Secure Digital Host Controller (SDHC) Block Guide Revision 1.8

Original Release Date: 30 Jan 2002
Revised: 29 Nov 2004

TSPG 8/16 bit MCU
Freescale Semiconductor, Inc.

Revision History

Version Number	Revision Date	Effective Date	Author	Description of Changes
1.8	29 Nov 2004	29 Nov 2004	Wai-On Law	Changed company logo.
1.7	5 Dec 2003	5 Dec 2003	K.W.Lam	1. Removed POPI messages.
1.6	9 June 2003	9 June 2003	K.W.Lam	1. Added BLK_RDY bit to SDSTAT register in Page 20, Figure 3-2 2. Replaced RDTOUT with SDRDTOUT for RDTO bit in Page 21. 3. Replaced RFMT with RSPNO in Section 5.9 in Page 38. 4. Changed ‘Start and Stop SD/MMC Card Clock’ to ‘SD/MMC Card Clock Control’ in Section 5.2 in Page 36.
1.5	29 May 2003	29 May 2003	K.W.Lam	1. version 1.5 is drafted.
1.2	17 Apr 2002	17 Apr 2002	K.W.Lam	1. Version 1.2 is drafted.
1.1	15 Mar 2002	15 Mar 2002	K.W.Lam	1. Version 1.1 is drafted.
1.0	30 Jan 2002	30 Jan 2002	K.W.Lam	1. Version 1.0 is drafted. 2. PSCLK is added to SDCLK register.
0.3	19 Dec 2001	19 Dec 2001	K.W.Lam	1. Version 0.3 is drafted.
0.2	10 Dec 2001	10 Dec 2001	K.W.Lam	Version 0.2 is drafted after peer review on 7 Dec 2001.
0.1	12 Nov 2001	12 Nov 2001	K.W.Lam	1. Initial version 0.1 is drafted.

Table of Contents

Section 1 Introduction

Section 2 External Signal Description

Section 3 Memory Map/Register Definition

Section 4 Functional Description

Section 5 Initialization/Application Information

List of Figures

List of Tables

Section 1 Introduction

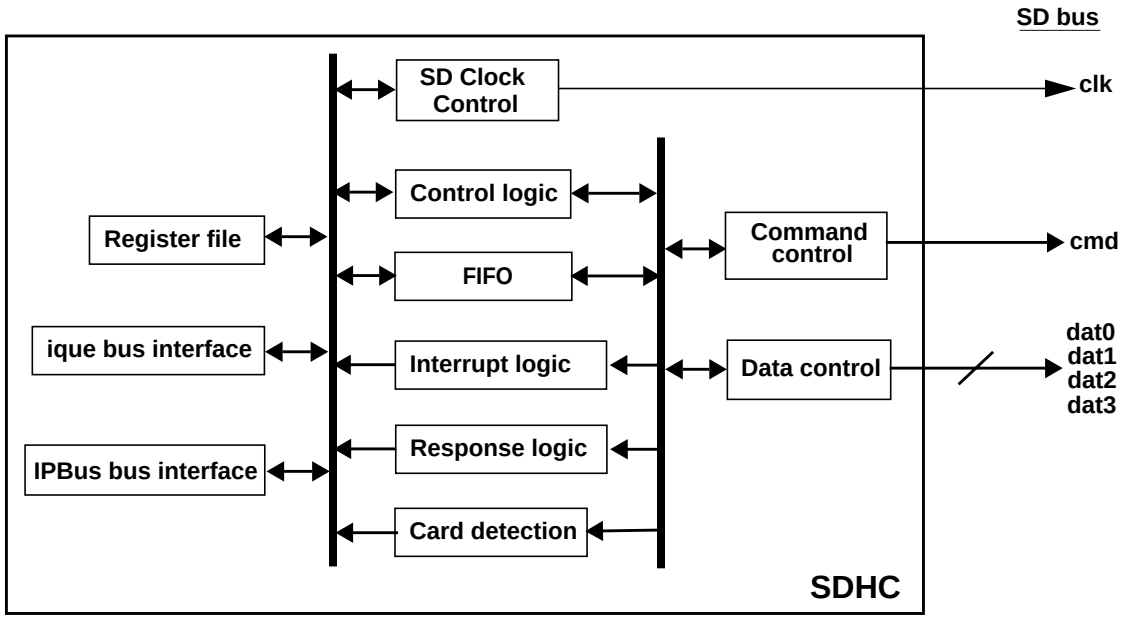


Figure 1-1 SDHC Block Diagram

1.1 Overview

The Secure Digital Host Controller (SDHC) is the link between the system and the Secure Digital (SD)/MultiMediaCard (MMC) card. It translates the protocol of the standard SD/MMC bus to the IPBus/IQUE interface bus, or vice versa and controls all activities on the SD/MMC bus.

The SD/MMC transfer protocol requires a CRC check to ensure data transfer quality in command, response and data transfer, so CRC code is automatically generated by hardware and added to the bit pattern when a command or data is transmitted. When a response or data is received, a CRC check is carried out automatically.

The SHDC incorporates a 16-byte FIFO to enhance the data transfer performance. In data transfer, data is written to the FIFO first when transmitting or receiving data. A set of commands will be programmed to the SD/MMC card before data transfer begins. When the command is issued, the SD/MMC card will return a response about the card information (depending on command type). The application then may check the response content. Data transfer may start between SDHC and the SD/MMC card. Moreover, the SD clock can be enabled or disabled and a range of frequencies can be selected.

1.1.1 Interface

In the SD (Secure Digital) Host Controller is a 6-pin serial interface, and it supports two communication protocol modes - SD mode and MMC mode. The maximum clock operating frequency for the SD/MMC card in both modes is 20MHz.

1.1.2 Card identification

In card identification mode, the host resets a SD/MMC card that are in card identification mode, validates operation voltage range, identifies and asks them to publish Relative Card Address (RCA). All data communication in the card identification mode uses the command line (cmd) only.

Every command on the SD/MMC bus may be prefixed with the 80 card clocks required for card initialization. This function is useful for initializing new card that are hot-inserted on the SD/MMC bus.

1.1.3 Bus Protocol

1.1.3.1 SD mode

In SD mode, six signals are used to interface to the cards: clk, cmd, and dat3-dat0. The clk signal is used to maintain synchronization between the system and the card. The cmd signal is used to issue the commands from the host to the cards and to return responses from the card to the host. The dat3-dat0 signals are used to write and read data to and from the card. The data transfer can be carried out in 1-bit or 4-bit mode. The cmd and dat3-dat0 signals are bidirectional bus signals.

1.1.3.2 MMC mode

In MMC mode, three signals are used to interface to the cards: clk, cmd and dat0. These three signals perform the same function as in SD mode.

1.1.4 Transfer modes

1.1.4.1 SD mode

SD mode supports two transfer methods: single block and multiple block. A default block size is stipulated according to the card; one block can be transferred in response to one command in single block transfer, and multiple blocks in multiple block transfer. The block size can be changed with block size change command. Moreover, the data be transferred in 1-bit or 4-bit bus mode.

1.1.4.2 MMC mode

MMC mode supports single block transfer mode. multiple block transfer mode and stream transfer mode. In stream transfer, there is no concept of block: any number of bytes can be transferred in byte units from any address. However, since CRCs are not used, this method is not suitable for applications requiring data reliability.

1.1.5 CRC

CRC (cyclic redundancy code) are added to SD/MMC commands, responses and data before they are transferred. Checking the CRC enables erroneous transfer of data to be detected, and allows re-transfer processing to be carried out when error is detected. In SD/MMC mode, CRCs are mandatory in transfer modes, except stream mode transfer. A 7-bit CRC is added to each command, and a 16-bit CRC to every block of data.

1.1.6 Commands

All processing, including card identification, reading writing and erasing, is initialized by commands issued from the host to the card. The command are listed in table 1-1.

Table 1-1

CMD index	Abbreviation	Function	Argument	Mode
CMD0	GO_IDLE_STATE	Card reset	[31:0] stuff bits	SD/MMC
CMD1	SEND_OP_COND	Poll operating voltage	[31:0] OCR without busy	MMC only
CMD2	ALL_SEND_CID	CID transmission request	[31:0] stuff bits	SD/MMC
CMD3	SET_RELATIVE_ADDR	RCA setting	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD4	SET_DSR	DSR setting	[31:16] DSR [15:0] stuff bits	SD/MMC
CMD7	SELECT/DESELECT CARD	Selection of SD/MMC to be accessed	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD9	SEND_CSD	CSD transmission request	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD10	SEND_CID	CID transmission request	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD11	READ_DAT_UNTIL_STOP	Stream read	[31:0] data address	MMC only
CMD12	STOP_TRANSMISSION	Read/write stop command	[31:0] stuff bits	SD/MMC
CMD13	SEND_STATUS	Card status request	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD15	GO_INACTIVE_STATE	Place card in inactive state	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD16	SET_BLOCKLEN	Read/write block length change	[31:0] Block length	SD/MMC
CMD17	READ_SINGLE_BLOCK	Single block read	[31:0] data address	SD/MMC
CMD18	READ_MULTIPLE_BLOCK	Multiple block read	[31:0] data address	SD/MMC
CMD20	WRITE_DAT_UNTIL_STOP	Stream write	[31:0] data address	MMC only
CMD24	WRITE_BLOCK	Single block write	[31:0] data address	SD/MMC
CMD25	WRITE_MULTIPLE_BLOCK	Multiple block write	[31:0] data address	SD/MMC
CMD26	PROGRAM_CID	CID programming	[31:0] stuff bits	MMC only

Table 1-1

CMD index	Abbreviation	Function	Argument	Mode
CMD27	PROGRSM_CSD	CSD programming	[31:0] stuff bits	SD/MMC
CMD28	SET_WRITE_PROT	Protection setting	[31:0] data address	SD/MMC
CMD29	CLR_WRITE_PROT	Protection clearing	[31:0] data address	SD/MMC
CMD30	SEND_WRITE_PROT	Protect bit status transmission	[31:0] write protect data address	SD/MMC
CMD32	ERASE_WR_BLK_START	Set address of the first write block to be erased	[31:0] data address	SD/MMC
CMD33	ERASE_WR_BLK_END	Set address of the last write block of the continuous range to be erased	[31:0] data address	SD/MMC
CMD35	ERASE_GROUP_START	Set address of the first group to be erased	[31:0] data address	MMC
CMD36	ERASE_GROUP_END	Set address of the last group to be erase	[31:0] data address	MMC
CMD38	ERASE	Erases selected area	[31:0] stuff bits	SD/MMC
CMD42	LOCK_UNLOCK	Password setting	[31:0] stuff bits	SD/MMC
CMD55	APP_CMD	Application command	[31:16] RCA [15:0] stuff bits	SD/MMC
CMD56	GEN_CMD	General purpose application command	[31:16] RCA [15:0] stuff bits	SD/MMC
ACMD6	SET_BUS_WIDTH	Define the data width	[31:2] stuff bits [1:0] bus width	SD
ACMD13	SD_STATUS	Send SD memory card status	[31:0] stuff bits	SD
ACMD22	SEND_NUM_WR_BLOCKS	Send the number of the written blocks	[31:0] stuff bits	SD
ACMD23	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing	[31:23] stuff bits [22:0] number of blocks	SD
ACMD41	SD_APP_OP_COND	Send OCR from the accessed card	[31:0] OCR without busy	SD
ACMD42	SET_CLR_CARD_DETECT	Connect/Disconnect the pull-up resistor on DAT3	[31:1] stuff bits [0] set_cd	SD
ACMD51	SEND_SCR	Read the SD configuration register	[31:0] stuff bits	SD

1.1.7 Response

When a command is issued from the host to a card, the card may return a response (depending on the command type) to the host in a format stipulated for each command. Response formats are as follows:

- R1, R1b - normal response commands: 48 bits
- R2 - CID and CSD register response command: 136 bits
- R3 - OCR register response commands: 136 bits

1.2 Features

The SDHC includes these distinctive features:

- Compatible with the SD Memory Card specification -- Physical Layer version 1.0 (SPI mode not supported)
- Upward compatibility to MMC System specification version 3.0. (SPI mode not supported)
- Programmable SD memory clock frequencies.
- one 16-byte FIFO for both transmit and receive.
- Hardware CRC7, CRC16 generation and validation for command and data tokens.
- Stream mode (MMC only) or block mode for data transfer.
- Maskable hardware interrupts.
- Auto card detection.
- Operating frequency: 60MHz.
- 20Mbps (1-bit bus mode) or 80Mbps (4-bit bus mode) maximum data rate.
- Support one SD card interface.

1.3 Modes of Operation

- Run mode

  The SDEN bit in SDCON should be set to enable the SDHC for data and command transfers.
- Wait mode

  Resetting the SDEN bit in SDCON will disable the SDHC (except registers).
- Freeze mode

  Same as the run mode.
- Stop mode

  Same as the run mode.

Section 2 External Signal Description

2.1 Overview

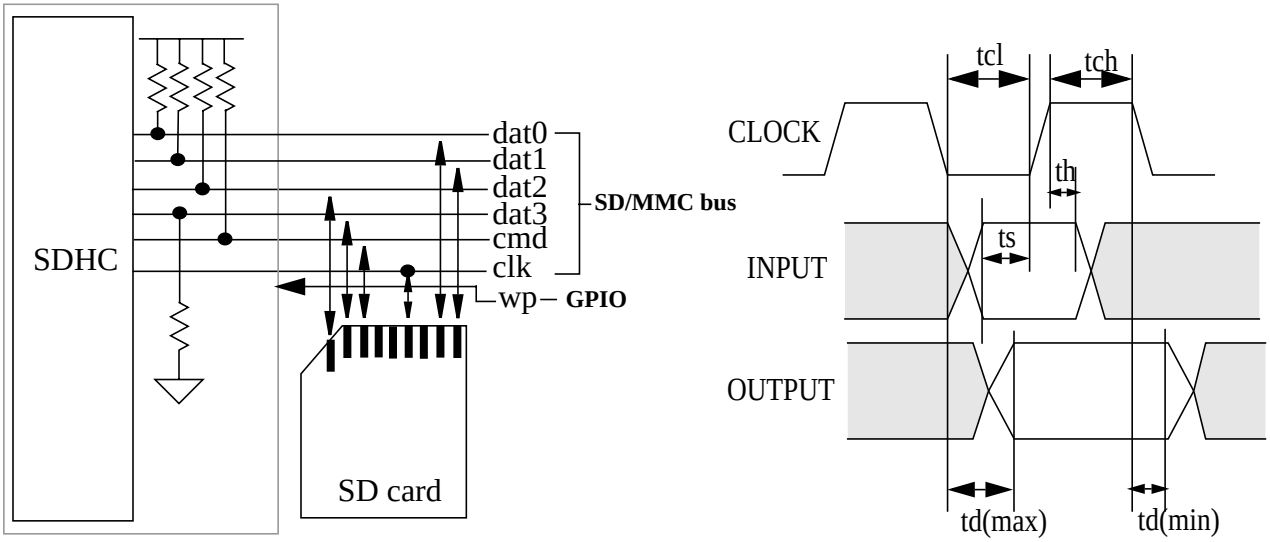


Figure 2-1 SD memory card Hardware Interface and I/O timing

Table 2-1

Parameter	Symbol	Min	Max	Unit
clock low time	tcl	10		ns
clock high time	tch	10		ns
Input hold time	th	5		ns
Input setup time	ts	5		ns
output delay time	td	0	14	ns

Table 2-2 Signal Properties

Name	Port	Function	Pull up/Pull down
clock	clk	SD memory card Clock	—
command	cmd	Transmit command/Receive response	Pull up disabled after reset; enabled when PUEN bit is set.
data 0	dat0	Data Transfer to and from the card for dat0	Pull up disabled after reset; enabled when PUEN bit is set.
data 1	dat1	Data Transfer to and from the card for dat1	Pull up disabled after reset; enabled when PUEN bit is set.
data 2	dat2	Data Transfer to and from the card for dat2	Pull up disabled after reset; enabled when PUEN bit is set.
data 3	dat3	Data Transfer to and from the card for dat3	Pull down disabled after reset; enabled when PDEN bit is set.
write protect	wp	A given card is write-protected or not	—

2.2 Detailed Signal Descriptions

2.2.1 clk — SD/MMC card clock

This signal is to provide a clock signal to the SD/MMC card. The maximum clock frequency for SD/MMC card is 20MHz.

2.2.2 cmd — Bidirectional command/response signal

This signal is used to send a command from the host to the card(s) or receive a response from the card to the host.

2.2.3 dat0 — Bidirectional data signal for DAT0

This signal provides data transfer between the host and the card.

2.2.4 dat1 — Bidirectional data signal for DAT1

This signal provides data transfer between the host and the card.

2.2.5 dat2 — Bidirectional data signal for DAT2

This signal provides data transfer between the host and the card.

2.2.6 dat3 — Bidirectional data signal for DAT3

This signal provides data transfer between the host and the card. It is also used for card detection.

2.2.7 wp — write protect

This signal is used to indicate a given card is write-protected or not and implemented in the general purpose I/O port (GPIO).

Section 3 Memory Map/Register Definition

The SDHC memory map includes 14 16-bit registers as shown in **Table 3-1**. These registers define parameters to bridge the system and the SD/MMC card. All registers (except SDSTAT and SDRSP) can be read or written. Attempts to write a read-only register will have no effect. All registers may be accessed as bytes (8-bit) or words (16-bit) except SDATA which must be word (16-bit) access.

Table 3-1 Module Memory Map

Address	Use	Access
$00	SDHC Control Register (SDCON)	R/W
$02	SDHC Status Register (SDSTAT)	R
$04	SD Clock Rate Register (SDCLKCON)	R/W
$06	Command and Data Control Register (SDCMDATCON)	R/W
$08	Response Timeout Register (SDRTOUT)	R/W
$0A	Read Timeout Register (SDRDTOUT)	R/W
$0C	Block Length Register (SDBLKLN)	R/W
$0E	Number of Blocks Register (SDNOBLK)	R/W
$10	Interrupt Enable Register (SDINTREN)	R/W
$12	Command Register (SDCMDNO)	R/W
$14	Argument High Register (SDARGH)	R/W
$16	Argument Low Register (SDARGL)	R/W
$18	Response Register (SDRSP)	R
$1A	SD Data Register (SDATA)	R/W
$1C	Reserved	--
$1E	Reserved	--

R: Read

W: Write

3.1 Register Descriptions

Register address offset: $00

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	LBUF	IRST	PUEN	PDEN	QIEN	SDEN
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-1 SDHC Control Register (SDCON)

LBUF — Last data buffer

In multiple write block mode, this bit indicates that the next block of data is the last data block to be sent. In multiple read block mode, it indicates one more data read block is required.

1 = the last block of data to be sent or received
0 = two or more blocks of data to be sent or received

IRST— Internal reset enable

This bit resets the SDHC to its reset state. To generate the internal reset, this bit should be set first, then write the SDCON with any value.

1 = internal reset enable
0 = internal reset disable

PUEN — Pull-up enable

This bit is used to control the internal pull-up resistors for command and dat0, dat1 and dat2 pins.

1 = pull-up enable
0 = pull-up disable

PDEN — Pull-down enable

This bit is used to control the internal pull-down resistor for dat3 pins.

1 = pull-down enable
0 = pull-down disable

QIEN — fast data transfer enable

This bit is used to enable data transfer between the system and the SDHC via the IQUE module which provides faster data transfer rate.

1 = IQUE transfer
0 = programmed I/O transfer

SDEN — module enable

This bit will be used to enable or disable the SDHC for power-saving purpose. When disabled, no data or command transfer can be started. However, the registers can also be read and written.

1 = module enable
0 = module disable

Register address offset: $02

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	BLK_RDY	FFULL	FEMP-TY	CD	ECR	WRDN	DTDN	SDCK-ON	WR_ECRC_CODE		RSP_ECRC	RD_ECRC	WR_ECRC	RSPTO	RDTO
W																
RESET:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-2 SDHC Status Register (SDSTAT)

BLK_RDY — Data block ready

Data block is about to start to be transmitted.
1 = block data start
0 = block data is not going to start

FFULL — Data FIFO full

Data FIFO full status for both transmit and receive operations.
1 = transmit or receive FIFO full
0 = transmit or receive FIFO not full

FEMPTY — Data FIFO empty

Data FIFO empty status for both transmit and receive operations.
1 = transmit or receive FIFO empty
0 = transmit or receive FIFO has at least one word of data

CD — Card Detected

This bit is used to show a SD/MMC card is inserted and detected.
1 = memory card inserted and detected
0 = no memory card inserted

ECR — End command response

It is to indicate that a response is received successfully for the last command and it is now ready for the next command.
1 = next command ready
0 = current command not completed

WRDN — Write SD/MMC card done

This bit is to show the write operation to the card is completed. After every write operation, a busy status is returned. When deasserted, it shows the card is programmed successfully.
1 = write card operation completed
0 = write card operation not finished

DTDN — Data transfer done

In write and read operations, this bit indicates that the current data transfer is completely finished (including the CRC bits).

1 = Data transfer done
0 = Data transfer not done

SDCKON — SD/MMC card clock running

This bit is to indicate the current status of SD card clock.
1 = SD card clock running
0 = SD card clock stopped

WR_ECRC_CODE — Write CRC Error code

These two bits are valid only if the WR_ECRC bit is set.

Table 3-2 Write CRC error code

Value	Meaning
00	No transmit error (CRC status = 010)
01	Transmit error (CRC status = 101)
10	No CRC response (CRC status = 111)
11	Reserved

RSP_ECRC — Response CRC error

An transmission error occurs in Command - Response sequence. This bit is cleared by read.
1 = response CRC error
0 = no response CRC error

RD_ECRC — CRC read error

An transmission error occurs during reading data from the SD/MMC card. This bit is cleared by read.
1 = read data CRC error
0 = no read data CRC error

WR_ECRC — CRC write error

It indicates that an transmission error occurs when writing data to the SD/MMC card. This bit is cleared by read.
1 = write data CRC error
0 = no write data CRC error

RSPTO — Response timeout

No response is received in the pre-defined time period from the SD/MMC card after sending a command. The timeout value is defined in RTOUT. This bit is cleared by read.
1 = response timeout
0 = response not timeout

RDTO — Read data timeout

The expected data read from the SD/MMC card is not received in the pre-defined time period specified in SDRDTOUT. This bit is cleared by read.
1 = read data timeout
0 = read data not timeout

Register address offset: $04

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	0	0	SCLK-EN	CLKRATE		
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

= Unimplemented or Reserved

Figure 3-3 SD Clock Control Register (SDCLKCON)

SCLKEN — SD/MMC card clock enable

This bit is to enable the SD card clock for transfers.

1 = SD card clock enable
0 = SD card clock disable

CLKRATE — SD/MMC card clock rate

Programmable clock rate for SD card clock. For the operating frequency to be 60MHz, the recommended clock rate value is 010 so that the SD/MMC clock is set to 20MHz. The SDHC will be operated at the maximum throughput. Setting the clock rate to 001 will put the SD/MMC clock to 30MHz. The data and command transfers for the SDHC are not guaranteed. Therefore, it is not recommended.

Table 3-3 SD Clock rate

Value	Frequency
000	Unimplemented
001	1/2 Module clock
010	1/3 Module clock
011	1/4 Module clock
100	1/5 Module clock
101	1/6 Module clock
110	1/10 Module clock
111	400kHz

Register address offset: $06

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	NOBEN	WBUS	0	INIT	BUSY	MBLK	SB-MOD	WRD	DA-TAEN	RSPNO	
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-4 Command and Data Control Register (SDCMDATCON)

NOBEN — NOB enable

This bit enables the use of no of blocks in multiple block mode.
1 = NOB enable
0 = NOB disable

WBUS — Wide bus selection

This bit specifies whether the data transfer is in 1-bit mode or in 4-bit mode.
1 = 4-bit bus mode
0 = 1-bit bus mode

INIT — Card initialization

This bit is to prefix the command with 80 SD/MMC clock for card initialization.
1 = SD clock init enable
0 = SD clock init disable

BUSY — Expected busy status

A busy status is expected after the current command. It includes the stop transmission command, card select, erase, program CSD, etc. Refer to the SD Memory Card Specification for details.
1 = busy status expected
0 = busy status not expected

MBLK — Multiple block mode

This bit specifies whether the data transfer of the current command is a multiple block mode command.
1 = multiple block mode
0 = single block mode

SBMOD — Data transfer mode selection

This bit specifies whether the data transfer of the current command is in stream or block mode.
1 = stream mode
0 = block mode

WRD — Data read or write operation

This bit specifies whether the data transfer of the current command is a write or read operation.
1 = write operation
0 = read operation

DATAEN — Data transfer enable

This bit specifies whether the current command includes a data transfer.
1 = data transfer enabled
0 = no data transfer

RSPNO — Response types

This field specifies the response types. The response types (R1 - R3) can be referred to Section 4.9 in SD Memory Card specification -- Physical Layer version 1.0 for details.

Table 3-4 Response types for SD card

Description	Response types for SD card
No response	00
Format R1	01
Format R2	10
Format R3	11

Register address offset: $08

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	RSP_TO						
W																
RESET:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-5 Response Timeout Register (SDRTOUT)

RSP_TO — Response timeout

This register specifies the number of SD/MMC card clocks after the command issued and before the host turns on the time-out error for the received response. The maximum and the minimum values are 64 and 2 respectively. Writing any values less than 2 or greater than 64 will set the value to 2 or 64 respectively. The default value is 64.

Register address offset: $0A

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RD_TO															
W																
RESET:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3-6 Read Timeout Register (SDRDTOUT)

RD_TO — Read timeout

This register specifies the number of the SD/MMC card clocks after the command, before the host turns on the time-out error for the received data. 0 indicates no timeout. The minimum value is 2. Writing any values less than 2 will set the value to 2. The maximum value depends on the value in the CSD register from the SD/MMC card. The default value is 65536 in unit of 256*(SD card clock).

Register address offset: $0C

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	BLEN									
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

= Unimplemented or Reserved

Figure 3-7 Block Length Register (SDBLKLN)

BLEN — Block length

This register specifies the maximum number of bytes in a single block. The value is specified in the CSD register from the SD/MMC card. The maximum and minimum values are 512 and 1. Writing any values less than 1 or greater than 512 will set the value to 1 or 512 respectively.

Register address offset: $0E

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NOB															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Figure 3-8 Number of Blocks Register (SDNOBLK)

NOB — Number of blocks

This register specifies the number of blocks in block mode transmission. The minimum number of the block is a single block. Writing any values less than 1 will set the value to 1.

Register address offset: $10

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	CDIE	ECRIE	WRD-NIE	DTD-NIE	CRCIE	TOIE
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-9 Interrupt Enable Register (SDINTREN)

CDIE — Auto card detect interrupt enable

This bit is used for enabling card detection interrupt. After a card has been detected, this bit should be cleared to disable the card detection interrupt generation during the normal data transfer.

1 = card detect interrupt enable
0 = card detect interrupt disable

ECRIE — End command response interrupt enable

Interrupt enable for end command response.

1 = end command response interrupt enable
0 = end command response interrupt enable

WRDNIE — Write operation done interrupt enable

Interrupt enable for write operation done.

1 = write operation done interrupt enable
0 = write operation done interrupt disable

DTDNIE — Data transfer done interrupt enable

Interrupt enable for data transfer done.

1 = data transfer done interrupt enable
0 = data transfer done interrupt disable

CRCIE — CRC error interrupt enable

This bit is to enable the CRC error interrupt. The RSP_ECRC bit, RD_ECRC bit or WR_ECRC bit may be checked to determine the sources of CRC error.

1 = CRC error interrupt enable
0 = CRC error interrupt disable

TOIE — Timeout interrupt enable

This bit is to enable the timeout interrupt. There are two sources of timeout conditions, response timeout and read data timeout. The RSPTO bit and RDTO bit in the SDSTAT register should be read to determine which source produces the timeout interrupt.

1 = timeout interrupt enable
0 = timeout interrupt disable

Register address offset: $12

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	CMDNO					
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-10 Command Register (SDCMDNO)

CMDNO — Command number

This register specifies the command number.

Register address offset: $14

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HARG															
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-11 Argument High (SDARGH)

HARG — High-word argument

This register specifies the most significant word part of the argument in the current command.

Register address offset: $16

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LARG															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-12 Argument Low Register (SDARGL)

LARG — Low-word argument

This register specifies the least significant word part of the argument in the current command.

Register address offset: $18

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CRSP															
W																
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

= Unimplemented or Reserved

Figure 3-13 Response Register (SDRSP)

CRSP — SD/MMC card response

This register will contain the response after every command sent to the host.

Register address offset: $1A

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SDAT															
RESET:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 3-14 SD/MMC Data Register (SDATA)

SDAT — SD/MMC fifo data

Reads of SDATA return data from the receive data FIFO. Writes store data to the transmit data FIFO. Both transmit and receive are buffered through a 16-byte FIFO. Reads of an empty FIFO return undefined data.

Section 4 Functional Description

4.1 Reset

4.1.1 Module Reset

An asynchronous power-on reset puts the module to reset states. FIFO content and pointer are cleared Registers are reset to their reset values, returning the module to the idle state.

4.1.2 Internal Reset

Besides the asynchronous reset from power-on, the module can also be reset by setting the IRST bit. When the IRST bit is set, any write to SDCON will produce an internal reset, the module will go into reset state.

4.2 SD/MMC Clock Control

The SDHC has a clock control unit which can start and stop the SD/MMC clock for power-saving purpose, as well as controlling the data transfer for the SD/MMC card.

4.2.1 Programmable SD clock

The SD/MMC clock can be scaled to 20MHz or lower frequencies for SD/MMC card. To change the SD card clock frequency, the application has to update the clock rate value in the SDCLKCON. Normally, this register should be programmed after the system reset but before starting any transfer.

4.2.2 Enable/disable SD clock

The SD/MMC card clock can be enabled or disabled by setting or clearing the SCLKEN bit. For normal operation, the SD/MMC card clock should be enabled before command/data transfer.

4.3 Card Detection

dat3 is used for card detection. It may be programmed to be pulled down. When a SD card/MMC card is inserted, dat3 is pulled up by the card’s internal pull-up resistor. An interrupt is generated to indicate that a card has been inserted and detected.

4.4 Card Identification

4.4.1 Card Reset

A SD/MMC card does not have an external reset pin; instead, a reset command is issued for this purpose. A reset command can be executed by means of the GO_IDLE_STATE (CMD0) command. When a reset command is executed, the SD/MMC card is forced to the Idle state regardless of the current state. However, a SD/MMC card in the Inactive state will ignore this reset command. All the card's output pins go to high-impedance state and each card's RCA register value is set to it initial value.

4.4.2 Operating Voltage Range Validation

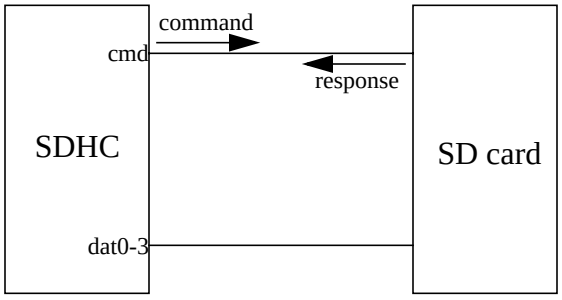
All cards should be able to establish communication with the SDHC using any operating voltage in the maximal allowed voltage range specified in this standard. The supported minimum and maximum values are defined in the OCR register.

4.4.3 Card Initialization

The SDHC starts the card identification process with the identification clock rate (400kHz). Cards are required to send their valid operation conditions (ACMD 41 preceding with APP_CMD - CMD55 with RCA = 0x0000). The ALL_SEND_CID (CMD2) is issued to each card to get its unique card identification. Thereafter, CM3 (SEND_RELATIVE_ADDR) is sent to ask cards to publish a new relative card address (RCA) which will be used in address the card in future data transfer.

4.5 Command-Response sequence

The command-response sequence is performed on the SD/MMC bus. This sequence may end with card response or no card response, depending on the command type. The command control unit formats the command and, if applicable, it generates and appends CRC bits, waits for the response start bit, receives the response data, validate the card CRC and adds eight SD/MMC card clock after the card response end bit. Moreover, it validates all card response CRC except type R3, the CSD and CID registers. The timings can be referred to Section 4.12 in SD Memory Card specification -- Physical Layer version 1.0 for details.



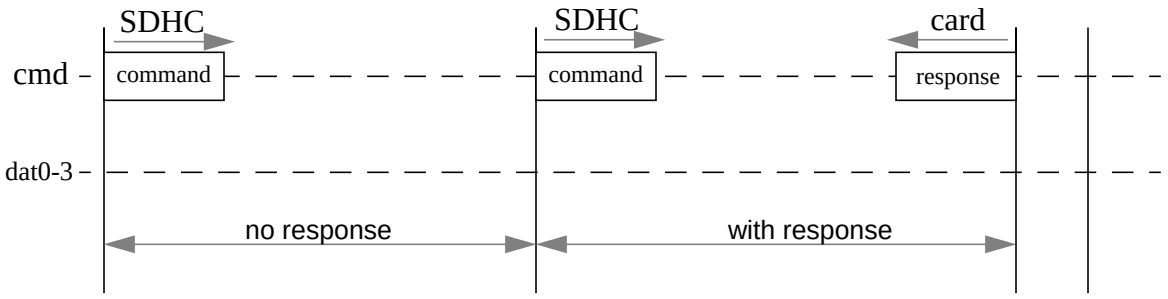


Figure 4-1 Command-response sequence

4.6 Data Transfer

The data transfer between the system and the SD/MMC card will be assisted by the use of FIFO. The FIFO data can be transferred via IPBus or IQUE interface. The IQUE interface is used for fast data transfer between the system and the SDHC.

To start the data transfer, with proper configuration for write operation, one can write data to the FIFO. At the same time, whenever the FIFO is not empty, data will transferred to the card until the FIFO is empty or block transfer finishes, while, for read operation, the card writes data to the host until the FIFO is full. The FIFO is read until it is empty, then the card resumes sending data to the system.

At the end of any data transfer or busy status on the SD/MMC bus, eight SD/MMC card clock is added before notifying the application that the data transfer is done.

It is not allowed to disconnect a card while data is transmitting or receiving data on the SD/MMC data bus. At any other time, the system can connect or disconnect card, as described in the SD card specifications. The timings can be referred to Section 4.12 in SD Memory Card specification -- Physical Layer version 1.0 for details

Before data transfer starts in SD/MMC mode, an initial setting up is required. Initial settings are used to carry out the following for all cards on the bus:

- acquisition and specification of card operating conditions.
- obtaining card attribute information.
- specification of relative addresses of cards on the bus.

ACMD1 for SD/CMD1 for MMC is used to acquire and specify card operating conditions. CMD2 for SD/MMC is applied to get card attribute information. Then, CMD3 will also be used to specify relative addresses of cards on the bus.

After the above procedures have been completed, data transfer commands such as data read and data write commands can be used on the SD/MMC card.

4.6.1 Block Data Write

This operation is performed after the command-response sequence as described in Section **4.5**. The system will fill the FIFO and indicated that the FIFO is ready. The basic data write transaction is performed. It generates the data start bit, sends the data, generates and appends the CRC, receives response CRC, validates the CRC response and waits for the busy status from the SD/MMC card to end.

In a single block write, one block of data is transmitted and the number of the bytes for this block is specified in SDBLKLN. In a multiple block write, a single block data write transfer on the SD/MMC bus is performed several times. The no of block may also need to be specified in SDNOB.

It is waited until the current block of data is transferred completely before filling the FIFO with the next block of data. The stop transmission command is generated after the last block of data is transmitted. The system will indicate that the last block of data is sent.

At the end of every busy status after every block of data transfer, eight SD/MMC card clock is waited before notifying the application that it can fill the FIFO with next block of data. This is necessary for the card to finish its internal operations properly.

If the system fails to write to the FIFO on time, the SD/MMC card clock may be turned off so as not to lose data on the SD bus.

In case the system starts a write sequence and detects an error condition. An interrupt will be generated and the SDSTAT should be checked.

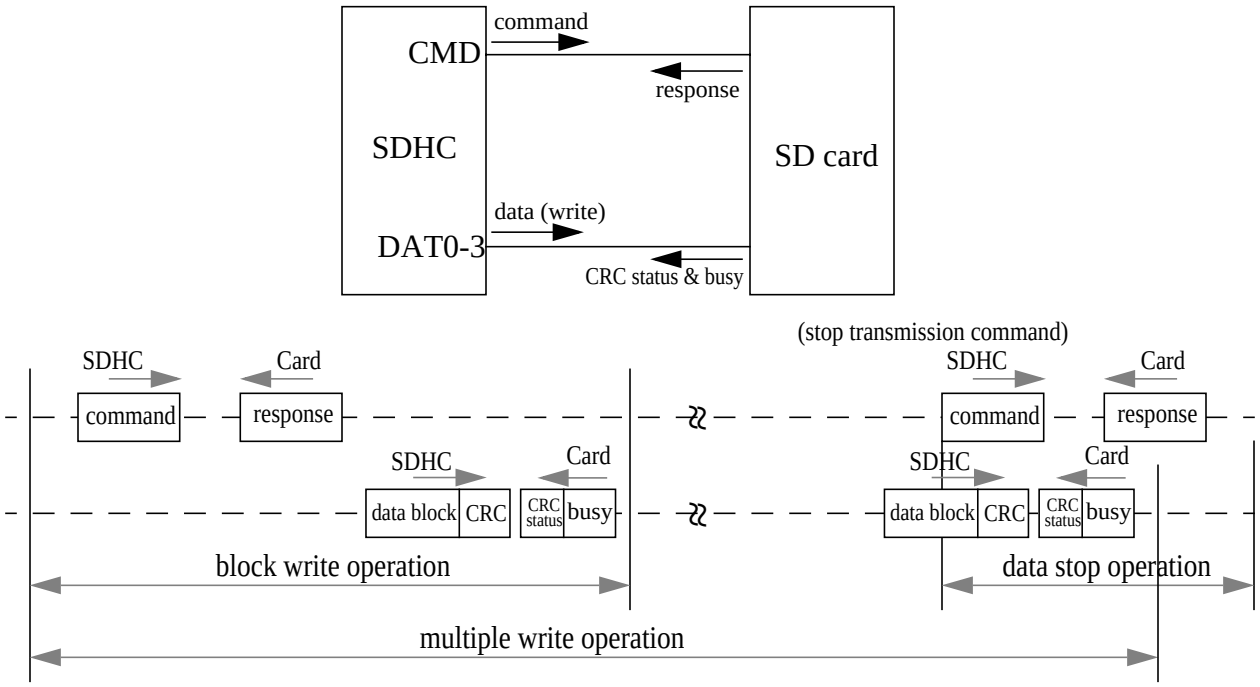


Figure 4-2 (Multiple) Block write operation

4.6.2 Block Data Read

After performing the command-response sequence as described in Section **4.5**, in the basic read sequence, it waits for the data start bit from the card. Then, it starts to receive the data and validates the CRC.

In a single block read, one block of data is read from the card. In a multiple block read, a single block data read transfer is performed on the SD bus many times.

After the card has finished filling the FIFO, it signals the system that the FIFO is full. The system then empties the FIFO. If the system fails to read from the FIFO on time, the SD/MMC card clock may be stopped so as not to lose data on the SD bus.

Before every block of data, it is checked whether the data has arrived before the time specified (RDTOUT) by the application.

In multiple block read, before reading the last block of data, it is needed to notify that the next block of data is the last one. This is required to prevent reading data if block count is not used.

At the end of a data block in a multiple block read, if a CRC error is detected on the incoming data block, it informs the system that an error has occurred by generating an interrupt and checks the SDSTAT. In this case, the system has to send a stop transmission command to the card.

After the end bit of the last data block is received, the application can send the stop transmission command and read the last data. The SD/MMC card clock will be stopped.

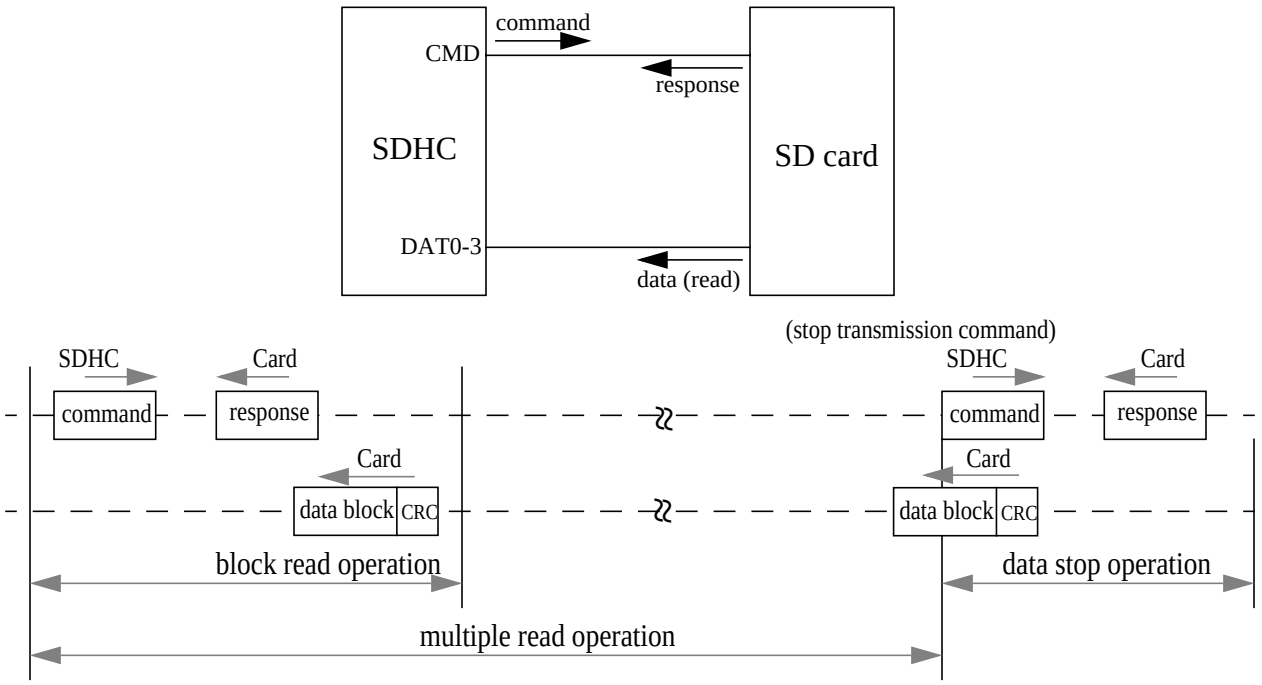


Figure 4-3 (Multiple) Block read operation

4.6.3 Stream Data Write (MMC mode only)

In steam data write, the SDHC generates the data start bit and sends the data. The system has to fill the FIFO with the stop transmission command and its parameters as the last data to be sent.

At the last buffer, the application must fill the buffer leaving out the last 8 bytes. The next step is to send the last 8 bytes of data and the stop transmission command together. BUSY bit should be set in the SDCMDATCON register.

According to MMC specification, in stream data transfer mode, the stop transmission command is to be sent during the data transfer and cannot be sent after the data is finished.

4.6.4 Stream Data Read (MMC mode only)

In stream data read, the SDHC waits for the start data bit, receives the data and stops the data read at the end of the data stop transmission command.

4.7 Busy sequence

A busy status is expected automatically after every block of data write operation in single or multiple block mode. It will also expect a busy status at the end of a command every time the system specifies that a busy signal is expected (after stop transmission command, card select, erase, program CSD, etc.). When busy status is expected, the system should wait for WRDN=1.

While a busy status is indicated in the SD/MMC bus, sending data over the SD/MMC bus is not allowed and only command can be transmitted. When deselecting the card, the busy status will be deasserted. When reselecting the card, busy status should be checked to see if the card is busy.

4.8 Hardware Interrupts

The interrupt is generated only at the following points:

- SD card is inserted and detected.
- The end of command-response sequence.
- At the end of successful write operation.
- data transfer is done.
- CRC error.
- Timeout conditions, such as response or read timeout.

After an interrupt has occurred, the application should read SDSTAT register to check the status and the interrupt will be cleared automatically.

4.9 Error Detection

The SDHC can detect the following errors on the SD bus:

- Response CRC error.
- Response time out.
- Write CRC error.
- Read CRC error.
- Read data timeout error.

If an error is detected, the application will be notified by the corresponding interrupt. The SDSTAT should be checked then.

4.10 Wide bus selection

The SD data bus can be operated either in single bus mode (1 bit) or wide bus mode (4 bit), which can be changed (WBUS) only after a card is selected. The default bus width is 1 bit.

4.11 CRC Generation

The CRC is intended to validate the SD memory card command, responses and data transfers. One CRC is generated for every command and checked for every response on the command line. For data blocks one CRC per transferred block is generated.

4.11.1 CRC7

The CRC7 is applied to all commands for all response except type R3, and for the CSD and CID registers. The CRC7 is a 7-bit value. The number of bits to be protected is 40 for commands and responses, and 120 for the CSD and CID.

4.11.2 CRC16

In case of one DAT line usage, the CRC16 is used for data protection in block transfer mode. The CRC checksum is a 16-bit value. The same CRC16 method will be used in single DAT line mode and in wide bus mode. In wide bus mode, the CRC16 is done on each line separately.

Section 5 Initialization/Application Information

5.1 Startup initialization

After reset, the module is in disabled state. It is required to set SDEN = 1 to enable the module before any transfer can be started.

5.2 SD/MMC Card Clock Control

The SD/MMC card clock can be disabled or enabled. Before sending any command/data on SD/MMC bus, SCLKEN should be enabled. i.e., set SCLKEN = 1.

The status of the SD card clock can be checked by reading SDCKON bit in SDSTAT to see if the SD card clock has stopped or started.

5.3 Prefix a Command with 80 Clock Cycles

To prefix a command with 80 SD clock cycles,

- set INIT=1 in SDCMDATCON.

5.4 Command-Response Sequence

In the command-response sequence,

- Command code is written to SDCMDNO.
- The MSB and LSB arguments are updated in SDARGH and SDARGL.
- In SDCMDATCON, the bits are configured as follows:
  - update RSPNO bits.
  - set NOBEN=0; WBUS=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=0; SBMOD=0; WRD=0;DATAEN=0.

When command-response sequence is finished, an interrupt is generated. It is required to check the ECR bit in SDSTAT for completed transfer and the response can be checked in SDRSP. Any write to SDCMDARCON will issue the command/data transfer.

5.5 Erase

An erase command is performed as command-response transaction with the following additions.

The BUSY bit must be set to ‘1’. After reading response from SDRSP, the application should check for busy status to end by waiting for WRDN=1 in SDSTAT.

5.6 Single Data Block Write

In the single block write command,

- Specify and send the write command in command-response sequence.
- Program BLEN in SDBLKLN.
- Command code is updated in SDCMDNO.
- The MSB and LSB arguments are updated in SDARGH and SDARGL.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=0; SBMOD=0; WRD=1;DATAEN=1.

The system starts to write data if the FIFO is ready. When the FIFO is full, the SD/MMC card clock will be stopped until the FIFO is ready. The module will also stop transmitting data when the bytes sent to the SD/MMC card reach the specified block length value.

The system waits until the data transfer has finished successfully (DTDN=1) and the busy status is deasserted (WRDN=1).

5.7 Single Data Block Read

To perform a single block read,

- Specify and send the read command read in command-response sequence.
- Program BLEN value in SDBLKLN.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=0; SBMOD=0; WRD=0;DATAEN=1.

The system will read data from the FIFO whenever the FIFO is ready and continues until there is no data left in the FIFO.

5.8 Multiple Block Write

To perform a multiple block write,

- Specify and send the write command in command-response sequence.
- Program BLEN in SDBLKLN.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=1; SBMOD=0; WRD=1; DATAEN=1.

After the SDCMDATCON has been written and command transfer is initialized, the system must wait for ECR interrupt. At this point, it reads the response from SDRSP. The system writes data to the FIFO block by block until the last data block. At the last block, the application updates the stop transmission command, and writes data to the last block. The application must then wait for DTDN =1. The stop transmission command is sent with the BUSY bit set to 1. After receiving the response the application waits for the end of the busy status (WRDN=1) in SDSTAT or sends the send status command to the card and check the state of the card in the response.

5.9 Multiple Block Read

To perform a multiple block read,

- Specify and send the read command in command-response sequence.
- Program BLEN in SDBLKLN.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=1; SBMOD=0; WRD=0; DATAEN=1.

After the SDCMDATCON has been written and command transfer is initialized, ECR interrupt is checked. At this point, the application reads the response from SDRSP. The system read data from the FIFO whenever the FIFO is ready and continues for a data block. This is repeated until the last data block. Before the last block, it is required to inform the next block is the last block (LBUF=1). After the last data block is received, the stop transmission command is sent.

5.10 Multiple Block Write using Number of Blocks

To perform a multiple block write using number of blocks,

- Specify and send the write command in command-response sequence.
- Program BLEN in SDBLKLN.
- Program NOB in SDNOBLK.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=1; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=1; SBMOD=0; WRD=1; DATAEN=1.

After the SDCMDATCON has been written and command transfer is initialized, the system does exactly what it is specified in Multiple Block Write section until the last block. At the last block, the LBUF bit is set. The system waits for the end of the busy status (WRDN=1) in SDSTAT or sends the send status command to the card and checks the state of the card in the response.

5.11 Multiple Block Read using Number of Blocks

To perform a multiple block read using number of blocks,

- Specify and send the read command in command-response sequence.
- Program BLEN value in SDBLKLN.
- Program NOB value in SDNOBLK.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - WBUS bit is set as required.
  - set NOBEN=1; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=1; SBMOD=0; WRD=0; DATAEN=1.

After the SDCMDATCON has been written and command transfer is initialized, the system follows exactly what it is specified in Multiple Block Read section.

5.12 Stream Write

To perform a stream write,

- Specify and send the write command itself in command-response sequence.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - set NOBEN=0; WBUS=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=0; SBMOD=1; WRD=1; DATAEN=1.

After the response has returned and been checked, the system fills the FIFO. It should be noted that the system has to fill the FIFO with the stop transmission command and its parameters as the last data to be sent.

At the last buffer, the system must fill the buffer leaving out the last 8 bytes. The next step is to send the last 8 bytes of data and the stop transmission command together. BUSY bit should be set in the SDCMDATCON register.

5.13 Stream Read

To perform a stream read,

- Specify and send the read command in command-response sequence.
- Program RD_TO value in SDRDTOUT.
- SDCMDATCON is updated as follows:
  - RSPNO bits are updated.
  - set NOBEN=0; WBUS=0; NOCMD=0; INIT=0; BUSY=0;
  - set MBLK=0; SBMOD=1; WRD=1; DATAEN=1.

The system read the FIFO whenever it is ready. It will stop the data read when the stop transmission command is issued.

Index

Block Guide End Sheet

FINAL PAGE OF
42
PAGES