SMSxxxDF

512-Mbyte, 1-Gbyte and 2-Gbyte 3.3 V supply Secure Digital™ card

Features

- SD memory card specification version 2.00-standard
- Up to 2 Gbytes of formatted data storage
- Bus mode
  - SD protocol (1 to 4 data lines)
  - SPI protocol
- Operating voltage range:
  - Commands and memory access: 2.7 V to 3.6 V
- Variable clock rate: 0 to 25 MHz
- Read access (using 4 data lines)
  - Sustained multiple block: up to 16 Mbytes/s
- Write access (using 4 data lines)
  - Sustained multiple block: up to 9 Mbytes/s
- Maximum data rate with up to 10 cards
- Aimed at portable and stationary applications
- Communication channel protocol attributes:
  - Six-wire communication channel (clock, command, 4 data lines)
  - Error-proof data transfer
  - Single or multiple block oriented data transfer

MicroSD

- Memory field error correction
- Safe card removal during read
- Write protect feature using mechanical switch
- Built-in write protection features (permanent and temporary)
- MicroSD packages
  - ECOPACK® compliant
  - Halogen free
  - Antimony free

Table 1. Device summary

Part number	Speed class	Package form factor	Operating voltage range
SMS512DF	2	MicroSD	2.7 V to 3.6 V
SMS01GDF	2		
SMS02GDF	4		

Contents

List of tables

List of figures

1 Description

The secure digital memory card (SD memory card) is a flash-based memory card. It is specifically designed to meet the security, capacity, performance and environmental requirements of the latest-generation audio and video consumer electronic devices, that is mobile phones, digital cameras, digital recorders, PDAs, organizers, electronic toys, etc.

The secure digital memory card is a high-mobility, high-performance, low-cost and low-power consumption device that features high data throughput at the memory card interface. It includes a copyright protection mechanism that complies with the security of the SDMI (Secure Digital Music Initiative) standard. The secure digital memory card security system uses mutual authentication and a ‘cipher algorithm’ that protects the card from illegal use. Unsecured access to the user's personal content is also available.

The secure digital memory cards have an advanced communication interface designed to operate in a low voltage range. The full-size secure digital memory card has a 9-pin interface whereas the mini secure digital memory card has a 11-pin interface but can be fitted with a 9-pin adapter. Only the 9-pin interface is described in this document. The MicroSD memory card has an 8-pin interface, and can also be fitted with a 9-pin adapter.

Table 2, *Table 3*, *Table 4*, *Table 5*, and *Table 6* give an overview of the secure digital memory card features.

The microSD packages are also halogen free and antimony free.

Related documentation

- Secure digital memory card specifications: part 1 physical layer specification, version 2.00
- MicroSD memory card specifications: addendum to SD memory card specifications part 1, physical layer specification, version 2.01

Table 2. System performance

System performance	512 Mbytes		1 Gbyte		2 Gbytes		Unit
	Max	Typ	Max	Typ	Max	Typ	
Sleep to ready	30		30		30		µs
Sustained multiple block read(1)		16 (107X)		16 (107X)		16 (107X)	Mbytes/s
Burst single block read(1)		1.8 (12X)		1.8 (12X)		1.8 (12X)	Mbytes/s
Sustained multiple block write(1)		5 (34X)		5 (34X)		9 (60X)	Mbytes/s
Burst single block write(1)		0.3 (2X)		0.3 (2X)		0.3 (2X)	Mbytes/s
Power-up to ready	200		200		200		ms

1. 107X, 60X, 34X, 12X and 2X speed grade markings where 1X = 150 Kbytes/s.

Table 3. Power consumption(1)

Mode	Max. current consumption
Standby	60 µA
Read	40 mA
Write	40 mA

1. T_A = 25 °C, V_{DD} = 3.6 V.

Table 4. Environmental specifications(1)

Environmental specifications		Operating	Non-operating
Temperature		– 25 °C to 85 °C	– 40 °C to 85 °C
Humidity (non-condensing)		NA	85 °C - 85% RH
ESD protection	Contact pads	NA	±4 kV, human body model according to ANSI EOS/ESD-S5.1-1998
	Other		±8 kV (coupling plane discharge)
			±15 kV (air discharge) human body model per IEC61000-4-2
Salt water spray		NA	T_A = 35 °C 3% NaCl (MIL Std method 1009)
Vibration (peak-to-peak)		NA	15 Gmax
Shock		NA	1,000G
Drop		NA	2,000G
Bending			20N (middle of the card)
			20N (border of the card)
UV light exposure			254 nm, 15 Ws/cm^2

1. NA = not applicable; RH = relative humidity; ESD = electrostatic discharge

Table 5. Physical dimensions

Parameter	MicroSD	Unit
Width	11	mm
Height	15	mm
Thickness	Inter connect area 0.7±0.1	mm
	Max. card thickness 0.95	
	Max. pull area 1.0±0.1	
Weight	<1	g
Number of pins	8	N/A

Table 6. System reliability and maintenance

MTBF(1)	>1,000,000 hrs
Preventive maintenance	None
Data reliability	1 non-recoverable bit in 10^{14} bit read

1. MTBF = mean time between failures.

2 Memory array partitioning

The basic unit of data transfer to/from the SD memory card is the byte. The memory array is divided into several structures as described below and summarized in *Table 7: Memory array structures*.

2.1 Block

The block is the unit structure related to block-oriented read and write commands. Its size is the number of bytes that are transferred when a block-oriented read or write command is sent by the host. The SD memory card block size is either programmable or fixed. The information about allowed block sizes and programmability is stored in the CSD register. The details of the memory array structure and the number of addressable blocks are shown in *Table 7: Memory array structures*.

2.2 Sector

The sector is the unit structure related to the erase commands. Its size is the number of blocks that are erased at any one time. The sector size is fixed for each device. The information about the sector size (in blocks) is stored in the CSD register.

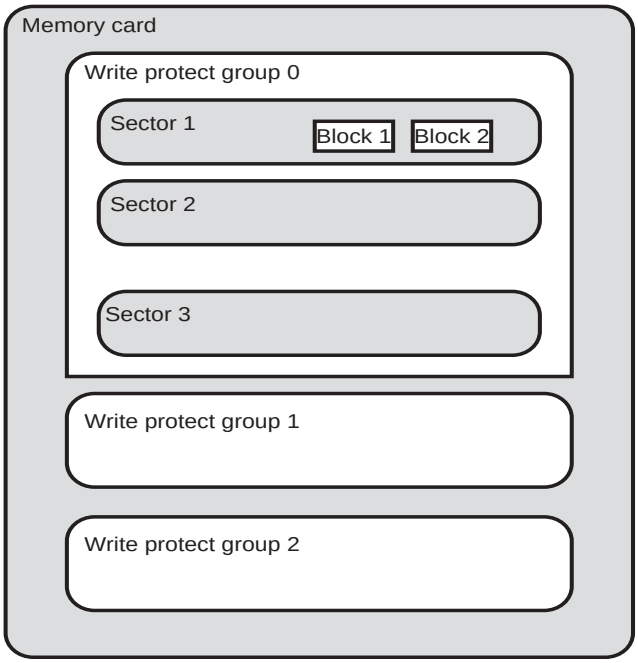
2.3 Write protect group (WP-group)

The WP-group is the smallest structure that may be individually protected. Its size is the number of sectors that are write protected with one bit. The information about the write protect group size is stored in the CSD register.

Table 7. Memory array structures

Type of structure	Number of structures in 512-Mbyte, 1-Gbyte devices			Number of structures in 2-Gbyte devices	
	Unit	512-Mbyte	1-Gbyte	Unit	2-Gbyte
Blocks	512 bytes	978944	1999872	1024 bytes	1960448
Sector	Block	128	128	Block	128
WP-groups	Sector	16	32	Sector	64
Protected area	Block	10240	28672	Block	20480

Figure 1. Write protection hierarchy



3 Secure digital memory card interface

This section applies to the MicroSD card when used with an adapter.

The secure digital memory card has an advanced 9-pin communication interface (clock, command, 4 data pins and 3 power supply pins) designed to operate in a low voltage range. The secure digital card has its nine pins exposed on one side (see *Figure 2*). The signal/pin assignments are listed in *Table 8*. The pin types are power supply, input, output and push-pull. The signals include six communication lines CMD, DAT0, DAT1, DAT2, DAT3, CLK and three supply lines V_{DD}, V_{SS1} and V_{SS2}.

Figure 2. Full size secure digital memory card form factor

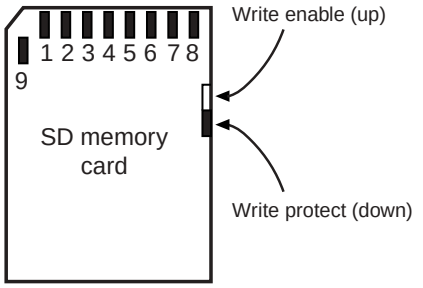


Table 8. Full-size SD memory card pin assignment

Pin #	SD mode			SPI mode		
	Name	Type(1)	Description	Name	Type	Description
1	CD/DAT3(2)	I/O/PP(3)	Card detect / data line [bit 3]	CS	I	Chip select (active Low)
2	CMD	PP	Command/response	DI	I	Data in
3	V_{SS1}	S	Supply voltage ground	V_{SS}	S	Supply voltage ground
4	V_{DD}	S	Supply voltage	V_{DD}	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS2}	S	Supply voltage ground	V_{SS2}	S	Supply voltage ground
7	DAT0	I/O/PP	Data line [bit 0]	DO	O/PP	Data out
8	DAT1(2)	I/O/PP	Data line [bit 1]	Reserved		
9	DAT2(2)	I/O/PP	Data line [bit 2]	Reserved		

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.
3. After power-up this line is input with 50 kOhm pull-up (can be used for card detection or SPI mode selection). The pull-up should be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command.

Figure 3. MicroSD pin assignment

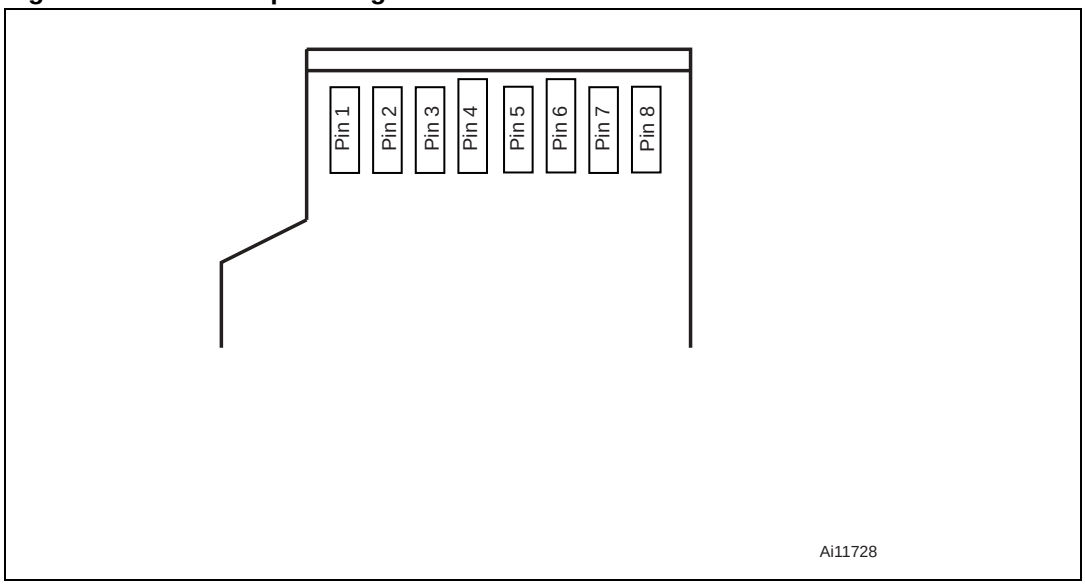


Table 9. MicroSD contact pad assignment

Pin	SD mode			SPI mode		
	Name	Type(1)	Description	Name	Type	Description
1	DAT2	I/O/PP	Data line [bit 2]	RSV		Reserved
2	CD/DAT3(2)	I/O/PP (3)	Card detect / data line [bit 3]	CS	I	Chip select (neg true)
3	CMD	PP	Command/response	DI	I	Data in
4	V_{DD}	S	Supply voltage	V_{DD}	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V_{SS}	S	Supply voltage ground	V_{SS}	S	Supply voltage ground
7	DAT0	I/O/PP	Data line [bit 0]	DO	O/PP	Data out
8	DAT1			RSV		Reserved

1. S: power supply; I: input; O: output using push-pull drivers; PP: I/O using push-pull drivers.
2. The extended DAT lines (DAT1-DAT3) are input on power up. They start to operate as DAT lines after SET_BUS_WIDTH command. The Host shall keep its own DAT1-DAT3 lines in input mode, as well, while they are not used. It is defined so, in order to keep compatibility to MultiMediaCards.
3. After power up this line is input with 50 KOhm pull-up (can be used for card detection or SPI mode selection). The pull-up should be disconnected by the user, during regular data transfer, with SET_CLR_CARD_DETECT (ACMD42) command.

3.1 Secure digital memory card bus topology

The secure digital memory card system defines two alternative communications protocols: SD and SPI that correspond to two operating modes.

Either mode can be selected in the application, mode selection is transparent to the host. The host automatically detects the operating mode of the card by issuing the Reset command (refer to *Section 7.2.1: Mode selection*) and will expect all further communications to use the same mode. Therefore, applications that use only one communication mode do not have to be aware of the other.

The SD bus includes the following signals:

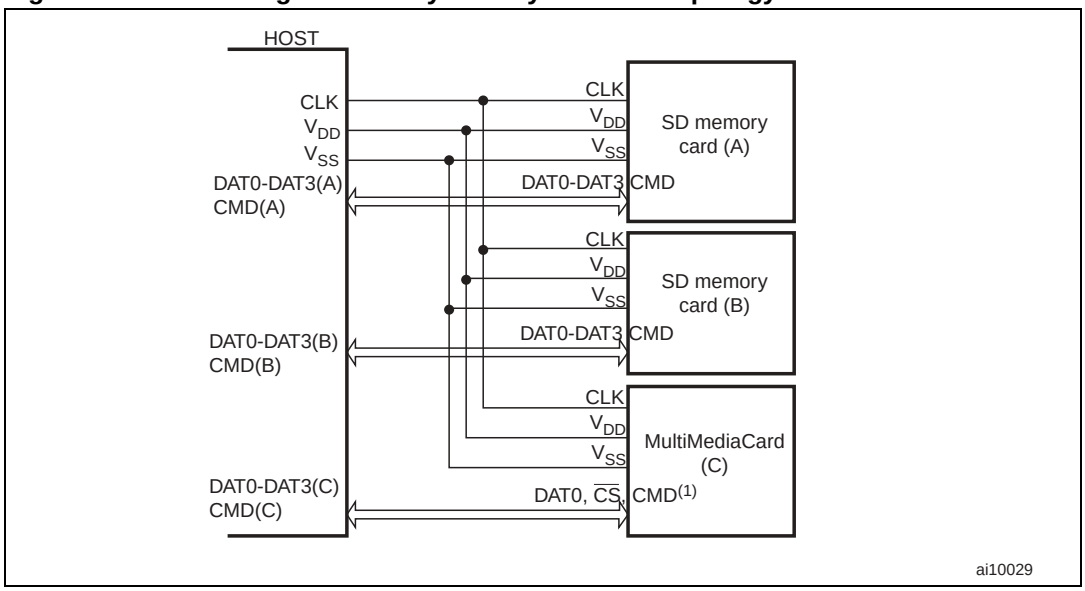
- CLK: host to card clock signal
- CMD: bi-directional command/response signal
- DAT0 - DAT3: 4 bi-directional data signals.
- V_{DD}, V_{SS1}, V_{SS2}: power and ground signals.

The SD memory card bus has a synchronous star topology (refer to *Figure 4: Secure digital memory card system bus topology*) with a single master (the application) and multiple slaves (the cards). The Clock, power and ground signals are common to all cards. The command (CMD) and data (DAT0 - DAT3) signals are dedicated to the cards, they provide continuous point-to-point connection to all the cards.

During the initialization process, commands are sent to each card individually, allowing the application to detect the cards and assign logical addresses to the physical slots. Data is always sent (received) to (from) each card individually. However, in order to simplify 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.

The SD bus allows dynamic configuration of the number of data lines. After power-up the SD memory card defaults to using only DAT0 for data transfer. After initialization the host can change the bus width (number of active data lines). This feature is an easy trade off between hardware cost and system performance.

Figure 4. Secure digital memory card system bus topology



1. DAT1 and DAT2 not connected.

3.2 SD bus protocol

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

- **Command:** a command is a token which starts an operation. A command is sent from the host either to a single card (addressed command) or to all connected cards (broadcast command). Commands are transferred serially on the CMD line. See *Figure 5: 'No response' and 'no data' operations*.
  The command token format is shown in *Figure 8*.
- **Response:** a response is a token which is sent from an addressed card, or (simultaneously) from all connected cards, to the host, as an answer to a previously received command. Responses are transferred serially on the CMD line. A response is illustrated in *Figure 5: 'No response' and 'no data' operations*.
  The response token format is shown in *Figure 9*.
- **Data:** data can be transferred from the card to the host or from the host to the card. Data is transferred via the data lines. See *Figure 6: (Multiple) block read operation* for an illustration.
  The data packet format is shown in *Figure 10*.

Card addressing is implemented using a session address assigned to the card during the initialization phase (see *Section 4: SD memory card hardware interface*). The basic transaction on the SD bus is the command/response transaction. In this type of bus transactions, the information is directly transferred 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 followed by CRC bits.

Single and multiple block operations are supported. Note that the multiple block operation mode improves the speed of write operations. A multiple block transmission is terminated by issuing a STOP_TRANSMISSION command on the CMD line (see *Figure 6: (Multiple) block read operation* and *Figure 7: (Multiple) block write operation*).

Data transfer can be configured by the host to use single or multiple data lines (provided that the card supports this feature).

A busy signal on DAT0 is used to indicate that a block write operation is ongoing (see *Figure 7: (Multiple) block write operation*). The same busy signaling is used regardless of the number of data lines used to transfer the data.

Response tokens (see *Figure 9: Response token format*) have four coding schemes depending on their content. The token length is either 48 or 136 bits (see SD memory card specification, chapters 4.7 and 4.9, for detailed definitions of the commands and responses). The CRC protection algorithm for data block is a 16-bit CCITT polynomial (see SD memory card specification, chapter 4.5).

On the CMD line, the MSB bit is transmitted first and the LSB bit last. When the wide bus option is used, the data is transferred 4 bits at a time (refer to *Figure 10*). Start bits, end bits and CRC bits, are transmitted on all the DAT lines used. CRC bits are calculated and checked for every DAT line individually. The CRC status response and busy indication are sent by the card to the host on DAT0 only (DAT1-DAT3 are don't care).

Figure 5. 'No response' and 'no data' operations

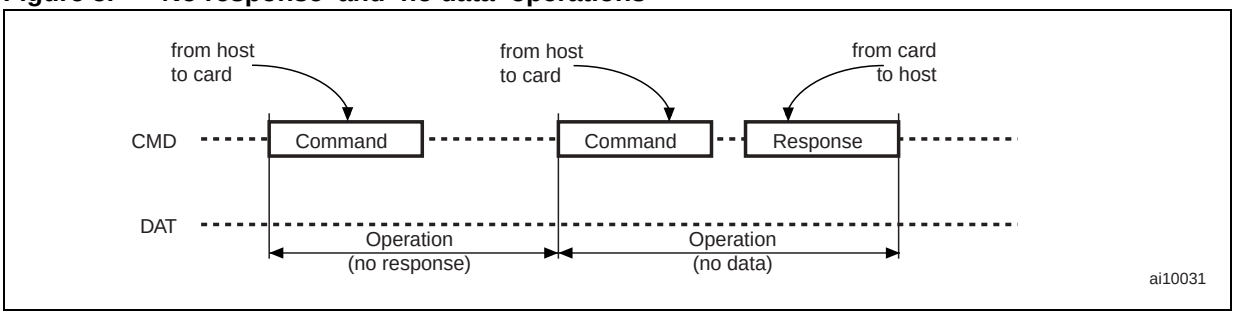


Figure 6. (Multiple) block read operation

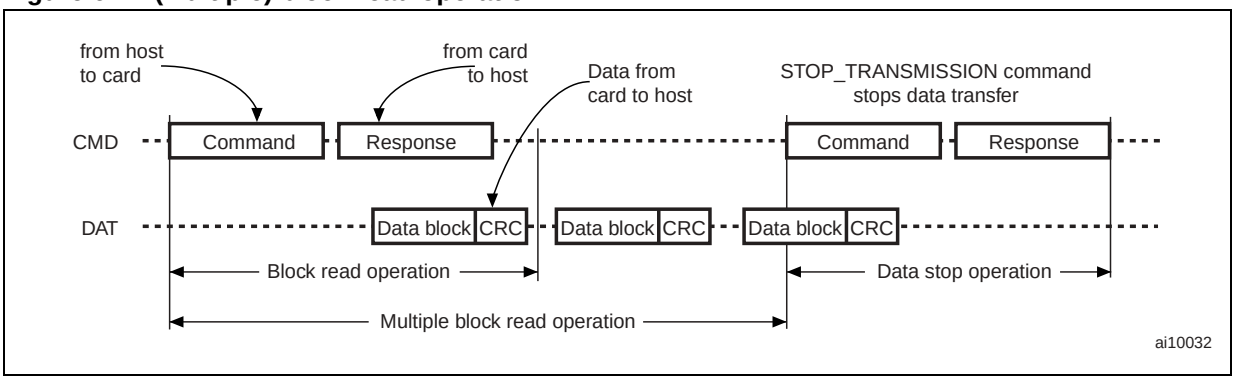


Figure 7. (Multiple) block write operation

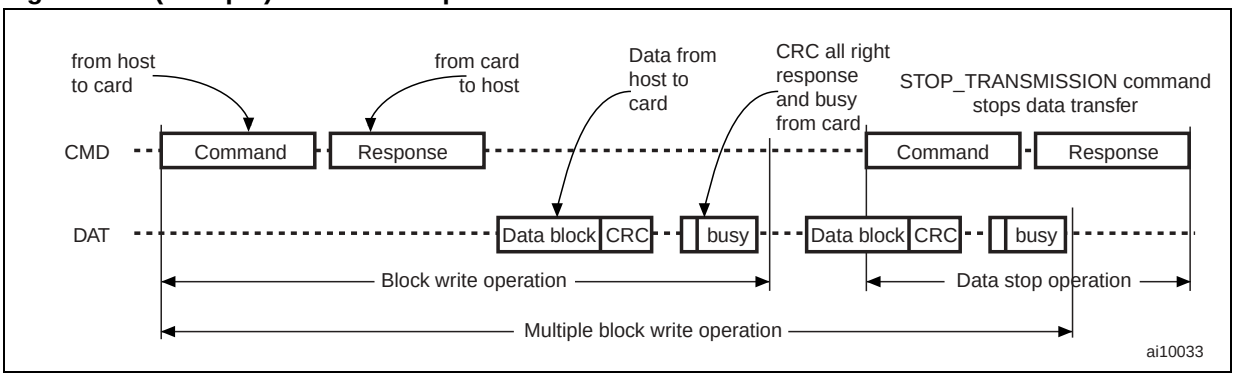


Figure 8. Command token format

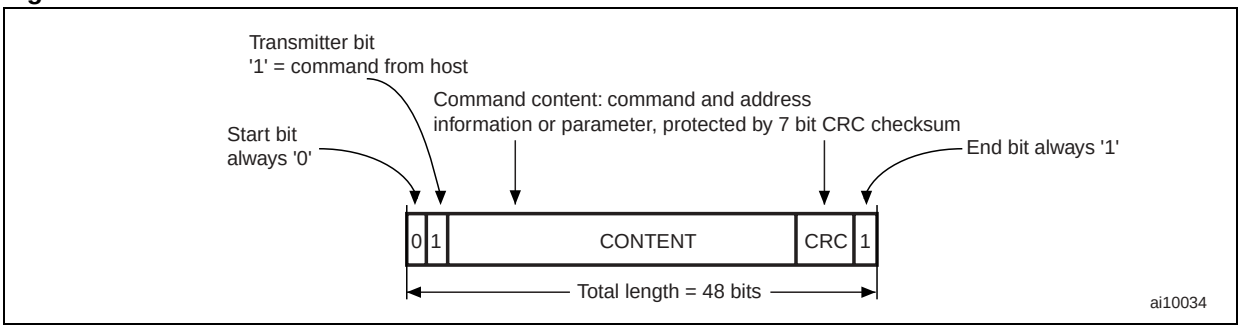


Figure 9. Response token format

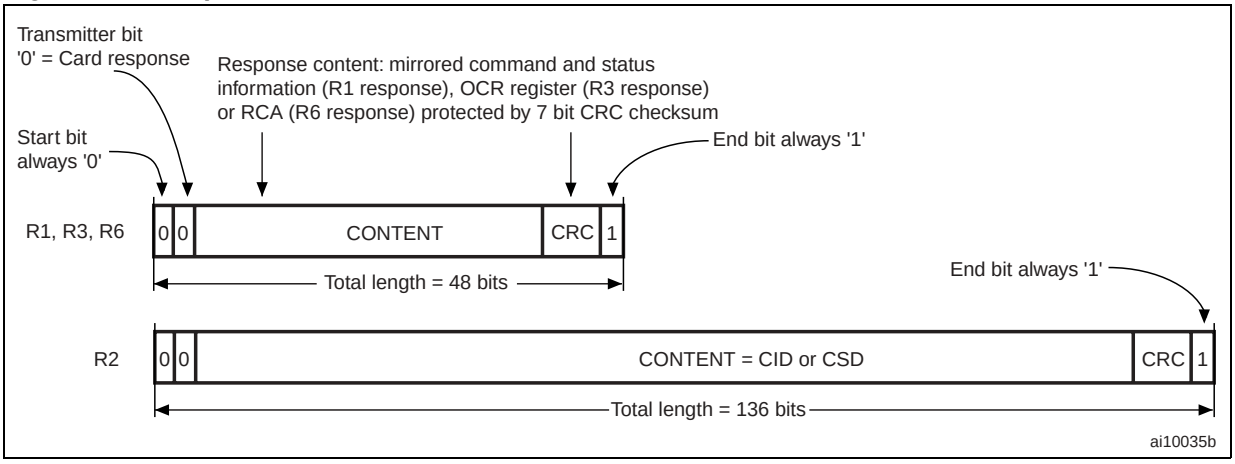
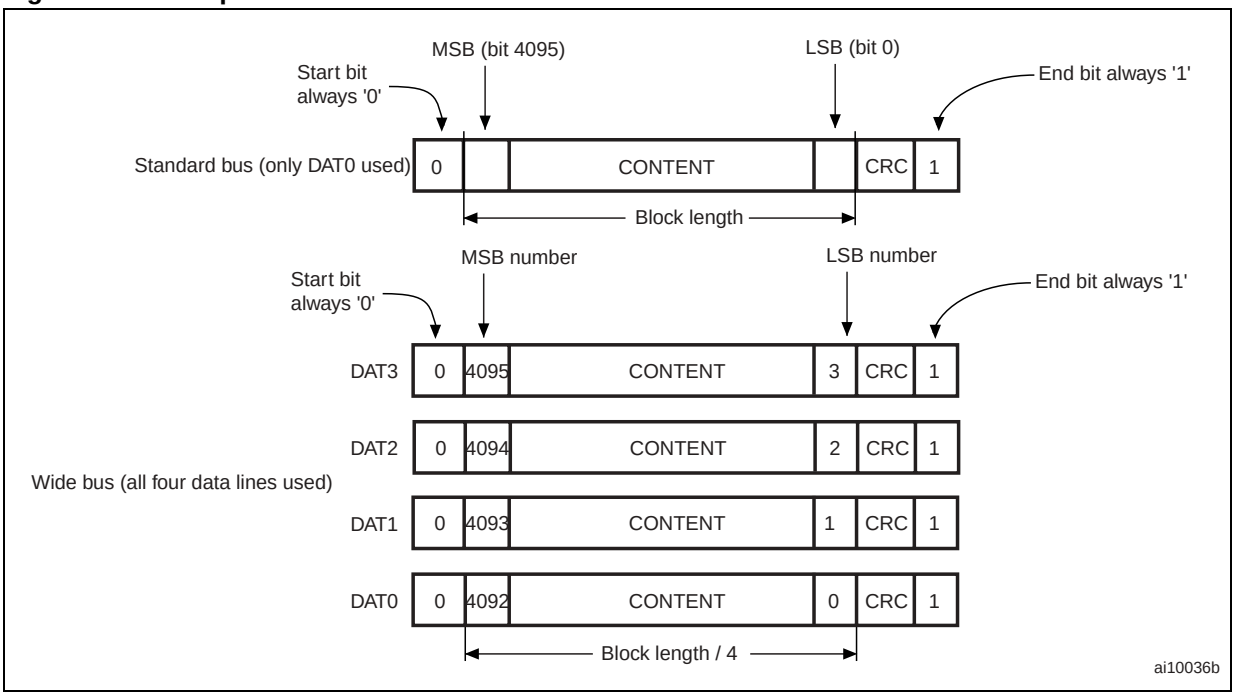


Figure 10. Data packet format



3.3 SD memory card functional description

All communications between the host and the cards are controlled by the host (master).

The host sends commands of two types:

- **Broadcast commands** which are intended for all cards. Some of these commands require a response.
- **Addressed (point-to-point) commands** that are sent to the addressed card and are followed by a response from the card.

3.4 Operation modes

Figure 11 and *Figure 12* show an overview of the command flow for the card identification mode and the data transfer mode, respectively.

Table 10 shows the relationship between operation modes and card states. Each state in the SD memory card state diagram (see *Figure 16* and *Figure 18*) is associated with one operation mode.

Table 10. Card states vs. operation modes

Card state	Operation mode
Inactive state	Inactive
Idle state	Card identification mode
Ready state	
Identification state	
Standby state	Data transfer mode
Transfer state	
Sending-data state	
Receive-data state	
Programming state	
Disconnect state	

3.4.1 Card identification mode

The host enters the card identification mode after reset and remains in this mode until it has finished searching for new cards on the bus.

Cards enter the card identification mode after reset and remain in this mode until they receive the SEND_RCA command (CMD3) (or the SET_RCA command for MultiMediaCards).

While in card identification mode the host resets all the cards that are in card identification mode, validates the operation voltage range, identifies every card and asks them to publish their relative card addresses (RCA). This operation is done separately for each card on its own CMD line. In this mode, all data communications use the command line (CMD) only.

The host starts the card identification process at the identification clock rate f_{OD}. The SD memory card has push-pull CMD line output drives.

When the communication between the host and the card starts, the host may not know which voltage is supported by the card 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. Then, the SEND_IF_COND (CMD8) is used to verify the SD memory card interface operating condition. In particular, the card checks the validity of operating condition by analyzing the argument of CMD8 while the host checks the validity by analyzing the response of CMD8.

The receipt of CMD8 makes the cards realize that the host supports the physical layer version 2.00 and that the card can enable new functions. It is mandatory to send CMD8 before ACMD41 for the low-voltage host. If a dual voltage card does not receive CMD8, it will work as a high-voltage only card and will go to inactive state at ACMD41.

SD_SEND_OP_COND (ACMD41) is designed to provide SD memory card hosts with a mechanism to identify and reject cards that 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 SD memory card specification, chapter 5.1). Cards which can not 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 SD memory card specification, chapter 5.1). Note that ACMD41 is an

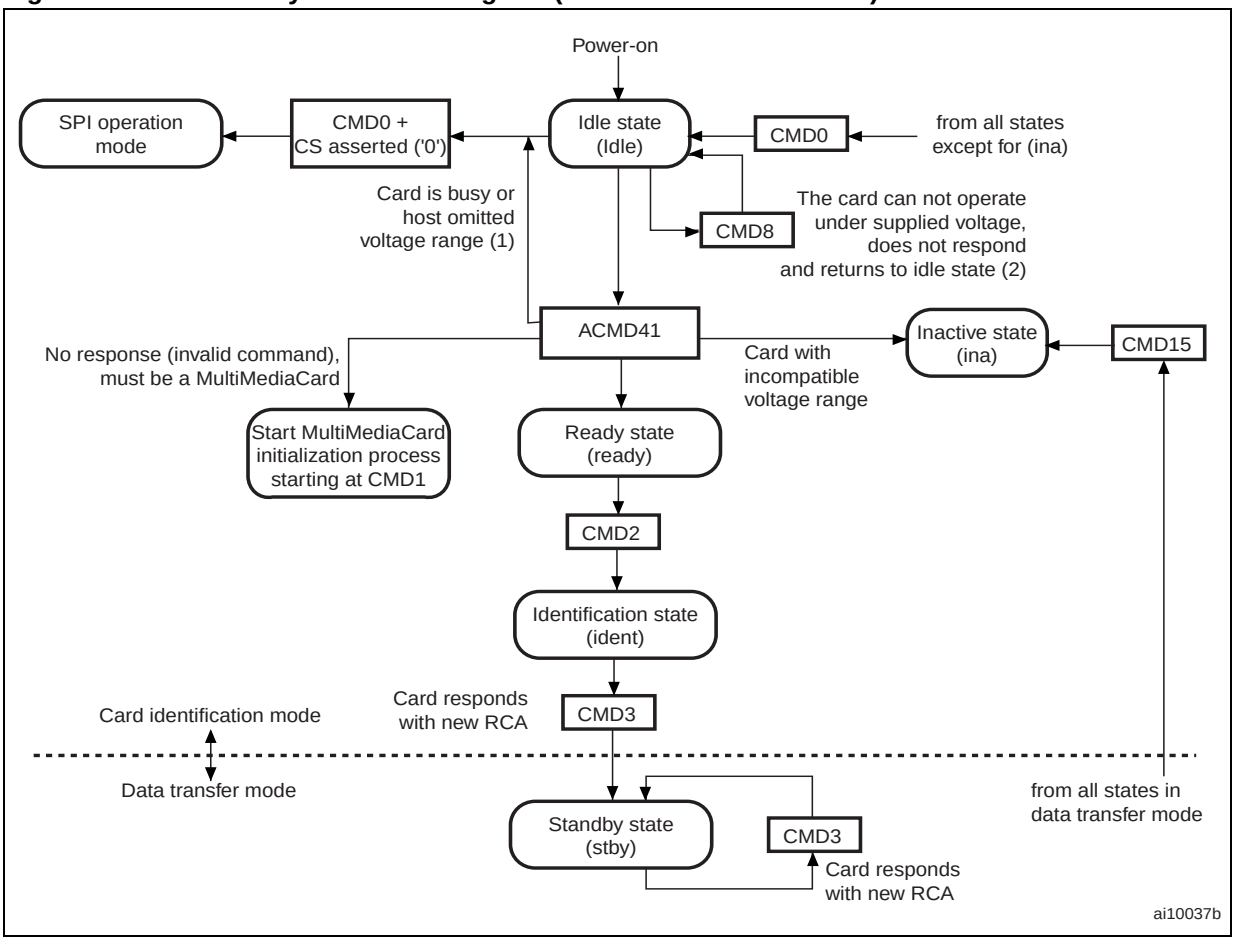
application specific command, therefore APP_CMD (CMD55) shall always precede ACMD41. The RCA to be used for CMD55 in idle state shall be the card's default RCA = 0x0000.

After the host issues a reset command (CMD0) to reset the card, the host shall issue CMD8 before ACMD41 to re-initialize the SD memory card.

The host then issues the ALL_SEND_CID command (CMD2), to every card to get their unique card identification (CID) numbers. All unidentified cards (which are in ready state) answer by sending their CID numbers (on the CMD line) and switch to the identification state. Then the host issues a CMD3 (SEND_RELATIVE_ADDR) command to ask the cards to publish a relative card address (RCA). The RCA is shorter than the CID, and will be used to address the card (typically at a clock rate higher than f_{OD}) once this is in data transfer mode. Once the RCA is received the card state changes to standby. At this point, the host may ask the card to publish another RCA number by sending another SEND_RELATIVE_ADDR command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process, that is the cycles with CMD2 and CMD3, for each card in the system. Once all the SD memory cards have been initialized, the host initializes the MultiMediaCards that are in the system (if any) by issuing CMD2 and CMD3 as explained in the MultiMediaCard specification. Note that in the SD system all the cards are connected separately so each MultiMediaCard has to be initialized individually.

Figure 11. SD memory card state diagram (card identification mode)



1. The card returns busy when:
   - the card executes an internal initialization process
   - the card is an high capacity SD memory card and the host does not support high capacity
   This means that CMD8 is mandatory to initialize high capacity SD memory card.
2. For the host compliant to physical specifications version 2.00, it is mandatory to send CMD8 before ACMD41.

3.4.2 Data transfer mode

Cards enter the data transfer mode once their relative card addresses (RCA) have been published.

The host enters the data transfer mode after identifying all the cards on the bus.

The host issues SEND_CSD (CMD9) to obtain the contents of the card specific data (CSD) register for each card. The CSD register contains information like the block length and the card storage.

Until the host knows the contents of all the CSD registers, the f_{PP} clock rate must remain at f_{OD} because some cards may have operating frequency limitations.

The broadcast command SET_DSR (CMD4) configures the driver stages of all identified cards. It programs their driver stage registers (DSR) according to the application bus layout (length), the number of cards on the bus and the data transfer frequency. The clock rate is changed from f_{OD} to f_{PP} at that point. The SET_DSR command is an option for the card and the host.

CMD7 is used to select one card and switch it to the transfer state. Only one card can be in transfer state at a given time. If a previously selected card is still in transfer state when the host uses CMD7 to switch another card to the transfer state, then the connection between the previously selected card and the host is released and the card reverts to the standby state.

When CMD7 is issued with the reserved relative card address ‘0000h’, all cards revert to the standby state. This function may be used before identifying new cards, to avoid resetting already registered cards. When in standby state the cards that already have an RCA do not respond to identification commands (CMD41, CMD2, CMD3).

Note that a card is deselected when it receives a CMD7 with an RCA that does not match. Card deselection is automatic if another card in a system is selected and the cards share the same CMD lines.

So, in an SD memory card system, the host may either have a common CMD line for all SD memory cards (in which case card deselection is automatic just like in a MultiMediaCard system) or the host may have separate CMD lines, in which case it must be aware of the necessity of deselecting cards.

All data communications in the data transfer mode are point-to point between the host and the selected card (using addressed commands). All addressed commands are acknowledged by a response on the CMD line.

The relationships between the various states in the data transfer mode are summarized below (see *Figure 12*):

- All Data Read commands (CMD17, CMD18, CMD30, CMD56, ACMD51) can be aborted at any time using the Stop command (CMD12). The data transfer will terminate and the card will return to the transfer state
- All Data Write commands (CMD24,CMD25, CMD26, CMD27, CMD42, CMD56) can be aborted at any time using the Stop command (CMD12). The write commands must be stopped prior to deselecting the card using CMD7
- As soon as the data transfer has completed, the card switches from the data write state to either the programming state (if the transfer was successful) or the transfer state (if the 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 can provide buffering during block write. This means that the data to be programmed to the next block can be sent to the card while the previous block is being programmed.

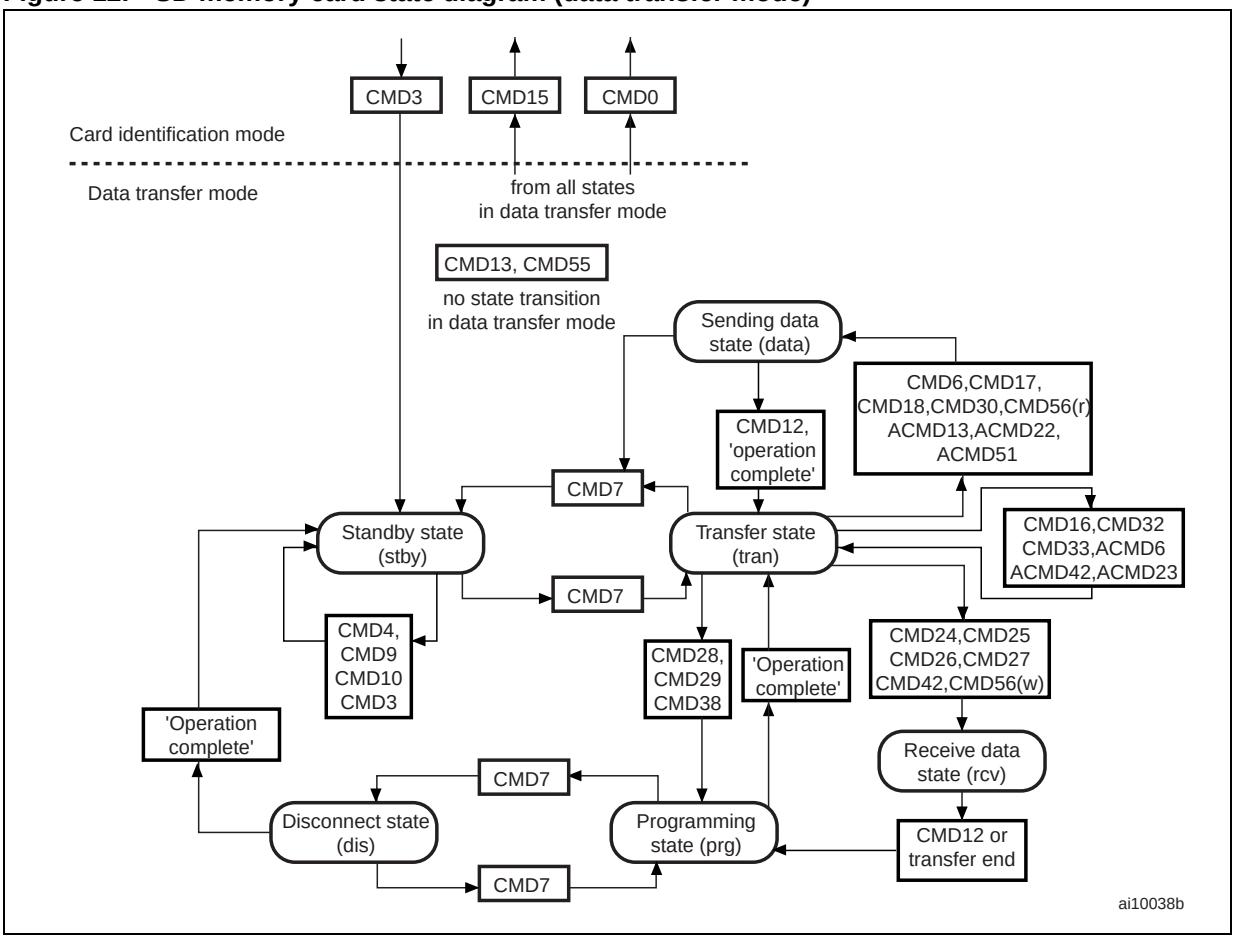
If all write buffers are full, the DAT0 line will remain Low (busy) as long as the card is in the programming state (see *Figure 12*).

- There is no buffering option for write CSD, write CID, write protection and erase. This means that while the card is busy with any one of these commands, no other Data

Transfer command will be accepted. The DAT0 line will remain Low as long as the card is busy and in the programming stat.

- Parameter Set commands (CMD16, CMD32, CMD33) are not allowed while the card is programming
- Read commands are not allowed while the card is programming
- Switching another card from the standby to the transfer state (using CMD7) will not terminate erase and programming operations. The card will switch to the disconnect state and release the DAT line.
- A card in the disconnect state can be reselected using CMD7. The card will then revert to the programming state and reactivate the busy signaling
- Resetting a card (using CMD0 or CMD15) will terminate any pending or ongoing programming operation. This may result in the loss of card contents. It is up to the host to prevent possible data loss.

Figure 12. SD memory card state diagram (data transfer mode)



3.5 Commands

Four types of commands are used to control the SD memory card:

1. **Broadcast commands (bc), no response:** the broadcast feature is available only if all the CMD lines are interconnected at the level of the host. If they are not interconnected then each individual card will accept the command in turn
2. **Broadcast commands with response (bcr):** since there is no open drain mode in SD memory cards, this type of command is used only if the host does not use a common CMD line. The command is accepted by every individual card and the responses from all cards are sent simultaneously
3. **Addressed (point-to-point) commands (ac):** there is no data transfer on DAT.
4. **Addressed (point-to-point) data transfer commands (adtc):** there is a data transfer on DAT.

All commands have a fixed code length of 48 bits for a transmission time of 1.92 µs at 25 MHz and 0.96 µs at 50 MHz. All commands and responses are sent over the CMD line of the SD memory card. Command transmission always starts with the most significant bit (MSB) of the command codeword. All commands are protected by a CRC. All command codewords are terminated by the end bit (always '1'). *Table 11* shows the command format. All commands and their arguments are specified in the SD memory card specification.

Table 11. SD card command format

Bit position	47	46	45:40	39:8	7:1	0
Width	1	1	6	32	7	1
Value	‘0’	‘1’	x	x	x	'1'
Description	Start bit	Transmission bit	Command index	Argument	CRC7	End bit

3.6 Responses

All responses are sent via the command line CMD. Response transmission always starts with the leftmost bit of 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 (from card = '0').

A value denoted by 'X' in *Table 12*, *Table 13*, *Table 14* and *Table 15* indicates a variable entry.

All responses (except for R3 responses) are protected by a CRC. All response codewords are terminated by the end bit (always '1').

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

1. **R1 (normal response command):** the code length is 48 bits. Bits 45 to 40 indicate the index of the command to respond to. The index is a binary coded number (between 0 and 63). The status of the card is coded in 32 bits (see *Table 12*).

   Note that if data transfer to the card takes place, then a busy signal may appear on the data line after the transmission of each block of data. The host has to check for busy after data block transmission

2. **R1b** is identical to R1 with an optional busy signal transmitted on the data line. The card may become busy after receiving these commands, depending on the state it was in prior to receiving the command. The host has to check for busy in the response
3. **R2 (CID, CSD register):** the code length is 136 bits. The contents of the CID register are sent as a response to the CMD2 and CMD10 commands. The contents of the CSD register are sent as a response to CMD9. Only the bits [127...1] of the CID and CSD registers are transferred, the reserved bit [0] of these registers is replaced by the end bit of the response (see *Table 13*)
4. **R3 (OCR register):** the code length is 48 bits. The contents of the OCR register are sent as a response to ACMD41 (see *Table 14: Response R3*).
5. **R6 (published RCA response):** the code length is 48 bits. Bits 45 to 40 indicate the index of the command to respond to. In this case it is '000011' (together with bit 5 in the status bits it means = CMD3) as shown in *Table 15: Response R6*. The 16 MSB bits of the argument field are used for the published RCA number
6. **R7(card interface condition):** the code length is 48 bits. The card support voltage information is sent by the response of CMD8. Bits 19 to 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.

For more details about response formats, please refer to the SD memory card specification.

Table 12. Response R1

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	‘0’	‘0’	X	X	X	‘1’
Description	Start bit	Transmission bit	Command index	Card status	CRC7	End bit

Table 13. Response R2

Bit position	135	134	[133:128]	[127:1]	0
Width (bits)	1	1	6	127	1
Value	‘0’	‘0’	‘111111’	X	‘1’
Description	Start bit	Transmission bit	Reserved	CID or CSD register incl. internal CRC7	End bit

Table 14. Response R3

Bit position	47	46	[45:40]	[39:8]	[7:1]	0
Width (bits)	1	1	6	32	7	1
Value	‘0’	‘0’	‘111111’	X	‘111111’	‘1’
Description	Start bit	Transmission bit	Reserved	OCR register	Reserved	End bit

Table 15. Response R6

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 and 12 to 0	CRC7	End bit

Table 16. Response R7

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

4 SD memory card hardware interface

4.1 SD memory card bus circuitry

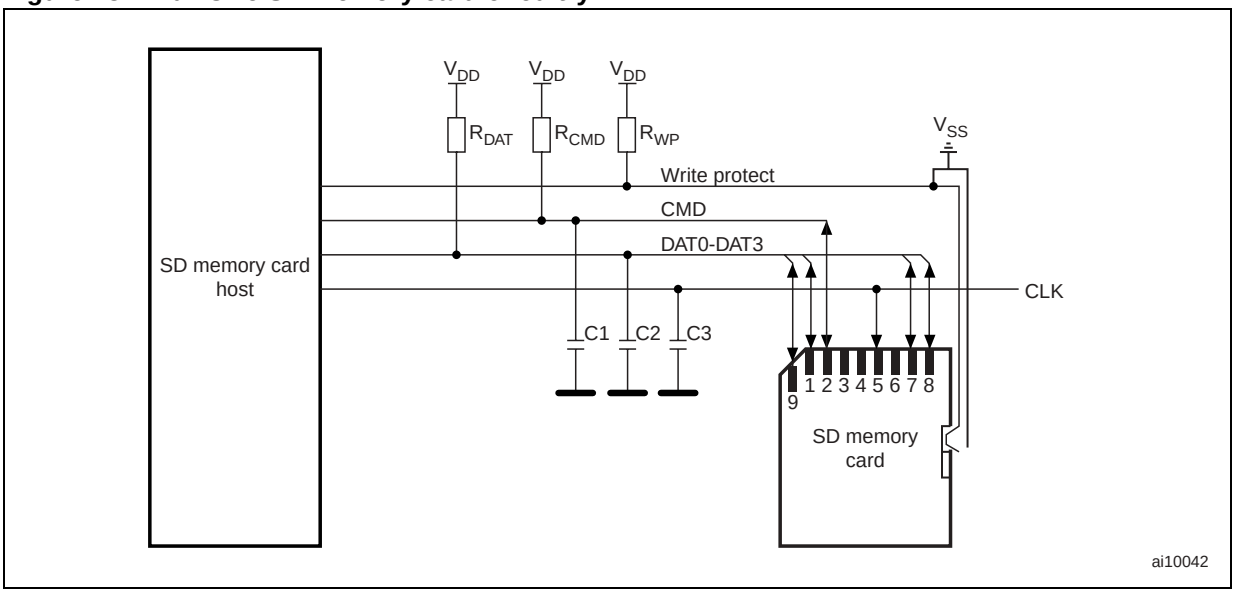
Figure 13 shows the internal bus circuitry required for the SD memory card.

The SD memory card may also feature two additional contacts, that are not part of the internal circuitry. When present in the device, these contacts are located at the level of the write protect/card detect switch in the socket, and should be connected as illustrated in *Figure 13*.

When DAT3 is used for card detection, the R_{DAT} resistor connected to DAT3 should be disconnected and another resistor should be connected to ground.

R_{DAT} and R_{CMD} are pull-up resistors used to protect the DAT and CMD lines, respectively, against bus floating when no card is inserted or when all card drivers are high impedance. R_{WP} is used to protect the write protect/card detection switch.

Figure 13. Full size SD memory card circuitry



4.2 Power-up

The power-up time is defined as the voltage rising time from 0 V to V_{DD}(min) and depends on application parameters such as the maximum number of SD cards, the bus length and the characteristic of the power supply unit.

The supply ramp-up time indicates the time that the power needs to reach the operating level (the bus master supply voltage) and the time to wait until the SD card can accept the first command.

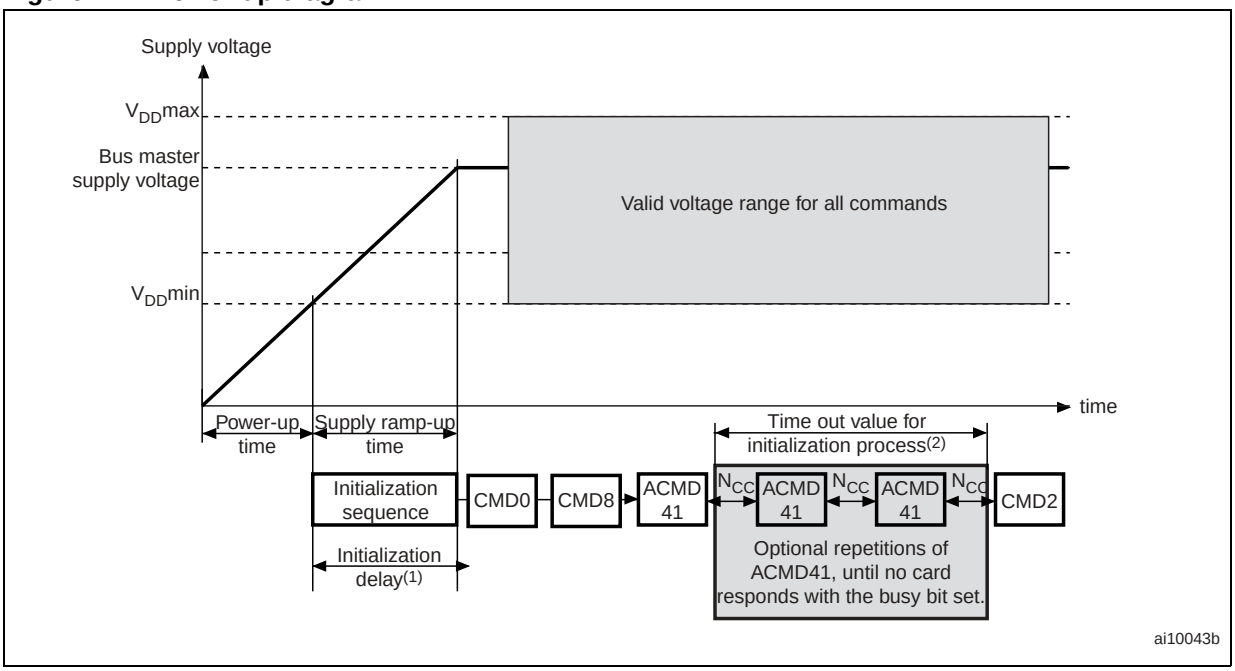
The host supplies power to the card so that the voltage reaches V_{DD}(min) within 250 ms and starts to supply at least 74 SD clocks to the SD card with keeping CMD line to High. In case of SPI mode, CS is held to High during 74 clock cycles.

After power-up (including hot insertion) the SD card enters the idle state. In case of SD host, CMD0 is not necessary. In case of SPI host, CMD0 is 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 to check whether the card supports supplied voltage. The version 2.00 host issues CMD8 and verifies voltage before card initialization. The host that does not support CMD8 supplies 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 checks that all cards satisfy the supplied voltage. Otherwise, the host should select one of the cards and initialize.

Figure 14. Power-up diagram



1. Initialization delay = 1 ms (max) + 74 clock cycles + supply ramp-up time.
2. Timeout value for initialization process is 1 s.

4.3 Hot insertion/removal

To guarantee a reliable initialization during hot insertion, some measures must be taken on by the host.

For example, a special hot-insertion capable card connector may be used to guarantee the sequence of the card pin connection.

The card contacts are connected 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 be connected first on insertion, and be disconnected last on extraction.

Another method is a switch which could ensure that the power is switched on only after all card pads are connected.

Inserting a card in or removing it from the SD memory card bus with the power on will not damage the card. Data transfer operations are protected by CRC codes, therefore any bit changes induced by card insertion and removal can be detected by the SD memory card bus master.

- The inserted card must be properly reset even when the clock frequency is f_{PP}
- Each card should be fitted with a protection from the power supply to prevent damage to the card (and host).
- Data transfer failures induced by removal/insertion are detected by the bus master. They must be corrected by the application, which may repeat the issued command.

4.4 Power protection

Cards have to be inserted in or removed from the bus without being damaged. If one of the supply pins (V_{DD} or V_{SS}) is not connected properly, then the current is drawn through a data line. All the card outputs should also be able to withstand shortcuts to either supply. If the hot insertion feature is implemented in the host, then the host has to be able to withstand an instant shortcut between V_{DD} and V_{SS} without being damaged.

4.5 Electrical specifications

Table 17 defines the bus operating conditions for the SD memory card. The total capacitance C_L of the CLK line of the SD memory card bus is the sum of the bus master capacitance C_{HOST}, the bus capacitance C_{BUS} and the capacitances C_{CARD} of all the cards connected to this line.

$C_L = C_{HOST} + C_{BUS} + N \times C_{CARD}$, where:

- N is the number of cards connected to the line
- $C_{HOST} + C_{BUS}$ must be lower than 30 pF for up to 10 cards and lower than 40 pF for up to 30 cards
- The values in *Table 17* should not be exceeded.

As the bus can be supplied with a variable supply voltage, all signal levels are related to the supply voltage. See *Figure 15: Bus signal levels* and *Table 18: Bus signal condition - I/O signal voltages*.

Table 17. Bus operating conditions

Symbol	Parameter	Min	Max	Unit	Remark
	Peak voltage on all lines	–0.3	V_{DD}+0.3	V	
	Input leakage current	–10	10	A	
	Output leakage current	–10	10	A	
V_{DD}	Supply voltage	2.0	3.6	V	
	Supply voltage specified in OCR register				
V_{SS1}, V_{SS2}	Supply voltage differentials	–0.3	0.3	V	
	Power-up time		250	ms	
R_{CMD}, R_{DAT}	Pull-up resistance	10	100	KΩ	
C_L	Bus signal line capacitance		250	pF	f_{PP} < 5 MHz 21 cards
			100	pF	f_{PP} < 20 MHz 21 cards
C_{CARD}	Single card capacitance		10	pF	
	Maximum signal line inductance		16	nH	f_{PP} < 20 MHz
R_{DAT3}	Pull-up resistance inside card (pin1)	10	90	KΩ	

Figure 15. Bus signal levels

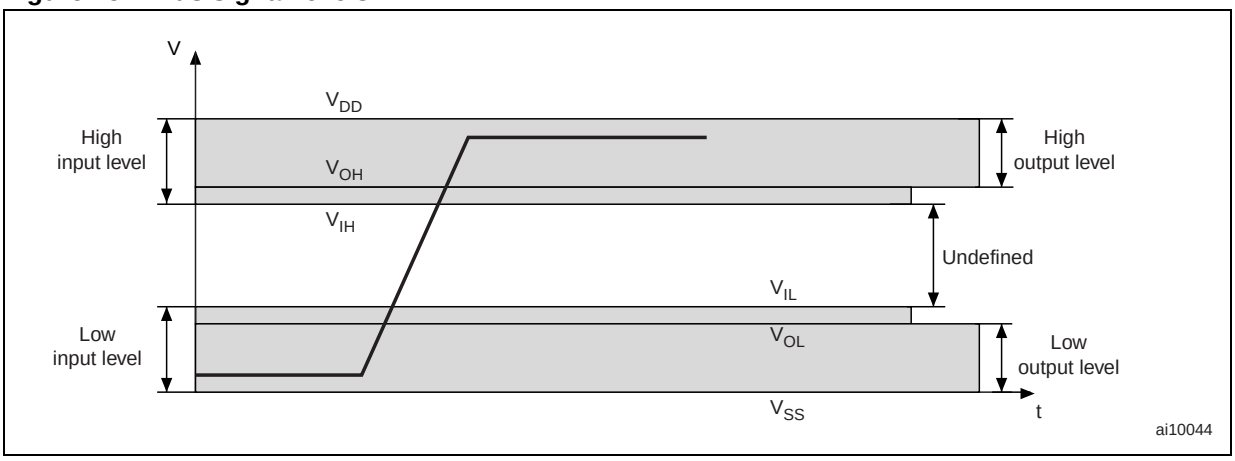


Table 18. Bus signal condition - I/O signal voltages

Symbol	Parameter	Conditions	Min	Max	Unit
V_{OH}	High output voltage	I_{OH} = –100 µA at V_{DD}(min)	0.75V_{DD}		V
V_{OL}	Low output voltage	I_{OL} = 100 µA at V_{DD}(min)		0.125V_{DD}	V
V_{IH}	High input voltage		0.625V_{DD}	V_{DD} + 0.3	V
V_{IL}	Low input voltage		V_{SS} – 0.3	0.25V_{DD}	V

Figure 16. Data input/output timings referenced to clock (default)

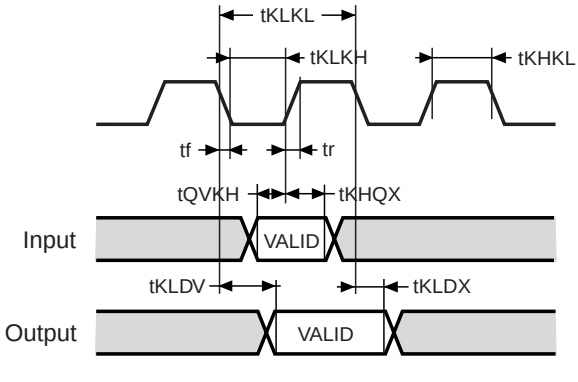


Table 19. Bus timings (default) (1)

Symbol	Alt	Parameter	Condition	Min	Max	Unit
t_{KLKL}	f_{PP}	Clock frequency data transfer mode	C card <= 10 pF (1 card)	0	25	MHz
f_{OD}		Clock frequency identification mode (the low frequency is required for MultiMediaCard compatibility).	C card <= 10 pF (1 card)	0	400	kHz
t_{KLKH}	t_{WL}	Clock low time	C card <= 10 pF (1 card)	10		ns
t_{KHKL}	t_{WH}	Clock high time	C card <= 10 pF (1 card)	10		ns
t_r	t_{TLH}	Clock rise time	C card <= 10 pF (1 card)		10	ns
t_f	t_{THL}	Clock fall time	C card <= 10 pF (1 card)		10	ns
Inputs CMD, DAT (referenced to CLK)						
t_{QVKH}	t_{ISU}	Input set-up time	C card <= 10 pF (1 card)	5		ns
t_{KHQX}	t_{IH}	Input hold time	C card <= 10 pF (1 card)	5		ns
Outputs CMD, DAT (referenced to CLK)						
t_{KLDX} t_{KLDV}	t_{ODLY}	Output delay time	C card <= 40 pF (1 card)		14	ns

1. Clock CLK: all values are referred to min (V_{IL}) and max (V_{IH}).

Figure 17. Data input/output timings referenced to clock (high-speed mode)

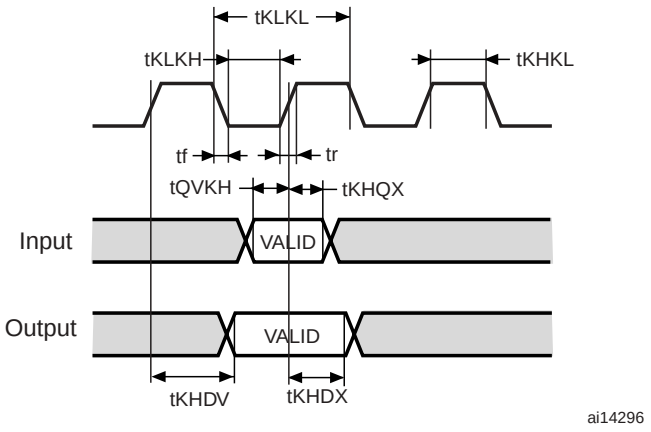


Table 20. Bus timings (high speed)(1)

Symbol	Alt	Parameter	Condition	Min	Max	Unit
t_{KLKL}	f_{PP}	Clock frequency data transfer mode	C card <= 10 pF (1 card)	0	50	MHz
t_{KLKH}	t_{WL}	Clock low time	C card <= 10 pF (1 card)	7		ns
t_{KHKL}	t_{WH}	Clock high time	C card <= 10 pF (1 card)	7		ns
t_r	t_{TLH}	Clock rise time	C card <= 10 pF (1 card)		3	ns
t_f	t_{THL}	Clock fall time	C card <= 10 pF (1 card)		3	ns
Inputs CMD, DAT (referenced to CLK)						
t_{QVKH}	t_{ISU}	Input set-up time	C card <= 10 pF (1 card)	6		ns
t_{KHQX}	t_{IH}	Input hold time	C card <= 10 pF (1 card)	2		ns
Outputs CMD, DAT (referenced to CLK)						
t_{KHDV}	t_{ODLY}	Output delay time	C card <= 40 pF (1 card)		14	ns
t_{KHDX}	t_{OH}	Output hold time	C card >= 15 pF (1 card)	2.5		ns

1. Clock CLK: all values are referred to min (V_{IL}) and max (V_{IH}).

5 Card registers

Six registers are defined in the card interface: OCR, CID, CSD, RCA, DSR and SCR. See *Table 21* for a description.

The registers are accessed by using the corresponding commands. The OCR, CID, CSD and SCR registers contain the card/content specific information, whereas the RCA and DSR registers are configuration registers that store the actual configuration parameters.

For more details about the register structure, please refer to the SD memory card specification version 2.00.

Table 21. SD memory card registers

Name	Width	Description
CID	128	**Card identification number register**. It contains the card's individual identification number. It is mandatory.
RCA(1)	16	**Relative card address register**. It contains the local system address of the card, that is dynamically suggested by the card and approved by the host during initialization. It is mandatory.
DSR	16	**Driver stage register**. It is used to configure the card's output drivers. It is optional.
CSD	128	**Card specific data register**. It contains the information about the card's operation conditions. It is mandatory.
SCR	64	**SD configuration register**. It contains the information about the SD memory card's special feature capabilities. It is mandatory
OCR	32	**Operation condition register**. It is mandatory.

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

5.1 OCR register

The 32-bit operation conditions register contains the V_{DD} voltage profile of the card. It also includes a status information bit that goes High (set to '1') once the card power-up sequence has completed. The OCR register is used by the cards that do not support the full operating voltage range of the SD memory card bus, or by cards whose power-up sequence does not match the definition given in *Figure 14: Power-up diagram*.

Bit 7 of OCR is newly defined for the dual voltage card and set to '0' by 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'. This register includes bit 30: this bit is set to '1' if the card is an high capacity SD memory card, while '0' indicates that the card is a 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 reads this status bit to identify a standard or high capacity SD memory card.

Table 22. OCR register definition

OCR bit position	V_{DD} voltage range
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 V to 2.8 V
16	2.8 V to 2.9 V
17	2.9 V to 3.0 V
18	3.0 V to 3.1 V
19	3.1 V to 3.2 V
20	3.2 V to 3.3 V
21	3.3 V to 3.4 V
22	3.4 V to 3.5 V
23	3.5 V to 3.6 V
24-29	reserved
30	Card capacity status (CCS)
31	Card power-up status bit (busy)

5.2 CID register

The card identification (CID) register contains the card identification information used during the card identification phase. Each flash memory card should have a unique identification number. The structure of the CID register is defined in *Figure 16*.

Table 23. CID fields

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]

5.3 CSD register

The card specific data register provides information on how to access the card contents. The CSD register defines the data format, error correction type, maximum data access time, data transfer speed, whether the DSR register can be used etc. The programmable register parameters (entries with cell type W or R, listed in *Table 24*) can be changed using CMD27.

Table 24. CSD fields compatible with CSD structure V1 / MM card specification V2.11

Name	Field	Width	Cell type(1)	CSD-slice
CSD structure	CSD_STRUCTURE	2	R	[127:126]
Reserved	–	6	R	[125:120]
Data read access-time-1	TAAC	8	R	[119:112]
Data read access-time-2 in CLK cycles (NSAC*100)	NSAC	8	R	[111:104]
Max. data transfer rate	TRAN_SPEED	8	R	[103:96]
Card command classes	CCC	12	R	[95:84]
Max. read data block length	READ_BL_LEN	4	R	[83:80]
Partial blocks for read allowed	READ_BL_PARTIAL	1	R	[79:79]
Write block misalignment	WRITE_BLK_MISALIGN	1	R	[78:78]
Read block misalignment	READ_BLK_MISALIGN	1	R	[77:77]
DSR implemented	DSR_IMP	1	R	[76:76]
Reserved	–	2	R	[75:74]
Device size	C_SIZE	12	R	[73:62]
Max. read current @V_{DD}(min)	VDD_R_CURR_MIN	3	R	[61:59]
Max. read current @V_{DD}(max)	VDD_R_CURR_MAX	3	R	[58:56]
Max. write current @V_{DD}(min)	VDD_W_CURR_MIN	3	R	[55:53]
Max. write current @V_{DD}(max)	VDD_W_CURR_MAX	3	R	[52:50]
Device size multiplier	C_SIZE_MULT	3	R	[49:47]
Erase single block enable	ERASE_BLK_EN	1	R	[46:46]
Sector size	SECTOR_SIZE	7	R	[45:39]
Write protect group size	WP_GRP_SIZE	7	R	[38:32]
Write protect group enable	WP_GRP_ENABLE	1	R	[31:31]
Reserved for MultiMediaCard compatibility	–	2	R	[30:29]
Write speed factor	R2W_FACTOR	3	R	[28:26]
Max. write data block length	WRITE_BL_LEN	4	R	[25:22]
Partial blocks for write allowed	WRITE_BL_PARTIAL	1	R	[21:21]
Reserved	–	5	R	[20:16]
File format group	FILE_FORMAT_GRP	1	R/W(1)	[15:15]
Copy flag (OTP)	COPY	1	R/W(1)	[14:14]
Permanent write protection	PERM_WRITE_PROTECT	1	R/W(1)	[13:13]
Temporary write protection	TMP_WRITE_PROTECT	1	R/W	[12:12]
File format	FILE_FORMAT	2	R/W(1)	[11:10]
Reserved	–	2	R/W	[9:8]
CRC	CRC	7	R/W	[7:1]
Not used, always '1'	–	1	-	[0:0]

1. R = readable, W(1) = can be written once, W = can be written several times.

5.4 RCA register

The writable 16-bit relative card address register contains the card address published by the card during the identification phase. This address is used for addressed host-card communications after the card identification phase. The default value of the RCA register is 0000h. This value is reserved, the CMD7 command uses it to set all the cards to the standby state.

5.5 DSR register (optional)

The 16-bit driver stage register is not used in Numonyx cards.

5.6 SCR register

The SD card configuration register (SCR) is a configuration register. The SCR provides information on the special features that are configured in the SD memory card. The size of SCR register is 64 bits.

This register is programmed in the factory by the SD memory card manufacturer. *Table 25* describes the SCR contents.

Table 25. SCR fields

Description	Field	Width	Cell type(1)	SCR slice
SCR structure	SCR_STRUCTURE	4	R	[63:60]
SD memory card - specification. 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 width supported	SD_BUS_WIDTHS	4	R	[51:48]
Reserved	–	16	R	[47:32]
Reserved for manufacturer usage	–	32	R	[31:0]

1. R = readable.

6 Timings

The symbols listed in *Table 26* are used in all timing diagrams.

The difference between P-bits and Z-bits is that P-bits are actively driven to High by the card or the host output driver whereas Z-bits are driven to High and kept High by the pull-up resistors R_{CMD} and R_{DAT}. P-bits, which are actively driven High, are less sensitive to noise. All timing values are defined in *Table 27*.

Table 26. Timing diagram symbols

Symbol	Description
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

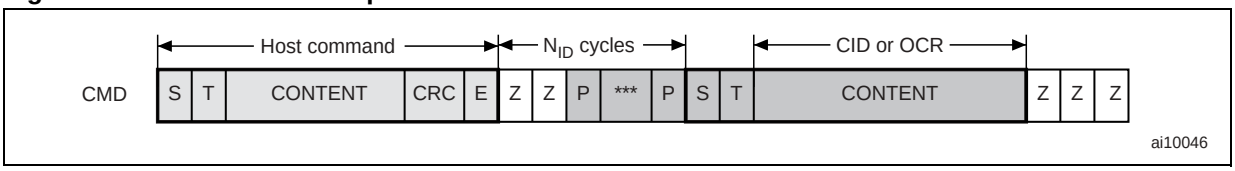
6.1 Command and response

The host command and the card response are both clocked out on the rising edge of the host clock.

6.1.1 Card identification and operating conditions timings

The timings for CMD2 (ALL_SEND_CID) and ACMD41 are shown in *Figure 18*. The command is followed by two Z-bits (to leave time for the bus to switch direction) and then by P-bits pushed up by the responding card. The card response to the host command starts after N_{ID} clock cycles.

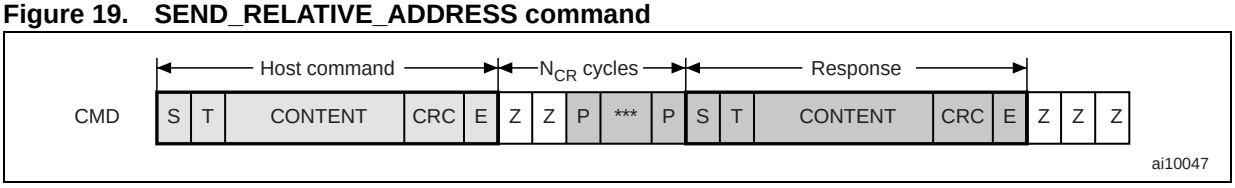
Figure 18. Identification sequence



6.1.2 Card relative address timings

The SD memory card timings for CMD3 (SEND_RELATIVE_ADDR) are given in *Figure 19*. The minimum delay between the host command and the card response is N_{CR} clock cycles.

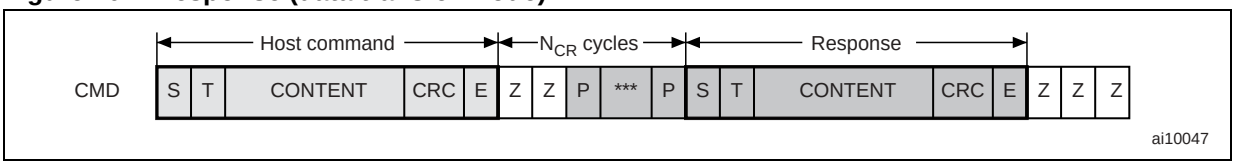
Figure 19. SEND_RELATIVE_ADDRESS command



6.1.3 Data transfer mode

After publishing its RCA the card switches to the data transfer mode. The command is followed by two Z-bits (to leave time for the bus to switch direction) and then by P-bits pushed by the responding card as shown in *Figure 20*. The timing diagram presented in *Figure 20* applies to all host commands followed by card responses, and to ACMD41 and CMD2 commands.

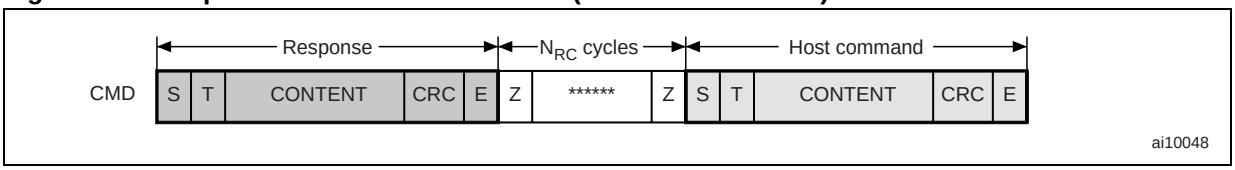
Figure 20. Response (data transfer mode)



6.1.4 Last card response, next host command timings

After receiving the last card response, the host can start the next command transmission after N_{RC} clock cycles as shown in *Figure 21*. The timing diagram presented in *Figure 21* applies to any host command.

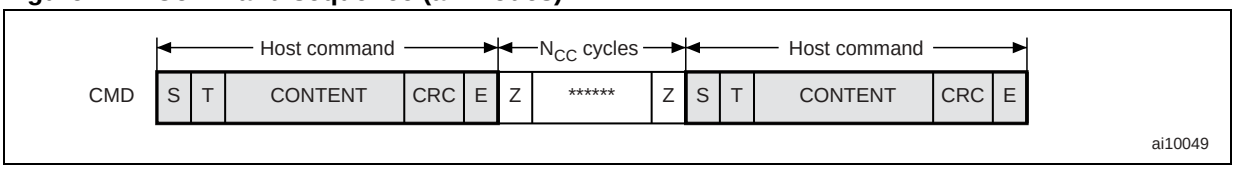
Figure 21. Response end to next CMD start (data transfer mode)



6.1.5 Last host command, next host command timings

The host can send a new command N_{CC} clock cycles after sending the previous one as shown in *Figure 22*.

Figure 22. Command sequence (all modes)



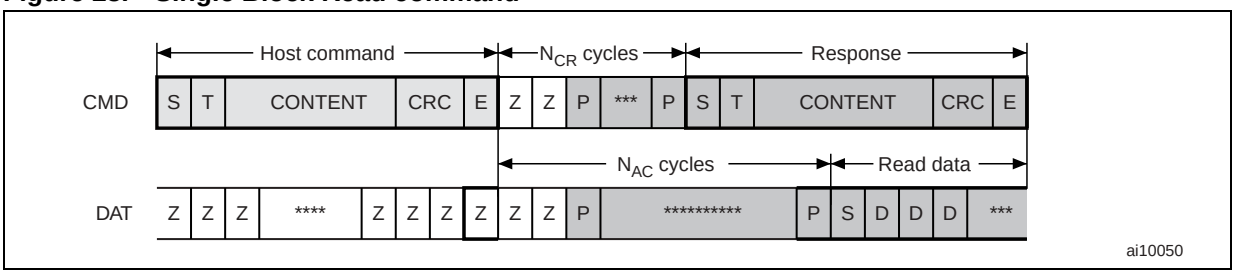
6.2 Data read

6.2.1 Single block read

The host selects one card for the data read operation by issuing CMD7, and sets the valid block length for oriented data transfer by issuing CMD16. *Figure 23* shows the timings for a basing bus read operation. The sequence starts with a Single Block Read command (CMD17) which specifies the start address in the argument field. The response is sent on the CMD line.

Data transmission from the card starts N_{AC} after the end bit of the read command, where N_{AC} is the access time. CRC check bits are appended to the data bits to allow the host to check for transmission errors.

Figure 23. Single Block Read command



6.2.2 Multiple block read

In multiple block read mode, the card responds to the read command from the host by sending a continuous flow of data blocks. The data flow is terminated by a STOP_TRANSMISSION command (CMD12). *Figure 24* describes the Multiple Block Read command followed by the data blocks and *Figure 25*, the response to a STOP_TRANSMISSION command. The data transmission stops two clock cycles after the end bit of the STOP_TRANSMISSION command.

Figure 24. Multiple Block Read command

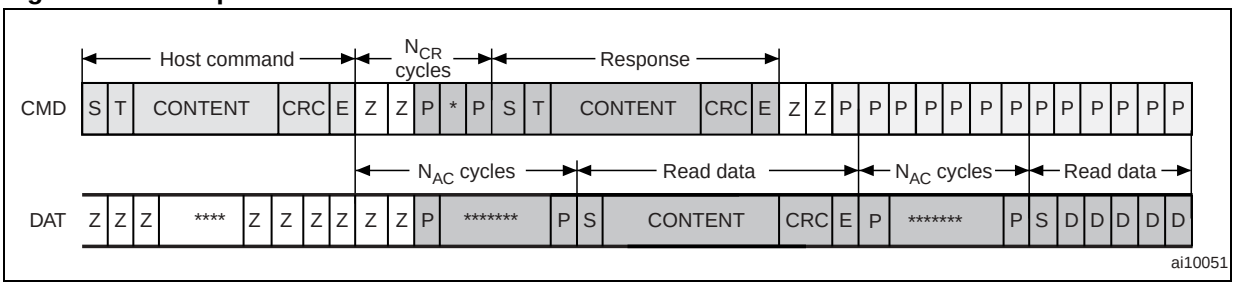
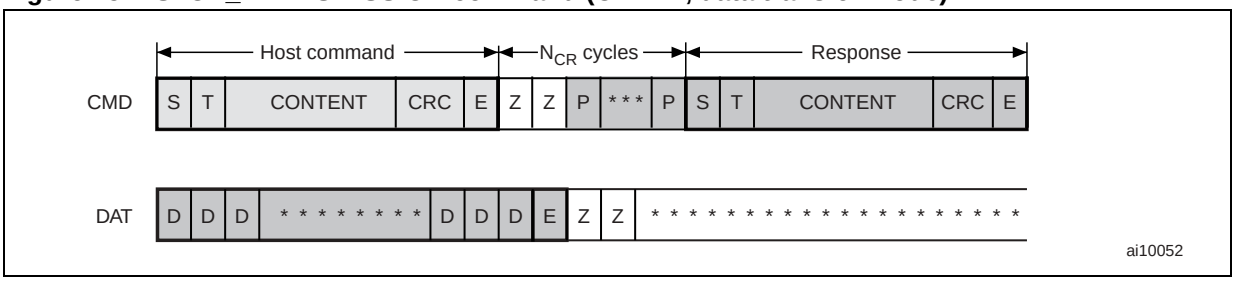


Figure 25. STOP_TRANSMISSION command (CMD12, data transfer mode)



6.3 Data write

6.3.1 Single block write

The host selects one card for the data write operation by issuing CMD7. The host sets the valid block length for block oriented data transfer by issuing CMD16. *Figure 26* shows the timings of a basic bus write operation. The sequence starts with a Single Block Write command (CMD24) which determines (in the argument field) the start address. The card responds on the CMD line.

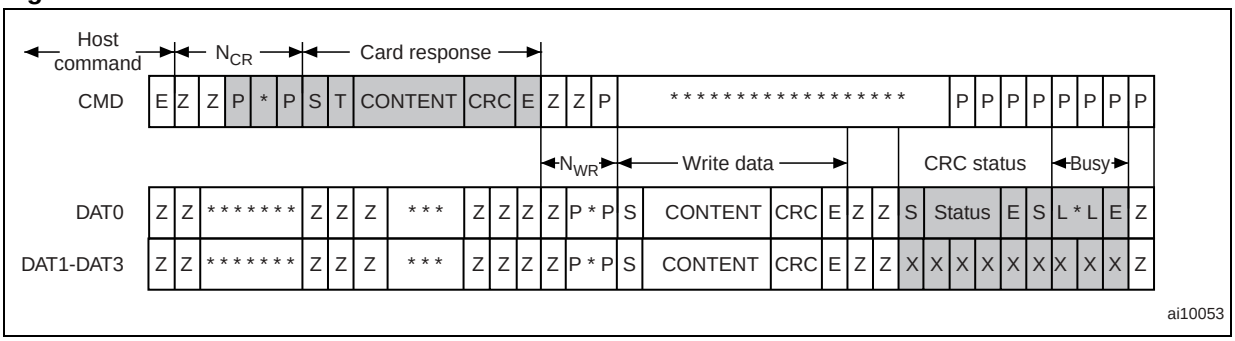
Data transfer from the host starts N_{WR} clock cycles after the card response is received. CRC check bits are appended to the data sent by the host to allow the card to check for transmission errors. The card returns the CRC check result as a CRC status token on the DAT0 line. If a transmission error occurred, the card returns a negative CRC status ('101'). If the transmission completed successfully, the card returns a positive CRC status ('010') and starts programming the data.

If an error occurred while programming the flash memory, the card ignores all further data blocks. In this case the card will not send any CRC response and so, there will be no CRC start bit on the bus and the three CRC status bits will read ('111').

Note that the CRC response is always output two clock cycles after the data.

If the card does not have any data receive buffer available, it indicates this condition by pulling the DAT0 data line to Low. It will stop pulling DAT0 to Low as soon as at least one data receive buffer for the defined data transfer block length becomes available. The level of DAT0 does not give any information about the data write status. The host can obtain this information by issuing a CMD13 (SEND_STATUS) to the card.

Figure 26. Block Write command



6.3.2 Multiple block write

In multiple block write mode, the write command from the host is followed by a continuous flow of data blocks from the host. The data flow is terminated by a STOP_TRANSMISSION command (CMD12).

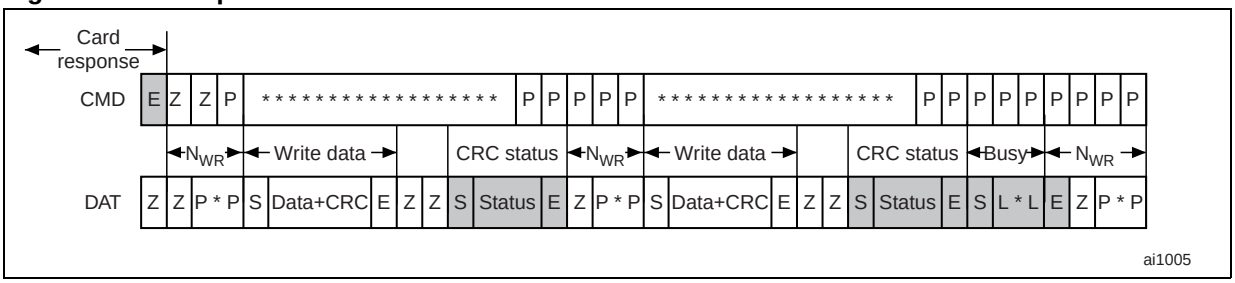
As in the case of a single block write operation, CRC check bits are appended to the data sent to allow the card to check for transmission errors. The card returns the CRC check result as a CRC status token on the DAT0 line.

If a transmission error occurred, the card returns a negative CRC status ('101'). If the transmission completed successfully, the card returns a positive CRC status ('010') and starts programming the data.

If an error occurred while programming the flash memory, the card ignores all further data blocks. In this case the card will not send any CRC response and so, there will be no CRC start bit on the bus and the three CRC status bits will read ('111').

Figure 27 describes a multiple block data transmission with and without a card busy signal.

Figure 27. Multiple Block Write command



6.4 STOP_TRANSMISSION command

The STOP_TRANSMISSION command works in the same way as in the read mode. *Figure 28* to *Figure 31* describe the timings of the STOP_TRANSMISSION command in different card states.

The card will consider that a data block was successfully received and is ready for programming only if the CRC data of the block was validated and the CRC status token, returned to the host.

Figure 29 is an example of an interrupted (by a STOP_TRANSMISSION command from the host) attempt to transmit the CRC status token. The result is the same as in other examples

where the STOP_TRANSMISSION command is implemented: the end bit of the STOP_TRANSMISSION command from the host is followed, on the data line, by one more data bit, then an end bit and two Z-bits. The two Z-bits, which correspond to two clock cycles, are used to switch the bus direction. The received data block is considered incomplete and will not be programmed.

In the previous stop transmission examples, the host stopped the data transmission during an active data transfer.

In *Figure 30* and *Figure 31* the STOP_TRANSMISSION command is received by the card after all the data blocks have been sent.

In *Figure 30*, the card is busy programming the last block when the STOP_TRANSMISSION command is received whereas in *Figure 31* the card is idle but the input buffers still contain data blocks to be programmed. In the second case, the card starts programming the blocks upon reception of the STOP_TRANSMISSION command and activates the busy signal.

Figure 28. STOP_TRANSMISSION during data transfer from the host

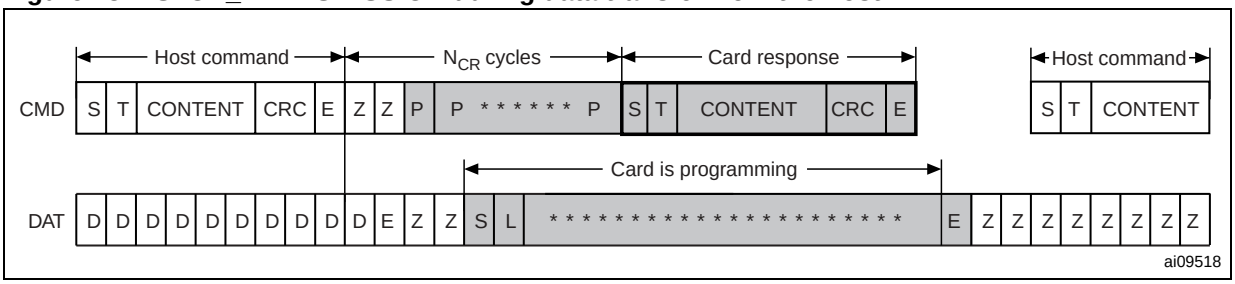
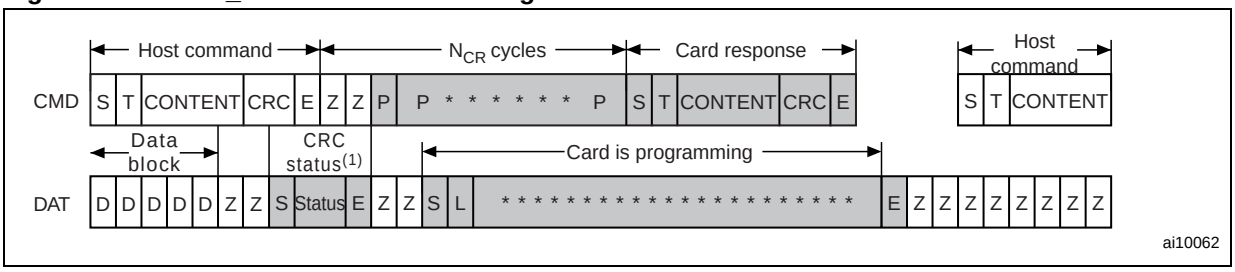


Figure 29. STOP_TRANSMISSION during CRC status transfer from card



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

Figure 30. STOP_TRANSMISSION received after last data block with card busy programming

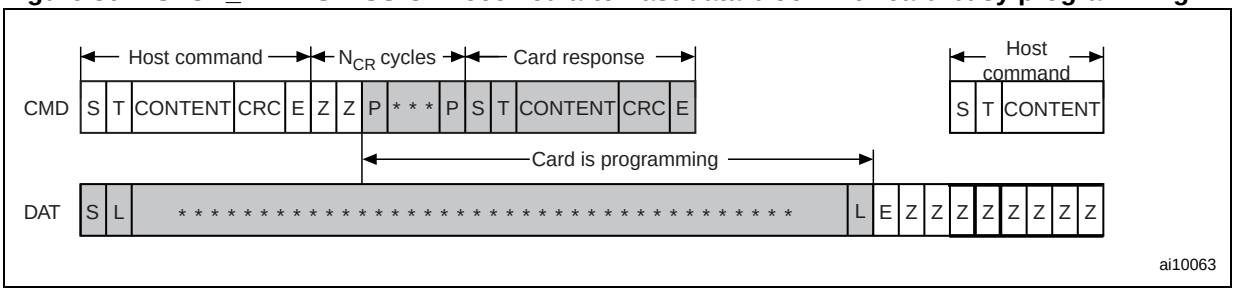
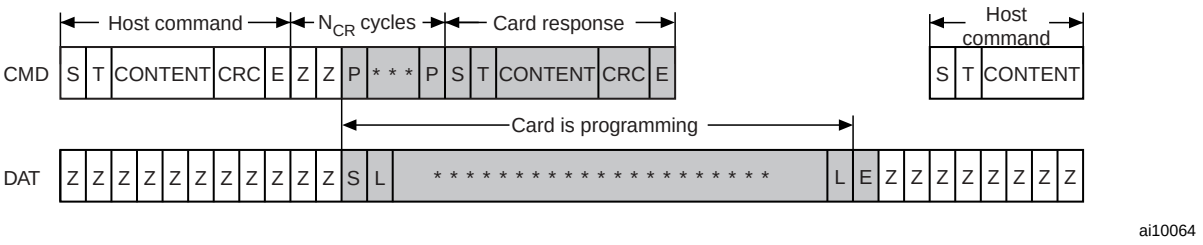


Figure 31. STOP_TRANSMISSION received after last data block with card idle



6.4.1 Erase, set and clear write protect timings

The host must first tag the start (CMD32) and end (CMD33) addresses of the range to be erased. The Erase command (CMD38), once issued, will erase all the selected write blocks. Similarly, Set and Clear Write Protect commands start a programming operation as well. The card will signal 'busy' (by pulling the DAT line Low) for the duration of the erase or program operation. The bus transaction timings are the same as those given for the STOP_TRANSMISSION command in *Figure 31*.

6.4.2 Re-selecting a busy card

When a busy card in disabled state is reselected, it restores its busy signaling on the data line. The timing diagram for this command / response / busy transaction is the same as that for STOP_TRANSMISSION command illustrated in *Figure 31*.

6.5 Timing values

Figure 27 gives all timings.

Table 27. Timing values

Parameter	Min	Max	Unit
N_{CR}	2	64	clock cycle
N_{ID}	5	5	clock cycle
N_{AC}	2	$T_{AAC} + N_{SAC}$	clock cycle
N_{RC}	8	–	clock cycle
N_{CC}	8	–	clock cycle
N_{WR}	2	–	clock cycle

7 Serial peripheral interface (SPI) mode

The SPI mode is a secondary communication protocol, which is available in flash memory-based SD memory cards. 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 off-the-shelf host, hence reducing the design-in effort to a minimum. The disadvantage is the loss of performance (for instance, single data line and hardware $\overline{CS}$ signal per card). The SPI mode is selected during the first Reset command after power-up (CMD0) and cannot be changed once the part is powered on.

7.1 SPI bus topology

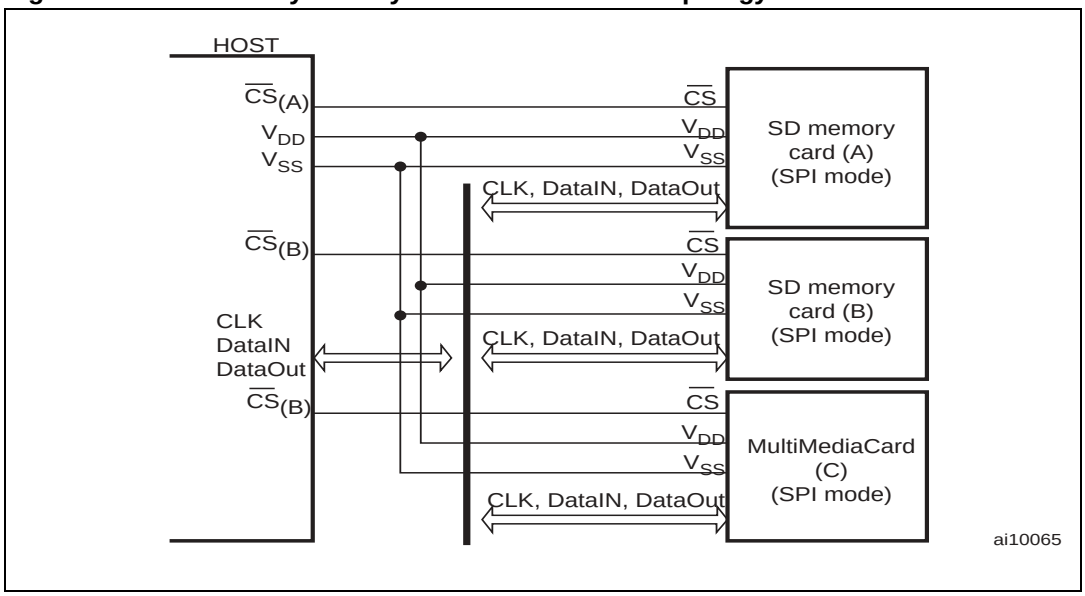
The SPI compatible communication mode of the SD memory card is designed to communicate with an SPI channel, commonly found in various microcontrollers on the market. The SPI standard defines the physical link only, and not the complete data transfer protocol. The SD memory card SPI and SD modes use the same command set.

Like all SPI devices, the SD memory card SPI channel uses the four following signals:

- $\overline{CS}$: host to card Chip Select signal.
- CLK: host to card Clock signal
- DataIn: host to card Data signal.
- DataOut: card to host Data signal.

All data tokens are multiples of bytes (8 bits) and always byte-aligned to the $\overline{CS}$ signal. The card identification and addressing methods are replaced by a hardware Chip Select ($\overline{CS}$) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active Low) the $\overline{CS}$ signal (see *Figure 32: SD memory card system SPI mode bus topology*). The $\overline{CS}$ signal must be continuously active for the duration of the SPI transaction. The only exception occurs during card programming, when the host can de-assert the $\overline{CS}$ signal without affecting the programming process. The SPI interface uses 7 out of the 9 SD signals (DAT1 and DAT2 are not used, DAT3 is the $\overline{CS}$ signal) of the SD bus.

Figure 32. SD memory card system SPI mode bus topology



7.2 SPI bus protocol

Whereas the SD channel is based on command and data bit streams initiated by a start bit and terminated by a stop bit, the SPI channel is byte-oriented. Every command or data block is built up with 8-bit bytes and is byte-aligned to the $\overline{CS}$ signal (that is, the length is a multiple of 8 clock cycles).

Like in the SD protocol, messages in the SPI protocol consist of command, response and data-block tokens. All communications between host and cards are controlled by the host (master). The host starts every bus transaction by asserting the $\overline{CS}$ signal Low.

The response behaviors in SPI mode and SD mode differ in three ways. In the SPI mode:

- the selected card always responds to the command
- two additional (8- & 16-bit) response structures are used
- when the card encounters a data retrieval problem, it sends an error response in place of the expected data block (in the SD mode the card does not respond but implements a timeout).

In addition to returning a response for every command received, the card returns a special data response token for every data block received during write operations.

7.2.1 Mode selection

The SD memory card wakes up in the SD mode. It will enter the SPI mode if the $\overline{CS}$ signal is asserted (Low) when the Reset command (CMD0) is received.

The only way to return to the SD mode is to start a new power-down/power-up sequence.

In SPI mode, the SD card protocol state machine does not apply.

7.2.2 Bus transfer protection

On entering the SPI mode the card defaults to the non-protected mode where there is no CRC (Cyclic Redundancy Check). So systems using reliable data links are not obliged to have the hardware and firmware necessary to implement CRC functions.

In non-protected mode, the CRC bits are still present but are don't care. The CRC option can be turned on and off by the host through the CRC_ON_OFF command (CMD59).

7.2.3 Data read

Single and multiple block read operations are supported in SPI mode. The main difference with the SD mode is that in SPI mode data and responses to the host are both sent on the DataOut line. As a consequence the data transfer may be interrupted and the last data block, replaced by the response to a STOP_TRANSMISSION command.

The basic unit of data transfer is the block. The maximum size of a block is defined in the CSD register (READ_BL_LEN).

If READ_BL_PARTIAL is set, smaller blocks entirely contained in a physical block (as defined by READ_BL_LEN) may also be transmitted. Single block read operations are initiated by issuing the READ_SINGLE_BOCK command (CMD17). The start address can be any byte in the valid address range of the card. Every block, however, must be contained in a single physical card sector.

Multiple block read operations are initiated by issuing the READ_MULTIPLE_BLOCK command (CMD18) and every transferred block has a 16-bit CRC appended to it. The STOP_TRANSMISSION command (CMD12) will actually stop the data transfer operation (just like in the SD mode).

Figure 33. Read operation mechanism

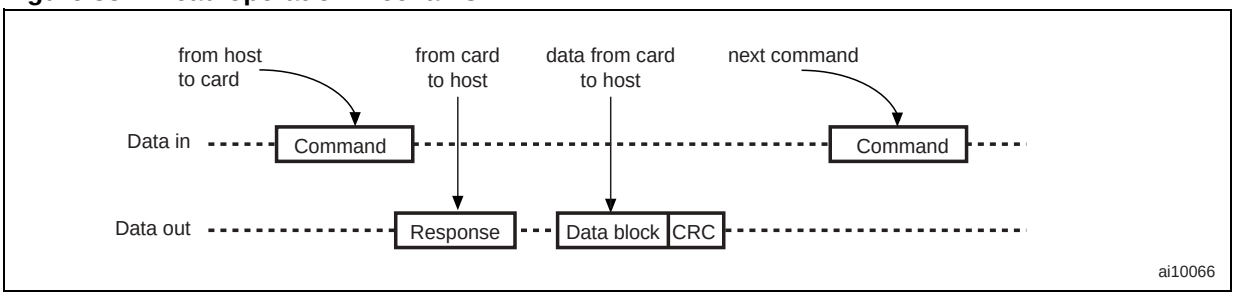


Figure 34. Multiple block read operation

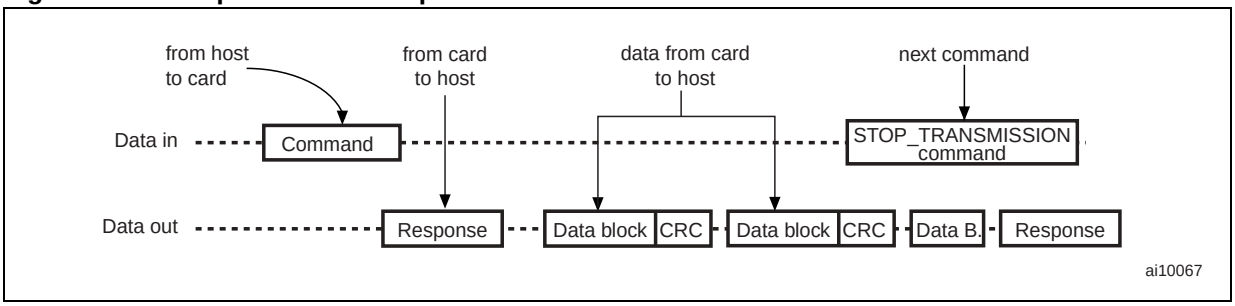
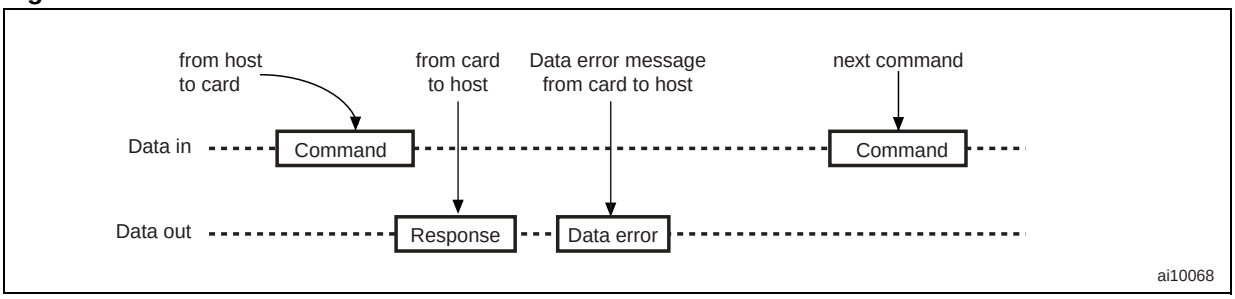


Figure 35. Read data error



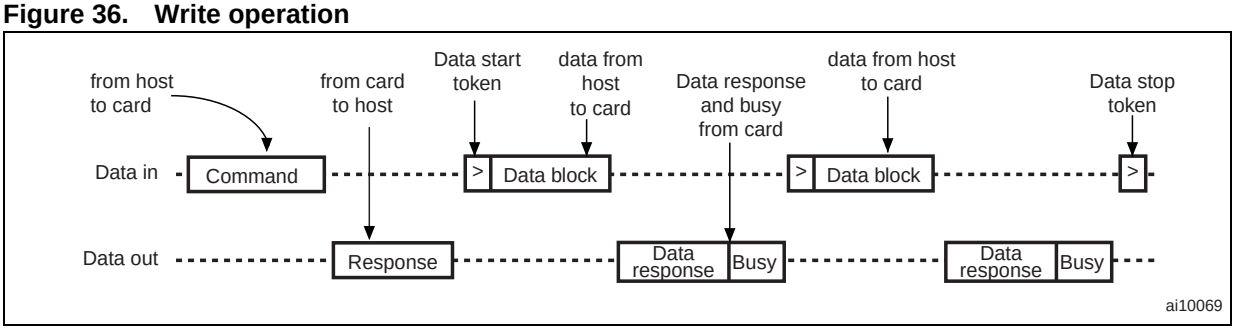
7.2.4 Data write

Single and multiple block write operations are supported in SPI mode.

Upon reception of a valid write command, the card sends a response token and waits for a data block to be sent from the host. Write operations, as illustrated in *Figure 36*, follow the same rules as read operations (refer to *Section 7.2.4: Data write*) as regards the CRC, block length and start address.

After receiving a data block, the card returns a data response token. If the data block received contains no error, it is programmed. Throughout the programming operation the card sends a continuous stream of busy tokens to the host (by holding the DataOut line Low).

Figure 36. Write operation

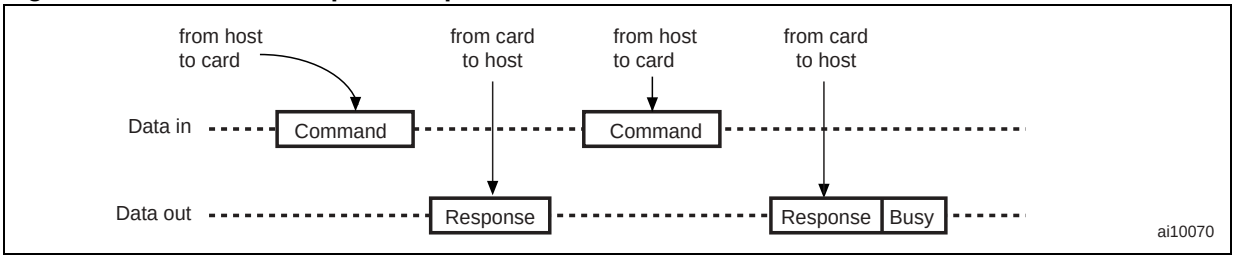


7.2.5 Erase & write protect management

The erase and write protect management procedures are the same in the SPI and SD modes.

While the card is erasing or changing the write protection bits, it remains in the busy state and holds the DataOut line Low.

Figure 37. Erase & write protect operations



7.2.6 Read CID/CSD registers

In SPI mode the CID and CSD registers use a block read operation. When a Read command is issued, the card returns a response message followed by a 16-byte data block with a 16-bit CRC.

As T_{AAC}, the data read access time, is stored in the CSD register, it cannot be used as the read latency of the CSD register. N_{CR} (see *Table 27: Timing values*) is used instead.

7.2.7 Reset sequence

The SD memory card requires a defined reset sequence. After power-on reset or CMD0 (software reset) the card enters an idle state. When idle, the only host commands the card will accept are CMD1 (SEND_OP_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 MMC card. For the thin (1.4 mm) standard size SD memory card, CMD1 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.

In the SPI mode, as opposed to the SD mode, CMD1 (and ACMD41) has no operands and does not return the contents of the OCR register. Instead, the host may use CMD58 (available in SPI mode only) to read the OCR register.

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 the SD mode and the response format of CMD8 is R7 (see SD memory card specification version 2.00).

7.2.8 Memory array partitioning

It is the same as in the SD mode.

7.2.9 Card Lock/Unlock commands

In the SPI mode, the Lock and Unlock commands are the same as in the SD mode.

7.2.10 Application Specific commands

The only difference between the SD and SPI modes is the APP_CMD status bit, which is not available in the SPI mode.

7.3 SPI mode commands

All the SPI commands are 6-byte long. The command always starts with the MSB of the string, which corresponds to the command code. See *Table 28* for details of the command format.

Like in the SD mode, the commands in the SPI mode are divided into classes. However, the classes supported by the two modes are different. See *Table* for details.

The commands supported in the SPI mode are described in detail in *Table* . If no argument is required in the command, the value of the field should be set to '0'. Reserved commands are reserved in both the MultiMediaCard and SPI modes. The contents of the command index field are binary: for example, it is '000000' for CMD0 and '100111' for CMD39.

For more details about commands and arguments, please refer to the SD memory card specification.

Table 28. Command format

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
Descriptions	Start bit	Transmission bit	Command index	Argument	CRC7	End bit

Table 29. Command classes in SPI mode

Card CMD class (CCC)	Class descr.	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 (optional)										+																+								
Class 8	Application specific																														+	+			
Class 9	Not supported in SPI			+																									+	+					
Class 10[1]	Switch				+																	+	+	+	+			+					+		
Class 11	Reserved																																		

1. This command class has been added in specification version 1.10.

7.4 Responses

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

7.4.1 R1 format

The card sends this response token after every command except for the SEND_STATUS command.

R1 format responses are 1-byte long. The MSB is always zero and the other bits indicate errors, an error being indicated by a '1'.

7.4.2 R1b format

This response token is similar to the R1 format response token but for the option of adding the busy signal.

The busy signal token can be any number of bytes. A zero value indicates that the card is busy. A non-zero value indicates that the card is ready for the next command.

7.4.3 R2 format

This response token is 2-byte long. It is sent as a response to the SEND_STATUS command.

7.4.4 R3 format

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

For more details about responses, please refer to the SD memory card specification v.2.00.

7.4.5 R7 format

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 the argument and are specified by the same definition as R7 response in SD mode.

7.5 Clearing status bits

In the SPI mode, as described in the previous paragraphs, status bits are reported to the host in three different formats: R1 format response, R2 format response and data error token (the same bits may exist in several response types - for instance 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 cleared either when read by the host or in accordance with the card state. For more details, please refer to the SD memory card specification.

7.6 SPI bus timings

Figure 39 illustrates the basic command/response transaction in SPI mode (that is, when the card is ready).

Figure 40 describes a command/ response transaction when the card is busy (R1b response format). For timings, refer to *Table 27*.

Table 30. SPI timing symbols

S	Start bit (=0)
T	Transmitter bit (host = 1, card = 0)
P	One-cycle pull-up (=1)
E	End bit (= 1)
Z	High impedance stage
D	Data bits
*	Repeater

Table 31. SPI timing values

Timing	Min	Max	Unit
N_{CS}	0		8 clock cycles
N_{CR}	1	8	8 clock cycles
N_{CX}	0	8	8 clock cycles
N_{RC}	1		8 clock cycles
N_{AC}	1	Specified in CSD register	8 clock cycles
N_{WR}	1		8 clock cycles
N_{EC}	0		8 clock cycles
N_{DS}	0		8 clock cycles
N_{BR}	1	1	8 clock cycles

Figure 38. Host command to card response - card is ready

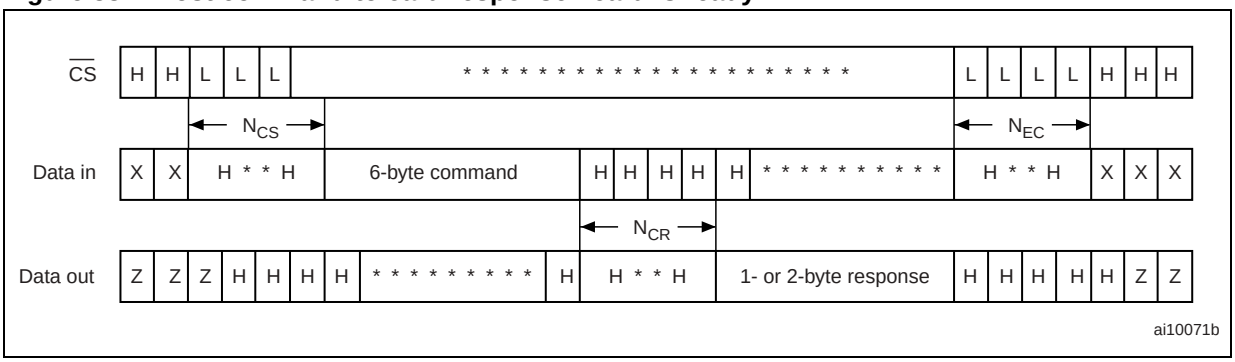


Figure 39. Host command to card response - card is busy

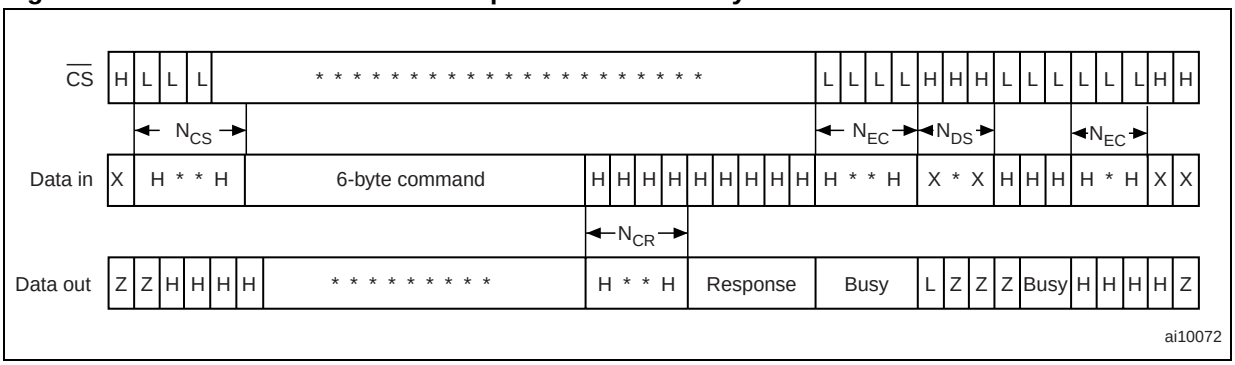
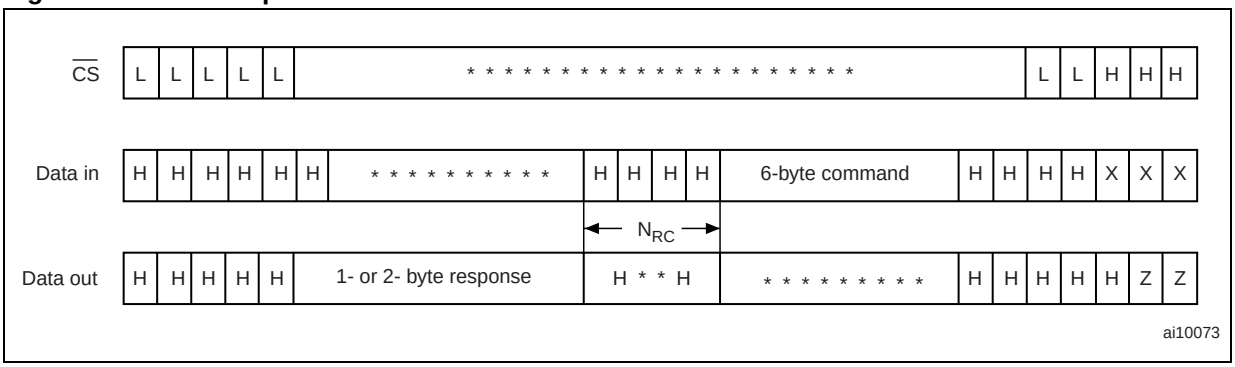


Figure 40. Card response to Host command



7.6.1 Data read timings

The timing diagram for deselecting the card by de-asserting $\overline{CS}$ after the last card response corresponds to a standard command-to-response timing diagram as illustrated in *Figure 40* During open-ended multiple block read operations, the STOP_TRANSMISSION command may be sent while the card is transmitting data to the host. In this case, the card stops transmitting the data block within two clock cycles (the bits in the first byte may not all be set to '1') and returns the response message after a time measured in numbers of clock cycles (N_{CR}). See *Figure 41* for details. For timings, refer to *Table 27*.

Figure 41. Single block read operation

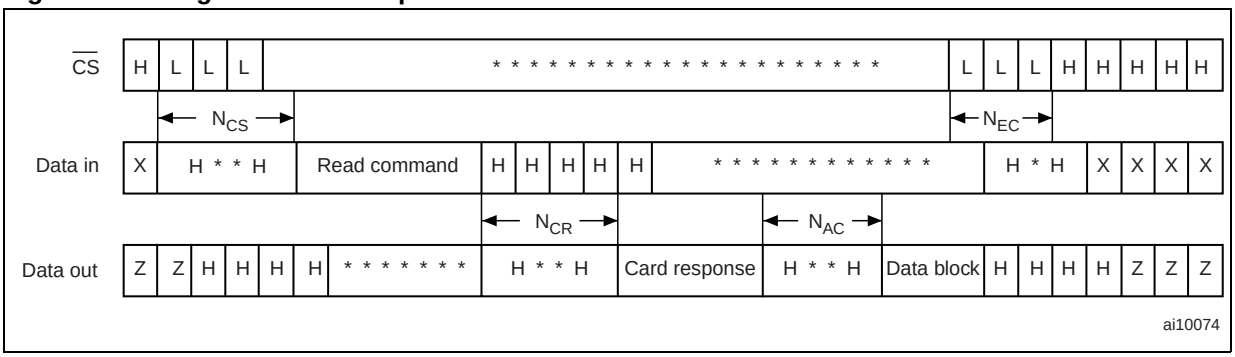


Figure 42. STOP_TRANSMISSION between blocks during multiple block read

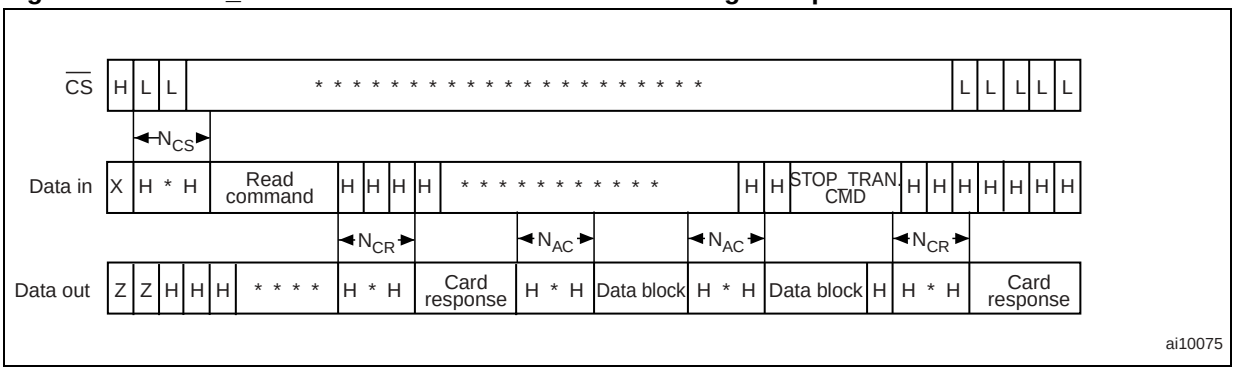


Figure 43. STOP_TRANSMISSION within a block during multiple block read

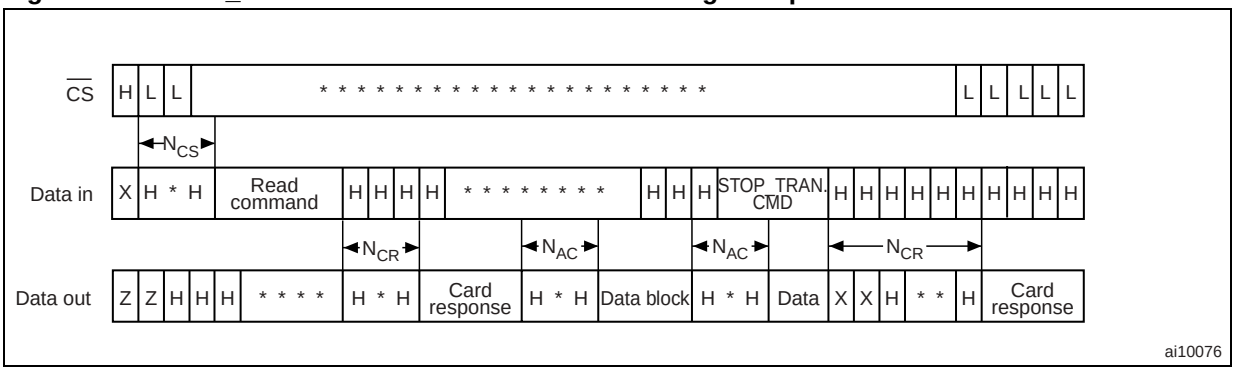
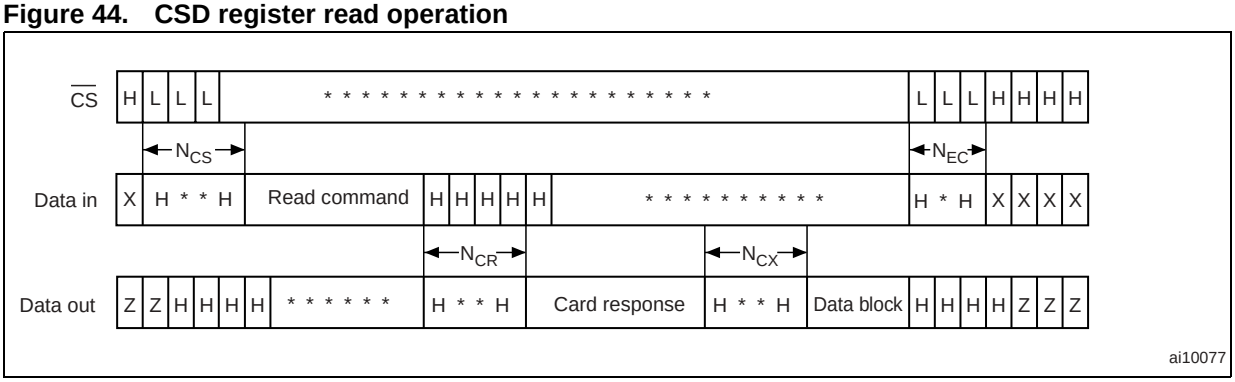


Figure 44. CSD register read operation



7.6.2 Data write timings

The host may deselect a card at any moment during single and multiple block write operations. The card will release the DataOut line one clock cycle after it is deselected ($\overline{CS}$ High). To check whether the card is still busy, the host must reselect it by driving $\overline{CS}$ Low. The card will then take control of the DataOut line one clock cycle after being reselected. In multiple block write operations, the timings from the command being issued to the first data block being transmitted by the card are the same as for single block write operations (see *Figure 45* for details). The timing of stop tran prefixes is the same as that of data blocks. After the card receives the STOP_TRANSMISSION command, the data on the DataOut line is undefined for one byte (N_{BR}), then a busy message may be sent by the card. For timings, refer to *Table 27*.

Figure 45. Single block write operation

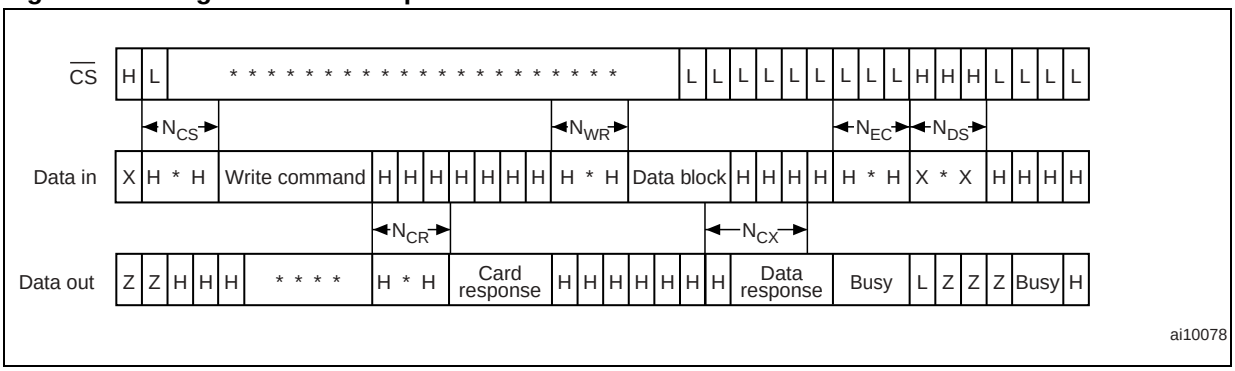
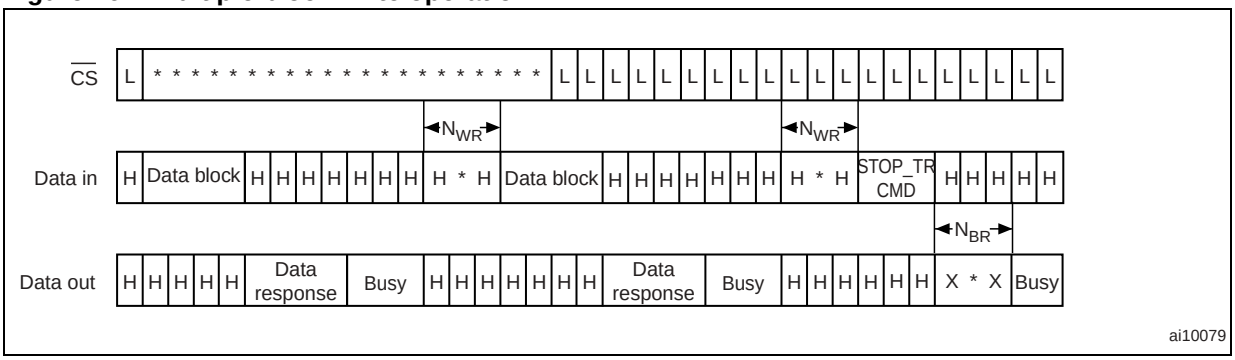


Figure 46. Multiple block write operation



8 Package mechanical data

To meet environmental requirements, Numonyx offers these devices in ECOPACK® packages. ECOPACK packages are lead-free. The category of second level interconnect is marked on the package and on the inner box label, in compliance with JEDEC Standard JESD97. The maximum ratings related to soldering conditions are also marked on the inner box label.

Figure 47. MicroSD card mechanical dimensions

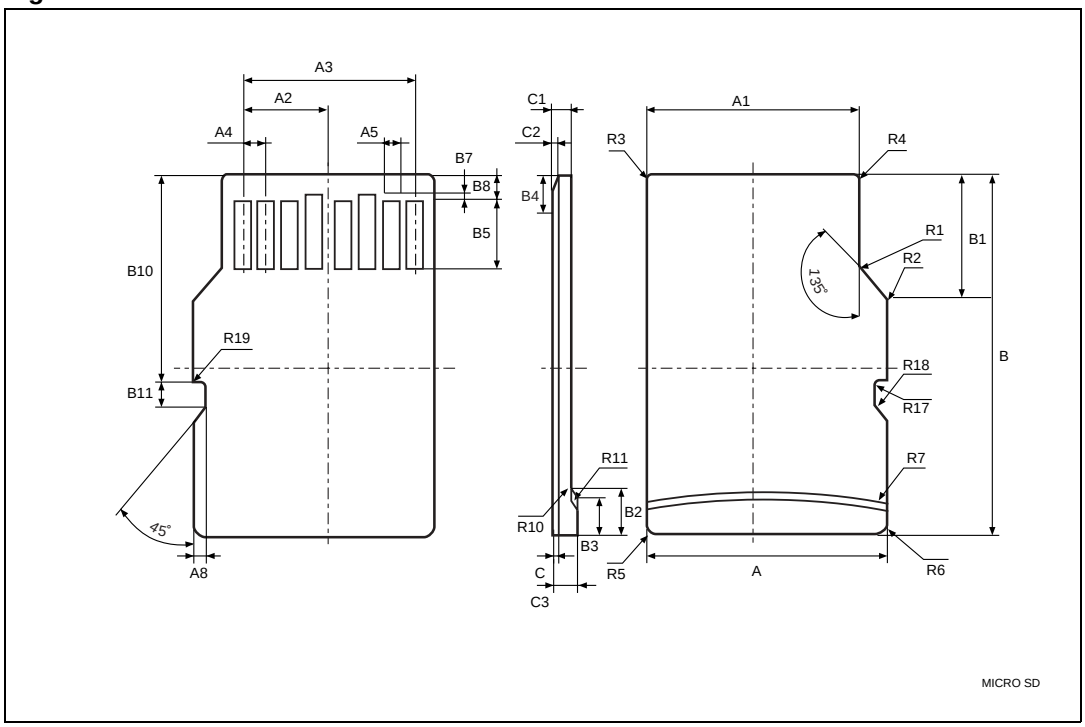


Table 32. MicroSD package mechanical data

Symbol	millimeters			inches		
	Min	Typ	Max	Min	Typ	Max
A	10.90	11.00	11.10	0.429	0.433	0.437
A1	9.60	9.70	9.80	0.378	0.382	0.386
A2	–	3.85	–	–	0.152	–
A3	7.60	7.70	7.80	0.299	0.303	0.307
A4	–	1.10	–	–	0.043	–
A5	0.75	0.80	0.85	0.030	0.031	0.033
A8	0.60	0.70	0.80	0.024	0.028	0.031
B	14.90	15.00	15.10	0.587	0.591	0.594
B1	6.30	6.40	6.50	0.248	0.252	0.256
B2	1.64	1.84	2.04	0.065	0.072	0.080
B3	1.30	1.50	1.70	0.051	0.059	0.067
B4	0.42	0.52	0.62	0.017	0.020	0.024
B5	2.80	2.90	3.00	0.110	0.114	0.118
B7	0.20	0.30	0.40	0.008	0.012	0.016
B8	1.00	1.10	1.20	0.039	0.043	0.047
B10	7.80	7.90	8.00	0.307	0.311	0.315
B11	1.10	1.20	1.30	0.043	0.047	0.051
c	0.17	0.21	0.25	0.007	0.008	0.010
C1	0.60	0.70	0.80	0.024	0.028	0.031
C2	0.20	0.30	0.40	0.008	0.012	0.016
C3	0.90	1.00	1.10	0.035	0.039	0.043
R1	0.20	0.40	0.60	0.008	0.016	0.024
R2	0.20	0.40	0.60	0.008	0.016	0.024
R3	0.70	0.80	0.90	0.028	0.031	0.035
R4	0.70	0.80	0.90	0.028	0.031	0.035
R5	0.70	0.80	0.90	0.028	0.031	0.035
R6	0.70	0.80	0.90	0.028	0.031	0.035
R7	29.50	30.00	30.50	1.161	1.181	1.201
R10	-	0.20	-	-	0.008	-
R11	-	0.20	-	-	0.008	-
R17	0.10	0.20	0.30	0.004	0.008	0.012
R18	0.20	0.40	0.60	0.008	0.016	0.024
R19	0.05	-	0.20	0.002	-	0.008

9 Ordering information

Table 33. Ordering information scheme

Example: SMS 02G D F B 5 E

Memory card standard

SMS = storage medium, secure digital

Density

512 = 512 Mbytes

01G = 1 Gbyte

02G = 2 Gbytes

Options of the standard

D = MicroSD

Memory type

F = flash memory

Card version

A, B = version depending on internal card configuration

Temperature range

5 = –25 to 85 °C

Packing

E = ECOPACK package, standard packing (tray)

Note: Other digits may be added to the ordering code for preprogrammed parts or other options.

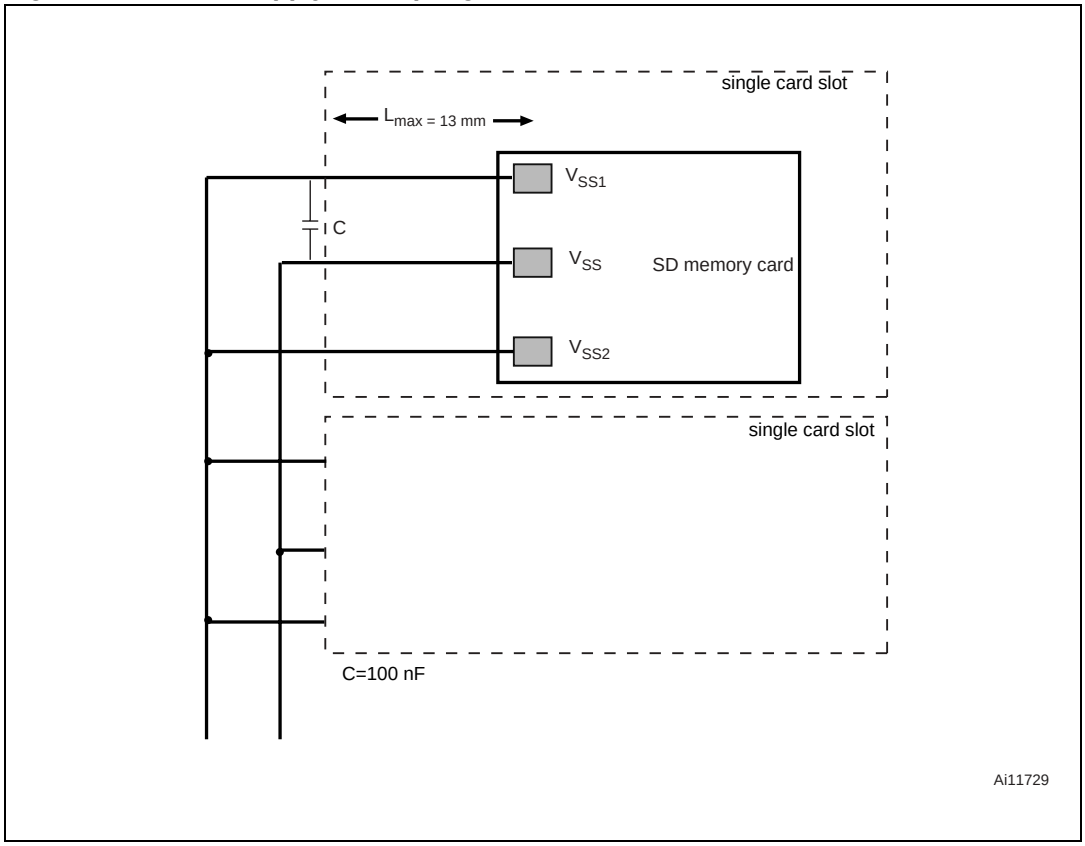
Devices are shipped from the factory with the memory content bits erased to '1'.

For further information on any aspect of the device, please contact the nearest Numonyx sales office.

Appendix A Power supply decoupling

The V_{SS1}, V_{SS2} and V_{DD} lines supply the card with the operating voltage. To do this, decoupling capacitors for buffering current peak are used. These capacitors are placed on the bus side corresponding to *Figure 48*.

Figure 48. Power supply decoupling



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

10 Revision history

Table 34. Document revision history

Date	Revision	Changes
10-Apr-2008	1	Initial release.
25-Jun-2008	2	Modified: read and write access for sustained multiple block *on page 1*, *Table 1: Device summary*, *Table 2: System performance*, *Table 3: Power consumption*, *Table 7: Memory array structures*, and title of *Table 19: Bus timings (default)*. Added *Figure 17: Data input/output timings referenced to clock (high-speed mode)* and *Table 20: Bus timings (high speed)*.
20-Nov-2008	3	Document's status promoted from preliminary data to full datasheet.

Please Read Carefully: