SanDisk MultiMediaCard and Reduced-Size MultiMediaCard

Product Manual

Version 1.3

Document No. 80-36-00320

April 2005

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 "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. CompactFlash is a U.S. registered trademark 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. 80-36-00320 Rev.1.3 04/05 Printed in U.S.A.

Revision History

April 2004	*Revision 0.1—Initial release*
April 2004	*Revision 0.2—Minor edits and addition of RS-MMC physical dimensions*
May 2004	*Revision 1.0—Added document number and updated to v1.0 release*
July 2004	*Revision 1.1—Corrected TAAC value in CSD Register; updated App B*
Oct 2004	*Revision 1.2—Changed specific values in Table 3-10; updated sales office information*
Mar/Apr 2005	*Revision 1.3—Added part numbers to Table 1-1, revised values to reflect some changes to the MMCA spec, other minor and technical changes*

1 Introduction

1.1 General Description

The SanDisk MultiMediaCard and Reduced-Size MultiMediaCard (RS-MMC) are very small, removable flash storage devices, designed specifically for storage applications that put a premium on small form factor, low power and low cost. Flash is the ideal storage medium for portable, battery-powered devices. It features low power consumption and is non-volatile, requiring no power to maintain the stored data. It also has a wide operating range for temperature, shock and vibration.

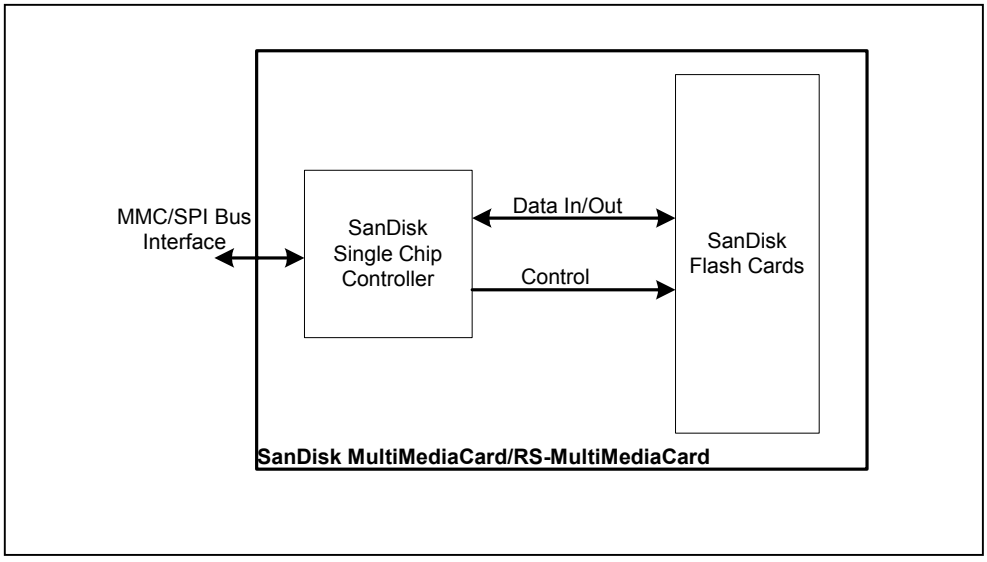
The MultiMediaCard and RS-MultiMediaCard are well suited to meet the needs of small, low power, electronic devices. With form factors of 32 mm x 24 mm and 1.4 mm thick for the MultiMediaCard and 18 mm x 24 mm x1.4 mm for the RS-MultiMediaCard, these cards can be used in a wide variety of portable devices like mobile phones, and voice recorders.

To support this wide range of applications, the MultiMediaCard Protocol, a simple seven-pin serial interface, is designed for maximum scalability and configurability. All device and interface configuration data (such as maximum frequency, card identification, etc.) are stored on the card.

The SanDisk MultiMediaCard/RS-MultiMediaCard interface allows for easy integration into any design, regardless of microprocessor used. For compatibility with existing controllers, the card offers, in addition to the card interface, an alternate communication protocol, which is based on the Serial Peripheral Interface (SPI) standard.

The MultiMediaCard/RS-MultiMediaCard provides up to 256 million bytes of memory using SanDisk Flash memory chips, which were designed by SanDisk especially for use in mass storage applications. In addition to the mass storage specific flash memory chip, the MultiMediaCard/RS-MultiMediaCard includes an on-card intelligent controller which manages interface protocols and data storage and retrieval, as well as Error Correction Code (ECC) algorithms, defect handling and diagnostics, power management and clock control.

Figure 1-1 MultiMediaCard/RS-MultiMediaCard Block Diagram



1.2 Features

The SanDisk MultiMediaCard/RS-MultiMediaCard features include:

- **MultiMediaCard Protocol compatible**
- **SPI Mode supported**
- **Targeted for portable and stationary applications**
- **Voltage range**
  - Basic communication: 2.7 to 3.6 V
  - Memory access: 2.7 to 3.6 V
- **Maximum data rate with up to 10 cards**
- **Correction of memory field errors**
- **Built-in write protection features (permanent and temporary)**
- **Variable clock rate 0 - 20 Mhz**
- **Multiple cards stackable on a single physical bus**

1.3 Document Scope

This document describes the key features and specifications of the SanDisk MultiMediaCard (full-size) and RS-MultiMediaCard (reduced-size), as well as the information required to interface this product to a host system. Retail card specifications are not covered in this manual.

1.4 Product Models

The SanDisk MultiMediaCard and RS-MultiMediaCard is available in a variety of capacities as shown in Table 2-6, Chapter 2: *Product Specifications*. The MultiMediaCard is available in full-size and reduced-size (RS-MMC) form factors.

1.5 MultiMediaCard Standard

MultiMediaCards and RS-MultiMediaCards are fully compatible with the MultiMediaCard standard specification listed below:

The MultiMediaCard System Specification, Version 3.3

This specification may be obtained from:

MultiMediaCard Association
19672 Stevens Creek Blvd., Suite 404
Cupertino, CA 95014-2465
USA
Phone: 408-253-0441
Fax: 408-253-8811
Email: prophet2@mmca.org

http://www.mmca.org

1.6 Functional Description

The MultiMediaCard and RS-MultiMediaCard contain a high level, intelligent subsystem as shown by the block diagram 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
- Power management for low power operation

1.7 Flash-Independent Technology

The 512-byte sector size of the MultiMediaCard and RS-MultiMediaCard 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 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 MultiMediaCard and RS-MultiMediaCard 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 SanDisk MultiMediaCard/RS-MultiMediaCard today will be able to access future cards built with new flash technology without having to update or change host software.

1.8 Defect and Error Management

The SanDisk MultiMediaCard and RS-MultiMediaCard 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 example, 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. MultiMediaCards and RS-MultiMediaCards 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, the cards replace it with a spare bit within the sector header. If necessary, the card will replace the entire sector with a spare sector. This is completely transparent to the host and does not consume any user data space.

The 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, a SanDisk MultiMediaCard/RS-MultiMediaCard possesses 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 SanDisk MultiMediaCards and RS-MultiMediaCards unparalleled reliability.

1.9 Endurance

The SanDisk MultiMediaCard and RS-MultiMediaCard have a typical endurance specification for each sector of 100,000 writes (reading a logical sector is unlimited). This far exceeds what is required in nearly all card applications. For example, very heavy use of the MultiMediaCard/RS-MultiMediaCard in cellular phones, personal communicators, pagers and voice recorders will use only a fraction of the total endurance over the device's typical lifetime. That means it would take over 34 years to wear out an area of the 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. However, for the vast majority of users employing typical applications, the endurance limit is not of any practical concern.

1.10 Automatic Sleep Mode

A unique feature of the SanDisk MultiMediaCard/RS-MultiMediaCard is automatic entrance and exit from sleep mode. Upon completion of an operation, the card enters the sleep mode to conserve power if no further commands are received in less than five milliseconds (ms). The host does not have to take any action for this to occur. However, in order to achieve the lowest sleep current, the host needs to shut down its clock to the card. In most systems, the MultiMediaCard/RS-MultiMediaCard is in sleep mode except when the host is accessing it, thus conserving power.

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

1.11 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. This approach is similar to that used in PCMCIA and MMCA devices to allow for hot insertion and applies to MultiMediaCard and SPI modes.

1.12 MultiMediaCard Mode

The following sections provide valuable information on the MultiMediaCard and RS-MultiMediaCard in MultiMediaCard mode.

1.12.1 MultiMediaCard Standard Compliance

The MultiMediaCard and RS-MultiMediaCard are fully compliant with *the MultiMediaCard Standard Specification, v3.3*. The structure of the Card Specific Data (CSD) Register is compliant with *CSD Structure v1.2*.

1.12.2 Negotiating Operating Conditions

The MultiMediaCard and RS-MultiMediaCard support the operation condition verification sequence defined in the *MultiMediaCard Standard Specification, v3.3*. If the host defines an operating voltage range not supported by the card, the card will go into an inactive state

and ignore any bus communication. The only way to get a card out of an inactive state is by powering the card down and up again.

In addition, the host can explicitly send either card to an inactive state by using the GO_INACTIVE_STATE command.

1.12.3 Card Acquisition and Identification

The MultiMediaCard and RS-MultiMediaCard bus is a single master (host application) and multi-slaves (cards) bus. The host can query the bus and determine how many cards, and of which type, are connected. The card's CID Register is pre-programmed with a unique card identification number used during the identification procedure.

In addition, the card host can read either card's CID Register using the READ_CID command. The CID Register is programmed during MultiMediaCard and RS-MultiMediaCard testing and formatting procedures during manufacturing. The card host can only read, and not write to the CID Register.

1.12.4 Card Status

The MultiMediaCard and RS-MultiMediaCard status is stored in a 32-bit status register that is sent as the data field, in the card response to host commands. The Status Register provides information about the card's active state and completion codes for the last host command

The card status can be explicitly polled with the SEND_STATUS command.

1.12.5 Memory Array Partitioning

The MultiMediaCard/RS-MultiMediaCard memory space is byte addressable with addresses ranging from 0 to the last byte, and is divided into several structures.

Memory bytes are grouped into 512-byte blocks called **sectors**. Every block can be read, written and erased individually. *Sectors* are grouped into **erase groups** of 16 or 32 sectors depending on card size. Any combination of sectors within one group, or any combination of erase groups can be erased with a single erase command. A write command implicitly erases the memory before writing new data into it. An explicit erase command can be used for pre-erasing memory and speed up the next write operation. *Erase groups* are grouped into **Write Protect Groups** (WPG) of 32 erase groups. The write/erase access to each WPG can be limited individually. The last (highest in address) WPG will be smaller and contain less than 32 erase groups.

A diagram of the memory structure hierarchy is shown in Figure 1-2. Table 1-1 summarizes the number of various memory structures for the different MultiMediaCards and RS-MultiMediaCards.

Figure 1-2 Memory Array Partitioning

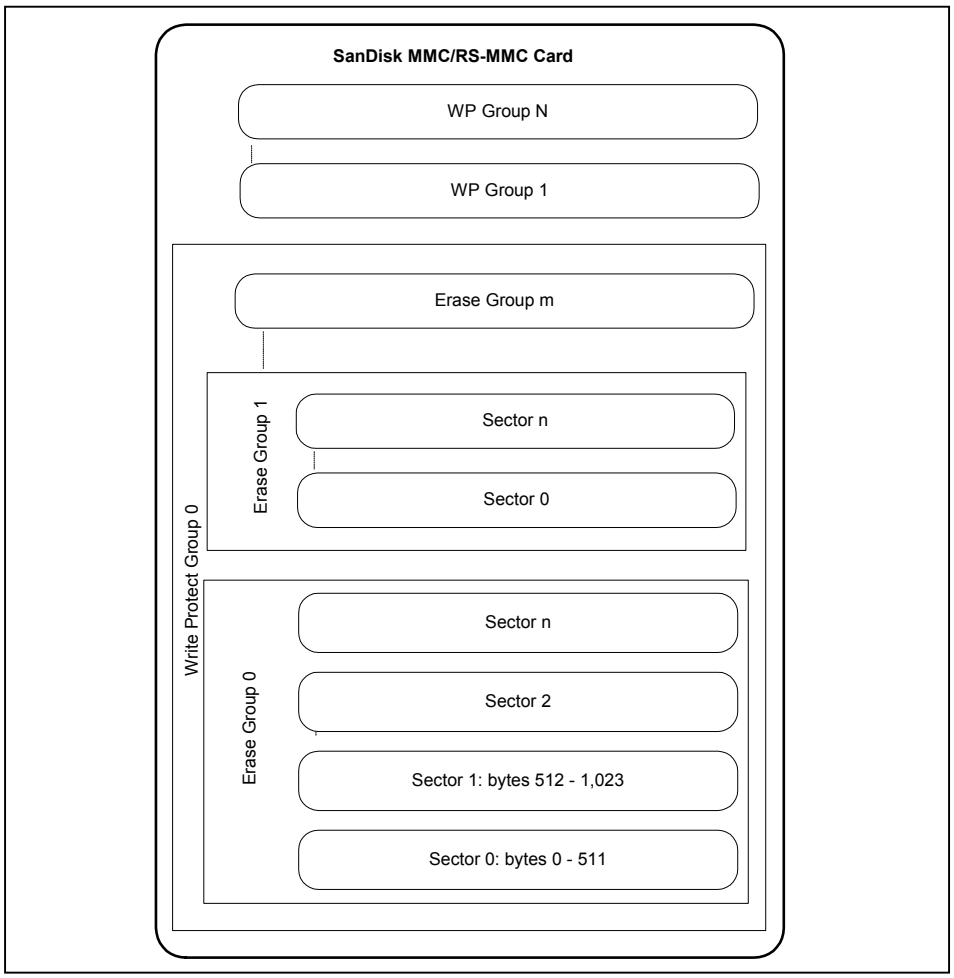


Table 1-1 Memory Array Structures Summary[1]

	Bytes	Sector	Erase Group Size	No. of Erase	WP Group	No. of Write
SDMJ-32 **SDMRJ-32**	32 MB	62,688	32	1,959	32	62
SDMJ-64 **SDMRJ-64**	64 MB	125,408	32	3,919	32	123
SDMJ-128 **SDMRJ-128**	128 MB	250,816	32	7,838	32	245
SDMJ-256 **SDMRJ-256**	256 MB	501,632	32	15,676	32	490
SDMJ-512 **SDMRJ-512**	512 MB	1,003,264	32	31,352	32	980
SDMJ-1024 **SDMRJ-1024**	1 GB	2,006,528	32	62,704	32	1,960

[1] All measurements are units-per-card.

1.12.6 Read and Write Operations

The MultiMediaCard/RS-MultiMediaCard supports two read/write modes as shown in Figure 1-3 and defined in Table 1-2.

Figure 1-3 Data Transfer Formats

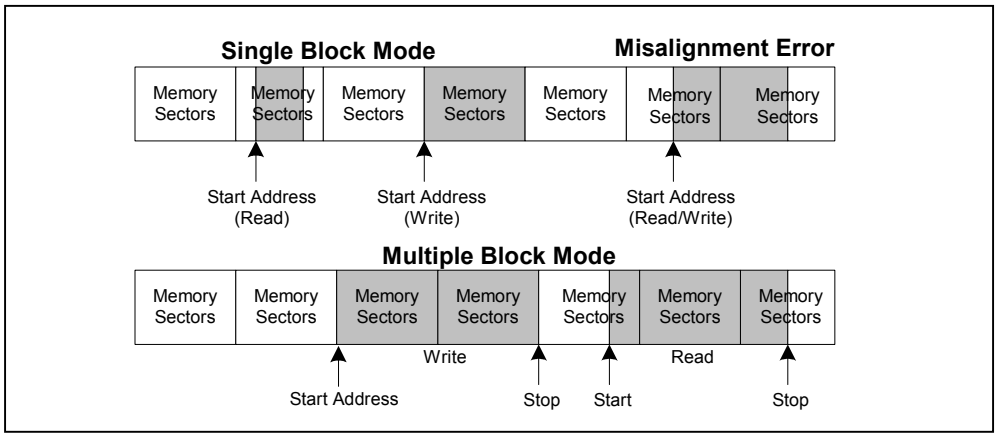


Table 1-2 Mode Definitions

Mode	Description
Single Block	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	This mode is similar to the single block mode, except for the host can read/write multiple data blocks (all have the same length) that are 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 and are identical to the single block read/write operations.

1.12.7 Data Protection in the Flash Card

Every sector is protected with an error correction code. 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.

1.12.8 Erase

The smallest erasable unit in the MultiMediaCard/RS-MultiMediaCard is a **sector**. In order to speed up the erase procedure, multiple sectors can be erased at the same time. The erase operation is divided into two stages as shown in Table 1-3.

Table 1-3 Erase Operation Stages

Stage	Name	Description
1	Tagging	**Selecting the Sectors for Erasing**. 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.
2	Erasing	**Starting the Erase Process**. The sectors are grouped into erase groups of 16 or 32 sectors. Tagging can address sectors or erase groups. Either an arbitrary set of sectors within a single erase group, or an arbitrary selection of erase groups may be erased at one time, but not both together. That is, the unit of measure for determining an erase is either a sector or an erase group. If sectors are tagged, then all selected sectors must lie within the same erase group. Tagging and erasing sectors must follow a strict command sequence.

1.12.9 Write Protection

Two-card level write-protection options are available: permanent and temporary. Both can be set using the PROGRAM_CSD command (refer to *CSD Programming*, Section 4.2.3). The permanent write protect bit, once set, cannot be cleared. This feature is implemented in the MultiMediaCard /RS-MultiMediaCard controller firmware and not with a physical OTP cell.

1.12.10 Copy Bit

MultiMediaCard/RS-MultiMediaCard content can be marked as an original or a copy using the copy bit. The copy bit of the card is programmed as a copy when testing and formatting are performed during manufacturing. When set, the copy bit in the CSD Register is a copy and cannot be cleared.

The card is available with the copy bit set or cleared. The bit set indicates that the card is a master. This feature is implemented in the card's controller firmware and not with a physical OTP cell.

1.12.11 CSD Register

All MultiMediaCard/RS-MultiMediaCard configuration information 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, the user file format indication and ECC Register.

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

1.13 SPI Mode

The SPI mode is a secondary communication protocol for the MultiMediaCard and RS-MultiMediaCard. This mode is a subset of the MultiMediaCard Protocol, designed to communicate with an SPI channel, commonly found in Motorola and other vendors' microcontrollers.

Table 1-4 SPI Mode

Function	Description
Negotiating Operating Conditions	The operating condition negotiation function of the MultiMediaCard/RS-MMC bus is not supported in SPI Mode. The host must work within the valid voltage range, 2.7 to 3.6 V, of the card.
Card Acquisition and Identification	This function is not supported in SPI Mode. The host must know the number of cards currently connected on the bus. Specific card selection is done using the CS signal.
Card Status	In SPI mode, only 16 bits containing errors relevant to SPI mode can be read out of the 32-bit Status Register.
Memory Array Partitioning	Memory partitioning in SPI mode is equivalent to MultiMediaCard mode. All read and write commands are byte addressable.
Read/Write Operations	In SPI mode, single and multiple block data transfers are supported. Stream mode is not supported.
Data Transfer Rate	Same as MultiMediaCard mode.
Data Protection in MultiMediaCard/RS-MultiMediaCard	Same as MultiMediaCard mode.
Erase	Same as MultiMediaCard mode.
Write Protection	Same as MultiMediaCard mode.

2 Product Specifications

2.1 Overview

In this section, all values are defined at an ambient temperature and nominal supply voltage unless otherwise stated.

2.2 System Environmental Specifications

Table 2-1 defines the environmental specifications for the SanDisk MultiMediaCard and RS-MultiMediaCard.

Table 2-1 Environmental Specification Summary

Temperature	Operating	-25° C to 85° C
	Non-operating	-40° C to 85° C
Humidity	Operating	8% to 95%, non condensing
	Non-operating	8% to 95%, non condensing
Acoustic Noise		0 dB
ESD Protection	Contact Pads	+/- 4kV, Human body model according to ANSI EOS/ESD-S5.1-1998
	Non Contact Pad Area	+/- 8kV (coupling plane discharge) +/- 15kV (air discharge) Human body model per IEC61000-4-2.
Vibration	Operating	15 G peak-to-peak max.
	Non-operating	15 G peak-to-peak max.
Shock	Operating	1,000 G max.
	Non-operating	1,000 G max.
Altitude[1]	Operating	80,000 ft. max.
	Non-operating	80,000 ft. max.

[1] Relative to sea level.

2.3 System Power Requirements

All values quoted in Table 2-2 are at *room temperature* unless otherwise stated.

Table 2-2 System Power Requirements

Operation	Max. Power Dissipation @3.6 V
Read	50 mA
Write	60 mA
Sleep	150 uA

2.4 System Performance

All performance values for the MultiMediaCard and RS-MultiMediaCard in Table 2-3 are under the following conditions:

- Voltage range 2.7 V to 3.6 V
- Temperature -25° C to 85° C
- Independent of the MMC clock frequency

Table 2-3 Performance

Timing	Typical	Maximum
Block Read Access Time	0.5 ms	100 ms
Block Write Access Time	0.5 ms	240 ms
CMD1 to Ready after Power-up	150 ms	500 ms
Sleep to Ready	1 ms	2 ms

2.5 System Reliability and Maintenance

Table 2-4 Reliability and Maintenance Summary

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

2.6 Physical Specifications

Refer to Table 2-5 and Figure 2-1 for MultiMediaCard, and Figure 2-2, Table 2-6 for RS-MultiMediaCard physical specifications and dimensions.

Table 2-5 MultiMediaCard Physical Specification Summary

Specification	MultiMediaCard
Weight	1.8 g maximum
Length	32 mm ± 0.1 mm
Width	24 mm ± 0.08 mm
Thickness	1.4 mm ± 0.1 mm

Figure 2-1 Full-size MultiMediaCard Dimensions

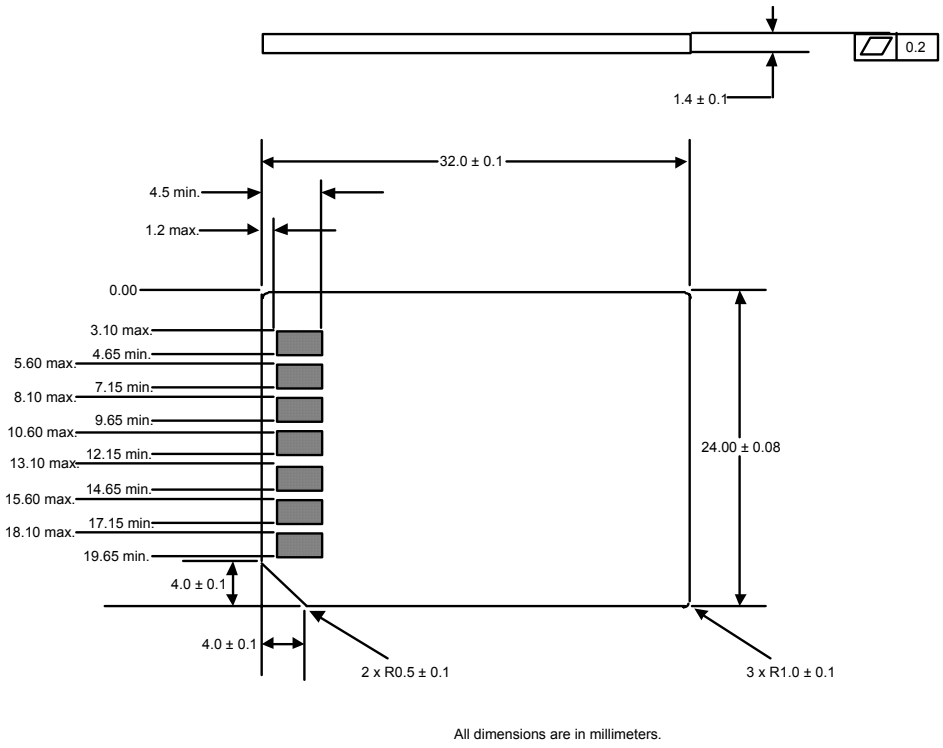
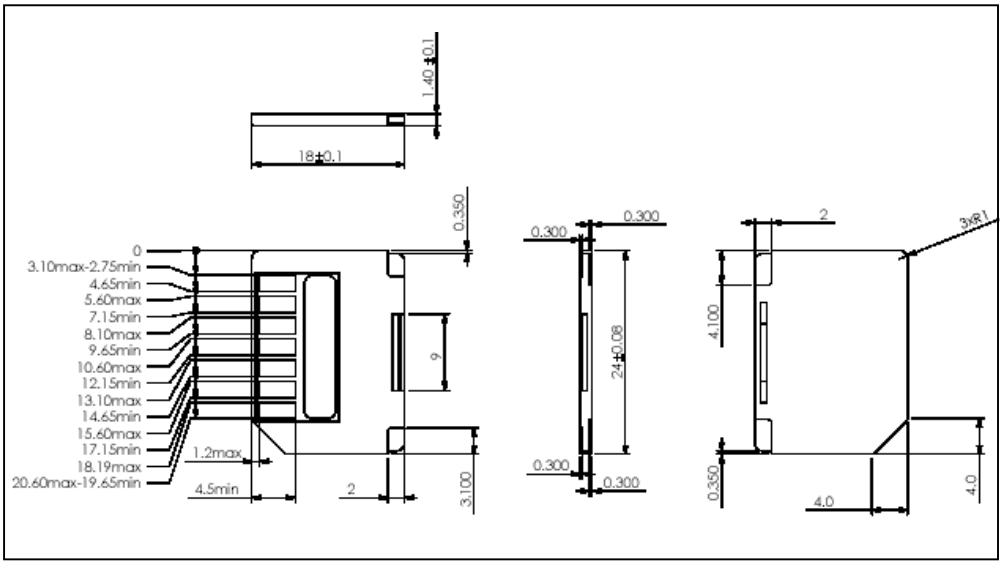


Table 2-6 RS-MultiMediaCard Physical Specification Summary

Specification	RS-MMC
Weight	1.0 g maximum
Length	18 mm ± 0.1 mm
Width	24 mm ± 0.08 mm
Thickness	1.4 mm ± 0.1 mm

Figure 2-2 Reduced-size MultiMediaCard Dimensions



2.7 Capacity Specifications

Table 2-7 shows the specific capacity for the various models.

Table 2-7 Model Capacity Summary

Model No.	Form Factor	Capacity
SDMJ-32	Full-size MultiMediaCard	32 MB
SDMJ-64	Full-size MultiMediaCard	64 MB
SDMJ-128	Full-size MultiMediaCard	128 MB
SDMJ-256	Full-size MultiMediaCard	256 MB
SDMJ-512	Full-size MultiMediaCard	512 MB
SDMJ-1024	Full-size MultiMediaCard	1 GB
SDMRJ-32	Reduced-size MultiMediaCard	32 MB
SDMRJ-64	Reduced-size MultiMediaCard	64 MB
SDMRJ-128	Reduced-size MultiMediaCard	128 MB
SDMRJ-256	Reduced-size MultiMediaCard	256 MB
SDMRJ-512	Reduced-size MultiMediaCard	512 MB
SDMRJ-1024	Reduced-size MultiMediaCard	1 GB

3 Interface Description

3.1 Physical Description

The MultiMediaCard and RS-MultiMediaCard has seven exposed contacts on one side. The host is connected to the card using a dedicated seven-pin connector.

3.1.1 Pin Assignments

Table 3-1 MultiMediaCard and RS-MultiMediaCard Pad Assignment

Pin No.	Name	Type[1]	Description
MultiMediaCard Mode			
1	RSV	NC	Not connected or Always “1”
2	CMD	I/O, PP, OD	Command/Response
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DAT0	I/O, PP	Data 0
SPI Mode			
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

[1] Type Key: S=power supply; I=input; O=output using push-pull drivers; PP=I/O using push-pull drivers

3.2 MultiMediaCard/RS-MultiMediaCard Bus Topology

The MultiMediaCard/RS-MultiMediaCard bus has three communication lines and four supply lines.

- CMD
- DAT
- CLK
- VDD
- VSS[1:2]

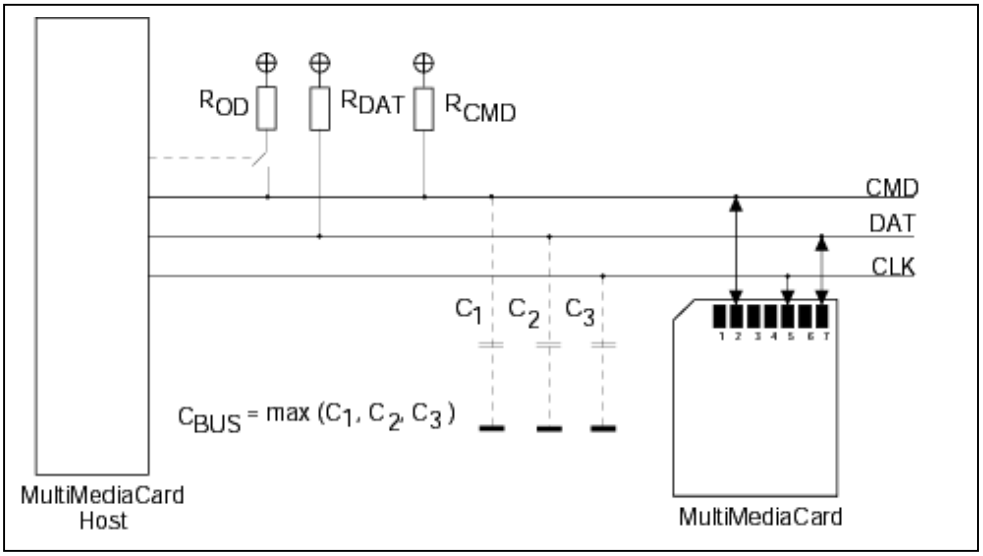
The description of each signal is contained in Table 3-2.

Table 3-2 Bus Signal Descriptions

Name	Description
CMD	Command is a bi-directional signal. Host and card drivers are operating in two modes: open drain and push-pull.
DAT	Data line is a bi-directional signal. Host and card drivers are operating in push-pull mode.
CLK	Clock is a host to card signal. CLK operates in push-pull mode.
VDD	VDD is the power supply line for all cards.
VSS [1:2]	VSS are two ground lines.

Figure 3-1 shows the bus circuitry with one host in MultiMediaCard mode.

Figure 3-1 Bus Circuitry Diagram



The R_{OD} is switched on and off by the host synchronously to the open-drain and push-pull mode transitions. R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and DAT line against bus floating when no card is inserted or all card drivers are in a hi-impedance mode.

A constant current source can replace the R_{OD} in order to achieve better performance (constant slopes for the signal rising and falling edges). If the host does not allow the switchable R_{OD} implementation, a fixed R_{CMD} can be used. Consequently the maximum operating frequency in the open-drain mode has to be reduced in this case.

3.2.1 Hot Insertion and Removal

Hot insertion and removal are allowed; inserting or removing the card into or from the MultiMediaCard bus will not damage the card. This also applies when the power is up.

Data transfer operations are protected by CRC codes; therefore, the bus master can detect any bit changes induced by hot insertion and removal. The inserted card will be properly reset when CLK carries a clock frequency (f_{pp}).

3.2.2 Power Protection

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

If the hot insertion feature is implemented in the host, the host must withstand a shortcut between V_{DD} and V_{SS} without damage.

3.3 SPI Bus Topology

The MultiMediaCard/RS-MultiMediaCard SPI Interface is compatible with SPI hosts available on the market. Similar to any other SPI device, the card's 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

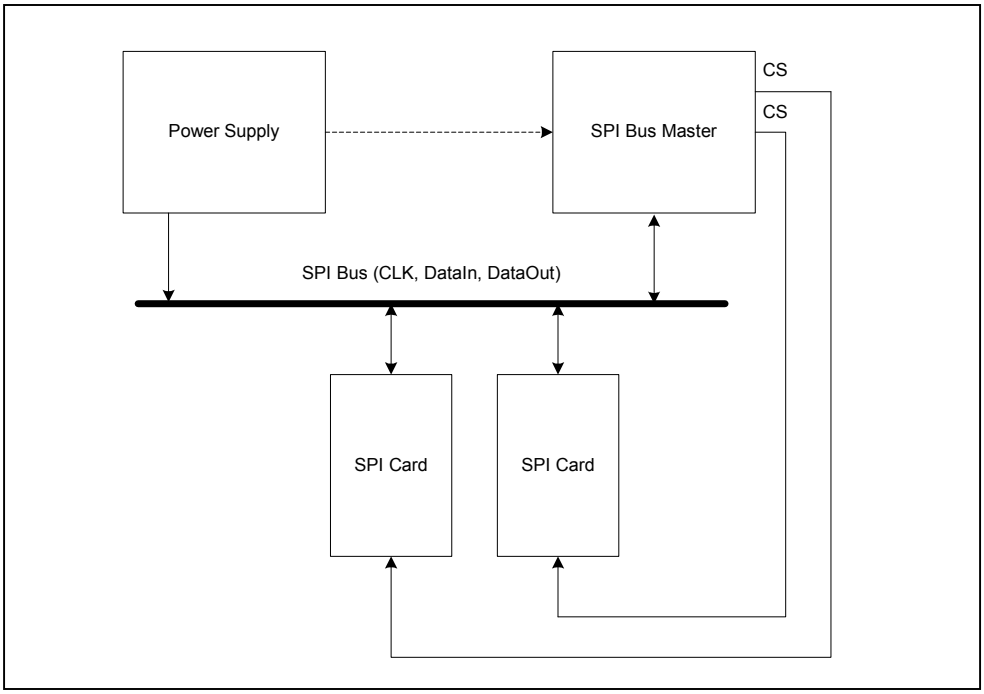
Another SPI common characteristic implemented in the MultiMediaCard and RS-MultiMediaCard are byte transfers. All data tokens are multiples of 8-bit bytes and always byte-aligned to the CS signal. The SPI standard defines the physical link only and not the complete data transfer protocol. In SPI Bus mode, the card uses a subset of the MultiMediaCard Protocol and command set.

The card identification and addressing algorithms are replaced by the hardware CS signal. There are no broadcast commands. A card (slave) is selected for every command by asserting the CS signal (active low). Refer to Figure 3-2.

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 unidirectional dataIn and dataOut signals. This eliminates the ability to execute commands while data is being read or written which prevents sequential multi read/write operations. Only single block read/write is supported by the SPI channel.

Figure 3-2 MultiMediaCard and RS-MultiMediaCard Bus System



3.3.1 Power Protection

Power protection is the same as it is in MultiMediaCard mode.

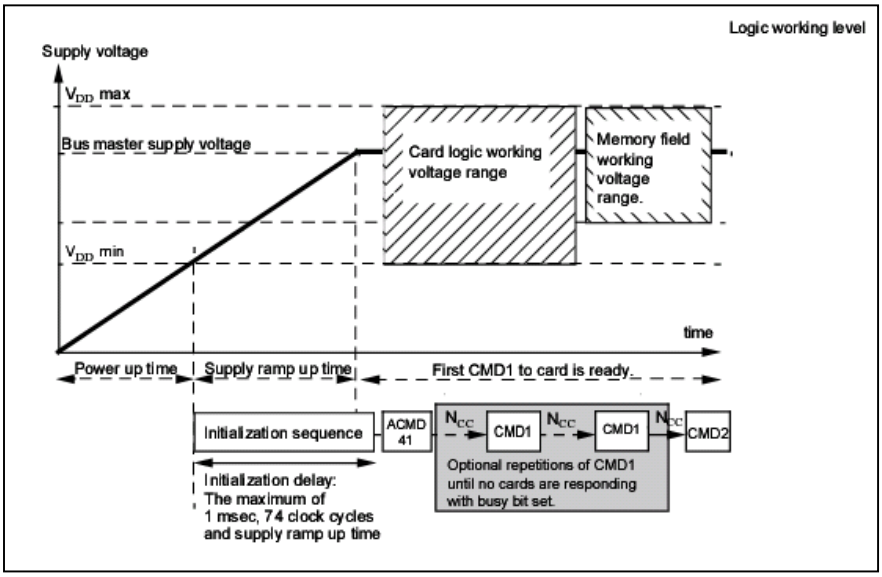
3.4 Electrical Interface

The following sections provide valuable information about the electrical interface.

3.4.1 Power Up

The power-up of the MultiMediaCard/RS-MultiMediaCard bus is handled locally in each card and the bus master. See Figure 3-3.

Figure 3-3 Power-up Diagram



After power up, including hot insertion[2] the card enters an idle state and ignores all bus transactions until CMD1 is received.

CMD1 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 CMD1 contains a busy flag, indicating that the card is still working on its power-up procedure and not ready for identification. This bit informs the host that the card is not ready. The host must wait, while continuing to poll the cards, until this bit is cleared. The MultiMediaCard/RS-MultiMediaCard initialization sequence will be completed within 500 ms.

The bus master has the task of getting the individual cards and the entire card system out of idle state. Because the power-up and the supply ramp-up time depend on application parameters such as the maximum number of cards, the bus length and 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 CMD1) before CMD1 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 1 ms, 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) are provided to eliminate power-up synchronization problems.

[2]Inserting a card when the bus is operating

3.4.2 Bus Operating Conditions

SPI Mode Bus operating conditions are identical to those in MultiMediaCard mode. Table 3-3 lists the power supply voltages. The CS signal timing is identical to the input signal timing (see Figure 3-5).

Table 3-3 Bus Operating Conditions Summary

Parameter	Symbol	Min	Max	Unit	Remark
General					
Peak voltage on all lines	---	-0.5	3.6	V	
All Inputs					
Input Leakage Current	---	-10	10	uA	
All Outputs					
Output Leakage Current	---	-10	10	uA	
Power Supply Voltage[3]					
Supply Voltage	V_{DD}	2.7	3.6	V	
Supply voltage differentials (V_{SS1}, V_{SS2})	---	-0.5	0.5	V	
Capacitance					
V_{DD} Capacitance	C (V_{DD})	---	3.0	uF	

3.4.3 Bus Signal Line Load

The total capacitance CL of each line in the MultiMediaCard 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 3-4 must not be exceeded.

Table 3-4 Host and Bus Capacities

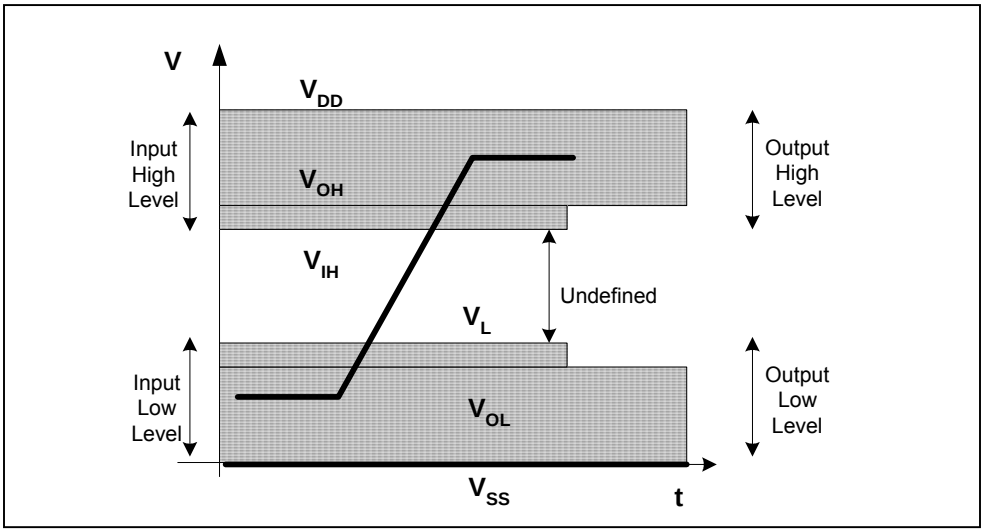
Parameter	Symbol	Min.	Max.	Unit	Remark
Pull-up resistance	R_{CMD}, R_{DAT}	50	100	kΩ	Prevents bus floating
Bus signal line capacitance	C_L	---	250	pF	$f_{PP} \leq$ 5 MHz, 30 cards
Bus signal line capacitance	C_L	---	100	pF	$f_{PP} \leq$ 20 MHz, 10 cards
Signal card capacitance	C_{CARD}	---	7	pF	---
Max. signal line inductance	---	---	16	nH	$f_{PP} \leq$20 MHz

3.4.4 Bus Signal Levels

Because the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage (see Figure 3-4).

[3] The current consumption of any card during the power-up procedure must not exceed 10 mA.

Figure 3-4 Bus Signal Levels



3.4.5 Open-drain Mode Bus Signal Level

The input levels shown in Table 3-5 are identical to the push-pull mode bus signal levels.

Table 3-5 Open Drain Mode Bus Signal Levels

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output high voltage	VOH	$V_{DD-0.2}$	---	V	IOH = -100 uA
Output low voltage	VOL	---	0.3	V	IOL = 2 mA

3.4.6 Push-pull Mode Bus Signal Level

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

Table 3-6 Push-pull Mode Bus Signal Level

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output high voltage	VOH	0.75 * V_{DD}	---	V	IOH = -100 uA @ V_{DD} (min.)
Output low voltage	VOL	---	0.125 * V_{DD}	V	IOL = 100 uA @ V_{DD} (min.)
Input high voltage	VIH	0.625 * V_{DD}	V_{DD} + 0.3	V	
Input low voltage	VIL	V_{SS}-0.3	0.25 * V_{DD}	V	---

3.4.7 Bus Timing

SanDisk's MultiMediaCards and RS- MultiMediaCards clock data in on the rising edge and out on the falling edge. The data contained in the shaded areas is not valid in Figure 3-5.

Figure 3-5 Data In/Out Referenced to Clock Timing

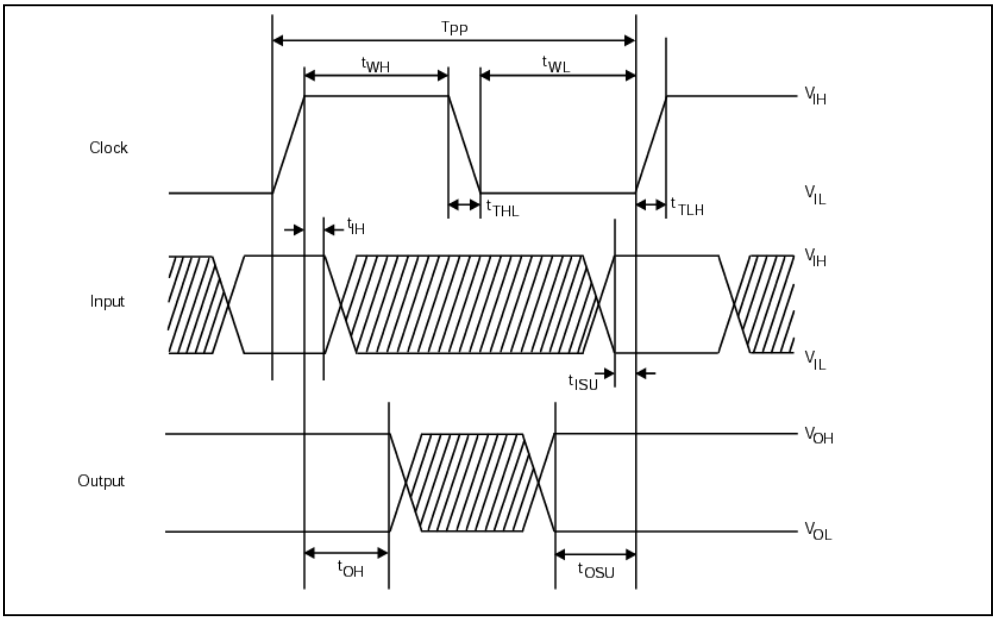


Table 3-7 Bus Timing

Parameter	Symbol	Min	Max	Unit	Remark
Clock (CLK) – all values referred to min. V_{IH} and max. V_{IL}					
Clock Freq. Data Transfer Mode	f_{PP}	0	20	MHz	$C_L \leq$ 100 pF (10 cards)
Clock Freq. Identification Mode	f_{OD}	0	400	kHz	$C_L \leq$ 250 pF (30 cards)
Clock Low Time	t_{WL}	10	---	ns	$C_L \leq$ 100 pF (10 cards)
Clock High Time	t_{WH}	10	---	ns	$C_L \leq$ 100 pF (10 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 (10 cards)
Clock Low Time	t_{WL}	50	---	ns	$C_L \leq$ 250 pF (30 cards)
Clock High Time	t_{WH}	50	---	ns	$C_L \leq$ 250 pF (30 cards)
Clock Rise Time	t_{TLH}	---	50	ns	$C_L \leq$ 250 pF (30 cards)
Clock Fall Time	t_{THL}	---	50	ns	$C_L \leq$ 250 pF (30 cards)
Inputs CMD, DAT – referenced to CLK					
Input setup time	t_{ISU}	3	---	ns	
Input hold time	t_{IH}	3	---	ns	
Outputs CMD, DAT – referenced to CLK					
Output setup time	t_{OSU}	5	---	ns	
Output hold time	t_{ODLY}	5	---	ns	

3.5 MultiMediaCard/RS-MultiMediaCard Registers

There is a set of six registers within the card interface. The OCR, CID, and CSD registers carry the card configuration information. The RCA register holds the card relative communication address for the current session.

3.5.1 Operating Conditions Register

The 32-bit **Operation Conditions Register (OCR)** stores the V_{DD} voltage profile of the MultiMediaCard/RS-MultiMediaCard. In addition, this register includes a status information bit. This status bit is set if the card power-up procedure has been finished. All cards will implement the OCR Register.

The supported voltage range is coded as shown in Table 3-8, for high voltage and low voltage MultiMediaCards and RS-MultiMediaCards. As long as the card is busy, the corresponding bit (31) is set to low, the 'wired-and' operation yields low if at least one card is still busy.

Table 3-8 Operating Conditions Register

OCR Bit	VDD Voltage Window	High Voltage MultiMediaCard
6:0	Reserved	0000000b
7	1.65 - 1.95	0b
14:8	2.0 - 2.6	0000000b
23:15	2.7 - 3.6	111111111b
30:24	Reserved	0000000b
31	Card power-up status bit (busy)	

3.5.2 Card Identification Register

The **Card Identification Register (CID)** 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 MultiMediaCard/RS-MultiMediaCard hosts.

Table 3-9 CID Register Fields

Name	Type	Width	CID-Slice	CID Value	Comments
Manufacturer ID (MID)	Binary	8	[127:120]	0x02	Manufacturer IDs are controlled and assigned by the MMC Association

Name	Type	Width	CID-Slice	CID Value	Comments
OEM/Application ID (OID)	Binary	16	[119:104]	0x0000	Identifies the card OEM and/or the card contents. The OID is assigned by the MMCA. This field may be specifically configured for OEM customers
Product Name (PNM)	String	48	[103:56]	SDMJ-32 SDM032 SDMJ-64 SDM064 SDMJ-128 SDM128 SDMJ-256 SDM256 SDMJ-512 SDM512 SDMJ-1024 SDM01G SDMRJ-32 SDR032 SDMRJ-64 SDR064 SDMRJ-128 SDR128 SDMRJ-256 SDR256 SDMRJ-512 SDR512 SDMRJ-1024 SDR01Gp	Six ASCII characters long
Product Revision[4]	BCD	8	[55:48]	---	Two binary-coded
Serial Number (PSN)	Binary	32	[47:16]	---	32-bit unsigned integer
Manufacturing Date Code (MDT)	BCD	8	[15:8]	Manufacture date (for ex. March 2001= 00110100	Manufacturing date—mm/yy (offset from 1997)
CRC7 checksum (CRC)	Binary	7	[7:1]	CRC7*	Calculated
Not used, always “1”	---	1	[0.0]	---	---

*The CRC checksum is computed by using the following formula:

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

$M(x)=(MID\text{-}MSB)*x^{119}+\ldots+(CIN\text{-}LSB)*x^{0}$

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

3.5.3 Card Specific Data Register

The **Card Specific Data (CSD) Register** configuration information is required to access the card data.

[4] The product revision is composed of two binary-coded decimal (BCD) digits (4 bits ea.) representing and “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) would be “01100010”.

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). The values are presented in "real world" units for each field and coded according to the CSD structure.

Table 3-10 CSD Register Fields

Field	Width	Cell Type	CSD Slice	CSD Value	CSD Code	Description
CSD_STRUCTURE	2	R	[127:126]	v1.2	2	CSD structure
SPEC_VERS	4	R	[125:122]	v3.3	3	MMC Spec.
---	2	R	[121:120]	0	0	Reserved
TAAC	8	R	[119:112]	10 ms	0x0F	Data read access time
NSAC	8	R	[111:104]	0	0	Data read access time-2 in CLK cycles (NSAC*100)
TRAN_SPEED	8	R	[103:96]	20 MHz	00x2A	Max. data transfer rate
CCC	12	R	[95:84]	Not Supported[5]	0x0F5	Card command classes
READ_BL_LEN	4	R	[83:80]	512	9	Max. read data block length
READ_BL_PARTIAL	1	R	[79:79]	Yes	1	Partial blocks for read allowed
WRITE_BLK_MISALIGN	1	R	[78:78]	No	0	Write block misalignment
READ_BLK_MISALIGN	1	R	[77:77]	No	0	Read block misalignment
DSR_IMP	1	R	[76:76]	No	0	DSR implemented
---	2	R	[75:74]	0	0	Reserved
C_SIZE	12	R	[73:62]			Device size (C_SIZE)
VDD_R_CURR_MIN	"	"	"	35 mA	5	**RS-MMC** max. read current @VDD min. for 16MB to 1GB cards only.
VDD_R_CURR_MAX	"	"	"	45 mA	5	**RS-MMC** max. read current @VDD min for 16MB to 1GB cards only.
VDD_W_CURR_MIN				60 mA	6	**RS-MMC** max. write current @VDD min. for 16MB to1GB cards only.
VDD_W_CURR_MAX				45 mA	5	**RS-MMC** max. read current @VDD min. for 16MB to 128MB

[5] CSD Value does not support I/O, application-specific, stream write and stream read.

Field	Width	Cell Type	CSD Slice	CSD Value	CSD Code	Description
						cards only.
VDD_W_CURR_MAX				80 mA	6	**RS-MMC** max. read current @VDD min. for 256MB to 1GB cards only.
C_SIZE_MULT	3	R	[49:47]			Device size multiplier (C_SIZE_MULT)
ERASE_GRP_SIZE	5	R	[46:42]	32	0x1F	Erase group size
ERASE_GRP_MULT	5	R	[41:37]	---	“0” for cards <= 256MB “3” for cards > 256MB	Erase group size multiplier
WP_GRP_SIZE	5	R	[36:32]	32	0x1F	Write protect group size
WP_GRP_ENABLE	1	R	[31:31]	Yes	1	Write protect group enable
			[30:29]			
R2W_FACTOR	3	R	[28:26]	1:4	2	Read to write speed factor
WRITE_BL_LEN	4	R	[25:22]	512	9	Max. write data block length
WRITE_BL_PARTIAL	1	R	[21:21]	No	0	Partial blocks for write allowed
---	4	R	[20:17]	0	0	Reserved
CONTENT_PROT_APP	1	R	[16:16]	---	---	Content protection application
FILE_FORMAT_GRP	1	R/W	[15:15]	0	0	Indicates file format of selected group
COPY	1	R/W	[14:14]	Copy	1	Copy flag (OTP)
PERM_WRITE_PROTECT	1	R/W	[13:13]	No	0	Permanent write protection
TMP_WRITE_PROTECT	1	R/W/E	[12:12]	No	0	Temporary write protection
FILE_FORMAT	2	R/W	[11:10]	0	0	File format of card
ECC	2	R/W/E	[9:8]	None	0	ECC code
CRC	7	R/W/E	[7:1]	---	---	CRC
---	1	---	[0:0]	1	1	Not used. Always “1”

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 System Specification Version
0	CSD Version 1.0	v1.0 to 1.2
1	CSD Version 1.1	v1.4 to 2.2
2	CSD Version 1.2	v3.1 to 3.3
3	Reserved	Reserved

- **SPEC_VERSION—**Defines the MultiMediaCard System Specification version supported by the card.

Table 3-12 System Specification Version

SPEC_VERSION	System Specification Version Number
0	v1.0 to 1.2
1	v1.4
2	v2.0 to 2.2
3	v3.1 to 3.3
4 - 15	Reserved

- **TAAC**—defines the asynchronous part (relative to the card clock (CLK)) of the read access time.

Table 3-13 TAAC Access Time Definition

TAAC Bit Position	Code
2:0	Time exponent 0=1 ns, 1=10 ns, 2=100 ns, 3=1 ums, 4=10 ums, 5=100 ums, 6=1 ms, 7=10 ms
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} 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 3-14 defines the maximum data transfer rate TRAN_SPEED.

Table 3-14 Max. Data Transfer Rate Definition

TRAN_SPEED Bit	Code
2:0	Transfer rate exponent 0=100 kb/s, 1=1 Mb/s, 2=10 Mb/s, 3=100 Mb/s, 4…7=reserved
6:3	Time mantissa 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 MultiMediaCard 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 3-15 Supported Card Command Classes

CCC Bit	Supported Card Command Class
0	Class 0
1	Class 1

11	Class 11

- **READ_BL_LEN**—The read data block length is computed as $2^{READ\_BL\_LEN}$. The maximum block length might therefore be in the range 1, 2, 4…2048 bytes.

Table 3-16 Data Block Length

READ_BL_LEN	Block Length
0	2^0 = 1 byte
1	2^1 = 2 bytes
.....	
11	2^{11} = 2048 bytes
12-15	Reserved

- **READ_BL_PARTIAL**—defines whether partial block sizes can be used in block read commands.

Table 3-17 Bit Definition

READ_BL_PARTIAL	Definition
0	Only the READ_BL_LEN block size can be used for block-oriented data transfers.
1	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.

Table 3-18 Bit Definition

WRITE_BLK_MISALIGN	Definition
0	Signals that crossing physical block boundaries is invalid.
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.

Table 3-19 Bit Definition

READ_BLK_MISALIGN	Definition
0	Signals that crossing physical block boundaries is invalid.
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 3-20 DSR Implementation Code Table

DSR_IMP	DSR Type
0	No DSR implemented
1	DSR Implemented

- **C_SIZE (Device Size)**—computes the card capacity. 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 = $^{2C\_SIZE\_MULT+2}$ (C_SIZE_MULT < 8)

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

Therefore, the maximum capacity that can be coded is 4096*512*2048=4 GB. For example, 4-MB 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**—minimum values for read and write currents at the VDD power supply are coded in Table 3-21.

Table 3-21 V_{DD} Minimum Current Consumption

VDD_R_CURR MIN VDD_W_CURR MIN	Code for Current Consumption @ V_{DD}
2:0	0=0.5 mA, 1=1 mA, 2=5 mA, 3=10 mA, 4=25 mA, 5=35 mA, 6=60 mA, 7=100 mA

- **VDD_R_CURR_MAX, VDD_W_CURR_MAX**—maximum values for read and write currents on VDD power supply are coded Table 3-22.

Table 3-22 V_{DD} Maximum Current Consumption

VDD_R_CURR MAX VDD_W_CURR MAX	Code for Current Consumption @ V_{DD}
2:0	0=1 mA, 1=5 mA, 2=10 mA, 3=25 mA, 4=35 mA, 5=45 mA, 6=80 mA, 7=200 mA

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

Table 3-23 Device Size Multiplying Factor

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_GRP_SIZE**—contents of this register is a 5-bit binary coded value, used to calculate the size of the erasable unit of the card. The size of the erase unit (also referred to as erase group) is determined by the ERASE_GRP_SIZE and the ERASE_GRP_MULT entries of the CSD, using the following equation:

  Size of erasable unit = (ERASE_GRP_SIZE + 1) * (ERASE_GRP_MULT + 1)

This size is given as the minimum number of write blocks that can be erased in a single erase command.

- **ERASE_GRP_MULT**—5-bit binary coded value used for calculating the size of the erasable unit of the card. See ERASE_GRP_SIZE section for detailed description.
- **WP_GRP_SIZE**—write protected group size. The content of this register is a 5-bit binary coded value, defining the number of Erase Groups (see ERASE_GRP_SIZE). The actual size is computed by increasing this number by one. A value of 0 means 1 erase group, 127 means 128 erase groups.
- **WP_GRP_ENABLE**—A value of "0" means group write protection is not possible.
- **R2W_FACTOR**—defines the typical block program time as a multiple of the read access time. Table 3-24 defines the field format.

Table 3-24 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	64
7	128

- **WRITE_BL_LEN**—block length for write operations. See READ_BL_LEN for field coding.
- **WRITE_BL_PARTIAL**—defines whether partial block sizes can be used in block write commands.

Table 3-25

WRITE_BL_PARTIAL	Definition
0	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.
1	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.
- **COPY**—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 does not change card characteristics.
- **PERM_WRITE_PROTECT**—permanently protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled). The default value is 0 (i.e., not permanently write protected).
- **TMP_WRITE_PROTECT**—temporarily protects the whole card content from being overwritten or erased (all write and erase commands for this card are temporarily disabled). This bit can be set and reset. The default value is 0 (i.e., not write protected).

- **CONTENT_PROT_APP—**indicates whether the content protection application is supported. MultiMediaCards that implement the content protection application will have this bit set to 1.
- **FILE_FORMAT**—indicates the file format on the card. This field is read-only for ROM. The formats are defined in Table 3-26.

Table 3-26 File Format

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

- **ECC**—defines the ECC code that was used for storing data on the card. This field is used to decode user data by the host (or application). Table 3-27 defines the field format.

Table 3-27 ECC Type

ECC	ECC Type	Max. Number of Correctable Bits per Block
0	none (default)	none
1	BCH (542,512)	3
2 - 3	Reserved	---

- **CRC**—carries the checksum for the CSD contents. The host must recalculate the checksum for any CSD modification. The default corresponds to the initial CSD contents.

3.5.4 Status Register

The MMC/RS-MMC Status Register structure is 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 Status Register Description

Bit	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	E R	0= no error 1= error	Command argument was out of the allowed range for the card	C
30	ADDRESS_ERROR	E R X	0= no error 1= error	Misaligned address that didn't match the block length was used in the command	C
29	BLOCK_LEN_ERROR	E R	0= no error 1= error	Transferred block length is not allowed for the 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	Error in the sequence of erase commands occurred	C
27	ERASE_PARAM	E X	0= no error 1= no error	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
25-24	Not applicable; bit always set to 0				
23	COM_CRC_ERROR	E R	0= no error 1= error	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
21-20	Not applicable; bit always set to 0				
19	ERROR	E R X	0= no error 1= error	General or unknown error occurred during the operation	C
17	Not applicable. Bit always set to 0.				
16	CID/CSD_OVERWRITE	E R X	0= no error 1= error	Can be any of the following errors: - CID Register has been written and cannot be over written. - CSD read-only section does not match the card content. (con't next page)	C

Bit	Identifier	Type	Value	Description	Clear Condition
16 con't	CID/CSD_OVERWRITE (con't)	(con't)	(con't)	- Attempt made to reverse copy (set as original) or permanent WP (unprotected) bits.	(con't)
15	WP_ERASE_SKIP	S X	0 = not protected 1 = protected	Only partial address space was erased due to existing write protected blocks.	C
14	CARD_ECC_DISABLED	S X	0 = enabled 1 = disabled	Command executed without using the internal ECC	A
13	ERASE_RESET	S R	0 = cleared 1 = set	Erase sequence was cleared before command execution 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	Card state when receiving the command. If the command execution causes a state change, to the host it will be visible 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 signaling on the bus (RDY/BSY).	A
7-6	Reserved				
4-0	Reserved				

3.5.5 Relative Card Address Register

The 16-bit Relative Card Address (RCA) 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.

3.5.6 MultiMediaCard Registers in SPI Mode

In SPI mode, only the MultiMediaCard CSD and CID registers are accessible. Their format is identical to the format in the MultiMediaCard 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 File System Format

SanDisk MultiMediaCards and RS-MultiMediaCards are formatted with a “hard disk-like” partitioned DOS FAT file system.

Similar to hard disks in PCs, the first data block of the memory consists of a partition table. Thus, using the same notation as for hard disks, i.e., partitioning the memory field into logical sectors of 512 bytes each, the first sector is reserved for this partition table. Table 3-29 shows how the data in this sector is structured.

Table 3-29

Byte Position	Length (bytes)	Entry Description	Value/Range
0x0	446	Consistency check routine	---
0x1be	16	Partition Table entry	See Table 3-30
0xce	16	Partition Table entry	See Table 3-30
0x1de	16	Partition Table entry	See Table 3-30
0x1ee	16	Partition Table entry	See Table 3-30
0x1fe	1	Signature	0x55
0x1ff	1	Signature	0xaa

Table 3-30 Partition Entry Description

Byte Position	Length (bytes)	Entry Description	Value/Range
0x0	1	Boot descriptor	0x00 (Non-bootable device) 0x80 (Bootable device)
0x1	3	First partition sector	Address of First Sector
0x4	1	File system descriptor	0 = Empty 1 = DOS 12-bit FAT < 16 MB 4 = DOS 16-bit FAT < 32 MB 5 = Extended DOS 6 = DOS 16-bit FAT >= 32 MB 0x10-0xff = Free for other File Systems*

Byte Position	Length (bytes)	Entry Description	Value/Range
0x5	3	Last partition sector	Address of Last Sector
0x8	4	First sector position relative to beginning of device	Number of First Sector (Linear Address)
0xc	4	Number of sectors in partition	Between one and max. amount of sectors on device

E*Not used in DOS-based systems.

v

Every DOS partition is based on a 12-bit, 16-bit FAT or VFAT, respectively. All sector numbers are stored in Little-Endian format (least significant byte first). The start and end addresses of the partition are given in terms of heads, tracks and sectors, and can therefore be ignored for the MultiMediaCard, since the position of the partition can be determined by the last two entries.

The boot sector is described in Table 3-31.

Table 3-31 Boot Sector Configuration

Byte Position	Length (bytes)	Entry Description	Value/Range
0x0	3	Jump command	0xeb 0xXX 0x90
0x3	8	OEM name	XXX
0xb	2	Bytes/sector	512
0xd	1	Sectors/cluster	XXX (range: 1-64)
0xe	2	Reserved Sectors (Number of reserved sectors at the beginning of the media including the boot sector.)	1
0x10	1	Number of FATs	2
0x11	2	Number of root directory entries	512
0x13	2	Number of sectors on media	XXX (Depends on card capacity, if the media has more than 65535 sectors, this field is zero and the ‘number of total sectors’ is set.)
0x15	1	Media descriptor	0xf8 (hard disk)
0x16	2	Sectors/FAT	XXX
0x18	2	Sectors/track	32 (no meaning)

Byte Position	Length (bytes)	Entry Description	Value/Range
0x1a	2	Number of heads	2 (no meaning)
0x1c	4	Number of hidden sectors	0
0x20	4	Number of total sectors	XXX (depends on capacity)
0x24	1	Drive number	0x80
0x25	1	Reserved	0
0x26	1	Extended boot signature	0x29
0x27	4	Volume ID or serial number	XXX
0x2b	11	Volume label	XXX (ASCII characters padded with blanks if less than 11 characters.)
0x36	8	File system type	XXX (ASCII characters identifying the file system type FAT12 or FAT16.)
0x3e	448	Load program code	XXX
0x1fe	1	Signature	0x55
0x1ff	1	Signature	0xaa

All ‘X’ entries are denoting card dependent or non-fixed values. The number of sectors per track and the number of heads are meaningless for the MultiMediaCard and can be ignored.

4 MultiMediaCard Protocol Description

The host (master) controls all communication between the host and MultiMediaCard/RS-MultiMediaCard. The host sends the following two types of commands:

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

A general overview of the command flow is shown in Figure 4-1 for the Card Identification Mode and Figure 4-2 for the Data Transfer Mode. The commands are listed in Table 4-5 to 4-7. The dependencies between the current MultiMediaCard/RS-MultiMediaCard state, received command and following state are listed in Table 4-1. In the following sections, the different card operation modes will be described first. Thereafter, the restrictions for controlling the clock signal are defined. All card commands together with the corresponding responses, state transitions, error conditions and timings are presented in the following sections.

The MultiMediaCard/RS-MultiMediaCard has two operation modes.

- Card Identification Mode— The host will be in card identification mode after reset and while it is looking for new cards on the bus. MultiMediaCards will be in this mode after reset until the SET_RCA command (CMD3) is received.
- The Interrupt Mode option defined in the MultiMediaCard Standard is not implemented on the SanDisk MultiMediaCard.
- Data Transfer Mode—MultiMediaCards will enter data transfer mode once an RCA is assigned to them. The host will enter data transfer mode after identifying all the MultiMediaCards on the bus.

Table 4-1 lists the dependencies between operation modes and card states. Each state in the card state diagram (Figure 4-1 and 4-2) is associated with one operation mode.

Table 4-1 Card States vs. Operation Modes Overview

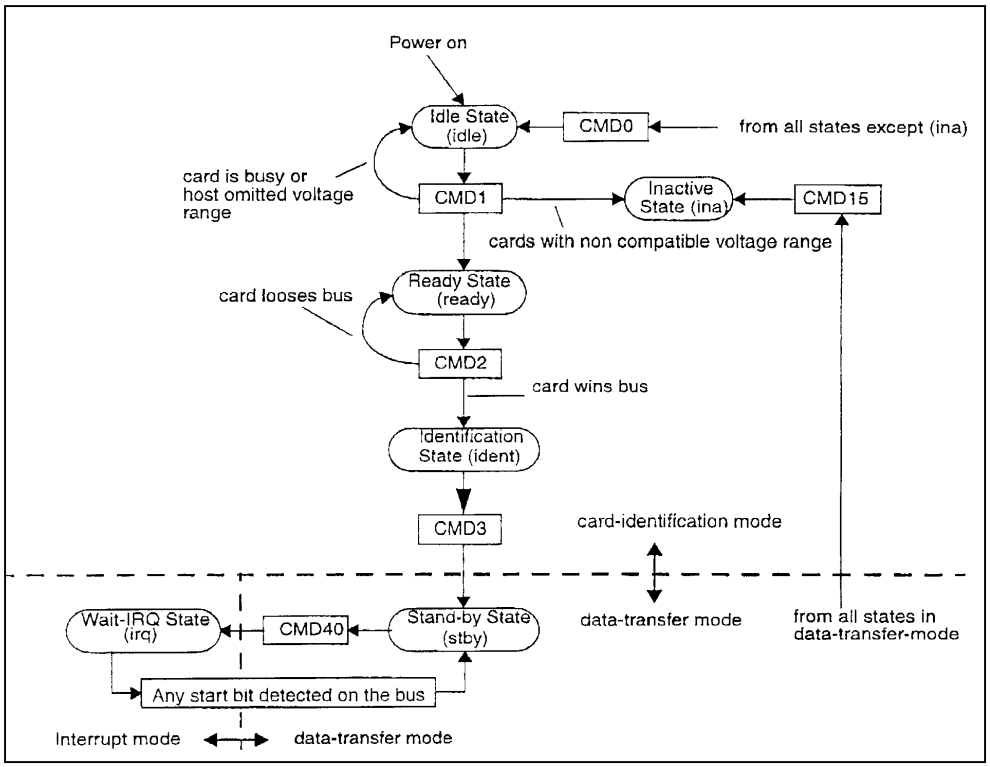
Card State	Operation Mode	Bus Mode
Inactive	Inactive	
Idle, Ready, Identification	Card Identification Mode	Open-drain
Standby, Transfer, Send data, Receive data, Programming, Disconnect	Data Transfer Mode	Push-pull

If a command with improper CRC was received, it is ignored. If there was a command execution (e.g., continuous data read) the card continues in the operation until it gets a correct host command.

4.1 Card Identification Mode

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

Figure 4-1 MultiMediaCard State Diagram—Card Identification Mode



4.1.1 Reset

GO_IDLE_STATE (CMD0) is the software-reset command that sets each MultiMediaCard or RS-MultiMediaCard to Idle State regardless of the current state of the card. Cards already in an inactive state are not affected by this command.

After power-on by the host, all cards are in Idle State, including the cards that were in Inactive State.[1]

After power-on or CMD0, all card 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).

4.1.2 Operating Voltage Range Validation

The MultiMediaCard standard requires that all MultiMediaCards and RS-MultiMediaCards be able to establish communication with the host using any operating voltage between

[1] At least 74 clock cycles are required prior to starting bus communication.

VDD-min and VDD-max. However, during data transfer, minimum and maximum values for VDD are defined in the CSD Register and may not cover the entire range. Card hosts are expected to read the card's CSD Register and select the proper VDD values or reject the card.

A MultiMediaCard/RS-MultiMediaCard that stores the CID and CSD data in the payload memory can communicate this information only under data-transfer VDD conditions. This means if the host and card have incompatible VDD ranges, the card will not be able to complete the identification cycle, nor to send CSD data.

SEND_OP_COND (CMD1) is designed to provide card hosts with a mechanism to identify and reject cards that do not match the host's desired VDD range. To accomplish this task, the host sends the required VDD voltage window as the operand of this command. MultiMediaCards/RS-MultiMediaCards that cannot perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. All other cards will respond concurrently (same method as card identification) sending back their VDD range. The wired-or result of the response will show all voltage ranges, some of which the cards do not support.

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

The MultiMediaCard or RS-MultiMediaCard can use the busy bit in the CMD1 response to tell the host that it is still working on the power-up/reset procedure (e.g., downloading the register information from memory field) and is not ready for communication. In this case the host must repeat CMD1 until the busy bit is cleared.

During the initialization procedure, the host is not allowed to change OCR values; the card will ignore OCR content changes. If there is a actual change in the operating conditions, the host must reset the card stack (using CMD0) and begin the initialization procedure again.

GO_INACTIVE_STATE (CMD15) can also be used to send an addressed MultiMediaCard/RS-MultiMediaCard into the Inactive State. CMD15 is used when the host explicitly wants to deactivate a card—for example, the host changes VDD into a range not supported by this card.

4.1.3 Card Identification Process

The host starts the card identification process in open-drain mode with the identification clock rate f_{OD}. The open-drain driver stages on the CMD line allow parallel card operation during card identification.

After the bus is activated and a valid operation condition is obtained, the host asks all cards for their unique card identification (CID) number with the broadcast command, ALL_SEND_CID (CMD2). All remaining unidentified cards, those in Ready State, simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. The MultiMediaCards and RS-MultiMediaCards with outgoing CID bits that do not match the corresponding bits on the command line, in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle (cards stay in the Ready State). Because CID numbers are unique for each card, there should be only one card that successfully sends its full CID number to the host; the cards then goes into Identification State. The host issues CMD3, (SET_RELATIVE_ADDR) to assign this card a relative address (RCA), which is shorter than CID and used to address the card in future data transfer mode communication typically with a higher clock rate than

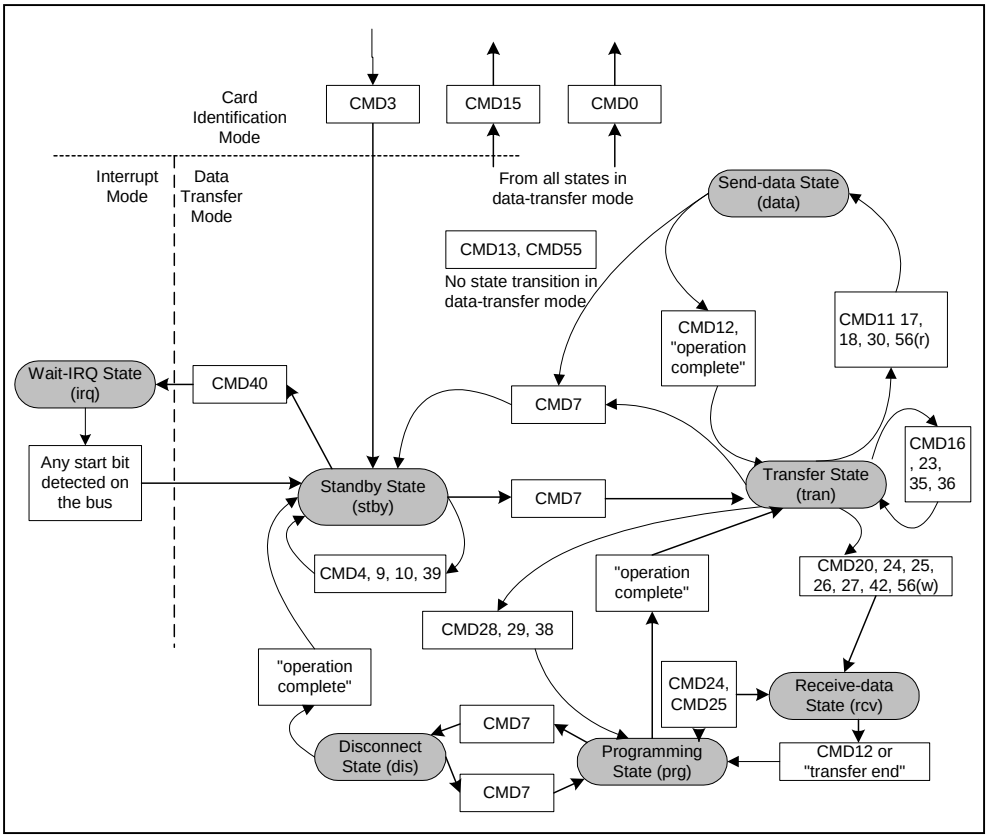
f_{OD}. When the RCA is received, the card transfers to the Stand-by State and does not react to further identification cycles. The MultiMediaCard/RS-MultiMediaCard also switches output drivers from open-drain to push-pull.

The host repeats the identification process as long as it receives a response (CID) to its identification command (CMD2). All cards have been identified when a card does not respond to CMD2. The time-out condition that recognizes completion is the absence of a start bit for more than five clock periods after sending CMD2.

4.2 Data Transfer Mode

When all cards are in Stand-by State, communication over the CMD and DAT lines are in push-pull mode. Until the host knows all CSD Register content, 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 content—for example, ECC type, block length, card storage capacity, and maximum clock rate.

Figure 4-2 MultiMediaCard State Diagram—Data Transfer Mode



CMD7 is used to select one MultiMediaCard/RS-MultiMediaCard and place it in the Transfer State. Only one card can be in the Transfer State at a given time. If a previously selected 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. 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 command flow in this state.

All data communication in the Data Transfer Mode is point-to point between the host and the selected MultiMediaCard (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 the card state diagram, Figure 4-2, and in the following paragraphs:

- The stop command (CMD12) can abort all data read commands at any given time. 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), and send write protect (CMD30).
- The stop command (CMD12) can abort all data write commands at any given time. The write commands must be stopped prior to deselecting the card with CMD7. The write commands are block write (CMD24 and CMD25), write CID (CMD26), and write CSD (CMD27).
- When the data transfer is complete, the MultiMediaCard/RS-MultiMediaCard exits the data write state and moves to either the Programming State (successful transfer) or Transfer State (failed transfer).
- 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: the next block can be sent to the card while the previous is being programmed. If all write buffers are full, and the MMC/RS-MMC is in Programming State (see MultiMediaCard state diagram Figure 4-2), the DAT line will be kept low.
- No buffering option is available for write CSD, write CID, write protection, or erase: no other data transfer commands will be accepted when the MMC/RS-MMC is busy servicing any one of the aforementioned commands. The DAT line will be kept low throughout the period when the card is busy and in Programming State.
- Parameter-set commands are not allowed when the card is programming. Parameter-set commands are set block length (CMD16), and erase tagging/untagging (CMD32-37).
- Read commands are *not* allowed while the card is programming.
- Moving another MultiMediaC/RS-MultiMediaCard from Stand-by to Transfer State (using CMD7) will not terminate a programming operation. The card will switch to the Disconnect State and 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 the card (using CMD0 or CMD15) will terminate any pending or active programming operation, which may destroy the data contents on the card. It is the host's responsibility to prevent the potential destruction of data.

4.2.1 Data Read Format

When data is not being transmitted, the DAT bus line is high. 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 a block-oriented data transfer is preserved by a CRC checksum. The generator polynomial is standard CCITT format: $x^{16} + x^{12} + x^5 + 1$.

Block Read

The basic unit of data transfer is a block whose maximum size is defined by READ_BL_LEN in the CSD. Smaller blocks with a starting and ending address contained entirely within one physical block, as defined by READ_BL_LEN, may also be transmitted. A CRC appended to the end of each block ensures data transfer integrity. CMD17 or *READ_SINGLE_BLOCK* starts a block read and returns the card to the Transfer State after a complete transfer. CMD18 or *READ_MULTIPLE_BLOCK* starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued.

If the host uses partial blocks with an accumulated length that is not block aligned, the card, at the beginning of the first misaligned block, will detect a block misalignment error, set the ADDRESS_ERROR bit in the Status Register, abort transmission, and wait in the Data State for a stop command.

4.2.2 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. Prior to an operation, the MultiMediaCard/RS-MultiMediaCard performs a CRC check for each such received data block.[2] This mechanism prevents the erroneous writing of transferred data.

Block Write

Block write or *CMD24-27* means that one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card will indicate the failure on the DAT line; the transferred data will be discarded and not written and all further transmitted blocks (in multiple block write mode) will be ignored.

If the host uses partial blocks with an accumulated length that is not block aligned, the card, at the beginning of the first misaligned block, will detect a block misalignment error, set the ADDRESS_ERROR bit in the Status Register, abort programming, and wait in the Data State for a stop command.

The write operation will also be aborted if the host tries to write over a write-protected area. In this case, however, the card will set the WP_VIOLATION bit.

After receiving a block of data and completing the CRC check, the card will begin programming and hold the DAT 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 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

[2]The polynomial is the same one used for a read operation.

select a different card), which will place the card in 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 write buffer is unavailable.

4.2.3 CSD Programming

Programming of the CSD register does not require a previous block length setting. After sending CMD27 and receiving an R1 response, the start bit (=0) is sent, the modified CSD Register (=16 bytes), CRC16 (=2 bytes), and end bit (=1). The host can change only the least significant 16 bits [15:0] of the CSD. The rest of the CSD register content must match the MultiMediaCard/RS-MultiMediaCard CSD Register. If the card detects a content inconsistency between the old and new CSD register, it will not reprogram the CSD in order to ensure validity of the CRC field in the CSD Register.

Bits [7:1] are the CRC7 of bits [127:8] of the CSD Register, which should be recalculated once the register changes. After calculating CRC7, the CRC16 should also be calculated for all of the CSD Register [127:0].

4.2.4 Erase

Identification of sectors is accomplished with the TAG_* commands. Either an arbitrary set of sectors within a single erase group or an arbitrary selection of erase groups may be erased at one time but not together; that is, the unit of measure for determining an erase is either a sector or an erase group. If it is a sector, all selected sectors must lay within the same erase group.

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. After a range is selected, an individual sector (or group) within that range can be removed using the UNTAG command.

The host must adhere to the following command sequence: TAG_SECTOR_START, TAG_SECTOR_END, UNTAG_SECTOR (up to 16 UNTAG sector commands can be sent for one erase cycle) and ERASE (or the same sequence for group tagging). Condition exceptions the MultiMediaCard/RS-MultiMediaCard may detect includes:

- An erase or TAG/UNTAG command is received out of sequence. The card will set the ERASE_SEQ_ERROR bit in the Status Register and reset the entire sequence.
- An out-of-sequence command (except SEND_STATUS) is received. The card will 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 will remain intact and only the non-protected sectors will be erased. The WP_ERASE_SKIP status bit in the Status Register will be set.

The address field in the TAG commands is a sector or a group address in byte units. The card will ignore all LSBs below the group or sector size. The number of UNTAG commands (CMD34 and CMD37) used in a sequence is limited for up to 16.

As described for block write, the MultiMediaCard/RS-MultiMediaCard will indicate that an erase is in progress by holding DAT low.

4.2.5 Write Protect Management

Write protection features may protect MultiMediaCard/RS-MultiMediaCard data 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. Portions of the data may also be protected in units of WP_GRP_SIZE sectors as specified in the CSD. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_WRITE_PROT command clears the write protection of the addressed write-protect group.

The SEND_WRITE_PROT command is similar to a single block read command. The card will send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSBs below the group size.

4.2.6 Card Lock/Unlock Operation

The password protection feature enables the host to lock a card while providing a password that will be used later for unlocking the card. The password and its size are kept in 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile which protects a power cycle erase.

Locked cards respond to, and execute, all commands in the *basic* command class (class 0) and *lock card* command class. Therefore, the host is allowed to reset, initialize, select, query for status, etc., but not access data on the MultiMediaCard/RS-MultiMediaCard. If the password was previously set, it will be locked automatically after power on.

Similar to the existing CSD and CID register write commands, the lock/unlock command is available in *transfer state* only; it does not include an address argument and the card has to be selected before using it.

The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all required information of the command—for example, password setting mode, PWD itself, card lock/unlock, etc. Table 4-2 describes the structure of the command data block.

Table 4-2 Lock Card Data Structure

<table>
<tr><th>Byte</th><th>Bit 7</th><th>Bit 6</th><th>Bit 5</th><th>Bit 4</th><th>Bit 3</th><th>Bit 2</th><th>Bit 1</th><th>Bit 0</th></tr>
<tr><td>0</td><td colspan="4">Reserved</td><td>ERASE</td><td>LOCK_UNLOCK</td><td>CLR_PWD</td><td>SET_PWD</td></tr>
<tr><td>1</td><td colspan="8">PWD_LEN</td></tr>
<tr><td>2</td><td colspan="8" rowspan="3">Password Data</td></tr>
<tr><td>…</td></tr>
<tr><td>PWD_LEN + 1</td></tr>
</table>

Bit Name	Description
ERASE	1 = Forced Erase Operation (all other bits shall be ‘0’) and only the CMD byte is sent.
LOCK/UNLOCK	1 = Lock the card. 0 = Unlock the card (it is valid to set this bit together with SET_PWD but it is not

Bit Name	Description
	allowed to set it together with CLR_PWD).
CLR_PWD	1 = Clear PWD.
SET_PWD	1 = Set new password to PWD.
PWD_LEN	Define the following password length (in bytes).
PWD	Password (new or currently used depending on the command).

The host will define the data block size before it sends the card lock/unlock command. This will allow for different password sizes.

- Set Password

The sequence for setting the password is as follows:

1. Select a card (CMD7), if not previously selected already.
2. Define the block length (CMD16), given by the 8bit card lock/unlock mode, the 8 bits password size (in bytes), and the number of bytes of the new password. In case that a password replacement is done, then the block size shall consider that both passwords, the old and the new one, are sent with the command.
3. Send Card Lock/Unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode (SET_PWD), the length (PWD_LEN) and the password itself. In case that a password replacement is done, then the length value (PWD_LEN) shall include both passwords, the old and the new one, and the PWD field shall include the old password (currently used) followed by the new password.
4. In case that the sent old password is not correct (not equal in size and content) then LOCK_UNLOCK_FAILED error bit will be set in the status register and the old password does not change. In case that PWD matches the sent old password then the given new password and its size will be saved in the PWD and PWD_LEN fields, respectively. Note that the password length register (PWD_LEN) indicates if a password is currently set. When it equals '0' there is no password set. If the value of PWD_LEN is not equal to zero the card will lock itself after power up. It is possible to lock the card immediately in the current power session by setting the LOCK/UNLOCK bit (while setting the password) or sending additional command for card lock.

- Reset Password

The sequence for resetting the password is as follows:

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

- Lock Card

The sequence for locking a card is as follows:

1. Select a card (CMD7), if not previously selected already.

2. Define the block length (CMD16), given by the 8-bit card lock/unlock mode, the 8-bit password size (in bytes), and the number of bytes of the currently used password.
3. Send the card lock/unlock command with the appropriate data block size on the data line including 16-bit CRC. The data block shall indicate the mode LOCK, the length (PWD_LEN) and the password (PWD) itself. If the PWD content equals to the sent password then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct then LOCK_UNLOCK_FAILED error bit will be set in the status register. Note that it is possible to set the password and to lock the card in the same sequence. In such case the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent. If the password was previously set (PWD_LEN is not '0'), then the card will be locked automatically after power on reset. An attempt to lock a locked card or to lock a card that does not have a password will fail and the LOCK_UNLOCK_FAILED error bit will be set in the Status Register.

- Unlock Card

The sequence for unlocking a card is as follows:

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

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

Unlocking occurs only for the current power session. As long as the PWD is not cleared the card will be locked automatically on the next power up. The only way to unlock the card is by clearing the password.

An attempt to unlock an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the Status Register.

- Force Erase

In case the user forgot the password (the PWD content) it is possible to erase all the card data content along with the PWD content. This operation is called Forced Erase:

1. Select a card (CMD7), if not previously selected already.
2. Define the block length (CMD16) to 1 byte (8-bit card lock/unlock command). Send the card lock/unlock command with the appropriate data block of one byte on the data line including 16-bit CRC. The data block shall indicate the mode ERASE (the ERASE bit shall be the only bit set).

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

An attempt to force erase on an unlocked card will fail and LOCK_UNLOCK_FAILED error bit will be set in the status register.

4.3 Clock Control

The host can use the MultiMediaCard/RS-MultiMediaCard bus clock signal to set the cards to energy-saving mode or control the bus data flow. The host is allowed to lower the clock frequency or shut it down.

A few restrictions the host must follow include:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the MultiMediaCard/RS-MultiMediaCard and the identification frequency).
- It is an obvious requirement that the clock must be running for the card to output data or response tokens. After the last bus transaction, the host is required, to provide eight clock cycles for the card to complete the operation before shutting down the clock. Following is a list of various 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 card that is busy; the card will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge the card (unless previously disconnected by a deselect command -CMD7) will permanently force the DAT line down.

4.4 Cyclic Redundancy Codes

The Cyclic Redundancy Check (CRC) is intended to protect MultiMediaCard/RS-MultiMediaCard commands, responses, and data transfer against transmission errors on its bus. One CRC is generated for every command and checked for every response on the CMD line. For data blocks, CRCs are generated for each DAT line per transferred block. The CRC is generated and checked as shown in the following subsections.

4.4.1 CRC7

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

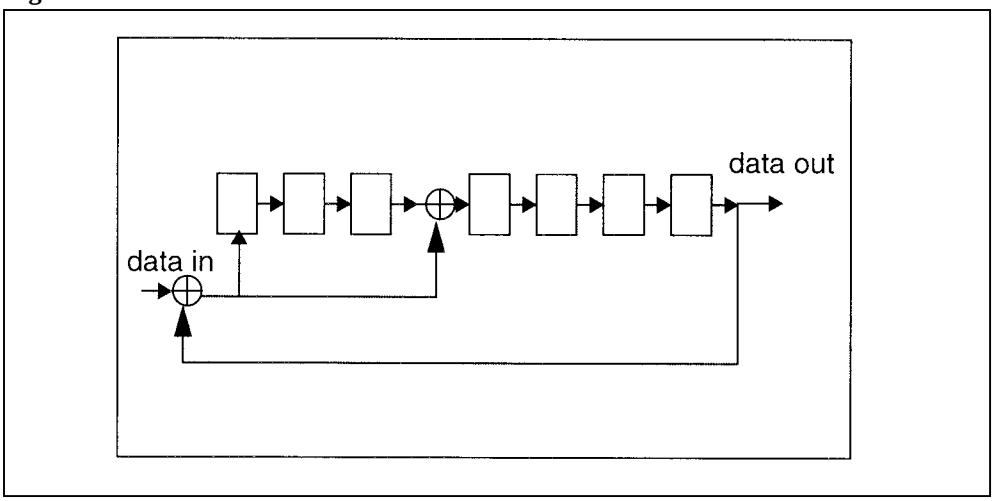
generator polynomial: $G(x) = x^7 + x^3 + 1$

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

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

All CRC registers are initialized to zero. 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*) registers.

Figure 4-3 CRC7 Generator/Checker



4.4.2 CRC16

The CRC16 is used for payload protection in block transfer mode. The CRC checksum is a 16-bit value and is computed as follows:

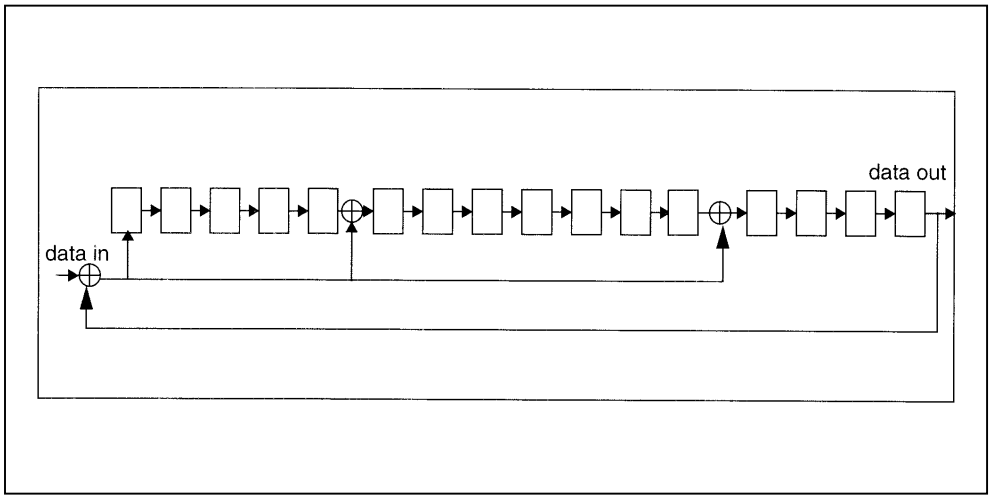
generator polynomial: $G(x) = x^{16} + x^{12} + x^{5+1}$

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

$CRC[15...0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

All CRC registers are initialized to zero. 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 < 16,383).

Figure 4-4 CRC16 Generator/Checker



4.5 Error Conditions

The following sections provide valuable information on error conditions.

4.5.1 CRC and Illegal Commands

CRC bits protect all commands. If the addressed MultiMediaCard/RS-MultiMediaCard CRC check fails, the card does not respond and the command is not executed. The card does not change its state, and the COM_CRC_ERROR bit is set in the Status Register.

Similarly, if an illegal command has been received, an MMC/RS-MMC will not change its state or respond, and will set the ILLEGAL_COMMAND error bit in the Status Register. Only the non-erroneous state branches are shown in the state diagrams (Figure 4-1 and Figure 4-2). Table 4-7 contains a complete state transition description.

Different types of illegal commands include:

- Commands belonging to classes not supported by the MMC/RS-MMC (e.g., I/O command CMD39).
- Commands not allowed in the current state (e.g., CMD2 in Transfer State).
- Commands not defined (e.g., CMD6).

4.5.2 Read, Write and Erase Time-out Conditions

The period after which a time-out condition for read/write/erase operations occurs is (card independent) 10 times longer than the typical access/program times for the operations given in Table 4-3. A card will complete the command within this time period, or give up and return an error message. If the host does not get a response within the defined time-out it should assume the card is not going to respond any more and try to recover (e.g., reset the card, power cycle, reject). The typical access and program times are defined as shown in Table 4-3.

Table 4-3 Typical Access and Program Time

Operation	Definition
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 sectors to be erased multiplied by the block write delay.

4.6 Commands

The following sections provide important information on commands.

4.6.1 Command Types

There are four kinds of commands defined to control the MultiMediaCard/RS-MultiMediaCard bus as shown in Table 4-4.

Table 4-4 Command Definition

Command	Abbreviation	Definition
Broadcast	bc	Sent on CMD, no response
Broadcast w/Response	bcr	Sent on CMD, response[3] on CMD
Addressed point-to-point	ac	Sent on CMD, response on CMD
Addressed point-to-point data transfer	adtc	Sent on CMD, response on CMD, data transfer on DAT

The command transmission always starts with the most significant bit (MSB).

4.6.2 Command Format

The command length shown in Figure 4-5 is 48 bits.

Figure 4-5 Format (2.4 µs @ 20 MHz)

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

Commands and arguments are listed in Table 4-6.

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

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

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

4.6.3 Command Classes

The command set of the MultiMediaCard/RS-MultiMediaCard is divided into several classes (refer to Table 4-5). Each class supports a set of card functions.

The supported Card Command Classes (CCC) are coded as a parameter in the CSD Register data of each card, providing the host with information on how to access the card.

[3] All cards simultaneously.

Table 4-5 Card Command Classes

Class	0	2	4	5	6	7	8	9	10-11
CMD	**Basic**	**Block Read**	**Block Write**	**Erase**	**Write Protec-tion**	**Lock Card**	**Appli-cation Spe-cific**	**I/O Mode**	**R**
CMD0	+								
CMD1	+								
CMD2	+								
CMD3	+								
CMD4	+								
CMD7	+								
CMD9	+								
CMD10	+								
CMD11									
CMD12	+								
CMD13	+								
CMD15	+								
CMD16		+	+						
CMD17		+							
CMD18		+							
CMD20									
CMD23		+	+						
CMD24			+						
CMD25			+						
CMD26			+						
CMD27			+						
CMD28					+				
CMD29					+				
CMD30					+				
CMD32				+					
CMD33				+					
CMD34				+					
CMD35				+					
CMD36				+					
CMD37				+					
CMD38				+					
CMD39								+	
CMD40								+	
CMD42						+			
CMD55							+		

Class	0	2	4	5	6	7	8	9	10-11
CMD	**Basic**	**Block Read**	**Block Write**	**Erase**	**Write Protection**	**Lock Card**	**Application Specific**	**I/O Mode**	**R**
CMD56							+		

4.6.4 Command Description

All future reserved commands and their responses must be 48 bits long. Responses may not have any response. Table 4-6 details the MultiMediaCard/RS-MultiMediaCard bus commands.

Table 4-6 Command Descriptions

CMD Index	Type	Argument	Resp.	Abbreviation	Description
Basic Commands (Class 0 and Class 1)					
CMD0	bc	[31:0] don't care	---	GO_IDLE_STATE	Reset all cards to Idle State.
CMD1	bcr	[31:0] OCR w/out busy	R3	SEND_OP_COND	Ask all cards in idle state to send their operation conditions register content in the response on the CMD line.
CMD2	bcr	[31:0] don't care	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3	ac	[31:16]RCA [15:0] don't care	R1	SEND_RELATIVE_ADDR	Assign relative address to the card
CMD4	Not Supported				
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16]RCA [15:0] don't care	R1 (from selected card only)	SELECT/DESELECT_CARD	Toggles card between the Stand-by and Transfer states or Programming and Disconnect states. In both cases, the card is selected by its own relative address and deselected by any other address; address 0 deselects all.
CMD8	Reserved				
CMD9	ac	[31:16]RCA [15:0] don't care	R2	SEND_CSD	Sends addressed card's card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16]RCA [15:0] don't care	R2	SEND_CID	Sends addressed card's card identification (CID) on the CMD line.

CMD Index	Type	Argument	Resp.	Abbreviation	Description
CMD11	Not supported				
CMD12	ac	[31:0] don't care	R1b	STOP_TRANSMISSION	Terminates a multiple block read/write operation.
CMD13	ac	[31:16]RCA [15:0] don't care	R1	SEND_STATUS	Sends addressed card's status register.
CMD14	Reserved				
CMD15	ac	[31:16]RCA [15:0] don't care	---	GO_INACTIVE_STATE	Sets the card to inactive state.
Block Read Commands (Class 2)					
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all subsequent block commands (read and write).
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Sends blocks of data continuously until interrupted by a stop transmission or a new read command.
CMD19	Reserved				
Block Write Commands (Class 4)					
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Writes blocks of data continuously until a STOP_TRANSMISSION is received.
CMD26	Not applicable				
CMD27	adtc	[31:0] don't care	R1	PROGRAM_CSD	Programs the programmable bits of the CSD.
Write Protection Commands (Class 6)					
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	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	Clears the write protection bit of the addressed group if the card provides write protection features.

CMD Index	Type	Argument	Resp.	Abbreviation	Description
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	Asks the card to send the status of the write protection bits if the card provides write protection features.
CMD31	Reserved				
Erase Commands (Class 5)					
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the first sector's address of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selected erase group, or the address of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be erased.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] don't care	R1b	ERASE	Erases all previously selected write blocks.
I/O Mode Commands (Class 9)					
CMD39 CMD40	MMCA Optional Command, currently not supported.				
CMD41	Reserved				
Lock Card Commands (Class 7)					
CMD42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43 … CMD54	MMCA Optional Command, currently not supported.				
Application-specific Commands (Class 8)					

CMD Index	Type	Argument	Resp.	Abbreviation	Description
CMD55 … CMD56	MMCA Optional Command, currently not supported				

4.7 Card State Transition

Table 4-7 contains information about the MultiMediaCard/RS-MultiMediaCard state-transition dependence on the received command.

Table 4-7 Card State Transition Table

	Current Status										
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**	**irq**
Command	**Changes to**										
Class Independent											
CRC error	---	---	---	---	---	---	---	---	---	---	stby
Command not supported	---	---	---	---	---	---	---	---	---	---	stby
Class 0											
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	---	stby
CMD1, card VDD range compatible	ready	---	---	---	---	---	---	---	---	---	stby
CMD1, card is busy	idle	---	---	---	---	---	---	---	---	---	stby
CMD1, card VDD range not compatible	ina	---	---	---	---	---	---	---	---	---	stby
CMD2, card wins bus	---	ident	---	---	---	---	---	---	---	---	stby
CMD2, card loses bus	---	ready	---	---	---	---	---	---	---	---	stby
CMD3	---	---	stby	---	---	---	---	---	---	---	stby
CMD4	Not supported										
CMD7, card is addressed	---	---	---	tran	---	---	---	---	prg	---	stby
CMD7, card is not addressed	---	---	---	--	stby	stby	---	dis	---	---	stby
CMD9	---	---	---	stby	---	---	---	---	---	---	stby
CMD10	---	---	---	stby	---	---	---	---	---	---	stby
CMD12	---	---	---	---	---	tran	prg	---	---	---	stby
CMD13	---	---	---	stby	tran	data	rcv	prg	dis	---	stby
CMD15	---	---	---	ina	ina	ina	ina	ina	ina	---	stby

	Current Status										
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**	**irq**
Command	**Changes to**										
Class 1											
CMD11	---	---	---	---	data	---	---	---	---	---	stby
Class 2											
CMD16	---	---	---	---	tran	---	---	---	---	---	stby
CMD17	---	---	---	---	data	---	---	---	---	---	stby
CMD18	---	---	---	---	data	---	---	---	---	---	stby
CMD23	---	---	---	---	tran	---	---	---	---	---	stby
Class 3											
CMD20	---	---	---	---	rcv	---	---	---	---	---	stby
Class 4											
CMD16	See Class 2										
CMD23	See Class 2										
CMD24	---	---	---	---	rcv	---	---	---	---	---	stby
CMD25	---	---	---	---	rcv	---	---	---	---	---	stby
CMD26	---	---	---	---	rcv	---	---	---	---	---	stby
CMD27	---	---	---	---	rcv	---	---	---	---	---	stby
Class 6											
CMD28	---	---	---	---	prg	---	---	---	---	---	stby
CMD29	---	---	---	---	prg	---	---	---	---	---	stby
CMD30	---	---	---	---	data	---	---	---	---	---	stby
Class 5											
CMD35	---	---	---	---	tran	---	---	---	---	---	stby
CMD36	---	---	---	---	tran	---	---	---	---	---	stby
CMD38	---	---	---	---	prg	---	---	---	---	---	stby
Class 7											
CMD42	---	---	---	---	rcv	---	---	---	---	---	stby
Class 8											
CMD55	---	---	---	stby	tran	data	rcv	prg	dis	---	irq
CMD56: RD/WR = 0	---	---	---	---	rcv	---	---	---	---	---	---
CMD56: RD/WR = 1	---	---	---	---	data	---	---	---	---	---	stby
Class 9											
CMD39	---	---	---	stby	---	---	---	---	---	---	stby
CMD40	---	---	---	irq	---	---	---	---	---	---	stby

	Current Status										
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**	**irq**
Command	**Changes to**										
Class 10-11											
CMD41; CMD43…CMD 54; CMD57-CMD59 …CMD59	Reserved										
CMD60-CMD63	Reserved for manufacturer										

4.7.1 Responses

All responses are sent on 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 (0), followed by the bit indicating the direction of transmission (card = 0). A value denoted by *x* in the Table 4-8 to 4-10 indicates a variable entry. All responses except for the type R3 are protected by a CRC. The end bit (1) terminates every response.

There are five types of responses supported in the SanDisk MultiMediaCard and RS-MultiMediaCard. Their formats are defined as follows:

1) **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.

Table 4-8 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

2) **R1b** is identical to R1 with the additional busy signal transmitted on the data line.

3) **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. The only bits transferred are [127...1] of the CID and CSD; Bit [0] of these registers is replaced by the end bit of the response.

Table 4-9 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 inc. internal CRC7	end bit

4) **R3** (OCR register): response length 48 bits.

The contents of the OCR Register are sent as a response to CMD1.

Table 4-10 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	111111	1
Description	start bit	transmission bit	reserved	OCR Register	reserved	end bit

R4 and R5: responses are not supported.

4.8 Timing Diagrams

All timing diagrams use schematics and abbreviations listed in Table 4-11.

Table 4-11 Timing Diagram Symbols

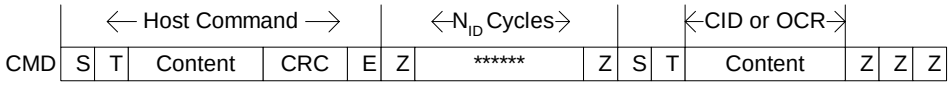
Symbol	Definition
S	Start Bit (= 0)
T	Transmitter Bit (Host = 1, Card = 0)
P	One-cycle pull-up (= 1)
E	End Bit (= 1)
Z	High Impedance State (-> = 1)
D	Data bits
X	Repeater
CRC	Cyclic Redundancy Check Bits (7 bits)
	Card active
	Host active

4.8.1 Command and Response

Card Identification and Card Operation Conditions Timing

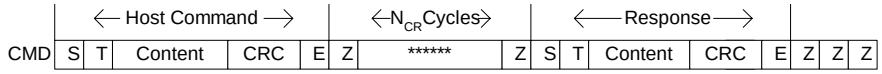
The card identification (CMD2) and card operation conditions (CMD1) timing are processed in the open-drain mode. The card response to the host command starts after exactly NID clock cycles.

Identification Timing (Card ID Mode)



The minimum delay between the host command and card response is N_{CR} clock cycles. This timing diagram is relevant for host command CMD3.

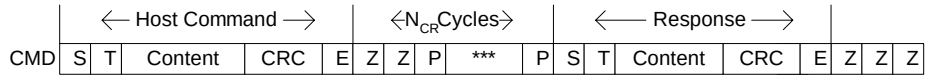
Command Response Timing (ID Mode)



Data Transfer Mode

There is just one Z bit period followed by P bits pushed up by the responding card. This timing diagram is relevant for all responded host commands except CMD1, 2, 3.

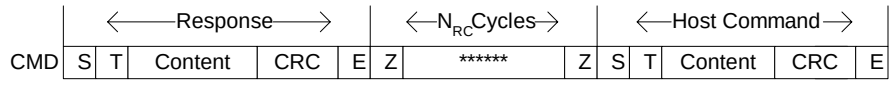
Command Response Timing (Data Transfer Mode)



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.

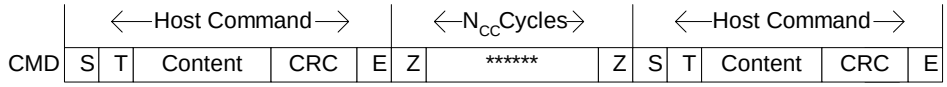
Timing Response End to Next CMD Start (Data Transfer Mode)



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. This timing is relevant for any host command that does not have a response.

Timing CMD_n End to CMD_{n+1} Start (all modes)



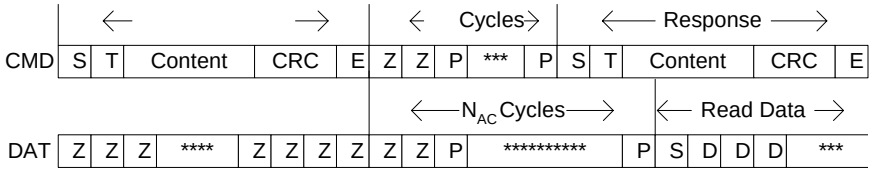
In the case where the CMD_n command was a last acquisition command with no further response by any card, then the next CMD_{n+1} command is allowed to follow after at least N_{CC} +136 (the length of the R2 response) clock periods.

4.9 Data Read

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 shown in the *Transfer of Single Block Read* timing diagram. The sequence starts with a single block read command, CMD17 that specifies the start address in the argument field. The response is sent on the CMD line as usual.

Transfer of Single Block Read

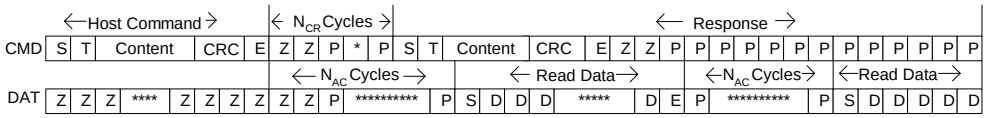


Data transmission from the card starts after the access time delay N_{AC} beginning from the end bit of the read command. After the last data bit, the CRC check bits are suffixed to allow the host to check for transmission errors.

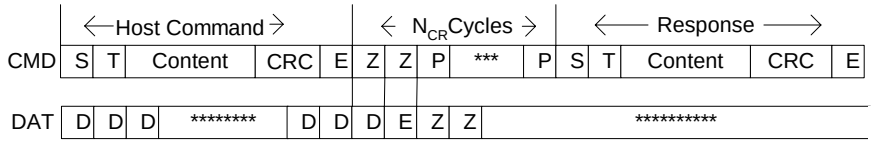
Multiple Block Read

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. The *Timing of Multiple Block Read Command* timing diagram describes the timing of the data blocks, and the *Timing of Stop Command (CMD12,Data Transfer Mode)* timing diagram describes the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.

Timing of Multiple Block Read Command



Timing of Stop Command (CMD12, Data Transfer Mode)



4.10 Data Write

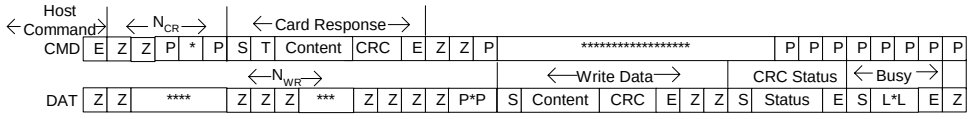
Single Block Write

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 shown in the *Block Write Command* timing diagram. The sequence starts with a single block write command, CMD24 that determines (in the argument field) the start address. The card responds on the CMD line, and 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 data 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.

Block Write Command Timing

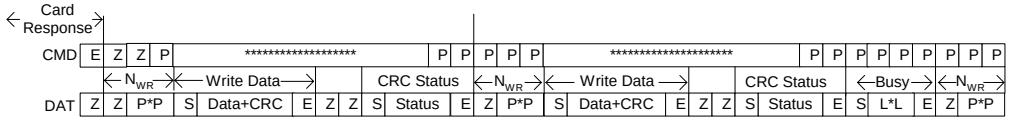


If the MultiMediaCard/RS-MultiMediaCard does not have a free data receive buffer, it indicates this condition by pulling down the data line to *low*. The card stops pulling down the data line as soon as at least one receive buffer for the defined data transfer block length becomes free. This signaling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

Multiple Block Write

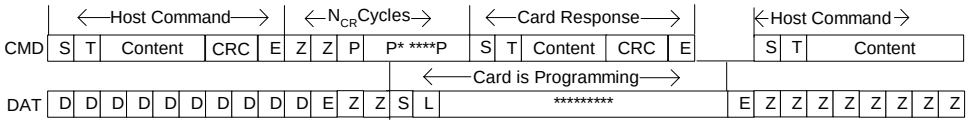
In multiple block write mode, the card expects continuous flow of data blocks following the initial host write command. The data flow is terminated by a stop transmission command (CMD12). The *Multiple Block Write Command* timing diagram describes the timing of the data blocks with and without card busy signal.

Multiple Block Write Command Timing



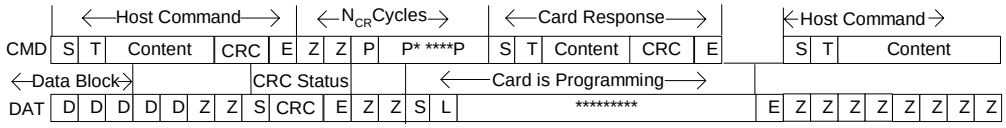
In write mode, the stop transmission command works similarly to the stop transmission command in the read mode. The following figure describes the timing of the stop command in different card states.

Stop Transmission During Data Transfer from the Host



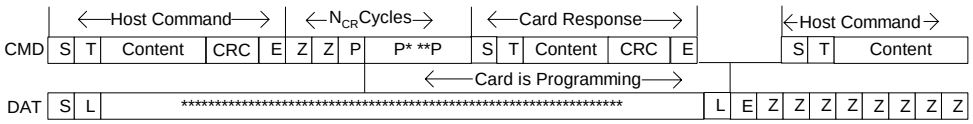
The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status token sent back to the host. The figure below is an example of an interrupted (by a host stop command) attempt to transmit the CRC status block. The sequence 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.

Stop Transmission during CRC Status Transfer from the Card

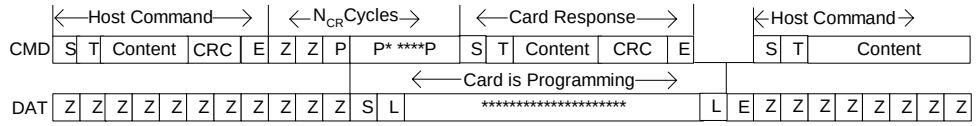


All previous examples dealt with the scenario of the host stopping the data transmission during an active data transfer. The following two diagrams describe a scenario of receiving the stop transmission between data blocks. In the first example, the card is busy programming the last block while the card is idle as shown in the second diagram. However, there remains un-programmed 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.

Stop Transmission Received after Last Data Block—Card Busy Programming



Stop Transmission Received after Last Data Block—Card becomes Busy



Erase, Set and Clear Write Protect Timing

The host must first tag the sectors to erase using the tag commands, CMD32-CMD37. The erase command, CMD38 once issued, will erase all tagged sectors. 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.

4.11 Timing Values

Table 4-12 defines all timing values.

Table 4-12 Timing Values

Value	Min.	Max.	Unit
N_{CR}	2	64	Clock cycles
N_{ID}	5	5	Clock cycles
N_{AC}	2	Note	Clock cycles
N_{RC}	8	---	Clock cycles
N_{CC}	8	---	Clock cycles
N_{WR}	2	---	Clock cycles

NOTE: 10 * ((TAAC*f) + (100*NSAC))]* where *f* is the clock frequency.

5 SPI Mode

SPI mode is a secondary, optional communication protocol, which is offered by the MultiMediaCard and RS-MultiMediaCard. This mode is a subset of the MultiMediaCard Protocol, designed to communicate with an SPI channel, commonly found in some vendors' microcontrollers. The interface is selected during the first reset command after power up (CMD0) and cannot be changed once the part is powered on.

The SPI standard defines the physical link only, and not the complete data transfer protocol. The MultiMediaCard/RS-MultiMediaCard SPI implementation uses a subset of the MultiMediaCard Protocol and command set because it pertains to systems that require a small number of cards (typically one) and have lower data transfer rates (compared to MultiMediaCard Protocol-based systems). From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to a minimum. The disadvantage is the loss of performance with SPI mode as compared to MultiMediaCard mode (lower data transfer rate, fewer cards, hardware CS per card, etc.).

5.1 SPI Interface Concept

The SPI is a general-purpose synchronous serial interface originally found on certain Motorola microcontrollers. A virtually identical interface can now be found on some other microcontrollers as well.

The MultiMediaCard SPI interface is compatible with SPI hosts available on the market. As in any other SPI device, the MultiMediaCard 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

Byte transfers are another common SPI characteristic. They are implemented in the card as well. All data tokens are multiples of bytes (8-bit) and always byte aligned to the CS signal.

5.2 SPI Bus Topology

Card identification and addressing methods are replaced by the hardware Chip Select (CS) signal; there are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal. See the following figure.

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

The bi-directional CMD and DAT lines are replaced by unidirectional *dataIn* and *dataOut* signals. This eliminates the ability to execute commands while data is being read or written and, therefore, makes the sequential and multi block read/write operations obsolete. The SPI channel supports single block read/write commands only.

The SPI interface uses the same seven signals as the standard MultiMediaCard bus (Figure 5-1).

Figure 5-1 MultiMediaCard/RS-MultiMediaCard Bus System

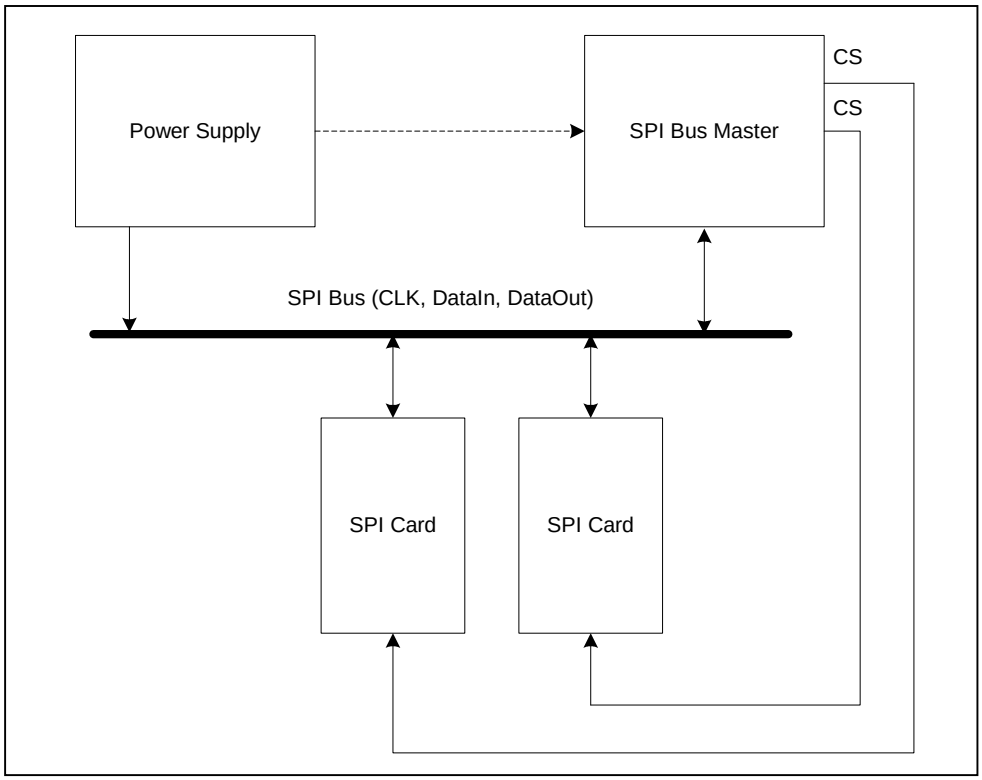


Table 5-1 SPI Interface Pin Configuration

Pin #	MultiMediaCard Mode			SPI Mode		
	Name	Type[1]	Description	Name	Type	Description
1	RSV	NC	Reserved for future use	CS	I	Chip Select (neg true)
2	CMD	I/O/PP/OD	Command/Response	DI	I/PP	Data In
3	VSS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	VDD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	VSS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT	I/O/PP	Data	DO	O/PP	Data Out

1) S: power supply; I: input; O: output; PP: push-pull; OD: open-drain; NC: Not connected (or logical high).

5.3 Card Registers in SPI Mode

The register usage in SPI mode is summarized in Table 5-2. Most of them are inaccessible.

Table 5-2 MultiMediaCard/RS-MultiMediaCard Registers in SPI Mode

Name	Available in SPI mode	Width [Bytes]	Description
CID	Yes	16	Card identification data (serial number, manufacturer ID, etc.).
RCA	No		
DSR	No		
CSD	Yes	16	Card-specific data, information about the card operation conditions.
OCR	Yes	32	Operation condition register.

5.4 SPI Bus Protocol

While the MultiMediaCard channel is based on command and data bit streams, which are initiated by a start bit and terminated by a stop, bit, the SPI channel is byte oriented. Every command or data block is built of 8-bit bytes and is byte aligned to the CS signal (i.e., the length is a multiple of 8 clock cycles).

Similar to the MultiMediaCard Protocol, SPI messages consist of command, response and data-block tokens. The host (master) controls all communication between the host and cards. The host starts every bus transaction by asserting the CS signal low.

The response behavior in SPI mode differs from MultiMediaCard mode in the following three aspects:

· Selected card always responds to the command.

· Additional (8, 16 and 40 bit) response structures are used.

· Card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than by a time-out, as in the MultiMediaCard mode.

Only single and multiple block read/write operations are supported in SPI mode (sequential mode is not supported). In addition to the command response, every data block sent to the card during a write operation will be responded to with a special data response token. A data block may be as big as one card sector and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

5.5 Mode Selection

The MultiMediaCard and RS_MultiMediaCard “wake up” in the MultiMediaCard mode. The card will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0). Selecting SPI mode is not restricted to Idle state (the state the card enters after power up) only. Every time the card receives CMD0, including while in Inactive state, CS signal is sampled.

If the card recognizes that MultiMediaCard mode is required (CS signal is high), it will not respond to the command and remain in MultiMediaCard mode. If SPI mode is required

(CS signal low), the module will switch to SPI mode and respond with the SPI mode R1 response.

The only way to return to MultiMediaCard mode is by a power cycle (turning the power off and on). In SPI mode, the MultiMediaCard protocol state machine is not observed. All of the MultiMediaCard commands supported in SPI mode are always available.

5.6 Bus Transfer Protection

CRC bits protect every MultiMediaCard/RS-MultiMediaCard token transferred on the bus. In SPI mode, the card offers a non-protected mode that enables systems built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.

The SPI interface is initialized in the non-protected mode. However, the RESET command (CMD0), which is used to switch the card to SPI mode, is received by the card while in MultiMediaCard mode and, therefore, must have a valid CRC field.

Since CMD0 has no arguments, the content of all the fields, including the CRC field, are constants and need not be calculated in run time. A valid reset command is:

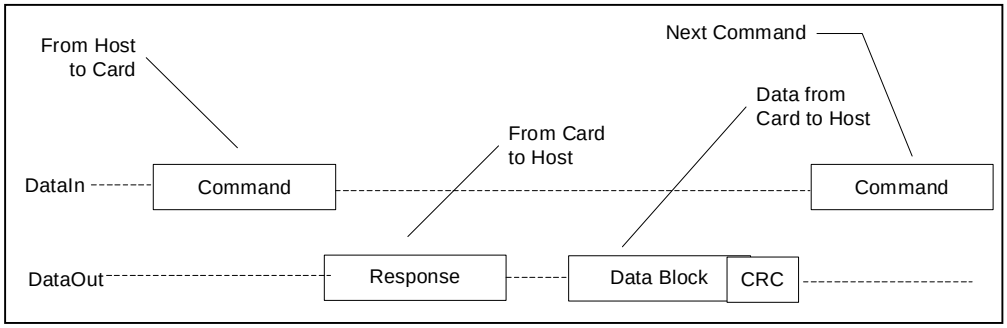
0x40, 0x0, 0x0, 0x0, 0x0, 0x95

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

5.7 Data Read

SPI Mode supports single block and multiple-block read operations The main difference between SPI and MultiMediaCard modes is that the data and the response are both transmitted to the host on the DataOut signal. Therefore, the card response to the STOP_COMMAND might end abruptly and replace the last data block. (Figure 5-2).

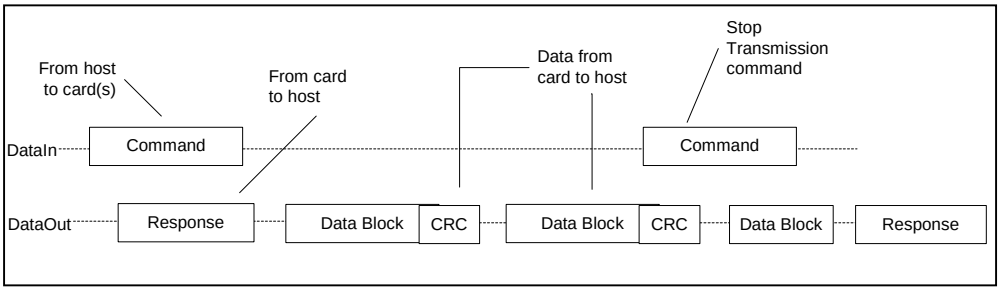
Figure 5-2 Single Block Read Operation



The basic unit of data transfer is a block whose maximum size is defined in the CSD (READ_BL_LEN). If READ_BL_PARTIAL is set, smaller blocks whose starting and ending address are entirely 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) initiates a single block read.

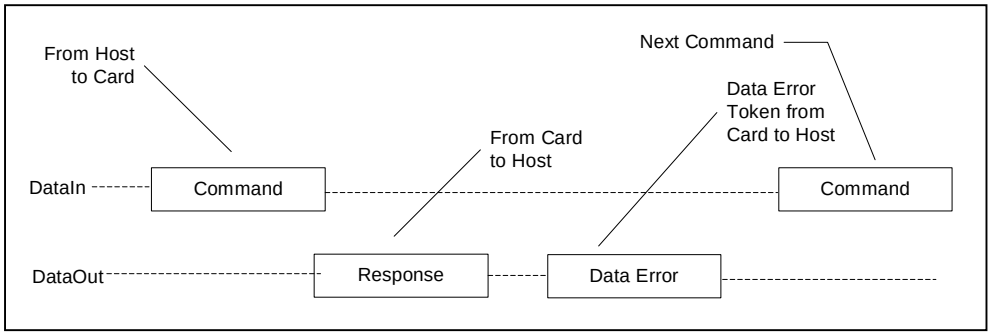
CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. The number of blocks for the multiple block read operation is not defined. The card will continuously transfer data blocks until a stop transmission command is received.

Figure 5-2 Multiple Block Read Operation



In case of a data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 5-4 shows a single block-read operation, which terminated with an error token rather than a data block.

Figure 5-4 Read Operation—Data Error



The multiple block read operation can be terminated the same way by the error token replacing a data block anywhere in the sequence. The host must then abort the operation by sending the Stop Transmission command.

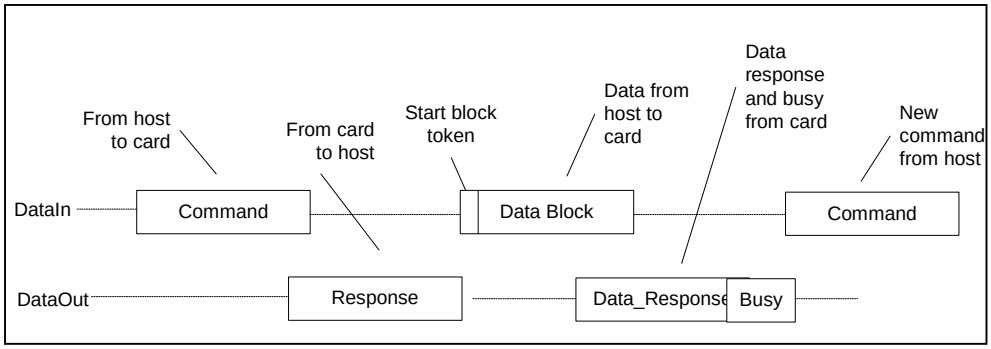
If the host sends a Stop Transmission command out of the valid sequence, it will be responded to as an illegal command.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed, the card detects a block misalignment error condition at the beginning of the first misaligned block (ADDRESS_ERROR error bit is set in the data error token).

5.8 Data Write

In SPI Mode, the MultiMediaCard/RS-MultiMediaCard supports single block or multiple block write operations. Upon reception of a valid write command (CMD24 or CMD25), the card will respond with a response token and wait for a data block to be sent from the host. CRC suffix, block address, and start address restrictions are identical to the read operation. If a CRC error is detected it will be reported in the data-response token and the data block will not be programmed.

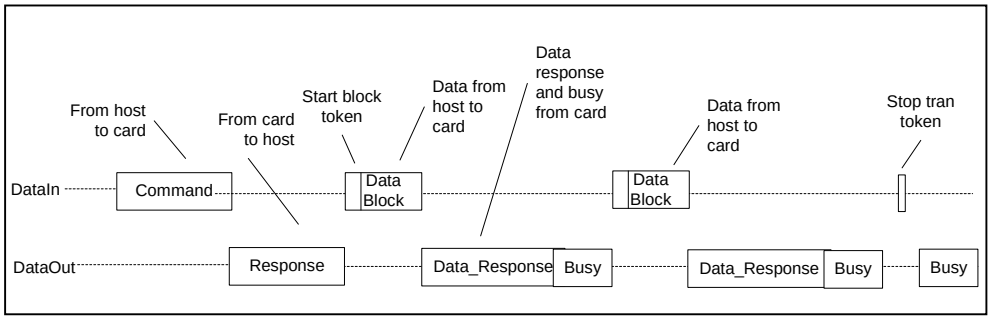
Figure 5-7 Single Block Write Operation



Every data block has a prefix or “start block” token (one byte). After a data block is received, the MultiMediaCard/RS-MultiMediaCard will respond with a data-response token, and if the data block is received with no errors, it will be programmed. A continuous stream of busy tokens will be sent to the host (effectively holding the dataOut line low) for the duration of time the card is busy, programming.

In multiple-block write operations, the stop transmission is accomplished by sending, at the beginning of the next block, a Stop Tran token, instead of a Start Block token.

Figure 5-9 Multiple Block Write Operation



The number of blocks for the write multiple block operation is not defined. The card will continuously accept and program data blocks until a ‘Stop Tran’ token is received.

If the card detects a CRC error or a programming error (e.g., write protect violation, out of range, address misalignment, internal error) during a multiple block write operation, it will report the failure in the data-response token and ignore any further incoming data blocks. The host must then abort the operation by sending the ‘Stop Tran’ token.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error before the beginning of the first misaligned block and responds with an error indication in the data response token, ignoring any further incoming data blocks. The host must then abort the operation by sending the ‘Stop Tran’ token.

Once the programming operation is completed (either successfully or with an error), the host must check the results of the programming (or the cause of the error if already reported in the data-response token) using the SEND_STATUS command (CMD13).

Resetting the CS signal while the card is busy does not terminate the programming process. Instead, the card releases the dataOut line (tri-state) and continues programming. If the card

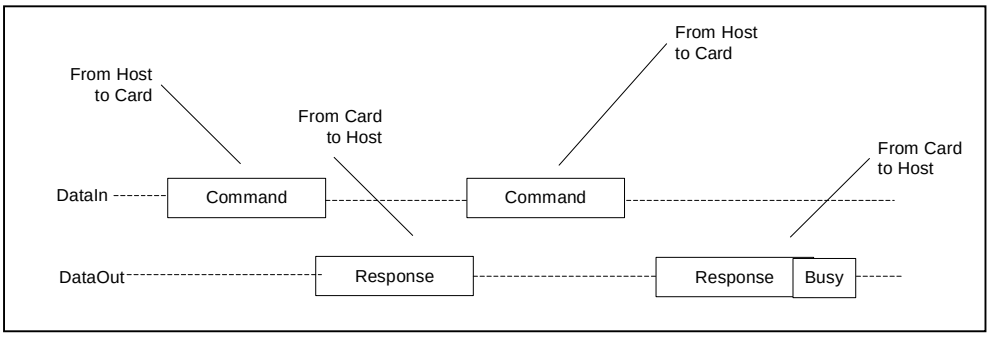
is reselected before the programming has finished, the dataOut line will be forced back to low and all commands will be rejected.

Resetting a card (using CMD0) will terminate any pending or active programming operation. However, this may destroy data formats on the card. It is the host's responsibility to prevent that from happening.

5.9 Erase and Write Protect Management

Erase and Write Protect Management procedures in SPI Mode and MultiMediaCard mode are identical. When the card is erasing or changing the write protection bits of the predefined write-protect group list, it will be in a busy state and hold the dataOut line low. Figure 5-11 illustrates a "no data" bus transaction with and without busy signaling.

Figure 5-11 No Data Operations



5.10 Read CID/CSD Registers

Unlike MultiMediaCard Protocol where the register contents are sent as a command response, SPI Mode provides a simple read block transaction for reading the contents of the CSD and CID registers. 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 timeout for the CSD command cannot be set to the card TAAC because its value is stored in the CSD. Therefore, the standard response timeout value (N_{CR}) is used for read latency of the CSD register.

5.11 Reset Sequence

The MultiMediaCard/RS-MultiMediaCard requires a defined reset sequence. After power-on reset or software reset (CMD0), the card enters an idle state. In this state, the only legal host commands are CMD1 (SEND_OP_COND) 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.

However, in SPI mode, 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 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.

5.12 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 MultiMediaCard/RS-MultiMediaCard.
- The clock must be running for the card to output data or response tokens.

After the last SPI bus transaction, the host is required to provide eight clock cycles for the card to complete the operation before shutting down the clock. The state of the CS signal is irrelevant throughout this eight-clock period. The signal can be asserted or de-asserted. Various SPI bus transactions are listed below.

  - Command/response sequence; occurs eight clocks after the card response end bit. The CS signal can be asserted or de-asserted during the eight clocks.
  - Read data transaction; occurs eight clocks after the end bit of the last data block.
  - Write data transaction; occurs eight clocks after the CRC status token.

- The host is allowed to shut down the clock of a busy card. The MultiMediaCard/RS-MultiMediaCard will complete the programming operation regardless of the host clock. However, the host must provide a clock edge for the card to turn off its busy signal. Without a clock edge, the card (unless previously disconnected by de-asserting the CS signal) will permanently force the dataOut line down.

5.13 Error Conditions

The following sections provide valuable information on error conditions.

5.13.1 CRC and Illegal Commands

The CRC bits provide an optional protection of all commands. If the addressed MultiMediaCard/RS-MultiMediaCard CRC check fails, the COM_CRC_ERROR bit will be set in the card's response. Similarly, if an illegal command has been received, the ILLEGAL_COMMAND bit will be set in the card’s response.

Types of illegal commands include:

- Command belongs to a class not supported by the MultiMediaCard/RS-MultiMediaCard
  - For example, Interrupt and I/O commands
- Command not allowed in SPI Mode
- Command is not defined
  - For example, CMD6

5.13.2 Read, Write and Erase Timeout Conditions

The time period after which a timeout condition for read/write/erase operations occur is card independent and ten times longer than the typical access/program times for these operations given in Table 5-1. A card will complete the command within the stated time

period or return an error message. If the host does not get any response within the given timeout, it ascertains that the card is not going to respond any further and will try to recover (e.g., reset module, power cycle, reject). The typical access and program times are defined in Table 5-3.

Table 5-3 Timeout Conditions

Operation	Access and Program Times
Read	The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block. This number is card dependent.
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 block write commands).
Erase	The duration of an erase command will be (order of magnitude) the number of sectors to be erased multiplied by the block write delay.

5.14 Read Ahead in Multiple Block Read Operation

In Multiple Block read operations, in order to improve read performance, the card may fetch data from the memory array, ahead of the host. In this case, when the host is reading the last addresses of the memory, the card attempts to fetch data beyond the last physical memory address and generates an OUT_OF_RANGE error. Therefore, even if the host times the Stop Transmission command to stop the card immediately after the last byte of data was read, the card may already have generated the error, which will show in the response to the Stop Transmission command. The host should ignore this error.

5.15 Memory Array Partitioning

Refer to MultiMediaCard Mode section.

5.16 Card Lock/Unlock Operation

Refer to MultiMediaCard Mode section.

5.17 SPI Command Set

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

5.17.1 Command Classes

As in MultiMediaCard Mode, the SPI commands are divided into several classes, and each class supports a set of card functions as shown in Table 5-4. The MultiMediaCard/RS-MultiMediaCard supports the same set of optional command classes in both communication modes.[2] The available command classes and supported commands for a specific class, however, are different in the MultiMediaCard and the SPI Communication modes.

[2] There is only one command class table in the CSD register.

Table 5-4 Command Classes in SPI Mode

	Class	Supported Commands																								
CCC	Descrip.	0	1	9	10	12	13	16	17	18	23	24	25	27	28	29	30	35	36	38	42	55	56	58	59
0	Basic	+	+	+	+		+																	+	+
1	NS																								
2	Block read					+		+	+	+	+														
3	NS																								
4	Block write							+			+	+	+	+											
5	Erase																	+	+	+					
6	Write-protect														+	+	+								
7	Lock card																				+				
8	App-specific																					+	+		
9	NS																								
10-11	R																								

Key: CCC = Card CMD Class
NS = Not supported in SPI mode.
R = Reserved

5.17.2 Command Description

Table 5-5 provides a detailed description of the SPI Mode commands. The responses are defined in Section 4-16.

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. Reserved SPI commands are also reserved in MultiMediaCard mode.

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

Table 5-5 SPI Bus Command Description

CMD Index	SPI Mode	Argument	Resp	Abbreviation	Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets MultiMediaCard or RS-MultiMediaCard.
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	---	---	---	---

CMD Index	SPI Mode	Argument	Resp	Abbreviation	Description
CMD8	Reserved				
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its identification (CID).
CMD11	No	---	---	---	---
CMD12	Yes	None	R1	STOP_ TRANSMISSION	Forces 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	Reserved				
CMD15	No	---	---	---	---
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read & write).
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
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 or the requested number of data blocks transmitted.
CMD19	Reserved				
CMD20	No	---	---	---	---
CMD21 … CMD23	Reserved				
CMD24	Yes	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPLE_ BLOCK	Continuously writes blocks of data until a stop transmission token is sent or the requested number of blocks is received.
CMD26	No	---	---	---	---
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	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. Write protection properties are coded in the card-specific

CMD Index	SPI Mode	Argument	Resp	Abbreviation	Description
					data (WP_GRP_SIZE).
CMD29	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	Yes	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	Yes	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selected erase group, or the address of a single sector to be selected for erase.
CMD34	Yes	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	Yes	[31:0] data address	R1	TAG_ERASE_GROUP _START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	Yes	[31:0] data address	R1	TAG_ERASE_GROUP _END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	Yes	[31:0] data address	R1	UNTAG_ERASE_ GROUP	Removes one previously selected erase group from the erase selection.
CMD38	Yes	[31:0] stuff bits	R1b	ERASE	Erases all previously selected sectors.
CMD39	No	---	---	---	---
CMD40	No	---	---	---	---
CMD41	Reserved				
CMD42	Yes	[31:0] stuff bits	R1b	LOCK_UNLOCK	Set/resets the password or lock/unlock the card. The size of the Data Block is defined by the SET_BLOCK_LEN command.
CMD43 … CMD54	Reserved				
CMD55	Yes	This optional MMCA command is not supported in the SanDisk MultiMediaCard/RS-MultiMediaCard.			
CMD56	Yes	This optional MMCA command is not supported in the SanDisk MultiMediaCard/RS-MultiMediaCard.			
CMD57	Reserved				

CMD Index	SPI Mode	Argument	Resp	Abbreviation	Description
CMD58	Yes	None	R3	READ_OCR	Reads the OCR Register of a card.
CMD59	Yes	[31:1] stuff bits, [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A “1” in the CRC option bit will turn the option on; a “0” will turn it off.
CMD60-CMD63	No	---	---	---	---

5.18 Responses

There are several types of response tokens and all are transmitted MSB first.

5.18.1 R1 Format

The card sends this response token after every command with the exception of SEND_STATUS commands. It is one-byte long, the MSB is always set to zero, and the other bits are error indications (1= error).

The structure of the R1 format is shown in Figure 5-13 and the error definitions are listed in Table 5-6.

Figure 5-13 R1 Response Format

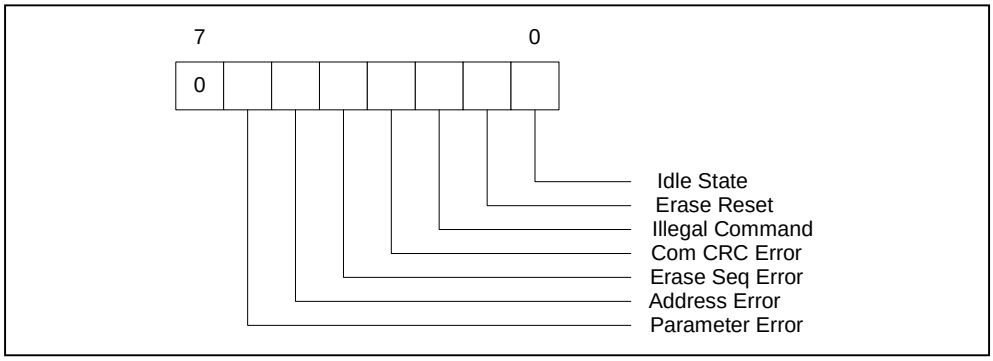


Table 5-6 R1 Response

Error Indication	Definition
Idle State	The card is in an idle state and running initializing process.
Erase reset	An erase sequence was cleared before execution 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 that 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.

5.18.2 R1b Format

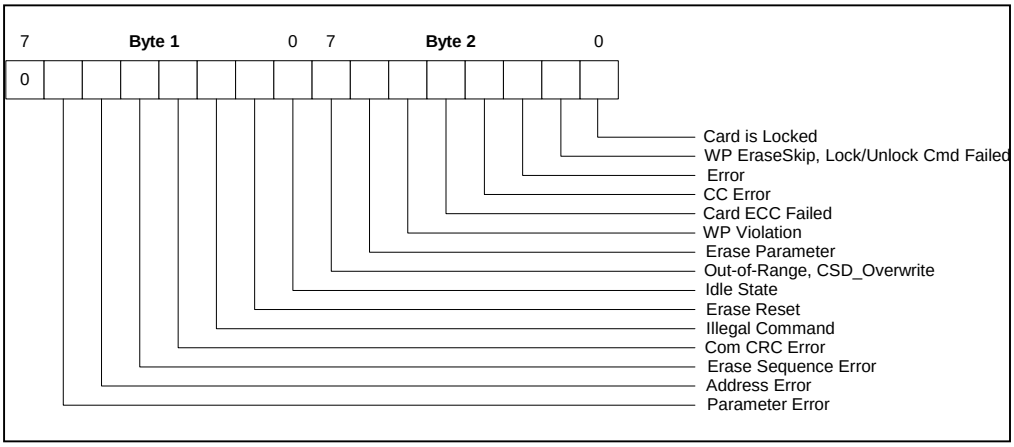
This response token is identical to R1 format with the optional addition of the busy signal.

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.

5.18.3 R2 Format

The card sends the two-byte-long response token a response to the SEND_STATUS command. The format of the R2 status is shown in Figure 5-15.

Figure 5-15 R2 Response Format



The first byte is identical to response R1. The content of the second byte is defined in Table 5-7.

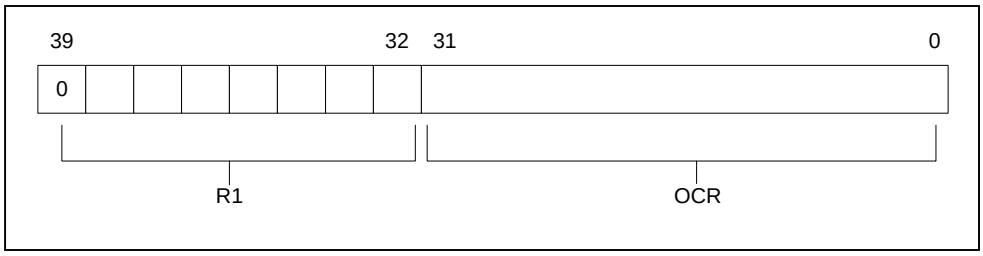
Table 5-7 R2 Response

Error Indication	Definition
Out of range/CSD overwrite	This status bit has two functions. It is set if the command argument was out of its valid range, or if the host is trying to change the ROM section or reverse the copy bit (set as original) or permanent WP bit (un-protect) of the CSD register.
Erase parameter	An invalid selection, sectors, or groups for erase.
Write protect violation	The command tried to write to a write-protected block.
Card ECC failed	The card's 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	This status bit has two functions. It is set when the host attempts to erase a write-protected sector or if a sequence or password error occurred during a card lock/unlock operation.
Card is locked	This bit is set when the user locks the card. It is reset when it is unlocked.

5.18.4 R3 Format

The card sends this response token when an READ_OCR command is received. The response length is five bytes. The structure of the first byte (MSB) is identical to response type R1. The other four bytes contain the OCR Register.

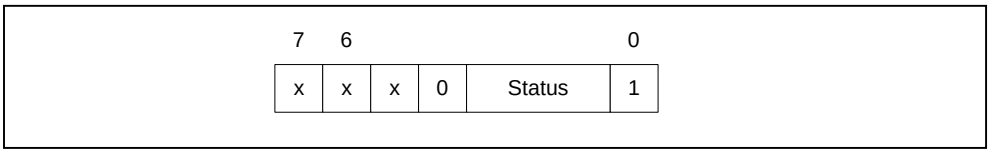
Figure 5-17 R3 Response Format



5.18.5 Data Response

Every data block written to the card is acknowledged by a data response token. The token is one byte long and has the format shown in Figure 5-18.

Figure 5-18 Data Response Format



The meaning of the status bits is defined as follows:

- 010 —Data accepted.
- 101 —Data rejected due to a CRC error.
- 110 —Data rejected due to a write error

In case of any error, CRC or Write, during a Write Multiple Block operation, the host will abort the operation using the Stop Transmission token. In case of a write error, the host may send CMD13 (SEND_STATUS) in order to discover the cause of the write problem.

5.19 Data Tokens

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

Data tokens are 4 to (N+3) bytes long[3] and have the following format:

- First byte: Start Block.
- Bytes 2-(N+1): User data.
- Last two bytes: 16-bit CRC.

[3] N = the data block length set by the SET_BLOCK_LENGTH command

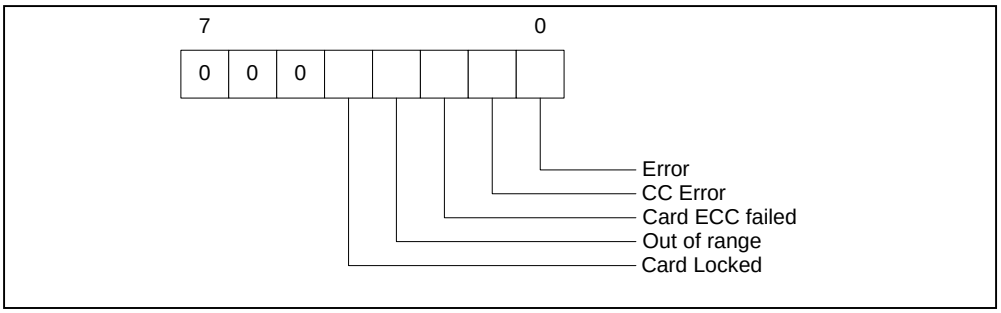
Table 5-8 Start Data Block Token Format

Token Type	Transaction Type	7			Bit Position				0
Start Block	Single Block Read	1	1	1	1	1	1	1	0
Start Block	Multiple Block Read	1	1	1	1	1	1	1	0
Start Block	Single Block Write	1	1	1	1	1	1	1	0
Start Block	Multiple Block Write	1	1	1	1	1	1	0	0
Stop Tran	Multiple Block Write	1	1	1	1	1	1	0	1

5.20 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 5-19.

Figure 5-19 Data Error Token



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

5.21 Clearing Status Bits

In SPI Mode, status bits are reported to the host in three different formats: response R1, response R2, and data-error token.[4] Similar to MultiMediaCard Mode, error bits are cleared when read by the host, regardless of the response format.

Table 5-9 SPI Mode Status Bits

Identifier	Inc in Resp.	Type	Value	Description	Clear Cond.
Out of range	R2 DataErr	E R X	0 = no error 1 = error	Command argument was out of the allowed range for this card.	C
Address error	R1 R2	E R X	0 = no error 1 = error	A misaligned address that did not match the block length was used in the command.	C
Erase sequence error	R1 R2	E R	0 = no error 1 = error	An error in the sequence of erase commands occurred.	C
Erase parameter	R2	E X	0 = no error 1 = error	An error in the parameters of the erase command sequence occurred.	C

[4] The same bits may exist in multiple response types—e.g., Card ECC failed.

Identifier	Inc in Resp.	Type	Value	Description	Clear Cond.
Parameter error	R1 R2	E R X	0 = no error 1 = error	An error in the parameter of the command	C
WP violation	R2	E R X	0 = not protected 1 = protected	Attempt to program a write-protected block.	C
Com CRC error	R1 R2	E R	0 = no error 1 = error	The CRC check of the previous command failed.	C
Illegal command	R1 R2	E R	0 = no error 1 = error	Command not legal for the card state	C
Card ECC failed	R2 DataEr	E X	0 = success 1 = failure	Card internal ECC was applied but failed to correct the data.	C
CC error	R2 DataEr	E R X	0 = no error 1 = error	Internal card controller error	C
Error	R2 DataEr	E R X	0 = no error 1 = error	A general or an unknown error occurred during the operation	C
WP erase skip	R2	S X	0 = not protected 1 = protected	Only partial address space was erased due to existing write protected blocks	C
Lock/unlock CMD failed	R2	E X	0 = no error 1 = error	Sequence or password error during card lock/unlock operation	C
Card is locked	R2 DataEr	S X	0 = card is not locked 1 = card is locked	Card is locked by a user password	A
Erase reset	R1 R2	S R	0 = cleared 1 = set	An erase sequence was cleared before executing because an out of erase sequence command was received.	C
in Idle state	R1 R2	S R	0 = card ready 1 = card in idle state	The card enters the idle state after a power up or reset command. It will exit this state and become ready upon completion of its initialization procedures.	A
CSD overwrite	R2	E X	0 = no error 1 = error	The host is trying to change the ROM section, or is trying to reverse the copy bit (set as original) or permanent WP bit (un-protect) of the CSD register.	C

5.22 Card Registers

In SPI Mode, only the OCR, CSD and CID registers are accessible. Although the registers' format is identical to those in MultiMediaCard Mode, a few fields are irrelevant in SPI Mode.

5.23 SPI Bus Timing Diagrams

All timing diagrams use the schematics and abbreviations listed in Table 5-10.

Table 5-10 SPI Bus Timing Abbreviations

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

The host must keep the clock running for at least N_{CR} clock cycles after the card response is received. This restriction applies to command and data response tokens.

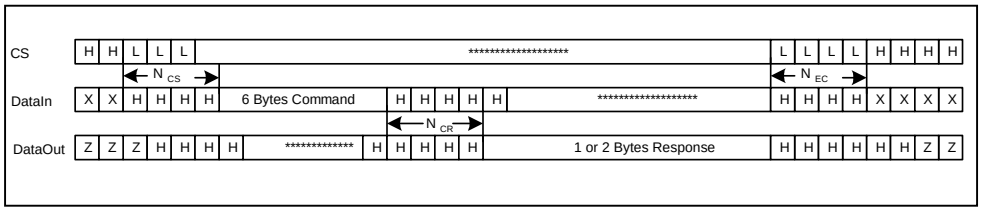
5.23.1 Command and Response

This section provides valuable information on commands and responses.

Host Command to Card Response—Card is Ready

Figure 5-20 describes the basic command response (no data) in an SPI transaction.

Figure 5-20 Host Command to Card Response—Card is Ready

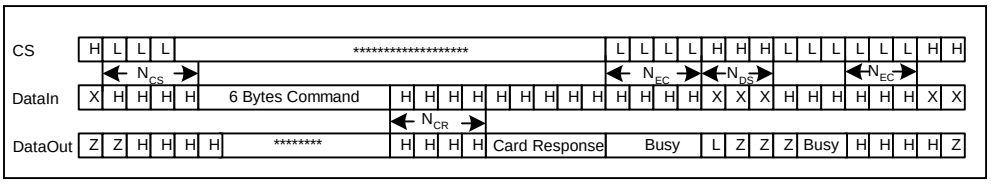


Host Command to Card Response--Card is Busy

The timing diagram in Figure 5-21 illustrates the command response transaction for commands when the card response is of type R1b—for example, SET_WRITE_PROT and ERASE.

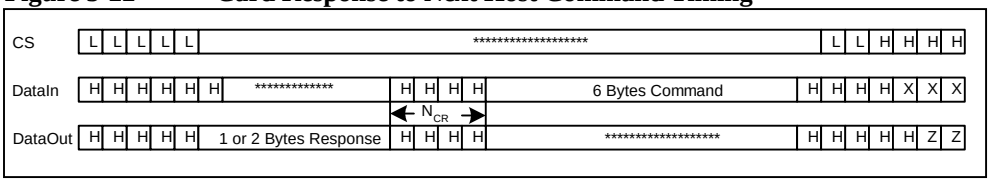
When the card is signaling busy, the host may de-select it by raising the CS at any time. The card will release the DataOut line one clock after CS goes high. To check if the card is still busy, it needs to be re-selected by asserting the CS signal (set to low). The card will resume the busy signal, pulling DataOut low, one clock after the falling edge of CS.

Figure 5-21 Command Response Transaction Timing, Card is Busy



Card Response to Host Command

Figure 5-22 Card Response to Next Host Command Timing

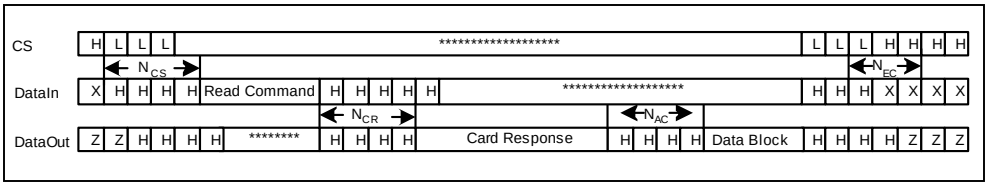


5.23.2 Data Read

This section provides valuable information on the Data Read function.

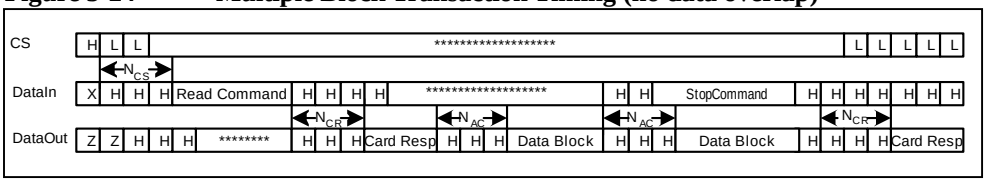
Card Response to Host Command

Figure 5-23 Single Block Read Transaction Timing



Multiple Block Read— Stop transmission is sent between blocks

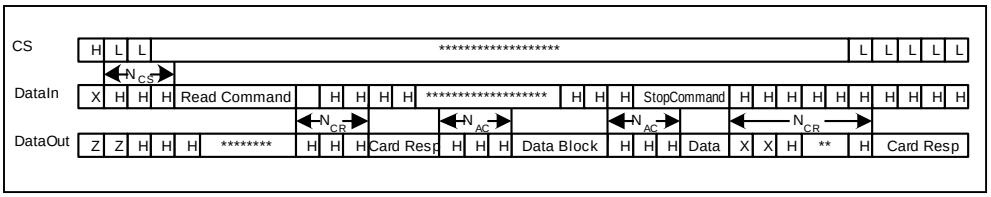
Figure 5-24 Multiple Block Transaction Timing (no data overlap)



The timing for de-asserting the CS signal after the last card response is identical to a standard command/response transaction.

Multiple Block Read—Stop transmission is sent within a block

Figure 5-25 Multiple Block Transaction (data overlap)



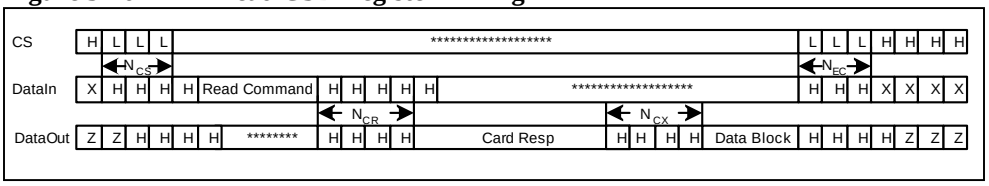
The Stop Transmission command may be sent asynchronously to the data transmitted out of the card and may overlap the data block. In this case the card will stop sending the data and transmit the response token as well. The delay between command and response is standard N_{CR} clocks. The first byte, however, is not guaranteed to be all set to '1.' The card is allowed up to two clocks to stop data transmission.

The timing for de-asserting the CS signal after the last card response is identical to a standard command/response transaction.

Reading the CSD Register

The following timing diagram describes the SEND_CSD command bus transaction. The timeout values between the response and the data block are N_{CX} because the N_{AC} remains unknown.

Figure 5-26 Read CSD Register Timing



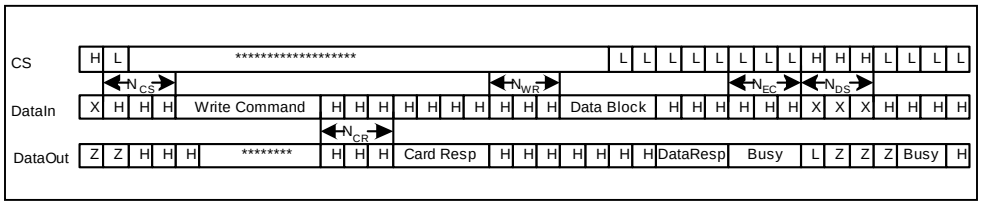
5.23.3 Data Write

This section provides valuable information on the Data Write function.

Single Block Write

The host may de-select 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 the CS signal (set to low). The card will resume the busy signal (pulling DataOut low) one clock cycle after the falling edge of CS.

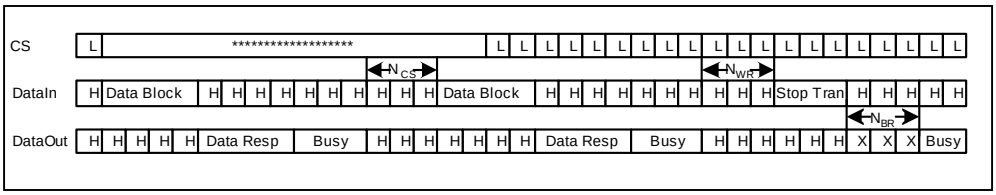
Figure 5-27 Single Block Write Timing



Multiple Block Write

The timing of the multiple block write transaction starting from the command up to the first data block is identical to the single block write. The figure below describes the timing between the data blocks of a multiple block write transaction. Timing of the 'Stop Tran' token is identical to a standard data block. After the card receives the Stop Transmission token, the data on the DataOut line is undefined for one byte (NBR), after which a Busy token may appear. The host may de-select and re-select the card during every busy period between the data blocks. Timing for toggling the CS signal is identical to the Single block write transaction.

Figure 5-28 Multiple Block Write Timing



5.24 Timing Values

Table 5-11 shows the timing values and definitions.

Table 5-11 Timing Constants Definitions

Value	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	[10*((TAAC*f)+(100*NSAC))]*1/8*	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

*Where *f* is the clock frequency.

5.25 SPI Electrical Interface

The SPI Electrical Interface is identical to MultiMediaCard mode with the exception of the programmable card output-drivers option, which is not supported in SPI mode.

5.26 SPI Bus Operating Conditions

SPI Bus operating conditions are identical to MultiMediaCard mode.

5.27 SPI Bus Timing

SPI Bus timing is identical to MultiMediaCard mode. The timing of the CS signal is the same as any other card input.

Appendix A Ordering Information

A.1 MultiMediaCard and RS-MultiMediaCard

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

Part Number	Form Factor	Capacity
SDMJ-32	Full-size MultiMediaCard	32 MB
SDMJ-64	Full-size MultiMediaCard	64 MB
SDMJ-128	Full-size MultiMediaCard	128 MB
SDMJ-256	Full-size MultiMediaCard	256 MB
SDMJ-512	Full-size MultiMediaCard	512 MB
SDMJ-1024	Full-size MultiMediaCard	1 GB
SDMRJ-32	Reduced-size MultiMediaCard	32 MB
SDMRJ-64	Reduced-size MultiMediaCard	64 MB
SDMRJ-128	Reduced-size MultiMediaCard	128 MB
SDMRJ-256	Reduced-size MultiMediaCard	256 MB
SDMRJ-512	Reduced-size MultiMediaCard	512 MB
SDMRJ-1024	Reduced-size MultiMediaCard	1 GB

Appendix B 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 Offices

Northwest, 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

International Industrial/OEM Sales Offices

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 & 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
Suite 902-903
Bank of East Asia Harbour View Centre
56 Gloucester Road, Wanchai
Hong Kong
Tel: 852-2712-0501
Fax: 852-2712-9385

Appendix C 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.

II. GENERAL PROVISIONS

This warranty sets forth the full extent of SanDisk's responsibilities regarding the SanDisk MultiMediaCard. In satisfaction of its obligations hereunder, SanDisk, at its sole option, will 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. Accordingly, in any use of products in life support systems or other applications where failure could cause injury or loss of life, the products should only be incorporated in systems designed with appropriate redundancy, fault tolerant or back-up features.

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. A warranty registration card for each defective product was submitted and is on file at SanDisk. If not, a warranty registration card must accompany each returned defective product. This card is included in each product's original retail package.

B. The defective product is returned to SanDisk for failure analysis as soon as possible after the failure occurs.

C. An incident card filled out by the user, explaining the conditions of usage and the nature of the failure, accompanies each returned defective product.

D. 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, serial numbers, 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 D Disclaimer of Liability

D.1 SanDisk Corporation Policy

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