ATP Industrial Grade
microSD Card Specification

AF512UDI-ADV002

AF1GUDI-ADV002

AF2GUDI-ADV002

Revision 3.9_ADV

ATP Electronics, Inc.

Table of Contents

Disclaimer:

ATP Electronics Inc. shall not be liable for any errors or omissions that may appear in this document, and disclaims responsibility for any consequences resulting from the use of the information set forth herein.

ATP may make changes to specifications and product descriptions at any time, without notice. The information in this paper is furnished for informational use only so ATP assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

Revision History

Date	Version	Changes compared to previous issue
Dec. 20th, 2016	3.9	- Customized Specification for Advantech - OEM PN AF512UDI-ADV002 added - OEM PN AF1GUDI-ADV002 added - OEM PN AF2GUDI-ADV002 added

1.0 ATP Industrial Grade microSD Card Overview

1.1 ATP Product Availability

Figure 1-1: Product Pictures

ATP P/N	CAPACITY
AF512UDI-ADV002	512MB
AF1GUDI-ADV002	1GB
AF2GUDI-ADV002	2GB

Table 1-1: Capacities

1.2 Main Features

- Compliant with SD Specification version 2.0
- Supports SD mode, SPI mode
- High reliability, operating at -40°C to 85°C
- Single-Level-Cell (SLC) NAND Flash memory
- Water proof, Dust proof and ESD Resistant
- SIP (System In Package) process
- Enhanced endurance by Advanced Dynamic/ Static Wear Leveling algorithm
- Read Disturb Protector -AutoRefresh technology to ensure data integrity especially in frequent read operations
- Enhanced power cycling support
- Support BCH ECC engine up to 72bit/1KByte
- Support CPRM
- RoHS compliant
- CE & FCC certification
- Compact form factor : 15mm x 11mm x 1.0mm
- Controlled BOM
- Customized service: adjustable CID registers, firmware & setting and label by projects

1.3 Application

ATP Industrial Grade microSD cards are designed for demanding industrial applications, such as handheld computing, military/aerospace, aviation, automotive, marine navigation, embedded systems, communication equipment or networking, medical equipment, and automation, where mission-critical data requires the highest level of reliability, durability, and data integrity.

2.0 Product Specifications

2.1 Environment Specifications

Type		Measurement
Temperature	Operation	-40°C to 85°C
	Non-Operation	-55°C to 90°C
Humidity	Operation	8%~95% relative humidity, non-condensing
	Non-Operation	8%~95% relative humidity, non-condensing
Random Vibration Test	Non-Operation	10~2000Hz, 6Grms, 30mins per axis
Bend Test	Non-Operation	10N to the center of the card
Torque Test	Non-Operation	0.1N-m or +/-2.5°
Salt Spray Test (MIL-STD-883G Method1009.8)	Non-Operation	35°C,Over 85% RH,3% Salt Concentration, 24 hours
UV Light Exposure Test (ISO 7816-1)	Non-Operation	254nm, 15Ws/cm2
Drop Test	Non-Operation	150cm/Free fall, total 6 drops

Table 2-1: Environment

2.2 Extra Features

Type	Measurement
Water Proof	IEC 60529 Edition 2.1: 2001-02—IPX7, below 1000mm water, 30min
Dust Proof	IEC 60529 Edition 2.1: 2001-02—IP5X
ESD Resistant	IEC 61000-4-2: contact pad +/- 4KV, non-contact pad (Coupling plane discharge) +/- 8KV, non-contact pad (Air discharge) +/- 15KV
RoHS Compliant	Yes

Table 2-2: Extra Features

2.3 Reliability

Type	Measurement		
Number of insertions	10,000 minimum		
Endurance	Advanced Dynamic / Static Wear Leveling algorithm 100K P/E cycles per Block		
TBW (Total Bytes Written)	512MB	10	Terabytes random write
		20	Terabytes sequential write
	1GB	20	Terabytes random write
		40	Terabytes sequential write
	2GB	40	Terabytes random write
		80	Terabytes sequential write
MTBF(@ 25°C)	>5,000,000 hours		

Table 2-3: Reliability

Note 1:
TBW (total bytes written) is an index of how many TB (Terabytes) can be used for written under product life time.
The endurance for flash cards can be predicted based on the usage conditions applied to the device, the internal NAND flash cycles, the write amplification factor, and the wear leveling efficiency of the flash devices.
Above TBW is for reference only. Please contact ATP for TBW in real applications.

1 TeraBytes = 1000 GigaBytes (Disk storage)

Note 2:
MTBF highly depends on testing method. All ATP products are tested with Bellcore Method II (Combines Method I <Parts Count> predictions with laboratory data).

2.4 Electrical Characteristics

Type	Measurement
Card supported Voltage	2.7~3.6V
Card supported Frequency	0~50 MHz
Data Bus Width Supported	1 or 4 bits

Table 2-4: Electrical Characteristics

2.5 Data Retention

Endurance Used	Number of P/E Cycles Used (NAND Flash block level)	Corresponding Data Retention at 25 °C use condition
≦ 10% P/E cycles	≦ 10,000 Cycles	10 years
＞ 10% P/E cycles	＞10,000 Cycles～ 100,000 Cycles	1 year

Table 2-5: Data Retention

Note 1: Data retention refers to the ability of a memory bit to retain its data state over a period of time after the data is written in NAND Flash regardless of whether the part is powered on or powered off.
A data retention failure is when there is at least 1 bit of data that cannot be read or is read incorrectly.

Note 2: NAND Flash suppliers refer to JEDEC JESD47 & JESD22 for Data Retention testing.

2.6 Performance

Capacity	Seq. Read (MB/s)	Seq. Write (MB/s)
512MB	23.75	15.50
1GB	24.01	17.96
2GB	24.01	17.97

Table 2-6: Performance

Note: Tested by CrystalDiskMark with 1000MB file size. The performance may vary depending on the configuration, firmware, setting, application and test environment.

2.7 Advanced Dynamic/ Static Wear Leveling- Longer Life Expectancy

The program / erase cycle of each sector/page/block is finite. Writing constantly on the same spot will cause the flash to wear out quickly. Furthermore, bit errors are not proportioned to P/E cycles; sudden death may occur when the block is close to its P/E cycle limit. Then unrecoverable bit errors will cause fatal data loss (especially for system data or FAT).

Advanced Dynamic/ Static wear leveling algorithm evenly distributes the P/E cycles of each block to minimize the possibility of one block exceeding its max P/E cycles before the rest. In return, the life expectancy of memory storage device is prolonged and the chance/occurrence of unrecoverable bit errors could be reduced.

2.8 Read Disturb Protector – AutoRefresh Technology – Ensure Data Integrity

Over time the error bits accumulate to the threshold in the flash memory cell and eventually become uncorrectable despite using the ECC engine. In the traditional handling method, the data is moved to a different location in the flash memory; despite the corrupted data is beyond repaired before the transition.

The situation is worse in frequent read applications, such as navigation systems or OS boot-up devices. The map or operation system is preloaded into the SD/microSD card and there may be one time write and following by read operation only. Read disturbance is the result of electrical interference from multiple read operations in surrounding pages. After NAND flash accumulates 100,000 read cycles, uncorrectable ECC errors may occur in the affected pages which results in data failure in the same block.

To prevent data corruption, ATP microSD/SD card monitors the error bit levels in each read operation; when it reaches the preset threshold value, AutoRefresh is activated by programming the data into another block before the data is corrupted. After the re-programming operation is completed, the controller reads the data and compares the data/parity to ensure data integrity.

Owing to different user experiences, please contact ATP for AutoRefresh in real applications.

2.9 Physical Dimension (Units in MM)

Type	Measurement
Length	15mm +/- 0.1mm
Width	11mm +/- 0.1mm
Thickness	1.0mm +/- 0.1mm
Weight	0.4 g Max.

Table 2-9: Physical Dimension

2.10 Mechanical Form Factor (Units in MM)

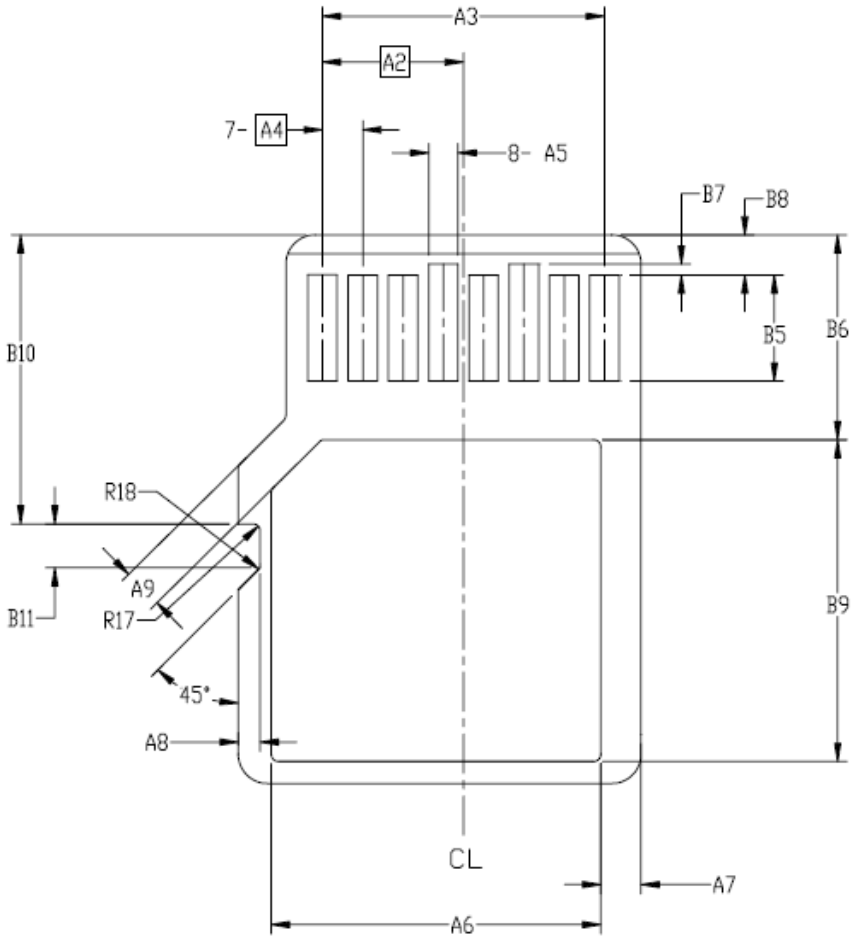


Figure 2-10: Dimensions of microSD (Bottom View)

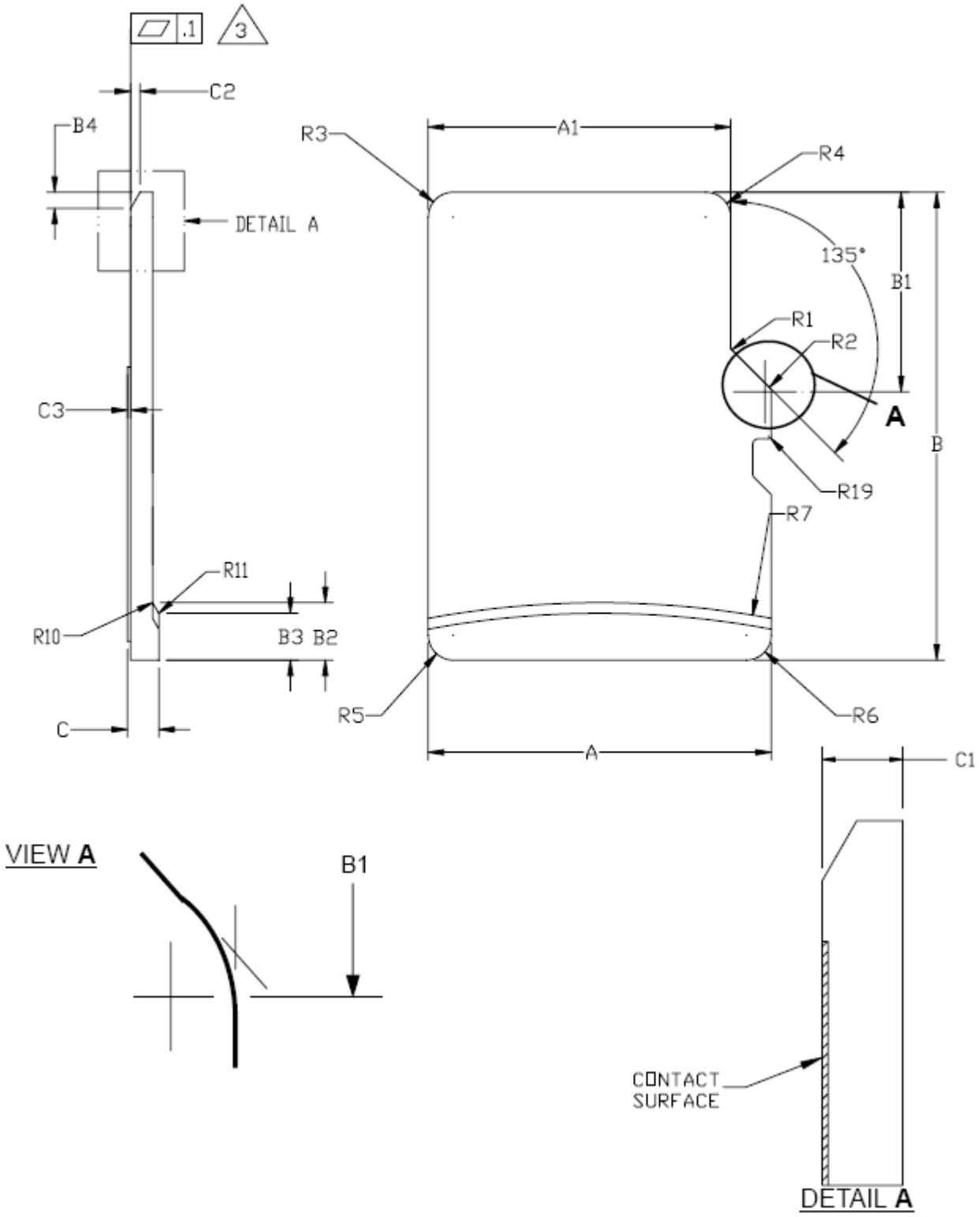


Figure 2-11: Dimensions Of microSD (Top View)

	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	

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	

3.0 Electrical Characteristics

3.1 DC Characteristics

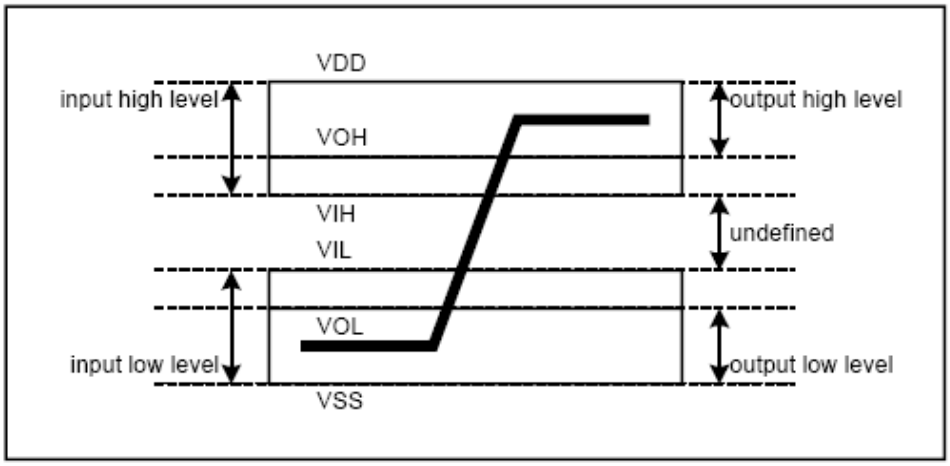


Figure 3-1: Bus Signal Level

Parameter	Symbol	Min	Typical	Max	Unit	Remark
Supply Voltage	V_{DD}	2.7	3.3	3.6	V	
Operating Current*	I_{CC1}	-	50	-	mA	
Standby Current	I_{SB}	-	-	200	µA	
Input Leakage Current	I_{LI}	-10	-	10	µA	
Output Leakage Current	I_{LO}	-10	-	10	µA	
Input High Voltage	V_{IH}	0.625 x V_{DD}	-	V_{DD} + 0.3	V	
Input Low Voltage	V_{IL}	Vss -0.3	-	0.25 x V_{DD}	V	
Output High Voltage	V_{OH}	0.75x V_{DD}	-	-	V	
Output Low Voltage	V_{OL}	-	-	0.125 x V_{DD}	V	

Table 3-1: DC Characteristics

Note*: Operation current might subject to working configuration and applications, and is highly dependent on density. Please contact ATP for detailed information.

3.2 AC Characteristics

Default Bus Timing(Backward Compatible):

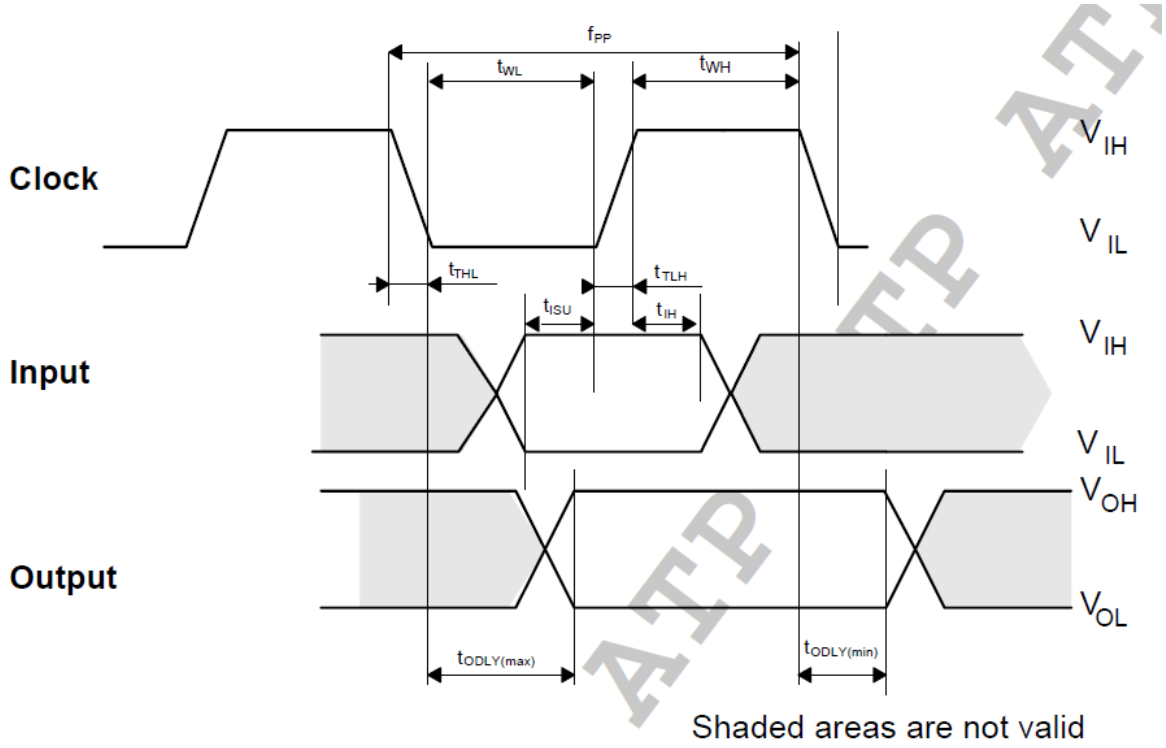


Figure 3-2: Timing diagram data input/output referenced to clock (Default)

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_{CARD} ≤ 10 pF (1 card)
Clock frequency Identification Mode	f_{OD}	$0_{(1)}$/100	400	kHz	C_{CARD} ≤ 10 pF (1 card)
Clock low time	t_{WL}	10		ns	C_{CARD} ≤ 10 pF (1 card)
Clock high time	t_{WH}	10		ns	C_{CARD} ≤ 10 pF (1 card)
Clock rise time	t_{TLH}		10	ns	C_{CARD} ≤ 10 pF (1 card)
Clock fall time	t_{THL}		10	ns	C_{CARD} ≤ 10 pF (1 card)
Inputs CMD, DAT (referenced to CLK)					

Parameter	Symbol	Min	Max.	Unit	Remark
Input set-up time	t_{ISU}	5		ns	$C_{CARD} \leq 10$ pF (1 card)
Input hold time	t_{IH}	5		ns	$C_{CARD} \leq 10$ pF (1 card)
Outputs CMD, DAT (referenced to CLK)					
Output Delay time during Data Transfer Mode	t_{ODLY}	0	14	ns	$C_L \leq 40$ pF (1 card)
Output Delay time during Identification Mode	t_{ODLY}	0	50	ns	$C_L \leq 40$ pF (1 card)

Table 3-2: Bus Timing - Parameters Values (Default)

(1) 0Hz means to stop the clock. The given minimum frequency range is for cases were continues clock is required.

High Speed Mode Bus Timing:

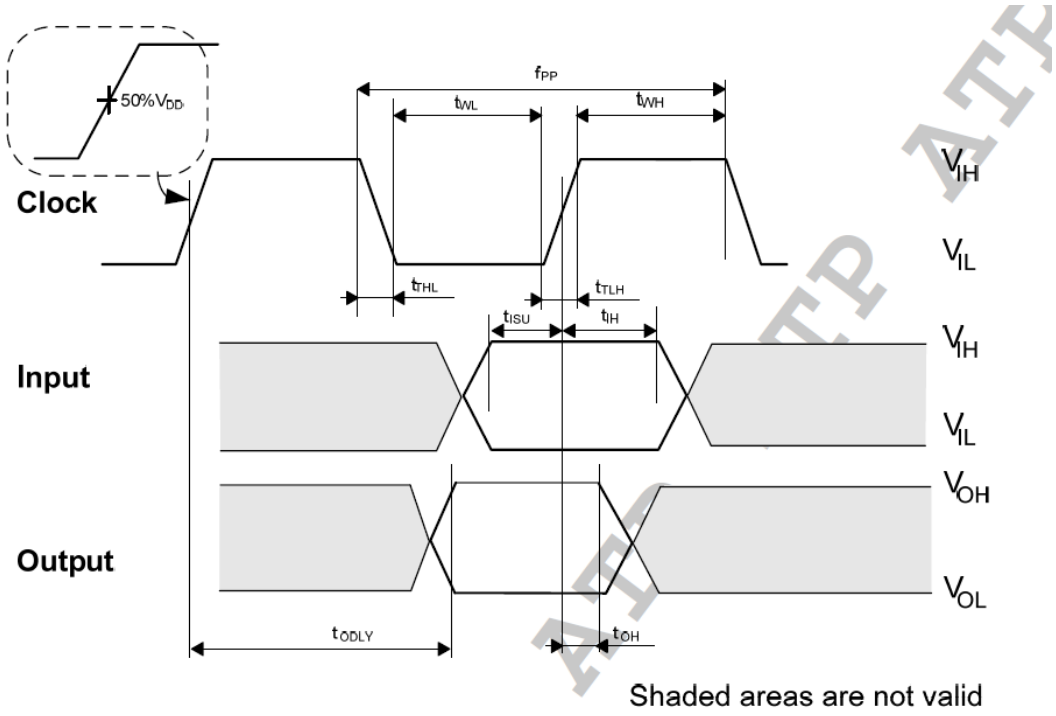


Figure 3-3: Timing diagram data input/output referenced to clock (High-Speed)

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	$C_{CARD} \leq 10$ pF (1 card)
Clock low time	t_{WL}	7		ns	$C_{CARD} \leq 10$ pF (1 card)
Clock high time	t_{WH}	7		ns	$C_{CARD} \leq 10$ pF (1 card)
Clock rise time	t_{TLH}		3	ns	$C_{CARD} \leq 10$ pF (1 card)
Clock fall time	t_{THL}		3	ns	$C_{CARD} \leq 10$ pF (1 card)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	6		ns	$C_{CARD} \leq 10$ pF (1 card)
Input hold time	t_{IH}	2		ns	$C_{CARD} \leq 10$ pF

Parameter	Symbol	Min	Max.	Unit	Remark
					(1 card)
Outputs CMD, DAT (referenced to CLK)					
Output Delay time during Data Transfer Mode	t_{ODLY}		14	ns	$C_L \leq 40$ pF (1 card)
Output Hold time	t_{OH}	2.5		ns	$C_L \geq 15$pF (1 card)
Total System capacitance for each line[1]	C_L		40	pF	1 card

Table 3-3: Bus Timing - Parameters Values (High Speed Mode)

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

4.0 microSD Card Hardware System

4.1 microSD Card Description

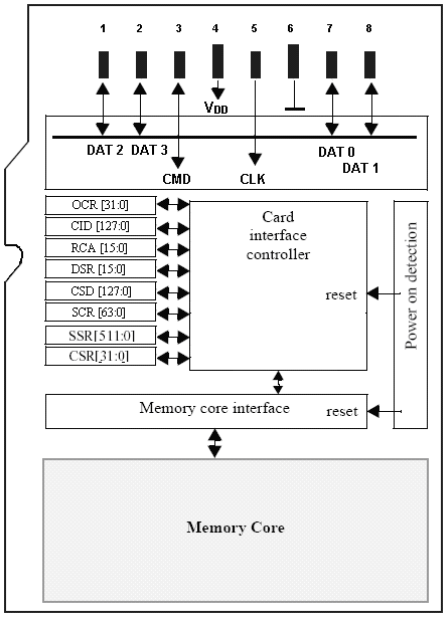


Figure 4-1: microSD Card Function Block Diagram

Pin #	SD Interface			SPI Interface		
	Name	Type[1]	Description	Name	Type[1]	Description
1	DAT2	I/O/PP	Data Line [Bit 2]	RSV		
2	CD/DAT3[2]	I/O/PP[3]	Card Detect / Data Line [Bit 3]	CS	I[3]	Chip Select (Active Low)
3	CMD	PP	Command/ Response	DI	I	Data In
4	V_{DD}	S	Supply Voltage	V_{DD}	S	Supply Voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS}	S	Supply Voltage Ground	V_{SS}	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		

Table 4-1: 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 50Kohm pull-up (can be used for card detection or SPI mode selection). The pull-up should be disconnected by user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command.

Each card has a set of information registers. Please refer to chapter 5 for the details of registers.

Name	Width	Description
CID	128	Card identification number; card individual number for identification.
RCA	16	Relative card address; local system address of a card, dynamically suggested by the card and approved by the host during initialization.
CSD	128	Card Specific Data; information about the card operation conditions.
SCR	64	microSD Configuration Register; information about the SD Card's Special Features capabilities.
OCR	32	Operation conditions register.
SSR	512	microSD Status; information about the card proprietary features.
CSR	32	Card Status; information about the card status.

Table 4-2: microSD Card registers

4.2 microSD BUS Topology

The microSD Card bus has a single master (application), multiple slaves (cards), synchronous star topology. 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 initialization process commands are sent to each card individually, 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 simply the handling of the card stack, after the initialization process, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.

microSD bus allows dynamic configuration of the number of data lines. After power up, by default, 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 easy trade off between HW 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).

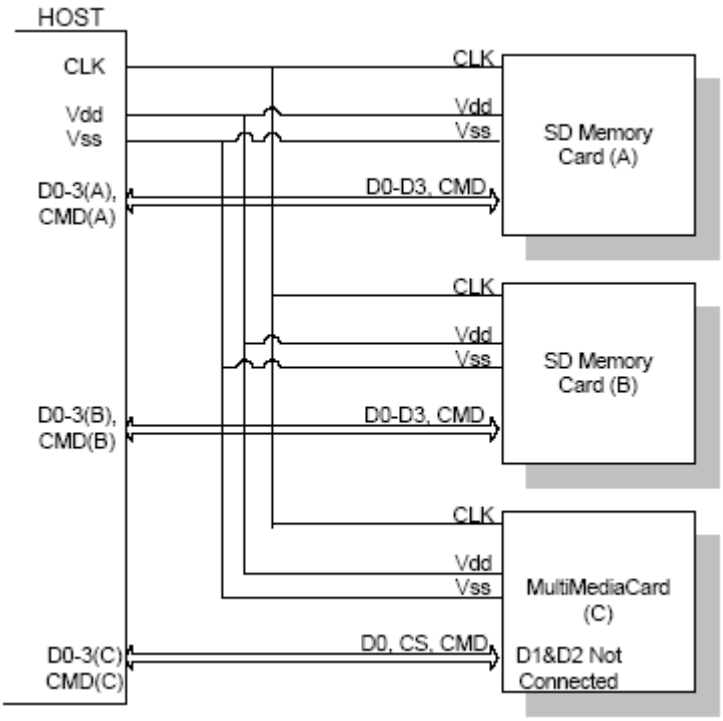


Figure 4-2: microSD Card system bus Topology

4.3 microSD Card Hardware Interface

The microSD Card has six communication lines and three supply lines:

- CMD: Command is a bidirectional signal. The host and card drivers are operating in push pull mode.
- DAT0-3: Data lines are bidirectional signals. Host and card drivers are operating in push pull mode
- CLK: Clock is a host to card signal. CLK operates in push pull mode
- VDD: VDD is the power supply line for all cards.
- VSS1, VSS2 are two ground lines.

In addition to those lines that are connected to the internal card circuitry there are two contacts of the Write Protect/Card Detect switch that are part of the socket. Those contacts are not mandatory but if they exist they should be connected as given in the following figure. When DAT3 is used for card detection, R_{DAT} for DAT3 should be unconnected and another resistor should be connected to the ground.

R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and the DAT lines against bus floating when no card is inserted or when all card drivers are in a high-impedance mode. The host shall pull-up all DAT0-3 lines by RDAT, 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. R_{WP} is used for the Write Protect/Card Detection switch.

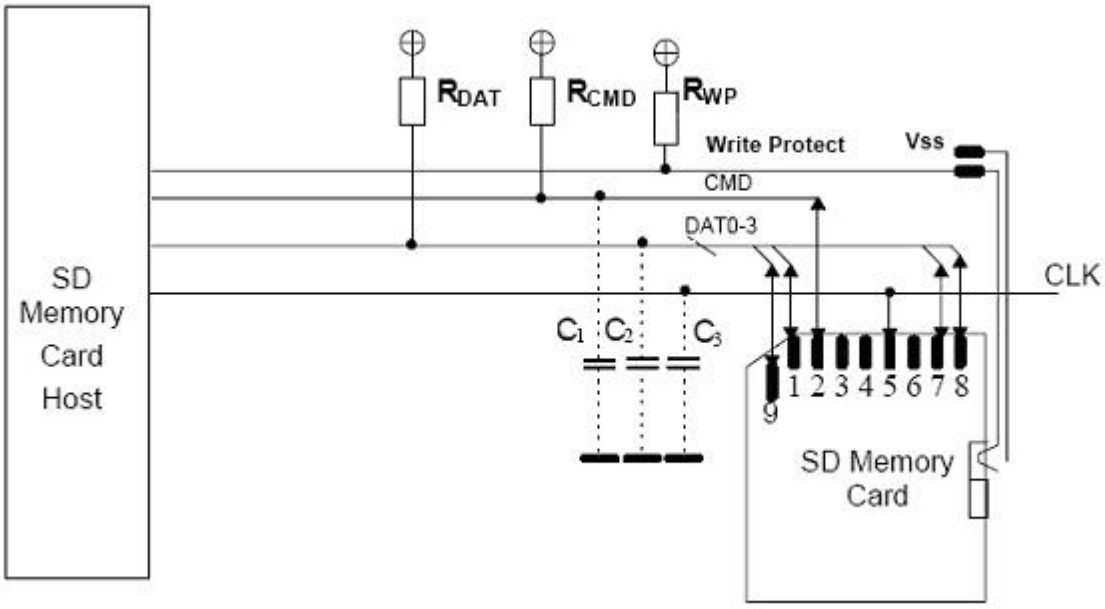


Figure 4-3: Bus circuitry diagram

4.4 Bus Signal Line Load

The total capacitance CL of each line of the microSD bus is the sum of the bus master capacitance C_{HOST}, the bus capacitance CBUS itself and the capacitance C_{CARD} of each card connected to this line:
$C_L = C_{HOST} + C_{BUS} + N*C_{CARD}$
N is the number of connected cards.

Parameter	Symbol	Min	Max.	Unit	Remark
Pull-up resistance for CMD	R_{CMD}	10	100	Kohm	to prevent bus floating
Pull-up resistance for DAT	R_{DAT}	10	100	Kohm	to prevent bus floating
Total bus capacitance for each signal line	C_L		40	pF	1 card CHOST+CBUS shall not exceed 30 pF
Single card capacitance	C_{CARD}		10	pF	
Maximum signal line inductance			16	nH	f_{PP} <= 20 MHz
Pull-up resistance inside card (pin1)	RDAT3	10	90	Kohm	May be used for card detection

Table 4-3: Bus Signal Line Load

4.5 Hot Insertion and Removal

To guarantee the proper sequence of card pin connection during hot insertion, the use of either a special hot-insertion capable card connector or an auto-detect loop on the host side (or some similar mechanism) is mandatory. No card shall be damaged by inserting or removing a card into the microSD Card bus even when the power (VDD) 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 microSD Card 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. They must be corrected by the application, which may repeat the issued command.

4.6 Power up

The power up of the microSD Card bus is handled locally in each SD Card and in the bus master.

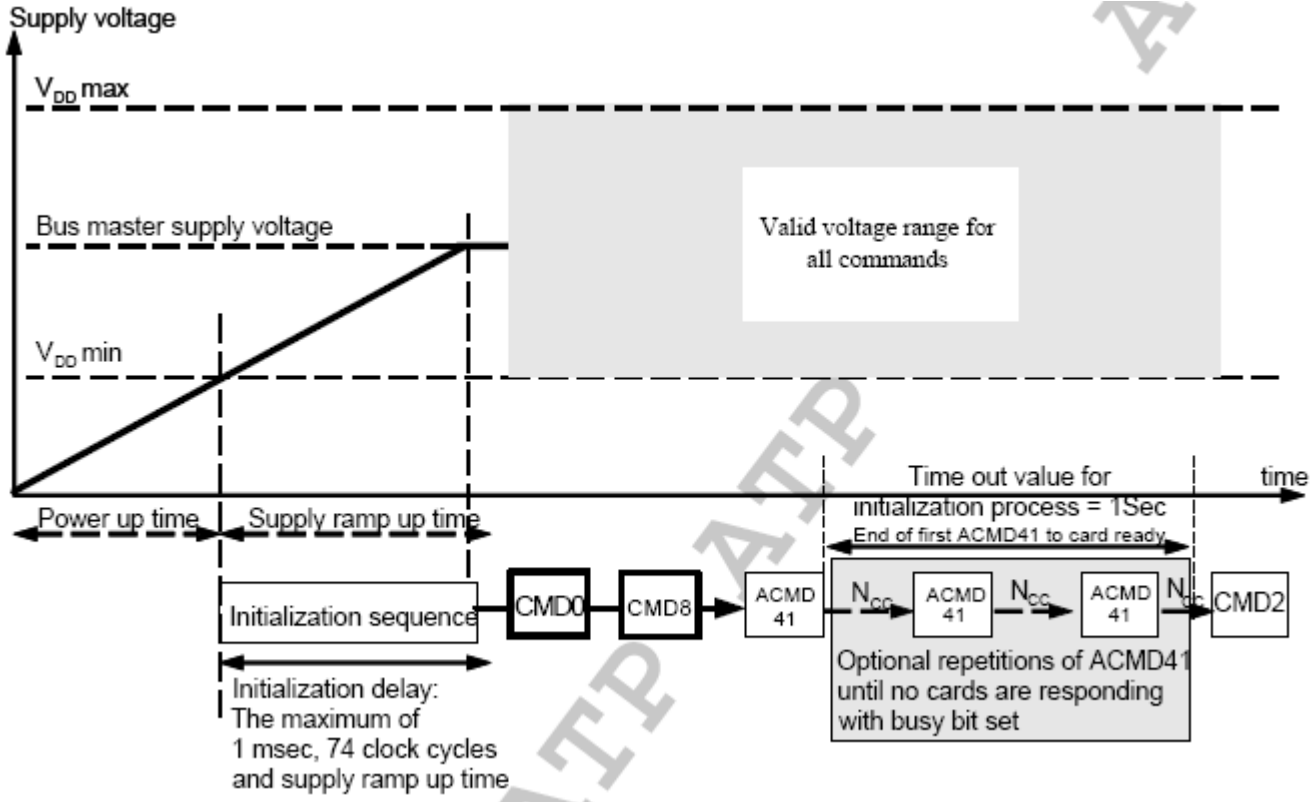


Figure 4-4: Power-up diagram

• 'Power up time' is defined as voltage rising time from 0 volt to VDD min and depends on application parameters such as the maximum number of microSD Cards, the bus length and the characteristic of the power supply unit.

• 'Supply ramp up time' provides the time that the power is built up to the operating level (the bus master supply voltage) and the time to wait until the microSD card can accept the first command.

• The host shall supply power to the card so that the voltage is reached to Vdd_min within 250ms and start to supply at least 74 SD clocks to the microSD card with keeping CMD line to high. In case of SPI mode, CS shall be held to high during 74 clock cycles.

• After power up (including hot insertion, i.e. inserting a card when the bus is operating) the microSD Card enters the *idle state*. In case of SD host, CMD0 is not necessary. In case of SPI host, CMD0 shall be the first command to send the card to SPI mode.

• CMD8 is newly added in the Physical Layer Specification Version 2.00 to support multiple voltage ranges and used to check whether the card supports supplied voltage. The version 2.00 host shall issue CMD8 and verify voltage before card initialization. The host that does not support CMD8 shall supply high voltage range.

• ACMD41 is a synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. In case the host system connects multiple cards, the host shall check that all cards satisfy the supplied voltage. Otherwise, the host should select one of the cards and initialize.

4.7 Compatibility to Multi Media Card

The microSD Card protocol is designed to be a super-set of the Multi Media Card Version 2.11 protocol. For complete details refer to Multi Media Card specification.

4.8 Card Capacity

• Standard Capacity microSD Memory Cards supports capacity up to and including 2 G bytes (231 bytes). All versions of the Physical Specifications define the Standard Capacity microSD Memory Card.
• High Capacity microSD Memory Cards supports capacity more than 2 G bytes (231 bytes) and this version of specification limits capacity up to and including 32 GB. High Capacity SD Memory Card is newly defined from the Physical Layer Specification Version 2.00. Only hosts that are compliant to the Physical Layer Specification version 2.00 or higher and the SD File System Specification Ver2.00 can access High Capacity microSD Memory Cards. Other hosts fail to initialize High Capacity microSD Memory Cards.

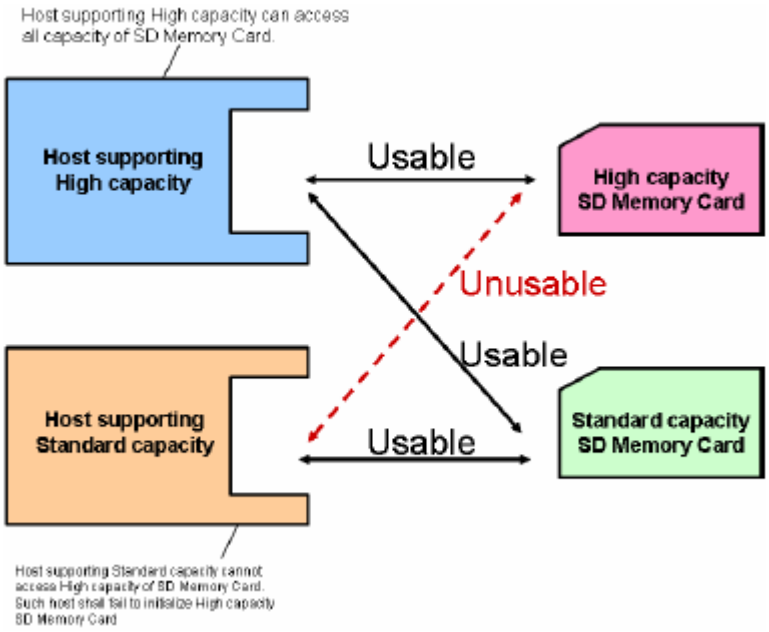


Figure 4-5: Hosts-Cards Usability

5.0 Card Registers

Within the card interface seven registers are defined: OCR, CID, CSD, RCA, SCR, SSR and CSR. These can be accessed only by corresponding commands. The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA register is configuration register storing actual configuration parameters and SSR and CSR are two status fields.

5.1 OCR Register

The 32-bit operation conditions register stores the VDD 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.

OCR bit position	VDD voltage window
0-3	reserved
4	reserved
5	reserved
6	reserved
7	Reserved for Low Voltage Range
8	reserved
9	reserved
10	reserved
11	reserved
12	reserved
13	reserved
14	reserved
15	2.7-2.8
16	2.8-2.9
17	2.9-3.0
18	3.0-3.1
19	3.1-3.2
20	3.2-3.3
21	3.3-3.4
22	3.4-3.5
23	3.5-3.6
24-29	reserved
30	Card Capacity Status (CCS) [1]
31	card power up status bit (busy) [2]

1) This bit is valid only when the card power up status bit is set.
2) This bit is set to LOW if the card has not finished the power up routine

Table 5-1: OCR register definition

The supported voltage range is coded as shown in Table 5-1. 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.

5.2 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 card shall have a unique identification number. The structure of the CID register is defined in the following paragraphs:

Name	Field	Width	CID-slice
Manufacturer ID	MID	8	[127:120]
OEM/Application ID	OID	16	[119:104]
Product name	PNM	40	[103:64]
Product revision	PRV	8	[63:56]
Product serial number	PSN	32	[55:24]
reserved	--	4	[23:20]
Manufacturing date	MDT	12	[19:8]
CRC7 checksum	CRC	7	[7:1]
not used, always '1'	-	1	[0:0]

Table 5-2: The CID fields

- **MID**

An 8 bit binary number identifies the card manufacturer. The MID number is controlled, defined and allocated to a SD Card manufacturer by the SD Group. This procedure is established to ensure uniqueness of the CID register.

- **OID**

A 2 ASCII string characters that identifies the card OEM and/or the card contents (when used as a distribution media either on ROM or FLASH cards). The OID number is controlled, defined and allocated to a microSD Card manufacturer by the SD Group. This procedure is established to ensure uniqueness of the CID register.

- **PNM**

The product name is a string, 5 ASCII characters long.

- **PRV**

The product revision is composed of two Binary Coded Decimal (BCD) digits, four bits each, representing an "n.m" revision number. The "n" is the most significant nibble and "m" is the least significant nibble. As an example, the PRV binary value field for product revision "6.2" will be: 0110 0010

• **PSN**

The Serial Number is 32 bits of binary number.

• **MDT**

The manufacturing date composed of two hexadecimal digits, one is 8 bit representing the year(y) and the other is four bits representing the month(m). The “m” field [11:8] is the month code. 1 = January.

The “y” field [19:12] is the year code. 0 = 2000. As an example, the binary value of the Date field for production date “April 2001” will be: 00000001 0100.

• **CRC**

CRC7 checksum (7 bits). This is the checksum of the CID contents.

5.3 CSD Register

The Card-Specific Data register provides information regarding access to the card contents. The CSD defines the data format, error correction type, maximum data access time, 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 types of the entries in the table below are coded as follows: R = readable, W(1) = writable once, W = multiple writable.

• **CSD_STRUCTURE**

Version number of the related CSD structure

CSD_STRUCTURE	CSD structure version	Valid for SD Card Physical Specification Version
0	CSD version 1.0	Version 1.0-1.10 Version 2.00/Standard Capacity
1	CSD version 2.0	Version 2.00 /High Capacity
2-3	reserved	

Table 5-3: CSD register structure

5.3.1 CSD Register (CSD Version 1.0)

Name	Field	Width	Value	Cell Type	CSD-slice
CSD structure	CSD_STRUCTURE	2	00b	R	[127:126]
reserved	-	6	00 0000b	R	[125:120]
data read access-time-1	TAAC	8	xxh	R	[119:112]
data read access-time-2 in CLK cycles (NSAC*100)	NSAC	8	xxh	R	[111:104]
max. data transfer rate	TRAN_SPEED	8	32h or 5Ah	R	[103:96]
card command classes	CCC	12	01x110110101b	R	[95:84]
max. read data block length	READ_BL_LEN	4	xh	R	[83:80]
partial blocks for read allowed	READ_BL_PARTIAL	1	1b	R	[79:79]
write block misalignment	WRITE_BLK_MISALIGN	1	xb	R	[78:78]
read block misalignment	READ_BLK_MISALIGN	1	xb	R	[77:77]
DSR implemented	DSR_IMP	1	xb	R	[76:76]
reserved	-	2	00b	R	[75:74]
device size	C_SIZE	12	xxxh	R	[73:62]
max. read current @VDD min	VDD_R_CURR_MIN	3	xxxb	R	[61:59]
max. read current @VDD max	VDD_R_CURR_MAX	3	xxxb	R	[58:56]
max. write current @VDD min	VDD_W_CURR_MIN	3	xxxb	R	[55:53]
max. write current @VDD max	VDD_W_CURR_MAX	3	xxxb	R	[52:50]
device size multiplier	C_SIZE_MULT	3	xxxb	R	[49:47]
erase single block enable	ERASE_BLK_EN	1	xb	R	[46:46]
erase sector size	SECTOR_SIZE	7	xxxxxxxb	R	[45:39]
write protect group size	WP_GRP_SIZE	7	xxxxxxxb	R	[38:32]
write protect group enable	WP_GRP_ENABLE	1	xb	R	[31:31]
reserved (Do not use)		2	00b	R	[30:29]
write speed factor	R2W_FACTOR	3	xxxb	R	[28:26]
max. write data block length	WRITE_BL_LEN	4	xxxxb	R	[25:22]
partial blocks for write allowed	WRITE_BL_PARTIAL	1	xb	R	[21:21]
reserved	-	5	00000b	R	[20:16]
File format group	FILE_FORMAT_GRP	1	xb	R/W(1)	[15:15]
copy flag (OTP)	COPY	1	xb	R/W(1)	[14:14]
permanent write protection	PERM_WRITE_PROTECT	1	xb	R/W(1)	[13:13]
temporary write protection	TMP_WRITE_PROTECT	1	xb	R/W	[12:12]
File format	FILE_FORMAT	2	xxb	R/W(1)	[11:10]
reserved		2	00b	R/W	[9:8]
CRC	CRC	7	xxxxxxxb	R/W	[7:1]
not used, always'1'	-	1	1b	-	[0:0]

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

NSAC

Defines the worst case for the clock-dependant factor of the data access time. The unit for NSAC is 100clock cycles. Therefore, the maximal value for the clock-dependent part of the data access time is 25.5k clock cycles.

The total access time NAC as expressed in the Table 4-47 is the sum of TAAC and NSAC. It should 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 or stream

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=2.5, 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

Note that for current SD Memory Cards, this field shall be always 0_0110_010b (032h) which is equal to25 MHz - the mandatory maximum operating frequency of SD Memory Card.

In High-Speed mode, this field shall be always 0_1011_010b (05Ah) which is equal to 50 MHz, and when the timing mode returns to the default by CMD6 or CMD0 command, its value will be 032h.

CCC

The SD Memory 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.

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

BL_LEN

The maximum read data block length is computed as 2READ_BL_LEN. The maximum block length might therefore be in the range 512...2048 bytes (see Chapter 4.11 for details). Note that in an SD Memory 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
10	2^{10} = 1024 Bytes
11	2^{11} = 2048 Bytes
12-15	reserved

READ_BL_PARTIAL (always = 1 in SD Memory Card)

Partial Block Read is always allowed in an SD Memory Card. It means that smaller blocks can be used as well. The minimum block size will be 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) shall be implemented

DSR_IMP	DSR type
0	no DSR implemented
1	DSR implemented

C_SIZE

This parameter is used to compute the user's data 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:

$$\text{memory capacity} = \text{BLOCKNR} * \text{BLOCK\_LEN}$$

$$\text{BLOCKNR} = (\text{C\_SIZE}+1) * \text{MULT}$$
$$\text{MULT} = 2^{\text{C\_SIZE\_MULT}+2} \quad (\text{C\_SIZE\_MULT} < 8)$$
$$\text{BLOCK\_LEN} = 2^{\text{READ\_BL\_LEN}} \quad (\text{READ\_BL\_LEN} < 12)$$

To indicate 2 GByte card, BLOCK_LEN shall be 1024 bytes.
Therefore, the maximal capacity that can be coded is 4096*512*1024 = 2 G bytes.
Example: A 32 Mbyte 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 VDD are coded as follows

VDD_R_CURR_MIN VDD_W_CURR_MIN	Code for Current Consumption @ VDD
2:0	0=0.5mA; 1=1mA; 2=5mA; 3=10mA; 4=25mA; 5=35mA; 6=60mA; 7=100mA

VDD_R_CURR_MAX, VDD_W_CURR_MAX

The maximum values for read and write currents at the maximal power supply VDD are coded as follows:

VDD_R_CURR_MAX VDD_W_CURR_MAX	Code for Current Consumption @ VDD
2:0	0=1mA; 1=5mA; 2=10mA; 3=25mA; 4=35mA; 5=45mA; 6=80mA; 7=200mA

C_SIZE_MULT

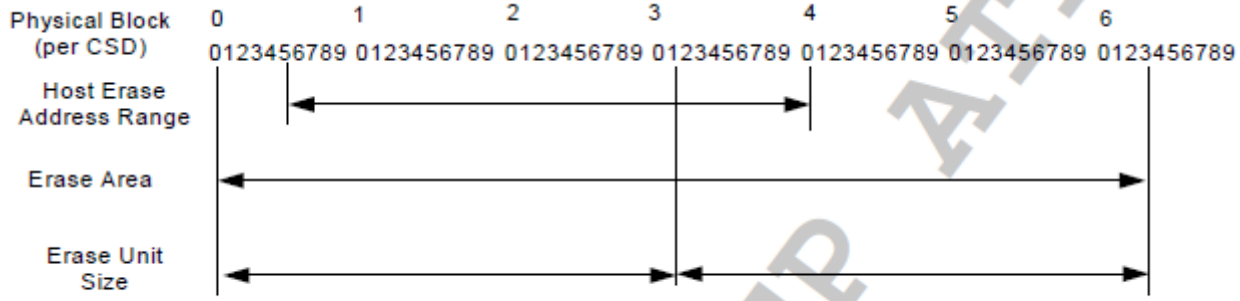
This parameter is used for coding a factor MULT for computing the total device size (see 'C_SIZE'). The factor MULT is defined as $2^{C\_SIZE\_MULT+2}$.

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	$2^9 = 512$

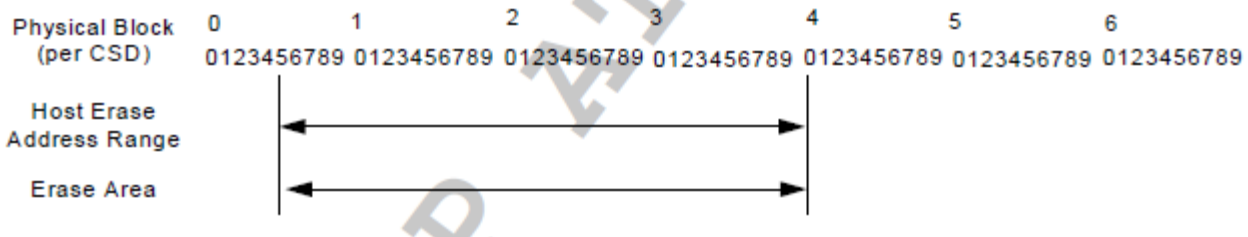
ERASE_BLK_EN

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 512 bytes 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 below figure



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 as below



SECTOR_SIZE

The size of an erasable sector. The content 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 one write block, 127 means 128 write blocks.

WP_GRP_SIZE

The size of a write protected group. The content 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 one 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

WRITE_BL_LEN

The maximum write data block length is computed as 2WRITE_BL_LEN. The maximum block length might therefore be in the range from 512 to 2048 bytes. Write Block Length of 512 bytes is always supported.
Note that in the SD Memory Card, the WRITE_BL_LEN is always equal to READ_BL_LEN.

WRITE_BL_LEN	Block Length
0-8	reserved
9	2^9 = 512 bytes
10	2^{10} = 1024 Bytes
11	2^{11} = 2048 Bytes
12-15	reserved

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 512 bytes, 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 5-15 (see FILE_FORMAT).

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 a one time programmable bit.

PERM_WRITE_PROTECT

Permanently protects the entire 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 entire 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

CRC

The CRC field carries the check sum for the CSD contents. It is computed according to Chapter 4.5. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents

5.3.2 CSD Register (CSD Version 2.0)

Below table shows Definition of the CSD for the High Capacity SD Memory Card (CSD Version 2.0). The following sections describe the CSD fields and the relevant data types for the High Capacity SD Memory Card.
CSD Version 2.0 is applied to only the High Capacity SD Memory Card. The field name in parenthesis is set to fixed value and indicates that the host is not necessary to refer these fields. The fixed values enables host, which refers to these fields, to keep compatibility to CSD Version 1.0. The Cell Type field is coded as follows: R = readable, W(1) = writable once, W = multiple writable

Name	Field	Width	Value	Cell Type	CSD-slice
CSD structure	CSD_STRUCTURE	2	01b	R	[127:126]
reserved	-	6	00 0000b	R	[125:120]
data read access-time	(TAAC)	8	0Eh	R	[119:112]
data read access-time in CLK cycles (NSAC*100)	(NSAC)	8	00h	R	[111:104]
max. data transfer rate	(TRAN_SPEED)	8	32h or 5Ah	R	[103:96]
card command classes	CCC	12	01x110110101b	R	[95:84]
max. read data block length	(READ_BL_LEN)	4	9	R	[83:80]
partial blocks for read allowed	(READ_BL_PARTIAL)	1	0	R	[79:79]
write block misalignment	(WRITE_BLK_MISALIGN)	1	0	R	[78:78]
read block misalignment	(READ_BLK_MISALIGN)	1	0	R	[77:77]
DSR implemented	DSR_IMP	1	x	R	[76:76]
reserved	-	6	00 0000b	R	[75:70]
device size	C_SIZE	22	00 xxxxh	R	[69:48]
reserved	-	1	0	R	[47:47]
erase single block enable	(ERASE_BLK_EN)	1	1	R	[46:46]
erase sector size	(SECTOR_SIZE)	7	7Fh	R	[45:39]
write protect group size	(WP_GRP_SIZE)	7	0000000b	R	[38:32]
write protect group enable	(WP_GRP_ENABLE)	1	0	R	[31:31]
reserved		2	00b	R	[30:29]
write speed factor	(R2W_FACTOR)	3	010b	R	[28:26]
max. write data block length	(WRITE_BL_LEN)	4	9	R	[25:22]
partial blocks for write allowed	(WRITE_BL_PARTIAL)	1	0	R	[21:21]
reserved	-	5	00000b	R	[20:16]
File format group	(FILE_FORMAT_GRP)	1	0	R	[15:15]
copy flag (OTP)	COPY	1	x	R/W(1)	[14:14]
permanent write protection	PERM_WRITE_PROTECT	1	x	R/W(1)	[13:13]
temporary write protection	TMP_WRITE_PROTECT	1	x	R/W	[12:12]
File format	(FILE_FORMAT)	2	00b	R	[11:10]
reserved	-	2	00b	R	[9:8]
CRC	CRC	7	xxxxxxxb	R/W	[7:1]
not used, always'1'	-	1	1	-	[0:0]

- **TAAC**

This field is fixed to 0Eh, which indicates 1 ms. The host should not use TAAC, NSAC, and R2W_FACTOR to calculate timeout and should uses fixed timeout values for read and write operations

- **NSAC**

This field is fixed to 00h. NSAC should not be used to calculate time-out values.

- **TRAN_SPEED**

Definition of this field is same as in CSD Version1.0.

- **CCC**

Definition of this field is same as in CSD Version1.0.

- **READ_BL_LEN**

This field is fixed to 9h, which indicates READ_BL_LEN=512 Byte.

- **READ_BL_PARTIAL**

This field is fixed to 0, which indicates partial block read is inhibited and only unit of block access is allowed.

- **WRITE_BLK_MISALIGN**

This field is fixed to 0, which indicates write access crossing physical block boundaries is always disabled in High Capacity SD Memory Card.

- **READ_BLK_MISALIGN**

This field is fixed to 0, which indicates read access crossing physical block boundaries is always disabled in High Capacity SD Memory Card.

- **DSR_IMP**

Definition of this field is same as in CSD Version1.0.

- **C_SIZE**

This field is expanded to 22 bits and can indicate up to 2 TBytes (It is the same as the maximum memory space specified by a 32-bit block address.)
This parameter is used to calculate the user data area capacity in the SD memory card (not include the protected area). The user data area capacity is calculated from C_SIZE as follows:
memory capacity = (C_SIZE+1) * 512K byte
As the maximum capacity of the Physical Layer Specification Version 2.00 is 32 GB, the upper 6 bits of this field shall be set to 0.

- **ERASE_BLK_EN**

This field is fixed to 1, which means the host can erase one or multiple units of 512 bytes.

- **SECTOR_SIZE**

This field is fixed to 7Fh, which indicates 64 KBytes. This value does not relate to erase operation. Version 2.00 cards indicates memory boundary by AU size and this field should not be used.

- **WP_GRP_SIZE**

This field is fixed to 00h. The High Capacity SD Memory Card does not support write protected groups.

- **WP_GRP_ENABLE**

This field is fixed to 0. The High Capacity SD Memory Card does not support write protected groups.

- **R2W_FACTOR**

This field is fixed to 2h, which indicates 4 multiples. Write timeout can be calculated by multiplying the read access time and R2W_FACTOR. However, the host should not use this factor and should use 250ms for write timeout

- **WRITE_BL_LEN**

This field is fixed to 9h, which indicates WRITE_BL_LEN=512 Byte.

- **WRITE_BL_PARTIAL**

This field is fixed to 0, which indicates partial block read is inhibited and only unit of block access is allowed.

- **FILE_FORMAT_GRP**

This field is set to 0. Host should not use this field.

- **COPY**

Definition of this field is same as in CSD Version1.0.

- **PERM_WRITE_PROTECT**

Definition of this field is same as in CSD Version1.0.

- **TMP_WRITE_PROTECT**

Definition of this field is same as in CSD Version1.0.

- **FILE_FORMAT**

This field is set to 0. Host should not use this field.

- **CRC**

Definition of this field is same as in CSD Version1.0.

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 SCR Register

In addition to the CSD register there is another configuration register that named - SD CARD Configuration Register (SCR). SCR provides information on SD Card's special features that were configured into the given card. The size of SCR register is 64 bit. This register is set in the factory by ATP.

The following table describes the SCR register content.

Description	Field	Width	Cell Type	SCR Slice
SCR Structure	SCR_STRUCTURE	4	R	[63:60]
SD Card - Spec. Version	SD_SPEC	4	R	[59:56]
data_status_after erases	DATA_STAT_AFTER_ERASE	1	R	[55:55]
SD Security Support	SD_SECURITY	3	R	[54:52]
DAT Bus widths supported	SD_BUS_WIDTHS	4	R	[51:48]
reserved	-	16	R	[47:32]
reserved for manufacturer usage	-	32	R	[31:0]

Table 5-9: The SCR Fields

- **SCR_STRUCTURE**

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

SCR_STRUCTURE	SCR structure version	Valid for SD Physical Layer Specification Version
0	SCR version No. 1.0	Version 1.01-2.00
1-15	reserved	

Table 5-10: SCR register structure version

- **microSD_SPEC**

Describes the microSD Card Physical Layer Specification version supported by this card.

microSD_SPEC	Physical Layer Specification Version Number

0	Version 1.0-1.01
1	Version 1.10
2	Version 2.00
3-15	reserved

Table 5-11: microSD Card Physical Layer Specification Version

• DATA_STAT_AFTER_ERASE

Defines the data status after erase, whether it is '0' or '1'.

• SD_SECURITY

Describes the security algorithm supported by the card.

SD_SECURITY	Supported algorithm
0	no security
1	Not used
2	Version 1.01
3	Version 2.0
4-7	reserved

Table 5-12: SD Supported security algorithm

• SD_BUS_WIDTHS

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

SD_BUS_WIDTHS	Supported Bus Widths
Bit 0	1 bit (DAT0)
Bit 1	reserved
Bit 2	4 bit (DAT0-3)
Bit 3 [MSB]	reserved

Table 5-13: SD Card Supported Bus Widths

5.6 SSR Register

SD Status; information about the card proprietary features (See 6.5)

5.7 CSR Register

Card Status; information about the card status (See 6.5).

6.0 SD Card Functional 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 (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line.

• **Response**: a response is a token which is sent from an addressed card, or (synchronously) 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.

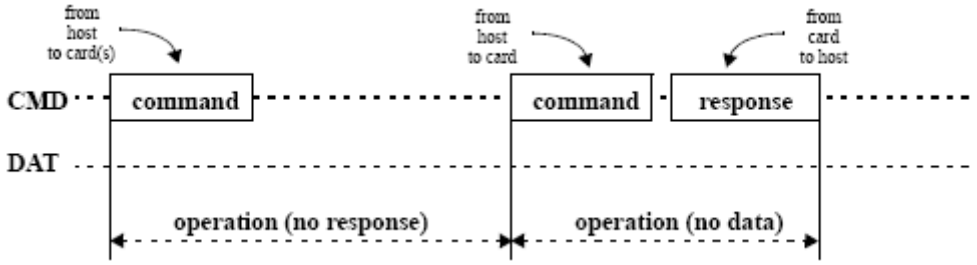


Figure 6-1: "no response" And "no data" Operations

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. This type of bus transactions transfers their information directly within the command or response structure. In addition, some operations have a data token.

Data transfers to/from the microSD Card are done in blocks. Data blocks always were 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.

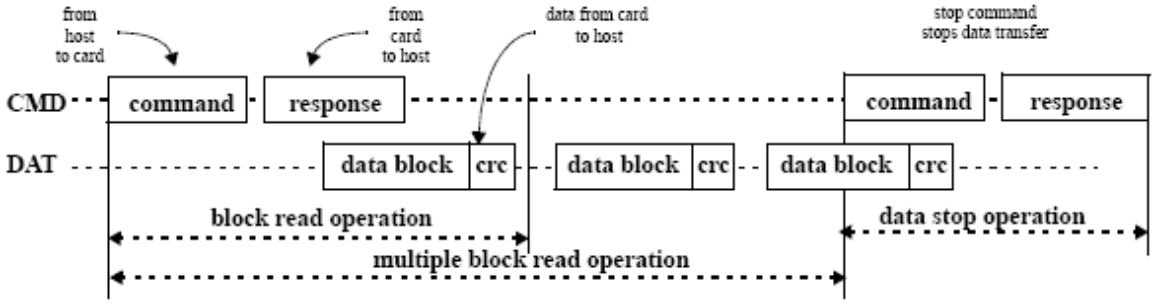


Figure 6-2: (Multiple) Block Read Operation

The block write operation uses a simple busy signaling of the write operation duration on the DAT0 data line regardless of the number of data lines used for transferring the data

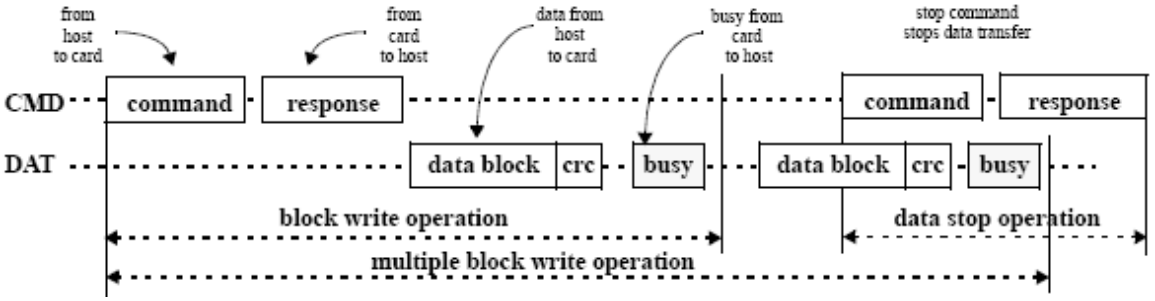


Figure 6-3: (Multiple) Block Write Operation

Command tokens have the following coding scheme:

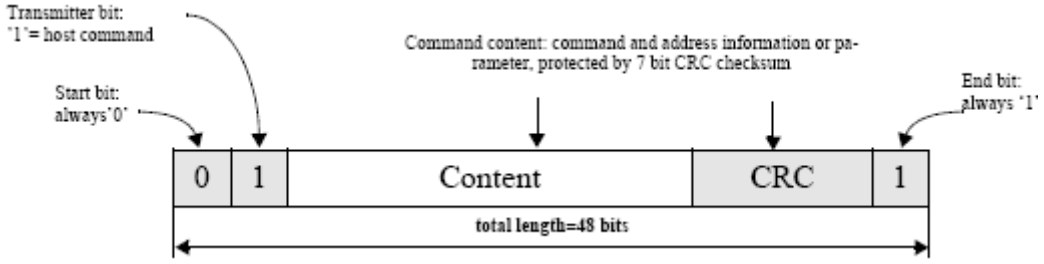


Figure 6-4: Command Token Format

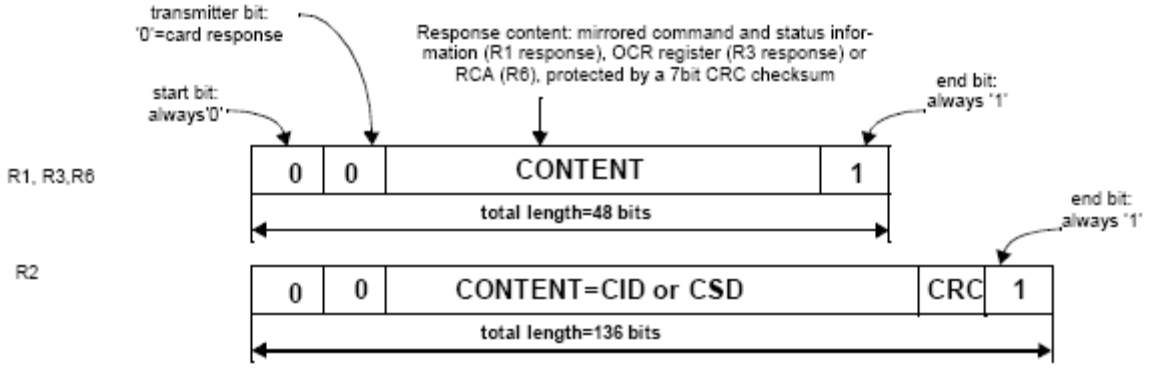


Figure 6-5: Response Token Format

In the CMD line the MSB bit is transmitted first the LSB bit is the last. When the wide bus option is used, the data is transferred 4 bits at a time. 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 individually. The CRC status response and Busy indication will be sent by the card 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.

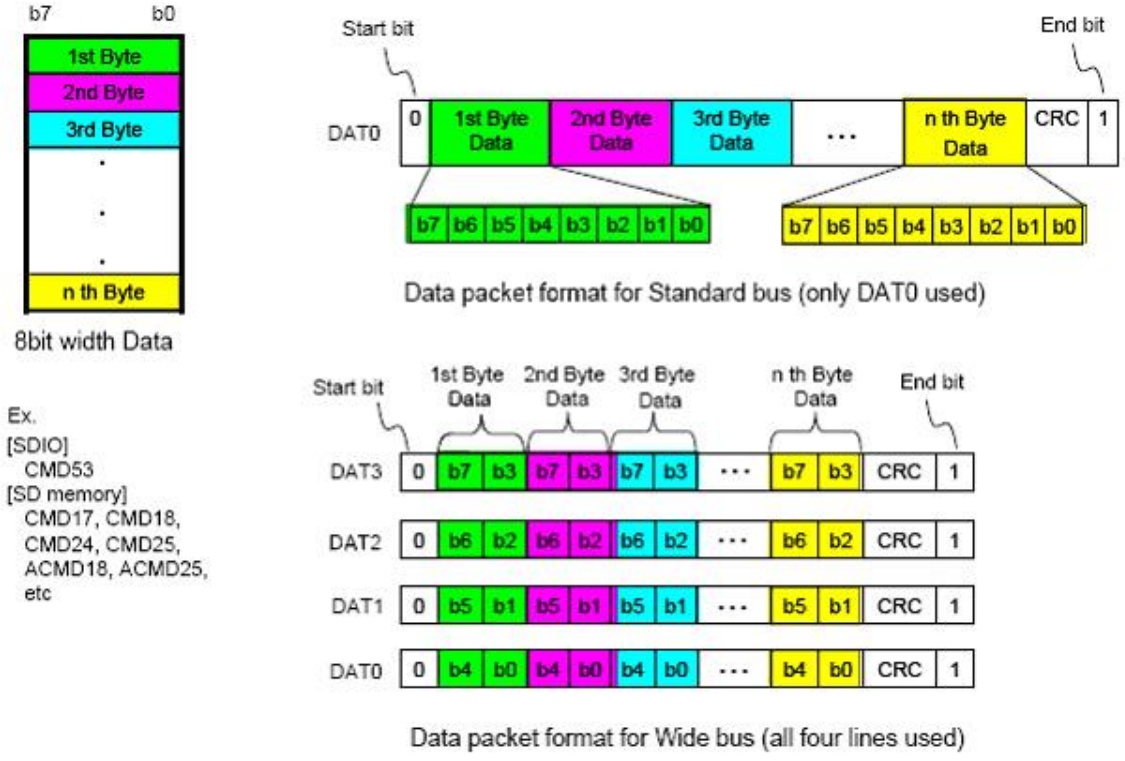


Figure 6-6: Data packet format - Usual data

(2) Wide width data (SD Memory Register) The wide width data is shifted from MSB bit.

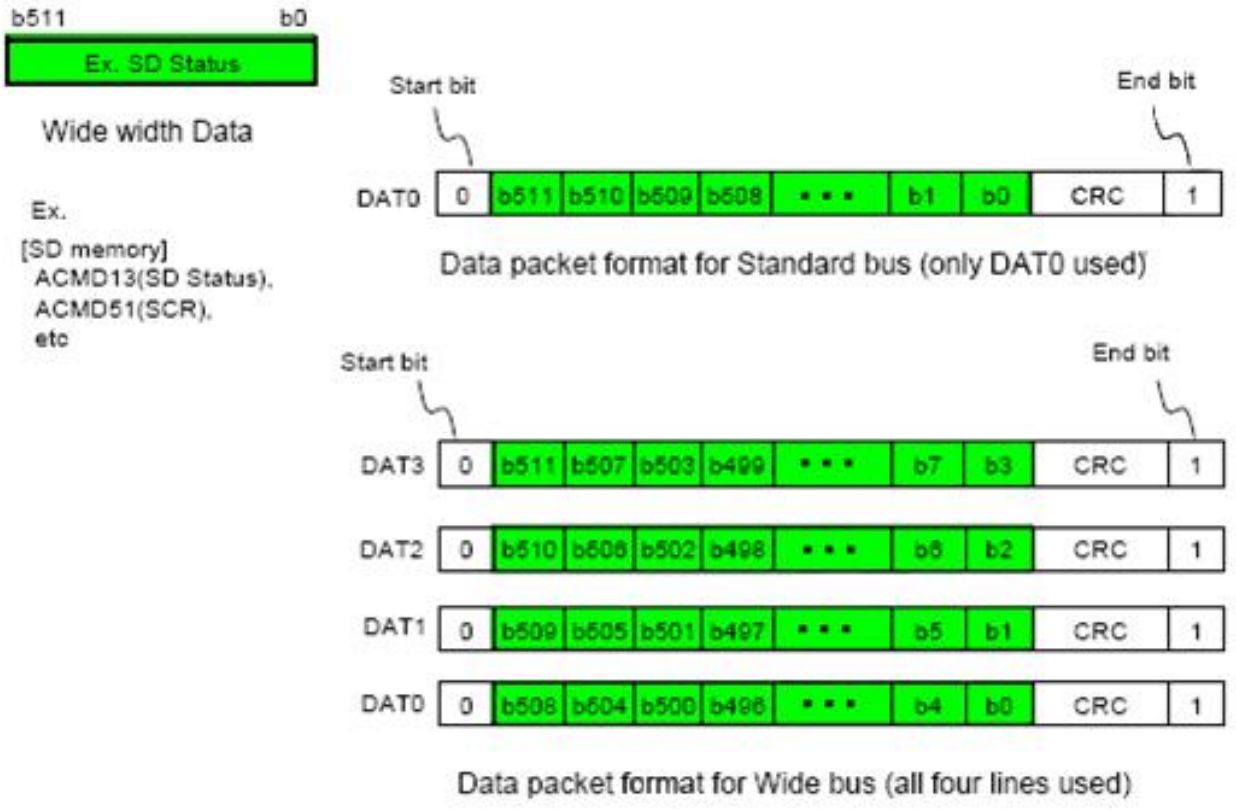


Figure 6-7: Data packet format - Wide width data

6.2 Command

6.2.1 Command Types and Format

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.

- **Command Format**

All commands have a fixed code length of 48 bits, needing a transmission time of 2.4 µs @ 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 6-1: 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 table above indicates this variable is dependent on the command. All commands are protected by a CRC. Every command codeword is terminated by the end bit (always '1'). All commands and their arguments are listed in Table 6-3-Table 6-11.

6.2.2 Command Classes

The command set of the SD Card system is divided into several classes (See Table 6-2). Each class supports a set of card functionalities.

Class 0, 2, 4, 5 and 8 are mandatory supported by ATP SD 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.

	CARD COMMAND CLASS												
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
SUPPORTED COMMANDS	class description	**basic**	**reserved**	**block read**	**Reserved**	**block write**	**erase**	**write protection**	**lock card**	**application specific**	**I/O mode**	**switch**	**reserved**
CMD0	Mandatory	+											
CMD2	Mandatory	+											
CMD3	Mandatory	+											
CMD4	Mandatory	+											
CMD5	Optional										+		
CMD6	Mandatory											+	
CMD7	Mandatory	+											

CMD8	Mandatory	+											
CMD9	Mandatory	+											
CMD10	Mandatory	+											
CMD12	Mandatory	+											
CMD13	Mandatory	+											
CMD15	Mandatory	+											
CMD16	Mandatory			+		+			+				
CMD17	Mandatory			+									
CMD18	Mandatory			+									
CMD24	Mandatory					+							
CMD25	Mandatory					+							
CMD27	Mandatory					+							
CMD28	Optional							+					
CMD29	Optional							+					
CMD30	Optional							+					
CMD32	Mandatory						+						
CMD33	Mandatory						+						
CMD34-37	Optional											+	
CMD38	Mandatory						+						
CMD42	Optional								+				
CMD50	Optional											+	
CMD52	Optional										+		
CMD53	Optional										+		
CMD55	Mandatory									+			
CMD56	Mandatory									+			
CMD57	Optional											+	
ACMD6	Mandatory									+			
ACMD13	Mandatory									+			
ACMD22	Mandatory									+			
ACMD23	Mandatory									+			
ACMD41	Mandatory									+			
ACMD42	Mandatory									+			
ACMD51	Mandatory									+			

Table 6-2: Card Command Classes (CCCs)

6.2.3 Detailed Command Description

The following tables define in detail all microSD Card bus commands.

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	resets all cards to idle state
CMD1	reserved				
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	asks any card to send the 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	ask the card to publish a new relative address (RCA)
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 gets deselected by any other address; address 0 deselects all. In case that the RCA equal 0 then the host may do one of the following: - Use other RCA number to perform card deselection. - Re-send CMD3 to change its RCA number to other than 0 and then use CMD7with RCA=0 for card de-selection.
CMD8	bcr	[31:12]reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to '0'.
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 CMD the line.
CMD11	reserved				
CMD12	ac	[31:0] stuff bits	R1b	STOP_ TRANSMISSION	forces 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 6-3: Basic commands (class 0)

CMD INDEX	Ttype	aArgument	rResp	aAbbreviation	cCommand dDescription
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD Memory Card, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default block length is fixed to 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used. This command is effective for LOCK_UNLOCK command. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_ BLOCK	In the case of a Standard Capacity SD Memory Card, this command, this command reads a block of the size selected by the SET_BLOCKLEN command[1]. In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
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	reserved				
...					
CMD23					

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

Table 6-4: Block oriented read commands (class 2)

CMD INDEX	tType	aArgument	Respresp	aAbbreviation	cCommand dDescription
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD Memory Card, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default block length is fixed to 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used. This command is effective for LOCK_UNLOCK command. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	In the case of a Standard Capacity SD Memory Card, this command writes a block of the size selected by the SET_BLOCKLEN command[1]. In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	Reserved For Manufacturer				
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	programming of the programmable bits of the CSD.

1) 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 block length (given in CSD).

Table 6-5: Block oriented write commands (class 4)

CMD INDEX	Ttype	aArgument	rResp	aAbbreviation	cCommand dDescription
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.[1]
CMD31	reserved				

1)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

Table 6-6 Block oriented write protection commands (class 6)

CMD INDEX	tType	aArgument	rResp	aAbbreviation	cCommand dDescription
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 to be erased.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	erases all previously selected write blocks.
CMD39	reserved				
CMD40					Non Valid in SD Card - Reserved for MultiMediaCard I/O mode
CMD41	reserved				

Table 6-7: Erase commands (class 5)

CMD INDEX	Ttype	aArgument	rResp	aAbbreviation	cCommand Ddescription
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD Memory Card, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default block length is fixed to 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used. This command is effective for LOCK_UNLOCK command. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD42	adtc	[31:0] stuff bits.	R1	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	reserved				
CMD51	Reserved				

Table 6-8: Lock card (class 7)

CMD INDEX	tType	aArgument	rResp	aAbbreviation	cCommand dDescription
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/WR1	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. In the case of a Standard Capacity SD Memory Cards, the size of the data block shall be set by the SET_BLOCK_LEN command. In the case of a High Capacity SD Memory Cards, the size of the data block is fixed to 512 byte. The host sets RD/WR=1 for reading data from the card and sets to 0 for writing data to the card.
CMD58-59	reserved				
CMD60-63	reserved for manufacturer				

1) RD/WR: "1" the host gets a block of data from the card. "0" the host sends block of data to the card. All the application specific commands (given in Table 21) are supported if Class 8 is allowed (mandatory in SD Card).

Table 6-9: Application specific commands (class 8)

CMD INDEX	Ttype	aArgument	rResp	aAbbreviation	cCommand Ddescription
CMD52. CMD54	reserved for I/O mode (refer to "SDIO Card Specification")				

Table 6-10: I/O mode commands (class 9)

The following table describes all the application specific commands supported/reserved by the SD Card. All the following ACMDs shall be preceded with APP_CMD command (CMD55).

CMD INDEX	**Ttype**	**Aargument**	r**Resp**	a**Abbreviation**	c**Command** d**Description**
ACMD6	ac	[31:2] stuff bits [1:0]bus width	R1	SET_BUS_WIDTH	Defines the data bus width (’00’=1bit or ’10’=4 bits 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 SD Card status.
ACMD17	reserved				
ACMD18	--	--	--	--	Reserved for SD security applications1
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. Responds with 32bit+CRC data block. If WRITE_BL_PARTIAL='0', the unit of ACMD22 is always 512byte.If WRITE_BL_PARTIAL='1', the unit of ACMD22 is a block length which was used when the write command was executed.
ACMD23	ac	[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	--	--	--	--	Reserved for SD security applications1
ACMD26	--	--	--	--	Reserved for SD security applications1
ACMD38	--	--	--	--	Reserved for SD security applications1
ACMD39 to ACMD40	reserved				
ACMD41	bcr	[31]reserved bit [30]HCS(OCR[30]) [29:24]reserved bits [23:0] VDD Voltage Window(OCR[23:0])	R3	SD_SEND_OP_COND	Asks the accessed card to send its operating condition register (OCR) content in the 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 (pin 1) of the card.

ACMD43	--	--	--	--	Reserved for SD security applications1
ACMD49					
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

(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.

Table 6-11: Application Specific Commands used/reserved by SD Card

CMD INDEX	t**Type**	a**Argument**	r**Resp**	a**Abbreviation**	c**Command D**d**escription**
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 (mode 0) and switch card function (mode 1).
CMD34	Reserved for each command system set by switch function command (CMD6).				
CMD35					
CMD36					
CMD37					
CMD50					
CMD57					

Table 6-12: Switch function commands (class 10)

6.3 Card State Transition Table

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	-	-
CMD8	idle	-	-	-	-	-	-	-	-	-
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 class 2									
CMD24	-	-	-	-	rcv	-	-	-	-	-
CMD25	-	-	-	-	rcv	-	-	-	-	-
CMD27	-	-	-	-	rcv	-	-	-	-	-
class 6										
CMD28	-	-	-	-	prg	-	-	-	-	-
CMD29	-	-	-	-	prg	-	-	-	-	-
CMD30	-	-	-	-	data	-	-	-	-	-
class 5										
CMD32	-	-	-	-	tran	-	-	-	-	-
CMD33	-	-	-	-	tran	-	-	-	-	-
CMD38	-	-	-	-	prg	-	-	-	-	-
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	-	-	-	-	-

	Current state									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
ACMD22	-	-	-	-	data	-	-	-	-	-
ACMD23	-	-	-	-	tran	-	-	-	-	-
ACMD18,25,26,38, 43,44,45,46,47,48,49	Refer to “SD Card Security Specification” for explanation about the SD Security Features									
ACMD41, card VDD range compatible	ready	-	-	-	-	-	-	-	-	-
ACMD41, card is busy	idle	-	-	-	-	-	-	-	-	-
ACMD41, card VDD range not compatible	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 6-13: Card state transition table

6.4 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 code 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 command codeword is terminated by the end bit (always ‘1’). There are five types of responses for SD Card. Their formats are defined as follows:

• **R1** (normal response command): code 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 in case that data transfer to the card is involved then a busy signal may appear on the data line after the transmission of each block of data. The host shell check for busy after data block transmission.

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

Table 6-14: Response R1

• **R1b** is identical to R1 with an optional busy signal transmitted on the data line. 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. Refer to Chapter 4.12.3 for detailed description and timing diagrams.

• **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 6-15: 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 6-16: Response R3

• **R6** (Published RCA response): code length 48 bit. The bits 45:40 indicate the index of the

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 [31:16] of the card	[15:0] card status bits: 23,22,19,12:0 (see Table 30)	CRC7	end bit

Table 6-17: Response R6

command to be 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.

• **R7**(Card interface condition): Code length is 48 bits. The card support voltage information is sent by the response of CMD8. Bits 19-16 indicate the voltage range that the card supports. The card that accepted the supplied voltage returns R7 response. In the response, the card echoes back both the voltage range and check pattern set in the argument.

Bit position	47	46	[45:40]	[39:20]	[19:16]	[15:8]	[7:1]	0
Width (bits)	1	1	6	20	4	8	7	1
Value	'0'	'0'	'001000'	'00000h'	x	x	x	'1'
Description	start bit	transmission bit	command index	reserved bits	voltage accepted	echo-back of check pattern	CRC7	end bit

Table 6-18: Response R7

Table 6-19 shows the format of 'voltage accepted' in R7.

Voltage Accepted	Value Definition
0000b	Not Defined
0001b	2.7-3.6V
0010b	Reserved for Low Voltage Range
0100b	Reserved
1000b	Reserved
Others	Not Defined

Table 6-19: Voltage Accepted in R7

6.5 SD Card Status

SD Card supports two card status field as follows:

- '*Card Status*': compatible to the MultiMediaCard protocol.

- *'SD_Status*': Extended status field of 512bits that supports special features of the SD Card and future Application Specific features.

6.5.1 Card Status

The response format R1 contains a 32-bit 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 semantics of this register is according to the CSD entry SPEC_VERS, indicating the version of the response formats (possibly used for later extensions). Table 6-19 defines the different entries of the status. The type and clear condition fields in the table are abbreviated 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 order 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: Clear by read.

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	E R X	'0'= no error '1'= error	The command's 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 which did not match the block 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’= not protected ’1’= protected	Attempt to program a write protected block.	C
25	CARD_IS_LOCKED	S X	‘0’ = card unlocked ‘1’ = card locked	When set, signals that the card is locked by the host	A

Bits	Identifier	Type	Value	Description	Clear Condition
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 failed.	B
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
17, 18	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 pro tected blocks.	C
14	CARD_ECC_DISABLED	S X	’0’= enabled 1’= disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	S 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	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 to the next command. The four bits	B

			5 = data; 6 = rcv; 7 = prg 8 = dis 9-14 = reserved 15 = reserved	are interpreted as a binary coded number between 0 and 15.	
8	READY_FOR_DATA	S X	'0'= not ready '1'= ready	corresponds to buffer empty signaling on the bus	A
7,6	reserved				
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				

Bits	Identifier	Type	Value	Description	Clear Condition
3	AKE_SEQ_ERROR	E R	'0' = no error '1' = error	Error in the sequence of authentication process	
2,1,0	reserved				

Table 6-20: Card status

The following table defines for each command responded by a R1 response the affected bits in the status field. An 'x' means the error/status bit may be set in the response to the respective command.

CMD#	Response Format 1 Status bit #																					
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12:9	8	5
3									x	x			x							x		
6	x						x		x	x	x	x	x	x	x					x		
7					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
12	x	x				x	x		x	x	x	x	x	x	x			x		x		
13	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	
16			x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
17	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
18	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
24	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
25	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
26					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
27					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
28	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
29	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
30	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		

32	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
33	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
38				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			
55					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x
56					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ACMD6	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x
ACMD13					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x			x

CMD#	**Response Format 1 Status bit #**																					
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12:9**	**8**	**5**
ACMD22					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD23					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD51					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x

Table 6-21: Card status field / command - cross reference

6.5.2 SD Status

The SD Status contains status bits that are related to the SD 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 16 bit 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 bellow. The same abbreviation for 'type' and 'clear condition' were used as for the Card Status above.

Bits	**Identifier**	**Type**	**Value**	**Description**	**Clear Condition**
511:510	DAT_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 SET_BUS_WIDTH command	A
509	SECURED_MODE	S R	'0'= Not in the mode '1'= In Secured Mode	Card is in Secured Mode of operation	A
508:496	reserved				

495: 480	SD_CARD_TYPE	SR	'00xxh'= SD Memory Cards as defined in Physical Spec Ver1.01-2.00 ('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 an SD Memory Card (Each bit will define different SD Types). The 8 MSBs will be used to define SD Cards that do not comply with current	A
479: 448	SIZE_OF_PROTECTED_AREA	SR	in units of MULT*BLOCK_LEN refer to CSD register	The actual area = (SIZE_OF_PROTECTED_AREA) * MULT * BLOCK_LEN.	A
447: 440	SPEED_CLASS	SR	Speed Class of the card (See below)	(See below)	A
439: 432	PERFORMANCE_MOVE	SR	Performance of move indicated by 1 [MB/s] step. (See below)	(See below)	A
431: 428	AU_SIZE	SR	Size of AU (See below)	(See below)	A
427:	reserved				
424					
423: 408	ERASE_SIZE	SR	Number of AUs to be erased at a time	(See below)	A
407: 402	ERASE_TIMEOUT	SR	Timeout value for erasing areas specified by UNIT_OF_ERASE_AU	(See below)	A
401: 400	ERASE_OFFSET	SR	Fixed offset value added to erase time.	(See below)	A
399:312	reserved				
311:0	reserved				

Table 6-22: SD Card Status

• SIZE_OF_PROTECTED_AREA

Setting this field differs between Standard and High Capacity Cards.
In the case of a Standard Capacity Card, the capacity of protected area is calculated as follows:
Protected Area = SIZE_OF_PROTECTED_AREA_* MULT * BLOCK_LEN.
SIZE_OF_PROTECTED_AREA is specified by the unit in MULT*BLOCK_LEN.
In the case of a High Capacity Card, the capacity of protected area is specified in this field:
Protected Area = SIZE_OF_PROTECTED_AREA
SIZE_OF_PROTECTED_AREA is specified by the unit in byte.

• SPEED_CLASS

This 8-bit field indicates the Speed Class and the value can be calculated by Pw/2.

SPEED_CLASS	Value Definition
00h	Class 0

01h	Class 2
02h	Class 4
03h	Class 6
04h – FFh	Reserved

Table 6-23: Speed Class Code Field

• PERFORMANCE_MOVE

This 8-bit field indicates Pm and the value can be set by 1 [MB/sec] step. If the card does not move used RUs, Pm should be considered as infinity. Setting to FFh means infinity. The minimum value of Pm
is defined by in Table 6-24.

PERFORMANCE_MOVE	Value Definition
00h	Not Defined
01h	1 [MB/sec]
02h	2 [MB/sec]
.....	.....
FEh	254 [MB/sec]
FFh	Infinity

Table 6-24: Performance Move Field

• AU_SIZE

This 4-bit field indicates AU Size and the value can be selected in power of 2 from 16 KB.

AU_SIZE	Value Definition
0h	Not Defined
1h	16 KB
2h	32 KB
3h	64 KB
4h	128 KB
5h	256 KB
6h	512 KB
7h	1 MB
8h	2 MB
9h	4 MB
Ah – Fh	Reserved

Table 6-25: AU_SIZE Field

The maximum AU size, depends on the card capacity, is defined in
Table 6-26. The card can set any AU size between RU size and maximum AU size.

Capacity	16 MB – 64 MB	128 MB-256 MB	512 MB	1 GB – 32 GB
Maximum AU Size	512 KB	1 MB	2 MB	4 MB

Table 6-26: Maximum AU size

- **ERASE_SIZE**

This 16-bit field indicates NERASE. When NERASE numbers of AUs are erased, the timeout value is specified by ERASE_TIMEOUT (Refer to ERASE_TIMEOUT). The host should determine proper number of AUs to be erased in one operation so that the host can indicate progress of erase operation. If this field is set to 0, the erase timeout calculation is not supported.

ERASE_SIZE	Value Definition
0000h	Erase Time-out Calculation is not supported.
0001h	1 AU
0002	2 AU
0003	3 AU
…….	…….
FFFFh	65535 AU

Table 6-27: Erase Size Field

- **ERASE_TIMEOUT**

This 6-bit field indicates the TERASE and the value indicates erase timeout from offset when multiple AUs are erased as specified by ERASE_SIZE. The range of ERASE_TIMEOUT can be defined as up to 63 seconds and the card manufacturer can choose any combination of ERASE_SIZE and ERASE_TIMEOUT depending on the implementation. Once ERASE_TIMEOUT is determined, it determines the ERASE_SIZE. The host can determine timeout for any number of AU erase by the Equation (6). Refer to 4.14 for the concept of calculating erase timeout. If ERASE_SIZE field is set to 0, this field shall be set to 0.

ERASE_TIMEOUT	Value Definition
00	Erase Time-out Calculation is not supported.
01	1 [sec]
02	2 [sec]
03	3 [sec]
…….	…….
63	63 [sec]

Table 6-28: Erase Timeout Field

- **ERASE_OFFSET**

This 2-bit field indicates the TOFFSET and one of four values can be selected. The erase offset adjusts the line by moving in parallel on the upper side. Refer to Figure 4-33 and Equation (6) in 4.14. This field is meaningless if ERASE_SIZE and ERASE_TIMEOUT fields are set to 0.

ERASE_OFFSET	Value Definition
0h	0 [sec]
1h	1 [sec]
2h	2 [sec]
3h	3 [sec]

Table 6-29: Erase Offset Field

6.6 Card Identification Mode and Data Transfer Mode

Two operation modes are defined for the SD Card system:

- **Card identification mode**

The host will be in card identification mode after reset and 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 their RCA is first published. The host will enter data transfer mode after identifying all the cards on the bus. The following table shows the dependencies between operation modes and card states. Each state in the SD Card state diagram (see Figure 6-8) 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 6-30: Overview of Card States vs. Operation modes

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

6.6.1 Card Identification Mode

While in card identification mode the host resets all the cards that are in card identification mode, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only. During the card identification process, the card shall operate in the SD clock frequency of the identification clock rate fOD .

The command GO_IDLE_STATE (CMD0) is the software reset command and sets each card into Idle State regardless of the current card state. Cards in Inactive State are not affected by this command. After power-on by the host, all cards are in Idle State, including the cards that have been in Inactive State before. After power-on or CMD0, all cards' CMD lines are in input mode, waiting for 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).

At the start of communication between the host and the card, the host may not know the card supported voltage and the card may not know whether it supports the current supplied voltage. The host issues a reset command (CMD0) with a specified voltage while assuming it may be supported by the card. To verify the voltage, a following new command (CMD8) is defined in the Physical Layer Specification Version 2.00. SEND_IF_COND (CMD8) is used to verify SD Memory Card interface operating condition. The card checks the validity of operating condition by analyzing the argument of CMD8 and the host checks the validity by analyzing the response of CMD8. The supplied voltage is indicated by VHS filed in the argument. The card assumes the voltage specified in VHS as the current supplied voltage. Only 1-bit of VHS shall be set to 1 at any given time. Both CRC and check pattern are used for the host to check validity of communication between the host and the card. If the card can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument. If the card cannot operate on the supplied voltage, it returns no response and stays in idle state. It is mandatory to issue CMD8 prior to first ACMD41 for initialization of High Capacity SD Memory Card (See Figure 6-30). Receipt of CMD8 makes the cards realize that the host supports the Physical Layer Version 2.00 and the card can enable new functions. It is also mandatory for low-voltage host to send CMD8 before ACMD41. In case that a Dual Voltage Card is not receiving CMD8 the card will work as a high-voltage only card, and in this case that a low voltage host didn't send CMD8 the card will go to inactive at ACMD41. SD_SEND_OP_COND (ACMD41) is designed to provide SD Memory Card hosts with a mechanism to identify and reject cards which do not match the VDD range desired by the host. This is accomplished by the host sending the required VDD voltage window as the operand of this command. Cards which cannot perform data transfer in the specified range shall discard themselves from further bus operations and go into Inactive State. The levels in the OCR register shall be defined accordingly. Note that ACMD41 is application specific command, therefore APP_CMD (CMD55) shall always precede ACMD41. The RCA to be used for CMD55 in idle_state shall be the card's default RCA = 0x0000. After the host issues a reset command (CMD0) to reset the card, the host shall issue CMD8 prior to ACMD41 to re-initialize the SD Memory card.

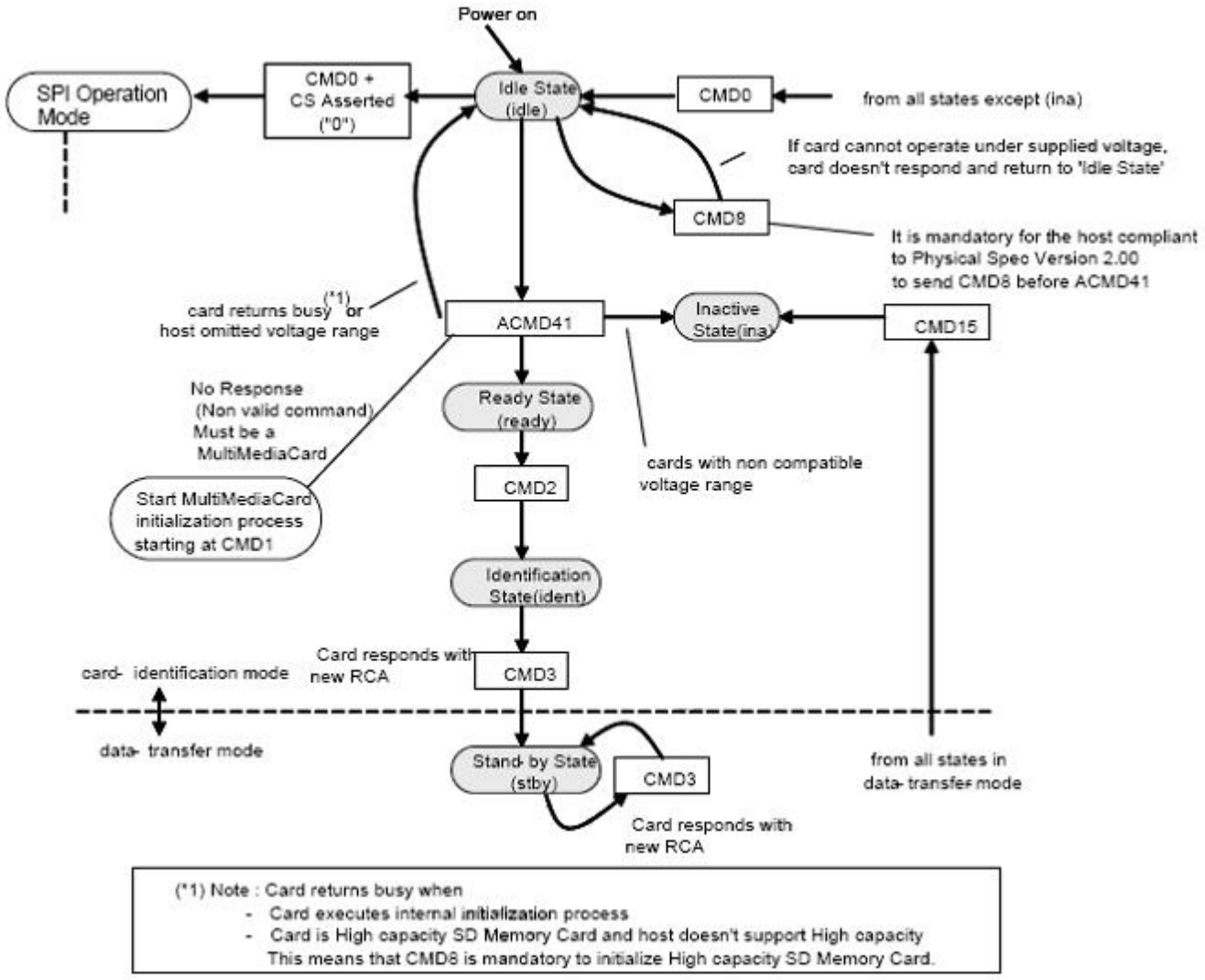


Figure 6-30: SD Card state diagram (card identification mode)

By setting the OCR to zero in the argument of ACMD41, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive State (query mode). This query should be used if the host is able to select a common voltage range or if a notification to the application of non usable cards in the stack is desired. The card does not start initialization if ACMD41 is issued as a query. Afterwards, the host may choose a voltage for operation and reissue ACMD41 with this condition, sending incompatible cards into the Inactive State. During the initialization procedure, the host is not allowed to change the operating voltage range.

After the bus is activated the host starts card initialization and identification process (See Figure 6-31). The initialization process starts with SD_SEND_OP_COND (ACMD41) by setting its operational conditions and the HCS bit in the OCR. The HCS (Host Capacity Support) bit set to 1 indicates that the host supports High Capacity SD Memory card. The HCS (Host Capacity Support) bit set to 0 indicates that the host does not support High Capacity SD Memory card.
Receiving of CMD8 expands the ACMD41 function; HCS in the argument and CCS (Card Capacity Status) in the response. HCS is ignored by cards, which didn't respond to CMD8. However the host

should set HCS to 0 if the card returns no response to CMD8. Standard Capacity SD Memory Card ignores HCS. If HCS is set to 0, High Capacity SD Memory Card never return ready statue (keep busy bit to 0). The busy bit in the OCR is used by the card to inform the host that initialization of ACMD41 is completed. Setting the busy bit to 0 indicates that the card is still initializing. Setting the busy bit to 1 indicates completion of initialization. The host repeatedly issues ACMD41 until the busy bit is set to 1. The card checks the operational conditions and the HCS bit in the OCR only at the first ACMD41. While repeating ACMD41, the host shall not issue another command except CMD0.

If the card responds to CMD8, the response of ACMD41 includes the CCS field information. CCS is valid when the card returns ready (the busy bit is set to 1). CCS=1 means that the card is a High Capacity SD Memory Card.

CCS=0 means that the card is a Standard Capacity SD Memory Card.

The host performs the same initialization sequence 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 was sent by the card it goes into Identification State. Thereafter, the host issues CMD3 (SEND_RELATIVE_ADDR) asks the card to publish a new relative card address (RCA), which is shorter than CID and which is used to address the card in the future data transfer mode. Once the RCA is received the card state changes to the Stand-by State. At this point, if the host wants to assign another RCA number, it can ask the card to publish a new number by sending another CMD3 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.

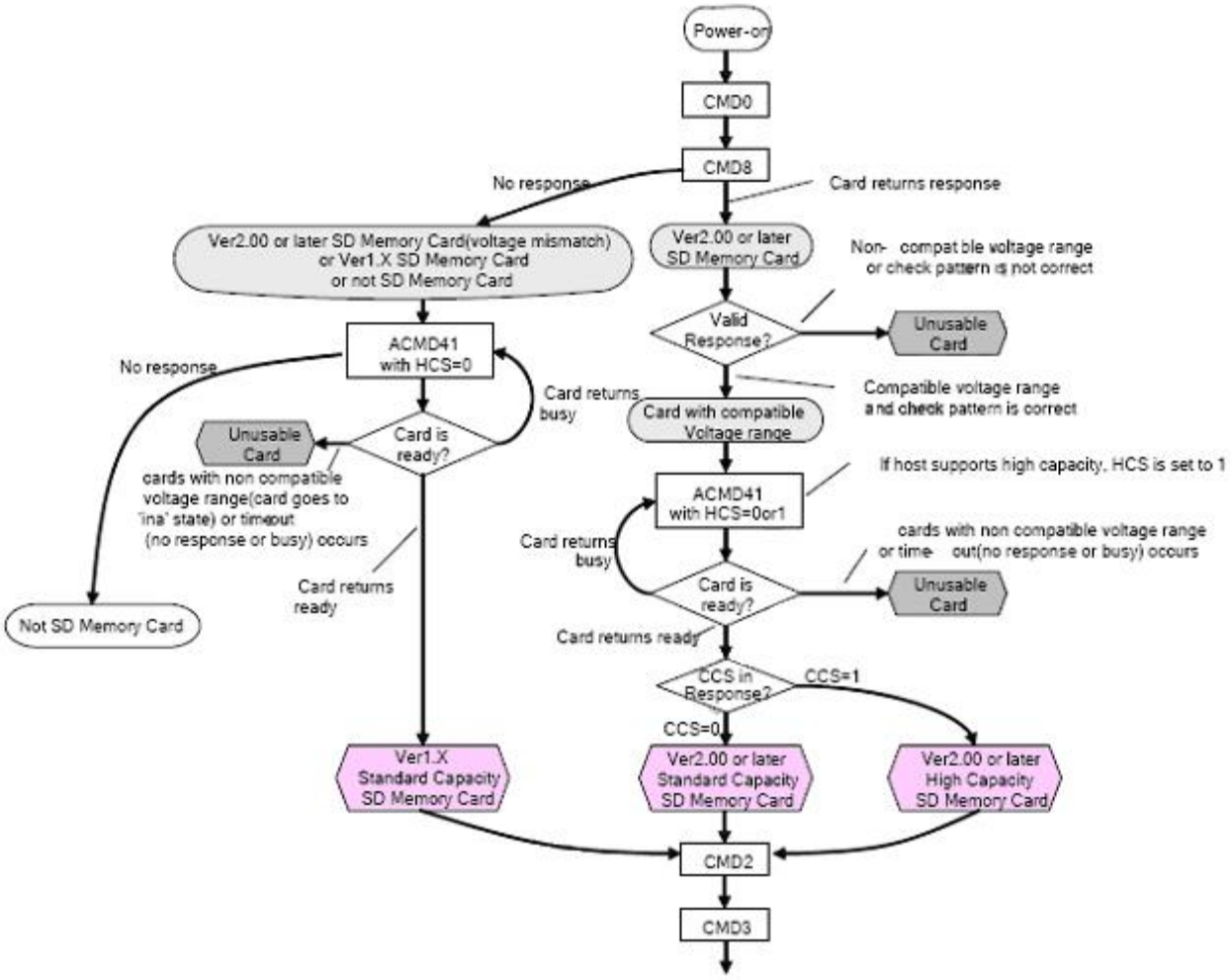


Figure 6-31: Card Initialization and Identification Flow (SD mode)

6.6.2 Data Transfer Mode

Until the end of Card Identification Mode the host must remain at f_{OD} frequency because some cards may have operating frequency restrictions during the card identification mode. In Data Transfer Mode the host may operate the card in f_{PP} frequency range. The host issues SEND_CSD (CMD9) to obtain the Card Specific Data (CSD register), e.g. block length, card storage capacity, etc.

CMD7 is used to select one card and put it into the *Transfer State*. When CMD7 is issued with the reserved relative card address "0x0000", all cards are put back to *Stand-by State* (Note that it is the responsibility of the Host to reserve the RCA=0 for card de-selection).

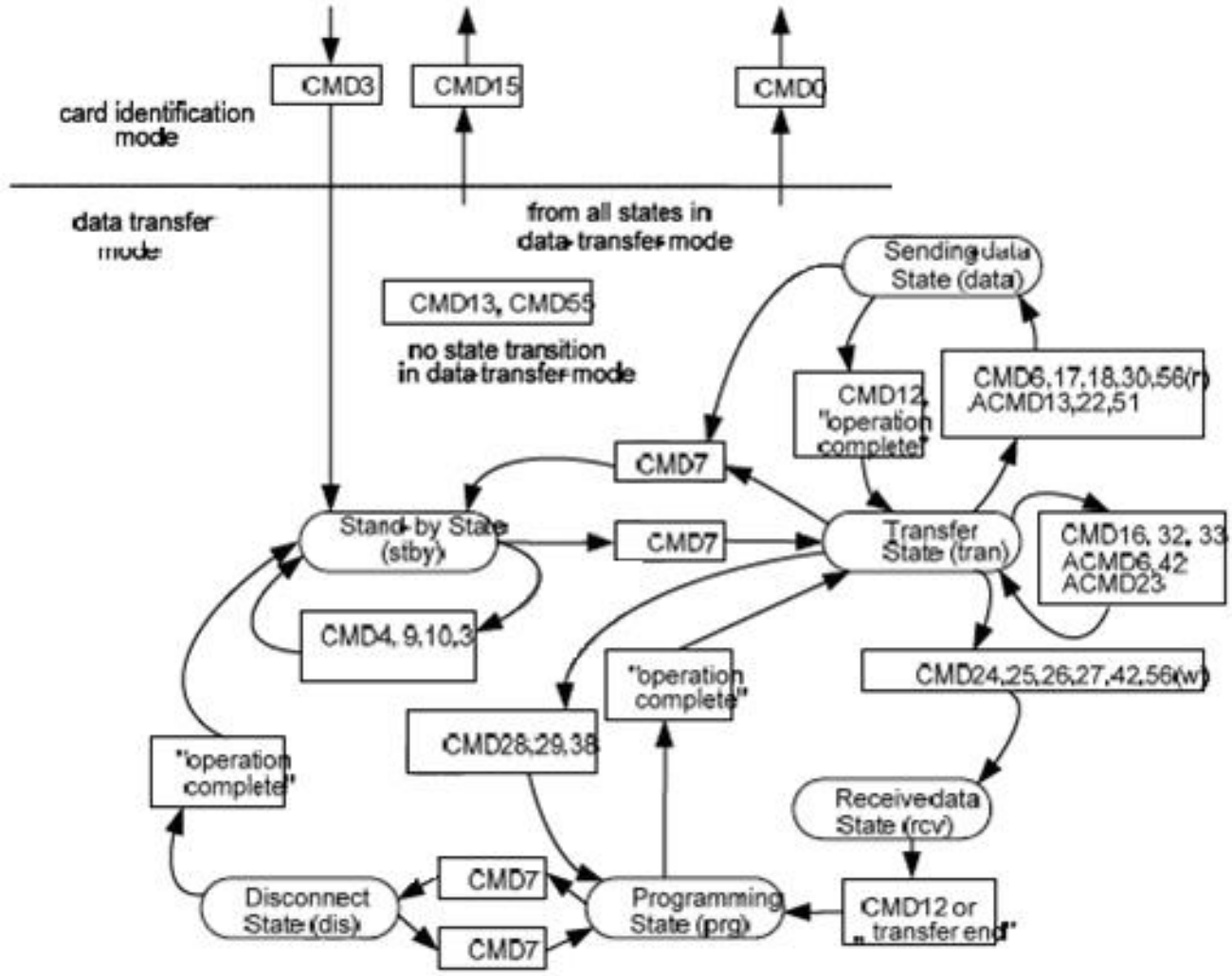


Figure 6-32: SD Card state diagram (data transfer mode)

6.7 Error Handling

To correct defects in the memory field inside card the card include error correction codes in the payload data (ECC). This correction is intended to correct static errors. Additionally two methods of detecting errors generated during the data transfer (dynamic errors) via a cyclic redundancy check (CRC) are implemented

6.7.1 Error Correction Code (ECC)

The ATP SD Card is free of static errors. All errors are covered inside the card, even errors occurring during the lifetime of the card are covered for the user. The only effect which may be notified by the end user is, that the overall memory capacity may be reduced by small number of blocks. All flash handling is done on card, so that no external error correction is needed.

6.7.2 Cyclic Redundancy Check (CRC)

The CRC is intended for protecting SD Card commands, responses and data transfer against transmission errors on the SD 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 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: $G(x) = x^7 + x^3 + 1$.
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + ... + (\text{last bit}) * x^0$
$CRC[6...0] = \text{Remainder } [(M(x) * x^7) / G(x)]$

The first bit is the most left bit of the corresponding bitstring (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).

- **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 $G(x) = x^{16} + x^{12} + x^5 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + ... + (\text{last bit}) * x^0$
$CRC[15...0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

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 is used in single DAT line mode and in wide bus mode. In wide bus mode, the CRC16 is done on each line separately.

6.7.3 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 will not change its state, will not response and will set the ILLEGAL_COMMAND error bit in the status register. Only the non-errodata neous state branches are shown in the state diagrams 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).

6.7.4 Read, Write and Erase Time-out

The times after which a time-out condition for read operations occurs are (card independent) **either 100 times longer** than the typical access times for these operations given below **or 100ms (the lower of them)**. The times after which a time-out condition for Write/Erase operations occurs are (card independent) **either 100 times longer** than the typical program times for these operations given below **or 250ms (the lower of them).** 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 with the given 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. This number is card dependent and should be used by the host to calculate throughput and the maximal frequency for stream read.

• **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(CLR)_WRITE_PROTECT, PROGRAM_CSD and the block write commands).

- **Erase**

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

7.0 SPI Mode

7.1 Introduction

The SPI mode consists of a secondary communication protocol which is offered by SD Cards. This mode is a subset of the SD Card protocol, designed to communicate with a SPI channel, The interface is selected during the first reset command after power up (CMD0) and cannot be changed once the part is powered on.

7.2 SPI BUS Topology

The ATP SD Card SPI interface is compatible with SPI hosts available on the market. As any other SPI device the ATP SD Card SPI channel consists of 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.

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 7-1).

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.

The SPI interface uses the 7 out of the SD 9 signals (DAT1 and DAT 2 are not used, DAT3 is the CS signal) of the SD bus.

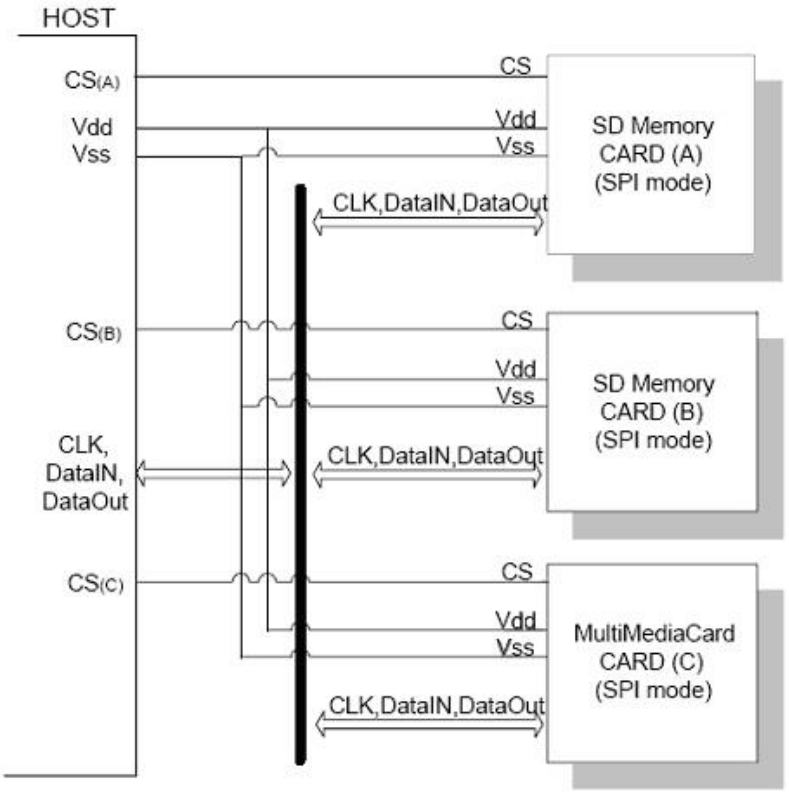


Figure 7-1: SD Card system (SPI mode) bus topology

7.3 SPI Bus Protocol

While the SD Memory Card channel is based on command and data bit streams that 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). The card starts to count SPI bus clock cycle at the assertion of the CS signal. Every command or data token shall be aligned to 8-clock cycle boundary. Similar to the SD Memory Card protocol, the SPI messages consist of command, response and datablock tokens. All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low. The selected card always responds to the command as opposed to the SD mode. When the card encounters a data retrieval problem in a read operation, it will respond with an error response (which replaces the expected data block) rather than by a timeout as in the SD mode. Additionally, every data block sent to the card during write operations will be responded with a data response token.

In the case of a Standard Capacity Memory Card, a data block can 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. In the case of a High Capacity SD Memory Card, the size of data block is fixed to 512 bytes. The block length set by CMD16 is only used for CMD42 and not used for memory data

transfer. So, partial block read/write operations are also disabled. Furthermore, Write Protected commands (CMD28, CMD29 and CMD30) are not supported.

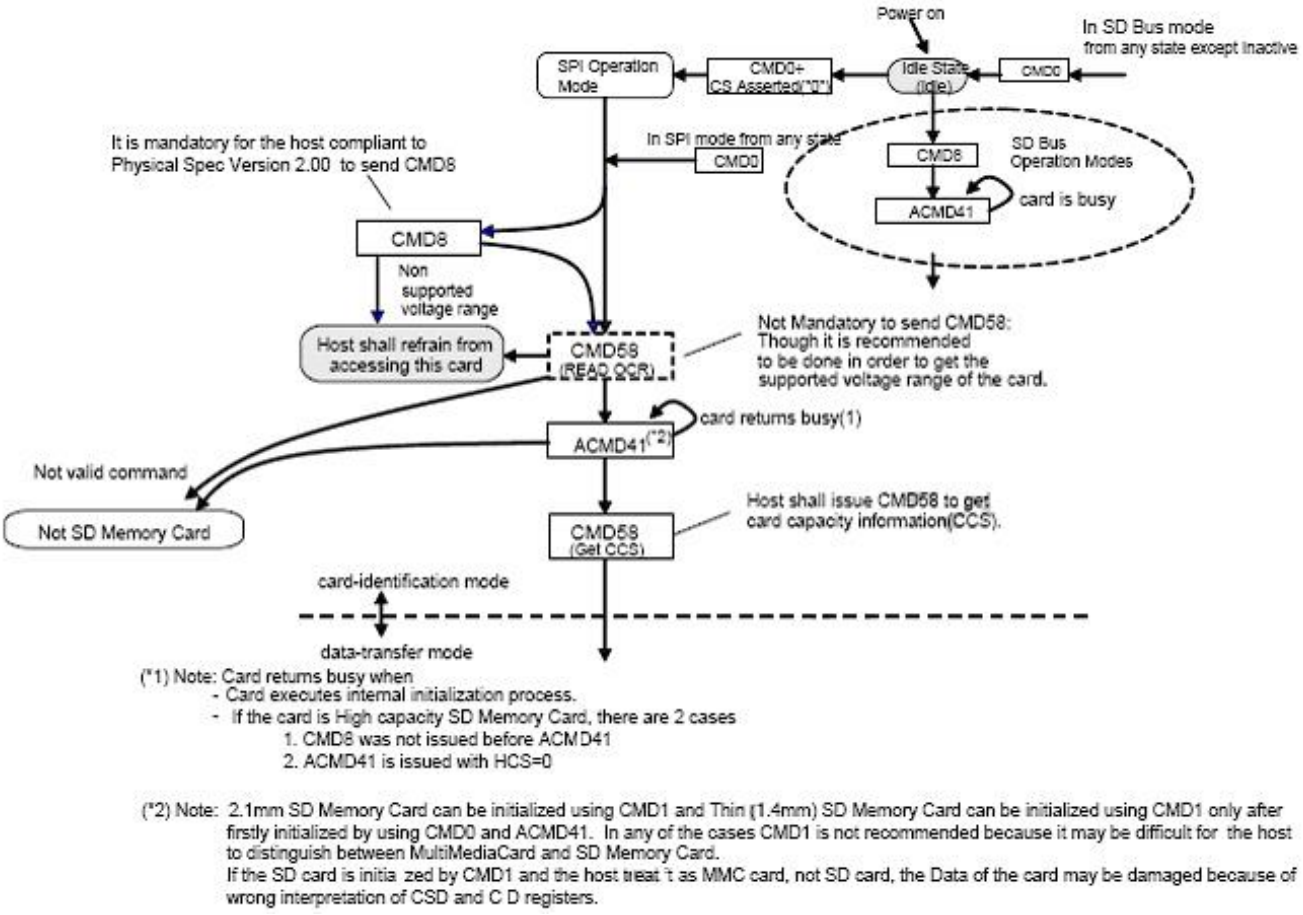


Figure 7-2: state diagram (SPI mode)

7.3.1 Mode Selection and Initialization

The SD Card is powered up in the SD mode. It will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0). If the card recognizes that the SD mode is required it will not respond to the command and remain in the SD 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 mode is by entering the power cycle. In SPI mode, the SD Card protocol state machine in SD mode is not observed. All the SD Card commands supported in SPI mode are always available. Figure 7-3 shows the initialization sequence of SPI mode. SEND_IF_COND (CMD8) is used to verify SD Memory Card interface operating condition. The argument format of CMD8 is the same as defined in SD mode and the response format of CMD8 is defined in Section 7.3.2.6. The card checks the validity of operating condition by analyzing the argument of CMD8 and the host checks the validity by analyzing the response of CMD8. The supplied voltage is indicated by VHS filed in the argument. The card assumes the voltage specified in VHS as the current supplied voltage. Only 1-bit of VHS shall be set to

1 at any given time. Check pattern is used for the host to check validity of communication between the host and the card. If the card indicates an illegal command, the card is legacy and does not support CMD8. If the card supports CMD8 and can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument. If VCA in the response is set to 0, the card cannot operate on the supplied voltage. If check pattern is not matched, CMD8 communication is not valid. In this case, it is recommended to retry CMD8 sequence.

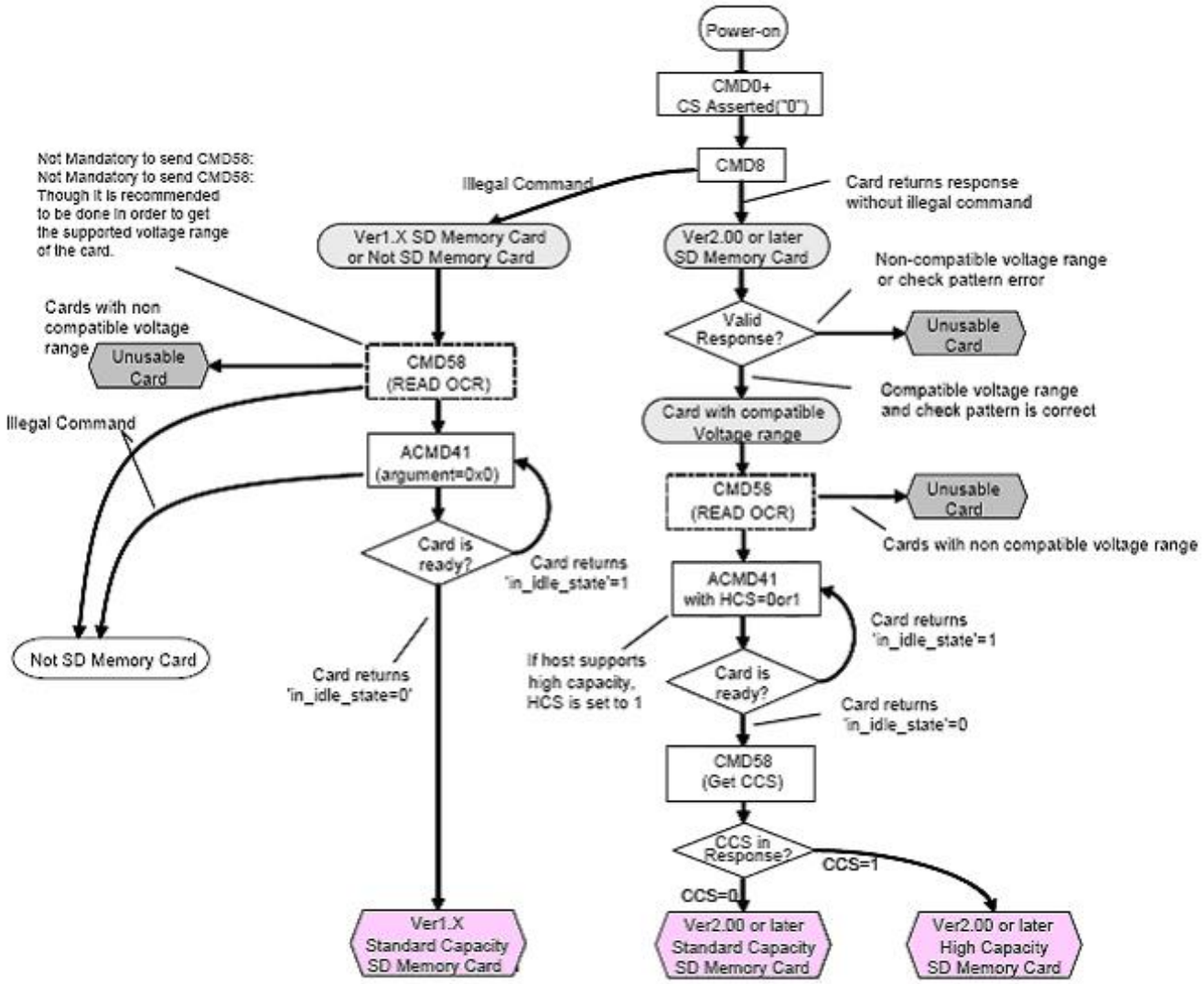


Figure 7-3: SPI Mode Initialization Flow

READ_OCR (CMD58) is designed to provide SD Memory Card hosts with a mechanism to identify cards that do not match the VDD range desired by the host. If the host does not accept voltage range, it shall not proceed further initialization sequence. The levels in the OCR register shall be defined accordingly. SD_SEND_OP_COND (ACMD41) is used to start initialization and to check if the card has completed initialization. It is mandatory to issue CMD8 prior to the first ACMD41. Receiving of CMD8 expands the CMD58 and ACMD41 function; HCS (High Capacity Support) in the argument of ACMD41 and CCS (Card Capacity Status) in the response of CMD58. HCS is ignored by the card, which didn't accept CMD8. Standard Capacity SD Memory Card ignores HCS. The "in idle state" bit in the R1 response of ACMD41 is used by the card to inform the host if initialization of ACMD41 is completed.

Setting this bit to “1” indicates that the card is still initializing. Setting this bit to “0” indicates completion of initialization. The host repeatedly issues ACMD41 until this bit is set to “0”. The card checks the HCS bit in the OCR only at the first ACMD41. While repeating ACMD41, the host shall not issue another command except CMD0. After initialization is completed, the host should get CCS information in the response of CMD58. CCS is valid when the card accepted CMD8 and after the completion of initialization. CCS=1 means that the card is a High Capacity SD Memory Card. CCS=0 means that the card is a Standard Capacity SD.

7.3.2 Bus Transfer Protection

Every SD Card token transferred on the bus is protected by CRC bits. In SPI mode, the SD 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 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).

7.3.3 Data Read

The SPI mode supports single block read and Multiple Block read operations (CMD17 or CMD18 in the SD Card protocol). 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 7-4).

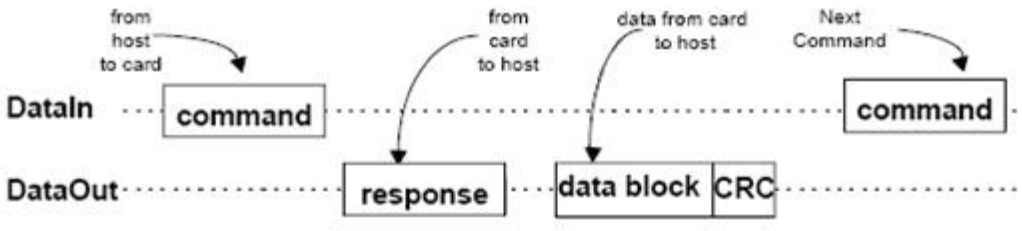


Figure 7-4: Single Block Read operation

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 7-5 shows a data read operation which terminated with an error token rather than a data block.

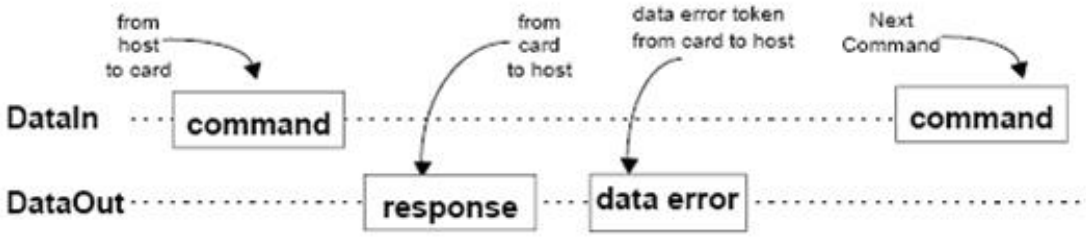


Figure 7-5: Read operation - data error

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 Card operation mode).

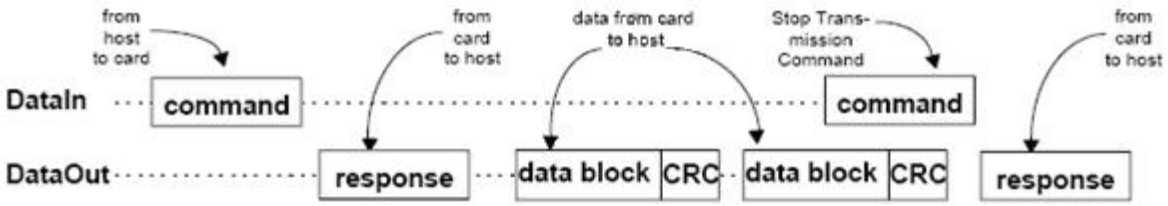


Figure 7-6: Multiple Block Read operation

7.3.4 Data Write

In SPI mode the SD Card supports single block and Multiple block write commands. Upon reception of a valid write command (CMD24 or CMD25 in the SD Card 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.

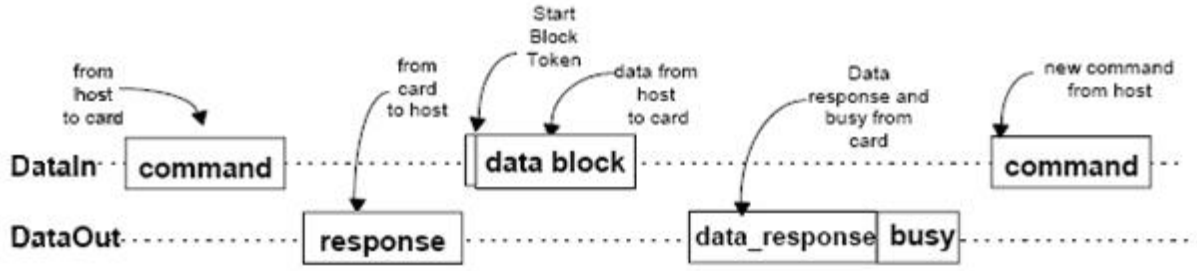


Figure 7-7: Single Block Write operation

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).

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.

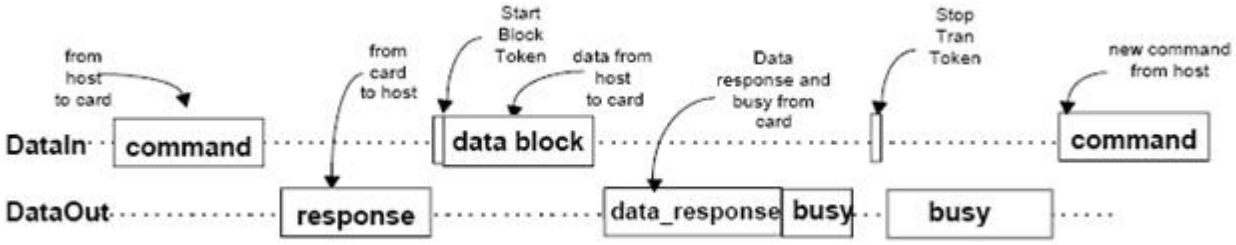


Figure 7-8: Multiple Block Write operation

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 formats on the card. It is in the responsibility of the host to prevent it.

7.3.5 Erase & Write Protect Management

The erase and write protect management procedures in the SPI mode are identical to those of the SD mode. While the card is erasing or changing the write protection bits of the predefined sector list, it will be in a busy state and hold the DataOut line low. Figure 7-9 illustrates a 'no data' bus transaction with and without busy signaling.

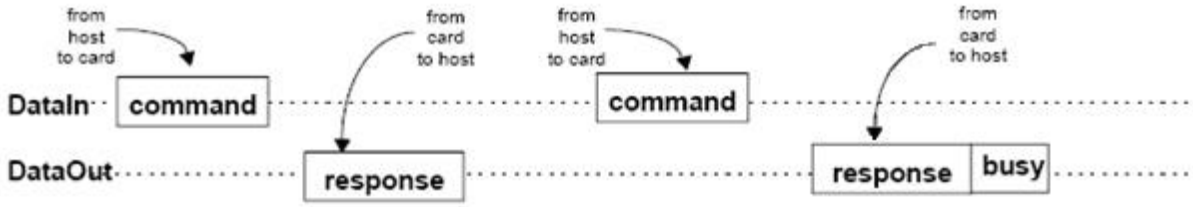


Figure 7-9: 'No data' operations

7.3.6 Read CID/CSD Registers

Unlike the SD Card 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 TAAC since this value is stored in the card's CSD. Therefore the standard response time-out value (NCR) is used for read latency of the CSD register.

7.3.7 Reset Sequence

The SD 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 valid host commands are ACMD41 (SD_SEND_OP_COND), CMD58 (READ_OCR) and CMD59 (CRC_ON_OFF). CMD1 (SEND_OP_COND) is also valid - that means that in SPI mode CMD1 and ACMD41 have the same behavior. 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.

The host must poll the card (by repeatedly sending CMD1 or ACMD41) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card completed its initialization processes and is ready for the next command.

In SPI mode, as opposed to SD mode, ACMD41 (or CMD1 as well) has no operands and does not return the contents of the OCR register. Instead, the host may 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 cards that do not support its voltage range. The usage of CMD58 is not restricted to the initializing phase only, but can be issued at any time.

ATP TAIWAN(HQ)
TEL: +886-2-2659-6368
FAX: +886-2-2659-4982
sales-apac@atpinc.com

ATP USA
TEL: +1-408-732-5000
FAX: +1-408-732-5055
sales@atpinc.com

ATP JAPAN
TEL: +81-03-6206-8097
FAX: +81-03-6206-8098
sales-japan@atpinc.com

ATP EUROPE
TEL: +49-89-374-9999-0
FAX: +49-89-374-9999-29
sales-europe@atpinc.com

ATP CHINA
TEL: +86-21-5080-2220
FAX: +86-21-5080-2219
sales@cn.atpinc.com

www.atpinc.com

LifeElectronics
Живое партнерство

ООО «ЛайфЭлектроникс» **“LifeElectronics” LLC**

ИНН 7805602321 КПП 780501001 Р/С 40702810122510004610 ФАКБ "АБСОЛЮТ БАНК" (ЗАО) в г.Санкт-Петербурге К/С 30101810900000000703 БИК 044030703

Компания «Life Electronics» занимается поставками электронных компонентов импортного и отечественного производства от производителей и со складов крупных дистрибьюторов Европы, Америки и Азии.

С конца 2013 года компания активно расширяет линейку поставок компонентов по направлению коаксиальный кабель, кварцевые генераторы и конденсаторы (керамические, пленочные, электролитические), за счёт заключения дистрибьюторских договоров

Мы предлагаем:

- *Конкурентоспособные цены и скидки постоянным клиентам.*
- *Специальные условия для постоянных клиентов.*
- *Подбор аналогов.*
- *Поставку компонентов в любых объемах, удовлетворяющих вашим потребностям.*

- *Приемлемые сроки поставки, возможна ускоренная поставка.*
- *Доставку товара в любую точку России и стран СНГ.*
- *Комплексную поставку.*
- *Работу по проектам и поставку образцов.*
- *Формирование склада под заказчика.*

- *Сертификаты соответствия на поставляемую продукцию (по желанию клиента).*
- *Тестирование поставляемой продукции.*
- *Поставку компонентов, требующих военную и космическую приемку.*
- *Входной контроль качества.*
- *Наличие сертификата ISO.*

В составе нашей компании организован Конструкторский отдел, призванный помогать разработчикам, и инженерам.

Конструкторский отдел помогает осуществить:

- *Регистрацию проекта у производителя компонентов.*
- *Техническую поддержку проекта.*
- *Защиту от снятия компонента с производства.*
- *Оценку стоимости проекта по компонентам.*
- *Изготовление тестовой платы монтаж и пусконаладочные работы.*

Тел: +7 (812) 336 43 04 (многоканальный)
Email: org@lifeelectronics.ru

www.lifeelectronics.ru