SanDisk Secure Digital Card

Product Manual

Version 1.9

Document No. 80-13-00169

December 2003

SanDisk Corporation

Corporate Headquarters • 140 Caspian Court • Sunnyvale, CA 94089

Phone (408) 542-0500 • Fax (408) 542-0503

www.sandisk.com

SanDisk® Corporation general policy does not recommend the use of its products in life support applications where in a failure or malfunction of the product may directly threaten life or injury. Per SanDisk Terms and Conditions of Sale, the user of SanDisk products in life support applications assumes all risk of such use and indemnifies SanDisk against all damages. See "Limited Warranty and Disclaimer of Liability."

This document is for information use only and is ***subject to change without prior notice****. SanDisk Corporation assumes no responsibility for any errors that may appear in this document, nor for incidental or consequential damages resulting from the furnishing, performance or use of this material. No part of this document may be reproduced, transmitted, transcribed, stored in a retrievable manner or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written consent of an officer of SanDisk Corporation.*

SanDisk and the SanDisk logo are registered trademarks of SanDisk Corporation.

Product names mentioned herein are for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.

SanDisk products are covered or licensed under one or more of the following U.S. Patent Nos. 5,070,032; 5,095,344; 5,168,465; 5,172,338; 5,198,380; 5,200,959; 5,268,318; 5,268,870; 5,272,669; 5,418,752; 5,602,987. Other U.S. and foreign patents awarded and pending.

Lit. No. 80-13-00169 Rev. 1.9 12/2003 *Printed in U.S.A.*

Revision History

- *Revision 1.0 dated 6/2001–initial release.*
- *Revision 1.1 dated 11/2001–minor editorial and technical changes.*
- *Revision 1.2 dated 6/2002–minor editorial and technical changes.*
- *Revision 1.3 dated 7/2002–minor editorial and technical changes.*
- *Revision 1.4 dated 11/2002—minor editorial change.*
- *Revision 1.5 dated 3/2003— changed power requirements in Section 2.3, Table 2.3; updated addresses in Appendix A; adjusted footers and front matter.*
- *Revision 1.6 dated 8/2003—added 512- and 1024-Mb capacities; updated "Limited Warranty" appendix; added "Disclaimer of Liability" appendix.*
- *Revision 1.7 dated 9/2003—minor revisions; added appnote as Appendix A*
- *Revision 1.8 dated 11/2003—changed VDD read/write values in Section 2 and Table 3-10*
- *Revision 1.9 dated 12/2003—changed VDD read/write values in Table 3-10*

Table of Contents

1. Introduction to the SD Card

The Secure Digital Card is a flash-based memory card that is specifically designed to meet the security, capacity, performance and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The SD Card includes a copyright protection mechanism that complies with the security of the SDMI standard, and is faster and capable of higher Memory capacity. The SD Card security system uses mutual authentication and a “new cipher algorithm” to protect from illegal usage of the card content. A non-secured access to the user‘s own content is also available. The physical form factor, pin assignment and data transfer protocol are forward compatible with the SD Card, with some additions.

The SD Card communication is based on an advanced nine-pin interface (Clock, Command, 4xData and 3xPower lines) designed to operate in a low voltage range. The communication protocol is defined as part of this specification. The SD Card host interface supports regular MultiMediaCard operation as well. In other words, MultiMediaCard forward compatibility was kept. Actually the main difference between SD Card and MultiMediaCard is the initialization process. The SD Card specifications were originally defined by MEI (Matsushita Electric Company), Toshiba Corporation and SanDisk Corporation. Currently, the specifications are controlled by the Secure Digital Association (SDA). The SanDisk SD Card was designed to be compatible with the SD Card Physical Specification.

The SD Card interface allows for easy integration into any design, regardless of microprocessor used. For compatibility with existing controllers, the SanDisk SD Card offers, in addition to the SD Card interface, an alternate communication protocol, which is based on the SPI standard.

The current SD Card provides up to 1024 million bytes of memory using flash memory chips, which were designed especially for use in mass storage applications. In addition to the mass storage specific flash memory chip, the SD Card includes an on-card intelligent controller which manages interface protocols, security algorithms for copyright protection, data storage and retrieval, as well as Error Correction Code (ECC) algorithms, defect handling and diagnostics, power management and clock control.

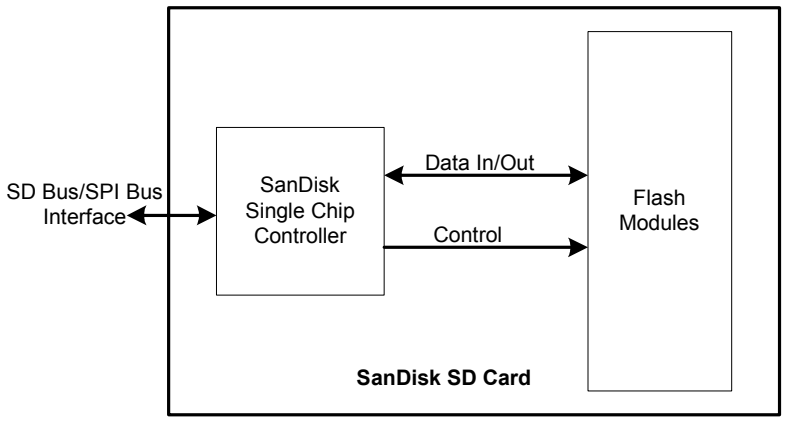


Figure 1-1. SD Card Block Diagram

1.1. Scope

This document describes the key features and specifications of the SD Card, as well as the information required to interface this product to a host system.

1.2. Product Models

The SD Card is available in the capacities shown in Table 1-1.

Table 1-1. SD Card Capacities

Model No.	Capacities
SDSDB-16	16 MB
SDSDB-32	32 MB
SDSDJ-64	64 MB
SDSDJ-128	128 MB
SDSDJ-256	256 MB
SDSDJ-512	512 MB
SDSDJ-1024	1024 MB

SDSDB = Binary NAND technology.
SDSDJ = Multi Level Cell (MLC) NAND technology.

1.3. System Features

The SD Card provides the following features:

- Up to 1-GB of data storage.
- SD Card protocol compatible.
- Supports SPI Mode.
- Targeted for portable and stationary applications for secured (copyrights protected) and non-secured data storage.
- Voltage range:
  - Basic communication (CMD0, CMD15, CMD55, ACMD41): 2.0—3.6V.
  - Other commands and memory access: 2.7—3.6V.
- Variable clock rate 0—25 MHZ.
- Up to 12.5 MB/sec data transfer rate (using 4 parallel data lines).
- Maximum data rate with up to 10 cards.
- Correction of memory field errors.
- Copyrights Protection Mechanism—Complies with highest security of SDMI standard.
- Password Protected of Cards (not on all models).
- Write Protect feature using mechanical switch.
- Built-in write protection features (permanent and temporary).
- Card Detection (Insertion/Removal).
- Application specific commands.

- Comfortable erase mechanism.

The performance of the communication channel is described in Table 1-2.

Table 1-2. SD Bus/SPI Bus Comparison

SD Card Using SD Bus	SD Card Using SPI Bus
Six-wire communication channel (clock, command, 4 data lines).	Three-wire serial data bus (Clock, dataIn, dataOut) + card specific CS signal (hardwired card selection).
Error-protected data transfer.	Optional non-protected data transfer mode available.
Single or multiple block oriented data transfer.	Single or multiple block oriented data transfer.

1.4. SD Card Standard

SanDisk SD Cards are fully compatible with the following *SD Card Physical Layer Specification* standard:

The SD Card Physical Layer System Specification, Version 1.01

This specification may be obtained from:

SD Card Association
53 Muckelemi St.
P.O. Box 189
San Juan Bautista, CA 95045-0189
USA
Phone: 831-623-2107
Fax: 831-623-2248
Email: rcreech@sdcard.org
http://www.sdcard.org

1.5. Functional Description

SanDisk SD Cards contain a high level, intelligent subsystem as shown in Figure 1-1. This intelligent (microprocessor) subsystem provides many capabilities not found in other types of memory cards. These capabilities include:

- Host independence from details of erasing and programming flash memory.
- Sophisticated system for managing defects (analogous to systems found in magnetic disk drives).
- Sophisticated system for error recovery including a powerful error correction code (ECC).
- Power management for low power operation.

1.5.1. Flash Technology Independence

The 512-byte sector size of the SD Card is the same as that in an IDE magnetic disk drive. To write or read a sector (or multiple sectors), the host computer software simply issues a Read or Write command to the SD Card. This command contains the address. The host software then waits for the command to complete. The host software does not get involved in the details of how the flash memory is erased, programmed or read. This is extremely important as flash devices are expected to get more and more complex in the future. Because the SD Card uses an intelligent on-board controller, the host system software will not require changing as new flash memory evolves. In other words, systems that support the SD Card today will be able to access future SanDisk SD Cards built with new flash technology without having to update or change host software.

1.5.2. Defect and Error Management

SD Cards contain a sophisticated defect and error management system. This system is analogous to the systems found in magnetic disk drives and in many cases offers enhancements. For instance, disk drives do not typically perform a read after write to confirm the data is written correctly because of the performance penalty that would be incurred. SD Cards do a read after write under margin conditions to verify that the data is written correctly. In the rare case that a bit is found to be defective, SD Cards replace this bad bit with a spare bit within the sector header. If necessary, SD Cards will even replace the entire sector with a spare sector. This is completely transparent to the host and does not consume any user data space.

The SD Card's soft error rate specification is much better than the magnetic disk drive specification. In the extremely rare case a read error does occur, SD Cards have innovative algorithms to recover the data. This is similar to using retries on a disk drive but is much more sophisticated. The last line of defense is to employ a powerful ECC to correct the data. If ECC is used to recover data, defective bits are replaced with spare bits to ensure they do not cause any future problems. These defect and error management systems coupled with the solid-state construction give SD Cards unparalleled reliability.

1.5.3. Copyright Protection

A detailed description of the Copyright Protection mechanism and related security SD Card commands can be found in the SD Card Security Specification document from the SD Card Association. All SD Card security related commands operate in the data transfer mode.

As defined in the SDMI specification, the data content that is saved in the card is saved already encrypted and it passes transparently to and from the card. No operation is done on the data and there is no restriction to read the data at any time. Associated with every data packet (song, for example) that is saved in the unprotected memory there is a special data that is saved in a protected memory area. For any access (any Read, Write or Erase command) from/to the data in the protected area. For an authentication procedure is done between the card and the connected device, either the LCM (PC for example) or the PD (portable device, such as SD player). After the authentication process passes, the card is ready to accept or give data from/to the connected device. While the card is in the secured mode of operation (after the authentication succeeded) the argument and the associated data that is sent to the card or read from the card are encrypted. At the end of the Read, Write or Erase operation, the card gets out automatically of its secured mode.

1.5.4. Endurance

SanDisk SD Cards have an endurance specification for each sector of 100,000 writes typical (reading a logical sector is unlimited). This far exceeds what is typically required in nearly all applications of SD Cards. For example, even very heavy use of the SD Card in cellular phones, personal communicators, pagers and voice recorders will use only a fraction of the total endurance over the typical device's lifetime. For instance, it would take over 10 years to wear out an area on the SD Card on which a file of any size (from 512 bytes to maximum capacity) was rewritten 3 times per hour, 8 hours a day, 365 days per year.

With typical applications, the endurance limit is not of any practical concern to the vast majority of users.

1.5.5. Wear Leveling

Wear-leveling is an intrinsic part of the Erase Pooling functionality of SD using NAND memory. The SD Card's Wear Level command is supported as a NOP operation to maintain backward compatibility with existing software utilities.

1.5.6. Using the Erase Command

The Erase (sector or group) command provides the capability to substantially increase the write performance of the SD Card. Once a sector has been erased using the Erase command, a write to that sector will be much faster. This is because a normal write operation includes a separate sector erase prior to write.

1.5.7. Automatic Sleep Mode

A unique feature of the SanDisk SD Card (and other SanDisk products) is automatic entrance and exit from sleep mode. Upon completion of an operation, the SD Card will enter the sleep mode to conserve power if no further commands are received within 5msec. The host does not have to take any action for this to occur. In most systems, the SD Card is in sleep mode except when the host is accessing it, thus conserving power.

When the host is ready to access the SD Card and it is in sleep mode, any command issued to the SD Card will cause it to exit sleep and respond.

1.5.8. Hot Insertion

Support for hot insertion will be required on the host but will be supported through the connector. Connector manufacturers will provide connectors that have power pins long enough to be powered before contact is made with the other pins. Please see connector data sheets for more details. This approach is similar to that used in PCMCIA and MMCA devices to allow for hot insertion.

1.5.9. SD Card—SD Bus Mode

The following sections provide valuable information on the SC Card in SD Bus mode.

1.5.9.1. SD Card Standard Compliance

The SD Card is fully compliant with SD Card Physical Layer Standard Specification V1.01. The structure of the Card Specific Data (CSD) register is compliant with CSD Structure 1.0.

1.5.9.2. Negotiating Operation Conditions

The SD Card supports the operation condition verification sequence defined in the SD Card standard specifications. Should the SD Card host define an operating voltage range, which is not supported by the SD Card it will put itself in an inactive state and ignore any bus communication. The only way to get the card out of the inactive state is by powering it down and up again.

In Addition the host can explicitly send the card to the inactive state by using the GO_INACTIVE_STATE command.

1.5.9.3. Card Acquisition and Identification

The SD Card bus is a single master (SD Card host application) and multi-slaves (cards) bus. The Clock and Power lines are common to all cards on the bus. During the identification process, the host accesses each card separately through its own command lines. The SD Card's CID register is pre-programmed with a unique card identification number, which is used during the identification procedure.

In addition, the SD Card host can read the card's CID register using the READ_CID SD Card command. The CID register is programmed during the SD Card testing and formatting procedure, on the manufacturing floor. The SD Card host can only read this register and not write to it.

An internal pull-up resistor on the DAT3 line may be used for card detection (insertion/removal). The resistor can be disconnected during data transfer (using ACMD42). Additional practical card detection methods can be found in SD Physical Specification's Application Notes given by the SDA.

1.5.9.4. Card Status

The card status is separated into the following two fields:

- **Card Status** is stored in a 32-bit status register that is sent as the data field in the card respond to host commands. Status register provides information about the card's current state and completion codes for the last host command. The card status can be explicitly read (polled) with the SEND_STATUS command.
- **SD_Status** is stored in 512 bits that are sent as a single data block after it was requested by the host using the SD_STATUS (ACMD13) command. SD_STATUS contains extended status bits that relate to BUS_WIDTH, security related bits and future specific applications.

1.5.9.5. Memory Array Partitioning

The basic unit of data transfer to/from the SD Card is one byte. All data transfer operations which require a block size always define block lengths as integer multiples of bytes. Some special functions need other partition granularity. Figure 1-2 shows the Memory Array Partitioning.

For block-oriented commands, the following definition is used:

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

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

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

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

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

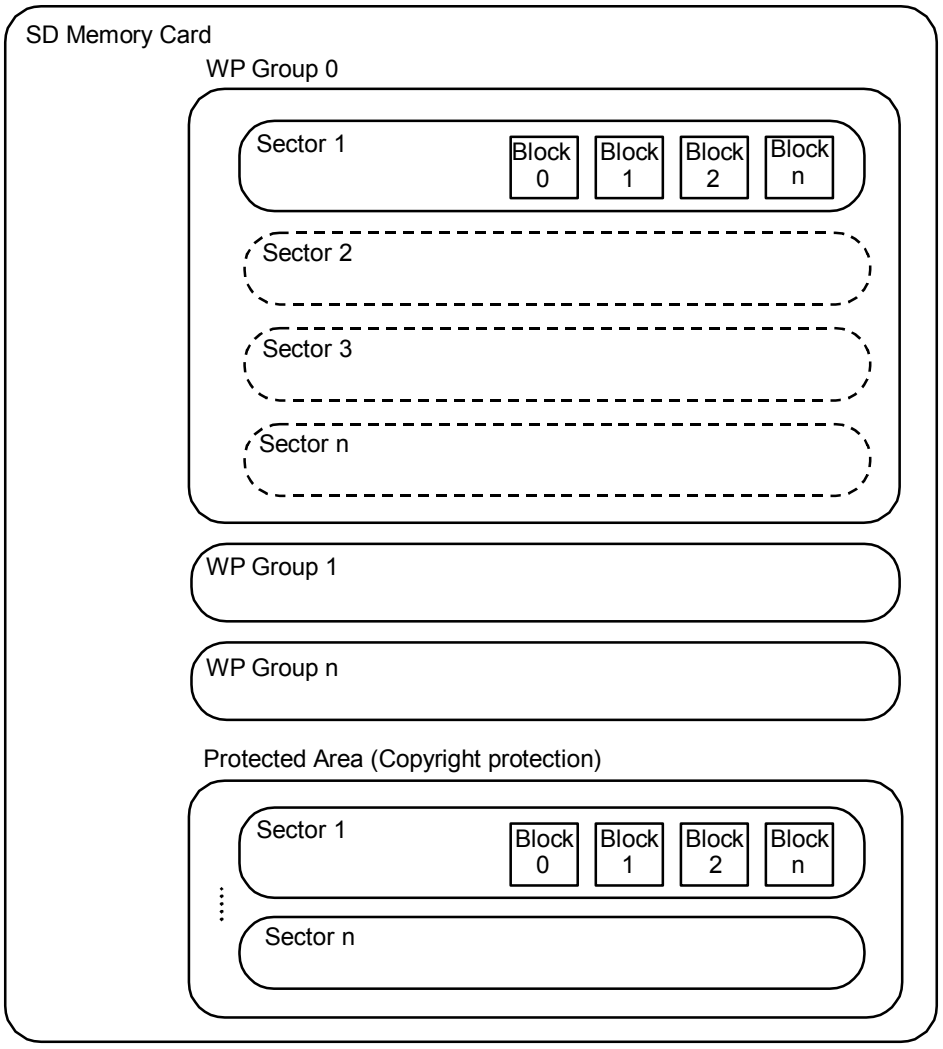


Figure 1-2. Memory Array Partitioning

Table 1-3. Memory Array Structures Summary

Part Number	Block Size (Byte)	Data Area + Protected size (Blocks)	Protected Area size (Blocks)	User Area (Blocks)
SDSDJ-1024	512	2,004,224	20,480	1,983,744
SDSDJ-512	512	1,001,104	10,240	940,864
SDSDJ-256	512	499,456	5,376	494,080
SDSDJ-128	512	248,640	2,624	246,016
SDSDJ-64	512	123,232	1,376	121,856
SDSDB-32	512	60,512	736	59,776
SDSDB-16	512	29,152	352	28,800

NOTE: All measurements are in units per card.

Protected Area—The part of the Card that relates to the secured copyright management and has separate DOS partitioning including sectors and blocks. The card write protection mechanism does not effect this area.

1.5.9.6. Read and Write Operations

The SD Card supports two read/write modes as shown in Figure 1-3.

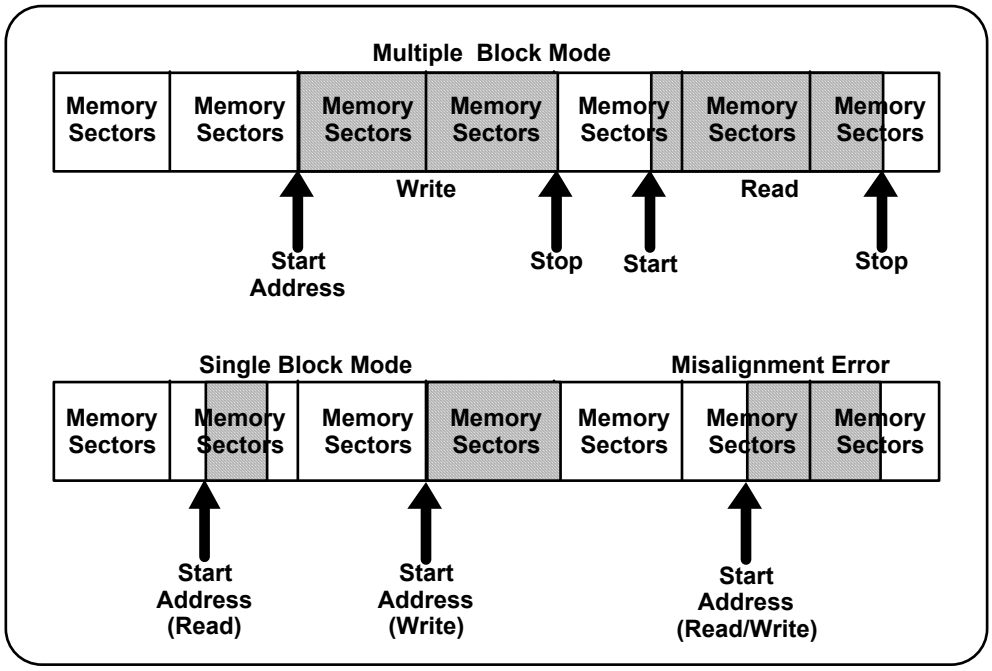


Figure 1-3. Data Transfer Formats

Single Block Mode

In this mode the host reads or writes one data block in a pre-specified length. The data block transmission is protected with 16-bit CRC that is generated by the sending unit and checked by the receiving unit.

The block length for read operations is limited by the device sector size (512 bytes) but can be as small as a single byte. Misalignment is not allowed. Every data block must be contained in a single physical sector. The block length for write operations must be identical to the sector size and the start address aligned to a sector boundary.

Multiple Block Mode

This mode is similar to the single block mode, but the host can read/write multiple data blocks (all have the same length) which will be stored or retrieved from contiguous memory addresses starting at the address specified in the command. The operation is terminated with a stop transmission command.

Misalignment and block length restrictions apply to multiple blocks as well and are identical to the single block read/write operations.

1.5.9.7. Data Transfer Rate

The SD Card can be operated using either a single data line (DAT0) or four data lines (DAT0-DAT3) for data transfer. The maximum data transfer rate for a single data line is 25 Mbit per second and for four data lines it is 100 Mbit (12 MB) per second.

1.5.9.8. Data Protection in the Flash Card

Every sector is protected with an Error Correction Code (ECC). The ECC is generated (in the memory card) when the sectors are written and validated when the data is read. If defects are found, the data is corrected prior to transmission to the host.

1.5.9.9. Erase

The smallest erasable unit in the SD Card is a sector. In order to speed up the erase procedure, multiple sectors can be erased at the same time. To facilitate selection, a first command with the starting address is followed by a second command with the final address, and all sectors within this range will be selected for erase.

1.5.9.10. Write Protection

Two-card level write protection options are available: permanent and temporary. Both can be set using the PROGRAM_CSD command (see below). The permanent write protect bit, once set, cannot be cleared. This feature is implemented in the SD Card controller firmware and not with a physical OTP cell.

NOTE: Use the Write Protect (WP) Switch located on the card's side edge to prevent the host from writing to or erasing data on the card. The WP switch does not have any influence on the internal Permanent or Temporary WP bits in the CSD.

1.5.9.11. Copy Bit

The content of a SD Card can be marked as an original or a copy using the copy bit in the CSD register. Once the Copy bit is set (marked as a copy) it cannot be cleared. The Copy bit of the SD Card is programmed (during test and formatting on the manufacturing floor) as a copy. The SD Card can be purchased with the copy bit set (copy) or cleared, indicating the card is a master. This feature is implemented in the SD Card controller firmware and not with a physical OTP cell.

1.5.9.12. The CSD Register

All the configuration information of the SD Card is stored in the CSD register. The MSB bytes of the register contain manufacturer data and the two least significant bytes contain the host-controlled data, the card Copy, write protection and the user file format indication.

The host can read the CSD register and alter the host controlled data bytes using the SEND_CSD and PROGRAM_CSD commands.

1.5.10. SD Card—SPI Mode

The SPI mode is a secondary communication protocol for SD Cards. This mode is a subset of the SD Card protocol, designed to communicate with an SPI channel, commonly found in Motorola's (and lately a few other vendors') microcontrollers.

1.5.10.1. Negotiating Operating Conditions

The operating condition negotiation function of the SD Card bus is supported differently in SPI mode by using the READ_OCR (CMD58) command. The host shall work within the valid voltage range (2.7 to 3.6 volts) of the card or put the card in inactive state by sending a GO_INACTIVE command to the card.

1.5.10.2. Card Acquisition and Identification

The host must know the number of cards currently connected on the bus. Specific card selection is done via the CS signal (CD/DAT3). The internal pullup resistor on the CD/DAT3 line may be used for card detection (insertion/removal). Additional practical card detection methods can be found in SD Physical Specification's Application Notes given by the SDA.

1.5.10.3. Card Status

In SPI mode, only 16 bits (containing the errors relevant to SPI mode) can be read out of the 32-bit SD Card status register. The SD_STATUS can be read using ACMD13, the same as in SD Bus mode.

1.5.10.4. Memory Array Partitioning

Memory partitioning in SPI mode is equivalent to SD Bus mode. All read and write commands are byte addressable with the limitations given in Section 1.5.9.5.

1.5.10.5. Read and Write Operations

In SPI mode, both single and multiple block data transfer modes are supported.

1.5.10.6. Data Transfer Rate

In the SPI mode, only one data line is used for each direction. The SPI mode data transfer rate is the same as the SD Bus mode data transfer rate when using one data line only (up to 25 Kbits per second).

1.5.10.7. Data Protection in the SD Card

Same as for the SD Card mode.

1.5.10.8. Erase

Same as in SD Card mode.

1.5.10.9. Write Protection

Same as in SD Card mode.

1.5.10.10. Copyright Protection

Same as in SD Card mode.

This page intentionally left blank.

2. Product Specifications

For all the following specifications, values are defined at ambient temperature and nominal supply voltage unless otherwise stated.

2.1. System Environmental Specifications

Table 2-1. System Environmental Specifications

Temperature	Operating: Non-Operating:	-25° C to 85° C -40° C to 85° C
Humidity	Operating: Non-Operating:	25% to 95%, non-condensing 25% to 95%, non-condensing
ESD Protection	Contact Pads: Non Contact Pad Area:	± 4kV, Human body model according to ANSI EOS/ESD-S5.1-1998 ± 8kV (coupling plane discharge) ± 15kV (air discharge) Human body model per IEC61000-4-2

2.2. Reliability and Durability

Table 2-2. Reliability and Durability Specifications

Durability	10,000 mating cycles
Bending	10N
Torque	0.15N.m or ±2.5 deg.
Drop Test	1.5m free fall
UV Light Exposure	UV: 254nm, 15Ws/cm^2 according to ISO 7816-1
Visual Inspection/Shape and Form	No warpage; no mold skin; complete form; no cavities; surface smoothness ≤ -0.1 mm/cm^2 within contour; no cracks; no pollution (oil, dust, etc.)
Minimum Moving Force of WP Switch	40 gf (ensures that the WP switch will not slide while it is inserted in the connector).
WP Switch Cycles	Minimum 1,000 Cycles @ slide force 0.4N to 5N

2.3. Typical Card Power Requirements

Table 2-3. Card Power Requirements

VDD (ripple: max, 60 mV peak to peak)	2.7 V – 3.6 V

(Ta = 25°C @3 V)

	Value	Measurement	Notes
Sleep	250	uA	Max
Read	65	mA	Max
Write	75	mA	Max

2.4. System Performance

Table 2-4. System Performance

	Typical	Maximum
Block Read Access Time Binary Products MLC Products	 1.5msec 10msec	 100msec 100msec
Block Write Access Time Binary Products MLC Products	 24msec 40msec	 250msec 250msec
CMD1 to Ready (after power up)	50msec	500msec
Sleep to Ready	1msec	2msec

NOTES: All values quoted are under the following conditions:

1) Voltage range: 2.7 V to 3.6 V.
2) Temperature range: -25° C to 85° C.
3) Are independent of the SD Card clock frequency.

2.5. System Reliability and Maintenance

Table 2-5. System Reliability and Maintenance Specifications

MTBF	> 1,000,000 hours
Preventive Maintenance	None
Data Reliability	< 1 non-recoverable error in 10^{14} bits read
Endurance	100,000 write/erase cycles (typical)

2.6. Physical Specifications

Refer to Table 2-6 and to Figures 2-1 through 2-3 for SD Card physical specifications and dimensions.

Table 2-6. Physical Specifications

Weight	2.0 g. maximum
Length:	32mm ± 0.1mm
Width:	24mm ± 0.1mm
Thickness:	2.1mm ± 0.15mm (in substrate area only, 2.25mm maximum)

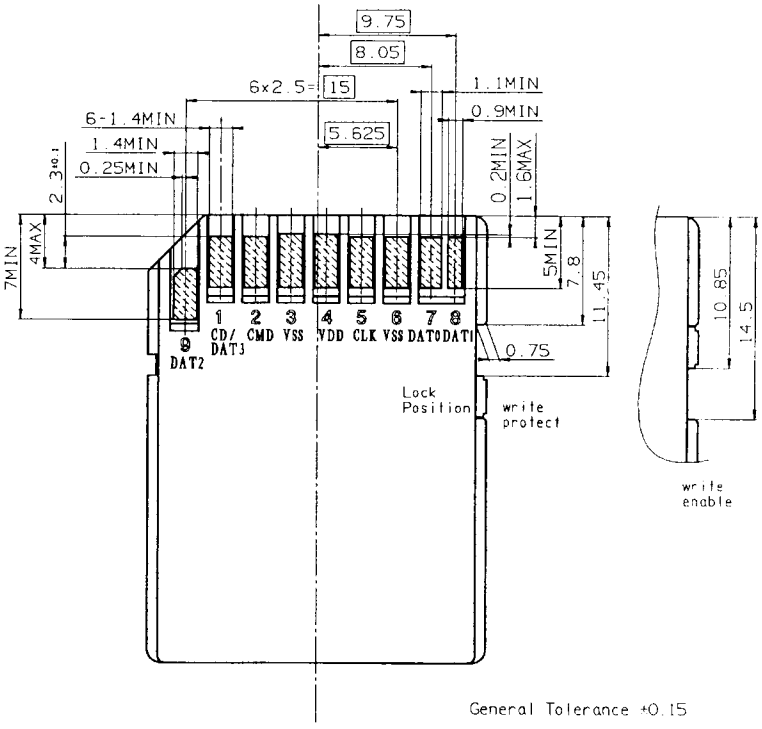


Figure 2-1. SD Card Dimensions

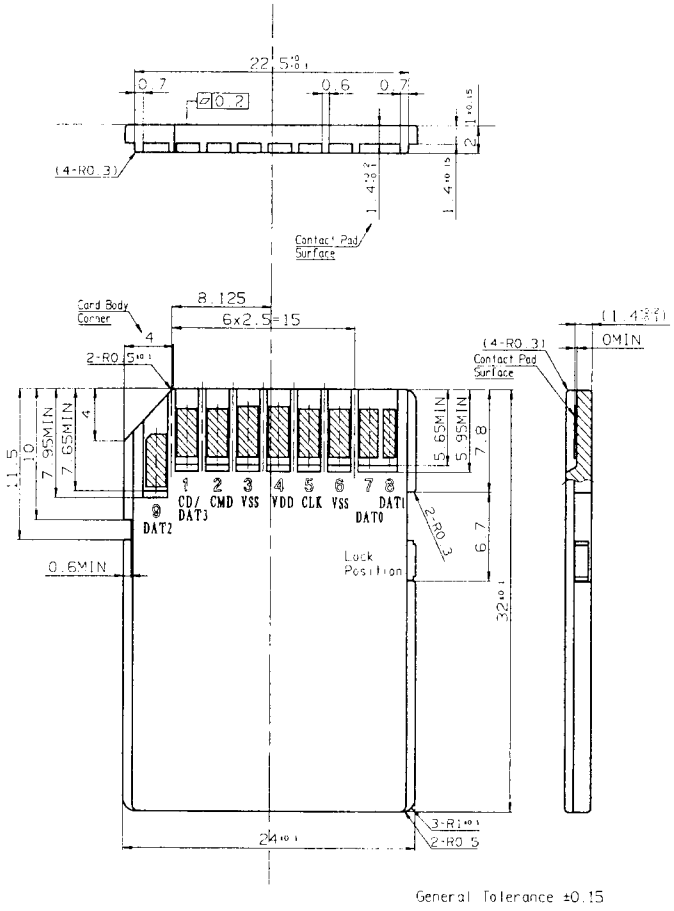
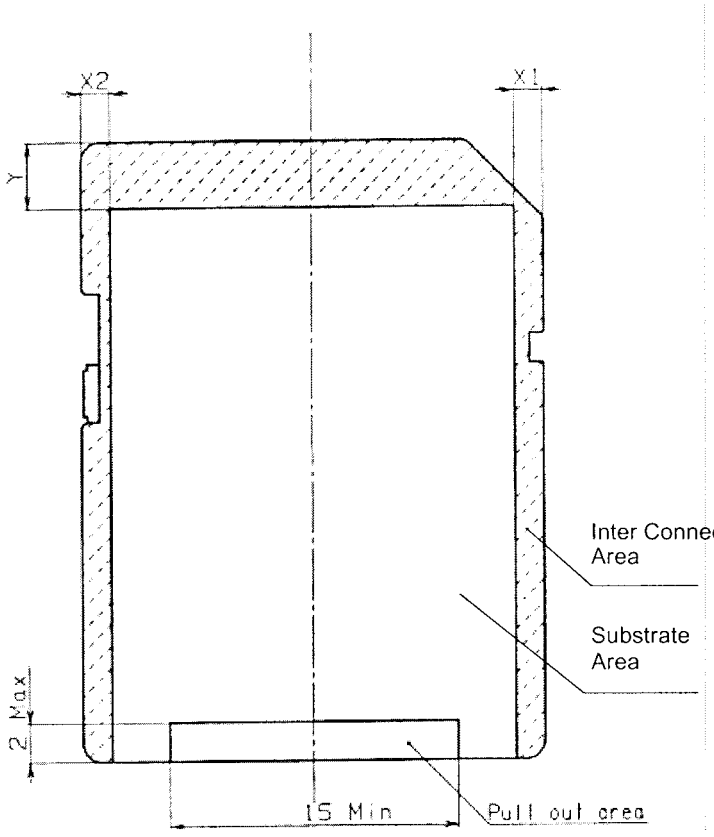


Figure 2-2. SD Card Dimensions



X1	X2	Y1
Min 1.5mm	Min 1.5mm	Min 3.4mm

Figure 2-3. SD Card Dimensions

This page intentionally left blank.

3. SD Card Interface Description

3.1. General Description of Pins and Registers

The SD Card has nine exposed contacts on one side (see Figure 3-1). The host is connected to the SD Card using a dedicated 9-pin connector.

3.1.1. Pin Assignments in SD Card Mode

Table 3-1 lists the pin assignments and definitions in SD Card Mode.

Table 3-1. SD Bus Mode Pad Definition

Pin #	Name	Type[1]	SD Description
1	CD/DAT3[2]	I/O[3]	Card Detect/Data Line [Bit 3]
2	CMD	I/O	Command/Response
3	V_{SS1}	S	Supply voltage ground
4	V_{DD}	S	Supply voltage
5	CLK	I	Clock
6	V_{SS2}	S	Supply voltage ground
7	DAT0	I/O	Data Line [Bit 0]
8	DAT1	I/O	Data Line [Bit 1]
9	DAT2	I/O	Data Line [Bit 2]

NOTES: 1) S=power supply; I=input; O=output using push-pull drivers.
2) The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after the SET_BUS_WIDTH command. It is the responsibility of the host designer to connect external pullup resistors to all data lines even if only DAT0 is to be used. Otherwise, non-expected high current consumption may occur due to the floating inputs of DAT1 & DAT2 (in case they are not used).
3) After power up, this line is input with 50Kohm(+/-20Kohm) pull-up (can be used for card detection or SPI mode selection). The pull-up may be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command.

3.1.2. Pin Assignments in SPI Mode

Table 3-2 lists the pin assignments and definitions in SPI Mode.

Table 3-2. SPI Bus Mode Pad Definition

Pin #	Name	Type[1]	SPI Description
1	CS	I	Chip Select (Active low)
2	DataIn	I	Host to Card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card to Host Data and Status
8	RSV(2)	I	Reserved
9	RSV(2)	I	Reserved

NOTES: 1) S=power supply; I=input; O=output.
2) The 'RSV' pins are floating inputs. It is the responsibility of the host designer to connect external pullup resistors to those lines. Otherwise non-expected high current consumption may occur due to the floating inputs.

Each card has a set of information registers (refer to Table 3-3). Detailed descriptions are provided in Section 3.5.

Table 3-3. SD Card Registers

Name	Width	Description
CID	128	Card identification number: individual card number for identification.
RCA[1]	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	SD Configuration Register: information about the SD Card's special features capabilities.
OCR	32	Operation Condition Register

NOTE: 1) The RCA register is not available in SPI Mode.

The host may reset the cards by switching the power supply off and on again. The card has its own power-on detection circuitry which puts the card into an idle state after the power-on. The card can also be reset by sending the GO_IDLE (CMD0) command.

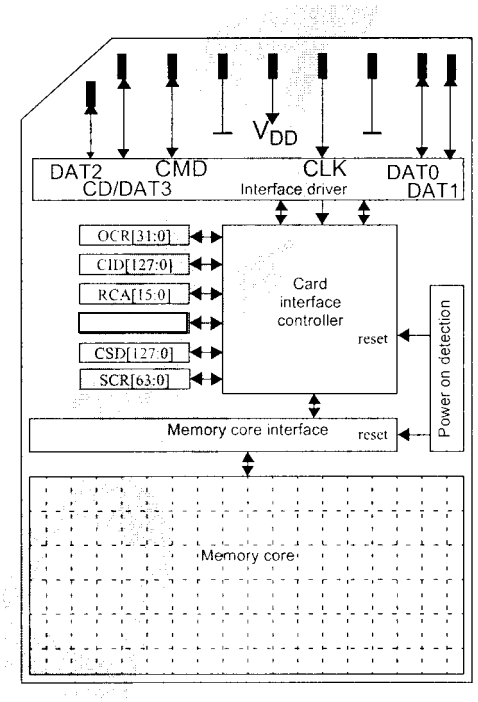


Figure 3-1. SD Card Architecture

3.2. SD Bus Topology

The SD bus has six communication lines and three supply lines:

- **CMD**—Command is a bi-directional signal. (Host and card drivers are operating in push pull mode.)
- **DAT0-3**—Data lines are bi-directional signals. (Host and card drivers are operating in push pull mode.)
- **CLK**—Clock is a host to cards signal. (CLK operates in push pull mode.)
- **VDD**—VDD is the power supply line for all cards.
- **VSS[1:2]**—VSS are two ground lines.

Figure 3-2 shows the bus topology of several cards with one host in SD Bus mode.

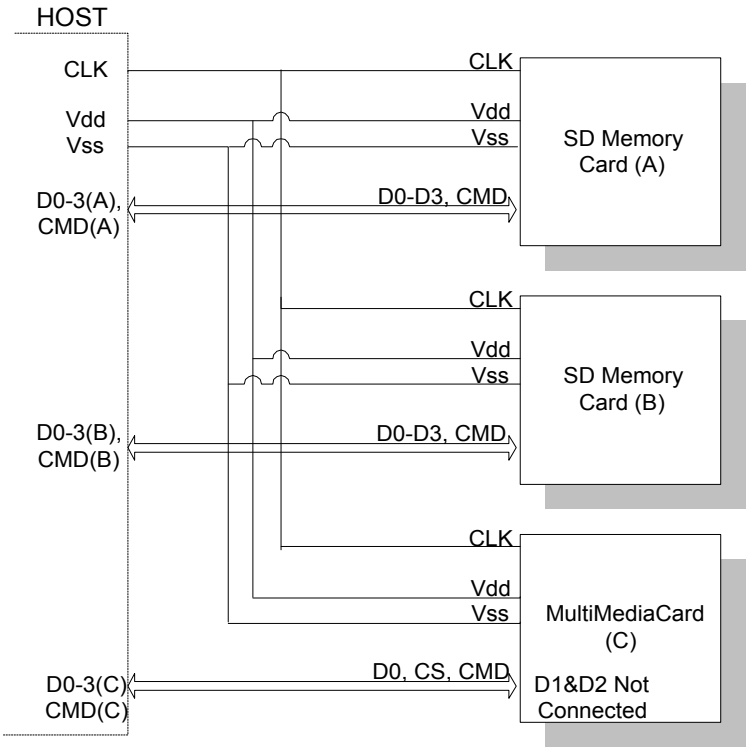


Figure 3-2. SD Card System Bus Topology

During the 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 to each card individually. However, to simplify the handling of the card stack, after initialization, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.

The SD Bus allows dynamic configuration of the number of data lines. After power-up, by default, the SD Card will use only DAT0 for data transfer. After initialization, the host can change the bus width (number of active data lines). This feature allows and easy trade off between hardware cost and system performance.

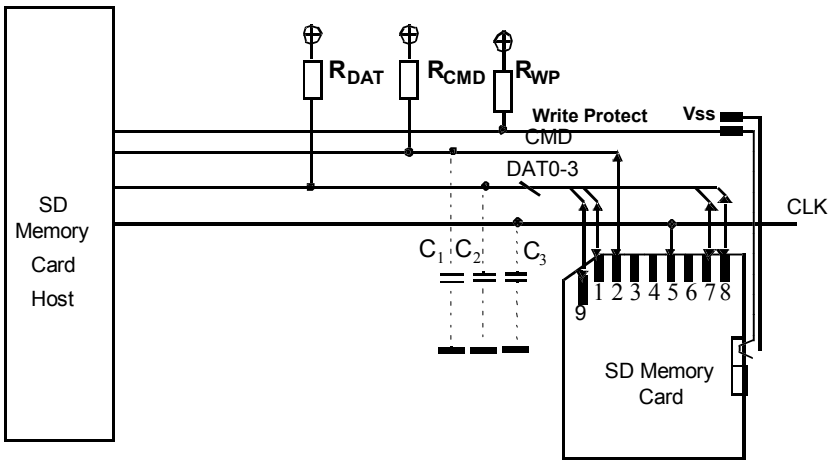


Figure 3-3. Bus Circuitry Diagram

R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and the DAT line against bus floating when no card is inserted or when all card drivers are in a hi-impedance mode. RWP is used for the Write Protect Switch. See Section 5.4.2 for the component values and conditions.

Hot Insertion/Removal

Hot insertion and removal are allowed. The SanDisk SD Card will not be damaged by inserting or removing it into the SD bus even when the power is up:

- The inserted card will be properly reset also when CLK carries a clock frequency f_{PP}.
- Data transfer failures induced by removal/insertion should be detected by the bus master using the CRC codes that suffix every bus transaction.

3.2.1. Power Protection

Cards can be inserted into or removed from the bus without damage. If one of the supply pins (V_{DD} or V_{SS}) is not connected properly, then the current is drawn through a data line to supply the card.

Data transfer operations are protected by CRC codes; therefore, any bit changes induced by card insertion and removal can be detected by the SD bus master. The inserted card must be properly reset also when CLK carries a clock frequency $f_{pp.}$ If the hot insertion feature is implemented in the host, than the host has to withstand a shortcut between V_{DD} and V_{SS} without damage.

3.3. SPI Bus Topology

The SD Card SPI interface is compatible with SPI hosts available on the market. As any other SPI device the 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, which is implemented in the SD Card as well, is byte transfers. All data tokens are multiples of 8-bit bytes and always byte aligned to the CS signal. The SPI standard defines the physical link only and not the complete data transfer protocol. In SPI Bus mode, the SD Card uses a subset of the SD Card protocol and command set.

The SD Card identification and addressing algorithms are replaced by a hardware Chip Select (CS) signal. A card (slave) is selected, for every command, by asserting (active low) the CS signal (see Figure 3-4). The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception is card programming time. At this time the host can de-assert the CS signal without affecting the programming process.

The bi-directional CMD and DAT lines are replaced by uni-directional dataIn and dataOut signals. This eliminates the ability of executing commands while data is being read or written. An exception is the multi read/write operations. The Stop Transmission command can be sent during data read. In the multi block write operation a Stop Transmission token is sent as the first byte of the data block.

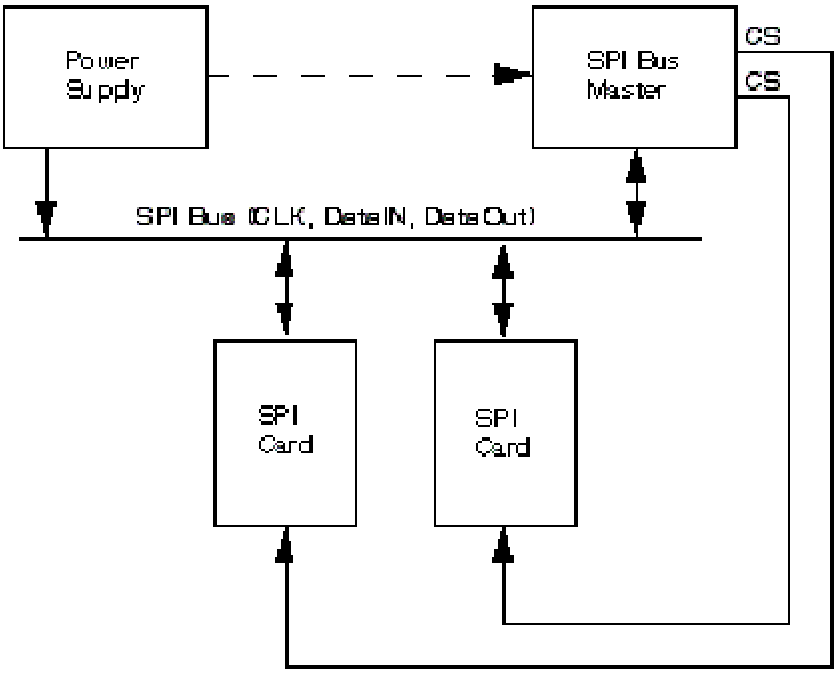


Figure 3-4. SD Card Bus System

3.3.1. Power Protection

Same as for SD Card mode.

3.4. Electrical Interface

The following sections provide valuable information for the electrical interface.

3.4.1. Power-up

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

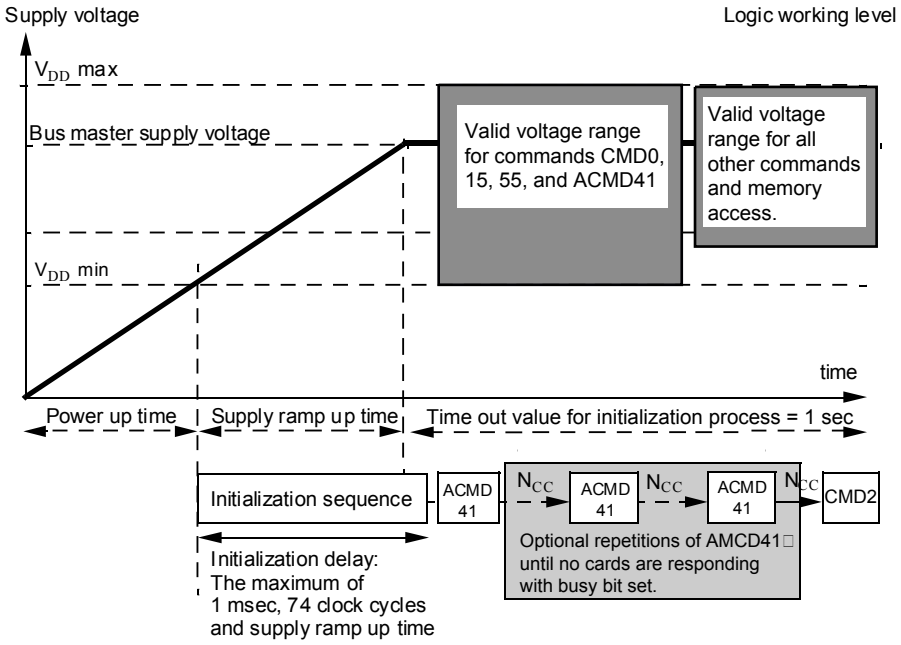


Figure 3-5. Power-up Diagram

After power up, including hot insertion (i.e., inserting a card when the bus is operating) the SD Card enters the *idle state*. During this state the SD Card ignores all bus transactions until ACMD41 is received (ACMD command type shall always precede with CMD55).

ACMD41 is a special synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. Besides the operation voltage profile of the cards, the response to ACMD41 contains a busy flag, indicating that the card is still working on its power-up procedure and is not ready for identification. This bit informs the host that the card is not ready. The host has to wait (and continue to poll the cards, each one on his turn) until this bit is cleared. The maximum period of power up procedure of single card shall not exceed 1 second.

Getting individual cards, as well as the whole SD Card system, out of *idle state* is up to the responsibility of the bus master. Since the power up time and the supply ramp up time depend on application parameters such as the maximum number of SD Card s, the bus length and the power supply unit, the host must ensure that the power is built up to the operating level (the same level which will be specified in ACMD41) before ACMD41 is transmitted.

After power up, the host starts the clock and sends the initializing sequence on the CMD line. This sequence is a contiguous stream of logical '1's. The sequence length is the maximum of 1msec, 74 clocks or the supply-ramp-up-time; the additional 10 clocks (over the 64 clocks after what the card should be ready for communication) is provided to eliminate power-up synchronization problems.

Every bus master shall have the capability to implement ACMD41 and CMD1. CMD1 will be used to ask MultiMediaCards to send their Operation Conditions. In any case the ACMD41 or the CMD1 shall be send separately to each card accessing it through its own CMD line.

3.4.2. Bus Operating Conditions

SPI Mode bus operating conditions are identical to SD Card mode bus operating conditions. Table 3-4 lists the power supply voltages. The CS (chip select) signal timing is identical to the input signal timing (see Figure 3-7).

Table 3-4. Power Supply Voltage

General					
Parameter	**Symbol**	**Min.**	**Max.**	**Unit**	**Remark**
Peak voltage on all lines		-0.3	VDD+0.3	V	
All Inputs					
Input Leakage Current		-10	10	µA	
All Outputs					
Output Leakage Current		-10	10	µA	
Power Supply Voltage					
Parameter	**Symbol**	**Min.**	**Max.**	**Unit**	**Remark**
Supply Voltage	V_{DD}	2.0	3.6	V	CMD0, 15, 55, ACMD41 commands
Supply Voltage		2.7	3.6	V	Except CMD0, 15, 55, ACMD41 commands
Supply voltage differentials (V_{SS1}, V_{SS2})		-0.3	0.3	V	
Power up Time			250	mS	From 0V to V_{DD} Min.

3.4.3. Bus Signal Line Load

The total capacitance CL of the CLK line of the SD Card bus is the sum of the bus master capacitance CHOST, the bus capacitance CBUS itself and the capacitance CCARD of each card connected to this line:

$$CL = CHOST + CBUS + N*CCARD$$

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

Table 3-5. Signal Line's Load

Parameter	Symbol	Min.	Max.	Unit	Remark
Pull-up resistance	R_{CMD} R_{DAT}	10	100	kΩ	To prevent bus floating
Bus signal line capacitance	C_L		250	pF	$f_{PP} \leq 5$ MHz, 21 cards
Bus signal line capacitance	C_L		100	pF	$f_{PP} \leq 20$ MHz, 7 cards
Single card capacitance	C_{CARD}		10	pF	
Maximum signal line inductance			16	nH	$f_{PP} \leq 20$ MHz
Pull-up resistance inside card (pin 1)	R_{DAT3}	10	90	kΩ	May be used for card detection

3.4.4. Bus Signal Levels

As the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage (see Figure 3-6).

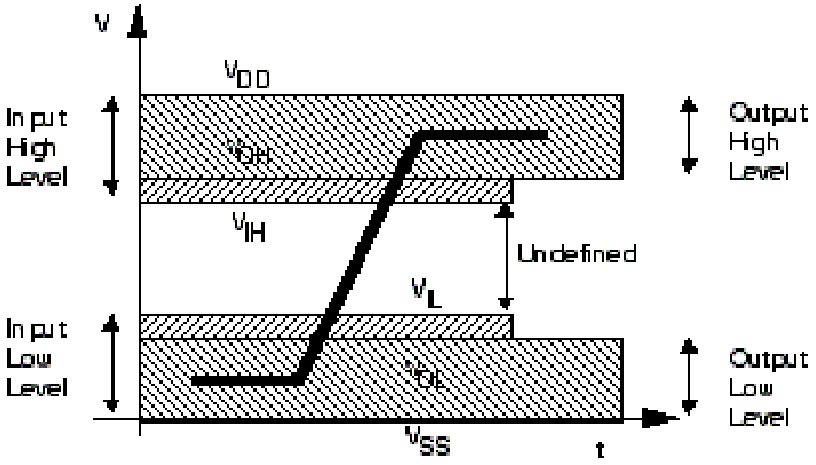


Figure 3-6. Bus Signal Levels

To meet the requirements of the JEDEC specification JESD8-1A, the card input and output voltages shall be within the specified ranges in Table 3-6 for any VDD of the allowed voltage range.

Table 3-6. Input and Output Voltages

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output HIGH voltage	VOH	0.75∗VDD		V	IOH=-100 µA @V_{DD} (min.)
Output LOW voltage	VOL		0.125∗VDD	V	IOL=100 µA @V_{DD} (min.)
Input HIGH voltage	VIH	0.625∗VDD	VDD + 0.3	V	
Input LOW voltage	VIL	VSS-0.3	0.25∗VDD	V	

3.4.5. Bus Timing

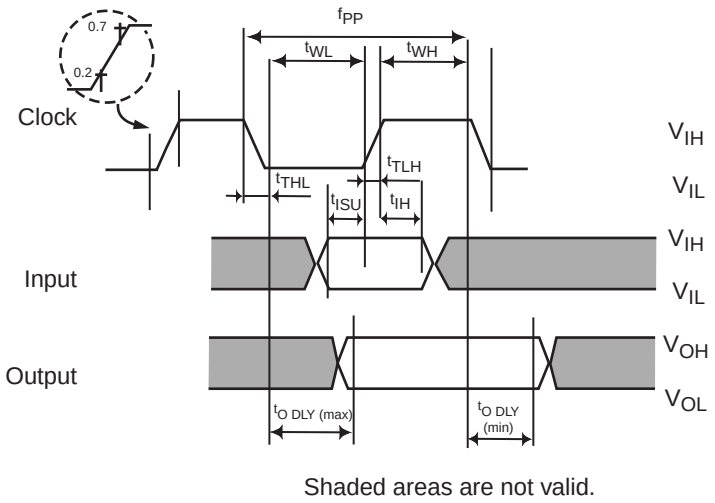


Figure 3-7. Timing Diagram Data Input/Output Referenced to Clock

Table 3-7. Bus Timing

Parameter	Symbol	Min.	Max.	Unit	Remark
Clock CLK (All values are referred to min. (V_{IH}) and max. (V_{IL})					
Clock Frequency Data Transfer Mode	f_{PP}	0	25	MHz	$C_L \leq 100$ pF (7 cards)
Clock Frequency Identification Mode (The low frequency is required for MultiMediaCard compatibility)	f_{OD}	$0_{(1)}$/ 100KHz	400	kHz	$C_L \leq 250$ pF (21 cards)
Clock Low Time	t_{WL}	10		ns	$C_L \leq 100$ pF (7 cards)
Clock High Time	t_{WH}	10		ns	$C_L \leq 100$ pF (7 cards)
Clock Rise Time	t_{TLH}		10	ns	$C_L \leq 100$ pF (10 cards)
Clock Fall Time	t_{THL}		10	ns	$C_L \leq 100$ pF (7 cards)
Clock Low Time	t_{WL}	50		ns	$C_L \leq 250$ pF (21 cards)
Clock High Time	t_{WH}	50		ns	$C_L \leq 250$ pF (21 cards)
Clock Rise Time	t_{TLH}		50	ns	$C_L \leq 250$ pF (21 cards)
Clock Fall Time	t_{THL}		50	ns	$C_L \leq 250$ pF (21 cards)

Parameter	Symbol	Min.	Max.	Unit	Remark
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	5		ns	$C_L \leq 25$ pF (1 cards)
Input hold time	t_{IH}	5		ns	$C_L \leq 25$ pF (1 cards)
Outputs CMD, DAT (referenced to CLK)					
Output delay time during Data Transfer Mode	t_{ODLY}	0	14	ns	$C_L \leq 25$ pF (1 cards)
Output delay time during Identification Mode	t_{ODLY}	0	50	ns	$C_L \leq 25$ pF (1 cards)

NOTE: OHz stops the clock. The given minimum frequency range is for cases where a continuous clock is required.

3.5. SD Card Registers

There is a set of seven registers within the card interface. The OCR, CID, CSD and SCR registers carry the card configuration information. The RCA register holds the card relative communication address for the current session. The card status and SD status registers hold the communication protocol related status of the card.

3.5.1. Operating Conditions Register (OCR)

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. The SD Card is capable of executing the voltage recognition procedure (CMD1) with any standard SD Card host using operating voltages form 2 to 3.6 Volts.

Accessing the data in the memory array, however, requires 2.7 to 3.6 Volts. The OCR shows the voltage range in which the card data can be accessed. The structure of the OCR register is described in Table 3-8.

Table 3-8. OCR Register Definition

OCR Bit	VDD Voltage Window
0-3	Reserved
4	1.6-1.7
5	1.7-1.8
6	1.8-1.9
7	1.9-2.0
8	2.0-2.1
9	2.1-2.2
10	2.2-2.3
11	2.3-2.4
12	2.4-2.5
13	2.5-2.6
14	2.6-2.7
15	2.7-2.8
16	2.8-2.9

OCR Bit	VDD Voltage Window
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
24-30	reserved
31	Card power up status bit (busy)

The level coding of the OCR register is as follows:

- Restricted voltage windows=LOW
- Card busy=LOW (bit 31)

The least significant 31 bits are constant and will be set as described in Figure 4-8. If bit 32 (the busy bit) is set, it informs the host that the card power up procedure is finished.

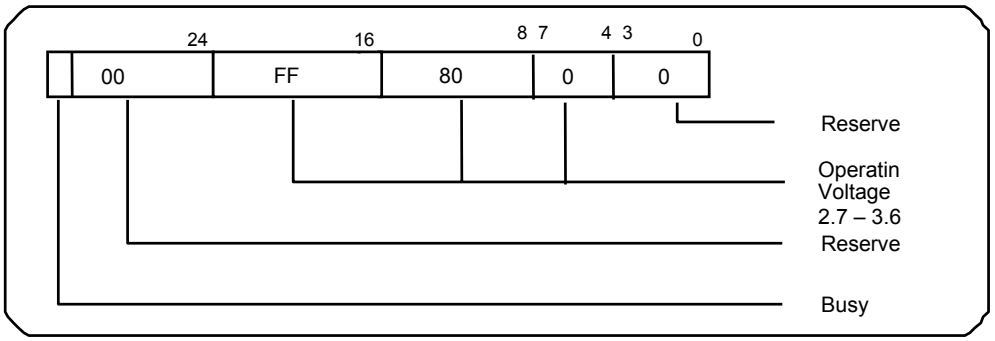


Figure 3-8. OCR Structure

3.5.2. Card Identification (CID) Register

The CID register is 16 bytes long and contains a unique card identification number as shown in Table 3-9. It is programmed during card manufacturing and cannot be changed by SD Card hosts. Note that the CID register in the SD Card has a different structure than the CID register in the MultiMediaCard.

Table 3-9. CID Fields

Name	Type	Width	CID—Slice	Comments	CID Value
Manufacturer ID (MID)	Binary	8	[127:120]	The manufacturer IDs are controlled and assigned by the SD Card Association.	0x03
OEM/Application ID (OID)	ASCII	16	[119:104]	Identifies the card OEM and/or the card contents. The OID is assigned by the 3C.*	SD ASCII Code 0x53, 0x44
Product Name (PNM)	ASCII	40	[103:64]	5 ASCII characters long	SD128, SD064, SD032, SD016, SD008
Product Revision** (PRV)	BCD	8	[63:56]	Two binary coded decimal digits	Product Revision (30)
Serial Number (PSN)	Binary	32	[55:24]	32 Bits unsigned integer	Product Serial Number
Reserved		4	[23:20]		
Manufacture Date Code (MDT)	BCD	12	[19:8]	Manufacture date–yym (offset from 2000)	Manufacture date(for example: Apr 2001 = 0x014)
CRC7 checksum*** (CRC)	Binary	7	[7:1]	Calculated	CRC7
Not used, always '1'		1	[0:0]		

* 3C = The 3 SDA founding companies: Toshiba, SanDisk, and MEI.

** 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 the "m" is the least significant nibble. Example: The PRV binary value filed for product revision "6.2" will be: 0110 0010.

*** The CRC Checksum is computed by the following formula:
CRC Calculation: G(x)=x7+3+1
M(x)=(MID-MSB)*x119+...+(CIN-LSB)*x0
CRC[6...0]=Remainder[(M(x)*x7)/G(x)]

3.5.3. CSD Register

The Card Specific Data (CSD) register contains configuration information required to access the card data. In Table 3-10, the cell type column defines the CSD field as Read only (R), One Time Programmable (R/W) or erasable (R/W/E). This table shows the value in "real world" units for each field and coded according to the CSD structure. The Model dependent column marks (with a check mark, √) the CSD fields that are model dependent. Note that the CSD register in the SD Card has a different structure than the CSD in the MultiMediaCard.

Table 3-10. CSD Register

Name	Field	Width	Cell Type	CSD-Slice	CSD Value	CSD Code
CSD structure	CSD_STRUCTURE	2	R	[127:126]	1.0	00b
Reserved	-	6	R	[125:120]	-	000000b
data read access-time-1	TAAC Binary MLC	 8 8	 R R	 [119:112] [119:112]	 1.5msec 10msec	 00100110b 00001111b
data read access-time-2 in CLK cycles (NSAC*100)	NSAC	8	R	[111:104]	0	00000000b
max. data transfer rate	TRAN_SPEED	8	R	[103:96]	25MHz	00110010b
card command classes	CCC	12	R	[95:84]	All (incl. WP, Lock/unlock)	1F5h

Name	Field	Width	Cell Type	CSD-Slice	CSD Value	CSD Code
max. read data block length	READ_BL_LEN	4	R	[83:80]	512byte	1001b
partial blocks for read allowed	READ_BL_PARTIAL	1	R	[79:79]	Yes	1b
write block misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]	No	0b
read block misalignment	READ_BLK_MISALIGN	1	R	[77:77]	No	0b
DSR implemented	DSR_IMP	1	R	[76:76]	No	0b
Reserved	-	2	R	[75:74]	-	00b
device size	C_SIZE	12	R	[73:62]	SD128=3843 SD064=3807 SD032=1867 SD016=899 SD008=831	F03h EDFh 74Bh 383h 33Fh
max. read current @VDD min	VDD_R_CURR_MIN	3	R	[61:59]	100mA	111b
max. read current @VDD max	VDD_R_CURR_MAX	3	R	[58:56]	80mA	110b
max. write current @VDD min	VDD_W_CURR_MIN	3	R	[55:53]	100mA	111b
max. write current @VDD max	VDD_W_CURR_MAX	3	R	[52:50]	80mA	110b
device size multiplier	C_SIZE_MULT	3	R	[49:47]	SD128=64 SD064=32 SD032=32 SD016=32 SD008=16	100b 011b 011b 011b 010b
erase single block enable	ERASE_BLK_EN	1	R	[46:46]	Yes	1b
erase sector size	SECTOR_SIZE	7	R	[45:39]	32blocks	0011111b
write protect group size	WP_GRP_SIZE	7	R	[38:32]	128sectors	1111111b
write protect group enable	WP_GRP_ENABLE	1	R	[31:31]	Yes	1b
Reserved for MultiMediaCard compatibility		2	R	[30:29]	-	00b
write speed factor Binary MLC	 R2W_FACTOR R2W_FACTOR	 3 3	 R R	 [1:16] [1:4]	 X16 X4	 100b 010b
max. write data block length	WRITE_BL_LEN	4	R	[25:22]	512Byte	1001b
partial blocks for write allowed	WRITE_BL_PARTIAL	1	R	[21:21]	No	0
Reserved	-	5	R	[20:16]	-	00000b
File format group	FILE_FORMAT_GRP	1	R/W(1)	[15:15]	0	0b
copy flag (OTP)	COPY	1	R/W(1)	[14:14]	Not Original	1b
permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]	Not Protected	0b
temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]	Not Protected	0b
File format	FILE_FORMAT	2	R/W(1)	[11:10]	HD w/partition	00b
Reserved		2	R/W	[9:8]	-	00b
CRC	CRC	7	R/W	[7:1]	-	CRC7
not used, always'1'	-	1	-	[0:0]	-	1b

NOTE: The device size indicates the user area size. It does not include the protected area that is used for security applications and is about 1 percent of the total card size.

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

CSD_STRUCTURE—describes the version of the CSD structure.

Table 3-11. CSD Register Structure

CSD_STRUCTURE	CSD Structure Version	Valid for SD Card Physical Specification Version
0	CSD version No. 1.0	Version 1.0-1.01
1-3	Reserved	

TAAC—Defines the asynchronous part (relative to the SD Card clock (CLK)) of the read access time.

Table 3-12. TAAC Access Time Definition

TAAC Bit Position	Code
2:0	time unit 0=1ns, 1=10ns, 2=100ns, 3=1µms, 4=10µms, 5=100µms, 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 dependent factor of the data access time. The unit for NSAC is 100 clock cycles. Therefore, the maximal value for the clock dependent part of the read access time is 25.5k clock cycles.

The total read access time N_{AC} as expressed in the Table 5-17 is the sum of TAAC and NSAC. It has to be computed by the host for the actual clock rate. The read access time should be interpreted as a typical delay for the first data bit of a data block from the end bit on the read commands.

TRAN_SPEED—Table 3-13 defines the maximum data transfer rate TRAN_SPEED.

Table 3-13. Maximum Data Transfer Rate Definition

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

CCC—The SD 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. Table 3-14 lists the supported card command classes; refer to Table 4-2 for command class definitions.

Table 3-14. Supported Card Command Classes

CCC Bit	Supported Card Command Class
0	class 0
1	class 1
.....	
11	class 11

READ_BL_LEN—The maximum read data block length is computed as $2^{READ\_BL\_LEN}$. The maximum block length might therefore be in the range 512...2048 bytes. Note that in the SD Card, the WRITE_BL_LEN is always equal to READ_BL_LEN.

Table 3-15. Data Block Length

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

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

READ_BL_PARTIAL=0 means that only the READ_BL_LEN block size can be used for block oriented data transfers.

READ_BL_PARTIAL=1 means that smaller blocks can be used as well. The minimum block size will be equal to minimum addressable unit (one byte)

WRITE_BLK_MISALIGN—Defines if the data block to be written by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in WRITE_BL_LEN.

WRITE_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.

WRITE_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

READ_BLK_MISALIGN—Defines if the data block to be read by one command can be spread over more than one physical block of the memory device. The size of the memory block is defined in READ_BL_LEN.

READ_BLK_MISALIGN=0 signals that crossing physical block boundaries is invalid.

READ_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

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

Table 3-16. DSR Implementation Code Table

DSR_IMP	DSR Type
0	no DSR implemented

1	DSR implemented

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

memory capacity = BLOCKNR * BLOCK_LEN

Where:

BLOCKNR = (C_SIZE+1) * MULT

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

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

Therefore, the maximum capacity which can be coded is 4096*512*2048 = 4 GBytes. Example: A four MByte card with BLOCK_LEN = 512 can be coded with C_SIZE_MULT = 0 and C_SIZE = 2047.

VDD_R_CURR_MIN, VDD_W_CURR_MIN—The maximum values for read and write currents at the minimal VDD power supply are coded in Table 3-17.

Table 3-17. V_{DD} Minimum Current Consumption

VDD_R_CURR_MIN VDD_W_CURR_MIN	Code For Current Consumption @ V_{DD}
2:0	0=0.5mA; 1=1mA; 2=5mA; 3=10mA; 4=25mA; 5=35mA; 6=60mA; 7=100mA

VDD_R_CURR_MAX, VDD_W_CURR_MAX—The maximum values for read and write currents at the maximum VDD power supply are coded Table 3-18.

Table 3-18. V_{DD} Maximum Current Consumption

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

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

Table 3-19. Multiply Factor For The Device Size

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—defines whether erase of one write block (see WRITE_BL_LEN) is allowed (other than SECTOR_SIZE given below). If ERASE_BLK_EN is 0, the host can erase a unit of SECTOR_SIZE. If ERASE_BLK_EN is 1, the host can erase either a unit of SECTOR_SIZE or a unit of WRITE_BLK_LEN.

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

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

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. Table 3-20 defines the field format.

Table 3-20. R2W_FACTOR

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 $2^{WRITE\_BL\_LEN}$. The maximum block length might therefore be in the range from 512 up to 2048 bytes. A Write Block Length of 512 bytes is always supported. Note that in the SD Card, the WRITE_BL_LEN is always equal to READ_BL_LEN.

Table 3-21. Data Block Length

WRITE_BL_LEN	Block Length
0-8	Reserved
9	29 = 512 Bytes
….	
11	211 = 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 blocks, 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_GROUP—Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in Table 3-22.

COPY—This bit marks the card as an original (‘0’) or non-original (‘1’). Once set to non-original, this bit cannot be reset to original. The definition of “original” and “non-original” is application dependent and changes no card characteristics.

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

TMP_WRITE_PROTECT—Temporarily protects the whole card content, except the secured protected area, 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.

Table 3-22. File Format

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. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents.

3.5.4. SCR Register

In addition to the CSD register, there is another configuration register that is 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 shall be set in the factory by the SD Card manufacturer. Table 3-23 describes the SCR register content.

Table 3-23. SCR Fields

Description	Field	Width	Cell Type	SCR Slice	SCR Value	SCR Code
SCR Structure	SCR_STRUCTURE	4	R	[63:60]	V1.0	0
SD Card—Spec. Version	SD_SPEC	4	R	[59:56]	V1.01	0
data_status_after erases	DATA_STAT_AFTER_ERASE	1	R	[55:55]	0	0
SD Security Support	SD_SECURITY	3	R	[54:52]	Prot 2, Spec V1.01	2
DAT Bus widths supported	SD_BUS_WIDTHS	4	R	[51:48]	1 & 4	5
Reserved	-	16	R	[47:32]	0	0
Reserved for manufacturer usage	-	32	R	[31:0]	0	0

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

Table 3-24. SCR Register Structure Version

CSD_STRUCTURE	CSD Structure Version	Valid for SD Physical Layer Specification Version
0	SCR version No. 1.0	Version 1.0-1.01
1-15	reserved	

SD_SPEC—Describes the SD Card Physical Layer Specification version supported by this card.

Table 3-25. SD Card Physical Layer Specification Version

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

DATA_STAT_AFTER_ERASE—Defines the data status after erase, whether it is '0' or '1' (the status is card vendor dependent).

SD_SECURITY—Describes the security algorithm supported by the card.

Table 3-26. SD Supported Security Algorithm

SD_SECURITY	Supported Algorithm
0	No security
1	Security protocol 1.0 Security Spec Ver 0.96
2	Security protocol 2.0 Security Spec Ver 1.0-1.01
3 .. 7	Reserved

NOTE: It is mandatory for a Writable SD Card (OTP or R/W) to support Security Protocol.

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

Table 3-27. SD Card Supported Bus Widths

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

Since SD Card shall support at least the two bus modes 1bit or 4bit width then any SD Card shall set at least bits 0 and 2 (SD_BUS_WIDTH=0101).

3.5.5. Status Register

The SD Card supports the following two card status fields:

- **Card Status**—This status field is compatible to the MultiMediaCard protocol.
- **SD_Status**—This extended status field of 512 bits supports special features unique to the SD Card and future application specific features.

The SD Card status registers' structures are defined in Table 3-28. The Type and Clear-Condition fields in the table are coded as follows:

Type:

- E—Error bit.
- S—Status bit.
- R—Detected and set for the actual command response.
- X—Detected and set during command execution. The host must poll the card by sending 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.

Table 3-28. Card Status

Bits	Identifier	Type	Value	Description	Clear Cond.
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	'0'= no error '1'= error	A misaligned address that did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	E R	'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
24	LOCK_UNLOCK_FAIL ED	E R X	'0' = no error '1' = error	Set when a sequence or password error has been detected in lock/ unlock card command or if there was an attempt to access a locked card	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

Bits	Identifier	Type	Value	Description	Clear Cond.
18	Reserved				
17	Reserved				
16	CID/ CSD_OVERWRITE	E R X	’0’= no error ’1’= error	Can be either one of the following errors: - The CID register has been already written and can not be overwritten - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
15	WP_ERASE_SKIP	S X	’0’= not protected ’1’= protected	Only partial address space was erased due to existing write protected blocks.	C
14	CARD_ECC_DISABLE D	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 5 = data 6 = rcv 7 = prg 8 = dis 9-15 = reserved	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15.	B
8	READY_FOR_DATA	S X	’0’= not ready ’1’= ready	Corresponds to buffer empty signalling on the bus.	A
7:6					
5	APP_CMD	S R	‘0’ = Disabled ‘1’ = Enabled	The card will expect ACMD, or indication that the command has been interpreted as ACMD.	C
4	Reserved				
3	AKE_SEQ_ERROR (SD Card Security spec.)	E R	‘0’ = no error ‘1’ = error	Error in the sequence of authentication process.	C
2	Reserved for application specific commands				
1, 0	Reserved for manufacturer test mode				

3.5.6. 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 512 bits. The content of this register is transmitted to the Host over the DAT bus along with 16 bits 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 selected). The SD Status structure is listed in Table 3-29. The same abbreviations for 'type' and 'clear condition' were used as for the Card Status above.

Table 3-29. SD Card Status

Bits	Identifier	Type	Value	Description	Clear Cond.
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 the 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 (refer to the SD Security Specifications document).	A
508: 496	Reserved				
495: 480	SD_CARD_TYPE	S R	'00xxh'=SD Memory Cards as defined in Physical Spec. Ver. 1.01 ('x'=don't care). The following cards are currently defined: '0000'=Regular SD RD/WR Card. '0001'=SD ROM Card	In the future, the 8 LSBs will be used to define different variations of an SD Card (each bit will define different SD types). The 8 MSBs will be used to define SD Cards that do not comply with the SD Memory Card as defined in the Specification Ver. 1.01	A
479: 448	SIZE_OF_PROTECTED_AREA	S R	Size of protected area (in units of MULT*BLOCK_LEN refer to CSD register.	Shows the size of the protected area. The actual area = (SIZE_OF_PROTECTED_AREA) * MULT * BLOCK_LEN.	A
447: 312	Reserved				
311: 0	Reserved for Manufacturer				

3.5.7. RCA Register

The 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.

3.5.8. SD Card Registers in SPI Mode

In SPI mode, all the card's registers are accessible. Their format is identical to the format in the SD Card mode. However, a few fields are irrelevant in SPI mode. In SPI mode, the card status register has a different, shorter, format as well. Refer to the SPI Protocol section for more details.

3.6. Data Interchange Format and Card Sizes

In general, SD Card data is structured by means of a file system. The SD Card File System Specification, published by the SD Association, describes the file format system that is implemented in the SanDisk SD Card. In general, each SD Card is divided into two separate DOS-formatted partitions as follows:

- **The User Area**—used for secured and non-secured data storage and can be accessed by the user with regular read/write commands.
- **Security Protected Area**—used by copyright protection applications to save security related data and can be accessed by the host using the secured read/write command after doing authentication as defined in the SD Security Specification. The security protected area size is defined by SanDisk as approximately one percent of the total size of the card. Tables 3-30 and 3-31 describe the user and protected areas for all SanDisk SD Cards.

Table 3-30. Parameters for User Area DOS Image

Capacity	Total LBAs	No. of Partition Sys. Area Sectors	Total Partition Sectors	User Data Sectors	User Data Bytes
16 MB	28,800	39	28,743	28,704	14,696,448
32 MB	59,776	45	59,725	59,680	30,556,160
64 MB	121,856	57	121,817	121,760	62,341,120
128 MB	246,016	95	245,919	245,824	125,861,888
256 MB	494,080	155	493,979	493,824	252,837,888
512 MB	990,864	275	990,627	990,352	507,060,224
1024 MB	1,983,744	519	1,983,495	1,982,976	1,015,283,712

Table 3-31. Parameters for Protected Area DOS Image

Capacity	Total LBAs	No. of Partition Sys. Area Sectors	Total Partition Sectors	User Data Sectors	User Data Bytes
16 MB	352	35	351	316	161,792
32 MB	736	37	733	696	356,352
64 MB	1,376	37	1,373	1,336	684,032
128 MB	2,624	35	2,611	2,576	1,318,912
256 MB	5376	37	5365	5328	2,727,936
512 MB	10,240	37	10,213	10,176	5,210,112
1024 MB	20,480	37	20,421	20,384	10,436,608

This page intentionally left blank.

4. Secure Digital (SD) Card Protocol Description

4.1. 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 that 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 that 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.

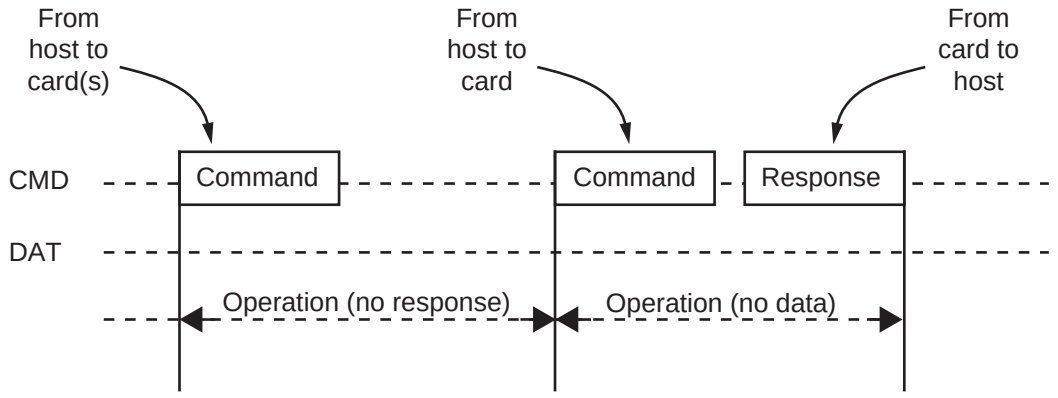


Figure 4-1. "No Response" and "No Data" Operations

Card addressing is implemented using a session address that is assigned to the card during the initialization phase. The basic transaction on the SD bus is the command/response transaction (see Figure 4-1). This type of bus transaction transfers their information directly within the command or response structure. In addition, some operations have a data token.

Data transfers to/from the SD Card are done in blocks. Data blocks are always followed 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 (as long as the card supports this feature).

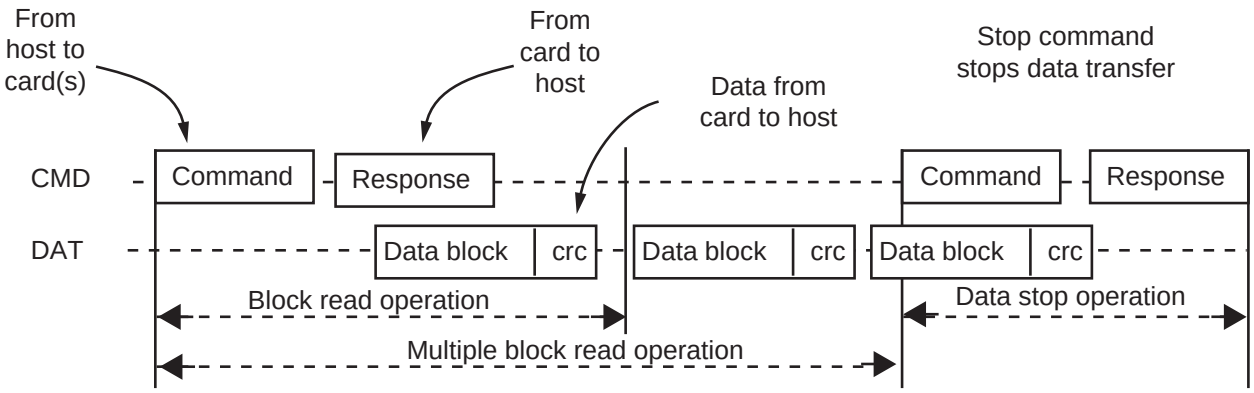


Figure 4-2. Multiple Block Read Operation

The block write operation uses a simple busy signaling of the write operation duration on the DAT0 data line (see Figure 4-3) regardless of the number of data lines used for transferring the data.

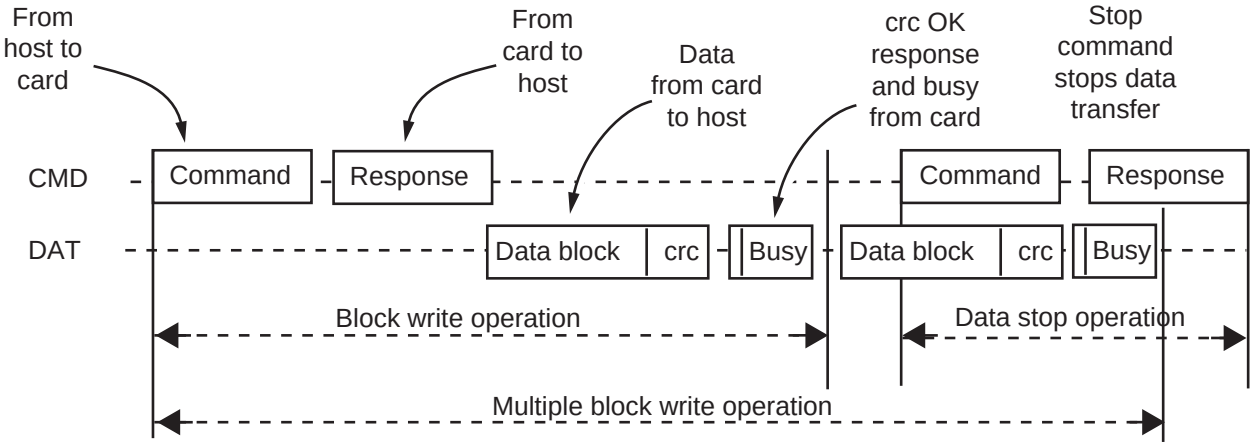


Figure 4-3. Multiple Block Write Operation

Command tokens have the coding scheme shown in Figure 4-4.

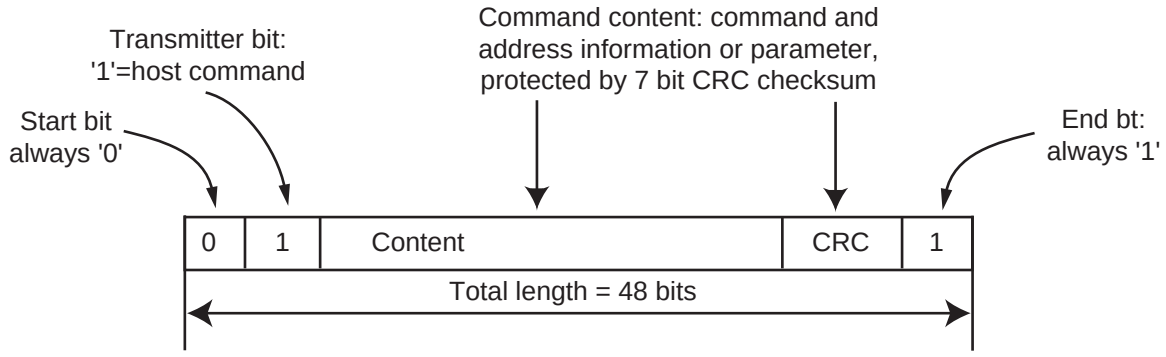


Figure 4-4. Command Token Format

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

Response tokens have four coding schemes depending on their content. The token length is either 48 or 136 bits. The CRC protection algorithm for block data is a 16-bit CCITT polynomial. All used CRC types are described in Section 4.6.

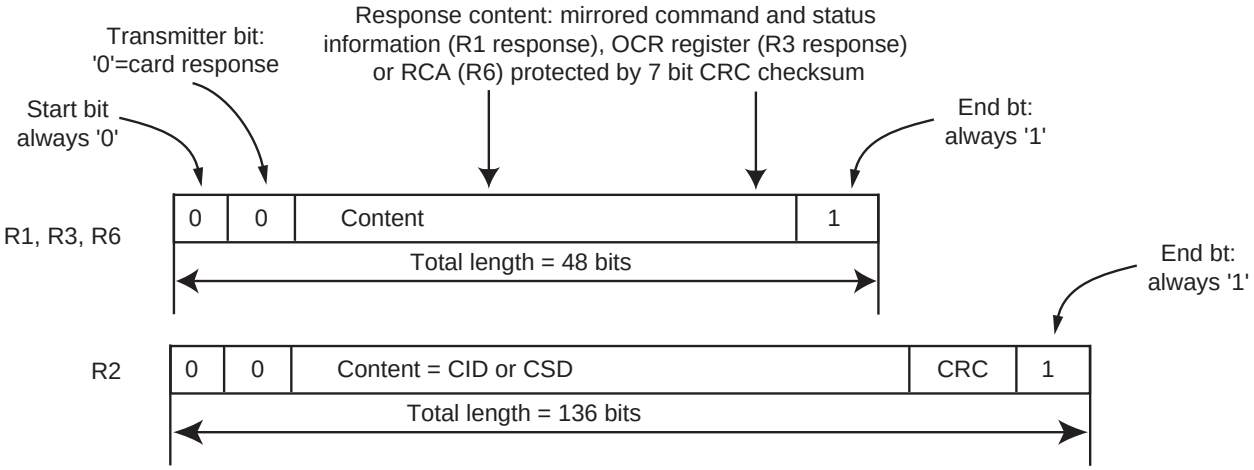


Figure 4-5. Response Token Format

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

When the wide bus option is used, the data is transferred 4 bits at a time (see Figure 4-6). 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”).

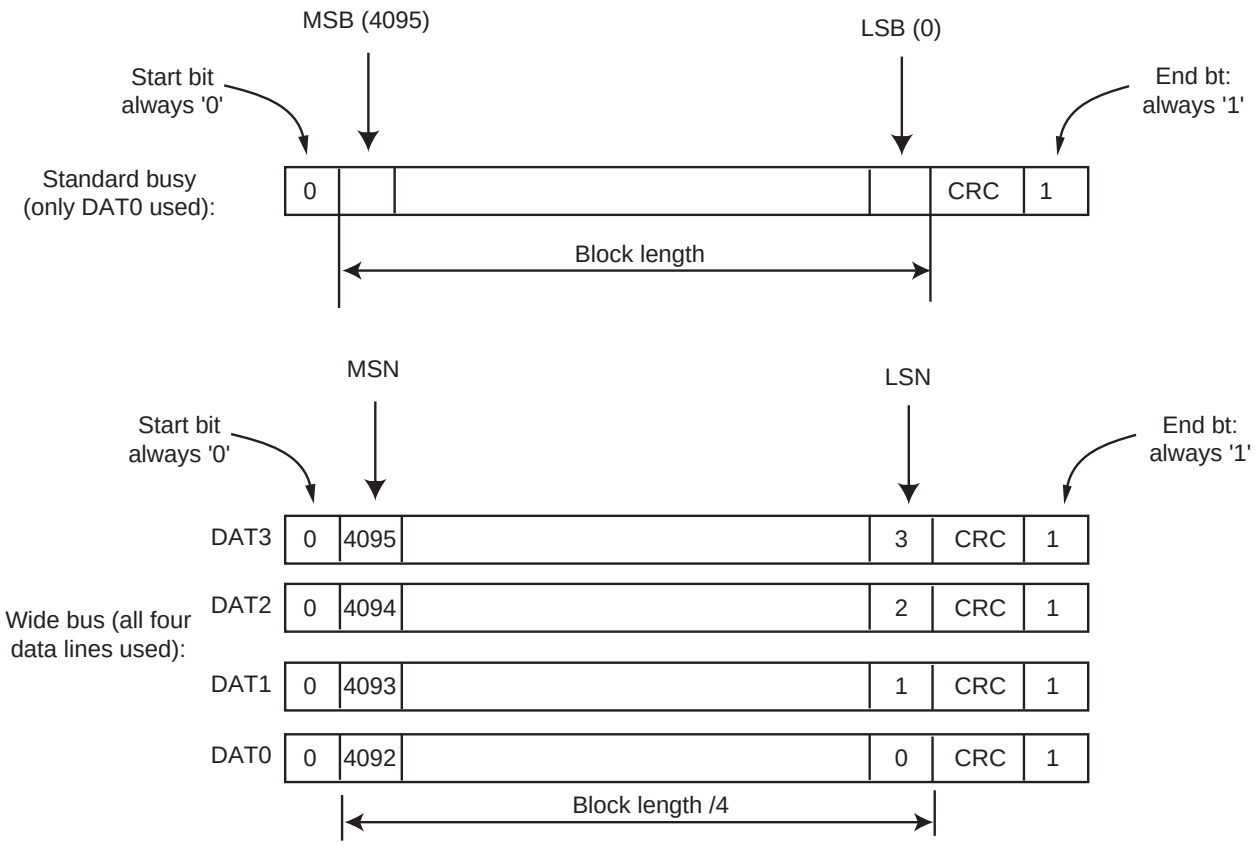


Figure 4-6. Data Packet Format

4.2. Protocol's Functional Description

All communication between the host and SD Cards is controlled by the host (master). The host sends the following two types of commands:

- **Broadcast Commands**—Broadcast commands are intended for all SD Cards. Some of these commands require a response.
- **Addressed (Point-to-Point) Commands**—The addressed commands are sent to the addressed SD Card and cause a response to be sent from this card.

A general overview of the command flow is shown in Figure 5-7 for the Card Identification Mode and in Figure 5-8 for the Data Transfer Mode. The commands are listed in the command tables (Tables 4-3 through 4-10). The dependencies between the current SD Card state, received command and following state are listed in Table 4-11. In the following sections, the different card operation modes will be described first. Thereafter, the restrictions for controlling the clock signal are defined. All SD Card commands together with the corresponding responses, state transitions, error conditions and timings are presented in the following sections.

Two operation modes are defined for SD Cards:

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

Table 4-1 lists the dependencies between operation modes and card states. Each state in the SD Card state diagram (Figures 4-7 and 4-8) is associated with one operation mode.

Table 4-1. Overview of Card States versus Operation Modes

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	

4.3. 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 the data communication in the Card Identification Mode uses only the command line (CMD).

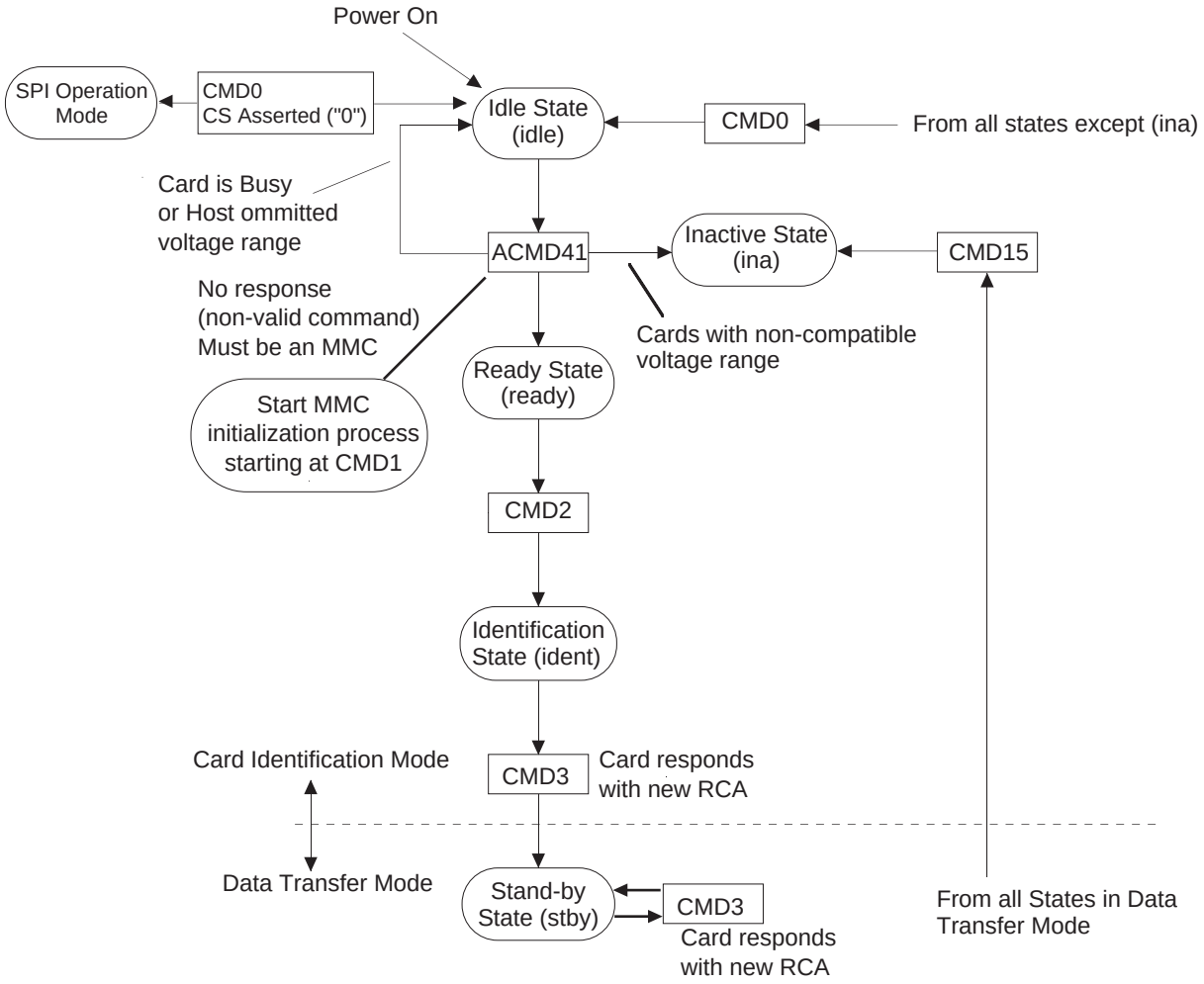


Figure 4-7. SD Card State Diagram (Card Identification Mode)

4.3.1. Reset

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

After power-on by the host, all SD Cards are in Idle State, including the cards that were in Inactive State. Note that at least 74 clock cycles are required prior to starting bus communication.

After power-on or CMD0, all SD Cards' CMD lines are in input mode, waiting for the start bit of the next command. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

4.3.2. Operating Voltage Range Validation

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

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

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

The MultiMediaCard will not respond to ACMD41 (actually it will not respond to APP_CMD—CMD55, that precedes it). The MultiMediaCard will be initialized as per the MultiMediaCard spec, using SEND_OP_COND command (CMD1 of MultiMediaCard). The host should ignore an ILLEGAL_COMMAND status in the MultiMediaCard response to CMD3, since it is a residue of ACMD41 which is invalid in the MultiMediaCard (CMD0, 1, 2 do not clear the status register). Actually, ACMD41 and CMD1 will be used by the host to distinguish between MultiMediaCard and SD Cards in a system.

By omitting the voltage range in the command, the host can query each card and determine if there are any non compatibilities before sending out-of-range cards into the Inactive State. This query should be used if the host can select a common voltage range or wants to notify the application of non-usable cards in the stack.

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

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

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

4.3.3. Card Identification Process

The host starts the card identification process with the identification clock rate f_{OD} (see Section 3.4.4). In SD Card the CMD line output drives are push-pull drivers.

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

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

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

4.4. Data Transfer Mode

Until the content of all CSD registers is known by the host, the f_{PP} clock rate must remain at f_{OD} because some cards may have operating frequency restrictions. The host issues SEND_CSD (CMD9) to obtain the Card Specific Data (CSD register), e.g., block length, card storage capacity, maximum clock rate. Figure 4-8 shows a block diagram of the Data Transfer Mode.

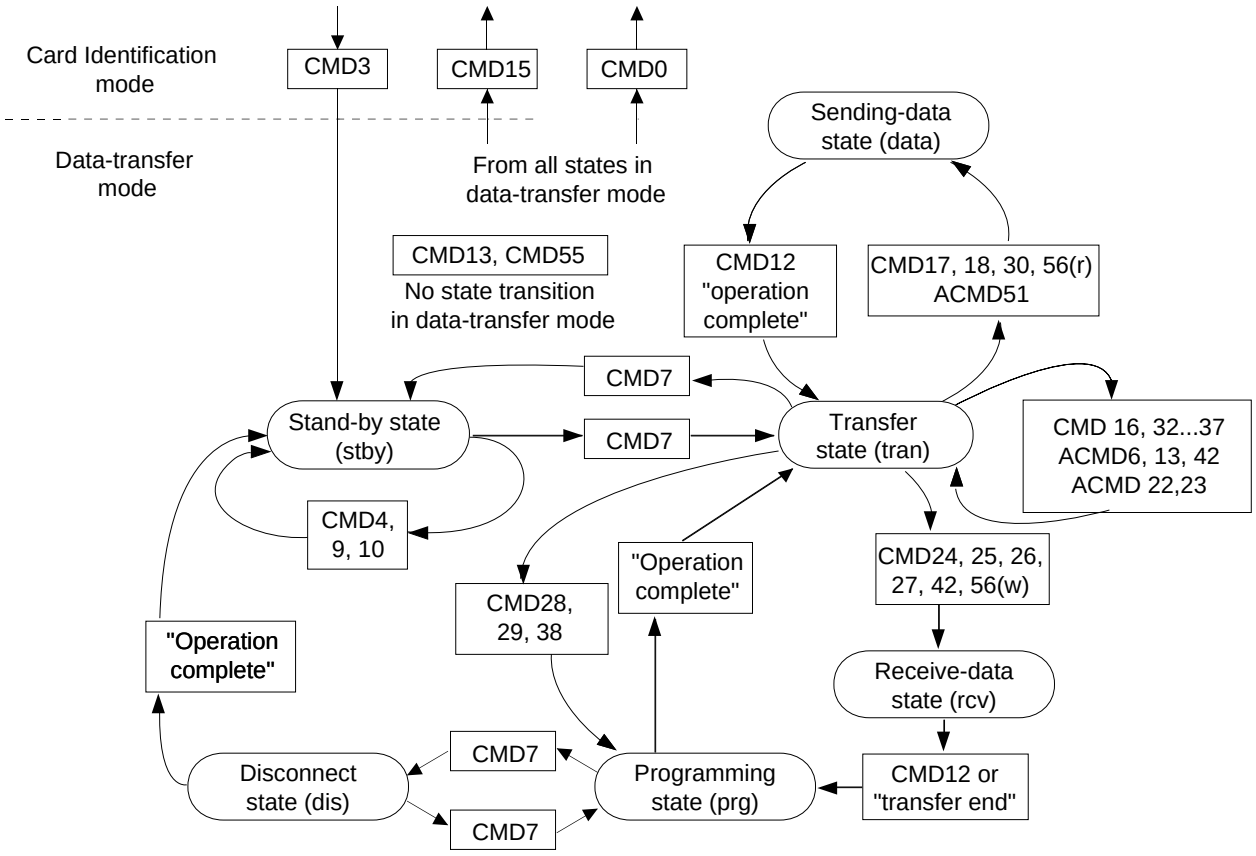


Figure 4-8. SD Card State Diagram (Data Transfer Mode)

CMD7 is used to select one SD Card and place it in the Transfer State. Only one SD Card can be in the Transfer State at a given time. If a previously selected SD Card is in the Transfer State, its connection with the host is released and it will move back to the Stand-by State. When CMD7 is issued with the reserved relative card address "0x0000," all cards transfer back to Stand-by State. (Note that it is the responsibility of the Host to reserve the RCA=0 for card de-selection—refer to Table 4-3, CMD7 description). This may be used before identifying new cards without resetting other already registered cards. Cards that already have an RCA do not respond to identification commands (ACMD41, CMD2, CMD3) in this state.

Important Note: The card de-selection is done if a certain card gets CMD7 with un-matched RCA. That happens automatically if selection is done to another card and the *CMD lines are common*. So, in the SD Card system, it will be the responsibility of the host either:

- To work with the common CMD line (after initialization is done). In this case the card de-selection will be done automatically (as in MultiMediaCard system).
- If the CMD lines are separate, to be aware of the necessity to deselect cards.

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

The relationship between the various data transfer modes is summarized in Figure 4-8, and in the following paragraphs:

- All data read commands may be aborted any time by the stop command (CMD12). The data transfer will terminate and the card will return to the *Transfer State.* The read commands are: block read (CMD17), multiple block read (CMD18), send write protect (CMD30), send scr (ACMD51) and general command in read mode (CMD56).
- All data write commands can be aborted any time by the stop command (CMD12). The write commands must be stopped prior to deselecting the card by CMD7. The write commands are: block write (CMD24 and CMD25), write CID (CMD26), write CSD (CMD27), lock/unlock command (CMD42) and general command in write mode (CMD56).
- As soon as the data transfer is completed, the card will exit the data write state and move either to the *Programming State* (transfer is successful) or *Transfer State* (transfer failed).
- If a block write operation is stopped and the block length and CRC of the last block are valid, the data will be programmed.
- The card may provide buffering for block write. This means that the next block can be sent to the card while the previous is being programmed. If all write buffers are full, and as long as the card is in *Programming State* (see SD Card state diagram Figure 5-8), the DAT0 line will be kept low (BUSY).
- There is no buffering option for write CSD, write CID, write protection and erase. This means that while the card is busy servicing any one of these commands, no other data transfer commands will be accepted. DAT0 line will be kept low as long as the card is busy and in the *Programming State*. Actually if the CMD and DAT0 lines of the cards are kept separated and the host keeps the busy DAT0 line disconnected from the other DAT0 lines (of the other cards), the host may access the other cards while the card is in busy.
- Parameter set commands are *not* allowed while the card is programming. Parameter set commands are: set block length (CMD16), erase block start (CMD32) and erase block end (CMD33).
- Read commands are *not* allowed while the card is programming.
- Moving another card from *Stand-by* to *Transfer State* (using CMD7) will not terminate erase and programming operations. The card will switch to the *Disconnect State* and will release the DAT line.
- A card can be reselected while in the *Disconnect State*, using CMD7. In this case the card will move to the *Programming State* and reactivate the busy indication.
- Resetting a card (using CMD0 or CMD15) will terminate any pending or active programming operation. This may destroy the data contents on the card. It is the host's responsibility to prevent this.

4.4.1. Wide Bus Selection/Deselection

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

4.4.2. Data Read Format

The DAT bus line is high when no data is transmitted. A transmitted data block consists of a start bit (LOW), followed by a continuous data stream. The data stream contains the net payload data (and error correction bits if an off-card ECC is used). The data stream ends with an end bit (HIGH). The data transmission is synchronous to the clock signal.

The payload for block-oriented data transfer is preserved by a CRC check sum. The generator polynomial is a standard CCITT polynomial:

$$x^{16}+x^{12}+x^{5}+1$$

The code is a shortened BCH code with d=4 and is used for payload length of up to 2048 Bytes. Note that the CRC check sum is calculated and attached to each DAT line at the end of the block. In the case of a wide bus operation (DAT0-DAT3), the 16-bit CRC is calculated separately for each DAT line.

Block Read

A block read is a block-oriented data transfer. The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ_BL_LEN). Smaller blocks whose starting and ending address are wholly contained within one physical block (as defined by READ_BL_LEN) may also be transmitted. A CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ_SINGLE_BLOCK) starts a block read, and after a complete transfer the card goes back to Transfer State. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued. The stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the stop command.

If the host uses partial blocks whose accumulated length is not block aligned, the card will, at the beginning of the first misaligned block, detect a block misalignment error, set the ADDRESS_ERROR error bit in the status register, abort transmission and wait (in the *Data State*) for a stop command.

4.4.3. Data Write Format

The data transfer format is similar to the data read format. For block-oriented write data transfer, the CRC check bits are added to each data block. The card performs a CRC check for each data line at the end of each received data block prior to a write operation. (The polynomial is the same one used for a read operation.) With this mechanism, writing of erroneously transferred data can be prevented.

Block Write

During block write (CMD24—27,42,56(w)), one or more blocks of data are transferred from the host to the card, with CRC appended to the end of each block by the host. The SanDisk SD Card is able to accept a block of data defined by WRITE_BL_LEN of 512 bytes. If the CRC fails, the card shall indicate the failure on the DAT line (see below); the transferred data will be discarded and not written, and all further transmitted blocks (in multiple block write mode) will be ignored.

Multiple block write command shall be used rather than continuous single write commands to make faster write operation. Partial block writes (blocks smaller than 512 bytes) are not allowed in the SanDisk SD Card.

The write operation will be aborted if the host tries to write over a write-protected area. In this case, the card sets the WP_VIOLATION bit in the status register, and while ignoring all further data transfer, waits in the *Receive-data-State* for a stop command.

Programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.

After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing CMD7 (to select a different card), which will displace the card into the *Disconnect State* and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable. Actually, the host may perform simultaneous write operations to several cards by using an inter- leaving process. The interleaving process can be done by accessing each card separately while other cards are in busy. This process can be done by proper CMD and DAT0-3 line manipulations (disconnection of busy cards).

Pre-erase setting prior to a multiple block write operation

Setting a number of write blocks to be pre_erased (ACMD23) will make a following Multiple Block Write operation faster compared to the same operation without preceding ACMD23. The host will use this command to define how many write blocks are going to be sent in the next write operation. If the host terminates the write operation (using stop transmission) before all the data blocks are sent to the card, the content of the remaining write blocks is undefined (can be either erased or still have the old data). If the host sends a greater number of write blocks than are defined in ACMD23, the card will erase blocks one by one (as new data is received). This number will be reset to the default (=1) value after Multiple Blocks Write operation.

It is recommended to use this command preceding CMD25, so that SanDisk's SD Card will be faster for Multiple Write Blocks operation. Note that the host must send ACMD23 just before the WRITE command if the host wants to use the pre-erase feature. If not, pre-erase-count might be cleared automatically when another command (ex: Security Application Commands) is executed.

Send Number of Written Blocks

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

Erase

It is desirable to erase many write blocks simultaneously in order to enhance the data throughput. Identification of these write blocks is accomplished with the ERASE_WR_BLK_START (CMD32), ERASE_WR_BLK_END (CMD33) commands.

The host must adhere to the following command sequence: ERASE_WR_BLK_START, ERASE_WR_BLK_END and ERASE (CMD38).

If an erase (CMD38) or address setting (CMD32, 33) command is received out of sequence, the card shall set the ERASE_SEQ_ERROR bit in the status register and reset the whole sequence.

If an out of sequence command (except SEND_STATUS) is received, the card shall set the ERASE_RESET status bit in the status register, reset the erase sequence and execute the last command.

If the erase range includes write protected sectors, they shall be left intact and only the non-protected sectors shall be erased. The WP_ERASE_SKIP status bit in the status register shall be set.

The address field in the address setting commands is a write block address in byte units. The card will ignore all LSBs below the WRITE_BLK_LEN (see CSD) size.

As described above for block write, the card will indicate that an erase is in progress by holding DAT0 low. The actual erase time may be quite long, and the host may issue CMD7 to deselect the card or perform card disconnection, as described in the Block Write section, above.

The data at the card after an erase operation is either '0' or '1', depending on the card vendor. The SCR register bit DATA_STAT_AFTER_ERASE (bit 55) defines whether it is '0' or '1'.

4.4.4. Write Protect Management

Three write protect methods are supported in the SD Card as follows:

- Mechanical write protect switch (Host responsibility only)
- Card internal write protect (Card's responsibility)
- Password protection card lock operation.

4.4.4.1. Mechanical Write Protect Switch

A mechanical sliding tablet on the side of the card (refer to the mechanical description) will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is close the card is not write protected.

A proper, matched, switch on the socket side will indicate to the host that the card is write-protected or not. It is the responsibility of the host to protect the card. The position of the write protect switch is *un-known to the internal circuitry of the card*.

4.4.4.2. Card's Internal Write Protection (Optional)

Card data may be protected against either erase or write. The entire card may be permanently write-protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD.

4.4.5. Application Specific Commands

The SD Card is defined to be protocol forward compatible to the MultiMediaCard Standard.

The SD Card system is designed to provide a standard interface for a variety application types. In order to keep future compatibility to the MultiMediaCard standard together with new SD Card specific commands, the SD Card uses the Application Specific commands feature to implement its proprietary commands. Following is a description of APP_CMD and GEN_CMD as they were defined in the MultiMediaCard Specification.

Application Specific Command—APP_CMD (CMD55)

This command, when received by the card, will cause the card to interpret the following command as an application specific command (ACMD). The ACMD has the same structure as regular MultiMediaCard standard commands and it may have the same CMD number. The card will recognize it as ACMD by the fact that it appears after APP_CMD.

The only effect of the APP_CMD is that if the command index of the immediately following command has an ACMD overloading it, the non standard version will be used. For example, a card has a definition for ACMD13 but not for ACMD7. Therefore, if Command 13 is received immediately after APP_CMD command, it would be interpreted as the non standard ACMD13, whereas command 7, similarly received, would be interpreted as the standard CMD7. In order to use one of the manufacturer specific ACMDs the host does one of the following:

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

If a non-valid command is sent (neither ACMD nor CMD) then it will be handled as a standard SD Card illegal command error.

From the SD Card protocol point of view the ACMD numbers will be defined by the manufacturers with some restrictions. The following ACMD numbers are reserved for the SD Card proprietary applications and may not be used by any SD Card manufacturer:

ACMD6, ACMD13, ACMD17-25, ACMD38-49, ACMD51.

General Command—GEN_CMD (CMD56)

The bus transaction of the GEN_CMD is the same as the single block read or write commands (CMD24 or CMD17). The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not memory payload data but has a vendor specific format and meaning. The card shall be selected (‘*tran_state’*) before sending CMD56. The data block size is the BLOCK_LEN that was defined with CMD16. The response to CMD56 will be R1.

Currently, there are no defined commands or usage for CMD56 in SanDisk’s SD Card, but new commands may be easily defined and tailored for OEM application specific requirements (upon request to SanDisk).

4.5. Clock Control

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

There are a few restrictions the SD Card host must follow:

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

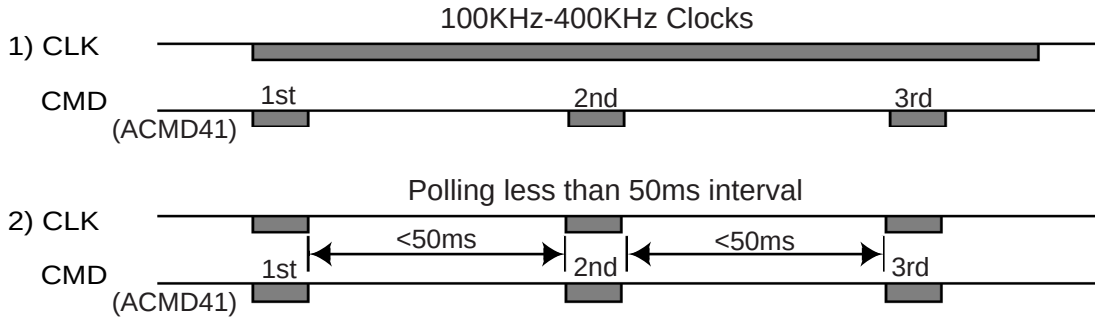


Figure 4-9. Host Procedures Waiting for Card to be Ready

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

4.6. Cyclic Redundancy Codes (CRC)

The Cyclic Redundancy Check (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, CRC is generated for each DAT line per transferred block. 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 significant bit of the corresponding bit string (of the command, response, CID or CSD). The degree *n* of the polynomial is the number of CRC protected bits decreased by one. The number of bits to be protected is 40 for commands and responses (n = 39), and 120 for the CSD and CID (n = 119).

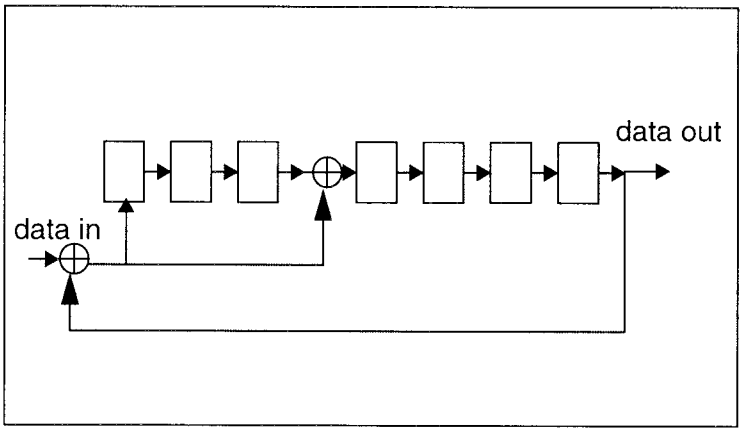


Figure 4-10. CRC7 Generator/Checker

CRC16

When one DAT line is used (as in the MultiMediaCard), 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) = x16 + x12 +x5 +1

M(x) = (first bit) * xn + (second bit)* xn-1 +...+ (last bit) * x0

CRC[15...0] = Remainder [(M(x) * x16) / 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. For example, n = 4,095 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 2,048 bytes ($n \leq 16,383$). 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.

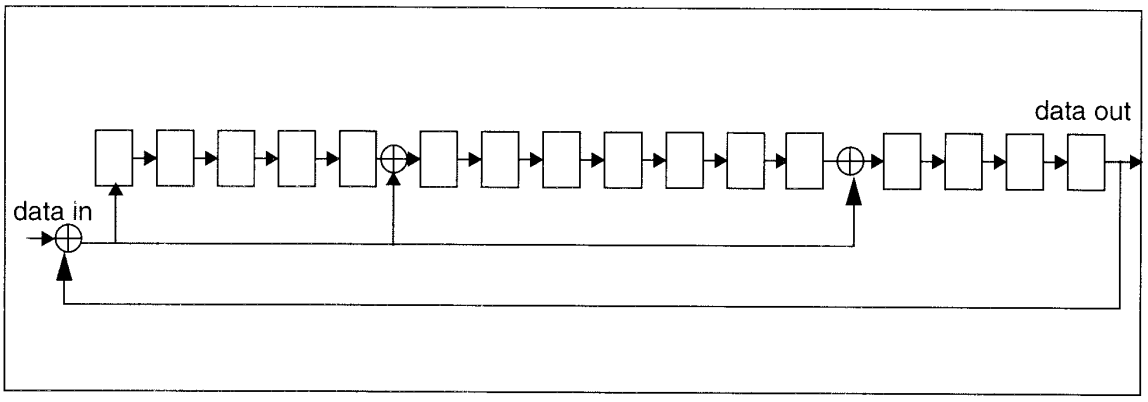


Figure 4-11. CRC16 Generator/Checker

4.7. Error Conditions

The following sections provide valuable information on error conditions.

4.7.1. CRC and Illegal Command

All commands are protected by CRC bits. If the addressed SD Card's CRC check fails, the card does not respond and the command is not executed. The SD 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, an SD Card shall not change its state, shall not respond and shall set the ILLEGAL_COMMAND error bit in the status register. Only the non-erroneous state branches are shown in the state diagrams (Figure 5-7 and Figure 5-8). Table 5-11 contains a complete state transition description.

There are different kinds of illegal commands:

- Commands belonging to classes not supported by the SD Card (e.g., write commands in read-only cards).
- Commands not allowed in the current state (e.g., CMD9 in Transfer State).
- Commands not defined (e.g., CMD5).

4.7.2. Read, Write and Erase Time-out Conditions

The times after which a time-out condition for Read operations occur are (card independent) **either 100 times longer** than the typical access times for these operations given below **or 100ms**. The times after which a time-out condition for Write/Erase operations occur are (card independent) **either 100 times longer** than the typical program times for these operations given below **or 250ms**. A card shall complete the command within this time period, or give up and return an error message. If the host does not get any response 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). The typical access and program times are defined as follows:

Read

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

Write

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

Erase

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

4.8. Commands

The following sections provide valuable information on commands.

4.8.1. Command Types

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

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

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

4.8.2. Command Format

(Command length 48 bits, 1.92 µs @ 25 MHz)

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC7[1]	end bit

Commands and arguments are listed in Table 5-3 through Table 5-10.

7-bit CRC Calculation: $G(x) = x^{7} + x^{3} + 1$

$M(x) = (\text{start bit}) * x^{39} + (\text{host bit}) * x^{38} + ... + (\text{last bit before CRC}) * x^{0}$

$CRC[6...0] = \text{Remainder}[(M(x) * x^{7})/G(x)]$

4.8.3. Command Classes

The command set of the SD Card is divided into several classes (refer to Figure 4-2). Each class supports a set of SD Card functions.

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.

1) 7-bit Cyclic Redundancy Check.

Table 4-2. Card Command Classes (CCCs)

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

4.8.4. Detailed Command Description

Tables 4-3 through 4-9 define in detail the SD Card bus commands.

Table 4-3. Basic Commands (Class 0 And Class 1)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] don't care*	-	GO_IDLE_STATE	Resets all cards to Idle State.
CMD1	Reserved				
CMD2	bcr	[31:0] don't care*	R2	ALL_SEND_CID	Asks any card to send their CID numbers on the CMD line. (Any card that is connected to the host will respond.)
CMD3	bcr	[31:0] don't care*	R6	SEND_RELATIVE_ ADDR	Asks the card to publish a new relative address (RCA).
CMD4[1]	Not Supported				
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16] RCA [15:0] don't care*	R1 (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 state. In both cases the card is selected by its own relative address and deselected by any other address; address 0 deselects all. When the RCA equals 0, the host may do one of the following: —use other RCA number to perform card de-selection or —re-send CMD3 to change its RCA number to other then 0 and then use CMD7 with RCA=0 for card de-selection.
CMD8	Reserved				
CMD9	ac	[31:16] RCA [15:0] don't care*	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] don't care*	R2	SEND_CID	Addressed card sends its card identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address[2]	R1	READ_DAT_UNTIL_ STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] don't care*	R1b[3]	STOP_ TRANSMISSION	Terminates a multiple block read/write operation.
CMD13	ac	[31:16] RCA [15:0] don't care*	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] don't care*	-	GO_INACTIVE_ STATE	Sets the card to inactive state.

* The bit places must be filled but the value is irrelevant.

1) The DSR option (as well as the SET_DSR command) is not supported by the SanDisk SD Card.
2) The addressing capability @ 8 bit address resolution is 232 = 4 Gbyte.
3) The card may become busy after this command. Refer to Figure 5-25 for more details.

Table 4-4. Block Read Commands (Class 2)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read and write).[1]
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[2]
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously send blocks of data until interrupted by a stop transmission command.
CMD19 – CMD23	Reserved				

Table 4-5. Block Write Commands (Class 4)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[3]
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_ BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	Not Applicable				
CMD27	adtc	[31:0] don't care*	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

* The bit places must be filled but the value is irrelevant.

Table 4-6. Write Protection (Class 6)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD28*	ac	[31:0] data address	R1b	SET_WRITE_PROT	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	This command clears the write protection bit of the addressed group.
CMD30*	adtc	[31:0] write protect data address	R1	SEND_WRITE_ PROT	This command asks the card to send the status of the write protection bits.
CMD31	Reserved				

1) The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.
2) The data transferred must not cross a physical block boundary.
3) All data blocks are responded to with a data response token followed by a busy signal. The data transferred must not cross a physical block boundary.

Table 4-7. Erase Commands (Class 5)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD32	ac	[31:0] data address	R1	ERASE_WR_BLK_START	Sets the address of the first write block to be erased.
CMD33	ac	[31:0] data address	R1	ERASE_WR_BLK_END	Sets the address of the last write block of the continuous range to be erased.
CMD34 … CMD37	Reserved				
CMD38	ac	[31:0] don't care*	R1b	ERASE	Erases all previously selected write blocks.
CMD39 … CMD41	Reserved				

* The bit places must be filled but the value is irrelevant.

Table 4-8. Lock Card Commands (Class 7)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD42 …. CMD54	SDA Optional Commands, currently supported by SanDisk SD Card.				

Table 4-9. Application Specific Commands (Class 8)

CMD INDEX	Type	Argument	Resp.	Abbreviation	Command Description
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command
CMD56	adtc	[31:1] stuff bits. [0]: RD/ WR[1]	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block shall be set by the SET_BLOCK_LEN command.
CMD57 ... CMD59	Reserved				
CMD60 -63	Reserved for Manufacturer				

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

Table 4-10 describes all the application specific commands supported/reserved by the SD Card. All the following ACMDs shall be preceded with APP_CMD command (CMD55).

Table 4-10. Application Specific Commands Used/Reserved by SD Card

ACMD INDEX	Type	Argument	Resp.	Abbreviation	Command Description
ACMD6	ac	[31:2] stuff bits [1:0]bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4 bits bus) to be used for data transfer.
ACMD13	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Card status. The status fields are given in Table 4-28.
ACMD17	Reserved				
ACMD18	--	--	--	--	Reserved for SD security applications.[1]
ACMD19 to ACMD21	Reserved				
ACMD22	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_ BLOCKS	Send the number of the written (without errors) write blocks. Responds with 32bit+CRC data block.
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 applications.[1]
ACMD26	--	--	--	--	Reserved for SD security applications.[1]
ACMD38	--	--	--	--	Reserved for SD security applications.[1]
ACMD39 to ACMD40	Reserved				
ACMD41	bcr	[31:0]OCR without busy	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) con tent 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. The pull-up may be used for card detection.
ACMD43 ACMD49	--	--	--	--	Reserved for SD security applications.[1]
ACMD51	adtc	[31:0] staff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

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

4.9. Card State Transition Table

Table 4-11 defines the SD Card state transitions in dependency of the received command.

Table 4-11. Card State Transition Table

	Current State									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
Command	**Changes to**									
Class Independent										
CRC error	-	-	-	-	-	-	-	-	-	-
command not sup ported	-	-	-	-	-	-	-	-	-	-
Class 0										
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	-
CMD2	-	ident	-	-	-	-	-	-	-	-
CMD3	-	-	stby	stby	-	-	-	-	-	-
CMD4	-	-	-	stby	-	-	-	-	-	-
CMD7, card is addressed	-	-	-	tran	-	-	-	-	prg	-
CMD7, card is not addressed	-	-	-	stby	stby	stby	-	dis	-	-
CMD9	-	-	-	stby	-	-	-	-	-	-
CMD10	-	-	-	stby	-	-	-	-	-	-
CMD12	-	-	-	-	-	tran	prg	-	-	-
CMD13	-	-	-	stby	tran	data	rcv	prg	dis	-
CMD15	-	-	-	ina	ina	ina	ina	ina	ina	-
Class 2										
CMD16	-	-	-	-	tran	-	-	-	-	-
CMD17	-	-	-	-	data	-	-	-	-	-
CMD18	-	-	-	-	data	-	-	-	-	-
Class 4										
CMD16	See Class 2									
CMD24	-	-	-	-	rcv	-	-	-	-	-
CMD25	-	-	-	-	rcv	-	-	-	-	-
CMD27	-	-	-	-	rcv	-	-	-	-	-
Class 6										
CMD28	-	-	-	-	prg	-	-	-	-	-
CMD29	-	-	-	-	prg	-	-	-	-	-
CMD30	-	-	-	-	data	-	-	-	-	-
Class 5										

	Current State									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
CMD32	-	-	-	-	tran	-	-	-	-	-
CMD33	-	-	-	-	tran	-	-	-	-	-
CMD38	-	-	-	-	prg	-	-	-	-	-
Class 7										
CMD42	This is an SDA optional command supported by the SanDisk SD Card.									
Class 8										
CMD55	idle	-	-	stby	tran	data	rcv	prg	dis	-
CMD56; RD/WR = 0	-	-	-	-	rcv	-	-	-	-	-
CMD56; RD/WR = 1	-	-	-	-	data	-	-	-	-	-
ACMD6	-	-	-	-	tran	-	-	-	-	-
ACMD13	-	-	-	-	tran	-	-	-	-	-
ACMD22	-	-	-	-	tran	-	-	-	-	-
ACMD23	-	-	-	-	tran	-	-	-	-	-
ACMD18,25,26,38, 43,44,45,46,47,48,49	Refer to *SD Card Security Specification* for an explanation of the SD Security Features. The SanDisk SD Card supports all the security related commands as explained in the specification.									
ACMD41, card V_{DD} range compatible	ready	-	-	-	-	-	-	-	-	-
ACMD41, card is busy	idle	-	-	-	-	-	-	-	-	-
ACMD41, card V_{DD} range not compatible	ina	-	-	-	-	-	-	-	-	-
ACMD42	-	-	-	-	tran	-	-	-	-	-
ACMD51	-	-	-	-	data	-	-	-	-	-
class 9- 11										
CMD41; CMD43...CMD54, CMD57-CMD59	Reserved									
CMD60...CMD63	Reserved for manufacturer									

4.10. Responses

All responses are sent via the CMD line. The response transmission always starts with the MSB. The response length depends on the response type.

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

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

R1 (standard response): response length 48 bit.

Bits 45:40 indicate the index of the command to which it is responding. The status of the card is coded in 32 bits. Note that when a data transfer to the card is involved, a busy signal may appear on the data line after the transmission of each block of data. The host will check for busy after the data block transmission.

Table 4-12. Response R1

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

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 will check for busy at the response.

R2 (CID, CSD register): response length 136 bits.

The content of the CID register is sent as a response to CMD2 and CMD10. The content of the CSD register is sent as a response to CMD9. Only bits [127...1] of the CID and CSD are transferred, bit [0] of these registers is replaced by the end bit of the response.

Table 4-13. Response R2

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

R3 (OCR register): response length 48 bits.

The contents of the OCR register are sent as a response to ACMD41.

Table 4-14. Response R3

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

R4 and R5: responses are not supported.

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

Table 4-15. R6 Response

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 4-28)	CRC7	end bit

4.11. Timings

All timing diagrams use the schematics and abbreviations listed in Table 4-16.

Table 4-16. Timing Diagram Symbols

S	Start Bit (= 0)
T	Transmitter Bit (Host = 1, Card = 0)
P	One-cycle Pull-up (= 1)
E	End Bit (=1)
Z	High Impedance State (-> = 1)
D	Data Bits
X	Don't Care Data Bits (from Card)
*	Repeater
CRC	Cyclic Redundancy Check Bits (7 Bits)
	Card Active
	Host Active

The difference between the P-bit and Z-bit is that a P-bit is actively driven to HIGH by the card respectively host output driver, while Z-bit is driven to (respectively kept) HIGH by the pull-up resistors R_{CMD} respectively R_{DAT}. Actively-driven P-bits are less sensitive to noise. All timing values are defined in Table 4-17.

4.11.1. Command and Response

Both host command and card responses are clocked out with the rising edge of the host clock.

Card identification and card operation conditions timing

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

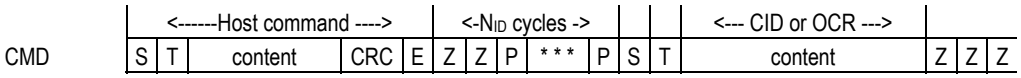


Figure 4-12. Identification Timing (Card Identification Mode)

Assign a card relative address

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

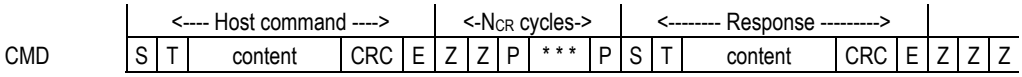


Figure 4-13 SEND_RELATIVE_ADDR Timing

Data transfer mode

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

Figure 4-13. Command Response Timing (Data Transfer Mode)

Last Card Response—Next Host Command Timing

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

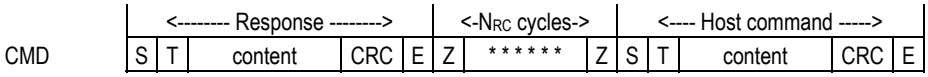


Figure 4-14. Timing Response End to Next CMD Start (Data Transfer Mode)

Last Host Command—Next Host Command Timing

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

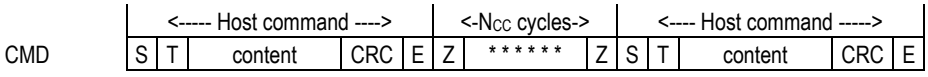


Figure 4-15. Timing of Command Sequences (All Modes)

4.11.2. Data Read

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

Single Block Read

The host selects one card for data read operation by CMD7, and sets the valid block length for block oriented data transfer by CMD16. The basic bus timing for a read operation is given in Figure 5-17. The sequence starts with a single block read command (CMD17) which specifies the start address in the argument field. The response is sent on the CMD line as usual.

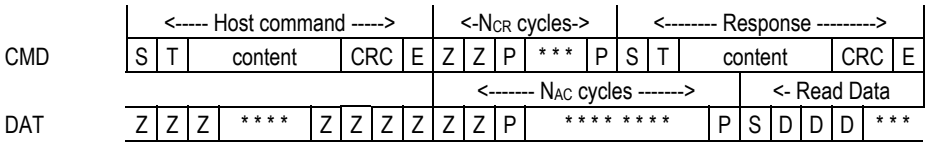


Figure 4-16. Timing of Single Block Read

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

Multiple Block Read

In multiple block read mode, the card sends a continuous flow of data blocks following the initial host read command. The data flow is terminated by a stop transmission command (CMD12). Figure 4-17 describes the timing of the data blocks and Figure 4-18 describes the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.

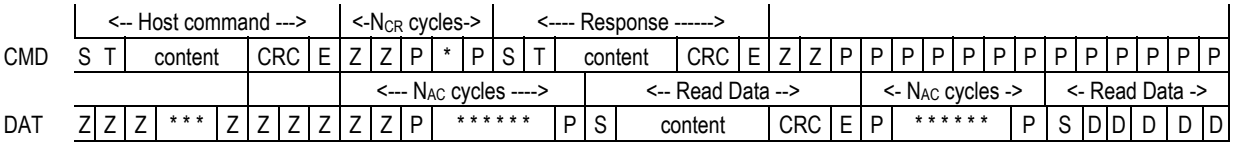


Figure 4-17. Timing of Multiple Block Read Command

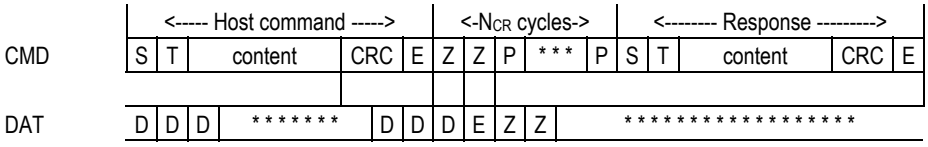


Figure 4-18. Timing of Stop Command (CMD12, Data Transfer Mode)

4.11.3. Data Write

Single Block Write

The host selects one card for data write operation by CMD7. The host sets the valid block length for block-oriented data transfer by CMD16.

The basic bus timing for a write operation is given in Figure 5-20. The sequence starts with a single block write command (CMD24) that determines (in the argument field) the start address. It is responded by the card on the CMD line as usual. The data transfer from the host starts N_{WR} clock cycles after the card response was received.

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

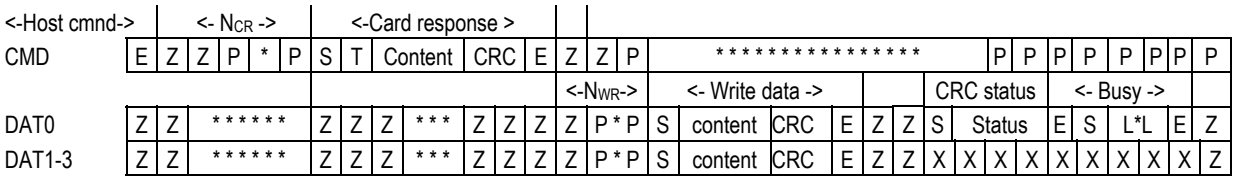


Figure 4-19. Timing of the Block Write Command

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

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

Multiple Block Write

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

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

The data flow is terminated by a stop transmission command (CMD12). Figure 4-20 describes the timing of the data blocks with and without card busy signal.

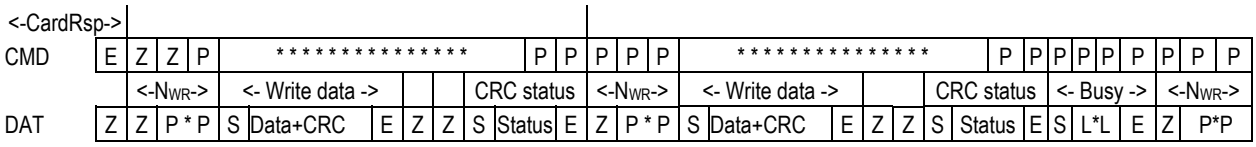


Figure 4-20. Timing of the Multiple Block Write Command

The stop transmission command works similar as in the read mode. Figures 4-21 through 4-24 describe the timing of the stop command in different card states.

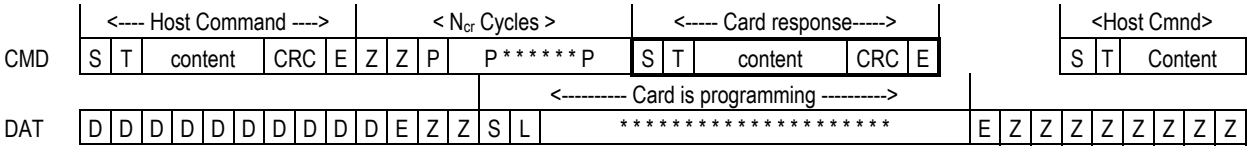
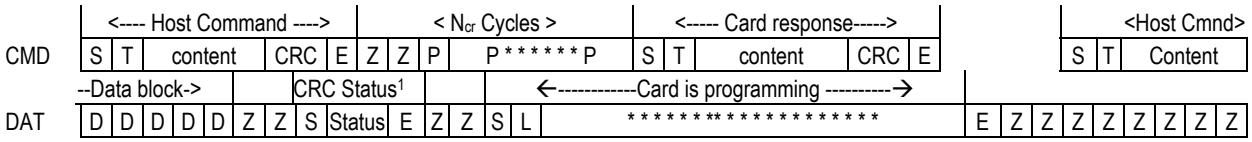


Figure 4-21. Stop Transmission During Data Transfer from the Host

The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status token sent back to the host. Figure 4-22 is an example of an interrupted (by a host stop command) attempt to transmit the CRC status block. The sequence is identical to all other stop transmission examples. The end bit of the host command is followed, on the data line, with one more data bit, end bit and two Z clock for switching the bus direction. The received data block, in this case is considered incomplete and will not be programmed.



1) The card CRC status response was interrupted by the host.

Figure 4-22. Stop Transmission During CRC Status Transfer from the Card

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

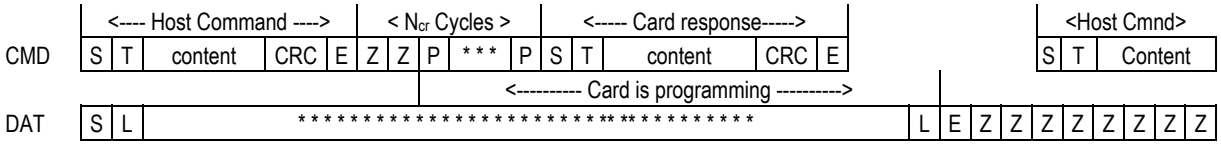


Figure 4-23. Stop Transmission Received After Last Data Block. Card is Busy Programming

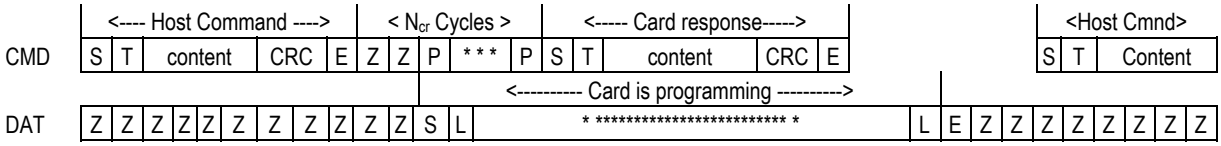


Figure 4-24. Stop Transmission Received After Last Data Block. Card becomes Busy

Erase, Set and Clear Write Protect Timing

The host must first tag the start (CMD32) and end (CMD33) addresses of the range to be erased. The erase command (CMD38), once issued, will erase all the selected write blocks. Similarly, set and clear write protect commands start a programming operation as well. The card will signal "busy" (by pulling the DAT line low) for the duration of the erase or programming operation. The bus transaction timings are the same as given for stop tran command in Figure 4-23.

Reselecting a Busy Card

When a busy card, which is currently in the dis state, is reselected it will reinstate its busy signaling on the data line. The timing diagram for this command/response/busy transaction is the same as given for stop tran command in Figure 4-24.

4.11.4. Timing Values

Table 4-17 defines all timing values. For more information, refer to Table 5-5 and 5.1.9.2 in Section 5.0, and the applications note in Appendix A, "Host Design Considerations: NAND MMC and SD-based Products."

Table 4-17. Timing Values

	Min.	Max.	Unit
N_{CR}	2	64	Clock Cycles
N_{ID}	5	5	Clock Cycles
N_{AC}	2	See note.	Clock Cycles
N_{RC}	8	-	Clock Cycles
N_{CC}	8	-	Clock Cycles
N_{WR}	2	-	Clock Cycles

NOTE: min [{(TAAC ∗ f) + (NSAC ∗ 100)}, {(100ms ∗ f)}] where units = (clocks) and "f" is the clock frequency.

5. SPI Protocol Definition

5.1. SPI Bus Protocol

While the SD Card channel is based on command and data bit-streams, which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of eight bit bytes and is byte aligned (multiples of eight clocks) to the CS signal.

Similar to the SD Bus protocol, the SPI messages are built from command, response and data-block 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 response behavior in SPI Bus mode differs from the SD Bus mode in the following three ways:

- The selected card always responds to the command.
- An eight or 16-bit response structure is used.
- When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than time-out as in the SD Bus mode.

In addition to the command response, every data block sent to the card during write operations will be responded with a special data response token. A data block may be as big as one card write block (WRITE_BL_LEN) and as small as a single byte.[1]

5.1.1. Mode Selection

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

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

The default command structure/protocol for SPI mode is that CRC checking is disabled. Since the card powers up in SD Bus mode, CMD0 must be followed by a valid CRC byte (even though the command is sent using the SPI structure). Once in SPI mode, CRCs are disabled by default.

CMD0 is a static command and always generates the same 7-bit CRC of 4Ah. Adding the "1," end bit (bit 0) to the CRC creates a CRC byte of 95h. The following hexadecimal sequence can be used to send CMD0 in all situations for SPI mode, since the CRC byte (although required) is ignored once in SPI mode. The entire CMD0 sequence appears as 40 00 00 00 00 95 (hexadecimal).

1) The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.

5.1.2. 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 transmitters and ignored by the receivers.

The SPI interface is initialized in the non-protected mode. The host can turn this option on and off using CRC_ON_OFF command (CMD59).

The CRC7/CRC16 polynomials are identical to that used in SD Bus mode. Refer to this section in the SD Bus mode chapter.

5.1.3. Data Read

SPI mode supports single block and multiple block read operations (SD Card CMD17 or CMD18). Upon reception of a valid read command the card will respond with a response token followed by a data token in the length defined in a previous SET_BLOCK_LENGTH (CMD16) command (see Figure 5-1).

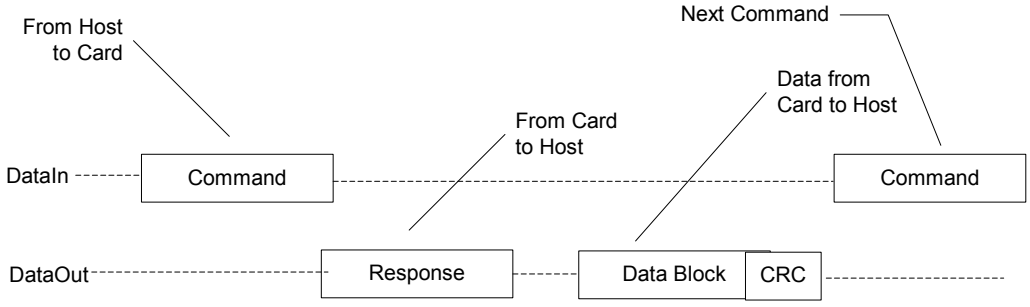


Figure 5-1. Single Block Read Operation

A valid data block is suffixed with a 16-bit CRC generated by the standard CCITT polynomial:

$$x^{16}+x^{12}+x^{5}+1.$$

The maximum block length is 512 bytes as defined by READ_BL_LEN (CSD parameter). Block lengths can be any number between 1 and READ_BL_LEN.

The start address can be any byte address in the valid address range of the card. Every block, however, must be contained in a single physical card sector.

In case of data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 5-2 shows a data read operation, which terminated with an error token rather than a data block.

Figure 5-2. Read Operation—Data Error

In the case of a Multiple Block Read operation, every transferred block has a 16-bit CRC suffix. The Stop Transmission command (CMD12) will actually stop the data transfer operation (the same as in SD Bus mode).

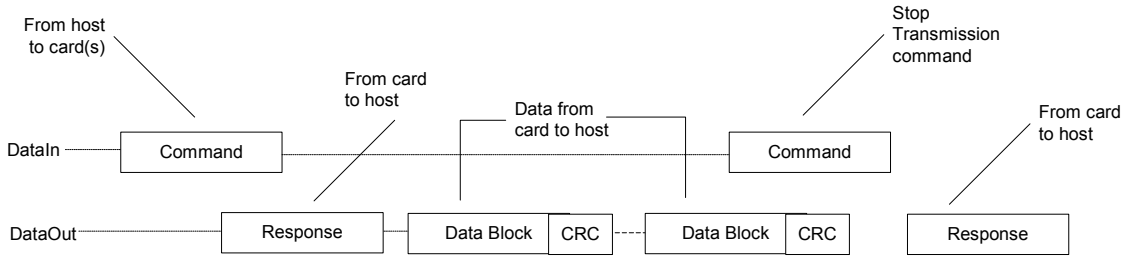


Figure 5-3. Multiple Block Read Operation

5.1.4. Data Write

In SPI mode, the SD Card supports single block or multiple block write operations. Upon reception of a valid write command (SD Card CMD24 or CMD25), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix and start address restrictions are identical to the read operation (see Figure 5-4). The only valid block length, however, is 512 bytes. Setting a smaller block length will cause a write error on the next write command.

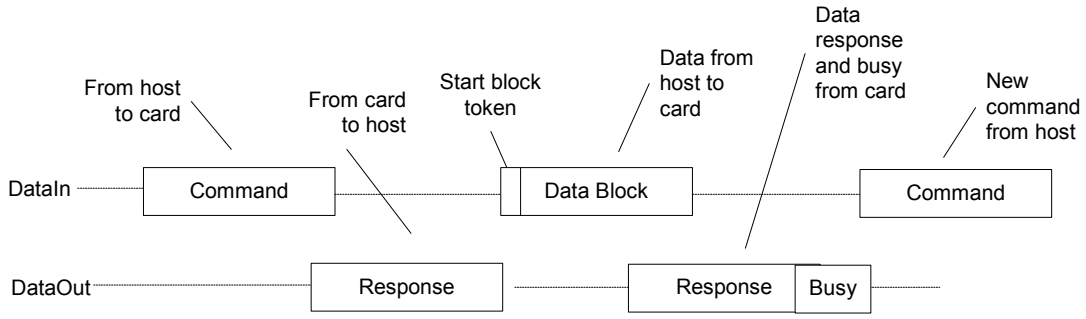


Figure 5-4. Single Block Write Operation

Every data block has a prefix or 'start block' token (one byte). After a data block is received the card will respond with a data-response token, and if the data block is received with no errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the dataOut line low).

Once the programming operation is completed, the host must check the results of the programming using the SEND_STATUS command (CMD13). Some errors (e.g., address out of range, write protect violation, etc.) are detected during programming only. The only validation check performed on the data block and communicated to the host via the data-response token is CRC and general Write Error indication.

In Multiple Block write operation the stop transmission will be done by sending 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block. In case of Write Error indication (on the data response) the host shall use SEND_NUM_WR_BLOCKS (ACMD22) in order to get the number of well written write blocks. The data token's description is given in Section 5.2.4.

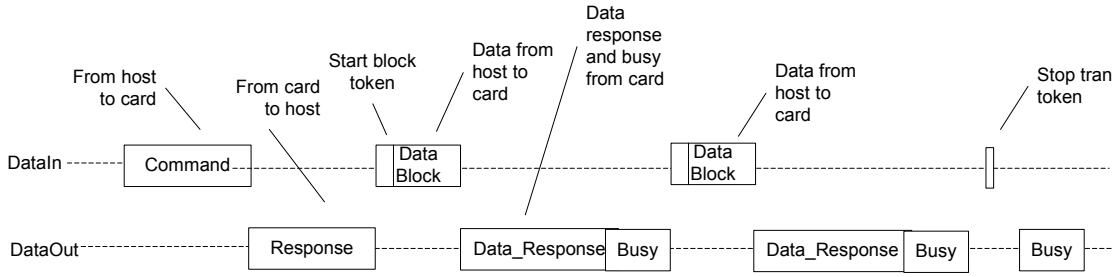


Figure 5-5. Multiple Block Write Operation

Resetting the CS signal while the card is busy does not terminate the programming process. The card releases the dataOut line (tristate) and continue to program. If the card is reselected before the programming is done, 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 the host's responsibility to prevent it.

5.1.5. Erase and Write Protect Management

The erase and write protect management procedures in the SPI mode are identical to the SD Bus 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 will hold the dataOut line low. Figure 5-6 illustrates a "no data" bus transaction with and without busy signaling.

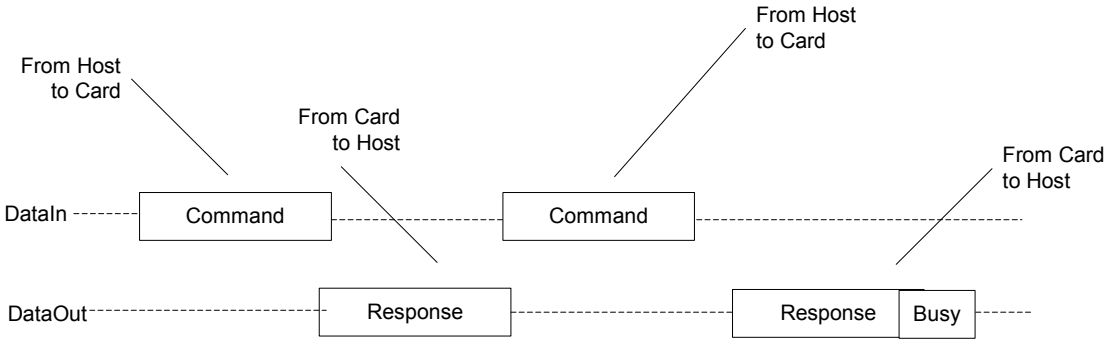


Figure 5-6. "No Data" Operations

5.1.6. Read CID/CSD Registers

Unlike the SD Bus protocol (where the register contents are 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 card TAAC since this value is stored in the CSD. Therefore, the standard response time-out value (N_{CR}) is used for read latency of the CSD register.

5.1.7. 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 legal host commands are CMD1 (SEND_OP_COND), ACMD41 (SD_SEND_OP_COND), CMD59 (CRC_ON_OFF) and CMD58 (READ_OCR).

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

In SPI mode, however, CMD1 has no operands and does not return the contents of the OCR register. Instead, the host can use CMD58 (SPI Mode Only) to read the OCR register. It is the responsibility of the host to refrain from accessing cards that do not support its voltage range.

The use of CMD58 is not restricted to the initialization phase only, but can be issued at any time. The host must poll the card (by repeatedly sending CMD1) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card has completed its initialization process and is ready for the next command.

5.1.8. Clock Control

The SPI bus clock signal can be used by the SPI host to set the cards to energy-saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to change the clock frequency or shut it down.

There are a few restrictions the SPI host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the SD Cards).
- It is an obvious requirement that the clock must be running for the SD Card to output data or response tokens. After the last SPI bus transaction, the host is required to provide 8 (eight) clock cycles for the card to complete the operation before shutting down the clock. Throughout this 8-clock period, the state of the CS signal is irrelevant. It can be asserted or de-asserted. Following is a list of the various SPI bus transactions:
  - A command/response sequence. Eight clocks after the card response end bit. The CS signal can be asserted or de-asserted during these 8 clocks.
  - A read data transaction. Eight clocks after the end bit of the last data block.
  - A write data transaction. Eight clocks after the CRC status token.
- The host is allowed to shut down the clock of a "busy" card. The SD Card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge, the SD Card (unless previously disconnected by de-asserting the CS signal) will force the dataOut line down, permanently.

5.1.9. Error Conditions

The following sections provide valuable information on error conditions.

5.1.9.1. CRC and Illegal Commands

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

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

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

5.1.9.2. Read, Write and Erase Time-out Conditions

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

For more information, refer to Table 4-17 in Section 4.0, Table 5-5 in Section 5.0 and the applications note in Appendix A, "Host Design Considerations: NAND MMC and SD-based Products."

Read

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

Write

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

Erase

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

5.1.10. Memory Array Partitioning

Same as for SD Card mode.

5.1.11. Card Lock/Unlock

The Card Lock/Unlock feature is currently in the SanDisk SD Card.

5.1.12. Application Specific Commands

The Application Specific commands are identical to SD mode with the exception of the APP_CMD status bit (see Section 3.5.5), which is not available in SPI.

5.1.13. Copyright Protection Commands

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

5.2. SPI Command Set

The following sections provide valuable information on the SPI Command Set.

5.2.1. Command Format

All the SD Card commands are 6 bytes long and transmitted MSB first.

Byte 1			Bytes 2—5	Byte 6	
7	6	5 0	31 0	7	0
0	1	Command	Command Argument	CRC	1

Commands and arguments are listed in Table 5-2.

7-bit CRC Calculation: $G(x) = x^{7} + x^{3} + 1$
$M(x) = (\text{start bit}) * x^{39} + (\text{host bit}) * x^{38} + ... + (\text{last bit before CRC}) * x^{0}$
$CRC[6...0] = \text{Remainder}[(M(x) * x^{7})/G(x)]$

5.2.2. Command Classes

As in SD mode, the SPI commands are divided into several classes (See Table 5-1). Each class supports a set of card functions. An SD Card will support the same set of optional command classes in both communication modes (there is only one command class table in the CSD register). The available command classes, and the supported commands for a specific class, however, are different in the SD Card and the SPI communication mode.

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

Table 5-1. Command Classes in SPI Mode

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

* The Lock Card command is supported in the SD Card.

5.2.2.1. Detailed Command Description

The following table provides a detailed description of the SPI bus commands. The responses are defined in Section 5.2.3. Table 5-2 lists all SD Card commands. A “yes” in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are reserved in SD Card mode as well.

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

Table 5-2. Description of SPI Bus Commands

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the SD Card
CMD1	Yes	None	R1	SEND_OP_COND	Activates the card's initialization process.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	Reserved				
CMD6	Reserved				
CMD7	No				
CMD8	Reserved				
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID).
CMD11	No				
CMD12	Yes	None	R1b	STOP _TRANSMISSION	Forces the card to stop transmission during a multiple block read operation.
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register.
CMD14	No				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read & write).[1]
CMD17	Yes	[31:0] data address	R1	READ_SINGLE _BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[2]
CMD18	Yes	[31:0] data address	R1	READ_MULTIPLE _BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_ TRANSMISSION command.
CMD19	Reserved				
CMD20	No				
CMD21 ... CMD23	Reserved				
CMD24	Yes	[31:0] data address	R1[3]	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[4]
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a stop transmission token is sent (instead of 'start block').

1) The only valid block length for write is 512 bytes. The valid block length for read is 1 to 512 bytes. A set block length of less than 512 bytes will cause a write error. The card has a default block length of 512 bytes. CMD16 is not mandatory if the default is accepted.
2) The start address and block length must be set so that the data transferred will not cross a physical block boundary.
3) Data followed by data response plus busy.
4) The start address must be aligned on a sector boundary. The block length is always 512 bytes.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD26	No				
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28[1]	Yes	[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[4]	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_ PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.[2]
CMD31	Reserved				
CMD32	Yes	[31:0] data address	R1	ERASE_WR_BLK_START _ADDR	Sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address	R1	ERASE_WR_BLK_END _ADDR	Sets the address of the last write block in a continuous range to be erased.
CMD34 .... CMD37	Reserved				
CMD38	Yes	[31:0] don't care*	R1b	ERASE	Erases all previously selected write blocks.
CMD39	No				
CMD40	No				
CMD41 ... CMD54	Reserved				
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Notifies the card that the next command is an application specific command rather than a standard command.
CMD56	Yes	[31:0] stuff bits [0]: RD/WR.[3]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. The size of the Data Block is defined with SET_BLOCK_LEN command.
CMD57	Reserved				
CMD58	Yes	None	R3	READ_OCR	Reads the OCR register of a card.
CMD59	Yes	[31:1] don't care* [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off.
CMD60-63	No				

* The bit places must be filled but the values are irrelevant.

1) These features are not currently supported in the SanDisk SD Card.
2) 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.
3) RD/WR_: "1"=the host will get a block of data from the card. "0"=the host sends a block of data to the card.

Table 5-3 describes all the application specific commands supported or reserved by the SD Card. All the following commands should be preceded with APP_CMD (CMD55).

Table 5-3. Application Specific Commands Used or Reserved by the SD Card–SPI Mode

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
ACMD6	No				
ACMD13	Yes	[31:0] stuff bits	R2	SD_STATUS	Send the SD Card status. The status fields are given in Table 4-21
ACMD17	Reserved				
ACMD18	Yes	--	--	--	Reserved for SD security applications1
ACMD19 to ACMD21	Reserved				
ACMD22	Yes	[31:0] stuff bits	R1	SEND_NUM_WR_ BLOCKS	Send the numbers of the well-written (without errors) blocks. Responds with 32bit+CRC data block.
ACMD23	Yes	[31:23] stuff bits [22:0]Number of blocks	R1	SET_WR_BLK_ ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). “1”=default (one wr block)(2).
ACMD24	Reserved				
ACMD25	Yes	--	--	--	Reserved for SD security applications1
ACMD26	Yes	--	--	--	Reserved for SD security applications1
ACMD38	Yes	--	--	--	Reserved for SD security applications1
ACMD39 to ACMD40	Reserved				
ACMD41	Yes	None	R1	SEND_OP_ COND	Activates the card’s initialization process.
ACMD42	Yes	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_ DETECT	Connect[1]/Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin 1) of the card. The pull-up may be used for card detection.
ACMD43 ... ACMD49	Yes	--	--	--	Reserved for SD security applications.[1]
ACMD51	Yes	[31:0] staff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

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

5.2.3. Responses

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

5.2.3.1. Format R1

This response token is sent by the card after every command with the exception of SEND_STATUS commands. It is 1 byte long, the MSB is always set to zero and the other bits are error indications. A '1' signals error.

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

The structure of the R1 format is shown in Figure 5-7.

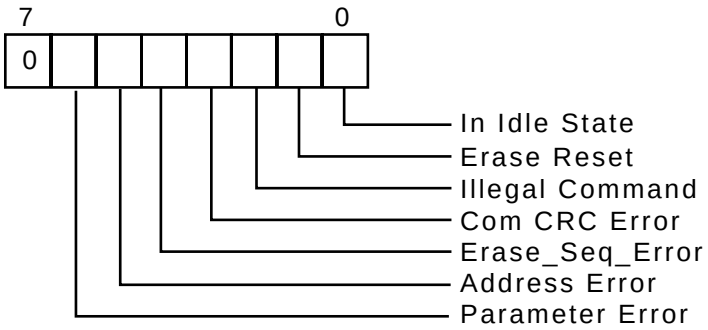


Figure 5-7. R1 Response Format

5.2.3.2. Format R1b

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

5.2.3.3. Format R2

This 2-bytes long response token is sent by the card as a response to the SEND_STATUS command. The format of the R2 status is shown in Figure 5-8.

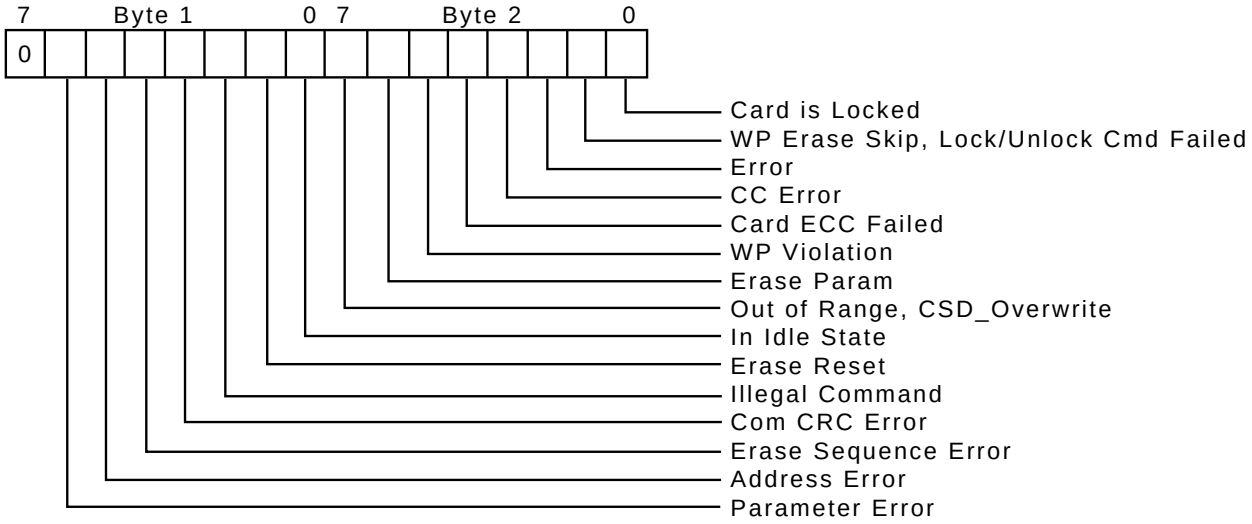


Figure 5-8. R2 Response Format

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

- **Erase param**—An invalid selection, sectors for erase.
- **Write protect violation**—The command tried to write a write-protected block.
- **Card ECC failed**—Card internal ECC was applied but failed to correct the data.
- **CC error**—Internal card controller error.
- **Error**—A general or an unknown error occurred during the operation.
- **Write protect erase skip**—Only partial address space was erased due to existing WP blocks.
- **Card is locked**—Supported by the SanDisk SD Card.

5.2.3.4. Format R3

This response token is sent by the card when a READ_OCR command is received. The response length is 5 bytes. The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

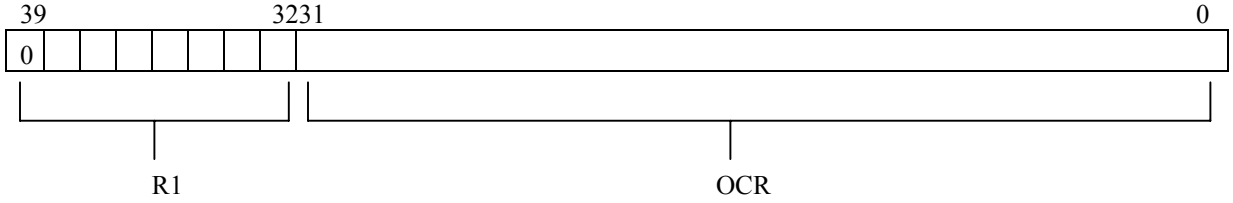
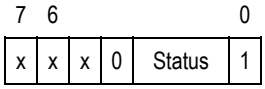


Figure 5-9. R3 Response Format

5.2.3.5. Data Response

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



The meaning of the status bits is defined as follows:

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

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

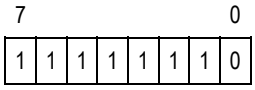
5.2.4. Data Tokens

Read and write commands have data transfers associated with them. Data is being transmitted or received via data tokens. All data bytes are transmitted MSB.

Data tokens are 4 to 515 bytes long and have the following format:

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

- First byte: Start Block.

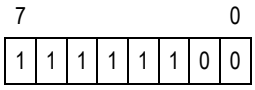


- Bytes 2-513 (depends on the data block length): User data.
- Last two bytes: 16-bit CRC.

For Multiple Block Write operation:

- First byte of each block.

If data is to be transferred then—Start Block



If Stop transmission is requested—Stop Tran

1	1	1	1	1	1	0	1

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

5.2.5. Data Error Token

If a read operation fails and the card cannot provide the required data it will send a data error token, instead. This token is one byte long and has the format shown in Figure 5-10.

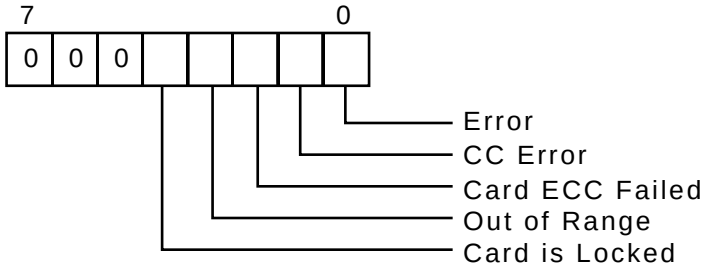


Figure 5-10. Data Error Token

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

5.2.6. Clearing Status Bits

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

5.3. Card Registers

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

5.4. SPI Bus Timing Diagrams

All timing diagrams use the schematics and abbreviations listed in Table 5-5.

Table 5-4. SPI Bus Timing Abbreviations

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

All timing values are defined in Table 5-5. The host must keep the clock running for at least N_{CR} clock cycles after the card response is received. This restriction applied to command and data response tokens.

5.4.1. Command/Response

Host Command to Card Response–Card is Ready

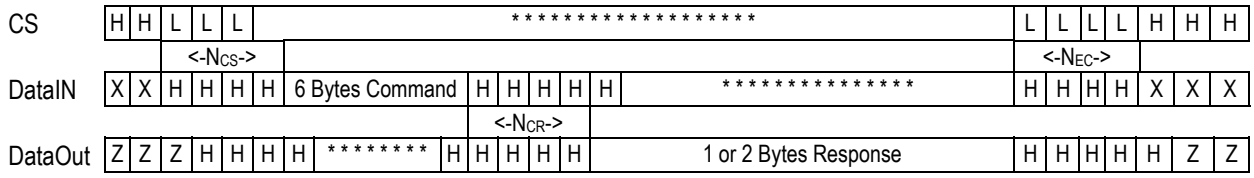


Figure 5-11. Host Command to Card Response–Card is Ready

Host Command to Card Response–Card is Busy

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

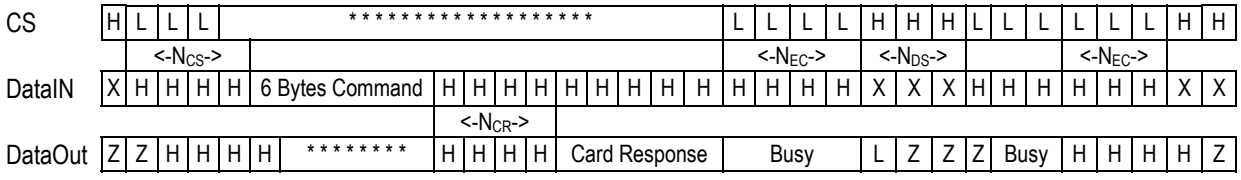


Figure 5-12. Host Command to Card Response–Card is Busy

Card Response to Host Command

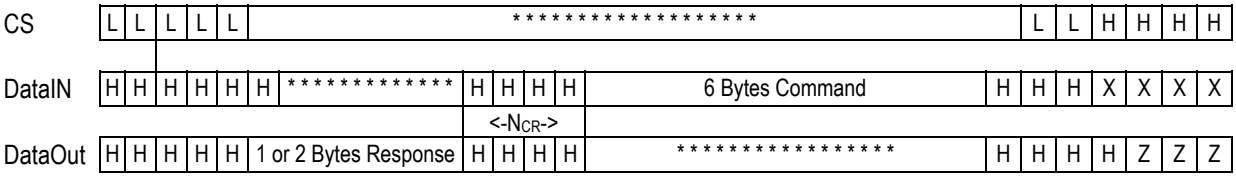


Figure 5-13. Card Response to Host Command

5.4.2. Data Read

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

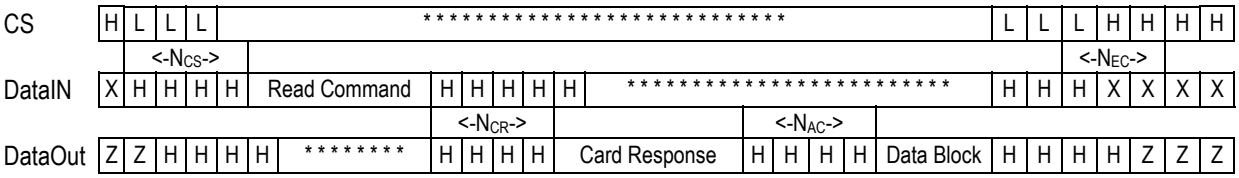


Figure 5-14. Single Block Read Timing

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

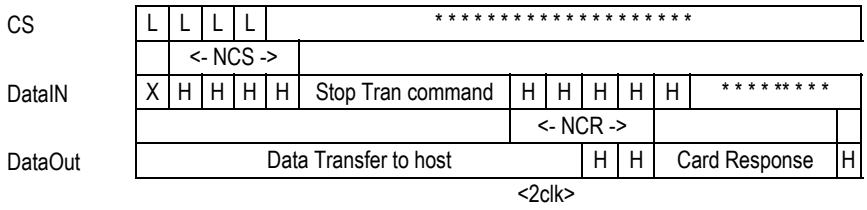


Figure 5-15. Multiple Block Read Timing

Reading the CSD Register

The following timing diagram describes the SEND_CSD command bus transaction. The timeout values for the response and the data block are N_{CR} (Since the N_{AC} is still unknown).

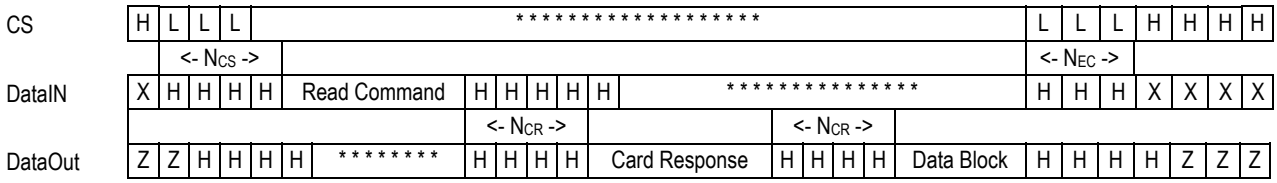


Figure 5-16. Reading the CSD Register

5.4.3. Data Write

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

The card will resume busy signal (pulling DataOut low) one clock after the falling edge of CS.

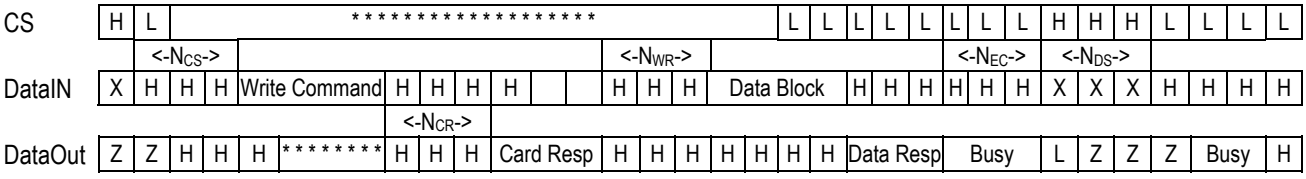
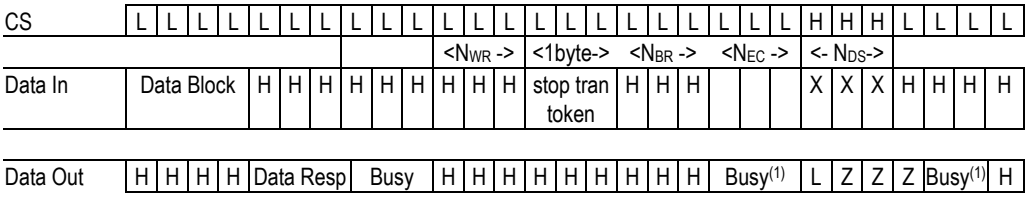


Figure 5-17. Device Write Timing

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



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

Figure 5-18. Stop Transmission Timing—Multiple Block Write

5.4.4. Timing Values

Table 5-5 shows the timing values and definitions. For more information, refer to Table 4-17 in Section 4.0, Section 5.1.9.2, and the applications note in Appendix A, "Host Design Considerations: NAND MMC and SD-based Products."

Table 5-5. Timing Constants Definitions

	Min	Max	Unit
N_{CS}	0	-	8 Clock Cycles
N_{CR}	0	8	8 Clock Cycles
N_{RC}	1	-	8 Clock Cycles
N_{AC}	1	See Note	8 Clock Cycles
N_{WR}	1	-	8 Clock Cycles
N_{EC}	0	-	8 Clock Cycles
N_{DS}	0	-	8 Clock Cycles
N_{BR}	0	1	8 Clock Cycles

NOTE: min [{{(TAAC * f) + (NSAC * 100)} * 1/8}, {(100ms * f) * 1/8}] where units = (8 clocks) and "f" is the clock frequency.

5.5. SPI Electrical Interface

The SPI Mode electrical interface is identical to that of the SD Card mode.

5.6. SPI Bus Operating Conditions

Identical to SD Card mode.

5.7. Bus Timing

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

This page intentionally left blank.

Appendix A. Application Note

Host Design Considerations: NAND MMC and SD-based Products

Introduction

SanDisk's MultiMediaCard (MMC) and Secure Digital (SD) Card have been designed into a wide variety of consumer electronic products: MP3 players, cell phones, PDAs, digital still and video cameras, data loggers, and more. Although these cards were designed to support this wide range of products, there are many options an engineer needs to consider before designing a card slot into a product. Design considerations include how the end product handles timeout delays, bus type selection, block mode selection, and other options. These can have a major impact on the performance and compatibility of the product. This Application Note will review these options and provide recommendations on the optimum way to manage them.

Timing

There are important timing issues for the engineer to consider when designing products that integrate the MultiMediaCard and/or SD Card.

Timing specifications

Design engineers must meet the rise, fall, setup, hold, and other SD Card and MultiMediaCard bus timing specifications. If they want to support MultiMediaCards in their design, the clock speed should be controllable by the host. This is due to the MultiMediaCard's open-drain mode; the MultiMediaCard powers up in the open-drain mode and cannot handle a clock faster than 400 Khz. Once the MultiMediaCard completes the initialization process, the card switches to the push-pull mode. In the push-pull mode the MultiMediaCard can run at the maximum clock speed.

Refer to www.mmca.org and www.sdcard.org for timing specifications published by MultiMediaCard and SD Card Associations.

Read access and program times

Read access and program times are also very critical to the proper operation of a product design. If the time-out values for read access and program time are not met, data read from and written to the card may be incorrect or invalid. MultiMediaCard and SD Card manufacturers have different read and write time-out values, and the designer must ensure that the product time-out value is not set below the maximum specification.

The maximum read and write time-out values for the MultiMediaCard and SD Card are shown in Table 1.

Table 1. MultiMediaCard and SD Card Maximum Read/Write Time-out Values

Product	Time-out Values	
MultiMediaCard	**Typical**	**Maximum**
Read	(TAAC + NSAC)	10 * (TAAC + NSAC)
Write	(TAAC + NSAC) * R2W_FACTOR	(TAAC) + NSAC) * R2W_FACTOR * 10
SD Card		

Read	(TAAC + NSAC)	100ms
Write	(TAAC + NSAC) * R2W_FACTOR	250 ms

The factors used in calculating the values in Table 1—TAAC, NSAC, and R2W_FACTOR—can be read directly from the CSD register of the MultiMediaCard and SD Card.

The TAAC factor's unit is time, and the NSAC factor has units of 100 clocks. You can convert TAAC units to clock cycles by multiplying by the frequency of the clock and calculate the time-outs in units of clock cycles if desired. Alternatively, given the frequency of the clock, you can convert the NSAC units to time and calculate the time-outs in units of time.

The R2W_FACTOR is a read-to-write factor and has no units. A design engineer can use the time-out values derived from the CSD register to make the design compatible with all MultiMediaCards and SD cards regardless of customer brand.

Interface

The MultiMediaCard and SD Card support multiple busses. Both cards support the 1-bit SPI bus that includes bus pins DATin, DATout, CLK, and CS. The SPI bus is generally found on Motorola and other major MCU manufacturer products.

The SD Card also supports a 4-bit and a 1-bit SD bi-directional bus mode. SD bus pins are CLK, CMD, and DAT in 1-bit mode and CLK, CMD, and DAT[0:3] in 4-bit mode.

The MultiMediaCard also supports the 1-bit bi-directional MMC bus mode that has CLK, CMD, and DAT bus pins. The CMD and DAT pins are bi-directional on the SD 1-bit, SD 4-bit, and MMC 1-bit.

The maximum burst rate achievable with the SD Card and MultiMediaCard depends on the clock speed and bus mode. The burst rate is the data transfer rate between the card's buffer and host.

Table 2. MultiMediaCard and SD Card Clock Speed and Burst Rate

Product	Maximum Clock Speed and Burst Rate	
MultiMediaCard	**Clock Speed**	**Burst Rate**
SPI Bus mode	20 MHz	2.5 MB/s
MMC 1-bit mode	20 MHz	2.5 MB/s
SD Card		
SPI Bus mode	25 MHz	3.125 MB/s
SD 1-bit mode	25 MHz	3.125 MB/s
SD 4-bit mode	25 MHz	12.5 MB/s

The write and read throughput rates of the SD Card and MultiMediaCard are slower than the burst rate because each card includes the busy time to write data from the card's buffers to its internal Flash RAM, and busy time to read data from the internal Flash RAM to the card's buffer. Since most designs use this write and read busy time to complete other processes, choosing a 1- or 4-bit bus mode can have a 4x speed effect on the time spent servicing the SD Card.

The example in Table 3 shows the difference between moving 512 bytes of data to and from a MultiMediaCard or SD Card internal buffer using different bus modes.

Table 3. MultiMediaCard and SD Card Clock Speed and Transfer Time

Product	Maximum Clock Speed and Time Req. to move 512 bytes	
MultiMediaCard	**Clock Speed**	**Time**
SPI Bus mode	20 MHz	204.8 us
MMC 1-bit mode	20 MHz	204.8 us
SD Card		
SPI Bus mode	25 MHz	163.8 us
SD 1-bit mode	25 MHz	163.8 us
SD 4-bit mode	25 MHz	41 us

Read/Write Mode Selection

Another major MultiMediaCard and SD Card design consideration is the use of Singleblock or Multiblock command modes. **Singleblock** mode reads and writes data one block at a time; **Multiblock** mode reads and writes multiple blocks until a stop command is received.

Multiblock mode takes advantage of the multiple internal block buffers present in all MultiMediaCards or SD Cards. In Multiblock mode, when one block buffer gets full during write, the card gives the host access to the other empty block buffers to fill while programming the first block. The card does not enter a busy state until all block buffers are full.

In Singleblock mode, the card enters a busy state by forcing the DAT line low when the first block buffer is full and remains busy until the write process is complete. During the busy state, the host cannot send any additional data to the card because the card forces the DAT line low.

If speed is critical in a design, Multiblock mode is the faster and recommended mode. The more blocks that can be written in Multiblock mode the better the performance of the design. Therefore when planning the design, ensure that enough system RAM is designed in to support the multiblock capability. The performance gain will always outweigh the cost of the extra RAM. However, if speed is not critical—for example, a data-logger design that records only 512 bytes of data every minute—Singleblock mode is more than adequate.

Power and Clock Control

Power control should be considered when creating designs using the MultiMediaCard and/or SD Card. The ability to have software power control of the cards makes the design more flexible and robust. The host will have the ability to turn power to the card on or off independent of whether the card is inserted or removed. This can help with card initialization when there is contact bounce during card insertion. The host waits a specified time after the card is inserted before powering up the card and starting the initialization process. Also, if the card goes into an unknown state, the host can cycle the power and start the initialization process again. When card access is unnecessary, allowing the host to power-down the bus can reduce overall power consumption.

Clock control is another option that should be implemented in a MultiMediaCard or SD Card design. As mentioned in the ***Timing*** section, if the design needs to support the MultiMediaCard, the clock should be lowered to 400 kHz or less during initialization. When the initialization process is complete, the host can raise the clock speed to the card's maximum.

Initialization Algorithm

The initialization algorithm needs to be considered for products designed to support the MultiMediaCard and SD Card or SD Card only. An SD socket is physically thicker which allows both types of cards to be inserted. Therefore, the host needs to be able to detect which card is inserted into the socket.

When the SD initialization command is used first, it causes the MultiMediaCard to return an error that provides the host with an identification of the card type. If the host is supporting both the MultiMediaCard and SD Card, it can continue the initialization using the MMC commands. If the host does not support both cards, it issues an error message instructing the user to insert an SD Card.

If the design uses a MultiMediaCard socket, the host can start the initialization with the MMC command. The host does not need to detect which type of card is inserted because the SD Card will not physically fit into an MMC socket.

File System Support

If a design needs to support a file system, such as SanDisk's Host Developers Tool Kit (HDTK), additional considerations are necessary.

Reading and writing to an SD Card and MultiMediaCard is generally done in 512 byte blocks, however, erasing often occurs in much larger blocks. The NAND architecture used by SanDisk and other card vendors currently has Erase Block sizes of (32) or (64) 512 byte blocks, depending on card capacity. In order to re-write a single 512 byte block, all other blocks belonging to the same Erase Block will be simultaneously erased and need to be rewritten.

For example—writing a file to a design using a FAT file system takes three writes/updates of the system area of FAT and one write/update of the data area to complete the file write. First, the directory has to be updated with the new file name. Second, the actual file is written to the data area. Third, the FAT table is updated with the file data location. Finally, the directory is updated with the start location, length, date and time the file was modified. Therefore, when selecting the file size to write into a design, the size should be as large as possible and a multiple of the Erase Block size. This takes advantage of the architecture.

Some designs update the FAT table for every cluster of the data file written. This can slow the write performance, because the FAT table is constantly being erased and re-written. The best approach is to write all the file clusters then update the FAT table once to avoid the performance hit of erasing and re-writing all the blocks within the Erase Block multiple times.

Appendix B. Ordering Information

To order SanDisk products directly from SanDisk, call 408-542-0595.

Secure Digital Card

SDSDB-16	16 MB
SDSDB-32	32 MB
SDSDJ-64	64 MB
SDSDJ-128	128 MB
SDSDJ-256	256 MB
SDSDJ-512	512 MB
SDSDJ-1024	1024 MB

Appendix C. SanDisk Worldwide Sales Offices

To order SanDisk products directly from SanDisk, call 408-542-0595.

SanDisk Corporate Headquarters

140 Caspian Court
Sunnyvale, CA 94089
Tel: 408-542-0500
Fax: 408-542-0503
http://www.sandisk.com

U.S. Industrial/OEM Sales

Northwest USA
2241 Fremont Dr., Suite B
Havasu City, AZ 86406
Tel: 928-505-4258
Fax: 928-505-4259

Southwest USA & Mexico
140 Caspian Court
Sunnyvale, CA 94089
Tel: 408-542-0730
Fax: 408-542-0410

North Central USA & South America
134 Cherry creek Circle, Suite 150
Winter Springs, FL 32708
Tel: 407-366-6490
Fax: 407-366-5945

Northeastern USA & Canada
620 Herndon Pkwy. Suite 200
Herndon, VA 22070
Tel: 703-481-9828
Fax: 703-437-9215

U.S. Retail Sales

Americas
10 Flagstone
Trabuco Canyon, CA 92679
Tel: 949-589-8351
Fax: 949-589-8364

Retail Account Sales
32500 Mills Rd.
Avon, OH 44011
Tel: 440-327-0490
Fax: 440-327-0295

International Retail Sales

European Retail Sales
Wilhelminastraat 10
2011 VM Haarlem
The Netherlands
Tel: 31-23-5514226
Fax: 31-23-5348625

Southern European Retail Sales
Centre Hoche Condorcet
3 Rue Condorcet—B.P. 9
91263 Juvisy Sur Orge Cedex
France
Tel: 33-169-12-16-04
Fax: 33-169-12-16-24

Japan Retail Sales
Umeda-Shinmichi Bldg. 10F
1-1-5 Dojima, Kita-ku
Osaka 530-0003
Tel: 81-6-6343-6480
Fax: 81-6-6343-6481

International Industrial/OEM Sales

Europe
SanDisk GmbH
Karlsruher Str. 2C
D-30519 Hannover, Germany
Tel: 49-511-875-9131
Fax: 49-511-875-9187

Northern Europe
Videroegatan 3 B
S-16440 Kista, Sweden
Tel: 46-08-75084-63
Fax: 46-08-75084-26

Central and Southern Europe
Rudolf-Diesel-Str. 3
40822 Mettmann, Germany
Tel: 49-210-495-3433
Fax: 49-210-495-3434

Japan
8F Nisso Bldg. 15
2-17-19 Shin-Yokohama,
Kohoku-ku
Yokohama 222-0033,
Japan
Tel: 81-45-474-0181
Fax: 81-45-474-0371

Asia/Pacific Rim
89 Queensway, Lippo Center
Tower I, Suite 3402
Admiralty, Hong Kong
Tel: 852-2712-0501
Fax: 852-2712-9385

This page intentionally left blank.

Appendix D. Limited Warranty

I. WARRANTY STATEMENT

SanDisk warrants its products to be free of any defects in materials or workmanship that would prevent them from functioning properly for one year from the date of purchase. This express warranty is extended by SanDisk Corporation to its customers.

II. GENERAL PROVISIONS

This warranty sets forth the full extent of SanDisk's responsibilities regarding the SanDisk Secure Digital Card. In satisfaction of its obligations hereunder, SanDisk, at its sole option, will either repair, replace or refund the purchase price of the product.

NOTWITHSTANDING ANYTHING ELSE IN THIS LIMITED WARRANTY OR OTHERWISE, THE EXPRESS WARRANTIES AND OBLIGATIONS OF SELLER AS SET FORTH IN THIS LIMITED WARRANTY, ARE IN LIEU OF, AND BUYER EXPRESSLY WAIVES ALL OTHER OBLIGATIONS, GUARANTIES AND WARRANTIES OF ANY KIND, WHETHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OR INFRINGEMENT, TOGETHER WITH ANY LIABILITY OF SELLER UNDER ANY CONTRACT, NEGLIGENCE, STRICT LIABILITY OR OTHER LEGAL OR EQUITABLE THEORY FOR LOSS OF USE, REVENUE, OR PROFIT OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES, INCLUDING WITHOUT LIMITATION PHYSICAL INJURY OR DEATH, PROPERTY DAMAGE, LOST DATA, OR COSTS OF PROCUREMENT OF SUBSTITUTE GOODS, TECHNOLOGY OR SERVICES. IN NO EVENT SHALL THE SELLER BE LIABLE FOR DAMAGES IN EXCESS OF THE PURCHASE PRICE OF THE PRODUCT, ARISING OUT OF THE USE OR INABILITY TO USE SUCH PRODUCT, TO THE FULL EXTENT SUCH MAY BE DISCLAIMED BY LAW.

SanDisk's products are not warranted to operate without failure.

III. WHAT THIS WARRANTY COVERS

For products found to be defective within one year of purchase, SanDisk will have the option of repairing or replacing the defective product, if the following conditions are met:

A. The defective product is returned to SanDisk for failure analysis as soon as possible after the failure occurs.
B. An incident card filled out by the user, explaining the conditions of usage and the nature of the failure, accompanies each returned defective product.
C. No evidence is found of abuse or operation of products not in accordance with the published specifications, or of exceeding storage or maximum ratings or operating conditions.

All failing products returned to SanDisk under the provisions of this limited warranty shall be tested to the product's functional and performance specifications. Upon confirmation of failure, each product will be analyzed, by whatever means necessary, to determine the root cause of failure. If the root cause of failure is found to be not covered by the above provisions, then the product will be returned to the customer with a report indicating why the failure was not covered under the warranty.

This warranty does not cover defects, malfunctions, performance failures or damages to the unit resulting from use in other than its normal and customary manner, misuse, accident or neglect; or improper alterations or repairs.

SanDisk reserves the right to repair or replace, at its discretion, any product returned by its customers, even if such product is not covered under warranty, but is under no obligation to do so.

SanDisk may, at its discretion, ship repaired or rebuilt products identified in the same way as new products, provided such cards meet or exceed the same published specifications as new products. Concurrently, SanDisk also reserves the right to market any products, whether new, repaired, or rebuilt, under different specifications and product designations if such products do not meet the original product's specifications.

IV. RECEIVING WARRANTY SERVICE

According to SanDisk's warranty procedure, defective product should be returned only with prior authorization from SanDisk Corporation. Please contact SanDisk's Customer Service department at 408-542-0595 with the following information: product model number and description, nature of defect, conditions of use, proof of purchase and purchase date. If approved, SanDisk will issue a Return Material Authorization or Product Repair Authorization number. Ship the defective product to:

SanDisk Corporation
Attn: RMA Returns
(Reference RMA or PRA #)
140 Caspian Court
Sunnyvale, CA 94089

V. STATE LAW RIGHTS

SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF INCIDENTAL OR CONSEQUENTIAL DAMAGES, OR LIMITATION ON HOW LONG AN IMPLIED WARRANTY LASTS, SO THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU. This warranty gives you specific rights and you may also have other rights that vary from state to state.

Appendix E. Disclaimer of Liability

SanDisk Corporation general policy does not recommend the use of its products in life support applications wherein a failure or malfunction of the product may directly threaten life or injury. Accordingly, in any use of products in life support systems or other applications where failure could cause damage, injury or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features.

SanDisk shall not be liable for any loss, injury or damage caused by use of the Products in any of the following applications:

- Special applications such as military-related equipment, nuclear reactor control, and aerospace
- Control devices for automotive vehicles, train, ship and traffic equipment
- Safety system for disaster prevention and crime prevention
- Medical-related equipment including medical measurement device