SAMSUNG microSD Card

Product Datasheet

Version 0.3
March 2007

Document Title

SAMSUNG microSD Card

Revision History

Revision No	History	Draft Date	Remark
0.0	Initial issue	June.15.2006	
0.1	Added 256MB microSD Card	October.16.2006	
0.2	Added 512MB & 1GB microSD Card in product line-up Updated Card register values Several typos were corrected	December.4.2006	
0.3	Added 2GB microSD Card in product line-up	March. 7. 2007	

TABLE OF CONTENTS

1. Ordering Information....4

2. Product Line-up....4

3. Introduction
   3.1 General Description....5
   3.2 System Features....5

4. Mechanical Specifications for microSD Card
   4.1 Card Package....6
   4.2 Mechanical Form Factor....8

5. Interface Description
   5.1 Pin Assignment....12
   5.2 microSD Card Bus Topology....13
   5.3 SPI Bus Topology....15
   5.4 Electrical Interface....16
   5.5 microSD Card Registers....21

6. microSD Card Protocol Description
   6.1 microSD Card Bus Protocol....33
   6.2 microSD Card Functional Description....37
   6.3 Card Identification Mode....38
   6.4 Data Transfer Mode....41
   6.5 Clock Control....62
   6.6 Cyclic Redundancy Codes....62
   6.7 Error Conditions....64
   6.8 Commands....65
   6.9 Card State Transition....72
   6.10 Responses....73
   6.11 microSD Card Status....75
   6.12 Memory Array Partioning....77
   6.13 Timing Diagrams....79
   6.14 Data Read....81
   6.15 Data Write....82
   6.16 Timing Values....84

7. SPI Mode
   7.1 Introduction....85
   7.2 SPI Bus Protocol....85
   7.3 SPI Command Set....91
   7.4 Responses....97
   7.5 Data Tokens....99
   7.6 Data Token Error....99
   7.7 Clearing Status Bits....100
   7.8 Card Registers....101
   7.9 SPI Bus Timinig Diagrams....102
   7.10 Timing Values....105
   7.11 SPI Electrical Interface....105
   7.12 SPI Bus Operation Conditions....105
   7.13 Bus Timing....105

Samsung may make changes to specifications and product descriptions at any time, without notice.

1. Ordering Information

M X X X X X X X X X X X - X X X X X

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

1. Module: M

2. Module Configuration
   C: Flash Card (SLC)
   M: Flash Card (MLC)

3~4. Flash Density

64 : 64M	28 : 128M
56 : 256M	12 : 512M
5D : 512M DDP	1G : 1G
1D : 1G DDP	2G : 2G
2D : 2G DDP	4D : 4G DDP
4Q : 4G QDP	8Q : 8G QDP

5. Feature
   R : microSD

6~8. Card Density

008 : 8M Byte	016 : 16M Byte
032 : 32M Byte	048 : 48M Byte
064 : 64M Byte	096 : 96M Byte
128 : 128M Byte	192 : 192M Byte
256 : 256M Byte	384 : 384M Byte
512 : 512M Byte	01G : 1G Byte
02G : 2G Byte	04G : 4G Byte
08G : 8G Byte	

9. Card Type
   U : microSD (Normal)

10. Component Generation

M : 1st Generation	A : 2nd Generation
B : 3rd Generation	C : 4th Generation
D : 5th Generation	

11. Flash Package

C : CHIP	Y : TSOP1
V : WSOP	B : TBGA

12. PCB Revision AND Production Site

13. " - "

14. Packing Type
   0 : Bulk Type I
   1 : Bulk Type II (By White Case)
   2 : Bulk Type I (No Label)
   3 : Bulk Type II (No Label)
   4 : Bulk Type I (only Back Label)
   5 : Bulk Type II (only Back Label)

15. Controller
   P: Sillicon Motion

16. Controller Firmware Revision
   A : None
   B : 1st Rev.
   C : 2nd Rev.
   D : 3rd Rev.
   E : 4th Rev.

17 ~ 18. Customer Grade
   " Customer List Reference "

2. Product Line-up

Model Number	Capacities	Remarks
MC12R064UBCY - XPAXX	64MB	microSD Card
MC1GR128UACY - XPAXX	128MB	
MC2GR256UACY - XPAXX	256MB	
MM4GR512UACY - XPAXX	512MB	
MM4GR01GUACY - XPAXX	1GB	
MM4GR02GUACY - XPAXX	2GB	

3. Introduction

3.1 General Description

The microSD is a memory card that is specifically designed to meet the security, capacity, performance and enviroment requirements inherent in newly emerging audio and video consumer electronic devices. The microSD will include a copyright protection mechanism that complies with the security of the SDMI standard and will be faster and capable for higher Memory capacity. The microSD security system uses mutual authentication and a "new ciper algorithm" to protect from illegal usage of the card content. A none secured access to the user's won content is also available. The microSD communication is based on an advanced 8-pin interface (Clock, Command, 4xData and 2xPower lines) designed to operate in at maximum operating frequency of 50MHz and low voltage range. The communication protocol is defined as a part of this specification.

3.2 System Features

- Complinat with SD Memory Card Specifications PHYSICAL LAYER SPECIFICATION Version 1.01 and 1.10
- Supports SPI Mode.
- Targeted for portable and stationary applications
- Voltage(VDD) Range:

  Basic communication (CMD0, CMD15, CMD55, ACMD41): 2.0 - 3.6V[1]

  Other commands and memory access: 2.7 - 3.6V
- Default mode: Variable clock rate 0~25MHz, up to 12.5MB/sec interface speed (using 4 parallel data lines)
- High-Speed mode: Variable clock rate 0~50MHz, up to 25MB/sec interface speed (using 4 parallel data lines)
- Switch function command supports High-Speed and futrue functions
- Correction of memory field errors
- Copyrights Protection Mechanism - Complies with highest security of SDMI standard.
- Password Protected of Cards
- Apllication specific commands
- Card Detection (Insertion/Removal)
- Built-in write protection features(permanent and temporary)
- Comfortable erase mechanism
- Protocol attributes of the communication channel is described below :

microSD Card Commuication Channel
Six-wire communication channel(clock, command, 4 data line)
Error-protected data transfer.
Single or multiple bolock oriented data transfer.

1) It is highly recommended not to use voltage range 2.0 - 2.7 for basic communication

4. Mechanical Speicfication for microSD Card

4.1 Card Package

4.1.1 Design and Format

Dimensions microSD Memory Card package	11mm x 15mm; (min. 10.9mm x 14.9mm; max. 11.1mm x 15.1mm) other dimensions see Figure 4-1 testing according to MIL STD 883, Method 2016
Thickness	'Inter Connect Area': 0.7mm+/-0.1mm see Figure 4-1 (C1) 'Card Thickness': 0.95mm Max see Figure 4-1 (C1 + C3) 'Pull Area': 1.0mm +/-0.1mm see Figure 4-1 (C)
Printable area	'Suggested Outside Keep out Area' : see Figure 4-3
Surface	plain (except contact area)
Edges	smooth edges
Inverse insertion	protection on right corner (top view)
Position of ESC contacts	along middle of shorter edge

Table 4-1 : microSD - Dimensions Summary

4.1.2. Reliability and durability

Temperature	Operation : -25°C / 85°C Storage : -40°C (168h) / 85°C (500h)
Moisture and corrosion	Operation : 25°C / 95% rel. humidity Storage : 40°C / 93% rel. hum./500h Salt water spray: 3% NacCl/35°C; 24h acc. MIL STD Method 1009
Durability	10.000 mating cycles
Bending	10N
Torque	0.10N.m, +/-2.5°max
Drop test	1.5m free fall
UV light exposure	UV: 254nm, 15Ws/cm² according to ISO 7816-1
Visual inspection/ Shape and form	no mold skin; complete form; no cavities surface smoothness <= -0.1 mm/cm^2 within contour; no cracks; no pollution (fat, oil dust, etc.)

Table 4-2 : Reliability and Durability

4.1.3 Electrical Static Discharge (ESD) requirement

ESD testing should be conducted according to IEC 61000-4-2
Required ESD parameters are:
(1) Human body model +-4 KV 100pf / 1.5K ohm
(2) Machine model +- 0.25 KV 200pf / 0 ohm

Contact Pads:
+/-4kV , Human body model according to IEC 61000-4-2

Non Contact Pads area:
+/-8kV (coupling plane discharge)
+/-15kV (air discharge)
Human body model according to IEC61000-4-2
The SDA's recommended test methods for the non-contact/air discharge tests are given in a seperate Application Note document.

4.1.4 External Signal contacts

Number of ESC	8 minimum
Distance from front edge	1.1 mm
ESC grid	1.1 mm
Contact dimensions	0.8 mm x 2.9 mm
Electrical resistance	30m-ohm (worst case: 100m-ohm)

Table 4-3 : microSD Package - External Signal Contacts

Contact discontinuity /micro-interrupt in accordance with application notes of SD Memory Card Specification Part 1 Physical Layer Specification Version 1.01

4.2 Mechanical Form Factor

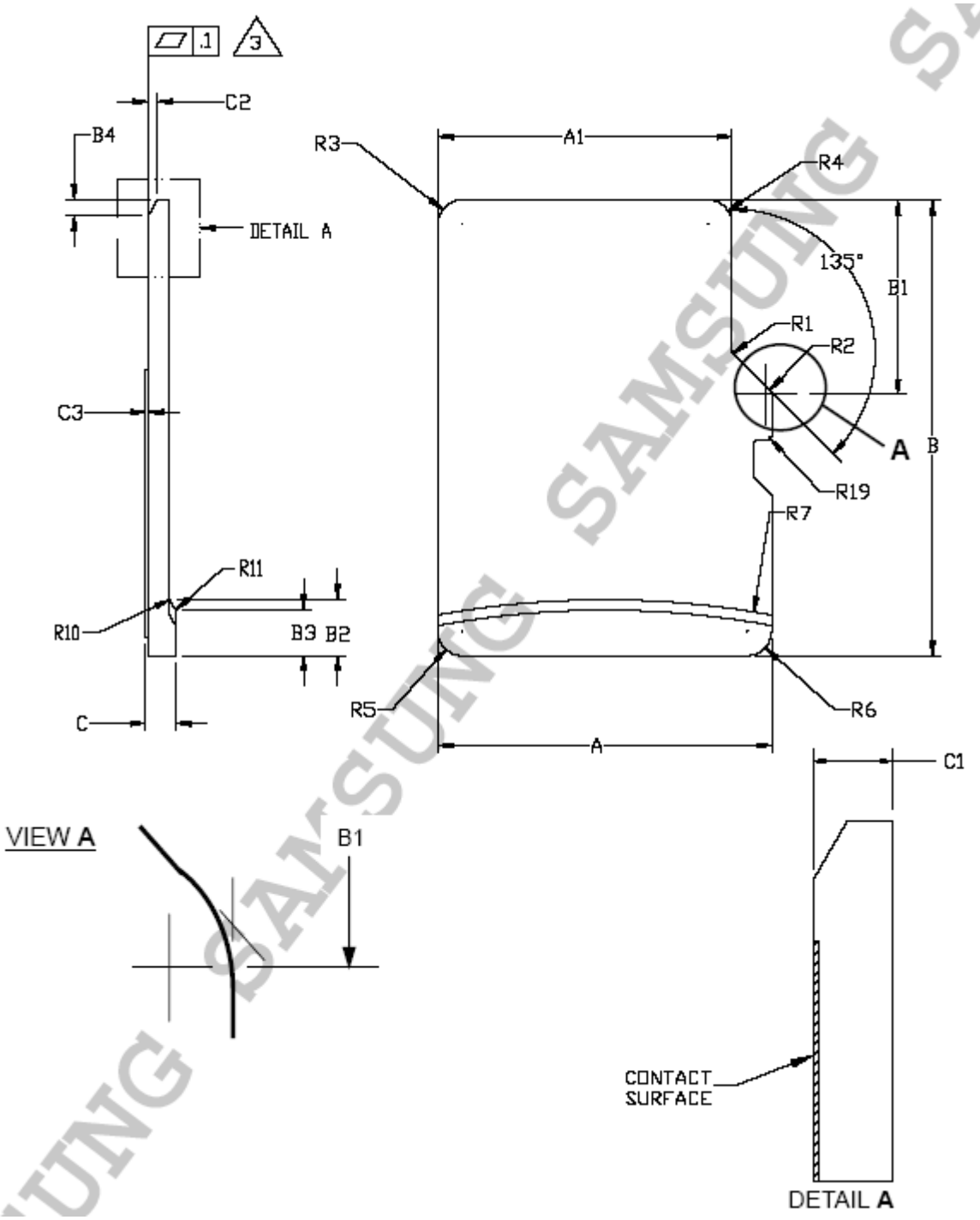


Figure 4-1 : Mechanical Description: Top View

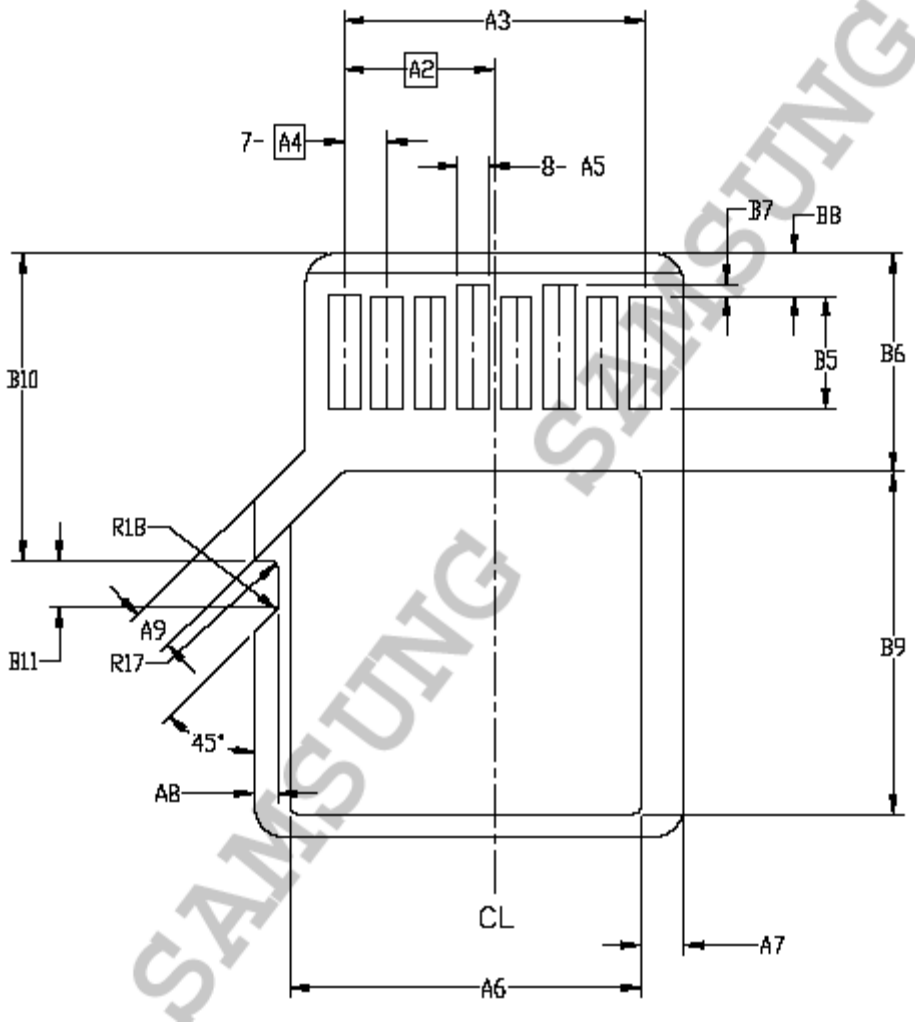


Figure 4-2 : Mechanical Description - Bottom View

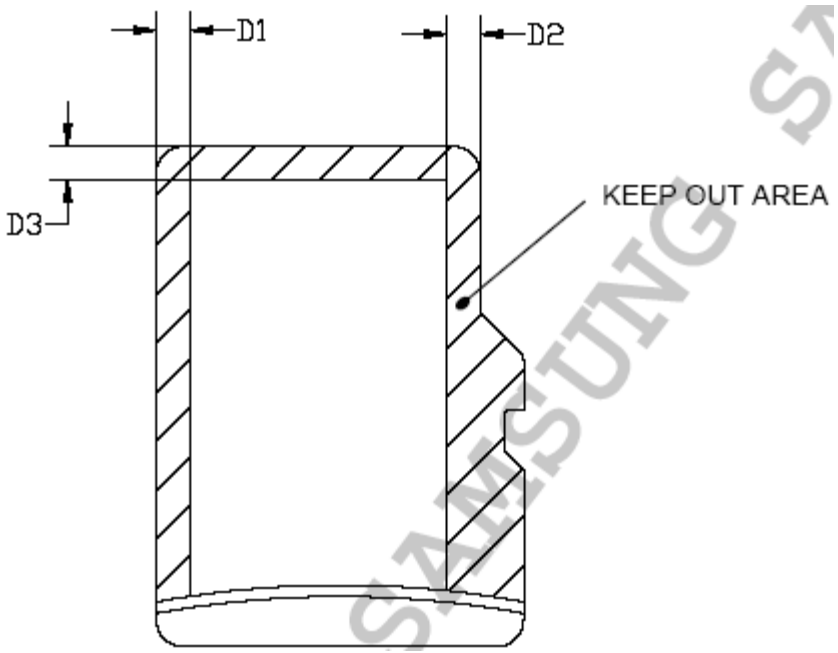


Figure 4-3 : Mechanical Description: Keep Out Area

	COMMON DIMENSIONS			
SYMBOL	MIN	NOM	MAX	NOTE
A	10.90	11.00	11.10	
A1	9.60	9.70	9.80	
A2	-	3.85	-	BASIC
A3	7.60	7.70	7.80	
A4	-	1.10	-	BASIC
A5	0.75	0.80	0.85	
A6	-	-	8.50	
A7	0.90	-	-	
A8	0.60	0.70	0.80	
A9	0.80	-	-	
B	14.90	15.00	15.10	
B1	6.30	6.40	6.50	
B2	1.64	1.84	2.04	
B3	1.30	1.50	1.70	
B4	0.42	0.52	0.62	
B5	2.80	2.90	3.00	
B6	5.50	-	-	
B7	0.20	0.30	0.40	
B8	1.00	1.10	1.20	
B9	-	-	9.00	
B10	7.80	7.90	8.00	
B11	1.10	1.20	1.30	
C	0.90	1.00	1.10	
C1	0.60	0.70	0.80	
C2	0.20	0.30	0.40	
C3	0.00	-	0.15	
D1	1.00	-	-	
D2	1.00	-	-	
D3	1.00	-	-	
R1	0.20	0.40	0.60	
R2	0.20	0.40	0.60	
R3	0.70	0.80	0.90	
R4	0.70	0.80	0.90	
R5	0.70	0.80	0.90	
R6	0.70	0.80	0.90	
R7	29.50	30.00	30.50	
R10	-	0.20	-	
R11	-	0.20	-	
R17	0.10	0.20	0.30	
R18	0.20	0.40	0.60	
R19	0.05	-	0.20	

NOTES

1. DIMENSIONING AND TOLERTANCING PER ASME Y14.5M-1994.
2. DIMENSIONS ARE IN MILLIMETERS.
3. COPLANARITY IS ADDITIVE TO C1 MAX THICKNESS.

Table 4-4 : microSD Package: Dimensions

5. Interface Description

5.1 Pin Assignment

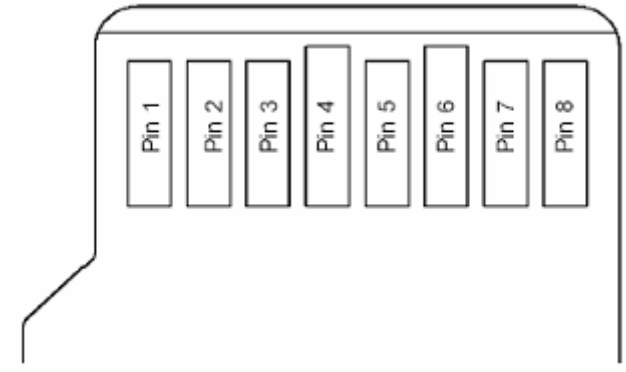


Figure 5-1 : Contact Area

Pin No.	Name	Type[1]	Description	Name	Type	Description
SD Mode				**SPI Mode**		
1	DAT2	I/O/PP	Data Line [Bit 2]	RSV		Reserved
2	CD/DAT3[2]	I/O/PP[3]	Card Detect / Data Line [Bit 3]	CS	I[3]	Chip Select (neg true)
3	CMD	PP	Command/Response	DI	I	Data In
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS	S	Supply voltage ground	VSS	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1	I/O/PP	Data Line [Bit 1]	RSV		Reserved

Table 5-1 : microSD Contact Pad Assignment

1) S: power supply; I: input; O: output using push-pull drivers; PP: I/O using push-pull drivers;
2) The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after SET_BUS_WIDTH command. The host shall keep its own DAT1-DAT3 lines in input mode, as well, while they are not used. It is defined so, in order to keep compatibility to MultiMediaCards.
3) After power up this line is input with 50K Ohm pull-up (can be used for card detection or SPI mode selection). The pull-up should be disconnected by the user, during regular data transfer, with SET_CLR _CARD_DETECT (ACMD42) command.

5.2 SD(microSD) Bus Topology

The SD(microSD) bus includes the following signals:

- **CMD**: Bidirectional Command/Resoponse signal
- **DAT0 - DAT3**: 4 Bidirectional data signals
- **CLK**: Host to card clock signal
- **V_{DD}**, **V_{SS}** : Power and ground signals

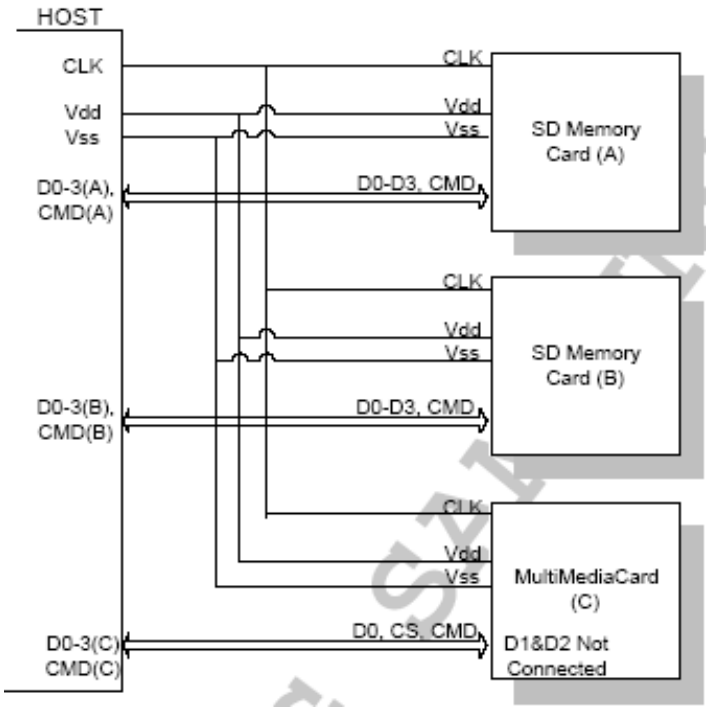


Figure 5-2 : SD Memory Card System Bus Topology

The SD(microSD) Card bus has a single master (application), multiple slaves (cards), synchronous start topology (refer to Fiugure 5-2). Clock, power and ground signals are common to all cards. Command (CMD) and data (DAT0-DAT3) signals are dedicated to each card providing continues point to point connection to all the cards.

During the initalization process, commands are sent to each card individualy, allowing the application to detect the cards and assign logical addresses to the physical slots. Data is always sent (received) to (from) each card individually. However, in order to simplify the handling of the card stack, after initialization process, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.

SD(microSD) Bus allows dynamic configuration of the number of data lines. After power-up, be defalut, the microSD Card will use only DAT0 for data transfer. After initialization, the host can change the bus width(number of active data lines). This feature allows and easy trade off between hardware cost and system performance. Note that while DAT1-DAT3 are not in use, the related Host's DAT lines should be in tri-state (input mode). For SDIO cards DAT1 and DAT2 are used for signaling.

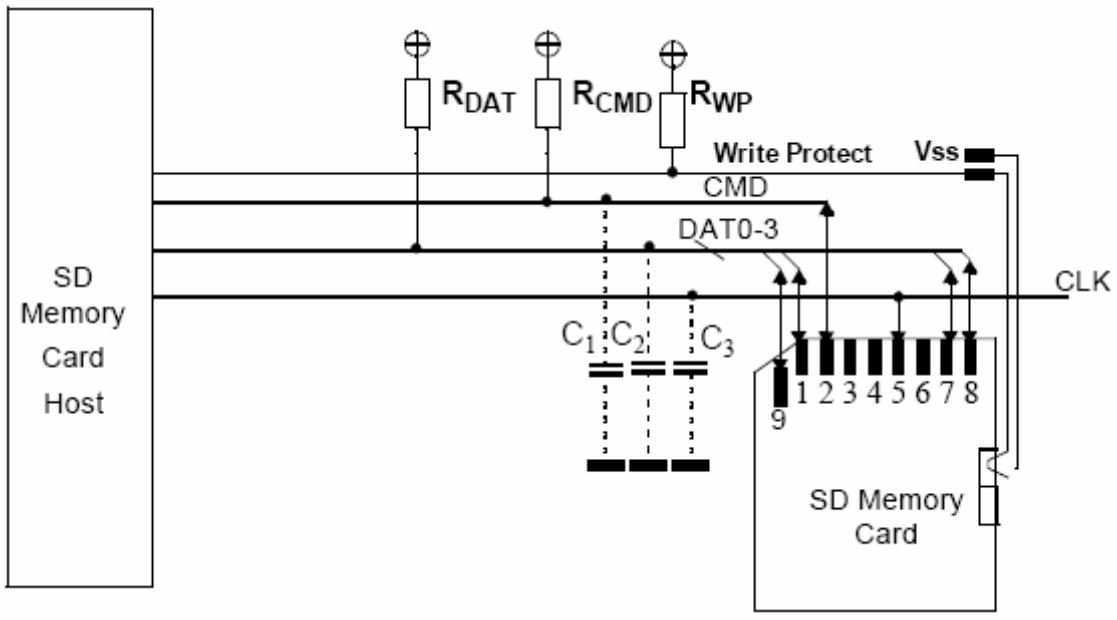


Figure 5-3 : Bus Circuitry Diagram

R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and the DAT line against bus floating when no card is inserted or when all card drivers are in a hi-impedance mode.
The host shall pull-up all DAT0-3 lines by R_{DAT} , even if the host uses microSD card as 1 bit mode only in SD mode. Also the host shall pull-up all "RSV" lines in SPI mode, even though they are not used.

5.2.1 Hot Insertion and Removal

Hot insertion and removal are allowed: the Samsung microSD Card will not be damaged by inserting or removing it into the SD bus even when the power is up.
Data transfer operations are protected by CRC codes, therefore any bit changes induced by card insertion and removal can be detected by the SD bus master.
The inserted card must be properly reset also when CLK carries a clock frequency f_{PP}. Each card shall have power protection to prevent card (and host) damage. Data transfer failures induced by removal/insertion are detected by the bus master.

5.2.2 Power Protection

Cards shall be inserted/removed into/from the bus without damage. If one of the supply pins (V_{DD} or V_{SS}) is not connected properly, then the current is drawn through a data line to supply the card. Every card's output also shall be able to withstand shortcuts to either supply. If hot insertion feature is implemented in the host, than the host has to withstand a shortcut between V_{DD} and V_{SS} without damage.

5.3 SPI Bus Topology

The SPI compatible communication mode of the microSD Memory Card is designed to communicate with a SPI channel, commonly found in various microcontrollers in the market. The interface is selected during the first reset command after power up and cannot be changed as long as the part is powered on.
The SPI standard defines the physical link only, and not complete data transfer protocol. The microSD Card SPI implementation uses the same command set of the SD mode. From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum. The disadvantage is the loss of performance, relatively to the SD mode which enables the wide bus option.

The microSD Card SPI interface is compatible with SPI hosts available on the market. As any other SPI device the microSD Card SPI channel consists the following four signals:

CS: Host to card Chip Select signal
CLK: Host to card clock signal
DataIn: Host to card data signal
DataOut: Card to host data signal

Another SPI common characteristic are byte transfers, which is implemented in the card as well. All data tokens are multiples of bytes (8 bit) and always byte aligned to the CS signal.

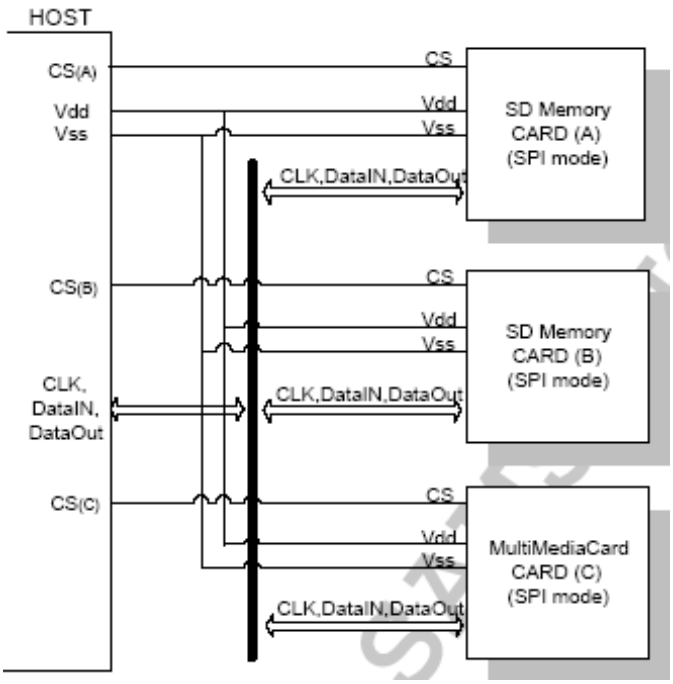


Figure 5-4 : SD Memory Card system (SPI mode) Bus Topology

The card identification and addressing methods are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal (see Figure 5-4).

The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception occurs during card programming, when the host can de-assert the CS signal without affecting the programming process.

5.3.1 Power Protection

Same as SD(microSD) Card mode

5.4 Electrical Interface

The following sections provide valuable information about the electrical interface.

5.4.1 Power Up

The power-up of the SD(microSD) Card bus is handled locally in each microSD Card and the bus master. See Figure 5-5

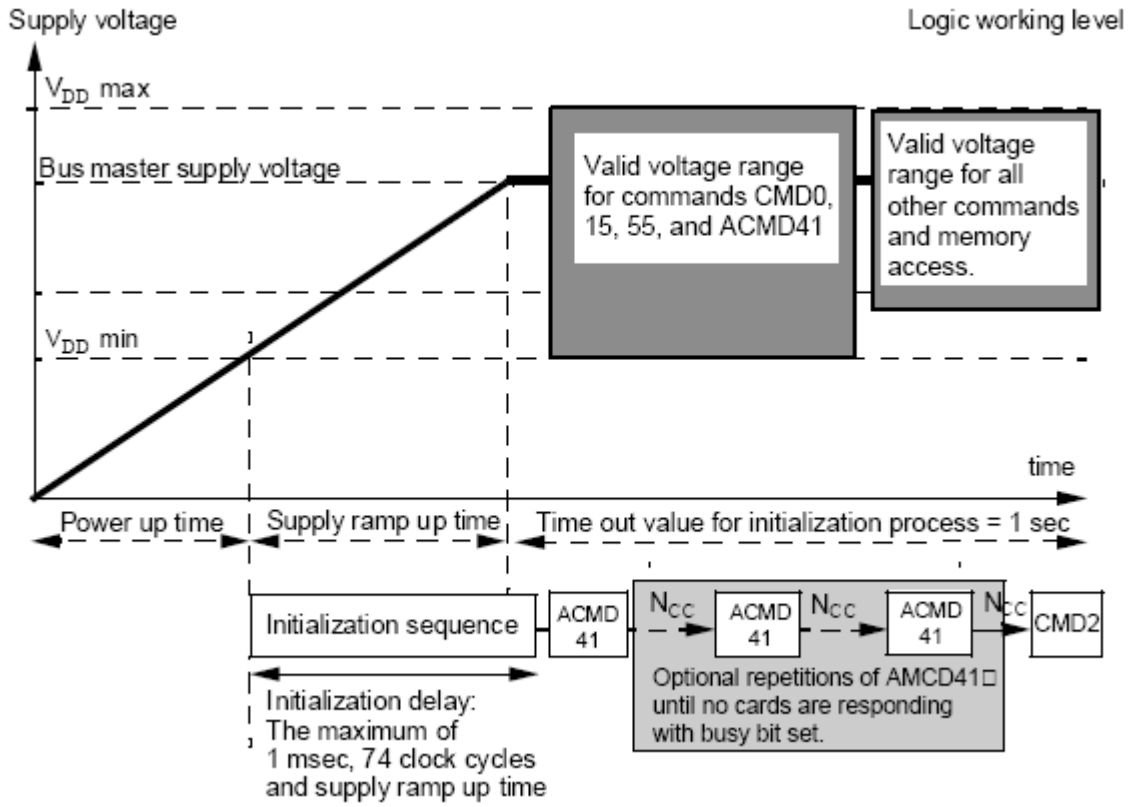


Figure 5-5 : Power-up Diagram

- After power up (including hot insertion, i.e. inserting a card when the bus is operating) the microSD card enters the *idle state*. During this state the microSD Card ignores all bus transactions until ACMD41 is received. (ACMD command type shall always precede with CMD55).
- ACMD41 is a special synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. Besides the operation voltage profile of the cards, the response ACMD41 contains a busy flag, indicating that the card is still working on its power-up procedure and is not ready for identification. This bit informs the host that the card is not ready. The host has to wait (and continue to poll the cards, each one on his turn) until this bit is cleared. The maximum period of power up procedure of single card shall not exceed 1 second.
- Getting individual cards, as well as the whole microSD Card system, out of *idle* is up to the responsibility of the bus master. Since the power up time and the supply ramp up time depend on application parameters such as the maximum number of microSD Cards, the bus length and the power supply unit, the host must ensure that the power is built up to the operating level (the same level which will be specified in ACMD41) before ACMD41 is transmitted.
- After power up the host starts the clock and sends the initializing sequence on the CMD line. This sequence is a contiguous stream of logical '1's. The sequence length is the longest of 1msec, 74 clocks or the supply-ramp-up-time; The additional 10 clocks (over the 64 clocks after what the card should be ready for communication) is provided to eliminate power-up synchronization problems.
- Every bus master shall have the capability to implement ACMD41 and CMD1. The CMD1 will be used to ask MultiMedi

aCards to sent their Operation Conditions. In any case the ACMD41 or the CMD1 shall be send separately to each card accessing it through its own CMD line.

5.4.2 Bus Operating Conditions

SPI Mode bus operating conditions are identical to SD Card mode bus operating conditions, Table 5-2 lists the power supply voltage. The CS(Chip Select) signal timing is identical to the input signal timing (see Figure 5-7)

General

Parameter	Symbol	Min	Max.	Unit	Remark
Peak voltage on all lines		-0.3	V_{DD}+0.3	V	
All Inputs					
Input Leakage Current		-10	10	uA	
All Outputs					
Output Leakage Current		-10	10	uA	

Power Supply Voltage

Parameter	Symbol	Min	Max.	Unit	Remark
Supply voltage	V_{DD}	2.0	3.6	V	CMD0, 15, 55, ACMD41 commands
Supply voltage		2.7	3.6	V	Except CMD0, 15, 55 ACMD41 commands
Supply voltage differentials (V_{SS1}, V_{SS2})		-0.3	0.3	V	
Power up Time			250	ms	From 0V to V_{DD} Min.

Table 5-2 : Power Supply Voltage

5.4.3 Bus Signal Line Load

The total capacitance C_L of each line of the microSD Card bus is the sum of the bus master capacitance C_{HOST}, the bus capacitance C_{BUS} itself and the capacitance C_{CARD} of the card connected to this line:

$C_L = C_{HOST} + C_{BUS} + N*C_{CARD}$

When N is the number of connected cards. Requiring the sum of the host and bus capacitances not exceed 30pF for up to 10 cards, and 40pF for up to 30cards, the following values in Table 5-3 must not be exceeded.

Parameter	Symbol	Min	Max.	Unit	Remark
Pull-up resistance for CMD	R_{CMD} R_{DAT}	10	100	KOhm	to prevent bus floating
Bus signal line capacitance	C_L		250	pF	f_{PP} <= 5 MHz, 21 cards
Bus signal line capacitance	C_L		100	pF	f_{PP} <= 20 MHz, 7 cards
Single card capacitance	C_{CARD}		10	pF	
Maximum signal line inductance			16	nH	f_{PP} <= 20 MHz
Pull-up resistance inside card (pin1)	R_{DAT3}	10	90	KOhm	May be used for detection

Table 5-3 : Signal Line's Load

5.4.4 Bus Signal Levels

As the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage.

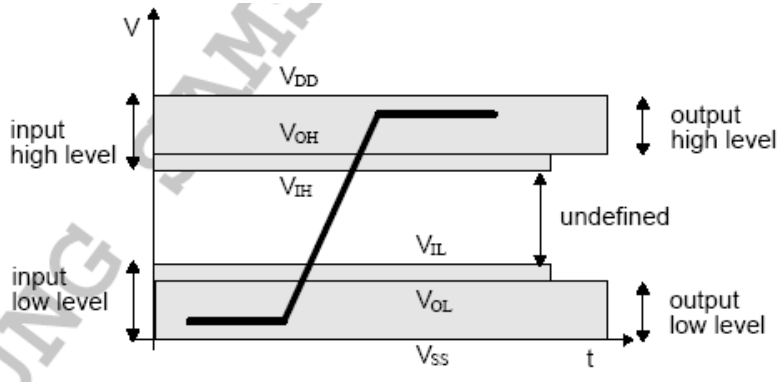


Figure 5-6 : Bus Signal Levels

To meet the requirements of the JEDEC specification JESD8-1A, the card input and output voltages shall be within the specified ranges in Tabel 5-4 for any V_{DD} of the allowed voltage range:

Parameter	Symbol	Min	Max.	Unit	Conditions
Output HIGH voltage	V_{OH}	0.75*V_{DD}		V	I_{OH}=-100 μA @V_{DD} min
Output LOW voltage	V_{OL}		0.125*V_{DD}	V	I_{OL}=100 μA @V_{DD} min
Input HIGH voltage	V_{IH}	0.625*V_{DD}	V_{DD} + 0.3	V	
Input LOW voltage	V_{IL}	VSS-0.3	0.25*V_{DD}	V	

Table 5-4: Bus Signal Cond. - I/O Signal Voltages

5.4.5 Bus Timing (Default)

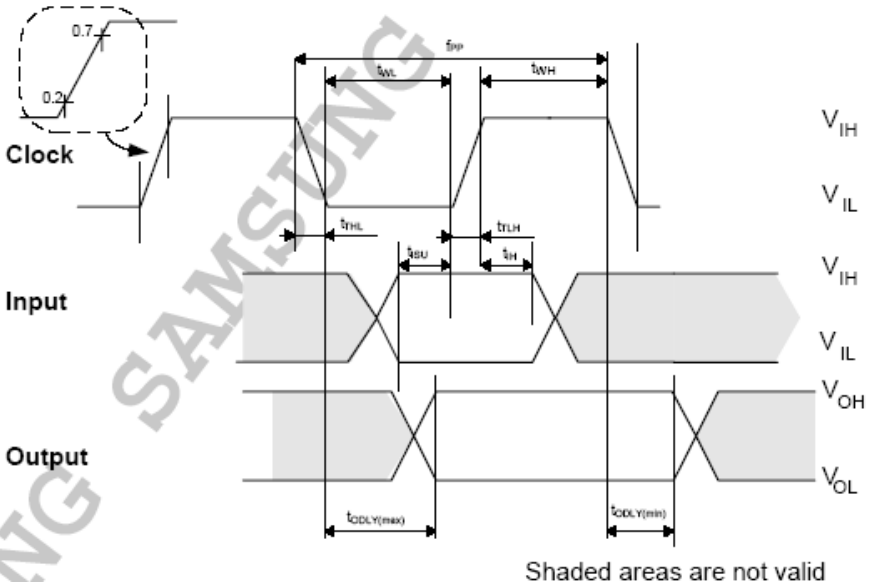


Figure 5-7 : Timing diagram data input/outpout referenced to clock (Default)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min. (V_{IH}) and max. (V_{IL})					
Clock frequency Data Transfer Mode	f_{PP}	0	25	MHz	C_L <= 100 pF (7 cards)
Clock frequency Identification Mode (The low frequency is required for MultiMediaCard compatibilty)	f_{OD}	$0_{(1)}$ / 100kHz	400	kHz	C_L <= 250 pF (21 cards)
Clock low time	t_{WL}	10		ns	C_L <= 100 pF (7 cards)
Clock high time	t_{WH}	10		ns	C_L <= 100 pF (7 cards)
Clock rise time	t_{TLH}		10	ns	C_L <= 100 pF (10 cards)
Clock fall time	t_{THL}		10	ns	C_L <= 100 pF (7 cards)
Clock low time	t_{WL}	50		ns	C_L <= 250 pF (21 cards)
Clock high time	t_{WH}	50		ns	C_L <= 250 pF (21 cards)
Clock rise time	t_{TLH}		50	ns	C_L <= 250 pF (21 cards)
Clock fall time	t_{THL}		50	ns	C_L <= 250 pF (21 cards)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	5		ns	C_L <= 25 pF (1 card)
Input hold time	t_{IH}	5		ns	C_L <= 25 pF (1 card)
Outputs CMD, DAT (referenced to CLK)					
Output delay time during Data Transfer Mode	t_{ODLY}	0	14	ns	C_L <= 25 pF (1 card)
Output delay time during Identification Mode	t_{ODLY}	0	50	ns	C_L <= 25 pF (1 card)

(1) OHz means stops the clock. The given minimum frequency range is for cases where a continuous clock is required

Table 5-5 : Bus Timing - Parameters Values (Default)

5.4.6 Bus Timing (High-speed Mode)

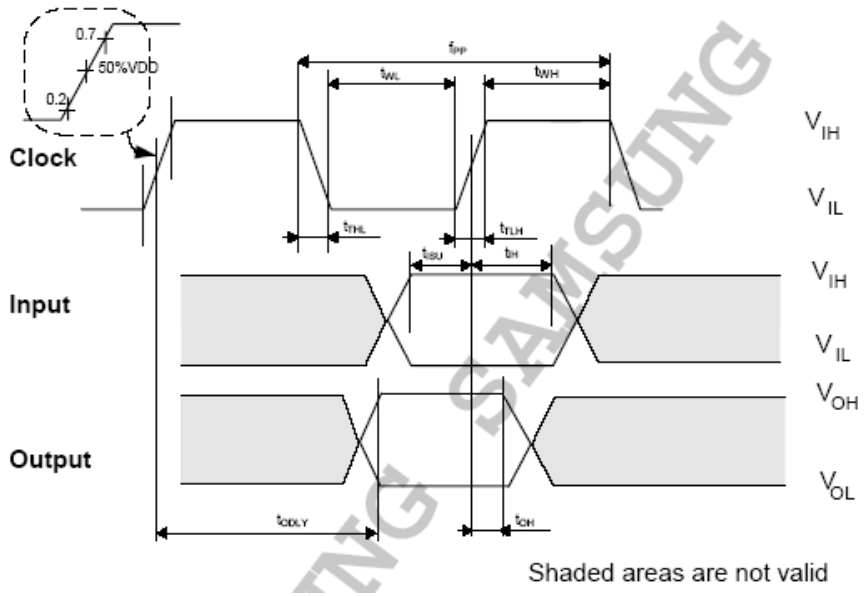


Figure 5-8 : Timing diagram data input/output referenced to clock (High-Speed)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min. (V_{IH}) and max. (V_{IL})					
Clock frequency Data Transfer Mode	f_{PP}	0	50	MHz	
Clock low time	t_{WL}	7		ns	
Clock high time	t_{WH}	7		ns	
Clock rise time	t_{TLH}		3	ns	
Clock fall time	t_{THL}		3	ns	
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	6		ns	
Input hold time	t_{IH}	2		ns	
Outputs CMD, DAT (referenced to CLK)					
Output delay time during Data Transfer Mode	t_{ODLY}	0	14	ns	
Output Hold time	t_{OH}	2.5		ns	
Total Systme capacitance for each line[1]	C_L		40	pF	

Table 5-6 : Bus Timing - Parameters Values (High-Speed)

1) In order to satisfy severe timing, host shall drive only one card.

5.5 microSD Card Registers

Within the card interface six registers are defined: OCR, CID, CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands (see Chapter 6.8). The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters.

5.5.1 OCR Register

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. In addition, this register includes a status information bit. This status bit is set if the card power up procedure has been finished. The OCR register shall be implemented by all cards which do not support the full operating range of the SD Memory Card bus, or if the card power up extends the definition in the timing diagram. (Figure 5-5)

OCR bit	VDD Voltage Window	OCR Value
0-3	reserved	0
4	reserved	0
5	reserved	0
6	reserved	0
7	reserved	0
8	2.0 - 2.1	0
9	2.1 - 2.2	0
10	2.2 - 2.3	0
11	2.3 - 2.4	0
12	2.4 - 2.5	0
13	2.5 - 2.6	0
14	2.6 - 2.7	0
15	2.7 - 2.8	1
16	2.8 - 2.9	1
17	2.9 - 3.0	1
18	3.0 - 3.1	1
19	3.1 - 3.2	1
20	3.2 - 3.3	1
21	3.3 - 3.4	1
22	3.4 - 3.5	1
23	3.5 - 3.6	1
24 - 30	Reserved	0
31	Card power up status bit(busy)[1]	-

1) This bit is set to LOW if the card has not finished power up routine

Table 5-7 : OCR Register Definition

The supported voltage range is coded as shown in Table 5-7. A voltage range is not supported if the corresponding bit value is set to LOW. As long as the card is busy, the corresponding bit (31) is set to LOW

5.5.2 CID Register

The Card IDentification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase. Every individual flash shall have an unique identification number. The structure of the CID register is defined in the following paragraphs:

Name	Type	Width	CID-slice	Comments	CID Value
Manufacturer ID (MID)	Binary	8	[127:120]	The MID number is controlled and allocated by the SD Group	0x1B
OEM/Application ID (OID)	ASCII	16	[119:104]	Identifies the card OEM and/ or the card contents. The OID is allocated by the SD Group	0x534D
Product name (PNM)	ASCII	40	[103:64]	5 ASCII characters long	---
Product revision (PRV)	BCD	8	[63:56]	Two binary coded decimal digits	---
Product serial number (PSN)	Binary	32	[55:24]	32bits of binary number	---
Reserved		4	[23:20]		
Manufacturing date (MDT)	BCD	12	[19:8]	Manufacture date-yym(offset from 2000)	Manufacture date (ex. March 2001=00110100)
CRC7 checksum	Binary	7	[7:1]	Checksum of the CID contents	---
not used, always ’1’		1	[0:0]		---

Table 5-8 : CID Register Fields

5.5.3 CSD Register

The Card-Specific Data register provides information on how to access the card contents. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used etc. The programmable part of the register (entries marked by W or E, see below) can be changed by CMD27. The type of the entries in the table below is coded as follows: R = readable, W(1) = writable once, W=multiple writable

Name	Field	Width	Cell Type	CSD-slice	CSD Value
CSD structure	CSD_STRUCTURE	2	R	[127:126]	ver.1.0
Reserved	-	6	R	[125:120]	-
Data read access-time 1	TAAC	8	R	[119:112]	5ms
Data read access-time 2 in CLK cycles (NSAC*100)	NSAC	8	R	[111:104]	0
Max. Data transfer rate	TRAN_SPEED	8	R	[103:96]	25Mbit/sec
Card command classes	CCC	12	R	[95:84]	(*1)
Max. read data block length	READ_BL_LEN	4	R	[83:80]	512Bytes
Partial blocks for read allowed	READ_BL_PARTIAL	1	R	[79:79]	Support

Table 5-9 : CSD Register Fields

.

Name	Field	Width	Cell Type	CSD-slice
Write block misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]
Read block misalignment	READ_BLK_MISALIGN	1	R	[77:77]
DSR implemented	DSR_IMP	1	R	[76:76]
Reserved	-	2	R	[75:74]
Device size	C_SIZE	12	R	[73:62]
Max. read current @ V_{DD} min	VDD_R_CURR_MIN	3	R	[61:59]
Max. read current @ V_{DD} max	VDD_R_CURR_MAX	3	R	[58:56]
Max. write current @ V_{DD} min	VDD_W_CURR_MIN	3	R	[55:53]
Max. write current @ V_{DD} max	VDD_W_CURR_MAX	3	R	[52:50]
Device size multiplier	C_SIZE_MULT	3	R	[49:47]
Erase single block enable	ERASE_BLK_EN	1	R	[46:46]
Erase sector size	SECTOR_SIZE	7	R	[45:39]
Write protect group size	WP_GRP_SIZE	7	R	[38:32]
Write protect group enable	WP_GRP_ENABLE	1	R	[31:31]
Reserved for MulitMediaCard compatibility		2	R	[30:29]
Write speed factor	R2W_FACTOR	3	R	[28:26]
Max. write data block length	WRITE_BL_LEN	4	R	[25:22]
Partial blocks for write allowed	WRITE_BL_PARTIAL	1	R	[21:21]
Reserved	-	5	R	[20:16]
File format group	FILE_FORMAT_GRP	1	R/W(1)	[15:15]
Copy flag (OTP)	COPY	1	R/W(1)	[14:14]
Permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]
Temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]
File format	FILE_FORMAT	2	R/W(1)	[11:10]
Reserved		2	R/W	[9:8]
CRC	CRC	7	R/W	[7:1]
Not used, always ’1’	-	1	-	[0:0]

Table 5-10 : CSD Register Fields

(*1) Support command class 0,2,4,5,6,7,8,10 Include: Basic, Block read/write, Erase, Write protection, Lock card and application command. Not support 1,3 Include: Stream read/write

The following sections describe the CSD fields and the relevant data types. If not explicitly defined otherwise, all bit strings are interpreted as binary coded numbers starting with the left bit first.

- **CSD_STRUCTURE**

Version number of the related CSD structure.

CSD_STRUCTURE	CSD structure version	Valid for System Specification Version
0	CSD version No. 1.0	Version 1.0 - 1.01
1 - 3	Reserved	

Table 5-11 : CSD Register Structure

- **TAAC**

Defines the asynchronous part of the data access time.

TAAC bit position	code
2:0	Time unit 0=1ns, 1=10ns, 2=100ns, 3=1µs, 4=10µs, 5=100µs, 6=1ms, 7=10ms
6:3	Time value 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
7	Reserved

Table 5-12 : TAAC Access Time Definition

- **NSAC**

Defines the worst case for the clock dependent factor of the data access time. The unit for NSAC is 100 clock cycles. Therefore, the maximal value for the clock dependent part of the data access time is 25.5k clock cycles.
The total access time N_{AC} as expressed in Table 6-29 is the sum of TAAC and NSAC. It has to be computed by the host for the actual clock rate. The read access time should be interpreted as a typical delay for the first data bit of a data block from the end bit on the read commands.

- **TRAN_SPEED**

The following table defines the maximum data transfer rate per one data line TRAN_SPEED.

TRAN_SPEED bit	Code
2:0	Transfer rate unit 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4...7=reserved
6:3	Time value 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.6, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.2, C=5.5, D=6.0, E=7.0, F=8.0
7	Reserved

Table 5-13 : Maximum Data Transfer Rate Definition

Note that for the current microSD cards that field must be always 0_0110_010b (032h) which is equal to 25MHz - the mandatory maximum operating frequency of microSD card.
In High speed mode, that field must be always 0_1011_010b (05Ah) which is equal to 50MHz. And when the timing mode returns to the default by CMD6 or CMD0 command, its value will be 032h.

- **CCC**

The microSD Card command set is divided into subsets (command classes). The card command class register CCC defines which command classes are supported by this card. A value of '1' in a CCC bit means that the corresponding command class is supported. For command class definition refer to Table 6-12.

CCC bit	Supported card command class
0	class 0
1	class 1
.....	
11	class 11

Table 5-14 : Supported Card Command Classes

- **READ_BL_LEN**

The maximum read data block length is computed as $2^{READ\_BL\_LEN}$. The maximum block length might therefore be in the range 512...2048 bytes (see Chapter 6.12 for details).Note that in the microSD Card, the WRITE_BL_LEN is always equal to READ_BL_LEN

READ_BL_LEN	Block length
0-8	Reserved
9	2^9 = 512 Bytes
.....	
11	2^{11} = 2048 Bytes
12-15	Reserved

Table 5-15 : Data Block Length

- **READ_BL_PARTIAL**

READ_BL_PARTIAL is always set to 1 in the microSD Card. Partial BLock Read is always allowed in the microSD Card. It means that smaller blocks can be used as well. The minimum block size is one byte.

- **WRITE_BLK_MISALIGN**

Defines if the data block to be written by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in WRITE_BL_LEN.
WRITE_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.
WRITE_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

- **READ_BLK_MISALIGN**

Defines if the data block to be read by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in READ_BL_LEN.
READ_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.
READ_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

- **DSR_IMP**

Defines if the configurable driver stage is integrated on the card. If set, a driver stage register (DSR) must be implemented also.

DSR_IMP	DSR type
0	DSR is not implemented
1	DSR implemented

Table 5-16 : DSR Implementation code Table

- **C_SIZE**

This parameter is used to compute the card capacity (not include the security protected area). The memory capacity of the card is computed from the entries C_SIZE, C_SIZE_MULT and READ_BL_LEN as follows:

memory capacity = BLOCKNR * BLOCK_LEN

where

BLOCKNR = (C_SIZE+1) * MULT

MULT = $2^{C\_SIZE\_MULT+2}$ (C_SIZE_MULT < 8)

BLOCK_LEN = $2^{READ\_BL\_LEN}$, (READ_BL_LE< 12)

Maximum capacity of the card, compliant to SD Physical Specification Version 1.01 shall be up to 2GB bytes (2^{31}bytes) to be consistent with the maximum capacity (2G bytes) of SD Memory Card File System Specification Ver.1.01.
To indicate 2GByte card, BLOCK_LEN shall be 1024 bytes.
Therefore, the maximal capacity which can be coded is 4096*512*1024 = 2G bytes.
Example: A 32Mbyte card with BLOCK_LEN = 512 can be coded by C_SIZE_MULT = 3 and C_SIZE = 2000.

- **VDD_R_CURR_MIN, VDD_W_CURR_MIN**

The maximum values for read and write currents at the minimal power supply V_{DD} are coded as follows:

VDD_R_CURR_MIN **VDD_W_CURR_MIN**	Code for current consumption @ V_{DD}
2:0	0=0.5mA; 1=1mA; 2=5mA; 3=10mA; 4=25mA; 5=35mA; 6=60mA; 7=100mA

Table 5-17 : $V_{DD,min}$ Current Consumption

- **VDD_R_CURR_MAX, VDD_W_CURR_MAX**

The maximum values for read and write currents at the maximal power supply V_{DD} are coded as follows:

VDD_R_CURR_MAX VDD_W_CURR_MAX	Code for current consumption @ V_{DD}
2:0	0=1mA; 1=5mA; 2=10mA; 3=25mA; 4=35mA; 5=45mA; 6=80mA; 7=200mA

Table 5-18 : $V_{DD,max}$ Current Consumption

- **C_SIZE_MULT**

This parameter is used for coding a factor MULT for computing the total device size (see 'C_SIZE'). The factor MULT is defined as $2^{C\_SIZE\_MULT+2}$.

C_SIZE_MULT	MULT
0	$2^2 = 4$
1	$2^3 = 8$
2	$2^4 = 16$
3	$2^5 = 32$
4	$2^6 = 64$
5	$2^7 = 128$
6	$2^8 = 256$
7	$2^9 = 512$

Table 5-19 : Multiply Factor For The Device Size

- **ERASE_BLK_EN**

The ERASE_BLK_EN defines the granularity of the unit size of the data to be erased. The erase operation can erase either one or multiple units of WRITE_BL_LEN or one or multiple units (or sectors) of SECTOR_SIZE (see definition below).

If ERASE_BLK_EN = '0', the host can erase one or multiple units of SECTOR_SIZE. The erase will start from the beginning of the sector that contains the start address to the end of the sector that contains the end address. For example, if SECTOR_SIZE=31 and the host sets the Erase Start Address to 5 and the Erase End Address to 40, the physical blocks from 0 to 63 will be erased as shown in Figure 5-9.

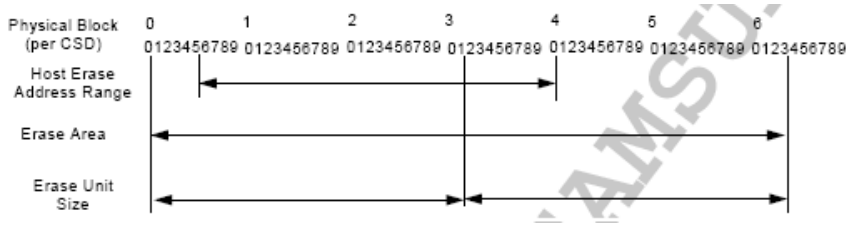


Figure 5-9 : ERASE_BLK_EN = '0' example

If ERASE_BLK_EN = '1' the host can erase one or multiple units of 512 bytes. All blocks that contain data from start address to end address are erased. For example, if the host sets the Erase Start Address to 5 and the Erase End Address to 40, the physical blocks from 5 to 40 will be erased as shown in Figure 5-10.

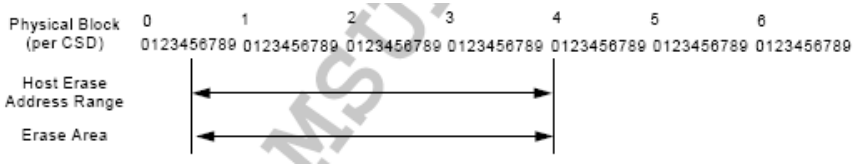


Figure 5-10 : ERASE_BLK_EN = '1' example

- **SECTOR_SIZE**

The size of an erasable sector. The contents of this register is a 7 bit binary coded value, defining the number of write blocks (see WRITE_BL_LEN). The actual size is computed by increasing this number by one. A value of zero means 1 write block, 127 means 128 write blocks.

- **WP_GRP_SIZE**

The size of a write protected group. The contents of this register is a 7 bit binary coded value, defining the number of erase sectors (see SECTOR_SIZE). The actual size is computed by increasing this number by one. A value of zero means 1 erase sector, 127 means 128 erase sectors.

- **WP_GRP_ENABLE**

A value of '0' means no group write protection possible.

- **R2W_FACTOR**

Defines the typical block program time as a multiple of the read access time. The following table defines the field format.

R2W_FACTOR	Multiples of read access time
0	1
1	2 (write half as fast as read)
2	4
3	8
4	16
5	32
6, 7	Reserved

Table 5-20 : R2W_FACTOR

- **WRITE_BL_LEN**

The maximum write data block length is computed as $2^{WRITE\_BL\_LEN}$. The maximum block length might therefore be in the range from 512 up to 2048 bytes. Write Blcok Length of 512bytes is always supported.

Note that in the microSD Card, the WRITE_BL_LEN is always equal to READ_BL_LEN.

READ_BL_LEN	Block length
0-8	Reserved
9	2^9 = 512 Bytes
.....	
11	2^{11} = 2048 Bytes
12-15	Reserved

Table 5-21 : Data Block Length

- **WRITE_BL_PARTIAL**

Defines whether partial block sizes can be used in block write commands.
WRITE_BL_PARTIAL='0' means that only the WRITE_BL_LEN block size and its partial derivatives, in resolution of units of 512bytes, can be used for block oriented data write.
WRITE_BL_PARTIAL='1' means that smaller blocks can be used as well. The minimum block size is one byte.

- **FILE_FORMAT_GRP**

Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in Table 5-20

- **COPY**

Defines if the contents is original (= '0') or has been copied ('1'). The COPY bit for OTP and MTP devices, sold to end consumers, is set to '1' which identifies the card contents as a copy. The COPY bit is an one time programmable bit.

- **PERM_WRITE_PROTECT**

Permanently protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled). The default value is '0', i.e. not permanently write protected.

- **TMP_WRITE_PROTECT**

Temporarily protects the whole card content from being overwritten or erased (all write and erase commands for this card are temporarily disabled). This bit can be set and reset. The default value is '0', i.e. not write protected.

- **FILE_FORMAT**

Indicates the file format on the card. This field is read-only for ROM. The following formats are defined:

FILE_FORMAT_GRP	FILE_FORMAT	Type
0	0	Hard disk-like file system with partition table
0	1	DOS FAT (floppy-like) with boot sector only (no partition table)
0	2	Universal File Format
0	3	Others / Unknown
1	0, 1, 2, 3	Reserved

Table 5-22 : File Formats

- **CRC**

The CRC field carries the check sum for the CSD contents. It is computed according to Chapter 6.6. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents.

	Command classes							
CSD Field	0	2	4	5	6	7	8	9
CSD_STRUCTURE	+	+	+	+	+	+	+	+
TAAC		+	+	+	+	+	+	
NSAC		+	+	+	+	+	+	
TRAN_SPEED		+	+					
CCC	+	+	+	+	+	+	+	+

	Command classes							
CSD Field	**0**	**2**	**4**	**5**	**6**	**7**	**8**	**9**
READ_BL_LEN		+						
WRITE_BLK_MISALIGN			+					
READ_BLK_MISALIGN		+						
DSR_IMP	+	+	+	+	+	+	+	+
C_SIZE_MANT		+	+	+	+	+	+	
C_SIZE_EXP		+	+	+	+	+	+	
VDD_R_CURR_MIN		+						
VDD_R_CURR_MAX		+						
VDD_W_CURR_MIN			+	+	+	+	+	
VDD_W_CURR_MAX			+	+	+	+	+	
ERASE_BLK_EN				+	+	+	+	
SECTOR_SIZE				+	+	+	+	
WP_GRP_SIZE					+	+	+	
WP_GRP_ENABLE					+	+	+	
R2W_FACTOR			+	+	+	+	+	
WRITE_BL_LEN			+	+	+	+	+	
WRITE_BL_PARTIAL			+	+	+	+	+	
FILE_FORMAT_GRP								
COPY	+	+	+	+	+	+	+	
PERM_WRITE_PROTECT	+	+	+	+	+	+	+	
TMP_WRITE_PROTECT	+	+	+	+	+	+	+	
FILE_FORMAT								
CRC	+	+	+	+	+	+	+	+

Table 5-23: Cross reference of CSD fields vs. command classes

5.5.4 RCA Register

The writable 16-bit relative card address register carries the card address that is published by the card during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The default value of the RCA register is 0x0000. The value 0x0000 is reserved to set all cards into the *Stand-by State* with CMD7.

5.5.5 SCR Register

In addition to the CSD register, there is the microSD Configuration Register (SCR). This register provides information on microSD Card's special features that were configured into the given card. The size of SCR register is 64bit. The register shall be set in the factory by the microSD Card manufacturer. The following table describes the SCR register content.

Name	Field	Width	Cell Type	SCR-slice	SCR Value
SCR structure	SCR_STRUCTURE	4	R	[63:60]	Version 1.0~1.10
SD Memory Card - Spec. Version	SD_SPEC	4	R	[59:56]	Version 1.10
data_status_after erases	DATA_STAT_AFTER _ERASE	1	R	[55:55]	Zero after erase
SD Security Support	SD_SECURITY	3	R	[54:52]	Secure protocol 2.0
DAT Bus widths supported	SD_BUS_WIDTHS	4	R	[51:48]	1bit & 4bit
Reserved	-	16	R	[47:32]	-
Reserved for manufacturer usage	-	32	R	[31:0]	-

Table 5-24 : The SCR Fields

- **SCR_STRUCTURE**

Version number of the related SCR structure in the microSD Card Physical Layer Specification.

SCR_STRUCTURE	SCR structure version	Valid for SD Physical Layer Sepcification Version
0	SCR version No. 1.0	Version 1.0-1.10
1-15	Reserved	

Table 5-25 : SCR Register Structure Version

- **SD_SPEC**

Describes the SD Memory Card Physical Layer Specfication version supported by this card

SPEC_VERS	Physical Layer Specification Version Number
0	Version 1.0-1.01
1	Version 1.10
2-15	Reserved

Table 5-26 : SD Memory Card Physical Layer Specification Version

- **DATA_STAT_AFTER_ERASE**

Defines the data status after erase, whether it is ’0’ or ’1’ (the status is card vendor dependent)

- **SD_SECURITY**

Describes the security algorithm supported by the card.

SD_SECURITY	Supported Algorithm
0	no security
1	Security protocol 1.0
2	Security protocol 2.0
3 .. 7	Reserved

Table 5-27 : SD Supported Security Algorithm

Security Protocol 1.0 relates to Security Specification Version 0.96.
Security Protocol 2.0 relates to Security Specification Version 1.0-1.01

Note that it is mandatory for a regular writable SD Memory Card to support Security Protocol.

- **SD_BUS_WIDTHS**

Describes all the DAT bus widths that are supported by this card.

SD_SECURITY	Supported Algorithm
Bit 0	1bit (DAT0)
Bit 1	Reserved
Bit 2	4 bit (DAT0-3)
Bit 3 [MSB]	Reserved

Table 5-28 : microSD Card Supported Bus Widths

Since SD Memory Card shall support at least two bus modes: 1bit or 4 bit width, then any microSD Card shall set at least bits0 and 2 (SD_BUS_WIDTH="0101")

6. microSD Card Protocol Description

6.1 SD Bus Protocol

Communication over the SD bus is based on command and data bit streams which are initiated by a start bit and terminated by a stop bit.

- **Command** : a command is a token which starts an operation. A command is sent from the host either to a single card (address command) or all connected cards(broadcst command). A command is transferred serially on the CMD line.
- **Response** : a response is token which is sent from an addressed card, or (syncronously) from all connected cards, to the host as an answer to a previously received command. A response is transferred serially on the CMD line.
- **Data** : data can be transferred from the card to the host or vice versa. Data is transferred via the data lines.

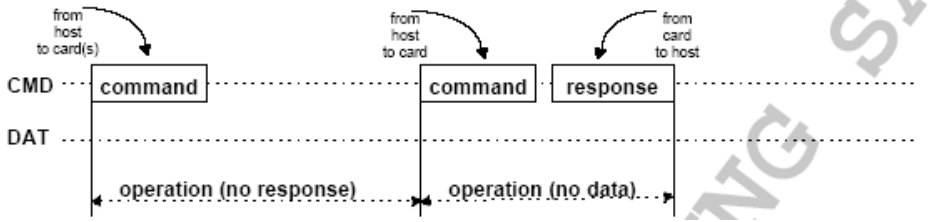


Figure 6-1 : "no response" And "no data" Operations

Card addressing is implemented using a session address, assigned to the card during the initialization phase. The basic transaction on the SD bus is the command/response transaction (refer to Figure 6-1).This type of bus transactions transfer their information directly within the command or response structure. In addition, some operations have a data token.

Data transfers to/from the SD Memory Card are done in blocks. Data blocks always succeeded by CRC bits. Single and multiple block operations are defined. Note that the Multiple Block operation mode is better for faster write operation.
A multiple block transmission is terminated when a stop command follows on the CMD line. Data transfer can be configured by the host to use single or multiple data lines.

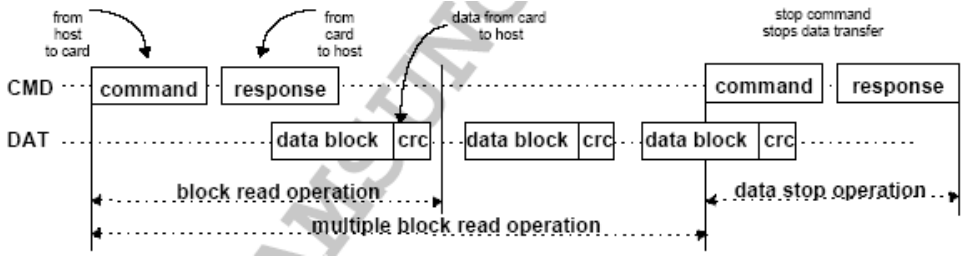


Figure 6-2 : (Multiple) Block read operation

The block write operation uses a simple busy signaling of the write operation duration on the DAT0 data line (see Figure 6-3) regardless of the number of data lines used for transferring the data.

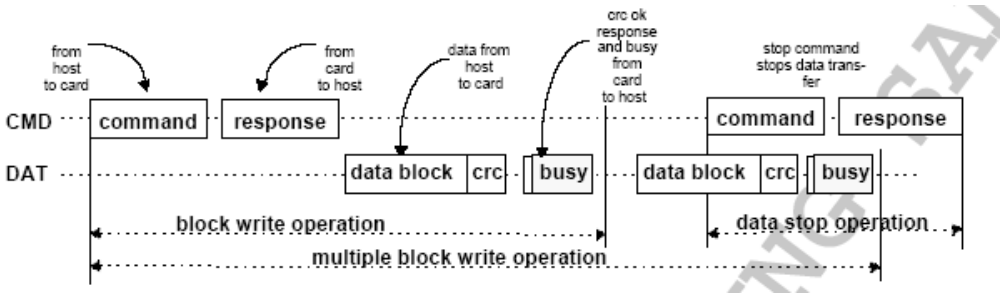


Figure 6-3 : (Multiple) Block Write Operation

Command tokens have the following coding sheme:

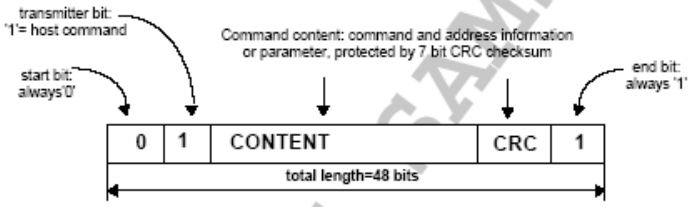


Figure 6-4 : Command token format

Each command token is preceded by a start bit(0) and succeeded by an end bit(1). The total length is 48 bits. Each token is protected by CRC bits so that transmission errors can be detected and the operation may be repreated.

Response tokens have four coding schemes depending on their content. The token length is either 48 or 136 bits.
The CRC protection algorithm for block data is a 16-bit CCITT polynomial. All used CRC types are described in Chapter 6.6

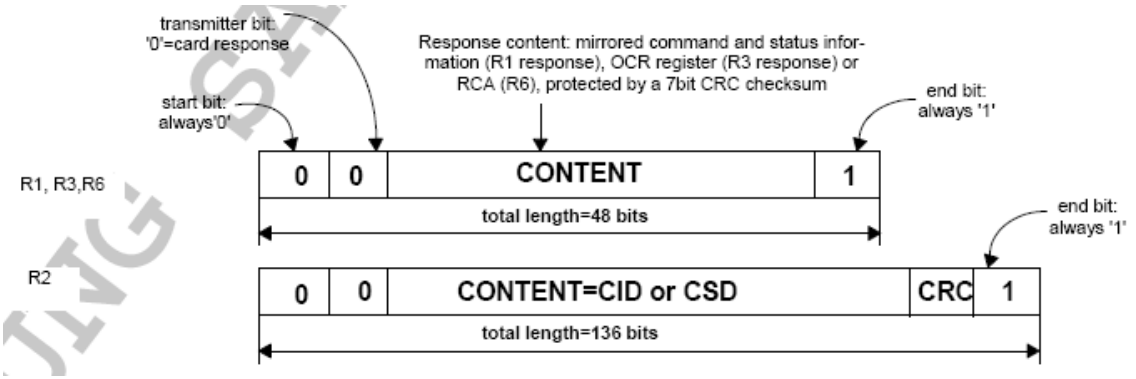


Figure 6-5 : Response Token Format

In the CMD line, the MSB bit is transmitted first, whereas the LSB bit is transmitted last.

When the wide bus option is used, the data is transmitted 4 bits at a time(see Figure 6-7). Start and end bits, as well as the CRC bits, are transmitted for every one of the DAT lines. CRC bits are calculated and checked for every DAT line individuality. The card sends the CRC status response and Busy indication to the host on DAT0 only (DAT1-DAT3 during that period are "don't care")

There are two types of Data packet format for the SD card.
(1) Usual data (8 bit width) The usual data (8 bit width) are sent in LSB (Least Significant Byte) first, MSB (Most Significant Byte) last manner. But in the individual byte it is MSB (Most Significant Bit) first, LSB (Least Significant Bit) last.
(2) Wide width data (SD Memory Register) The wide width data is shifted from MSB bit.

1. Data packet format for usual data (8bit width)

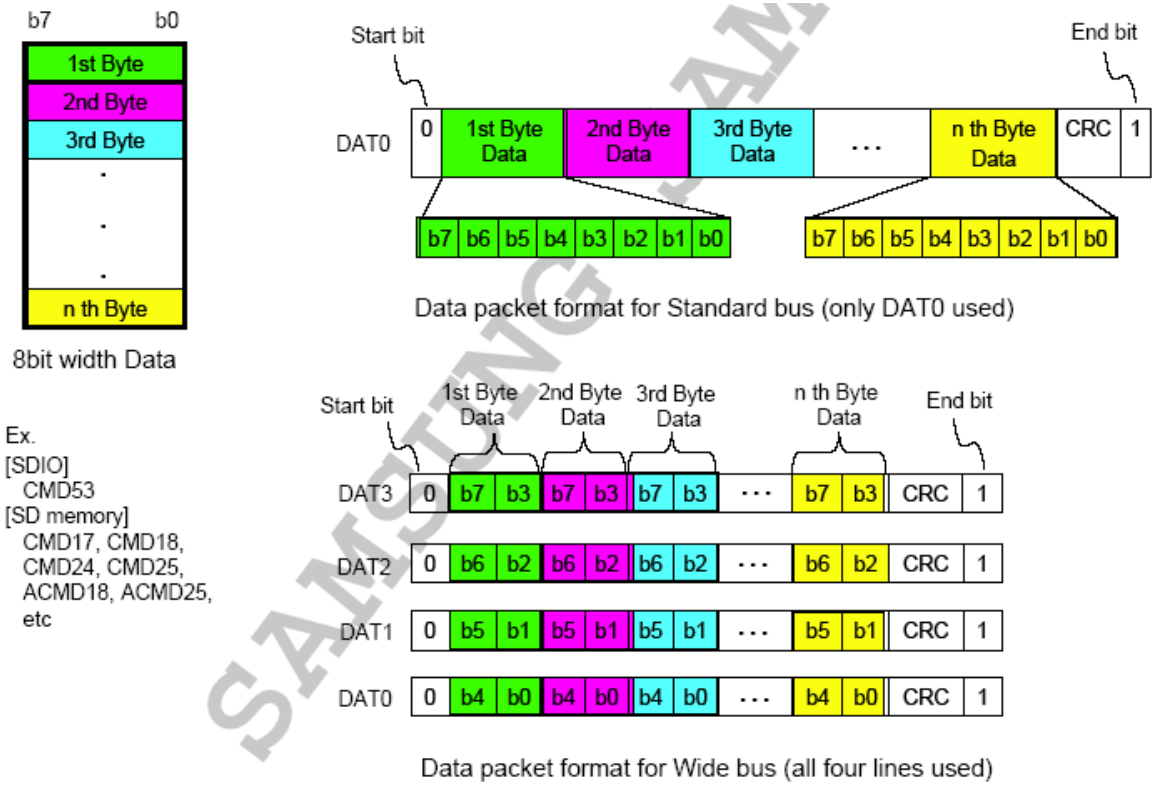


Figure 6-6 : Data Packet Format - Usual data

2. Data packet format for wide width data (Ex. ACMD13)

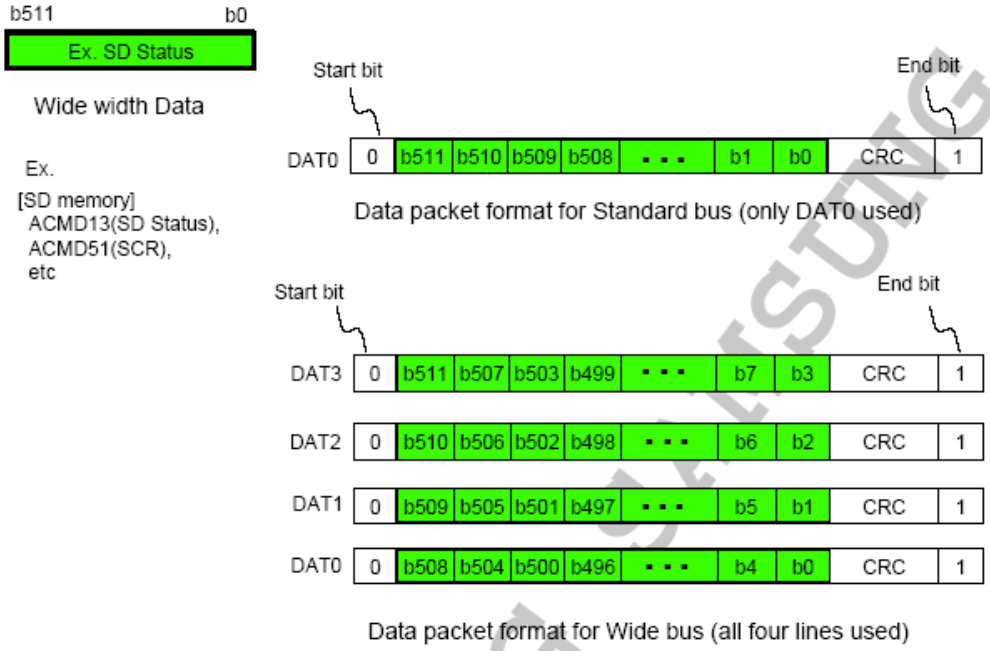


Figure 6-7 : Data Packet Format - Wide width data

6.2 SD(microSD) Card Functional Description

All communication between host and cards is controlled by the host (master). The host sends commands of two types: broadcast and addressed (point-to-point) commands.

- **Broadcast commands** : Broadcast commands are intended for all cards. Some of these commands require a response.
- **Addressed (point-to-point) commands** : The addressed commands are sent to the addressed card and cause a response from this card.

A general overview of the command flow is shown in Figure 6-8 for the card identification mode and in Figure 6-9 for the data transfer mode. The commands are listed in the command tables (Table 6-13 - Table 6-20). The dependencies between current state, received command and following state are listed in Table 6-21. In the following sections, the different card operation modes are described first. Thereafter, the restrictions for controlling the clock signal are defined. All SD(microSD) Card commands together with the corresponding responses, state transitions, error conditions and timings are presented in the succeeding sections.

Samsung microSD Cards have two opertion modes.

- **Card identification mode**
  The host will be in card identification mode after reset, while it is looking for new cards on the bus. Cards will be in this mode after reset, until the SEND_RCA command (CMD3) is received.
- **Data transfer mode**
  Cards will enter data transfer mode once theri RCA is first published. The host will enter data transfer mode after identifying all of the cards on the bus.

The following table shows the dependencies between operation modes and card states. Each state in the microSD Card state diagram (see Figure 6-8 and Figure 6-9) is associated with one operation mode:

Card state	Operation mode
Inactive State	Inactive
Idle State	Card identification mode
Ready State	
Identification State	
Stand-by State	Data transfer mode
Transfer State	
Sending-data State	
Receive-data State	
Programming State	
Disconnect State	

Table 6-1 : Overview of Card State vs. Operation Modes

6.3 Card Identification Mode

While in card identification mode the host resets all the cards that are in card identificaiton mode, validates operation voltage range, identifies cards and asks them to publish a Relative Card Address (RCA). This operation is daone to each card sperately on its own command (CMD) line. All data communication in the Card Identification Mode uses the command line (CMD) only.

6.3.1 Card Reset

GO_IDLE_STATE (CMD0) is the software-reset command and sets each microSD Card to *Idle State* regardless of the current card state. Cards in Inactive State are not affected by this command.

After power-on by host, all cards are in Idle State, including the cards that were in Inactive State.

After power-on or COMD0, all card's CMD lines are in input mode, wating for the start bit of the next command. The cards are initialized with a default relative card address. (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

6.3.2 Operating Voltage Range Validation

All cards shall be able to establish communication with the host using any operating voltage between V_{DD}-min and V_{DD}-max. However, during data transfer, minimum and maximum values for V_{DD} are defined in the operation condition register (OCR) and may not cover the whole range. microSD Card hosts are expected to read the card's OCR register and select proper V_{DD} values or reject the card.

Samsung microSD Cards that store the CID and CSD data in the payload memory can communicate this information only under data-transfer V_{DD} conditions. This means if host and card have non-compatible V_{DD}ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

SD_SEND_OP_COND(ACMD41) is designed to provide microSD card hosts with a mechanism to identify and reject cards which do not match the host's desired V_{DD} range. To accomplish this task, the host sends the required V_{DD} voltage window as the operand of this command. Samsung microSD Cards that cannot perform data transfer in the specified range must discard themselves from further bus operations and go into *Inactive State*. Note that ACMD41 is an application-specific command. Therefore, APP_CMD(CMD55) will always precede ACMD41. The RCA to be used for CMD55 in *idel_state* will be the card's default RCA = 0x0000.

The MultiMediaCard will not respond to ACMD41(actually it will not respond to APP_CMD---CMD55, that precedes it). The MultiMediacard will be initialized as per the MultiMediaCard specification, using SEND_OP_COND command(CMD1 of MultimediaCard), The host should ignore all ILLEGAL_COMMAND status in the MultiMediaCard response to CMD3, since it is a residue of ACMD41 which is invalid in the MultiMediaCard(CMD0,1,2 do not clear the status register). The host uses ACMD41 and CMD1 to distinguish between MultiMediaCard and microSD Cards in a system.

By omitting the voltage range in the command, the host can query each card and determine if there are any incompatibilities before sending out-of-range cards into the *Inactive State*. This query should be used if the host can select a common voltage range or wants to notify the application of non-usable cards in the stack. Afterwards, the host must choose a voltage for operation and reissue ACMD41 with this condition, sending incompatible cards into the *Inactive State*

The card can use the busy bit in the ACMD41 response to tell the host that it is still working on its power-up/reset procedure(e.g., downloading the register information from memory field) and is not ready yet for communication. In this case the host must repeat ACMD41 until the busy bit is cleared.

During the initialization procedure, the host is not allowed to change the operating voltage range. Changes in the OCR content will be ignored by the microSD Card. If there is a real change in the operating conditions, the host must reset the card stack (using CMD0) and begin the initialization procedure once more. However, for accessing the cards already in *Inactive State*, a hard reset must be done by switching the power supply off and on.

GO_INACTIVE_STATE(CMD15)can also be used to send an addressed microSD Card into the *Inactive State*. This command is used when the host explicitly wants to deactivate a card(e.g., host is changing V_{DD} into a range which is known to be not supported by this card).

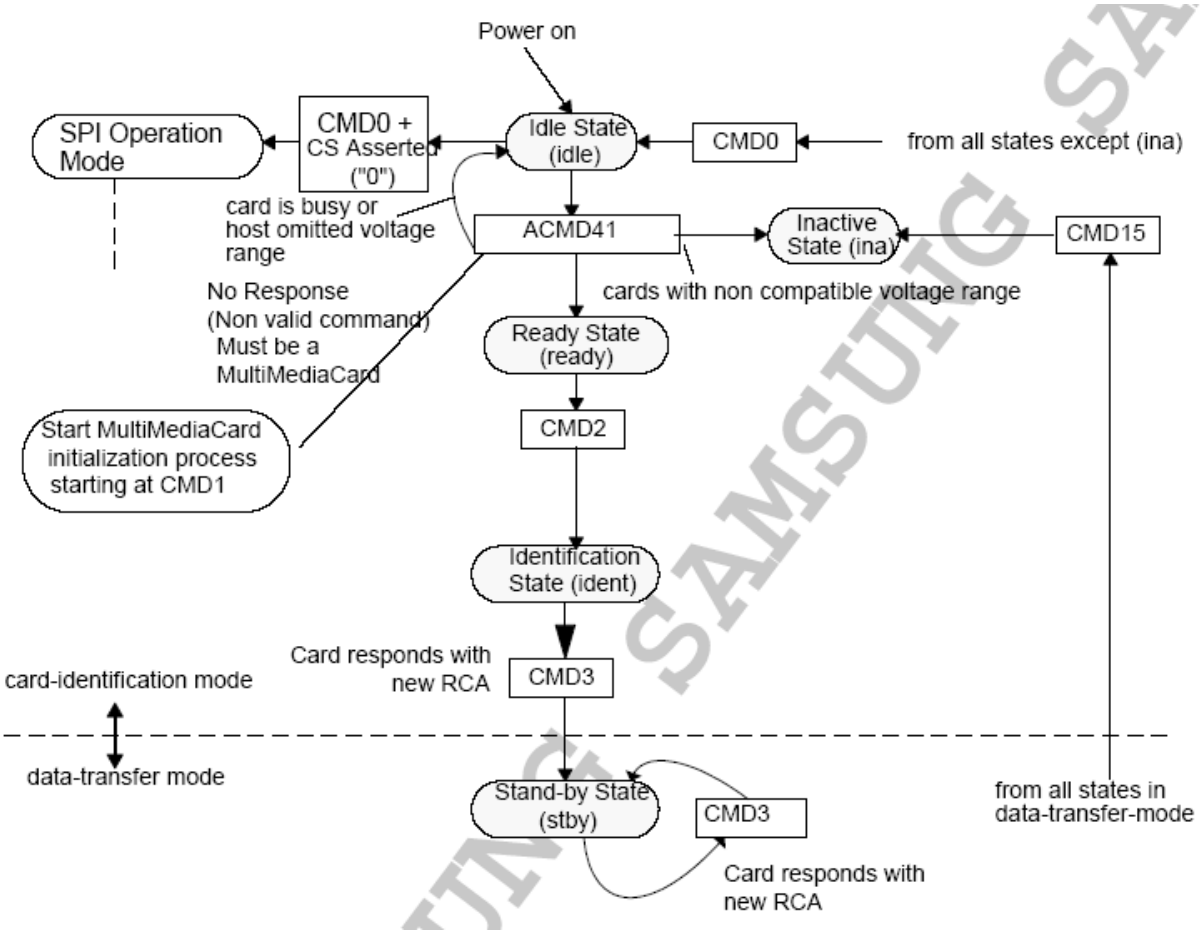


Figure 6-8: microSD Card State Diagram (Card Identification Mode)

6.3.3 Card Identification Process

The host starts the card identification process with the identification clock rate f_{OD} . In SD memory card the CMD line output drives are push-pull drivers.
After the bus is activated, the host will request the cards to send their valid operation conditions(ACMD41 preceding with APP_CMD---CMD55 with RCA=0x0000). The response to ACMD41 is the operation condition register of the card. The same command shall be send to all of the new cards in the system. Incompatible cards are sent into *Inactive State*. The host then issues the command ALL_SEND_CID(CMD2) to each card to get its unique card identification(CID)number. Card that is unidentified(i.e.,which is in *Ready State*) sends its CID number as the response(on the CMD line). After the CID is sent by the card, it goes into *Indentification State*. Thereafter, the host issues CMD3(SEND_RELATIVE_ADDR) asking the are to publish a new relative card address(RCA), which is shorter than CID and which will be used to address the card in the future data transfer mode(typically with a higher clock rate than f_{OD}). Once the RCA is received, the card state changes to the *Stand-by State*. At this point, if the host wants the card to have another RCA number, it may ask the card to publish a new number by sending another SEND_RELATIVE_ADDR command to the card. The last published RCA is the actual RCA number of the card.

The host repeats the identification process(i.e., the cycles with CMD2 and CMD3 for each card in the system).

After all the microSD Cards are initialized, the host will initialize the MultiMediaCards that are in the system (if any), using the CMD2 and CMD3 as given in the MultiMediaCard specification. Note that in the SD system, all the cards are connected separately so each MultiMediaCard will be initialized individually.

6.4 Data Transfer Mode

Until the content of all CSD registers is known by the host, the f_{PP} clock rate must remain at f_{OD} because some cards may have operating frequency restrictions. The host issues SEND _CSD (CMD9) to obtain the Card Specific Data (CSD register), e.g., block length, card storage capacity, maximum clock rate. Figure 6-8 shows a block diagram of the Data Transfer Mode.

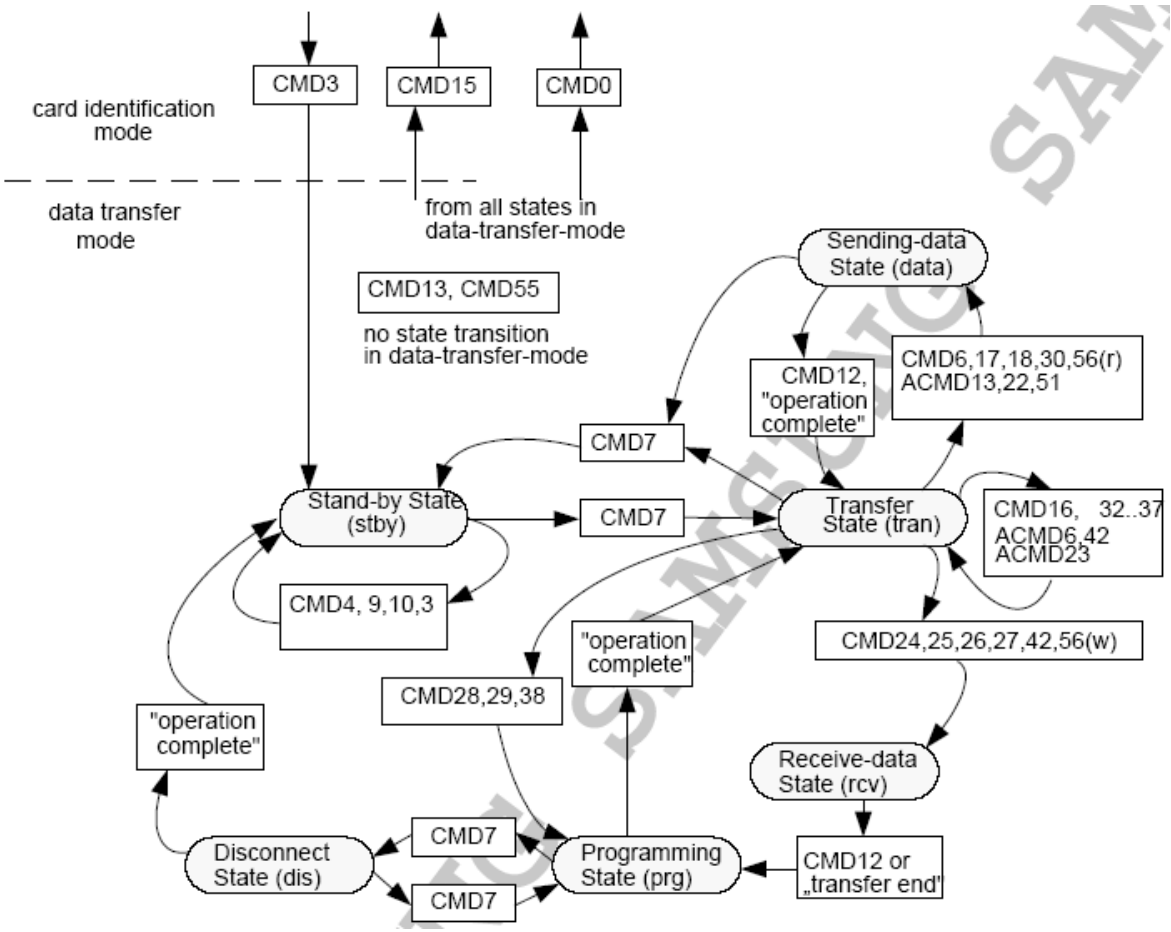


Figure 6-9 : microSD Card State Diagram (Data Transfer Mode)

CMD7 is used to select one card and place it in the Transfer State. Only one microSD Card can be in the Transfer State at a given time. If a previously selected microSD Card is in the Transfer State, its connection with the host is released and it will move back to the Stand-by State. When CMD7 is issued with the reserved relative card address "0x0000", all cards transfer back to Stand-by State. (Note that it is the responsibility of the Host to reserve the RCA=0 for card de-selection-refer to Table 6-13, CMD7 description). This may be used before identifying new cards without resetting other already registered cards. Cards that already have an RCA do not respond to identification commands. (ACMD41, CMD2) in this state.

Important Note: The card de-selection is done if a certain card gets CMD7 with un-matched RCA. That happens automatically if selection is done to another card and the CMD lines are common. So, in the microSD Card system, it will be the responsibility of the host either to work with the common CMD line (after initialization is done). In this case the card de-selection will be done automatically (as in MultiMediaCard system) or if the CMD lines are seperate then the host shall be aware to the necessity to deselect cards.

All data communication in the Data Transfer Mode is point-to point between the host and the selected card (using addressed commands). All addressed commands get acknowledged by a response on the CMD line.

The relationship between the various data transfer modes is summarized below

- All data read commands can be aborted any time by the stop command (CMD12). The data transfer will terminate and the card will return to the *Transfer State.* The read commands are: block read (CMD17), multiple block read (CMD18) send write protect (CMD30), send scr(ACMD51) and general command in read mode (CMD56).
- All data write commands can be aborted any time by the stop command (CMD12). The write commands must be stopped prior to deselecting the card by CMD7. The write commands are: block write (CMD24 and CMD25), write CID (CMD26), and write CSD (CMD27), lock/unlock command(CMD42) and general command in write mode(CMD56)
- As soon as the data transfer is completed, the card will exit the data write state and move either to the *Programming State* (transfer is successful) or *Transfer State* (transfer failed).
- If a block write operation is stopped and the block length and CRC of the last block are valid, the data will be pro grammed.
- The card may provide buffering for block write. This means that the next block can be sent to the card while the previous is being programmed.
- If all write buffers are full, and as long as the card is in *Programming State* (see microSD Card state diagram), the DAT0 line will be kept low(BUSY).
- There is no buffering option for write CSD, write CID, write protection and erase. This means that while the card is busy servicing any one of these commands, no other data transfer commands will be accepted. DAT0 line will be kept low as long as the card is busy and in the *Programming State*. Actually if the CMD and DAT0 lines of the cards are kept seperated and the host keeps the busy DAT0 lines disconnected from the other DAT0 lines (of the other cards), the host may access the other cards while the card is busy.
- Parameter set commands are *not* allowed while card is programming. Parameter set commands are: set block length (CMD16), erase block start (CMD32) and erase block end (CMD33).
- Read commands are *not* allowed while the card is programming.
- Moving another card from *Stand-by* to *Transfer State* (using CMD7) will not terminate a programming operation. The card will switch to the *Disconnect State* and will release the DAT0 line.
- A card can be reselected while in the *Disconnect State*, using CMD7. In this case the card will move to the *Program ming State* and reactivate the busy indication.
- Resetting a card (using CMD0 or CMD15) will terminate any pending or active programming operation. This may destroy the data contents on the card. It is up to the host's responsibility to prevent this.

6.4.1 Wide Bus Selection/ Deselection

Wide Bus (4 bit bus width) operation mode may be selected/ deselected using ACMD6. The default width after power up or GO_IDLE CMD0) is 1 bit bus width. AMD6 command is valid in '*tran state'* only. That means that the bus width may be changed only after a card was selected (CMD7).

6.4.2 2GByte Card

To make 2GByte card, the Maximum Block Length (READ_BL_LEN=WRITE_BL_LEN) shall be set to 1024 bytes. But Block Length set by CMD16 shall be up to 512 bytes to keep consistency with 512 bytes Maximum Block Length cards (Less than and equal 2GByte cards).

6.4.3 Data Read

The DAT bus line level is high by the pull-up when no data is transmitted. A transmitted data block consists of a start bit (1 or 4 bits LOW) followed by a continuous data stream. The data stream contains the payload data (and error correction bits if an off-card ECC is used). The data stream ends with an end bit (1 or 4 bits HIGH). The data transmission is synchronous to the clock signal. The payload for block oriented data transfer is protected by a 1 or 4 bits CRC check sum. (see Chapter 6.6)
The Read operation from SD Memory Card may be interrupted by turning the power off. The SD Memory Card ensures that data is not destroyed during all the conditions except write or erase operations issued by the host even in the event of sudden shut down or removal.
Read command is rejected if BLOCK_LEN_ERROR or ADDRESS_ERROR occurred and no data transfer is performed.

- **Block Read**

Block read is block oriented data transfer. The basic unit of data transfer is a block whose maxium size is always 512bytes Smaller blicks whose starting and ending address are entirely contained within 512 bytes boundary may be transmitted. Block Length set by CMD16 can be set up to 512 bytes regardless of READ_BL_LEN.
A CRC is appended to the end of each block ensuring data transfer integrity. CMD17(READ_SINGLE_BLOCK) initiates a block read and after completing the transfer, the card returns to the *Transfer State.* CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a STOP_TRANSMISSION command(CMD12) is issued. The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.
If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card shall detect a block misalignment at the beginning of the first mis aligned block, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait in the *Data State* for a stop command. The following table defines the card behavior when partial block accesses is enabled.

CSD value			Current Blocklen*1	Read CMD Start Address
Max block size (READ_BL_LEN)	Misalign	Partial		
512Bytes	0 (Disable)	1 (Enable)	1-512 bytes	Any address is forgiven*2
1kBytes	0 (Disable)	1 (Enable)	1-512 bytes	Any address is forgiven*2
2kBytes	0 (Disable)	1 (Enable)	1-512 bytes	Any address is forgiven*2

Table 6-2 : Read command blocklen

*1 : " Current Blocklen" size is set or changed by CMD 16. If value is less than or equal 512 bytes (there are no relations with Misalign and Partial option), it is set with no error.

*2 : When the Blocklen size data range crosses 512 bytes block boundary, card outputs the data until the 512 bytes block

boundary" and then the data becomes invalid and CRC error also may occur. The card will send "ADDRESS_ERRO" on the next command response. Host should issue CMD12 to recover.

6.4.4 Data Write

The data transfer format is similar to the data read. For block oriented write data transfer, the CRC check bits are added to each data block. The card performs a 1 or 4 bits CRC parity check for each received data block prior to the write operation. By this mechanism, writing of erroneously transferred data can be prevented.
Write command is rejected if BLOCK_LEN_ERROR or ADDRESS_ERROR occurred and no data transfer is performed.

- **Block Write**

During block write (CMD24 - 27,42,56(w)) one or more blocks of data are transferred from the host to the card with 1 or 4 bits CRC appended to the end of each block by the host. A card supporting block write shall be required that Block Length set by CMD16 shall be 512 bytes regardless of WRITE_BL_LEN is set to 1k or 2k bytes.

The following table defines the card behavior when partial block accesses is disabled (WRITE_BL_PARTIAL=0).

CSD value			Current Blocklen*1	Write CMD Start Address
Max block size (READ_BL_LEN)	Misalign	Partial		
512Bytes	0 (Disable)	0 (Disable)	512 bytes*2	n*512 bytes *3(n : Integer)
1kBytes	0 (Disable)	0 (Disable)	512 bytes*2	n*512 bytes *3(n : Integer)
2kBytes	0 (Disable)	0 (Disable)	512 bytes*2	n*512 bytes *3(n : Integer)

Table 6-3: Write command blocklen

*1 : "Current Blocklen" size is set or changed by CMD16. If value is less than 512bytes (there are no relations with Misalign and Partial option), it is set with no error. And then "Current Blocklen" size is tested when write command execution.
*2 : If the current Blocklen is other than this value, the card indicates "BLOCK_LEN_ERROR" on the Write command response.
*3 : If start address is other than this value, the card will send "ADDRESS_ERROR" on the Write command response.

If WRITE_BL_PARTIAL is allowed(=1) then smaller blocks, up to resolution of one byte, can be used as well. if the CRC fails, the card shall indicate the failure on the DAT line(see below); the transferred data will be discarded and not written, and all further transmitted blocks(in multiple block write mode) will be ignored.
Multiple block write command shall be used rather than continuous single write command to make faster write operation. If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed(CSD parameter WRITE_BLK_MISALIGN is not set), the card shall detect the block misalignment error and abort programming before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, wait in the *Receive-data-State* for a stop command.
The write operation shall also be aborted if the host tries to write over a write protected area. In this case, however, the card shall set the WP_VIOLATION bit.

Programming of the CSD register does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.
Some cards may require long and unpredictable times to write a block data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command(CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the *Disconnect State* and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable. Actually, the host may perform simultaneous write operation to several cards with inter-leaving process. The interleaving process can be done by accessing each card separately
while other cards are in busy. This process can be done by proper CMD and DAT0-3 line manipulations (disconnection of busy cards).

- **Pre-erase setting prior to a multiple block write operation**

Setting a number of write blocks to be pre_erased(ACMD23) will make a following Multiple Block Write operation faster compared to the same operation without preceding ACMD23. The host will use this command to define how many number of write blocks are going to be send in the next write operation. If the host will terminate the write operation (Using stop transmission) before all the data blocks sent to the card the content of the remaining write blocks is undefined(can be either erased or still have the old data). If the host will send more number of write blocks than defined in ACMD23 the card will erase block one by one(as new data is received). This number will be reset to the default(=1) value after Multiple Blocks Write operation.
It is recommended using this command preceding CMD25, some of the cards will be faster for Multiple Write Blocks operation. Note that The host must send ACMD23 just before WRITE command if the host wants to use the pre-erase feature. If not, pre-erase-count might be cleared automatically when another commands (ex : Security Application Commands) are executed.

- **Send Number of Written Blocks**

Systems that use Pipeline mechanism for data buffers management are, in some cases, unable to determine which block was the last to be well written to the flash if an error occurs in the middle of a Multiple Blocks Write operation. The card will respond to ACMD22 with the number of well written blocks.

6.4.5 Erase

It is desirable to erase many write blocks simultaneously in order to enhance the data throughput. Identification of these write blocks is accomplished with the ERASE_WR_BLK_START(CMD32), ERASE_WR_BLK_END(CMD33) commands. The host must adhere to the following command sequence: ERASE_WR_BLK_START, ERASE_WR_BLK_END and ERASE(CMD38).
If an erase(CMD38) or address setting (CMD32, 33) command is received out of sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole sequence.
If an out of sequence command (except SEND_STATUS) is received, the card shall set the ERASE_RESET status bit in the status register, reset the erase sequence and execute the last command.
If the erase range includes write protected sectors, they shall be left intact and only the non protected sectors shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set.
The address field in the address setting commands is a write block address in byte units. The card will ignore all LSB's below the WRITE_BL_LEN(see CSD) size.
As described above for block write, the card will indicate that an erase is in progress by holding DAT0 low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card or perform card disconnection, as described in the Block Write section, above.
The data at the card after an erase operation is either '0 or '1, depends on the card vendor.
The SCR register bit DATA_STAT_AFTER_ERASE(bit 55) defines whether it is '0 or '1.

6.4.6 Write Protect Management

Three write protect methods are supported in the SD Memory Card as follows:
- Mechanical write protect switch (Host responsibility only)
- Card internal write protect(Card's responsibility)
- Password protection card lock operation.

• Mechanical Write Protect Switch

A mechanical sliding tablet on the side of the card(refer to the mechanical description) will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is close the card is not write protected.
A proper, matched, switch on the socket side will indicate to the host that the card is write protected or not. It is the responsibility of the host to protect the card. The position of the write protect switch is un-known to the internal circuitry of the card.

• microSD Card's internal Write Protection

Card data may be protected against either erase or write. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD, portions of the data may be protected(in units of WP_GRP_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_WRITTE_PROT command clears the write protection of the addressed write-protect group.
The SEND_WRITE_PROT command is similar to a single block read command. The card shall send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSB's below the group size.
The Password Card Lock protection is described in the following section.

6.4.7 Card Lock/Unlock Operation

The password protection feature enables the host to lock the card by providing a password, which later will be used for unlocking the card. The password and its size is kept in an 128 bit PWD and 8 bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.
Locked card responds to (and executes) all commands in the “basic” command class (class 0), ACMD41 and “lock card” command class. Thus the host is allowed to reset, initialize, select, query for status, etc., but not to access data on the card. If the password was previously set (the value of PWD_LEN is not ‘0’) the card will be locked automatically after power on.
Similar to the existing CSD and CID register write commands the lock/unlock command is available in “transfer state” only. This means that it does not include an address argument and the card has to be selected before using it.
The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). The following table describes the structure of the command data block.

Byte #	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password data							
...								
PWDS_LEN + 1								

Table 6-4 : Lock Card Data Structure

- **ERASE:** ‘1’ Defines Forced Erase Operation. In byte 0 bit 3 will be set to "1" (all other bits shall be ‘0’). All other bytes of this command will be ignored by the card.
- **LOCK/UNLOCK**: ‘1’ = Locks the card. ‘0’ = Unlock the card (note that it is valid to set this bit together with SET_PWD but it is not allowed to set it together with CLR_PWD).
- **CLR_PWD**: ‘1’ = Clears PWD.
- **SET_PWD**: ‘1’ = Set new password to PWD
- **PWDS_LEN**: Defines the following password/s length (in bytes). In case of Password change, this field include the toatal password lengths of old and new passwords. The password length is up to 16 bytes. In case of password change the total length of the old password and the new password can be up to 32 bytes.
- **Password data:** In case of set new password contains the new password. In case of password change it contains the old password followed by new password.

The data block size shall be defined by the host before it sends the card lock/unlock command. The block length shall be set to greater than or equal required data structure of lock/unlock command. In the following explanation, changing block size by CMD16 is not mandatory requirement for the lock/unlock command.

- **Setting the Password**
  - Select the card (CMD7), if not previously selected already
  - Define the block length (CMD16), given by the 8bit card lock/unlock mode, the 8 bits password size (in bytes), and the number of bytes of the new password. In case that a password *replacement* is done, then the block size shall consider that both passwords, the old and the new one, are sent with the command.
  - Send Card Lock/Unlock command with the appropriate data block size on the data line including 16 bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWD_LEN) and the password itself. In case that a password *replacement* is done, then the length value (PWD_LEN) shall include both passwords, the old and the new one, and the PWD field shall include the old password (currently used) followed by the new pass word. Note that card shall handle internally the calculation of the new password length by subscribing the old password length from PWDS_LEN filed.
  - In case that the sent old password is not correct (not equal in size and content) then LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In case that the sent old password is correct (equal in size and content) then the given new password and its size will be saved in the PWD and PWD_LEN fields, respectively.

Note that the password length register (PWD_LEN) indicates if a password is currently set. When it equals '0' there is no password set. If the value of PWD_LEN is not equal to zero the card will lock itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (while setting the password) or sending additional command for card lock.

- **Reset the Password:**
  - Select the card (CMD7), if not previously selected already
  - Define the block length (CMD16), given by the 8 bit card lock/unlock mode, the 8 bit password size (in bytes), and the number of bytes of the currently used password.
  - Send the card lock/unlock command with the appropriate data block size on the data line including 16 bit CRC. The data block shall indicate the mode CLR_PWD, the length (PWD_LEN) and the password (PWD) itself (LOCK/ UNLOCK bit is don't care). If the PWD and PWD_LEN content match the sent password and its size, then the con tent of the PWD register is cleared and PWD_LEN is set to 0. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

- **Locking a card:**
  - Select the card (CMD7), if not previously selected already
  - Define the block length (CMD16), given by the 8 bit card lock/unlock mode, the 8 bit password size (in bytes), and the number of bytes of the currently used password.
  - Send the card lock/unlock command with the appropriate data block size on the data line including 16 bit CRC. The data block shall indicate the mode LOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct then LOCK_UNLOCK_FAILED error bit will be set in the status register.
Note that it is possible to set the password and to lock the card in the same sequence. In such case the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent.
If the password was previously set (PWD_LEN is not '0'), then the card will be locked automatically after power on reset. An attempt to lock a locked card or to lock a card that does not have a password will fail and the LOCK_UNLOCK_FAILED error bit will be set in the status register, unless it was done during a password definition or change operations.

- **Unlocking the card:**
  - Select the card (CMD7), if not previously selected already.
  - Define the block length (CMD16), given by the 8 bit card lock/unlock mode, the 8 bit password size (in bytes), and the number of bytes of the currently used password.
  - Send the card lock/unlock command with the appropriate data block size on the data line including 16 bit CRC. The data block shall indicate the mode UNLOCK, the length (PWD_LEN) and the password (PWD) itself.

If the PWD content equals to the sent password then the card will be unlocked and the card-locked status bit will be cleared in the status register. If the password is not correct then the LOCK_UNLOCK_FAILED error bit will be set in the status register.

Note that the unlocking is done only for the current power session. As long as the PWD is not cleared the card will be locked automatically on the next power up. The only way to unlock the card is by clearing the password.
An attempt to unlock an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register, unless it was done during a password definition or change operations.

- **Forcing Erase:**

In case that the user forgot the password (the PWD content) it is possible to erase all the card data content along with the PWD content. This operation is called *Forced Erase*.
  - Select a card (CMD7), if not previously selected already.
  - Define the block length (CMD16) to 1 byte (8bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of one byte on the data line including 16 bit CRC. The data block shall indicate the mode ERASE (the ERASE bit shall be the only bit set).

If the ERASE bit is not the only bit in the data field then the LOCK_UNLOCK_FAILED error bit will be set in the status register and the erase request is rejected.If the command was accepted then ALL THE CARD CONTENT WILL BE ERASED including the PWD and PWD_LEN register content and the locked card will get unlocked. An attempt to force erase on an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

- **Parameter and the Result of CMD42:**

The block length shall be greater than or equal required data structure of CMD42; otherwise, the result of CMD42 is undefined and the card may be in the unexpected locked state. Table 6-5 clarifies the behavior of CMD42. The reserved bits in the parameter (bit7-4) of CMD42 shall be dont's care. In the case of CMD42 requires the password, it is assumed that the old password and the new password are set correctly; otherwise the card indicates error regardless of Table 6-5. If the password length is 0 or greater than 128 bits, the card indicates error. If errors occur during execution of CMD42, the LOCK_UNLOCK_FAILED (Bit24 of Card Status) shall be set to 1 regardless of Table 6-5. The CARD_IS_LOCKED (Bit25 of Card Status) in the response of CMD42 shall be the same as Current Card State in Table 6-5. In the field of Card Status, "0" to 1" means the card change to Locked and "1 to 0" means the card change to Unlocked after execution of CMD42. It can be seen in the response of CMD13 after the CMD42. The LOCK_UNLOCK_FAILED (Bit24 of Card Status) as the result of CMD42 can be seen in the response of either CMD42 or following CMD13.

CMD42 Parameter in the data
Bit3: ERASE
BIt2: LOCK_UNLOCK
Bit1: CLR_PWD
Bit0: SET_PWD

Related bits in the Card Status
Bit25: CARD_IS_LOCKED
Bit24: LOCK UNLOCK FAILED

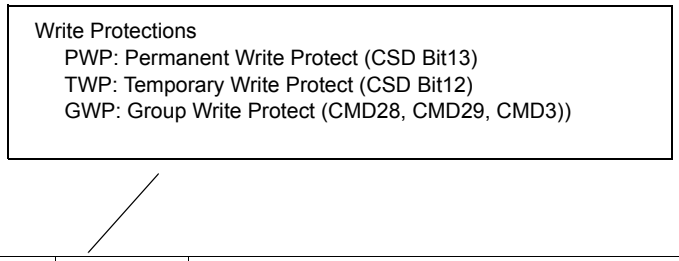
CMD42 Parameter				Current Card State	PWDS_LEN and PWD	Result of Function	Card Status	
Bit3	Bit2	Bit1	Bit0				Bit25	Bit24
After Power On					Exist Cleared	The card is locked The card is unlocked	1 0	0 0
1 1 1	0 0 0	0 0 0	0 0 0	Locked Unlocked Unlocked	Exist Exist Cleared	Force Erase (Refer to Table 6-6) Error Error	Table6-6 0 0	Table6-6 1 1
0 0 0	1 1 1	0 0 0	0 0 0	Locked Unlocked Unlocked	Exist Exist Cleared	Error Lock the card Error	1 0 to 1 0	1 0 1
0 0 0	1 1 1	0 0 0	1 1 1	Locked Unlocked Unlocked	Exist Exist Cleared	Replace password and the card is still locked Replace password and the card is locked Set Password and lock the card	1 0 to 1 0 to 1	0 0 0
0 0 0	0 0 0	1 1 1	0 0 0	Locked Unlocked Unlocked	Exist Exist Cleared	Clear PWD_LEN and PWD and the card is unlocked Clear PWD_LEN and PWD Error (Note *4 Refer to Table 6-8)	1 to 0 0 0	0 0 1
0 0 0	0 0 0	0 0 0	1 1 1	Locked Unlocked Unlocked	Exist Exist Cleared	Replace password and the card is unlocked Replace password and the card is unlocked Set password and the card is still unlocked	1 to 0 0 0	0 0 0
0 0 0	0 0 0	0 0 0	0 0 0	Locked Unlocked Unlocked	Exist Exist Cleared	Unlock the card Error Error	1 to 0 0 0	0 1 1
Other combinations				Don't care	Don't care	Error (Note *1 Refer to Table 6-8)	0 or 1	1

Table 6-5: Lock Unlock Function (Basic Sequence for CMD42)

Application Note:
To replace password, the host should consider following cases. When PWD_LEN and password data exist, the card assumes old and new passwords are set in the data structure. When PWD_LEN and PWD are cleared, the card assumes only new password is set in the data structure. In this case, the host shall not set old password in the data structure; otherwise, unexpected password is set.

- **Force Erase Function to the Locked Card:**

Table 6-6 clarifies the relation between force erase and Write Protection. The force erase does not erase the secure area. The card shall keep locked state during the erase execution and change to unlocked state after the erase of all user area is completed. Similarly, The card shall keep Temporary and Group Write Protection during the erase execution and clear Write Protection after the erase of all user area is completed. In the case of erase error occur, the card can continue force erase if the data of error sectors are destroyed.



CMD42 Parameter				PWP	TWP GWP	Result of Function	Card Status	
Bit3	Bit2	Bit1	Bit0				Bit25	Bit24
1	0	0	0	Yes	don't care	Error (Note *2 Refer to Table 6-8)	1	1
1	0	0	0	No	Yes	Execute force erase and clear Temporary Write Protect and Group Write Protect. (Note *3 Refer to Table 6-8)	1 to 0	0
1	0	010		No	No	Execute force erase	1 to 0	0

Table 6-6: Force Erease Function to the Locked card (Relation to the Write Protects)

- **Relation Between ACMD6 and Lock / Unlock State:**

ACMD6 is rejected when the card is locked and bus width can be changed only when the card is unlocked.

Table 6-7 shows the relation between ACMD6 and Lock / Unlock state.

Card State	Bus mode	Result of the Function
Unlocked	1-bit mode	ACMD6 is accepted
Locked	1-bit mode	ACMD6 is rejected and still in 1-bit mode
Unlocked	4-bit mode	ACMD6 is accepted
Locked	4-bit mode	ACMD6 is rejected and still in 4-bit mode. CMD0 change to 1-bit mode

Table 6-7 : Relation Between ACMD6 and Lock / Unlock State

Application Note:
After power on (in 1-bit mode), if the card is locked, the SD mode host shall issue CMD42 in 1-bit mode. If the the card is locked in 4-bit mode, the SD mode host shall issue CMD42 in 4-bit mode.

- **Commands accepted for Locked card:**

The locked card shall accept commands listed below and return response with setting CARD_IS_LOCKED.

1) Basic class (0)
2) Lock card class (7)
3) CMD16
4) ACMD41
5) ACMD42

All other commands including security commands are treated as illegal commands.

Application Note:
After power on, the host can recognize the card lock/unlock state by the CARD_IS_LOCKED in the response of CMD7 or CMD13

- **Two types of Lock / Unlock Card:**

There will be two types of lock / unlock function-supported card. The Type 1 is earlier version of SD memory card and the Type 2 is new version defined in this version of the specification. Table 6-8 shows the difference between these types of card. The SD memory cards that support Lock / Unlock and comply with Version 1.01, can take either Type 1 or Type 2. The SD memory cards that support Lock / Unlock and comply with Version 1.10, shall take Type 2.

Notes	Type 1 Card (Earlier version)	Type 2 Card (New version)
*1 in Table 6-5	Treat CMD42 Parameter=0011b as 0001b. Treat CMD42 Parameter=0111b as 0101b. Treat CMD42 Parameter=0110b as 0010b. Results of other combinations are Error.	All results are Error
*2 in Table 6-6	Execute force erase and set Permanent Write Protect. If force erase is completed, the CARD_IS_LOCKED is changed from 1 to 0. A priority is given to force erase from Permanent Write Protect.	The result is Error A priority is given to Permanent Write Protect from force erase.
*3 in Table 6-6	Execute force erase but Temporary Write Protect and Group Write Protect are not cleared. It is in need of the host clear.	Execute force erase and clear Temporary Write Protect and Group Write Protect.
*4 in Table 6-5	CMD42 Parameter=0010 and CMD42 Parameter=0110 The result is no error. Card status Bit24 will be 0	The result is Error. Card status Bit24 will be 1

Table 6-8 : Difference of earlier version and new version of lock / unlock function

Application Note:
The host can use both types of card without checking difference by taking account of following points.
(1) The host should not set the parameters of CMD42 that return error in Table 6-5. (For *1)
(2) The host should not issue force erase command if the Permanent Write Protect is set to 1. otherwise the Type 1 card cannot be used any more even if the user remembers the pass word. (For *2)
(3) After the force erase, if the Temporary Write Protect is not cleared, the host should clear it. (For *3)

6.4.8 Copyright Protection

Detailed description of the Copyrights Protection mechanism and the related security microSD Card commands can be founded in *"SD Memory Card Security Specification"* document from the SD Association. All SD Card security related commands operate in the data transfer mode.

As defined in the SDMI spec, the data content that is saved in the card is saved already encrypted and it pass transparently to/form the card. NO operation is done on the data and there is no restriction to read the data at any time. Associated to every data packet (song, for example) that is saved in the un-protected memory there is a special data that shall be saved in a protected memory area. For any access (any Read or Write or Erase Command) from/to the data in a protected memory area, an authentication procedure shall be done between the card and the connected device, either the LCM(PC for example) or the PD(Portable Device - SD player for example). After the authentication process was passed OK, the card is ready to accept or give data from/to the connected device. While the card is in the secured mode of operation (after the authentication succeeded) the argument and the associated data that is sent to the card or read from the card are encrypted. At the end of the Read/Write/Erase operation the card gets out automatically of its secured mode.

6.4.9 Application specific commands

The SD(microSD) Card system is designed to provide a standard interface for a variety applications types. In this environment, it is anticipated that there will be a need for specific customers/applications features. To enable a common way of implementing these features, two types of generic commands are defined in the standard:

- **Application Specific Command – APP_CMD (CMD55)**

This command, when received by the card, will cause the card to interpret the following command as an application specific command, ACMD. The ACMD has the same structure as of regular MultiMediaCard standard commands and it may have the same CMD number. The card will recognize it as ACMD by the fact that it appears after APP_CMD.
The only effect of the APP_CMD is that if the command index of the, immediately, following command has an ACMD overloading, the non standard version will used. If, as an example, a card has a definition for ACMD13 but not for ACMD7 then, if received immediately after APP_CMD command, Command 13 will be interpreted as the non standard ACMD13 but, command 7 as the standard CMD7. In order to use one of the manufacturer specific ACMD's the host will:

- Send APP_CMD. The response will have the APP_CMD bit (new status bit) set signaling to the host that ACMD is now expected.
- Send the required ACMD. The response will have the APP_CMD bit set, indicating that the accepted command was interpreted as ACMD. If a non-ACMD is sent then it will be respected by the card as normal microSD Card command and the APP_CMD bit in the Card Status stays clear.

If a non valid command is sent (neither ACMD nor CMD) then it will be handled as a standard microSD Card illegal command error.
From the SD(microSD) Card protocol point of view the ACMD numbers will be defined by the manufacturers with some restrictions. The following ACMD numbers are reserved for the SD(microSD) Card proprietary applications and may not be used by any SD(microSD) Card manufacturer:
ACMD6, ACMD13, ACMD17-25, ACMD38-49, ACMD51

- **General Command - GEN_CMD (CMD56)**

The bus transaction of the GEN_CMD is the same as the single block read or write commands (CMD24 or CMD17). The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not a memory payload data but has a vendor specific format and meaning. The card shall be selected ('*tran_state*') before sending CMD56. The data block size is the BLOCK_LEN that was defined with CMD16. The response to CMD56 will be R1b.
Note that there are no reserved data pattern (of the associated data block) for SD(microSD) Card specific applications.

6.4.8 Switch function command

Switch function command (CMD6)[1] is used to switch or expand memory card functions. Currently there are two function groups defined:

- Card access mode - 12.5MB/sec interface speed (default) or 25MB/sec interface speed. (highspeed)
- Card command system - Standard command set (default) or eCommerce command set or Vendor Specific Command set.

This is a new feature, introduced in SD physical Layer Specification Version 1.10. Therefore, cards that are compatible with earlier versions of the spec do not support it. The host shall check the "SD_SPEC" field in the SCR register to recognize what version of the spec the card complies with before using CMD6. It is mandatory for SD memory card of Ver1.10 to support CMD6.

CMD6 is valid under the "Transfer State". Once selected, via the switch command, all functions only return to the default function after a power cycle, CMD6 (Mode 1 operation with Function 0 in each function group) or CMD0. Executing a power cycle or issuing CMD0, will cause the card to reset to the "idle" state and all the functions to switch back to the default function.

As a response to CMD6, the SD Memory Card will send R1 response on the CMD line, and 512 bits of status on the DAT lines. From the SD bus transaction point of view, this is a standard single block read transaction and the time out value of this command is 100ms, same as in read command. If CRC error occurs on the status data the host should issue a power cycle.

CMD6 function switching period is within 8 clocks after the end bit of status data. When CMD6 changes the bus behavior (i.e. access mode) the host is allowed to use the new functions (increase/decrease CLK frequency beyond the current max CLK frequency), at least 8 clocks after at the end of the switch command transaction (see Figure 6-10).

In response to CMD0, the switching period is within 8 clocks after the end bit of CMD0. When CMD6 has changed the bus behavior (i.e. access mode) the host is allowed to start the initialization process, at least 8 clocks after at the CMD0.

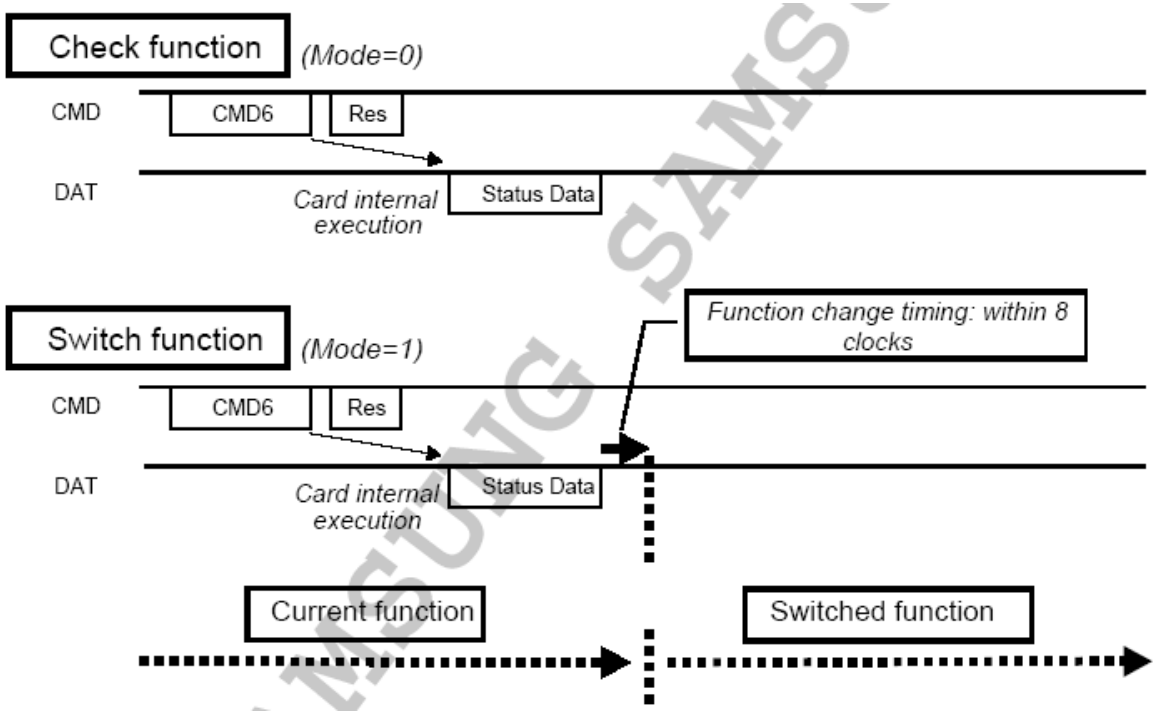


Figure 6-10: Usage of Switch command

CMD6 supports six function groups, and each function group supports sixteen branches (functions). Only one function can be chosen and active in a given function group. Function 0 in each function group is the default function (compatible with Spec. 1.01).

CMD6 can be used in two modes:
- Mode 0 (Check function) is used to query if the card supports a specific function or functions.
- Mode 1 (set function) is used to switch the functionality of the card.

- **Mode 0 Operation - (Check function)**

CMD6 is used in mode 0 to query which functions the card supports, and to identify the maximum current consumption of the card under the selected functions.
Refer to Table 6-20: Switch function commands (class 10) for the argument definition of CMD6.

A query is done by setting the argument field of the command, as follows:
• Setting the Mode bit to "0"
• Selecting only one function in each function group. Selection of default function is done by setting the function to 0x0. Selecting a specific function is done by using appropriate values from Table 6-10. Selecting 0xF will keep the current function that has been selected for the function group.

In response to a query, the switch function will return the following 3 statuses (see Table 6-10):
• The functions that are supported by each of the function groups
• The function that the card will switch to, in each of the function groups. This value is identical to the provided argument if the host made a valid selection or 0xF if the selected function was invalid.
• Maximum current consumption under the selected functions. If one of the selected functions was wrong the return value will be 0.

- **Mode 1 Operation - Set Function**

CMD6 is used in mode 1 to switch the functionality of the card.

Switching to a new functionality is done by:
• Setting the Mode bit to "1"
• Selecting only one function in each function group. Selection of default function is done by setting the function to 0x0. It is recommended to specify 0xF (no influence) for all selected functions, except for functions, which need to be changed. Selecting 0xF will keep the current function for the function group.

In response to a set function, the switch function will return the following 3 statuses:
• The functions that are supported by each of the function groups
• The function, which is the result of the switch command. In case of invalid selection of one function or more, all set values are ignored and no change will be done (identical to the case where the host selects 0xF for all functions groups). The response to an invalid selection of function will be 0xF.
• Maximum current consumption under the selected functions. If one of the selected functions was wrong the return value will be 0.

Arg. Slice	[23:20]	[19:16]	[15:12]	[11:8]	[7:4]	[3:0]
Group No.	6	5	4	3	2	1
Function name	Reserved	Reserved	Reserved	Reserved	Command system	Access mode
0x0	Default (Ver. 1.01)					
0x1	Reserved	Reserved	Reserved	Reserved	For eC	High-Speed
0x2	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x3	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x4	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x5	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x6	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x7	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x8	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0x9	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xA	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xB	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xC	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xD	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
0xE	Reserved	Reserved	Reserved	Reserved	Vendor specific	Reserved
0xF	No influence					

Table 6-9 : Functions

- **Switch function status**

The switch function status is the returned data block that contains function and current consumption information. The block length is predefined to 512 bits and the use of SET_BLK_LEN command is not necessary. Table 6-10 describes the status data structure.

The status bits of the response contain the information of the function group. Maximum current consumption will be used only for the new function added through this command. In this case VDD_R_CURR_MIN, VDD_W_CURR_MIN, VDD_R_CURR_MAX and VDD_W_CURR_MAX values in the CSD register provides the current consumption when all card functions are set to the default state and can be used by spec 1.01 compatible hosts.

Bits	Description	Width
511:496	Maximum current consumption (0:Error, 1:1mA, 2:2mA… , 65,535:65,535mA) under the function shown with [399:376] bits. The voltage to calculate current consumption is defined by ACMD41 (SD memory card) or CMD5 (SD I/O card). Maximum current consumption indicates the total card current(memory portion) if the functions are switched. The host should check the maximum current consumption and verify that it can supply the necessary current before mode 1 operation. Maximum current consumption is average over 1second.	16
495:480	Function group 6, information . If a bit i is set, function i is supported	16
479:464	Function group 5, information . If a bit i is set, function i is supported	16
463:448	Function group 4, information . If a bit i is set, function i is supported	16
447:432	Function group 3, information . If a bit i is set, function i is supported	16
431:416	Function group 2, information . If a bit i is set, function i is supported	16
415:400	Function group 1, information . If a bit i is set, function i is supported	16
399:396	mode 0 - The function which will be switched in function group 6. mode 1 - The function which is result of the switch command, in function group 6. 0xF shows function set error with the argument	4
395:392	mode 0 - The function which will be switched in function group 5. mode 1 - The function which is result of the switch command, in function group 5. 0xF shows function set error with the argument	4
391:388	mode 0 - The function which will be switched in function group 4. mode 1 - The function which is result of the switch command, in function group 4. 0xF shows function set error with the argument	4
387:384	mode 0 - The function which will be switched in function group 3. mode 1 - The function which is result of the switch command, in function group 3. 0xF shows function set error with the argument	4
383:380	mode 0 - The function which will be switched in function group 2. mode 1 - The function which is result of the switch command, in function group 2. 0xF shows function set error with the argument	4
379:376	mode 0 - The function which will be switched in function group 1. mode 1 - The function which is result of the switch command, in function group 1. 0xF shows function set error with the argument	4
375:0	Reserved (All ’0’)	376

Table 6-10 : Status data structure

6.4.11 Relationship between CMD6 data & other commands

This card may accept the commands using only CMD line (CMD12, CMD13, etc) during CMD6 transaction but its response and result is undefined.

Application Note: The host is advised not to issue any command during CMD6 transaction. If the host cannot get valid data of CMD6, it advised to issue CMD0 and try re-initialization.

- **Relationship between CMD6 data & CMD12**

Case1: Not complete case (The card does not output all data)

In case that the host sends the end bit of CMD12 before CRC bit 15, CMD6 is stopped by CMD12, card shall terminate data transfer of CMD6. The card behavior is not guaranteed and re-initialization from CMD0 is the only way to recover from undefined sate. The end bit of the host command is followed, on the data line, with one more data bit and one end bit.

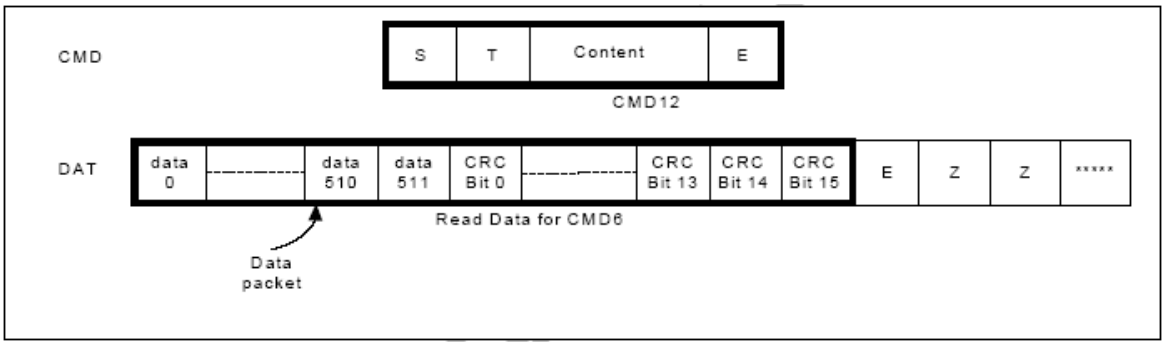


Figure 6-11: CMD12 during CMD6; Case 1

Case 2: Complete case (The card outputs all data)

The card shall complete CMD6 execution and its behavior is guaranteed. Complete case includes the later timing of CMD12 than Figure 6-12. The end bit of the host command is followed, by the end bit on the data line.

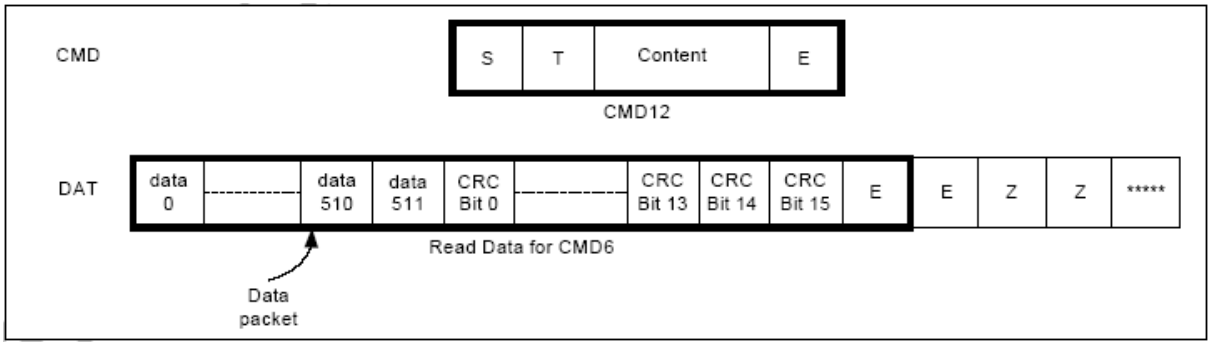


Figure 6-12 : CMD12 during CMD6; Case 2

Application Note: The host is advised not to issue CMD12 during CMD6 transaction

- **Switch function flow example**

Application Note: The host is recommended to take the following flow for switching the function.

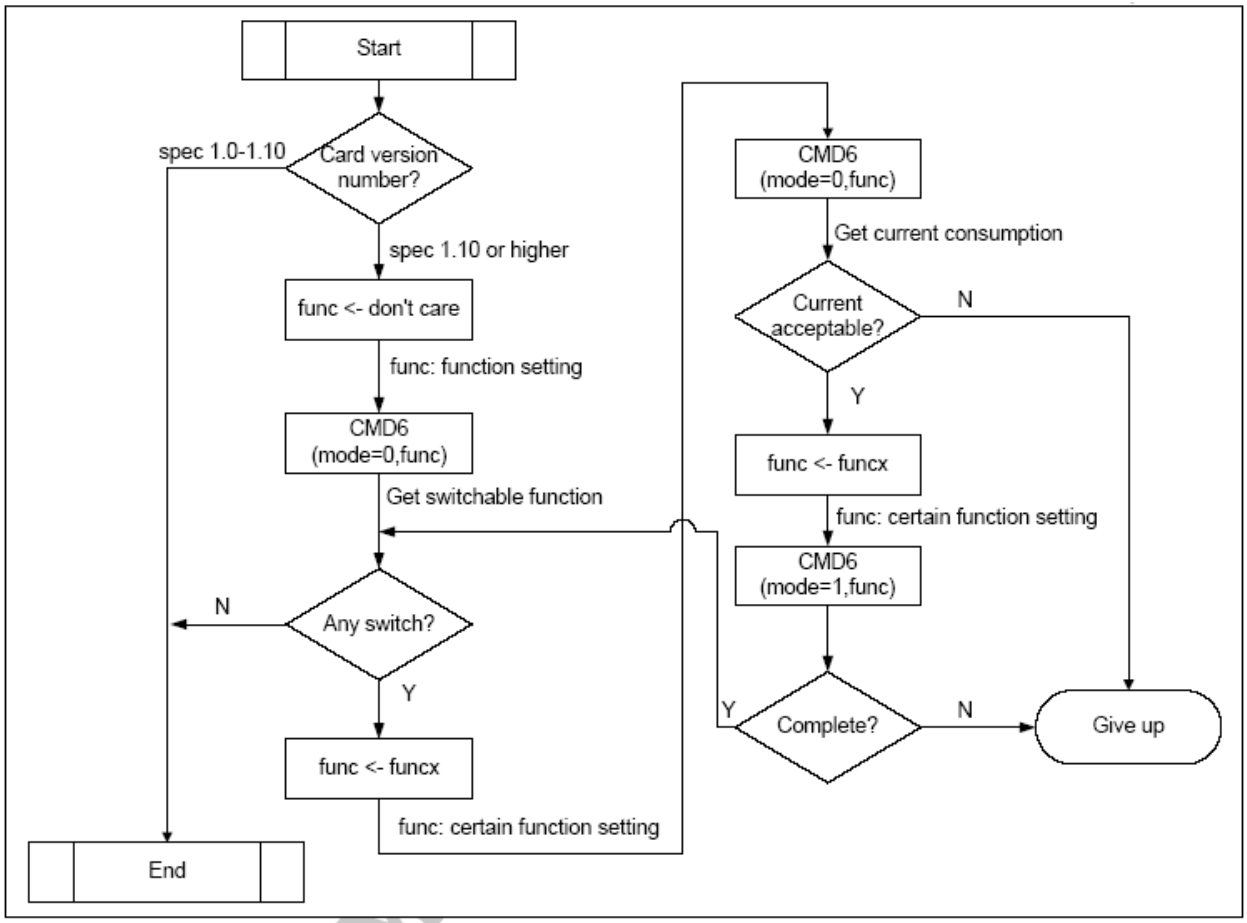


Figure 6-13 : Switching function flow

6.4.12 High-Speed mode (25MB/sec interface speed)

Though the Rev 1.01 SD memory card supports up to 12.5MB/sec interface speed, the speed of 25MB/sec is necessary to support increasing performance needs of the host and because of memory size which continues to grow.
To achieve 25MB/sec interface speed, clock rate is increased to 50MHz and CLK/CMD/DAT signal timing and circuit conditions are reconsidered and changed from Physical Layer Specification Version 1.01.
After power up, the SD memory card is in the default speed mode, and by using Switch Function command(CMD6)), the rev 1.10 (and greater) SD memory card can be placed in High-Speed mode. The High-Speed function is a function in the access mode group (see Table 6-9). Supporting High-Speed mode is optional.

Because it is not possible to control two cards or more in case that each of them have a different timing mode (Default and High-Speed mode) and in order to satisfy severe timing, host shall drive only one card. CLK/CMD/DAT signal shall be connected in 1 to 1 between the host and the card.
Maximum current consumption for SD memory cards operating in High-Speed mode is 200mA (average over 1 second).
VDD_R_CURR_MIN, VDD_W_CURR_MIN, VDD_R_CURR_MAX and VDD_W_CURR_MAX values in CSD register are meaningless in High-Speed mode.
In High-Speed mode, TRAN_SPEED value in CSD register is 0_1011_010b (05Ah) which is equal to 50MHz.
If the host wants to know the current consumption, the current consumption indicated in the switch function status returned by the Switch Function command (CMD6) shall be checked.

6.4.13 Command system

SD commands CMD34-37, CMD50, CMD57 are reserved for SD command system expansion via the switch command. Switching between the various functions of the command system function group, will change the interpretation and associated bus transaction (i.e. command without data transfer, single block read, multiple block write, etc.) of these commands. Supporting Command system is optional

• When the "standard command set" (default function 0x0) is selected these commands will not be recognized by the card and will be considered as illegal commands (as defined in revision 1.01 of the SD physical layer specification)
• When the “vendor specific” (function 0xE) is selected the behavior of these commands are vendor specific. They are not defined by this standard and may change for different card vendors.
• When the “mobile e-commerce” (function 0x1) is selected the behavior of these commands is governed by the SD memory card spec part A1: Mobile Commerce Extension Specification.

When either one of these extensions is used, special care should be given to proper selection of the command set function, otherwise the host command may be interpreted incorrectly.

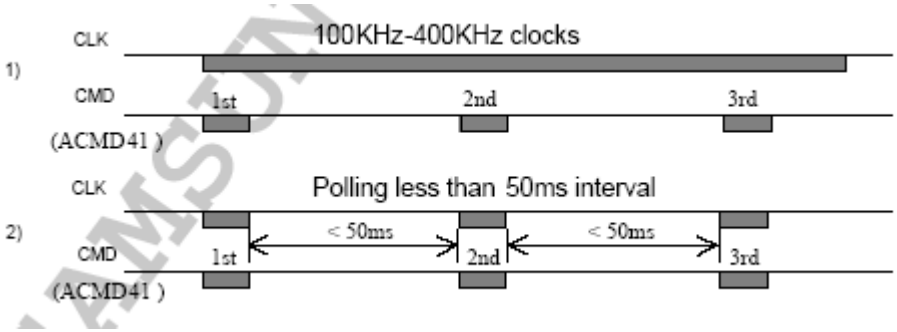
All other commands of the SD memory card (not reserved for the switch commands) are always available and will be executed as defined in this document regardless of the currently selected commands set.

6.5 Clock Control

The SD(microSD) Card bus clock signal can be used by the host to set the card into energy saving mode, or to control the data flow on the bus. The host is allowed to lower the clock frequency or shut it down.

There are a few restrictions the host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by microSD Card and the identification frequency)
- An exception to the above is ACMD41 (SD_APP_OP_CMD). After issuing command ACMD41, the following 1) or 2) procedure shall be done by the host until the card becomes ready.
  1) Issue continuous clock in frequency range of 100KHz ~ 400KHz.
  2) If the host wants to stop the clock, poll busy bit by ACMD41 command at less than 50ms intervals.



- It is an obvious requirement that the clock must be running for the card to output data or response tokens. After the last SD(microSD) Card bus transaction, the host is required, to provide **8 (eight)** clock cycles for the card to complete the operation before shutting down the clock. Following is a list of the various bus transactions:
  - A command with no response. 8 clocks after the host command end bit.
  - A command with response. 8 clocks after the card response end bit.
  – A read data transaction. 8 clocks after the end bit of the last data block.
  – A write data transaction. 8 clocks after the CRC status token.
- The host is allowed to shut down the clock of a “busy” card. The card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge the card (unless previously disconnected by a deselect command -CMD7) will force the DAT0 line down, permanently.

6.6 Cyclic Redundancy Codes (CRC)

The CRC is intended for protecting microSD Card commands, responses and data transfer against transmission errors on the microSD Card bus. One CRC is generated for every command and checked for every response on the CMD line. For data blocks one CRC per transferred block, per data line, is generated. The CRC is generated and checked as described in the following.

- **CRC7**

The CRC7 check is used for all commands, for all responses except type R3, and for the CSD and CID registers. The CRC7 is a 7-bit value and is computed as follows:

Generator polynomial $G(x) = x^7 + x^3 + 1$

$$M(x) = (\text{first bit}) \times x^n + (\text{second bit}) \times x^{n-1} + \ldots + (\text{last bit}) \times x^0$$

$$CRC[6\ldots0] = \text{Remainder}[(M(x) \cdot x^7)/G(x)]$$

The first bit is the most left bit of the corresponding bit string (of the command, response, CID or CSD). The degree *n* of the polynomial is the number of CRC protected bits decreased by one. The number of bits to be protected is 40 for commands and responses (n = 39), and 120 for the CSD and CID (n = 119).

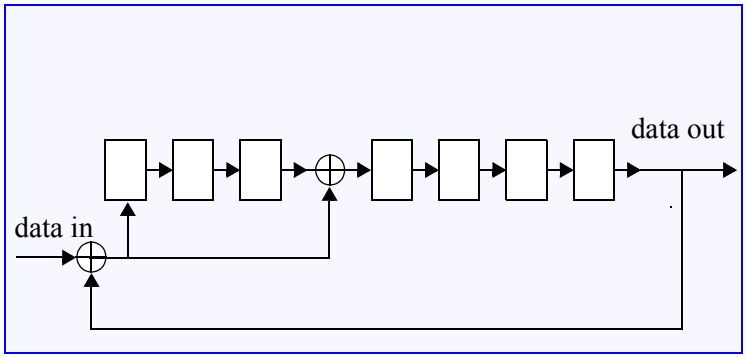


Figure 6-14 : CRC7 Generator/Checker

- **CRC16**

In case of one DAT line usage (as in MultiMediacard) than the CRC16 is used for payload protection in block transfer mode. The CRC check sum is a 16-bit value and is computed as follows:

Generator polynomial $G(x) = x^{16} + x^{12} + x^5 + 1$

$$M(x) = (\text{first bit}) \times x^n + (\text{second bit}) \times x^{n-1} + \ldots + (\text{last bit}) \times x^0$$

$$CRC[15\ldots0] = \text{Remainder}[(M(x) \cdot x^{16}) / G(x)]$$

The first bit is the first data bit of the corresponding block. The degree *n* of the polynomial denotes the number of bits of the data block decreased by one (e.g. n = 4095 for a block length of 512 bytes). The generator polynomial $G(x)$ is a standard CCITT polynomial. The code has a minimal distance d=4 and is used for a payload length of up to 2048 Bytes (n <= 16383).
The same CRC16 method shall be used in single DAT line mode and in wide bus mode. In wide bus mode, the CRC 16 is done on each line seperately.

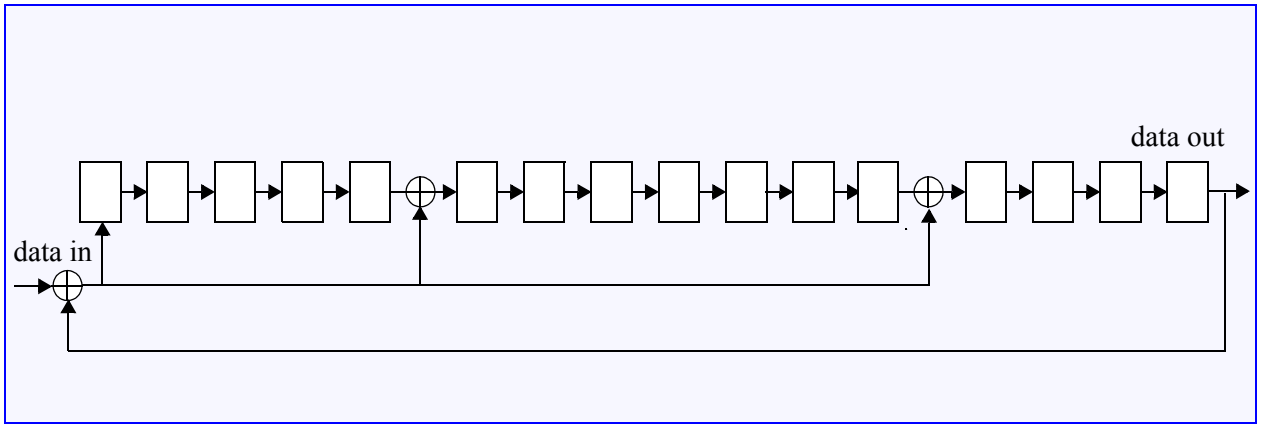


Figure 6-15 : CRC16 Generator/Checker

6.7 Error Conditions

6.7.1 CRC and Illegal Command

All commands are protected by CRC (cyclic redundancy check) bits. If the addressed card's CRC check fails, the card does not respond, and the command is not executed; the card does not change its state, and COM_CRC_ERROR bit is set in the status register.

Similarly, if an illegal command has been received, the card shall not change its state, shall not respond and shall set the ILLEGAL_COMMAND error bit in the status register. Only the non-erroneous state branches are shown in the state diagrams (see Figure 6-8 to Figure 6-9). Table 6-21 contains a complete state transition description.

There are different kinds of illegal commands:

- Commands which belong to classes not supported by the card (e.g. write commands in read only cards).
- Commands not allowed in the current state (e.g. CMD2 in Transfer State).
- Commands which are not defined (e.g. CMD5).

6.7.2 Read, Write, Erase And Erase Time-out Conditions

The times after which a time-out condition for read/write/erase operations occurs are (card independent) **either 100 times longer** than the typical access/program times for these operations given below **or 100ms (the lower of them)**. The times after which a time-out condition for Write/ Erase operations occur are (card independent) **either 100times longer** than the typical program times for these operations given below **or 250ms.** A card shall complete the command within this time period, or give up and return an error message. If the host does not get any response within the defined time-out it should assume the card is not going to respond anymore and try to recover (e.g. reset the card, power cycle, reject, etc.). The typical access and program times are defined as follows:

- **Read**

The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block.

- **Write**

The R2W_FACTOR field in the CSD is used to calculate the typical block program time obtained by multiplying the read access time by this factor. It applies to all write/erase commands (e.g. SET(CLEAR)_WRITE_PROTECT, PROGRAM_CSD(CID) and the block write commands).

- **Erase**

The duration of an erase command will be (order of magnitude) the number of write blocks to be erased multiplied by the block write delay.

6.8 Commands

6.8.1 Command Types

There are four kinds of commands defined to control the microSD Card:

* broadcast commands (**bc**), no response - The broadcast feature is only if all the CMD lines are connected together in the host. If they are separated then each card will accept it separately on his turn.
* broadcast commands with response (**bcr**) - response from all cards simultaneously. Since there is no Open Drain mode in SD(microSD) Card, this type of command is used only if all the CMD lines are separated. The command will be accepted and responded to by every card seperately.
* addressed (point-to-point) commands (**ac**) - no data transfer on DAT lines
* addressed (point-to-point) data transfer commands (**adtc**), data transfer on DAT lines

All commands and responses are sent over the CMD line of the SD(microSD) Card bus. The command transmission always starts with the left bit of the bitstring corresponding to the command code word.

6.8.2 Command Format

All commands have a fixed code length of 48 bits, needing a transmission time of 2.4us @ 20 MHz

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

Table 6-11 : Command Format

A command always starts with a start bit (always '0'), followed by the bit indicating the direction of transmission (host = '1'). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by 'x' in the above table indicates this variable is dependent on the command. All commands are protected by a CRC (see Chapter 6.6 for the definition of CRC7). Every command codeword is terminated by the end bit (always '1'). All commands and their arguments are listed in Table 6-13 - Table 6-20.

6.8.3 Command Classes

The command set of the SD(microSD) Card system is divided into several classes (See Table 6-12). Each class supports a set of card functions.

Class 0, 2, 4, 5 and 8 are mandatory and shall be supported by all SD(micro) cards. The other classes are optional. The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

	0	1	2	3	4	5	6	7	8	9	10	11
Supported Commands	Basic	Reserved	Block Read	Reserved	Block Write	Erase	Write Protection	Lock Card	Application Specific	I/O mode	Switch	Reserved
CMD0	+											
CMD2	+											
CMD3	+											
CMD4	+											
CMD6											+	
CMD7	+											
CMD9	+											
CMD10	+											
CMD12	+											
CMD13	+											
CMD15	+											
CMD16			+		+			+				
CMD17			+									
CMD18			+									
CMD24					+							
CMD25					+							
CMD27					+							
CMD28							+					
CMD29							+					
CMD30							+					
CMD32						+						
CMD33						+						
CMD38						+						
CMD42								+				
CMD55									+			
CMD56									+			
ACMD6									+			
ACMD13									+			
ACMD22									+			
ACMD23									+			
ACMD41									+			
ACMD42									+			
ACMD51									+			

Table 6-12 : Card Command Classes (CCCs)

6.8.4 Detailed Command Description

The following tables define in detail all SD(microSD) Card bus commands. The responses R1-R3, R6 are defined in Chapter 6.10 The registers CID, CSD and DSR are described in Chapter 5.5

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all cards idle state
CMD1	Reserved				
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks any card to send their CID numbers on the CMD line. (Any card that is connected to the host will respond)
CMD3	bcr	[31:0] stuff bits	R6	SEND_RELATIVE_ADDR	Asks the card to publish a new relative address (RCA)
CMD4[1]	Not Supported				
CMD5	Reserved for I/O cards (refer to "SDIO Card Specification")				
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b (only from the selected card)	SELECT/ DESELECT_CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases the card is selected by its own relative address and deselected by any other address; address 0 deselects all the card. When RCA equals 0, the host may do one of the following: -Use other RCA number to perform card deselection -Resend CMD3 to change its RCA number to other then 0 and then use CMD7 with RCA=0 for card deselection.
CMD8	Reserved				
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card identification (CID) on the CMD line.
CMD11	Reserved				
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Foreces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.

Table 6-13 : Basic commands (class 0)

1) The DSR option (as well as SET_DSR command) is not supported by the Samsung microSD Card.

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is fixed 512Bytes. If block length is set bigger than 512Bytes, the card will set the BLOCK_LEN_ERROR bit. Supported only if Partial block RD/WR operation are allowed in CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command. [1]
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command
CMD19-CMD23	Reserved				

Table 6-14 : Block Oriented Read Commands (Class 2)

1) The data transfer must not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command [2]
CMD25	adtc	[31:0] data address	R1	WRITE_MULTI_BLOCK	Continuously writes blocks of data until a STOP_TRANSIMISSION follows.
CMD26	Reserved for Manufacturer				
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programing of the programmable bits of the CSD.

Table 6-15 : Block Write Commands (Class 4)

2) The data transferred must not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD. In case that write partial blocks is not supported then the block length=default blcok length (given in CSD)

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits. [3]
CMD31	Reserved				

Table 6-16 : Block oriented Write Protection commands (Class 6)

3)32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (leaset significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits shall be set to zero

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK_START	Sets the address of the first write block to be erased.
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK_END	Sets the address of the last write block of the continuous range ot be erased.
CMD38	ac	[31:0] data address	R1b	ERASE	Erases all previously selected write blocks
CMD39	Reserved				
CMD40					Non Valid in SD Memory Card - Reserved for MultiMediaCard I/O mode
CMD41	Reserved				

Table 6-17 : Erase Commands (Class 5)

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD42	adtc	[31:0] stuff bits.	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43-49 CMD54	Reserved				

Table 6-18 : Lock Card (Class 7)

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command
CMD56	adtc	[31:1] stuff bits [0] RD/WR[1]	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block shall be set by the SET_BLOCK_LEN command.
CMD58-59	Reserved				
CMD60-63	Reserved for manufacturer				

Table 6-19 : Application Specific Commands (Class 8)

1) RD/WR: "1 = the host gets a block of data from the card.
"0= the host sends a block of data to the card

Table 6-12 describes all the application specific commands supported/ reserved by the SD(microSD) Card. All the following ACMDs shall be preceeded with APP_CMD command (CMD55)

ACMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
ACMD6	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width (’00=1bit or ’10=4bits bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the microSD Card status. The status fields are given in Table 6-27
ACMD17	Reserved				
ACMD18	-	-	-	-	Reserved for SD security applications.[1]
ACMD19 to ACMD21	Reserved				
ACMD22	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_BLOCKS	Send the number of the written (without errors) write blocks. Respond with 32bit + CRC data block. If WRITE_BL_PARTIAL=’0’, the unit of ACMD22 is always 512byte. If WRITE_BL_PARTIA=’1’, the unit of ACMD22 is a block length which was used when the write command was executed.
ACMD23	ac	[31"0} stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Mulitple Block WR command). "1 =default (one wr block)[2]
ACMD24	Reserved				
ACMD25	-	-	-	-	Reserved for SD security applications.[1]
ACMD26	-	-	-	-	Reserved for SD security applications.[1]
ACMD38	-	-	-	-	Reserved for SD security applications.[1]
ACMD39 to ACMD40	Reserved				
ACMD41	bcr	[31:0] OCR without busy	R3	SD_SEND_OP_COND	Asks the accessed card to send its operating condition register (OCR) content in response on the CMD line.
ACMD42	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin1) of the card. The pull-up may be used for card detection.

Table 6-20: Application Specific Commands Used/ Reserved by SD Card

1) Refer to *SD Card Security Specification* for detailed explanation about the SD Security Features
2) Command STOP_TRAN (CMD12) shall be used to stop the transmission in Write Mulitple Block whether the preerase (ACMD23) feature is used or not.

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD6	adtc	[31] Mode 0:Check function 1:Switch function [30:24] reserved (All '0') [23:20] reserved for function group 6 (All '0' or 0xF) [19:16] reserved for function group 5 (All '0' or 0xF) [15:12] reserved for function group 4 (All '0' or 0xF) [11:8] reserved for function group 3 (All '0' or 0xF) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Checks switchable function (mode0) and switch card function (mode1).
CMD34	Reserved for each command system set by switch function command (CMD6). Detail definition is refer to each command system specification.				
CMD35					
CMD36					
CMD37					
CMD50					
CMD57					

Table 6-20: Switch function commands (class 10)

6.9 Card State Transition

Table 6-13 defines the card state transitions in dependency of the received command.

	Current State									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
Trigger of state change	**Changes to**									
Class Independent										
"Operation Complete"	-	-	-	-	-	-	-	tran	stby	-
Class 0										
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	-
CMD2	-	ident	-	-	-	-	-	-	-	-
CMD3	-	-	stby	stby	-	-	-	-	-	-
CMD4	-	-	-	stby	-	-	-	-	-	-
CMD7, card is addressed	-	-	-	tran	-	-	-	-	prg	-
CMD7, card is not addressed	-	-	-	stby	stby	stby	-	dis	-	-
CMD9	-	-	-	stby	-	-	-	-	-	-
CMD10	-	-	-	stby	-	-	-	-	-	-
CMD12	-	-	-	-	-	tran	prg	-	-	-
CMD13	-	-	-	stby	tran	data	rcv	prg	dis	-
CMD15	-	-	-	ina	ina	ina	ina	ina	ina	-
Class 2										
CMD16	-	-	-	-	tran	-	-	-	-	-
CMD17	-	-	-	-	data	-	-	-	-	-
CMD18	-	-	-	-	data	-	-	-	-	-
Class 4										
CMD16	See Class2									
CMD24	-	-	-	-	rcv	-	-	-	-	-
CMD25	-	-	-	-	rcv	-	-	-	-	-
CMD27	-	-		-	rcv	-	-	-	-	-
Class 6										
CMD28	-	-	-	-	prg	-	-	-	-	-
CMD29	-	-	-	-	prg	-	-	-	-	-
CMD30	-	-	-	-	data	-	-	-	-	-
Class 5										
CMD32	-	-	-	-	tran	-	-	-	-	-
CMD33	-	-	-	-	tran	-	-	-	-	-
CMD38	-	-	-	-	prg	-	-	-	-	-

Table 6-21: Card State Transition Table

	Current State									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
Class 7										
CMD42					rcv					
Class 8										
CMD55	idle	-	-	stby	tran	data	rcv	prg	dis	-
CMD56; RD/WR = 0	-	-	-	-	rcv	-	-	-	-	-
CMD56; RD/WR = 1	-	-	-	-	data	-	-	-	-	-
ACMD6	-	-	-	-	tran	-	-	-	-	-
ACMD13	-	-	-	-	data	-	-	-	-	-
ACMD22	-	-	-	-	data	-	-	-	-	-
ACMD23	-	-	-	-	tran	-	-	-	-	-
ACMD18,25,26,38,43,44,45,46,47,48,49	Refer to SD Card Security Specification for an explanation of the SD Security Features. The Samsung microSD Card supports all the security related commands as explained in the specification.									
ACMD41, card V_{DD} range compatilbe	ready	-	-	-	-	-	-	-	-	-
ACMD41, card is busy	idle	-	-	-	-	-	-	-	-	-
ACMD41, card V_{DD} range not compatilbe	ina	-	-	-	-	-	-	-	-	-
ACMD42	-	-	-	-	tran	-	-	-	-	-
ACMD51	-	-	-	-	data	-	-	-	-	-
class 9										
CMD52-CMD54	Refer to "SDIO Card Specification"									
class 10										
CMD6	-	-	-	-	data	-	-	-	-	-
CMD34-37, 50, 57	-	-	-	-	tran	-	-	-	-	-
class 11										
CMD41: CMD43...CMD49, CMD58-CMD59	Reserved									
CMD60...CMD63	Reserved for manufacturer									

Table 6-21 : Card State Transition Table

6.10 Responses

All responses are sent via the command line CMD. The response transmission always starts with the left bit of the bit string corresponding to the response codeword. The response length depends on the response type.

A response always starts with a start bit (always '0'), followed by the bit indicating the direction of transmission (card = '0'). A value denoted by 'x' in the tables below indicates a variable entry. All responses except for the type R3 (see below) are protected by a CRC. Every response is terminated by the end bit (always '1').

There are four types of responses that are supported in the SD(microSD) Card. Their formats are defined as follows:

• **R1** (normal response command): response length 48 bit. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that when a data transfer to the card is involved, a busy signal may appear on the data line after the transmission of each block of data. The host will check for busy after the data block transmission.

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	x	x	x	'1'
Description	start bit	transmission bit	command index	card status	CRC7	end bit

Table 6-22 : Response R1

• **R1b** is identical to R1 with an optional busy signal transmitted on the data line DAT0. The card may become busy after receiving these commands based on its state prior to the command reception. The Host shell check for busy at the response.
• **R2** (CID, CSD register): code length 136 bits. The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127...1] of the CID and CSD are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response.

Bit position	135	134	[133:128]	[127:1]	0
Width (bits)	1	1	6	127	1
Value	'0'	'0'	'111111'	x	'1'
Description	start bit	transmission bit	reserved	CID or CSD register incl. internal CRC7	end bit

Table 6-23 : Response R2

• **R3** (OCR register): code length 48 bits. The contents of the OCR register is sent as a response to ACMD41.

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'0'	'111111'	x	'1111111'	'1'
Description	start bit	transmission bit	reserved	OCR register	reserved	end bit

Table 6-24 : Response R3

• **R6** (Published RCA response): code length 48 bits. The bits 45:40 indicate the index of the command to responded to - in that case it will be '000011 (together with bit 5 in the status bits it means = CMD3). The 16 MSB bits of the argument field are used for the published RCA number.

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	transmission bit	Command index ('000011)	New published RCA [31:6] of the card	[15:0] card status bits: 23,22,19,12:0 (see Table 6-26)	CRC7	end bit

Table 6-25 : Response R6

6.11 microSD Card Status

6.11.1 Card Status

The response format R1 contains a 32bit field named *card status.* This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command. The SD(microSD) Card supports the following two card status fiels:

- Card Status - This status filed is compatible to the MultiMediaCard protocol.
- SD_Status - This extended status field o f 512 bits supports special features unique to the microSD Card and future application specific features

The microSD Card status register's structures are defined in Table 6-18. The Type and Clear-Condition fields in the table are coded as follows:

Type:
- E : Error bit
- S : Status bit
- R : Detected and set for the actual command response.
- X : Detected and set during command execution. The host must poll the card by issuing the status command in oreder to read these bits.

Clear Condition:
- A : According to the card current state
- B : Always related to the previous command. Reception of a valid command will clear it. (with a delay of one command).
- C: Celar by read

Bits	Identifier	Type	Value	Descript	Clear Cond
31	OUT_OF_RANGE	E R X	'0'= no error '1'= error	The command's address argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	E R X	'0'= no error '1'= error	A misaligned address that did not match the blcok length was used in the command	C
29	BLOCK_LEN_ERROR	E R X	'0'= no error '1'= error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	E R	'0'= no error '1'= error	An error in the sequence of erase commands occurred	C
27	ERASE_PARAM	E R X	'0'= no error '1'= error	An invalid selection of write-blocks for erase occurred	C
26	WP_VIOLATION	E R X	'0'= no protected '1'= protected	Attempt to program a write-protected block	C
25	CARD_IS_LOCK	S X	'0'= card unlocked '1'= card locked	When set, signals that the card is locked by the host	A
24	LOCK_UNLOCK_FAILED	E R X	'0'= no error '1'= error	Set when a sequence or password error has been detected in lock/ unlock card command	C
23	COM_CRC_ERROR	E R	'0'= no error '1'= error	The CRC check of the previous command field	B

Table 6-26 : Card Status

Bits	Identifier	Type	Value	Descript	Clear Cond
22	ILLEGAL_COMMAND	E R	'0'= no error '1'= error	Command not legal for the card state	B
21	CARD_ECC_FAILED	E R X	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	E R X	'0'= no error '1'= error	Internal card controller error	C
19	ERROR	E R X	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	Reserved				
17	Reserved				
16	CSD_OVERWRITE	E R X	'0'= no error '1'= error	Can be either one of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
15	WP_ERASE_SKIP	S X	'0'= not protected '1'= protected	Only partial address space was erased due to existing write protected blocks.	C
14	CARD_ECC_DISABLED	S X	'0'= enabled '1'= disabled	The command has been excuted without using the internal ECC	A
13	ERASE_RESET	E R	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
12:9	CURRENT_STATE	S X	0 = Idle 1 = Ready 2 = Ident 3 = Stby 4 = Tran 5 = Data 6 = Rcv 7 = Prg 8 = Dis 9-15 = reserved	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response on the next command. The four bits are interpreted as a binary number between 0 and 15.	B
8	READY_FOR_DATA	S X	'0'= not ready '1'= ready	Corresponds to buffer empty signalling on the bus	A
7: 6					
5	APP_CMD	S R	'0'= disabled '1'= enabled	The card will expect ACMD, or indication that the command has been interpreted as ACMD	C
4	Reserved for SD I/O Card				
3	AKE_SEQ_ERROR (SD Card Security spec.)	E R	'0'= no error '1'= error	Error in the sequence of authentication process	C
2:1	Reserved for application specific commands				
1,0	Reserved for manufacturer test mode				

6.11.2 SD Status

The SD Status contains status bits that are related to the SD Memory Card proprietary features and may be used for future application specific usage. The size of the SD Status is one data block of 512bit. The content of this register is transmitted to the Host over the DAT bus along with 16bit CRC. The SD Status is sent to the host over the DAT bus if ACMD13 is sent (CMD55 followed with CMD13). ACMD13 can be sent to a card only in *'tran_state'* (card is selected). SD Status structure is described in below.

Bits	Identifier	Type	Value	Descript	Clear Cond
511: 510	DATA_BUS_WIDTH	S R	'00'=1 (default) '01'= reserved '10'=4 bit width '11'=reserved	Shows the currently defined data bus width that was defined by the SET_BUS_WIDTH command	A
509	SECURE_MODE	S R	'0'=Not in the mode '1'=in Secured mode	Card is in Secured Mode of operation(refer to the *SD Security Specifications* document)	A
508: 406	Reserved				
495: 480	SD_CARD_TYPE	S R	'00xxh'=SD Memory Cards as defined in Physical Spec. Ver.1.01 ('x'=don't care). The following cards are currently defined: '0000'=Regular SD RD/WR Card. '0001'=SD ROM Card	In the future, the 8 LSBs will be used to define different variations of a SD Card (each bit will define different SD types). The 8 MSBs will be used to define SD Cards that do not comply with the SD Memory Card as defined in the Specification Ver.1-1.01	A
479: 448	SIZE_OF_PROTECTED_AREA	S R	Size of protected area (in units of MULT*BLOCK_LEN refer to CSD register	Shows the size of the protected area. The actual area= (SIZE_OF_PROTECTED_AREA)*MULT*BLOCK_LEN	A
447: 312	Reserved				
311:0	Reserved for Manufacturer				

Table 6-27 : mniSD Card Status

6.12 Memory Array Partitioning

The basic unit of data transfer to/from the microSD Card is one byte. All data transfer operations which require a block size always define block lengths as integer multiples of bytes. Some special functions need other partition granularity.

For block oriented commands, the following definition is used:

- **Block**: is the unit which is related to the block oriented read and write commands. Its size is the number of bytes which will be transferred when one block command is sent by the host. The size of a block is either programmable or fixed. The information about allowed block sizes and the programmability is stored in the CSD.

The granularity of the erasable units is in general not the same as for the block oriented commands:

- **Sector** : The unit that is related to the erase commands, Its size is the number of blocks that are erased in one operation. The size of a sector is fixed for each device. The information about the sector size (in blocks)is stored in CSD.

For devices that include write protection, the following definition is used.

- **WP-Group :** The minimal unit that may have individual write protection. Its size is the number of groups which will be write protected by one bit. The size of a WP-group is fixed for each device. The information about the size is sotred in the CSD.

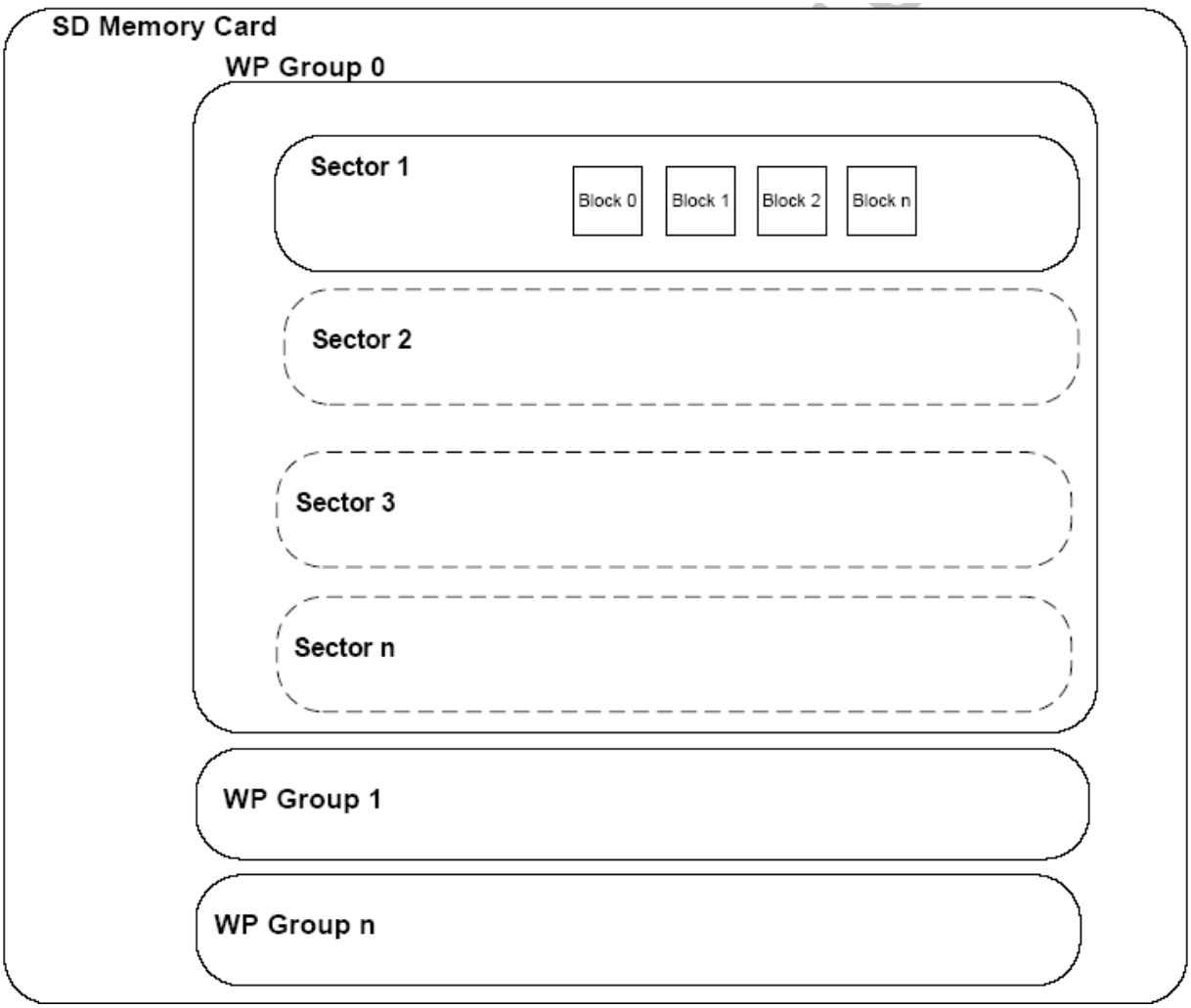


Figure 6-16 : Write Protection Hierarchy

Each WP-group may have an additional write protection bit. The write protection bits are programmable via special commands.

Both functions are optional and only useful for writable/erasable devices. The write protection may also be useful for multi type cards (e.g. a ROM - Flash combination). The information about the availability is stored in the CSD.

6.13 Timing Diagrams

All timing diagrams use the following schematics and abbreviations:

Symbol	Definition
S	Start bit (= ‘0’)
T	Transmitter bit (Host = ‘1’, Card = ‘0’)
P	One-cycle pull-up (= ‘1’)
E	End bit (=’1’)
Z	High impedance state (-> = ‘1’)
X	Don’t Care Data Bits (from Card)
D	Data bits
*	Repetition
CRC	Cyclic redundancy check bits (7 bits)
	Card active
	Host active

Table 6-28 : Timing Diagram Symbols

The difference between the P-bit and Z-bit is that a P-bit is actively driven to HIGH by the card respectively host output driver, while Z-bit is driven to (respectively kept) HIGH by the pull-up resistors R_{CMD} respectively R_{DAT}. Actively-driven P-bits are less sensitive to noise. All timing values are defined in Table 6-29.

6.13.1 Command and Response

Both host command and card response are clocked out per timing specified in Chapter 5.4.5 (and Chapter 5.4.6 for high speed card)

- **Card identification and card operation conditions timing**

The timing for CMD2 and ACMD41 is below. The command is followed by a period of two Z bits (allowing time for direction switching on the bus) and then by the responding card. The card response to the host command starts after N_{ID} clock cycles.

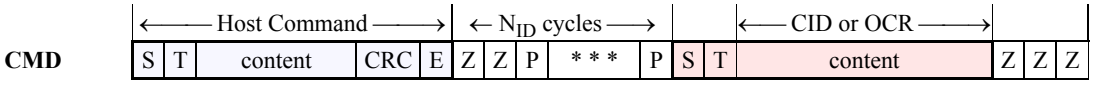


Figure 6-17 : Identification Timing (Card Identification Mode)

- **Assign a card relative address**

The SEND_RELATIVE_ADDR (CMD 3) for SD(microSD) Card timing is given below. Note that CMD3 command’s content, functionality and timing are different for MultiMediaCard. The minimum delay between the host command and card response is N_{CR} clock cycles.

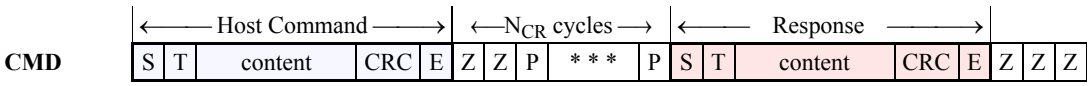


Figure 6-18 : SET_RELATIVE_ADDR Timing

- **Data transfer mode.**

After a card receives its own RCA it will switch to data transfer mode. The command is followed by a period of two Z bits (allowing time for direction switching on the bus) and than by P bits pushed up by the responding card. This timing diagram is relevant for all responded host commands except ACMD41 and CMD2:

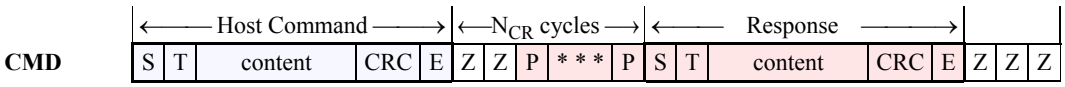


Figure 6-19 : Command Response Timing (Data Transfer Mode)

- **Last Card Response - Next Host Command Timing**

After receiving the last card response, the host can start the next command transmission after at least N_{RC} clock cycles. This timing is relevant for any host command.

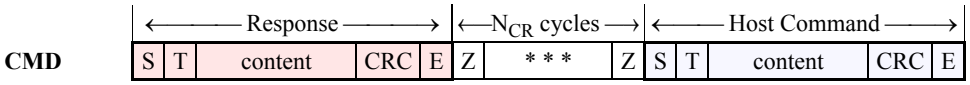


Figure 6-20 : Timing Response End To Next Command Start (Data Transfer Mode)

- **Last Host Command - Next Host Command Timing**

After the last command has been sent, the host can continue sending the next command after at least N_{CC} clock periods.

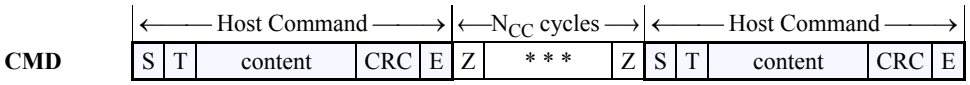


Figure 6-21 : Timing Of Command Sequences (All Modes)

6.14 Data Read

Note that the DAT line represents the data bus (either 1 or 4 bits).

- **Single Block Read**

The host selects one card for data read operation by CMD7, and sets the valid block length for block oriented data transfer by CMD16. The basic bus timing for a read operation is given in Figure 6-22. The sequence starts with a single block read command (CMD17) which specifies the start address in the argument field. The response is sent on the CMD line as usual.

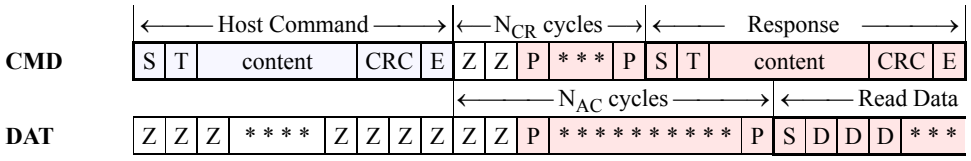


Figure 6-22 : Timinig of Single Block Read

Data transmission from the card starts after the access time delay N_{AC} beginning from the end bit of the read command. After the last data bit, the CRC check bits are suffixed to allow the host to check for transmission errors.

- **Multiple Block Read**

In multiple block read mode, the card sends a continuous flow of data blocks following the initial host read command. The data flow is terminated by a stop transmission command (CMD12). Figure 6-23 describes the timing of the data blocks and Figure 6-24 the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.

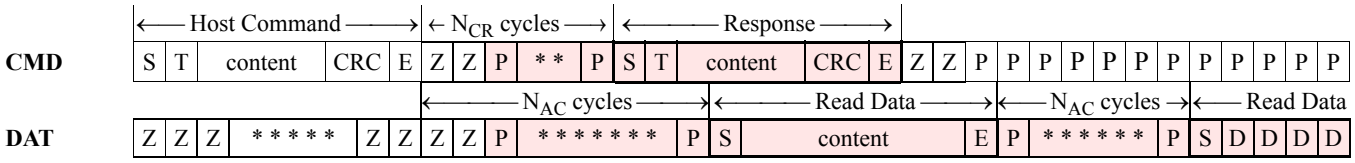


Figure 6-23 : Timing of Multiple Block Read

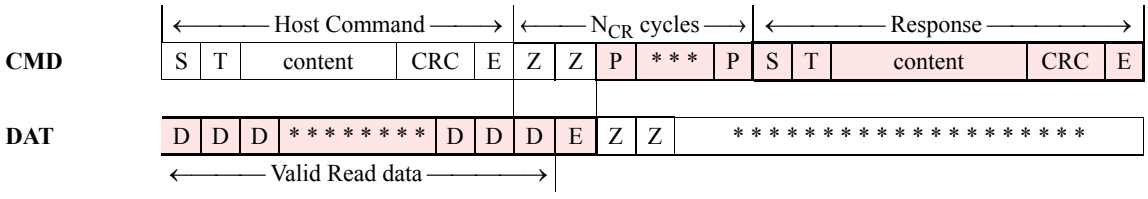


Figure 6-24 : Stop Command Timing (CMD12, Data Transfer Mode)

6.15 Data Write

- **Single Block Write**

The host selects the card for data write operation by CMD7. The host sets the valid block length for block oriented data transfer (a stream write mode is also available) by CMD16.

The basic bus timing for a write operation is given in Figure 6-25. The sequence starts with a single block write command (CMD24) which determines (in the argument field) the start address. It is responded by the card on the CMD line as usual. The data transfer from the host starts N_{WR} clock cycles after the card response was received.
The data is suffixed with CRC check bits to allow the card to check it for transmission errors. The card sends back the CRC check result as a CRC status token on DAT0. In the case of transmission error the card sends a negative CRC status (‘101’) on DAT0. In the case of non-erroneous transmission, the card sends a positive CRC status (‘010’) on DAT0 and starts the data programming procedure. When a flash programing error occurs the card will ignore all further data blocks. In this case no CRC response will be sent to the host, therefore, there will not be CRC start bit on the bus and the three CRC status bits will read (’111’).

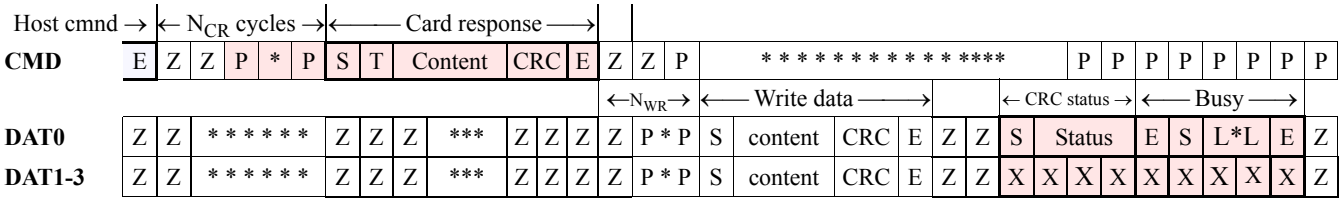


Figure 6-25 : Timing of Block Write Command

Note that the CRC response output is always two clocks after the end of data.

If the card does not have a free data receive buffer, the card indicates this condition by pulling down the data line to LOW. The card stops pulling down DAT0 as soon as at least one receive buffer for the defined data transfer block length becomes free. This signalling does not give any information about the data write status which must be polled by the host.

- **Multiple Block Write**

In multiple block write mode, the card expects continuous flow of data blocks following the initial host write command.

As in the case of single block write, the data is suffixed with CRC checks bits to allow the card to check it for transmission errors. The cards sends back the CRC check result as a CRC status token on the DAT0 line. In the case of transmission error the card sends a negative CRC status (’101’). In the case of non-erroneous transmission the card sends a positive CRC status (’010’) and starts the data programming procedure. When a flash programming error occurs the card will ignore all further data blocks. In this case no CRC response will be sent to the host and, therefore, there will not be CRC start bit on the bus and the three CRC status bits will read (’111’).

The data flow is terminated by a stop transmission command (CMD12). Figure 6-26 describes the timing of the data blocks with and without card busy signal.

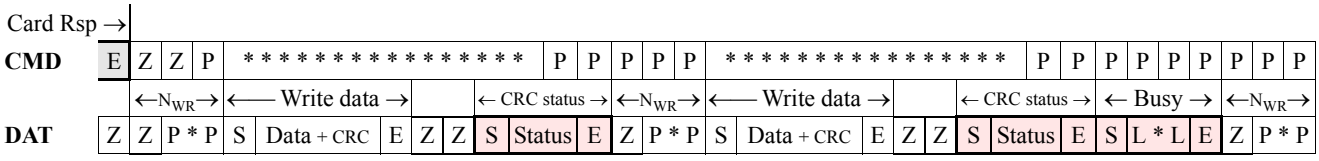


Figure 6-26 : Timing of Multiple Block Write

The stop transmission command works similar as in the read mode. Figure 6-27 to Figure 6-29 describe the timing of the stop command in different card states.

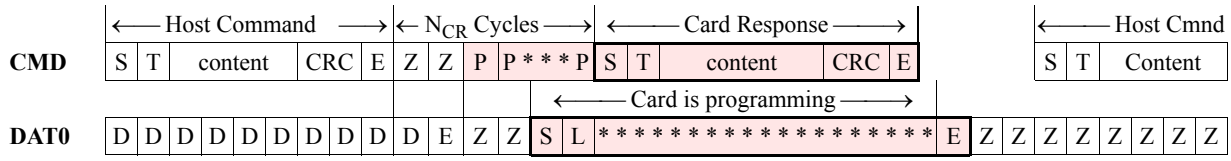
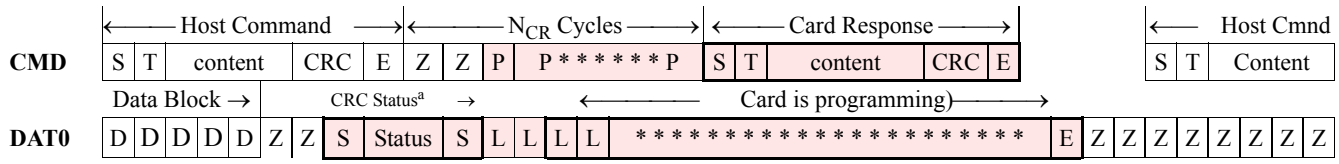


Figure 6-27 : Stop Transmission During Data Transfer From The Host

The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status tokens sent back to the host. Figure 6-28 is an example of an interrupted (by a host stop command) attempt to transmit the CRC status block. The sequence is identical to all other stop transmission examples. The end bit of the host command is followed, on the data lines, with one more data bit, an end bit and two Z clocks for switching the bus direction. The received data block, in this case is considered incomplete and will not be programmed.



a.The host interrupted the card's CRC status response.

Figure 6-28 : Stop Transmission During CRC Status Transfer From The Card

All previous examples dealt with the scenario of the host stopping the data transmission during an active data transfer. The following two diagrams describe a scenario of receiving the stop transmission between data blocks. In the first example the card is busy programming the last block while in the second the card is idle. However, there are still unprogrammed data blocks in the input buffers. These blocks are being programmed as soon as the stop transmission command is received and the card activates the busy signal.

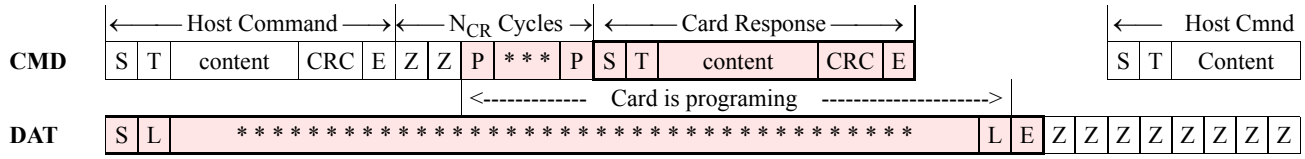


Figure 6-29 : Stop Transmission received after last data block. Card Is Busy Programming.

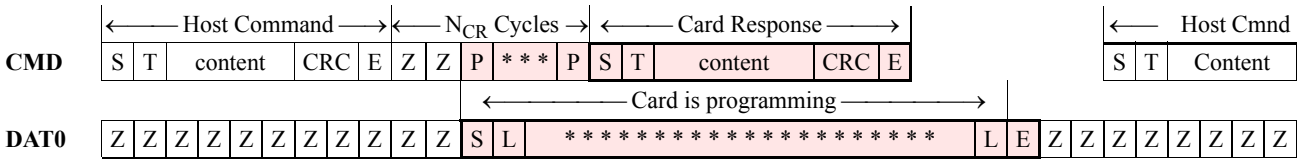


Figure 6-30 : Stop Transmission received after last data Block. Card Becomes Busy.

- **Erase, Set and Clear Write Protect Timing**

The host must first tag the start (CMD32) and end (CMD33) addressed of the range to be erased. The erase command (CMD38), once issued, will erase all selected erase groups. Similarly, set and clear write protect commands start a programming operation as well. The card will signal “busy” (by pulling the DAT0 line low) for the duration of the erase or programming operation. The bus transaction timings are the same as given for stop tran command in Figure 6-30.

- **Reselecting a busy card**

When a busy card which is currently in the dis state is reselected it will reinstate its busy signaling on the data line. The timing diagram for this command / response / busy transaction is the same as given for stop tran command in Figure 6-30.

6.15 Timing Values

	Min	Max	Unit
N_{CR}	2	64	clock cycles
N_{ID}	5	5	clock cycles
N_{AC}[1]	2	-	clock cycles
N_{RC}	8	-	clock cycles
N_{CC}	8	-	clock cycles
N_{WR}	2	-	clock cycles

Table 6-29 : Timing Parameters

1) The maximum read access time shall be calculated by host as follows:
Nac(max) = 100 ((TAAC*fpp) + (100*NSAC)) ;
fpp is the interface clock rate and TAAC & NSAC are given in the CSD (Chapter 5.5.3)

7. SPI Mode

7.1 Introduction

The SPI mode consists of a secondary communication protocol which is offered by Flash-based microSD Cards. This mode is a subset of the SD Memory Card protocol, designed to communicate with a SPI channel, commonly found in Motorola's (and lately a few other vendors') microcontrollers. The interface is selected during the first reset command after power up (CMD0) and cannot be changed once the part is powered on.

The SPI standard defines the physical link only, and not the complete data transfer protocol. The microSD Card SPI implementation uses a subset of the SD Memory Card protocol and command set. The advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum. The disadvantage is the loss of performance of the SPI mode versus SD Bus mode (e.g. Single data line and hardware CS signla per card).

7.2 SPI Bus Protocol

While the microSD Card channel is based on command and data bit streams which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of 8-bit bytes and is byte aligned to the CS signal (i.e. the length is a multiple of 8 clock cycles).
Similar to the SD Bus protocol, the SPI messages consist of command, response and data-block tokens (see Chapter 3 for a detailed description). All communication between host and card is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.
The response behavior in the SPI mode differs from the SD Bus mode in the following three aspects:

- The selected card always responds to the command.
- Additional (8 bit) response structures are used
- When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than by a time-out, as in the SD Bus mode.

In addition to the command response, every data block sent to the card during write operations will be responded to with a special data response token. A data block may be as big as one card write block and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

7.2.1 Mode Selection

The SD(microSD) Card wakes up in the SD Bus mode. It will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0) and the card is in *Idle* state. If the card recognizes that the SD Bus mode is required , it will not respond to the command and remain in the SD Bus mode. If SPI mode is required, the card will switch to SPI and respond with the SPI mode R1 response.

The only way to return to the SD Bus mode is by entering the power cycle. In SPI mode, the microSD Card protocol state machine is not observed. All the microSD Card commands supported in SPI mode are always available.

During the initialization sequence, if the host gets illegal Command indication for ACMD41 sent to the card, it may assume that the card is MultiMediaCard. In that case it should re-start the card as MultiMediaCard using CMD0 and CMD1.

7.2.2 Bus Transfer Protection

Every SD(microSD) Card token transferred on the bus is protected by CRC bits. In SPI mode, the microSD Card offers a non-protected mode which enables systems built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.
In the non-protected mode, the CRC bits of the command, response and data tokens are still required in the tokens. However, they are defined as 'don't care' for the transmitter and ignored by the receiver.
The SPI interface is initialized in the non-protected mode. However, the RESET command (CMD0), which is used to switch the card to SPI mode, is received by the card while in SD Bus mode and, therefore, must have a valid CRC field. Since CMD0 has no arguments, the content of all the fields, including the CRC field, are constants and need not be calculated in run time. A valid reset command is:

0x40, 0x0, 0x0, 0x0, 0x0, 0x95

The host can turn the CRC option on and off using the CRC_ON_OFF command (CMD59).

7.2.3 Data Read

The SPI mode supports single and multiple block read operations (CMD17 or COMD18 in the SD(microSD) Card portocol). Upon reception of a valid read command the card will respond with a response token followed by a data token of the length defined in a previous SET_BLOCKLEN (CMD16) command. (refer to Figure 7-1)

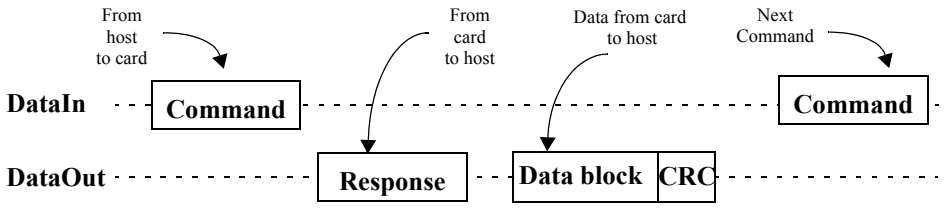


Figure 7-1 : Single Block Read Operation

A valid data block is suffixed with a 16 bit CRC generated by the standard CCITT polynomial $X^{16} + X^{12} + X^5 + 1$.

The maximum block length is given by READ_BL_LEN, defined in the CSD. If partial blocks are allowed (i.e. the CSD parameter READ_BL_PARTIAL equals 1), the block length can be any number between 1 and the maximum block size. Otherwise, the only valid block length for data read is given by READ_BL_LEN.
The start address can be any byte address in the valid address range of the card. Every block, however, must be contained in a single physical card sector.
In case of a data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 7-2 shows a data read operation which terminated with an error token rather than a data block.

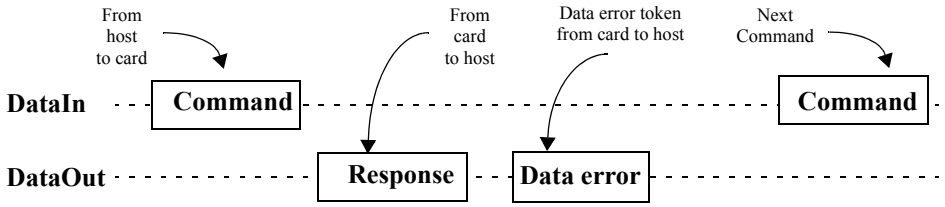


Figure 7-2 : Read Operation - Data Error

In case of Multiple block read operation every transferred block has its suffixed of 16 bit CRC.
Stop transmission command (CMD12) will actually stop the data transfer operation (the same as in SD Bus mode).

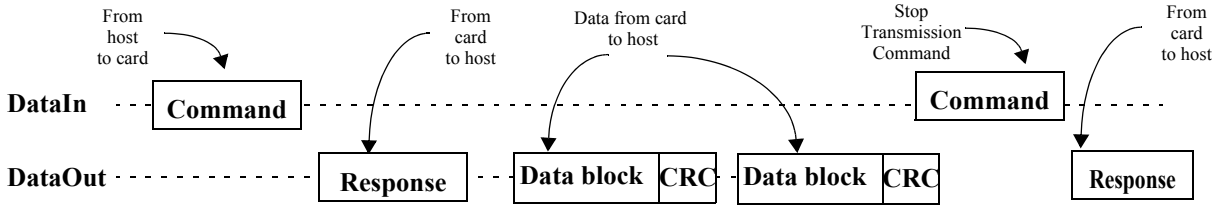


Figure 7-3 : Multiple Block Read Operation

7.2.4 Data Write

In SPI mode the SD(microSD) Card supports single block and Multiple block write commands. Upon reception of a valid write command (CMD24 or CMD25 in the microSD protocol), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix, block length and start address restrictions are (with the exception of the CSD parameter WRITE_BL_PARTIAL controlling the partial block write option) identical to the read operation (see Figure 7-4).

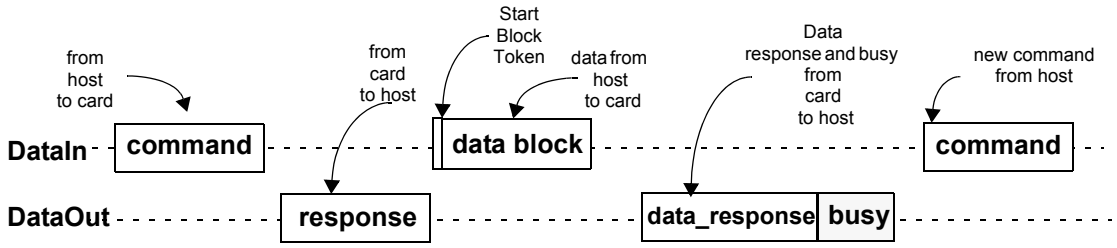


Figure 7-4 : Single Block Write Operation

Every data block has a prefix of 'Start Block' token (one byte).
After a data block has been received, the card will respond with a data-response token. If the data block has been received without errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the DataOut line low).
Once the programing operation is completed, the host must check the results of the programming using the SEND_STATUS command (CMD13). Some errors (e.g. address out of range, write protection violation etc.) are detected during programming only. The only validation check performed on the data block and communicatted to the host via the data-response token is the CRC and general Write Error indication.

In Multiple Block write operation the stop transmission will be done by sending 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block. In case of Write Error indication (on the data response) the host shall use SEND_NUM_WR_BLOCKS (ACMD22) in order to get the number of well written write blocks. The data tokens description is given in Chapter 7.5

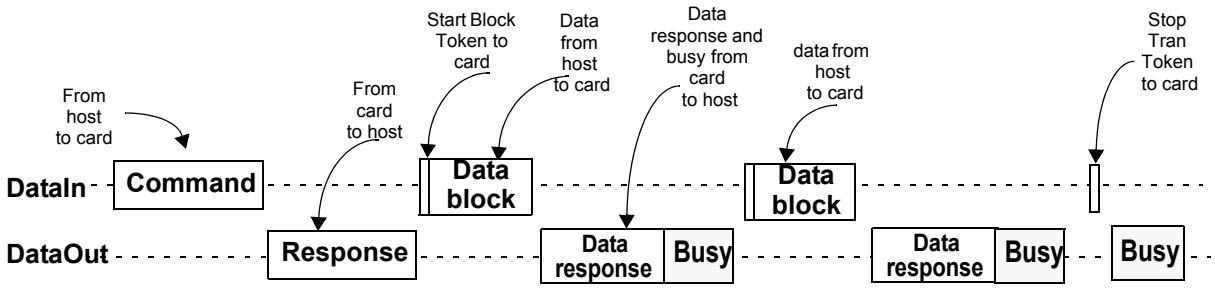


Figure 7-5 : Multiple Block Write Operation

While the card is busy, resetting the CS signal will not terminate the programming process. The card will release the DataOut line (tri-state) and continue with programming. If the card is reselected before the programming is finished, the DataOut line will be forced back to low and all commands will be rejected.
Resetting a card (using CMD0) will terminate any pending or active programming operation. This may destroy the data froms on the card. It is in responsibility of the host to prevent it.

7.2.5 Erase & Write Protect Management

The erase and write protect management procedures in the SPI mode are identical to those of the SD Bus mode. While the card is erasing or changing the write protection bits of the predefined erase groups list, it will be in a busy state and hold the DataOut line low. Figure 7-6 illustrates a 'no data' bus transaction with and without busy signalling.

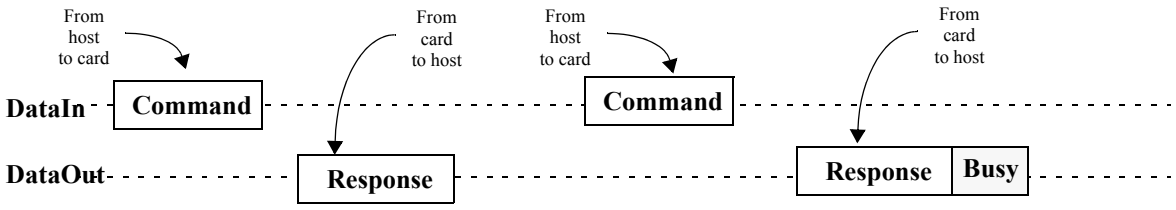


Figure 7-6 : 'No data' Operations

7.2.6 Read CID/CSD Registers

Unlike the SD Bus protocol (where the register contents is sent as a command response), reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token followed by a data block of 16 bytes suffixed with a 16 bit CRC.

The data time out for the CSD command cannot be set to the cards TACC since this value is stored in the card's CSD. Therefore the standard response time-out value (N_{AC}) is used for read latency of the CSD register.

7.2.7 Reset Sequence

The SD(microSD) Card requires a defined reset sequence. After power on reset or CMD0 (software reset) the card enters an idle state. At this state the only legal host commands are CMD1 (SEND_OP_COND), ACMD41 (SD_SEND_OP_COND) , CMD59 (CRC_ON_OFF) and CMD58 (READ_OCR). For the Thick (2.1mm) SD Card - CMD1 (SEND_OP_COND) is also vaild - that means that in SPI mode CMD1 and ACMD41 have the same behavior, though the usage of CMD41 is perferable since it allows easy distinguishing between SD Memory Card nad MultiMediaCard. **For the Thin (1.4mm) SD Memory Card CMD1 (SEND_OP_COND) is illegal command during the initialization that is done after power on. After Power On, once the card accepted valid ACMD41, it will be able to accept also CMD1 even if used after re-initializing (CMD0) the card.** It was defined in such way in order to be able to distinguish between Thin SD Memory Card and MultiMediaCards (that supports CMD1 as well).
The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card has completed its initialization processes and is ready for the next command.

In SPI mode, as opposed to SD mode, ACMD41 (or CMD1 as well, for 2.1mm-SD Memory Card) has no operands and does not return the contents of the OCR register. Instead, the host can use CMD58 (available in SPI mode only) to read the OCR register. Furthermore, it is in the responsibility of the host to refrain from accessing a card that does not support its voltage range.
The usage of CMD58 is not restricted to the initializing phase only, but can be issued at any time.

7.2.8 Clock Control

The SPI bus clock signal can be used by the SPI host to put the card into energy saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to change the clock frequency or shut it down. There are a few restrictions the SPI host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the microSD Cards)
- It is an obvious requirement that the clock must be running for the microSD Card to output data or response tokens. After the last SPI bus transaction, the host is required, to provide 8 (eight) clock cycles for the card to complete the operation before shutting down the clock. Throughout this 8 clocks period the state of the CS signal is irrelevant. It can be asserted or de-asserted.

Following is a list of the various SPI bus transactions:
- A command / response sequence. 8 clocks after the card response end bit. The CS signal can be asserted or de -asserted during these 8 clocks.
- A read data transaction. 8 clocks after the end bit of the last data block.
- A write data transaction. 8 clocks after the CRC status token.
- The host is allowed to shut down the clock of a "busy" card. The microSD Card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge the microSD Card (unless previously disconnected by de-asserting the CS signal) will force the dataOut line down, permanently.

7.2.9 Error Conditions

Unlike the SD(microSD) Card protocol, in SPI mode the card will always respond to a command. The response indicates acceptance or rejection of the command. A command may be rejected in any one of the following case:

- It is sent while the card is in read operation (except CMD12 which is legal).
- It is sent while the card is in Busy.
- Card is locked and it is other than Class 0 or 7 commands.
- It is not supported (illegal opcode).
- CRC check failed.
- It contains an illegal operand.
- It was out of sequence during an erase sequence.

Note that in case the host sends command while the card sends data in read operation then the response with an illegal command indication may disturb the data transfer.

7.2.10 Memory Array Partitioning

Same as for SD mode.

7.2.11 Card Lock/unlock

Usage of card lock and unlock commands in SPI mode is identical to SD mode. In both cases, the command response is of type R1b. After the busy signal clears, the host should obtain the result of the operation by issuing a GET_STATUS command. Please refer to Chapter 6.4.7 for details.

7.2.12 Application Specific Commands

The Application Specific commands are identical to SD mode with the exception of the APP_CMD status bit (refer to Chapter 6.11.1)

7.2.13 Copyright Protection Commands

All the special Copyright Protection ACMDs and security functionality are the same as for SD mode.

7.2.14 Switch function command

Same as for SD mode with two exceptions:

- The command is valid under the "not idle state"
- In SPI mode, CMD0 switching period is within 8 clocks after the end bit of the CMD0 command R1 response

7.2.15 High-Speed mode (25MB/sec interface speed)

Same as for SD mode.

7.3 SPI Command Set

7.3.1 Command Format

All the SD(microSD) Card commands are 6 bytes long. The command transmission always starts with the left bit of the bit-string corresponding to the command codeword. All commands are protected by a CRC. The commands and arguments are listed in Table 7-3

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	'0'	'1'	x	x	x	'1'
Description	start bit	transmission bit	command index	argument	CRC7	end bit

Table 7-1 : Command format in SPI Mode

7.3.2 Command Classes

As in SD mode, the SPI commands are divided into several classes (See Table 7-2). Each class supports a set of card functions. A SD(microSD) Card will support the same set of optional command classes in both communication modes (there is only one command class table in the CSD register). The available comand classes, and the supported command for a specific class, however, are different in the microSD Card and the SPI communication mode.

Note that except the classes that are not supported in SPI mode (class 1,3 and 9), the mandatory required classes for the SD mode are the same for the SPI mode.

Card CMD Class (CCC)	Class Description	Supported commands																												
		0	1	6	9	10	12	13	16	17	18	24	25	27	28	29	30	32	33	34	35	36	37	38	42	50	55	56	57	58	59
class 0	Basic	+	+		+	+	+	+																						+	+
class 1	Not supported in SPI																														
class 2	Block read								+	+	+																				
class 3	Not supported in SPI																														
class 4	Block write								+			+	+	+																	
class 5	Erase																	+	+					+							
class 6	Write-protec-tion (Optional)														+	+	+														
class 7	Lock Card (Optional)								+																+						
class 8	Application specific																										+	+			
class 9	Not supported in SPI																														
class 10	Switch			+																+	+	+	+			+			+		
class 11	Reserved																														

Table 7-2 : Command Classes In SPI Mode

7.3.3 Command Description

The following table provides a detailed description of the SPI mode commands. The responses are defined in Chapter 7.4. Table 7-3 lists all microSD Card commands. A “yes” in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are also reserved in microSD mode sa well.

The binary code of a command is defined by the mnemonic symbol. As an example, the content of the **command index** field is (binary) ‘000000’ for CMD0 and ‘100111’ for CMD39.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the microSD Card
CMD1	Yes[1]	None	R1	SEND_OP_COND	Activates the card’s initialization process (In Thin-1.4mm SD Memroy Card it is valid only if used after re-initializing card -refer to Chapter 7.2.7)
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	reserved for I/O Mode (refer to "SDIO Card Specification")				
CMD6[8]	Yes	[31] Mode 0:Check function 1:Switch function [30:24] reserved (All ’0’) [23:20] reserved for function group 6 (All ’0’ or 0xF) [19:16] reserved for function group 5 (All ’0’ or 0xF) [15:12] reserved for function group 4 (All ’0’ or 0xF) [11:8] reserved for function group 3 (All ’0’ or 0xF) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Checks switchable function (mode 0) and switches card function (mode 1).
CMD7	No				
CMD8	reserved				

Table 7-3 : Commands and Arguments

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD)
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID)
CMD11	No				
CMD12	Yes	None	R1	STOP_TRANSMISSION	Stop transmission on Multiple Block Read Operation
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	selects a block length (in bytes) for all following block commands (read and write)[2]
CMD17	Yes	[31:0] data address	R1	READ_ SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command[3]
CMD18	Yes	[31:0] data address	R1	READ_ MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.
CMD19	reserved				
CMD20	No				
CMD21 ... CMD23	reserved				
CMD24	Yes	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command. [3]
CMD25	Yes	[31:0] data address	R1	WRITE_ MULTIPLE_BLOCK	Continuously writes blocks of data until a “Stop Tran’ Token is sent (instead ’Strat Block’).
CMD26	No				
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD
CMD28	Yes	[31:0] data address	R1b[5]	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits [5]
CMD31	Reserved				

Table 7-3 : Commands and Arguments

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD32	Yes	[31:0] data address	R1	ERASE_WR_BLK_START_ADDR	Sets the address of the first write block to be erased
CMD33	Yes	[31:0] data address	R1	ERASE_WR_BLK_END_ADDR	Sets the address of the last write block of the continuous range to be erased.
CMD34 .... CMD37[8]	Reserved for each command system set by switch function command (CMD6).				
CMD38	Yes	[31:0] stuff bits	R1b	ERASE	Erases all previously selected erase groups
CMD39	No				
CMD40	No				
CMD41	reserved				
CMD42	Yes	[31:0] stuff bits.	R1b	LOCK_UNLOCK	Used to Set/Reset the Password or lock/ unlock the card. A transferred data block inclues all the command details - refer to Chapter 6.4.7 . The size of the Data Block is defined with SET_BLOCK_LEN command.
CMD43-49 CMD51	Reserved				
CMD50[8]	Reserved for each command system set by switch function command (CMD6).				
CMD52...CMD54	Reserved for I/O mode (Refer to "SDIO Card Specification")				
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Defines to the card that the next command is an application specific command rather than a standard command
CMD56	Yes	[31:1] stuff bits [0] RD/WR [7]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose / application specific commands. The size of the Data Block shall be defiend with SET_BLOCK_LEN command.
CMD57[8]	Reserved for each command system set by switch function command (CMD6).				
CMD58	Yes	None	R3	READ_OCR	Reads the OCR register of a card
CMD59	Yes	[31:1] stuff bits [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A ‘1’ in the CRC option bit will turn the option on, a ‘0’ will turn it off
CMD60-CMD63	Reserved for Manufacturer				

Table 7-3: Commands and arguments

1) CMD1 is valid command for the Thin (1.4mm) SD Memory Card only if used after re-initializing a card (not after power on reset).
2) The default block length is as specified in the CSD.
3) The data transferred must not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.
4) The data transferred must not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD.
5) R1b: R1 response with an optional trailing busy signal
6) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (least significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits shall be set to zero
7) RD/WR_: “1” the Host shall get a block of data from the card.
“0” the host sends block of data to the card.
8) This command is added in spec version 1.10

Table 7-3: Commands and arguments

The following table describes all the application specifc commands supported/ reserved by the microSD Card. All the following commands should be preceded with APP_CMD (CMD55)

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
ACMD6	No				
ACMD13	Yes	[31:0] stuff bits	R2	SD_STATUS	Send the SD Memory Card status. The status fields are given in Table 6-27
ACMD17	reserved				
ACMD18	Yes	--	--	--	Reserved for SD security applications[1]
ACMD19 to ACMD21	reserved				
ACMD22	Yes	[31:0] stuff bits	R1	SEND_NUM_WR_BLOCKS	Send the numbers of the well written (without errors) blocks. Responds with 32bit + CRC data block.
ACMD23	Yes	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"= default (one wr block)[2]
ACMD24	reserved				
ACMD25	Yes	--	--	--	Reserved for SD security applications[1]
ACMD26	Yes	--	--	--	Reserved for SD security applications[1]
ACMD38	Yes	--	--	--	Reserved for SD security applications[1]
ACMD39 to ACMD40	reserved				
ACMD41	Yes	None	R1	SEND_OP_COND	Activates the card's initialization process
ACMD42	Yes	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/ Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin1) of the card. The pull-up may be used for card detectioin.
ACMD43 ... ACMD49	Yes	--	--	--	Reserved for SD security applications[1]
ACMD51	Yes	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

Table 7-4 : Application Specific Commands Used or Reserved by the microSD Card - SPI Mode

1) Refer to *"SD Memory Card Security Specification"* for detailed explanation about the SD Security Features
2) Command STOP_TRAN (CMD12) shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

7.4 Responses

There are several types of response tokens. As in the microSD mode, all are transmitted MSB first.

7.4.1 Format R1

This response token is sent by the card after every command, with the exception of SEND_STATUS commands. It is one byte long, and the MSB is always set to zero. The other bits are error indications, an error being signaled by a '1'. The structure of the R1 format is given in Figure 7-7. The meaning of the flags is defined as follows:

- **In idle state**: The card is in idle state and running the initializing process.
- **Erase Reset:** An erase sequence was cleared before executing because an out of erase sequence command was received.
- **Illegal Command**: An illegal command code was detected
- **Communication CRC Error**: The CRC check of the last command failed.
- **Erase Sequence Error**: An error occurred in the sequence of erase commands
- **Address error**: A misaligned address, which did not match the block length, was used in the command.
- **Parameter Error**: The command's argument (e.g. address, block length) was out of the allowed range for this card.

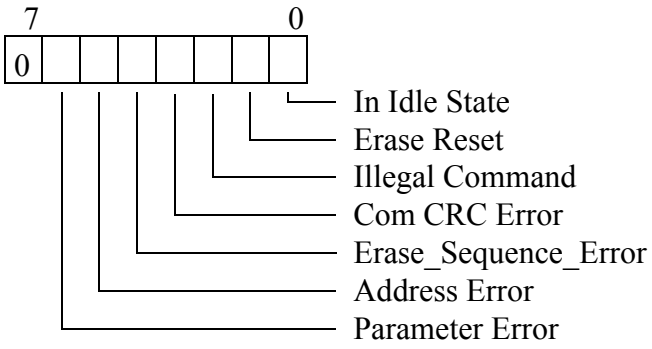


Figure 7-7 : R1 Response Format

7.4.2 Format R1b

This response token is identical to the R1 format with the optional addition of an busy signal. The signal token can be any number of bytes. A zero value indicates card is busy. A non-zero value indicates card is ready for the next command.

7.4.3 Format R2

This response token is two bytes long and sent as a response to the SEND_STATUS command. The format is given in Figure 7-8.

The first byte is identical to the response R1. The content of the second byte is described in the following:

- **Erase Param**: An invalid selection of erase groups, for erase.
- **Write Protect Violation**: The command tried to write a write-protected block.
- **Card ECC Failed**: Card internal ECC was applied but failed to correct the data.
- **CC Error**: Internal card controller error
- **Error**: A general or an unknown error occured during the operatoin
- **Write Protect Erase Skip | Lock/Unlock Command Failed**: This status bit has two functions overloaded. It is set when the host attempts to erase a write-protected sector or makes a sequence or password erro during card lock/unlock operation.
- **Card Is Locked**: Set when the card is locked by the user. Reset when it is unlocked.

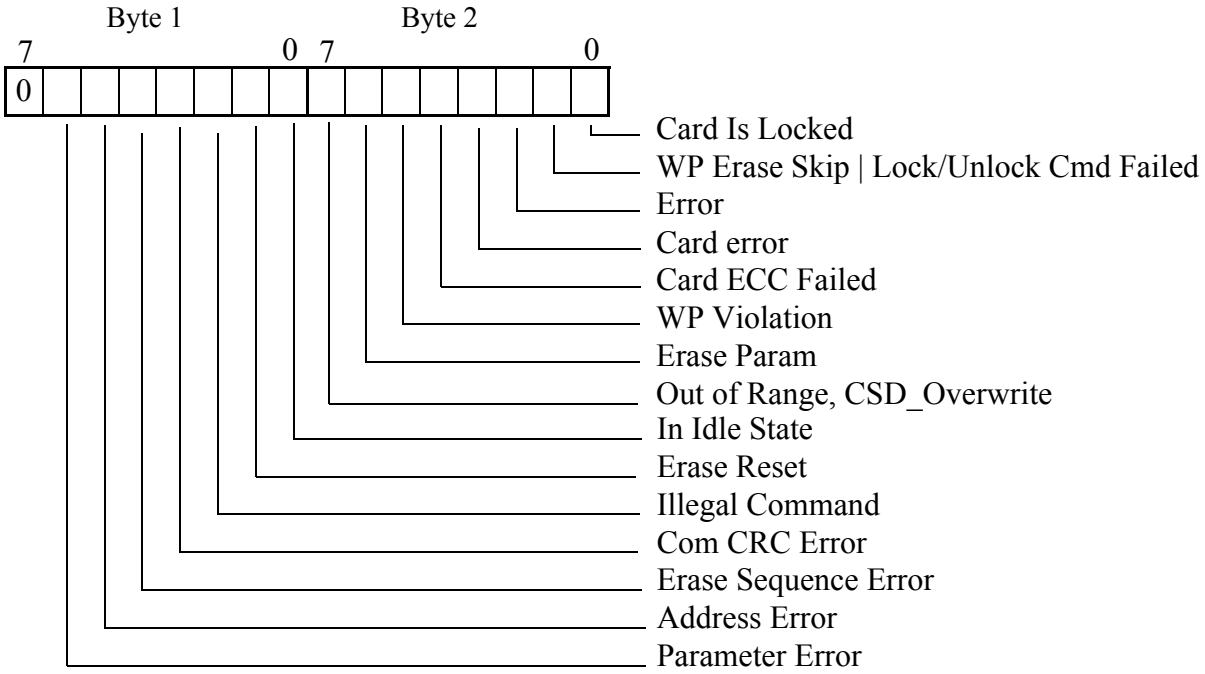


Figure 7-8 : R2 Response Format

7.4.4 Format R3

The card sends a response token when a READ_OCR command is received. The response length is 5 bytes. (see Figure 7-9) The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

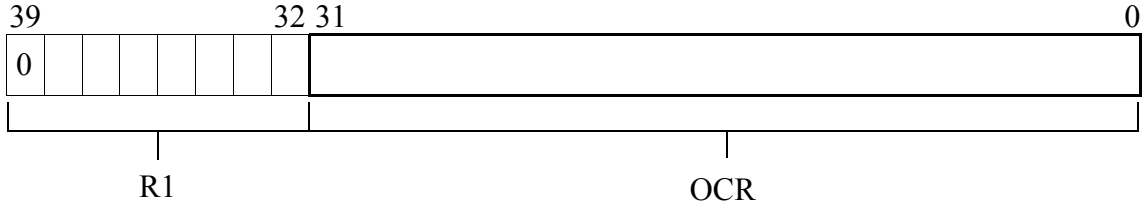


Figure 7-9 : R3 Response Format

7.4.5 Data Response

Every data block written to the card will be acknowledged by a data response token. It is one byte long and has the following format:

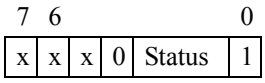


Figure 7-10 : Data Response Format

The meaning of the status bits is defined as follows:

- '010' - Data accepted.
- '101' - Data rejected due to a CRC error.
- '110' - Data Rejected due to a Write Error

In case of any error (CRC or Write Error) during Write Multiple Block operation, the host shall stop the data transmission using CMD12. In case of Write Error (response '110') the host may send CMD13 (SEND_STATUS) in order to get the cause of the write problem. ACMD22 can be used to find the number of well written write blocks.

7.5 Data Tokens

Read and write commands have data transfers associated with them. Data is being transmitted or received via data tokens. All data bytes are transmitted MSB first.
Data tokens are 4 to 512 bytes long and have the following format:

For Single Block Read, Single Block Write and Multiple Block Read:

- First byte: Start Block

7							0
1	1	1	1	1	1	1	0

- Bytes 2 - 513 (depends on the data block length): User data
- Last two bytes: 16 bit CRC.

For Multiple Block Write operation:

- First byte of each block.

If data is to be transferred then - Start Blcok

7							0
1	1	1	1	1	1	0	0

If Stop transmission is requested - Stop Tran

7							0
1	1	1	1	1	1	0	1

Note that this format is used only for Multiple Block Write. In case of Multiple Block Read the stop transmission is done using STOP_TRAN Command (CMD12).

7.6 Data Token Error

If a read operation fails and the card cannot provide the required data, it will send a data error token instead. This token is one byte long and has the following format:

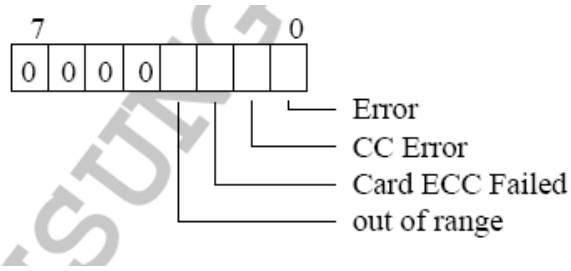


Figure 7-11 : Data Error Token

The 4 least significant bits (LSB) are the same error bits as in the response format R2.

7.7 Clearing Status Bits

As described in the previous paragraphs, in SPI mode, error and status bits are reported to the host in three different formats: response R1, response R2 and data error token (the same bits may exist in multiple response types—e.g Card ECC failed). As in the SD mode, error bits are cleared when by the host, regardless of the response format. State indicators are either cleared by reading or in accordance with the card state.

The following table summarizes the set and clear conditions for the various status bits:

Idnetifier	Included in rep	Type[1]	Value	Description	Clear Cond[2]
Out Of Range	R2 DataErr	E R X	'0'= no error '1'= error	The command's address argument was out of the allowed range for this card.	C
Address error	R1 R2	E R X	'0'= no error '1'= error	A misaligned address which did not match the block length was used in command.	C
Erase Sequence Error	R1 R2	E R	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C
Erase Param	R2	E X	'0'= no error '1'= error	An error in the parameters of the erase command sequence	C
Parameter error	R1 R2	E R X	'0'= no error '1'= error	An error in the parameters of the command	C
WP wiolation	R2	E R X	'0'= not protected '1'= protected	Attempt to program a write protected block	C
Com CRC Error	R1 R2	E R	'0'= no error '1'= error	The CRC check of the previous command failed	C
Illegal command	R1 R2	E R	'0'= no error '1'= error	Command not legal for the card state	C
Card ECC failed	R2 DataEr	E X	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data	C

Table 7-5 : SPI Mode Status Bits

Idnetifier	Included in rep	Type[1]	Value	Description	Clear Cond[2]
CC error	R2 DataErr	E R X	’0’= no error ’1’= error	Internal card controller error	C
Error	R2 DataErr	E R X	’0’= no error ’1’= error	A general or an unknown error occurred during the operation	C
CID/ CSD_OVERWRITE	R2	E R X	’0’= no error ’1’= error	can be either one of the following errors: - The CID register has been already written and can not be overwritten - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bit was made.	C
WP erase skip	R2	S X	’0’=not protected ’1’=protected	Only partial address sapce was erased due to existing write protected blocks.	C
Lock/Unlock Cmd Failed	R2	X	’0’= no error ’1’= error	Sequence or password error during card lock/unlock operation.	C
Card Is Locked	R2	S X	‘0’ = card is not locked ‘1’ = card is locked	Card is locked by a user password	A
Erase Reset	R1 R2	S R	’0’= cleared ’1’= set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
In Idle State	R1 R2	S R	0 = Card is ready 1 = Card is in idle state	The card enters the idle state after power up or reset command. It will exit this state and become ready upon completion of its initialization procedures.	A

Table 7-5 : SPI Mode Status BIts

1) Type:

E: Error bit.
S: State bit.
R: Detected and set for the actual command response.
X: Detected and set during command execution. The host must poll the card by issuing the status command in oreder to read these bits.

2) Clear Condition:

A: According to the card current state.
C: Clear by read

7.8 Card Registers

In SPI mode only the OCR, CSD and CID registers are accessible. Their format is identical to the format in the SD Mode. However, a few fields are irrelevant in SPI mode.

7.9 SPI Bus Timing Diagrams

All timing diagrams use the following schematics and abbreviations:

Symbol	Definition
H	Signal is high (logical ‘1’)
L	Signal is low (logical ‘0’)
X	Don’t care (Undefined Value)
Z	High impedance state (-> = 1)
*	Repeater
Busy	Busy Token
Command	Command token
Response	Response token
Data block	Data token

Table 7-6 : SPI Bus Timing Abbreviations

All timing values are defined in Table7-7. The host must keep the clock running for at least N_{CR} clock cycles after receiving the card response. This restriction applies to both command and data response tokens.

7.9.1 Command / Response

- **Host Command to Card Response - Card is ready**

The following timing diagram describes the basic command response (no data) SPI transaction.

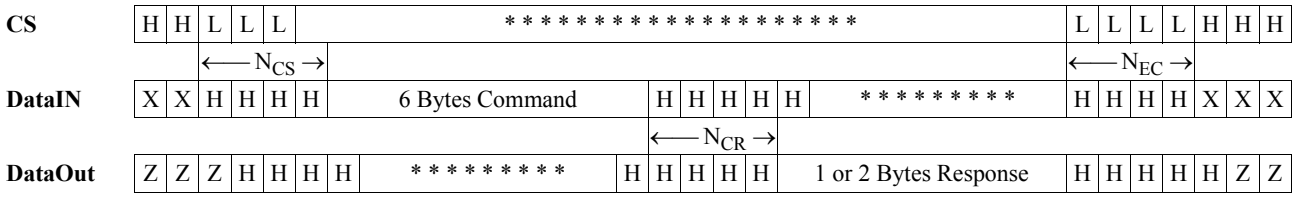


Figure 7-12: Host Command to Card Response - Card Is Ready

- **Host Command to Card Response - card is busy**

The following timing diagram describes the command response transaction for commands when the card response is of type R1b (e.g. SET_WRITE_PROT and ERASE). When the card is signaling busy, the host may deselect it (by raising the CS) at any time. The card will release the DataOut line one clock after the CS going high. To check if the card is still busy, it needs to be reselected by asserting (set to low) the CS signal. The card will resume busy signal (pulling DataOut low) one clock after the falling edge of CS.

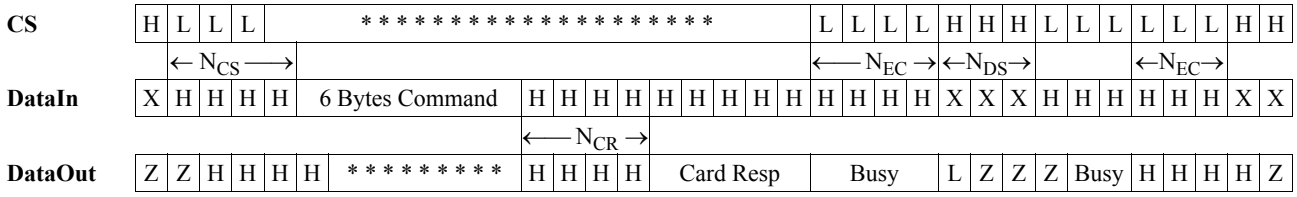


Figure 7-13 : Host Command to Card Response - Card Is Busy (R1b)

- **Card Response to Host Command**

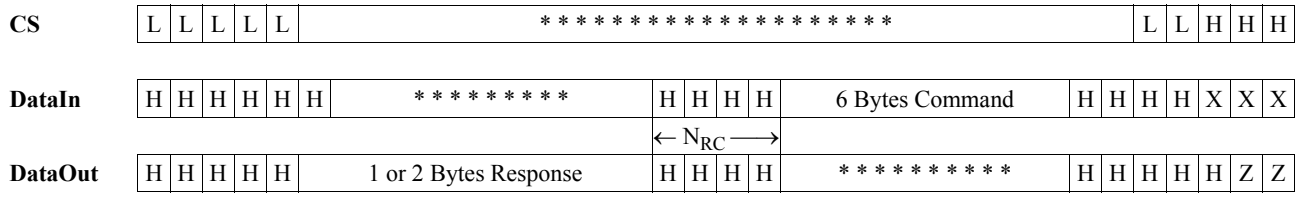


Figure 7-14 : Card Response To The New Host Command

7.9.2 Data read

The following timing diagram describes all single block read operations with the exception of SEND_CSD and SEND_CID commands.

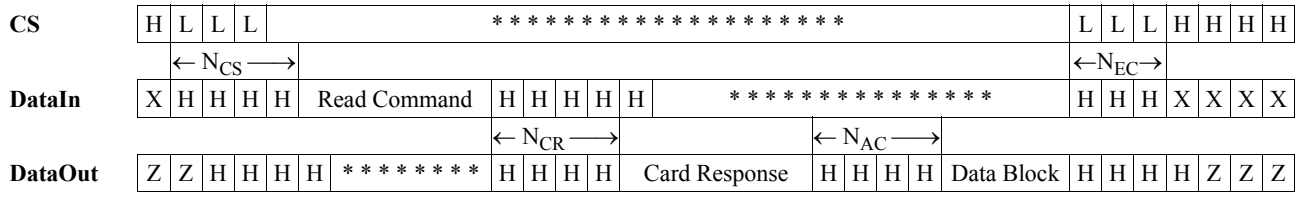


Figure 7-15 : Single Block Read Timing

The following table describes Stop transmission operation in case of Multiple Block Read.

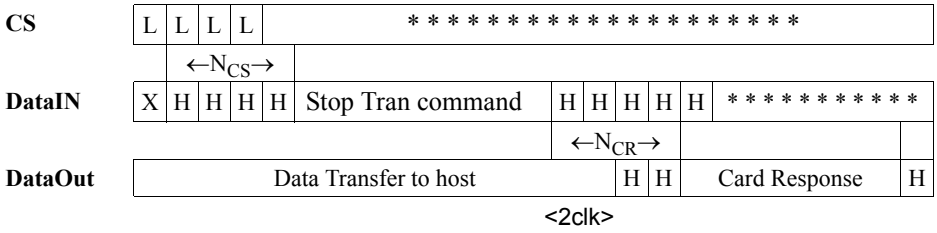


Figure 7-16 : Multiple Block Read Timing

- **Reading the CSD and CID registers**

The following timing diagram describes the SEND_CSD and SEND_CID command bus transaction. The time-out values for the response and the data block is N_{CR} and N_{cx} respectively (Since the N_{ac} is still unknown).

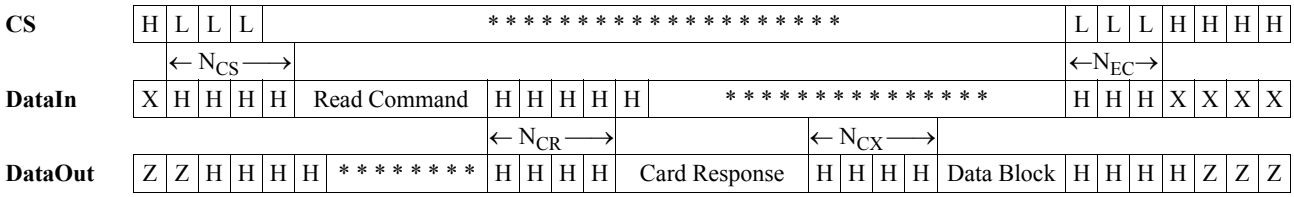


Figure 7-17 : Reading CSD and CID Registers

7.9.3 Data write

The host may deselect a card (by raising the CS) at any time during the card busy period (refer to the given timing diagram). The card will release the DataOut line one clock after the CS going high. To check if the card is still busy it needs to be reselected by asserting (set to low) the CS signal. The card will resume busy signal (pulling DataOut low) one clock after the falling edge of CS.

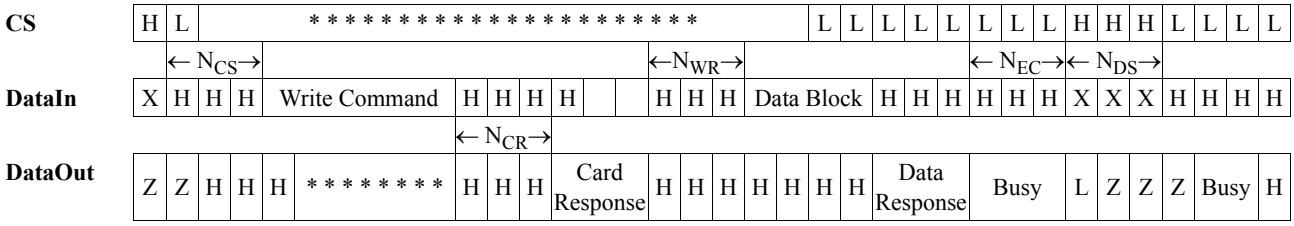
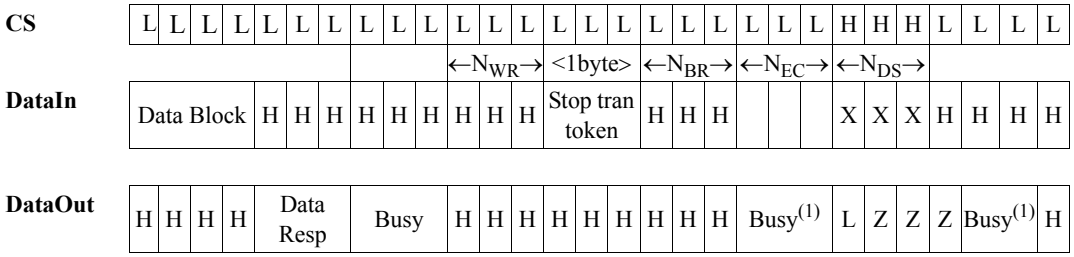


Figure 7-18 : Write operation

The following figure describes stop ransmission operation in Multiple Block Write transfer.



(1) The Busy may appear within N_{BR} clocks after Stop Tran Token. If there is no Busy the host may continue to the next command

Figure 7-19 : SPI Multiple Block Write

7.10 Timing Values

Symbol	Min	Max	Unit
N_{CS}	0	-	8 clock cycles
N_{CR}	1	8	8 clock cycles
N_{RC}	1	-	8 clock cycles
N_{AC}[1]	1	spec. in the CSD	8 clock cycles
N_{WR}	1	-	8 clock cycles
N_{EC}	0	-	8 clock cycles
N_{DS}	0	-	8 clock cycles
N_{BR}	1	1	8 clock cycles
N_{CX}	0	8	8 clock cycles

Table 7-7 : SPI Timing Values

1) The maximum read access time shall be calculated by host as follows:
Nac(max) = 100 ((TAAC * fpp) + (100*NSAC)) :
fpp is the interface clock and TAAC & NSAC are given in the CSD

7.11 SPI Electrical Interface

Identical to SD mode, with the exception of the programmable card output drivers option, which is not supported in SPI mode.

7.12 SPI Bus Operation Conditions

Identical to SD mode.

7.13 Bus Timing

Identical to microSD Card mode. The timing of the CS signal is the same as any other card input.