SD Specifications

Part E1

SDIO Specification

Version 3.00 Draft 1.01

March 29, 2010

Technical Committee
SD Card Association

CONFIDENTIAL

Draft Only Page

Draft Revision History for Version 2.00

Date	Version	Changes compared to previous issue
July 8, 2005	Draft	Add Items discussed in Feb 2005 San Francisco Meeting • Add High-Speed Register to CCCR and section on High-Speed Mode • Add CCCR Revision 1.2 • MANFID tuple in each function's CIS • Clarify definition of total number of function in CMD5 response • Change Operational Voltage Requirements • Combine Chapters 12 (Physical Properties) and 13 (Mechanical Extensions) and add miniSDIO to the new Chapter 13 (Physical Properties) • Add pins 10 and 11 for miniSDIO to SDIO pin definitions • Update Physical Spec version for Normative Reference
October 7, 2005	Draft	Includes changes from issue tracking document (2005-09-20) and Sep 2005 Lisbon F2F. • Add Embedded SDIO ATA Function Code • Remove Embedded SDIO section • Change description of high-speed switching • Change formatting to reflect SD templates • Pins 10 and 11 should be left open • Writing EHS is ignored if SHS=0 • Add timing of 8 clocks to speed mode switch • Either CMD6 or CMD52 (SHS) will set speed for both memory and IO portions of combo card • Either CMD0 or CMD52 (RES) will reset speed for both memory and IO portions of combo card
December 20, 2005	Draft	Editorial changes in Section 12, 12.1 and 12.21 Renumber Tables and Figures; Update Draft and Revision History and Keyword section per SD Template Add v1.20 to SDIOx Add Figure 12-1 for High-Speed timing switch Add CMD6 to Tables A-1 and A-2
June 9, 2006	Draft	Incorporate changes from Shanghai F2F Meeting Sections 4.5 and 12
July 26, 2006	Draft	(p. iii) Reword Revision History (p. v) Change heading from "Exemption" to "Disclaimer" Change "SD Host products" to "SD Host/Ancillary products" (p. 1) Change Physical Spec reference from 1.01 to 1.10 Add clarification that v1.20 applies to Physical Spec 1.10 or later Add note that Physical Spec 1.10 recommends not to use 2.0-2.7V for basic communication (p.14) Reword section 3.4.5, Enabling CRC in SPI Combo Card (p. 17) Add paragraph in 4.5 on changing bus width of locked cards (p. 21) Remove sentence about bit 4 in 4.10.8 (p. 34) The restriction to SD mode applies to SRW, not SMB. The

		sentence was moved from SMB to SRW.
August 30, 2006	Draft	Change Version to 2.00 since Physical Specification 2.00 has been released, and change primary reference to Physical Spec v 2.00. Replace initialization flowcharts 3-2 and 3-3 to include high capacity cards. Change Physical Spec references from 1.10 to having no version number Add v2.00 to SDIOx
September 7, 2006r	Draft	Final draft for TC Review
September 12, 2006	Draft	Final draft for TC Review A mistake of locked card in section 4.5 is fixed, (P19) Links to Tables and Figures are fixed. (P6,P24,P32,P61,P62)
September 26, 2006	Draft	Final draft for IP Review
October 12, 2006	Draft	Final draft for IP Review Remove the 2.0-2.7V communication voltage range. (p.1,p.11,p.69)

Draft Revision History for Version 3.00

Date	Version	Changes compared to previous issue
June 1, 2009	Draft 0.30	Chapter 1: Definition of Embedded SDIO is added
June 24, 2009	Draft 0.30	Figure 1-1 is added Table 2-1 ANT1 and ANT2 are added Figure 2-2 is added Section 3.2 S18R is added to CMD5 argument Section 3.3 S18A is added to CMD5 response Figure 3-1 MMC is removed Figure 3-3, 3-4 Flow Charts are modified Section 4.7 P bit definition is changed as the Physcial 3.00. Section 6.3 Asynchronous Interrupt is added Section 6.9 CCCR New bits or new registers are added Section 10 New Sections are created Section 10.4 Read Wait Timing (More Than 50MHz) is added Figure 10-4 is added Order of Appendix is changed according to the template. Special Terms and Abbreviation are divided Hex is indicated "h" instead of "x0" Chapter and Section
July 10, 2009	Draft 0.50	Some sentences are modified according to feedback by WG Chair. Section 11.2.4 High-Power Tuples is modified. Appendix Table and Figure numbers are fixed. The first draft to open SDIO WG.

Date	Version	Changes compared to previous issue
September 4, 2009	Draft 0.51	Page3: Figure 1-1 and explanation in Section 1.4 are changed Page18: Reference to the Physical Spec is added for CMD11/CMD19 Page35: Table 6-6 1 is changed Page37: 8-bit bus mode is added in Bus Width Page37: S8B is added for support bit of 8-bit mode Page45: Interface Code of Bluetooth Type A AMP (9h) is added in FBR Page67: Power State Control is mandatory for > 500mA consumption Page74: Fix explanation of WP Swich Page90: The tuple of Extended Data 03h is used for embedded Page92: Section 17.1.3 8-Bit Bus Mode is added
October 1, 2009	Draft 0.70	English language check is performed. Page 8: "the Physical Layer Specification" is added. Page33: "built to the Physical Layer Specification" is removed. Page43: Reference to Physical Ver3.0x is added in Table 6-2. Page46: VDDH is changed to device in Table 6-4 Page68: VDDH is changed to device in Table 11-1 Page68: Explanation of Tuple for Power State is changed. Page72: Refer to Mechanical Addendum Page72: Tile of Table 13-1 is changed Page79: Section 14.3 SDIO Current is changed. Page90: Title of Table 16-15 is added Page93: Appendix A1 Physical Ver3.01 is added
October 2, 2009	Draft 0.70	Some portions referred to the Physical Specification are fixed. "microSDIO Card" is added in Appendix A1.
October 20, 2009	Draft 0.71	Section 1.1.2 "Embedded SDIO initialization sequence" is removed. Relation of eSD Addendum is changed. Section 3.1.2 Explanation regarding CMD5 is changed. Application Notes is added for host implementation. CMD5 arg=0 is retrieved in Figure 3-2 and 3-3. Section 6.9 S8B in CCCR Add a sentence that 8-bit mode is added in future. Section 17.1.3 8-Bit Bus Mode 8-bit mode reference to eSD Version 3.00 is removed. Add a sentence that 8-bit mode is added in future. Reference to eSD Version 3.00 is removed below Figure 17-1 eS Addendum Version 2.10 is added in Appendix A.1
October 22, 2009	Draft 0.80	Section 4.10.2, 4.10.3, 4.10.6, 4.10.7 Fixed to "Physical Layer Specification Version 1.01" P16 Regarding CMD5, reference of Physical Ver1.01 is removed. Page 4,7,8,14,15,16,19,20,21,24,26,30,31,33,34,37,40,43,47,49,52, 55,64,67,69,71,73,74,76,81,82,83,84,85,87,90 Fixed to "Refer to Section xxx" Figure 12-1 to 12-2 numbers are fixed. P78: TBD is removed from the title of Section 13.4 microSDIO. Deleted lines are removed.
October 22, 2009	Draft 0.81	The Final Draft Page 34, 08h:SDC Table 6-3 is changed to Table 6-2 Page 59, Figure 8-11 is fixed. Block Gap Interrupt shall be disabled in SDR50, SDR104 and DDR50.

Date	Version	Changes compared to previous issue
October 22, 2009	Draft 1.00	Draft for TC review
December 16, 2009	Draft 1.01	Draft for TC review Page26: Table 4-8 Fix Typo of OUT_OF_RANGE and Error Page29: Table 5-1 Fix Typo of OUT_OF_RANGE. Page43: 200mA is changed to 720mW in SMPC. Page44: 200mA is changed to 720mW in EMPC. Page48: Explanation of PSx is changed. Page68-72: Explanation of Power Control is changed. Explanation is changed mW basis. Page88: Add new fields to Tuple for Function 0. Page89: Explanation of TPLFE_TC_n and TPLFE_CP_n are added. Page90: The voltage condition is added to TPLFE_OP_MIN_PWR TPLFE_OP_AVG_PWR and TPLFE_OP_MAX_PWR. Page92: 3.3V is removed from Table 16-12. Page93: Table 6-13 1.8V Power State Tuple is removed. Page93: New Table 6-13 is added to explain TPLFE_PS_n. Unit of Power State Tuple is changed to mW.
March 29, 2010	Draft 1.01	Draft for IP review Clean up changes. Section 16.7.6 is removed. Then 16.7.7 and 16.7.8 are renumbered. Fixed Typos Page 28 "5.2.1" is changed to "Section 5.2.1" Page 81 "Section 15" is changed to "Chapter 15" Page 91 "Chapter 11.2.4" is changed to "Section 11.2.4" Appendix A1: Added final release date of reference specifications.

Revision History

Date	Version	Changes compared to previous issue
October, 2001	1.00	Base version initial release
August 18, 2004	1.10	Followings are modified in this version (1) Voltage Range is now either Standard (2.7-3.6V) or Minimal (3.1-3.5V) (2) Added High-Power support for cards requiring more than 200mA. (3) Added support for greater than 15 standard SDIO standard functions (4) Defined inrush current limits (5) Changed method to disable card detect resistor in combo cards (6) Removed Inhibited Interrupt Section (7) Added requirement to support CMD52 during Data Transfer (8) Added Small Form-Factor SDIO physical description (9) Update tuples to implement the changes noted above (10) Changed R5 ERROR type from E R X to E R (11) Fix typos and add clarifications
January 30, 2007	2.00	Followings are modified in this version (1) Added method to change bus speed (Normal Speed up to 25MHz and High Speed up to 50 MHz) (2) Operational Voltage Requirement is extended to 2.7-3.6V (3) Combine sections 12 (Physical Properties) and 13 (Mechanical Extensions) and add miniSDIO to the new section 13 (Physical Properties) (4) Add Embedded SDIO ATA Standard Function Interface Code (5) Reference of Physical Ver2.00 supports SDHC combo card. (6) Some typos in Ver1.10 are fixed.
March 29, 2010	3.00	Followings functions are added in this version (1) Definition of Embedded SDIO (2) Ultra High Speed Bus Speed Mode (UHS-I) (3) Power Control Extension (Power State Control) (4) Asynchronous Interrupt in 4-bit mode (5) Shared Bus for supporting multiple devices (6) Interface Signal of Embedded Device applicable to Shared Bus

To the extent this proposed specification, which is being submitted for review under the IP Policy, implements, incorporates by reference or refers to any portion of versions 1.0 or 1.01 of the SD Specifications (including Parts 1 through 4), adoption of the proposed specification shall require Members utilizing the adopted specification to obtain the appropriate licenses from the SD-3C, LLC, as required for the utilization of those portion(s) of versions 1.0 or 1.01 of the SD Specifications.
For example, implementation of the SD Specifications in a host device under versions 1.0 or 1.01 and under the adopted specification requires the execution of a SD Host Ancillary License Agreement with the SD-3C, LLC; and implementation of the SD Specifications under versions 1.0 or 1.01 and under the proposed specification in a SD Card containing any memory storage capability (other than for storage of executable code for a controller or microprocessor within the SD Card) requires the execution of a SD Memory Card License Agreement with the SD-3C, LLC.

Conditions for publication

Publisher and Copyright Holder:

SD Card Association
2400 Camino Ramon, Suite 375
San Ramon, CA 94583 USA
Telephone: +1 (925) 275-6615,
Fax: +1 (925) 886-4870
E-mail: office@sdcard.org

Confidentiality:

The contents of this document are deemed confidential information of the SD-3C LLC and/or the SD Card Association (the "Disclosers"). As such, the contents and your right to use the contents are subject to the confidentiality obligations stated in the written agreement you entered into with the Disclosers which entitled you to receive this document, such as a Non-Disclosure Agreement, the License Agreement for SDA Memory Card Specifications (also known as "LAMS"), the SD Host/Ancillary Product License Agreement (also known as "HALA") or the IP Policy.

Disclaimers:

The information contained herein is presented only as a standard specification for SD Card and SD Host/Ancillary products. No responsibility is assumed by SD Card Association for any damages, any infringements of patents or other right of the third parties, which may result from its use. No license is granted by implication or otherwise under any patent or rights of SD Card Association or others.

Conventions Used in This Document

Naming Conventions

- Some terms are capitalized to distinguish their definition from their common English meaning. Words not capitalized have their common English meaning.

Numbers and Number Bases

- Hexadecimal numbers are written with a lower case "h" suffix, e.g., FFFFh and 80h.
- Binary numbers are written with a lower case "b" suffix (e.g., 10b).
- Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1000 0101 0010b.
- All other numbers are decimal.

Key Words

- May: Indicates flexibility of choice with no implied recommendation or requirement.
- Shall: Indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interchangeability and to claim conformance with the specification.
- Should: Indicates a strong recommendation but not a mandatory requirement. Designers should give strong consideration to such recommendations, but there is still a choice in implementation.

Application Notes

Some sections of this document provide guidance to the host implementers as follows:

Application Note: This is an example of an application note.

Table of Contents

Table of Tables

Table of Figures

1. General Description

This Part E1 SDIO (SD Input/Output) Specification defines the SD bus interface specification for SDIO including register specification. Not only is the SDIO Card defined, but also the Embedded SDIO Device and Combo Card are defined. SDIO is based on the Physical Layer Specification and the SDIO Specification provides extension and modification of the Physical Layer Specification for SDIO card and device.

1.1 Definitions

1.1.1 SDIO Card

The SDIO (SD Input/Output) card is a removable product that utilizes the SD bus and SD commands. The same form factor as a memory card can be used and an extended form factor for SDIO card is defined by the SDIO Specification. (No extended form factor is defined for microSDIO in this version.)

The SDIO Card shall comply with the Part E1 SDIO Specification and the Part 1 Physical Layer Specification (The SDIO Specification provides extension and modification of the Physical Layer specification for SDIO. The SDIO Specification takes precedence).
The SDIO card shall have compatibility to the SD Memory Card regarding SD Bus pins layout, electrical, power and signaling. The SPI mode is mandatory in the SDIO Cards but not all features may be available in SPI mode.

1.1.2 Embedded SDIO

An Embedded SDIO Device is a product that utilizes the SD bus and SD commands.
"Embedded" is defined as a permanently soldered on a PCB (Printed Circuit Board) non-removable device or mounted through a socket permanently soldered on a PCB such that the Embedded SDIO Device can't be removed by end users.

The Embedded SDIO Device shall comply with the Part E1 SDIO Specification and the Part 1 Physical Layer Specification. (The SDIO Specification provides extension and modification of the Physical Layer specification for SDIO card and device. The SDIO Specification takes precedence.) When standard interface specification (for example, host interface voltage and timing) is required, SDIO card and device should follow the Part 1 eSD Addendum.
The Embedded SDIO Device may be any form factor and any pin layout. The SPI mode is optional in the Embedded SDIO Devices. Not all features may be available in SPI mode.

1.2 SDIO Features

1.2.1 Common SDIO Features

- Targeted for portable and stationary applications
- Minimal or no modification to the SD Physical bus is required
- Minimal change to the memory initialization sequence.
- Multi-function support including multiple I/O and combined I/O and memory
- Up to 7 I/O functions plus one memory function is supported on one card.
- An interrupt is supported to request the host to service to an event
- Standard Application Specifications for Standard SDIO Functions can be defined by Part Ex
- Card information is provided by Tuples.

1.2.2 SDIO Card Features

- Supply Voltage range: 2.7-3.6V (Operational Voltage is used for Initialization)
- Removable and Plug and play (PnP) support

- Multiple Form Factors:
  - Standard-Size SDIO (Extended form factor can be used)
  - miniSDIO (Extended form factor can be used)
  - microSDIO
- A card per slot connection

1.2.3 Embedded SDIO Device Features

- Supply Voltage range: 2.7-3.6V or 1.7-1.95V.
- Non removable device.
- Any form factor and any pin layout
- Shared Bus connection is possible

1.3 Document Structure

Reference documents are described in the Appendix A.
SDIO Specification describes modifications of the Physical Layer Specification and new functions for SDIO. The standard card form factors are defined in the Part 1 Mechanical Addenda. The extended card form factors for Standard-Size SDIO card are defined in the SDIO Specification.
This specification refers any released versions of the Physical Layer Specification Version 3.00 and later and the eSD Addendum Version 2.10 and later.

1.4 Standard SDIO Functions

Figure 1-1 shows the SDIO related specifications. The SDIO bus specification is defined by the Physical Layer specification and this SDIO specification. The Memory portion of a Combo card is specified by the Physical Layer specification. An SDIO and a Combo Card have an SDIO function on the backend.
Associated with the base SDIO specification, there are several Application Specifications for Standard SDIO Functions. The feature of SDIO is determined by a SDIO function. Card driver and application software are required to control the function. There are two SDIO function types: standard SDIO function and non standard SDIO function.
The standard functions such as cameras, Bluetooth cards and GPS receivers have a standard register interface, a common operation method and a standard CIS extension. By defining a standard register interface for a specific function, an OS vendor can provide a standard card driver, application software and API for the functions. Full information on the standard functions is defined by Part E2 to Part E6.
In the case of non standard SDIO functions, a card or device manufacturer needs to provide a card driver. The user may need to install the card driver and the application software to use the non standard SDIO function. Implementation of the non standard register interfaces is optional for any card vendor, but compliance with the standard allows the use of standard drivers and applications.

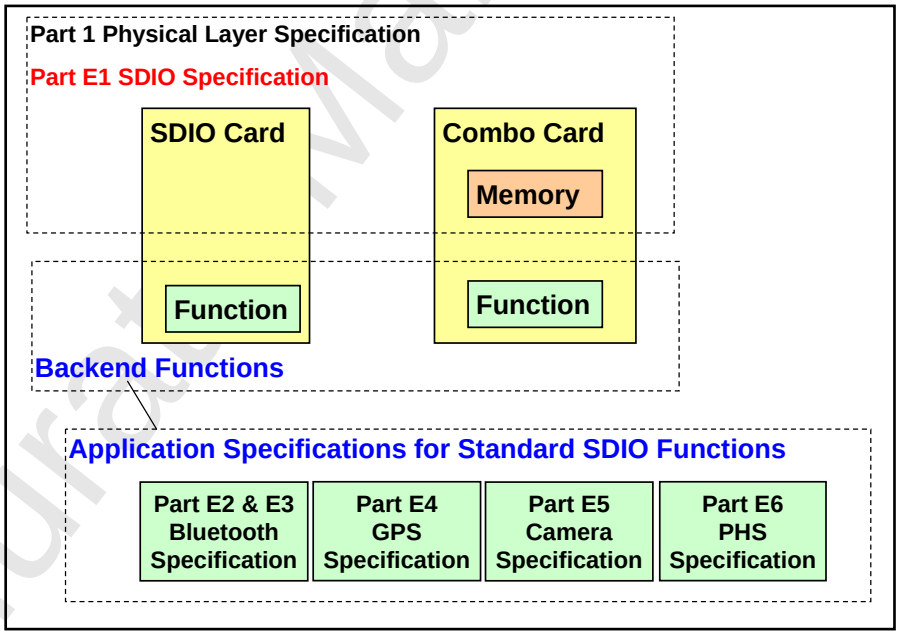


Figure 1-1 : SDIO Related Specifications

In the following Chapters and Sections, the term "card" needs to be interpreted as not only "SDIO Card and Combo Card" but also "Embedded SDIO Device" unless otherwise noted.

2. SDIO Signaling Definition

2.1 SDIO Card Types

This specification defines two types of SDIO cards. The Full-Speed card supports SPI, 1-bit SD and the 4-bit SD transfer modes at the full clock range of 0-25MHz. Full-Speed SDIO cards have a data transfer rate of over 100 Mb/second (10 MB/Sec). The second version of the SDIO card is the Low-Speed SDIO card. This card requires only the SPI and 1-bit SD transfer modes. 4-bit support is optional. In addition, Low-Speed SDIO cards shall support a full clock range of 0-400 KHz. The intended use of Low-Speed cards is to support low-speed I/O capabilities with a minimum of hardware. The Low-Speed cards support such functions as modems, bar-code scanners, GPS receivers etc. If a card is a 'Combo card' (memory plus SDIO) then Full-Speed and 4-bit operation is mandatory for both the memory and SDIO portions of the card.

2.2 SDIO Card Modes

There are three bus modes defined for SD memory cards that also apply to SDIO Cards:

2.2.1 SPI (Card mandatory support)

The SPI bus topology is defined in Section 3.5.2 and the protocol is defined in Sections 3.6.2 and Chapter 7 of the Physical Layer Specification Version 3.0x. In this mode pin 8, which is undefined for memory, is used as the interrupt pin. All other pins and signaling protocols are identical to the Physical Layer Specification.

New functions defined after the SDIO Version 2.00 may not be supported in the SPI mode.

2.2.2 1-bit SD Data Transfer Mode (Card Mandatory Support)

This mode is identical to the 1 data bit (narrow) mode defined for SD Memory in Section 3.6.1 of the Physical Layer Specification. In this mode, data is transferred on the DAT[0] pin only. In this mode pin 8, which is undefined for memory, is used as the interrupt pin and pin 9, which is undefined for memory, is used as the read wait pin. All other pins and signaling protocols are identical to the Physical Layer Specification.

2.2.3 4-bit SD Data Transfer Mode (Mandatory for High-Speed Cards, Optional for Low-Speed)

This mode is identical to the 4 data bit mode (wide) defined for SD Memory in Section 3.6.1 of the Physical Layer Specification. In this mode, data is transferred on all 4 data pins (DAT[3:0]). In this mode the interrupt pin is not available for exclusive use as it is utilized as a data transfer line. Thus, if the interrupt function is required, special timing is required to provide interrupts. Refer to Section 8.1.2 for details of this operation. UHS-I bus speed mode has been added from SDIO Version 3.00 onward. The 4-bit SD mode provides the highest data transfer possible, up to 104 MB/sec.

2.3 SDIO Host Modes

If a SDIO aware host supports the SD transfer mode, it is recommended that both the 1-bit and 4-bit modes be supported. While an SDIO host that supports *only* the 4-bit transfer mode is possible, its performance with a Low-Speed SDIO card may be reduced. This is because the only means to transfer data to and from a Low-Speed card would be the single byte per command transfer (using the IO_RW_DIRECT command (CMD52), refer to Section 5.1).

2.4 Signal Pins

2.4.1 Signal Pins for SDIO Card

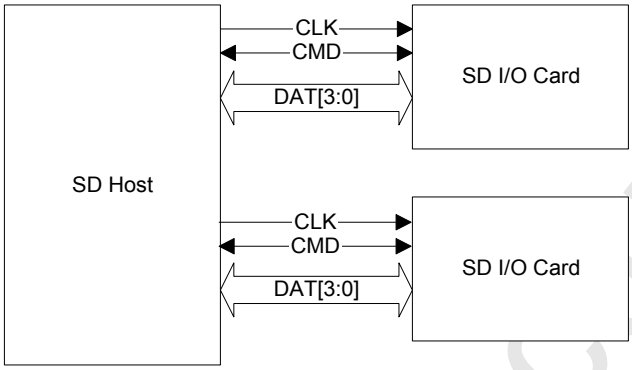


Figure 2-1 : Signal connection to two 4-bit SDIO cards

Pin	SD 4-bit mode		SD 1-bit mode		SPI mode	
1	CD/DAT[3]	Data line 3	N/C	Not Used	CS	Card Select
2	CMD	Command line	CMD	Command line	DI	Data input
3	VSS1	Ground	VSS1	Ground	VSS1	Ground
4	VDD	Supply voltage	VDD	Supply voltage	VDD	Supply voltage
5	CLK	Clock	CLK	Clock	SCLK	Clock
6	VSS2	Ground	VSS2	Ground	VSS2	Ground
7	DAT[0]	Data line 0	DATA	Data line	DO	Data output
8	DAT[1]	Data line 1 or Interrupt (optional)	IRQ	Interrupt	IRQ	Interrupt
9	DAT[2]	Data line 2 or Read Wait (optional)	RW	Read Wait (optional)	N/C	Not Used

Table 2-1 : SDIO Pin Definitions

It is recommended that multi-slot hosts intending to support SDIO (SDIO aware) provide a separate CLK to each slot, to allow the I/O cards to be placed in a low power state on a slot-by-slot basis. After reset, all data lines (DAT[3:0]) shall be in the hi-Z state on both the host and card(s) to avoid bus conflict. Access to the Bus Interface Control register within the CCCR (Table 6-1) determines DAT line mode.

Interface for the Embedded SDIO is described in Chapter 17.

2.5 Host Requirements for SDIO

In order for a host to completely support all of the capabilities of the SDIO cards, some signal connections should be supported. In order to support interrupts, the host should have Pin 8 connected from the card to the host in order to provide interrupt signaling. This is true even if the host only supports the SPI or 1 bit SD mode. In addition, if the host supports more than 1 card in either SD mode, the CMD and all 4 data lines (DAT[3:0]) should not be bussed together, but rather routed separately to the host. This allows the mixing of card types in the different sockets without interference. Both the Physical Layer Specification and the SDIO Specification support the concept of "unifying" (connecting together) the CMD lines in a multi-slot system after initialization.

In addition, there some additional design details that the designer of a host intending to support SDIO cards must be aware of:

If a host supports both the 4-bit SD bus and interrupts during 4-bit transfer the host shall control the value placed on the DAT[3:1] lines. Those conditions are:

1) During a multiple block write:
   According to the Physical Layer Specification Version 1.01 figures 9 and 28, the DAT[3:1] lines are described as "don't care" (X) during the CRC status period. If a host actively drives these lines during this period, it may interfere with interrupt signaling from an SDIO card. In order to prevent this conflict, if a host supports interrupts during 4-bit data transfers, it shall not drive DAT[3:1] during this period (hi-Z rather than don't care).

2) During a multiple block read:
   According to the Physical Layer Specification Version 3.0x Figure 4-29, the DAT[1] line is described as "P" (one cycle pull-up) between read data packets. In order to support interrupts during the 4-bit mode, the host shall not drive the DAT[1] line during the 2 clock Interrupt Period defined in Section 8.1.2.

3. SDIO Card Initialization

3.1 Initialization Sequence

3.1.1 Initialization by Non-I/O Aware Host

A requirement for the SDIO specification is that an SDIO card shall not cause non-I/O aware hosts to fail when inserted. In order to prevent operation of I/O functions in non-I/O aware hosts, a change to the SD card identification mode flowchart is needed. A new command (IO_SEND_OP_COND, CMD5) is added to replace the ACMD41 for SDIO initialization by I/O aware hosts (Refer to Section 3.2).

After reset or power-up, all I/O functions on the card are disabled and the I/O portion of the card shall not execute any operation except CMD5 or CMD0 with CS=low. If there is SD memory on the card (also called a combo card), that memory shall respond normally to all mandatory memory commands.

An I/O only card shall not respond to the ACMD41 command. The host shall then give up and disable this card. Thus, the non-aware host receives no response from an I/O only card and forces it to the inactive state. The operation of an I/O card with a non-I/O aware host is shown in Figure 3-1 Note that the solid lines are the actual paths taken while the dashed lines are not executed.

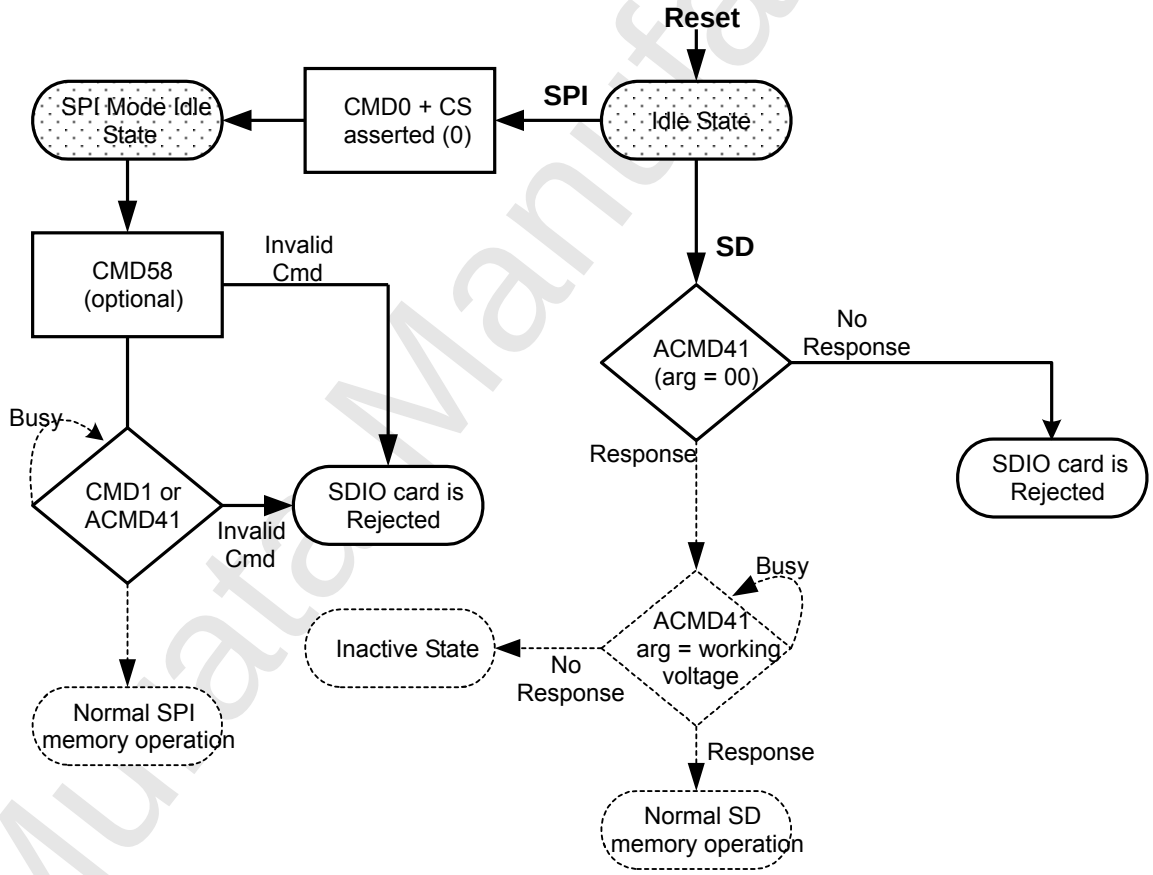


Figure 3-1 : SDIO Response to Non-I/O Aware Initialization

3.1.2 Initialization by I/O Aware Host

An SDIO aware host sends CMD5 prior to the CMD55/ACMD41 pair, and thus would receive a valid OCR in the R4 response to CMD5 and continue to initialize the card. Figure 3-2 shows the operation of an SDIO aware host operating in the SD modes and Figure 3-3 shows the same operation for a host that operates in the SPI mode.

If the I/O portion of a card has received no CMD5, the I/O Section remains inactive and shall not respond to any command except CMD5. A combo card stays in the memory-only mode. If no memory is included on the card, the card would not respond to any memory command. If the I/O aware host sends a CMD5 to the card, the card responds with R4. The host then reads that R4 value and knows the number of available I/O functions and if any SD memory exists.
A host that supports UHS-I sets S18R to 1 in the argument of CMD5 to request a change of the signal voltage to 1.8V. If the card supports UHS-I and the current signal voltage is 3.3V, S18A is set to 1 in the R4 response. If the signal voltage is already 1.8V, the card sets S18A to 0 so that host maintains the current signal voltage. UHS-I is supported in SD mode and S18A is always 0 in SPI mode.

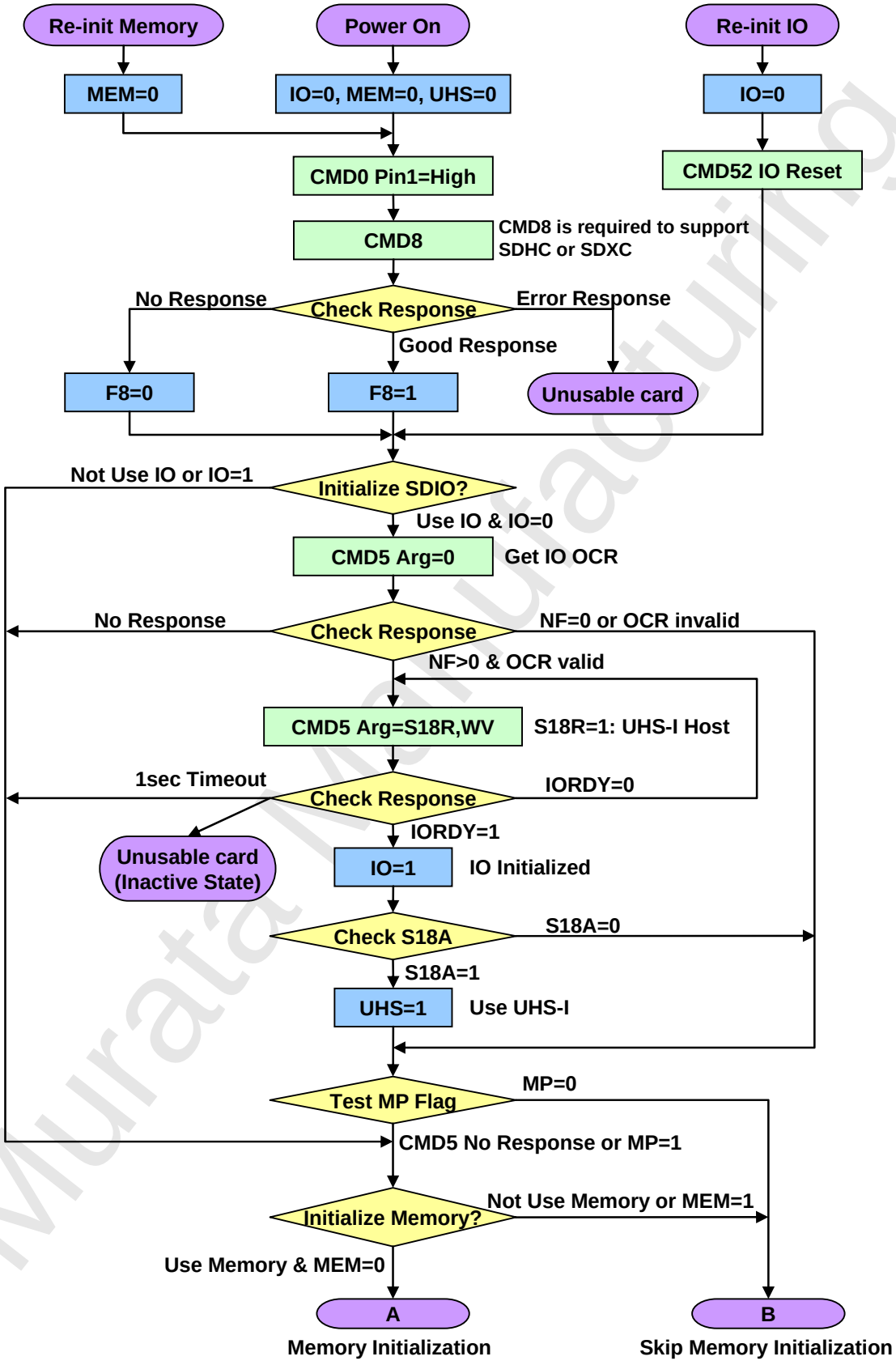
After the host has initialized the I/O portion of the card, it then reads the Common Information Area (CIA) of the card (Refer to Section 6.8). This is done by issuing a read command, starting with the byte at address 00h, of I/O function 0. The CIA contains the Card Common Control Registers (CCCR) and the Function Basic Registers (FBR). Also included in the CIA are pointers to the card's common Card Information Structure (CIS) and each individual function's CIS. The CIS structure is defined in Chapter 16. The CIS includes information on power, function, manufacturer and other things the host needs to determine if the I/O function(s) is appropriate to power-up. If the host determines that the card should be activated, a register in the CCCR area enables the card and each individual function. At this time, all functions of the I/O card are fully available. In addition, the host can control the power consumption and enable/disable interrupts on a function-by-function basis. This access to I/O does not interfere with memory access to the card if present.

Figure 3-2 shows Card Initialization Flow in SD mode and Figure 3-3 shows Card Initialization Flow in SPI mode. UHS-I is not supported in SPI mode. Initialization Flow for an embedded device can be simplified by removing unnecessary checks.
When receiving CMD5 with arg=0, the SDIO card returns an R4 response but does not start initialization of the I/O function. The Version 3.00 SDIO card should start initialization sequence by receiving CMD5 without waiting for CMD5 with WV=0.
Combo Cards can accept CMD15 with RCA=0000, as described in the Physical Layer Specification, but there is an exception for SD memory only cards. Memory only cards require a non-zero RCA before the host may issue CMD15. Thus, CMD15 shall be issued after CMD3 in the Standby state. In the case of ACMD41, it shall accept RCA=0000h.

Application Notes:
As shown in Figure 3-2 and Figure 3-3, an SDIO aware host should send CMD5 arg=0 as part of the initialization sequence after either Power On or a CMD 52 with write to I/O Reset. CMD5 arg=0 is required for using a removal card due to the following reasons:

(1) There was 3.1V-3.5V voltage range legacy card and then whether host power supply can provide the voltage in this range should be checked.
(2) There may be the card which requires CMD5 with WV= 0 to initialize the card.

CMD5 arg=0 may not be necessary for embedded system if the device does not require it.



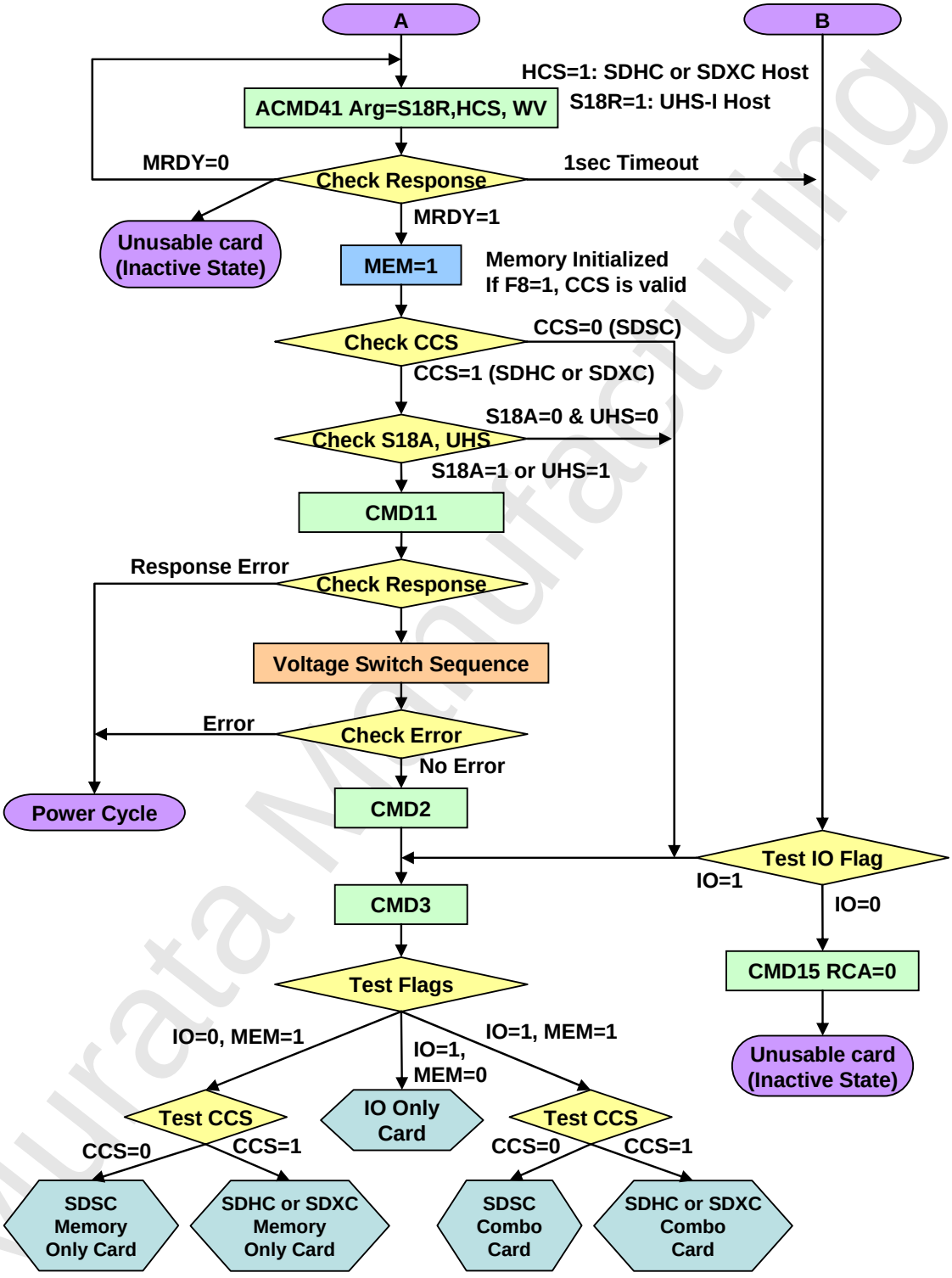
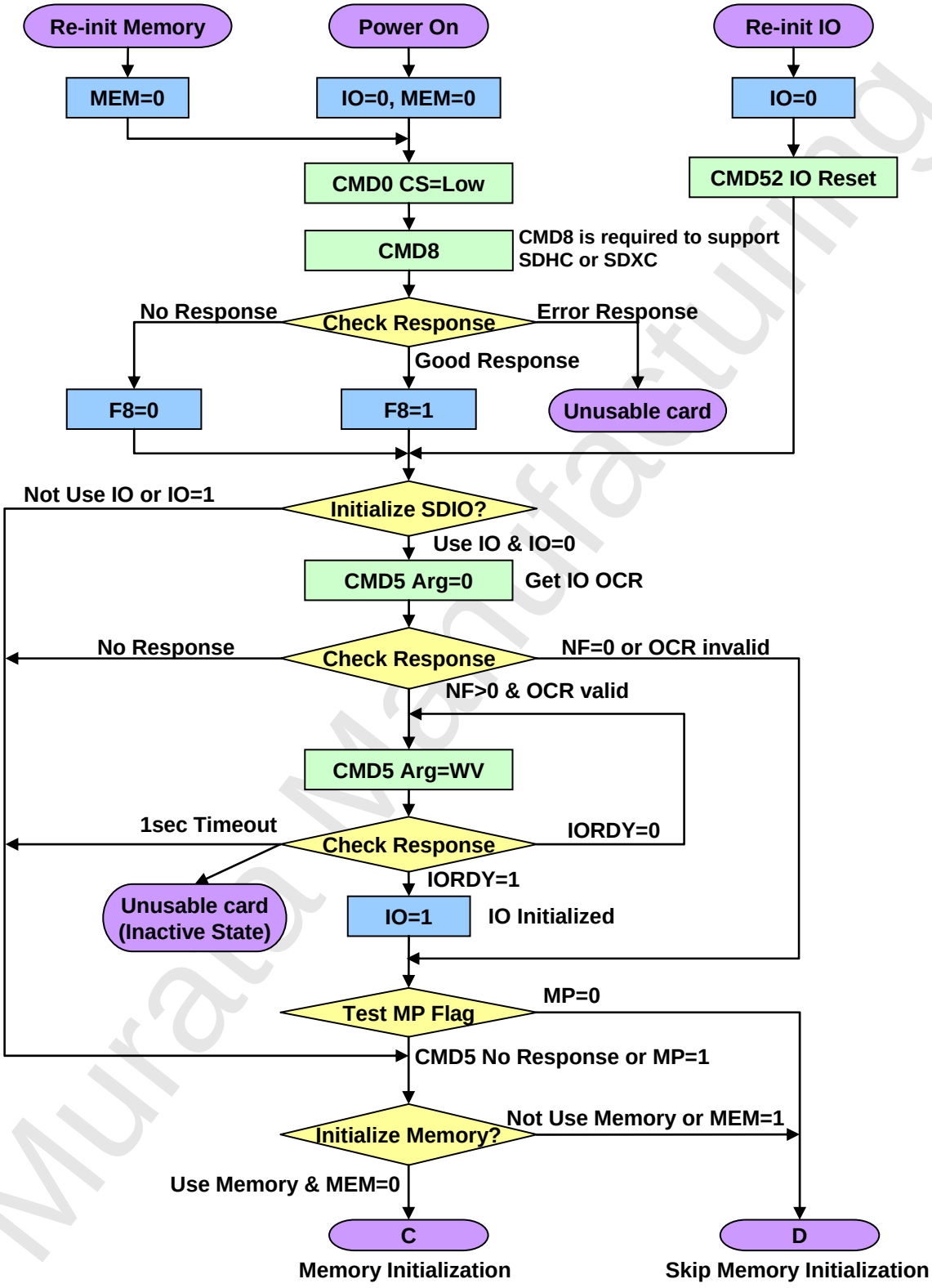


Figure 3-2 : Card Initialization Flow in SD mode (SDIO Aware Host)

Variables	
NF:	Number of I/O Functions (CMD5 Response)
MP:	Memory Present Flag (CMD5 Response)
IORDY:	I/O Power-up Status (C bit in the CMD5 response)
MRDY:	Memory Power-up Status (OCR Bit31)
HCS:	Host Capacity Support (ACMD41 Argument)
CCS:	Card Capacity Status (ACMD41 Response)
S18R:	Switching to 1.8V Request
S18A:	Switching to 1.8V Accepted

Flags	
IO:	I/O Functions Initialized Flag
MEM:	Memory Initialized Flag
F8:	CMD8 Flag
UHS:	SDIO S18A Flag



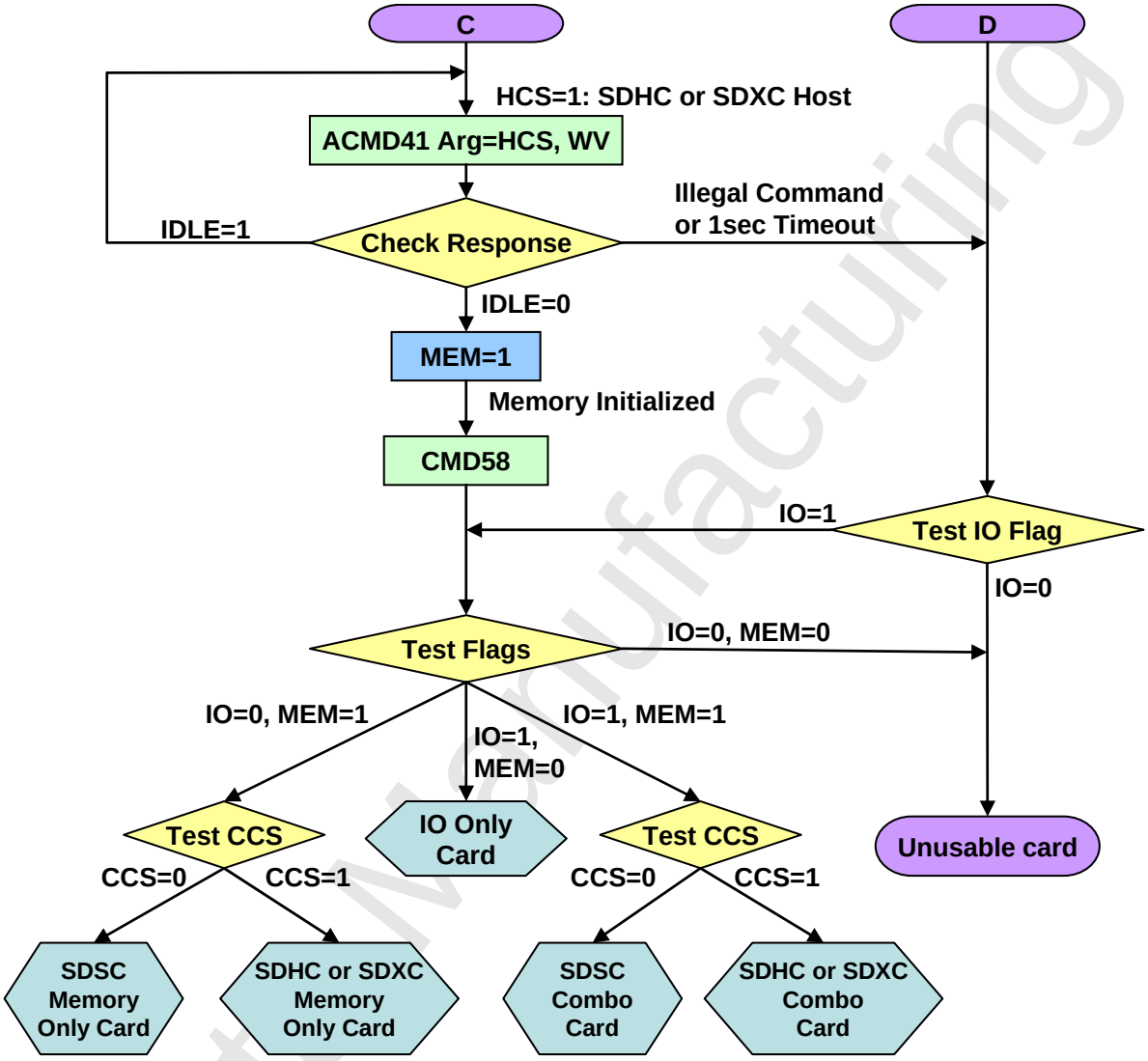


Figure 3-3 : Card initialization flow in SPI mode (SDIO aware host)

Variables	
NF:	Number of I/O Functions (CMD5 Response)
MP:	Memory Present Flag (CMD5 Response)
IORDY:	I/O Power-up Status (C bit in the CMD5 response)
IDLE:	Memory Power-up Status (ACMD41 R1 Response)
HCS:	Host Capacity Support (ACMD41 Argument)
CCS:	Card Capacity Status (ACMD41 Response)

Flags	
IO:	I/O Functions Initialized Flag
MEM:	Memory Initialized Flag
F8:	CMD8 Flag

3.2 The IO_SEND_OP_COND Command (CMD5)

Figure 3-4 : shows the format of the IO_SEND_OP_COND command (CMD5). The function of CMD5 for SDIO cards is similar to the operation of ACMD41 for SD memory cards. It is used to inquire about the voltage range needed by the I/O card. The normal response to CMD5 is R4 in either SD or SPI format. The R4 response in SD mode is shown in Figure 3-5 and the SPI version is shown in Figure 3-6.

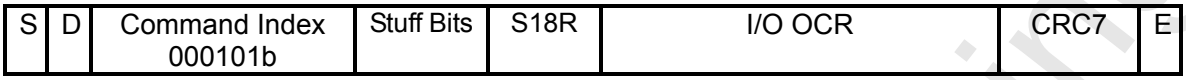


Figure 3-4 : IO_SEND_OP_COND Command (CMD5)

The IO_SEND_OP_COND Command contains the following fields:

S(tart bit): Start bit. Always 0
D(irection): Direction. Always1 indicates transfer from host to card.
Command Index: Identifies the CMD5 command with a value of 000101b
Stuff Bits: Not used, shall be set to 0.
S18R: Switching to 1.8V Request
I/O OCR: Operation Conditions Register. The supported minimum and maximum values for VDD. The layout of the OCR is shown in Table 3-1. Refer to Section 4.10.1 for additional information.
CRC7: 7 bits of CRC data
E(nd bit): End bit, always 1

I/O OCR bit position	VDD Voltage Window (in Volts)
0-3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	2.0-2.1
9	2.1-2.2
10	2.2-2.3
11	2.3-2.4
12	2.4-2.5
13	2.5-2.6
14	2.6-2.7
15	2.7-2.8
16	2.8-2.9
17	2.9-3.0
18	3.0-3.1
19	3.1-3.2
20	3.2-3.3
21	3.3-3.4
22	3.4-3.5
23	3.5-3.6

Table 3-1 : OCR Values for CMD5

The SDIO Version 2.00 cards shall support the operational voltage range 2.7-3.6V and are not necessary to support the voltage range 2.0-2.7V for basic communication. The hosts, which support SDIO Version 2.00, shall not use voltage range 2.0-2.7V for basic communication.

3.3 The IO_SEND_OP_COND Response (R4)

An SDIO card receiving CMD5 shall respond with a SDIO unique response, R4. The format of R4 for both the SD and SPI modes is:

S	D	Reserved	C	Number of I/O functions	Memory Present	Stuff Bits	S18A	I/O OCR	Reserved	E
1	1	6	1	3	1	2	1	24	7	1

Figure 3-5 : Response R4 in SD mode

Modified R1	C	Number of I/O functions	Memory Present	Stuff Bits	I/O OCR
8	1	3	1	3	24

Figure 3-6 : Response R4 in SPI mode

The Response, R4 contains the following data:

S(tart bit):	Start bit. Always 0
D(irection):	Direction. Always 0. Indicates transfer from card to host.
Reserved:	Bits reserved for future use. These bits shall be set to 1.
C:	Set to 1 if Card is ready to operate after initialization
Number of I/O Functions:	Indicates the total number of I/O functions supported by this card. The range is 0-7. Note that the common area present on all I/O cards at Function 0 is not included in this count. The I/O functions shall be implemented sequentially beginning at function 1.
Memory Present:	Set to 1 if the card also contains SD memory. Set to 0 if the card is I/O only.
S18A:	Switching to 1.8V Accepted (Supported in SD mode only)
I/O OCR:	Operation Conditions Register. The supported minimum and maximum values for VDD. The layout of the OCR is shown in Table 3-1. Refer to Section 4.10.1 for additional information.
Modified R1:	The SPI R1 response byte as described in the Physical Layer Specification modified for I/O as described in Figure 3-7.
Stuff Bits	Not used, shall be set to 0.

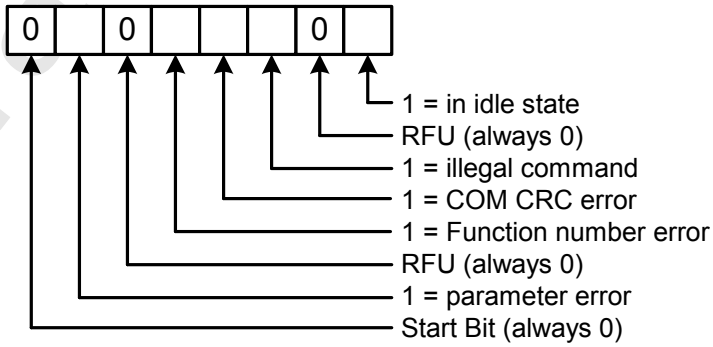


Figure 3-7 : Modified R1 Response

Once an SDIO card has received a CMD5, the I/O portion of that card is enabled to respond normally to all further commands. This I/O enable of the functions within the I/O card shall remain set until a reset, power cycle or CMD52 with write to I/O reset is received by the card. Note that a SD memory only card *may* respond to a CMD5. The proper response for a memory only card would be *Memory Present* = 1 and *Number of I/O Functions* = 0. SD memory card detects the CMD5 as an illegal command and not respond. Note that unlike the similar memory command ACMD41, The SPI response to CMD5 does contain the OCR value from the card.

An I/O aware host will send a CMD5. If the card responds with a response R4 within the timeout value of Ncr as defined in the Physical Layer Specification, the host determines the card's configuration based on the data contained within the R4.

3.4 Special Initialization Considerations for Combo Cards

The host shall be aware of some special situations when initializing a Combo card (SDIO plus SD Memory on the same card). This is because an implementation of the Combo card could actually use two separate controllers (Memory and I/O) in the same package and sharing the same bus lines. It is important for the host to both detect and properly configure both parts (controllers) of a Combo card in order to prevent conflicts between the SDIO and the SD memory controller. These concerns are due to the different responses to a reset (hard or soft) by the two controllers. Another issue is the value of the RCA (Relative Card Address) that exists within the Memory controller.

Note that this consideration is for SD 1-bit and SD 4-bit modes only. In SPI mode, card select/de-select is accomplished using the hardware CS line rather than the RCA.

3.4.1 Re-initialize Both I/O and Memory

When the host re-initializes both the I/O and Memory controllers, it is *strongly* recommended that the host either execute a power reset (power off then on) or issues reset commands to *both* controllers prior to *any* other operation. If the host chooses to use the reset commands, it shall issue CMD52 (I/O Reset) first, because it cannot issue CMD52 after CMD0 (Refer to Section 4.4). After the reset, the host shall re-initialize both the I/O and Memory controller as defined in Figure 3-2.

3.4.2 Using a Combo Card as SDIO only or SD Memory only after Combo Initialization

If a host intends to use only the SDIO or the Memory portion of a Combo Card, it is *strongly* recommended that the host power reset (power off then on) or issues reset commands to *both* controllers prior to *any* other operation. If the host chooses to use the reset commands, it shall issue CMD52 (I/O Reset) first, because it cannot issue CMD52 after CMD0 (Refer to Section 4.4). After the resets, the host re-initializes either the I/O and Memory controller as defined in Figure 3-2.

3.4.3 Acceptable Commands after Initialization

When the host re-initializes a Combo card, the acceptable commands that the host can issue are restricted until the I/O controller is placed into the command state and memory controller enters the transfer state. The prohibited commands are identified in the next section. Combo cards may not work correctly when the host issues these prohibited commands. The proper command sequence for the I/O controller and the memory controller are shown in the following sections. Note that CMD15 (GO_INACTIVE_STATE) can be sent at any time after initialization in order to put any addressed memory controller into the inactive state.

3.4.4 Recommendations for RCA after Reset

The RCA specification was not fully defined in the SDIO Specification Ver1.00. There are two types of cards (SDIO or Combo) with different responses to CMD0 or SDIO reset. The possible responses are:

The card clears RCA to 0000h
The card keeps current RCA value

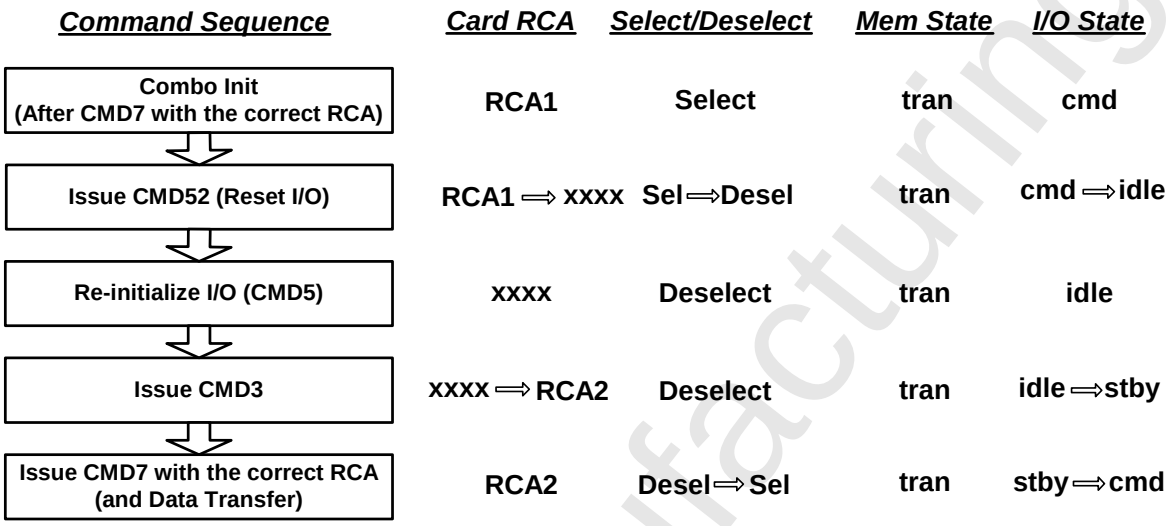


Figure 3-8 : Re-Initialization Flow for I/O Controller

Figure 3-8 : shows the re-initialization flow for the I/O controller of a Combo card. The flow of commands on the left side is matched with the RCA and controller states on the right side. The RCA value of xxxx denotes an RCA value of either 0000h or the prior RCA value. For new controller designs, a reset RCA value of 0000h is recommended. The host shall not issue any commands to the Combo Card except for CMD0, CMD5, CMD3 or CMD7 until the I/O controller has transitioned to the cmd state.

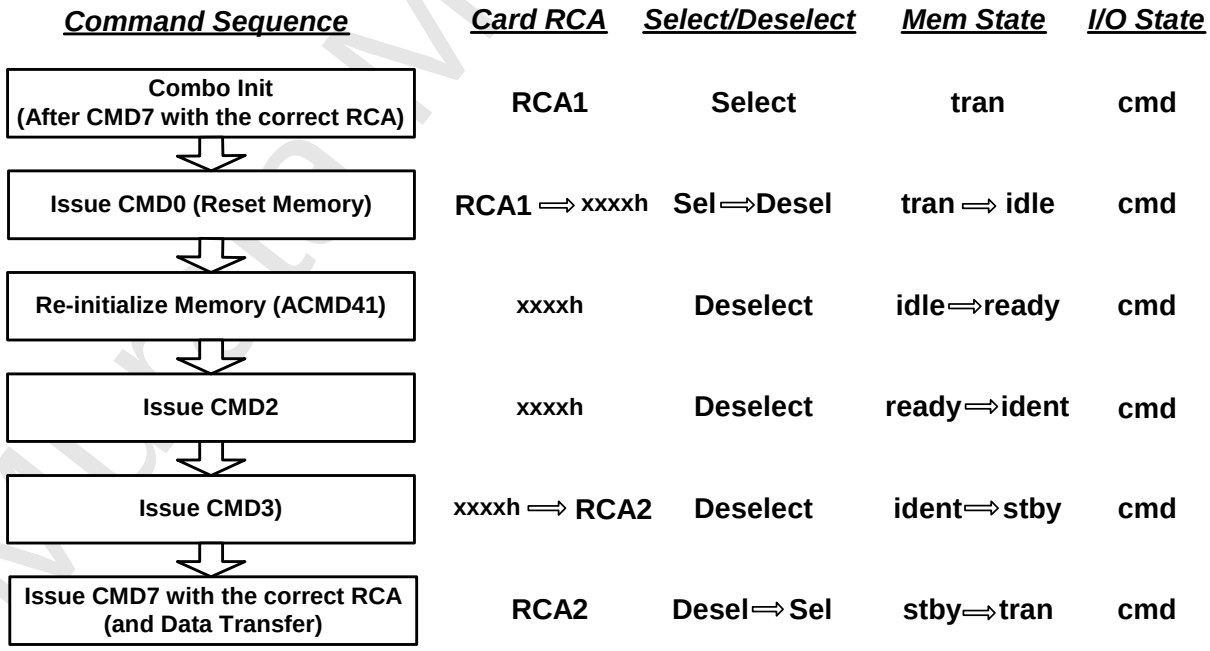


Figure 3-9 : Re-Initialization Flow for Memory controller

Figure 3-9 shows the equivalent command flow to re-initialize the memory controller of a Combo card. The RCA value of xxxxh denotes an RCA value of either 0000h or the prior RCA value. For new controller designs, a reset value of 0000h is recommended. An important fact for the host designer to note is that the host shall not issue any commands except for CMD0, ACMD41 (with RCA=0000h), CMD2, CMD3 or CMD7 to the Combo Card until the memory controller has transitioned to the tran state.

3.4.5 Enabling CRC in SPI Combo Card

When receiving CMD59, Combo cards shall synchronize CRC enable in both SDIO and memory portions of the card. If a host enables CRC using CMD59 and subsequently re-initializes either the I/O or memory controller, the CRC for that controller will be off by default and the host shall issue a CMD59 to re-enable CRC. When CMD59 is received, Combo Cards return the R1 response token while SDIO only cards return the modified R1 response token.

4. Differences with SD Memory Specification

4.1 SDIO Command List

Table C- 1 shows the list of commands accepted by SD memory and SDIO cards when using the SD bus interface. Table C- 2 shows the list of commands accepted by SD memory and SDIO cards when using the SPI bus interface.

4.2 Memory Commands and SDIO Commands

4.2.1 Supported SD Memory Commands

SD Memory Command	SDIO Command	Comment
CMD0	CMD0	The first CMD0 is used to select either SD mode or SPI mode. If CMD0 is not received, SDIO Card shall be started in SD mode.
CMD11	CMD11	This is a common command between the memory and SDIO. A signal voltage switch sequence is started by this command to enter UHS-I mode.
CMD19	CMD19	This is a common command between memory and SDIO. This command is effective in 1.8V signaling level, otherwise it is treated as an illegal command. The tuning block is sent to the host so that host can control a suitable sampling position.

Refer to the Physical Layer Specification Version 3.0x regarding CMD11 and CMD19 functions.

Table 4-1 : Supported SD Memory Commands

4.2.2 Unsupported SD Memory Commands

Several commands required for SD Memory cards are not supported by either SDIO-only cards or the I/O portion of Combo cards. Some of these commands have no use in SDIO cards such as Erase commands and thus are not supported in SDIO. In addition, there are several commands for SD memory cards that have different commands when used with the SDIO section of a card. Table 4-2 lists these SD Memory commands and the equivalent SDIO commands. For a complete list of supported and unsupported commands, refer to Table C- 1 and Table C- 2.

SD Memory Command	SDIO Command	Comment
CMD0	CMD52 (write to I/O reset in CCCR)	The reset command (CMD0) is only used for memory or the memory portion of Combo cards. An I/O only card or the I/O portion of a combo card is **not** reset with CMD0 but reset with I/O Reset (by writing 1 to the RES bit (bit 3 of address 6 in the CCCR) by CMD52). I/O Reset is invoked after the response of CMD52.
CMD12	CMD52 (write to I/O abort)	In order to abort a block transfer of data, SD memory cards use CMD12. In order to abort an I/O transaction, use CMD52 to write to the abort register in the CCCR (bits 2:0 of register 6). Refer to Section 4.8 for details.
CMD16	CMD52 (write to I/O Block Length)	CMD16 sets the block length for SD memory. In order to set the block length for each I/O function, use CMD52 to write the block length in the FBR
CMD2	NONE	The CID register does not exist in an SDIO only card
CMD4	NONE	The DSR register does not exist in an SDIO only card
CMD9	NONE	The CSD register does not exist in an SDIO only card

SD Memory Command	SDIO Command	Comment
CMD10	NONE	The CID register does not exist in an SDIO only card
CMD13	NONE	An SDIO only card or the I/O portion of a combo card does not support the same SEND_STATUS (CMD13) protocol the SD memory uses. Refer to Section 4.10.8.
ACMD6	CMD52 (write to Bus_Width [1:0] in CCCR)	SET_BUS_WIDTH is handled by a write to the CCCR. Refer to Section 4.4 for details.
ACMD13	NONE	The SD Status register does not exist in an SDIO only card
ACMD41	CMD5	SDIO cards and hosts use the IO_SEND_OP_COND Command (CMD5). Refer to Section 3.2
ACMD42	CMD52	In the SD mode, the pull-up resistor on DAT[3] is controlled by writing to the CD Disable bit in the CCCR. For Combo Cards, this resistor is enabled unless both the memory and the I/O control registers are set to disable the resistor. Refer to Section 4.6 for details.
ACMD51	NONE	The SCR register does not exist in an SDIO only card
CMD17, CMD18, CMD24, CMD25	CMD53	I/O block operations use CMD53, rather than memory block read/write commands.

Table 4-2 : Unsupported SD Memory Commands

4.3 Modified R6 Response

The normal response to CMD3 by a memory card is R6 as shown in Table 4-3 :. The *card status* bits (23-8) are changed when CMD3 is sent to an I/O only card. In this case, the 16 bits of response shall be the SDIO-only values shown in Table 4-4

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width (bits)	1	1	6	16	16	7	1
Value	'0'	'0'	X	X	X	X	'1'
Description	Start bit	Direction bit	Command index ('000011')	New published RCA [31:16] of the card	[15:0] *Card status* (Refer to Table 4-4)	CRC7	end bit

Table 4-3 : R6 response to CMD3

Bits	Identifier	Type	Value	Description	Clear Condition
15	COM_CRC_ERROR	E R	'0'= no error '1'= error	The CRC check of the previous command failed	B
14	ILLEGAL_COMMAND	E R	'0'= no error '1'= error	Command not legal for the card state	B
13	ERROR	E R X	'0'= no error '1'= error	A general or an unknown error occurred during the operation	C
12: 0	***Undefined. Should read as 0 for SDIO only cards. Host should ignore these bits.***				

Note: Please refer to Section 7.3.4 of the Physical Layer Specification for explanation of the entries in the *Type* and *Clear Condition* columns.

Table 4-4 : SDIO R6 Status Bits

4.4 Reset for SDIO

In order to reset all functions within an SDIO card or the SDIO portion of a combo card, a method different than that used for SD memory is defined. The reset command (CMD0) is only used for memory or the memory portion of Combo cards. In order to reset an I/O only card or the I/O portion of a combo card, use CMD52 to write a 1 to the RES bit in the CCCR (bit 3 of register 6). Note that in the SD mode, CMD0 is only used to indicate entry into SPI mode and shall be supported. An I/O only card or the I/O portion of a combo card is **not** reset by CMD0.

4.5 Bus Width

For an SD memory card, the bus width for SD mode is set using ACMD6. For an SDIO card a write to the CCCR using CMD52 is used to select bus width. In the case of a combo card, both selection methods exist. In this case, the host shall set the bus width in both locations by issuing both the ACMD6 and the CCCR write using CMD52 with the same width before starting any data transfers. For details on changing the bus for an SDIO card, refer to Table 6-2. For a Combo Card, changing bus width is handled as shown in Table 4-5.

I/O	Memory	Control Method
Initialized	Not Initialized	CCCR
Not Initialized	Initialized	ACMD6
Initialized	Initialized	CCCR & ACMD6

Table 4-5 : Combo Card 4-bit Control

As shown in Table 4-5, if only the I/O function of a combo card is active, writing to the CCCR is all that is required to change the bus width mode. If only memory is active then ACMD6 is all that is needed to change bus widths. If both I/O and Memory are active then both CCCR and ACMD6 are needed to change the bus width. In the combo card, both the memory and I/O controllers shall be set to the same bus width

Note that Low-Speed SDIO cards support 4-bit transfer as an option. When communicating with a Low-Speed SDIO card, the host shall first determine if the card supports 4-bit transfer prior to attempting to select that mode.

If a Combo card supports the lock/unlock operation, it cannot change bus width of a locked card and returns an illegal command error to a bus width switch command. The host needs to unlock the card by CMD42 before changing bus width. This also implies that the host should not change bus width during initialization before managing a locked card.

Only 4-bit bus mode is supported in UHS-I except CMD42 (As unlocking is required before changing 4-bit mode, CMD42 sends a data block in 1-bit mode). UHS-I operating in 1-bit mode is not assured.

4.6 Card Detect Resistor

SD memory and I/O cards use a pull-up resistor on DAT[3] to detect card insertion. The procedure to enable/disable this resistor is different between SD memory and SDIO. SD memory uses ACMD42 to control this resistor while SDIO uses writes to the CCCR using CMD52. In the case of a combo card, both control locations exist and shall be managed by the host. For a combo card, the resistor is enabled only when both the memory and the I/O control registers have the resistor enabled. That is, after a power on, the host shall disable the resistor by sending ACMD42 to the memory controller or a CCCR write to the SDIO controller since the resistor enable is a logical AND of the two enables. Table 4-6 shows the effect of each resistor enable on the card's resistor. After power-up, both locations default to resistor enabled. Note

that after an I/O reset, the I/O resistor enable is not changed. Also note that SDIO Specification Version 1.00 required that both the SDIO and Memory resistor be disabled in order for the resistor to actually be disabled (logical OR of the 2 enables). Combo cards built to that specification require the host to disable both enables. It is recommended the host disable both enables of any combo card to avoid problems with the difference between 1.0 and current specification based cards.

I/O Resistor	Memory Resistor	Card Resistor
Enabled	Enabled	Resistor Connected
Enabled	Disabled	Resistor Disconnected
Disabled	Enabled	Resistor Disconnected
Disabled	Disabled	Resistor Disconnected

Table 4-6 : Card Detect Resistor States

4.7 Timings

All timing diagrams in this specification use the following schematics and abbreviations:

Symbol	Data Line State
S	Start bit (= '0')
E	End bit (='1')
Z	High impedance state (-> = '1')
P	One-cycle pull-up (= '1')
X	Don't Care data bits (from card)
*	Repetition
D	Data bit
L	Drive low
H	Drive high

Table 4-7 : Timing Diagram Symbols

In general, P-bits are less sensitive to noise because they are actively driven to logic '1' by the card or the host output driver exclusively. Z-bits by contrast, are only weakly pulled-up to logic '1', thus it is possible to replace the sequences as follows:

- "EP*" with "EZ*"
- "EZ*P*" with "EZ*Z*"

A Z-bit after S-bit or D='0' is not allowed.

4.8 Data Transfer Block Sizes

SDIO cards may transfer data in either a multi-byte (1 to 512 bytes) or an optional block format, while the SD memory cards are fixed in the block transfer mode. The Physical Layer Specification limits the block size for data transfer to powers of 2 (i.e. 512, 1024, 2048) unless using partial read and write. The SDIO Specification allows any block size from 1 byte to 2048 bytes in order to accommodate the various natural block sizes for I/O functions. Note that an SDIO card function may define a maximum block size or byte count in the CIS that is smaller than the maximum values described above.

4.9 Data Transfer Abort

A host communicating with a SD memory card uses CMD12 to abort the transfer of read or write data to/from the card. For an SDIO card, CMD12 abort is replaced by a write to the ASx bits in the CCCR. Normally, the abort is used to stop an infinite block transfer (block count=0). If an exact number of blocks are to be transferred, it is recommended that the host issue a block command with the correct block count, rather than using an infinite count and aborting the data at the correct time.

4.9.1 Read Abort

The host may issue an I/O abort by writing to the CCCR at any time during I/O extended read operation. The data transmission stops 2 clocks cycles after the end bit of the I/O abort command, even if the card has already begun transferring an unwanted data block while the host is issuing the abort.

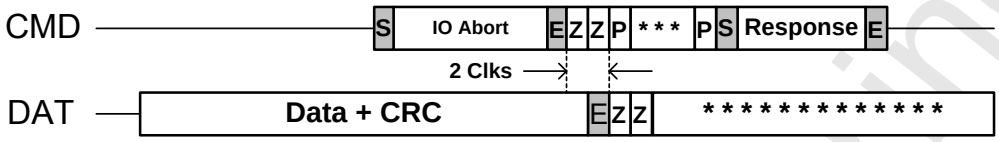
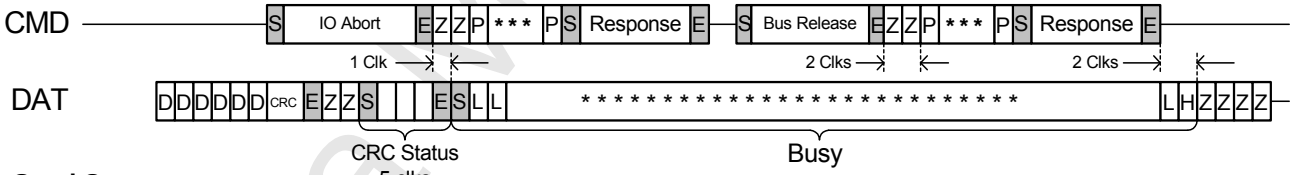


Figure 4-1 : I/O Abort during Read Data Transfer

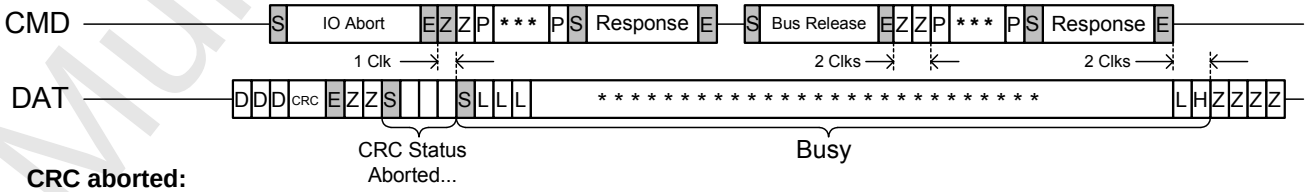
4.9.2 Write Abort

The host may issue an I/O abort by writing to the CCCR at any time between data blocks during I/O extended write operation. In this case, the final block transfer (including the CRC response from the card) shall be completed. This requires that the end bit of the I/O abort command appear a maximum of two clocks before the end bit of the CRC response to the last data block. Note that the I/O abort command may be sent any time after the CRC response to the last data block. The host shall not abort in the middle of a write block. After the I/O abort is sent to the card, the card signals 'Busy' (by pulling DAT[0] line to '0') until it has finished processing the last transferred data block. During that Busy period, the host may release the bus by writing to the CCCR BR bit. There exist some special cases when the abort is issued near the end of the CRC response to a write multiple-block command. Figure 4-2 shows the case where the abort occurs after the CRC end bit (1 clock delayed). In this case, the CRC response is valid and the card accepts the prior block of write data. Figure 4-3 shows the illegal case where the abort is sent prior to the CRC response completion. In this case, the CRC response is invalid and the last block of data is not written to the card. Figure 4-4 shows the case where the abort is issued after the end of the CRC response.



Good Case:
-Good CRC response accepted by host
-Data accepted by card

Figure 4-2 : I/O Abort during Write CRC Response (Good)



CRC aborted:
-Partial CRC response accepted by host
-Data was not accepted by card

Figure 4-3 : I/O Abort during Write CRC Response (Bad)

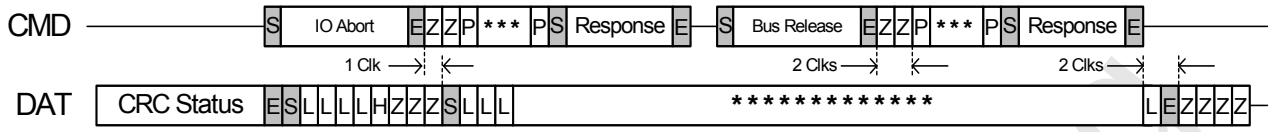


Figure 4-4 : I/O Abort after Write CRC Response

4.10 Changes to SD Memory Fixed Registers

The Physical Layer Specification defines 7 fixed card registers. They are:

1. OCR Register (32 bits)
2. CID Register (128 bits)
3. CSD Register (128 bits)
4. RCA Register (16 bits)
5. DSR Register (16 bits, optional)
6. SCR Register (64 bits)
7. SD_CARD_STATUS (512 bits)

In addition, within an SD memory card there is a status register whose value is returned to the host in the form of several responses (i.e. the R1b response). An SDIO only card eliminates some registers and changes some of the bits in the remaining registers. The description of these register changes follows:

4.10.1 OCR Register

All SD cards (memory, I/O and combo) shall have at least one OCR register. If the card is a combo card, it *may* have two OCR's (one for memory and one for I/O). The memory portion of a combo card has an OCR accessed using ACMD41 and CMD58. The I/O portion of a card has an OCR with the same structure that is accessed via CMD5. If there are multiple OCR's the voltage range may not be identical. Some I/O functions may have a wider VDD range than that reflected in the I/O OCR register. The I/O OCR shall be the logical AND of the voltage ranges(s) of all I/O functions. Note that the I/O OCR format is different from the memory version in that it is only 24 bits long. For details, refer to Table 3-1. The per-function voltage for each I/O function can be read in the CIS for the card.

4.10.2 CID Register

There shall be a *maximum* of one CID register per SD card. If the card contains both memory and I/O, the CID register information is unchanged from the Physical Layer Specification Version 1.01 and reflects the information from the memory portion of the card. If the card is I/O only, the CID register and the associated access command (CMD10) are **not** supported. If the host attempts to access this register in an I/O only card, a card in SPI mode shall respond with an "Invalid Command" error response and a card in SD mode shall not respond.

4.10.3 CSD Register

There shall be a *maximum* of one CSD register per SD card. If the card contains both memory and I/O, the CSD register information is unchanged from the Physical Layer Specification Version 1.01 and reflects the information from the memory portion of the card. If the card is I/O only, the CSD register and the associated access command (CMD9) are **not** supported. If the host attempts to access this register in an I/O only card, a card in SPI mode shall respond with an "Invalid Command" error response and a card in SD mode shall not respond.

4.10.4 RCA Register

There shall only be one RCA register per SD card. The RCA value shall apply to the card as a whole. All functions and any memory share the same card address.

4.10.5 DSR Register

SDIO only cards do not support the DSR register. In the case of combo cards, support is optional as defined in the Physical Layer Specification.

4.10.6 SCR Register

There shall be a *maximum* of one SCR register per SD card. If the card contains both memory and I/O, the SCR register information is unchanged from the Physical Layer Specification Version 1.01 and reflects the information from the memory portion of the card. If the card is I/O only, the SCR register and the associated access command (ACMD51) are **not** supported. If the host attempts to access this register in an I/O only card, a card in SPI mode shall respond with an "Invalid Command" error response and a card in SD mode shall not respond.

4.10.7 SD Status

There shall be a *maximum* of one SD Status register per SD card. If the card contains both memory and I/O, the SD Status register information is unchanged from the Physical Layer Specification Version 1.01 and reflects the information from the memory portion of the card. If the card is I/O only, the SD Status register and the associated access command (ACMD13) are **not** supported. If the host attempts to access this register in an I/O only card, a card in SPI mode shall respond with an "Invalid Command" error response and a card in SD mode shall not respond.

4.10.8 Card Status Register

The structure of the SDIO status register is shown in Table 4-8. For SDIO specific operations in the SD mode that return the card status register contents (i.e. the response to CMD7), some bits are not applicable to I/O operations and shall be returned as 0. These unused bits are identified as type N/A. For combo cards, the values returned shall reflect the memory status. The CURRENT_STATE bits (12:9) shall reflect the memory Controller State. For an I/O only card, the unused bits are 0 and the Current_State bits (12:9) shall be Fh (15) to identify it as an I/O only response.

I/O specific status is reported by I/O response and Memory specific status is reported by Memory response except for the following case: In the SD bus mode, the card shall not respond to an Illegal Command or a command with a CRC error. The indication of those two error cases shall be given by the card in the following command's response. This is true for an I/O only card as well as for combo cards, even in cases where the erroneous command and the command that follows are not targeting the same card module (Memory or I/O).

Bit	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ERX	'0'= no error '1'= error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	N/A	0	Not used with SDIO operation	C
29	BLOCK_LEN_ERROR	N/A	0	Not used with SDIO operation	C
28	ERASE_SEQ_ERROR	N/A	0	Not used with SDIO operation	C
27	ERASE_PARAM	N/A	0	Not used with SDIO operation	C
26	WP_VIOLATION	N/A	0	Not used with SDIO operation	C
25	CARD_IS_LOCKED	N/A	0	Not used with SDIO operation	C
24	LOCK_UNLOCK_FAILED	N/A	0	Not used with SDIO operation	C
23	COM_CRC_ERROR	ER	'0'= no error '1'= error	The CRC check of the previous command failed. (Note 1)	B
22	ILLEGAL_COMMAND	ER	'0'= no error '1'= error	Previous command not legal for the card state. (Note 2)	B
21	CARD_ECC_FAILED	N/A	0	Not used with SDIO operation	C
20	CC_ERROR	N/A	0	Not used with SDIO operation	C
19	ERROR	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	UNDERRUN	N/A	0	Not used with SDIO operation	C
17	OVERRUN	N/A	0	Not used with SDIO operation	C
16	CID/ CSD_OVERWRITE	N/A	0	Not used with SDIO operation	C
15	WP_ERASE_SKIP	N/A	0	Not used with SDIO operation	C
14	CARD_ECC_DISABLED	N/A	0	Not used with SDIO operation	C
13	ERASE_RESET	N/A	0	Not used with SDIO operation	C
12:9	CURRENT_STATE	SX	15=I/O only	For an I/O only card, the current state shall be fixed at a value of 0Fh. This indicates that it is an I/O only card and the normal memory states do not apply	B
8	READY_FOR_DATA	N/A	0	Not used with SDIO operation	C
7:6	***Reserved***				
5	APP_CMD	N/A	0	CMD55 not used in SDIO operation	C
4	***Reserved***				
3	AKE_SEQ_ERROR	N/A	0	Not used with SDIO operation	C
2	***Reserved for application specific commands***				
1, 0	***Reserved for manufacturer test mode***				

Table 4-8 : SDIO Status Register Structure

Note 1: In the SPI mode, if the card detects a CRC error, it returns a com CRC error in the R1 response immediately following the command (Refer to Figure 3-7). In this situation, the note that the CRC error is for the previous command does not apply.

Note 2: In the SPI mode, if the card detects an Illegal Command, it returns an Illegal Command error in the R1 response immediately following the command (Refer to Figure 3-7). In this situation, the note that the Illegal Command error is for the previous command does not apply.

5. New I/O Read/Write Commands

Two additional data transfer instructions have been added to support I/O. IO_RW_DIRECT and IO_RW_EXTENDED, which allows fast access with byte or block addresses. Both commands are in class 9 (I/O Commands).

5.1 IO_RW_DIRECT Command (CMD52)

The IO_RW_DIRECT is the simplest means to access a single register within the total 128K of register space in any I/O function, including the common I/O area (CIA). This command reads or writes 1 byte using only 1 command/response pair. A common use is to initialize registers or monitor status values for I/O functions. This command is the fastest means to read or write single I/O registers, as it requires only a single command/response pair.

S	D	Command Index 110100b	R/W flag	Function Number	RAW flag	Stuff	Register Address	Stuff	Write Data or Stuff Bits	CRC7	E
1	1	6	1	3	1	1	17	1	8	7	1

Figure 5-1 : IO_RW_DIRECT Command

The IO_RW_DIRECT Command contains the following fields:

S(tart bit): Start bit. Always 0

D(irection): Direction. Always1 indicates transfer host to card.

Command Index: Identifies the "IO_RW_DIRECT" command with a value of 110100b

R/W Flag: This bit determines the direction of the I/O operation. If this bit is 0, this command shall read data from the SDIO card at the address specified by the Function Number and the Register Address to the host. The data byte is returned in the response, R5. If this bit is set to 1, the command shall write the bytes in the Write Data field to the I/O location addressed by the Function Number and the Register Address. If the RAW flag is 0, then the data in the register that was written shall be read and that value returned in the response.

RAW Flag: The Read after Write flag. If this bit is set to 1 and the R/W flag is set to 1, then the command shall read the value of the register after the write. This is useful to allow writing to a control register and reading the status at the same address. If this bit is cleared, the value returned in the R5 response shall be the same as the write data in the command. If this bit is set, the data field of the R5 response shall contain the value read from the addressed register after the write operation.

Function Number: The number of the function within the I/O card you wish to read or write. Note that function 0 selects the common I/O area (CIA).

Register Address: This is the address of the byte of data inside of the selected function to read or write. There are 17 bits of address available so the register is located within the first 128K (131,072) addresses of that function.

Write Data/Stuff Bits: For a direct write command (R/W=1), this is the byte that is written to the selected address. For a direct read (R/W=0), this field is not used and shall be set to 0.

CRC7: 7 bits of CRC data

E(nd bit): End bit, always 1

5.2 IO_RW_DIRECT Response (R5)

The SDIO card's response to CMD52 shall be in one of two formats. If the communication between the card and host is in the 1-bit or 4-bit SD mode, the response shall be in a 48-bit response (R5) as described in Section 5.2.1. If the communication is using the SPI mode, the response shall be a 16-bit R5 response as described in Section 5.2.2.

5.2.1 CMD52 Response (SD Modes)

The SDIO card's response to CMD52 in the SD mode is shown in Figure 5-2. If the operation was a read command, the data being read is returned as an 8-bit value. In addition, 15 bits of status information is returned. The format of the SD response is as follows:

S	D	Command Index 110100b	Stuff	Response Flags Bit 7----------0	Read or Write Data	CRC7	E
1	1	6	16	8	8	7	1

Figure 5-2 : R5 IO_RW_DIRECT Response (SD Modes)

The IO_RW_DIRECT response (R5) contains the following fields:

S(tart bit): Start bit. Always 0
D(irection): Direction. 0 indicates transfer card to host (Response)
Command Index: Identifies the "IO_RW_DIRECT" command with a value of 110100b
Stuff Bits: Not used, shall be set to 0
Response Flags: 8 Bits of flag data indicating the status of the SDIO card. Table 5-1 shows the format of these flag bits.
Read or Write Data: For an I/O write (R/W=1) with the RAW Flag set (RAW=1) this field shall contain the value read from the addressed register after the write of the data contained in the command. Note that in this case, the read-back data may not be the same as the data written to the register, depending on the design of the hardware. For an I/O write with the RAW bit=0, the SDIO function shall not do a read after write operation, and the data in this field shall be identical to the data byte in the write command. For an I/O read (R/W=0), the actual value read from that I/O location is returned in this field.
CRC7: 7 bits of CRC data
E(nd bit): End bit, always 1

Bits	Identifier	Type	Value	Description	Clear Condition
7	COM_CRC_ERROR	E R	'0'= no error '1'= error	The CRC check of the previous command failed.	B
6	ILLEGAL_COMMAND	E R	'0'= no error '1'= error	Command not legal for the card State.	B

Bits	Identifier	Type	Value	Description	Clear Condition
5-4	IO_CURRENT_STATE	S	00=DIS 01=CMD 02=TRN 03=RFU	**DIS=Disabled:** Initialize, Standby and Inactive States (card not selected) **CMD=DAT lines free:** 1. Command waiting (No transaction suspended) 2. Command waiting (All CMD53 transactions suspended) 3. Executing CMD52 in CMD State **TRN=Transfer:** Command executing with data transfer using DAT[0] or DAT[3:0] lines	B
3	ERROR	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
2	RFU	--	Fixed at 0	Reserved for Future Use	C
1	FUNCTION_NUMBER	ER	'0'= no error '1'= error	An invalid function number was requested	C
0	OUT_OF_RANGE	ERX	'0'= no error '1'= error	ER: The command's argument was out of the allowed range for this card. EX: Out of range occurs during execution of CMD53.	C

Table 5-1 : Flag Data for IO_RW_DIRECT SD Response

5.2.2 R5, IO_RW_DIRECT Response (SPI mode)

The SDIO card's response to CMD52 in the SPI mode is shown in Figure 5-3. If the operation was a read command, the data being read is returned as an 8-bit value. In addition, 8 bits of status information is returned in a SPI R1 response byte as described in Figure 7-9 of the Physical Layer Specification Version 3.0x modified for SDIO as shown in Figure 5-3.

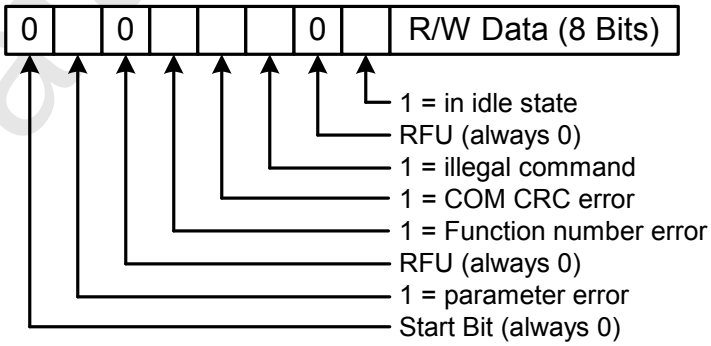


Figure 5-3 : IO_RW_DIRECT Response in SPI Mode

Note the read/write (R/W) data is identical to the read/write data described for the SD R5 response (Refer to Section 5.2.1). Parameter error status in SPI mode corresponds to OUT_OF_RANGE and ERROR in

the SD mode response. In the case of CMD53, Data Error Token should also be used to indicate OUT_OF_RANGE and ERROR.

5.3 IO_RW_EXTENDED Command (CMD53)

In order to read and write multiple I/O registers with a single command, a new command, IO_RW_EXTENDED is defined. This command is included in command class 9 (I/O Commands). This command allows the reading or writing of a large number of I/O registers with a single command. Since this is a data transfer command, it provides the highest possible transfer rate.

S	D	Command Index 110101b	R/W flag	Function Number	Block Mode	OP Code	Register Address	Byte/Block Count	CRC7	E
1	1	6	1	3	1	1	17	9	7	1

Figure 5-4 : IO_RW_EXTENDED Command

The IO_RW_EXTENDED Command contains the following fields:

S(tart bit): Start bit. Always 0

D(irection): Direction. Always1 indicates transfer host to card.

Command Index: Identifies the "IO_RW_EXTENDED" command with a value of 110101b

R/W Flag: This bit determines the direction of the I/O operation. If this bit is 0, this command reads data from the SDIO card at the address specified by the Function Number and the Register Address to the host. The read data shall be returned on the DAT[x] lines. If this bit is set to 1, the command shall write the bytes from the DAT[x] lines to the I/O location addressed by the Function Number and the Register Address.

Function Number: The number of the function within the I/O card you wish to read or write. Note that function 00h selects the common I/O area (CIA).

Block Mode (Optional) this bit, if set to 1, indicates that the read or write operation shall be performed on a block basis, rather than the normal byte basis. If this bit is set, the Byte/Block count value shall contain the number of blocks to be read/written. The block size for functions 1-7 is set by writing the block size to the I/O block size register in the FBR (Refer to Table 6-3 and Table 6-4). The block size for function 0 is set by writing to the FN0 Block Size register in the CCCR. Card and host support of the block I/O mode is optional. The host can determine if a card supports block I/O by reading the *Card supports MBIO* bit (SMB) in the CCCR (Refer to Table 6-2). The block size used when Block Mode = 1 and the maximum byte count per command used when Block Mode = 0 can be read from the CIS in the tuple TPLFE_MAX_BLK_SIZE (Refer to Section 16.7.4) on a per-function basis.

OP code Defines the read/write operation as described in Table 5-2

OP code	Command operation
0	Multi byte R/W to fixed address
1	Multi byte R/W to incrementing address

Table 5-2 : IO_RW_ EXTENDED command Op Code Definition

- OP Code 0 is used to read or write multiple bytes of data to/from a single I/O register address. This command is useful when I/O data is transferred using a FIFO inside of the I/O card. In this case, multiple bytes of data are transferred to/from a single register address. For this operation, the address of the register is set into the Register Address field. Data is transferred on the DAT[0] or DAT[3:0] lines as defined for SD memory cards.

- OP Code 1 is used to read or write multiple bytes of data to/from an I/O register address that increment by 1 after each operation. This command is used when large amounts of I/O data exist within the I/O card in a RAM like data buffer. In this operation, the start address is loaded into the Register Address field. The first operation occurs at that address within the I/O card. The next operation shall occur at address+1 with the address incrementing by 1 until the operation has completed. As with OP Code 0, the number of bytes is set in the Byte Count field of the command.

Register Address: Start Address of I/O register to read or write. Range is [1FFFFh:0]

Byte/Block Count If the command is operating on bytes (Block Mode = 0), this field contains the number of bytes to read or write. A value of 000h shall cause 512 bytes to be read or written.

Count Value	000h	001h	002h ---- 1FFh
Bytes Transferred	512	1	2 511
Block Transferred	infinite	1	2 511

Table 5-3 : Byte Count Values

If the command is in block mode (Block Mode=1), the Block Count field specifies the number of Data Blocks to be transferred following this command. A value of 000h indicates that the count set to infinite. In this case, the I/O blocks shall be transferred until the operation is aborted by writing to the I/O abort function select bits (ASx) in the CCCR (Refer to Table 6-1 and Table 6-2). Table 5-3 shows the relationship between the value in the command and the actual number of bytes transferred.

CRC7: 7 bits of CRC data
E(nd bit): End bit, always 1

The response from the SDIO card to CMD53 shall be R5 (the same as CMD52) as defined in 5.2. For CMD53, the 8-bit data field shall be stuff bits and shall be read as 00h. Also, the ERROR response bit shall be type "E R X" (Refer to Table 5-1).

5.3.1 CMD53 Data Transfer Format

When executing the IO_RW_EXTENDED (CMD53), the multi-byte or multi-block data transfer is similar to the data transfer for memory. For the multi-byte transfer modes (block mode=0) the following applies:
IO_RW_EXTENDED byte read is similar to CMD17 (READ_SINGLE_BLOCK)
IO_RW_EXTENDED byte write is similar to CMD24 (WRITE_BLOCK)
Note that the byte count for this transfer is set in the command, rather than the fixed block size. Thus, the size of the data payload is in the range of 1-512 bytes. The block mode is similar to the following memory commands:
IO_RW_EXTENDED block read is similar to CMD18 (READ_MULTIPLE_BLOCK)
IO_RW_EXTENDED block write is similar to CMD25 (WRITE_MULTIPLE_BLOCK)
For the block mode the only difference is that for a fixed block count, the host does not need to stop the transfer, as it continues until the block count is satisfied. If the block count is set to zero, the operation is identical to the memory mode in that the host must stop the transfer.

5.3.2 Special Timing for CMD53 Multi-Block Read

The optional read multi-block operation using CMD53 has a special timing requirement for the SDIO card. In order to allow the host to stall the read data of the first block of read data; the transfer of data from the card to the host shall not start until after the end bit of the response. Figure 5-5 shows the timing for the first block of read data.

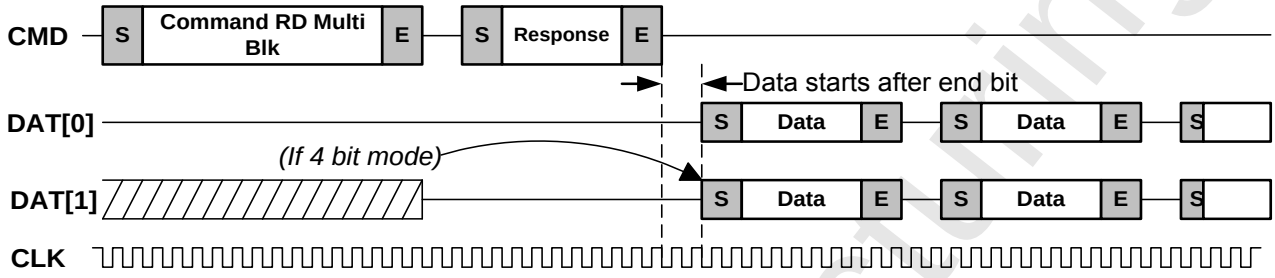


Figure 5-5 : Multi-Block Read Timing

6. SDIO Card Internal Operation

I/O access differs from memory in that the registers can be written and read individually and directly without a FAT file structure or the concept of blocks (although block access is supported). These registers allow access to the I/O data, control of the I/O function and report on status or transfer I/O data to/from the host. SD memory relies on the concept of a fixed block length with commands reading/writing multiples of these fixed size blocks. I/O may or may not have fixed block lengths and the read size may be different from the write size. Because of this, I/O operations may be based on either a length (byte count) or a block size.

6.1 Overview

Each SDIO card may have from 1 to 7 functions plus one memory function. A function is a self contained I/O device. I/O functions may be identical or completely different from each other. All I/O functions are organized as a collection of registers. There is a maximum of 131,072 (2^{17}) registers possible for each I/O function. These registers and their individual bits may be read Only (RO), Write Only (WO) or Read/Write (R/W). These registers can be 8, 16 or 32 bits wide within the card. All addressing is based on byte access. These registers can be written and/or read one at a time, multiply to the same address or multiply to an incrementing address. The single R/W access is often used to initialize the I/O function or to read a single status or data value. The multiple reads to a fixed address are used to read or write data from a data FIFO register in the card. The read to incrementing addresses is used to read or write a collection of data to/from a RAM area inside of the card. Figure 6-1 shows the mapping of the CIA and optional CSA space for an SDIO card.

6.2 Register Access Time

All registers in SDIO only cards and the SDIO portion of Combo cards shall complete read and write data transfers in less than one second. This timeout value relates to the time for the requested data to be transferred to/from the host on the DAT[x] lines and not the timing between the command and the response. This wait time is signaled to the host by the card using busy for a write or delaying the start bit for a read operation. The host can use one second as the timeout value for a non-responding location. If a functions needs to support an access time greater than one second, the card maker may use some function specific method that is not defined in this specification.

6.3 Interrupts

All SDIO hosts should support hardware interrupts. If a host does not support interrupts, it may have difficulties working with SDIO cards that expect fast response to interrupt conditions. Each function within an SDIO or Combo card may implement interrupts as needed. The interrupt used on SDIO functions is a type commonly called "level sensitive". Level sensitive means that any function may signal for an interrupt at any time, but once the function has signaled an interrupt, it shall not release (stop signaling) the interrupt until the cause of the interrupt is removed or commanded to do so by the host. Since there is only 1 interrupt line, it may be shared by multiple interrupt sources. The function shall continue to signal the interrupt until the host responds and clears the interrupt. Since multiple interrupts may be active at once, it is the responsibility of the host to determine the interrupt source(s) and deal with it as needed. This is done on the SDIO function by the use of two bits, the interrupt enable and interrupt pending. Each function that may generate an interrupt has an interrupt enable bit. In addition, the SDIO card has a master interrupt enable that controls all functions. An interrupt shall only be signaled to the SD bus if both the function's enable and the card's master enable are set. The second interrupt bit is called interrupt pending. This read-only bit tells the host which function(s) may be signaling for an interrupt. There is an interrupt pending bit for each function that can generate interrupts. These bits are located in the CCCR area. For more details, refer to Table 6-1 and Table 6-2. Interrupt operation is described more fully in Section 8.

Asynchronous Interrupt is defined SDIO Version 3.00. Asynchronous Interrupt is effective in SD 4-bit mode. Asynchronous Interrupt Period, which can generate interrupt without SD clock, is defined in the Synchronous Interrupt Period after the last data block and until a next command is received. Refer to Section 8.2 for more detail.

6.4 Suspend/Resume

Within a multi-function SDIO or a Combo card, there are multiple devices (I/O and memory) that share access to the SD bus. In order to allow the sharing of access to the host among multiple devices, SDIO and combo cards can implement the *optional* concept of suspend/resume. If a card supports suspend/resume, the host may temporarily halt a data transfer operation to one function or memory (suspend) in order to free the bus for a higher priority transfer to a different function or memory. Once this higher-priority transfer is complete, the original transfer is re-started where it left off (resume). Support of suspend/resume is optional on a per-card basis. If suspend/resume is implemented, it shall be supported by the memory of a Combo card and all I/O functions *except* 0 (the CIA). Note that the host can suspend multiple transactions and resume them in any order desired. I/O function 0 does ***not*** support suspend/resume. Suspend/Resume is described in more detail in Chapter 9. Any card that supports Suspend/Resume shall also support Read Wait and Direct Commands (08h: SRW and 08h: SDC = 1) Note that Suspend/Resume is defined only for the SD 1 and 4-bit modes. It does not apply to SPI transfers.

6.5 Read Wait

Host devices shall control the SDCLK to stop the read data block output from a card executing a multiple read command whenever the host cannot accept more data. During the time that the host has stopped the SDCLK, a CMD52 cannot be issued. This limitation causes a problem in that a host device cannot perform the I/O command during a multiple read cycle.

In order to eliminate this limitation, the SDIO Specification adds the Read Wait control to enable the host to issue CMD52 during a multiple read cycle. Read Wait uses the DAT[2] line to allow the host to signal the card to temporarily halt the sending of read data by a card. This feature is optional for an SDIO or combo card. However, if an SDIO or combo supports Read Wait, all functions and any memory shall support Read Wait. Read Wait is described in more detail in Chapter 10. Any card that supports Suspend/Resume shall also support Read Wait. Note that Read Wait is defined only for the SD 1 and 4-bit modes. It does not apply to SPI transfers.

6.6 CMD52 During Data Transfer

A card may accept CMD52 during data transfer if it supports Direct Commands (Refer to 08h: SDC, Table 6-2). For both SD and SPI modes, if an error occurs during data transfer the SDIO card shall accept CMD52 to allow I/O abort and reset regardless of this bit value of the value of 08h: SDC.

6.7 SDIO Fixed Internal Map

The SDIO card has a fixed internal register space and a function unique area. The fixed area contains information about the card and certain mandatory and optional registers in fixed locations. The fixed locations allow any host to obtain information about the card and perform simple operations such as enable in a common manner. The function unique area is a per-function area, which is defined either by the Application Specifications for Standard SDIO functions or by the vendor for non-standard functions. Figure 6-1 shows the internal map of an SDIO card with multiple functions.

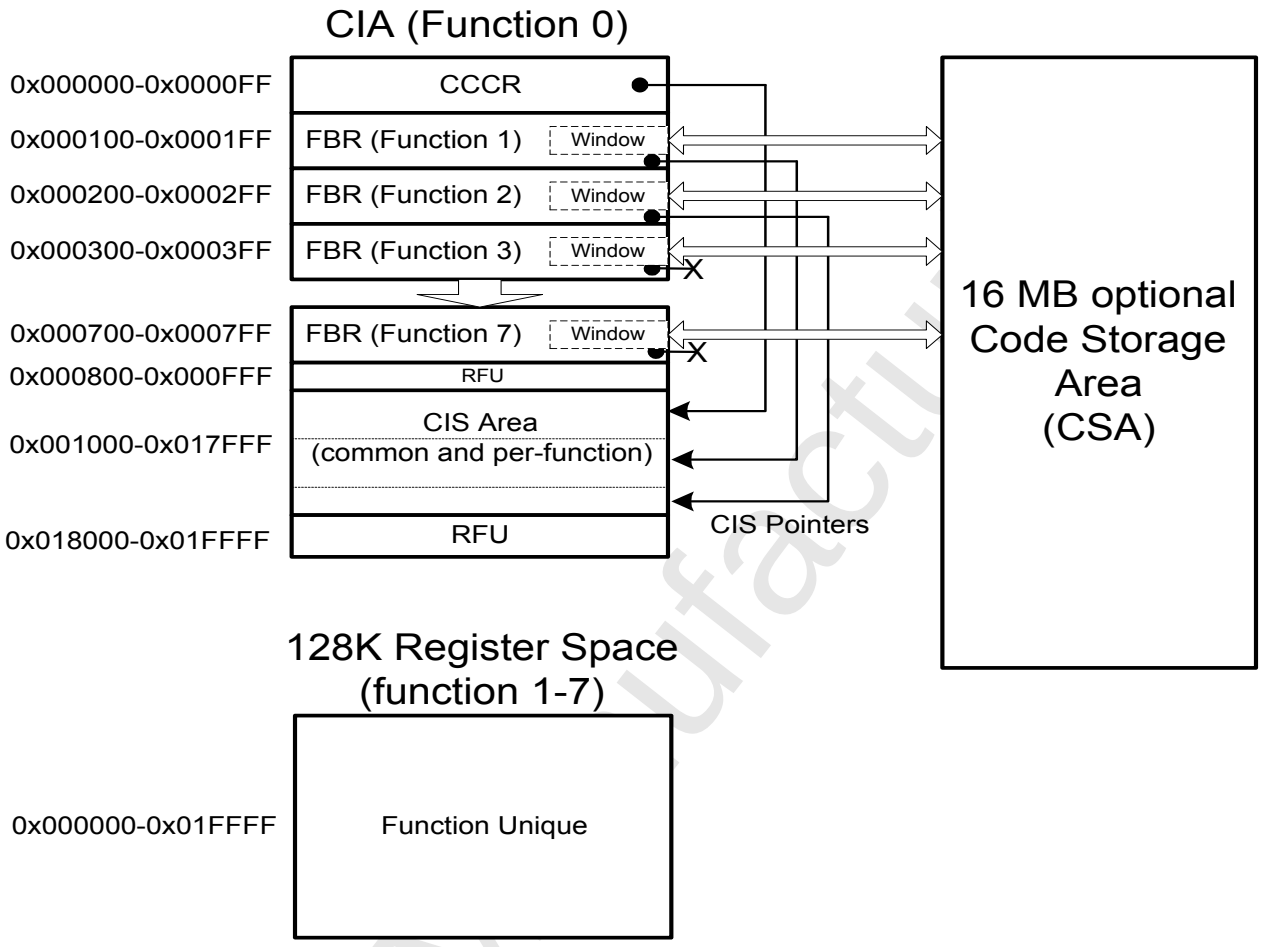


Figure 6-1 : SDIO Internal Map

6.8 Common I/O Area (CIA)

The Common I/O Area (CIA) shall be implemented on all SDIO cards. The CIA is accessed by the host via I/O reads and writes to function 0. The registers within the CIA are provided to enable/disable the operation of the I/O function(s), control the generation of interrupts and optionally load software to support the I/O functions. The registers in the CIA also provide information about the function(s) abilities and requirements. There are three distinct register structures supported within the CIA. They are:

1. Card Common Control Registers (CCCR)
2. Function Basic Registers (FBR)
3. Card Information Structure (CIS)

6.9 Card Common Control Registers (CCCR)

The Card Common Control Registers allow for quick host checking and control of an I/O card's enable and interrupts on a per card (master) and per function basis. The bits in the CCCR are mixed Read/Write and read only. If any of the possible 7 functions are not provided on an SDIO card, the bits corresponding to unused functions shall all be read-only and read as 0. All reserved for future use bits (RFU) shall be read-only and return a value of 0. All writeable bits are set to 0 after power-up or reset. Access to the CCCR is possible even after initialization when the I/O functions are disabled. Access is performed using the I/O read and write commands defined in Chapter 5. This allows the host to enable functions after

initialization. The CCCR is organized as follows:

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
00h	CCCR/SDIO Revision	SDIO bit 3	SDIO bit 2	SDIO bit 1	SDIO bit 0	CCCR bit 3	CCCR bit 2	CCCR bit 1	CCCR bit 0
01h	SD Specification Revision	RFU	RFU	RFU	RFU	SD bit 3	SD bit 2	SD bit 1	SD bit 0
02h	I/O Enable	IOE7	IOE6	IOE5	IOE4	IOE3	IOE2	IOE1	RFU
03h	I/O Ready	IOR7	IOR6	IOR5	IOR4	IOR3	IOR2	IOR1	RFU
04h	Int Enable	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IENM
05h	Int Pending	INT7	INT6	INT5	INT4	INT3	INT2	INT1	RFU
06h	I/O Abort	RFU	RFU	RFU	RFU	RES	AS2	AS1	AS0
07h	Bus Interface Control	CD Disable	SCSI	ECSI	RFU	RFU	S8B	Bus Width 1	Bus Width 0
08h	Card Capability	4BLS	LSC	E4MI	S4MI	SBS	SRW	SMB	SDC
09h-0Bh	Common CIS Pointer	Pointer to card's *common* Card Information Structure (CIS)							
0Ch	Bus Suspend	RFU	RFU	RFU	RFU	RFU	RFU	BR	BS
0Dh	Function Select	DF	RFU	RFU	RFU	FS3	FS2	FS1	FS0
0Eh	Exec Flags	EX7	EX6	EX5	EX4	EX3	EX2	EX1	EX0
0Fh	Ready Flags	RF7	RF6	RF5	RF4	RF3	RF2	RF1	RF0
11h-10h	FN0 Block Size	I/O block size for Function 0							
12h	Power Control	RFU	RFU	RFU	RFU	RFU	RFU	EMPC	SMPC
13h	Bus Speed Select	RFU	RFU	RFU	RFU	BSS2	BSS1	BSS0	SHS
14h	UHS-I Support	RFU	RFU	RFU	RFU	RFU	SDDR50	SSDR104	SSDR50
15h	Driver Strength	RFU	RFU	DTS1	DTS0	RFU	SDTD	SDTC	SDTA
16h	Interrupt Extension	RFU	RFU	RFU	RFU	RFU	RFU	EAI	SAI
EFh-17h	RFU	Reserved for Future Use (RFU)							
FFh-F0h	Reserved for Vendors	Area Reserved for Vendor Unique Registers							

Table 6-1 : Card Common Control Registers (CCCR)

Addr: Field	Type	Description
00h: CCCRx	R/O	**CCCR Format Version number.** These 4 bits contain the version of the CCCR and FBR format that this card supports. Any change to the CCCR and/or the FBR structure shall cause a new version code to be assigned. The codes for the CCCR/FBR formats are as follows: Value — CCCR/FBR Format Version 00h — CCCR/FBR defined in SDIO Version 1.00 01h — CCCR/FBR defined in SDIO Version 1.10 02h — CCCR/FBR defined in SDIO Version 2.00 03h — CCCR/FBR defined in SDIO Version 3.00 04h-0Fh — Reserved for Future Use

Addr: Field	Type	Description
00h: SDIOx	R/O	**SDIO Specification Revision Number** These 4 bits contain the version of the SDIO Specification that this card supports. The codes for the SDIO Specifications are as follows: Value — SDIO Specification 00h — SDIO Specification Version 1.00 01h — SDIO Specification Version 1.10 02h — SDIO Specification Version 1.20 (unreleased) 03h — SDIO Specification Version 2.00 04h — SDIO Specification Version 3.00 05h-0Fh — Reserved for Future Use
01h: SDx	R/O	**SD Format Version Number** These 4 bits contain the codes for the version of the Physical Layer Specification that this card supports. Value — Physical Layer Specification 00h — Physical Layer Specification Version 1.01 (March 2000) 01h — Physical Layer Specification Version 1.10 (October 2004) 02h — Physical Layer Specification Version 2.00 (May 2006) 03h — Physical Layer Specification Version 3.0x 04h-0Fh — Reserved for Future Use
02h: IOEx	R/W	**Enable Function** If this bit is reset to 0, the function is disabled. If this bit is set to 1, the function is enabled to start its initialization. The completion of initialization is indicated in IORx. On power up or after a reset, the card shall reset this bit to 0. The host can also use IOEx as a per function reset for error recovery. The host sequence for a per function reset is to reset IOEx to 0, wait until IORx becomes 0 and then set IOEx to 1 again. If the error is not recovered by this sequence, SDIO reset should be used noting that the operation of all functions will be aborted. Refer to Chapter 11 for relation to Master Power Control and Power Select.
03h: IORx	R/O	**I/O Function Ready** If this bit is reset to 0, the function is not ready to operate. This may be caused by the function being disabled or not ready due to internal causes such as a built-in self-test in progress. If this bit is set to 1, the function is ready to operate. The functions shall set this bit to 1 within the timeout value defined in the TPLFE_ENABLE_TIMEOUT_VAL tuple. On power up or after a reset, this bit shall be set to 0. For any function that is not implemented on an SDIO card, this bit shall always be 0.
04h: IENx	R/W	**Interrupt Enable for Function x** If this bit is cleared to 0, any interrupt from this function shall not be sent to the host. If this bit is set to 1, then this function's interrupt shall be sent to the host if the master Interrupt Enable (bit 0) is also set to 1.
04h: IENM	R/W	**Interrupt Enable Master** If this bit is cleared to 0, no interrupts from this card shall be sent to the host. If this bit is set to 1, then any function's interrupt shall be sent to the host.
05h: INTx	R/O	**Interrupt Pending for Function x** If this bit is cleared to 0, this indicates that no interrupts are pending from this function. If this bit is set to 1, then this function has interrupt pending. Note that if the IENx *or* IENM bits are not set, the host cannot receive this pending interrupt.

Addr: Field	Type	Description
06h: ASx	W/O	**Abort Select** In order to abort an I/O read or write and free the SD bus, the function that is currently transferring data must be addressed. These 3 bits define which function's transfer to stop. For example, the abort the transfer to function number 3, the value of 03h would be written to these bits using CMD52 only. If the abort is addressed to a suspended function, it does not affect current data transaction. Note that this is an abort, not a reset. The addressed function shall return to the CMD state and data transfer pending to that function shall be halted. This abort procedure does not work for SPI write operations. To abort an SPI write data transfer use the STOP_TRAN token as defined in Section 7.3.3 of the Physical Layer Specification. This form of abort applies only to the functions of an SDIO card. For the memory of a combo card, the abort methods defined in the Physical Layer Specification shall be used to abort transfers to/from memory
06h: RES	W/O	**I/O CARD RESET** Setting the RES to 1 shall cause all I/O functions in an SDIO or Combo card to perform a soft reset. Setting the RES to 1 does not affect the current card protocol selection (SD vs. SPI mode) and CD Disable. Setting of the RES bit shall only be performed using CMD52. When RES=1, the values of AS2-0 are don't-cares. The RES bit is auto cleared, so there is no need to rewrite a value of 0. This bit is write-only, any read returns an undetermined value. Memory in a combo card is not affected.
07h: Bus Width 1:0	R/W	**Bus Width** This field selects the bus width of the DAT line for a data transfer. All Full-Speed SDIO cards shall support both 1 and 4-bit bus widths. A Low-Speed SDIO card's support of 4-bit bus is optional. 1-bit bus width is the default. On reset or power-on these bits are cleared to 00. 8-bit bus mode can be supported only for an embedded device. 8-bit mode can be set if S8B is set to 0. 00b: 1-bit 01b: Reserved 10b: 4-bit bus 11b: 8-bit bus (only for embedded SDIO)
07h: S8B	R/O	**Support 8-bit Bus Mode** This support bit is valid for embedded SDIO. SDIO card shall set this bit to 0. 0: 8-bit bus mode is not supported. 1: 8-bit bus mode is supported. (The details of 8-bit mode are not defined in this document but it will be defined in a future specification.)
07h: ECSI	R/W	**Enable Continuous SPI Interrupt** If the SCSI bit is set, then this R/W bit is used to allow the SDIO card to assert the interrupt line in the SPI mode at any time, irrespective of the state of the CS line. This bit is cleared to zero on reset or power-up. If the SCSI bit is clear, this bit shall be read only and set to zero. This bit controls the assertion of interrupts in the SPI mode for all functions in the SDIO card.
07h: SCSI	R/O	**Support Continuous SPI interrupt** This read-only bit is set to indicate that this SDIO card supports the assertion of interrupts in the SPI mode at any time, irrespective of the status of the card select (CS) line. If this bit is zero, then this SDIO card can only assert the interrupt line in the SPI mode when the CS line is asserted. This bit signals the capability of all functions in the SDIO card.

Addr: Field	Type	Description
07h: CD Disable	R/W	**Card Detect Disable** Connect[0]/Disconnect[1] the 10K-90K ohm pull-up resistor on CD/DAT[3] (pin 1) of the card. The pull-up may be used for card detection. This bit is cleared to 0 on power-on (connected). Its state is not affected by a reset command. This bit shall be set to 1 before issuing CMD53.
08h: SDC	R/O	**Support Direct Command (CMD52)** Support bit of Direct Commands. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to execute CMD52 while data transfer is in progress. If this bit is set, all I/O functions shall accept and execute the CMD52 while data transfer is underway on the DAT[x] lines. Also, any memory in a combo card shall allow the CMD52 to execute while it is transferring data. Since the CMD52 does not use the DAT[x] lines, it is possible to execute while data transfer to a different address on the card is underway. CMD52 is described in 5.1. In SD or SPI mode, if an error occurs during data transfer the SDIO card shall accept a CMD52 to allow I/O abort and reset regardless of this bit value. If the card supports suspend/resume then it shall also support this bit.
08h: SMB	R/O	**Support Multiple Block Transfer** Support bit of Multi-Block transfer. This flag bit reports the SDIO card's ability to execute the IO_RW_EXTENDED command (CMD53) in the block mode. If this bit is set, all I/O functions (0-7) shall accept and execute CMD53 with the optional block mode bit set. The IO_RW_EXTENDED command is described in 5.3
08h: SRW	R/O	**Support Read Wait** Support bit of Read Wait control. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to support the Read Wait Control (RWC) operation. If set, all functions on the card are able to accept the wait signal on DAT[2]. RWC operation is described in Section 6.5. Any card that supports Suspend/Resume shall also support Read Wait
08h:SBS	R/O	**Support Bus Control** Support bit of Suspend/Resume. This bit applies only to the SD modes, it does not apply to SPI mode. This flag bit reports the SDIO card's ability to Suspend and Resume operations at the request of the host. If this bit is set, all functions except 0 shall accept a request to suspend operations and resume under host control. Suspend/Resume operation is described in Section 6.4. If this bit is 0, registers (0Ch-0Fh) shall not be supported.
08h:S4MI	R/O	**Support Block Gap Interrupt** Support bit of interrupt between blocks of data in 4-bit SD mode. This flag bit reports the SDIO card's ability to generate interrupts during a 4-bit multi-block data transfer. If this bit is 0, then the SDIO card is not able to signal an interrupt during a multi-block data transfer in 4-bit mode. In this case, the interrupt is not signaled until after the data transfer is complete. If this bit is 1, then the SDIO card is able to signal an interrupt between blocks while data transfer is in progress. This operation is described in 8.1.4 Note, even if a card does not support the interrupt during 4-bit block transfer (S4MI=0), the card may signal interrupts during all other Interrupt Periods while the interrupt is enabled (IENx=1).

Addr: Field	Type	Description
08h:E4MI	R/W	**Enable Block Gap Interrupt** Enable bit of interrupt between blocks of data in 4-bit SD mode. Enable the multi-block IRQ during 4-bit transfer for the SDIO card. When this bit is 0, the card shall not signal interrupts during a 4-bit multi-block data transfer. If this bit is 1, the card shall generate interrupts during 4 bit multi-block data transfers as described in 8.1.4 If this SDIO card does not support 4 bit multi-block IRQs (S4MI=0), then this bit shall be R/O and always read as 0. This bit shall be cleared to 0 by any reset. When operating more than 25MB/sec data rate in SDR50, SDR104 or DDR50, Block Gap Interrupt shall not be used (this bit is set to 0). It is difficult for a host to use the narrow interrupt period considering propagation delay from card to host.
08h:LSC	R/O	**Low-Speed Card** If this bit is set, it indicates that the SDIO card is a Low-Speed card (Refer to Section 2.1). If this bit is clear, the SDIO card is a Full-Speed card.
08h:4BLS	R/O	**4-bit Mode Support for Low-Speed Card** If the SDIO card is a Low-Speed card (LSC=1) and it supports 4-bit data transfer, then this bit shall be set. If the card is not Low-Speed or if the card does not support 4-bit transfer, then this bit shall be zero.
09h-0Bh: Pointer to card's common CIS	R/O	**Common CIS Pointer** This 3-byte pointer points to the start of the card's common CIS. The common CIS contains information relation to the entire card. The card common CIS shall be located within the CIS space of function 0 (001000h- 017FFFh) as described in Section 6.11. A card common CIS is mandatory for all SDIO cards. This pointer is stored in little-endian format (LSB first).
0Ch: BS	R/O	**Bus Status** If this bit is set to 1, then the currently addressed function (selected by FSx or by the function number in an I/O command) is currently executing a command which transfers data on the DAT[x] line(s). If this bit is 0, then the addressed function is not using the data bus. This bit is used by the host to determine which function of a multi-function or combo card is currently performing data transfer. Note that this bit is a part of the optional Suspend/Resume protocol. If the card does not support Suspend/Resume, this bit shall be read as 0. Any access to the CIA may not be suspended, so in this case, BS shall always be set to 1, irrespective of the host setting BR to 1.

Addr: Field	Type	Description
0Ch: BR	R/W	**Bus Release Request** This bit is used to request that the addressed function (selected by FSx or by the function number in CMD53 or Memory commands using DAT line) release the Data lines and suspend operation. If the host sets this bit to 1, the addressed function shall temporarily halt data transfer on the DAT[x] lines and suspend the command that is in process. The BR bit shall remain set to 1 until the release is complete. If the card can never accept the suspend request while executing transactions, the card shall return response with BR cleared to 0 and BS set to 1. This indicates that the suspend request is cancelled by the card and thus the host should not issue a cancel suspend command. The followings are the cases where the card can cancel a suspend request: Transaction addressed to function 0. The card knows the transfer will terminate soon. The card knows the transfer is timing critical (i.e. If suspended, the transfer cannot proceed). A Multi function card that indicates SBS=1, but contains a function that does not support suspend/resume. Once the function is in suspend, it shall signal the host by clearing the BS and BR bits. The host can monitor the status of the suspend request by reading the BR bit. If it is set, the suspend request is still in progress. A pending suspend request can be cancelled by the host by writing 0 to the BR bit. The Standard Host Specification defines following suspend sequence: If the suspend request is not accepted, the host retries with a cancel suspend request command. Even if the card received a cancel suspend command, it should accept suspend if possible. If the card does not accept suspend, the host considers the function to have never suspend. However, the host should monitor the BR, BS and EXx bits to confirm that the suspend request was cancelled. If SBS=0, this bit shall be R/O and read as 0.
0Dh: FSx	R/W	**Select Function** These four bits are used to select a function number (0-7) or the memory of a combo card (8) for Suspend/Resume. There are two methods to write the value of FSx. First, an I/O write to the register in the CCCR and second, a new I/O command that causes the FSx to be set to the function number in that command. The value of FSx shall remain until overwritten. If a function or memory is currently suspended, the writing of its number to FSx shall re-start (resume) the data transfer operation When reading FSx, the value returned shall be the number of the currently addressed function. Note that when reading FSx, if the Bus Status is 0 (BS=0), the FSx value is undefined. The FSx bits are coded as follows: **FSx** — **Current Transaction** 0000 — Transaction of function 0 (CIA) 0001-0111 — Transaction to functions 1-7 1000 — Transaction of memory in combo card 1001-1111 — Not defined, reserved for future use If SBS=0 these bits shall be R/O.

Addr: Field	Type	Description
0Dh: DF	R/O	**Data Flag** A data transaction is resumed by writing its number to FSx. Once the transaction is resumed, the DF indicates if more data will be transferred. If DF is cleared to 0, then no additional data will be transferred after the function or memory is resumed. If DF is set to 1, then there is more data to transfer that will begin after the function or memory in resumed. The DF flag can be used to control the interrupt cycle in 4-bit mode. If DF=1, there is more data to transfer after restoring the function. In this case, the interrupt cycle should be disabled. If DF=0, the function or memory was suspended at end of data transfer (during busy). In this case, no data transfer shall begin after resume so the host can detect a start interrupt cycle after restore. When resuming, if the suspended function cannot continue data transfer the card shall return DF=0 to abort the transfer.
0Eh: EXx	R/O	**Execution Flag** These 8 bits are used by the host to determine the current execution status of all functions (1-7) and memory (0). The bit is set to 1 for each function or memory that is currently executing a command. The EXx bits tell the host that a function or memory is currently executing a command so no additional command should be issued to that function/memory. These bits are only defined if SBS=1. This bit is set if the function is active (either currently executing or suspended Refer to Figure D- 2). If SBS=0 these bits shall be read as zero.
0Fh: RFx	R/O	**Ready Flag** These 8 bits tell the host the read or write busy status for functions (1-7) and memory (0). If a function or memory is executing a write transaction, an RFx bit cleared to 0 indicates the function/memory is busy and not ready to accept more data. If the RFx bit is set to 1, then the function/memory can accept write data. If a function/memory is executing a read command, if the RFx bit is cleared to 0, it indicates that read data is NOT available. If the bit is set to 1, it indicates that read data is ready to be transferred. These bits are only defined if SBS=1. Setting a bit to 1 indicates the function is ready to accept the resume command. There are two conditions where the function will set the bit to 1. One is when the function (executing or suspended) is ready to continue data transfer. The second is when the suspended function cannot continue data transfer. (Refer to Figure 9-5 and Figure 9-6). If SBS=0 these bits shall be read as zero.
10h-11h: FN0 Block Size	R/W	**Block Size for Function 0** This 16-bit register sets the block size for I/O block operations for Function 0 only. If this card does not support I/O block operations (SMB=0), then this register becomes read-only and shall always read 0000h. The maximum block size is 2048 (0800h) and the minimum is 1. At power-up or reset, this register shall be initially loaded with a value of 0000h. The host is responsible for setting the appropriate value for the block size supported by function 0. This pointer is stored in little-endian format (LSB first).
12h: SMPC	R/O	**Support Master Power Control** These bits tell the host if the card supports Master Power Control. SMPC=0: The total card power is up to 720mW (3.6Vx200mA), even if all functions are active (IOEx=1). EMPC, SPS and EPS shall be zero. SMPC=1: The total card power may exceed 720mW (3.6Vx200mA). Controls of EMPC, SPS and EPS are available.

Addr: Field	Type	Description
12h: EMPC	R/W	**Enable Master Power Control** EMPC=0 (default): The total card power shall be up to 720mW (3.6Vx 200mA). The card automatically switches the mode of function(s) to lower current or does not allow some functions to become enabled, regardless of the value of EPS, so that total card power is 720mW or less. (The card manufacturer determines which functions operate and their modes to guarantee this limit.) EMPC=1: The total card power may exceed 720mW and SPS and EPS are available. The host uses SPS, EPS in FBR and IOEx to enable higher current function modes based on the host's ability to supply the necessary current. If **PS[3:0]=0** (Ver1.10 compatible), two power modes can be selected by using SPS, EPS in FBR for an SDIO Card supporting 2.7V-3.6V Power supply range. **PS[3:0] >0** is defined by Ver3.00 Power Control Extension (Applicable to multiple voltages device).
13h: SHS	R/O	**Support High-Speed** This flag bit reports the card's ability to operate in High-Speed mode SHS=0: The card does not support High-Speed mode SHS=1: The card supports High-Speed mode. The host enables High-Speed mode via the EHS bit. Refer to Chapter 12 for details on switching between default and High-Speed mode.
13h: BSSx	R/W	**Bus Speed Select** This field is used to select bus speed mode. This field shall be set by CMD52 and the new bus speed mode is changed within 8 clock cycles after the response of CMD52. Refer to Section 12 for details on switching between default and High-Speed mode. If the card is initialized in 3.3V signaling, then BSS0 is effective. When SHS is set to 0, writing to this bit is ignored and always indicates 0. The card operates in High-Speed timing mode with a clock rate up to 50MHz. BSS[2:0] Bus Speed 3.3V Max. Clock Frequency xx0b: Default Speed 25MHz xx1b: High Speed 50MHz If the card is initialized in 1.8V signaling, then BSS[2:0] is effective. A card that supports UHS-I shall support High Speed mode and SHS shall be set to 1. One of the UHS-I modes is selected as follows. BSS[2:0] Bus Speed 1.8V Max Clock Frequency 000b: SDR12 25MHz 001b: SDR25 50MHz 010b: SDR50 100MHz 011b: SDR104 208MHz 100b: DDR50 50MHz 101b-111b: Reserved
14h: SSDR50	R/O	**Support SDR50** This bit indicates support of SDR50. If this bit is set to 1, SDR12 and SDR25 shall be supported. High Speed mode in 3.3V signaling also shall be supported. Support bit of SDR50 0b: SDR50 is not supported 1b: SDR50 is supported

Addr: Field	Type	Description
14h: SSDR104	R/O	**Support SDR104** This bit indicates support of SDR104. If this bit is set to 1, SDR12, SDR25 and SDR50 shall be supported. High Speed mode in 3.3V signaling also shall be supported. Support bit of SDR104 0b: SDR104 is not supported 1b: SDR104 is supported
14h: SDDR50	R/O	**Support DDR50** This bit indicates support of DDR50. If this bit is set to 1, SDR12, SDR25 and SDR50 shall be supported. High Speed mode in 3.3V signaling also shall be supported. Support bit of DDR50 0b: DDR50 is not supported 1b: DDR50 is supported
15h: SDTA	R/O	**Support Driver Type A** This bit indicates support of Driver Type A. Support bit of SDR50 0b: Driver Type A is not supported 1b: Driver Type A is supported
15h: SDTC	R/O	**Support Driver Type C** This bit indicates support of Driver Type C. Support bit of SDR50 0b: Driver Type C is not supported 1b: Driver Type C is supported
15h: SDTD	R/O	**Support Driver Type D** This bit indicates support of Driver Type D. Support bit of SDR50 0b: Driver Type D is not supported 1b: Driver Type D is supported
15h: DTSx	R/W	**Driver Type Select** This 2-bit field is used to select driver type. Type B is defined as the default driver strength. If unsupported driver type is selected, Type B is selected. Refer to the Physical Layer Specification Version 3.0x for more detail. DTS[1:0] Driver Type Selected 00b: Type B 01b: Type A (SDTA=1) or Type B (SDTA=0) 10b: Type C (SDTC=1) or Type B (SDTC=0) 11b: Type D (SDTD=1) or Type B (SDTD=0)
16h: SAI	R/O	**Support Asynchronous Interrupt** Support bit of Asynchronous Interrupt. If the card supports asynchronous interrupt in SD 4-bit mode (interrupt can be asserted without SD clock during specified period), this bit is set to 1.

Addr: Field	Type	Description
16h: EAI	R/W	**Enable Asynchronous Interrupt** Enable bit of Asynchronous Interrupt. When SAI is set to 0, writing to this bit is ignored and always indicates 0. This bit is effective in SD 4-bit mode. EAI Interrupt Mode 0b: Asynchronous Interrupt is disabled 1b: Asynchronous Interrupt is enabled
RFU	R/O	Any bit defined as Reserved for Future Use (RFU) shall be read-only and shall be read as 0.
Reserved for Vendors	R/W	These 16 registers are reserved for the manufacturer of the I/O card to be used for any operations that are defined by and specific to any vendor unique operation. Information about the use of these optional registers needs to be obtained from the SDIO card maker. Reading and/or writing these registers without understanding the vendor's definitions may cause unexpected behavior or even damage to the card.

Table 6-2 : CCCR bit Definitions

6.10 Function Basic Registers (FBR)

In addition to the CCCR, each supported I/O function has a 256-byte area used to allow the host to quickly determine the abilities and requirements of each function, enable power selection for each function and to enable software loading. The address of this area is from 00n00h to 00nFFh where n is the function number (1 to 7). This per-function area is structured as follows:

Address	7	6	5	4	3	2	1	0
100h	Function 1 CSA enable	Function 1 supports CSA	RFU	RFU	Function 1 Standard SDIO Function Interface Code			
101h	Function 1 Extended standard SDIO Function interface code							
102h	PS3	PS2	PS1	PS0	RFU	RFU	EPS	SPS
103h-108h	Reserved for Future Use (RFU)							
109h-10Bh	Pointer to Function 1 Card Information Structure (CIS)							
10Ch-10Eh	Pointer to Function 1 Code Storage Area (CSA)							
10Fh	Data access window to Function 1 Code Storage Area (CSA)							
110h-111h	I/O block size for Function 1							
112h-1FFh	Reserved for Future Use							
200h-7FFh	Function 2 to 7 Function Basic Information Registers (FBR)							
800h-FFFh	Reserved for Future Use							

Table 6-3 : Function Basic Information Registers (FBR)

The Individual bits and fields in the FBA are defined below in Table 6-4.

Field	Type	Description
SDIO Standard Function Interface Code	R/O	The SDIO Standard Function code identifies those I/O functions, which implement the recommended standard interface as defined in a separate Application Specification. A complete and current list of assigned standard codes shall be maintained and published in any addendums to this specification. The codes assigned to those standard interfaces at the time this specification was published are: 0h: No SDIO standard interface supported by this function 1h: This function supports the SDIO Standard UART 2h: This function supports the SDIO Bluetooth Type-A standard interface 3h: This function supports the SDIO Bluetooth Type-B standard interface 4h: This function supports the SDIO GPS standard interface 5h: This function supports the SDIO Camera standard interface 6h: This function supports the SDIO PHS standard interface 7h: This function supports the SDIO WLAN interface 8h: This function supports the Embedded SDIO-ATA standard interface (Embedded SDIO-ATA shall be implemented only on devices following the "Embedded SDIO Specification"). 9h: This function supports the SDIO Bluetooth Type-A AMP standard interface (AMP: Alternate MAC PHY) 10h-Eh: Not assigned, reserved for future use Fh: This function supports an SDIO standard interface number greater than Eh. In this case, the value in byte 101h identifies the standard SDIO interfaces type.

Field	Type	Description
Function Supports CSA	R/O	If this function supports and contains a Code Storage Area (CSA), this bit shall be set to 1. If this function does not support a CSA, this bit shall be cleared to 0. CSA enable is controlled by bit 7 of register n00h.
Function CSA Enable	R/W	This bit controls access to the Code Storage Area for this function. If this bit is cleared to 0, then any read or write access to the CSA shall be blocked. If this bit is set to 1, then access to the CSA is allowed. This bit is cleared to 0 upon reset. If this function does not support CSA (0n00h bit 6=0), then this bit shall be R/O and always read as 0.
Extended SDIO Standard Function interface code	R/O	This is the extension of the SDIO Standard Function interface code. If the SDIO Standard Function interface code is greater than Eh, then this byte shall contain the code and the standard code (100h bits 3-0) shall contain a value of Fh. If the standard code is less than Fh, then this byte shall be 00h.
SPS	R/O	**Support Power Selection** This bit indicates if the function has Power Selection. SPS=0: This function has no Power Selection. EPS shall be zero. SPS=1: This function has 2 power modes which are selected by EPS. This bit is effective when EMPC=1 in CCCR and PS[3:0]=0.
EPS	R/W	**Enable Power Selection** EPS=0(default): The function operates in Higher Current Mode The maximum current for the function shall be given in TPLFE_HP_MAX_PWR_3.3V EPS=1: The function works in Lower Current Mode The maximum current for the function shall be given in TPLFE_LP_MAX_PWR_3.3V This bit shall be reset when IOEx=0. This bit is effective when EMPC=1 in CCCR and PS[3:0]=0.
PSx	R/W	**Power State PS[3:0]** If PS[3:0] is set to 0, TPL_CODE CISTPL_FUNCE (22h) extension 01h is used and the card power is controlled by EMPC and EPS (SDIO Version 2.00 Compatible). Power State control is defined by SDIO Version 3.00 and is effective when EMPC is set to 1 and PS[3:0] is set to larger than 0. In this case, a list of card supported power states is described in the TPL_CODE CISTPL_FUNCE (22h) extension 02h (Power State Tuple). The host driver finds an affordable power in the Tuple and the number "n"(>0) is set to this field. The total current of a card is the sum of selected current of each function. Refer to Chapter 11 about power control and Section 16.7.5 about Power State Tuple.
Address pointer to Function CIS	R/O	These three bytes make up a 24-bit pointer (only the lower 17 bits are used) to the start of the Card Information Structure (CIS) that is associated with each function. The CIS is defined in Section 6.11. A CIS is mandatory for each function on an SDIO card. This pointer is stored in little-endian format (LSB first). This register points to the End of Chain tuple if the function is not supported on the card.
Address pointer to Function CSA	R/W	These three bytes make up a 24-bit pointer to the desired byte in the CSA to read or write. After any read or write to the CSA access window register, this pointer shall be automatically incremented by 1. If this function does not support CSA (n00h bit 6=0), then these 24 bits shall be R/O and always read as 000000h. This pointer is stored in little-endian format (LSB first).
Data access window to CSA	R/W	Any read or write to this address when the CSA is enabled (n00h bit 7=1), shall pass data to/from the byte addressed by the CSA address pointer. If this function does not support CSA (n00h bit 6=0), then these 8 bits shall be R/O and always read as 00h.

Field	Type	Description
Function 1-7 I/O Block Size	R/W	This 16-bit register sets the block size for I/O block operations for each function (1-7). If this card does not support I/O block operations (SMB=0), then this register becomes read-only and shall always read 0000h. The maximum block size is 2048 (0800h) and the minimum is 1 (0001h). At power-up or reset, this register shall be initially loaded with a value of 0000h. The host is responsible for setting the appropriate value for the block size supported by each function. This pointer is stored in little-endian format (LSB first).

Table 6-4 : FBR bit and field definitions

6.11 Card Information Structure (CIS)

The Card Information Structure provides more complete information about the card and the individual functions. The CIS is the common area to read information about all I/O functions that exist in a card. The design is based on the PC Card16 design standardized by PCMCIA. All cards that support I/O shall have a common CIS and a CIS for each function. The CIS is accessed by reads to a fixed area as shown in Table 6-5 This one area serves the card as a Common CIS and also as the storage area for each function. The common area and each function have a pointer to the start of its CIS within this memory space.

Address	7	6	5	4	3	2	1	0
001000h-017FFFh	Card Common Card Information Structure (CIS) area for card common and all functions							
018000h-01FFFFh	Reserved for Future Use							

Table 6-5 : Card Information Structure (CIS) and reserved area of CIA

The valid tuples (storage structures) from the PCMCIA specification and new tuples created for SDIO are defined in Section 16.7.

6.12 Multiple Function SDIO Cards

Multiple Function SDIO Cards shall have a separate set of Configuration registers for each function on the card. Multiple Function SDIO Cards shall use a combination of a CIS common to all functions on the card and a separate function-specific CIS specific to each function on the card. The common CIS describes features that are common to all functions on the card. Each function-specific CIS describes features specific to a particular function on the SDIO Card. Functions are numbered sequentially beginning with 1. The CMD5 response indicates the total number of functions, which includes 'dummy' functions. The host shall iterate through the CIS entries based on the CMD5 response.

The ERROR status flag of an R5 response is type "E R X", (Refer to Section 5.2.1) and can indicate an error in the previous command. Since the host software needs a method to determine which function detected the error, a Multiple Function SDIO cards shall only return the R5 ERROR status flag in the subsequent command issued to the same function.

6.13 Setting Block Size with CMD53

The host sets the block size for a function's multiple block transfers by writing to the 16-bit Function I/O Block Size register in the FBR (Refer to Table 6-4). The host shall not write this register using CMD53 with Block Mode set to 1. If the card detects an invalid block size before executing CMD53 with Block Mode set to 1, it shall indicate an OUT_OF_RANGE error in the current response and shall not perform data transfer. This will also stop the interrupt period (Refer to Section 8.1.3)

6.14 Bus State Diagram

Figure 6-2 shows the Bus State Diagram for an SDIO card. It shows the bus states and their relations to SDIO commands and Suspend/Resume.

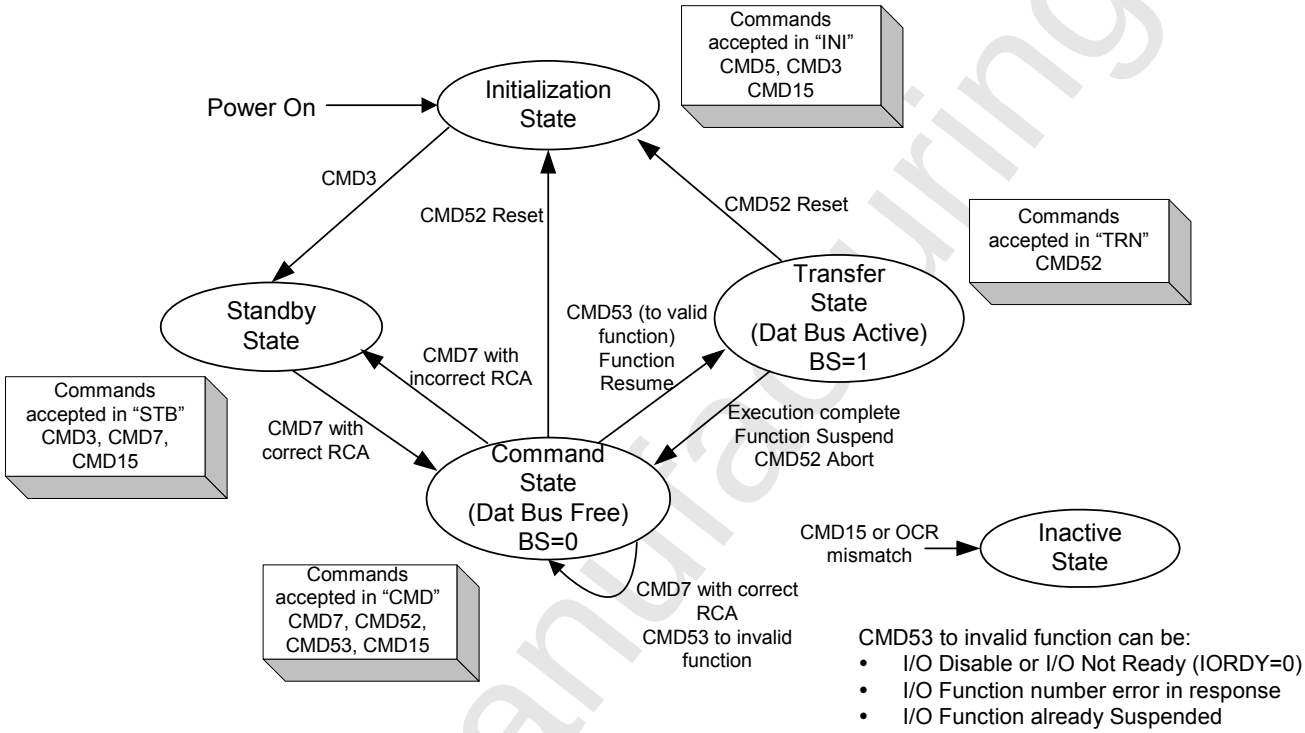


Figure 6-2 : State Diagram for Bus State Machine

7. Embedded I/O Code Storage Area (CSA)

In order to support the concept of "Plug-and-Play" for SDIO cards, each function contained in a card may need to contain a block of memory for the storage of drivers and/or applications. In addition, since the same SDIO card may be used on multiple different host platforms, several different versions of the code may be needed for each function. One option is to store these programs in a standard SD Memory section of a combo card. Alternately, a standard access means to load the code is contained in the optional Code Storage Area (CSA). The CSA is a separate 16MB memory area that is accessed using the CSA address pointer and the CSA window register contained in the FBR registers. Note that each function may have its own CSA to support it. The CSA data can be read only or R/W. The actual storage method for the CSA is not a part of this specification and left to implementers.

7.1 CSA Access

In order for the host to access a function's CSA, it first shall determine if that function supports a CSA. The host reads the FBR register at address 00n00h where n is the function number (1 to 7). If bit 6=1, then the function supports a CSA and the host enables access by writing bit 7=1. The next step is for the host to load the 24 bit address to start reading or writing. This is accomplished by writing the 24 bits (A23-0) to registers 00n0Ch to 00n0Eh where n is the function number (1 to 7). Once the start address is written, data can be read or written by accessing register 00n0Fh, the CSA data window register. If more than 1 byte needs to be read or written, an extended I/O command (byte or block) can be performed with an OP code of 0 (fixed address). The address pointer shall be automatically incremented with each access to the window register, so the access will be to sequential addresses within the CSA. Once the operation is complete, the address of the NEXT operation shall be held in the 24 bit address register for the host to read.

7.2 CSA Data Format

The data stored in the CSA shall be structured using the FAT12/FAT16 format. This format is defined in the ISO specification*: ISO/IEC9293:1994 Information technology - Volume and file structure of disk cartridges for information interchange*. This specification is also the basis for the SD memory cards. The information on the SD memory implementation can be found in the SDA publication*: Part 2 FILE SYSTEM SPECIFICATION Version 2.00 May 9, 2006*. The actual layout of files within the CSA is undefined by this specification.

The use of the CSA for program or data storage for different host types requires that the SDIO card manufacturer load the programs and data in a file format that may be recognized by the host. An example of this would be the use of a specific file name saved within a specific subdirectory that is recognized and executed by a particular host operating system. Such formats are specific and sometimes proprietary to different host implementations and operating systems.

7.3 CSA Licensing Notice

An SDIO card with CSA requires the Card License Agreement (CLA) with SD-3C, LLC. When the CSA is less than 2 Megabytes (2,097,152 bytes), it is royalty free.

8. SDIO Interrupts

In order to allow the SDIO card to interrupt the host, an interrupt function is added to a pin on the SD interface. Pin number 8, which is used as DAT[1] when operating in the 4-bit SD mode, is used to signal the card's interrupt to the host. The use of interrupt is optional for each card or function within a card. The SDIO interrupt is "level sensitive", that is, the interrupt line shall be held active (low) until it is either recognized and acted upon by the host or de-asserted due to the end of the Interrupt Period (Refer to 8.1.2). Once the host has serviced the interrupt, it is cleared via some function unique I/O operation. All hosts shall provide pull-up resistors on all data lines DAT[3:0] as described in Chapter 6 of the Physical Layer Specification.

8.1 Interrupt Timing

The operation of the interrupt pin is different between the SPI mode and the SD mode. The operation of the interrupt pin is defined as follows:

8.1.1 SPI and SD 1-bit Mode Interrupts

In the SPI and 1-bit SD mode, Pin 8 is dedicated to the interrupt function. Thus, in the SPI and SD 1-bit modes there are no timing constraints on interrupts. A card in the SPI or 1-bit SD mode signals an interrupt to the host at any time by asserting pin 8 low. The host detects this pending interrupt using a level sensitive input. The host is responsible for clearing the interrupt. If the SDIO card is operating in the SPI mode, the interrupt from the card may not be asserted if the card is not selected.(CS=0). The exception to this requirement occurs only if the card is both capable of interrupting when not selected (the SCSI bit in the CCCR = 1), and has that feature turned on (the ECSI bit = 1). In this case, the card may assert the interrupt irrespective of the state of the CS line. For more information, refer to Table 6-1.

8.1.2 SD 4-bit Mode

Since Pin 8 is shared between the IRQ and DAT[1] when used in 4-bit SD mode, an interrupt shall only be sent by the card and recognized by the host during a specific time. The time that a low on Pin 8 shall be recognized as an interrupt is defined as the ***Interrupt Period***.

An SDIO host shall only sample the level on Pin 8 (DAT[1]/IRQ) into the interrupt detector during the Interrupt Period. At all other times, the host interrupt controller shall ignore the level on Pin 8. Note that the Interrupt Period is applicable for both memory and I/O operations

The Interrupt Period timing is described as synchronous to the SD clock at the connector timing. However, when operating more than 25MB/sec data rate in SDR50, SDR104 or DDR50, the host should treat the interrupt signal as an asynchronous input considering propagation delay from card to host.

The definition of the Interrupt Period is different for operations with single block and multiple block data transfer.

8.1.3 Interrupt Period Definition

The Interrupt Period initially begins after the following: a function is initialized (02h: IOEx), the card is placed into 4-bit SD mode and interrupts are enabled (04h: IENM and 04h: IENx). A card, depending on design, may require a longer time until it is able to issue interrupts.

The Interrupt Period then ends

- At the next clock from the end bit of a command that transfers data block(s) using DAT[x] lines

And resumes

- Two clocks after the completion of the last data block transfer in a transaction.

If the command causes a CRC or Illegal Command error the card chooses whether or not it will continue

the Interrupt Period. If the card chooses to end the Interrupt Period, it shall resume by an abort command to any IO function or by the card itself after 64 clocks. In case of a Function Number error in CMD53, the Interrupt Period shall end and resume by an abort command to any IO function.

Figure 8-1 and Figure 8-2 show the timing of the Interrupt Period for two different types of single data transaction read cycles.

Figure 8-3 shows the timing of a single data block transaction write cycle.

This Interrupt Period is intended to prevent the interaction between the DAT[1] data and the interrupt signals on a common pin. Note that the Interrupt Period includes the times when there is no command or data activity between the host and the card. In the case where the interrupt mechanism is used to wake the host while the card is in a low power state (i.e. no clocks), there are two methods. One is both the card and the host shall be placed into the 1-bit SD mode prior to stopping the clock. The other is using asynchronous interrupt explained in Section 8.2.

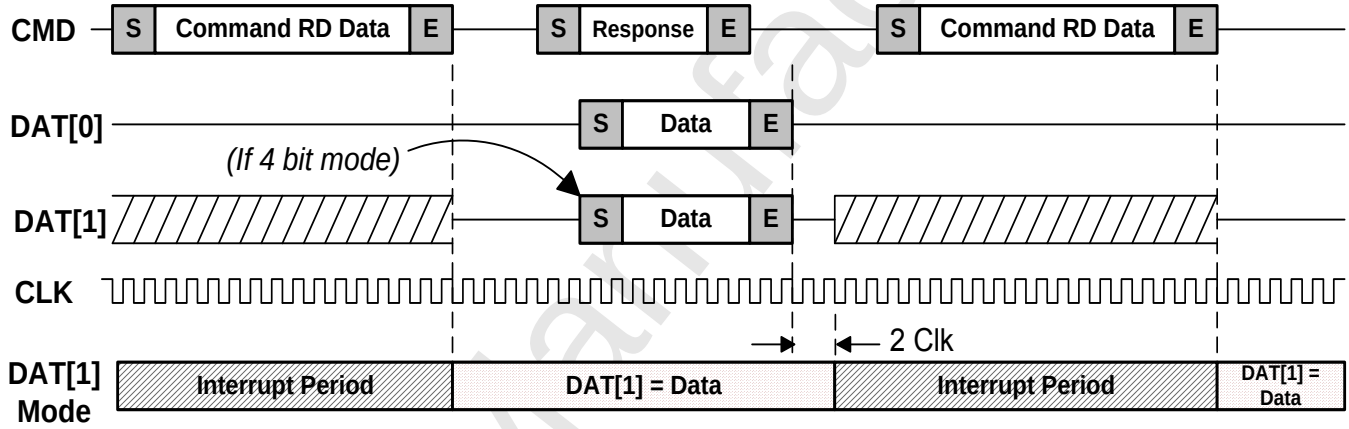


Figure 8-1 : Read Interrupt Cycle Timing

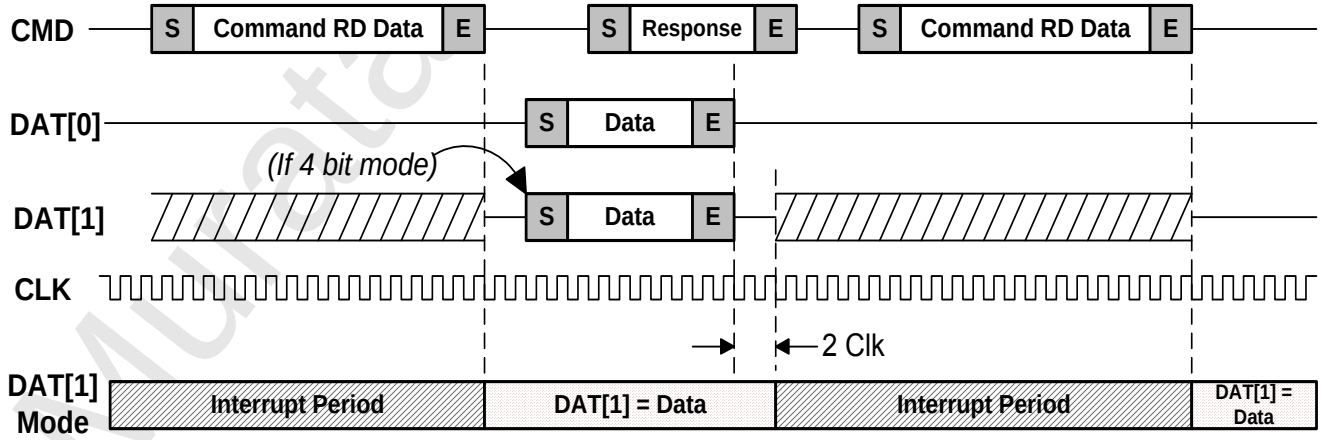


Figure 8-2 : Alternate Read Interrupt Cycle Timing

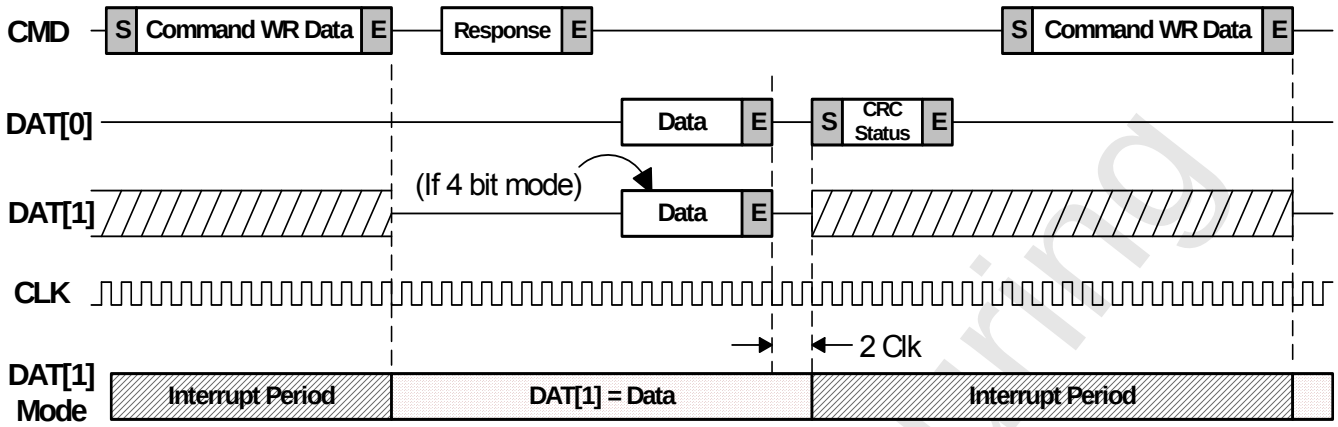


Figure 8-3 : Write Interrupt Cycle Timing

Figure 8-4 describes another case of the Interrupt Period. This case is where a command is followed by neither data blocks nor a response after the current command. In this case, the end bit timing of this command should not terminate the Interrupt Period

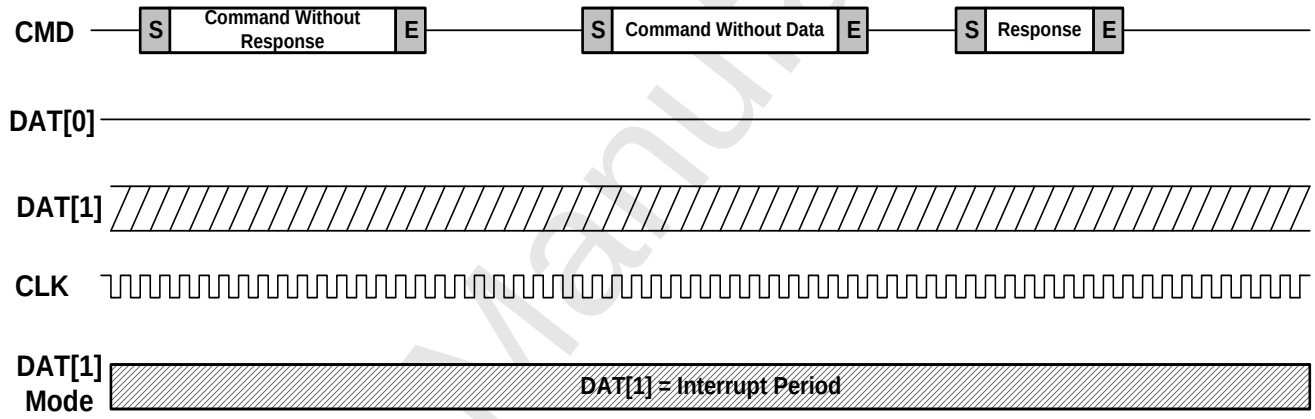


Figure 8-4 : Continuous Interrupt Cycle

8.1.4 Interrupt Period at the Data Block Gap in 4-bit SD Mode (Optional)

Due to the tight timing and limited data line availability when transferring multiple blocks of data in the 4-bit SD mode, a special definition of the Interrupt Period is required. An additional Interrupt Period is optionally available between data blocks in the 4-bit data transfer mode. The Interrupt Period(s) defined elsewhere are always available for an SDIO card to signal interrupts in all transfer modes.

In order to allow the highest possible data throughput, the Interrupt Period is limited to a 2-clock Interrupt Period that begins 2 clocks after the End bit of the data blocks. This Interrupt Period is the same for Read and Write operations. Because only 2 clock cycles are available in the Interrupt Period, any card that wishes to signal an interrupt to the host during 4-bit multi-block operations shall assert DAT[1] low for the first clock and high for the second clock. The card shall then release DAT[1] into the hi-Z State. Figure 8-5 shows this operation for an interrupt during a 4-bit multi-block read and Figure 8-6 illustrates the interrupt during a write sequence.

It is important to note that if a combo card supports interrupts, then the interrupt can occur during data transfers by both the I/O and the memory portions of the card. It is also important to note that since the interrupt can occur only between block of read or write data in the 4-bit mode, certain conditions may

cause very large delays in signaling the interrupt to the host. For example, if the host stops read data by stopping the clock, or delays the transmission or write data blocks, the interrupt is delayed until after the block of data is transferred.

An important point to note is that the Interrupt Period occurs during the same time that the Start bit could be placed on the bus by the card for a read operation. In order to allow the host to differentiate an interrupt from a Start bit, the card shall maintain the DAT[0], DAT[2] and DAT[3] lines in the hi-Z State during the Interrupt Period. The host can then differentiate the interrupt from a start bit by looking for a low on all 4 lines (DAT[3:0]) (Start bit) or a low only on DAT[1] (IRQ).

After sending the last data block, the Interrupt Period starts two cycles after the End bit of the last data block, and ends after the End bit of the next command, which is identical to the definition described in Section 8.1.3.

Card support of the 4-bit SDIO Interrupt on multiple block data transfers as described in Section 8.1.4 is optional. In the case of combo cards (SD memory plus IO) that support 4-bit interrupts, to avoid the conflict between interrupt cycle and data transfer, it is necessary to design memory control to be able to insert an interrupt cycle between data blocks

Hosts can determine if a card supports this mode by reading the S4MI bit in the CCCR as described in Table 6-1.

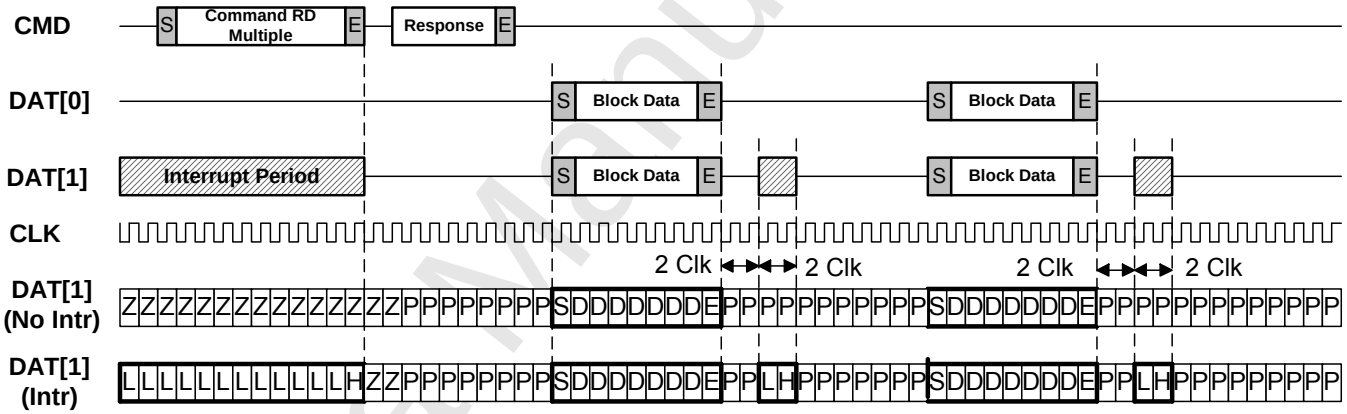


Figure 8-5 : Multiple Block 4-Bit Read Interrupt Cycle Timing

Important Note: The SDIO Specification Version 1.00 and 1.10 indicated that the state of the DAT[x] lines in the 4-bit read multi-block would be hi-Z (Z) between data packets. In order to match the signaling of the SD memory cards, the value was changed to pull-up (P). The reader is directed to the Physical Layer Specification for information on which device (Card or Host) drives the P state at any given time in an operation.

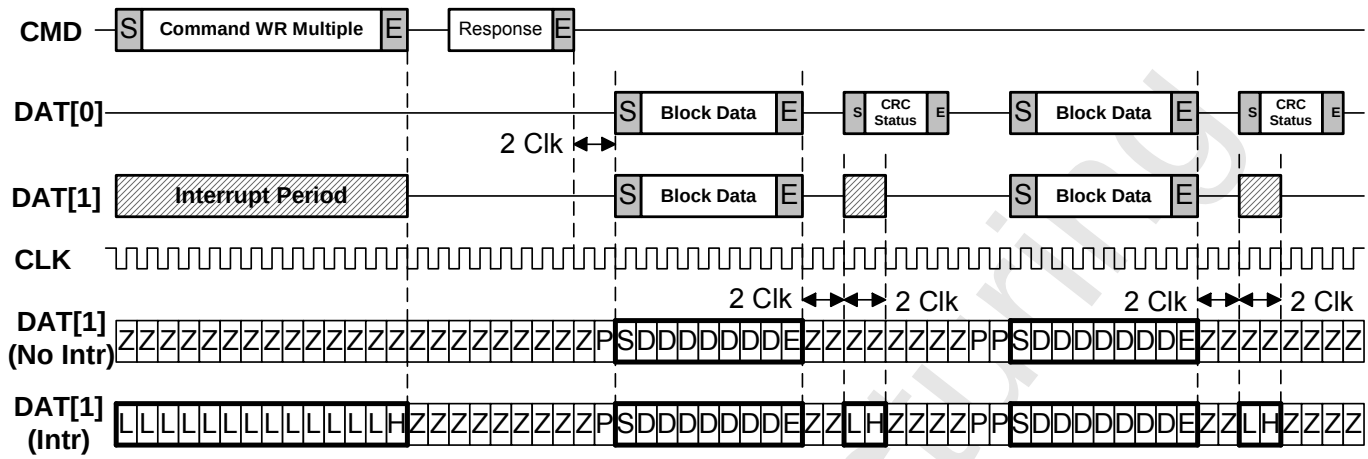


Figure 8-6 : Multiple Block 4-Bit Write Interrupt Cycle Timing

8.1.5 End of Interrupt Cycles

In the SD 4-bit mode the card shall drive DAT[1] high for a minimum of 1 clock cycle before terminating the Interrupt Period. This signaling is shown in Figure 8-7 : and shows the situation where the period of interrupt signaling is shorter than that of the Interrupt Period. In this example, the interrupt is cleared by a CMD52 write operation prior to the end of the Interrupt Period. The SDIO card can release the interrupt in one of two possible ways. Option #1 is for the SDIO card to place the DAT[1] into a hi-z after the single clock time of driving DAT[1] high (H). Option #2 is for the SDIO card to keep the DAT[1] line high (H) until the end of the Interrupt Period and then place the line into the hi-z state. Figure 8-8 shows the case where interrupt is de-asserted at the end of the Interrupt Period. In this case, the DAT[1] is driven to a high level prior to the card releasing the line to the hi-Z state.

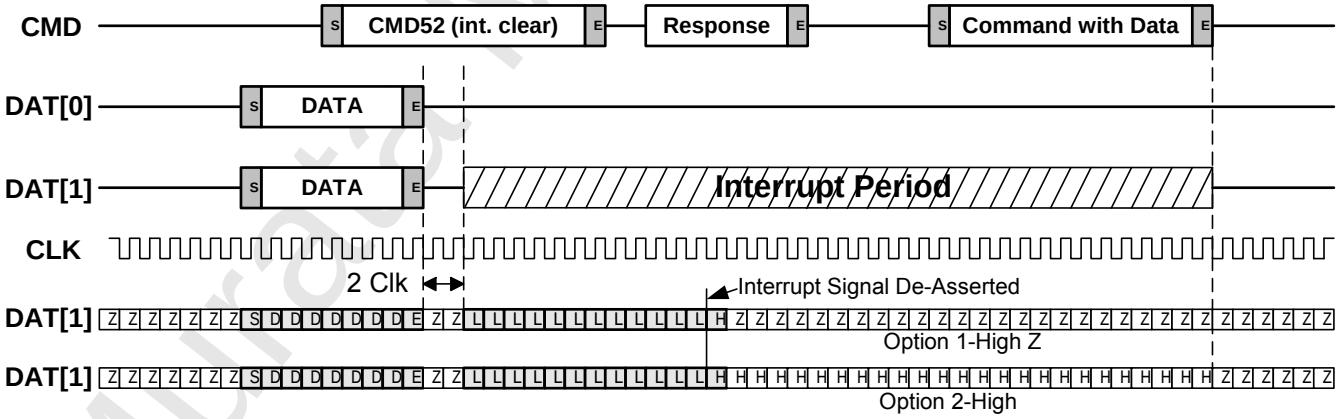


Figure 8-7 : Interrupt Cycle Timing

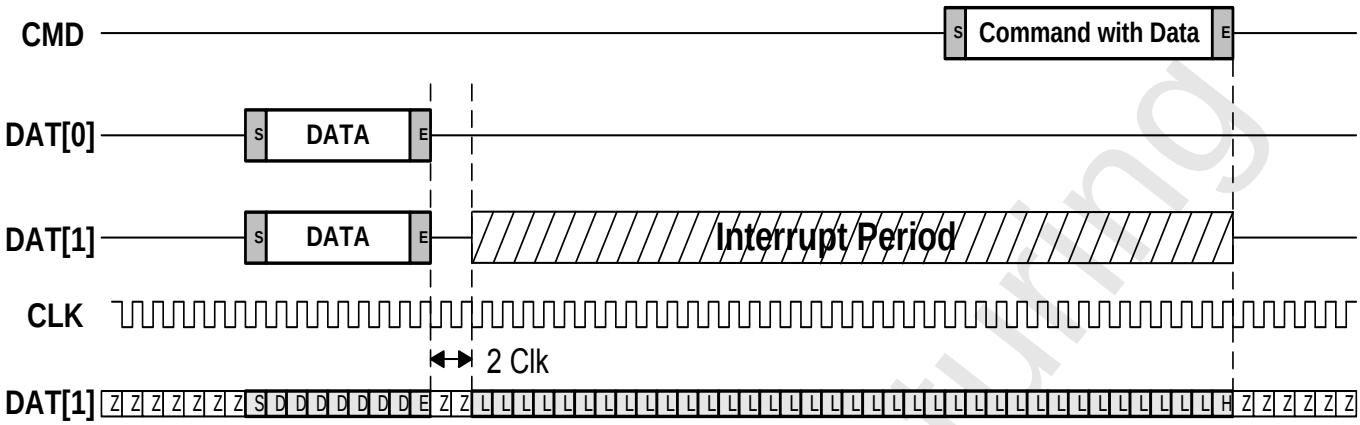


Figure 8-8 : Alternate Interrupt Cycle Timing

8.1.6 Terminated Data Transfer Interrupt Cycle

In the case of the Read Multiple Block or Write Multiple Block commands to the memory of a combo card, a special case for the timing of the Interrupt Period is created when the transfer is terminated. The Read Multiple and Write Multiple commands cause the card or host to transmit blocks of data until the operation is terminated by the host issuing a CMD12 or I/O Abort. In the case of a host transfer abort (memory or I/O), the timing of the Interrupt Period is the same. The Interrupt Period (if active) ends 2 clocks after the end bit of the stop transmission command (CMD12 or I/O Abort). The Interrupt Period re-starts 2 clocks after the end bit of the response. There are 2 cases for the Interrupt Period, depending on when the host issues the CMD12 or I/O Abort. Figure 8-9 shows the timing of the Interrupt Period for a Read Multiple command if the stop transmission is issued after a block of data has begun. In this case, the normal Interrupt Period begins 2 clocks after the end of the last data block. The Interrupt Period again begins 2 clocks after the end of the response to the transfer abort.

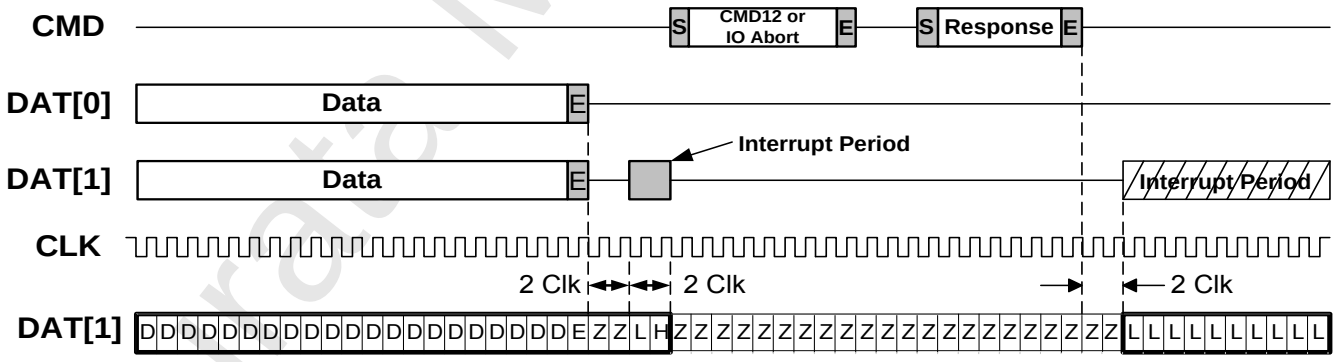


Figure 8-9 : Terminated Read Multiple Interrupt timing (Case 1)

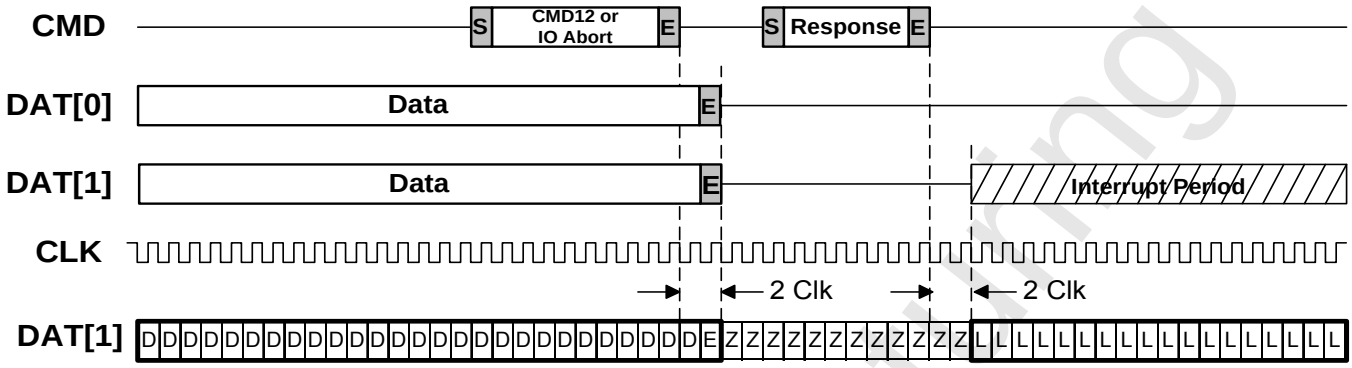


Figure 8-10 : Terminated Read Multiple Interrupt timing (Case 2)

Figure 8-10 shows a second case where the read multiple transfer was terminated during data transfer and the effect on the Interrupt Period. The read data transfer is terminated by a CMD12 (memory) or an abort I/O write (CMD52). One clock after the abort, the data shall terminate and the card shall send a single end bit before releasing the data bus. Two clocks after the response end, the Interrupt Period begins when the card can signal interrupt by asserting a 0 (L) on the DAT[1] line.
The termination of a Write Multiple command has the same timing as shown in Figure 8-9 and Figure 8-10.

8.1.7 Interrupt Clear Timing

Since the SDIO card uses level sensitive interrupts, the host shall clear pending interrupts with an I/O read or write to some function unique area. In some host implementations, the sending of a CMD52 to the card is handled by host adapter hardware while the host CPU can execute other operations. This condition may allow an interrupt that has already been handled to re-interrupt the host if the timing of the interrupt clear is not controlled. To prevent this condition, Any SDIO card that implements interrupts shall follow some required timing with respect to removing the interrupt from the DAT[1] line after the write to the function unique area that clears the interrupt. The clearing of the interrupt can be caused by an I/O write in a function unique method, or by a function unique I/O read. An example of clearing an interrupt using an I/O read would be a function where the reading of a data register may automatically clear the data ready interrupt. Since the clearing of the pending interrupt may be caused by a read or write operation to function unique areas of the SDIO card, the following requirements are placed on the card:

- In the case of a CMD52 read or write, the interrupt shall be removed from the DAT[1] line prior to the end of the R5 response.
- In the case of a CMD53 read or write the interrupt shall be removed prior to the end of the data response (in write operations) or the end of CRC in the data token (in read operations) for the data block that cleared the interrupt.

8.2 Asynchronous Interrupt

Figure 8-11 shows timing of Asynchronous Interrupt Period in a multiple block write operation. If support bit of SAI is set to 0, writing to EAI is ignored and EAI is set to 0. A Synchronous Interrupt Period is started 2 clocks after from the last data block. If EAI is set to 1 in the CCCR, the Asynchronous Interrupt Period is started 4 clocks from the start of the Synchronous Interrupt Period. During the Asynchronous Interrupt Period, the host can stop providing an SD clock to the card and the card can generate an interrupt without the SD clock. To issue the next command, the host starts supplying an SD clock. 4 clocks after the start bit of the next command; the Asynchronous Interrupt Period ends and goes back to the Synchronous Interrupt Period.

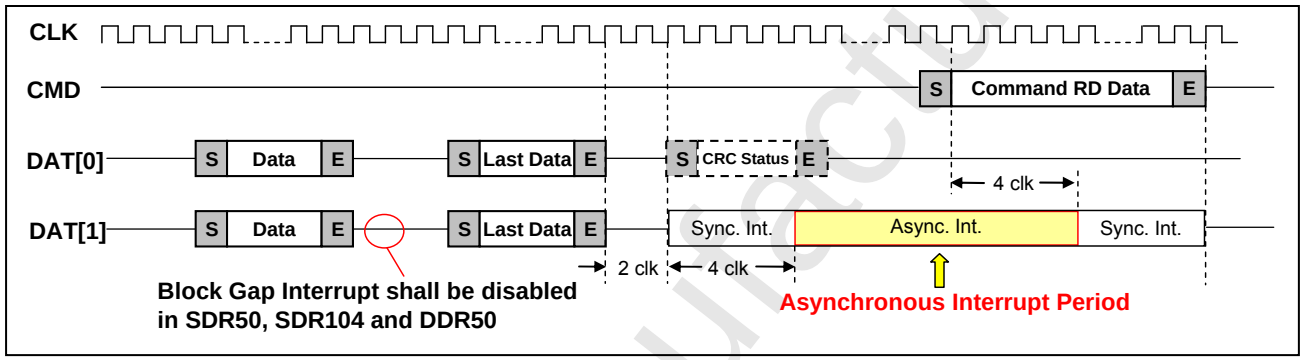


Figure 8-11 : Asynchronous Interrupt Period

9. SDIO Suspend/Resume Operation

The procedure used to perform the Suspend/Resume operation on the SD bus is:

1. The host determines which function is currently using the DAT[3:0] line(s).
2. The host requests the lower priority or slower transaction to suspend.
3. The host checks for the transaction suspension to complete.
4. The host begins the higher priority transaction.
5. The host waits for the completion of the higher priority transaction.
6. The host restores the suspended transaction.

If the current transaction can accept suspend and the card receives a suspend command during Read Wait, it shall accept the suspend request.

The operation of suspend resume is diagrammed in Figure 9-1 to Figure 9-6. For these figures, showing suspend request using DIRECT command in the Read after Write (RAW) mode, suspend is requested by setting BR=1 at write, and in the response, a return of BS=0 indicate bus release. BR is automatically cleared by the condition BS=0.

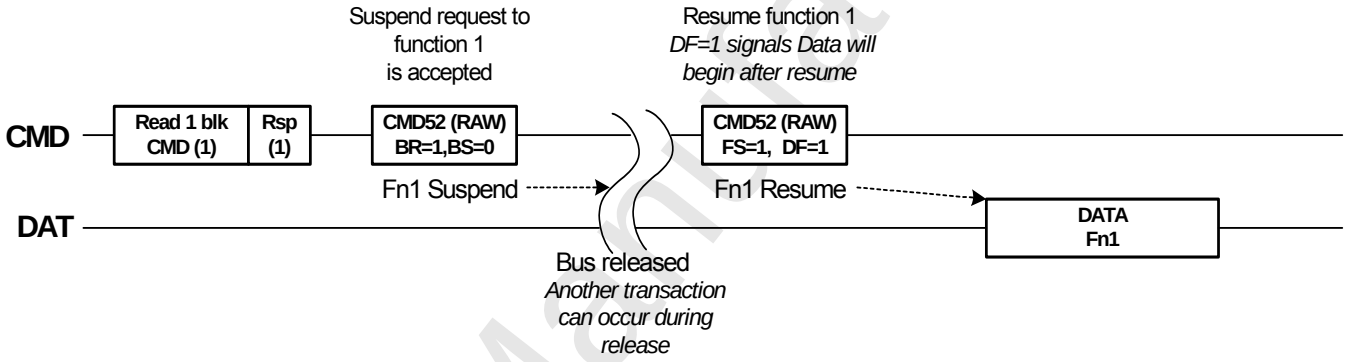


Figure 9-1 : Card with long read latency shall accept bus suspend

Figure 9-1 shows a card with a long read latency receiving and accepting a suspend request prior to the start of data transfer. The request is accepted and the bus is free for another transaction. At the end of this transaction, the resume causes the delayed read transfer to begin.

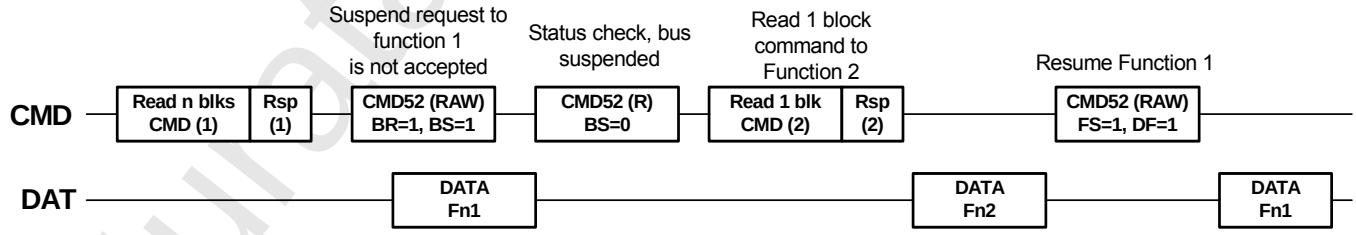


Figure 9-2 : Function2 read cycle inserted during Function1 multiple read cycle

Figure 9-2 shows a condition where the first suspend request is not immediately accepted. The host then checks the status of the request with a read and determines that the bus has now been released (BS=0). At this time, a read to function 2 is started. Once that single block read is complete, the resume is issued to function, causing the data transfer to resume (DF=1).

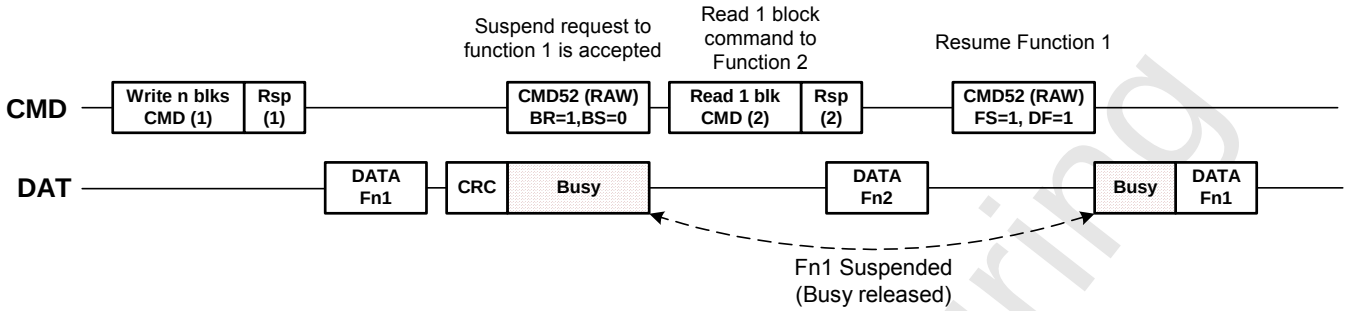


Figure 9-3 : Write suspended during busy (case 1)

Figure 9-3 shows a write operation that causes a busy condition. The host then suspends that transaction and begins another transaction to function 2. Once that transaction is complete, the host resumes the suspended function 1, which is still busy. Once the busy is clear, the write data transfer begins again.

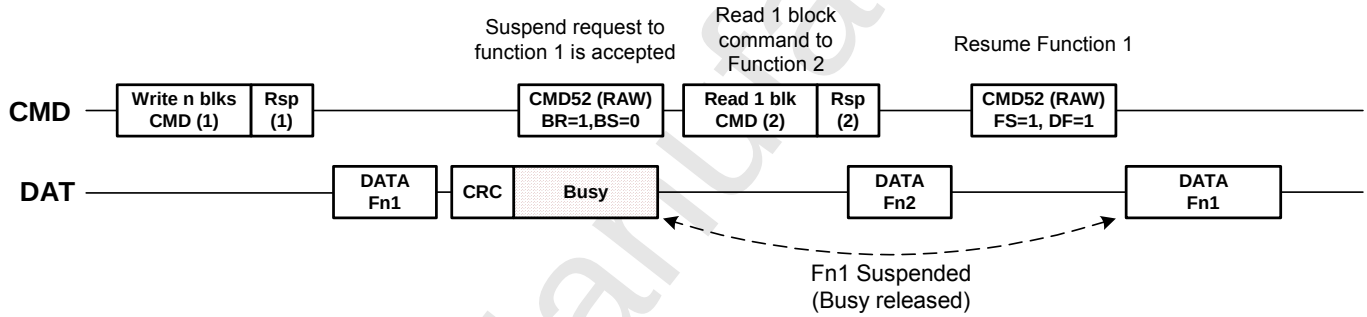


Figure 9-4 : Write suspended during busy (case 2)

Figure 9-4 is identical to Figure 9-3 except that the write busy was terminated by function 1 during the time it was suspended. In this case, upon resume function 1 begins to transfer write data immediately.

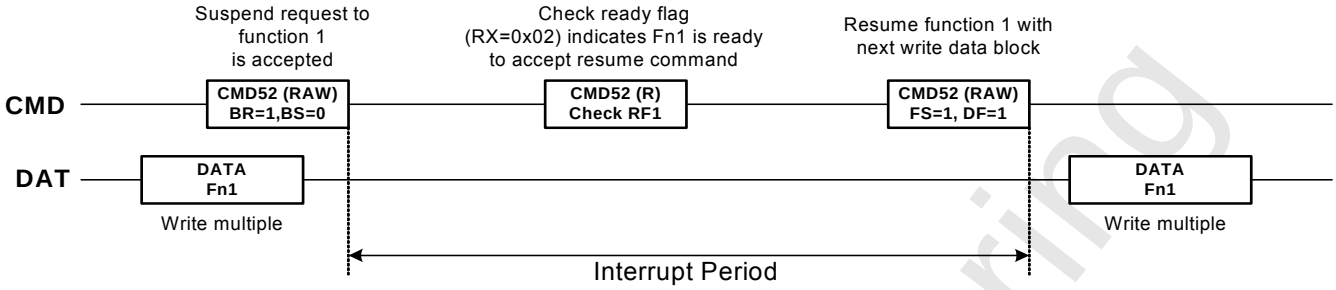


Figure 9-5 : Relationship between Interrupt Period and Suspend/Resume (case 1)

Figure 9-5 shows the relationship between the Interrupt Period and the suspend/resume operation. In this case, the Interrupt Period begins after the suspend request is accepted. The Interrupt Period continues until the resume is executed with data pending (DF=1).

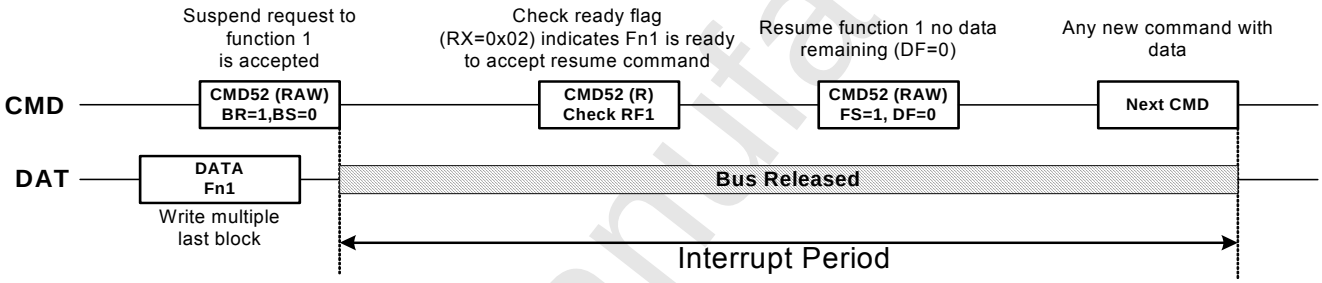


Figure 9-6 : Relationship between Interrupt Period and Suspend/Resume (case 2)

Figure 9-6 is similar to Figure 9-5 but in this case, there is no additional data to transfer after function 1 is resumed. In this case, the Interrupt Period shall extend until there is a command that requires data transfer (the next command in this example).

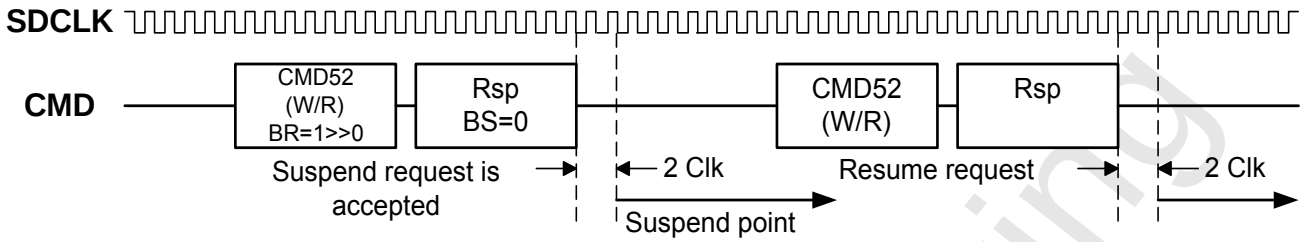


Figure 9-7 : Suspend/Resume timing

Figure 9-7 shows the timing of the actual suspend and resume operations. The suspend point is 2 clocks after the accepted suspend request response (BS=0). At the suspend point:

- If a function is executing, the function suspends
- The Interrupt Period re-starts
- If the Busy signal (DAT[0]) is asserted (low), it is de-asserted

The resume point is 2 clocks after the response to the resume command. At the resume point:

- If the selected function is suspended, the function resumes
- If DF=1, the Interrupt Period ends
- If DF=0, the Interrupt Period continues
- If the selected function is still busy, the busy signal (DAT[0]) is re-asserted (low).

10. Timing to Pause SDIO Read Operation

For pausing a read operation, there are two methods: first is stopping the SDCLK and the second is using the Read Wait signal.

10.1 Pause Read Operation by Stopping SDCLK

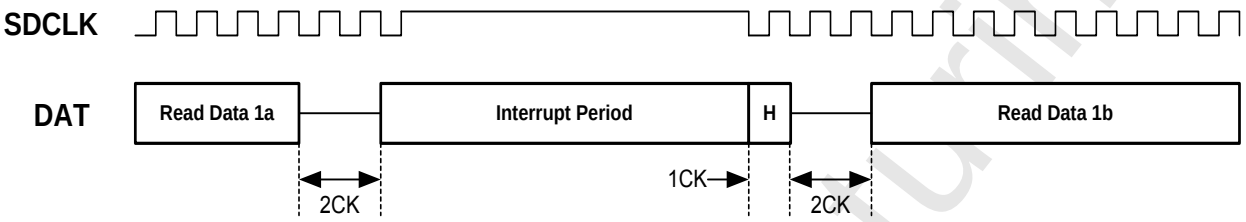


Figure 10-1 : Read Wait Control by Stopping SDCLK

Figure 10-1 shows the effect of the host stopping the SDCLK in order to stop read data. As usual, the Interrupt Period begins 2 clocks after the end of data block 1a. In this case, the host wishes to temporarily stop the read data. It does this by stopping the SDCLK after the 2-clock wait period. Because the card has no clock, it cannot begin to send the next block of data and the Interrupt Period is stretched until the clock is resumed. When ready, the host resumes by restarting the SDCLK signal. Since the end of the Interrupt Period is determined by the SDCLK, it ends as the clock resumes. 2 clocks after the end of the Interrupt Period, the card begins to transmit the next block of read data. The limitation of this method is that with the clock stopped, the host cannot issue any commands, and so cannot perform other operations during the delay time.

10.2 Pause Read Operation by using Read Wait

The optional Read Wait (RW) operation is defined only for the SD 1-bit and 4-bit modes. The read Wait operation allows a host to signal a card that is executing a read multiple (CMD53) operation to temporarily stall the data transfer while allowing the host to send commands to any function within the SDIO card. To determine if a card supports the Read Wait protocol, the host shall test 08h: SRW capability bit in the Card Capability byte of the CCCR (Refer to Table 6-1). The timing for Read Wait less than 50MHz is based on the Interrupt Period that is defined in Section 8.1

If a card does not support the Read Wait protocol, the only means a host has to stall (not abort) data in the middle of a read multiple command is to control the SDCLK. Read wait support is mandatory for the card to support suspend/resume.

10.3 Read Wait Timing (Less then 50MHz)

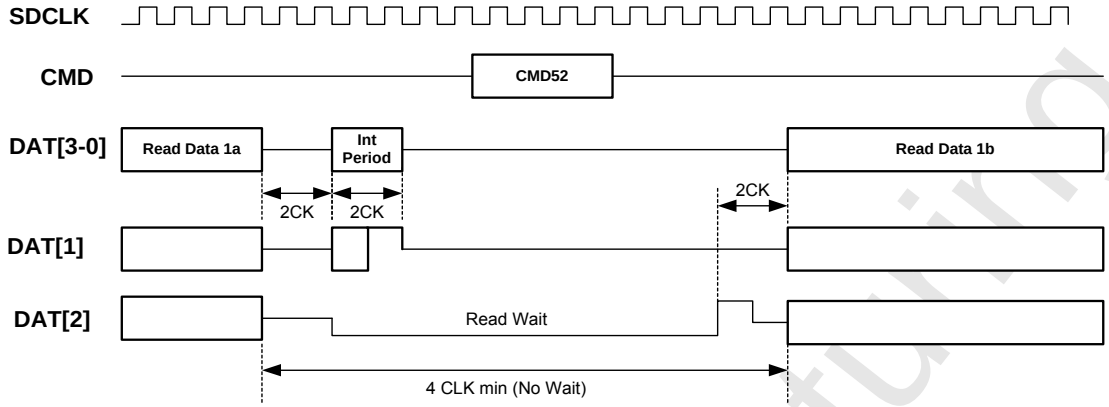


Figure 10-2 : Read Wait Delay using DAT[2] for Less Than 50MHz

Figure 10-2 shows a Read Wait operation using DAT[2] to pause the read operation at the block gap. This timing is effective in Default Speed, High Speed, SDR12 and SDR25. During the Interrupt Period, which starts 2 clocks after the end of the data block, the host can signal the card to enter the read wait state. This is done by asserting DAT[2] low during the Interrupt Period. The card detects the low on DAT[2] and stalls the data transfer at that point. The host can maintain the low on DAT[2] for as long as it needs to communicate with other functions. During the Read Wait time, the host can communicate with any function using CMD52. This allows the host to temporarily stall the data transfer from one function or memory and access other functions. In order to restart the stalled transfer, the host shall raise DAT[2] to a high for 1 clock and then release it. At that time, the card restarts the stalled transfer. If a card supports the Read Wait protocol, it shall allow a minimum of 4 clocks between data blocks rather than the normal 2 in order to allow times for the Read Wait signaling.

Because an SDIO card that supports Read Wait shall monitor the status of the DAT[2] line during the Interrupt Period between the data blocks of a read multiple command, there is a difference in the DAT[2] line drive for these cards. If an SDIO card does not support the read wait operation, it shall drive the DAT[2] line to the P (pull-up) state between data blocks. If the SDIO card supports Read Wait (08h: SRW=1), it shall release DAT[2] to the hi-z state after the end bit of the data block. This allows the host to signal the read wait without a driver conflict between the host and the card. Figure 10-3 shows the timing of the DAT[2] line for a card that supports read wait.

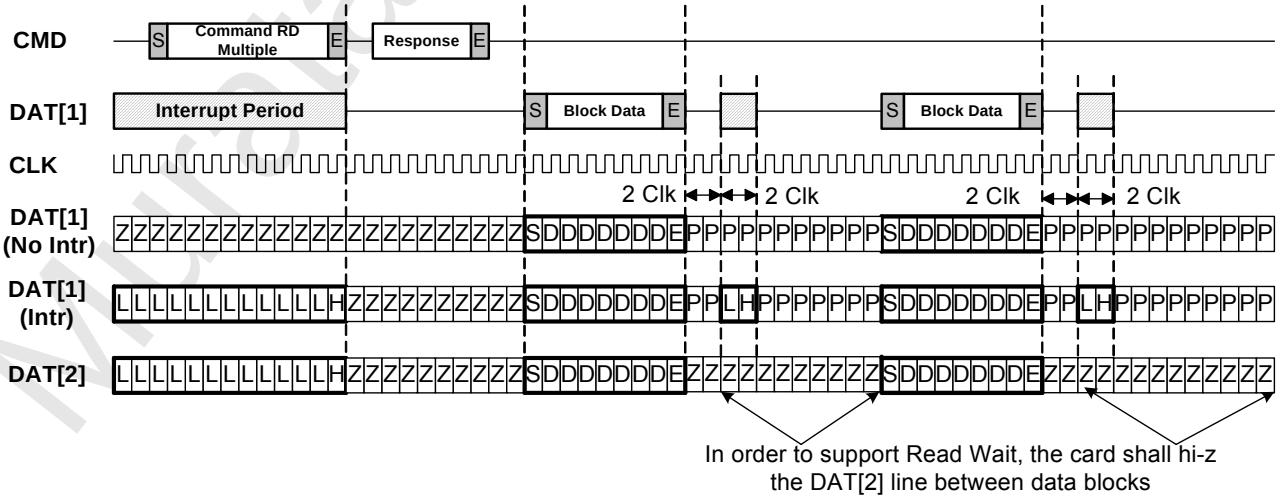


Figure 10-3 : SDIO Card DAT[2] Drive Timing for Read Wait

10.4 Read Wait Timing (More than 50MHz)

In the case of SDR50 and SDR104 (more than 50MHz), it is difficult to control the Read Wait signal in one clock period considering propagation delay between host to card. Therefore, the card shall treat the Read Wait as an asynchronous input. As the block gap interrupt becomes very narrow it cannot be used. Figure 10-4 shows Read Wait timing for operating more than 50MHz. Read Wait can be driven to low 2 clocks after and within 5 clocks from the end bit of a data block. The minimum block gap is defined as 8 clocks. The card needs to synchronize the Read Wait signal and pause sending the next data block while Read Wait is driven to low.

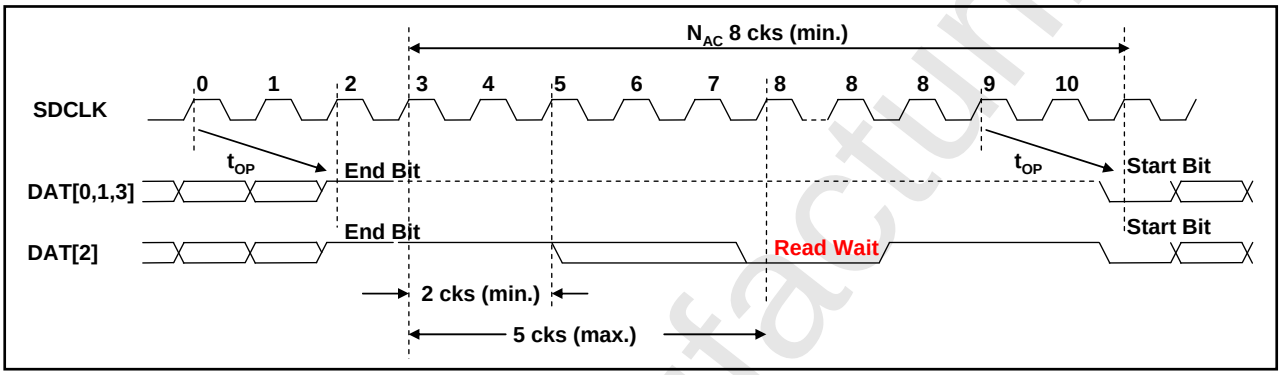


Figure 10-4 : Read Wait Timing for More Than 50MHz

11. Power Control

11.1 Power Control Overview

The concept of High-Power SDIO cards and its Power Control method was introduced in the SDIO Specification Version 1.10. The Power State Control is defined by the SDIO Version 3.00 to provide more flexible power control.

The temperature requirement is also updated by SDIO Version 3.00. SDIO cards created prior to Version 3.00 shall operate under the temperature requirement of Ta=85 deg.C (Ta is based on heat removal through the air without air blow). However, High Power SDIO cards may generate more heat and thus host devices will need to support another heat removal method. The Thermal Specification, which is defined by the Mechanical Addenda Version 3.00, specifies the Card Case Temperature (Tc) as the new temperature requirement and the concept of heat removal method through materials (Refer to the Mechanical Addenda Version 3.00 for more details). The concept of the Thermal Specification is also applied to SDIO.

Power Control supports following two features:

- **High-Power Support**

  SDIO cards created prior to Version 1.10 of the SDIO Specification were limited to a maximum current of 200mA at any time, irrespective of the number or types of functions supported. With the creation of wireless communication devices in the SDIO form factor, a need was seen to provide more current to accommodate the higher power requirements of some SDIO cards. Since backward compatibility is a primary concern for any changes made to this specification, a method was chosen to prevent a high-power card from drawing excessive amounts of current from hosts designed to only support the SDIO Version 1.00 cards. The Master Power Control prevents excessive current damage to the host when a high-power cards to be inserted into any host. It is important to note that there exists the possibility of trying to use a card that requires high-power in a standard power host and having that card fail to operate. The Master Power Control is supported on a per card basis and available to the host in the CCCR. A high-power card may have a mix of both high and standard power functions. Support of the Master Power Control is mandatory for the SDIO Version 3.00 Card, which has an operating mode more than 720mW (3.6V x 200mA).

- **Power Tuning Support**

  Support of Power Tuning capability is useful for the hosts. For example, even though the power supply capability is limited depending on the hosts, support of the Power Tuning enables the widest range of host support for high-power cards. Operating in lower power mode increases operation time.

  Two types of Power Tuning can be used in SDIO Version 3.00 and higher.

  (1) Power Selection
  Support of Power Selection is optional. The two-level Power Selection is available by EPS in FBR (per function basis) for an SDIO Card supporting 2.7V-3.6V Power supply range. The maximum current of a card is up to 500mA and the card also needs to satisfy the temperature requirement of Ta=85 deg.C.

  (2) Power State Control
  Support of Power State Control is optional. Up to 15 power states can be selected by PS[3:0] in FBR (per function basis). The temperature requirement is defined by Tc basis and the maximum power of a card is restricted by Tc (Refer to Section 11.2.5). Usage of the Power State is recommended rather than using EPS due to the flexibility of power control.

11.2 Details of SDIO Power Control

11.2.1 Master Power Control

SDIO version 1.10 cards indicate their support for the power control functions with the SMPC (Support Master Power Control) bit in the CCCR (Refer to Section 6.9). Hosts enable the card's power control functions with the EMPC (Enable Master Power Control) bit.
SMPC can be set to 0 if the card maximum power is less than 720mW (3,6V x 200mA) and Power Selection (Refer to Section 11.2.2) is not supported. A SDIO version 1.10 card which has SMPC set to 1 shall maintain backward compatibility when EMPC is set to 0. A SDIO version 1.0 host may not be aware of EMPC, which will remain 0 (its default state).

In the case where EMPC is set to 0, the card total power shall not exceed 720mW (3,6V x 200mA). Functions that exceed 720mW shall not set IORx to 1 and TPLFE_OP_MAX_PWR shall be set to 0. If a multi-function card's total power exceeds 200mA the card shall not set all IORx to 1, even if all IOEx are set to 1. Some of the functions' IORx can be set to 1 as long as the card's total current is less than 720mW. If the host tries to enable a function (IOEx =1) that will cause the card's total current to exceed 200mA, the card shall disable (IORx=0) one or more functions to keep the card's total current less than 200mA. Which functions are enabled is implementation dependant.

In the case where EMPC is set to 1, the card power can exceed 720mW. SDIO Cards should be designed, where possible, to not require the maximum current, thus functioning in as many hosts as possible with sufficient power. One of the following Power Tuning features should be supported.

11.2.2 Power Selection

If Power State Control is disabled (PS[3:0]=0 in FBR), Power Selection is supported by the SPS bit (Support Power Selection) in the FBR and the EPS bit (Enable Power Selection) in the FBR.
Power Selection defines two power modes for a function: Lower Current Mode and Higher Current Mode. A card implementing Power Selection gives the host the choice between these two power modes. These modes can be used for functions, which can operate in full performance (EPS=0: Higher Current Mode) or reduced performance (EPS=1: Lower Current Mode).
The Power Selection can be applied to an SDIO Card supporting 2.7V-3.6V Power supply range because the High Power Tuple is defined only for 3.3V supply voltage.

11.2.3 Power State Control

The Power State Control provides more flexible power control than the Power Selection. A SDIO function can define up to 15 power states that are specified by the Power State Tuple. The tuple consists of a list of power consumption of each state. The Host chooses an affordable power from the Tuple and sets a power state number to PS[3.0] in FBR. The card shall operate below the power specified in the Power State Tuple corresponding to the power state number. Since the Power State Control is on a per function basis, the total power of the card is calculated by the sum of the power of enabled functions. In case of Combo Card, adding the current of a memory portion is needed that is indicated by CMD6. If a host cannot provide the estimated total card power, host needs to disable some of the functions or select smaller power state. There is another requirement that the total power of the card shall be smaller than that calculated from the maximum card power restricted by Tc (Refer to Section 11.2.5).
The Power State can be applied to not only single voltage device but also multiple voltages device.

11.2.4 High-Power Tuples

Six tuple fields are defined in Version 1.10 for each function tuple.

- Average power required when Master Power Control is not enabled (TPLFE_SP_AVG_PWR_3.3V)
- Peak power required when Master Power Control is not enabled (TPLFE_SP_MAX_PWR_3.3V)
- Average power required in Higher Current Mode (TPLFE_HP_AVG_PWR_3.3V)
- Peak power required in Higher Current Mode (TPLFE_HP_MAX_PWR_3.3V)
- Average power required in Lower Current Mode (TPLFE_LP_AVG_PWR_3.3V)

- Peak power required in Lower Current Mode (TPLFE_LP_MAX_PWR_3.3V)

Each tuple field is a 16 bit value with a 1mA/step resolution. This allows a value of 0 to 65,535 mA to be used. Current varies depending on the voltage. These tuple fields are defined as a 3.1-3.5V or 2.7-3.6V range.

A Power State Tuple is defined in version 3.00 for each function tuple.

- Maximum power at XX Power State for 3.3V device (TPLFE_PS_XX)

Each tuple field is a 16 bit value with a 1mW/step resolution. This allows a value of 0 to 65,535 mW to be used. The Power State is regardless of voltages and can be applied to a multiple voltages device.

Table 11-1 shows which tuples a host shall refer to depending on the host version and the settings of EMPC, EPS and PS[3:0].

Host Ver.	EMPC (CCCR)	PS[3:0] (FBR)	EPS (FBR)	Reference TPLs	Comments
1.00	0	0h	X	TPLFE_OP_MIN_PWR TPLFE_OP_AVG_PWR TPFLE_OP_MAX_PWR	For SDIO Version 1.00 Host up to 200mA 3.3V
1.10, 2.00 and 3.00	0	0h	X	TPLFE_SP_AVG_PWR_3.3V TPLFE_SP_MAX_PWR_3.3V	Same as TPLFE_OP_AVG_PWR Same as TPFLE_OP_MAX_PWR
	1	0h	0	TPLFE_HP_AVG_PWR_3.3V TPLFE_HP_MAX_PWR_3.3V	Non zero value shall be set when SMPC=1.
	1	0h	1	TPLFE_LP_AVG_PWR_3.3V TPLFE_LP_MAX_PWR_3.3V	Non zero value shall be set when SMPC=1 and SPS=1
3.00	1	Non 0	X	TPLFE_PS_XX	Multiple voltages device can be supported.

Table 11-1 : Reference of Tuples Depends on Host Version

If an SDIO card is set to SMPC to 1 it shall implement the CISTPL_FUNCE Tuple for Functions 1-7 (Extended Data 01h). A Non-zero value shall be set in TPLFE_SP_AVG_PWR_3.3V, TPLFE_SP_MAX_PWR_3.3, TPLFE_HP_AVG_PWR_3.3V and TPLFE_HP_MAX_PWR_3.3V. If the card is set to SPS to 1, a non-zero value shall be set in TPLFE_LP_AVG_PWR_3.3V and TPLFE_LP_MAX_PWR_3.3V.
TPLFE_SP_AVG_PWR_3.3V and TPLFE_SP_MAX_PWR_3.3V should be set to the same value as TPFLE_OP_AVG_PWR and TPLFE_OP_MAX_PWR respectively.

A version 3.00 SDIO card that supports power selection shall implement either of the tuples depending on supply voltage: CISTPL_FUNCE Tuple for Function 1-7 (Extended Data 02h) for 3.3V device or CISTPL_FUNCE Tuple for Function 1-7 (Extended Data 03h) for 1.8V device. Existence of the tuple indicates which voltage is supported. Dual voltage supported device may have both tuples.

11.2.5 The Maximum Card Power Restricted by Tc

The temperature specification of a default power consumption card up to 720mW (3.6Vx200mA) was specified by the ambient temperature Ta=85 deg.C and heat of the card is supposed to be removed through the air.
To use an SDIO card more than 720mW power consumption, Version 3.00 Host should support a heat removal method though materials specified by the Mechanical Addenda Version 3.00. According to the thermal specification defined in the Mechanical Addenda Version 3.00, Card Surface Temperature (Tc) is defined as the temperature requirement for a high-power supported host and card.

The maximum operating power of a card is restricted by the Tc the host supports. The lower Tc environment enables the card to consume more power. The list of card operable conditions is added optionally in the CISTPL_FUNCE Tuple for Function 0 (Common) that consists of the pairs of Tc and card power. Each pair indicates the card maximum consumable power under the Tc (Refer to Section 16.7.3 for more details about the tuple). The list is required for an SDIO card more than 720mW power consumption. The High Power Version 3.00 SDIO Card shall include the maximum operable power at the Tc=80 deg.C in the Tuple for Function 0 and shall include sections of power states less than or equal to 1440mW in the Power State Tuple so that UHS-I supported host (3.6V 400mA Tc=80 deg.C) can support the SDIO card.

Presuming that host has the two capabilities: Tc the host supports and the maximum removable power to keep the card surface temperature below the Tc. The Host scans the Tuple for Function 0 and compares the host and card capabilities, and then finds the maximum consumable power of the card under the Tc host supports.
For example, when a SDIO card, which indicates 1800mW (3.6Vx500mA) consumable power at Tc=80deg.C, is inserted to the host (3.6V 400mA at Tc=80 deg.C), the host needs to select less than 1440mW (3.6Vx 400mA) Power State to use the SDIO card. If the host supports heat removal of 1800mW at Tc=80 deg.C, the card can be used at the maximum power by selecting 1800mW Power State.

For Multiple Functions Card or Combo Card, host needs to estimate the total card power by adding the power of memory and the power of enabled functions and check whether the total card power shall be less than the maximum consumable power of the card. If not, the host needs to reduce the total card power by controlling the Current Limit of a memory, the Power State of each function or disabling a part of functions.

11.3 Power Control Support for the SDIO Host

11.3.1 Version 1.10 Host

The following are requirements for a version 1.10 host:

- The host shall recognize new Power Control registers and tuples as defined in SDIO Specification Version 1.10.
  - Power Control bits (SMPC, EMPC) in CCCR
  - Power Selection bits (SPS, EPS) in FBR
  - High-Power Tuples (Refer to Table 16-8)
- The host shall know its own power supply ability
- The host shall have the ability to manage power by calculating maximum current shown in the tuples and control EMPC, EPS and IOEx not to exceed total current that the host can supply. If the host does not have enough power to use the card, the host shall not enable the card.

11.3.2 Version 3.00 Host

The following are requirements for a version 3.00 host which supports over 720mW:

- The host needs to recognize new Power Control registers and tuples as defined in SDIO Specification Version 3.00.
  - Power Control bits (SMPC, EMPC) in CCCR
  - Power Selection bits (SPS, EPS) in FBR
  - Power State bits (PS[3:0]) in FBR
  - The Maximum Card Power restricted by Tc (Refer to Section 11.2.5)
  - High-Power Tuples (Refer to Table 16-8)
  - Power State Tuples (Refer to Table 16-12)
- The host shall know its own power supply ability and heat removal ability of Tc
- The host shall have the ability to manage power by calculating maximum current shown in the tuples

and control EMPC, EPS, PS[3:0] and IOEx not to exceed total current that the host can supply. The host also needs to manage to satisfy the maximum card power restricted by Tc. If the host does not have enough power to use the card, the host shall not enable the card.

Application Note:
Most card connectors limit the maximum current at 500mA. A card slot that can support more than 500mA will require a special connector to handle the higher current.
Host systems using an Embedded SDIO device may need to support a specific Tc requirement as defined by the device manufacturer. Since Embedded SDIO devices usually do not use normal card connectors, a heat removal method is easier to support than on a removable card and current limitations are also easier to contend with.

12. High-Speed Mode

High-Speed mode increases the bus clock rate to 50MHz and the SD bus throughput from 12.5MB/sec to 25MB/sec. SDIO and combo cards may also support High-Speed mode.

12.1 SDIO High-Speed Mode

SDIO version 1.20 cards indicate their support for High-Speed mode with the SHS (Support High-Speed) bit in the CCCR (Refer to section 6.9). Hosts switch between default and High-Speed mode with the EHS (Enable High-Speed) bit in the CCCR by CMD52. Following a command to set or clear EHS, cards shall switch speed mode within 8 clocks after the end bit of the corresponding response.

When switching from default to High-Speed mode the host can try to set EHS without first checking SHS. The host issues CMD52 in RAW mode, setting EHS to one, and after getting the response of CMD52 the host checks SHS and EHS. If SHS=0 or EHS=0, the command will be ignored and the card is still in default mode. If SHS=1 and EHS=1, the card is in High-Speed Mode.

12.2 Switching Bus Speed Mode in a Combo Card

A combo card that supports High-Speed shall support it for both memory and IO. Two bus speed switch commands are defined; SD memory command (CMD6) and SDIO command (EHS in CCCR is changed using CMD52). The timing of changing bus speed mode of the card is defined in Figure 12-1 and Figure 12-2. To avoid bus timing conflicts between the host and card, the relation of the two bus speed switch commands and the timing of changing host bus speed mode are defined as follows.

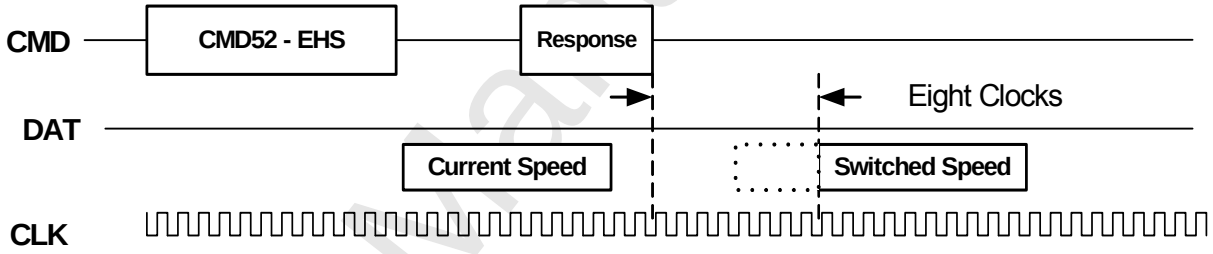


Figure 12-1 : Timing for CMD52 default <-> High-Speed switch

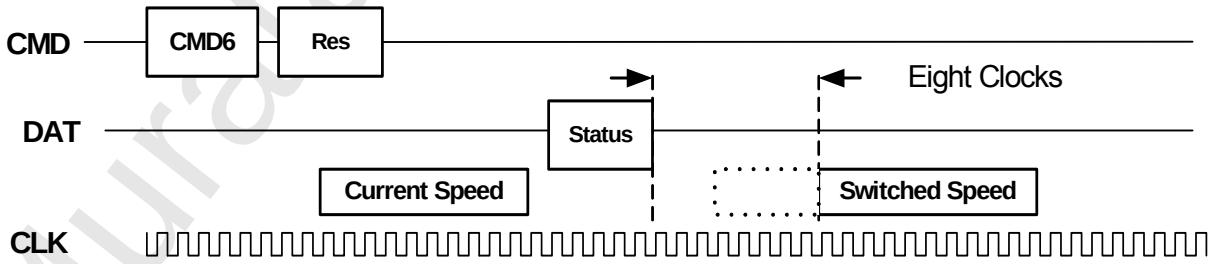


Figure 12-2 : Timing for CMD6 default <-> High-Speed switch

When one bus speed switch commands is executed successfully, the card switches the card bus speed mode. If two bus speed switch commands are executed in turn (to the same bus speed mode), only the first successful command is effective to switch bus speed mode.

The host needs to check the success of executing a bus speed switch command and then the host can switch the host bus speed mode to the same one.

Success of switching bus speed mode is determined by checking the receipt of a good response and the result of switching the bus speed mode is the same as the switch requested.

The status of current bus speed mode is read by the bus speed switch commands. For example, when the bus speed mode is switched by CMD6, the result can be read from EHS. If switching using the RAW mode of CMD52 has failed, there can be one of two responses. First is no response with an illegal command error. The second is that CMD52 is accepted and the status of RAW mode indicates EHS is not changed.

A reset of either the memory or IO portion of a combo card will also reset both portions to default speed mode. Within 8 clocks after the response of a reset by CMD52 (write to RES in CCCR) or CMD0 the card shall change the speed mode to default speed mode.

Note that when changing the bus speed the host bus driver should treat a bus speed change request from any driver as an atomic operation. The host should mask interrupts and not issue any command to the card until the bus speed change is complete.

If a combo card supports the Lock/unlock function, a locked card cannot change bus speed mode. A locked card indicates an illegal command error to a bus speed switch command. The host needs to unlock the card by CMD42 before changing bus speed. It also implies that a host should not change bus speed mode during initialization before managing a locked card.

13. SDIO Physical Properties

13.1 SDIO Form Factors

The SDIO definition encompasses different form factors:

- Full-Size SDIO - compatible with host sockets designed for SD memory cards
- miniSDIO - compatible with host sockets designed for miniSD memory cards
- microSDIO - compatible with host sockets designed for microSD memory cards

13.2 Full-Size SDIO

The SDIO card is compatible with host sockets designed for SD memory cards. In addition, the SDIO cards can be extended to allow for external connectors, antennas etc. With the exception of the write protect switch, all SDIO cards shall meet the mechanical specifications described in the Standard Size SD Card Mechanical Addendum for that portion of the card that is not extended.

A Full-Size SDIO card shall meet all requirements for Card Packaging identified in Section 3.1 of the Standard Size SD Card Mechanical Addendum except as noted in Table 13-1. The minimum size for any SDIO card is the standard 24mm by 32mm size of a SD memory card.

Title	Exceptions for SDIO
External signal contacts (ESC)	NONE
Design and format	32mm size limit does not apply to SDIO. In addition, any dimension limits do not apply in the SDIO extended area. Refer to Section 13.2.3.
Reliability and durability	Bending and Torque shall be measured at a point 32mm away from contact end irrespective of actual length. WP requirements apply only if WP switch installed (not supported on SDIO only cards). Drop test does not apply to SDIO cards.
Electrical Static Discharge (ESD) Requirements	Contact Pads test shall apply only to the 9 SD pins, not any additional vendor specific contacts. Non Contact Pads area for ESD discharge testing shall be in the 24mm by 32mm area of a standard SD card
Quality assurance	Not applicable, SDIO card support of unique ID is optional

Table 13-1 : Exceptions of SDIO Full-Size Card Regarding Mechanical Requirement

13.2.1 Full-Size SDIO Mechanical Extensions

In order to implement some function in the SD card form factor, some extensions to the standard card size and constructions may be needed. There are two areas of extension defined for Full-Size SDIO cards. Both of these extensions are optional, and may be used by card vendors based upon their needs. The two extensions are:

1. Additional optional ESD/EMI ground contact
2. Optional extended case dimensions

Figure 13-1 shows the details of both optional extensions.

13.2.2 Additional ESD/EMI Ground Point

For some SDIO cards, there may be a need for a lower impedance ground connection to the host. This may be needed to reduce the card's EMI emission or susceptibility. Also, an additional ground may be needed to discharge ESD on insertion or operation. If additional grounding is required, the card may implement an additional ground contact as shown in Figure 13-1. Note that additional ground contact this is optional for both the SDIO card and the SDIO host.

13.2.3 Full-Size SDIO Extended Case

In order to provide useful I/O devices in the SDIO form factor, it may be necessary to extend the size of the case to provide room for connectors, antennas etc. In order to provide the extension room and maintain compatibility with existing SD hosts, the SDIO extension area has been defined. Figure 13-1 shows this extension area. The SDIO card vendor may extend the case of its SDIO products within the area identified. Please note that there is no limit in this specification as to the amount or direction of growth in this area. The vendor is cautioned that extensions may cause interference problems with SD hosts, depending on the amount and direction of growth. The drawing shows 2 additional lock notches beyond the normal SD area. These notches are intended to provide additional retention for SDIO cards to prevent accidental withdrawal due to cables or additional weight of extended cases. The support of these notches is optional for both the host and the card vendor, but their use is highly recommended for both cards and hosts to prevent unintended disconnects.

13.2.4 Write Protect Switch

The WP switch is not supported by SDIO only cards. Because the host does not support the Write Protect function for SDIO only cards, it is suggested that these cards be built without cutouts or notches in the case at the write protect location. If the maker of an SDIO card chooses to add the Write Protect cutout, it shall be made to signal a write-enabled card (Refer to Figure 13-1). In case of a standard size combo card, the WP switch is mandatory. Refer to the Standard Size SD Card Mechanical Addendum about WP switch.

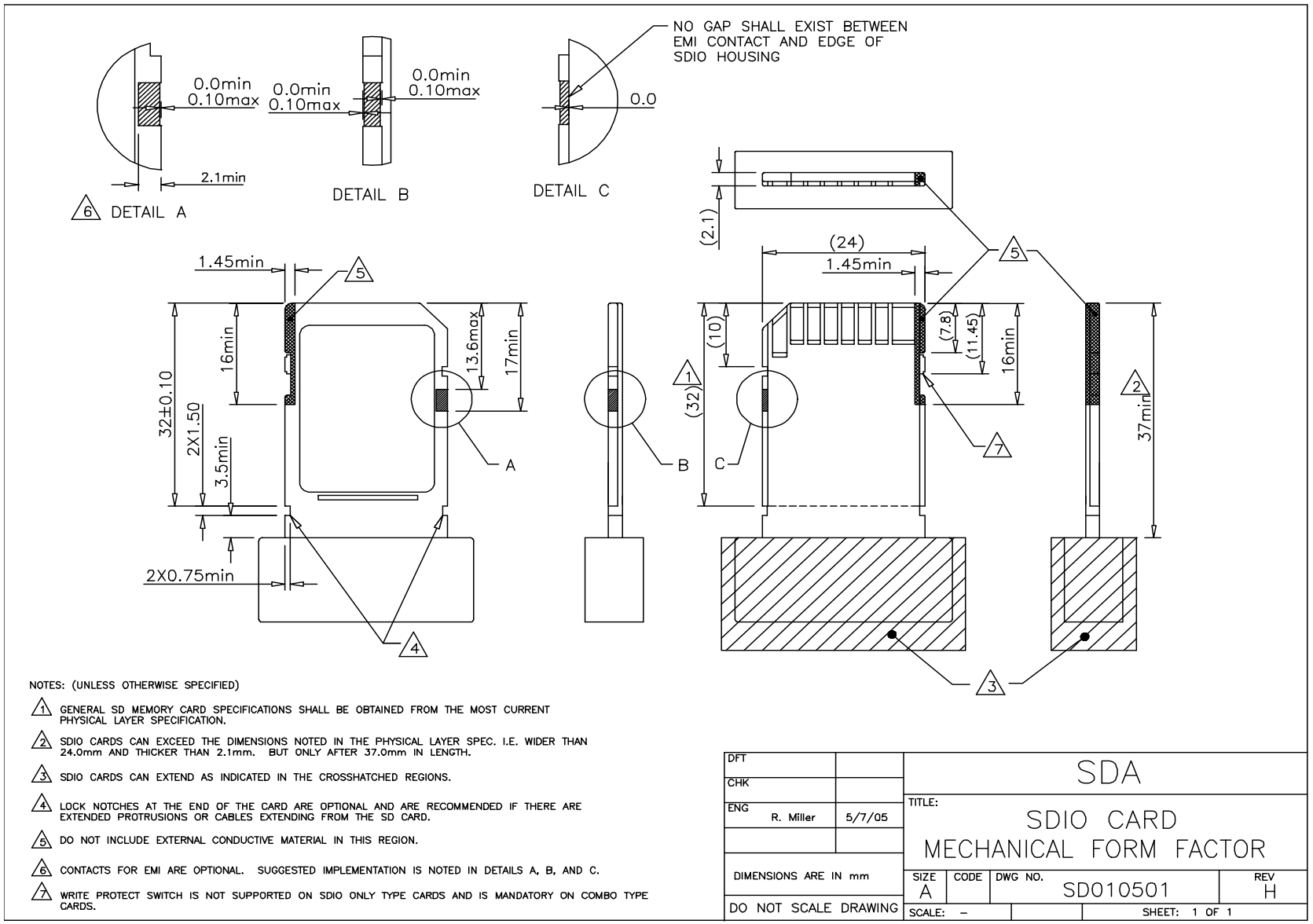


Figure 13-1 : Full-Size SDIO Mechanical Extensions

13.3 miniSDIO

The miniSDIO card is compatible with host sockets designed for miniSD memory cards. In addition, the miniSDIO cards can be extended to allow for external connectors, antennas etc. All miniSDIO cards shall meet the mechanical specifications described in *miniSD Card Addendum.*
When a miniSDIO card is inserted into a miniSD to SD memory card adaptor, and plugged-into an SDIO host slot, it shall appear to the host as an SDIO card.

The use of pins 10 and 11 (Refer to Table 2-1) are not yet defined for miniSDIO.

13.3.1 miniSDIO Mechanical Extensions

In order to provide useful I/O devices in the miniSDIO form factor, it may be necessary to extend the size of the case to provide room for connectors, antennas etc. In order to provide the extension room and maintain compatibility with existing miniSD hosts, the miniSDIO extension area has been defined.

Figure 13-2 shows this extension area. The miniSDIO card vendor may extend the case of its miniSDIO products within the area identified. Please note that there is no limit in this specification as to the amount

or direction of growth in this area. The vendor is cautioned that extensions may cause interference problems with miniSD hosts, depending on the amount and direction of growth. There are 2 lock notches outside of the normal miniSD memory card area. These notches are intended to provide additional retention for miniSDIO cards to prevent accidental withdrawal due to cables or additional weight of extended cases. The support of these notches is optional for both the host and card vendor, but their use is highly recommended to prevent unintended disconnects.

Note that these notches are located at the same distance from the pins' edge of the card as in the full-size SDIO card, and have the same width and depth.

13.3.2 Additional ESD/EMI Ground Point

Unlike Full-Size SDIO, the miniSDIO form factor does not support additional ESD/EMI ground points.

13.3.3 Write Protect Switch

The WP switch is not supported by either cards or hosts for either miniSDIO cards or miniSDIO combo cards.

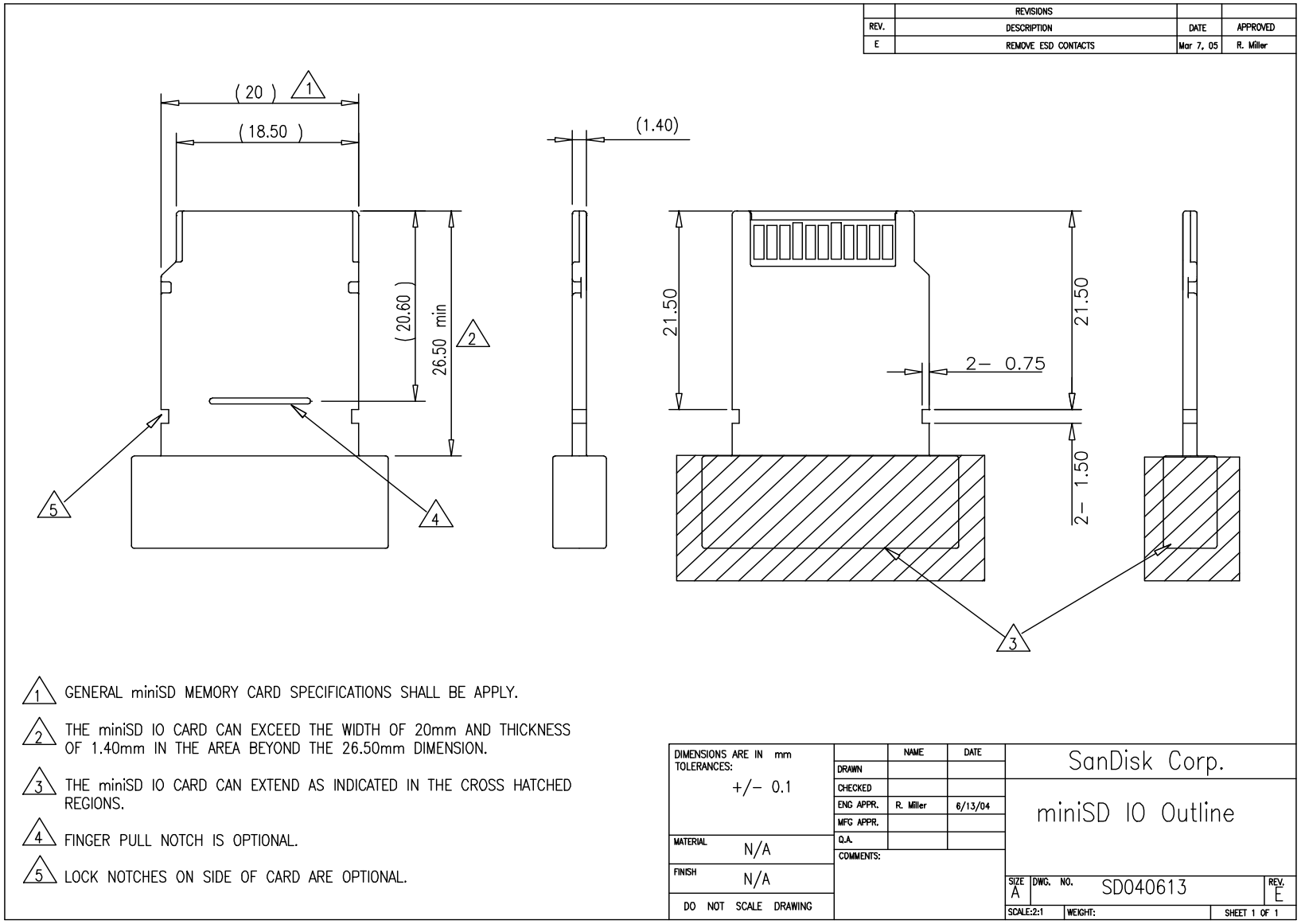


Figure 13-2 : miniSDIO Mechanical Extensions

13.4 microSDIO

The microSDIO card shall be compatible with host sockets designed for microSD memory cards. All microSDIO cards shall meet the mechanical specifications described in *microSD Card Addendum.* Mechanical Extensions are not supported. microSDIO shall be the same as microSD memory card form factor.

14. SDIO Power

14.1 SDIO Card Initialization Voltages

SDIO Specification Version 2.00 eliminated the voltage range 2.0-2.7V for basic communication because the Physical Layer Specification Version 2.00 eliminates it. SDIO cards follow the same voltage and current requirements as SD memory cards. This means that an SDIO or combo card shall be used at an operating voltage range of 2.7 to 3.6V. Hosts shall not supply the voltage range 2.0-2.7V for basic communication to the SDIO Version 2.00 or later cards.

14.2 SDIO Power Consumption

The SDIO cards are intended to operate in mobile devices that have limited power sources available. Because the host's battery life may be significantly reduced if the SDIO card draws excessive power, a primary goal of SDIO designers should be low power. By reducing power consumption to a minimum, Host battery life and consumer satisfaction will be enhanced. The following power data represents the maximum that a SDIO card may draw. It is important for designers to note that a low power host may reject any SDIO card that identifies itself as drawing more power that the host is willing to supply, thus lower power cards may have a competitive advantage in the market.

14.3 SDIO Current

Upon initial insertion, SDIO and combo cards shall draw a maximum of 15mA, averaged over a 1 second interval.

Once the card receives the CMD5 or ACMD41 initialization commands, the average current shall be:

- 50 mA or less, averaged over a 400 µS period for SDIO only cards
- 100 mA or less, averaged over a 1 second period for combo cards

This current limit shall continue until IOEx bits are set to 1.

The maximum current of SDIO card is less than 200 mA at any time, except for inrush current as defined in Chapter 15, unless the Master Power Control is enabled (High-power mode).
Refer to Chapter 11 about the maximum current in the High-power mode.
The current of each function is read from the CIS area. In case of a multi functions card, the total current can be calculated by the sum of the current of enabled functions. In case of a Combo card memory current consumption should be added that is available by using CMD6.

14.4 SDIO Card Operational Voltages

All SDIO cards shall support an operational voltage range of 2.7-3.6V.

A SDIO card may support voltage ranges in addition to those mentioned. The operational range of the card shall be properly identified in the OCR during the card's R4 response to CMD5.

15. Inrush Current Limiting

When an SDIO card is plugged into a "hot" socket or when power is quickly applied to an unpowered SDIO card, the SDIO card has a certain amount of on-board capacitance between V_{dd} and V_{ss} that will need to charge. In addition, an internal regulator on the SDIO card may supply current to its output bypass capacitance and to the function as soon as power is applied or at some later time when that function is enabled. Consequently, if no measures are taken to prevent it, there could be a surge of current into the SDIO card, which might pull the V_{dd} on the host below its minimum operating level or exceed the current supply ability of the host, possibly causing damage. Inrush currents can also occur when a high-power function is switched into its high-power mode. This problem must be solved by both limiting the inrush current by the card and by providing sufficient capacitance in each host socket to prevent the power supply of the host from going out of tolerance. An additional motivation for limiting inrush current is to minimize contact arcing, thereby prolonging connector contact life.

It is important for SDIO card makers to note that the actual capacitance of commercially available bypass capacitors may have a tolerance of +80%/-20%. The maximum capacitance values should be chosen with the tolerance take into account to prevent exceeding the specification limits.

In order to prevent power supply droop or damage, all SDIO cards shall meet the following conditions:

1. The maximum initial load (i.e. upon power-up) that can be placed in an SDIO card is 10µF in parallel with 220Ω. The 10µF capacitance represents all bypass capacitor(s) directly connected across the V_{dd} lines in the card common area or function(s) plus any capacitive effects visible through any regulators in the card that are enabled at power-up. This is represented in Figure 15-1 by C1. The 220Ω resistance represents the maximum current (15 mA) drawn by the card during connect. This is shown as Rin.
2. If more bypass capacitance is required by the design of the SDIO card, then the card shall incorporate some form of V_{dd} surge current limiting, such that it matches the characteristics of the above load.
3. The hosts' V_{dd} power lines shall be able to withstand a hot insertion of the above loads without causing an upset of the hosts' operation. The host may achieve this tolerance by many methods. For example, a soft-start reduces the inrush current to the SDIO card on power-up. Additionally, a large capacitor on the V_{dd} lines (shown as Ch in Figure 15-1) can supply the needed inrush current. Standard bypass methods should be used to minimize inductance and resistance between the bypass capacitors and the connectors to reduce droop. The bypass capacitors themselves should have a low dissipation factor to allow decoupling at higher frequencies.

If any regulator(s) or power switches on the card are enabled after the card is initialized (i.e. a power supply for a function that is enabled by the host), then the following requirements shall be met by the SDIO card:

1. The maximum in-rush spike shall not exceed 300mA at any time and the total time period where the current is over 200mA shall not exceed 100uSec in any 1 mS period. It is the responsibility of the SDIO card to limit these secondary inrush spikes.
2. Maximum capacitive load changes shall not exceed 10uF when an internal regulator is enabled or disabled. This is shown as C2 in Figure 15-1.

When a function is enabled in the high-power mode, the card shall meet the following requirements:

1. The maximum in-rush spike shall not exceed 2 times the maximum current selected by the host (from either TPLFE_HP_MAX_PWR_3.3V or TPLFE_LP_MAX_PWR_3.3V) at any time and the total time period where the current is over this maximum current shall not exceed 100uSec in any 1 mS period. It is the responsibility of the SDIO card to limit these secondary inrush spikes.
2. Maximum capacitive load changes shall not exceed 10uF when an internal regulator is enabled or disabled. This is shown as C2 in Figure 15-1.

15.1 Current Limit Design Example

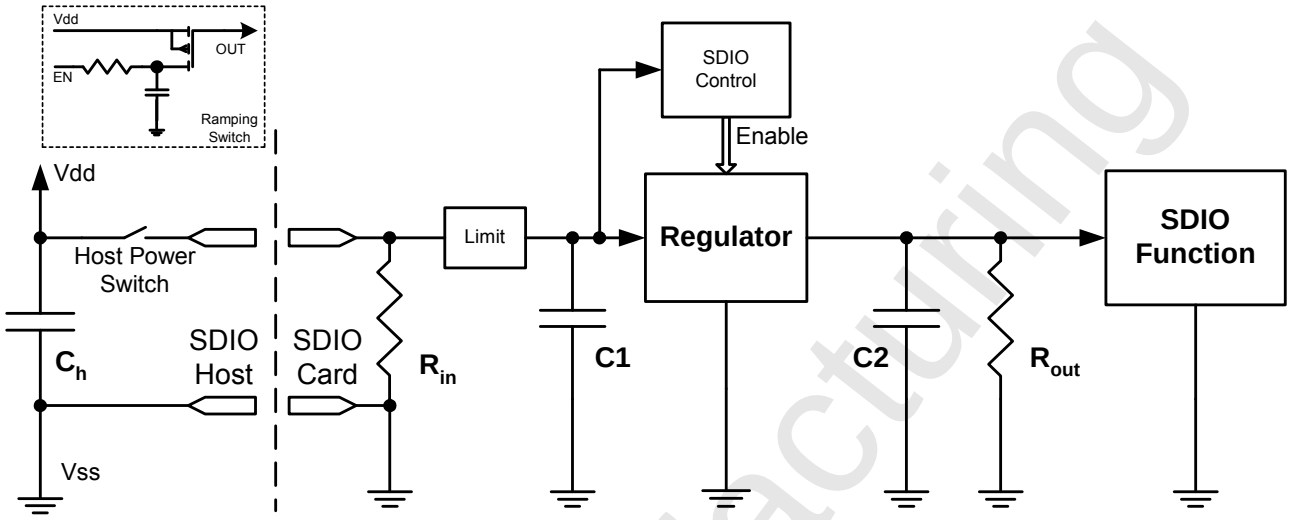


Figure 15-1 : SDIO Inrush Current

Figure 15-1 Shows the various components that contribute to inrush current. In this diagram, the host has an optional power switch so that the card is inserted into a socket without power applied (i.e. a cold socket). The initial load is defined by C1 and Rin where C1 is the sum of all capacitance from V_{dd} to ground on the input side of the regulator and Rin represents the DC load of the regulator and an the SDIO control circuits. The Host can control the timing of the application of power to the SDIO card by using the optional host power switch. If the host does not include a power switch, then the inrush current will begin to flow once the card is inserted and contact is made to the V_{dd} and V_{ss} pins. In this diagram, C1 has a maximum value of 10uF and the Rin equivalent resistance is 220Ω minimum. The host shall be designed so that the insertion of a SDIO card that meets the input capacitance and resistance values defined above shall not be damaged or have it's operation disrupted (i.e. a reset). On card insertion or enabling of the host power switch, charge will rapidly flow from the charge host capacitor (CH) to the discharged input capacitor on the card (C1). The charge will transfer and the voltage on the host V_{dd} line will drop in proportion to the ratio of C_H to C1. If C_H =C1, the resulting V_{dd} droop will be 50%. If the host must limit the V_{dd} droop to 10%, then C_H should be 10 times the value of C1 (100uF). Alternately, the host may limit the inrush current by using a semiconductor switch such as a FET for the host power switch. This is shown in Figure 15-1 as the ramping switch. By using a RC network to drive the FET gate, the V_{dd} applied to the card will be a ramp, rather then a near-instantaneous step. This reduced rise time will greatly reduce the inrush current, allowing a smaller C_H value to be used.

If the design of a SDIO card requires a input capacitance (C1) larger than the 10uF maximum value, then some sort of limit is required as shown as the Limit block in Figure 15-1. This limit function reduces the inrush current to a value equal to or less than the current inrush of a 10uF capacitor. This limit device could be a ramping switch similar to the one shown for the host.

Figure 15-1 also illustrates the use of a secondary voltage regulator (or switch) to power the SDIO function. Since the enabling of the power to the function may cause a secondary current surge to charge C2 and since the host may not have control of the timing of this secondary surge, it is entirely the responsibility of the SDIO card to limit the effect on the host V_{dd} due to these secondary surges. Again, a ramping type switch is a possible solution as is a "soft start" regulator. The actual design is left to the SDIO card maker.

16. CIS Formats

16.1 CIS Reference Document

The CIS used by SDIO is based directly upon the metaformat specification used by PCMCIA and Compact Flash. For details on the metaformat, refer to:

PC Card Standard
Volume 4
Metaformat Specification

Published by:
PCMCIA (Personal Computer Memory Card International Association)
2635 North First Street, Suite 218 per the PCMCIA .org website
San Jose, CA 95134 USA
+1-408-433-2273
+1-408-433-9558 (Fax)
www.pcmcia.org

16.2 Basic Tuple Format and Tuple Chain Structure

The Card Information Structure is one or more chains (or linked lists) of data blocks or tuples. The basic format of tuples is shown in Table 16-1.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE Tuple code: CISTPL_xxx							
01h	TPL_LINK Offset to next tuple in chain. This is the number of bytes in the tuple body. (n)							
02h.. (n+2)	The tuple body. (n bytes)							

Table 16-1 : Basic Tuple Format

Byte 0 of each tuple contains a tuple code. A tuple code of FFh is a special mark indicating that there are no more tuples in the chain. There are 2 tuples with only a tuple code, the CISTPL_NULL and the CISTPL_END (Refer to Table 16-2). These tuples do not have any additional bytes. For all other tuples, byte 1 of each tuple contains a link to the next tuple in the chain. If the link field is 0, then the tuple body is empty. If the link field contains FFh, then this tuple is the last tuple in its chain. There are two ways of marking the end of a tuple chain for SDIO cards: a tuple code of FFh, or a tuple link of FFh. The use of an FFh link value is allowed in SDIO cards, but it is recommended to use the End of Chain tuple. System software shall use the link field to validate tuples. No SDIO card tuple can be longer than 257 bytes: 1 byte *TPL_CODE* + 1 byte *TPL_LINK* + FFh byte tuple body (and this 257 byte tuple ends the chain). Some tuples provide a termination or stop byte that marks the end of the tuple. In this case, the tuple can effectively be shorter than the value implied by its link field. However, software shall not scan beyond the implied length of the tuple, even if a termination byte has not been seen.

16.3 Byte Order Within Tuples

Within tuples, all multi-byte numeric data shall be recorded in little-endian order. That is, the least-significant byte of a data item shall be stored in the first byte of a given field. Within tuples, all character data shall be stored in the natural order. That is, the first character of the field shall be stored in the first byte of the field. Fixed-length character fields shall be padded with null characters, if necessary.

16.4 Tuple Version

With the introduction of SDIO Specification Version 1.10, a different format for the tuple information is possible based on the changes made by each specification revision. These changes could be added fields in a tuple or entirely new tuples. In order to maintain backward compatibility, new data fields are added *after* existing fields in order to maximize backward compatibility. It is the responsibility of the host program that is scanning the tuple chain (sometimes called "walking the tuples") to first determine the SDIO specification version that the card was designed to. The program shall be designed to anticipate that SDIO cards will be encountered that are built to a later version of the specification. This requires the program to ignore the additional data existing in SDIO cards built to a later specification. Specifically, the program shall first read the SDIO specification version that the card was designed to meet from the SDIOx field in the CCCR area. If the version is less than or equal to the version of the tuple scan program, then the program shall know and properly decode all tuple information. If the card version is greater than the scan program's version, the scan program may need to ignore additional information fields or tuples. The unknown tuples can be ignored by simply skipping those tuples with unrecognized codes. Skipping is accomplished by using the TPL_LINK field (always the second byte) to jump over the unknown tuple. In a similar manner, the additional data fields in tuples should be ignored using the link field. For example, if a scan program is expecting 15h bytes of data and the TPL_LINK field indicates a size of 19h bytes, the scan program should ignore and skip over the last 4 bytes of data.

16.5 SDIO Card Metaformat

Unlike the PCMCIA card, the SDIO card has multiple CIS areas. There is a common CIS for the entire card and a CIS assigned to each function. Because of the multiple CIS areas, the SDIO card does not need to support the *CISTPL_LONGLINK_MFC* tuple or the *CISTPL_LINKTARGET* as described in Section 2.3.6 of the PCMCIA spec. Table 16-2 lists the tuple codes supported by SDIO cards. The type field indicates if a tuple is Optional (O), Mandatory (M), Recommended (R), or not applicable (n/a) for the common (function 0) tuple and for each function (1-7) supported by the card. For more details on each tuple, refer to the PCMCIA metaformat specification section referenced.

Code	Name	Description	PCMCIA Reference	Type Common	Type Function
00h	CISTPL_NULL	Null tuple	3.1.9	O	O
10h	CISTPL_CHECKSUM	Checksum control	3.1.1	R	R
15h	CISTPL_VERS_1	Level 1 version/product-information	3.2.10	O	O
16h	CISTPL_ALTSTR	The Alternate Language String Tuple	3.2.1	O	O
20h	CISTPL_MANFID	Manufacturer Identification String Tuple	3.2.9	M	O
21h	CISTPL_FUNCID	Function Identification Tuple	3.2.7	n/a	M
22h	CISTPL_FUNCE	Function Extensions	3.2.6	n/a	M
80h-8Fh		Vendor Unique Tuples	None	O	O
91h	CISTPL_SDIO_STD	Additional information for functions built to support application specifications for standard SDIO functions.	6.1.2	n/a	M[1]:Standard Function O[1]:Non-Standard Function
92h	CISTPL_SDIO_EXT	Reserved for future use with SDIO devices.	6.1.3	n/a	n/a
FFh	CISTPL_END	The End-of-chain Tuple	3.1.2	M	M

Table 16-2 : Tuples Supported by SDIO Cards

Note 1: the use of CISTPL_SDIO_STD is mandatory for all functions that claim to support a SDIO standard interface specification (Refer to Section 1.4). If the function does not support a standard SDIO interface, this tuple should be used with a value of 0.

16.6 CISTPL_MANFID: Manufacturer Identification String Tuple

The manufacturer identification tuple contains information about the manufacturer of a SDIO Card. Two types of information are provided: the SDIO Card's manufacturer and a manufacturer card number. This tuple shall be present in the card common CIS.

This should also be present in each function's CIS. This allows a function to override the card common manufacturer information so the driver can take advantage of unique features.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_MANFID (20h)							
01h	TPL_LINK Link to next tuple (at least 4)							
02h-03h	TPLMID_MANF SDIO Card manufacturer code							
04h-05h	TPLMID_CARD manufacturer information (Part Number and/or Revision)							

Table 16-3 : CISTPL_MANFID: Manufacturer Identification Tuple

The *TPLMID_MANF* field identifies the SDIO Card's manufacturer. New codes are assigned by both PCMCIA and JEIDA. The first 256 identifiers (0000h through 00FFh) are reserved for manufacturers who have JEDEC IDs assigned by JEDEC Publication 106. Manufacturers with JEDEC IDs may use their eight-bit JEDEC manufacturer code as the least significant eight bits of their SDIO Card manufacturer code. In this case, the most significant eight bits shall be zero (0). For example, if a JEDEC manufacturer code is 89h, their SDIO Card manufacturer code is 0089h. If a SDIO card manufacturer does not currently have a *TPLMID_MANF* assigned, one can be obtained at little or no cost from the PCMCIA. The TPLMID_CARD field is reserved for the use of the SDIO Card's manufacturer. It is anticipated that the field will be used to store card identifier and revision information.

16.7 SDIO Specific Extensions

SDIO cards use two to four tuples to provide additional information about the card (common) and each function. The first is the Function ID tuple. The changes for SDIO are detailed in the next sections.

16.7.1 CISTPL_FUNCID: Function Identification Tuple

To identify an SDIO card, the CISTPL_FUNCID tuple shall exist in all CIS areas. This means there shall be a CISTPL_FUNCID in the common CIS space chain and one in each function's CIS space chain. The format of this tuple is shown in

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_FUNCID (21h)							
01h	TPL_LINK Link to next tuple (02h)							
02h	TPLFID_FUNCTION Card function code (0Ch)							
03h	TPLFID_SYSINIT System initialization bit mask. (Not used, set to 00h)							

Table 16-4 : CISTPL_FUNCID Tuple

The function identification tuple contains information about the functionality provided by an SDIO Card. Information is also provided to enable system utilities to decide if the SDIO Card should be configured during system initialization. Since additional function specific information is available, one or more function extension tuples follow this tuple. The TPLFID_FUNCTION field contains an identifier assigned by PCMCIA (0Ch) to identify the SDIO device class.

16.7.2 CISTPL_FUNCE: Function Extension Tuple

The CISTPL_FUNCE tuple provides standard information about the card (common) and each individual function. There shall be one CISTPL_FUNCE in each function's CIS immediately following the CISTPL_FUNCID tuple. The format of the CISTPL_FUNCE is shown in Table 16-5.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_FUNCE (22h)							
01h	TPL_LINK Link to next tuple (Refer to following sections)							
02h	TPLFE_TYPE Type of extended data (Refer to following sections)							
03h-n	TPLFE_DATA Function information (Refer to following sections)							

Table 16-5 : CISTPL_FUNCE Tuple General Structure

There are two versions of the CISTPL_FUNCE tuple, one for the common CIS (function 0) and a version used by the individual function's CIS (1-7). Both types are described below.

16.7.3 CISTPL_FUNCE Tuple for Function 0 (Extended Data 00h)

The CISTPL_FUNCE tuple gives the host common information about the card. There shall be only one of these tuples, located in the CIS for function 0 following the CISTPL_FUNCID. The format of this tuple is shown in Table 16-6.

The optional fields are added in the SDIO Version 3.00 to indicate the list of the Card Maximum Power restricted by Card Case Temperature (Tc). If the card maximum power is less than 720mW (3.6V x 200mA), these optional field is not required and TPL_LINK Link to next tuple is set to 04h.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_FUNCE (22h)							
01h	TPL_LINK Link to next tuple (2n+4)							
02h	TPLFE_TYPE Type of Extended Data (00h)							
03h-04h	TPLFE_FN0_BLK_SIZE							
05h	TPLFE_MAX_TRAN_SPEED							
06h	TPLFE_TC_01							
07h	TPLFE_CP_01							
08h	TPLFE_TC_02							
09h	TPLFE_CP_02							
.....	.....							
2n+4	TPLFE_TC_n							
2n+5	TPLFE_CP_n							

Table 16-6 : TPLFID_FUNCTION Tuple for Function 0 (Common)

The fields in this tuple have the following definition:

Field	Description
TPLFE_FN0_BLK_SIZE	This is both the maximum block size and byte count that function 0 can support. A value of zero is not valid and shall not be used.
TPLFE_MAX_TRAN_SPEED	This byte indicates the maximum transfer rate per one data line during data transfer. This value applies to all functions in the SDIO card. This value shall be 25 Mb/Sec (32h) for all Full-Speed SDIO cards. The minimum value for Low-Speed SDIO cards shall be 400 Kb/Sec (48h). The format is identical to the TRAN_SPEED value stored in the CSD of SD memory cards. The maximum data transfer rate is coded according to the following method: **Bits 2:0 contain the transfer rate unit coded as follows:** 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4... 7=reserved

	Bits 6:3 contain the time value codes as follows: 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0 **Bit 7 is reserved and shall be zero** A Combo Card shall support 25MHz clock (32h) SDR104 supported SDIO Card sets TRAN_SPEED to 2Bh (up to 200Mbit/sec) SDR50 and DDR50 supported SDIO Cards set TRAN_SPEED to 0Bh (up to 100Mbit/sec),
TPLFE_TC_n	This 8-bit value indicates the Card Case Temperature (Tc) required for the card heat removal and the range is from 1 to 255 deg.C in 1 deg.C step. The number "n" indicates the number of Tc and the maximum power pairs in this tuple. As the power is inverse proportion to Tc, The larger "n" field shall indicate the smaller temperature value. If a value is set to 0, the host should ignore and skip it. At least 80 deg.C is required in this tuple.
TPLFE_CP_n	This 8-bit value indicates the maximum card power that is restricted by the correspondent Tc and the range is from 10 to 2550 mW in 10mW step. The number "n" indicates the number of Tc and the maximum power pairs in this tuple. The larger "n" field shall indicate the larger power value. If a value is set to 0, the host should ignore and skip it.

Table 16-7 : TPLFID_FUNCTION Field Descriptions for Function 0 (common)

16.7.4 CISTPL_FUNCE Tuple for Function 1-7 (Extended Data 01h)

The CISTPL_FUNCE tuple gives the host common information about each individual function on a per-function basis. There shall be one of these tuples, located in the CIS for each function following the CISTPL_FUNCID.

Extended Data 01h is defined for Power Control using EMPC, SMPC in CCCR and SPS, EPS in FBR instead of by the power state control (PS[3:0]=0h).

The format of this tuple is shown in Table 16-8.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_FUNCE (22h)							
01h	TPL_LINK Link to next tuple (2Ah)							
02h	TPLFE_TYPE Type of Extended Data (01h)							
03h	TPLFE_FUNCTION_INFO							
04h	TPLFE_STD_IO_REV							
05h-08h	TPLFE_CARD_PSN							
09h-0Ch	TPLFE_CSA_SIZE							
0Dh	TPLFE_CSA_PROPERTY							
0Eh-0Fh	TPLFE_MAX_BLK_SIZE							
10h-13h	TPLFE_OCR							
14h	TPLFE_OP_MIN_PWR							
15h	TPLFE_OP_AVG_PWR							
16h	TPLFE_OP_MAX_PWR							
17h	TPLFE_SB_MIN_PWR							
18h	TPLFE_SB_AVG_PWR							
19h	TPLFE_SB_MAX_PWR							
1Ah-1Bh	TPLFE_MIN_BW							
1Ch-1Dh	TPLFE_OPT_BW							

1Eh-1Fh	TPLFE_ENABLE_TIMEOUT_VAL
20h-21h	TPLFE_SP_AVG_PWR_3.3V
22h-23h	TPLFE_SP_MAX_PWR_3.3V
24h-25h	TPLFE_HP_AVG_PWR_3.3V
26h-27h	TPLFE_HP_MAX_PWR_3.3V
28h-29h	TPLFE_LP_AVG_PWR_3.3V
2Ah-2Bh	TPLFE_LP_MAX_PWR_3.3V

Table 16-8 : TPLFID_FUNCTION Tuple for Function 1-7 (High Power Tuple)

The fields in this tuple have the following definition:

Field	Description
TPLFE_FUNCTION_INFO	Bit significant information about the Function The bits are defined in Table 16-10
TPLFE_STD_IO_REV	This 8-bit value contains the version level of the Application Specification for Standard SDIO Functions that this function supports. The format is x.y where x is the major version (4-bits) and y is the minor version level. For example if the version is 2.4 the value would be 24h. If this function does not support an SDIO standard function, the value shall be 00h.
TPLFE_CARD_PSN	The Product Serial Number is a 32 bit unsigned binary integer. Support of a serial number is optional, if there is no serial number, this field shall be 00000000h While a unique serial number is not required for all devices, it is strongly recommended that SDIO card vendors place a unique serial number in this field to assist the operating systems is differentiating multiple cards of the same type. Also note that some Application Specifications for Standard SDIO Functions require a unique serial number in this field. The individual Application Specification will indicate if support of this field is mandatory.
TPLFE_CSA_SIZE	Size of the CSA space available for this function in bytes
TPLFE_CSA_PROPERTY	This byte contains flags identifying properties of this function's CSA. The bits are defined in Table 16-11.
TPLFE_MAX_BLK_SIZE	This is both the maximum block size and byte count that this function can support. A value of zero is not valid and shall not be used.
TPLFE_OCR	This is the OCR value for this function. The format is identical to the 32-bit OCR format used by SD memory devices. For more details, refer to Section 5.1 of the Physical Layer Specification.
TPLFE_OP_MIN_PWR	This is the minimum current at 2.7V operating voltage, in mA required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero.
TPLFE_OP_AVG_PWR	This is the average current at 3.3V operating voltage, in mA, required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero
TPLFE_OP_MAX_PWR	This is the maximum (peak) current at 3.6V operating voltage, in mA, required by this function when operating. This value is valid for all voltages supported by this function. If the required current exceeds 200mA, this value shall be zero

TPLFE_SB_MIN_PWR	This is the minimum current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 00h. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (FFh) is used to indicate a value of 255mA or greater.
TPLFE_SB_AVG_PWR	This is the average current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 00h. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO devices cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (FFh) is used to indicate a value of 255mA or greater.
TPLFE_SB_MAX_PWR	This is the maximum current, in mA, required by this function when in the standby condition. If this function does not support standby, this value shall be 00h. The method to place this function in the standby state and the capabilities it has while in standby are vendor defined. This value is valid for all voltages supported by this function. Note that this value is valid only for standard power SDIO devices cards or high-power cards when EMPC is 0. With an 8-bit field, the range is from 0 to 254 mA. A value of 255 (FFh) is used to indicate a value of 255mA or greater.
TPLFE_MIN_BW	This is the minimum data transfer bandwidth, in KB/sec, needed by this function to successfully operate. If this function has no minimum necessary bandwidth, these bytes shall be 0000h.
TPLFE_OPT_BW	This is the data transfer bandwidth, in KB/sec, needed by this function to function at an optimum level. If this function has no optimum bandwidth, these bytes shall be 0000h.
TPLFE_ENABLE TIMEOUT_VAL	(*Added in SDIO Rev 1.1*) This 16-bit value indicates the function's required time-out value for coming ready after being enabled. This per-function value indicates the time a host should wait from asserting IOEx until expecting the card to indicate ready by asserting IORx. Different SDIO functions take different amounts of time to become ready after being enabled due to different internal initialization requirements. The required time-out limit is in 10mS steps, allowing a range of 0-655.35 seconds. If the cards required no time-out, this field shall be set to 0000h.
TPLFE_SP_AVG_PWR_3.3V	This value is the same as that of TPLFE_OP_AVG_PWR.
TPLFE_SP_MAX_PWR_3.3V	This value is the same as that of TPLFE_OP_MAX_PWR.
TPLFE_HP_AVG_PWR_3.3V	This 16-bit value indicates the average current, in mA, required by this function when operating in the Higher Current Mode (EMPC=1 & EPS=0). This value indicates a current range from 1 to 65535 mA.
TPLFE_HP_MAX_PWR_3.3V	This 16-bit value indicates the peak current, in mA, required by this function when operating in the Higher Current Mode. This value indicates a current range from 1 to 65535 mA.
TPLFE_LP_AVG_PWR_3.3V	This 16-bit value indicates the average current, in mA, required by this function when operating in the Lower Current Mode (EMPC=1 & EPS=1). This value indicates a current range from 1 to 65535 mA

TPLFE_LP_MAX_PWR_3.3V	This 16-bit value indicated the maximum (peak) current, in mA, required by this function when operating in the Lower current mode (EMPC=1 & EPS=1). This value indicates a current range from 1 to 65535 mA

Table 16-9 : TPLFID_FUNCTION Field Descriptions for Functions 1-7

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	RFU	FN_WUS

Table 16-10 : TPLFE_FUNCTION_INFO Definition

The FN_WUS (Wake Up Support) bit signals the function's support of wake-up when the function is placed into a low power state and the SDCLK is stopped. The intent is to allow a SDIO card to be placed into a low power condition and still signal the host of a wake-up event. If this bit is set to 1, then this function may be placed into a low power state using some function standard or vendor defined method and the SDCLK can be stopped with the power remaining applied to the card. Irrespective of the state of this bit, the host may stop the SDCLK as allowed by the Physical Layer Specification.

Bit	7	6	5	4	3	2	1	0
Name	RFU	RFU	RFU	RFU	RFU	RFU	CSA_NF	CSA_WP

Table 16-11 : TPLFE_CSA_PROPERTY Definition

The CSA_WP bit is used to write protect the CSA for this function, or identifies it as read-only. If this bit is set to 1, then the CSA of this function is either write protected or read-only. If this bit is clear, then the host may write the CSA. Setting of this bit for R/W devices is handled in a vendor-defined method.

The CSA_NF bit is used to indicate to the host that the CSA area should not be reformatted. If the bit is set to 1, the host may not format the CSA file structure. If this bit is cleared to 0, then the host may reformat the file system of the CSA.

16.7.5 CISTPL_FUNCE Tuple for Function 1-7 (Extended Data 02h)

The CISTPL_FUNCE tuple gives the host common information about each individual function on a per-function basis. Existence of this tuple indicates that the card supports the Power State.
Extended Data 02h consists of a list of power consumption up to 15 power states. The Power State is enabled when EMPC is set to 1 and PS[3:0] in FBR is set to a selected Power State Number and then EPS in FBR is not effective. Refer to Section 11.2.4 for more detail.

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_FUNCE (22h)							
01h	TPL_LINK Link to next tuple (2n+2)							
02h	TPLFE_TYPE Type of Extended Data (02h)							
03h	00h (Fixed value)							
04h, 05h	TPLFE_PS_01							
06h, 07h	TPLFE_PS_02							
08h, 09h	TPLFE_PS_03							
.....	.....							
2n+2, 2n+3	TPLFE_PS_n							

Table 16-12 : TPLFID_FUNCTION Tuple for Function 1-7 (Power State Tuple)

Field	Description
TPLFE_PS_n	This 16-bit value indicates the maximum power consumption per function in mW and the range is from 1 to 65535 mW in 1mW step. The number "n" indicates the power state number. Up to 15 power states can be defined. The larger "n" field shall indicate the larger power value. If a value is set to 0, the host should ignore and skip it. The host needs to set a power state number to PS[3:0] in FBR and the card shall operate below the power specified in this tuple.

Table 16-13 : TPLFID_FUNCTION Field Descriptions for Functions 1-7 (Power State Tuple)

16.7.6 CISTPL_SDIO_STD: Function is a Standard SDIO Function

This tuple code (91h) has been reserved for use by SDIO devices that conform to the application specifications for standard SDIO functions, as defined in those separate specifications. The exact format for this tuple can be found in those specifications. The basic format only is provided here and the reader is directed to the appropriate Application Specification for Standard SDIO Functions for more complete details:

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_SDIO_STD (91h)							
01h	TPL_LINK Link to next tuple ($2 \leq n \leq 255$) n is the number of bytes in the tuple body							
02h	TPLSDIO_STD_ID							
03h	TPLSDIO_STD_TYPE							
04h ... (n+1)	TPLSDIO_STD_DATA If n = 2, TPLSDIO_STD_DATA does not exist.							

Table 16-14 : CISTPL_SDIO_STD: Tuple Reserved for SDIO Cards

The fields in this tuple have the following definition:

Field	Description
TPLSDIO_STD_ID	This 8-bit code identifies the Standard SDIO Function type for which this tuple supplies additional information. The available codes can be found in the I/O device interface code entry of the FBR (Refer to Section 6.10)
TPLSDIO_STD_TYPE	This 8-bit value identifies the format and type of data contained within the body of this tuple. If this value is 00h, then only 1 standard data structure has been defined for this Standard SDIO Function. If this value is non-zero, then this byte identifies which tuple data format is being used for the data. This code is defined in the Application Specification for Standard SDIO Functions.
TPLSDIO_STD_DATA	The data component of this tuple (n-2 bytes). The format of this data structure is defined in the Application Specification for Standard SDIO Functions.

Table 16-15 : TPL_CODE CISTPL_SDIO_STD Definition

16.7.7 CISTPL_SDIO_EXT: Tuple Reserved for SDIO Cards

This tuple code (92h) has been reserved for future use by SDIO cards. The actual format of the data has not been established at this time. The basic format of this tuple is:

Byte	7	6	5	4	3	2	1	0
00h	TPL_CODE CISTPL_SDIO_EXT (92h)							
01h	TPL_LINK Link to next tuple ($0 \leq n \leq 255$) n is the number of bytes in the tuple body							
02h ... (n+1)	TPLSDIO_EXT_DATA The data component of this tuple (n bytes) If n=0, TPLSDIO_EXT_DATA does not exist.							

Table 16-16 : CISTPL_SDIO_EXT: Tuple Reserved for SDIO Cards

17. Embedded SDIO

17.1 Relaxing SDIO Requirements for Embedded Device

17.1.1 Form Factor

The requirements regarding form factor are not applicable to an Embedded Device.

17.1.2 CIS Information

Supporting of CIS is optional for an Embedded Device. If the host system has another method to recognize an embedded device, the CIS can be omitted.

17.1.3 8-Bit Bus Mode

8-bit mode is defined for only embedded usage. Achieving higher performance in the lower frequency is the objective of this mode. In the case of 8-bit mode, skew adjustment and cross talk need to be considered. The SD bus clock frequency is then limited to 50MHz and the UHS-I mode tuning by CMD19 is out of scope in 8 bit mode.
(The details of 8-bit mode are not defined in this document but it will be defined in a future specification.)

17.2 Interface Signal for Embedded SDIO Device

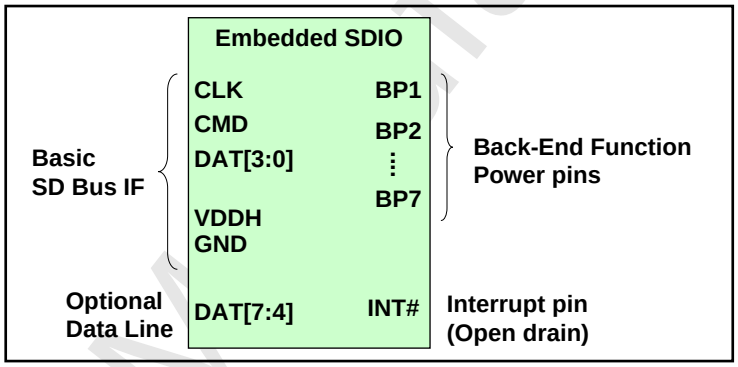


Figure 17-1 : Embedded SDIO Interface

Figure 17-1 shows the recommended interface for the Embedded SDIO Device. This interface is suitable to configure a Shared Bus which is defined by the Part A2 Host Controller Specification Version 3.00.

An Embedded Device can provide the back-end function power pins with any voltage. Therefore, host interface power and backend function power can be separated. The Sleep State of SDIO is defined by the IOEx=0 and back-end power can be off during the Sleep State.
The Embedded Device should provide Interrupt pins (low active) to be able to use the device in a Shared Bus configuration.

17.3 Shared Bus Configuration

Figure 17-2 shows an example configuration of a system supporting a card slot and shared bus. A card slot bus and shared bus are separated so that the removable card does not influence the devices on the shared bus. The SD Bus signals except the clock signal are connected together on the shared bus. Each device is selected by individual clock pins. An example timing of clock signals is shown in Figure 17-3. Stopping the clock for an unselected device enables the system to reduce power consumption of the devices. Even if the SD bus clock is stopped, an SDIO device can generate an interrupt by the INT# pin to request a service to Host System. INT is an asynchronous interrupt, low active, open drain and then can be wired-or with the interrupt pin of another device. A Pull-up resister is required for the INT# signal.

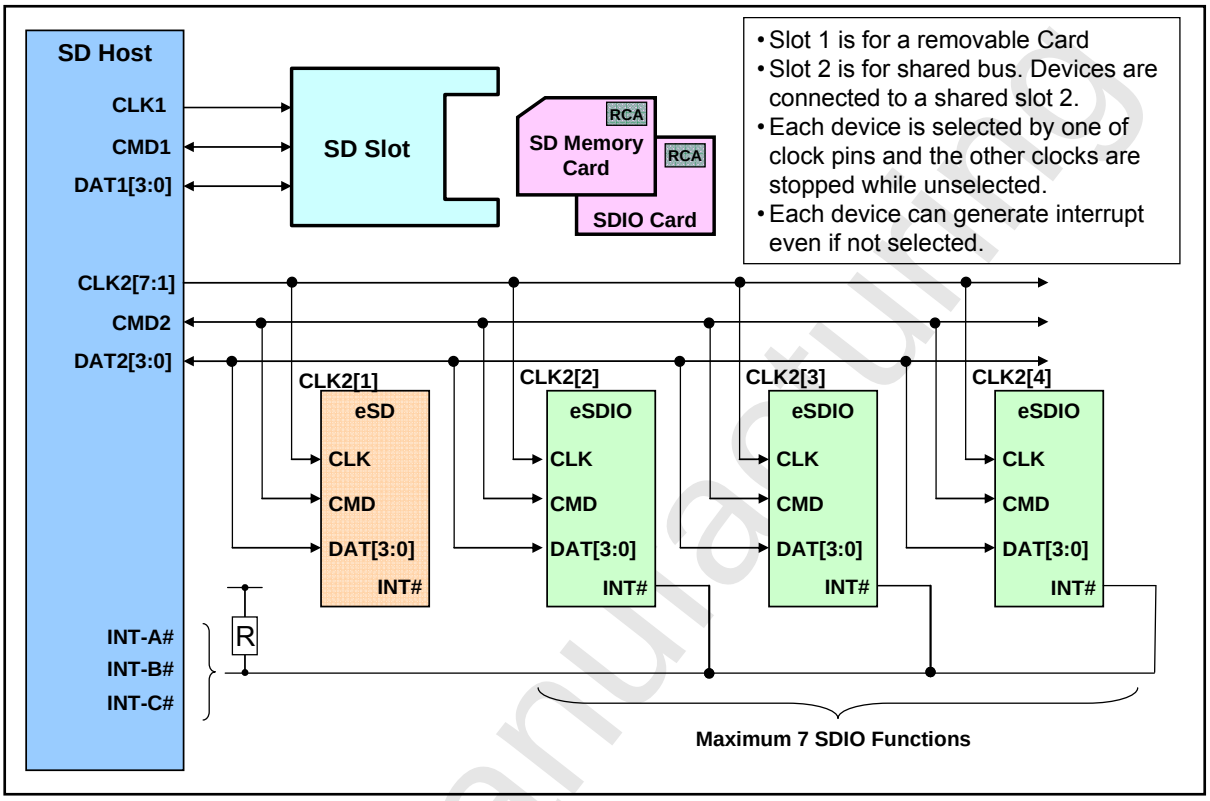


Figure 17-2 : Example Configuration Supporting Card Slot and Shared Bus

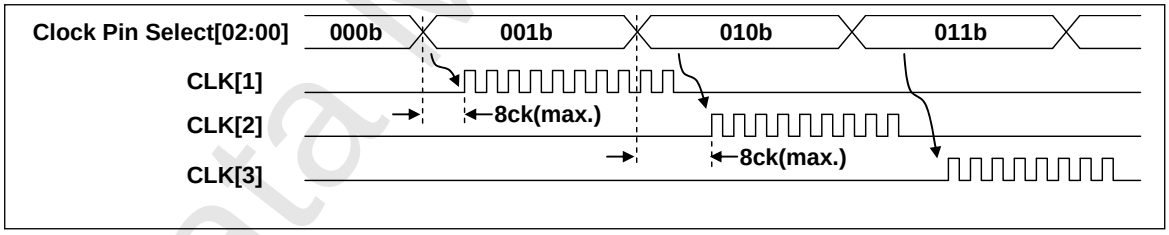


Figure 17-3 : Device Select Using Clock Signal

A clock signal is assigned per device
One of the clocks is oscillated without overlap
A Clock shall be generated within 8 clocks from when the Clock Pin Select is changed.
A device can be switched while the selected device does not drive the SD Bus; e.g., while busy is not indicated. An Un-selected device can request the host to service with an interrupt.

Appendix A (Normative) : Reference

A.1 Reference

The following documents are referenced by this specification.
This specification can apply to any released versions of below SD Specifications after Version 2.00.

1) SD Specifications Part 1 Physical Layer Specification Version 3.01 February 18, 2010
2) SD Specifications Part 1 Standard Size SD Card Mechanical Addendum Ver3.00 February 18, 2010
3) SD Specifications Part 1 miniSD Card Addendum Version 2.01 March 27, 2008
4) SD Specifications Part 1 microSD Card Addendum Version 3.00 February 18, 2010
5) SD Specifications Part 1 eS Addendum Version 2.10 November 25, 2008
6) SD Specifications Part 2 File System Specification Version 3.00 April 16, 2009
7) ISO/IEC9293:1994
   Information technology - Volume and file structure of disk cartridges for information interchange
8) PC CARD STANDARD Release 8.1 2002
   Volume 4 Metaformat Specification

Appendix B (Normative) : Special Terms

B.1 Terminology

Embedded SDIO	I/O or combo devices that are embedded in a host device that utilize the SDIO electrical and command interface, but are not intended to be removed.
Flash	A type of multiple time programmable non volatile memory
Full-Size SDIO Card	A SDIO card with physical dimensions based on the Standard Size SD Card described in the Standard Size SD Card Mechanical Addendum.
Function	An I/O device contained within an SDIO card
High-Power SDIO	A SDIO device that requires up to 500 mA rather than the standard 200 mA allowed for a Standard-Power SDIO device
hi-Z	A three-state driver in the high-impedance state
Interrupt	A signal from the SDIO device to the host signaling the need for attention
Interrupt Period	The times that a card may generate an interrupt signal on the SD bus
Legacy Slot	SD Slot that supports only the SD Physical Specification Version 1.01
LOW, HIGH	Binary interface states with defined assignment to a voltage level
MBIO	Multi-Block I/O
microSDIO Card	A SDIO card based on the microSD card form factor.
miniSDIO Card	A SDIO card based on the miniSD card form factor.
SDIO aware	A host designed to support the signals and protocol of SDIO devices
Standard-Power SDIO	A SDIO device that operates with 200 mA or less of current at all times.
Stuff bit(s)	Filling bit(s) to ensure fixed length frames for commands and responses.
Suspend	Temporarily halting the transfer of data
Three-state driver	A driver stage which has three output driver states: HIGH, LOW and high impedance (which means that the interface does not have any influence on the interface level)

B.2 Abbreviations

3C	The original 3 companies that established the SDA and the SD card specification
ACMD	Application CommanD An additional set of commands created to support SD specific requirements.
Block	A number of bytes, basic data transfer unit
CCCR	Common Card Control Registers
CD	Connect/Disconnect
CIA	Common Information Area

CID	Card IDentification number register
CIS	Card Information Structure
CLK	Clock signal
CMD	Command line or SD bus command (if extended CMDXX)
Combo Card	A card that includes both SDIO and SD memory
CRC	Cyclic Redundancy Check
CS	Chip or Card Select
CSA	Code Storage Area
CSD	Card Specific Data register
DAT[x]	Data line where x is in the range of 0 to 3
DSR	Driver Stage Register
ECC	Error Correction Code
EMI	Electro-Magnetic Interference
ESC	External Signal Contacts
ESD	Electro-Static Discharge
FAT	File Allocation Table
FBR	Function Basic Registers
FIFO	First In-First Out buffer
MSB, LSB	The Most Significant Bit or Least Significant Bit
OCR	Operation Conditions Register
PCMCIA	Personal Computer Memory Card International Association
PnP	Plug and Play a means to identify an SDIO device and optionally load applications and/or drivers without user intervention
R/O	Read Only
R/W	Read or Write
RAW	Read After Write
RCA	Relative Card Address register
Resume	Re-starting the temporarily halted data transfer
RFU	Reserved for Future Use. Normally Read-Only and set to 0
ROM	Read Only Memory
RWC	Read Wait Control
SCR	SD Configuration Register
SDA	SD Association
SDCLK	SD clock signal
SDIO	SD Input/Output
SPI	Serial Peripheral Interface
TBD	To Be Determined (in the future)
Tuple	Data blocks in a linked list or chain format
UHS	Ultra High Speed
VDD	+ Power supply

VSS	Power supply ground
W/O	Write Only
WP	Write Protect

Appendix C (Normative) : Command List

C.1 SD and SPI Command List

Table C- 1 and Table C- 2 show the commands that are supported by SD memory and SDIO devices in both SPI and SD modes. If a command is not identified as either mandatory or optional, then it is not supported by that device.

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD2	ALL_SEND_CID	Mandatory		CID not supported by SDIO
CMD3	SEND_RELATIVE_ADDR	Mandatory	Mandatory	
CMD4	SET_DSR	Optional		DSR not supported by SDIO
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory[1]	Mandatory[1]	Added in Part 1 v1.10
CMD7	SELECT/DESELECT_CARD	Mandatory	Mandatory	
CMD8	SEND_IF_COND	Optional	Optional	SDHC or SDXC
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD11	VOLTAGE_SWITCH	Optional	Optional	Mandatory if UHS-I supported
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information
CMD15	GO_INACTIVE_STATE	Mandatory	Mandatory	
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD19	SEND_TUNING_BLOCK	Optional	Optional	Mandatory if UHS-I supported
CMD23	SET_BLOCK_COUNT	Optional		Mandatory if SDR104 supported
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
ACMD6	SET_BUS_WIDTH	Mandatory		
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR not supported by SDIO

Table C- 1 : SD Mode Command List

[1] For Part 1 v1.10 or higher Memory or Combo Cards

C.2 SD and SPI Command List

Supported Commands	Abbreviation	SDMEM System	SDIO System	Comments
CMD0	GO_IDLE_STATE	Mandatory	Mandatory	Used to change from SD to SPI mode
CMD1	SEND_OP_COND	Mandatory		
CMD5	IO_SEND_OP_COND		Mandatory	
CMD6	SWITCH_FUNC	Mandatory[1]	Mandatory[1]	Added in Part 1 v1.10
CMD9	SEND_CSD	Mandatory		CSD not supported by SDIO
CMD10	SEND_CID	Mandatory		CID not supported by SDIO
CMD12	STOP_TRANSMISSION	Mandatory		
CMD13	SEND_STATUS	Mandatory		Card Status includes only SDMEM information.
CMD16	SET_BLOCKLEN	Mandatory		
CMD17	READ_SINGLE_BLOCK	Mandatory		.
CMD18	READ_MULTIPLE_BLOCK	Mandatory		
CMD24	WRITE_BLOCK	Mandatory		
CMD25	WRITE_MULTIPLE_BLOCK	Mandatory		
CMD27	PROGRAM_CSD	Mandatory		CSD not supported by SDIO.
CMD28	SET_WRITE_PROT	Optional		
CMD29	CLR_WRITE_PROT	Optional		
CMD30	SEND_WRITE_PROT	Optional		
CMD32	ERASE_WR_BLK_START	Mandatory		
CMD33	ERASE_WR_BLK_END	Mandatory		
CMD38	ERASE	Mandatory		
CMD42	LOCK_UNLOCK	Optional		
CMD52	IO_RW_DIRECT		Mandatory	
CMD53	IO_RW_EXTENDED		Mandatory	Block mode is optional
CMD55	APP_CMD	Mandatory		
CMD56	GEN_CMD	Mandatory		
CMD58	READ_OCR	Mandatory		
CMD59	CRC_ON_OFF	Mandatory	Mandatory	
ACMD13	SD_STATUS	Mandatory		
ACMD22	SEND_NUM_WR_BLOCKS	Mandatory		
ACMD23	SET_WR_BLK_ERASE_COUNT	Mandatory		
ACMD41	SD_APP_OP_COND	Mandatory		
ACMD42	SET_CLR_CARD_DETECT	Mandatory		
ACMD51	SEND_SCR	Mandatory		SCR includes only SD-MEM information.

Table C- 2 : SPI Mode Command List

[1] For Part 1 v1.10 or higher Memory or Combo Cards

Appendix D (Normative) : Reference

D.1 Example SDIO Controller Design

Figure D- 1 shows an example of an SDIO controller design. In this example, two independent state machines are used. The first is the Bus State machine, which communicates with the host and maintains bus states. This is described in Figure D- 1. The second state machine is used to communicate and control the function(s) in the card. This state machine maintains control of function states such as Exec, Ready and interrupts. An example state table for this machine can be seen in Figure D- 2.

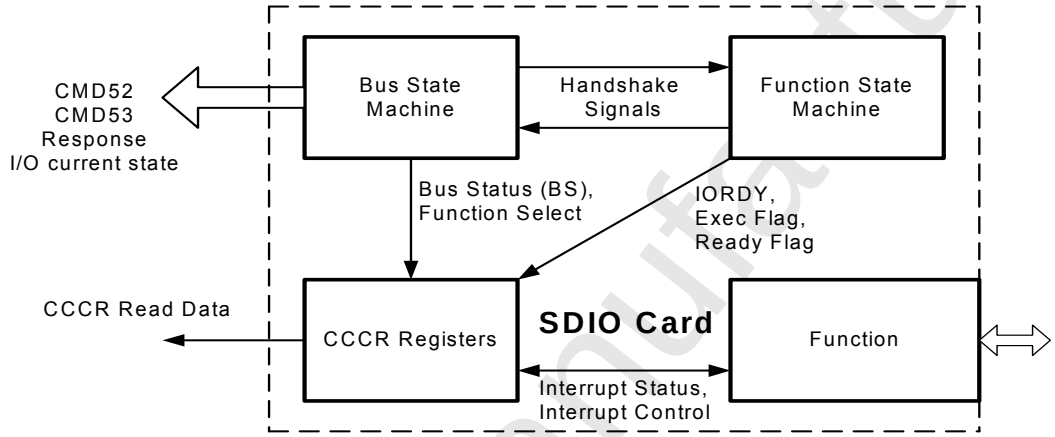


Figure D- 1 : SDIO Internal State Machine Example

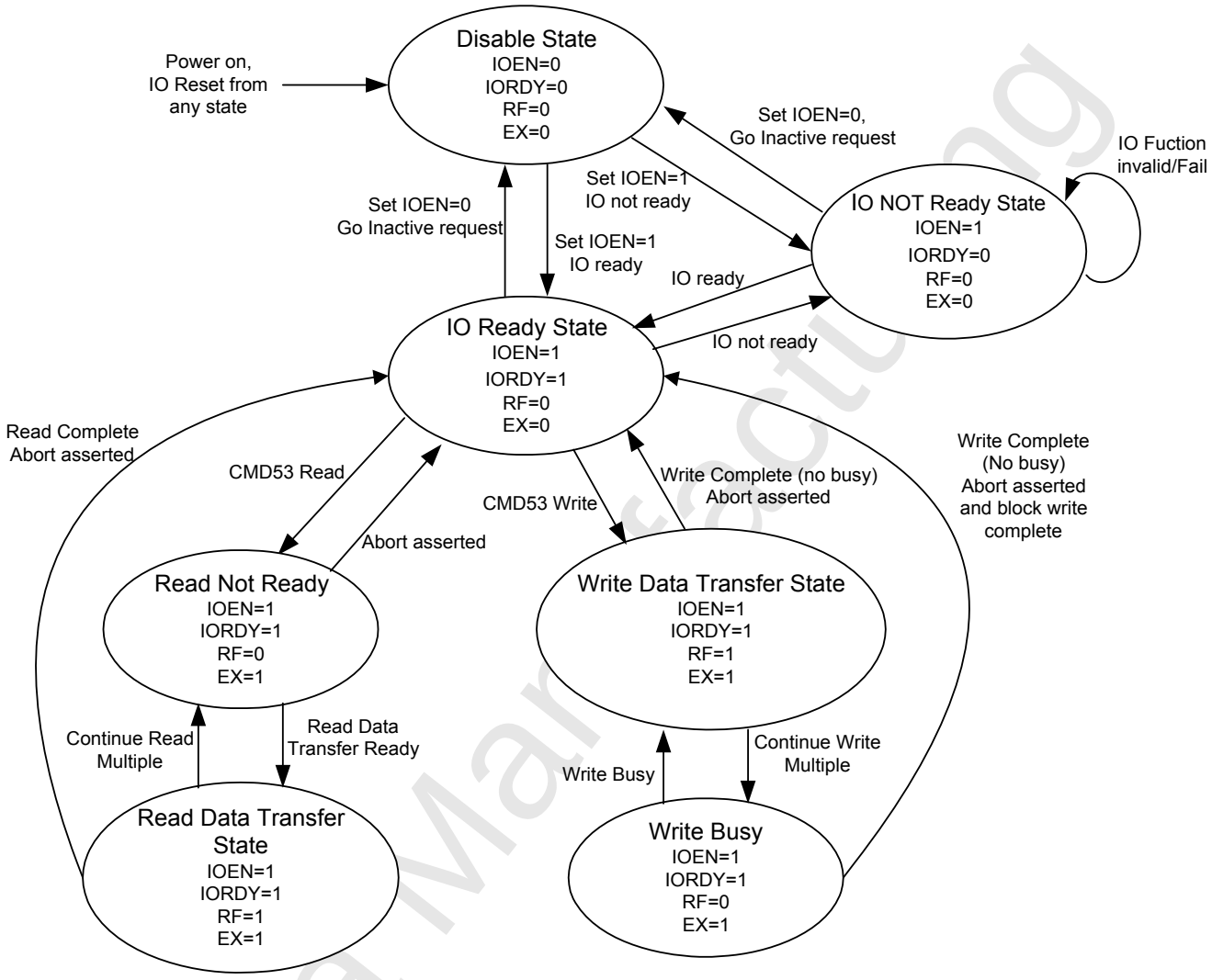


Figure D- 2 : State Diagram for Function State Machine