7 SPI Mode

7.1 Introduction

The SPI mode consists of a secondary communication protocol which 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 which are initiated by a start bit and terminated by a stop bit, the SPI channel is byte oriented. Every command or data block is built of 8-bit bytes and is byte aligned to the CS signal (i.e. the length is a multiple of 8 clock cycles).

Similar to the 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 response behavior in the SPI mode differs from the SD mode in the following three aspects:

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

In addition to the command response, every data block sent to the card during write operations will be responded with a special data response token. A data block may be as big as one card write block (WRITE_BL_LEN) and as small as a single byte. Partial block read/write operations are enabled by card options specified in the CSD register.

7.2.1 Mode Selection

The SD Memory Card wakes 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) and the card is in *idle_state*. 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 Memory Card protocol state machine is not observed. All the SD Memory Card commands supported in SPI mode are always available.

7.2.2 Bus Transfer Protection

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

In the non-protected mode the CRC bits of the command, response and data tokens are still required in the tokens. However, they are defined as 'don't care' for the transmitter and ignored by the receiver.

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

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

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

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

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 of the length defined in a previous SET_BLOCKLEN (CMD16) command (refer to Figure 41).

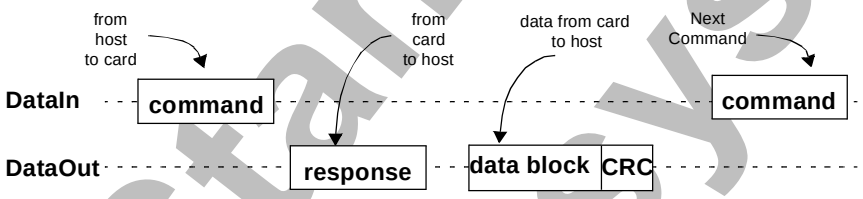


Figure 41: Single Block Read operation

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

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

In case of a data retrieval error, the card will not transmit any data. Instead, a special data error token will be sent to the host. Figure 42 shows a data read operation which terminated with an error

token rather than a data block.

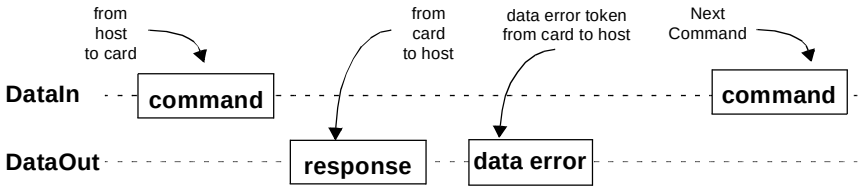


Figure 42: Read operation - data error

In case of Multiple block read operation every transferred block has its suffixed of 16 bit CRC.

Stop transmission command (CMD12) will actually stop the data transfer operation (the same as in SD Memory Card operation mode).

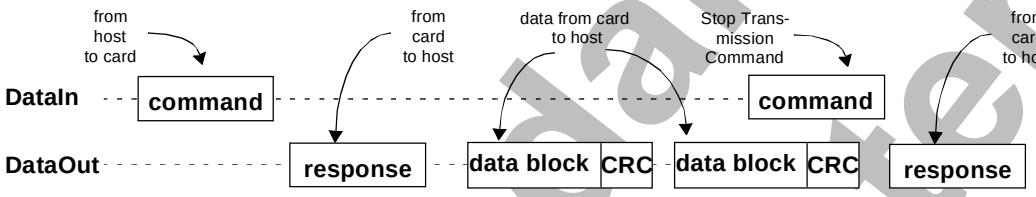


Figure 43: Multiple Block Read operation

7.2.4 Data Write

In SPI mode the SD Memory Card 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) identical to the read operation (see Figure 44).

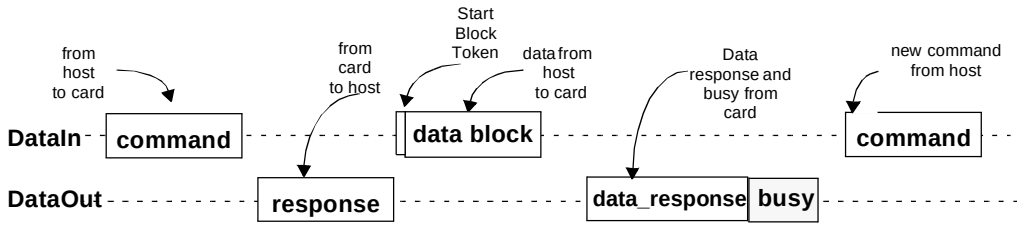


Figure 44: Single Block Write operation

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

In Multiple Block write operation the stop transmission will be done by sending 'Stop Tran' token instead of 'Start Block' token at the beginning of the next block. In case of Write Error indication (on the data response) the host shall use SEND_NUM_WR_BLOCKS (ACMD22) in order to get the number of well written write blocks. The data tokens description is given in Chapter 7.3.3.

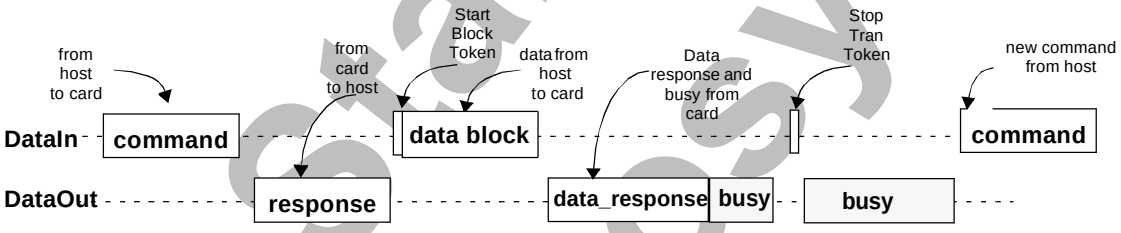


Figure 45: Multiple Block Write operation

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) 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 it.

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 46 illustrates a 'no data' bus transaction with and without busy signalling.

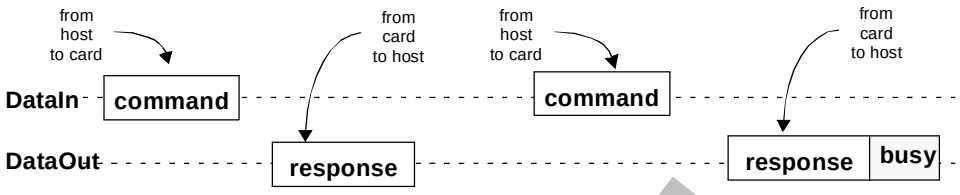


Figure 46: 'No data' operations

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 42) followed by a data block of 16 bytes suffixed with a 16 bit CRC.

The data time out for the CSD command cannot be set to the cards TAAC since this value is stored in the card's CSD. Therefore the standard response time-out 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. After power on reset or CMD0 (software reset) the card enters an idle state. At this state the only legal host commands are CMD1 (SEND_OP_COND), ACMD41 (SD_SEND_OP_COND) and CMD58 (READ_OCR).

In SPI mode CMD1 and ACMD41 have the same behavior.

The host must poll the card (by repeatedly sending CMD1 or ACMD41) until the 'in-idle-state' bit in the card response indicates (by being set to 0) that the card completed its initialization processes and is ready for the next command.

In SPI mode, as opposed to SD mode, CMD1 (and ACMD41 as well) 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. Furthermore, it is in the responsibility of the host to refrain from accessing cards that do not support its voltage range.

The usage of CMD58 is not restricted to the initializing phase only, but can be issued at any time.

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 if it is not supported (illegal opcode), if the CRC check failed, if it contained an illegal operand, or if it was out of sequence during an erase sequence.

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 with a R1b response type. After the busy signal clears, the host should obtain the result of the operation by issuing a GET_STATUS command. Refer to Chapter 4.3.6 for details.

7.2.11 Application Specific commands

Identical to SD mode with the exception of the APP_CMD status bit (refer to Chapter 4.10.1) which is not available in SPI.

7.2.12 Copyright Protection commands

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

7.3 SPI Mode Transaction Packets

7.3.1 Command Tokens

- **Command Format**

All the SD Memory Card commands are 6 bytes long. The command transmission always starts with the left bit of the bitstring corresponding to the command codeword. All commands are protected by a CRC (see Chapter 7.2). The commands and arguments are listed in Table 57.

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 55: Command Format

- **Command Classes**

As in SD mode, the SPI commands are divided into several classes (See Table 56). 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 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	9	10	12	13	16	17	18	24	25	27	28	29	30	32	33	38	42	55	56	58	59
class 0	Basic	+	+	+	+	+	+																+	+
class 1	Not supported in SPI																							
class 2	Block read							+	+	+														
class 3	Not supported in SPI																							
class 4	Block write										+	+	+											
class 5	Erase																+	+	+					
class 6	Write-protection (Optional)													+	+	+								
class 7	Lock Card (Optional)																			+				
class 8	Application specific																				+	+		
class 9	Not supported in SPI																							
class 10-11	Reserved																							

Table 56: Command classes in SPI mode

• Detailed Command Description

The following table provides a detailed description of the SPI bus commands. The responses are defined in Chapter 7.3.2. The Table 57 lists all SD Memory Card commands. A “yes” in the SPI mode colon 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.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	resets the SD Memory Card
CMD1	Yes	None	R1	SEND_OP_ COND	Activates the card’s initialization process.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	reserved				
CMD6	reserved				
CMD7	No				

Table 57: Commands and arguments

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD8	reserved				
CMD9	Yes	None	R1	SEND_CSD	asks the selected card to send its card-specific data (CSD)
CMD10	Yes	None	R1	SEND_CID	asks the selected card to send its card identification (CID)
CMD11	No				
CMD12	Yes	Non	R1b	STOP_ TRANSMISSION	forces the card to stop transmission in Multiple Block Read Operation
CMD13	Yes	None	R2	SEND_STATUS	asks the selected card to send its status register.
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	selects a block length (in bytes) for all following block commands (read and write).[1]
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_ BLOCK	reads a block of the size selected by the SET_BLOCKLEN command.[2]
CMD18	Yes	[31:0] data address	R1	READ_MULTIPL E_BLOCK	continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.
CMD19	reserved				
CMD20	No				
CMD21 ... CMD23	reserved				
CMD24	Yes	[31:0] data address	R1	WRITE_BLOCK	writes a block of the size selected by the SET_BLOCKLEN command. [3]
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPL E_BLOCK	continuously writes blocks of data until stop Tran' token is sent (instead'Start Block').
CMD26	No				
CMD27	Yes	None	R1	PROGRAM_CSD	programming of the programmable bits of the CSD.
CMD28	Yes	[31:0] data address	R1b[4]	SET_WRITE_ PROT	if the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	Yes	[31:0] data address	R1b	CLR_WRITE_ PROT	if the card has write protection features, this command clears the write protection bit of the addressed group.

Table 57: Commands and arguments

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. [5]
CMD31	reserved				
CMD32	Yes	[31:0] data address	R1	ERASE_WR_BLK _START_ADDR	sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address	R1	ERASE_WR_BLK _END_ADDR	sets the address of the last write block of the continuous range to be erased.
CMD34 ... CMD37	reserved				
CMD38	Yes	[31:0] stuff bits	R1b	ERASE	erases all previously selected write blocks
CMD39	No				
CMD40	No				
CMD41	reserved				
CMD42	Yes	[31:0] stuff bits.	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 4.3.6. The size of the Data Block is defined with SET_BLOCK_LEN command.
CMD43 ... CMD54	reserved				
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_ [6]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose / application specific commands. The size of the Data Block shall be defined with SET_BLOCK_LEN command.
CMD57	Reserved				
CMD58	Yes	None	R3	READ_OCR	Reads the OCR register of a card.
CMD59	Yes	[31:1] 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				

Table 57: Commands and arguments

1)The default block length is as specified in the CSD.
2)The data transferred must not cross a physical block boundary unless READ_BLK_MISALIGN is set in the CSD.
3)The data transferred must not cross a physical block boundary unless WRITE_BLK_MISALIGN is set in the CSD.
4)R1b: R1 response with an optional trailing busy signal.
5) 32 write protection bits (representing 32 write protect groups starting at the specified address) followed by 16 CRC bits are transferred in a payload format via the data line. 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.
6) **RD/WR**_: "1" the Host shall get a block of data from the card.
"0" the host sends block of data to the card.

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 Memory Card status. The status fields are given in Table 24
ACMD17	reserved				
ACMD18	yes	--	--	--	Reserved for SD security applications[1]
ACMD19 to ACMD21	reserved				
ACMD22	yes	[31:0] stuff bits	R1	SEND_NUM_WR_ BLOCKS	Send the numbers of the well written (without errors) blocks. Responds with 32bit+CRC data block.
ACMD23	yes	[31:23] stuff bits [22:0]Number of blocks	R1	SET_WR_BLK_ ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for faster Multiple Block WR command). "1"=default (one wr block)[(2)].
ACMD24	reserved				
ACMD25	yes	--	--	--	Reserved for SD security applications[1]
ACMD26	yes	--	--	--	Reserved for SD security applications[1]
ACMD38	yes	--	--	--	Reserved for SD security applications[1]
ACMD39 to ACMD40	reserved				

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
ACMD41	yes	None	R1	SEND_OP_ COND	Activates the card's initialization process.
ACMD42	yes	[31:1] stuff bits [0]set_cd	R1	SET_CLR_CARD_ DETECT	Connect[1]/Disconnect[0] the 50KOhm pull-up resistor on CD/DAT3 (pin 1) of the card. The pull-up may be used for card detection.
ACMD43 ... ACMD49	yes	--	--	--	Reserved for SD security applications[1]
ACMD51	yes	[31:0] staff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

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

Table 58: Application Specific Commands used/reserved by SD Memory Card - SPI Mode

7.3.2 Responses

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

- **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 47. 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, which did not match the block length, was used in the command.
- **Parameter error**: The command's argument (e.g. address, block length) was out of the allowed range for this card.

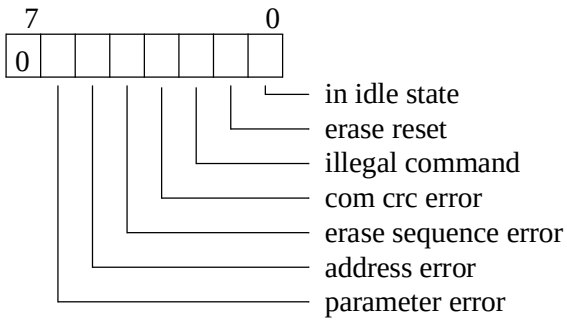


Figure 47: R1 Response Format

- **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.

- **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 48.

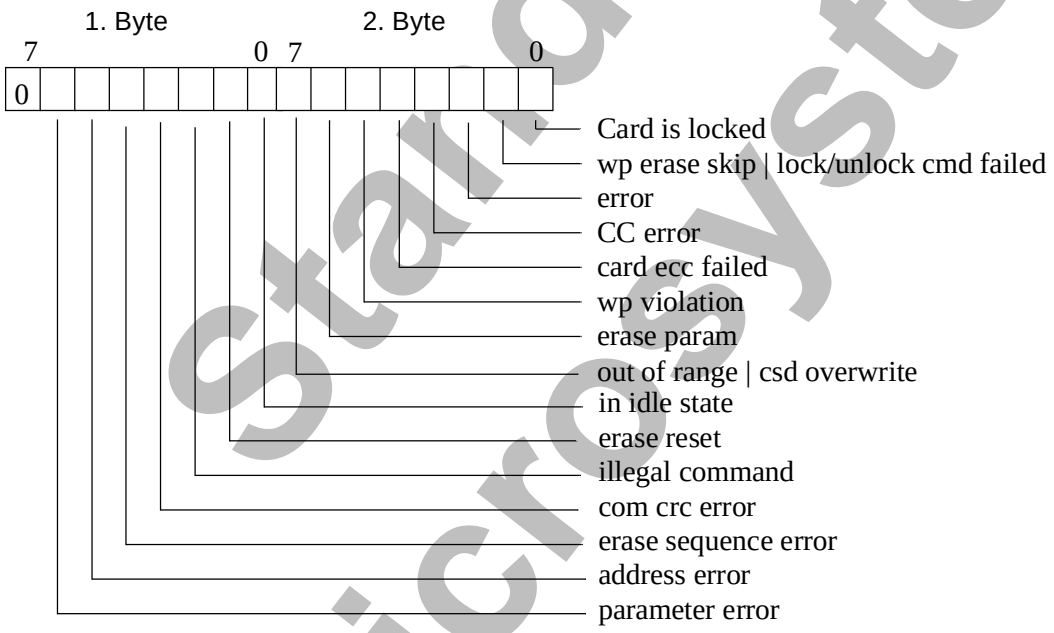


Figure 48: R2 response format

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, sectors or groups, for erase.
- **Write protect violation**: The command tried to write a write protected block.
- **Card ECC failed**: Card internal ECC was applied but failed to correct the data.
- **CC error**: Internal card controller error
- **Error**: A general or an unknown error occurred during the operation.
- **Write protect erase skip | lock/unlock command failed**: This status bit has two functions over-loaded. It is set when the host attempts to erase a write protected sector or makes a sequence or password error during card lock/unlock operation.
- **Card is locked**: Set when the card is locked by the user. Reset when it is unlocked.

- **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 49). The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

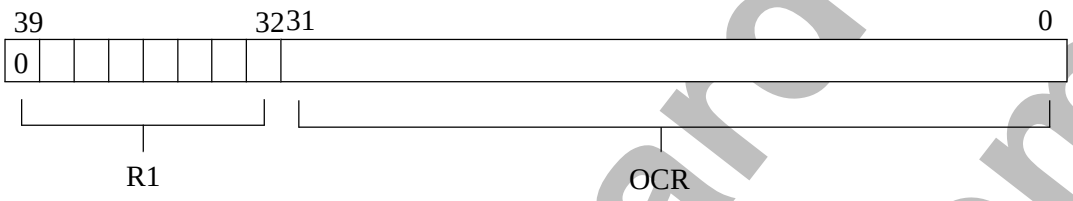


Figure 49: R3 Response Format

- **Data Response**

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:

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

If the host gets status field'110' it shall stop the data transmission using CMD12 and send CMD13 in order to verify which write problem caused the Write Error indication. ACMD22 can be used to find the number of well written write blocks.

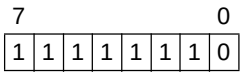
7.3.3 Data Tokens

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

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

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

- First byte: Start Block

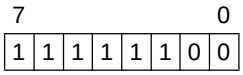


- Bytes 2-513 (depends on the data block length): User data
- Last two bytes: 16 bit CRC.

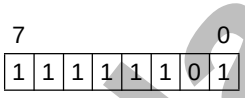
For Multiple Block Write operation:

- First byte of each block:

  If data is to be transferred then - Start Block



If Stop transmission is requested - Stop Tran



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

7.3.4 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:

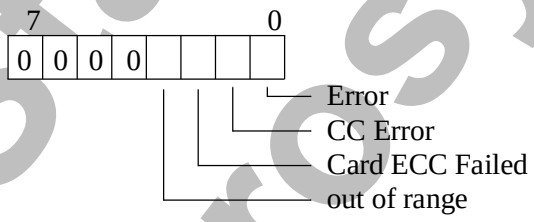


Figure 50: Data Error Token

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

7.3.5 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 multi-

ple 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 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 previous command failed.	C
Illegal command	R1 R2	E R	'0'= no error '1'= error	Command not legal for the card state	C
Card ECC failed	R2 DataEr	E X	'0'= success '1'= failure	Card internal ECC was applied but failed to correct the data.	C
CC error	R2 dataEr	E R X	'0'= no error '1'= error	Internal card controller error	C
Error	R2 dataEr	E R X	'0'= no error '1'= error	A general or an unknown error occurred during the operation.	C
CID/ CSD_OVERWRITE	R2	E R X	'0'= no error '1'= error	can be either one of the following errors: - The CID register has been already written and can not be overwritten - The read only section of the CSD does not match the card content. - An attempt to reverse the copy (set as original) or permanent WP (unprotected) bits was made.	C
WP erase skip	R2	S X	'0'= not protected '1'= protected	Only partial address space was erased due to existing write protected blocks.	C

Table 59: SPI mode status bits

Identifier	Included in resp	Type[1]	Value	Description	Clear Condition[2]
Lock/Unlock cmd failed	R2	X	'0'= no error '1'= error	Sequence or password error 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 and	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 59: 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 must poll the card by issuing the status command in order to read these bits.

2) Clear Condition:

A: According to the card current state.

C: Clear by read

7.4 Card Registers

In SPI mode only the OCR, CSD and CID registers are accessible. Their format is identical to the format in the SD mode. However, a few fields are irrelevant in SPI 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 60. The host must 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

- **Host Command to Card Response - Card is ready**

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

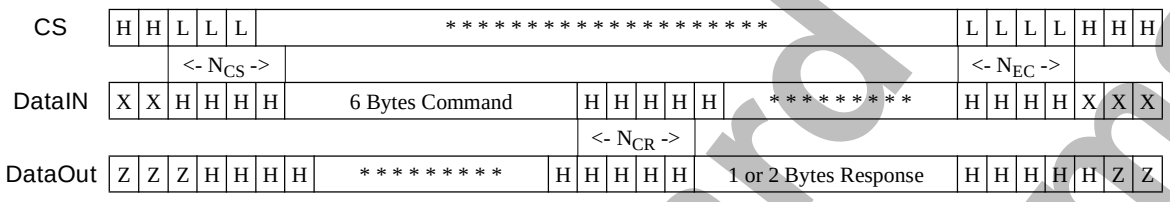


Figure 51: Basic command response

- **Host Command to Card Response - card is busy**

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

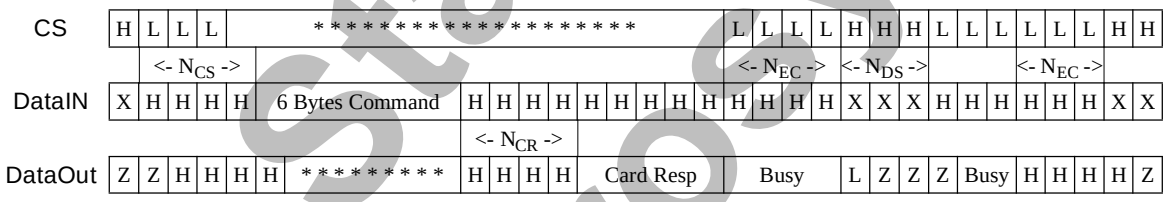


Figure 52: Command response with busy indication (R1b)

- **Card Response to Host Command**

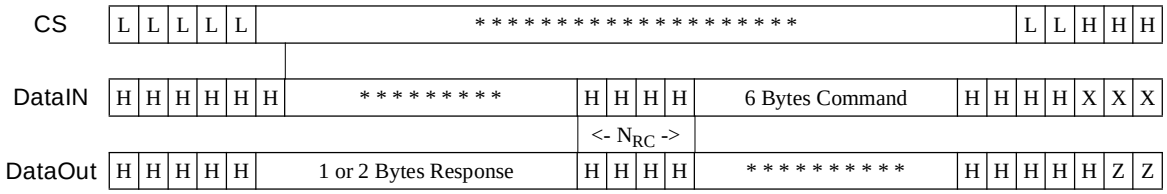


Figure 53: Timing between card response to new host command

7.5.2 Data read

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

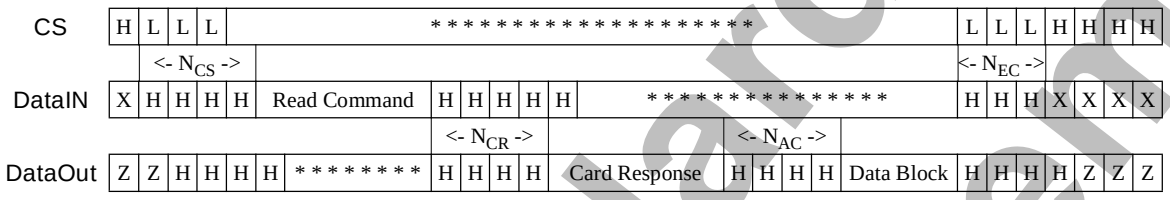


Figure 54: Read Single Block operations - bus timing

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

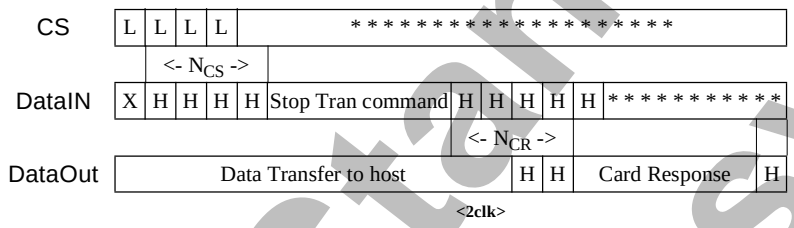


Figure 55: Stop Transmission in Read Multiple Block

- **Reading the CSD register**

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

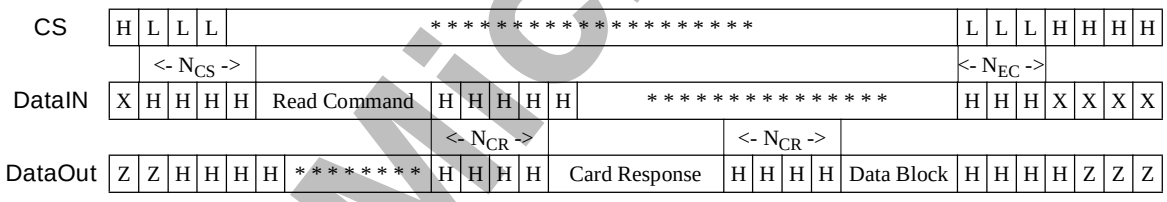


Figure 56: Read CSD - bus timing

7.5.3 Data write

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

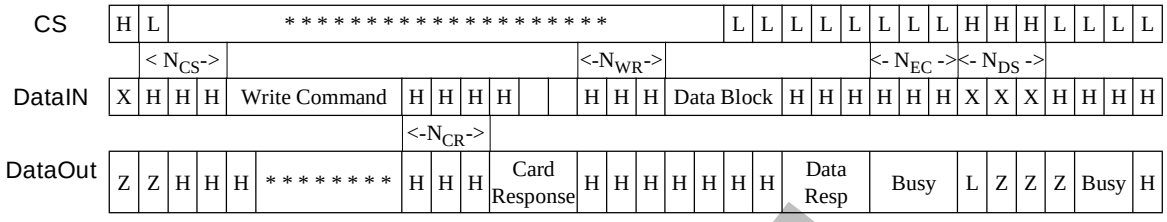


Figure 57: Write operation - bus timing

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

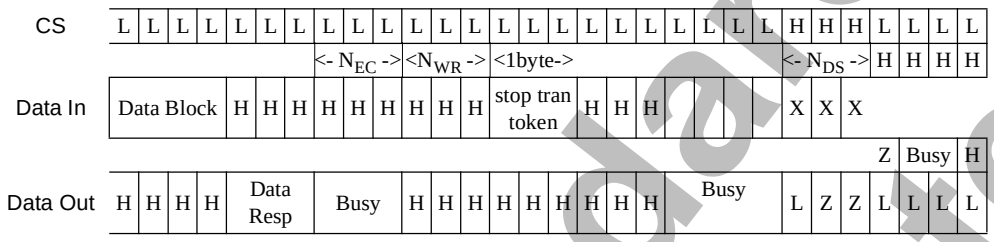


Figure 58: Stop Transmission in Write Multiple Block

7.5.4 Timing Values

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

Table 60: Timing values

7.6 SPI Electrical Interface

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

Identical to SD mode

7.8 Bus Timing

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