



# Gowin\_EMPU\_M1 硬件设计 参考手册

IPUG531-1.4,2020-01-16

## **版权所有©2020 广东高云半导体科技股份有限公司**

未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、翻译本档内容的部分或全部，并不得以任何形式传播。

### **免责声明**

本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档进行适时的更新。

## 版本信息

日期	版本	说明
2019/02/19	1.0	初始版本。
2019/07/18	1.1	MCU 硬件设计支持扩展外部设备 CAN、Ethernet、SPI-Flash、RTC、DualTimer、TRNG、I2C、SPI、SD-Card。
2019/08/18	1.2	<ul style="list-style-type: none"><li>● MCU 硬件设计与软件编程设计支持扩展外部设备 DDR3 Memory;</li><li>● 修复已知 ITCM、DTCM Size 和 IDE 问题。</li></ul>
2019/09/27	1.3	<ul style="list-style-type: none"><li>● MCU 硬件设计与软件编程设计支持外部设备 SPI-Flash 的读、写和擦除功能;</li><li>● MCU 软件编程设计支持外部设备 I2C 一次连续多字节读、写功能;</li><li>● 修复已知 MCU 软件编程设计中 AHB2 扩展接口和 APB2 扩展接口地址映射问题;</li><li>● 修复已知 MCU 软件编程设计中 DDR3 Memory 连续读、写问题。</li></ul>
2019/12/06	1.4	<ul style="list-style-type: none"><li>● MCU 硬件设计与软件编程设计支持外部设备 PSRAM;</li><li>● 更新 MCU 编译软件 GMD V1.0;</li><li>● 更新 RTOS 参考设计;</li><li>● 增加 AHB2 和 APB2 扩展总线接口硬件和软件参考设计。</li></ul>

# 目录

目录 .....	i
图目录 .....	iii
表目录 .....	iii
<b>1 硬件架构 .....</b>	<b>1</b>
1.1 系统架构 .....	1
1.2 系统特征 .....	2
1.2.1 Cortex-M1 子系统 .....	2
1.2.2 AHB-Lite Extension 子系统 .....	3
1.3 系统端口 .....	3
1.3.1 Cortex-M1 端口 .....	3
1.3.2 AHB-Lite Extension 端口 .....	3
<b>2 硬件设计流程 .....</b>	<b>8</b>
2.1 硬件环境 .....	8
2.2 软件环境 .....	8
2.3 软核生成器 .....	8
2.4 下载软件 .....	8
2.5 设计流程 .....	8
<b>3 工程模板 .....</b>	<b>10</b>
3.1 工程创建 .....	10
3.1.1 新建工程 .....	10
3.1.2 设定工程名称和路径 .....	10
3.1.3 选择器件 .....	11
3.1.4 完成工程创建 .....	11
3.2 硬件设计 .....	12
3.2.1 Cortex-M1 硬件设计 .....	13
3.2.2 AHB-Lite Extension 硬件设计 .....	18
3.3 用户设计 .....	36
3.4 约束 .....	36

---

3.5 配置 .....	37
3.5.1 顶层模块配置 .....	37
3.5.2 Post-Place File 配置 .....	37
3.5.3 Dual-Purpose Pin 配置 .....	38
3.6 综合 .....	39
3.7 布局布线.....	39
<b>4 参考设计 .....</b>	<b>42</b>

# 图目录

图 1-1 Gowin_EMPU_M1 系统架构 .....	1
图 3-1 新建 FPGA Design 工程 .....	10
图 3-2 设定工程名称和路径 .....	11
图 3-3 选择器件 .....	11
图 3-4 完成工程创建 .....	12
图 3-5 选择 Gowin_EMPU_M1 .....	12
图 3-6 Gowin_EMPU_M1 系统架构 .....	13
图 3-7 Cortex-M1 配置页面 .....	14
图 3-8 Cortex-M1 通用配置 .....	15
图 3-9 Cortex-M1 调试配置 .....	16
图 3-10 Cortex-M1 存储配置 .....	17
图 3-11 GPIO 配置 .....	19
图 3-12 CAN 配置 .....	20
图 3-13 Ethernet 配置 .....	21
图 3-14 DDR3 配置 .....	22
图 3-15 SPI-Flash 配置 .....	23
图 3-16 AHB2 Extension 配置 .....	24
图 3-17 UART 配置 .....	25
图 3-18 Timer 配置 .....	26
图 3-19 WatchDog 配置 .....	27
图 3-20 RTC 配置 .....	28
图 3-21 DualTimer 配置 .....	29
图 3-22 TRNG 配置 .....	30
图 3-23 I2C 配置 .....	31
图 3-24 SPI 配置 .....	32
图 3-25 SD-Card 配置 .....	33
图 3-26 APB2 Extension 配置 .....	34
图 3-27 PSRAM 配置 .....	35
图 3-28 综合工具配置 .....	36

---

图 3-29 顶层模块配置 .....	37
图 3-30 Post-Place File 配置 .....	38
图 3-31 Dual-Purpose Pin 配置 .....	38
图 3-32 综合 .....	39
图 3-33 Place & Route .....	39
图 3-34 GW1N 系列 Configure Device .....	40
图 3-35 GW2A 系列 Configure Device .....	41

# 表目录

表 1-1 Cortex-M1 端口 .....	3
表 1-2 AHB-Lite Extension 端口 .....	3
表 3-1 Cortex-M1 配置选项 .....	13
表 3-2 AHB-Lite Extension 配置选项 .....	18

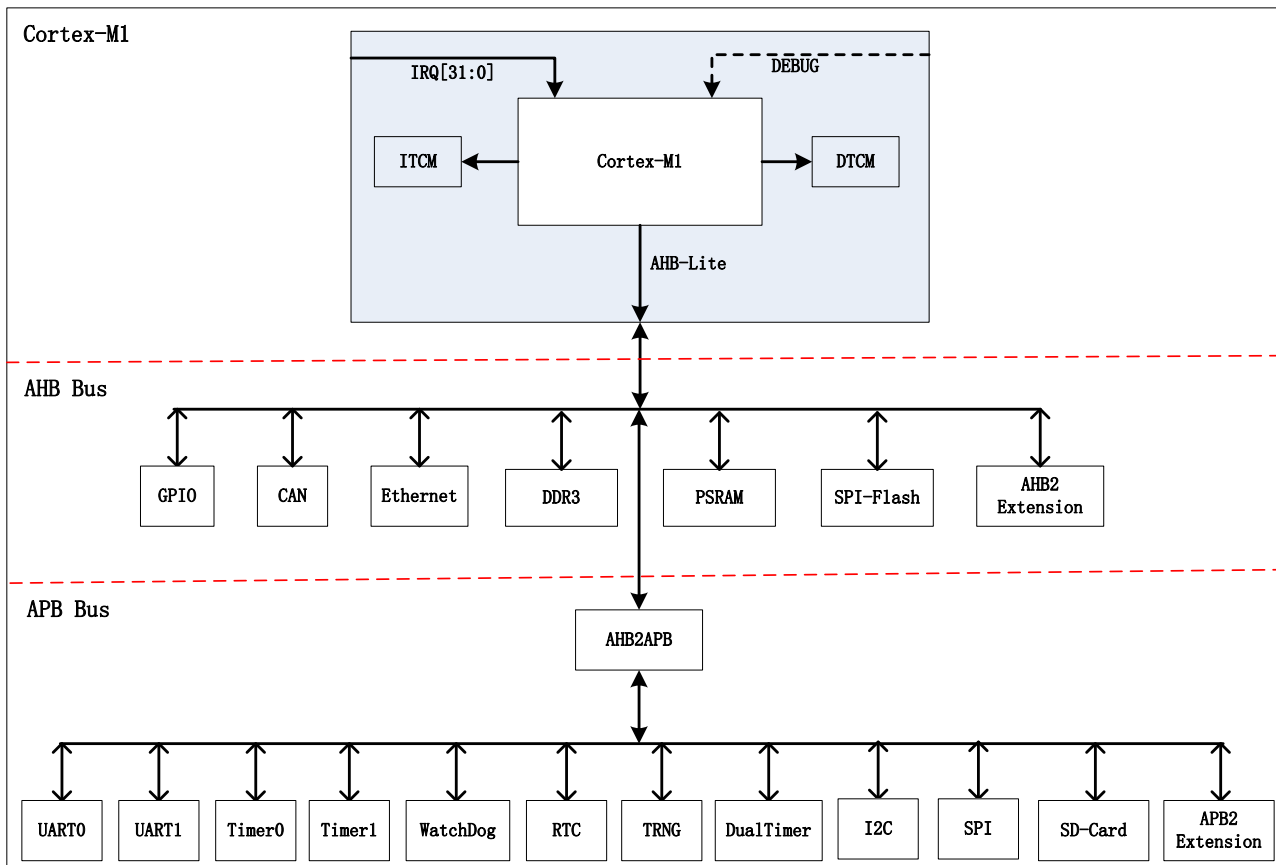


# 1 硬件架构

## 1.1 系统架构

Gowin\_EMPU\_M1 包括三级结构，如图 1-1 所示。

图 1-1 Gowin\_EMPU\_M1 系统架构



第一级，Cortex-M1 及 ITCM、DTCM

第二级，AHB 总线及 GPIO、CAN、Ethernet、DDR3 Memory、PSRAM、SPI-Flash、AHB2 Extension Bus

第三级，APB 总线及 UART0、UART1、Timer0、Timer1、Watch Dog、

RTC、TRNG、DualTimer、I2C、SPI、SD-Card、APB2 Extension Bus

## 1.2 系统特征

Gowin\_EMPU\_M1 包括两个子系统:

- Cortex-M1 内核子系统
- AHB-Lite 扩展 AHB 总线和 APB 总线, 及外设接口子系统

### 1.2.1 Cortex-M1 子系统

#### 处理器内核

- ARM architecture v6-M Thumb 指令集架构, 支持 16-bit Thumb 和 32-bit Thumb2 指令集
- 可配置扩展操作系统
- 系统异常处理
- 中断异常处理和正常线程模式
- 栈指针, 正常是一个栈指针, 扩展操作系统时两个栈指针
- 大小端格式
  - 可配置数据大小端格式
  - 指令和系统控制寄存器小端格式
  - 调试系统小端格式

#### NVIC

- 可配置外部中断数量, 1、8、16、32
- 4 个优先级等级
- 进入中断处理时自动保存处理器状态, 中断处理结束时自动恢复状态

#### 调试系统

- 不支持调试系统
- 支持调试系统
  - 可配置完整 (full) 模式和简化 (reduced) 模式
    - 完整模式: 4 个 BreakPoint Unit 和 2 个 Data Watchpoint
    - 简化模式: 2 个 BreakPoint Unit 和 1 个 Data Watchpoint
  - 可配置 DAP 端口
    - JTAG/SW
    - JTAG
    - SW

#### Memory

- ITCM: 指令存储器, 可配置大小, Upper 或 Lower alias 方式和读入初始值
- DTCM: 数据存储器, 可配置大小

#### 32 位硬件乘法器

- Normal 模式
- Small 模式

## 1.2.2 AHB-Lite Extension 子系统

- AHB 总线, 及 GPIO、CAN、Ethernet、DDR3 Memory、PSRAM、SPI-Flash、AHB2 Extension
- APB 总线, 及 UART0、UART1、Timer0、Timer1、Watch Dog、RTC、DualTimer、TRNG、I2C、SPI、SD-Card、APB2 Extension

## 1.3 系统端口

### 1.3.1 Cortex-M1 端口

Cortex-M1 端口如表 1-1 所示。

表 1-1 Cortex-M1 端口

名称	I/O	位宽	描述
HCLK	in	1	System clock
SYSRESETn	in	1	System reset
DBGRESETn	in	1	Power on debug reset
LOCKUP	out	1	Core is in lockup state
HALTED	out	1	Core is in Halt debug state
nTRST	in	1	JTAG reset
SWCLKTCK	in	1	Serial wire and JTAG clock
SWDITMS	in	1	SW Data / JTAG Test Mode Select
TDI	in	1	JTAG data input
JTAGNSW	out	1	JTAG = 1, serial wire = 0
JTAGTOP	out	1	state controller indicator
TDO	out	1	JTAG data output
nTDOEN	out	1	JTAG data out enable
SWDO	out	1	Serial wire data out
SWDOEN	out	1	Serial data output enable
IRQ	in	[31:0]	External interrupts
NMI	in	1	Non-maskable interrupt
SYSRESETREQ	out	1	System reset require
EDBGRQ	in	1	External debug request
DBGRESTART	in	1	Restart from halt request
DBGRESTARTED	out	1	Restart from halt acknowledge
HREADY	in	1	Slave ready signal
HRESP	in	1	Slave response signal
HRDATA	in	[31:0]	Data from slave to master
HTRANS	out	[1:0]	Transfer type
HBURST	out	[2:0]	Burst type
HPROT	out	[3:0]	Transfer protection bits
HSIZE	out	[2:0]	Transfer size
HWRITE	out	1	Transfer direction
HMASTLOCK	out	1	Transfer is a locked transfer
HADDR	out	[31:0]	Transfer address
HWDATA	out	[31:0]	Data from master to slave

### 1.3.2 AHB-Lite Extension 端口

AHB-Lite Extension 端口如表 1-2 所示。

表 1-2 AHB-Lite Extension 端口

名称	I/O	位宽	描述	所属模块
----	-----	----	----	------

名称	I/O	位宽	描述	所属模块
SYSRESETn	in	1	系统复位	-
PORESETn	in	1	上电复位	-
IRQ	out	[31:0]	外部中断	-
NMI	out	1	Non-Maskable 中断	-
GPIO	inout	[15:0]	通用输入输出端口	GPIO
UART0RXD	in	1	UART0 接收端口	UART0
UART0TXD	out	1	UART0 发送端口	
UART1RXD	in	1	UART1 接收端口	UART1
UART1TXD	out	1	UART1 发送端口	
TIMER0EXTIN	in	1	Timer0 外部中断	Timer0
TIMER1EXTIN	in	1	Timer1 外部中断	Timer1
WDOGRESREQ	out	1	Watch Dog 复位请求	Watch Dog
RTCSRCCLK	in	1	RTC 时钟源 32.768KHz	RTC
SCL	inout	1	串行时钟	I2C
SDA	inout	1	串行数据	
MOSI	out	1	主设备输出/从设备输入	SPI
MISO	in	1	主设备输入/从设备输出	
SCLK	out	1	时钟信号	
NSS	out	1	从设备选择信号	
SD_CLK	out	1	时钟信号	SD-Card
SD_CS	out	1	片选信号	
SD_DATAIN	out	1	数据输入	
SD_DATAOUT	in	1	数据输出	
SD_CARD_INIT	out	1	初始化"0"	
SD_CHECKIN	in	1	输入检查	
SD_CHECKOUT	out	1	输出检查	
CAN_RX	in	1	数据输入	CAN
CAN_TX	out	1	数据输出	
RGMII_TXC	out	1	RGMII 发送时钟	Ethernet RGMII Interface
RGMII_TX_CTL	out	1	RGMII 发送控制	
RGMII_TXD	out	[3:0]	RGMII 发送数据	
RGMII_RXC	in	1	RGMII 接收时钟	
RGMII_RX_CTL	in	10	RGMII 接收控制	
RGMII_RXD	in	[3:0]	RGMII 接收数据	
GTX_CLK	in	1	RGMII 125MHz 时钟输入	
GMII_RX_CLK	in	1	GMII 接收时钟	Ethernet GMII Interface
GMII_RX_DV	in	1	GMII 接收使能	
GMII_RXD	in	[7:0]	GMII 接收数据	

名称	I/O	位宽	描述	所属模块
GMII_RX_ER	in	1	GMII 接收错误	
GTX_CLK	in	1	GMII 125MHz 时钟输入	
GMII_GTX_CLK	out	1	GMII 发送时钟	
GMII_TXD	out	[7:0]	GMII 发送数据	
GMII_TX_EN	out	1	GMII 发送使能	
GMII_TX_ER	out	1	GMII 发送错误	
MII_RX_CLK	in	1	MII 接收时钟	Ethernet MII Interface
MII_RXD	in	[3:0]	MII 接收数据	
MII_RX_DV	in	1	MII 接收使能	
MII_RX_ER	in	1	MII 接收错误	
MII_TX_CLK	in	1	MII 发送时钟	
MII_TXD	out	[3:0]	MII 发送数据	
MII_TX_EN	out	1	MII 发送使能	
MII_TX_ER	out	1	MII 发送错误	
MII_COL	in	1	MII 冲突信号	
MII_CRD	in	1	MII 载波信号	
MDC	out	1	管理通道时钟	Ethernet
MDIO	inout	1	管理通道数据	
DDR_CLK_I	in	1	50MHz 时钟输入	DDR3
DDR_INIT_COMPLETE_O	out	1	初始化完成信号	
DDR_ADDR_O	out	[15:0]	Row 地址、Column 地址	
DDR_BA_O	out	[2:0]	Bank 地址	
DDR_CS_N_O	out	1	片选信号	
DDR_RAS_N_O	out	1	Row 地址选通信号	
DDR_CAS_N_O	out	1	Column 地址选通信号	
DDR_WE_N_O	out	1	Row 写使能	
DDR_CLK_O	out	1	提供给 DDR3 SDRAM 的时钟信号	
DDR_CLK_N_O	out	1	与 DDR_CLK_O 组成差分信号	
DDR_CKE_O	out	1	DDR3 SDRAM 时钟使能信号	
DDR_ODT_O	out	1	内存信号端接电阻控制	
DDR_RESET_N_O	out	1	DDR3 SDRAM 复位信号	
DDR_DQM_O	out	[1:0]	DDR3 SDRAM 数据屏蔽信号	
DDR_DQ_IO	inout	[15:0]	DDR3 SDRAM 数据	
DDR_DQS_IO	inout	[1:0]	DDR3 SDRAM 数据选通信号	

名称	I/O	位宽	描述	所属模块
DDR_DQS_N_IO	inout	[1:0]	与 DDR_DQS_IO 组成差分信号	
O_psrām_ck	out	[1:0]	提供给 PSRAM 的时钟信号	PSRAM
O_psrām_ck_n	out	[1:0]	与 O_psrām_ck 组成差分信号	
IO_psrām_rwds	inout	[1:0]	PSRAM 数据选通信号及掩码信号	
IO_psrām_dq	inout	[15:0]	PSRAM 数据	
O_psrām_reset_n	out	[1:0]	PSRAM 复位信号	
O_psrām_cs_n	out	[1:0]	片选, 低有效	
init_calib	out	1	初始化完成信号	
psrām_ref_ck	in	1	参考输入时钟, 一般为板载晶振时钟	
psrām_memory_ck	in	1	用户输入颗粒工作时钟, 一般为 PLL 倍频出来的高速时钟, 也可以不使用 PLL	
FLASH_SPI_HOLDN	inout	1	NC	
FLASH_SPI_CSN	inout	1	从设备选择信号	
FLASH_SPI_MISO	inout	1	主设备输入/从设备输出	
FLASH_SPI_MOSI	inout	1	主设备输出/从设备输入	
FLASH_SPI_WPN	inout	1	NC	
FLASH_SPI_CLK	inout	1	时钟信号	
APB2PSTRB	out	[3:0]	APB2 PSTRB	APB2 Extension
APB2PPROT	out	[2:0]	APB2 PPROT	
APB2PSEL	out	1	APB2 PSEL	
APB2PENABLE	out	1	APB2 PENABLE	
APB2PADDR	out	[31:0]	APB2 PADDR	
APB2PWRITE	out	1	APB2 PWRITE	
APB2PWDATA	out	[31:0]	APB2 PWDATA	
APB2PRDATA	in	[31:0]	APB2 PRDATA	
APB2PREADY	in	1	APB2 PREADY	
APB2PSLVERR	in	1	APB2 PSLVERR	
APB2PCLK	out	1	APB2 PCLK	
APB2PRESET	out	1	APB2 RESET	
AHB2HSEL	out	1	AHB2 HSEL	
AHB2HADDR	out	[31:0]	AHB2 HADDR	
AHB2HTRANS	out	[1:0]	AHB2 HTRANS	
AHB2HWRITE	out	1	AHB2 HWRITE	
AHB2HSIZE	out	[2:0]	AHB2 HSIZE	

名称	I/O	位宽	描述	所属模块
AHB2HBURST	out	[2:0]	AHB2 HBURST	
AHB2HPROT	out	[3:0]	AHB2 HPROT	
AHB2HWDATA	out	[31:0]	AHB2 HWDATA	
AHB2HMASTLOCK	out	1	AHB2 HMASTLOCK	
AHB2HREADYMUX	out	1	AHB2 HREADYMUX	
AHB2HRDATA	in	[31:0]	AHB2 HRDATA	
AHB2HREADYOUT	in	1	AHB2 HREADYOUT	
AHB2HRESP	in	[1:0]	AHB2 HRESP	
AHB2HMASTER	out	[3:0]	AHB2 MASTER	
AHB2HCLK	out	1	AHB2 HCLK	
AHB2HRESET	out	1	AHB2 RESET	

# 2 硬件设计流程

## 2.1 硬件环境

- DK-START-GW2A18 V2.0
- GW2A-LV18PG256C8/I7
- DK-START-GW1N9 V1.1  
GW1N-LV9LQ144C6/I5
- DK-START-GW2A55 V1.1
- GW2A-LV55PG484C8/I7
- DK-START-GW2AR18 V1.1  
GW2AR-LV18ELQ144PC6/I5

## 2.2 软件环境

Gowin\_V1.9.3.01 Beta

## 2.3 软核生成器

Gowin 云源软件提供软核生成器 IP Core Generator，用于配置和产生 Cortex-M1 和 AHB-Lite Extension 硬件设计。

IP Core Generator 软件使用方法请参考 [SUG284](#)，Gowin IP Core Generator 用户指南。

## 2.4 下载软件

Gowin\_EMPU\_M1 支持 Programmer 软件下载码流文件。

Programmer 软件使用方法请参考 [SUG502](#)，Gowin Programmer 用户指南。

## 2.5 设计流程

Gowin\_EMPU\_M1 硬件设计流程：

1. IP Core Generator 软核生成器配置 Cortex-M1、APB Bus Peripherals 和 AHB Bus Peripherals，产生 Gowin\_EMPU\_M1 硬件设计，导入工程；
2. 实例化 Gowin\_EMPU\_M1，导入用户设计，连接用户设计与 Gowin\_EMPU\_M1；



3. 物理约束和时序约束;
4. Synplify\_Pro 或 GowinSynthesis 综合;
5. Place & Route 布局布线, 产生码流文件;
6. Programmer 下载码流到 GW1N-9/GW1NR-9/GW2A-18/  
GW2AR-18/GW2A-55。

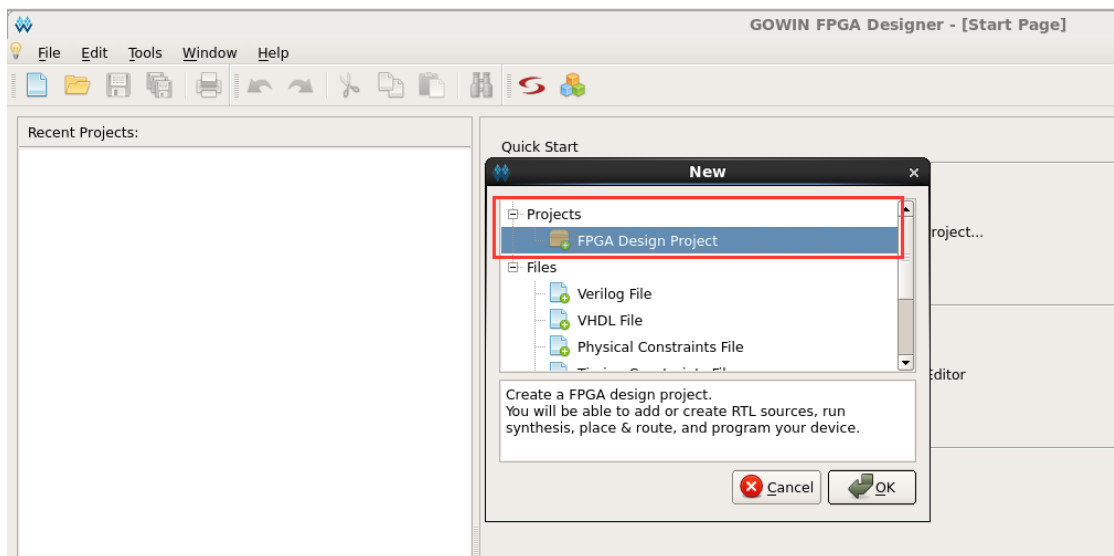
# 3 工程模板

## 3.1 工程创建

### 3.1.1 新建工程

双击“GOWIN FPGA Designer”，选择菜单栏“File > New... > FPGA Design Project”如图 3-1 所示。

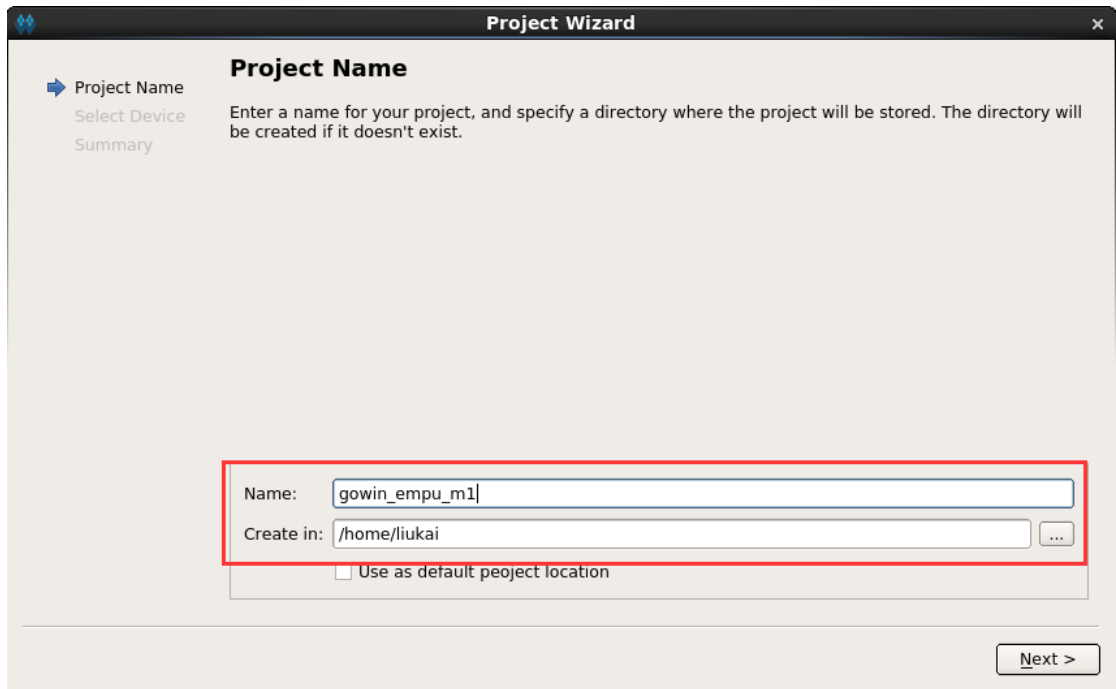
图 3-1 新建 FPGA Design 工程



### 3.1.2 设定工程名称和路径

输入工程名称，选择工程路径，如图 3-2 所示。

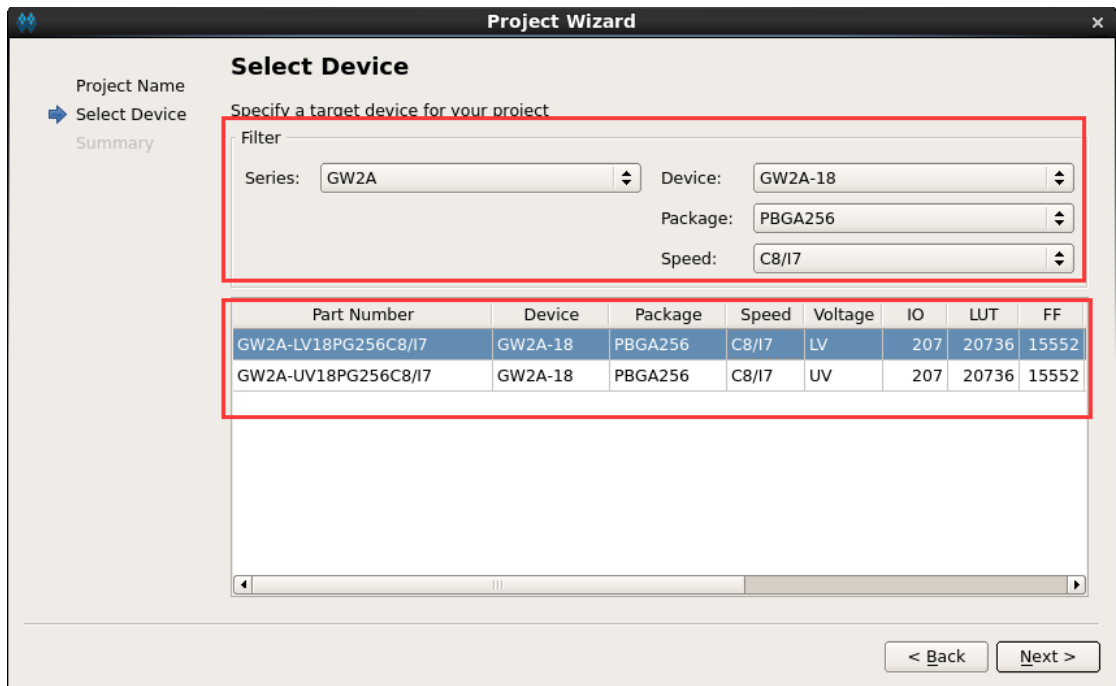
图 3-2 设定工程名称和路径



### 3.1.3 选择器件

选择器件、封装和速度等，如图 3-3 所示。

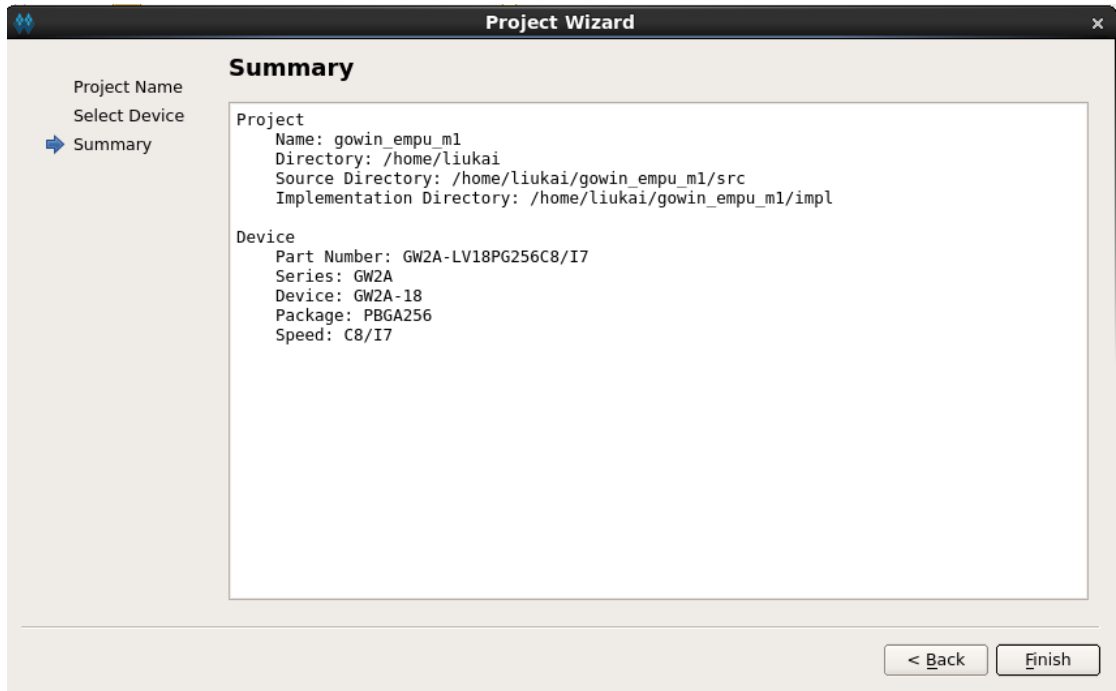
图 3-3 选择器件



### 3.1.4 完成工程创建

如图 3-4 所示，完成新建工程。

图 3-4 完成工程创建

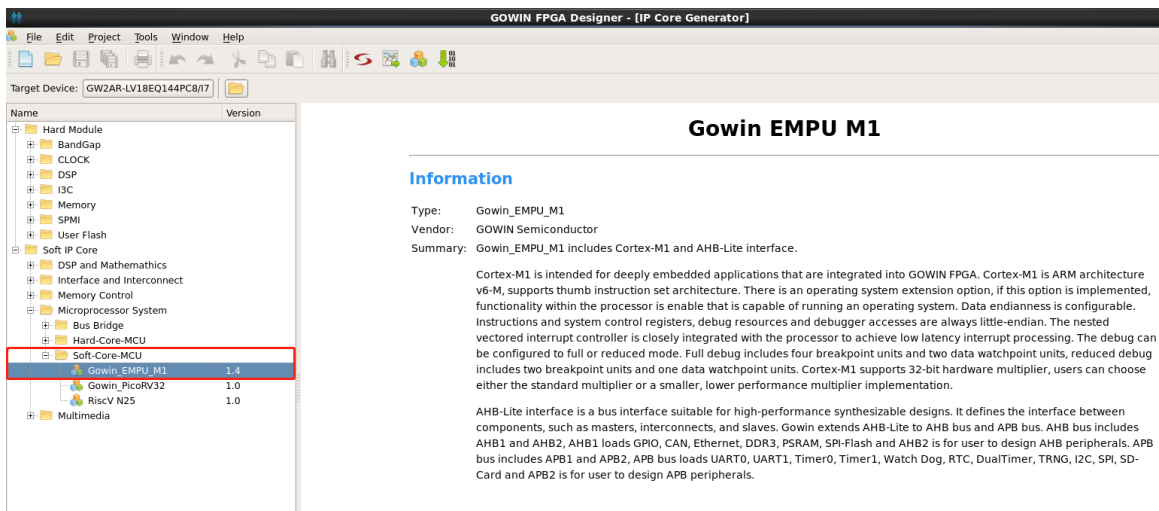


### 3.2 硬件设计

使用 IP Core Generator 产生 Gowin\_EMPU\_M1 硬件设计。

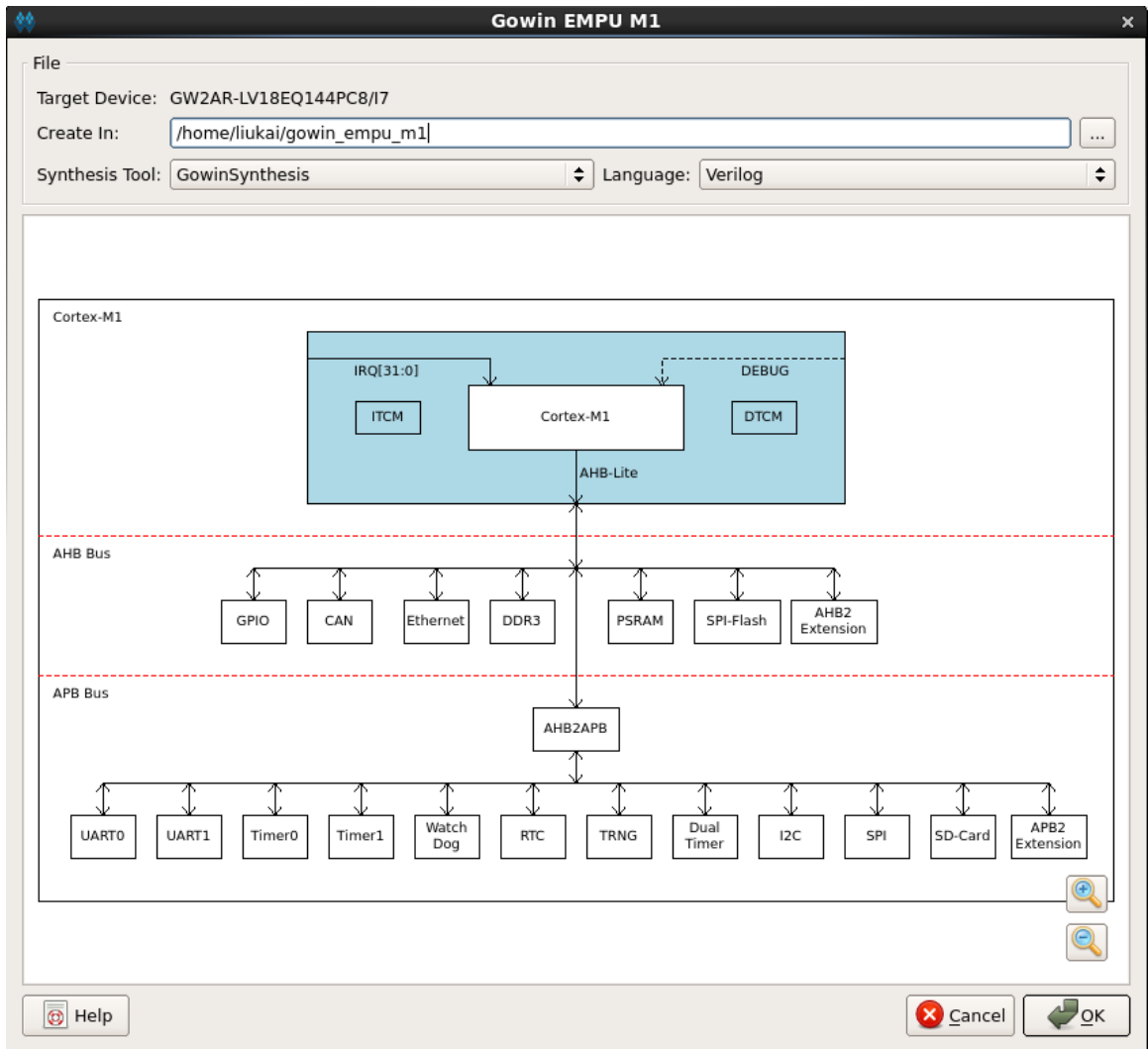
选择菜单栏“Tools > IP Core Generator”或工具栏“IP Core Generator”，打开 IP Core Generator，选择“Soft IP Core > Microprocessor System > Soft-Core-MCU > Gowin\_EMPU\_M1 1.4”，如图 3-5 所示。

图 3-5 选择 Gowin\_EMPU\_M1



打开 Gowin\_EMPU\_M1，Gowin\_EMPU\_M1 系统架构如图 3-6 所示，包括 Cortex-M1 和 AHB Lite Extension（APB Bus Peripherals 和 AHB Bus Peripherals）选择配置。

图 3-6 Gowin\_EMPU\_M1 系统架构



### 3.2.1 Cortex-M1 硬件设计

Cortex-M1 硬件设计配置选项如表 3-1。

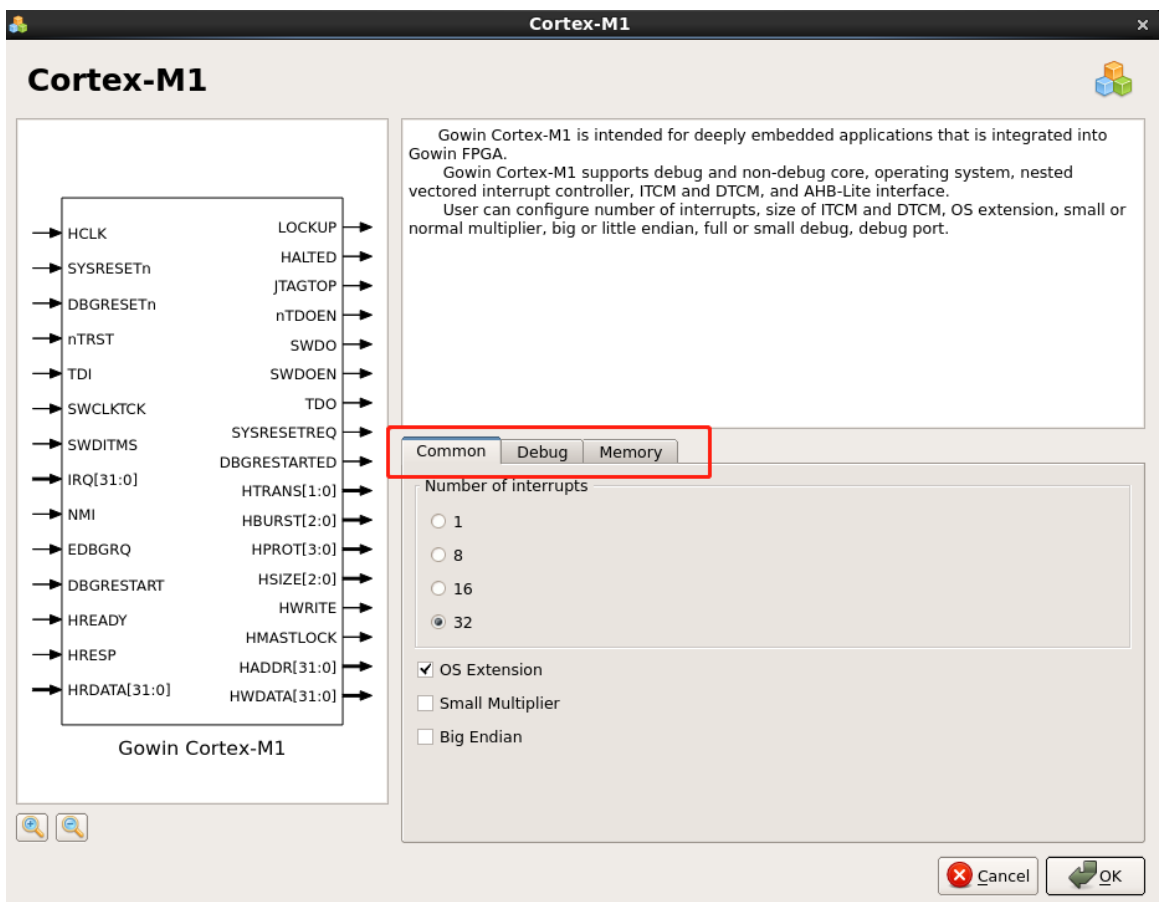
表 3-1 Cortex-M1 配置选项

配置选项	描述
Number of interrupts	配置 Cortex-M1 外部中断数量，可选择 1 或 8 或 16 或 32，默认为 32
OS Extension	配置 Cortex-M1 是否支持操作系统，默认为支持
Small Multiplier	配置 Cortex-M1 small 模式硬件乘法器，默认为 normal 模式
Big Endian	配置 Cortex-M1 数据大端格式，默认为小端格式
Enable Debug	使能 Cortex-M1 Debug 功能，默认为使能 Debug
Debug Port Select	配置调试器接口，可以选择 JTAG 或 Serial Wire 或 JTAG and Serial Wire，默认为 JTAG and Serial Wire
Small Debug	配置 Small 模式调试器，默认为 Full 模式调试器
ITCM Alias Select	配置 ITCM Alias，可以选择 Upper 或 Lower，默认为 Lower Alias
ITCM Size	配置 ITCM 的大小，可以选择 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB

配置选项	描述
	<ul style="list-style-type: none"> <li>● GW1N-9 系列最大选择 32KB，默认为 16KB</li> <li>● GW1NR-9 系列最大选择 32KB，默认为 16KB</li> <li>● GW2A-18 系列最大选择 64KB，默认为 32KB</li> <li>● GW2AR-18 系列最大选择 64KB，默认为 32KB</li> <li>● GW2A-55 系列最大选择 256KB，默认为 64KB</li> </ul>
Initialize ITCM	使能 ITCM 初始化，默认为禁用
ITCM Initialization Path	ITCM 初始值文件路径
DTCM Size	配置 DTCM 的大小，可以选择 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB <ul style="list-style-type: none"> <li>● GW1N-9 系列最大选择 32KB，默认为 16KB</li> <li>● GW1NR-9 系列最大选择 32KB，默认为 16KB</li> <li>● GW2A-18 系列最大选择 64KB，默认为 32KB</li> <li>● GW2AR-18 系列最大选择 64KB，默认为 32KB</li> <li>● GW2A-55 系列最大选择 256KB，默认为 64KB</li> </ul>

双击 Cortex-M1，打开 Cortex-M1 的配置页面，如图 3-7 所示，包括通用配置、调试配置和存储配置。

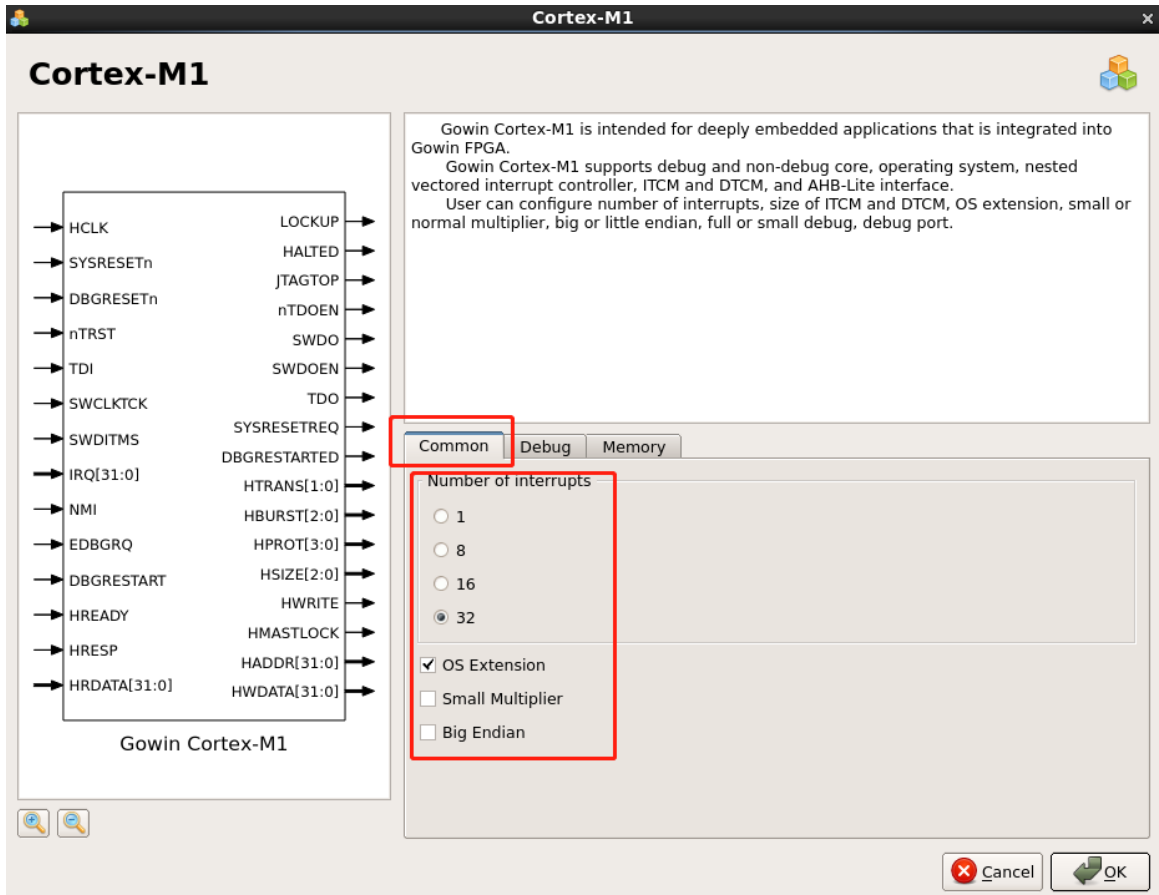
图 3-7 Cortex-M1 配置页面



## 通用配置

选择通用配置选项卡，如图 3-8 所示，可以配置中断数量、操作系统扩展、乘法器模式和数据存储格式。

图 3-8 Cortex-M1 通用配置



- 中断数量配置

可以选择 1 或 8 或 16 或 32，可以配置 1 个或 8 个或 16 个或 32 个外部中断，默认为 32 个。

- 操作系统扩展配置

如果选择，则 Cortex-M1 扩展支持操作系统，默认为支持操作系统扩展。

- 乘法器模式配置

如果选择，则 Cortex-M1 支持 Small 乘法器，否则支持 Normal 乘法器，默认为 Normal 乘法器。

- 数据存储格式配置

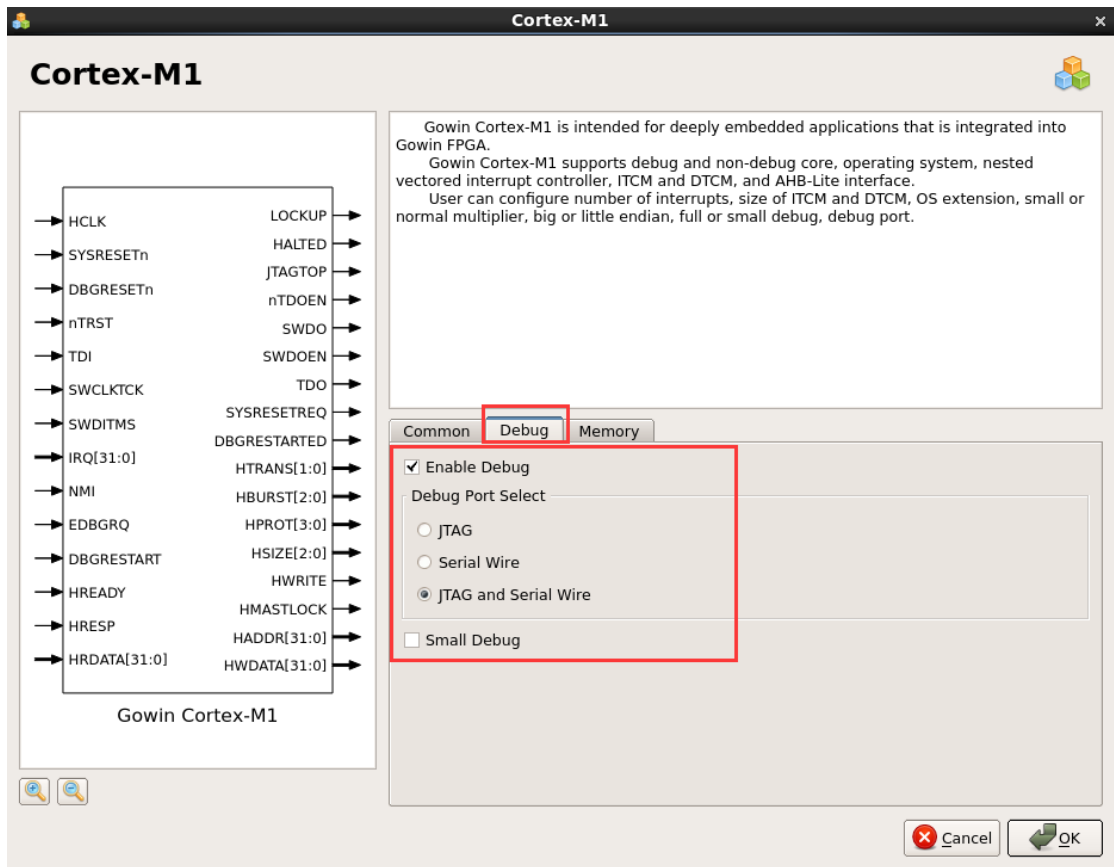
如果选择，则 Cortex-M1 支持数据大端格式，否则支持数据小端格式，默认为小端格式。

## 调试配置

选择调试配置选项卡，如图 3-9 所示，可以配置使能调试、调试接口和

调试器模式。

图 3-9 Cortex-M1 调试配置



- Enable Debug

如果选择 Enable Debug，Cortex-M1 支持调试功能，否则 Cortex-M1 不支持调试功能，默认为使能 Debug。

- 调试接口配置

可以选择 JTAG 或 Serial Wire 或 JTAG and Serial Wire，默认为 JTAG and Serial Wire。

- 调试器模式配置

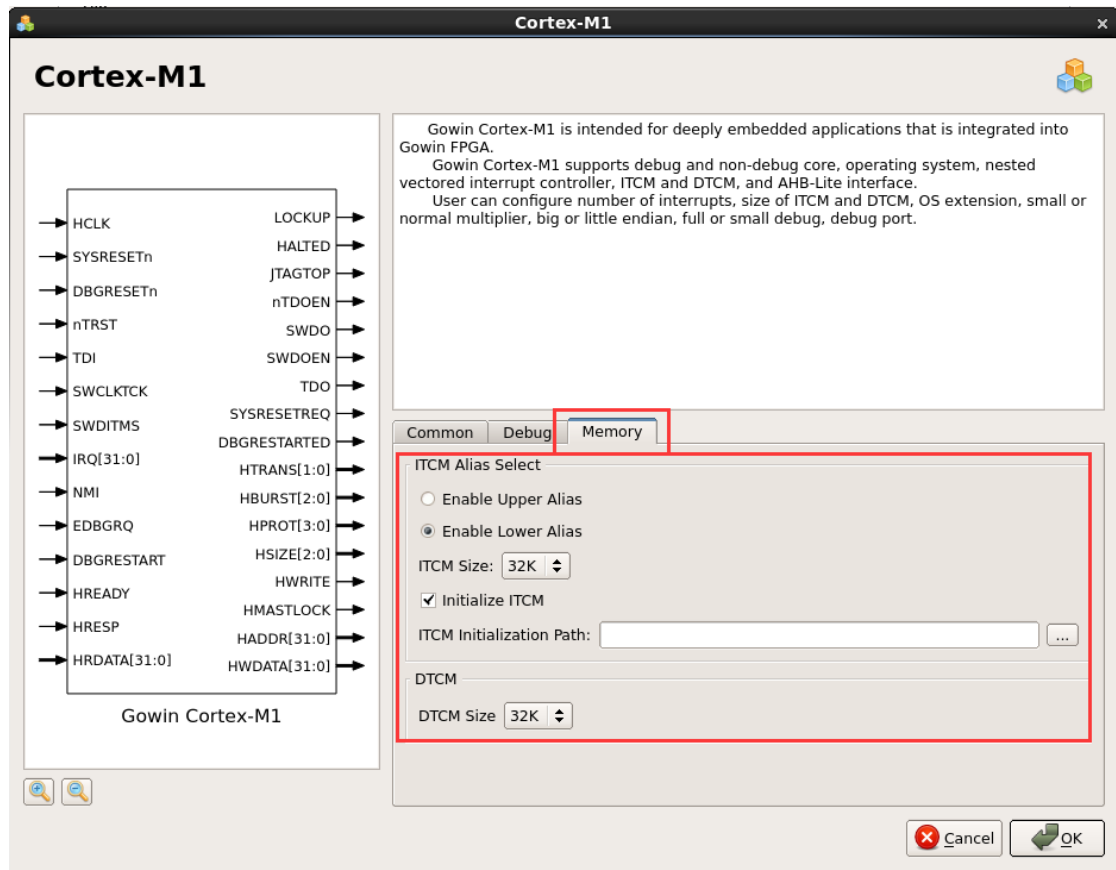
如果选择，则 Cortex-M1 支持 Small 模式调试器，否则支持 Full 模式调试器，默认为 Full 模式调试器。

### 存储配置

选择存储配置选项卡，如图 3-10 所示，可以配置 ITCM 和 DTCM。



图 3-10 Cortex-M1 存储配置



- ITCM 配置

- ITCM Alias 配置可以选择 Enable Upper 或 Lower Alias，默认为 Lower Alias
- ITCM Size 配置
- 可以选择 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB。
- GW1N-9 系列 ITCM Size 最大选择为 32KB，默认为 16KB。
- GW1NR-9 系列 ITCM Size 最大选择为 32KB，默认为 16KB。
- GW2A-18 系列 ITCM Size 最大选择为 64KB，默认为 32KB。
- GW2AR-18 系列 ITCM Size 最大选择为 64KB，默认为 32KB。
- GW2A-55 系列 ITCM Size 最大选择为 256KB，默认为 64KB。
- ITCM Initialization 配置如果选择 Initialize ITCM，则支持 ITCM 初始化，可以在 ITCM Initialization Path 导入 ITCM 初始值文件路径。
- 如果选择使用片外 SPI-Flash 下载启动方式，ITCM 初始值导入 bootloader 文件路径。
- ITCM Initialization Path 导入的路径，不能有以数字命名的文件夹路径！

- DTCM 配置

- DTCM Size 配置可以选择 1KB 或 2KB 或 4KB 或 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB。
- GW1N-9 系列 DTCM Size 最大选择为 32KB，默认为 16KB。

- GW1NR-9 系列 DTCM Size 最大选择为 32KB，默认为 16KB。
- GW2A-18 系列 DTCM Size 最大选择为 64KB，默认为 32KB。
- GW2AR-18 系列 DTCM Size 最大选择为 64KB，默认为 32KB。
- GW2A-55 系列 DTCM Size 最大选择为 256KB，默认为 64KB。
- ITCM 与 DTCM 配置限制
  - GW1N-9 系列，ITCM 或 DTCM 最大可配置为 32KB，如果 ITCM 或 DTCM 某个存储器已配置为 32KB，则另一个存储器最大只能配置为 16KB。
  - GW1NR-9 系列，ITCM 或 DTCM 最大可配置为 32KB，如果 ITCM 或 DTCM 某个存储器已配置为 32KB，则另一个存储器最大只能配置为 16KB。
  - GW2A-18 系列，ITCM 或 DTCM 最大可配置为 64KB，如果 ITCM 或 DTCM 某个存储器已配置为 64KB，则另一个存储器最大只能配置为 16KB。
  - GW2AR-18 系列，ITCM 或 DTCM 最大可配置为 64KB，如果 ITCM 或 DTCM 某个存储器已配置为 64KB，则另一个存储器最大只能配置为 16KB。
  - GW2A-55 系列，ITCM 或 DTCM 最大可配置为 256KB，如果 ITCM 或 DTCM 某个存储器已配置为 256KB，则另一个存储器最大只能配置为 16KB。

### 3.2.2 AHB-Lite Extension 硬件设计

AHB-Lite Extension 配置选项如表 3-2 所示。

表 3-2 AHB-Lite Extension 配置选项

配置选项	描述
Enable GPIO	使能 GPIO，默认关闭
Enable CAN	使能 CAN，默认关闭
Buffer Depth	CAN 选择 Buffer Depth，默认值为 256
Enable Ethernet	使能 Ethernet，默认关闭
Interface	Ethernet 选择 Interface (RGMII/GMII/MII)，默认为 RGMII
RGMII Input Delay	RGMII input delay，默认值为 100
MIIM Clock Divider	MIIM clock divider，默认值为 20
Enable DDR3	使能 DDR3 Memory，默认关闭
Enable PSRAM	使能 PSRAM，默认关闭
Enable SPI-Flash	使能 SPI-Flash 下载功能和 Memory 读、写、擦除功能，默认关闭
Enable AHB2 Extension	使能 AHB2 扩展接口，默认关闭
Enable UART0	使能串口 0，默认关闭
Enable UART1	使能串口 1，默认关闭
Enable Timer0	使能定时器 0，默认关闭
Enable Timer1	使能定时器 1，默认关闭
Enable WatchDog	使能看门狗，默认关闭
Enable RTC	使能 RTC，默认关闭
Enable TRNG	使能 TRNG，默认关闭
Enable DualTimer	使能 DualTimer，默认关闭

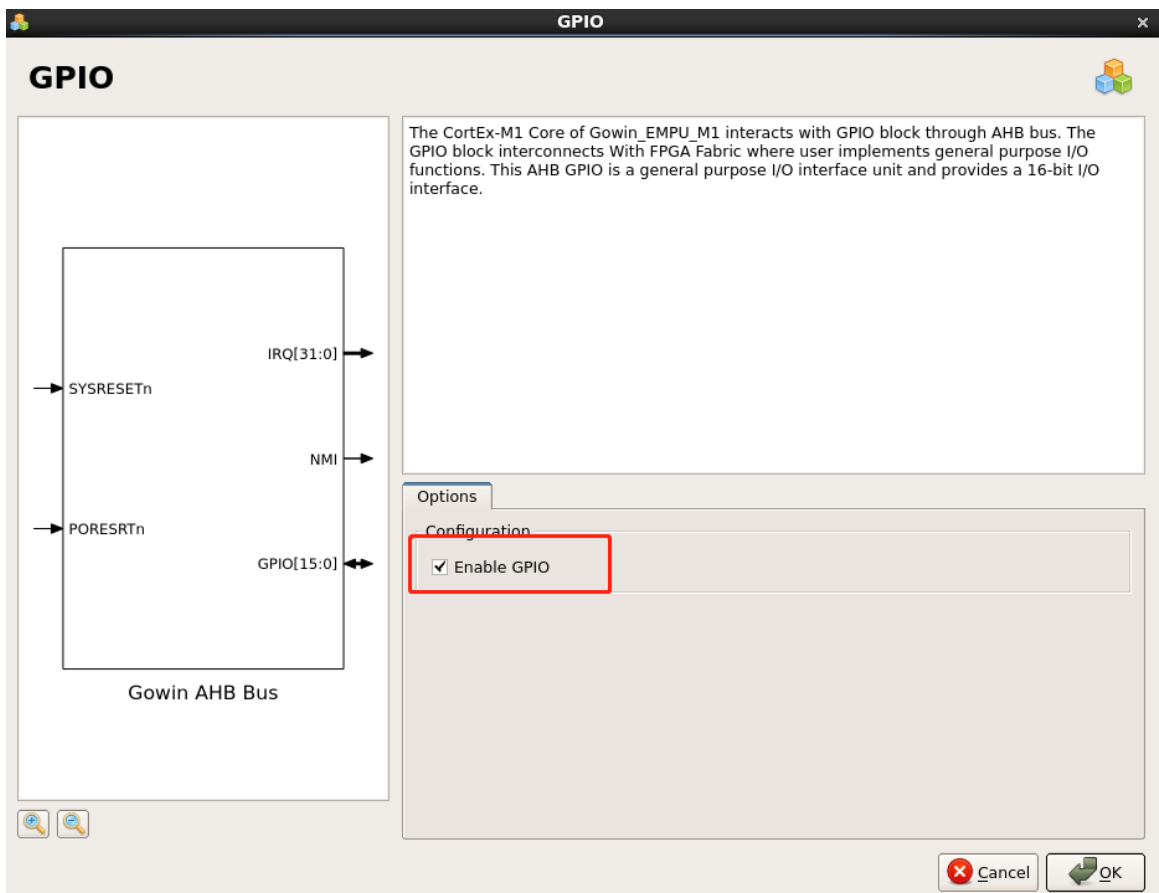
配置选项	描述
Enable I2C	使能 I2C，默认关闭
Enable SPI	使能 SPI，默认关闭
Enable SD-Card	使能 SD-Card，默认关闭
Enable APB2 Extension	使能 APB2 扩展接口，默认关闭

## GPIO 配置

双击打开 GPIO，可以选择配置 GPIO，如图 3-11 所示。

默认关闭，如果选择 Enable GPIO，则 Gowin\_EMPU\_M1 支持 GPIO。

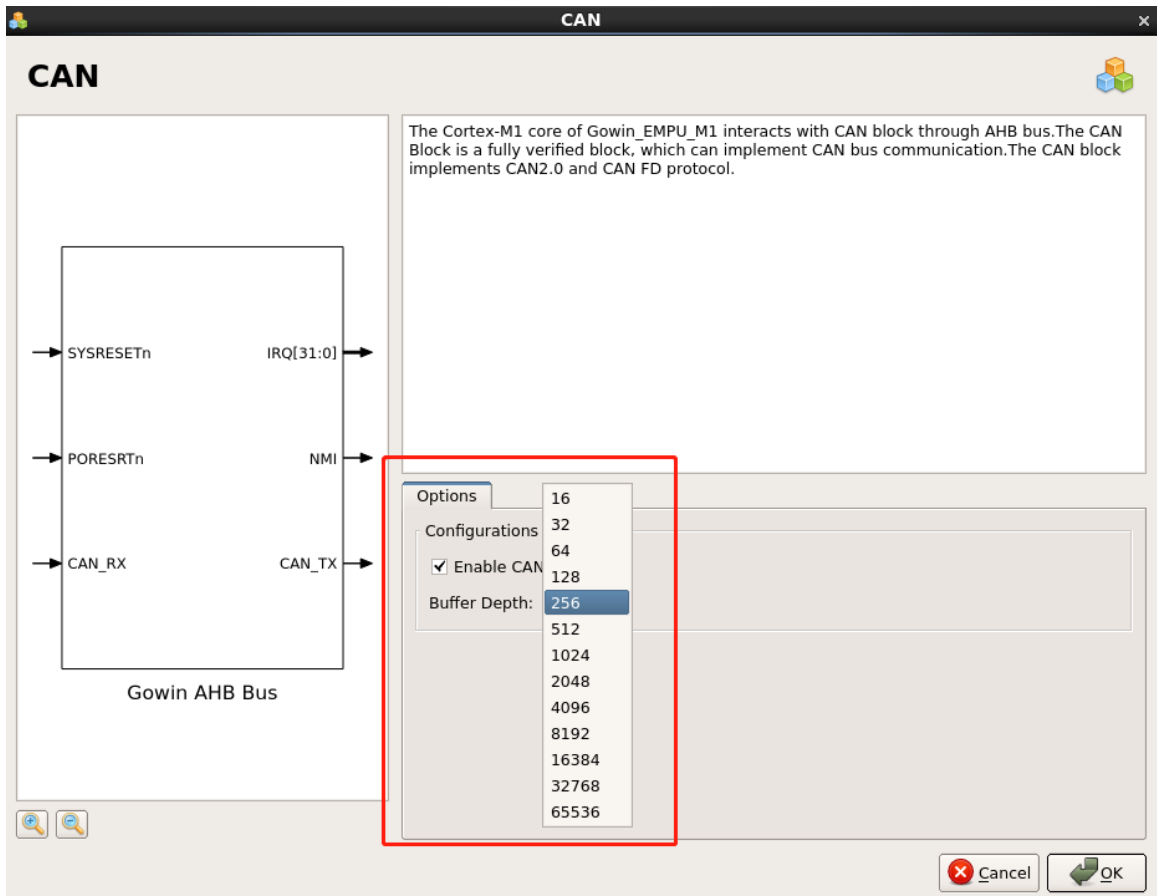
图 3-11 GPIO 配置



## CAN 配置

- 双击打开 CAN，可以选择配置 CAN，如图 3-12 所示。
- 默认关闭，如果选择 Enable CAN，则 Gowin\_EMPU\_M1 支持 CAN。
- 如果已经选择 Enable CAN，则可以配置 Buffer Depth。
- 选择 Buffer Depth，配置 Buffer Depth，默认值为 256。

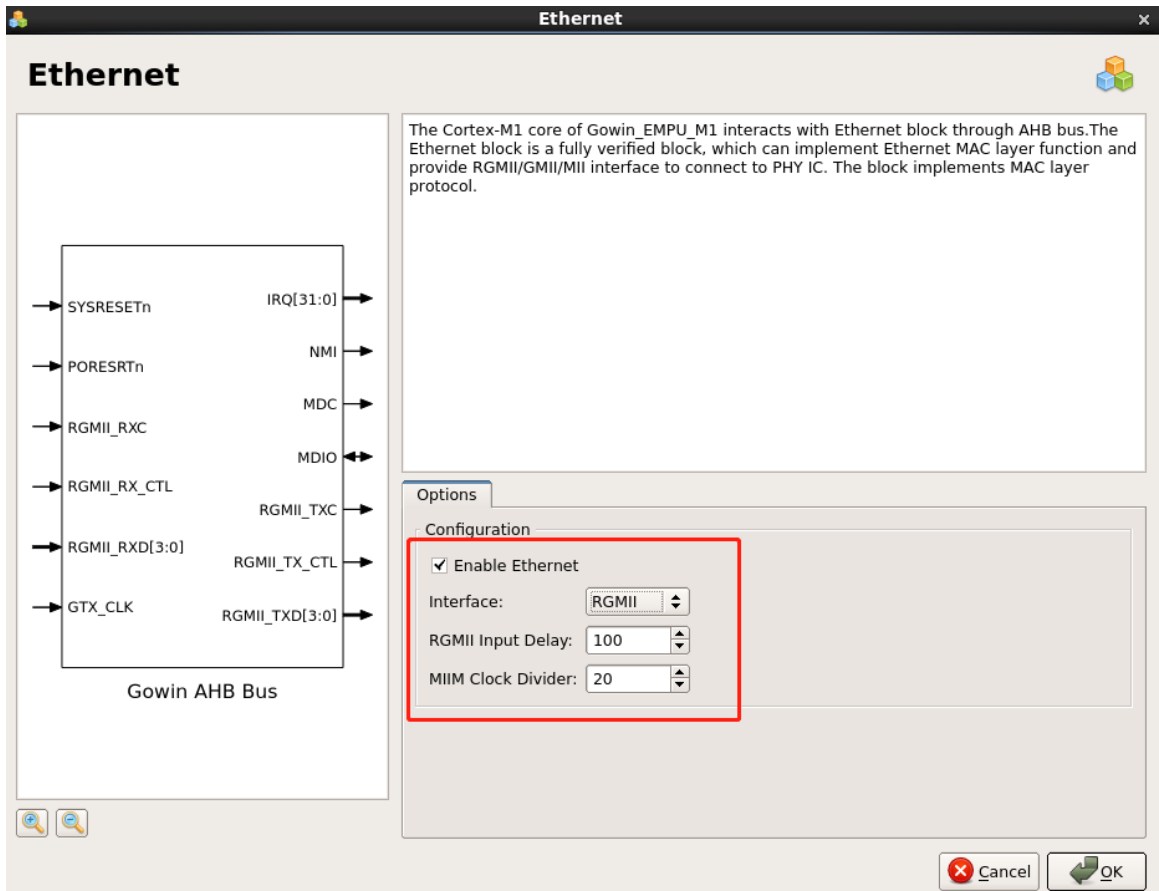
图 3-12 CAN 配置



### Ethernet 配置

- 双击打开 Ethernet，可以选择配置 Ethernet，如图 3-13 所示。
- 默认关闭，如果选择 Enable Ethernet，则 Gowin\_EMPU\_M1 支持 Ethernet。
- 如果已经选择 Enable Ethernet，则可以配置 Interface、RGMII Input Delay、MIIM Clock Divider。
  - 选择 Interface，可以选择配置 RGMII、GMII、MII 或 GMII/MII，默认为 RGMII。
  - 如果选择 Interface 为 RGMII，则可以选择配置 RGMII Input Delay，默认值为 100。
  - 选择 MIIM Clock Divider，可以配置 MIIM Clock Divider，默认值为 20。
- 如果 Interface 选择 RGMII 或 GMII，则端口 GTX\_CLK 必须接入 125MHz 时钟输入。

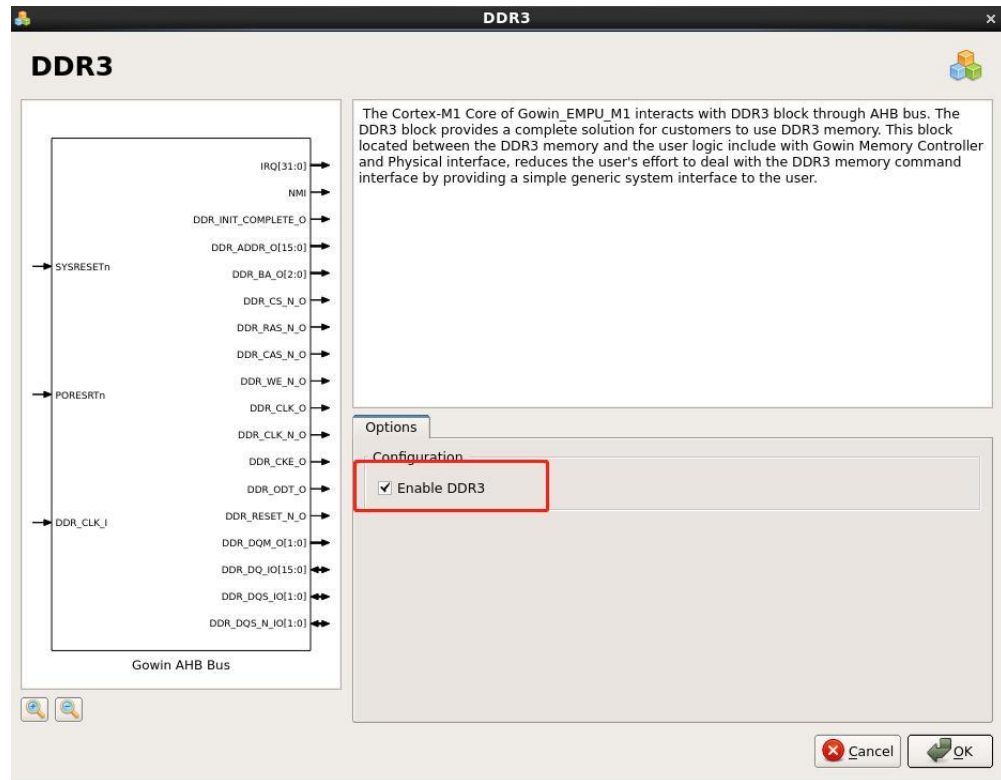
图 3-13 Ethernet 配置



### DDR3 配置

- 双击打开 DDR3，可以选择配置 DDR3，如图 3-14 所示。
- 默认关闭，如果选择 Enable DDR3，则 Gowin\_EMPU\_M1 支持 DDR3。
- DDR3 端口 DDR\_CLK\_I 必须接入 50MHz 时钟输入。

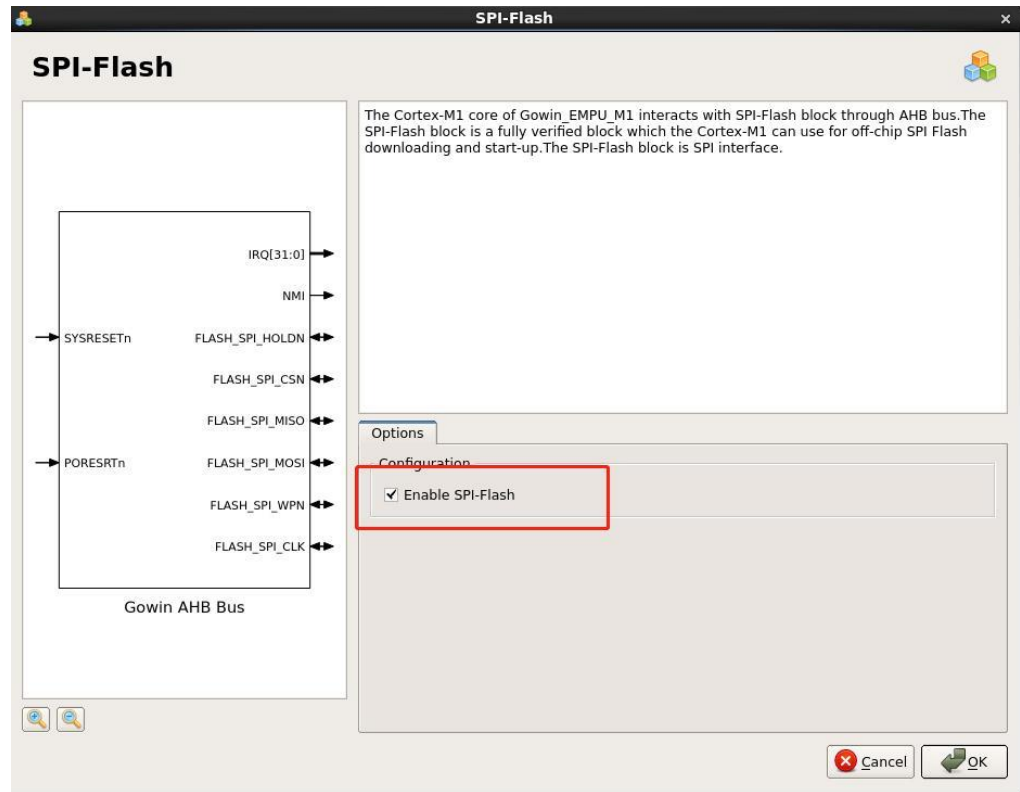
图 3-14 DDR3 配置



### SPI-Flash 配置

- SPI-Flash 支持下载功能，和 Memory 读、写、擦除功能。
- 双击打开 SPI-Flash，可以选择配置 SPI-Flash，如图 3-15 所示。
- 默认关闭，如果选择 Enable SPI-Flash，则 Gowin\_EMPU\_M1 支持 SPI-Flash。
- 如果 Gowin\_EMPU\_M1 使用片外 SPI-Flash 下载启动方式，则必须选择 Enable SPI-Flash。

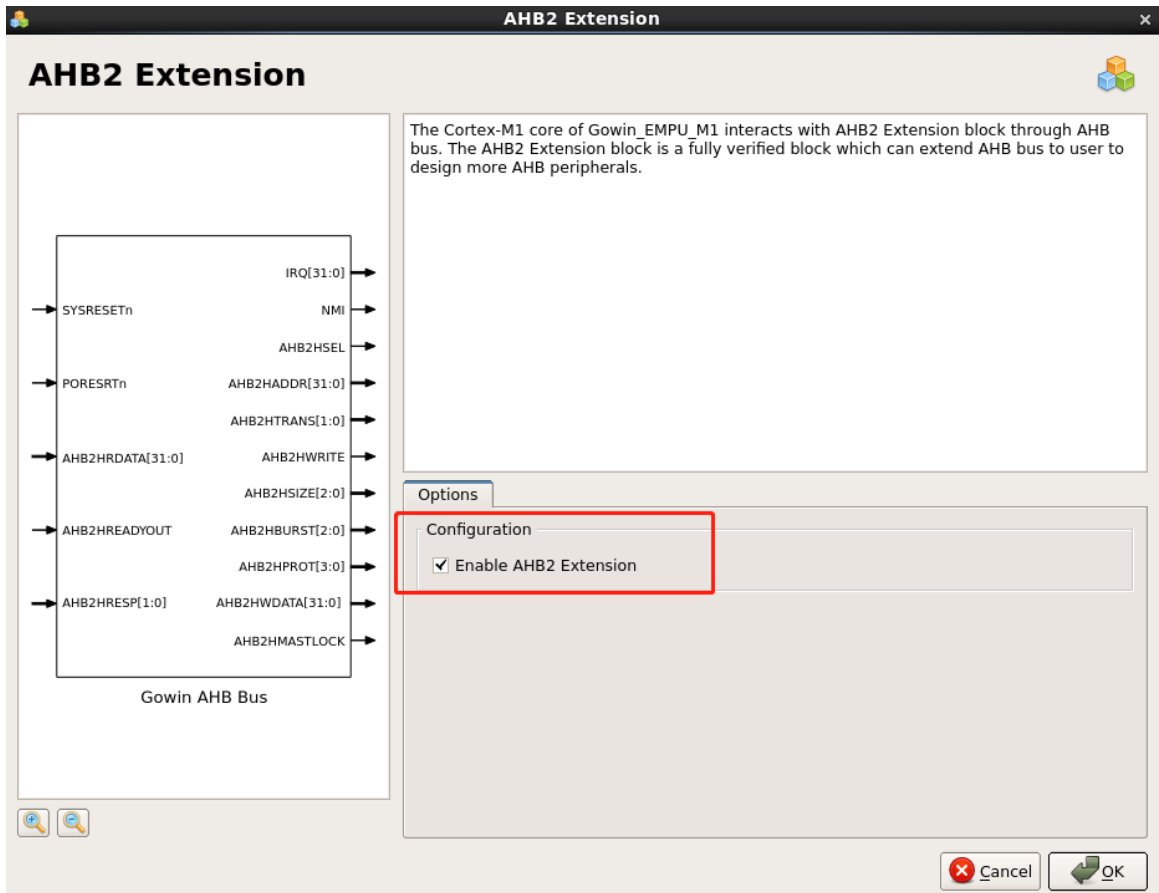
图 3-15 SPI-Flash 配置



### AHB2 Extension 配置

- 双击打开 AHB2 Extension，可以选择配置 AHB2 Extension，如图 3-16 所示。
- 默认关闭，如果选择 Enable AHB2 Extension，则 Gowin\_EMPU\_M1 支持 AHB2 Extension，用户可以在此接口自行连接扩展的 AHB 外部设备。

图 3-16 AHB2 Extension 配置

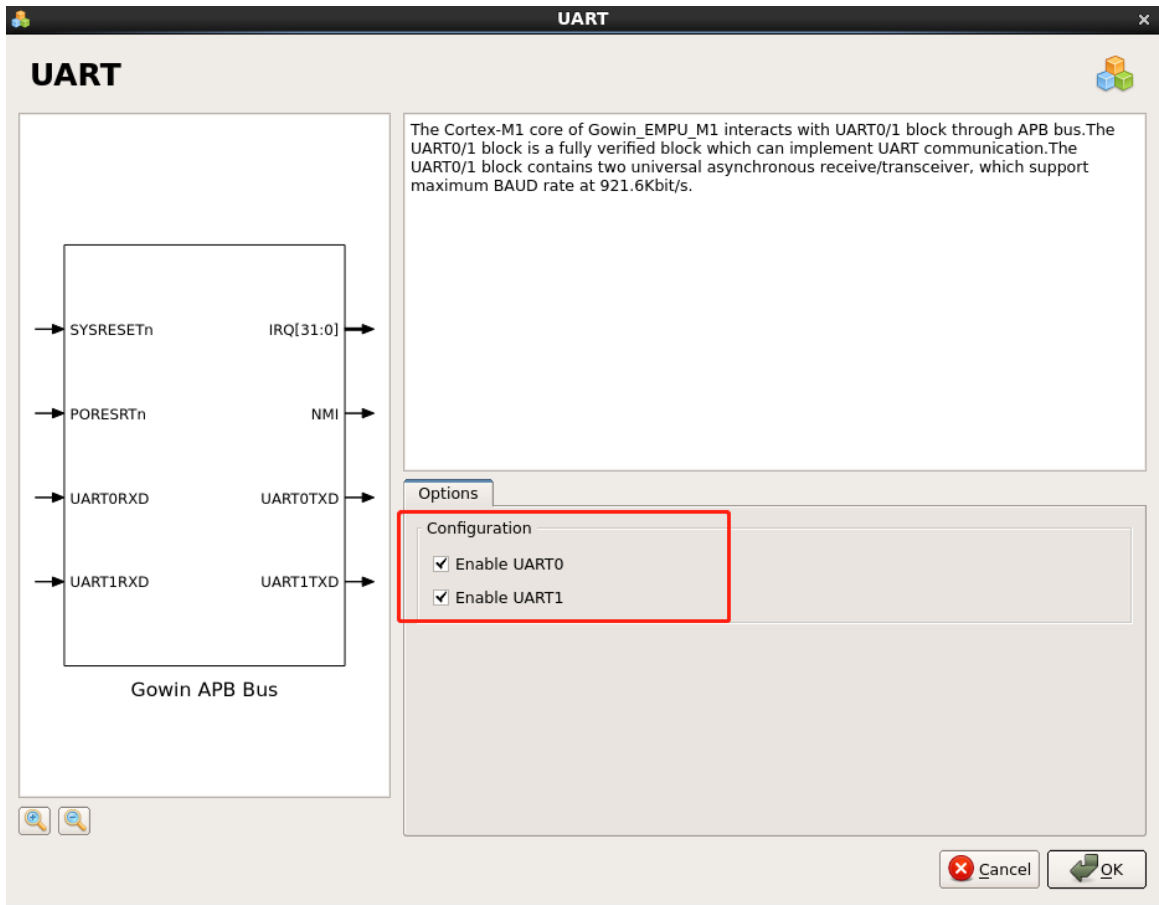


### UART 配置

- 双击打开 UART0 或 UART1，可以选择配置 UART0 或 UART1，如图 3-17 所示。
- 默认关闭，如果选择 Enable UART0，则 Gowin\_EMPU\_M1 支持 UART0。
- 默认关闭，如果选择 Enable UART1，则 Gowin\_EMPU\_M1 支持 UART1。



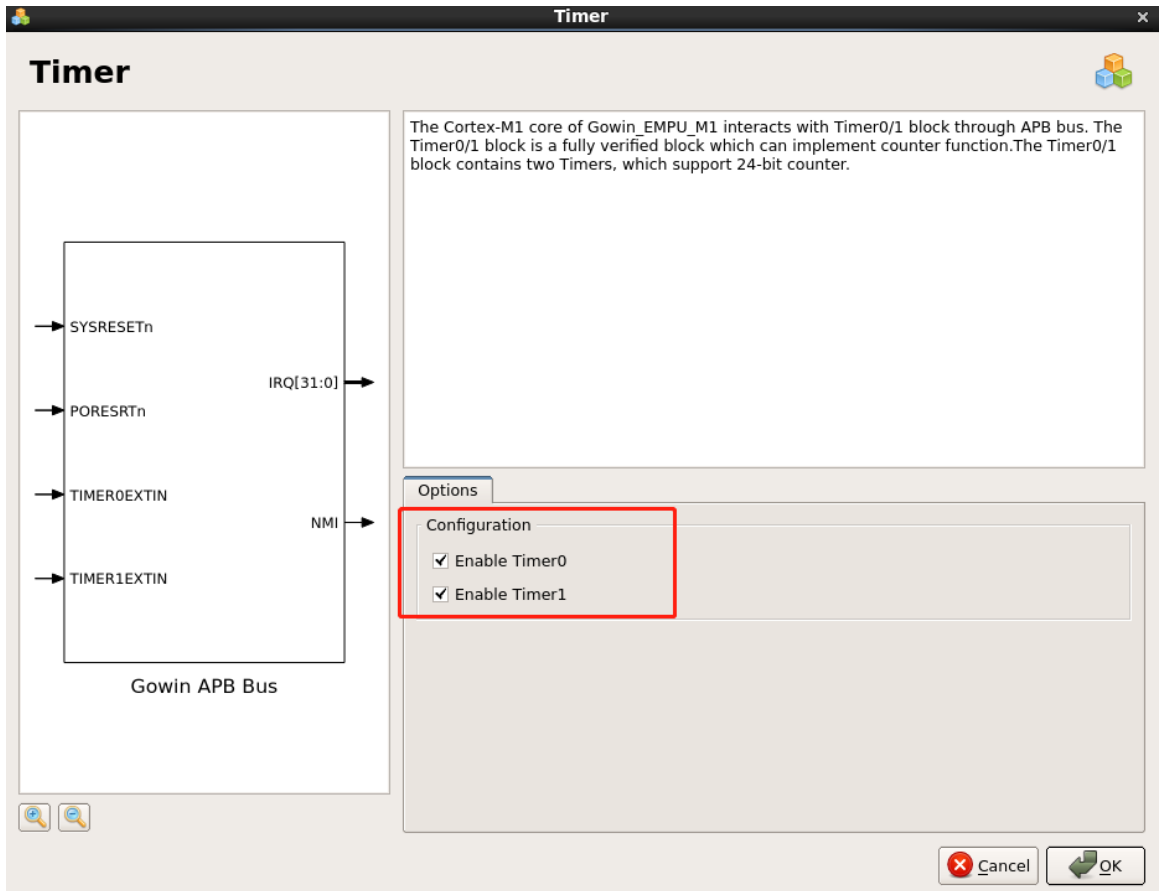
图 3-17 UART 配置



### Timer 配置

- 双击打开 Timer0 或 Timer1，可以选择配置 Timer0 或 Timer1，如图 3-18 所示。
- 默认关闭，如果选择 Enable Timer0，则 Gowin\_EMPU\_M1 支持 Timer0。
- 默认关闭，如果选择 Enable Timer1，则 Gowin\_EMPU\_M1 支持 Timer1。

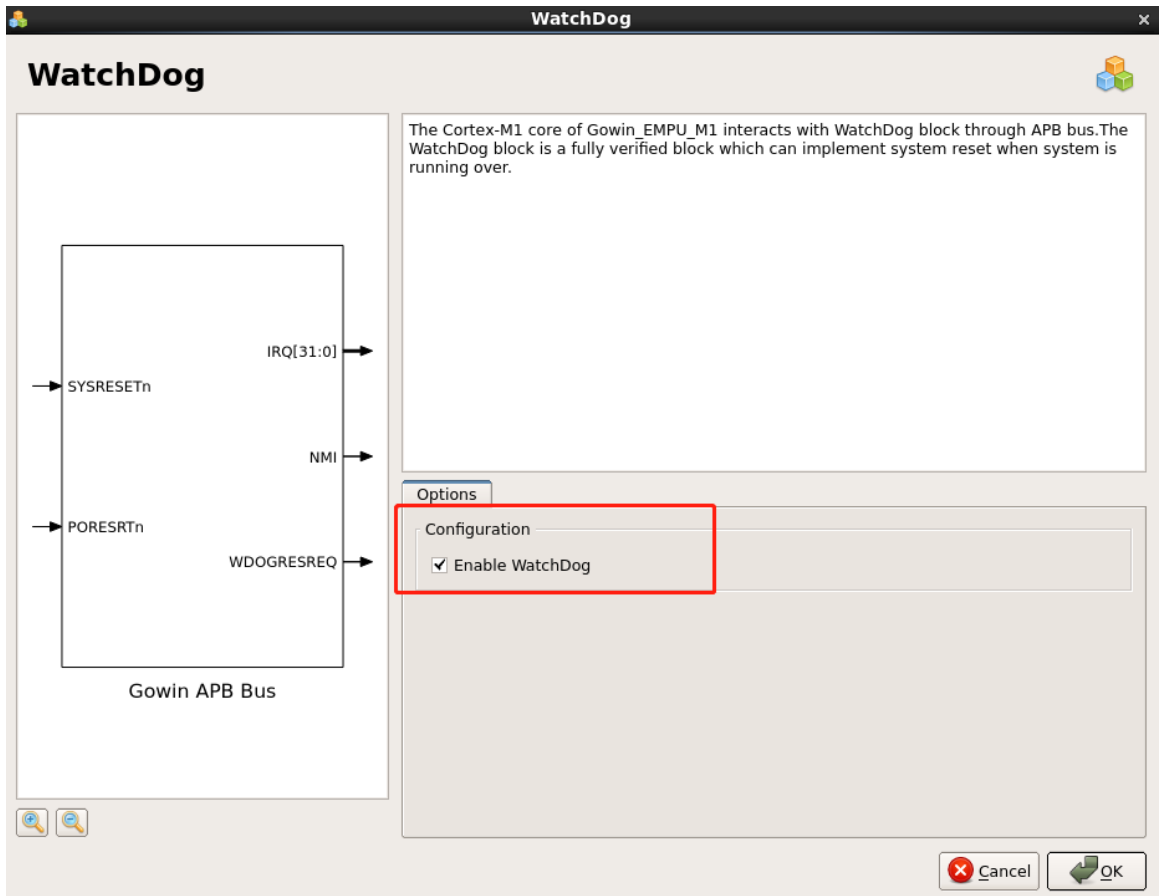
图 3-18 Timer 配置



### WatchDog 配置

- 双击打开 WatchDog，可以选择配置 WatchDog，如图 3-19 所示。
- 默认关闭，如果选择 Enable WatchDog，则 Gowin\_EMPU\_M1 支持 WatchDog。

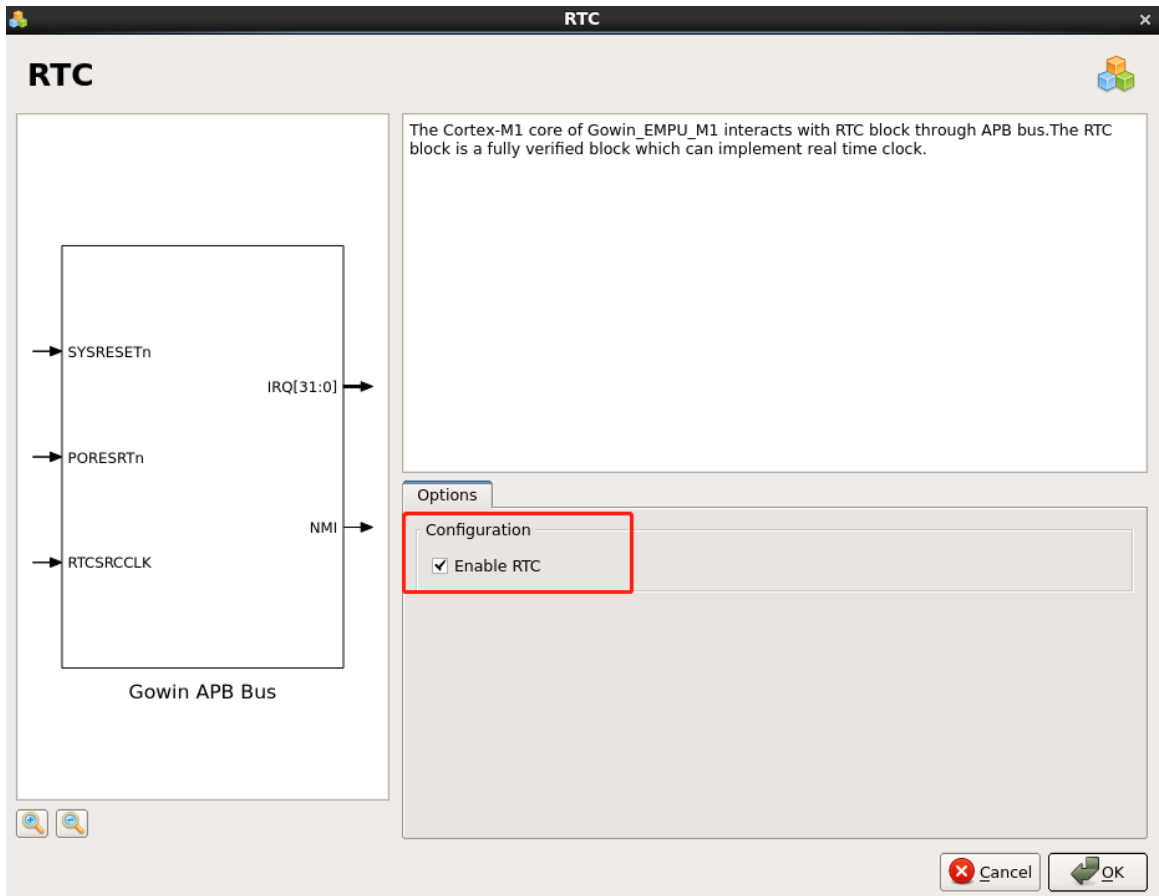
图 3-19 WatchDog 配置



### RTC 配置

- 双击打开 RTC，可以选择配置 RTC，如图 3-20 所示。
- 默认关闭，如果选择 Enable RTC，则 Gowin\_EMPU\_M1 支持 RTC。

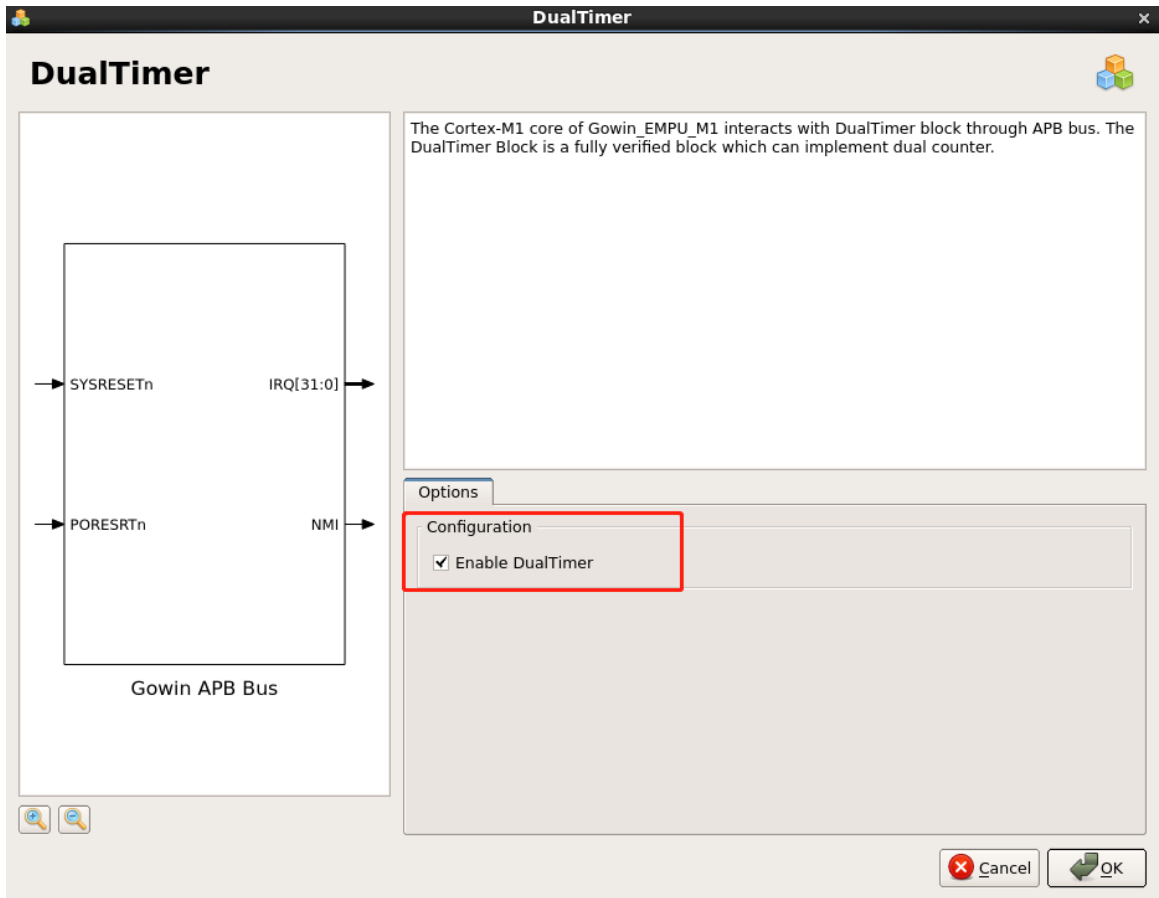
图 3-20 RTC 配置



### DualTimer 配置

- 双击打开 DualTimer，可以选择配置 DualTimer，如图 3-21 所示。
- 默认关闭，如果选择 Enable DualTimer，则 Gowin\_EMPU\_M1 支持 DualTimer。

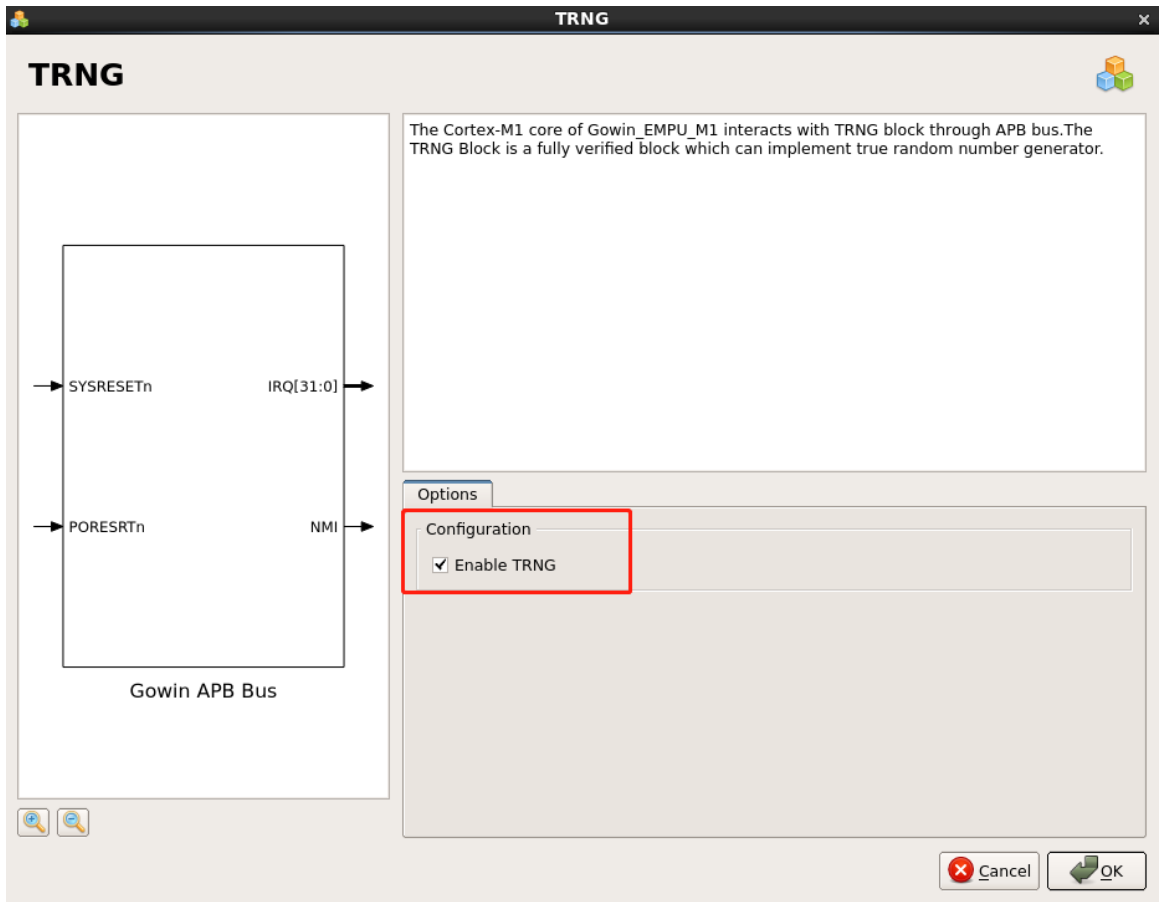
图 3-21 DualTimer 配置



### TRNG 配置

- 双击打开 TRNG，可以选择配置 TRNG，如图 3-22 所示。
- 默认关闭，如果选择 Enable TRNG，则 Gowin\_EMPU\_M1 支持 TRNG。

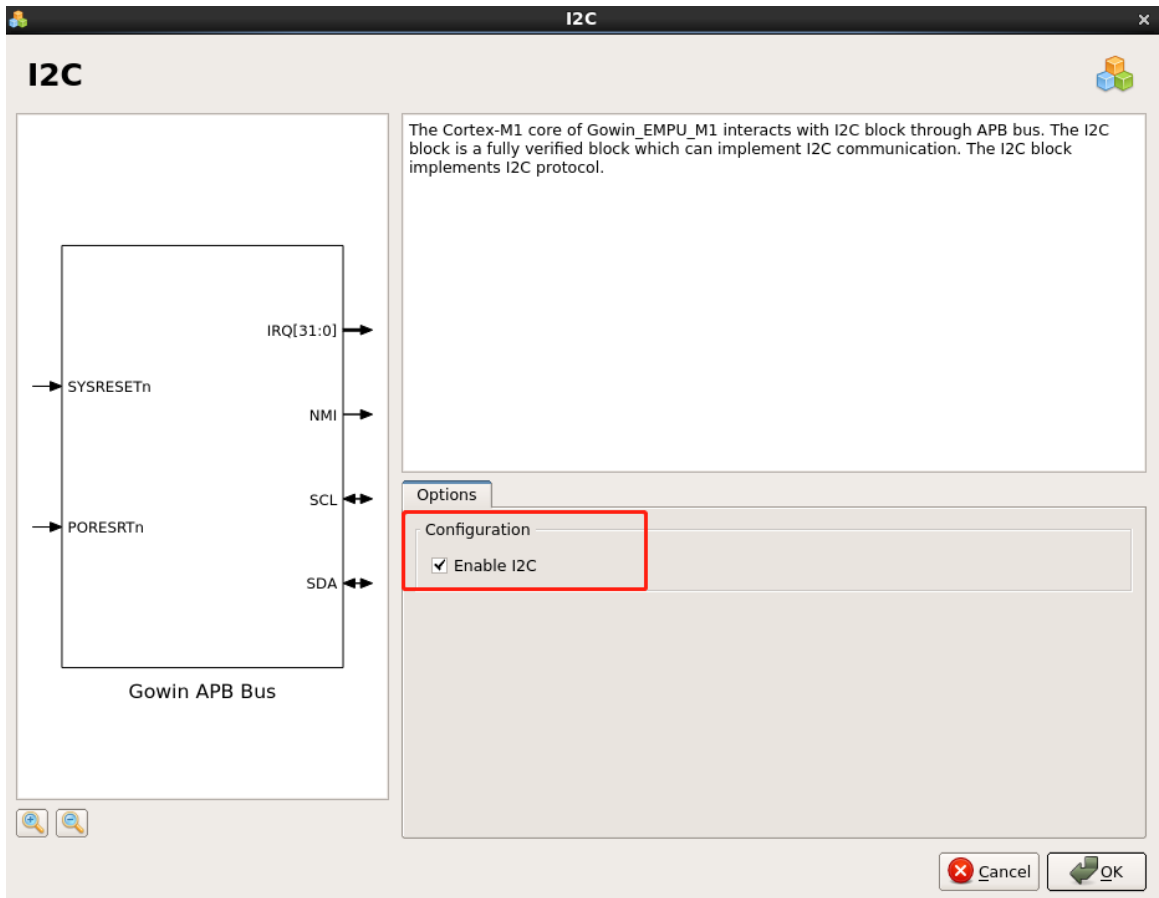
图 3-22 TRNG 配置



### I2C 配置

- 双击打开 I2C，可以选择配置 I2C，如图 3-23 所示。
- 默认关闭，如果选择 Enable I2C，则 Gowin\_EMPU\_M1 支持 I2C。

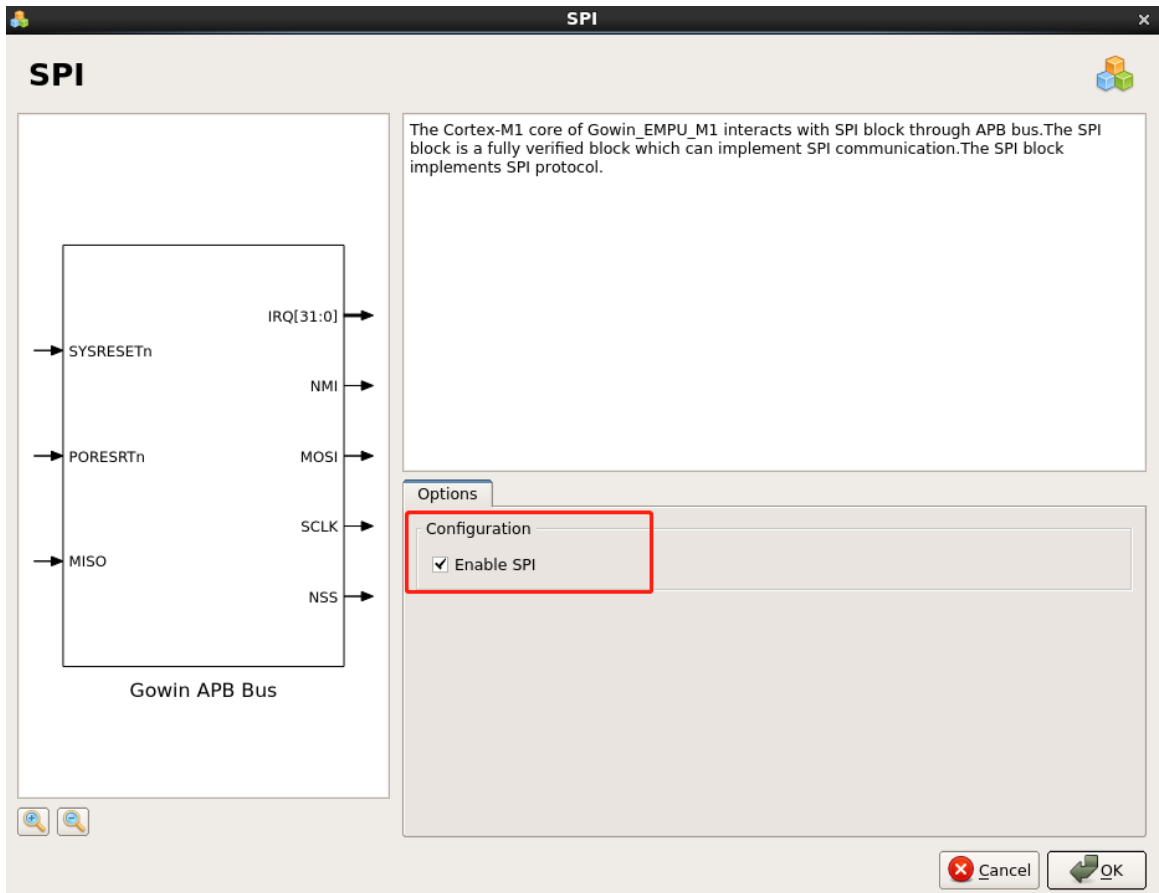
图 3-23 I2C 配置



### SPI 配置

- 双击打开 SPI，可以选择配置 SPI，如图 3-24 所示。
- 默认关闭，如果选择 Enable SPI，则 Gowin\_EMPU\_M1 支持 SPI。

图 3-24 SPI 配置

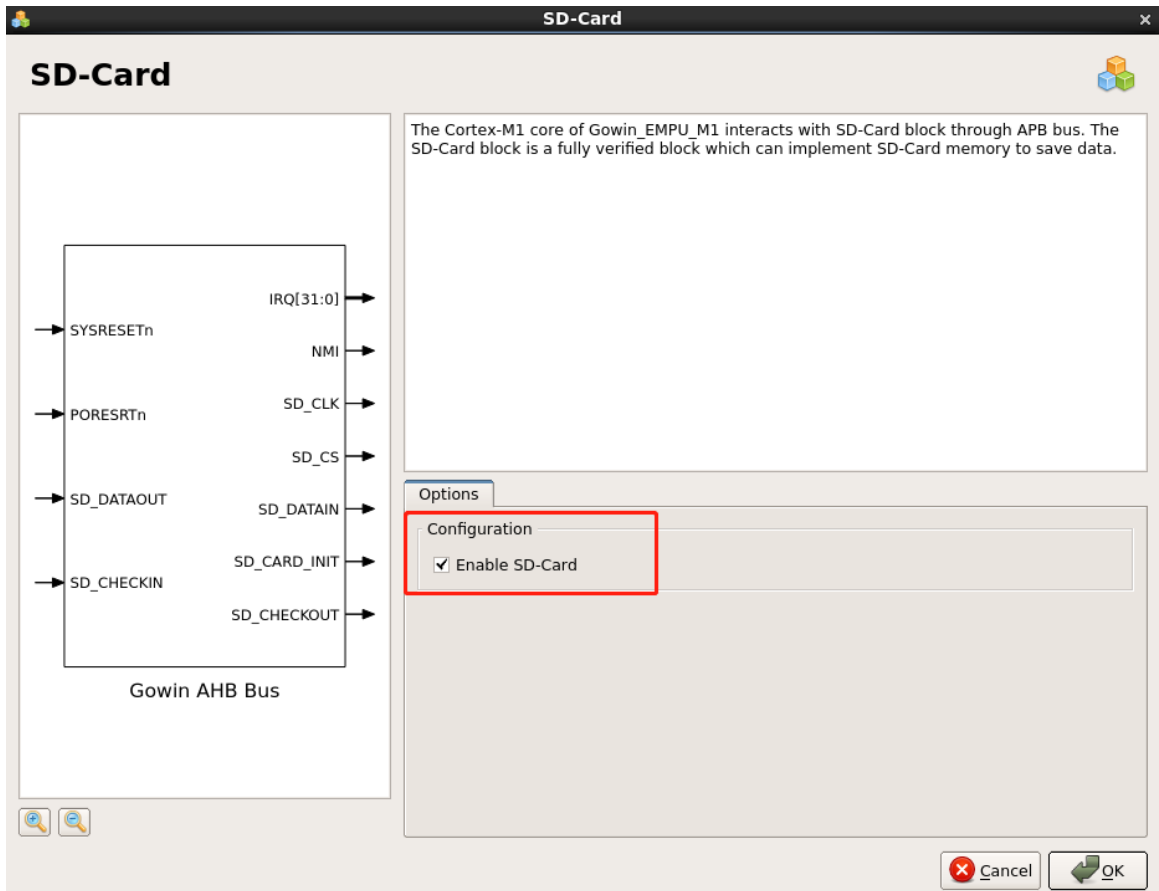


### SD-Card 配置

- 双击打开 SD-Card，可以选择配置 SD-Card，如图 3-25 所示。
- 默认关闭，如果选择 Enable SD-Card，则 Gowin\_EMPU\_M1 支持 SD-Card。
- 此版本 SD-Card 属性为只读 SD-Card。



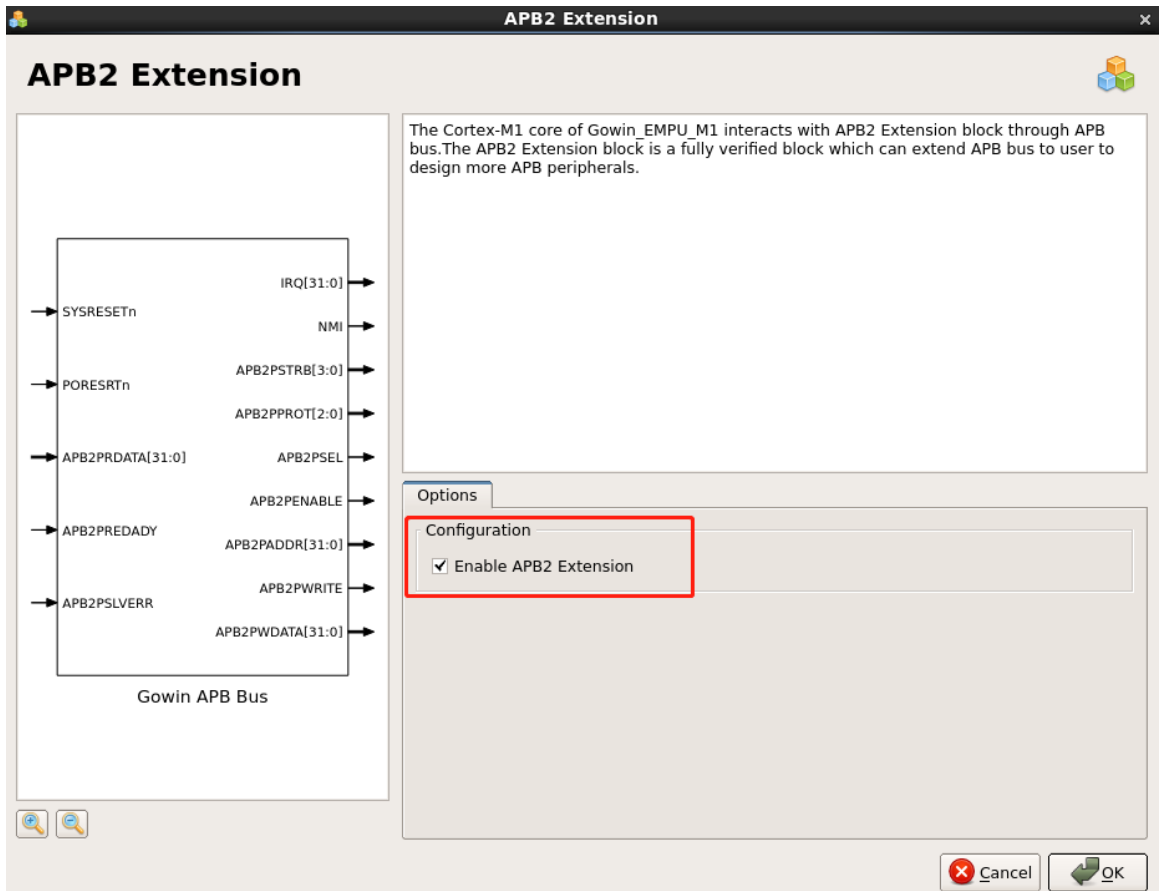
图 3-25 SD-Card 配置



### APB2 Extension 配置

- 双击打开 APB2 Extension，可以选择配置 APB2 Extension，如图 3-26 所示。
- 默认关闭，如果选择 Enable APB2 Extension，则 Gowin\_EMPU\_M1 支持 APB2 Extension，用户可以在此接口自行连接扩展的 APB 外部设备。

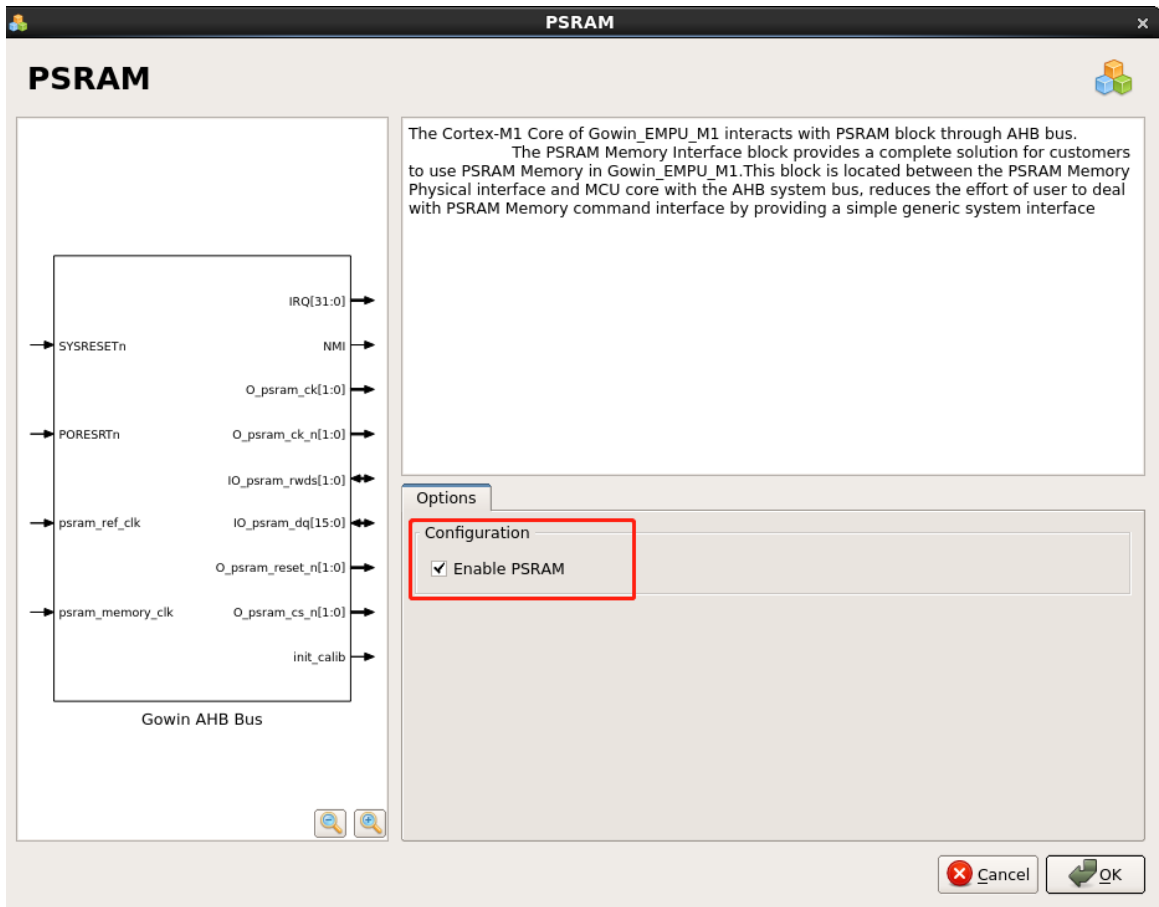
图 3-26 APB2 Extension 配置



### PSRAM 配置

- 双击打开 PSRAM，可以选择配置 PSRAM，如图 3-27 所示。
- 默认关闭，如果选择 Enable PSRAM，则 Gowin\_EMPU\_M1 支持 PSRAM。
- 如下器件支持 Gowin\_EMPU\_M1 PSRAM：
  - GW2AR-LV18QN88PES
  - GW2AR-LV18QN88PC8/I7
  - GW2AR-LV18QN88PC7/I6
  - GW2AR-LV18EQ144PES
  - GW2AR-LV18EQ144PC8/I7
  - GW2AR-LV18EQ144PC7/I6
  - GW2AR-LV18EQ144PC9/I8

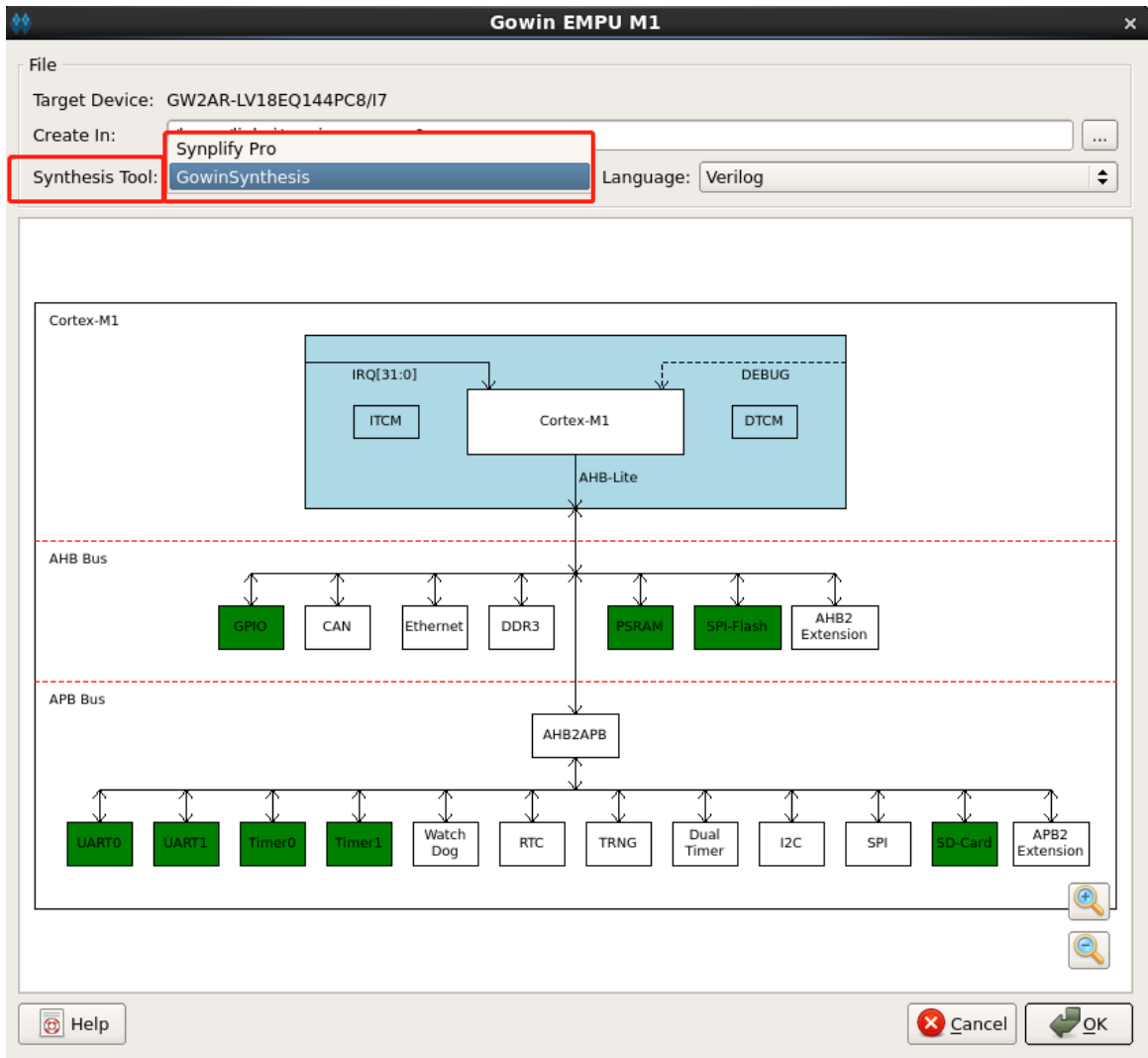
图 3-27 PSRAM 配置



### 综合工具配置

完成 Gowin\_EMPU\_M1 功能配置后, 选择产生 Gowin\_EMPU\_M1 的综合工具 Synplify Pro 或 GowinSynthesis, 如图 3-30 所示。

图 3-28 综合工具配置



### 3.3 用户设计

- 完成 Gowin\_EMPU\_M1 配置后，产生 Gowin\_EMPU\_M1 硬件设计
- 实例化 Gowin\_EMPU\_M1
- 导入用户设计，连接 Gowin\_EMPU\_M1，形成完整的 RTL 设计

### 3.4 约束

完成用户 RTL 设计后，根据使用的开发板和需要输出的 IO，产生物理约束文件。

根据时序要求，产生时序约束文件。

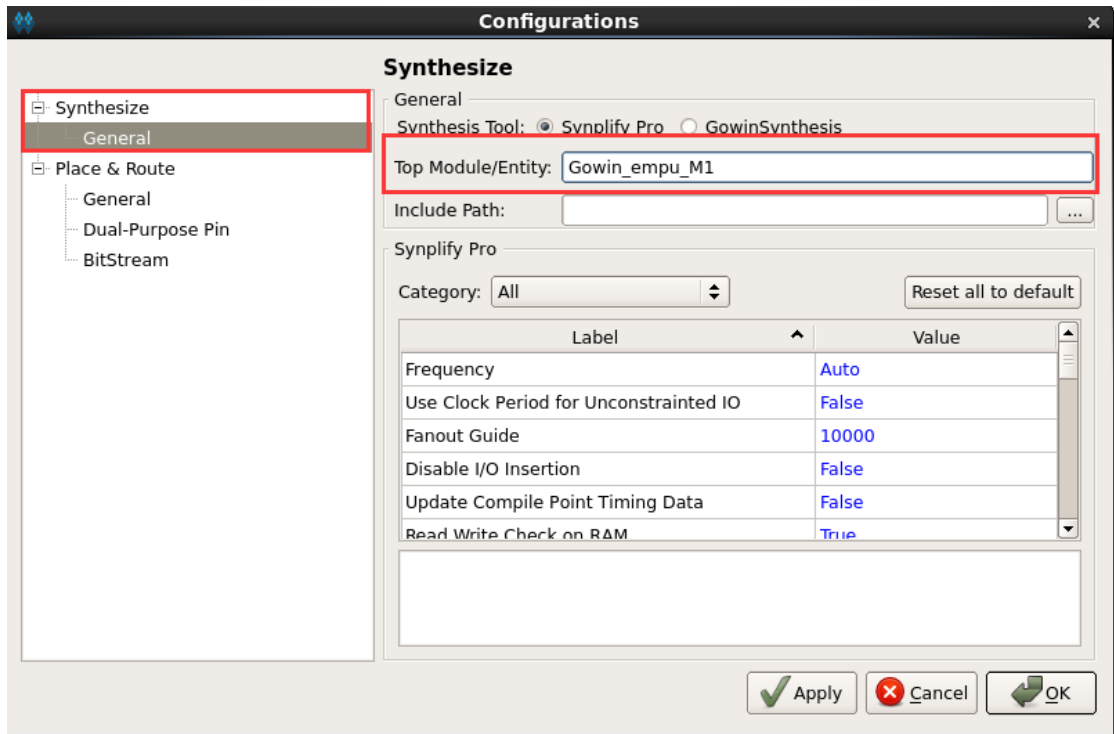
物理约束的产生方法请参考 [SUG101](#)，Gowin 设计约束指南。

## 3.5 配置

### 3.5.1 顶层模块配置

综合配置中，根据工程设计中的实际顶层模块名称配置此处顶层模块名称，如图 3-29 所示。

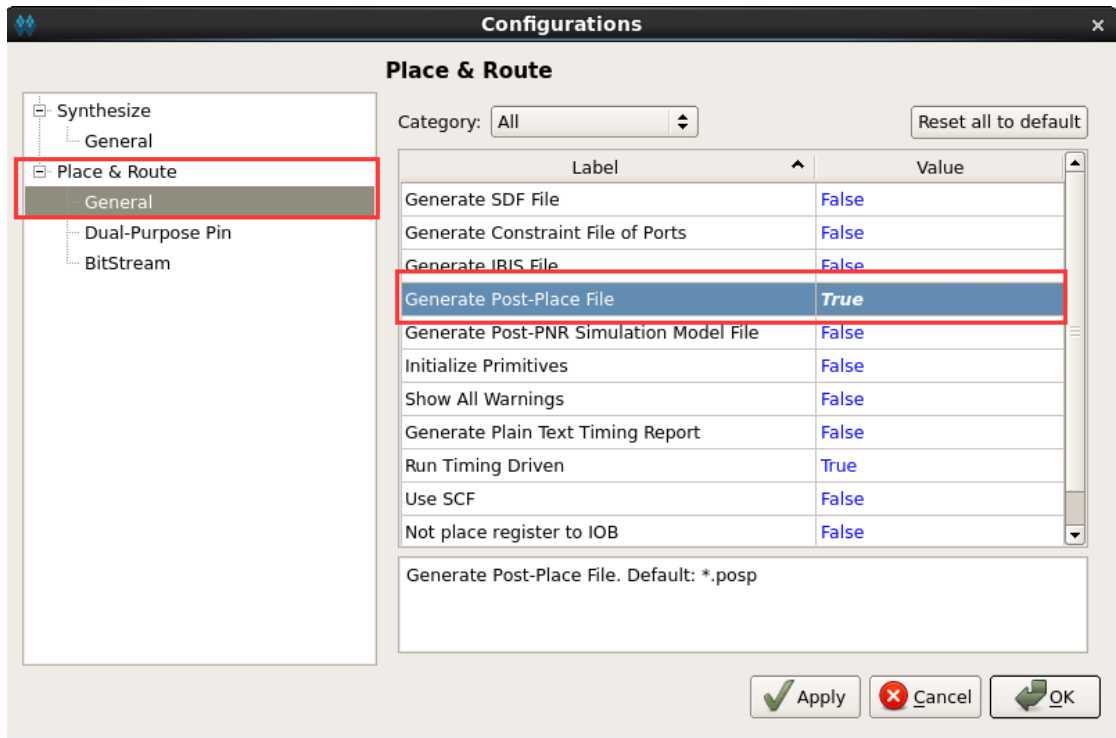
图 3-29 顶层模块配置



### 3.5.2 Post-Place File 配置

如果使用 Gowin\_EMPU\_M1 软件编程设计和硬件设计自动化合并的下载方法，则配置 Place & Route 产生 Post-Place File，如图 3-30 所示，否则不需要配置输出 Post-Place File。

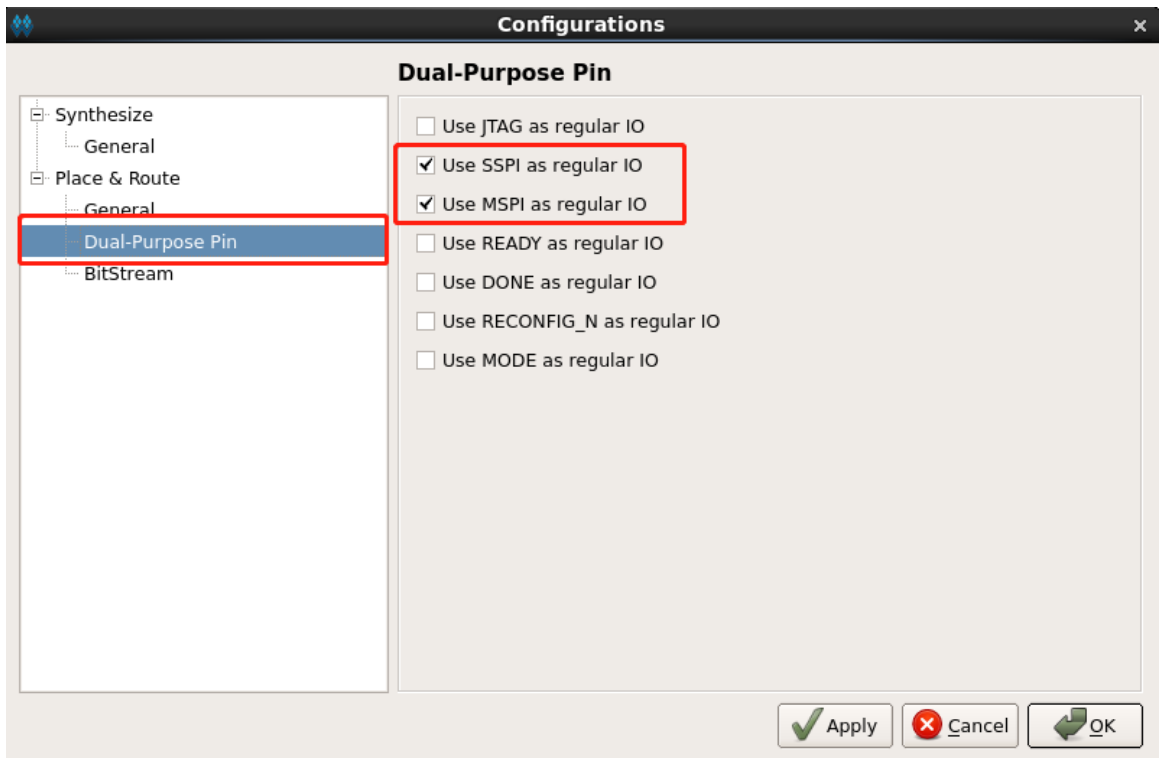
图 3-30 Post-Place File 配置



### 3.5.3 Dual-Purpose Pin 配置

如果 Gowin\_EMPU\_M1 使用片外 SPI-Flash 下载启动方式，复用 SSPI 和 MSPI 端口为通用端口，如图 3-31 所示，否则不需要配置端口复用。

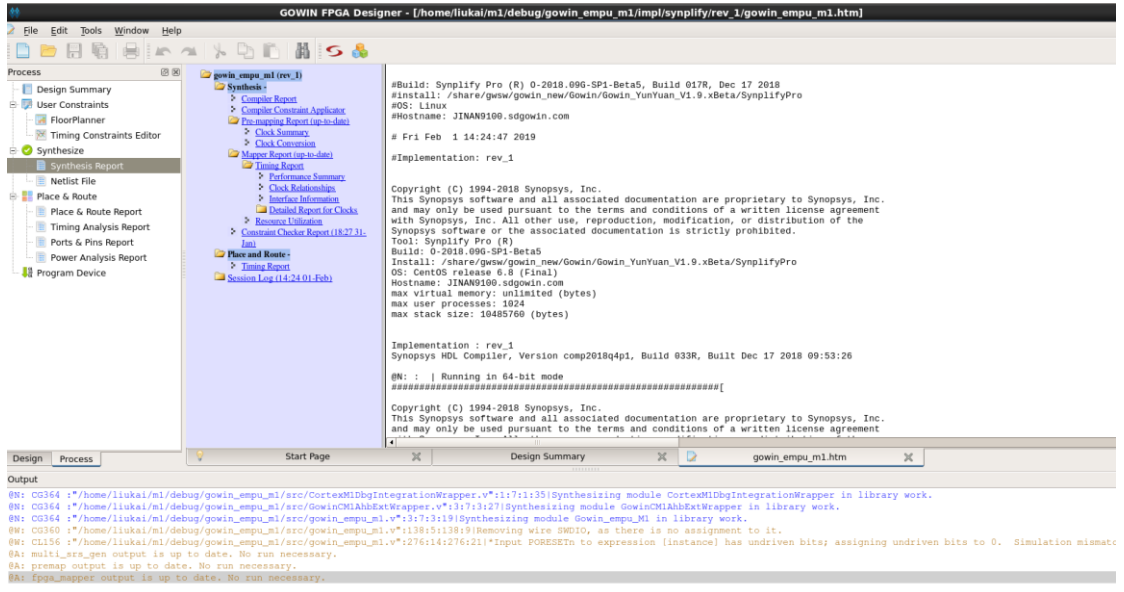
图 3-31 Dual-Purpose Pin 配置



### 3.6 综合

运行 Gowin 云源软件的综合工具 Synplify\_Pro 或 GowinSynthesis，完成 RTL 设计的综合，如图 3-32 图所示。

图 3-32 综合

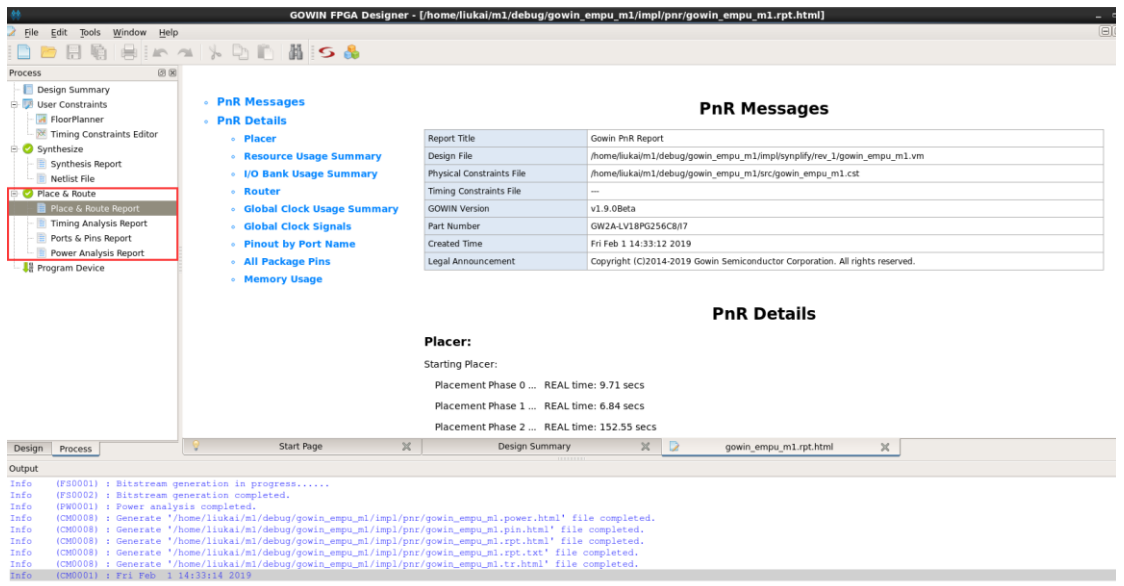


综合工具的使用方法请参考 [SUG100](#)，Gowin 云源软件用户指南。

### 3.7 布局布线

运行 Gowin 云源软件的布局布线工具 Place & Route，完成布局布线和生成码流文件，如图 3-33 所示。


图 3-33 Place & Route



布局布线工具使用方法请参考 [SUG100](#), Gowin 云源软件用户指南。

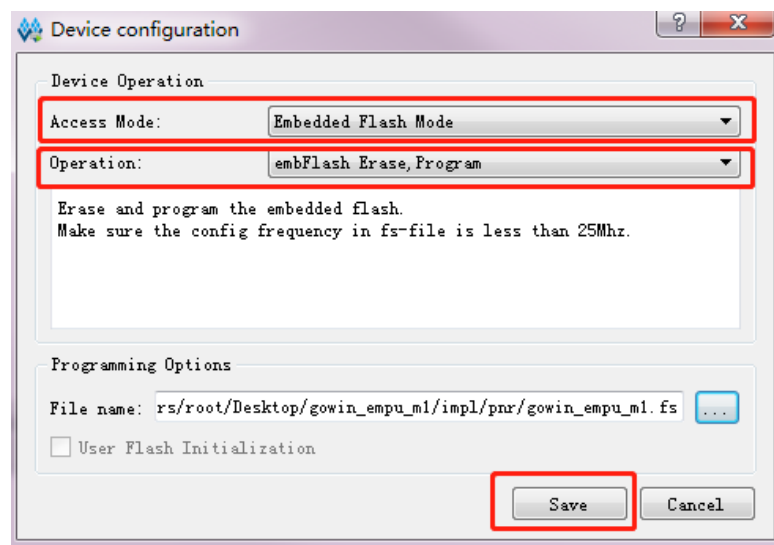
## 3.8 下载

运行 Gowin 云源软件的下载工具 Programmer, 完成码流文件的下载。

单击 “Programmer > Edit/Configure Device” 或工具栏 “Configure Device ”, 打开 “Device configuration” 对话框。

如果板载器件 GW1N-9/GW1NR-9 系列, “Access Mode” 下拉列表中选择 “Embedded Flash Mode”, “Operation” 下拉列表中选择 “embFlash Erase, Program”, “Programming Options” 中导入需要下载的码流文件, 单击 “Save”, 如图 3-34 所示。

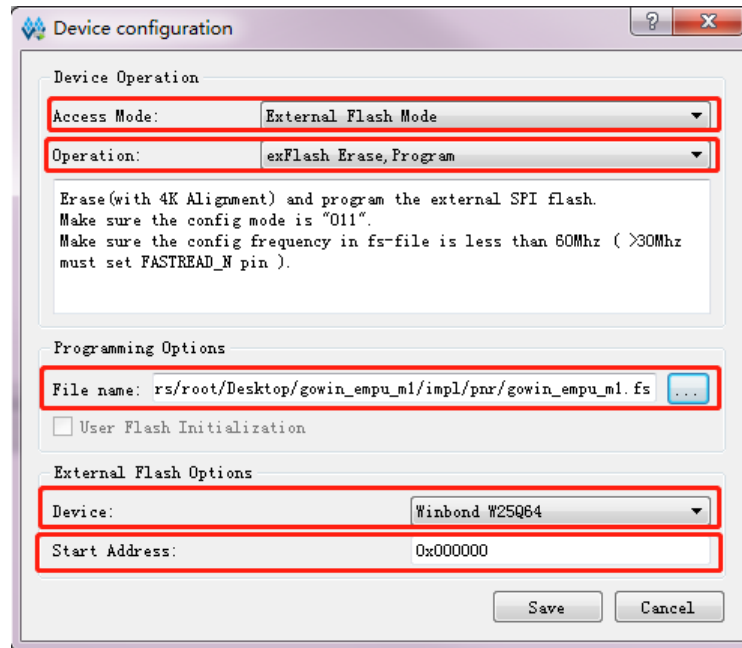
图 3-34 GW1N 系列 Configure Device




如果板载器件 GW2A-18/GW2AR-18/GW2A-55, “Access Mode” 下拉列表中选择 “External Flash Mode”, “Operation” 下拉列表中选择 “exFlash Erase, Program”, “Programming Options” 中导入需要下载的码流文件, “External Flash Options” 中 “Device” 根据板载 Flash 芯片选择, “Start Address” 选择 “0x000000”, 单击 “Save”, 如图 3-35 所示。



图 3-35 GW2A 系列 Configure Device



完成 Device configuration 后，单击“Programmer”工具链 Program/Configure “”，完成码流下载。

下载工具 Programmer 使用方法请参考 [SUG502](#), Gowin Programmer 用户指南。

# 4 参考设计

Gowin\_EMPU\_M1 提供 Debug 和 No Debug 参考设计:

- Gowin\_EMPU\_M1\ref\_design\FPGA\_RefDesign\Debug\_RefDesign
- Gowin\_EMPU\_M1\ref\_design\FPGA\_RefDesign\NoDebug\_RefDesign

