



Gowin IP 核产生工具 用户指南

SUG284-1.8, 2019-05-25

版权所有©2019 广东高云半导体科技股份有限公司

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

免责声明

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

版本信息

日期	版本	说明
2017/03/07	1.0	初始版本。
2018/01/30	1.2	<ul style="list-style-type: none">● 支持 GW1NR-4、GW1N-6、GW1N-9、GW1NR-9;● 更新了 BSRAM 部分;● 更新了 DSP 部分;● 更新了 PLL;● 更新了 User Flash。
2018/08/25	1.3	<ul style="list-style-type: none">● 支持 GW1N-2B、GW1N-4B、GW1N-6ES、GW1N-9ES、GW1NR-4B、GW1NR-9ES、GW1NS-2、GW1NS-2C;● 新增 IP DDR3、DDR3 PHY;● 更新 PLL;● 更新 OSC;● 更新 User Flash;● 优化界面显示;● 新增 IP CORDIC、Complex Multiplier、DIVIDER
2018/10/26	1.4	<ul style="list-style-type: none">● 支持 GW1NZ-1、GW1NSR-2C;● 新增 I3C、SPMI 硬核。
2018/11/15	1.5	<ul style="list-style-type: none">● 支持 GW1NSR-2;● 删除 GW1N-6ES、GW1NS-9ES、GW1NR-9ES。
2019/02/12	1.6	<ul style="list-style-type: none">● 新增 IP RiscV N25、CAN;● 更新 PSRAM、DDR_x、MIPI。
2019/02/25	1.7	<ul style="list-style-type: none">● 新增 IP Basic FIR Filter、FD Adaptive Filter、Integer Multiply Divider、NLMS Adaptive Filter、XCORR、Triple Speed Ethernet MAC;● 优化界面显示 (删除界面 Add to Current Project 选项)。
2019/05/17	1.8	<ul style="list-style-type: none">● 支持 GW1N-1S● 新增 IP PSRAM Memory Interface 2CH、Advanced FIR Filter、Gowin_EMPU_M1、HyperRAM Memory Interface● 新增 Hard module 部分 Shadow Memory, 包括 RAM16S、RAM16SDP、ROM16● 更新 IP MIPI、DDR、DDR2、DDR3、GOWIN_EMPU

目录

目录.....	i
图目录.....	iii
表目录.....	viii
1 关于本手册.....	1
1.1 手册内容.....	1
1.2 适用产品.....	1
1.3 相关文档.....	1
1.4 术语、缩略语.....	2
1.5 技术支持与反馈.....	2
2 简介.....	3
3 使用.....	4
3.1 Block Memory.....	9
3.1.1 SP.....	9
3.1.2 DP.....	14
3.1.3 SDP.....	20
3.1.4 ROM.....	26
3.2 DSP.....	31
3.2.1 ALU54.....	31
3.2.2 MULT.....	37
3.2.3 MULTADDALU.....	42
3.2.4 MULTALU.....	48
3.2.5 PADD.....	54
3.3 CLOCK.....	60
3.3.1 PLL.....	60
3.3.2 DLL.....	70
3.3.3 OSC.....	75
3.4 User Flash.....	79
3.5 I3C.....	84
3.6 SPMI.....	92

3.7 Shadow Memory.....	99
3.7.1 RAM16S	99
3.7.2 RAM16SDP	103
3.7.3 ROM16	108

图目录

图 3-1 IP Core Generator 界面	4
图 3-2 Select Device 窗口	6
图 3-3 IP Customization 窗口	7
图 3-4 ipc 文件的 IP Customization 窗口	8
图 3-5 SP 的信息概要	9
图 3-6 SP 的 IP Customization 窗口结构	10
图 3-7 Language 下拉列表框	10
图 3-8 Module 名与原语名称一致时报 Error	11
图 3-9 Help 信息	12
图 3-10 配置的 IP Customization	13
图 3-11 例化 SP 设计文件	13
图 3-12 用户例化该 IP 设计文件的模板文件	14
图 3-13 SP IP Customization 设置	14
图 3-14 DP 的信息概要	15
图 3-15 DP 的 IP Customization 窗口结构	16
图 3-16 DP 配置 Error 提示	17
图 3-17 Help 信息	17
图 3-18 配置的 IP Customization	18
图 3-19 例化 DP 设计文件	19
图 3-20 用户例化该 IP 设计文件的模板文件	19
图 3-21 DP IP Customization 设置	20
图 3-22 SDP 的信息概要	21
图 3-23 SDP 的 IP Customization 窗口结构	21
图 3-24 SDP 配置 Error 信息	22
图 3-25 Help 信息	23
图 3-26 配置的 IP Customization	24
图 3-27 例化 SDP 设计文件	25
图 3-28 用户例化该 IP 设计文件的模板文件	25
图 3-29 SDP 的 IP Customization 设置	26

图 3-30 ROM 的信息概要	27
图 3-31 ROM 的 IP Customization 窗口结构	27
图 3-32 Help 信息	28
图 3-33 配置的 IP Customization	29
图 3-34 例化 ROM 设计文件	30
图 3-35 用户例化 IP 设计文件的模板文件	30
图 3-36 ROM 的 IP Customization 设置	31
图 3-37 ALU54 的信息概要	32
图 3-38 ALU54 的 IP Customization 窗口结构	32
图 3-39 DSP 部分界面变灰	33
图 3-40 Help 信息	34
图 3-41 配置的 IP Customization	35
图 3-42 例化 ALU54 设计文件	36
图 3-43 用户例化该 IP 设计文件的模板文件	36
图 3-44 ALU54 IP Customization 设置	37
图 3-45 MULT 的信息概要	38
图 3-46 MULT 的 IP Customization 窗口结构	38
图 3-47 Help 信息	40
图 3-48 配置的 IP Customization	41
图 3-49 例化 MULT 设计文件	41
图 3-50 用户例化该 IP 设计文件的模板文件	42
图 3-51 MULT IP Customization 设置	42
图 3-52 MULTADDALU 的信息概要	43
图 3-53 MULTADDALU 的 IP Customization 窗口结构	43
图 3-54 Help 信息	45
图 3-55 配置的 IP Customization	46
图 3-56 例化 MULTADDALU 设计文件	47
图 3-57 用户例化该 IP 设计文件的模板文件	47
图 3-58 MULTADDALU IP Customization 设置	48
图 3-59 MULTALU 的信息概要	49
图 3-60 MULTALU 的 IP Customization 窗口结构	50
图 3-61 Help 信息	51
图 3-62 配置的 IP Customization	52
图 3-63 例化 MULTALU 设计文件	53
图 3-64 用户例化该 IP 设计文件的模板文件	53
图 3-65 MULTALU IP Customization 设置	54
图 3-66 PADD 信息概要	55

图 3-67 PADD 的 IP Customization 窗口结构.....	55
图 3-68 Help 信息.....	57
图 3-69 配置的 IP Customization.....	58
图 3-70 例化 PADD 设计文件.....	59
图 3-71 用户例化该 IP 设计文件的模板文件.....	59
图 3-72 PADD IP Customization 设置.....	60
图 3-73 PLL 的信息概要.....	61
图 3-74 PLL 的 IP Customization 窗口结构.....	62
图 3-75 CLKIN/CLKFB Divide Factor 配置不合理的 error 窗口.....	64
图 3-76 CLKIN Divide Factor 配置不合理的 error 窗口.....	64
图 3-77 CLKOUTD 分频参数配置不合理的 Error 窗口.....	65
图 3-78 CLKOUT 期望频率与计算频率不等的 Error 窗口.....	65
图 3-79 CLKOUTD 期望频率与计算频率不等的 Error 窗口.....	65
图 3-80 VCO 参数配置不合理的 error 窗口.....	65
图 3-81 参数配置合理的 info 窗口.....	65
图 3-82 Help 信息.....	66
图 3-83 配置的 IP Customization.....	68
图 3-84 例化 PLL 设计文件.....	69
图 3-85 用户例化该 IP 设计文件的模板文件.....	70
图 3-86 PLL IP Customization 设置.....	70
图 3-87 DLL 的信息概要.....	71
图 3-88 DLL 的 IP Customization 窗口结构.....	71
图 3-89 Help 信息.....	72
图 3-90 配置的 IP Customization.....	73
图 3-91 例化 DLL 设计文件.....	74
图 3-92 用户例化该 IP 设计文件的模板文件.....	74
图 3-93 DLL IP Customization 设置.....	75
图 3-94 OSC 的信息概要.....	75
图 3-95 OSC 的 IP Customization 窗口结构.....	76
图 3-96 Help 信息.....	77
图 3-97 配置的 IP Customization.....	77
图 3-98 例化 OSC 设计文件.....	78
图 3-99 用户例化该 IP 设计文件的模板文件.....	78
图 3-100 OSC IP Customization 设置.....	78
图 3-101 User Flash 的信息概要.....	79
图 3-102 User Flash 的 IP Customization 窗口结构.....	80
图 3-103 Help 信息.....	81

图 3-104 配置的 IP Customization.....	82
图 3-105 例化 User Flash 设计文件.....	83
图 3-106 用户例化该 IP 设计文件的模板文件.....	83
图 3-107 User Flash IP Customization 设置.....	84
图 3-108 I3C SDR 的信息概要	84
图 3-109 I3C 的 IP Customization 窗口结构.....	85
图 3-110 Help 信息	86
图 3-111 配置的 IP Customization	87
图 3-112 例化 I3C 设计文件.....	88
图 3-113 用户例化该 IP 设计文件的模板文件.....	91
图 3-114 I3C IP Customization 设置.....	92
图 3-115 SPMI 的信息概要	93
图 3-116 SPMI 的 IP Customization 窗口结构.....	93
图 3-117 Help 信息	95
图 3-118 配置的 IP Customization	96
图 3-119 例化 SPMI 设计文件	97
图 3-120 用户例化该 IP 设计文件的模板文件.....	98
图 3-121 SPMI IP Customization 设置	99
图 3-122 RAM16S 的信息概要	100
图 3-123 RAM16S 的 IP Customization 窗口结构.....	100
图 3-124 Help 信息	101
图 3-125 配置的 IP Customization.....	102
图 3-126 例化 RAM16S 设计文件.....	102
图 3-127 用户例化该 IP 设计文件的模板文件.....	103
图 3-128 RAM16S IP Customization 设置.....	103
图 3-129 RAM16SDP 的信息概要	104
图 3-130 RAM16SDP 的 IP Customization 窗口结构.....	104
图 3-131 Help 信息	105
图 3-132 配置的 IP Customization.....	106
图 3-133 例化 RAM16SDP 设计文件.....	107
图 3-134 用户例化该 IP 设计文件的模板文件.....	107
图 3-135 SDP 的 IP Customization 设置	108
图 3-136 ROM16 的信息概要	108
图 3-137 ROM16 的 IP Customization 窗口结构.....	109
图 3-138 Help 信息	110
图 3-139 配置的 IP Customization.....	111
图 3-140 例化 ROM16 设计文件.....	112

图 3-141 用户例化 IP 设计文件的模板文件	112
图 3-142 ROM16 的 IP Customization 设置	113

表目录

表 1-1 术语、缩略语 2

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（Gowin 云源软件）的 IP 核产生工具（IP Core Generator）的使用，旨在帮助用户快捷实现复杂设计。Gowin 云源软件支持 Linux 和 Windows 系统。有关本手册中的软件界面截图和支持的产品列表等信息，参考的是 Windows 系统的 1.9.1Beta 版本。因软件版本更新，部分信息可能会略有差异，具体以用户软件版本信息为准。

1.2 适用产品

本手册中描述的信息适用于以下产品：

1. GW1N 系列 FPGA 产品：GW1N-1, GW1N-2, GW1N-2B, GW1N-4, GW1N-4B, GW1N-6, GW1N-9, GW1N-1S
2. GW1NR 系列 FPGA 产品：GW1NR-4, GW1NR-4B, GW1NR-9
3. GW1NS 系列 FPGA 产品：GW1NS-2, GW1NS-2C
4. GW2A 系列 FPGA 产品：GW2A-55, GW2A-18
5. GW2AR 系列 FPGA 产品：GW2AR-18
6. GW1NZ 系列 FPGA 产品：GW1NZ-1
7. GW1NSR 系列 FPGA 产品：GW1NSR-2, GW1NSR-2C

1.3 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：

1. GW1N 系列 FPGA 产品数据手册
2. GW1NR 系列 FPGA 产品数据手册
3. GW1NS 系列 FPGA 产品数据手册
4. GW2A 系列 FPGA 产品数据手册
5. GW2AR 系列 FPGA 产品数据手册
6. GW1NZ 系列 FPGA 产品数据手册
7. GW1NSR 系列 FPGA 产品数据手册

1.4 术语、缩略语

本手册中出现的相关术语、缩略语及相关释义请参考表 1-1。

表 1-1 术语、缩略语

术语、缩略语	全称	含义
FPGA	Field Programmable Gate Array	现场可编程门阵列
IDE	Integrated Development Environment	集成开发环境
IP Core	Intellectual Property Core	知识产权核
DP/DPX9	Dual Port	双端口存储器
SP/SPX9	Single Port	单端口存储器
SDP/SDPX9	Semi-dual Port	半双端口存储器
ROM/ROMX9	Read Only Memory	只读存储器
PADD	Pre-adder	前加器
MULT	Multiplier	乘法器
PLL	Phase-locked Loop	锁相环
DLL	Delay Lock Loop	延迟锁相环
OSC	Oscillator	片内晶振
SPMI	System Power Management Interface	系统电源管理接口

1.5 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 简介

Gowin 云源软件 IP 核产生工具主要用于产生实例化的元件以及 IP，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，帮助用户快捷实现复杂设计。目前 IP Core Generator 包括原语相关的 Hard Module 部分和参考设计相关的 Soft IP Core 两个组成部分。

3 使用

通过选择菜单栏“Tools > IP Core Generator”，或工具栏，打开 IP Core Generator 界面，如图 3-1 所示。

该界面主要分为以下两个部分：

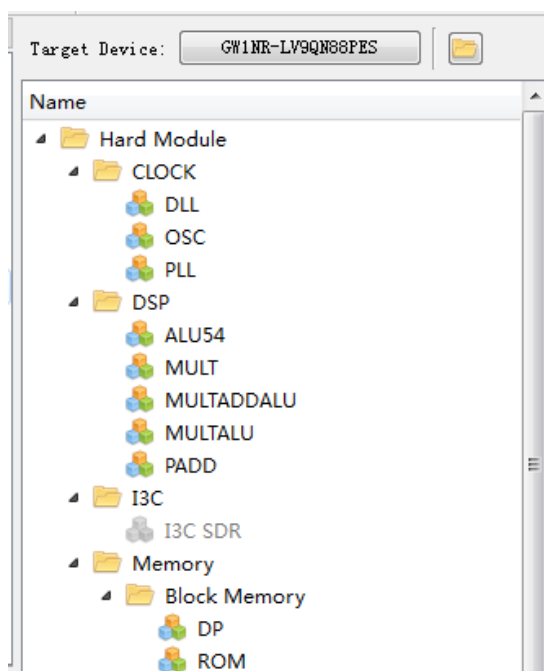
- 原语相关的 **Hard Module** 部分；
- 参考设计相关的 **Soft IP Core** 部分。

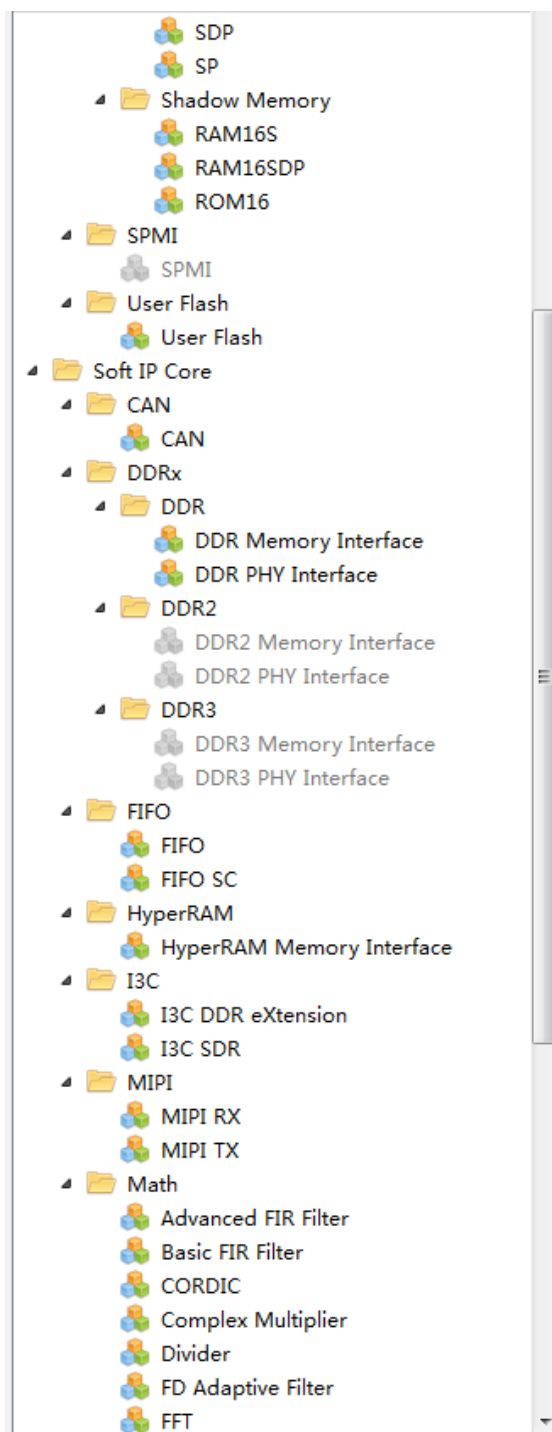
Hard Module 部分包括 CLOCK、DSP、I3C、Memory、SPMI 和 User Flash 等；

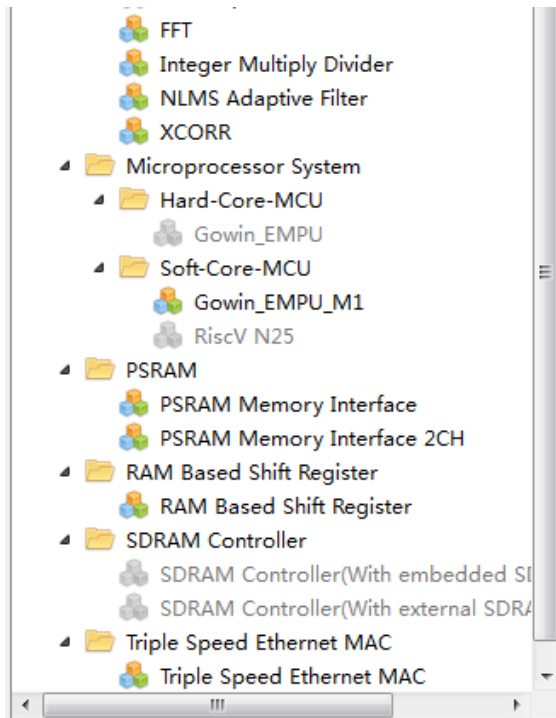
Soft IP Core 部分包括 CAN、DDR_x、FIFO、HyperRAM、I3C、MIPI、Math、Microprocessor System、PSRAM、RAM Based Shift Register、SDRAM Controller、Triple Speed Ethernet MAC 等。


本文档主要介绍 **Hard Module** 模块的使用。

图 3-1 IP Core Generator 界面







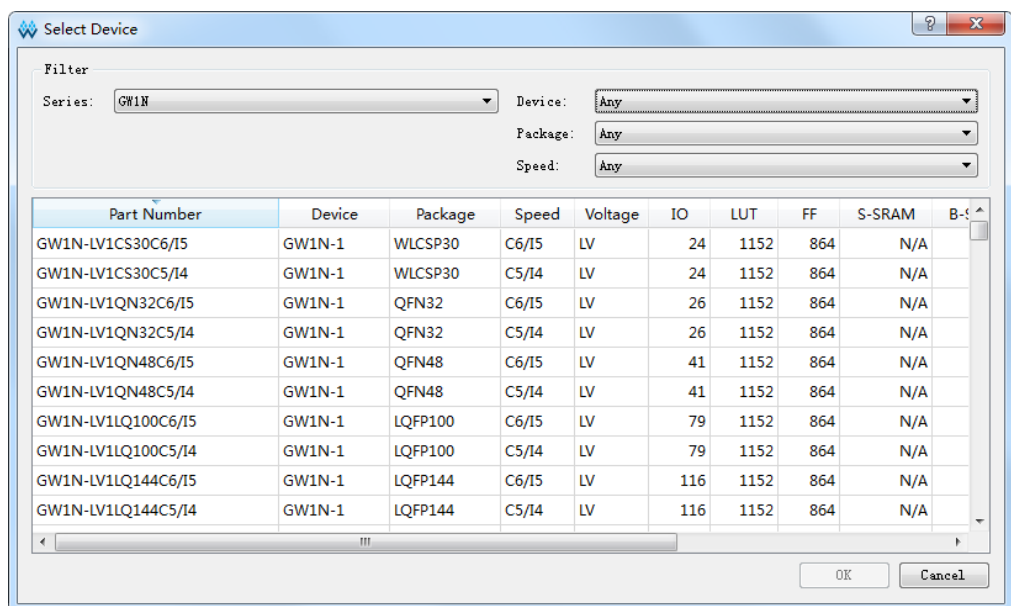
界面顶端有两个控件 **Target Device** 按钮和 **IP Core** 配置文件打开按钮“”:

- **Target Device**, 配置 Device。

选择右侧显示框，弹出“**Select Device**”窗口，如图 3-2 所示。

通过该窗口可修改 **Device** 信息，修改后的内容显示在 **Target Device** 的右侧显示框中，双击某高亮显示的 **IP**，可打开 **IP Customization** 的对话框，修改后的 **Device** 信息也会显示在 **IP Customization** 对话框的 **File** 配置窗口的“**Target Device**”显示框中。

图 3-2 Select Device 窗口



选定 **Device** 后，**IP Core Generator** 会根据设定的 **Device** 自动判断是否

支持特定模块。

- 若支持，模块名为高亮，双击可打开 IP Customization 配置窗口，如图 3-3 中所示，用户可通过 IP Customization 配置窗口来配置 IP，配置完成后点击 OK 生成 IP，各 IP 的配置界面将在第三章的各个小节介绍；
- 若不支持，则模块名为灰色，不可用。如图 3-1 中所示，GW1NR-9 不支持 SPMI。

图标“”，用于打开已配置好的 IP Core 文件，并可根据需要进行编辑。点击图标，弹出“Select IP Config file”对话框，选择 IP Core 配置文件“.ipc”，弹出如图 3-4 所示的“IP Customization”页面，可对配置重新编辑，其中文件路径不可更改。

图 3-3 IP Customization 窗口

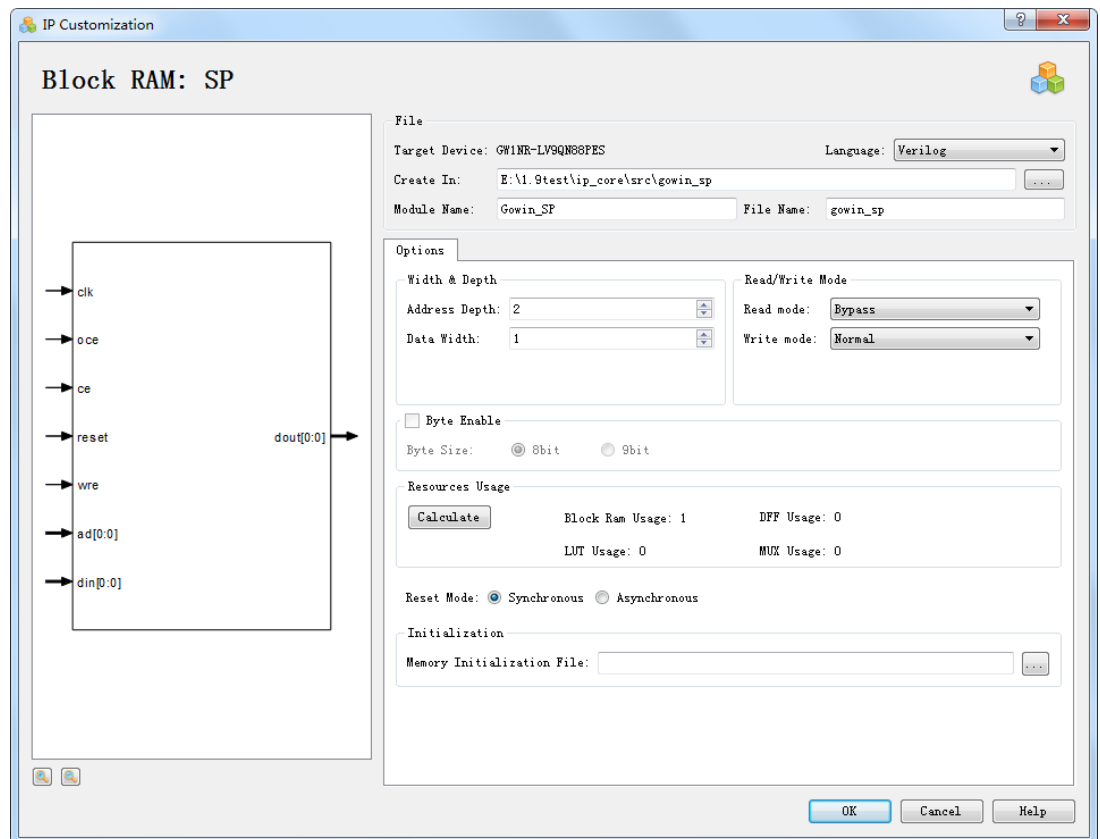
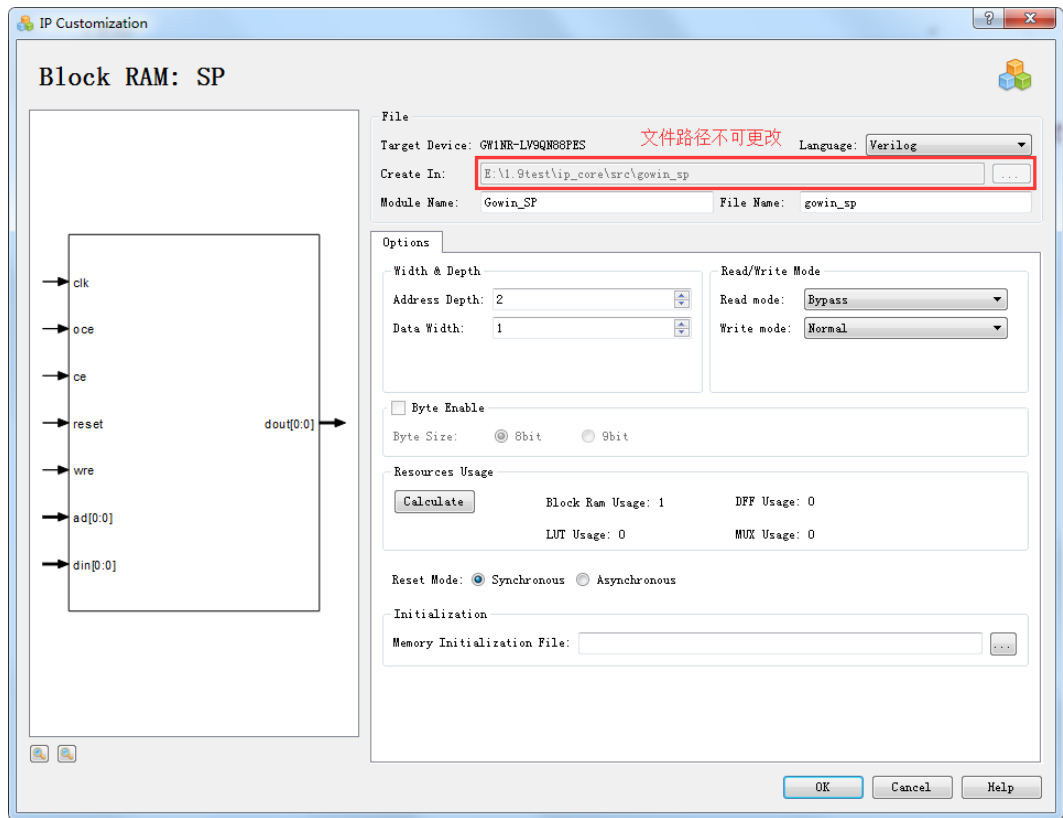


图 3-4 ipc 文件的 IP Customization 窗口



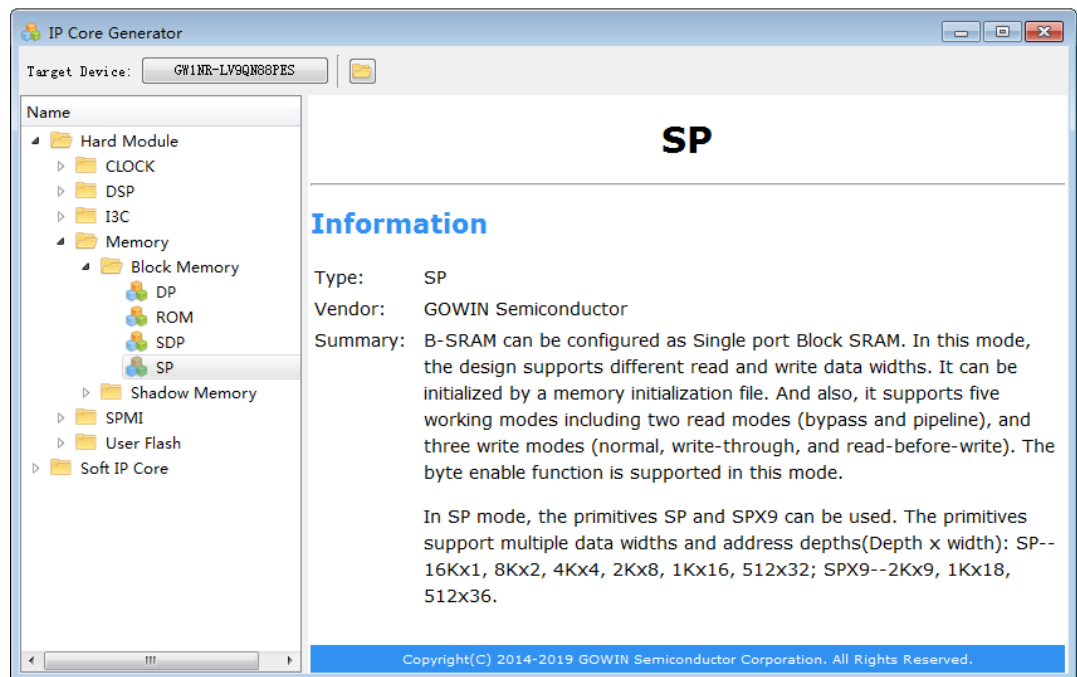
3.1 Block Memory

当前，Block Memory（BSRAM）模块可实现 SP（单端口模式）、SDP（半双端口模式）、DP（双端口模式）和 ROM（只读模式）。

3.1.1 SP

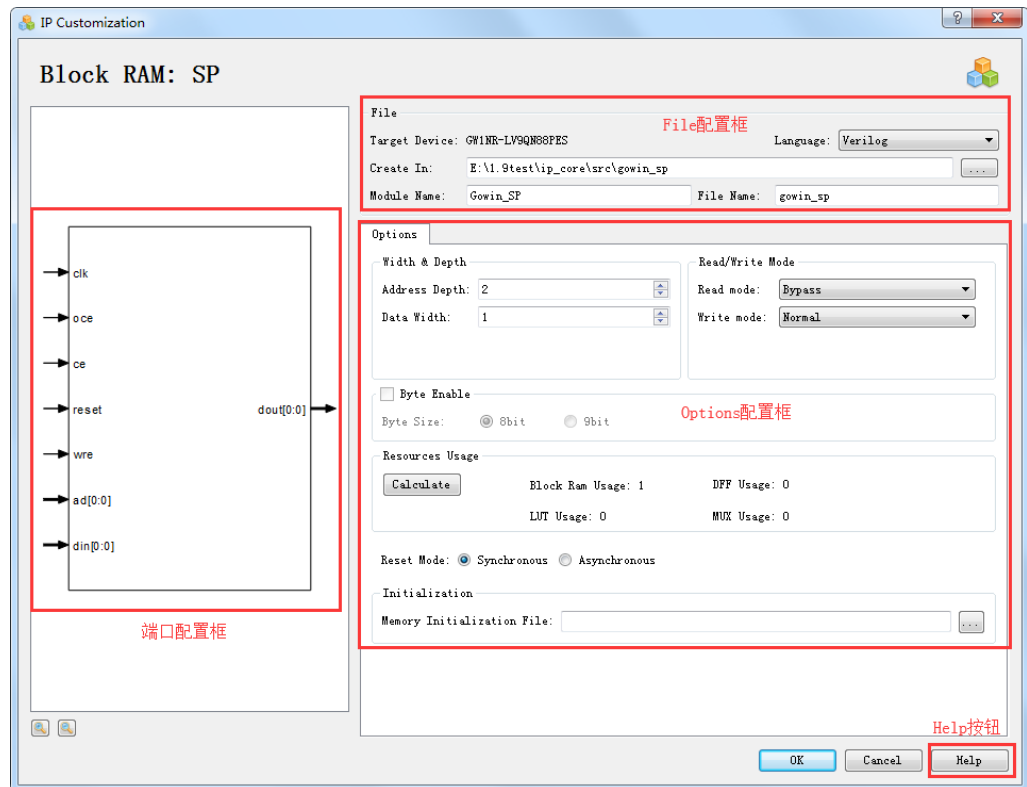
SP 为单端口工作模式，可以通过 SP、SPX9 两种高云器件实现。BSRAM 的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中，单击 SP，界面右侧会显示 SP 的相关信息概要。如图 3-5 所示。

图 3-5 SP 的信息概要



在 IP Core Generator 界面中，双击“SP”，弹出“IP Customization”窗口，该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-6 所示。

图 3-6 SP 的 IP Customization 窗口结构



1. File 配置框

- File 配置框用于配置产生的 SP 实例化文件的相关信息。
- Target Device: 显示已配置的 Device 信息；
- Language: 配置产生的 IP Core 文件的硬件描述语言。选择右侧下拉列表框，选择目标语言，支持 Verilog 和 VHDL，如图 3-7 所示；
- Module Name: 配置产生的 IP Core 文件的 module name。在右侧文本框可重新编辑模块名称。Module Name 不能与原语名称相同，若相同，而报出 Error，提示“Module name is the same as primitive name!”如图 3-8 所示；
- File Name: 配置产生的 IP Core 文件的文件名。在右侧文本框可重新编辑文件名称；
- Create In: 配置产生的 IP Core 文件的目标路径。可在右侧文本框中重新编辑目标路径，也可通过文本框右侧选择按钮选择目标路径；

图 3-7 Language 下拉列表框

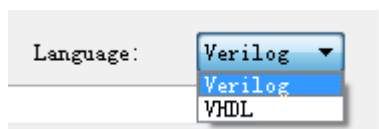
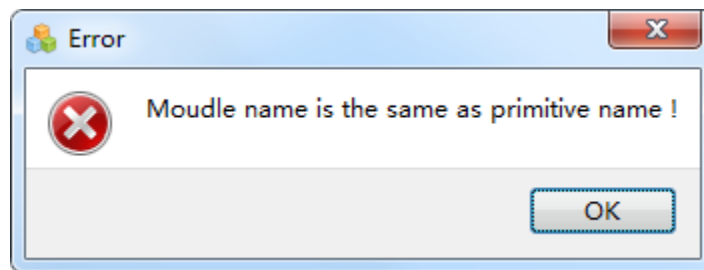


图 3-8 Module 名与原语名称一致时报 Error



2. Options 配置框。

- Options 配置框用于用户自定义单端口模式存储器的配置信息，Options 配置框”如图 3-6 所示。
- Width & Depth: 配置 SP 的地址深度 (Address Depth) 和数据宽度 (Data Width)。当配置的地址深度和数据宽度无法通过单个模块实现时，IP Core 会实例化多个模块组合实现；
- Byte Enable: 用于配置是否使用字节写使能功能，当 Data Width 大于等于 9 时可勾选；当使用字节写使能功能时配置使能字节的大小，可选用 8 bit 或 9 bit；
- Resource Usage: 计算并显示当前容量配置上占用的 Block Ram、DFF、LUT、MUX 的资源情况；
- Read/Write Mode: 配置读写模式。
SP 支持以下模式：
两种读模式：Bypass 和 Pipeline；
三种写模式：Nomal、Write-Through、Read-before-Write。
- Reset Mode: 配置 SP 的复位模式。
支持同步模式 “Synchronous” 和异步模式 “Asynchronous”。
- Initialization: 配置 SP 的 INIT 值。
INIT 值以二进制或十六进制的格式写在初始化文件中，文件可通过 “Memory Initialization File” 指定。

注！

“Memory Initialization File” 选取的初始化文件可通过手写或者 IDE 菜单栏 “File->New->Memory Initialization File” 产生，具体产生方式及初始化文件的格式请参考文档 [《Gowin 云源软件用户指南》](#)。

3. 端口配置框图

- 端口配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-6 中标注的 “配置框图” 所示；
- Options 配置中的地址深度 “Address Depth” 配置影响 ad 的位宽，数据位宽 “Data Width” 配置影响 din 和 dout 的位宽。

4. Help 按钮

单击 “Help”，显示 IP Core 的配置信息的页面，如图 3-9 所示。

图 3-9 Help 信息

SP

Information

Type:	SP
Vendor:	GOWIN Semiconductor
Summary:	<p>B-SRAM can be configured as Single port Block SRAM. In this mode, the design supports different read and write data widths. It can be initialized by a memory initialization file. And also, it supports five working modes including two read modes (bypass and pipeline), and three write modes (normal, write-through, and read-before-write). The byte enable function is supported in this mode.</p> <p>In SP mode, the primitives SP and SPX9 can be used. The primitives support multiple data widths and address depths(Depth x width): SP--16Kx1, 8Kx2, 4Kx4, 2Kx8, 1Kx16, 512x32; SPX9--2Kx9, 1Kx18, 512x36.</p>

Options

Option	Description
Width & Depth	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the data width.
Read/Write Mode	Read Mode - Set whether the read mode is bypass mode or pipeline mode.
	Write Mode - Set the write mode as normal mode, write-through mode or read-before-write mode.
Byte Enable	Byte Enable - Set whether to use byte enable function or not.
	Byte Size - Set whether the byte size is 8bit or 9bit if the byte enable selected.
	Note: Assume that the data width is represented by Width. (1) If Width<=8, byte enable function is invalid; (2) If Width=9, only 8 bit is valid; (3) If Width>9, both 8 bit and 9 bit are valid.
Resource Usage	Calculate - Calculate the resource usage in the design and display results below.
	Block Ram Usage - Display the number of Block Ram used.
	DFF Usage - Display the number of DFF used.
	LUT Usage - Display the number of LUT used.
Reset Mode	MUX Usage - Display the number of MUX used.
	Reset Mode - Set whether the reset mode is synchronous mode or asynchronous mode.
Initialization	Memory Initialization File - Set the memory initialization file (.mi) path.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

IP 生成文件

如图 3-10 所示, SP 的“IP Customization”窗口完成配置后, 单击“OK”, 产生以配置文件“File Name”命名的三个文件:

- 例化高云原语 SP 设计文件“gowin_sp.v”;
- 用户例化该 IP 设计文件的模板文件“gowin_sp_tmp.v”;
- 例化原语 SP 的配置文件“gowin_sp.ipc”。

注!

如配置中选择的语言是 VHDL, 则产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 SP 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-12 所示。

图 3-12 用户例化该 IP 设计文件的模板文件

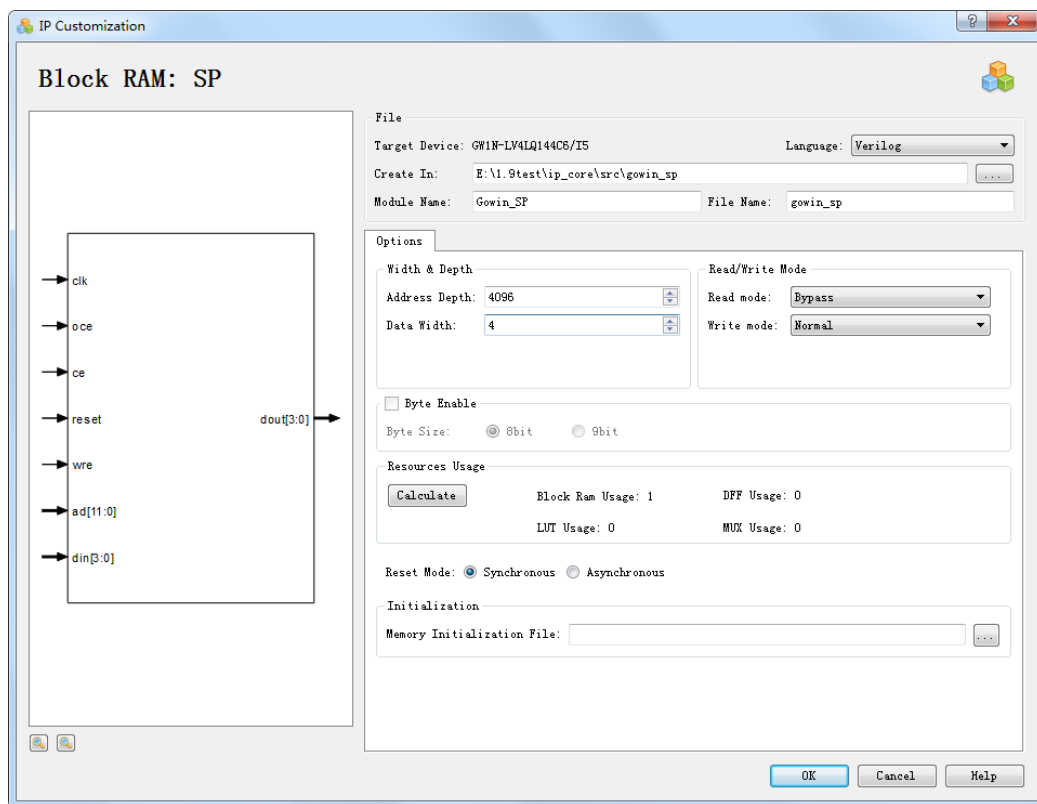
```
Gowin_SP your_instance_name(
    .dout(dout_o), //output [0:0] dout
    .clk(clk_i), //input clk
    .oce(oce_i), //input oce
    .ce(ce_i), //input ce
    .reset(reset_i), //input reset
    .wre(wre_i), //input wre
    .ad(ad_i), //input [0:0] ad
    .din(din_i) //input [0:0] din
);
```

IP Core Generator 生成 SP 示例

如用户需产生 4096X4、Bypass 读模式、Normal 写模式、同步复位模式下的 SP IP，以 device 选择 GW1N-LV4LQ144C6/I5 为例，界面配置如图 3-13 所示，可根据用户需要在 Initialization 窗口配置初始化文件，单击“OK”，产生用户所需的 SP IP 设计文件。

产生的 SP IP 设计文件所在目录即为配置界面中“Create In”设置路径。

图 3-13 SP IP Customization 设置

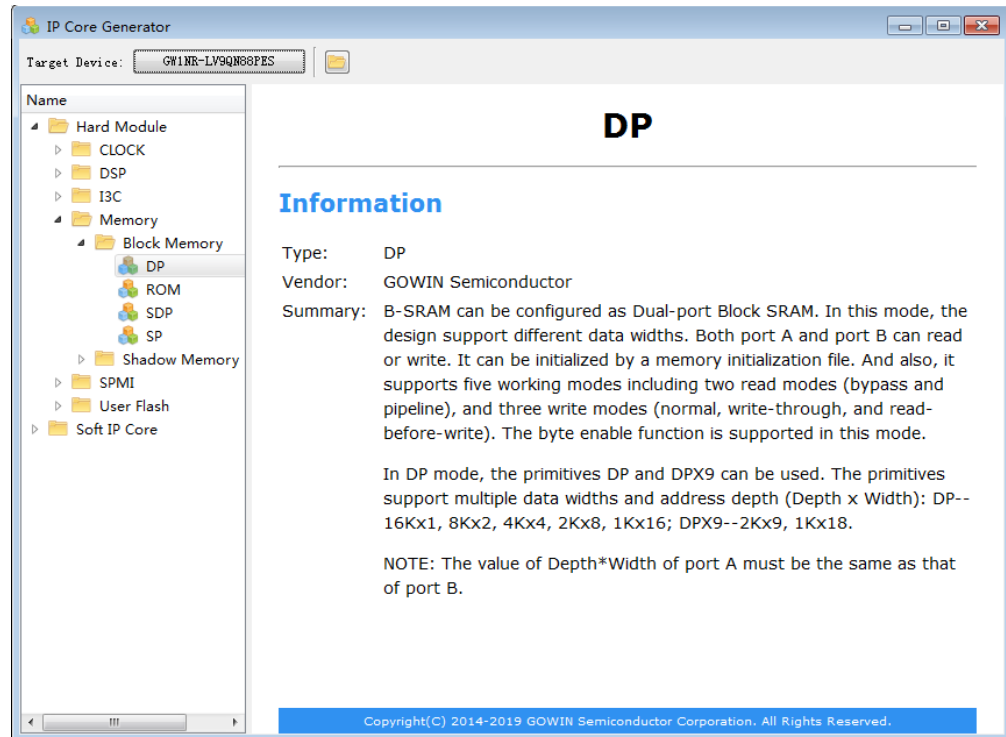


3.1.2 DP

DP 是双端口工作模式，可通过 DP、DPX9 两种高云器件实现，BSRAM

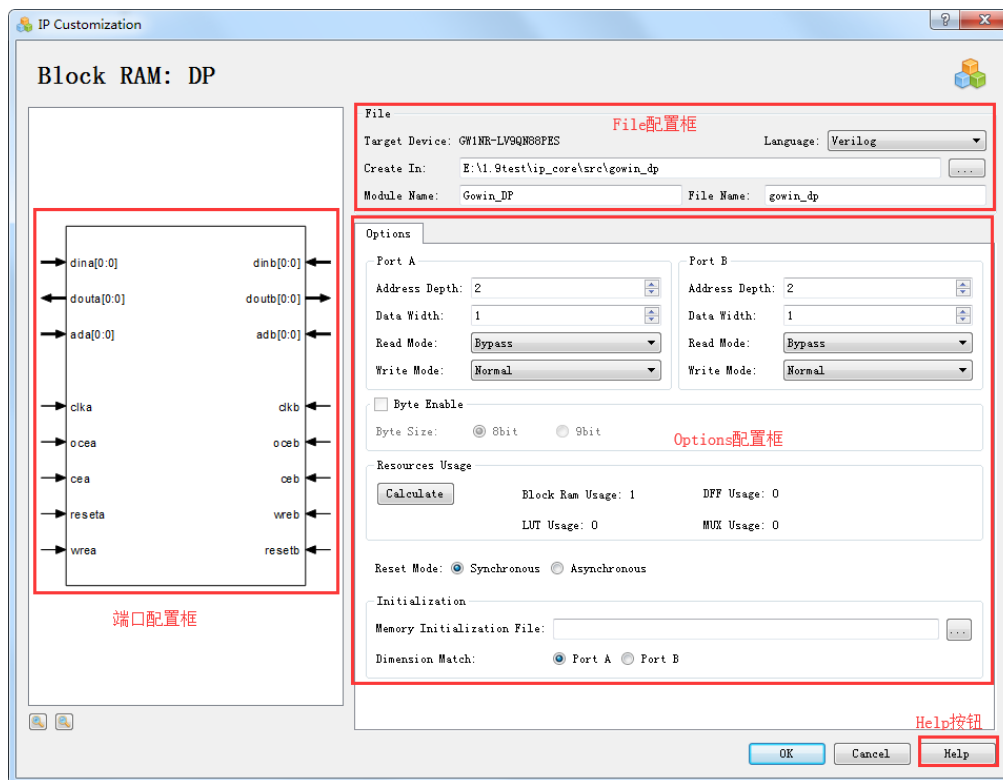
的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中，单击“DP”，界面右侧会显示 DP 的相关信息概要，如图 3-14 所示。

图 3-14 DP 的信息概要



在 IP Core Generator 界面中，双击 DP，弹出 DP 的“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-15 所示。

图 3-15 DP 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 DP 实例化文件的相关信息，如图 3-15 中标注的“File 配置框”所示。

DP 的 File 配置框的使用和 SP 模块类似，具体请参考 [3.1 Block Memory>3.1.1 SP 的 File 配置框](#)。

2. Options 配置框

Options 配置框用于用户自定义双端口模式存储器的配置信息，如图 3-15 中“Options 配置框”所示。

DP 的 Options 配置框的使用和 SP 模块类似，具体请参考 [3.1 Block Memory>3.1.1 SP 中的 Options 配置框](#)。

配置 DP 时，需注意以下事项：

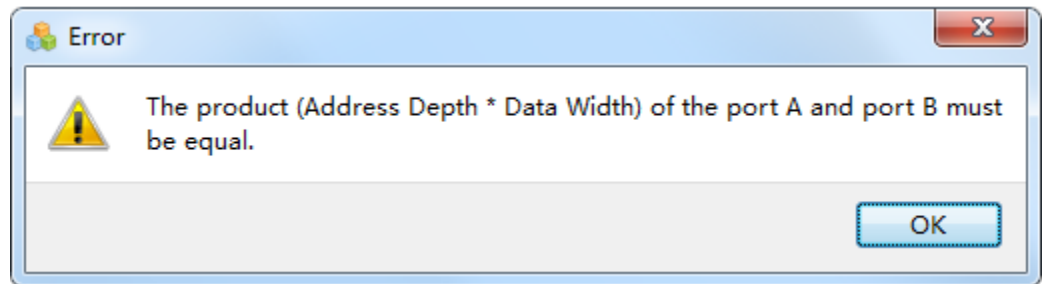
- Options 配置框中可独立配置 DP 的 Port A 和 Port B 的地址深度、数据宽度和读写模式。
- DP 的 Port A 和 Port B 是对同一块 memory 进行读写，因此 Port A 和 Port B 的 Address Depth*Data Width 的结果必须相同。
- Options 配置中的初始化文件（Memory initialization File）中的数据宽度应与 Dimension Match 选择的 Port 数据宽度一致。

注！

- 如 Port A 和 Port B 的 Address Depth*Data Width 的结果不同，则会弹出如图 3-16 所示的 Error 提示信息。
- 如数据宽度不一致，则产生的 DP 实例 Init 值默认初始化为 0，并且在 Output 窗口中，会弹出如下提示信息：

Error (MG2105) : Initial values' width is unequal to user's width.

图 3-16 DP 配置 Error 提示



3. 端口配置框图

- 配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-15 中标注的“端口配置框”所示；
- Options 配置中的 Port A 和 Port B 的地址深度“Address Depth”配置影响 ada 和 adb 的位宽，数据位宽“Data Width”配置影响 dina/douta 和 dinb/doutb 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-17 所示。

图 3-17 Help 信息

DP	
Information	
Type:	DP
Vendor:	GOWIN Semiconductor
Summary:	<p>B-SRAM can be configured as Dual-port Block SRAM. In this mode, the design support different data widths. Both port A and port B can read or write. It can be initialized by a memory initialization file. And also, it supports five working modes including two read modes (bypass and pipeline), and three write modes (normal, write-through, and read-before-write). The byte enable function is supported in this mode.</p> <p>In DP mode, the primitives DP and DPX9 can be used. The primitives support multiple data widths and address depth (Depth x Width): DP--16Kx1, 8Kx2, 4Kx4, 2Kx8, 1Kx16; DPX9--2Kx9, 1Kx18.</p> <p>NOTE: The value of Depth*Width of port A must be the same as that of port B.</p>
Options	
Option	Description
Port A	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the Data width.
	Read Mode - Set whether the read mode is bypass mode or pipeline mode.
	Write Mode - Set the write mode as normal mode, write-through mode or read-before-write mode.
Port B	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the Data width.
	Read Mode - Set whether the read mode is bypass mode or pipeline mode.
Byte Enable	Write Mode - Set the write mode as normal mode, write-through mode or read-before-write mode.
	Byte Enable - Set whether to use byte enable function or not.
Resource Usage	Byte Size - Set whether the byte size is 8bit or 9bit if the byte enable checkbox selected.
	Note: Assume that the data width is represented by Width. (1) If Width<=8, byte enable function is invalid; (2) If Width=9, only 8 bit is valid; (3) If Width>9, both 8 bit and 9 bit are valid.
	Calculate - Calculate the resource usage in the design and display results below.
	Block Ram Usage - Display the number of Block Ram used.
Reset Mode	DFF Usage - Display the number of DFF used.
	LUT Usage - Display the number of LUT used.
	MUX Usage - Display the number of MUX used.
Initialization	Reset Mode - Set whether the reset mode is synchronous mode or asynchronous mode.
	Memory Initialization File - Set the memory initialization file (.mi) path.
	Dimension Match - Set which port's dimensions the memory initialization file should conform to.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

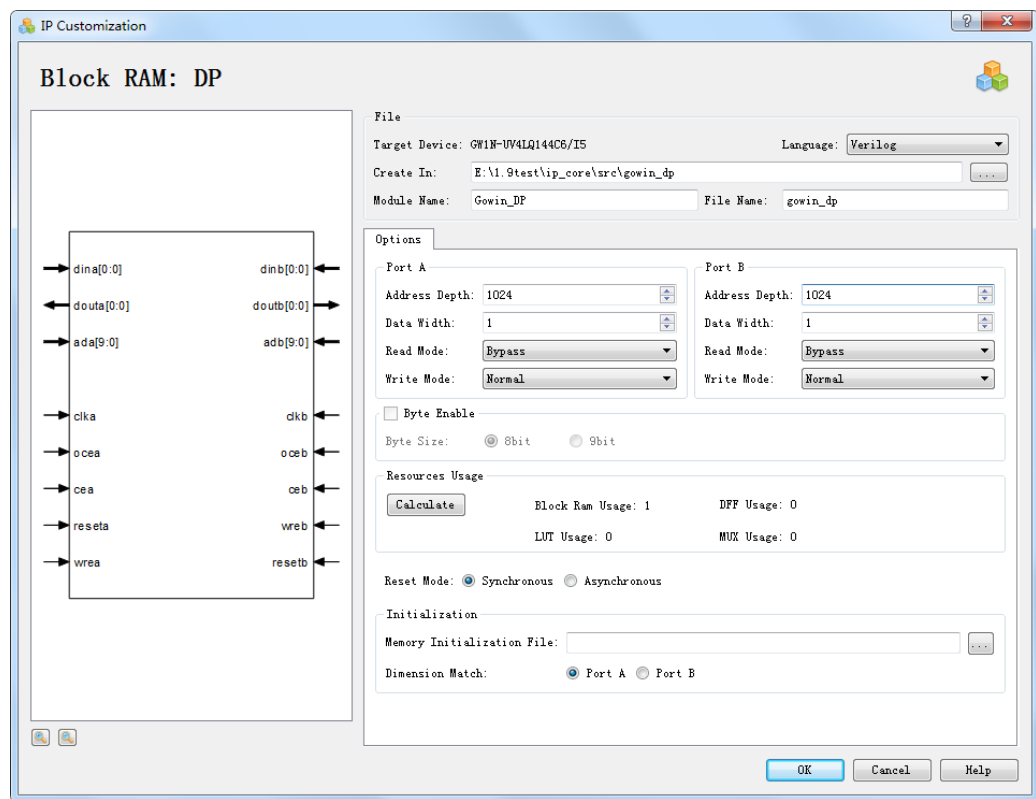
IP 生成文件

如图 3-18 所示, DP 的“IP Customization”窗口配置完成后,单击“OK”,产生以配置文件“File Name”命名的三个文件:

- 例化高云原语 DP 设计文件“gowin_dp.v”;
- 用户例化该 IP 设计文件的模板文件“gowin_dp_tmp.v”;
- 例化原语 DP 的配置文件“gowin_dp.ipc”。

如配置中选择的语言是 VHDL, 产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-18 配置的 IP Customization



例化 DP 设计文件

例化 DP 设计文件为完整的 verilog 模块,模块中根据“IP Customization”中的 DP 配置,产生实例化的 DP,如图 3-19 所示。

图 3-19 例化 DP 设计文件

```

module Gowin_DP (douta, doutb, clka, ocea, cea, reseta, wrea, clkb, oceb, ceb, resetb, wreb, ada, dina, adb, dinb);

output [0:0] douta;
output [0:0] doutb;
input clka;
input ocea;
input cea;
input reseta;
input wrea;
input clkb;
input oceb;
input ceb;
input resetb;
input wreb;
input [0:0] ada;
input [0:0] dina;
input [0:0] adb;
input [0:0] dinb;

wire gw_gnd;

assign gw_gnd = 1'b0;

DP bram_dp_0 (
    .DOA(douta[0]),
    .DOB(doutb[0]),
    .CLKA(clka),
    .OCEA(ocea),
    .CEA(cea),
    .RESETA(reseta),
    .WREA(wrea),
    .CLKB(clkb),
    .OCEB(oceb),
    .CEB(ceb),
    .RESETB(resetb),
    .WREB(wreb),
    .BLKSEL({gw_gnd,gw_gnd,gw_gnd}),
    .ADA({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,ada[0]}),
    .DIA(dina[0]),
    .ADB({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,adb[0]}),
    .DIB(dinb[0])
);

defparam bram_dp_0.READ_MODE0 = 1'b0;
defparam bram_dp_0.READ_MODE1 = 1'b0;
defparam bram_dp_0.WRITE_MODE0 = 2'b00;
defparam bram_dp_0.WRITE_MODE1 = 2'b00;
defparam bram_dp_0.BIT_WIDTH_0 = 1;
defparam bram_dp_0.BIT_WIDTH_1 = 1;
defparam bram_dp_0.BLK_SEL = 3'b000;
defparam bram_dp_0.RESET_MODE = "SYNC";

endmodule //Gowin_DP

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 DP 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-20 所示。

图 3-20 用户例化该 IP 设计文件的模板文件

```

Gowin_DP your_instance_name(
    .douta(douta_o), //output [0:0] douta
    .doutb(doutb_o), //output [0:0] doutb
    .clka(clka_i), //input clka
    .ocea(ocea_i), //input ocea
    .cea(cea_i), //input cea
    .reseta(reseta_i), //input reseta
    .wrea(wrea_i), //input wrea
    .clkb(clkb_i), //input clk
    .oceb(oceb_i), //input oceb
    .ceb(ceb_i), //input ceb
    .resetb(resetb_i), //input resetb
    .wreb(wreb_i), //input wreb
    .ada(ada_i), //input [9:0] ada
    .dina(dina_i), //input [0:0] dina
    .adb(adb_i), //input [9:0] adb
    .dinb(dinb_i) //input [0:0] dinb
);

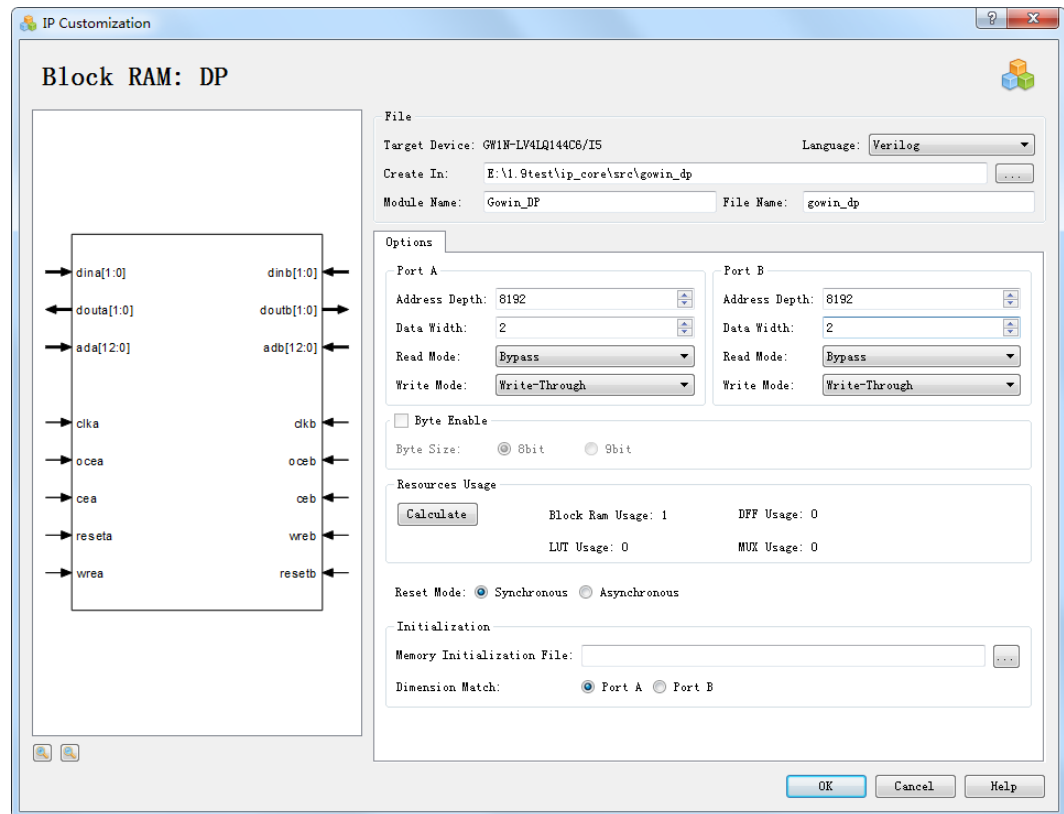
```

IP Core Generator 生成 DP 示例

如用户需产生 8192X2、Bypass 读模式、Write-Through 写模式、同步复位模式下的 DP IP，以 device 选择 GW1N-LV4LQ144C6/I5 为例，界面配置如图 3-21 所示，初始化文件可根据用户需要在 Initialization 窗口配置，单击“OK”，产生用户所需的 DP IP 设计文件。

产生的 DP IP 设计文件所在目录即为配置界面中“Create In”设置路径。

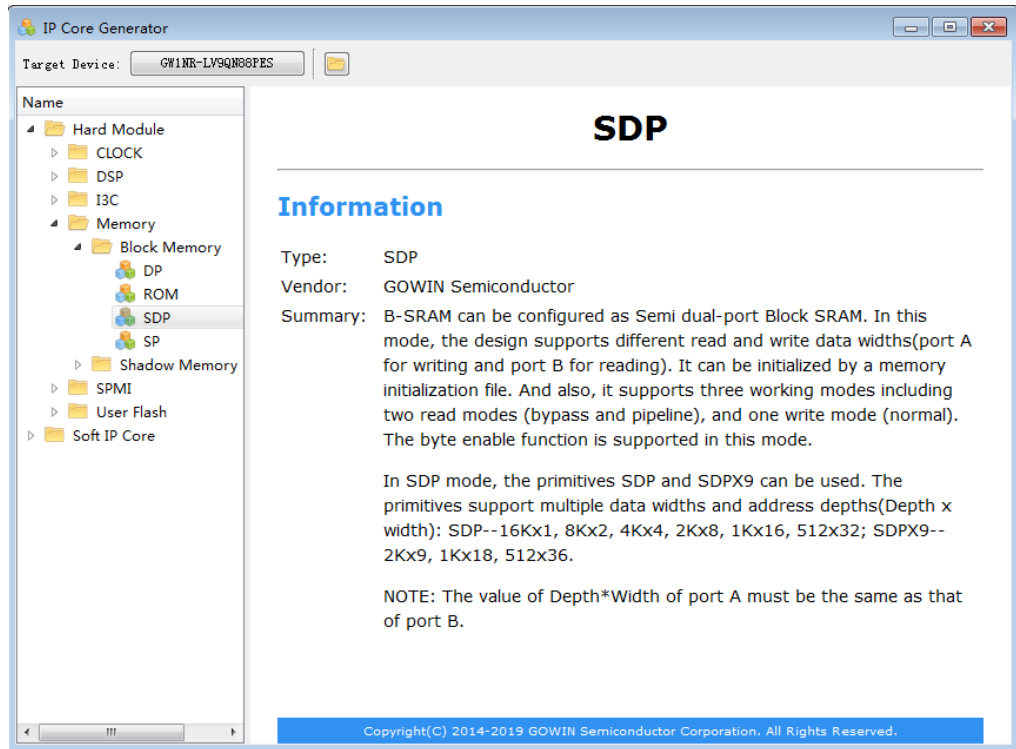
图 3-21 DP IP Customization 设置



3.1.3 SDP

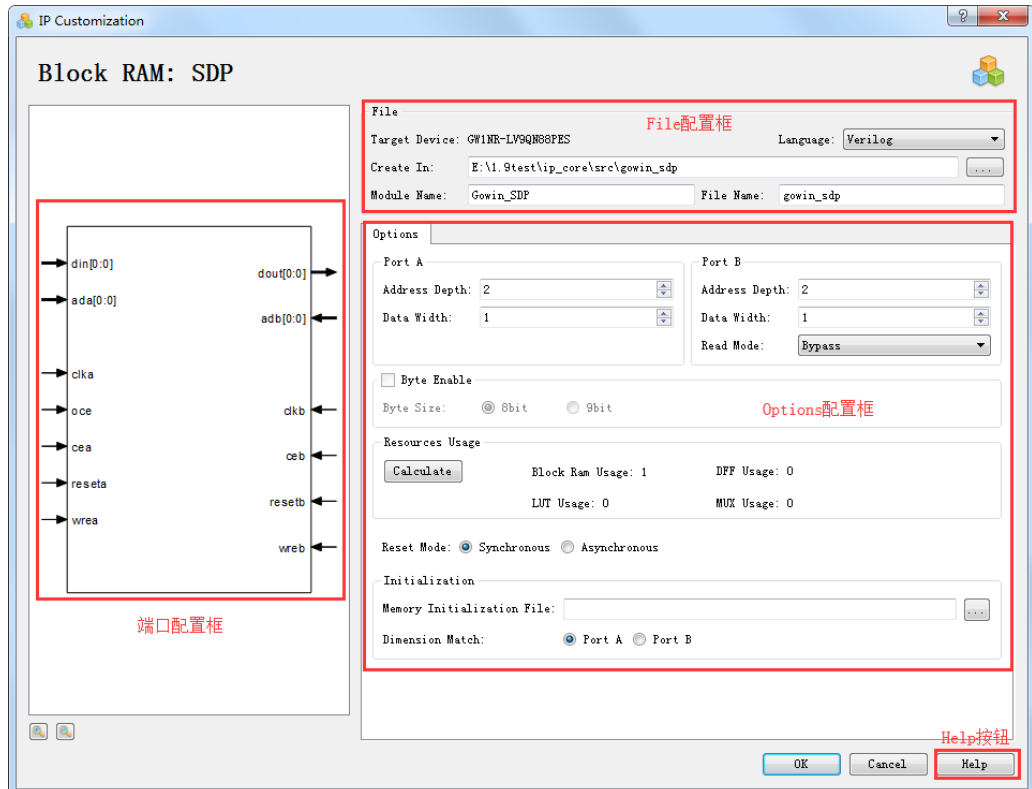
SDP 是半双端口工作模式，可通过 SDP、SDPX9 两种高云器件实现，BSRAM 的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中，单击“SDP”，界面右侧会显示 SDP 的相关信息概要，如图 3-22 所示。

图 3-22 SDP 的信息概要



在 IP Core Generator 界面中，双击“SDP”，弹出“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-23 所示。

图 3-23 SDP 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 SDP 实例化文件的相关信息，标注的“File 配置框”如图 3-23 所示。

SDP 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中 File 配置框介绍。

2. Options 配置框

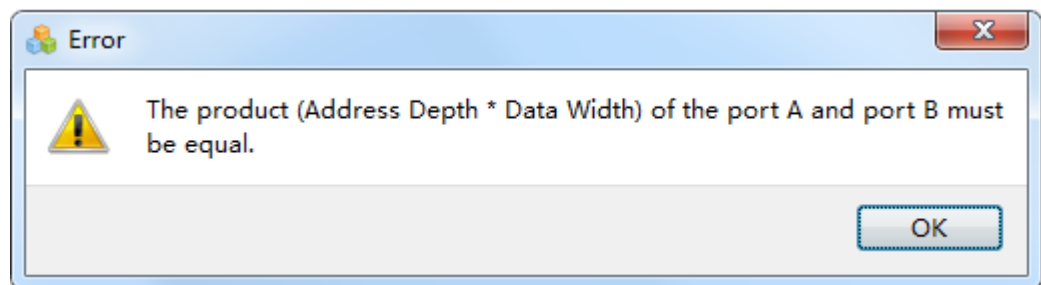
Options 配置框用于用户自定义半双端口模式存储器的配置信息，如图 3-23 中标注的“Options 配置框”所示。

SDP 的 Options 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中 Options 配置框。

注！

- SDP 只支持 PortA 写操作，PortB 读操作；Options 配置框中 Read Mode 配置框可配置 PortB 的 Read Mode 为 Bypass 和 Pipeline；
- Options 配置框中可独立配置 SDP 的 Port A 和 Port B 的地址深度、数据宽度；
- SDP 的 Port A 和 Port B 是对同一块 memory 进行读写，所以 Port A 和 Port B 的 Address Depth*Data Width 的结果须相同，否则会报出如图 3-24 所示的 Error 提示信息；
- Options 配置中的初始化文件(Memory initialization File)中的数据宽度应和 Dimension Match 选择的 Port 数据宽度一致，否则产生的 SDP 实例 Init 值默认初始化为 0，并且会在 Output 窗口显示如下信息：Error (MG2105) : Initial values' width is unequal to user's width.

图 3-24 SDP 配置 Error 信息



3. 端口配置框图

- 配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-23 中标注的“配置框图”所示；
- Options 配置中的 Port A 的地址深度“Address Depth”配置影响 ada 的位宽，数据位宽“Data Width”配置影响 din 的位宽；PortB 的地址深度“Address Depth”配置影响 adb 的位宽，数据位宽“Data Width”配置影响 dout 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息页面，如图 3-25 所示。

图 3-25 Help 信息

SDP	
Information	
Type:	SDP
Vendor:	GOWIN Semiconductor
Summary:	<p>B-SRAM can be configured as Semi dual-port Block SRAM. In this mode, the design supports different read and write data widths(port A for writing and port B for reading). It can be initialized by a memory initialization file. And also, it supports three working modes including two read modes (bypass and pipeline), and one write mode (normal). The byte enable function is supported in this mode.</p> <p>In SDP mode, the primitives SDP and SDPX9 can be used. The primitives support multiple data widths and address depths(Depth x width): SDP--16Kx1, 8Kx2, 4Kx4, 2Kx8, 1Kx16, 512x32; SDPX9--2Kx9, 1Kx18, 512x36.</p> <p>NOTE: The value of Depth*Width of port A must be the same as that of port B.</p>
Options	
Option	Description
Port A	<p>Address Depth - Set the size of the address depth.</p> <p>Data Width - Set the size of the Data width.</p>
Port B	<p>Address Depth - Set the size of the address depth.</p> <p>Data Width - Set the size of the Data width.</p> <p>Read Mode - Set whether the read mode is bypass mode or pipeline mode.</p>
Byte Enable	<p>Byte Enable - Set whether to use byte enable function or not.</p> <p>Byte Size - Set whether the byte size is 8bit or 9bit if the byte enable checkbox selected.</p> <p>Note: Assume that the data width is represented by Width. (1) If Width<=8, byte enable function is invalid; (2) If Width=9, only 8 bit is valid; (3) If Width>9, both 8 bit and 9 bit are valid.</p>
Resource Usage	<p>Calculate - Calculate the resource usage in the design and display results below.</p> <p>Block Ram Usage - Display the number of Block Ram used.</p> <p>DFF Usage - Display the number of DFF used.</p> <p>LUT Usage - Display the number of LUT used.</p> <p>MUX Usage - Display the number of MUX used.</p>
Reset Mode	Reset Mode - Set whether the reset mode is synchronous mode or asynchronous mode.
Initialization	<p>Memory Initialization File - Set the memory initialization file (.mi) path.</p> <p>Dimension Match - Set which port's dimensions the memory initialization file should conform to.</p>

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

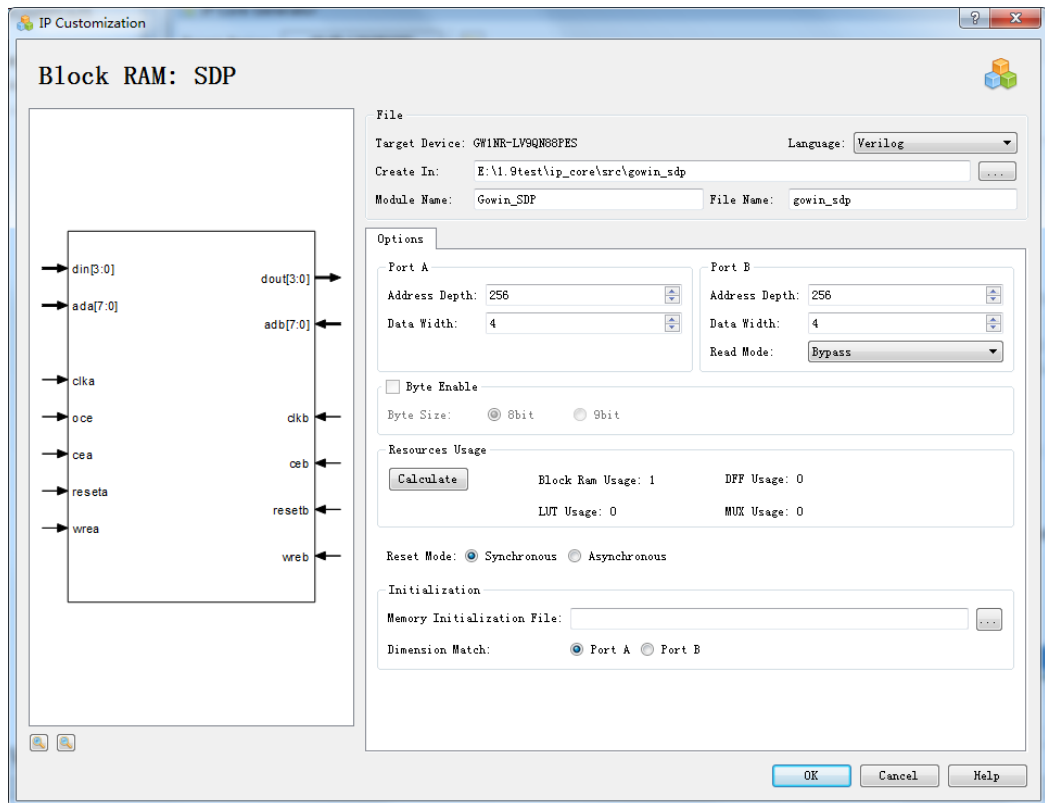
IP 生成文件

如图 3-26 所示，SDP 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 SDP 设计文件“gowin_sdp.v”；
- 用户例化该 IP 设计文件的模板文件“gowin_sdp_tmp.v”；
- 例化原语 SDP 的配置文件“gowin_sdp.ipc”。

如配置中选择的语言是 VHDL，则产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-26 配置的 IP Customization



例化 SDP 设计文件

例化 SDP 设计文件为完整的 verilog 模块, 模块中根据“IP Customization”中的 SDP 配置, 产生实例化的 SDP, 如图 3-27 所示。

注!

产生的实例 SDP 的 din、dout 的数据宽度和“IP Customization”中的 SDP 配置一致。

图 3-27 例化 SDP 设计文件

```

module Gowin_SDP (dout, clka, cea, reseta, wrea, clkb, ceb, resetb, wreb, oce, ada, din, adb);

output [3:0] dout;
input clka;
input cea;
input reseta;
input wrea;
input clkb;
input ceb;
input resetb;
input wreb;
input oce;
input [7:0] ada;
input [3:0] din;
input [7:0] adb;

wire gw_gnd;

assign gw_gnd = 1'b0;

SDP bram_sdp_0 (
    .DO(dout[3:0]),
    .CLKA(clka),
    .CEA(cea),
    .RESETA(reseta),
    .WREA(wrea),
    .CLKB(clkb),
    .CEB(ceb),
    .RESETB(resetb),
    .WREB(wreb),
    .OCE(oce),
    .BLKSEL({gw_gnd,gw_gnd,gw_gnd}),
    .ADA({gw_gnd,gw_gnd,gw_gnd,gw_gnd,ada[7:0],gw_gnd,gw_gnd}),
    .DI(din[3:0]),
    .ADB({gw_gnd,gw_gnd,gw_gnd,gw_gnd,adb[7:0],gw_gnd,gw_gnd})
);

defparam bram_sdp_0.READ_MODE = 1'b0;
defparam bram_sdp_0.BIT_WIDTH_0 = 4;
defparam bram_sdp_0.BIT_WIDTH_1 = 4;
defparam bram_sdp_0.BLK_SEL = 3'b000;
defparam bram_sdp_0.RESET_MODE = "SYNC";

endmodule //Gowin_SDP

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 SDP 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-28 所示。

图 3-28 用户例化该 IP 设计文件的模板文件

```

Gowin_SDP your_instance_name(
    .dout(dout_o), //output [3:0] dout
    .clka(clka_i), //input clka
    .cea(cea_i), //input cea
    .reseta(reseta_i), //input reseta
    .wrea(wrea_i), //input wrea
    .clkb(clkb_i), //input clkb
    .ceb(ceb_i), //input ceb
    .resetb(resetb_i), //input resetb
    .wreb(wreb_i), //input wreb
    .oce(oce_i), //input oce
    .ada(ada_i), //input [7:0] ada
    .din(din_i), //input [3:0] din
    .adb(adb_i) //input [7:0] adb
);

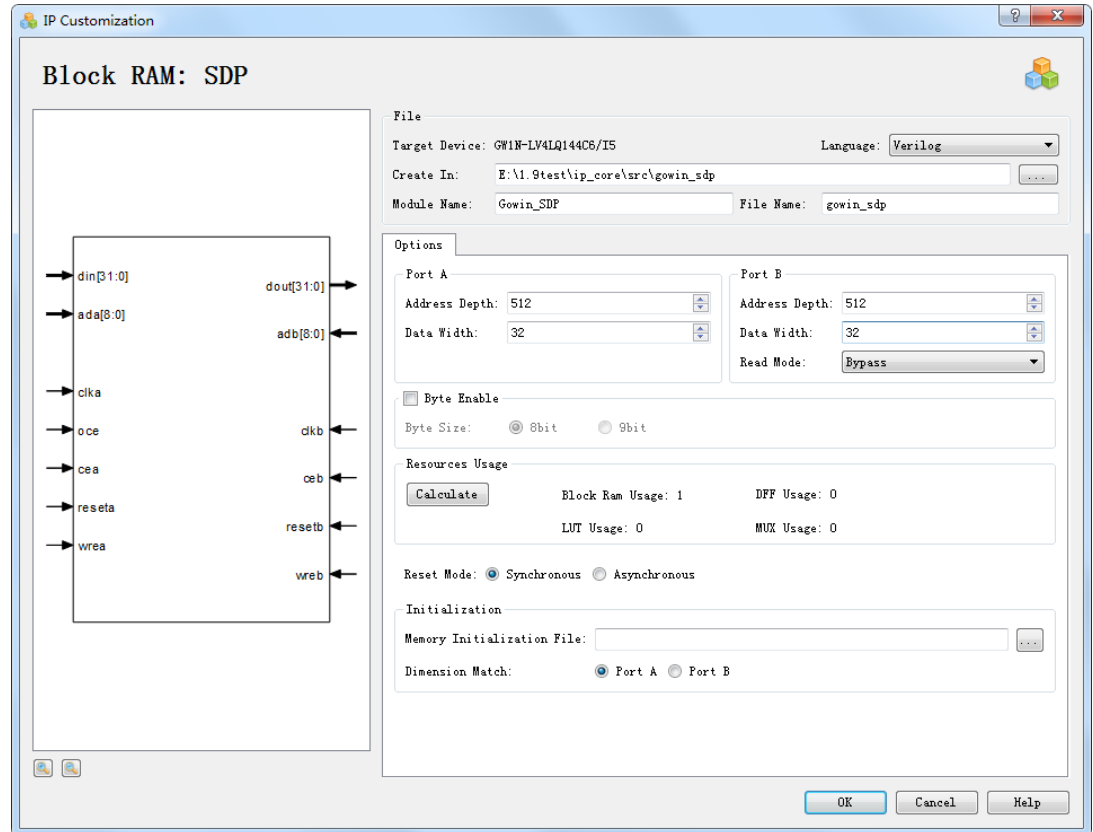
```

IP Core Generator 生成 SDP 示例

如用户需产生 512X32、Bypass 读模式、同步复位模式下的 SDP IP，以 device 选择 GW1N-LV4LQ144C6/I5 为例，界面配置如图 3-29 所示，初始化文件可根据用户需要在 Initialization 窗口配置，单击“OK”，产生用户所需的 SDP IP 设计文件。

产生的 SDP IP 设计文件所在目录即为配置界面中“Create In”设置路径。

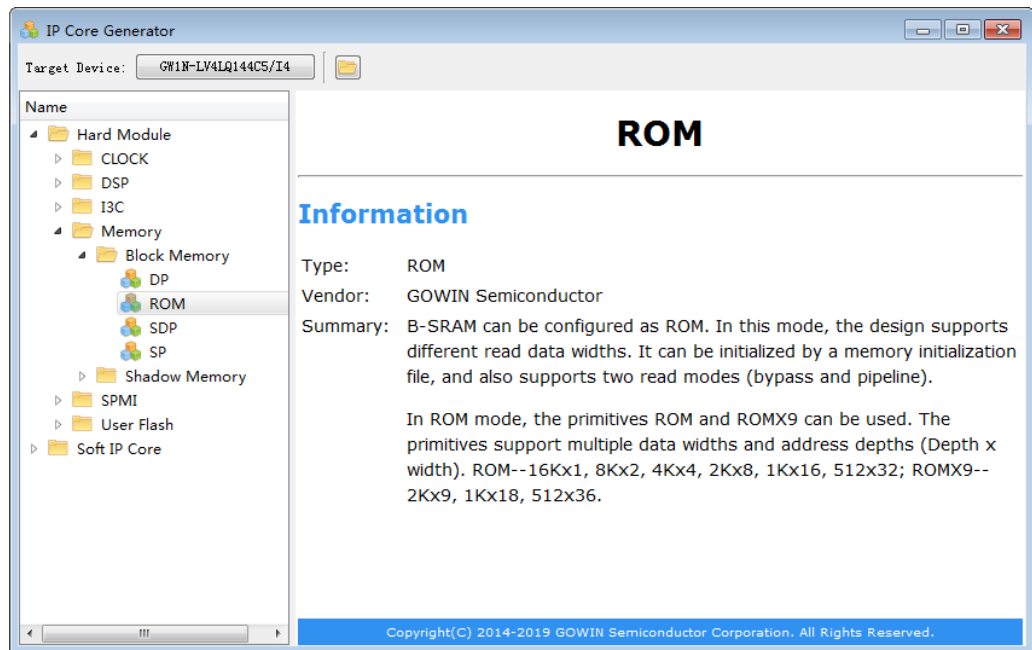
图 3-29 SDP 的 IP Customization 设置



3.1.4 ROM

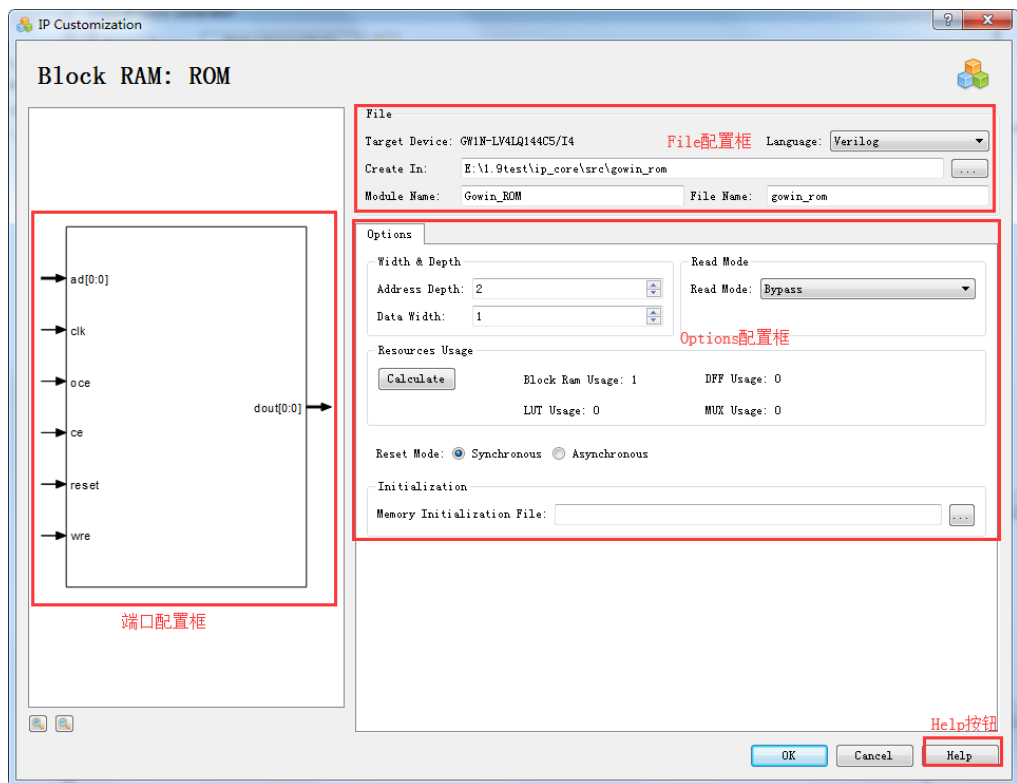
ROM 是只读模式，可通过 ROM、ROMX9 两种高云器件实现，BSRAM 的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中，单击“ROM”，界面右侧会显示 ROM 的相关信息概要，如图 3-30 所示。

图 3-30 ROM 的信息概要



在 IP Core Generator 界面中，双击“ROM”，弹出 ROM 的“IP Customization”窗口。窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-31 所示。

图 3-31 ROM 的 IP Customization 窗口结构



1. File 配置框

File 配置窗口用于配置产生的 ROM 实例化文件的相关信息，如图 3-31

中标注的“File 配置框”所示。

ROM 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于用户自定义只读端口模式存储器的配置信息，如图 3-31 中标注的“Options 配置框”所示。

ROM 的 Options 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 Options 配置。

注！

- ROM 的工作模式为只读模式，Options 配置框中可配置的 Read Mode 为 Bypass 和 Pipeline。
- Options 配置中的初始化文件（Memory initialization File）中的数据宽度应和配置中的数据宽度（Data Width）保持一致，否则产生的 ROM 实例 Init 值默认初始化为 0，并在 Output 窗口中弹出如下信息：Error (MG2105) : Initial values' width is unequal to user's width.

3. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-31 中标注的“配置框图”所示；

Options 配置中的地址深度“Address Depth”配置影响 ad 的位宽，数据位宽“Data Width”配置影响 dout 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息页面，如图 3-32 所示。

图 3-32 Help 信息

ROM	
Information	
Type:	ROM
Vendor:	GOWIN Semiconductor
Summary:	B-SRAM can be configured as ROM. In this mode, the design supports different read data widths. It can be initialized by a memory initialization file, and also supports two read modes (bypass and pipeline). In ROM mode, the primitives ROM and ROMX9 can be used. The primitives support multiple data widths and address depths (Depth x width). ROM--16Kx1, 8Kx2, 4Kx4, 2Kx8, 1Kx16, 512x32; ROMX9--2Kx9, 1Kx18, 512x36.
Options	
Option	Description
Width & Depth	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the data width.
Read Mode	Read Mode - Set whether the read mode is bypass mode or pipeline mode.
Resources Usage	Calculate - Calculate the resource usage in the design and display results below.
	Block Ram Usage - Display the number of Block Ram used.
	DFF Usage - Display the number of DFF used.
	LUT Usage - Display the number of LUT used.
	MUX Usage - Display the number of MUX used.
Reset Mode	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
Initialization	Memory Initialization File - Set the memory initialization file (.mi) path.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

IP 生成文件

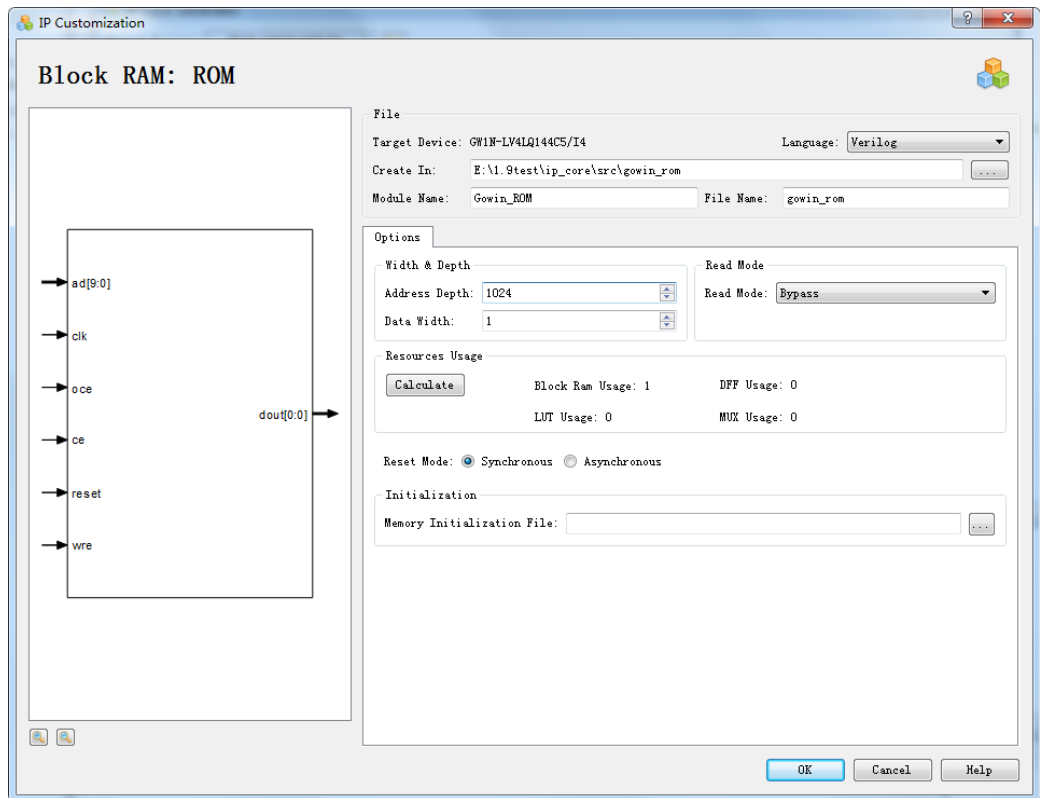
如图 3-33 所示，ROM 的“IP Customization”窗口配置完成后，单击

“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 ROM 设计文件“gowin_rom.v”；
- 用户例化该 IP 设计文件的模板文件“gowin_rom_tmp.v”；
- 例化原语 ROM 的配置文件“gowin_rom.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-33 配置的 IP Customization



例化 ROM 设计文件

例化 ROM 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 ROM 配置，产生实例化的 ROM，如图 3-34 所示。

注！

产生的实例 ROM 的 dout 的数据宽度和“IP Customization”中的 ROM 配置一致。

图 3-34 例化 ROM 设计文件

```

module Gowin_ROM (dout, clk, oce, ce, reset, wre, ad);

    output [0:0] dout;
    input clk;
    input oce;
    input ce;
    input reset;
    input wre;
    input [9:0] ad;

    wire gw_gnd;

    assign gw_gnd = 1'b0;

    ROM bram_rom_0 (
        .DO(dout[0]),
        .CLK(clk),
        .OCE(oce),
        .CE(ce),
        .RESET(reset),
        .WRE(wre),
        .BLKSEL({gw_gnd,gw_gnd,gw_gnd}),
        .AD({gw_gnd,gw_gnd,gw_gnd,gw_gnd,ad[9:0]})
    );

    defparam bram_rom_0.READ_MODE = 1'b0;
    defparam bram_rom_0.BIT_WIDTH = 1;
    defparam bram_rom_0.BLK_SEL = 3'b000;
    defparam bram_rom_0.RESET_MODE = "SYNC";

endmodule //Gowin_ROM

```

用户例化 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 ROM 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-35 所示。

图 3-35 用户例化 IP 设计文件的模板文件

```

Gowin_ROM your_instance_name(
    .dout(dout_o), //output [0:0] dout
    .clk(clk_i), //input clk
    .oce(oce_i), //input oce
    .ce(ce_i), //input ce
    .reset(reset_i), //input reset
    .wre(wre_i), //input wre
    .ad(ad_i) //input [9:0] ad
);

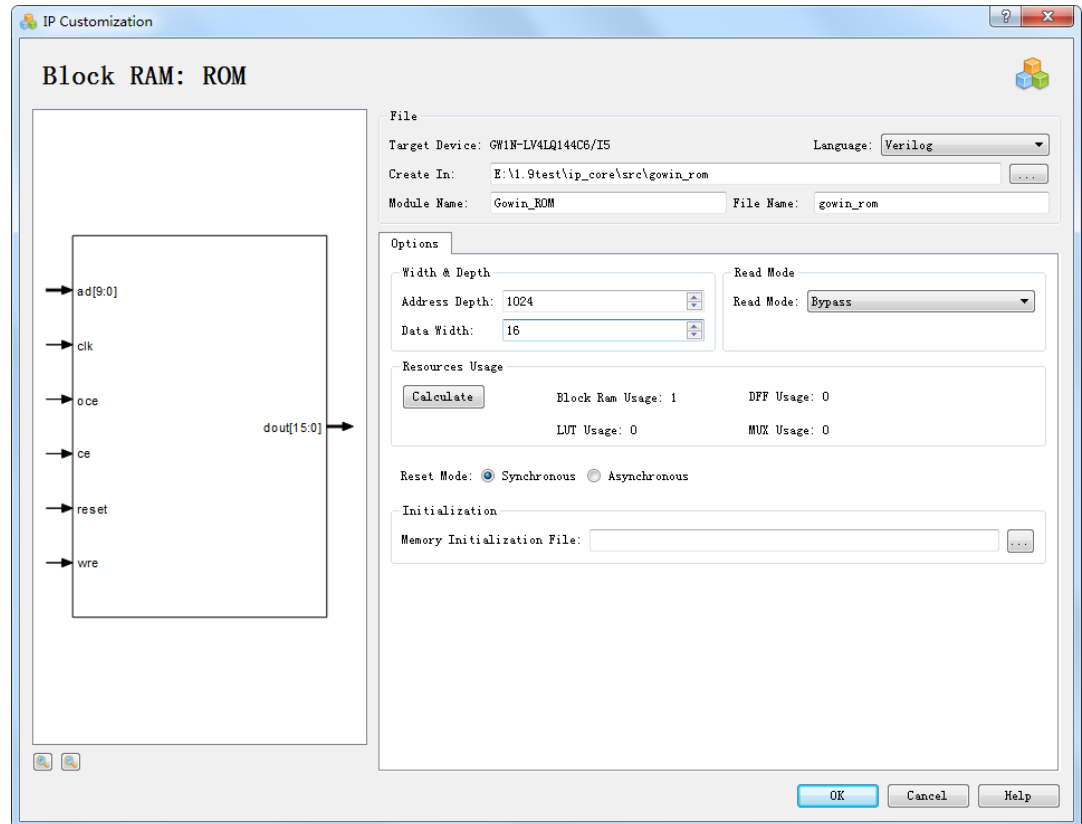
```

IP Core Generator 生成 ROM 示例

如用户需产生 1024X16、Bypass 读模式、同步复位模式下的 ROM IP，以 device 选择 GW1N-LV4LQ144C6/I5 为例，界面配置如图 3-36 所示，可根据用户需要，在 Initialization 窗口配置初始化文件，单击“OK”，产生用户所需的 ROM IP 设计文件。

产生的 ROM IP 设计文件所在目录即为配置界面中“Create In”设置路径。

图 3-36 ROM 的 IP Customization 设置



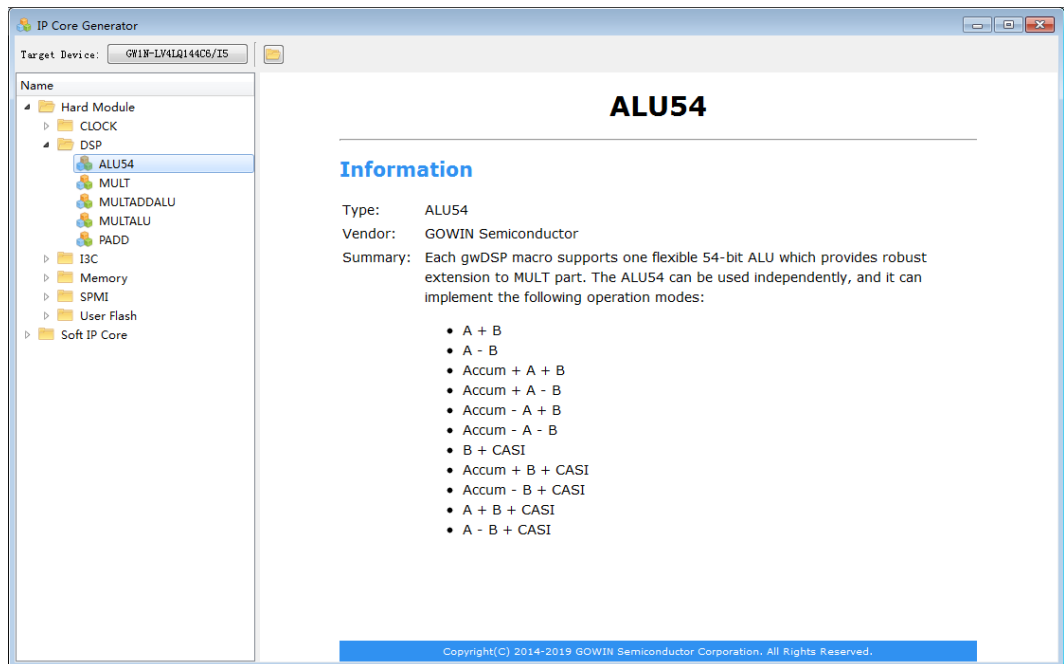
3.2 DSP

当前，DSP 模块支持五种高云器件的产生：ALU54、MULT、MULTADDALU、MULTALU、PADD。

3.2.1 ALU54

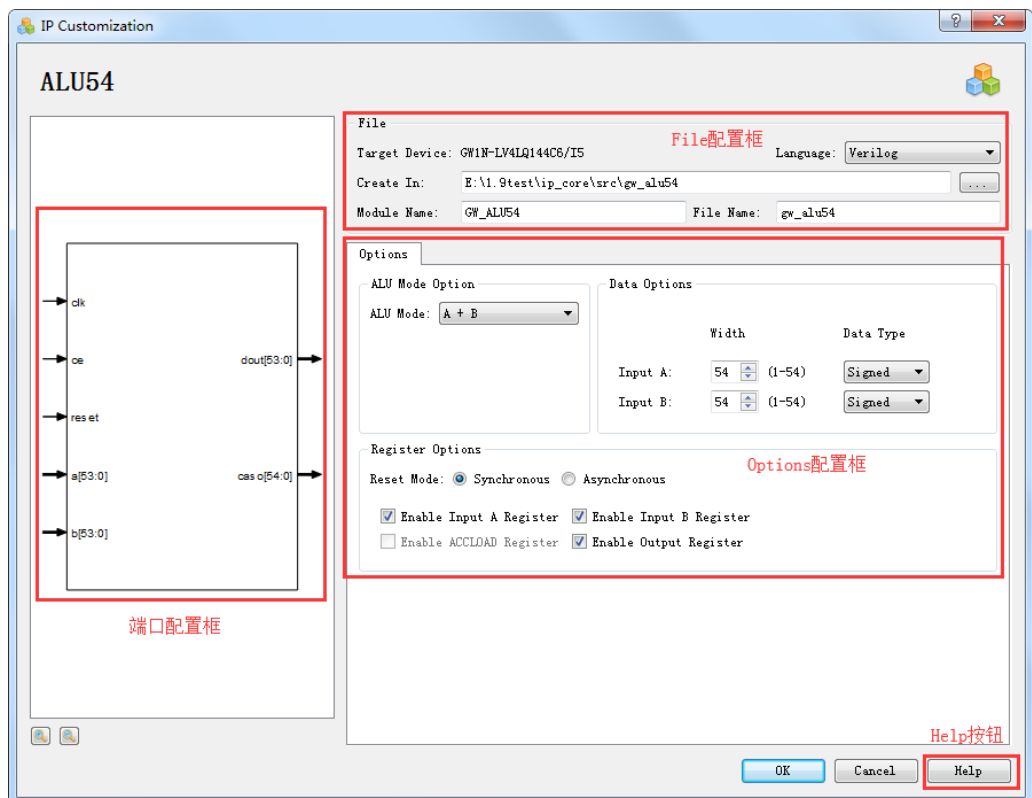
ALU54 实现 54 位算数逻辑运算。在 IP Core Generator 界面中，单击“ALU54”，界面右侧会显示 ALU54 的相关信息概要，如图 3-37 所示。

图 3-37 ALU54 的信息概要



在 IP Core Generator 界面中，双击“ALU54”，弹出 ALU54 的“IP Customization”窗口，如图 3-38 所示。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”。

图 3-38 ALU54 的 IP Customization 窗口结构



5. File 配置框

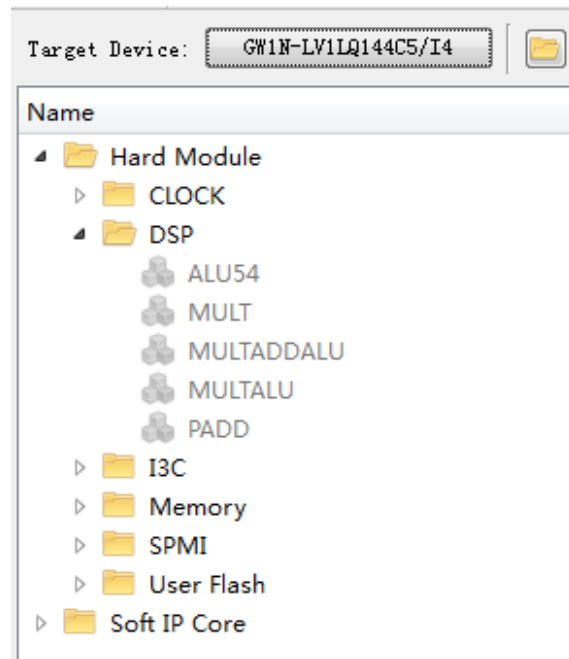
File 配置框用于配置产生 ALU54 实例化文件的相关信息，标注的“File 配置框”如图 3-38 所示。

ALU54 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

注！

当选择 Device GW1N-1 时，由于该 Device 不支持 DSP，在 IP Core Generator 界面的左侧，DSP 的各个模块置灰不可选，如图 3-39 所示。

图 3-39 DSP 部分界面变灰



6. Options 配置框

Options 配置框用于配置例化高云原语 ALU54 设计文件中 ALU54 的配置信息，标注的“Options 配置框”如图 3-38 所示。

- **ALU Mode Option:** 配置 ALU54 的运算模式。可选择：

- A + B;
- A - B;
- Accum + A + B;
- Accum + A - B;
- Accum - A + B;
- Accum - A - B;
- B + CASI;
- Accum + B + CASI;
- Accum - B + CASI;
- A + B + CASI;
- A - B + CASI;

- **Data Options:** 配置数据选项。

配置 ALU54 输入数据位宽。输入 A/B 端的数据可配置为 1-54 位；输出端口数据位宽无需用户配置，其会根据输入位宽自动调整位宽；“Data Type”选项可配置为 Signed、Unsigned。

- Register Options: 配置寄存器工作模式。
 - “Reset Mode” 选项配置 ALU54 的复位模式，支持同步模式 “Synchronous” 和异步模式 “Asynchronous”；
 - “Enable Input A Register” 配置 Input A register；
 - “Enable Input B Register” 配置 Input B register；
 - “Enable ACCLOAD Register” 配置 ACCLOAD register；
 - “Enable Output Register” 配置 Output register。

7. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，标注的“配置框图”如图 3-38 所示。

8. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-40 所示。

图 3-40 Help 信息

ALU54	
Information	
Type:	ALU54
Vendor:	GOWIN Semiconductor
Summary:	<p>Each gwDSP macro supports one flexible 54-bit ALU which provides robust extension to MULT part. The ALU54 can be used independently, and it can implement the following operation modes:</p> <ul style="list-style-type: none"> • A + B • A - B • Accum + A + B • Accum + A - B • Accum - A + B • Accum - A - B • B + CASI • Accum + B + CASI • Accum - B + CASI • A + B + CASI • A - B + CASI
Options	
Option	Description
ALU54 Mode Option	ALU54 Mode - Set one of the ALU54 operation modes.
Data Options	Input A Width - Set the size of the first item in the ALU54.
	Input B Width - Set the size of the second item in the ALU54.
	Data Type - Set the data format of the inputs as signed or unsigned.
Register Options	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
	Enable ... Register - Enable or disable registers. For example, if you choose Enable Input A Register, the input data will go through one register.

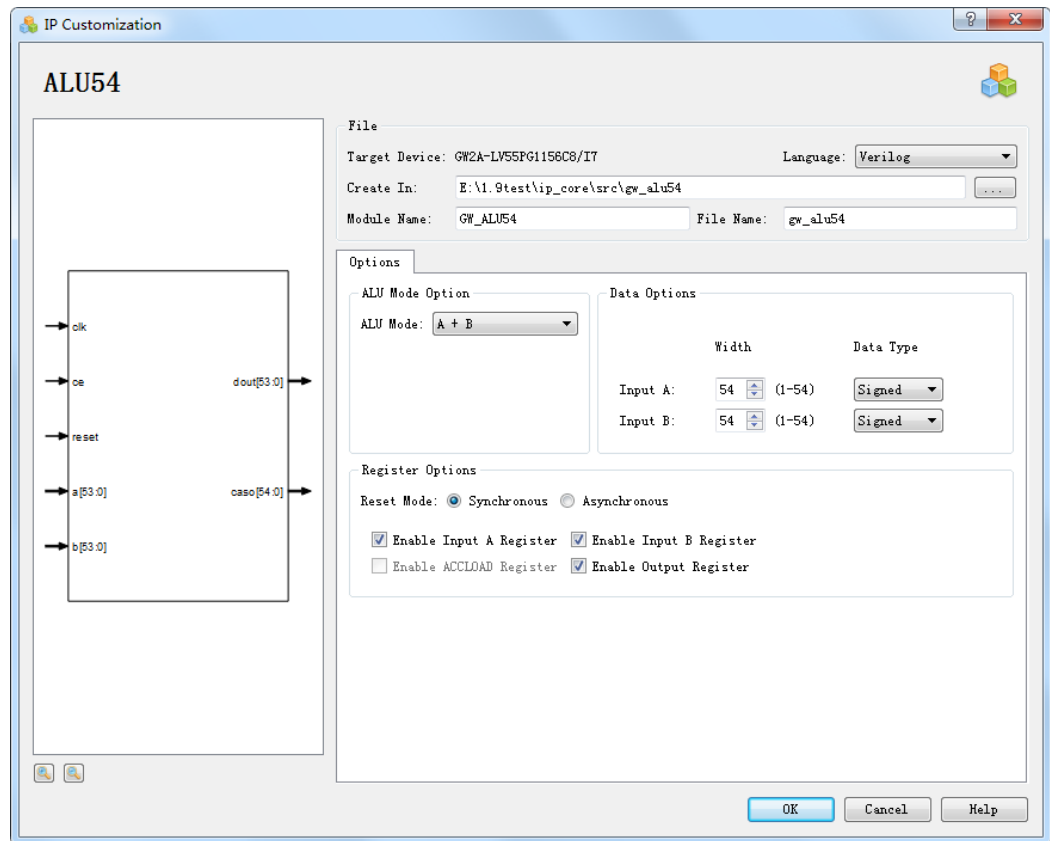
IP 生成文件

如图 3-41 所示，ALU54 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 ALU54 设计文件 “gw_alu54.v”；
- 用户例化该 IP 设计文件的模板文件 “gw_alu54_tmp.v”；
- 例化原语 ALU54 的配置文件 “gw_alu54.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-41 配置的 IP Customization



例化 ALU54 设计文件

例化 ALU54 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 ALU54 配置，产生了实例化的 ALU54，如图 3-42 所示。

图 3-42 例化 ALU54 设计文件

```

module GW_ALU54 (dout, caso, a, b, ce, clk, reset);

output [53:0] dout;
output [54:0] caso;
input [53:0] a;
input [53:0] b;
input ce;
input clk;
input reset;

wire gw_vcc;
wire gw_gnd;

assign gw_vcc = 1'b1;
assign gw_gnd = 1'b0;

ALU54D alu54d_inst (
    .DOUT(dout),
    .CASO(caso),
    .A(a),
    .B(b),
    .ASIGN(gw_vcc),
    .BSIGN(gw_vcc),
    .CASI({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd},
    .ACCLOAD(gw_gnd),
    .CE(ce),
    .CLK(clk),
    .RESET(reset)
);

defparam alu54d_inst.AREG = 1'b1;
defparam alu54d_inst.BREG = 1'b1;
defparam alu54d_inst.ASIGN_REG = 1'b0;
defparam alu54d_inst.BSIGN_REG = 1'b0;
defparam alu54d_inst.ACCLOAD_REG = 1'b0;
defparam alu54d_inst.OUT_REG = 1'b1;
defparam alu54d_inst.B_ADD_SUB = 1'b0;
defparam alu54d_inst.C_ADD_SUB = 1'b0;
defparam alu54d_inst.ALUD_MODE = 0;
defparam alu54d_inst.ALU_RESET_MODE = "SYNC";

endmodule //GW_ALU54

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 ALU54 设计文件的同时，还提供用户例化该 IP 设计文件的模板文件，如图 3-43 所示。

图 3-43 用户例化该 IP 设计文件的模板文件

```

GW_ALU54 your_instance_name(
    .dout(dout_o), //output [53:0] dout
    .caso(caso_o), //output [54:0] caso
    .a(a_i), //input [53:0] a
    .b(b_i), //input [53:0] b
    .ce(ce_i), //input ce
    .clk(clk_i), //input clk
    .reset(reset_i) //input reset
);

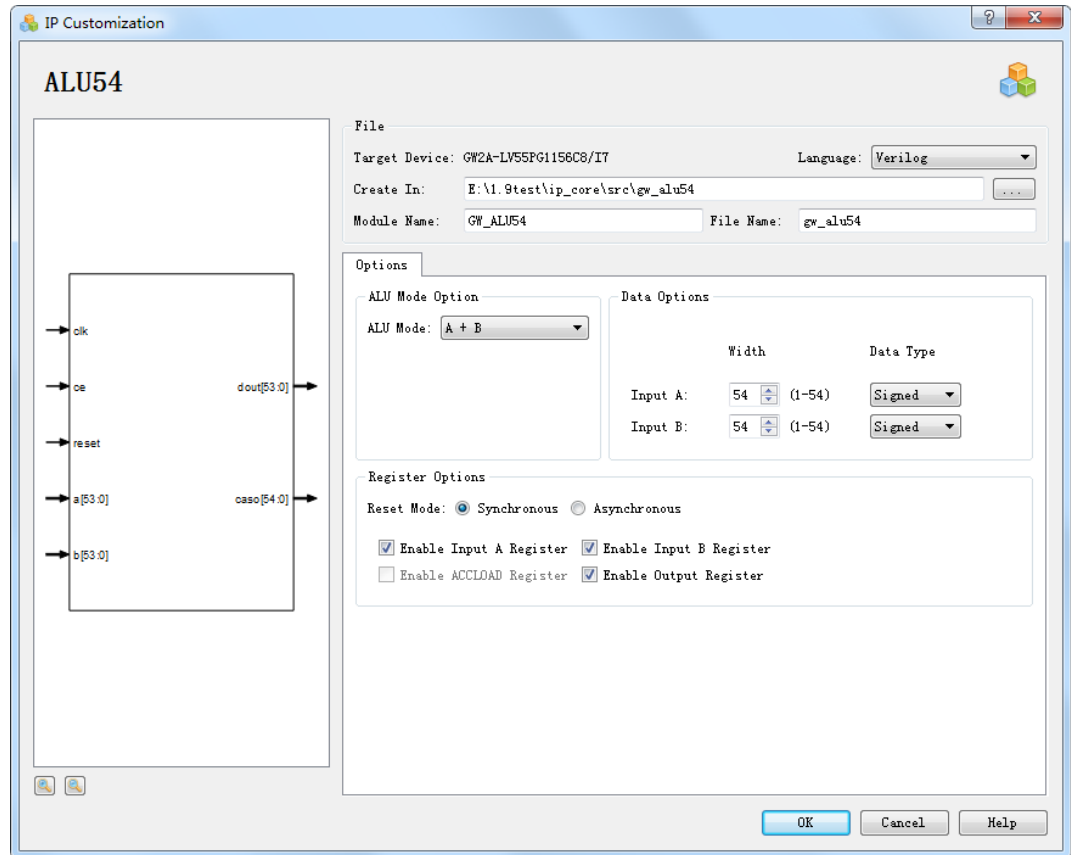
```

IP Core Generator 生成 ALU54 示例

如用户需产生 54 位与 54 位相加、有 Register 的同步 ALU54 IP，以 device 选择 GW2A-LV55PG1156C8/I7 为例，界面配置如图 3-44 所示，单击“OK”，即可产生用户所需的 ALU54 IP 设计文件。

产生的 ALU54 IP 设计文件所在目录即为配置界面中“Create In”设置路径。

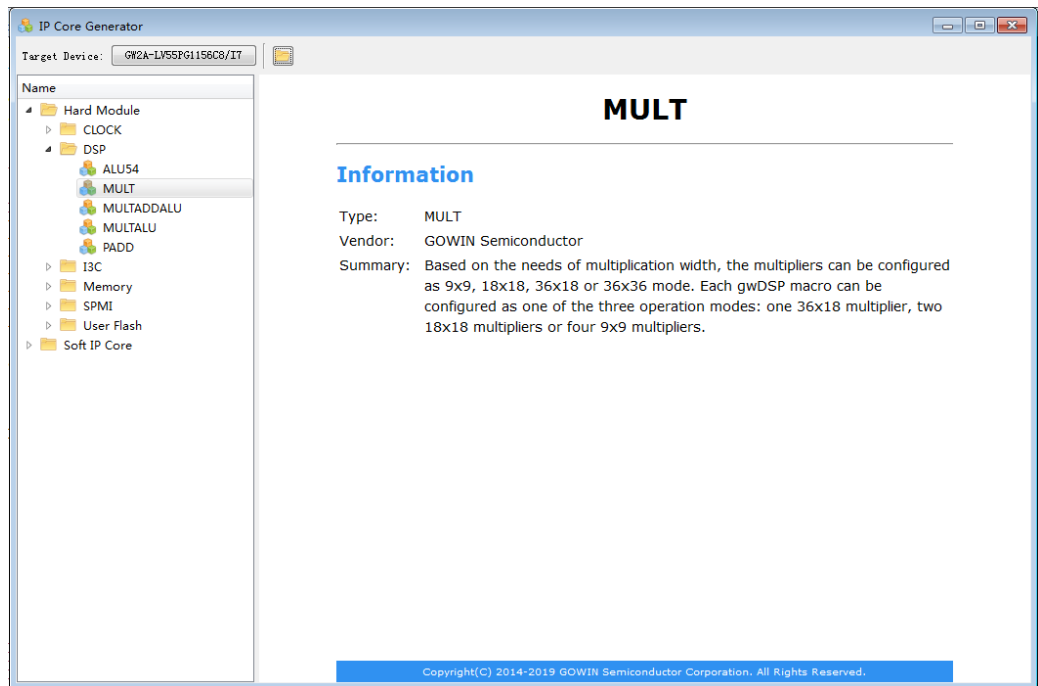
图 3-44 ALU54 IP Customization 设置



3.2.2 MULT

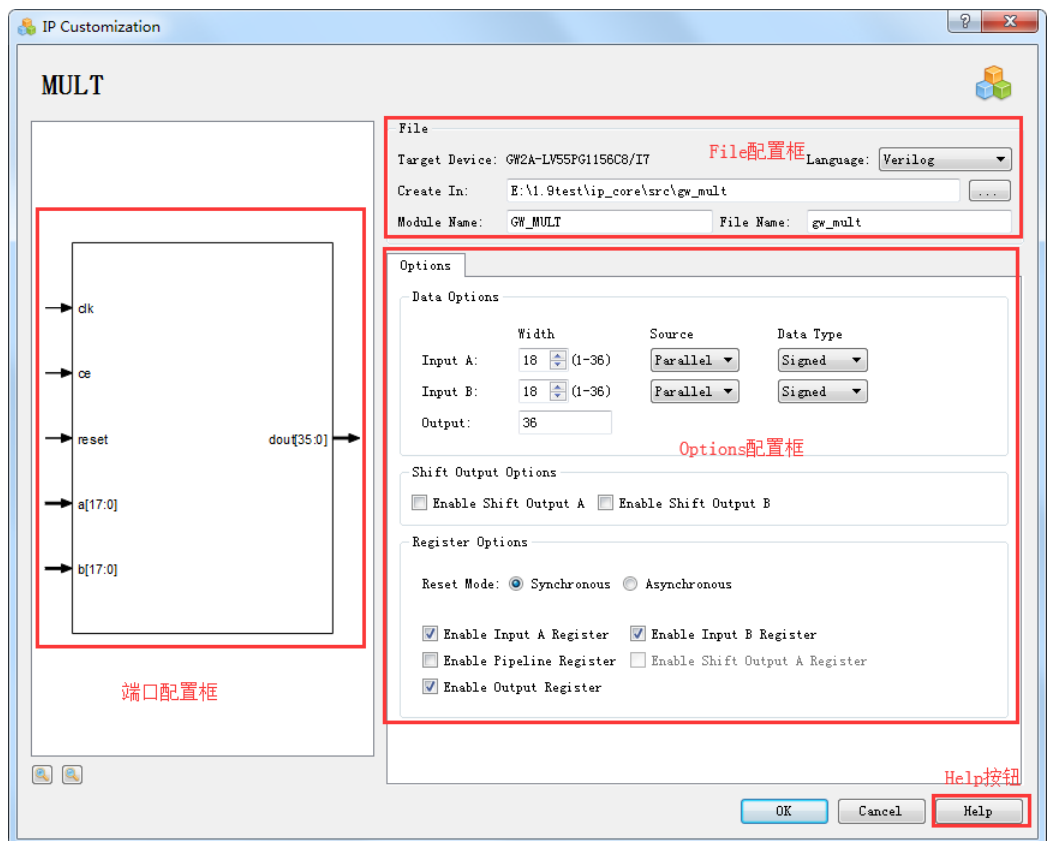
MULT 实现乘法运算功能。在 IP Core Generator 界面中单击 MULT，界面右侧会显示 MULT 的相关信息概要，如图 3-45 所示。

图 3-45 MULT 的信息概要



在 IP Core Generator 界面中双击 MULT，弹出 MULT 的“IP Customization”窗口，如图 3-46 所示。该窗口包括 File 配置框、Options 配置框、端口配置框图以及帮助按钮“Help”。

图 3-46 MULT 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 MULT 实例化文件的相关信息,如图 3-46 中标注的“File 配置框”所示。

MULT 的 File 配置框的使用和 SP 模块的类似,请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 MULT 设计文件中 MULT 的配置信息,如图 3-46 标注的“Options 配置框”所示。

- **Data Options:** 配置数据选项。
 - 输入端口 (Input A Width/ Input B Width) 最大数据位宽为 36;
 - 输出端口数据位宽 (Output Width) 无需用户配置,它会根据输入位宽自动调整位宽,例化时会根据位宽生成 MULT9X9, MULT18X18, MULT36X36。
 - 输入端口 A/B 可配置为 Parallel、Shift;
 - 数据类型可配置为 Unsigned、Signed。
- **Shift Output Options:** 能否使能 shift out 功能,输入端口 (Input A Width/ Input B Width) 都小于等于 18 时,可使用此功能。

注!

输入端口 (Input A Width/ Input B Width) 任一项大于 18 时, Shift Output Options 置灰,不可使用。

- **Register Options:** 该选项的功能、用法与 ALU54 的 Register Options 选项相同,请参考 [3.2.1ALU54](#) 中的 Option 配置框。

3. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图,输入输出端口的个数以及位宽根据 Options 配置实时更新,如图 3-46 中标注的“配置框图”所示。

4. Help 按钮

单击“Help”,显示 IP Core 的配置信息的页面,如图 3-47 所示。Help 页面包括当前 IP Core 的概要介绍,以及 Options 各项配置的简要说明。

图 3-47 Help 信息

MULT	
Information	
Type:	MULT
Vendor:	GOWIN Semiconductor
Summary:	Based on the needs of multiplication width, the multipliers can be configured as 9x9, 18x18, 36x18 or 36x36 mode. Each gwDSP macro can be configured as one of the three operation modes: one 36x18 multiplier, two 18x18 multipliers or four 9x9 multipliers.
Options	
Option	Description
Data Options	Input A Width - Set the size of the first item in the multiplication.
	Input B Width - Set the size of the second item in the multiplication.
	Output Width - Size of the output. The output size is the sum of the input A and input B bit sizes.
	Source - Set the source of the input A/B as Parallel or Shift.
Shift Output Options	Data Type - Set the data format of the inputs as signed or unsigned.
	Enable Shift Output A - Enable or disable the shift out port A of the multiplication.
	Enable Shift Output B - Enable or disable the shift out port B of the multiplication.
Register Options	Note: If either of the A and B inputs is greater than 18 bits, the input and output shift options are not available.
	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
	Enable ... Register - Enable or disable registers. For example, if you choose Enable Input A Register, the input data will go through one register.

IP 生成文件

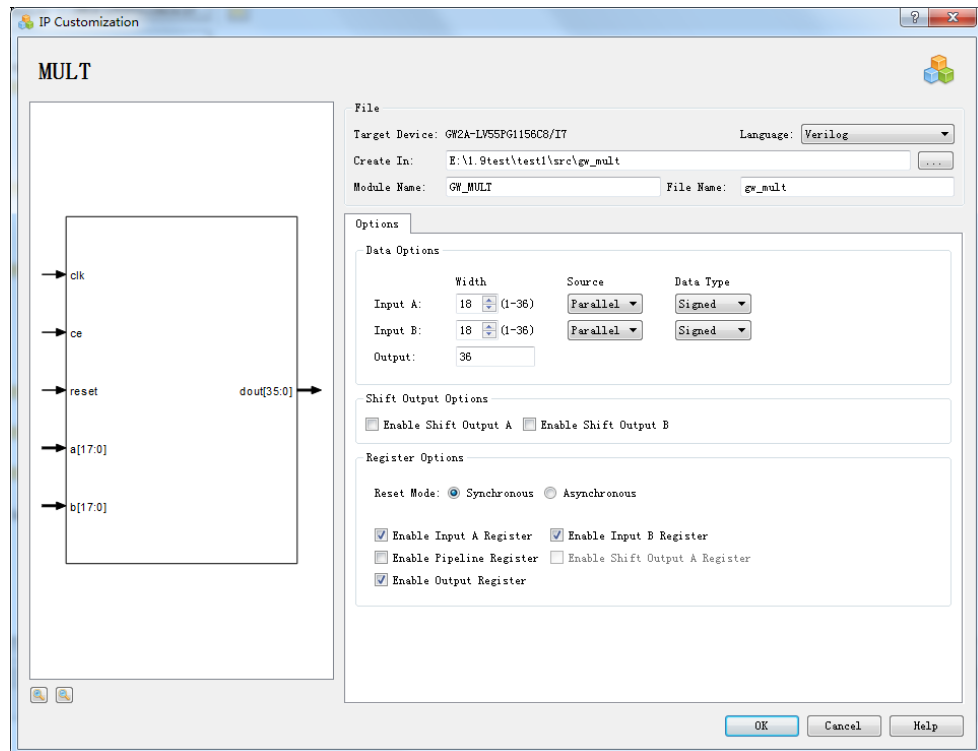
如图 3-48 所示，MULT 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 MULT 设计文件“gw_mult.v”；
- 用户例化该 IP 设计文件的模板文件“gw_mult_tmp.v”；
- 例化原语 MULT 的配置文件“gw_mult.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

配置 MULT 的“IP Customization”窗口，如图 3-48 所示。

图 3-48 配置的 IP Customization



例化 MULT 设计文件

例化 MULT 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 MULT 配置，产生了实例化的 MULT，如图 3-49 所示。

图 3-49 例化 MULT 设计文件

```

module GW_MULT (dout, a, b, ce, clk, reset);
    output [35:0] dout;
    input [17:0] a;
    input [17:0] b;
    input ce;
    input clk;
    input reset;

    wire [17:0] soa_w;
    wire [17:0] sob_w;
    wire gw_vcc;
    wire gw_gnd;

    assign gw_vcc = 'b1;
    assign gw_gnd = 'b0;

    MULT18X18 mult18x18_inst (
        .DOUT(dout),
        .SOA(soa_w),
        .SOB(sob_w),
        .A(a),
        .B(b),
        .ASIGN(gw_vcc),
        .BSIGN(gw_vcc),
        .SIA({gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd}),
        .SIB({gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd, gw_gnd}),
        .CE(ce),
        .CLK(clk),
        .RESET(reset),
        .ASEL(gw_gnd),
        .BSEL(gw_gnd)
    );

    defparam mult18x18_inst.AREG = 'b1;
    defparam mult18x18_inst.BREG = 'b1;
    defparam mult18x18_inst.OUT_REG = 'b1;
    defparam mult18x18_inst.PIPE_REG = 'b0;
    defparam mult18x18_inst.ASIGN_REG = 'b0;
    defparam mult18x18_inst.BSIGN_REG = 'b0;
    defparam mult18x18_inst.SOA_REG = 'b0;
    defparam mult18x18_inst.MULT_RESET_MODE = "SYNC";

endmodule //GW_MULT

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 MULT 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-50 所示。

图 3-50 用户例化该 IP 设计文件的模板文件

```

GW_MULT your_instance_name (
    .dout(dout_o), //output [35:0] dout
    .a(a_i), //input [17:0] a
    .b(b_i), //input [17:0] b
    .ce(ce_i), //input ce
    .clk(clk_i), //input clk
    .reset(reset_i) //input reset
);

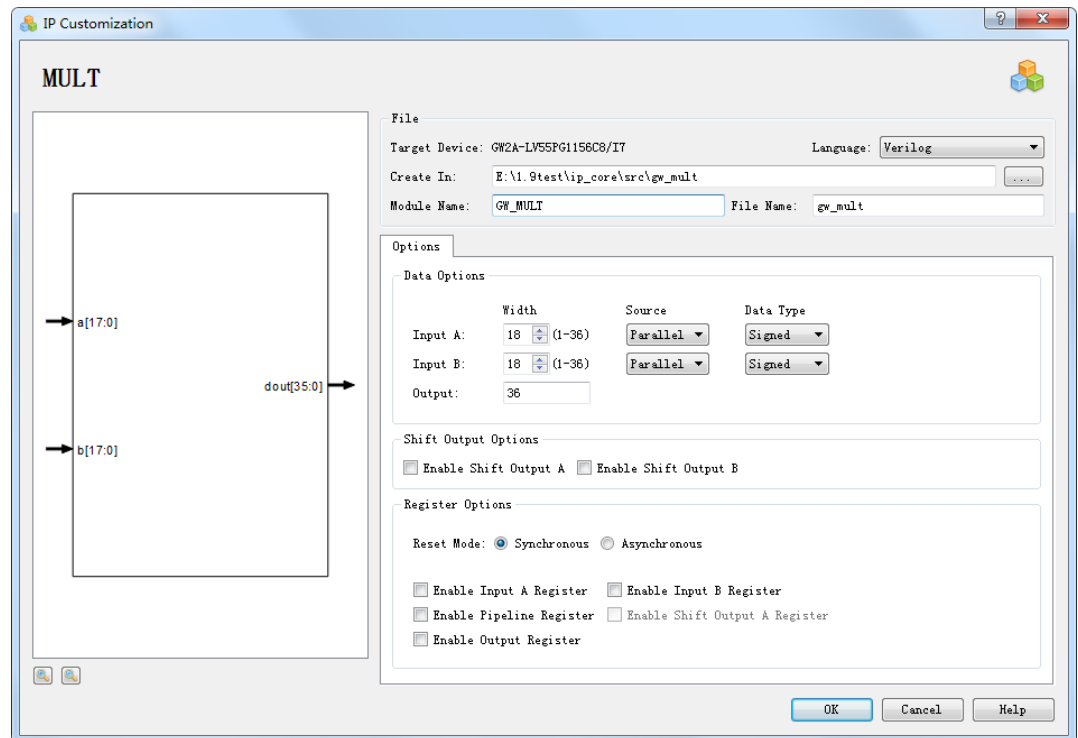
```

IP Core Generator 生成 MULT 示例

如用户需产生 9 位有符号数乘法、Bypass 模式的 MULT IP，以 device 选择 GW2A-LV55PG1156C8/I7 为例，界面配置如图 3-51 所示，单击“OK”，产生用户所需的 MULT IP 设计文件。

产生的 MULT IP 设计文件所在目录即为配置界面中“Create In”设置路径。

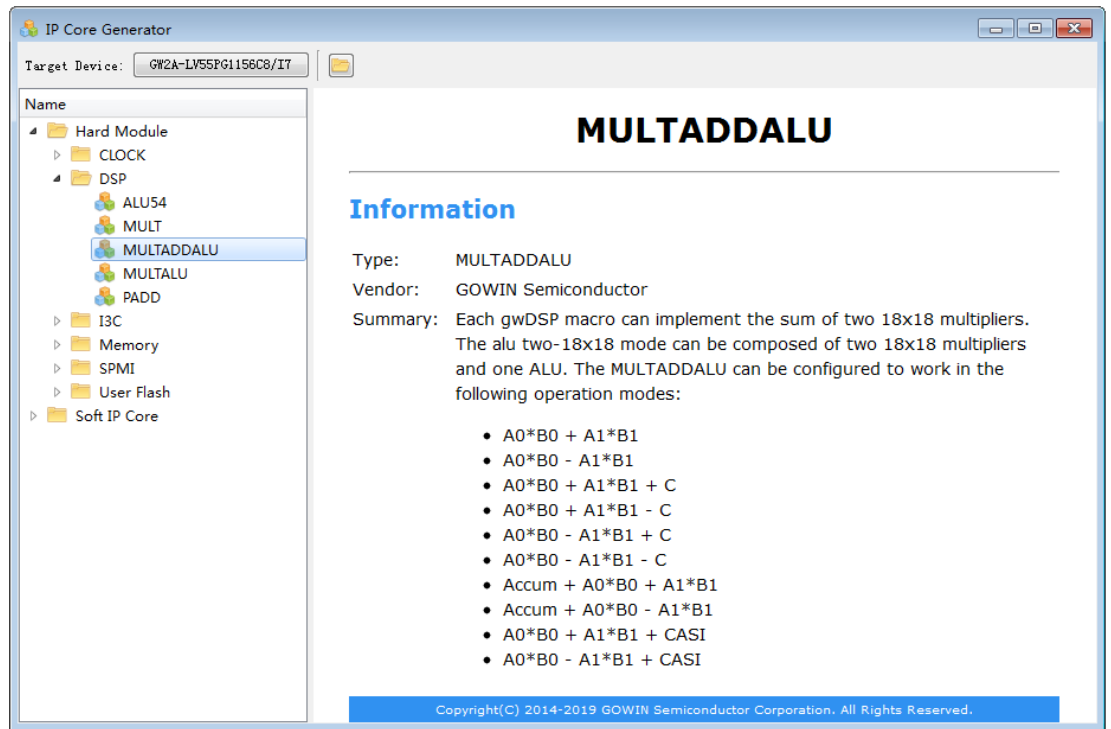
图 3-51 MULT IP Customization 设置



3.2.3 MULTADDALU

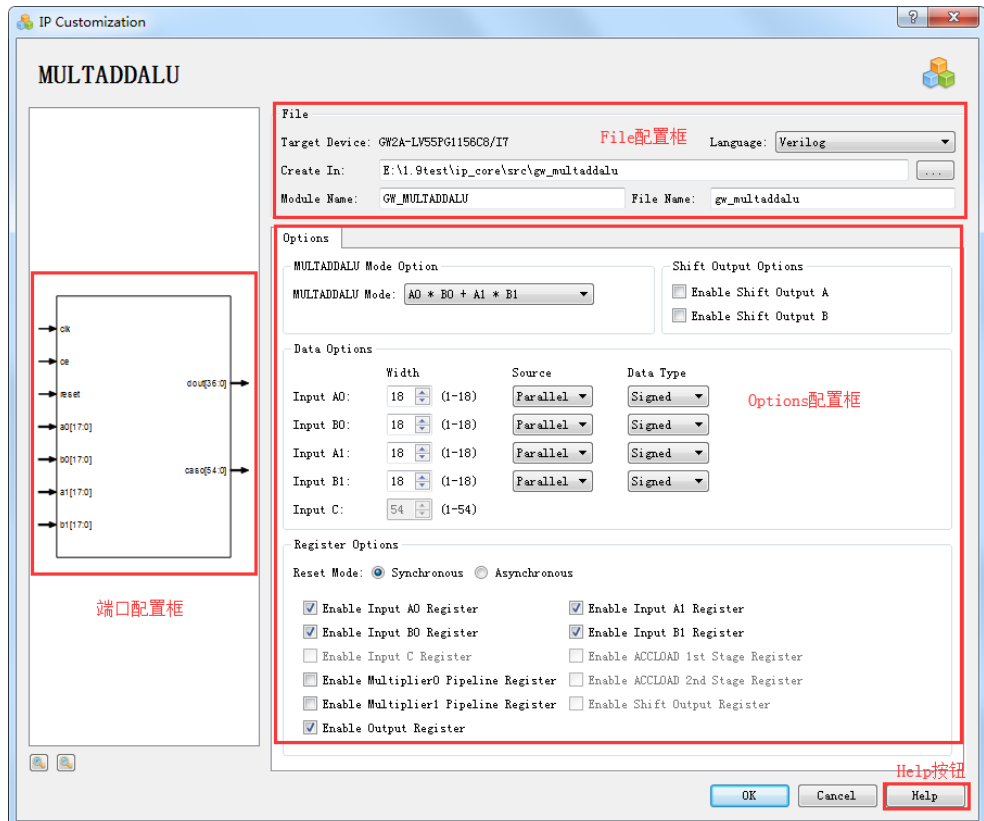
MULTADDALU 实现乘法器二次求和或累加功能。在 IP Core Generator 界面中，单击“MULTADDALU”，界面右侧会显示 MULTADDALU 的相关信息概要，如图 3-52 所示。

图 3-52 MULTADDALU 的信息概要



在 IP Core Generator 界面中，双击“MULTADDALU”，弹出 MULTADDALU 的“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-53 所示。

图 3-53 MULTADDALU 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 MULTADDALU 实例化文件的相关信息，标注的“File 配置框”如图 3-53 所示。

MULTADDALU 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 MULTADDALU 设计文件中 MULTADDALU 的配置信息，标注的“Options 配置框”如图 3-53 所示。

- MULTADDALU Mode Option: 配置 MULTADDALU 的运算模式。可选择：
 - $A0*B0 + A1*B1$
 - $A0*B0 - A1*B1$
 - $A0*B0 + A1*B1 + C$
 - $A0*B0 + A1*B1 - C$
 - $A0*B0 - A1*B1 + C$
 - $A0*B0 - A1*B1 - C$
 - $Accum + A0*B0 + A1*B1$
 - $Accum + A0*B0 - A1*B1$
 - $A0*B0 + A1*B1 + CASI$
 - $A0*B0 - A1*B1 + CASI;$
- MULTADDALU 的 Data Options 和 Register Options 配置框的使用和 MULT 模块的类似，请参考 [3.2.2MULT](#)。

3. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Data Options 和 Register Options 配置实时更新，如图 3-53 中标注的“配置框图”所示。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-54 所示。

图 3-54 Help 信息

MULTADDALU	
Information	
Type:	MULTADDALU
Vendor:	GOWIN Semiconductor
Summary:	<p>Each gwDSP macro can implement the sum of two 18x18 multipliers. The alu two-18x18 mode can be composed of two 18x18 multipliers and one ALU. The MULTADDALU can be configured to work in the following operation modes:</p> <ul style="list-style-type: none"> • $A0*B0 + A1*B1$ • $A0*B0 - A1*B1$ • $A0*B0 + A1*B1 + C$ • $A0*B0 + A1*B1 - C$ • $A0*B0 - A1*B1 + C$ • $A0*B0 - A1*B1 - C$ • Accum + $A0*B0 + A1*B1$ • Accum + $A0*B0 - A1*B1$ • $A0*B0 + A1*B1 + CASI$ • $A0*B0 - A1*B1 + CASI$
Options	
Option	Description
MULTADDALU Mode Option	MULTADDALU Mode - Set one of the MULTADDALU operation modes.
Shift Output Options	Enable Shift Output A - Enable or disable the shift out port A of the DSP.
	Enable Shift Output B - Enable or disable the shift out port B of the DSP.
Data Options	Input A0 Width - Set the size of the first item in the first multiplication.
	Input B0 Width - Set the size of the second item in the first multiplication.
	Input A1 Width - Set the size of the first item in the second multiplication.
	Input B1 width - Set the size of the second item in the second multiplication.
	Input C width - Set the size of input C.
Register Options	Source - Set the source of the input A0/B0/A1/B1 as Parallel or Shift.
	Data Type - Set the data format of the input A0/B0/A1/B1 as signed or unsigned.
	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
	Enable ... Register - Enable or disable registers. For example, if you choose Enable Input A0 Register, the input data will go through one register.

Help 页面包括 IP Core 的概要介绍，以及 Data Options 和 Register Options 各项配置的简要说明。

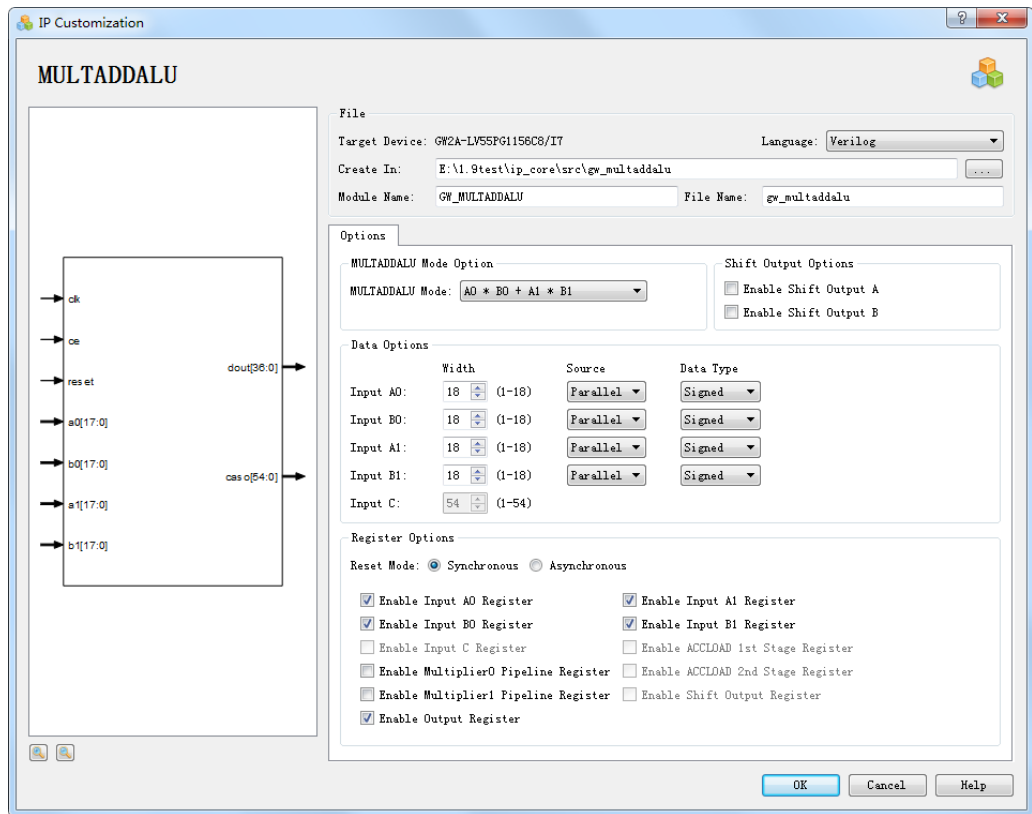
IP 生成文件

如图 3-55 所示，MULTADDALU 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 SP 设计文件“gw_multaddalu.v”；
- 用户例化该 IP 设计文件的模板文件“gw_multaddalu_tmp.v”；
- 例化原语 SP 的配置文件“gw_multaddalu.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-55 配置的 IP Customization



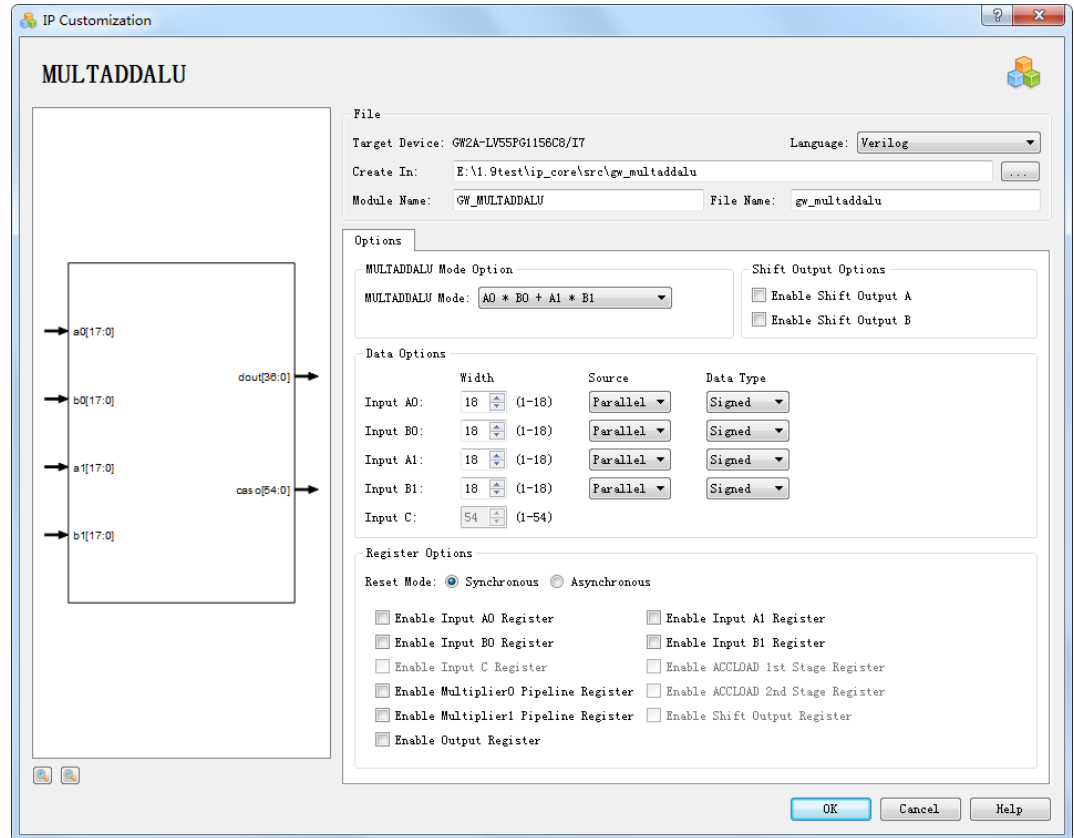
例化 MULTADDALU 设计文件

例化 MULTADDALU 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 MULTADDALU 配置，产生了实例化的 MULTADDALU，如图 3-56 所示。

配置如图 3-58 所示，单击“OK”，即可产生用户所需的 MULTADDALU IP 设计文件。

产生的 MULTADDALU IP 设计文件所在目录即为配置界面中“Create In”设置路径。

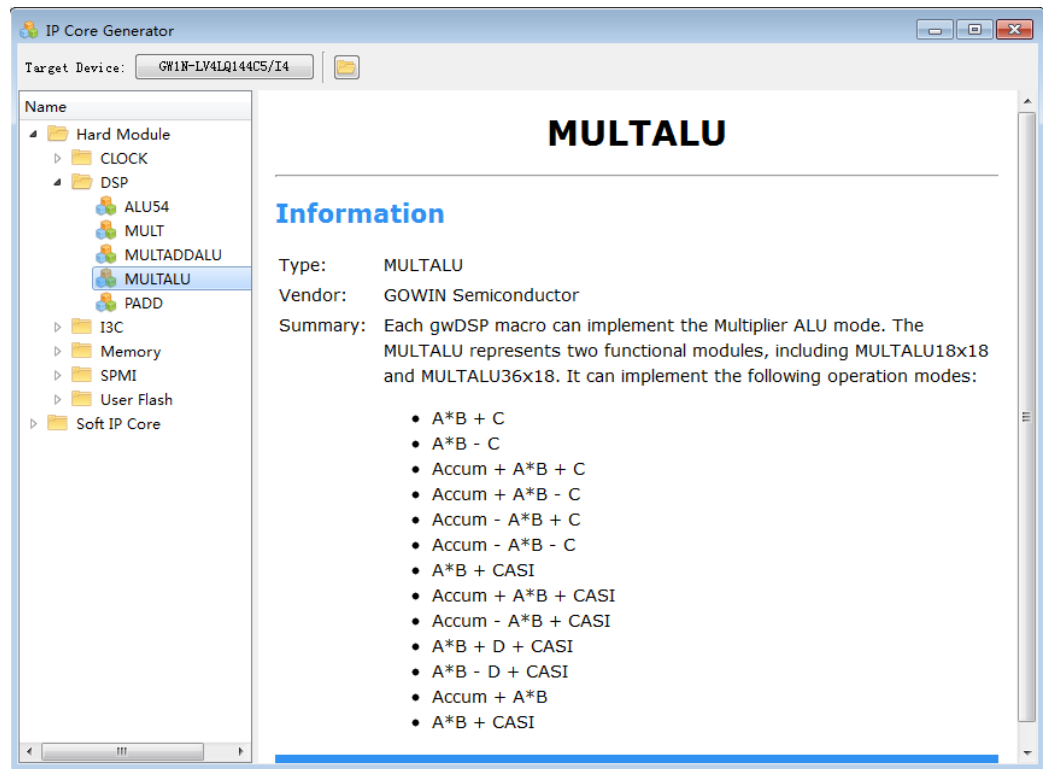
图 3-58 MULTADDALU IP Customization 设置



3.2.4 MULTALU

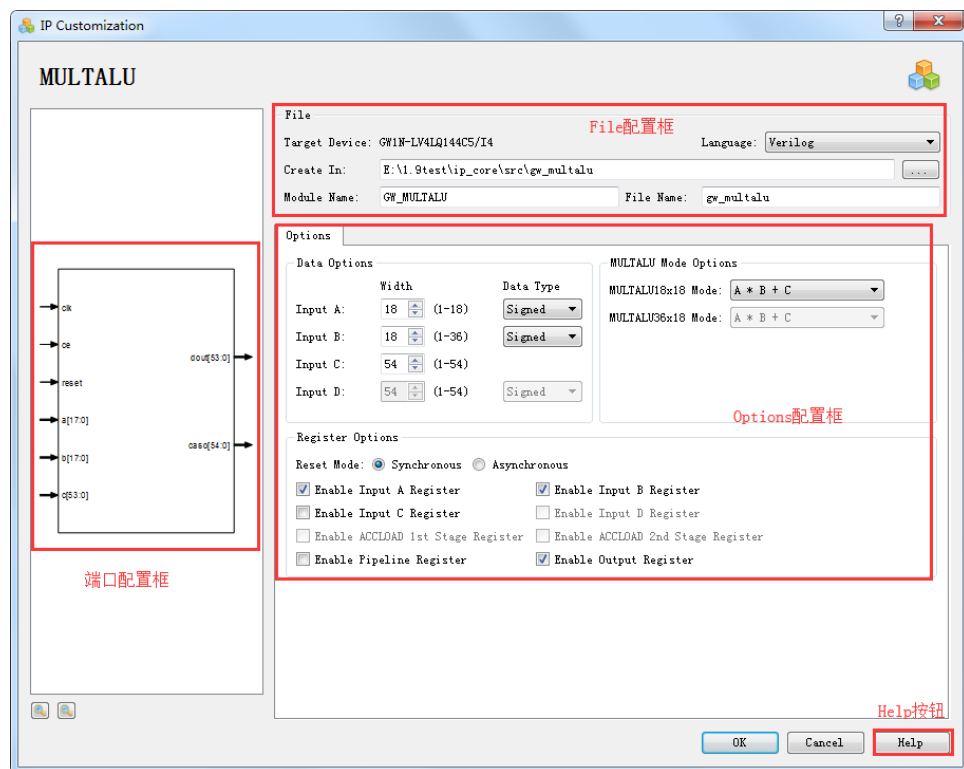
MULTALU 实现乘积求和或累加的功能。在 IP Core Generator 界面中，单击 MULTALU，界面右侧会显示 MULTALU 的相关信息概要，如图 3-59 所示。

图 3-59 MULTALU 的信息概要



在 IP Core Generator 界面中，双击“MULTALU”，弹出“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-60 所示。

图 3-60 MULTALU 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 MULTALU 实例化文件的相关信息，如图 3-60 中标注的“File 配置框”所示。

MULTALU 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 Option 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 MULTALU 设计文件中 MULTALU 的配置信息，标注的“Options 配置框”如图 3-60 所示。

● MULTALU Mode Option:

IP Core 中的 MULTALU 根据输入端口的位宽可以生成两种模块：MULTALU36X18 或 MULTALU18X18。当 Input A 和 Input B 的 width 都小于或等于 18 位时，Options 配置框右侧的 MULTALU Mode Options 中 MULTALU36X18 Mode 置灰，MULTALU18X18 Mode 可以配置为：

- A*B + C
- A*B - C
- Accum + A*B + C
- Accum + A*B - C
- Accum - A*B + C
- Accum - A*B - C
- A*B + CASI
- Accum + A*B + CASI
- Accum - A*B + CASI
- A*B + D + CASI
- A*B - D + CASI

- 当 Input B 的 width 大于 18 位时，MULTALU18X18 Mode 置灰，MULTALU36X18 Mode 可以配置为：
 - $A*B + C$
 - $A*B - C$
 - $Accum + A*B$
 - $A*B + CASI$
 - MULTALU 的 Data Options 和 Register Options 配置框的使用和 MULT 模块的类似，请参考 3.2.2MULT。
3. 端口配置框图
- 配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-60 中标注的“配置框图”所示。
4. Help 按钮
- 单击“Help”，显示 IP Core 的配置信息的页面，如图 3-61 所示。

图 3-61 Help 信息

MULTALU	
Information	
Type:	MULTALU
Vendor:	GOWIN Semiconductor
Summary:	<p>Each gwDSP macro can implement the Multiplier ALU mode. The MULTALU represents two functional modules, including MULTALU18x18 and MULTALU36x18. It can implement the following operation modes:</p> <ul style="list-style-type: none"> • $A*B + C$ • $A*B - C$ • $Accum + A*B + C$ • $Accum + A*B - C$ • $Accum - A*B + C$ • $Accum - A*B - C$ • $A*B + CASI$ • $Accum + A*B + CASI$ • $Accum - A*B + CASI$ • $A*B + D + CASI$ • $A*B - D + CASI$ • $Accum + A*B$ • $A*B + CASI$
Options	
Option	Description
Data Options	Input A Width - Set the size of the first item in the multiplication.
	Input B Width - Set the size of the second item in the multiplication.
	Input C Width - Set the size of input C.
	Input D width - Set the size of input D.
	Data Type - Set the data format of the input A/B/D as signed or unsigned.
MULTALU Mode Options	MULTALU18x18 Mode - Set one of the MULTALU18X18 operation modes, the option is available only when widthB <= 18.
	MULTALU36x18 Mode - Set one of the MULTALU36X18 operation modes, the option is available only when widthB > 18.
Register Options	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
	Enable ... Register - Enable or disable registers. For example, if you choose Enable Input A Register, the input data will go through one register.

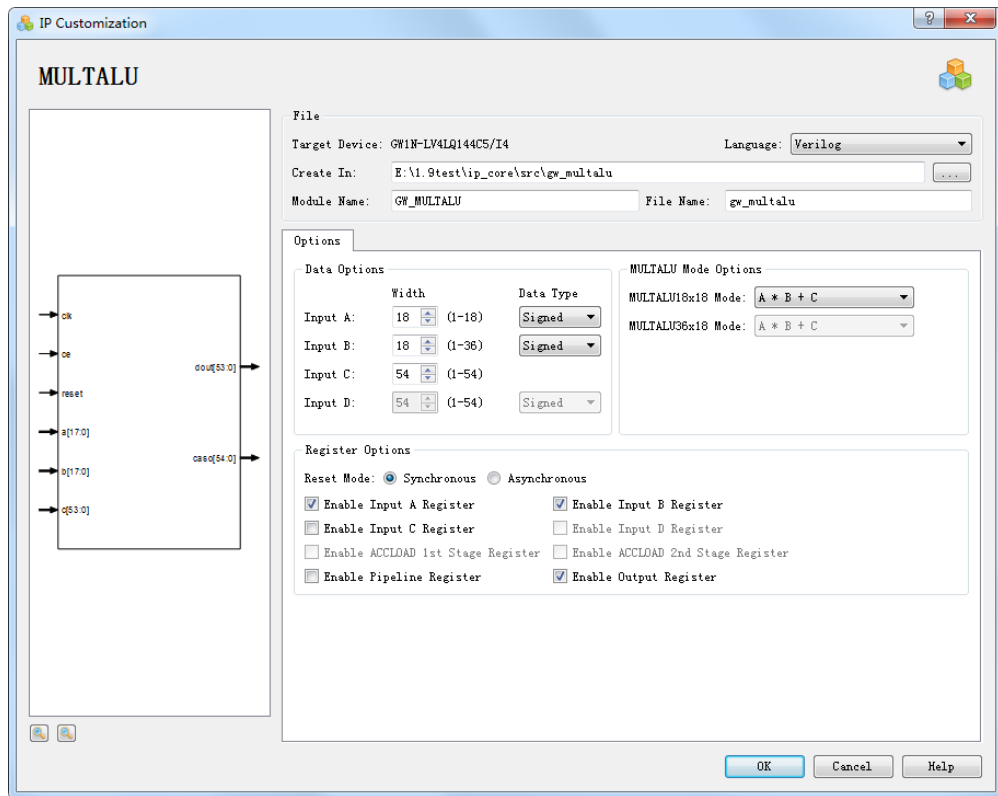
IP 生成文件

如图 3-62 所示，MULTALU 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 MULTALU 设计文件“gw_multalu.v”；
- 用户例化该 IP 设计文件的模板文件“gw_multalu_tmp.v”；
- 例化原语 MULTALU 的配置文件“gw_multalu.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-62 配置的 IP Customization



例化 MULTALU 设计文件

例化 MULTALU 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 MULTALU 配置，产生实例化的 MULTALU，如图 3-63 所示。

图 3-63 例化 MULTALU 设计文件

```

module GW_MULTALU (dout, caso, a, b, c, ce, clk, reset);

output [53:0] dout;
output [54:0] caso;
input [17:0] a;
input [17:0] b;
input [53:0] c;
input ce;
input clk;
input reset;

wire gw_vcc;
wire gw_gnd;

assign gw_vcc = 1'b1;
assign gw_gnd = 1'b0;

MULTALU18X18 multalu18x18_inst (
    .DOUT(dout),
    .CASO(caso),
    .A(a),
    .B(b),
    .C(c),
    .D({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd},
    .ASIGN(gw_vcc),
    .BSIGN(gw_vcc),
    .DSIGN(gw_vcc),
    .CASI({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd},
    .ACCLOAD(gw_gnd),
    .CE(ce),
    .CLK(clk),
    .RESET(reset)
);

defparam multalu18x18_inst.AREG = 1'b1;
defparam multalu18x18_inst.BREG = 1'b1;
defparam multalu18x18_inst.CREG = 1'b0;
defparam multalu18x18_inst.DREG = 1'b0;
defparam multalu18x18_inst.OUT_REG = 1'b1;
defparam multalu18x18_inst.PIPE_REG = 1'b0;
defparam multalu18x18_inst.ASIGN_REG = 1'b0;
defparam multalu18x18_inst.BSIGN_REG = 1'b0;
defparam multalu18x18_inst.DSIGN_REG = 1'b0;
defparam multalu18x18_inst.ACCLD_LOAD_REG0 = 1'b0;
defparam multalu18x18_inst.ACCLD_LOAD_REG1 = 1'b0;
defparam multalu18x18_inst.B_ADD_SUB = 1'b0;
defparam multalu18x18_inst.C_ADD_SUB = 1'b0;
defparam multalu18x18_inst.MULTALU18X18_MODE = 0;
defparam multalu18x18_inst.MULT_RESET_MODE = "SYNC";

endmodule //GW_MULTALU

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 MULTALU 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-64 所示。

图 3-64 用户例化该 IP 设计文件的模板文件

```

GW_MULTALU your_instance_name (
    .dout(dout_o), //output [53:0] dout
    .caso(caso_o), //output [54:0] caso
    .a(a_i), //input [17:0] a
    .b(b_i), //input [17:0] b
    .c(c_i), //input [53:0] c
    .ce(ce_i), //input ce
    .clk(clk_i), //input clk
    .reset(reset_i) //input reset
);

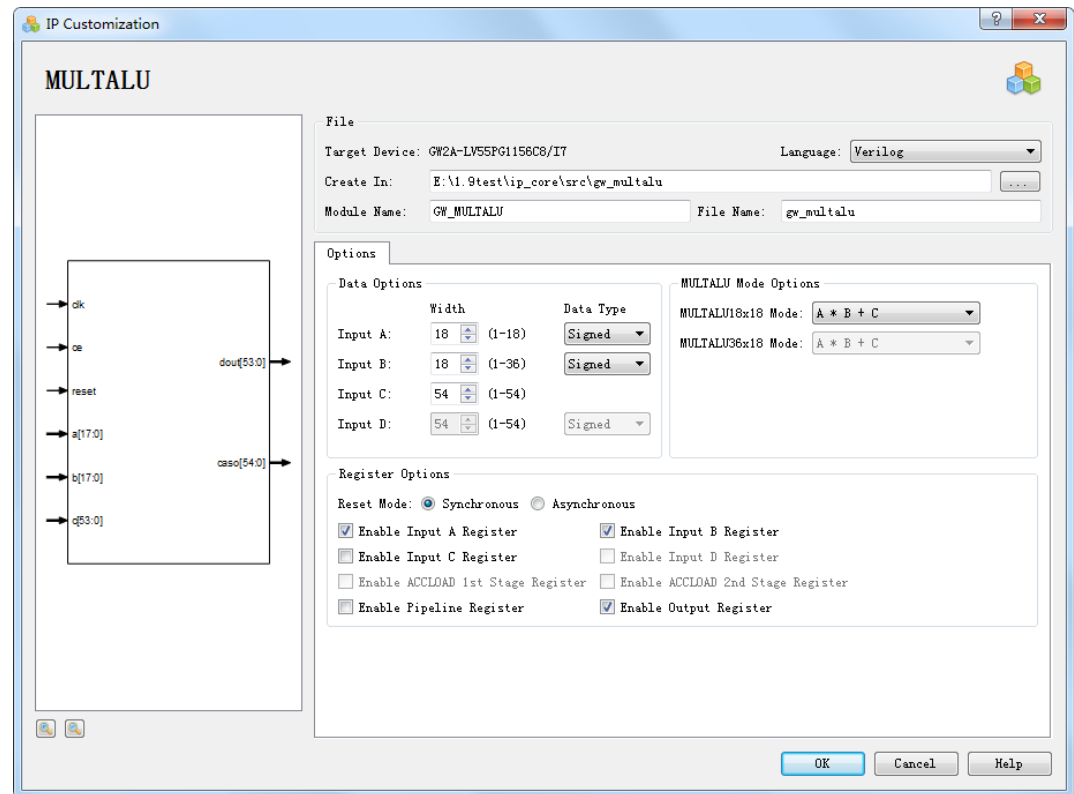
```


IP Core Generator 生成 MULTALU 示例

如用户需产生 18 位有符号数乘法器求和，Register 模式的同步 MULTALU IP，以 device 选择 GW2A-LV55PG1156C8/I7 为例，界面配置如图 3-65 所示，单击“OK”，产生用户所需的 MULTALU IP 设计文件。

产生的 MULTALU IP 设计文件所在目录即为配置界面中“Create In”设置路径。

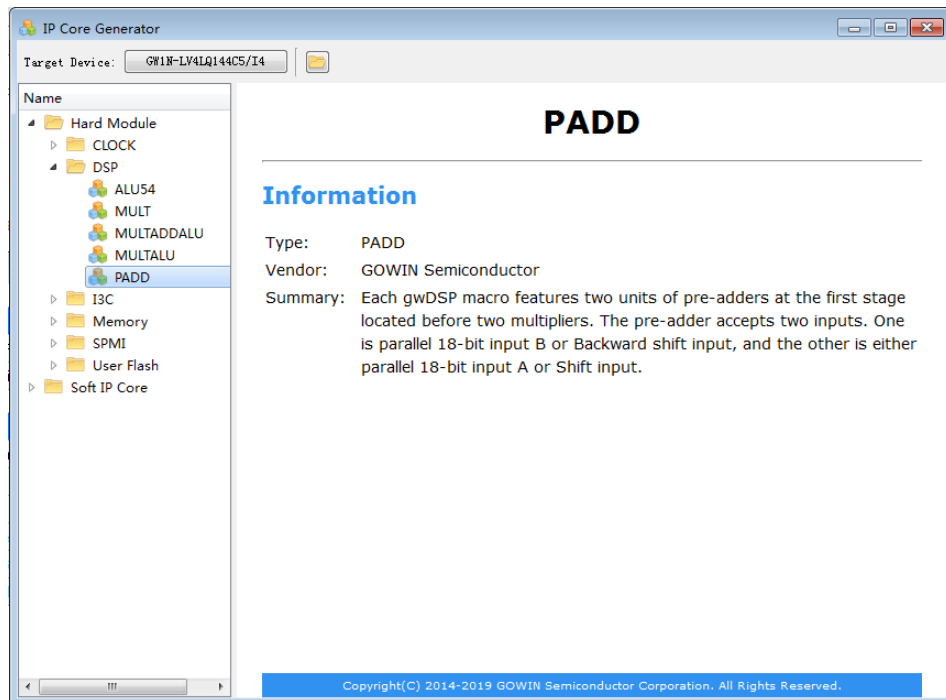
图 3-65 MULTALU IP Customization 设置



3.2.5 PADD

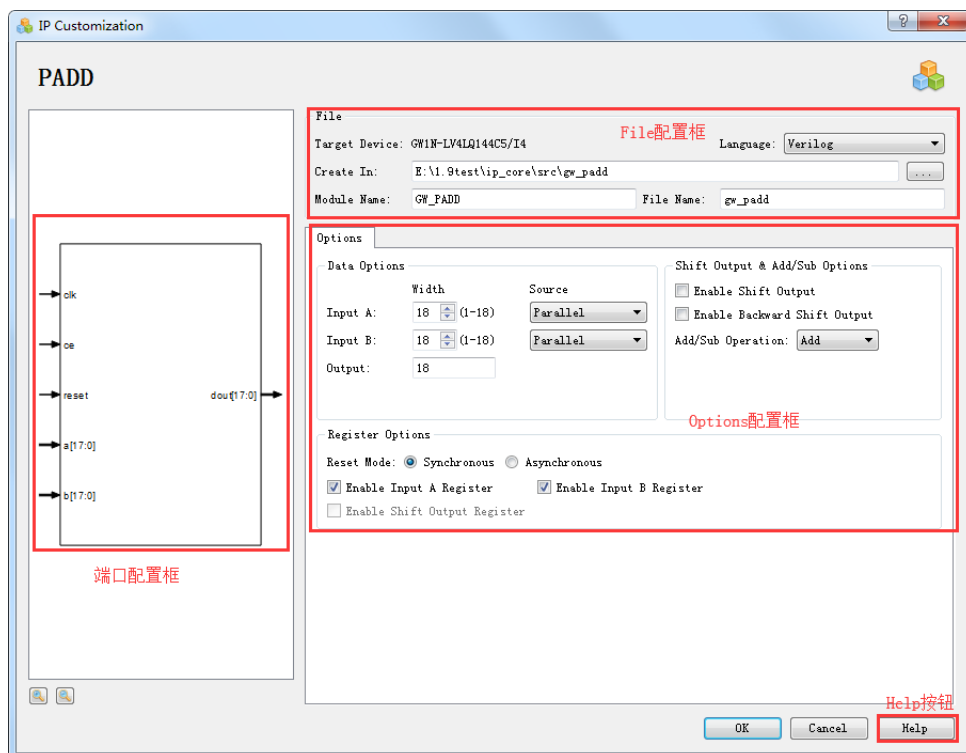
PADD 实现预加、预减或移位功能。在 IP Core Generator 界面中，单击“PADD”，界面右侧即会显示 PADD 的相关信息概要，如图 3-66 所示。

图 3-66 PADD 信息概要



在 IP Core Generator 界面中, 双击“PADD”, 弹出 “IP Customization” 窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮 “Help”, 如图 3-67 所示。

图 3-67 PADD 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生的 PADD 实例化文件的相关信息，如图 3-67 中标注的“File 配置框”所示。

PADD 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 PADD 设计文件中 PADD 的配置信息，标注的“Options 配置框”如图 3-67 所示。

- **Data Options:** 配置数据选项。
 - 输入端口 (Input A Width/ Input B Width) 最大数据位宽为 18;
 - 输出端口数据位宽 (output Width) 无需用户配置，它会根据输入位宽自动调整位宽，例化时会根据位宽决定生成 PADD9 或 PADD18。
 - 输入端口 A 的数据来源可通过“Input A Source”选项配置为 Parallel 和 Shift;
 - 输入端口 B 的数据来源可通过“Input B Source”选项配置为 Parallel、Backward Shift。
- **Shift Output & Add/Sub Options:** 使能 Shift Output、Backward Shift Output 和加减操作配置。
 - 使能 Shift Output 通过选中“Enable Shift Output”进行配置;
 - 使能 Backward Shift Output 通过选中“Enable Backward Shift Output”选项进行配置;
 - PADD 可通过配置“Add/Sub Operation”选项执行加法、减法。
- **Register Options:** 配置寄存器工作模式。
 - “Reset Mode”选项配置 PADD 的复位模式，支持同步模式“Synchronous”和异步模式“Asynchronous”;
 - “Enable Input A Register”配置 Input A register;
 - “Enable Input B Register”配置 Input B register;
 - “Enable Output Register”配置 Output register。

3. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的个数以及位宽根据 Options 配置实时更新，标注的“配置框图”如图 3-67 所示。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-68 所示。Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

图 3-68 Help 信息

PADD	
Information	
Type:	PADD
Vendor:	GOWIN Semiconductor
Summary:	Each gwDSP macro features two units of pre-adders at the first stage located before two multipliers. The pre-adder accepts two inputs. One is parallel 18-bit input B or Backward shift input, and the other is either parallel 18-bit input A or Shift input.
Options	
Option	Description
Data Options	Input A Width - Set the size of the first item in the Pre-adder.
	Input B Width - Set the size of the second item in the Pre-adder.
	Output Width - Size of the output.
	Input A Source - Set the source of the input A as Parallel or Shift.
Shift Output & Add/Sub Options	Input B Source - Set the source of the input B as Parallel or Backward Shift.
	Enable Shift Output - Enable or disable the shift out port of the Pre-adder.
	Enable Backward Shift Output - Enable or disable the backward shift out port of the Pre-adder.
Register Options	Add/Sub Operation - Set whether the mode is in add or subtract mode.
	Reset Mode - Set whether the reset mode is synchronous or asynchronous.
	Enable ... Register - Enable or disable registers. For example, if you choose Enable Input A Register, the input data will go through one register.

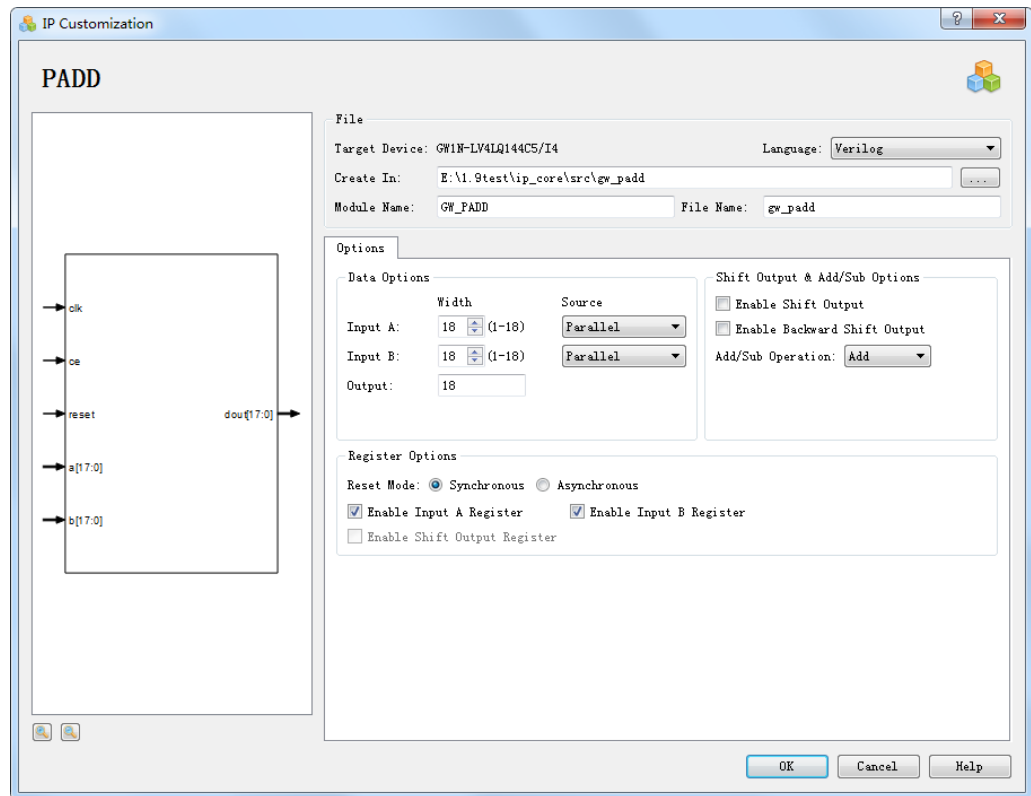
IP 生成文件

如图 3-69 所示，PADD 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 PADD 设计文件“gw_padd.v”；
- 用户例化该 IP 设计文件的模板文件“gw_padd_tmp.v”；
- 例化原语 PADD 的配置文件“gw_padd.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-69 配置的 IP Customization



例化 PADD 设计文件

例化 PADD 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 PADD 配置，产生实例化的 PADD，如图 3-70 所示。

图 3-70 例化 PADD 设计文件

```

module GW_PADD (dout, a, b, ce, clk, reset);

output [17:0] dout;
input [17:0] a;
input [17:0] b;
input ce;
input clk;
input reset;

wire [17:0] so_w;
wire [17:0] sbo_w;
wire gw_gnd;

assign gw_gnd = 1'b0;

PADD18 padd18_inst (
    .DOUT(dout),
    .SO(so_w),
    .SBO(sbo_w),
    .A(a),
    .B(b),
    .SI({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,
        gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .SBI({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,
        gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .CE(ce),
    .CLK(clk),
    .RESET(reset),
    .ASEL(gw_gnd)
);

defparam padd18_inst.AREG = 1'b1;
defparam padd18_inst.BREG = 1'b1;
defparam padd18_inst.ADD_SUB = 1'b0;
defparam padd18_inst.PADD_RESET_MODE = "SYNC";
defparam padd18_inst.BSEL_MODE = 1'b0;
defparam padd18_inst.SOREG = 1'b0;

endmodule //GW_PADD

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 PADD 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-71 所示。

图 3-71 用户例化该 IP 设计文件的模板文件

```

GW_PADD your_instance_name(
    .dout(dout_o), //output [17:0] dout
    .a(a_i), //input [17:0] a
    .b(b_i), //input [17:0] b
    .ce(ce_i), //input ce
    .clk(clk_i), //input clk
    .reset(reset_i) //input reset
);

```

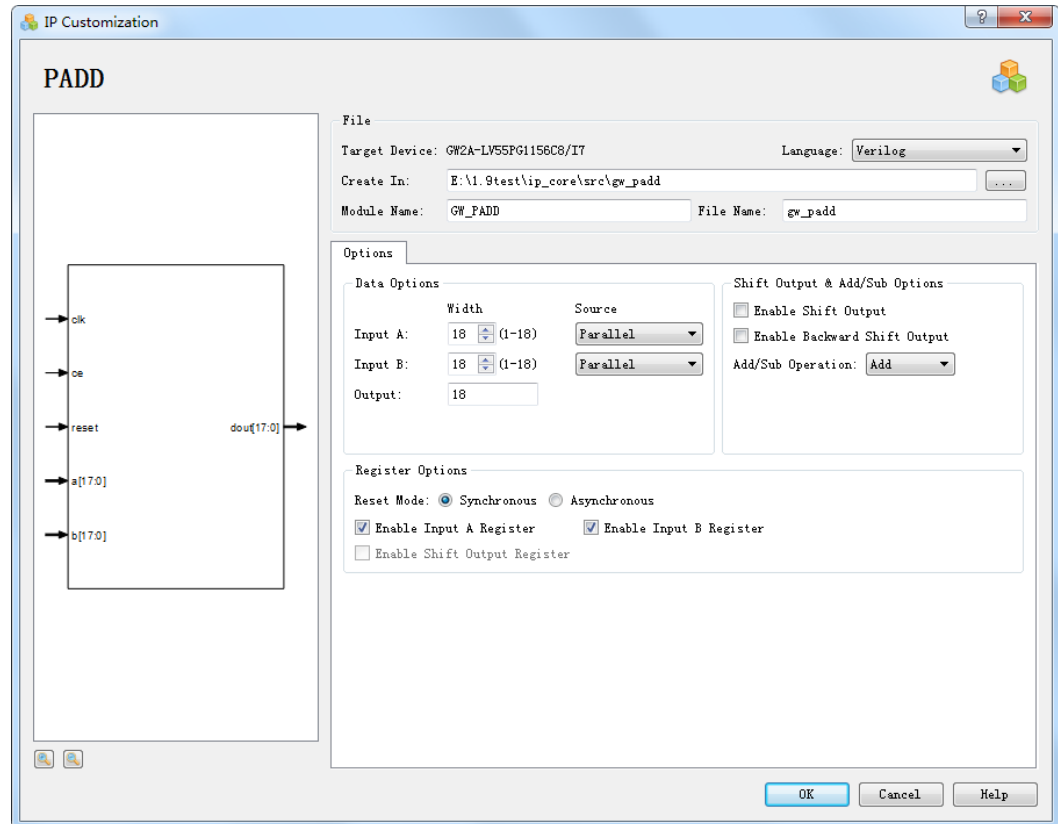
IP Core Generator 生成 PADD 示例

如用户需产生 18 位加法的同步 Register 模式 PADD IP，以 device 选择 GW2A-LV55PG1156C8/I7 为例，界面配置如图 3-72 所示，单击“OK”，产生用户所需的 PADD IP 设计文件。

产生的 PADD IP 设计文件所在目录即为配置界面中“Create In”设置路

径。

图 3-72 PADD IP Customization 设置



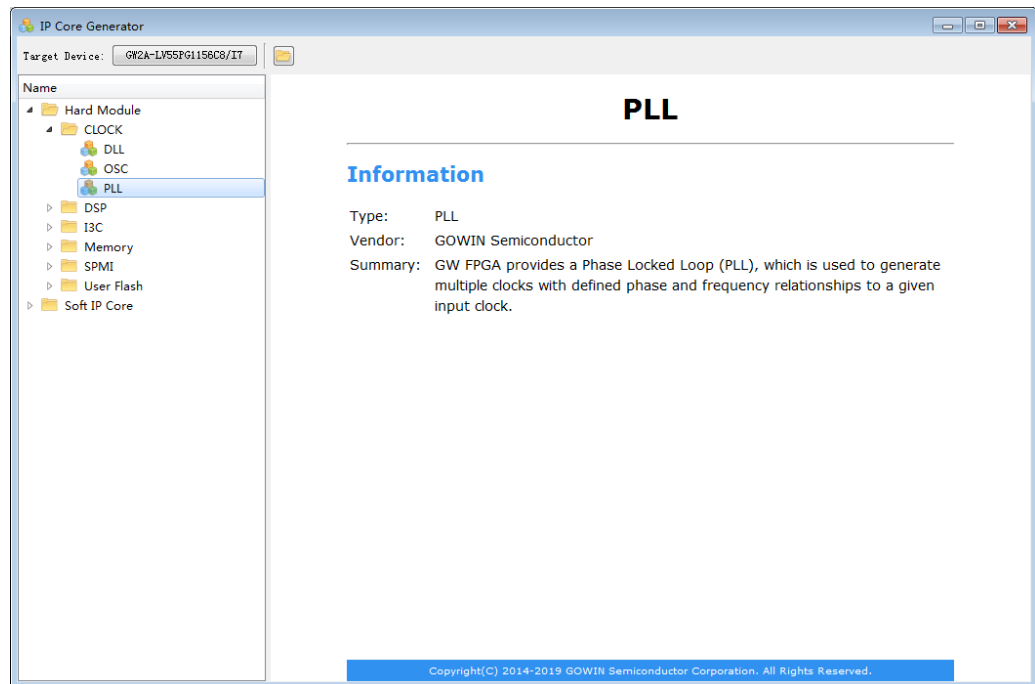
3.3 CLOCK

当前，CLOCK 模块支持三种高云器件的产生：PLL、DLL、OSC。

3.3.1 PLL

PLL 可基于给定的输入 `clkin` 进行时钟相位调整、占空比调整、频率调整（倍频和分频）等来产生不同相位和频率的输出。在 IP Core Generator 界面中，单击“PLL”，界面右侧会显示 PLL 的相关信息概要，如图 3-73 所示。

图 3-73 PLL 的信息概要



PLL 输出数据计算公式如下：

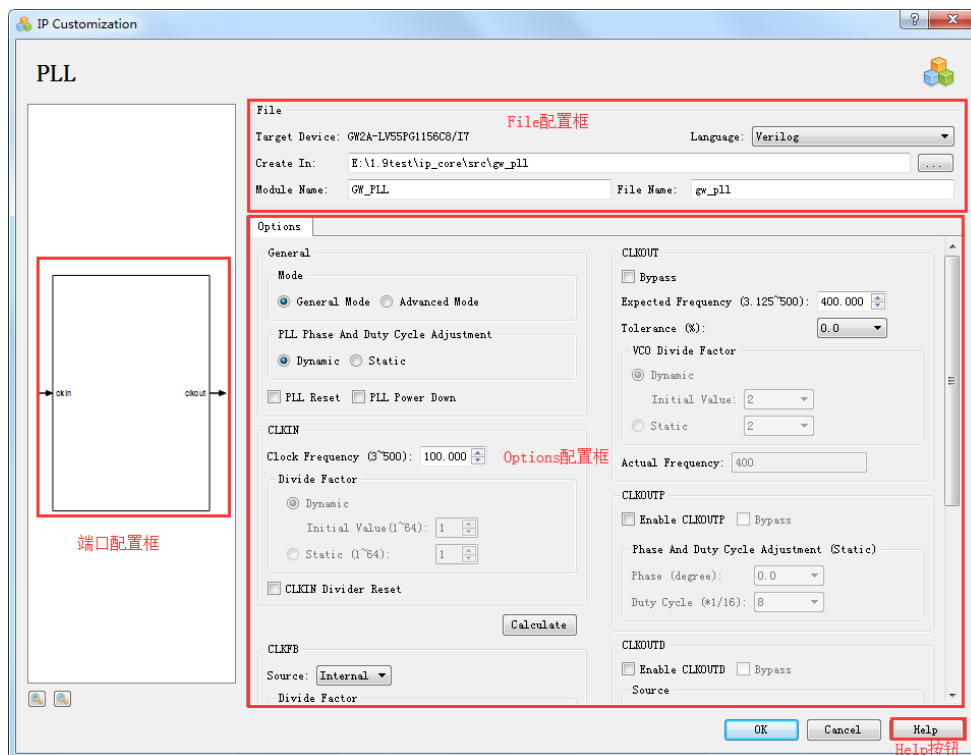
1. $f_{\text{CLKOUT}} = (f_{\text{CLKIN}} * \text{FDIV}) / \text{IDIV}$
2. $f_{\text{CLKOUTD}} = f_{\text{CLKOUT}} / \text{SDIV}$
3. $f_{\text{VCO}} = f_{\text{CLKOUT}} * \text{ODIV}$

注！

- f_{CLKIN} 为输入时钟 CLKIN 频率；
- f_{CLKOUT} 为输出时钟 CLKOUT 频率；
- f_{CLKOUTD} 为输出时钟 CLKOUTD 频率，CLKOUTD 是 CLKOUT 分频后的时钟；
- f_{VCO} 为 VCO 震荡频率。

在 IP Core Generator 界面中双击“PLL”，弹出 PLL 的“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-74 所示。

图 3-74 PLL 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生的 PLL 实例化文件的相关信息，如图 3-74 中标注的“File 配置框”所示。

PLL 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 PLL 设计文件中 PLL 的配置信息，标注的“Options 配置框”如图 3-74 所示。

- **General:** 配置一般模式和高级模式，配置输出相位和占空比调整的动态、静态模式和使能 PLL Reset。
 - “Mode”选项配置 IP Core 配置的模式，支持一般模式“General Mode”和高级模式“Advanced Mode”；
 - “PLL Phase And Duty Cycle Adjustment”选项配置输出的占空比和相位调整的模式，支持动态调整“Dynamic”和静态调整“Static”；
 - “PLL Reset”选项配置 PLL 的 Reset 使能模式；
 - “PLL Power Down”选项配置 reset_p 端口使 PLL 处于节电模式。
- **CLKIN:** 配置 PLL 输入时钟的频率，分频参数的设置和 IDSEL Reset 使能模式。
 - “Clock Frequency(频率范围)”配置输入时钟的频率，范围由 device 决定；
 - “Divide Factor”可在高级模式下配置分频参数，支持动态模式“Dynamic”和静态模式“Static”，静态模式下可配置分频参数的具体数值，范围为 1~64。若 CLKOUT 的输出频率不在相应 device 要求的范围内，单击“Calculate”或“OK”，会弹出提示窗口提示错

误，如图 3-75 所示；若 CLKIN/IDIV 的频率不在相应 device 要求的 Clock Frequency 范围内，单击“Calculate”或“OK”，会弹出提示窗口提示错误，如图 3-76 所示；

- “CLKIN Divider Reset”选项配置 CLKIN Divider 复位端口。
- **CLKFB**: 配置 PLL 反馈时钟的源和倍频参数。
 - 配置反馈时钟的源时，“Source”选项可选择 Internal 和 External；
 - “Divide Factor”可在高级模式下配置倍频参数，支持动态模式“Dynamic”和静态模式“Static”，静态模式下可配置倍频参数的具体数值，范围为 1~64，配置不合理时，单击“Calculate”按钮或“OK”按钮，会弹出提示窗口提示错误，如图 3-75 所示。
- **Enable LOCK**: 使能 LOCK 端口。
- **CLKOUT**: 配置 PLL 输出时钟期望频率，配置 VCO 参数，配置输出时钟周期的微调参数。
 - “Bypass”选项可配置输出时钟的旁路功能；
 - “Expected Frequency（频率范围）”在一般模式下配置期望的输出时钟 CLKOUT 的频率，范围由 device 决定；
 - “Tolerance（%）”配置 CLKOUT 期望频率和计算出的实际频率的允许误差。
 - “VCO Divide Factor”在高级模式下配置 VCO 参数支持动态模式“Dynamic”和静态模式“Static”，静态模式下可配置分频参数的具体数值，范围为 2/4/8/16/32/48/64/80/96/112/128，配置不合理时，单击“Calculate”或“OK”，会弹出提示窗口提示错误，如图 3-75 所示；
 - “Actual Frequency”显示经计算得出的 CLKOUT 实际频率，无需用户配置。
- **CLKOUTP**: 配置相移时钟周期微调参数，配置相移时钟的相位和占空比调整参数，使能/失能相移时钟的 Reset。
 - “Enable CLKOUTP”选项配置相移时钟输出使能；
 - “Bypass”选项配置相移时钟的旁路功能使能；
 - “Phase And Duty Cycle Adjustment（Static）”可在静态模式下配置相位（Phase（degree））和占空比（Duty Cycle（*1/16））；
- **CLKOUTD**: 配置分频时钟输出的时钟源，配置期望分频时钟输出频率，配置分频时钟分频输出参数，使能/失能分频时钟输出的 Reset。
 - “Enable CLKOUTD”选项配置分频时钟输出使能；
 - “Bypass”选项配置分频时钟输出的旁路功能使能；
 - “Source”选项配置分频时钟输出的时钟源，可选 CLKOUT 和 CLKOUTP；
 - “Expected Frequency（频率范围）”在一般模式下配置期望的分频时钟输出的频率，范围由 device 决定；
 - “Tolerance（%）”配置分频时钟输出期望频率和计算出的实际频率的允许误差；
 - “Divide Factor（2~128）”在高级模式下配置分频时钟输出的分频参数，范围为 2~128 之间的偶数，设置为奇数时单击 OK 会提示错误，如图 3-77 所示；

- “Actual Frequency” 显示经计算得出的分频时钟输出的实际频率，无需用户配置；
- CLKOUTD3: 配置三分频时钟输出的时钟源。
 - “Enable CLKOUTD3” 选项配置三分频时钟输出使能；
 - “Source” 选项配置三分频时钟输出的时钟源，可选 CLKOUT 和 CLKOUTP。
- “CLKOUTD/CLKOUTD3 Divider Reset” 选项在 CLKOUTD 或 CLKOUTD3 使能时可勾选，配置 CLKOUTD/ CLKOUTD3 Divider 的复位端口。
- Calculate: “Calculate” 按钮在一般模式 “General Mode” 下，根据输入输出频率计算配置分频参数，倍频参数和 VCO 参数，计算出的实际频率和期望频率不相等时，单击 “Calculate” 按钮后会弹出 “error” 窗口提示错误，并将不合理位置标红。
 - 如图 3-78 所示，为 CLKOUT 计算出的实际频率和期望频率不相等弹出的提示窗；
 - 如图 3-79 所示，为分频时钟输出计算出的实际频率和期望频率不相等弹出的提示窗。
 - 在高级模式 “Advanced Mode” 下，计算配置的静态分频参数，倍频参数和 VCO 参数是否合理。
 - 若不合理，单击 “Calculate”，弹出 “error” 窗口提示错误，并将不合理位置标红，如图 3-80 所示，为 “VCO Divide Factor” 配置不合理时的情况；
 - 若配置正确，单击 “Calculate”，弹出 “info” 窗口提示配置成功，如图 3-81 所示。

图 3-75 CLKIN/CLKFB Divide Factor 配置不合理的 error 窗口

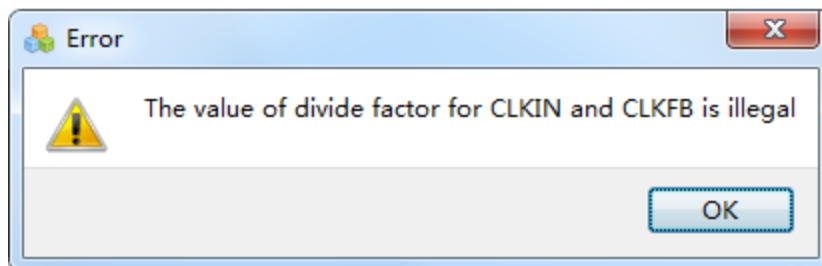


图 3-76 CLKIN Divide Factor 配置不合理的 error 窗口

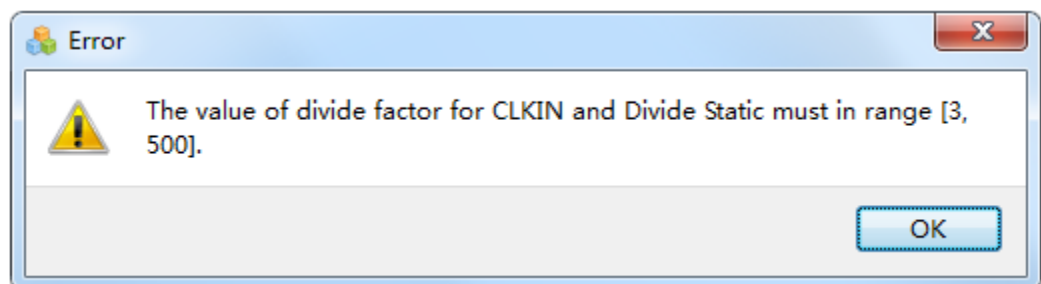


图 3-77 CLKOUTD 分频参数配置不合理的 Error 窗口

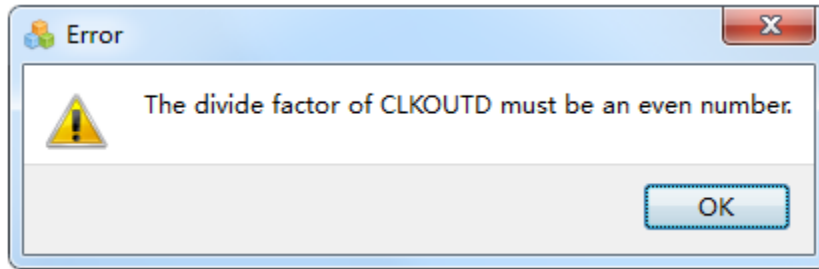


图 3-78 CLKOUT 期望频率与计算频率不等的 Error 窗口

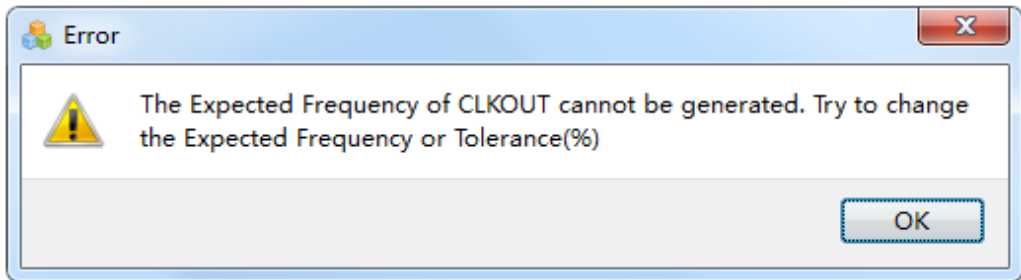


图 3-79 CLKOUTD 期望频率与计算频率不等的 Error 窗口

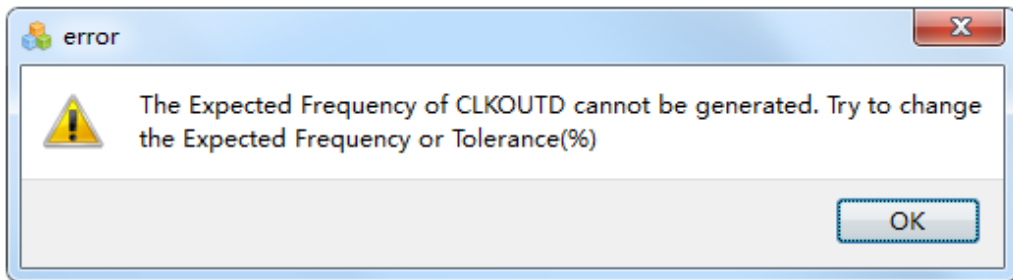


图 3-80 VCO 参数配置不合理的 error 窗口

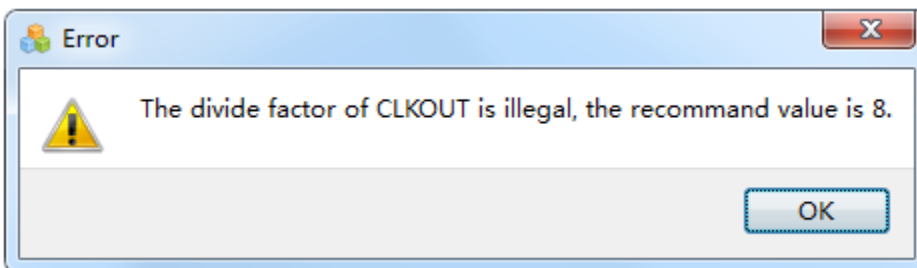
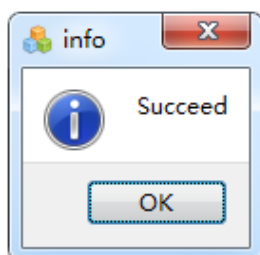


图 3-81 参数配置合理的 info 窗口



3. 端口配置框图

配置框图显示 IP Core 的配置结果示例框图，输入输出端口的个数以及 Options 配置实时更新，如图 3-74 中标注的“配置框图”所示。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-82 所示。Help 页面包括 IP Core 的概要介绍以及 Options 各项配置的简要说明

图 3-82 Help 信息

PLL	
Information	
Type:	PLL
Vendor:	GOWIN Semiconductor
Summary:	GW FPGA provides a Phase Locked Loop (PLL), which is used to generate multiple clocks with defined phase and frequency relationships to a given input clock.
Options	
Option	Description
General	General Mode - In this mode, entering input clock frequency and expected frequencies, software will automatically calculate divide factors.
	Advanced Mode - This mode is for advanced users. Allows you to enter input clock frequency and divide factors to achieve expected output frequency.
	PLL Phase And Duty Cycle Adjustment - Allows you to select Static Mode or Dynamic Mode.
	PLL Reset - Provides a reset pin to reset the PLL.
	PLL Power Down - Provides a reset_p port to power down the PLL.
CLKIN	CLKIN is the input reference clock for the PLL.
	CLKIN Frequency - Specify its frequency in MHz.
	Divide Factor - If in Advanced mode, also choose a divide factor which is from Dynamic or Static mode to achieve the expected output frequency. Static mode means select a static value from the drop-down list as divide factor, while Dynamic mode means that choose the value of port idsel as dynamic divide factor. When the Dynamic mode is selected, the user needs to set an initial value.
CLKFB	CLKIN Divider Reset - Provides a reset_i port to reset the input clock divider.
	Source - Specify the source of feedback.
LOCK	Divide Factor - In Advanced mode, the divide factor in the feedback path can be selected from port fbdsel or from the drop-down list. In General mode, the divide factor is shown when the "Calculate" button is clicked. When the Dynamic mode is selected, the user needs to set an initial value.
CLKOUT	Enable LOCK - Selecting this option will produce the lock port in the generated module.
	Bypass - The bypass option means clkout = clkIn, it connects the output to the input, bypassing the PLL circuit. Bypassing CLKOUT disables the CLKOUT expected frequency and tolerance fields. If both CLKOUT and CLKOUTP are in Bypass, then everything is disabled except the CLKIN frequency option. The following options are not available when CLKOUT is in Bypass mode.
	VCO Divide Factor - In General mode, VCO Divide Factors cannot be selected. Clicking the "Calculate" button displays the actual values. In Advanced mode, Dynamic or Static mode can be selected. When the Dynamic mode is selected, the user needs to set an initial value.
	Expected Frequency - In General mode, set the output clock frequency.
	Tolerance - Set a tolerance for the clkout frequency, as a percentage of requested frequency. Since the divide factors can only take a certain set of values, not all frequencies can be generated. The tolerance value is used to guide the tool to select a suitable divide factor.
	Actual Frequency - Clicking the "Calculate" button displays the actual frequency that the PLL can produce.

CLKOUTP	<p>Enable CLKOUTP – Selecting this option will produce the clkoutp port in the generated module. Clkoutp has the same frequency as clkout and it has the specified phase relation with clkout.</p> <p>Bypass – This option will connect the output to the input, bypassing the PLL circuit. Bypassing clkoutp disables its frequency, tolerance, and phase shift fields.</p> <p>PLL Phase And Duty Cycle Adjustment(Static)</p> <p>Phase - You can select phase shift in 22.5-degree increments from 0 to 360.</p> <p>Duty Cycle - Allows duty cycle selection in 1/16 increments.</p>
CLKOUTD	<p>Enable CLKOUTD - Selecting this option will produces clkoutd port in the generated module. In general mode, select the expected frequency. In advanced mode, select the divide factor for the expected frequency.</p> <p>Bypass - The bypass option means clkoutd = clkout, it connects the output to the input, bypassing the PLL circuit.</p> <p>Source - If clkoutd is enabled, you can select CLKOUT or CLKOUTP as the source of clkoutd. If CLKOUTP is not enabled, CLKOUT is used as the source.</p> <p>Expected Frequency - In General mode, set the CLKOUTD Frequency. This option is disabled in advanced mode.</p> <p>Divide Factor - In General mode, the divide factor cannot be selected. Clicking the "Calculate" button the tool will automatically displays the actual values. In Advanced mode, select the CLKOUTD divide factor from the drop-down list.</p> <p>Tolerance - Set a tolerance for the CLKOUTD frequency, as a percentage of requested frequency. Since the divide factors can only take a certain set of values, not all frequencies can be generated. The tolerance value is used to guide the tool to select a suitable divide factor.</p> <p>Actual Frequency - Clicking the "Calculate" button displays the actual frequency that the PLL can produce.</p>
CLKOUTD3	<p>Enable CLKOUTD3 - Selecting this option will produces clkoutd3 port in the generated module, and it is equal to clkout/3.</p> <p>Source - If CLKOUTD3 is enabled, you can select CLKOUT or CLKOUTP as the source of CLKOUTP. If CLKOUTP is not enabled, CLKOUT is used as the source.</p>
CLKOUTD/CLKOUTD3 Divider Reset	Select it if CLKOUTD or CLKOUTD3 divider reset is required as an input port. When CLKOUTD or CLKOUTD3 is enabled and this option is selected, an input port reset_s is generated.
Calculate	<p>General Mode: The tool calculates Divide Factor settings based on input/output frequency.</p> <p>Advanced Mode: The tool calculates output frequencies based on divide factors.</p>

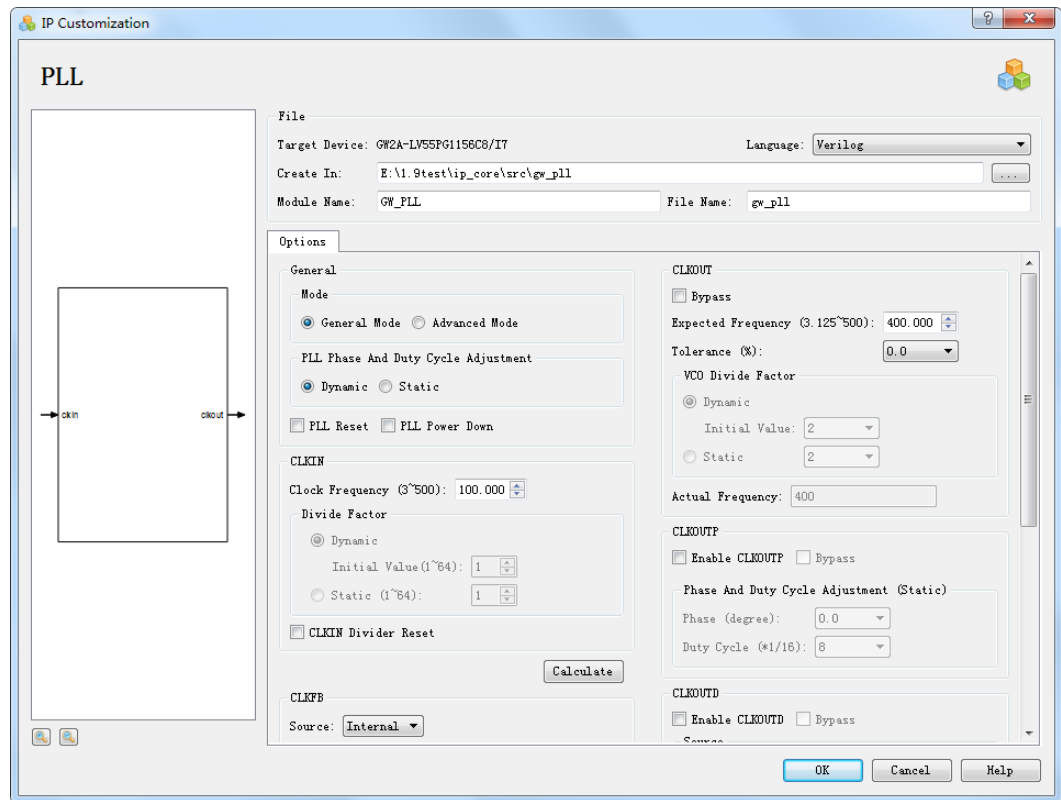
IP 生成文件

如图 3-83 所示，PLL 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 PLL 设计文件“gw_pll.v”；
- 用户例化该 IP 设计文件的模板文件“gw_pll_tmp.v”；
- 例化原语 PLL 的配置文件“gw_pll.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-83 配置的 IP Customization



例化 PLL 设计文件

例化 PLL 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 PLL 配置，产生了实例化的 PLL，如图 3-84 所示。

图 3-84 例化 PLL 设计文件

```

module GW_PLL (clkout, clkkin);

output clkout;
input clkkin;

wire lock_o;
wire clkoutp_o;
wire clkoutd_o;
wire clkoutd3_o;
wire gw_gnd;

assign gw_gnd = 1'b0;

PLL pll_inst (
    .CLKOUT(clkout),
    .LOCK(lock_o),
    .CLKOUTP(clkoutp_o),
    .CLKOUTD(clkoutd_o),
    .CLKOUTD3(clkoutd3_o),
    .RESET(gw_gnd),
    .RESET_P(gw_gnd),
    .RESET_I(gw_gnd),
    .RESET_S(gw_gnd),
    .CLKIN(clkkin),
    .CLKFB(gw_gnd),
    .FBDSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .IDSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .ODSEL({gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .PSDA({gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .DUTYDA({gw_gnd,gw_gnd,gw_gnd,gw_gnd}),
    .FDLY({gw_gnd,gw_gnd,gw_gnd,gw_gnd})
);

defparam pll_inst.FCLKIN = "100";
defparam pll_inst.DYN_IDIV_SEL = "false";
defparam pll_inst.IDIV_SEL = 0;
defparam pll_inst.DYN_FBDIV_SEL = "false";
defparam pll_inst.FBDIV_SEL = 3;
defparam pll_inst.DYN_ODIV_SEL = "false";
defparam pll_inst.ODIV_SEL = 2;
defparam pll_inst.PSDA_SEL = "0000";
defparam pll_inst.DYN_DA_EN = "true";
defparam pll_inst.DUTYDA_SEL = "1000";
defparam pll_inst.CLKOUT_FT_DIR = 1'b1;
defparam pll_inst.CLKOUTP_FT_DIR = 1'b1;
defparam pll_inst.CLKOUT_DLY_STEP = 0;
defparam pll_inst.CLKOUTP_DLY_STEP = 0;
defparam pll_inst.CLKFB_SEL = "internal";
defparam pll_inst.CLKOUT_BYPASS = "false";
defparam pll_inst.CLKOUTP_BYPASS = "false";
defparam pll_inst.CLKOUTD_BYPASS = "false";
defparam pll_inst.DYN_SDIV_SEL = 2;
defparam pll_inst.CLKOUTD_SRC = "CLKOUT";
defparam pll_inst.CLKOUTD3_SRC = "CLKOUT";
defparam pll_inst.DEVICE = "GW2A-55";

endmodule //GW_PLL

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 PLL 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-85 所示。

图 3-85 用户例化该 IP 设计文件的模板文件

```

GW_PLL your_instance_name(
    .clkout(clkout_o), //output clkout
    .clkin(clkin_i) //input clkin
);

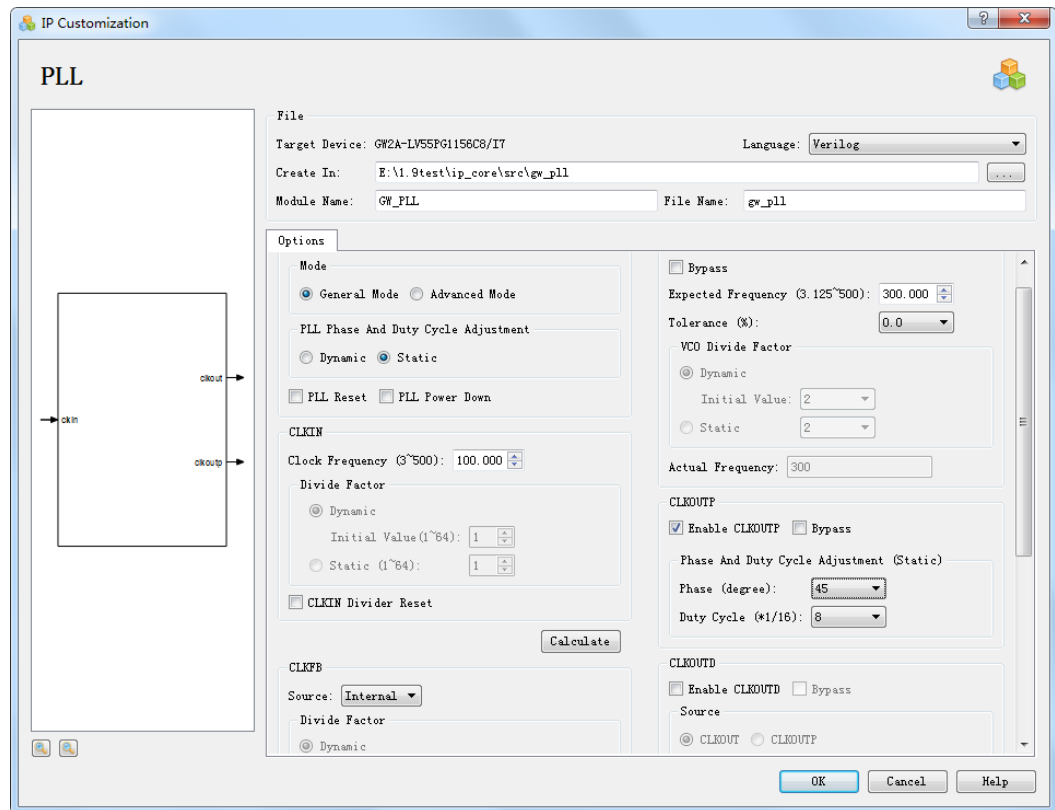
```

IP Core Generator 生成 PLL 示例

如用户需产生输入时钟频率为 100MHz，输出时钟频率为 300MHz，相移时钟输出使能且相位调整为 45，分频时钟输出使能且输出频率为 150MHz 的 PLL IP，以 device 选择 GW2A-LV55PG1156C8/I7，General Mode 为例，界面配置如图 3-86 所示，单击“OK”，产生用户所需的 PLL IP 设计文件。

产生的 PLL IP 设计文件所在目录即为配置界面中“Create In”设置路径。

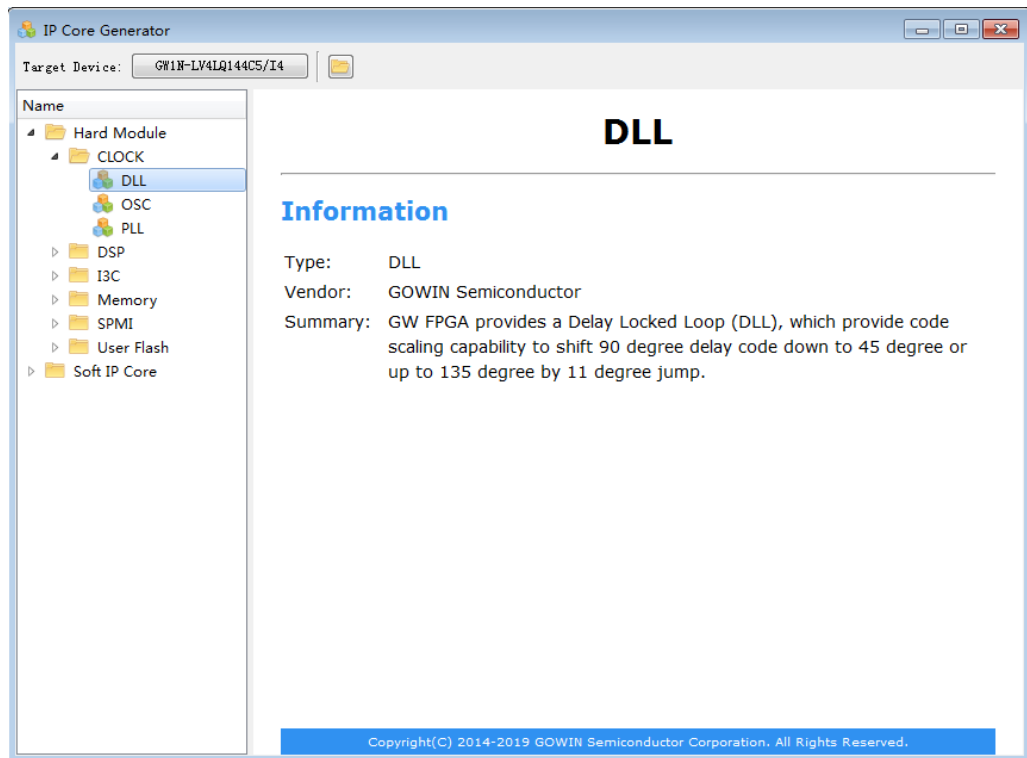
图 3-86 PLL IP Customization 设置



3.3.2 DLL

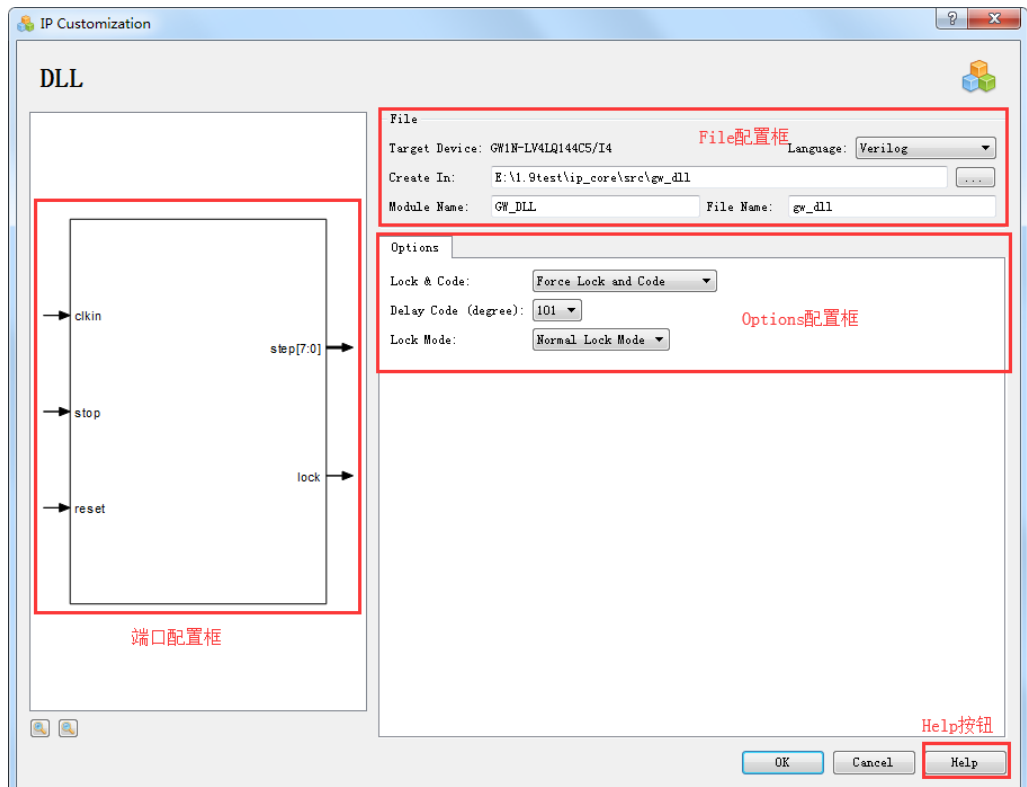
DLL 全称 Delay Lock Loop（延迟锁相环），主要用于通过对输入信号的周期做精确等分，产生一个精确地时间延时。在 IP Core Generator 界面中单击 DLL，界面右侧会显示 DLL 的相关信息概要，如图 3-87 所示。

图 3-87 DLL 的信息概要



在 IP Core Generator 界面中，双击“DLL”，弹出“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图，以及帮助按钮“Help”，如图 3-88 所示。

图 3-88 DLL 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 DLL 实例化文件的相关信息，如图 3-88 中标注的“File 配置框”所示。

DLL 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 DLL 设计文件中 DLL 的配置信息，如图 3-88 中标注的“Options 配置框”所示。

- **Lock & Code:** 选择 LOCK 和 STEP 的输出方式。
 - 当选择“Force Lock and Code”时，LOCK 输出为 1，STEP 输出为 255；
 - 当选择“Generated From DLL Loop”时，LOCK 和 STEP 的输出由 DLL 产生。
- **Delay Code (degree):** 相位差。可选择 101°、112°、123°、135°、79°、68°、57°、45°、90°相位差。
- **Lock Mode:** 输出 Lock 模式选择。
 - 当选择“Normal Lock Mode”时，DLL 的参数 DIV_SEL 设置为 1'b0，
 - 当选择“Fast Lock Mode”时，DLL 的参数 DIV_SEL 设置为 1'b1。

3. 端口配置框图

配置框图显示 IP Core 的配置结果示例框图，如图 3-88 的“配置框图”所示。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-89。

图 3-89 Help 信息

DLL	
Information	
Type:	DLL
Vendor:	GOWIN Semiconductor
Summary:	GW FPGA provides a Delay Locked Loop (DLL), which provide code scaling capability to shift 90 degree delay code down to 45 degree or up to 135 degree by 11 degree jump.
Options	
Option	Description
Lock & Code	Force Lock and Code - In this mode, the STEP value is forcibly set to 255. Generated From DLL Loop - This mode ensures STEP value is generated by the DLL Loop.
Delay Code(degree)	Specify a delay code (degree) for DLL.
Lock Mode	Allows you to select Normal Lock Mode or Fast Lock Mode .

IP 生成文件

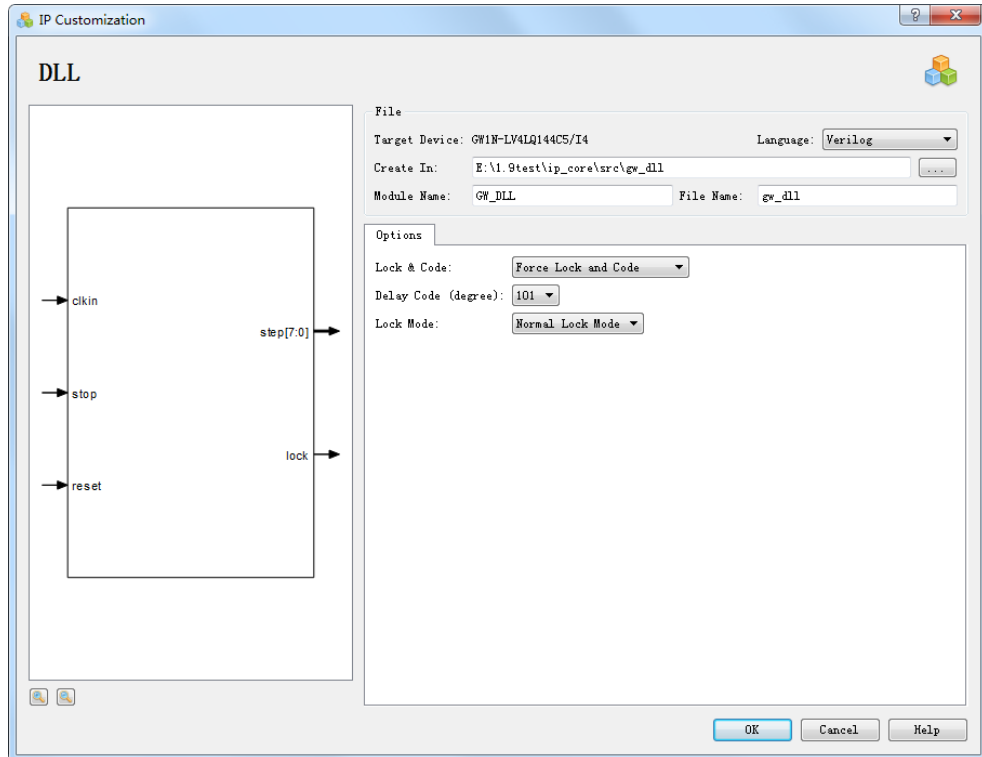
如图 3-90 所示，DLL 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 DLL 设计文件“gw_dll.v”；

- 用户例化该 IP 设计文件的模板文件 “gw_dll_tmp.v”;
- 例化原语 DLL 的配置文件 “gw_dll.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-90 配置的 IP Customization



例化 DLL 设计文件

例化 DLL 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 DLL 配置，产生了实例化的 DLL，如图 3-91 所示。

图 3-91 例化 DLL 设计文件

```

module GW_DLL (step, lock, clk_in, stop, reset);

output [7:0] step;
output lock;
input clk_in;
input stop;
input reset;

wire gw_gnd;

assign gw_gnd = 1'b0;

DLL dll_inst (
    .STEP(step),
    .LOCK(lock),
    .CLKIN(clk_in),
    .STOP(stop),
    .RESET(reset),
    .UPDNCNTL(gw_gnd)
);

defparam dll_inst.DLL_FORCE = 1;
defparam dll_inst.CODESCAL = "000";
defparam dll_inst.SCAL_EN = "true";
defparam dll_inst.DIV_SEL = 1'b0;

endmodule //GW_DLL

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 DLL 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-92 所示。

图 3-92 用户例化该 IP 设计文件的模板文件

```

GW_DLL your_instance_name(
    .step(step_o), //output [7:0] step
    .lock(lock_o), //output lock
    .clk_in(clk_in_i), //input clk_in
    .stop(stop_i), //input stop
    .reset(reset_i) //input reset
);

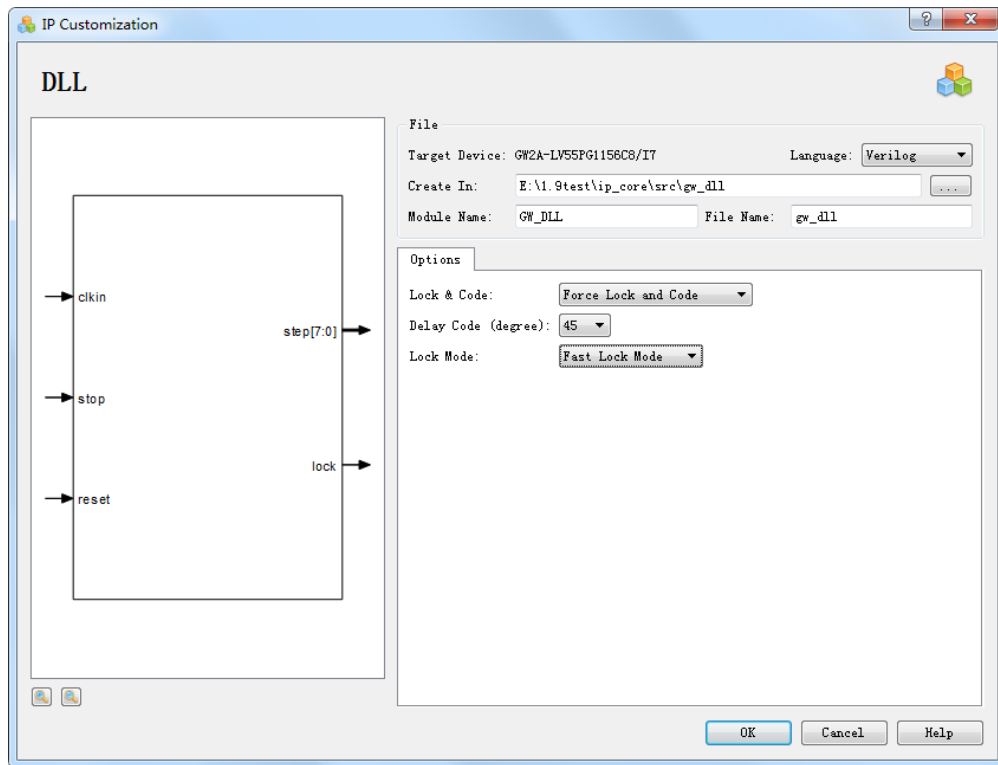
```

IP Core Generator 生成 DLL 示例

如用户需产生由 DLL 产生时钟延时，相位调整为 45°，Lock 为 Fast 模式的 DLL IP，以 device 选择 GW2A-LV55PG1156C8/I7 为例，界面配置如图 3-93 所示，单击“OK”，产生用户所需的 DLL IP 设计文件。

产生的 DLL IP 设计文件所在目录即为配置界面中“Create In”设置路径。

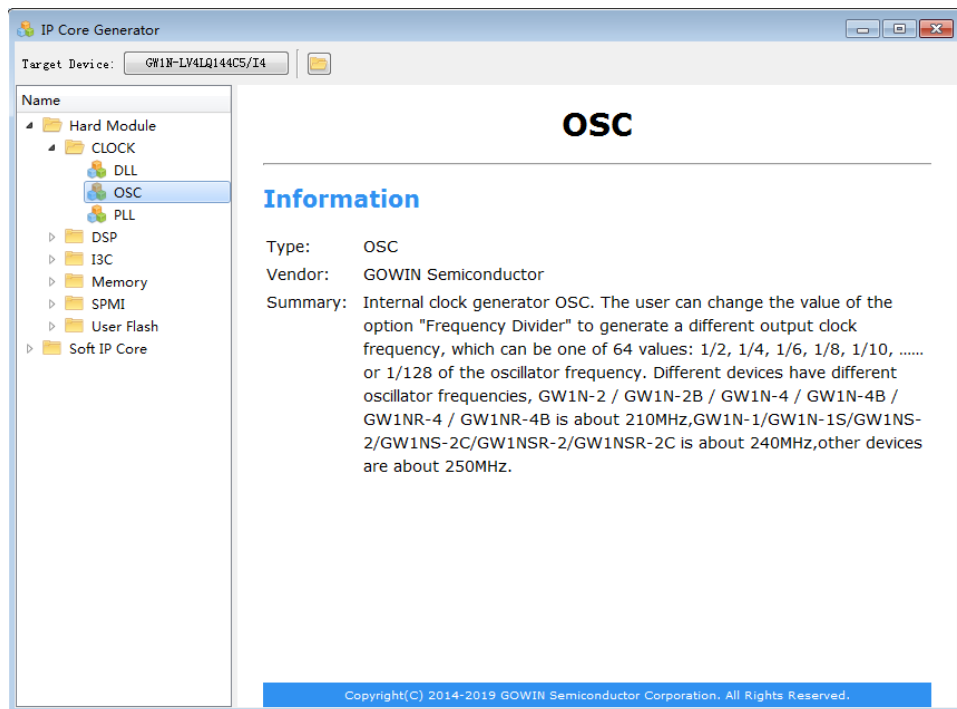
图 3-93 DLL IP Customization 设置



3.3.3 OSC

OSC 是片内晶振，其最大频率为 250MHz，在 IP Core Generator 界面中单击 OSC，界面右侧会显示 OSC 的相关信息概要，如图 3-94 所示。

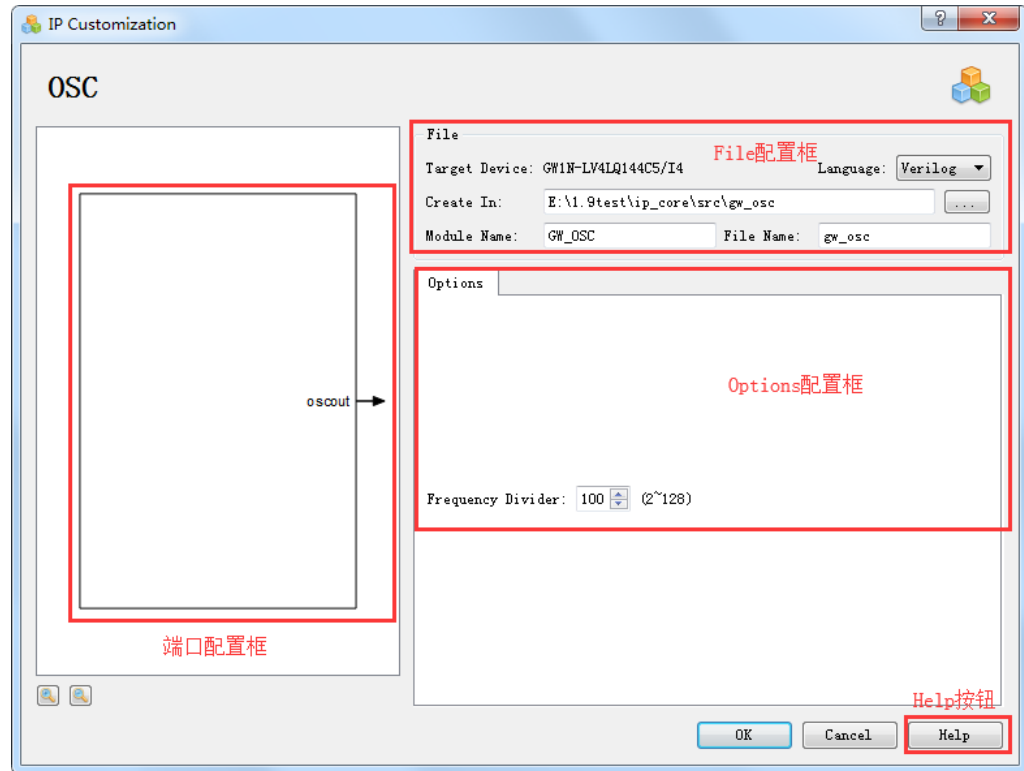
图 3-94 OSC 的信息概要



在 IP Core Generator 界面中，双击“OSC”，弹出 OSC 的“IP

Customization” 窗口，该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-95 所示。

图 3-95 OSC 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 OSC 实例化文件的相关信息，如图 3-95 中标注的“File 配置框”所示。

OSC 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于配置例化高云原语 OSC 设计文件中 OSC 的配置信息，如图 3-95 中标注的“Options 配置框”所示。

Frequency Divider: 分频值。该值为 2 的整数倍，取值范围为 2~128。

3. 端口配置框图

配置框图显示 IP Core 的配置结果示例框图，如图 3-95 中标注的“配置框图”所示

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-96 所示。Help 页面包括 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

图 3-96 Help 信息

OSC	
Information	
Type:	OSC
Vendor:	GOWIN Semiconductor
Summary:	Internal clock generator OSC. The user can change the value of the option "Frequency Divider" to generate a different output clock frequency, which can be one of 64 values: 1/2, 1/4, 1/6, 1/8, 1/10, or 1/128 of the oscillator frequency. Different devices have different oscillator frequencies, GW1N-2 / GW1N-2B / GW1N-4 / GW1N-4B / GW1NR-4 / GW1NR-4B is about 210MHz, GW1N-1/GW1N-1S/GW1NS-2/GW1NS-2C/GW1NSR-2/GW1NSR-2C is about 240MHz, other devices are about 250MHz.
Options	
Option	Description
Frequency Divider	Allows you to select any even number between 2 ~ 128.

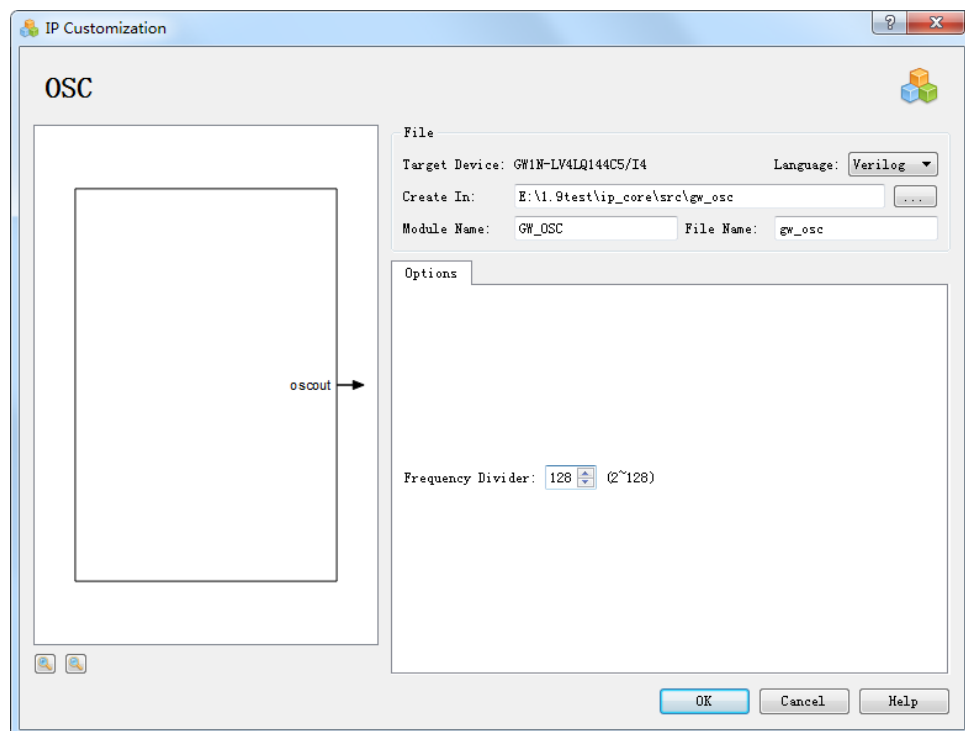
IP 生成文件

如图 3-97 所示，OSC 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 OSC 设计文件“gw_osc.v”；
- 用户例化该 IP 设计文件的模板文件“gw_osc_tmp.v”；
- 例化原语 OSC 的配置文件“gw_osc.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-97 配置的 IP Customization



例化 OSC 设计文件

例化 OSC 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 OSC 配置，产生了实例化的 OSC，如图 3-98 所示。

图 3-98 例化 OSC 设计文件

```

module GW_OSC (oscout);

output oscout;

OSC osc_inst (
    .OSCOUT(oscout)
);

defparam osc_inst.FREQ_DIV = 128;
defparam osc_inst.DEVICE = "GW2A-55";

endmodule //GW_OSC

```

用户例化该 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 OSC 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-99 所示。

图 3-99 用户例化该 IP 设计文件的模板文件

```

GW_OSC your_instance_name (
    .oscout(oscout_o) //output oscout
);

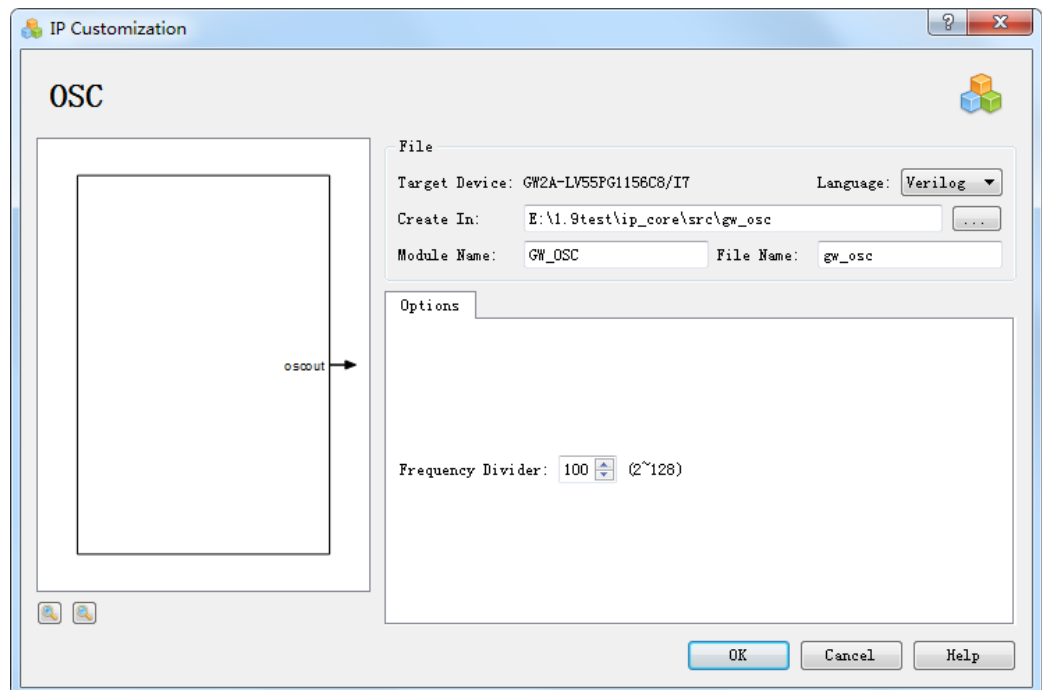
```

IP Core Generator 生成 OSC 示例

如用户需产生时钟频率为 2.5MHz 的 OSC IP，以 device 选择 GW2A-LV55PG1156C8/I7 例，界面配置如图 3-100 所示，单击“OK”，产生用户所需的 OSC IP 设计文件。

产生的 OSC IP 设计文件所在目录即为配置界面中“Create In”设置路径。

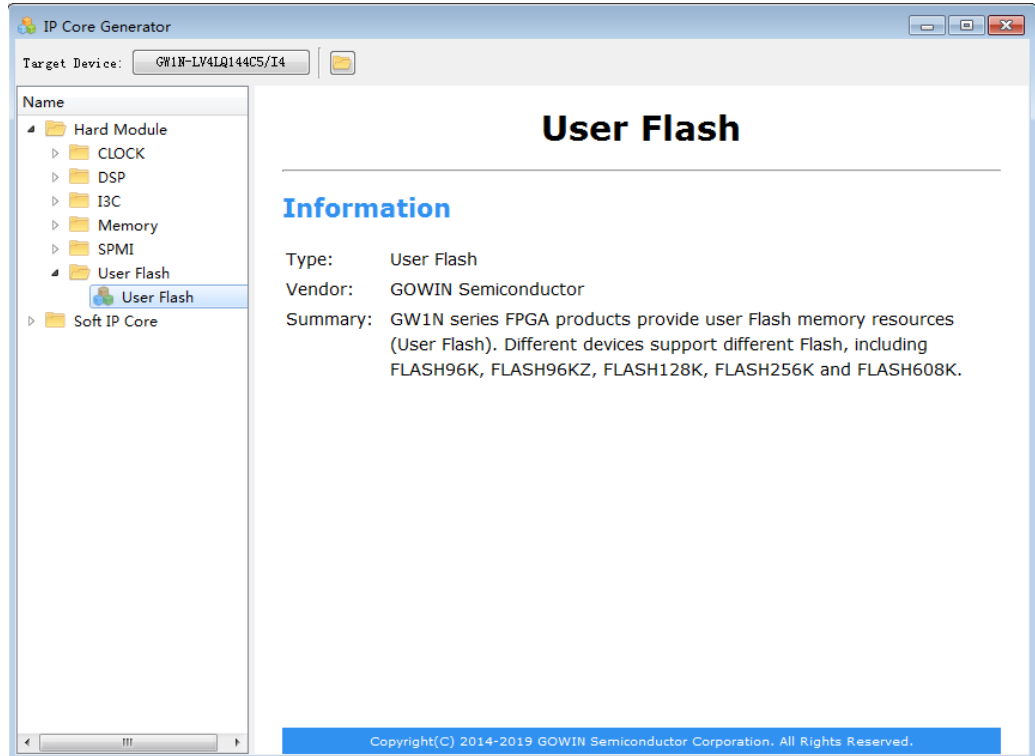
图 3-100 OSC IP Customization 设置



3.4 User Flash

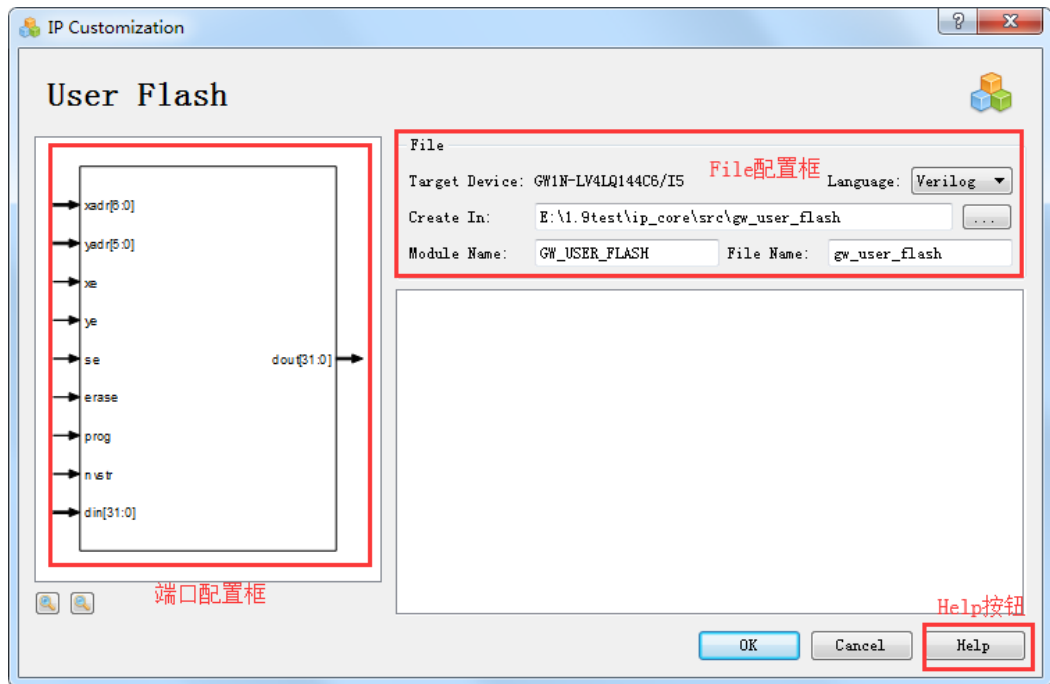
User Flash 是用户闪存。在 IP Core Generator 界面中单击 User Flash，界面右侧会显示 User Flash 的相关信息概要，如图 3-101 所示。

图 3-101 User Flash 的信息概要



在 IP Core Generator 界面中，双击“User Flash”，弹出 User Flash 的“IP Customization”窗口，该窗口包括 File 配置框和端口配置框图，如图 3-102 所示。

图 3-102 User Flash 的 IP Customization 窗口结构



5. File 配置框

File 配置框用于配置产生 User Flash 实例化文件的相关信息,如图 3-102 标注的“File 配置框”所示。

User Flash 的 File 配置框的使用和 SP 模块的类似,请参考 [3.1 Block Memory > 3.1.1 SP 中的 File 配置框](#)。

注!

- 目前支持 FLASH96K 的 device 有: GW1N-1/GW1N-1S
- 支持 FLASH96KZ 的 device 有: GW1NZ-1
- 支持 FLASH128K 的 device 有: GW1NS-2/ GW1NS-2C /GW1NSR-2/GW1NSR-2C
- 支持 FLASH256K 的 device 有: GW1N-2/ GW1N-2B/GW1N-4/ GW1N-4B/ GW1NR-4/ GW1NR-4B
- 支持 FLASH608K 的 device 有: GW1N-6/ GW1N-9/ GW1NR-9
- 若 Target Device 选择除上述之外的其他 device, User Flash 置灰,无法产生对应的 IP。

6. 端口配置框图

User Flash 的输入位宽与 Device 的选择有关,根据不同的 Device 信息会在配置框图中显示当前 IP Core 的配置结果示例框图,如图 3-102 中标注的“配置框图”所示。

7. Help 按钮

单击“Help”,显示 IP Core 的配置信息的页面,如图 3-103 所示。Help 页面包括 IP Core 的概要介绍,以及 Options 各项配置的简要说明。

图 3-103 Help 信息

User Flash

Information

Type:	User Flash
Vendor:	GOWIN Semiconductor
Summary:	GW1N series FPGA products provide user Flash memory resources (User Flash). Different devices support different Flash, including FLASH96K, FLASH96KZ, FLASH128K, FLASH256K and FLASH608K.

Note

Description
If the target device is GW1N-1/GW1N-1S, the primitive FLASH96K will be instantiated in the customized module. For the primitive FLASH96K, data input and data output's width is 32, input RA and CA's width is 6.
If the target device is GW1NZ-1, the primitive FLASH96KZ will be instantiated in the customized module. For the primitive FLASH96KZ, data input and data output's width is 32, input XADR and YADR's width is 6.
If the target device is GW1NS-2/GW1NS-2C/GW1NSR-2/GW1NSR-2C, the primitive FLASH128K (128K Bytes) will be instantiated in the customized module. For the primitive FLASH128K, data input and data output's width is 32, input ADDR's width is 15.
If the target device is GW1N-2/GW1N-2B/GW1N-4/ GW1N-4B/GW1NR-4/GW1NR-4B, the primitive FLASH256K will be instantiated in the customized module. For the primitive FLASH256K, data input and data output's width is 32, input XADR's width is 7, input YADR's width is 6.
If the target device is GW1N-6/GW1N-9/GW1NR-9, the primitive FLASH608K will be instantiated in the customized module. For the primitive FLASH608K, data input and data output's width is 32, input XADR's width is 9, input YADR's width is 6.

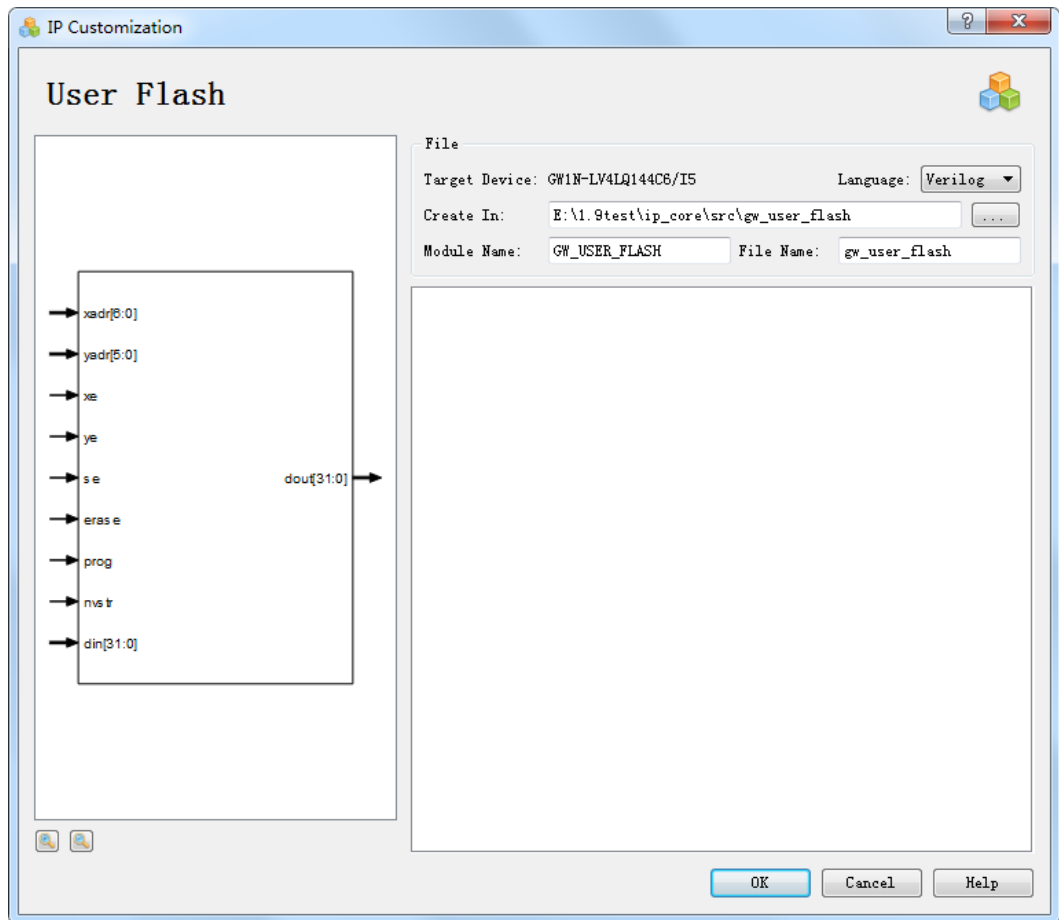
IP 生成文件

如图 3-104 所示，User Flash 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 User Flash 设计文件“gw_user_flash.v”；
- 用户例化该 IP 设计文件的模板文件“gw_user_lash_tmp.v”；
- 例化原语 User Flash 的配置文件“gw_user_flash.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-104 配置的 IP Customization



例化 User Flash 设计文件

例化 User Flash 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 User Flash 配置，产生了实例化的 User Flash，如图 3-105 所示。GW1N-4 产生的设计文件中实例化的是原语 FLASH256K。

图 3-105 例化 User Flash 设计文件

```

module GW_USER_FLASH (dout, xe, ye, se, prog, erase, nvstr, xadr, yadr, din);

output [31:0] dout;
input xe;
input ye;
input se;
input prog;
input erase;
input nvstr;
input [6:0] xadr;
input [5:0] yadr;
input [31:0] din;

FLASH256K flash_inst (
    .DOUT(dout),
    .XE(xe),
    .YE(ye),
    .SE(se),
    .PROG(prog),
    .ERASE(erase),
    .NVSTR(nvstr),
    .XADR(xadr),
    .YADR(yadr),
    .DIN(din)
);

endmodule //GW_USER_FLASH

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 User Flash 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-106 所示。

图 3-106 用户例化该 IP 设计文件的模板文件

```

GW_USER_FLASH your_instance_name(
    .dout(dout_o), //output [31:0] dout
    .xe(xe_i), //input xe
    .ye(ye_i), //input ye
    .se(se_i), //input se
    .prog(prog_i), //input prog
    .erase(erase_i), //input erase
    .nvstr(nvstr_i), //input nvstr
    .xadr(xadr_i), //input [6:0] xadr
    .yadr(yadr_i), //input [5:0] yadr
    .din(din_i) //input [31:0] din
);

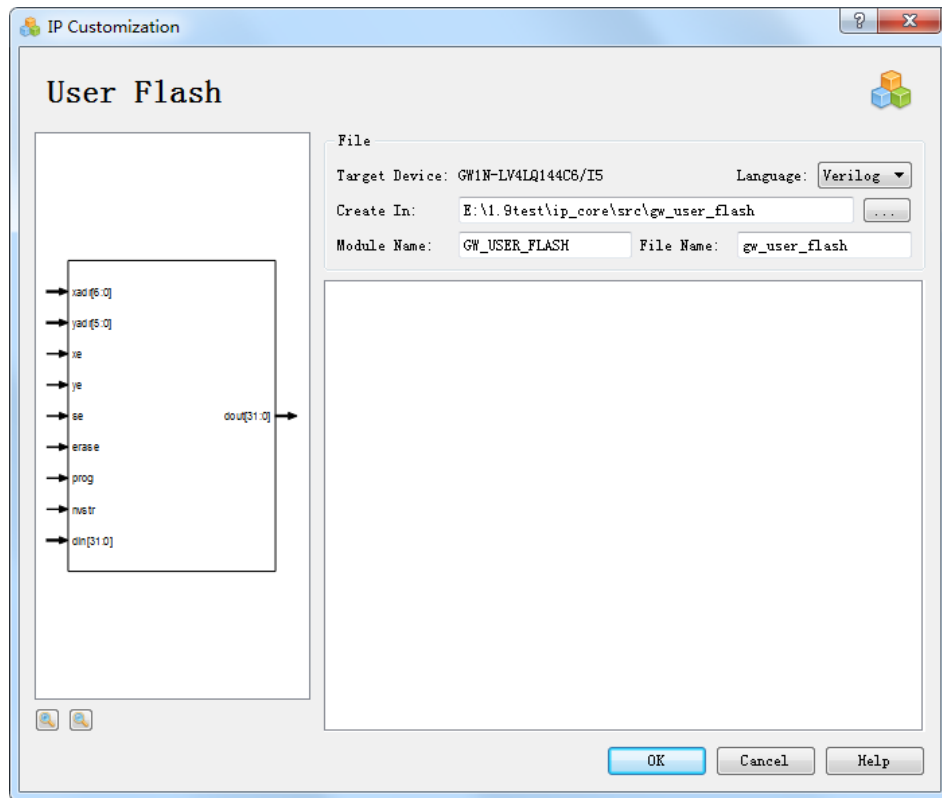
```

IP Core Generator 生成 User Flash 示例

以产生 GW1N-4 器件支持的 FLASH256K 为例，只需在 IP 配置界面选择 device 为 GW1N-LV4LP144C6/I5，界面配置如图 3-107 示，单击“OK”，产生用户所需的 User Flash IP 设计文件。

产生的 User Flash IP 设计文件所在目录即为配置界面中“Create In”设置路径。

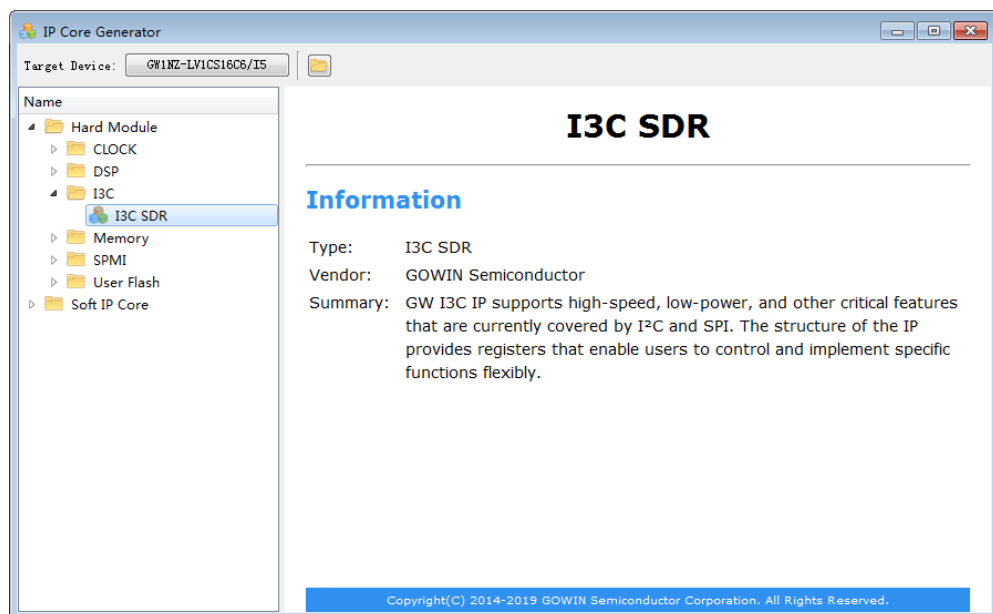
图 3-107 User Flash IP Customization 设置



3.5 I3C

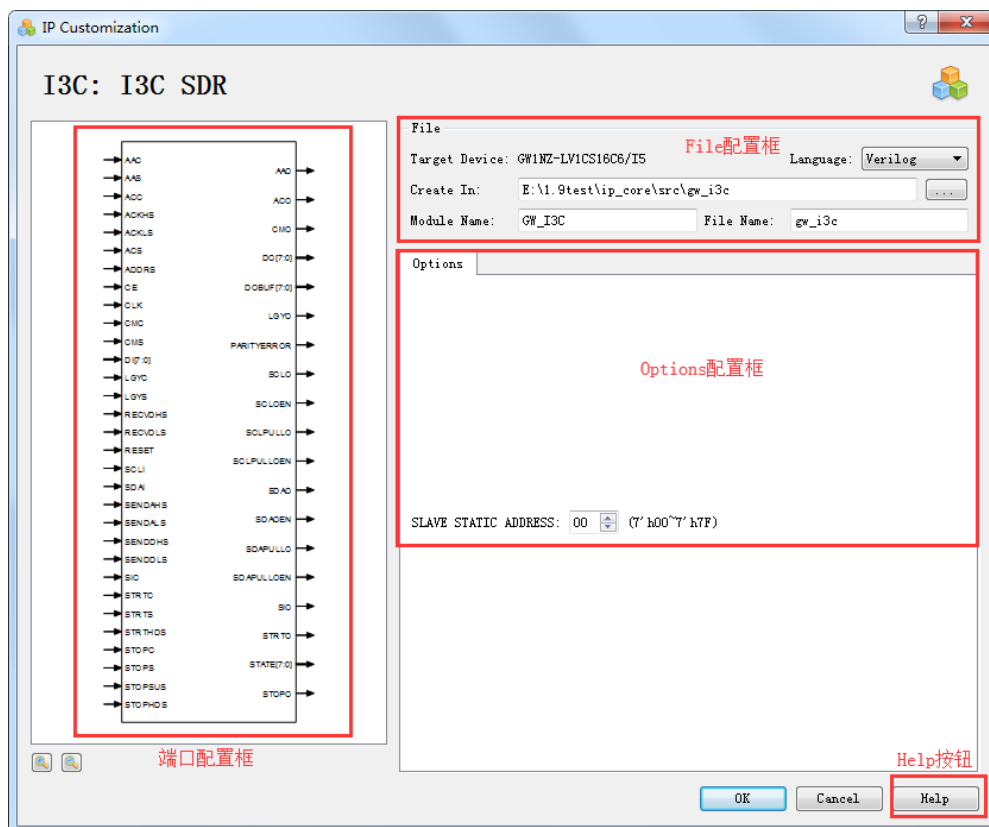
I3C IP 具有高速、低功耗，兼容 I2C 和 SPI 的其他关键特性。IP 结构提供了寄存器，使用户能够灵活地控制和实现特定的功能。在 IP Core Generator 界面中单击 I3C 下的 I3C SDR，界面右侧会显示 I3C SDR 的相关信息概要，如图 3-108 所示。

图 3-108 I3C SDR 的信息概要



在 IP Core Generator 界面中，双击“I3C SDR”，弹出 I3C 的“IP Customization”窗口，该窗口包括 Options 配置框、File 配置框和端口配置框图，如图 3-109 所示。

图 3-109 I3C 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 I3C 实例化文件的相关信息，如图 3-109 标注的“File 配置框”所示。

I3C 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

注！

目前仅有 GW1NZ-1 支持 I3C，如 Target Device 选择其他 device，I3C 置灰，无法产生对应的 IP。

2. 端口配置框图

配置框图显示 IP Core 的配置结果示例框图，如图 3-109 中标注的“配置框图”所示。

3. Options 配置框图

Options 配置框用于配置例化原语 I3C 设计文件中 I3C 的配置信息，如图 3-109 中标注的“Options 配置框”所示。

SLAVE STATIC ADDRESS - 指定从机的静态地址。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-110 所示。Help 页面包括 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

图 3-110 Help 信息

I3C SDR

Information

Type:	I3C SDR
Vendor:	GOWIN Semiconductor
Summary:	GW I3C IP supports high-speed, low-power, and other critical features that are currently covered by I ² C and SPI. The structure of the IP provides registers that enable users to control and implement specific functions flexibly.

Options

Option	Description
SLAVE STATIC ADDRESS	SLAVE STATIC ADDRESS - Specify the static address of slave.

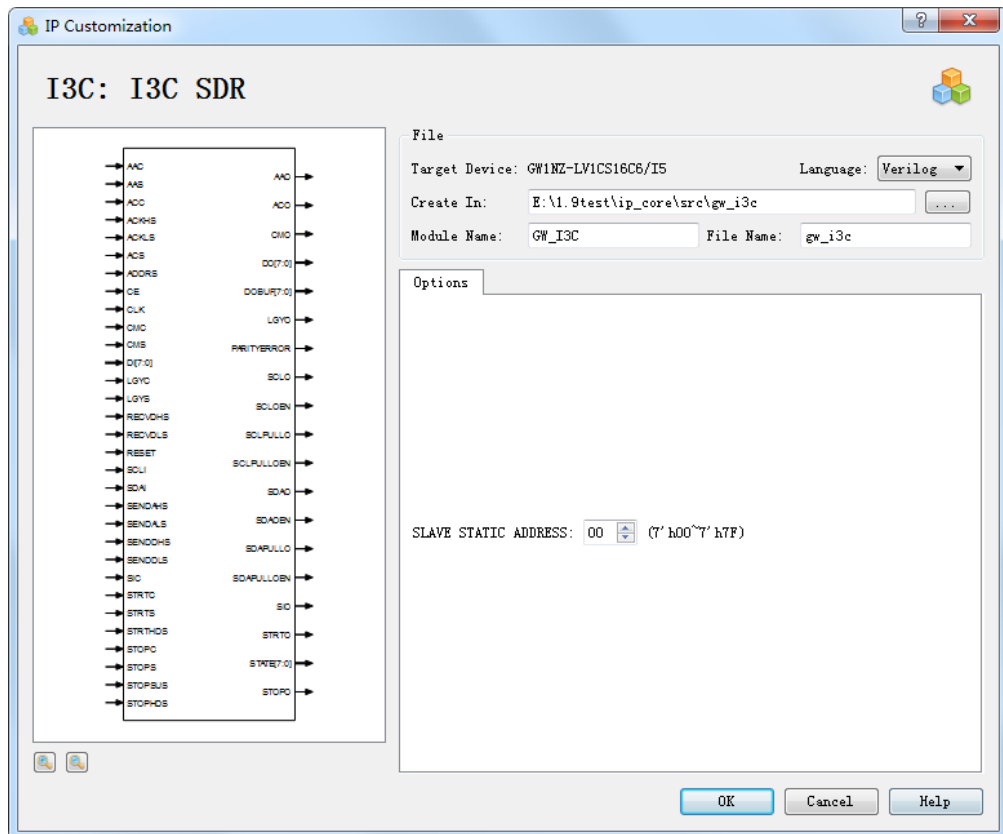
IP 生成文件

如图 3-111 所示, I3C 的“IP Customization”窗口配置完成后, 单击“OK”, 产生以配置文件“File Name”命名的三个文件:

- 例化高云原语 I3C 设计文件 “gw_i3c.v”;
- 用户例化该 IP 设计文件的模板文件 “gw_i3c_tmp.v”;
- 例化原语 I3C 的配置文件 “gw_i3c.ipc”。

下述是 verilog 语言产生的文件。

图 3-111 配置的 IP Customization



例化 I3C 设计文件

例化 I3C 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 I3C 配置，产生了实例化的 I3C，如图 3-112 所示。GW1NZ-1 产生的设计文件中实例化的是硬核原语 I3C。

图 3-112 例化 I3C 设计文件

```
module GW_I3C (lgyo, cmo, aco, aao, sio, stopo, strto, parityerror, dobuf, dout,
               state, sdao, sclo, sdaoen, scloen, sdapullo, sclpullo, sdapulloen,
               sclpulloen, lgys, cms, acs, aas, stops, strts, lgyc, cmc, acc, aac,
               sic, stopc, strtc, strthds, sendahs, sendals, ackhs, ackls, stopsus,
               stophds, senddhs, senddls, recvdhs, recvdls, addr, di, sdai, scli,
               ce, reset, clk);

    output lgyo;
    output cmo;
    output aco;
    output aao;
    output sio;
    output stopo;
    output strto;
    output parityerror;
    output [7:0] dobuf;
    output [7:0] dout;
    output [7:0] state;
    output sdao;
    output sclo;
    output sdaoen;
    output scloen;
    output sdapullo;
    output sclpullo;
    output sdapulloen;
    output sclpulloen;
    input lgys;
    input cms;
    input acs;
    input aas;
    input stops;
    input strts;
    input lgyc;
    input cmc;
    input acc;
    input aac;
    input sic;
    input stopc;
    input strtc;
    input strthds;
    input sendahs;
    input sendals;
    input ackhs;
    input ackls;
    input stopsus;
    input stophds;
    input senddhs;
    input senddls;
    input recvdhs;
    input recvdls;
    input addr;
    input [7:0] di;
    input sdai;
    input scli;
    input ce;
    input reset;
    input clk;
```

```

I3C i3c_inst (
    .LGYO(lgyo),
    .CMO(cmo),
    .ACO(aco),
    .AAO(aao),
    .SIO(sio),
    .STOPO(stopo),
    .STRTO(strto),
    .PARITYERROR(parityerror),
    .DOBUF(dobuf),
    .DO(dout),
    .STATE(state),
    .SDAO(sdao),
    .SCL0(scl0),
    .SDAOEN(sdaoen),
    .SCL0EN(scl0en),
    .SDAPULL0(sdapullo),
    .SCLPULL0(sclpullo),
    .SDAPULL0EN(sdapulloen),
    .SCLPULL0EN(sclpulloen),
    .LGYS(lgys),
    .CMS(cms),
    .ACS(acs),
    .AAS(aas),
    .STOPS(stops),
    .STRTS(strts),
    .LGYC(lgyc),
    .CMC(cmc),
    .ACC(acc),
    .AAC(aac),
    .SIC(sic),
    .STOPC(stopc),
    .STRTC(strtc),
    .STRTHDS(strthds),
    .SENDAHS(sendahs),
    .SENDALS(sendals),
    .ACKHS(ackhs),
    .ACKLS(ackls),
    .STOPSUS(stopsus),
    .STOPHDS(stophds),
    .SENDDHS(senddhs),
    .SENDDL0(senddl0),
    .RECVDHS(recvdhs),
    .RECVDL0(recvdl0),
    .ADDRS(addr0),
    .DI(di),
    .SDAI(sdai),
    .SCLI(scli),
    .CE(ce),
    .RESET(reset),
    .CLK(clk)
);

defparam i3c_inst.ADDRESS = 7'b0000000;

endmodule //GW_I3C

module I3C (
    AAC,          //assert ACK clear
    AAO,          //assert ACK output
    AAS,          //assert ACK set
    ACC,          //assert continuity clear
    ACKHS,       //ACK high period divider
    ACKLS,       //ACK low period divider
    ACO,          //assert continuity output
    ACS,          //assert continuity set
    ADDRS,       //set dynamic address

```

```

CE,          //clock enable
CLK,        //clock input
CMC,        //current master set
CMO,        //current master output
CMS,        //current master set
DI,         //data input
DO,         //unbuffered data output
DOBUF,      //buffered data output
LGYC,       //legacy mode clear
LGYO,       //legacy mode output
LGYS,       //enter legacy mode set
PARITYERROR,//indicator of parit bit error
RECVDHS,    //set receiving data high period divider
RECVDLS,    //set receiving data low period divider
RESET,      //asyn.reset, active high
SCLI,       //scl input
SCLO,       //scl output
SCLOEN,     //scl output enable, active low
SCLPULLO,  //scl pull-up output
SCLPULLOEN,//scl pull-up output enable, active low
SDAI,       //sda input
SDAO,       //sda output
SDAOEN,     //sda output enable, active low
SDAPULLO,  //sda pull-up output
SDAPULLOEN,//sda pull-up output enable, active low
SENDAHS,    //set sending address high period divider
SENDALS,    //set sending address low period divider
SENDHDS,    //set sending data high period divider
SENDLDS,    //set sending data low period divider
SIC,        //system interrupt clear
SIO,        //system interrupt output
STRTC,      //start celar
STRTO,      //start output
STRTS,      //start set
STATE,      //state output
STRTHDS,    //set start hold time
STOPC,      //stop clear
STOPO,      //stop output
STOPS,      //stop set
STOPSUS,    //set stop setup time
STOPHDS     //set stop hold time
);

parameter ADDRESS = 7'b0;

input  LGYS, CMS, ACS, AAS, STOPS, STRTS;
output LGYO, CMO, ACO, AAO, SIO, STOPO, STRTO;
input  LGYC, CMC, ACC, AAC, SIC, STOPC, STRTC;
input  STRTHDS, SENDAHS, SENDALS, ACKHS;
input  ACKLS, STOPSUS, STOPHDS, SENDDHS;
input  SENDLDS, RECVDHS, RECVDLS, ADDR;
output PARITYERROR;
input  [7:0] DI;
output [7:0] DOBUF;
output [7:0] DO;
output [7:0] STATE;
input  SDAI, SCLI;
output SDAO, SCLO;
output SDAOEN, SCLOEN;
output SDAPULLO, SCLPULLO;
output SDAPULLOEN, SCLPULLOEN;
input  CE, RESET, CLK;

endmodule

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 I3C 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-113 所示。

图 3-113 用户例化该 IP 设计文件的模板文件

```

GW_I3C your_instance_name(
    .lgyo(lgyo_o), //output lgyo
    .cmo(cmo_o), //output cmo
    .aco(aco_o), //output aco
    .aao(aao_o), //output aao
    .sio(sio_o), //output sio
    .stopo(stopo_o), //output stopo
    .strto(strto_o), //output strto
    .parityerror(parityerror_o), //output parityerror
    .dobuf(dobuf_o), //output [7:0] dobuf
    .dout(dout_o), //output [7:0] dout
    .state(state_o), //output [7:0] state
    .sdao(sdao_o), //output sdao
    .sclo(sclo_o), //output sclo
    .sdaoen(sdaoen_o), //output sdaoen
    .scloen(scloen_o), //output scloen
    .sdapullo(sdapullo_o), //output sdapullo
    .sclpullo(sclpullo_o), //output sclpullo
    .sdapulloen(sdapulloen_o), //output sdapulloen
    .sclpulloen(sclpulloen_o), //output sclpulloen
    .lgys(lgys_i), //input lgys
    .cms(cms_i), //input cms
    .acs(acs_i), //input acs
    .aas(aas_i), //input aas
    .stops(stops_i), //input stops
    .strts(strts_i), //input strts
    .lgyc(lgyc_i), //input lgyc
    .cmc(cmc_i), //input cmc
    .acc(acc_i), //input acc
    .aac(aac_i), //input aac
    .sic(sic_i), //input sic
    .stopc(stopc_i), //input stopc
    .strtc(strtc_i), //input strtc
    .strthds(strthds_i), //input strthds
    .sendahs(sendahs_i), //input sendahs
    .sendals(sendals_i), //input sendals
    .ackhs(ackhs_i), //input ackhs
    .ackls(ackls_i), //input ackls
    .stopsus(stopsus_i), //input stopsus
    .stophds(stophds_i), //input stophds
    .senddhs(senddhs_i), //input senddhs
    .senddls(senddls_i), //input senddls
    .recvdhs(recvdhs_i), //input recvdhs
    .recvdlb(recvdlb_i), //input recvdlb
    .addrb(addrb_i), //input addrb
    .di(di_i), //input [7:0] di
    .sdai(sdai_i), //input sdai
    .scli(scli_i), //input scli
    .ce(ce_i), //input ce
    .reset(reset_i), //input reset
    .clk(clk_i) //input clk
);

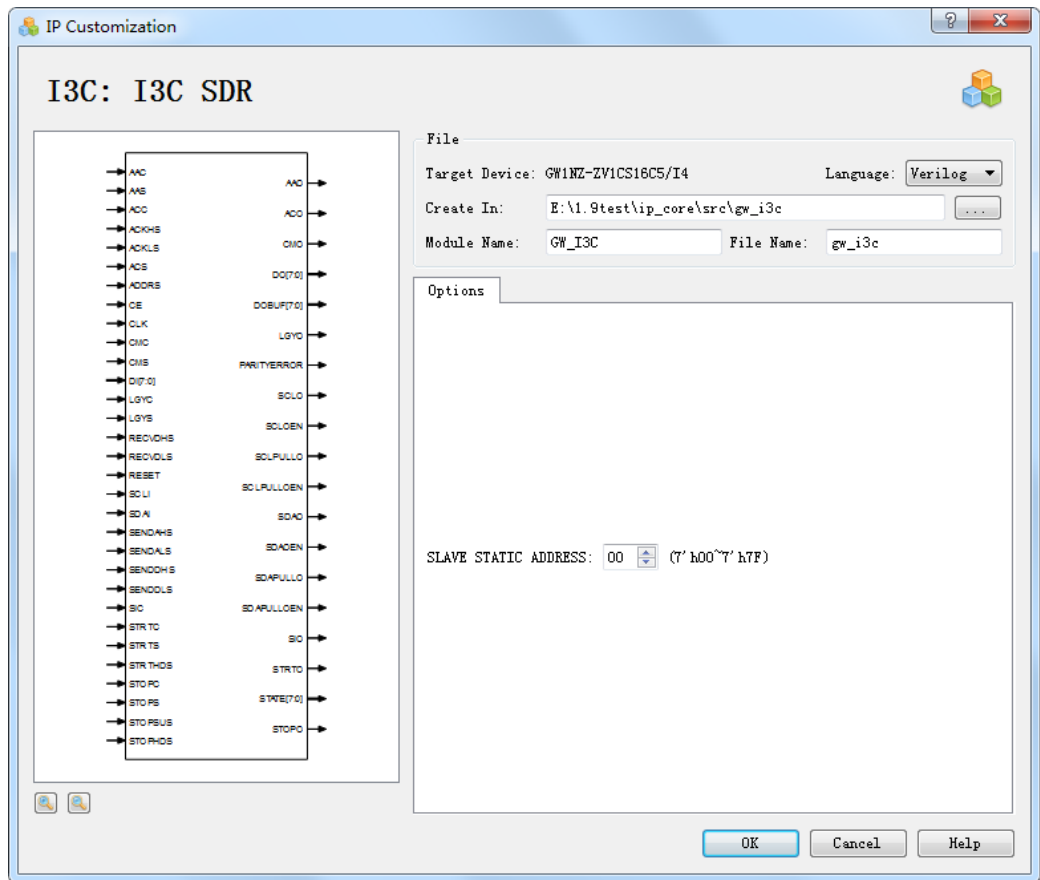
```

IP Core Generator 生成 I3C 示例

以产生 GW1NZ-1 器件支持的 I3C 为例，在 IP 配置界面选择 device 为 GW1NZ-LV1CS16C5/I4，并根据用户要求配置 Options 配置信息，界面配置如图 3-114 示，单击“OK”，产生用户所需的 I3C IP 设计文件。

产生的 I3C IP 设计文件所在目录即为配置界面中“Create In”设置路径。

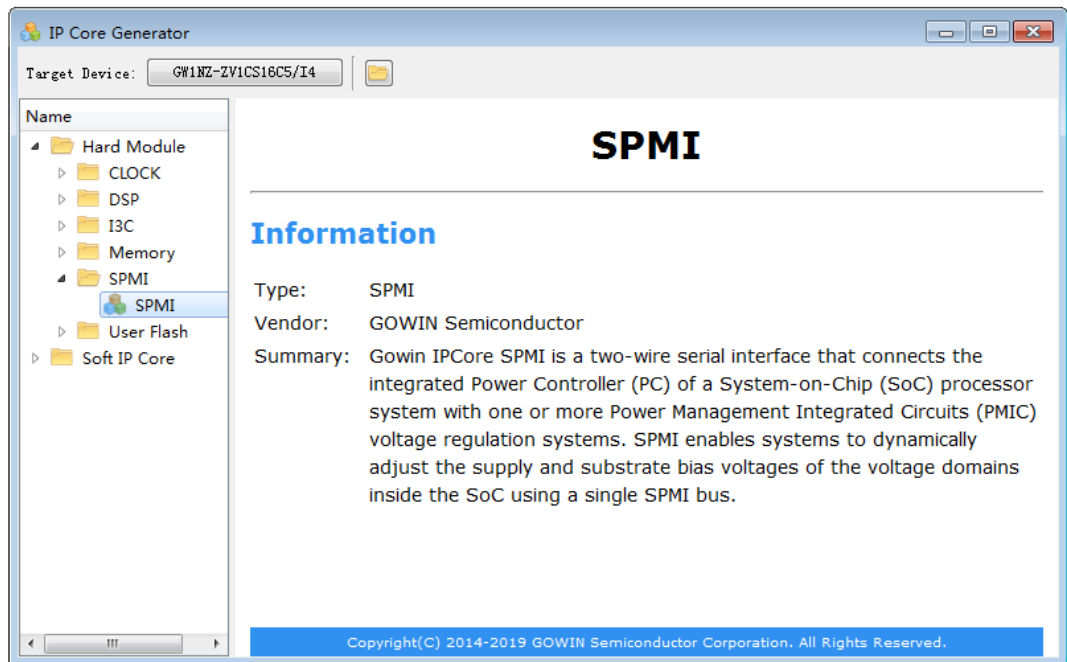
图 3-114 I3C IP Customization 设置



3.6 SPMI

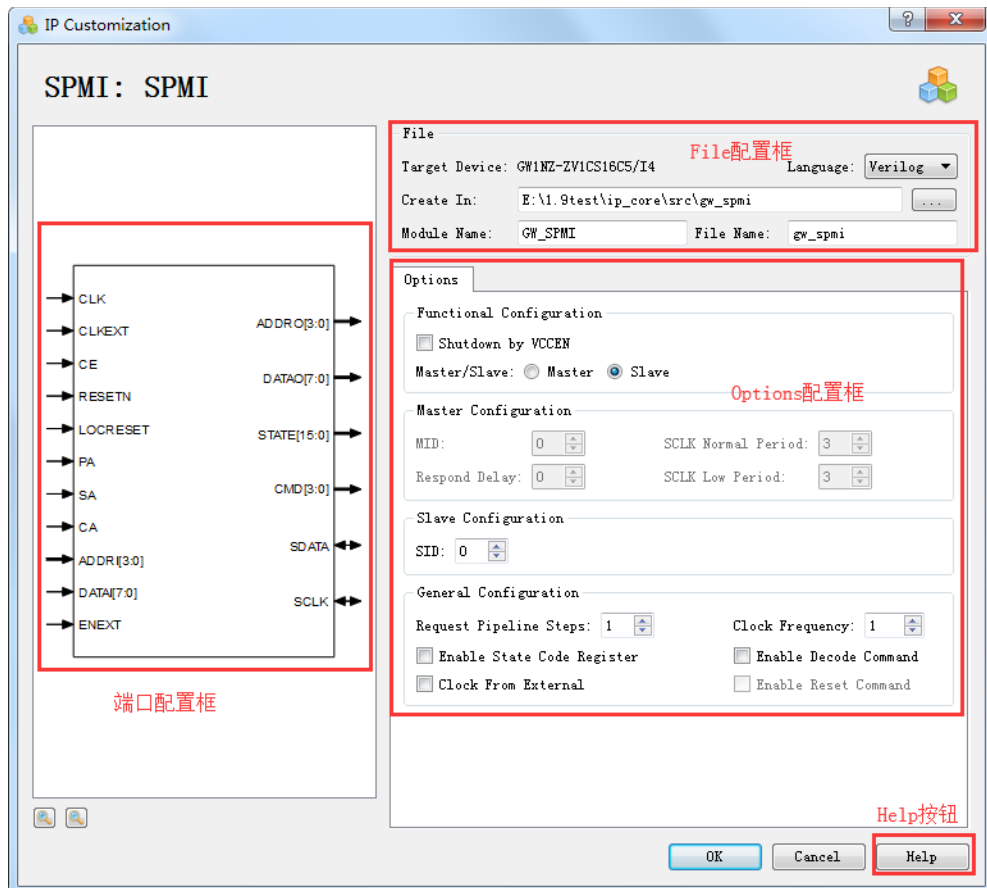
SPMI 是一种双线串行接口，可将片上处理器系统（SoC）的集成电源控制器（PC）与一个或多个电源管理集成电路（PMIC）电压调节系统相连。SPMI 使系统能够使用单个 SPMI 总线动态调整 SoC 内部电压域的电源和衬底偏置电压。在 IP Core Generator 界面中，单击“SPMI”，界面右侧会显示 SPMI 的相关信息概要，如图 3-115 所示。

图 3-115 SPMI 的信息概要



在 IP Core Generator 界面中，双击 SPMI，弹出 SPMI 的“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-116 所示。

图 3-116 SPMI 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 SPMI 实例化文件的相关信息，如图 3-116 中标注的“File 配置框”所示。

SPMI 的 File 配置框的使用和 SP 模块类似，具体请参考 [3.1Block Memory>3.1.1SP](#) 的 File 配置框。

注！

目前仅有 GW1NZ-1 支持 SPMI，如 Target Device 选择其他 device，SPMI 置灰，无法产生对应的 IP。

2. Options 配置框

Options 配置框用于配置例化高云硬核原语 SPMI 设计文件中 SPMI 的配置信息，如图 3-116 中标注的“Options 配置框”所示。

Functional Configuration:

- Shutdown by VCCEN: 通过外部引脚 VCCEN 关闭。如果选择此选项，则 SPMI 的通信功能将不可用。
- Master/Slave: 将 SPMI 设置为主机或从机。

Master Configuration:

- MID: 主机的 ID，设置范围为 0-3，默认值为 0。
- Respond Delay: 设置响应延迟时间。
- SCLK Normal Period: Normal 模式下 sclk 的周期长度。
- SCLK Low Period: 睡眠模式下 sclk 的周期长度。

Slave Configuration:

- SID: 设置 SPMI 从机的 ID。

General configuration:

- Enable State Code Register: 启用或禁用寄存器。例如，如果选择“启用状态代码寄存器”选项，则输出 STATE 数据将通过一个寄存器。
- Request Pipeline Steps: 设置请求信号采样时间的延迟步长。
- Enable Decode Command: 启用或禁用解码。如果选择启用解码命令，SPMI 将解码复位，睡眠，关闭和唤醒命令。
- Enable Reset Command: 启用或禁用重置命令。
- Clock From External: 启用或禁用外部时钟。
- Clock Frequency: 系统时钟频率。

3. 端口配置框

配置框图显示当前 IP Core 的配置结果示例框图，如图 3-116 中标注的“配置框图”所示。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-117 所示。

图 3-117 Help 信息

SPMI

Information

Type:	SPMI
Vendor:	GOWIN Semiconductor
Summary:	Gowin IPCore SPMI is a two-wire serial interface that connects the integrated Power Controller (PC) of a System-on-Chip (SoC) processor system with one or more Power Management Integrated Circuits (PMIC) voltage regulation systems. SPMI enables systems to dynamically adjust the supply and substrate bias voltages of the voltage domains inside the SoC using a single SPMI bus.

Options

Option	Description
Functional Configuration	Shutdown by VCCEN - Shutdown by external pin VCCEN. If choose this option, SPMI's communication function will not be available.
	Master/Slave - Set SPMI to master or slave.
Master Configuration	MID - Set the identifier of the SPMI master
	Respond Delay - Set the response delay time.
	SCLK Normal Period - Set the period of the sclk in normal mode. SCLK Low Period - Set the period of the sclk in sleep mode.
Slave Configuration	SID - Set the identifier of the SPMI slave.
General configuration	Enable State Code Register - Enable or disable registers. For example, If you choose the Enable State Code Register option, the output STATE data will go through one register.
	Request Pipeline Steps - Set the delay step size of the request signal sampling time.
	Enable Decode Command - Enable or disable decoding .If you choose Enable Decode Command, SPMI will decode the reset, sleep, shutdown, and wakeup commands.
	Enable Reset Command - Enable or disable the reset command.
	Clock From External - Enable or disable the external clock. Clock Frequency - System clock frequency

Copyright(C)2014-2018 GOWIN Semiconductor Corporation. All Rights Reserved.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

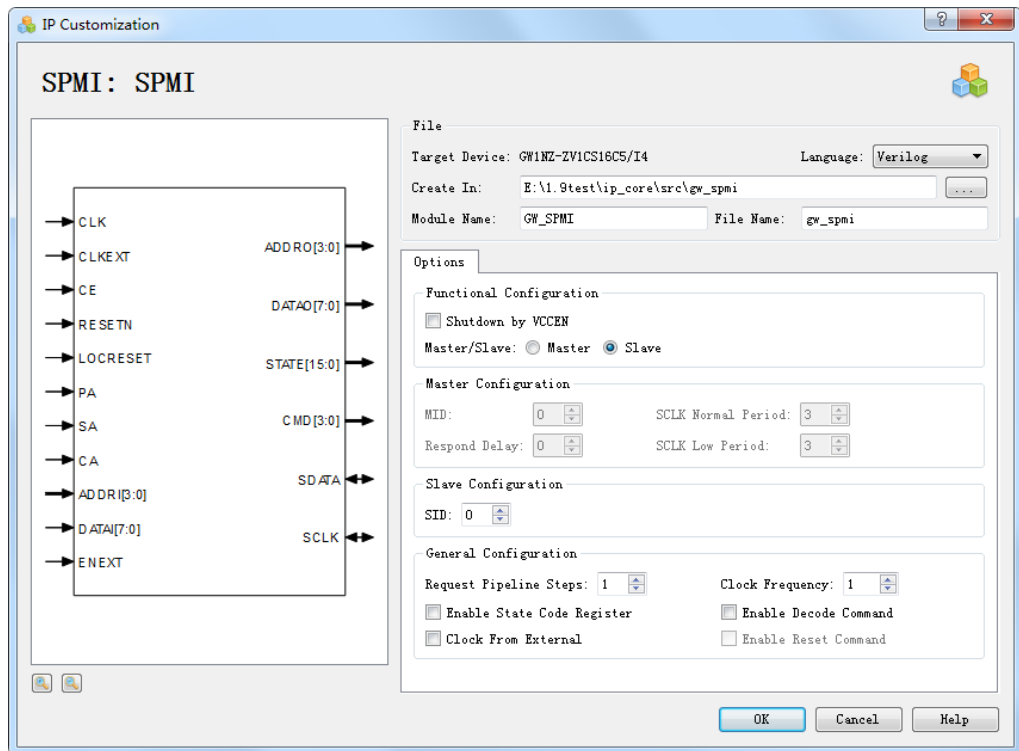
IP 生成文件

如图 3-118 所示，SPMI 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 SPMI 设计文件“gw_spmi.v”；
- 用户例化该 IP 设计文件的模板文件“gw_spmi_tmp.v”；
- 例化原语 SPMI 的配置文件“gw_spmi.ipc”。

下述是 verilog 语言产生的文件。

图 3-118 配置的 IP Customization



例化 SPMI 设计文件

例化 SPMI 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 SPMI 配置，产生实例化的 SPMI，如图 3-119 示。

图 3-119 例化 SPMI 设计文件

```

module GW_SPMI (addro, dataao, state, cmd, sdata, sclk, clk, ce, resetn,
                locreset, pa, sa, ca, addri, datai, clkext, enext);

    output [3:0] addro;
    output [7:0] dataao;
    output [15:0] state;
    output [3:0] cmd;
    inout sdata;
    inout sclk;
    input clk;
    input ce;
    input resetn;
    input locreset;
    input pa;
    input sa;
    input ca;
    input [3:0] addri;
    input [7:0] datai;
    input clkext;
    input enext;

    SPMI spmi_inst (
        .ADDRO(addro),
        .DATAO(dataao),
        .STATE(state),
        .CMD(cmd),
        .SDATA(sdata),
        .SCLK(sclk),
        .CLK(clk),
        .CE(ce),
        .RESETN(resetn),
        .LOCRESET(locreset),
        .PA(pa),
        .SA(sa),
        .CA(ca),
        .ADDRI(addri),
        .DATAI(datai),
        .CLKEXT(clkext),
        .ENEXT(enext)
    );

    defparam spmi_inst.FUNCTION_CTRL = 7'b0000100;
    defparam spmi_inst.MSID_CLKSEL = 7'b0000000;
    defparam spmi_inst.RESPOND_DELAY = 4'b0000;
    defparam spmi_inst.SCLK_NORMAL_PERIOD = 7'b00000011;
    defparam spmi_inst.SCLK_LOW_PERIOD = 7'b00000011;
    defparam spmi_inst.CLK_FREQ = 7'b00000000;
    defparam spmi_inst.SHUTDOWN_BY_ENABLE = 1'b0;

endmodule //GW_SPMI

module SPMI (CLK, CLKEXT, CE, RESETN, ENEXT, LOCRESET, PA, SA, CA, ADDRI,
            DATAI, ADDRO, DATAO, STATE, CMD, SDATA, SCLK)
/* synthesis syn_black_box black_box_pad_pin="SDATA,SCLK" syn_noprune = 1+*/

    parameter FUNCTION_CTRL = 7'b0;
    parameter MSID_CLKSEL = 7'b0;
    parameter RESPOND_DELAY = 4'b0;
    parameter SCLK_NORMAL_PERIOD = 7'b0;
    parameter SCLK_LOW_PERIOD = 7'b0;
    parameter CLK_FREQ = 7'b0;
    parameter SHUTDOWN_BY_ENABLE = 1'b0;

    input CLKEXT, ENEXT;
    inout SDATA, SCLK;
    input CLK, CE, RESETN, LOCRESET;
    input PA, SA, CA;
    input [3:0] ADDRI;
    input [7:0] DATAI;
    output [3:0] ADDRO;
    output [7:0] DATAO;
    output [15:0] STATE;
    output [3:0] CMD;

endmodule

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 SPMI 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-120 所示。

图 3-120 用户例化该 IP 设计文件的模板文件

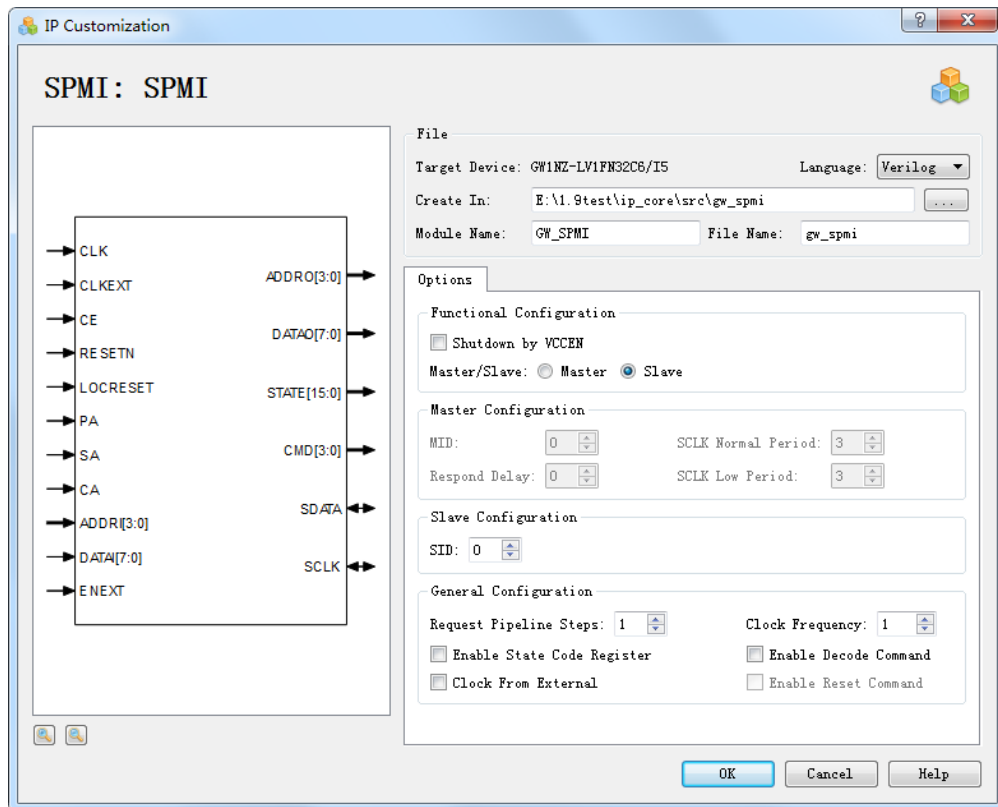
```
GW_SPMI your_instance_name (  
    .addro(addro_o), //output [3:0] addro  
    .datao(datao_o), //output [7:0] datao  
    .state(state_o), //output [15:0] state  
    .cmd(cmd_o), //output [3:0] cmd  
    .sdata(sdata_io), //inout sdata  
    .sclk(sclk_io), //inout sclk  
    .clk(clk_i), //input clk  
    .ce(ce_i), //input ce  
    .resetn(resetn_i), //input resetn  
    .loreset(loreset_i), //input loreset  
    .pa(pa_i), //input pa  
    .sa(sa_i), //input sa  
    .ca(ca_i), //input ca  
    .addri(addri_i), //input [3:0] addri  
    .datai(datai_i), //input [7:0] datai  
    .clkext(clkext_i), //input clkext  
    .enext(enext_i) //input enext  
);
```

IP Core Generator 生成 SPMI 示例

以产生 GW1NZ-1 器件支持的 SPMI 为例，在 IP 配置界面选择 device 为 GW1NZ-LV1FN32C6/I5，并根据用户要求设置 Options 配置信息，界面配置如图 3-121 示，单击“OK”，产生用户所需的 SPMI IP 设计文件。

产生的 SPMI IP 设计文件所在目录即为配置界面中“Create In”设置路径。

图 3-121 SPMI IP Customization 设置



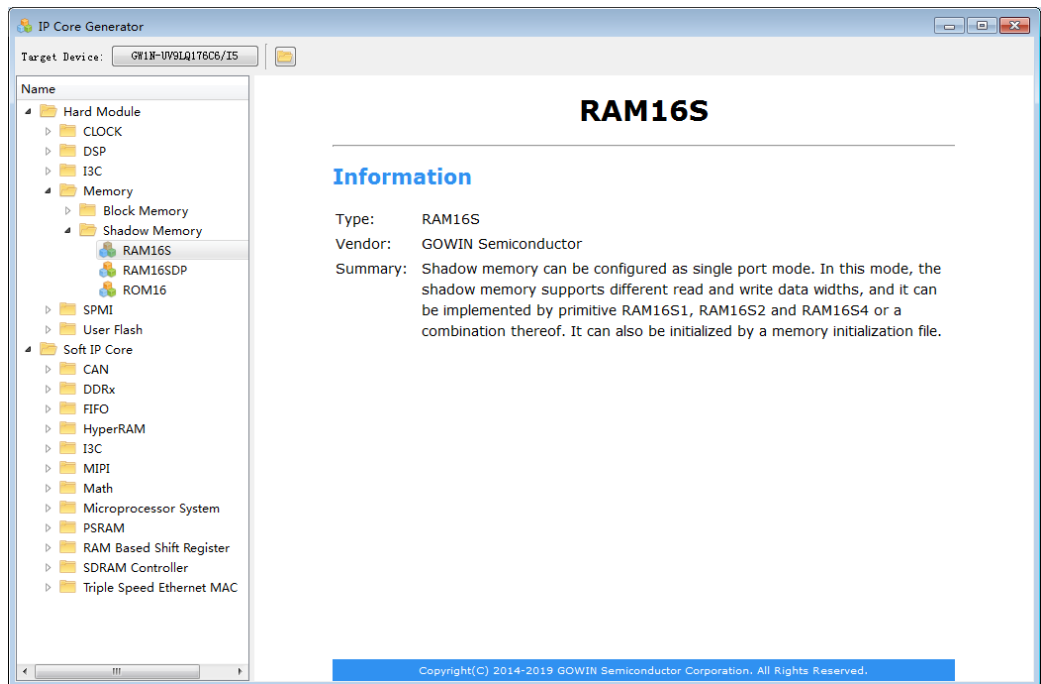
3.7 Shadow Memory

当前, Shadow Memory (SSRAM) 模块可实现 RAM16S (单端口模式)、RAM16SDP (半双端口模式)、ROM16 (只读模式)。

3.7.1 RAM16S

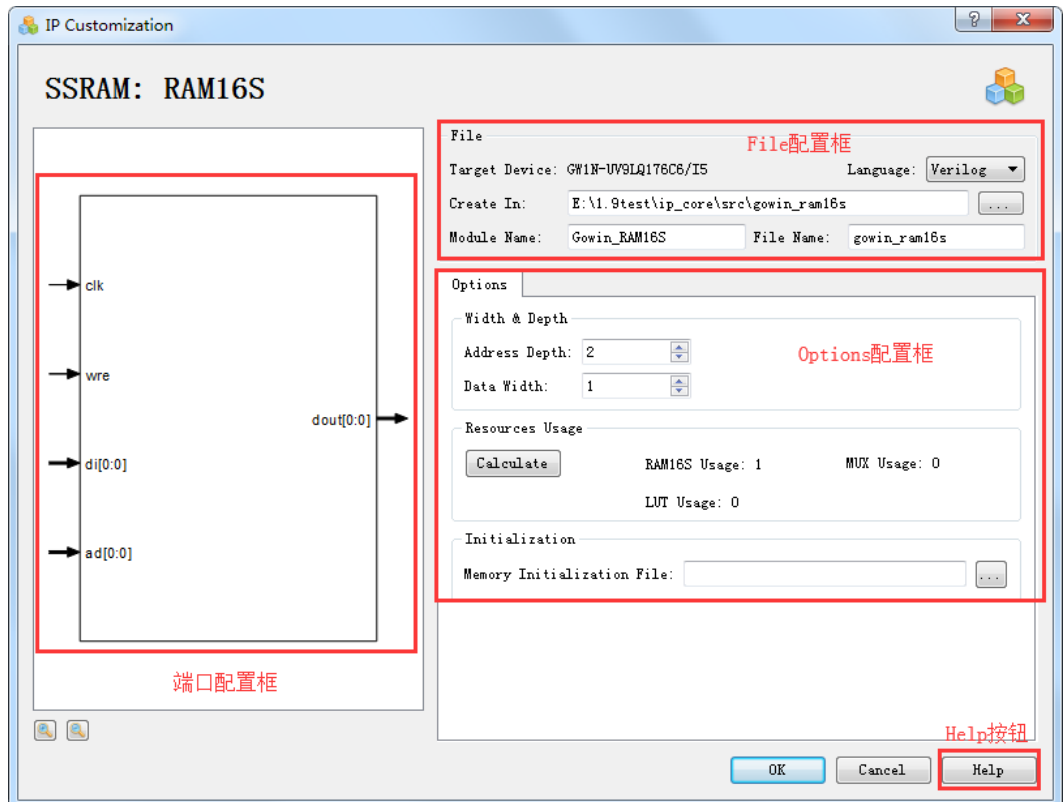
RAM16S 为单端口工作模式, 可以通过 RAM16S1、RAM16S2、RAM16S4 三种高云器件实现。SSRAM 的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中, 单击“RAM16S”, 界面右侧会显示 RAM16S 的相关信息概要, 图 3-122 所示。

图 3-122 RAM16S 的信息概要



在 IP Core Generator 界面中，双击 RAM16S，弹出 RAM16S 的“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如错误!未找到引用源。3 错误!未找到引用源。所示。

图 3-123 RAM16S 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 RAM16S 实例化文件的相关信息，如图 3-123 中标注的“File 配置框”所示。

RAM16S 的 File 配置框的使用和 SP 模块类似，具体请参考 [3.1Block Memory>3.1.1SP 的 File 配置框](#)。

2. Options 配置框

Options 配置框用于用户自定义单端口存储器的配置信息，如图 3-123 中“Options 配置框”所示。

RAM16S 的 Options 配置框的使用和 SP 模块类似，具体请参考 [3.1Block Memory>3.1.1SP 中的 Options 配置框](#)。

3. 端口配置框图

- 配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-123 中标注的“端口配置框”所示；
- Options 配置中的地址深度“Address Depth”配置影响 ad 的位宽，数据位宽“Data Width”配置影响 di 和 dout 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息的页面，如图 3-124 所示。

图 3-124 Help 信息

RAM16S	
Information	
Type:	RAM16S
Vendor:	GOWIN Semiconductor
Summary:	Shadow memory can be configured as single port mode. In this mode, the shadow memory supports different read and write data widths, and it can be implemented by primitive RAM16S1, RAM16S2 and RAM16S4 or a combination thereof. It can also be initialized by a memory initialization file.
Options	
Option	Description
Width & Depth	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the data width.
Resources Usage	Calculate - Calculate the resource usage in the design and display results below.
	RAM16S Usage - Display the number of RAM16S used.
	LUT Usage - Display the number of LUT used.
Initialization	MUX Usage - Display the number of MUX used.
	Memory Initialization File - Set the memory initialization file (.mi) path.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

IP 生成文件

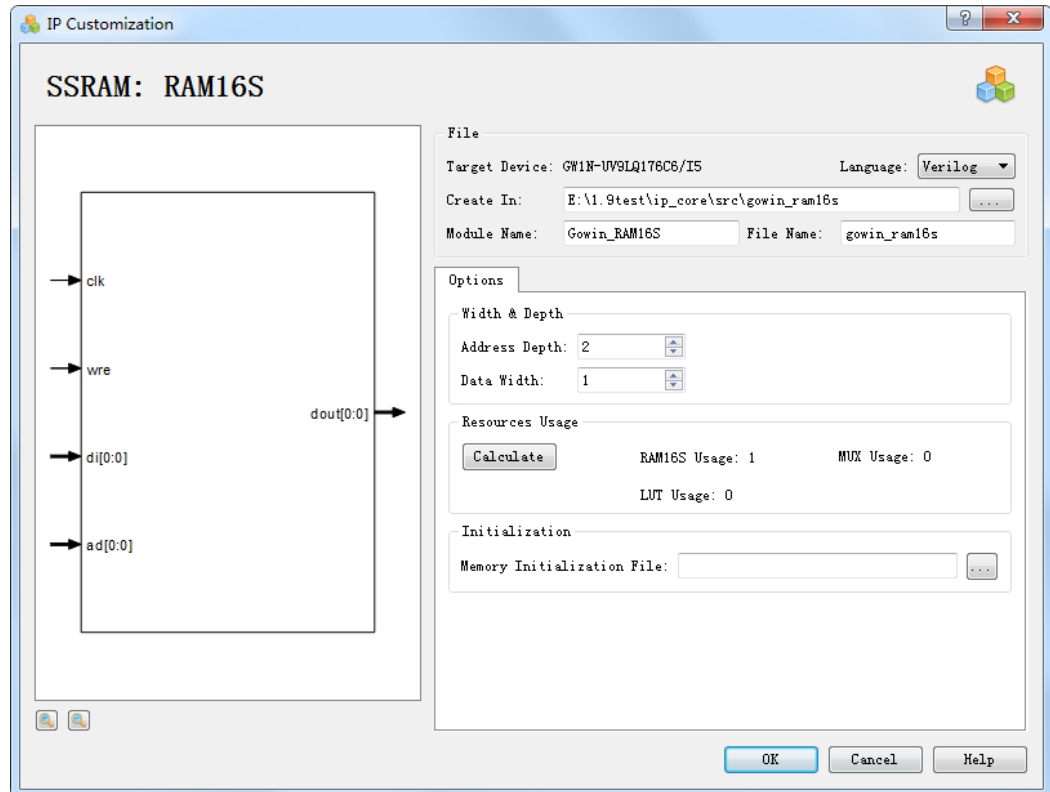
如图 3-125 所示，RAM16S 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 RAM16S 设计文件“gowin_ram16s.v”；
- 用户例化该 RAM16S 设计文件的模板文件“gowin_ram16s_tmp.v”；
- 例化原语 RAM16S 的配置文件“gowin_ram16s.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述

以 verilog 语言为例介绍产生的文件。

图 3-125 配置的 IP Customization



例化 DP 设计文件

例化 RAM16S 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 RAM16S 配置，产生实例化的 RAM16S，如图 3-126 所示。

图 3-126 例化 RAM16S 设计文件

```

module Gowin_RAM16S (dout, di, ad, wre, clk);

    output [0:0] dout;
    input [0:0] di;
    input [0:0] ad;
    input wre;
    input clk;

    wire gw_gnd;

    assign gw_gnd = 1'b0;

    RAM16S1 ssram_spx1_0 (
        .DO(dout[0]),
        .DI(di[0]),
        .AD({gw_gnd,gw_gnd,gw_gnd,ad[0]}),
        .WRE(wre),
        .CLK(clk)
    );

    defparam ssram_spx1_0.INIT_0 = 16'h0000;

endmodule //Gowin_RAM16S

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 RAM16S 设计文件的同时，亦提供用户例化该 RAM16S 设计文件的模板文件，如图 3-127 所示。

图 3-127 用户例化该 IP 设计文件的模板文件

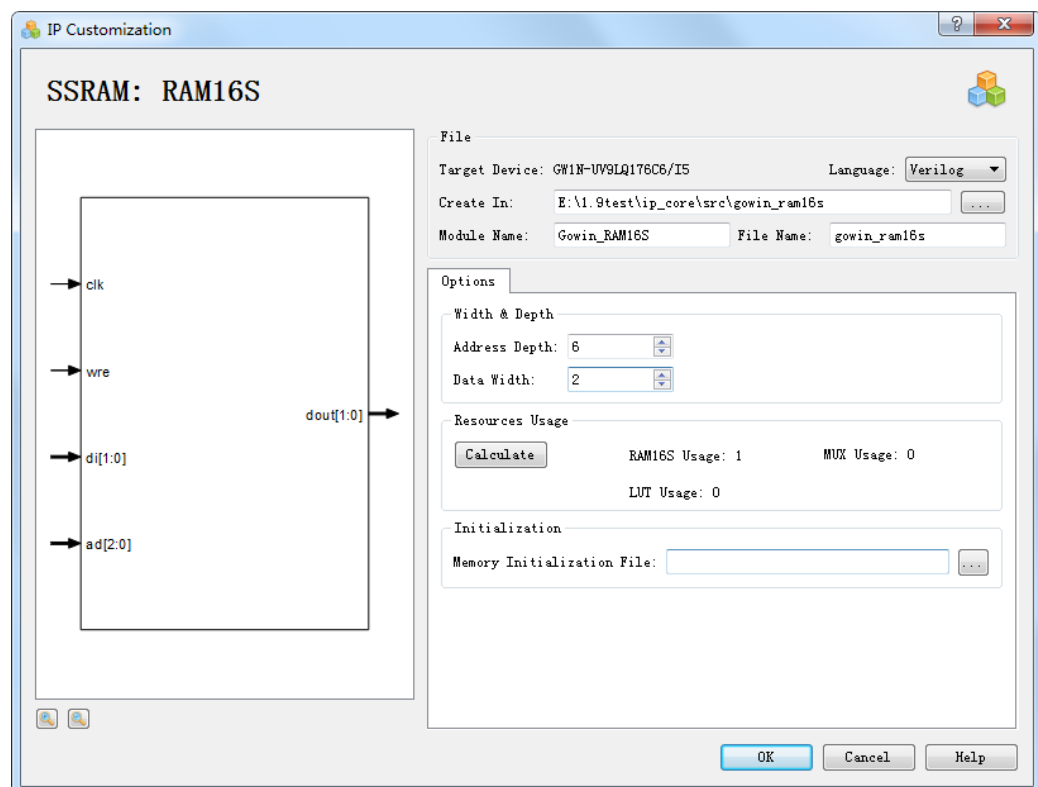
```
Gowin_RAM16S your_instance_name (
    .dout(dout_o), //output [0:0] dout
    .di(di_i), //input [0:0] di
    .ad(ad_i), //input [0:0] ad
    .wre(wre_i), //input wre
    .clk(clk_i) //input clk
);
```

IP Core Generator 生成 RAM16S 示例

如用户需产生地址位宽为 6，数据宽度为 2 的 RAM16S IP，以 device 选择 GW1N-UV9LQ176C6/I5 为例，界面配置如图 3-128 所示，初始化文件可根据用户需要在 Initialization 窗口配置，单击“OK”，产生用户所需的 RAM16S IP 设计文件。

产生的 RAM16S IP 设计文件所在目录即为配置界面中“Create In”设置路径。

图 3-128 RAM16S IP Customization 设置

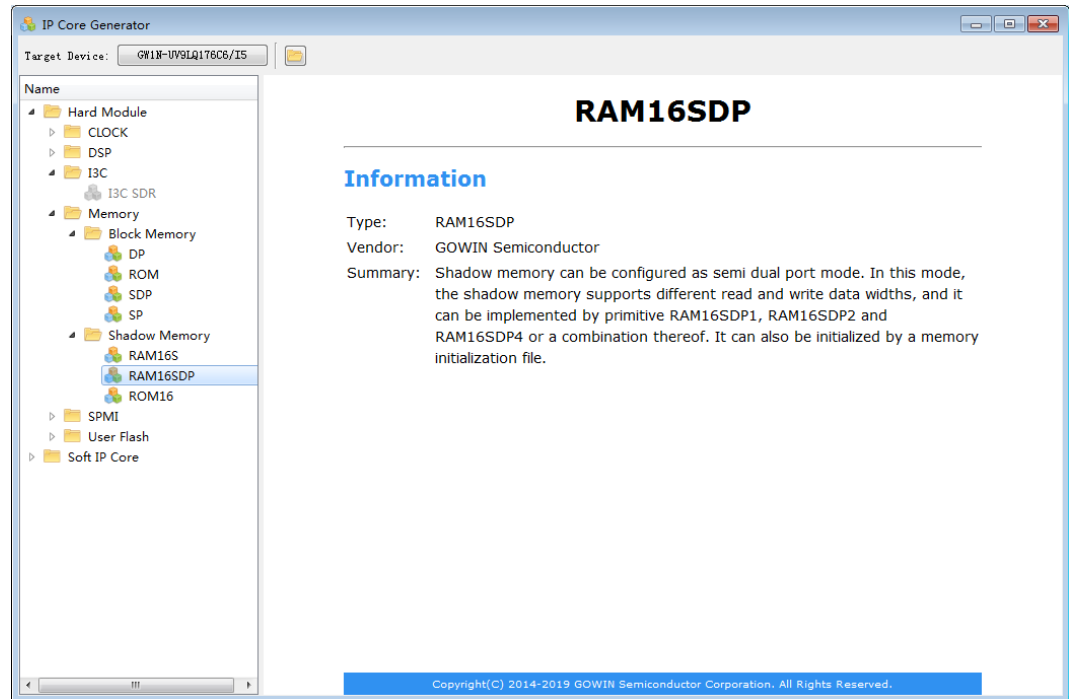


3.7.2 RAM16SDP

RAM16SDP 是半双端口工作模式，可通过 RAM16SDP1、RAM16SDP2、RAM16SDP4 三种高云器件实现。SSRAM 的最大存储容量根据芯片型号的

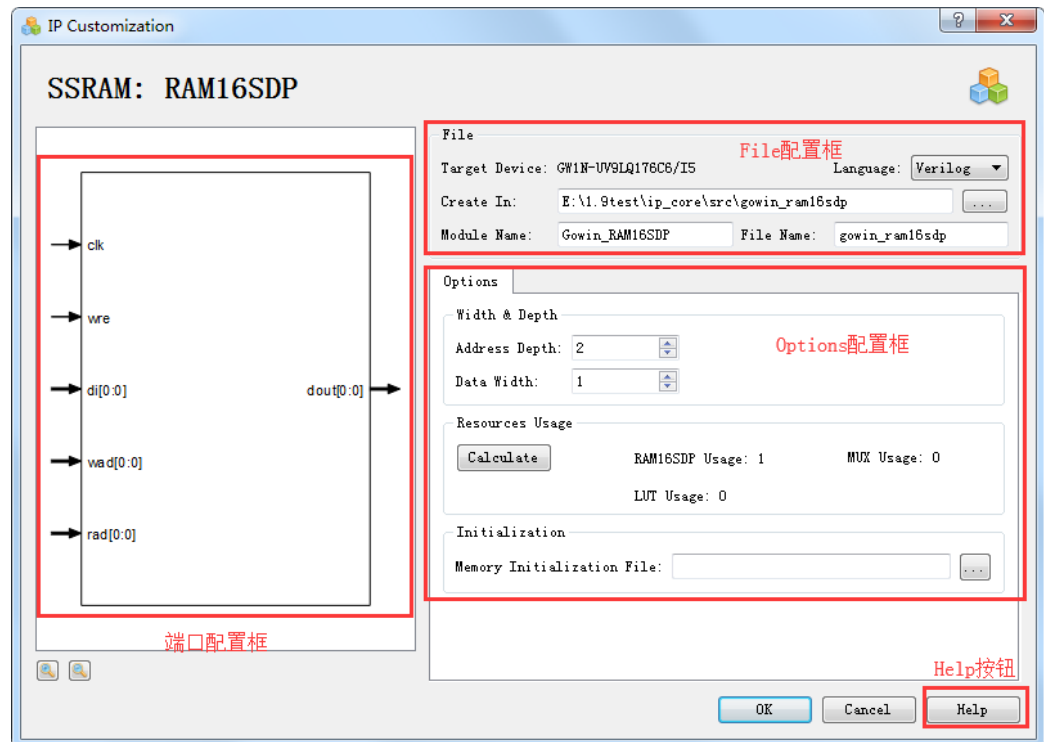
不同而不同。在 IP Core Generator 界面中，单击“RAM16SDP”，界面右侧会显示 RAM16SDP 的相关信息概要，如图 3-129 所示。

图 3-129 RAM16SDP 的信息概要



在 IP Core Generator 界面中，双击“RAM16SDP”，弹出“IP Customization”窗口。该窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-130 所示。

图 3-130 RAM16SDP 的 IP Customization 窗口结构



1. File 配置框

File 配置框用于配置产生 RAM16SDP 实例化文件的相关信息，标注的“File 配置框”如图 3-130 所示。

RAM16SDP 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中 File 配置框介绍。

2. Options 配置框

Options 配置框用于用户自定义半双端口模式存储器的配置信息，如图 3-130 中标注的“Options 配置框”所示。

RAM16SDP 的 Options 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中 Options 配置框。

3. 端口配置框图

- 配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-130 中标注的“配置框图”所示；
- Options 配置中的地址深度“Address Depth”配置影响 wad 和 rad 的位宽，数据位宽“Data Width”配置影响 di 和 dout 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息页面，如图 3-131 所示。

图 3-131 Help 信息

RAM16SDP	
Information	
Type:	RAM16SDP
Vendor:	GOWIN Semiconductor
Summary:	Shadow memory can be configured as semi dual port mode. In this mode, the shadow memory supports different read and write data widths, and it can be implemented by primitive RAM16SDP1, RAM16SDP2 and RAM16SDP4 or a combination thereof. It can also be initialized by a memory initialization file.
Options	
Option	Description
Width & Depth	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the data width.
Resources Usage	Calculate - Calculate the resource usage in the design and display results below.
	RAM16SDP Usage - Display the number of RAM16SDP used.
	LUT Usage - Display the number of LUT used.
Initialization	MUX Usage - Display the number of MUX used.
	Memory Initialization File - Set the memory initialization file (.mi) path.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

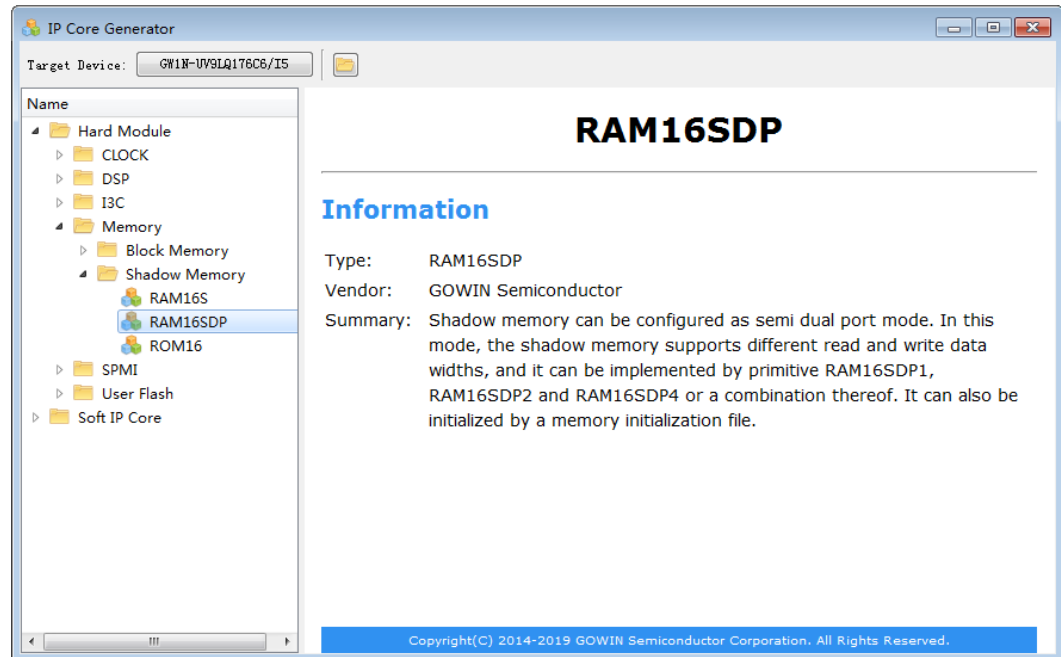
IP 生成文件

如图 3-132 所示，RAM16SDP 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 RAM16SDP 设计文件“gowin_ram16sdp.v”；
- 用户例化该 IP 设计文件的模板文件“gowin_ram16sdp_tmp.v”；
- 例化原语 RAM16SDP 的配置文件“gowin_ram16sdp.ipc”。

如配置中选择的语言是 VHDL，则产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-132 配置的 IP Customization



例化 RAM16SDP 设计文件

例化 RAM16SDP 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 RAM16SDP 配置，产生实例化的 RAM16SDP，如图 3-133 所示。

注！

产生的实例 RAM16SDP 的 di、dout 的数据宽度和“IP Customization”中的 RAM16SDP 配置一致。

图 3-133 例化 RAM16SDP 设计文件

```

module Gowin_RAM16SDP (dout, di, wad, rad, wre, clk);

output [0:0] dout;
input [0:0] di;
input [0:0] wad;
input [0:0] rad;
input wre;
input clk;

wire gw_gnd;

assign gw_gnd = 1'b0;

RAM16SDP1 ssram_sdp1_0 (
    .DO(dout[0]),
    .DI(di[0]),
    .WAD({gw_gnd,gw_gnd,gw_gnd,wad[0]}),
    .RAD({gw_gnd,gw_gnd,gw_gnd,rad[0]}),
    .WRE(wre),
    .CLK(clk)
);

defparam ssram_sdp1_0.INIT_0 = 16'h0000;

endmodule //Gowin_RAM16SDP

```

用户例化该 IP 设计文件的模板文件

考虑用户的实际应用，IP Core Generator 工具在产生例化 RAM16SDP 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-134 所示。

图 3-134 用户例化该 IP 设计文件的模板文件

```

Gowin_RAM16SDP your_instance_name(
    .dout(dout_o), //output [0:0] dout
    .di(di_i), //input [0:0] di
    .wad(wad_i), //input [0:0] wad
    .rad(rad_i), //input [0:0] rad
    .wre(wre_i), //input wre
    .clk(clk_i) //input clk
);

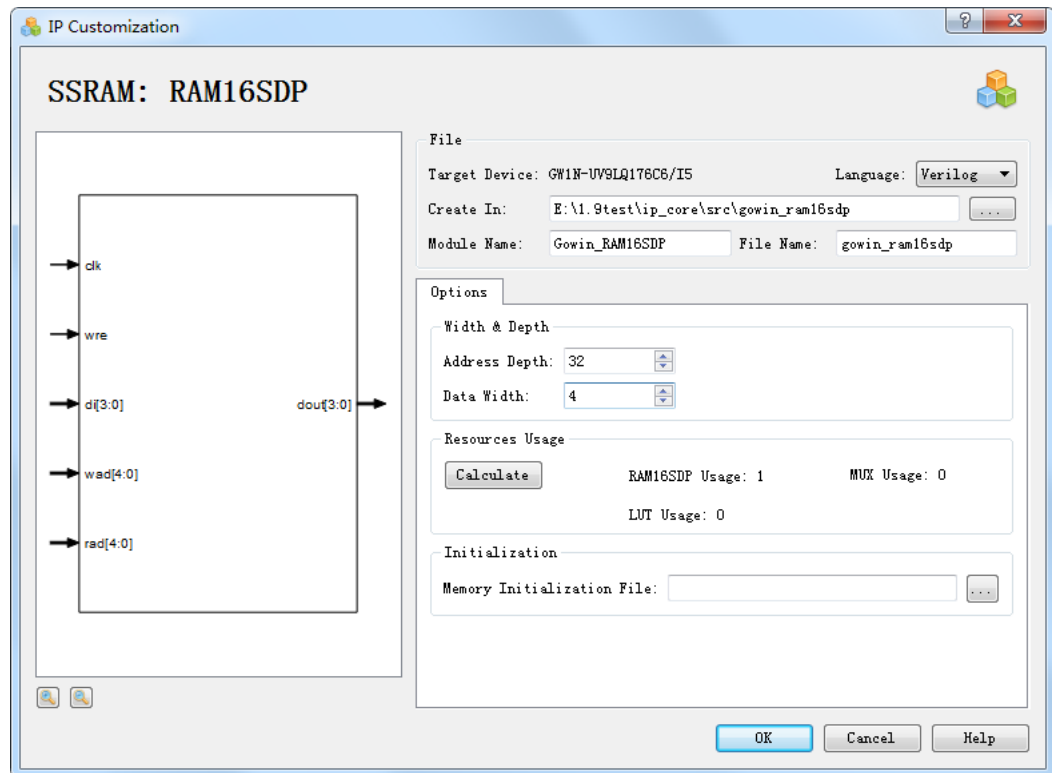
```

IP Core Generator 生成 RAM16SDP 示例

如用户需产生的地址位宽为 32，数据宽度为 4 的 RAM16SDP IP，以 device 选择 GW1N-UV9LQ176C6/I5 为例，界面配置如图 3-135 所示，初始化文件可根据用户需要在 Initialization 窗口配置，单击“OK”，产生用户所需的 RAM16SDP IP 设计文件。

产生的 RAM16SDP IP 设计文件所在目录即为配置界面中“Create In”设置路径。

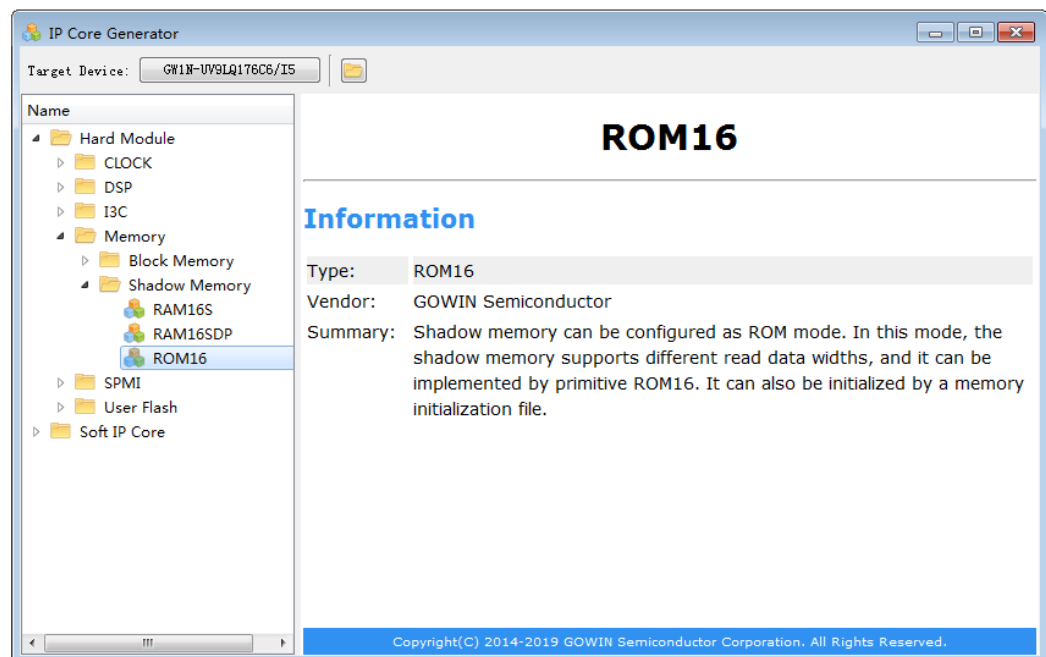
图 3-135 SDP 的 IP Customization 设置



3.7.3 ROM16

ROM16 是只读模式。SSRAM 的最大存储容量根据芯片型号的不同而不同。在 IP Core Generator 界面中，单击“ROM16”，界面右侧会显示 ROM16 的相关信息概要，如图 3-136 所示。

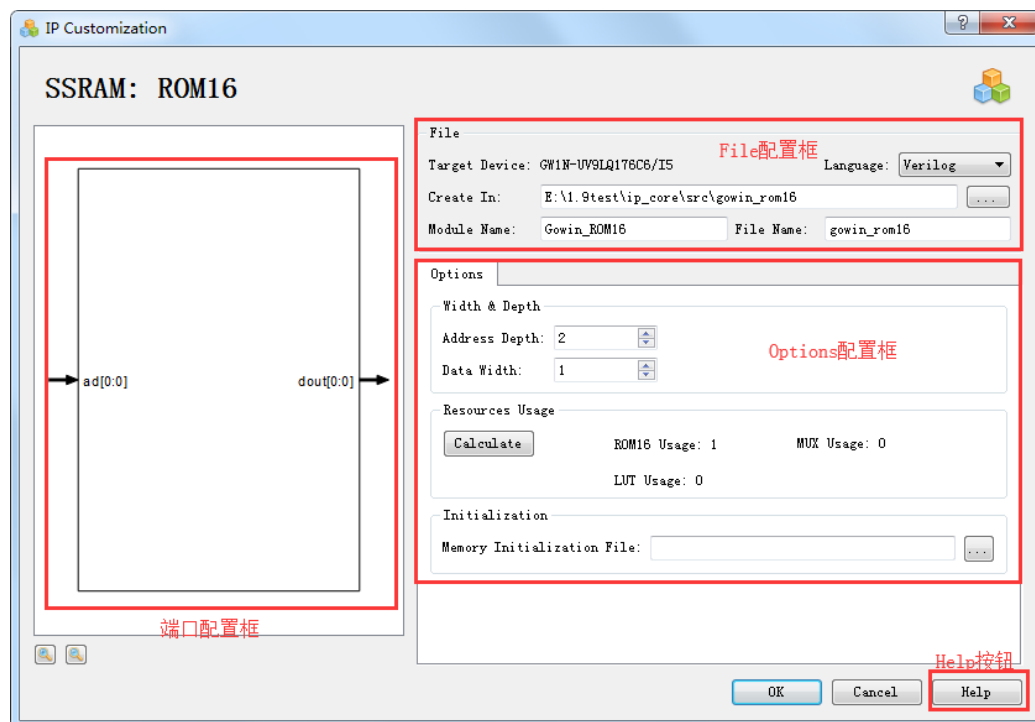
图 3-136 ROM16 的信息概要



在 IP Core Generator 界面中，双击“ROM16”，弹出 ROM16 的“IP

Customization” 窗口。窗口包括 File 配置框、Options 配置框、端口配置框图和帮助按钮“Help”，如图 3-137 所示。

图 3-137 ROM16 的 IP Customization 窗口结构



1. File 配置框

File 配置窗口用于配置产生的 ROM16 实例化文件的相关信息，如图 3-137 中标注的“File 配置框”所示。

ROM16 的 File 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 File 配置框。

2. Options 配置框

Options 配置框用于用户自定义只读端口模式存储器的配置信息，如图 3-137 中标注的“Options 配置框”所示。

ROM16 的 Options 配置框的使用和 SP 模块的类似，请参考 [3.1Block Memory>3.1.1SP](#) 中的 Options 配置。

注！

3. 端口配置框图

配置框图显示当前 IP Core 的配置结果示例框图，输入输出端口的位宽根据 Options 配置实时更新，如图 3-137 中标注的“配置框图”所示；

Options 配置中的地址深度“Address Depth”配置影响 ad 的位宽，数据位宽“Data Width”配置影响 dout 的位宽。

4. Help 按钮

单击“Help”，显示 IP Core 的配置信息页面，如图 3-138 所示。

图 3-138 Help 信息

ROM16	
Information	
Type:	ROM16
Vendor:	GOWIN Semiconductor
Summary:	Shadow memory can be configured as ROM mode. In this mode, the shadow memory supports different read data widths, and it can be implemented by primitive ROM16. It can also be initialized by a memory initialization file.
Options	
Option	Description
Width & Depth	Address Depth - Set the size of the address depth.
	Data Width - Set the size of the data width.
Resources Usage	Calculate - Calculate the resource usage in the design and display results below.
	Block Ram Usage - Display the number of ROM16 used.
	LUT Usage - Display the number of LUT used.
Initialization	MUX Usage - Display the number of MUX used.
	Memory Initialization File - Set the memory initialization file (.mi) path.

Help 页面包括当前 IP Core 的概要介绍，以及 Options 各项配置的简要说明。

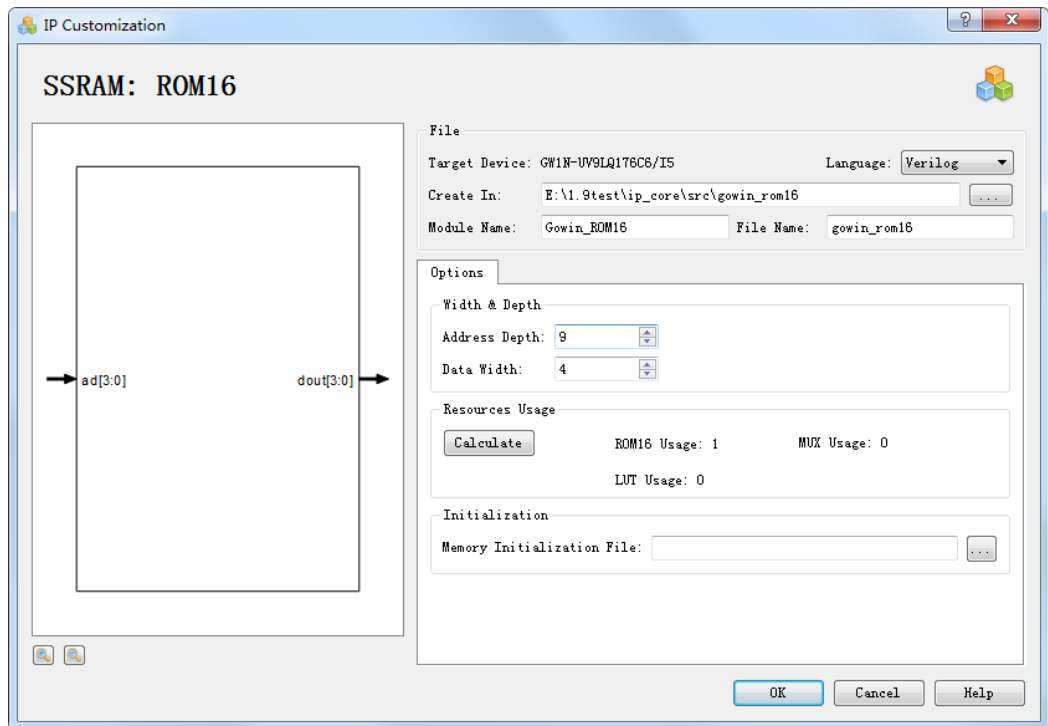
IP 生成文件

如图 3-139 所示，ROM16 的“IP Customization”窗口配置完成后，单击“OK”，产生以配置文件“File Name”命名的三个文件：

- 例化高云原语 ROM16 设计文件“gowin_rom16.v”；
- 用户例化该 IP 设计文件的模板文件“gowin_rom16_tmp.v”；
- 例化原语 ROM16 的配置文件“gowin_rom16.ipc”。

如配置中选择的语言是 VHDL，产生的前两个文件名后缀为.vhd。下述以 verilog 语言为例介绍产生的文件。

图 3-139 配置的 IP Customization



例化 ROM 设计文件

例化 ROM16 设计文件为完整的 verilog 模块，模块中根据“IP Customization”中的 ROM16 配置，产生实例化的 ROM16，如图 3-140 所示。

注！

产生的实例 ROM16 的 dout 的数据宽度和“IP Customization”中的 ROM16 配置一致。

图 3-140 例化 ROM16 设计文件

```

module Gowin_ROM16 (dout, ad);

output [3:0] dout;
input [3:0] ad;

ROM16 ssram_rom_0 (
    .DO(dout[0]),
    .AD(ad[3:0])
);

defparam ssram_rom_0.INIT_0 = 16'h0000;

ROM16 ssram_rom_1 (
    .DO(dout[1]),
    .AD(ad[3:0])
);

defparam ssram_rom_1.INIT_0 = 16'h0000;

ROM16 ssram_rom_2 (
    .DO(dout[2]),
    .AD(ad[3:0])
);

defparam ssram_rom_2.INIT_0 = 16'h0000;

ROM16 ssram_rom_3 (
    .DO(dout[3]),
    .AD(ad[3:0])
);

defparam ssram_rom_3.INIT_0 = 16'h0000;

endmodule //Gowin_ROM16

```

用户例化 IP 设计文件的模板文件

IP Core Generator 工具考虑用户的实际应用，在产生例化 ROM16 设计文件的同时，亦提供用户例化该 IP 设计文件的模板文件，如图 3-141 所示。

图 3-141 用户例化 IP 设计文件的模板文件

```

Gowin_ROM16 your_instance_name (
    .dout(dout_o), //output [3:0] dout
    .ad(ad_i) //input [3:0] ad
);

```

IP Core Generator 生成 ROM16 示例

如用户需产生地址位宽为 32，数据宽度为 4 的 ROM16 IP，以 device 选择 GW1N-UV9LQ176C6/I5 为例，界面配置如图 3-142 所示，可根据用户需要，在 Initialization 窗口配置初始化文件，单击“OK”，产生用户所需的 ROM16 IP 设计文件。

产生的 ROM16 IP 设计文件所在目录即为配置界面中“Create In”设置路径。

图 3-142 ROM16 的 IP Customization 设置

