




Gowin SDIO SDR104 Slave Controller IP 用户指南

IPUG774-1.0.1, 2023/11/16

版权所有 © 2023 广东高云半导体科技股份有限公司

GOWIN高云、、Gowin、GowinSynthesis、云源以及高云均为广东高云半导体科技股份有限公司注册商标，本手册中提到的其他任何商标，其所有权利属其拥有者所有。未经本公司书面许可，任何单位和个人都不得擅自摘抄、复制、翻译本档内容的部分或全部，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，或以禁止反言或其它方式授予任何知识产权许可。除高云半导体在其产品的销售条款和条件中声明的责任之外，高云半导体概不承担任何法律或非法律责任。高云半导体对高云半导体产品的销售和 / 或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性、适销性或对任何专利权、版权或其它知识产权的侵权责任等，均不作担保。高云半导体对档中包含的文字、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
2021/05/07	1.0	初始版本。
2023/11/16	1.0.1	<ul style="list-style-type: none">● 更新表 4-2 CPU Slave Interface 寄存器定义;● 更新界面截图。

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	2
2 概述	3
3 特征与性能	4
3.1 主要特征	4
3.2 工作频率	5
3.3 资源利用	5
4 功能描述	6
4.1 整体结构	6
4.2 Gowin SDIO 接口定义	7
4.3 Non UHS Card 和 UHS-I Card	8
4.3.1 Non UHS Card	8
4.3.2 UHS-I Card	8
4.4 Gowin SDIO IP 初始化	9
4.4.1 Non UHS Card 初始化流程	10
4.4.2 UHS-I Card 初始化流程	11
4.5 CIA (Function0) 寄存器	12
4.6 CMD52	12
4.7 CMD53	12
4.8 Abort	12
4.9 CMD19	12
4.10 中断功能	13
4.10.1 生成中断	13
4.10.2 清除中断	13
4.11 用户接口	14

4.11.1 CPU Slave Interface	14
4.11.2 SDIO CMD52 Interface	21
4.11.3 SDIO CMD53 Interface	24
4.11.4 SDIO Tuning Interface	29
4.12 IP 初始化流程	30
5 端口列表	31
6 参数配置	36
7 界面配置	37
8 参考设计	39

图目录

图 4-1 Gowin SDIO SDR104 Slave Controller IP 结构图.....	6
图 4-2 SDIO 管脚连接示意图	8
图 4-3 Non UHS Card 初始化流程	10
图 4-4 UHS-I Card 初始化流程.....	11
图 4-5 Tuning Block Pattern	13
图 4-6 CPU Slave Interface 接口写时序.....	19
图 4-7 CPU Slave Interface 接口读时序.....	20
图 4-8 CPU Slave Interface 读写交替操作时序.....	20
图 4-9 SDIO CMD52 Interface 写操作 (RAW flag = 1) 时序.....	22
图 4-10 SDIO CMD52 Interface 接口写操作 (RAW flag = 0) 时序	23
图 4-11 SDIO CMD52 Interface 接口读操作时序	24
图 4-12 SDIO CMD53 Interface 接口 4-bit 模式写操作时序.....	25
图 4-13 SDIO SDR104 CMD53 Interface 接口 4-bit 模式读操作	26
图 4-14 Write Bad Abort	27
图 4-15 Write Good Abort Case1	27
图 4-16 Write Good Abort Case2	28
图 4-17 Read Bad Abort	28
图 4-18 Read Good Abort Case1	29
图 4-19 Read Good Abort Case2	29
图 4-20 SDIO Tuning Interface 时序.....	29
图 7-1 IP 核产生工具	37
图 7-2 SDIO SDR104 Slave Controller IP 配置界面	38

表目录

表 1-1 术语、缩略语	1
表 2-1 Gowin SDIO SDR104 Slave Controller IP	3
表 3-1 资源利用	5
表 4-1 SDIOSDR104 Slave Controller IP 接口定义与 SD 标准接口定义对应关系	7
表 4-2 CPU Slave Interface 寄存器定义	14
表 5-1 Gowin SDIO SDR104 Slave Controller IP IO 端口	31
表 6-1 Gowin SDIO SDR104 Slave Controller 静态参数与时序参数	36

1 关于本手册

1.1 手册内容

Gowin® SDIO SDR104 Slave Controller IP 用户指南主要包括特征性能、功能描述、端口列表、参数配置、界面配置以及参考设计，旨在帮助用户快速了解 Gowin SDIO SDR104 Slave Controller IP 的产品特性及使用方法。本手册中的软件界面截图参考的是 1.9.9 Beta-6 版本，因软件版本升级，部分信息可能会略有差异，具体以用户软件版本的信息为准。

1.2 相关文档

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

- [DS100, GW1N 系列 FPGA 产品数据手册](#)
- [DS117, GW1NR 系列 FPGA 产品数据手册](#)
- [DS821, GW1NS 系列 FPGA 产品数据手册](#)
- [DS861, GW1NSR 系列 FPGA 产品数据手册](#)
- [DS891, GW1NSE 系列安全 FPGA 产品数据手册](#)
- [DS881, GW1NSER 系列安全 FPGA 产品数据手册](#)
- [DS102, GW2A 系列 FPGA 产品数据手册](#)
- [DS226, GW2AR 系列 FPGA 产品数据手册](#)
- [SUG100, Gowin 云源软件用户指南](#)

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
CSA	Code Storage Area	代码存储区
DDR	Double Data Rate	双倍数据率

术语、缩略语	全称	含义
IP	Intellectual Property	知识产权
LUT	Look-up Tables	查找表
SDIO	Secure Digital Input and Output	安全数字输入输出
SDR	Single Data Rate	单倍数据率
UHS	Ultra High Speed	超高速

1.4 技术支持与反馈

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

网址：www.gowinsemi.com.cn

E-mail：support@gowinsemi.com

Tel: 86 755 8262 0391

2 概述

本手册描述了 Gowin SDIO SDR104 Slave Controller IP，为用户提供一个支持 SDR104 速率的 SDIO Slave 的通用接口。用户可以把此 IP 集成到一个 SDIO Device 设计中，方便实现 SDIO Device 功能。它可以集成到需要 SDIO Slave 的设备中，这种连接可用于各种应用。

表 2-1 Gowin SDIO SDR104 Slave Controller IP

Gowin SDIO SDR104 Slave Controller IP	
逻辑资源	见表 3-1。
交付文件	
设计文件	Verilog (加密)
参考设计	Verilog
测试平台	Verilog
测试设计流程	
综合软件	GowinSynthesis®
应用软件	Gowin Software (V1.9.7.03Beta 及以上)

3 特征与性能

3.1 主要特征

- 支持 SD Physical Layer Specification Version 4.00
- 支持 SDIO Specification Version 4.00
- 支持 SD 传输模式，其中 SD 传输模式支持 4 位模式
- 不支持 SPI 传输模式
- 支持 UHS-I 模式，支持 SDR104 速率
- UHS-I 模式支持 1.8V 电压，不支持电压切换
- 支持 SDIO Command 类型: CMD0, CMD3, CMD5, CMD7, CMD11, CMD15, CMD19, CMD52, CMD53
- 支持 CRC7 和 CRC16 自动生成和校验
- 支持 Function0 和 Function1
- 支持用户侧 CPU Slave Interface 配置 IP
- 支持用户侧 CMD52 和 CMD53 接口
- 支持 4-bit 模式中断，但中断不支持 Data Block Gap 和 Asynchronous Interrupt
- 支持读写 Abort 功能
- 不支持 Suspend/Resume 功能
- 不支持 Read Wait 功能
- 不支持 Combo Card
- 不支持 CSA
- 不支持 DDR 模式
- 不支持 Card Detection

3.2 工作频率

Gowin SDIO SDR104 Slave Controller IP 的工作频率取决于 SDIO Host 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 资源利用

Gowin SDIO SDR104 Slave Controller IP 采用 Verilog 语言。不同的模式有不同的资源消耗。以 Gowin GW2A-18 器件为例，其资源利用情况如表 3-1，关于其它器件的资源利用请参阅相关的后期发布信息。

表 3-1 资源利用

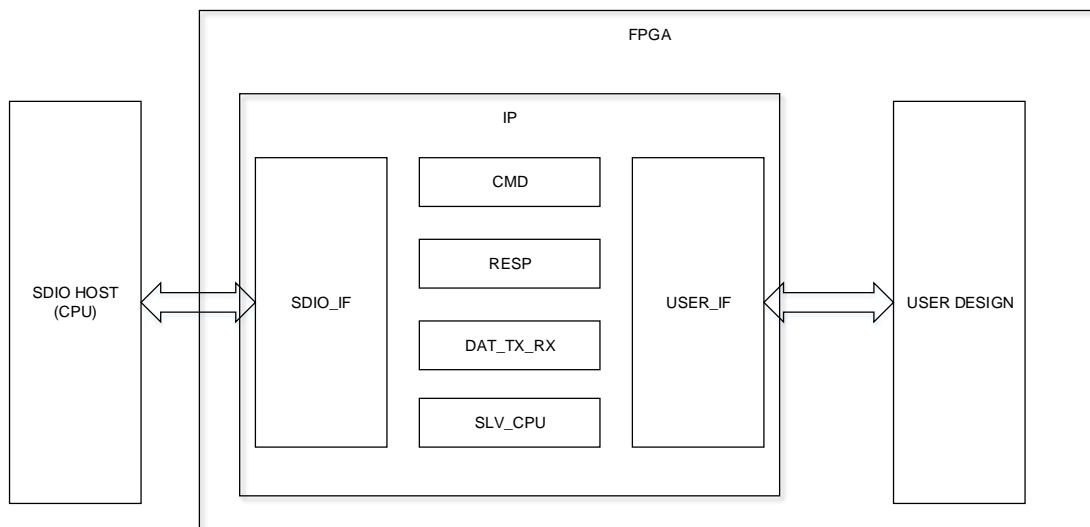
Type	LUTs	REGs	Device	Speed Grade
Non UHS	3193	1667	GW2A-18	-8
UHS-I	3043	1695	GW2A-18	-8

4 功能描述

4.1 整体结构

Gowin SDIO SDR104 Slave Controller IP 基本结构如图 4-1 所示，主要包含 SDIO_IF、CMD、RESP、DAT_TX_RX、SLV_CPU、USER_IF 等模块。图 4-1 中的 User Design 是 FPGA 中的用户设计，SDIO HOST 是 SDIO 主机。

图 4-1 Gowin SDIO SDR104 Slave Controller IP 结构图



- SDIO_IF 模块实现 SDIO 接口数据选择及双向传输功能
- CMD 模块实现 SDIO Command 解析功能
- RESP 模块实现 SDIO Response 生成功能
- DAT_TX_RX 模块实现 CMD19 和 CMD53 数据读写功能
- SLV_CPU 模块实现用户配置 IP 功能
- USER_IF 模块实现用户侧与 IP 交换数据功能

4.2 Gowin SDIO 接口定义

Gowin SDIO SDR104 Slave Controller IP 接口定义与 SD 标准接口定义对应关系如表 4-1 所示。

表 4-1 SDIOSDR104 Slave Controller IP 接口定义与 SD 标准接口定义对应关系

Pin	SD 标准				IP
	SD 4-bit mode		SD 1-bit mode		
1	CD/DAT[3]	Data line 3	N/C	Not Used	sdio_dat3_in sdio_dat3_out sdio_dat3_oen
2	CMD	Command line	CMD	Command line	sdio_cmd_in sdio_cmd_out sdio_cmd_oen
3	VSS1	Ground	VSS1	Ground	-
4	VDD	Supply voltage	VDD	Supply voltage	-
5	CLK	Clock	CLK	Clock	sdio_clk
6	VSS2	Ground	VSS2	Ground	-
7	DAT[0]	Data line 0	DATA	Data line	sdio_dat0_in sdio_dat0_out sdio_dat0_oen
8	DAT[1]	Data line 1 or Interrupt	IRQ	Interrupt	sdio_dat1_in sdio_dat1_out sdio_dat1_oen
9	DAT[2]	Data line 2 or Read Wait	RW	Read Wait	sdio_dat2_in sdio_dat2_out sdio_dat2_oen

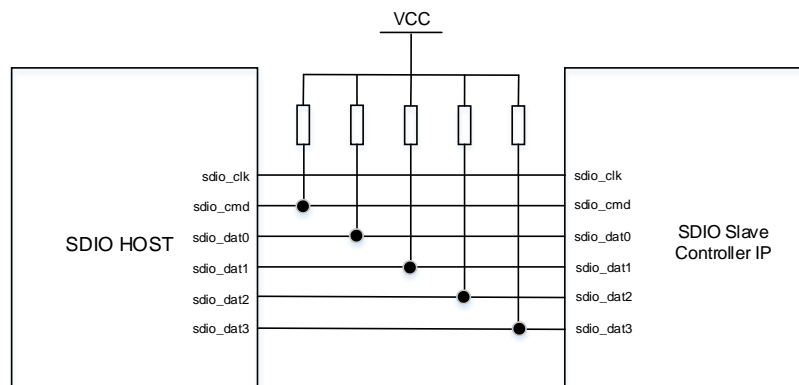
用户可通过以下语句把 IP 定义的 SDIO 接口转化为 INOUT 格式。

```
assign sdio_cmd_in = sdio_cmd;
assign sdio_cmd = (!sdio_cmd_oen) ? sdio_cmd_out : 1'bz;
assign sdio_dat0_in = sdio_dat0;
assign sdio_dat0 = (!sdio_dat0_oen) ? sdio_dat0_out : 1'bz;
assign sdio_dat1_in = sdio_dat1;
assign sdio_dat1 = (!sdio_dat1_oen) ? sdio_dat1_out : 1'bz;
assign sdio_dat2_in = sdio_dat2;
assign sdio_dat2 = (!sdio_dat2_oen) ? sdio_dat2_out : 1'bz;
assign sdio_dat3_in = sdio_dat3;
assign sdio_dat3 = (!sdio_dat3_oen) ? sdio_dat3_out : 1'bz。
```

电路连接

在电路连接时，CMD 和 DAT 线需要通过上拉电阻连接到 VCC。当 IP 设置为 Non UHS Card 时，VCC 为 3.3V；当 IP 设置为 UHS-I Card 时，VCC 为 1.8V，参考电路如图 4-2 所示。

图 4-2 SDIO 管脚连接示意图



4.3 Non UHS Card 和 UHS-I Card

IP 支持两种 Card 类型：Non UHS Card 和 UHS-I Card。在生成 IP 时，用户需要通过界面配置。

4.3.1 Non UHS Card

当 IP 工作在 Non UHS Card 时，Bank 电压和 SDIO 接口上拉电阻电压应为 3.3V。

Non UHS Card 支持 Low-Speed 和 Full-Speed。其中 Full-Speed 支持 Default-Speed 和 High-Speed。

用户配置 IP 为 Non UHS Card 后，在初始化阶段，无论 IP 收到 CMD5 S18R 为 0 还是 1，IP 回应 R4 中 S18A 始终为 0。当 IP 工作在 Non UHS Card 时，IP 不支持 CMD11 和 CMD19。

4.3.2 UHS-I Card

当 IP 工作在 UHS-I Card 时，Bank 电压和 SDIO 接口上拉电阻电压应为 1.8V。

UHS-I Card 支持 SDR12、SDR25、SDR50 以及 SDR104。

用户配置 IP 为 UHS-I Card 后，在初始化阶段，当 IP 收到的 CMD5 S18R 为 0 时，IP 回应 R4 中 S18A 为 0；当 IP 收到的 CMD5 S18R 为 1 时，IP 回应 R4 中 S18A 为 1。当 IP 工作在 UHS-I Card 时，IP 支持 CMD11 和 CMD19。需要注意的是，当 IP 收到 CMD11 会回应 R1，并支持 Voltage Switch Sequence，但是不改变 IO 电压。

4.4 Gowin SDIO IP 初始化

IP 在上电之后需要被 SDIO Host 初始化，之后才可以正常工作。初始化流程遵循 SDIO 标准。初始化时钟频率范围 100KHz~400KHz。

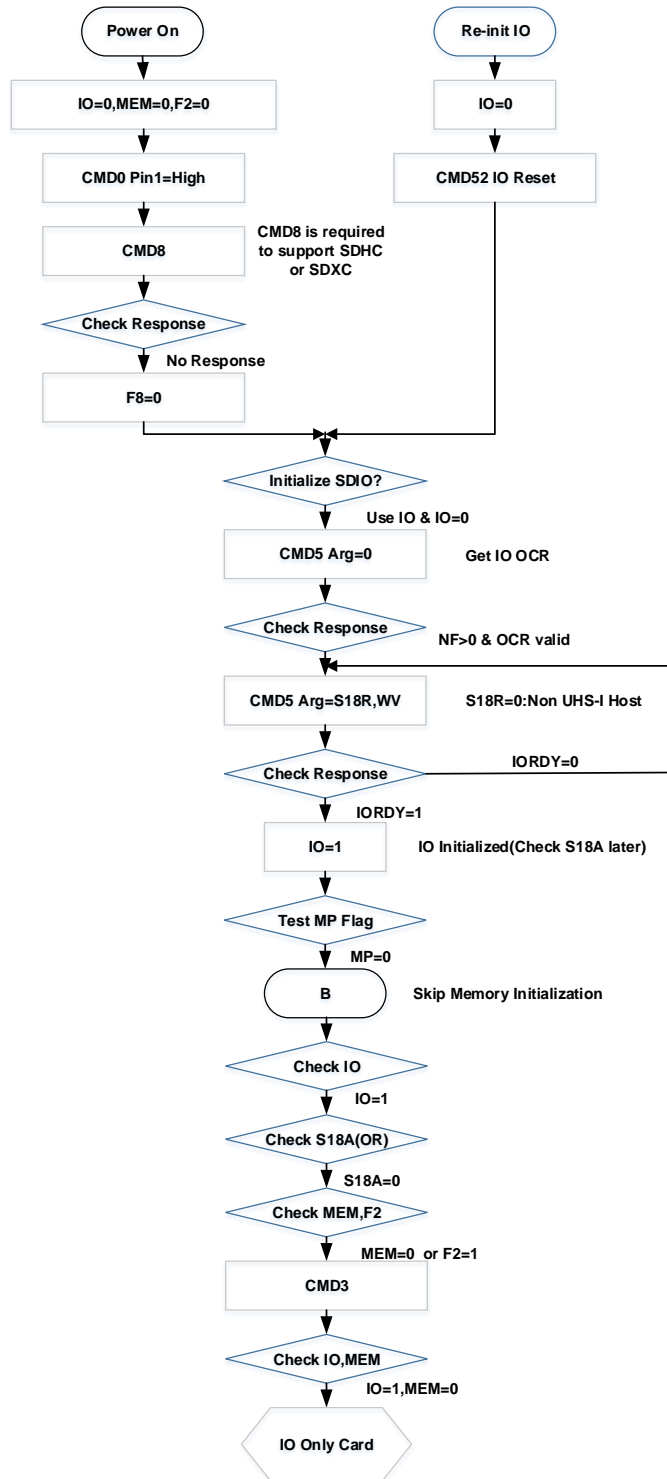
由于 IP 不支持 SPI 模式，因此在初始化接收到 CMD0 时，无论 `sdio_dat3` 是否为高电平，IP 都会保持在 SD mode。

在 IP 设置为 UHS-I Card 时，IP 支持接收 CMD11 命令进入 Voltage Switch Sequence，但是不改变 IO 电压。当 Voltage Switch Sequence 结束后，IP 进入 UHS-I 状态。

4.4.1 Non UHS Card 初始化流程

Non UHS Card 初始化流程如图 4-3 所示。

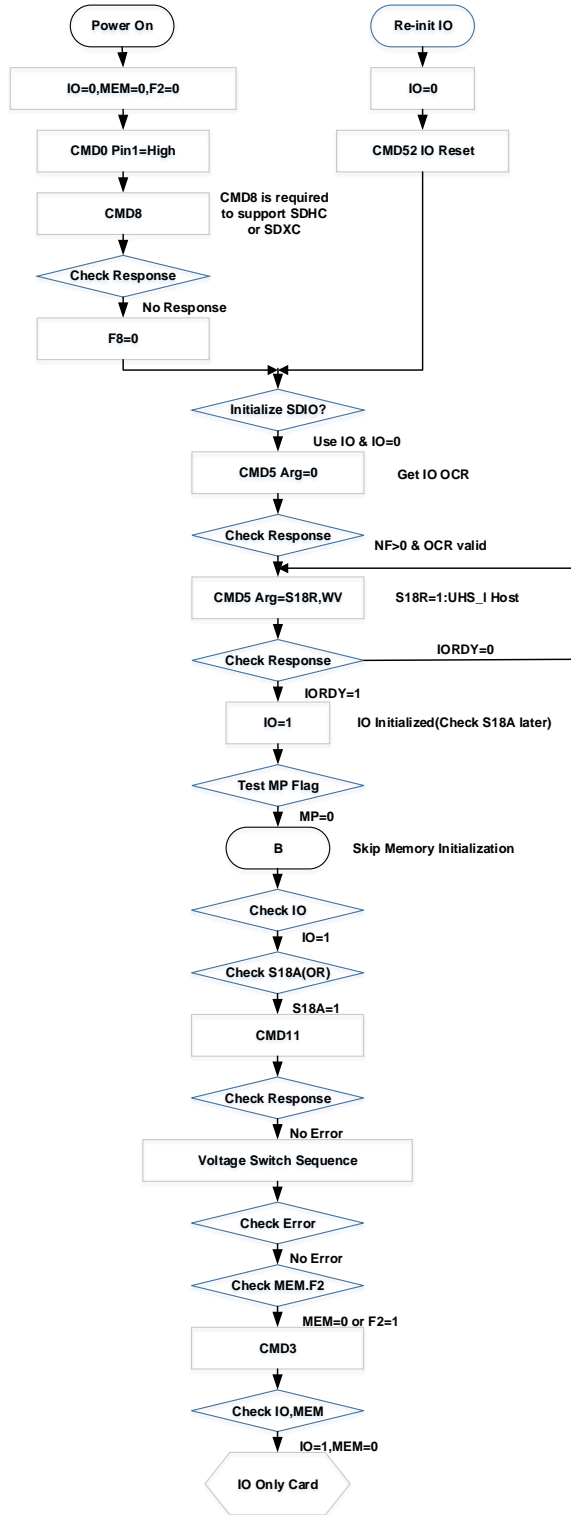
图 4-3 Non UHS Card 初始化流程



4.4.2 UHS-I Card 初始化流程

UHS-I Card 初始化流程如图 4-4 所示。

图 4-4 UHS-I Card 初始化流程



4.5 CIA (Function0) 寄存器

SDIO 标准定义 CIA 寄存器, 允许 Host 读取 Slave 状态和配置 Slave 功能。Host 可通过 CMD52 和 CMD53 命令读写 CIA 寄存器。

CIA 寄存器包括 CCCR 寄存器、FBR 寄存器、CIS 寄存器和 CSA 寄存器。

CCCR 寄存器和 FBR (Function1) 寄存器已在 IP 中实现, 用户在初始化阶段, 可通过 CPU Slave Interface 访问相应寄存器。当 Host 读写此部分寄存器时, 用户无需参与。

FBR (Function2~7) 寄存器暂时不支持访问。

CIS 寄存器没有在 IP 中实现, 当 Host 访问此部分寄存器时, 需要用户在 SDIO CMD52 Interface 或 SDIO CMD53 Interface 提供数据。

CSA 寄存器暂时不支持访问。

4.6 CMD52

当 IP 进入 Command State 后, Host 可以通过 CMD52 以单字节的方式访问 IP 寄存器。由于 IP 支持 Function0 和 Function1, 因此, IP 仅对 Host 针对于 Function0 和 Function1 的 CMD52 有响应。

当 Host 通过 CMD52 访问 CCCR 和 FBR 寄存器时, IP 自动响应, 用户无需参与。

当 Host 通过 CMD52 访问 CIS 和 Function1 寄存器时, 用户需要通过 SDIO CMD52 Interface 参与数据的收发。

4.7 CMD53

当 IP 进入 Command State 后, Host 可以通过 CMD53 命令连续访问 IP 寄存器。由于 IP 支持 Function0 和 Function1, 因此, IP 仅对 Host 针对于 Function0 和 Function1 的 CMD53 有响应。

当 Host 通过 CMD53 访问 CCCR 和 FBR 寄存器时, IP 自动响应, 用户无需参与。

当 Host 通过 CMD53 访问 CIS 和 Function1 寄存器时, 用户需要通过 SDIO CMD53 Interface 参与数据的收发。

4.8 Abort

当 Host 发送 CMD53 以 Infinite Block Mode 读写数据时, 可以通过 CMD52 发送 Abort 命令来中断读写过程。

4.9 CMD19

当 IP 被配置为 UHS-I Card 时, 支持 CMD19。IP 接收到正确的 CMD19 后, 会自动回应 R1。用户需要通过 SDIO Tuning Interface 来提供 Tuning Block Pattern。当 IP 被配置为 Non UHS Card 时, 用户无需关注 SDIO Tuning

Interface。Tuning Block Pattern 共 144 个半字节，如图 4-5 所示。

图 4-5 Tuning Block Pattern

```
FF0FFF00_FFCCC3CC_C33CCCFE_FFFFFFFF
FFDFFFDD_FFFBFFFB_BFFF7FFF_77F7BDEF
FFF0FFF0_0FFCCC3C_CC33CCCF_FFEFFFFE
FFFDFFFD_DFFFBFFF_BBFFF7FF_F77F7BDE
F9503A4B_C5488FBC
```

用户需要按照从左至右、从上至下的顺序依次发送。

4.10 中断功能

IP 支持 4-bit 模式下的中断。

4.10.1 生成中断

在 4-bit 模式下，DAT1 作为数据和中断复用信号线。当用户拉高 fun1_interrupt 管脚为 1 时，IP 在标准定义的 Interrupt Period 时刻把 DAT1 拉低产生中断。4-bit 模式中断不支持 Data Block Gap 和 Asynchronous Interrupt。4-bit 模式下的中断信号与 sdio_clk 同步。因此，当 sdio_clk 停止后，IP 无法在 DAT1 产生中断信号。当 IP 工作在 4-bit 模式时，如果在 SDIO Host 设计中需要停止时钟，且需要中断来唤起时钟，可通过以下方式实现：

1. Host 通过 CMD52 设置 IP 为 1-bit 模式；
2. Host 停止时钟；
3. 用户拉高 fun1_interrupt 管脚为 1，在 1-bit 模式下产生中断；
4. Host 开始提供时钟；
5. Host 通过 CMD52 设置 IP 为 4-bit 模式；
6. Host 开始访问 IP。

4.10.2 清除中断

当用户检测到 SDIO Host 访问清除中断寄存器时，用户需拉低 fun1_interrupt 管脚为 0，以清除中断。清除中断寄存器由用户自定义。

SDIO 标准建议清除中断时间：

- 若用户通过 CMD52 访问来清除中断时，需要在 R5 结束之前清除中断；
- 若用户通过 CMD53 访问来清除中断时，需要在数据完成传输之前清除中断。

4.11 用户接口

4.11.1 CPU Slave Interface

CPU Slave Interface 时钟

CPU Slave Interface 时钟为 `cpu_clk`，与 `sdio_clk` 相互独立。在正常使用中，`cpu_clk` 可以连接到 `sdio_clk`，以达到两个接口时钟同步的目的。

CPU Slave Interface 功能

CCCR 寄存器和 FBR（Function1）寄存器已在 IP 中实现，用户可通过 CPU Slave Interface 访问相应寄存器。当 Host 访问此部分寄存器时，用户无需参与。另外，用户也可以通过 CPU Slave Interface 设置 IP 的其他功能。具体寄存器定义如表 4-2 所示。

CPU Slave Interface 数据接口为 32 位，地址按照 4 字节对齐。

CPU Slave Interface 寄存器定义

CPU Slave Interface 寄存器定义如表 4-2 所示。

表 4-2 CPU Slave Interface 寄存器定义

寄存器地址	Bit	名称	默认值	类型	描述
0x00	3:0	CCCR/FBR Format Version number	0x3	RO	CCCR/FBR 版本 对应 CCCR Reg0x0 bit 3~bit 0
	7:4	SDIO Specification Revision Number	0x5	RO	SDIO 版本 对应 CCCR Reg0x0 bit 7~bit 4
	11:8	SD Format Version Number	0x4	RO	SD 版本 对应 CCCR Reg0x1 bit 3~bit 0
	16	Support Master Power Control	0x0	RW	主机功率控制 对应 CCCR Reg0x12 bit0
	17	Enable Master Power Control	0x0	RO	主机功率控制使能 对应 CCCR Reg0x12 bit 1
	20:18	Total Card Power	0x0	RO	卡总功率 对应 CCCR Reg0x12 bit 4~bit 2
	24	Support High-Speed	0x1	RW	高速模式支持 对应 CCCR Reg0x13 bit0
	27:25	Bus Speed Select	0x0	RO	总线速度模式选择 对应 CCCR Reg0x13 bit 3~bit 1
0x04	0	Interrupt Enable Master	0x0	RO	总中断使能 对应 CCCR Reg0x4 bit0
	7:1	Interrupt Enable for Function x	0x00	RO	Function x 中断使能 对应 CCCR Reg0x4 bit 7~bit 1
	9:8	Bus Width	0x0	RO	总线宽度 对应 CCCR Reg0x7 bit1~bit0
	10	Support 8-bit Bus	0x0	RO	支持 8bit 总线模式

寄存器地址	Bit	名称	默认值	类型	描述
		Mode			对应 CCCR Reg0x7 bit 2
	13	Enable Continuous SPI Interrupt	0x0	RO	连续 SPI 中断使能 对应 CCCR Reg0x7 bit 5
	14	Support Continuous SPI interrupt	0x1	RW	连续 SPI 中断支持 对应 CCCR Reg0x7 bit6
	15	Card Detect Disable	0x0	RO	禁止卡检测 对应 CCCR Reg0x7 bit 7
	16	Support Direct Command (CMD52)	0x1	RO	支持 Direct 命令 (CMD52) 对应 CCCR Reg0x8 bit 0
	17	Support Multiple Block Transfer	0x1	RO	支持多块传输 对应 CCCR Reg0x8 bit 1
	18	Support Read Wait	0x0	RO	支持读等待 对应 CCCR Reg0x8 bit 2
	19	Support Bus Control	0x0	RO	总线控制支持 对应 CCCR Reg0x8 bit 3
	20	Support Block Gap Interrupt	0x0	RO	支持数据块间中断 对应 CCCR Reg0x8 bit 4
	21	Enable Block Gap Interrupt	0x0	RO	使能数据块间中断 对应 CCCR Reg0x8 bit 5
	22	Low-Speed Card	0x0	RW	低速卡 对应 CCCR Reg0x8 bit 6
	23	4-bit Mode Support for Low-Speed Card	0x0	RW	低速卡支持 4bit 模式 对应 CCCR Reg0x8 bit 7
	24	Support SDR50	0x0	RW	支持 SDR50 对应 CCCR Reg0x14 bit 0
	25	Support SDR104	0x1	RW	支持 SDR104 对应 CCCR Reg0x14 bit 1
	26	Support DDR50	0x0	RW	支持 DDR50 对应 CCCR Reg0x14 bit 2
	27	Support Asynchronous Interrupt	0x0	RO	支持异步中断 对应 CCCR Reg0x16 bit 0
	28	Enable Asynchronous Interrupt	0x0	RO	使能异步中断 对应 CCCR Reg0x16 bit 1
0x08	23:0	Common CIS Pointer	0x001000	RW	Common CIS 指针 [23:16]对应 CCCR Reg0x0B bit 7~bit 0 [15:8]对应 CCCR Reg0x0A bit 7~bit 0 [7:0]对应 CCCR Reg0x09 bit 7~bit 0
	24	Support Driver Type A	0x0	RW	支持驱动类型 A 对应 CCCR Reg0x15 bit 0

寄存器地址	Bit	名称	默认值	类型	描述
	25	Support Driver Type C	0x0	RW	支持驱动类型 C 对应 CCCR Reg0x15 bit 1
	26	Support Driver Type D	0x0	RW	支持驱动类型 D 对应 CCCR Reg0x15 bit 2
	29:28	Driver Type Select	0x0	RO	驱动类型选择 对应 CCCR Reg0x15 bit 5~bit 4
0x0C	15:0	FN0 Block Size	0x0000	RO	Function 0 块大小 [15:8]对应 CCCR Reg0x11 bit 7~bit 0 [7:0]对应 CCCR Reg0x10 bit 7~bit 0
0x10	7:0	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF0 bit 7~bit 0
	15:8	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF1 bit 7~bit 0
	23:16	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF2 bit 7~bit 0
	31:24	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF3 bit 7~bit 0
0x14	7:0	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF4 bit 7~bit 0
	15:8	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF5 bit 7~bit 0
	23:16	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF6 bit 7~bit 0
	31:24	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF7 bit 7~bit 0
0x18	7:0	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF8 bit 7~bit 0
	15:8	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xF9 bit 7~bit 0
	23:16	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xFA bit 7~bit 0
	31:24	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xFB bit 7~bit 0
0x1C	7:0	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xFC bit 7~bit 0
	15:8	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xFD bit 7~bit 0
	23:16	Reserved for Vendors	0x00	RO	Vendors 保留 对应 CCCR Reg0xFE bit 7~bit 0
	31:24	Reserved for	0x00	RO	Vendors 保留

寄存器地址	Bit	名称	默认值	类型	描述
		Vendors			对应 CCCR Reg0xFF bit7~bit0
0x20	3:0	Standard SDIO Function Interface Code	0xF	RW	标准 SDIO 功能接口代码 对应 FBR1 Reg0x100 bit3~bit0
	15:8	Extended Standard SDIO Function Interface Code	0x00	RW	扩展标准 SDIO 功能接口代码 对应 FBR1 Reg0x101 bit7~bit0
	23:16	iSDIO Type Support Code	0x00	RW	iSDIO 类型支持代码 对应 FBR1 Reg0x108 bit7~bit0
	31:24	Standard iSDIO Function Interface Code	0x00	RW	标准 iSDIO 功能接口代码 对应 FBR1 Reg0x103 bit7~bit0
0x24	15:0	Function 1 I/O Block Size	0x0000	RO	Function 1 块大小 [15:8]对应 FBR1 Reg0x111 bit7~bit0 [7:0]对应 FBR1 Reg0x110 bit7~bit0
	16	Support Power Selection	0x0	RW	支持功率选择 对应 FBR1 Reg0x102 bit0
	20:17	Power State	0x0	RO	功率状态 对应 FBR1 Reg0x102 bit7~bit4
	21	Enable Power Selection	0x0	RO	功率选择使能 对应 FBR1 Reg0x102 bit1
	24	Function1 SupportsCSA	0x0	RO	支持 CSA 对应 FBR1 Reg0x100 bit6
	25	Function1 CSAEnable	0x0	RO	使能 CSA 对应 FBR1 Reg0x100 bit7
0x28	15:0	Function1 MID_CARD Manufacturer Information	0x0000	RW	Function1 MID_CARD 厂商信息 [15:8]对应 FBR1 Reg0x107 bit7~bit0 [7:0]对应 FBR1 Reg0x106 bit7~bit0
	31:16	Function1 SDA_MID_MANF SDIO Card Manufacturer Code	0x0000	RW	Function1 SDA_MID_MANFSDIO 卡厂商代码 [31:24]对应 FBR1 Reg0x105 bit7~bit0 [23:16]对应 FBR1 Reg0x104 bit7~bit0
0x2C	23:0	Address pointer to Function1 CIS	0x002000	RW	Function1 CIS 地址指针 [23:16]对应 FBR1 Reg0x10B bit7~bit0 [15:8]对应 FBR1 Reg0x10A bit7~bit0 [7:0]对应 FBR1 Reg0x109 bit7~bit0
0x30	0	IO_Ready	0x0	RW	SDIO Slave 准备完成标志。 此 bit 对应 R4 中 C 字段。上电后，此 bit 为 0。当用户通过 CPU Slave Interface 初始化完寄存器后，设置此 bit 为 1。
	18:16	SDIO Bus State	0x0	RO	SDIO Bus State。用户可以通过此寄存器读取 IP Bus State。

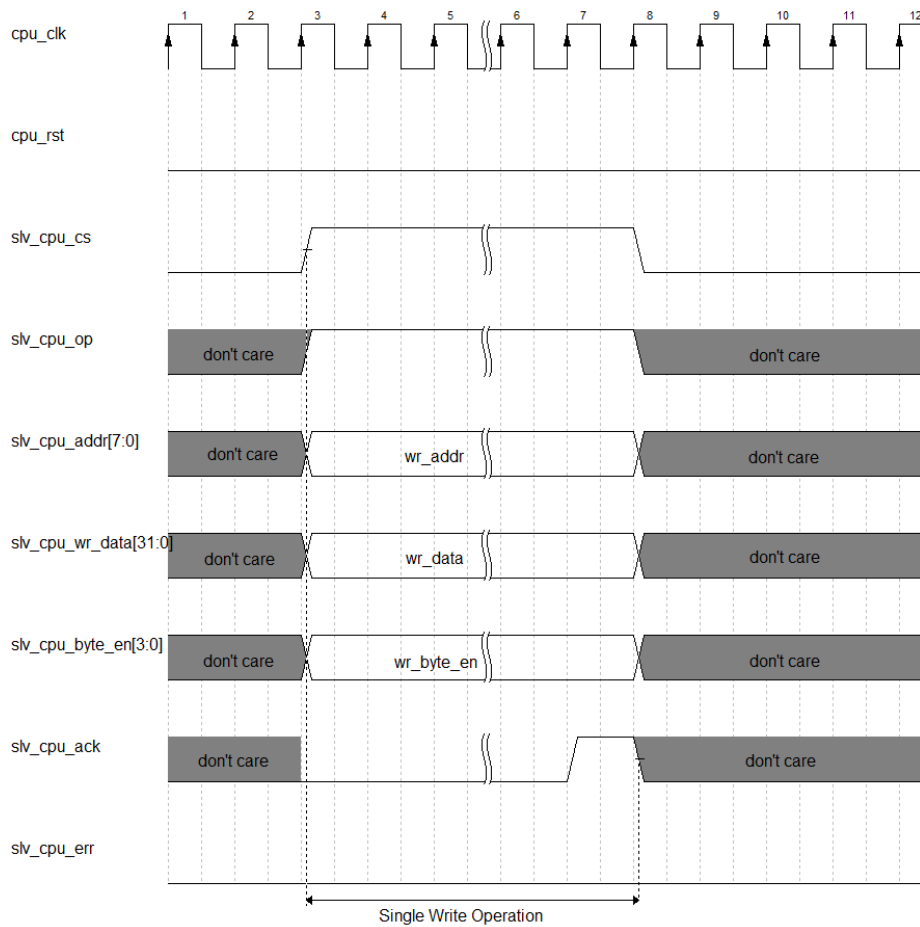
寄存器地址	Bit	名称	默认值	类型	描述
					0x0: Idle State, 表示 IP 未通过 CMD5 与 Host 完成协商; 0x1: Initialization State, 表示 IP 已通过 CMD5 与 Host 完成协商; 0x2: Standby State 0x3: Command State 0x4: Transfer State 0x5: Inactive State 其他: 未定义。
	24	Manual TX Clock to DataPhase	0x0	RW	SDIO 发送数据与 SDIO 时钟相位, 当 Manual TX Clock to DataPhase Enable 寄存器使能时有效。 0: SDIO 时钟下降沿发送数据; 1: SDIO 时钟上升沿发送数据。
	25	Manual TX Clock to Data Phase Enable	0x0	RW	SDIO 发送数据与 SDIO 时钟相位手动配置使能。 1: 使能, 按照 Manual TX Clock to DataPhase 值配置; 0: 禁止, 按照 SDIO 标准自动配置。
0x34	15:0	CCCR Max Block Size	0x0800	RW	CCCR 允许的最大 Block Size
	31:16	FBR1 Max Block Size	0x0800	RW	Function1 允许的最大 Block Size

CPU Slave Interface 操作时序

- CPU Slave Interface 写操作。

当用户需要通过 CPU Slave Interface 写寄存器时, 需要把 `slv_cpu_cs` 拉高来选中接口, 同时拉高 `slv_cpu_op` 表示写操作。用户通过 `slv_cpu_addr`、`slv_cpu_wr_data` 和 `slv_cpu_byte_en` 信号发送寄存器地址、写数据和字节使能。当 IP 完成此次写操作后, 会拉高 `slv_cpu_ack` 信号。`slv_cpu_cs`、`slv_cpu_op`、`slv_cpu_addr`、`slv_cpu_wr_data` 和 `slv_cpu_byte_en` 信号需保持不变, 直到 `slv_cpu_ack` 信号拉高。CPU Slave Interface 接口写时序如图 4-6 所示。

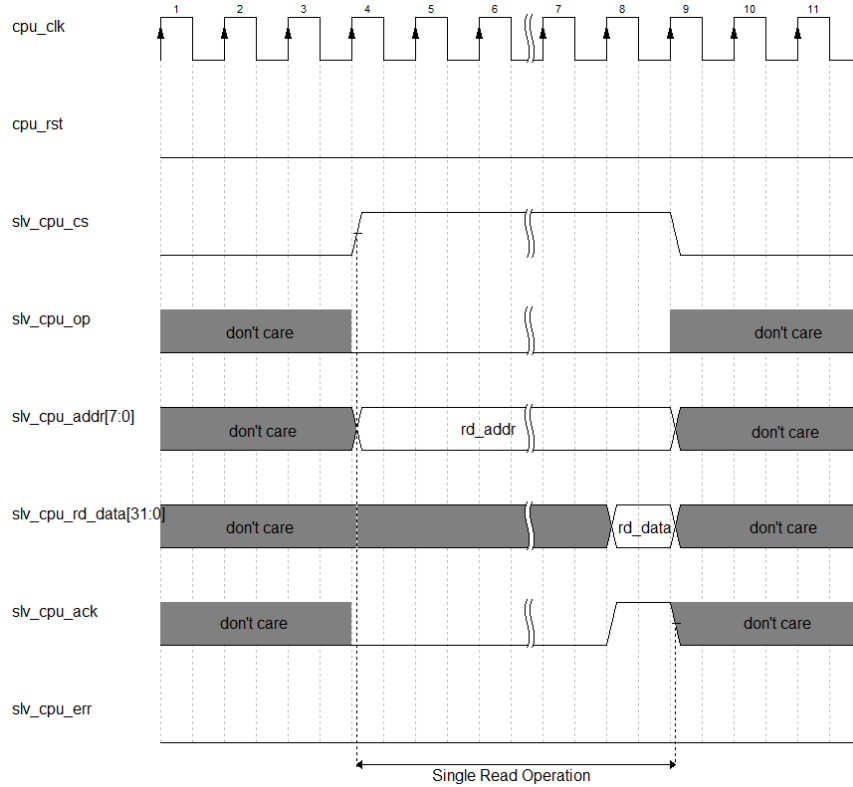
图 4-6 CPU Slave Interface 接口写时序



● CPU Slave Interface 读操作。

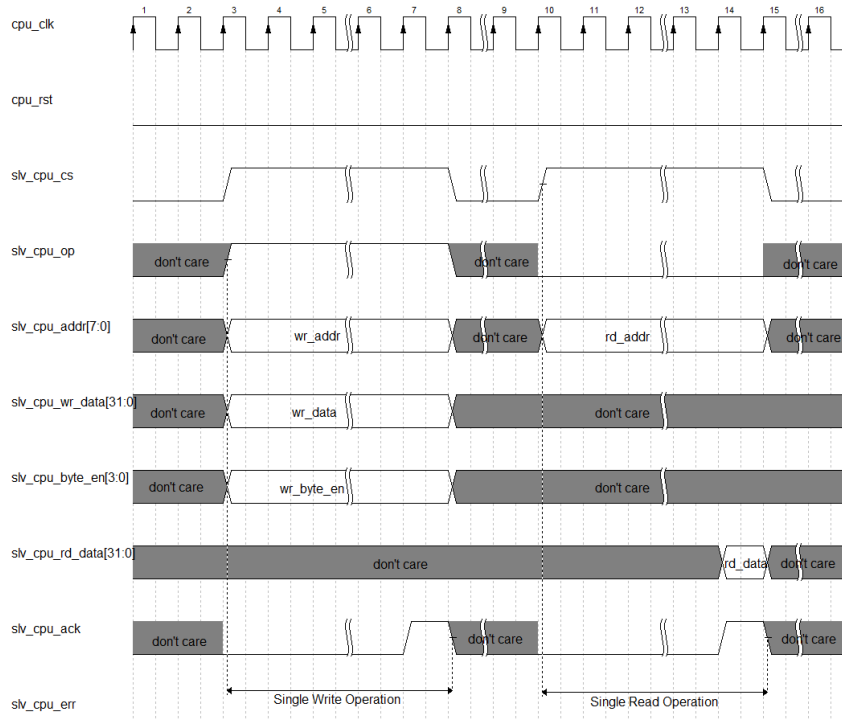
当用户需要通过 CPU Slave Interface 读寄存器时，需要把 **slv_cpu_cs** 拉高来选中接口，同时拉低 **slv_cpu_op** 表示读操作。用户通过 **slv_cpu_addr** 信号发送寄存器地址。当 IP 完成此次读操作后，会拉高 **slv_cpu_ack** 信号，同时读数据出现在 **slv_cpu_rd_data**。**slv_cpu_cs**、**slv_cpu_op**、**slv_cpu_addr** 信号需保持不变，直到 **slv_cpu_ack** 信号拉高。CPU Slave Interface 读操作时，IP 会返回 4 个字节的读数据。CPU Slave Interface 接口读时序如图 4-7 所示。

图 4-7 CPU Slave Interface 接口读时序



CPU Slave Interface 读写交替操作时序如图 4-8 所示。

图 4-8 CPU Slave Interface 读写交替操作时序



4.11.2 SDIO CMD52 Interface

当 Host 通过 CMD52 访问 CIS 或 Function1 寄存器时，用户需要通过 SDIO CMD52 Interface 控制数据的收发。IP 收到访问 CIS 或 Function1 寄存器的 CMD52 后，在 SDIO CMD52 Interface 产生相应的数据请求，用户需按照相应时序作出响应。

SDIO CMD52 Interface 写操作（RAW flag=1）

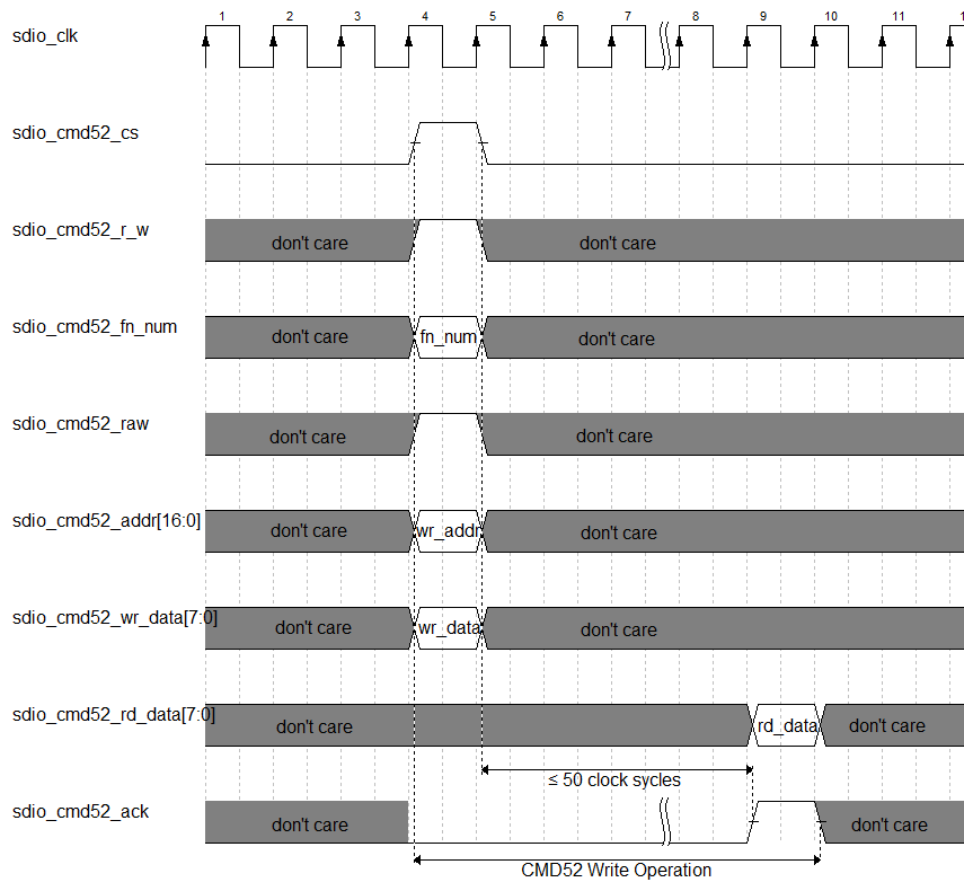
当 Host 通过 CMD52 进行写操作并且 RAW flag 为 1 时，要求用户在完成寄存器写操作后，把寄存器当前值返回。

当 IP 接收到一个正确的 CMD52 写操作且寄存器地址为 Function1 时，IP 在 SDIO CMD52 Interface 产生写数据请求。此时，IP 拉高 `sdio_cmd52_cs` 信号，表示 SDIO CMD52 Interface 有数据请求。同时 `sdio_cmd52_r_w` 信号和 `sdio_cmd52_raw` 信号拉高，表示此次操作为 RAW flag = 1 的写操作。当 `sdio_cmd52_cs = 1` 时，`sdio_cmd52_fn_num`、`sdio_cmd52_addr`、`sdio_cmd52_wr_data` 信号有效。用户完成写操作后，需把 `sdio_cmd52_ack` 拉高一个周期，通知 IP 此次写操作完成。当 IP 检测到 `sdio_cmd52_ack` 为 1 后，会自动向 Host 发送 R5。

由于此次写操作 RAW flag = 1，用户在拉高 `sdio_cmd52_ack` 的同时，需在 `sdio_cmd52_rd_data` 返回所写寄存器当前值，此值将会在 R5 的 Read or Write Data 字段返回 Host。

SDIO CMD52 Interface 写操作（RAW flag = 1）时序如图 4-9 所示。

图 4-9 SDIO CMD52 Interface 写操作 (RAW flag = 1) 时序



SDIO CMD52 Interface 写操作 (RAW flag=0)

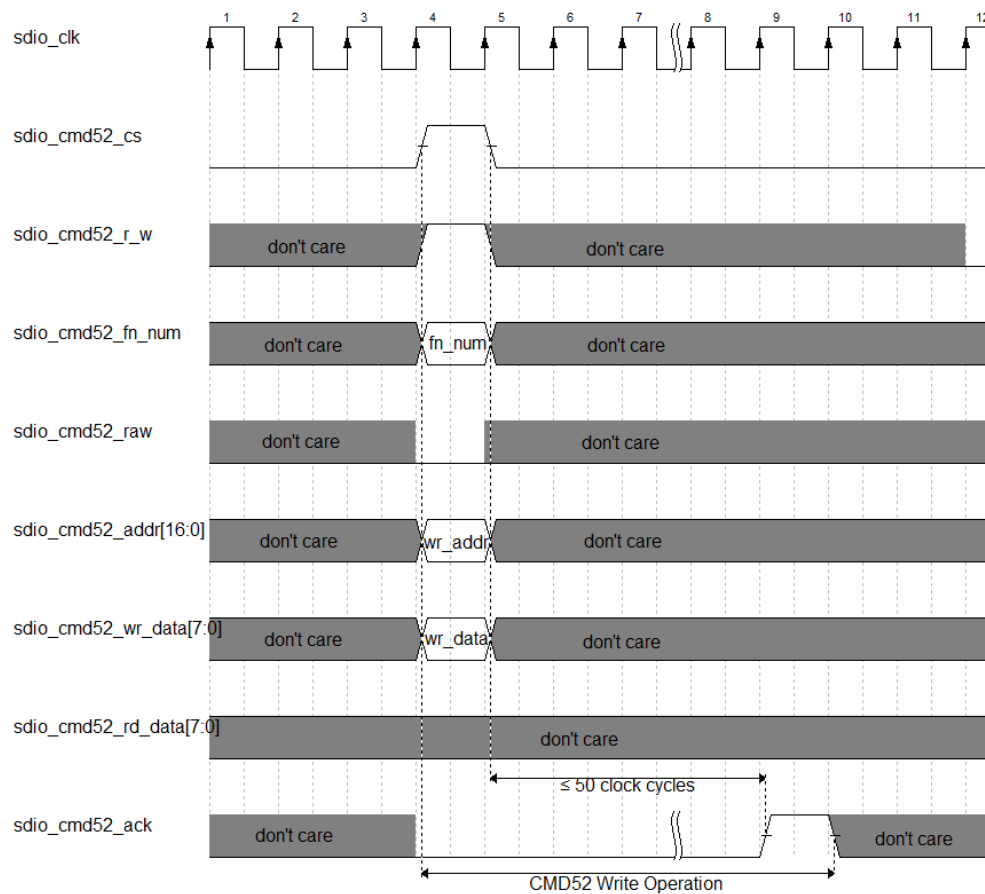
当 Host 通过 CMD52 进行写操作并且 RAW flag 为 0 时，用户在完成寄存器写操作后无需把寄存器当前值返回。

当 IP 接收到一个正确的 CMD52 写操作且寄存器地址为 Function1 时，IP 在 SDIO CMD52 Interface 产生写数据请求。此时，IP 拉高 sdio_cmd52_cs 信号，表示 SDIO CMD52 Interface 有数据请求。同时 sdio_cmd52_r_w 信号拉高，sdio_cmd52_raw 信号拉低，表示此次操作为 RAW flag = 0 的写操作。当 sdio_cmd52_cs = 1 时，sdio_cmd52_fn_num、sdio_cmd52_addr、sdio_cmd52_wr_data 信号有效。用户完成写操作后，需把 sdio_cmd52_ack 拉高一个周期，通知 IP 此次写操作完成。当 IP 检测到 sdio_cmd52_ack 为 1 后，会自动向 Host 发送 R5。

由于此次写操作 RAW flag = 0，用户在拉高 sdio_cmd52_ack 的同时，无需在 sdio_cmd52_rd_data 返回写寄存器当前值，IP 会自动将 CMD52 的 Write Data 字段值放在 R5 的 Read or Write Data 字段返回 Host。

SDIO CMD52 Interface 接口写操作 (RAW flag = 0) 时序如图 4-10 所示。

图 4-10 SDIO CMD52 Interface 接口写操作 (RAW flag = 0) 时序



SDIO CMD52 Interface 读操作

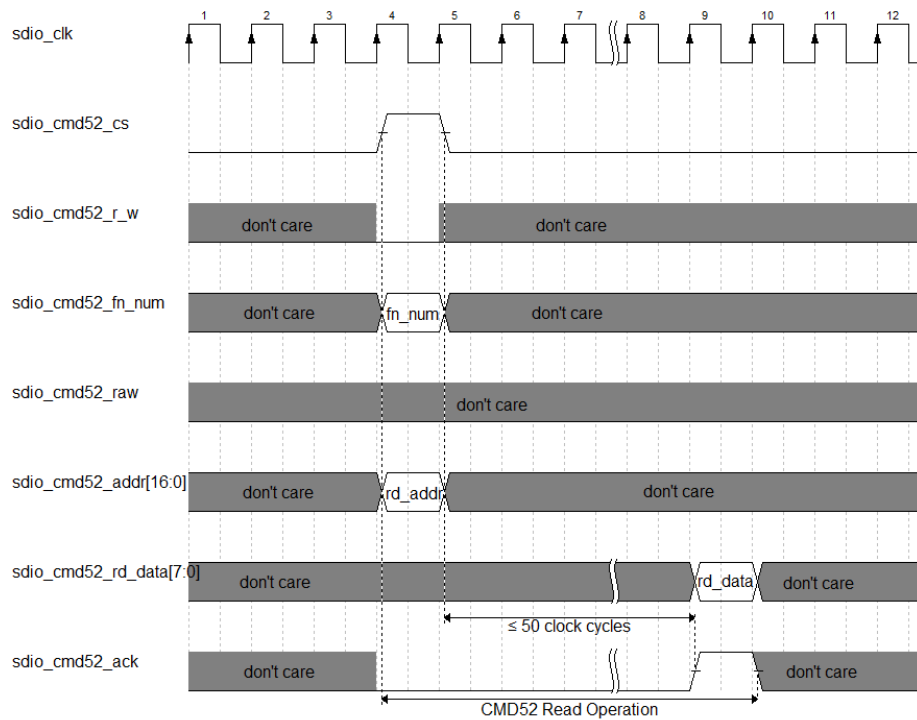
当 Host 通过 CMD52 进行读操作时,用户在收到读请求后需把寄存器的值返回。

当 IP 接收到一个正确的 CMD52 读操作且寄存器地址为 CIS 或 Function1 时, IP 在 SDIO CMD52 Interface 产生读数据请求。此时, IP 拉高 sdio_cmd52_cs 信号,表示 SDIO CMD52 Interface 有数据请求。同时 sdio_cmd52_r_w 信号拉低,表示此次操作为读操作。当 sdio_cmd52_cs = 1 时, sdio_cmd52_fn_num、sdio_cmd52_addr 信号有效。用户完成读操作后,需把 sdio_cmd52_ack 拉高一个周期,通知 IP 此次读操作完成。当 IP 检测到 sdio_cmd52_ack 为 1 后,会自动向 Host 发送 R5。

由于此次为读操作,用户在拉高 sdio_cmd52_ack 的同时,需在 sdio_cmd52_rd_data 返回读寄存器的值,此值将会在 R5 的 Read or Write Data 字段返回 Host。

SDIO CMD52 Interface 接口读操作时序如图 4-11 所示。

图 4-11 SDIO CMD52 Interface 接口读操作时序



CMD52 超时

SDIO 标准规定 Command 的最后一位数据到 Response 的第一位数据延时 NCR 范围为 2~64 clock cycles。因此用户在 `sdio_cmd52_cs` 有效到 `sdio_cmd52_ack` 有效之间的时钟间隔不要超过 50 clock cycles（由于 IP 内部有采样延时）。在 `sdio_cmd52_cs` 有效后的 50 clock cycles 内，用户无法返回 `sdio_cmd52_ack` 时，请放弃此次 CMD52 操作，`sdio_cmd52_ack` 无需再拉高。此时 Host 会认为此次操作超时。

4.11.3 SDIO CMD53 Interface

当 Host 通过 CMD53 读写 CIS 或 Function1 寄存器时，用户需要通过 SDIO CMD53 Interface 控制数据的收发。IP 收到读写 CIS 或 Function1 寄存器的 CMD53 后，在 SDIO CMD53 Interface 产生相应的数据请求，用户需按照相应时序作出响应。

SDIO CMD53 命令分为 Bytes Transfer 模式和 Block Transfer 模式。当 CMD53 为 Bytes Transfer 模式时，SDIO CMD53 Interface 产生一次读写操作，读写长度为 CMD53 Byte/Block Count 字段；当 CMD53 为 Block Transfer 模式时，SDIO CMD53 Interface 产生 N 次读写操作，N 为 CMD53 Byte/Block Count 字段，读写长度为相应 Function Block Size。因此用户无法感知此次操作是 Bytes Transfer 还是 Block Transfer，只需按照相应时序收发数据即可。

SDIO SDR104 CMD53 Interface 写操作

每一次写操作请求代表 Bytes Transfer 写操作或 Block Transfer 中的单次 Block 写操作。

当 IP 接收到一个正确的 CMD53 写操作且地址为 Function1 时, IP 自动回应 R5。之后在 SDIO CMD53 Interface 产生写数据请求。IP 拉高 `sdio_cmd53_wr_en` 信号, 表示 SDIO CMD53 Interface 有写数据请求。当 `sdio_cmd53_wr_en = 1` 时, `sdio_cmd53_fn_num`、`sdio_cmd53_addr`、`sdio_cmd53_len`、`sdio_cmd53_op_code` 信号有效。

当 Host 开始通过 `sdio_dat` 向 IP 写数据时, 用户可以通过 `sdio_cmd53_wr_valid` 和 `sdio_cmd53_wr_data` 接收写数据。一次写数据的字节数等于 `sdio_cmd53_len` 的值。IP 自动校验 CRC16 和 End bit 的正确性。

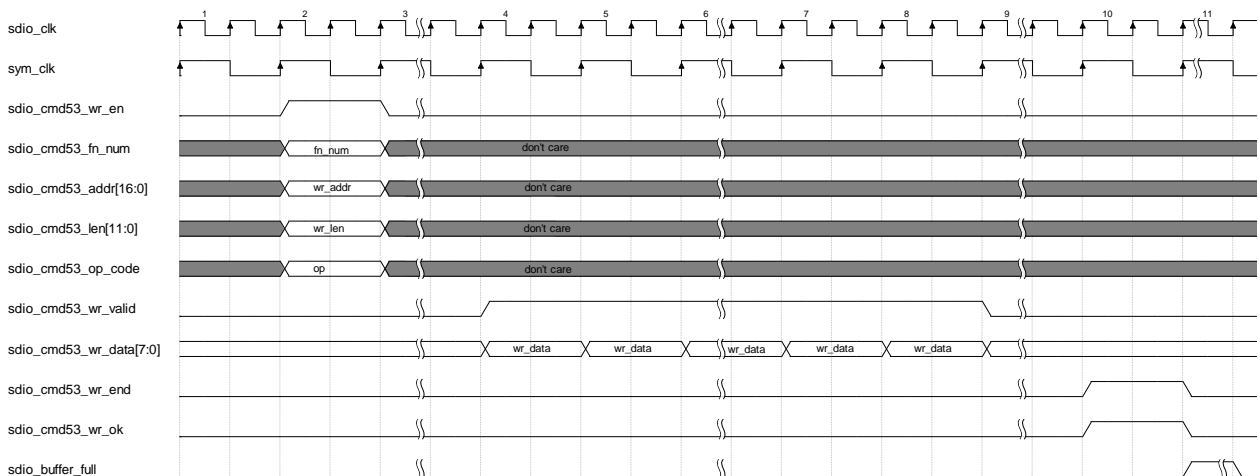
当 Host 完成一次 Bytes Transfer 写操作或 Block Transfer 中的单次 Block 写操作, 且 IP 已经向 Host 发送完成 CRC status 和 End bit 时, `sdio_cmd53_wr_end` 拉高, 用来通知用户此次写操作结束。同时 `sdio_cmd53_wr_ok` 用来指示此次写数据是否正确。`sdio_cmd53_wr_ok` 同步于 `sdio_cmd53_wr_end`。`sdio_cmd53_wr_ok` 为 1 时, 说明此次写操作正确, 用户需要处理此次写操作的数据; `sdio_cmd53_wr_ok` 为 0 时, 说明此次写操作有 CRC16 或 End bit 或 CRC status 发送错误, 用户需要丢弃此次写操作的数据。

在 Block Transfer 中, 若 IP 收到某个 block 写操作有错误, IP 将放弃之后的数据接收。

当用户检测到 `sdio_cmd53_wr_end = 1` 后, 需要通过 `sdio_buffer_full` 通知 IP 是否有空闲的 buffer 继续接收之后的数据。若用户的 buffer 已满, 无法继续接收数据, 需要在检测到 `sdio_cmd53_wr_end = 1` 后立即把 `sdio_buffer_full` 拉高。当用户把 `sdio_buffer_full` 拉高后, IP 将把 `sdio_dat0` 拉低, 处在 card busy 状态, 直到用户把 `sdio_buffer_full` 拉低。Host 检测到 `sdio_dat0` 为 0 时, 将不会继续下一次写操作, 直到检测到 `sdio_dat0` 被拉高为止。

SDIO CMD53 Interface 接口 4-bit 模式写操作时序如图 4-12 所示。

图 4-12 SDIO CMD53 Interface 接口 4-bit 模式写操作时序



SDIO SDR104 CMD53 Interface 读操作

每一次读操作请求代表 Bytes Transfer 读操作或 Block Transfer 中的单次 Block 读操作。

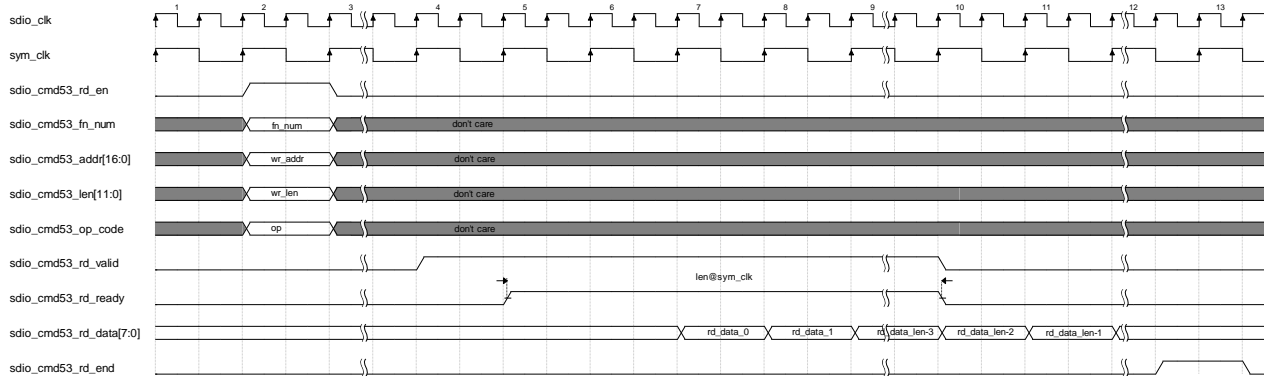
当 IP 接收到一个正确的 CMD53 读操作且地址为 CIS 或 Function1 时，IP 自动回应 R5。之后在 SDIO CMD53 Interface 产生读数据请求。IP 拉高 `sdio_cmd53_rd_en` 信号，表示 SDIO CMD53 Interface 有读数据请求。当 `sdio_cmd53_rd_en = 1` 时，`sdio_cmd53_fn_num`、`sdio_cmd53_addr`、`sdio_cmd53_len`、`sdio_cmd53_op_code` 信号有效。

当用户收到 CMD53 读请求后，用户可以通过 `sdio_cmd53_rd_valid`、`sdio_cmd53_rd_data` 和 `sdio_cmd53_rd_ready` 发送读数据。`sdio_cmd53_rd_ready` 信号每拉高一次，表明 IP 准备好接收一个读数据，为了时序优化，用户需要在两拍之后把新的数据再次在 `sdio_cmd53_rd_data` 信号准备好。一次读数据的字节数等于 `sdio_cmd53_len` 的值，用户需要根据 `sdio_cmd53_len` 的值来控制 `sdio_cmd53_rd_valid` 的长度，在发送完足够的字节后主动拉低 `sdio_cmd53_rd_valid`。IP 根据 Host 请求数据的长度向 `sdio_dat` 发送数据。发送数据完成后，IP 自动发送 CRC16 字段和 End bit，之后 IP 拉高 `sdio_cmd53_rd_end` 信号一个周期，通知用户此次读操作结束。

IP 不支持 Read Wait 功能。

SDIO CMD53 Interface 接口 4-bit 模式读操作时序如图 4-13 所示。

图 4-13 SDIO SDR104 CMD53 Interface 接口 4-bit 模式读操作



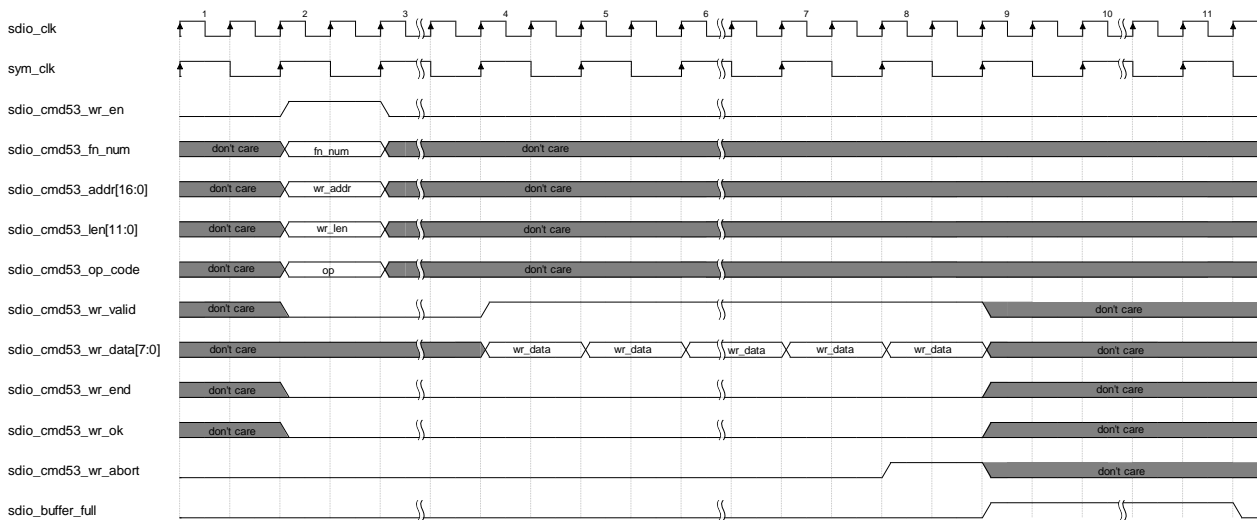
Abort 操作

当 Host 通过 CMD53 以 Infinite Block Mode 读写数据时，可以通过 CMD52 发送 Abort 命令来中断读写过程。

当 IP 收到 CMD52Abort 且正在进行写数据操作时，IP 将把 `sdio_cmd53_wr_abort` 拉高一个时钟周期，来通知用户结束此次写操作。

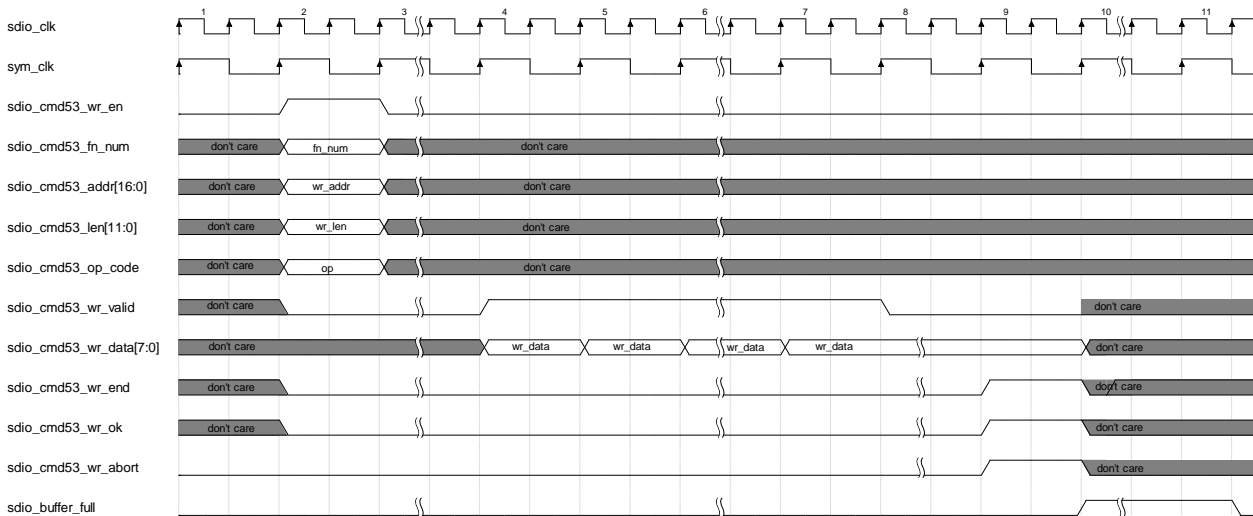
如果用户检测到 `sdio_cmd53_wr_abort = 1`，且在此之前在没有检测到 `sdio_cmd53_wr_end = 1`，说明此次 abort 触发一次 Bad Abort，即 IP 在完成 CRC status 和 End bit 发送之前收到 abort。此时 IP 不会再产生 `sdio_cmd53_wr_end = 1`。用户需要结束此次写操作且丢弃此次写操作的数据，如图 4-14 所示。

图 4-14 Write Bad Abort



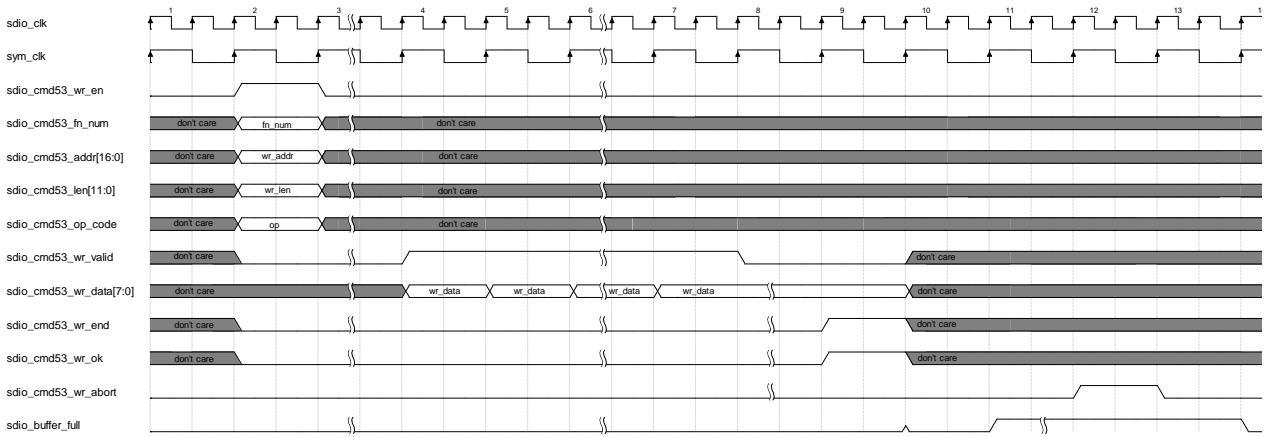
如果用户同时接收到 `sdio_cmd53_wr_abort = 1` 和 `sdio_cmd53_wr_end = 1`，说明此次 abort 触发一次 Good Abort，即 IP 在发送 CRC status 和 End bit 结束时收到 abort。用户需要接收此次写操作的数据，如图 4-15 所示。

图 4-15 Write Good Abort Case1



如果用户接收到 `sdio_cmd53_wr_abort = 1` 在 `sdio_cmd53_wr_end = 1` 之后，说明此次 abort 触发于 card busy 状态，即 IP 在完成 CRC status 和 End bit 发送后收到 abort，为 Good Abort。用户需要接收此次写操作的数据，如图 4-16 所示。

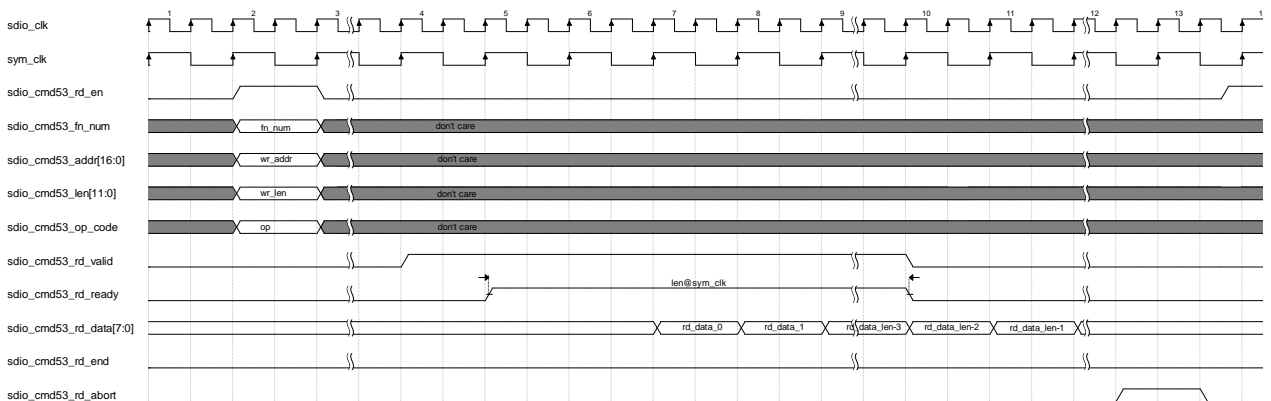
图 4-16 Write Good Abort Case2



当 IP 收到 CMD52 Abort 且正在进行读数据操作时，IP 将把 sdio_cmd53_rd_abort 拉高一个时钟周期，来通知用户结束此次读操作。

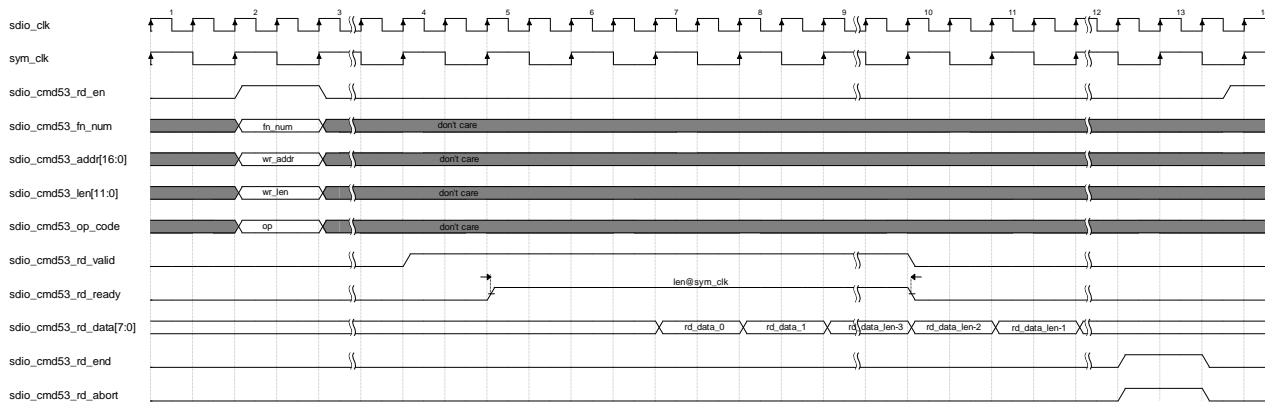
如果用户检测到 sdio_cmd53_rd_abort = 1，且在此之前在没有检测到 sdio_cmd53_rd_end = 1，说明此次 abort 触发一次 Bad Abort，即 IP 在完成 CRC16 和 End bit 发送之前收到 abort。此时 IP 不会再产生 sdio_cmd53_rd_end = 1。用户需要结束此次读操作，且此次读操作不成功，如图 4-17 所示。

图 4-17 Read Bad Abort



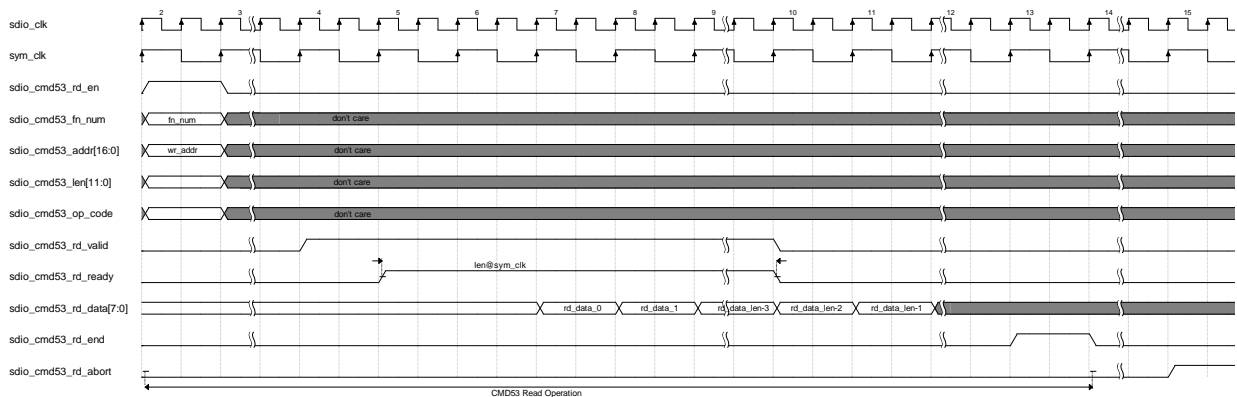
如果用户同时接收到 sdio_cmd53_rd_abort = 1 和 sdio_cmd53_rd_end = 1，说明此次 abort 触发一次 Good Abort，即 IP 在发送 CRC16 和 End bit 结束时收到 abort，此次读操作成功，如图 4-18 所示。

图 4-18 Read Good Abort Case1



如果用户接收到 `sdio_cmd53_rd_abort = 1` 在 `sdio_cmd53_rd_end = 1` 之后,说明此次 abort 触发于两次读操作之间,即 IP 在完成 CRC16 和 End bit 发送后收到 abort, 为 Good Abort, 此次读操作成功, 如图 4-19 所示。

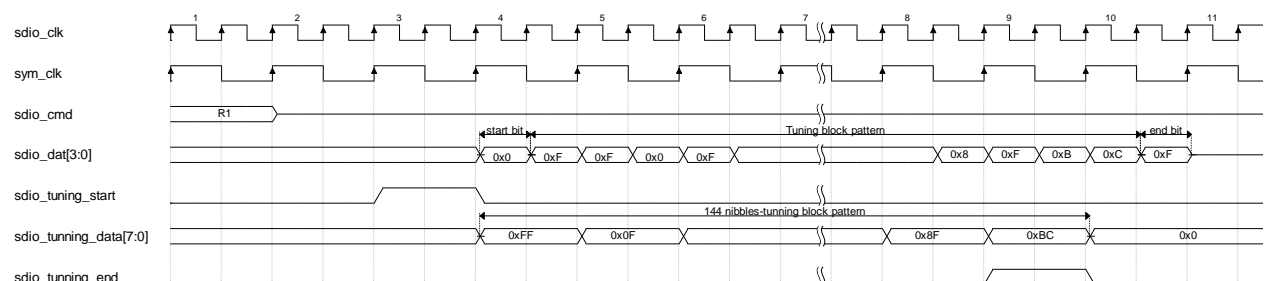
图 4-19 Read Good Abort Case2



4.11.4 SDIO Tuning Interface

当 IP 被配置为 UHS-I Card 时,支持 CMD19。IP 接收到正确的 CMD19 后,会自动回应 R1。在发送完 R1 之后,需要用户通过 SDIO Tuning Interface 发送 Tuning Block Pattern。Tuning Block Pattern 参见图 4-5。SDIO Tuning Interface 时序如图 4-20 所示。

图 4-20 SDIO Tuning Interface 时序



4.12 IP 初始化流程

1. 芯片上电。
2. rstn 拉低开始全局复位。
3. rstn 拉高完成全局复位。
4. 用户通过 CPU Slave Interface 配置 IP 相关参数，以下寄存器可参考：
 - 配置 Reg0x04 bit22: IP 是否为低速卡
 - 配置 Reg0x00 bit24: IP 是否支持 High Speed
 - 配置 Reg0x08 bit23~0: 配置 CCCR CIS Pointer
 - 配置 Reg0x2C bit23~0: 配置 Function1 CIS Pointer
 - 配置 Reg0x34 bit15~0: 配置 CCCR Max Block Size
 - 配置 Reg0x34 bit31~16: 配置 Function1 Max Block Size
 - 配置 Reg0x30 bit0: 配置 IO Ready = 1
5. 完成以上配置后，当 IP 接收 CMD5 时，会在 R4 C 字段回应 1，以通知 SDIOHost IO Ready。
6. 当 SDIO Host 通过 CMD7 选中 IP 后，IP 进入 Command State，Host 可以通过 CMD52 和 CMD53 读写 IP CIA 寄存器。
7. SDIO Host 通过 CMD52 配置 CCCR IOE1 = 1，使能 Function1，此时 IP 把 fun1_ioe 拉高。
8. 用户开始初始化 Function1。
9. Function1 初始化完成后，用户拉高 fun1_ior，即设置 CCCR IOR1 = 1。
10. SDIO Host 通过 CMD52 读取到 CCCR IOR1 为 1 后，开始访问 Function1 寄存器。
11. 如果 Host 要进行 block 模式读或写操作，需要先通过 CMD52 命令配置相应 Function 的 block size 寄存器，且配置的值须小于等于上述第 4 点中的对应 Max Block Size。Function0 Block Size 寄存器为 CCCR 寄存器 10h-11h；Function1 Block Size 寄存器为 FBR1 寄存器 110h-111h。这两个寄存器均为小端模式，且配置范围为 1~2048。
12. 如果 Host 需要接收 IP 的中断信号，Host 需要通过 CMD52 打开 IP 的中断使能寄存器 CCCR 04h。
13. 以上 SDIO 寄存器的具体说明请参考 SDIO 协议。

5 端口列表

Gowin SDIO SDR104 Slave Controller IP 的 IO 端口如表 5-1 所示。

表 5-1 Gowin SDIO SDR104 Slave Controller IP IO 端口

信号	方向	位宽	描述
SDIO Interface			
sdio_clk	input	1	SDIO 时钟输入信号
sym_clk	output	1	SDIO 输出的用户时钟
sdio_cmd_in	input	1	SDIO Command/Response 信号输入
sdio_cmd_out	output	1	SDIO Command/Response 信号输出
sdio_cmd_oen	output	1	SDIO Command/Response 信号输出使能, 低有效。
sdio_dat0_in	input	1	SDIO 1-bit 模式为 dat 信号输入, SDIO 4-bit 模式为 dat0 信号输入。
sdio_dat0_out	output	1	SDIO 1-bit 模式为 dat 信号输出, SDIO 4-bit 模式为 dat0 信号输出。
sdio_dat0_oen	output	1	SDIO 1-bit 模式为 dat 信号输出使能, SDIO 4-bit 模式为 dat0 信号输出使能, 低有效。
sdio_dat1_in	input	1	SDIO 1-bit 模式为中断信号输入, SDIO 4-bit 模式为 dat1 和中断复用信号输入。
sdio_dat1_out	output	1	SDIO 1-bit 模式为中断信号输出, SDIO 4-bit 模式为 dat1 和中断复用信号输出。
sdio_dat1_oen	output	1	SDIO 1-bit 模式为中断信号输出使能, SDIO 4-bit 模式为 dat1 和中断复用信号输出使能, 低有效。
sdio_dat2_in	input	1	SDIO 1-bit 模式为 Read Wait 信号输入, SDIO 4-bit 模式为 dat2 和 Read Wait 复用信号输入。
sdio_dat2_out	output	1	SDIO 1-bit 模式为 Read Wait 信号输出, SDIO 4-bit 模式为 dat2 和 Read Wait 复用信号输出。
sdio_dat2_oen	output	1	SDIO 1-bit 模式为 Read Wait 信号输出使能, SDIO 4-bit 模式为 dat2 和 Read Wait 复用信号输出使能, 低有效。
sdio_dat3_in	input	1	SDIO 1-bit 模式未定义, SDIO 4-bit 模式为 dat3

信号	方向	位宽	描述
			信号输入。
sdio_dat3_out	output	1	SDIO 1-bit 模式未定义，SDIO 4-bit 模式为 dat3 信号输出。
sdio_dat3_oen	output	1	SDIO 1-bit 模式未定义，SDIO 4-bit 模式为 dat3 信号输出使能，低有效。
Reset			
rstn	input	1	全局异步复位，低有效。
cmd52_rst	output	1	Host 通过 cmd52 复位 SDIO 时，此信号为 1。同步于 sdio_clk。
Function1 Enable/Ready			
fun1_ioe	output	1	Function1 使能信号，反映的是 CCCRIOE1 的值。同步于 sdio_clk。
fun1_ior	input	1	Function1 准备完成标志。对应 CCCR IOR1 Reg0x3 bit1。
Function1 Interrupt			
fun1_interrupt	input	1	0: Function1 无中断 1: Function1 有中断
CPU Slave Interface (同步于 cpu_clk)			
cpu_clk	input	1	cpu 接口时钟输入
cpu_rst	input	1	cpu 接口同步复位输入信号，同步于 cpu_clk，1 有效。
slv_cpu_cs	input	1	cpu 接口片选信号，高有效。
slv_cpu_op	input	1	cpu 接口操作模式 0: 读操作；1: 写操作。
slv_cpu_addr	input	8	cpu 接口操作地址
slv_cpu_wr_data	input	32	cpu 接口写数据
slv_cpu_byte_en	input	4	cpu 接口写操作字节使能 1: 使能；0: 禁止。 bit 3 对应 slv_cpu_wr_data[31:24]; bit 2 对应 slv_cpu_wr_data[23:16]; bit 1 对应 slv_cpu_wr_data[15:8]; bit 0 对应 slv_cpu_wr_data[7:0]
slv_cpu_rd_data	output	32	cpu 接口读数据
slv_cpu_ack	output	1	cpu 接口操作响应，此位为 1 时，说明此次操作结束。
slv_cpu_err	output	1	cpu 接口操作错误指示。当 slv_cpu_ack 为 1 时有效。 1: 操作错误；0: 操作正确。
SDIO CMD52 Interface (同步于 sdio_clk)			
sdio_cmd52_cs	output	1	SDIO CMD52 命令使能。

信号	方向	位宽	描述
			0: 禁止; 1: 使能
sdio_cmd52_r_w	output	1	SDIO CMD52 命令读写操作, 对应 IO_RW_DIRECT Command (CMD52)命令 R/W flag 字段。 0: 读操作; 1: 写操作
sdio_cmd52_fn_num	output	1	SDIO CMD52 命令 Function Number, 对应 IO_RW_DIRECT Command (CMD52)命令 Function Number 字段。 0: Function0; 1: Function1
sdio_cmd52_raw	output	1	SDIO CMD52 命令 RAW 使能, 对应 IO_RW_DIRECT Command (CMD52)命令 RAW flag 字段。 0: 禁止 RAW; 1: 使能 RAW
sdio_cmd52_addr	output	17	SDIO CMD52 命令操作地址, 对应 IO_RW_DIRECT Command (CMD52)命令 Register Address 字段。
sdio_cmd52_wr_data	output	8	SDIO CMD52 命令写数据, 对应 IO_RW_DIRECT Command (CMD52)命令 Write Data 字段。当 SDIO CMD52 接口在写操作模式下有效。
sdio_cmd52_rd_data	input	8	SDIO CMD52 命令读数据或 RAW 数据, 对应 IO_RW_DIRECT Response (R5)命令 Read Data 字段。 当 SDIO CMD52 接口在读操作模式下, 用户需返回所读寄存器的值; 当 SDIO CMD52 接口在 RAW 操作模式下, 用户需返回所写寄存器的值。
sdio_cmd52_ack	input	1	SDIO CMD52 命令 ack 信号。当用户把 ack 信号拉高单周期后, IP 认为用户对于此次 CMD52 操作结束, 并开始发送 R5。 当此次操作为读操作时, 用户在拉高 ack 的同时, 需要把读地址寄存器的值在 sdio_cmd52_rd_data 处返回给 IP; 当此次操作为 RAW 操作时, 用户在拉高 ack 的同时, 需要把写地址寄存器当前值在 sdio_cmd52_rd_data 处返回给 IP。
SDIO CMD53 Interface (同步于 sdio_clk)			
sdio_cmd53_wr_en	output	1	SDIO CMD53 写使能。 0: 禁止; 1: 使能
sdio_cmd53_rd_en	output	1	SDIO CMD53 读使能。 0: 禁止; 1: 使能
sdio_cmd53_fn_num	output	1	SDIO CMD53 命令 Function Number, 对应 IO_RW_EXTENDED Command (CMD53)命令 Function Number 字段。 0: Function0; 1: Function1

信号	方向	位宽	描述
sdio_cmd53_addr	output	17	SDIO CMD53 命令操作首地址，对应 IO_RW_EXTENDED Command (CMD53)命令 Register Address 字段。
sdio_cmd53_len	output	12	SDIO CMD53 命令操作长度，表示此次操作用户需要读写的字节数。
sdio_cmd53_op_code	output	1	SDIO CMD53 命令 OP Code，对应 IO_RW_EXTENDED Command (CMD53)命令 OP Code 字段。 0: Multi byte R/W to fixed address; 1: Multi byte R/W to incrementing address。
sdio_cmd53_wr_valid	output	1	SDIO CMD53 写数据有效。此位为 1 时，说明 sdio_cmd53_wr_data 有效。
sdio_cmd53_wr_data	output	8	SDIO CMD53 写数据。当 sdio_cmd53_wr_valid 为 1 时，说明此写数据有效。
sdio_cmd53_wr_end	output	1	SDIO CMD53 写数据结束标志。此位为 1 时，说明此次写操作结束，并且 IP 已经向 Host 发送完成 CRC status 和 End bit。
sdio_cmd53_wr_ok	output	1	SDIO CMD53 写数据正确。此位同步于 sdio_cmd53_wr_end。 1: 此次写操作正确 0: 此次写操作有 CRC16 或 End bit 或 CRC status 发送错误
sdio_cmd53_wr_abort	output	1	写操作 abort 标志。Host 通过 CMD52 命令 abort 此次写操作时，此位为 1。当用户检测到此位为 1 时，意味着此次写操作被终止。
sdio_cmd53_rd_valid	input	1	SDIO CMD53 读数据有效。在读操作时，当用户准备好读数据后，需要将此位拉高，通知 IP 数据已准备好，之后 IP 开始在 sdio_data 发送读数据。
sdio_cmd53_rd_data	input	8	SDIO CMD53 读数据。在读操作时，用户提供给 IP 的读数据。
sdio_cmd53_rd_ready	output	1	SDIO CMD53 读数据 ready 信号。此位为 1 时，说明 IP 已经取得 sdio_cmd53_rd_data，用户需要把下一个 sdio_cmd53_rd_data 准备好。
sdio_cmd53_rd_end	output	1	SDIO CMD53 读数据结束标志。此位为 1 时，说明此次读操作结束，并且 IP 已经向 Host 发送完成 CRC 和 End bit。
sdio_cmd53_rd_abort	output	1	读操作 abort 标志。Host 通过 CMD52 命令 abort 此次读操作时，此位为 1。当用户检测到此位为 1 时，意味着此次读操作被终止。
sdio_buffer_full	input	1	用户写 buffer full 标志。 1: 用户写 buffer 满； 0: 用户写 buffer 不满。 当一次写操作结束时（IP 发送完成 CRC status 和 End bit），如果用户 buffer 为满，无法接收下一个

信号	方向	位宽	描述
			数据，需要把此位拉高。此时 IP 会把 <code>sdio_dat0</code> 拉低，来通知 Host card busy。当用户把此位拉低后，IP 把 <code>sdio_dat0</code> 拉高，之后 Host 才有可能进行下一次写操作。
SDIO Tuning Interface			
<code>sdio_tuning_start</code>	output	1	SDIO Tuning Block Pattern 开始信号。若 IP 不需要发送 Tuning Block Pattern，此管脚可悬空。
<code>sdio_tuning_data</code>	input	4	SDIO Tuning Block Pattern 数据信号。若 IP 不需要发送 Tuning Block Pattern，此管脚可输入常 0。
<code>sdio_tuning_end</code>	input	1	SDIO Tuning Block Pattern 结束信号。若 IP 不需要发送 Tuning Block Pattern，此管脚可输入常 0。
2MHz Clock			
<code>clk_2mhz</code>	input	1	2MHz 时钟输入。当 IP 配置为 UHS-I Card，且需要 CMD11 命令执行切换电压时序时，需要在此接口输入 2MHz 时钟，用来检测 <code>sdio_clk</code> 是否停止。若 IP 配置为 Non UHS Card 或不需要 CMD11 命令时，此管脚可输入常 0。

6 参数配置

用户需根据设计要求配置 Gowin SDIO SDR104 Slave Controller 的各个静态参数与时序参数，如表 6-1 所示。

表 6-1 Gowin SDIO SDR104 Slave Controller 静态参数与时序参数

名称	描述	选项
Card Type	<ul style="list-style-type: none">● Non UHS Card 支持 Low-Speed 和 Full-Speed。其中 Full-Speed 支持 Default-Speed 和 High-Speed;● UHS-I Card 支持 SDR104。	Non UHS/UHS-I

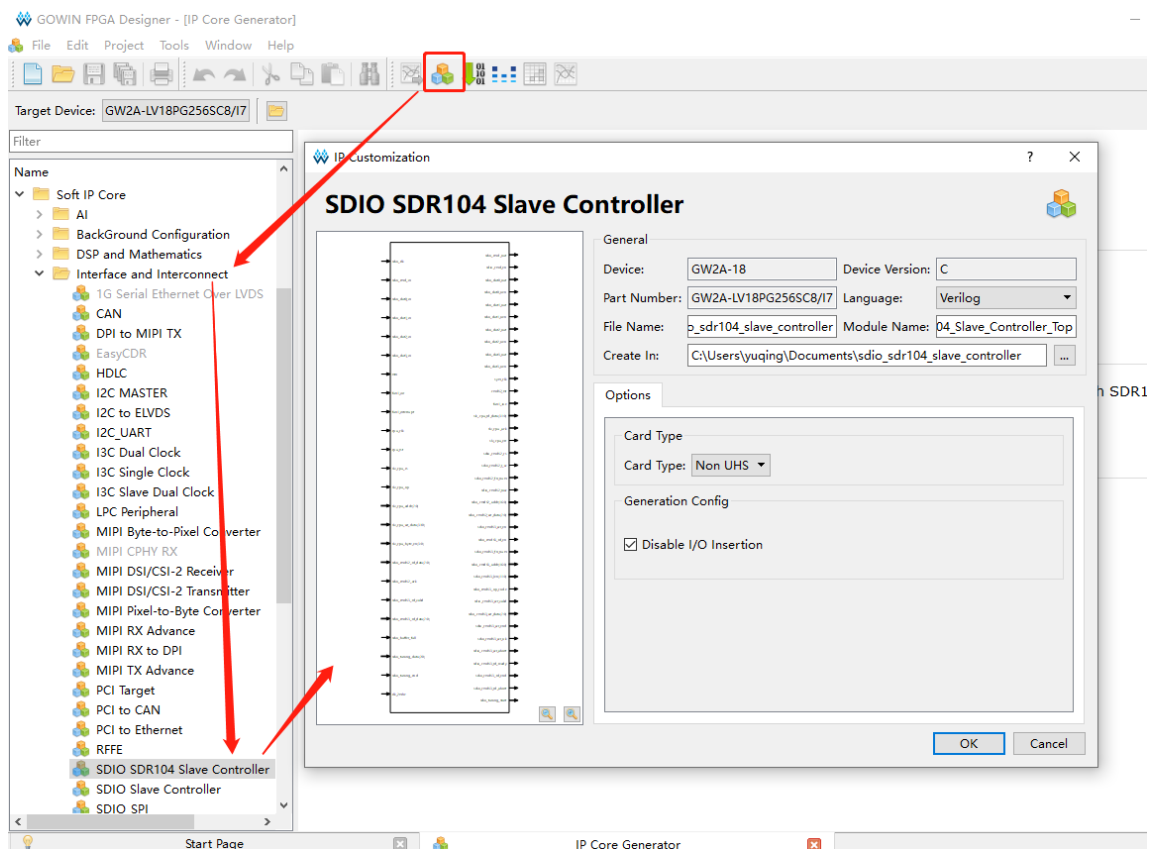
7 界面配置

用户可在高云半导体云源®软件中通过 IP Core Generator 工具调用并配置 SDIO SDR104 Slave Controller IP。本章节以选择使用 Non UHS Card 为例，介绍了主要配置界面、配置流程以及各配置选项含义。

1. 打开 IP Core Generator

用户建立工程后，点击左上角 Tools 选项卡，下拉单击“IP Core Generator”选项，就可打开 Gowin 的 IP 核产生工具，选择“SDIO SDR104 Slave Controller”，如图 7-1 所示。

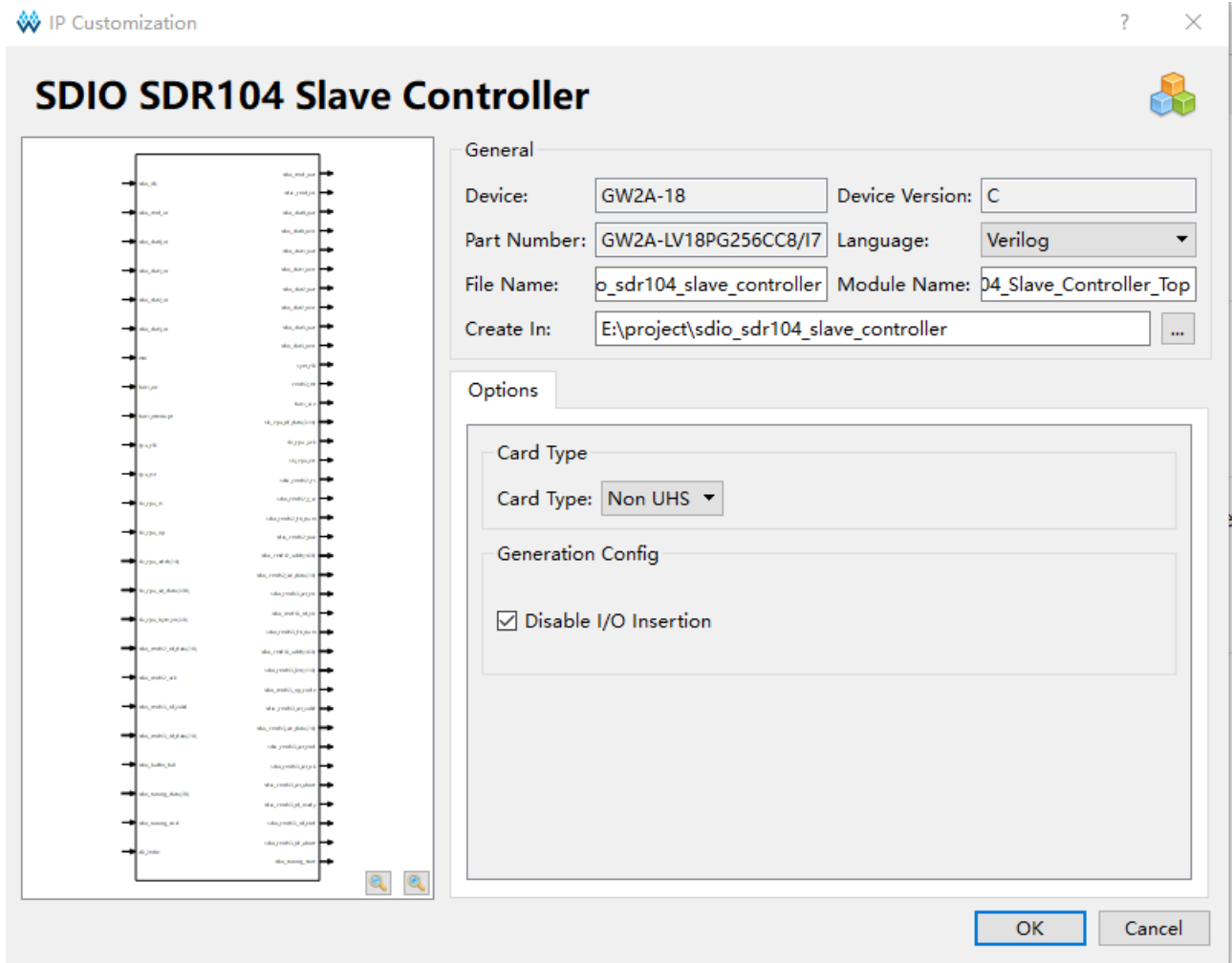
图 7-1 IP 核产生工具



2. SDIO SDR104 Slave Controller 端口界面

配置界面左端是 SDIO SDR104 Slave Controller IP 的接口示意图，右端是 IP 选项，如图 7-2 所示。

图 7-2 SDIO SDR104 Slave Controller IP 配置界面



8 参考设计

详细信息请参见高云半导体官网 [SDIO SDR104 Slave Controller](#) 参考设计。

