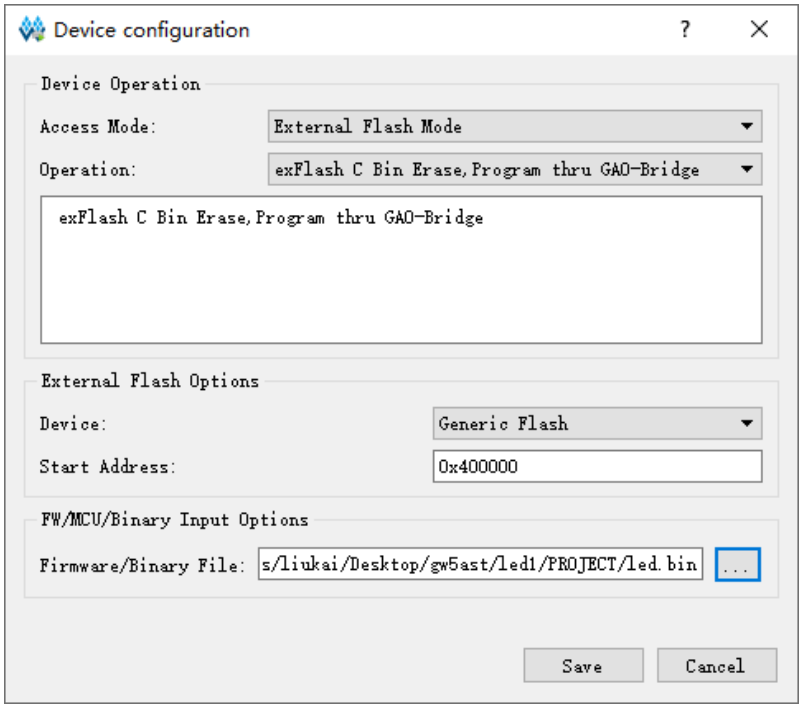
For GW2AN-9X/18X, GW2A/R/NR-18, and GW2A/N-55, the option configuration is as shown in Figure 4-6.

- Select "External Flash Mode" in "Access Mode" drop-down list.
- Select "exFlash C Bin Erase, Program thru GAO-Bridge" or "exFlash C Bin Erase, Program, Verify thru GAO-Bridge" in "Operation" drop-down list.
- Select "FW/MCU Input Options > Firmware/Binary File" to import the BIN files in software design to download.
- In "External Flash Options > Device", select "Generic Flash".
- In "External Flash Options > Start Address", you can set according to

the selected Flash start address in bootload, such as 0x400000.

- Click "Save" to complete the configuration as shown in Figure 4-5.

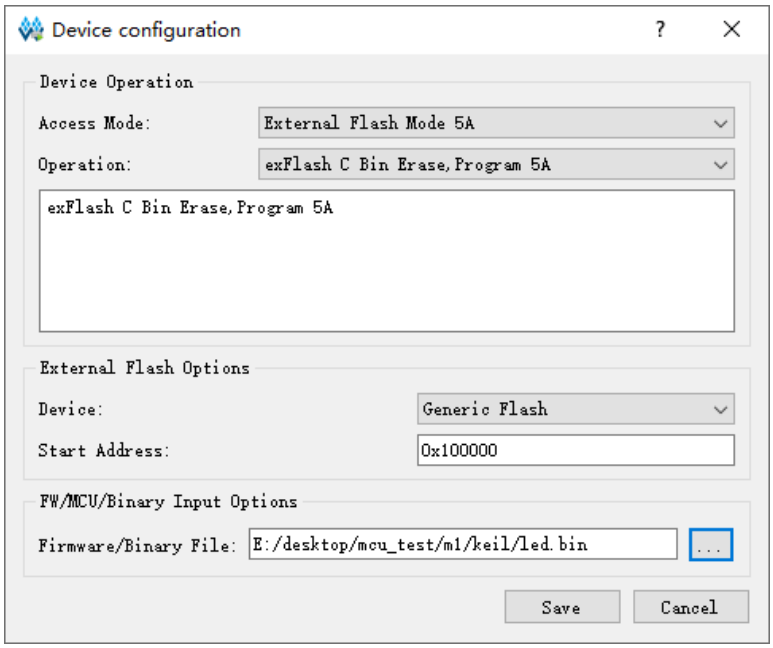
Figure 4-5 Device Configuration for GW2AN-9X/18X, GW2A/R/NR-18, GW2A/N-55



For Arora V, the option configuration is as shown in Figure 4-7.

- Select "External Flash Mode 5A" in "Access Mode" drop-down list.
- Select "exFlash C Bin Erase, Program 5A" or "exFlash C Bin Erase, Program, Verify 5A" in "Operation" drop-down list.
- Select "FW/MCU Input Options > Firmware/Binary File" to import the BIN files in software design to download.
- In "External Flash Options > Device", select "Generic Flash".
- In "External Flash Options > Start Address", you can set according to the selected Flash start address in bootload, such as "0x600000" (GW5A/T/ST/S-138, GW5AT-75), "0x400000" (GW5A/T-60), "0x100000" (GW5A/R/S-25, GW5AT/RT-15).
- Click "Save" to complete the configuration as shown in Figure 4-7.

Figure 4-6 Device Configuration for Arora V



After device configuration, click "Program/Configure" ( ) in the Programmer tool bar to download BIN files in software design.

4.5 Devices Supported

- GW2AN-9X/18X, GW2A/R/NR-18, GW2A/N-55
- GW5A/T/ST/S-138, GW5AT-75, GW5A/R/S-25, GW5A/T-60, GW5AT/RT-15

4.6 Reference Design

Click this link to access the reference design:

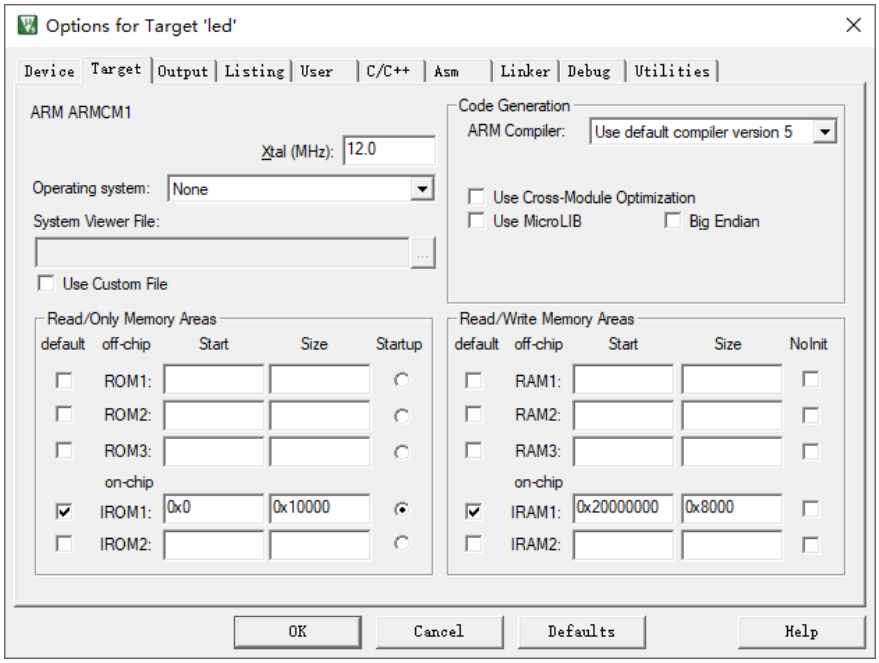
…\bootload\ref_design\FPGA_RefDesign\gowin_empu_m1

5 Embedded UserFlash Memory Download Method

5.1 Software Configuration

If you use ARM Keil MDK (tested software version V5.26), set the IROM1 start address to 0x0 and set the IROM1 Size to 0x10000 (64KB), as shown in Figure 5-1.

Figure 5-1 ROM Start Address and Size Configuration

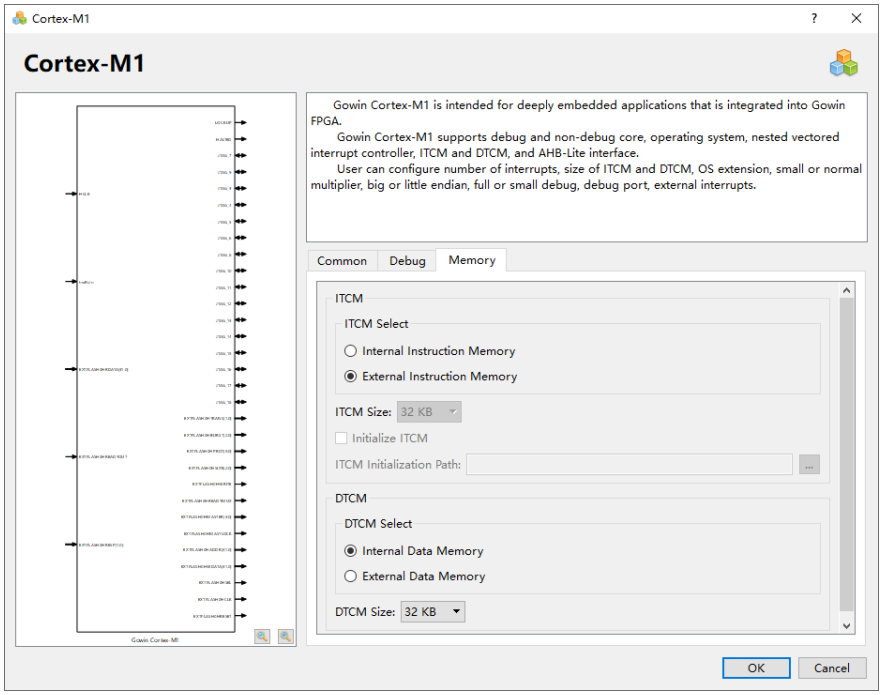


If you use GMD (tested software version V1.2) software development environment, choose the Flash linker GOWIN_M1_flash_xip.ld. Set the Flash start address "FLASH ORIGIN" to 0x00000000, and the Flash Size "LENGTH" to 64K.

5.2 Hardware Configuration

Use IP Core Generator tool of Gowin Software to configure and generate Gowin_EMPU_M1 hardware design. In this process, select External Instruction Memory as the instruction memory of Gowin_EMPU_M1 in ITCM Select options, as shown in Figure 5-2.

Figure 5-2 ITCM Select Option Configuration



5.3 Design Flow

1. In Gowin_EMPU_M1 hardware design, configure ITCM Select as External Instruction Memory.
2. IP Core Generator generates Gowin_EMPU_M1 hardware design.
3. Instantiate UserFlash Controller (GW1N/R-9 FLASH608K) with Memory Map as the instruction memory of Gowin_EMPU_M1.
4. Synthesize, place & route to generate bitstream files in hardware design.
5. Build and link to generate BIN files in software design.
6. Use Programmer to download the bitstream files in Gowin_EMPU_M1 hardware design and BIN files in Gowin_EMPU_M1 software design.

5.4 Download

For the usage of Gowin Programmer, please see *SUG502, Gowin Programmer User Guide*.

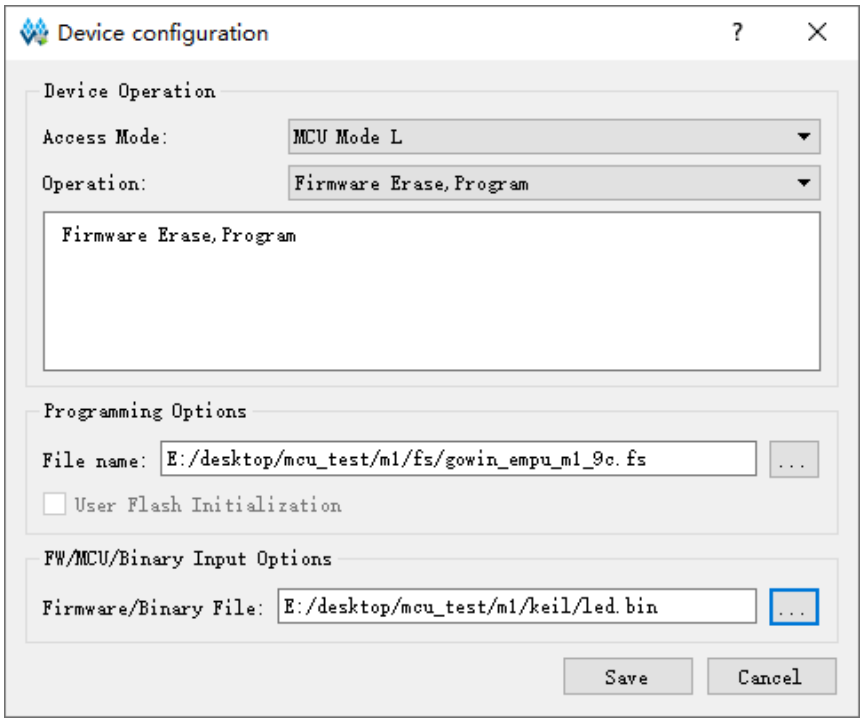
Select "Tools > Programmer" in the menu bar or "Programmer" ( ) in the tool bar in Gowin Software to open Programmer, the download tool.

Select "Edit > Configure Device" in Programmer menu bar or "Configure Device" ( ) in the tool bar to open the "Device configuration",

as shown in Figure 5-3.

- Select "MCU Mode L" in "Access Mode" drop-down list.
- Select "Firmware Erase, Program" or "Firmware Erase, Program, Verify" in "Operation" drop-down list.
- Import the hardware design bitstream file required to download in "Programming Options > File name" option.
- Select "FW/MCU/Binary Input Options > Firmware/Binary File" to import the BIN files in software programming design required to download.
- Click "Save" to complete the download configuration of the bitstream files in hardware design and BIN files in software design.

Figure 5-3 Device Configuration for GW1N/R-9



After device configuration, click "Program/Configure" ( ) in the Programmer tool bar to download the bitstream files in hardware design and the BIN files in software programming design at the same time.

5.5 Devices Supported

GW1N/R-9

5.6 Reference Design

Click this link to access the reference design and document:

...\solution\Embedded_Memory\ref_design\FPGA_RefDesign\gowin_empu_m1

...\solution\Embedded_Memory\ref_design\MCU_RefDesign\cm1_demo\project\led

...\solution\Embedded_Memory\ref_design\MCU_RefDesign\cm1_demo\project\printf

…\solution\Embedded_Memory\doc\ReadMe.txt


GOWIN
PROGRAMMING FOR THE FUTURE