



Gowin SPI_UART IP 用户指南

IPUG926-1.0,2020-04-15

版权所有© 2020 广东高云半导体科技股份有限公司

未经本公司书面许可，任何单位和个人都不得擅自摘抄、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
2020/04/15	1.0	初始版本。

目录

目录	i
图目录	iii
表目录	iv
1 关于本手册	1
1.1 手册内容	1
1.2 相关文档	1
1.3 术语、缩略语	1
1.4 技术支持与反馈	1
2 概述	2
2.1 特征	2
2.2 性能	2
2.3 资源占用	2
3 信号定义	4
3.1 系统信号	4
3.2 SPI 侧信号	4
3.3 UART 侧信号	4
4 参数配置	5
5 功能描述	6
5.1 系统框图	6
5.2 整体结构	7
5.3 SPI_UART IP 寄存器	7
5.3.1 RHR 寄存器	8
5.3.2 THR 寄存器	8
5.3.3 IER 寄存器	8
5.3.4 IIR 寄存器	8
5.3.5 LSR 寄存器	9
5.3.6 MSR 寄存器	9
5.3.7 MCR 寄存器	10
5.3.8 FCR 寄存器	10
5.3.9 TXLVL 寄存器	11
5.3.10 RXLVL 寄存器	11

5.3.11 DLL 寄存器.....	11
5.3.12 DLH 寄存器.....	11
5.4 基本操作流程.....	11
5.4.1 初始化流程.....	11
5.4.2 写操作流程.....	12
5.4.3 读操作流程.....	12
5.5 接口时序.....	12
5.5.1 UART 接口时序.....	12
5.5.2 SPI 接口寄存器写时序.....	13
5.5.3 SPI 接口寄存器读时序.....	13
5.5.4 SPI 接口读 LSR/MSR 清中断时序.....	13
5.5.5 SPI 接口读 RHR 清中断时序.....	13
5.5.6 SPI 接口写 THR 清中断时序.....	14
6 界面配置.....	15
7 参考设计.....	17

图目录

图 5-1 系统框图	6
图 5-2 整体结构	7
图 5-3 UART 接口时序	12
图 5-4 SPI 接口寄存器写时序	13
图 5-5 SPI 接口寄存器读时序	13
图 5-6 SPI 接口读 LSR/MSR 清中断时序	13
图 5-7 SPI 接口读 RHR 清中断时序	13
图 5-8 SPI 接口写 THR 清中断时序	14
图 6-1 打开工程	15
图 6-2 SPI_UART 配置界面	16
图 6-3 例化 SPI_UART	16

表目录

表 1-1 术语、缩略语	1
表 2-1 资源占用	3
表 3-1 系统信号	4
表 3-2 SPI 侧信号	4
表 3-3 UART 侧信号	4
表 5-1 Gowin SPI_UART IP 寄存器	7
表 5-2 RHR 寄存器	8
表 5-3 THR 寄存器	8
表 5-4 IER 寄存器	8
表 5-5 IIR 寄存器	8
表 5-6 LSR 寄存器	9
表 5-7 MSR 寄存器	9
表 5-8 MCR 寄存器	10
表 5-9 FCR 寄存器	10
表 5-10 TXLVL 寄存器	11
表 5-11 RXLVL 寄存器	11
表 5-12 DLL 寄存器	11
表 5-13 DLH 寄存器	11

1 关于本手册

1.1 手册内容

SPI_UART 用户手册主要包括性能特征、信号定义、参数介绍、整体框图、功能描述、GUI 调用、参考设计等，旨在帮助用户快速了解高云半导体 SPI_UART 的产品特性和使用方法，加速用户产品开发

1.2 相关文档

通过登录高云半导体网站 www.gowinsemi.com.cn 可以下载、查看 FPGA 产品文档

1.3 术语、缩略语

表 1-1 中列出了本手册中出现的相关术语、缩略语及相关释义。

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
SPI	Serial Peripheral Interface	串行外设接口
UART	Universal Asynchronous Receiver/Transmitter	通用异步收发传输器

1.4 技术支持与反馈

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

网址：<http://www.gowinsemi.com.cn>

E-mail：support@gowinsemi.com

Tel: +86 755 8262 0391

2 概述

SPI (Serial Peripheral Interface) 是一种高速、全双工、同步的通信总线。

UART (Universal Asynchronous Receiver/Transmitter), 通用异步收发传输器。是电脑硬件的一部分, 将资料由串行通信与并行通信间作传输转换。

为了更好地服务 FPGA 用户, 降低用户的系统开发难度, 提高产品开发速度, 设计一款 SPI_UART IP, 实现 SPI 与 UART 间的接口转换功能。

2.1 特征

- 实现 1 路 SPI 与 UART 接口之间的数据通信;
- SPI 接口为四线从端模式;
- UART 接口为四线模式, 1 停止位, 无校验, 8 位宽度;
- UART 总线波特率可通过设置寄存器灵活设置;
- 支持中断方式;
- UART 收 FIFO 深度 64Bytes;
- UART 发 FIFO 深度 64Bytes;
- 输入工作时钟必须不小于输入 SPI 数据速率的 6 倍;
- 完全可综合;
- 设计语言为 Verilog。

2.2 性能

SPI_UART 的工作频率取决于 SPI 主端提供的时钟频率和 IP 在所选芯片中支持的最大工作频率。

2.3 资源占用

以高云 GW2A-LV18PG256C7/I6 为例, 其资源占用情况如表 2-1 所示, 有关在其他高云 FPGA 上的应用验证, 请关注后期发布信息

表 2-1 资源占用

芯片型号	编程语言	LUT4 资源	REG 资源
GW2A-LV18PG256C7/I6	Verilog	503	263

3 信号定义

3.1 系统信号

表 3-1 系统信号

序号	信号名称	方向	描述	备注
1	l_rst_n	I	复位	所有信号输入输出方向均以 IP 为参考
2	l_clk	I	时钟	

3.2 SPI 侧信号

表 3-2 SPI 侧信号

序号	信号名称	方向	描述	备注
1	l_spi_cs_n	I	SPI 片选	所有信号输入输出方向均以 IP 为参考
2	l_spi_clk	I	SPI 时钟	
3	O_spi_so	O	SPI 输出数据	
4	l_spi_si	I	SPI 输入数据	
5	O_spi_irq_n	O	中断请求信号	

3.3 UART 侧信号

表 3-3 UART 侧信号

序号	信号名称	方向	描述	备注
1	l_uart_rxd	I	输入串口数据	所有信号输入输出方向均以 IP 为参考
2	O_uart_txd	O	输出串口数据	
3	O_uart_rts_n	O	允许发送信号	
4	l_uart_cts_n	I	对端准备好信号	

4 参数配置

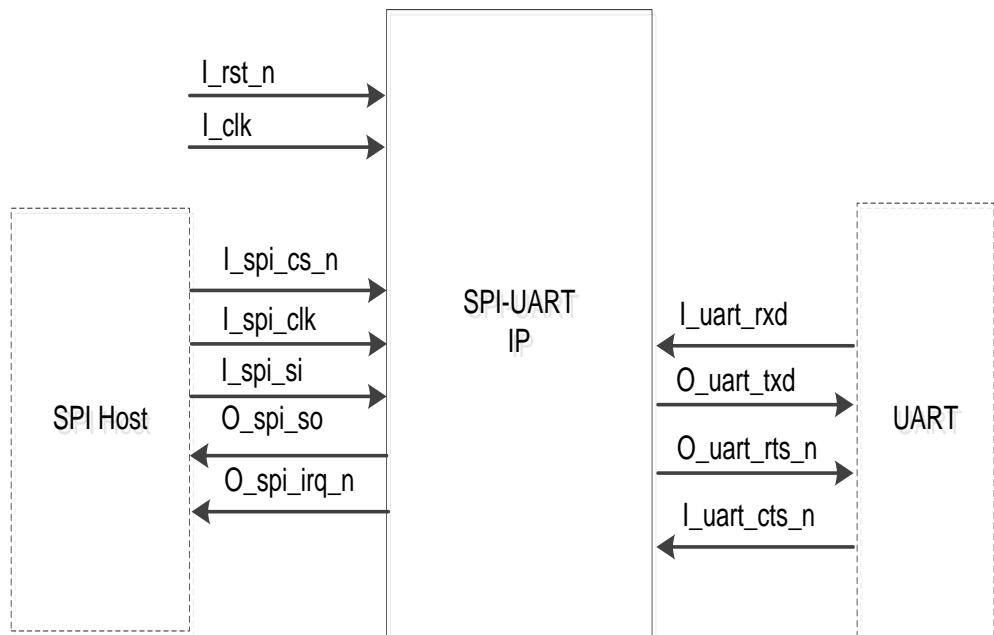
TBD

5 功能描述

5.1 系统框图

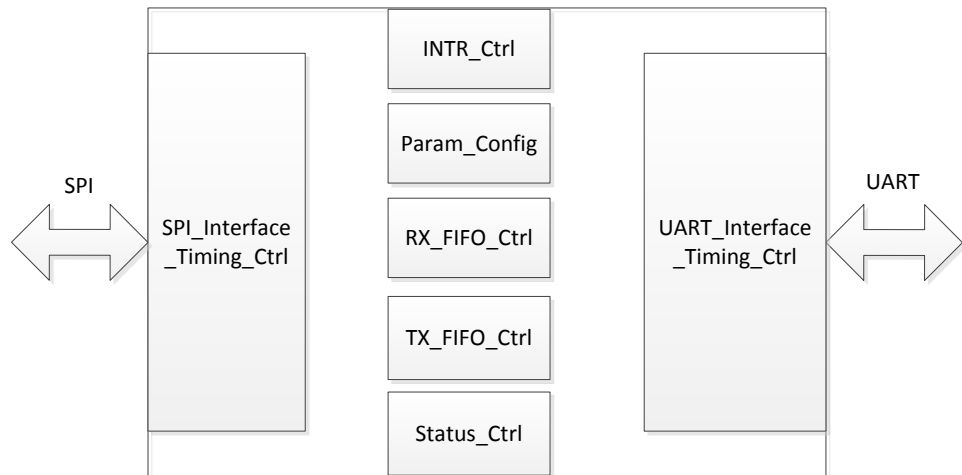
SPI_UART 介于 SPI 主端与 UART 终端之间，实现 SPI 到 UART 的接口转换，完成数据通信功能。SPI 主端包括数据、片选、时钟、中断信号，UART 侧包括收数据、发数据及流控信号。本地输入包括复位和时钟信号。

图 5-1 系统框图



5.2 整体结构

图 5-2 整体结构



整个 IP 设计包括 SPI_interface_Timing_Ctrl、INTR_Ctrl、Param_Config、RX_FIFO_Ctrl、TX_FIFO_Ctrl、Status_Ctrl、UART_Interface_Timing_Ctrl 模块。

- SPI_interface_Timing_Ctrl: 实现与 SPI 主端数据通信，完成 SPI 接口时序控制和协议解析功能；
- INTR_Ctrl: 根据 IP 内部状态完成相应的中断控制功能；
- Param_Config: 接收 SPI 主端发送来的配置参数，并下发到对应模块；
- RX_FIFO_Ctrl: 实现对从 UART 接口接收来的数据缓存；
- TX_FIFO_Ctrl: 实现对 SPI 主端发送来的数据缓存；
- Status_Ctrl: 实现对 IP 内部接口状态和工作状态的检测和统计；
- UART_Interface_Timing_Ctrl: 实现与 UART 端数据通信，完成 UART 接口时序控制和协议解析功能。

5.3 SPI_UART IP 寄存器

SPI_UART IP 寄存器如表 5-1 所示：

表 5-1 Gowin SPI_UART IP 寄存器

序号	名称	寄存器地址（4bit位宽）	默认值（8bit位宽）	操作类型	描述
1	RHR	0x0	-	只读	接收保持寄存器
2	THR	0x0	-	只写	发送保持寄存器
3	IER	0x1	0x00	读/写	中断使能寄存器
4	IIR	0x2	0x01	只读	中断标识寄存器
5	LSR	0x3	0x60	只读	线状态寄存器
6	MSR	0x4	0x10	只读	调制解调状态寄存器
7	MCR	0x5	0x00	读/写	调制解调控制寄存

序号	名称	寄存器地址 (4bit位宽)	默认值 (8bit位宽)	操作类型	描述
					器
8	FCR	0x6	0x00	读/写	FIFO控制寄存器
9	TXLVL	0x7	0x00	只读	发送FIFO有效数据个数
10	RXLVL	0x8	0x00	只读	接收FIFO有效数据个数
11	DLL	0x9	0x10	读/写	UART分频寄存器 (低8bit)
12	DLH	0xA	0x00	读/写	UART分频寄存器 (高8bit)

5.3.1 RHR 寄存器

RHR 寄存器定义如表 5-2 所示。

表 5-2 RHR 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x0	Bit[7:0]	-	只读	缓存接收 UART 侧的数据, 缓存深度 64 字节。

5.3.2 THR 寄存器

THR 寄存器定义如表 5-3 所示。

表 5-3 THR 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x0	Bit[7:0]	-	只写	缓存发送 UART 侧的数据; 缓存深度 64 字节。

5.3.3 IER 寄存器

IER 寄存器定义如表 5-4 所示。

表 5-4 IER 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x1	Bit[7:4]	-	-	保留
	Bit[3]	0	只写	Modem 状态中断。0: 关断; 1: 使能
	Bit[2]	0	只写	接收线状态中断。0: 关断; 1: 使能
	Bit[1]	0	只写	发送保持寄存器中断。0: 关断; 1: 使能
	Bit[0]	0	只写	接收保持寄存器中断。0: 关断; 1: 使能

5.3.4 IIR 寄存器

IIR 寄存器定义如表 5-5 所示。

表 5-5 IIR 寄存器

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
0x2	Bit[7:3]	-	-	保留
	Bit[2:1]	00	只读	中断类型:

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
				11: 接收线状态错误中断 (最高优先级) 10: 接收保持寄存器中断 (第二优先级) 01: 发送保持寄存器中断 (第三优先级) 00: Modem状态中断 (最低优先级)
	Bit[0]	1	只读	中断状态。0: 中断正在等待; 1: 无中断等待

5.3.5 LSR 寄存器

LSR 寄存器定义如表 5-6 所示。

表 5-6 LSR 寄存器

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
0x3	Bit[7]	-	-	保留
	Bit[6]	1	只读	THR/TSR数据状态。 0: THR或TSR有数据待发送; 1: THR和TSR均为空
	Bit[5]	1	只读	THR数据状态。 0: THR有数据待发送; 1: THR为空
	Bit[4]	0	只读	通讯中断。 0: 无通讯中断情况; 1: 检测到通讯中断 (整个数据传输中保持为低电平, 包括起始位+数据位+停止位), 读后清零
	Bit[3]	0	只读	帧错误。 0: 无帧错误; 1: 检测到帧错误 (丢失停止位), 读后清零
	Bit[2]	-	-	保留
	Bit[1]	0	只读	溢出错误。 0: 无益处错误; 1: RHR寄存器数据溢出, 读后清零
	Bit[0]	0	只读	RHR数据状态。 0: 无数据; 1: 至少一个数据存入RHR

5.3.6 MSR 寄存器

MSR 寄存器定义如表 5-7 所示。

表 5-7 MSR 寄存器

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
0x4	Bit[7:5]	-	-	保留

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
	Bit[4]	X	只读	允许发送 (高有效)
	Bit[3:1]	-	-	保留
	Bit[0]	0	只读	CTS信号状态变化指示(高有效), 读后清零

注!

X 表示信号值由入口信号状态决定。

5.3.7 MCR 寄存器

MCR 寄存器定义如表 5-8 所示。

表 5-8 MCR 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x5	Bit[7:3]	-	-	保留
	Bit[2]	0	读/写	收发环回。 0: 正常数据通讯; 1: 环回开启
	Bit[1]	0	读/写	要求发送。 0: 驱动 RTS 为高; 1: 驱动 RTS 为低
	Bit[0]	-	-	保留

5.3.8 FCR 寄存器

FCR 寄存器定义如表 5-9 所示。

表 5-9 FCR 寄存器

寄存器地址 (4bit位宽)	比特位	默认值	操作类型	描述
0x6	Bit[7:6]	00	读/写	接收保持寄存器中断触发条件 (当FIFO中数据数量大于等于如下值时触发): 00: 8字节; 01: 16字节; 10: 32字节; 11: 48字节;
	Bit[5:4]	00	读/写	发送保持寄存器中断触发条件 (当FIFO中空余空间大于等于如下值时触发): 00: 8字节; 01: 16字节; 10: 32字节; 11: 48字节;
	Bit[3]	-	-	保留
	Bit[3]	0	读/写	复位发送FIFO。 0: 不复位发送FIFO; 1: 复位发送FIFO
	Bit[1]	0	读/写	复位接收FIFO。 0: 不复位接收FIFO; 1: 复位接收FIFO
	Bit[0]	0	读/写	FIFO使能。

				0: 关闭发送和接收FIFO; 1: 启用发送和接收FIFO
--	--	--	--	-----------------------------------

5.3.9 TXLVL 寄存器

TXLVL 寄存器定义如表 5-10 所示。

表 5-10 TXLVL 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x7	Bit[7]	-	-	保留
	Bit[6:0]	0x00	只读	TX FIFO 有效数据数量。 取值范围 0~64

5.3.10 RXLVL 寄存器

RXLVL 寄存器定义如表 5-11 所示。

表 5-11 RXLVL 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x8	Bit[7]	-	-	保留
	Bit[6:0]	0x00	只读	RX FIFO 有效数据数量。取 值范围 0~64

5.3.11 DLL 寄存器

DLL 寄存器定义如表 5-12 所示。

表 5-12 DLL 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0x9	Bit[7:0]	0x10	读/写	UART 分频寄存器的低 8 位

5.3.12 DLH 寄存器

DLH 寄存器定义如表 5-13 所示。

表 5-13 DLH 寄存器

寄存器地址 (4bit 位宽)	比特位	默认值	操作类型	描述
0xA	Bit[7:0]	0x00	读/写	UART 分频寄存器的高 8 位

注!

UART 波特率=输入时钟频率/分频寄存器值 (DLH+DLL), 且分频寄存器值不小于 10。

5.4 基本操作流程

5.4.1 初始化流程

1. 上电后, SPI 主端首先需要对 IP 进行参数配置, 建议顺序如下:
 - a). MCR
 - b). FCR
 - c). DLL
 - d). DLH

e). IER（使能寄存器最后设置）

2. 然后 SPI 主端可以根据读写控制操作流程进行与 UART 端数据通信。

5.4.2 写操作流程

禁用中断写操作

禁用发送保持寄存器中断情况，SPI 主端发起写操作流程如下：

1. 首先 SPI 主端通过查询 TXLVL 寄存器，读取当前发送 FIFO 缓存状态；
2. 当发送 FIFO 有足够空间时，SPI 主端可通过 THR 寄存器写入数据。

启用中断写操作

启用发送保持寄存器中断情况，SPI 主端发起写操作流程如下：

1. 当 SPI 主端收到 SPI 从端发送来的中断信号后，查询 IIR 寄存器，读取当前中断信息；
2. 若中断为发送保持寄存器中断，则 SPI 主端可以向 THR 寄存器写入数据。

5.4.3 读操作流程

禁用中断读操作

禁用接收保持寄存器中断情况下，SPI 主端发起读操作流程如下：

1. SPI 主端查询 RXLVL 寄存器，读取当前接收 FIFO 缓存状态；
2. 当发现接收 FIFO 缓存中有数据时，SPI 主端可通过 RHR 寄存器读取数据。

启用中断读操作

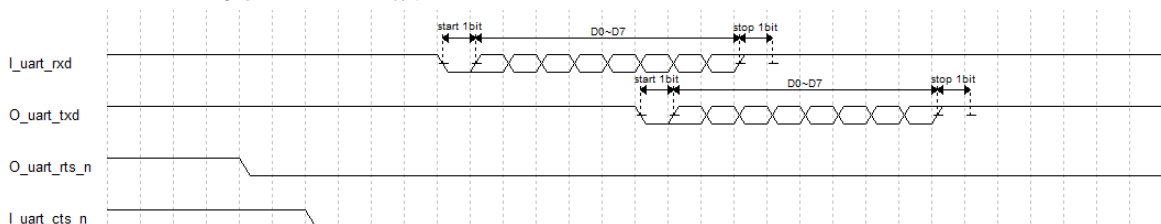
启用接收保持寄存器中断情况下，SPI 主端发起读操作流程如下：

1. 当 SPI 主端收到 SPI 从端发送来的中断信号后，查询 IIR 寄存器，读取当前中断信息；
2. 若中断为接收保持寄存器中断，则 SPI 主端通过 RHR 寄存器读取数据。

5.5 接口时序

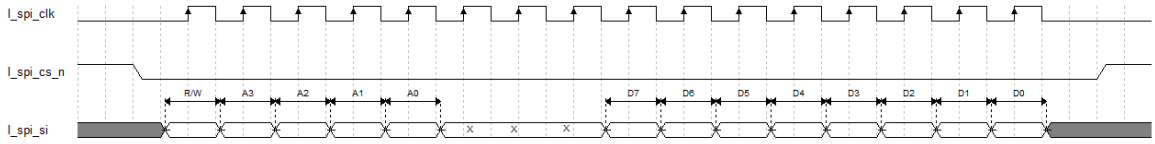
5.5.1 UART 接口时序

图 5-3 UART 接口时序



5.5.2 SPI 接口寄存器写时序

图 5-4 SPI 接口寄存器写时序

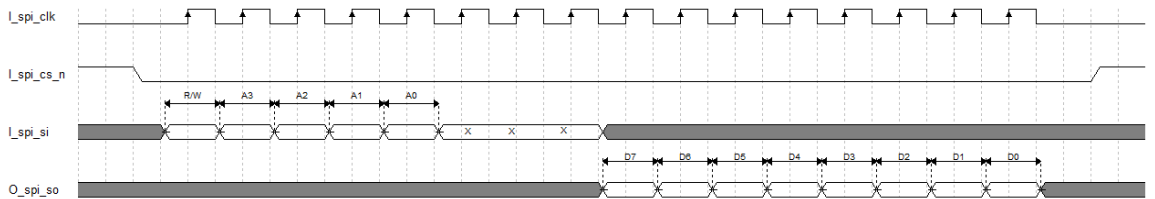


注!

- R/W: 0 表示写操作;
- X: 不关心位;
- A3~A0: 寄存器地址, 4bit 位宽;
- 输入工作时钟必须不小于输入 SPI 数据速率的 6 倍。

5.5.3 SPI 接口寄存器读时序

图 5-5 SPI 接口寄存器读时序

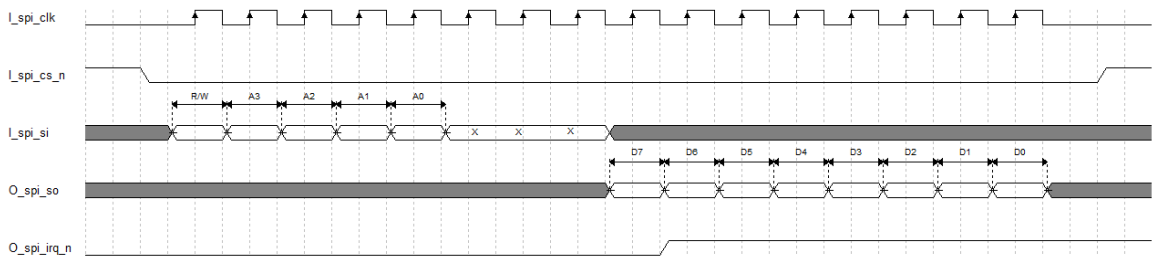


注!

- R/W: 1 表示读操作;
- X: 不关心位;
- A3~A0: 寄存器地址, 4bit 位宽。

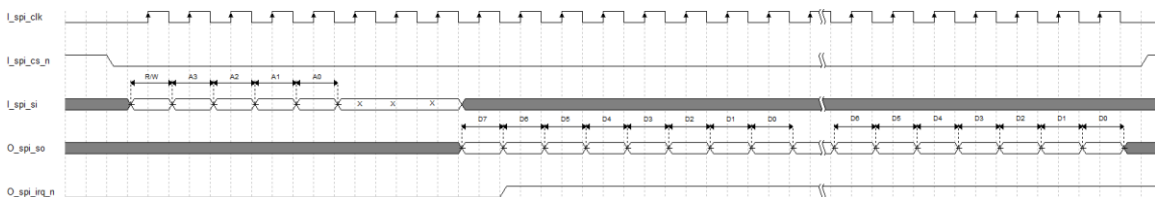
5.5.4 SPI 接口读 LSR/MSR 清中断时序

图 5-6 SPI 接口读 LSR/MSR 清中断时序



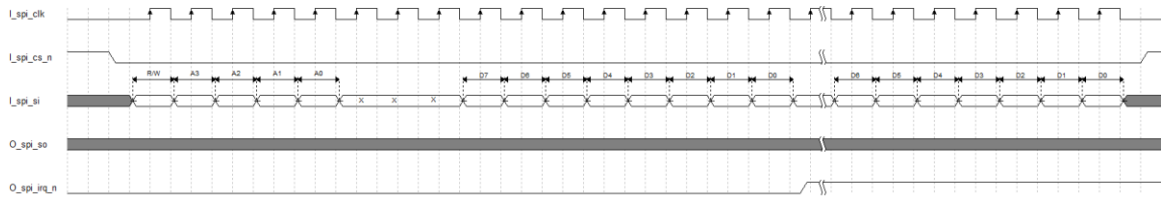
5.5.5 SPI 接口读 RHR 清中断时序

图 5-7 SPI 接口读 RHR 清中断时序



5.5.6 SPI 接口写 THR 清中断时序

图 5-8 SPI 接口写 THR 清中断时序

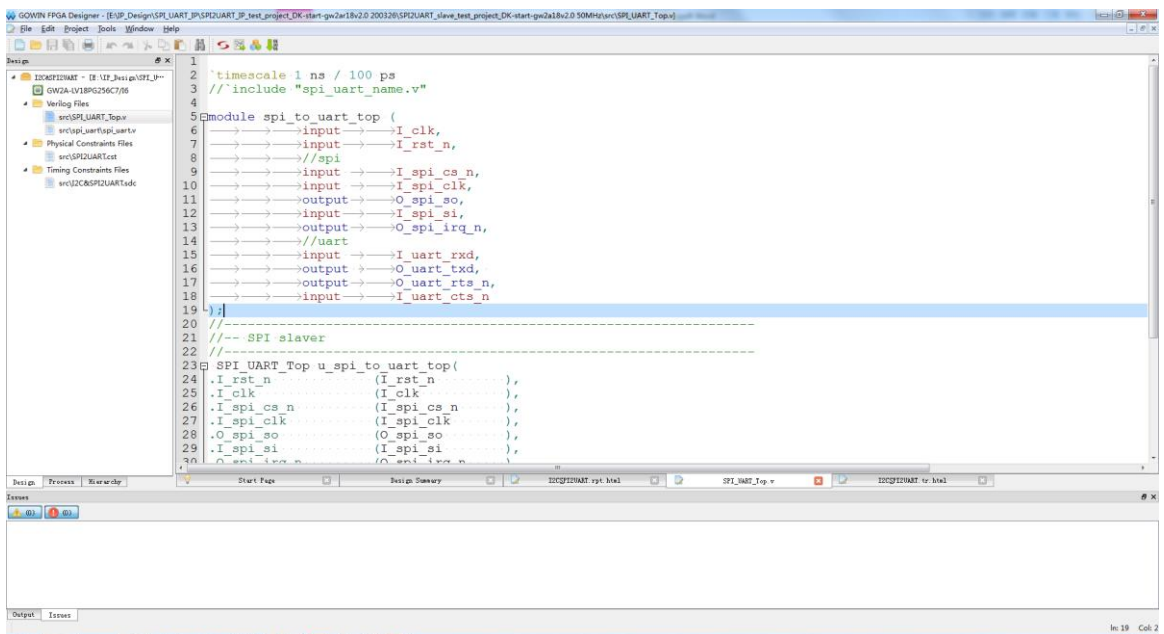


6 界面配置

在高云云源软件界面菜单栏 Tools 下，可启动 IP Core Generator 工具，完成调用并配置 SPI_UART IP。

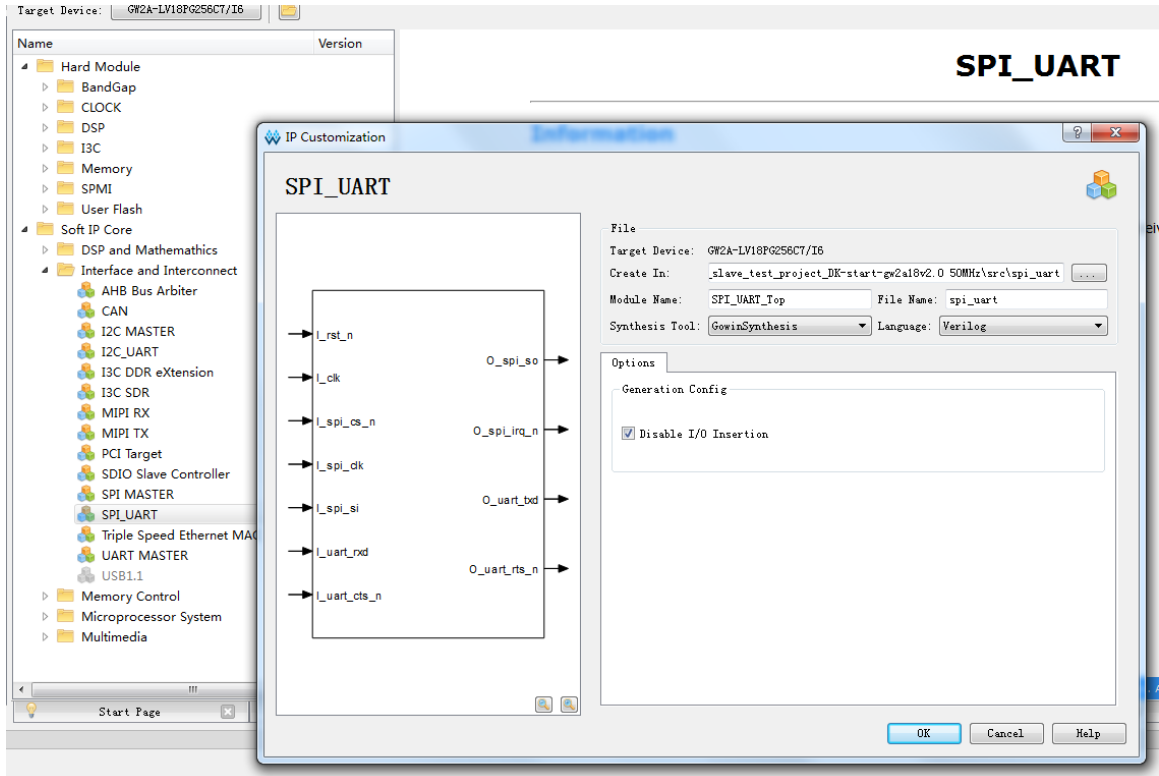
1. 启动高云云源软件后，打开工程，如图 6-1 所示：

图 6-1 打开工程



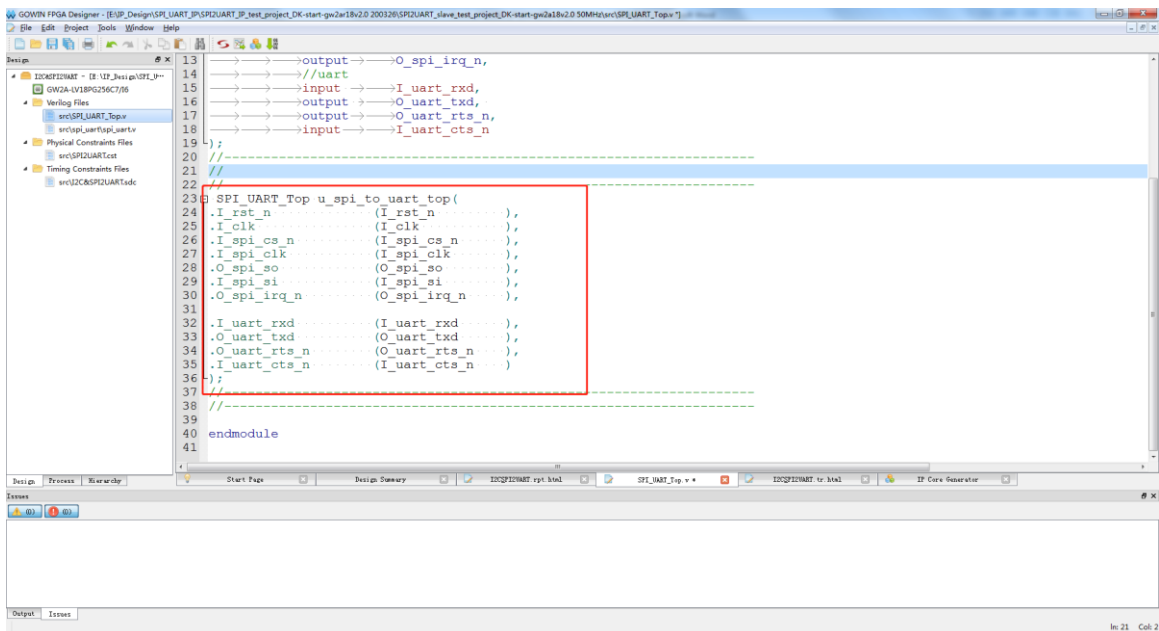
2. 在“Tools”菜单下点击“IP Core Generator”，调用 SPI_UART ，然后点击“OK”后生成 SPI_UART Module，如图 6-2 所示。

图 6-2 SPI_UART 配置界面



3. 在用户程序中例化 SPI_UART，如图 6-3 所示：

图 6-3 例化 SPI_UART



综合、P&R、下载 bitstream 文件到 FPGA 芯片后可实现 SPI_UART 功能。

7 参考设计

详细信息请参见高云半导体官网 [SPI_UART](#) 相关参考设计。

