

Gowin_EMPU_M1 Download **Reference Design**

IPUG532-2.4E, 01/17/2025

Copyright © 2025Guangdong Gowin Semiconductor Corporation. All Rights Reserved.

GOWIN is a trademark of Guangdong Gowin Semiconductor Corporation and is registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders. No part of this document may be reproduced or transmitted in any form or by any denotes, electronic, mechanical, photocopying, recording or otherwise, without the prior written consent of GOWINSEMI.

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. I²C 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

Contentsi
List of Figuresiii
List of Tablesiv
1 Download Methods1
2 Software Programming Output as ITCM Initialization Value
2.1 Tool
2.2 Command Parameter
2.3 Software Configuration
2.4 Hardware Configuration4
2.5 Design Flow
2.6 Devices Supported5
2.7 Reference Design
3 Merge Software Design and Hardware Design7
3.1 Tool
3.2 Command Parameters7
3.3 Hardware Configuration7
3.4 Design Flow
3.4.1 Merge
3.4.2 Download
3.5 Devices Supported9
3.6 Reference Design
4 Off-chip SPI-Flash Memory Download Method10
4.1 Software Configuration
4.2 Hardware Configuration 11
4.3 Design Flow
4.4 Download
4.4.1 Download Bitstream Files in Hardware Design13
4.4.2 Download BIN File in Software Programing Design15
4.5 Devices Supported17

4.6 Reference Design	. 17
5 Embedded UserFlash Memory Download Method	18
5.1 Software Configuration	. 18
5.2 Hardware Configuration	. 19
5.3 Design Flow	. 19
5.4 Download	. 19
5.5 Devices Supported	. 20
5.6 Reference Design	. 20

List of Figures

Figure 2-1 Configure External Tool	4
Figure 2-2 Configure ITCM Initialization	5
Figure 3-1 Post-Place File Configuration	8
Figure 3-2 Merge Software Programming Design and Hardware Design	8
Figure 4-1 ROM Start Address and Size Configuration	10
Figure 4-2 Configure ITCM	11
Figure 4-3 Device Configuration for GW2AN-9X/18X, GW2A/R/NR-18, and GW2A/N-55	14
Figure 4-4 Device Configuration for Arora V	15
Figure 4-5 Device Configuration for GW2AN-9X/18X, GW2A/R/NR-18, GW2A/N-55	16
Figure 4-6 Device Configuration for Arora V	17
Figure 5-1 ROM Start Address and Size Configuration	18
Figure 5-2 ITCM Select Option Configuration	19
Figure 5-3 Device Configuration for GW1N/R-9	20

List of Tables

Table 3-1 merge_bit Commands and Parameters 7	7
Table 4-1 Bootload 1	12

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: <u>cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip</u>

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.

Command Items	User Command		Stop on Exi	S
Before Compile C/C++ File				
🔽 Run #1		2	Not Specified	
Run #2		2	Not Specified	
Before Build/Rebuild				
🔽 Run #1		2	Not Specified	
🗌 🗌 Run #2		2	Not Specified	
After Build/Rebuild				
🔽 Run #1	D:\Keil_v5\ARM\ARMCC\bin\fromelf.exebin	2	Not Specified	
🔽 Run #2	make_hex.exe led.bin	2	Not Specified	
☑ Bun 'After-Build' Conditionally				

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
- $...\tool\mbox{make_hex\ref_design\MCU_RefDesign\cm1_demo\project\led}$
- ...\tool\make_hex\ref_design\MCU_RefDesign\cm1_demo\project\prin

tf

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: <u>cdn.gowinsemi.com.cn/Gowin_EMPU_M1.zip</u>

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.

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	

Table 3-1 Command Parameters

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.

🔆 Configuration		
	Place & Route	
✓ Global General	Category: All	Reset all to defau
Constraints	Label	Value
✓ Synthesize	Generate SDF File	False
General	Generate Post-Place File	True
✓ Place & Route	Generate Post-PnR Verilog Simulation Model File	False
General	Generate Post-PnR VHDL Simulation Model File	False
Voltage	Generate Plain Text Timing Report	False
Place	Promote Physical Constraint Warning to Error	True
Dual-Purpose Pin	Show All Warnings	False
Unused Pin	Report Auto-Placed IO Information	False
Incremental PnR	Convert SDP32/36 to SDP16/18	True
✓ BitStream General		
sysControl Feature sysControl	Generate post-place file only included locations o *.posp.	f BSRAM and DSP. Default:
Feature sysControl	Generate post-place file only included locations o *.posp.	f BSRAM and DSP. Default:

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.

Elemente	2 2 Norman	Cofference	Dreaman	min a Daci	and 1	Tandrurana	Dacian
rigure	5-2 Merge	Sonware	Program	ning Desi	gn ang i	пагимаге	Design
					n		

选择 C:\windows\system32\cmd.exe	-		×
C:\Users\liukai\Desktop\demo>call merge_bit.exe led.bin gowin_empu_ml.fs 32 gowin_em	pu_m1	.posp	^
Read original bsram initial value map			
Read location file gowin_empu_ml.posp Bsram R10[3] initial value convert to fusemap success.			
Bsram R10[2] initial value convert to fusemap success.			
Bsram R10[0] initial value convert to fusemap success.			
Bsram R10[10] initial value convert to fusemap success. Bsram R10[6] initial value convert to fusemap success.			
Bsram R10[5] initial value convert to fusemap success.			
Dsram RI0[4] initial value convert to fusemap success. Bsram R10[14] initial value convert to fusemap success.			
Bsram R10[13] initial value convert to fusemap success. Berem R10[12] initial value convert to fuseman success			
Bsram R10[12] initial value convert to fusemap success.			
Bsram K10[18] initial value convert to fusemap success. Bsram R10[17] initial value convert to fusemap success.			
Bsram R10[16] initial value convert to fusemap success.			
Replace new bsram initial value map to file new_gowin_empu_ml.fs			
Build bsram initial value replace completed.			- U

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 <u>SUG502, Gowin</u> <u>Programmer User Guide</u>.

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\prin

tf

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

🕅 Options for Target 'led'	×
Device Target Output Listing User C/C++	Asm Linker Debug Vtilities
ARM ARMCM1 <u>X</u> al (MHz):	Code Generation ARM Compiler: Use default compiler version 5 💌
Operating system: None	Use Cross-Module Optimization
System Viewer File:	Use MicroLIB Big Endian
Read/Only Memory Areas	Read/Write Memory Areas
default off-chip Start Size Startup	default off-chip Start Size NoInit
□ ROM1: □ 0	□ RAM1: □ □
□ ROM2: ○	□ RAM2: □
□ ROM3: □ 0	RAM3:
on-chip	on-chip
IROM1: 0x400 0x7C00 €	IRAM1: 0x20000000 0x8000 □
□ IROM2: □ 0	□ IRAM2: □ □
OK C	ancel Defaults Help

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

👶 Cortex-M1	? ×
Cortex-M1	
	Gowin Cortex-M1 is intended for deeply embedded applications that is integrated into Gowin FPGA. Gowin Cortex-M1 supports debug and non-debug core, operating system, nested vectored interrupt controller, ITCM and DTCM, and AHB-Lite interface. User can configure number of interrupts, size of ITCM and DTCM, OS extension, small or normal multiplier, big or little endian, full or small debug, debug port, external interrupts.
LOCKUP	Common Debug Memory
	ITCM Select Internal Instruction Memory
JTAG_7	O External Instruction Memory ITCM Size: 32 KB
JTAG_9	ITCM Initialization Path: \gowin_empu_m1\bootload\SA_0x100000\ITCM_Size_32KB
Gowin Cortex-M1	DTCM Select Internal Data Memory C External Data Memory
<u>.</u>	DTCM Size: 32 KB V
	OK Cancel

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

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

Table 4-1 Bootload

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 <u>SUG502, Gowin</u> <u>Programmer User Guide</u>.

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" (\downarrow) in the tool bar in Gowin Software to open Programmer, the download tool.

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

Device" (*P*) 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

🙀 Device configuration		?	×
Device Operation			
Access Mode:	External Flash Mode		-
Operation:	exFlash Erase, Program thru GAO-Bridge		-
exFlash Erase, Program	thru GAO-Bridge		
Frogramming Uptions			_
File name: E./desktop/	ncu_test/mi/is/gowin_empu_mi_556.is ation		
-External Flash Options			
Device:	Generic Flash		•
Start Address:	0x000000		
	Save	Cano	el

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.

 \times

 \sim \sim

 \sim

Cancel

gure 4-4 Devic	e Configuration for Arora V	7
🍦 Device configurat	ion	?
Device Operation —		
Access Mode:	External Flash Mode 5A	
Operation:	exFlash Erase, Program 5A	
Programming Option	15	
File name: kai/Des	ktop/gowin_empu_m1/cm1_demo/impl/pn	r/cm1_demo.fs
Retained User F.	lash Data	

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

Generic Flash

Save

0x000000

4.4.2 Download BIN File in Software Programing Design

External Flash Options

Device:

Start Address:

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" (\bigcirc) 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

🙀 Device configuration	?	×
Device Operation		
Access Mode:	External Flash Mode	•
Operation:	exFlash C Bin Erase, Program thru GAO-Bridge	•
exFlash C Bin Erase, P	rogram thru GAO-Bridge	
- External Flash Options		
Device:	Generic Flash	•
Start Address:	0x400000	
-FW/MCU/Binary Input Opt Firmware/Binary File:	tions s/liukai/Desktop/gw5ast/led1/PROJECT/led.bin Save Cance	

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.

Device configurat	ion	?	×	
Device Operation —				
Access Mode:	External Flash Mode 5A		\sim	
Dperation:	exFlash C Bin Erase, Program 5A	exFlash C Bin Erase, Program 5A 🗸 🗸		
exFlash C Bin Eras	se, Program 5A		1	
External Flash Opt	ions Generic Flash		~	
External Flash Opt Device:	ions Generic Flash		~	
External Flash Opt Device: Start Address:	ions Generic Flash Ox100000		~	
External Flash Opt Device: Start Address: FW/MCU/Binary Inpu	ions Generic Flash Ox100000 t Options		~	
External Flash Opt Device: Start Address: FW/MCU/Binary Inpu Firmware/Binary Fi	ions Generic Flash Ox100000 t Options le: E:/desktop/mcu_test/m1/keil/led.bin		~ 	

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 <u>link</u> 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

🕅 Options for Target 'led'	×					
Device Target Output Listing Vser C/C++ Asm Linker Debug Vilities						
ARM ARMCM1	Code Generation ARM Compiler: Use default compiler version 5 💌					
<u>Xtal (MHz):</u> 12.0						
Operating system: None	Use Cross-Module Optimization					
System Viewer File:	Use MicroLIB Big Endian					
Use Custom File						
Read/Only Memory Areas	Read/Write Memory Areas					
default off-chip Start Size Sta	artup default off-chip Start Size Nolnit					
□ ROM1: 0	C □ RAM1: □ □ □					
□ ROM2: □ 0	C 🗆 RAM2:					
ROM3:	C 🗆 RAM3:					
on-chip	on-chip					
IROM1: 0x0 0x10000 0						
IROM2:	C □ IRAM2: □ □					
OK Cancel Defaults Help						

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.



Cortex-M1		e e e e e e e e e e e e e e e e e e e
	 •• 1000 	Gowin Cortex-M1 is intended for deeply embedded applications that is integrated into Gowir FPGA. Gowin Cortex-M1 supports debug and non-debug core, operating system, nested vectored interrupt controller, ITCM and DTCM, and AHB-Lite interface. User can configure number of interrupts, size of ITCM and DTCM, OS extension, small or nor multiplier, big or little endian, full or small debug, debug port, external interrupts.
	2200, A	Common Debug Memory
	••• 3.00. •• 5.00. <td>ITCM Select O Internal Instruction Memory External Instruction Memory ITCM Size: 32 KB 7 Initialize ITCM ITCM Initialization Path:</td>	ITCM Select O Internal Instruction Memory External Instruction Memory ITCM Size: 32 KB 7 Initialize ITCM ITCM Initialization Path:
• EFG.AHIMAN723		DTCM DTCM Select © Internal Data Memory O External Data Memory
		DTCM Size: 32 KB •

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 <u>SUG502, Gowin</u> <u>Programmer User Guide</u>.

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

Select "Edit > Configure Device" in Programmer menu bar or "Configure Device" (\bigcirc) 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.

🙀 Device configuration		?	\times
Device Operation			
Access Mode:	MCV Mode L		•
Operation:	Firmware Erase, Program		•
Firmware Erase, Progra	un		
Programming Options File name: E:/desktop/ User Flash Initializ	mcu_test/m1/fs/gowin_empu_m1_9c.fs		
FW/MCU/Binary Input Op Firmware/Binary File: [tions E:/desktop/mou_test/m1/keil/led.bin		

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:

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

 $...\solution\Embedded_Memory\ref_design\MCU_RefDesign\cm1_demovproject\led$

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

 $... \verb|solution|Embedded_Memory|doc|ReadMe.txt|$

