MultiMediaCard
Product Manual

This manual covers the SanDisk MultiMediaCard which was developed by SanDisk's Design Center located in Tefen, Israel. The MultiMediaCard supports version 1.4 of the MultiMediaCard Specification.

SanDisk®
CORPORATE HEADQUARTERS
140 Caspian Court
Sunnyvale, CA 94089-9820
408-542-0500
FAX: 408-542-0503
URL: http://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.

The information in this manual is subject to change without notice.

SanDisk Corporation shall not be liable for technical or editorial errors or omissions contained herein; nor for incidental or consequential damages resulting from the furnishing, performance, or use of this material.

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

Lit. No. 80-13-00089 Rev 1 4/99 *Printed in U.S.A.*

Revision History

- *Revisions dated before 1/98—initial release and general changes.*
- *Revision dated 1/98—general editorial changes, manual reorganized, technical changes to reflect support of MultiMediaCard Specification version 1.3, new timing diagrams added. Pin 6 definition changed in SPI mode from SPI select to VSS2 (supply voltage ground).*
- *Revision dated 4/98— changes reflect support of MultiMediaCard Specification version 1.4, updated timing for Multiple Write with no Busy, updated SPI command class definition, added Error Protection section, changed operating temperature specification to -25° to 85°C.*
- *Revision dated 4/28/98—Updated C_SIZE and C_SIZE_MULT field definitions.*
- *Revision 1 dated 4/99—Added 32 MB MultiMediaCard, general technical and editorial changes, added power up section.*

Table of Contents

Table of Contents (continued)

Table of Contents (continued)

Table of Contents (continued)

1.0 Introduction to the MultiMediaCard

The SanDisk MultiMediaCard is a very small, removable flash storage device, 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 is well suited to meet the needs of small, low power, electronic devices. With a form factor of 32mm by 24mm and 1.4mm thick, MultiMediaCards are expected to be used in a wide variety of portable devices like mobile phones, pagers and voice recorders. This ultra-small form factor is part of a new, emerging, proposed open standard.

To support this wide range of applications, the MultiMediaCard protocol, a high performance 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 MultiMediaCard interface allows for easy integration into any design, regardless of microprocessor used. For compatibility with existing controllers, the MultiMediaCard offers, in addition to the MultiMediaCard interface, an alternate communication protocol which is based on the SPI standard.

The MultiMediaCard provides up to 32 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 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.

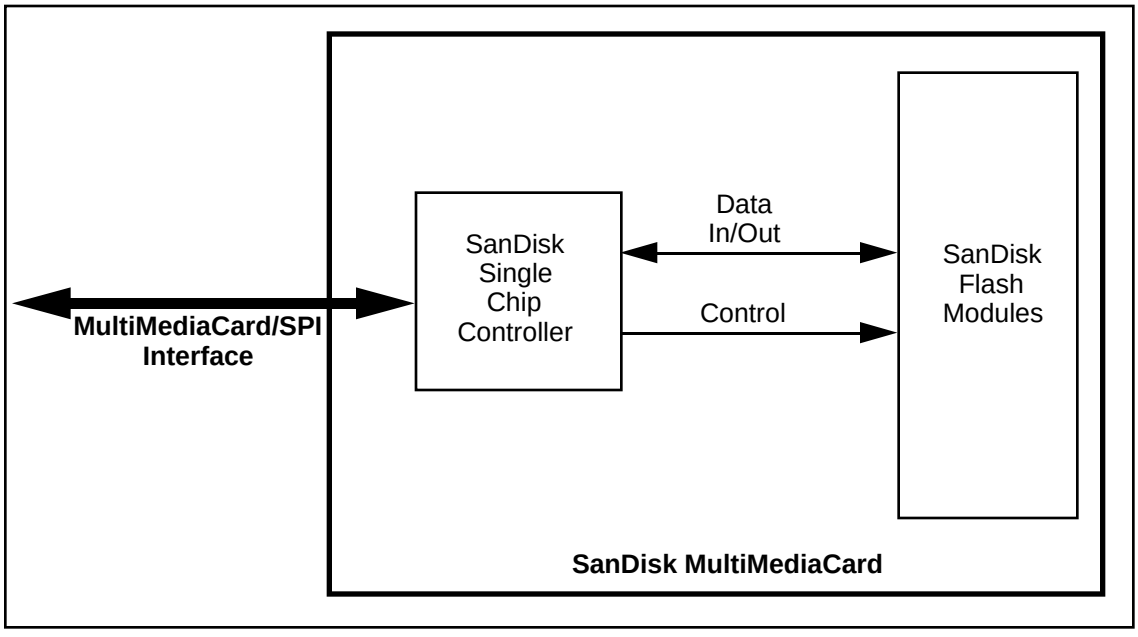


Figure 1-1 MultiMediaCard Block Diagram

1.1 Scope

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

1.2 Product Models

The MultiMediaCard is available in the capacities shown in the following table:

Table 1-1 MultiMediaCard Capacities

Model No.	Capacities
SDMB-2	2.0 MB
SDMB-4	4.0 MB
SDMB-8	8.0 MB
SDMB-16*	16.0 MB
SDMB-32*	32.1 MB

* Preliminary information.

1.3 System Features

- Up to 32 Mbytes of data storage
- MultiMediaCard protocol compatible
- Supports SPI Mode
- Targeted for portable and stationary applications
- Voltage range — Communication: 2.0 - 3.6V, Memory Access: 2.7 - 3.6V
- Maximum data rate with up to 10 cards
- Correction of memory field errors
- Built-in write protection features (permanent and temporary)
- Comfortable erase mechanism
- Variable clock rate 0 - 20 Mhz
- Multiple cards stackable on a single physical bus

The performance of the communication channel is described in the table below:

Table 1-2 MultiMediaCard/SPI Comparison

MultiMediaCard	SPI
Three-wire serial data bus (Clock, command, data)	Three-wire serial data bus (Clock, dataIn, dataOut) + card specific CS signal.
Up to 64k cards addressable by the bus protocol	Card selection via a hardware CS signal
Easy card identification	Not available
Error-protected data transfer	Optional. A non protected data transfer mode is available.
Sequential and single/multiple block oriented data transfer	Single block read/write

1.4 MultiMediaCard Standard

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

The MultiMediaCard System Specification Version 1.4

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

1.5 Functional Description

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

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

1.5.1 Flash Technology Independence

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

1.5.2 Defect and Error Management

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

The MultiMediaCard's soft error rate specification is much better than the magnetic disk drive specification. In the extremely rare case a read error does occur, MultiMediaCards have innovative algorithms to recover the data. This is similar to using retries on a disk drive but is much more sophisticated. The last line of defense is to employ a powerful ECC to correct the data. If ECC is used to recover data, defective bits are replaced with spare bits to ensure they do not cause any future problems.

These defect and error management systems coupled with the solid-state construction give MultiMediaCards unparalleled reliability.

1.5.3 Endurance

SanDisk MultiMediaCards have an endurance specification for each sector of 300,000 writes (reading a logical sector is unlimited). This is far beyond what is needed in nearly all applications of MultiMediaCards. Even very heavy use of the MultiMediaCard in cellular phones, personal communicators, pagers and voice recorders will use only a fraction of the total endurance over the typical device's five year lifetime. For instance, it would take over 34 years to wear out an area on the MultiMediaCard on which a file of any size (from 512 bytes to capacity) was rewritten 3 times per hour, 8 hours a day, 365 days per year.

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

1.5.4 Wear Leveling

SanDisk MultiMediaCards do not require or perform a Wear Level operation.

1.5.5 Using the Erase Command

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

1.5.5.1 Limitations and Issues

The advantage of the Erase command is that it shifts the bulk of the erase and write time to the Erase command. The Erase command performs most of the normal tasks needed. To increase the speed of the Write command, the final margin verify done in a normal write command is skipped for the first 16K writes. When the cycle count (hot count) of a sector exceeds 16K, the system controller automatically reverts to a full write, including the final margin verify. Since the erase is not required in this case, a write to a pre-erased sector with a hot count of over 16K is still faster than to a sector that has not been pre-erased.

1.5.6 Automatic Sleep Mode

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

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

1.5.7 Hot Insertion

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

Since the MultiMediaCard can retain data and information at very low current levels (sleep mode), voltage applied through the card's clock and Data lines can provide enough current to retain card information. This means that simply removing VDD power (pin 4) may keep the card's last state intact if the CLK (pin 5), CMD (pin 2), DAT (pin 7) or CS (pin 1) are still being sourced.

1.5.8 MultiMediaCard Mode

1.5.8.1 MultiMediaCard Standard Compliance

The MultiMediaCard is fully compliant with MultiMediaCard Standard Specification V1.4. The structure of the Card Specific Data (CSD) register is compliant with CSD structure V1.4.

1.5.8.2 Negotiating Operation Conditions

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

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

1.5.8.3 Card Acquisition and Identification

The MultiMediaCard bus is a single master (MultiMediaCard host) and multi-slaves (cards) bus. The host can query the bus and find out how many cards of which type are currently connected. The MultiMediaCard's CID register is pre-programmed with a unique card identification number which is used during the acquisition and identification procedure.

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

1.5.8.4 Card Status

MultiMediaCard status is stored in a 32 bit status register which is sent as the data field in the card respond to host commands. Status register provides information about the card's current state and completion codes for the last host command.

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

1.5.8.5 Memory Array Partitioning

Although the MultiMediaCard memory space is byte addressable with addresses ranging from 0 to the last byte, it is not a simple byte array but 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 sectors. Any combination of sectors within one group or, any combination of erase groups can be erased in a single erase command. A write command implicitly erases the memory before writing new data into it. Explicit erase command can be used for pre-erasing of memory which will 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 number of various memory structures, for the different MultiMediaCards are summarized in Table 1-3. The last (highest in address) WPG will be smaller and contain less than 32 erase groups.

Table 1-3 Memory Array Structures Summary

Structure	SDMB-2	SDMB-4	SDMB-8	SDMB-16*	SDMB-32*
Bytes	2.03 MB	4.03 MB	8.03 MB	16.06 MB	32.11 MB
Sector	3,968	7,872	15,680	31,360	62,720
Erase Group	248	492	980	1,960	3,920
WPG	8	16	31	62	123

* Preliminary information.
Note: All measurements are in units per card.

1.5.8.6 Read and Write Operations

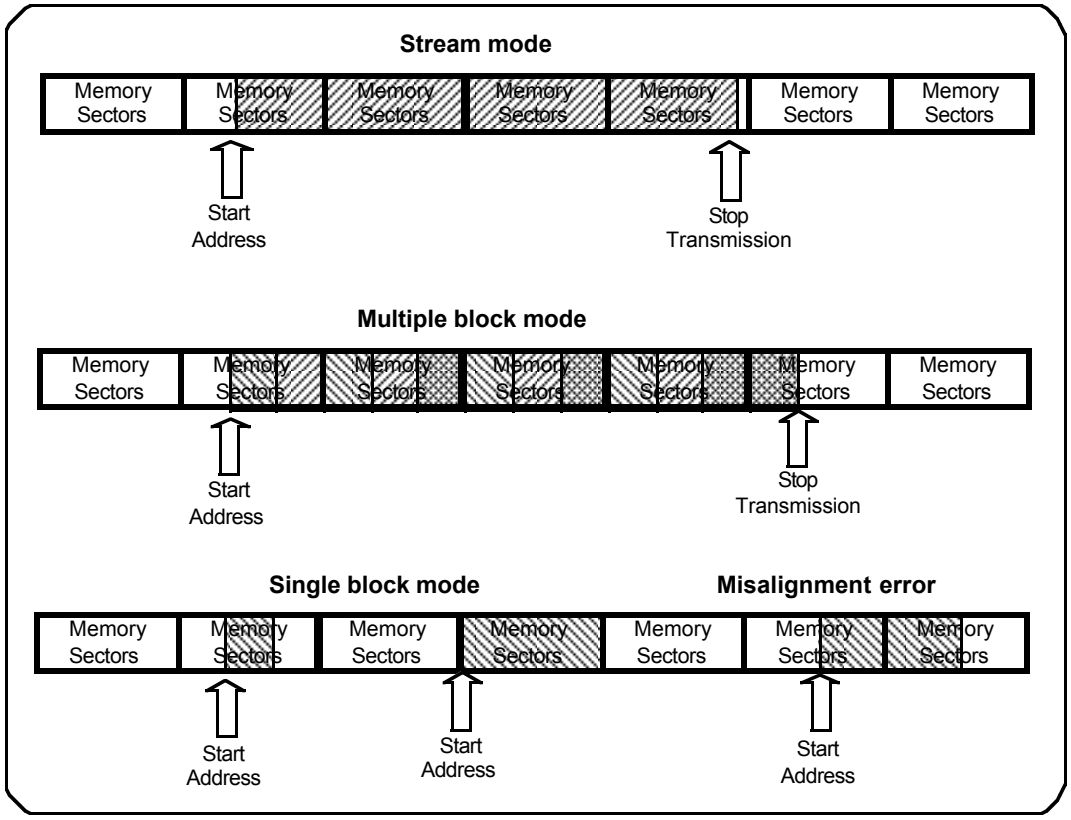


Figure 1-3 Data Transfer Formats

The MultiMediaCard supports three read/write modes as shown in the above figure.

Stream Mode
In stream mode the host reads or writes continues stream of data. The starting address is specified in the read/write command and the operation ends when the host sends a stop transmission command.

In this mode there is no validity check on the transferred data.

The start address for a read operation can be any random byte address in the valid address space of the memory card. For a write operation, the start address must be sector aligned and the data length must be an integer multiplication of the sector length.

Single Block Mode
In this mode the host reads or writes one data block in a pre-specified length. The data block transmission is protected with 16 bit CRC which

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 but can be as small as a single byte. If a partial block is allowed, any size from one byte to the maximum block size as defined in the card's CSD, can be used. Otherwise, the block size has to be the same as the maximum block size as defined in the card's CSD. A block may not be larger than the maximum size as defined in the card's CSD. Misalignment is not allowed. Every data block must be contained in a single physical sector.

The block length for write operations must be identical to the sector size and the start address aligned to a sector boundary.

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

The operation is terminated with a stop transmission command.

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

1.5.8.7 Data Protection in the Flash Card

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

The MultiMediaCard can be considered error free and no additional data protection is needed. However, if an application uses additional, external, ECC protection, the data organization is defined in the user writeable section of the CSD register.

1.5.8.8 Erase

The smallest erasable unit in the MultiMediaCard is a sector. In order to speed up the erase procedure, multiple sectors can be erased in the same time. The erase operation is divided into two stages:

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

Erasing - Starting the Erase Process
The sectors are grouped into erase groups of 16 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, but if a sector, all selected sectors must lie within the same erase group. Tagging and erasing sectors must follow a strict command sequence.

1.5.8.9 Write Protection

The MultiMediaCard erase groups are grouped into write protection groups. Commands are provided for limiting and enabling write and erase privileges for each group individually. The current write protect map can be read using the SEND_WRITE_PROT command.

In addition two, permanent and temporary, card level write protection options are available. Both can be set using the PROGRAM_CSD command (see below). The permanent write protect bit, once set, cannot be cleared.

The One Time Programmable (OTP) characteristic of the permanent write protect bit is implemented in the MultiMediaCard controller firmware and not with a physical OTP cell.

1.5.8.10 Copy Bit

The content of a MultiMediaCard can be marked as an original or a copy using the copy bit in the CSD register. Once the Copy bit is set (marked as a copy) it cannot be cleared. The Copy bit of the MultiMediaCard is programmed (during test and formatting on the manufacturing floor) as a copy. The MultiMediaCard can be purchased with the copy bit set (copy) or cleared, indicating the card is a master.

The One Time Programmable (OTP) characteristic of the Copy bit is implemented in the MultiMediaCard controller firmware and not with a physical OTP cell.

1.5.8.11 The CSD Register

All the configuration information of the MultiMediaCard is stored in the CSD register. The MSB bytes of the register contain manufacturer data and the two least significant bytes contains the host controlled data - the card Copy and write protection and the user ECC register.

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

1.5.9 SPI Mode

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

1.5.9.1 Negotiating Operation Conditions

The operating condition negotiation function of the MultiMediaCard bus is not supported in SPI mode. The host must work within the valid voltage range (2.7 to 3.6) volts of the card.

1.5.9.2 Card Acquisition and Identification

The card acquisition and identification function of the MultiMediaCard bus is not supported in SPI mode. The host must know the number of cards currently connected on the bus. Specific card selection is done via the CS signal.

1.5.9.3 Card Status

In SPI mode only 16 bits (containing the errors relevant to SPI mode) can be read out of the MultiMediaCard status register.

1.5.9.4 Memory Array Partitioning

Memory partitioning in SPI mode is equivalent to MultiMediaCard mode. All read and write commands are byte addressable.

1.5.9.5 Read and Write Operations

In SPI mode, only single block read/write mode is supported.

1.5.9.6 Data Transfer Rate

In SPI mode only block mode is supported. The maximum clock frequency is 5MHZ. The typical access time (latency) for each data block, in read operation, is 1.5mS. The write block operation is done in handshake mode. The card will keep DataOut line low as long as the write operation is in progress and there are no write buffers available.

1.5.9.7 Data Protection in the MultiMediaCard

Same as for the MultiMediaCard mode.

1.5.9.8 Erase

Same as in MultiMediaCard mode.

1.5.9.9 Write Protection

Same as in MultiMediaCard mode.

2.0 Product Specifications

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

2.1 System Environmental Specifications

Temperature	Operating: Non-Operating:	-25° C to 85° C -40° C to 85° C
Humidity	Operating: Non-Operating:	8% to 95%, non-condensing 8% to 95%, non-condensing
Acoustic Noise:		0 dB
Vibration	Operating: Non-Operating:	15 G peak to peak max. 15 G peak to peak max.
Shock	Operating: Non-Operating:	1,000 G max. 1,000 G max.
Altitude (relative to sea level)	Operating: Non-Operating:	80,000 feet max. 80,000 feet max.

2.2 System Power Requirements

Operation	@ 3.3 V	@ 2.7 V
Read: Write: Standby:	<33 mA <35 mA <50 µA	<23 mA <27 mA <41 µA

2.3 System Performance

Block Read Access Time	1.5 msec
Power-up to Ready	50 msec
Sleep to Read	1 msec
Sleep to Write	1 msec

Notes: 1) All values quoted are typical at ambient temperature and nominal supply voltages unless otherwise stated.
2) All performance assumes the controller is in the default (i.e. fastest) mode.

2.4 System Reliability and Maintenance

MTBF	> 1,000,000 hours
Preventive Maintenance	None
Data Reliability	< 1 non-recoverable error in 10^{14} bits read
Endurance	300,000 write/erase cycles

2.5 Physical Specifications

Refer to the following table and to Figure 2-1 for MultiMediaCard physical specifications and dimensions.

Weight	1.5 g. maximum
Length:	32mm ± 0.1mm
Width:	24mm ± 0.08mm
Thickness:	1.4mm ± 0.1mm

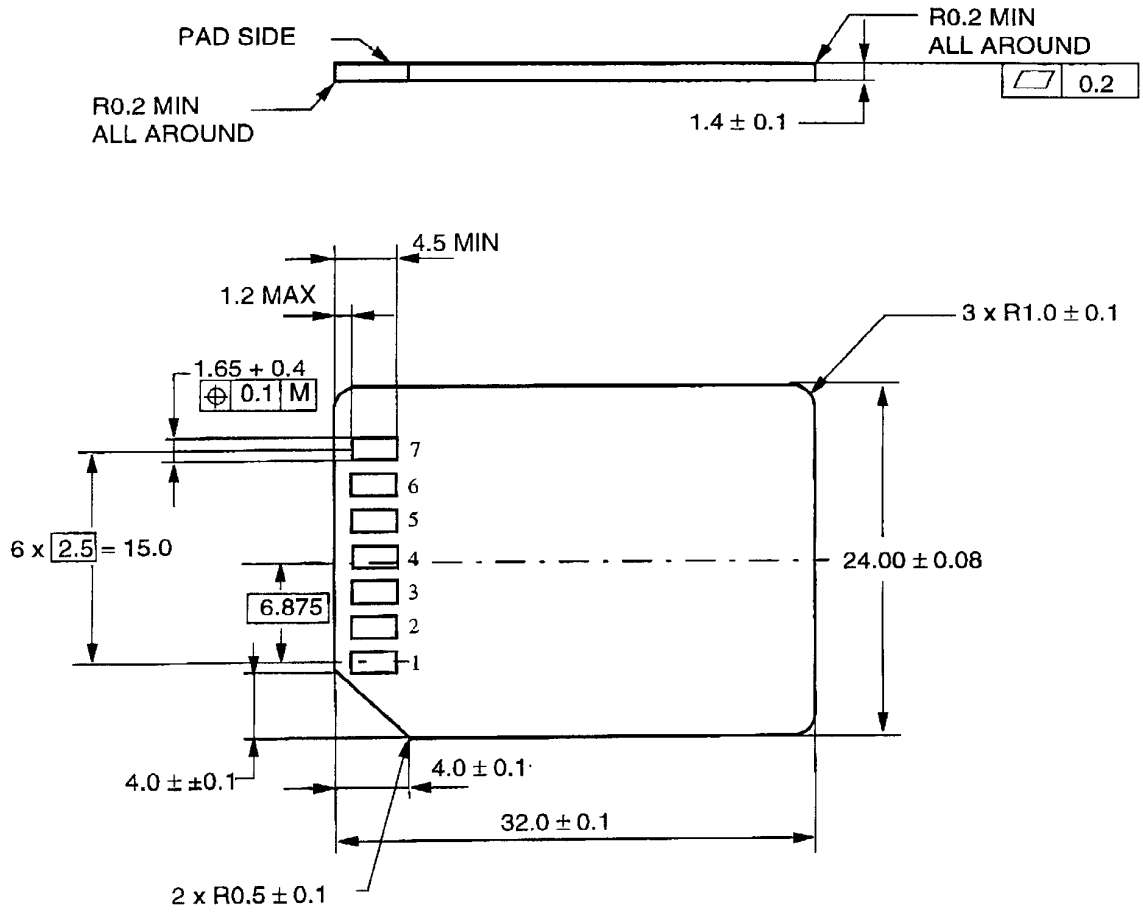


Figure 2-1 MultiMediaCard Dimensions

3.0 Installation

3.1 Mounting

The MultiMediaCard can be installed in any platform that has a MultiMediaCard slot and complies with the MultiMediaCard Standard.

4.0 MultiMediaCard Interface Description

4.1 Physical Description

The MultiMediaCard has seven exposed contacts on one side. (See Figure 2-1.) The host is connected to the MultiMediaCard using a seven pin connector as shown in the Appendix at the end of this manual.

4.1.1 Pin Assignments in MultiMediaCard Mode

Table 4-1 MultiMediaCard Pad Definition

Pin #	Name	Type*	MultiMediaCard Description
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	DAT[0]	I/O/PP	Data 0

*Note: S=power supply; I=input; O=output; PP=push-pull; OD=open-drain; NC=not connected.

4.1.2 Pin Assignments in SPI Mode

Table 4-2 SPI Pad Definition

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

*Note: S=power supply; I=input; O=output.

4.2 MultiMediaCard Bus Topology

The MultiMediaCard bus has three communication lines and four supply lines:

- CMD: Command is a bi-directional signal. Host and card drivers are operating in two modes, open drain and push pull.
- DAT: Data 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.

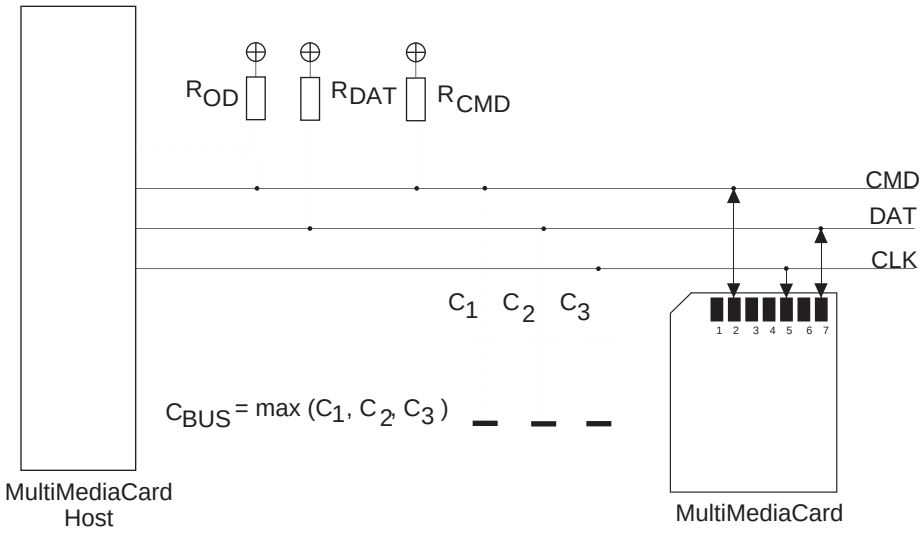


Figure 4-1 Bus Circuitry Diagram

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 the DAT line against bus floating when no card is inserted or when all card drivers are in a hi-impedance mode.

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

- Hot insertion/removal

In general hot insertion and removal is allowed. No card should be damaged by inserting or removing a card into the MultiMediaCard bus even when the power is up.

The data transfer operations are protected by CRC codes, so that bit changes induced by card insertion/remove could be detected by the MultiMediaCard bus master.

- The inserted card must be properly reset also when CLK carries a clock frequency f_{PP}.
- Each card has to have power protection to prevent card (and host) damage.
- Data transfer failures induced by removal/insertion are detected by the bus master. They must be corrected by the application. Correction should be done by command repetition in most cases.

4.2.1 Power Protection

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

Every cards output must also be able to withstand short cuts to either supply.

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

4.2.2 *Power-up*

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

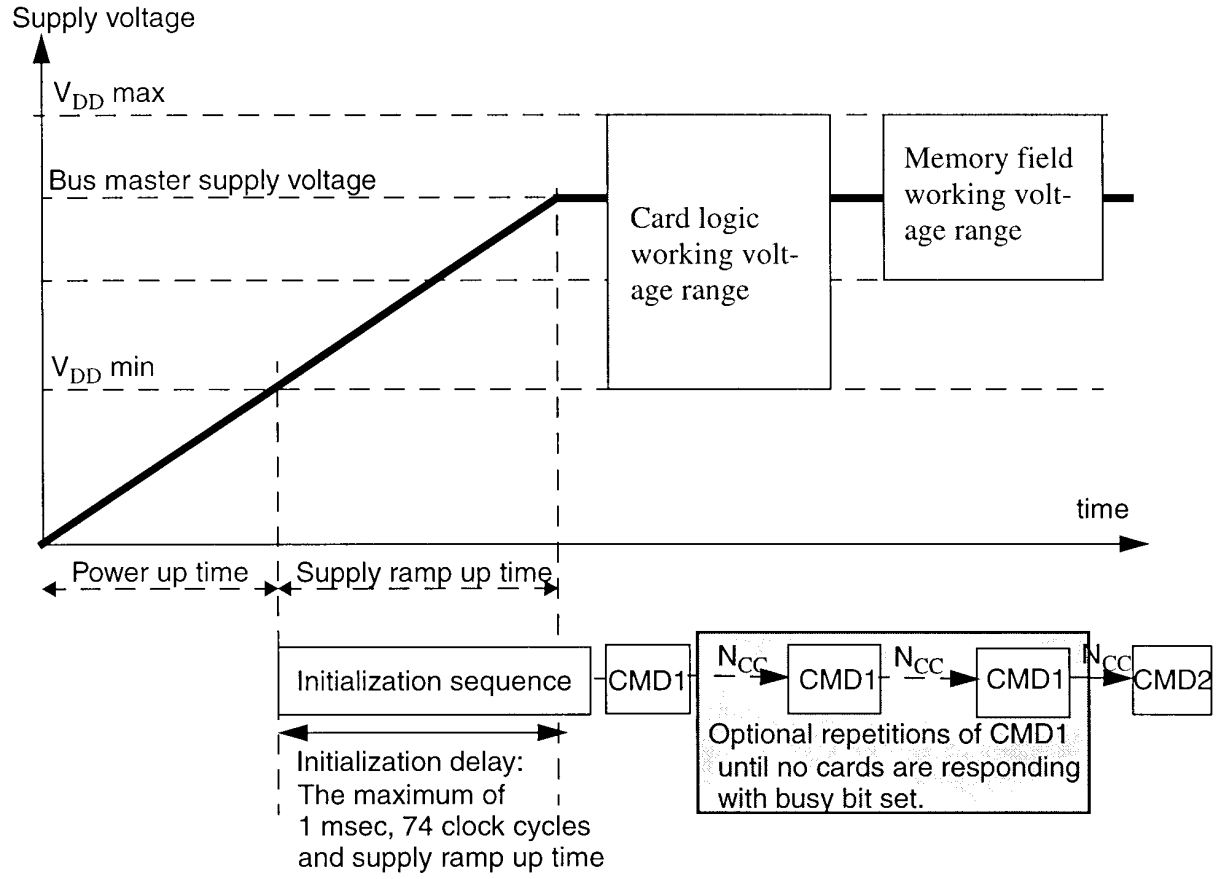


Figure 4-2 Power-up Diagram

After power-up (including hot insertion, that is, inserting a card when the bus is operating), the MultiMediaCard enters the Idle State. During this state, the MultiMediaCard 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 is not ready for identification. This bit informs the host that at least one card is not ready. The host has to wait (and continue to poll the cards) until this bit is cleared.

Getting individual cards, as well as the whole MultiMediaCard system, out of Idle State is up to the responsibility of the bus master. Since the power-up time and the supply ramp up time depend on application parameters such as the maximum number of MultiMediaCards, the bus length and the power supply unit, the host must ensure that the power is built up to the operating level (the same level which will be specified in 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 ones. The sequence length is the maximum of one msec, 74 clocks or the supply ramp up time. The additional ten clocks (beyond the 64 clocks after

which the card should be ready for communication) are provided to eliminate power-up synchronization problems.

Every bus master has to implement CMD1. The implementation is optional for the MultiMediaCard. Cards which support the complete voltage range of the MultiMediaCard bus, and are guaranteed to be ready for identification within 64 clocks from power up may ignore this command. Cards which do not support CMD1 will change into the ready state as soon as their power-up sequence has been finished. The Idle, Ready and Inactive states, however, are mandatory for all MultiMediaCards.

4.2.3 Programmable Card Output Driver

This option, defined in chapter 6 of the MultiMediaCard standard, is not implemented in the SanDisk MultiMediaCard.

4.3 SPI Bus Topology

4.3.1 SPI Interface Concept

The Serial Peripheral Interface (SPI) is a general purpose synchronous serial interface originally found on certain Motorola microcontrollers. A virtually identical interface can now be found on certain TI and SGS Thompsen microcontrollers as well.

The MultiMediaCard SPI interface is compatible with SPI hosts available on the market. As any other SPI device the MultiMediaCard SPI channel consists of the following 4 signals:

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

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

The SPI standard defines the physical link only and not the complete data transfer protocol. The MultiMediaCard uses a subset of the MultiMediaCard protocol and command set.

4.3.2 SPI Bus Topology

The MultiMediaCard identification and addressing algorithms are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. A card (slave) is selected, for every command, by asserting (active low) the CS signal (see Figure 4-3).

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 bidirectional CMD and DAT lines are replaced by unidirectional dataIn and dataOut signals. This eliminates the ability of executing commands while data is being read or written and, therefore, eliminates the sequential and multi block read/write operations. Only single block read/write is supported by the SPI channel.

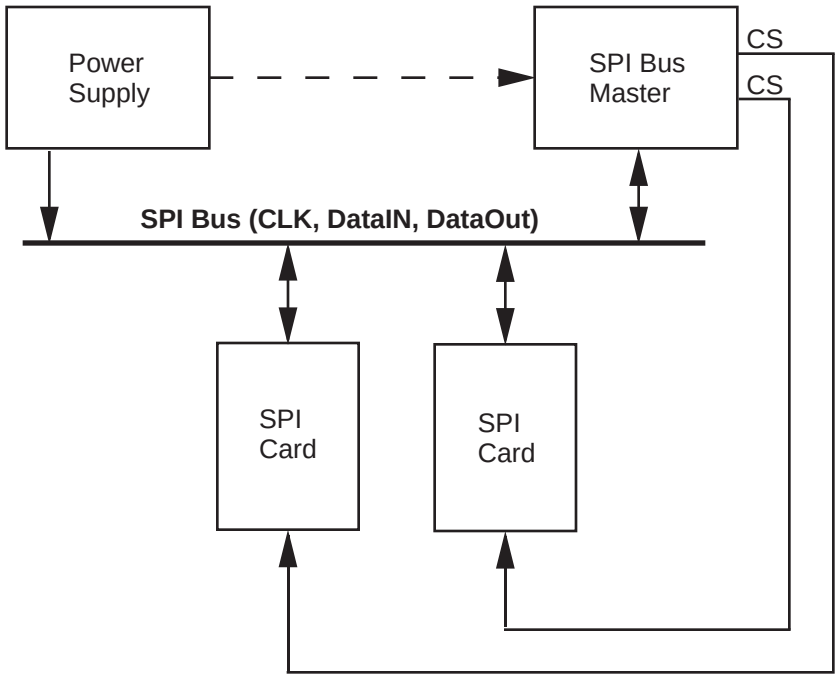


Figure 4-3 MultiMediaCard Bus System

4.4 Electrical Interface

4.4.1 Bus Operating Conditions

General

Parameter	Symbol	Min.	Max.	Unit	Remark
Peak voltage on all lines		-0.5	3.6	V	
All Inputs					
Input Leakage Current		-10	10	µA	
All Outputs					
Output Leakage Current		-10	10	µA	

Power supply voltage

Parameter	Symbol	Min.	Max.	Unit	Remark
Supply voltage	V_{DD}	**2.0**	**3.6**	**V**	
Supply voltage differentials (V_{SS1}, V_{SS2})		**-0.5**	**0.5**	**V**	

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

Bus Signal Line Load

The total capacitance CL of each line of 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 following values must not be exceeded:

Parameter	Symbol	Min.	Max.	Unit	Remark
Pull-up resistance	RCMD RDAT	50	100	kΩ	To prevent bus floating
Bus signal line capacitance	CL		250	pF	fPP # 5 MHz, 30 cards
Bus signal line capacitance	CL		100	pF	fPP # 20 MHz, 10 cards
Single card capacitance	CCARD		7	pF	
Maximum signal line inductance			16	nH	fPP # 20 MHz

4.4.1.2 Bus Signal Levels

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

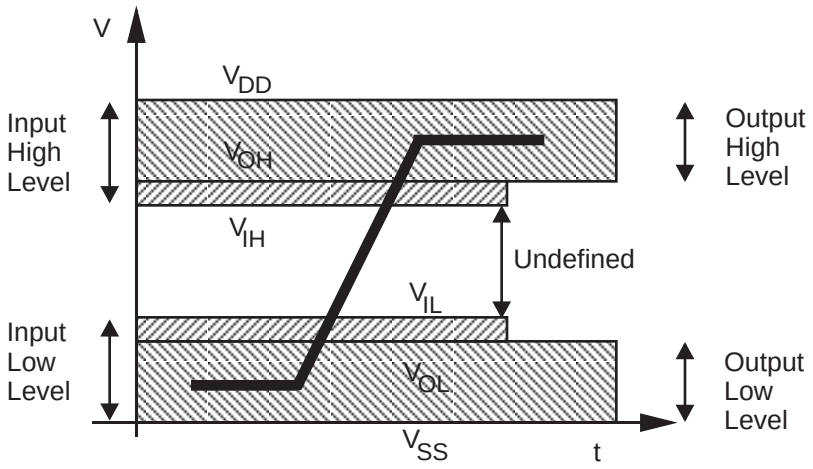


Figure 4-4 Bus Signal Levels

4.4.1.2.1 Open-Drain Mode Bus Signal Level

Parameter	Symbol	Min.	Max.	Unit	Conditions
Output HIGH voltage	VOH	$V_{DD-0.2}$		V	IOH = -100 µA
Output LOW voltage			0.3	V	IOL = 2 mA

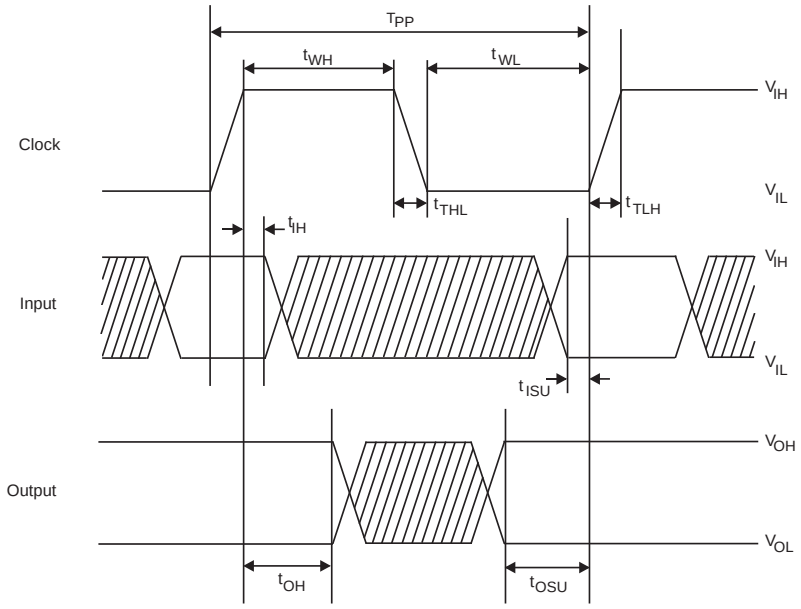
The input levels are identical with the push-pull mode bus signal levels.

4.4.1.2.2 Push-pull Mode Bus Signal Level

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

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

4.4.1.2.3 Bus Timing



Note: Data in the shaded areas is not valid.

Figure 4-5 Timing Diagram Data Input/Output Referenced to Clock

Table 4-3 Bus Timing

Parameter	Symbol	Min.	Max.	Unit	Remark
Clock CLK (All values are referred to min. (VIH) and max. (VIL)					
Clock Frequency Data Transfer Mode (PP)	f_{PP}	0	20	MHz	$C_L \leq 100$ pF (10 cards)
Clock Frequency Identification Mode (OD)	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 set-up time	t_{ISU}	3		ns	
Input hold time	t_{IH}	3		ns	
Outputs CMD, DAT (referenced to CLK)					
Output set-up time	t_{OSU}	5		ns	
Output hold time	t_{OH}	5		ns	

4.4.2 SPI Mode Bus Operating Conditions

SPI Mode bus operating conditions are identical to MultiMediaCard Mode bus operating conditions. The CS (chip select) signal timing is identical to the input signal timing. (See Figure 4-5.)

4.5 MultiMediaCard Registers

4.5.1 Operating Conditions Register (OCR)

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. The MultiMediaCard is capable of communicating (execute the acquisition and identification procedure) with any standard MultiMediaCard host using operating voltages form 2 to 3.6 Volts.

Accessing the data in the memory array, however, requires 2.7 to 3.6 Volts. The OCR shows the voltage range in which the card data can be accessed. The least significant 31 bits are constant and will be set as described in Figure 4-6. Bit 32 is the busy bit as defined in the MultiMediaCard specification document.

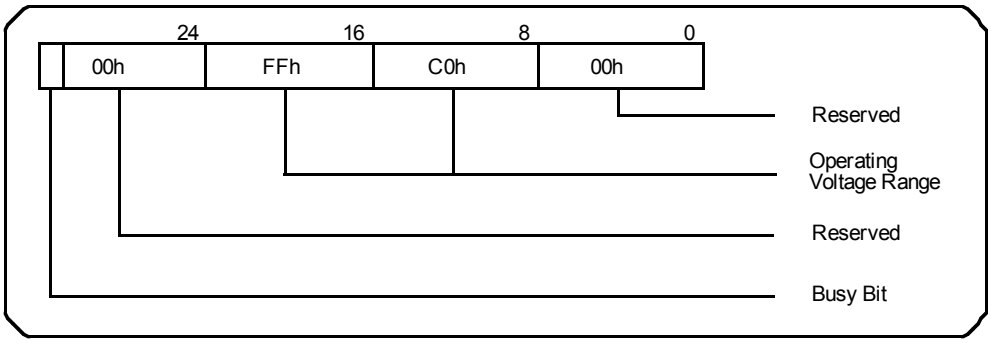


Figure 4-6 OCR Structure

4.5.2 DSR Register

The DSR Register is not implemented in SanDisk MultiMediaCards.

4.5.3 Card Identification (CID) Register

The CID register is 16 bytes long and contains a unique card identification number as shown in the table below. It is programmed during card manufacturing and can not be changed by MultiMediaCard hosts.

Name	Type	Width	CID - Slice
Manufacture ID	Binary	24	[127:104]
Product name	String	56	[103:48]
HW Revision	Binary	4	[47:44]
FW Revision	Binary	4	[43:40]
Serial Number	Binary	24	[39:16]
Month code	Binary	4	[15:12]
Year code	Binary	4	[11:8]
CRC7 checksum*	Binary	7	[7:1]
Not used, always '1'		1	[0:0]

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

4.5.4 CSD Register

The Card Specific Data (CSD) register contains all the configuration information required in order to access the card data.

In the table below, the cell type column defines the CSD field as Read only (R), One Time Programmable (R/W) or erasable (R/W/E). This table shows, for each field, the value in "real world" units and coded according to the CSD structure. The Model dependent column marks (with a check mark—√) the CSD fields which are model dependent.

Table 4-4 CSD Register

Name	Width [bits]	Cell Type	CSD-slice	CSD Value	CSD Code	Model Dep.
CSD Structure	2	R	[127:126]	V1.1	1	
MultiMediaCard Protocol Version	4	R	[125:122]	V1.4	1	
Reserved	2	R	[121:120]	0	0	
Data Read Access-Time-1	8	R	[119:112]	1.5ms	0x26	
Data Read Access-Time-2 in CLK Cycles (NSAC*100)	8	R	[111:104]	0	0	
Max. Data Transfer Rate	8	R	[103:96]	20MHZ	0x2a	
Card Command Classes	12	R	[95:84]	All but I/O	0x1ff	
Max. Read Data Block Length	4	R	[83:80]	512	9	
Partial Blocks for Read Allowed	1	R	[79:79]	Yes	1	
Write Block Misalignment	1	R	[78:78]	No	0	
Read Block Misalignment	1	R	[77:77]	No	0	
DSR Implemented	1	R	[76:76]	No	0	
Reserved	2	R	[75:74]	0	0	
Device Size (C_SIZE)	12	R	[73:62]	1967	0x7b0	*
Max. Read Current @V_{DD} Min.	3	R	[61:59]	25ma	4	
Max. Read Current @V_{DD} Max.	3	R	[58:56]	35ma	4	
Max. Write Current @V_{DD} Min.	3	R	[55:53]	35ma	5	
Max. Write Current @V_{DD} Max.	3	R	[52:50]	45ma	5	
Device Size Multiplier (C_SIZE_MULT)	3	R	[49:47]	4	0	*
Erase Sector Size	5	R	[46:42]	1	0	
Erase Group Size	5	R	[41:37]	16	0xf	
Write Protect Group Size	5	R	[36:32]	32	0x1f	
Write Protect Group Enable	1	R	[31:31]	Yes	1	
Manufacturer Default ECC	2	R	[30:29]	None	0	
Read to Write Speed Factor	3	R	[28:26]	1:16	4	
Max. Write Data Block Length	4	R	[25:22]	512	9	
Partial Blocks for Write Allowed	1	R	[21:21]	No	0	
Reserved	5	R	[20:16]	0	0	
Reserved	1	R/W	[15:15]	0	0	
Copy Flag (OTP)	1	R/W	[14:14]	Copy	1	
Permanent Write Protection	1	R/W	[13:13]	No	0	
Temporary Write Protection	1	R/W/E	[12:12]	No	0	

Reserved	2	R/W	[11:10]	0	0	
ECC Code	2	R/W/E	[9:8]	None	0	
CRC	7	R/W/E	[7:1]	TDB	TDB	
Not Used, Always '1'	1	-	[0:0]	1	1	

Note 1: This register is the same as the CID register.

4.5.5 Status Register

The MultiMediaCard status register structure is defined in the following table. 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 4-5 Status Register

Bits	Type	Value	Description	Clear Cond.
31	ER	'0'= no error '1'= error	The commands argument was out of allowed range for this card.	C
30	ERX	'0'= no error '1'= error	A misaligned address, which did not match the block length was used in the command.	C
29	ER	'0'= no error '1'= error	The transferred block length is not allowed for this card or the number of bytes transferred does not match the block length	C
28	ER	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C
27	EX	'0'= no error '1'= error	An invalid selection, sectors or groups, for erase.	C
26	ERX	'0'= not protected '1'= protected	The command tried to write a write protected block.	C
25-24	Reserved			
23	ER	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ER	'0'= no error '1'= error	Command not legal for the current state	B
21	EX	'0'= success '1'= failure	Card internal ECC was applied but the correction of data is failed.	C
20	ERX	'0'= no error '1'= error	Internal card controller error	C
19	ERX	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	EX	'0'= no error '1'= error	The card could not sustain data transfer in stream read mode	C
17	EX	'0'= no error '1'= error	The card could not sustain data programming in stream write mode	C
16	ER	'0'= no error '1'= error	Can be one of the following errors: - The CID register has been already written and can not be overwritten. - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotect) bits was made.	C
15	SX	'0'= not protected '1'= protected	Only partial address space was erased due to existing WP blocks.	C
14			Not applicable. This bit is always set to '0.'	
13	SR	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
12-9	SX	0 = idle 1 = ready 2 = ident 3 = stby 4 = tran 5 = data 6 = rcv 7 = prg 8 = dis 9-15 = reserved	The state of the card when the command was received. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15.	B
8	SX	'0'= not ready '1'= ready	Corresponds to buffer empty signaling on the bus. (RDY/BSY)	A
7-0	Reserved. Always set to '0.'			

4.5.6 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.

Table 4-6 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	No		

5.0 MultiMediaCard Protocol Description

5.1 General

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

- Broadcast Commands

Broadcast commands are intended for all MultiMediaCards. Some of these commands require a response.

- Addressed (Point-to-Point) Commands

The addressed commands are sent to the addressed MultiMediaCard and cause a response from this card.

A general overview of the command flow is shown in Figure 5-1 for the Card Identification Mode and in Figure 5-2 for the Data Transfer Mode. The commands are listed in the command tables (Table 5-3 - Table 5-9). The dependencies between the current MultiMediaCard state, received command and following state are listed in Table 5-10. In the following sections, the different card operation modes will be described first. Thereafter, the restrictions for controlling the clock signal are defined. All MultiMediaCard commands together with the corresponding responses, state transitions, error conditions and timings are presented in the following sections.

Three operation modes are defined for MultiMediaCards:

- 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.

- Interrupt Mode

The Interrupt Mode is not implemented on the current 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.

The following table shows the dependencies between bus modes, operation modes and card states. Each state in the MultiMediaCard state diagram (Figure 5-1 and Figure 5-2) is associated with one bus mode and one operation mode:

Table 5-1 Bus Modes Overview

Card State	Operation Mode	Bus Mode
Inactive State	Inactive	Open-Drain
Idle State	Card Identification Mode	
Ready State		
Identification State		
Stand-by State	Data Transfer Mode	Push-Pull
Transfer State		
Sending-data State		
Receive-data State		
Programming State		
Disconnect State		

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.

5.2 Card Identification Mode

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

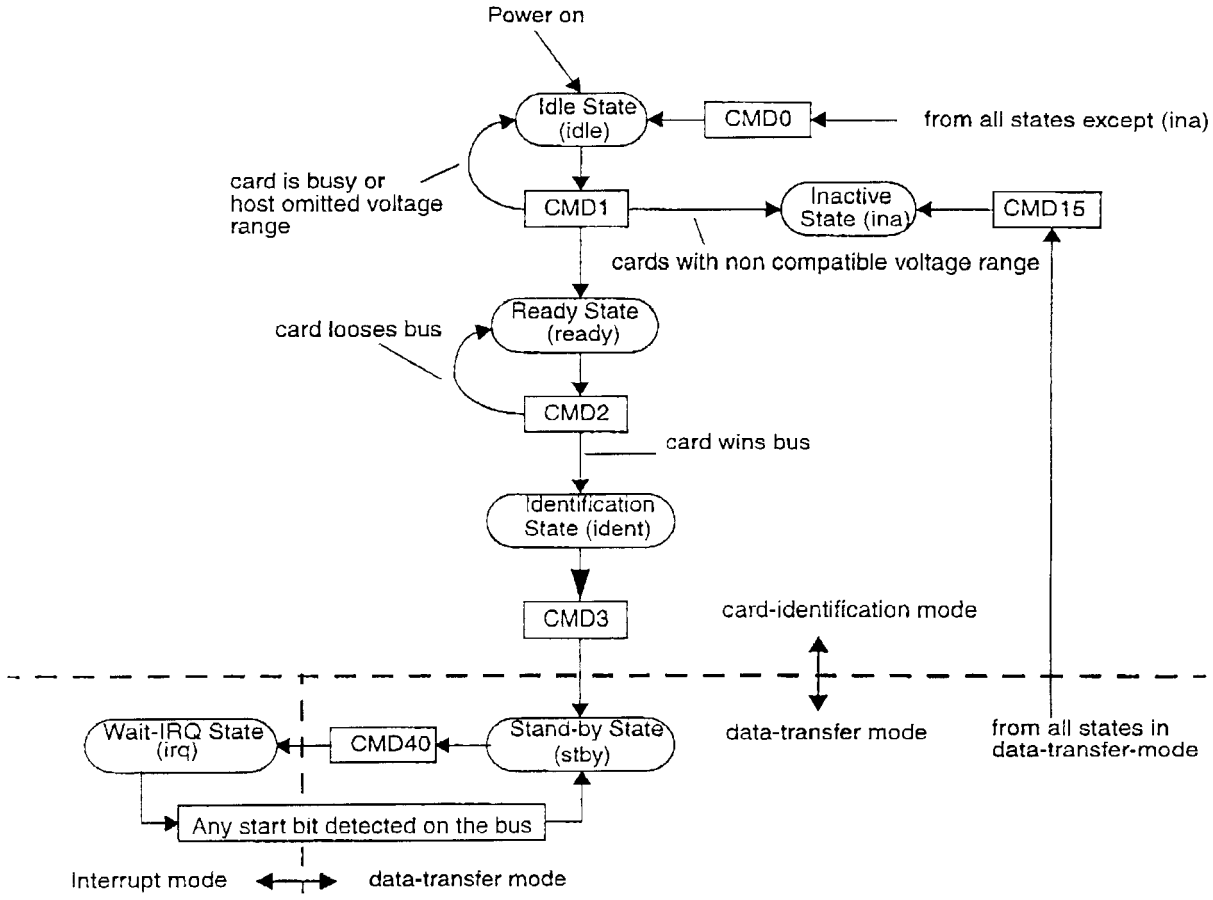


Figure 5-1 MultiMediaCard State Diagram (Card Identification Mode)

5.2.1 Reset

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

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

After power-on or CMD0, all MultiMediaCards' output bus drivers are in a high-impedance state and the cards are initialized with a default relative card address ("0x0001"). The host drives the bus at the identification clock rate f_{OD} (generated by a push-pull driver stage).

5.2.2 Operating Voltage Range Validation

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

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

SEND_OP_COND (CMD1) is designed to provide MultiMediaCard hosts with a mechanism to identify and reject cards which do not match the host's desired V_{DD} range. This is accomplished by the host sending the required V_{DD} voltage window as the operand of this command. MultiMediaCards which can not perform data transfer in the specified range must discard themselves from further bus operations and go into Inactive State. All other MultiMediaCards will respond concurrently (same method as card identification) sending back their V_{DD} range. The wired-or result of the response will show all voltage ranges which some of the cards do not support.

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

The busy bit in the CMD1 response can be used by a card to tell the host that it is still working on its power-up/reset procedure (e.g. downloading the register information from memory field) and is not ready yet for communication. In this case the host must repeat CMD1 until the busy bit is cleared.

During the initialization procedure, the host is not allowed to change the OCR values. Changes in the OCR content will be ignored by the MultiMediaCard. If there is a real change in the operating conditions the host must reset the card stack (using CMD0) and begin the initialization procedure once more.

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

5.2.3 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 the host will request the MultiMediaCards to send their valid operation conditions (CMD1). The response to CMD1 is the 'wired or' operation condition restrictions of all MultiMediaCards in the system. The host must then choose a voltage for operation, reissue CMD1 with this condition, and notify the application that cards with out of range parameters are connected to the bus. Incompatible cards are sent into Inactive State. The host then issues the broadcast command asks all cards for their unique card identification (CID) number with the broadcast command ALL_SEND_CID (CMD2). All remaining unidentified cards (i.e. those which are in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits 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). Since CID numbers are unique for each MultiMediaCard, there should be only one card which successfully sends its full CID-number to the host. This card 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 which will be used to address the card in future data transfer mode communication (typically with a higher clock rate than f_{OD}).

Once the RCA is received the card transfers to the Stand-by State and does not react to further identification cycles. The MultiMediaCard also switches its 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). When no MultiMediaCard responds to this command, all cards have been identified. The time-out condition to recognize completion is the absence of a start bit for more than 5 clock periods after sending CMD2.

5.3 Interrupt Mode

The Interrupt Mode option defined in chapter 4 of the MultiMediaCard standard is not implemented in the SanDisk MultiMediaCard.

5.4 Data Transfer Mode

When all cards are in Stand-by State communication over the CMD and DAT lines will be in push-pull mode. Until the content of all CSD registers is known by the host, the f_{PP} clock rate must remain at f_{OD} because some cards may have operating frequency restrictions. The host issues SEND_CSD (CMD9) to obtain the Card Specific Data (CSD register), e.g. ECC type, block length, card storage capacity, maximum clock rate, etc.

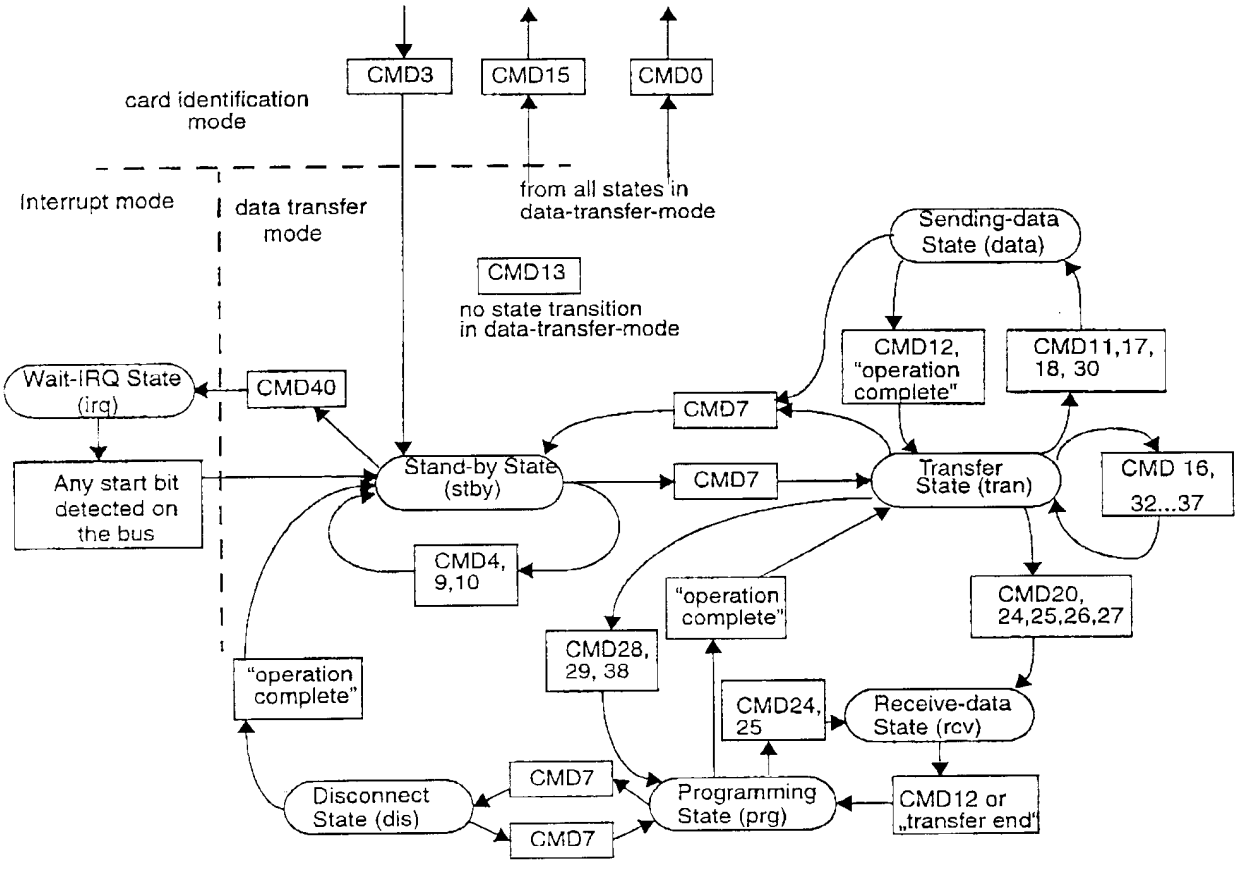


Figure 5-2 MultiMediaCard State Diagram (Data Transfer Mode)

CMD7 is used to select one MultiMediaCard and place it in the Transfer State. Only one MultiMediaCard can be in the Transfer State at a given time. If a previously selected MultiMediaCard 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 command is used to identify new cards without resetting other already acquired cards. MultiMediaCards which already have an RCA do not respond to the 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 5-2, and in the following paragraphs:

- All data read commands can be aborted any time by the stop command (CMD12). The data transfer will terminate and the MultiMediaCard will return to the Transfer State. The read commands are: stream read (CMD11), block read (CMD17), multiple block read (CMD18) and send write protect (CMD30).
- All data write commands can be aborted any time by the stop command (CMD12). The write commands must be stopped prior to deselecting the MultiMediaCard by CMD7. The write commands are: stream write (CMD20), block write (CMD24 and CMD25), write CID (CMD26), and write CSD (CMD27).
- If a stream write operation is stopped prior to reaching the block boundary and partial blocks are allowed (as defined in the CSD), the part of the last block will be packed as a partial block and programmed. If partial blocks are not allowed, the data will be discarded.
- As soon as the data transfer is completed, the MultiMediaCard will exit the data write state and move either to the Programming State (transfer is successful) or Transfer State (transfer failed).
- If a block write operation is stopped and the block length and CRC of the last block are valid, the data will be programmed.
- If data transfer in stream write mode is stopped, not byte aligned, the bits of the incomplete byte are ignored and not programmed.
- The MultiMediaCard may provide buffering for stream and block write. This means that the next block can be sent to the card while the previous is being programmed. If all write buffers are full, and as long as the MultiMediaCard is in Programming State (see MultiMediaCard state diagram Figure 5-2), the DAT line will be kept low.
- There is no buffering option for write CSD, write CID, write protection and erase. This means that while the MultiMediaCard is busy servicing any one of these commands, no other data transfer commands will be accepted. DAT line will be kept low as long as the MultiMediaCard is busy and in the Programming State.
- Parameter set commands are *not* allowed while the MultiMediaCard is programming. Parameter set commands are: set block length (CMD16), and erase tagging/untagging (CMD32-37).
- Read commands are *not* allowed while the MultiMediaCard is programming.
- Moving another MultiMediaCard from Stand-by to Transfer State (using CMD7) will not terminate a programming operation. The MultiMediaCard will switch to the Disconnect State and will release the DAT line.
- A MultiMediaCard can be reselected while in the Disconnect State, using CMD7. In this case the MultiMediaCard will move to the Programming State and reactivate the busy indication.
- Resetting a MultiMediaCard (using CMD0 or CMD15) will terminate any pending or active programming operation. This may destroy the data contents on the MultiMediaCard. It is up to the host's responsibility to prevent this.

5.4.1 Data Read Format

The DAT bus line is high when no data is transmitted. A transmitted data block consists of a start bit (LOW), followed by a continuous data

stream. The data stream contains the net payload data (and error correction bits if an off-card ECC is used). The data stream ends with an end bit (HIGH) (see Figure 5-10 through Figure 5-12). The data transmission is synchronous to the clock signal.

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

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

The code is a shortened BCH code with d=4 and is used for payload length of up to 2048 Bytes.

- Stream read

There is a stream oriented data transfer controlled by READ_DAT_UNTIL_STOP (CMD11). This command instructs the card to send its payload, starting at a specified address, until the host sends a STOP_TRANSMISSION command (CMD12). Note that the host stop command has an execution delay due to the serial command transmission. The data transfer stops after the end bit of the next command, which has the interrupt ability.

If the end of the memory range is reached while sending data and no stop command has yet been sent by the host, the content of the further transferred payload is undefined.

The maximum clock frequency for stream read operation is given by the following formula:

max. speed = min ((TRAN_SPEED), (READ_BL_LEN/(NSAC+TAAC))[1]

If the host attempts to use a higher frequency, the card may not be able to sustain data transfer. Should this happen, the card will set the UNDERRUN error bit in the status register, abort the transmission and wait in the data state for a stop or a new read command.

- Block read

Block read is similar to stream read, except 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 wholly contained within one physical block (as defined by READ_BL_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. CMD17 (READ_SINGLE_BLOCK) starts a block read and after a complete transfer the card goes back to Transfer State. CMD18 (READ_MULTIPLE_BLOCK) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued.

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

5.4.2 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. The card performs a CRC check for each such received data block prior to a write operation. (The polynomial is the same one used for a read operation.) By this mechanism, writing of erroneously transferred data can be prevented.

- Stream write

Stream write (CMD20) means that data is transferred beginning from the starting address until the host issues a stop command. If partial blocks are allowed (CSD parameter WRITE_BL_PARTIAL is set) the data stream can start and stop at any address within the card address space, otherwise it must start and stop on block boundaries. Since the amount of data to be transferred is not determined in advance, CRC can not be used. If the end of the memory range is reached while sending data and no stop command has been sent by the host, the content of the further transferred payload is discarded.

The maximum clock frequency for stream write operation is given by the following formula:

max. speed = min ((TRAN_SPEED), ((WRITE_BL_LEN/(NSAC+TAAC)) /R2W_FACTOR)

1) All upper case names are defined in the CSD.

If the host attempts to use a higher frequency, the card may not be able to process the data and will stop programming, set the OVERRUN error bit in the status register, and while ignoring all further data transfer, wait (in the *Receive-Data-State*) for a stop command. The write operation will also be aborted if the host tries to write over a write over a write protected area. In this case, however, the card will set the WP_VIOLATION bit.

- Block Write

Block write (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 (see below); 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 whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card will detect the block misalignment error and abort programming before the beginning of the first misaligned block. The card will set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, wait (in the *Receive-Data-State*) for a stop command.

The write operation 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.

Programming of the CID and CSD register does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM it will not be overwritten and the card will not check the ROM data with the content of the receive buffer.

Sometimes cards might take longer to write a block of data due to internal management. After receiving a block of data and completing the CRC check, the card will begin writing and hold DAT 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 MultiMediaCard can accept new data or whether the write process is still in progress. The host may deselect the card by issuing CMD7 (to select a different card) which will 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.

- Erase

It is desirable to erase many sectors simultaneously in order to enhance the data throughput. Identification of these 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 both together. That is, the unit of measure for determining an erase is either a sector or an erase group, but if a sector, all selected sectors must lie 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). The following exception conditions are detected by the MultiMediaCard:

- An erase or tag/untag command is received out of sequence. The card will set the ERASE_SEQ_ERROR error bit in the status register and reset the whole 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 be left 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) which are used in a sequence is limited up to 16.

As described above for block write, the MultiMediaCard will indicate that an erase is in progress by holding DAT low. The actual erase time may be quite long, and the host may chose to deselect the card using CMD7.

- Write Protect Management

Card data may be protected against either erase or write by the write protection features. The entire card may be permanently write protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of smaller groups (the WP_GRP_ENABLE bit in the CSD is set), portions of the data may be protected (in units of WP_GRP_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_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.

5.4.3 Clock Control

The MultiMediaCard bus clock signal can be used by the MultiMediaCard 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 lower the clock frequency or shut it down.

There are a few restrictions the MultiMediaCard host must follow:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency, defined by the MultiMediaCards, and the identification frequency defined by the MMC specification document).
- It is an obvious requirement that the clock must be running for the MultiMediaCard to output data or response tokens. After the last MultiMediaCard bus transaction, the host is required, to provide **8 (eight)** clock cycles for the card to complete the operation before shutting down the clock. Following is a list of the various MultiMediaCard bus transactions:
- A command with no response. 8 clocks after the host command end bit.
- A command with response. 8 clocks after the card response end bit.
- A read data transaction. 8 clocks after the end bit of the last data block.
- A write data transaction. 8 clocks after the CRC status token.
- The host is allowed to shut down the clock of a "busy" card. The 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 MultiMediaCard (unless previously disconnected by a deselect command -CMD7) will force the DAT line down, permanently.

5.4.4 Error Conditions

5.4.4.1 CRC and Illegal Command

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

Similarly, if an illegal command has been received, a MultiMediaCard shall not change its state, shall not response and shall set the

ILLEGAL_COMMAND error bit in the status register. Only the non-erroneous state branches are shown in the state diagrams (Figure 5-1 and Figure 5-2). Table 5-10 contains a complete state transition description.

There are different kinds of illegal commands:

- Commands which belong to classes not supported by the MultiMediaCard (e.g. write commands in read only cards).
- Commands not allowed in the current state (e.g. CMD2 in Transfer State).
- Commands which are not defined (e.g. CMD6).

5.4.4.2 Read, Write and Erase Time-out Conditions

The times after which a time-out condition for read/write/erase operations occurs are (card independent) **10 times longer** than the typical access/program times for these operations given below. A card shall 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, etc.). The typical access and program times are defined as follows:

Read
The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC. These card parameters define the typical delay between the end bit of the read command and the start bit of the data block. This number is card dependent and should be used by the host to calculate throughput and the maximal frequency for stream read.

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

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.

5.5 Commands

5.5.1 Command Types

There are four kinds of commands defined on the MultiMediaCard bus:

- Broadcast commands (bc)—sent on CMD, no response
- Broadcast commands with response (bcr)—sent on CMD, response (all cards simultaneously) on CMD
- Addressed (point-to-point) commands (ac)—sent on CMD, response on CMD
- Addressed (point-to-point) data transfer commands (adtc)—sent on CMD, response on CMD, data transfer on DAT

The command transmission always starts with the MSB.

5.5.2 Command Format

(Command length 48 bits, 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[1]	end bit

Commands and arguments are listed in Table 5-3 through Table 5-9.

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

5.5.3 Command Classes

The command set of the MultiMediaCard is divided into several classes (See Table 5-2). Each class supports a set of MultiMediaCard functions.

Class 0 is mandatory and supported by all MultiMediaCards. The other classes are optional and can be interpreted as a tool box. By using different classes, several configurations can be chosen (e.g. a block writable card or a stream readable card). The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

Table 5-2 Card Command Classes (CCCs)

Card Command Class (CCC)	Class Description	Supported Commands															
		0	1	2	3	4	7	9	10	11	12	13	15	16	17	18	20
Class 0	Basic	+	+	+	+	+	+	+	+		+	+	+				
Class 1	Stream Read									+							
Class 2	Block Read													+	+	+	
Class 3	Stream Write																+
Class 4	Block Write													+			
Class 5	Erase																
Class 6	Write Write-Protection																
Class 7	Read Write-Protection																
Class 8	Erase Write-Protection																
Class 9	I/O Mode																
Class 10-11	Reserved																

1) 7-bit Cyclic Redundancy Check.

Card Command Class (CCC)	Class Description	Supported Commands																	
		24	25	26	27	28	29	30	32	33	34	35	36	37	38	39	40		
Class 0	Basic																		
Class 1	Stream Read																		
Class 2	Block Read																		
Class 3	Stream Write																		
Class 4	Block Write	+	+	+	+														
Class 5	Erase								+	+	+	+	+	+	+				
Class 6	Write Write-Protection					+		+											
Class 7	Read Write-Protection							+											
Class 8	Erase Write-Protection					+	+	+											
Class 9	I/O Mode															+	+		
Class 10-11	Reserved																		

5.5.4 Detailed Command Description

All future reserved commands have to be 48 bit long, their responses have to be also 48 bits long or they might also have no response.

The following tables define in detail the MultiMediaCard bus commands.

Table 5-3 Basic Commands and Read Stream Commands (Class 0 And Class 1)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] don't cares*	-	GO_IDLE_STATE	Resets all cards to Idle State.
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks 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 cares*	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 cares*	R1	SET_RELATIVE_ ADDR	Assigns relative address to the card.
CMD4	bc	[31:16] DSR [15:0] don't cares*	-	SET_DSR	Programs the DSR of all cards.
CMD5	Reserved				
CMD6	Reserved				
CMD7	ac	[31:16] RCA [15:0] don't cares*	R1 (only from the selected card)	SELECT/DESELECT_ CARD	Command toggles a card between the Stand-by and Transfer states or between the Programming and Disconnect state. In both cases the card is selected by its own relative address and deselected by any other address; address 0 deselects all.
CMD8	Reserved				
CMD9	ac	[31:16] RCA [15:0] don't cares*	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] don't cares*	R2	SEND_CID	Addressed card sends its card identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address[1]	R1	READ_DAT_UNTIL_ STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] don't cares*	R1	STOP_ TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:16] RCA [15:0] don't cares*	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:16] RCA [15:0] don't cares*	-	GO_INACTIVE_ STATE	Sets the card to inactive state.

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

1) The addressing capability @ 8 bit address resolution is 2^{32} = 4 Gbyte.

Table 5-4 Block Oriented Read Commands (Class 2)

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

Table 5-5 Sequential Write Commands (Class 3)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD20	adtc	[31:0] data address	R1b	WRITE_DAT_UNTIL_ STOP	Writes data stream from the host in receiving-data state, starting at the supplied address, until a STOP_TRANSMISSION follows.
CMD21. .. CMD23	Reserved				

1) The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.
2) The data transferred must not cross a physical block boundary unless RD_BLK_MISALIGN is set in the CSD.

Table 5-6 Block Oriented Write Commands (Class 4)

Cmd Index	Type	Argument	Resp[1]	Abbreviation	Command Description
CMD24	adtc	[31:0] data address	R1[2]	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command. [3]
CMD25	adtc	[31:0] data address	R1[2]	WRITE_MULTIPLE_ BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] don't cares*	R1b	PROGRAM_CID	Programming of the card identification register. This command is only done once per MultiMediaCard. The card contains hardware to prevent this operation after the first programming. Normally this command is preserved for the manufacturer.
CMD27	adtc	[31:0] don't cares*	R1b	PROGRAM_CSD	Programming of the programmable bits of the CSD.

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

Table 5-7 Block Oriented Write Commands (Class 6 - 8)

Cmd Index	Type	Argument	Resp[4]	Abbreviation	Command Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_ PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits. [5]
CMD31	Reserved				

1) R1b: These commands are indicating the busy status of the MultiMediaCard via the data channel.
2) Data followed by data response plus busy.
3) The data transferred must not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD
4) R1b: these commands are indicating the busy status of the MultiMediaCard via the data channel.
5) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line.

Table 5-8 Erase Commands (Class 5)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector 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 selected for erase.
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 cares*	R1b	ERASE	Erases all previously selected sectors or erase groups.

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

Table 5-9 I/O Mode Commands (Class 9)

Cmd Index	Type	Argument	Resp	Abbreviation	Command Description
CMD39 CMD40	Currently not supported.				
CMD41. . . CMD59	Reserved				
CMD60-63	Reserved for manufacturer				

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

5.6 Card State Transition Table

Table 5-10 defines the MultiMediaCard state transitions in dependency of the received command.

Table 5-10 Card State Transition Table

	Current State										
	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	-	-	-	stby	-	-	-	-	-	-	stby
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
class 1											
CMD11	-	-	-	-	data	-	-	-	-	-	stby
class 2											
CMD16	-	-	-	-	tran	-	-	-	-	-	stby
CMD17	-	-	-	-	data	-	-	-	-	-	stby
CMD18	-	-	-	-	data	-	-	-	-	-	stby
class 3											
CMD20	-	-	-	-	rcv	-	-	-	-	-	stby
class 4											
CMD16	see class 2										

CMD24	-	-	-	-	rcv	-	-	rcv	-	-	stby
CMD25	-	-	-	-	rcv	-	-	rcv	-	-	stby
CMD26	-	-	-	-	rcv	-	-	-	-	-	stby
CMD27	-	-	-	-	rcv	-	-	-	-	-	stby
class 6 - 8											
CMD28	-	-	-	-	prg	-	-	-	-	-	stby
CMD29	-	-	-	-	prg	-	-	-	-	-	stby
CMD30	-	-	-	-	data	-	-	-	-	-	stby
class 5											
CMD32	-	-	-	-	tran	-	-	-	-	-	stby
CMD33	-	-	-	-	tran	-	-	-	-	-	stby
CMD34	-	-	-	-	tran	-	-	-	-	-	stby
CMD35	-	-	-	-	tran	-	-	-	-	-	stby
CMD36	-	-	-	-	tran	-	-	-	-	-	stby
CMD37	-	-	-	-	tran	-	-	-	-	-	stby
CMD38	-	-	-	-	prg	-	-	-	-	-	stby
class 9											
CMD39, CMD40	Currently not supported.										
class 10 - 11											
CMD41...CMD59	Reserved										
CMD60...CMD63	Reserved for manufacturer										

5.7 Responses

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

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

There are five types of responses. Their formats are defined as follows:

R1 (normal response command): code length 48 bit.
The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits.

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

R1b is identical with R1, but additional busy signaling via the data channel as defined in the MultiMediaCard Specification.

R2 (CID, CSD register): code length 136 bits.
The contents of the CID register are sent as a response to the commands CMD2 and CMD10. The contents of the CSD register are sent as a response to CMD9. Only the bits [127...1] of the CID and CSD are transferred, bit [0] of these registers is replaced by the end bit of the response.

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

R3 (OCR register): code length 48 bits.
The contents of the OCR register is sent as a response to CMD1.

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

Responses R4 and R5 are not supported.

5.8 Card Register

There are a set of six registers within the card interface. These registers are used for the serial data communication. The OCR, CID and CSD registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters.

5.8.1 OCR Register

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. In addition this register includes a status information bit. This status bit informs the host that the card power up procedure is already finished. The OCR register (as well as CMD1) must be implemented by cards which do not support the full operating voltage range of the MultiMediaCard bus or the card power up extends the definition defined in the timing diagram.

Table 5-11 OCR Register Definition

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

The level coding of the OCR register is as follows:

- restricted voltage windows=LOW
- card busy=LOW (bit 31)

5.8.2 CID Register

This register carries the card identification information (Card ID) used during the card identification procedure. It is a 128 bit wide register with a content defined below. This data has to be error-free.

The CID is divided into three slices.

1) This bit is set if the card has not finished the power up routine yet.

Name	Type	Width	CID - Slice	Comments
Manufacturer ID	Binary	24	[127:104]	The manufacturer IDs are controlled and assigned by the MultiMediaCard Association.[1]
Product name	String	56	[103:48]	
HW Revision	Binary	4	[47:44]	Card hardware revision.
FW Revision	Binary	4	[43:40]	Card firmware revision.
Serial Number	Binary	24	[39:16]	A unique card ID number.
Month code	Binary	4	[15:12]	Manufacturing date - month
Year code	Binary	4	[11:8]	Manufacturing date - year (offset from 1997)
CRC7 checksum*	Binary	7	[7:1]	Calculated
Not used, always '1'		1	[0:0]	

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

5.8.3 CSD Register

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

Table 5-13 The CSD Fields

Name	Field	Width	Cell Type	CSD-slice
CSD Structure	CSD_STRUCTURE	2	R	[127:126]
MultiMediaCard Protocol Version	MMC_PROT	4	R	[125:122]
Reserved	-	2	R	[121:120]
Data Read Access-Time-1	TAAC	8	R	[119:112]
Data Read Access-Time-2 In CLK Cycles (NSAC*100)	NSAC	8	R	[111:104]
Max. Data Transfer Rate	TRAN_SPEED	8	R	[103:96]
Card Command Classes	CCC	12	R	[95:84]
Max. Read Data Block Length	READ_BL_LEN	4	R	[83:80]
Partial Blocks For Read Allowed	READ_BL_PARTIAL	1	R	[79:79]
Write Block Misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]

1) Two numbers have been assigned so far: 0x001 to Siemens, 0x002 to SanDisk.

Read Block Misalignment	READ_BLK_MISALIGN	1	R	[77:77]
DSR Implemented	DSR_IMP	1	R	[76:76]
Reserved	-	2	R	[75:74]
Device Size	C_SIZE	12	R	[73:62]
Max. Read Current @V_{DD} Min.	VDD_R_CURR_MIN	3	R	[61:59]
Max. Read Current @V_{DD} Max.	VDD_R_CURR_MAX	3	R	[58:56]
Max. Write Current @V_{DD} Min.	VDD_W_CURR_MIN	3	R	[55:53]
Max. Write Current @V_{DD} Max.	VDD_W_CURR_MAX	3	R	[52:50]
Device Size Multiplier	C_SIZE_MULT	3	R	[49:47]
Erase Sector Size	SECTOR_SIZE	5	R	[46:42]
Erase Group Size	ERASE_GRP_SIZE	5	R	[41:37]
Write Protect Group Size	WP_GRP_SIZE	5	R	[36:32]
Write Protect Group Enable	WP_GRP_ENABLE	1	R	[31:31]
Manufacturer Default ECC	DEFAULT_ECC	2	R	[30:29]
Stream Write Speed Factor	R2W_FACTOR	3	R	[28:26]
Max. Write Data Block Length	WRITE_BL_LEN	4	R	[25:22]
Partial Blocks For Write Allowed	WRITE_BL_PARTIAL	1	R	[21:21]
Reserved	-	5	R	[20:16]
Reserved	-	1	R/W	[15:15]
Copy Flag (OTP)	COPY	1	R/W	[14:14]
Permanent Write Protection	PERM_WRITE_PROTECT	1	R/W	[13:13]
Temporary Write Protection	TMP_WRITE_PROTECT	1	R/W/E	[12:12]
Reserved	-	2	R/W	[11:10]
ECC Code	ECC	2	R/W/E	[9:8]
Crc	CRC	7	R/W/E	[7:1]
Not Used, Always ‘1’	-	1	-	[0:0]

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 5-14 CSD Register Structure

CSD_STRUCTURE	CSD Structure Version	Valid for MultiMediaCard Protocol Version
0	CSD version No. 1.0	MultiMediaCard protocol version 1.0-1.2
1	CSD version No. 1.1	MultiMediaCard protocol version 1.4
2-3	reserved	

MMC_PROT—Defines the MultiMediaCard protocol version supported by the card. It includes the definition of the command set and the card responses. The card identification procedure is compatible for all protocol versions.

Table 5-15 MultiMediaCard Protocol Version

MMC_PROT	MultiMediaCard Protocol Version
0	MultiMediaCard Protocol Version 1.0-1.2
1	MultiMediaCard Protocol Version 1.4
2-15	reserved

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

Table 5-16 TAAC Access Time Definition

TAAC Bit Position	Code
2:0	time exponent 0=1ns, 1=10ns, 2=100ns, 3=1µms, 4=10µms, 5=100µms, 6=1ms, 7=10ms
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

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

The total read access time N_{AC} as expressed in the Table 5-27 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 or stream from the end bit on the read commands.

TRAN_SPEED—The following table defines the maximum data transfer rate TRAN_SPEED:

Table 5-17 Maximum Data Transfer Rate Definition

TRAN_SPEED bit	code
2:0	transfer rate exponent 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/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. For command class definition refer to Table 5-2.

Table 5-18 Supported Card Command Classes

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

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

Table 5-19 Data Block Length

READ_BL_LEN	Block Length	Remark
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.

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

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

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

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

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

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

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

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

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

Table 5-20 DSR Implementation Code Table

DSR_IMP	DSR Type
0	no DSR implemented
1	DSR implemented

C_SIZE (Device Size)—This parameter is used to compute the card capacity. The memory capacity of the card is computed from the entries C_SIZE, C_SIZE_MULT and READ_BL_LEN as follows:

memory capacity = BLOCKNR * BLOCK_LEN

where

BLOCKNR = (C_SIZE+1) * MULT
MULT = $2^{C\_SIZE\_MULT+2}$ (C_SIZE_MULT < 8)
BLOCK_LEN = $2^{READ\_BL\_LEN}$ (READ_BL_LEN < 12)

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

VDD_R_CURR_MIN, VDD_W_CURR_MIN—The minimum values for read and write currents on VDD power supply are coded as follows:

Table 5-21 V_{DD} Minimum Current Consumption

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

VDD_R_CURR_MAX, VDD_W_CURR_MAX—The maximum values for read and write currents on VDD power supply are coded as follows:

Table 5-22 V_{DD} Maximum Current Consumption

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

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

Table 5-23 Multiply Factor For The Device Size

C_SIZE_MULT	MULT	Remark
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	$2^9 = 512$	

SECTOR_SIZE—The size of an erasable sector. The contents of this register is a 5 bit binary coded value, defining the number of write blocks (see WRITE_BL_LEN). The actual size is computed by increasing this number by one. A value of zero means 1 write block, 31 means 32 blocks.

ERASE_GRP_SIZE—The size of an erasable group. The contents of this register is a 5 bit binary coded value, defining the number of sectors (see SECTOR_SIZE). The actual size is computed by increasing this number by one. A value of zero means 1 sector, 31 means 32 sectors.

WP_GRP_SIZE—The size of a write protected group. The contents 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 zero means 1 erase group, 31 means 32 erase groups.

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

DEFAULT_ECC—Set by the card manufacturer. It defines the ECC code which is recommended for use. The field definition is the same as for the ECC field described later.

R2W_FACTOR—Defines the typical block program time as a multiple of the read access time. The following table defines the field format.

Table 5-24 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
6,7	reserved

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.

WRITE_BL_PARTIAL='0' means that only the WRITE_BL_LEN block size can be used for block oriented data write.

WRITE_BL_PARTIAL='1' means that smaller blocks can be used as well. The minimum block size is one byte.

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

PERM_WRITE_PROTECT—Permanently protects the whole card content 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.

ECC—Defines the ECC code that was used for storing data on the card. This field is used by the host (or application) to decode the user data. The following table defines the field format:

Table 5-25 ECC Type

ECC	ECC Type	Maximum Number Of Correctable Bits Per Block
0	none (default)	none
1	BCH (542,512)	3
2-3	reserved	-

CRC—The CRC field carries the check sum for the CSD contents. The checksum has to be recalculated by the host for any CSD modification. The default corresponds to the initial CSD contents.

5.8.4 RCA Register

The 16-bit relative card address register carries the card address assigned by the host during the card identification. This address is used for the addressed host-card communication after the card identification procedure. The default value of the RCA register is 0x0001. The value 0x0000 is reserved to set all cards in Stand-by State with CMD7.

5.9 Memory Array Partitioning

The basic unit of data transfer to/from the MultiMediaCard is one byte. All data transfer operations which require a block size always define block lengths as integer multiples of bytes.

For block oriented commands:

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

  For devices which have erasable memory cells, special erase commands are defined. The granularity of the erasable units is general not the same as for the block oriented commands:
- **Sector**—The unit which is related to the erase commands. Its size is the number of blocks which will be erased in one portion. The size of a sector is fixed for each device. The information about the sector size (in blocks) is stored in the CSD.
- **Group**—A number of sectors. Its size is the number of consecutive sectors which will be erased in one portion. The size of a group is fixed for each device. The information about the size is stored in the CSD.

For devices which include a write protection:

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

Each erasable unit (group and sector) has a special "tag" bit. This bit may be set or cleared by special commands to tag the unit. All tagged units will be erased in parallel by one erase command following a number of tag commands. All tag bits are cleared by each command except a tag or untag command. So usually immediately after a set of tag commands an erase command has to be sent by the host. No-tag and no-erase commands abort a tag-erase cycle irregularly.

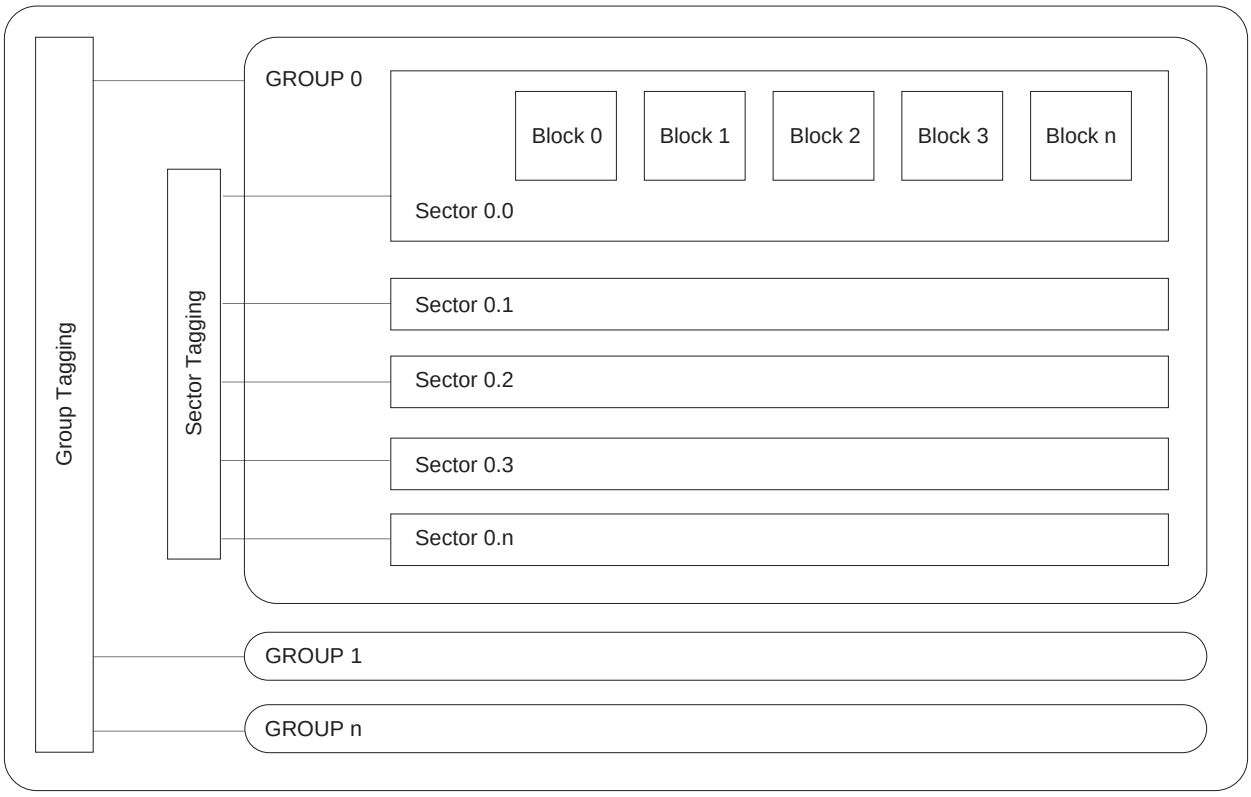


Figure 5-3 Erase Tagging Hierarchy

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

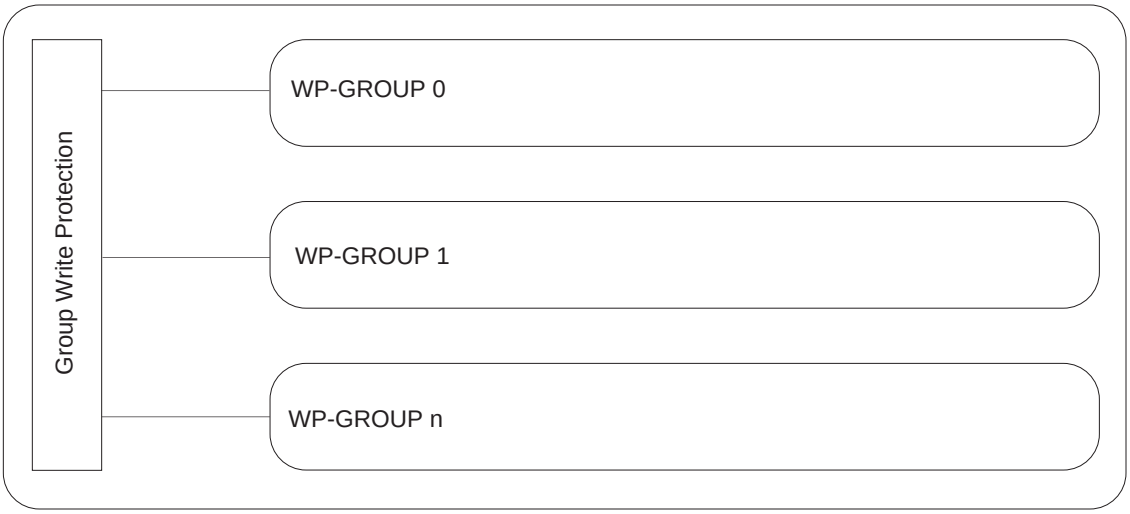


Figure 5-4 Write Protection

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

5.10 Timings

All timing diagrams use the following schematics and abbreviations:

Table 5-26 Timing Diagram Symbols

S	Start Bit (= 0)
T	Transmitter Bit (Host = 1, Card = 0)
P	One-cycle Pull-up (= 1)
E	End Bit (=1)
Z	High Impedance State (-> = 1)
D	Data Bits
*	Repeater
CRC	Cyclic Redundancy Check Bits (7 Bits)
	Card Active
	Host Active

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

All timing values are defined in Table 5-27.

5.10.1 Command and Response

Both host command and card response are clocked out with the rising edge of the host clock. The minimum delay between the host command and card response is N_{CR} clock cycles. This timing diagram is relevant for host command CMD3.

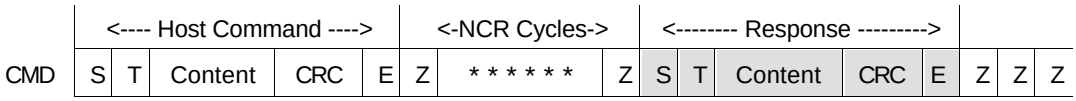


Figure 5-5 Command Response Timing (Identification Mode)

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.

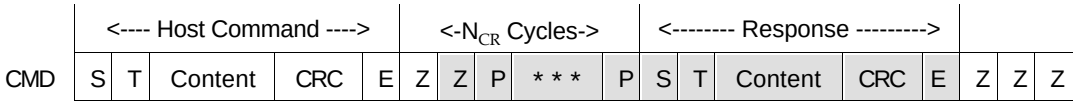


Figure 5-6 Command Response Timing (Data Transfer Mode)

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 N_{ID} clock cycles.

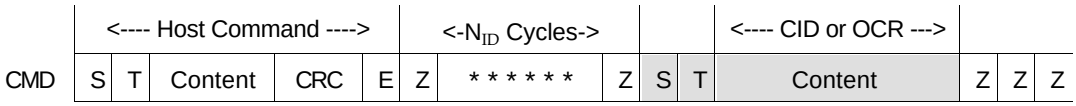


Figure 5-7 Identification Timing (Card Identification Mode)

Last Card Response - Next Host Command Timing—After receiving the last card response, the host can start the next command transmission after at least N_{RC} clock cycles. This timing is relevant for any host command.

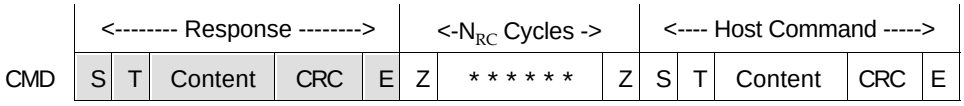


Figure 5-8 Timing Response End to Next CMD Start (Data Transfer Mode)

Last Host Command - Next Host Command Timing Diagram—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.

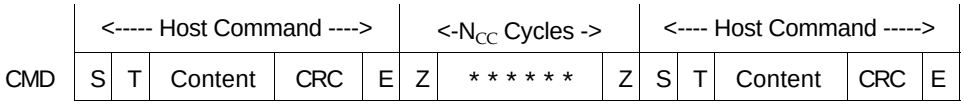


Figure 5-9 Timing CMD_n End to CMD_{n+1} Start (All Modes)

In the case the $\mathbf{CMD}_n$ command was a last acquisition command no more responded by any card, than the next $\mathbf{CMD}_{n+1}$ command is allowed to follow after at least N_{CC} +136 (the length of the R2 response) clock periods.

5.10.2 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 given in Figure 5-10. The sequence starts with a single block read command (CMD17) which specifies the start address in the argument field. The response is sent on the CMD line as usual.

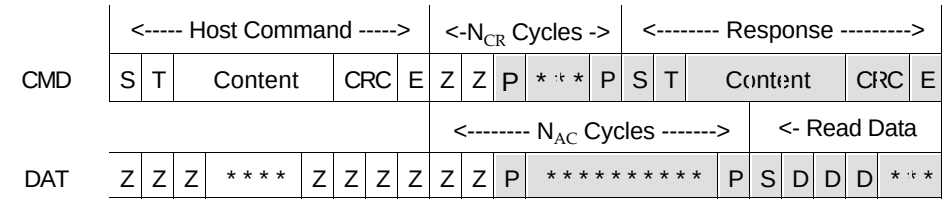


Figure 5-10 Transfer of Single Block Read

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

Multiple Block Read—In multiple block read mode, the card sends a continuous flow of data blocks following the initial host read command. The data flow is terminated by a stop transmission command (CMD12). Figure 5-11 describes the timing of the data blocks and Figure 5-12 the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.

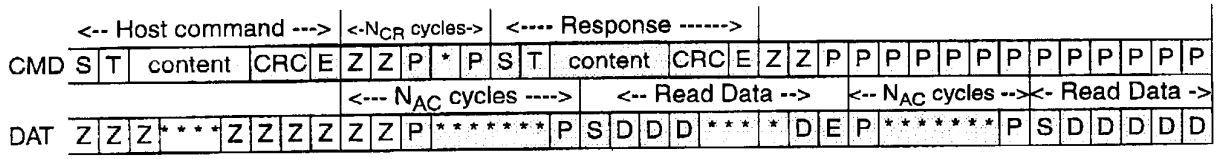


Figure 5-11 Timing of Multiple Block Read Command

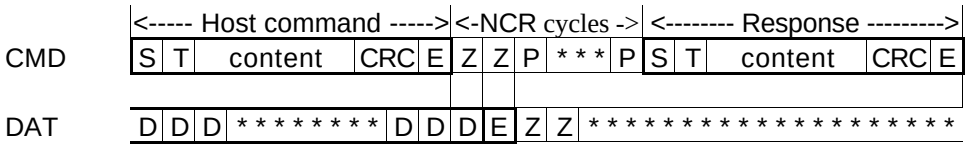


Figure 5-12 Timing of Stop Command (CMD12, Data Transfer Mode)

Stream Read—The data transfer starts N_{AC} clock cycles after the end bit of the host command. The bus transaction is identical to that of a read block command (see Figure 5-10). As the data transfer is not block oriented, the data stream does not include the CRC checksum. Consequently, the host can not check for data validity. The data stream is terminated by a stop command. The corresponding bus transaction is identical to the stop command for the multiple read block (see Figure 5-12).

5.10.3 Data Write

Single Block Write—The host selects one card for a data write operation by CMD7.

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

The basic bus timing for a write operation is given in Figure 5-13. The sequence starts with a single block write command (CMD24) which determines (in the argument field) the start address. It is responded by the card on the CMD line as usual. The data transfer from the host starts N_{WR} clock cycles after the card response was received.

The data is suffixed with CRC check bits to allow the card to check it for transmission errors. The card sends back the CRC check result as a CRC status token on 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.

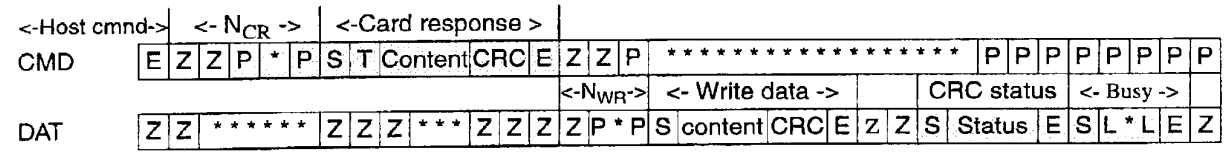


Figure 5-13 Timing Of The Block Write Command

If the MultiMediaCard does not have a free data receive buffer, the card indicates this condition by pulling down the data line to LOW. The card stops pulling down the 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 which must be polled by the host.

Multiple Block Write—In multiple block write mode, the card expects continuous flow of data blocks following the initial host write command. The data flow is terminated by a stop transmission command (CMD12). Figure 5-14 describes the timing of the data blocks with and without card busy signal.

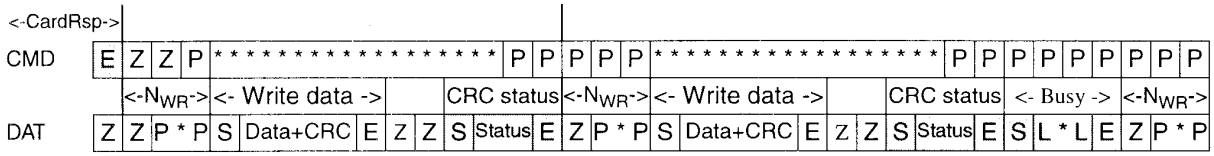


Figure 5-14 Timing of Multiple Block Write Command

In write mode, the stop transmission command works similarly to the stop transmission command in the read mode. Figures 5-15 to 5-18 describe the timing of the stop command in different card states.

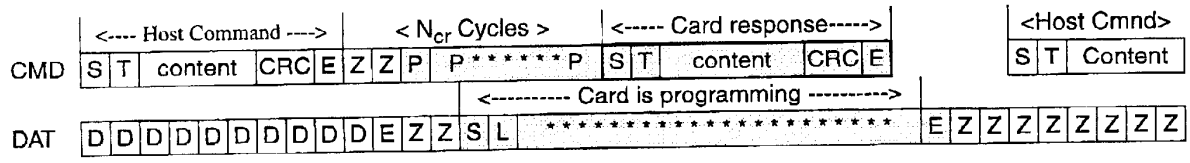
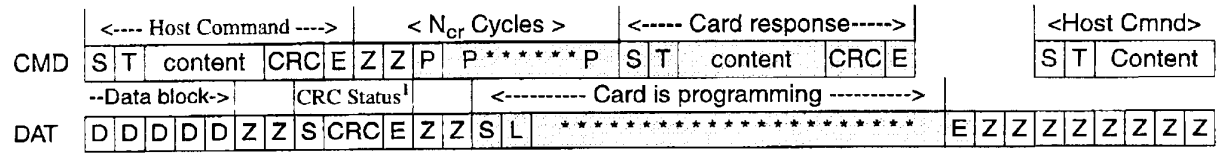


Figure 5-15 Stop Transmission During Data Transfer from the Host

The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status token sent back to the host. Figure 5-16 is an example of an interrupted (by a host stop command) attempt to transmit the CRC status block. The sequence is identical to all other stop transmission examples. The end bit of the host command is followed, on the data line, with one more data bit, end bit and two Z clock for switching the bus direction. The received data block, in this case is considered incomplete and will not be programmed.



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

Figure 5-16 Stop Transmission During CRC Status Transfer from the Card

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

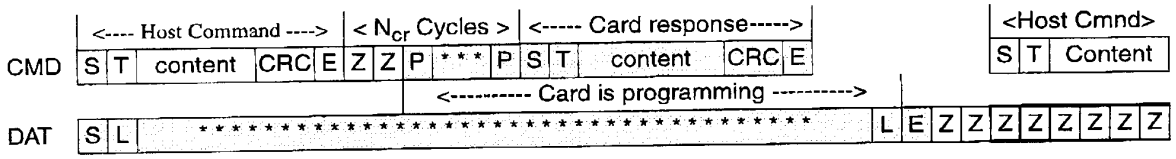


Figure 5-17 Stop Transmission Received After Last Data Block. Card is Busy Programming.

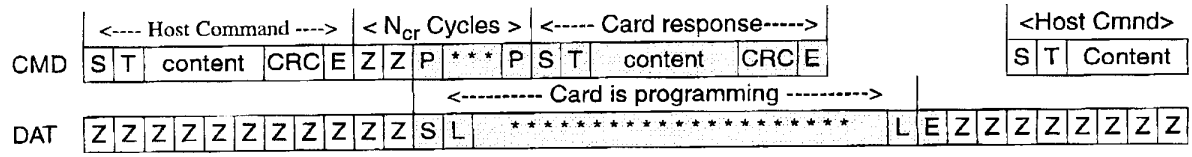


Figure 5-18 Stop Transmission Received After Last Data Block. Card Becomes Busy.

Stream Write—The data transfer starts N_{WR} clock cycles after the card response to the sequential write command was received. The bus transaction is identical to that of a write block command (see Figure 5-13). As the data transfer is not block oriented, the data stream does not include the CRC checksum. Consequently the host can not receive any CRC status information from the card. The data stream is terminated by a stop command. The bus transaction is identical to the write block option when a data block is interrupted by the stop command (see Figure 5-15).

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. The bus transaction timings are described in Figure 5-18.

5.10.4 Timing Values

Table 5-27 defines all timing values.

Table 5-27 Timing Values

	Min	Max	Unit
N_{CR}	2	64	Clock Cycles
N_{ID}	5	5	Clock Cycles
N_{AC}	2	TAAC + NSAC	Clock Cycles
N_{RC}	8	-	Clock Cycles
N_{CC}	8	-	Clock Cycles
N_{WR}	2	-	Clock Cycles

5.11 Error Protection

5.11.1 Cyclic Redundancy Codes (CRC)

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

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

generator polynomial: $G(x) = x^7 + x^3 + 1$.
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + ... + (\text{last bit}) * x^0$
$CRC[6...0] = \text{Remainder}\ [(M(x) * x^7) / G(x)]$

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

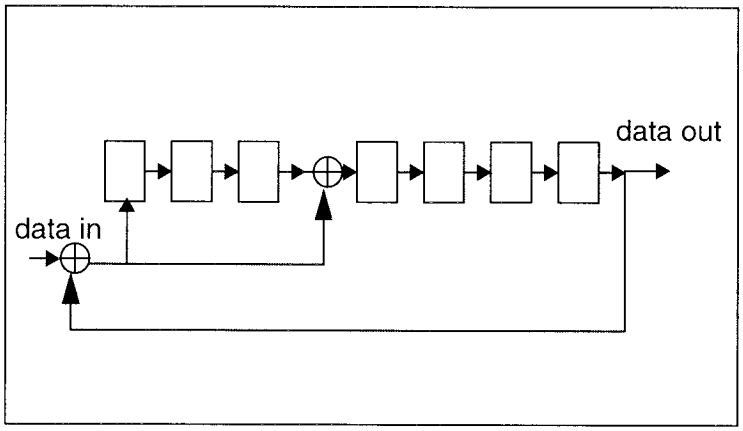


Figure 5-19 CRC7 Generator/Checker

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

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

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 poly-nomial. The code has a minimal distance d=4 and is used for a payload length of up to 2,048 bytes ($n \leq 16,383$).

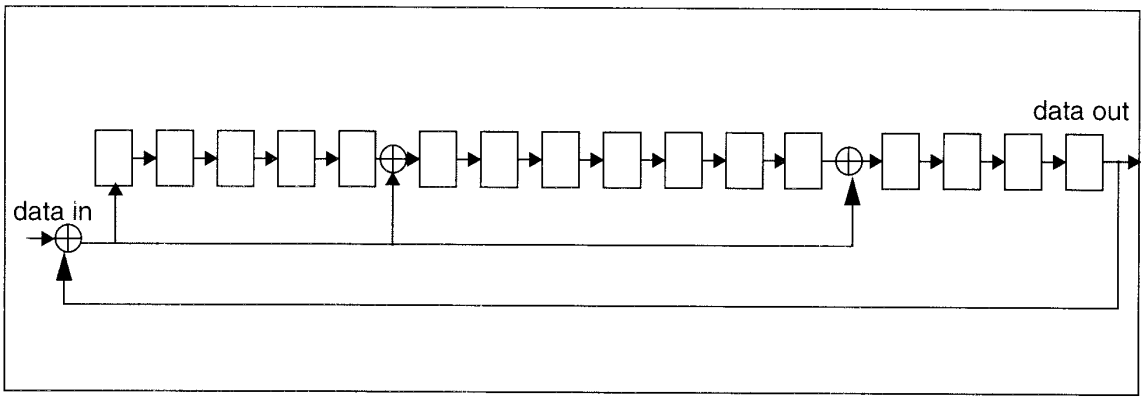


Figure 5-20 CRC16 Generator/Checker

6.0 SPI Protocol Definition

6.1 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 (multiples of 8 clocks) to the CS signal.

Similar to the MultiMediaCard protocol, the SPI messages are built from command, response and data-block tokens. All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.

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

- The selected card always responds to the command.
- An 8 or 16 bit response structure is used.
- When the card encounters a data retrieval problem, it will respond with an error response (which replaces the expected data block) rather than time-out as in the MultiMediaCard mode.

Only single block read write operations are supported in SPI mode. In addition to the command response, every data block sent to the card during write operations will be responded with a special data response token. A data block may be as big as one card sector and as small as a single byte.[1]

6.1.1 Mode Selection

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

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

The CMD (pin 2) and DAT[0] (pin 7) lines start up in "Open Drain"[2] mode and must be sourced externally. Once the card is in SPI mode, the lines are in "push-pull" mode until power is cycled.

Since the card defaults to MultiMediaCard mode after a power cycle, Pin 1 (CS) must be pulled low and CMD0 (40h) must be sent on the CMD (DataIn, pin 2) line in order for the card to enter SPI mode.

The default command structure/protocol for MultiMediaCard mode is to have CRC checking enabled.

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

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

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

2) See section 4.2.

6.1.2 Bus Transfer Protection

Every MultiMediaCard token transferred on the bus is protected by CRC bits. In SPI mode, the MultiMediaCard offers a non protected mode which enables systems built with reliable data links to exclude the hardware or firmware required for implementing the CRC generation and verification functions.

In the non protected mode the CRC bits of the command, response and data tokens are still required in the tokens however, they are defined as "don't cares" for the transmitters and ignored by the receivers.

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

6.1.2.1 Data Read

SPI mode supports single block read operation only (MultiMediaCard CMD17). Upon reception of a valid read command the card will respond with a response token followed by a data token in the length defined in a previous SET_BLOCK_LENGTH (CMD16) command (refer to Figure 6-1).

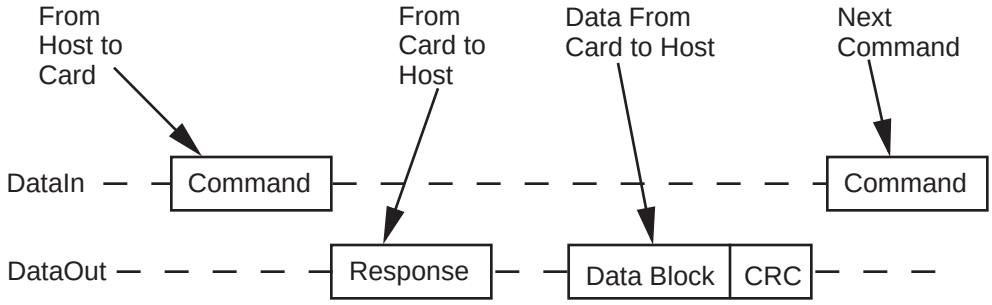


Figure 6-1 Read Operation

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

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

The maximum block length is defined by READ_BL_LEN (CSD parameter). If partial blocks are allowed (the CSD parameter READ_BL_PARTIAL equals 1) the block length can be any number between 1 and READ_BL_LEN. Otherwise the only valid block length for data read is READ_BL_LEN.

The start address can be any byte address in the valid address range of the card. Every block, however, must be contained in a single physical card sector unless write misalignment is allowed. Refer to section 1.5.8.6.

In case of data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 6-2 shows a data read operation which terminated with an error token rather than a data block.

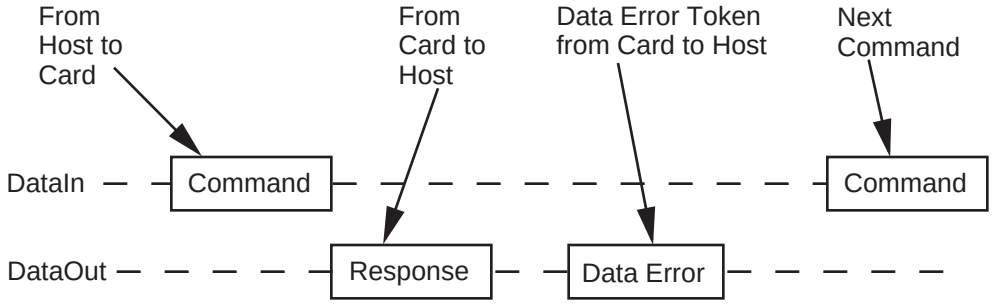


Figure 6-2 Read Operation - Data Error

6.1.2.2 Data Write

As for the read operation, while in SPI mode the MultiMediaCard supports single block write only. Upon reception of a valid write command (MultiMediaCard CMD24) the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix, block length and start address restrictions are identical to the read operation (refer to Figure 6-3).[1]

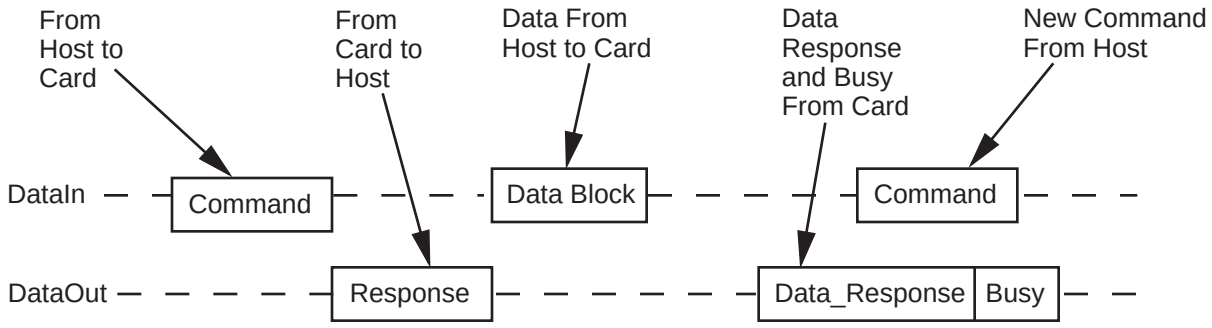


Figure 6-3 Write Operation

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

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

Resetting the CS signal while the card is busy, will not terminate the programming process. The card will release the dataOut line (tristate) and continue to program. If the card is reselected before the programming is done, the dataOut line will be forced back to low and all commands will be rejected.

Resetting a card (using CMD0) will terminate any pending or active programming operation. This may destroy the data formats on the card. It is the host's responsibility to prevent it.

6.1.2.3 Erase & Write Protect Management

The erase and write protect management procedures in the SPI mode are identical to the MultiMediaCard mode. While the card is erasing or changing the write protection bits of the predefined sector list it will be in a busy state and will hold the dataOut line low. Figure 6-4 illustrates a "no data" bus transaction with and without busy signaling.

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

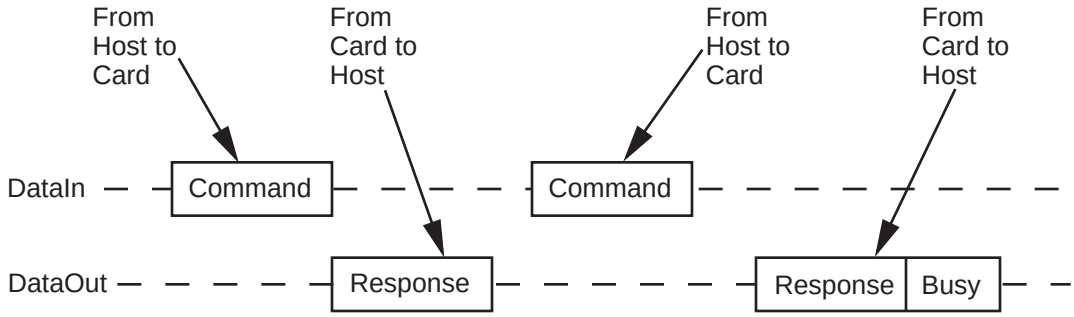


Figure 6-4 "No Data" Operations

6.1.2.4 Read CID/CSD Registers

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

6.1.2.5 Reset Sequence

The MultiMediaCard requires a defined reset sequence. After power on reset or CMD0 (software reset) the card enters an idle state. Note that at least 74 clock cycles are required prior to starting bus communication. At this state the only legal host command is CMD1 (SEND_OP_COND). In SPI mode, however, CMD1 has no operands.

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.

6.1.2.6 Error Conditions

Unlike the MultiMediaCard protocol, in the SPI mode the card will always respond to a command. The response indicates acceptance or rejection of the command. A command may be rejected if it is not supported (illegal opcode), if the CRC check failed, if it contained an illegal operand, or if it was out of sequence during an erase sequence.

6.1.3 Memory Array Partitioning

Same as for MultiMediaCard mode.

6.2 SPI Command Set

6.2.1 Command Format

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

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

Commands and arguments are listed in Table 6-4.

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

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

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

6.2.1.1 Detailed Command Description

The following table provides a detailed description of the SPI bus commands. The responses are defined in section 6.2.2. This table below lists all MultiMediaCard commands. A "yes" in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are reserved in MultiMediaCard mode as well.

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

Table 6-1 Description of SPI Bus Commands

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the 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				
CMD8	reserved				
CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID).
CMD11	No				
CMD12	No				
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register.
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read and write).[1]
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[2]

1) The default block length is as specified in the CSD (512 bytes). A set block length of less than 512 bytes will cause a write error. The only valid write set block length is 512 bytes. CMD16 is not mandatory if the default is accepted.
2) The data transferred must not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.

CMD18	No				
CMD19	reserved				
CMD20	No				
CMD21. . CMD23	reserved				
CMD24	Yes	[31:0] data address	R1[3]	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[4]
CMD25	No				
CMD26	No				
CMD27	Yes	None	R1b[5]	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. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_ PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.[6]
CMD31	reserved				
CMD32	Yes	[31:0] data address	R1	TAG_SECTOR_STA RT	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_GRO UP_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	R1	UNTAG_ERASE_	Removes one previously selected erase

3) Data followed by data response plus busy.
4) The data transferred must not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD.
5) R1b: R1 response with an optional trailing busy signal.
6) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line.

		address		GROUP	group from the erase selection.
CMD38	Yes	[31:0] don't cares*	R1b	ERASE	Erases all previously selected sectors.
CMD39	No				
CMD40	No				
CMD41. . CMD58	reserved				
CMD59	Yes	[31:1] don't cares* [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off
CMD60-63	No				

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

6.2.2 Responses

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

6.2.2.1 Format R1

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

The structure of the R1 format is given in Figure 6-5.

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

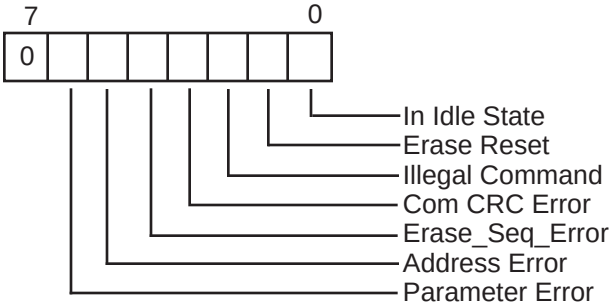


Figure 6-5 R1 Response Format

6.2.2.2 Format R1b

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

6.2.2.3 Format R2

This, 2 bytes long, response token is sent by the card as a response to the SEND_STATUS command. The format of the R2 status is given in Figure 6-6.

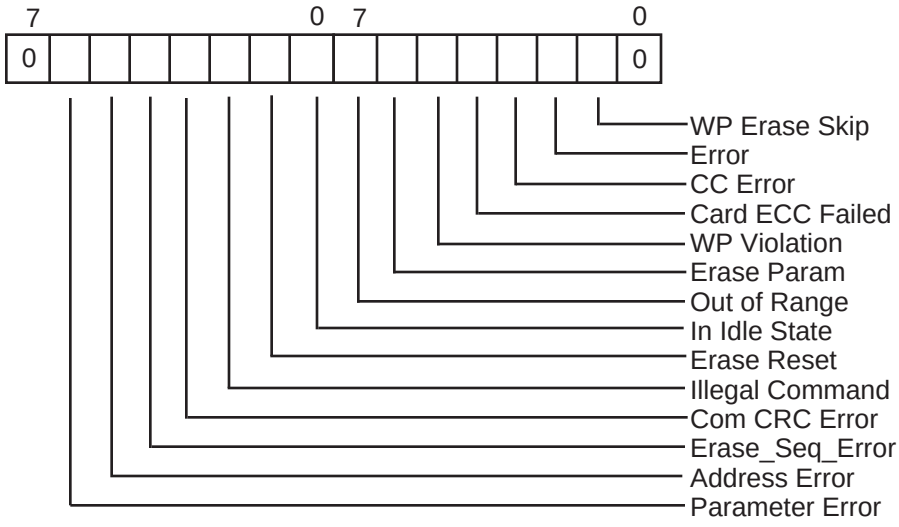


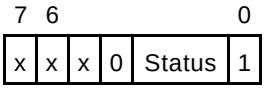
Figure 6-6 R2 Response Format

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

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

6.2.2.4 Data Response

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



The status bits may be:

‘010’ - Data accepted.

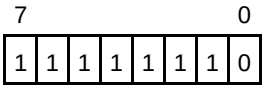
‘101’ - Data rejected due to a CRC error.

6.2.3 Data Tokens

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

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

- Byte 1: Start Byte



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

6.2.4 Data Error Token

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

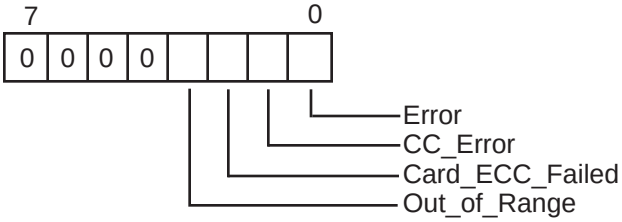


Figure 6-7 Data Error Token

The 4 LSBs are the same error bits as in response format R2.

6.3 Card Registers

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

6.4 SPI Bus Timing Diagrams

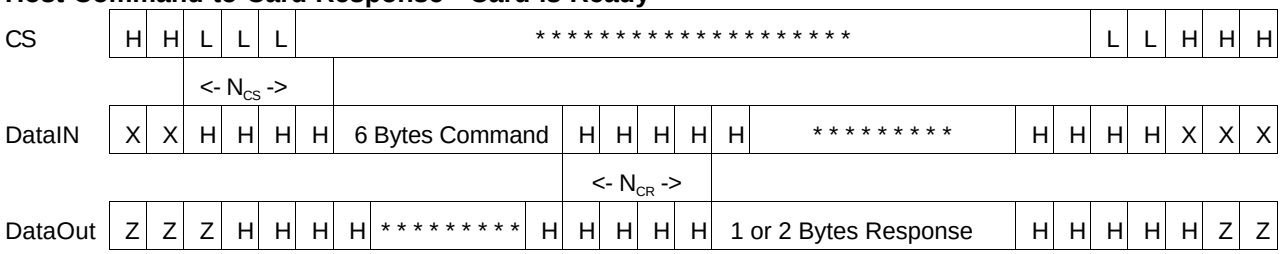
All timing diagrams use the following schematics and abbreviations:

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

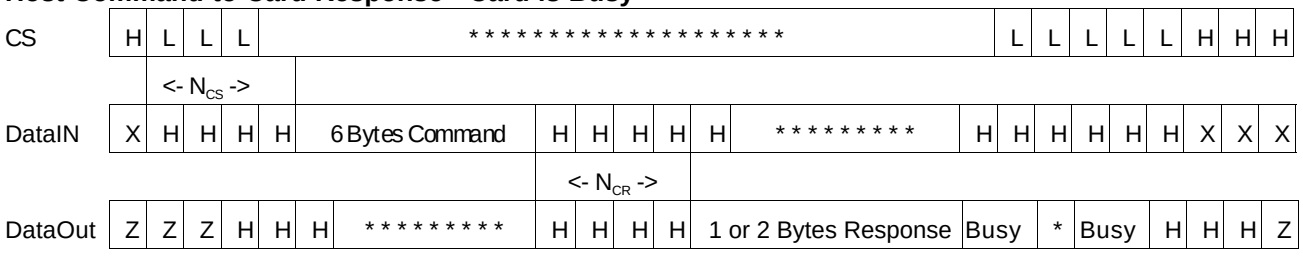
All timing values are defined in Table 6-2. The host must keep the clock running for at least N_{CR} clock cycles after the card response is received. This restrictions applied to command and data response tokens.

6.4.1 Command/Response

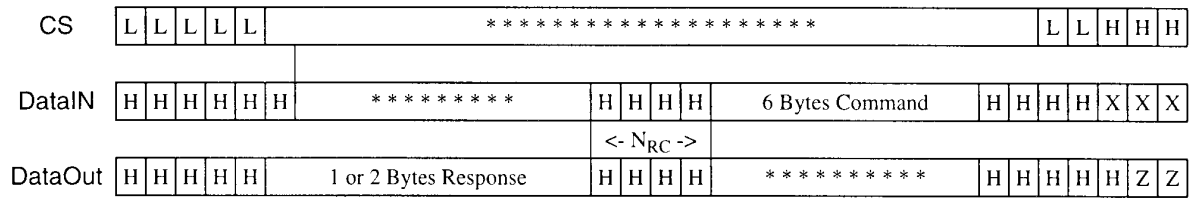
Host Command to Card Response - Card is Ready



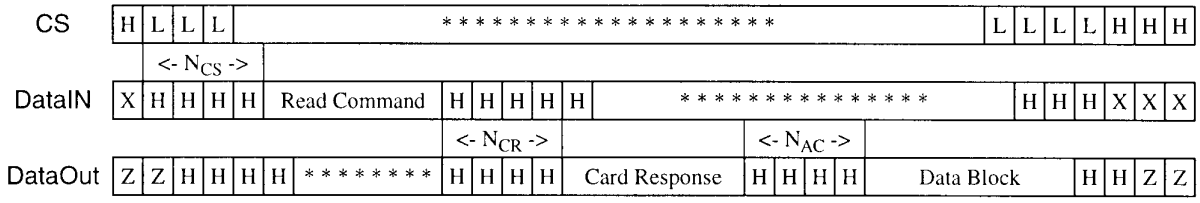
Host Command to Card Response - Card is Busy



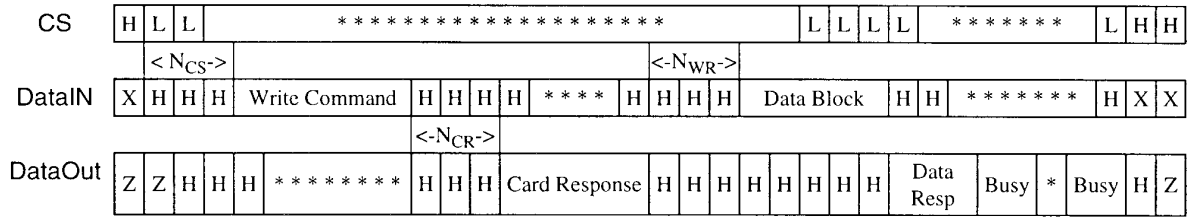
Card Response to Host Command



6.4.2 Data Read



6.4.2.1 Data Write



6.4.3 Timing Values

Table 6-2 Timing Constants Definitions

	Min	Max	Unit
N_{CS}	0	-	8 clock cycles
N_{CR}	1	8	8 clock cycles
N_{RC}	1	-	8 clock cycles
N_{AC}	1	spec. in the CSD	8 clock cycles
N_{WR}	1	-	8 clock cycles

6.5 SPI Electrical Interface

The SPI Mode electrical interface is identical to that of the MultiMediaCard mode with the exception of the programmable card output drivers option which is not supported in SPI mode.

6.6 SPI Bus Operating Conditions

Identical to MultiMediaCard mode.

6.7 Bus Timing

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

6.8 Error Protection

Identical to MultiMediaCard mode. See section 5.11 of this manual.

Ordering Information and Technical Support

Ordering Information

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

MultiMediaCard

Model SDMB-2	2.0 MB
SDMB-4	4.0 MB
SDMB-8	8.0 MB
SDMB-16*	16.0 MB
SDMB-32*	32 .1MB

*Preliminary information.

Technical Support Services

Direct SanDisk Technical Support

Call SanDisk Applications Engineering at 408-542-0405 for technical support.

SanDisk Worldwide Web Site

Internet users can obtain technical support and product information along with SanDisk news and much more from the SanDisk Worldwide Web Site, 24 hours a day, seven days a week. The SanDisk Worldwide Web Site is frequently updated. Visit this site often to obtain the most up-to-date information on SanDisk products and applications. The SanDisk Web Site URL is http://www.sandisk.com.

SanDisk Sales Offices

SanDisk Worldwide Sales Offices

Americas

SanDisk Corporate Headquarters
140 Caspian Court
Sunnyvale, CA 94089-9820
408-542-0500
FAX 408-542-0503
http://www.sandisk.com

Sales Offices

Northwest Region USA
408-542-0730
FAX 408-542-0403

Western Region USA
949-442-8370
FAX 949-442-8371

Central Region USA
614-760-3700
FAX 614-760-3701

New England & Canada
203-483-4390
FAX 203-483-4399

Mid-Atlantic Region USA
703-481-9828
FAX 703-437-9215

Southern Region USA
407-667-4880
FAX 407-667-4834

Latin & South America
407-667-4880
FAX 407-667-4834

Europe

SanDisk Corporation
Karlsruher Str. 2C
D-30519 Hannover, Germany
011-49-511-8759185
FAX 011-49-511-8759187

Southern Europe

SanDisk Corporation
4, rue de l'abreuvoir
92415 Courbevoie Cedex, France
011-33-1-4717-6510
FAX 011-33-1-4717-6531

Japan

SanDisk K.K.
8F Nisso Bldg. 15
2-17-19 Shin-Yokohama, Kohoku-ku
Yokohama 222-0033, Japan
81-45-474-0181
FAX 81-45-474-0371

Asia/Pacific Rim

89 Queensway, Lippo Center
Tower II, Suite 2207-9
Admiralty, Hong Kong
852-2712-0501
FAX 852-2712-9385

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

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 FlashDisk. In satisfaction of its obligations hereunder, SanDisk, at its sole option, will either repair, replace or refund the purchase price of the product.

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

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

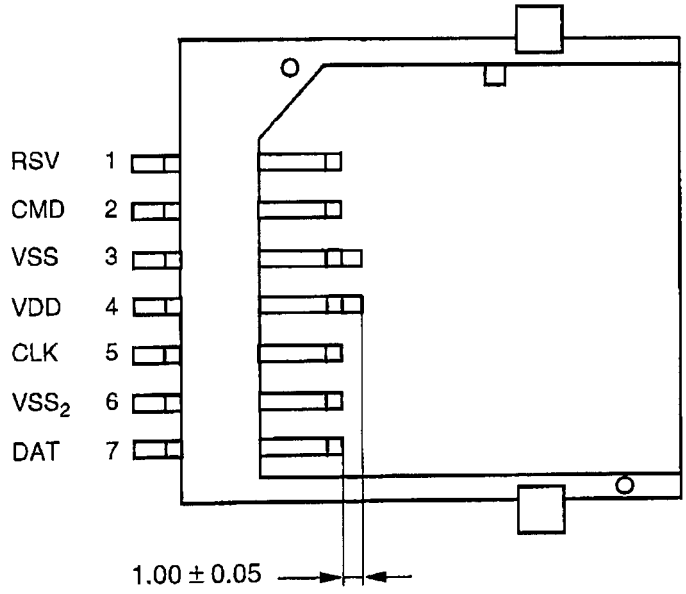
VI. OUT OF WARRANTY REPAIRS

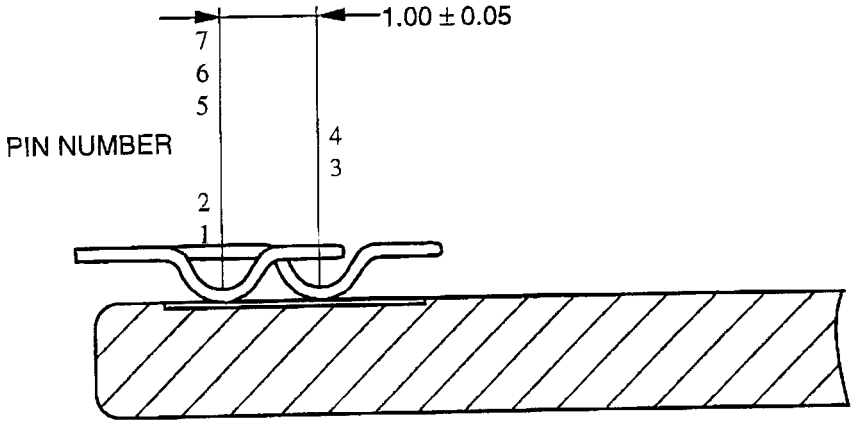
Please contact SanDisk Customer Service at 408-542-0595 for the current out of warranty and repair price list.

Appendix MultiMediaCard Connectors

MultiMediaCard Connector Vendors

Company	Contact	Phone	Fax
AMP	Eric Dickinson	717-810-4443	717-810-4523
AVX (Kyocera) Corp. (East Coast)	Craig Hunter	843-946-0601	843-626-5814
AVX (Kyocera) Corp. (West Coast)	Ken Yamada	408-436-5400	408-437-1500
ITT Canon	Bernard Poulain	33-1-6024-5110	33-1-6433-1682
JST Corp.	Kenji Iwahashi	408-734-7905	408-734-7901
JST Corp.	Hiro Handa or Nobuko Mitoh	81-045-543-1278	81-045-549-1300
Molex	Jeroen Deen	49-89-4130-9233	49-89-40-15-27

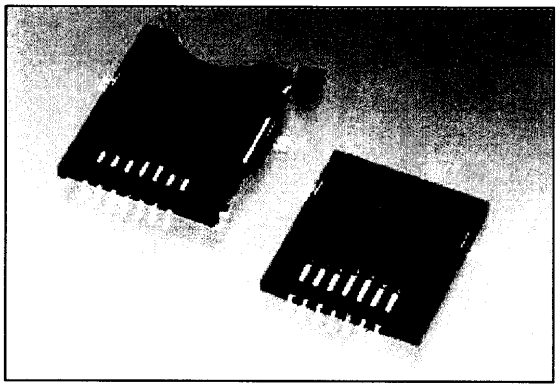




MultiMediaCard Host Connector

In Development

AMP MMC (Multi-Media Card) Connector



Description/ Applications

- The MMC connector is used to read an MMC card which addresses the need for small form factor portable memory in either ROM or RAM
- The MMC connector is a low profile, 7-position (2 first make/last break contacts) memory read socket to be used to read MMC cards
- MMC connectors are available in an eject and a non-eject version
- Applications include small hand held devices such as cell phones, data tracking, POS, PDA's, digital cameras, games, info toys, GPS, and digital maps

Product Strengths

- Improved reliability through the card pick-up location pegs
- Half-moon feature on front of eject version provides easier card grip
- Low 2.8mm profile
- Two sequencing levels eliminates the need for a switch resulting in reduced cost and improved reliability
- Similar proven contact design to SIM and Smart Card products

Availability/ Part Numbers

- Pre-production product is available now,
  - P/N 97-8415-01-1 without eject
  - P/N 97-8415-08-1 with eject

For More Information

• **Ron Jansen (Americas)** **rpjansen@amp.com**	**717-592-4991**	**Fax: 717-592-2470**
• **Mike Willis (Americas)** **mlwillis@amp.com**	**717-592-6409**	**Fax: 717-592-6361**
• **Maurice Laguette (EMEA)** **m.laguette@amp.com**	**+33-1-3420-8302**	**Fax: +33-1-3420-8398**
• **Keith Volz (Asia/Pacific)** **klvolz@amp.com**	**336-727-5721**	**Fax: 336-727-5532**

Connecting at a HIGHER level.™

AMP

5/98–2M