



Gowin 云源软件 用户指南

SUG100-4.4.1, 2025-04-30

版权所有 © 2025 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin、GOWIN、gowin、GowinSynthesis、高云、云源、小蜜蜂以及晨熙均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本文档内容的部分或全部，并不得以任何形式传播。

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，或以禁止反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
2021/06/17	2.5	● 更新界面截图及部分描述； ● 移除Synplify Pro相关的内容。
2021/11/02	2.6	● 更新SSPI和MSPI的复用管脚信息； ● 更新-ireg_in_iob/-oreg_in_iob/-ioreg_in_iob的描述； ● 移除MODE复用管脚的配置； ● Place & Route新增BitStream配置项：Power On Reset； ● 增加仿真文件介绍。
2022/05/20	2.7	更新Loading Rate的取值。
2022/07/28	2.8	● Place & Route新增配置选项：Route Maxfan； ● 新增Library的使用方法。
2022/10/28	2.9	● 更新配置选项power_on_reset名称； ● 新增Bitstream配置选项：Turn Off Bandgap； ● 新增DSim仿真验证云平台； ● 新增第9章 附录。
2022/12/16	3.0	● 新增Device Version信息； ● Synthesize新增配置项TclPre； ● 将配置选项Background Programming的取值Internal改为GoConfig/UserLogic； ● Place & Route新增配置选项Generate Post-PnR VHDL Simulation Model File。
2023/03/31	3.1	● 将配置选项Background Programming的取值GoConfig/UserLogic分拆为GoConfig、UserLogic两个选项，并更新相关描述； ● 工具栏中增加FloorPlanner和Timing Constraints Editor； ● 新增Tcl命令-clock_route_order。
2023/04/20	3.2	● 更新Place & Route和Bitstream配置选项； ● 新增Bitstream配置选项Multi Boot和MSPI JUMP。
2023/05/25	3.3	● 新增Bitstream配置选项Enable External Master Config Clock和Enable CMSE； ● 新增配置选项Global，可设置VCCX值。
2023/06/30	3.4	综合配置选项Ram R/W Check默认值改为不勾选。
2023/08/18	3.5	● GW5A-25-MBGA121N的配置选项Use SSPI as regular IO为勾选状态且不可更改； ● GW5AT-138/GW5AST-138/GW5A-138器件的配置选项Place input register to IOB、Place output register to IOB和Place inout register to IOB的默认值改为False； ● GW5AT-138/GW5AST-138/GW5A-138器件的Place & Route新增配置选项Replicate Resources。
2023/09/28	3.6	● 更新GW5A(S)(T)-138、GW5A(R)-25 Loading Rate的取值； ● 更新配置选项CMSE； ● 更新配置选项Feature sysControl下的MSPI JUMP选项。
2023/10/31	3.7	● 更新图4-16工程文件属性对话框；

日期	版本	说明
		● 新增Tcl命令create_project和import_files。
2023/11/30	3.8	● Program Device名称更新为Programmer; ● Place & Route配置选项Place Option新增取值2; ● 更新小蜜蜂家族和晨熙®家族器件的Loading Rate取值; ● 新增Tcl命令run close; ● 新增Output Base Name配置选项。
2024/02/02	3.9	● 更新GW2AN-9X、GW2AN-18X Loading Rate的取值; ● 配置选项Enable CMSE更新为Enable SEU Handler; ● 新增配置选项Constraints，可用于配置Frequency。
2024/03/29	4.0	● 新增GW5AT-60 Loading Rate的取值; ● 新增Bitstream配置选项“Enable CTP”。
2024/06/28	4.1	● 新增Tcl命令open_project; ● Place & Route下新增配置选项“VCC”; ● 新增虚拟输入输出调试工具。
2024/08/09	4.1.1	● Place & Route 配置选项 Place Option 新增取值 3; ● 更新 open_project 命令说明。
2024/10/25	4.2	● Place & Route 配置选项 Place Option 新增取值 4; ● C 版 GW1N-2/GW1NR-2/GW1N-1P5 的配置选项 Background Programming 新增取值“GoConfig Mode1”; ● Place & Route 下新增配置选项“Incremental PnR”。
2024/12/31	4.3	● Hierarchy 层级显示窗口增加搜索功能; ● 新增眼图分析工具 GoBert。
2025/02/28	4.4	● BitStream 配置选项中 SPI Flash 地址访问模式 Normal 改为 Single; ● 更新第 8 章 Tcl 命令说明。
2025/04/30	4.4.1	● 界面新增夜间模式配置; ● 器件选择窗口中新增搜索功能。

目录

目录 i

图目录..... v

表目录..... viii

1 关于本手册 1

 1.1 手册内容..... 1

 1.2 相关文档..... 1

 1.3 术语、缩略语 2

 1.4 技术支持与反馈..... 2

2 云源概述..... 3

 2.1 简介..... 3

 2.2 芯片支持..... 4

 2.3 云源的安装与启动 4

3 云源用户界面 6

 3.1 标题栏 7

 3.2 菜单栏 7

 3.2.1 File 菜单栏..... 7

 3.2.2 Edit 菜单栏 7

 3.2.3 Project 菜单栏 8

 3.2.4 Tools 菜单栏 8

 3.2.5 Window 菜单栏..... 9

 3.2.6 Help 菜单栏 9

 3.3 工具栏 10

 3.4 工程管理区（Design） 11

 3.5 过程管理区（Process） 11

 3.6 设计层级显示区（Hierarchy） 11

 3.6.1 右键菜单..... 11

 3.6.2 资源显示..... 12

 3.6.3 文件加密 12

3.6.4 搜索功能	15
3.7 源文件编辑区	15
3.8 信息输出区	16
4 云源使用	19
4.1 新建工程	19
4.2 打开工程	22
4.3 编辑工程	23
4.3.1 编辑工程器件	23
4.3.2 编辑工程文件	24
4.3.3 编辑工程配置	30
4.4 管理工程过程	54
4.4.1 Design Summary	55
4.4.2 User Constraints	56
4.4.3 Synthesize	56
4.4.4 Place & Route	57
4.4.5 Programmer	58
4.5 工程存档及恢复	59
4.5.1 工程存档	59
4.5.2 恢复存档工程	60
4.6 退出软件	61
5 云源集成工具	62
5.1 物理约束编辑器	62
5.2 时序约束编辑器	64
5.3 IP Core 产生器	64
5.4 在线逻辑分析仪	66
5.5 功耗分析工具	68
5.6 存储器初始化文件编辑器	69
5.7 User Flash 初始化文件编辑器	71
5.7.1 二进制格式 (Bin File)	71
5.7.2 十六进制格式 (Hex File)	72
5.8 原理图查看器	74
5.9 虚拟输入输出调试工具	75
5.10 眼图分析工具 GoBert	76
6 云源输出文件	78
6.1 综合报告	78
6.2 布局布线报告	79

6.3 端口属性报告	80
6.4 时序报告	81
6.5 功耗分析报告	81
7 仿真文件.....	83
7.1 功能仿真文件	83
7.2 时序仿真文件	83
8 Tcl 命令说明.....	85
8.1 启动命令行模式.....	85
8.1.1 gw_sh.exe	85
8.2 命令	86
8.2.1 命令分类	86
8.2.2 命令列表	86
8.3 命令介绍.....	87
8.3.1 add_file	87
8.3.2 create_ipc	88
8.3.3 create_project.....	89
8.3.4 generate_target	90
8.3.5 get_ips	90
8.3.6 import_files	91
8.3.7 list_property	92
8.3.8 open_project.....	93
8.3.9 read_ipc.....	94
8.3.10 report_property	94
8.3.11 rm_file	95
8.3.12 run	96
8.3.13 run close	96
8.3.14 saveto	96
8.3.15 set_device	97
8.3.16 set_file_enable	98
8.3.17 set_file_prop	99
8.3.18 set_option	99
8.3.19 set_property	140
8.3.20 source.....	141
8.3.21 write_ip_tcl	142
9 附录.....	144
9.1 文件说明	144
9.2 文件和文件夹命名规则	145

9.3 安全声明.....	145
---------------	-----

图目录

图 3-1 软件用户界面	6
图 3-2 Hierarchy 窗口右键菜单显示	12
图 3-3 Hierarchy 窗口资源信息显示	12
图 3-4 Pack User Design 对话框	13
图 3-5 Pack User Design 对话框打印信息	14
图 3-6 Pack User Design 对话框提取错误信息	14
图 3-7 Hierarchy 窗口搜索功能	15
图 3-8 Find & Replace 对话框	16
图 3-9 Search Result 窗口	16
图 3-10 信息输出窗口	17
图 3-11 Tcl 编辑窗口	18
图 4-1 新建工程	19
图 4-2 新建工程向导	20
图 4-3 设置 FPGA 器件信息	21
图 4-4 工程信息	21
图 4-5 打开文件	22
图 4-6 工程文件窗口	23
图 4-7 工程配置芯片型号	24
图 4-8 新建文件对话框	25
图 4-9 新建 Verilog File 对话框	25
图 4-10 新建配置文件对话框	26
图 4-11 GPA 配置文件窗口	26
图 4-12 Design 窗口右键菜单	27
图 4-13 工程文件编辑菜单	27
图 4-14 外部编辑器	28
图 4-15 Save Modified Files 对话框	28
图 4-16 工程文件属性对话框	29
图 4-17 工程配置选项对话框	30
图 4-18 General 配置选项	30

图 4-19 Frequency(MHz)配置选项	31
图 4-20 GowinSynthesis 参数配置项.....	31
图 4-21 配置布局布线选项.....	33
图 4-22 配置 Voltage 选项	34
图 4-23 配置布局选项.....	34
图 4-24 配置布线选项.....	35
图 4-25 配置复用管脚选项.....	36
图 4-26 Unused Pin 选项.....	37
图 4-27 Incremental PnR 选项	38
图 4-28 配置 Bitstream General 选项	39
图 4-29 配置 sysControl 选项	40
图 4-30 Background Programming 设为 I2C.....	49
图 4-31 Background Programming 设为 I2C/JTAG/SSPI/QSSPI	50
图 4-32 Frequency Divider 选项.....	51
图 4-33 Enable SEU Handler 选项.....	52
图 4-34 Enable Error Injection 选项.....	52
图 4-35 非 GW5A(N)(S)(R)(T)系列器件配置 Feature sysControl 选项	53
图 4-36 GW5A(N)(S)(R)(T)系列器件配置 Feature sysControl 选项	53
图 4-37 工程 Process 窗口	55
图 4-38 工程信息显示.....	55
图 4-39 右击 Synthesize.....	57
图 4-40 Programmer 界面	59
图 4-41 工程存档对话框	60
图 4-42 恢复存档工程对话框	60
图 5-1 Chip Array 窗口	63
图 5-2 Package View 窗口	63
图 5-3 创建时序约束界面.....	64
图 5-4 IP Core Generator 窗口	65
图 5-5 GAO 配置文件窗口	67
图 5-6 GAO 界面	67
图 5-7 GPA 配置文件窗口.....	68
图 5-8 初始化文件新建 New 对话框	70
图 5-9 初始化文件新建 New File 对话框.....	70
图 5-10 初始化文件配置窗口	70
图 5-11 列数配置	71
图 5-12 批量设置	71
图 5-13 初始化文件新建 New 对话框	72

图 5-14 初始化文件新建 New File 对话框	73
图 5-15 初始化文件配置窗口	73
图 5-16 批量设置	74
图 5-17 RTL Design Viewer 窗口	74
图 5-18 Post-Synthesis Netlist Viewer 窗口	75
图 5-19 GVIO 配置文件窗口	76
图 5-20 GoBert 窗口	77
图 6-1 GowinSynthesis 综合报告	78
图 6-2 Place & Route Report	79
图 6-3 Ports & Pins Report.....	80
图 6-4 时序报告内容.....	81
图 6-5 功耗分析报告.....	82
图 7-1 选项配置	84

表目录

表 1-1 术语、缩略语	2
表 3-1 常见 Warning 和 Error 信息	17
表 4-1 PnR 配置选项含义描述	32
表 4-2 BitStream 配选项含义描述	39
表 4-3 Loading Rate 值及计算方式（一）	42
表 4-4 Loading Rate 值及计算方式（二）	43
表 4-5 Loading Rate 值及计算方式（三）	44
表 4-6 Loading Rate 值及计算方式（四）	46
表 4-7 Loading Rate 值及计算方式（五）	47
表 4-8 Loading Rate 值及计算方式（六）	47
表 4-9 Loading Rate 值及计算方式（七）	48
表 4-10 Background Programming 取值	50
表 4-11 子配置项	54
表 4-12 子配置项	54
表 9-1 源文件	144
表 9-2 执行文件	145

1 关于本手册

1.1 手册内容

本手册主要描述高云半导体云源软件（以下简称云源）的安装及操作，旨在帮助用户快速熟悉云源的使用流程，提高设计效率。本手册中的软件界面截图参考的是 V1.9.11.02 版本，因软件版本升级，部分信息可能会略有差异，具体以用户软件版本的信息为准。

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可下载、查看以下相关文档：

- [SUG940, Gowin 设计时序约束指南](#)
- [SUG935, Gowin 设计物理约束指南](#)
- [SUG114, Gowin 在线逻辑分析仪用户指南](#)
- [SUG282, Gowin 功耗分析工具用户指南](#)
- [SUG502, Gowin Programmer 用户指南](#)
- [UG285, Gowin 存储器（BSRAM & SSRAM）用户指南](#)
- [SUG283, Gowin 原语用户指南](#)
- [UG286, Gowin 时钟资源（Clock）用户指南](#)
- [UG287, Gowin 数字信号处理（DSP）模块用户指南](#)
- [UG289, Gowin 可编程通用管脚（GPIO）用户指南](#)
- [UG295, Gowin 闪存资源（User Flash）用户指南](#)
- [UG299, Arora V 模拟数字转换器（ADC）用户指南](#)
- [UG306, Arora V 时钟资源（Clock）用户指南](#)
- [UG305, Arora V 数字信号处理（DSP）模块用户指南](#)
- [UG304, Arora V 可编程通用管脚（GPIO）用户指南](#)
- [UG300, Arora V 存储器（BSRAM & SSRAM）用户指南](#)

- [SUG1018, Arora V设计物理约束用户指南](#)
- [SUG1189, Gowin 虚拟输入输出调试工具用户指南](#)
- [SUG1198, Gowin GoBert 工具用户指南](#)

1.3 术语、缩略语

本手册中的相关术语、缩略语及相关释义如表 1-1 所示。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
CRC	Cyclic Redundancy Check	循环冗余校验
FloorPlanner	FloorPlanner	物理约束编辑器
FPGA	Field Programmable Gate Array	现场可编程门阵列
GAO	Gowin Analyzer Oscilloscope	在线逻辑分析仪
GowinSynthesis	GowinSynthesis	高云半导体综合工具
GPA	Gowin Power Analyzer	功耗分析工具
GVIO	Gowin Virtual Input/Output	虚拟输入输出
IP Core	Intellectual Property Core	知识产权核
PCIe	Peripheral Component Interconnect Express	外设组件互连表达
PnR	Place & Route	布局布线
Schematic Viewer	Schematic Viewer	HDL原理图查看器
SEU Handler	Single-Event Upsets Handler	单粒子翻转处理器
Tcl	Tool Command Language	工具命令语言
Timing Constraints Editor	Timing Constraints Editor	时序约束编辑器

1.4 技术支持与反馈

高云半导体提供全方位技术支持，在使用过程中如有任何疑问或建议，可直接与公司联系：

网址: www.gowinsemi.com.cn

E-mail: support@gowinsemi.com

Tel: +86 755 8262 0391

2 云源概述

2.1 简介

云源设计系统是专为高云半导体 **FPGA** 芯片而配套的集成电路设计与实现工具。云源系统针对高云 **FPGA** 芯片构架的低功耗、低成本特点进行了全面的优化设计，覆盖了从 **RTL** 电路功能描述到生成 **FPGA** 码流文件的完整流程，包括了优化设计、自动设计、图形交互设计等功能，具有性能优越、容易使用等特点。

云源设计系统主要功能：

- 软件系统支持高云 **FPGA** 芯片所有功能，覆盖从 **RTL** 电路功能描述到生成 **FPGA** 码流文件的完整设计流程
- 综合优化工具 **GowinSynthesis** 支持高性能的逻辑设计和综合
- 自动设计和交互式图形设计并用
- 支持 Centos6.8/7.0/7.3/7.5/8.2(64 bits)、Ubuntu18.04/20.04LTS、Win7/8/10/11(32 bits/64 bits)、Win XP (32 bits)系统
- 千万门级软件
- 支持 **VHDL**、**Verilog HDL** 和 **System Verilog** 语言
- 支持高云产品优化的芯片架构
- 具有独创的快速、高性能算法的布局和布线系统
- 精确的时序分析和时序报告
- 时钟分析和控制保证了更好的时序性能
- 支持各种时序约束和物理约束
- 支持实时监测硬件电路信号并加以存储，同时以时序波形图直观显示
- 资源共享技术可提高芯片利用率、降低成本

云源设计系统主要特点：

- 一体化设计
 - 设计可分阶段完成，也可一揽子自动完成

- 可选择命令行模式或图形交互模式完成设计
- 利用脚本设计，可灵活地设计任何单一模块而不影响一体化设计全程
- 优化设计
 - 网表优化设计
 - 快速时序优化分析和设计
 - 资源分析和优化
- 分层设计和分析
 - 支持分层网表结构输入和输出
 - 同时支持扁平化网表结构输入和输出
 - 分层图形显示、追踪、分析网表
- 方便灵活的交互图形设计
 - 用户界面简单清晰
 - 包含项目、设计模块、工具和输出部分
 - 设计约束输入、选择和更新
 - 快速时序分析和报告
 - Push button 设计技术

2.2 芯片支持

云源目前支持小蜜蜂家族芯片和晨熙家族芯片，具体的芯片类型、芯片主要资源及封装方式等详见高云官方网站介绍。

- 小蜜蜂家族芯片：www.gowinsemi.com.cn/prod_view.aspx
- 晨熙家族芯片：www.gowinsemi.com.cn/prod_view.aspx
- Arora V 芯片：http://www.gowinsemi.com.cn/prod_view.aspx

注！

软件版本不同，支持的芯片可能会略有差异，具体信息请参考所使用的软件版本。

2.3 云源的安装与启动

云源在 Windows 系统下的安装方式一致，双击云源安装包，根据提示进行安装，安装结束后会默认在 PC 桌面创建快捷方式。云源在 Linux 系统下解压安装包文件夹即可完成软件安装。

软件安装完成后在第一次启动时需要先对 License 进行配置，软件 License 是一种格式合同，由高云半导体与用户签订，用以规定和限制软件用户使用软件的权利，以及高云半导体应尽的义务。

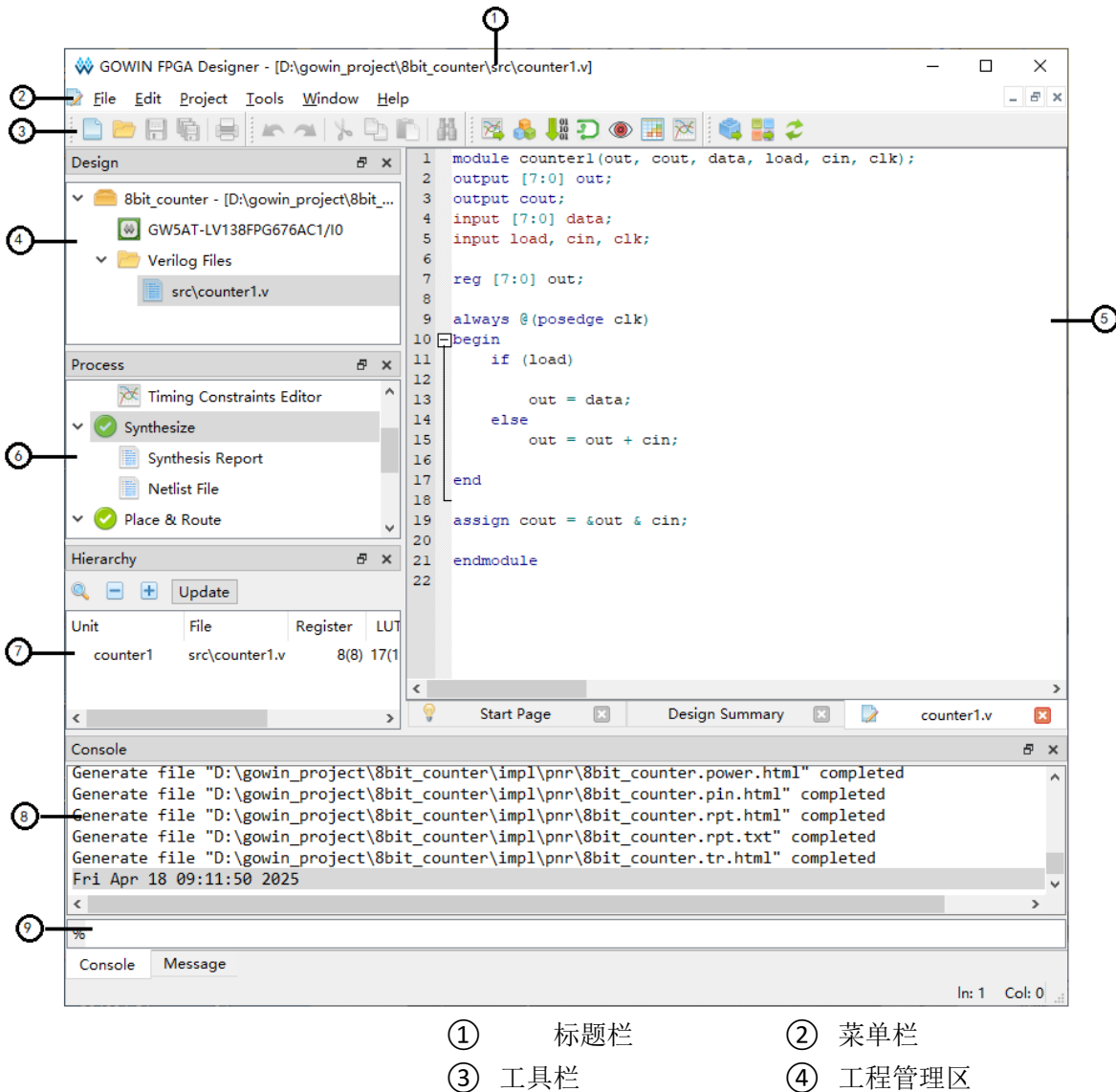
注！

云源软件的安装地址不支持含有中文的路径，云源软件详细的安装与启动指南请参考 [SUG501, Gowin 云源软件快速安装启动指南](#)。

3 云源用户界面

云源用户界面如图 3-1 所示，主要分为标题栏、菜单栏、工具栏、工程管理区（Design）、过程管理区（Process）、源文件编辑区、设计层级显示区（Hierarchy）、信息输出区和 Tcl 命令编辑区。

图 3-1 软件用户界面



- ① 标题栏
- ② 菜单栏
- ③ 工具栏
- ④ 工程管理区

- | | |
|-------------|---------|
| ⑤ 源文件编辑区 | ⑥ 过程管理区 |
| ⑦ 设计层级显示区 | ⑧ 信息输出区 |
| ⑨ Tcl 命令编辑区 | |

3.1 标题栏

主要显示云源软件名称和当前打开的文件名称。

3.2 菜单栏

主要提供一些常用菜单以及工程所需的启动工具，包括 File、Edit、Project、Tools、Window 和 Help 选项，详情如下：

3.2.1 File 菜单栏

- Open Example Project...: 打开示例工程
- New (Ctrl+N): 新建
- Open (Ctrl+O): 打开
- Save (Ctrl+S): 保存
- Save As...: 另存为
- Save All (Ctrl+Shift+S): 保存所有文档
- Close: 关闭
- Close All: 关闭所有打开的文档
- Close Project: 关闭当前工程
- Print Preview...: 打印预览
- Print... (Ctrl+P): 打印
- Recent Files: 最近打开的文件，可以重新选择打开
- Recent Projects: 最近打开过的工程，可以重新选择打开
- Exit: 软件退出

3.2.2 Edit 菜单栏

- Undo (Ctrl+Z): 撤销
- Redo (Ctrl+Y): 重做
- Cut (Ctrl+X): 剪切
- Copy (Ctrl+C): 复制
- Paste (Ctrl+V): 粘贴
- Select All (Ctrl+A): 选择全部
- Find & Replace (Ctrl+F): 查找替换关键词
- Toggle Comment Selection (Ctrl+I): 对所选内容添加注释

- **Increase Indent (Tab):** 增加缩进
- **Decrease Indent (Shift+Tab):** 减少缩进
- **Macros:** 宏, 该选项下包含三个子选项:
 - **Start Record**, 单击 **Start Record** 后, IDE 中对可编辑文件进行的编辑操作会被录制下来;
 - **Stop Recording**, 停止录制;
 - **Play Macro (Alt+R)**, 单击 **Play Macro** 会对可编辑文件执行所录制的操作。

3.2.3 Project 菜单栏

- **Archive Project:** 工程存档
- **Restore Archived Project:** 恢复存档工程
- **Set Device:** 设置当前工程的器件信息
- **Configuration:** 打开配置窗口
- **Design Summary:** 显示当前工程的具体信息

3.2.4 Tools 菜单栏

- **Start Page:** 开始页, 包含四部分内容, 即 **Recent Projects**、**Quick Start**、**Tools** 和 **User Manuals**.
 - **Recent Projects:** 最近打开过的工程列表, 最多会保留 10 个工程;
 - **Quick Start** 包含三项: **New Project** (新建工程)、**Open Project** (打开工程)、**Open Example Project** (打开示例工程);
 - **Tools** 包含三项: **Floorplanner** (启动物理约束编辑器)、**Timing Constraints Editor** (启动时序约束编辑器)、**Programmer** (启动编程器);
 - **User Manuals** 包含两项: **Manual for LittleBee** (小蜜蜂家族用户手册)、**Manual for Arora** (晨熙家族用户手册)。
- **Gowin Analyzer Oscilloscope:** 高云在线逻辑分析仪
- **Schematic Viewer:** HDL 设计原理图查看器
- **IP Core Generator:** IP Core 产生器
- **Programmer:** 编程器
- **FloorPlanner:** 物理约束编辑器
- **Timing Constraints Editor:** 时序约束编辑器
- **DSim:** 仿真验证云平台
- **GoBert:** SerDes 分析工具

- **Options:** 包含三个选项: Environment、Text Editor、External Editor
 - **Environment:** 设置 IDE 参数, 包括显示模式 (Theme)、语言 (Language)、工具栏图标大小 (Toolbar Icon Size) 以及默认新建工程的路径。显示模式 (Theme) 有两个可配置项, 传统模式 (Classic) 和夜间模式 (Dark)。设置显示模式和语言后需要重新启动 IDE 后设置才会生效;
 - **Text Editor:** 设置文本编辑器属性, 包括字体、字号、配色方案、是否显示行号、是否可视化空白符、是否高亮显示当前行、是否高亮显示匹配的括号;
 - **External Editor:** 设置第三方文本编辑器, 可以选择是否始终使用第三方编辑器打开设计文件。

3.2.5 Window 菜单栏

- **Full Screen (F11):** 对 IDE 界面进行全屏显示
- **Tile:** 平铺显示
- **Cascade:** 覆盖显示
- **Reset Layout:** 恢复初始设置
- **Panels:** 对界面中各区域显示模块的选择, 共有五个选项:
 - Design (工程管理区)、Hierarchy (设计层级显示区)、Process (过程管理区)、Message (输出信息分类汇总区)、Console (全部输出信息显示区)
- **Start Page:** 在源文件编辑区显示开始页
- **Design Summary:** 在源文件编辑区显示 Design Page, 页面包含 General 和 Target Device 两项内容, 详细介绍如下:
 - **General:** 工程概述信息, 包括工程路径信息和所使用的综合工具;
 - **Target Device:** 工程器件信息, 包括封装信息、速度等级和内核电压。

3.2.6 Help 菜单栏

- **View Help:** 查看软件编译过程中的输出信息帮助文档
- **Contact Us:** 官网联系我们地址
- **Manage License:** 许可证管理, 详细的许可证配置管理方法请参考 [SUG501, Gowin 云源软件快速安装启动指南](#)
- **About:** 软件版本信息

3.3 工具栏

提供一些常用功能的快速访问入口，从左至右依次为：

- “” (Ctrl+N)：新建文件或工程 (New File or Project)
- “” (Ctrl+O)：打开文件或工程 (Open File or Project)
- “” (Ctrl+S)：保存文件 (Save)
- “” (Ctrl+Shift+S)：一键保存所有被打开编辑过的文件 (Save All)
- “” (Ctrl+P)：打印 (Print)
- “” (Ctrl+Z)：撤销 (Undo)
- “” (Ctrl+Y)：重做 (Redo)
- “” (Ctrl+X)：剪切 (Cut)
- “” (Ctrl+C)：复制 (Copy)
- “” (Ctrl+V)：粘贴 (Paste)
- “” (Ctrl+F)：查找 (Find)
- “”：启动在线逻辑分析仪 (Gowin Analyzer Oscilloscope)，相关文档见 [SUG114, Gowin 在线逻辑分析仪用户指南](#)。
- “”：启动 IP 核产生器 (IP Core Generator)
- “”：启动编程器 (Programmer)，相关文档见 [SUG502, Gowin Programmer 用户指南](#)。
- “”：启动 DSim 仿真验证云平台
- “”：启动眼图分析工具
- “”：启动物理约束编辑器 (FloorPlanner)，相关文档见 [SUG935, Gowin 设计物理约束指南](#)。
- “”：启动时序约束编辑器 (Timing Constraints Editor)，相关文档见 [SUG940, Gowin 设计时序约束指南](#)。
- “”：执行综合 (Run Synthesis)
- “”：执行布局布线 (Run Place & Route)
- “”：执行综合、布局布线 (Run All)

3.4 工程管理区 (Design)

提供工程及其相关文件的管理和显示功能，显示或编辑工程所用器件信息、用户设计文件、用户设计约束文件以及配置文件等。

3.5 过程管理区 (Process)

提供用户 FPGA 设计流程，包括综合 (Synthesize)、布局布线 (Place & Route) 以及编程器件 (Programmer)，同时可双击启动时序约束和物理约束工具进行约束文件编辑。

3.6 设计层级显示区 (Hierarchy)

加载设计文件后，软件会先对设计文件进行解析，Hierarchy 窗口会显示当前工程的设计层级关系。通过 Hierarchy 窗口可以定位某个 module 的定义以及实例化在设计文件中的位置，还可以将某个 module 设为 top module。在 Hierarchy 窗口中，Unit 列显示设计文件的 module 层级结构，Files 列显示 module 定义所在的文件。另外，可以在 Hierarchy 窗口中对工程设计文件进行搜索，可以按 module name 进行搜索，也可以按 file name 搜索。目前 Hierarchy 已支持对 Verilog 语言、VHDL 语言以及 System Verilog 语言的解析。

3.6.1 右键菜单

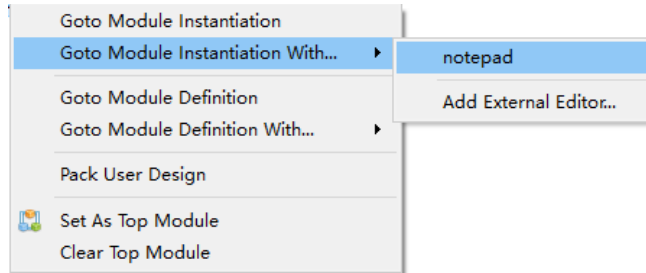
Hierarchy 窗口中 module 处的右键菜单中目前已支持的功能：

- **Goto Module Instantiation:** 跳转到该 module 在源文件中的实例化位置，默认用云源自带的编辑器打开。如果在菜单栏 Tools > Options > External Editor 中配置了第三方编辑器且勾选了 “Always Use External Editor”，Goto Module Instantiation 会默认用第三方编辑器打开源文件；
- **Goto Module Instantiation With...:** 跳转到该 module 在源文件中的实例化位置，在右键菜单中提供已设置的第三方编辑器和 “Add External Editor”，如图 3-2 所示。如果选择 “Add External Editor”，则弹出 “External Editor” 的 Options 选项对话框，可以对第三方编辑器进行设置；
- **Goto Module Definition:** 跳转到该 module 在源文件中的定义位置，默认用云源自带的编辑器打开。如果在菜单栏 Tools > Options > External Editor 中配置了第三方编辑器且勾选了 “Always Use External Editor”，Goto Module Definition 会默认用第三方编辑器打开源文件；
- **Goto Module Definition With...:** 跳转到该 module 在源文件中的定义位置，在右键菜单中提供已设置的第三方编辑器和 “Add External Editor”，如图 3-2 所示。如果选择 “Add External Editor”，则弹出 “External Editor” 的 Options 选项对话框，可以对第三方编辑器进行设置；
- **Pack User Design:** 加密用户设计，可对该 module 及其子 module 进

行加密；

- **Set As Top Module:** 将该 module 设为 top module，被设置为 top 的 module 会添加标记 “” 表明当前 module 是 top module，且原有的 hierarchy 层级结构不变；
- **Clear Top Module:** 清除该 module 的 top module 设置。

图 3-2 Hierarchy 窗口右键菜单显示



如果对工程文件进行 hierarchy 解析时存在 error 信息，在 Hierarchy 窗口右上方会显示红色字体标识的“RTL Analysis Error”提示，单击该提示会弹出提示框，提示框内为具体的 Error 信息。

3.6.2 资源显示

综合完成后 Hierarchy 窗口会自动显示当前工程设计中的资源信息，如图 3-3 所示。如果设计中 module 的定义为加密模块，则不会显示它的资源信息，该加密模块的资源信息会被统计到其上层模块中。每个 module 使用的各类资源会显示两个数据，如图 3-3 中 module alttop 的 LUT 资源数据 2827 (6)，其中，6 为该 module 自身所使用的 LUT 个数，2827 为该 module 及其子 module 所使用的 LUT 的总数。

图 3-3 Hierarchy 窗口资源信息显示

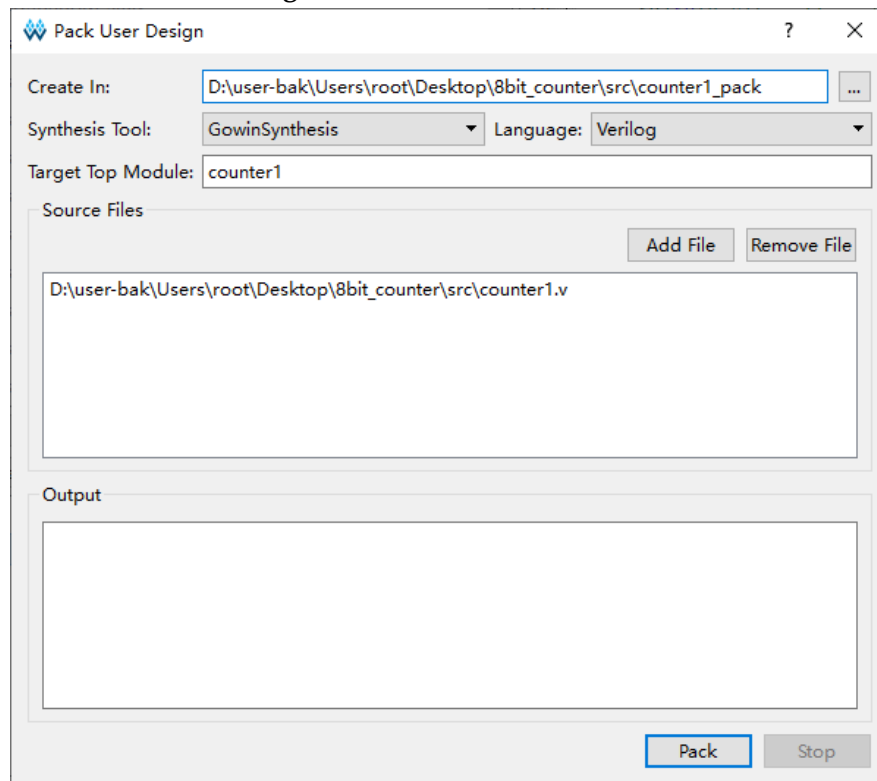
Unit	File	Register	LUT	ALU	BSRAM	SSRAM
▼ alttop	src\alttop.v	1804(0)	2827(6)	474(0)	9(0)	0(0)
rxuart(rcvuart)	src\rxuart.v	80(80)	93(93)	27(27)	0(0)	0(0)
txuart(tcvuart)	src\txuart.v	43(43)	84(84)	0(0)	0(0)	0(0)
▼ altbusmaster(slavedbus)	src\altbusmaster.v	1681(76)	2644(161)	447(0)	9(0)	0(0)
deppbyte(deppdrive)	src\deppbyte.v	51(51)	10(10)	0(0)	0(0)	0(0)
> wbusbus(busbdriver)	src\wbusbus.v	660(20)	1017(11)	186(18)	6(0)	0(0)
icontrol(pic)	src\icontrol.v	27(27)	34(34)	0(0)	0(0)	0(0)
ziptimer(zipt_a)	src\ziptimer.v	65(65)	116(116)	0(0)	0(0)	0(0)
ziptimer(zipt_b)	src\ziptimer.v	33(33)	84(84)	0(0)	0(0)	0(0)
rtlight(thetime)	src\rtlight.v	153(153)	145(145)	90(90)	0(0)	0(0)
wbpwmaudio(theaudio)	src\wbpwmaudio.v	66(66)	11(11)	44(44)	0(0)	0(0)
spio(thesprio)	src\spio.v	22(22)	11(11)	0(0)	0(0)	0(0)
wbgpio(thegpio)	src\wbgpio.v	49(49)	17(17)	16(16)	0(0)	0(0)
▼ wbspiflashp(flashmem)	src\wbspiflashp.v	272(156)	760(572)	30(24)	0(0)	0(0)
llqspi(lldriver)	src\llqspi.v	116(116)	188(188)	6(6)	0(0)	0(0)
▼ wbcap6(fpga_cfg)	src\wbcap6.v	107(68)	195(150)	15(15)	1(0)	0(0)
wbicapesimple(spartancfg)	src\wbicapesimple_G.v	39(39)	45(45)	0(0)	1(1)	0(0)
wbscope(wbcfgscope)	src\wbscope.v	100(100)	83(83)	66(66)	2(2)	0(0)

3.6.3 文件加密

在打开工程的情况下，当用户需要对 FPGA 设计中全部源文件或者部分源文件进行加密操作时，可以在 Hierarchy 窗口中右击所要加密的

module，通过右键菜单中的“Pack User Design”生成综合后的加密文件。“Pack User Design”对话框如图 3-4 所示。

图 3-4 Pack User Design 对话框



Pack User Design 对话框各项配置含义如下：

- **Create In:** 生成加密文件的目标路径，仅支持绝对路径的形式，默认为工程路径\src<topmodule_name>_pack；
- **Language:** 选择硬件语言，下拉列表支持 Verilog 和 VHDL，默认选择 Verilog；
- **Target Top Module:** 想要进行加密的 top module。默认为在 Hierarchy 窗口选择的 module，允许用户进行修改；
- **Source Files:** 列出 Hierarchy 窗口选择的 module 及其 sub module 所在的源文件；
- **Add File:** 添加待加密的设计文件；
- **Remove File:** 移除无需加密的设计文件；
- **Output 窗口:** 执行信息窗口；
- **Pack:** 执行加密；
- **Stop:** 终止加密。

开始加密及加密成功后会在 Output 窗口打印相关信息，如图 3-5 所示。加密过程中如果有 error 信息，会把 error 信息显示到 Output 窗口，同时会打印加密失败的信息，如图 3-6 所示。

图 3-5 Pack User Design 对话框打印信息

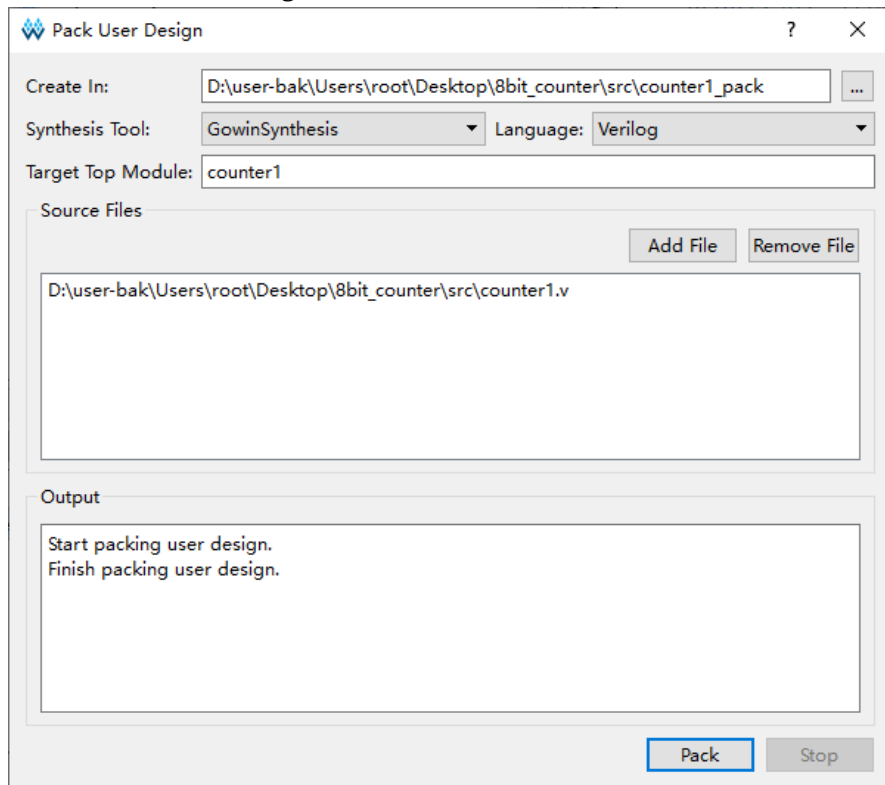
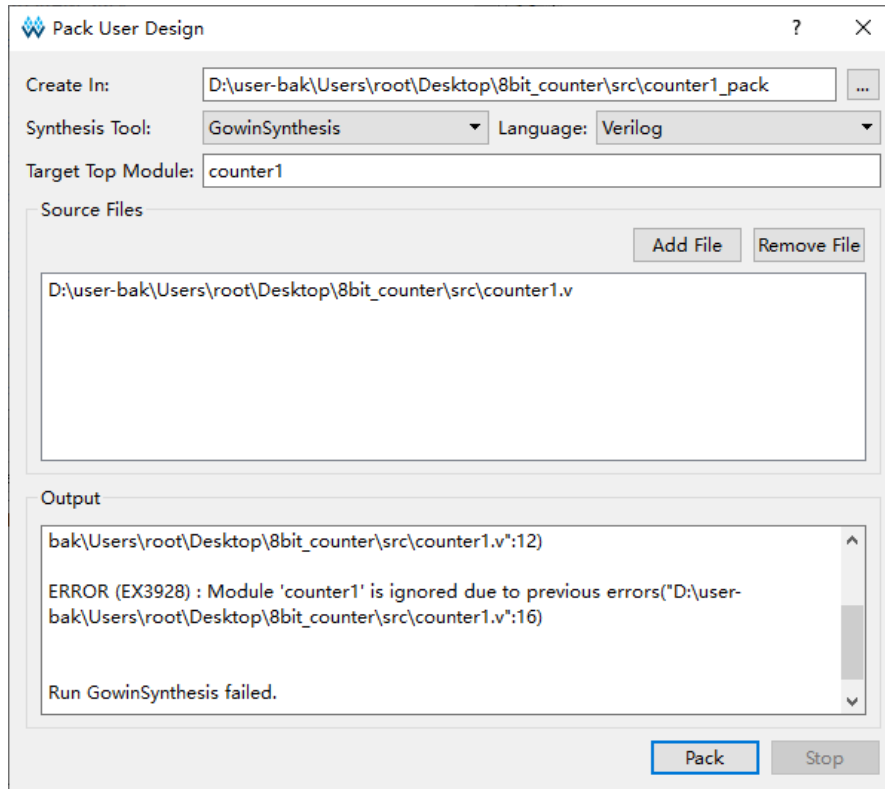


图 3-6 Pack User Design 对话框提取错误信息



加密完成后会在目标路径下生成两个文件，如果 Language 选择的是 Verilog，则生成的文件为<topmodule_name>_gowin.vp 和

<topmodule_name>_sim.v, 如果 Language 选择的是 VHDL, 则生成的文件为<topmodule_name>_gowin.vhdp 和<topmodule_name>_sim.v。
<topmodule_name>_gowin.vp 和<topmodule_name>_gowin.vhdp 是用于保密的加密文件, 可以提供给他人使用。<topmodule_name>_sim.v 是打平的综合后明文网表文件, 可以用于加密模块的功能仿真。

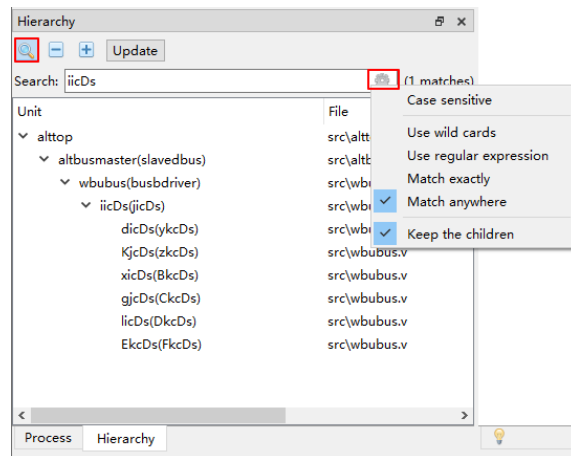
注!

如果工程中有多个 module 都实例化了同一个 sub module, 分别 pack 这几个 module 后产生的文件中都会有该 sub module 的定义, 如果将产生的这几个文件用于同一个工程, 会报出 sub module 被重复定义的错误, 该使用方式需要规避。

3.6.4 搜索功能

在打开工程且工程设计文件较多的情况下, 单击 Hierarchy 窗口中的搜索图标可以按照 module name 或者 file name 进行搜索。搜索时有多种选项可供选择, 如使用通配符、精准匹配等, 对话框如图 3-7 所示。

图 3-7 Hierarchy 窗口搜索功能



3.7 源文件编辑区

提供基本的文件编辑、查看及选中关键词高亮功能。

新建或打开的文件、综合后生成的文件以及 Place & Route 后生成的文件均可显示在文本编辑区, 另外“Start Page”及工程的“Design Summary”同样会显示在文本编辑区。

如文件显示在文本编辑区内, 在外部对文件进行了修改操作, 则在文本编辑区内会弹出“File Changed”提示, 选择“Reload”则会重新加载该文件。

单击“File”菜单栏中“Close”选项或文本编辑区当前显示文件名右侧“✕”, 会关闭文本编辑区内当前显示的文件。

单击“File”菜单栏中“Close All”选项, 则会关闭文本编辑区内显示的所有文件。

打开某个文件后, 可以通过快捷键 Ctrl+F 或单击工具栏中的 Find & Replace 来打开“Find & Replace”对话框。对话框中的 Find All 选项可以

选择三个查找范围：Current File、Open Files 以及 Current Project，如图 3-8 所示。单击 Find All 后，软件下方会弹出“Search Result”窗口，搜索内容会高亮显示，匹配结果的总数会显示在首行的行尾处，如图 3-9 所示。

图 3-8 Find & Replace 对话框

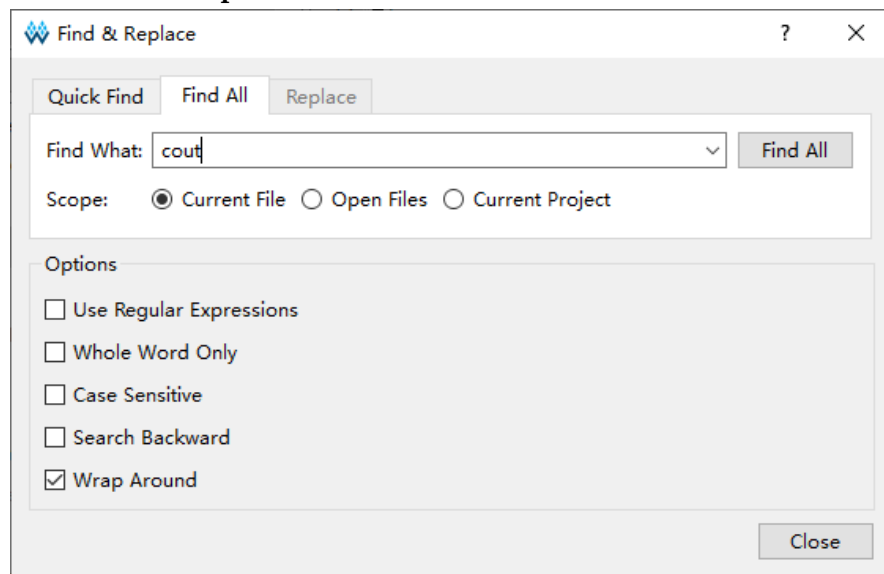
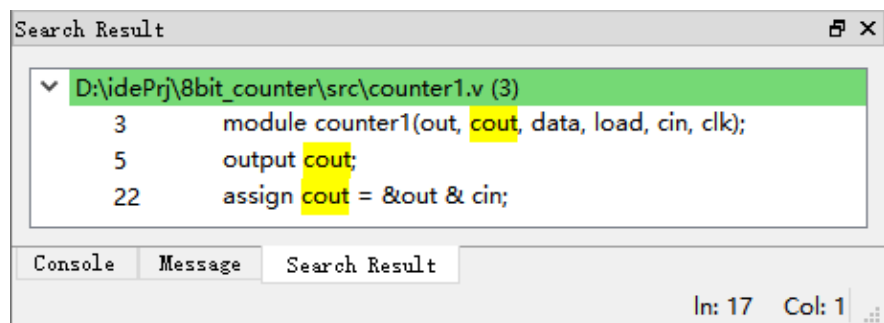


图 3-9 Search Result 窗口



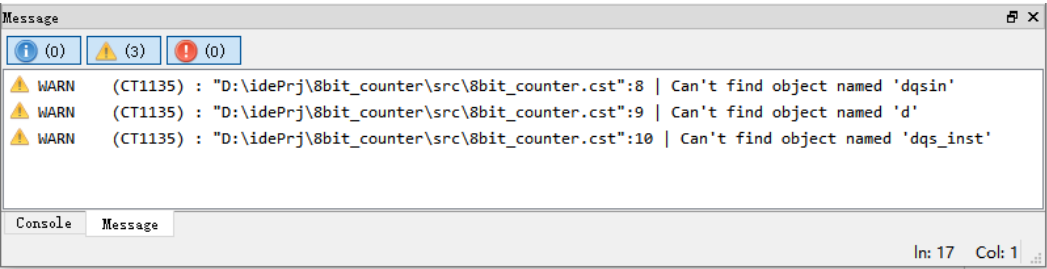
3.8 信息输出区

显示软件在运行过程中的处理信息，可手动切换标签页查看不同类型的输出信息：

- 全部信息（Console 窗口），包括 Tcl 命令窗口、警告信息、错误信息等；
- 信息汇总（Message 窗口），包括 Note 信息、警告信息、错误信息。

在 Console 窗口右击选择“Clear”，可清空所有页的信息。Message 窗口包含 Note、Warning 和 Error 三个子选项，可控制 Message 窗口只显示 Note 信息、Warning 信息或 Error 信息，每个子选项会在相关的信息页标签处记录出现信息的个数，如图 3-10 所示。在 Message 窗口右击选择“Clear”，可清空当前页面的信息。

图 3-10 信息输出窗口



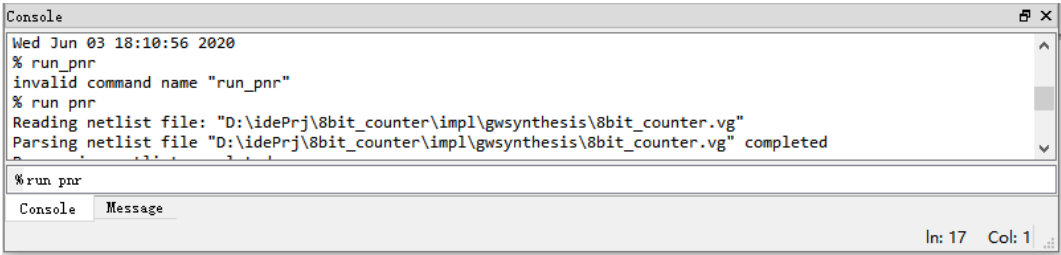
选中 PnR 报出的某条 Error 或 Warning 信息后右击选择“Help”，或者按快捷键“F1”，会弹出该条 Error 或 Warning 的“GOWIN Help”帮助文档，文档中会详细描述该条 Error 或 Warning 的帮助信息。一些常见的 Warning 或 Error 信息如表 3-1 所示，详细的帮助文档可通过菜单栏 Help > View Help 查看，该帮助文档支持中英文显示。

表 3-1 常见 Warning 和 Error 信息

名称	代码	描述
Warning	WARN (PA1002): <file>:<line> Invalid parameterized value <value>(<parameter>) specified for instance <instanceName>	指定位置处的器件设置了错误的参数值。
	WARN (PA1008): <file>:<line> Object <name> is already defined	存在连线或接口的重复定义。
	WARN (PA1001) : Dangling net <netName>(source:<instanceName>) in module <moduleName> has no destination	指定模块中的连线没有连接目标。
	WARN (CT1098) : <file>:<line> Group name <name> is already defined	约束文件中存在约束组的重复定义。
	WARN (CT1101) : <file>:<line> Location column <number> is out of chip range(<maxColumn>)	约束位置信息中的列超出了芯片的范围。
Error	ERROR (PA2000): <file>:<line> Syntax error near token <name>	指定位置处存在语法错误。
	ERROR (PA2001): <file>:<line> Module <moduleName> is already defined	存在模块的重复定义。
	ERROR (PA2017): The number(<value>) of <instType> in the design exceeds the resource limit(<maxValue>) of current device	设计文件中模块的数量超出了该器件的总数。
	ERROR (PA2025): No <instType> resource in current device	设计文件中含有本芯片不支持的器件资源。
	ERROR (PA2054): <file>:<line> <name> is already declared	设计文件中存在器件名称重复。

在 Console 信息输出窗口的最下方是 Tcl 命令编辑窗口，可以在窗口中输入 Tcl 命令后通过回车键来执行相关命令，如图 3-11 所示。详细的 Tcl 命令使用方法见本文档 [8 Tcl 命令说明](#)。

图 3-11 Tcl 编辑窗口



4 云源使用

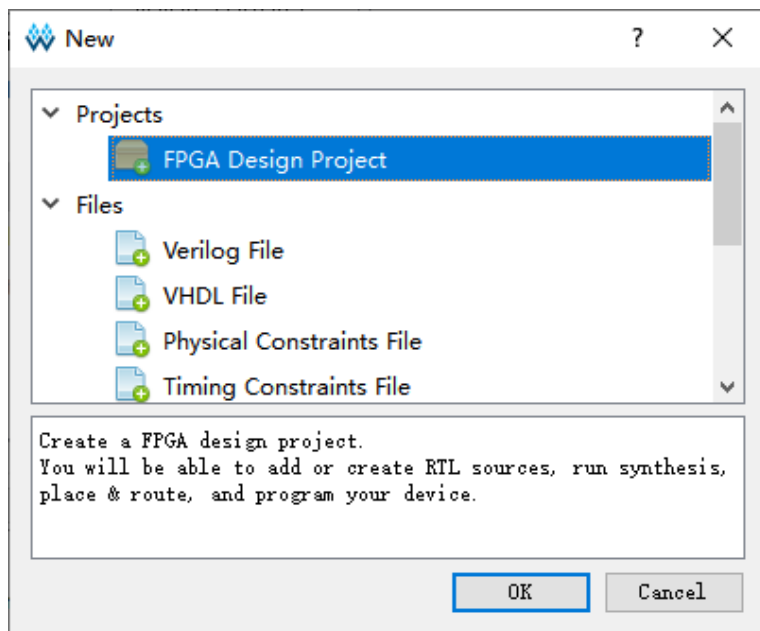
云源支持界面模式和命令行模式运行，命令行模式的使用可参考本文档的 8Tcl 命令说明。

以 Windows10 版本云源界面运行为例，介绍云源的使用方法。

4.1 新建工程

1. 单击“File > New...”，打开“New”对话框，如图 4-1 所示。

图 4-1 新建工程



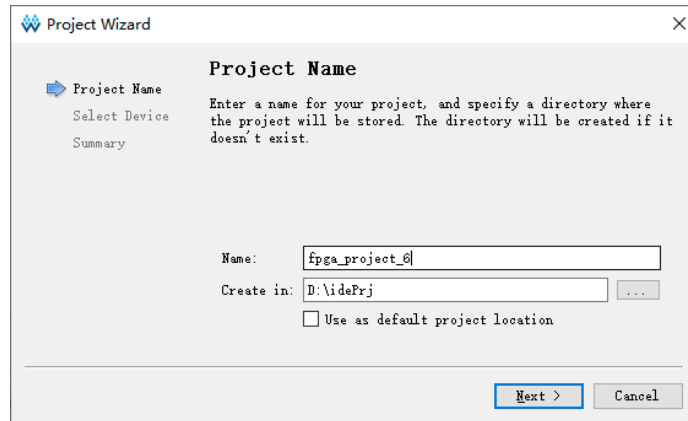
注！

也可通过以下三种方式打开“New”对话框：

- 使用快捷键 **Ctrl+N**；
- 单击工具栏上“New File or Project”图标；
- 单击 Start Page 窗口上“Quick Start > New Project”。

2. 打开“New”对话框后默认已选择“FPGA Design Project”，单击“OK”，打开“Project Wizard”对话框，如图 4-2 所示。

图 4-2 新建工程向导



3. 创建工程名和路径，如图 4-2 所示：

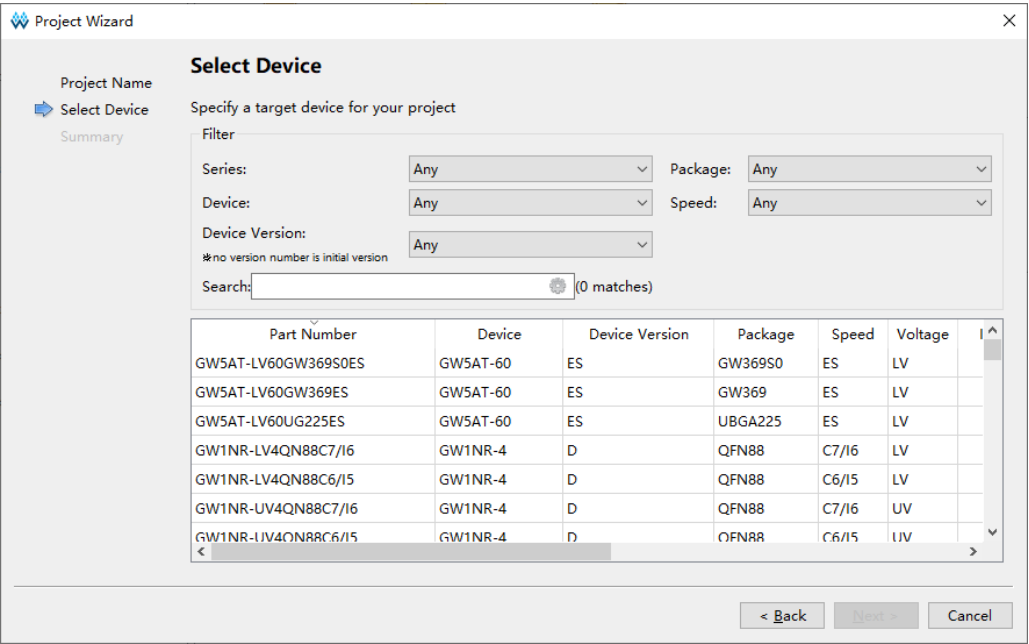
- a) 在“Name”文本框中输入工程名；
- b) 单击“...”选择工程路径。

若选中“Use as default project location”，即将该工程路径设置为默认路径，下次新建工程时会默认创建在该路径。

注！

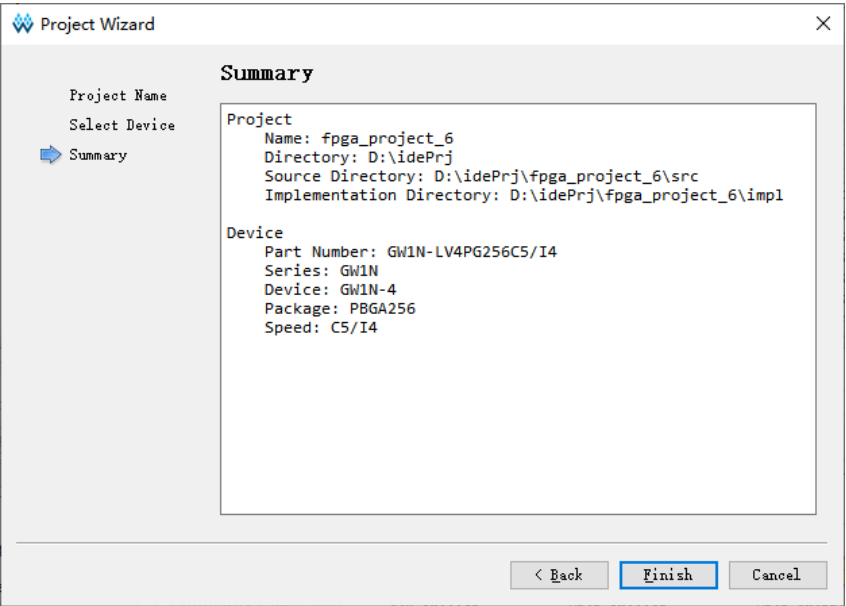
- Windows 和 Linux 下的文件路径长度均有限制，Windows 系统路径长度限制 260 个字符，Linux 系统限制 4096 个字符。在路径下存在文件长度超出系统限制的情况下，删除或者拷贝路径都会失败；
 - 与 Linux 不同，在 Windows 中路径分隔符为“\”，如 E:\Gowin\ide。
4. 单击“Next”，设置 FPGA 的器件信息，共有五个过滤选项 Series、Device、Package、Speed 和 Device Version，也可通过“Search”对器件进行搜索。
- 在 Series 中过滤芯片系列
 - 在 Device 中过滤器件
 - 在 Package 中过滤封装类型
 - 在 Speed 中过滤速度等级
 - 在 Device Version 中过滤器件的版本
 - 在“Part Number”栏中选择芯片型号，并且该栏中会显示所选择器件的详细资源信息。对于没有 Device Version 的器件，版本信息栏显示为空，对于同一器件的 Device Version 以倒叙显示，如图 4-3 所示。

图 4-3 设置 FPGA 器件信息



5. 单击“Next”，在 Summary 中核对新建 FPGA 工程的工程信息及器件信息，如图 4-4 所示。

图 4-4 工程信息



6. 单击“Finish”，工程创建完成。

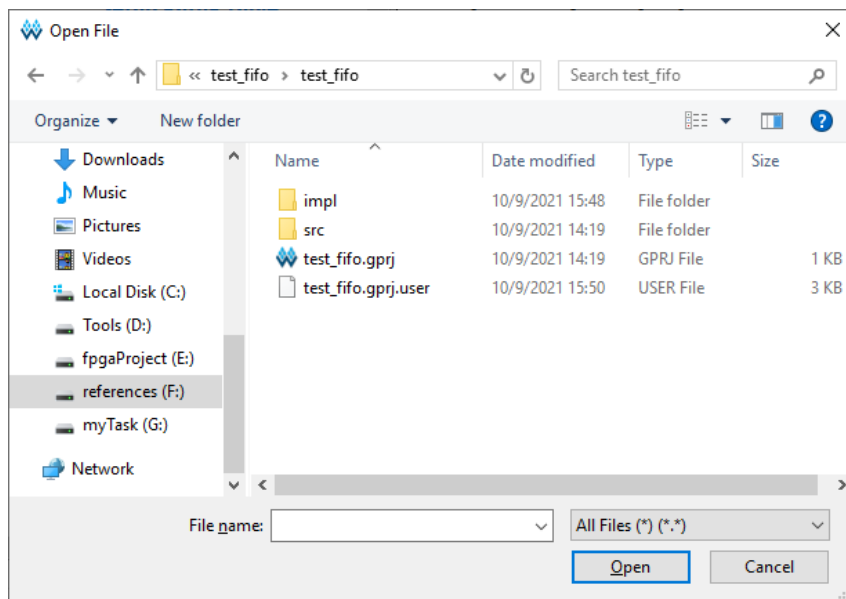
4.2 打开工程

用户可直接通过云源界面打开已创建的高云软件工程，有以下五种方式打开工程：

工具栏打开

1. 单击工具栏上的“”图标，打开“Open File”对话框，如图 4-5 所示；
2. 选择工程文件*.gprj，单击“Open”，打开工程。

图 4-5 打开文件



菜单栏打开

1. 菜单栏中，选择“File > Open ...”打开“Open File”对话框，如图 4-5 所示；
2. 选择工程文件*.gprj，单击“Open”，打开工程。

Start Page 打开

1. 在“Start Page”页面中，单击“”，打开“Open Project”对话框；
2. 选择工程文件*.gprj，单击“Open”，打开工程。

Recent Projects 打开

菜单栏中，单击“File > Recent Projects”，选择需打开的工程。

注！

- 亦可在“Start Page > Recent Projects”列表中，选择需打开的工程；
- “Recent Projects”为最近打开过的工程列表，最多会保留 10 个工程；
- 若该工程已被删除，会弹出“Open Project”提示框，提示找不到指定路径。

工程文件打开

双击工程文件*.gprj，自动使用最近一次安装的云源打开该工程。

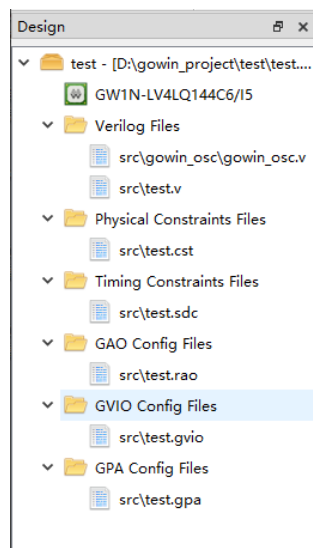
4.3 编辑工程

新建或打开工程后，可在工程管理窗口对 FPGA 工程器件信息及相关文件等进行编辑，如图 4-6 所示。

工程管理区主要包含以下信息：

- FPGA 工程的路径信息；
- 使用的芯片型号；
- 当前工程包含的文件信息，包括用户设计文件（Source Files）、约束文件和配置文件。其中，约束文件包括物理约束文件（.cst）、时序约束文件（.sdc），配置文件包括 GAO 配置文件（.gao、.rao）、功耗分析配置文件（.gpa）、虚拟输入输出配置文件（.gvio）等。

图 4-6 工程文件窗口



4.3.1 编辑工程器件

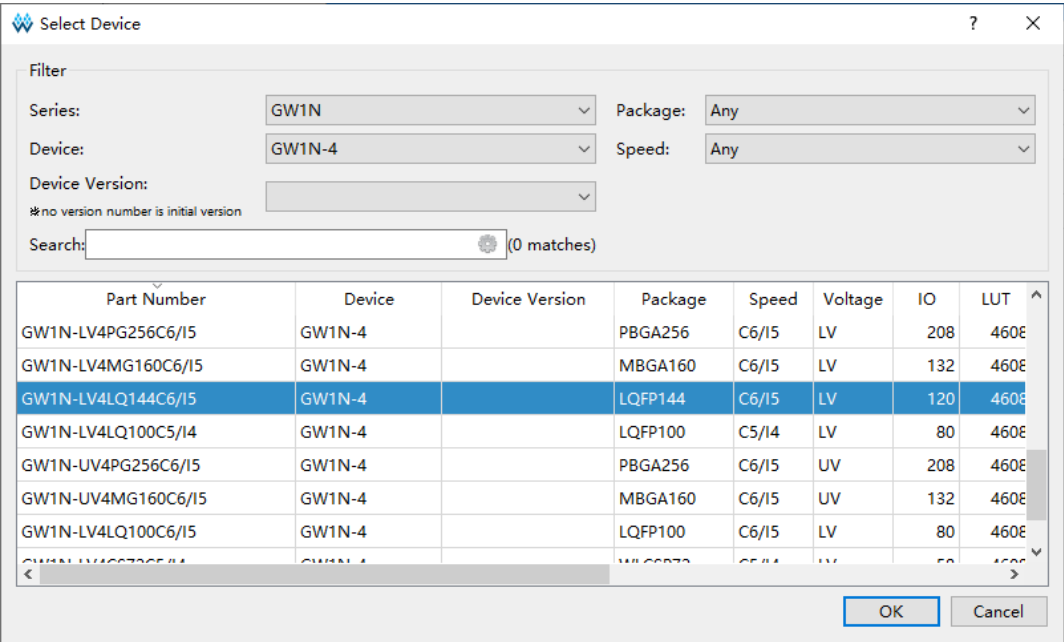
可在工程管理窗口内编辑当前 FPGA 工程所使用的芯片型号：

1. 如图 4-6 所示，双击“GW1N-LV4LQ144C6/I5”，打开“Select Device”对话框，也可以通过菜单栏 Project 下拉列表中的 Set Device 项打开，如图 4-7 所示；
2. 在“Select Device”对话框中设置 FPGA 的器件信息，在“Part Number”栏中选择芯片型号，即可编辑当前工程所用的器件。“Part Number”栏中会显示所选芯片的详细信息，包括器件信息、器件的版本、封装信息、速度等级、内核电压、芯片包含的 IO/LUT/FF/SSRAM/BSRAM/User Flash/DSP/PLL 资源数量。

注！

器件的版本信息 **Device Version** 为空表示是初始版本。在芯片的 **date code** 后面会有器件版本信息的标识，“**Part Number**”栏中所选的器件需要与使用的芯片器件版本保持一致。

图 4-7 工程配置芯片型号



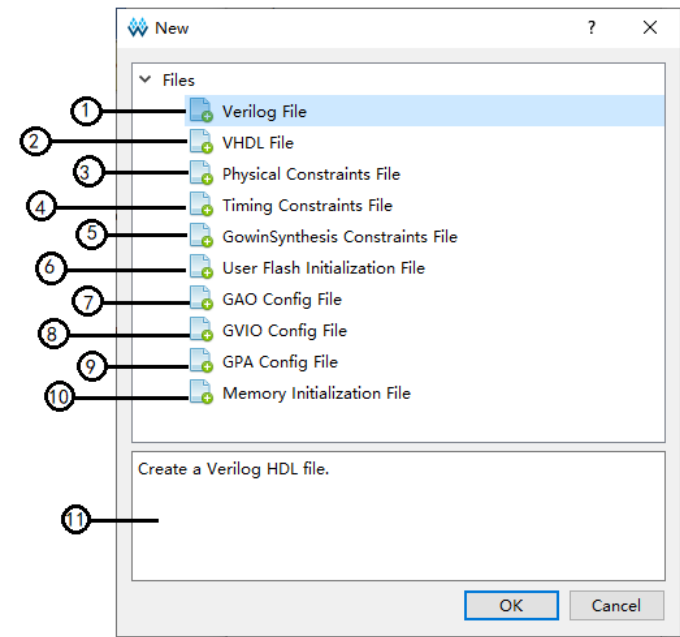
4.3.2 编辑工程文件

工程中需添加的文件类型包括用户 **RTL** 设计文件（**Source Files**）、约束文件和配置文件。下面介绍如何编辑工程所需文件。

新建设计及约束文件

- 1. 单击工具栏中的新建工程或文件图标“”，或通过菜单栏 **File** 下拉列表中的“**New...**”项打开 **New** 对话框；
- 2. 选择需新建的文件类型，如图 4-8 所示；

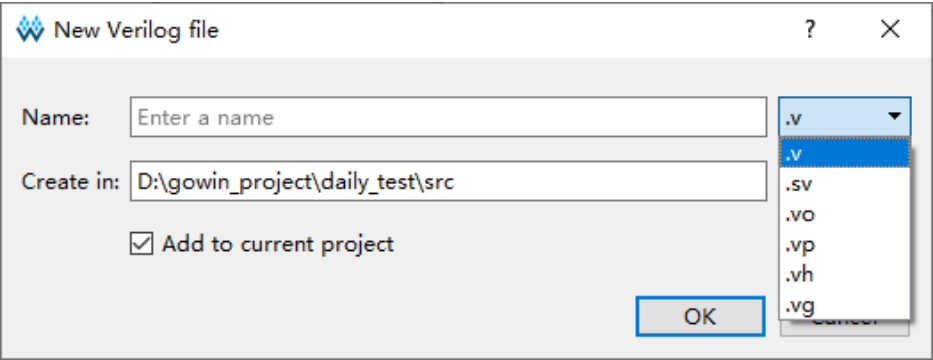
图 4-8 新建文件对话框



- | | |
|-------------------|--------------------|
| ① 用户 Verilog 设计文件 | ② 用户 VHDL 设计文件 |
| ③ 物理约束文件 | ④ 时序约束文件 |
| ⑤ 综合优化约束文件 | ⑥ User Flash 初始化文件 |
| ⑦ 在线逻辑分析仪配置文件 | ⑧ 虚拟输入输出配置文件 |
| ⑨ 功耗分析配置文件 | ⑩ 存储器初始化文件 |
| ⑪ 文件类型解释区 | |

3. 以新建 Verilog File 为例，选中“Verilog File”打开新建 Verilog File 对话框，可选择文件扩展格式，如图 4-9 所示。默认勾选“Add to current project”，即新建的设计文件默认会加到当前工程中；

图 4-9 新建 Verilog File 对话框



4. 填写文件名，单击“OK”。

新建配置文件

1. 单击工具栏中的新建工程或文件图标“”，或通过菜单栏 File 下拉列表中的“New...”项打开 New 对话框，如图 4-8 所示；

2. 在图 4-8 中，选择需新建的配置文件类型。以新建功耗分析配置文件为例，选中“GPA Config File”打开新建 New GPA Config File 对话框，如图 4-10 所示，输入文件名，单击“OK”，新建的 GPA 配置文件会自动添加到工程管理区；
3. 在工程管理区双击配置文件，以窗口形式打开空白配置文件，进行编辑，如图 4-11 所示。

图 4-10 新建配置文件对话框

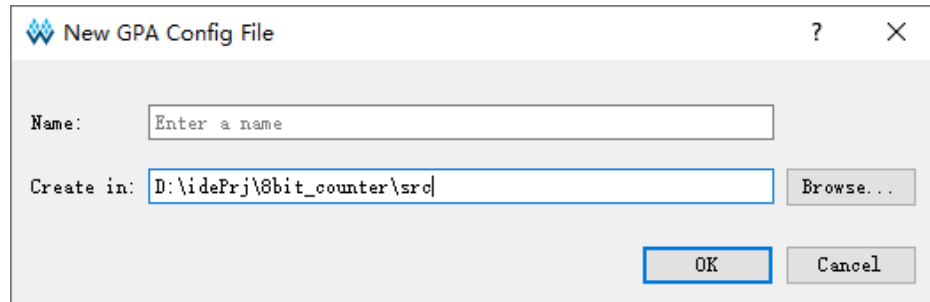
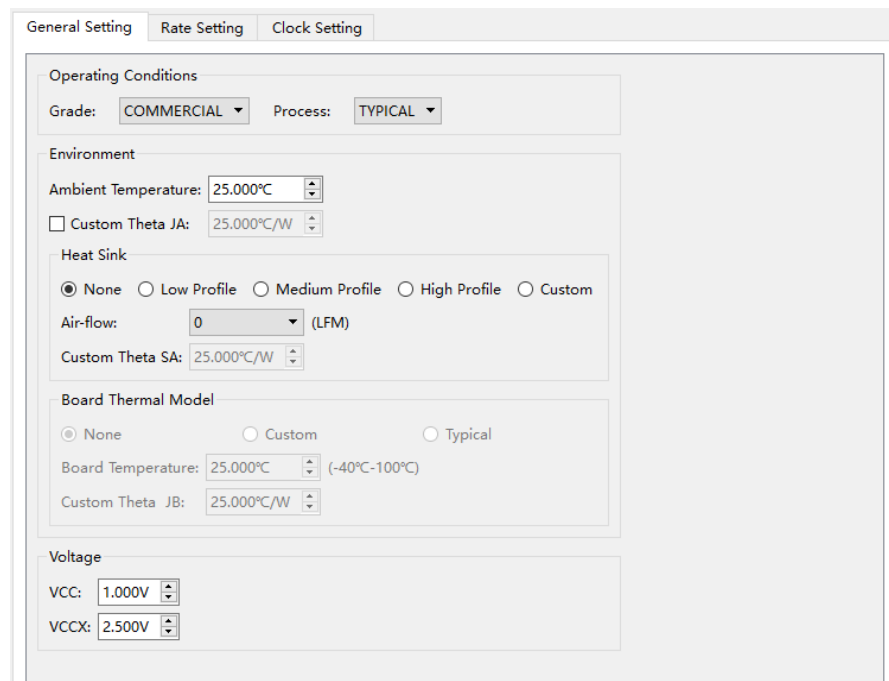


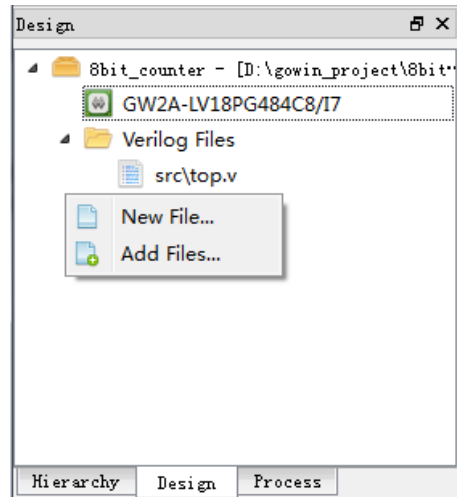
图 4-11 GPA 配置文件窗口



添加工程文件

1. 如图 4-12 所示，在工程管理区空白处右击选择“Add Files...”，打开“Select Files”对话框；
2. 选择工程文件，可同时选择多个文件或单个文件添加，完成添加用户设计文件。添加后，软件会自动在工程文件管理区将文件进行分类，如果添加的文件不是 RTL 设计文件、网表文件、约束文件、GPA 配置文件、GAO 配置文件，则在工程管理区会增加一个文件分类“Other Files”。

图 4-12 Design 窗口右键菜单

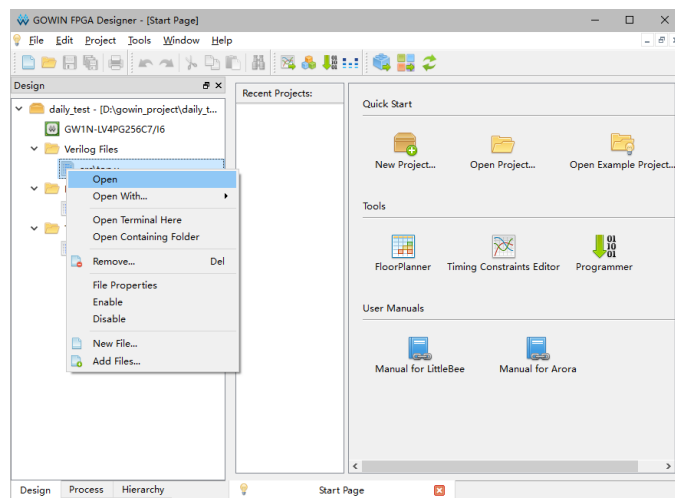


修改工程文件

如图 4-13 所示，可通过以下两种方式打开文件：

1. 双击工程管理区内的任一文件，文件即会显示在源文件编辑区；
2. 右击需要修改的文件，单击“Open”。

图 4-13 工程文件编辑菜单

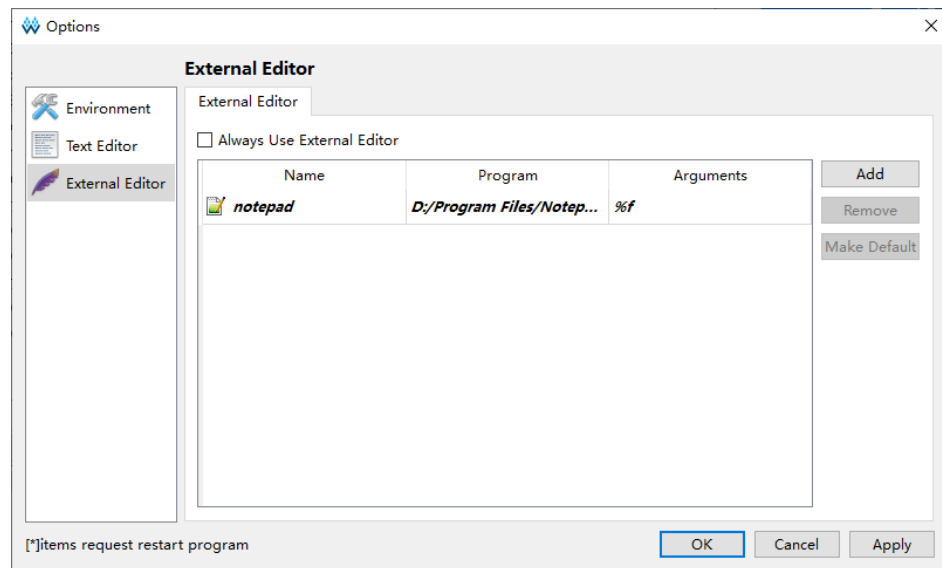


如果用户已经通过菜单栏 **Tools > Options** 配置了第三方文本编辑软件，选择“Open With...”可以用第三方的文本编辑软件打开设计文件。如果选择的是“Add External Editor”，用户打开 **Options** 对话框添加其他外部编辑器，如图 4-14 所示。如果勾选“Always Use External Editor”，则会始终使用外部编辑器打开设计文件。在右键菜单中选择“Open Containing Folder”可打开文件所在的文件夹，选择“Open Terminal Here”可打开命令行运行窗口，用户可通过命令行模式运行。

如果用户使用外部编辑器对已经在云源编辑区打开的文件进行修改并保存，云源会弹出文件已变更是否重新加载的提示。

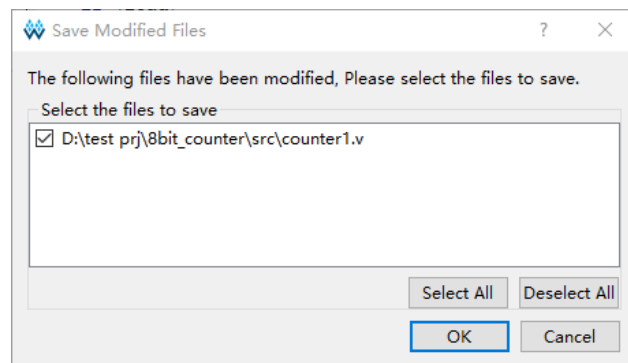
如果用户关闭编辑后未保存的文件，云源会弹出是否保存更改文件的提示框。

图 4-14 外部编辑器



在工程文件修改后，如果忘记保存直接执行 **Synthesize** 或 **Place & Route**，会弹出“**Save Modified Files**”的对话框，如图 4-15 所示。

图 4-15 Save Modified Files 对话框



单击“OK”后，文件会被保存且自动执行弹出对话框之前想要执行的流程。如果单击“Cancel”，则文件不会被保存，且不会再执行 **Synthesize** 或 **Place & Route**。

删除工程文件

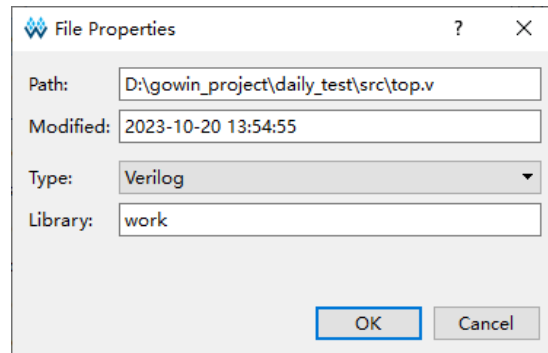
1. 选中工程管理区内的文件；
2. 右击选择“**Remove**”或直接按键盘“**Delete**”键，弹出“**Remove Files**”对话框，若选中“**Remove Permanently on Disk**”复选框，则该文件从当前工程删除且在磁盘上删除，否则该文件仅从当前工程删除。

编辑工程文件属性

在工程管理区内的任一文件处右击选择“**File Properties**”，会弹出该文

件的文件属性对话框，如图 4-16 所示。对话框中包含该文件的路径信息、修改时间、Type 以及 Library 信息。通过 Type 下拉菜单可以选择编辑文件的 Type 信息，单击“OK”后，该文件会在 Design 窗口中自动移动到所选的 Type 类型下。Library 用于指定综合 VHDL 文件时所用到的编译库，默认为 work，如果有多个库，需要用分号分隔。

图 4-16 工程文件属性对话框



如果选中了多个 Verilog 文件或多个 VHDL 文件，则“File Properties”对话框中不会显示路径信息和修改时间。如果既选中了 Verilog 文件也选中了 VHDL 文件，则“File Properties”对话框中不会显示路径信息、修改时间和 Type。

Library 的使用方法如下：

- 设计中顶层（或上一层）实体有底层实体的 component，则不需要关心底层实体属于哪个 library，可以采用默认值 work；
- 如果设计中顶层（或上一层）实体采用“uut1:entity 库名.底层实体名称”的形式调用底层实体（如 uut1:entity mb.sub1），则底层实体所在的 vhd 文件属性 library 应该为库名（如 mb）；
- 如果程序包中有底层实体的 component，顶层（或上一层）实体通过程序包调用底层实体时，不需要关心底层实体属于哪个 library，可以采用默认值 work；
- 如果程序包中有底层实体的 component，顶层（或上一层）实体采用“uut: 程序包所属库名.程序包名称.底层实体名称”的形式调用底层实体（如 uut1:work.pack.sub1）时，则不需要关心底层实体属于哪个 library，可以采用默认值 work。

使能工程文件

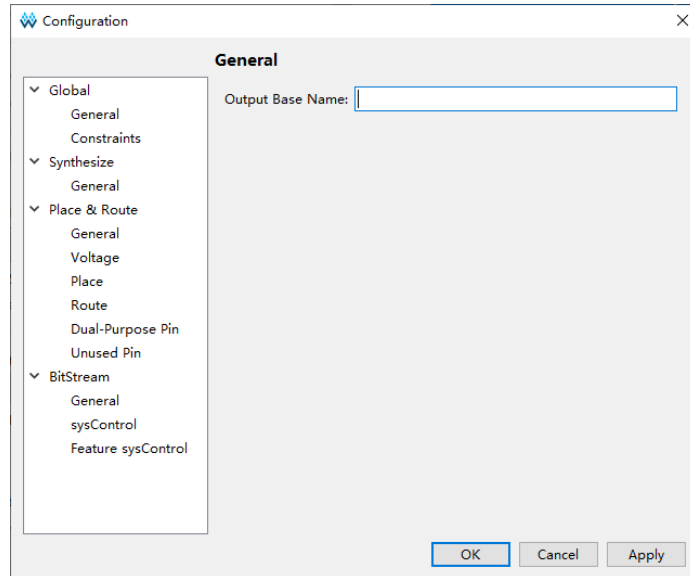
在工程管理区内的任一文件处右击，弹出的右键菜单中有“Enable”和“Disable”选项，如图 4-13。文件被 Enable 时，参与工程编译过程，被 Disable 时不参与工程编译过程。

1. 通过右键菜单选项设置文件的 Enable/Disable（包括单独文件的设置及文件的批量设置）；
2. 对于同一种约束类文件或配置类文件，工程中只有一个会处于 Enable 状态，再次新建或添加时，会将前一个约束文件或配置文件 Disable。

4.3.3 编辑工程配置

在工程管理区内的 Synthesize 处或 Place & Route 处右击选择“Configuration”，打开工程配置选项对话框，如图 4-17 所示。

图 4-17 工程配置选项对话框



如图 4-17 所示，可配置的工程选项包含“Global”、“Synthesize”、“Place & Route”和“BitStream”。有关可配置的各个工程选项的详细介绍如下：

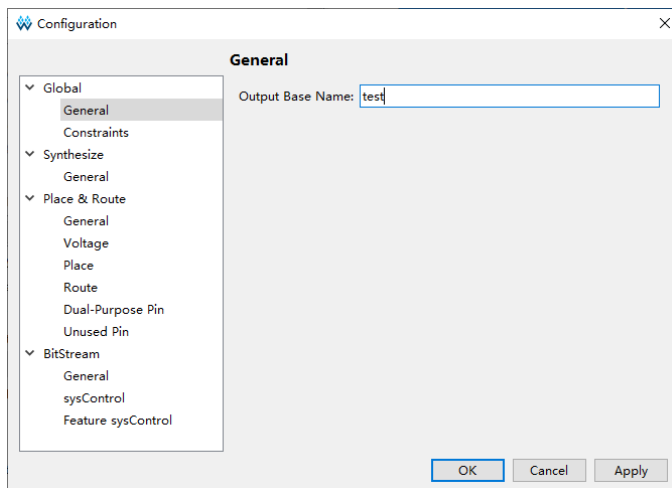
Global

Global 选项包括 General 和 Constraints。

General

General 选项配置如图 4-18 所示，指定输出文件的基本名称，默认为当前工程的名称。

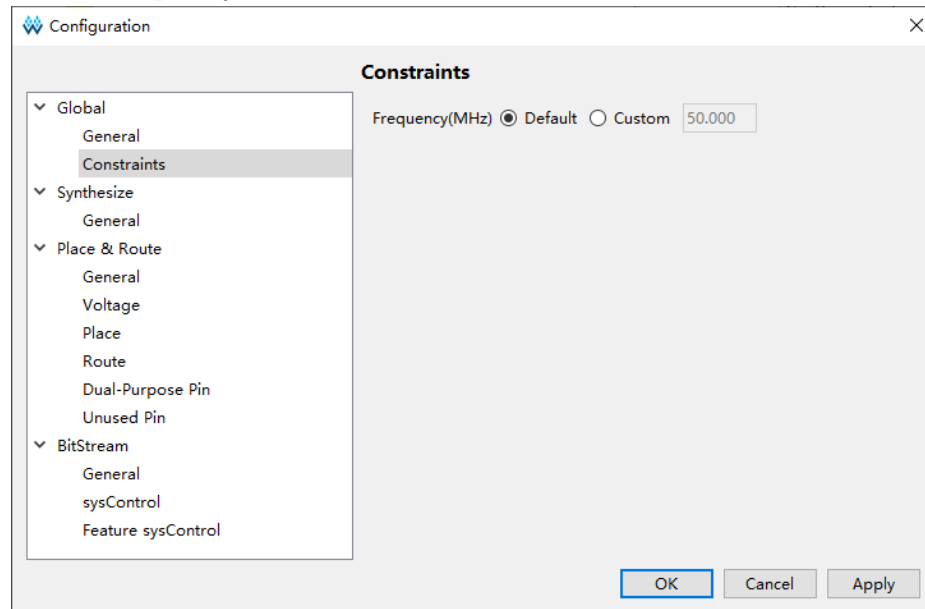
图 4-18 General 配置选项



Constraints

Constraints 选项配置如图 4-19 所示。可以通过该选项设置用户设计的全局时钟频率值，如果时序约束中指定了时钟频率，则优先使用时序约束中的频率值。默认为 **Default**（小蜜蜂家族芯片默认是 50MHz，晨熙家族芯片默认是 100MHz）。

图 4-19 Frequency(MHz)配置选项



Synthesize

General

Synthesize 的 **General** 选项配置如图 4-20 所示。**Synthesis Tool** 为 **GowinSynthesis**，可以对综合选项进行配置，将鼠标悬浮在部分选项处时会显示其解释。

图 4-20 GowinSynthesis 参数配置项

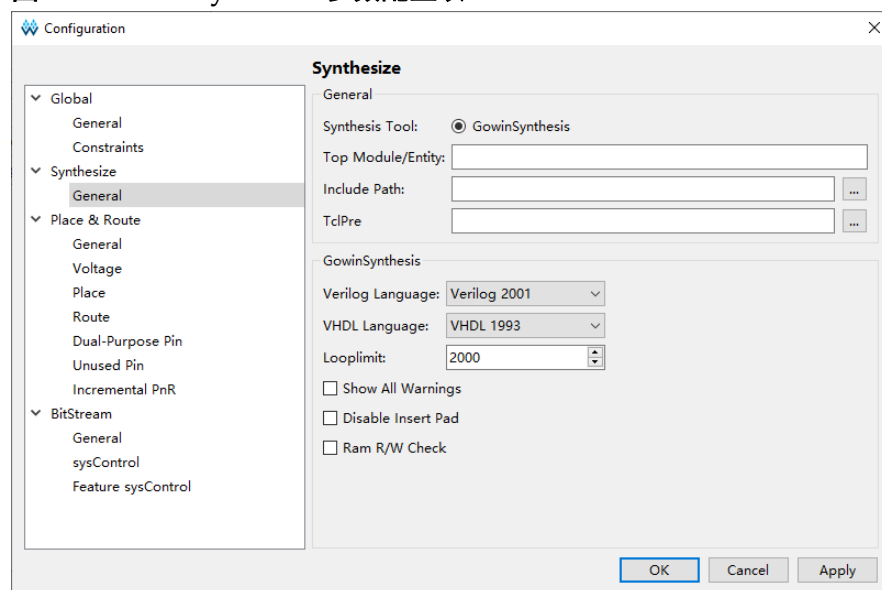


图 4-20 中的各项参数释义如下所示：

- **Top Module/Entity:** 指定顶层模块；
- **Include Path:** 指定文件包含路径；
- **TclPre:** 指定软件自动化版本管理文件，每次综合时自动更改版本号 and 版本日期，根据版本号，可以找到运行的设计版本；
- **Verilog Language:** 指定 verilog 语言，支持 System Verilog 2017、Verilog 2001 和 Verilog 95，默认值为 Verilog 2001；
- **VHDL Language:** 指定 VHDL 语言，支持 VHDL1993、VHDL 2008 和 VHDL 2019，默认值为 VHDL1993。当检测到当前工程中含有 VHDL 格式的设计文件时才会在界面中显示该选项；
- **Looplimit:** 设置 RTL 中默认的编译器循环限制值，默认值为 2000；
- **Show All Warnings:** 勾选该选项后综合过程中会打印出所有的 Warning 信息，默认不勾选。
- **Disable Insert Pad:** 综合后的网表中是否插入 I/O Buffer，默认不勾选；
- **Ram R/W Check:** 如果 RAM 存在读或写冲突，勾选该选项后会在 RAM 周围插入旁路逻辑以防止仿真不匹配，禁用该选项不会生成旁路逻辑，默认为不勾选；

注！

关于所有选项配置的详细用法，具体可参考 [SUG550. GowinSynthesis 用户指南](#)。

Place & Route

Place & Route 选项包括 General、Voltage、Place、Route、Dual-Purpose 和 Unused Pin，各选项含义如表 4-1 所示。

表 4-1 PnR 配置选项含义描述

工程选项	描述
General	用于配置运行布局布线时的基本参数
Voltage	用于设置电压 VCCX
Place	用于配置布局参数
Route	用于配置布线参数
Dual-Purpose Pin	用于配置所选器件信息中封装方式对应的 I/O 信息，主要用于配置复用管脚。
Unused Pin	用于对未使用的 GPIO 设置不同的 IO 属性

单击“Reset all to default”，当前配置页面会全部恢复为默认设置。

General

General 选项配置如图 4-21 所示。

图 4-21 配置布局布线选项

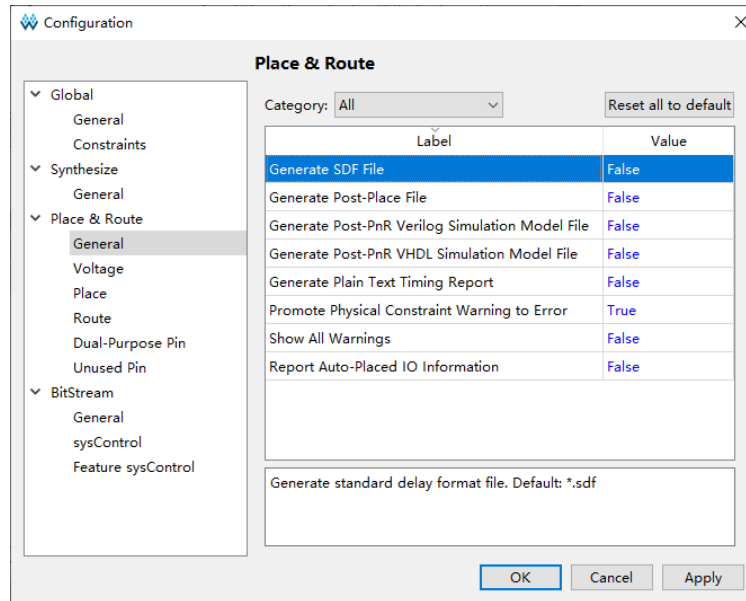


图 4-21 中的各选项使用说明如下：

- **Generate SDF File**: 产生标准延迟格式文件，扩展名为.sdf，用于布局布线后的网表时序仿真，默认值为 **False**。具体使用方法参考本文档第 7 章 仿真文件；
- **Generate IBIS File**: 产生输入/输出缓冲区信息指定文件，扩展名为.ibs，默认为 **False**。
- **Generate Post-Place File**: 产生只含有 BSRAM 布局信息的文件，扩展名为.posp，默认值为 **False**；
- **Generate Post-PNR Verilog Simulation Model File**: 产生 Verilog 语言的时序仿真模型文件用于时序仿真，扩展名为.vo，默认值为 **False**；
- **Generate Post-PNR VHDL Simulation Model File**: 产生 VHDL 语言的时序仿真模型文件用于时序仿真，扩展名为.vho，默认值为 **False**；
- **Generate Plain Text Timing Report**: 产生文本格式的时序报告，扩展名为.tr，默认值为 **False**；
- **Promote Physical Constraint Warning to Error**: 将物理约束警告提升为错误信息，默认值为 **True**；
- **Show All Warnings**: PNR 运行时输出所有的 **Warning** 信息，默认值为 **False**；
- **Report Auto-Placed IO Information**: 输出信息打印自动布局 IO 的位置信息，默认值为 **False**。

Voltage

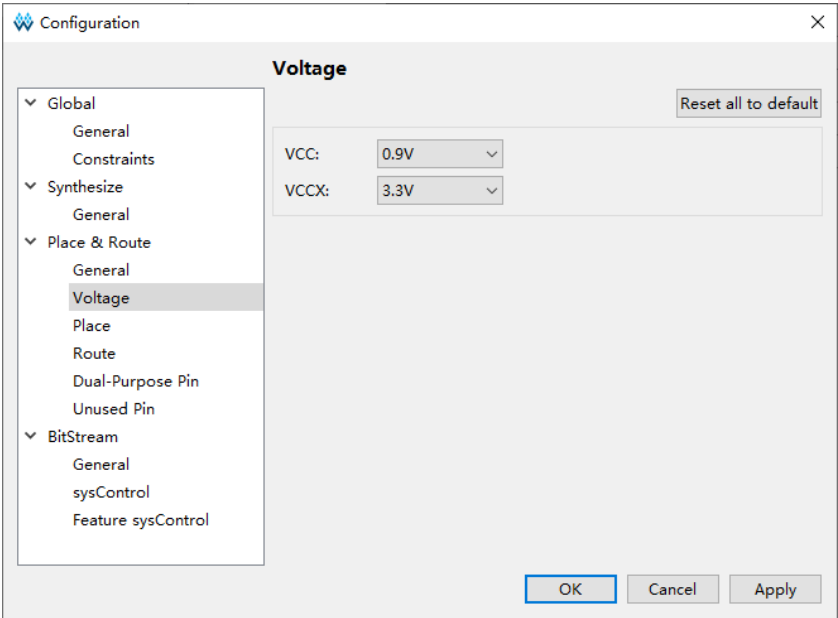
Voltage 选项配置如图 4-23 所示。可通过该选项设置电压 VCC 和 VCCX，不同器件可设置的 VCC 及 VCCX 不同，单击 “Reset all to

default”，所配置的 VCC 和 VCCX 会恢复为默认值。

注!

- VCC 的配置会影响 GW1NZ-1/GW1NZ-2 器件的延时数据。
- VCCX 的配置会影响功耗计算的结果。

图 4-22 配置 Voltage 选项



Place

Place 选项配置如图 4-23 所示。

图 4-23 配置布局选项

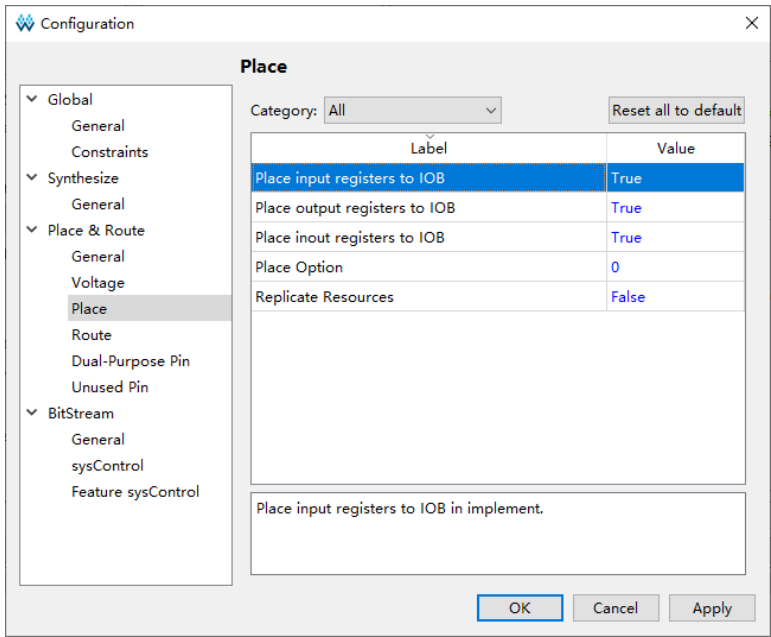


图 4-23 中的各选项使用说明如下：

- **Place input register to IOB:** 布局输入 Buffer 驱动的寄存器到 IOB 上，对于 GW5A(S)(T)-138/GW5AT-75 器件，默认值为 **False**，其他器件默认值为 **True**；
- **Place output register to IOB:** 布局输出/三态 Buffer 驱动的寄存器到 IOB 上，对于 GW5A(S)(T)-138 器件，默认值为 **False**，其他器件默认值为 **True**；
- **Place inout register to IOB:** 布局双向 Buffer 驱动的寄存器到 IOB 上，对于 GW5A(S)(T)-138 器件，默认值为 **False**，其他器件默认值为 **True**；
- **Place Option:** 布局算法选项。GW5A(N)(S)(R)(T)系列器件可选项有 0、1、2、3 和 4，对于其他系列器件可选项有 0、1、2。默认值为 0。
- **Replicate Resources:** 对高扇出的资源进行复制来降低扇出，获得更优的时序结果，默认值为 **False**。仅 GW5A(N)(S)(R)(T)系列器件支持该选项的配置，其他器件下配置界面中不会显示该配置选项。

Route

Route 选项配置如图 4-24 所示。

图 4-24 配置布线选项

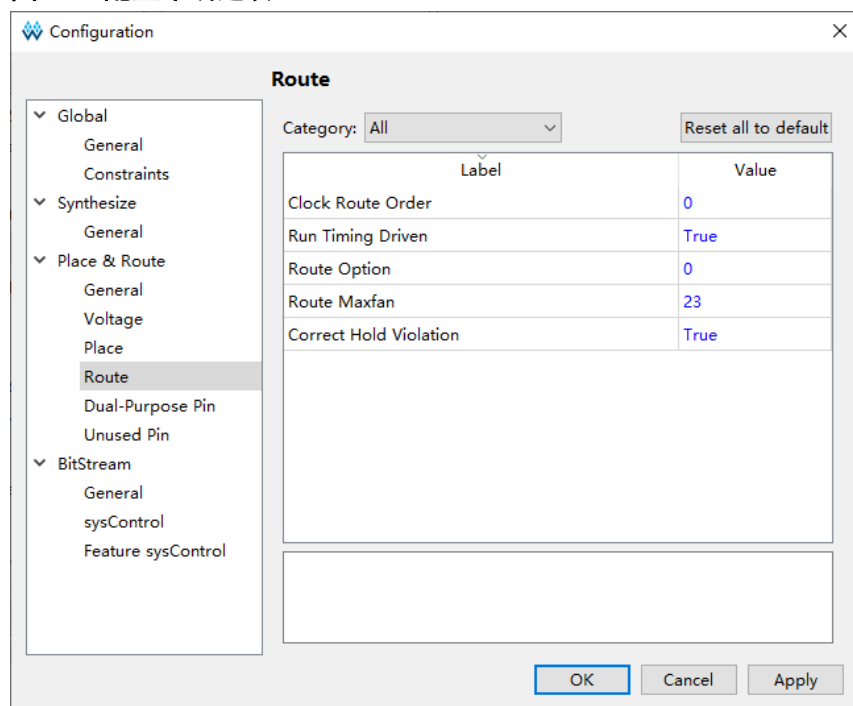


图 4-24 中的各选项使用说明如下：

- **Clock Route Order:** 指定除时钟原语产生的时钟线之外的时钟线的绕线分配顺序，可选项有 0 和 1，默认值为 0：
 - 为 0 时，根据 net 的扇出数量由多到少的顺序进行分配；
 - 为 1 时，根据频率由高到底的顺序进行分配。

- **Run Timing Driven:** 时序驱动优化布线结果，默认值为 True；
- **Route Option:** 布线算法选项，可选项有 0、1 和 2，默认值为 0：
 - 为 0 时，默认布线算法；
 - 为 1 时，牺牲编译速度来尝试找到更优的布线结果；
 - 为 2 时，提高布线速度。
- **Route Maxfan:** 基于绕线优化，设置绕线最大扇出数目，取值应为大于 0 且小于等于 100 的整数，设置的数值较小时可能会出现绕线失败的情况。该选项不会对 LW 和时钟相关的绕线进行控制。器件为 GW1NZ-1/GW1N-2/GW1NR-2/GW1N-1P5 时，Route Maxfan 默认值为 10，其他器件默认值为 23；
- **Correct Hold Violation:** 通过布线对时序 Hold 问题进行自动修复，默认值为 True。

Dual-Purpose Pin

Dual-Purpose Pin 选项是符合高云器件定制的复用管脚配置，将鼠标悬浮在选项处会显示其解释。具体配置项目如图 4-25 所示。

图 4-25 配置复用管脚选项

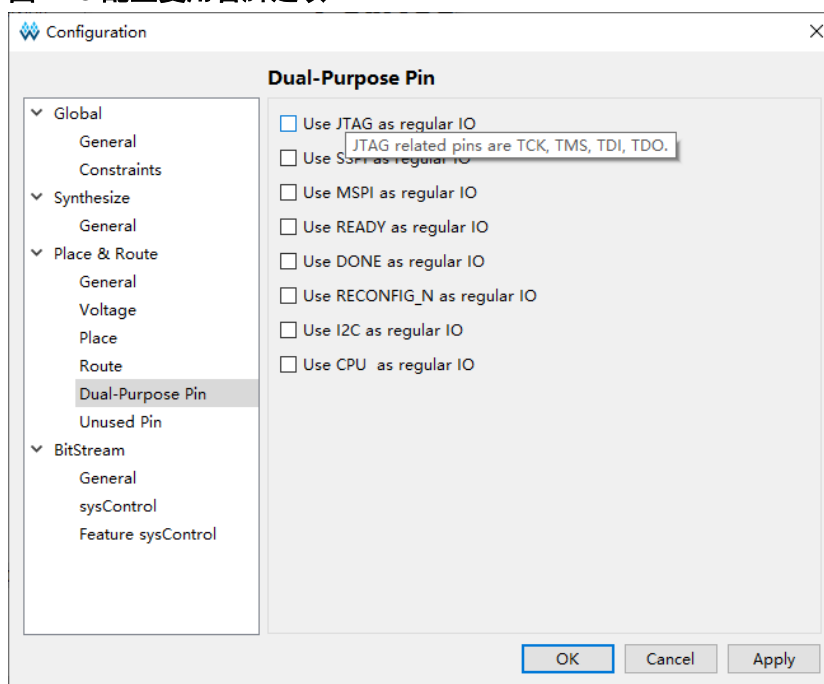


图 4-25 中的各选项使用说明如下：

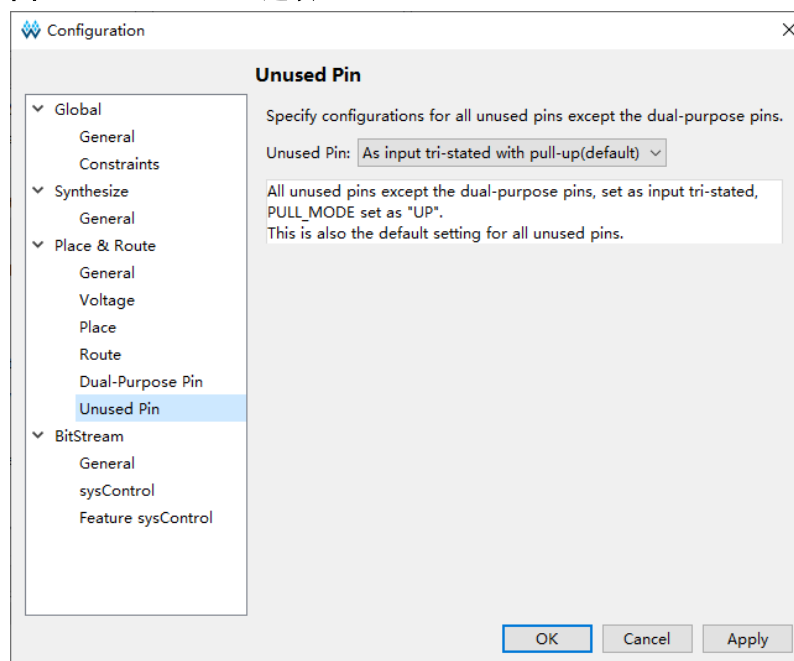
- **Use JTAG as regular IO:** 将 JTAG 相关管脚复用为普通 IO 管脚；
- **Use SSPI as regular IO:** 将 SSPI 相关管脚复用为普通 IO 管脚。对于 GW5A-25 的封装 MBGA121N，该选项默认为勾选状态且不可更改；
- **Use MSPI as regular IO:** 将 MSPI 相关管脚复用为普通 IO 管脚；
- **Use READY as regular IO:** 将 READY 相关管脚复用为普通 IO 管脚；

- Use DONE as regular IO: 将 DONE 相关管脚复用为普通 IO 管脚;
- Use RECONFIG_N as regular IO: 将 RECONFIG_N 相关管脚复用为普通 IO 管脚;
- Use I2C as regular IO: 将 I2C 相关管脚复用为普通 IO 管脚;
- Use CPU as regular IO: 仅 GW5A(N)(S)(R)(T)系列器件支持该选项配置, 将 CPU 相关管脚复用为普通 IO 管脚。

Unused Pin

Unused Pin 选项可以对未使用的 GPIO 设置不同的 IO 属性。有两个选项: As input tri-stated with pull-up (default) 、As open drain driving ground, 如图 4-26 所示。

图 4-26 Unused Pin 选项



- As input tri-stated with pull-up (default) : 默认选项, 所有未使用的 GPIO 会配置为输入三态弱上拉;
- As open drain driving ground: 所有未使用的 GPIO 会配置为输出类型, 并且 OPEN DRAIN 配置为 ON。

Incremental PnR

Incremental PnR 选项实现增量编译功能, 可以复用上一次布局或布局、布线的结果, 减少重新布局布线的时间, 提高运行效率。选项配置如图 4-27 所示。

图 4-27 Incremental PnR 选项

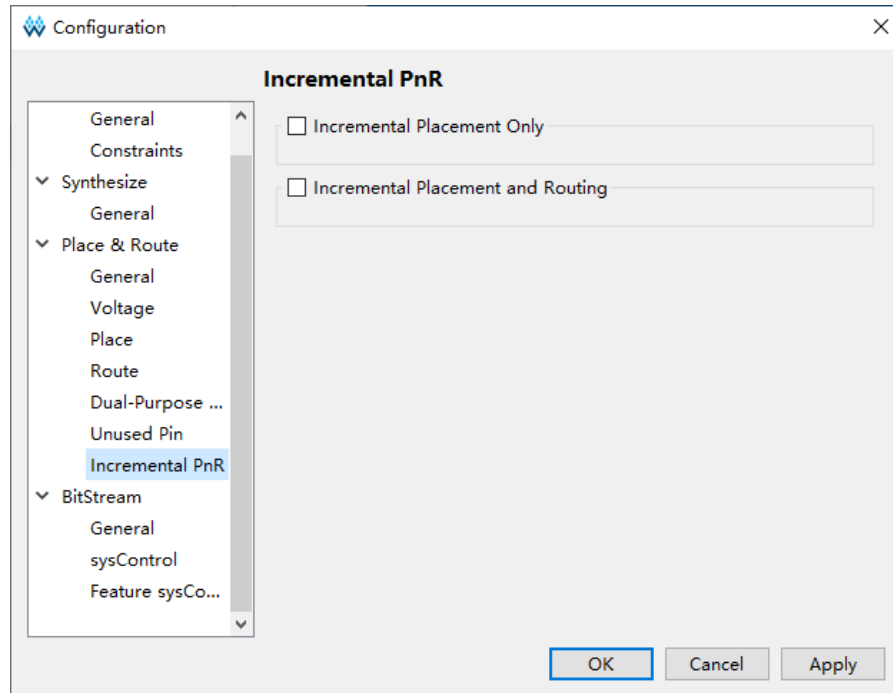


图 4-27 中的各选项使用说明如下所示：

Incremental Placement Only: 仅增量布局信息，默认不勾选。勾选之后联动显示以下子配置项。

- **Auto:** 重新布局时自动选择工程路径下已有的布局增量文件*.p，若工程路径下不存在*.p 文件，则弹出提示。
- **Specify the previous placement file:** 手动指定布局增量文件*.p。

Incremental Placement and Routing: 增量布局和布线信息，默认不勾选。勾选之后联动显示以下子配置项，并且 **Incremental Placement Only** 项不可选。

- **Auto:** 重新布局布线时自动选择工程路径下已有的布局布线增量文件*.pr，若工程路径下不存在*.pr 文件，则弹出提示。
- **Specify the previous placement and routing file:** 手动指定布局和布线增量文件*.pr。

BitStream

BitStream 是符合高云芯片下载模式的码流文件格式选项以及下载频率选项等，将鼠标悬浮在选项处会显示其解释。**BitStream** 选项又包括 **General**、**sysControl**、**Feature sysControl**，各选项含义如表 4-2 所示。

表 4-2 BitStream 配选项含义描述

工程选项	描述
General	用于配置 BitStream 的基本参数
sysControl	用于配置 BitStream 系统控制参数
Featrue sysControl	用于配置 BitStream 功能性系统控制参数

General

General 选项配置如图 4-28 所示。

图 4-28 配置 Bitstream General 选项

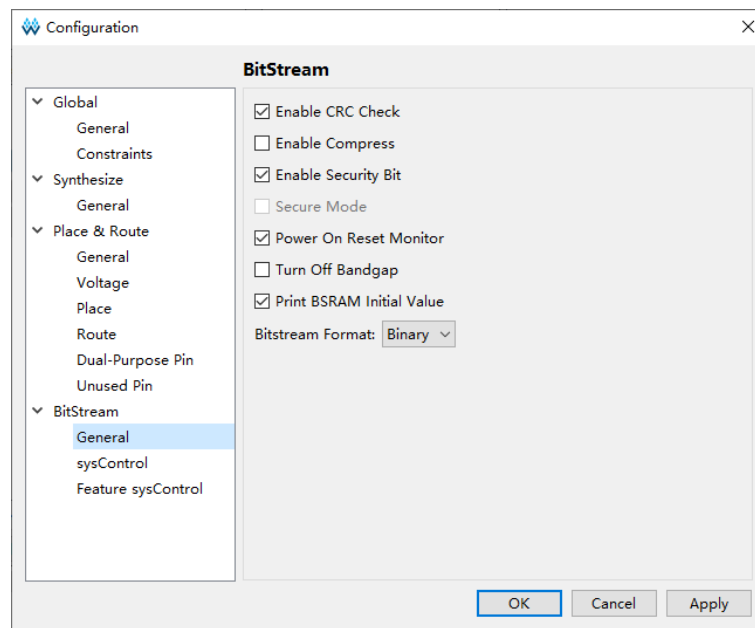


图 4-28 的各项参数配置使用说明如下：

- **Enable CRC Check:** 使能循环冗余校验，默认为勾选；
- **Enable Compress:** 使能码流文件压缩，默认为不勾选；
- **Enable Encryption:** 对码流文件进行加密处理，仅晨熙家族器件支持该选项的配置，其他器件下配置界面中不会显示该配置选项，默认为不勾选；
- **Key (Hex) :** 勾选“Enable Encryption”后才可以对该项进行编辑，该项可以使用户对加密的密钥进行自定义，仅晨熙家族器件支持该选项的配置，其他器件下配置界面中不会显示该配置选项，默认 key 全为 0。勾选该选项后执行布局布线，会生成 key 的密钥文件.ekey；
- **Enable Security Bit:** 使能安全位控制，对码流文件添加安全位，添加之后码流无法再次回读，默认为勾选；
- **Secure Mode:** 启用安全模式，此时 JTAG 管脚为 GPIO，码流文件只能对设备编程一次。只有器件 GW1NSER-4C 支持该功能，默认为不勾选状态；

- **Power On Reset Monitor:** 上电复位监视，默认为勾选状态。勾选该选项后会持续监测电源导轨中任何可能的电压压降。如果电源导轨电压低于 POR 阈值，会将所有 RAM 位清除，并通过内部弱上拉电阻将使用的 I/O 置为三态，然后依次完成配置、初始化工作；
- **Turn Off Bandgap:** 关闭 Bandgap 功能，默认为不勾选状态。Bandgap 的功能是为芯片中的某些模块提供恒定的电压和电流，关掉 Bandgap 后可以起到降低器件功耗的作用。只有器件 GW1N-1 支持该选项的配置，其他器件下配置界面中不会显示该配置选项。
- **Print BSRAM Initial Value:** 打印 BSRAM 的初始值到码流文件中，默认为勾选。对于 GW1N 系列器件和 GW2A 系列器件，勾选该选项后会将所有位置 BSRAM 的初始值都打印到码流文件中，没有被占用到的 BSRAM 位置的初始值会打印为 0。对于 GW5A(N)(S)(R)(T)器件，勾选该选项后会将所占用位置 BSRAM 所在的列的所有 BSRAM 的初始值都打印到码流文件中，该列中没有被占用到的 BSRAM 位置的初始值会打印为 0。
- **Bitstream Format:** 用于指定生成的码流文件内容的格式，可选项有 Text 和 Binary，默认为 Binary。当选择 Text 选项时，会生成纯文本格式的*.fs 文件；选择 Binary 选项时会生成*.fs、*.bin 和*.binx 格式的码流文件。*.bin 和*.binx 是二进制格式的码流文件，*.binx 文件含有头部注释信息，*.bin 没有头部注释信息。

sysControl

sysControl 选项配置如图 4-29 所示。

图 4-29 配置 sysControl 选项

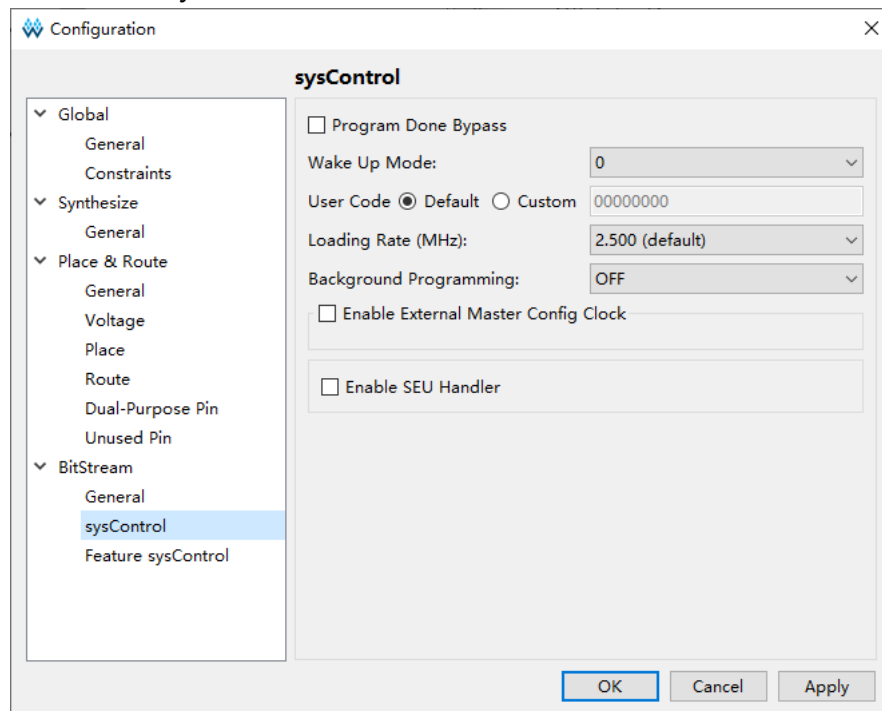


图 4-29 的各项参数配置使用说明如下：

- **Program Done Bypass:** 在 Done Final 内部信号生效时，同时外部 Done 信号保持低电平，使码流加载完成后可以转发新的码流数据；
- **Wake Up Mode:** 使能芯片唤醒模式，可选值有 0 和 1，默认值为 0；
 - Wake Up Mode 为 0: DONE 管脚拉高或拉低对 Wake Up 没有影响；
 - Wake Up Mode 为 1:
 - a) 如果 DONE 管脚处于拉高状态，可以正常下载且芯片正常工作；
 - b) 如果 DONE 管脚处于拉低状态，可以正常下载，下载完后 DONE 管脚需要拉高且同时保持 TCK 连接脉冲信号芯片才能 Wake Up。
- **User Code:** 用户可以自定义 User Code，定义的值会体现在产生的码流文件中，通过编程器下载码流文件时会对 User Code 进行校验。默认为 Default(00000000)；
- **Loading Rate:** AutoBoot 配置模式和 MSPI 配置模式下，码流文件从 Flash 到 SRAM 的加载速度。GW1N-4/GW1NRF-4B/GW1NR-4 默认为 2.100MHz。GW1NS-4/GW1NSR-4/GW1NSER-4C 的速度等级为 C7/I6 的芯片型号默认为 2.6MHz，GW5A(S)(T)-138/GW5A(R)-25 的默认为 35.000MHz，其他器件默认为 2.500MHz。AutoBoot 配置模式和 MSPI 配置模式参考文档 [UG290, Gowin FPGA 产品编程配置手册](#), [UG704, Arora V 138K FPGA 产品编程配置手册](#), [UG714, Arora V 25K FPGA 产品编程配置手册](#)。不同器件 Loading Rate 的取值及其计算方式不同；
 - 以下器件封装的 Loading Rate 取值只支持 2.500MHz
 - a) GW1N-2: LQFP100X/LQFP144X/MBGA132X/WLCSP42H/MBGA49
 - b) B 版 GW1N-2: LQFP100X/LQFP144X/MBGA132X/MBGA121X
 - c) C 版 GW1N-2: LQFP100X/LQFP144X/MBGA132X/MBGA121X/MBGA49/QFN32X
 - d) GW1NR-2: MBGA49P/MBGA49PG/MBGA49G
 - e) B 版 GW1NR-2: MBGA49P/MBGA49PG/MBGA49G
 - f) C 版 GW1NR-2: MBGA49P/MBGA49PG/MBGA49G
 - g) GW1N-1P5: LQFP100X
 - h) B 版 GW1N-1P5: LQFP100X/QFN48X
 - i) C 版 GW1N-1P5: LQFP100X/QFN48X

- 以下器件的 Loading Rate 取值及计算方式如表 4-3 所示
 - a) GW1NZ-1
 - b) GW1N-2/GW1N-1P5/GW1NR-2 除上述仅支持 2.500MHz 封装外的其他封装
 - c) GW1NSER-4C/GW1NS-4/GW1NSR-4/GW1NS-4C/GW1NSR-4C 速度等级为非 C7/I6 的芯片型号
 - d) GW1N-9/GW1NR-9
 - e) GW2A-18/GW2AR-18/C 版 GW2ANR-18
 - f) GW2A-55/C 版 GW2AN-55

表 4-3 Loading Rate 值及计算方式（一）

Loading Rate (MHz)	分数表示法
2.500 (default)	250 / 100
5.435	250 / 46
5.682	250 / 44
5.952	250 / 42
6.250	250 / 40
6.579	250 / 38
6.944	250 / 36
7.353	250 / 34
7.812	250 / 32
8.333	250 / 30
8.929	250 / 28
9.615	250 / 26
10.417	250 / 24
11.364	250 / 22
12.500	250 / 20
13.889	250 / 18
15.625	250 / 16
17.857	250 / 14
20.833	250 / 12
25.000	250 / 10
31.250	250 / 8
41.667	250 / 6
62.500	250 / 4

- 以下器件的 Loading Rate 取值及计算方式如表 4-4 所示
GW1N-1/GW1N-1S/GW1NR-1

表 4-4 Loading Rate 值及计算方式（二）

Loading Rate(MHz)	分数表示法
2.500 (default)	240 / 96
2.553	240 / 94
2.609	240 / 92
2.667	240 / 90
2.727	240 / 88
2.791	240 / 86
2.857	240 / 84
2.927	240 / 82
3.000	240 / 80
3.077	240 / 78
3.158	240 / 76
3.243	240 / 74
3.333	240 / 72
3.429	240 / 70
3.529	240 / 68
3.636	240 / 66
3.750	240 / 64
3.871	240 / 62
4.000	240 / 60
4.138	240 / 58
4.286	240 / 56
4.444	240 / 54
4.615	240 / 52
4.800	240 / 50
5.000	240 / 48
5.217	240 / 46
5.455	240 / 44
5.714	240 / 42
6.000	240 / 40
6.316	240 / 38
6.667	240 / 36
7.059	240 / 34
7.500	240 / 32
8.000	240 / 30
8.571	240 / 28
9.231	240 / 26
10.000	240 / 24

Loading Rate(MHz)	分数表示法
10.909	240 / 22
12.000	240 / 20
13.333	240 / 18
15.000	240 / 16
17.143	240 / 14
20.000	240 / 12
24.000	240 / 10
30.000	240 / 8
40.000	240 / 6
60.000	240 / 4

– 以下器件的 Loading Rate 取值及计算方式如表 4-5 所示

a) GW2AN-9X

b) GW2AN-18X

表 4-5 Loading Rate 值及计算方式（三）

Loading Rate(MHz)	分数表示法
2.500 (default)	200 / 80
1.562	200 / 128
1.587	200 / 126
1.613	200 / 124
1.639	200 / 122
1.667	200 / 120
1.695	200 / 118
1.724	200 / 116
1.754	200 / 114
1.786	200 / 112
1.818	200 / 110
1.852	200 / 108
1.887	200 / 106
1.923	200 / 104
1.961	200 / 102
2.000	200 / 100
2.041	200 / 98
2.083	200 / 96
2.128	200 / 94
2.174	200 / 92
2.222	200 / 90

Loading Rate(MHz)	分数表示法
2.273	200 / 88
2.326	200 / 86
2.381	200 / 84
2.439	200 / 82
2.564	200 / 78
2.632	200 / 76
2.703	200 / 74
2.778	200 / 72
2.857	200 / 70
2.941	200 / 68
3.030	200 / 66
3.125	200 / 64
3.226	200 / 62
3.333	200 / 60
3.448	200 / 58
3.571	200 / 56
3.704	200 / 54
3.846	200 / 52
4.000	200 / 50
4.167	200 / 48
4.348	200 / 46
4.545	200 / 44
4.762	200 / 42
5.000	200 / 40
5.263	200 / 38
5.556	200 / 36
5.882	200 / 34
6.250	200 / 32
6.667	200 / 30
7.143	200 / 28
7.692	200 / 26
8.333	200 / 24
9.091	200 / 22
10.000	200 / 20
11.111	200 / 18
12.500	200 / 16
14.286	200 / 14
16.667	200 / 12

Loading Rate(MHz)	分数表示法
20.000	200 / 10
25.000	200 / 8
33.333	200 / 6
50.000	200 / 4
100.000	200 / 2

- 以下器件的 Loading Rate 取值及计算方式如表 4-6 所示

GW1N-4/GW1NRF-4B/GW1NR-4

表 4-6 Loading Rate 值及计算方式（四）

Loading Rate (MHz)	分数表示法
2.100 (default)	210 / 100
4.565	210 / 46
4.773	210 / 44
5.000	210 / 42
5.250	210 / 40
5.526	210 / 38
5.833	210 / 36
6.176	210 / 34
6.563	210 / 32
7.000	210 / 30
7.500	210 / 28
8.077	210 / 26
8.750	210 / 24
9.545	210 / 22
10.500	210 / 20
11.667	210 / 18
13.125	210 / 16
15.000	210 / 14
17.500	210 / 12
21.000	210 / 10
26.250	210 / 8
35.000	210 / 6
52.500	210 / 4

- 以下器件的 Loading Rate 取值及计算方式如表 4-7 所示

GW1NSER-4C/GW1NS-4/GW1NSR-4/GW1NS-4C/GW1NSR-4C 速度等级为 C7/I6 的芯片型号

表 4-7 Loading Rate 值及计算方式（五）

Loading Rate (MHz)	分数表示法
2.600	260 / 100
5.652	260 / 46
5.909	260 / 44
6.190	260 / 42
6.500	260 / 40
6.842	260 / 38
7.222	260 / 36
7.647	260 / 34
8.125	260 / 32
8.667	260 / 30
9.286	260 / 28
10.000	260 / 26
10.833	260 / 24
11.818	260 / 22
13.000	260 / 20
14.444	260 / 18
16.250	260 / 16
18.571	250 / 14
21.667	260 / 12
26.000	260 / 10
32.500	260 / 8
43.333	260 / 6
65.000	260 / 4

- 以下器件的 Loading Rate 取值及计算方式如表 4-8 所示

GW5A(S)(T)-138/GW5A(R)-25/GW5AT-75

表 4-8 Loading Rate 值及计算方式（六）

Loading Rate (MHz)	分数表示法
35.000(default)	210 / 6
52.500	210 / 4
70.000	210 / 3
105.000	210 / 2

- 以下器件的 Loading Rate 取值及计算方式如表 4-9 所示

GW5AT-60/GW5A(N)(R)T-15

表 4-9 Loading Rate 值及计算方式（七）

Loading Rate (MHz)	分数表示法
2.500 (default)	210 / 84
1.667	210 / 126
1.694	210 / 124
1.721	210 / 122
1.750	210 / 120
1.780	210 / 118
1.810	210 / 116
1.842	210 / 114
1.875	210 / 112
1.909	210 / 110
1.944	210 / 108
1.981	210 / 106
2.019	210 / 104
2.059	210 / 102
2.100	210 / 100
2.143	210 / 98
2.188	210 / 96
2.234	210 / 94
2.283	210 / 92
2.333	210 / 90
2.386	210 / 88
2.442	210 / 86
2.561	210 / 82
2.625	210 / 80
2.692	210 / 78
2.763	210 / 76
2.838	210 / 74
2.917	210 / 72
3.000	210 / 70
3.088	210 / 68
3.182	210 / 66
3.281	210 / 64
3.387	210 / 62
3.500	210 / 60
3.621	210 / 58
3.750	210 / 56
3.889	210 / 54

Loading Rate (MHz)	分数表示法
4.038	210 / 52
4.200	210 / 50
4.375	210 / 48
4.565	210 / 46
4.773	210 / 44
5.000	210 / 42
5.250	210 / 40
5.526	210 / 38
5.833	210 / 36
6.176	210 / 34
6.563	210 / 32
7.000	210 / 30
7.500	210 / 28
8.077	210 / 26
8.750	210 / 24
9.545	210 / 22
10.500	210 / 20
11.667	210 / 18
13.125	210 / 16
15.000	210 / 14
17.500	210 / 12
21.000	210 / 10
26.250	210 / 8
35.000	210 / 6
52.500	210 / 4
70.000	210 / 3
105.000	210 / 2

- **Background Programming:** 背景升级功能，在不中断 FPGA 芯片当前正在执行的码流文件的前提下对 Flash 进行再次烧录。如果当前器件的 Background Programming 取值只有 OFF，则在配置界面中不会显示该配置选项。

图 4-30 Background Programming 设为 I2C

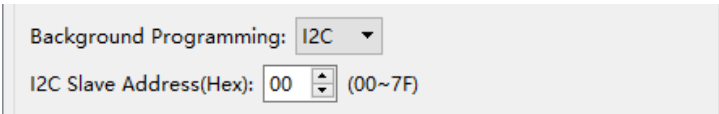
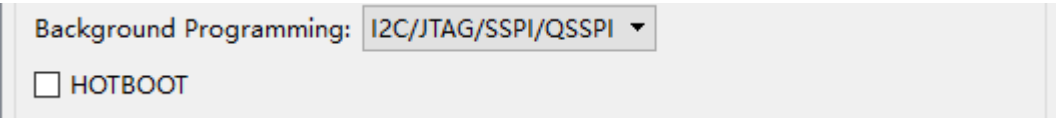


图 4-31 Background Programming 设为 I2C/JTAG/SSPI/QSSPI



支持背景升级的器件及其取值情况所示。

表 4-10 Background Programming 取值

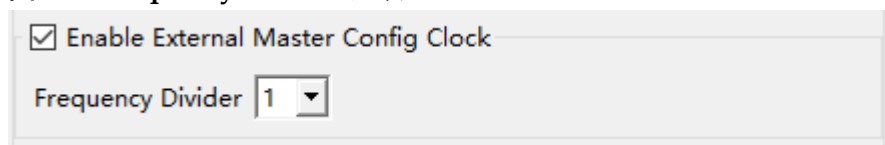
器件类型	Background Programming取值
● GW1N-1P5/GW1N-2/GW1NR-2 ● B版: GW1N-4/GW1NR-4、GW1NRF-4 ● D版: GW1NR-4 ● GW1NS-4/GW1NSR-4 ● GW1N-9/GW1NR-9 ● GW1NZ-1	OFF、JTAG 默认为OFF
B版: GW1N-1P5/GW1N-2/GW1NR-2	OFF、JTAG、I2C 默认为OFF
C版: GW1N-2/GW1NR-2/GW1N-1P5	OFF、GoConfig、GoConfig Mode1、JTAG、I2C 默认为OFF
GW2AN-18X/GW2AN-9X	OFF、GoConfig、UserLogic、I2C/JTAG/SSPI/QSSPI 默认为OFF
GW5A(N)(S)(R)(T)	OFF、UserLogic、JTAG/SSPI/QSSPI 默认为OFF

Background Programming 各取值功能及使用注意事项介绍如下：

- OFF：关闭 Background Programming 功能，如果器件为 GW2AN-18X 或 GW2AN-9X，“Dual-Purpose Pin”对话框中的“Use MSPI as regular IO”为不勾选且不可配置状态。
- JTAG：使用 JTAG 模式进行背景升级。
- I2C：使用 I2C 模式进行背景升级。对于 B 版 GW1N-1P5/GW1N-2/GW1NR-2，配置对话框中会出现选项“I2C Slave Address(Hex)”，对用户操作 I2C 设备的地址进行设置，可设置值范围为 00~7F，如图 4-30 所示。选择 I2C 后“Dual-Purpose Pin”对话框中的“Use JTAG as regular IO”为不勾选且不可配置状态。对于 C 版本 GW1N-2/GW1N-1P5/GW1NR-2，使用 I2C 模式进行背景升级时，配置对话框中不会出现选项“I2C Slave Address(Hex)”，“Dual-Purpose Pin”对话框中的“Use RECONFIG as regular IO”为不勾选且不可配置状态。
- GoConfig：使用 goConfig IP 实现背景升级。

- **UserLogic:** 使用内部逻辑进行背景升级。
- **I2C/JTAG/SSPI/QSSPI:** 使用 I2C/JTAG/SSPI/QSSPI 模式进行背景升级。
- **JTAG/SSPI/QSSPI:** 使用 JTAG/SSPI/QSSPI 模式进行背景升级。
- 在 GW2AN-18X/GW2AN-9X 下, 选择 GoConfig 或 UserLogic 或 I2C/JTAG/SSPI/QSSPI 时, 对话框中会出现配置选项 “HOTBOOT”, 如图 4-31 所示, 为可选状态, 默认不勾选。
- 在 GW5A(S)(T)-138/GW5A(R)-25 下, 选择 UserLogic 或 JTAG/SSPI/QSSPI 时, 对话框中会出现配置选项 “HOTBOOT”, 如图 4-31 所示, 为可选状态, 默认不勾选。
- 对于 B 版 GW1N-1P5/GW1N-2/GW1NR-2, 如果配置项切换前后包含 I2C, 则 Synthesize 和 Place & Route 的状态都会改为过期状态。对于 C 版 GW1N-2/GW1NR-2/GW1N-1P5, 如果配置项切换前后包含 I2C, 则只有 Place & Route 的状态会改为过期状态。
- 对于 GW2AN-18X/GW2AN-9X, 如果配置项 GoConfig、UserLogic 与 I2C/JTAG/SSPI/QSSPI、OFF 相互切换时, 则 Synthesize 和 Place & Route 的状态都会改为过期状态, 否则只有 Place & Route 的状态会改为过期状态。
- **Enable External Master Config Clock:** 启用外部主配置时钟, 仅 GW5A(N)(S)(R)(T)系列器件支持该选项的配置, 其他器件下配置界面中不会显示该配置选项, 默认为不勾选。当勾选该选项时, 对话框中会出现配置选项 “Frequency Divider”, 如图 4-32 所示。对于 GW5A(S)(T)-138/GW5AT-75 器件, 可选值有 1、2、4、8, 默认为 1。对于 GW5A(S) (R)-25/GW5AT-60/GW5A(N)(R)T-15 器件, 可选值有 1、2~1022 偶数, 共 512 个值, 默认为 1。

图 4-32 Frequency Divider 选项



- **Enable SEU Handler:** 启用单粒子翻转处理器 (Single-Event Upsets Handler, SEU Handler)。只有 GW5A(N)(S)(R)(T)系列器件支持该功能, 默认为不勾选。当勾选该选项时, 对话框中会出现配置子选项 “Enable SEU Handler CheckSum”、“Enable Error Detection only”、“Enable Error Detection and Correction”、“Stop SEU Handler when detected uncorrectable ECC error or CRC checksum mismatch error”、“Mode”、“Enable Error Injection”, 其中, “Enable Error Detection only” 和 “Enable Error Detection and Correction” 不能同时勾选。如图 4-33 所示。

图 4-33 Enable SEU Handler 选项

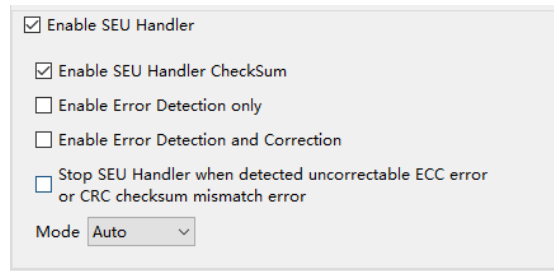
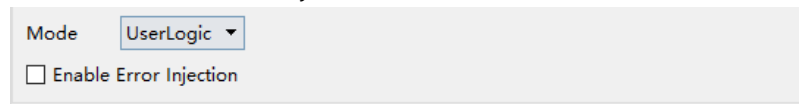


图 4-34 Enable Error Injection 选项

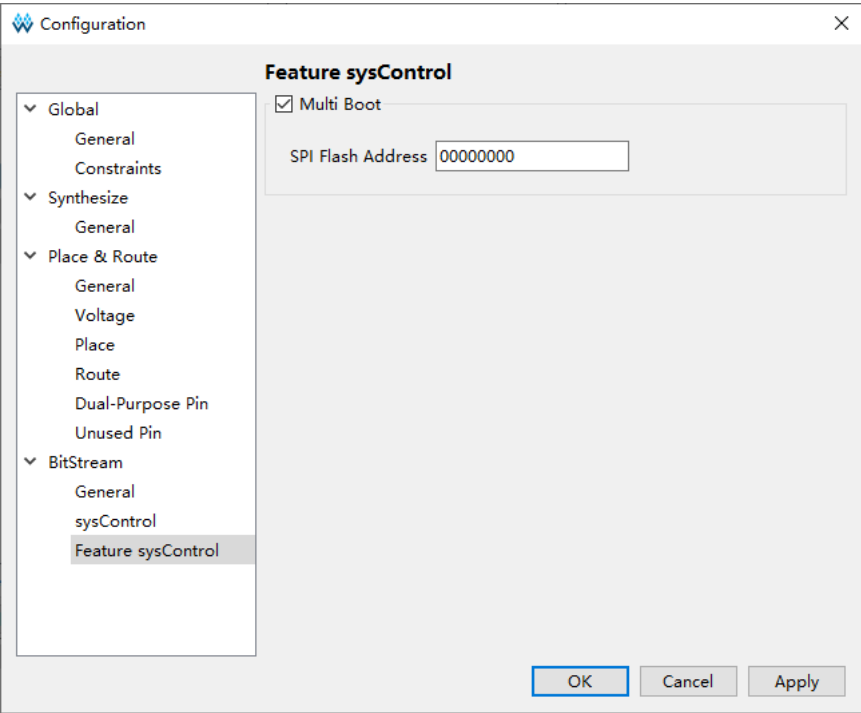


- **Enable SEU Handler CheckSum:** 启用单粒子翻转处理、检验、计算和对比，默认不勾选。
- **Enable Error Detection only:** 仅启用错误检验，默认不勾选。
- **Enable Error Detection and Correction:** 启用错误检验和更正，默认不勾选。
- **Stop SEU Handler when detected uncorrectable ECC error or CRC checksum mismatch error:** 当检测到不可纠正的 ECC 错误或 CRC 校验和不匹配错误时，停止单粒子翻转处理器。默认不勾选。
- **Mode:** 选择 SEU Handler 启动或停止的模式，可取值有 Auto 和 UserLogic，默认为 Auto。
- **Enable Error Injection:** 启用错误注入，当 Mode 选择 UserLogic 时会出现该选项，默认不勾选，如图 4-34 所示。

Feature sysControl

对于非 GW5A(N)(S)(R)(T)系列器件，Feature sysControl 选项配置如图 4-35 所示。

图 4-35 非 GW5A(N)(S)(R)(T)系列器件配置 Feature sysControl 选项

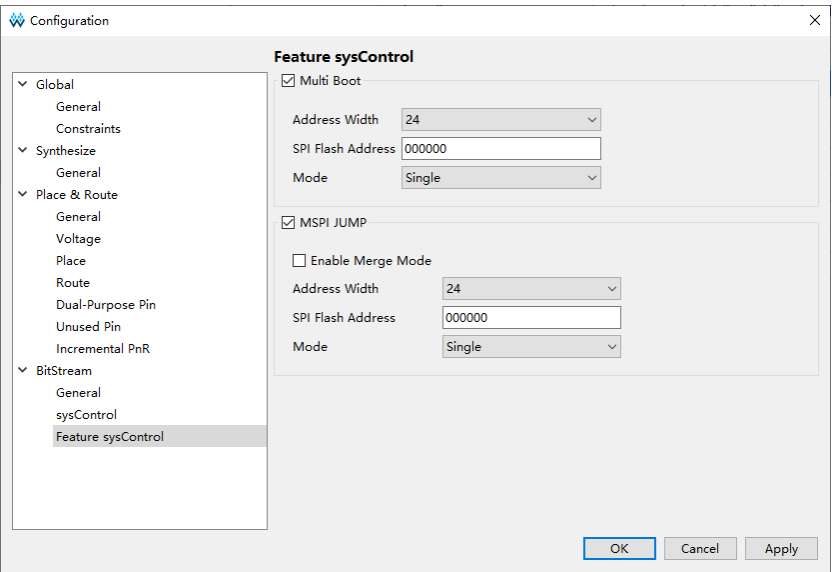


默认勾选 Multi Boot，会联动显示子配置项 SPI Flash Address。

- SPI Flash Address: 指定 SPI Flash 地址。SPI Flash 地址是指下一次 multiboot 时，加载码流文件的起始地址。对于 GW2AN-18X 和 GW2AN-9X，默认为 000000，对于其他的非 GW5A(N)(S)(R)(T)系列器件，默认值为 00000000。具体可参考 [SUG502, Gowin Programmer 用户指南](#);

对于 GW5A(N)(S)(R)(T)系列器件，Feature sysControl 选项配置如图 4-36 所示。

图 4-36 GW5A(N)(S)(R)(T)系列器件配置 Feature sysControl 选项



Multi Boot 默认不勾选，勾选之后会联动显示以下子配置项。

表 4-11 子配置项

名称	描述
Address Width	配置SPI Flash地址位宽，可取值有24、32，默认为24。
SPI Flash Address	指定SPI Flash地址。SPI Flash地址是指下一次multiboot时，加载码流文件的起始地址，默认为0000000，具体可参考 SUG502, Gowin Programmer用户指南 。
Mode	配置SPI Flash地址访问模式，可取值有Single、Fast、Dual和Quad，默认为Single。

MSPI JUMP 默认不勾选，勾选之后会联动显示以下子配置项。

表 4-12 子配置项

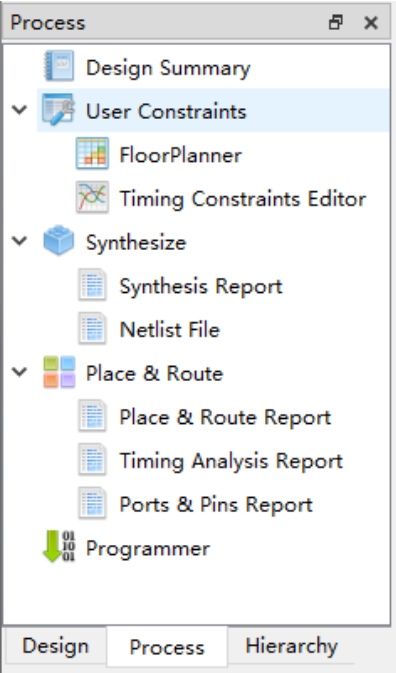
名称	描述
Enable Merge Mode	使用该选项会将MSPI JUMP码流文件合并到通用的码流文件中，默认不勾选。
Address Width	配置SPI Flash地址位宽，可取值有24、32，默认为24
SPI Flash Address	指定SPI Flash地址，默认为0000000。
Mode	配置SPI Flash地址访问模式，可取值有Single、Fast、Dual和Quad，默认为Single。

4.4 管理工程过程

在过程管理区，列出了云源的过程管理，如图 4-37 所示。在该窗口中可进行以下操作：

- 查看 Design 概述；
- 启动物理约束编辑器；
- 启动时序约束编辑器；
- 执行综合；
- 查看综合设计报告；
- 执行布局布线；
- 查看 Place&Route 后生成的报告；
- 启动编程器等功能。

图 4-37 工程 Process 窗口



4.4.1 Design Summary

新建工程时，软件会分析工程并提供一份报告 **Design Summary**，报告中包括工程文件路径、综合工具的信息以及器件信息，如图 4-38 所示，有三种方法打开 **Design Summary**：

- 在 GOWIN FPGA Designer 菜单栏上，选择 “Window > Design Summary” ；
- 在 Process 窗口中，双击 “Design Summary” ；
- 在 Process 窗口中，右击 “Design Summary” 选择 “Open” 。

图 4-38 工程信息显示

General	
Project File:	D:\gowin_project\daily_test\daily_test.gprj
Synthesis Tool:	GowinSynthesis

Target Device	
Part Number:	GW1N-UV4PG256C6/I5
Series:	GW1N
Device:	GW1N-4
Device Version:	B
Package:	PBGA256
Speed Grade:	C6/I5
Core Voltage:	UV

注！
对于没有 **Device Version** 的器件，表格中不会显示 **Device Version** 这一行。

4.4.2 User Constraints

User Constraints 提供了快速打开和创建约束文件的方法。User Constraints 分为物理约束和时序约束。

有关约束编辑器的详细使用方法，请参考 [SUG940, Gowin 设计时序约束指南](#); [SUG935, Gowin 设计物理约束指南](#); [SUG1018, Arora V 设计物理约束用户指南](#)。

4.4.3 Synthesize

GowinSynthesis 是高云研发的综合软件，支持高云半导体的库文件及其实实现，目前 Verilog 语言支持 System Verilog 2017、Verilog 2001 和 Verilog 95，VHDL 语言支持 VHDL1993 和 VHDL 2008。

通过在工程管理区的 Synthesize 处右击选择“Configuration”，打开综合选项对话框，选择综合工具，如图 4-20 所示。

Synthesize 提供了运行综合、设置综合选项参数及管理网表文件（Netlist File）和综合报告（Synthesis Report）的功能，综合报告的介绍请参考 6.1 综合报告。

参考以下步骤运行 Synthesize:

1. 配置 Synthesis 选项，有关 Synthesis 选项，请参考 [4.3.3 编辑工程配置](#)；
2. 运行 Synthesize；
3. 在过程管理区，双击“Synthesize”或右击“Synthesize > Run”，启动综合工具对源文件进行综合。若综合成功，则 Synthesize 栏前会出现图标“

若综合之前（Synthesize 图标为“

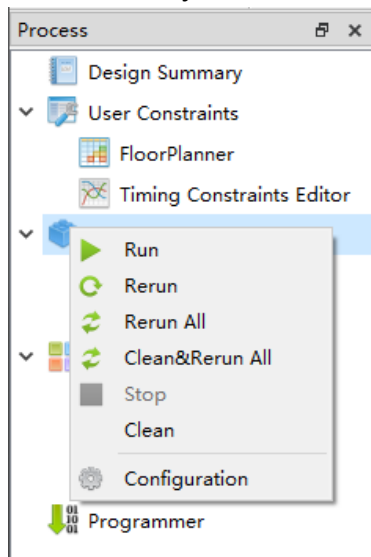
右击“Synthesize”，如图 4-39 可进行的操作包含：

- Run: 只有 Synthesize 栏前图标为初始状态“SUG100-4.4.1

夹并重新执行 Synthesize 及 Place & Route;

- Stop: 停止运行 Synthesize 过程;
- Clean: 清除综合后产生的文件夹 (GowinSynthesis 产生的文件夹为 gwsynthesis), 单击该选项会弹出提示框;
- Configuration: 可对 Synthesize 中参数进行设置。

图 4-39 右击 Synthesize



4.4.4 Place & Route

Place & Route 提供运行布局布线、设置布局布线参数及管理布局布线后生成文件的功能。

注!

Place & Route 依赖于综合过程, 执行该步骤时, 如其依赖项 (Synthesize) 未执行, 则会先执行 Synthesize, 再执行该步骤。

参考以下步骤运行 Place & Route:

1. 配置 Place & Route 选项, 关于 Place & Route 选项的介绍请参考 [4.3.3 编辑工程配置](#);
2. 运行 Place & Route, 双击 “Place & Route” 或右击 Place & Route 选择 “Run” 执行布局布线产生码流文件和相关报告文件。如运行成功, 则 Place & Route 栏前会出现图标 “✓”, 否则出现图标 “! ”;
3. Place & Route 运行成功后, 在 Place & Route 下方双击文件或右击选择 “Open” 可在文本编辑区浏览报告文件;
4. 可查看生成的四种报告文件, 包括布局布线报告 (Place & Route Report)、时序分析报告 (Timing Analysis Report)、端口属性报告 (Ports & Pins Report) 及功耗分析报告 (Power Analysis Report), 这四种文件均不可编辑。具体请参考 [6.2 布局布线报告](#)、[6.3 端口属性报告](#)、[6.4 时序报告](#)、[6.5 功耗分析报告](#)。

注!

- 若当前已经打开报告文件，再运行 **Place & Route** 重新生成报告文件后会提示是否更新文件；
- 若运行 **Place & Route** 之前（**Place & Route** 栏前图标为 “”），双击报告文件或在该报告文件右击选择“Open”，则会先运行 **Place & Route**，运行成功后打开报告文件。

右击 **Place & Route**，可进行的操作包含：

- **Run**：只有 **Place & Route** 栏前图标为初始状态 “” 或失败状态 “” 或过期状态 “” 时，选择该选项会运行 **Place & Route**；
- **Rerun**：无论 **Place & Route** 是何种状态，选择该选项，重新运行 **Place & Route**；
- **Rerun All**：无论 **Synthesize** 和 **Place & Route** 是何种状态，选择该选项，都会重新对源文件进行 **Synthesize** 及 **Place & Route**；
- **Clean & Rerun All**：清除工程文件夹 impl 下的 gwsynthesis 和 pnr 文件夹并重新执行 **Synthesize** 及 **Place & Route**；
- **Stop**：停止运行 **Place & Route** 过程；
- **Clean**：清除运行 **Place & Route** 后产生的文件夹（pnr），单击该选项会弹出是否要删除文件夹的提示框。如果删除文件夹失败会报出 Warning 提示；
- **Configuration**：可对 **Place & Route** 参数进行设置。

4.4.5 Programmer

云源在布局布线运行成功后，会生成码流文件，需启动编程器才可将码流文件下载到芯片，实现用户所需的功能。

注！

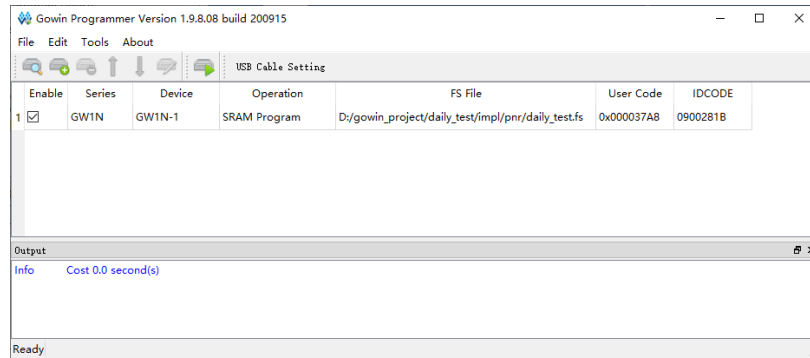
Programmer 依赖于 **Synthesize** 及 **Place & Route** 步骤，执行该步骤时，如其依赖项（**Synthesize** 及 **Place & Route**）未执行，会弹出警告提示。

双击 “**Programmer**” 或在该项右击 “**Run**” 选项，打开编程器，如图 4-40 所示。

注！

Linux 安装包中的编程器不适用于 Linux 版本 Red Hat 5.10，只支持 Red Hat 6 以上的版本，且 Linux 内核版本需要在 2.18 及以上。

图 4-40 Programmer 界面



有关编程器的详细使用方法，请参考 [SUG502, Gowin Programmer 用户指南](#)。

4.5 工程存档及恢复

云源支持对当前工程进行存档以及恢复被存档的工程。通过菜单栏 Project 中的“Archive Project”和“Restore Archived Project”对工程进行存档或恢复。

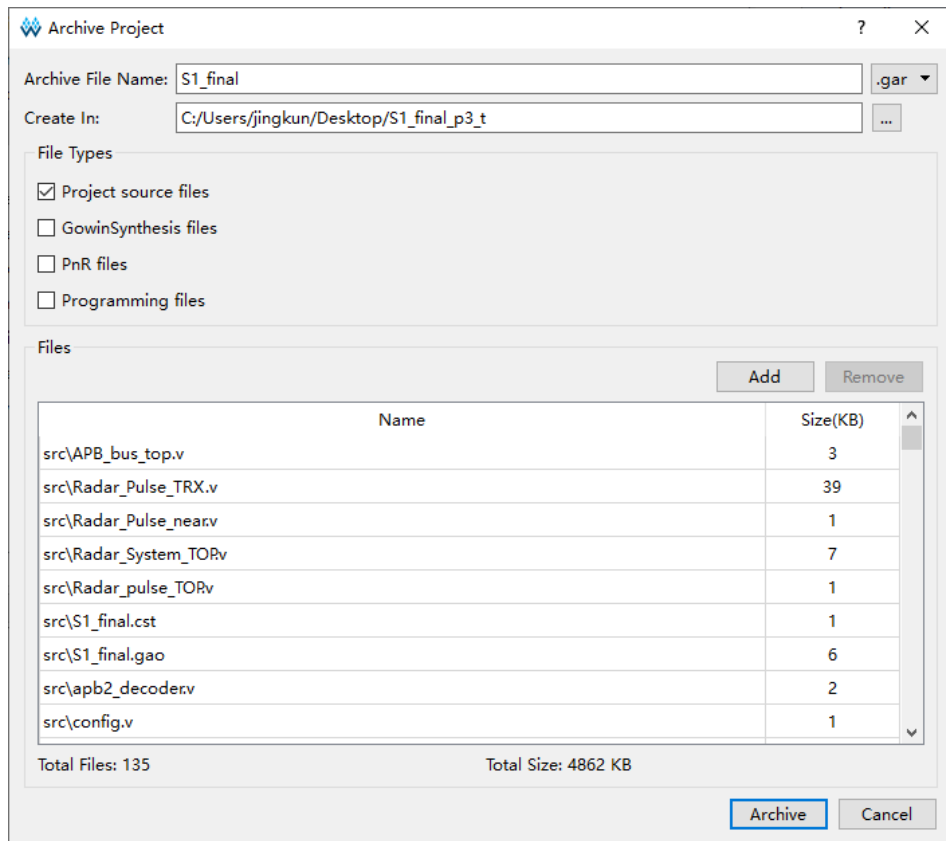
4.5.1 工程存档

单击菜单栏 Project 中的“Archive Project”会弹出存档工程对话框，如图 4-41 所示，将鼠标悬浮在选项处时会显示其解释。

- **Archive File Name** 是存档后的文件名，默认和当前要存档的工程名一致，扩展名是.gar；
- **Create In** 为存档后的文件要存放的路径，默认为当前工程路径；
- 存档工程项包括 **Project source files**（默认勾选）、**GowinSynthesis files**、**PnR files**、**Programming files**；
 - **Project source files**: 包含工程所在路径/src 下的所有文件；
 - **GowinSynthesis files**: 包含工程所在路径/impl/gwsynthesis 下运行综合产生的工程文件 (*.prj)、网表文件 (*.vg)、综合报告 (*_syn.rpt.html)、资源统计文件 (*_syn_rsc.xml)；
 - **PnR files**: 包含工程所在路径/impl/pnr 下运行布局布线产生的文件；
 - **Programming files**: 包含工程所在路径/impl/pnr 下运行布局布线产生的码流文件*.fs、*.bin 和*.binx。
- 当勾选了某一存档工程项后，下方会显示该项在当前工程下的源文件、所在路径以及大小；
- **Add** 和 **Remove** 可以用来自行增删存档的文件；
- 单击 **Archive** 后，如果工程中的文件未保存，则会弹出警告提示框，提示先保存文件；

- 存档完成后会弹出提示框，提示存档成功或失败；
- 存档完成后在 **Create In** 路径下会生成两个文件：存档工程*.gar 和与 gar 同名的存档文件*.garlog。扩展名为.gar 的文件压缩存储所有被存档的文件，日志文件*.garlog 供参考使用，可以查看哪些文件被存档以及存档是否成功。

图 4-41 工程存档对话框



4.5.2 恢复存档工程

单击菜单栏 **Project** 中的“Restore Archived Project”会弹出恢复存档工程对话框，如图 4-42 所示。

图 4-42 恢复存档工程对话框



单击“Archived File”右面的路径选择按钮，选择要恢复的存档文件。选择要恢复的存档文件后，“Destination Folder”会自动更新为存档文件所在的路径。单击“OK”后会弹出恢复成功的对话框。

4.6 退出软件

通过以下方式退出云源：

1. 单击“**File > Exit**”选项；
2. 单击软件界面右上角图标“”。

注！

- 若有未保存的文件，则会先提示是否对文件进行保存；
- 软件提供的保存（**Save**）、保存所有（**Save All**）和另存为（**Save As...**）功能只针对文本编辑动作的保存；
- 软件对于工程配置（**Configuration**）信息的修改或对工程中文件的增删等操作不会即时保存到工程配置文件中，在软件关闭时自动保存；
- 如果软件在运行中，无法通过单击退出软件。

5 云源集成工具

5.1 物理约束编辑器

FloorPlanner 是高云半导体面向市场自主研发的物理约束编辑器，支持对 I/O、Primitive（原语）、Block（BSRAM、DSP）、Group 等的属性及位置信息的读取与编辑功能，同时可根据用户的配置生成新的布局与约束文件，文件中规定了 I/O 的属性信息，原语、模块的位置信息等。

FloorPlanner 提供了快捷的布局与约束编辑功能，有效地提高编写物理约束文件的效率，同时可以根据器件布局和时序路径进行时序优化，可支持高云半导体的各款 FPGA 器件产品。

启动 FloorPlanner 有两种方式：

1. 未建立 FGPA 工程时，可直接在软件菜单栏的“Tools”下拉列表中选择“FloorPlanner”，此时需通过“File > New...”加载网表及所需器件信息；
2. 建立 FPGA 工程时，直接在过程管理区运行 Synthesize 后双击“FloorPlanner”，此时，FloorPlanner 会直接加载工程文件并显示在 FloorPlanner 界面。FloorPlanner 分为 Summary、Netlist、Chip Array、Package View 以及各项约束窗口，如图 5-1 和图 5-2 所示。

注！

- 该工具的详细使用方法请参考 [SUG935, Gowin 设计物理约束指南](#); [SUG1018, Arora V 设计物理约束用户指南](#)。
- 除此之外 FloorPlanner 还可以进行时序优化。

图 5-1 Chip Array 窗口

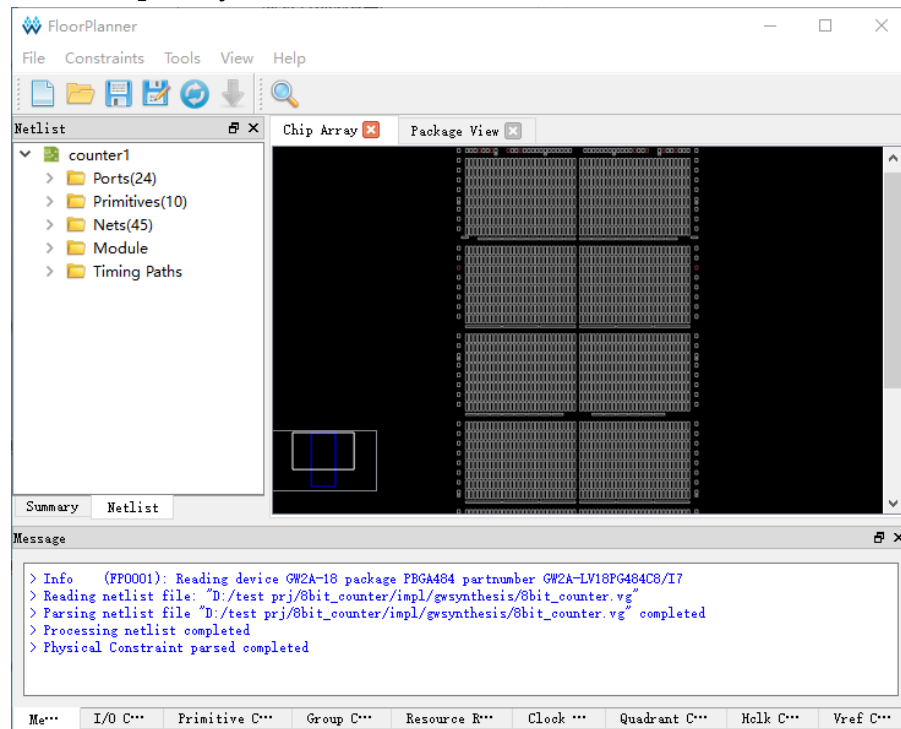
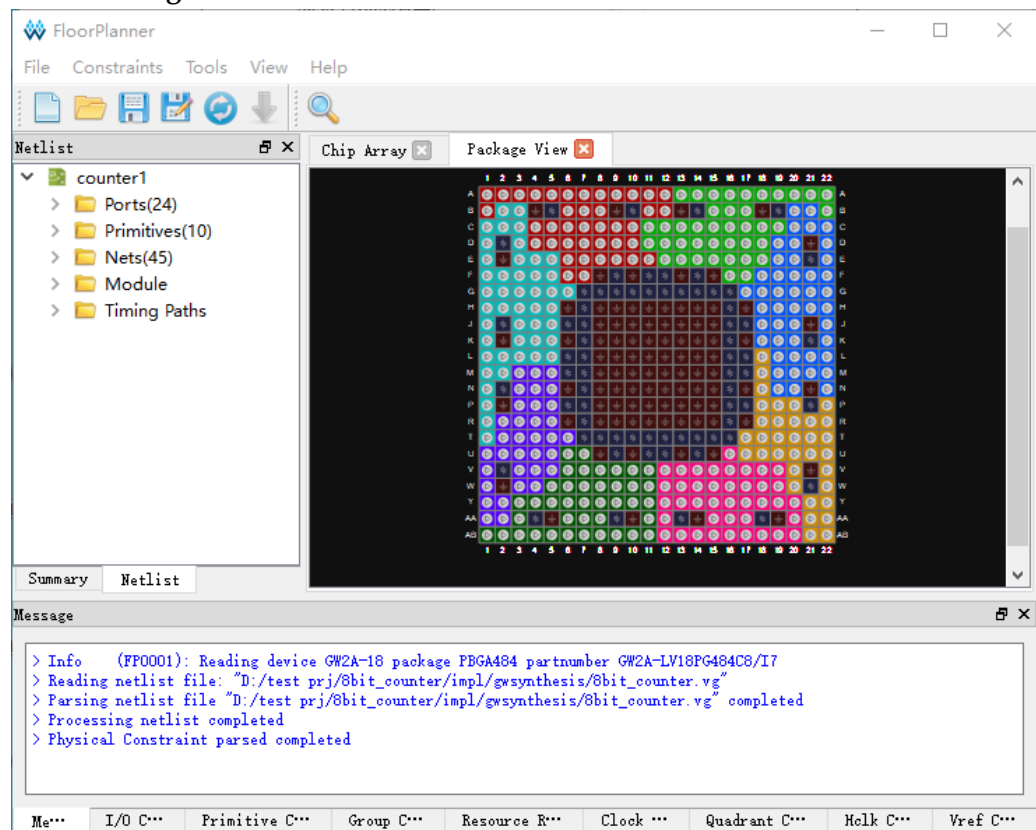


图 5-2 Package View 窗口



5.2 时序约束编辑器

时序约束编辑器（Timing Constraints Editor）是高云半导体面向市场自主研发的时序约束编辑工具，支持多种时序约束命令的编辑，包括时钟约束、输入输出约束、路径约束和时钟报告等约束编辑。Timing Constraints Editor 提供了简单快捷的时序约束编辑功能，可支持高云半导体的各 FPGA 器件产品。

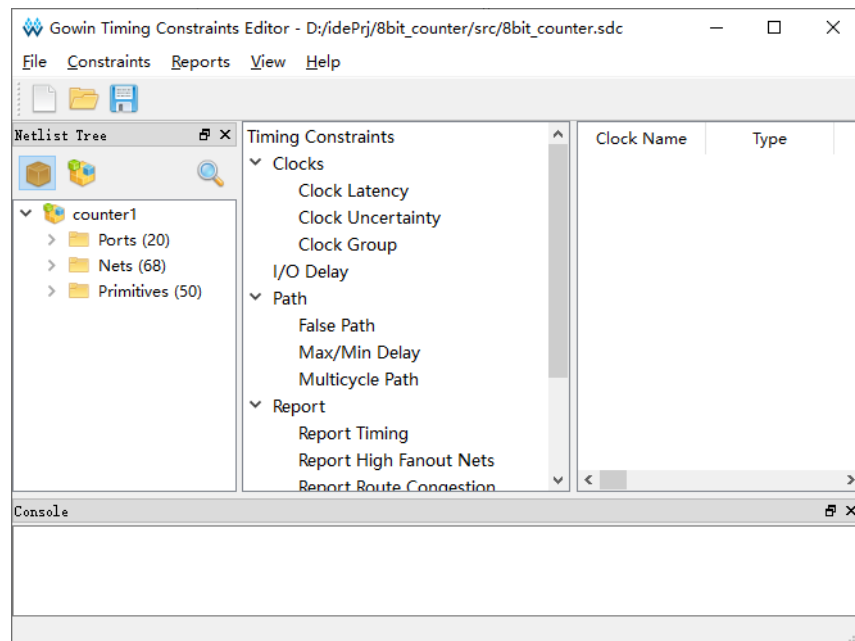
启动 Timing Constraints Editor 的方式有两种：

1. 如未建立 FPGA 工程，在菜单栏中，选择“Tools > Timing Constraints Editor”，此时，需要通过“File > New...”加载网表文件；
2. 如已建立 FPGA 工程，在过程管理区运行 Synthesize 后，双击“Timing Constraints Editor”，则 Timing Constraints Editor 会直接加载工程文件并显示在 Gowin Timing Constraints Editor 界面，如图 5-3 所示。

注！

有关时序约束编辑器的详细使用方法，请参考 [SUG940, Gowin 设计时序约束指南](#)。

图 5-3 创建时序约束界面



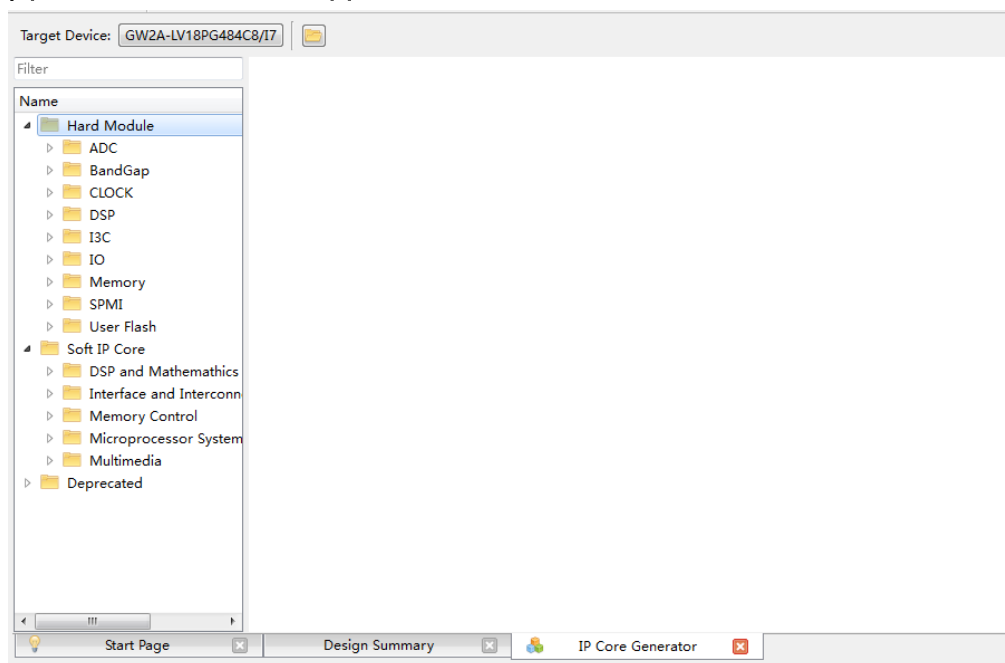
5.3 IP Core 产生器

IP Core Generator 是基于高云 FPGA 的 IP 产生工具，包括硬核和软核两部分，通过工具产生实例化的设计之后，用户可调用该实例化模块实现设计所需的功能，帮助用户快速实现复杂设计。如图 5-4 所示，主要功能有：

- 支持 Soft IP core、Hard module 的信息预览；
- 支持 Soft IP core、Hard module 定制生成；
- 支持 Hard module 实例化示例案例生成；

- 支持自动保存用户配置；
- 支持 IP 生成代码语言选择；
- 部分 Soft IP 支持自动产生激励文件；
- 支持器件信息自动过滤显示可用 IP。

图 5-4 IP Core Generator 窗口



通过单击菜单栏中 Tools 下拉列表中“IP Core Generator”启动 IP Core Generator 工具，进行 IP 调用。相关参考文档如下：

- 模数转换器（ADC）的产生参考 [SUG283, Gowin 原语用户指南](#)；[UG299, Arora V 系列模拟数字转换器（ADC）用户指南](#)。
- BANDGAP 的产生参考 [SUG283, Gowin 原语用户指南](#)；
- 时钟资源（CLOCK）的产生参考 [UG286, Gowin 时钟资源（Clock）用户指南](#)；[UG306, Arora V 时钟资源（Clock）用户指南](#)。
- 数字信号处理器（DSP）的产生参考 [UG287, Gowin 数字信号处理（DSP）模块用户指南](#)；[UG305, Arora V 数字信号处理（DSP）模块用户指南](#)。
- I3C 的产生参考 [SUG283, Gowin 原语用户指南](#)；
- 输入输出逻辑（IO Logic）的产生参考 [UG289, Gowin 可编程通用管脚（GPIO）用户指南](#)；[UG304, Arora V 可编程通用管脚（GPIO）用户指南](#)。
- 存储器（BSRAM & SSRAM）的产生参考 [UG285, Gowin 存储器（BSRAM & SSRAM）用户指南](#)；[UG300, Arora V 存储器（BSRAM & SSRAM）用户指南](#)。

- 闪存资源（User Flash）的产生参考 [UG295, Gowin 闪存资源（User Flash）用户指南](#);
- 双线串行接口（SPMI）的产生参考 [SUG283, Gowin 原语用户指南](#);
- 软核相关的文档请参考官网的 IP 参考设计：
<http://www.gowinsemi.com.cn/enrollment.aspx?FId=n27:27:4>。

注！

置灰的 Hard Module 或者 Soft IP Core 为当前 device 所不支持的。

5.4 在线逻辑分析仪

高云在线逻辑分析仪（GAO）是高云半导体自主研发的一款数字信号分析工具，旨在帮助用户更加简便地分析设计中信号之间的时序关系，快速进行系统分析和故障定位，提高设计效率。

GAO 支持 RTL 级信号、综合后网表级信号捕获的标准版（Standard）和精简版（Lite）两个版本。标准版 GAO 最多可以支持 16 个功能内核，每个内核可配置一个或多个触发端口，支持多级静态或动态触发表达式。精简版 GAO 配置简便，无需设置触发条件，精简版 GAO 还可以捕获信号的初始值，方便用户分析上电瞬间的工作状态。信号捕获后可以将其波形导出，支持*.csv、*.vcd 和*.prn 三种导出文件格式，*.csv 和*.prn 两种类型的文件可直接用于 Matlab 等第三方仿真工具使用，*.vcd 类型文件可用 ModelSim 工具使用。

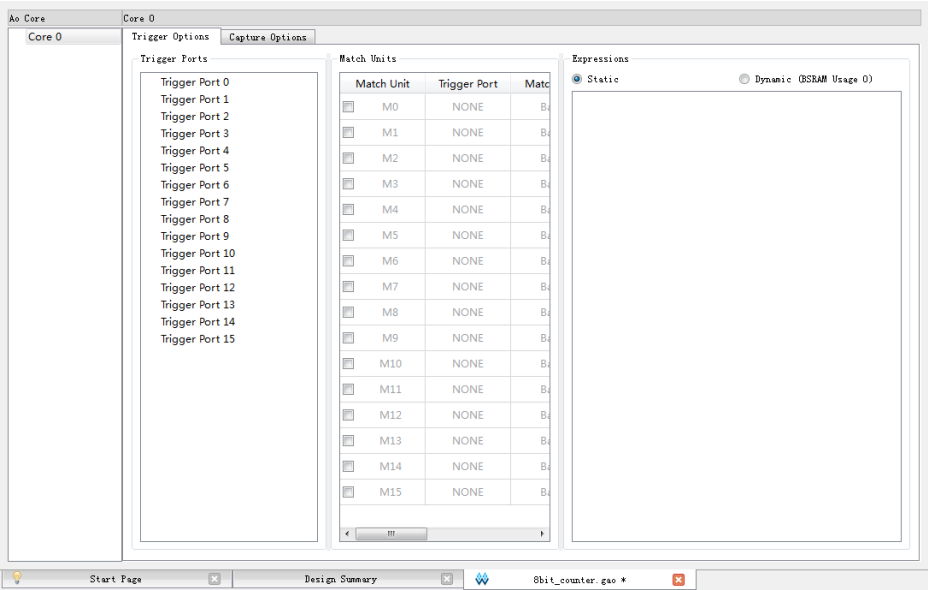
注！

Matlab、ModelSim 工具的使用需要取得第三方授权。

GAO 包括 Gowin GAO 配置和 Gowin Analyzer Oscilloscope 两个工具。Gowin GAO 配置主要用于把定位信息配置到设计中，这些定位信息主要基于采样时钟、触发单元和触发表达式；Gowin Analyzer Oscilloscope 通过 JTAG 接口连接软件和目标硬件，将 GAO 配置文件设置的采样信号的数据直观地通过波形显示出来。

启动 GAO 配置文件窗口之前，需在工程管理区新建 GAO 配置文件，打开配置文件窗口，以下是以标准版 GAO 为例，如图 5-5 所示。

图 5-5 GAO 配置文件窗口

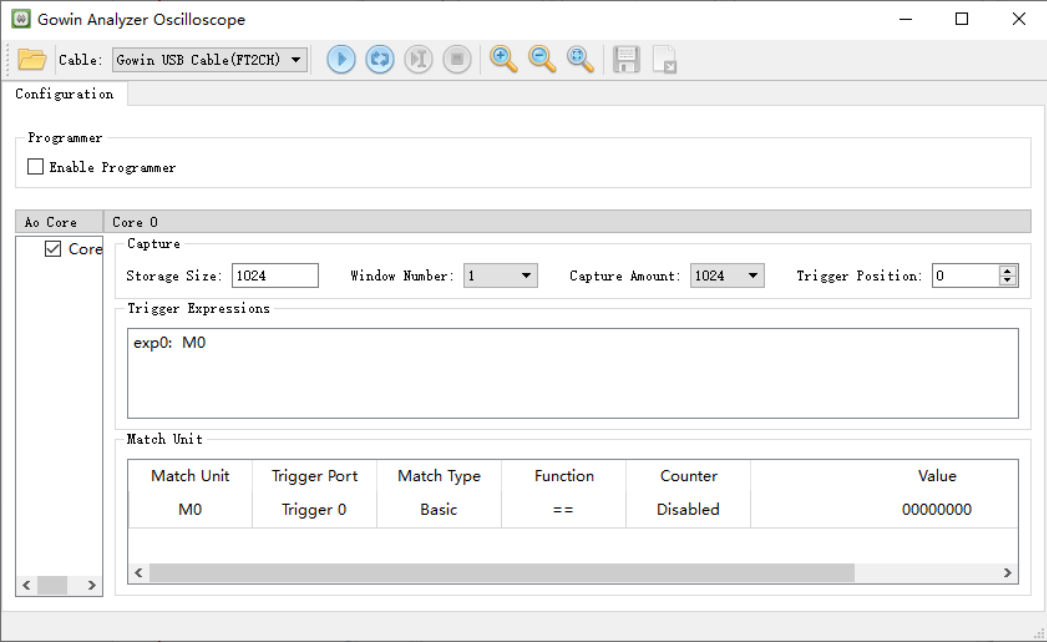


配置文件建立之后，在菜单栏中，选择“Tools > Gowin Analyzer Oscilloscope”，打开在线逻辑分析仪工具，如图 5-6 所示。

注！

在线逻辑分析仪的配置和使用方法请参考 [SUG114, Gowin 在线逻辑分析仪用户指南](#)。

图 5-6 GAO 界面



5.5 功耗分析工具

高云功耗分析工具（GPA）为用户提供尽可能准确的功耗分析，提供了丰富的用户设置选项。用户可根据设计工程的实际情况，设置影响功耗的芯片型号、工作环境以及信号翻转率等参数。功耗分析工具根据用户设置的参数，自动估算用户设计的功耗，产生功耗分析报告。

目前，GPA 工具的启动首先是以新建配置文件（.gpa）为基础，步骤如下所示：

1. 在软件工程管理区（Design），单击“File > New...”，打开“New”对话框；
2. 选择“GPA Config File”，在弹出的对话框中填写“Name”；
3. 单击“OK”，即可在“Design”窗口看到新建的 GPA Config File；
4. 双击文件名，在源文件编辑区对 GPA 配置文件进行配置，如图 5-7 所示。

注！

有关高云功耗分析工具的配置和使用方法，请参考 [SUG282, Gowin 功耗分析工具用户指南](#)。

图 5-7 GPA 配置文件窗口

The screenshot displays the GPA Configuration File window with the following settings:

- Tabs:** General Setting (selected), Rate Setting, Clock Setting.
- Operating Conditions:**
 - Grade: COMMERCIAL (dropdown)
 - Process: TYPICAL (dropdown)
- Environment:**
 - Ambient Temperature: 25.000°C (spin box)
 - ☐ Custom Theta JA: 25.000°C/W (spin box)
- Heat Sink:**
 - None (selected), Low Profile, Medium Profile, High Profile, Custom (radio buttons)
 - Air-flow: 0 (dropdown) (LFM)
 - Custom Theta SA: 25.000°C/W (spin box)
- Board Thermal Model:**
 - None (selected), Custom, Typical (radio buttons)
 - Board Temperature: 25.000°C (spin box) (-40°C-100°C)
 - Custom Theta JB: 25.000°C/W (spin box)
- Voltage:**
 - VCC: 1.000V (spin box)
 - VCCX: 2.500V (spin box)

The taskbar at the bottom shows: Start Page, Design Summary, IP Core Generator, and test.gpa.

5.6 存储器初始化文件编辑器

存储器初始化文件是一个 ASCII 文件，其扩展名为.mi；用户可根据自身设计要求，生成相应格式的初始化文件，用以指定存储器中每个地址下的初始值。如果已有.mi 文件，可以在 IDE 中用存储器初始化文件编辑器打开该.mi 文件，可以再次编辑后进行保存。

存储器初始化文件的文件名为*.mi (file_name.mi)，文件中每一行代表一个存储单元，行数即为存储单元的个数，也代表存储器的地址深度 Address Depth；列数代表每个存储单元有多少位，即内存的数据宽度 Data Width。地址从上到下依次递增，每行数据高位在前，低位在后。

高云存储器初始化文件的编辑是以新建配置文件（.mi）为基础，具体格式请参考 [UG285, Gowin 存储器\(BSRAM & SSRAM\)用户指南](#)；[UG300, Arora V 存储器\(BSRAM & SSRAM\)用户指南](#)。初始化文件编辑器具体使用步骤如下：

1. 在软件工程管理区（Design），单击“File > New...”，打开“New”对话框；
2. 选择“Memory Initialization File”，如图 5-8 所示，单击“OK”，在弹出的 New File 对话框中填写初始化文件名字后单击“OK”，如图 5-9 所示；
3. 启动如图 5-10 所示的初始化文件配置窗口，窗口左侧部分表格填写初始值，右侧部分配置初始化文件大小和视图格式；
4. 在窗口的右侧配置初始化文件的 Depth 和 Width，以及左侧表格中地址和初始值的数值显示格式；
 - Depth 和 Width 要与用户在 IP Core Generator 窗口上所选择 Block Memory 或 Shadow Memory 的 Address Depth 及 Data Width 一致，若初始化文件中 Address Depth 或 Data Width 大于窗口上所选择的值，IP Core Generator 将会提示错误信息；若 Address Depth 或 Data Width 小于各自窗口上所选择的值，则未指定的地址下的值默认初始化为 0，设置完单击“Update”；
 - 左侧表格中地址和数值的显示格式可以选择二进制，十六进制，带地址十六进制等格式。
5. 在配置窗口的左侧表格中进行初始值的写入，此外在左侧表格中可以对表格的视图格式进行设置；
 - 通过表头右击可以配置列数的显示，有 1、8、16 三种选择，如图 5-11 所示；
 - 表格中的初始值既可以通过双击后手动写入，也可以通过右击进行设置，在要输入数值处右击，选择“Fill with 0”是指初始值每位都为 0，“Fill with 1”是指初始值每位都为 1，“Custom Fill”使用户可以根据需要进行数值写入，同样也可批量设置初始值如图 5-12 所示。

6. 保存文件。

图 5-8 初始化文件新建 New 对话框

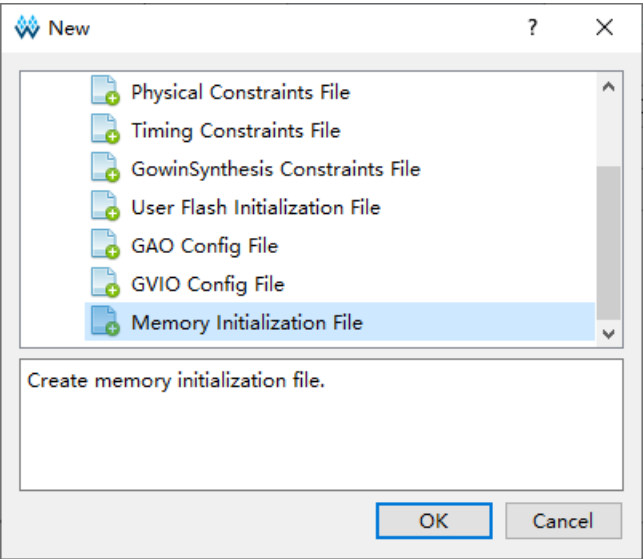


图 5-9 初始化文件新建 New File 对话框

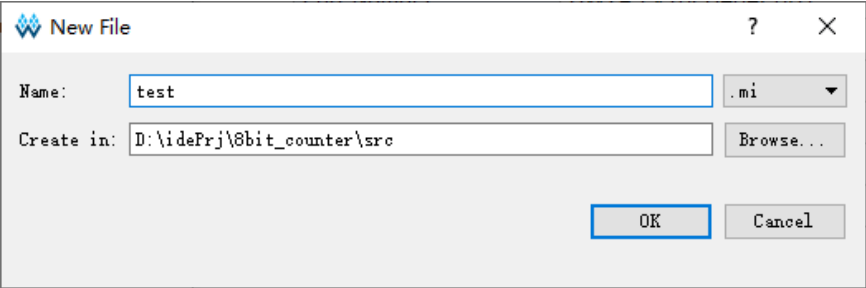


图 5-10 初始化文件配置窗口

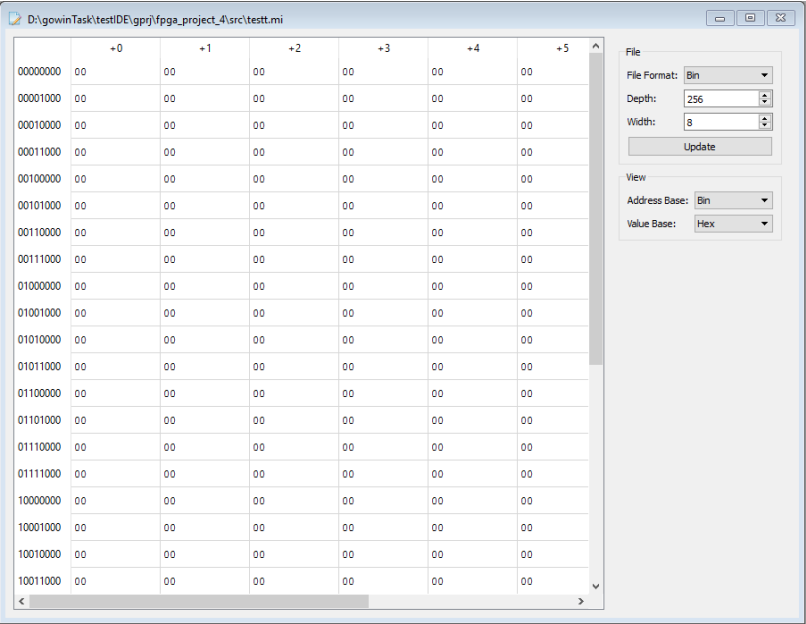


图 5-11 列数配置

	+0	+1	+2	+3	+4	+5	+6	+7	
00000000	00	00	00	00	00	00	00	00	1 Column
00001000	00	00	00	00	00	00	00	00	8 Column
									16 Column

图 5-12 批量设置

	+0	+1	+2	+3	+4	+5	+6	+7
00000000	00	00	00	00	00	00	00	00
00001000	00	00	00	00	00	00	00	00
00010000	00	00	00	00	00	00	00	00
00011000	00	00	00	00	00	00	00	00
00100000	00	00	00	00	00	00	00	00
00101000	00	00	00	00	00	00	00	00
00110000	00	00	00	00	00	00	00	00
00111000	00	00	00	00	00	00	00	00
01000000	00	00	00	00	00	00	00	00
01001000	00	00	00	00	00	00	00	00

5.7 User Flash 初始化文件编辑器

User Flash 初始化文件是一个 ASCII 文件，其扩展名为.fi；用户可根据自身设计要求，生成相应格式的初始化文件，用以指定 User Flash 中每个地址下的初始值。如果已有.fi 文件，可以在软件中用 User Flash 初始化文件编辑器打开该 fi 文件，可以再次编辑后进行保存。

User Flash 初始化文件的文件名为*.fi(file_name.fi)，文件中每一行代表一个存储单元，行数即为需要初始化数据的存储单元的个数；行头中括号内容分别表示纵坐标地址、横坐标地址，使用分号隔开；每行中括号后的内容表示存储单元初始化的数据，数据支持二进制和十六进制，数据高位在前，低位在后。.fi 文件内容格式举例如下：

5.7.1 二进制格式（Bin File）

Bin 文件是存储数据由二进制数 0 和 1 组成的文本文件。

```
//Copyright (C)2014-2024 Gowin Semiconductor Corporation.
//All rights reserved.
//File Title: User Flash Initialization File
//Tool Version: V1.9.10(64-bit)
//Part Number: GW1N-LV4PG256C6/I5
//Device-package: GW1N-4-PBGA256
//Device Version: D
//Flash Type: FLASH256K
//File Format: Bin
//Created Time: 2024-06-28 14:31:12
[0:0] 0000000000001000000001000100010000
[1:1] 000000000000100010001000000000001
```

5.7.2 十六进制格式（Hex File）

Hex 文件与 Bin 文件格式类似，存储数据由十六进制数 0~F 组成。

```
//Copyright (C)2014-2024 Gowin Semiconductor Corporation.
```

```
//All rights reserved.
```

```
//File Title: User Flash Initialization File
```

```
//Tool Version: V1.9.10 (64-bit)
```

```
//Part Number: GW1N-LV4PG256C6/I5
```

```
//Device-package: GW1N-4-PBGA256
```

```
//Device Version: D
```

```
//Flash Type: FLASH256K
```

```
//File Format: Hex
```

```
//Created Time: 2024-06-28 14:41:24
```

```
[0:0] 00101110
```

```
[1:1] 00111001
```

高云 User Flash 初始化文件的编辑是以新建配置文件（.fi）为基础，初始化文件编辑器具体使用步骤如下：

1. 在软件工程管理区（Design），单击“File > New...”，打开“New”对话框；
2. 选择“User Flash Initialization File”，如图 5-13 所示单击“OK”，在弹出的 New File 对话框中填写初始化文件名字，默认.fi 文件生成的路径是当前工程下的 src 文件夹，选择所需器件后单击“OK”，如图 5-14 所示。目前 User Flash 初始化文件编辑器所支持的器件与 User Flash 原语所支持的器件信息一致，如果选择的器件不支持 User Flash，则单击“OK”后会在 New File 对话框的底部打印提示信息“Current device do not support flash”；

图 5-13 初始化文件新建 New 对话框

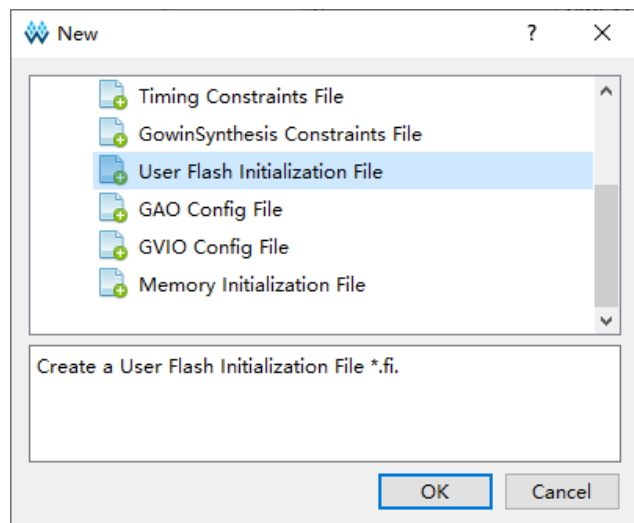
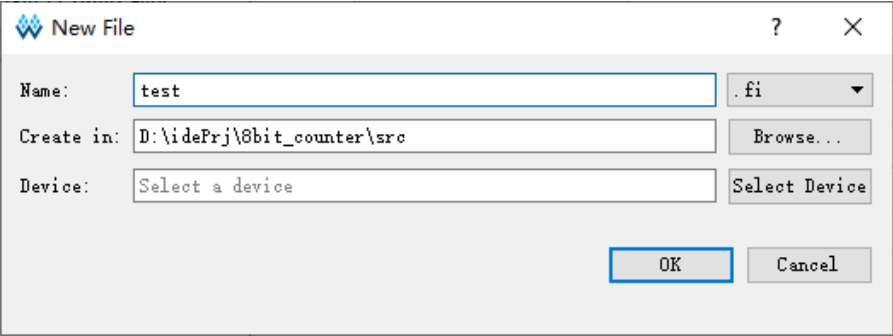
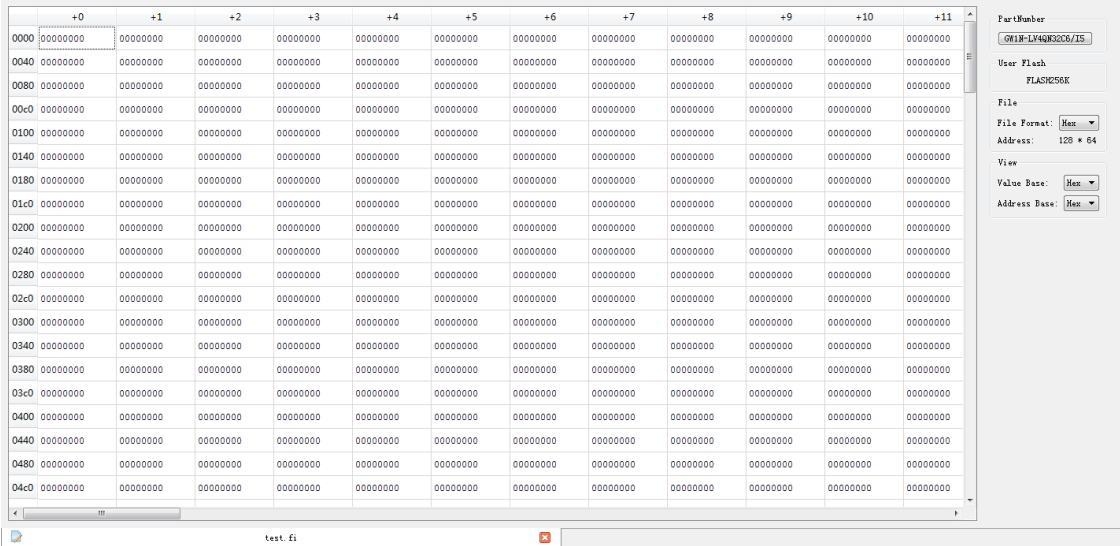


图 5-14 初始化文件新建 New File 对话框



3. 启动如图 5-15 所示的初始化文件配置窗口，窗口左侧部分表格填写初始值，右侧部分配置初始化文件格式和视图格式，并且会显示芯片型号以及 User Flash 类型；

图 5-15 初始化文件配置窗口



4. 在配置窗口的右侧配置芯片型号、初始化文件的显示格式以及左侧表格中地址和初始值的数值显示格式；
- 单击 Part Number 信息后会弹出 “Select Device” 对话框，可以重新选择其他芯片型号；
 - 左侧表格中地址和数值的显示格式可以选择二进制、八进制、十进制、十六进制等格式。
5. 在配置窗口的左侧表格中进行初始值的写入，此外在左侧表格中可以对表格的视图格式进行设置。表格中的初始值既可以通过双击后手动写入，也可以通过右击进行设置，在要输入数值处右击，选择 “Fill with 0” 是指初始值每位都为 0，“Fill with 1” 是指初始值每位都为 1，“Fill Custom” 使用户可以根据需要进行数值写入，同样也可批量设置初始值如图 5-16 所示；

图 5-16 批量设置

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11
0000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0040	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0080	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
00c0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0100	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0140	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
0180	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
01c0	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000

6. 保存文件。

5.8 原理图查看器

通过 HDL 原理图查看器（Schematic Viewer）可以直观的了解设计的逻辑构成，对后期修改设计有很好的帮助作用。Schematic Viewer 包括 RTL Design Viewer 和 Post-Synthesis Netlist Viewer，使用通用的元件符号构建电路，包括加法器、乘法器、寄存器、与门、非门和反向器等。

单击菜单栏中“Tools”下拉列表中“Schematic Viewer-RTL Design Viewer”或“Schematic Viewer-Post-Synthesis Netlist Viewer”会分别打开 RTL 设计的原理图或综合后网表的原理图。“RTL Design Viewer”窗口如图 5-17 所示，“Post-Synthesis Netlist Viewer”窗口图 5-18 所示。

图 5-17 RTL Design Viewer 窗口

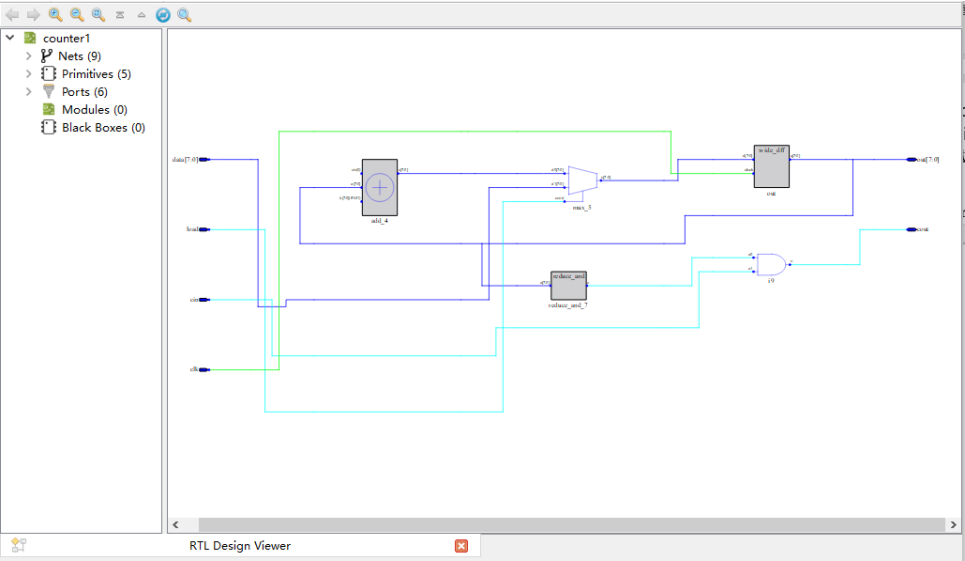
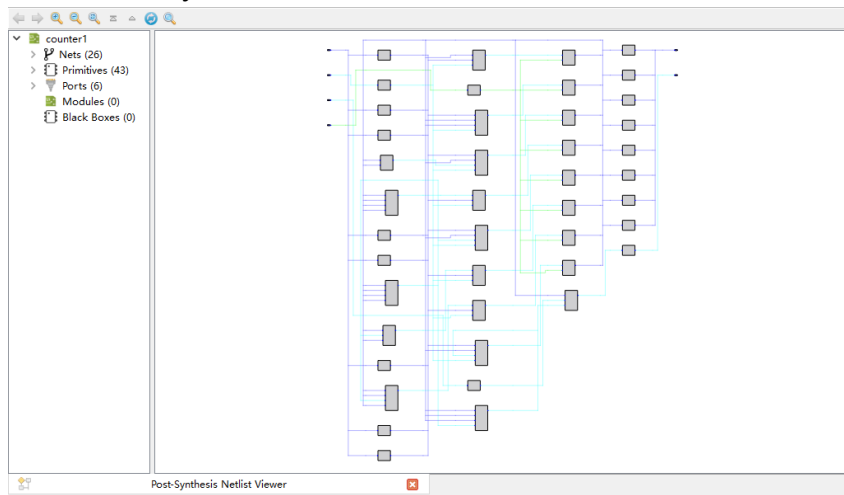


图 5-18 Post-Synthesis Netlist Viewer 窗口



在 Schematic Viewer 窗口的工具栏中，含有后退“”、前进“”、放大“”、缩小“”、缩放“”、顶层视图“”、上一层级视图“”、重新加载“”和查找“”。在窗口的左侧会显示设计的层级结构，包括构成设计的各逻辑元件，如 Modules、Ports、Nets、Primitives、Black Boxes。

注！

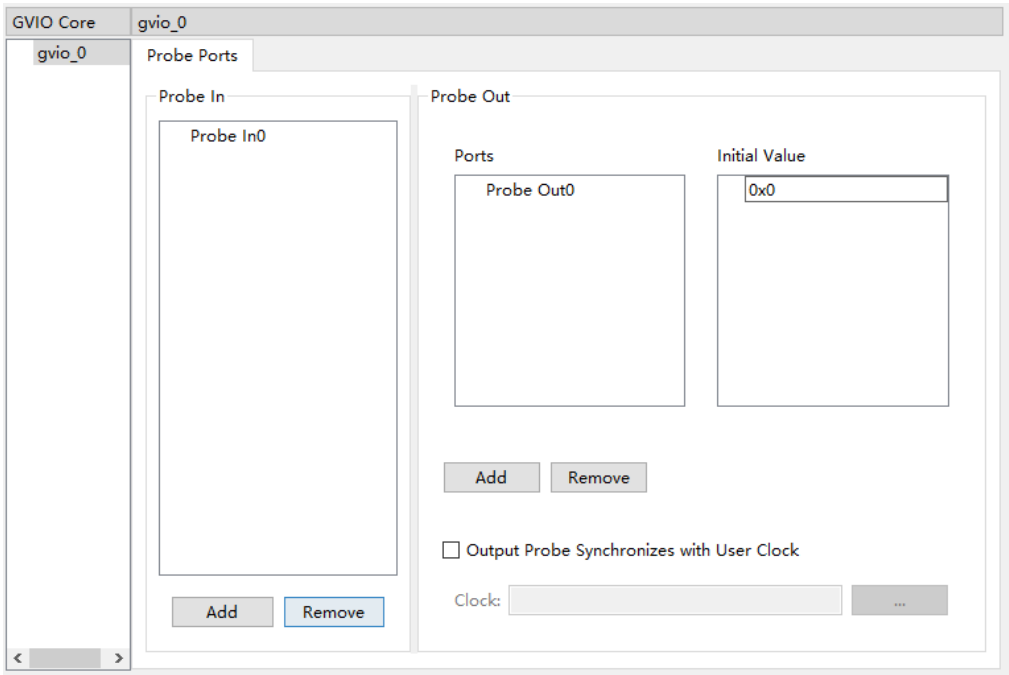
Schematic Viewer 的详细使用方法请参考 [SUG755, Gowin HDL 设计原理图查看器用户指南](#)。

5.9 虚拟输入输出调试工具

高云虚拟输入输出核（GVIO）是一款可定制内核，能够实时监控并驱动内部 FPGA 信号。其输入端口用于监控 FPGA 信号，相当于虚拟 LED，输出端口用于驱动 FPGA 信号，相当于虚拟开关。

启动 GVIO 配置文件窗口之前，需在工程管理区新建 GVIO 配置文件，打开配置文件窗口，如图 5-19 所示。

图 5-19 GVIO 配置文件窗口



注！
虚拟输入输出核的配置和使用方法请参考 [SUG1189, Gowin 虚拟输入输出调试工具用户指南](#)。

5.10 眼图分析工具 GoBert

高云眼图分析工具 **GoBert** 是高云半导体自主研发的一款用于分析 SerDes 接收信号眼图的工具，旨在帮助用户分析 SerDes 接收质量，提高用户设计的性能和可靠性。用户在使用 **GoBert** 测试接收信号质量时，需要在开发板上加载客户功能。在客户功能运行正常情况下，开启眼图测试。

点击云源中菜单栏“Tools”或工具栏中的“”按钮，启动 **GoBert** 窗口，如图 5-20 所示。

图 5-20 GoBert 窗口



注！
眼图分析工具 GoBert 的配置和使用方法请参考 [SUG1198, Gowin GoBert 工具用户指南](#)。

6云源输出文件

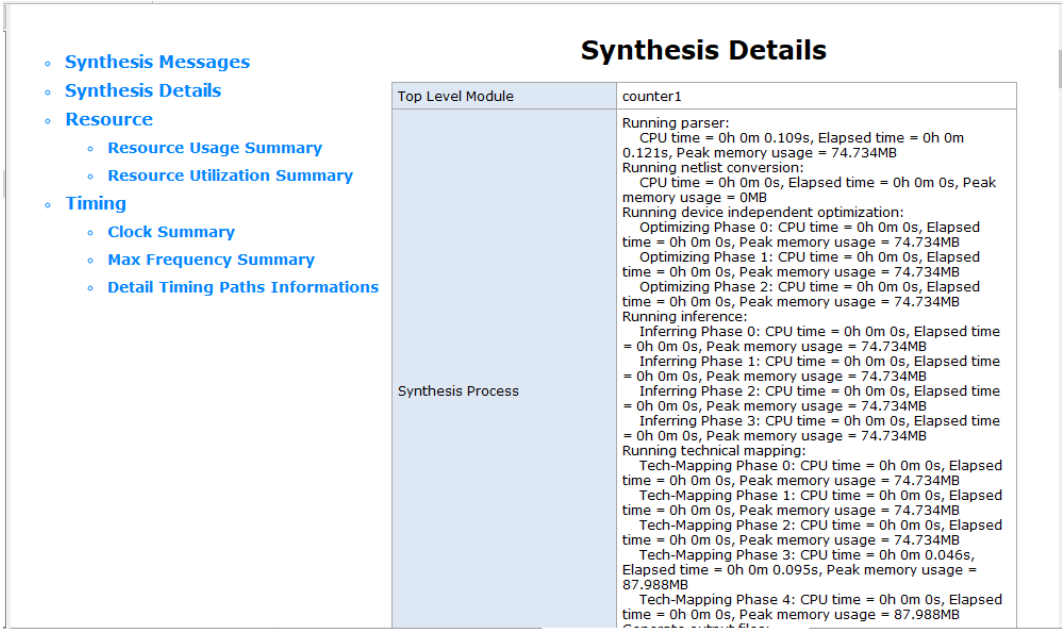
高云云源在 **FPGA** 设计过程中，除生成码流文件以外，亦可通过使用不同的运行参数，生成多个云源报告供用户参考。默认生成的报告包括综合报告、布局布线报告、端口属性报告、时序报告和功耗分析报告等。此外，用户可通过右击 **Place & Route**，修改配置选项生成管脚约束文件、时序仿真模型文件等。

6.1 综合报告

GowinSynthesis 综合完成后会生成相应的综合报告及 Netlist 文件。

报告名为 *_syn.rpt.html，包含 Synthesis Message、Synthesis Details、Resource、Timing，如图 6-1 所示。

图 6-1 GowinSynthesis 综合报告



报告的左侧会显示具体的索引信息，详细介绍如下：

- **Synthesis Message:** 综合报告基本信息，主要包括综合的设计文件、约束文件、软件版本号、器件信息、报告创建时间及法律声明等信息；

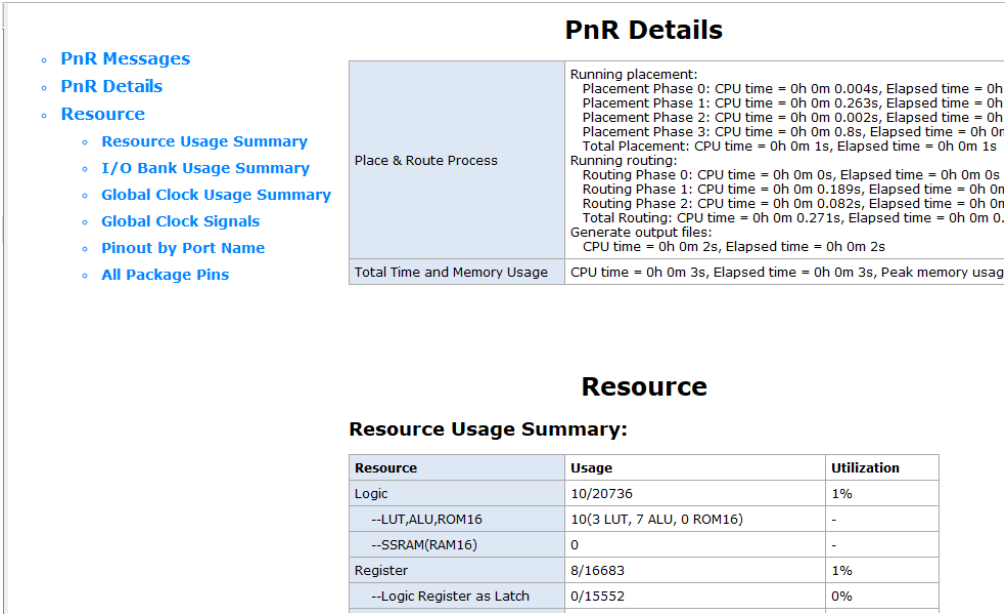
- **Synthesis Details:** 设计文件的顶层模块、综合各子阶段的实际运行时间和 CPU 运行时间以及内存占用峰值情况、综合过程总的 CPU 运行时间和内存占用峰值情况；
- **Resource:** 资源信息，主要包括资源使用统计和器件占用统计；
- **Timing:** 时序分析报告，主要包括 Clock Summary、Max Frequency Summary、Detail Timing Paths Informations 等信息。

6.2 布局布线报告

布局布线报告，列出了用户设计占用的芯片资源信息、内存消耗信息、时间消耗信息等，方便用户了解设计的大小及与目标芯片是否匹配等内容。文件扩展名名为.rpt.html，具体信息可查看*.rpt.html 文件。

用户可在过程管理区的 Place & Route 区，双击“Place & Route Report”，打开 FPGA 工程对应的布局布线报告，如图 6-2 所示。

图 6-2 Place & Route Report



布局布线报告的左侧会显示具体的索引信息，详细介绍如下：

- **PnR Messages:** 布局布线报告基本信息，包括报告名称、设计案例的路径及名称、物理约束文件、时序约束文件、软件版本号、器件信息、报告创建时间及法律声明；
- **PnR Details:**
 - place 各阶段所用时间及 place 总时间，如果工程中有 GAO，则包含 GAO place 的时间；
 - route 各阶段所用时间及 route 总时间，如果工程中有 GAO，则包含 GAO route 的时间；
 - 产生输出文件所用时间。

- Resource: 包括如下几项:
 - Resource Usage Summary: 用户设计所占用的器件资源信息;
 - I/O Bank Usage Summary: 用户设计所占用的 I/O bank 信息;
 - Global Clock Usage Summary: 所用全局时钟信息;
 - Global Clock Signals: 用户设计中所用的时钟信号;
 - Pinout by Port Name: 用户设计中的 port 所占用的引脚信息;
 - All Package Pins: 当前器件封装下所有引脚的详细信息。
- 如果工程中有 GAO, 则还包括 GAO Resource Usage Summary: 用户设计中 GAO 所占用的器件资源信息。

6.3 端口属性报告

端口属性报告, 是在布局之后输出的端口属性文件, 包括端口的类型、属性及端口位置信息等, 生成的文件扩展名名为.pin.html, 具体信息可查看 *.pin.html 文件。

用户可在过程管理区的 Place & Route 区, 双击 “Ports & Pins Report”, 打开 FPGA 工程对应的端口属性报告, 如图 6-3 所示。

图 6-3 Ports & Pins Report

- Pin Messages
- Pin Details
 - Pinout by Port Name
 - All Package Pins

Pin Details

Pinout by Port Name:

Port Name	Diff Pair	Loc./Bank	Constraint	Dir.	Site	IO Type	Drive	Pull Mode	PCI
clk		L1/7	N	in	IOL25[A]	LVC MOS18	OFF	DOWN	OFF
cout[0]		M2/7	N	out	IOL25[B]	LVC MOS18	8	NONE	OFF
cout[1]		F6/8	N	out	IOL3[A]	LVC MOS18	8	NONE	OFF
cout[2]		G7/8	N	out	IOL3[B]	LVC MOS18	8	NONE	OFF
cout[3]		D3/8	N	out	IOL2[A]	LVC MOS18	8	NONE	OFF
cout[4]		D4/8	N	out	IOL2[B]	LVC MOS18	8	NONE	OFF
cout[5]		A2/0	N	out	IOT2[B]	LVC MOS18	8	NONE	OFF
cout[6]		E6/0	N	out	IOT3[A]	LVC MOS18	8	NONE	OFF
cout[7]		F5/8	N	out	IOL4[B]	LVC MOS18	8	NONE	OFF

All Package Pins:

Loc./Bank	Signal	Dir.	Site	IO Type	Drive	Pull Mode	PCI Clamp	Hysteresis	Open Drain
B1/0	-	out	IOT2[A]	LVC MOS18	8	NONE	OFF	OFF	ON
A2/0	cout[5]	out	IOT2[B]	LVC MOS18	8	NONE	OFF	OFF	OFF
E6/0	cout[6]	out	IOT3[A]	LVC MOS18	8	NONE	OFF	OFF	OFF
F7/0	-	out	IOT3[B]	LVC MOS18	8	NONE	OFF	OFF	ON
B2/0	-	out	IOT4[A]	LVC MOS18	8	NONE	OFF	OFF	ON
A3/0	-	out	IOT4[B]	LVC MOS18	8	NONE	OFF	OFF	ON

- 端口属性报告的左侧会显示具体的索引信息, 详细介绍如下:
- Pin Messages: 端口属性报告基本信息, 包括报告名称、设计案例的路径及名称、物理约束文件、时序约束文件、软件版本号、器件信息、报告创建时间及法律声明;
 - Pin Details: 包括如下几项:
 - Pinout by Port Name: 用户设计中的 port 所占用的引脚信息;
 - All Package Pins: 当前器件封装下所有引脚的详细信息。

注！
对于 GW1N-1P5/GW1N-2/GW1NR-2 和 GW2AN-18X/GW2AN-9X 外的其他器件，在不添加 Bank V_{CCIO} 约束的情况下，端口属性报告中可能会出现某些单端输入端口 IO Type 对应的电压值与 Bank V_{CCIO} 的值不一致的情况，这是正常的。例如报告中 IO Type 为 LVCMOS18，对应的电压值为 1.8，但 Bank V_{CCIO} 为 1.2。

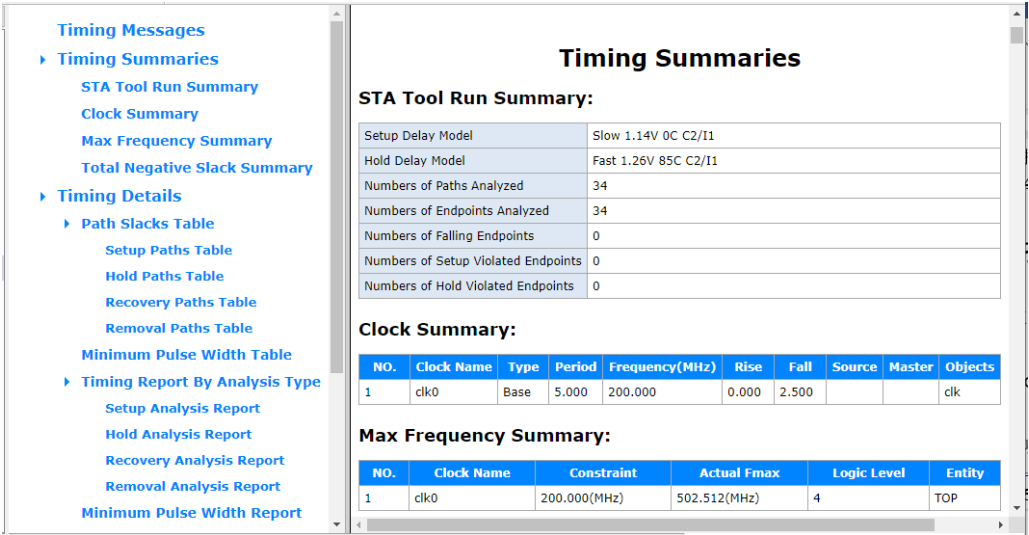
6.4 时序报告

时序报告会对电路网表中的时序模型进行全面的分析，计算电路中时序路径延迟，并判断其是否满足要求。时序报告包括建立时间检查、保持时间检查、恢复时间检查、移除时间检查、最小时钟脉冲检查、最大扇出路径、布线拥塞度报告等部分，默认情况下对上述所有检查进行报告，并提供最大频率的报告。

用户可在过程管理区的 Place & Route 区，双击“Timing Analysis Report”，打开 FPGA 工程对应的时序报告，如图 6-4 所示。

注！
时序报告的详细信息请参考 [SUG940, Gowin 设计时序约束指南](#) 中的时序报告部分。

图 6-4 时序报告内容



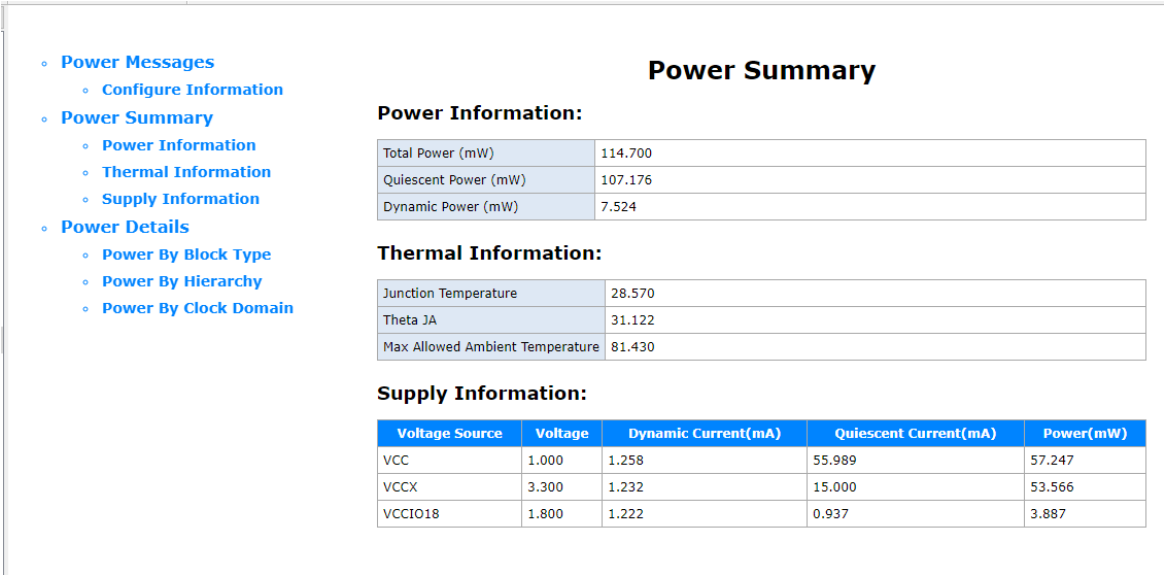
6.5 功耗分析报告

功耗分析报告主要是针对用户设计，根据器件的特性做一个预估的功耗计算，帮助用户评估设计的基本功耗值。

用户可在过程管理区的 Place & Route 区，双击“Power Analysis Report”，打开 FPGA 工程对应的功耗分析报告，如图 6-5 所示。

注！
有关功耗分析的影响因素及功耗分析报告的详细内容，请参考 [SUG282, Gowin 功耗分析工具用户指南](#)。

图 6-5 功耗分析报告



7 仿真文件

云源提供用于仿真的输入文件，仿真分为功能仿真和时序仿真，功能仿真也称为前仿真，主旨在于验证电路的功能是否符合设计要求，其特点是不考虑电路门延迟与线延迟，主要是验证电路与理想情况是否一致。

时序仿真也称为后仿真或者布局布线后仿真，是指电路已经映射到特定的工艺环境以后，综合考虑电路的路径延迟与门延迟的影响，验证电路能否在一定时序条件下满足设计构想的过程，是否存在时序违规。

7.1 功能仿真文件

功能仿真包括综合前的用户 RTL 设计功能仿真和综合后逻辑网表功能仿真。以 Verilog 语言描述的设计功能仿真为例，所需要的文件有：综合前用户设计 RTL 文件、综合后网表文件 (*.vg)、激励文件 (testbench) *tb.v 以及功能仿真库文件 prim_sim.v。

注！

- 仿真库文件所在目录：installPath\IDE\simlib;
- Vhdl 格式的仿真库文件名为 prim_sim.vhd;
- 由于产生的软核 IP 为密文，因此，设计中含有软核 IP 时，需要将软核 IP 产生后的.vo/.vho 文件作为功能仿真文件，.vo/.vho 文件在当前工程目录 src 下产生的软核 IP 目录 srcIipName 下。

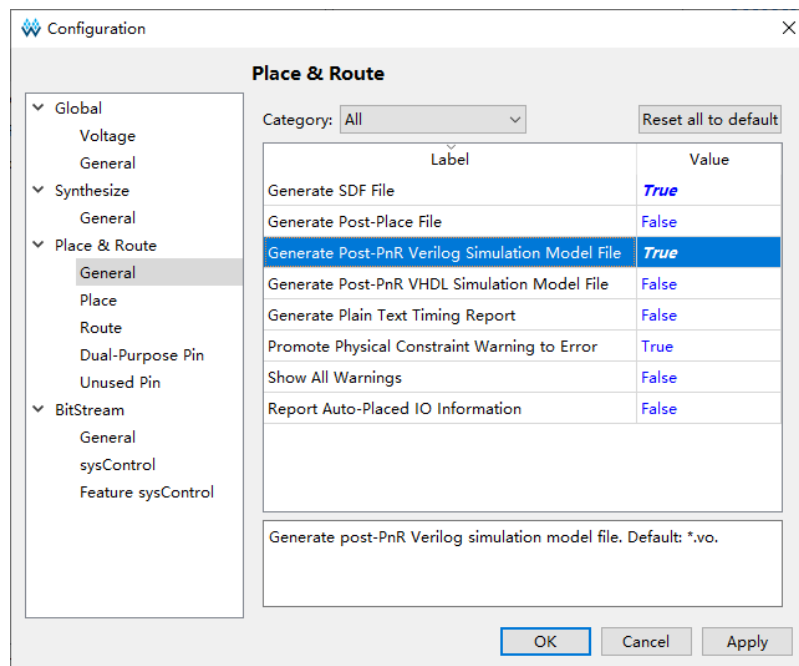
7.2 时序仿真文件

以 Verilog 语言描述的设计时序仿真为例，时序仿真所需要的文件有：软件产生的 Verilog 语言的时序仿真逻辑网表文件*.vo、对应的延时文件*.sdf、对应的激励文件*tb.v 以及时序仿真库 prim_tsim.v。

时序仿真逻辑网表文件*.vo 和延时文件*.sdf 可以通过云源界面运行工程生成。具体步骤如下：

1. 建立/打开工程后，Configuration 进行设置，在 Place & Route 选项中把“Generate SDF File”和“Generate Post-PNR Verilog Simulation Model File”的 Value 设置成 True，单击“OK”，如图 7-1 所示。

图 7-1 选项配置



2. 运行 Place & Route，运行成功后在工程所在路径下的 impl/pnr/下可以找到所需要的 vo 和 sdf 文件。

8Tcl 命令说明

云源支持命令行模式运行。在以下说明中，尖括号<>包含的内容为必须指定的内容，方括号[]包含的内容为可选的内容。创建文件名称时不支持“/”、“*”、“-”、空格等特殊字符，同时指定 IP module_name 时也不支持数字开头。

8.1 启动命令行模式

8.1.1 gw_sh.exe

语法

命令：以 Windows 系统为例，启动安装目录下\X.X\IDE\bin\gw_sh.exe

参数：

[script file]

参数为空：直接进入命令行控制台模式；

script file：可选项，执行一个指定的脚本文件。

应用示例：

#启动命令行模式

gw_sh.exe

#执行脚本文件，具体示例请参考 [SUG918, Gowin 云源软件快速入门指南](#)的 Tcl 命令快速入门指南章节。

gw_sh.exe script_file

8.2 命令

8.2.1 命令分类

[IPFlow](#)

[Project](#)

8.2.2 命令列表

IPFlow:

[create_ipc](#)

[generate_target](#)

[get_ips](#)

[list_property](#)

[read_ipc](#)

[report_property](#)

[set_property](#)

[source](#)

[write_ip_tcl](#)

Project:

[add_file](#)

[create_project](#)

[import_files](#)

[open_project](#)

[rm_file](#)

[run](#)

[run_close](#)

[saveto](#)

[set_device](#)

[set_file_enable](#)

[set_file_prop](#)

[set_option](#)

[source](#)

8.3 命令介绍

8.3.1 add_file

指定要添加到工程的设计文件。

语法

add_file [-type] [-disable] [-h/--help] <file>

使用

名称	描述
[-type]	添加的设计文件类型
[-disable]	设置为失效状态
[-h/--help]	显示帮助信息
<file>	要添加的文件

分类

[Project](#)

描述

添加设计文件，Windows 系统和 Linux 系统下文件路径的分隔符支持两种书写形式：/或\\。支持相对路径和绝对路径，相对路径形式在云源界面下相对的是当前工程的路径，在命令行模式下相对的是启动 gw_sh 时的路径。

参数

- <file>要添加的设计文件，可以指定多个，多个文件以空格分隔；
- [-type <type>] add_file 命令会根据文件扩展名自动判断文件类型，也可以使用该选项来明确指定文件类型，支持的文件类型有 verilog、vhdl、sv、vg、cst、sdc、gao、gpa、gsc 等；
- [-disable]将添加的文件置为失效状态。处于失效状态的文件仅添加到文件列表中，不参与流程实现。相关命令：set_file_enable；
- [-h/--help]显示帮助信息。

示例

```
add_file abc.v
add_file -type vhdl 1.vhd 2.vhdl 3.vhd
add_file D:/gowin_project/top.v
add_file D:\\gowin_project\\top.v
```

8.3.2 create_ipc

生成默认配置的 ipc 文件。

语法

```
create_ipc -name <ipName> -module_name <moduleName> [-  
language <arg>] [-file_name <fileName>] [-dir <path>] [-force]
```

使用

名称	描述
-name	IP name
-module_name	将要创建的IP的module名称
[-language]	IP文件、模板文件以及仿真文件的语言
[-file_name]	IP文件的名称
[-dir]	IP生成路径
[-force]	覆盖已有文件

分类

[IPFlow](#)

描述

该命令为 IPCoreGenerator 中的 IP 核创建 IPC 文件。

参数

- -name <ipName>指定 IP 的名称，该名称可通过 IP Core Generator 获取；
- -module_name <moduleName>指定要创建的 IP 的 module 名称；
- [-language <arg>]指定生成的 IP 文件、模板文件、仿真文件的语言（Verilog/VHDL），若不指定则默认使用 Verilog；
- [-file_name <fileName>]指定生成的 IP 文件的名称，若不指定则与 module_name 指定的名称一致；
- [-dir] <path>指定 IP 生成路径，如果没有指定，生成到当前工程的 src 文件夹下；
- [-force]覆盖已有的文件。

示例

下面的示例在当前工程的 src 文件夹下创建由-name 指定的 IP 核，并指定模块名称、语言以及文件名称：

```
create_ip -name fifo -module_name FIFO_Top -language Verilog -  
file_name fifo
```

另请参阅

[generate_target](#)

8.3.3 create_project

创建工程。

语法

```
create_project [-name <prjName>] [-dir <path>] [-pn <pnName>] [-device_version <arg>] [-force] [-h/--help]
```

使用

名称	描述
[-name]	要创建工程的名称
[-dir]	要创建的工程路径
[-pn]	要创建工程的Part Number
[-device_version]	要创建工程的device version
[-force]	覆盖已有文件
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

新建工程，文件路径形式请参考 [8.3.1 add_file](#)。

参数

- [-name <prjName>]指定要创建工程的名称；
- [-dir <path>]指定创建的工程所在的路径，如果指定的路径不存在则创建一个新路径。如果没有指定该选项，则默认使用当前工作路径或 tcl 脚本所在的路径；
- [-pn <pnName>]指定创建工程的 Part Number；
- [-device_version <arg>]指定 Part Number 对应的 device version，对于只有初始版本的器件，device version 的值为 NA；
- [-force]该选项会覆盖已有的工程；
- [-h/--help]显示帮助信息。

示例

```
create_project -name prj0 -dir D:/tclprj -pn GW1N-UV4LQ144C6/15 -device_version B
create_project -name prjlest -pn GW1N-UV4LQ144C6/15 -device_version B -force
```

8.3.4 generate_target

为指定的对象生成目标文件。

语法

```
generate_target <objects> [-force]
```

使用

名称	描述
<objects>	指定要产生目标文件的对象
[-force]	覆盖已有文件

分类

[IPFlow](#)

描述

该命令为指定的 IP 对象（`get_ips`）生成目标文件，并将 IP 核设计文件添加到当前工程中。

参数

- <objects>用于指定生成设计文件的一个或多个 IP 对象，一个对象通过 `[get_ips module_name]`指定，如果是指定多个对象，则使用`[get_ips module_name0 module_name1 ...]`。
- [-force]覆盖已有文件

示例

下面的示例为指定的 IP 生成设计文件，模板文件并加载到当前工程中：

```
generate_target [get_ips FIFO_Top]
```

另请参阅

- [generate_target](#)
- [create_ipc](#)
- [read_ipc](#)

8.3.5 get_ips

指定 IP 对象。

语法

```
get_ips <module_name>
```

使用

名称	描述
<module_name>	指定 IP 的 module_name

分类

[IPFlow](#)

描述

指定当前工程中的 IP 对象。

参数

<module_name>指定 IP 对象的 module_name，可以是一个也可以是多个。

示例

下面的示例为指定当前工程的一个 IP 对象：

```
get_ips FIFO_Top
```

下面的示例为指定当前工程的多个 IP 对象：

```
get_ips FIFO_Top FIFO_Top_1
```

另请参阅

- [generate_target](#)
- [list_property](#)
- [report_property](#)
- [set_property](#)

8.3.6 import_files

拷贝文件或目录到当前工程。

语法

```
import_files [-file <file>] [-dir <path>] [-fileList <fileList>] [-force] [-h/--help]
```

使用

名称	描述
[-file]	指定要拷贝的文件
[-dir]	指定要拷贝的路径
[-fileList]	指定列表文件
[-force]	覆盖已有同名文件
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

将文件或目录拷贝到当前工程路径/src 下。 -file、-dir 和-fileList 后面的 value 可以跟绝对路径，也可以是相对路径。如果是通过 gwsh.exe 执行 tcl 脚本，则相对的是 tcl 脚本的路径，如果是通过云源软件中的 console 窗口执行 import_files 命令，则相对的是当前工作目录(pwd)。如果 impot_files 后面没有跟任何选项，则默认将 add_file 命令指定的所有文件拷贝到工程路径/src 下。

参数

- [-file <file>]将一个或多个文件添加到工程路径/src 下；
- [-dir <path>]将路径下的所有文件及子文件夹添加到工程路径/src 下；
- [-fileList <fileList>]指定列表文件，文件中每一行为一个要添加的工程文件，内容示例如下：
 - D:/test1.v
 - D:/test2.v
 - 该选项可将文件中每一行所指定的文件都添加到工程路径/src 下；
- [-force]覆盖工程路径/src 下的同名文件；
- [-h/--help] 显示帮助信息。

示例

```
import_files -file D/test1 .v -force
import_files -file D:/test1 .v D:/test2.v -force
import_files -dir D:/sourceFile
import_files -fileList log, log 文件的内容为:
D:/Test1.v
D:/Test2.v
```

8.3.7 list_property

列举对象的属性。

语法

```
list_property <object>
```

使用

名称	描述
<object>	查询属性的对象

分类

[IPFlow](#)

描述

获取指定 IP 对象的所有选项的列表。

参数

<object>指定查询选项的某个 IP，通过[get_ips module_name]指定。

示例

下面的示例列出指定 IP 对象的所有属性：

```
list_property [get_ips FIFO_Top]
```

另请参阅

- [report_property](#)
- [set_property](#)

8.3.8 open_project

打开工程。

语法

```
open_project <file>] [-pn] [-device_version] [-h/--help]
```

使用

名称	描述
<file>	指定工程文件
[-pn]	指定Part Number信息
[-device_version]	指定工程的device version
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

打开工程，可以对要打开的工程指定一个新的 Part Number。文件路径形式请参考 [8.3.1 add_file](#)。

参数

- <file>指定要打开的工程文件名称；
- [-pn] 指定工程的 Part Number 信息；
- [-device_version] 指定工程器件的版本号信息；
- [-h/--help] 显示帮助信息。

示例

```
open_project D:\test.gprj
```

8.3.9 read_ipc

读取 ipc 文件。

语法

```
read_ipc <file>
```

使用

名称	描述
<file>	IPC 文件

分类

[IPFlow](#)

描述

读取指定的 IPC 文件，通过 IPC 文件配置选项，文件路径形式请参考[8.3.1 add file](#)。

参数

<file>指定要打开的 IPC 文件名称。

示例

下面的示例为读取指定路径下的 IPC 文件：

```
read_ipc D:/gowin_project/src/fifo/fifio.ipc
```

另请参阅

[generate_target](#)

8.3.10 report_property

报告对象的属性。

语法

```
report_property <object>
```

使用

名称	描述
<object>	查询属性的对象

分类

[IPFlow](#)

描述

获得指定 IP 对象的选项名称，选项类型以及选项值。

参数

<object>指定查询选项的某个 IP，通过[get_ips module_name]指定。

示例

下面的示例列出指定 IP 对象的所有属性：

```
report_property [get_ips FIFO_Top]
```

另请参阅

- [set_property](#)
- [list_property](#)

8.3.11 rm_file

移除设计文件。

语法

```
rm_file [-h/--help ] <files>
```

使用

名称	描述
<-files>	要移除的设计文件
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

移除设计文件，文件路径形式请参考 [8.3.1 add_file](#)。

参数

- <-files>指定要移除的设计文件，可以指定多个文件，以空格分隔；
- [-h/--help] 显示帮助信息。

示例

```
rm_file a.v
rm_file a.v b.v c.v
rm_file D:/gowin_project/top.v
rm_file D:\\gowin_project\\top.v
```

8.3.12 run

运行流程。

语法

```
run [-h/--help] <syn/pnr/all>
```

使用

名称	描述
<syn/pnr/all>	指定要运行的流程名称
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

运行全部流程或某个流程。

参数

- <syn/pnr/all> 指定要运行的流程名称，可以被执行的流程名称有 **syn** 和 **pnr**，分别表示综合及布局布线。也可以指定 **all**，表示运行全部的流程；
- [-h/--help] 显示帮助信息。

示例

```
run pnr
```

```
run all
```

8.3.13 run close

关闭工程。

语法

```
run close
```

分类

[Project](#)

描述

关闭当前工程。

示例

```
run close
```

8.3.14 saveto

保存工程数据到 Tcl 脚本。

语法

```
saveto [-all_options] [-h/--help] <file>
```

使用

名称	描述
[-all_options]	保存所有的选项信息
[-h/--help]	显示帮助信息
<file>	保存的文件名

分类

[Project](#)

描述

将当前工程设计数据保存到 Tcl 脚本。文件路径形式请参考 [8.3.1 add file](#)。

参数

- [-all_options] saveto 命令默认只保存修改过的选项信息，即与默认值不同的选项。可以通过指定-all_options 来保存所有的选项信息；
- [-h/--help] 显示帮助信息；
- <file> 保存的文件名。

示例

```
saveto project.tcl
saveto -all_options project.tcl
saveto -all_options D:/gowin_project/project.tcl
saveto -all_options D:\\gowin_project\\project.tcl
```

8.3.15 set_device

设置芯片型号。

语法

```
set_device [-device_version <value>] [-h/--help ] <part number>
```

使用

名称	描述
[-device_version <value>]	要设置的device version
[-h/--help]	显示帮助信息
<part number>	要设置的Part Number

分类

[Project](#)

描述

设置芯片型号。

参数

- <part number> 指定目标器件的 Part Number，如 GW1N-UV4LQ144C6/I5；
- [-device_version<value>] 指定器件版本号信息，支持的版本号取值有 NA|B|C|D；
- [-h/--help] 显示帮助信息。

示例

```
set_device GW1N-LV1CS30C6/I5
set_device -device_version C GW1N-UV4LQ144C6/I5
```

8.3.16 set_file_enable

设置文件使能属性。

语法

```
set_file_enable <file> <true|false> [-h/--help]
```

使用

名称	描述
<file>	指定要设置的设计文件
<true false>	文件是否被使用
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

设置文件是否可以被使用，文件路径形式请参考 [8.3.1 add_file](#)。

参数

- <file> 指定要设置的文件；
- <true|false> true 表示文件可以被使用，false 表示不可被使用；
- [-h/--help] 显示帮助信息。

示例

```
set_file_enable top.v false
set_file_enable D:/gowin_project/top.v
set_file_enable D:\\gowin_project\\top.v
```

8.3.17 set_file_prop

设置文件属性。

语法

```
set_file_prop <file> [-lib <name>] [-h/--help]
```

使用

名称	描述
<file>	指定要设置的设计文件
[-lib <name>]	设置文件的library name
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

设置文件属性，文件路径形式请参考 [8.3.1 add_file](#)。

参数

- <file> 指定要设置的文件，可以指定多个，多个文件之间以空格分隔；
- [-lib <name>] 设置文件的 library name。该选项仅对 VHDL 类型的文件有效；
- [-h/--help] 显示帮助信息。

示例

```
set_file_prop -lib work top .vhd
set_file_prop -lib work D:/gowin_project/top.vhd
set_file_prop -lib work D:\\gowin_project\\top.vhd
```

8.3.18 set_option

设置工程相关的属性配置及流程选项。

语法

```
set_option [options] [-h/--help]
```

使用

名称	描述
[options]	指定配置或流程选项
[-h/--help]	显示帮助信息

分类

[Project](#)

描述

设置工程相关的属性配置及流程选项。

参数

- [options] 指定配置或流程选项；
- [-h/--help] 显示帮助信息。

Global 属性配置

-output_base_name

指定输出文件的文件名。

语法

-output_base_name <name>

使用

名称	描述
<name>	指定输出文件的文件名

分类

[Project](#)

描述

指定输出文件的文件名。此选项仅指定文件的 **base name**，不同流程会根据输出文件的类型使用合适的扩展名。如-output_base_name abc，则gowinsynthesis 综合产生的网表文件名为 abc.vg；

参数

<name>指定输出文件的文件名。

示例

```
set_option -output_base_name abc
```

-global_freq

指定 frequency 值。

语法

-global_freq <default|value>

使用

名称	描述
<default value>	指定frequency值

分类

[Project](#)

描述

指定 **frequency** 值，默认为 **default**（小蜜蜂家族芯片默认是 50MHz，晨熙家族芯片默认是 100MHz）。

参数

<default|value>frequency 值。

示例

```
set_option -global_freq 80
```

综合属性配置

-synthesis_tool

指定综合工具。

语法

-synthesis_tool <tool>

使用

名称	描述
<tool>	指定综合工具

分类

[Project](#)

描述

指定综合工具 GowinSynthesis。

参数

<tool>指定综合工具 GowinSynthesis。

示例

```
set_option -synthesis_tool GowinSynthesis
```

-top_module

指定 Top Module/Entity。

语法

-top_module <name>

使用

名称	描述
<name>	指定top module

分类

[Project](#)

描述

指定 top module。

参数

<name>指定 top module。

示例

```
set_option -top_module test
```

-include_path

指定包含路径。

语法

-include_path <path or path list>

使用

名称	描述
<path or path list>	指定包含路径

分类

[Project](#)

描述

指定包含路径。当指定多个包含路径时，路径之间需要使用分号进行分隔，并使用一对大括号{}包含所有的路径信息，如**-include_path** {/path1;/path2;/path3}。支持相对路径和绝对路径，相对路径为相对程序当前运行路径。

参数

<path or path list>指定包含路径。

示例

```
set_option -include_path D:/project
```

-verilog_std

指定 Verilog 语言标准。

语法

-verilog_std<v1995|v2001|sysv2017>

使用

名称	描述
<v1995 v2001 sysv2017>	指定Verilog语言标准

分类

[Project](#)

描述

指定 Verilog 语言标准：Verilog 95/Verilog 2001/System Verilog 2017，默认为 Verilog 2001。

参数

<v1995|v2001|sysv2017>指定 Verilog 语言标准。

示例

```
set_option -verilog_std v1995
```

-vhdl_std

指定 VHDL 语言标准。

语法

-vhdl_std <vhdl1993|vhdl2008|vhdl2019>

使用

名称	描述
<vhdl1993 vhdl2008 vhdl2019>	指定VHDL语言标准

分类

[Project](#)

描述

指定 VHDL 语言标准：VHDL 1993/VHDL 2008/VHDL 2019，默认为 VHDL1993。

参数

<vhdl1993|vhdl2008|vhdl2019>指定 VHDL 语言标准。

示例

```
set_option -vhdl_std vhdl2008
```

-print_all_synthesis_warning <0|1>

指定是否打印所有的综合 warning 信息，默认为 0。

语法

-print_all_synthesis_warning <0|1>

使用

名称	描述
<0 1>	0: 不会打印所有的warning信息; 1: 会打印所有的warning信息。

分类

[Project](#)

描述

指定是否打印所有的综合 warning 信息，默认为 0。

参数

<0|1>指定是否打印所有的综合 warning 信息。

示例

```
set_option -print_all_synthesis_warning 1
```

-disable_io_insertion

启用或禁用 I/O 插入的使能控制。

语法

```
-disable_io_insertion <0|1>
```

使用

名称	描述
<0 1>	0: 启用I/O插入的使能控制; 1: 禁用I/O插入的使能控制。

分类

[Project](#)

描述

启用或禁用 I/O 插入的使能控制，默认为 0。

参数

<0|1>启用或禁用 I/O 插入的使能控制。

示例

```
set_option -disable_io_insertion 1
```

-looplimit <value>

RTL 中默认的编译器循环限制值。

语法

```
-looplimit <value>
```

使用

名称	描述
<value>	looplimit值

分类

[Project](#)

描述

设置 RTL 中默认的编译器循环限制值，默认值为 2000。

参数

<value> RTL 中默认的编译器循环限制值。

示例

set_option -looplimit 1000

-maxfan <value>

设置扇出值。

语法

-maxfan <value>

使用

名称	描述
<value>	maxfan值

分类

[Project](#)

描述

设置一个输入端口、net 或寄存器输出端的扇出值，默认为 10000。

参数

<value>输入端口、net 或寄存器输出端的扇出值。

示例

set_option -maxfan 5000

-rw_check_on_ram

在 RAM 周围插入旁路逻辑。

语法

-rw_check_on_ram <0|1>

使用

名称	描述
<0 1>	0: 不启用该选项; 1: 启用该选项。

分类

[Project](#)

描述

如果 RAM 存在读或写冲突，使用该选项后会在 RAM 周围插入旁路逻辑以防止仿真不匹配，默认为 0。

参数

<0|1>启用或禁用在 RAM 周围插入旁路逻辑使能控制。

示例

set_option -rw_check_on_ram 1

Place & Route 属性配置

-VCCX

指定 vccx 值

语法

-vccx <value>

使用

名称	描述
<value>	指定vccx值

分类

[Project](#)

描述

指定 vccx 值。

参数

<value>指定 vccx 值。

示例

```
set_option -vccx 3.3
```

-VCC

语法

-vcc <value>

使用

名称	描述
<value>	指定vcc值

分类

[Project](#)

描述

指定 vcc 值。

参数

<value>指定 vcc 值。

示例

```
set_option -vcc 3.3
```

-gen_sdf

是否产生 SDF 文件。

语法

-gen_sdf <0|1>

使用

名称	描述
<0 1>	0: 不产生SDF文件; 1: 产生SDF文件。

分类

[Project](#)

描述

指定 Place & Route 是否产生 SDF 文件，默认为 0。

参数

<0|1>产生 SDF 文件使能控制。

示例

```
set_option -gen_sdf 1
```

-gen_io_cst

是否产生文件名为*.io.cst 的 port 端口的物理约束文件。

语法

-gen_io_cst <0|1>

使用

名称	描述
<0 1>	0: 不产生*.io.cst文件; 1: 产生*.io.cst文件。

分类

[Project](#)

描述

指定 Place & Route 是否产生文件名为*.io.cst 的 port 端口的物理约束文件，默认为 0。

参数

<0|1>产生 port 端口的物理约束文件使能控制。

示例

```
set_option -gen_io_cst 1
```

-gen_ibis

是否产生文件名为*.ibs 的输入/输出缓冲区信息指定文件。

语法

-gen_ibis <0|1>

使用

名称	描述
<0 1>	0: 不产生*.ibs文件; 1: 产生*.ibs文件。

分类

[Project](#)

描述

指定 **Place & Route** 是否产生文件名为*.ibs 的输入/输出缓冲区信息指定文件，默认为 0。

参数

<0|1>产生输入/输出缓冲区信息指定文件使能控制。

示例

```
set_option -gen_ibis 1
```

-gen_posp

是否产生器件布局文件。

语法

```
-gen_posp <0|1>
```

使用

名称	描述
<0 1>	0: 不产生*.posp文件; 1: 产生*.posp文件。

分类

[Project](#)

描述

指定 **Place & Route** 是否产生文件名为*.posp 的器件布局文件，文件中只含有 BSRAM 布局信息，默认为 0。

参数

<0|1>产生器件布局文件使能控制。

示例

```
set_option -gen_posp 1
```

-gen_text_timing_rpt

产生文本格式的时序报告。

语法

```
-gen_text_timing_rpt <0|1>
```

使用

名称	描述
<0 1>	0: 不产生*.tr文件; 1: 产生*.tr文件。

分类

[Project](#)

描述

指定 Place & Route 是否产生文件名为*.tr 的文本格式的时序报告，默认为 0。

参数

<0|1>产生文本格式的时序报告使能控制。

示例

```
set_option -gen_text_timing_rpt 1
```

-gen_verilog_sim_netlist

是否产生 Verilog 语言的时序仿真模型文件。

语法

```
-gen_verilog_sim_netlist <0|1>
```

使用

名称	描述
<0 1>	0: 不产生*.vo文件; 1: 产生*.vo文件。

分类

[Project](#)

描述

指定 Place & Route 是否产生文件名为*.vo 的 Verilog 语言的时序仿真模型文件，默认为 0。

参数

<0|1>产生 Verilog 语言的时序仿真模型文件使能控制。

示例

```
set_option -gen_verilog_sim_netlist 1
```

-gen_vhdl_sim_netlist

是否产生 VHDL 语言的时序仿真模型文件。

语法

```
-gen_vhdl_sim_netlist <0|1>
```

使用

名称	描述
<0 1>	0: 不产生*.vho文件; 1: 产生*.vho文件。

分类

[Project](#)

描述

指定 **Place & Route** 是否产生文件名为*.vho 的 VHDL 语言的时序仿真模型文件，默认为 0。

参数

<0|1>产生 VHDL 语言的时序仿真模型文件使能控制。

示例

```
set_option -gen_vhdl_sim_netlist 1
```

-show_init_in_vo

将默认初始值添加到时序仿真模型文件中的 **instance** 中。

语法

```
-show_init_in_vo <0|1>
```

使用

名称	描述
<0 1>	0: 不会将默认初始值添加到时序仿真模型文件中的instance中; 1: 将默认初始值添加到时序仿真模型文件中的instance中。

分类

[Project](#)

描述

将默认初始值添加到生成的 **Place & Route** 时序仿真模型文件中的 **instance** 中，默认为 0。

参数

<0|1>将默认初始值添加到生成的 **Place & Route** 时序仿真模型文件中的 **instance** 中的使能控制。

示例

```
set_option -show_init_in_vo 1
```

-show_all_warn

是否输出所有的 **warning** 信息。

语法

-show_all_warn<0|1>

使用

名称	描述
<0 1>	0: 不会输出Place & Route运行时所有的warning信息; 1: 输出Place & Route运行时所有的warning信息。

分类

[Project](#)

描述

Place & Route 运行时输出所有的 warning 信息，默认为 0。

参数

<0|1>输出 Place & Route 运行时所有的 warning 信息使能控制。

示例

```
set_option -show_all_warn 1
```

-timing_driven

是否对布局布线进行时序驱动优化。

语法

-timing_driven <0|1>

使用

名称	描述
<0 1>	0: 不会进行布局布线时序驱动优化; 1: 进行布局布线时序驱动优化。

分类

[Project](#)

描述

Place & Route 运行时对布局布线进行时序驱动优化，默认为 1。

参数

<0|1>运行布局布线时是否启动时序驱动优化使能控制。

示例

```
set_option -timing_driven 1
```

-cst_warn_to_error

将物理约束警告提升为错误信息。

语法

-cst_warn_to_error <0|1>

使用

名称	描述
<0 1>	0: 不会将物理约束警告提升为错误信息; 1: 将物理约束警告提升为错误信息。

分类

[Project](#)

描述

将 Place & Route 运行时的物理约束警告提升为错误信息，默认为 1。

参数

<0|1>是否将 Place & Route 运行时的物理约束警告提升为错误信息使能控制。

示例

set_option -cst_warn_to_error 1

-rpt_auto_place_io_info

报告自动 place 的 IO 位置信息。

语法

-rpt_auto_place_io_info <0|1>

使用

名称	描述
<0 1>	0: 不会报告自动place的IO位置信息; 1: 报告自动place的IO位置信息。

分类

[Project](#)

描述

Place & Route 运行时报告自动 place 的 IO 位置信息，默认为 0。

参数

<0|1>是否报告自动 place 的 IO 位置信息使能控制。

示例

set_option -cst_warn_to_error 1

-place_option

布局算法选项。

语法

-place_option <0|1|2|3|4>

使用

名称	描述
<0 1 2 3 4>	0: 采用默认布局算法; 1: 采用布局算法 1; 2: 采用布局算法 2。 3: 采用布局算法 3; 4: 采用布局算法 4。

分类

[Project](#)

描述

布局算法选项，默认为 0。

参数

<0|1|2|3|4>布局算法选项。

示例

set_option -place_option 1

-route_option

布线算法选项。

语法

-route_option <0|1|2>

使用

名称	描述
<0 1 2>	0: 采用默认布线算法; 1: 采用布线算法1; 2: 采用布线算法2。

分类

[Project](#)

描述

布线算法选项，默认为 0。

参数

<0|1|2>布线算法选项。

示例

set_option -route_option 1

-ireg_in_iob

将输入 Buffer 连接的寄存器布局到 IOB 上。

语法

`-ireg_in_job <0|1>`

使用

名称	描述
<code><0 1></code>	0: 不将输入Buffer连接的寄存器布局到IOB上; 1: 将输入Buffer连接的寄存器布局到IOB上。

分类

[Project](#)

描述

启用该选项，Place & Route 会将输入 Buffer 连接的寄存器布局到 IOB 上，默认为 1。

参数

`<0|1>`将输入 Buffer 连接的寄存器布局到 IOB 上使能控制。

示例

```
set_option -ireg_in_job 1
```

-oreg_in_job

将输出/三态 Buffer 连接的寄存器布局到 IOB 上。

语法

`-oreg_in_job <0|1>`

使用

名称	描述
<code><0 1></code>	0: 不将输出/三态Buffer连接的寄存器布局到IOB上; 1: 将输出/三态Buffer连接的寄存器布局到IOB上。

分类

[Project](#)

描述

启用该选项，Place & Route 会将输出/三态 Buffer 连接的寄存器布局到 IOB 上，默认为 1。

参数

`<0|1>`将输出/三态 Buffer 连接的寄存器布局到 IOB 上使能控制。

示例

```
set_option -oreg_in_job 1
```

-ioreg_in_job

将双向 Buffer 连接的寄存器布局到 IOB 上。

语法

-ioreg_in_iob <0|1>

使用

名称	描述
<0 1>	0: 不将双向Buffer连接的寄存器布局到IOB上; 1: 将双向Buffer连接的寄存器布局到IOB上。

分类

[Project](#)

描述

启用该选项，Place & Route 会将双向 Buffer 连接的寄存器布局到 IOB 上，默认为 1。

参数

<0|1>将双向 Buffer 连接的寄存器布局到 IOB 上使能控制。

示例

```
set_option -ioreg_in_iob 1
```

-replicate_resources

对高扇出的资源进行复制来降低扇出，获得较好的时序结果。

语法

-replicate_resources <0|1>

使用

名称	描述
<0 1>	0: 不对高扇出的资源进行复制来降低扇出; 1: 对高扇出的资源进行复制来降低扇出。

分类

[Project](#)

描述

启用该选项，Place & Route 会对高扇出的资源进行复制来降低扇出，获得较好的时序结果，默认为 0。

参数

<0|1>对高扇出的资源进行复制来降低扇出使能控制。

示例

```
set_option -replicate_resources 1
```

-clock_route_order

指定除时钟原语产生的时钟线之外的时钟线的绕线分配顺序。

语法

-clock_route_order <0|1>

使用

名称	描述
<0 1>	0: 根据net的扇出数量由多到少的顺序进行分配; 1: 根据频率由高到底的顺序进行分配。

分类

[Project](#)

描述

指定除时钟原语产生的时钟线之外的时钟线的绕线分配顺序，可选项有 0 和 1，默认值为 0。

参数

<0|1>指定除时钟原语产生的时钟线之外的时钟线的绕线分配顺序。

示例

```
set_option -clock_route_order 1
```

-route_maxfan

设置绕线最大扇出数目。

语法

-route_maxfan <value>

使用

名称	描述
< value>	设置绕线最大扇出数目。

分类

[Project](#)

描述

基于绕线优化，使用该选项设置绕线最大扇出数目，取值应为大于 0 且小于等于 100 的整数。器件为 GW1NZ-1/GW1N-2/GW1NR-2/GW1N-1P5 时，选项默认值为 10，其他器件默认值为 23。

参数

<value>设置绕线最大扇出数目。

示例

```
set_option -route_maxfan 60
```

-correct_hold_violation

布线对时序 Hold 问题进行自动修复。

语法

-correct_hold_violation <0|1>

使用

名称	描述
<0 1>	0: 布线不对时序Hold问题进行自动修复; 1: 布线对时序Hold问题进行自动修复。

分类

[Project](#)

描述

启用该选项，布线会对时序 Hold 问题进行自动修复，默认为 1。

参数

<0|1>布线会对时序 Hold 问题进行自动修复使能控制。

示例

```
set_option -correct_hold_violation 1
```

-inc_place <0|auto|file>

增量布局。

语法

-inc_place <0|auto|file>

使用

名称	描述
<0 auto file >	0: 关闭增量布局; auto: 自动增量布局; file: 指定*.p文件进行增量布局。

分类

[Project](#)

描述

启用该选项，会使用增量布局，默认为 0。

参数

<0|auto|file >增量布局使能控制。

示例

```
set_option -inc_place auto
```

- -inc_pnr <0|auto|file>

增量布局布线。

语法

-inc_pnr <0|auto|file>

使用

名称	描述
<0 auto file >	0: 关闭增量布局布线; auto: 自动增量布局布线; file: 指定*.p文件进行增量布局布线。

分类

[Project](#)

描述

启用该选项，会使用增量布局布线，默认为 0。

参数

<0|auto|file >增量布局布线使能控制。

示例

set_option -inc_pnr auto

注！

Place & Route 相关运行选项的详细用法，请参考本文档 4.3.3 小节中的 [Place & Route](#) 部分。

复用管脚属性配置

-use_jtag_as_gpio

将 JTAG 相关管脚复用为普通 IO 管脚。

语法

-use_jtag_as_gpio <0|1>

使用

名称	描述
<0 1>	0: 作为 JTAG 专用管脚; 1: 复用为普通 IO 管脚。

分类

[Project](#)

描述

将 JTAG 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 JTAG 相关管脚复用为普通 IO 管脚使能控制。

示例

```
set_option -use_jtag_as_gpio 1
```

-use_sspi_as_gpio

将 SSPI 相关管脚复用为普通 IO 管脚。

语法

```
-use_sspi_as_gpio <0|1>
```

使用

名称	描述
<0 1>	0: 作为SSPI专用管脚; 1: 复用为普通IO管脚。

分类

[Project](#)

描述

将 SSPI 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 SSPI 相关管脚复用为普通 IO 管脚使能控制。

示例

```
set_option -use_sspi_as_gpio 1
```

-use_mspi_as_gpio

将 MSPI 相关管脚复用为普通 IO 管脚。

语法

```
-use_mspi_as_gpio <0|1>
```

使用

名称	描述
<0 1>	0: 作为MSPI专用管脚; 1: 复用为普通IO管脚。

分类

[Project](#)

描述

将 MSPI 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 MSPI 相关管脚复用为普通 IO 管脚使能控制。

示例

set_option -use_mspi _as_gpio 1

-use_ready_as_gpio

将 READY 相关管脚复用为普通 IO 管脚。

语法

-use_ready_as_gpio <0|1>

使用

名称	描述
<0 1>	0: 作为READY专用管脚; 1: 复用为普通IO管脚。

分类

[Project](#)

描述

将 READY 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 READY 相关管脚复用为普通 IO 管脚使能控制。

示例

set_option -use_ready_as_gpio 1

-use_done_as_gpio

将 DONE 相关管脚复用为普通 IO 管脚。

语法

-use_done_as_gpio <0|1>

使用

名称	描述
<0 1>	0: 作为 DONE 专用管脚; 1: 复用为普通 IO 管脚。

分类

[Project](#)

描述

将 DONE 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 DONE 相关管脚复用为普通 IO 管脚使能控制。

示例

set_option -use_done_as_gpio 1

-use_reconfign_as_gpio

将 RECONFIG_N 相关管脚复用为普通 IO 管脚。

语法

`-use_reconfig_as_gpio <0|1>`

使用

名称	描述
<0 1>	0: 作为RECONFIG_N专用管脚; 1: 复用为普通IO管脚。

分类

[Project](#)

描述

将 RECONFIG_N 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 RECONFIG_N 相关管脚复用为普通 IO 管脚使能控制。

示例

`set_option -use_reconfig_as_gpio 1`

`-use_i2c_as_gpio`

将 I2C 相关管脚复用为普通 IO 管脚。

语法

`-use_i2c_as_gpio <0|1>`

使用

名称	描述
<0 1>	0: 作为 I2C 专用管脚; 1: 复用为普通 IO 管脚。

分类

[Project](#)

描述

将 I2C 相关管脚复用为普通 IO 管脚，默认为 0。

参数

<0|1>将 I2C 相关管脚复用为普通 IO 管脚使能控制。

示例

`set_option -use_i2c_as_gpio 1`

BitStream 属性配置

`-bit_format`

指定生成的码流文件内容的格式。

语法

-bit_format <txt|bin>

使用

名称	描述
<txt bin>	码流文件内容的格式。

分类

[Project](#)

描述

用于指定生成的码流文件内容的格式。默认为 bin。

参数

<txt|bin>生成的码流文件内容的格式。

示例

set_option -bit_format txt

-bit_crc_check

循环冗余校验。

语法

-bit_crc_check <0|1>

使用

名称	描述
<0 1>	0: 不启用循环冗余校验; 1: 启用循环冗余校验。

分类

[Project](#)

描述

对码流文件启用循环冗余校验。默认为 1。

参数

<0|1>对码流文件启用循环冗余校验。

示例

set_option -bit_crc_check 1

-bit_compress

对码流文件进行压缩。

语法

-bit_compress <0|1>

使用

名称	描述
<0 1>	0: 不对码流文件进行压缩; 1: 对码流文件进行压缩。

分类

[Project](#)

描述

对生成的码流文件进行压缩。默认为 1。

参数

<0|1>对码流文件进行压缩。

示例

set_option -bit_compress 1

-bit_encrypt

对码流文件进行加密处理。

语法

-bit_encrypt <0|1>

使用

名称	描述
<0 1>	0: 不进行加密处理; 1: 进行加密处理。

分类

[Project](#)

描述

对码流文件进行加密处理，仅支持晨熙家族器件。默认为 0。

参数

<0|1>对码流文件进行加密。

示例

set_option -bit_encrypt 1

-bit_encrypt_key

定义加密的密钥。

语法

-bit_encrypt_key <key>

使用

名称	描述
<key>	加密的秘钥。

分类

Project

描述

该项与“-bit_encrypt”配合使用，用户可以对加密的密钥进行自定义。默认全为 0。

参数

<key>加密的密钥。

示例

```
set_option -bit_encrypt_key 000000000000000000000000000000001101
```

-bit_security

安全位使能控制。

语法

`-bit_security <0|1>`

使用

名称	描述
<0 1>	0: 关闭安全位使能控制; 1: 使用安全位使能控制。

分类

Project

描述

安全位使能控制。默认为 1。

参数

<0|1>安全位使能控制。

示例

```
set_option -bit_security 1
```

-bit incl bsram init

将 BSRAM 的初始值打印进码流文件中。

语法

```
-bit incl bsram init <0|1>
```

使用

名称	描述
<0 1>	0: 不将BSRAM的初始值打印进码流文件中; 1: 将BSRAM的初始值打印进码流文件中。

分类

[Project](#)

描述

将 BSRAM 的初始值打印进码流文件中。默认为 1。对于 GW1N 系列器件和 GW2A 系列器件，该项值为 1 时会将所有位置 BSRAM 的初始值都打印到码流文件中，没有被占用到的 BSRAM 位置的初始值会打印为 0。对于 GW5A(N)(S)(R)(T)器件，该项值为 1 时会将所占用位置 BSRAM 所在的所有 BSRAM 的初始值都打印到码流文件中，该列中没有被占用到的 BSRAM 位置的初始值会打印为 0。

参数

<0|1>BSRAM 的初始值打印进码流文件中使能控制。

示例

```
set_option -bit_incl_bsram_init 1
```

-bg_programming

背景升级功能。

语法

```
-bg_programming <off | jtag | i2c | goconfig | userlogic |  
i2c_jtag_ssapi_qsspi | jtag_ssapi_qsspi>
```

使用

名称	描述
<off jtag i2c goconfig userlogic i2c_jtag_ssapi_qsspi jtag_ssapi_qsspi>	off: 不使用背景升级功能; jtag: 使用JTAG模式进行背景升级; i2c: 使用I2C模式进行背景升级; goconfig: 使用goConfig IP实现背景升级; userlogic: 使用FPGA内部逻辑进行背景升级; i2c_jtag_ssapi_qsspi: 使用 I2C/JTAG/SSPI/QSSPI模式进行背景升级; jtag_ssapi_qsspi: 使用 JTAG/SSPI/QSSPI模式进行背景升级。

分类

[Project](#)

描述

背景升级功能，在不中断 FPGA 芯片现有功能执行的前提下对 Flash 进行烧录。默认为 off。

参数

< off | jtag | i2c | goconfig | userlogic | i2c_jtag_ssapi | jtag_ssapi>背景升级方式。

示例

set_option -bg_programming userlogic

-hotboot

热启动模式。

语法

-hotboot <0|1>

使用

名称	描述
<0 1>	0: 不使用热启动模式; 1: 使用热启动模式。

分类

[Project](#)

描述

热启动模式使能控制。默认为 0。

参数

<0|1>热启动模式使能控制。

示例

set_option -hotboot 1

-i2c_slave_addr

设置 I2C 设备的地址。

语法

-i2c_slave_addr <value>

使用

名称	描述
<value>	I2C 设备的地址。

分类

[Project](#)

描述

设置 I2C 设备的地址，可配置值范围为 00~7F。默认为 00。

参数

<value>I2C 设备的地址。

示例

set_option -i2c_slave_addr 2F

-secure_mode

启用安全模式。

语法

-secure_mode <0|1>

使用

名称	描述
<0 1>	0: 不启用安全模式; 1: 启用安全模式。

分类

[Project](#)

描述

启用安全模式，此时 JTAG 管脚为 GPIO，码流文件只能对设备编程一次。默认为 0。

参数

<0|1>启用安全模式使能控制。

示例

set_option -secure_mode 1

-loading_rate

AutoBoot 配置模式和 MSPI 配置模式下，码流数据从 Flash 到 SRAM 的加载速度。

语法

-loading_rate <value>

使用

名称	描述
<value>	AutoBoot配置模式和MSPI配置模式下，码流数据从Flash到SRAM的加载速度。

分类

[Project](#)

描述

AutoBoot 配置模式和 MSPI 配置模式下，码流数据从 Flash 到 SRAM

的加载速度，默认为 2.500MHz。

参数

<value>码流数据从 Flash 到 SRAM 的加载速度。

示例

set_option -loading_rate 21.000MHz

-seu_handler

启用单粒子翻转处理器。

语法

-seu_handler <0|1>

使用

名称	描述
<0 1>	0: 不启用单粒子翻转处理器; 1: 启用单粒子翻转处理器。

分类

[Project](#)

描述

启用单粒子翻转处理器。默认为 0。

参数

<0|1>启用单粒子翻转处理器使能控制。

示例

set_option -seu_handler 1

-seu_handler_mode

选择 SEU Handler 启动或停止的模式。

语法

-seu_handler_mode <auto|userlogic>

使用

名称	描述
<0 1>	auto: 芯片唤醒后自动启用单粒子翻转处理器; userlogic: 使用逻辑启用或停止单粒子翻转处理器。

分类

[Project](#)

描述

选择 SEU Handler 启动或停止的模式，默认为 auto。

参数

< auto|userlogic>SEU Handler 启动或停止的模式。

示例

set_option -seu_handler_mode userlogic

-seu_handler_checksum

启用单粒子翻转处理、检验、计算和对比。

语法

-seu_handler_checksum <0|1>

使用

名称	描述
<0 1>	0: 不启用单粒子翻转处理、检验、计算和对比； 1: 启用单粒子翻转处理、检验、计算和对比。

分类

[Project](#)

描述

启用单粒子翻转处理、检验、计算和对比。默认为 0。

参数

<0|1>启用单粒子翻转处理、检验、计算和对比使能控制。

示例

set_option -seu_handler_checksum 1

-error_detection

仅启用错误检验。

语法

-error_detection <0|1>

使用

名称	描述
<0 1>	0: 不启用错误检验； 1: 仅启用错误检验。

分类

[Project](#)

描述

仅启用错误检验。默认为 0。

参数

<0|1>仅启用错误检验使能控制。

示例

```
set_option -error_detection 1
```

-error_detection_correction

启用错误检验和更正。

语法

```
-error_detection_correction <0|1>
```

使用

名称	描述
<0 1>	0: 不启用错误检验和更正; 1: 启用错误检验和更正。

分类

[Project](#)

描述

启用错误检验和更正。默认为 0。

参数

<0|1>启用错误检验和更正使能控制。

示例

```
set_option -error_detection_correction 1
```

-stop_seu_handler

停止 SEU Handler。

语法

```
-stop_seu_handler <0|1>
```

使用

名称	描述
<0 1>	0: 当检测到不可纠正的ECC错误或CRC校验和不匹配错误时，不停止SEU Handler; 1: 当检测到不可纠正的ECC错误或CRC校验和不匹配错误时，停止SEU Handler。

分类

[Project](#)

描述

当检测到不可纠正的 ECC 错误或 CRC 校验和不匹配错误时，停止 SEU Handler。默认为 0。

参数

<0|1>停止 SEU Handler 使能控制。

示例

set_option -stop_seu_handler 1

-osc_div

设置扩展控制寄存器的分频比。

语法

-osc_div <4|8|16|32>

使用

名称	描述
<4 8 16 32>	4: 扩展控制寄存器的分频比设置为4; 8: 扩展控制寄存器的分频比设置为8; 16: 扩展控制寄存器的分频比设置为16; 32: 扩展控制寄存器的分频比设置为32。

分类

[Project](#)

描述

设置扩展控制寄存器的分频比。默认为 8。

参数

<4|8|16|32>扩展控制寄存器的分频比。

示例

set_option -osc_div 8

-error_injection

启用错误注入。

语法

-error_injection <0|1>

使用

名称	描述
<0 1>	0: 不启用错误注入; 1: 启用错误注入。

分类

[Project](#)

描述

启用错误注入。默认为 0。

参数

<0|1>启用错误注入使能控制。

示例

```
set_option -error_injection 1
```

-ext_cclk

启用外部主配置时钟。

语法

```
-ext_cclk <0|1>
```

使用

名称	描述
<0 1>	0: 不启用外部主配置时钟; 1: 启用外部主配置时钟。

分类

[Project](#)

描述

启用外部主配置时钟。默认为 0。

参数

<0|1>启用外部主配置时钟使能控制。

示例

```
set_option -ext_cclk 1
```

-ext_cclk_div

设置分频器参数。

语法

```
-ext_cclk_div <value>
```

使用

名称	描述
<value>	分频器参数。

分类

[Project](#)

描述

设置分频器参数。

参数

< value>设置分频器参数。

示例

```
set_option -ext_cclk_div 4
```

-multi_boot

Multi Boot 使能控制。

语法

```
-multi_boot <0|1>
```

使用

名称	描述
<0 1>	0: 不启用Multi Boot; 1: 启用Multi Boot。

分类

[Project](#)

描述

Multi Boot 使能控制，默认为 0。

参数

<0|1> Multi Boot 使能控制。

示例

```
set_option -multi_boot 1
```

-multiboot_address_width

配置 SPI Flash 地址位宽。

语法

```
-multiboot_address_width<24|32>
```

使用

名称	描述
<24 32>	24: SPI Flash地址位宽设为24; 32: SPI Flash地址位宽设为32。

分类

[Project](#)

描述

配置 SPI Flash 地址位宽，默认为 24。

参数

<24|32> SPI Flash 地址位宽。

示例

```
set_option -multiboot_address_width 32
```

-multiboot_spi_flash_address

指定 SPI Flash 地址。

语法

```
-multiboot_spi_flash_address <value>
```

使用

名称	描述
<value>	SPI Flash 地址。

分类

[Project](#)

描述

指定 SPI Flash 地址。SPI Flash 地址是指下一次 multiboot 时，加载码流文件的起始地址，默认为 000000。

参数

< value>SPI Flash 地址。

示例

```
set_option -multiboot_spi_flash_address 000110
```

-multiboot_mode

配置 SPI Flash 地址访问模式。

语法

```
-multiboot_mode <single | fast | dual | quad>
```

使用

名称	描述
< single fast dual quad >	single: 使用single模式; fast: 使用fast模式; dual: 使用dual模式; quad: 使用quad模式。

分类

[Project](#)

描述

配置 SPI Flash 地址访问模式，默认为 single。

参数

< single | fast | dual | quad > SPI Flash 地址访问模式。

示例

```
set_option -multiboot_mode single
```

-mspi_jump

MSPI JUMP 使能控制。

语法

```
-mspi_jump<0|1>
```

使用

名称	描述
<0 1>	0: 不启用MSPI JUMP; 1: 启用MSPI JUMP。

分类

[Project](#)

描述

MSPI JUMP 使能控制，默认为 0。

参数

<0|1> MSPI JUMP 使能控制。

示例

```
set_option -mspi_jump 1
```

-merge_jumpbit

将 MSPI JUMP 码流文件合并到通用的码流文件中。

语法

```
-merge_jumpbit <0|1>
```

使用

名称	描述
<0 1>	0: 不合并码流文件; 1: 合并码流文件。

分类

[Project](#)

描述

将 MSPI JUMP 码流文件合并到通用的码流文件中，默认为 0。

参数

<0|1> MSPI JUMP 码流文件合并到通用的码流文件中使能控制。

示例

```
set_option -merge_jumpbit 1
```

-mspijump_address_width

配置 SPI Flash 地址位宽。

语法

-mspijump_address_width <24|32>

使用

名称	描述
<24 32>	24: SPI Flash地址位宽设为24; 32: SPI Flash地址位宽设为32。

分类

[Project](#)

描述

配置 SPI Flash 地址位宽，默认为 24。

参数

<24|32> SPI Flash 地址位宽。

示例

```
set_option - mspijump_address_width 32
```

-mspijump_spi_flash_address

指定 SPI Flash 地址。

语法

-mspijump_spi_flash_address <value>

使用

名称	描述
<value>	SPI Flash地址。

分类

[Project](#)

描述

指定 SPI Flash 地址，默认为 000000。

参数

<value>SPI Flash 地址。

示例

```
set_option - mspijump_spi_flash_address 000110
```

-mspijump_mode<single | fast | dual | quad>

配置 SPI Flash 地址访问模式。

语法

-mspijump_mode <single | fast | dual | quad>

使用

名称	描述
< single fast dual quad >	single: 使用single模式; fast: 使用fast模式; dual: 使用dual模式; quad: 使用quad模式。

分类

[Project](#)

描述

配置 SPI Flash 地址访问模式，默认为 single。

参数

< single | fast | dual | quad > SPI Flash 地址访问模式。

示例

```
set_option -mspijump_mode single
```

-program_done_bypass

转发新的码流数据。

语法

-program_done_bypass <0|1>

使用

名称	描述
<0 1>	0: 不启用该功能; 1: 启用该功能。

分类

[Project](#)

描述

配置该选项后，在 Done Final 内部信号生效时，同时外部 Done Pin 保持低电平，使码流加载完成后可以转发新的码流数据，默认为 0。

参数

<0|1> 转发新的码流数据使能控制。

示例

```
set_option -program_done_bypass 1
```

-power_on_reset_monitor <0|1>

上电复位。

语法

-power_on_reset_monitor <0|1>

使用

名称	描述
<0 1>	0: 不启用上电复位功能; 1: 启用上电复位功能。

分类

[Project](#)

描述

上电复位功能的使能控制。默认为 0。

参数

<0|1> 上电复位功能的使能控制。

示例

set_option -power_on_reset_monitor 1

-turn_off_bg

Bandgap 功能。

语法

-turn_off_bg <0|1>

使用

名称	描述
<0 1>	0: 启用Bandgap功能; 1: 不启用Bandgap功能。

分类

[Project](#)

描述

Bandgap 功能的使能控制。默认为 0。

参数

<0|1> Bandgap 功能的使能控制。

示例

set_option -turn_off_bg 1

-wakeup_mode

Wake Up Mode 使能控制。

语法

-wakeup_mode <0|1>

使用

名称	描述
<0 1>	0: 关闭Wake Up Mode的使能控制; 1: 打开Wake Up Mode的使能控制。

分类

[Project](#)

描述

Wake Up Mode 的使能控制。默认为 0。

参数

<0|1> Wake Up Mode 的使能控制。

示例

```
set_option -wakeup_mode 1
```

-user_code

自定义 User Code。

语法

-user_code <default|value>

使用

名称	描述
<default value>	定义User Code值。

分类

[Project](#)

描述

用户可以自定义 User Code。默认为 default(00000000)。

参数

<default|value>自定义 User Code。

示例

```
set_option -user_code 00000010
```

注！
BitStream 相关选项的详细解释，请参考本文档 4.3.3 小节中的 BitStream 部分。

Unused Pin 属性配置

-unused_pin

对未使用的 GPIO 设置不同的 IO 属性。

语法

-unused_pin <default|open_drain>

使用

名称	描述
<default open_drain>	default: 所有未使用的GPIO会配置为输入三态弱上拉; open_drain: 所有未使用的GPIO会配置为输出类型, 并且OPEN DRAIN配置为ON。

分类

[Project](#)

描述

对未使用的 GPIO 设置不同的 IO 属性。

参数

<default|open_drain>对未使用的 GPIO 设置不同的 IO 属性。

示例

set_option -unused_pin open_drain

注!
Unused Pin 相关选项的详细解释, 请参考本文档 4.3.3 小节中的 [Unused Pin](#) 部分。

8.3.19 set_property

设置对象的属性。

语法

set_property [-dict <args>] <name> <value> <objects>

使用

名称	描述
[-dict]	要设置的 (name/value) 对的属性列表
<name>	要设置的属性的名称, 使用-dict时无效
<value>	要设置的属性的值, 使用-dict时无效
<objects>	要设置的属性的对象

分类

[IPFlow](#)

描述

配置指定 IP 对象的选项及其对应的选项值。

参数

- [-dict]指定一个包含多对选项及对应的选项值的字典, 通过 (<name>

<value>) 指定一对，多对之间通过空格隔开，字典需要通过大括号{}括起来。

- <name>指定要配置的选项的名称，描述形式为 CONFIG.property，property 指代的是选项名称；
- <value>指定要配置的选项对应的选项值，需要根据属性类型判断该值，以确保该值是有效的，如果选项值是字符串形式，统一规范为原形式；
- <objects>指定要配置选项的一个或多个 IP 对象，一个对象通过[get_ips module_name]指定，如果是指定多个对象，则使用[get_ips module_name0 module_name1 ...]。

示例

使用-dict 选项在当前设计上一次指定多个属性：

```
set_property -dict {CONFIG.Data_Width 16 CONFIG.Write_Depth 1024 CONFIG.Read_Depth 1024} [get_ips FIFO_Top]
```

使用 (name, value, objects) 在当前设计上指定一个属性：

```
set_property CONFIG.Data_Width {16} [get_ips FIFO_Top]
```

该示例展示了如何设置一个包含波折号“-”或空格的属性值：

```
set_property {CONFIG.Almost_Full_Type} {Full-Single Threshold Constant Parameter} [get_ips FIFO_Top]
```

注！
某些情况下选项值中可能包含特殊字符，比如破折号“-”或空格，这有可能导致该选项值被错误解析，这种情况下必须使用大括号{}将选项值括起来。

另请参阅

- [list_property](#)
- [report_property](#)

8.3.20 source

在云源的 tcl 命令编辑窗口中或启动命令行模式后，使用该命令可以执行 tcl 脚本。文件路径形式请参考 [8.3.1 add_file](#)。

语法

```
source <file>
```

使用

名称	描述
<file>	要执行的tcl脚本。

分类

[IPFlow](#), [Project](#)

描述

对未使用的 GPIO 设置不同的 IO 属性。

参数

<file> 要被执行的 Tcl 脚本文件。

示例

```
source project.tcl
source D:/gowin_project/project.tcl
source D:\\gowin_project\\project.tcl
```

8.3.21 write_ip_tcl

导出 Tcl 脚本，该脚本可以重新生成给定的 IP。

语法

```
write_ip_tcl [-ip_name <newModuleName>] [-multiple_files] [-force]
[<tcl_filename>] <objects>
```

使用

名称	描述
[-ip_name]	设置 IP module 名称
[-multiple_files]	为每个 IP 对象创建一个.tcl 文件
[-force]	覆盖已有文件
[<tcl_filename>]	导出的 tcl 文件
<objects>	导出 tcl 文件的 IP 对象

分类

[IPFlow](#)

描述

该命令为指定的 IP 对象导出 Tcl 脚本文件，便于通过 Tcl 脚本再次生成 IP，文件路径形式请参考 [8.3.1 add file](#)。

参数

- [<tcl_filename>] 生成的 Tcl 脚本文件名称，不指定时使用当前工程名称命名；
- <objects>指定要生成 Tcl 脚本的一个或多个 IP 对象，一个对象通过 [get_ips module_name]指定，如果是指定多个对象，则使用[get_ips module_name0 module_name1 ...]。
- [-ip_name <newModuleName>]重命名生成的 Tcl 脚本中的 IP 的 module_name，只能指定一个对象；
- [-multiple_files]为指定的所有 IP 生成 Tcl 脚本文件，脚本名称以对应 IP 的 module_name 命名，该选项与-ip_name 以及<tcl_filename>互斥；

- [-force]覆盖已有的同名的 Tcl 文件。

示例

该示例编写了一个 Tcl 文件，该文件指定 IP 对象 FIFO_Top，但在运行 source 命令时将使用新名称 FIFO_Top_new 创建它：

```
write_ip_tcl -ip_name FIFO_Top_new [get_ips FIFO_Top]
```

该示例为工程中指定的每个 IP 模块编写独立的 Tcl 文件：

```
write_ip_tcl -multiple_files [get_ips FIFO_Top FIFO_Top_1]
```

该示例将工程中的指定的多个 IP 模块写入一个 Tcl 文件：

```
write_ip_tcl [get_ips FIFO_Top FIFO_Top_1] my_fifo.tcl
```

9 附录

9.1 文件说明

云源支持在工程设计中添加物理约束、时序约束等文件，在设计运行综合及布局布线过程中会产生多种执行文件，本章节以表的形式介绍云源支持的各种文件类型。

表 9-1 源文件

类型	定义	描述
.gsc	设计综合约束文件	用于 GowinSynthesis 综合工具的约束文件
.ipc	IP 配置文件	IP Core Generator 可加载.ipc 文件对 IP 进行修改配置后重新产生
.cst	物理约束文件	用于对设计添加物理约束
.sdc	时序约束文件	用于对设计添加时序约束
.fi	User Flash 初始化文件	对 User Flash 进行初始化赋值，通过编程器下载码流时可以选择加载该初始化文件。
.rao	RTL 级 GAO 配置文件	用于捕获综合优化前的 RTL 信号
.gao	综合后 GAO 配置文件	用于捕获综合优化后的 Netlist 信号
.gvio	虚拟输入输出配置文件	用于实时监控并驱动内部 FPGA 信号
.gpa	功耗分析配置文件	用于对设计进行功耗分析
.mi	存储器初始化文件	对存储器进行初始化赋值，通过 IP Core Generator 产生存储器时使用该初始化文件。
.v	Verilog 源文件	包含电路结构和功能的 Verilog 描述文件
.sv	System Verilog 源文件	包含电路结构和功能的 System Verilog 描述文件
.vhd	VHDL 源文件	包含电路结构和功能的 VHDL 描述文件

表 9-2 执行文件

类型	定义	描述
.vg	综合后的网表文件	使用 GowinSynthesis 综合后产生的网表文件
_syn.rpt.html	综合报告文件	可以查看综合后的资源占用和时序分析等信息
.fs	BitStream 文件	用于云源编程器进行下载
.bin	bin 格式的 BitStream 文件	用于云源编程器进行下载
.ekey	密钥文件	用于云源编程器进行下载时对加密的 BitStream 文件进行解密
.vo	布局布线后的 verilog 格式的时序仿真模型文件	用于时序仿真的 flatten 结构的 Verilog 模型文件
.vho	布局布线后的 vhdl 格式的时序仿真模型文件	用于时序仿真的 flatten 结构的 VHDL 模型文件
.sdf	标准延迟格式文件	用于布局布线后的网表时序仿真
.ibs	输入/输出缓冲区信息指定文件	-
.tr	文本格式的时序报告	-
.rpt.txt	文本格式的布局布线报告	-
.rpt.html	html 格式的布局布线报告	-
.tr.html	html 格式的时序分析报告	-
.pin.html	html 格式的端口属性报告	-
.power.html	html 格式的功耗分析报告	-
.p	增量布局文件	用于增量布局
.pr	增量布局布线文件	用于增量布局布线

9.2 文件和文件夹命名规则

云源文件夹和文件命名规则：名字中不可以包含“\ / < > * | :”字符，文件夹名字不允许包含空格，文件名字可以包括空格但空格不能出现在名字的首尾。

在云源各个组件的对话框中填写文件路径名时，会按照上述规则进行判断，不符合规则的会弹出提示窗口。

9.3 安全声明

在用户安装与使用的过程中，云源不会在后台收集用户的任何信息或者访问网络数据端口，所有数据信息都保留在本地，也不会对软件进行自动更新。

