



Gowin SDIO SDR104 Slave Controller IP ユーザーガイド

IPUG774-1.0.1J, 2023/11/16

著作権について(2023)

著作権に関する全ての権利は、**Guangdong Gowin Semiconductor Corporation** に留保されています。

GOWIN **高云**、、Gowin、GowinSynthesis、及びGOWINSEMIは、当社により、中国、米国特許商標庁、及びその他の国において登録されています。商標又はサービスマークとして特定されたその他全ての文字やロゴは、それぞれの権利者に帰属しています。何れの団体及び個人も、当社の書面による許可を得ず、本文書の内容の一部もしくは全部を、いかなる視聴覚的、電子的、機械的、複写、録音等の手段によりもしくは形式により、伝搬又は複製をしてはなりません。

免責事項

当社は、GOWINSEMI Terms and Conditions of Sale (GOWINSEMI取引条件)に規定されている内容を除き、(明示的か又は黙示的に拘わらず)いかなる保証もせず、また、知的財産権や材料の使用によりあなたのハードウェア、ソフトウェア、データ、又は財産が被った損害についても責任を負いません。当社は、事前の通知なく、いつでも本文書の内容を変更することができます。本文書を参照する何れの団体及び個人も、最新の文書やエラッタ(不具合情報)については、当社に問い合わせる必要があります。

バージョン履歴

日付	バージョン	説明
2021/05/07	1.0J	初版。
2023/11/16	1.0.1J	● 表 4-2 CPU Slave Interface レジスタの定義を更新。 ● スクリーンショットを更新。

目次

目次	i
図一覧.....	iii
表一覧.....	iv
1 本マニュアルについて	1
1.1 マニュアルの内容	1
1.2 関連ドキュメント	1
1.3 用語、略語	1
1.4 テクニカル・サポートとフィードバック	2
2 概要.....	3
3 特徴及び性能	4
3.1 主な特徴.....	4
3.2 動作周波数.....	5
3.3 リソース使用量.....	5
4 機能の説明.....	6
4.1 IP の構造.....	6
4.2 Gowin SDIO インターフェースの定義	7
4.3 Non UHS Card と UHS-I Card.....	8
4.3.1 Non UHS Card.....	8
4.3.2 UHS-I Card	8
4.4 Gowin SDIO IP の初期化	9
4.4.1 Non UHS Card の初期化プロセス	10
4.4.2 UHS-I Card の初期化プロセス	11
4.5 CIA(Function0)レジスタ	11
4.6 CMD52.....	12
4.7 CMD53.....	12
4.8 Abort	12
4.9 CMD19.....	13
4.10 割り込み.....	13
4.10.1 割り込みの生成.....	13
4.10.2 割り込みのクリア	14
4.11 ユーザーインターフェース	14

4.11.1 CPU Slave Interface	14
4.11.2 SDIO CMD52 Interface	22
4.11.3 SDIO CMD53 Interface	25
4.11.4 SDIO Tuning Interface	31
4.12 IP の初期化プロセス	32
5 ポート一覧	34
6 パラメータの構成	40
7 GUI での構成	41
8 リファレンス・デザイン	43

図一覧

図 4-1 Gowin SDIO SDR104 Slave Controller IP の構造.....	6
図 4-2 SDIO ピンの接続図	8
図 4-3 Non UHS Card の初期化プロセス	10
図 4-4 UHS-I Card の初期化プロセス	11
図 4-5 Tuning Block Pattern	13
図 4-6 CPU Slave Interface の書き込みタイミング.....	20
図 4-7 CPU Slave Interface の読み出しタイミング.....	21
図 4-8 CPU Slave Interface の交互の読み出しと書き込みタイミング	21
図 4-9 SDIO CMD52 Interface の書き込み(RAW flag = 1)タイミング	23
図 4-10 SDIO CMD52 Interface の書き込み(RAW flag = 0)のタイミング	24
図 4-11 SDIO CMD52 Interface の読み出しタイミング	25
図 4-12 SDIO CMD53 Interface の 4 ビットモードの書き込みのタイミング	27
図 4-13 SDIO SDR104 CMD53 Interface の 4 ビットモードの読み出し	28
図 4-14 Write Bad Abort	28
図 4-15 Write Good Abort Case1	29
図 4-16 Write Good Abort Case2	29
図 4-17 Read Bad Abort.....	30
図 4-18 Read Good Abort Case1	30
図 4-19 Read Good Abort Case2	31
図 4-20 SDIO Tuning Interface のタイミング	31
図 7-1 IP Core Generator	41
図 7-2 SDIO SDR104 Slave Controller IP の構成画面.....	42

表一覧

表 1-1 用語、略語	1
表 2-1 Gowin SDIO SDR104 Slave Controller IP	3
表 3-1 リソース使用量	5
表 4-1 SDIO SDR104 Slave Controller IP のインターフェース定義と SD 標準インターフェース定義の対応関係	7
表 4-2 CPU Slave Interface レジスタの定義	14
表 5-1 Gowin SDIO SDR104 Slave Controller IP の IO ポート	34
表 6-1 Gowin SDIO SDR104 Slave Controller の静的パラメータとタイミングパラメータ	40

1 本マニュアルについて

1.1 マニュアルの内容

本マニュアルは、ユーザーが Gowin SDIO SDR104 Slave Controller IP を使いこなせるよう、その機能、ポート、呼び出しと構成、およびリファレンス・デザインなどについて説明します。本マニュアルに記載のソフトウェア GUI のスクリーンショットは、Gowin ソフトウェア 1.9.9 Beta-6 バージョンの場合のものです。ソフトウェアのアップデートにより、一部の内容が変更される場合があります。

1.2 関連ドキュメント

GOWIN セミコンダクターのホームページ www.gowinsemi.com/ja から、以下の関連ドキュメントがダウンロード、参考できます：

- GW1N シリーズ FPGA 製品データシート([DS100](#))
- GW1NR シリーズ FPGA 製品データシート([DS117](#))
- GW1NS シリーズ FPGA 製品データシート([DS821](#))
- GW1NSR シリーズ FPGA 製品データシート([DS861](#))
- GW1NSE シリーズ安全 FPGA 製品データシート([DS891](#))
- GW1NSER シリーズ安全 FPGA 製品データシート([DS881](#))
- GW2A シリーズ FPGA 製品データシート([DS102](#))
- GW2AR シリーズ FPGA 製品データシート([DS226](#))
- Gowin ソフトウェア ユーザーガイド([SUG100](#))

1.3 用語、略語

表 1-1 に、本マニュアルで使用される用語、略語、及びその意味を示します。

表 1-1 用語、略語

用語、略語	正式名称	意味
-------	------	----

用語、略語	正式名称	意味
CSA	Code Storage Area	コード・ストレージ・エリア
DDR	Double Data Rate	ダブル・データレート
IP	Intellectual Property	設計資産
LUT	Look-up Tables	ルックアップテーブル
SDIO	Secure Digital Input and Output	セキュアデジタル入出力
SDR	Single Data Rate	シングル・データレート
UHS	Ultra High Speed	超高速

1.4 テクニカル・サポートとフィードバック

GOWIN セミコンダクターは、包括的な技術サポートをご提供しています。使用に関するご質問、ご意見については、直接弊社までお問い合わせください。

ホームページ : www.gowinsemi.com/ja

E-mail : support@gowinsemi.com

2 概要

このマニュアルでは、ユーザーに SDR104 モード対応 SDIO Slave の汎用インターフェースを提供する Gowin SDIO SDR104 Slave Controller IP について説明します。この IP を SDIO Device デザインに統合することにより、ユーザーは、SDIO Device 機能を容易に実装することができます。SDIO Slave を必要とするデバイスに統合することにより、さまざまなアプリケーションに使用できます。

表 2-1 Gowin SDIO SDR104 Slave Controller IP

Gowin SDIO SDR104 Slave Controller IP	
ロジックリソース	表 3-1 参照。
提供されるファイル	
設計ファイル	Verilog (暗号化)
リファレンス・デザイン	Verilog
テストベンチ	Verilog
テストおよび設計フロー	
合成ソフトウェア	GowinSynthesis®
統合開発環境	Gowin ソフトウェア(V1.9.7.03Beta 以降)

3 特徴及び性能

3.1 主な特徴

- SD Physical Layer Specification Version 4.00 をサポート。
- SDIO Specification Version 4.00 をサポート。
- 4 ビットの SD 転送モードをサポート。
- SPI 転送モードをサポートしない。
- UHS-I モード SDR104 速度をサポート。
- UHS-I モードは 1.8V 電圧をサポートし、電圧スイッチングをサポートしない。
- SDIO Command(CMD0, CMD3, CMD5, CMD7, CMD11, CMD15, CMD19, CMD52, CMD53)をサポート。
- CRC7 および CRC16 の自動生成とチェックをサポート。
- Function0 と Function1 をサポート。
- ユーザー側の CPU Slave Interface を使用した IP の構成をサポート。
- ユーザー側の CMD52 と CMD53 インターフェースをサポート。
- 4 ビットモードの割り込みをサポート(Data Block Gap および Asynchronous Interrupt はサポートしない)。
- 読み出しと書き込みの Abort をサポート。
- Suspend/Resume をサポートしない。
- Read Wait をサポートしない。
- Combo Card をサポートしない。
- CSA をサポートしない。
- DDR モードをサポートしない。
- Card Detection をサポートしない。

3.2 動作周波数

Gowin SDIO SDR104 Slave Controller IP の動作周波数は、SDIO Host が提供するクロック周波数と、選択したチップで IP がサポートする最大動作周波数によって決まります。

3.3 リソース使用量

Gowin SDIO SDR104 Slave Controller IP を Verilog で実装します。リソース使用量は、モードによって異なります。例えば、GW2A-18 FPGA の場合、そのリソース使用量は表 3-1 に示すようになります。他のデバイスでのリソース使用量については、その最新情報を参照してください。

表 3-1 リソース使用量

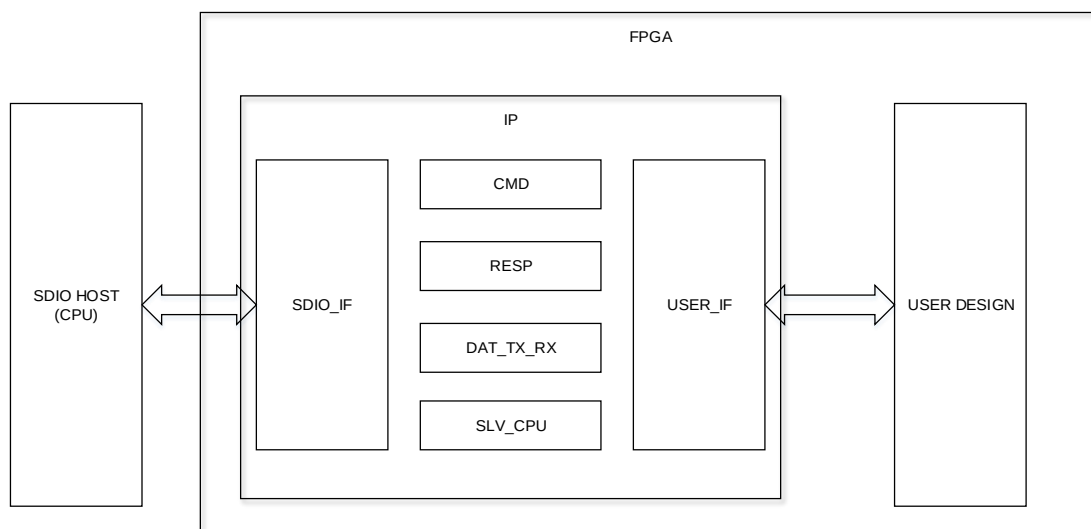
Type	LUTs	Registers	Device	Speed Grade
Non UHS	3193	1667	GW2A-18	-8
UHS-I	3043	1695	GW2A-18	-8

4機能の説明

4.1 IP の構造

図 4-1 に示すように、Gowin SDIO SDR104 Slave Controller IP には、SDIO_IF、CMD、RESP、DAT_TX_RX、SLV_CPU、USER_IF などのモジュールが含まれています。User Design は FPGA のユーザーデザインであり、SDIO HOST は SDIO ホストです。

図 4-1 Gowin SDIO SDR104 Slave Controller IP の構造



- SDIO_IF モジュールは、SDIO インターフェースのデータ選択と双方向転送を実現します。
- CMD モジュールは、SDIO Command の解析を実現します。
- RESP モジュールは、SDIO Response の生成を実現します。
- DAT_TX_RX モジュールは、CMD19 および CMD53 の読み出しと書き込みを実現します。
- SLV_CPU モジュールは、ユーザーによる IP 機能の構成を実現します。
- USE_IF モジュールは、ユーザー側と IP 間のデータ交換を実現します。

4.2 Gowin SDIO インターフェースの定義

Gowin SDIO SDR104 Slave Controller IP のインターフェース定義と SD 標準インターフェース定義の対応関係を表 4-1 に示します。

表 4-1 SDIO SDR104 Slave Controller IP のインターフェース定義と SD 標準インターフェース定義の対応関係

Pin	SD 標準				IP
	SD 4 ビットモード		SD 1 ビットモード		
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

ユーザーは、次のステートメントを使用して、IP で定義された SDIO インターフェースを入力および出力形式に変換できます。

```

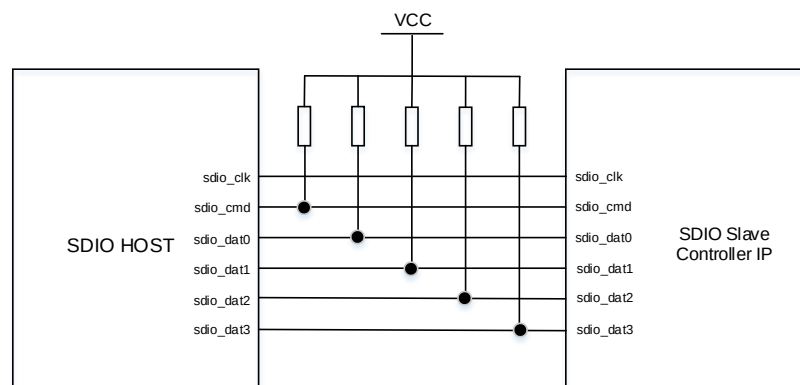
assign sdio_cmd_in = sdio_cmd;
assign sdio_cmd = (!sdio_cmd_oen) ? sdio_cmd_out : 1'bz;
assign sdio_dat0_in = sdio_dat0;
assign sdio_dat0 = (!sdio_dat0_oen) ? sdio_dat0_out : 1'bz;
assign sdio_dat1_in = sdio_dat1;
assign sdio_dat1 = (!sdio_dat1_oen) ? sdio_dat1_out : 1'bz;
assign sdio_dat2_in = sdio_dat2;
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。

```

回路の接続

回路の接続を行う際、CMD および DAT ラインをプルアップ抵抗を介して VCC に接続する必要があります。IP が Non UHS Card に設定されている場合、VCC は 3.3V、IP が UHS-I Card に設定されている場合、VCC は 1.8V です。そのリファレンス回路を図 4-2 に示します。

図 4-2 SDIO ピンの接続図



4.3 Non UHS Card と UHS-I Card

IP は、Non UHS Card と UHS-I Card の 2 種類の Card をサポートしています。ユーザーは IP を生成する時、GUI でこれを構成する必要があります。

4.3.1 Non UHS Card

IP が Non UHS Card で動作する場合、バンク電圧と SDIO インターフェースのプルアップ抵抗の電圧を 3.3V にする必要があります。

Non UHS Card は Low-Speed および Full-Speed をサポートします。そのうち Full-Speed には Default-Speed と High-Speed が含まれます。

ユーザーが IP を Non UHS Card として構成した後、初期化の際は、IP が受信した CMD5 S18R が 0 か 1 かにかかわらず、IP 応答の R4 での S18A は常に 0 です。IP が Non UHS Card で動作する場合、IP は CMD11 および CMD19 をサポートしません。

4.3.2 UHS-I Card

IP が UHS-I Card で動作する場合、バンク電圧と SDIO インターフェースのプルアップ抵抗の電圧を 1.8V にする必要があります。

UHS-I Card は、SDR12、SDR25、SDR50、および SDR50 をサポートしています。

ユーザーが IP を UHS-I Card として構成した後、初期化の際は、IP が受信した CMD5 S18R が 0 の場合、IP 応答の R4 での S18A は 0 で、IP が受信した CMD5 S18R が 1 の場合、IP 応答の R4 での S18A は 1 です。IP が UHS-I Card で動作する場合、IP は CMD11 および CMD19 をサポートします。IP が CMD11 を受信すると、R1 で応答します。また、Voltage Switch Sequence をサポートしていますが、IO 電圧が変更されることはあ

りません。

4.4 Gowin SDIO IP の初期化

IP は電源投入後、正常に動作する前に、SDIO Host によって初期化される必要があります。初期化プロセスは、SDIO 規格に準拠します。初期化クロックの周波数範囲は 100KHz～400KHz です。

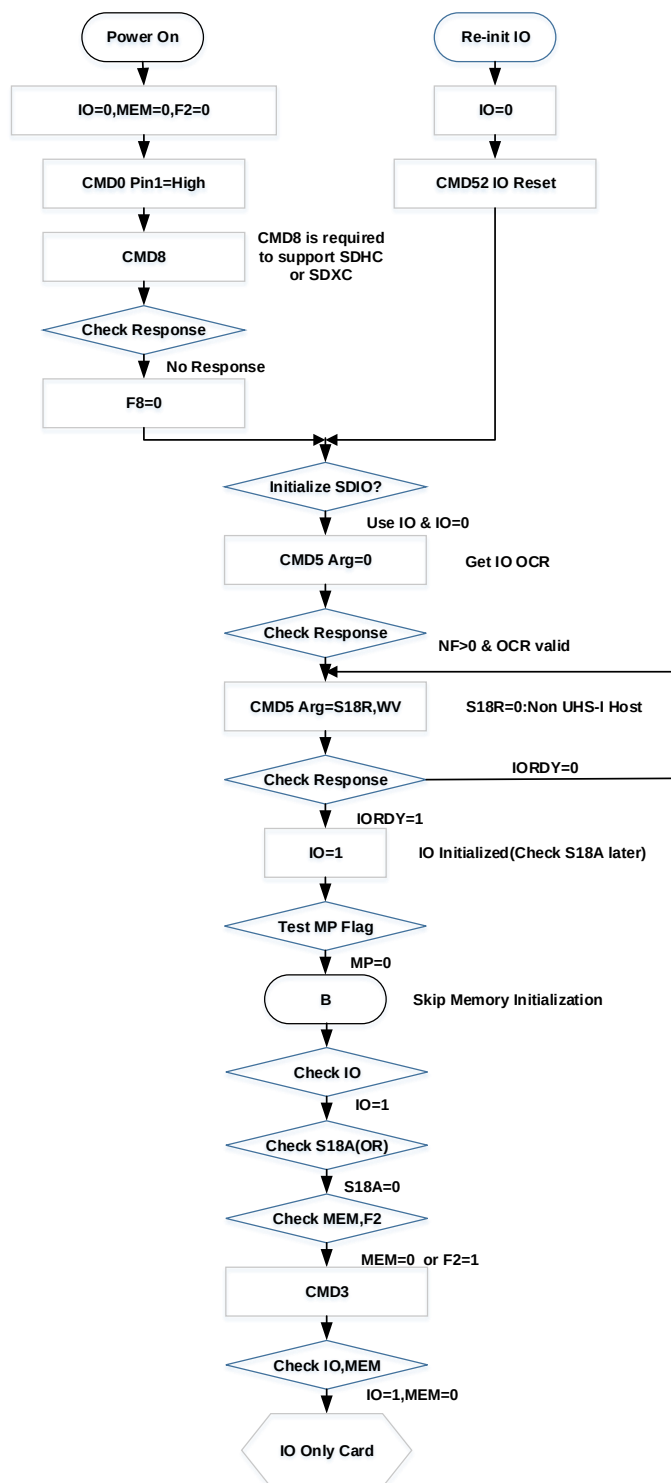
この IP は SPI モードをサポートしていないため、初期化中に CMD0 を受信したときに、sdio_dat3 が High かどうかに関係なく IP は SD mode のままです。

IP が UHS-I Card に構成されている場合、IP は CMD11 コマンドを受信して Voltage Switch Sequence に入ることができますが、IO 電圧が変更されることはありません。Voltage Switch Sequence が終了すると、IP は UHS-I 状態になります。

4.4.1 Non UHS Card の初期化プロセス

Non UHS Card の初期化プロセスを図 4-3 に示します。

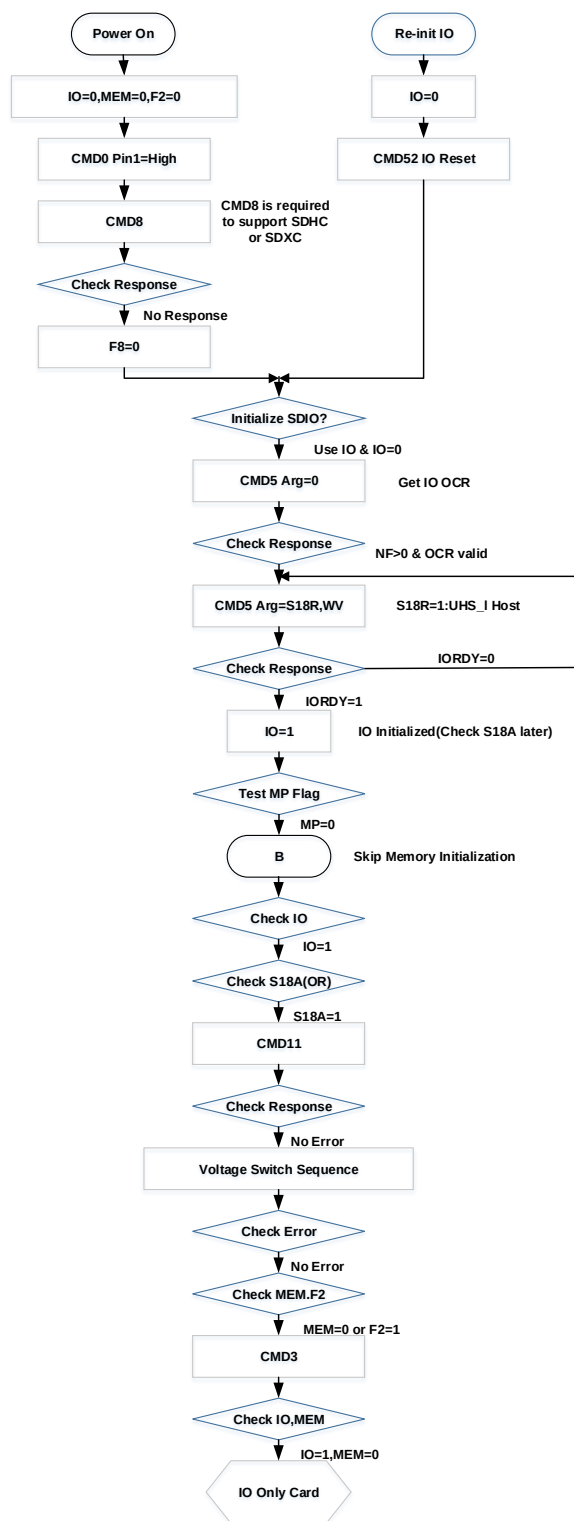
図 4-3 Non UHS Card の初期化プロセス



4.4.2 UHS-I Card の初期化プロセス

UHS-I Card の初期化プロセスを図 4-4 に示します。

図 4-4 UHS-I Card の初期化プロセス



4.5 CIA(Function0)レジスタ

CIA レジスタは SDIO 規格により定義されます。CIA レジスタは、Host

が **Slave** 状態を読み出し、**Slave** 機能を構成することをサポートしています。**Host** は、**CMD52** および **CMD53** コマンドを使用して **CIA** レジスタを読み出しと書き込みできます。

CIA レジスタには、**CCCR** レジスタ、**FBR** レジスタ、**CIS** レジスタ、および **CSA** レジスタが含まれます。

CCCR レジスタと **FBR(Function1)** レジスタは **IP** に実装されているため、初期化の際は、ユーザーは **CPU Slave Interface** を介して対応するレジスタにアクセスできます。**Host** がこれらのレジスタを読み出しと書き込みするとき、**IP** は自動的に応答します。

FBR(Function2～7) レジスタは現在のところ、アクセスできません。

CIS レジスタは **IP** に実装されていません。**Host** がこれらのレジスタにアクセスする場合、ユーザーは **SDIO CMD52 Interface** または **SDIO CMD53 Interface** でデータを提供する必要があります。

CSA レジスタは現在のところ、アクセスできません。

4.6 CMD52

IP が **Command State** に入った後、**Host** は **CMD52** を介してバイトで **IP** レジスタにアクセスできます。**IP** は **Function0** と **Function1** をサポートしているため、**IP** は、**Host** の、**Function0** と **Function1** の **CMD52** にのみ応答します。

Host が **CMD52** を介して **CCCR** および **FBR** レジスタにアクセスするとき、**IP** は自動的に応答します。

Host が **CMD52** を介して **CIS** および **Function1** レジスタにアクセスする場合、ユーザーは **SDIO CMD52 Interface** を介してデータの送受信に参加する必要があります。

4.7 CMD53

IP が **Command State** に入った後、**Host** は **CMD53** コマンドを介して **IP** レジスタに継続的にアクセスできます。**IP** は **Function0** と **Function1** をサポートしているため、**IP** は、**Host** の、**Function0** と **Function1** の **CMD53** にのみ応答します。

Host が **CMD53** を介して **CCCR** および **FBR** レジスタにアクセスするとき、**IP** は自動的に応答します。

Host が **CMD53** を介して **CIS** および **Function1** レジスタにアクセスする場合、ユーザーは **SDIO CMD53 Interface** を介してデータの送受信に参加する必要があります。

4.8 Abort

Host が **CMD53** を送信して **Infinite Block Mode** でデータを読み出しと書き込みする場合、**CMD52** を介して **Abort** コマンドを送信し、読み出しと

書き込みを中断できます。

4.9 CMD19

IP が UHS-I Card に構成されている場合は、CMD19 をサポートします。正しい CMD19 を受信すると、IP は自動的に R1 に応答します。ユーザーは、SDIO Tuning Interface を介して Tuning Block Pattern を提供する必要があります。それに対して、IP が Non UHS Card に構成されている場合、SDIO Tuning Interface を介して Tuning Block Pattern を提供する必要はありません。図 4-5 に示すように、Tuning Block Pattern は合計 144 ニブルです。

図 4-5 Tuning Block Pattern

```
FF0FFF00_FFCCC3CC_C33CCCFE_FFFFFFFF
FFDEFFDD_FFFBFFFB_BFFF7FFF_77F7BDEF
FFF0FFF0_0FFCCC3C_CC33CCCF_FFEFFFEF
FFFDFFFD_DFFBFFF_BBFFF7FF_F77F7BDE
F9503A4B_C5488FBC
```

ユーザーは左から右、上から下の順に送信する必要があります。

4.10 割り込み

IP は、4 ビットモードの割り込みをサポートしています。

4.10.1 割り込みの生成

4 ビットモードでは、DAT1 はデータおよび割り込み多重化信号ラインとして使用されます。ユーザーが fun1_interrupt ピンを High にプルアップすると、IP は標準の Interrupt Period で DAT1 を Low にプルダウンして、割り込みを生成します。4 ビットモードの割り込みは Data Block Gap および Asynchronous Interrupt をサポートしません。4 ビットモードの割り込み信号は、sdio_clk と同期するため、sdio_clk が停止すると、IP は DAT1 で割り込み信号を生成できません。IP が 4 ビットモードで動作する場合、SDIO Host デザインでクロックを停止する必要があり、クロックを起動するために割り込みが必要な場合、次の方法で実現できます。

1. Host が CMD52 を使用して IP を 1 ビットモードに設定します。
2. Host がクロックを停止します。
3. ユーザーが fun1_interrupt ピンを 1 にプルアップして、1 ビットモードで割り込みを生成します。
4. Host がクロックの提供を開始します。
5. Host が CMD52 を使用して IP を 4 ビットモードに設定します。
6. Host が IP へのアクセスを開始します。

4.10.2 割り込みのクリア

SDIO Host が割り込みクリアレジスタにアクセスしていることが検出された場合、ユーザーは **fun1_interrupt** ピンを 0 にプルダウンして、割り込みをクリアする必要があります。割り込みクリアレジスタはユーザーが定義します。

SDIO 規格による推奨割り込みクリア時間：

- ユーザーが **CMD52** アクセスを介して割り込みをクリアする場合、**R5** が終了する前に割り込みをクリアする必要があります。
- ユーザーが **CMD53** アクセスを介して割り込みをクリアする場合、データの送信を完了する前に割り込みをクリアする必要があります。

4.11 ユーザーインターフェース

4.11.1 CPU Slave Interface

CPU Slave Interface のクロック

CPU Slave Interface のクロックである **cpu_clk** は、**sdio_clk** には依存していません。通常の使用では、**cpu_clk** を **sdio_clk** に接続することで、2 つのインターフェースのクロックを同期させることができます。

CPU Slave Interface の機能

CCCR レジスタと FBR(Function1)レジスタは IP に実装されているため、ユーザーは CPU スレーブインターフェースを介して対応するレジスタにアクセスできます。Host がこれらのレジスタにアクセスする際、IP は自動的に応答します。さらに、ユーザーは CPU Slave Interface を介して IP の他の機能を設定できます。レジスタの定義を表 4-2 に示します。

CPU Slave Interface のデータインターフェースは 32 ビットで、アドレスは 4 バイトに揃えられます。

CPU Slave Interface レジスタの定義

CPU Slave Interface レジスタの定義を表 4-2 に示します。

表 4-2 CPU Slave Interface レジスタの定義

レジスタの アドレス	Bit	名称	デフォ ルト値	タイ プ	説明
0x00	3:0	CCCR/FBR Format Version number	0x3	RO	CCCR/FBR バージョン CCCR Reg0x0 bit 3~bit 0 に対応
	7:4	SDIO Specification Revision Number	0x5	RO	SDIO バージョン CCCR Reg0x0 bit 7~bit 4 に対応
	11:8	SD Format Version Number	0x4	RO	SD バージョン CCCR Reg0x1 bit 3~bit 0 に対応
	16	Support Master Power Control	0x0	RW	マスター電力制御 CCCR Reg0x12 bit0 に対応

レジスタの アドレス	Bit	名称	デフォ ルト値	タイ プ	説明
	17	Enable Master Power Control	0x0	RO	マスター電力制御イネーブル CCCR Reg0x12 bit 1 に対応
	20:18	Total Card Power	0x0	RO	カードの総電力 CCCR Reg0x12 bit 4~bit 2 に対応
	24	Support High-Speed	0x1	RW	高速モードをサポート CCCR Reg0x13 bit0 に対応
	27:25	Bus Speed Select	0x0	RO	バス速度モードの選択 CCCR Reg0x13 bit 3~bit 1 に対応
0x04	0	Interrupt Enable Master	0x0	RO	マスター割り込みイネーブル CCCR Reg0x4 bit0 に対応
	7:1	Interrupt Enable for Function x	0x00	RO	Function x 割り込みイネーブル CCCR Reg0x4 bit 7~bit 1 に対応
	9:8	Bus Width	0x0	RO	バス幅 CCCR Reg0x7 bit1~bit0 に対応
	10	Support 8-bit Bus Mode	0x0	RO	8bit バスモードをサポート CCCR Reg0x7 bit 2 に対応
	13	Enable Continuous SPI Interrupt	0x0	RO	連続 SPI 割り込みイネーブル CCCR Reg0x7 bit 5 に対応
	14	Support Continuous SPI interrupt	0x1	RW	連続 SPI 割り込みをサポート CCCR Reg0x7 bit6 に対応
	15	Card Detect Disable	0x0	RO	カード検出を無効にする CCCR Reg0x7 bit 7 に対応
	16	Support Direct Command (CMD52)	0x1	RO	Direct コマンド(CMD52)をサポート CCCR Reg0x8 bit 0 に対応
	17	Support Multiple Block Transfer	0x1	RO	複数ブロック転送をサポート CCCR Reg0x8 bit 1 に対応
	18	Support Read Wait	0x0	RO	読み出し待機をサポート CCCR Reg0x8 bit 2 に対応
	19	Support Bus Control	0x0	RO	バス制御をサポート CCCR Reg0x8 bit 3 に対応
	20	Support Block Gap Interrupt	0x0	RO	ブロックギャップ割り込みをサポート CCCR Reg0x8 bit 4 に対応
	21	Enable Block Gap Interrupt	0x0	RO	ブロックギャップ割り込みをイネーブル CCCR Reg0x8 bit 5 に対応
	22	Low-Speed Card	0x0	RW	低速カード CCCR Reg0x8 bit 6 に対応
	23	4-bit Mode Support for Low-Speed Card	0x0	RW	低速カードが 4bit モードをサポート CCCR Reg0x8 bit 7 に対応

レジスタの アドレス	Bit	名称	デフォ ルト値	タイ プ	説明
	24	Support SDR50	0x0	RW	SDR50 をサポート CCCR Reg0x14 bit 0 に対応
	25	Support SDR104	0x1	RW	SDR104 をサポート CCCR Reg0x14 bit 1 に対応
	26	Support DDR50	0x0	RW	DDR50 をサポート CCCR Reg0x14 bit 2 に対応
	27	Support Asynchronous Interrupt	0x0	RO	非同期割り込みをサポート CCCR Reg0x16 bit 0 に対応
	28	Enable Asynchronous Interrupt	0x0	RO	非同期割り込みをイネーブル CCCR Reg0x16 bit 1 に対応
0x08	23:0	Common CIS Pointer	0x0010 00	RW	Common CIS ポインター [23:16]は CCCR Reg0x0B bit 7~bit 0 に対応 [15:8]は CCCR Reg0x0A bit 7~bit 0 に 対応 [7:0]は CCCR Reg0x09 bit 7~bit 0 に対 応
	24	Support Driver Type A	0x0	RW	ドライバータイプ A をサポート CCCR Reg0x15 bit 0 に対応
	25	Support Driver Type C	0x0	RW	ドライバータイプ C をサポート CCCR Reg0x15 bit 1 に対応
	26	Support Driver Type D	0x0	RW	ドライバータイプ D をサポート CCCR Reg0x15 bit 2 に対応
	29:28	Driver Type Select	0x0	RO	ドライバータイプの選択 CCCR Reg0x15 bit 5~bit 4 に対応
0x0C	15:0	FN0 Block Size	0x0000	RO	Function 0 ブロックのサイズ [15:8]は CCCR Reg0x11 bit 7~bit 0 に 対応 [7:0]は CCCR Reg0x10 bit 7~bit 0 に対 応
0x10	7:0	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF0 bit 7~bit 0 に対応
	15:8	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF1 bit 7~bit 0 に対応
	23:16	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF2 bit 7~bit 0 に対応
	31:24	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF3 bit 7~bit 0 に対応
0x14	7:0	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF4 bit 7~bit 0 に対応

レジスタの アドレス	Bit	名称	デフォ ルト値	タイ プ	説明
	15:8	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF5 bit 7~bit 0 に対応
	23:16	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF6 bit 7~bit 0 に対応
	31:24	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF7 bit 7~bit 0 に対応
0x18	7:0	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF8 bit 7~bit 0 に対応
	15:8	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xF9 bit 7~bit 0 に対応
	23:16	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFA bit 7~bit 0 に対応
	31:24	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFB bit 7~bit 0 に対応
0x1C	7:0	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFC bit 7~bit 0 に対応
	15:8	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFD bit 7~bit 0 に対応
	23:16	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFE bit7~bit0 に対応
	31:24	Reserved for Vendors	0x00	RO	ベンダーのために予約 CCCR Reg0xFF bit7~bit0 に対応
0x20	3:0	Standard SDIO Function Interface Code	0xF	RW	標準 SDIO 機能インターフェースコード FBR1 Reg0x100 bit3~bit0 に対応
	15:8	Extended Standard SDIO Function Interface Code	0x00	RW	拡張標準 SDIO 機能インターフェース コード FBR1 Reg0x101 bit7~bit0 に対応
	23:16	iSDIO Type Support Code	0x00	RW	サポートされる iSDIO タイプのコード FBR1 Reg0x108 bit7~bit0 に対応
	31:24	Standard iSDIO Function Interface Code	0x00	RW	標準 iSDIO 機能インターフェースコード FBR1 Reg0x103 bit7~bit0 に対応
0x24	15:0	Function 1 I/O Block Size	0x0000	RO	Function 1 ブロックのサイズ [15:8]が FBR1 Reg0x111 bit7~bit0 に対 応 [7:0]が FBR1 Reg0x110 bit7~bit0 に対 応
	16	Support Power Selection	0x0	RW	電力選択をサポート FBR1 Reg0x102 bit0 に対応

レジスタの アドレス	Bit	名称	デフ ォルト値	タイ プ	説明
	20:17	Power State	0x0	RO	電力状態 FBR1 Reg0x102 bit7~bit4 に対応
	21	Enable Power Selection	0x0	RO	電力選択イネーブル FBR1 Reg0x102 bit1 に対応
	24	Function1 SupportsCSA	0x0	RO	CSA をサポート FBR1 Reg0x100 bit6 に対応
	25	Function1 CSAEnable	0x0	RO	CSA イネーブル FBR1 Reg0x100 bit7 に対応
0x28	15:0	Function1 MID_CARD Manufacturer Information	0x0000	RW	Function1 MID_CARD メーカー情報 [15:8]が FBR1 Reg0x107 bit7~bit0 に対応 [7:0]が FBR1 Reg0x106 bit7~bit0 に対応
	31:16	Function1 SDA_MID_MANF SDIO Card Manufacturer Code	0x0000	RW	Function1 SDA_MID_MANF SDIO カードメーカーのコード [31:24]が FBR1 Reg0x105 bit7~bit0 に対応 [23:16]が FBR1 Reg0x104 bit7~bit0 に対応
0x2C	23:0	Address pointer to Function1 CIS	0x002000	RW	Function1 CIS のアドレスポインター [23:16]が FBR1 Reg0x10B bit7~bit0 に対応 [15:8]が FBR1 Reg0x10A bit7~bit0 に対応 [7:0]が FBR1 Reg0x109 bit7~bit0 に対応
0x30	0	IO_Ready	0x0	RW	SDIO Slave 準備完了フラグ。 このビットは、R4 の C フィールドに対応。電源投入後、このビットは 0。ユーザーが CPU Slave Interface を介してレジスタを初期化した後、このビットを 1 に設定。
	18:16	SDIO Bus State	0x0	RO	SDIO Bus State。ユーザーは、このレジスタを介して IP Bus State を読み出し可能。 0x0 : Idle State。IP が CMD5 を介して Host とのネゴシエーションを完了していないことを示す。 0x1 : Initialization State。IP が CMD5 を介して Host とのネゴシエーションを完了したことを示す。 0x2 : Standby State 0x3 : Command State

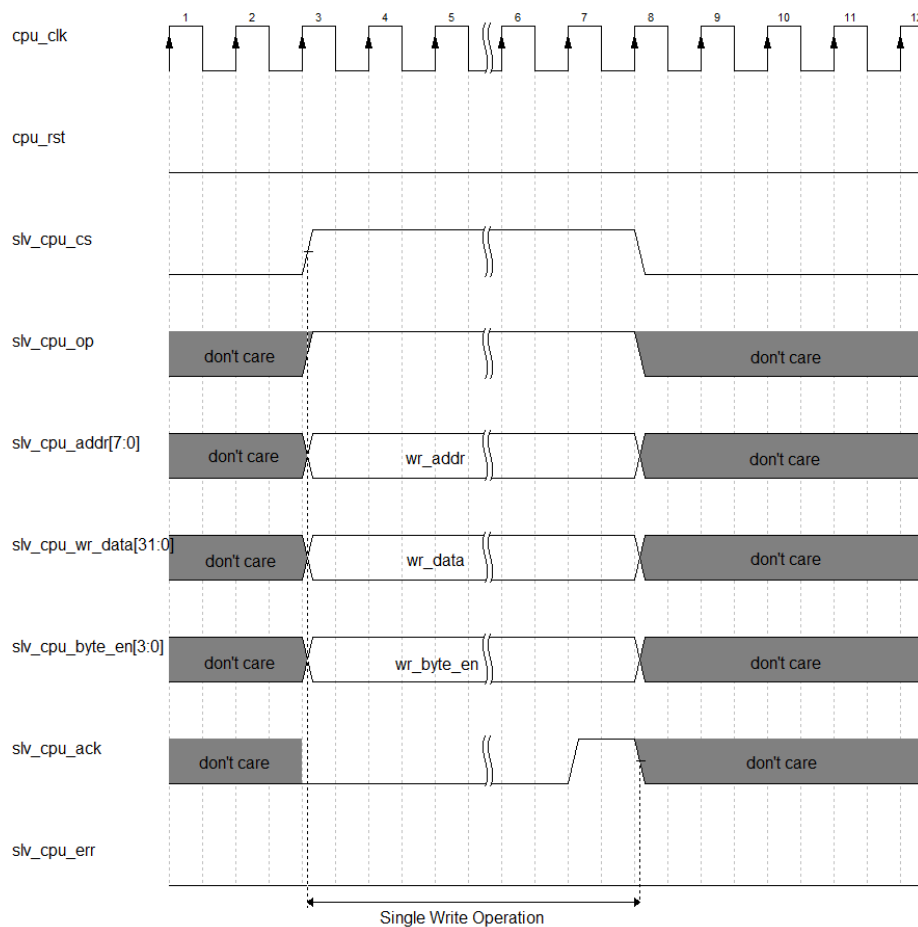
レジスタの アドレス	Bit	名称	デフォ ルト値	タイ プ	説明
					0x4 : Transfer State 0x5 : Inactive State その他 : 定義されていません。
	24	Manual TX Clock to DataPhase	0x0	RW	SDIO 送信データおよび SDIO クロ ック位相は、Manual TX Clock to Data Phase Enable レジスタが有効な場合 に有効。 0 : SDIO クロックの立ち下がりエッジ でデータを送信。 1 : SDIO クロックの立ち上がりエッジ でデータを送信。
	25	Manual TX Clock to Data Phase Enable	0x0	RW	SDIO 送信データと SDIO クロック位 相を手動でイネーブル。 1 : 有効。Manual TX Clock to Data Phase の値に従って構成。 0 : 無効。SDIO 規格に従って自動構成。
0x34	15:0	CCCR Max Block Size	0x0800	RW	CCCR の最大 Block Size
	31:16	FBR1 Max Block Size	0x0800	RW	Function1 の最大 Block Size

CPU Slave Interface の操作タイミング

- CPU Slave Interface の書き込み。

ユーザーが CPU Slave Interface を介してレジスタに書き込む場合、インターフェースを選択するために `slv_cpu_cs` を High にプルアップすると同時に、書き込みを示すために `slv_cpu_op` を High にプルアップする必要があります。ユーザーは、信号 `slv_cpu_addr`、`slv_cpu_wr_data`、および `slv_cpu_byte_en` を介して、レジスタアドレス、書き込みデータ、およびバイトイネーブルを送信します。IP がこの書き込みを完了すると、`slv_cpu_ack` 信号が High にプルアップされます。`slv_cpu_cs`、`slv_cpu_op`、`slv_cpu_addr`、`slv_cpu_wr_data`、および `slv_cpu_byte_en` 信号は、`slv_cpu_ack` 信号が High にプルアップされるまで変更されないようにする必要があります。CPU Slave Interface の書き込みタイミングを図 4-6 に示します。

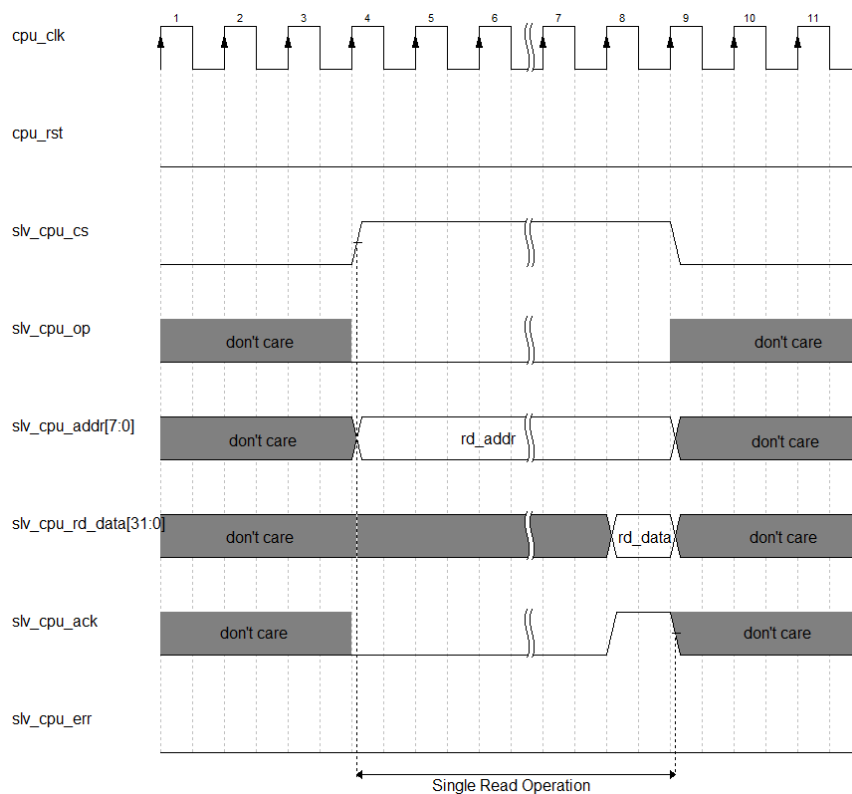
図 4-6 CPU Slave Interface の書き込みタイミング



● CPU Slave Interface の読み出し

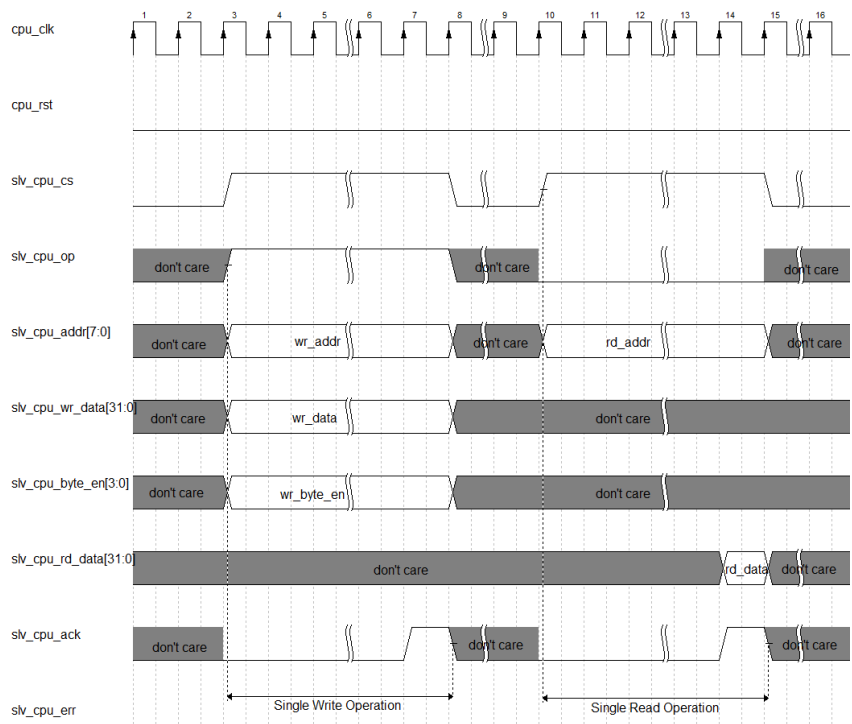
ユーザーが CPU Slave Interface を介してレジスタを読み出す場合、インターフェースを選択するために **slv_cpu_cs** を High にプルアップすると同時に、読み出しを示すために **slv_cpu_op** を Low にプルダウンする必要があります。ユーザーは、**slv_cpu_addr** 信号を介してレジスタアドレスを送信します。IP がこの読み出しを完了すると、**slv_cpu_ack** 信号が High にプルアップされると同時に、読み出しデータが **slv_cpu_rd_data** に表示されます。**slv_cpu_cs**、**slv_cpu_op**、**slv_cpu_addr** 信号は、**slv_cpu_ack** 信号が High にプルアップされるまで変更されないままである必要があります。CPU Slave Interface の読み出し中に、IP は 4 バイトの読み出しデータを返します。CPU Slave Interface の読み出しタイミングを図 4-7 に示します。

図 4-7 CPU Slave Interface の読み出しタイミング



CPU Slave Interface の交互の読み出しと書き込みタイミングを図 4-8 に示します。

図 4-8 CPU Slave Interface の交互の読み出しと書き込みタイミング



4.11.2 SDIO CMD52 Interface

Host が CMD52 を介して CIS または Function1 レジスタにアクセスする場合、ユーザーは SDIO CMD52 Interface を介してデータを送受信する必要があります。IP は、CIS または Function1 レジスタにアクセスするための CMD52 を受信した後、SDIO CMD52 インターフェースで対応するデータリクエストを生成し、ユーザーは対応するタイミングに従って応答する必要があります。

SDIO CMD52 Interface の書き込み(RAW flag=1)

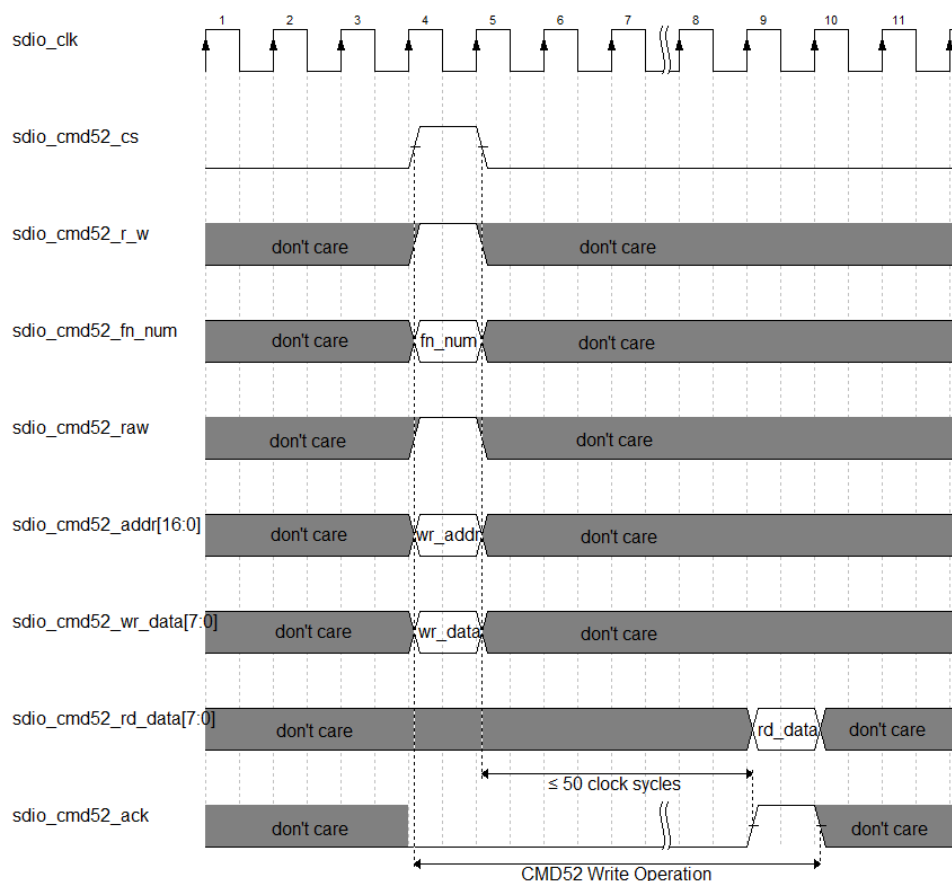
Host が CMD52 を介して書き込みを実行し、かつ RAW flag が 1 の場合、ユーザーはレジスタの書き込みの完了後に現在のレジスタ値を返す必要があります。

IP が正しい CMD52 書き込みを受信し、レジスタアドレスが Function1 の場合、IP は SDIO CMD52 Interface でデータ書き込みリクエストを生成します。このとき、IP は、SDIO CMD52 Interface にデータリクエストがあることを示すために、sdio_cmd52_cs 信号を High にプルアップします。それと同時に、この操作が RAW flag = 1 の書き込みであることを示すよう、sdio_cmd52_r_w 信号と sdio_cmd52_raw 信号が High にプルアップされます。sdio_cmd52_cs = 1 の場合、sdio_cmd52_fn_num、sdio_cmd52_addr、および sdio_cmd52_wr_data 信号は有効です。ユーザーが書き込みを完了した後、書き込みが完了したことを IP に通知するために、sdio_cmd52_ack を 1 サイクル High にプルアップする必要があります。IP は、sdio_cmd52_ack が 1 であることを検出すると、R5 を Host に自動的に送信します。

この書き込みでは RAW flag = 1 であるため、ユーザーは sdio_cmd52_ack を High にプルアップすると同時に、sdio_cmd52_rd_data に書き込まれたレジスタの現在の値を返す必要があります。この値は R5 の Read or Write Data フィールドで Host に返されます。

SDIO CMD52 Interface の書き込み(RAW flag=1)タイミングを図 4-9 に示します。

図 4-9 SDIO CMD52 Interface の書き込み(RAW flag = 1)タイミング



SDIO CMD52 Interface の書き込み(RAW flag=0)

Host が CMD52 を介して書き込みを実行し、かつ RAW flag が 0 の場合、ユーザーはレジスタの書き込みの完了後に現在のレジスタ値を返す必要がありません。

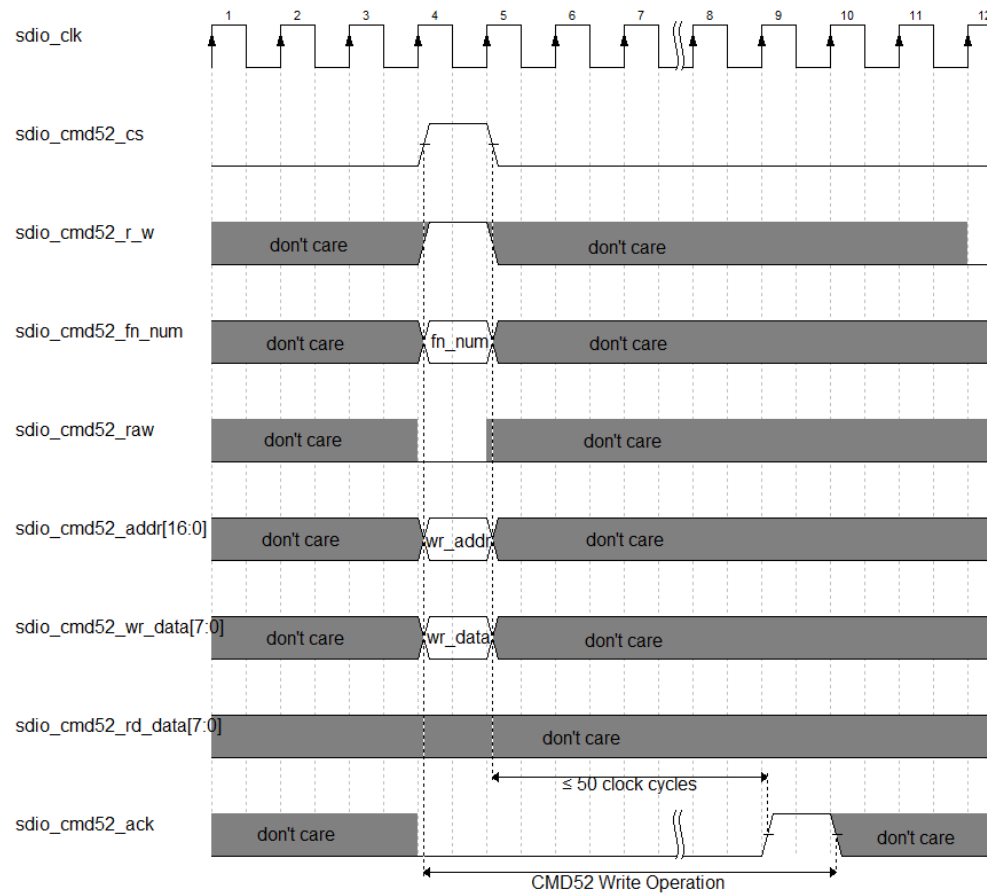
IP が正しい CMD52 書き込みを受信し、レジスタアドレスが Function1 の場合、IP は SDIO CMD52 Interface でデータ書き込みリクエストを生成します。このとき、IP は、SDIO CMD52 Interface にデータリクエストがあることを示すために、sdio_cmd52_cs 信号を High にプルアップします。それと同時に、この操作が RAW flag = 0 の書き込みであることを示すよう、sdio_cmd52_r_w 信号が High にプルアップされ、sdio_cmd52_raw 信号が High にプルアップされます。sdio_cmd52_cs = 1 の場合、sdio_cmd52_fn_num、sdio_cmd52_addr、および sdio_cmd52_wr_data 信号は有効です。ユーザーが書き込みを完了した後、書き込みが完了したことを IP に通知するために、sdio_cmd52_ack を 1 サイクル High にプルアップする必要があります。IP は、sdio_cmd52_ack が 1 であることを検出すると、R5 を Host に自動的に送信します。

この書き込みでは RAW flag = 0 であるため、ユーザーは sdio_cmd52_ack を High にプルアップすると同時に、sdio_cmd52_rd_data に書き込まれたレジスタの現在の値を返す必要がありません。IP は、CMD52 の Write Data フィールドの値を R5 の Read or Write Data フィー

ルドに自動的に入れて、Host に返します。

SDIO CMD52 Interface の書き込み(RAW flag=0)のタイミングを図 4-10 に示します。

図 4-10 SDIO CMD52 Interface の書き込み(RAW flag = 0)のタイミング



SDIO CMD52 Interface の読み出し

Host が CMD52 を介して読み出す場合、ユーザーは読み出しリクエストを受信した後にレジスタの値を返す必要があります。

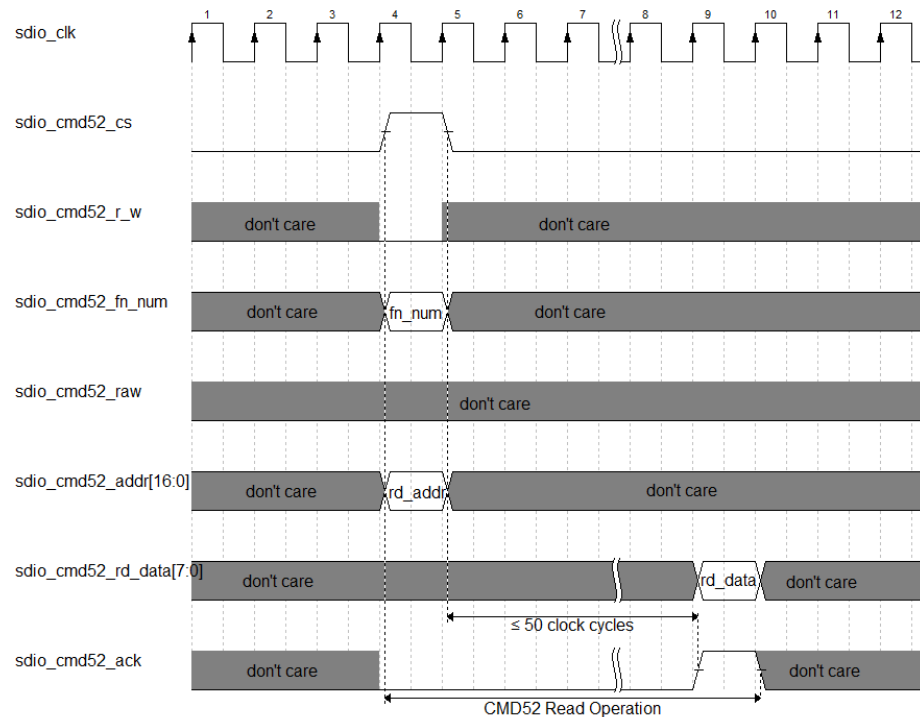
IP が正しい CMD52 読み出しを受信し、レジスタアドレスが CISFunction1 の場合、IP は SDIO CMD52 Interface でデータ読み出しリクエストを生成します。このとき、IP は、SDIO CMD52 Interface にデータリクエストがあることを示すために、sdio_cmd52_cs 信号を High にプルアップします。それと同時に、この操作が読み出しであることを示すよう、sdio_cmd52_r_w 信号が Low にプルダウンされます。sdio_cmd52_cs = 1 の場合、sdio_cmd52_fn_num 信号と sdio_cmd52_addr 信号は有効です。ユーザーが読み出しを完了した後、読み出しが完了したことを IP に通知するために、sdio_cmd52_ack を 1 サイクル High にプルアップする必要があります。IP は、sdio_cmd52_ack が 1 であることを検出すると、R5 を Host に自動的に送信します。

読み出しであるため、ユーザーは sdio_cmd52_ack を High にプルアッ

プすると同時に、`sdio_cmd52_rd_data` に読み出されたレジスタの値を返す必要があります。この値は R5 の Read or Write Data フィールドで Host に返されます。

SDIO CMD52 Interface の読み出しタイミングを図 4-11 に示します。

図 4-11 SDIO CMD52 Interface の読み出しタイミング



CMD52 タイムアウト

SDIO 規格では、Command のデータの最後のビットから Response のデータの最初のビットまでの遅延 NCR 範囲が 2～64 クロックサイクルであることを指定しています。したがって、`sdio_cmd52_cs` 有効から `sdio_cmd52_ack` valid 有効までのクロック間隔は、50 clock cycles を超えてはなりません(IP の内部サンプリング遅延のため)。sdio_cmd52_cs が有効になってから 50 clock cycles 以内に、ユーザーが `sdio_cmd52_ack` を返すことができない場合、この CMD52 操作を中止してください。sdio_cmd52_ack を再度 High にプルアップする必要はありません。この時点で、Host はこの操作がタイムアウトだと判断します。

4.11.3 SDIO CMD53 Interface

Host が CMD53 を介して CIS または Function1 レジスタを読み出しと書き込みする場合、ユーザーは SDIO CMD53 Interface を介してデータの受送信を制御する必要があります。IP が CIS または Function1 レジスタを読み出しと書き込みするための CMD53 を受信した後、SDIO CMD53 Interface は対応するデータリクエストを生成し、ユーザーは対応するタイミングに従って応答する必要があります。

CMD53 のコマンドには、Bytes Transfer モードと Block Transfer モー

ドが含まれています。Bytes Transfer モードの場合、SDIO CMD53 Interface は、CMD53 Byte/Block Count フィールドの長さの読み出しと書き込みを生成します。Block Transfer モードの場合、SDIO CMD53 Interface は N 回の読み出しと書き込みを生成し、N は CMD53 Byte/Block Count フィールドで、長さは、対応する Function Block Size です。したがって、ユーザーは操作が Bytes Transfer か Block Transfer かを認識できません。対応するタイミングに従ってデータを転送するだけです。

SDIO SDR104 CMD53 Interface の書き込み

各書き込みリクエストは、Bytes Transfer 書き込みまたは Block Transfer の単一 Block 書き込みを指します。

IP が正しい CMD53 書き込みを受信し、アドレスが Function1 の場合、IP は自動的に R5 に応答します。その後、SDIO CMD53 Interface でデータ書き込みリクエストを生成します。IP は、SDIO CMD53 Interface にデータ書き込みリクエストがあることを示すために、sdio_cmd53_wr_en 信号を High にプルアップします。sdio_cmd53_wr_en = 1 の場合、sdio_cmd53_fn_num、sdio_cmd53_addr、sdio_cmd53_len、sdio_cmd53_op_code 信号は有効です。

Host が sdio_dat を介して IP ヘデータを書き込み始めると、ユーザーは sdio_cmd53_wr_valid および sdio_cmd53_wr_data を介して書き込みデータを受信できます。一度に書き込まれるデータのバイト数は、sdio_cmd53_len の値に相当します。IP は CRC16 と End bit の正確さを自動的に検証します。

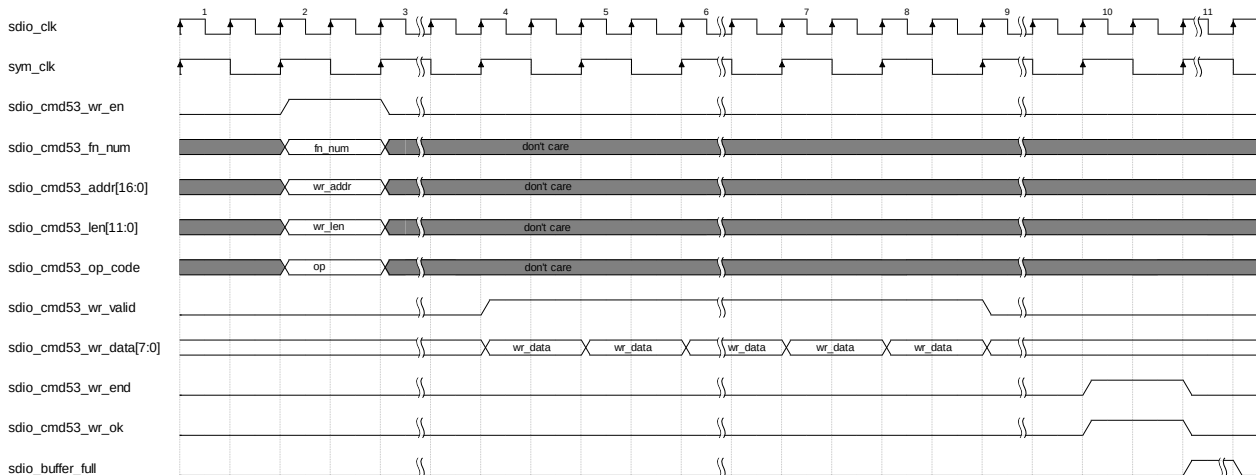
Host が Bytes Transfer 書き込みまたは Block Transfer の単一 Block 書き込みを完了し、かつ IP が Host に CRC status と End bit をホストに送信した場合、書き込みが終了したことをユーザーに通知するために、sdio_cmd53_wr_end が High にプルアップされます。また、sdio_cmd53_wr_ok は、今回の書き込みデータが正しいかどうかを示します。sdio_cmd53_wr_ok は sdio_cmd53_wr_end と同期します。sdio_cmd53_wr_ok が 1 の場合、書き込みが正しいことを意味し、ユーザーは書き込みのデータを処理する必要があります。

Block Transfer では、IP が block 書き込みでエラーを受信すると、IP は後続のデータを受信しないことになります。

sdio_cmd53_wr_end = 1 を検出した場合、ユーザーは、後続のデータを受信し続けるアイドルバッファがあるかどうかを sdio_buffer_full を介して IP に通知する必要があります。ユーザーの buffer が full で、データの受信を続行できない場合、sdio_cmd53_wr_end = 1 を検出した直後に sdio_buffer_full を High にプルアップする必要があります。ユーザーが sdio_buffer_full を High にプルアップすると、IP は sdio_dat0 を Low にプルダウンし、ユーザーが sdio_buffer_full を Low にプルダウンするまで card busy 状態のままにします。Host は、sdio_dat0 が 0 であることを検出すると、sdio_dat0 が High にプルアップされたことを検出するまで次の書き込みを続行しません。

SDIO CMD53 Interface の 4 ビットモードの書き込みのタイミングを図 4-12 に示します。

図 4-12 SDIO CMD53 Interface の 4 ビットモードの書き込みのタイミング



SDIO SDR104 CMD53 Interface の読み出し

各読み出しリクエストは、Bytes Transfer 読み出しまたは Block Transfer の単一 Block 読み出しを指します。

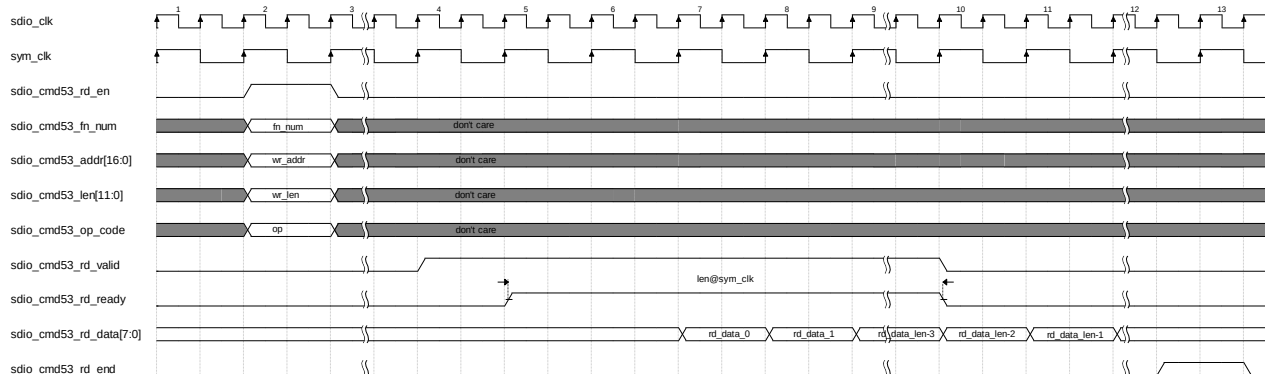
IP が正しい CMD53 読み出しを受信し、アドレスが CIS または Function1 の場合、IP は自動的に R5 に応答します。その後、SDIO CMD53 Interface でデータ読み出しリクエストを生成します。IP は、SDIO CMD53 Interface にデータ読み出しリクエストがあることを示すために、sdio_cmd53_rd_en 信号を High にプルアップします。sdio_cmd53_rd_en = 1 の場合、sdio_cmd53_fn_num、sdio_cmd53_addr、sdio_cmd53_len、sdio_cmd53_op_code 信号は有効です。

ユーザーが CMD53 読み出しリクエストを受信すると、ユーザーは sdio_cmd53_rd_valid、sdio_cmd53_rd_data、および sdio_cmd53_rd_ready を介して読み出しデータを送信できます。sdio_cmd53_rd_ready 信号が High にプルアップされることは、IP が 1 つの読み出しデータを受信する準備ができていることを示します。タイミングを最適化するために、ユーザーは 2 サイクル後に sdio_cmd53_rd_data 信号で新しいデータを再度準備する必要があります。一度に読み出されるデータのバイト数は sdio_cmd53_len の値に相当します。ユーザーは、sdio_cmd53_len の値に応じて sdio_cmd53_rd_valid の長さを制御し、十分なバイトを送信した後に sdio_cmd53_rd_valid を Low にプルダウンする必要があります。IP は、Host のリクエストデータの長さに応じてデータを sdio_dat に送信します。データが送信された後、IP は CRC16 フィールドと End bit を自動的に送信します。その後、IP は sdio_cmd53_rd_end 信号を 1 サイクルプルアップして、読み出しが終了したことをユーザーに通知します。

IP は Read Wait をサポートしていません。

SDIO CMD53 Interface の 4 ビットモードの読み出しのタイミングを図 4-13 に示します。

図 4-13 SDIO SDR104 CMD53 Interface の 4 ビットモードの読み出し



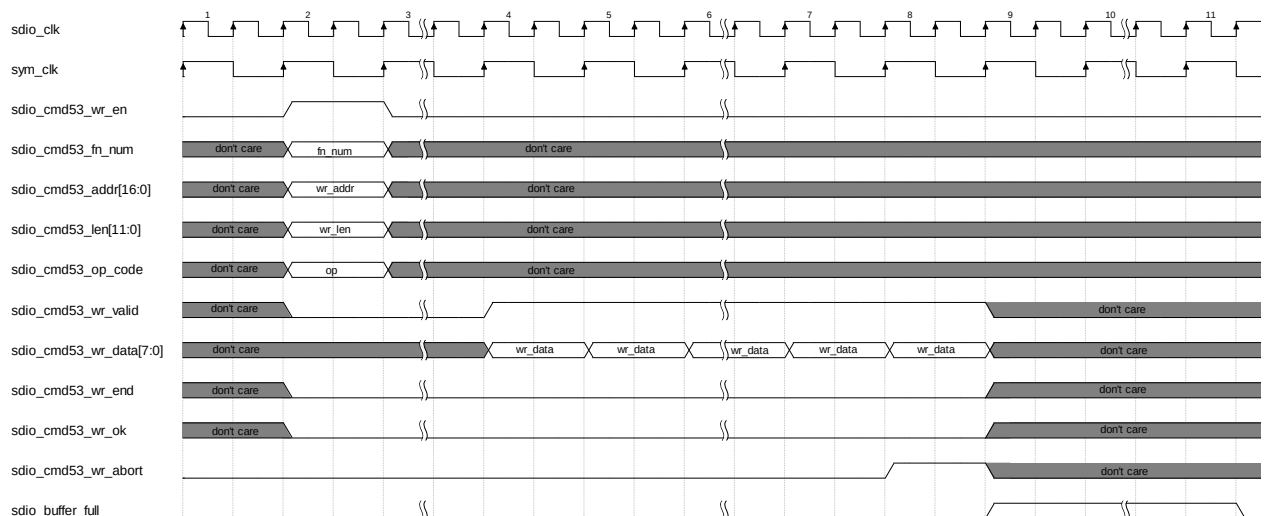
Abort 操作

Host が CMD53 を送信して Infinite Block Mode でデータを読み出しと書き込みする場合、CMD52 を介して Abort コマンドを送信し、読み出しと書き込みを中断できます。

IP が CMD52 Abort を受信してデータを書き込んでいるとき、IP は `sdio_cmd53_wr_abort` を 1 クロックサイクルプルアップして、書き込みを終了するようユーザーに通知します。

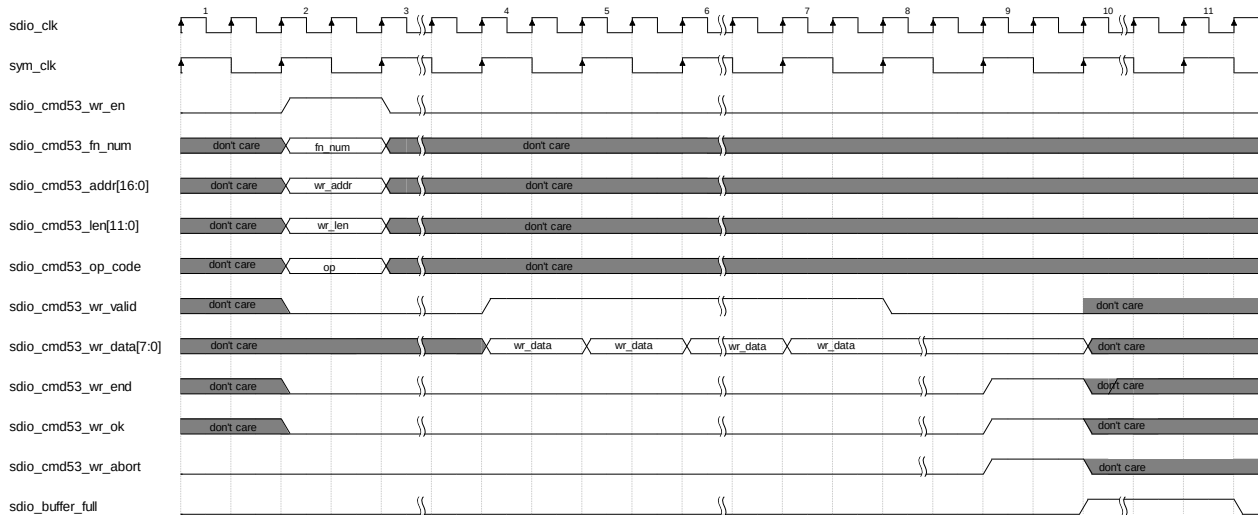
ユーザーが `sdio_cmd53_wr_abort = 1` を検出し、かつこれより前に `sdio_cmd53_wr_end = 1` を検出していない場合、この abort では Bad Abort をトリガしたということになります。つまり、CRC status と End bit の送信を完了する前に IP が abort を受信したということです。この場合、IP は `sdio_cmd53_wr_end = 1` を生成しません。ユーザーは今回の書き込みを終了し、書き込みのデータを破棄する必要があります(図 4-14)。

図 4-14 Write Bad Abort



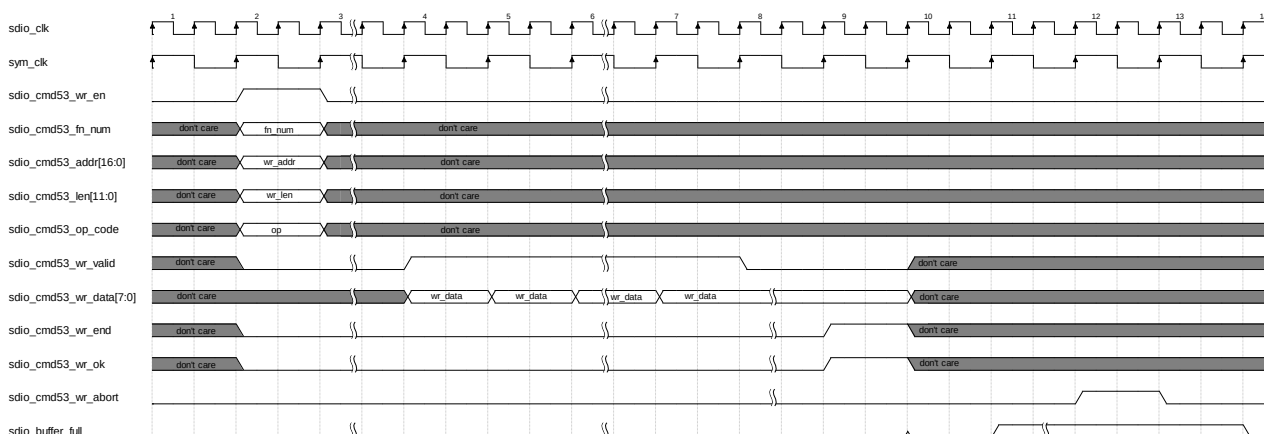
ユーザーが `sdio_cmd53_wr_abort = 1` と `sdio_cmd53_wr_end = 1` を同時に受信した場合、この `abort` では **Good Abort** をトリガしたということになります。つまり、**CRC status** と **End bit** の送信を完了するときに IP が `abort` を受信したということです。ユーザーはこの書き込みのデータを受信する必要があります(図 4-15)。

図 4-15 Write Good Abort Case1



ユーザーが `sdio_cmd53_wr_end = 1` の後に `sdio_cmd53_wr_abort = 1` を受信したことは、`abort` が **card busy** 状態でトリガされたことを意味します。つまり、IP は **CRC status** と **End bit** の送信を完了した後に `abort` を受信し、**Good Abort** をトリガしたということです。ユーザーはこの書き込みのデータを受信する必要があります(図 4-16)。

図 4-16 Write Good Abort Case2

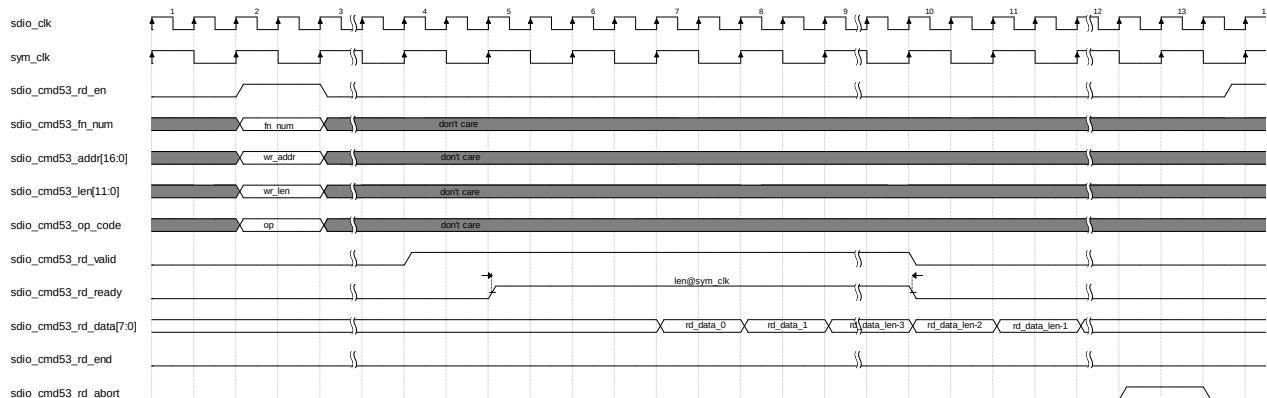


IP が **CMD52 Abort** を受信してデータを読み出しているとき、IP は `sdio_cmd53_rd_abort` を 1 クロックサイクルプルアップして、読み出しを終了するようユーザーに通知します。

ユーザーが `sdio_cmd53_rd_abort = 1` を検出し、かつこれより前に `sdio_cmd53_rd_end = 1` を検出していない場合、この `abort` では **Bad Abort** をトリガしたということになります。つまり、**CRC16** と **End bit** の送信を

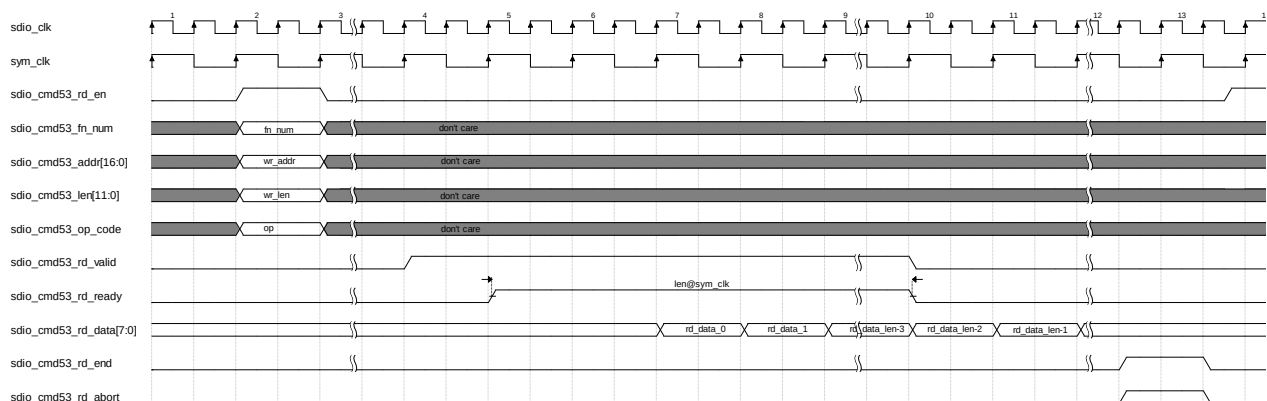
完了する前に IP が **abort** を受信したということです。この場合、IP は **sdio_cmd53_rd_end = 1** を生成しません。ユーザーは今回の失敗した読み出しを終了する必要があります(図 4-17)。

図 4-17 Read Bad Abort



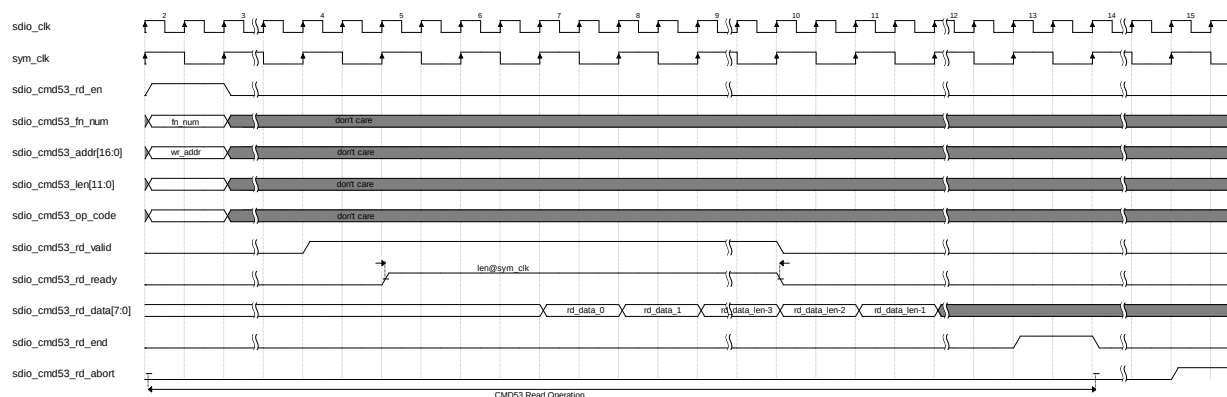
ユーザーが **sdio_cmd53_rd_abort = 1** と **sdio_cmd53_rd_end = 1** を同時に受信した場合、この **abort** では **Good Abort** をトリガしたということになります。つまり、**CRC16** と **End bit** の送信を完了するときに IP が **abort** を受信したこととなり、今回の読み出しが成功したということです(図 4-18)。

図 4-18 Read Good Abort Case1



ユーザーが **sdio_cmd53_rd_end = 1** の後に **sdio_cmd53_rd_abort = 1** を受信したことは、**abort** が 2 つの読み出しの間でトリガされたことを意味します。つまり、IP は **CRC16** と **End bit** の送信を完了した後に **abort** を受信し、**Good Abort** をトリガしたということです(図 4-19)。

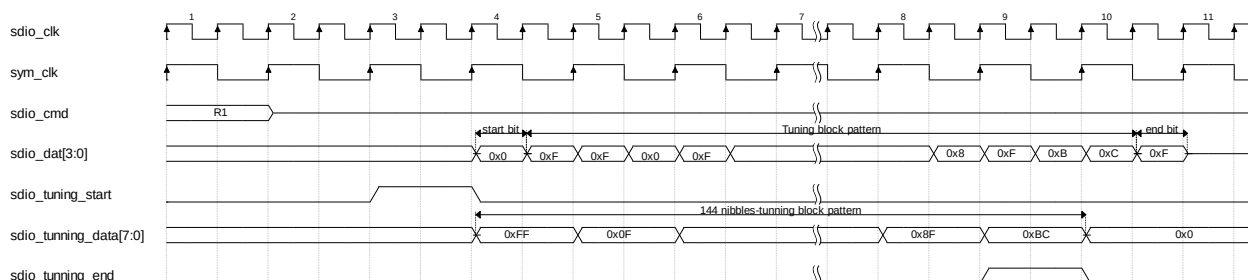
図 4-19 Read Good Abort Case2



4.11.4 SDIO Tuning Interface

IP が UHS-I Card に構成されている場合は、CMD19 をサポートします。正しい CMD19 を受信すると、IP は自動的に R1 に応答します。R1 を送信した後、ユーザーは SDIO Tuning Interface を介して Tuning Block Pattern を送信する必要があります。Tuning Block Pattern については、図 4-5 を参照して下さい。SDIO Tuning Interface のタイミングを図 4-20 に示します。

図 4-20 SDIO Tuning Interface のタイミング



4.12 IP の初期化プロセス

1. 電源投入。
2. rstn を Low にプルダウンしてグローバルリセットを開始。
3. rstn を High にプルアップしてグローバルリセットを完了。
4. ユーザーは、CPU Slave Interface を介して IP 関連のパラメータを設定可能。以下のレジスタを参照してください：
 - Reg0x04 bit22 の構成：IP が低速カードかどうか
 - Reg0x00 bit24 の構成：IP が High Speed をサポートするかどうか
 - Reg0x08 bit23～0 の構成：CCCR CIS Pointer
 - Reg0x2C bit23～0 の構成：Function1 CIS Pointer
 - Reg0x34 bit15～0 の構成：CCCR Max Block Size
 - Reg0x34 bit31～16 の構成：Function1 Max Block Size
 - Reg0x30 bit0 の構成：IO Ready = 1
5. 上記の構成が完了した後、IP は CMD5 を受信すると、R4 C フィールドで 1 を応答して SDIO Host に IO Ready を通知。
6. SDIO Host が CMD7 を介して IP を選択した後、IP は Command State に入り、Host は CMD52 および CMD53 を介して IP CIA レジスタを読み出しと書き込み可能。
7. SDIO Host は、CMD52 を介して CCCR IOE1 = 1 のように構成し、Function1 をイネーブルするには、IP は fun1_ioe を High にプルアップする必要がある。
8. ユーザーは Function1 の初期化を開始。
9. Function1 が初期化された後、ユーザーは fun1_ior を High にプルアップし、つまり CCCR IOR1 = 1 のように設定する必要がある。
10. SDIO Host は、CMD52 を介して CCCR IOR1 = 1 を読み出した後、Function1 レジスタへのアクセスを開始。
11. Host が block モードで読み出しと書き込みしたい場合、まず CMD52 コマンドを使用して対応する Function の block size レジスタを構成する必要がある、かつ構成された値は上記のポイント 4 の対応する Max Block Size 以下でなければならない。Function0 Block Size レジスタは CCCR レジスタ 10h-11h、Function1 Block Size レジスタは FBR1 レジスタ 110h-111h。これらの 2 つのレジスタはリトルエンディアンモードで、構成範囲は 1～2048。
12. IP の割り込み信号を受信する必要がある場合、Host は CMD52 を介して IP の割り込みイネーブルレジスタ CCCR 04h を開く必要がある。

13. 上記の SDIO レジスタの詳細については、SDIO プロトコルを参照してください。

5ポート一覧

Gowin SDIO SDR104 Slave Controller IP の IO ポートは表 5-1 に示すとおりです。

表 5-1 Gowin SDIO SDR104 Slave Controller IP の IO ポート

信号	方向	幅	説明
SDIO Interface			
sdio_clk	入力	1	SDIO クロック入力信号
sym_clk	出力	1	SDIO が出力するユーザークロック
sdio_cmd_in	入力	1	SDIO Command/Response 信号入力
sdio_cmd_out	出力	1	SDIO Command/Response 信号出力
sdio_cmd_oen	出力	1	SDIO Command/Response 信号出力イネーブル、アクティブ Low
sdio_dat0_in	入力	1	SDIO 1 ビットモードの場合は dat 信号入力、SDIO 4 ビットモードの場合は dat0 入力
sdio_dat0_out	出力	1	SDIO 1 ビットモードの場合は dat 信号出力、SDIO 4 ビットモードの場合は dat0 信号出力
sdio_dat0_oen	出力	1	SDIO 1 ビットモードの場合は dat 信号出力イネーブル、SDIO 4 ビットモードの場合は dat0 信号出力イネーブル、アクティブ Low
sdio_dat1_in	入力	1	SDIO 1 ビットモードの場合は割り込み信号入力、SDIO 4 ビットモードの場合は dat1 と割り込み多重化信号入力
sdio_dat1_out	出力	1	SDIO 1 ビットモードの場合は割り込み信号出力、SDIO 4 ビットモードの場合は dat1 と割り込み多重化信号出力
sdio_dat1_oen	出力	1	SDIO 1 ビットモードの場合は割り込み信号出力イネーブル、SDIO 4 ビットモードの場合は dat1 および割り込み多重化信号出力イネーブル、アクティブ Low
sdio_dat2_in	入力	1	SDIO 1 ビットモードの場合は Read Wait 信号入力、SDIO 4 ビットモードの場合は dat2 と Read Wait の多重化信号入力

信号	方向	幅	説明
sdio_dat2_out	出力	1	SDIO 1 ビットモードの場合は Read Wait 信号出力、SDIO 4 ビットモードの場合は dat2 と Read Wait の多重化信号出力
sdio_dat2_oen	出力	1	SDIO 1 ビットモードの場合は Read Wait 信号出力イネーブル、SDIO 4 ビットモードの場合は dat2 と Read Wait の多重化信号出力イネーブル、アクティブ Low
sdio_dat3_in	入力	1	SDIO 1 ビットモードの場合は定義されておらず、SDIO 4 ビットモードの場合は dat3 入力。
sdio_dat3_out	出力	1	SDIO 1 ビットモードの場合は定義されておらず、SDIO 4 ビットモードの場合は dat3 出力。
sdio_dat3_oen	出力	1	SDIO 1 ビットモードの場合は定義されておらず、SDIO 4 ビットモードの場合は dat3 出力イネーブル、アクティブ Low。
Reset			
rstn	入力	1	グローバル非同期リセット、アクティブ Low
cmd52_rst	出力	1	Host が cmd52 を介して SDIO をリセットする場合、この信号は 1 です。sdio_clk と同期。
Function1 Enable/Ready			
fun1_ioe	出力	1	CCCRIOE1 の値を示す Function1 イネーブル信号。sdio_clk と同期。
fun1_ior	入力	1	Function1 準備完了フラグ。CCCR IOR1 Reg0x3 bit1 に対応。
Function1 Interrupt			
fun1_interrupt	入力	1	0 : Function1 割り込みなし 1 : Function1 割り込みあり
CPU Slave Interface(cpu_clk と同期)			
cpu_clk	入力	1	cpu インターフェースクロック入力
cpu_rst	入力	1	cpu_clk と同期する cpu インターフェース同期リセット入力信号、アクティブ High
slv_cpu_cs	入力	1	cpu インターフェースのチップセレクト信号、アクティブ High
slv_cpu_op	入力	1	cpu インターフェースの操作モード 0 : 読み出し。1 : 書き込み。
slv_cpu_addr	入力	8	cpu インターフェースの操作アドレス
slv_cpu_wr_data	入力	32	cpu インターフェースの書き込みデータ
slv_cpu_byte_en	入力	4	CPU インターフェースの書き込みバイトのイネーブル 1 : イネーブル。0 : ディセーブル。 bit 3 は slv_cpu_wr_data[31:24]に対応。 bit 2 は slv_cpu_wr_data[23:16]に対応。 bit 1 は slv_cpu_wr_data[15:8]に対応。

信号	方向	幅	説明
			bit 0 は slv_cpu_wr_data[7:0]に対応。
slv_cpu_rd_data	出力	32	cpu インターフェースの読み出しデータ
slv_cpu_ack	出力	1	cpu インターフェースの操作応答。このビットが 1 ということは、操作が終了したことを示します。
slv_cpu_err	出力	1	CPU インターフェースの操作エラーの表示。 slv_cpu_ack が 1 の場合に有効。 1 : 操作エラー; 0 : 操作は正しい。
SDIO CMD52 Interface(sdio_clk と同期)			
sdio_cmd52_cs	出力	1	SDIO CMD52 コマンドイネーブル。 0 : ディセーブル。1 : イネーブル
sdio_cmd52_r_w	出力	1	IO_RW_DIRECT Command (CMD52)コマンド R/W flag フィールドに対応する SDIO CMD52 コマンドの読み出しと書き込み。 0 : 読み出し。1 : 書き込み
sdio_cmd52_fn_num	出力	1	IO_RW_DIRECT Command (CMD52)コマンド Function Number フィールドに対応する SDIO CMD52 コマンド Function Number。 0 : Function0 ; 1 : Function1
sdio_cmd52_raw	出力	1	IO_RW_DIRECT Command (CMD52)コマンド RAW flag フィールドに対応する SDIO CMD52 コマンドの RAW イネーブル。 0 : ディセーブル。1 : イネーブル
sdio_cmd52_addr	出力	17	IO_RW_DIRECT Command (CMD52)コマンド Register Address フィールドに対応する SDIO CMD52 コマンドの操作アドレス
sdio_cmd52_wr_data	出力	8	IO_RW_DIRECT Command (CMD52)コマンド Write Data フィールドに対応する SDIO CMD52 コマンドの書き込みデータ。SDIO CMD52 インターフェースが書き込みモードの場合に有効。
sdio_cmd52_rd_data	入力	8	IO_RW_DIRECT Response (R5)コマンド Read Data フィールドに対応する SDIO CMD52 コマンドの読み出しデータまたは RAW データ。 SDIO CMD52 インターフェースが読み出しモードの場合、ユーザーは読み出されたレジスタの値を返す必要があります。 SDIO CMD52 インターフェースが RAW 操作モードの場合、ユーザーは書き込まれたレジスタの値を返す必要があります。
sdio_cmd52_ack	入力	1	SDIO CMD52 コマンドの ack 信号。ユーザーが ack 信号を 1 サイクル High にプルアップすると、IP はユーザーの CMD52 操作が終了したと判断し、R5 の送信を開始します。 この操作が読み出しの場合、ユーザーは、ack を High にプルアップすると同時に、読み出しアド

信号	方向	幅	説明
			レスレジスタの値を <code>sdio_cmd52_rd_data</code> で IP に返す必要があります。 この操作が RAW 操作の場合、ユーザーは、ack を High にプルすると同時に、書き込みアドレスレジスタの値を <code>sdio_cmd52_rd_data</code> で IP に返す必要があります。
SDIO CMD53 Interface(<code>sdio_clk</code> と同期)			
<code>sdio_cmd53_wr_en</code>	出力	1	SDIO CMD53 の書き込みイネーブル。 0 : ディセーブル。1 : イネーブル
<code>sdio_cmd53_rd_en</code>	出力	1	SDIO CMD53 の読み出しイネーブル。 0 : ディセーブル。1 : イネーブル
<code>sdio_cmd53_fn_num</code>	出力	1	IO_RW_EXTENDED Command (CMD53) コマンド Function Number フィールドに対応する SDIO CMD53 コマンド Function Number。 0 : Function0 ; 1 : Function1
<code>sdio_cmd53_addr</code>	出力	17	IO_RW_EXTENDED Command (CMD53) コマンド Register Address フィールドに対応する SDIO CMD53 コマンドの操作の先頭アドレス
<code>sdio_cmd53_len</code>	出力	12	この操作中にユーザーが読み出しと書き込みする必要があるバイト数を示す SDIO CMD53 コマンドの操作の長さ。
<code>sdio_cmd53_op_code</code>	出力	1	IO_RW_EXTENDED Command (CMD53) コマンド OP Code フィールドに対応する SDIO CMD53 コマンド OP Code。 0 : Multi byte R/W to fixed address ; 1 : Multi byte R/W to incrementing address。
<code>sdio_cmd53_wr_valid</code>	出力	1	SDIO CMD53 書き込みデータが有効。このビットが 1 の場合、 <code>sdio_cmd53_wr_data</code> が有効。
<code>sdio_cmd53_wr_data</code>	出力	8	SDIO CMD53 書き込みデータ。 <code>sdio_cmd53_wr_valid</code> が 1 の場合、書き込みデータが有効。
<code>sdio_cmd53_wr_end</code>	出力	1	SDIO CMD53 書き込みデータ終了フラグ。このビットが 1 とのことは、書き込みが終了し、IP が CRC status と End bit を Host に送信したことを示します。
<code>sdio_cmd53_wr_ok</code>	出力	1	SDIO CMD53 書き込みデータが正しい。このビットが <code>sdio_cmd53_wr_end</code> と同期します。 1 : この書き込みが正しい 0 : この書き込みで CRC16 または End bit または CRC status の送信エラーが発生しました。
<code>sdio_cmd53_wr_abort</code>	出力	1	書き込みの abort フラグ。Host が CMD52 コマンドを介して書き込みを終了する場合、このビットは 1 です。このビットが 1 とのことは、書き込みを終了したことを示します。

信号	方向	幅	説明
sdio_cmd53_rd_valid	入力	1	SDIO CMD53 読み出しデータが有効。読み出し中、ユーザーがデータを読み出す準備ができたなら、このビットを High にプルアップしてデータの準備ができたことを IP に通知する必要がある、そして IP は sdio_data で読み出しデータの送信を開始します。
sdio_cmd53_rd_data	入力	8	SDIO CMD53 読み出しデータ。読み出し中にユーザーが IP に提供した読み出しデータ。
sdio_cmd53_rd_ready	出力	1	SDIO CMD53 読み出しデータ ready 信号。このビットが 1 とのことは、IP が sdio_cmd53_rd_data を取得していることを示し、ユーザーは次の sdio_cmd53_rd_data を準備する必要があります。
sdio_cmd53_rd_end	出力	1	SDIO CMD53 読み出しデータ終了フラグ。このビットが 1 とのことは、読み出しが終了し、IP が CRC status と End bit を Host に送信したことを示します。
sdio_cmd53_rd_abort	出力	1	読み出しの abort フラグ。 Host が CMD52 コマンドを介して読み出しを終了する場合、このビットは 1 です。このビットが 1 とのことは、読み出しを終了したことを示します。
sdio_buffer_full	入力	1	buffer full フラグ。 1 : Full。 0 : Not full。 書き込みが完了したとき (IP が CRC status と End bit を送信したとき)、ユーザー buffer が full で次のデータを受信できない場合、このビットを High にプルアップする必要があります。この場合、IP は sdio_dat0 を Low にプルダウンして、 Host に card busy を通知します。ユーザーがこのビットを Low にプルダウンすると、 Host が次の書き込みを実行できるよう、IP は sdio_dat0 を High にプルアップする必要があります。
SDIO Tuning Interface			
sdio_tuning_start	出力	1	SDIO Tuning Block Pattern 開始信号。IP が Tuning Block Pattern を送信する必要がない場合、このピンはフローティングのままにします。
sdio_tuning_data	入力	4	SDIO Tuning Block Pattern データ信号。IP が Tuning Block Pattern を送信する必要がない場合、このピンの値を常に 0 に設定します。
sdio_tuning_end	入力	1	SDIO Tuning Block Pattern 終了信号。IP が Tuning Block Pattern を送信する必要がない場合、このピンの値を常に 0 に設定します。
2MHz Clock			
clk_2mhz	入力	1	2MHz クロック入力。IP が UHS-I Card に構成さ

信号	方向	幅	説明
			れ、電圧スイッチングタイミングを実行するために CMD11 コマンドが必要な場合、 sdio_clk が停止したかどうかを検出するために、このインターフェースに 2MHz のクロックを入力する必要があります。IP が Non UHS Card に構成され、または CMD11 コマンドが不要な場合、このピンの値を常に 0 に設定します。

6 パラメータの構成

ユーザーは、設計要件に従って Gowin SDIO SDR104 Slave Controller のさまざまな静的パラメータとタイミングパラメータを構成する必要があります(表 6-1)。

表 6-1 Gowin SDIO SDR104 Slave Controller の静的パラメータとタイミングパラメータ

名称	説明	オプション
Card Type	● Non UHS Card は Low-Speed および Full-Speed をサポートします。そのうち Full-Speed には Default-Speed と High-Speed が含まれます。 ● UHS-I Card は SDR104 をサポートします。	Non UHS/UHS-I

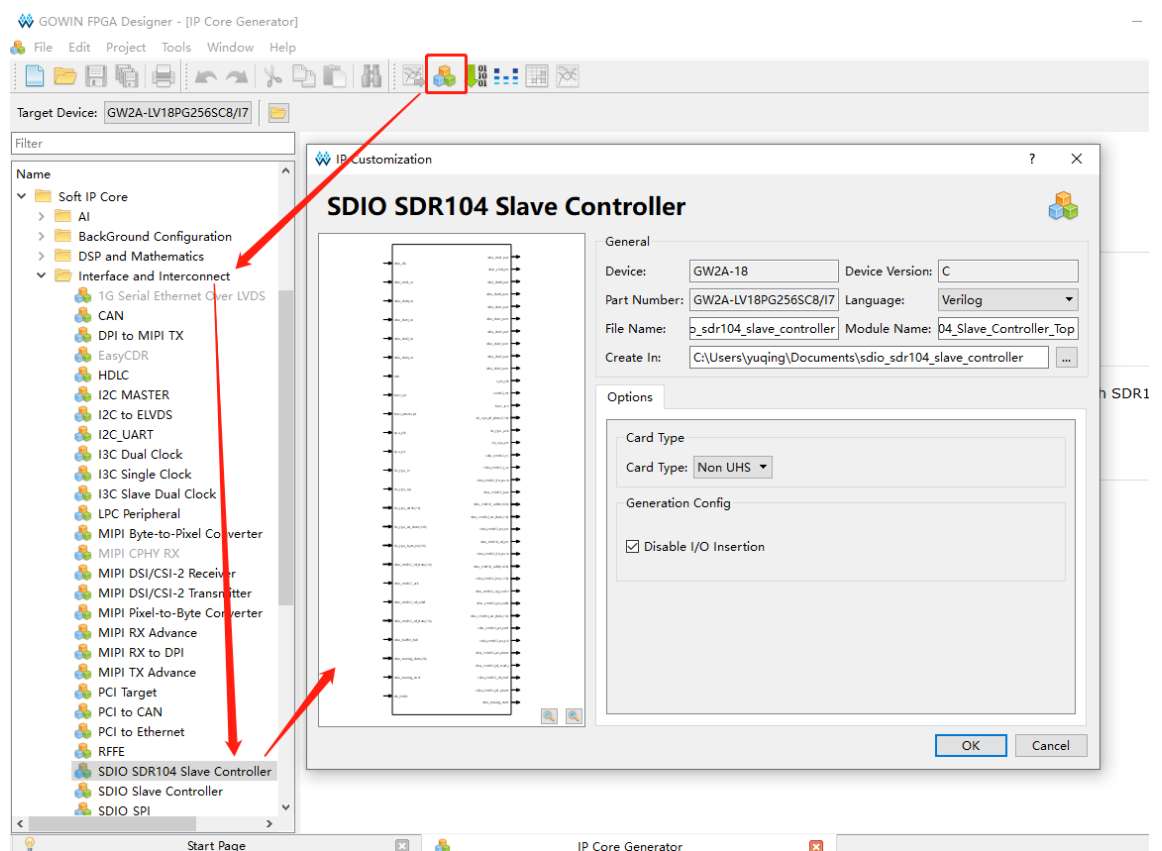
7 GUIでの構成

ユーザーは、Gowin ソフトウェアの IP Core Generator を使用して SDIO SDR104 Slave Controller IP を呼び出し・構成することができます。本章では、Non UHS Card を例に、その構成画面、構成手順、及び各構成オプションの意味を説明します。

1. IP Core Generator を開く

プロジェクトを作成した後、Tools > IP Core Generator を開き、SDIO SDR104 Slave Controller を選択します(図 7-1)。

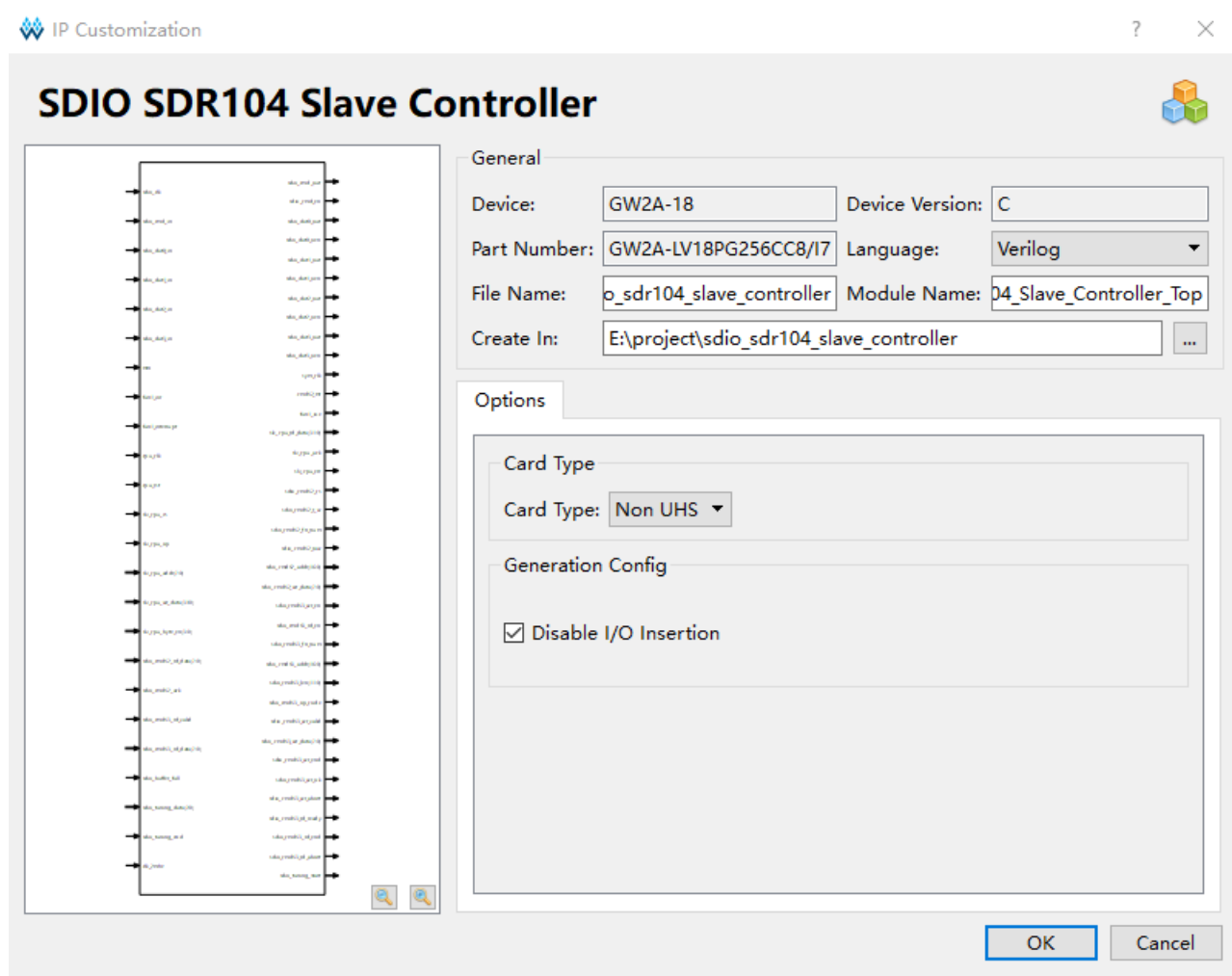
図 7-1 IP Core Generator



2. SDIO SDR104 Slave Controller インターフェース

構成画面の左側は、SDIO SDR104 Slave Controller IP のインターフェースの説明図で、右側は IP オプションです(図 7-2)。

図 7-2 SDIO SDR104 Slave Controller IP の構成画面



8 リファレンス・デザイン

詳細については、GOWIN のホームページの SDIO Slave Controller の [リファレンス・デザイン](#)を参照してください。

