SD Specifications

Part 1

Physical Layer Specification

Version 2.00

May 9, 2006

SD Group

Matsushita Electric Industrial Co., Ltd. (Panasonic)
SanDisk Corporation
Toshiba Corporation

Technical Committee

SD Card Association

Revision History

Date	Version	Changes compared to previous issue
March 22, 2000	1.0	Base version (Draft only)
April 15, 2001	1.01	Initial release version
October 15, 2004	1.10	- Version 1.01 + supplementary notes Version 1.01e(March 21, 2004) - CMD6 (Switch Function command) is specified and CMD34-37, 50 and 57 are reserved for new command system. - High-Speed mode is specified.(Up to 25 MB/sec Read/Write rate) - eCommerce command set and Vendor Specific command set are specified
May 9, 2006	2.00	- Version 1.10 + supplementary notes Version 1.00. - High Capacity SD Memory Card is specified. (Up to and including 32 GB)

To the extent this proposed specification, which is being submitted for review under the IP Policy, implements, incorporates by reference or refers to any portion of versions 1.0 or 1.01 of the SD Specifications (including Parts 1 through 4), adoption of the proposed specification shall require Members utilizing the adopted specification to obtain the appropriate licenses from the SD-3C, LLC, as required for the utilization of those portion(s) of versions 1.0 or 1.01 of the SD Specifications.
For example, implementation of the SD Specifications in a host device under versions 1.0 or 1.01 and under the adopted specification requires the execution of a SD Host Ancillary License Agreement with the SD-3C, LLC; and implementation of the SD Specifications under versions 1.0 or 1.01 and under the proposed specification in a SD Card containing any memory storage capability (other than for storage of executable code for a controller or microprocessor within the SD Card) requires the execution of a SD Memory Card License Agreement with the SD-3C, LLC.

Conditions for publication

Publisher:

SD Card Association
2400 Camino Ramon, Suite 375
San Ramon, CA 94583 USA
Telephone: +1 (925) 275-6615,
Fax: +1 (925) 886-4870
E-mail: office@sdcard.org

Copyright Holders:

The SD Group
Matsushita Electric Industrial Co., Ltd. (Panasonic)
SanDisk Corporation (SanDisk)
Toshiba Corporation (Toshiba)
The SD Card Association

Notes:
The copyright of the previous versions (Version 1.00 and 1.01) and all corrections or non-material changes thereto are owned by SD Group.
The copyright of material changes to the previous versions (Version 1.01) are owned by SD Card Association.

Confidentiality:

The contents of this document are deemed confidential information of the SD-3C LLC and/or the SD Card Association (the "Disclosers"). As such, the contents and your right to use the contents are subject to the confidentiality obligations stated in the written agreement you entered into with the Disclosers which entitled you to receive this document, such as a Non-Disclosure Agreement, the License Agreement for SDA Memory Card Specifications (also known as "LAMS"), the SD Host/Ancillary Product License Agreement (also known as "HALA") or the IP Policy.

Disclaimers:

The information contained herein is presented only as a standard specification for SD Card and SD Host/Ancillary products. No responsibility is assumed by SD Card Association for any damages, any infringements of patents or other right of the third parties, which may result from its use. No license is granted by implication or otherwise under any patent or rights of SD Group and SD Card Association or others.

Conventions Used in This Document

Naming Conventions

- Some terms are capitalized to distinguish their definition from their common English meaning. Words not capitalized have their common English meaning.

Numbers and Number Bases

- Hexadecimal numbers are written with a lower case "h" suffix, e.g., FFFFh and 80h.
- Binary numbers are written with a lower case "b" suffix (e.g., 10b).
- Binary numbers larger than four digits are written with a space dividing each group of four digits, as in 1000 0101 0010b.
- All other numbers are decimal.

Key Words

- May: Indicates flexibility of choice with no implied recommendation or requirement.
- Shall: Indicates a mandatory requirement. Designers shall implement such mandatory requirements to ensure interchangeability and to claim conformance with the specification.
- Should: Indicates a strong recommendation but not a mandatory requirement. Designers should give strong consideration to such recommendations, but there is still a choice in implementation.

Application Notes

Some sections of this document provide guidance to the host implementers as follows:

Application Note: This is an example of an application note.

Table of Contents

1. General Description....1

2. System Features....3

3. SD Memory Card System Concept....5
3.1 Read-Write Property....5
3.2 Supply Voltage....5
3.3 Card Capacity....6
3.4 Speed Class....7
3.5 Bus Topology....8
3.5.1 SD Bus....8
3.5.2 SPI Bus....9
3.6 Bus Protocol....10
3.6.1 SD Bus....10
3.6.2 SPI Bus....13
3.7 SD Memory Card–Pins and Registers....14

4. SD Memory Card Functional Description....16
4.1 General....16
4.2 Card Identification Mode....17
4.2.1 Card Reset....17
4.2.2 Operating Condition Validation....17
4.2.3 Card Initialization and Identification Process....19
4.3 Data Transfer Mode....21
4.3.1 Wide Bus Selection/Deselection....23
4.3.2 2 GByte Card....23
4.3.3 Data Read....23
4.3.4 Data Write....24
4.3.5 Erase....26
4.3.6 Write Protect Management....26
4.3.7 Card Lock/Unlock Operation....27
4.3.7.1 General....27
4.3.7.2 Parameter and the Result of CMD42....29
4.3.7.3 Forcing Erase....31
4.3.7.3.1 Force Erase Function to the Locked Card....31
4.3.7.4 Relation Between ACMD6 and Lock/Unlock State....32
4.3.7.5 Commands Accepted for Locked Card....32
4.3.7.6 Two Types of Lock/Unlock Card....33
4.3.8 Content Protection....33
4.3.9 Application-Specific Commands....34
4.3.9.1 Application-Specific Command – APP_CMD (CMD55)....34
4.3.9.2 General Command - GEN_CMD (CMD56)....34
4.3.10 Switch Function Command....35
4.3.10.1 General....35
4.3.10.2 Mode 0 Operation - Check Function....37
4.3.10.3 Mode 1 Operation - Set Function....37

4.3.10.4 Switch Function Status.......... 39
4.3.10.4.1 Busy Status Indication for Functions 40
4.3.10.4.2 Data Structure Version 41
4.3.10.4.3 Function Table of Switch Command.......... 41
4.3.10.5 Relationship between CMD6 data & other commands 43
4.3.10.6 Switch Function Flow Example.......... 44
4.3.10.7 Example of Checking.......... 46
4.3.10.8 Example of Switching 47
4.3.11 High-Speed Mode (25 MB/sec interface speed).......... 48
4.3.12 Command System.......... 48
4.3.13 Send Interface Condition Command (CMD8) 49
4.3.14 Command Functional Difference in High Capacity SD Memory Card.......... 50
4.4 Clock Control.......... 51
4.5 Cyclic Redundancy Code (CRC) 52
4.6 Error Conditions.......... 54
4.6.1 CRC and Illegal Command 54
4.6.2 Read, Write and Erase Timeout Conditions.......... 54
4.6.2.1 Read 54
4.6.2.2 Write 54
4.6.2.3 Erase 54
4.7 Commands 55
4.7.1 Command Types 55
4.7.2 Command Format 55
4.7.3 Command Classes.......... 55
4.7.4 Detailed Command Description 58
4.8 Card State Transition Table 65
4.9 Responses.......... 67
4.9.1 R1 (normal response command):.......... 67
4.9.2 R1b.......... 67
4.9.3 R2 (CID, CSD register) 67
4.9.4 R3 (OCR register) 68
4.9.5 R6 (Published RCA response) 68
4.9.6 R7 (Card interface condition) 69
4.10 Two Status Information of SD Memory Card 70
4.10.1 Card Status 70
4.10.2 SD Status 74
4.11 Memory Array Partitioning 78
4.12 Timings 80
4.12.1 Command and Response 80
4.12.2 Data Read 82
4.12.3 Data Write 83
4.12.4 Timing Values.......... 86
4.13 Speed Class Specification 87
4.13.1 Allocation Unit (AU) 87
4.13.2 Recording Unit (RU).......... 87
4.13.3 Write Performance 87
4.13.4 Read Performance 88
4.13.5 Performance Curve Definition 89
4.13.6 Speed Class Definition 89
4.13.7 Consideration for Inserting FAT Update during Recording.......... 91
4.13.7.1 Measurement Condition to determine Average T_{Fw}.......... 91

4.13.7.2 Maximum FAT Write Time ... 91
4.13.8 Measurement Conditions and Requirements of the Speed Class ... 92
4.13.8.1 Measurement Conditions ... 92
4.13.8.2 Requirements of the Performance Parameters for Each Speed Class ... 92
4.13.8.3 Requirements of SD File System ... 92
4.14 Erase Timeout Calculation ... 93
4.14.1 Erase Unit ... 93
4.14.2 Case Analysis of Erase Time Characteristics ... 93
4.14.3 Method for Erase Large Areas ... 94
4.14.4 Calculation of Erase Timeout Value Using the Parameter Registers ... 94

5. Card Registers ... 95
5.1 OCR register ... 96
5.2 CID register ... 97
5.3 CSD Register ... 99
5.3.1 CSD_STRUCTURE ... 99
5.3.2 CSD Register (CSD Version 1.0) ... 100
5.3.3 CSD Register (CSD Version 2.0) ... 108
5.4 RCA register ... 111
5.5 DSR register (Optional) ... 111
5.6 SCR register ... 111

6. SD Memory Card Hardware Interface ... 113
6.1 Hot Insertion and Removal ... 114
6.2 Card Detection (Insertion/Removal) ... 114
6.3 Power Protection (Insertion/Removal) ... 115
6.4 Power Scheme ... 116
6.4.1 Power Up ... 116
6.4.2 Power Down and Power Cycle ... 117
6.5 Programmable Card Output Driver (Optional) ... 118
6.6 Bus Operating Conditions ... 120
6.6.1 Threshold Level for High Voltage Range ... 120
6.6.2 Threshold Level for Low Voltage Range ... 120
6.6.3 General ... 120
6.6.4 Current Consumption ... 120
6.6.5 Bus signal line load ... 121
6.6.6 Bus Signal Levels ... 122
6.7 Bus Timing (Default) ... 123
6.8 Bus Timing (High-Speed Mode) ... 125

7. SPI Mode ... 127
7.1 Introduction ... 127
7.2 SPI Bus Protocol ... 127
7.2.1 Mode Selection and Initialization ... 128
7.2.2 Bus Transfer Protection ... 130
7.2.3 Data Read ... 130
7.2.4 Data Write ... 131
7.2.5 Erase & Write Protect Management ... 132
7.2.6 Read CID/CSD Registers ... 133
7.2.7 Reset Sequence ... 133
7.2.8 Error Conditions ... 133

7.2.9 Memory Array Partitioning ... 133
7.2.10 Card Lock/Unlock ... 133
7.2.11 Application Specific Commands ... 133
7.2.12 Content Protection Command ... 133
7.2.13 Switch Function Command ... 134
7.2.14 High-Speed Mode ... 134
7.2.15 Speed Class Specification ... 134
7.3 SPI Mode Transaction Packets ... 135
7.3.1 Command Tokens ... 135
7.3.1.1 Command Format ... 135
7.3.1.2 Command Classes ... 135
7.3.1.3 Detailed Command Description ... 136
7.3.1.4 Card Operation for CMD8 in SPI mode ... 141
7.3.2 Responses ... 143
7.3.2.1 Format R1 ... 143
7.3.2.2 Format R1b ... 143
7.3.2.3 Format R2 ... 144
7.3.2.4 Format R3 ... 145
7.3.2.5 Formats R4 & R5 ... 145
7.3.2.6 Format R7 ... 145
7.3.3 Control Tokens ... 146
7.3.3.1 Data Response Token ... 146
7.3.3.2 Start Block Tokens and Stop Tran Token ... 146
7.3.3.3 Data Error Token ... 147
7.3.4 Clearing Status Bits ... 147
7.4 Card Registers ... 148
7.5 SPI Bus Timing Diagrams ... 149
7.5.1 Command/Response ... 150
7.5.1.1 Host Command to Card Response - Card is ready ... 150
7.5.1.2 Host Command to Card Response - card is busy ... 150
7.5.1.3 Card Response to Host Command ... 150
7.5.2 Data Read ... 151
7.5.2.1 Timing of Single Block Read Operation ... 151
7.5.2.2 Stop Transmission Timing of Multiple Block Read Operation ... 151
7.5.2.3 Reading the CSD or CID register ... 151
7.5.3 Data Write ... 152
7.5.3.1 Timing of Multiple Block Write Operation ... 152
7.5.3.2 Stop Transmission Timing of Multiple Block Write Operation ... 152
7.5.4 Timing Values ... 153
7.6 SPI Electrical Interface ... 153
7.7 SPI Bus Operating Conditions ... 153
7.8 Bus Timing ... 153

8. SD Memory Card Mechanical Specification ... 154
8.1 Card Package ... 155
8.1.1 Design and Format ... 155
8.1.2 Reliability and Durability ... 155
8.1.3 Electrical Static Discharge (ESD) Requirement ... 156
8.1.4 Quality Assurance ... 156
8.2 Mechanical Form Factor ... 157
8.3 System: card and connector ... 160
8.3.1 Card Hot Insertion ... 160

8.3.2 Inverse Insertion 161
8.3.3 Card Orientation 161
8.3.4 Write Protect Detection 161
8.4 Thin (1.4 mm) Standard Size SD Memory Card 161
8.5 Nonconductive Area 163

Appendix A 164
A.1 Power Supply Decoupling 164
A.2 Connector 165
A.2.1 General 165
A.2.2 Card Insertion and Removal 165
A.2.3 Characteristics 167
A.3 Related Documentation 168

Appendix B 169
B.1 Abbreviations and terms 169

Table of Figures

Figure 1-1: SD Specifications Documentation Structure 1
Figure 3-1: Hosts-Cards Usability 6
Figure 3-2: SD Memory Card System Bus Topology 8
Figure 3-3: SD Memory Card System (SPI Mode) Bus Topology 9
Figure 3-4: “no response” and “no data” Operations 10
Figure 3-5: (Multiple) Block Read Operation 10
Figure 3-6: (Multiple) Block Write Operation 11
Figure 3-7: Command Token Format 11
Figure 3-8: Response Token Format 11
Figure 3-9: Data Packet Format - Usual Data 12
Figure 3-10: Data Packet Format - Wide Width Data 13
Figure 3-11: SD Memory Card Shape and Interface (Top View) 14
Figure 3-12: SD Memory Card Architecture 15
Figure 4-1: SD Memory Card State Diagram (card identification mode) 18
Figure 4-2: Card Initialization and Identification Flow (SD mode) 20
Figure 4-3: SD Memory Card State Diagram (data transfer mode) 21
Figure 4-4: Use of Switch Command 36
Figure 4-5: Busy Status of ‘Command System’ 40
Figure 4-6: CMD12 during CMD6; Case 1 43
Figure 4-7: CMD12 during CMD6; Case 2 43
Figure 4-8: Example of Switch Function flow 44
Figure 4-9: Switching Function Flow 45
Figure 4-10: CRC7 Generator/Checker 52
Figure 4-11: CRC16 Generator/Checker 53
Figure 4-12: Write Protection Hierarchy 79
Figure 4-13: Identification Timing (card identification mode) 80
Figure 4-14: SEND_RELATIVE_ADDR Timing 80
Figure 4-15: Command Response Timing (data transfer mode) 81
Figure 4-16: Timing of Response End to next CMD Start (data transfer mode) 81
Figure 4-17: Timing of Command Sequences (all modes) 81
Figure 4-18: Timing of Single Block Read Command 82
Figure 4-19: Timing of Multiple Block Read Command 82
Figure 4-20: Timing of Stop Command (CMD12, data transfer mode) 82
Figure 4-21: Timing of Single Block Write Command 83
Figure 4-22: Timing of Multiple Block Write Command 84
Figure 4-23: Stop Transmission during Data Transfer from the Host 84
Figure 4-24: Stop Transmission during CRC Status Transfer from the Card 84
Figure 4-25: Stop Transmission received after Last Data Block. Card is busy programming. 85
Figure 4-26: Stop Transmission received after Last Data Block. Card becomes busy. 85
Figure 4-27: Definition of Allocation Unit (AU) 87
Figure 4-28: Example of Writing Fragmented AU 87
Figure 4-29: Card Performances between 16 RUs 89
Figure 4-30: Three Performance Curves 90
Figure 4-31: Typical Sequence of FAT Update 91
Figure 4-32: Example Erase Characteristics (Case 1 T_{OFFSET}=0) 93
Figure 4-33: Example Erase Characteristics (Case 2 T_{OFFSET}=2) 94
Figure 5-1: ERASE_BLK_EN = 0 Example 105
Figure 5-2: ERASE_BLK_EN = 1 Example 105
Figure 6-1: Bus Circuitry Diagram 113

Figure 6-2: Improper Power Supply....115
Figure 6-3: Short Circuit Protection....115
Figure 6-4: Power-up Diagram....116
Figure 6-5: SD Memory Card Bus Driver....119
Figure 6-6: Bus Signal Levels....122
Figure 6-7: Timing Diagram Data Input/Output Referenced to Clock (Default)....123
Figure 6-8: Timing Diagram Data Input/Output Referenced to Clock (High-Speed)....125
Figure 7-1: SD Memory Card State Diagram (SPI mode)....128
Figure 7-2: SPI Mode Initialization Flow....129
Figure 7-3: Single Block Read Operation....130
Figure 7-4: Read Operation - Data Error....131
Figure 7-5: Multiple Block Read Operation....131
Figure 7-6: Single Block Write Operation....131
Figure 7-7: Multiple Block Write Operation....132
Figure 7-8: 'No data' Operations....132
Figure 7-9: R1 Response Format....143
Figure 7-10: R2 Response Format....144
Figure 7-11: R3 Response Format....145
Figure 7-12: R7 Response Format....145
Figure 7-13: Data Error Token....147
Figure 7-14: Basic Command Response....150
Figure 7-15: Command Response with Busy Indication (R1b)....150
Figure 7-16: Timing between Card Response to new Host Command....150
Figure 7-17: Read Single Block Operations - Bus Timing....151
Figure 7-18: Stop Transmission in Read Multiple Block....151
Figure 7-19: Read CSD/CID - Bus Timing....151
Figure 7-20: Write Operation - Bus Timing....152
Figure 7-21: Stop Transmission in Write Multiple Block....152
Figure 8-1: SD Memory Card - Mechanical Description (1 out of 3)....157
Figure 8-2: SD Memory Card - Mechanical Description (2 out of 3)....158
Figure 8-3: SD Memory Card - Mechanical Description (3 out of 3)....159
Figure 8-4: WP Switch Definition for All Cases and Card Types....160
Figure 8-5: Inverse Insertion....161
Figure 8-6: Mechanical Drawing of Thin SD Memory Card (1 out of 2)....162
Figure 8-7: Mechanical Drawing of Thin SD Memory Card (2 out of 2)....162
Figure 8-8: Nonconductive Areas of SD Memory Card....163
Figure 9-1: Power Supply Decoupling....164
Figure 9-2: Modified SD Memory Card Connector for Hot Insertion....166

Table of Tables

Table 3-1: SD Memory Card Pad Assignment 14
Table 3-2: SD Memory Card Registers 15
Table 4-1: Overview of Card States vs. Operation Modes 16
Table 4-2: Read Command Blocklen 24
Table 4-3: Write Command Blocklen 24
Table 4-4: Lock Card Data Structure 27
Table 4-5: Lock Unlock Function (Basic Sequence for CMD42) 30
Table 4-6: Force Erase Function to the Locked Card (Relation to the Write Protects) 31
Table 4-7: Relation between ACMD6 and the Lock/Unlock State 32
Table 4-8: Version Difference of Lock/Unlock Functions 33
Table 4-9: Available Functions 38
Table 4-10: Status Data Structure 40
Table 4-11: Data Structure Version 41
Table 4-12: Status Code of Mode 0 to Supported Function Group 41
Table 4-13: Status Code of Mode 1 to Supported Function Group 42
Table 4-14: Status Code of Mode 0 and 1 to Unsupported Function Group 42
Table 4-15: Format of CMD8 49
Table 4-16: Card Operation for CMD8 in SD Mode 50
Table 4-17: Command Format 55
Table 4-18: Card Command Classes (CCCs) 57
Table 4-19: Basic Commands (class 0) 59
Table 4-20: Block-Oriented Read Commands (class 2) 59
Table 4-21: Block-Oriented Write Commands (class 4) 60
Table 4-22: Block Oriented Write Protection Commands (class 6) 61
Table 4-23: Erase Commands (class 5) 61
Table 4-24: Lock Card (class 7) 62
Table 4-25: Application-specific Commands (class 8) 62
Table 4-26: I/O Mode Commands (class 9) 63
Table 4-27: Application Specific Commands used/reserved by SD Memory Card 64
Table 4-28: Switch Function Commands (class 10) 64
Table 4-29: Card State Transition Table 66
Table 4-30: Response R1 67
Table 4-31: Response R2 67
Table 4-32: Response R3 68
Table 4-33: Response R6 68
Table 4-34: Response R7 69
Table 4-35: Voltage Accepted in R7 69
Table 4-36: Card Status 72
Table 4-37: Card Status Field/Command - Cross Reference 73
Table 4-38: SD Status 75
Table 4-39: Speed Class Code Field 75
Table 4-40: Performance Move Field 75
Table 4-41: AU_SIZE Field 76
Table 4-42: Maximum AU size 76
Table 4-43: Erase Size Field 76
Table 4-44: Erase Timeout Field 77
Table 4-45: Erase Offset Field 77
Table 4-46: Timing Diagram Symbols 80
Table 4-47: Timing Values 86

Table 4-48: Measurement Condition of Speed Class ... 92
Table 4-49: Performance Requirements for Each Class ... 92
Table 5-1: OCR Register Definition ... 96
Table 5-2: The CID Fields ... 97
Table 5-3: CSD Register Structure ... 99
Table 5-4: The CSD Register Fields (CSD Version 1.0) ... 100
Table 5-5: TAAC Access Time Definition ... 101
Table 5-6: Maximum Data Transfer Rate Definition ... 101
Table 5-7: Supported Card Command Classes ... 102
Table 5-8: Data Block Length ... 102
Table 5-9: DSR Implementation Code Table ... 103
Table 5-10: $V_{DD, min}$ Current Consumption ... 103
Table 5-11: $V_{DD, max}$ Current Consumption ... 103
Table 5-12: Multiply Factor for the Device Size ... 104
Table 5-13: R2W_FACTOR ... 106
Table 5-14: Data Block Length ... 106
Table 5-15: File Formats ... 107
Table 5-16: The CSD Register Fields (CSD Version 2.0) ... 108
Table 5-17: The SCR Fields ... 111
Table 5-18: SCR Register Structure Version ... 111
Table 5-19: Physical Layer Specification Version ... 112
Table 5-20: SD Supported Security Algorithm ... 112
Table 5-21: SD Memory Card Supported Bus Widths ... 112
Table 6-1: DSR Register Contents ... 118
Table 6-2: Threshold Level for High Voltage ... 120
Table 6-3: Bus Operating Conditions – General ... 120
Table 6-4: Bus Operating Conditions - Signal Line's Load ... 121
Table 6-5: Bus Timing - Parameters Values (Default) ... 124
Table 6-6: Bus Timing - Parameters Values (High-Speed) ... 126
Table 7-1: Command Format ... 135
Table 7-2: Command Classes in SPI Mode ... 136
Table 7-3: Commands and Arguments ... 140
Table 7-4: Application Specific Commands used/reserved by SD Memory Card - SPI Mode ... 141
Table 7-5: Card Operation for CMD8 in SPI Mode ... 142
Table 7-6: SPI Mode Status Bits ... 148
Table 7-7: Timing Values ... 153
Table 8-1: SD Memory Card Package – Dimensions ... 155
Table 8-2: Reliability and Durability ... 155

1. General Description

SD Memory Card is a memory card that is specifically designed to meet the security, capacity, performance, and environment requirements inherent in newly emerging audio and video consumer electronic devices. The SD Memory Card will include a content protection mechanism that complies with the security of the SDMI standard and will be faster and capable of higher Memory capacity. The SD Memory Card security system uses mutual authentication and a "new cipher algorithm" to protect against illegal usage of the card content. A Non-secure access to the user's own content is also available.

SD memory cards may also support a second security system based on commonly used standards, such as ISO-7816, which can be used to interface the SD memory card into public networks and other systems supporting mobile e-commerce and digital signature applications.

SDIO

In addition to the SD Memory Card, there is the SD I/O (SDIO) Card. The SDIO Card specification is defined in a separate specification named: "SDIO Card Specification" that can be obtained from the SD Association. The SDIO Specification defines an SD card that may contain interfaces between various I/O units and an SD Host. The SDIO card may contain memory storage capability as well as its I/O functionality. The Memory portion of SDIO card shall be fully compatible to the given SD Memory Card specification. The SDIO card is based on and compatible with the SD Memory card. This compatibility includes mechanical, electrical, power, signalling, and software. The intent of the SD I/O card is to provide high-speed data I/O with low power consumption for mobile electronic devices. A primary goal is that an I/O card inserted into a non-SDIO aware host will cause no physical damage or disruption of that device or its software. In this case, the I/O card should simply be ignored. Once inserted into an SDIO aware host, the detection of the card will be via the normal means described in the given SD Physical Specification with some extensions that are described in the SDIO Specification.

The SD Memory Card communication is based on an advanced 9-pin interface (Clock, Command, 4xData and 3xPower lines) designed to operate in at maximum operating frequency of 50 MHz and low voltage range. The communication protocol is defined as a part of this specification.

The SD Specifications are divided into several documents. The SD Specifications documentation structure is given in Figure 1-1.

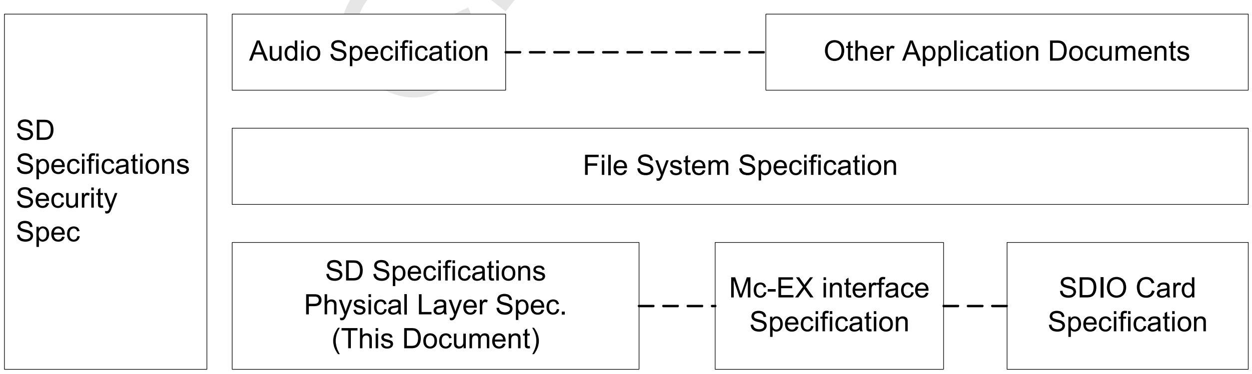


Figure 1-1: SD Specifications Documentation Structure

- **Audio Specification:**
  This specification, along with other application specifications, describes the specification of a specific application (in this case - Audio Application) and the requirements to implement it.

- **File System Specification:**
  The specification describes the specification of the file format structure of the data saved in the SD

Memory Card (in protected and un-protected areas).

- **Security Specification:**
  The specification describes the content protection mechanism and the application-specific commands that support it.

- **Physical Layer Specification (this document):**
  The specification describes the physical interface and the command protocol used by the SD Memory Card.
  The purpose of the Physical Layer specification is to define the SD Memory Card, its environment, and handling.
  The document is divided into several portions. Chapter 3 gives a general overview of the system concepts. The common SD Memory Card characteristics are described in Chapter 4. As this description defines an overall set of card properties, we recommend using the product documentation in parallel. The card registers are described in Chapter 5.
  Chapter 6 defines the electrical parameters of the SD Memory Card's hardware interface.
  Chapter 8 describes the physical and mechanical properties of the SD Memory Cards and the minimal recommendations to the card slots or cartridges.
  As used in this document, "shall" or "will" denote a mandatory provision of the standard. "Should" denotes a provision that is recommended but is not mandatory. "May" denotes a feature, which may or may not be present–at the option of the implementer–and whose presence does not preclude compliance.

- **Mc-EX Interface Specification:** (This section was added in version 1.10)
  Part A1 of the SD memory card specification (refer to Figure 1-1) serves as an extension to the SD card Physical Layer Specification and provides all of the definitions required to transfer the Mobile Commerce Extension (Mc-EX) command packets from the Mc-EX host to the Mc-EX enabled SD memory card, and vice versa.

2. System Features

- Targeted for portable and stationary applications
- Memory capacity:

  Standard Capacity SD Memory Card: Up to and including 2 GB

  High Capacity SD Memory Card: More than 2GB (This version of specification limits capacity up to and including 32GB)
- Voltage range:

  High Voltage SD Memory Card – Operating voltage range: 2.7-3.6 V

  Dual Voltage SD Memory Card – Operating voltage range: Low Voltage Range (T.B.D) and 2.7-3.6 V
- Designed for read-only and read/write cards.
- Default mode: Variable clock rate 0 - 25 MHz, up to 12.5 MB/sec interface speed (using 4 parallel data lines)
- High-Speed mode: Variable clock rate 0 - 50 MHz, up to 25 MB/sec interface speed (using 4 parallel data lines)
- Switch function command supports High-Speed, eCommerce, and future functions
- Correction of memory field errors
- Card removal during read operation will never harm the content
- Content Protection Mechanism - Complies with highest security of SDMI standard.
- Password Protection of cards (CMD42 - LOCK_UNLOCK)
- Write Protect feature using mechanical switch
- Built-in write protection features (permanent and temporary)
- Card Detection (Insertion/Removal)
- Application specific commands
- Comfortable erase mechanism

- **Protocol attributes of the communication channel:**

SD Memory Card Communication Channel
Six-wire communication channel (clock, command, 4 data lines)
Error-protected data transfer
Single or Multiple block oriented data transfer

- **SD Memory Card Form-factor**
  Standard Size SD Memory Card: Specified in this specification (See Chapter 6 and 8)
  miniSD Memory Card: Specified in "miniSD Memory Card Specification"
  microSD Memory Card: Specified in "microSD Memory Card Specification"

- **Standard Size SD Memory Card thickness is defined as both 2.1 mm (normal) and 1.4 mm (Thin SD Memory Card).**

All features in this specification are applied to Standard Size SD Memory Card.

3. SD Memory Card System Concept

The SD Memory Card provides application designers with a low cost mass storage device, implemented as a removable card that supports a high security level for content protection and a compact, easy-to-implement interface.
SD Memory Cards can be grouped into several card classes that differ in the functions they provide (given by the subset of SD Memory Card system commands).

An SD Memory Card system includes the SD Memory Card (or several cards) the bus and their Host/Application. The Host and Application specification is beyond the scope of this document. The following sections provide an overview of the card, bus topology, and communication protocols of the SD Memory Card system. The content protection (security) system description is given in "SD Memory Card Security Specification" document.

3.1 Read-Write Property

In terms of read-write property, two types of SD Memory Cards are defined:

- Read/Write (RW) cards (Flash, One Time Programmable - OTP, Multiple Time Programmable - MTP). These cards are typically sold as blank (empty) media and are used for mass data storage, end user video, audio or digital image recording
- Read Only Memory (ROM) cards. These cards are manufactured with fixed data content. They are typically used as a distribution media for software, audio, video etc.

3.2 Supply Voltage

In terms of operating supply voltage, two types of SD Memory Cards are defined:

- High Voltage SD Memory Cards that can operate within the voltage range of 2.7-3.6 V.
- Dual Voltage SD Memory Cards –Dual Voltage SD Memory Cards that can operate within the voltage range of Low Voltage Range (T.B.D) and 2.7-3.6 V.

Note that details of Dual Voltage SD Memory Card will be defined in future specification.

3.3 Card Capacity

In terms of capacity, two types of SD Memory Cards are defined:

- Standard Capacity SD Memory Cards supports capacity up to and including 2 G bytes (2^{31} bytes). All versions of the Physical Specifications define the Standard Capacity SD Memory Card.
- High Capacity SD Memory Cards supports capacity more than 2 G bytes (2^{31} bytes) and this version of specification limits capacity up to and including 32 GB. High Capacity SD Memory Card is newly defined from the Physical Layer Specification Version 2.00.

Only hosts that are compliant to the Physical Layer Specification version 2.00 or higher and the SD File System Specification Ver2.00 can access High Capacity SD Memory Cards. Other hosts fail to initialize High Capacity SD Memory Cards (See Figure 3-1).

Note:

1. **The Part 1 Physical Layer Specification Version 2.00 and Part 2 File System Specification Version 2.00 allow Standard Capacity SD Memory Cards to have capacity up to and including 2 GB and High Capacity SD Memory Cards to have capacity up to and including 32 GB. SD Memory Cards with a capacity greater than 32 GB will be available with updated versions of Part 1 and Part 2 Specifications.**
2. **Hosts that can access (read and/or write) SD Memory Cards with a capacity greater than 2 GB and up to and including 32 GB, shall also be able to access SD Memory Cards with a capacity of 2 GB or less.**

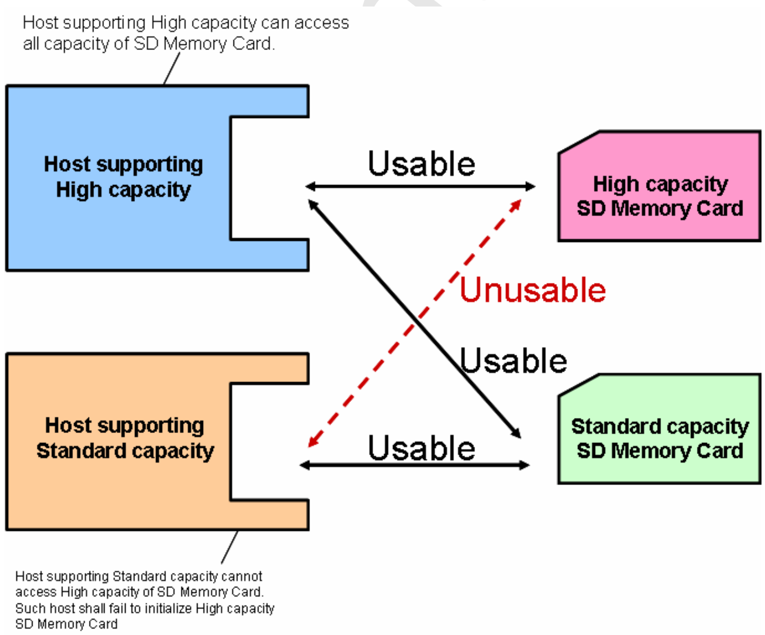


Figure 3-1: Hosts-Cards Usability

- 2 types of High Capacity SD Memory Card are specified. Type A (Single State Card) has single High Capacity memory area. Details of Type A are specified in the Physical Layer Specification version 2.00. Type B (Dual State Card) has both High Capacity memory area and Standard Capacity memory area. In Type B card, only one memory area can be used at any given time. A mechanical switch is used to select the desired memory area. Details of Type B will be defined in future specifications. It is not necessary for the host to distinguish card types.

3.4 Speed Class

Four Speed Classes are defined and indicate minimum performance of the cards

- Class 0 - These class cards do not specify performance. It includes all the legacy cards prior to this specification, regardless of its performance
- Class 2 - Are more than or equal to 2 MB/sec performance.
- Class 4 - Are more than or equal to 4 MB/sec performance.
- Class 6 - Are more than or equal to 6 MB/sec performance.

High Capacity SD Memory Cards shall support Speed Class Specification and have performance more than or equal to Class 2.

Note that the unit of performance [MB/sec] indicates 1000x1000 [Byte/sec] while the unit of data size [MB] indicates 1024x1024 [Byte]. This is because the maximum SD Bus speed is specified by the maximum SD clock frequency (25 [MB/sec] = 25000000 [Byte/sec] at 50 MHz) and data size is based on memory boundary (power of 2).

3.5 Bus Topology

The SD Memory Card system defines two alternative communication protocols: SD and SPI. The host system can choose either one of modes. The card detects which mode is requested by the host when the reset command is received and expects all further communication to be in the same communication mode. Common bus signals for multiple card slots are not recommended. A single SD bus should connect a single SD card. Where the host system supports a high-speed mode, a single SD bus shall be connected to a single SD card.

3.5.1 SD Bus

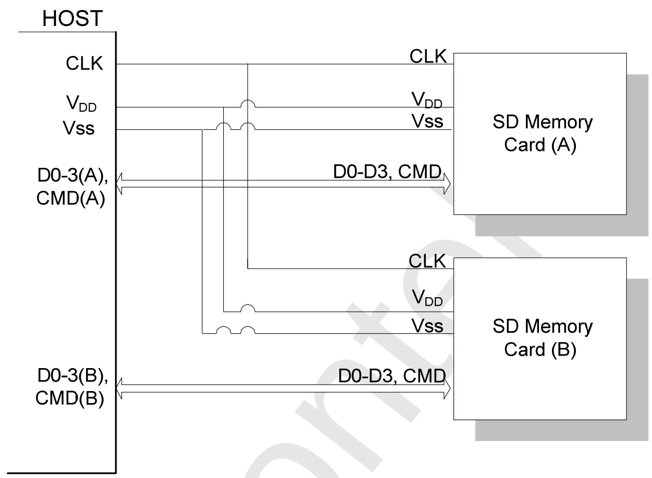


Figure 3-2: SD Memory Card System Bus Topology

The SD bus includes the following signals:

CLK:	Host to card clock signal
CMD:	Bidirectional Command/Response signal
DAT0 - DAT3:	4 Bidirectional data signals.
V_{DD}, V_{SS1}, V_{SS2}:	Power and ground signals.

The SD Memory Card bus has a single master (application), multiple slaves (cards), synchronous star topology (refer to Figure 3-2). Clock, power and ground signals are common to all cards. Command (CMD) and data (DAT0 - DAT3) signals are dedicated to each card providing continues point to point connection to all the cards.
During initialization process commands are sent to each card individually, allowing the application to detect the cards and assign logical addresses to the physical slots. Data is always sent (received) to (from) each card individually. However, in order to simply the handling of the card stack, after the initialization process, all commands may be sent concurrently to all cards. Addressing information is provided in the command packet.
SD bus allows dynamic configuration of the number of data lines. After power up, by default, the SD Memory Card will use only DAT0 for data transfer. After initialization the host can change the bus width (number of active data lines). This feature allows easy trade off between HW cost and system performance. **Note that while DAT1-DAT3 are not in use, the related Host's DAT lines should be in tri-state (input mode). For SDIO cards DAT1 and DAT2 are used for signaling.**

3.5.2 SPI Bus

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

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

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

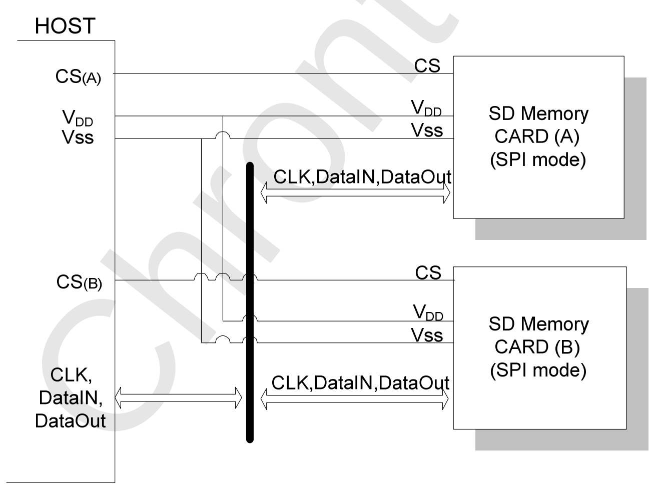


Figure 3-3: SD Memory Card System (SPI Mode) Bus Topology

The card identification and addressing methods are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal (see Figure 3-3).
The CS signal shall be continuously active for the duration of the SPI transaction (command, response and data). The only exception occurs during card programming, when the host can de-assert the CS signal without affecting the programming process.
The SPI interface uses the 7 out of the SD 9 signals (DAT1 and DAT 2 are not used, DAT3 is the CS signal) of the SD bus.

3.6 Bus Protocol

3.6.1 SD Bus

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

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

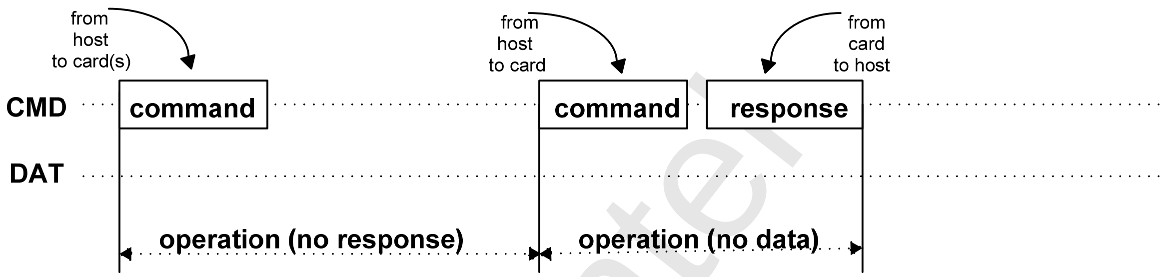


Figure 3-4: "no response" and "no data" Operations

Card addressing is implemented using a session address, assigned to the card during the initialization phase. The structure of commands, responses and data blocks is described in Chapter 4. The basic transaction on the SD bus is the command/response transaction (refer to Figure 3-4). This type of bus transaction transfers their information directly within the command or response structure. In addition, some operations have a data token.

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

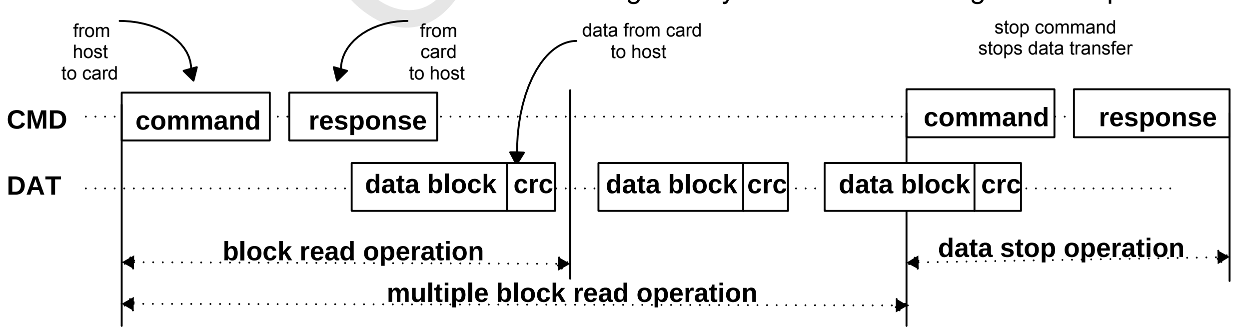


Figure 3-5: (Multiple) Block Read Operation

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

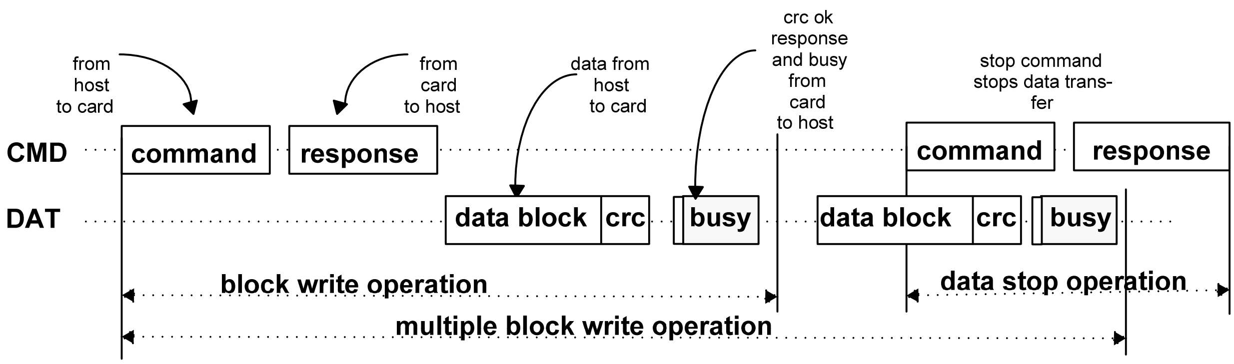


Figure 3-6: (Multiple) Block Write Operation

Command tokens have the following coding scheme:

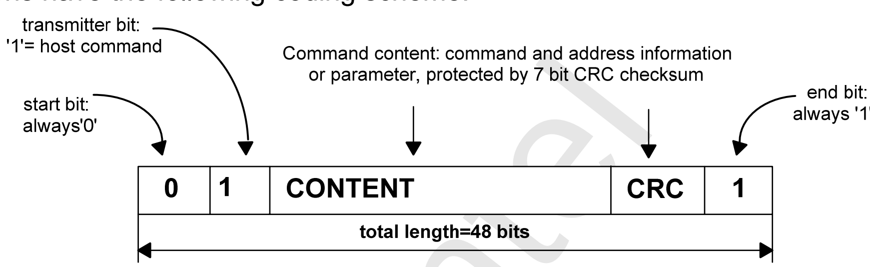


Figure 3-7: Command Token Format

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

Response tokens have one of four coding schemes, depending on their content. The token length is either 48 or 136 bits. The detailed commands and response definition is given in Chapter 4.7. The CRC protection algorithm for block data is a 16-bit CCITT polynomial. All allowed CRC types are described in Chapter 4.5.

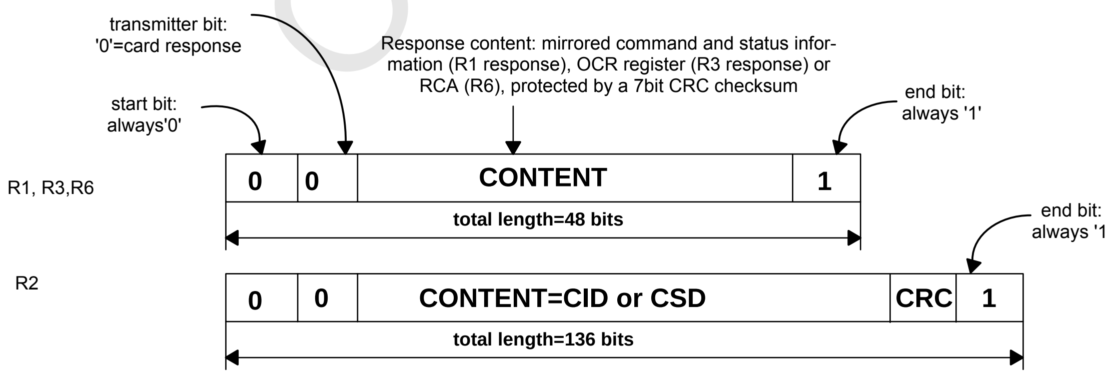


Figure 3-8: Response Token Format

In the CMD line the Most Significant Bit (MSB) is transmitted first, the Least Significant Bit (LSB) is the last.

When the wide bus option is used, the data is transferred 4 bits at a time (refer to Figure 3-10). Start and end bits, as well as the CRC bits, are transmitted for every one of the DAT lines. CRC bits are calculated and checked for every DAT line individually. The CRC status response and Busy indication will be sent by the card to the host on DAT0 only (DAT1-DAT3 during that period are don't care).

There are two types of Data packet format for the SD card.

(1) Usual data (8-bit width): The usual data (8-bit width) are sent in LSB (Least Significant Byte) first, MSB (Most Significant Byte) last sequence. But in the individual byte, it is MSB (Most Significant Bit) first, LSB (Least Significant Bit) last.

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

1. Data Packet Format for Usual Data (8-bit width)

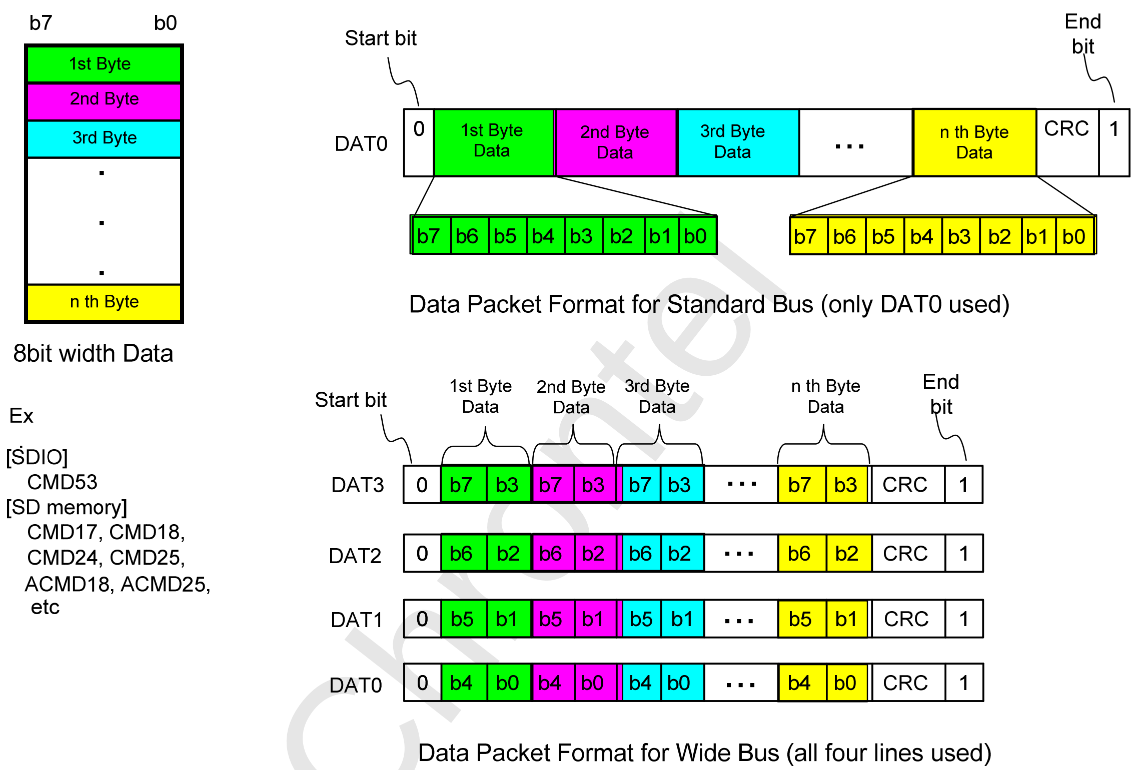


Figure 3-9: Data Packet Format - Usual Data

2. Data Packet Format for Wide Width Data (Ex. ACMD13)

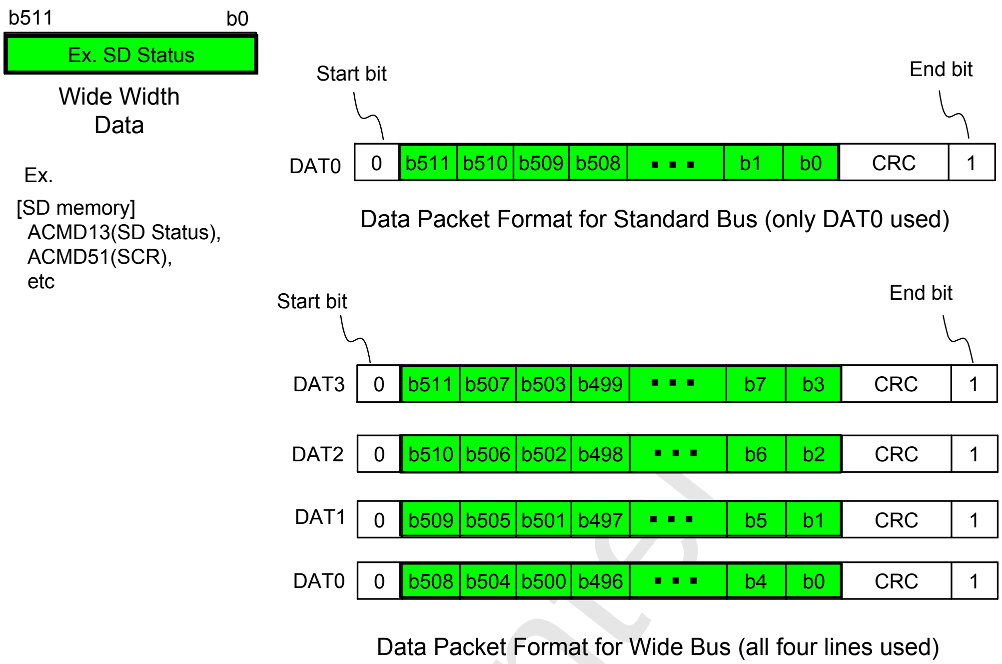


Figure 3-10: Data Packet Format - Wide Width Data

3.6.2 SPI Bus

Details of the SPI Bus protocol are described in Chapter 7.

3.7 SD Memory Card–Pins and Registers

The SD Memory Card has the form factor 24 mm x 32 mm x 2.1 mm or 24 mm x 32 mm x 1.4 mm.

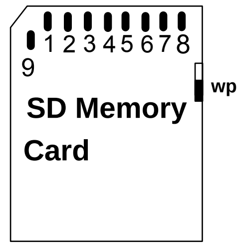


Figure 3-11: SD Memory Card Shape and Interface (Top View)

Figure 3-11 shows the general shape of the shape and interface contacts of the SD Memory Card. The detailed physical dimensions and mechanical description are given in chapter 8.
Table 3-1 defines the card contacts:

Pin #	SD Mode			SPI Mode		
	Name	Type[1]	Description	Name	Type[1]	Description
1	CD/DAT3[2]	I/O/PP[3]	Card Detect/ Data Line [Bit 3]	CS	I[3]	Chip Select (neg true)
2	CMD	PP	Command/Response	DI	I	Data In
3	V_{SS1}	S	Supply voltage ground	VSS	S	Supply voltage ground
4	V_{DD}	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS2}	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT0	I/O/PP	Data Line [Bit 0]	DO	O/PP	Data Out
8	DAT1[4]	I/O/PP	Data Line [Bit 1]	RSV		
9	DAT2[5]	I/O/PP	Data Line [Bit 2]	RSV		

1) S: power supply; I: input; O: output using push-pull drivers; PP: I/O using push-pull drivers;
2) The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after SET_BUS_WIDTH command. The Host shall keep its own DAT1-DAT3 lines in input mode, as well, while they are not used.
3) At power up this line has a 50KOhm pull up enabled in the card. This resistor serves two functions Card detection and Mode Selection. For Mode Selection, the host can drive the line high or let it be pulled high to select SD mode. If the host wants to select SPI mode it should drive the line low. For Card detection, the host detects that the line is pulled high. This pull-up should be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command
4) DAT1 line may be used as Interrupt Output (from the Card) in SDIO mode during all the times that it is not in use for data transfer operations (refer to "SDIO Card Specification" for further details).
5) DAT2 line may be used as Read Wait signal in SDIO mode (refer to "SDIO Card Specification" for further details).

Table 3-1: SD Memory Card Pad Assignment

Each card has a set of information registers (see also Chapter 5 in the SD Memory Card Physical Layer Specification):

Name	Width	Description
CID	128	Card identification number; card individual number for identification (See 5.2). **Mandatory**.
RCA[1]	16	Relative card address; local system address of a card, dynamically suggested by the card and approved by the host during initialization (See 5.4). **Mandatory**.
DSR	16	Driver Stage Register; to configure the card's output drivers (See 5.5). **Optional.**
CSD	128	Card Specific Data; information about the card operation conditions (See 5.3). **Mandatory**
SCR	64	SD Configuration Register; information about the SD Memory Card's Special Features capabilities (See 5.6). **Mandatory**
OCR	32	Operation conditions register (See 5.1). **Mandatory.**
SSR	512	SD Status; information about the card proprietary features (See 4.10.2). **Mandatory**
CSR	32	Card Status; information about the card status (See 4.10.1). **Mandatory**

(1) RCA register is not used (available) in SPI mode

Table 3-2: SD Memory Card Registers

The host may reset the cards by switching the power supply off and on again. Each card shall have its own power-on detection circuitry that puts the card into a defined state after the power-on. No explicit reset signal is necessary. The cards can also be reset by sending the GO_IDLE (CMD0) command.

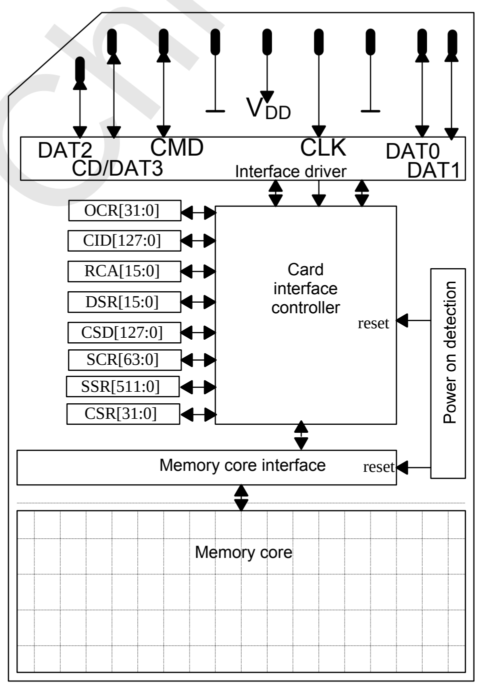


Figure 3-12: SD Memory Card Architecture

4. SD Memory Card Functional Description

4.1 General

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

- **Broadcast commands**
  Broadcast commands are intended for all cards. Some of these commands require a response.
- **Addressed (point-to-point) commands**
  The addressed commands are sent to the addressed card and cause a response from this card.
  A general overview of the command flow is shown in Figure 4-1 for card identification mode and in Figure 4-3 for data transfer mode. The commands are listed in the command tables (Table 4-19-Table 4-28). The dependencies between current state, received command, and following state are listed in Table 4-29. In the following sections, the various card operation modes will be described first. Afterwards, the restrictions for controlling the clock signal are defined. All SD Memory Card commands, along with the corresponding responses, state transitions, error conditions and timings are presented in the succeeding sections.
  Two operation modes are defined for the SD Memory Card system (host and cards):
- **Card identification mode**
  The host will be in card identification mode after reset and while it is looking for new cards on the bus. Cards will be in this mode after reset until the SEND_RCA command (CMD3) is received.
- **Data transfer mode**
  Cards will enter data transfer mode after their RCA is first published. The host will enter data transfer mode after identifying all the cards on the bus.

The following table shows the dependencies between operation modes and card states. Each state in the SD Memory Card state diagram (see Figure 4-1) is associated with one operation mode:

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

Table 4-1: Overview of Card States vs. Operation Modes

4.2 Card Identification Mode

While in card identification mode the host resets all the cards that are in card identification mode, validates operation voltage range, identifies cards and asks them to publish Relative Card Address (RCA). This operation is done to each card separately on its own CMD line. All data communication in the Card Identification Mode uses the command line (CMD) only.
During the card identification process, the card shall operate in the SD clock frequency of the identification clock rate f_{OD} (see Chapter 6.7).

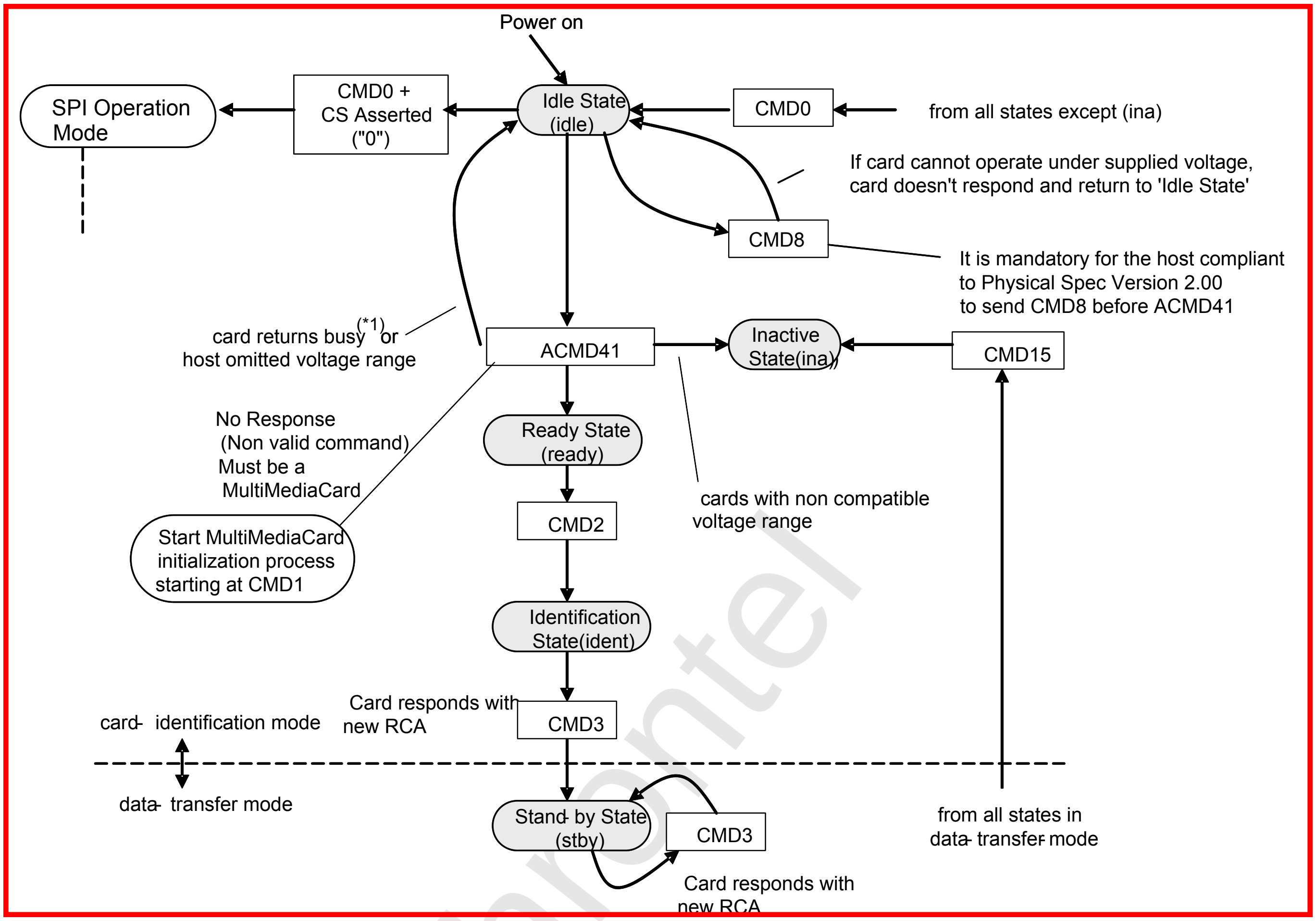
4.2.1 Card Reset

The command GO_IDLE_STATE (CMD0) is the software reset command and sets each card into *Idle State* regardless of the current card state. Cards in *Inactive State* are not affected by this command.
After power-on by the host, all cards are in *Idle State*, including the cards that have been in *Inactive State* before.
After power-on or CMD0, all cards' CMD lines are in input mode, waiting for start bit of the next command. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

4.2.2 Operating Condition Validation

At the start of communication between the host and the card, the host may not know the card supported voltage and the card may not know whether it supports the current supplied voltage. The host issues a reset command (CMD0) with a specified voltage while assuming it may be supported by the card. To verify the voltage, a following new command (CMD8) is defined in the Physical Layer Specification Version 2.00.
SEND_IF_COND (CMD8) is used to verify SD Memory Card interface operating condition. The card checks the validity of operating condition by analyzing the argument of CMD8 and the host checks the validity by analyzing the response of CMD8 (See Chapter 4.3.13). The supplied voltage is indicated by VHS filed in the argument. The card assumes the voltage specified in VHS as the current supplied voltage. Only 1-bit of VHS shall be set to 1 at any given time. Both CRC and check pattern are used for the host to check validity of communication between the host and the card.
If the card can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument.
If the card cannot operate on the supplied voltage, it returns no response and stays in idle state. It is mandatory to issue CMD8 prior to first ACMD41 for initialization of High Capacity SD Memory Card (See Figure 4-1). Receipt of CMD8 makes the cards realize that the host supports the Physical Layer Version 2.00 and the card can enable new functions.
It is also mandatory for low-voltage host to send CMD8 before ACMD41. In case that a Dual Voltage Card is not receiving CMD8 the card will work as a high-voltage only card, and in this case that a low-voltage host didn't send CMD8 the card will go to inactive at ACMD41.

SD_SEND_OP_COND (ACMD41) is designed to provide SD Memory Card hosts with a mechanism to identify and reject cards which do not match the V_{DD} range desired by the host. This is accomplished by the host sending the required V_{DD} voltage window as the operand of this command (See Chapter 5.1). Cards which cannot perform data transfer in the specified range shall discard themselves from further bus operations and go into *Inactive State*. The levels in the OCR register shall be defined accordingly (See Chapter 5.1). Note that ACMD41 is application specific command, therefore APP_CMD (CMD55) shall always precede ACMD41. The RCA to be used for CMD55 in *idle_state* shall be the card's default RCA = 0x0000.
After the host issues a reset command (CMD0) to reset the card, the host shall issue CMD8 prior to ACMD41 to re-initialize the SD Memory card.



(*1) Note : Card returns busy when
- Card executes internal initialization process
- Card is High capacity SD Memory Card and host doesn't support High capacity

This means that CMD8 is mandatory to initialize High capacity SD Memory Card.

Figure 4-1: SD Memory Card State Diagram (card identification mode)

By setting the OCR to zero in the argument of ACMD41, the host can query each card and determine the common voltage range before sending out-of-range cards into the *Inactive State* (query mode). This query should be used if the host is able to select a common voltage range or if a notification to the application of non usable cards in the stack is desired. The card does not start initialization if ACMD41 is issued as a query. Afterwards, the host may choose a voltage for operation and reissue ACMD41 with this condition, sending incompatible cards into the *Inactive State*.
During the initialization procedure, the host is not allowed to change the operating voltage range. Refer to the power up sequence as described in Chapter 6.4.

ACMD41
初始化卡

4.2.3 Card Initialization and Identification Process

After the bus is activated the host starts card initialization and identification process (See Figure 4-2). The initialization process starts with SD_SEND_OP_COND (ACMD41) by setting its operational conditions and the HCS bit in the OCR. The HCS (Host Capacity Support) bit set to 1 indicates that the host supports High Capacity SD Memory card. The HCS (Host Capacity Support) bit set to 0 indicates that the host does not support High Capacity SD Memory card.

Receiving of CMD8 expands the ACMD41 function; HCS in the argument and CCS (Card Capacity Status) in the response. HCS is ignored by cards, which didn't respond to CMD8. However the host should set HCS to 0 if the card returns no response to CMD8. Standard Capacity SD Memory Card ignores HCS. If HCS is set to 0, High Capacity SD Memory Card never return ready statue (keep busy bit to 0). The busy bit in the OCR is used by the card to inform the host that initialization of ACMD41 is completed. Setting the busy bit to 0 indicates that the card is still initializing. Setting the busy bit to 1 indicates completion of initialization. The host repeatedly issues ACMD41 until the busy bit is set to 1.

The card checks the operational conditions and the HCS bit in the OCR only at the first ACMD41. While repeating ACMD41, the host shall not issue another command except CMD0.

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

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

The host performs the same initialization sequence to all of the new cards in the system. Incompatible cards are sent into *Inactive State*. The host then issues the command ALL_SEND_CID (CMD2), to each card to get its unique card identification (CID) number. Card that is unidentified (i.e. which is in *Ready State*) sends its CID number as the response (on the CMD line). After the CID was sent by the card it goes into *Identification State*. Thereafter, the host issues CMD3 (SEND_RELATIVE_ADDR) asks the card to publish a new relative card address (RCA), which is shorter than CID and which is used to address the card in the future data transfer mode. Once the RCA is received the card state changes to the *Stand-by State.* At this point, if the host wants to assign another RCA number, it can ask the card to publish a new number by sending another CMD3 command to the card. The last published RCA is the actual RCA number of the card.

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

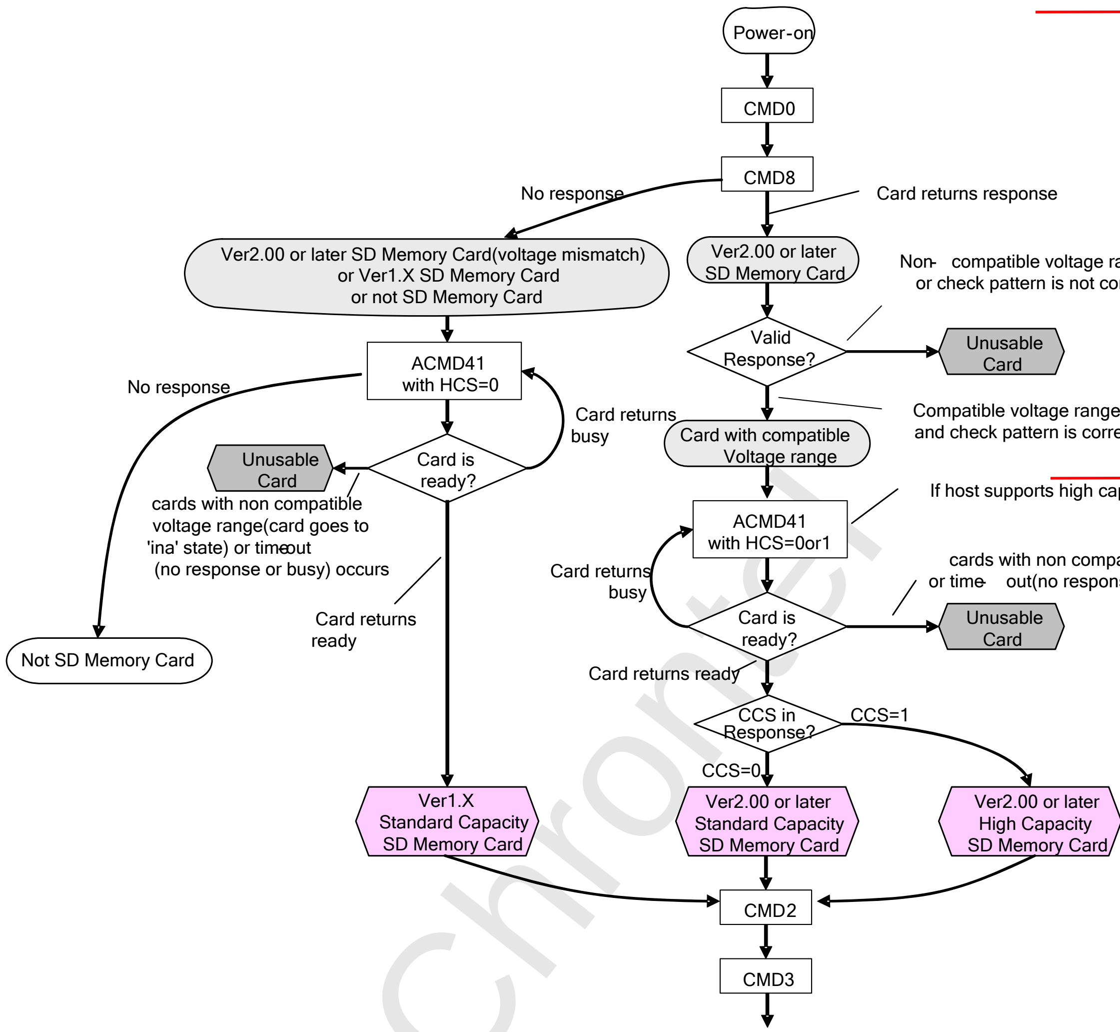


Figure 4-2: Card Initialization and Identification Flow (SD mode)

4.3 Data Transfer Mode

Until the end of Card Identification Mode the host shall remain at f_{OD} frequency because some cards may have operating frequency restrictions during the card identification mode. In Data Transfer Mode the host may operate the card in f_{PP} frequency range (see Chapter 6.7). The host issues SEND_CSD (CMD9) to obtain the Card Specific Data (CSD register), e.g. block length, card storage capacity, etc.
The broadcast command SET_DSR (CMD4) configures the driver stages of all identified cards. It programs their DSR registers corresponding to the application bus layout (length) and the number of cards on the bus and the data transfer frequency. The clock rate is also switched from f_{OD} to f_{PP} at that point. SET_DSR command is an option for the card and the host.
CMD7 is used to select one card and put it into the *Transfer State*. Only one card can be in the *Transfer State* at a given time. If a previously selected card is in the *Transfer State* its connection with the host is released and it will move back to the *Stand-by State*. When CMD7 is issued with the reserved relative card address "0x0000", all cards are put back to *Stand-by State* (Note that it is the responsibility of the Host to reserve the RCA=0 for card de-selection - refer to Table 4-19, CMD7.

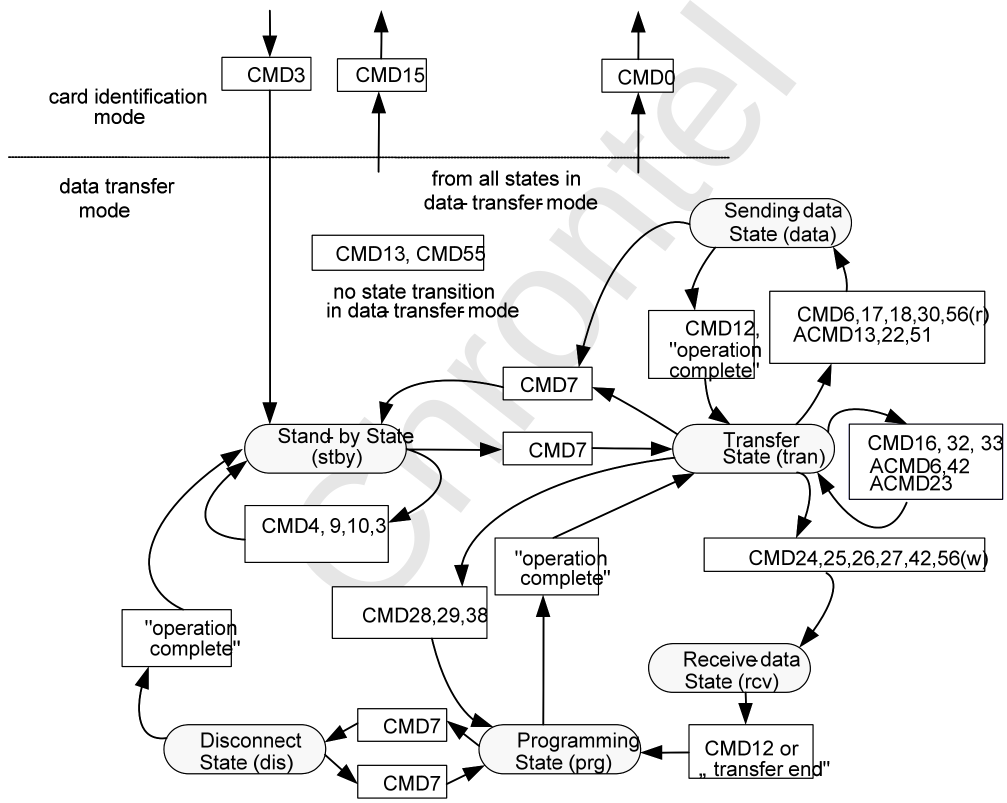


Figure 4-3: SD Memory Card State Diagram (data transfer mode)

This may be used before identifying new cards without resetting other already registered cards. Cards which already have an RCA do not respond to identification commands (ACMD41, CMD2, see Chapter 4.2.3) in this state.
Important Note: The card de-selection is done if certain card gets CMD7 with un-matched RCA. That happens automatically if selection is done to another card and the CMD lines are common. So, in SD Memory Card system it will be the responsibility of the host either to work with common CMD line (after initialization is done) - in that case the card de-selection will be done automatically or if the CMD lines are separate then the host shall be aware to the necessity to de-select cards.
All data communication in the Data Transfer Mode is point-to point between the host and the selected card (using addressed commands). All addressed commands get acknowledged by a response on the

CMD line.

The relationship between the various data transfer modes is summarized below.

- All data read commands can be aborted any time by the stop command (CMD12). The data transfer will terminate and the card will return to the *Transfer State.* The read commands are: block read (CMD17), multiple block read (CMD18), send write protect (CMD30), send scr (ACMD51) and general command in read mode (CMD56).
- All data write commands can be aborted any time by the stop command (CMD12). The write commands shall be stopped prior to deselecting the card by CMD7. The write commands are: block write (CMD24 and CMD25), program CSD (CMD27), lock/unlock command (CMD42) and general command in write mode (CMD56).
- As soon as the data transfer is completed, the card will exit the data write state and move either to the *Programming State* (transfer is successful) or *Transfer State* (transfer failed).
- If a block write operation is stopped and the block length and CRC of the last block are valid, the data will be programmed.
- The card may provide buffering for block write. This means that the next block can be sent to the card while the previous is being programmed.
  If all write buffers are full, and as long as the card is in *Programming State* (see SD Memory Card state diagram Figure 4-3), the DAT0 line will be kept low (BUSY).
- There is no buffering option for write CSD, write protection and erase. This means that while the card is busy servicing any one of these commands, no other data transfer commands will be accepted. DAT0 line will be kept low as long as the card is busy and in the *Programming State*. Actually if the CMD and DAT0 lines of the cards are kept separated and the host keep the busy DAT0 line disconnected from the other DAT0 lines (of the other cards) the host may access the other cards while the card is in busy.
- Parameter set commands are *not* allowed while card is programming.
  Parameter set commands are: set block length (CMD16), erase block start (CMD32) and erase block end (CMD33).
- Read commands are *not* allowed while card is programming.
- Moving another card from *Stand-by* to *Transfer State* (using CMD7) will not terminate erase and programming operations. The card will switch to the *Disconnect State* and will release the DAT line.
- A card can be reselected while in the *Disconnect State*, using CMD7. In this case the card will move to the *Programming State* and reactivate the busy indication.
- Resetting a card (using CMD0 or CMD15) will terminate any pending or active programming operation. This may destroy the data contents on the card. It is the host's responsibility to prevent this.
- CMD34-37, CMD50 and CMD57 are reserved for SD command system expansion. State transitions for these commands are defined in each command system specification.

4.3.1 Wide Bus Selection/Deselection

使用ACMD6改变数据线的宽度，默认情况下为1位数据宽度

Wide Bus (4 bit bus width) operation mode may be selected/deselected using ACMD6. The default bus width after power up or GO_IDLE (CMD0) is 1 bit bus width.
In order to change the bus width two conditions shall be met:
a) The card is in '*tran state*'.
b) The card is not locked

A locked card will responds to ACMD6 as illegal command.

4.3.2 2 GByte Card

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

4.3.3 Data Read

The DAT bus line level is high by the pull-up when no data is transmitted. A transmitted data block consists of start bits (1 or 4 bits LOW), followed by a continuous data stream. The data stream contains the payload data (and error correction bits if an off-card ECC is used). The data stream ends with end bits (1 or 4 bits HIGH) (see Figure 4-18-Figure 4-20). The data transmission is synchronous to the clock signal. The payload for block oriented data transfer is protected by 1 or 4 bits CRC check sum (See Chapter 3.6).
The Read operation from SD Memory Card may be interrupted by turning the power off. The SD Memory Card ensures that data is not destroyed during all the conditions except write or erase operations issued by the host even in the event of sudden shut down or removal.
Read command is rejected if BLOCK_LEN_ERROR or ADDRESS_ERROR occurred and no data transfer is performed.

- **Block Read**

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

Table 4-2 defines the card behavior when a partial block accesses is enabled.
If the misaligned block is the first data block of the command (i.e. ADDRESS_ERROR was reported in the actual response to the command), then no data is transferred and the card remains in the TRAN state.

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

*1: "Current Blocklen" size is set or changed by CMD16. If value is less than or equal 512 bytes (There are no relations with Misalign and Partial option), it is set with no error.
*2: When the Blocklen size data range crosses 512 bytes block boundary, card outputs the data until the 512 bytes block boundary" and then the data becomes invalid and CRC error also may occur. The card will send "ADDRESS_ERROR" on the next command response. Host should issue CMD12 to recover.

Table 4-2: Read Command Blocklen

4.3.4 Data Write

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 1 or 4 bits CRC parity check (See Chapter 4.5) for each received data block prior to the write operation. By this mechanism, writing of erroneously transferred data can be prevented.
Write command is rejected if BLOCK_LEN_ERROR or ADDRESS_ERROR occurred and no data transfer is performed.

- **Block Write**

During block write (CMD24 - 27, 42, 56(w)) one or more blocks of data are transferred from the host to the card with 1 or 4 bits CRC appended to the end of each block by the host. A card supporting block write shall be required that Block Length set by CMD16 shall be 512 bytes regardless of WRITE_BL_LEN is set to 1k or 2k bytes.
Table 4-3 defines the card behavior when partial block accesses is disabled (WRITE_BL_PARTIAL = 0).

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

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

Table 4-3: Write Command Blocklen

If WRITE_BL_PARTIAL is allowed (=1) then smaller blocks, up to resolution of one byte, can be used as well. If the CRC fails, the card shall indicate the failure on the DAT line (see below); the transferred data will be discarded and not be written, and all further transmitted blocks (in multiple block write mode) will

be ignored.
Multiple block write command shall be used rather than continuous single write command to make faster write operation.
If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card shall detect the block misalignment error and abort programming before the beginning of the first misaligned block. The card shall set the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, wait in the *Receive-data-State* for a stop command.
Note that the first data block is misaligned for write command (i.e. ADDRESS_ERROR is reported in the actual response of the write command), the card remains in tran state and no data is programmed.
The write operation shall also be aborted if the host tries to write over a write protected area. In this case, however, the card shall set the WP_VIOLATION bit.
Programming of the CSD register does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD register is stored in ROM, then this unchangeable part shall match the corresponding part of the receive buffer. If this match fails, then the card will report an error and not change any register contents.
Some cards may require long and unpredictable times to write a block of data. After receiving a block of data and completing the CRC check, the card will begin writing and hold the DAT0 line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) at any time, and the card will respond with its status. The status bit READY_FOR_DATA indicates whether the card can accept new data or whether the write process is still in progress). The host may deselect the card by issuing CMD7 (to select a different card) which will displace the card into the *Disconnect State* and release the DAT line without interrupting the write operation. When reselecting the card, it will reactivate busy indication by pulling DAT to low if programming is still in progress and the write buffer is unavailable. Actually, the host may perform simultaneous write operation to several cards with inter-leaving process. The interleaving process can be done by accessing each card separately while other cards are in busy. This process can be done by proper CMD and DAT0-3 line manipulations (disconnection of busy cards).

- **Pre-erased Setting prior to a Multiple Block Write Operation**

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

- **Send Number of Written Blocks**

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

4.3.5 Erase

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

4.3.6 Write Protect Management

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

- **Mechanical Write Protect Switch**

A mechanical sliding tablet on the side of the card (refer to the mechanical description Chapter 8) will be used by the user to indicate that a given card is write protected or not. If the sliding tablet is positioned in such a way that the window is open it means that the card is write protected. If the window is close the card is not write-protected.

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

- **Card's Internal Write Protection (Optional)**

Card data may be protected against either erase or write. The entire card may be permanently write-protected by the manufacturer or content provider by setting the permanent or temporary write protect bits in the CSD. For cards which support write protection of groups of sectors by setting the WP_GRP_ENABLE bit in the CSD, portions of the data may be protected (in units of WP_GRP_SIZE sectors as specified in the CSD), and the write protection may be changed by the application. The SET_WRITE_PROT command sets the write protection of the addressed write-protect group, and the CLR_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 shall send a data block containing 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits. The address field in the write protect commands is a group address in byte units. The card will ignore all LSB's below the group size.
The Password Card Lock protection is described in the following section.
Note that the High Capacity SD Memory Card does not support Write Protection and does not respond to write-protection commands (CMD28, CMD29 and CMD30).

4.3.7 Card Lock/Unlock Operation

4.3.7.1 General

The password protection feature enables the host to lock a card while providing a password, which later will be used for unlocking the card. The password and its size are kept in a 128-bit PWD and 8-bit PWD_LEN registers, respectively. These registers are non-volatile so that a power cycle will not erase them.

Locked cards respond to (and execute) all commands in the "basic" command class (class 0), ACMD41, CMD16 and "lock card" command class. Thus, the host is allowed to reset, initialize, select, query for status, etc., but not to access data on the card. If the password was previously set (the value of PWD_LEN is not 0), the card will be locked automatically after power on.

Similar to the existing CSD register write commands, the lock/unlock command is available in "transfer state" only. This means that it does not include an address argument and the card shall be selected before using it.

The card lock/unlock command has the structure and bus transaction type of a regular single block write command. The transferred data block includes all the required information of the command (password setting mode, PWD itself, card lock/unlock etc.). Table 4-4 describes the structure of the command data block. Note that the host compliant to the SD Physical Specification Version 2.00 shall set reserved bits (Bit7-4) to 0 when issuing CMD42.

Byte #	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Reserved (shall be set to 0)				ERASE	LOCK_UNLOCK	CLR_PWD	SET_PWD
1	PWDS_LEN							
2	Password data							
...								
PWDS_LEN + 1								

Table 4-4: Lock Card Data Structure

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

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

The following paragraphs define the various lock/unlock command sequences:

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

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

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

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

  If the PWD content is equal to the sent password, then the card will be locked and the card-locked status bit will be set in the status register. If the password is not correct, then the LOCK_UNLOCK_FAILED error bit will be set in the status register.
  Note that it is possible to set the password and to lock the card in the same sequence. In such a case, the host shall perform all the required steps for setting the password (as described above) including the bit LOCK set while the new password command is sent.
  If the password was previously set (PWD_LEN is not 0), then the card will be locked automatically after

power on reset.
An attempt to lock a locked card or to lock a card that does not have a password will fail and the LOCK_UNLOCK_FAILED error bit will be set in the status register, unless it was done during a password definition or change operations.

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

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

4.3.7.2 Parameter and the Result of CMD42

The block length shall be greater than or equal to the required data structure of CMD42; otherwise, the result of CMD42 is undefined and the card may be in the unexpected locked state. Table 4-5 clarifies the behavior of CMD42. The reserved bits in the parameter (bit7-4) of CMD42 shall be don't care. In the case that CMD42 requires the password, it is assumed that the old password and the new password are set correctly; otherwise the card indicates an error regardless of Table 4-5. If the password length is 0 or greater than 128 bits, the card indicates an error. If errors occur during execution of CMD42, the LOCK_UNLOCK_FAILED (Bit24 of Card Status) shall be set to 1 regardless of Table 4-5. The CARD_IS_LOCKED (Bit25 of Card Status) in the response of CMD42 shall be the same as Current Card State in Table 4-5. In the field of Card Status, 0 to 1 means the card changes to Locked and 1 to 0 means the card changes to Unlocked after execution of CMD42. It can be seen in the response of CMD13 after the CMD42. The LOCK_UNLOCK_FAILED (Bit24 of Card Status) as the result of CMD42 can be seen in the response of either CMD42 or the following CMD13.

CMD42 Parameter in the data
Bit3: ERASE
Bit2: LOCK_UNLOCK
Bit1: CLR_PWD
Bit0: SET_PWD

Related bits in the Card Status
Bit25: CARD_IS_LOCKED
Bit24: LOCK_UNLOCK_FAILED

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

Table 4-5: Lock Unlock Function (Basic Sequence for CMD42)

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

4.3.7.3 Forcing Erase

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

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

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

4.3.7.3.1 Force Erase Function to the Locked Card

Table 4-6 clarifies the relation between force erase and Write Protection. The force erase does not erase the secure area. The card shall keep its locked state during the erase execution and change to the unlocked state after the erase of all user area is completed. Similarly, the card shall keep Temporary and Group Write Protection during the erase execution and clear Write Protection after the erase of all user area is completed. In the case of an erase error occurs, the card can continue force erase if the data of error sectors are destroyed.

Write Protections
PWP: Permanent Write Protect (CSD Bit13)
TWP: Temporary Write Protect (CSD Bit12)
GWP: Group Write Protect (CMD28, CMD29, CMD30)

CMD42 Parameter				PWP	TWP	Result of the Function	Card Status	
Bit3	Bit2	Bit1	Bit0		GWP		Bit25	Bit24
1	0	0	0	Yes	don't care	Error (Note *2 Refer to Table 4-8)	1	1
1	0	0	0	No	Yes	Execute force erase and clear Temporary Write Protect and Group Write Protect. (Note *3 Refer to Table 4-8)	1 to 0	0
1	0	0	0	No	No	Execute force erase.	1 to 0	0

Table 4-6: Force Erase Function to the Locked Card (Relation to the Write Protects)

4.3.7.4 Relation Between ACMD6 and Lock/Unlock State

ACMD6 is rejected when the card is locked and bus width can be changed only when the card is unlocked. Table 4-7 shows the relation between ACMD6 and the Lock/Unlock state.

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

Table 4-7: Relation between ACMD6 and the Lock/Unlock State

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

4.3.7.5 Commands Accepted for Locked Card

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

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

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

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

4.3.7.6 Two Types of Lock/Unlock Card

There are two types of lock/unlock function-supported cards. The Type 1 is the earlier version of SD Memory Card and the Type 2 is defined in the Physical Layer Specification Version 1.10 and higher. Table 4-8 shows the difference between these types of cards. The SD memory cards that support Lock/Unlock and comply with Version 1.01, can take either Type 1 or Type 2. The SD Memory Cards that support Lock/Unlock and comply with Version 1.10 and higher, shall take Type 2.

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

Table 4-8: Version Difference of Lock/Unlock Functions

Application Note:
The host can use both types of cards without checking the difference by taking account of the following points.

(1) The host should not set the parameters of CMD42 that return an error listed in Table 4-5. (For *1)
(2) The host should not issue a force erase command if the Permanent Write Protect is set to 1, otherwise the Type 1 card can no longer be used even if the user remembers the password. (For *2)
(3) After the force erase, if the Temporary Write Protect is not cleared, the host should clear it. (For *3)

4.3.8 Content Protection

Detailed descriptions of the Content Protection mechanism and the related security SD Memory Card commands can be found in the “SD Specifications Part3Security Specification” document. All SD Memory Card security related commands shall be operated in the data transfer mode of operation.
As defined in the SDMI spec the data content that is saved in the card is saved already encrypted and it passes transparently to/from the card. NO operation is done on the data and there is no restriction on reading the data at any time. Associated with every data packet (song, for example) that is saved in the un-protected memory, there is a special data that shall be saved in a protected memory area. For any access (any Read or Write or Erase Command) from/to the data in the protected area, an authentication procedure shall be done between the card and the connected device, either the LCM (PC for example) or the PD (Portable Device - SD Player for example). After the authentication process has passed OK, the card is ready to accept or give data from/to the connected device. While the card is in the secured mode of operation (after the authentication succeeded), the argument and the associated data that is sent to the card or read from the card are encrypted. At the end of the Read/Write/Erase operation, the card leaves its secured mode automatically.

4.3.9 Application-Specific Commands

4.3.9.1 Application-Specific Command – APP_CMD (CMD55)

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

The only effect of the APP_CMD is that if the command index of the immediately following command has an ACMD overloading it, the non-regular version is used. If, as an example, a card has a definition for ACMD13 but not for ACMD7, then, if received immediately after APP_CMD command, Command 13 will be interpreted as the non-regular ACMD13 but command 7 as the regular CMD7. In order to use one of the manufacturer-specific ACMD's, the host should be:

- Send APP_CMD. The response has the APP_CMD bit (new status bit) set signaling to the host that ACMD is now expected.
- Send the required ACMD. The response has the APP_CMD bit set, indicating that the accepted command was interpreted as ACMD. If a non-ACMD is sent, then it is respected by the card as a normal SD Memory Card command and the APP_CMD bit in the Card Status stays clear.
- If multiple CMD55 are issued continuously, APP_CMD bit in each response is set to 1. The command issued immediately after the last CMD55 shall be interpreted as ACMD. When more than one command (except CMD55) is issued directly after CMD55, the first command should be interpreted as ACMD and the following commands should be interpreted as regular commands

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

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

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

4.3.9.2 General Command - GEN_CMD (CMD56)

GEN_CMD (CMD56) is a vendor-specific and optional command. The command operation is defined by this specification. The bus transaction of the GEN_CMD is the same as the single block read or write commands (CMD24 or CMD17) and accepted only in 'tran_state'. The response type is R1. The difference is that the argument denotes the direction of the data transfer (rather than the address) and the data block is not a memory payload data but has a vendor-specific format and meaning. The card shall be selected ('*tran_state*') before sending CMD56. In the case of the Standard Capacity SD Memory Card, the data block size is the BLOCK_LEN that was defined with CMD16. In the case of the High Capacity SD Memory Card, the data block size is fixed to 512byte.

The bit 0 of the argument indicates the direction of the data transfer; 0 means write operation and 1 means read operation. A vendor can define a specific format to bits 31-1 of the argument and content of data block of this command. However, it should be considered that the card shall prevent corruption of the card from receiving unexpected format.

The host should confirm CID before issuing CMD56 so that the card supports the format of CMD56.

4.3.10 Switch Function Command

4.3.10.1 General

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

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

This was introduced in the SD Physical Layer Specification Version 1.10. Therefore, cards that are compatible with earlier versions of the spec do not support it. The host shall check the "SD_SPEC" field in the SCR register to identify what version of the spec the card complies with before using CMD6. It is mandatory for an SD memory card of Version 1.10 and higher to support CMD6.

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

1. CMD6 is defined for memory card. SDIO card will use CCCR to switch functions.

As a response to CMD6, the SD Memory Card will send R1 response on the CMD line and 512 bits of status on the DAT lines. From the SD bus transaction point of view, this is a standard single block read transaction and the time out value of this command is 100 ms, the same as in read command. If CRC error occurs on the status data, the host should issue a power cycle.
CMD6 function switching period is within 8 clocks after the end bit of status data. When CMD6 changes the bus behavior (i.e. access mode), the host is allowed to use the new functions (increase/decrease CLK frequency beyond the current max CLK frequency), at least 8 clocks after at the end of the switch command transaction (see Figure 4-4).

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

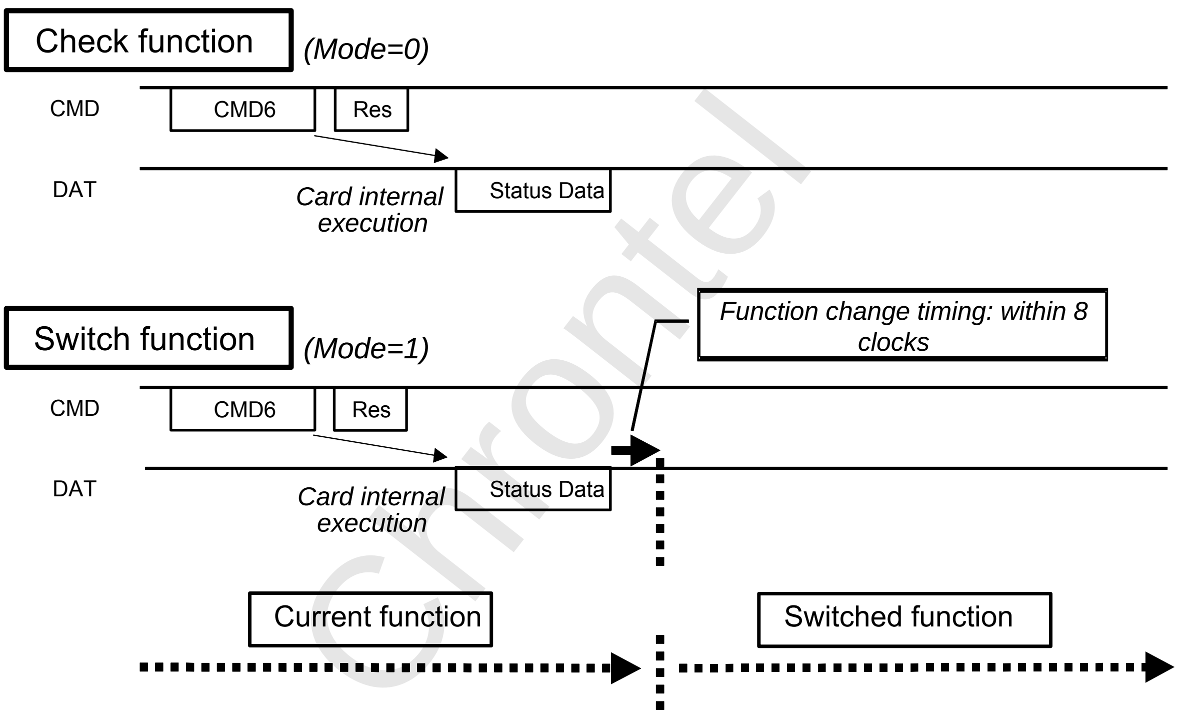


Figure 4-4: Use of Switch Command

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

CMD6 can be used in two modes:

- Mode 0 (Check function) is used to query if the card supports a specific function or functions.
- Mode 1 (set function) is used to switch the functionality of the card.

4.3.10.2 Mode 0 Operation - Check Function

CMD6 mode 0 is used to query which functions the card supports, and to identify the maximum current consumption of the card under the selected functions.
Refer to Table 4-28: Switch function commands (class 10) for the argument definition of CMD6.

A query is done by setting the argument field of the command, as follows:

- Set the Mode bit to 0
- Select only one function in each function group. Selection of default function is done by setting the function to 0x0. Select a specific function by using appropriate values from Table 4-9. Selecting 0xF will keep the current function that has been selected for the function group.
- When the function in query is ready, the card returns the inquired function number, if busy, the card returns the current function number (See Table 4-12).

In response to a query, the switch function will return the following 3 statuses (see Table 4-10):

- The functions that are supported by each of the function groups
- The function that the card will switch to in each of the function groups. This value is identical to the provided argument if the host made a valid selection or 0xF if the selected function was invalid.
- Maximum current consumption under the selected functions. If one of the selected functions was wrong, the return value will be 0.

4.3.10.3 Mode 1 Operation - Set Function

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

Switching to a new functionality is done by:

- Setting the Mode bit to 1
- Selecting only one function in each function group. Selection of default function is done by setting the function to 0x0. It is recommended to specify 0xF (no influence) for all selected functions, except for functions that need to be changed. Selecting 0xF will keep the current function for the function group.
- When a function cannot be switched because it is busy, the card returns the current function number (not returns 0xF), the other functions in the other groups may still be switched.

In response to a set function, the switch function will return the following 3 statuses:

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

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

Table 4-9: Available Functions

4.3.10.4 Switch Function Status

The switch function status is the returned data block that contains function and current consumption information. The block length is predefined to 512 bits and the use of SET_BLK_LEN command is not necessary. Table 4-10 describes the status data structure.

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

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

Bits	Description	Width
379:376	mode 0 - The function which can be switched in function group 1. mode 1 - The function which is result of the switch command, in function group 1. 0xF shows function set error with the argument.	4
375:368	Data Structure Version 00h – bits 511:376 are defined 01h – bits 511:272 are defined 02h-FFh – reserved	8
367:352	Busy Status of functions in group 6. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
351:336	Busy Status of functions in group 5. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
335:320	Busy Status of functions in group 4. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
319:304	Busy Status of functions in group 3. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
303:288	Busy Status of functions in group 2. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
287:272	Busy Status of functions in group 1. If bit [i] is set, function [i] is busy. This field can be read in mode 0 and mode 1	16
271:0	Reserved (All '0')	272

Table 4-10: Status Data Structure

4.3.10.4.1 Busy Status Indication for Functions

Each bit [367-272] shows the busy status of corresponding functions; 0 indicates ready and 1 indicates busy. While the status is busy, the host should not change the corresponding function. Switch command mode 1 can be applied only to ready functions.
If the function failed to be switched in mode 1 operation and returns the current function number in the response, the function is considered busy. The mode 1 operation may affect the behavior of a function. The mode 0 operations should be used to check the busy status of a function because it does not affect its behavior, especially, for function group 2 as defined below.

Function Group 2															
303	302	301	300	299	298	297	296	295	294	293	292	291	290	289	288
0	VS	0	0	0	0	0	0	0	0	0	0	0	0	eC	0

Busy Status for 'Vendor Specific'
0:ready
1:busy

Busy Status for ‘eC’
0:ready
1:busy

Figure 4-5: Busy Status of ‘Command System’

4.3.10.4.2 Data Structure Version

Data Structure Version indicates effective bit fields of the Switch Function Status. The cards can set either 00h or 01h. When this field is set to 01, busy status indication is effective.

Data Structure Version	Fields of Status Data Structure
00h	511:376 are defined
01h	511:272 are defined
02h-FFh	Reserved

Table 4-11: Data Structure Version

4.3.10.4.3 Function Table of Switch Command

Table 4-12, Table 4-13 and Table 4-14 shows possible combinations of the function switch.
"Argument" indicates 4-bit code specified in the argument of switch command (bits 23-0). "**Busy Status**" indicates the function is busy as defined below.
"Status Code" indicates 4-bit code in the Status Data Structure, bits 399-376.

Argument	Busy Status	Status Code	Comment
0	Don't Care	0	Status indicates a default function, which is always supported.
Supported function	Ready	=Arg.	Status indicates that the function specified in the argument is supported and can be switched.
	Busy	Current Selected	Status indicates that the function specified in the argument is supported but cannot be switched because the function is busy.
Not Supported function	Don't Care	Fh	Status indicates that the function specified in the argument is not supported.
Fh	Don't Care	Current Selected	Status indicates current selected function

Table 4-12: Status Code of Mode 0 to Supported Function Group

Argument	Busy Status	Status Code	Comment
0	Don't Care	0	Default function can always be switched.
Supported function	Ready	=Arg.	Status indicates the same function number as specified in the argument, which means successful function change.
	Busy	Current Selected	Switch function is canceled and status indicates current selected function.
Not Supported function	Don't Care	Fh	If one of the function groups indicates an error code (Fh), switch requests to all switch functions are canceled and the rest of the data in the Status Data Structure should be ignored.
Fh	Don't Care	Current Selected	Status indicates current selected function

Table 4-13: Status Code of Mode 1 to Supported Function Group

Argument	Busy Status	Status Code	Comment
0	Don't Care	0	Status always indicates 0.
Eh-1h	Don't Care	Fh	Status always indicates Fh.
Fh	Don't Care	0	Status always indicates 0.

Table 4-14: Status Code of Mode 0 and 1 to Unsupported Function Group

4.3.10.5 Relationship between CMD6 data & other commands

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

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

- **Relationship between CMD6 data &CMD12**

Case 1: Not complete case (The card does not output all data.)
In the case that the host sends the end bit of CMD12 before CRC bit 15, CMD6 is stopped by CMD12, and the card shall terminate data transfer of CMD6. The card behavior is not guaranteed and re-initialization from CMD0 is the only way to recover from an undefined state. The end bit of the host command is followed, on the data line, with one more data bit and one end bit.

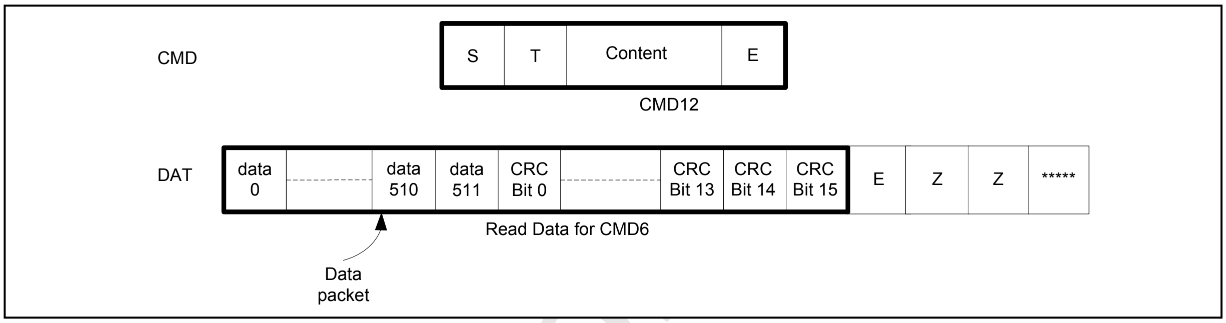


Figure 4-6: CMD12 during CMD6; Case 1

Case 2: Complete case (The card outputs all data.)
The card shall complete the CMD6 execution and its behavior is guaranteed. The complete case includes the later timing of CMD12 than Figure 4-7. The end bit of the host command is followed by the end bit on the data line.

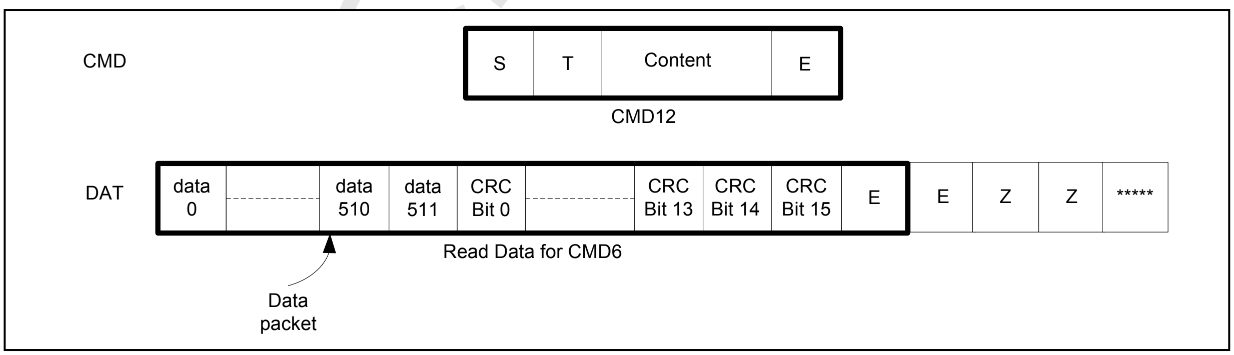


Figure 4-7: CMD12 during CMD6; Case 2

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

4.3.10.6 Switch Function Flow Example

Figure 4-8 (a) to (c) shows three possible cases of a switch function sequence. Depending on the busy status of the function, the function changes asynchronously to the sequence of CMD6. The host needs to cope with these three cases.

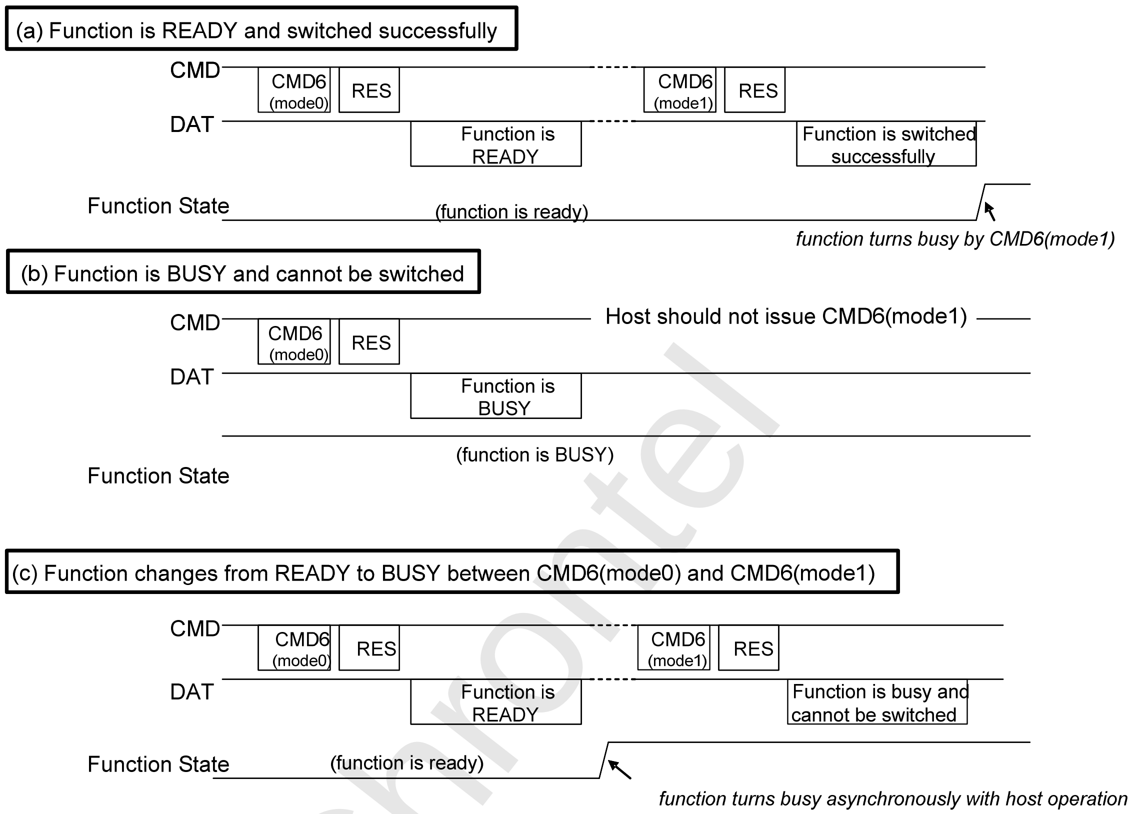


Figure 4-8: Example of Switch Function flow

Before switching functions, the host should issue CMD6 (mode0) to obtain the busy status and current consumption. If the current consumption is not acceptable, the host should find another combination of functions that meets the host's current limitation. If the busy status is read, the host can issue CMD6 (mode1) to switch function as described in Figure 4-8 (a) and (c). If a function busy status is busy, host should not issue CMD6 (mode1) as described in Figure 4-8(b). Figure 4-8 (c) shows the case that CMD6 (mode1) is cancelled due to the busy status change after the host receives ready status of the function at mode 0 operation. Figure 4-9 shows the sequence of the switching function.

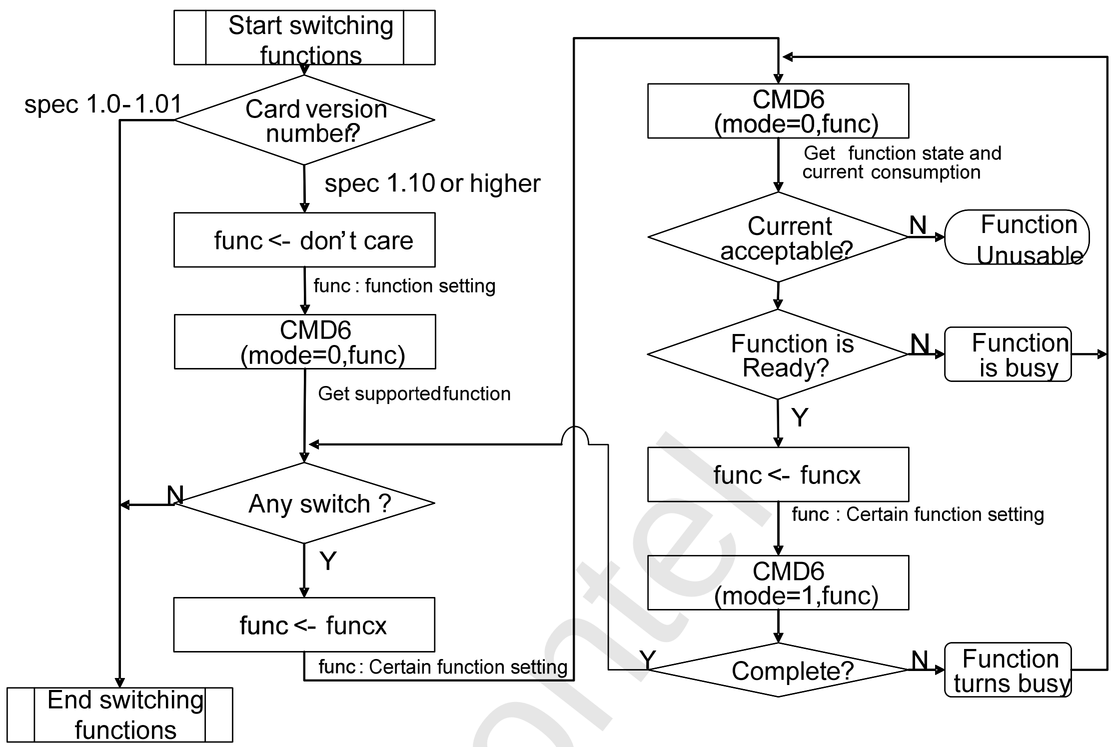


Figure 4-9: Switching Function Flow

Application Note:
It is recommended that the host should follow the illustrated sequence to switch the function.

4.3.10.7 Example of Checking

Card Condition

Support function = command system: For eC (0x1),   access mode: High-Speed (0x1)
Current function = command system: For eC (0x1),   access mode: Default (0x0)
Switch example: command system: For eC => Default,   access mode: Default => High-Speed

Case (1) - Check function with no error

CMD6 argument = ‘0000 0000 1111 1111 1111 1111 **0000 0001**’
Read Data = [511:496] = ‘0000 0000 0100 0000’ (=64mA)
[495:400] = ‘1000 0000 0000 0001’ & ‘1000 0000 0000 0001’ & ‘1000 0000 0000 0001’ &
‘1000 0000 0000 0001’ & ‘1000 0000 0000 00**1**1’ & ‘**1**000 0000 0000 00**11**’
[399:376] = ‘0000 0000 0000 0000 0000 0001’

[375:368] = Data Structure Version
[367:272] = Busy Status of functions in group 1-6
[271:0] = Reserved (All ‘0’)

For eC
No influence
High-Speed
Default

Case (2) - Check function with error

CMD6 argument = ‘0000 0000 1111 **1000** 1111 **0010** 0000 0001’
Read Data = [511:496] = ‘0000 0000 0000 0000’ (means error)
[495:400] = ‘1000 0000 0000 0001’ & ‘1000 0000 0000 0001’ & ‘1000 0000 0000 0001’ &
‘1000 0000 0000 0001’ & ‘1000 0000 0000 0011’ & ‘1000 0000 0000 0011’
[399:376] = ‘0000 **1111** 0000 **1111** 0000 0001’

[375:368] = Data Structure Version
[367:272] = Busy Status of functions in group 1-6
[271:0] = Reserved (All ‘0’)

Error

4.3.10.8 Example of Switching

Card Condition

Support function = command system: For eC (0x1), access mode: High-Speed (0x1)
Current function = command system: For eC (0x1), access mode: Default (0x0)
Switch example: command system: For eC => Default, access mode: Default => High-Speed

Case (3) - Switch function with no error

CMD6 argument = '1000 0000 1111 1111 1111 1111 **0000 0001**'
Read Data = [511:496] = '0000 0000 0100 0000' (=64mA)
[495:400] = '1000 0000 0000 0001' & '1000 0000 0000 0001' & '1000 0000 0000 0001' &
'1000 0000 0000 0001' & '1000 0000 0000 00**1**1' & '**1**000 0000 0000 00**11**'
[399:376] = '0000 0000 0000 0000 0000 0001'

For eC
High-Speed
No influence
Default

[375:368] = Data Structure Version
[367:272] = Busy Status of functions in group 1-6
[271:0] = Reserved (All '0')

Case (4) - Switch function with error

CMD6 argument = '1000 0000 1111 **1000** 1111 **0010** 0000 0001'
Read Data = [511:496] = **'0000 0000 0000 0000'** (means error)
[495:400] = '1000 0000 0000 0001' & '1000 0000 0000 0001' & '1000 0000 0000 0001' &
'1000 0000 0000 0001' & '1000 0000 0000 0011' & '1000 0000 0000 0011'
[399:376] = '0000 **1111** 0000 **1111** 0001 0000'

Error

[375:368] = Data Structure Version
[367:272] = Busy Status of functions in group 1-6
[271:0] = Reserved (All '0')

4.3.11 High-Speed Mode (25 MB/sec interface speed)

Although the Rev 1.01 SD memory card supports up to 12.5 MB/sec interface speed, the speed of 25 MB/sec is necessary to support increasing performance needs of the host and because memory size continues to grow.

To achieve the 25 MB/sec interface speed, the clock rate is increased to 50 MHz and CLK/CMD/DAT signal timing and circuit conditions are reconsidered and changed from the Physical Layer Specification Version 1.01.

After power up, the SD memory card is in the default speed mode, and by using Switch Function command (CMD6), the Version 1.10 and higher SD memory card can be placed in High-Speed mode. The High-Speed function is a function in the access mode group (see Table 4-9). Supporting High-Speed mode is optional.

Because it is not possible to control two cards or more in the case that each of them has a different timing mode (Default and High-Speed mode) and in order to satisfy severe timing, the host shall drive only one card. CLK/CMD/DAT signal shall be connected in 1-to-1 between the host and the card.

4.3.12 Command System

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

- When the "standard command set" (default function 0x0) is selected, these commands will not be recognized by the card and will be considered as illegal commands (as defined in Version 1.01 of the SD Physical Layer Specification)
- When the “vendor specific” (function 0xE) is selected, the behaviors of these commands are vendor specific. They are not defined by this standard and may change for different card vendors.
- When the “mobile e-commerce” (function 0x1) is selected, the behavior of these commands is governed by the SD Specifications Part A1: Mobile Commerce Extension Specification.

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

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

4.3.13 Send Interface Condition Command (CMD8)

CMD8 (Send Interface Condition Command) is defined to initialize SD Memory Cards compliant to the Physical Specification Version 2.00. CMD8 is valid when the card is in Idle state. This command has two functions.

- Voltage check:
  Checks whether the card can operate on the host supply voltage.
- Enabling expansion of existing command and response:
  Reviving CMD8 enables to expand new functionality to some existing commands by redefining previously reserved bits. ACMD41 is expanded to support initialization of High Capacity SD Memory Cards.

Table 4-15 shows the format of CMD8.

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

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

Table 4-15: Format of CMD8

When the card is in Idle state, the host shall issue CMD8 before ACMD41. In the argument, ‘voltage supplied’ is set to the host supply voltage and ‘check pattern’ is set to any 8-bit pattern.
The card checks whether it can operate on the host’s supply voltage. The card that accepted the supplied voltage returns R7 response. In the response, the card echoes back both the voltage range and check pattern set in the argument. If the card does not support the host supply voltage, it shall not return response and stays in Idle state. Table 4-16 shows the card operation for CMD8.

Application Note: It is recommended to use ‘10101010b’ for the ‘check pattern’.

Command Argument Check					Response of Card[1]					
Index	Reserved	VHS	Pattern	CRC	Index	Ver	Reserved	VCA	Pattern	CRC
Don't Care	Don't Care	Don't Care	Don't Care	Error	No Response (CRC Error Indication in the following command)					
Not 8	Don't Care	Don't Care	Don't Care	Correct	Depends on command index					
=8	Don't Care	Mismatch[2]	Don't Care	Correct	No Response					
=8	Don't Care	Match[2]	Don't Care	Correct	8	Ver=0	0	Echo Back	Echo Back	Calculate

1): Response indicates the actual response the card returns. (It does not include errors during response transfer.)
2): Match means AND of the following conditions a) and b). Mismatch is other cases.
a) Only one bit is set to 1 in VHS.
b) The card supports the host supply voltage.

Table 4-16: Card Operation for CMD8 in SD Mode

4.3.14 Command Functional Difference in High Capacity SD Memory Card

Memory access commands include block read commands (CMD17, CMD18), block write commands (CMD24, CMD25), and block erase commands (CMD32, CMD33).
Following are the functional differences of memory access commands between Standard Capacity and High Capacity SD Memory Cards:

- Command Argument
  In High Capacity Cards, the 32-bit argument of memory access commands uses the memory address in block address format. Block length is fixed to 512 bytes,
  In Standard Capacity Cards, the 32-bit argument of memory access commands uses the memory address in byte address format. Block length is determined by CMD16,
  i.e.:
  (a) Argument 0001h is byte address 0001h in the Standard Capacity Card and 0001h block in High Capacity Card
  (b) Argument 0200h is byte address 0200h in the Standard Capacity Card and 0200h block in High Capacity Card
- Partial Access and Misalign Access
  Partial access and Misalign access (crossing physical block boundary) are disabled in High Capacity Card as the block address is used. Access is only granted based on block addressing.
- Set Block Length
  When memory read and write commands are used in block address mode, 512-byte fixed block length is used regardless of the block length set by CMD16. The setting of the block length does not affect the memory access commands. CMD42 is not classified as a memory access command. The data block size shall be specified by CMD16 and the block length can be set up to 512 bytes. Setting block length larger than 512 bytes sets the BLOCK_LEN_ERROR error bit regardless of the card capacity.
- Write Protected Group
  High Capacity SD Memory Card does not support write-protected groups. Issuing CMD28, CMD29 and CMD30 generates the ILLEGAL_COMMAND error.

4.4 Clock Control

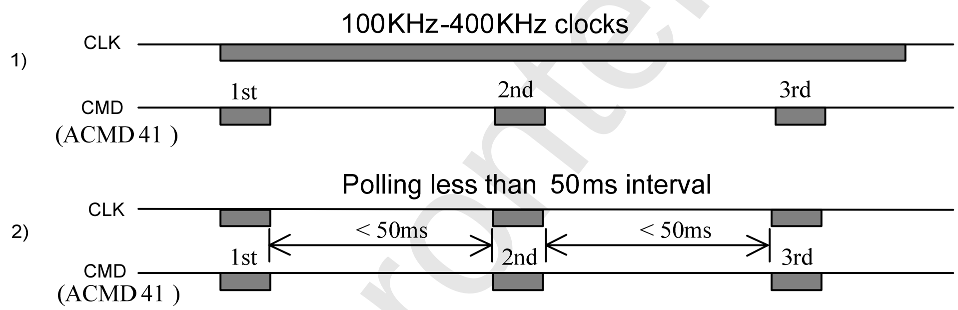
The SD Memory Card bus clock signal can be used by the host to change the cards to energy saving mode or to control the data flow (to avoid under-run or over-run conditions) on the bus. The host is allowed to lower the clock frequency or shut it down. For example, in the case that a host with 512 Bytes of data buffer would like to transfer data to a card with 1 KByte write blocks. So, to preserve a continuous data transfer, from the card's point of view, the clock to the card shall be stopped after the first 512 Bytes. Then the host will fill its internal buffer with another 512 Bytes. After the second half of the write block is ready in the host, it will continue the data transfer to the card by re-starting the clock supply. In such a way that the card does not recognize any interruptions in the data transfer.

There are a few restrictions the host shall consider:

- The bus frequency can be changed at any time (under the restrictions of maximum data transfer frequency and the identification frequency defined by the specification document).
- An exemption to the above is ACMD41 (SD_APP_OP_COND). After issuing the command ACMD41, the following 1) or 2) procedures shall be done by the host until the card becomes ready.

1) Issue continuous clock in the frequency range of 100 KHz-400 KHz.

If the host wants to stop the clock, poll busy bit by ACMD41 command at less than 50 ms intervals.



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

4.5 Cyclic Redundancy Code (CRC)

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

- **CRC7**

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

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

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

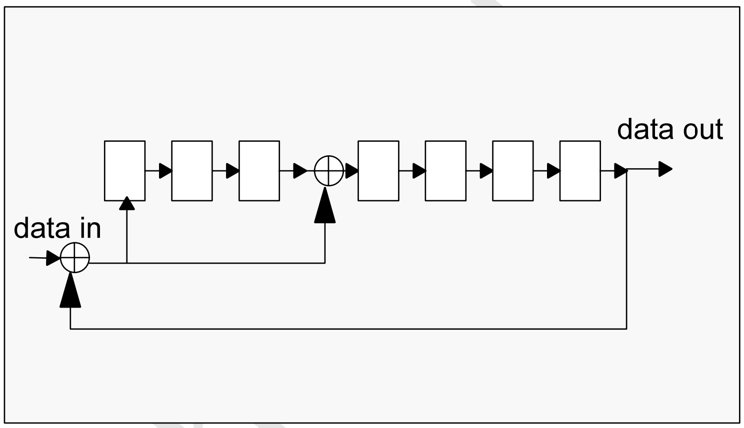


Figure 4-10: CRC7 Generator/Checker

- **CRC7 Examples**

The CRC section of the command/response is bolded.

CMD0 (Argument=0) --> 01 000000 00000000000000000000000000000000 "**1001010**" 1
CMD17 (Argument=0) --> 01 010001 00000000000000000000000000000000 "**0101010**" 1
Response of CMD17 --> 00 010001 00000000000000000000100100000000 "**0110011**" 1

- **CRC16**

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

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

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

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

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

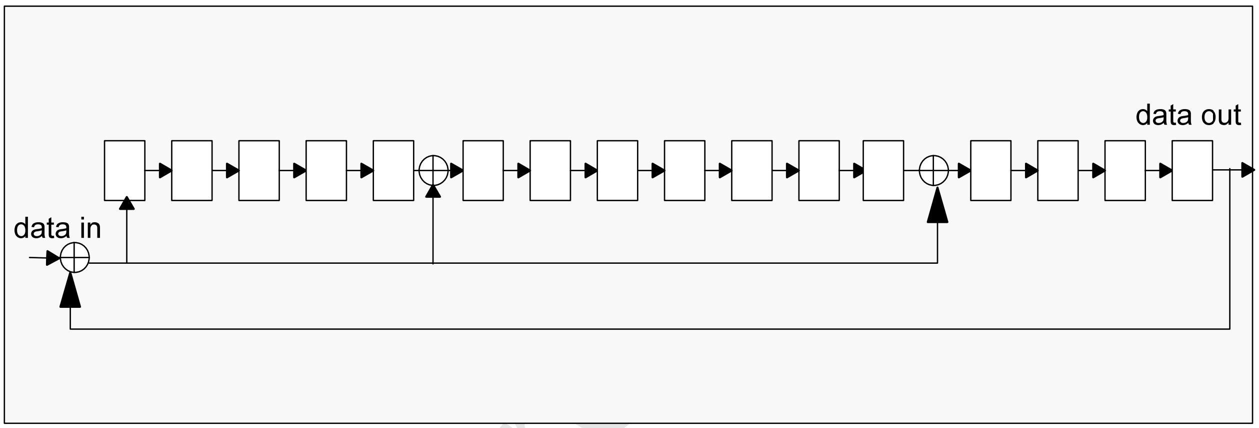


Figure 4-11: CRC16 Generator/Checker

- **CRC16 Example**

512 bytes with 0xFF data --> CRC16 = 0x7FA1

4.6 Error Conditions

4.6.1 CRC and Illegal Command

All commands are protected by CRC (cyclic redundancy check) bits. If the addressed card's CRC check fails, the card does not respond and the command is not executed. The card does not change its state, and COM_CRC_ERROR bit is set in the status register.
Similarly, if an illegal command has been received, a card 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 (see Figure 4-1 and Figure 4-3).
Table 4-29 contains a complete state transition description.
There are different kinds of illegal commands:

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

4.6.2 Read, Write and Erase Timeout Conditions

A card shall complete the command within the time period defined as follows or give up and return an error message. If the host does not get any response with the given timeout it should assume that the card is not going to respond and try to recover (e.g. reset the card, power cycle, reject, etc.).

4.6.2.1 Read

For a Standard Capacity SD Memory Card, the times after which a timeout condition for read operations occurs are (card independent) **either 100 times longer** than the typical access times for these operations given below **or 100 ms (the lower of the two)**. The read access time is defined as the sum of the two times given by the CSD parameters TAAC and NSAC (see Chapter 5.3). In the case of a single read operation, these card parameters define the typical delay between the end bit of the read command and the start bit of the data block. In the case of a multiple-read operation, they also define the typical delay between the end bit of a data block and the start bit of next data block.
A High Capacity SD Memory Card indicates TAAC and NSAC as fixed values. The host should use 100 ms timeout (minimum) for single and multiple read operations rather than using TAAC and NSAC.

4.6.2.2 Write

For a Standard Capacity SD Memory Card, the times after which a timeout condition for write operations occurs are (card independent) **either 100 times longer** than the typical program times for these operations given below **or 250 ms (the lower of the two).** 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 commands (e.g. SET(CLR)_WRITE_PROTECT, PROGRAM_CSD and the block write commands).
A High Capacity SD Memory Card indicates R2W_FACTOR as a fixed value.
Maximum length of busy is defined as 250 ms for all write operations. The host should use 250 ms timeout (minimum) for single and multiple write operations rather than using R2W_FACTOR.

4.6.2.3 Erase

If the card supports parameters for erase timeout calculation in the SD Status, the host should use them to determine erase timeout (see Chapter 4.10.2). If the card does not support these parameters, erase timeout can be estimated by block write delay.
The duration of an erase command can be estimated by the number of write blocks (WRITE_BL) to be erased multiplied by 250 ms.

4.7 Commands

4.7.1 Command Types

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

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

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

4.7.2 Command Format

All commands have a fixed code length of 48 bits, needing a transmission time of 1.92 μs @ 25 MHz and 0.96 μs @ 50 MHz.

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

Table 4-17: Command Format

A command always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (host = 1). The next 6 bits indicate the index of the command, this value being interpreted as a binary coded number (between 0 and 63). Some commands need an argument (e.g. an address), which is coded by 32 bits. A value denoted by ‘x’ in the table above indicates this variable is dependent on the command. All commands are protected by a CRC (see Chapter 4.5 for the definition of CRC7). Every command codeword is terminated by the end bit (always 1). All commands and their arguments are listed in Table 4-19-Table 4-28.

4.7.3 Command Classes

The command set of the SD Memory Card system is divided into several classes (See Table 4-18). Each class supports a set of card functionalities.
Table 4-18 determines the setting of CCC from the card supported commands. A CCC bit, which corresponds to a supported command number, is set to 1. A class in CCC includes mandatory commands is always set to 1. Cards with specific functions may need to support some optional commands. For example, Combo Card shall support CMD5.

Class 0, 2, 4, 5, 7 and 8 are mandatory and shall be supported by all SD Memory Cards. The other classes are optional. The supported Card Command Classes (CCC) are coded as a parameter in the card specific data (CSD) register of each card, providing the host with information on how to access the card.

	Card Command Class (CCC)	0	1	2	3	4	5	6	7	8	9	10	11
Supported commands	**class description**	**basic**	**reserved**	**block read**	**reserved**	**block write**	**erase**	**write pro-tec-tion**	**lock card**	**appli-cation spe-cific**	**I/O mode**	**switch**	**reserved**
CMD0	Mandatory	+											
CMD2	Mandatory	+											
CMD3	Mandatory	+											
CMD4	Mandatory	+											
CMD5	Optional										+		
CMD6[2]	Mandatory											+	
CMD7	Mandatory	+											
CMD8[3]	Mandatory	+											
CMD9	Mandatory	+											
CMD10	Mandatory	+											
CMD12	Mandatory	+											
CMD13	Mandatory	+											
CMD15	Mandatory	+											
CMD16	Mandatory			+		+			+				
CMD17	Mandatory			+									
CMD18	Mandatory			+									
CMD24	Mandatory[1]					+							
CMD25	Mandatory[1]					+							
CMD27	Mandatory[1]					+							
CMD28	Optional							+					
CMD29	Optional							+					
CMD30	Optional							+					
CMD32	Mandatory[1]						+						
CMD33	Mandatory[1]						+						
CMD34-37[2]	Optional											+	
CMD38	Mandatory[1]						+						
CMD42	Mandatory[4]								+				
CMD50[2]	Optional											+	
CMD52	Optional										+		
CMD53	Optional										+		
CMD55	Mandatory									+			
CMD56	Mandatory									+			
CMD57[2]	Optional											+	

	Card Command Class (CCC)	0	1	2	3	4	5	6	7	8	9	10	11
Supported commands	**class description**	**basic**	**reserved**	**block read**	**reserved**	**block write**	**erase**	**write protection**	**lock card**	**application specific**	**I/O mode**	**switch**	**reserved**
ACMD6	Mandatory									+			
ACMD13	Mandatory									+			
ACMD22	Mandatory[1]									+			
ACMD23	Mandatory[1]									+			
ACMD41	Mandatory									+			
ACMD42	Mandatory									+			
ACMD51	Mandatory									+			

Note (1): The write related commands are mandatory only for the Writable type of Cards (OTP and R/W).
Note (2): This command was defined in version 1.10
Note (3): This command is newly defined in version 2.00
Note (4): This command is optional in Version 1.01 and 1.10 and mandatory from Version 2.00

Table 4-18: Card Command Classes (CCCs)

4.7.4 Detailed Command Description

The following tables describe in detail all SD Memory Card bus commands. The responses R1-R3, R6 are defined in Chapter 4.9. The registers CID, CSD and DSR are described in Chapter 5. The card shall ignore stuff bits and reserved bits in a argument.

CMD INDEX	type	argument	resp	abbreviation	command description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all cards to idle state
CMD1	reserved				
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks any card to send the CID numbers on the CMD line (any card that is connected to the host will respond)
CMD3	bcr	[31:0] stuff bits	R6	SEND_RELATIVE_ ADDR	Ask the card to publish a new relative address (RCA)
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards
CMD5	reserved for I/O cards (refer to the "SDIO Card Specification")				
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1b (only from the selected card)	SELECT/DESELECT _CARD	Command toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address; address 0 deselects all. In the case that the RCA equals 0, then the host may do one of the following: - Use other RCA number to perform card de-selection. - Re-send CMD3 to change its RCA number to other than 0 and then use CMD7 with RCA=0 for card de-selection.
CMD8	bcr	[31:12]reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition, which includes host supply voltage information and asks the card whether card supports voltage. Reserved bits shall be set to '0'.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card identi-fication (CID) on CMD the line.
CMD11	reserved				
CMD12	ac	[31:0] stuff bits	R1b	STOP_ TRANSMISSION	Forces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	reserved				

CMD INDEX	type	argument	resp	abbreviation	command description
CMD15	ac	[31:16] RCA [15:0] reserved bits	-	GO_INACTIVE_ STATE	Sends an addressed card into the *Inactive State*. This command is used when the host explicitly wants to deactivate a card. Reserved bits shall be set to '0'.

Table 4-19: Basic Commands (class 0)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	In the case of a Standard Capacity SD Memory Card, this command sets the block length (in bytes) for all following block commands (read, write, lock). Default block length is fixed to 512 Bytes. Set length is valid for memory access commands only if partial block read operation are allowed in CSD. In the case of a High Capacity SD Memory Card, block length set by CMD16 command does not affect the memory read and write commands. Always 512 Bytes fixed block length is used. This command is effective for LOCK_UNLOCK command. In both cases, if block length is set larger than 512Bytes, the card sets the BLOCK_LEN_ERROR bit.
CMD17	adtc	[31:0] data address[2]	R1	READ_SINGLE_ BLOCK	In the case of a Standard Capacity SD Memory Card, this command, this command reads a block of the size selected by the SET_BLOCKLEN command. [1] In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address[2]	R1	READ_MULTIPLE_ BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command. Block length is specified the same as READ_SINGLE_BLOCK command.
CMD19 ... CMD23	reserved				

1) The data transferred shall not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.
2) Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.

Table 4-20: Block-Oriented Read Commands (class 2)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	See description in Table 4-20
CMD24	adtc	[31:0] data address[2]	R1	WRITE_BLOCK	In the case of a Standard Capacity SD Memory Card, this command writes a block of the size selected by the SET_BLOCKLEN command.[1] In the case of a High Capacity Card, block length is fixed 512 Bytes regardless of the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address[2]	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows. Block length is specified the same as WRITE_BLOCK command.
CMD26	Reserved For Manufacturer				
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

1) The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD. In the case that write partial blocks is not supported, then the block length=default block length (given in CSD).
2) Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.

Table 4-21: Block-Oriented Write Commands (class 4)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD28	ac	[31:0] data address[2]	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). A High Capacity SD Memory Card does not support this command.
CMD29	ac	[31:0] data address[2]	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group. A High Capacity SD Memory Card does not support this command.
CMD30	adtc	[31:0] write protect data address[2]	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.[1] A High Capacity SD Memory Card does not support this command.
CMD31	reserved				

1) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (least significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits shall be set to 0.
2) Data address is in byte units in a Standard Capacity SD Memory Card.

Table 4-22: Block Oriented Write Protection Commands (class 6)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD32	ac	[31:0] data address[1]	R1	ERASE_WR_BLK_START	Sets the address of the first write block to be erased.
CMD33	ac	[31:0] data address[1]	R1	ERASE_WR_BLK_END	Sets the address of the last write block of the continuous range to be erased.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erases all previously selected write blocks.
CMD39	reserved				
CMD40					Not valid in SD Memory Card - Reserved for MultiMediaCard I/O mode
CMD41	reserved				

1) Data address is in byte units in a Standard Capacity SD Memory Card and in block (512 Byte) units in a High Capacity SD Memory Card.

Table 4-23: Erase Commands (class 5)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	See description in Table 4-20
CMD42	adtc	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.
CMD43-49 CMD51	reserved				

Table 4-24: Lock Card (class 7)

CMD INDEX	type	argument	resp	abbreviation	command description
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather than a standard command
CMD56	adtc	[31:1] stuff bits. [0]: RD/WR	R1	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose/application specific commands. In the case of a Standard Capacity SD Memory Cards, the size of the data block shall be set by the SET_BLOCK_LEN command. In the case of a High Capacity SD Memory Cards, the size of the data block is fixed to 512 byte. The host sets RD/WR=1 for reading data from the card and sets to 0 for writing data to the card.
CMD58-59	reserved				
CMD60-63	reserved for manufacturer				

Table 4-25: Application-specific Commands (class 8)

All the application-specific commands (given in Table 4-25) are supported if Class 8 is allowed (mandatory in SD Memory Card).

CMD INDEX	type	argument	resp	abbreviation	command description
CMD52... CMD54	reserved for I/O mode (refer to the "SDIO Card Specification")				

Table 4-26: I/O Mode Commands (class 9)

All future reserved commands shall have a codeword length of 48 bits, as well as their responses (if there are any).

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

ACMD INDEX	type	argument	resp	abbreviation	command description
ACMD6	ac	[31:2] stuff bits [1:0]bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4 bits bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Status. The status fields are given in Table 4-38.
ACMD17	reserved				
ACMD18	--	--	--	--	Reserved for SD security applications[1]
ACMD19 to ACMD21	reserved				
ACMD22	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_ BLOCKS	Send the number of the written (without errors) write blocks. Responds with 32bit+CRC data block. If WRITE_BL_PARTIAL='0', the unit of ACMD22 is always 512 byte. If WRITE_BL_PARTIAL='1', the unit of ACMD22 is a block length which was used when the write command was executed.
ACMD23	ac	[31:23] stuff bits [22:0]Number of blocks	R1	SET_WR_BLK_ ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"=default (one wr block)[2].
ACMD24	reserved				
ACMD25	--	--	--	--	Reserved for SD security applications[1]
ACMD26	--	--	--	--	Reserved for SD security applications[1]
ACMD38	--	--	--	--	Reserved for SD security applications[1]
ACMD39 to ACMD40	reserved				

ACMD INDEX	type	argument	resp	abbreviation	command description
ACMD41	bcr	[31]reserved bit [30]HCS(OCR[30]) [29:24]reserved bits [23:0] V_{DD} Voltage Window(OCR[23:0])	R3	SD_SEND_OP_COND	Sends host capacity support information (HCS) and asks the accessed card to send its operating condition register (OCR) content in the response on the CMD line. HCS is effective when card receives SEND_IF_COND command. Reserved bit shall be set to '0'. CCS bit is assigned to OCR[30].
ACMD42	ac	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_ DETECT	Connect[1]/Disconnect[0] the 50 KOhm pull-up resistor on CD/DAT3 (pin 1) of the card.
ACMD43 ACMD49	--	--	--	--	Reserved for SD security applications[1]
ACMD51	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1) Refer to the "SD Specifications Part3 Security Specification" for a detailed explanation about the SD Security Features
2) Command STOP_TRAN (CMD12) shall be used to stop the transmission in Write Multiple Block whether or not the pre-erase (ACMD23) feature is used.

Table 4-27: Application Specific Commands used/reserved by SD Memory Card

Table 4-28 was added in version 1.10

CMD INDEX	type	argument	resp	abbreviation	command description
CMD6	adtc	[31] Mode 0:Check function 1:Switch function [30:24] reserved (All '0') [23:20] reserved for function group 6 (0h or Fh) [19:16] reserved for function group 5 (0h or Fh) [15:12] reserved for function group 4 (0h or Fh) [11:8] reserved for function group 3 (0h or Fh) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_F UNC	Checks switchable function (mode 0) and switch card function (mode 1). See Chapter 4.3.10.
CMD34 CMD35 CMD36 CMD37 CMD50 CMD57	Reserved for each command system set by switch function command (CMD6). Detailed definition is referred to each command system specification.				

Table 4-28: Switch Function Commands (class 10)

4.8 Card State Transition Table

Table 4-29 defines the card state transitions dependant on the received command.

	current state									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
Trigger of state change	**next state**									
class independent										
"Operation Complete"	-	-	-	-	-	-	-	tran	stby	-
class 0										
CMD0	idle	idle	idle	idle	idle	idle	idle	idle	idle	-
CMD2	-	ident	-	-	-	-	-	-	-	-
CMD3	-	-	stby	stby	-	-	-	-	-	-
CMD4	-	-	-	stby	-	-	-	-	-	-
CMD7, card is addressed	-	-	-	tran	-	-	-	-	prg	-
CMD7, card is not addressed	-	-	-	stby	stby	stby	-	dis	-	-
CMD8	idle	-	-	-	-	-	-	-	-	-
CMD9	-	-	-	stby	-	-	-	-	-	-
CMD10	-	-	-	stby	-	-	-	-	-	-
CMD12	-	-	-	-	-	tran	prg	-	-	-
CMD13	-	-	-	stby	tran	data	rcv	prg	dis	-
CMD15	-	-	-	ina	ina	ina	ina	ina	ina	-
class 2										
CMD16	-	-	-	-	tran	-	-	-	-	-
CMD17	-	-	-	-	data	-	-	-	-	-
CMD18	-	-	-	-	data	-	-	-	-	-
class 4										
CMD16	see class 2									
CMD24	-	-	-	-	rcv	-	-	-	-	-
CMD25	-	-	-	-	rcv	-	-	-	-	-
CMD27	-	-	-	-	rcv	-	-	-	-	-
class 6										
CMD28	-	-	-	-	prg	-	-	-	-	-
CMD29	-	-	-	-	prg	-	-	-	-	-
CMD30	-	-	-	-	data	-	-	-	-	-
class 5										
CMD32	-	-	-	-	tran	-	-	-	-	-
CMD33	-	-	-	-	tran	-	-	-	-	-
CMD38	-	-	-	-	prg	-	-	-	-	-
class 7										

	current state									
	idle	**ready**	**ident**	**stby**	**tran**	**data**	**rcv**	**prg**	**dis**	**ina**
CMD42	-	-	-	-	rcv	-	-	-	-	-
class 8										
CMD55	idle	-	-	stby	tran	data	rcv	prg	dis	-
CMD56; RD/WR = 0	-	-	-	-	rcv	-	-	-	-	-
CMD56; RD/WR = 1	-	-	-	-	data	-	-	-	-	-
ACMD6	-	-	-	-	tran	-	-	-	-	-
ACMD13	-	-	-	-	data	-	-	-	-	-
ACMD22	-	-	-	-	data	-	-	-	-	-
ACMD23	-	-	-	-	tran	-	-	-	-	-
ACMD18,25,26,38, 43,44,45,46,47,48,49	Refer to the "SD Specifications Part3 Security Specification" for information about the SD Security Features									
ACMD41, OCR check is OK and card is not busy	ready	-	-	-	-	-	-	-	-	-
ACMD41, OCR check is OK and card is busy[2]	idle	-	-	-	-	-	-	-	-	-
ACMD41, OCR check fails	ina	-	-	-	-	-	-	-	-	-
ACMD41, query mode	idle	-	-	-	-	-	-	-	-	-
ACMD42	-	-	-	-	tran	-	-	-	-	-
ACMD51	-	-	-	-	data	-	-	-	-	-
class 9										
CMD52-CMD54	refer to the "SDIO Card Specification"									
class 10[1]										
CMD6	-	-	-	-	data	-	-	-	-	-
CMD34-37,50,57	refer to specifications of each functions									
class 11										
CMD41, CMD43...CMD49, CMD58-CMD59	reserved									
CMD60...CMD63	reserved for manufacturer									

Note (1): Class 10 commands were defined in version 1.10
Note (2): Card returns busy in case of following.
- Card executes internal initialization process
- Card is a High Capacity SD Memory Card and host does not support High Capacity.

Table 4-29: Card State Transition Table

The state transitions of the SD Memory Card application-specific commands are given under Class 8, above.

4.9 Responses

All responses are sent via the command line CMD. The response transmission always starts with the left bit of the bit string corresponding to the response codeword. The code length depends on the response type.
A response always starts with a start bit (always 0), followed by the bit indicating the direction of transmission (card = 0). A value denoted by 'x' in the tables below indicates a variable entry. All responses except for the type R3 (see below) are protected by a CRC (see Chapter 4.5 for the definition of CRC7). Every command codeword is terminated by the end bit (always 1).

There are five types of responses for the SD Memory Card. The SDIO Card supports additional response types named R4 and R5. Refer to SDIO Card Spec for detailed information on the SDIO commands and responses. Their formats are defined as follows:

4.9.1 R1 (normal response command):

Code length is 48 bits. The bits 45:40 indicate the index of the command to be responded to, this value being interpreted as a binary coded number (between 0 and 63). The status of the card is coded in 32 bits. Note that if a data transfer to the card is involved, then a busy signal may appear on the data line after the transmission of each block of data. The host shall check for busy after data block transmission. The card status is described in Chapter 4.10.

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

Table 4-30: Response R1

4.9.2 R1b

R1b is identical to R1 with an optional busy signal transmitted on the data line. The card may become busy after receiving these commands based on its state prior to the command reception. The Host shall check for busy at the response. Refer to Chapter 4.12.3 for a detailed description and timing diagrams.

4.9.3 R2 (CID, CSD register)

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

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

Table 4-31: Response R2

4.9.4 R3 (OCR register)

Code length is 48 bits. The contents of the OCR register are sent as a response to ACMD41.

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

Table 4-32: Response R3

4.9.5 R6 (Published RCA response)

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

Bit position	47	46	[45:40]	[39:8] Argument field		[7:1]	0
Width (bits)	1	1	6	16	16	7	1
Value	'0'	'0'	x	x	x	x	'1'
Description	start bit	transmission bit	command index ('000011')	New published RCA [31:16] of the card	[15:0] card status bits: 23,22,19,12:0 (see Table 4-36)	CRC7	end bit

Table 4-33: Response R6

4.9.6 R7 (Card interface condition)

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

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

Table 4-34: Response R7

Table 4-35 shows the format of 'voltage accepted' in R7.

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

Table 4-35: Voltage Accepted in R7

4.10 Two Status Information of SD Memory Card

The SD Memory Card supports two status fields as follows:
- '*Card Status*': Error and state information of a executed command, indicated in the response
- '*SD Status*': Extended status field of 512 bits that supports special features of the SD Memory Card and future Application-Specific features.

4.10.1 Card Status

The response format R1 contains a 32-bit field named *card status*. This field is intended to transmit the card's status information (which may be stored in a local status register) to the host. If not specified otherwise, the status entries are always related to the previous issued command.
Table 4-36 defines the different entries of the status. The type and clear condition fields in the table are abbreviated as follows:

- Type:
  •E: Error bit.
  •S: Status bit.
  •R: Detected and set for the actual command response.
  •X: Detected and set during command execution. The host can get the status by issuing a command with R1 response.

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

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	E R X	'0'= no error '1'= error	The command's argument was out of the allowed range for this card.	C
30	ADDRESS_ERROR	E R X	'0'= no error '1'= error	A misaligned address which did not match the block length was used in the command.	C
29	BLOCK_LEN_ERROR	E R X	'0'= no error '1'= error	The transferred block length is not allowed for this card, or the number of transferred bytes does not match the block length.	C
28	ERASE_SEQ_ERROR	E R	'0'= no error '1'= error	An error in the sequence of erase commands occurred.	C
27	ERASE_PARAM	E R X	'0'= no error '1'= error	An invalid selection of write-blocks for erase occurred.	C
26	WP_VIOLATION	E R X	'0'= not protected '1'= protected	Set when the host attempts to write to a protected block or to the temporary or permanent write protected card.	C
25	CARD_IS_LOCKED	S X	'0' = card unlocked '1' = card locked	When set, signals that the card is locked by the host	A
24	LOCK_UNLOCK_FAILED	E R X	'0' = no error '1' = error	Set when a sequence or password error has been detected in lock/unlock card command.	C
23	COM_CRC_ERROR	E R	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	E R	'0'= no error '1'= error	Command not legal for the card state	B
21	CARD_ECC_FAILED	E R X	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
20	CC_ERROR	E R X	'0'= no error '1'= error	Internal card controller error	C
19	ERROR	E R X	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
18	reserved				
17	reserved				
16	CSD_OVERWRITE	E R X	'0'= no error '1'= error	Can be either one of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
15	WP_ERASE_SKIP	E R X	'0'= not protected '1'= protected	"Set when only partial address space was erased due to existing write	C

Bits	Identifier	Type	Value	Description	Clear Condition
				protected blocks or the temporary or permanent write protected card was erased.	
14	CARD_ECC_DISABLED	S X	'0'= enabled '1'= disabled	The command has been executed without using the internal ECC.	A
13	ERASE_RESET	S R	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
12:9	CURRENT_STATE	S X	0 = idle 1 = ready 2 = ident 3 = stby 4 = tran 5 = data 6 = rcv 7 = prg 8 = dis 9-14 = reserved 15 = reserved for I/O mode	The state of the card when receiving the command. If the command execution causes a state change, it will be visible to the host in the response to the next command. The four bits are interpreted as a binary coded number between 0 and 15.	B
8	READY_FOR_DATA	S X	'0'= not ready '1'= ready	Corresponds to buffer empty signaling on the bus	A
7:6					
5	APP_CMD	S R	'0' = Disabled '1' = Enabled	The card will expect ACMD, or an indication that the command has been interpreted as ACMD	C
4	reserved for SD I/O Card				
3	AKE_SEQ_ERROR (SD Memory Card app. spec.)	E R	'0' = no error '1' = error	Error in the sequence of the authentication process	C
2	reserved for application specific commands				
1, 0	reserved for manufacturer test mode				

Table 4-36: Card Status

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

CMD#	Response Format 1 Status bit #																					
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12:9**	**8**	**5**
3[1]									x	x			x							x		
6[2]	x						x		x	x	x	x	x	x	x					x		
7					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
12	x	x				x	x		x	x	x	x	x	x	x			x		x		
13	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	
16			x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
17	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
18	x	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
24	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
25	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
26					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
27					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
28	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
29	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
30	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
32	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
33	x			x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
38				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		
55					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
56					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
ACMD6	x				x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD13					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD22					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD23					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD42					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x
ACMD51					x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x

(1) The response to CMD3 is R6 that includes only bits 23, 22, 19 and 12:9 out of the Card Status
(2) This command was defined in version 1.10

Table 4-37: Card Status Field/Command - Cross Reference

4.10.2 SD Status

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

The same abbreviation for 'type' and 'clear condition' were used as for the Card Status above.

Bits	Identifier	Type	Value	Description	Clear Condition
511: 510	DAT_BUS_WIDTH	S R	'00'= 1 (default) '01'= reserved '10'= 4 bit width '11'= reserved	Shows the currently defined data bus width that was defined by SET_BUS_WIDTH command	A
509	SECURED_MODE	S R	'0'= Not in the mode '1'= In Secured Mode	Card is in Secured Mode of operation (refer to the "SD Security Specification").	A
508: 496	reserved				
495: 480	SD_CARD_TYPE	SR	'00xxh'= SD Memory Cards as defined in Physical Spec Ver1.01-2.00 ('x'=don't care). The following cards are currently defined: '0000'= Regular SD RD/WR Card. '0001'= SD ROM Card	In the future, the 8 LSBs will be used to define different variations of an SD Memory Card (Each bit will define different SD Types). The 8 MSBs will be used to define SD Cards that do not comply with current SD Physical Layer Specification.	A
479: 448	SIZE_OF_PROTECTED_AREA	SR	Size of protected area (See below)	(See below)	A
447: 440	SPEED_CLASS	SR	Speed Class of the card (See below)	(See below)	A
439: 432	PERFORMANCE_MOVE	SR	Performance of move indicated by 1 [MB/s] step. (See below)	(See below)	A
431: 428	AU_SIZE	SR	Size of AU (See below)	(See below)	A
427: 424	reserved				
423: 408	ERASE_SIZE	SR	Number of AUs to be erased at a time	(See below)	A
407: 402	ERASE_TIMEOUT	SR	Timeout value for erasing areas specified by UNIT_OF_ERASE_AU	(See below)	A
401: 400	ERASE_OFFSET	SR	Fixed offset value added to erase time.	(See below)	A

Bits	Identifier	Type	Value	Description	Clear Condition
399: 312	reserved				
311:0	reserved for manu-facturer				

Table 4-38: SD Status

- **SIZE_OF_PROTECTED_AREA**

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

In the case of a High Capacity Card, the capacity of protected area is specified in this field:
Protected Area = SIZE_OF_PROTECTED_AREA
SIZE_OF_PROTECTED_AREA is specified by the unit in byte.

- **SPEED_CLASS**

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

SPEED_CLASS	Value Definition
00h	Class 0
01h	Class 2
02h	Class 4
03h	Class 6
04h – FFh	Reserved

Table 4-39: Speed Class Code Field

- **PERFORMANCE_MOVE**

This 8-bit field indicates Pm and the value can be set by 1 [MB/sec] step. If the card does not move used RUs, Pm should be considered as infinity. Setting to FFh means infinity. The minimum value of Pm is defined by in Table 4-49.

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

Table 4-40: Performance Move Field

- **AU_SIZE**

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

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

Table 4-41: AU_SIZE Field

The maximum AU size, depends on the card capacity, is defined in Table 4-42.. The card can set any AU size between RU size and maximum AU size.

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

Table 4-42: Maximum AU size

Application Notes:
The host should use the maximum AU Size (4 MB) to determine host buffer size.
The host can treat multiple AUs combined as one unit.

- **ERASE_SIZE**

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

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

Table 4-43: Erase Size Field

- **ERASE_TIMEOUT**

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

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

Table 4-44: Erase Timeout Field

- **ERASE_OFFSET**

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

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

Table 4-45: Erase Offset Field

4.11 Memory Array Partitioning

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

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

- Block: is the unit that is related to the block oriented read and write commands. Its size is the number of bytes that 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 that have erasable memory cells, special erase commands are defined. The granularity of the erasable units is in general not the same as for the block oriented commands:
- Sector: is the unit that is related to the erase commands. Its size is the number of blocks that 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. Note that if the card specifies AU size, sector size should be ignored.
- AU (Allocation Unit): is a physical boundary of the card and consists of one or more blocks and its size depends on each card. The maximum AU size is defined for memory capacity. Furthermore AU is the minimal unit in which the card guarantees its performance for devices which complies with Speed Class Specification. The information about the size and the Speed Class are stored in the SD Status. AU is also used to calculate the erase timeout.
- WP-Group: is the minimal unit that may have individual write protection for devices which support write-protected group. Its size is the number of groups that 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. The High Capacity SD Memory Card does not support the write protect group command.

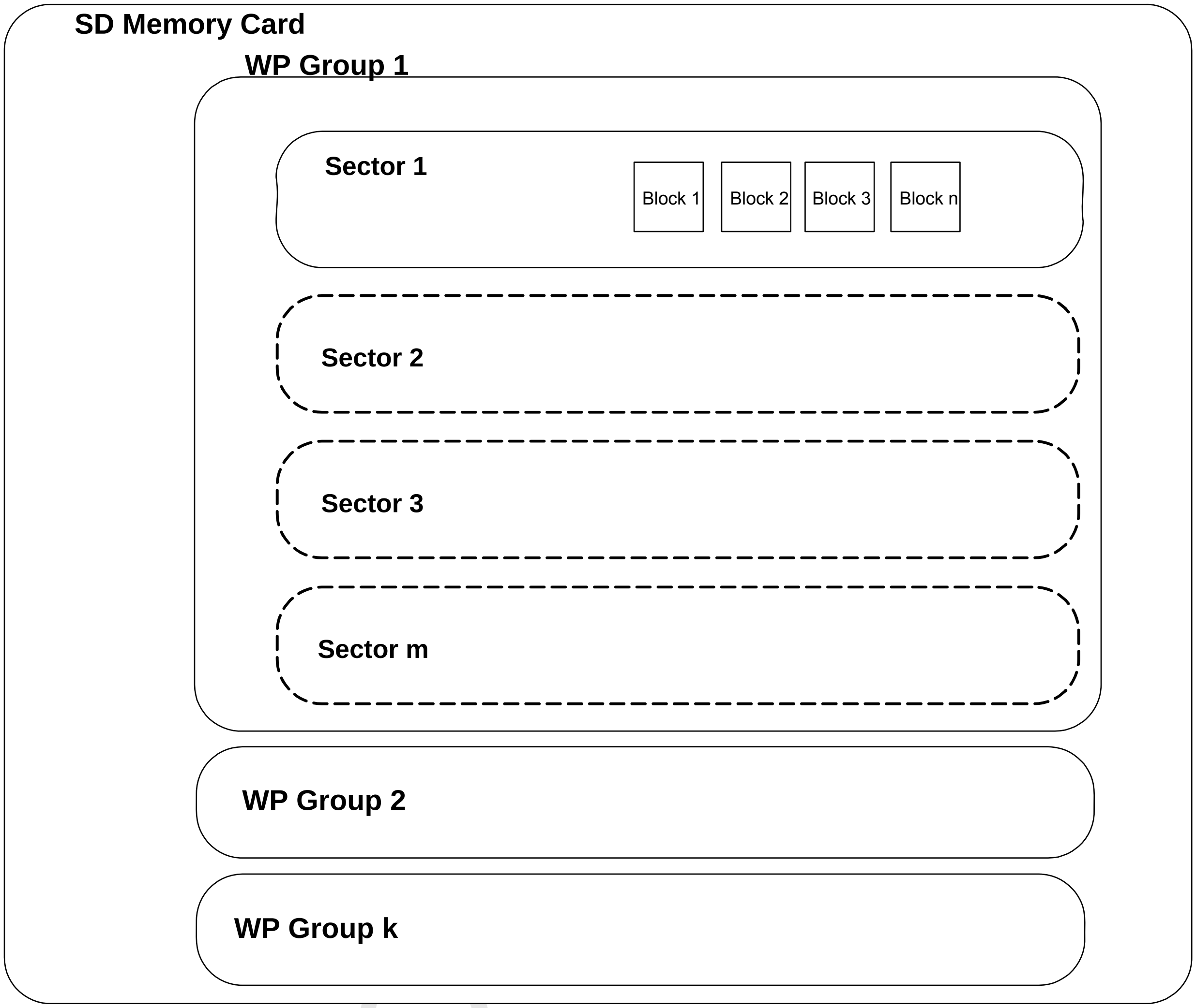


Figure 4-12: Write Protection Hierarchy

Each WP-group may have an additional write protection bit. The write protection bits are programmable via special commands (see Chapter 4.7.4).
Both functions are optional and only useful for writable/erasable devices. The write protection may also be useful for multi type cards (e.g. a ROM - Flash combination). The information about the availability is stored in the CSD.

4.12 Timings

All timing diagrams use the following schematics and abbreviations:

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

Table 4-46: Timing Diagram Symbols

The difference between the P-bit and Z-bit is that a P-bit is actively driven to HIGH by the card respective of the host output driver, while Z-bit is driven to (respectively kept) HIGH by the pull-up resistors R_{CMD} respectively R_{DAT}. Actively-driven P-bits are less sensitive to noise.
All timing values are defined in Table 4-47.

4.12.1 Command and Response

Both host commands and card responses are clocked per the timing specified in Section 6.8 (and Section 6.9 for high speed card)

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

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

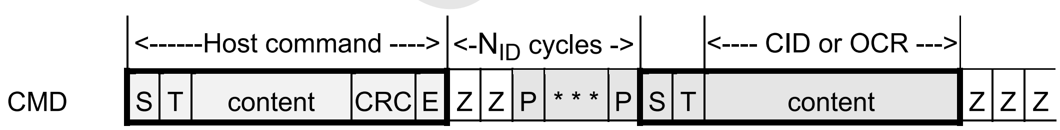


Figure 4-13: Identification Timing (card identification mode)

- **Assign a card relative address**

The SEND_RELATIVE_ADDR (CMD 3) for SD Memory Card timing is given bellow. The minimum delay between the host command and card response is N_{CR} clock cycles.

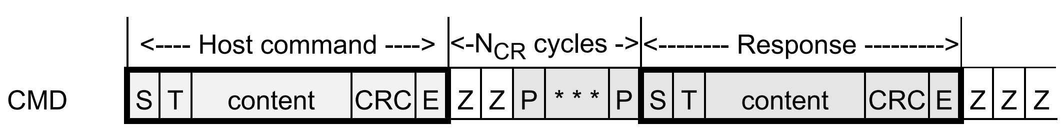


Figure 4-14: SEND_RELATIVE_ADDR Timing

- **Data transfer mode.**

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

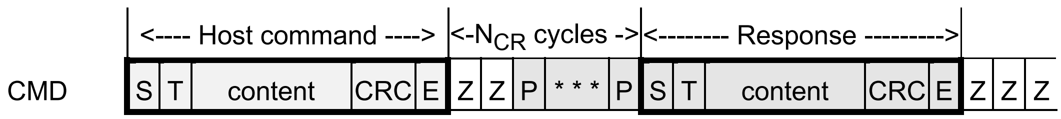


Figure 4-15: Command Response Timing (data transfer mode)

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

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

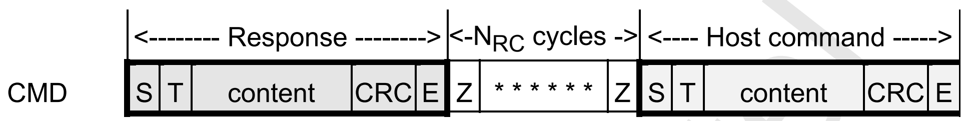


Figure 4-16: Timing of Response End to next CMD Start (data transfer mode)

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

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

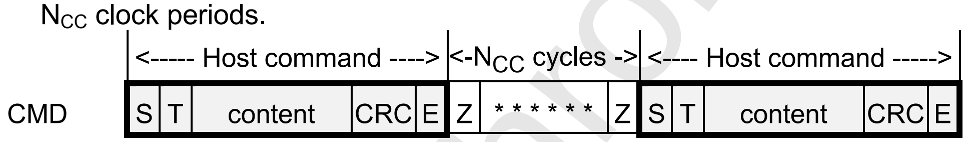


Figure 4-17: Timing of Command Sequences (all modes)

4.12.2 Data Read

*** Note: DAT line represents data bus (either 1 or 4 bits).**

- **Single Block Read**

The host selects one card for data read operation by CMD7, and sets the valid block length for block oriented data transfer by CMD16. The basic bus timing for a read operation is given in Figure 4-18. The sequence starts with a single block read command (CMD17) that specifies the start address in the argument field. The response is sent on the CMD line as usual.

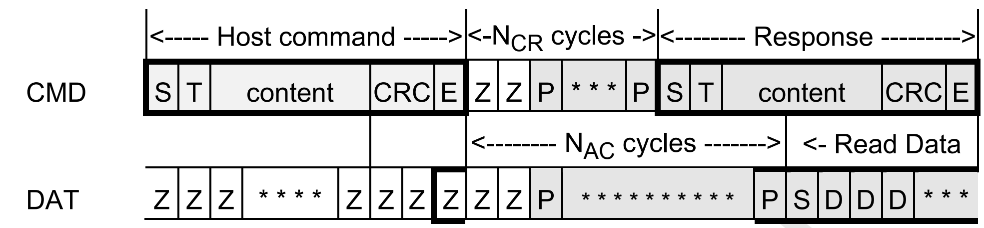


Figure 4-18: Timing of Single Block Read Command

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

- **Multiple Block Read**

In multiple block read mode, the card sends a continuous flow of data blocks following the initial host read command. The data flow is terminated by a stop transmission command (CMD12). Figure 4-19 describes the timing of the data blocks and Figure 4-20 the response to a stop command. The data transmission stops two clock cycles after the end bit of the stop command.

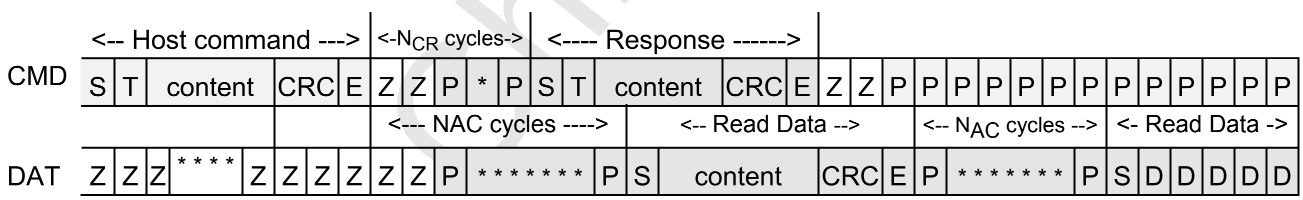


Figure 4-19: Timing of Multiple Block Read Command

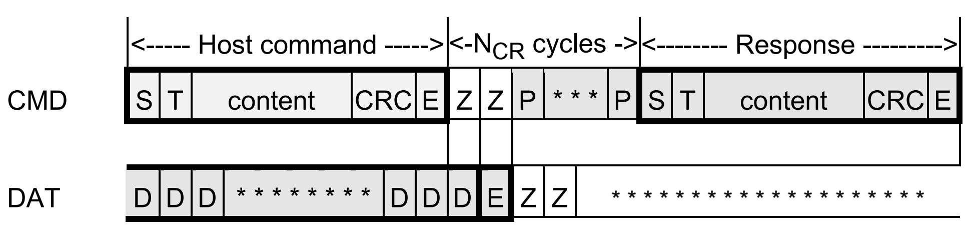


Figure 4-20: Timing of Stop Command (CMD12, data transfer mode)

4.12.3 Data Write

- **Single Block Write**

The host selects one card for data write operation by CMD7.
The host sets the valid block length for block-oriented data transfer by CMD16.
The basic bus timing for a write operation is given in Figure 4-21. The sequence starts with a single block write command (CMD24) that determines (in the argument field) the start address. It is responded by the card on the CMD line as usual. The data transfer from the host starts N_{WR} clock cycles after the card response was received.
The data is suffixed with CRC check bits to allow the card to check it for transmission errors. The card sends back the CRC check result as a CRC status token on the DAT0 line. In the case of a transmission error the card sends a negative CRC status (‘101’). In the case of a non-erroneous transmission, the card sends a positive CRC status (‘010’) and starts the data programming procedure. When a flash programming error occurs, the card will ignore all further data blocks. In this case, no CRC response will be sent to the host and, therefore, there will not be CRC start bit on the bus and the three CRC status bits will read (‘111‘).

	<-Host cmnd->	<- N_{CR} ->	<-Card response >																						
CMD	E	Z	Z	P	*	P	S	T	Content	CRC	E	Z	Z	P	******************	P	P	P	P	P	P	P	P
	<-N_{WR}->	<- Write data ->			CRC status	<- Busy ->																			
DAT0	Z	Z	******	Z	Z	Z	***	Z	Z	Z	Z	P * P	S	content	CRC	E	Z	Z	S	Status	E	S	L * L	E	Z
DAT1-3	Z	Z	******	Z	Z	Z	***	Z	Z	Z	Z	P * P	S	content	CRC	E	Z	Z	X	X	X	X	X	X	X	X	X	Z

Figure 4-21: Timing of Single Block Write Command

Note that the CRC response output is always two clocks after the end of data.
If the card does not have a free data receive buffer, the card indicates this condition by pulling down the data line to LOW. The card stops pulling down the DAT0 line as soon as at least one receive buffer for the defined data transfer block length becomes free. This signaling does not give any information about the data write status that should be polled by the host.

- ## Multiple Block Write

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

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

The data flow is terminated by a stop transmission command (CMD12). Figure 4-22 describes the timing of the data blocks with and without card busy signal.

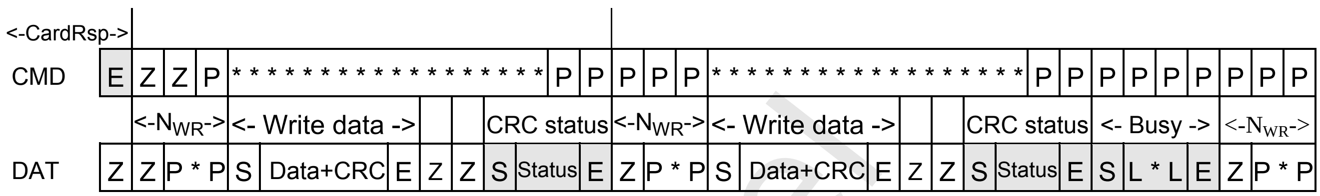


Figure 4-22: Timing of Multiple Block Write Command

The stop transmission command works similar as in the read mode. Figure 4-23 to Figure 4-26 describe the timing of the stop command in different card states.

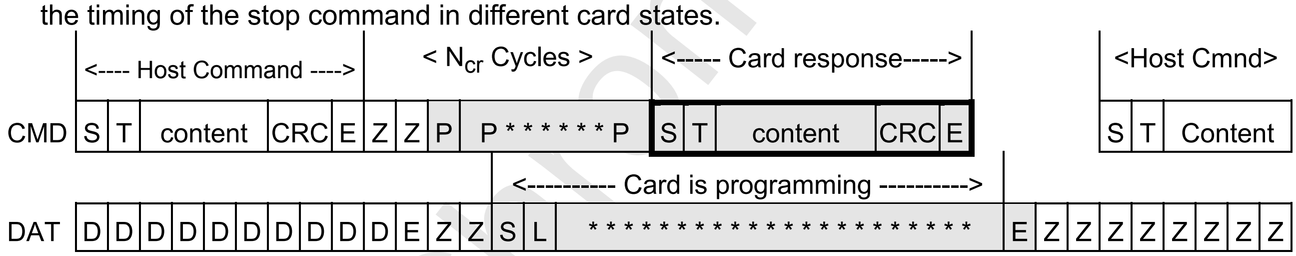
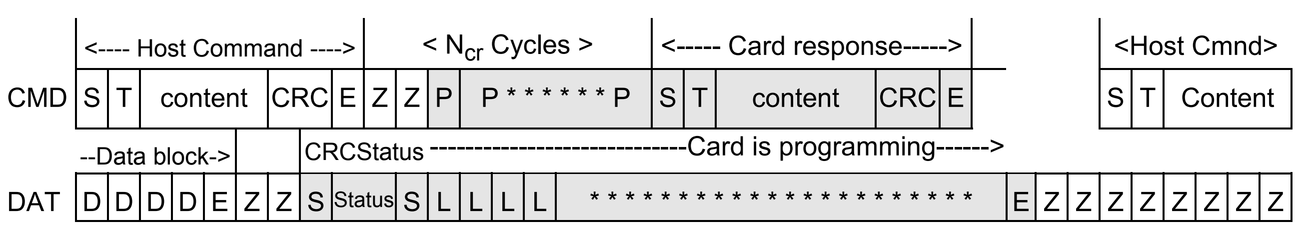


Figure 4-23: Stop Transmission during Data Transfer from the Host

The card will treat a data block as successfully received and ready for programming only if the CRC data of the block was validated and the CRC status token sent back to the host. Figure 4-24 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 and start of busy signaling. In that case, there are no Z clocks, for switching the bus direction, because the bus direction is already towards the host. The received data block in this case is considered incomplete and will not be programmed.



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

Figure 4-24: Stop Transmission during CRC Status Transfer from the Card

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

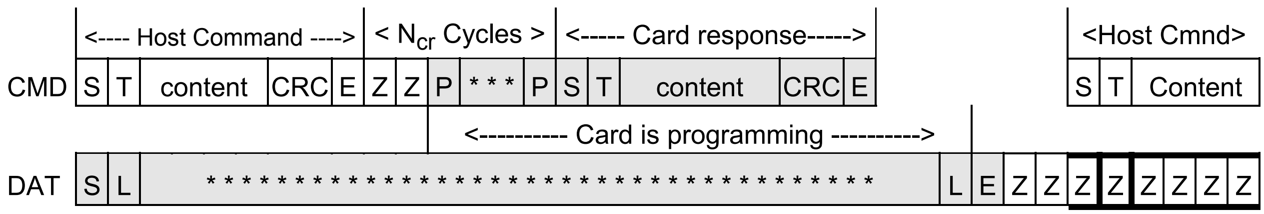


Figure 4-25: Stop Transmission received after Last Data Block. Card is busy programming.

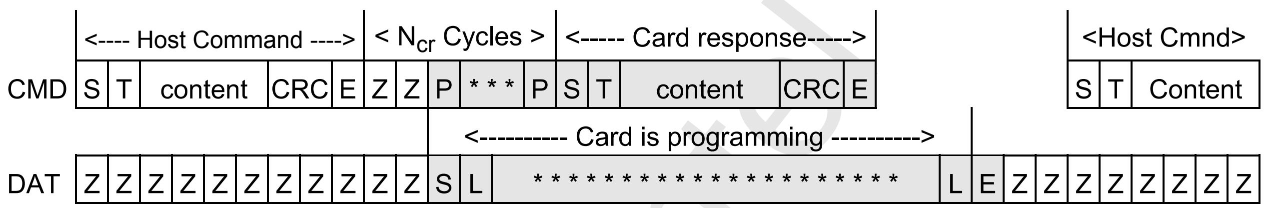


Figure 4-26: Stop Transmission received after Last Data Block. Card becomes busy.

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

The host should first tag the start (CMD32) and end (CMD33) addresses of the range to be erased. The erase command (CMD38), once issued, will erase all the selected write blocks. Similarly, set and clear write protect commands start a programming operation as well. The card will signal “busy” (by pulling the DAT line low) for the duration of the erase or programming operation. The bus transaction timings are the same as given for stop tran command in Figure 4-26.

- **Reselecting a busy card**

When a busy card that is currently in the dis state is reselected it will reinstate its busy signaling on the data line. The timing diagram for this command/response/busy transaction is the same as given for stop tran command in Figure 4-26.

4.12.4 Timing Values

Table 4-47 defines all timing values.

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

1) The maximum read access time for a Standard Capacity SD Memory Card shall be calculated by host as follows:
Nac(max)= 100 ((TAAC * fpp) + (100 * NSAC)) ;
fpp is the interface clock rate and TAAC & NSAC are given in the CSD (Chapter 5.3).
Details of read, write, and erase timeouts are described in 4.6.2
In the case of a High Capacity SD Memory Card, a fixed value (100 ms) shall be used for the maximum read access time.

Table 4-47: Timing Values

4.13 Speed Class Specification

The Speed Class Specification classifies card performance by Speed Class number and offers the method to calculate performance. The specification enables the host to support AV applications to perform real time recording to the SD memory card. Following sections describe the Speed Class specification for the card. Refer to the Application Notes for an example of host implementation.

4.13.1 Allocation Unit (AU)

User Area is divided into units called "**Allocation Unit (AU)**" (Refer to Figure 4-27). Each card has its own fixed **AU Size (S_{AU})** and the maximum AU Size is defined depending on the card's capacity. The host should manage data areas with the unit of AU. Several AUs from AU1 should not be used for real time recording because those may include system information. AV application should start recording from the first complete AU, to which only user data can be recorded. Note that this specification does not apply to Protected Area.

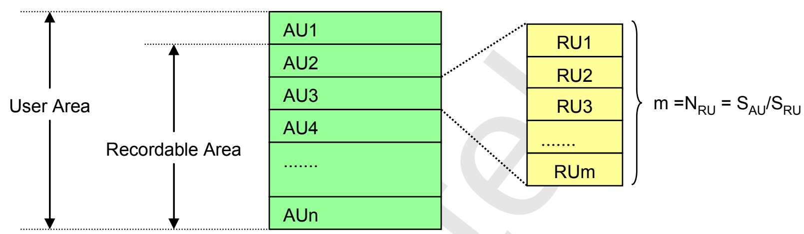


Figure 4-27: Definition of Allocation Unit (AU)

4.13.2 Recording Unit (RU)

Each AU is divided into units called **"Recording Unit (RU)"** (Refer to Figure 4-27). The **RU Size (S_{RU})** should be one or multiple of cluster size specified in the SD File System Specification. **The number of RUs in an AU (N_{RU})** is calculated from S_{AU}/S_{RU}.

4.13.3 Write Performance

Figure 4-28 shows typical data management of the card when the host writes RUs of an AU. When the host writes to a fragmented AU, the card would make copy of the AU. The location A is at the start of the AU boundary and location B is at the end of the AU boundary. From A to B, the host shall write data to free RUs contiguously and skip used RUs (shall not skip any free RU). The card may indicate busy to the host, so the host can wait, during the time the card controller is writing and moving data. The total write time from A to B can be calculated by summing up the write time of free RUs and the moving time of the used RUs. The number of used RUs (Nu) is available by counting it over one AU and number of free RUs is expressed by (N_{RU} – Nu).

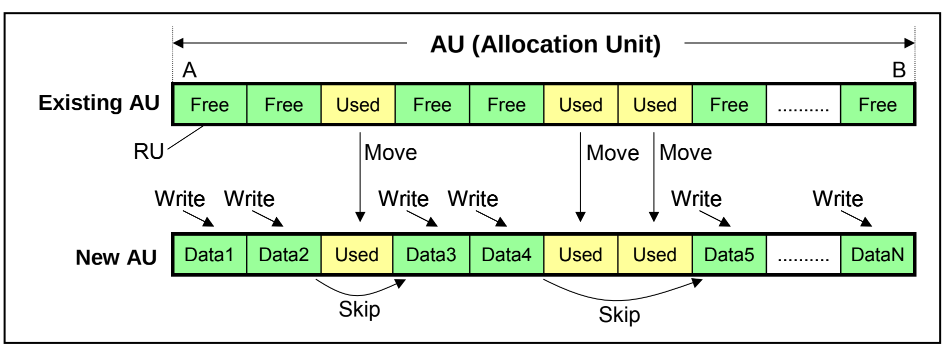


Figure 4-28: Example of Writing Fragmented AU

Average Performance of Fragmented AU can be calculated from dividing the amount of free RUs by the total execution time. It is expressed by using Performance Wite (Pw) and Performance Move (Pm).

Performance of Fragmented AU: $$P(Nu) = \frac{S_{RU}(N_{RU} - Nu)}{\frac{S_{RU}(N_{RU} - Nu)}{Pw} + \frac{S_{RU}Nu}{Pm}} = \frac{(N_{RU} - Nu)PmPw}{(N_{RU} - Nu)Pm + NuPw} \quad \text{..........................(1)}$$

The Performance Write (Pw) is defined as a minimum average write performance over an AU. It is calculated by taking the average over sequential RU write operations to one complete AU, which is not fragmented.

The Performance Move (Pm) is defined as a minimum average move performance. It is calculated by taking the average over sequential RU move operations to one complete AU. A move is an internal operation of the card, so SD clock frequency does not affect the time of the move operation. In case the card does not have to move RU, Pm should be considered as infinity (1/Pm = 0). Refer to Table 4-49 for the values defined for each Speed Class.

Application Notes: Performance may increase when larger data is written by one multiple write command. Therefore, the host may use larger RU sizes and transfer multiple RUs with one multiple-write command.

4.13.4 Read Performance

Two kinds of read performances are defined. Either type of read operation is possible to insert during write operations. Even if read operations, any read address and size, are performed, this performance specification shall be guaranteed.

(1) Read Performance of Stream Data
This is simply called **Read Performance (Pr). Pr** is defined as minimum average random RU read performance. Average is calculated over 256 random single RU read operations. Each RU is read by a multiple-read command. Pr shall be greater than or equal to Pw.

(2) FAT and Directory Entry Read Time
T_{FR}(4KB) is defined as the maximum time to read 4KB FAT and Directory Entry. The **FAT and Directory Entry Read Time** (**S_{FR} [KB]**) is defined using CEIL function:

FAT Read Time of S_{FR} [KB]: $$T_{FR}(S_{FR}) = \left\lceil \frac{S_{FR}}{4KB} \right\rceil \cdot T_{FR}(4KB) \quad \text{..(2)}$$

(: CEIL function. Convert decimal fraction x to the smallest integer more than or equal to x.)

Refer to Table 4-49 for the values defined for each Speed Class.

4.13.5 Performance Curve Definition

Figure 4-29 shows the write performances bar chart of P(Nu) of equitation (1). An AU consists of 16 RUs in this example. Joining the points of each bar shows the performance curve, which can be determined from the two parameters, Pw and Pm.

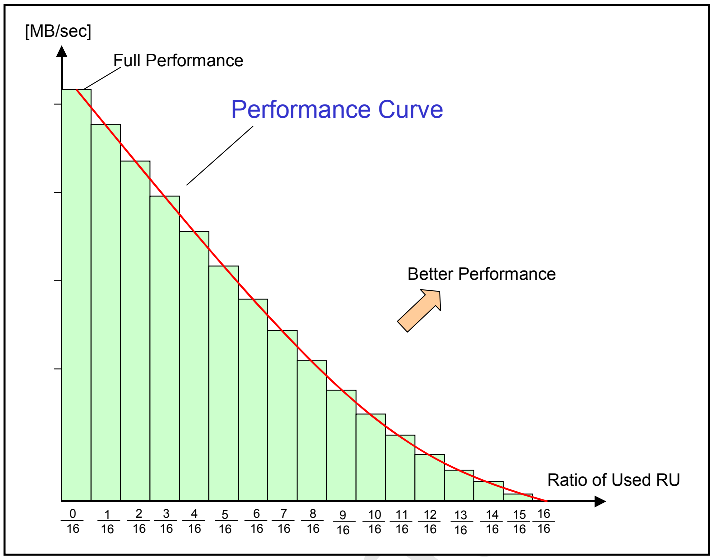


Figure 4-29: Card Performances between 16 RUs

The ratio of used RU (r) is defined as:

$$r = \frac{Nu}{N_{RU}}, \qquad Nu = rN_{RU}$$

The range of r is 0 to 1. (1 – r) means ratio of free RU, r=0 means all RUs are free. r=1 means all RUs are used and performance indicates zero at this point. By using r, Equation (1) is transformed into Equation (3).

Performance Curve: $P(r) = \frac{(1-r)PwPm}{rPw + (1-r)Pm} \quad (0 \le r \le 1)$ ……….................................…(3)

P(Nu) in Equation (1) is a discrete function but P(r) is treated as a continuous function.

4.13.6 Speed Class Definition

Figure 4-30 shows three performance curves. Pw indicates the performance of r=0 and Pm determines the shape of the curve. All performance curves converge at the point (1, 0). Therefore, there is little difference in performance where r is near to 1. These three curves divide the performance into four speed classes: Class 0, Class 2, Class4 and Class 6. The **Class 0** card provides no guarantee to be compliant to the Speed Class Specification. It does not report performance parameters even if the cards can achieve performance of higher speed classes. Class 0 also covers all legacy SD products prior to the introduction of this specification. The Classes are defined so that AV application, such as MPEG2 recording, can support SD card device. The performance of a Speed **Class 2** card shall be higher than performance curve 2. It is defined for standard TV image quality; approximately 2MB/sec performance will be required. The performance of a Speed **Class 4** card shall be higher than performance curve 4. Speed Class 4 is defined for HD video quality; approximately 4MB/sec performance will be required. Upper classes can be added in the future, if required. It is important that the host shall always accept cards which meet minimum speed class performance.

Speed Class shall be defined as SD Bus interface level performance, though the performance curve is derived from only back-end performance analysis in 4.13.3. SD clock frequency and RU size are defined as measurement conditions for Speed Classes. Refer to Section 4.13.8.

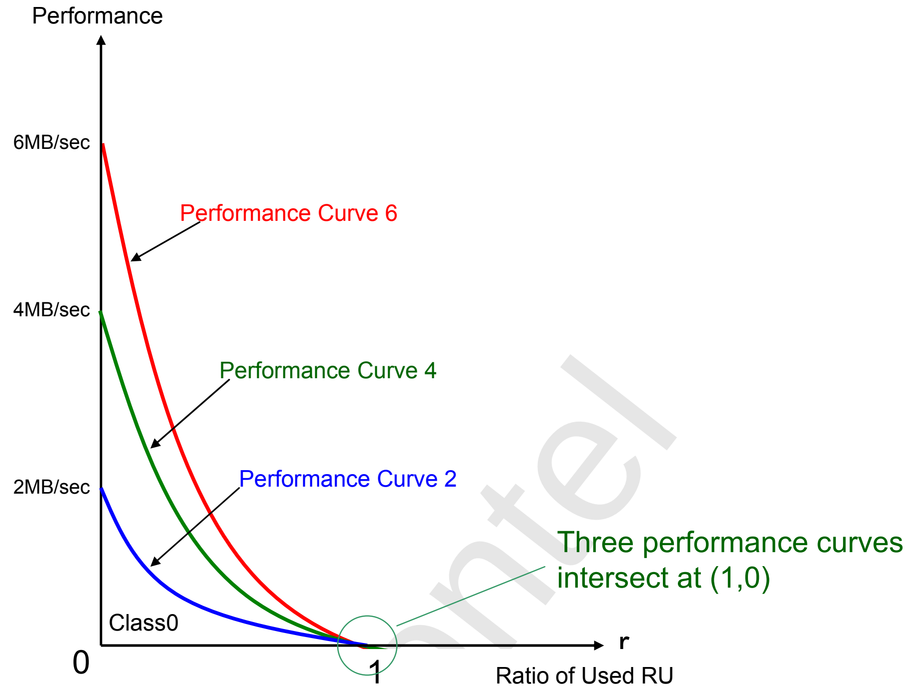


Figure 4-30: Three Performance Curves

Application Note: For the convenience of legacy card users, the host should try to use the card that has less performance than expected and attempt to record if necessary. When a mode provides operation only for specific Speed Class cards, one of the other modes should provide operation for lower Speed Class cards including Class 0.

4.13.7 Consideration for Inserting FAT Update during Recording

Figure 4-31 shows the typical sequence of the FAT update cycle for real time recording. FAT updates can be inserted between any RU accesses. The FAT update cycle consists of 3 FAT writes. The FAT1 and FAT2 means two FAT-table writes using one multiple write command for each FAT-table. The FAT table write can start any 512-byte boundary address and any size up to 16 KBytes. Only modified parts of FAT should be written. The DIR stands for directory entry write. A directory entry should be created before recording starts and only the modified parts should be written in the directory entry (512 Bytes). The **FAT Write Time (T_{FW})** is defined as the total time of 3 FAT writes sequences. The host shall take the sequence to be able to calculate degradation of performance by inserting a FAT update cycle. The card requires higher Card Performance (Pc) than Application Performance (Pa) to insert FAT update cycle.

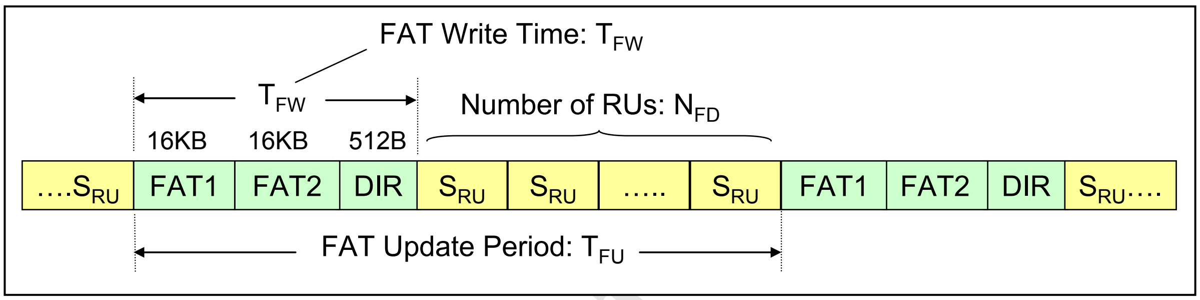


Figure 4-31: Typical Sequence of FAT Update

4.13.7.1 Measurement Condition to determine Average T_{FW}

The equation (4) defines **Average FAT Write Time (T_{FW}(ave.))**, which is the maximum sliding average of 8 times FAT write cycles.

Average FAT Write Time: $$T_{FW}(ave.) = \frac{\max(\sum_{i=1}^{8} T_{FW}(i))}{8}$$ ……………………........................(4)

4.13.7.2 Maximum FAT Write Time

During a FAT update, the host cannot write data to the card. Therefore, the host should prepare enough buffers to save the data temporarily. The **Maximum FAT Write Time (T_{FW} (max.))** is one of the factors to determine host buffer size. During 8 times FAT write cycle, occurrence of T_{FW} (max.) should not appear more than once. On the method of Host Buffer Size estimation, refer to the Implementation Guideline of the Speed Class Specification.

Maximum FAT Write Time: $$T_{FW}(\max.) \leq 750ms$$ …...........………………...(5)

4.13.8 Measurement Conditions and Requirements of the Speed Class

4.13.8.1 Measurement Conditions

Table 4-48 shows measurement conditions for each Speed Class. f_{SD} is the minimum SD clock frequency and S_{RU} is the minimum RU size to meet class performance. The Speed Class 6 card shall satisfy two measurement conditions and support high-speed mode. The higher Speed Class may require higher frequency or larger RU size. These values provide margin for host applications running at maximum speed.

	Minimum SD Clock Frequency f_{SD}	RU Size S_{RU}
Class 2	20 [MHz]	16 [KByte]
Class 4	20 [MHz]	16 [KByte]
Class 6	40 [MHz]	16 [KByte]
	20 [MHz]	64 [KByte]

Table 4-48: Measurement Condition of Speed Class

Application Note:
The host can choose one of the conditions to use a Class 6 card. One is that 16 KB RU size access requires supporting high-speed mode over 40 MHz. The other is that 64 KB RU size access can operate in normal-speed mode over 20 MHz.
The minimum performance is measured at 100% usage rate of the SD Bus (No idle time is assumed in accesses). Therefore, writing to the card at intervals decreases performance.

4.13.8.2 Requirements of the Performance Parameters for Each Speed Class

Table 4-49 identifies the requirement of the parameters for each class under measurement conditions. All conditions of any Class should apply simultaneously. Any cards having a specific Speed Class shall also satisfy the requirements and conditions of lower Classes. For example, Class 6 card shall satisfy Class 4 performance under Class 4 condition.

	Pw min.	Pm min.	Pr min.	T_{FW} (ave.) max.	T_{FW} (max.) max.	T_{FR} (4KB) max.
Class 2	2 [MB/sec]	1 [MB/sec]	2 [MB/sec]	100 [ms]	750 [ms]	4 [ms]
Class 4	4 [MB/sec]	2 [MB/sec]	4 [MB/sec]	100 [ms]	750 [ms]	4 [ms]
Class 6	6 [MB/sec]	3 [MB/sec]	6 [MB/sec]	100 [ms]	750 [ms]	4 [ms]

Table 4-49: Performance Requirements for Each Class

4.13.8.3 Requirements of SD File System

This specification can be applied only to the SD file system formatted card defined by the SD File System Specification Version 2.00. Furthermore, the Number of Hidden Sectors shall be adopted as minimum number that meets Boundary Unit Recommendation for Data Area.

4.14 Erase Timeout Calculation

This chapter provides the guideline for long erase and a method to calculate erase timeout value.

4.14.1 Erase Unit

The Speed Class Specification defines a new management unit of AU (Allocation Unit). Erase timeout calculation is defined as the basis of AU. SD memory card supports block erase but it takes more time to erase blocks, which are part of AU (partial erase AU). In this case, the host should add 250 ms to the result of timeout calculated on AU basis. When the start and end blocks are in the same partially erase AU, 500ms should be added.

4.14.2 Case Analysis of Erase Time Characteristics

Figure 4-32 shows an example of erase characteristics, number of AU erased versus erase time. Erase time is derived from erasing specified numbers of AUs by one erase command. Assuming that Erase is performed on AU basis and its erase characteristics can be approximated to a linear line. The line A illustrated in Figure 4-32 is an example characteristic.

The red line indicates the erase timeout value the host should use. The timeout value can be determined by line A. If the erase timeout is less than 1 second the host should use 1 second as timeout. If the timeout is bigger than 1 second the host should use the value determined by Line A.

Register parameters N_{ERASE}, T_{ERASE} and T_{OFFSET} define the shape of the line. T_{ERASE} indicates timeout for erasing N_{ERASE} AUs from T_{OFFSET}. T_{ERASE} and N_{ERASE} determine the slope of the line. T_{OFFSET} adjusts the line by moving in parallel on the upper side. The card manufacturer shall determine these parameters so that the line is always greater than the erase time of any AUs.

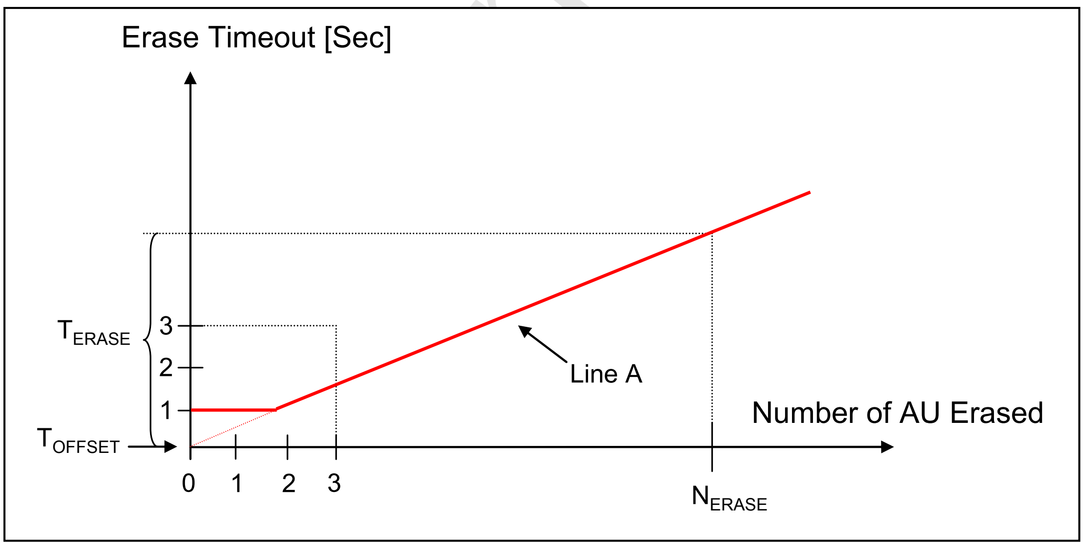


Figure 4-32: Example Erase Characteristics (Case 1 T_{OFFSET}=0)

The line B illustrated in Figure 4-33 shows another example of erase characteristics. The red line indicates the erase timeout value that the host should use. Since the time-out is bigger than 1 second, the red line and line B are equivalent.

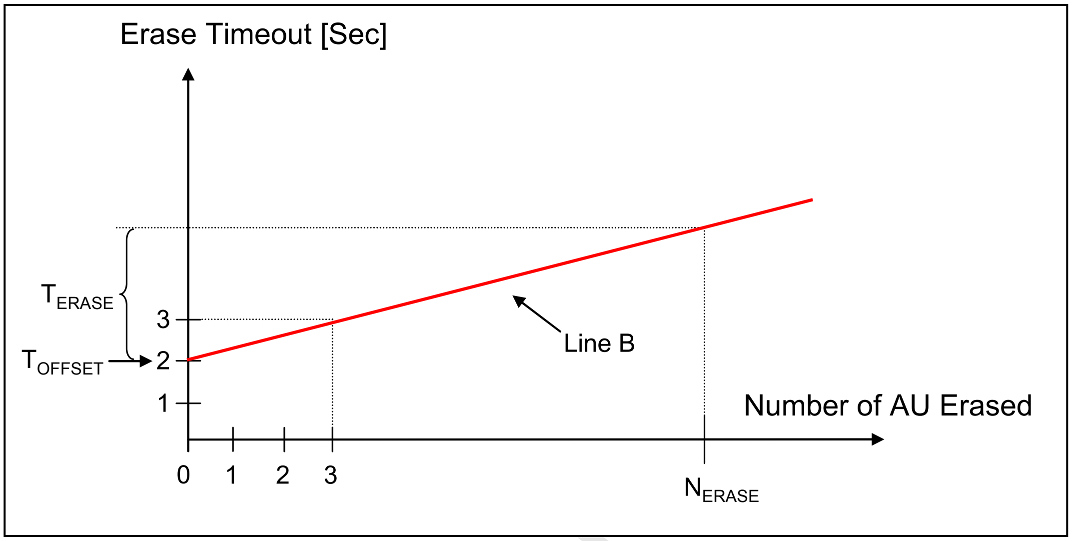


Figure 4-33: Example Erase Characteristics (Case 2 T_{OFFSET}=2)

4.14.3 Method for Erase Large Areas

The calculated erase timeout for multiple AUs might be too large compared with the actual erase time. The calculation of erase timeout is not accurate because calculated timeout includes a margin. A margin per AU accumulates and the result of calculating the timeout for large number of AUs will include large margins. Such calculations would be meaningless because the range of margin might be in order of minutes. Therefore, a small number of AUs should be erased at one time. This enables the host to calculate smaller timeout with fewer errors.

> Application Note:
> When a large area is erased, the host should divide it into small areas at the AU boundary and continuously erase the small areas using a small area erase timeout. It may take a long time to erase a large area, so the host should inform the user about the erase progress, otherwise the user might abort the execution of the erase.

4.14.4 Calculation of Erase Timeout Value Using the Parameter Registers

Erase Timeout of X AU can be calculated by Equation (6).

$$\text{Erase Time-out of X AU} = \frac{T_{ERASE}}{N_{ERASE}} \cdot X + T_{OFFSET} \quad \ldots\ldots(6)$$

Erase timeout is determined by following steps:

(1) Calculate Equation (6).
(2) If the result of (1) is less than 1 second, the timeout is set to 1 second.
(3) 250 ms should be added to the result of (2) for each partial erase AU. When the start and end blocks are in partially erase AUs, add 500 ms to the result of (2).

5. Card Registers

Six registers are defined within the card interface: OCR, CID, CSD, RCA, DSR and SCR. These can be accessed only by corresponding commands (see Chapter 4.7). The OCR, CID, CSD and SCR registers carry the card/content specific information, while the RCA and DSR registers are configuration registers storing actual configuration parameters.
In order to enable future extension, the card shall return 0 in the reserved bits of the registers.

5.1 OCR register

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. Additionally, this register includes status information bits. One status bit is set if the card power up procedure has been finished. This register includes another status bit indicating the card capacity status after set power up status bit. The OCR register shall be implemented by the cards.

The 32-bit operation conditions register stores the V_{DD} voltage profile of the card. Bit 7 of OCR is newly defined for Dual Voltage Card and set to 0 in default. If a Dual Voltage Card does not receive CMD8, OCR bit 7 in the response indicates 0, and the Dual Voltage Card which received CMD8, sets this bit to 1.

Additionally, this register includes 2 more status information bits.

Bit 31 - Card power up status bit, this status bit is set if the card power up procedure has been finished.

Bit 30 - Card capacity status bit, this status bit is set to 1 if card is High Capacity SD Memory Card. 0 indicates that the card is Standard Capacity SD Memory Card. The Card Capacity status bit is valid after the card power up procedure is completed and the card power up status bit is set to 1. The Host shall read this status bit to identify a Standard or High Capacity SD Memory Card.

The OCR register shall be implemented by the cards.

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

VDD Voltage Window (bits 0-23)

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

Table 5-1: OCR Register Definition

The supported voltage range is coded as shown in Table 5-1. A voltage range is not supported if the corresponding bit value is set to LOW. As long as the card is busy, the corresponding bit (31) is set to LOW.

5.2 CID register

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

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

Table 5-2: The CID Fields

- **MID**

An 8-bit binary number that identifies the card manufacturer. The MID number is controlled, defined, and allocated to a SD Memory Card manufacturer by the SD-3C, LLC. This procedure is established to ensure uniqueness of the CID register.

- **OID**

A 2-character ASCII string that identifies the card OEM and/or the card contents (when used as a distribution media either on ROM or FLASH cards). The OID number is controlled, defined, and allocated to a SD Memory Card manufacturer by the SD-3C, LLC. This procedure is established to ensure uniqueness of the CID register.

Note: SD-3C, LLC licenses companies that wish to manufacture and/or sell SD Memory Cards, including but not limited to flash memory, ROM, OTP, RAM, and SDIO Combo Cards.
SD-3C, LLC is a limited liability company established by Matsushita Electric Industrial Co. Ltd., SanDisk Corporation and Toshiba Corporation.

- **PNM**

The product name is a string, 5-character ASCII string.

- **PRV**

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

- **PSN**

  The Serial Number is 32 bits of binary number.

- **MDT**

  The manufacturing date is composed of two hexadecimal digits, one is 8 bits representing the year(y) and the other is 4 bits representing the month (m).
  The “m” field [11:8] is the month code. 1 = January.
  The “y” field [19:12] is the year code. 0 = 2000.
  As an example, the binary value of the Date field for production date “April 2001” will be:
  00000001 0100.

- **CRC**

  CRC7 checksum (7 bits). This is the checksum of the CID contents computed as described in Chapter 4.5.

5.3 CSD Register

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

5.3.1 CSD_STRUCTURE

Field structures of the CSD register are different depend on the Physical Specification Version and Card Capacity.
The CSD_STRUCTURE field in the CSD register indicates its structure version.
Table 5-3 shows the version number of the related CSD structure.

CSD_STRUCTURE	CSD structure version	Valid for SD Memory Card Physical Specification Version/Card Capacity
0	CSD Version 1.0	Version 1.01-1.10 Version 2.00/Standard Capacity
1	CSD Version 2.0	Version 2.00/High Capacity
2-3	reserved	

Table 5-3: CSD Register Structure

5.3.2 CSD Register (CSD Version 1.0)

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

Table 5-4: The CSD Register Fields (CSD Version 1.0)

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.

- **TAAC**

Defines the asynchronous part of the data access time.

TAAC bit position	code
2:0	time unit 0=1ns, 1=10ns, 2=100ns, 3=1µs, 4=10µs, 5=100µs, 6=1ms, 7=10ms
6:3	time value 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0
7	reserved

Table 5-5: TAAC Access Time Definition

- **NSAC**

Defines the worst case for the clock-dependant factor of the data access time. The unit for NSAC is 100 clock cycles. Therefore, the maximal value for the clock-dependent part of the data access time is 25.5 k clock cycles.
The total access time N_{AC} as expressed in the Table 4-47 is the sum of TAAC and NSAC. It should be computed by the host for the actual clock rate. The read access time should be interpreted as a typical delay for the first data bit of a data block or stream.

- **TRAN_SPEED**

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

TRAN_SPEED bit	code
2:0	transfer rate unit 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4... 7=reserved
6:3	time value 0=reserved, 1=1.0, 2=1.2, 3=1.3, 4=1.5, 5=2.0, 6=2.5, 7=3.0, 8=3.5, 9=4.0, A=4.5, B=5.0, C=5.5, D=6.0, E=7.0, F=8.0
7	reserved

Table 5-6: Maximum Data Transfer Rate Definition

Note that for current SD Memory Cards, this field shall be always 0_0110_010b (032h) which is equal to 25 MHz - the mandatory maximum operating frequency of SD Memory Card.
In High-Speed mode, this field shall be always 0_1011_010b (05Ah) which is equal to 50 MHz, and when the timing mode returns to the default by CMD6 or CMD0 command, its value will be 032h.

- **CCC**

The SD Memory Card command set is divided into subsets (command classes). The card command class register CCC defines which command classes are supported by this card. A value of 1 in a CCC bit means that the corresponding command class is supported. For command class definitions, refer to Table 4-18.

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

Table 5-7: Supported Card Command Classes

- **READ_BL_LEN**

The maximum read data block length is computed as $2^{READ\_BL\_LEN}$. The maximum block length might therefore be in the range 512...2048 bytes (see Chapter 4.11 for details). Note that in an SD Memory Card the WRITE_BL_LEN is always equal to READ_BL_LEN

READ_BL_LEN	Block length
0-8	reserved
9	2^9 = 512 Bytes
10	2^{10} = 1024 Bytes
11	2^{11} = 2048 Bytes
12-15	reserved

Table 5-8: Data Block Length

- **READ_BL_PARTIAL (always = 1 in SD Memory Card)**

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

- **WRITE_BLK_MISALIGN**

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

- **READ_BLK_MISALIGN**

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

- **DSR_IMP**

Defines if the configurable driver stage is integrated on the card. If set, a driver stage register (DSR) shall be implemented (also see Chapter 5.5).

DSR_IMP	DSR type
0	no DSR implemented
1	DSR implemented

Table 5-9: DSR Implementation Code Table

- **C_SIZE**

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

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)

To indicate 2 GByte card, BLOCK_LEN shall be 1024 bytes.
Therefore, the maximal capacity that can be coded is 4096*512*1024 = 2 G bytes.
Example: A 32 Mbyte card with BLOCK_LEN = 512 can be coded by C_SIZE_MULT = 3 and C_SIZE = 2000.

- **VDD_R_CURR_MIN, VDD_W_CURR_MIN**

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

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

Table 5-10: $V_{DD, min}$ Current Consumption

- **VDD_R_CURR_MAX, VDD_W_CURR_MAX**

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

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

Table 5-11: $V_{DD, max}$ Current Consumption

- **C_SIZE_MULT**

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

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

Table 5-12: Multiply Factor for the Device Size

- **ERASE_BLK_EN**

The ERASE_BLK_EN defines the granularity of the unit size of the data to be erased. The erase operation can erase either one or multiple units of 512 bytes or one or multiple units (or sectors) of SECTOR_SIZE (see definition below).

If ERASE_BLK_EN=0, the host can erase one or multiple units of SECTOR_SIZE. The erase will start from the beginning of the sector that contains the start address to the end of the sector that contains the end address. For example, if SECTOR_SIZE=31 and the host sets the Erase Start Address to 5 and the Erase End Address to 40, the physical blocks from 0 to 63 will be erased as shown in Figure 5-1.

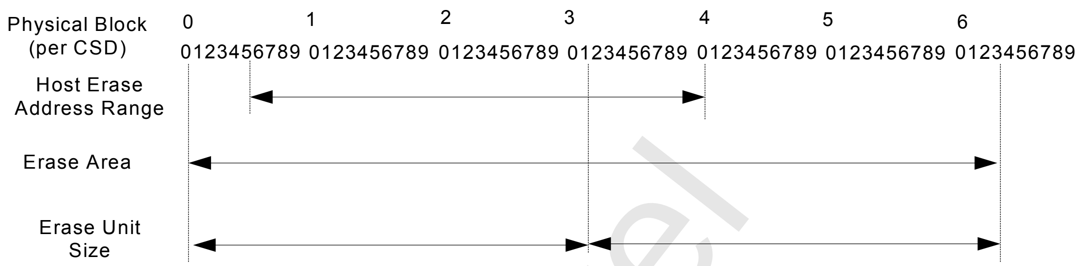


Figure 5-1: ERASE_BLK_EN = 0 Example

If ERASE_BLK_EN=1 the host can erase one or multiple units of 512 bytes. All blocks that contain data from start address to end address are erased. For example, if the host sets the Erase Start Address to 5 and the Erase End Address to 40, the physical blocks from 5 to 40 will be erased as shown in Figure 5-2.

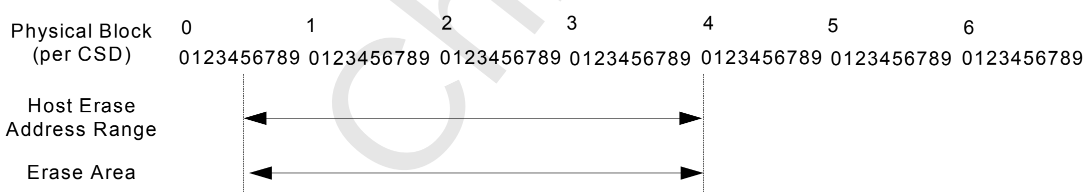


Figure 5-2: ERASE_BLK_EN = 1 Example

- **SECTOR_SIZE**

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

- **WP_GRP_SIZE**

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

- **WP_GRP_ENABLE**

A value of 0 means no group write protection possible.

- **R2W_FACTOR**

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

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

Table 5-13: R2W_FACTOR

- **WRITE_BL_LEN**

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

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

Table 5-14: Data Block Length

- **WRITE_BL_PARTIAL**

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

- **FILE_FORMAT_GRP**

Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in Table 5-15 (see FILE_FORMAT).

- **COPY**

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

- **PERM_WRITE_PROTECT**

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

- **TMP_WRITE_PROTECT**

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

- **FILE_FORMAT**

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

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

Table 5-15: File Formats

A more detailed description is given in the SD Memory Card File System specification.

- **CRC**

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

5.3.3 CSD Register (CSD Version 2.0)

Table 5-16 shows Definition of the CSD for the High Capacity SD Memory Card (CSD Version 2.0). The following sections describe the CSD fields and the relevant data types for the High Capacity SD Memory Card.

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

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

Table 5-16: The CSD Register Fields (CSD Version 2.0)

- **TAAC**

  This field is fixed to 0Eh, which indicates 1 ms. The host should not use TAAC, NSAC, and R2W_FACTOR to calculate timeout and should uses fixed timeout values for read and write operations (See 4.6.2).

- **NSAC**

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

- **TRAN_SPEED**

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

- **CCC**

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

- **READ_BL_LEN**

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

- **READ_BL_PARTIAL**

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

- **WRITE_BLK_MISALIGN**

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

- **READ_BLK_MISALIGN**

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

- **DSR_IMP**

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

- **C_SIZE**

  This field is expanded to 22 bits and can indicate up to 2 TBytes (It is the same as the maximum memory space specified by a 32-bit block address.)

  This parameter is used to calculate the user data area capacity in the SD memory card (not include the protected area). The user data area capacity is calculated from C_SIZE as follows:

  memory capacity = (C_SIZE+1) * 512K byte

  As the maximum capacity of the Physical Layer Specification Version 2.00 is 32 GB, the upper 6 bits of this field shall be set to 0.

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

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

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

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

- **R2W_FACTOR**
  This field is fixed to 2h, which indicates 4 multiples. Write timeout can be calculated by multiplying the read access time and R2W_FACTOR. However, the host should not use this factor and should use 250 ms for write timeout (See 4.6.2).

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

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

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

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

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

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

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

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

5.4 RCA register

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

5.5 DSR register (Optional)

The 16-bit driver stage register is described in detail in Chapter 6.5. It can be optionally used to improve the bus performance for extended operating conditions (depending on parameters like bus length, transfer rate or number of cards). The CSD register carries the information about the DSR register usage. The default value of the DSR register is 0x404.

5.6 SCR register

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

The following table describes the SCR register content.

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

Table 5-17: The SCR Fields

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

Table 5-18: SCR Register Structure Version

- **SD_SPEC**

Describes the Physical Layer Specification Version supported by the card.

SD_SPEC	**Physical Layer Specification Version Number**
0	Version 1.0-1.01
1	Version 1.10
2	Version 2.00
3-15	reserved

Table 5-19: Physical Layer Specification Version

- **DATA_STAT_AFTER_ERASE**

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

- **SD_SECURITY**

Describes the Security Specification Version supported by the card.

SD_SECURITY	Security Specification Version
0	no security
1	Not used
2	Version 1.01
3	Version 2.00
4 .. 7	reserved

Table 5-20: SD Supported Security Algorithm

Note that it is mandatory for a regular writable SD Memory Card to support Security Protocol. For ROM (Read Only) and OTP (One Time Programmable) types of the SD Memory Card, the security feature is optional. In the case of Standard Capacity SD Memory Card, this field shall be set to 2 (Version 1.01). In the case of High Capacity SD Memory Card, this field shall be set to 3 (Version 2.00).

- **SD_BUS_WIDTHS**

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

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

Table 5-21: SD Memory Card Supported Bus Widths

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

6. SD Memory Card Hardware Interface

The SD Memory Card has six communication lines and three supply lines:

- CMD: Command is a bidirectional signal. The host and card drivers operate in push pull mode.
- DAT0-3: Data lines are bidirectional signals. Host and card drivers operate in push pull mode
- CLK: Clock is a host to card signal. CLK operates in push pull mode
- V_{DD}: V_{DD} is the power supply line for all cards.
- V_{SS1}, V_{SS2} are two ground lines.

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

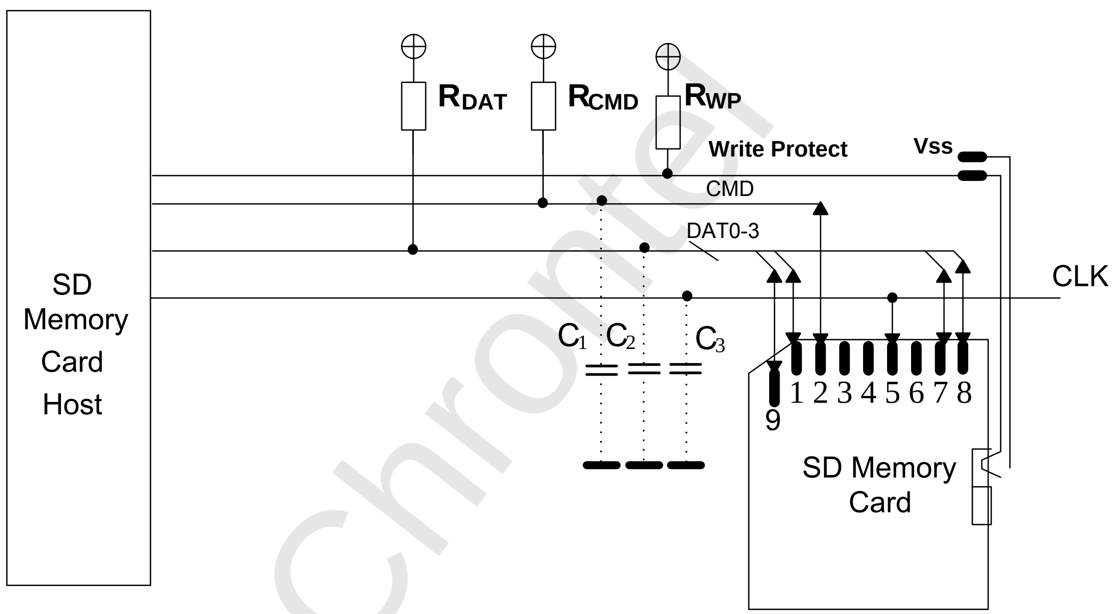


Figure 6-1: Bus Circuitry Diagram

R_{DAT} and R_{CMD} are pull-up resistors protecting the CMD and the DAT lines against bus floating when no card is inserted or when all card drivers are in a high-impedance mode.
The host shall pull-up all DAT0-3 lines by RDAT, even if the host uses the SD Memory Card as 1 bit-mode-only in SD mode. Also, the host shall pull-up all "RSV" lines in SPI mode, even though they are not used.
R_{WP} is used for the Write Protect/Card Detection switch.
Refer to Chapter 6.6 for components values and conditions.

6.1 Hot Insertion and Removal

To guarantee the proper sequence of card pin connection during hot insertion, the use of either a special hot-insertion capable card connector or an auto-detect loop on the host side (or some similar mechanism) is mandatory (see Chapter 8).
No card shall be damaged by inserting or removing a card into the SD Memory Card bus even when the power (V_{DD}) is up. Data transfer operations are protected by CRC codes, therefore any bit changes induced by card insertion and removal can be detected by the SD Memory Card bus master.
The inserted card shall be properly reset also when CLK carries a clock frequency f_{PP}. Each card shall have power protection to prevent card (and host) damage. Data transfer failures induced by removal/insertion are detected by the bus master. They should be corrected by the application, which may repeat the issued command.

6.2 Card Detection (Insertion/Removal)

In order to be able to give feedback indication to the users, the SD Memory Card system shall implement detection of card insertion or removal. One method is by sensing pin 1 of the card, and detecting the pull-up resistance on it. Detailed description of this and several other card detection options is given in"Application Notes Relating to SD Physical Specification".

6.3 Power Protection (Insertion/Removal)

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

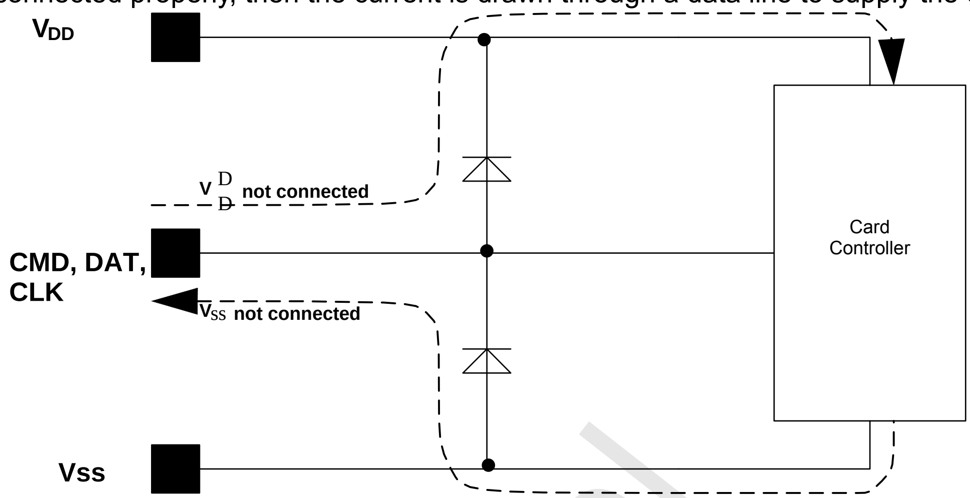


Figure 6-2: Improper Power Supply

Every card's output also shall be able to withstand short circuit to either supply.

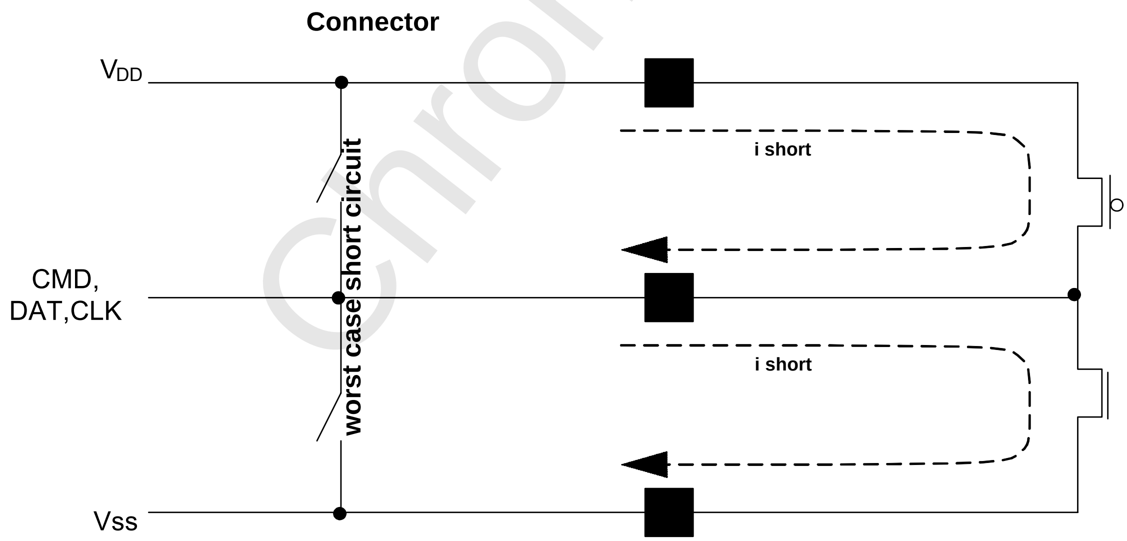


Figure 6-3: Short Circuit Protection

If hot insertion feature is implemented in the host, then the host has to withstand an instant short circuit between V_{DD} and V_{SS} without damage.

6.4 Power Scheme

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

6.4.1 Power Up

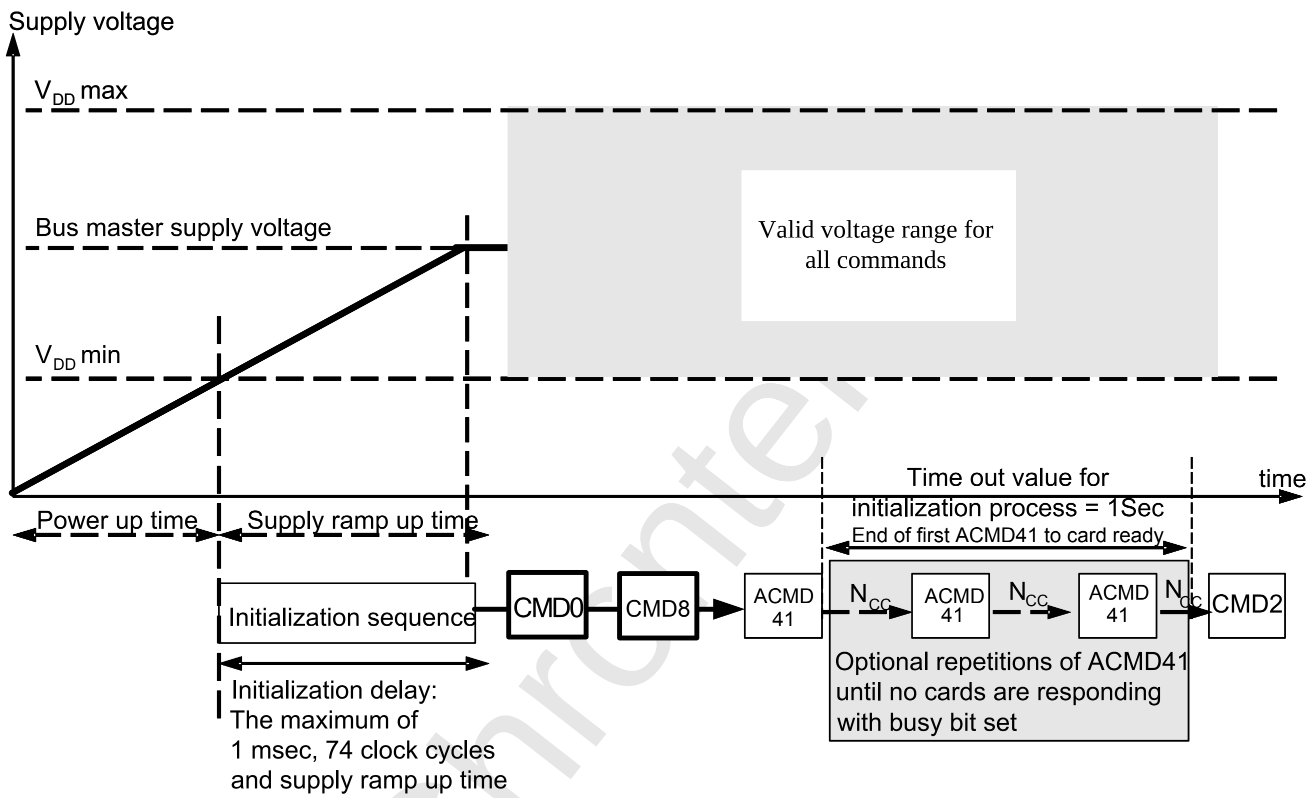


Figure 6-4: Power-up Diagram

- 'Power up time' is defined as voltage rising time from 0 volt to V_{DD} min (refer to 6.6) and depends on application parameters such as the maximum number of SD Cards, the bus length and the characteristic of the power supply unit.
- 'Supply ramp up time' provides the time that the power is built up to the operating level (the bus master supply voltage) and the time to wait until the SD card can accept the first command,
- The host shall supply power to the card so that the voltage is reached to Vdd_min within 250ms and start to supply at least 74 SD clocks to the SD card with keeping CMD line to high. In case of SPI mode, CS shall be held to high during 74 clock cycles.
- After power up (including hot insertion, i.e. inserting a card when the bus is operating) the SD Card enters the *idle state*. In case of SD host, CMD0 is not necessary. In case of SPI host, CMD0 shall be the first command to send the card to SPI mode.
- CMD8 is newly added in the Physical Layer Specification Version 2.00 to support multiple voltage ranges and used to check whether the card supports supplied voltage. The version 2.00 host shall issue CMD8 and verify voltage before card initialization. The host that does not support CMD8 shall supply high voltage range.
- ACMD41 is a synchronization command used to negotiate the operation voltage range and to poll the cards until they are out of their power-up sequence. In case the host system connects multiple cards, the host shall check that all cards satisfy the supplied voltage. Otherwise, the host should select one of the cards and initialize.

6.4.2 Power Down and Power Cycle

- When the host shuts down the power, the card V_{DD} shall be lowered to less than 0.5Volt for a minimum period of 1ms. During power down, DAT, CMD, and CLK should be disconnected or driven to logical 0 by the host to avoid a situation that the operating current is drawn through the signal lines.

- If the host needs to change the operating voltage, a power cycle is required. Power cycle means the power is turned off and supplied again. Power cycle is also needed for accessing cards that are already in *Inactive State.* To create a power cycle the host shall follow the power down description before power up the card (i.e. the card V_{DD} shall be once lowered to less than 0.5Volt for a minimum period of 1ms).

6.5 Programmable Card Output Driver (Optional)

The bus capacitance of each line of the SD bus is the sum of the bus master capacitance, the bus capacitance itself and the capacitance of each inserted card. The sum of host and bus capacitance is fixed for one application, but may vary between different applications. The card load may vary in one application with each of the inserted cards.

In the following, programmable card output drivers for the push pull mode are described as an optional method for ensuring the defined maximum clock rate independently of the topology and of the number of inserted cards.

Both data and command driver stages in the push-pull mode have programmable peak current driving capabilities and programmable rise and fall times. The driver stage register (DSR) consists of two 8-bit latches. The contents of the latches are calculated from the required transfer speed of the interface and the bus load.

The CMD and DAT bus drivers consist of a pre-driver stage and a complementary driver transistor (Figure 6-5). The pre-driver stage output rise and fall time is set with the DSR1 register and determines the speed of the driver stage. The complementary driver transistor size is set with the DSR2 register and determines the current driving capabilities of the driver stage and also influences the peak current consumption of the bus driver. The proper combination of both allows the optimum bus performance.

Table 6-1 defines the DSR register contents:

DSR1	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
$t_{switch\text{-}on\ max}$	reserved				5ns	20ns	100ns	500ns
$t_{switch\text{-}on\ min}$					2ns	10ns	50ns	200ns

DSR2	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
$i_{peak\ min}$	reserved				100mA	20mA	5mA	1mA
$i_{peak\ max}$					200mA	50mA	10mA	2mA
$t_{rise\ typ}$					5ns	20ns	100ns	500ns

Table 6-1: DSR Register Contents

The time in DSR1 specifies the switch-on time of the output driver transistors. At the external interface, it is measurable as a delay time between the clock and driver stage output signal (e.g. for testing).

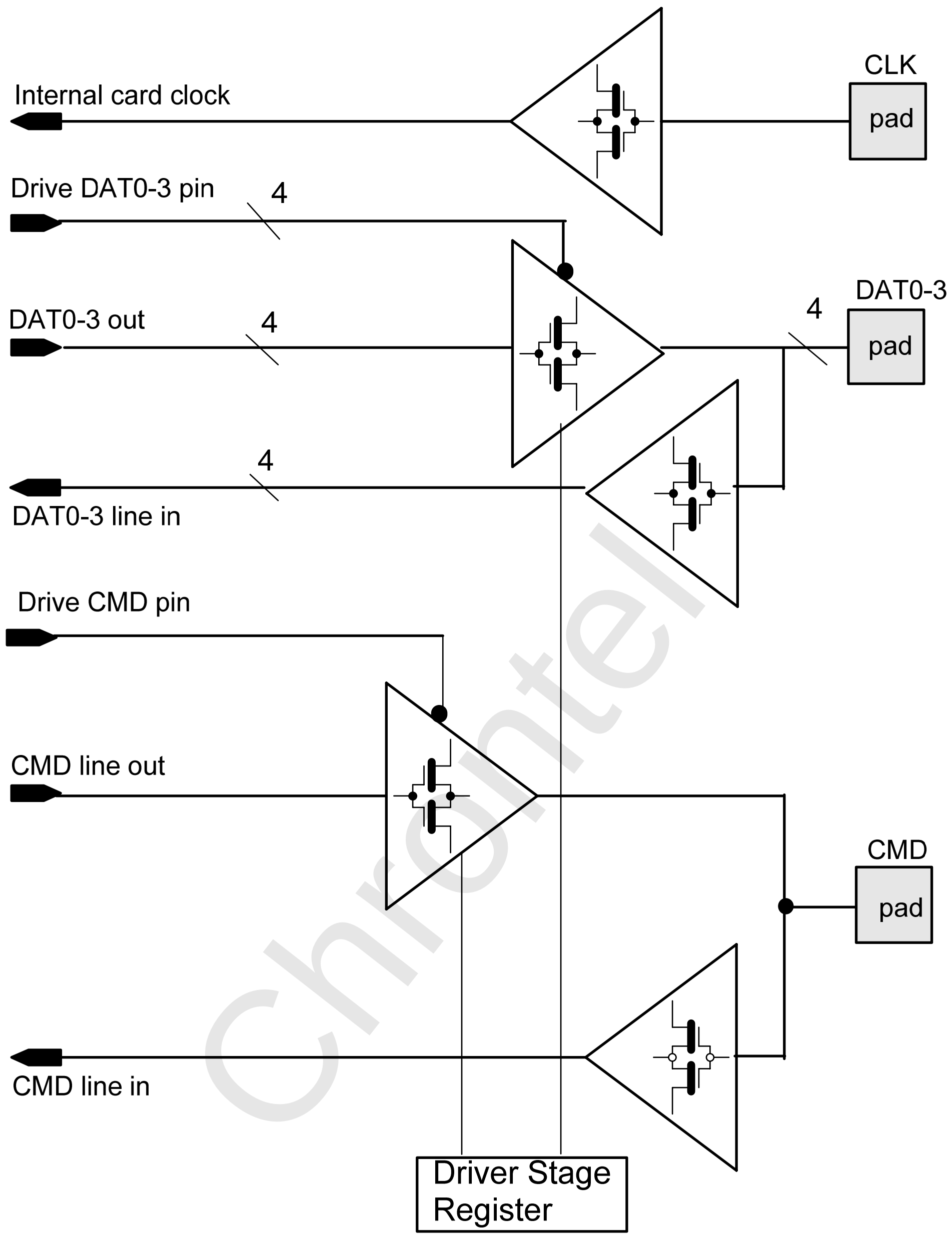


Figure 6-5: SD Memory Card Bus Driver

All data are valid for the specified operating range (voltage, temperature). Any combination of DSR1 and DSR2 bits may be programmed. DSR1 has to be programmed for the required clock frequency, where

$f_{clock} = (2\ t_{switch\text{-}on\ max})^{-1}$.

The DSR2 register shall be programmed with the required driver size. Hints for the proper driver stage selection are part of future application notes (see Appendix).

6.6 Bus Operating Conditions

6.6.1 Threshold Level for High Voltage Range

Parameter	Symbol	Min	Max	Unit	Remark
Supply Voltage	V_{DD}	2.7	3.6	V	
Output High Voltage	V_{OH}	0.75*V_{DD}		V	I_{OH}=-100uA $V_{DD\ min}$
Output Low Voltage	V_{OL}		0.125*V_{DD}	V	I_{OL} = 100uA $V_{DD\ min}$
Input High Voltage	V_{IH}	0.625*V_{DD}	V_{DD}+0.3	V	
Input Low Voltage	V_{IL}	V_{SS}-0.3	0.25 *V_{DD}	V	
Power Up Time			250	ms	From 0V to $V_{DD\ min}$

Table 6-2: Threshold Level for High Voltage

6.6.2 Threshold Level for Low Voltage Range

T.B.D.

6.6.3 General

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

Table 6-3: Bus Operating Conditions – General

6.6.4 Current Consumption

The current consumption is measured by averaging over 1 second.

- Before first command: Maximum 15 mA
- During initialization: Maximum 100 mA
- Operation in Default Mode: Maximum 100 mA
- Operation in High Speed Mode: Maximum 200 mA
- Operation with other functions: Maximum 500 mA.

6.6.5 Bus signal line load

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

Total bus capacitance = $C_{HOST} + C_{BUS} + N C_{CARD}$

Where N is the number of connected cards.

Parameter	Symbol	Min	Max.	Unit	Remark
Pull-up resistance	R_{CMD} R_{DAT}	10	100	kΩ	To prevent bus floating
Total bus capacitance for each signal line	C_L		40	pF	1 card $C_{HOST}+C_{BUS}$ shall not exceed 30 pF
Capacitance of the card for each signal pin	C_{CARD}		10	pF	
Maximum signal line inductance			16	nH	$f_{PP} \leq 20$ MHz
Pull-up resistance inside card (pin1)	R_{DAT3}	10	90	kΩ	May be used for card detection

Table 6-4: Bus Operating Conditions - Signal Line's Load

Note that the total capacitance of CMD and DAT lines will be consist of C_{HOST}, C_{BUS} and one C_{CARD} only because they are connected separately to the SD Memory Card host.

Host should consider total bus capacitance for each signal as the sum of C_{HOST}, C_{BUS}, and C_{CARD}, these parameters are defined by per signal. The host can determine C_{HOST} and C_{BUS} so that total bus capacitance is less than the card estimated capacitance load (C_L=40 pF). The SD Memory Card guarantees its bus timing when total bus capacitance is less than maximum value of C_L (40 pF).

6.6.6 Bus Signal Levels

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

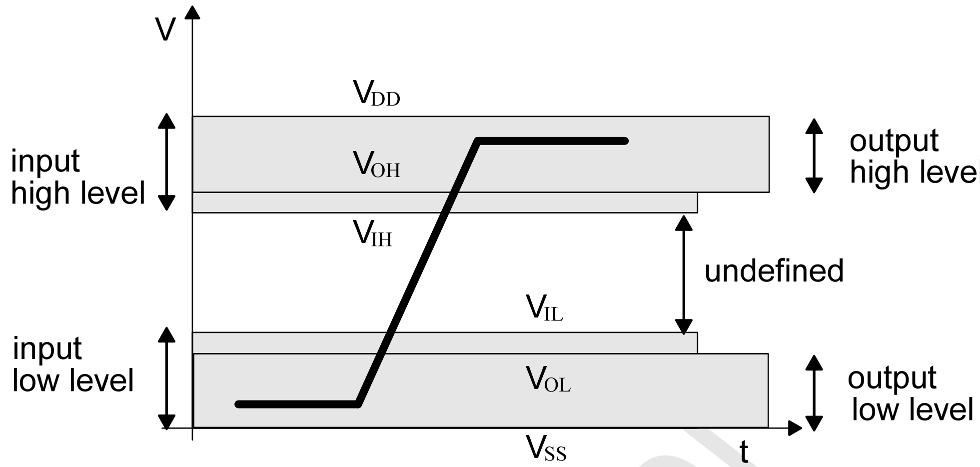


Figure 6-6: Bus Signal Levels

To meet the requirements of the JEDEC specification JESD8-1A and JESD8-7, the card input and output voltages shall be within the specified ranges shown in Table 6-2 for any V_{DD} of the allowed voltage range:

6.7 Bus Timing (Default)

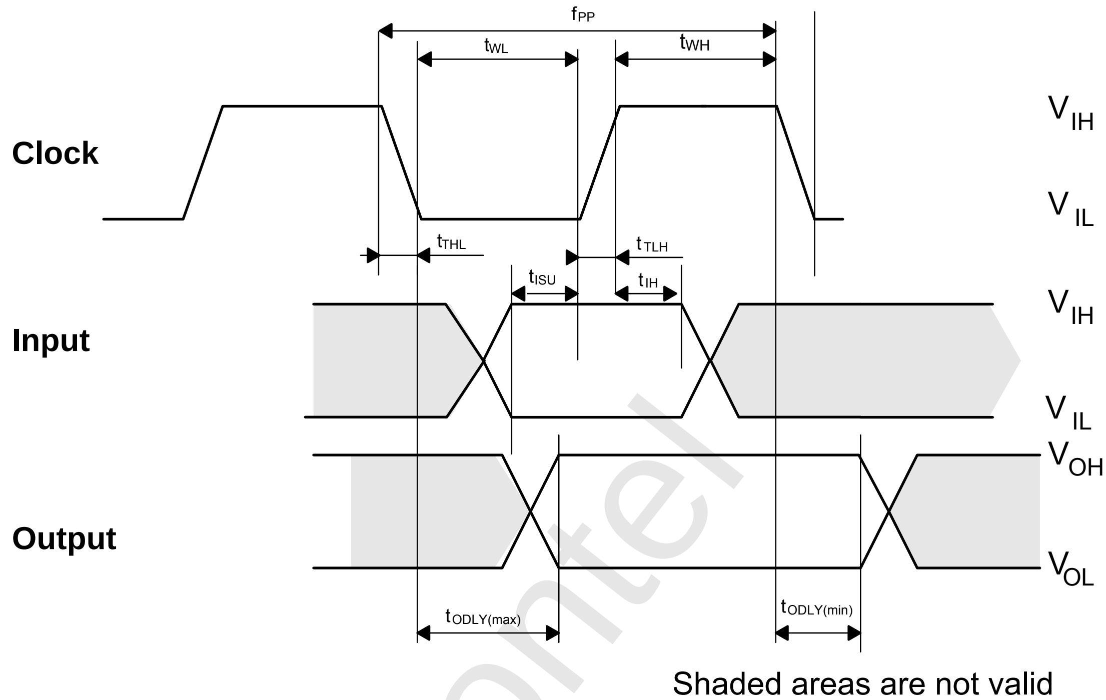


Figure 6-7: Timing Diagram Data Input/Output Referenced to Clock (Default)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min (V_{IH}) and max (V_{IL}),					
Clock frequency Data Transfer Mode	f_{PP}	**0**	**25**	**MHz**	**C_{CARD} ≤ 10 pF (1 card)**
Clock frequency Identification Mode	f_{OD}	**$0_{(1)}$/100**	**400**	**kHz**	**C_{CARD} ≤ 10 pF (1 card)**
Clock low time	t_{WL}	**10**		**ns**	**C_{CARD} ≤ 10 pF (1 card)**
Clock high time	t_{WH}	**10**		**ns**	**C_{CARD} ≤ 10 pF (1 card)**
Clock rise time	t_{TLH}		**10**	**ns**	**C_{CARD} ≤ 10 pF (1 card)**
Clock fall time	t_{THL}		**10**	**ns**	**C_{CARD} ≤ 10 pF (1 card)**
Inputs CMD, DAT (referenced to CLK)					

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

(1) 0 Hz means to stop the clock. The given minimum frequency range is for cases were continues clock is required (refer to Chapter 4.4- Clock Control).

Table 6-5: Bus Timing - Parameters Values (Default)

6.8 Bus Timing (High-Speed Mode)

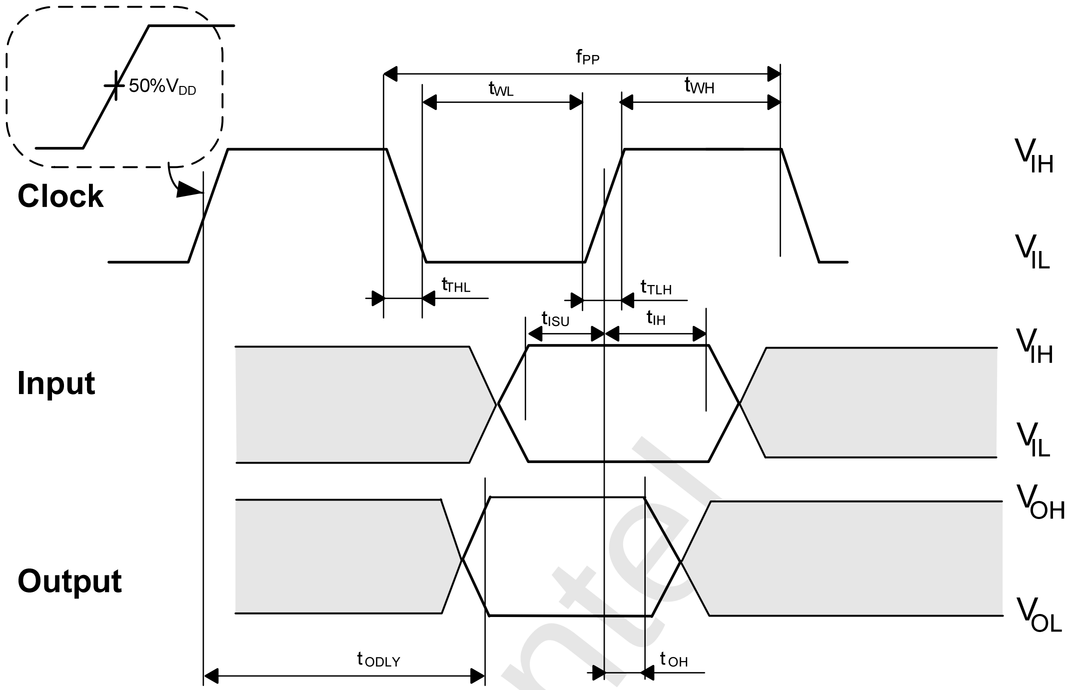


Figure 6-8: Timing Diagram Data Input/Output Referenced to Clock (High-Speed)

Parameter	Symbol	Min	Max.	Unit	Remark
Clock CLK (All values are referred to min (V_{IH}) and max (V_{IL}),					
Clock frequency Data Transfer Mode	f_{PP}	0	50	MHz	$C_{CARD} \leq$ 10 pF (1 card)
Clock low time	t_{WL}	7		ns	$C_{CARD} \leq$ 10 pF (1 card)
Clock high time	t_{WH}	7		ns	$C_{CARD} \leq$ 10 pF (1 card)
Clock rise time	t_{TLH}		3	ns	$C_{CARD} \leq$ 10 pF (1 card)
Clock fall time	t_{THL}		3	ns	$C_{CARD} \leq$ 10 pF (1 card)
Inputs CMD, DAT (referenced to CLK)					
Input set-up time	t_{ISU}	6		ns	$C_{CARD} \leq$ 10 pF (1 card)
Input hold time	t_{IH}	2		ns	$C_{CARD} \leq$ 10 pF

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

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

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

7. SPI Mode

7.1 Introduction

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

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

7.2 SPI Bus Protocol

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

The card starts to count SPI bus clock cycle at the assertion of the CS signal. Every command or data token shall be aligned to 8-clock cycle boundary.

Similar to the SD Memory Card protocol, the SPI messages consist of command, response and 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.

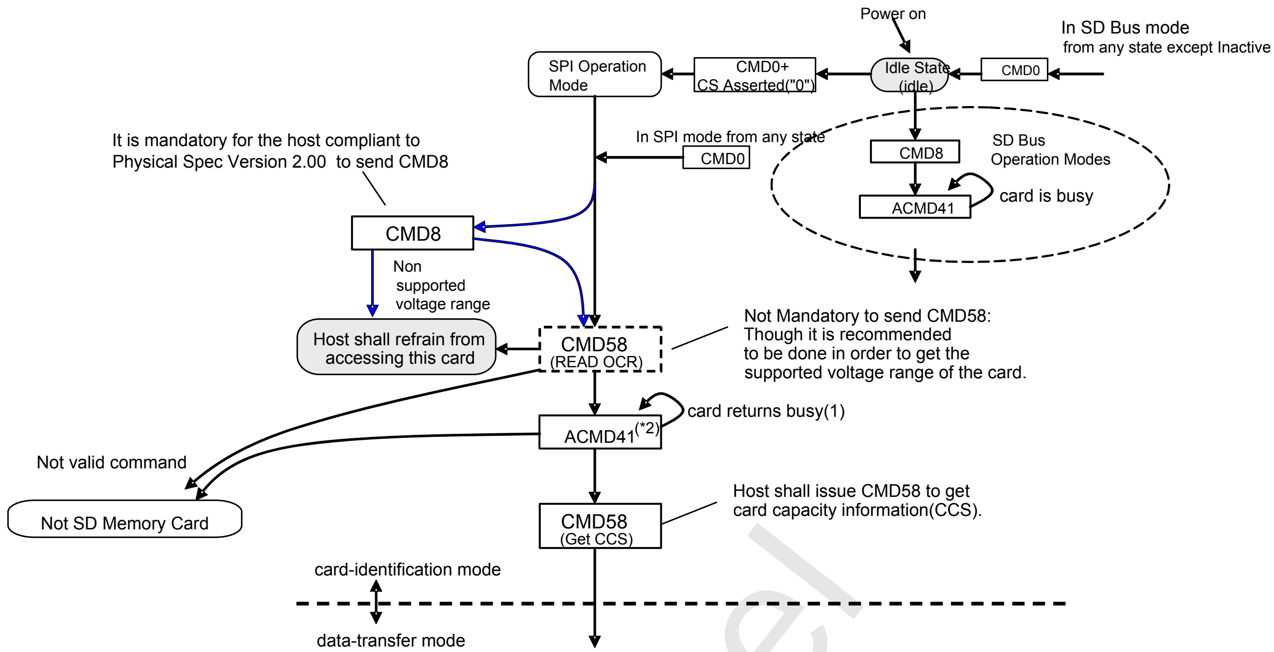
The selected card always responds to the command as opposed to the SD mode.

When the card encounters a data retrieval problem in a read operation, it will respond with an error response (which replaces the expected data block) rather than by a timeout as in the SD mode.

Additionally, every data block sent to the card during write operations will be responded with a data response token.

In the case of a Standard Capacity Memory Card, a data block can be as big as one card write block and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

In the case of a High Capacity SD Memory Card, the size of data block is fixed to 512 bytes. The block length set by CMD16 is only used for CMD42 and not used for memory data transfer. So, partial block read/write operations are also disabled. Furthermore, Write Protected commands (CMD28, CMD29 and CMD30) are not supported.



(*1) Note: Card returns busy when
- Card executes internal initialization process.
- If the card is High capacity SD Memory Card, there are 2 cases
  1. CMD8 was not issued before ACMD41
  2. ACMD41 is issued with HCS=0

(*2) Note: 2.1mm SD Memory Card can be initialized using CMD1 and Thin (1.4mm) SD Memory Card can be initialized using CMD1 only after firstly initialized by using CMD0 and ACMD41. In any of the cases CMD1 is not recommended because it may be difficult for the host to distinguish between MultiMediaCard and SD Memory Card.
If the SD card is initialized by CMD1 and the host treat it as MMC card, not SD card, the Data of the card may be damaged because of wrong interpretation of CSD and CID registers.

Figure 7-1: SD Memory Card State Diagram (SPI mode)

7.2.1 Mode Selection and Initialization

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

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

Figure 7-2 shows the initialization sequence of SPI mode.

SEND_IF_COND (CMD8) is used to verify SD Memory Card interface operating condition. The argument format of CMD8 is the same as defined in SD mode and the response format of CMD8 is defined in Section 7.3.2.6. The card checks the validity of operating condition by analyzing the argument of CMD8 and the host checks the validity by analyzing the response of CMD8. The supplied voltage is indicated by VHS filed in the argument. The card assumes the voltage specified in VHS as the current supplied voltage. Only 1-bit of VHS shall be set to 1 at any given time. Check pattern is used for the host to check validity of communication between the host and the card.

If the card indicates an illegal command, the card is legacy and does not support CMD8. If the card supports CMD8 and can operate on the supplied voltage, the response echoes back the supply voltage and the check pattern that were set in the command argument.

If VCA in the response is set to 0, the card cannot operate on the supplied voltage. If check pattern is not matched, CMD8 communication is not valid. In this case, it is recommended to retry CMD8 sequence.

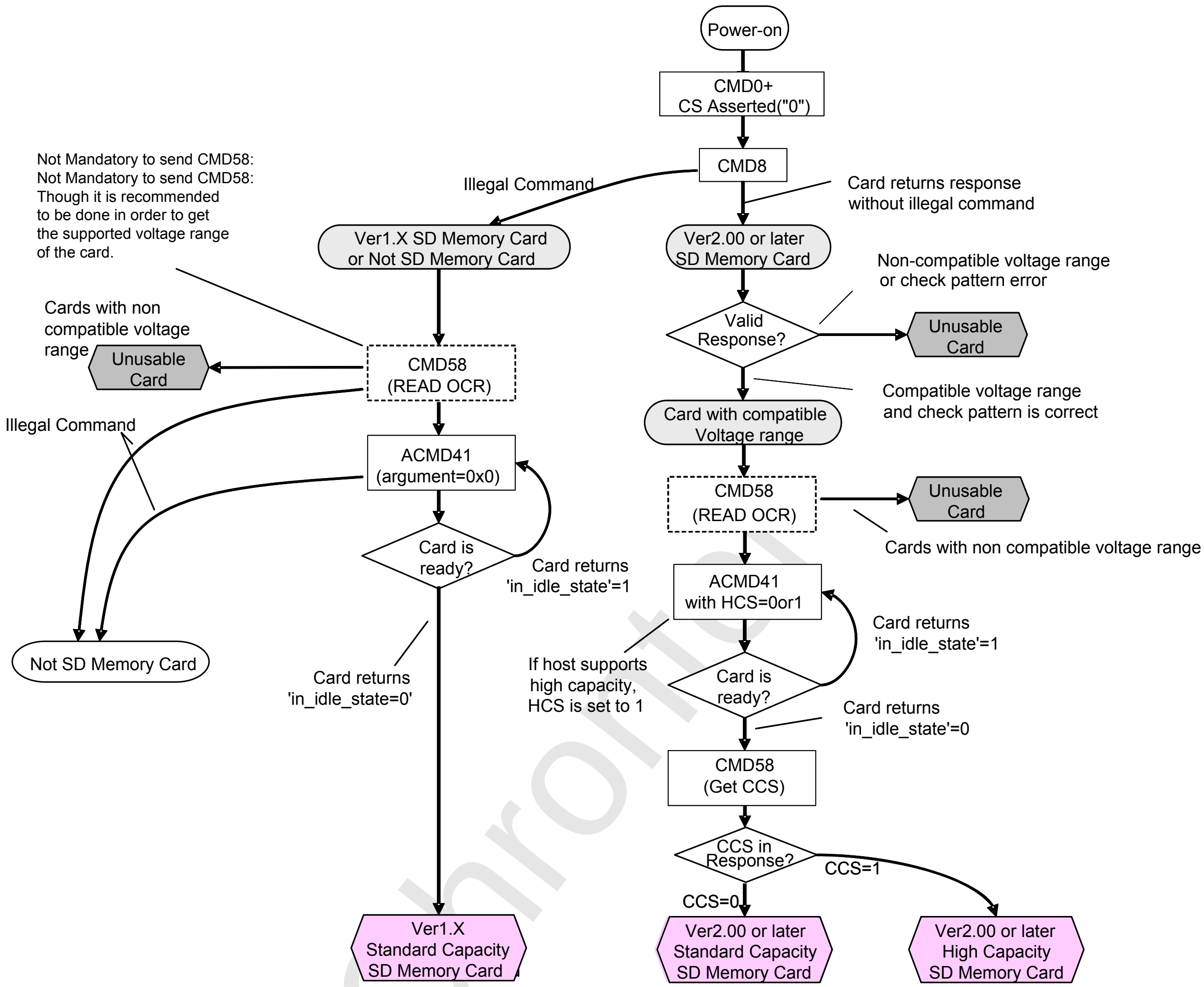


Figure 7-2: SPI Mode Initialization Flow

READ_OCR (CMD58) is designed to provide SD Memory Card hosts with a mechanism to identify cards that do not match the V_{DD} range desired by the host. If the host does not accept voltage range, it shall not proceed further initialization sequence. The levels in the OCR register shall be defined accordingly (See Chapter 5.1).

SD_SEND_OP_COND (ACMD41) is used to start initialization and to check if the card has completed initialization. It is mandatory to issue CMD8 prior to the first ACMD41. Receiving of CMD8 expands the CMD58 and ACMD41 function; HCS (High Capacity Support) in the argument of ACMD41 and CCS (Card Capacity Status) in the response of CMD58. HCS is ignored by the card, which didn't accept CMD8. Standard Capacity SD Memory Card ignores HCS. The "in idle state" bit in the R1 response of ACMD41 is used by the card to inform the host if initialization of ACMD41 is completed. Setting this bit to "1" indicates that the card is still initializing. Setting this bit to "0" indicates completion of initialization. The host repeatedly issues ACMD41 until this bit is set to "0". The card checks the HCS bit in the OCR only at the first ACMD41. While repeating ACMD41, the host shall not issue another command except CMD0.

After initialization is completed, the host should get CCS information in the response of CMD58. CCS is valid when the card accepted CMD8 and after the completion of initialization. CCS=1 means that the card is a High Capacity SD Memory Card. CCS=0 means that the card is a Standard Capacity SD Memory Card.

7.2.2 Bus Transfer Protection

Every SD Card command transferred on the bus is protected by CRC bits. In SPI mode, the SD Memory Card offers a CRC ON 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 CRC OFF mode, the CRC bits of the command are defined as ‘don’t care’ for the transmitter and ignored by the receiver.
The SPI interface is initialized in the CRC OFF mode in default. However, the RESET command (CMD0) that is used to switch the card to SPI mode, is received by the card while in SD mode and, therefore, shall have a valid CRC field.
Since CMD0 has no arguments, the content of all the fields, including the CRC field, are constants and need not be calculated in run time. A valid reset command is:

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

After the card is put into SPI mode, CRC check for all commands including CMD0 will be done according to CMD59 setting.
The host can turn the CRC option on and off using the CRC_ON_OFF command (CMD59). Host should enable CRC verification before issuing ACMD41.
The CMD8 CRC verification is always enabled. The Host shall set correct CRC in the argument of CMD8. If CRC error is detected, card returns CRC error in R1 response regardless of command index.

7.2.3 Data Read

The SPI mode supports single block read and Multiple Block read operations (CMD17 or CMD18 in the SD Memory Card protocol). Upon reception of a valid read command the card will respond with a response token followed by a data token (refer to Figure 7-3). In case of Standard Capacity Card, the size in the data token is determined by the block length set by SET_BLOCKLEN (CMD16). In the case of a High Capacity Card, the data size in the data token for is fixed to 512 Bytes regardless of the block length set by CMD16.

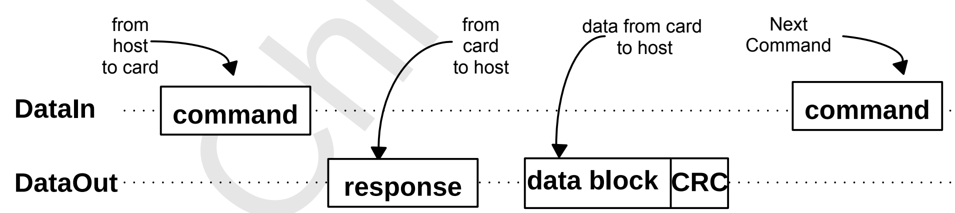


Figure 7-3: Single Block Read Operation

A valid data block is suffixed with a 16-bit CRC generated by the standard CCITT polynomial $x^{16}+x^{12}+x^5+1$.
The maximum block length is given by 512 Bytes regardless of READ_BL_LEN, defined in the CSD. If partial block access is enabled in Standard Capacity Card (i.e. the CSD parameter READ_BL_PARTIAL equals 1), the block length can be any number between 1 and 512 Bytes. The start address can be any byte address in the valid address range of the card. Every block, however, shall be contained in a single physical card sector.
If partial block access is disabled, only 512-Byte data length is supported.
The High Capacity SD Memory Card only supports 512-byte block length. The start address shall be aligned to the block boundary.

In the case of a data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 7-4 shows a data read operation that terminated with an error token rather than a data block.

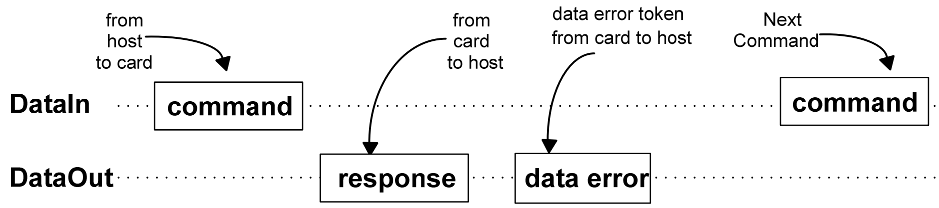


Figure 7-4: Read Operation - Data Error

In the case of a multiple block read operation every transferred block has its suffix of 16-bit CRC.
Stop transmission command (CMD12) will actually stop the data transfer operation (the same as in SD Memory Card operation mode).

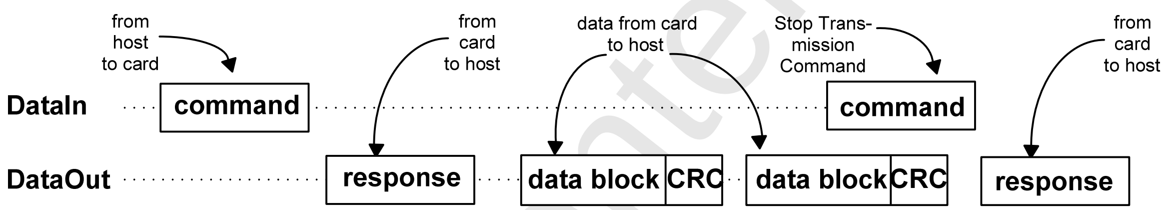


Figure 7-5: Multiple Block Read Operation

7.2.4 Data Write

The SPI mode supports single block and multiple block write commands. Upon reception of a valid write command (CMD24 or CMD25 in the SD Memory Card protocol), the card will respond with a response token and will wait for a data block to be sent from the host. CRC suffix, block length and start address restrictions are (with the exception of the CSD parameter WRITE_BL_PARTIAL controlling the partial block write option and WRITE_BL_LEN) identical to the read operation (see Figure 7-6).

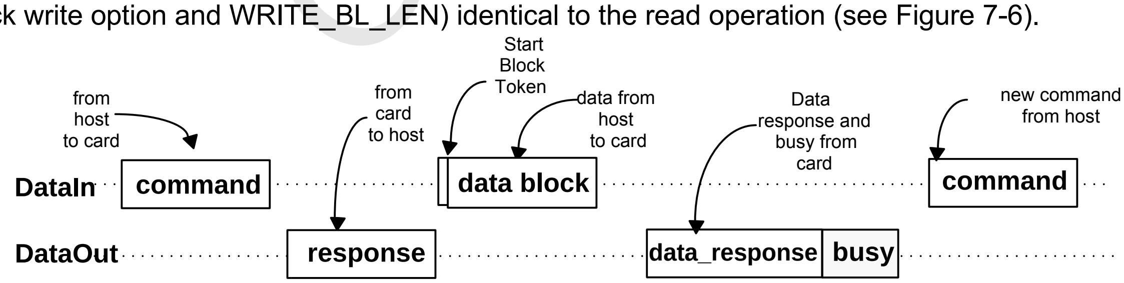


Figure 7-6: Single Block Write Operation

Every data block has a prefix of 'Start Block' token (one byte).
After a data block has been received, the card will respond with a data-response token. If the data block has been received without errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the DataOut line low).
Once the programming operation is completed, the host should 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 the CRC and general Write Error

indication.
In a Multiple Block write operation, the stop transmission will be done by sending 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block. In case of Write Error indication (on the data response) the host shall use SEND_NUM_WR_BLOCKS (ACMD22) in order to get the number of well written write blocks. The data tokens description is given in Chapter 7.3.3.2.

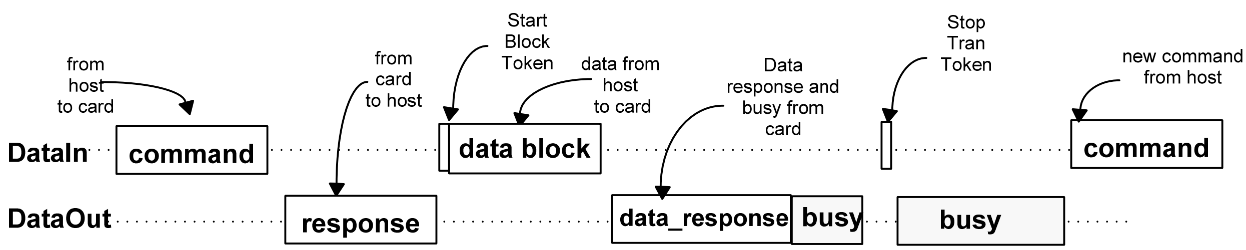


Figure 7-7: Multiple Block Write Operation

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

7.2.5 Erase & Write Protect Management

The erase and write protect management procedures in the SPI mode are identical to those of the SD mode. While the card is erasing or changing the write protection bits of the predefined sector list, it will be in a busy state and hold the DataOut line low. Figure 7-8 illustrates a 'no data' bus transaction with and without busy signaling.

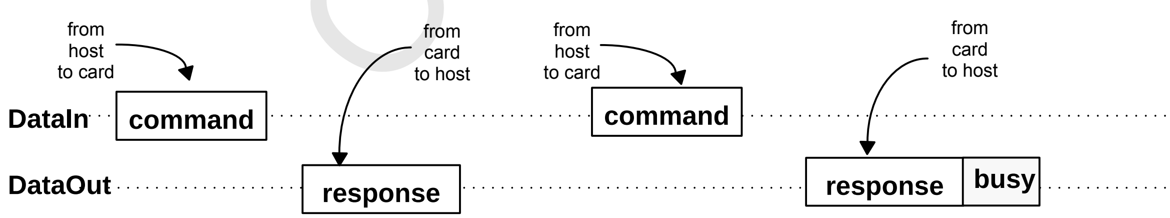


Figure 7-8: 'No data' Operations

7.2.6 Read CID/CSD Registers

Unlike the SD Memory Card protocol (where the register contents is sent as a command response), reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token (see Figure 7-3) followed by a data block of 16 bytes suffixed with a 16-bit CRC.
The data timeout for the CSD command cannot be set to the cards TAAC since this value is stored in the card's CSD. Therefore, the standard response timeout value (N_{CR}) is used for read latency of the CSD register.

7.2.7 Reset Sequence

The SD Memory Card requires a defined reset sequence. The card enters an idle state after power on reset or reset command (CMD0 for SD memory card). In this state, the only valid host commands are CMD8 (SEND_IF_COND), ACMD41 (SD_SEND_OP_COND), CMD58 (READ_OCR) and CMD59 (CRC_ON_OFF).
For the Thick (2.1 mm) SD Memory Card - CMD1 (SEND_OP_COND) is also valid - this means that in SPI mode, CMD1 and ACMD41 have the same behaviors, but the usage of ACMD41 is preferable since it allows easy distinction between an SD Memory Card and a MultiMediaCard. **For the Thin (1.4 mm) Standard Size SD Memory Card, CMD1 (SEND_OP_COND) is an illegal command during the initialization that is done after power on. After Power On, once the card has accepted valid ACMD41, it will be able to also accept CMD1 even if used after re-initializing (CMD0) the card.** It was defined in such way in order to be able to distinguish between a Thin SD Memory Card and a MultiMediaCard (that supports CMD1 as well).

7.2.8 Error Conditions

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

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

7.2.9 Memory Array Partitioning

Same as for SD mode.

7.2.10 Card Lock/Unlock

Usage of card lock and unlock commands in SPI mode is identical to SD mode. In both cases, the command is responded to with an R1b response type. After the busy signal clears, the host should obtain the result of the operation by issuing a SEND_STATUS command (CMD13). Refer to Chapter 4.3.7 for details.

7.2.11 Application Specific Commands

Identical to SD mode with the exception of the APP_CMD status bit (refer to Table 4-36), which is not available in SPI.

7.2.12 Content Protection Command

All the special Content Protection ACMDs and security functionality is the same as for SD mode.

7.2.13 Switch Function Command

Same as for SD mode with two exceptions:

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

7.2.14 High-Speed Mode

Same as for SD mode.

7.2.15 Speed Class Specification

As opposed to SD mode, the card cannot guarantee its Speed Class. In SPI mode, host shall treat the card as Class 0 no matter what Class is indicated in SD Status.

7.3 SPI Mode Transaction Packets

7.3.1 Command Tokens

7.3.1.1 Command Format

All the SD Memory Card commands are 6 bytes long. The command transmission always starts with the left most bit of the bit string corresponding to the command codeword. All commands are protected by a CRC (see Chapter 4.5). The commands and arguments are listed in Table 7-3.

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

Table 7-1: Command Format

7.3.1.2 Command Classes

As in SD mode, the SPI commands are divided into several classes (See Table 7-2). Each class supports a set of card functions. A SD Memory Card will support the same set of optional command classes in both communication modes (there is only one command class table in the CSD register). The available command classes, and the supported command for a specific class, however, are different in the SD Memory Card and the SPI communication mode.
Note that except for the classes that are not supported in SPI mode (class 1, 3 and 9), the mandatory required classes for the SD mode are the same for the SPI mode.

Card CMD Class (CCC)	**Class Description**	**Supported commands**																																	
		0	**1**	**5**	**6**	**8**	**9**	**10**	**12**	**13**	**16**	**17**	**18**	**24**	**25**	**27**	**28**	**29**	**30**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**42**	**50**	**52**	**53**	**55**	**56**	**57**	**58**	**59**
class 0	Basic	+	+			+	+	+	+	+																								+	+
class 1	Not supported in SPI																																		
class 2	Block read										+	+	+																						
class 3	Not supported in SPI																																		
class 4	Block write										+			+	+	+																			
class 5	Erase																			+	+					+									
class 6	Write-protection (Optional)																+	+	+																
class 7	Lock Card (Mandatory)										+																+								

Card CMD Class (CCC)	Class Description	Supported commands																																	
		0	1	5	6	8	9	10	12	13	16	17	18	24	25	27	28	29	30	32	33	34	35	36	37	38	42	50	52	53	55	56	57	58	59
class 8	Application specific																														+	+			
class 9	I/O mode			+																									+	+					
class 10[1]	Switch				+																	+	+	+	+			+					+		
class 11	Reserved																																		

Note (1): This command class was added in spec version 1.10

Table 7-2: Command Classes in SPI Mode

7.3.1.3 Detailed Command Description

The following table provides a detailed description of the SPI bus commands. The responses are defined in Chapter 7.3.2. Table 7-3 lists all SD Memory Card commands. A “yes” in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are reserved in SD mode as well.
The binary code of a command is defined by the mnemonic symbol. As an example, the content of the **command index** field is (binary) ‘000000’ for CMD0 and ‘100111’ for CMD39.
The card shall ignore stuff bits and reserved bits in a argument.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	[31:0] stuff bits	R1	GO_IDLE_STATE	Resets the SD Memory Card
CMD1	Yes[1]	[31]Reserved bit [30]HCS [29:0]Reserved bits	R1	SEND_OP_ COND	Sends host capacity support information and activates the card's initialization process. HCS is effective when card receives SEND_IF_COND command. Reserved bits shall be set to '0'.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	Reserved for I/O Mode (refer to the "SDIO Card Specification")				

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD6[8]	Yes	[31] Mode 0:Check function 1:Switch function [30:24] reserved (All '0') [23:20] reserved for function group 6 (All '0' or 0xF) [19:16] reserved for function group 5 (All '0' or 0xF) [15:12] reserved for function group 4 (All '0' or 0xF) [11:8] reserved for function group 3 (All '0' or 0xF) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Checks switchable function (mode 0) and switches card function (mode 1). See Chapter 4.3.10.
CMD7	No				
CMD8[9]	Yes	[31:12]Reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition that includes host supply voltage information and asks the accessed card whether card can operate in supplied voltage range. Reserved bits shall be set to '0'.
CMD9	Yes	[31:0] stuff bits	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD)
CMD10	Yes	[31:0] stuff bits	R1	SEND_CID	Asks the selected card to send its card identification (CID)
CMD11	No				

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD12	Yes	[31:0] stuff bits	R1b[5]	STOP_ TRANSMISSION	Forces the card to stop transmission in Multiple Block Read Operation
CMD13	Yes	[31:0] stuff bits	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	Sets a block length (in bytes) for all following block commands (read and write)[2] of a Standard Capacity Card. Block length of the read and write commands are fixed to 512 bytes in a High Capacity Card. The length of LOCK_UNLOCK command is set by this command in both capacity cards.
CMD17	Yes	[31:0] data address[10]	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[3]
CMD18	Yes	[31:0] data address[10]	R1	READ_MULTIPLE _BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.
CMD19	reserved				
CMD20	No				
CMD21... CMD23	reserved				
CMD24	Yes	[31:0] data address[10]	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[4]
CMD25	Yes	[31:0] data address[10]	R1	WRITE_MULTIPLE _BLOCK	Continuously writes blocks of data until 'Stop Tran' token is sent (instead 'Start Block').
CMD26	No				
CMD27	Yes	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28	Yes	[31:0] data address	R1b[5]	SET_WRITE_ PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). The High Capacity Card does not support this command.
CMD29	Yes	[31:0] data address	R1b[5]	CLR_WRITE_ PROT	If the card has write protection features, this command clears the write protection bit of the addressed group. The High Capacity Card does not support this command.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
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] The High Capacity Card does not support this command.
CMD31	reserved				
CMD32	Yes	[31:0] data address[10]	R1	ERASE_WR_BLK_START_ADDR	Sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address[10]	R1	ERASE_WR_BLK_END_ADDR	Sets the address of the last write block of the continuous range to be erased.
CMD34... CMD37[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD38	Yes	[31:0] stuff bits	R1b[5]	ERASE	Erases all previously selected write blocks
CMD39	No				
CMD40	No				
CMD41	Reserved				
CMD42	Yes	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to Set/Reset the Password or lock/unlock the card. A transferred data block includes all the command details - refer to Chapter 4.3.7. The size of the Data Block is defined with SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.
CMD43-49 CMD51	reserved				
CMD50[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD52... CMD54	Reserved for I/O Mode (refer to the "SDIO Card Specification")				
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Defines to the card that the next command is an application specific command rather than a standard command
CMD56	Yes	[31:1] stuff bits. [0]: RD/WR[7]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. In case of Standard Capacity SD Memory Card, the size of the Data Block shall be defined with SET_BLOCK_LEN command. Block length of this command is fixed to 512-byte in High Capacity Card.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD57[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD58	Yes	[31:0] stuff bits	R3	READ_OCR	Reads the OCR register of a card. CCS bit is assigned to OCR[30].
CMD59	Yes	[31:1] stuff bits [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off
CMD60-63	Reserved For Manufacturer				

1. CMD1 is valid command for the Thin (1.4mm) Standard Size SD Memory Card only if used after re-initializing a card (not after power on reset).
2. The default block length is as specified in the CSD.
3. The data transferred shall not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.
4. The data transferred shall not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD.
5. R1b: R1 response with an optional trailing busy signal
6. 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. The last (least significant) bit of the protection bits corresponds to the first addressed group. If the addresses of the last groups are outside the valid range, then the corresponding write protection bits shall be set to zero
7. RD/WR_: "1" the Host shall get a block of data from the card.
   "0" the host sends block of data to the card.
8. This command was added in spec version 1.10
9. This command is added in spec version 2.00
10. The unit of "data address" in argument is byte for Standard Capacity SD Memory Card and block (512 bytes) for High Capacity SD Memory Card.

Table 7-3: Commands and Arguments

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

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
ACMD6	No				
ACMD13	yes	[31:0] stuff bits	R2	SD_STATUS	Send the SD Status. The status fields are given in Table 4-38
ACMD17	reserved				
ACMD18	yes	--	--	--	Reserved for SD security applications1
ACMD19-ACMD21	reserved				
ACMD22	yes	[31:0] stuff bits	R1	SEND_NUM_WR_BLOCKS	Send the numbers of the well written (without errors) blocks. Responds with 32-bit+CRC data block.
ACMD23	yes	[31:23] stuff bits [22:0]Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"=default (one wr block)(2).
ACMD24	reserved				
ACMD25	yes	--	--	--	Reserved for SD security applications[1]
ACMD26	yes	--	--	--	Reserved for SD security applications[1]
ACMD38	yes	--	--	--	Reserved for SD security applications[1]
ACMD39 - ACMD40	reserved				
ACMD41	Yes	[31]Resetved bit [30]HCS [29:0]Reserved bits	R1	SD_SEND_OP_COND	Sends host capacity support information and activates the card's initialization process. Reserved bits shall be set to '0'
ACMD42	yes	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_DETECT	Connect[1]/Disconnect[0] the 50 KOhm pull-up resistor on CS (pin 1) of the card. The pull-up may be used for card detection.
ACMD43-ACMD49	yes	--	--	--	Reserved for SD security applications[1]
ACMD51	yes	[31:0] staff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

(1) Refer to the "SD Specifications Part3 Security Specification" for detailed explanation about the SD Security Features
(2) Stop Tran Token shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

Table 7-4: Application Specific Commands used/reserved by SD Memory Card - SPI Mode

7.3.1.4 Card Operation for CMD8 in SPI mode

In SPI mode, the card always returns response. Table 7-5 shows the card operation for CMD8.

Command Argument Check					Response of Card *1				
Index	Reserved	VHS	Pattern	CRC	R1	Ver	Reserved	VCA	Pattern
=8	Don't Care	Don't Care	Don't Care	Error	09h	(R1 only)			
Not 8	Don't Care	Don't Care	Don't Care	Don't Care	Depends on command index				
=8	Don't Care	Mismatch *2	Don't Care	Correct	01h	Ver=0	0	0	Echo Back
=8	Don't Care	Match *2	Don't Care	Correct	01h	Ver=0	0	Echo Back	Echo Back

*1: Response indicates the actual response that the card returns. (It does not include errors during transfer response.)
*2: 'Match' means AND of following condition a) and b). 'Mismatch' is other cases.
a) Only 1 bit is set to '1' in VHS.
b) The card supports the host supply voltage.

Table 7-5: Card Operation for CMD8 in SPI Mode

7.3.2 Responses

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

7.3.2.1 Format R1

This response token is sent by the card after every command with the exception of SEND_STATUS commands. It is one byte long, and the MSB is always set to zero. The other bits are error indications, an error being signaled by a 1. The structure of the R1 format is given in Figure 7-9. The meaning of the flags is defined as following:

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

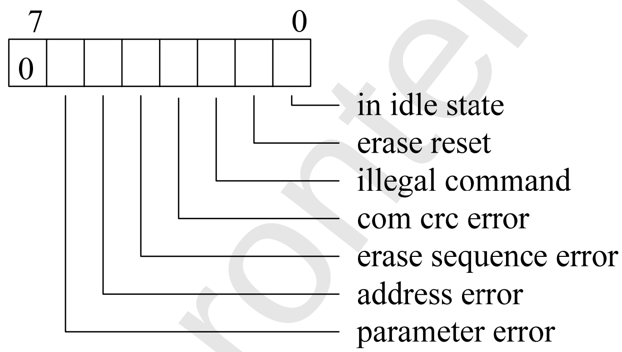


Figure 7-9: R1 Response Format

7.3.2.2 Format R1b

This response token is identical to the 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 the card is ready for the next command.

7.3.2.3 Format R2

This response token is two bytes long and sent as a response to the SEND_STATUS command. The format is given in Figure 7-10.

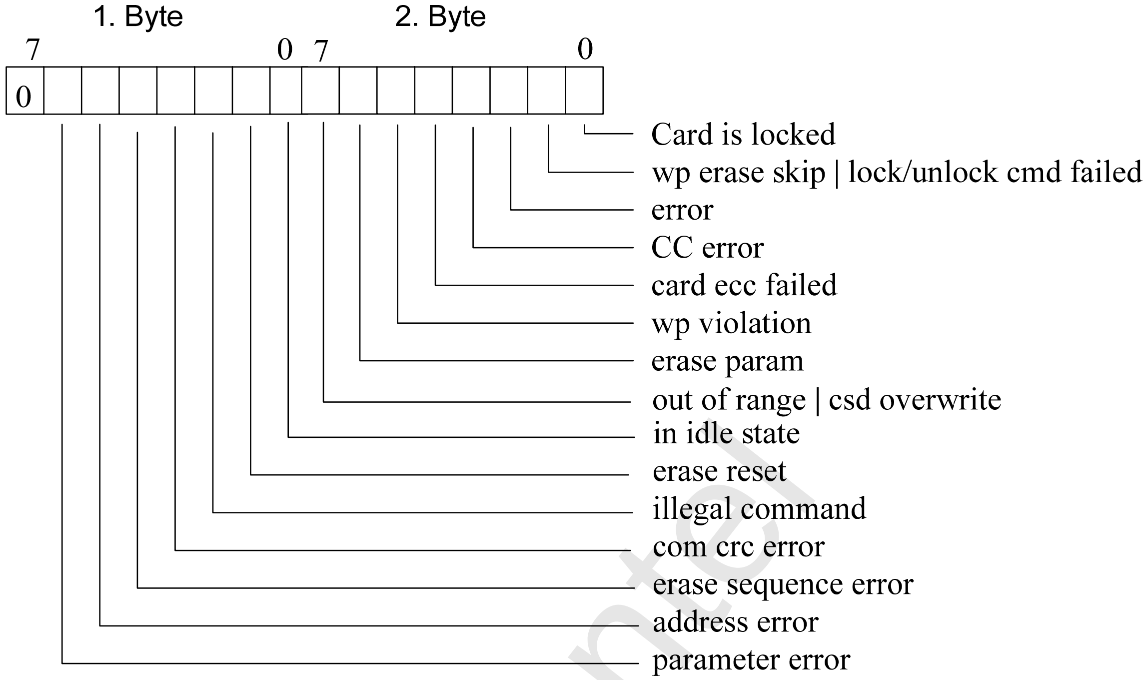


Figure 7-10: R2 Response Format

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

- **Erase param**: An invalid selection for erase, sectors or groups.
- **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 | lock/unlock command failed**: This status bit has two functions overloaded. It is set when the host attempts to erase a write-protected sector or makes a sequence or password errors during card lock/unlock operation.
- **Card is locked**: Set when the card is locked by the user. Reset when it is unlocked.

7.3.2.4 Format R3

This response token is sent by the card when a READ_OCR command is received. The response length is 5 bytes (see Figure 7-11). The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

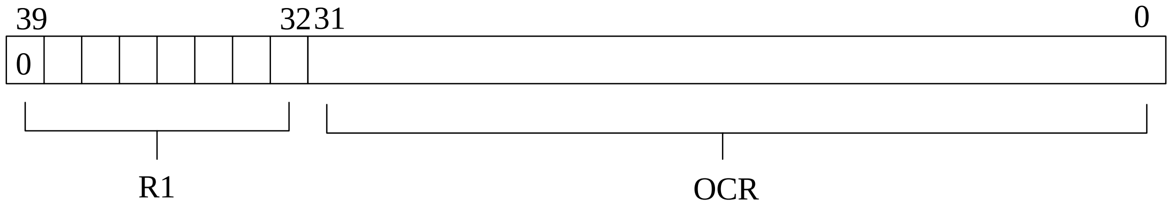


Figure 7-11: R3 Response Format

7.3.2.5 Formats R4 & R5

Those response formats are reserved for I/O mode (refer to the "SDIO Card Specification").

7.3.2.6 Format R7

This response token is sent by the card when a SEND_IF_COND command (CMD8) is received. The response length is 5 bytes. The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the card operating voltage information and echo back of check pattern in argument and are specified by the same definition as R7 response in SD mode. (Refer to Section 4.9).

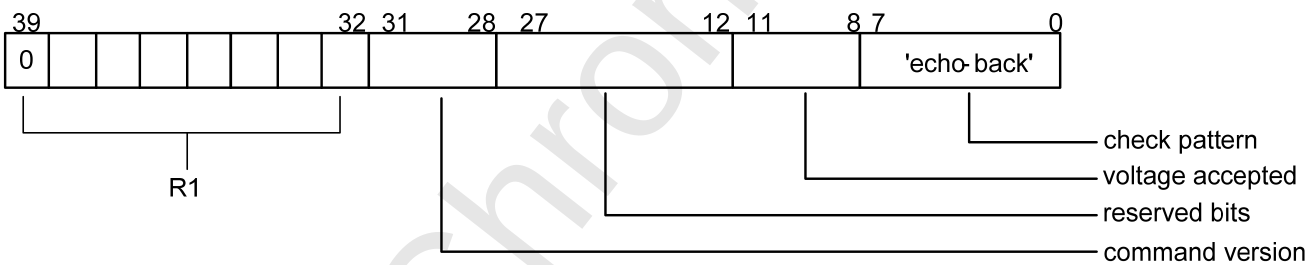


Figure 7-12: R7 Response Format

7.3.3 Control Tokens

Data block transfer is controlled by some tokens.

7.3.3.1 Data Response Token

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

7	6				0
x	x	x	0	Status	1

The meaning of the status bits is defined as follows:
‘010’ - Data accepted.
‘101’ - Data rejected due to a CRC error.
’110’ - Data Rejected due to a Write Error

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

7.3.3.2 Start Block Tokens and Stop Tran Token

Read and write commands have data transfers associated with them. Data is being transmitted or received via data tokens. All data bytes are transmitted MSB first.
Data tokens are 4 to 515 bytes long and have the following format:
<u>For Single Block Read, Single Block Write and Multiple Block Read:</u>

- First byte: Start Block

7							0
1	1	1	1	1	1	1	0

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

<u>For Multiple Block Write operation:</u>

- First byte of each block:

If data is to be transferred then - Start Block Token

7							0
1	1	1	1	1	1	0	0

If Stop transmission is requested - Stop Tran Token

7							0
1	1	1	1	1	1	0	1

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

7.3.3.3 Data Error Token

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

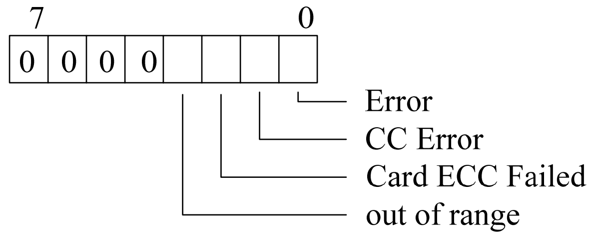


Figure 7-13: Data Error Token

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

7.3.4 Clearing Status Bits

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

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

Identifier	Included in resp	Type[1]	Value	Description	Clear Condition[2]
Error	R2 dataEr	E R X	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
CSD_OVERWRITE	R2	E R X	'0'= no error '1'= error	Can be either of the following errors: - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
WP erase skip	R2	S X	'0'= not protected '1'= protected	Only partial address space was erased due to existing write protected blocks.	C
Lock/Unlock cmd failed	R2	X	'0'= no error '1'= error	Sequence or password errors during card lock/unlock operation.	C
Card is locked	R2	S X	'0' = card is not locked '1' = card is locked	Card is locked by a user password.	A
Erase reset	R1 R2	S R	'0'= cleared '1'= set	An erase sequence was cleared before executing because an out of erase sequence command was received	C
In Idle state	R1 R2	S R	0 = Card is ready 1 = Card is in idle state	The card enters the idle state after power up or reset command. It will exit this state and become ready upon completion of its initialization procedures.	A

Table 7-6: SPI Mode Status Bits

1) Type:
E: Error bit.
S: State bit.
R: Detected and set for the actual command response.
X: Detected and set during command execution. The host can get the status by issuing a command with R1 response.

2) Clear Condition:
A: According to the current state of the card.
C: Clear by read

7.4 Card Registers

In SPI mode, only the RCA register is not accessible. Formats of other registers are identical to the formats in the SD mode.

7.5 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 7-7. The host shall keep the clock running for at least N_{CR} clock cycles after receiving the card response. This restriction applies to both command and data response tokens.

7.5.1 Command/Response

7.5.1.1 Host Command to Card Response - Card is ready

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

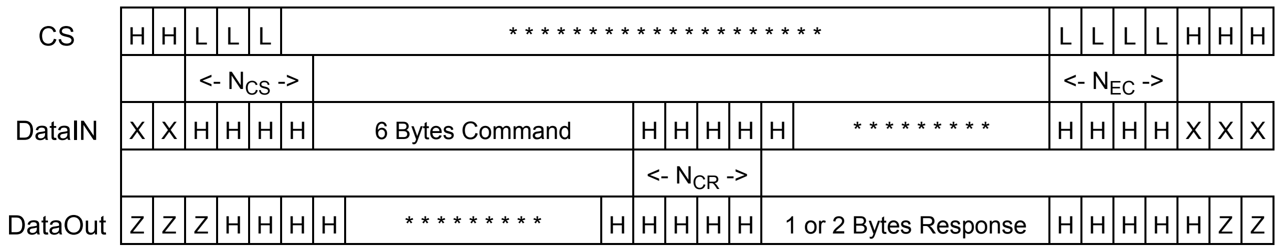


Figure 7-14: Basic Command Response

7.5.1.2 Host Command to Card Response - card is busy

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

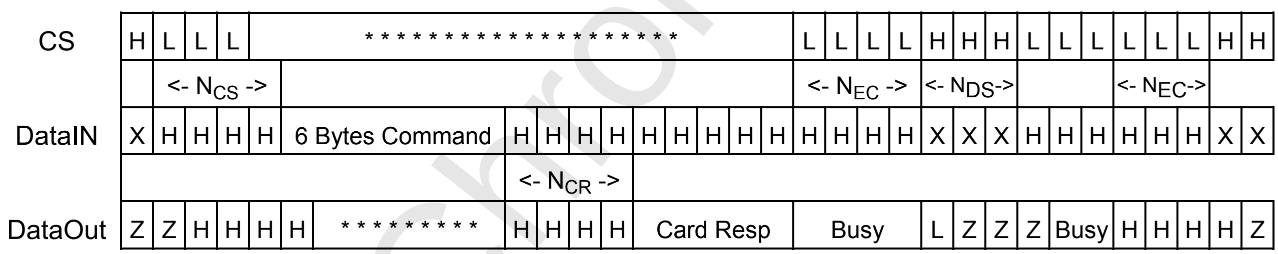


Figure 7-15: Command Response with Busy Indication (R1b)

7.5.1.3 Card Response to Host Command

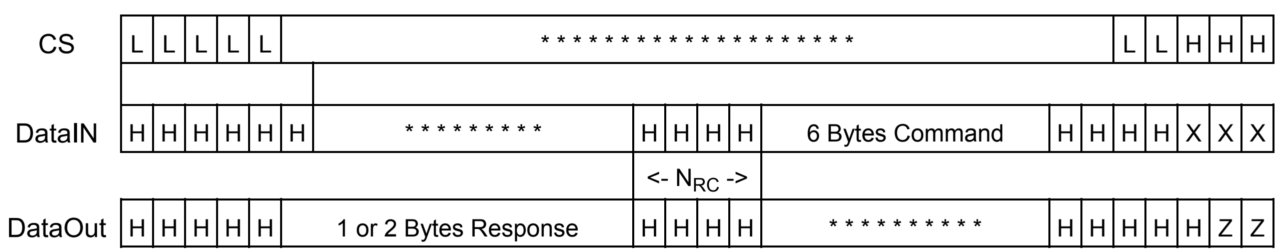


Figure 7-16: Timing between Card Response to new Host Command

7.5.2 Data Read

7.5.2.1 Timing of Single Block Read Operation

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

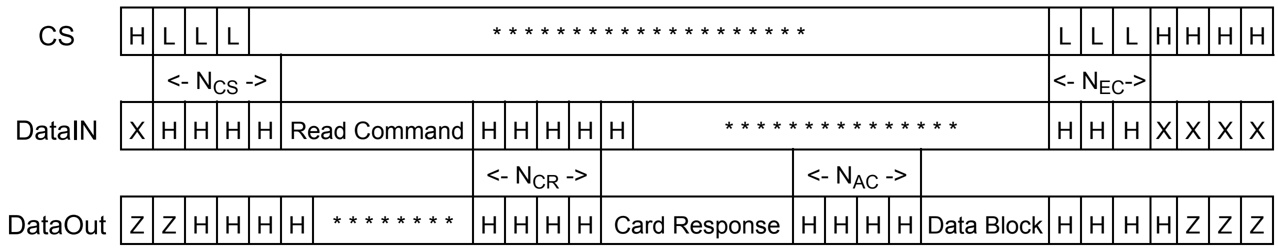


Figure 7-17: Read Single Block Operations - Bus Timing

7.5.2.2 Stop Transmission Timing of Multiple Block Read Operation

The following table describes Stop transmission operation in case of Multiple Block Read.
Clock cycle between read data blocks are defined by NAC (Not shown in Figure 7-18).

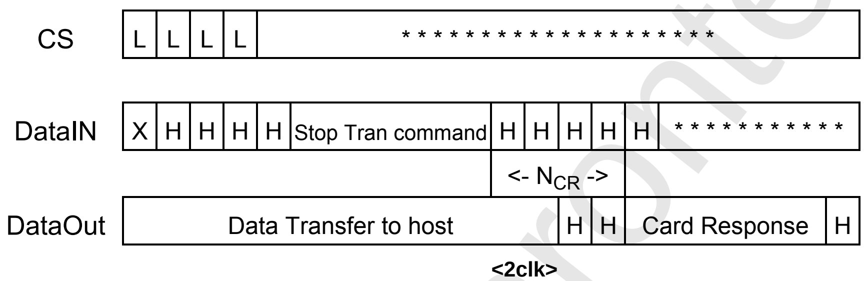


Figure 7-18: Stop Transmission in Read Multiple Block

7.5.2.3 Reading the CSD or CID register

The following timing diagram describes the SEND_CSD and SEND_CID command bus transactions. The timeout values for the response and the data block are Ncr and Ncx respectively (Since the Nac is still unknown).

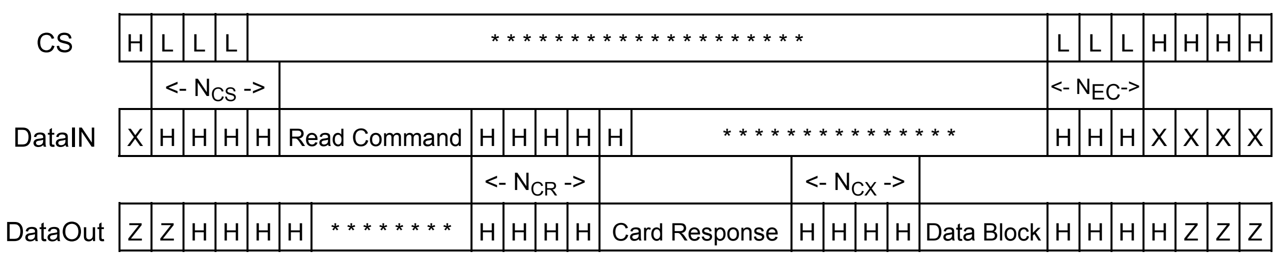


Figure 7-19: Read CSD/CID - Bus Timing

7.5.3 Data Write

7.5.3.1 Timing of Multiple Block Write Operation

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

CS	H, L, ********************, L, L, L, L, L, L, L, L, H, H, H, L, L, L, L
	< N_{CS}->, <-N_{WR}->, <- N_{EC} ->, <- N_{DS} ->
DataIN	X, H, H, H, Write Command, H, H, H, H, H, H, H, H, H, Data Block, H, H, H, H, H, H, X, X, X, H, H, H, H
	<-N_{CR}->
DataOut	Z, Z, H, H, H, ********, H, H, H, Card Response, H, H, H, H, H, H, H, Data Resp, Busy, L, Z, Z, Z, Busy, H

Figure 7-20: Write Operation - Bus Timing

7.5.3.2 Stop Transmission Timing of Multiple Block Write Operation

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

CS	L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, L, H, H, H, L, L, L, L
	<N_{WR} ->, <1byte->, <N_{BR} ->, <N_{EC} ->, <- N_{DS} ->
Data In	Data Block, H, H, H, H, H, H, H, H, H, stop tran token, H, H, H, H, H, H, X, X, X, H, H, H, H
Data Out	H, H, H, H, Data Resp, Busy, H, H, H, H, H, H, H, H, H, Busy(1), L, Z, Z, Z, Busy(1), H

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

Figure 7-21: Stop Transmission in Write Multiple Block

7.5.4 Timing Values

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

1) The maximum read access time for a Standard Capacity SD Memory Card shall be calculated by host as follows:
Nac(max)= 100 ((TAAC * fpp) + (100 * NSAC)) ;
fpp is the interface clock rate and TAAC & NSAC are given in the CSD (Chapter 5.3).
In the case of a High Capacity SD Memory Card, a fixed value (100 ms) shall be used for the maximum read access time.
Details of read, write and erase timeout are described in 4.6.2

Table 7-7: Timing Values

7.6 SPI Electrical Interface

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

7.7 SPI Bus Operating Conditions

Bus operating conditions are identical to SD mode

7.8 Bus Timing

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

8. SD Memory Card Mechanical Specification

This chapter describes the mechanical and electrical features, as well as the minimal recommendations for the SD Memory Card connector. The electrical features are applied to all types of SD Memory Cards and the mechanical features are applied to the Standard Size SD Memory Card. All technical drafts follow the DIN ISO standard.

The functions of the card package are:

- Protecting the chip
- Easy handling for the end user
- Reliable electrical interconnection
- Reliable write protect/card detection capability
- Bearing textual information and image
- Appealing appearance

The functions of the connector are:

- Attaching and fixing the card
- Electrical interconnecting the card to the system board
- Write protect/card detect indication
- Optional: switch on/off power supply
- Protection against card inverse insertion

8.1 Card Package

Every card package shall have the characteristics described in the following sections.

8.1.1 Design and Format

Dimensions SD Memory Card package	24 mm x 32 mm; (min. 23.9 mm x 31.9 mm; max.24.1 mm x 32.1 mm) Other dimensions Figure 8-1 Testing according to MIL STD 883, Meth 2016
Thickness	'Inter Connect Area': 2.1 mm +/- 0.15 mm or 1.4 mm+/-0.15 mm for Thin SD Card. 'Substrate Area': Max 2.25 mm or Max 1.55 for Thin SD Card - see Figure 8-2.
Label or printable area	In 'Substrate Area' only – see Figure 8-3
Surface	Plain (except contact area)
Edges	Smooth edges, see Figure 8-2, Figure 8-3
Inverse insertion	Protection on left corner (top view) see Figure 8-5
Position of ESC contacts	Along middle of shorter edge see Figure 8-1 and Figure 8-2

Table 8-1: SD Memory Card Package – Dimensions

8.1.2 Reliability and Durability

Temperature	Operation: -25°C/85°C Storage: -40°C (168h)/85°C (500 h) Junction temperature: max. 95°C
Moisture and corrosion	Operation: 25°C/95% rel. humidity Storage: 40°C/93% rel. hum./500h Salt water spray: 3% NaCl/35C; 24 h acc. MIL STD Method 1009
Durability	10.000 mating cycles; Test procedure: tbd.
Bending (note 1)	10 N
Torque (note 1)	0.15N.m or +/-2.5 deg .
Drop test	1.5 m free fall
UV light exposure	UV: 254 nm, 15 Ws/cm² according to ISO 7816-1
X-ray exposure	0.1 Gy of medium-energy radiation (70 keV to 140 keV, cumulative dose per year) to both sides of the card, according to ISO7816-1.
Visual inspection shape and form (note 1)	No warpage; no mold skin; complete form; no cavities surface smoothness <= -0.1 mm/cm² within contour; no cracks; no pollution (fat, oil dust, etc.)
Minimum moving force of WP switch	40 gf (Ensures that the WP switch will not slide while it is inserted to the connector).
WP Switch cycles	Minimum 1000 Cycles (@ Slide force 0.4 N to 5 N)

Note (1): The SDA's recommended test methods for Torque, bending and Warpage are defined in a separate Application Notes document.

Table 8-2: Reliability and Durability

8.1.3 Electrical Static Discharge (ESD) Requirement

ESD testing should be conducted according to IEC61000-4-2

Required ESD parameters are:

(1) Human body model +- 4 KV 100 pf/1.5 Kohm
(2) Machine model +- 0.25 KV 200 pf/0 ohm

Contact Pads:

+/- 4 kV, Human body model according to IEC61000-4-2

Non Contact Pads area:

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

8.1.4 Quality Assurance

The product traceability shall be ensured by an individual card identification number.

8.2 Mechanical Form Factor

The following 3 technical drawings define the card package of an SD Memory Card with 2.1+/-0.15 mm card thickness (the Thin SD Memory Card drawings are given in Chapter 8.4).

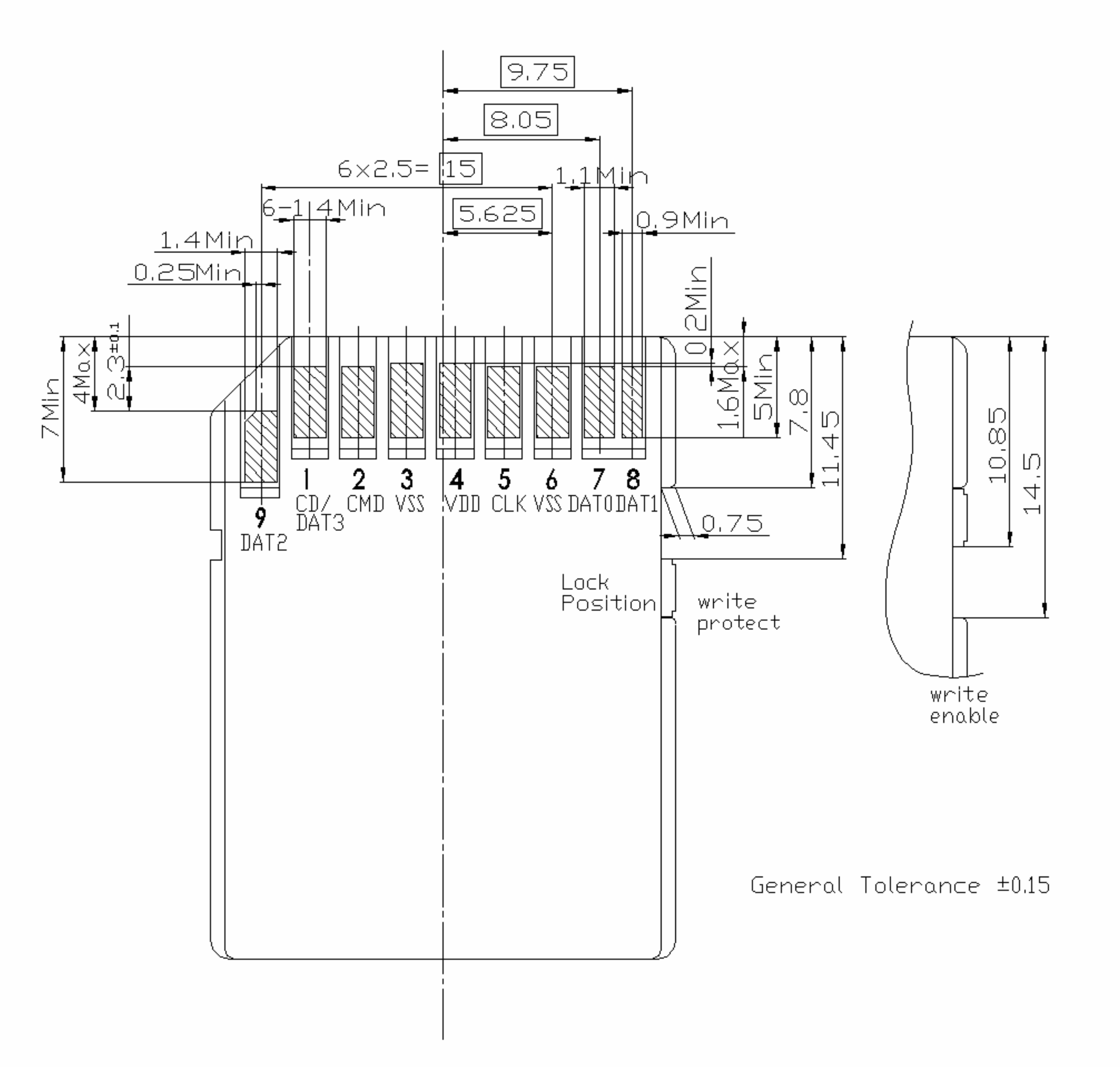


Figure 8-1: SD Memory Card - Mechanical Description (1 out of 3)

Notes for all the mechanical descriptions of the SD Memory Card (including Thin SD Memory Card):
Note 1): The numbers enclosed by a square indicate the distance between base lines. Those values are for information only (the given general tolerances are not related to them).
Note 2): Pins #3 and #4 are located "0.2 Min." ahead of pins #1, #2, #5, #6, #7 and #8.
Note 3): This value "1.6 Max." is for pins #1, #2, #5, #6, #7, #8.

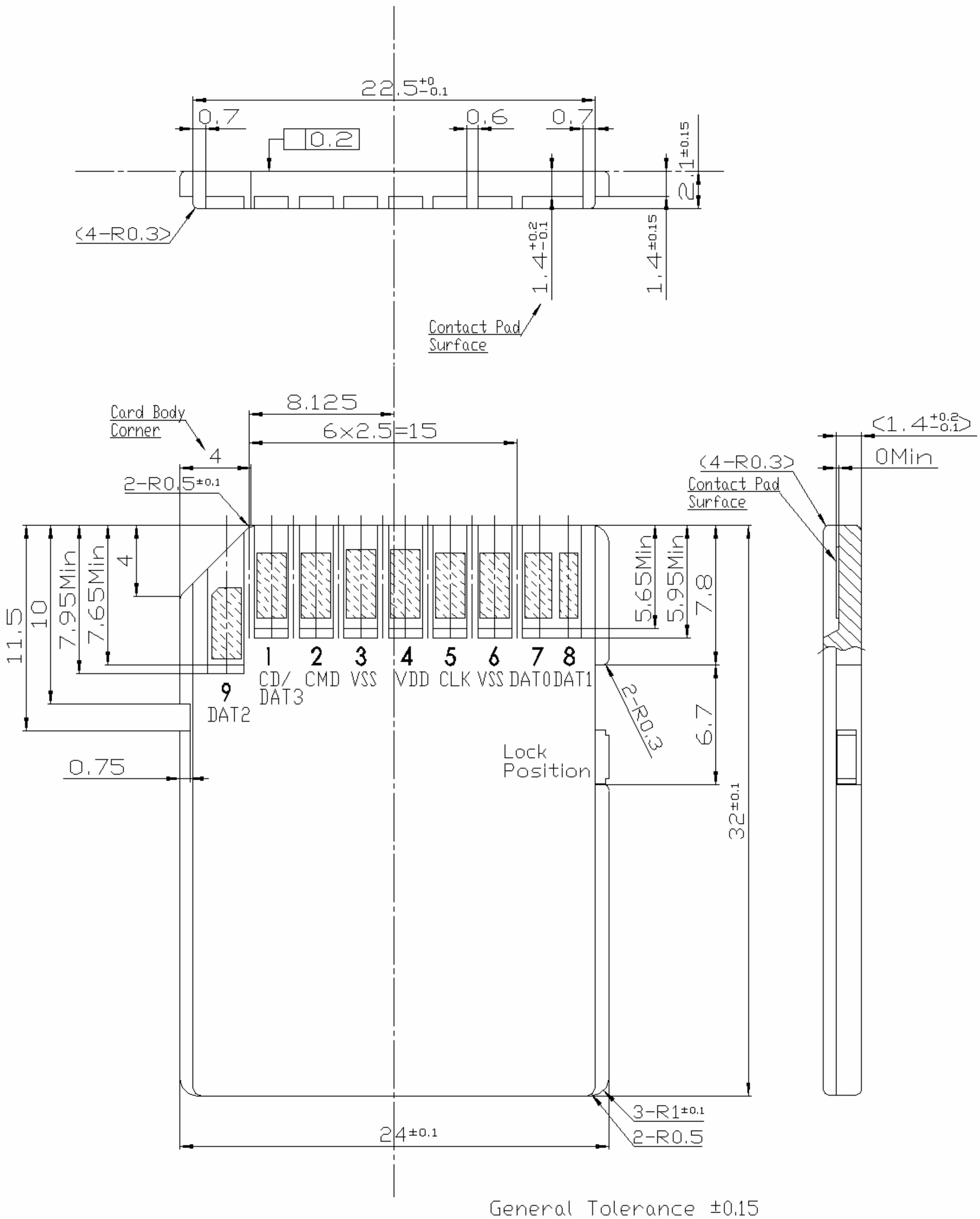
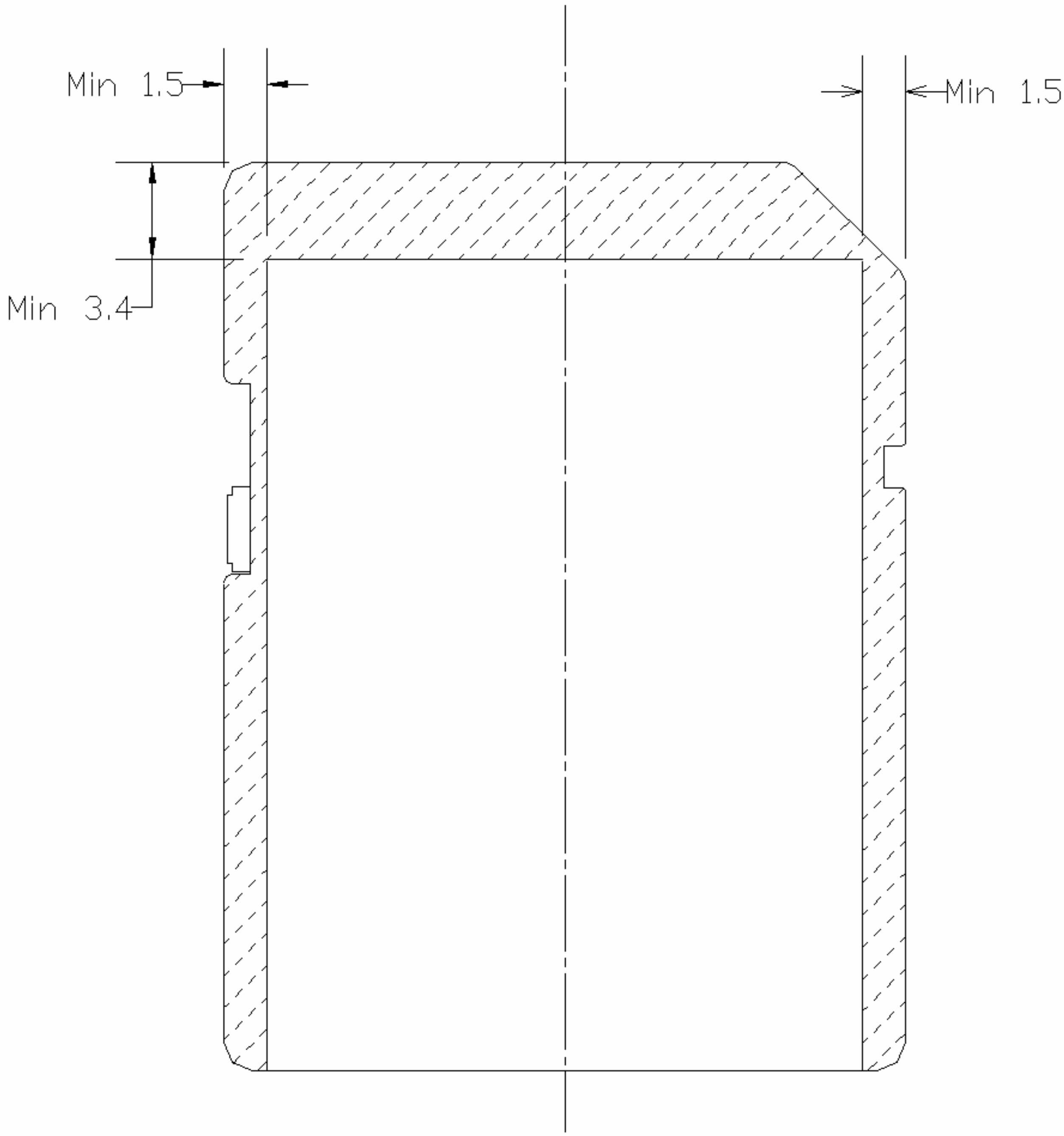


Figure 8-2: SD Memory Card - Mechanical Description (2 out of 3)



Note: Refer to the definition of 'substrate' and 'Inter Connect' areas (See Table 8-1).

Figure 8-3: SD Memory Card - Mechanical Description (3 out of 3)

Figure 8-4 describes the Write Protect switch position for all cases and card types.

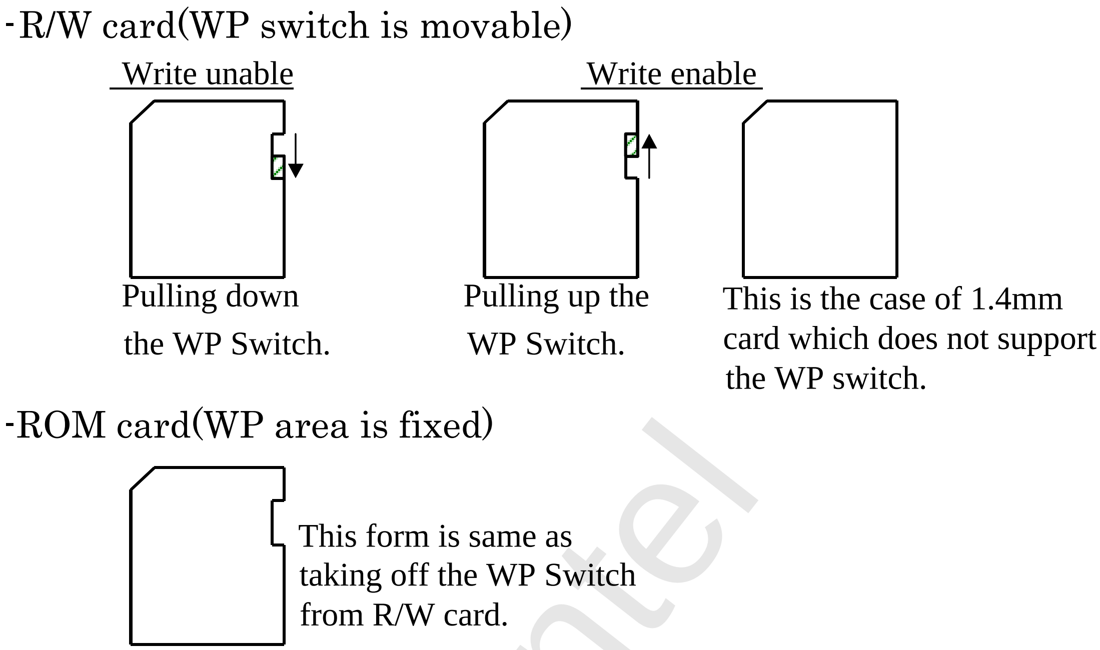


Figure 8-4: WP Switch Definition for All Cases and Card Types

8.3 System: card and connector

The description of the connector is outside the scope of this document. However, minimal recommendations for the connector comprise the ability to guarantee Write Protect and Card Detection, hot insertion and removal of the card, and to prevent inverse insertion.

8.3.1 Card Hot Insertion

To guarantee reliable initialization during hot insertion, some measures shall be taken on the host side. For instance, a special hot-insertion capable card connector may be used to guarantee the proper sequence of card pin connection.

The card contacts are contacted in three steps:

1) Ground V_{SS} (pin 3) and supply voltage V_{DD} (pin 4).
2) CLK, CMD, DAT0, DAT1, DAT2 and V_{SS} (pin 6).
3) CD/DAT3 (pin 1).

Pins 3 and 4 should make first contact when inserting, and release last when extracting.

As another method, a switch could ensure that the power is switched on only after all card pads are contacted. Of course, any other similar mechanism is allowed.

8.3.2 Inverse Insertion

Inverse insertion is prevented by the slanted corners of SD Memory Card and connector.
The connector should prevent inverse insertion by detecting the slanted corner of SD Memory Card.

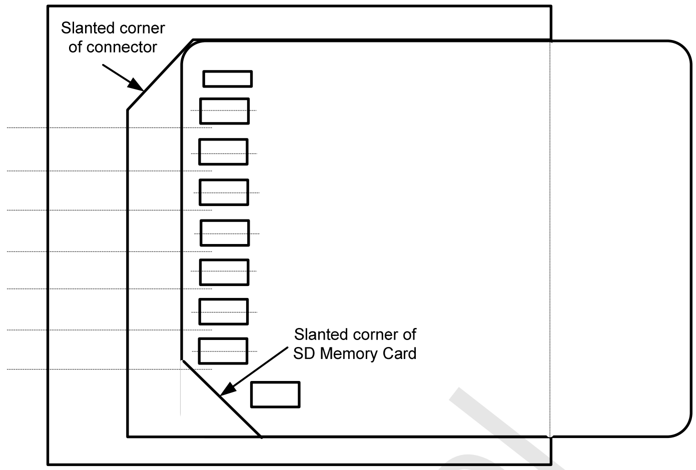


Figure 8-5: Inverse Insertion

8.3.3 Card Orientation

For the benefit of unified terminology when discussing the three dimensional orientation of a card (e.g. for the connector definition), the non contact-pads side (the side with the card label) is defined as the TOP side of the card and the contact-pads side of the card is defined as the BOTTOM side of the card.

8.3.4 Write Protect Detection

For an SD Memory Card, the Write Protect switch is always mandatory for the card side. Note that for hosts that support Read Only operations, the Write Protect detection capability in the SD Connector (Host side) is not necessary.

8.4 Thin (1.4 mm) Standard Size SD Memory Card

SD Memory Cards with mechanical dimensions that will be suitable for extra small applications will be available.
The "Thin Standard Size SD Memory card" has a form factor very similar to the SD Memory Card that is given in Chapter 8.2 above except for its thickness. The thickness of the "Thin Standard Size SD Memory Card" is 1.4 mm+/-0.15 mm.
Figure 8-6, Figure 8-7, and Figure 8-3 provide the mechanical drawings of a Thin Standard Size SD Memory Card.
Note that even though the WP switch appears in the given diagram, it was defined as an optional for a Thin Standard Size SD Memory Card.

Figure 8-6: Mechanical Drawing of Thin SD Memory Card (1 out of 2)

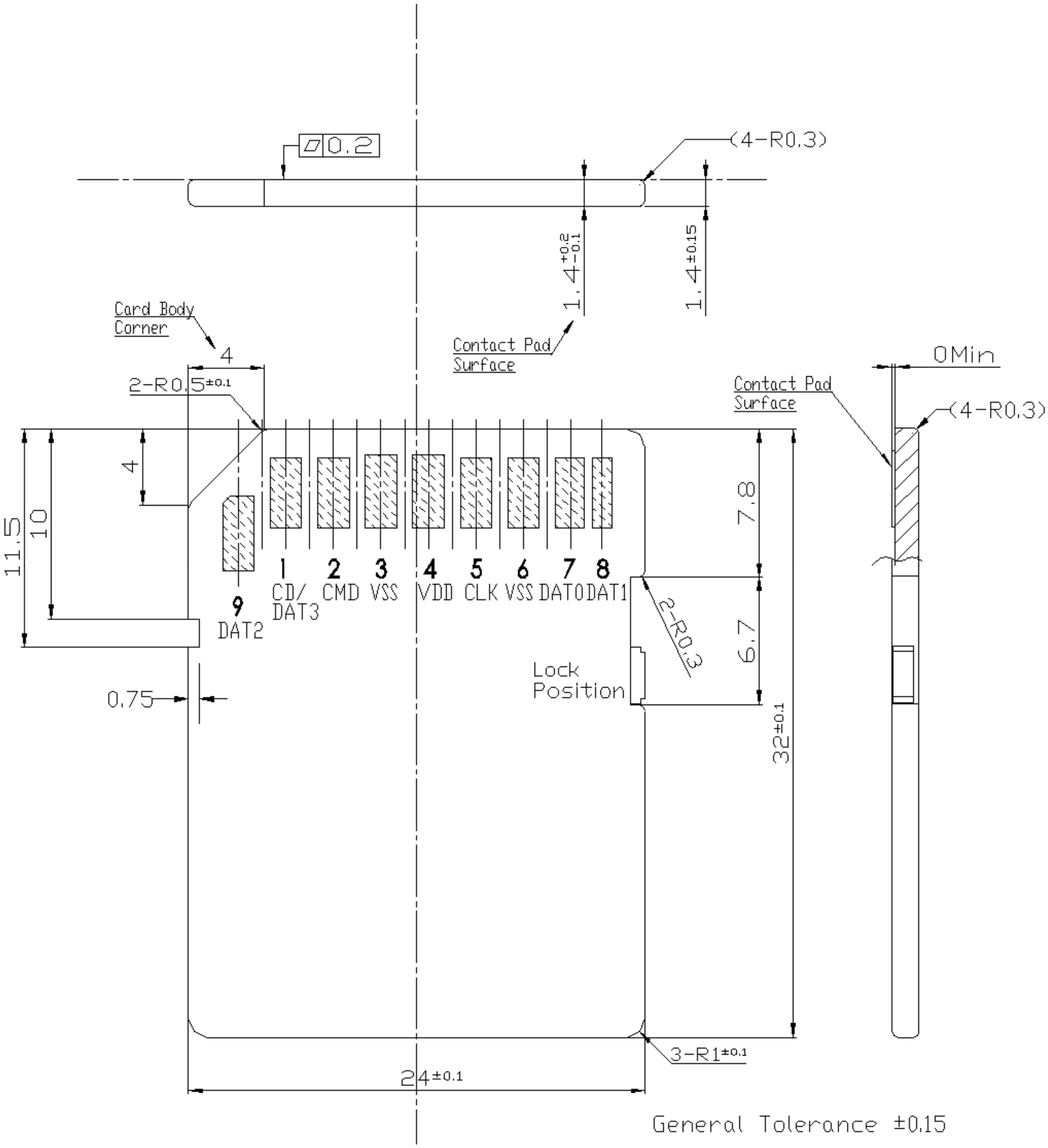


Figure 8-7: Mechanical Drawing of Thin SD Memory Card (2 out of 2)

8.5 Nonconductive Area

Figure 8-8 shows "Nonconductive Areas" in Standard Size SD Memory Card. Conductive materials shall not be included in the hatching areas. This is the same definition as in miniSD Memory Card adaptor (Refer to "Application Notes relating to miniSD Physical Specification").

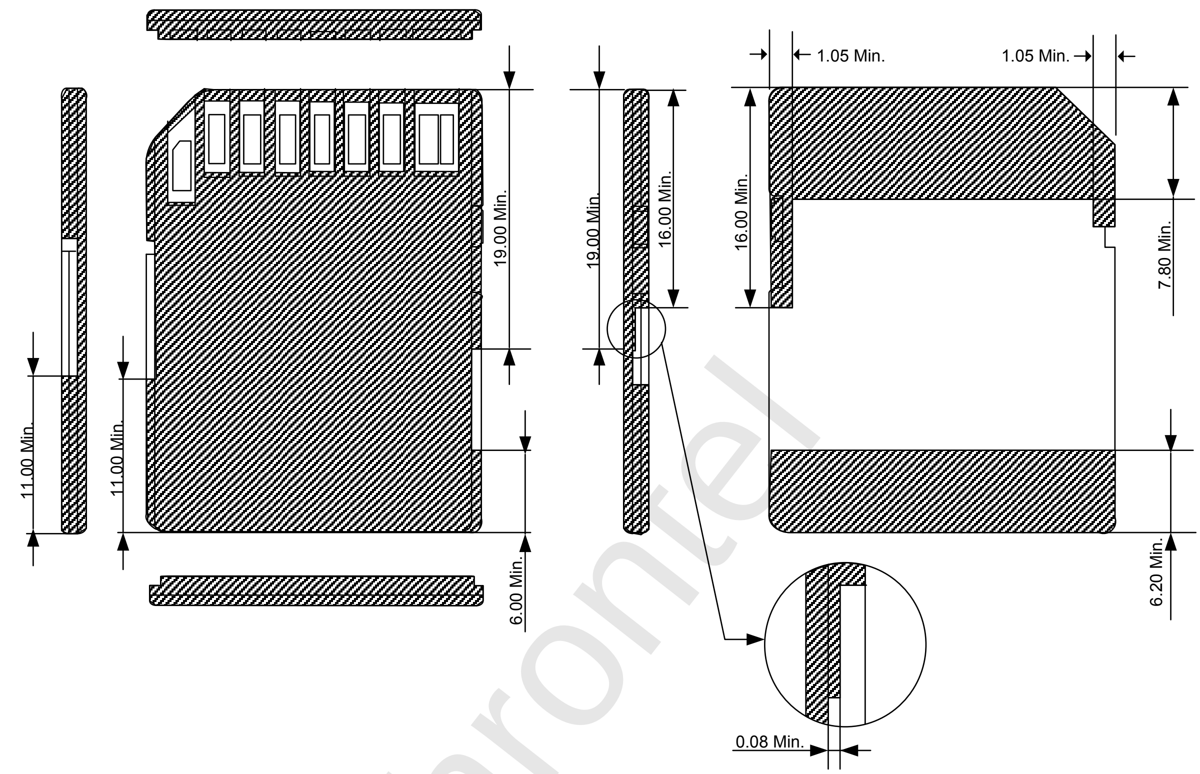


Figure 8-8: Nonconductive Areas of SD Memory Card

Appendix A

A.1 Power Supply Decoupling

The V_{SS1}, V_{SS2} and V_{DD} lines supply the card with operating voltage. For this, decoupling capacitors for buffering current peak are used. These capacitors are placed on the bus side corresponding to Figure 8-9.

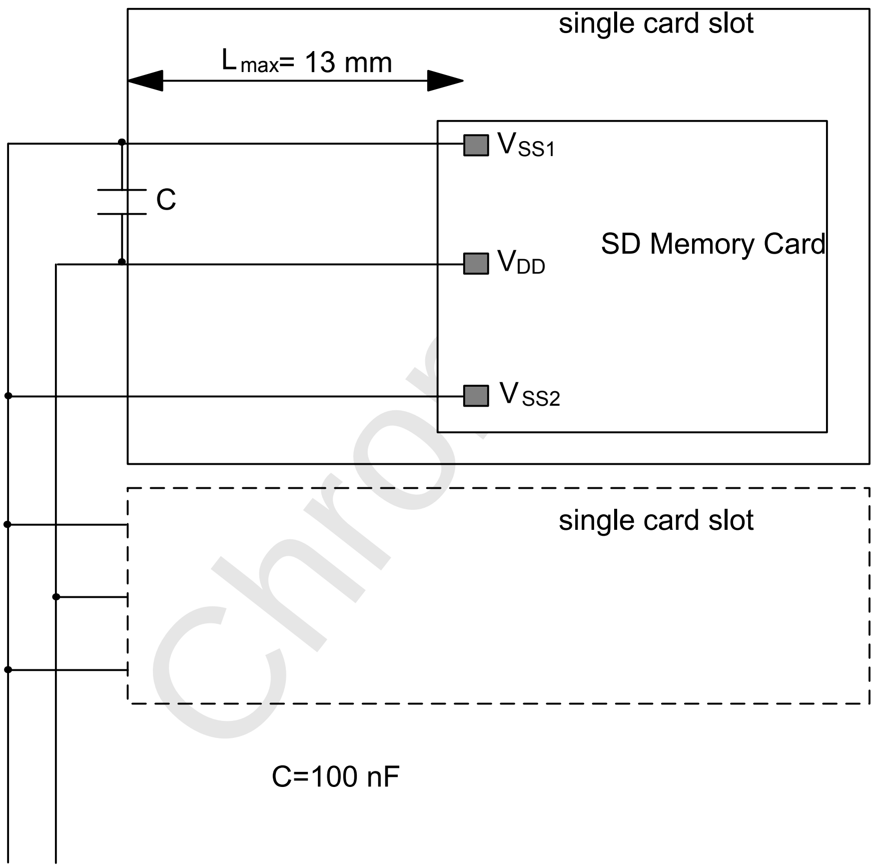


Figure 8-9: Power Supply Decoupling

The host controller includes a central buffer capacitor for V_{DD}. Its value is 1 µF/slot.

A.2 Connector

The connector described in this chapter serves as an example and is subject to further changes.

A.2.1 General

The connector housing which accommodates the card is formed of plastic. Inside are 9 contact springs for contacting the pads of the inserted card. When the connector supports Standard Size SD Memory Card, a Write Protect/Card Detection switch shall be part of the housing in order to be able to detect the position of the Write Protect sliding tablet on the card. Testing procedures are performed according to DIN IEC 68.

A.2.2 Card Insertion and Removal

Insertion of the SD Memory Card is only possible with the contact area of the card and the contact area of the connector in the correct position relative to each other. This is ensured by the slanted corners of the card and the connector, respectively.
To guarantee reliable initialization during hot insertion, some measures should be taken on the host side. One possible solution is shown in Figure 8-10. It is based on the idea of a defined sequence for card contact connection during the card insertion process. The card contacts are contacted in three steps:

1) Ground V_{SS} (pin 3) and supply voltage V_{DD} (pin 4).
2) CLK, CMD, DAT0, DAT1, DAT2 and V_{SS} (pin 6).
3) CD/DAT3 (pin 1).

Pins 3 and 4 should make contact first when inserting the card and release last when extracting the card.

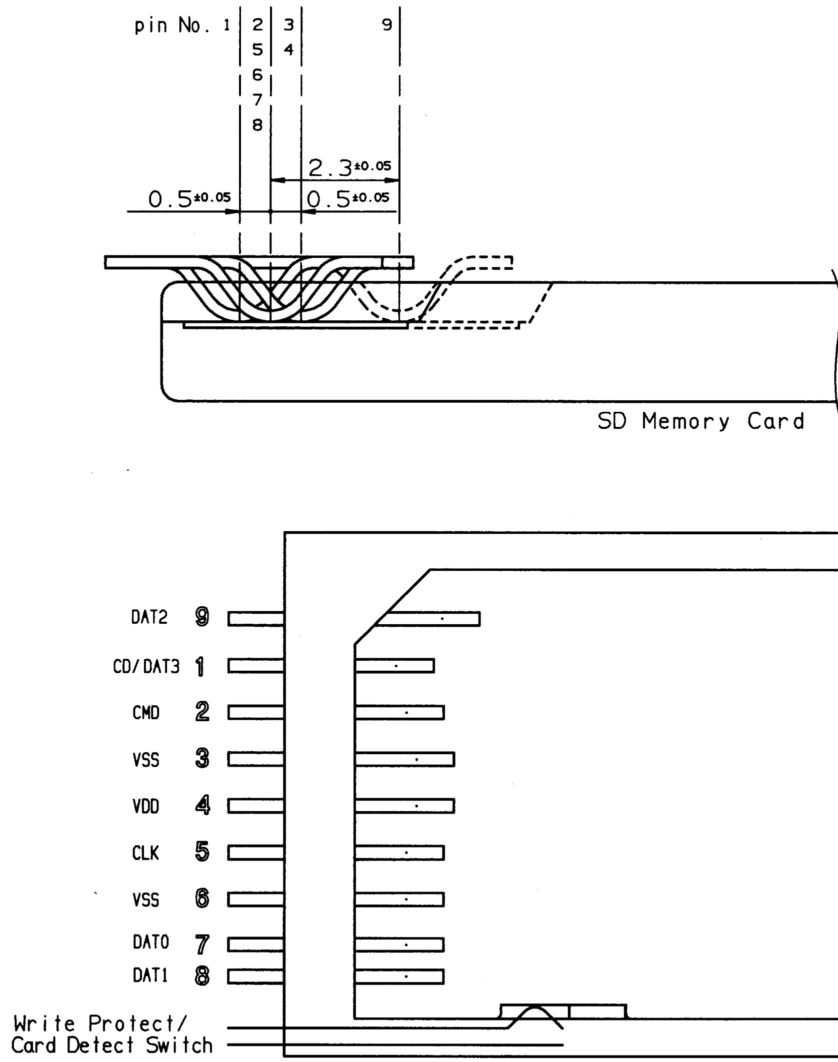


Figure 8-10: Modified SD Memory Card Connector for Hot Insertion

A.2.3 Characteristics

The features described in the following shall be considered when designing an SD Memory Card connector. The given values are typical examples.

- Mechanical Characteristics

- Max. number of mating operations	> 10000	
- Contact force	0.2 N (minimum contact force per one contact)	
- Total pulling force	min.1 N DIN IEC 512 part 7	
- Total insertion force	max. 40 N	DIN IEC 512 part 7
- Vibration and High Frequency		
- Mechanical frequency range	10.....2000 Hz	DIN IEC 512 part 2 and 4
- Acceleration	2 g	
- Shock:		
- Acceleration	5 g	

- Electrical Characteristics
  DIN IEC 512

- Contact resistance	100 mΩ
- Current carrying capacity at 25°C	0.5 A
- Insulation resistance	> 1000 MΩ> 100 MΩ after test
- Operating voltage	3.3 V
- Testing voltage	500 V
- Operating current	100 mA max.

- Climatic Characteristics
  DIN IEC 512 part 6-9

- Operating temperature	-25°C.....90°C
- Storage temperature	-40°C.....90°C
- Humidity	95% max. non condensing

A.3 Related Documentation

- miniSD Memory Card Specification
- microSD Memory Card Specification
- Supplementary Notes for SD Physical Specification
- Supplementary Notes for miniSD Memory Card Specification
- Application Notes Relating to SD Physical Specification
- Application Notes Relating to miniSD Physical Specification
- Speed Class Implementation Guideline
- SD Specifications Part2 File System Specification
- SD Specifications Part3 Security Specification

Appendix B

B.1 Abbreviations and terms

block	a number of bytes, basic data transfer unit
broadcast	a command sent to all cards on the SD bus
Blocklen	Block Length set by CMD16
CID	Card IDentification number register
CLK	clock signal
CMD	command line or SD bus command (if extended CMDXX)
CRC	Cyclic Redundancy Check
CSD	Card Specific Data register
DAT	data line
DSR	Driver Stage Register
ECC	Error Correction Code
Flash	a type of multiple time programmable non volatile memory
group	a number of sectors, composite erase and write protect unit
LOW, HIGH	binary interface states with defined assignment to a voltage level
NSAC	defines the worst case for the clock rate dependent factor of the data access time
MSB, LSB	the Most Significant Bit or Least Significant Bit
MTP	Multiple Time Programmable memory
OCR	Operation Conditions Register
open-drain	a logical interface operation mode. An external resistor or current source is used to pull the interface level to HIGH, the internal transistor pushes it to LOW
OTP	One Time Programmable memory
payload	net data
push-pull	a logical interface operation mode, a complementary pair of transistors is used to push the interface level to HIGH or LOW
RCA	Relative Card Address register
ROM	Read Only Memory
sector	a number of blocks, basic erase unit
stuff bit	filling bits to ensure fixed length frames for commands and responses
SPI	Serial Peripheral Interface
TAAC	defines the time dependent factor of the data access time
tag	marker used to select groups or sector to erase
TBD	To Be Determined (in the future)

three-state driver	a driver stage which has three output driver states: HIGH, LOW and high impedance (which means that the interface does not have any influence on the interface level)
token	code word representing a command
V_{DD}	+ power supply
V_{SS}	power supply ground
AU	Allocation Unit
RU	Recording Unit
P_w	Performance of Write
P_m	Performance of Move
P_r	Performance of Read
T_{fw}	FAT write time
T_{fr}	FAT read time
VHS	Host supplied voltage range
VCA	Card accepted voltage range
N_{ERASE}	Recommended number of AUs to be erased in one erase operation.
T_{ERASE}	Timeout value used for erasing multiple AU's as specified by ERASE_SIZE.
T_{OFFSET}	Offset time used for calculating erase timeout.

- This page was left empty intentionally -