



Gowin SDIO_UART IP 用户指南

IPUG946-1.0, 2020-05-07

版权所有©2020 广东高云半导体科技股份有限公司

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

免责声明

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

版本信息

日期	版本	说明
2020/05/07	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
3 特征与性能	3
3.1 特性	3
3.2 性能	3
3.3 资源占用	3
4 系统框图	4
4.1 系统框图	4
5 端口列表	5
5.1 系统信号	5
5.2 SDIO 侧信号	5
5.3 UART 侧信号	6
6 结构及功能描述	7
6.1 整体结构	7
6.2 初始化流程	8
6.3 UART 端口写操作流程	8
6.3.1 禁用中断	8
6.3.2 启用中断	9
6.4 UART 端口读操作流程	9
6.4.1 禁用中断	9
6.4.2 启用中断	9
6.5 UART 端口状态查询操作	10
6.5.1 禁用中断	10

6.5.2 启用中断	10
6.6 寄存器.....	10
6.6.1 寄存器概述.....	10
6.6.2 寄存器描述.....	11
6.7 接口时序	18
6.7.1 UART 接口时序（4 线模式）	18
6.7.2 SDIO 总线接口.....	18
7 参数配置	19
7.1 概述	19
7.2 参数	19
8 界面配置	20
8.1 概述	20
8.2 应用框图	20
8.3 GUI 调用.....	21

图目录

图 4-1 系统框图	4
图 6-1 整体结构	7
图 6-2 UART 接口时序	18
图 8-1 应用框图	20

表目录

表 1-1 术语、缩略语	1
表 3-1 性能参考	3
表 5-1 系统信号	5
表 5-2 SDIO 侧信号.....	5
表 5-3 UART4 侧信号.....	6
表 5-4 UART3 侧信号.....	6
表 5-5 UART2 侧信号.....	6
表 5-6 UART1 侧信号.....	6
表 6-1 寄存器总表.....	10
表 6-2 CIR	11
表 6-3 IITR.....	11
表 6-4 IER	11
表 6-5 UBDFR4	12
表 6-6 UBDFR3	12
表 6-7 UBDFR2	13
表 6-8 UBDFR1	13
表 6-9 MCR	13
表 6-10 TFCR.....	14
表 6-11 RFCR.....	14
表 6-12 TFR4.....	14
表 6-13 TFR3.....	14
表 6-14 TFR2.....	14
表 6-15 TFR1.....	14
表 6-16 IIR.....	15
表 6-17 RFR4	15
表 6-18 RFR3	15
表 6-19 RFR2	15
表 6-20 RFR1	15
表 6-21 LSR.....	16

表 6-22 MSR.....17

表 6-23 TFLVL17

表 6-24 RFLVL.....17

1 关于本手册

1.1 手册内容

SDIO-UART 用户手册主要包括功能特点、整体框图、工作原理、信号定义、参数介绍、GUI 调用、接口时序等。目的帮助用户快速了解高云半导体 SDIO-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。

SDIO IP 使用见 [IPUG905](#), *Gowin SDIO Slave Controller 用户指南*。

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
SDIO	Secure Digital Input and Output	安全数字输入输出
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

SDIO (Secure Digital Input and Output) 在 SD 标准上定义了一种外设接口。

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 SDIO-UART IP, 实现 SDIO 与 UART 间的接口转换功能。

SDIO-UART 通过软件界面调用后, 可通过 Gowin 软件综合、布局布线适用到 GW1N、GW1NR、GW1NS、GW1NZ、GW2A、GW2AR 等系列 FPGA 产品。

3 特征与性能

3.1 特性

- 实现 SDIO 与 UART 接口之间的数据通信；
- SDIO 接口为从端模式；
- UART 接口为二线或四线模式，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；
- 最大支持 4 路 UART 通道
- 支持中断方式；
- UART 收 FIFO 深度 64Bytes；
- UART 发 FIFO 深度 64Bytes；
- SDIO Host 端所有用户侧的操作都通过 CMD53 接口来实现，包括 Host 端清中断寄存器操作，即通过 CMD53 接口读写 Funtion1 寄存器实现与 UART 侧的参数配置、状态查询及数据读写。
- SDIO CMD53 命令 OP Code 固定取值为 0, 选取 Multi byte R/W to fixed address 的方式完成与 UART 接口的数据交互；
- 完全可综合；
- 设计语言为 Verilog。

3.2 性能

SDIO-UART 的工作频率取决于 SDIO 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。

3.3 资源占用

表 3-1 性能参考

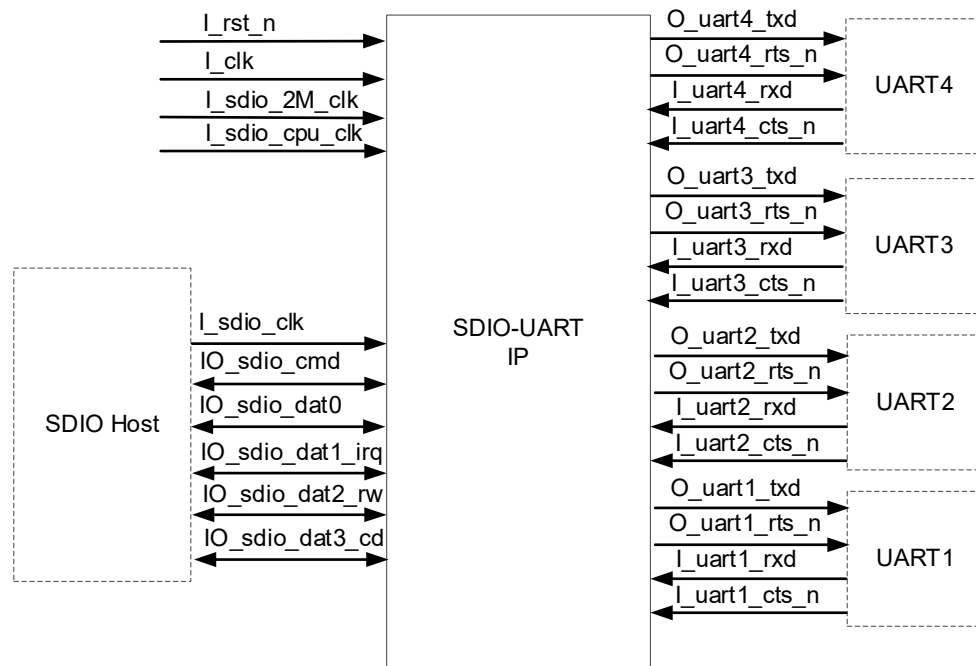
芯片型号	编程语言	LUT4资源	REG资源
GW2A-LV18PG256C7/I6	Verilog	6313	3289

4 系统框图

4.1 系统框图

SDIO-UART 介于 SDIO 主端与 UART 终端之间，实现 SDIO 到 UART 的接口转换，完成数据通信功能。SDIO 主端包括时钟、命令、数据、中断等信号，UART 侧信号包括收数据、发数据及流控信号。本地输入包括复位和时钟信号。

图 4-1 系统框图



注!

- 正常情况下，UART 接口侧通过中断来触发 SDIO Host 端进行操作；在 SDIO Host 端禁用中断寄存器所有中断后，SDIO Host 可以通过查询来读取状态和控制数据收发操作。
- 如果启用中断功能，则 UART 接口侧触发中断后，待用户收到中断时，需先查询 IIR 状态值，然后根据 IIR 的指示执行相应操作（如果用户跳过也可以），最后用户发送清中断指令，通知 UART 接口侧清除本次中断。

5 端口列表

5.1 系统信号

表 5-1 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	系统时钟，用于内部逻辑及UART接口处理	
3	l_sdio_2M_clk	I	SDIO从端所需时钟	
4	l_sdio_cpu_clk	I	SDIO从端所需时钟	

5.2 SDIO 侧信号

表 5-2 SDIO 侧信号

序号	信号名称	方向	描述	备注
1	l_sdio_clk	I	SDIO时钟输入信号	所有信号输入输出方向均以IP为参考
2	IO_sdio_cmd	IO	SDIO Command/Response信号	
3	IO_sdio_dat0	IO	SDIO 1bit模式为dat信号输入，SDIO 4bit模式为dat0信号输入	
4	IO_sdio_dat1_irq	IO	SDIO 1bit模式为中断信号输入，SDIO 4bit模式为dat1和中断复用信号输入	
5	IO_sdio_dat2_rw	IO	SDIO 1bit模式为Read Wait信号输入，SDIO 4bit模式为dat2和Read Wait复用信号输入	
6	IO_sdio_dat3_cd	IO	SDIO 1bit模式未定义，SDIO 4bit模式为dat3信号输入	

5.3 UART 侧信号

表 5-3 UART4 侧信号

序号	信号名称	方向	描述	备注
1	I_uart4_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_uart4_txd	O	输出串口数据	
3	O_uart4_rts_n	O	允许发送信号	
4	I_uart4_cts_n	I	对端准备好信号	

表 5-4 UART3 侧信号

序号	信号名称	方向	描述	备注
1	I_uart3_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_uart3_txd	O	输出串口数据	
3	O_uart3_rts_n	O	允许发送信号	
4	I_uart3_cts_n	I	对端准备好信号	

表 5-5 UART2 侧信号

序号	信号名称	方向	描述	备注
1	I_uart2_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_uart2_txd	O	输出串口数据	
3	O_uart2_rts_n	O	允许发送信号	
4	I_uart2_cts_n	I	对端准备好信号	

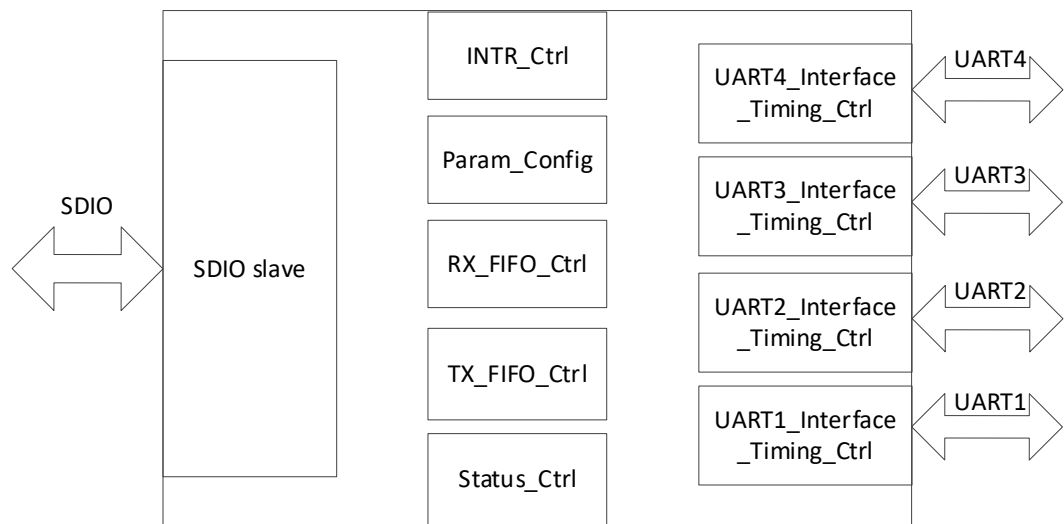
表 5-6 UART1 侧信号

序号	信号名称	方向	描述	备注
1	I_uart1_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_uart1_txd	O	输出串口数据	
3	O_uart1_rts_n	O	允许发送信号	
4	I_uart1_cts_n	I	对端准备好信号	

6 结构及功能描述

6.1 整体结构

图 6-1 整体结构



整个 IP 设计包括 SDIO_slave、INTR_Ctrl、Param_Config、RX_FIFO_Ctrl、TX_FIFO_Ctrl、Status_Ctrl、UART4_Interface_Timing_Ctrl、UART3_Interface_Timing_Ctrl、UART2_Interface_Timing_Ctrl、UART1_Interface_Timing_Ctrl 模块。

- SDIO_slave: 实现与 SDIO 主端数据通信，完成 SDIO 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 SDIO 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 SDIO 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；

- UART4_Interface_Timing_Ctrl: 实现与 UART4 端数据通信, 完成 UART4 接口时序控制和协议解析功能;
- UART3_Interface_Timing_Ctrl: 实现与 UART3 端数据通信, 完成 UART3 接口时序控制和协议解析功能;
- UART2_Interface_Timing_Ctrl: 实现与 UART2 端数据通信, 完成 UART2 接口时序控制和协议解析功能;
- UART1_Interface_Timing_Ctrl: 实现与 UART1 端数据通信, 完成 UART1 接口时序控制和协议解析功能。

6.2 初始化流程

1. 上电后, SDIO 主端首先需要对 IP 进行参数配置, 建议顺序如下:
 - a) UBDFR4 寄存器
 - b) UBDFR3 寄存器
 - c) UBDFR2 寄存器
 - d) UBDFR1 寄存器
 - e) MCR 寄存器
 - f) TFCR 寄存器
 - g) RFCR 寄存器
 - h) IITR 寄存器
 - i) IER 寄存器 (使能寄存器最后设置)
2. 然后 SDIO 主端可以根据读写控制操作流程进行与 UART 端数据通信。

6.3 UART 端口写操作流程

6.3.1 禁用中断

禁用发送 FIFO 寄存器中断情况, SDIO Host 主端发起写操作流程如下 (以 UART1 为例):

1. SDIO Host 查询 TFLVL 寄存器中 UART1 发送 FIFO 中的空余字节数量;
2. 当 UART1 发送 FIFO 中有足够空余字节且 CTS 为允许发送状态时, SDIO Host 通过 CMD53 接口向 TFR1 寄存器写数据;
3. 然后 UART1 发送模块会依次按照设定波特率发送到链路上。

6.3.2 启用中断

启用发送 FIFO 寄存器中断情况，UART 端发起写操作流程如下（以 UART1 为例）：

1. 当 UART 端口检测到发送 FIFO 中空余字节数量大于等于 TFCR 中 UART1 所设 Trigger 值时，触发中断；
2. SDIO Host 收到中断后，会通过 CMD53 接口查询 IIR 寄存器；
3. 当 SDIO Host 查询到发送 FIFO 中断时，会通过 CMD53 接口向 TFR1 寄存器写数据；
4. 待写完所有数据后，SDIO Host 端会向 UART1 端口发送清中断指令；
5. 最后 UART1 发送模块会依次按照设定波特率发送到链路上。

6.4 UART 端口读操作流程

6.4.1 禁用中断

禁用接收 FIFO 寄存器中断情况，SDIO Host 端发起读操作流程如下（以 UART1 为例）：

1. SDIO Host 通过 CMD53 接口查询 RFLVL 寄存器中 UART1 接收 FIFO 中的已接收字节数量；
2. 当 UART1 接收 FIFO 中有已接收数据时，SDIO Host 会通过 CMD53 接口对 RFR1 寄存器进行数据读取操作。

6.4.2 启用中断

启用接收 FIFO 寄存器中断情况，UART1 端口发起读操作流程如下（以 UART1 为例）：

1. 当 UART 端口检测到接收 FIFO 中接收字节数量大于等于 RFCR 中 UART1 所设 Trigger 值时，触发中断；
2. SDIO Host 收到中断后，通过 CMD53 接口查询 IIR 寄存器；
3. 当 SDIO Host 查询到为接收 FIFO 中断时，再通过查询 RFLVL 寄存器获取 UART1 接收 FIFO 中的已接收字节数量；
4. 然后会通过 CMD53 接口发起从 RFR1 寄存器读数据操作；
5. 待 SDIO Host 完成数据读取后，会向 UART1 端口发送清中断指令。

6.5 UART 端口状态查询操作

6.5.1 禁用中断

禁用 Modem 状态中断和接收线状态中断情况，SDIO Host 主端发起状态查询操作流程如下：

当 UART 接收 FIFO 中有已接收数据时，SDIO Host 会通过 CMD53 接口 SDIO Host 通过 CMD53 接口可实时查询 LSR 和 MSR 寄存器中对应的接收线和 CTS 状态；

6.5.2 启用中断

启用 Modem 状态中断和接收线状态中断情况，UART 端发起状态查询操作流程如下：

1. 当 UART 的 LSR 和 MSR 寄存器中接收线状态异常或 CTS 变化时，UART 端口会触发中断通知 SDIO Host；
2. 收到中断后，SDIO Host 先读取 IIR 中断寄存器，并根据信息再通过 CMD53 接口去查询 LSR 或 MSR 寄存器，最后清中断。

6.6 寄存器

6.6.1 寄存器概述

SDIO-UART 寄存器包括如下：

表 6-1 寄存器总表

序号	寄存器名称	寄存器地址 (17bit位宽)	默认值 (32bit位宽)	操作类型	描述
1	CIR	0x10000	0x00000000	写	清中断寄存器
2	IITR	0x10001	0x00000000	读/写	中断间隔时间寄存器
3	IER	0x10002	0x00000000	读/写	中断使能寄存器
4	UBDFR4	0x10003	0x00000000	读/写	UART4波特率分频寄存器
5	UBDFR3	0x10004	0x00000000	读/写	UART3波特率分频寄存器
6	UBDFR2	0x10005	0x00000000	读/写	UART2波特率分频寄存器
7	UBDFR1	0x10006	0x00000000	读/写	UART1波特率分频寄存器
8	MCR	0x10007	0x00000000	读/写	调制解调控制寄存器
9	TFCR	0x10008	0x40404040	读/写	发FIFO控制寄存器
10	RFCR	0x10009	0x01010101	读/写	收FIFO控制寄存器
11	TFR4	0x1000a	--	写	UART4发送FIFO寄存器
12	TFR3	0x1000b	--	写	UART3发送FIFO寄存器
13	TFR2	0x1000c	--	写	UART2发送FIFO寄存器
14	TFR1	0x1000d	--	写	UART1发送FIFO寄存器
15	IIR	0x00000	0x00000000	读	中断标识寄存器

序号	寄存器名称	寄存器地址 (17bit位宽)	默认值 (32bit位宽)	操作类型	描述
16	RFR4	0x00001	--	读	UART4接收FIFO寄存器
17	RFR3	0x00002	--	读	UART3接收FIFO寄存器
18	RFR2	0x00003	--	读	UART2接收FIFO寄存器
19	RFR1	0x00004	--	读	UART1接收FIFO寄存器
20	LSR	0x00005	0x00000000	读	接收线状态寄存器
21	MSR	0x00006	0x00000000	读	调制解调器状态寄存器
22	TFLVL	0x00007	0x00000000	读	发送FIFO指示寄存器
23	RFLVL	0x00008	0x00000000	读	接收FIFO指示寄存器

6.6.2 寄存器描述

表 6-2 CIR

寄存器地址	比特位	默认值	操作类型	描述
0x10000	Bit[31:1]	-	-	保留
	Bit[0]	0	写	Bit0: 1 清中断; 当收到Host端查询清中断寄存器且发送的bit0为1时, SDIO从端拉低 fun1_interrupt管脚, 以清除中断。

表 6-3 ITR

寄存器地址	比特位	默认值	操作类型	描述
0x10001	Bit[31:16]	-	-	保留
	Bit[15:0]	0x0000	读/写	中断触发间隔时间: 取值范围 1~65535。

表 6-4 IER

寄存器地址	Bit位	默认值	操作类型	描述
0x10002	Bit[31:24]	0x00	读/写	Bit[31:28]: 保留 Bit27: 接收线状态中断, 0: 关断, 1: 使能; Bit26: 接收FIFO寄存器中断, 0: 关断, 1: 使能; Bit25: 发送FIFO寄存器中断, 0: 关断, 1: 使能; Bit24: Modem状态中断, 0: 关断, 1: 使能;
	Bit[23:16]	0x00	读/写	Bit[23:20]: 保留 Bit19: 接收线状态中断, 0: 关断, 1: 使能; Bit18: 接收FIFO寄存器中断, 0: 关断, 1: 使能;

寄存器地址	Bit位	默认值	操作类型	描述
				Bit17: 发送FIFO寄存器中断, 0: 关断, 1: 使能; Bit16: Modem状态中断, 0: 关断, 1: 使能;
	Bit[15:8]	0x00	读/写	Bit[15:12]: 保留 Bit11: 接收线状态中断, 0: 关断, 1: 使能; Bit10: 接收FIFO寄存器中断, 0: 关断, 1: 使能; Bit9: 发送FIFO寄存器中断, 0: 关断, 1: 使能; Bit8: Modem状态中断, 0: 关断, 1: 使能;
	Bit[7:0]	0x00	读/写	Bit[7:4]: 保留 Bit3: 接收线状态中断, 0: 关断, 1: 使能; Bit2: 接收FIFO寄存器中断, 0: 关断, 1: 使能; Bit1: 发送FIFO寄存器中断, 0: 关断, 1: 使能; Bit0: Modem状态中断, 0: 关断, 1: 使能;

表 6-5 UBDFR4

寄存器地址	Bit位	默认值	操作类型	描述
0x10003	Bit[31:16]	-	-	保留
	Bit[15:0]	0x0000	读/写	UART4波特率: 取值范围 0~65525; UART4波特率=输入时钟频率/ (分频寄存器+10)

表 6-6 UBDFR3

寄存器地址	Bit位	默认值	操作类型	描述
0x10004	Bit[31:16]	-	-	保留
	Bit[15:0]	0x0000	读/写	UART3波特率: 取值范围 0~65525; UART4波特率=输入时钟频率/ (分频寄存器+10)

表 6-7 UBDFR2

寄存器地址	Bit位	默认值	操作类型	描述
0x10005	Bit[31:16]	-	-	保留
	Bit[15:0]	0x0000	读/写	UART2波特率：取值范围0~65525； UART4波特率=输入时钟频率/（分频寄存器+10）

表 6-8 UBDFR1

寄存器地址	Bit位	默认值	操作类型	描述
0x10006	Bit[31:16]	-	-	保留
	Bit[15:0]	0x0000	读/写	UART1波特率：取值范围0~65525； UART4波特率=输入时钟频率/（分频寄存器+10）

表 6-9 MCR

寄存器地址	Bit位	默认值	操作类型	描述
0x10007	Bit[31:24]	0x00	读/写	Bit[31:26]：保留 Bit25：UART4侧收发环回 0：正常数据通讯；1：环回开启 Bit24：RTS4控制 0：驱动RTS为高；1：驱动RTS为低
	Bit[23:16]	0x00	读/写	Bit[23:18]：保留 Bit17：UART3侧收发环回 0：正常数据通讯；1：环回开启 Bit16：RTS3控制 0：驱动RTS为高；1：驱动RTS为低
	Bit[15:8]	0x00	读/写	Bit[15:10]：保留 Bit9：UART2侧收发环回 0：正常数据通讯；1：环回开启 Bit8：RTS2控制 0：驱动RTS为高；1：驱动RTS为低
	Bit[7:0]	0x00	读/写	Bit[7:2]：保留 Bit1：UART1侧收发环回 0：正常数据通讯；1：环回开启 Bit0：RTS1控制 0：驱动RTS为高；1：驱动RTS为低 注！ RTS为低表示接收准备就绪

表 6-10 TFCR

寄存器地址	Bit位	默认值	操作类型	描述
0x10008	Bit[31:24]	0x40	读/写	UART4发送FIFO剩余空间Triger取值范围1~64（单位byte）
	Bit[23:16]	0x40	读/写	UART3发送FIFO剩余空间Triger取值范围1~64（单位byte）
	Bit[15:8]	0x40	读/写	UART2发送FIFO剩余空间Triger取值范围1~64（单位byte）
	Bit[7:0]	0x40	读/写	UART1发送FIFO剩余空间Triger取值范围1~64（单位byte）

表 6-11 RFCR

寄存器地址	Bit位	默认值	操作类型	描述
0x10009	Bit[31:24]	0x01	读/写	UART4接收FIFO有效数据Triger取值范围1~64（单位byte）
	Bit[23:16]	0x01	读/写	UART3接收FIFO有效数据Triger取值范围1~64（单位byte）
	Bit[15:8]	0x01	读/写	UART2接收FIFO有效数据Triger取值范围1~64（单位byte）
	Bit[7:0]	0x01	读/写	UART1接收FIFO有效数据Triger取值范围1~64（单位byte）

表 6-12 TFR4

寄存器地址	Bit位	默认值	操作类型	描述
0x1000a	Bit[7:0]	-	读/写	UART4 发送FIFO, 最大深度64bytes

表 6-13 TFR3

寄存器地址	Bit位	默认值	操作类型	描述
0x1000b	Bit[7:0]	-	读/写	UART3 发送FIFO, 最大深度64bytes

表 6-14 TFR2

寄存器地址	Bit位	默认值	操作类型	描述
0x1000c	Bit[7:0]	-	读/写	UART2 发送FIFO, 最大深度64bytes

表 6-15 TFR1

寄存器地址	Bit位	默认值	操作类型	描述
0x1000d	Bit[7:0]	-	读/写	UART1 发送FIFO, 最大深度64bytes

表 6-16 IIR

寄存器地址	Bit位	默认值	操作类型	描述
0x00000	Bit[31:6]	-	-	保留
	Bit[5:4]	00		Bit[5:4]: 11: 接收线状态中断 (最高优先级) 10: 接收FIFO状态中断 (第二优先级) 01: 发送FIFO状态中断 (第三优先级) 00: 调制解调器状态中断 (最低优先级)
	Bit[3:0]	0000	只读	Bit[3:0]: 1000: UART4 中断正在等待 0100: UART3 中断正在等待 0010: UART2 中断正在等待 0001: UART1 中断正在等待 0000: 无中断 其他: 保留

表 6-17 RFR4

寄存器地址	Bit位	默认值	操作类型	描述
0x00001	Bit[7:0]	-	只读	UART4 接收FIFO, 最大深度64bytes

表 6-18 RFR3

寄存器地址	Bit位	默认值	操作类型	描述
0x00002	Bit[7:0]	-	只读	UART3 接收FIFO, 最大深度64bytes

表 6-19 RFR2

寄存器地址	Bit位	默认值	操作类型	描述
0x00003	Bit[7:0]	-	只读	UART2 接收FIFO, 最大深度64bytes

表 6-20 RFR1

寄存器地址	Bit位	默认值	操作类型	描述
0x00004	Bit[7:0]	-	只读	UART1 接收FIFO, 最大深度64bytes

表 6-21 LSR

寄存器地址	Bit位	默认值	操作类型	描述
0x00005	Bit[31:24]	0x00	只读	Bit[31:27]:保留 Bit26: 0: 无通讯中断, 1: 检测到通讯中断; 读后清零 Bit25: 0: 无帧错误, 1: 检测到帧错误; 读后清零 Bit24: 0: 接收FIFO无溢出, 1: 接收FIFO检测到溢出; 读后清零
	Bit[23:16]	0x00	只读	Bit[23:19]:保留 Bit18: 0: 无通讯中断, 1: 检测到通讯中断; 读后清零 Bit17: 0: 无帧错误, 1: 检测到帧错误; 读后清零 Bit16: 0: 接收FIFO无溢出, 1: 接收FIFO检测到溢出; 读后清零
	Bit[15:8]	0x00	只读	Bit[15:11]:保留 Bit10: 0: 无通讯中断, 1: 检测到通讯中断; 读后清零 Bit9: 0: 无帧错误, 1: 检测到帧错误; 读后清零 Bit8: 0: 接收FIFO无溢出, 1: 接收FIFO检测到溢出; 读后清零
	Bit[7:0]	0x00	只读	Bit[7:3]:保留 Bit2: 0: 无通讯中断, 1: 检测到通讯中断; 读后清零 Bit1: 0: 无帧错误, 1: 检测到帧错误; 读后清零 Bit0: 0: 接收FIFO无溢出, 1: 接收FIFO检测到溢出; 读后清零

表 6-22 MSR

寄存器地址	Bit位	默认值	操作类型	描述
0x00006	Bit[31:24]	0x00	只读	Bit[31:26]:保留 Bit25: CTS4信号状态（高有效，代表对端接收准备好）； Bit24: CTS4信号状态变化指示（高有效），读后清零
	Bit[23:16]	0x00	只读	Bit[23:18]:保留 Bit17: CTS3信号状态（高有效，代表对端接收准备好）； Bit16: CTS3信号状态变化指示（高有效），读后清零
	Bit[15:8]	0x00	只读	Bit[15:10]:保留 Bit9: CTS2信号状态（高有效，代表对端接收准备好）； Bit8: CTS2信号状态变化指示（高有效），读后清零
	Bit[7:0]	0x00	只读	Bit[7:2]:保留 Bit1: CTS1信号状态（高有效，代表对端接收准备好）； Bit0: CTS1信号状态变化指示（高有效），读后清零

表 6-23 TFLVL

寄存器地址	Bit位	默认值	操作类型	描述
0x00007	Bit[31:24]	0x00	只读	UART4发送FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[23:16]	0x00	只读	UART3发送FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[15:8]	0x00	只读	UART2发送FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[7:0]	0x00	只读	UART1发送FIFO中的有效数据个数（单位Byte），取值范围0~64

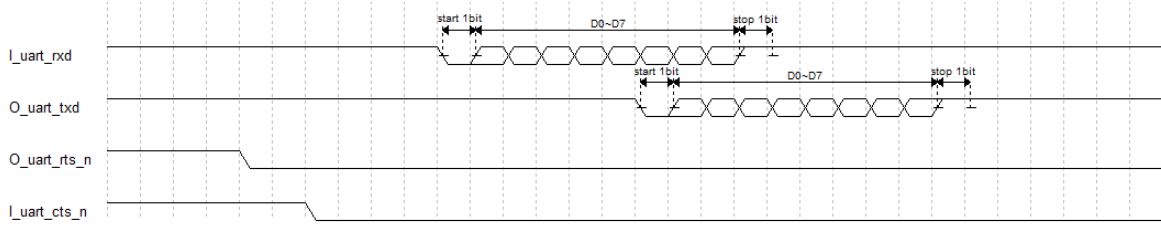
表 6-24 RFLVL

寄存器地址	Bit位	默认值	操作类型	描述
0x00008	Bit[31:24]	0x00	只读	UART4接收FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[23:16]	0x00	只读	UART3接收FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[15:8]	0x00	只读	UART2接收FIFO中的有效数据个数（单位Byte），取值范围0~64
	Bit[7:0]	0x00	只读	UART1接收FIFO中的有效数据个数（单位Byte），取值范围0~64

6.7 接口时序

6.7.1 UART 接口时序 (4 线模式)

图 6-2 UART 接口时序



6.7.2 SDIO 总线接口

符合标准 SDIO 总线协议和时序。

7 参数配置

7.1 概述

SDIO-UART 的 GUI 参数定义。

7.2 参数

Uart4 channel	Uart4 channel select : Yes or No Default: No
Uart4 4wire mode	Uart4 4wire mode select: Yes or No Default: No
Uart3 channel	Uart3 channel select : Yes or No Default: No
Uart3 4wire mode	Uart3 4wire mode select: Yes or No Default: No
Uart2 channel	Uart2 channel select : Yes or No Default: No
Uart2 4wire mode	Uart2 4wire mode select: Yes or No Default: No
Uart1 channel	Uart1 channel select : Yes or No Default: No
Uart1 4wire mode	Uart1 4wire mode select: Yes or No Default: No

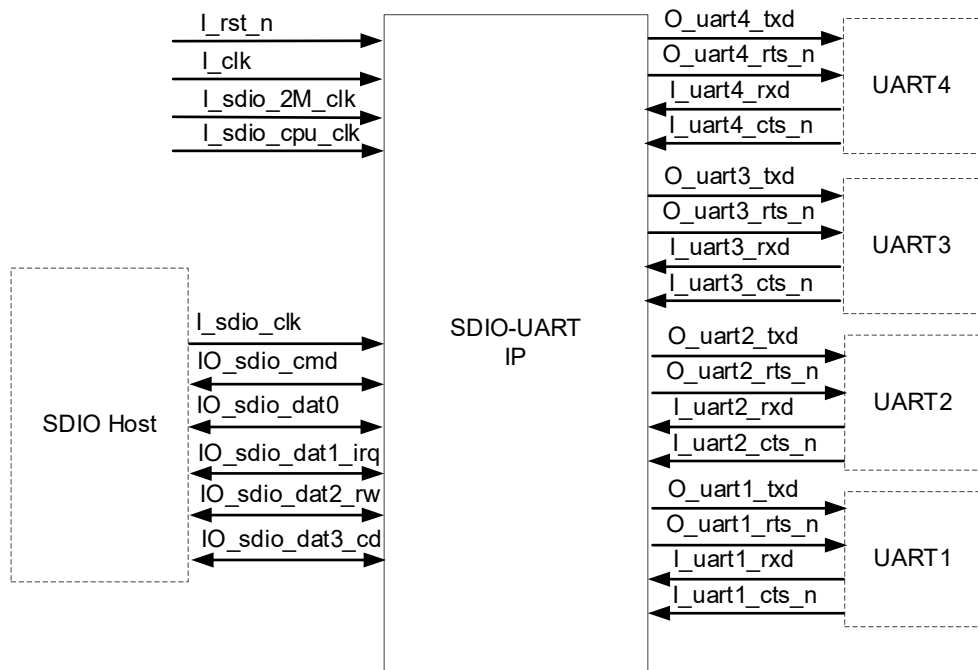
8 界面配置

8.1 概述

主要介绍如何通过 GUI 界面设置参数、生成可用 SDIO-UART Module。

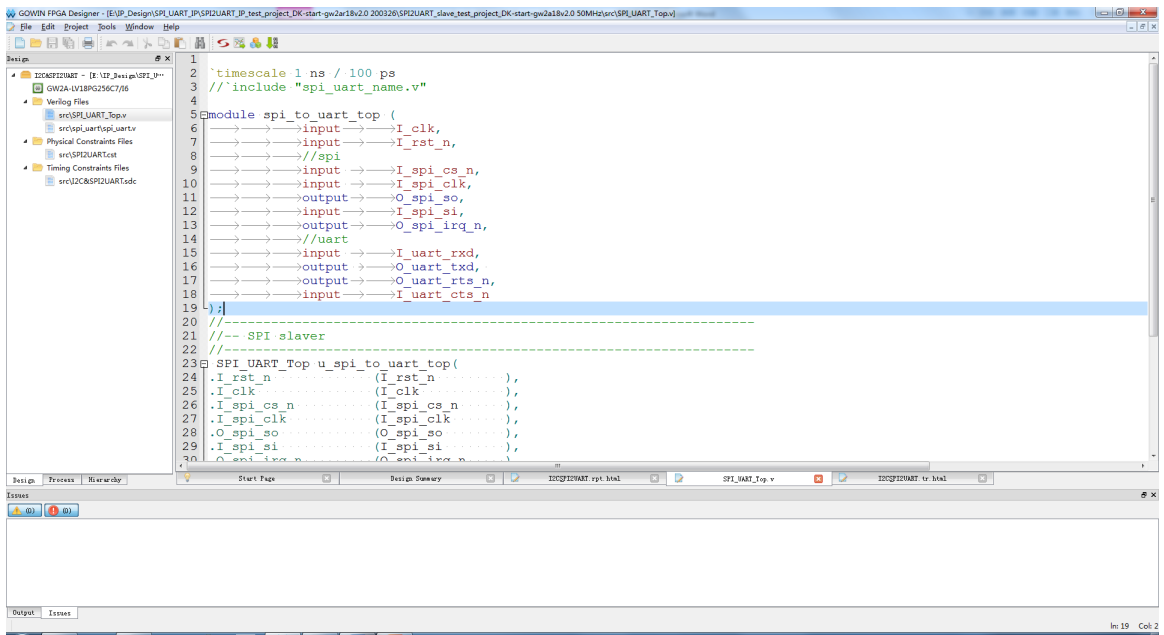
8.2 应用框图

图 8-1 应用框图



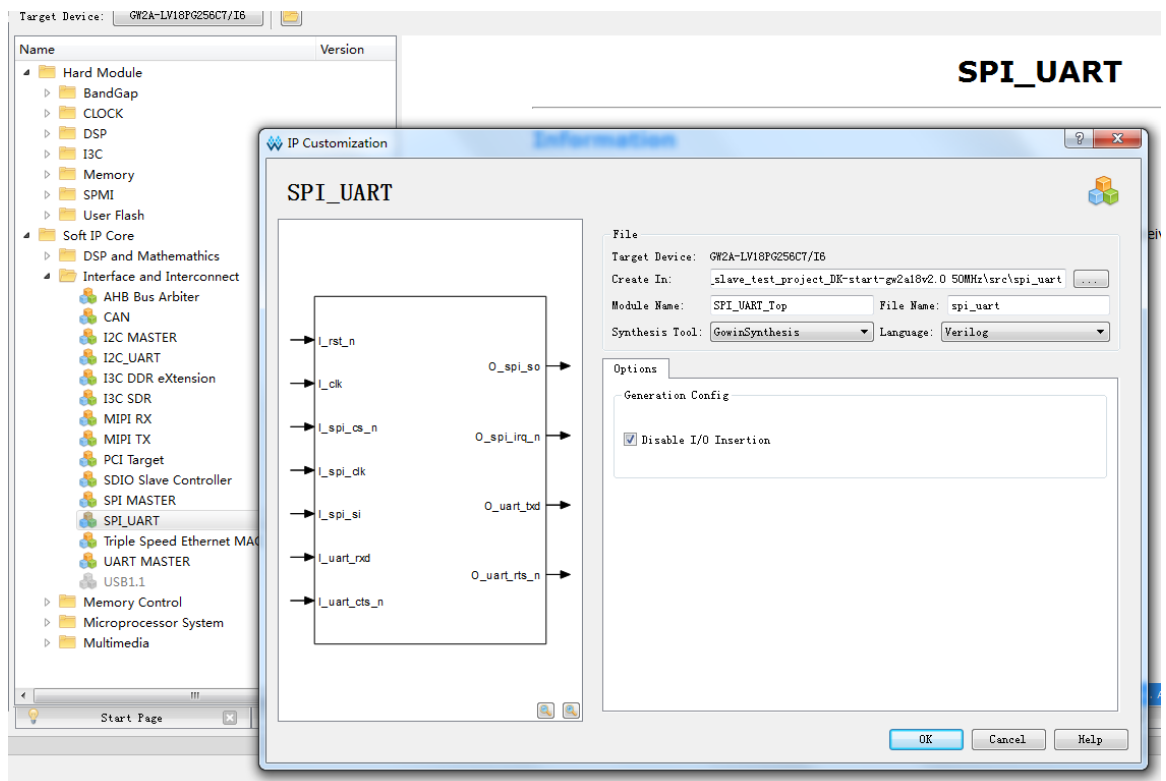
8.3 GUI 调用

1. 启动 GowinYunYuan 软件后，打开工程，如下图：

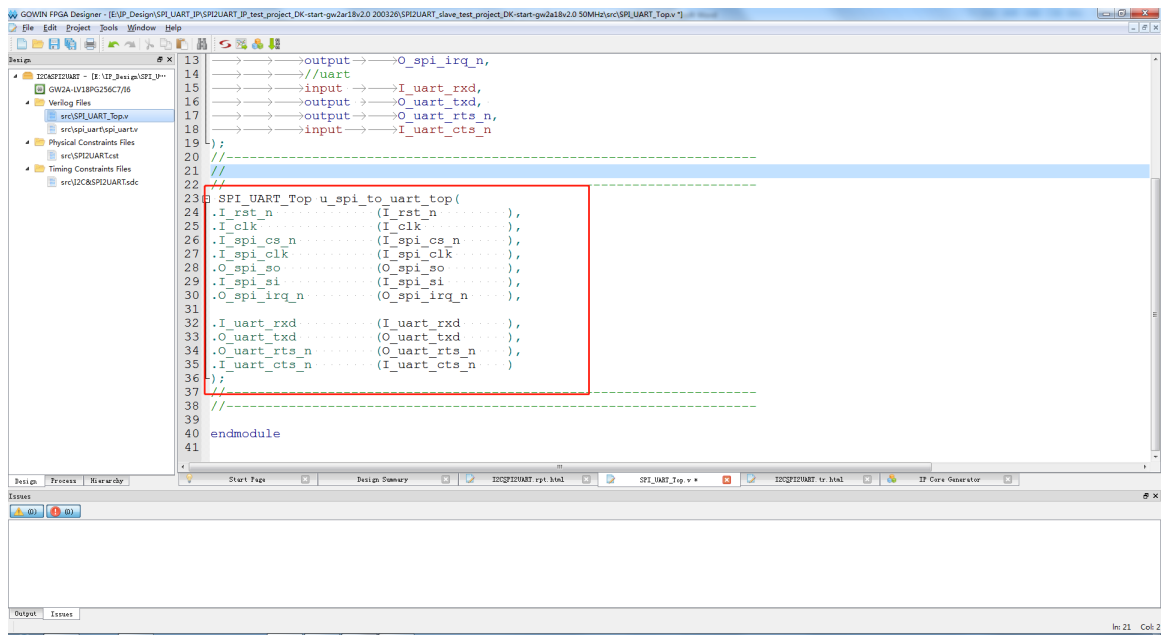


2. 在 Tools 菜单下点击 “IP Core Generator”，如下图：

调用的 SDIO_UART ， 然后点击 “OK” 后生成 SDIO_UART Module。



3. 在用户程序中例化 SDIO_UART，如下图：



```

13  >>>output->>>O_spi_irq_n,
14  >>>//uart
15  >>>input->>>I_uart_rxd,
16  >>>output->>>O_uart_txd,
17  >>>output->>>O_uart_rts_n,
18  >>>input->>>I_uart_cts_n
19  -);
20  //-----
21  //
22  //
23  SPI_UART_Top u_spi_to_uart_top(
24  .I_rst_n          (I_rst_n          ),
25  .I_clk            (I_clk            ),
26  .I_spi_cs_n      (I_spi_cs_n      ),
27  .I_spi_clk       (I_spi_clk       ),
28  .O_spi_so        (O_spi_so        ),
29  .I_spi_si        (I_spi_si        ),
30  .O_spi_irq_n     (O_spi_irq_n     ),
31
32  .I_uart_rxd      (I_uart_rxd      ),
33  .O_uart_txd      (O_uart_txd      ),
34  .O_uart_rts_n    (O_uart_rts_n    ),
35  .I_uart_cts_n    (I_uart_cts_n    )
36  -);
37  //-----
38  //
39
40
41  endmodule

```

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

