Gowin I2C Master & Slave

用户指南

IPUG504-1.1,2018-01-04

版本信息

日期	版本	说明
2017/10/10	1.0	初始版本。
2018/01/04	1.1	I2C 开源给用户，删除 GUI 和 IP 等说明。

目录

图目录

表目录

1 关于本手册

1.1 手册内容

Gowin I2C Master 和 Slave 用户指南主要包括功能简介、信号定义、工作原理、实例化等，旨在帮助用户快速了解 Gowin I2C Master 和 Slave 的特性及使用方法。

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-4、GW1N-6、GW1N-9
2. GW1NR 系列 FPGA 产品：GW1NR-4、GW1NR-9
3. GW2A 系列 FPGA 产品：GW2A-18、GW2A-55
4. GW2AR 系列 FPGA 产品：GW2AR-18

1.3 相关文档

通过登录高云半导体网站 http://www.gowinsemi.com.cn/可以下载、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. GW2A 系列 FPGA 产品数据手册
4. GW2AR 系列 FPGA 产品数据手册
5. Gowin 云源软件用户指南

1.4 术语、缩略语

本手册中出现的相关术语、缩略语及相关释义如表 1-1 所示。

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
AXI	Advanced Extensible Interface	高级可扩展接口
I2C Bus	Inter-Integrated Circuit Bus	I2C 串行总线

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：http://www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel：+86 755 8262 0391

2 功能简介

2.1 概述

I2C 总线是一种两线式串行总线，用于连接微控制器及其外围设备。I2C 总线具有简单、有效等显著优点。由于接口直接在组件上，因此 I2C 总线占用空间小，从而减少电路板的空间和芯片管脚的数量，降低互联成本。

Gowin I2C Master 现支持与带 AXI4-Lite 总线的处理器相连接。提供一种低速、双线、串行总线接口，接口通过数据引脚（SDA）和时钟引脚（SCL）连接到 I2C 总线，以完成数据的传输及外围器件的扩展。允许连接到标准（高达 100kHz）或快速（高达 400kHz）的 I2C 总线。

Gowin I2C Slave 遵循 I2C 总线协议，主要用于与 Master 通信。

2.2 特性

Gowin I2C Master

- 符合业界标准的 I2C 总线协议；
- 总线仲裁及仲裁丢失检测；
- 总线忙状态检测；
- 产生中断标志；
- 支持 I2C 不同的通信模式：
  - 标准模式(100kbps)
  - 快速模式(400kbps)
  - 快速(+)模式(1Mbps)
  - 高速模式(3.4Mbps)
- 产生起始、终止、重复起始和应答信息；
- 支持起始、终止和重复起始检测；
- 支持 7 位寻址模式。

Gowin I2C Slave

- 符合业界标准的 I2C 协议；
- 接收/发送数据功能；
- 支持中断产生；
- 支持 RAM 和 ROM 两种工作模式。

3 信号定义

3.1 Gowin I2C Master 信号定义

3.1.1 AXI4-Lite 总线侧信号

表 3-1 AXI4-Lite 总线侧信号定义

序号	信号名称	方向	描述	备注
1	S_AXI_ACLK	I	工作时钟，上升沿采样	-
2	S_AXI_ARESETN	I	复位信号	-
3	AXI_INTA_O	O	中断信号	-
4	S_AXI_AWADDR	I	写地址信号	AXI4-Lite 写地址通道信号
5	S_AXI_AWVALID	I	写地址有效信号	
6	S_AXI_AWREADY	O	写地址准备	
7	S_AXI_ARADDR	I	读地址信号	AXI4-Lite 读地址通道信号
8	S_AXI_ARVALID	I	读地址有效信号	
9	S_AXI_ARREADY	O	读地址准备信号	
10	S_AXI_WDATA	I	写数据信号	AXI4-Lite 写通道信号
11	S_AXI_WSTRB	I	写选通信号	
12	S_AXI_WVALID	I	写有效信号	
13	S_AXI_WREADY	O	写准备信号	
14	S_AXI_RDATA	O	读数据信号	AXI4-Lite 读通道信号
15	S_AXI_RRESP	O	读响应信号	
16	S_AXI_RVALID	O	读有效信号	
17	S_AXI_RREADY	I	读准备信号	
18	S_AXI_BRESP	O	写响应信号	AXI4-Lite 写响应通道信号
19	S_AXI_BVALID	I	写响应有效信号	
20	S_AXI_BREADY	I	写响应准备	

3.1.2 I2C 总线侧信号

表 3-2 I2C 总线侧信号定义

序号	信号名称	方向	描述	备注
1	SCL	IO	串行时钟线	-
2	SDA	IO	串行数据线	-

3.2 Gowin I2C Slave 信号定义

表 3-3 Gowin I2C Slave 信号定义

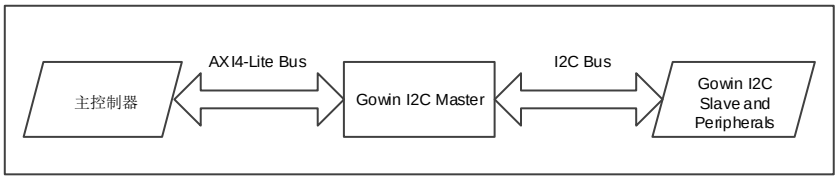
序号	信号名称	方向	描述	备注
1	clk	input	时钟信号	-
2	rst	input	复位信号	-
3	scl	input	串行时钟线	-
4	sda	inout	串行数据线	-
5	int_o	output	中断信号	-

4 工作原理

4.1 系统框图

如图 4-1 所示，主控制器将指令或数据通过 AXI4-Lite 总线传送给 Gowin I2C Master，然后 Gowin I2C Master 通过 I2C 总线下发给 Gowin I2C Slave and Peripherals，或将 Gowin I2C Slave and Peripherals 数据通过 AXI4-Lite 总线上传给主控制器。

图 4-1 系统框图



4.2 I2C 寄存器

Gowin I2C Master 共有 6 个 8 位宽的寄存器：

- 预分频寄存器
- 控制寄存器
- 发送寄存器
- 接收寄存器
- 指令寄存器
- 状态寄存器

注！

- 发送寄存器和接收寄存器地址相同，均为 0x03；
- 指令寄存器和状态寄存器地址相同，均为 0x04，详细信息如表 4-1 所示。

表 4-1 Gowin I2C Master 寄存器

寄存器名称	寄存器地址	寄存器位宽	类型	描述
Prescale_reg0	0x00	8	读/写	时钟预分频寄存器的低 8 位。
Prescale_reg1	0x01	8	读/写	时钟预分频寄存器的高 8 位。
Control_reg	0x02	8	读/写	控制寄存器
Transmit_reg	0x03	8	写	发送寄存器
Receive_reg	0x03	8	读	接收寄存器
Command_reg	0x04	8	写	指令寄存器
Status_reg	0x04	8	读	状态寄存器

4.2.1 时钟预分频寄存器

时钟预分频寄存器位宽为 16bit，由两个 8 位宽的寄存器组成。预分频寄存器中的数值用于实现对 I2C Master 主时钟进行分频。I2C Master 主时钟为 5*SCL，可由公式[master clock frequency / (5*sclk frequency) -1]计算得出预分频寄存器中数值。时钟预分频寄存器的结构和描述如图 4-2 和表 4-3 所示。

图 4-2 预分频寄存器

15	0
Prescale data	

表 4-2 预分频寄存器

比特	名称	默认值	访问类型	描述
15:0	Prescale data	0	读/写	16 比特预分频数据。

4.2.2 控制寄存器

控制寄存器位宽为 8bit。只使用两个比特([7:6])，其余 6 个比特（[5:0]）用作保留位。

控制寄存器的最高位用于使能 I2C Master：

- 若该位为 0，则 I2C Master 不工作；
- 若为 1，则 I2C Master 进入工作状态。

控制寄存器的第 6 位是中断使能控制位，如该位为 1，则使能中断。

控制寄存器的结构和描述如图 4-3 和表 4-3 所示。

图 4-3 控制寄存器

7	6	5	0
EN	IEN	保留	

表 4-3 控制寄存器

比特	名称	默认值	访问类型	描述
7	EN	0	读/写	I2C Master 使能位： ● “0”=不使能 Master； ● “1”=使能 Master。
6	IEN	0	读/写	I2C Master 中断使能位： ● “0”=不使能 Master 中断； ● “1”=使能 Master 中断。
5:0	保留	N/A	读/写	保留

4.2.3 发送寄存器

发送寄存器用于存储主机通过 I2C 总线发送给从机的数据，当发送地址时，第 0 位代表读写信号。发送寄存器的结果和描述如图 4-4 和表 4-4 所示。

图 4-4 发送寄存器

7 1	0
TX DATA	LSB/RW

表 4-4 发送寄存器

比特	名称	默认值	访问类型	描述
7:1	TX DATA	不确定	写	需通过 I2C 总线发送的数据。
0	LSB/RW	不确定	写	● 发送数据的最低位； ● 在发送从机地址时，为读写控制位：“1”为读；“0”为写。

4.2.4 接收寄存器

接收寄存器存储主机通过 I2C 总线由从机读取的数据，该寄存器的结构和描述如图 4-5 和表 4-5 所示。

图 4-5 接收寄存器

7	0
RX DATA	

表 4-5 接收寄存器

比特	名称	默认值	访问类型	描述
7:0	RX DATA	不确定	读	通过 I2C 接收的字节数据。

4.2.5 状态寄存器

状态寄存器描述了 I2C 总线接口的状态。复位时，状态寄存器被清零。状态寄存器的结构和描述如图 4-6 和表 4-6 所示。

图 4-6 状态寄存器

7	6	5	4:2	1	0
RX ACK	Busy	AL	保留	TIP	IF

表 4-6 状态寄存器

比特	名称	默认值	访问类型	描述
7	RX ACK	0	读	来自从机的应答信号。 ● “1”=主机没有接收到应答； ● “0”=主机接收到应答信号
6	Busy	0	读	I2C 总线忙碌信号。 ● “1”=忙碌； ● “0”=空闲。
5	AL	0	读	仲裁丢失信号。 当检测到不合要求的停止信号，或主机驱动 SDA 为高而 SDA 为低时，该位为“1”。
4:2	保留	N/A	读	保留
1	TIP	0	读	传输进行标识位： ● “1”=正在进行传输； ● “0”=传输结束
0	IF	0	读	中断标志位。 如果 IEN 为“1”，则该位置位，有中断发生时，会导致中断请求。 当仲裁丢失或一个字节的传输结束时，该位置“1”。

4.2.6 指令寄存器

Gowin I2C Master 通过写指令寄存器来配置 I2C 的操作模式。指令寄存器存储了 I2C 下一次操作的指令，每次操作完成后，指令寄存器自动清除。因此每次进行开始，读写，停止操作时，AXI4-Lite 总线都要重新写指令寄存器。指令寄存器的结构和描述如图 4-7 和表 4-7 所示。

图 4-7 指令寄存器

7	6	5	4	3	2 1	0
STA	STO	RD	WR	ACK	保留	IACK

表 4-7 指令寄存器

比特	名称	默认值	访问类型	描述
7	STA	0	写	(重复)开始
6	STO	0	写	停止
5	RD	0	写	读
4	WR	0	写	写
3	ACK	0	写	应答
2:1	保留	0	写	保留
0	IACK	0	写	中断应答。 当置位时，清除等待的中断。

4.3 基本操作流程

Gowin I2C Master 支持通用的 I2C 操作，下面主要对 I2C 写和读操作进行介绍。

4.3.1 I2C 主机总线初始化

向时钟预分频寄存器写入预定的值。该值由时钟频率和 I2C 总线的速度决定。

向控制寄存器写入 8'h80，使能 I2C Master。

4.3.2 主机写数据

1. 设置发送寄存器的值：Slave address + Write bit，例如：{7'b1000110,1'b0}，其中 7'b1000110 即 Slave address，Write bit 为 1'b0；
2. 设置指令寄存器的值为 8'h90，使能起始和写命令，使 I2C 总线开始传输数据；
3. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
4. 设置发送寄存器的值为从机内存的地址，主机发送的数据会写入相应地址的内存；
5. 设置指令寄存器的值为 8'h10，使能写命令，来发送从机内存的地址；
6. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
7. 设置发送寄存器为 8 比特的数据，该数据会写入从机；
8. 设置指令寄存器的值为 8'h10，使能写命令，来发送数据；
9. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
10. 重复步骤 7 到 9，不断向从机写数据；
11. 设置发送寄存器为最后 1 个字节的数据；
12. 设置指令寄存器的值为 8'h50，使能写命令来发送最后一个字节的数据，然后发送终止命令。

4.3.3 主机读数据

1. 设置发送寄存器的值：Slave address + Write bit；
2. 设置指令寄存器的值为 8’h90，使能起始和写命令，使 I2C 总线开始传输数据；
3. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
4. 设置发送寄存器的值为从机内存的地址，主机从该内存中读取数据；
5. 设置指令寄存器的值为 8’h10，使能写命令，来发送从机内存的地址；
6. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
7. 设置发送寄存器的值：Slave address + Read bit，例如：{7’b1000110,1’b1}，其中 7’b1000110 即 Slave address，Read bit 为 1’b1；
8. 设置指令寄存器的值为 8’h90，使能起始（这种情况是重复起始）和写，将发送寄存器中的数据写到从机；
9. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
10. 设置命令寄存器的值为 8’h20，来执行读和应答命令，完成由从机读数据；
11. 检查状态寄存器的 TIP 位，以确保命令执行完毕；
12. 重复 10~11 步骤，继续由从机读数据；
13. 当主机停止由从机读数据时，设置命令寄存器的值为 8’h68，由从机读取最后一个字节的数据，执行非应答。

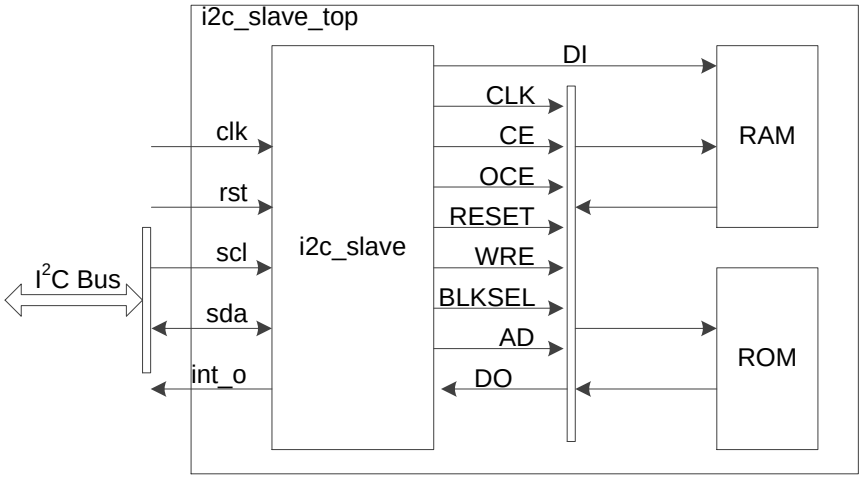
4.4 Gowin I2C Slave 工作方式

Gowin I2C Slave 主要包括 RAM 和 ROM 两种工作方式。若选择 RAM 方式，Gowin I2C Slave 可存储 I2C Master 发送过来的数据，或将 RAM 中的数据传输给 I2C Master；若选择 ROM 方式，I2C Master 只能读取 ROM 事先存储的数据，目前，ROM 事先存储的为 0~255 共 256 个数据。

Gowin I2C Slave 支持中断功能。若选择 RAM 方式，当写满 RAM 或读空 RAM 时，Gowin I2C Slave 会产生中断信号 int_o。Gowin I2C Slave 主要包括两种中断方式，若选择 0，则写停止，读出写过的数据；若选择 1，则写停止，读停止，回到初始状态。若选择 ROM 方式，Master 只进行读，因此不会出现写满读空的情况。

Gowin I2C Slave 主要包括 i2c_slave_top module 和 i2c_slave module。i2c_slave_top module 主要用于实现 i2c_slave module、RAM module、ROM module 之间的互连；i2c_slave module 主要用于驱动 RAM module 或 ROM module，进行数据的存储或发送。Gowin I2C Slave IP 整体架构如图 4-8 所示。

图 4-8 Gowin I2C Slave 架构图



5 应用举例

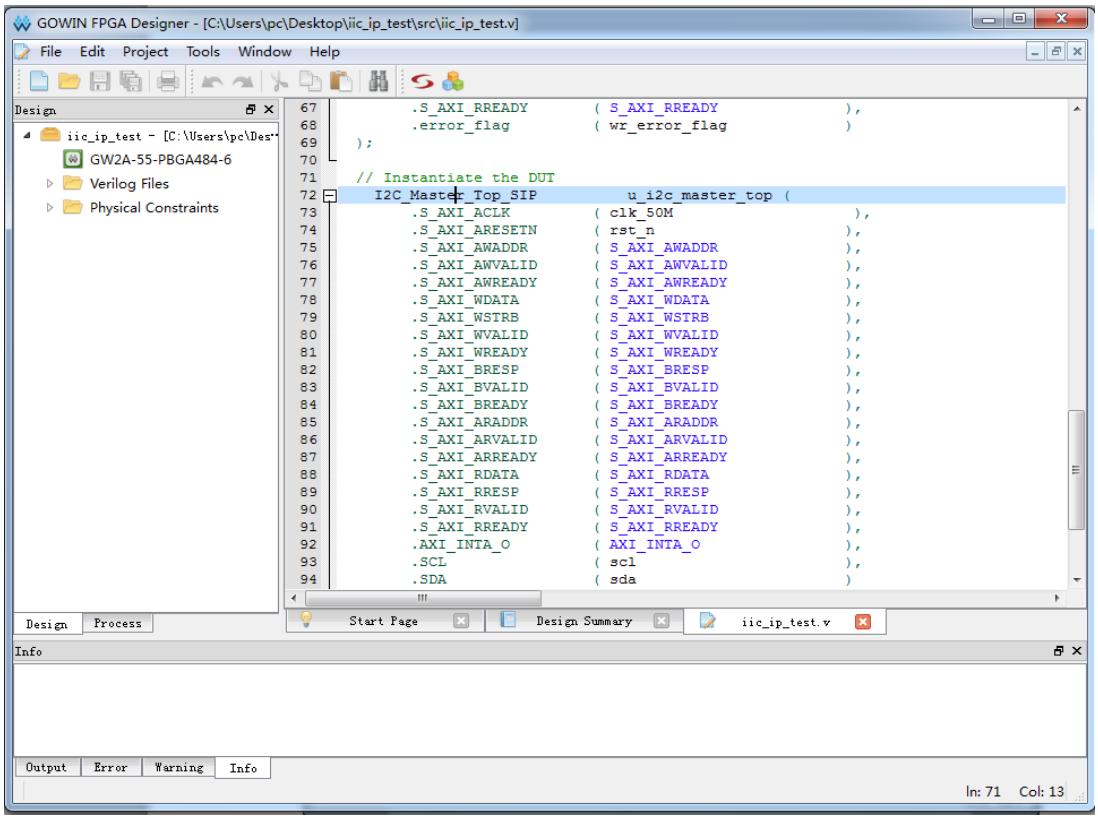
5.1 打开工程

启动 Gowin 云源软件后，单击“File> Open …”，打开“Open File”对话框，选择所需工程文件（*.gprj），打开工程，如图 5-1 所示。

注！

有三种方式打开工程，其它打开工程方式请参考*《Gowin 云源软件用户指南》> 5 云源软件使用> 5.2 打开工程*。

图 5-1 打开工程

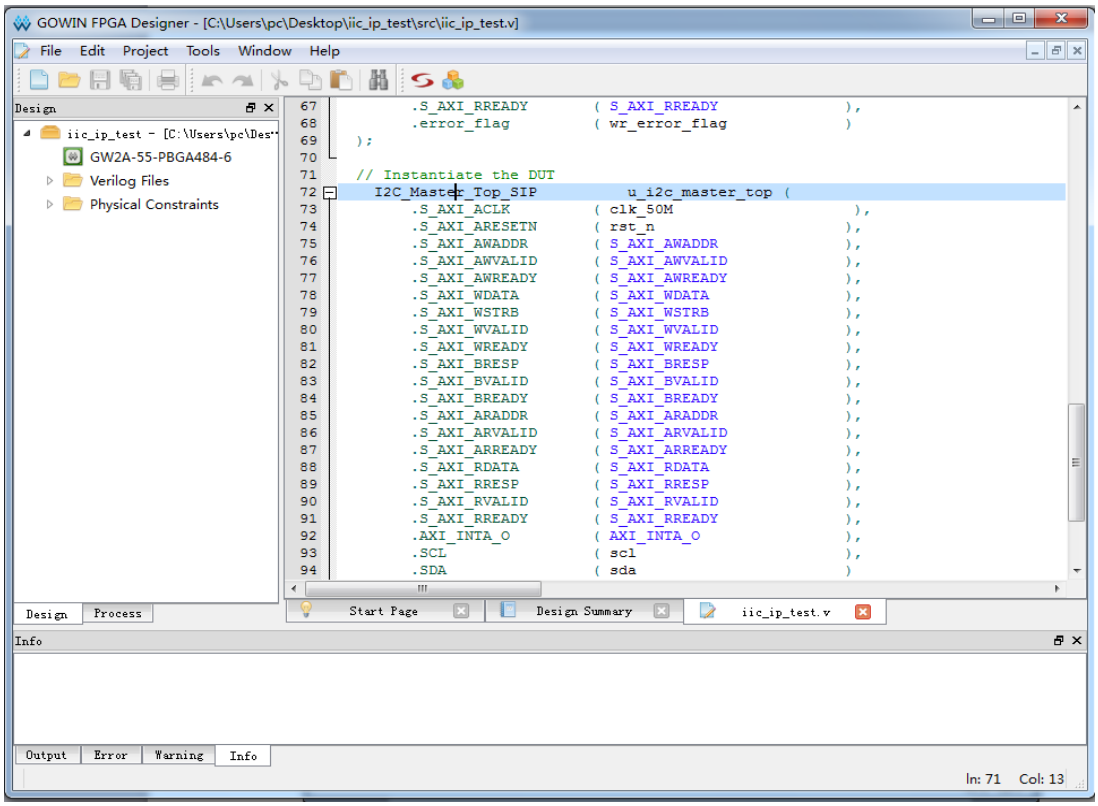


5.2 例化 Gowin I2C Master 和 Slave

5.2.1 例化 Gowin I2C Master

在工程中加载 Gowin I2C Master 相关源文件后例化 I2C_Master_Top_SIP，如图 5-2 所示。

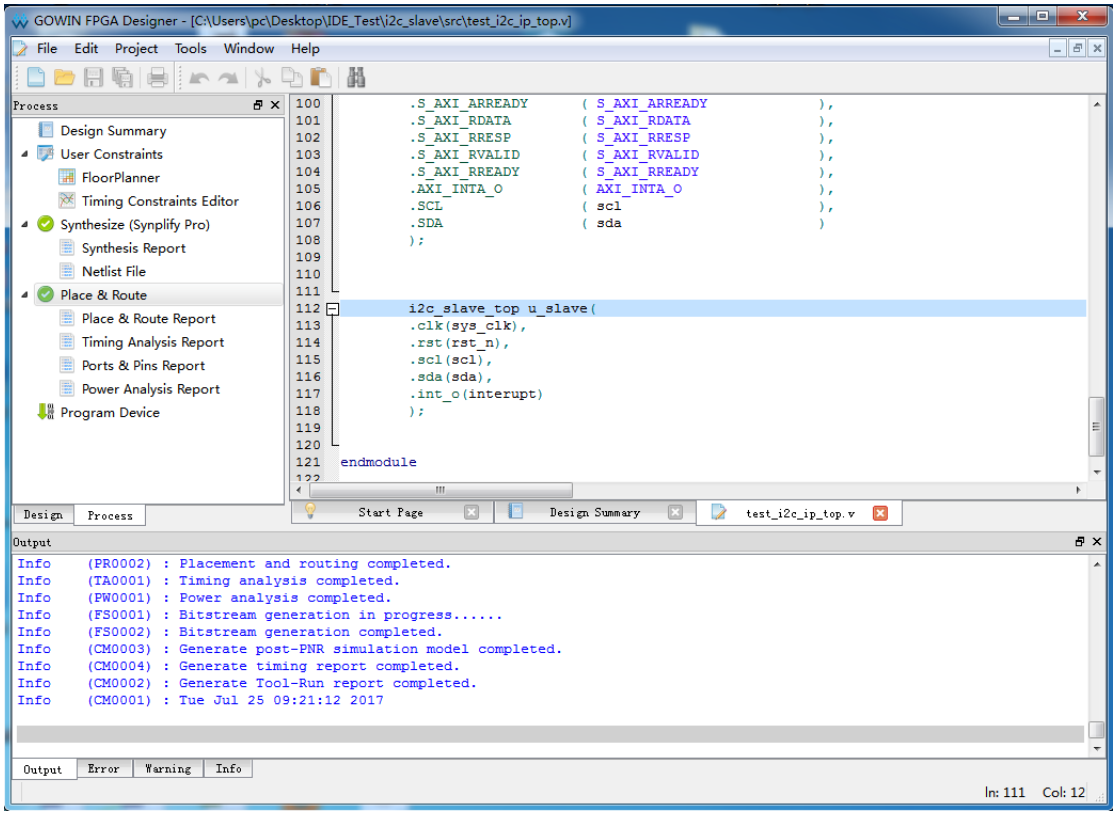
图 5-2 例化 Gowin I2C Master



5.2.2 例化 Gowin I2C Slave

在工程中加载 Gowin I2C Slave 相关源文件后例化 Gowin I2C Slave，如图 5-3 所示。

图 5-3 例化 Gowin I2C Slave



5.3 生成 bitstream 文件

进行必要的约束后，通过综合、布局布线产生 bitstream 文件。通过 Gowin 下载线将 bitstream 文件下载至开发板或测试板，可通过测试接口观测通信情况。


GOWIN高云
智慧逻辑 定制未来