1.General Description

The MultiMediaCard MX53L1281 is a highly integrated read only memory (ROM) with serial and random access capability using an innovative ultra high density cell design in the memory array. It is accessible via a dedicated serial interface optimized for fast and reliable data transmission. This interface allows several cards to be stacked by connecting their peripheral contacts. The MX53L1281 is fully compatible to a new consumer standard, called the MultiMediaCard system standard defined in the MultiMediaCard system specification [1].

The MultiMediaCard system is a new mass-storage system based on innovations in semiconductor technology. It has been developed to provide an inexpensive, mechanically robust storage medium in card form for multimedia consumer applications. MultiMediaCard allows the design of inexpensive players and drives without moving parts. A low power consumption and a wide supply voltage range favors mobile, battery-powered applications such as audio players, organizers, palmtops, electronic books, encyclopedia and dictionaries. Using very effective data compression schemes such as MPEG, the MultiMediaCard will deliver enough capacity for all kinds of multimedia data: software/programs, text, music, speech, images, video etc.

2. FEATURES

- 16 MByte memory capacity
  - Payload: 16,777,216 Bytes
- Small card-sized package: 24x32x1.4 mm (WxLxH)
- MultiMediaCard system standard compatibility
  - Sequential and block read supported (Command classes 0, 1 and 2)
  - Block size free programmable between 1~2048 byte in MMC mode, 1~512 byte in SPI mode
  - Multiple block mode supported in MMC mode
  - CRC protected data communication
  - ***2.0V to 3.6V*** operation voltage range of communication
  - ***2.7V to 3.6V*** operation voltage range of memory access
  - Damage free powered card insertion and removal
  - MMC and SPI mode available
- High speed serial interface with random access in block or serial mode
  - Byte addressable memory
  - up to 10 stacked card @ 20MHz @ ***2.7-3.6V***
  - up to 30 stacked card @ 5MHz @ ***2.7-3.6V***
  - Access time < 15 us @ 20MHz @ ***2.7-3.6V***, random byte access
- Low power dissipation
  - High speed: < 126 mW @ 20MHz @ ***3.6V***
  - Low power: < 13.5 mW @ 100kHz@ ***2.7V***
  - Power save: < 0.27 mW@ 0Hz @ ***2.7V*** (in stby state)

3.OVERVIEW

The following diagram shows an overview of the MX53L1281 internal architecture:

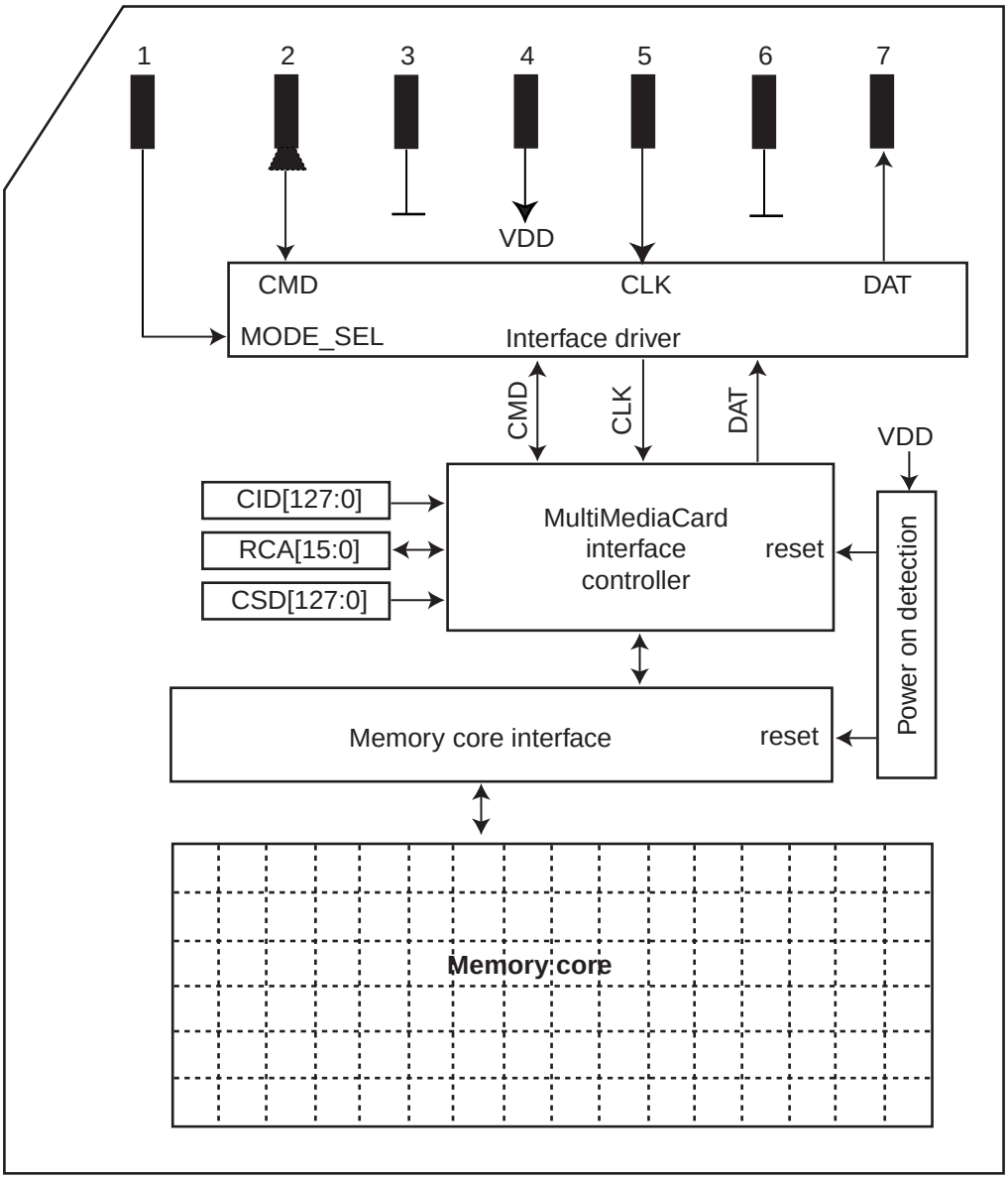


Figure 1 : MX53L1281 architecture

All controllers in the MX53L1281 are clocked by the interface signal CLK. The card is controlled by the three line MultiMediaCard interface containing the signals: CMD, CLK, DAT (see "Chapter 4: Inter-face" for more details). For the identification of the MX53L1281 in a stack of MultiMediaCards a card identification register (CID) and a relative card address register (RCA) is foreseen. An additional register contains different types of operation parameters. This register is called card specific data register (CSD). The communication using the MultiMediaCard lines to access either the memory field or the registers is defined by the MultiMediaCard standard (see "Chapter 6: Communication").

The card has its own power on detection unit. No additional master reset signal is required to setup the card after power on. It is protected against shortcut during insertion and removal while the Multi-MediaCard system is powered up (see "Chapter 9: Power supply").

4. INTERFACE

In the MX53L1281 all data is transferred over a minimal number of lines:

- CLK: with each cycle of this signal a one bit transfer on the command and data lines is done. The frequency may vary between zero and the maximum clock frequency. The MultiMediaCard bus master is free to generate these cycles without restrictions in the range of 0-20MHz.
- CMD: is a bidirectional command channel used for card initialization and data transfer commands. The CMD signal has two operation modes: open drain for initialization mode and push pull for fast command transfer. Commands are sent from the MultiMediaCard bus master to the MX53L1281 and responses vice versa.
- DAT: is a data channel with a width of one line. The DAT signal of the MX53L1281 operates in push pull mode.

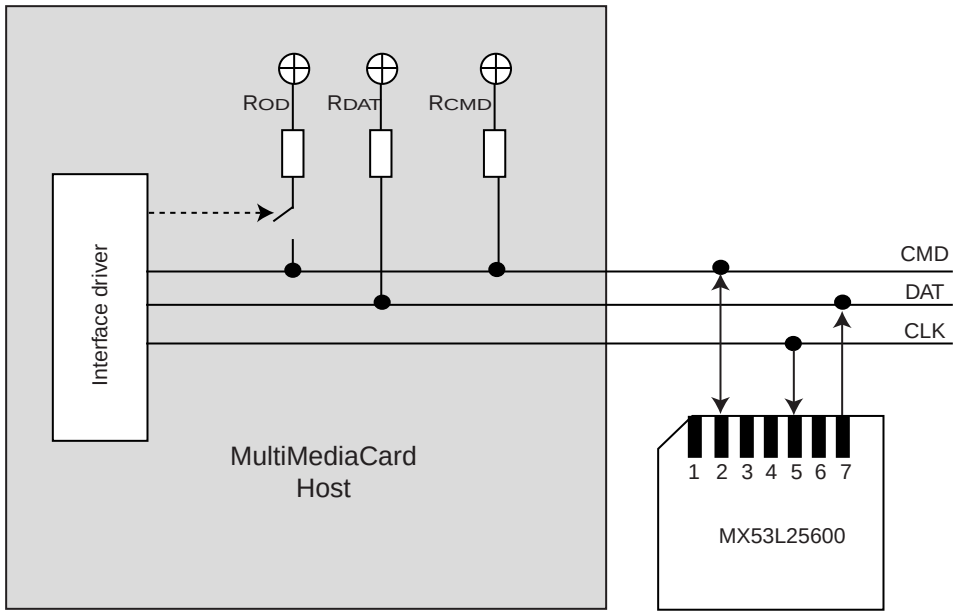


Figure 2: MX53L1281 interface

All MultiMediaCards are connected directly to the lines of the MultiMediaCard bus. The following table defines the card contacts.

Pin No.	Name	Type [1]	Description
1	NC	--	not connected
2	CMD	I/PP/OD	Command/Response
3	VSS1	S	Supply voltage ground
4	VDD	S	Supply voltage
5	CLK	I	Clock
6	VSS2	S	Supply voltage ground
7	DAT	PP	Data output

Table 1: MX53L1281 pad definition

[1] S: power supply; I: input; PP: push pull output; OD: open drain output

Pin 1 is not connected in the MX53L1281

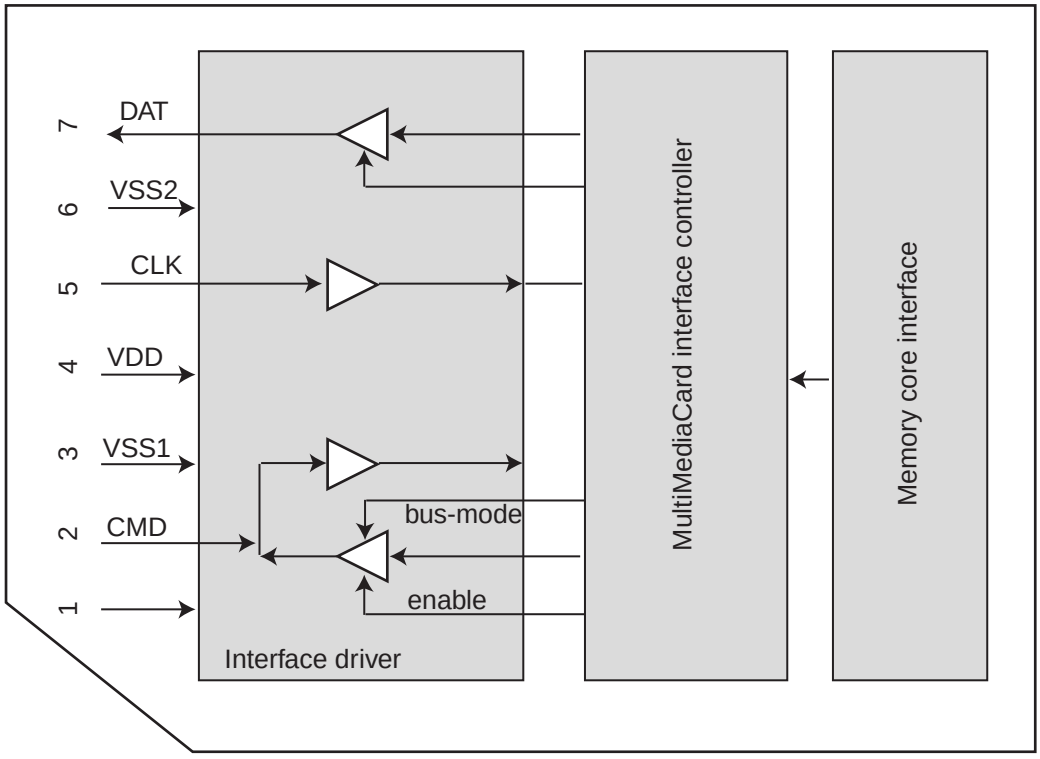


Figure 3: MX53L1281 I/O-drivers

5 Registers

The MX53L1281 has the following information registers:

Name	Width	Type	Description
CID	128	Mask programmable, read only for user	Card identification number, card individual number for identification.
RCA	16	Programmed during initialization, not readable	Relative card address, local system address of a card, dynamically assigned by the host during initialization.
CSD	128	Read only	Card specific data, information about the card operation conditions.

Table 2: MX53L1281 registers

CID and RCA are used for identifying and addressing the MX53L1281. The third register contains the card specific data record. This record is a set of information fields to define the operation conditions of the MX53L1281.

For the user the CID and the CSD are read only registers. They are read out by special commands (see "Chapter 6.1: Commands"). The RCA register is a write only register. Unlike CID and CSD, RCA looses its contents after powering down the card. Its value is reassigned in each initialization cycle. The complete CID and parts of the CSD are programmed by the content provider via the programming mask (see "chapter 8: Programming mask format").

5.1 Card identification (CID)

The Card IDentification (CID) register is 128 bits wide. It contains the card identification information used during the card identification phase (MultiMediaCard protocol). Every individual flash or I/O card shall have an unique identification number. Every type of MultiMediaCard ROM cards (defined by content) shall have an unique identification number.

The structure of the CID register is defined in the following paragraphs:

Name	Field	Width	CID-slice
Manufacturer ID	MID	8	[127:120]
OEM/Application ID	OID	16	[119:104]
Product name	PNM	48	[103:56]
Product revision	PRV	8	[55:48]
Product serial number	PSN	32	[47:16]
Manufacturing date	MDT	8	[15:8]
CRC7 checksum	CRC	7	[7:1]
not used, always '1''	-	1	[0:0]

Table 3: The CID fields

• MID
An 8 bit binary number that identifies the card manufacturer. The MID number is controlled, defined and allocated to a MultiMediaCard manufacturer by the MMCA. This procedure is established to ensure uniqueness of the CID register.

• OID
A 16 bit binary number that identifies the card OEM and/or the card contents (when used as a distri-bution media either on ROM or FLASH cards). The OID number is controlled, defined and allocated to a MultiMediaCard manufacturer by the MMCA. This procedure is established to ensure uniqueness of the CID register.

• PNM
The product name is a string, 6 ASCII characters long.

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

• PSN
A 32 bits unsigned binary integer.

• MDT
The manufacturing date is composed of two hexadecimal digits, four bits each, representing a two digits date code m/y;
The "m" field, most significant nibble, is the month code. 1 = January.
The "y" field, least significant nibble, is the year code. 0 = 1997.

As an example, the binary value of the MDT field for production date "April 2000" will be: 0100 0011.

• CRC
CRC7 checksum (7 bits). This is the checksum of the CID contents.

The CID has to be error free. To ensure the correctness of the CID a CRC checksum is added to the end of the CID. The CRC checksum is computed by the following formula:
CRC Calculation: $G(x) = x^7 + x^3 + 1$
$M(x) = CID[127]*x^{119} + ... + CID[8]*x^0$ $\quad CRC[6...0] = Remainder\ [(M(x)*x^7)/G(x)]$

In the MX53L1281 the CID is programmed with parameters defined by the content provider. The programming is done by the mask which is used for the data programming too. Details of the mask programming and the formats of data transfer between content provider and card manufacturer are defined in "Chapter 8: Programming mask format".

5.2 Relative card address (RCA)

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

The RCA is programmed with the command SET_RELATIVE_ADDRESS (CMD3) during the initialization procedure. The content of this register is lost after power down. The default value is assigned when an internal reset is applied by the power up detection unit of the MX53L1281.

5.3 Card specific data (CSD)

The card specific data register describes how to access the card content. The CSD defines the data format, error correction type, maximum data access time, data transfer speed, whether the wide or standard bus is implemented etc.

CSD-slice	Width	Value	Field
[127:126]	2	1	CSD_STRUCTURE
[125:122]	4	2	SPEC_VERS
[121:120]	2	don't care	_
[119:112]	8	0x08(1ns)	TAAC
[111:104]	8	0x03(300 cycles)	NSAC
[103:96]	8	0x2A(20 Mbit/s)	TRAN_SPEED
[95:84]	12	0x007 (class 0,1,2)	CCC
[83:80]	4	0xB (2048 bytes)	READ_BLK_LEN
[79:79]	1	"1"	READ_BLK_PARTIAL
[78:78]	1	don't care	WRITE_BLK_MISALIGN
[77:77]	1	"1"	READ_BLK_MISALIGN
[76:76]	1	"0"	DSR_IMP
[75:74]	2	don't care	-
[73:62]	12	0xF	C_SIZE
[61:59]	3	0x4(25mA)	VDD_R_CURR_MIN
[58:56]	3	0x4(35mA)	VDD_R_CURR_MAX
[55:53]	3	don't care	VDD_W_CURR_MIN
[52:50]	3	don't care	VDD_W_CURR_MAX
[49:47]	3	"7"	C_SIZE_MULT
[46:42]	5	don't care	SECTOR_SIZE
[41:37]	5	don't care	ERASE_GRP_SIZE
[36:32]	5	don't care	WP_GRP_SIZE
[31:31]	1	don't care	WP_GRP_ENABLE
[30:29]	2	don't care	DEFAULT_ECC
[28:26]	3	don't care	R2W_FACTOR

CSD-slice	Width	Value	Field
[25:22]	4	don't care	WRITE_BLK_LEN
[21:21]	1	don't care	WRITE_BLK_PARTIAL
[20:16]	5	don't care	-
[15:15]	1	see table 12	FILE_FORMAT_GRP
[14:14]	1	don't care	COPY
[13:13]	1	1	PERM_WRITE_PROTECT
[12:12]	1	1	TMP_WRITE_PROTECT
[11:10]	2	see table 12	FILE_FORMAT
[9:8]	2	0	ECC
[7:1]	7	CRC value	CRC
[0:0]	1	1	-

All CSD fields are read only for the user. All don't care tagged field are zero. The following section describles the CSD fields and their values for the MX53L1281:

CSD_STRUCTURE
The CSD version of the MX53L1281 is related to the CSD version 1.1 as defined in "MultiMediaCard system specification. The parameter CSD_STRUCTURE is permanently assigned to the value 1.

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

The MultiMediaCard protocol version of the MX53L1281 is related to the"MultiMediaCard system specification, Version 2.2". The parameter SPEC_VERS is permanently assigned to the value 2.

TAAC
Defines the asynchronous data access time:

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

Table 5: TAAC access time definition

The coded TAAC value is 0x08. For more details see "Chapter 10.2.4: Operating characteristics".

NSAC
Defines the worst case for the synchronous data access time. N_{AC} is defined as 100*NSAC clock cycles, where NSAC represents a binary value. Max. value for the data access time N_{AC} is 25.6k clock cycles.

The total access time is the sum of both TAAC and N AC * clock period. The value of NSAC for the MX53L1281 is 0x03 (300 cycles). For more details see "Chapter 10.2.4: Operating characteristics".

TRAN_SPEED

The following table defines the maximum data transfer rate TRAN_SPEED:

TRAN_SPEED bit	Description
2:0	transfer rate unit 0=100kbit/s, 1=1Mbit/s, 2=10Mbit/s, 3=100Mbit/s, 4... 7=reserved
6:3	time value 0x0=reserved, 0x1=1.0, 0x2=1.2, 0x3=1.3, 0x4=1.5, 0x5=2.0, 0x6=2.5, 0x7=3.0, 0x8=3.5, 0x9=4.0, 0xA=4.5, 0xB=5.0, 0xC=5.5, 0xD=6.0, 0xE=7.0, 0xF=8.0
7	reserved="0"

Table 6: Maximum data transfer rate definition

The MX53L1281 supports a transfer rate between 0 and 20 Mbit/s.The parameter TRAN_SPEED is 0x2A.

CCC

The MultiMediaCard command set is divided into subsets (command classes). The card command class register CCC defines which command classes are supported by this card. A set CCC bit means that the corresponding command class is supported. For command class definition refer to Table 14.

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

Table 7: Supported card command classes

The MX53L1281 supports the command classes 0, 1 and 2.The parameter CCC is permanently assigned to the value 0x007.

READ_BLK_LEN

The data block length can be computed as $2^{READ\_BLK\_LEN}$. The block length might therefore be in the range 1, 2,4...2048 bytes.

READ_BLK_LEN	Block length	Remark
0	2^0 = 1 byte	
1	2^1 = 2 byte	
.....		
11	2^{11} = 2048 byte	
12-15	reserved	

Table 8: Data block length coding

READ_BLK_PARTIAL

READ_BLK_PARTIAL defines whether partial block sizes can be used in block read.
READ_BLK_PARTIAL=0 means that only the READ_BLK_LEN block sizes can be used for block oriented data transfers.

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

The MX53L1281 supports partial block read. The parameter READ_BLK_PARTIAL is permanently assigned to the value '1'.

READ_BLK_MISALIGN

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

READ_BLK_MISALIGN=0 signals that crossing physical block boundaries is not allowed.
READ_BLK_MISALIGN=1 signals that crossing physical block boundaries is allowed.

The MX53L1281 supports read block operations with boundary crossing. The parameter READ_BLK_MISALIGN is permanently assigned to the value "1".

DSR_IMP

Defines if the configurable driver stage option is integrated on the card or not. If implemented a driver stage register (DSR) must be implemented also.

DSR_IMP	DSR type
0	no DSR implemented
1	DSR implemented

Table 9: DSR implementation

The MX53L1281 output drivers are not configurable. The parameter DSR_IMP is permanently assigned to the value"0".

C_SIZE, C_SIZE_MULT

This parameter is used to compute the card capacity. The memory capacity of the card is computed from the entries C_SIZE, C_SIZE_MULT and READ_BLK_LEN as follows:

BLOCKLEN = $2^{READBLKLEN}$ = 2K Byte (READBLKLEN < 12)
MULT = $2^{CSIZEMULT+2}$ = 4 (CSIZEMULT< 8)
BLOCKNR = (CSIZE+1) * MULT = 8K Byte
SIZE = BLOCKNR * BLOCKLEN = 16M Byte

VDD_R_CURR_MIN

The maximum supply current at the minimal supply voltage V_{DD} (2.7 V):

VDD_R_CURR_MIN[2:0]	0=0.5 mA; 1=1 mA; 2=5 mA; 3=10 mA; 4=25 mA; 5=35 mA; 6=60 mA; 7=100 mA

Table 10: Supply current consumption @ V_{DD} =2.7 V

The parameter **VDD_R_CURR_MIN** is permanently assigned to the value 4 (25 mA).

VDD_R_CURR_MAX

The maximum supply current at the maximum supply voltage V DD (3.6V):

VDD_R_CURR_MAX[2:0]	0=1 mA; 1=5 mA; 2=10 mA; 3=25 mA; 4=35 mA; 5=45 mA; 6=80 mA; 7=200 mA

Table 11: Supply current consumption @ V^{DD} =3.6V

The parameter **VDD_R_CURR_MAX** is permanently assigned to the value 4 (35 mA).

FILE_FORMAT_GRP

Indicates the selected group of file formats. This field is read-only for ROM. The usage of this field is shown in Table 12 (see FILE_FORMAT).

PERM_WRITE_PROTECT

Permanently protects the whole card content against overwriting or erasing (all write and erase commands for this card are permanently disabled). This parameter has permanently the value "1".

TMP_WRITE_PROTECT

Temporarily protects the whole card content from being overwritten or erased (all write and erase commands for this card are permanently disabled). This parameter has always the value "1".

FILE_FORMAT

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

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

Table 12:File Format

ECC

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

ECC	ECC type	Maximum number of correctable bits
0	none (default)	none
1	BCH (542,512)	3
2-15	reserved	-

Table 13: ECC type

No external error correction is needed for the MX53L1281. The parameter **ECC** is permanently assigned to the value 0.

CRC

The CRC register carries the check sum for the CSD content. The check sum is computed by the following formulas:

Generator polynomial:

$G(x) = x^7 + x^3 + 1$

$M(x) = CSD[127] * x^{119} + ... + CSD[8] * x^0$

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

6 Communication

All communication between host and cards is controlled by the host (master). The host sends com-mands and, depending on the command, receives a corresponding response from the selected card. In this chapter the commands to control the MX53L1281, the card responses and the contents of a status and error field, included in the responses, are defined.

6.1 Commands

The command set of the MultiMediaCard system is divided into classes corresponding to the type of card (see also [1]). The MX53L1281 supports the following command classes:

Card Command Class (CCC)	Class description	Supported commands														
		0	1	2	3	4	7	9	10	11	12	13	15	16	17	18
class 0	basic	+	+	+	+	+	+	+	+		+	+	+			
class 1	sequential read									+						
class 2	block read													+	+	+

Table 14: MX53L1281 command classes

Class 0 is mandatory and supported by all cards. It represents the card identification and initialization commands, which are intended to handle different cards and card types on the same bus lines. The Card Command Class (CCC) is coded in the card specific data register of each card, so that the host knows how to access the card.

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

- broadcast commands (bc)
  sent on CMD line, no response
- broadcast commands with response (bcr)
  sent on CMD line, response (all cards simultaneously) on CMD line
- addressed (point-to-point) commands (ac)
  sent on CMD line, response on CMD line
- addressed (point-to-point) data transfer commands (adtc)
  sent on CMD line, response on CMD line, data transfer on DAT line

The command transmission always starts with the MSB. Each command starts with a start bit and ends with an CRC command protection field followed by a end bit. The length of each command frame is fixed to 48 bits (2.4 ms @ 20 MHz):

0	1	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	host	command	argument	CRC [1]	end bit

[1] Cyclic Redundancy Check

The start bit is always "0" in command frames (sent from host to MultiMediaCard). The host bit is always "1" for commands. The command field contains the binary coded command number. The argument depends on the command (see Table 15 and Table 16). The CRC field is defined in "Chapter 7: Error handling".

The MX53L1281 supports the following MultiMediaCard command:

CMD INDEX	Type	Argument	Resp	Abbreviation	Command Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	resets all cards to Idle State
CMD1	bcr	argument ignored	R3	SEND_OP_COND	checks for cards not supporting the full range of 2.0 to 3.6V. After receiving CMD1 the card sends an R3 response (see "Chapter 6.5: Responses").
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	asks all cards in ready state to send their CID[1] numbers on CMD-line
CMD3	ac	[31:16] RCA [15:0] stuff bits	R1	SET_RELATIVE_ ADDR	assigns relative address to the card in identification state.
CMD4	bc	[31:16] DSR [15:0] stuff bits	-	SET_DSR	programs the DSR of all cards in stand-by state.
CMD7	ac	[31:16] RCA [15:0] stuff bits	R1 (only the selected card)	SELECT_ DESELECT_ CARD	command toggles a card between the standby and transfer states or between the programming and disconnect state. In both cases the card is selected by its own relative address while deselecting the prior selected card. Address 0 deselects all.
CMD9	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CSD	asks the addressed card to send its card-specific data (CSD)[2] on CMD-line.
CMD10	ac	[31:16] RCA [15:0] stuff bits	R2	SEND_CID	asks the addressed card to send its card identification (CID) on CMD-line.
CMD11	adtc	[31:0] data	R1	READ_DAT_ UNTIL_STOP	reads data stream from the card in sending-data state, starting at the supplied address, until STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1	STOP_ TRANSMISSION	forces the card to stop transmission
CMD13	ac	[31:16] RCA [15:0] stuff bits	R1	SEND_STATUS	Asks the addressed card to send its status register.
CMD15	ac	[31:16] RCA [15:0] stuff bits	-	GO_INACTIVE_ STATE	Sets the card to inactive state in order to protect the card stack against communications break-downs.

Table 15: Basic commands for read only devices (class 0 and class 1)

1.CID register consists of 128 bits (starting with MSB, it is preceded by an additional start bit, ends with an end bit)
2.CSD register consists of 128 bits (starting with MSB, it is preceded by an additional start bit, ends with an end bit)
3.The addressing capability @ 8 bit address resolution is 2^{32} = 4 Gbyte

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

Table 16: Block oriented read commands (class 2)

1.The default block length is as specified in the CSD.
2.The data transferred must not cross a physical block boundary unless RD_BLK_MISALIGN is set in the CSD.

6.2 Card identification mode

All the data communication in the card identification mode uses only the command line (CMD).

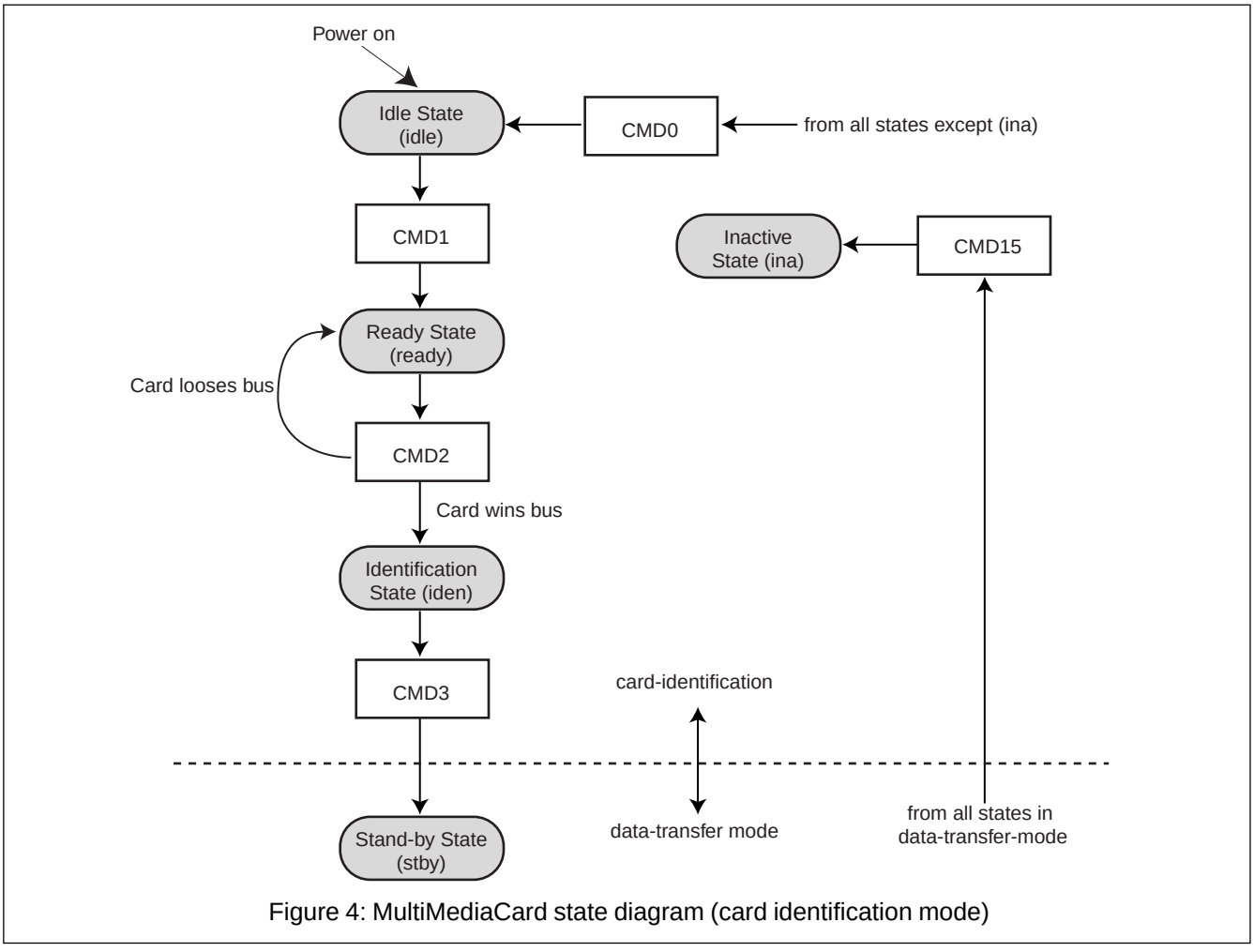


Figure 4: MultiMediaCard state diagram (card identification mode)

The host starts the card identification process in open drain mode with the identification clock rate f_{OD} (generated by a push pull driver stage). The open drain driver stages on the CMD line allow the parallel card operation during card identification.

After the bus is activated the host will request the cards to send their valid operation conditions with the command SEND_OP_COND(CMD1). Since the bus is in open drain mode, as long as there is more than one card with operating conditions restrictions, the host gets in the response to the CMD1 a "Wired or" operation condition restrictions of those cards. The host then must pick a common denominator for operation and notify the application that cards with out of range parameters (from the host perspective) are connected to the bus. Incompatible cards go into Inactive State. After an operating mode is established, the host asks all cards for their unique card identification (CID) number with the broadcast command ALL_SEND_CID (CMD2). All not already identified cards (i.e. those which are in Ready State) simultaneously start sending their CID numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle (cards stay in the Ready State). There should be only one card which successfully sends its full CID-number to the host. This card then goes into the Identification State. The host assigns to this card (using CMD3, SET_RELATIVE_ADDR) a relative card address (RCA, shorter than CID), which will be used to address the card in future communication (faster than with the CID). Once the RCA is received the card transfers to the Standby State and does not react to further identification cycles. The card also switches the output drivers from the open-drain to the push-pull mode in this state.

The host repeats the identification process as long as it receives a response (CID) to its identifica-tion command (CMD2). When no card responds to this command, all cards have been identified. The time-out condition to recognize this, is waiting for the start bit for more than 5 clock periods after sending CMD2.

6.3 Operating voltage range validation

The MultiMediaCard standards operating range validation is intended to support reduced voltage range MultiMediaCards. The MX53L1281 supports the full range of 2.7 to 3.6V supply voltage. So the MX53L1281 sends a R3 response to CMD1 which contains an OCR value of 0x00FFC000 (see "Chapter 6.5: Responses").

6.4 Data transfer mode

When in Standby State, both CMD and DAT lines are in the push-pull mode. As long as the content of all CSD registers is not known, the $f_{PushPull}$ clock rate is equal to the slow $f_{OpenDrain}$ clock rate. SEND_CSD (CMD9) allows the host to get the Card Specific Data (CSD register), e.g. ECC type, block length, card storage capacity, maximum clock rate etc.

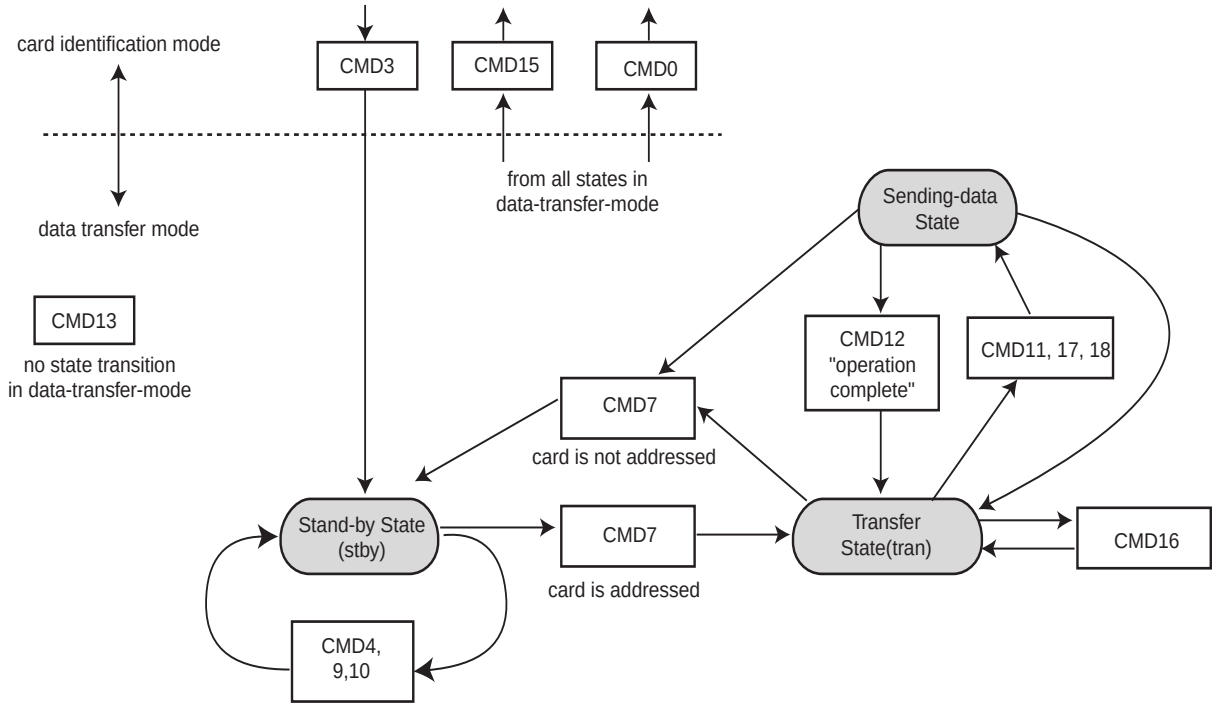


Figure 5: MX53L1281 state diagram (data transfer mode)

The command SELECT_DESELECT_CARD (CMD7) is used to select one card and place it in the Transfer State. If a previously selected card is in the Transfer State its connection with the host is released and it will move back to the Stand-by State. Only one card can be, at any time, in the Transfer State. A selected card is responding the CMD7, the deselected one does not respond to this command.

When CMD7 is sent including the reserved relative card address "0x0000", all cards transfer back to Stand-by State. This command is used to identify new cards without resetting other already acquired cards. Cards to which an RCA has already been assigned, do not respond to the identification command flow in this state.

All the data communication in the Data Transfer Mode is consequently a point-to point communication between the host and the selected card (using addressed commands). All addressed commands are acknowledged by a response on the CMD line.

All read commands (data is sent from the card via data lines) can be interrupted at any time, by another read or a stop command.

The DAT bus line is high when no data is transmitted. A transmitted data block consists of a start bit (LOW), followed by a continuous data stream. The data stream contains the net payload data (and error correction bits if an off-card ECC is used). The data stream ends with an end bit (HIGH). The data transmission is synchronous to the clock signal. The payload for block oriented data transfer is protected by a CRC check sum (see "Chapter 7:Error handling").

Stream read

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

If the end of the memory range is reached while sending data and no stop command has yet been sent by the host, the data transfer will continued. The data sent than is undefined. The host has to observe the boundaries of the memory range.

Block read

Block read is similar to stream read, except the basic unit of data transfer is a block whose maxi-mum size is defined in the CSD (READ_BLK_LEN). READ_BLK_PARTIAL is set, thus smaller blocks whose starting and ending address are wholly contained within one physical block (as defined by READ_BLK_LEN) may also be transmitted. Unlike stream read, a CRC is appended to the end of each block ensuring data transfer integrity. READ_SINGLE_BLOCK (CMD17) starts a block read and after a complete transfer the card goes back to Transfer State.

READ_MULTIPLE_BLOCK (CMD18) starts a transfer of several consecutive blocks. Blocks will be continuously transferred until a stop command is issued. Block misalignment is also allowed for the MX53L1281 .

State transition summary

The Table 17 defines the card state transitions as a function of received command.

	current state						
	idle	ready	ident	stby	tran	data	ina
CRC fail	- [1]	-	-	-	-	-	-
commands of not supported classes	-	-	-	-	-	-	-
class 0							
CMD0	idle	idle	idle	idle	idle	idle	-
CMD1	ready	-	-	-	-	-	-
CMD2,card wins bus	-	ident	-	-	-	-	
CMD2,card loses bus	-	ready	-	-	-	-	-
CMD3	-	-	stby	-	-	-	-
CMD4	-	-	-	stby	-	-	
CMD7,card is addressed	-	-	-	tran	-	-	
CMD7,card is not addressed	-	-	-	-	stby	stby	-
CMD9	-	-	-	stby	-	-	-
CMD10	-	-	-	stby	-	-	-
CMD12	-	-	-	-	-	tran	-
CMD13	-	-	-	stby	tran	data	-
CMD15	-	-	-	ina	ina	ina	-
class 1							
CMD11	-	-	-	-	data	-	-
class 2							
CMD16	-	-	-	-	tran	-	-
CMD17	-	-	-	-	data	-	-
CMD18	-	-	-	-	data	-	-

Table 17: Card state transition table

[1] Stay in the current state.

6.5 Responses

All responses are sent via command line (CMD), all data starts with the MSB.
Format R1 (response command): response length 48 bit.

0	0	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	card	command	status	CRC	end bit

The contents of the status field are described in "Chapter 6.6: Status"

Format R2 (CID, CSD register): response length 136 bits.

Note: Bit 127 down to bit 1 of CID and CSD are transferred, the reserved bit [0] is replaced by the end bit.

0	0	bit 5...bit 0	bit 127...bit 1	1
start bit	card	reserved	CID or CSD register including internal CRC	end bit

CID register is sent as a response to commands CMD2 and CMD10.
CSD register is sent as a response to the CMD9.

Format R3 (OCR): response length 48 bits.

0	0	bit 5...bit 0	bit 31...bit 0	bit 6...bit 0	1
start bit	card	reserved	OCR field	reserved	end bit

The OCR is sent as a response to the CMD1 to signalize the supported voltage range. The MX53L1281 supports the full range from 2.7 to 3.6 V. Respectively the value of all bits of the OCR field of the MX53L1281 are always set to high (0x00FFC000). The reserved bits are also high (0x3F and 0x7F). So the R3 frame of the MX53L1281 contains always the value 0x3F00FFC000FF.

6.6 Status

The response format R1 contains a 32-bit field with the name card status. This field is intended to transmit status information which is stored in a local status register of each card to the host. The fol-lowing table defines the status register structure.
The Type and Clear-Condition fields in the table are coded as follows:

- Type:
  - E-Error bit.
  - S-Status bit.
  - R-Detected and set for the actual command response.
  - X-Detected and set during command execution. The host must poll the card by sending status command in order to read these bits.
- Clear Condition:
  - A- According to the card state.
  - B- Always related to the previous command. Reception of a valid command will clear it (with a delay of one command).
  - C- Clear by read.

Bits	Identifier	Type	Value	Description	Clear Condition
31	OUT_OF_RANGE	ER '0		The command argument was out of the allowed range for this card.	C
30	Don't care		Permanently 0.		
29	BLOCK_LEN_ERROR	ER '0	"0"=no error "1"=error	The transferred block length is not allowed for this card or the number of bytes transferred does not match the block length.	C
28:26	Don't care		Permanently 0.		
25:24	reserved		Permanently 0.		
23	COM_CRC_ERROR	ER '0	"0"=no error "1"=error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	ER '0	"0"=no error	Command not legal for the current state.	B
21	Don't care		Permanently 0.		
20	Don't care		Permanently 0		
19	Don't care		Permanently 0.		
18	Don't care		Permanently 0		
17:13	Don't care		Permanently 0.	Current state of the card.	B
12:9	CURRENT_STATE	S X	0 = idle 1 = ready 2 = ident 3 = stby 4 = tran 5 = data 6-15 = reserved	Current state of the card.	B
8	Don't care		Permanently 0.		
7:0	reserved		Permanently 0.		

Table 18: Status

6.7 Command and response timings

All timing diagrams use the following schematics and abbreviations:

S	Start bit (= 0)
T	Transmitter bit (Host = 1, Card = 0)
P	One-cycle pull-up (= 1)
E	End bit (=1)
Z	high impedance state (-> = 1)
D	Data bits
*	repeater
CRC	Cyclic redundancy check bits (7 bits)
	Card active
	Host active

Table 19: Timing diagram symbols

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

For the timing of the MX53L1281 the following values are defined:

	Value [clock cycles]	Description
N_{CR}	5	Number of cycles between command and response
N_{ID}	5	Number of cycles between card identification or card operation conditions com-mand and the correspond-ing response.
N_{AC}	TAAC+NSAC	
N_{BAC}	8	Number of cycles between blocks in multiple block read
N_{RC}	≥ 8	
N_{CC}	≥ 8	Number of cycles between two commands, if no response will be sent after the first command (e.g.broadcast)

Table 20: Timing values

The host command and the card response are clocked out with the rising edge of the host clock.The delay between host command and card response is N_{CR} clock cycles.

The following timing diagram is relevant for host command CMD3:

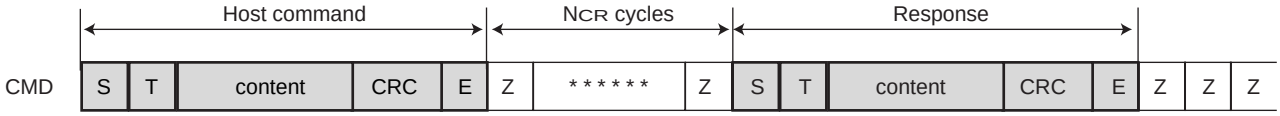


Figure 6: Command response timing (identification mode)

There is a two Z bit period followed by P bits pushed up by the responding card. The following tim-ing diagram is relevant for all host commands followed by a response, except CMD1, CMD2 and CMD3:

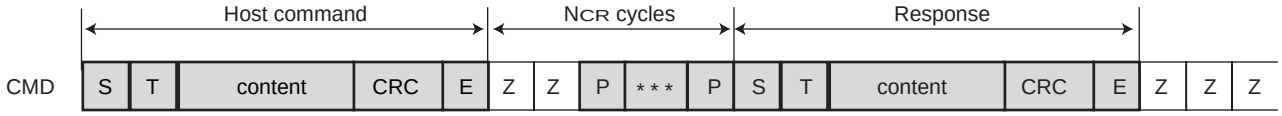


Figure 7: Command response timing (data transfer mode)

Card identification and card operation conditions timing

The card identification (CMD2) and card operation conditions (CMD1) timing are processed in the open-drain mode. The card response to the host command starts after exactly NID clock cycles.

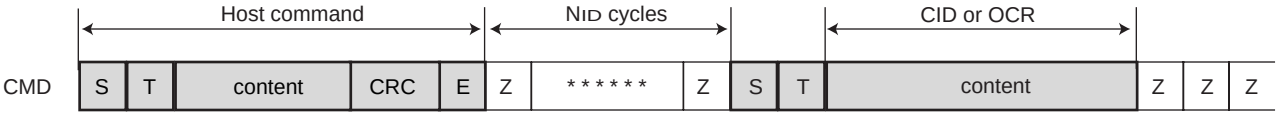


Figure 8: Identification timing (card identification mode)

Last card response - next host command timing

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

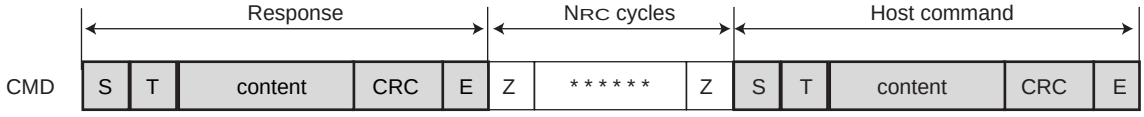


Figure 9: Timing response end to next CMD start (data transfer mode)

Last host command - next host command timing diagram

After the last command, which does not force a response, has been sent, the host can continue sending the next command after at least N_{CC} clock periods.

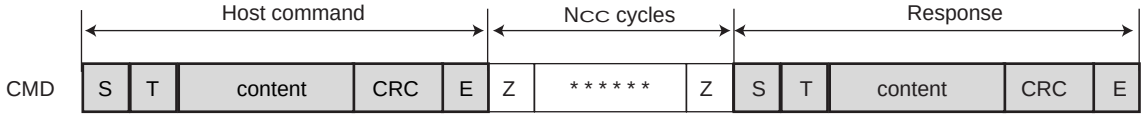


Figure 10: Timing CMD_n end to CMD_{n+1} start (all modes)

In the case the CMD_n command was a last identification command (no more response sent by a card), then the next CMD_{n+1} command is allowed to follow after at least N_{CC} +136 (the length of the R2 response) clock periods.

Data access timing

Data transmission starts with the access time delay t_{AC} (which corresponds to N_{AC}), beginning from the end bit of the data address command. The data transfer stops automatically in case of a data block transfer or by a transfer stop command.

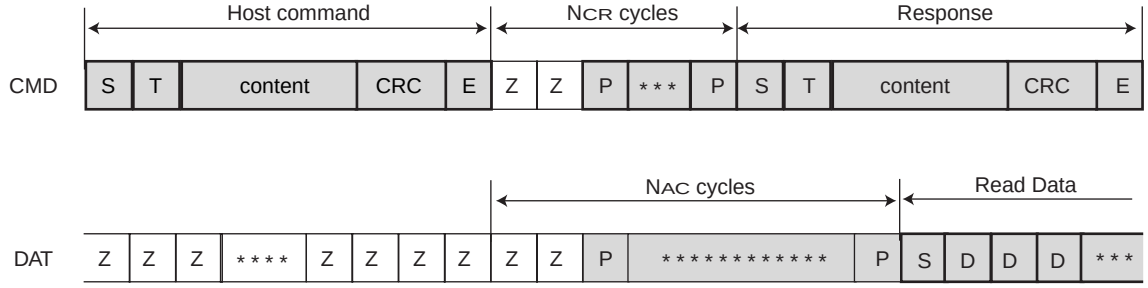


Figure 11: Data read timing (data transfer mode)

Data transfer stop command timing

The card data transmission can be stopped using the stop command. The data transmission stops immediately with the end bit of the stop command.

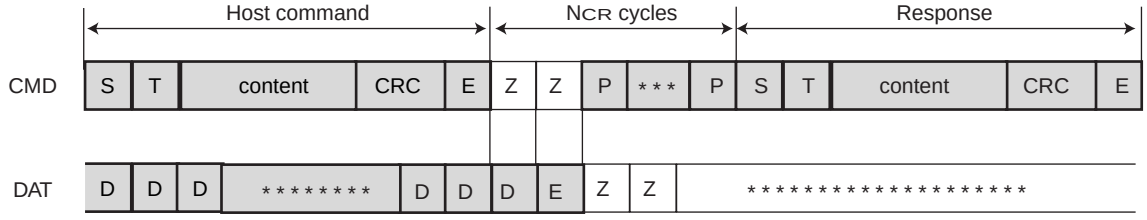


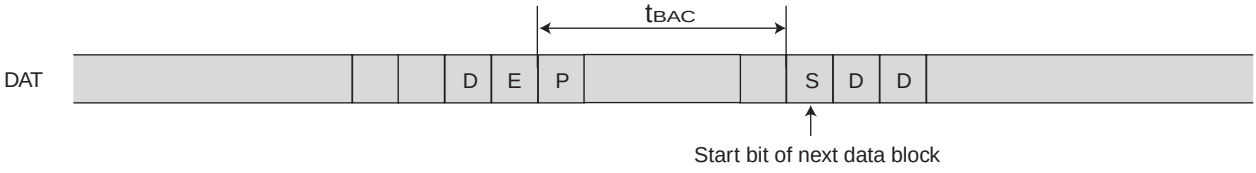
Figure 12: Timing of stop command (CMD12, data transfer mode)

Data transfer stop

The read command (CMD11, 17, 18) is ignored, while data transmission is active. Only STOP command(CMD12) or DESELECT_CARD (CMD7) is able to stop the data transmission task.

Next data block transfer timing

In multiple block read mode, the next data block transmission starts with the delay time t_{BAC}(N_{BAC} Clock cycles), beginning from the end bit of the previous data block.



6.8 Clock Control

The bus frequency can be changed at any time (under the restrictions of masimum data trandfer frequency, defined by the cards, and the identification frequency defined by the specification document).

It is an obvious requirement that the clock must be running for the card to output data or response tokens. After the last MultiMediaCard bus transaction, the host is required, to provide 8(eight) clock cycles for the card to complete the operation before shutting down the clock. Following is a list of the various bus transactions:

- A command with no response. 8 clocks after the host command end bit.
- A command with response. 8 clocks after the card response end it.
- A read data transaction. 8 clocks after the end bit of the last data block.

6.9 Reset

GO_IDLE_STATE(CMD0) is the software reset command, which sets the MX53L1281 into the Idle State independently of the current state. In the Inactive State the R008 is not affected by this command.

After power-on the MX53L1281 is always in the Idle State. After power-on or command the card will be initialized with a default relative card address ("0x0001"). The host runs at the identification clock rate fOD generated by a push-pull driver stage(see also "Chapter 9.2 Power On" for more details).

7 SPI Mode

7.1 Introduction

The SPI mode consists of a secondary, optional communication protocol which is offered by Flashbased MultiMediaCards. This mode is a subset of the MultiMediaCard 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 MultiMediaCard SPI implementation uses a subset of the MultiMediaCard protocol and command set. It is intended to be used by systems which require a small number of cards (typically one) and have lower data transfer rates (compared to MultiMediaCard protocol based systems). From the application point of view, the advantage of the SPI mode is the capability of using an off-the-shelf host, hence reducing the design-in effort to minimum. The disadvantage is the loss of performance of the SPI mode versus MultiMediaCard mode (lower data transfer rate, fewer cards, hardware CS per card, etc.).

7.2 SPI Interface Concept

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

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

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

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

7.3 SPI Bus Topology

The card identification and addressing methods are replaced by a hardware Chip Select (CS) signal. There are no broadcast commands. For every command, a card (slave) is selected by asserting (active low) the CS signal (see Figure 43).

The CS signal must be continuously active for the duration of the SPI transaction (command, response and data). The only exception occurs during card programming, when the host can deassert the CS signal without affecting the programming process.

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

The SPI interface uses the same 7 signals of the standard MultiMediaCard bus (see Table 21).

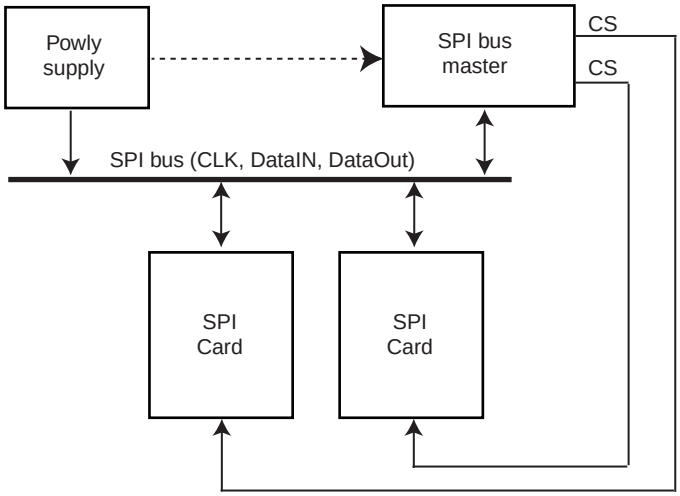


Figure 13: MultiMediaCard bus system

Pin#	MultiMediaCard Mode			SPI Mode		
	Name	Type 1	Description	Name	Type	Description
1	RSV	NC	Reserved for future use	CS	I	Chip Select (neg true)
2	CMD	I/O/PP/OD	Command/Response	DI	I/PP	Data In
3	V SS1	S	Supply voltage ground	VSS	S	Supply voltage ground
4	V DD	S	Supply voltage	VDD	S	Supply voltage
5	CLK	I	Clock	SCLK	I	Clock
6	V SS2	S	Supply voltage ground	VSS2	S	Supply voltage ground
7	DAT	I/O/PP	Data	DO	O/PP	Data Out

Table 21: SPI interface pin configuration

1) S: power supply; I: input; O: output; PP: push-pull; OD: open-drain; NC: Not connected (or logical high)

7.4 MultiMediaCard Registers in SPI Mode

The register usage in SPI mode is summarized in Table 22.
Most of them are inaccessible.

Name	Available in SPI mode	Width [Bytes]	Description
CID	Yes	16	Card identification data (serial number, manufacturer ID, etc.)
RCA	No		
DSR	No		
CSD	Yes	16	Card-specific data, information about the card operation conditions.
OCR	Yes	32	Operation condition register.

Table 22: MultiMediaCard registers in SPI mode

7.5 SPI Bus Protocol

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

Similar to the MultiMediaCard protocol, the SPI messages consist of command, response and data-block tokens (see Chapter 3 for a detailed description). 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 MultiMediaCard 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 MultiMediaCard mode.

Only single block read operations are supported in SPI mode. A data block may be as big as one card sector and as small as a single byte. Partial block read operations are enabled by card options specified in the CSD register.

7.5.1 Mode Selection

The MultiMediaCard wakes up in the MultiMediaCard mode. It will enter SPI mode if the CS signal is asserted (negative) during the reception of the reset command (CMD0). If the card recognizes that the MultiMediaCard mode is required, it will not respond to the command and remain in the MultiMe-diaCard 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 MultiMediaCard mode is by entering the power cycle. In SPI mode, the MultiMediaCard protocol state machine is not observed. All the MultiMediaCard commands sup-ported in SPI mode are always available.

7.5.2 Bus Transfer Protection

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

In the non-protected mode, the CRC bits of the command, response and data tokens are still required in the tokens. However, they are defined as "don't 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 MultiMediaCard 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).

7.5.3 Data Read

The SPI mode supports single block read operations only (CMD17 in the MultiMediaCard 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 14).

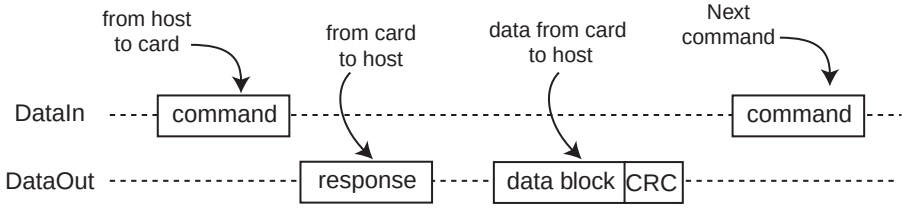


Figure 14:Read Operation

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.

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 15 shows a data read operation which terminated with an error token rather than a data block.

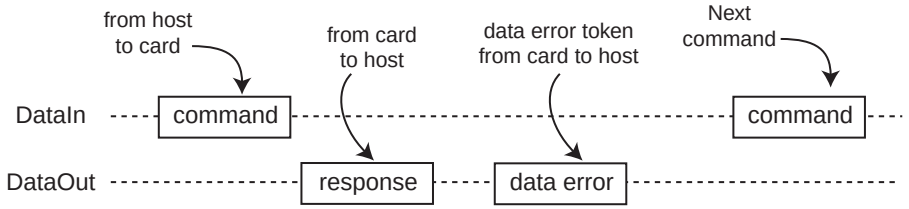


Figure 15:Read Operation- Data Error

7.5.4 Read CID/CSD Registers

Unlike the MultiMediaCard 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 14) 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 card TAAC since this value is stored in the CSD. Therefore, the standard response time-out value (N_{CR}) is used for read latency of the CSD register. And the time out value for CID command also is set to the standard response time (N_{CR}, minimux N_{AC}).

7.5.5 Reset Sequence

The MultiMediaCard 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) and CMD58 (READ_OCR).

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

In SPI mode, as opposed to MultiMediaCard mode, CMD1 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. The host must poll the card (by repeatedly sending CMD1) until the "in-idle-state " bit in the card response indicates (by being set to 0) that the card has completed its initialization processes and is ready for the next command.

7.5.6 Error Conditions

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

7.5.7 Memory Array Partitioning

Same as for MultiMediaCard mode.

7.5.8 Application Specific commands

Identical to MultiMediaCard mode with the exception of the APP_CMD status bit, which is not available in SPI.

7.6 SPI Mode Transaction Packets

7.6.1 Command Tokens

• Command Format

All the MultiMedia 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 8.1). The commands and arguments are listed in Table 24.

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

• Command Classes

As in MultiMediaCard mode, the SPI commands are divided into several classes (See Table 23). Each class supports a set of card functions. A MultiMediaCard 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 MultiMediaCard and the SPI communication mode.

Card CMD	Class Description	Supported commands																						
Class(CCC)		0	1	9	10	13	16	17	24	27	28	29	30	32	33	34	35	36	37	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																								

Table 23: 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 7.6.2. Table 24 lists all MultiMediaCard commands. A "yes" in the SPI mode column indicates that the command is supported in SPI mode. With these restrictions, the command class description in the CSD is still valid. If a command does not require an argument, the value of this field should be set to zero. The reserved commands are also reserved in MultiMediaCard mode.

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 MultiMedia 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				
CMD8	reserved				
CMD9	Yes	None	R1	SEND_CSD	asks the selected card to send its card-specific data (CSD)
CMD10	Yes	None	R1	SEND_CID	asks the selected card to send its card identification (CID)
CMD11	No				
CMD12	No				
CMD13	Yes	None	R2	SEND_STATUS	asks the selected card to send its status register
CMD14	reserved				
CMD15	No				
CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	selects a block length (in bytes) for all following block commands (read and write)[1]
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_ BLOCK	reads a block of the size selected by the SET_BLOCKLEN command[2]
CMD18	No				
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

Table 24: 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.

• Card State Transition Table

Although the MultiMedia protocol state machine is not abserved, the command sequence sent from host must follow the state transition table as below.

State / Command	Idle	ready	data
CMD0	idle	idle	idle
CMD1	ready	ready	-
CMD9	-	data	-
CMD10	-	data	-
CMD13	-	ready	-
CMD16	-	ready	-
CMD17	-	data	-
CMD58	idle	ready	-
CMD59	-	ready	-

Table 25:Card State Transition Table

When the card finishes the read operation (CMD9, CMD10, CMD17) the state is changed from data state to ready state, automatically.

7.6.2 Responses

There are several types of response tokens. As in the MultiMediaCard 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 indica-tions, an error being signaled by a '1'. The structure of the R1 format is given in Figure 48. The meaning of the flags is defined as follows:

- **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 occurred in the sequence of erase commands.
- **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.

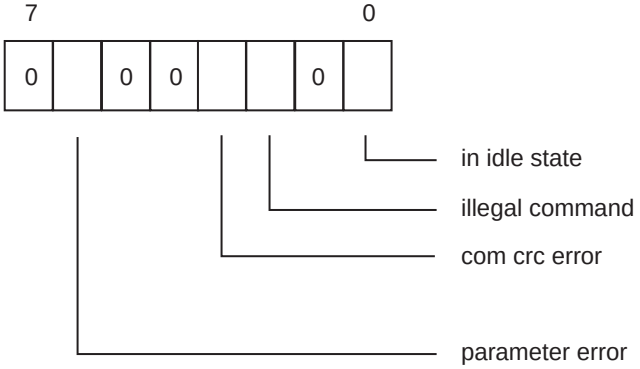


Figure 16: R1 Response Format

• 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 17.

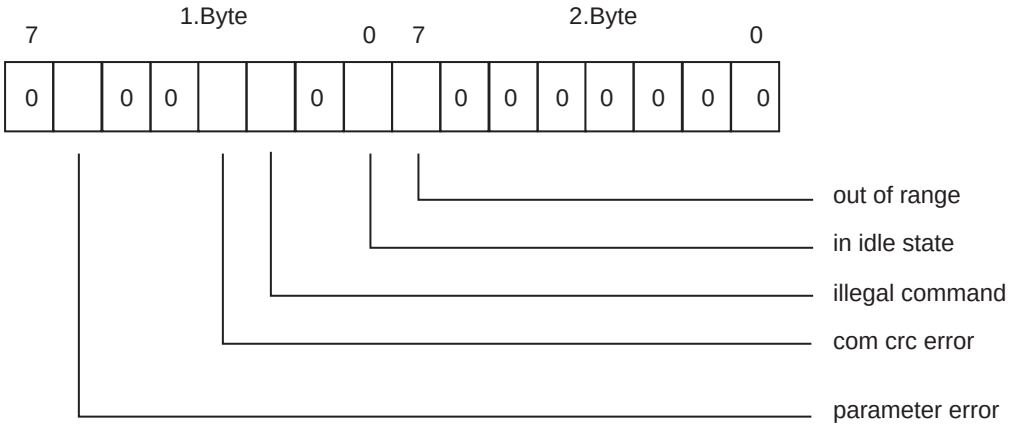


Figure 17: R2 response format

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

- **out of range | csd_overwrite:** This status bit has two functions. It is set if the command argument was out of its valid range or if the host is trying to change the ROM section or reverse the copy bit (set as original) or permanent WP bit (un-protect) of the CSD register.
- **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. It is set when the host attempts to erase a write-protected sector or if a sequence or password error occurred during a 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 18). The structure of the first (MSB) byte is identical to response type R1. The other four bytes contain the OCR register.

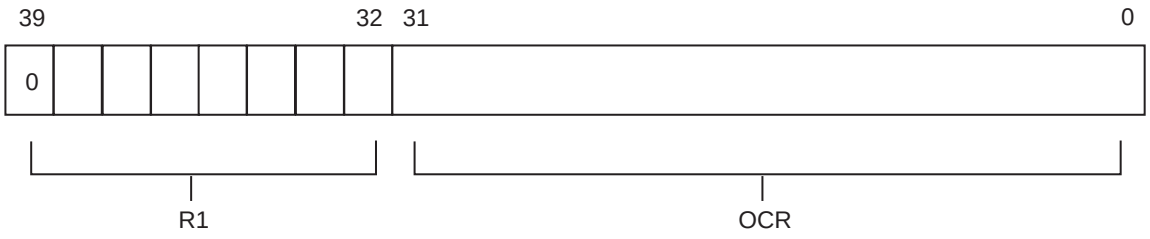
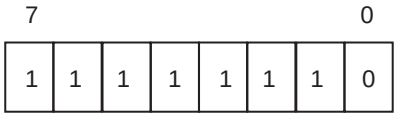


Figure 18:R3 Response Format

7.6.3 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:
• First byte: Start Byte



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

7.6.4 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:

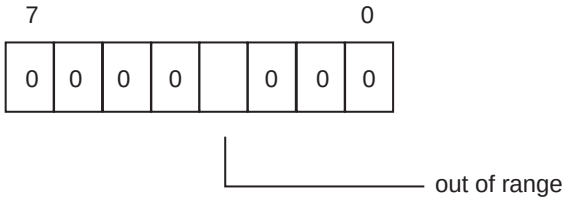


Figure 19: Data Error Token

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

7.6.5 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 multiple response types - e.g Card ECC failed)

As in the MultiMediaCard 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
Parameter command.	R1 R2	E R X	"0"=no error "1"=error	An error in the parameters of the command	C
Com CRC	R1 R2	E R X	"0"=no error "1"=error	The CRC check of the previous command failed	C
Illegal command	R1 R2	E R X	"0"=no error "1"=error	Command not legal for the card state	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 26: 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

7.7 Card Registers

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

7.8 SPI Bus Timing Diagrams

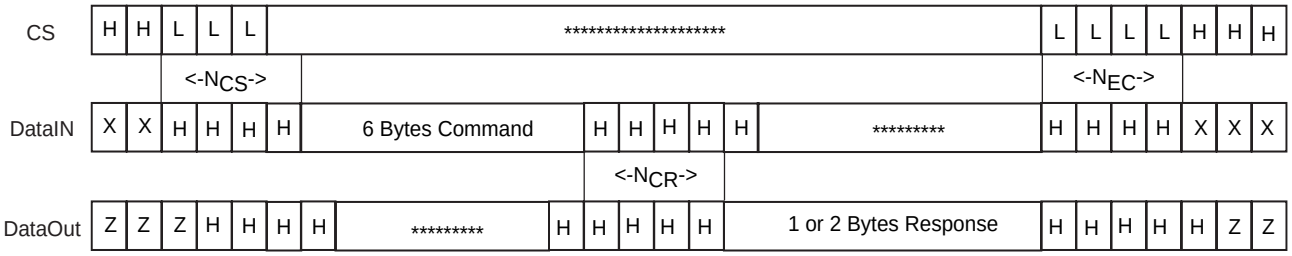
All timing diagrams use the following schematics and abbreviations:

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

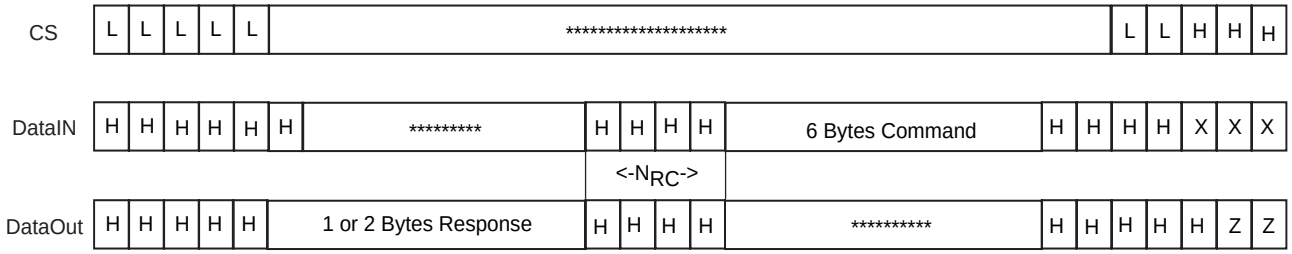
All timing values are defined in Table 27. The host must keep the clock running for at least N_{CR} clock cycles after receiving the card response. This restriction applies to both command and data response tokens.

7.8.1 Command / Response

• Host Command to Card Response - Card is ready
The following timing diagram describes the basic command response (no data) SPI transaction.

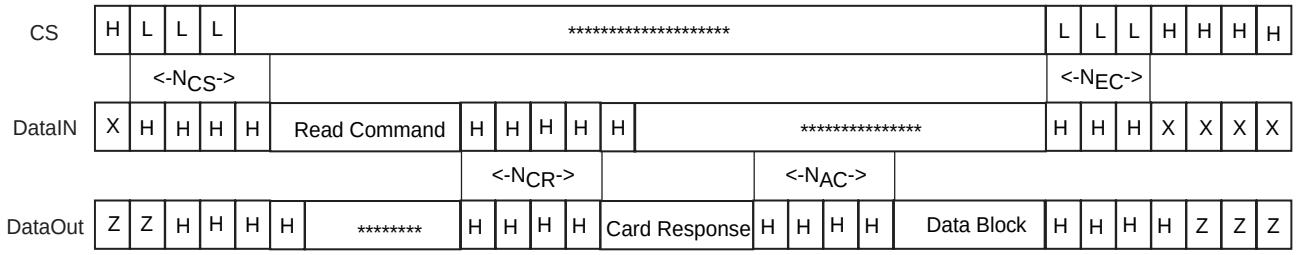


• Card Response to Host Command



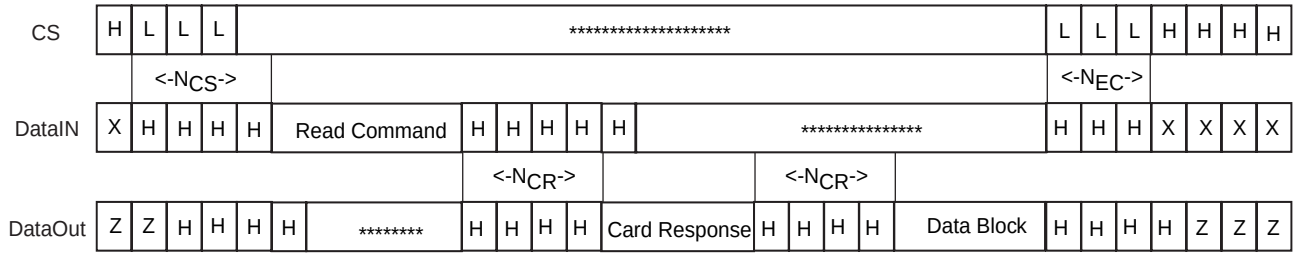
7.8.2 Data read

• The following timing diagram describes all read operations with the exception of SEND_CSD command.



• Reading the CSD,CID register

The following timing diagram describes the SEND_CSD command bus transaction. The timeout val-ues for the response and the data block are N_{CR} (Since the N_{AC} is still unknown).



7.8.3 Timing Values

	MIN	MAX	UNIT
N_{CS}	0	-	8 clock cycles
N_{CR}	1	1	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 27: Timing values

7.9 SPI Electrical Interface

Identical to MultiMediaCard mode with the exception of the programmable card output drivers option, which is not supported in SPI mode.

7.10 SPI Bus Operating Conditions

Identical to MultiMediaCard mode.

7.11 Bus Timing

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

8 Error handling

The MX53L1281 is defined as an error free device. To protect the data against errors generated during the transport over the MultiMediaCard bus dynamically, an additional feature is implemented: The cyclic redundancy check (CRC).

8.1 CRC

Following the MultiMediaCard standard, the MX53L1281 uses two different CRC codes to protect the data and the command/response transfer between card and host. The CRC is intended only to detect transfer errors and not to correct them "on the fly". If a CRC error is detected the host has to react. This is normally done by repeating the last command.

The first CRC code is intended to protect the command and response frames. They are also used to synchronize the data stream. This CRC is generated with and checked against the following polyno-mial:

CRC polynomial: $G(x) = x^7 + x^3 + 1$
$M(x) =$ (start bit) $* x^{39} +...+$ (last bit) $* x^0$
CRC[6...0] = Remainder $[(M(x) * x^7) / G(x)]$

One CRC is checked in the MX53L1281 for every command. For each response a CRC is generated in the MX53L1281. On CRC failure the command will be ignored and a response is sent to initiate a repeti-tion of the command by the host. Each data block read from the MX53L1281 will be succeeded by redun-dancy bits generated with the second CRC. The code is usable for payload lengths of up to 2048 Bytes:

CRC polynomial: $G(x) = x^{16} + x^{12} + x^5 + 1$,
$M(x) =$ (start bit) $* x^n + x^{n-1} +...+$ (last bit) $* x^0$, with $n < 2048*8$
CRC[15...0] = Remainder $[(M(x) * x^{16}) / G(x)]$

9 Power supply

9.1 Power supply decoupling

The VSS1, VSS2 and VDD lines supply the card with operating voltage. A decoupling capacitor (C) for current peak buffering has to be foreseen. This capacitor is placed on the bus side corresponding to Figure 20.

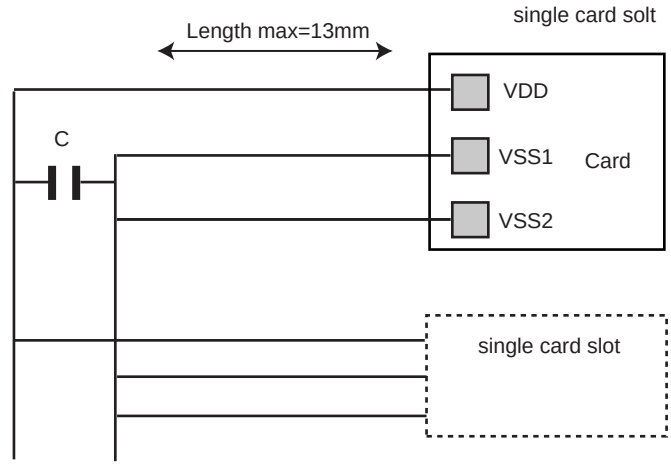


Figure 20: Power supply decoupling

9.2 Power on

Each card has its own power on detection circuitry which puts the card into a defined state after the power-on. No explicit reset signal is necessary. The cards can also be reset by a special software command: GO_IDLE_STATE (CMD0). In case of emergency the host may also reset the cards by switching the power supply off and on again.

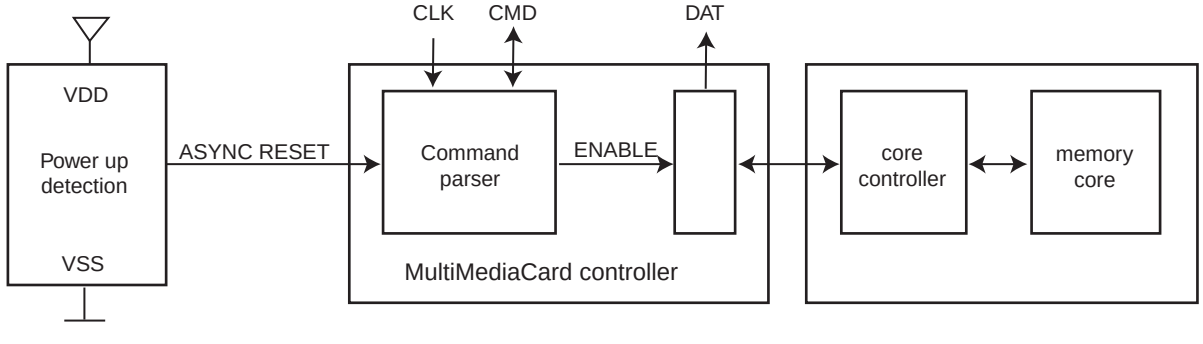


Figure 21: Power on detection

A power-on-reset is generated on chip as long as V_{DD} is below a certain value. After the power on reset the command parser of the MX53L1281 works properly, but the access to the memory core is not guaranteed as long as V_{DD} min is not reached. Therefore in the power up phase (or when the MX53L1281 is inserted during power up) the host has to wait after sending the SEND_OP_COND command (CMD1) for the identification delay. After that the ALL_SEND_CID command(CMD2) can be interpreted by the card:

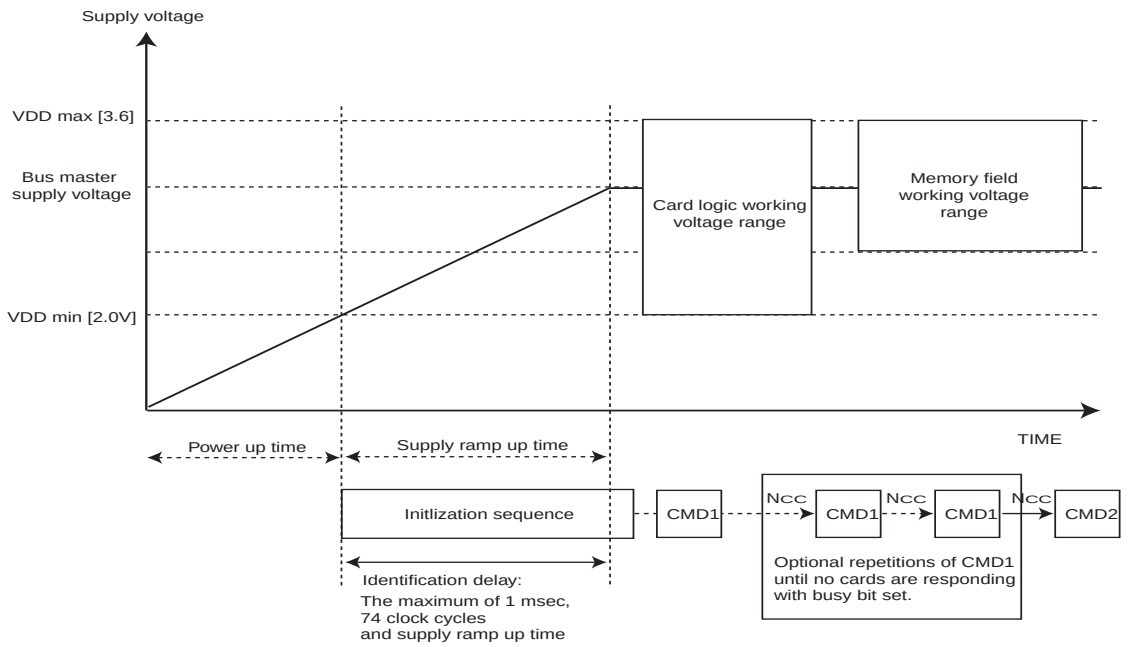


Figure 22: Power up diagram

For the MX53L1281 the following minimum initialization and identification delays are defined:

Description	Symbol	Minimum Value	
Initialization delay	t init	1 ms	@ f clk >64kHz
		64 cycles	@ f clk <64kHz
Identification delay	t ident	1 ms	

Table 28: Initialization and identification delays

The initialization delay is relevant only after the system power up (>1 ms, at least 64 clock cycles).
The identification delay is relevant for system power up and card hot insertion (> 1 ms). The MX53L1281 ignores all commands until the sequence CMD1, CMD2 is received and the RCA of the card is ini-tialized. The initialization delay guarantees enough time for V_{DD} to reach the minimum operating voltage on the MultiMediaCard bus. The identification delay guarantees enough time for V_{DD} to reach the minimum operating voltage internally in the MultiMediaCard.

9.3 Power consumption

The MX53L1281 power consumption depends on three parameters:
- The operating frequency
- The operating voltage
- The card state

In the following table the supply current and the power consumption of one MX53L1281 for typical operating conditions are listed. These parameters are typical values to give system designers some hints, all guaranteed parameters are listed in "Chapter 10.2: Electrical characteristics":

Description	Frequency	Card state	2.7V	3.6V
Clock off	0 Hz [1]	stby	< 100uA	< 200uA
Low speed	100kHz	data, tran		< 5mA
Initialzation	400kHz[2]	idle, ready, ident, ina		< 5mA
High speed	20MHz	stby		
		data, tran	< 25mA	< 35mA

Table 29: Typical MX53L1281 supply current values

[1] Host has stopped generation of clock pulses.
[2] In the initialization phase. No access to the memory core.

9.4 Short cut protection

The MX53L1281 can be inserted/removed into/from the bus without damage. If one of the supply pins (V_{DD} , V_{SS} or V_{PP}) is not connected properly, then the current is drawn through a data line to supply the card. Naturally the card can not operate properly under these conditions.

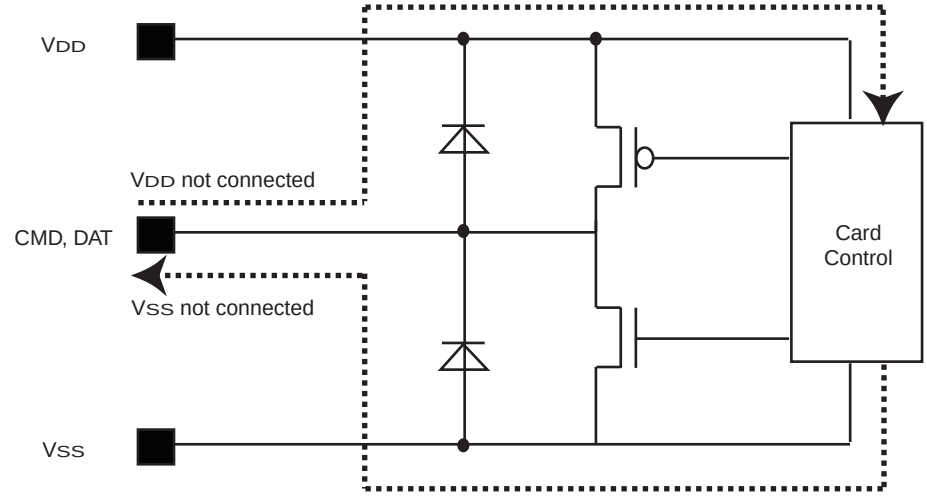


Figure 23: Improper power supply

Every MX53L1281 output withstands shortcuts to either supply.

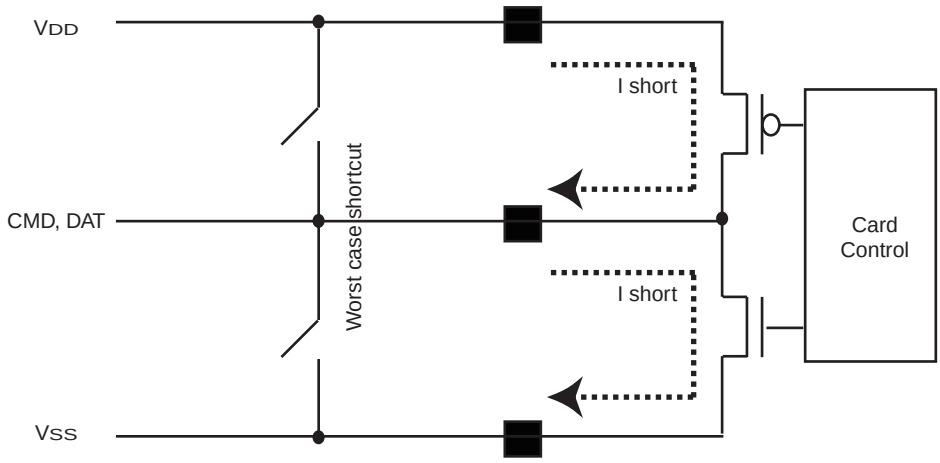


Figure 24: Short cut protection

10 Characteristics

This chapter defines the following characteristics:
- Temperature characteristics
- Electrical characteristics
- Mechanical characteristics

10.1 Temperature characteristics

Parameter	Symbol	Min	Max	Unit
Storage Temperature	TSTG	-40	85	℃
Operating temperature	TA	-20	85	℃

Table 30: Temperature characteristics

10.2 Electrical characteristics

In this chapter the electrical characteristics for the MX53L1281 are defined in three steps:
- Pad characteristics: properties of the external connectors
- Absolute maximum ratings: if exceeded the card may be damaged
- Recommended operating conditions: characterization model of the environment of the MX53L1281, requirements for the operating characteristics
- Operating characteristics: properties of the MX53L1281 measurable if the recommended operating conditions are considered

10.2.1 Pad characteristics

Parameter	Symbol	Min	Typ	Max	Unit	
Connector Resistance		10	30	100	m ohm	Counterpart is the MultiMediaCard connector defined in the MultiMediaCard system specification[1], Chapter 8, "Mechanical specification"
Input Capacitance				5	pF	

Table 31: Pad characteristics

10.2.2 Absolute maximum ratings

Absolute maximum ratings are those values beyond which damage to the device may occur. Func-tional operation under these conditions or at any other condition beyond those indicated in the operational sections of this specification is not implied:

TA -20 ... 85 ℃, VDD 2 ... 3.6V unless otherwise stated

Parameter	Symbol	Min	Max	Unit	Remark
supply voltage	V DD	-0.5	4.6	V	
total power dissipation			0.9	W	
ESD protection		-4	4	kV	Human Body Model
latch-up protection		-100	100	mA	all inputs/outputs
inputs					
input voltage	V_{Imax}	-0.5V	V_{DD} +0.5	V	$\leq V_{DDmax}$
outputs					
output voltage	V_{Omax}	-0.5V	V_{DD} +0.5	V	$\leq V_{DDmax}$
high-level output current	$\lvert I_{OH} \rvert$		100	mA	short cut protected
low-level output current	$\lvert I_{OL} \rvert$		150	mA	short cut protected

Table 32: Absolute Maximum Ratings

10.2.3 Recommended operating conditions

The recommended operating conditions define the parameter ranges for optimal performance and durability of the MX53L1281.

TA -20 ... 85 ℃, VDD 2 ... 3.6V unless otherwise stated

Parameter	Symbol	Min	Typ	Max	Unit	Remark
supply voltage	V_{DD}	2.0		3.6	V	
inputs						
low-level input voltage	V_{IL}	V_{SS}-0.3		0.25V_{DD}	V	
high-level input voltage	V_{IH}	0.625 V_{DD}		V_{DD}+0.3	V	
outputs (push pull mode)						
high-level output current	| I OH |	2			mA	
low-level output current	| I OL |	2			mA	
clock input clk (all values are referred to min(V_{IH}) and max(V_{IL}))						
clock frequency data transfer mode(pp)	f_{CLK}	0		20	MHz	<= 10 cards, $CL_{maxCMD,DAT}$ = 100pF, Rpullup = 4.7k
clock frequency data transfer mode(pp)	f_{CLK}	0		5	MHz	<= 30 cards, $CL_{maxCMD,DAT}$ = 250pF, Rpullup = 1.65k
clock frequency ident. mode (od)	f_{CLK}	0		400	kHz	<= 30 cards, $CL_{maxCMD,DAT}$ = 250pF, Ropen drain = 1.65k
clock low time	tWL	10			ns	S. Figure 20
clock high time	tWH	10			ns	S. Figure 20

Table 33. Recommended Operating Condition

10.2.4 Operating characteristics

The operating characteristics are parameters measured in a MultiMediaCard system assuming the recommended operating conditions (see "Chapter 10.2.3: Recommended operating conditions") and the temperature range as defined in "Chapter 10.1: Temperature characteristics").

The guaranteed power consumption does not include the current consumed by external units.

Parameter	Symbol	Min	Max	Unit	Remark
Supply Current	I DD1		100	uA	Standby state, Clock = 0Hz, V_{DD} = 2.7V, $V_{IH\ CMD,CLK}$ = V DD
Supply Current	I DD2		10	mA	Transfer state, Clock = 400kHz, V_{DD} = 2.7V, $V_{IH\ CMD,CLK}$ = V DD $V_{IL\ CMD,CLK}$ = GND
Supply Current	I DD3		25	mA	Transfer state, Clock = 20MHz, V_{DD} = 2.7V, $V_{IH\ CMD,CLK}$ = V DD $V_{IL\ CMD,CLK}$ = GND
Supply Current	I DD4		35	mA	Transfer state, Clock = 20MHz, V_{DD} = 3.6V, $V_{IH\ CMD,CLK}$ = V DD $V_{IL\ CMD,CLK}$ = GND

Table 34: Guaranteed power consumption

All operating characteristics are measured assuming the following load model for each output:

The standard input capacity is C_{card} =7pF. The standard bus capacitance is assumed to be C_{bus} =30pF. The maximum number of cards which can be driven in a MultiMediaCard stack at full speed (20MHz) is 10. This leads to the following reference capacitance:

$$C\,L10 = n * C\,catd + C\,bus + 10 * 7pF\ 30pF = 100pF$$

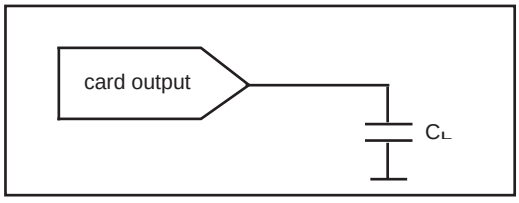


Figure 25: Test circuit

Parameter	Symbol	Min	Max	Unit	Remark
All Digital Inputs (Including I/O)					
Input Leakage Current	I_{LI}		10	uA	$0V < V_{IN} < V_{DD}$
All Outputs (push pull mode)					
High-Level Output Voltage	V_{OH}	$0.75V_{DD}$		V	at I_{OHMIN}
Low-Level Output Voltage	V_{OL}		$0.125V_{DD}$	V	at I_{OLMIN}
Low-Level Output Voltage (open drain mode)	V_{OL}		0.3V	V	at I_{OLMIN}
Command Input: CMD (Related to CLK)					
Input setup time	t ISU	3		ns	S. Figure 26
Input hold time	t IH	3		ns	S. Figure 26
Outputs: CMD, DAT (Related to CLK)					
Output setup time	t OSU	5		ns	S. Figure 26
Hold time	t OH	5		ns	S. Figure 26

Table 35. Operating characteristics

At a reduced clock rate the card can drive a load of up to 30 cards:

$$C\,L30 = n * C\,catd + C\,bus = 30 * 7pF + 40pF = 250pF$$

With this load capacity the operating frequency is reduced to 5MHz.

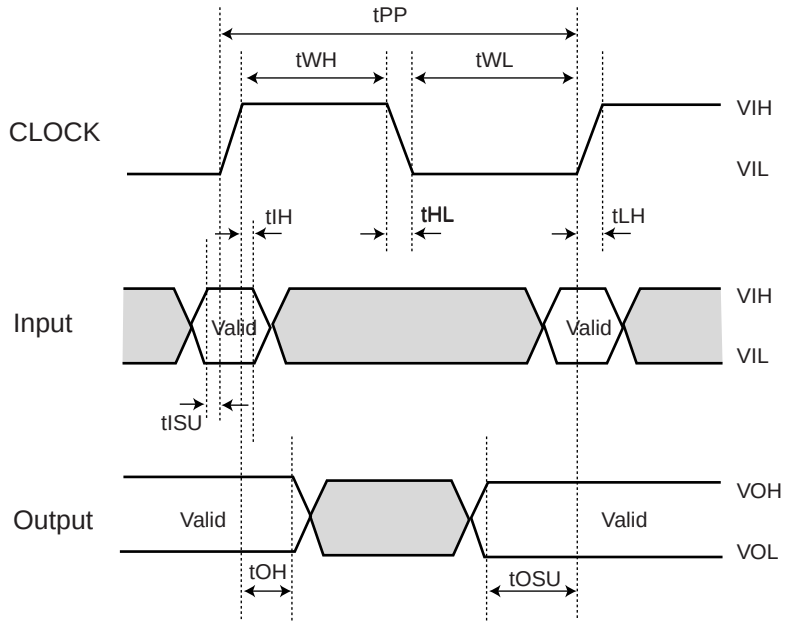


Figure 26: Timing Diagram of Data Input and Output

The access time (t_{AT}) is divided into two parts:

- T_{SAD}: The synchronous access time. This time defines the time of the maximum number of cycles which are required to access a byte of the memory field.
- T_{AAD}: The asynchronous access time to read a byte out of the memory field The synchronous part of the access time is seven cycles. At 20 MHz one cycle is 50 ns (1/f CLK), multiplied with NSAD the resulting frame time is T SAD =15 us. The asynchronous access delay of the MX53L1281 is T AAD =0. The resulting memory access time t AT is equal to the sum of both parts:

$$tAT = tAAD + TSAD \qquad \text{with} \qquad TSAD = \frac{NSAD}{fCLK}$$

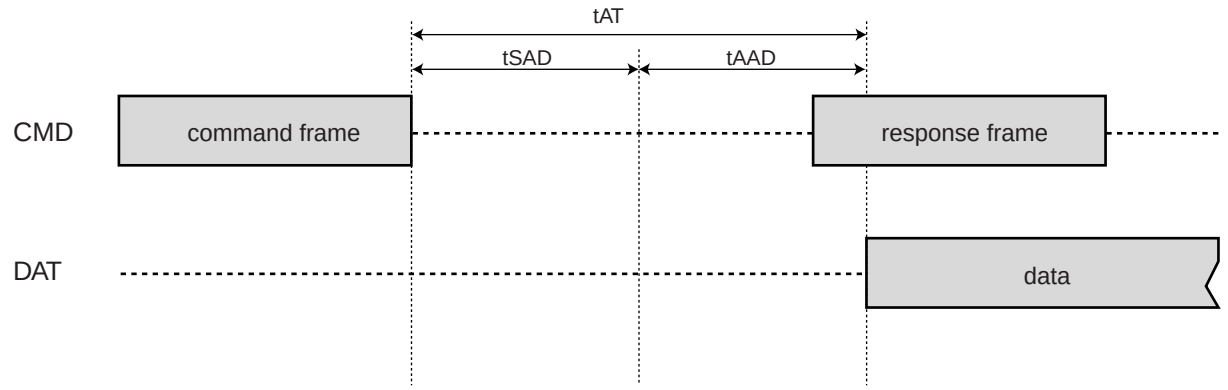


Figure 27: Access Time

Parameter	Symbol	Max	Unit	Remark
Synchronous access delay cycles	N_{SAD}	300	cycles	
Synchronous access delay	T_{SAD}	15	us	@20MHZ clock frequency
Asynchronous access delay	T_{AAD}	0	us	
Memory access time	t_{AT}	15	us	@20 MHZ clock frequency

Table 36: Access Time

In the CSD are two fields to code the asynchronous and the synchronous access delay time:

- TAAC, asynchronous access delay
- NSAC, maximum number of cycles for receiving and interpreting of a command frame
  The value for the CSD field NSAC is calculated from NSAD (maximum: 300 cycles) by division with 100 and rounding up to the next integer:
- NSAC=0x03(300 cycles)
- TAAC=0x08(1ns)

For more details on NSAC and TAAC CSD-entries see "Chapter 5.3: Card specific data (CSD)".

10.3 Mechanical characteristics

- MX53L1281 form factor: 24mm x 32mm x 1.4mm (WxLxH):

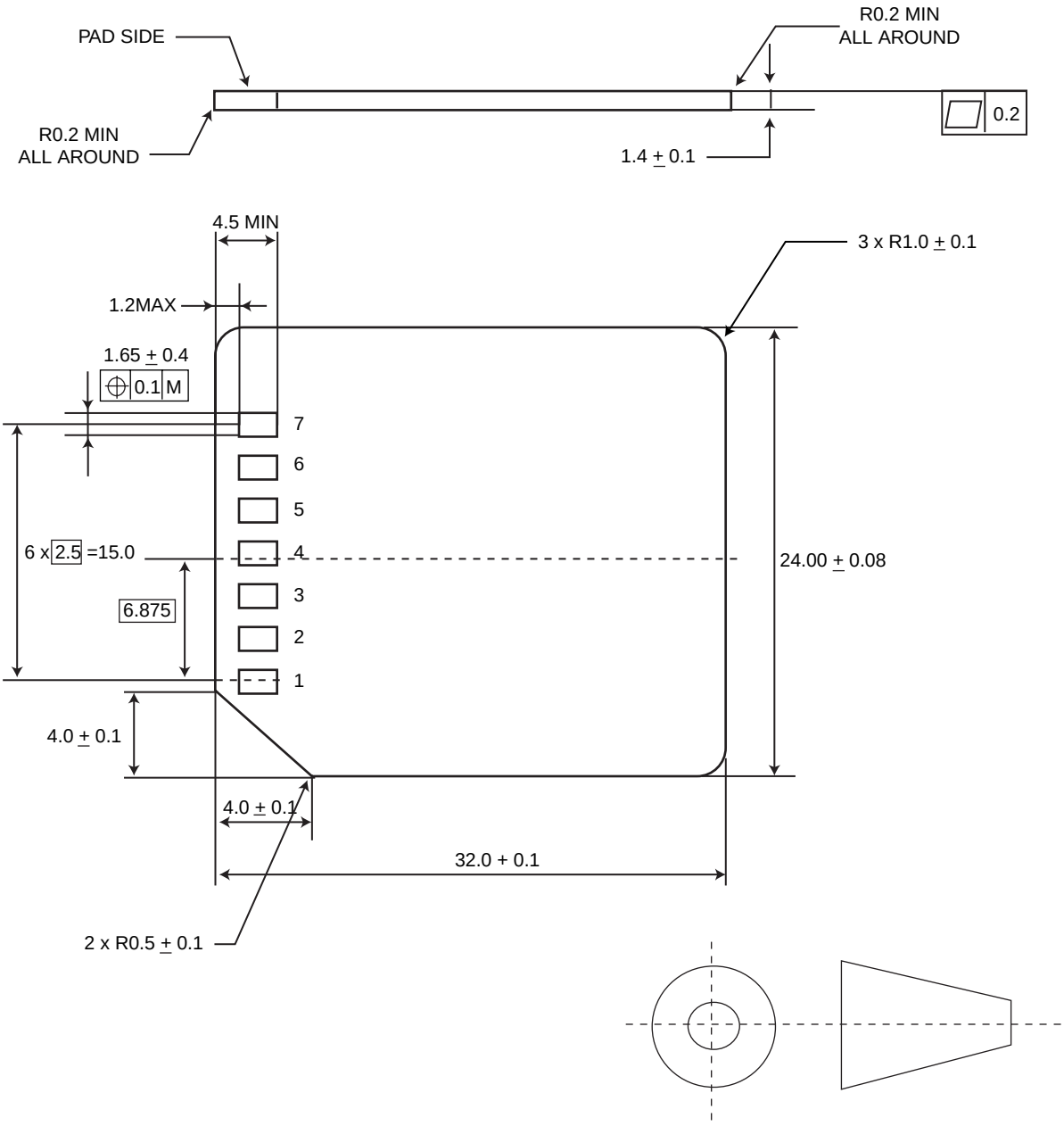


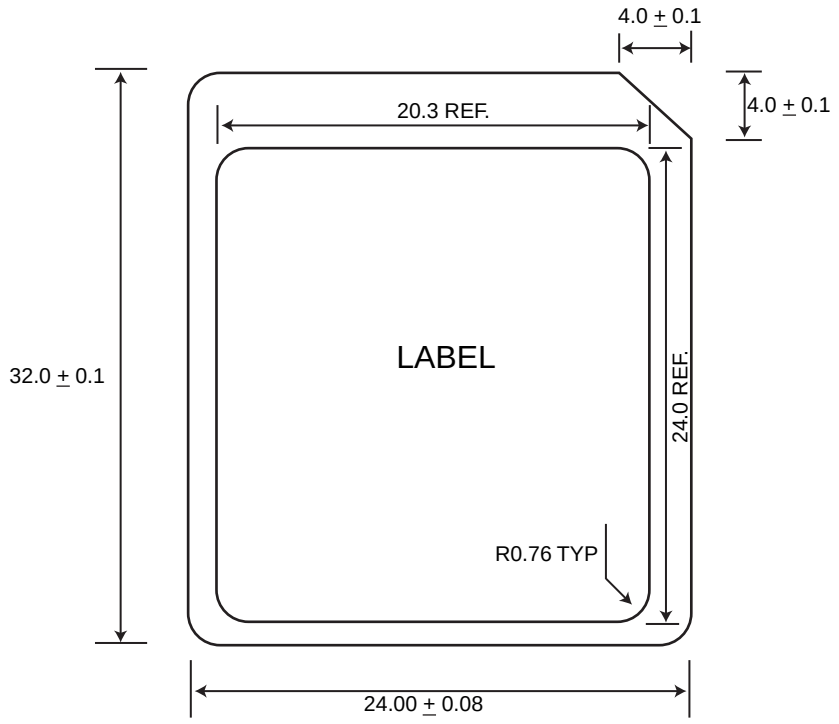
Figure 28: ROM MultiMediaCard shape

Additional informations regarding mechanical topics, like connectors, environmental and testing parameters are part of a special document: [2].

10.4 Label Dimension

Each card has two sides : "marking" side (the side with 7 pins) shows the factory's manufacturing information; "label" side is an area where where the label is stuck on the surface for the purpose of promotion and advertisement.

Label size is defined as following :



Label thickness: 0.076mm REF.

11.Application notes

Additional application specific informations are available for the following topics:
A guideline for MultiMediaCard integration into application: The MultiMediaCard Adapter[2].
This document describes also an generic VHDL model of the adapter.

12.References

[1] The MultiMediaCard, System Specification 1.4, MultiMediaCard Definition Group, March 1998
[2] The MultiMediaCard Adapter, Version 5.1, SIEMENS AG, June 1998

13.Number representations

- hexadecimal numbers: 0xAB, leading 0x, each digit represents 4 bits.
- binary numbers (single bit):"0".
- binary number (unsigned bit vector):"100100".
- 1k is equal to 1024.
- 1M is equal to 1k * 1k.

MACRONIX INTERNATIONAL CO., LTD.

HEADQUARTERS:
TEL:+886-3-578-6688
FAX:+886-3-563-2888

EUROPE OFFICE:
TEL:+32-2-456-8020
FAX:+32-2-456-8021

JAPAN OFFICE:
TEL:+81-44-246-9100
FAX:+81-44-246-9105

SINGAPORE OFFICE:
TEL:+65-348-8385
FAX:+65-348-8096

TAIPEI OFFICE:
TEL:+886-2-2509-3300
FAX:+886-2-2509-2200

MACRONIX AMERICA, INC.
TEL:+1-408-453-8088
FAX:+1-408-453-8488

CHICAGO OFFICE:
TEL:+1-847-963-1900
FAX:+1-847-963-1909

http : //www.macronix.com