



Gowin PicoRV32 硬件设计参考手册

IPUG914-1.0,2020-01-06

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
2020/01/06	1.0	初始版本。

目录

目录	i
图目录	iii
表目录	iv
1 硬件架构	1
1.1 系统架构	1
1.2 系统特征	2
1.2.1 PicoRV32 内核子系统	2
1.2.2 Wishbone Bus 子系统	3
1.3 系统端口定义	3
1.3.1 PicoRV32 内核子系统端口定义	3
1.3.2 Wishbone Bus 子系统端口定义	4
2 硬件设计流程	5
2.1 硬件环境	5
2.2 软件环境	5
2.3 软核生成器	5
2.4 下载软件	5
2.5 设计流程	5
3 工程模板	7
3.1 工程创建	7
3.1.1 新建工程	7
3.1.2 设定工程名称和路径	8
3.1.3 选择器件	9
3.1.4 完成工程创建	10
3.2 硬件设计	10
3.2.1 PicoRV32 内核子系统硬件设计	12
3.2.2 Wishbone Bus 子系统硬件设计	17
3.3 用户设计	24
3.4 约束	24

3.5 工程配置.....	24
3.5.1 综合选项配置	24
3.5.2 Post-Place File 生成配置	26
3.5.3 Dual-Purpose Pin 配置	26
3.6 综合	27
3.7 布局布线.....	28
3.8 下载.....	28
4 参考设计	30

图目录

图 1-1 Gowin_PicoRV32 系统架构.....	1
图 3-1 新建 FPGA Design 工程	7
图 3-2 设定工程名称和路径	8
图 3-3 选择器件	9
图 3-4 完成工程创建	10
图 3-5 选择 Gowin_PicoRV32	11
图 3-6 Gowin_PicoRV32 系统架构.....	12
图 3-7 Gowin PicoRV32 CORE 配置选项	13
图 3-8 指令存储器 ILM 配置选项	14
图 3-9 数据存储器 DLM 配置选项.....	15
图 3-10 Simple UART 配置选项	16
图 3-11 WB UART 配置选项.....	17
图 3-12 WB I2C Master 配置选项.....	18
图 3-13 WB SPI Master 配置.....	19
图 3-14 WB SPI Slave 配置.....	20
图 3-15 OPEN WB INTERFACE 配置	22
图 3-16 Top Module 配置.....	23
图 3-17 综合工具配置	24
图 3-18 综合选项配置	25
图 3-19 Post-Place File 配置	26
图 3-20 Dual-Purpose Pin 配置	27
图 3-21 综合	27
图 3-22 Place & Route	28
图 3-23 GW1N 系列 Configure Device	29
图 3-24 GW2A 系列 Configure Device	29

表目录

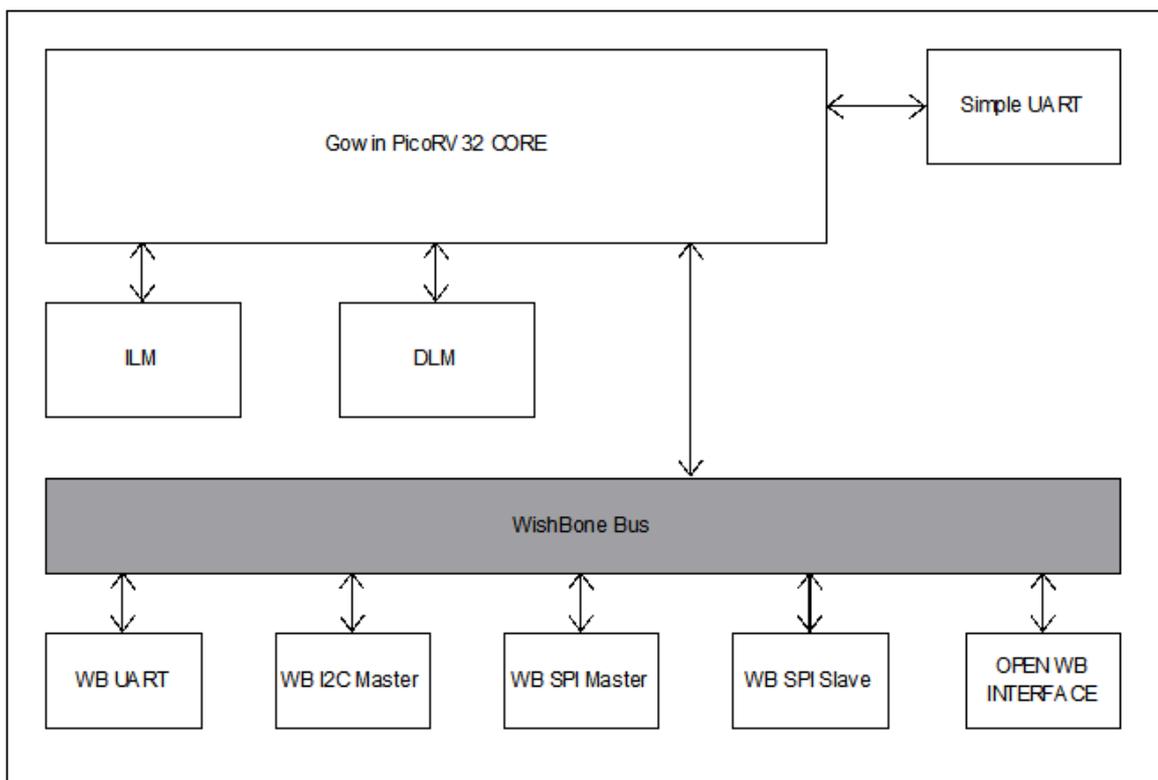
表 1-1 PicoRV32 内核子系统端口定义	3
表 1-2 Wishbone Bus 子系统端口定义	4
表 3-1 Gowin PicoRV32 CORE 配置选项	13
表 3-2 Wishbone Bus 子系统配置选项	17
表 3-3 WB SPI Master 参数配置选项	19
表 3-4 WB SPI Slave 参数配置选项	20

1 硬件架构

1.1 系统架构

Gowin_PicoRV32 的结构包括 PicoRV32 内核、指令存储器和数据存储器、轻量化 UART 和 Wishbone 总线外部设备，如图 1-1 所示。

图 1-1 Gowin_PicoRV32 系统架构



- Gowin PicoRV32 CORE 是 RISC-V 架构微控制器单元内核
- ILM 是指令存储器，DLM 是数据存储器
- Simple UART 是一个可配置的轻量化 UART
- Wishbone Bus 连接 PicoRV32 Core 及 Wishbone 总线接口外部设备，Wishbone Bus 接口外部设备包括 UART、I2C Master、SPI Master、SPI Slave 和 Wishbone 总线扩展接口

1.2 系统特征

Gowin_PicoRV32 包括两个子系统:

- PicoRV32 内核子系统, 包括微控制器单元内核、指令存储器、数据存储器和一个可配置的轻量化 UART
- Wishbone 总线, 以及 Wishbone 总线接口的外部设备子系统

1.2.1 PicoRV32 内核子系统

处理器内核

- RISC-V 32 位整型指令集架构
- 可配置的 RISC-V32M 乘法/除法指令集扩展和可配置的 RISC-V32C 压缩指令集扩展
- 可配置的 SPI FLASH 接口, 支持片外 SPI FLASH 下载启动方法
- 内置中断处理器模块, 采用自定义的中断管理指令, 支持 32 个中断源管理, 中断优先级可以通过软件控制
- 内置 32 位定时器模块, 采用自定义的定时器操作指令
- 大小端格式: RISC-V 标准仅支持小端格式

Memory

- ILM: 指令存储器, 可配置 Size, 小端模式
- DLM: 数据存储器, 可配置 Size, 小端模式

Simple UART

- 轻量化 UART 串行通信接口
- 使用极少逻辑资源

1.2.2 Wishbone Bus 子系统

Wishbone Bus 子系统包括 UART、I2C Master、SPI Master、SPI Slave、和 Wishbone 总线扩展接口。

1.3 系统端口定义

1.3.1 PicoRV32 内核子系统端口定义

PicoRV32 内核子系统端口定义，如表 1-1 所示。

表 1-1 PicoRV32 内核子系统端口定义

名称	I/O	位宽	描述
clk	in	1	系统时钟信号
resetn	in	1	系统复位信号
trap	out	1	运行异常指示信号
mem_valid	out	1	内核 valid 信号
mem_instr	out	1	指令标志信号，表示要读取的是指令数据
mem_addr	out	32	读/写地址信号
mem_wdata	out	32	写数据信号
mem_wstrb	out	4	读/写字节有效信号
mem_rdata	in	32	读数据信号
irq_mask_o	out	32	中断屏蔽信号
irq	in	32	中断输入信号
eoi	out	32	中断状态信号
trace_valid	out	1	trace 可用信号
trace_data	out	36	trace 数据信号
spimem_csb	out	1	SPI FLASH 的 SPI 片选信号
spimem_clk	out	1	SPI FLASH 的 SPI 时钟信号
spimem_io0	inout	1	SPI FLASH 的 MOSI 信号
spimem_io1	inout	1	SPI FLASH 的 MISO 信号
spimem_io2	inout	1	SPI FLASH 的 WPN 信号
spimem_io3	inout	1	SPI FLASH 的 HOLDN 信号
ser_tx	out	1	Simple UART 的输出信号
ser_rx	in	1	Simple UART 的输入信号

1.3.2 Wishbone Bus 子系统端口定义

Wishbone Bus 子系统端口定义，如表 1-2 所示。

表 1-2 Wishbone Bus 子系统端口定义

名称	I/O	位宽	描述	所属模块
clk_in	in	1	系统时钟信号	-
resetn_in	in	1	系统复位信号	-
irq_in	in	12	外部中断输入信号	-
wbuart_tx	out	1	Wishbone UART 的输出信号	WB UART
wbuart_rx	in	1	Wishbone UART 的输入信号	
wbi2c_sda	inout	1	Wishbone I2C Master 的数据信号	WB I2C
wbi2c_scl	inout	1	Wishbone I2C Mashter 的时钟信号	
wbspi_master_miso	in	1	Wishbone SPI Master 的 MISO 信号	WB SPI Master
wbspi_master_mosi	out	1	Wishbone SPI Master 的 MOSI 信号	
wbspi_master_ssn	out		Wishbone SPI Master 的 SLAVE 选定信号 每个 SLAVE 对应 1 位，最多支持 8 位	
wbspi_master_sclk	out	1	Wishbone SPI Master 的时钟信号	
wbspi_slave_miso	out	1	Wishbone SPI Slave 的 MISO 信号	WB SPI Slave
wbspi_slave_mosi	in	1	Wishbone SPI Slave 的 MOSI 信号	
wbspi_slave_ssn	in	1	Wishbone SPI Slave 的 SLAVE 选定信号	
wbspi_slave_sclk	in	1	Wishbone SPI Slave 的时钟信号	
slv_ext_stb_o	out	1	Wishbone 总线扩展接口的 strb 信号	OPEN WB INTERFACE
slv_ext_we_o	out	1	Wishbone 总线扩展接口的写操作信号	
slv_ext_cyc_o	out	1	Wishbone 总线扩展接口的 cyc 信号	
slv_ext_ack_i	in	1	Wishbone 总线扩展接口的 ack 信号	
slv_ext_adr_o	out	32	Wishbone 总线扩展接口的地址信号	
slv_ext_wdata_o	out	32	Wishbone 总线扩展接口的写数据信号	
slv_ext_rdata_i	in	32	Wishbone 总线扩展接口的读数据信号	
slv_ext_sel_o	out	4	Wishbone 总线扩展接口的字节选择信号	

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 Gowin_PicoRV32 的硬件设计。

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_PicoRV32 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_PicoRV32 硬件设计流程：

1. 根据用户设计需求，在 IP Core Generator 软核生成器中配置 Gowin_PicoRV32（如需通过 Wishbone 总线扩展接口支持自定义外部设备，则选择使能 OPEN WB INTERFACE，禁用“Use Gowin PicoRV32 as top module”选项）；
2. 完成 Gowin_PicoRV32 功能配置后，产生 Gowin_PicoRV32 硬件设计；

3. 实例化 Gowin_PicoRV32，导入用户设计，连接用户设计与 Gowin_PicoRV32；
4. 物理约束和时序约束；
5. Synplify_Pro 或 GowinSynthesis 综合；
6. Place & Route 布局布线，产生码流文件；
7. Programmer 下载码流文件到 GW1N-9/GW1NR-9/GW2A-18/GW2AR-18 /GW2A-55。

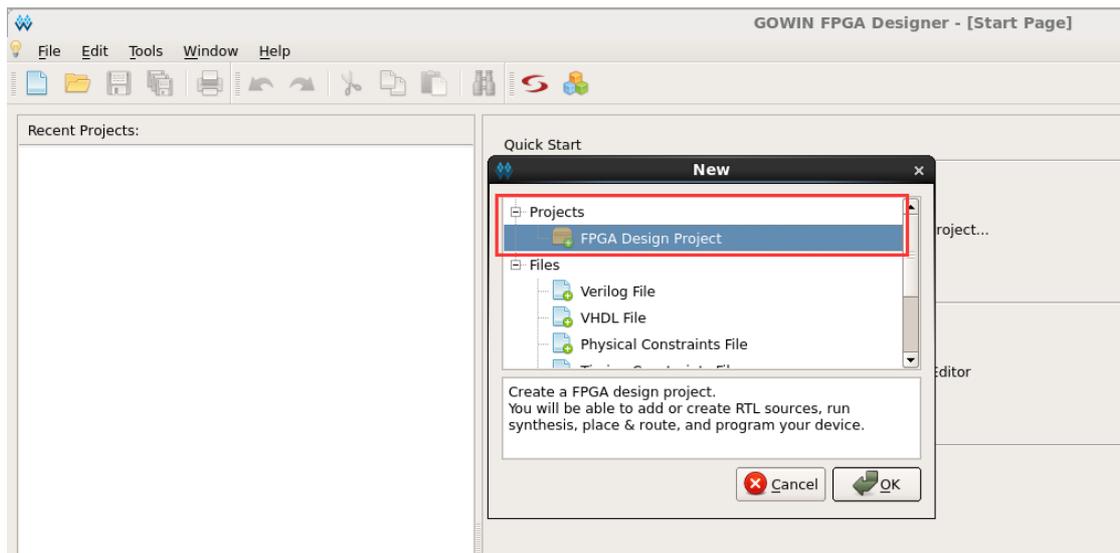
3 工程模板

3.1 工程创建

3.1.1 新建工程

双击 Gowin 云源软件，选择菜单栏“File”中的“New...”，选择 FPGA Design Project，如图 3-1 所示。

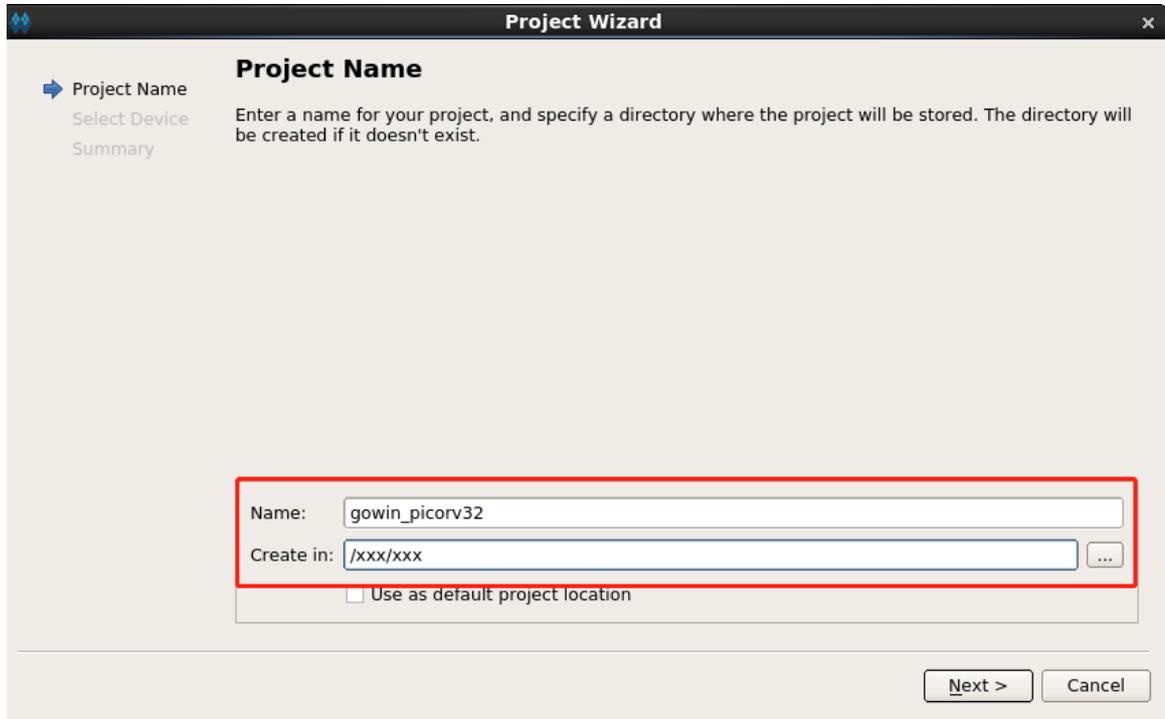
图 3-1 新建 FPGA Design 工程



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

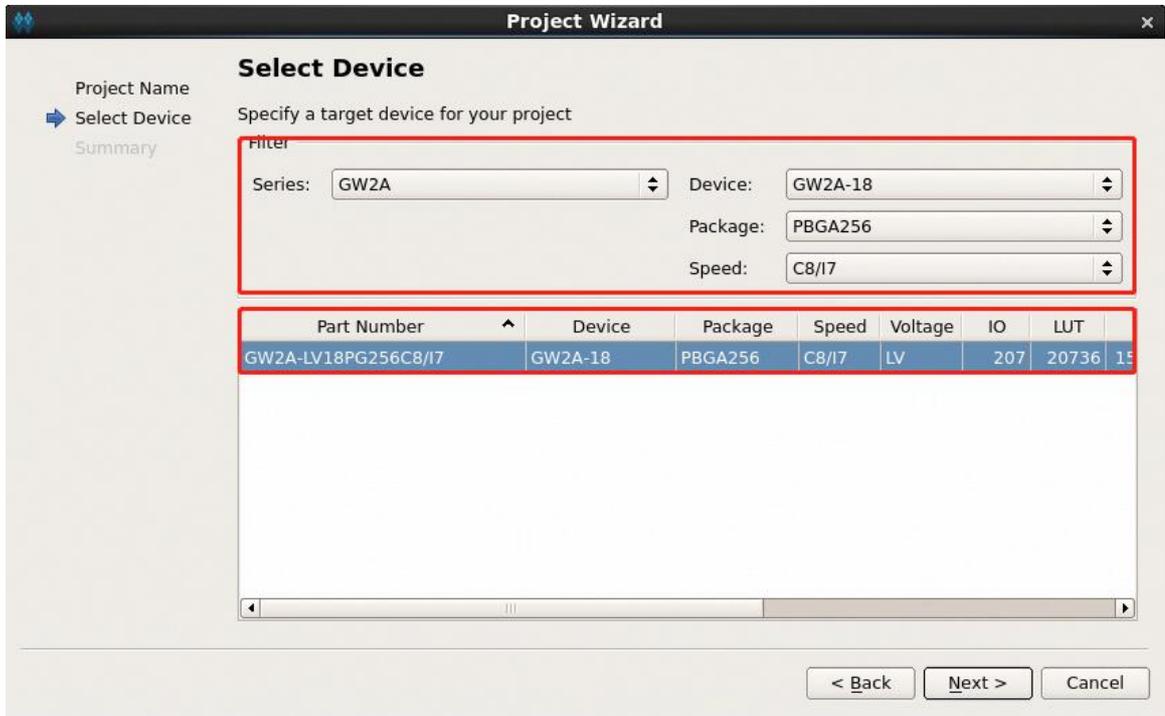


The screenshot shows the 'Project Wizard' dialog box with the 'Project Name' step selected. The 'Name' field is filled with 'gowin_picorv32' and the 'Create in' field is filled with '/xxx/xxx'. A red rectangular box highlights these two input fields. Below the 'Create in' field, there is an unchecked checkbox labeled 'Use as default project location'. At the bottom right of the dialog, there are two buttons: 'Next >' and 'Cancel'.

3.1.3 选择器件

选择 Series、Device、Package、Speed 和 Part Number，如图 3-3 所示。

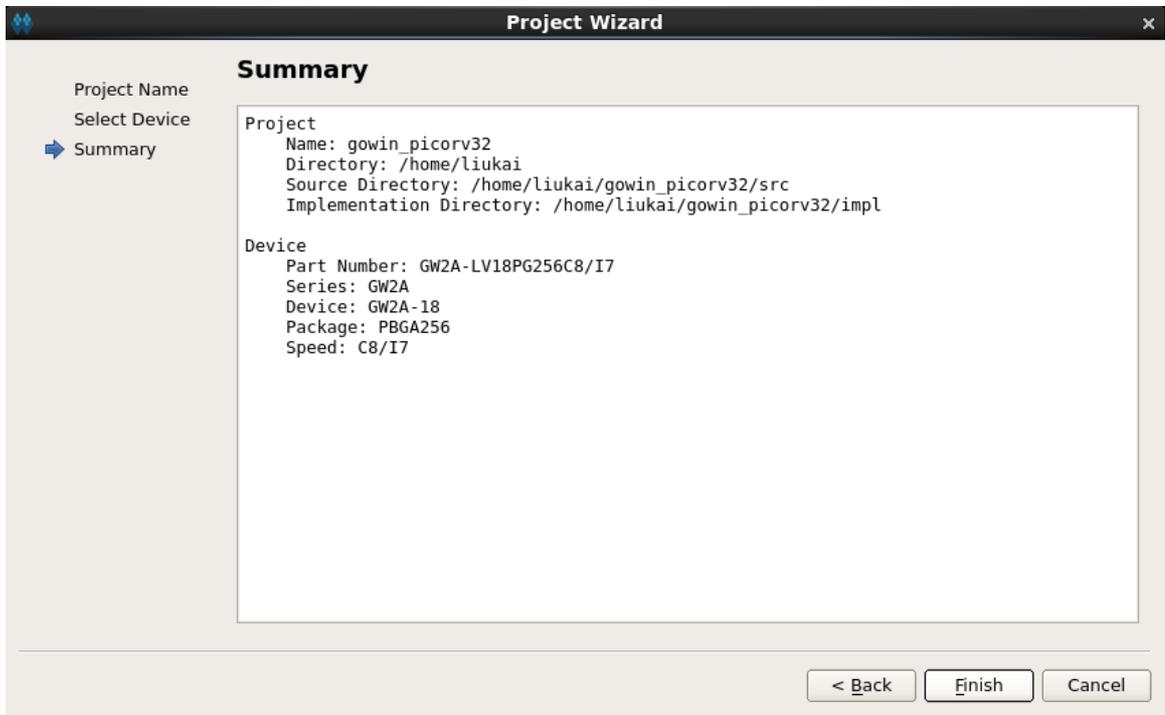
图 3-3 选择器件



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



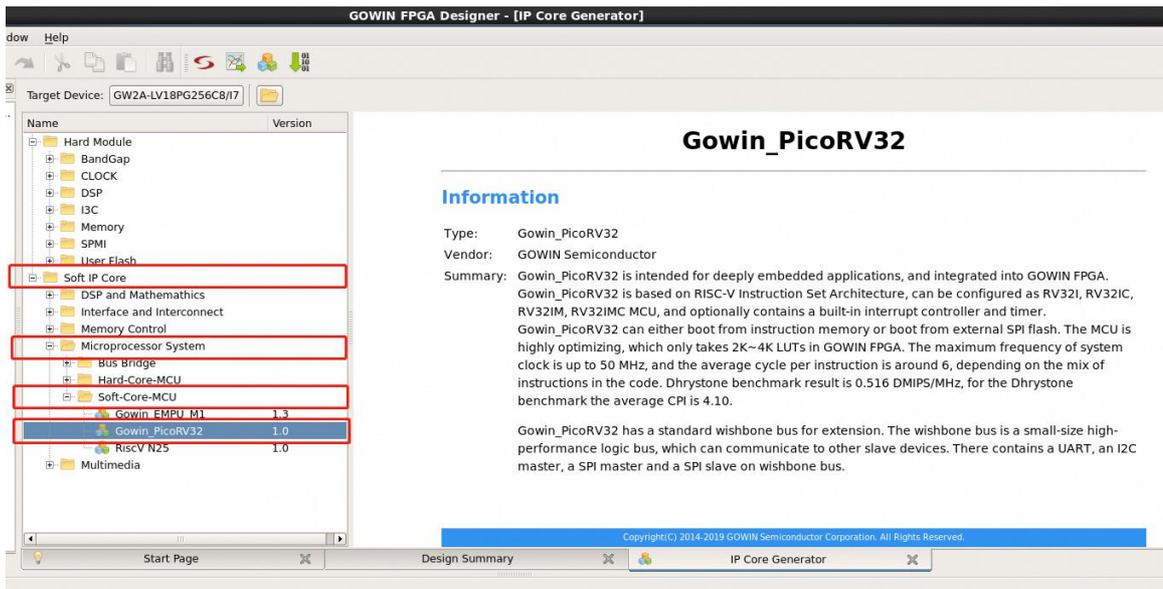
3.2 硬件设计

使用 IP Core Generator 产生 Gowin_PicoRV32 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_PicoRV32，如图 3-5 所示。

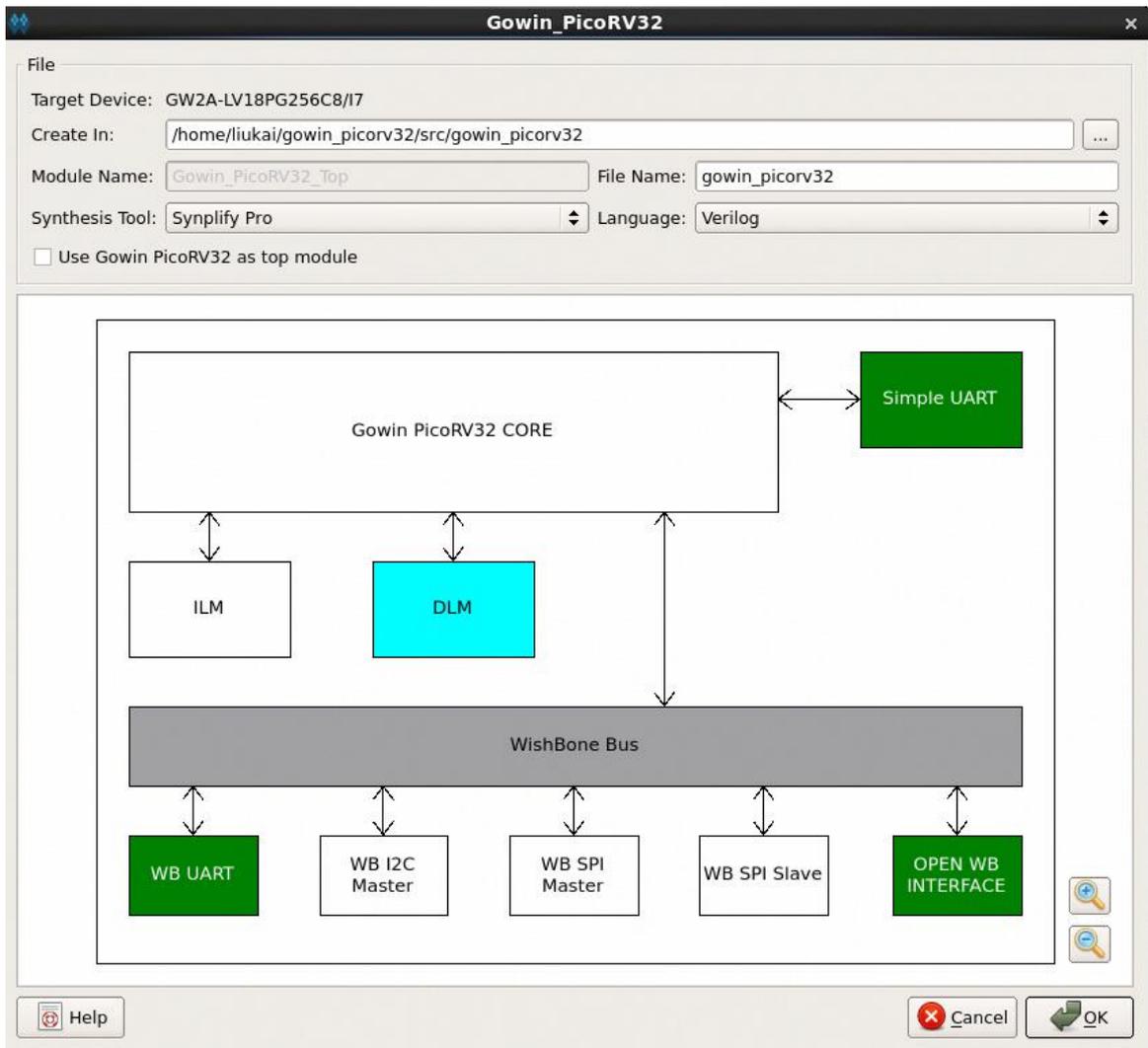
图 3-5 选择 Gowin_PicoRV32



打开 Gowin_PicoRV32，Gowin_PicoRV32 系统架构如图 3-6 所示，包括 Gowin PicoRV32 内核子系统和 Wishbone Bus 子系统。

如果模块配置使能，则该模块呈现绿色标示。

图 3-6 Gowin_PicoRV32 系统架构

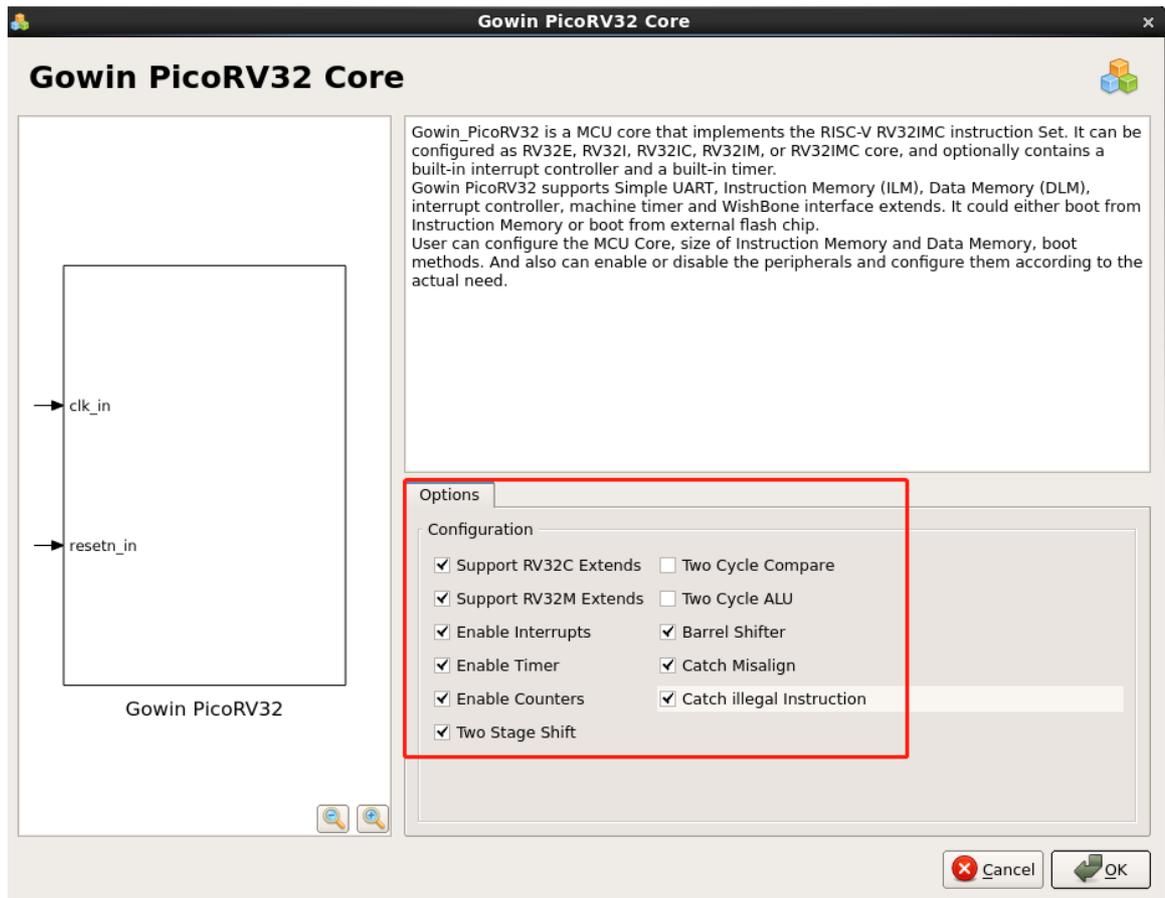


3.2.1 PicoRV32 内核子系统硬件设计

Gowin PicoRV32 CORE 配置

双击 Gowin PicoRV32 CORE，打开 Gowin PicoRV32 CORE 的配置页面，配置选项如图 3-7 所示。

图 3-7 Gowin PicoRV32 CORE 配置选项



Gowin PicoRV32 CORE 硬件设计配置选项，如表 3-1 所示。

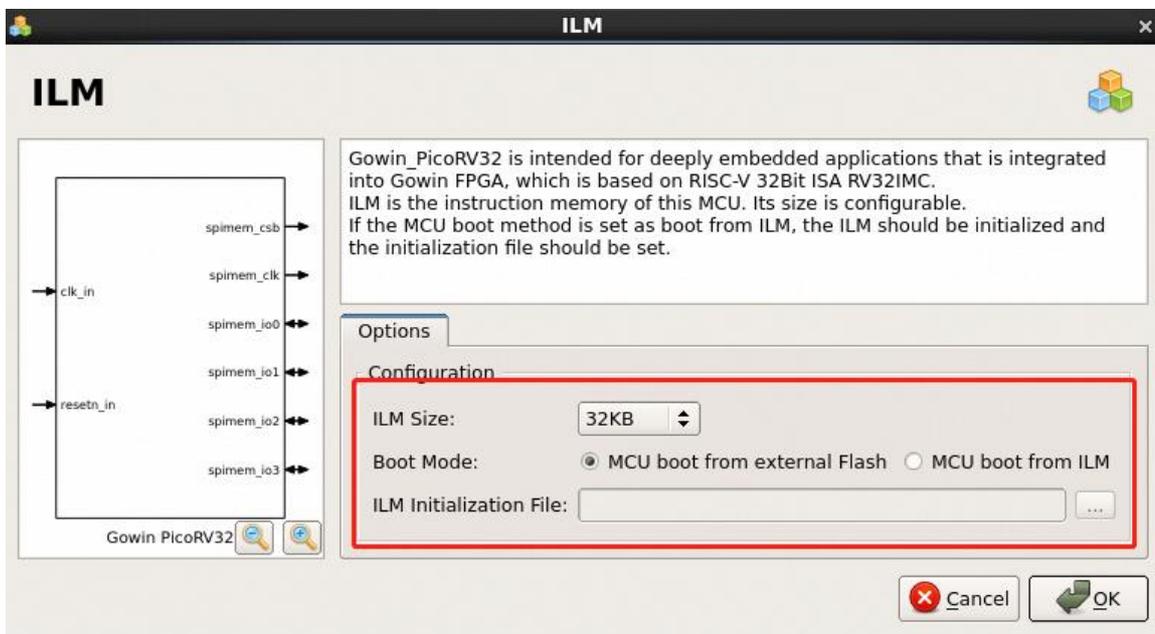
表 3-1 Gowin PicoRV32 CORE 配置选项

配置选项	描述
Support RV32C Extends	选择是否支持 RISC-V 压缩指令集扩展
Support RV32M Extends	选择是否支持 RISC-V 乘法/除法指令集扩展
Enable Interrupts	选择是否支持中断控制
Enable Timer	选择是否支持计时器模块
Enable Counters	选择是否支持计数指令 RDCYCLE[H]/RDTIME[H]/RDINSTRET[H]
Two Stage Shift	选择是否支持两段移位功能（如果支持，可以加速移位操作，但是会增加逻辑资源使用量）
Two Cycle Compare	选择是否支持两周期比较功能（如果支持，可以缩短数据路径长度、提高时序质量，但是会执行比较指令，从而增加一个时钟周期）
Two Cycle ALU	选择是否支持两周期运算功能（如果支持，可以缩短数据路径长度、提高时序质量，但是会执行运算指令，从而增加一个时钟周期）
Barrel Shifter	选择是否支持桶形移位功能
Catch Misalign	选择是否在存储器访问发生地址未对齐错误时进入 TRAP 并停止运行
Catch illegal Instruction	选择是否在执行非法指令时进入 TRAP 并停止运行

指令存储器 ILM 配置

双击指令存储器 ILM，打开指令存储器 ILM 的配置页面，配置选项如图 3-8 所示，可以配置 ILM Size、Gowin_PicoRV32 启动方式和 ILM 初始值文件。

图 3-8 指令存储器 ILM 配置选项

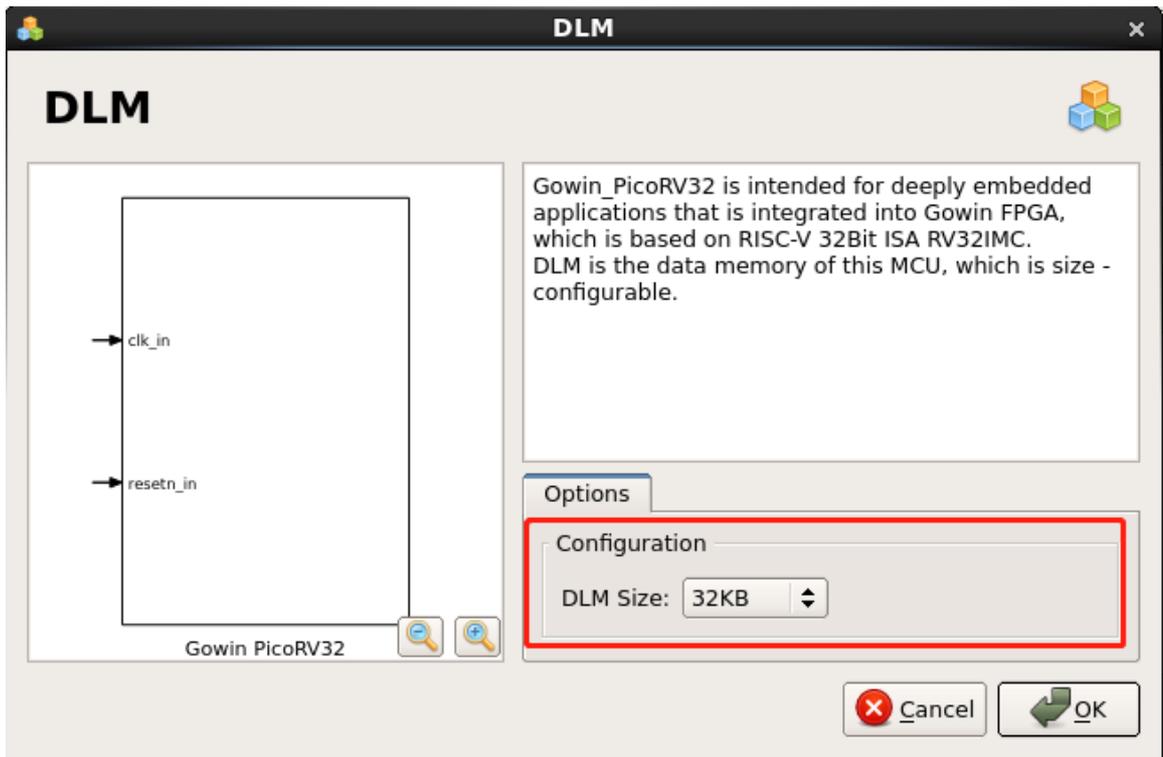


- ILM Size
 - 可以配置 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB
 - GW1N-9 系列 ILM Size 最大选择为 32KB，默认为 16KB
 - GW1NR-9 系列 ILM Size 最大选择为 32KB，默认为 16KB
 - GW2A-18 系列 ILM Size 最大选择为 64KB，默认为 32KB
 - GW2AR-18 系列 ILM Size 最大选择为 64KB，默认为 32KB
 - GW2A-55 系列 ILM Size 最大选择为 256KB，默认为 64KB
- Boot Mode
 - 支持两种启动方式：片外 SPI FLASH 启动（MCU boot from external Flash）和指令存储器 ILM 启动（MCU boot from ILM）
 - 如果选择指令存储器启动方式，则在 ILM Initialization File 中导入 ILM 初始化文件（即 Gowin_PicoRV32 软件编程设计）
 - ILM Initialization File 导入的路径，不能有以数字或 \n、\t 等转义字符存在的文件夹路径！

数据存储器 DLM 配置

双击数据存储器 DLM，打开数据存储器 DLM 的配置页面，配置选项如图 3-9 所示，可以配置 DLM Size。

图 3-9 数据存储 DLM 配置选项



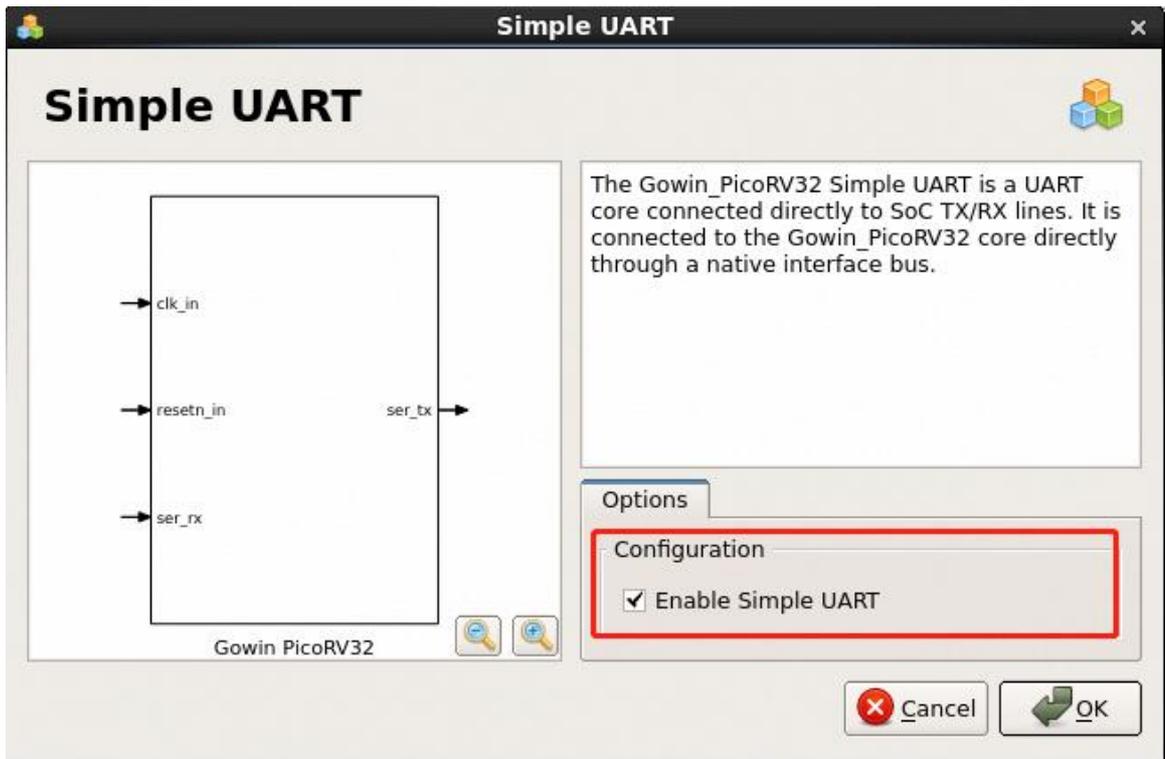
- DLM Size
 - 可以配置 8KB 或 16KB 或 32KB 或 64KB 或 128KB 或 256KB
 - GW1N-9 系列 DLM Size 最大选择为 32KB，默认为 16KB
 - GW1NR-9 系列 DLM Size 最大选择为 32KB，默认为 16KB
 - GW2A-18 系列 DLM Size 最大选择为 64KB，默认为 32KB
 - GW2AR-18 系列 DLM Size 最大选择为 64KB，默认为 32KB
 - GW2A-55 系列 DLM Size 最大选择为 256KB，默认为 64KB

Simple UART 配置选项

双击 Simple UART，打开 Simple UART 的配置页面，配置选项如图 3-10 所示，可以配置是否使能 Simple UART。

如果选择 Enable Simple UART，则 Gowin_PicoRV32 支持 Simple UART，默认为支持。

图 3-10 Simple UART 配置选项



3.2.2 Wishbone Bus 子系统硬件设计

Wishbone Bus 子系统，可以配置支持的外部设备包括 WB UART、WB I2C Master、WB SPI Master、WB SPI Slave 和 Wishbone 总线扩展接口。

Wishbone Bus 子系统配置选项，如表 3-2 所示。

表 3-2 Wishbone Bus 子系统配置选项

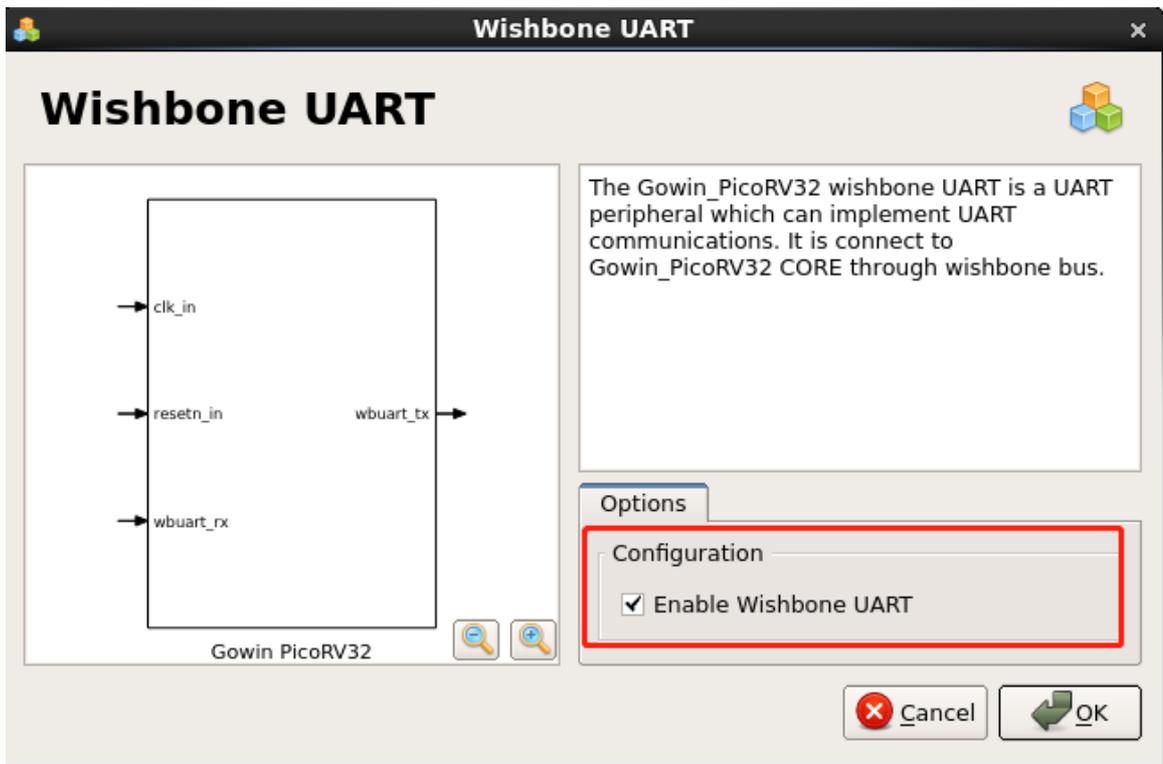
配置选项	描述
Enable Wishbone UART	使能 WB UART，默认关闭
Enable Wishbone I2C Master	使能 WB I2C Master，默认关闭
Enable Wishbone SPI Master	使能 WB SPI Master，默认关闭
Enable Wishbone SPI Slave	使能 WB SPI Slave，默认关闭
Enable Open Wishbone Interface	使能 OPEN WB INTERFACE，默认关闭

WB UART 配置选项

双击 WB UART，打开 Wishbone UART 的配置页面，配置选项如图 3-11 所示。

默认关闭，如果选择 Enable Wishbone UART，则 Gowin_PicoRV32 支持 Wishbone UART。

图 3-11 WB UART 配置选项

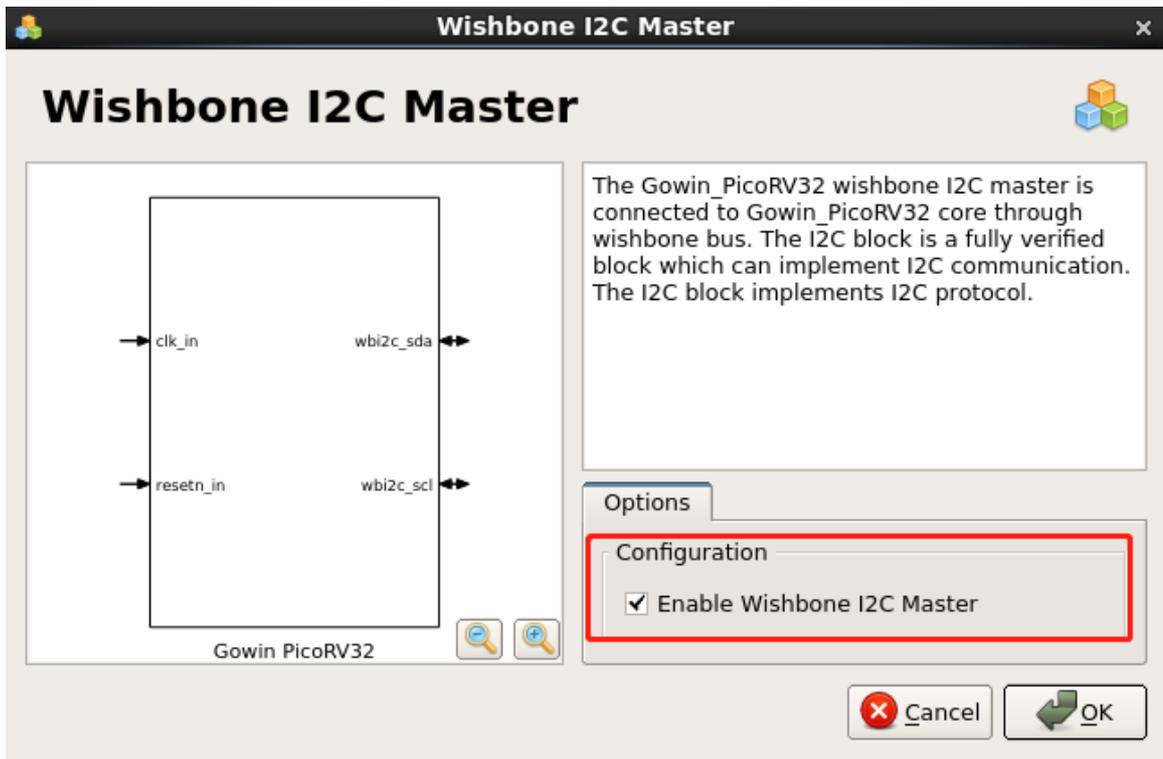


WB I2C Master 配置选项

双击 WB I2C Master，打开 Wishbone I2C Master 的配置页面，配置选项如图 3-12 所示。

默认关闭，如果选择 Enable Wishbone I2C Master，则 Gowin_PicoRV32 支持 Wishbone I2C Master。

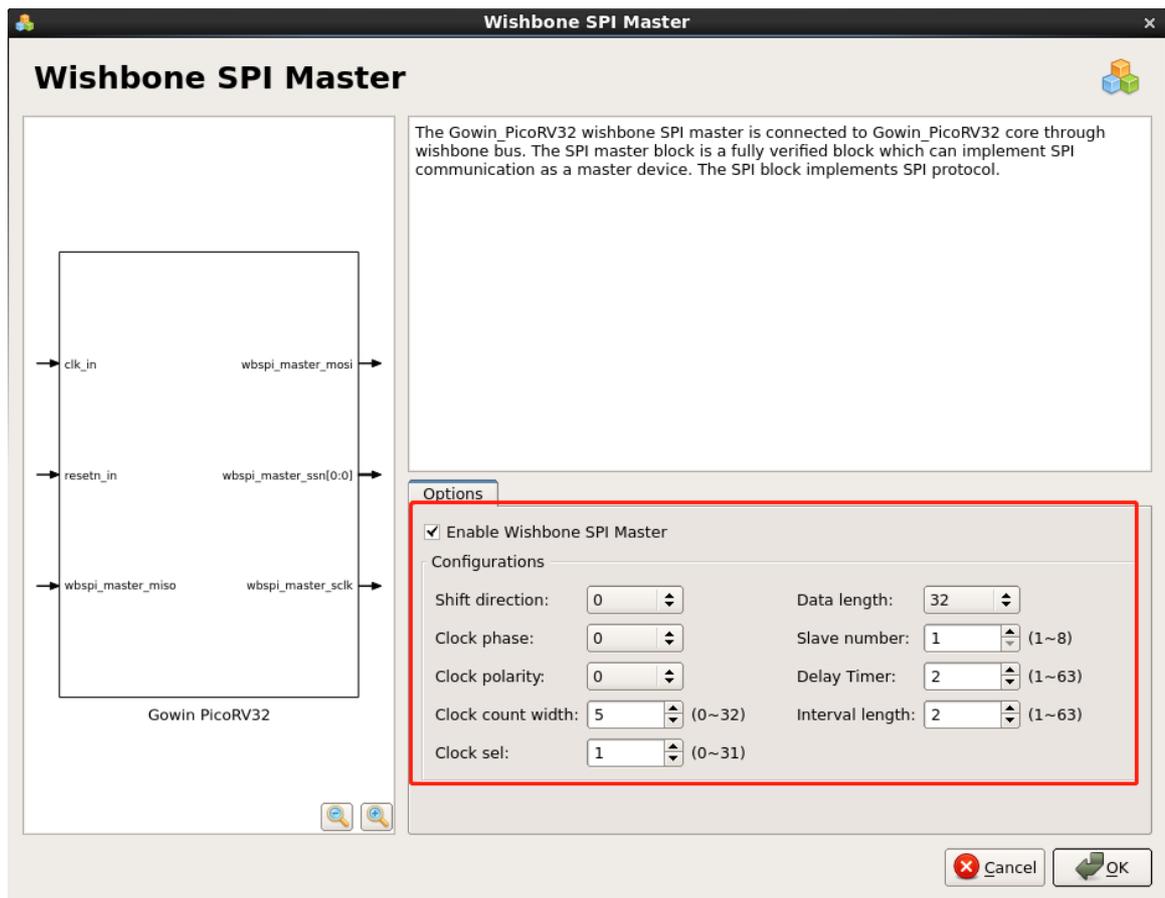
图 3-12 WB I2C Master 配置选项



WB SPI Master 配置选项

双击 WB SPI Master，打开 Wishbone SPI Master 的配置页面，配置选项如图 3-13 所示。

图 3-13 WB SPI Master 配置



默认关闭，如果选择 Enable Wishbone SPI Master，则 Gowin_PicoRV32 支持 Wishbone SPI Master。

如果选择使能 Wishbone SPI Master，则可以配置 Wishbone SPI Master 的参数，如表 3-3 所示。

表 3-3 WB SPI Master 参数配置选项

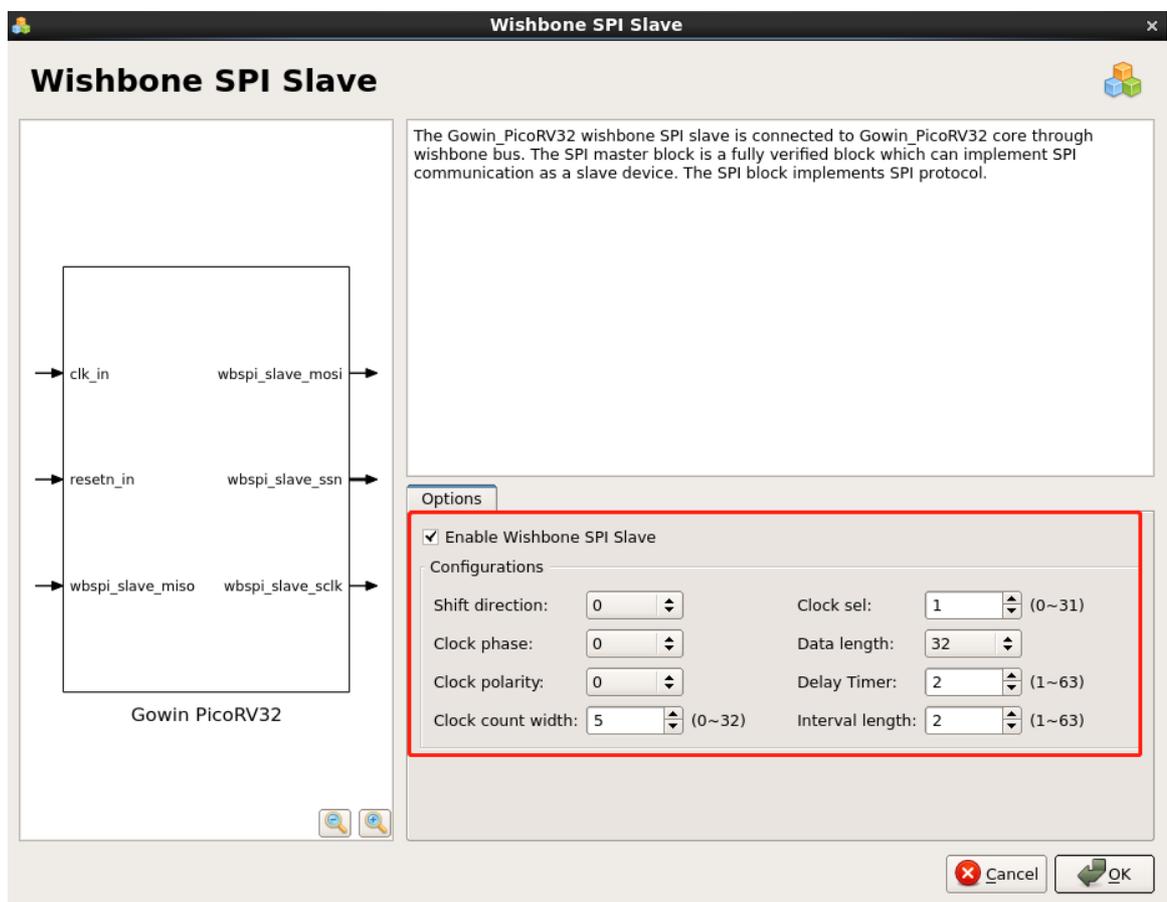
参数选项	描述
Shift direction	指定数据传输格式： - 值为 0 时，首先传输数据的最高位（MSB） - 值为 1 时，首先传输数据的最低位（LSB）
Clock phase	指定 WB SPI Master 时钟相位： - 值为 0 时，数据在 SCLK 的第一个边沿有效 - 值为 1 时，数据在 SCLK 的第二个边沿有效
Clock polarity	指定 WB SPI Master 时钟极性： - 值为 0 时，空闲状态时 SCLK 为低电平 - 值为 1 时，空闲状态时 SCLK 为高电平
Clock count width	指定 clock 计数器范围： - 必须有足够的位宽来满足 SCLK 的数据宽度
Clock sel	指定由 CLK_I 分频产生 SCLK 所需的分频系数： - SCLK 频率计算公式： $SCLK = CLK_I / (2 * (CLOCK_SEL) + 1)$ - 取值范围：0 ~ $2^{(clock\ count\ width)} - 1$
Data length	指定传输数据的位宽： - 取值范围：8/16/32/64

参数选项	描述
Slave number	指定可支持的从机 Slave 数量： - 取值范围：1 ~ 32
Delay time	指定在 SS_N 信号有效后，在进行第一位数据传输前需等待的延迟时间： - 延迟时间计算公式：Delay = Delay Time * (SCLK period / 2) - 取值范围：0 ~ 63
Interval length	指定当 SPI 传输请求后，SS_N 信号需等待的 SCLK 周期数： - 取值范围：0 ~ 63

WB SPI Slave 配置选项

双击 WB SPI Slave，打开 Wishbone SPI Slave 的配置页面，配置选项如图 3-14 所示。

图 3-14 WB SPI Slave 配置



默认关闭，如果选择 Enable Wishbone SPI Slave，则 Gowin_PicoRV32 支持 Wishbone SPI Slave。

如果选择使能 Wishbone SPI Slave，则可以配置 Wishbone SPI Slave 的参数，如表 3-4 所示。

表 3-4 WB SPI Slave 参数配置选项

参数选项	描述
Shift direction	指定数据传输格式： - 值为 0 时，首先传输数据的最高位（MSB） - 值为 1 时，首先传输数据的最低位（LSB）

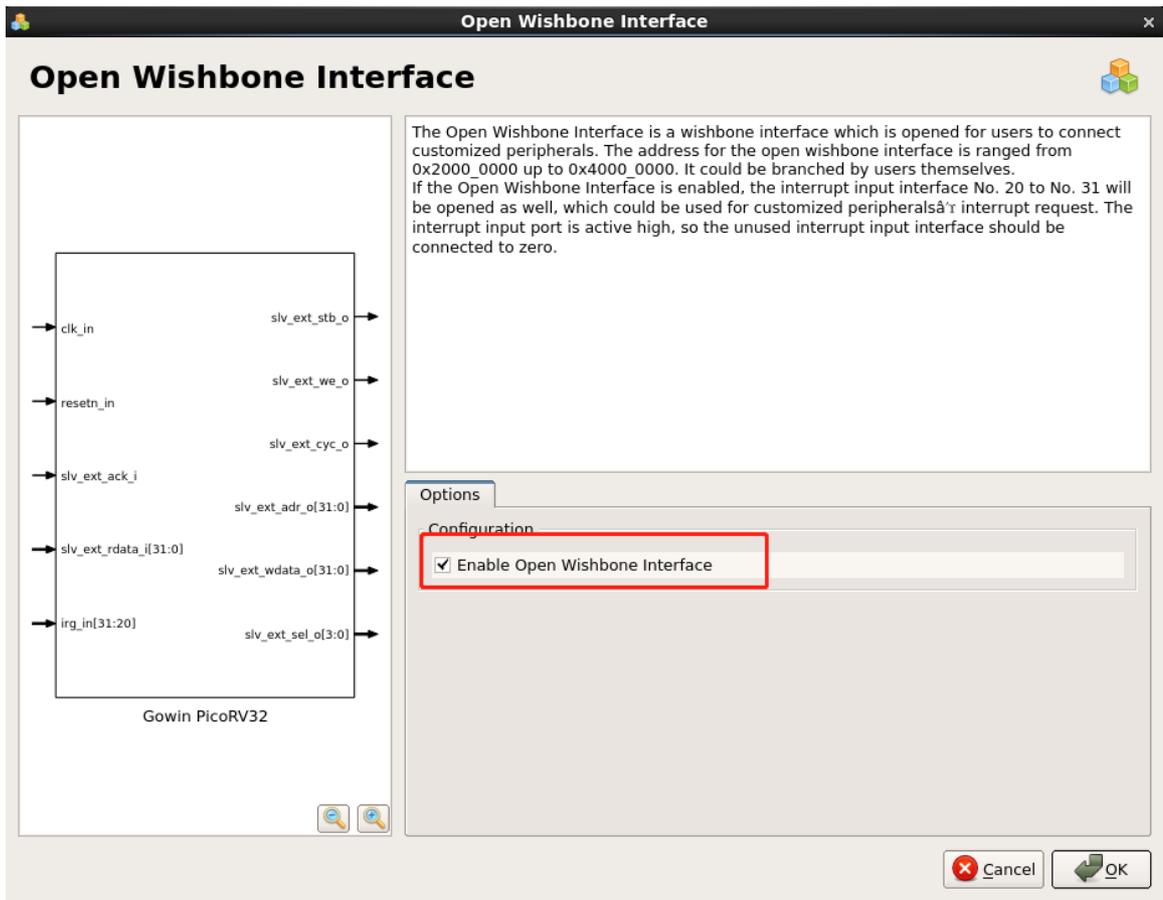
参数选项	描述
Clock phase	指定 WB SPI Slave 的时钟相位： - 值为 0 时，数据在 SCLK 的第一个边沿有效 - 值为 1 时，数据在 SCLK 的第二个边沿有效
Clock polarity	指定 WB SPI Slave 的时钟极性： - 值为 0 时，空闲状态时 SCLK 为低电平 - 值为 1 时，空闲状态时 SCLK 为高电平
Clock count width	指定 clock 计数器范围： - 必须有足够的位宽来满足 SCLK 的数据宽度
Clock sel	指定由 CLK_I 分频产生 SCLK 所需的分频系数： - SCLK 频率计算公式： $SCLK = CLK_I / (2 * (CLOCK_SEL) + 1)$ - 取值范围：0 ~ $2^{(clock\ count\ width)} - 1$
Data length	指定传输数据的位宽： - 取值范围：8/16/32/64
Delay time	指定在 SS_N 信号有效后，在进行第一位数据传输前需等待的延迟时间： - 延迟时间计算公式： $Delay = Delay\ Time * (SCLK\ period / 2)$ - 取值范围：0 ~ 63
Interval length	指定当 SPI 传输请求后，SS_N 信号需等待的 SCLK 周期数； 取值范围：0 ~ 63

OPEN WB INTERFACE 配置

双击打开 OPEN WB INTERFACE，可以选择配置 Open Wishbone Interface，如图 3-15 所示。

默认关闭，如果选择 Enable Open Wishbone Interface，则 Gowin_PicoRV32 支持 Open Wishbone Interface，用户可以在此接口连接扩展 Wishbone 总线接口的外部设备。

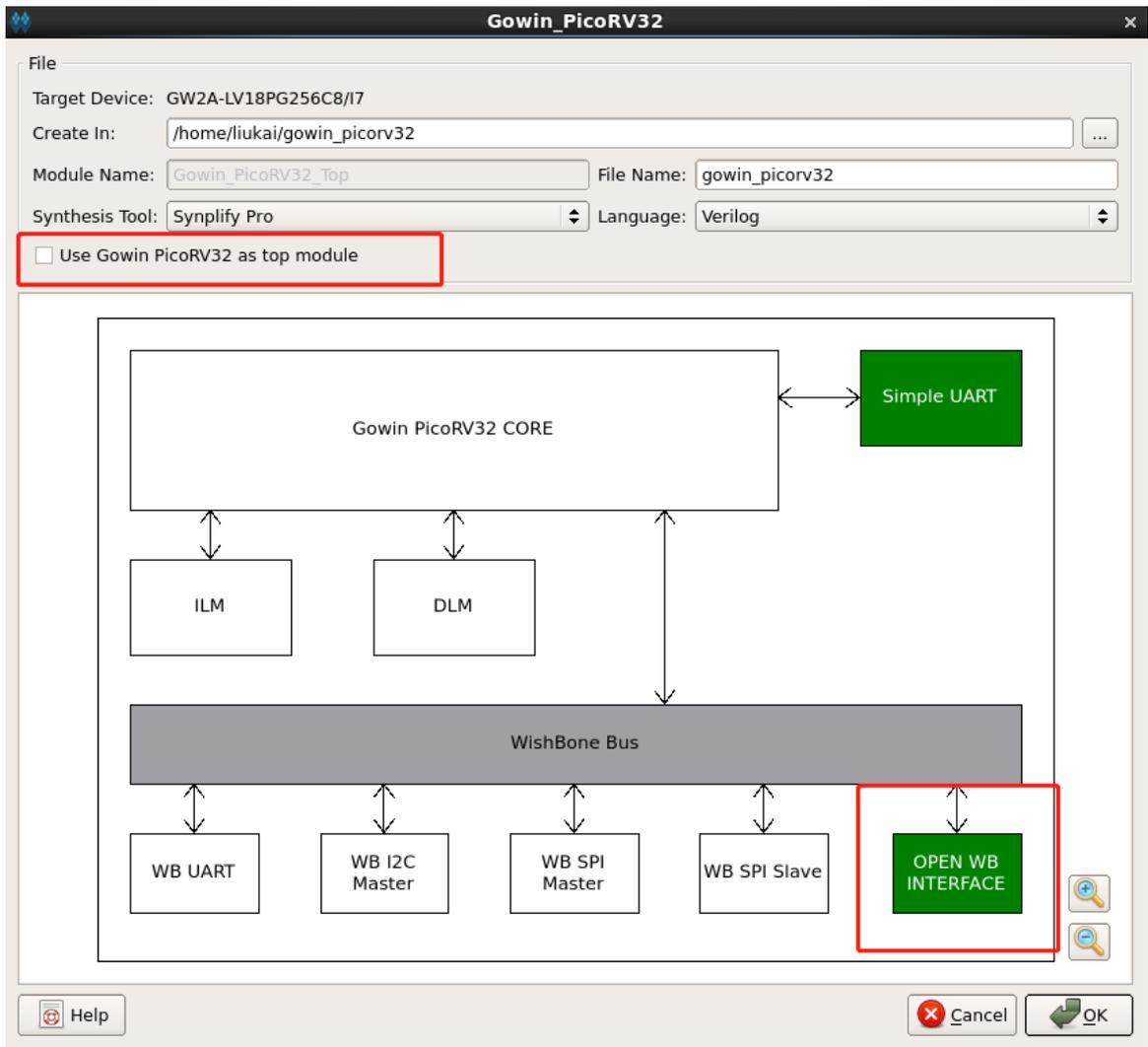
图 3-15 OPEN WB INTERFACE 配置



Top Module 配置

如果以 Gowin_PicoRV32 为总体设计的 Top Module, 则使能“Use Gowin PicoRV32 as top module”选项, 设置 Gowin_PicoRV32 为 Top Module; 如果选择配置 OPEN WB INTERFACE, 扩展用户 Wishbone 总线接口的外部设备, 则禁用“Use Gowin PicoRV32 as top module”选项, 如图 3-16 所示。

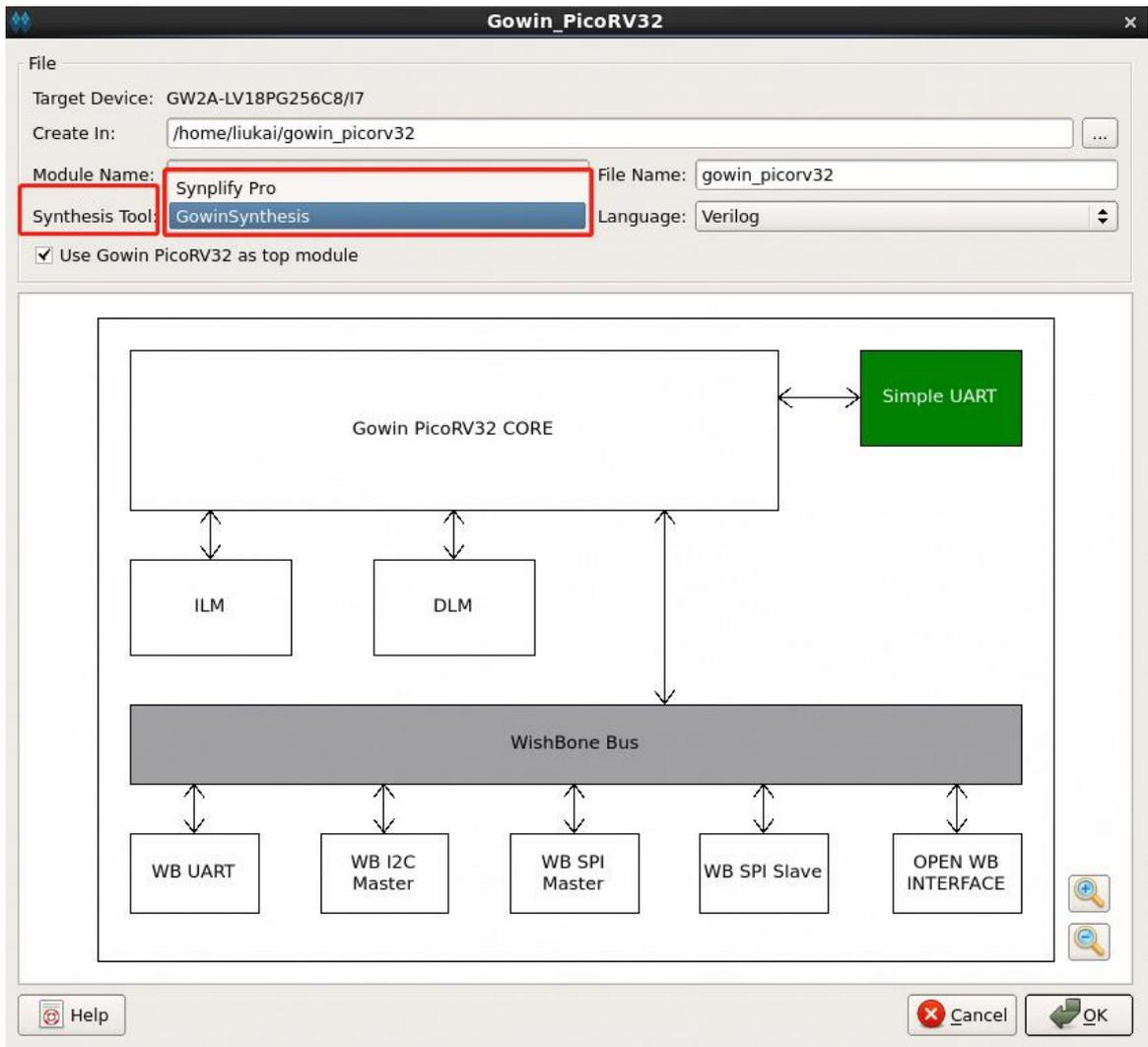
图 3-16 Top Module 配置



综合工具配置

完成 Gowin_PicoRV32 功能配置后，选择产生 Gowin_PicoRV32 的综合工具 Synplify Pro 或 GowinSynthesis，如图 3-17 所示。

图 3-17 综合工具配置



3.3 用户设计

完成 PicoRV32 配置后，产生 Gowin_PicoRV32 硬件设计；实例化 Gowin_PicoRV32，设置为 Top Module 或连接用户设计；导入用户设计，连接 Gowin_PicoRV32，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 工程配置

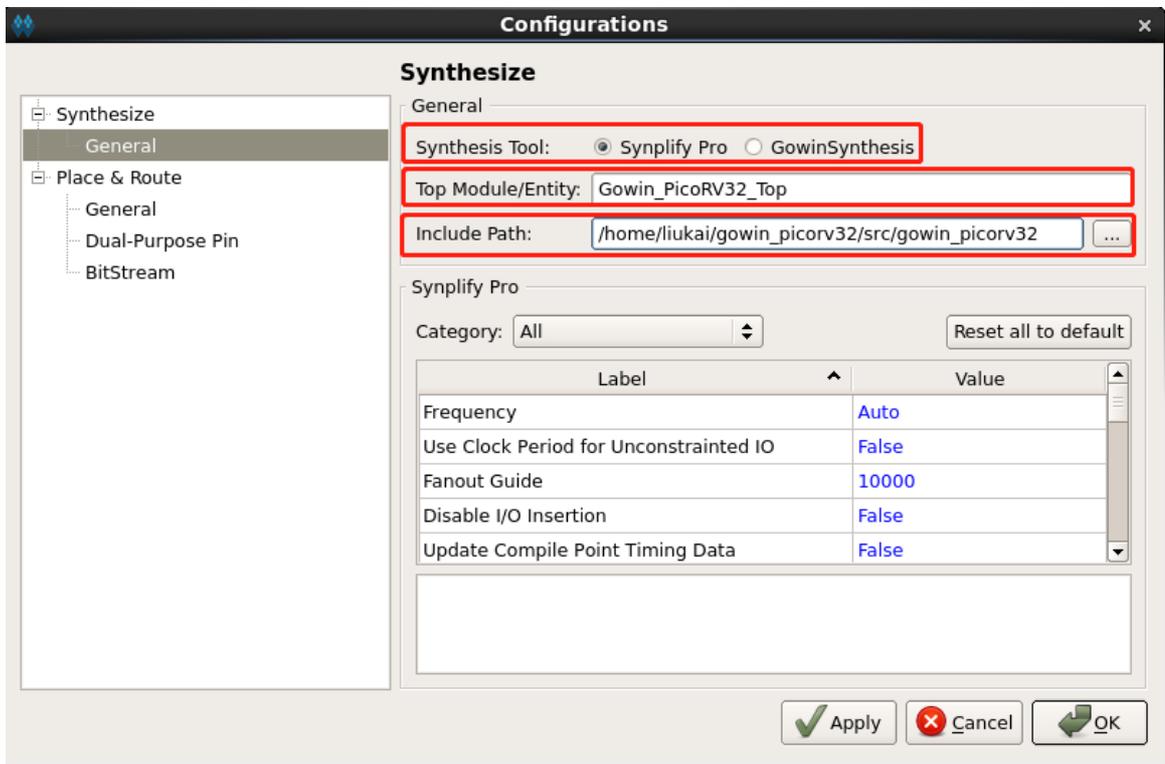
3.5.1 综合选项配置

综合选项配置，如图 3-18 所示。

根据工程设计中的实际顶层模块名称配置顶层模块名称；

根据工程设计中的实际文件路径配置文件引用路径；
配置综合工具，选择 Synplify Pro 或 GowinSynthesis。

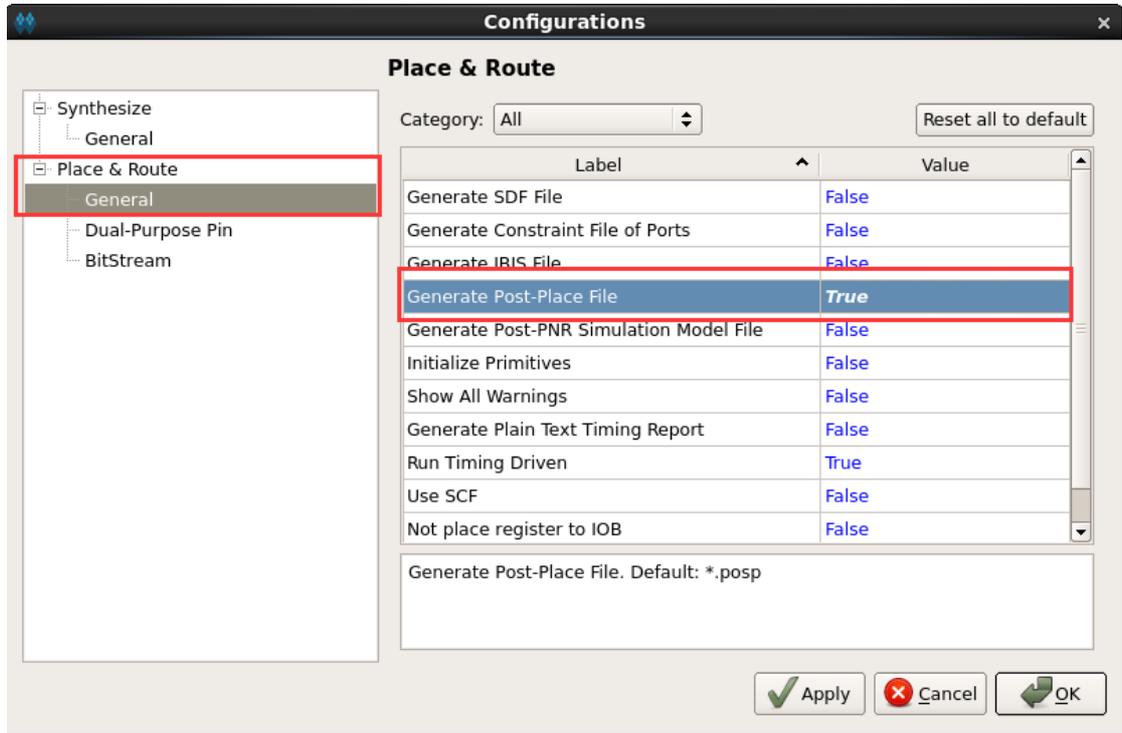
图 3-18 综合选项配置



3.5.2 Post-Place File 生成配置

如果使用 Gowin_PicoRV32 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 3-19 所示，否则不需要配置输出 Post-Place File。

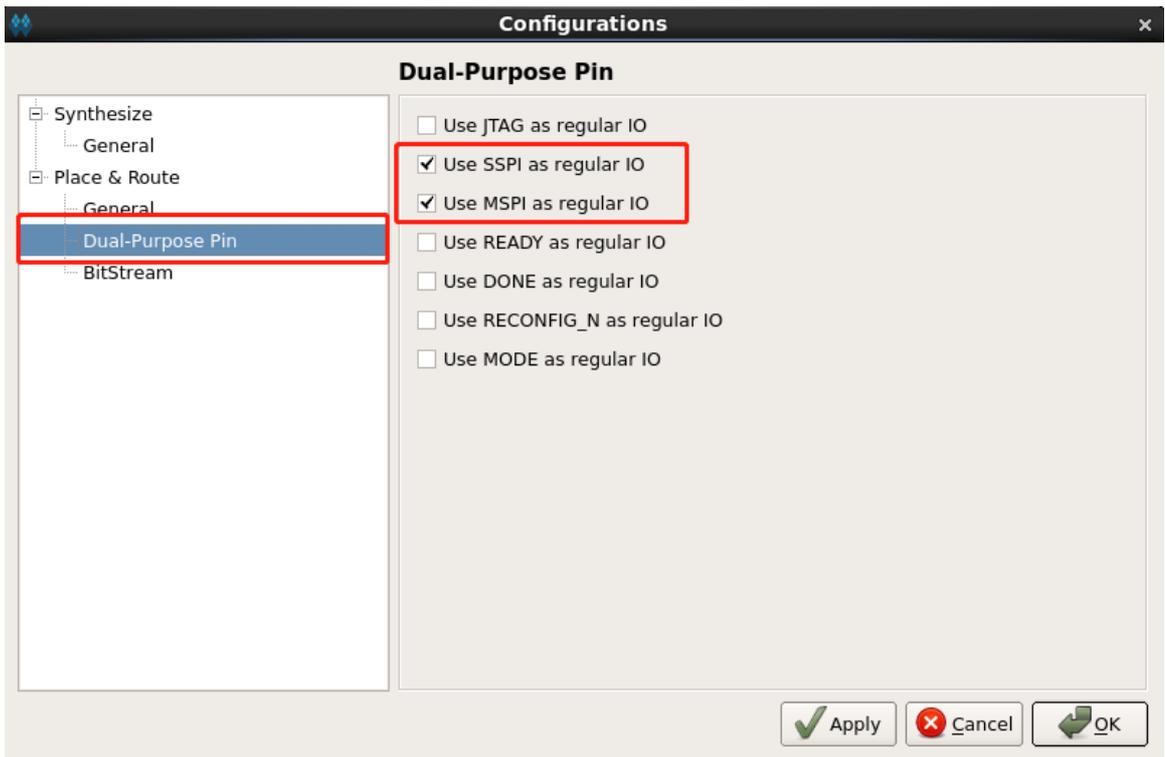
图 3-19 Post-Place File 配置



3.5.3 Dual-Purpose Pin 配置

如果 Gowin_PicoRV32 使用片外 SPI-Flash 下载启动方式，则复用 SSPI 和 MSPI 端口为通用端口，如图 3-20 所示，否则不需要配置端口复用。

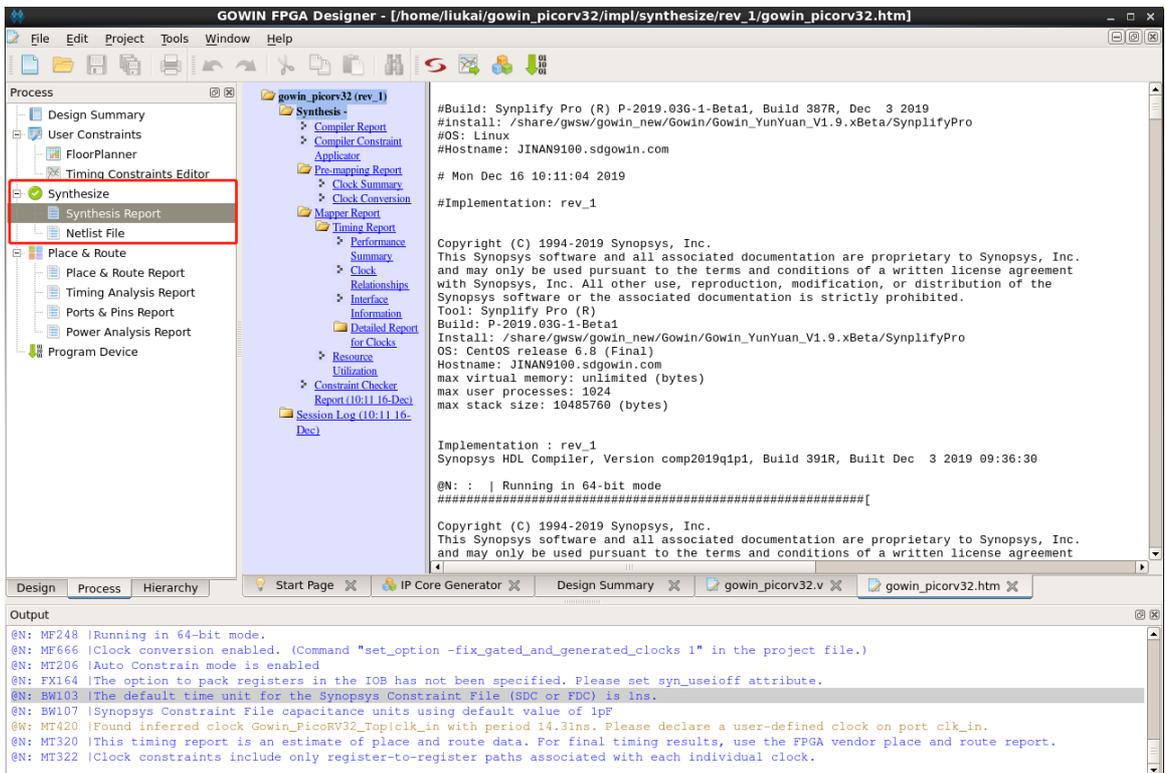
图 3-20 Dual-Purpose Pin 配置



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 3-21 所示。

图 3-21 综合

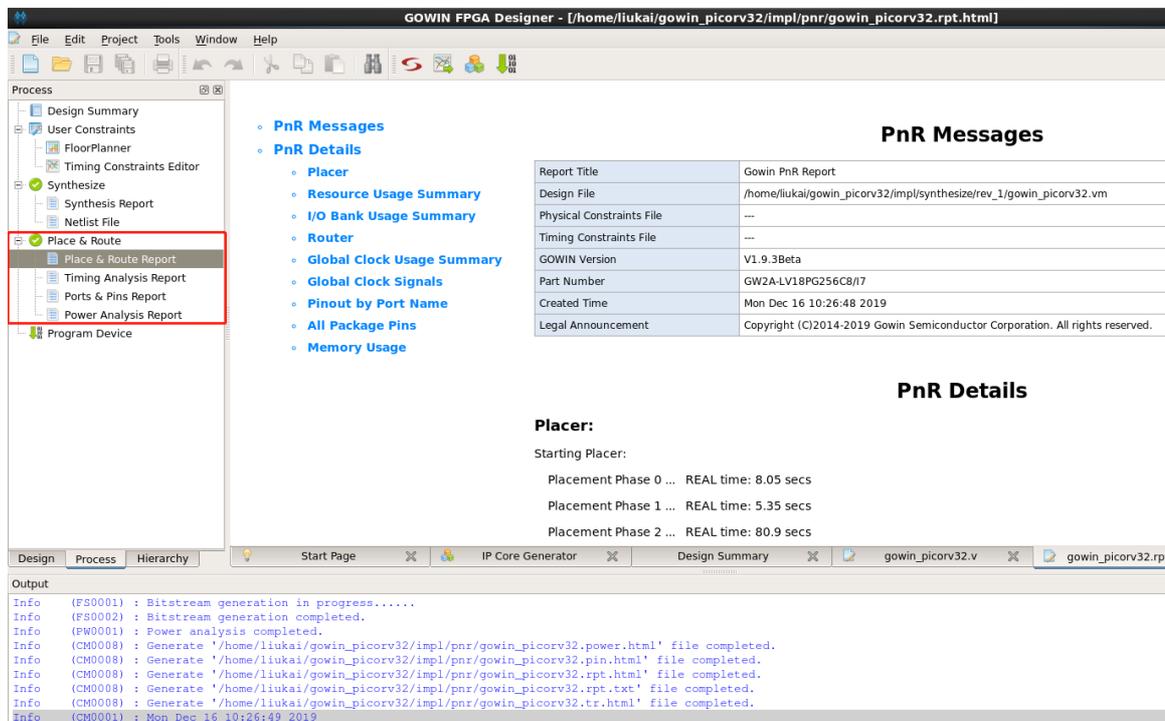


综合工具的使用方法请参考 [SUG100](#)，Gowin 云源软件用户指南。

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 3-22 所示。

图 3-22 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。

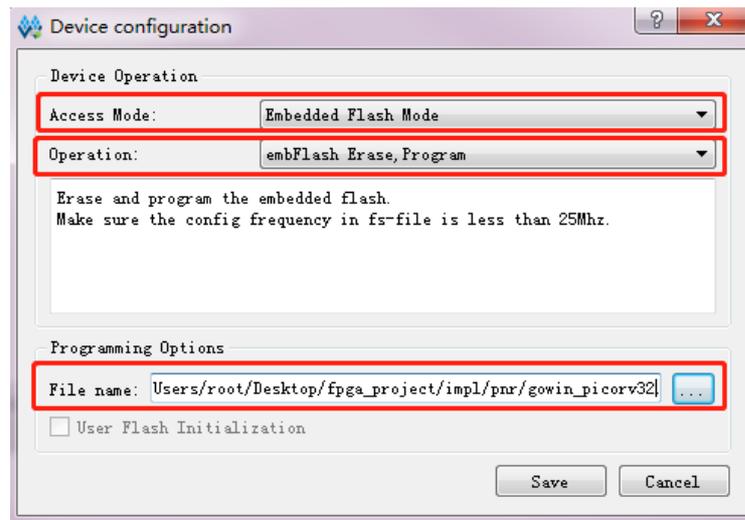
3.8 下载

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 菜单栏 Edit/Configure Device 或工具栏 Configure Device ，打开 Device configuration 对话框。

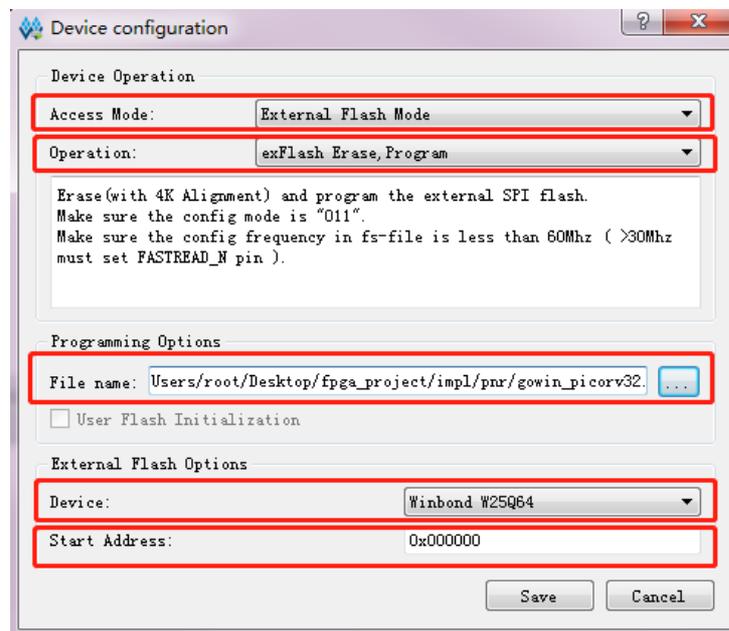
如果板载器件 GW1N-9/GW1NR-9 系列，Access Mode 下拉列表中选择“Embedded Flash Mode”，Operation 下拉列表中选择“embFlash Erase, Program”，Programming Options 中导入需要下载的码流文件，单击“Save”，如图 3-23 所示。

图 3-23 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 3-24 所示。

图 3-24 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_PicoRV32 提供硬件设计参考设计：
Gowin_PicoRV32\ref_design\FPGA_RefDesign\。

