

# Gowin SecureFPGA IP

# 用户指南

UG936-1.1,2021-08-30

#### 版权所有 © 2021 广东高云半导体科技股份有限公司

GO<sup>IN</sup>INT, INFT, INFT,

#### 免责声明

本文档并未授予任何知识产权的许可,并未以明示或暗示,或以禁止发言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外,高云半导体 概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和/或使用不作任何明示或暗示的担保,包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知 识产权的侵权责任等,均不作担保。高云半导体对文档中包含的文字、图片及其它内容的准确性和完整性不承担任何法律或非法律责任,高云半导体保留修改文档中任何内容的权利, 恕不另行通知。高云半导体不承诺对这些文档进行适时的更新。

版本信息

日期	版本	说明	
2020/05/15	1.0	初始版本。	
2021/08/30	1 1	● 支持安全芯片 GW1NSER-4C (Gowin_EMPU(GW1NS-4C));	
2021/00/30	1.1	● 支持安全芯片 GW1NE-9C (Gowin_EMPU_M1)。	

# 目录

E	录	. i
2	]目录	iii
耒	是目录	iv
1	参考设计	. 1
	1.1 硬件参考设计	. 1
	1.1.1 IID 硬件参考设计	. 1
	1.1.2 SHA3 硬件参考设计	. 1
	1.2 软件参考设计	. 2
	1.2.1 IID 软件参考设计	. 2
	1.2.2 SHA3 软件参考设计	. 2
2	硬件参考设计	.3
	2.1 硬件环境	. 3
	2.2 软件环境	. 3
	2.3 MCU 配置	. 3
	2.3.1 GW1NSE-2C(Gowin_EMPU(GW1NS-2C))	. 3
	2.3.2 GW1NSER-4C(Gowin_EMPU(GW1NS-4C))	. 4
	2.3.3 GW1NE-9C(Gowin_EMPU_M1)	. 5
	2.4 硬件属性配置	. 6
	2.4.1 Synthesize 属性配置	. 6
	2.4.2 Place&Route 属性配置	. 6
	2.4.3 BitStream 属性配置	. 8
	2.5 综合	. 8
	2.6 布局布线	. 9
	2.7 下载	. 9
	2.7.1 GW1NSE-2C(Gowin_EMPU(GW1NS-2C))	10
	2.7.2 GW1NSER-4C(Gowin_EMPU(GW1NS-4C))	10
	2.7.3 GW1NE-9C	11

i

	2.8 参考手册	. 12
3	软件参考设计	13
	3.1 软件环境	. 13
	3.2 软件属性配置	. 13
	3.2.1 Target Processor 配置	. 14
	3.2.2 GNU ARM Cross Assembler 配置	. 14
	3.2.3 GNU ARM Cross C Compiler 配置	. 15
	3.2.4 GNU ARM Cross C Linker 配置	. 16
	3.2.5 GNU ARM Cross Create Flash Image 配置	. 18
	3.3 编译	. 19
	3.4 下载	. 19
	3.4.1 GW1NSE-2C(Gowin_EMPU(GW1NS-2C))	. 20
	3.4.2 GW1NSER-4C(Gowin_EMPU(GW1NS-4C))	. 21
	3.4.3 GW1NE-9C	. 22
	3.5 参考手册	.23

## 图目录

图 2-1 Synthesize 属性配置	. 6
图 2-2 General 配置	. 7
图 2-3 Dual-Purpose Pin 配置	. 7
图 2-4 BitStream 属性配置	. 8
图 2-5 综合	. 9
图 2-6 布局布线	. 9
图 2-7 下载属性配置	. 10
图 2-8 下载属性配置	. 11
图 2-9 下载属性配置	. 12
图 3-1 软件属性配置	. 13
图 3-2 Target Processor 配置	. 14
图 3-3 GNU ARM Cross Assembler 配置	. 15
图 3-4 GNU ARM Cross C Compiler 配置	. 16
图 3-5 GNU ARM Cross C Linker > Script files 配置	. 17
图 3-6 GNU ARM Cross C Link > Libraries 配置	. 18
图 3-7 GNU ARM Cross Create Flash Image 配置	. 18
图 3-8 编译	. 19
图 3-9 Programmer 打开方式	. 20
图 3-10 下载属性配置	. 21
图 3-11 下载属性配置	. 22
图 3-12 下载属性配置	. 23

# 表目录

表 2-1 Gowin_EMPU(GW1NS-2C) MCU 配置	. 3
表 2-2 Gowin_EMPU(GW1NS-2C) BSRAM 位置约束	. 4
表 2-3 Gowin_EMPU(GW1NS-4C) MCU 配置	. 4
表 2-4 Gowin_EMPU(GW1NS-4C) BSRAM 位置约束	. 4
表 2-5 Gowin_EMPU_M1 MCU 配置	. 5
表 2-6 Gowin_EMPU_M1 BSRAM 位置约束	. 5



## 1.1 硬件参考设计

#### 1.1.1 IID 硬件参考设计

Gowin SecureFPGA产品GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和GW1NE-9C (Gowin\_EMPU\_M1),提供<u>IID硬件参考设计</u>,通过链接获取如下参考设 计:

- GW1NSE-2C
   Gowin\_SecureFPGA\GW1NSE\_2C\IID\ref\_design\FPGA\_RefDesign
- GW1NSER-4C
   Gowin\_SecureFPGA\GW1NSER\_4C\IID\ref\_design\FPGA\_RefDesig
   n
- GW1NE-9C
   Gowin\_SecureFPGA\GW1NE\_9C\IID\ref\_design\FPGA\_RefDesign

#### 1.1.2 SHA3 硬件参考设计

Gowin SecureFPGA 产品 GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和 GW1NE-9C (Gowin\_EMPU\_M1),提供 <u>SHA3 硬件参考设计</u>,通过链接获取如下参考 设计:

- GW1NSE-2C
   Gowin\_SecureFPGA\GW1NSE\_2C\SHA3\ref\_design\FPGA\_RefDesi gn
- GW1NSER-4C Gowin\_SecureFPGA\GW1NSER\_4C\SHA3\ref\_design\FPGA\_RefDe sign
- GW1NE-9C

Gowin\_SecureFPGA\GW1NE\_9C\SHA3\ref\_design\FPGA\_RefDesign

## 1.2 软件参考设计

## 1.2.1 IID 软件参考设计

Gowin SecureFPGA产品GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和GW1NE-9C (Gowin\_EMPU\_M1),提供GOWIN MCU Designer(V1.1)软件环境的<u>IID</u> 软件编程参考设计,通过链接获取如下参考设计:

- GW1NSE-2C
   Gowin\_SecureFPGA\GW1NSE\_2C\IID\ref\_design\MCU\_RefDesign
- GW1NSER\_4C
   Gowin\_SecureFPGA\GW1NSER\_4C\IID\ref\_design\MCU\_RefDesign
- GW1NE-9C Gowin\_SecureFPGA\GW1NE\_9C\IID\ref\_design\MCU\_RefDesign

### 1.2.2 SHA3 软件参考设计

Gowin SecureFPGA 产品 GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和 GW1NE-9C (Gowin\_EMPU\_M1),提供 GOWIN MCU Designer(V1.1)软件环境的 SHA3 软件编程参考设计,通过链接获取如下参考设计:

- GW1NSE-2C
   Gowin\_SecureFPGA\GW1NSE\_2C\SHA3\ref\_design\MCU\_RefDesig
   n
- GW1NSER-4C Gowin\_SecureFPGA\GW1NSER\_4C\SHA3\ref\_design\MCU\_RefDesi gn
- GW1NE-9C
   Gowin\_SecureFPGA\GW1NE\_9C\SHA3\ref\_design\MCU\_RefDesign

# 2硬件参考设计

## 2.1 硬件环境

- DK-START-GW1NSE2C V3.1
   GW1NSE-UX2CLQ144C5/I4
- DK-START-GW1NSER4C-QN48G V1.1 GW1NSER-LV4CQN48GC7/I6
- DK-START-GW1N9 V1.1
   GW1N-LV9EQ144C6/I5

## 2.2 软件环境

Gowin\_V1.9.8Beta

## 2.3 MCU 配置

## 2.3.1 GW1NSE-2C (Gowin\_EMPU(GW1NS-2C))

以软件开发工具包 DK\_START\_GW1NSE2\_V3.1 参考设计为例。

Gowin\_EMPU(GW1NS-2C) MCU 配置,如表 2-1 所示。

#### 表 2-1 Gowin\_EMPU(GW1NS-2C) MCU 配置

配置选项	配置参数
Frequency	25MHz
Sram Size	8KB
Flash Size	128KB
UART0	Enable
Timer0	Enable
Timer1	Enable
WatchDog	Enable

SRAM Size 配置为 8KB,使用 4 个 BSRAM,每个 BSRAM 的位置约 束,如表 2-2 所示。

表 2-2 Gowin\_EMPU(GW1NS-2C) BSRAM 位置约束

BSRAM	约束位置
bram_sp_0	BSRAM_R6[0]
bram_sp_1	BSRAM_R6[1]
bram_sp_2	BSRAM_R6[2]
bram_sp_3	BSRAM_R6[3]

#### 2.3.2 GW1NSER-4C (Gowin\_EMPU(GW1NS-4C))

以软件开发工具包 DK\_START\_GW1NSER4C\_QN48G\_V1.1 参考设计 为例。

Gowin\_EMPU(GW1NS-4C) MCU 配置,如表 2-3 所示。

#### 表 2-3 Gowin\_EMPU(GW1NS-4C) MCU 配置

配置选项	配置参数
Frequency	50MHz
Sram Size	8КВ
Flash Size	32KB
UART0	Enable
Timer0	Enable
Timer1	Enable
WatchDog	Enable

SRAM Size 配置为 8KB,使用 4 个 BSRAM,每个 BSRAM 的位置约 束,如表 2-4 所示。

表 2-4 Gowin\_EMPU(GW1NS-4C) BSRAM 位置约束

BSRAM	约束位置
bram_sp_0	BSRAM_R10[0]
bram_sp_1	BSRAM_R10[1]
bram_sp_2	BSRAM_R10[2]
bram_sp_3	BSRAM_R10[3]

## 2.3.3 GW1NE-9C (Gowin\_EMPU\_M1)

以软件开发工具包 DK\_START\_GW1N9\_V1.1 参考设计为例。

Gowin\_EMPU\_M1 MCU 配置,如表 2-5 所示。

#### 表 2-5 Gowin\_EMPU\_M1 MCU 配置

配置选项	配置参数
Frequency	20MHz
Debug	Enable
Debug Interface	JTAG and SW
OS Extension	Enable
ITCM	Embedded UserFlash 64KB
DTCM	32KB
UART0	Enable

**DTCM** 配置为 32KB,使用 16 个 BSRAM,每个 BSRAM 的位置约束, 如表 2-6 所示。

#### 表 2-6 Gowin\_EMPU\_M1 BSRAM 位置约束

BSRAM	约束位置
mem0_mem0_0_0_s	BSRAM_R28[14]
mem0_mem0_0_1_s	BSRAM_R10[4]
mem0_mem0_0_2_s	BSRAM_R28[7]
mem0_mem0_0_3_s	BSRAM_R10[3]
mem1_mem0_0_0_s	BSRAM_R28[9]
mem1_mem0_0_1_s	BSRAM_R28[13]
mem1_mem0_0_2_s	BSRAM_R10[5]
mem1_mem0_0_3_s	BSRAM_R28[8]
mem2_mem0_0_0_s	BSRAM_R28[11]
mem2_mem0_0_1_s	BSRAM_R28[10]
mem2_mem0_0_2_s	BSRAM_R28[12]
mem2_mem0_0_3_s	BSRAM_R10[6]
mem3_mem0_0_0_s	BSRAM_R10[8]
mem3_mem0_0_1_s	BSRAM_R10[7]
mem3_mem0_0_2_s	BSRAM_R10[9]
mem3_mem0_0_3_s	BSRAM_R10[10]

## 2.4 硬件属性配置

#### 2.4.1 Synthesize 属性配置

Synthesize 属性配置,例如 GW1NSER-4C,如图 2-1 所示。

- Synthesis Tool: GowinSynthesis<sup>™</sup>
- Top Module/Entity: 请根据硬件设计中的实际顶层模块名称配置
- Include Path: 请根据硬件设计中的实际文件路径配置
- Verilog Language: 例如 System Verilog 2017
   Synthesize 属性配置,请根据硬件设计实际需求配置。

#### 图 2-1 Synthesize 属性配置

	Synthesize
General Synthesize General Place & Route General Unused Pin Dual-Purpose Pin BitStream	General Synthesis Tool:  GowinSynthesis Top Module/Entity: Gowin_EMPU_template Include Path: GowinSynthesis Verilog Language: System Verilog 2017  VHDL Language: VHDL 2008 Looplimit: 2000 Disable Insert Pad Ram R/W Check DSP Balance Show All Warnings
	OK Cancel Apply

### 2.4.2 Place&Route 属性配置

Place&Route 属性配置,包括 General 配置、Unused Pin 配置和 Dual-Purpose Pin 配置。

#### General 配置

General 配置,例如 GW1NSER-4C,如图 2-2 所示。 General 配置,请根据硬件设计实际需求配置。 图 2-2 General 配置

	Place & Route		
General Y Synthesize	Category: All	Reset all t	to defaul
General	Label	Value	^
✓ Place & Route	Generate SDF File	False	
General	Generate Constraint File of Ports	False	
Unused Pin	Generate IBIS File	False	
Dual-Purpose Pin	Generate Post-Place File	True	
BitStream	Generate Post-PNR Simulation Model File	False	
	Initialize Primitives	False	
	Show All Warnings	False	
	Generate Plain Text Timing Report	False	
	Run Timing Driven	True	
	Use SCF	False	
	Promote Physical Constraint Warning to Error	False	
	Report Auto-Risced IO Information	Falce	
	Generate Post-Place File. Default: *.posp		

#### Dual-Purpose Pin 配置

Dual-Purpose Pin 配置,例如 GW1NSER-4C,如图 2-3 所示。

Dual-Purpose Pin 配置,请根据硬件设计实际约束 IO 位置以及所用开 发板 IO 设计配置。

#### 图 2-3 Dual-Purpose Pin 配置

🐳 Configuration	×
	Dual-Purpose Pin
General Synthesize General Place & Route General Unused Pin Dual-Purpose Pin BitStream	<ul> <li>Use JTAG as regular IO</li> <li>Use SSPI as regular IO</li> <li>Use MSPI as regular IO</li> <li>Use READY as regular IO</li> <li>Use DONE as regular IO</li> <li>Use RECONFIG_N as regular IO</li> <li>Use MODE as regular IO</li> <li>Use I2C as regular IO</li> </ul>
	OK Cancel Apply

## 2.4.3 BitStream 属性配置

BitStream 属性配置,例如 GW1NSER-4C,如图 2-4 所示。

- Print BSRAM Initial Value: 请不要勾选
- Secure Mode(device can be programmed only once): 如果用户处于安全芯片研发调试阶段,请不要勾选,否则安全芯片仅能 下载一次,不能在继续下载使用。 如果用户处于安全芯片量产出厂阶段,请勾选,实现安全芯片作用。

BitStream 其他属性配置,请根据硬件设计实际需求配置。

#### 图 2-4 BitStream 属性配置

W Configuration	>
	BitStream
General Synthesize General Place & Route General Unused Pin Dual-Purpose Pin BitStream	<ul> <li>☑ Enable CRC Check</li> <li>□ Enable Compress</li> <li>□ Enable Encryption (only support GW2A) Key(Hex): 0000000-0000000-00000000</li> <li>☑ Enable Security Bit</li> <li>□ Print BSRAM Initial Value</li> <li>Background Programming: OFF ▼</li> <li>□ Secure Mode(device can be programmed only once)</li> <li>□ Program Done Bypass</li> <li>□ Power On Reset</li> <li>Wake Up Mode: 0</li> <li>▼</li> <li>Loading Rate (MH2): 2.500 (default)</li> <li>▼</li> <li>SPI Flash Address: 0000000</li> <li>USERCODE: ● Default ○ Custom 0000000</li> <li>Bitstream Format: ● Text ○ Binary</li> </ul>
	OK Cancel Apply

## 2.5 综合

使用云源软件的综合工具 GowinSynthesis,综合硬件设计,产生综合 后的网表文件,例如 GW1NSER-4C,如图 2-5 所示。

<complex-block>

## 2.6 布局布线

完成综合后,使用云源软件的布局布线工具 Place & Route,布局布线,产生硬件设计码流文件,如图 2-6 所示。

图 2-6 布局布线



## 2.7 下载

完成布局布线,产生硬件设计码流文件后,使用云源软件的下载工具 Programmer,下载硬件设计码流文件。

在云源软件中或安装路径下,打开下载工具 Programmer。

单击 Programmer 菜单栏 Edit > Configure Device 或工具栏 Configure Device "☞", 打开 Device configuration。

## 2.7.1 GW1NSE-2C (Gowin\_EMPU(GW1NS-2C))

如果是安全芯片 GW1NSE-2C, 下载属性配置, 如图 2-7 所示。

Access Mode 下拉列表,选择"Embedded Flash Mode"选项。

Operation 下拉列表,选择 "embFlash Erase, Program"选项或 "embFlash Erase, Program, Verify"选项。

"Programming Options > File name"选项,导入硬件设计码流文件。
单击 Save,完成下载属性配置。

#### 图 2-7 下载属性配置

🙀 Device configuratio	n		?	×
Device Operation				
Access Mode:	Embedded Flash	Mode		•
Operation:	embFlash Erase,	Program		•
Erase and program the embedded flash.Make sure the config frequency must be less than 25Mhz.				
Programming Options				
File name: START_GW1	NSE2C_V3.1/secure_:	fpga/impl/pnr/secur	e_fpga.fs	
User Flash Initia	lization			

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NSE-2C 硬件设计码流文件下载。

## 2.7.2 GW1NSER-4C (Gowin\_EMPU(GW1NS-4C))

如果是安全芯片 GW1NSER-4C,下载属性配置,如图 2-8 所示。

Access Mode 下拉列表,选择"SecureFPGA Mode"选项。

Operation 下拉列表,选择"Firmware Erase, Program Securely"选项。

"Programming Options > File name"选项,导入硬件设计码流文件。

"FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入 软件编程二进制 BIN 文件。

单击 Save,完成下载属性配置。

#### 图 2-8 下载属性配置

🙀 Device configuration		?	$\times$		
Device Operation					
Access Mode:	SecureFPGA Mode		•		
Operation:	Firmware Erase, Program Securely		•		
Firmware Erase, Program, Verify with Security(One Time Programmable For Key Authentication Code)					
Programming Options					
Vser Flash Initialization					
-FW/MCU/Binary Input Opt	tions				
Firmware/Binary File:	ER_4C/Debug/bk_with_no_enroll_GW1NSER_4C.	bin .			
	Save	Cance	1		

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NSER-4C 硬件设计码流文件和 软件编程二进制 BIN 文件下载。

#### 2.7.3 GW1NE-9C

如果是安全芯片 GW1NE-9C,下载属性配置,如图 2-9 所示。

Access Mode 下拉列表,选择"SecureFPGA Mode L"选项。

Operation 下拉列表,选择"Firmware Erase, Program Securely"选项。

"Programming Options > File name"选项,导入硬件设计码流文件。

"FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入 软件编程二进制 BIN 文件。

单击 Save,完成下载属性配置。

#### 图 2-9 下载属性配置

🙀 Device configuration		?	$\times$	
Device Operation				
Access Mode:	SecureFPGA Mode L		•	
Operation:	Firmware Erase, Program Securely		•	
Firmware Erase, Program, Verify with Security(One Time Programmable For Key Authentication Code)				
Programming Options File name: C:/Users/li	ukai/Desktop/secure_fpga.fs			
User Flash Initializ	ation			
-FW/MCU/Binary Input Opt	tions			
Firmware/Binary File:	ers/liukai/Desktop/GW1NS-9C_bk_no_enroll.	bin .		
	Save	Cance	1	

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NE-9C 硬件设计码流文件和软件 编程二进制 BIN 文件下载。

## 2.8 参考手册

Gowin SecureFPGA产品GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和GW1NE-9C (Gowin EMPU M1)硬件设计,请参考:

- IPUG517, Gowin\_EMPU(GW1NS-2C)硬件设计参考手册
- IPUG932, Gowin\_EMPU(GW1NS-4C)硬件设计参考手册
- <u>IPUG531</u>, Gowin\_EMPU\_M1 硬件设计参考手册
- <u>SUG100</u>, Gowin 云源软件用户指南
- <u>SUG101</u>, Gowin 设计约束指南
- <u>SUG502</u>, Gowin Programmer 用户指南

3软件参考设计

## 3.1 软件环境

GMD V1.1.

## 3.2 软件属性配置

Project Explorer 视图中,选择当前工程,右键选择"Properties > C/C++ Build > Settings"选项,配置当前工程的参数属性,如图 3-1 所示。

filter text S	ettings					
Resource C/C++ Build Build Variables Environment	Configuration:	Debug [Active]			~	Manage Configurations
Logging Settings	🛞 Tool Setting	s 🛞 Toolchains	Devices	🎤 Build Steps 🏻 🖳	Build Artifact	Binary Parsers
Tool Chain Editor	🖄 Target	Processor		ARM family	cortex-m3	~
C/C++ General MCU	🖉 Optimi	zation		Architecture	Toolchain det	ault ~
Project References	🖄 Warnin	igs aina		Instruction set	Thumb (-mth	umb) v
Run/Debug Settings	V 🛞 GNU A	RM Cross Assemble	er	Thumb intervor	k (-mthumb-int	tenwork)
	🖄 Pre	processor		Endianness	Toolchain det	iault v
	🖄 Wai	uaes rnings		Float ABI	Toolchain det	fault v
	🖄 Mis	cellaneous		FPU Type	Toolchain det	fault v
	V 🛞 GNU A	RM Cross C Compil processor	er	Unaligned access	Toolchain det	fault v
	Incl Opt	udes imization		AArch64 family	Generic (-mc	ou=generic) ~
	🖉 Wa	rnings		Feature crc	Toolchain det	ault ~
	🖉 Mis	cellaneous RM Cross C Linker		Feature crypto	Toolchain def	ault 🗸
	🖉 Ger	ieral		Feature fp	Toolchain det	ault ~
	🖄 Libr	aries		Feature simd	Enabled (+sir	nd) 🗸 🗸
	V 🗞 GNU A	RM Cross Create Fl	ash Image	Code model	Small (-mcmc	del=small) v
	🖄 Ger	eral		Strict align (-mst	trict-align)	
	GRU A	RM Cross Print Size		Other target flags	_	
	l Ber	iciai				

#### 图 3-1 软件属性配置

## 3.2.1 Target Processor 配置

配置"Target Processor > ARM family"选项。

- 如果是安全芯片 GW1NSE-2C 和 GW1NSER-4C,该选项配置为 "cortex-m3"
- 如果是安全芯片 GW1NE-9C,该选项配置为 "cortex-m1"
   例如 GW1NSER-4C,如图 3-2 所示。

#### 图 3-2 Target Processor 配置

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🏻 🖳	Build Artifact 🗟 Binary Parsers	• •
🖄 Target Processor	ARM family	cortex-m3	$\sim$
🖄 Optimization	Architecture	Toolchain default	$\sim$
🖉 Debugging	Instruction set	Thumb (-mthumb)	$\sim$
SOLUTION OF STREET	Thumb interworl	k (-mthumb-interwork)	
Preprocessor Preprocessor	Endianness	Toolchain default	$\sim$
Warnings	Float ABI	Toolchain default	$\sim$
Miscellaneous     SOLU ARM Cross C Compiler	FPU Type	Toolchain default	$\sim$
Preprocessor	Unaligned access	Toolchain default	~
Includes Optimization	AArch64 family	Generic (-mcpu=generic)	~
🖉 Warnings	Feature crc	Toolchain default	$\sim$
➢ Miscellaneous ✓ ℕ GNU ARM Cross C Linker	Feature crypto	Toolchain default	~
🖉 General	Feature fp	Toolchain default	~
🖄 Libraries 🖄 Miscellaneous	Feature simd	Enabled (+simd)	~
✓  Solve GNU ARM Cross Create Flash Image	Code model	Small (-mcmodel=small)	~
Beneral	Strict align (-mst	trict-align)	
internal internal	Other target flags		

## 3.2.2 GNU ARM Cross Assembler 配置

配置"GNU ARM Cross Assembler > Preprocessor > Defined symbols (-D)"选项,该选项配置为"\_\_STARTUP\_CLEAR\_BSS",如图 3-3 所示。

Tool Settings	🛞 Toolchains	Devices	🎤 Build Steps	🚇 Build Artifact	🗟 Binary Parsers	14
🖄 Target Pr	ocessor		Use preproce	essor		
🖄 Optimizat	ion		Do not searc	h system directorie	es (-nostdinc)	
🖉 Warnings			Preprocess o	only (-E)		
🖄 Debuggin	g		Defined symb	ols (-D)		1.0
🗸 🛞 GNU ARN	1 Cross Assemble	er	Denned Symp	010 ( 0)		- M
🖄 Prepro	ocessor		STARTUP_CLE	EAR_BSS		
🖄 Includ	es					
🖄 Warnii	ngs					
🖄 Miscel	laneous					
🗸 🛞 GNU ARN	1 Cross C Compil	er				
🖄 Prepro	ocessor					
🖄 Includ	es					
🖄 Optim	ization					
🖄 Warnii	ngs					
🖄 Miscel	laneous					
🗸 🛞 GNU ARN	1 Cross C Linker					
🖄 Gener	al		Undefined com	ub ala (11)	<b>.</b>	
🖄 Librari	es		Underined syn	nbois (-0)	🔁 🛍 🗃 🕅	al 2
🖄 Miscel	laneous					
V 🛞 GNU ARN	1 Cross Create Fl	ash Image				
🖉 Gener	al 					
✓	1 Cross Print Size					
🖄 Gener	al					

图 3-3 GNU ARM Cross Assembler 配置

#### 3.2.3 GNU ARM Cross C Compiler 配置

配置 "GNU ARM Cross C Compiler > Includes > Include paths (-I)"选项,该选项配置为当前工程的 C 头文件引用路径,如图 3-4 所示。

例如 GW1NSER-4C, C 头文件路径配置,如下所示。

"\${workspace\_loc:/\${ProjName}/BroadKey/include}"

"\${workspace\_loc:/\${ProjName}/BroadKey/src}"

"\${workspace\_loc:/\${ProjName}/CORE}"

"\${workspace\_loc:/\${ProjName}/PERIPHERAL/Includes}"

"\${workspace\_loc:/\${ProjName}/SYSTEM}"

"\${workspace\_loc:/\${ProjName}/USER}"

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 🚇 Build Artifact 🗟 Binary Parsers 💽
<ul> <li>Target Processor</li> <li>Optimization</li> <li>Warnings</li> <li>Debugging</li> <li>SOLU ARM Cross Assembler</li> <li>Preprocessor</li> <li>Includes</li> <li>Warnings</li> </ul>	Include paths (-1) *{workspace_loc:/{ProjName}/BroadKey/src}* *{workspace_loc:/{ProjName}/CORE}* *{workspace_loc:/{ProjName}/CORE}* *{workspace_loc:/{ProjName}/PERIPHERAL/Includes}* *{workspace_loc:/{ProjName}/SYSTEM}* *{workspace_loc:/{ProjName}/USER}*
<ul> <li>Miscellaneous</li> <li>GNU ARM Cross C Compiler</li> <li>Preprocessor</li> <li>Includes</li> <li>Optimization</li> <li>Warnings</li> <li>Miscellaneous</li> <li>GNU ARM Cross C Linker</li> </ul>	Include system paths (-isystem) 🕢 🗿 🖗 🏠
<ul> <li>General</li> <li>Libraries</li> <li>Miscellaneous</li> <li>GNU ARM Cross Create Flash Image</li> <li>General</li> <li>GNU ARM Cross Print Size</li> <li>General</li> </ul>	Include files (-include) 🕢 🕢 🕅

图 3-4 GNU ARM Cross C Compiler 配置

## 3.2.4 GNU ARM Cross C Linker 配置

#### Script files 配置

配置 "GNU ARM Cross C Linker > General > Script files (-T)" 选项。

- 如果是安全芯片 GW1NSE-2C,该选项配置为
   Gowin\_EMPU(GW1NS-2C) GMD Flash 链接器 "gw1ns2c\_flash.ld"
- "\${workspace\_loc:/\${ProjName}/gw1ns2c\_flash.ld}"
- 如果是安全芯片 GW1NSER-4C,该选项配置为
   Gowin\_EMPU(GW1NS-4C) GMD Flash 链接器 "gw1ns4c\_flash.ld"
- "\${workspace\_loc:/\${ProjName}/gw1ns4c\_flash.ld}"
- 如果是安全芯片 GW1NE-9C,该选项配置为 Gowin\_EMPU\_M1 GMD Flash 链接器 "GOWIN\_M1\_flash.ld"

"\${workspace\_loc:/\${ProjName}/GOWIN\_M1\_flash.ld}"

例如,GW1NSER-4C GMD Flash 链接器配置,如图 3-5 所示。

Tool Settings Toolchains Devices	🎤 Build Steps  🚇 Build Artifact	Binary Parsers 🔹
🖄 Target Processor	Script files (-T)	🛃 🔊 🛜 🏹 灯
🖄 Optimization	\${workspace_loc:/\${ProiName}/S	cript/gw1ns4c_flash.ld}"
🖄 Warnings		on p ( g i no re_iconici)
🖄 Debugging		
🗸 🛞 GNU ARM Cross Assembler		
🖄 Preprocessor		
🖄 Includes		
🖄 Warnings		
🖄 Miscellaneous		
🗸 🛞 GNU ARM Cross C Compiler		
🖄 Preprocessor		
🖄 Includes		
🖄 Optimization		
🖄 Warnings		
🖄 Miscellaneous		
🗸 🛞 GNU ARM Cross C Linker		
🖄 General		
🖄 Libraries		
🖄 Miscellaneous		
🗸 🛞 GNU ARM Cross Create Flash Image		
🖄 General		
🗸 🛞 GNU ARM Cross Print Size		
🖄 General		

图 3-5 GNU ARM Cross C Linker > Script files 配置

#### Libraries 配置

配置"GNU ARM Cross C Linker > Libraries > Libraries (-I)"和"Library search path (-L)"选项。

● 如果是安全芯片 GW1NSE-2C 和 GW1NSER-4C, Libraries 配置如下所示。

```
Libraries (-I):
lidbroadkey_pro_256_noenroll_m3
Library search path (-L):
```

"{workspace\_loc:/\${ProjName}/BroadKey/lib}"

● 如果是安全芯片 GW1NE-9C, Libraries 配置如下所示。 Libraries (-I):

iid\_bk\_pro\_256\_no\_enroll\_m1

Library search path (-L):

"{workspace\_loc:/\${ProjName}/BK/lib}"

例如,GW1NSER-4C,Libraries 配置,如图 3-6 所示。

🛞 Tool Settings 🛞 Toolchains 📕 Devices	🎤 Build Steps 骨 Build Artifact 🗟 Binary 📢
🖄 Target Processor	Libraries (-I) 🛛 🗿 🖓 🖓
🖄 Optimization	jidbroadkey pro 256 noenroll m3
🖄 Warnings	
🖄 Debugging	
🗸 🛞 GNU ARM Cross Assembler	
🖄 Preprocessor	
🖄 Includes	
🖄 Warnings	
🖄 Miscellaneous	
🗸 🛞 GNU ARM Cross C Compiler	
Preprocessor	
🖄 Includes	
Optimization	
🖄 Warnings	
🖄 Miscellaneous	
V 🛞 GNU ARM Cross C Linker	Library search path (-L) 🛛 🔞 🎣 🖓 🖓
2 General	"\${workspace_loc:/\${ProjName}/BroadKey/lib}"
Libraries	
Miscellaneous	
V 🛞 GNU ARM Cross Create Flash Image	
😕 General	
V 🛞 GNU ARM Cross Print Size	
ڬ General	

图 3-6 GNU ARM Cross C Link > Libraries 配置

#### 3.2.5 GNU ARM Cross Create Flash Image 配置

配置 "GNU ARM Cross Create Flash Image > General > Output file format (-O)"选项,该选项配置为 "Raw binary",如图 3-7 所示。

图 3-7 GNU ARM Cross Create Flash Image 配置



## 3.3 编译

单击工具栏编译按钮 "≦",编译软件编程参考设计,产生软件编程二进制 BIN 文件,如图 3-8 所示。

图 3-8 编译

🐝 C/C++ - bk_with_no_enroll_GW1NSER_4C/USER	/main.c - GOWIN MCU Designer	- 🗆 ×
<u>F</u> ile <u>E</u> dit <u>S</u> ource Refac <u>t</u> or <u>N</u> avigate Se <u>a</u> rch	<u>P</u> roject <u>R</u> un <u>W</u> indow <u>H</u> elp	
	購:☆▼:@ ∥ ▼: / 刷 同 同: 例 ▼ 到 ▼ い ひ ▼ ひ ▼	Ouick Access
🏠 Project Explorer 🛛 📄 🖶 🍸 🖳 🗖	le main.c ⊠	Bo ≈ "2 □
📋 adc 🔨 🔨		🖻 🔓 🖉 🖋 🛛 🗰 🗸
🗀 ahb_arbiter	14	🖬 gw1ns4c.h
i ahb2_exi	15 /* Includes*/	broadkey_demo_app
📁 apb2_exi	16 #include "gw1ns4c.h"	iid_print.h
bk_with_enroll_manufacture	17 #include broadkey_demo_app.n 18 #include "iid print.h"	🛀 iid_usart.h
bk_with_enroll_manufacture_GW1NSER_4	19 #include "iid_usart.h"	_puf : int
bk_with_no_enroll_1.1	20	_ac : int
✓ <sup>Sb</sup> bk_with_no_enroll_GW1NSER_4C	22 / / imports	# FLASH_AC_BASE
> 🖑 Binaries	23 extern intac; //define in gw1ns4c_flash.ld AC_FLASH address	++ <sup>S</sup> ReadFlashDetain(uint:
> 🔊 Includes	24	++ <sup>\$</sup> hex2char(uint8_t) : uir
> 👝 BroadKey	25 /* MACTOS	*** print_byte(uint8_t) : vc
> 👝 CORE	27	main() : int
> 👝 Debug	28 /* Declaractions. */	S hex2char(uint8_t) : uir
> 👝 PERIPHERAL	29 Static winds the acchar(units t wa):	S print_byte(uint8_t) : vc     S
> 👝 Script	31 static void print_byte(uint8_t va);	<ul> <li><sup>S</sup> ReadFlashDetain(uint:</li> </ul>
> 👝 STARTUP	32	
> 👝 SYSTEM	340 int main()	
V 🗁 USER	35 {	
> c endian.c	36 uint8_t * sram_puf = (uint8_t *)&_puf; 37 uint8_t * sram_ac = (uint8_t *)&_ac;	
> 🔓 endian.h	38	
> h gw1ns4c_conf.h	<pre>39 uint32_t data[256] = {0};</pre>	
> _ gw1ns4c_it.c	40 uint32_t i = 0; 41 uint8 + *n = (uint8 + *)data;	
> h gw1ns4c_it.h	42	
> C main.c	43 SvstemInit()://init svstem	
cm1_ahb2_exi		
<pre>cm1_apb2_exi</pre>	🕼 Problems 🧟 Tasks 🖳 Console 🛛 🔲 Properties 🤬 Debugger Console 🛛 🕹 🗘 😫 🌆 🐻 😑	🖳 📑 🖳 🕶 📑 🗖 👘
Cm1_can	CDT Build Console [bk_with_no_enroll_GW1NSER_4C]	
Cm1_ddr3	14:44:46 **** Incremental Build of configuration Debug for project bk_with_no_enroll_GWINSER_4C ****	· ·
cm1_dualtimer	make all	
cml_ethernet	arm-none-eabi-sizeformat=berkeley "bk with no enroll GW1NSER 4C.elf"	
cm I_tatts	text data bss dec hex filename	
cm I_freertos	27928 1084 1164 30176 75e0 bk_with_no_enroll_GWINSER_4C.elf	
cmi_izc	Tarance personny primarine in ortination activity	
cmi_keyscan		
cmiled	14:44:47 Build Finished (took 656ms)	
cm (_led_boot_embfiash		~
< >>	<	>

## 3.4 下载

使用下载工具 Programmer,下载软件编程二进制 BIN 文件。

选择 GMD 菜单栏 Run > Programmer 或工具栏 "J题",打开下载工具 Programmer,如图 3-9 所示。

#### 注!

亦可通过以下方式打开 Programmer: 高云云源软件。



图 3-9 Programmer 打开方式

单击 Programmer 菜单栏 Edit > Configure Device 或工具栏 Configure Device "☞",打开 Device configuration。

## 3.4.1 GW1NSE-2C (Gowin\_EMPU(GW1NS-2C))

如果是安全芯片 GW1NSE-2C, 下载属性配置, 如图 3-10 所示。

Access Mode 下拉列表,选择"SecureFPGA Mode"选项。

Operation 下拉列表,选择"Firmware Erase, Program Securely"选项。

"FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入 软件编程二进制 BIN 文件。

单击"Save",完成下载属性配置。

#### 图 3-10 下载属性配置

🙀 Device configuration		?	×		
Device Operation					
Access Mode:	SecureFPGA Mode		•		
Operation:	Firmware Erase, Program Securely		•		
Firmware Erase, Progra Key Authentication Cod	um,Verify with Security(One Time Programm le)	able For	-		
FW/MCU/Binary Input Options					
Firmware/Binary File: o_enroll_1.1/Debug/bk_with_no_enroll_1.1.bin					
	Save	Cance	1		

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NSE-2C 软件编程二进制 BIN 文 件下载。

## 3.4.2 GW1NSER-4C (Gowin\_EMPU(GW1NS-4C))

如果是安全芯片 GW1NSER-4C,下载属性配置,如图 3-11 所示。

Access Mode 下拉列表,选择"SecureFPGA Mode"选项。

Operation 下拉列表,选择"Firmware Erase, Program Securely"选项。

"Programming Options > File name"选项,导入硬件设计码流文件。

"FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入 软件编程二进制 BIN 文件。

单击"Save",完成下载属性配置。

#### 图 3-11 下载属性配置

🙀 Device configuration		?	$\times$		
Device Operation					
Access Mode:	SecureFPGA Mode		•		
Operation:	Firmware Erase, Program Securely		•		
Firmware Erase, Program, Verify with Security(One Time Programmable For Key Authentication Code)					
Programming Options File name: W1NSER4C_QN48G_V1.1/secure_fpga/impl/pnr/secure_fpga.fs User Flash Initialization					
FW/MCU/Binary Input Options Firmware/Binary File: ER_4C/Debug/bk_with_no_enroll_GW1NSER_4C.bin					
	Save	Cance	1		

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NSER-4C 硬件设计码流文件和 软件编程二进制 BIN 文件下载。

#### 3.4.3 GW1NE-9C

如果是安全芯片 GW1NE-9C,下载属性配置,如图 3-12 所示。

Access Mode 下拉列表,选择"SecureFPGA Mode L"选项。

Operation 下拉列表,选择"Firmware Erase, Program Securely"选项。

"Programming Options > File name"选项,导入硬件设计码流文件。

"FW/MCU/Binary Input Options > Firmware/Binary File"选项,导入 软件编程二进制 BIN 文件。

单击"Save",完成下载属性配置。

#### 图 3-12 下载属性配置

🔆 Device configuration		?	$\times$			
Device Operation						
Access Mode:	SecureFPGA Mode L		•			
Operation:	Firmware Erase, Program Securely 🔹					
Firmware Erase, Program, Verify with Security(One Time Programmable For Key Authentication Code)						
Programming Options						
File name:         C:/Users/liukai/Desktop/secure_fpga.fs						
User Flash Initialization						
FW/MCU/Binary Input Options						
Firmware/Binary File:	ers/liukai/Desktop/GW1NS-9C_bk_no_enroll.	bin .				
	Save	Cance	1			

完成 Device configuration 后,单击 Programmer 工具栏 Program/Configure,完成安全芯片 GW1NSE-9C 硬件设计码流文件和软件编程二进制 BIN 文件下载。

## 3.5 参考手册

Gowin SecureFPGA产品GW1NSE-2C(Gowin\_EMPU(GW1NS-2C))、 GW1NSER-4C(Gowin\_EMPU(GW1NS-4C))和GW1NE-9C (Gowin EMPU M1)软件编程设计,请参考:

- IPUG519, Gowin\_EMPU(GW1NS-2C) IDE 软件参考手册
- IPUG516, Gowin\_EMPU(GW1NS-2C)软件编程参考手册
- IPUG928, Gowin\_EMPU(GW1NS-4C)\_IDE 软件参考手册
- IPUG931, Gowin\_EMPU(GW1NS-4C)软件编程参考手册
- IPUG536, Gowin\_EMPU\_M1 IDE 软件参考手册
- IPUG533, Gowin\_EMPU\_M1 软件编程参考手册
- SUG502, Gowin Programmer 用户指南

