7. SPI Mode

7.1 Introduction

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

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

The commands and functions in SD mode defined after the Version 2.00 are not supported in SPI mode. The card may respond to the commands and functions even if the card is in SPI mode but host should not use them in SPI mode.

7.2 SPI Bus Protocol

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

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

Similar to the SD Memory Card protocol, the SPI messages consist of command, response and data-block tokens. All communication between host and cards is controlled by the host (master). The host starts every bus transaction by asserting the CS signal low.

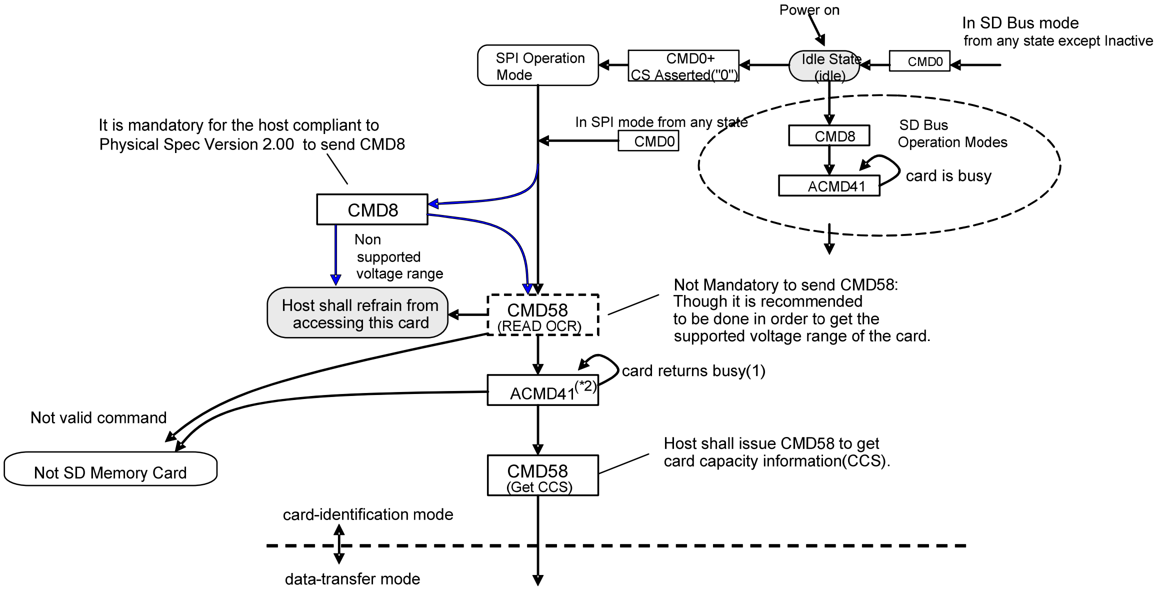
The selected card always responds to the command as opposed to the SD mode.

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

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

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

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


Power on
In SD Bus mode
from any state except Inactive
SPI Operation Mode
CMD0+ CS Asserted("0")
Idle State (idle)
CMD0
In SPI mode from any state
CMD0
CMD8
SD Bus Operation Modes
ACMD41
card is busy
It is mandatory for the host compliant to Physical Spec Version 2.00 to send CMD8
CMD8
Non supported voltage range
Host shall refrain from accessing this card
CMD58 (READ OCR)
Not Mandatory to send CMD58: Though it is recommended to be done in order to get the supported voltage range of the card.
ACMD41(*2)
card returns busy(1)
Not valid command
Not SD Memory Card
CMD58 (Get CCS)
Host shall issue CMD58 to get card capacity information(CCS).
card-identification mode
data-transfer mode

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

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

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

7.2.1 Mode Selection and Initialization

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

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

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

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

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

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

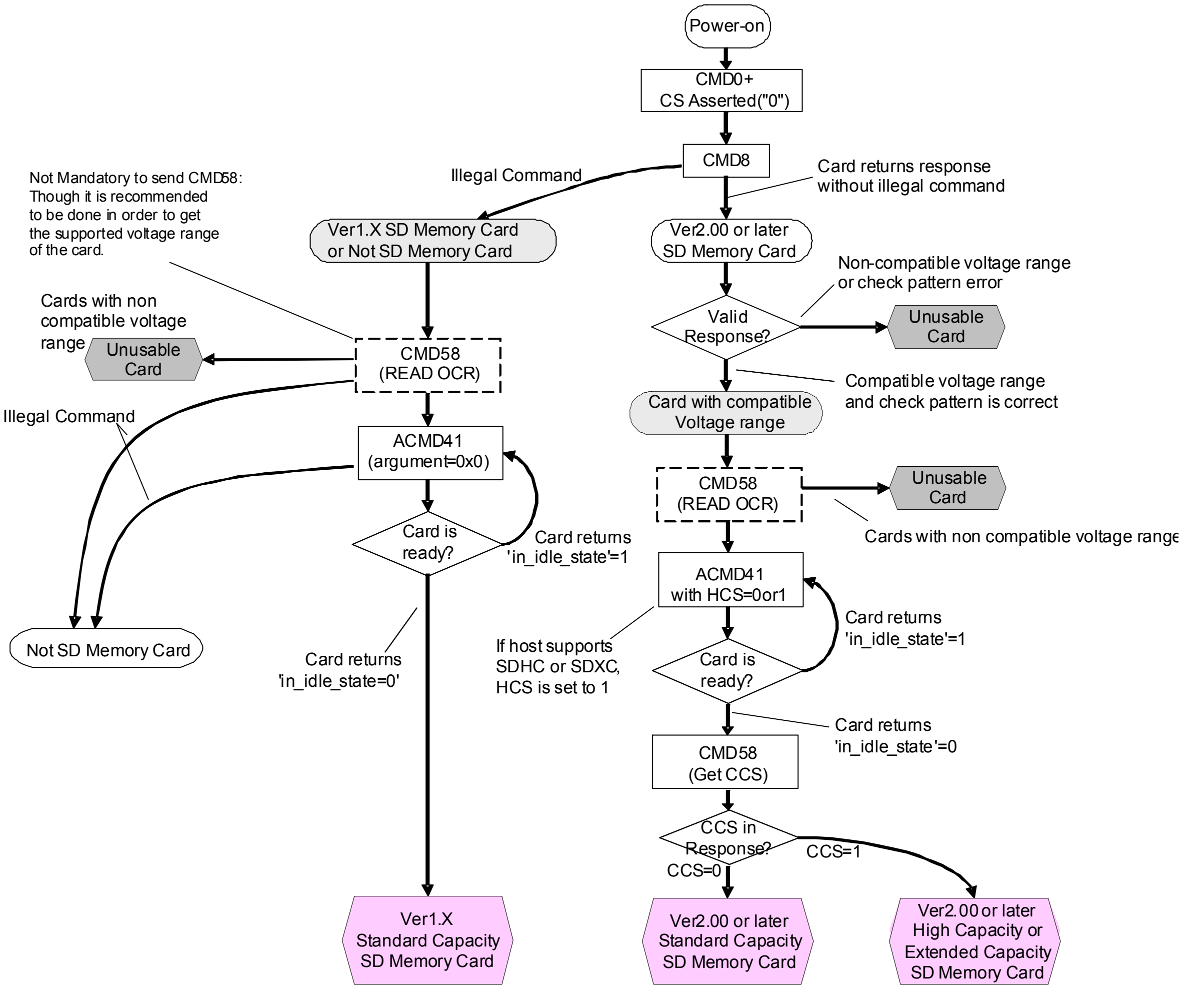


Figure 7-2: SPI Mode Initialization Flow

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

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

After initialization is completed, the host should get CCS information in the response of CMD58. CCS is valid when the card accepted CMD8 and after the completion of initialization. CCS=0 means that the card is SDSD. CCS=1 means that the card is SDHC or SDXC.

7.2.2 Bus Transfer Protection

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

In the CRC OFF mode, the CRC bits of the command are defined as 'don't care' for the transmitter and ignored by the receiver.

The SPI interface is initialized in the CRC OFF mode in default. However, the RESET command (CMD0) that is used to switch the card to SPI mode, is received by the card while in SD mode and, therefore, shall have a valid CRC field.

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

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

After the card is put into SPI mode, CRC check for all commands including CMD0 will be done according to CMD59 setting.

The host can turn the CRC option on and off using the CRC_ON_OFF command (CMD59). Host should enable CRC verification before issuing ACMD41.

The CMD8 CRC verification is always enabled. The Host shall set correct CRC in the argument of CMD8. If CRC error is detected, card returns CRC error in R1 response regardless of command index.

7.2.3 Data Read

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

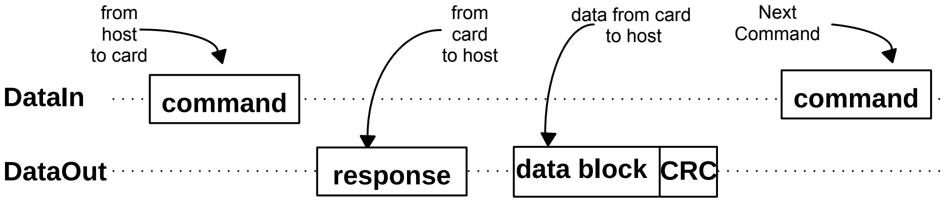


Figure 7-3: Single Block Read Operation

A valid data block is suffixed with a 16-bit CRC generated by the standard CCITT polynomial $x^{16}+x^{12}+x^5+1$.

The maximum block length is given by 512 Bytes regardless of READ_BL_LEN, defined in the CSD. If partial block access is enabled in Standard Capacity Card (i.e. the CSD parameter READ_BL_PARTIAL equals 1), the block length can be any number between 1 and 512 Bytes. The start address can be any byte address in the valid address range of the card. Every block, however, shall be contained in a single physical card sector.

If partial block access is disabled, only 512-Byte data length is supported.

SDHC and SDXC Cards only support 512-byte block length. The start address shall be aligned to the block boundary.

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

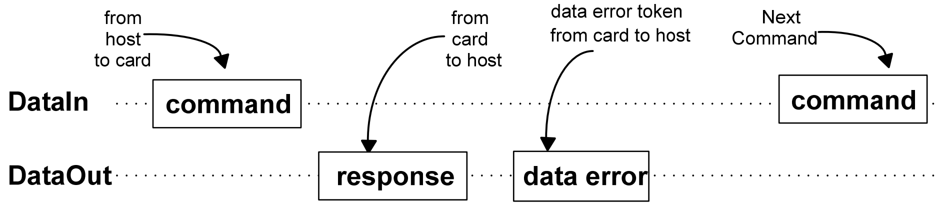


Figure 7-4: Read Operation - Data Error

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

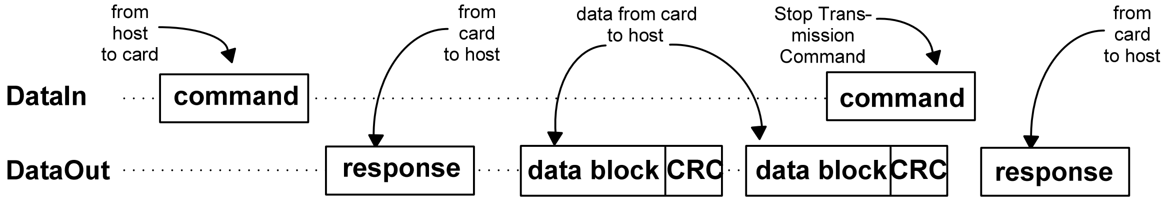


Figure 7-5: Multiple Block Read Operation

7.2.4 Data Write

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

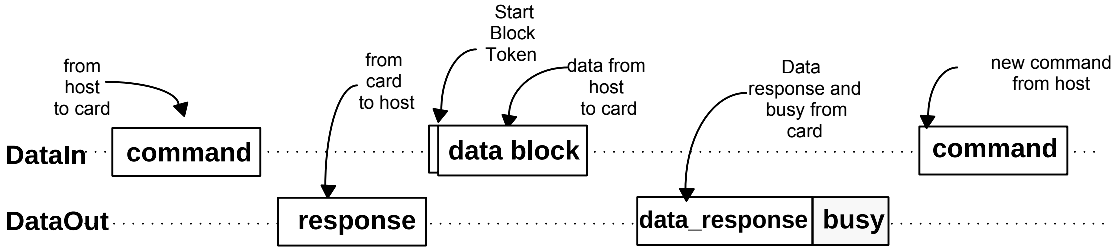


Figure 7-6: Single Block Write Operation

Every data block has a prefix of 'Start Block' token (one byte).
After a data block has been received, the card will respond with a data-response token. If the data block has been received without errors, it will be programmed. As long as the card is busy programming, a continuous stream of busy tokens will be sent to the host (effectively holding the DataOut line low).
Once the programming operation is completed, the host should check the results of the programming using the SEND_STATUS command (CMD13). Some errors (e.g. address out of range, write protect violation etc.) are detected during programming only. The only validation check performed on the data block, and communicated to the host via the data-response token, is the CRC and general Write Error

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

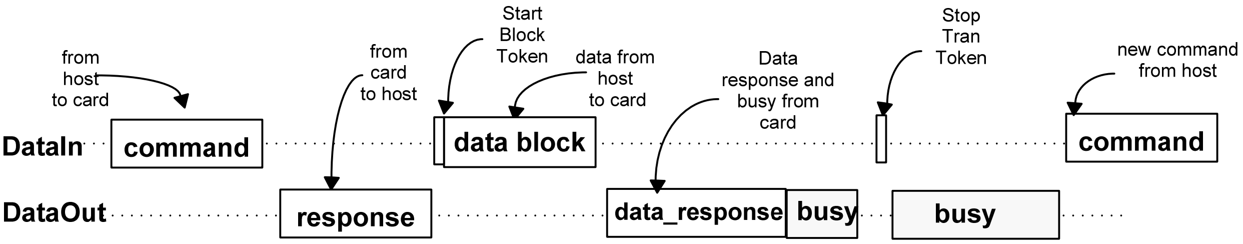


Figure 7-7: Multiple Block Write Operation

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

7.2.5 Erase & Write Protect Management

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

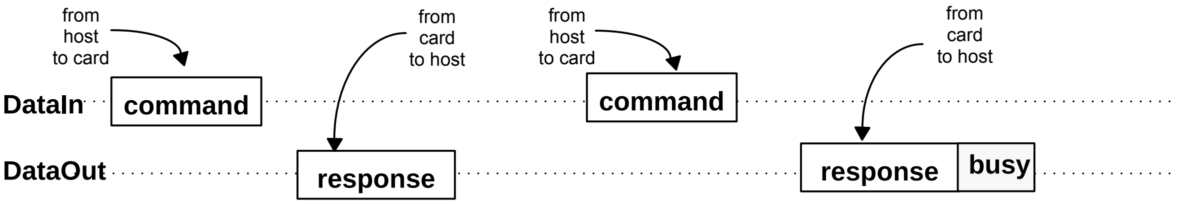


Figure 7-8: 'No data' Operations

7.2.6 Read CID/CSD Registers

Unlike the SD Memory Card protocol (where the register contents is sent as a command response), reading the contents of the CSD and CID registers in SPI mode is a simple read-block transaction. The card will respond with a standard response token (see Figure 7-3) followed by a data block of 16 bytes suffixed with a 16-bit CRC.

The data timeout for the CSD command cannot be set to the cards TAAC since this value is stored in the card's CSD. Therefore, the standard response timeout value (N_{CR}) is used for read latency of the CSD register.

7.2.7 Reset Sequence

The SD Memory Card requires a defined reset sequence. The card enters an idle state after power on reset or reset command (CMD0 for SD memory card). In this state, the only valid host commands are CMD8 (SEND_IF_COND), ACMD41 (SD_SEND_OP_COND), CMD58 (READ_OCR) and CMD59 (CRC_ON_OFF).

For the Thick (2.1 mm) SD Memory Card - CMD1 (SEND_OP_COND) is also valid - this means that in SPI mode, CMD1 and ACMD41 have the same behaviors, but the usage of ACMD41 is preferable since it allows easy distinction between an SD Memory Card and a MultiMediaCard. **For the Thin (1.4 mm) Standard Size SD Memory Card, CMD1 (SEND_OP_COND) is an illegal command during the initialization that is done after power on. After Power On, once the card has accepted valid ACMD41, it will be able to also accept CMD1 even if used after re-initializing (CMD0) the card.** It was defined in such way in order to be able to distinguish between a Thin SD Memory Card and a MultiMediaCard (that supports CMD1 as well).

7.2.8 Error Conditions

Unlike the SD Memory Card protocol, in the SPI mode, the card will always respond to a command. The response indicates acceptance or rejection of the command. A command may be rejected in any one of the following cases:

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

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

7.2.9 Memory Array Partitioning

Same as SD mode.

7.2.10 Card Lock/Unlock

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

7.2.11 Application Specific Commands

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

7.2.12 Content Protection Command

All the special Content Protection ACMDs and security functionality related to the CPRM is the same as SD mode.

7.2.13 Switch Function Command

Same as for SD mode with two exceptions:

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

7.2.14 High Speed Mode

Same as SD mode.

7.2.15 Speed Class Specification

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

7.3 SPI Mode Transaction Packets

7.3.1 Command Tokens

7.3.1.1 Command Format

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

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

Table 7-1: Command Format

7.3.1.2 Command Classes

As in SD mode, the SPI commands are divided into several classes (See Table 7-2). Each class supports a set of card functions. A SD Memory Card will support the same set of optional command classes in both communication modes (there is only one command class table in the CSD register). The available command classes, and the supported command for a specific class, however, are different in the SD Memory Card and the SPI communication mode.

Note that except for the classes that are not supported in SPI mode (class 1, 3 and 9), the mandatory required classes for the SD mode are the same for the SPI mode.

Card Command Class (CCC)		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Supported commands	**class description**	**basic**	**reser ved**	**block read**	**reser ved**	**block write**	**erase**	**write pro-tec-tion**	**lock card**	**appli-cat-ion spe-cific**	**I/O mode**	**switch**	**reser ved**
CMD0	Mandatory	+											
CMD1	Mandatory	+											
CMD5	Optional										+		
CMD6[2]	Mandatory											+	
CMD8[3]	Mandatory	+											
CMD9	Mandatory	+											
CMD10	Mandatory	+											
CMD12	Mandatory	+											
CMD13	Mandatory	+											
CMD16	Mandatory			+		+			+				
CMD17	Mandatory			+									
CMD18	Mandatory			+									
CMD24	Mandatory[1]					+							
CMD25	Mandatory[1]					+							
CMD27	Mandatory[1]					+							

Card Command Class (CCC)		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
Supported commands	**class description**	**basic**	**reserved**	**block read**	**reserved**	**block write**	**erase**	**write protection**	**lock card**	**application specific**	**I/O mode**	**switch**	**reserved**
CMD28	Optional							+					
CMD29	Optional							+					
CMD30	Optional							+					
CMD32	Mandatory[1]						+						
CMD33	Mandatory[1]						+						
CMD34-37[2]	Optional											+	
CMD38	Mandatory[1]						+						
CMD42[4]	(Note 4)								+				
CMD50[2]	Optional											+	
CMD52	Optional										+		
CMD53	Optional										+		
CMD55	Mandatory									+			
CMD56	Mandatory									+			
CMD57[2]	Optional											+	
CMD58	Mandatory	+											
CMD59	Mandatory	+											
ACMD6	Mandatory									+			
ACMD13	Mandatory									+			
ACMD22	Mandatory[1]									+			
ACMD23	Mandatory[1]									+			
ACMD41	Mandatory									+			
ACMD42	Mandatory									+			
ACMD51	Mandatory									+			

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

Table 7-2: Command Classes in SPI Mode

7.3.1.3 Detailed Command Description

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

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	[31:0] stuff bits	R1	GO_IDLE_STATE	Resets the SD Memory Card
CMD1	Yes[1]	[31]Reserved bit [30]HCS [29:0]Reserved bits	R1	SEND_OP_COND	Sends host capacity support information and activates the card's initialization process. HCS is effective when card receives SEND_IF_COND command. Reserved bits shall be set to '0'.
CMD2	No				
CMD3	No				
CMD4	No				
CMD5	Reserved for I/O Mode (refer to the "SDIO Card Specification")				
CMD6[8]	Yes	[31] Mode 0:Check function 1:Switch function [30:24] reserved (All '0') [23:20] reserved for function group 6 (All '0' or 0xF) [19:16] reserved for function group 5 (All '0' or 0xF) [15:12] reserved for function group 4 (All '0' or 0xF) [11:8] reserved for function group 3 (All '0' or 0xF) [7:4] function group 2 for command system [3:0] function group 1 for access mode	R1	SWITCH_FUNC	Checks switchable function (mode 0) and switches card function (mode 1). See Chapter 4.3.10.
CMD7	No				

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD8[9]	Yes	[31:12]Reserved bits [11:8]supply voltage(VHS) [7:0]check pattern	R7	SEND_IF_COND	Sends SD Memory Card interface condition that includes host supply voltage information and asks the accessed card whether card can operate in supplied voltage range. Reserved bits shall be set to '0'.
CMD9	Yes	[31:0] stuff bits	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD)
CMD10	Yes	[31:0] stuff bits	R1	SEND_CID	Asks the selected card to send its card identification (CID)
CMD11	No				
CMD12	Yes	[31:0] stuff bits	R1b[5]	STOP_ TRANSMISSION	Forces the card to stop transmission in Multiple Block Read Operation
CMD13	Yes	[31:0] stuff bits	R2	SEND_STATUS	Asks the selected card to send its status register.
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	In case of SDSC Card, block length is set by this command. In case of SDHC and SDXC Cards, block length of the memory access commands are fixed to 512 bytes. The length of LOCK_UNLOCK command is set by this command regardless of card capacity.
CMD17	Yes	[31:0] data address[10]	R1	READ_SINGLE_ BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.[3]
CMD18	Yes	[31:0] data address[10]	R1	READ_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[10]	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.[4]
CMD25	Yes	[31:0] data address[10]	R1	WRITE_MULTIPL E_BLOCK	Continuously writes blocks of data until 'Stop Tran' token is sent (instead 'Start Block').
CMD26	No				
CMD27	Yes	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD28	Yes	[31:0] data address	R1b[5]	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE). SDHC and SDXC Cards do not support this command.
CMD29	Yes	[31:0] data address	R1b[5]	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group. SDHC and SDXC Cards do not support this command.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits.[6] SDHC and SDXC Cards do not support this command.
CMD31	reserved				
CMD32	Yes	[31:0] data address[10]	R1	ERASE_WR_BLK_START_ADDR	Sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address[10]	R1	ERASE_WR_BLK_END_ADDR	Sets the address of the last write block of the continuous range to be erased.
CMD34-37[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD38	Yes	[31:0] stuff bits	R1b[5]	ERASE	Erases all previously selected write blocks
CMD39	No				
CMD40	No				
CMD41	Reserved				
CMD42	Yes	[31:0] Reserved bits (Set all 0)	R1	LOCK_UNLOCK	Used to Set/Reset the Password or lock/unlock the card. A transferred data block includes all the command details - refer to Chapter 4.3.7. The size of the Data Block is defined with SET_BLOCK_LEN command. Reserved bits in the argument and in Lock Card Data Structure shall be set to 0.
CMD43-49 CMD51	reserved				
CMD50[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD52-54	Reserved for I/O Mode (refer to the "SDIO Card Specification")				

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD55	Yes	[31:0] stuff bits	R1	APP_CMD	Defines to the card that the next command is an application specific command rather than a standard command
CMD56	Yes	[31:1] stuff bits. [0]: RD/WR[7]	R1	GEN_CMD	Used either to transfer a Data Block to the card or to get a Data Block from the card for general purpose/application specific commands. In case of Standard Capacity SD Memory Card, the size of the Data Block shall be defined with SET_BLOCK_LEN command. In case of SDHC and SDXC Cards, block length of this command is fixed to 512-byte.
CMD57[8]	Reserved for each command system set by switch function command (CMD6). Refer to each command system specification for more detail.				
CMD58	Yes	[31:0] stuff bits	R3	READ_OCR	Reads the OCR register of a card. CCS bit is assigned to OCR[30].
CMD59	Yes	[31:1] stuff bits [0:0] CRC option	R1	CRC_ON_OFF	Turns the CRC option on or off. A '1' in the CRC option bit will turn the option on, a '0' will turn it off
CMD60-63	Reserved For Manufacturer				

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

Table 7-3: Commands and Arguments

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

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

(1) Refer to the "Part3 Security Specification" for detailed explanation about the SD Security Features

(2) Stop Tran Token shall be used to stop the transmission in Write Multiple Block whether the pre-erase (ACMD23) feature is used or not.

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

7.3.1.4 Card Operation for CMD8 in SPI mode

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

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

*1: Response indicates the actual response that the card returns. (It does not include errors during transfer response.)

*2: 'Match' means AND of following condition a) and b). 'Mismatch' is other cases.

a) Only 1 bit is set to '1' in VHS.

b) The card supports the host supply voltage.

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

7.3.2 Responses

There are several types of response tokens. As in SD mode, all are transmitted MSB first.
Multiple bytes responses are defined in SPI mode but the card outputs only first byte (equivalent to R1) when Illegal Command Error or Command CRC Error is indicated in it. In this case, host never reads as the multiple bytes of response.

7.3.2.1 Format R1

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

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

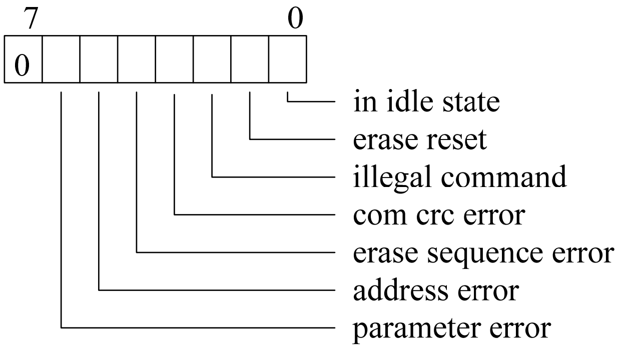


Figure 7-9: R1 Response Format

7.3.2.2 Format R1b

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

7.3.2.3 Format R2

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

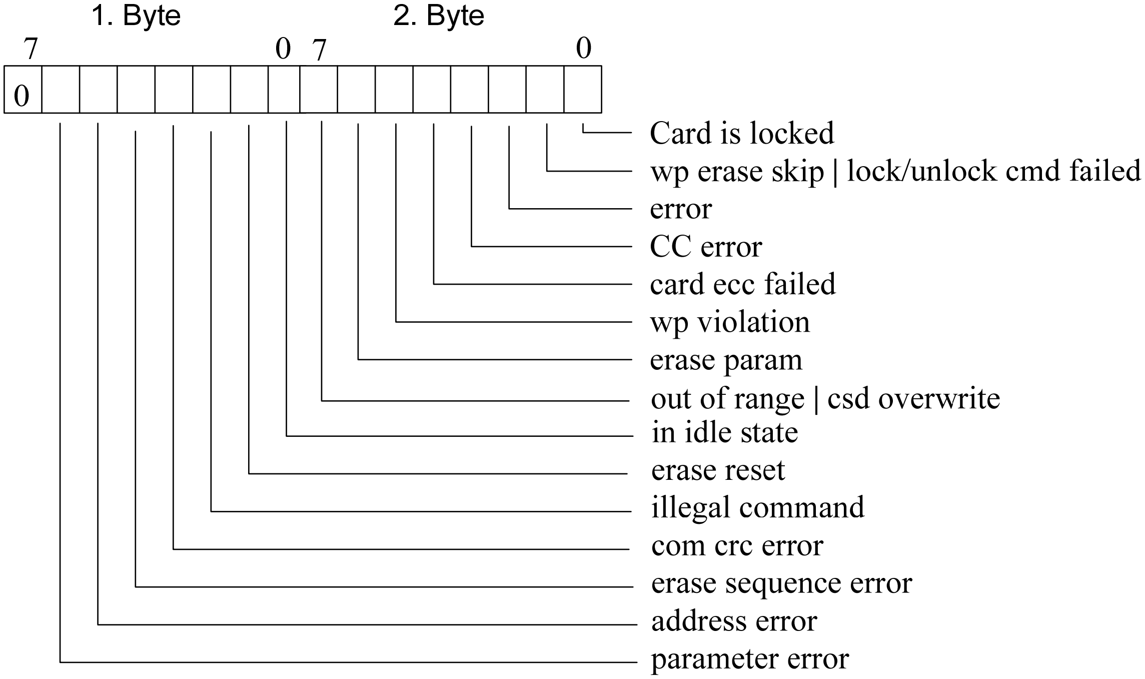


Figure 7-10: R2 Response Format

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

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

7.3.2.4 Format R3

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

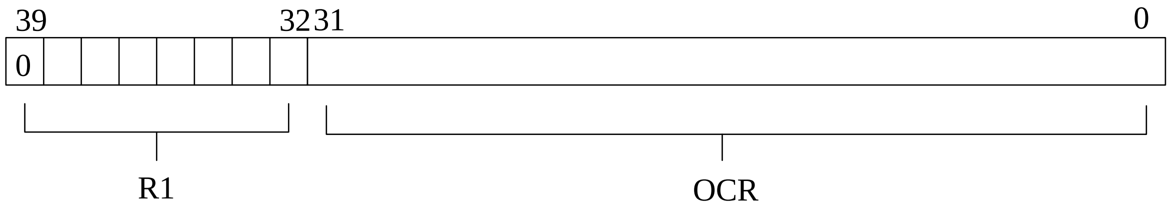


Figure 7-11: R3 Response Format

7.3.2.5 Formats R4 & R5

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

7.3.2.6 Format R7

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

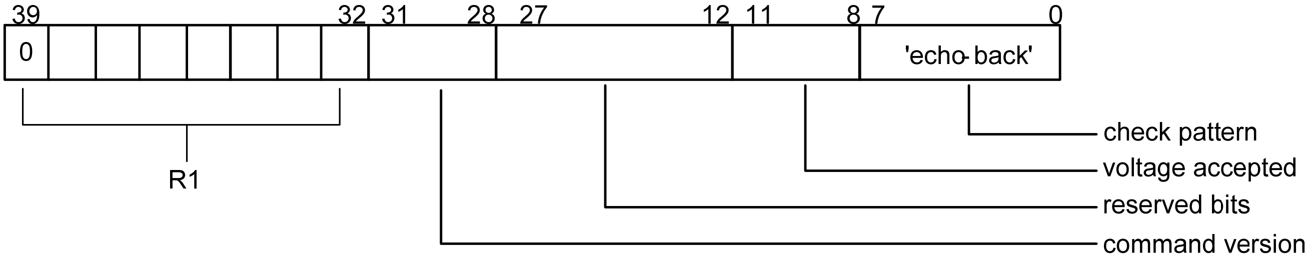


Figure 7-12: R7 Response Format

7.3.3 Control Tokens

Data block transfer is controlled by some tokens.

7.3.3.1 Data Response Token

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

7	6	5	4	3	2	1	0
x	x	x	0	Status			1

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

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

7.3.3.2 Start Block Tokens and Stop Tran Token

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

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

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

- First byte: Start Block

7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	0

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

For Multiple Block Write operation:

- First byte of each block:
  If data is to be transferred then - Start Block Token

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	0

If Stop transmission is requested - Stop Tran Token

7	6	5	4	3	2	1	0
1	1	1	1	1	1	0	1

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

7.3.3.3 Data Error Token

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

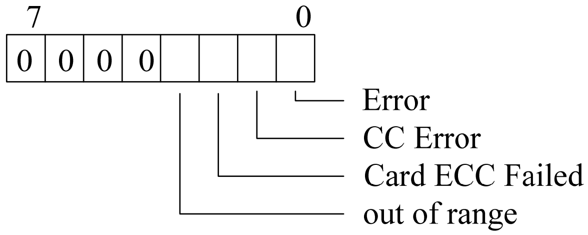


Figure 7-13: Data Error Token

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

7.3.4 Clearing Status Bits

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

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

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

Table 7-6: SPI Mode Status Bits

1) Type:

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

2) Clear Condition:

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

7.4 Card Registers

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

7.5 SPI Bus Timing Diagrams

This section is a blank for the Simplified Specification.

7.6 SPI Electrical Interface

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

7.7 SPI Bus Operating Conditions

Bus operating conditions are identical to SD mode

7.8 Bus Timing

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