Gowin SDIO Slave Controller

User Guide

IPUG905-1.0E, /11/08/2019

Disclaimer

GOWINSEMI®, LittleBee®, Arora, and the GOWINSEMI logos are trademarks of GOWINSEMI and are registered in China, the U.S. Patent and Trademark Office, and other countries. All other words and logos identified as trademarks or service marks are the property of their respective holders, as described at www.gowinsemi.com. GOWINSEMI assumes no liability and provides no warranty (either expressed or implied) and is not responsible for any damage incurred to your hardware, software, data, or property resulting from usage of the materials or intellectual property except as outlined in the GOWINSEMI Terms and Conditions of Sale. All information in this document should be treated as preliminary. GOWINSEMI may make changes to this document at any time without prior notice. Anyone relying on this documentation should contact GOWINSEMI for the current documentation and errata.

Revision History

Date	Version	Description
11/08/2019	1.0E	Initial version published.

Contents

List of Figures

List of Tables

1 About This Guide

1.1 Purpose

This Gowin SDIO Slave Controller IP user guide aims to help users quickly learn the functions of Gowin SDIO Slave Controller and quickly understand the features and usage of the Gowin SDIO Slave Controller IP.

1.2 Supported Products

The information presented in this guide applies to the following products:

- GW1N series of FPGA Products
- GW1NR series of FPGA Products
- GW2A series of FPGA Products
- GW2AR series of FPGA Products

1.3 Related Documents

The user guides are available on the GOWINSEMI Website. You can find the related documents at www.gowinsemi.com:

1. DS100, GW1N series of FPGA Products Data Sheet
2. DS117, GW1NR series of FPGA Products Data Sheet
3. DS102, GW2A series of FPGA Products Data Sheet
4. DS226, GW2AR series of FPGA Products Data Sheet
5. SUG100, Gowin YunYuan Software User Guide

1.4 Terminology and Abbreviations

The terminology and abbreviations used in this manual are as shown in Table 1-1.

Table 1-1 Terminology and Abbreviations

Terminology and Abbreviations	Meaning
IP	Intellectual Property
LUT	Look-up Table
SDIO	Secure Digital Input and Output

1.5 Support and Feedback

Gowin Semiconductor provides customers with comprehensive technical support. If you have any questions, comments, or suggestions, please feel free to contact us directly by the following ways.

Website: www.gowinsemi.com

E-mail: support@gowinsemi.com

+Tel: +86 755 8262 0391

2 Overview

This document describes the Gowin SDIO Slave Controller IP and provides the user with a general interface of SDIO Slave. Users can integrate this IP into SDIO designs to implement the functions of an SDIO device. It can be integrated into devices that require SDIO Slave and this interface can be applied to a wide range of applications.

Table 2-1 Gowin SDIO Slave Controller IP

Gowin SDIO Slave Controller IP	
Supported Devices	GW1N-4, GW1N-4B, GW1N-6, GW1N-9, GW1NR-4, GW1NR-4B, GW1NR-9, GW1NS-4, GW1NS-4C, GW1NSR-4, GW1NSER-4C, GW1NRF-4B, GW2A series, GW2AR series
Logic Resource	See Table 3-1.
Delivered Doc.	
Design Files	Verilog (encryption)
Reference Design	Verilog
TestBench	Verilog
Test and Design Flow	
Synthesis Software	Synplify_Pro
Application Software	Gowin YunYuan (V1.9.2.02Beta)

3 Features and Performance

3.1 Key Features

- Supports SD Physical Layer Specification Version 4.00;
- Supports SDIO Specification Version 4.00;
- Supports 1-bit and 4-bit SD transfer modes;
- SPI transfer mode is not supported;
- Supports Low-Speed and Full-Speed modes; The highest speed supported is 25MB/sec;
- Supports UHS-I mode and rates of SDR12, SDR25 and SDR50
- UHS-I mode supports 1.8V voltage and voltage switching is not supported
- Supports SDIO Command types: CMD0, CMD3, CMD5, CMD7, CMD11, CMD15, CMD19, CMD52, CMD53
- Supports automatic generation and validation of CRC7 and CRC16
- Supports Function 0 and Function1
- Supports configuring IP using CPU Slave interface of user side
- Supports CMD 52 and CMD53 interfaces of user side
- Supports interrupts of 1-bit mode and 4-bit mode, and the interrupt of 4-bit mode doesn’t support Data Block Gap and Asynchronous Interrupt
- Supports write abort and read abort
- Suspend/Resume functions are not supported;
- Read Wait function is not supported;
- Combo Card is not supported;
- CSA is not supported;
- DDR mode is not supported;
- Card Detection is not supported.

3.2 Working Frequency

The working frequency of Gowin SDIO Slave Controller IP depends on the clock frequency of the Gowin SDIO Host and the maximum working frequency supported by the IP on the selected chip.

3.3 Resource Utilization

Gowin SDIO Slave Controller IP adopts Verilog, which applies in GW1N series, GW1NR series, GW2A series and GW2AR series FPGA devices. Different modes have different resource consumption.

An overview of resource utilization is given in Table 3-1. For the resource utilization of other devices, please refer to the post release information.

Table 3-1 Resource Utilization

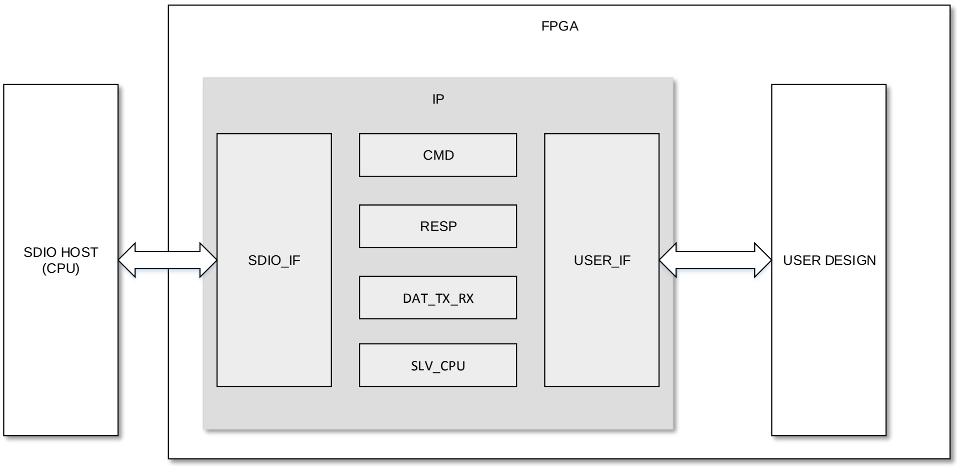
Type	LUTs	REGs	Device Series	Speed Level
Non UHS	3216	1566	GW2A18	-8
UHS-I	3214	1587	GW2A18	-8

4 Structure and Functional Description

4.1 Overall Structure

The basic structure of Gowin SDIO Slave Controller IP is shown in Figure 4-1, mainly including modules such as SDIO_IF, CMD, RESP, DAT_TX_RX, SLV_CPU and USER_IF. The User Design in the figure refers to the User Design in the FPGA, and the SDIO HOST is the SDIO host.

Figure 4-1 Structure of Gowin SDIO Slave Controller IP



- SDIO_IF module realizes data selection of SDIO interface and bi-directional transfer;
- CMD module realizes the analysis function of SDIO Command;
- RESP module realizes the generation function of SDIO Response;
- DAT_TX_RX module realizes CMD19 and CMD 53 data read and write;
- SLV_CPU module realizes user configuring IP function;
- USE_IF module realizes data exchange between user side and IP.

4.2 Definition of SDIO Interface

The corresponding relationship between SDIO Slave Controller IP interface definition and SD interface definition is shown in Table 4-1.

Table 4-1 Interface Definition Relationship between SDIO Slave Controller IP and SD

Pin	SD				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

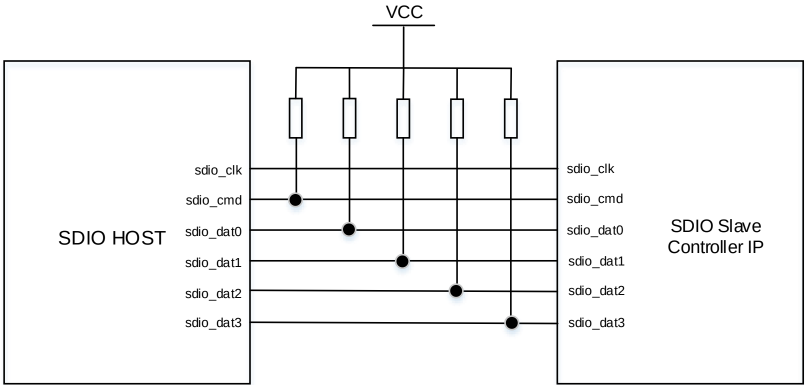
The user can convert SDIO interface defined by IP to the input/output format by the following statements.

```
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;
```

4.2.1 Circuit Connection

When the circuit is connected, the CMD and DAT wires need to be connected to the VCC through a pull-up resistor. When the IP is set to Non UHS Card, VCC is 3.3v; When the IP is set to UHS-I Card, VCC is 1.8v. The reference circuit is as shown in Figure 4-2.

Figure 4-2 Schematic Diagram of SDIO Pin Connection



4.3 Non UHS Card and UHS-I Card

IP supports Non UHS Card and UHS-I Card. Users need to configure the interface to generate IP.

4.3.1 Non UHS Card

When IP works on non-uhs Card, the voltage of Bank and the pull-up resistor on SDIO interface should be 3.3v.

Non UHS Card supports low-speed and full-speed. Full-speed supports default-speed and high-speed.

After the user configures IP to Non UHS Card, in the initialization stage, no matter whether IP receives CMD5 S18R as 0 or 1, S18A in IP response R4 is always 0. IP does not support CMD11 and CMD19 when IP works on Non UHS Card.

4.3.2 UHS-I Card

When IP works on UHS-I Card, the voltage of bank and pull-up resistor on SDIO interface should be 1.8v.

UHS-I Card supports SDR12, SDR25 and SDR50.

After the user configures IP to UHS-I Card, in the initialization stage, when IP receives CMD5 S18R as 0, S18A in IP response R4 is 0. When IP receives CMD5 S18R as 1, S18A in IP response R4 is 1. IP supports CMD11 and CMD19 when IP works on UHS-I Card. It should be noted that when IP receives CMD11, it will respond to R1 and support Voltage Switch Sequence without changing the IO Voltage.

4.4 SDIO Initialization

IP needs to be initialized by SDIO Host after power on before it can work. The initialization flow complies with SDIO standard. The initialized clock frequency is 100KHz~400KHz.

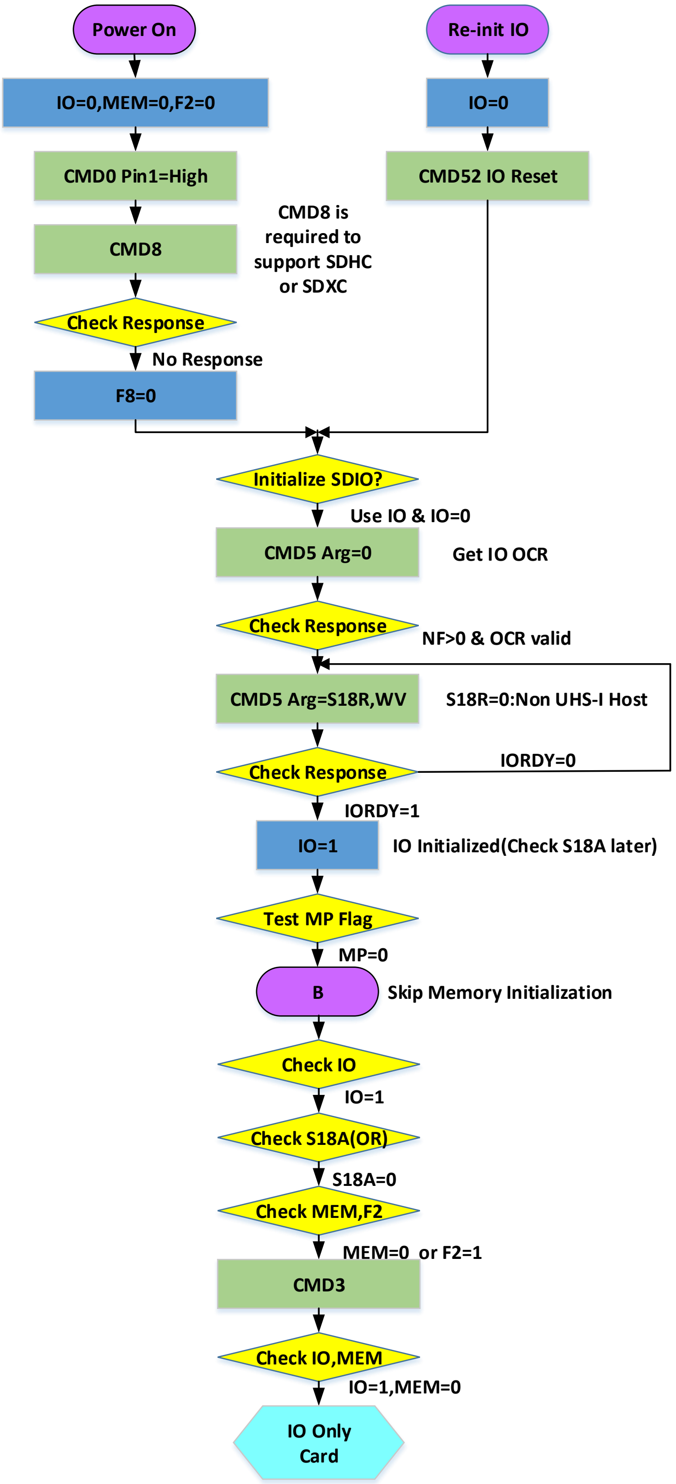
Since the IP does not support SPI mode, when CMD0 is received during initialization, the IP will remain in SD mode no matter whether sdio_dat3 is at high level or not.

When the IP is set to UHS-I Card, IP enters to the Voltage Switch Sequence through CMD11 without changing the IO Voltage. When the Voltage Switch Sequence ends, IP enters to the UHS-I state.

4.4.1 Non UHS Card Initialization Flow

Non UHS Card initialization flow is as shown in Figure 4-3.

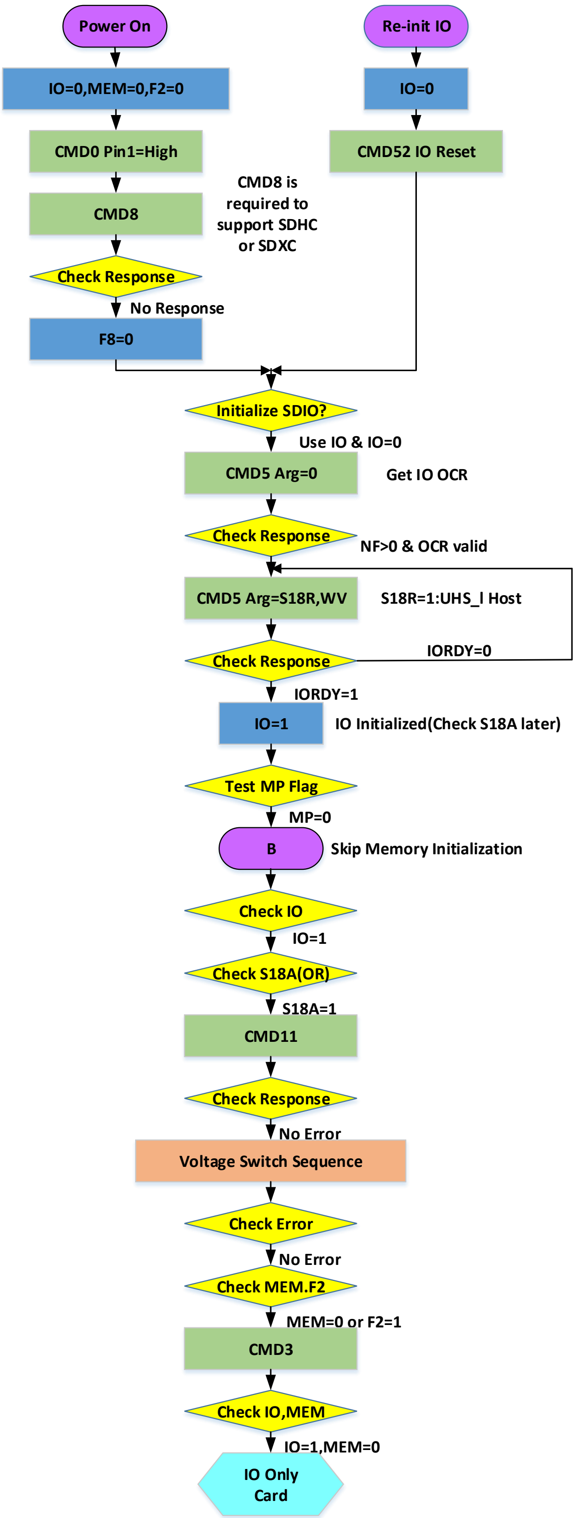
Figure 4-3 Non UHS Card Initialization Flow



4.4.2 UHS-I Card Initialization Flow

UHS-I Card initialization flow is as shown in Figure 4-4.

Figure 4-4 UHS-I Card Initialization Flow



4.5 CIA（Function0）Register

The SDIO standard defines CIA register that allows host to read slave status and configure slave function. The host can read and write CIA register through the CMD52 and CMD53 commands.

The CIA register includes CCCR register, FBR register, CIS register, and CSA register.

CCCR register and FBR (Function1) register have been implemented in IP. Users can access the corresponding register through the CPU Slave Interface in the initialization stage. When the host reads and writes the register in this part, the user does not need to participate.

The FBR (Function2~7) register does not support access temporarily.

CIS register is not implemented in IP. When host accesses register in this part, the user is required to provide data on SDIO CMD52 interface or SDIO CMD53 interface.

The CSA register does not support access temporarily.

4.6 CMD52

When the IP enters into Command State, the host can access the IP register as a single byte through CMD52. Since IP supports Function 0 and Function1, the IP only responds to host for CMD52 of Function0 and Function1.

When host accesses the CCCR and FBR registers through CMD52, IP responds automatically. The user does not to participate.

When host accesses CIS and Function1 registers through CMD52, the user needs to send and receive data through the SDIO CMD52 interface.

4.7 CMD53

When the IP enters into Command State, the host can access the IP register continuously through the CMD53 command. Since IP supports Function 0 and Function1, the IP only responds to host for CMD53 of Function 0 and Function1.

When host accesses the CCCR and FBR registers through CMD53, IP responds automatically. The user does not need to participate.

When host accesses CIS and Function1 registers through CMD53, the user needs to send and receive data through SDIO CMD53 interface.

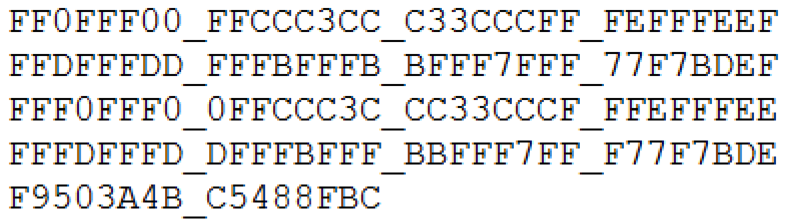
4.8 Abort

When host sends CMD53 to read and write data in Infinite Block Mode, the read and write can be interrupted by Abort sent by CMD52.

4.9 CMD19

CMD19 is supported when IP is configured to UHS-I Card. IP receives the correct CMD19 and responds to R1 automatically. Users need to provide Tuning Block Pattern through the SDIO Tuning interface. When the IP is configured to Non UHS Card, the user does not need to pay attention to SDIO Tuning interface. There are 144 nibbles in Tuning Block Pattern, as shown in Figure 4-5.

Figure 4-5 Tuning Block Pattern



```
FF0FFF00_FFCCC3CC_C33CCCFF_FEFFFEEF
FFDFFFDD_FFFBFFFB_BFFF7FFF_77F7BDEF
FFF0FFF0_0FFCCC3C_CC33CCCF_FFEFFFEE
FFFDFFFD_DFFFBFFF_BBFFF7FF_F77F7BDE
F9503A4B_C5488FBC
```

The user needs to send in order from left to right and from top to bottom.

4.10 Interrupt Function

IP supports interrupts in 1-bit and 4-bit modes.

4.10.1 Interrupt Generation

In 1-bit mode, DAT1 serves as the interrupt signal line. When the user pulls up the fun1_interrupt pin to 1, DAT1 is pulled down by IP to generate interrupt. The interrupt signal in 1-bit mode is independent of sdio_clk.

In 4-bit mode, DAT1 serves as the data and interrupt multiplexing signal line. When the user pulls up fun1_interrupt pin to 1, DAT1 is pulled down by IP in Interrupt Period to generate interrupt. 4-bit mode interrupts do not support Data Block Gap and Asynchronous Interrupt. The interrupt signal in 4-bit mode is synchronized with sdio_clk. Therefore, when sdio_clk stops, IP cannot generate an interrupt signal in DAT1. When IP works in 4-bit mode, if the clock needs to stop in SDIO Host design and interrupts are needed to call up the clock, it can be achieved in the following ways:

1. Host sets IP to 1-bit mode through CMD52;
2. Host stops the clock;
3. The user pulls up fun1_interrupt pin to 1 to generate interrupts in 1-bit mode.
4. The Host starts to provide the clock;
5. Host sets IP to 4-bit mode through CMD52;
6. The Host starts to access IP.

4.10.2 Clear Interrupt

When users detects that the SDIO Host is accessing register clearing interrupt, the user pulls down the fun1_interrupt pin to 0 to clear the interrupt. The clear interrupt register is defined by user.

The SDIO standard suggests time of clearing interrupt:

If users accesses interrupt clear through CMD52, it needs to clear interrupt before the end of R5;

If users accesses interrupt clear through CMD53, it needs to clear interrupt before the data transfer is completed.

4.11 User Interface

4.11.1 CPU Slave Interface

CPU Slave Interface Clock

The CPU Slave Interface clock is cpu_clk, independent of sdio_clk. In use, cpu_clk can connect to sdio_clk to synchronize clocks on the two interfaces.

CPU Slave Interface Function

CCCR register and FBR (Function1) register have been implemented in IP. Users can access the corresponding register through the CPU Slave interface in the initialization stage. When the host accesses these registers, users does not need to participate. In addition, users can also set other functions of IP through the CPU Slave interface. The register definition is shown in Table 4-2.

The data interface of CPU Slave interface is 32 bits and the address is aligned with 4 bytes.

Register Definition of CPU Slave Interface

The register definition of CPU Slave Interface is as shown in Table 4-2.

Table 4-2 Register Definition of CPU Slave Interface

Address	Bit	Name	Default Value	Type	Description
0x00	3:0	CCCR/FBR Format Version number	0x3	RO	CCCR/FBR version Corresponding to CCCR Reg0x0 bit3~bit0
	7:4	SDIO Specification Revision Number	0x5	RO	SDIO version Corresponding to CCCR Reg0x0 bit7~bit4
	11:8	SD Format Version Number	0x4	RO	SD version Corresponding to CCCR Reg0x1 bit3~bit0
	16	Support Master Power Control	0x0	RW	Master Power Control Corresponding to CCCR Reg0x12 bit0
	17	Enable Master Power Control	0x0	RO	Enable Master Power Control Corresponding to CCCR Reg0x12 bit1
	8:18 PM	Total Card Power	0x0	RO	Total Card Power Corresponding to CCCR Reg0x12 bit4~bit2
	24	Support High-Speed	0x1	RW	Support High-Speed Corresponding to CCCR Reg0x13 bit0
	27:25	Bus Speed Select	0x0	RO	Bus Speed Select Corresponding to CCCR Reg0x13 bit3~bit1
0x04	0	Interrupt Enable Master	0x0	RO	Interrupt Enable Master Corresponding to CCCR Reg0x4 bit0
	7:1	Interrupt Enable for Function x	0x00	RO	Interrupt Enable for Function x Corresponding to CCCR Reg0x4 bit7~bit1

Address	Bit	Name	Default Value	Type	Description
	9:8	Bus Width	0x0	RO	Bus Width Corresponding to CCCR Reg0x7 bit1~bit0
	10	Support 8-bit Bus Mode	0x0	RO	Support 8-bit Bus Mode Corresponding to CCCR Reg0x7 bit2
	13	Enable Continuous SPI Interrupt	0x0	RO	Enable Continuous SPI Interrupt Corresponding to CCCR Reg0x7 bit5
	14	Support Continuous SPI interrupt	0x1	RW	Support Continuous SPI interrupt Corresponding to CCCR Reg0x7 bit6
	15	Card Detect Disable	0x0	RO	Card Detect Disable Corresponding to CCCR Reg0x7 bit7
	16	Support Direct Command (CMD52)	0x1	RO	Support Direct Command (CMD52) Corresponding to CCCR Reg0x8 bit0
	17	Support Multiple Block Transfer	0x1	RO	Support Multiple Block Transfer Corresponding to CCCR Reg0x8 bit1
	18	Support Read Wait	0x0	RO	Support Read Wait Corresponding to CCCR Reg0x8 bit2
	19	Support Bus Control	0x0	RO	Support Bus Control Corresponding to CCCR Reg0x8 bit3
	20	Support Block Gap Interrupt	0x0	RO	Support Block Gap Interrupt Corresponding to CCCR Reg0x8 bit4
	21	Enable Block Gap Interrupt	0x0	RO	Enable Block Gap Interrupt Corresponding to CCCR Reg0x8 bit5
	22	Low-Speed Card	0x0	RW	Low-Speed Card Corresponding to CCCR Reg0x8 bit6
	23	4-bit Mode Support for Low-Speed Card	0x0	RW	4-bit Mode Support for Low-Speed Card Corresponding to CCCR Reg0x8 bit7
	24	Support SDR50	0x1	RW	Support SDR50 Corresponding to CCCR Reg0x14 bit0
	25	Support SDR104	0x0	RO	Support SDR104 Corresponding to CCCR Reg0x14 bit1
	26	Support DDR50	0x0	RO	Support DDR50 Corresponding to CCCR Reg0x14 bit2
	27	Support Asynchronous	0x0	RO	Support Asynchronous Interrupt

Address	Bit	Name	Default Value	Type	Description
		Interrupt			Corresponding to CCCR Reg0x16 bit0
	28	Enable Asynchronous Interrupt	0x0	RO	Enable Asynchronous Interrupt Corresponding to CCCR Reg0x16 bit1
0x08	23:0	Common CIS Pointer	0x001000	RW	Common CIS Pointer [23:16] corresponding to CCCR Reg0x0B bit7~bit0 [15:8] corresponding to CCCR Reg0x0A bit7~bit0 [7:0] corresponding to CCCR Reg0x09 bit7~bit0
	24	Support Driver Type A	0x0	RW	Support Driver Type A Corresponding to CCCR Reg0x15 bit0
	25	Support Driver Type C	0x0	RW	Support Driver Type C Corresponding to CCCR Reg0x15 bit1
	26	Support Driver Type D	0x0	RW	Support Driver Type D Corresponding to CCCR Reg0x15 bit2
	29:28	Driver Type Select	0x0	RO	Driver Type Select Corresponding to CCCR Reg0x15 bit5~bit4
0x0C	15:0	FN0 Block Size	0x0000	RO	Function0 Block Size [15:8] corresponding to CCCR Reg0x11 bit7~bit0 [7:0] corresponding to CCCR Reg0x10 bit7~bit0
0x10	7:0	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF0 bit7~bit0
	15:8	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF1 bit7~bit0
	23:16	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF2 bit7~bit0
	31:24	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF3 bit7~bit0
0x14	7:0	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF4 bit7~bit0
	15:8	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF5 bit7~bit0
	23:16	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF6 bit7~bit0
	31:24	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF7 bit7~bit0
0x18	7:0	Reserved for Vendors	0x00	RO	Reserved for Vendors

Address	Bit	Name	Default Value	Type	Description
					Corresponding to CCCR Reg0xF8 bit7~bit0
	15:8	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xF9 bit7~bit0
	23:16	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFA bit7~bit0
	31:24	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFB bit7~bit0
0x1C	7:0	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFC bit7~bit0
	15:8	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFD bit7~bit0
	23:16	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFE bit7~bit0
	31:24	Reserved for Vendors	0x00	RO	Reserved for Vendors Corresponding to CCCR Reg0xFF bit7~bit0
0x20	3:0	Standard SDIO Function Interface Code	0xF	RW	Standard SDIO Function Interface Code Corresponding to FBR1 Reg0x100 bit3~bit0
	15:8	Extended Standard SDIO Function Interface Code	0x00	RW	Extended Standard SDIO Function Interface Code Corresponding to FBR1 Reg0x101 bit7~bit0
	23:16	iSDIO Type Support Code	0x00	RW	iSDIO Type Support Code Corresponding to FBR1 Reg0x108 bit7~bit0
	31:24	Standard iSDIO Function Interface Code	0x00	RW	Standard iSDIO Function Interface Code Corresponding to FBR1 Reg0x103 bit7~bit0
0x24	15:0	Function 1 I/O Block Size	0x0000	RO	Function 1 Block Size [15:8] corresponding to FBR1 Reg0x111 bit7~bit0 [7:0] corresponding to FBR1 Reg0x110 bit7~bit0
	16	Support Power Selection	0x0	RW	Support Power Selection Corresponding to FBR1 Reg0x102 bit0
	20:17	Power State	0x0	RO	Power State Corresponding to FBR1 Reg0x102 bit7~bit4
	21	Enable Power Selection	0x0	RO	Enable Power Selection Corresponding to FBR1 Reg0x102 bit1
	24	Function1 Supports CSA	0x0	RO	Support CSA Corresponding to FBR1 Reg0x100 bit6

Address	Bit	Name	Default Value	Type	Description
	25	Function1 CSA Enable	0x0	RO	Function1 CSA Enable Corresponding to FBR1 Reg0x100 bit7
0x28	15:0	Function1 MID_CARD Manufacturer Information	0x0000	RW	Function1 MID_CARD Manufacturer Information [15:8] corresponding to FBR1 Reg0x107 bit7~bit0 [7:0] corresponding to FBR1 Reg0x106 bit7~bit0
	31:16	Function1 SDA_MID_MANF SDIO Card Manufacturer Code	0x0000	RW	Function1 SDA_MID_MANF SDIO Card Manufacturer Code [31:24] corresponding to FBR1 Reg0x105 bit7~bit0 [23:16] corresponding to FBR1 Reg0x104 bit7~bit0
0x2C	23:0	Address pointer to Function1 CIS	0x002000	RW	Address pointer to Function1 CIS [23:16] corresponding to FBR1 Reg0x10B bit7~bit0 [15:8] corresponding to FBR1 Reg0x10A bit7~bit0 [7:0] corresponding to FBR1 Reg0x109 bit7~bit0
0x30	0	IO_Ready	0x0	RW	IO_Ready This bit corresponding to the C field in R4. After power on, the bit is 0. After the user initializing the register through the CPU Slave interface, set the bit to 1
	18:16	SDIO Bus State	0x0	RO	SDIO Bus State User read IP Bus states through this register. 0x0: Idle State, indicating that IP has not completed negotiation with Host through CMD5; 0x1: Initialization State, indicating that the IP has completed the negotiation with Host through CMD5; 0x2：Standby State 0x3：Command State 0x4：Transfer State 0x5：Inactive State Others: undefined
	24	Manual TX Clock to Data Phase	0x0	RW	The data sent by SDIO and SDIO clock phase are valid when the Manual TX Clock to Data Phase Enable register is enabled. 0: SDIO clock negedge sending data; 1: SDIO clock posedge

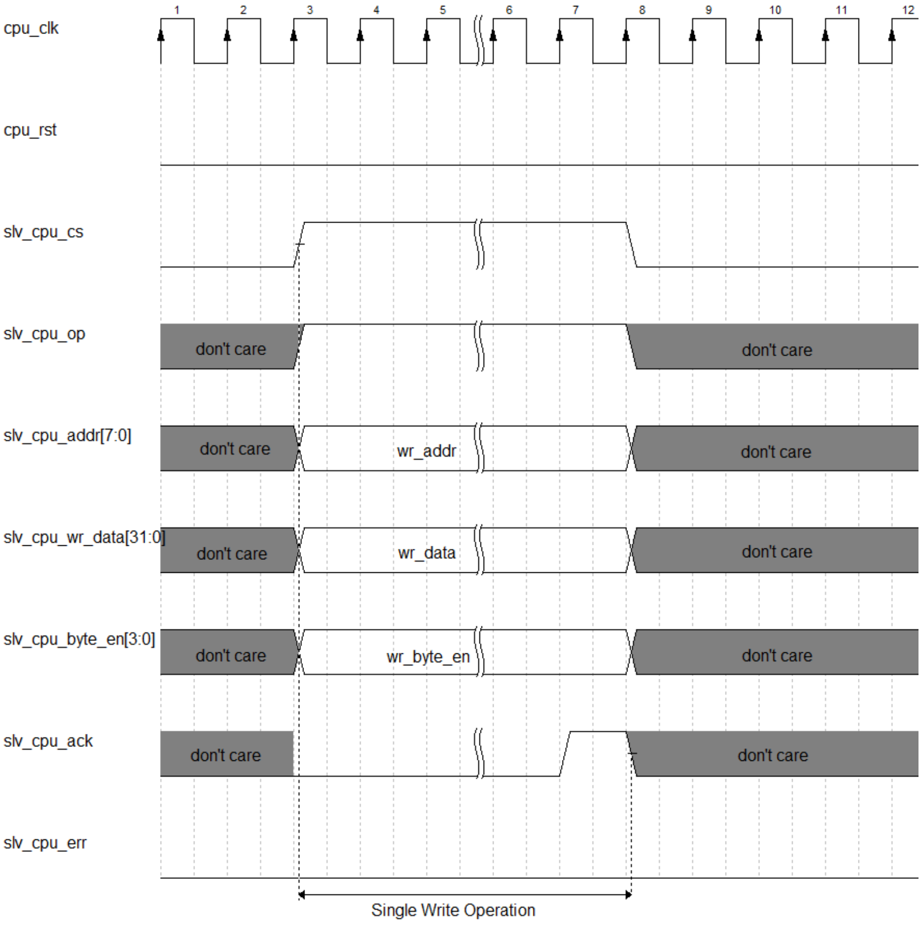
Address	Bit	Name	Default Value	Type	Description
					sending data;
	25	Manual TX Clock to Data Phase Enable	0x0	RW	The data sent by SDIO and SDIO clock phase are configured manually to enable 1: enable, configure based on the value of Manual TX Clock to Data Phase; 0: disable, automatic configuration according to the SDIO standard
0x34	15:0	CCCR Max Block Size	0x0800	RW	CCCR Max Block Size
	31:16	FBR1 Max Block Size	0x0800	RW	FBR1 Max Block Size

CPU Slave Interface Operating Timing

Write operating of CPU Slave Interface

When users need to write the register through the CPU Slave Interface, slv_cpu_cs needs to be pulled up to select the Interface, and slv_cpu_op also needs to be pulled up to indicate the write operation. Users send register address, write data, and enable byte through slv_cpu_addr, slv_cpu_wr_data, and slv_cpu_byte_en signals. When IP completes this write operation, the slv_cpu_ack signal is pulled up. Slv_cpu_cs, slv_cpu_op, slv_cpu_addr, slv_cpu_wr_data and slv_cpu_byte_en signals need to remain unchanged until slv_cpu_ack signal is pulled up. The write timing of CPU Slave Interface is as shown in Figure 4-6.

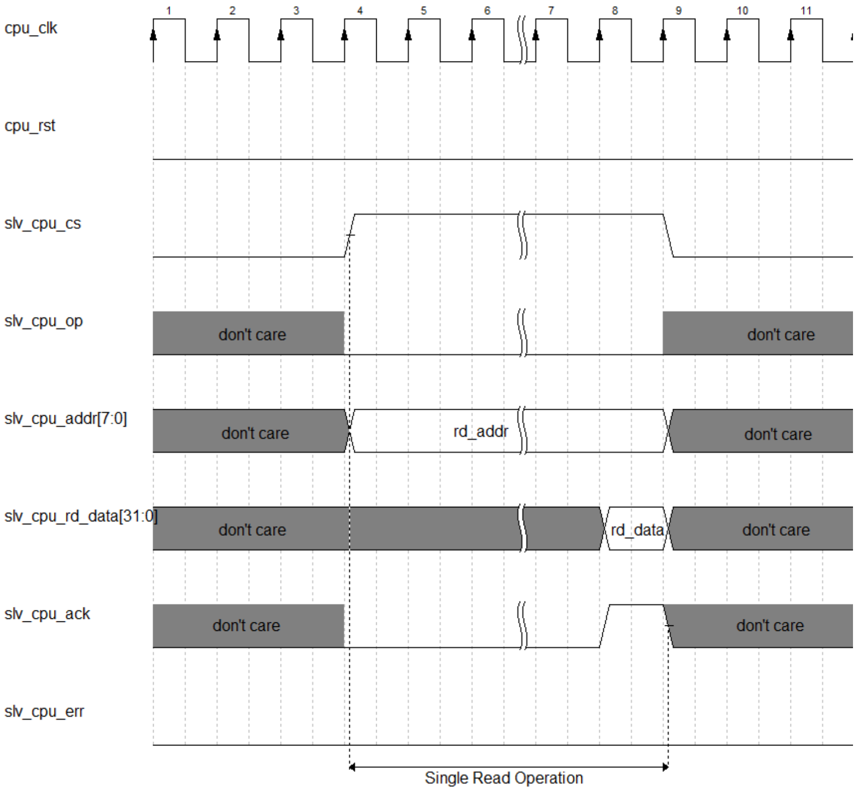
Figure 4-6 Write Timing of CPU Slave Interface



Read operating of CPU Slave Interface

When users need to read the register through the CPU Slave Interface, slv_cpu_cs should be pulled up to select the Interface and slv_cpu_op should be pulled down to indicate the read operation. Users send the register address through the slv_cpu_addr signal. When the IP completes the read operation, the slv_cpu_ack signal is pulled up and the read data is in slv_cpu_rd_data. Slv_cpu_cs, slv_cpu_op, slv_cpu_addr signals need to remain unchanged until slv_cpu_ack signal is pulled up. When the CPU Slave Interface executes read operation, IP will return 4 bytes of read data. The read timing of CPU Slave Interface is as shown in Figure 4-7.

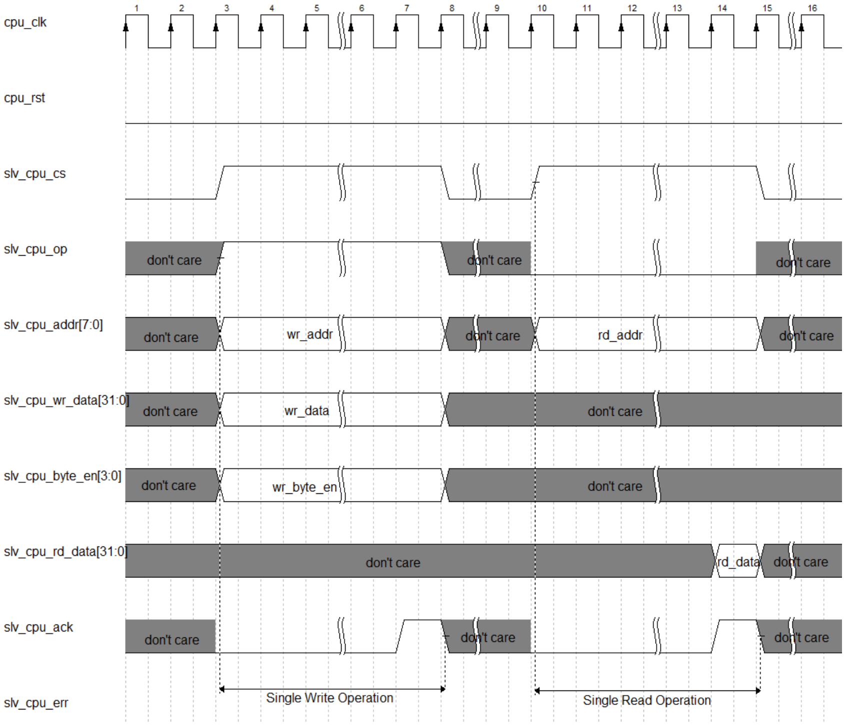
Figure 4-7 Read Timing of CPU Slave Interface



The read and write operating alternatively of CPU Slave Interface

The read and write operating alternatively timing of CPU Slave Interface are as shown in Figure 4-8.

Figure 4-8 Read and Write Alternatively Timing of CPU Slave Interface



4.11.2 SDIO CMD52 Interface

When Host accesses CIS or Function1 registers through CMD52, the user needs to send and receive data through SDIO CMD53 interface. After IP receives CMD52 accessing CIS or Function1 register, the Interface of SDIO CMD52 generates the corresponding data request, and the user shall respond according to the corresponding timing.

Write Operating of SDIO CMD52 Interface (RAW flag = 1)

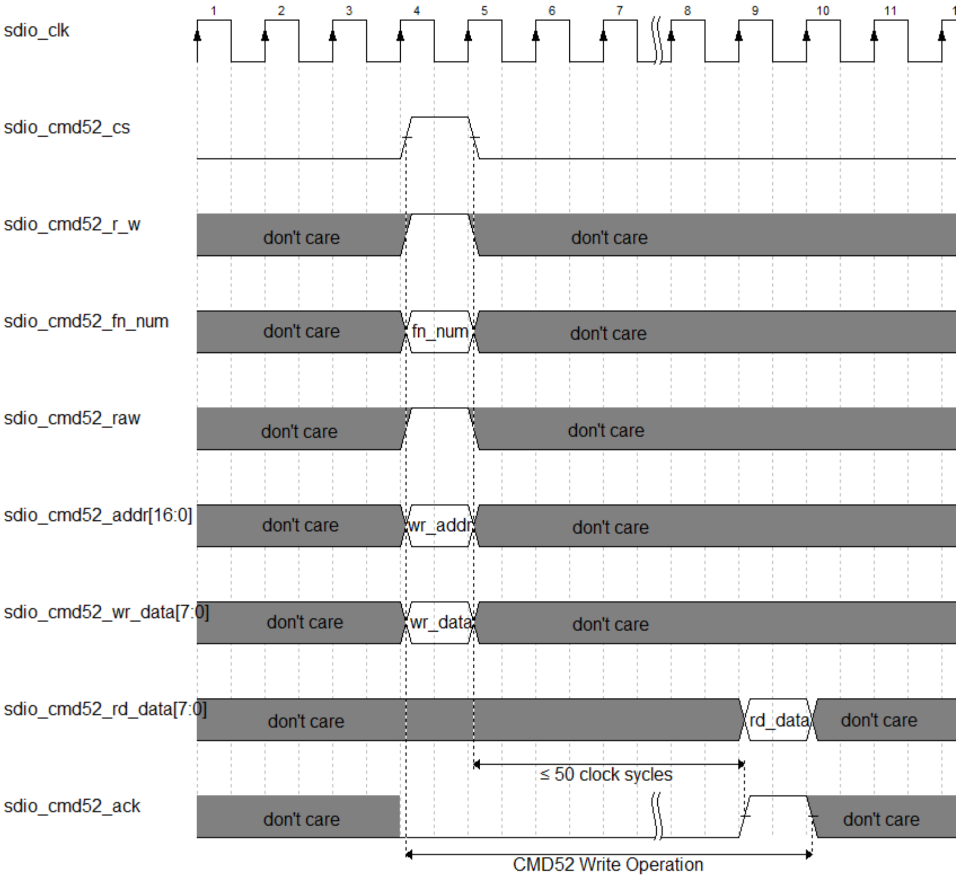
When Host executes write operation through CMD52, users need to return the current value of the register after completing the register write operation.

When IP receives a correct CMD52 write operation and the register address is Function1, IP generates a write data request on SDIO CMD52 Interface. At this time, IP pulls up sdio_cmd52_cs signal, indicating that the SDIO CMD52 Interface has a data request. At the same time, sdio_cmd52_r_w and sdio_cmd52_raw signals are pulled up, indicating that it is the write operation of RAW flag = 1. When sdio_cmd52_cs = 1, sdio_cmd52_fn_num, sdio_cmd52_addr, sdio_cmd52_wr_data signal are valid. After the user completes the write operation, it is required to pull up sdio_cmd52_ack one cycle to inform IP that the write operation is completed. When IP detects that sdio_cmd52_ack is 1, R5 is automatically sent to Host.

The write operation is executed when RAW flag = 1.Users need to return the current value of the write register in sdio_cmd52_rd_data while pulling up sdio_cmd52_rd_ack, which will return to the Host in the Read or Write Data field of R5.

The write timing of SDIO CMD52 Interface (RAW flag = 1) is as shown in Figure 4-9.

Figure 4-9 Write Timing of SDIO CMD52 Interface (RAW flag = 1)



Write Operating of SDIO CMD52 Interface (RAW flag = 0)

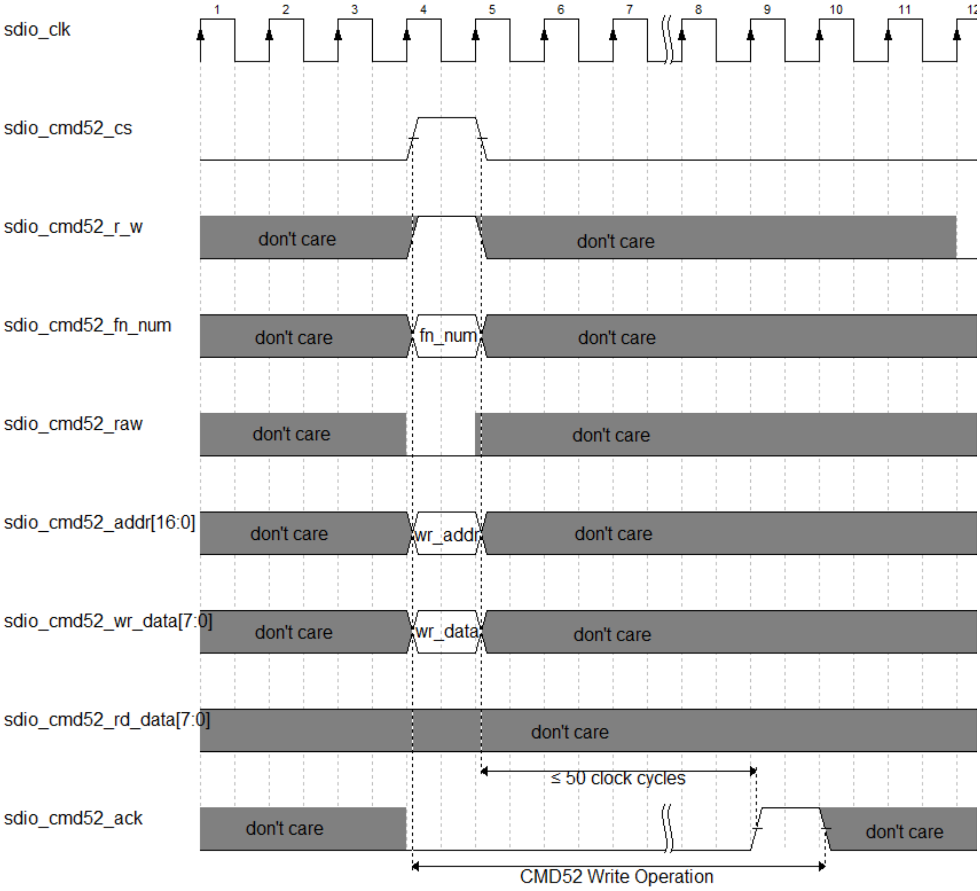
When Host executes write operation through CMD52 and RAW flag is 0, the user does not need to return the current value of the register after completing the register write operation.

When IP receives a correct CMD52 write operation and the register address is Function1, IP generates a write data request on SDIO CMD52 Interface. At this time, IP pulls up sdio_cmd52_cs signal, indicating that the SDIO CMD52 Interface has a data request. At the same time, sdio_cmd52_r_w signal is pulled up and sdio_cmd52_raw signal is pulled down, indicating that it is the write operation of RAW flag = 0. When sdio_cmd52_cs = 1, sdio_cmd52_fn_num, sdio_cmd52_addr, sdio_cmd52_wr_data signal are valid. After users completes the write operation, it is required to pull up sdio_cmd52_ack one cycle to inform IP that the write operation is completed. When IP detects that sdio_cmd52_ack is 1, R5 is automatically sent to Host.

The write operation is executed when RAW flag = 0. Users do not need to return the current value of the write register in sdio_cmd52_rd_data while pulling up sdio_cmd52_ack. IP will automatically put the field value of CMD52 Write Data in Read or Write Data field of R5, returning to Host.

The write timing of SDIO CMD52 Interface (RAW flag = 0) is as shown in Figure 4-10.

Figure 4-10 Write Timing of SDIO CMD52 Interface (RAW flag = 0)



SDIO CMD52 Interface Read Operating

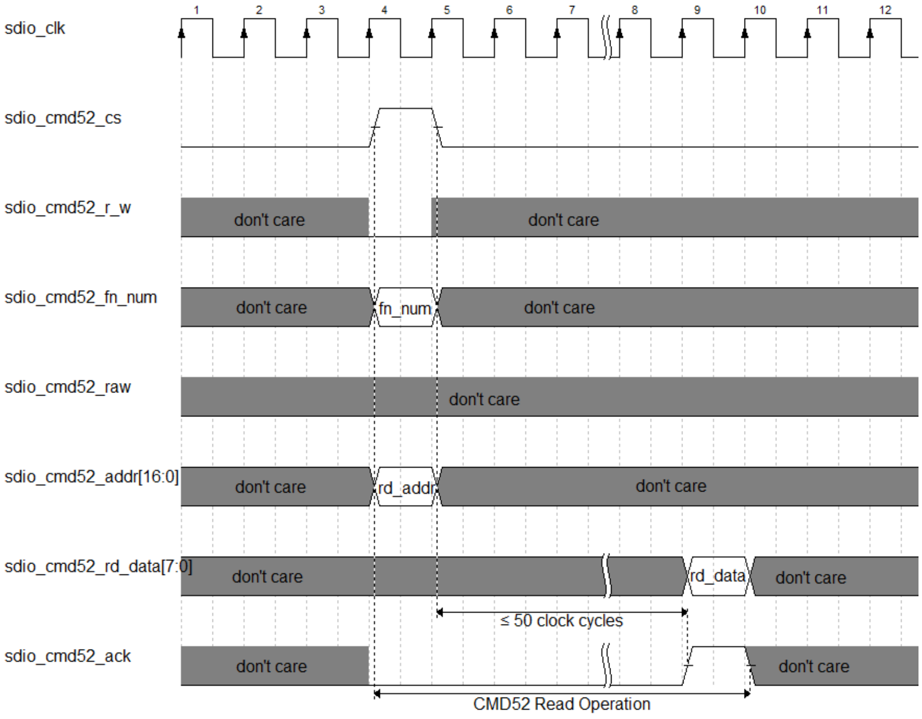
When Host executes read operation through CMD52, the user returns the register value after receiving the read request.

When IP receives a correct CMD52 write operation and the register address is Function1, IP generates a write data request on SDIO CMD52 Interface. At this time, IP pulls up sdio_cmd52_cs signal, indicating that the SDIO CMD52 Interface has a data request. At the same time, sdio_cmd52_r_w signal is pulled up, indicating that it is the read operaiton. When sdio_cmd52_cs = 1, sdio_cmd52_fn_num and sdio_cmd52_addr signal are valid. After the user completes the read operation, it is required to pull up sdio_cmd52_ack one cycle to inform IP that the read operation is completed. When IP detects that sdio_cmd52_ack is 1, R5 is automatically sent to Host.

Since it is a read operation, the user needs to return the read register value in sdio_cmd52_rd_data while pulling up sdio_cmd52_rd_ack. The value will return to the Host in the Read or Write Data field of R5.

The read operation timing of SDIO CMD52 Interface is as shown in Figure 4-11.

Figure 4-11 Read Timing of SDIO CMD52 Interface



CMD52 Timeout

According to the SDIO standard, the delay range of NCR from the last bit of Command to the first bit of Response is 2~64 clock cycles. Therefore, the user should not have more than 50 clock cycles interval between valid sdio_cmd52_cs and valid sdio_cmd52_ack (Due to sampling delays within IP). In 50 clock cycles after sdio_cmd52_cs is valid, if the user can not return to sdio_cmd52_ack, please give up this CMD52 operation, and sdio_cmd52_ack does not to pull up again. The Host considers this operation times out.

4.11.3 SDIO CMD53 Interface

When Host reads and writes CIS or Function1 registers through CMD53, the user need to control data transfer through SDIO CMD53 Interface. After IP receives CMD53 of reading and writing CIS or Function1 register, the interface of SDIO CMD53 generates the corresponding data request, and the user shall respond according to the corresponding timing.

The SDIO CMD53 command includes Bytes Transfer mode and Block Transfer mode. When CMD53 is in Bytes Transfer mode, SDIO CMD53 Interface generates a read-write operation, and the length is CMD53 Byte/Block Count field. When CMD53 is in Block Transfer mode, SDIO CMD53 Interface generates N times of read and write, N is CMD53 Byte/Block Count field, and the length is corresponding Function Block Size. Therefore, the user cannot perceive whether the operation is Bytes Transfer or Block Transfer. He only needs to transfer data in accordance with the corresponding timing.

SDIO CMD53 Interface Write Operation

Each write request represents a Bytes Transfer write or a single Block write in a Block Transfer.

When IP receives a correct CMD53 write operation and the register address is Function1, IP automatically responds to R5. The write data request is then generated on the SDIO CMD53 Interface. IP pulls up sdio_cmd52_cs signal, indicating there is a write data request on SDIO CMD53 Interface. When sdio_cmd52_cs = 1, sdio_cmd53_fn_num, sdio_cmd53_addr, sdio_cmd53_len, sdio_cmd53_op_code signals are valid.

When Host writes data to IP through sdio_dat, users can receive the write data through sdio_cmd53_wr_valid and sdio_cmd53_wr_data. The number of write bytes at one time is equal to the value of sdio_cmd53_len. IP automatically verifies the correctness of CRC16 and End bit.

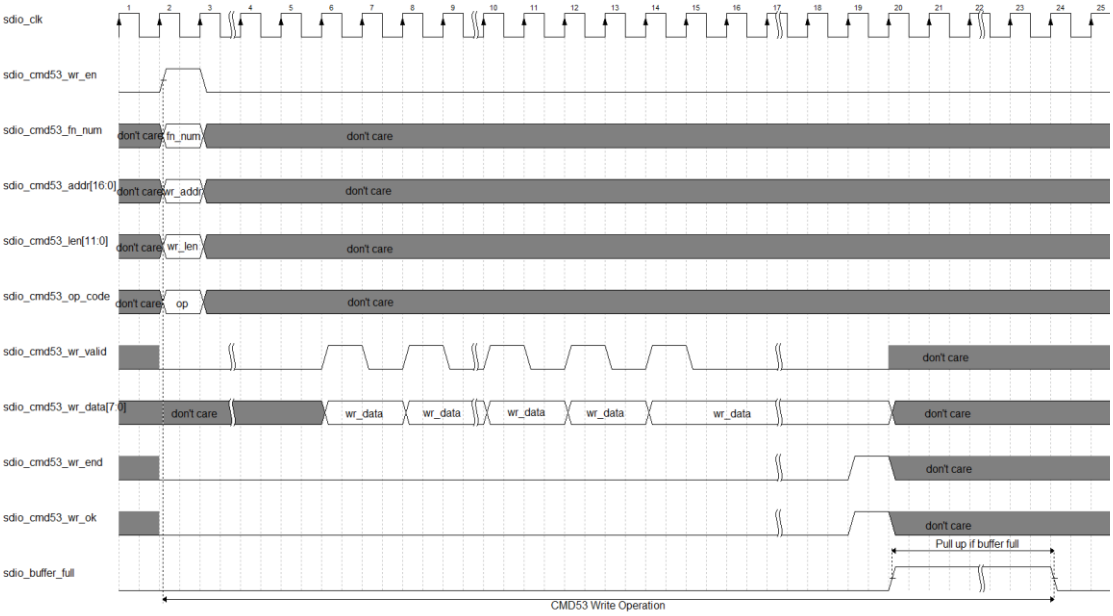
When Host completes a write operation in Bytes Transfer or a single Block write operation in Block Transfer, and IP has sent CRC status and End bit to Host, sdio_cmd53_wr_end is pulled up to inform the user that the write operation is completed. Sdio_cmd53_wr_ok is used to indicate whether the write data is correct or not. sdio_cmd53_wr_ok synchronizes with sdio_cmd53_wr_end. When sdio_cmd53_wr_ok is 1, it means that the write operation is correct and the user needs to process the data of the write operation. When sdio_cmd53_wr_ok is 0, it means that the write operation has CRC16 or End bit or CRC status sending error, and the user needs to abandon the data of the write operation.

In Block Transfer, if IP receives a Block write operation with an error, IP will abandon the subsequent data reception.

When user detects sdio_cmd53_wr_end = 1, he needs to inform IP whether there is an idle buffer to continue receiving the subsequent data through sdio_buffer_full. If the user's buffer is full and cannot continue to receive data, sdio_buffer_full needs to be pulled up immediately after sdio_cmd53_wr_end = 1 is detected. When the user pulls up sdio_buffer_full, IP will pull down sdio_dat0 to the card busy state, until the user pulls down sdio_buffer_full. When Host detects that sdio_dat0 is 0, the next write will not continue until it detects that sdio_dat0 is pulled up.

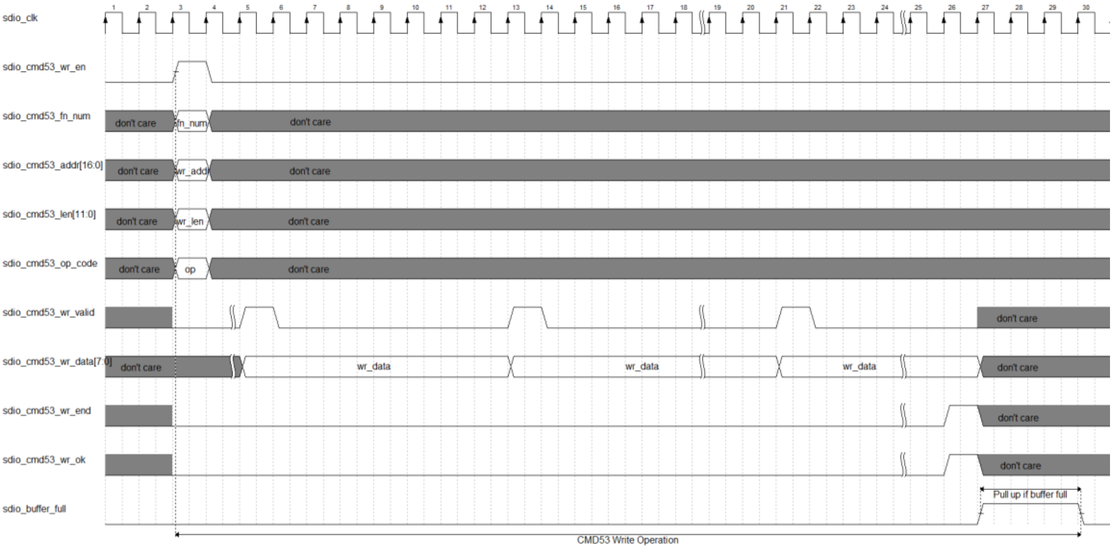
The write operation timing of SDIO CMD53 Interface in the 4-bit mode is shown in Figure 4-12.

Figure 4-12 The Write Timing of SDIO CMD53 Interface in the 4-bit Mode



The write timing of SDIO CMD53 Interface in the 1-bit mode is shown in Figure 4-13.

Figure 4-13 The Write Timing of SDIO CMD53 Interface in the 1-bit Mode



SDIO CMD53 Interface Read Operating

Each read operation request means a read operation in Bytes Transfer or a single Block read in Block Transfer.

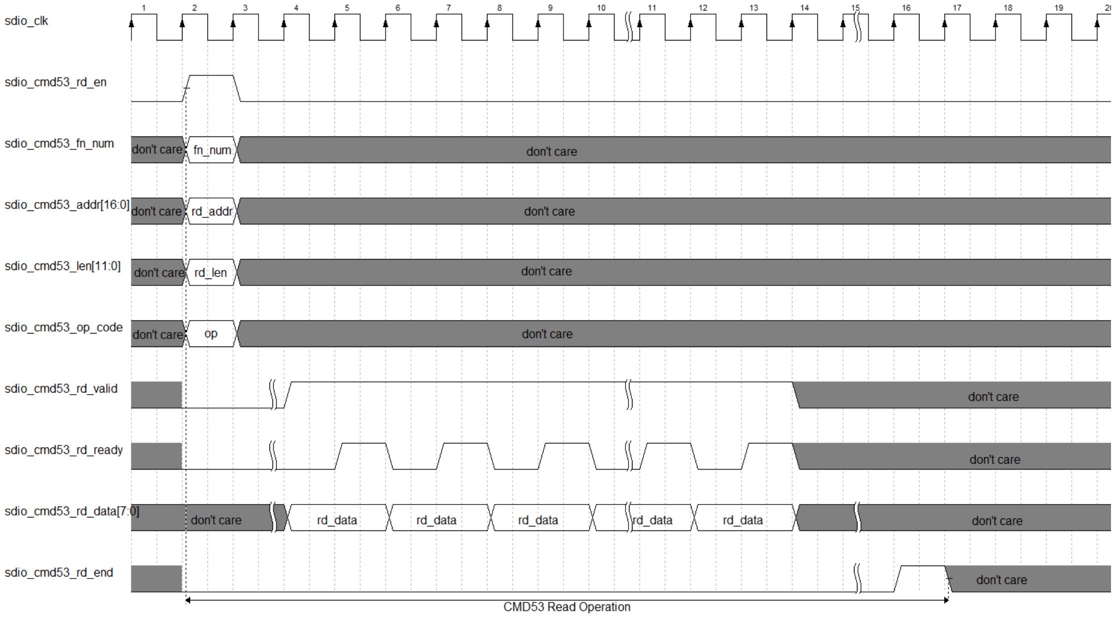
When IP receives a correct CMD53 read operation and the register address is Function1, IP automatically responds to R5. The read data request is then generated on the SDIO CMD53 Interface. IP pulls up sdio_cmd53_rd_en signal, indicating there is a read data request on the SDIO CMD53 Interface. When sdio_cmd52_cs = 1, sdio_cmd53_fn_num, sdio_cmd53_addr, sdio_cmd53_len, sdio_cmd53_op_code signals are valid.

When the user receives the CMD53 read request, the user can send the read data through sdio_cmd53_rd_valid, sdio_cmd53_rd_data and sdio_cmd53_rd_ready. The sdio_cmd53_rd_ready signal is pulled up every time, indicating that IP receives a byte of read data, and the user needs to get the new data ready again in the sdio_cmd53_rd_data signal. The number of read bytes is equal to the value of sdio_cmd53_len. The user needs to control the length of sdio_cmd53_rd_valid according to the value of sdio_cmd53_len. After sending enough bytes, the user pulls down sdio_cmd53_rd_valid. IP sends data to sdio_dat based on the length of request data of Host. After sending data, IP will automatically send CRC16 field and End bit, and then IP will pull up sdio_cmd53_rd_end signal one cycle to inform the user that the read operation is completed.

IP does not support Read Wait function.

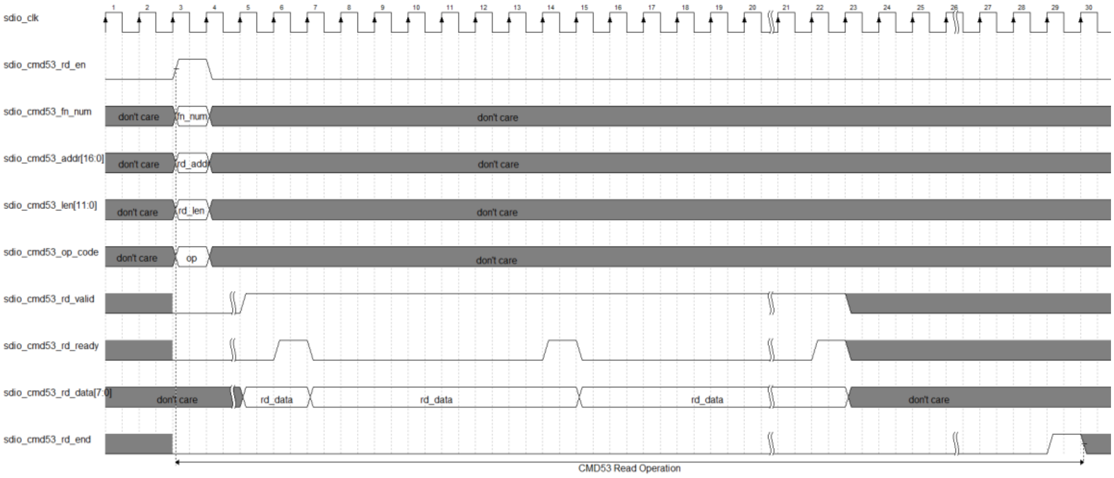
The read operation timing of SDIO CMD53 Interface in the 4-bit mode is shown in Figure 4-14.

Figure 4-14 The Read Timing of SDIO CMD53 Interface in the 4-bit Mode



The read timing of SDIO CMD53 Interface in the 1-bit mode is shown in Figure 4-15.

Figure 4-15 The Read Timing of SDIO CMD53 Interface in the 1-bit Mode



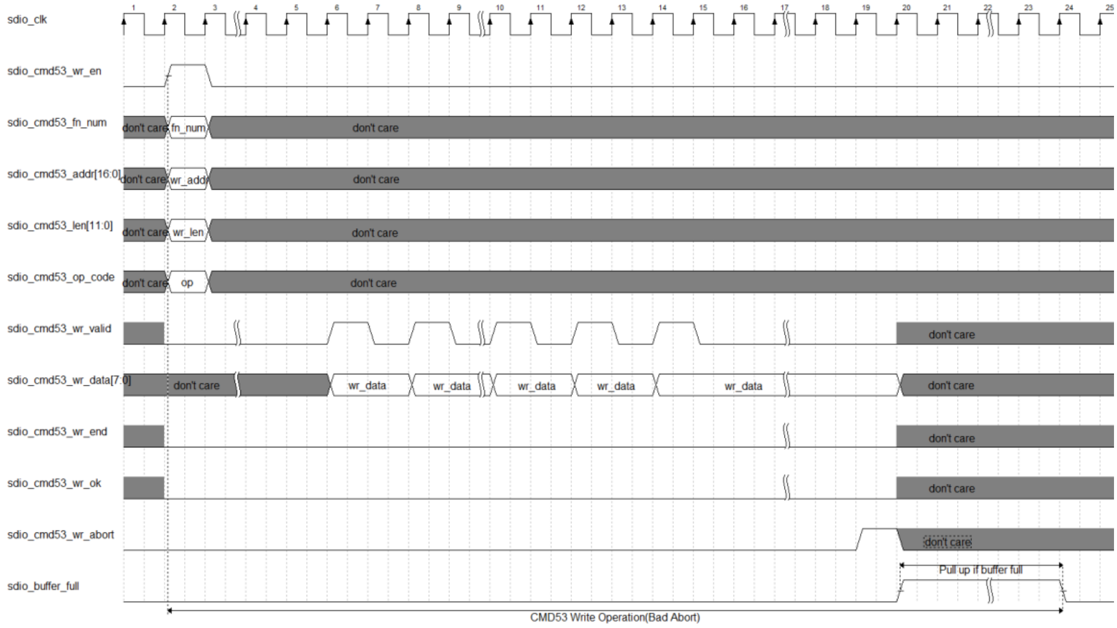
Abort

When Host reads and writes data through CMD53 in Infinite Block Mode, the read and write can be interrupted by Abort sent by CMD52.

When IP receives CMD52 Abort and is writing data, IP will pull up sdio_cmd53_wr_abort one cycle to inform the user to abort the write.

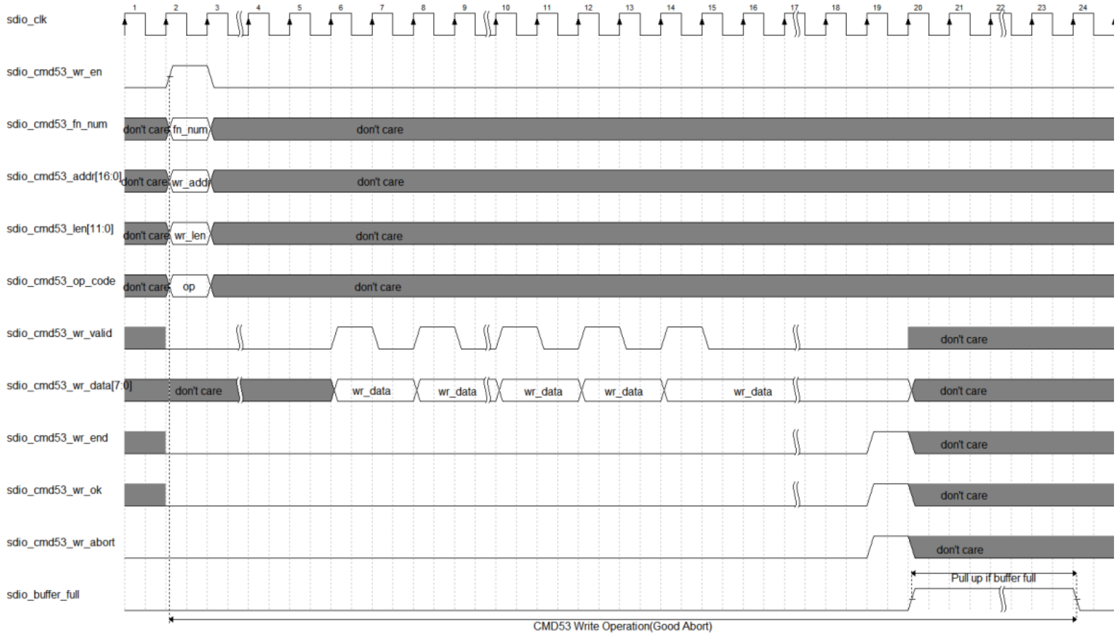
If the user detects sdio_cmd53_wr_abort = 1 and does not detect sdio_cmd53_wr_end = 1 before, then this abort triggers a Bad abort and IP will receive abort before CRC status and End bit are sent. IP no longer generates sdio_cmd53_wr_end = 1. The user needs to end the write operation and abandon the write data, as shown in Figure 4-16.

Figure 4-16 Write Bad Abort



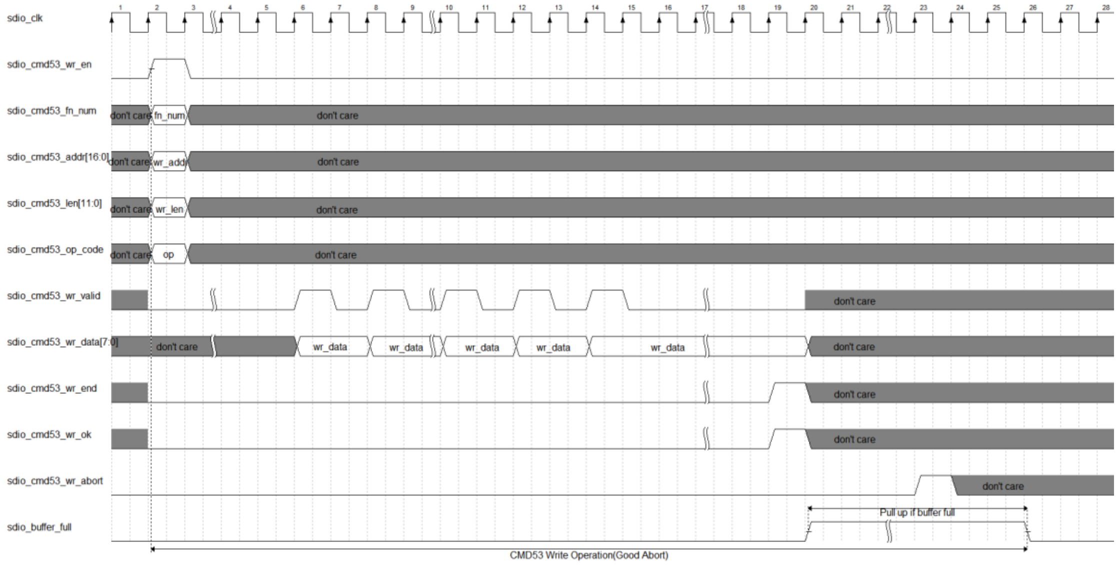
If the user receives both sdio_cmd53_wr_abort = 1 and sdio_cmd53_wr_end = 1, this abort triggers a good abort, which IP receives when it sends CRC status and End bit. The user needs to receive the write data, as shown in Figure 4-17.

Figure 4-17 Write Good Abort Case1



If the user receives sdio_cmd53_wr_abort = 1 after sdio_cmd53_wr_end = 1, it indicates that abort is triggered in the card busy state, which means IP receives abort after CRC status and End bit sent, and it is a good abort. The user needs to receive the write data, as shown in Figure 4-18.

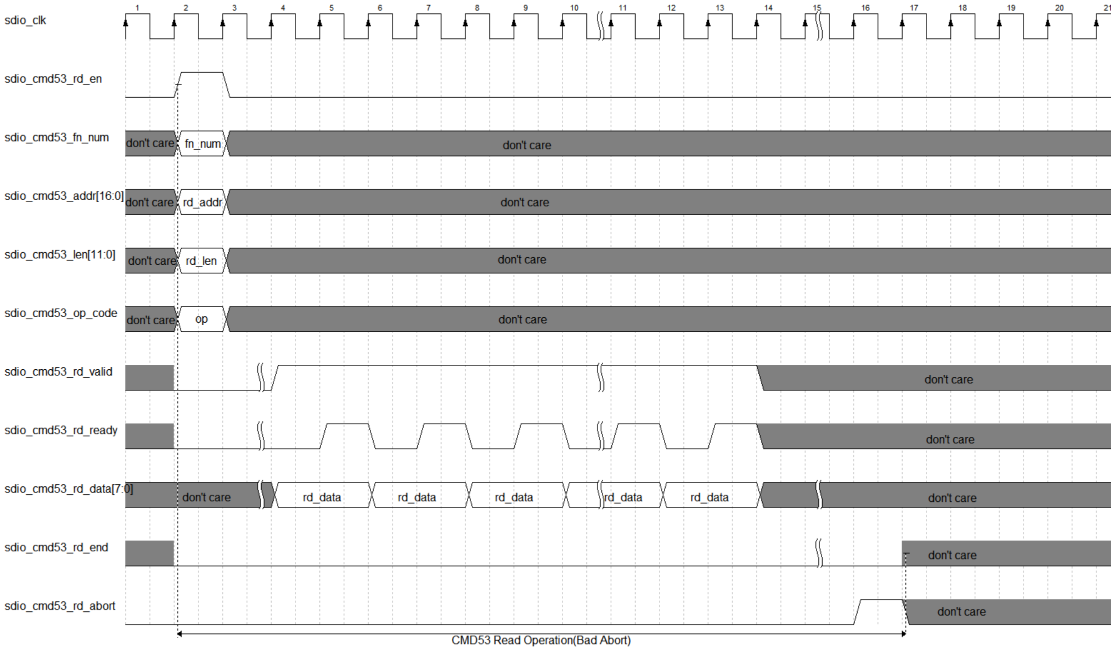
Figure 4-18 Write Good Abort Case2



When IP receives CMD52 Abort and is reading data, IP will pull up sdio_cmd53_rd_abort one cycle to inform the user to abort the read.

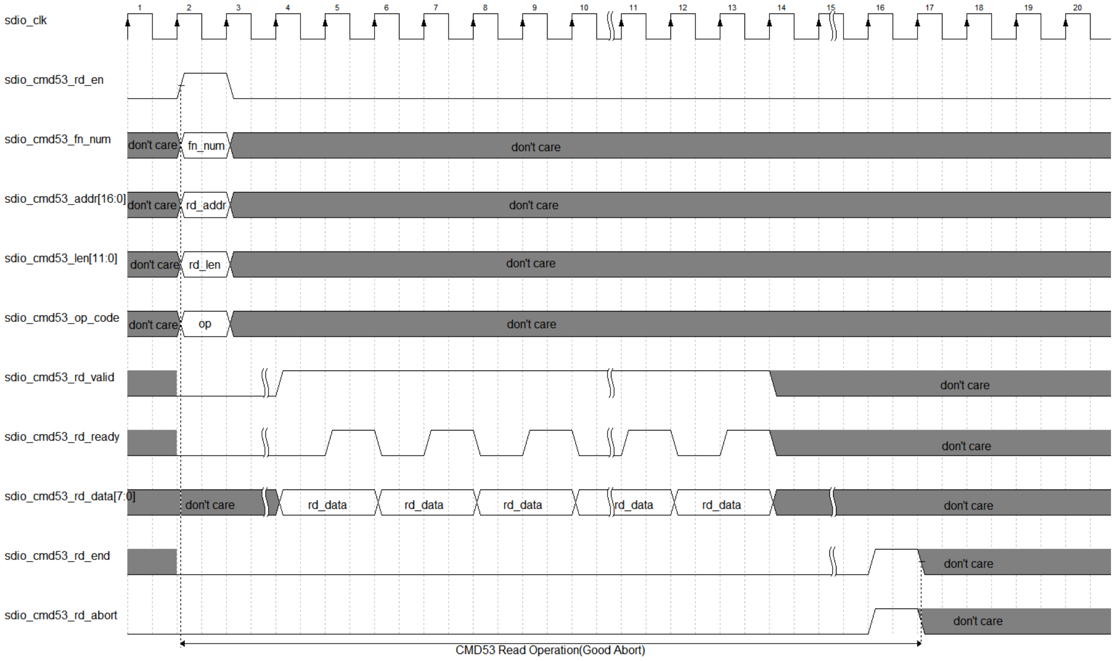
If the user detects sdio_cmd53_rd_abort = 1 and does not detect sdio_cmd53_rd_end = 1 before, then this abort triggers a Bad abort and IP will receive abort before CRC status and End bit are sent. IP no longer generates dio_cmd53_rd_end = 1. The user needs to end the read operation, which is unsuccessful, as shown in Figure 4-19.

Figure 4-19 Read Bad Abort



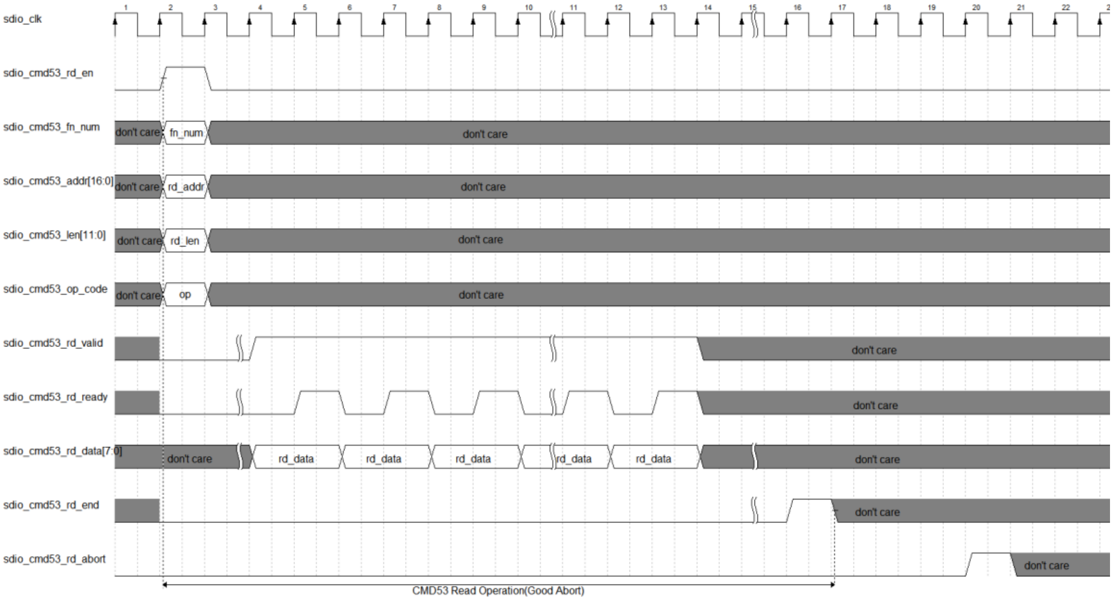
If the user receives both sdio_cmd53_rd_abort = 1 and sdio_cmd53_rd_end = 1, this abort triggers a good abort, which IP receives when it sends CRC status and End bit., and the read operation is successful as shown in Figure 4-20.

Figure 4-20 Read Good Abort Case1



If the user receives sdio_cmd53_rd_abort = 1 after sdio_cmd53_rd_end = 1, it indicates that abort is triggered between two reads. IP receives abort after CRC16 and End bit are sent, and it is a Good abort as shown in Figure 4-21.

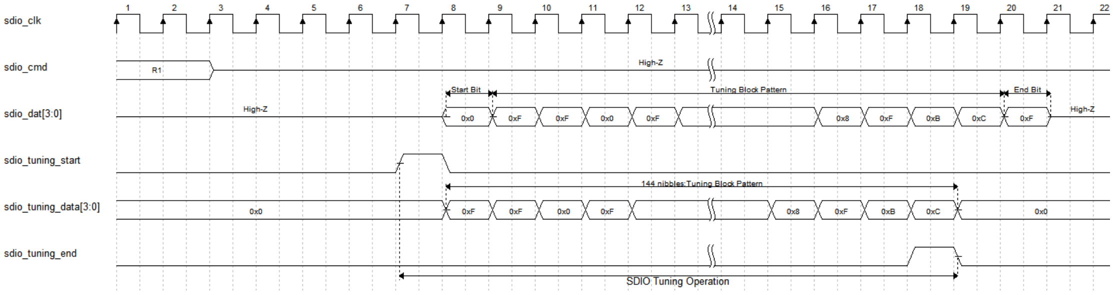
Figure 4-21 Read Good Abort Case2



4.11.4 SDIO Tuning Interface

CMD19 is supported when IP is configured to UHS-I Card. IP receives the correct CMD19 and responds to R1 automatically. After R1 is sent, user sends Tuning Block Pattern through the SDIO Tuning Interface. Please see Figure 4-5 for Tuning Block Pattern The timing of SDIO Tuning Interface is shown in Figure 4-22.

Figure 4-22 Timing of SDIO Tuning Interface



5 Port List

The IO ports of Gowin SDIO Slave Controller IP are shown in Table 5-1.

Table 5-1 IO Ports of Gowin SDIO Slave Controller IP

Signal	I/O	Bit Width	Description
SDIO Interface			
sdio_clk	input	1	SDIO clock input
sdio_cmd_in	input	1	SDIO Command/Response input
sdio_cmd_out	output	1	SDIO Command/Response output
sdio_cmd_oen	output	1	SDIO Command/Response output enable, active-low
sdio_dat0_in	input	1	The SDIO 1-bit mode is input of dat signal and the SDIO 4-bit mode is input of dat0 signal
sdio_dat0_out	output	1	The SDIO 1-bit mode is output of dat signal and the SDIO 4-bit mode is output of dat0 signal
sdio_dat0_oen	output	1	The SDIO 1-bit mode is output enable of dat signal and the SDIO 4-bit mode is output enable of dat0 signal
sdio_dat1_in	input	1	The SDIO 1-bit mode is the input of interrupt signal and the SDIO 4-bit mode is the input of dat1 and interrupt multiplexed signal
sdio_dat1_out	output	1	The SDIO 1-bit mode is output of interrupt signal and the SDIO 4-bit mode is output of dat1 and interrupt multiplexed signal
sdio_dat1_oen	output	1	The SDIO 1-bit mode is output enable of interrupt signal and the SDIO 4-bit mode is output enable of dat1 and interrupt multiplexed signal
sdio_dat2_in	input	1	The SDIO 1-bit mode is the input of Read Wait signal and the SDIO 4-bit mode is the input of dat2 and Read Wait multiplexed signal
sdio_dat2_out	output	1	SDIO 1-bit mode is the output of Read Wait signal and SDIO 4-bit mode is the output of dat2 and Read Wait multiplexed signal
sdio_dat2_oen	output	1	SDIO 1-bit mode is output enable of Read Wait signal and SDIO 4-bit mode is output enable of dat2 and Read Wait multiplexed signal, active-low

Signal	I/O	Bit Width	Description
sdio_dat3_in	input	1	The SDIO 1-bit mode is not defined and the SDIO 4-bit mode is the input of dat0
sdio_dat3_out	output	1	The SDIO 1-bit mode is not defined and the SDIO 4-bit mode is the output of dat0
sdio_dat3_oen	output	1	The SDIO 1-bit mode is not defined the SDIO 4-bit mode is the output enable of dat3 signal
Reset			
rstn	input	1	Global asynchronous reset, active-low
cmd52_rst	output	1	When Host reset SDIO through cmd52, this signal is 1. Synchronize with sdio_clk
Function1 Enable/Ready			
fun1_ioe	output	1	Function1 enable signal, reflecting the value of CCCR IOE1 Synchronize with sdio_clk
fun1_ior	input	1	Function1 is ready Corresponding to CCCR IOR1 Reg0x3 bit1
Function1 Interrupt			
fun1_interrupt	input	1	0: Function1 without interrupt 1: Function1 with interrupt
CPU Slave Interface（Synchronize with cpu_clk）			
cpu_clk	input	1	cpu interface clock input
cpu_rst	input	1	CPU interface synchronous reset input signal and synchronize with cpu_clk, 1 valid
slv_cpu_cs	input	1	CPU interface chip selection signal, active-high
slv_cpu_op	input	1	CPU interface operation mode; 0: read operation; 1: write operation
slv_cpu_addr	input	8	cpu interface operation address
slv_cpu_wr_data	input	32	cpu interface write data

slv_cpu_byte_en	input	4	CPU interface write operation byte enable:1: enable; 0: disable
			bit3 corresponding to slv_cpu_wr_data[31:24];
			bit2 corresponding to slv_cpu_wr_data[23:16];
			bit1 coreesponding to slv_cpu_wr_data[15:8];
			bit0 corresponding to slv_cpu_wr_data[7:0]

slv_cpu_rd_data	output	32	cpu interface read data
slv_cpu_ack	output	1	CPU interface operation response, when this bit is 1, indicating the operation is completed
slv_cpu_err	output	1	CPU interface operation error indication Valid when slv_cpu_ack is 1. 1: error operation; 0: correct operation.
SDIO CMD52 Interface（Synchronize with sdio_clk）			
sdio_cmd52_cs	output	1	SDIO CMD52 command enable 0: disable; 1: enbale
sdio_cmd52_r_w	output	1	SDIO CMD52 read and write operation, corresponding to IO_RW_DIRECT Command (CMD52) Command R/W flag field. 0: read operation; 1: write operation
sdio_cmd52_fn_num	output	1	SDIO CMD52 Function Number, corresponding to IO_RW_DIRECT Command (CMD52) Function Number field. 0：Function0；1：Function1

Signal	I/O	Bit Width	Description
sdio_cmd52_raw	output	1	SDIO CMD52 RAW enable, corresponding to IO_RW_DIRECT Command (CMD52) Command RAW flag field. 0: disable RAW; 1: enble RAW
sdio_cmd52_addr	output	17	SDIO CMD52 operation address, corresponding to IO_RW_DIRECT Command (CMD52) Command Register Address field.
sdio_cmd52_wr_data	output	8	SDIO CMD52 write data, corresponding to IO_RW_DIRECT Command (CMD52) Command Write Data field. It is valid When the SDIO CMD52 interface is in write mode.
sdio_cmd52_rd_data	input	8	The SDIO CMD52 reads data or RAW data, corresponding to IO_RW_DIRECT Response (R5) command Read Data field. When the SDIO CMD52 interface is in read mode, the user needs to return the value of the read register. When the SDIO CMD52 interface is in RAW mode, the user needs to return the value of the write register
sdio_cmd52_ack	input	1	SDIO CMD52 ack signal. When the user pulls up the ack signal a single cycle, IP considers that the user has completed the CMD52 operation and starts sending R5. When this operation is a read operation, the user needs to return the value of the read address register to IP in sdio_cmd52_rd_data while pulling up the ack. When this operation is RAW operation, the user needs to return the value of the write address register to IP in sdio_cmd52_rd_data while pulling up the ack
SDIO CMD53 Interface（Synchronize with sdio_clk）			
sdio_cmd53_wr_en	output	1	SDIO CMD53 write enable 0: disable; 1: enbale
sdio_cmd53_rd_en	output	1	SDIO CMD53 read enable 0: disable; 1: enbale
sdio_cmd53_fn_num	output	1	SDIO CMD53 Function Number, corresponding to IO_RW_EXTENDED Command (CMD52) Function Number field. 0: Function0; 1: Function1
sdio_cmd53_addr	output	17	SDIO CMD53 operation initial address, corresponding to IO_RW_EXTENDED Command (CMD52) Command Register Address field.
sdio_cmd53_len	output	12	The operation length of SDIO CMD53 command means the number of bytes that the user needs to read and write.
sdio_cmd53_op_code	output	1	SDIO CMD53 OP Code, corresponding to IO_RW_EXTENDED Command (CMD52) OP Code field. 0：Multi byte R/W to fixed address； 1：Multi byte R/W to incrementing address
sdio_cmd53_wr_valid	output	1	SDIO CMD53 write data is valid When this bit is 1, indicating sdio_cmd53_wr_data is valid

Signal	I/O	Bit Width	Description
sdio_cmd53_wr_data	output	8	SDIO CMD53 write data Whensdio_cmd53_wr_valid is 1, indicating write data is valid
sdio_cmd53_wr_end	output	1	SDIO CMD53 write data completed indication When this bit is 1, it indicates that the write operation has completed and that IP has sent CRC status and End bit to Host.
sdio_cmd53_wr_ok	output	1	SDIO CMD53 write data is correct Synchronize with sdio_cmd53_wr_end 1: the write operation is correct 0: the write operation has CRC16 or End bit or CRC status sending error
sdio_cmd53_wr_abort	output	1	Abort of write operation. This bit automatically changes to 1 when Host aborts this write operation through the CMD52. When the user detects that the bit is 1, the write operation is terminated.
sdio_cmd53_rd_valid	input	1	SDIO CMD53 read data is valid during the read operation, when the user is ready to read data, this bit needs to be pulled up to inform IP that the data is ready. IP starts sending read data in sdio_data.
sdio_cmd53_rd_data	input	8	SDIO CMD53 read data that the user provides to IP during read operation.
sdio_cmd53_rd_ready	output	1	SDIO CMD53 read data ready signal When this bit is 1, it means that IP has obtained sdio_cmd53_rd_data, and the user needs to prepare the next sdio_cmd53_rd_data.
sdio_cmd53_rd_end	output	1	SDIO CMD53 read data completed indication When this bit is 1, it indicates that the read operation has completed and that IP has sent CRC and End bit to Host.
sdio_cmd53_rd_abort	output	1	Abort of read operation This bit automatically changes to 1 when Host aborts this read operation through the CMD52. When the user detects that the bit is 1, the read operation is terminated.
sdio_buffer_full	input	1	Buffer full indication 1: full 0: not full When a write operation is completed (after IP sending CRC status and End bit), if the buffer of user is full and cannot receive the next data, this bit needs to be pulled up. IP pulls down sdio_dat0 to inform the Host of card busy. When the user pulls down this bit, IP pulls up sdio_dat0, and then the next write by Host is possible
SDIO Tuning Interface			
sdio_tuning_start	output	1	SDIO Tuning Block Pattern starts If the IP does not need to send Tuning Block Pattern, the pin can be suspended.
sdio_tuning_data	input	4	SDIO Tuning Block Pattern data signal If the IP does not need to send Tuning Block Pattern, the pin can input constant 0.

Signal	I/O	Bit Width	Description
sdio_tuning_end	input	1	SDIO Tuning Block Pattern ends If the IP does not need to send Tuning Block Pattern, the pin can input constant 0.
2MHz Clock			
clk_2mhz	input	1	2MHz clock input When IP is configured as UHS-I Card and CMD11 command is required to execute switching voltage timing, it is necessary to input a 2MHz clock on this interface to detect whether sdio_clk stops. If the IP is configured as Non UHS Card or the CMD11 command is not required, this pin can input constant 0.

6 Initialization

6.1 IP Initialization Flow

1. Power on the chip.
2. Pull down RSTN to start global reset.
3. Pull up RSTN to complete global reset.
4. The user can configure IP parameters through the CPU Slave Interface, and the following registers can be referred to:
   - Configure Reg0x04 bit22: whether IP is a low speed card
   - Configure Reg0x00 bit24: whether IP supports High Speed
   - Configure Reg0x08 bit23~0: configure CCCR CIS Pointer
   - Configure Reg0x2C bit23~0: configure Function1 CIS Pointer
   - Configure Reg0x34 bit15~0: configure CCCR Max Block Size
   - Configure Reg0x34 bit31~16: configure Function1 Max Block Size
   - Configure Reg0x30 bit0: configure IO Ready = 1
5. When the above configuration is completed, when the IP receives CMD5, it responds 1 in the R4 C field to inform SDIO Host of IO Ready.
6. When the SDIO Host selects IP through CMD7, IP enters into Command State, and Host can read and write IP CIA register through CMD52 and CMD53.
7. SDIO Host configures CCCR IOE1 = 1 through CMD52 to enable Function1. At this time, IP pulls up fun1_ioe.
8. The user starts to initialize Function1.
9. After Function1 initialization, the user pulls up fun1_ior to set CCCR IOR1 = 1.
10. After reading CCCR IOR1 is 1 through CMD52, SDIO Host starts to access Function1 register.
11. If Host wants to read or write in block mode, it is necessary to configure the block size register of the corresponding Function through the CMD52, and the configured value must be less than or equal to the corresponding Max block size in above NO. 4. Function0 Block Size register is CCCR register 10h-11h; The Function1 Block Size register is FBR1 register 110h-111h. Both registers are in small-end mode and are configured from 1 to 2048.
12. If Host needs to receive IP interrupt signal, Host needs to open IP interrupt enable register CCCR 04h through CMD52.
13. Please refer to the SDIO protocol for the detailed description of the above SDIO registers.

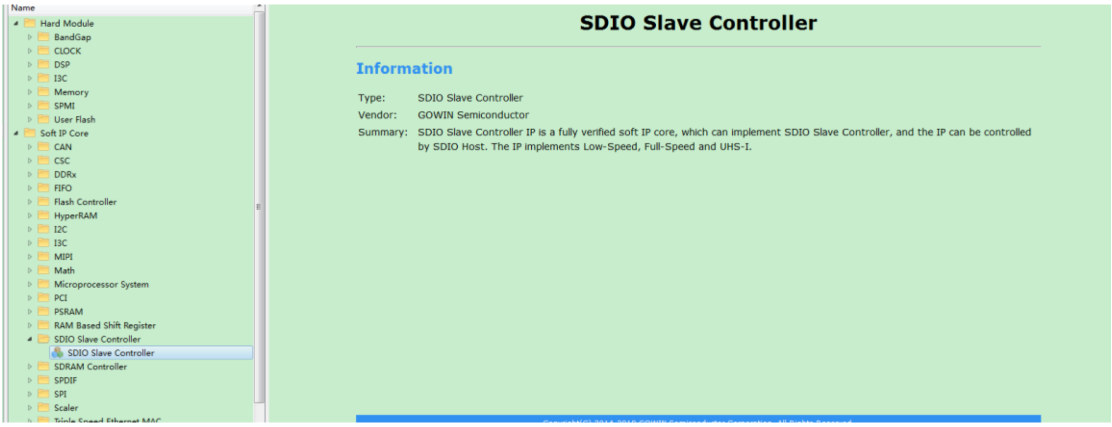
7 Interface Configuration

The user can call and configure SDIO Slave Controller IP by the IP Core Generator tool in the IDE. Taking Non UHS Card as an example, this section introduces the main configuration interface, flow and options.

1. Open the IP Core Generator

After creating a project, you can click the “Tools” tab in the upper left, open the IP Core Generator through the drop-down list and select SDIO Slave Controller, as shown in Figure 7-1.

Figure 7-1 IP Core Generator Tool



SDIO Slave Controller

Information

Type: SDIO Slave Controller

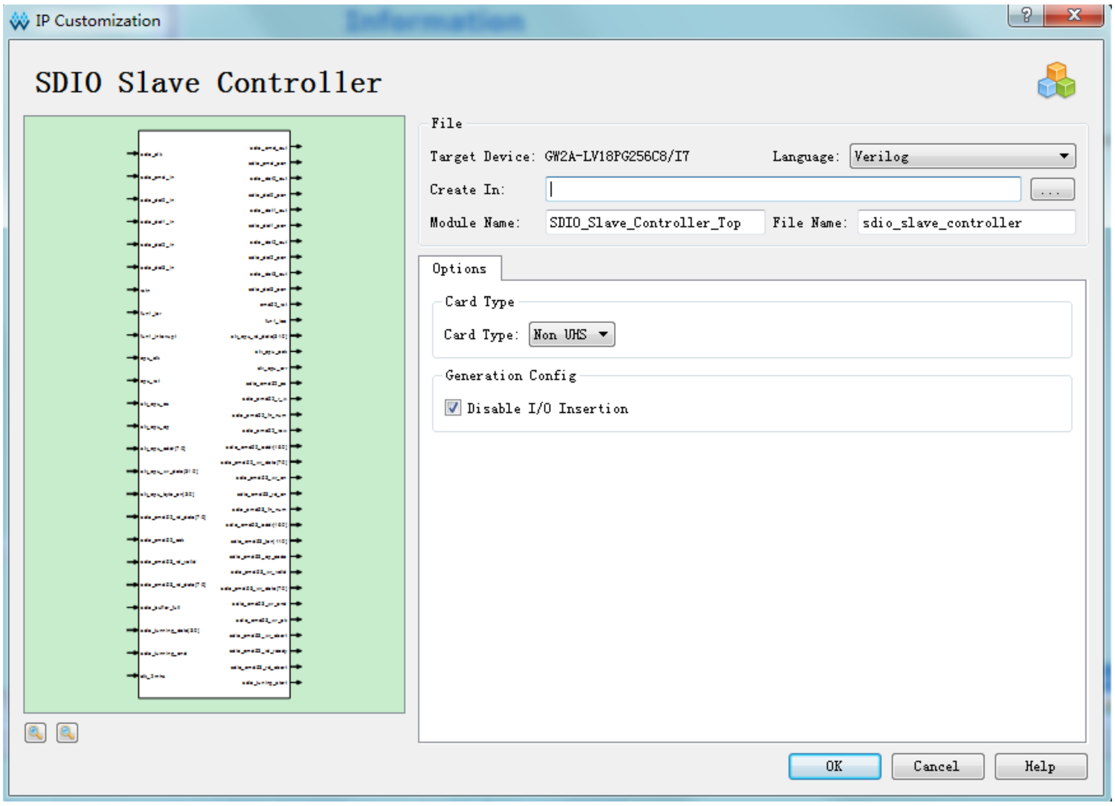
Vendor: GOWIN Semiconductor

Summary: SDIO Slave Controller IP is a fully verified soft IP core, which can implement SDIO Slave Controller, and the IP can be controlled by SDIO Host. The IP implements Low-Speed, Full-Speed and UHS-I.

2. SDIO Slave Controller ports interface

The left of the configuration interface is the SDIO Slave Controller IP interface schematic diagram, and the right is the IP option, as shown in Figure 7-2.

Figure 7-2 Configuration Interface of SDIO Slave Controller IP



SDIO Slave Controller

File

Target Device: GW2A-LV18PG256C8/I7

Language: Verilog

Create In:

Module Name: SDIO_Slave_Controller_Top

File Name: sdio_slave_controller

Options

Card Type

Card Type: Non UHS

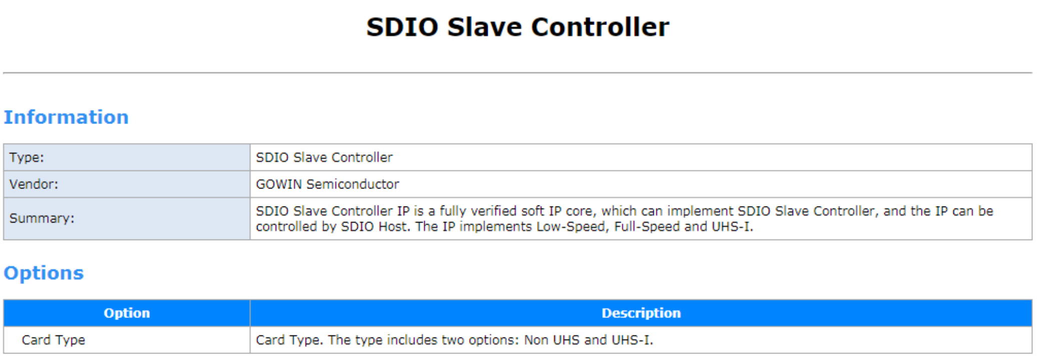
Generation Config

☑ Disable I/O Insertion

3. Open Help document

Users can click the Help button in the lower left corner of Figure 7-2 to learn the simple English introduction of various options, so as to facilitate users to quickly complete the configuration of IP core, as shown in Figure 7-3.

Figure 7-3 Help



SDIO Slave Controller

Information

Type:	SDIO Slave Controller
Vendor:	GOWIN Semiconductor
Summary:	SDIO Slave Controller IP is a fully verified soft IP core, which can implement SDIO Slave Controller, and the IP can be controlled by SDIO Host. The IP implements Low-Speed, Full-Speed and UHS-I.

Options

Option	Description
Card Type	Card Type. The type includes two options: Non UHS and UHS-I.

8 Parameter Configuration

Users configure the various static and timing parameters of Gowin SDIO Slave Controller according to the design requirements as shown in Table 8-1.

Table 8-1 Static and Timing Parameters of Gowin SDIO Slave Controller

Name	Description	Options
Card Type	● Non UHS Card supports low-speed and full-speed. Full-speed supports default-speed and high-speed. ● UHS-I Card supports SDR12, SDR25 and SDR50.	Non UHS/UHS-I

9 Reference Design

Please see the SDIO Slave Controller Reference Design for details at Gowinsemi website.

GOWIN

PROGRAMMING FOR THE FUTURE