Siemens AG   **Semiconductors**

**MultiMediaCard R0008
8 Mbyte ROM**

Product Manual

Preliminary Product Manual Version 3.1 (08.98)

Attention please!

As far as patents or other rights of third parties are concerned, liability is only assured for components, not for application, processes and circuits implemented within components or assemblies.

The information describes the type of component and shall not be considered as assured characteristics.

Terms of delivery and rights to change design reserved.

For questions on technology, delivery and prices, please contact the Semiconductor Group Offices in Germany or the Siemens companies and the Representatives world-wide (see address list).

Due to technical requirements components may contain dangerous substances. For information on the types in question please contact your nearest Siemens Office,
Semiconductor Group

Siemens AG is an approved CECC manufacturer.

Packing

Please use the recycling operators known to you. We can also help you - get in touch with your nearest sales office. By agreement we will take packaging material back, if it is sorted. You must bear the cost of transport.

For packaging material that is returned to us unsorted or which we are not obliged to accept, we shall have to invoice you for any costs incurred.

Components used in life-support devices or systems must be expressly authorized for such purpose!

Critical components [1] of the Semiconductor Group of Siemens AG, may only be used in life-support devices or systems [2] with the express written approval of the Semiconductor Group of Siemens AG.

1)
A critical component is a component used in a life-support device or system whose failure can reasonably be expected to cause the failure of that life-support device or system, or to affect its safety or effectiveness of that device or system.

2)
Life-support Devices or systems are intended (a) to be implanted in the human body, or (b) to support and/or maintain and sustain human life. If they fail, it is reasonable to assume that the health of the user may be endangered.

1 General Description

The SIEMENS MultiMediaCard R0008 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 through connecting their peripheral contacts. The R0008 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

- **8 MByte memory capacity**
- Payload: 7,888,896 Bytes
- Error correction: 499,712 Bytes

- **On card error correction**

- **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 and 2048 bytes per block
- Multiple block mode supported
- CRC protected data communication
- 2.0V to 3.6V operation voltage range
- Damage free powered card insertion and removal

- **High speed serial interface with random access in block or serial mode**
- Byte addressable memory
- up to 10 stacked card @ 20MHz @ 2.0-3.6V
- up to 30 stacked card @ 5MHz @ 2.0-3.6V
- Access time < 13 μs @ 20MHz @ 2.0-3.6V, random byte access

- **Low power dissipation**
- High speed: < 20 mW @ 20MHz @ 2.0V
- Low power: < 1 mW @ 100kHz @ 2.0V
- Power save: < 0.1 mW @ 0Hz @ 2.0V (in stby state)

3   Overview

The following diagram shows an overview of the R0008 internal architecture:
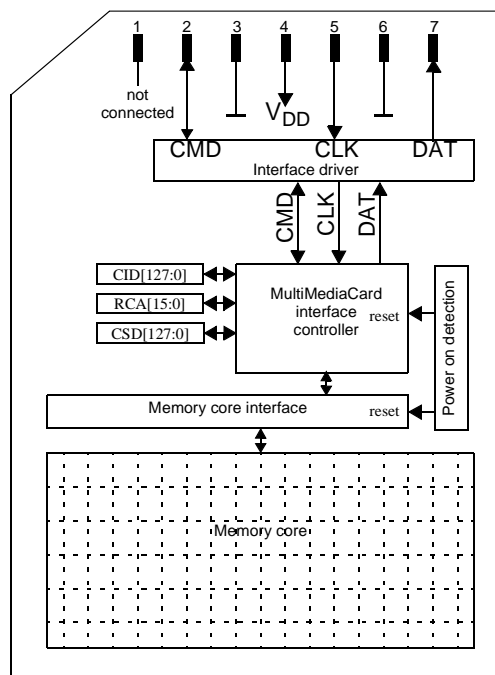
Figure 1: R0008 architecture

All controllers in the R0008 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: Interface" for more details). For the identification of the R0008 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 MultiMediaCard system is powered up (see "Chapter 9: Power supply").

4   Interface

In the R0008 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 R0008 and responses vice versa.
- DAT: is a data channel with a width of one line. The DAT signal of the R0008 operates in push pull mode.
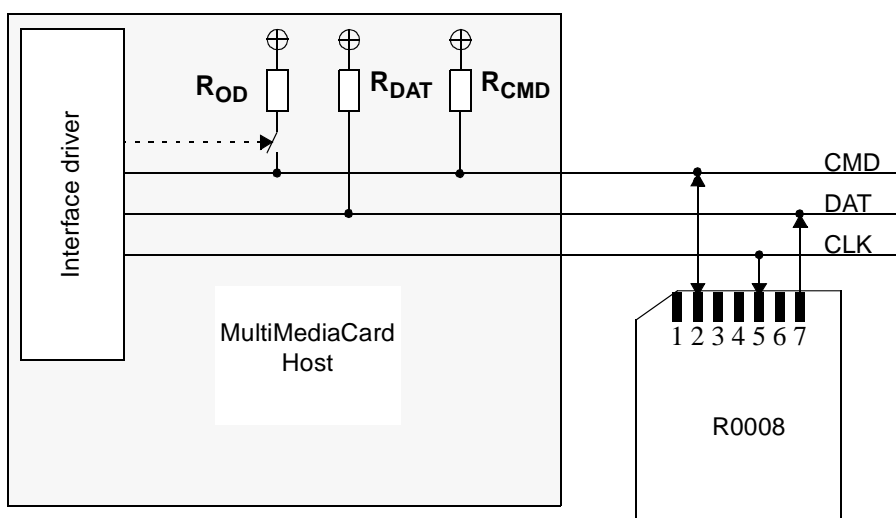
Figure 2: R0008 interface

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: R0008 pad definition

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

Pin 1 is not connected in the R0008:
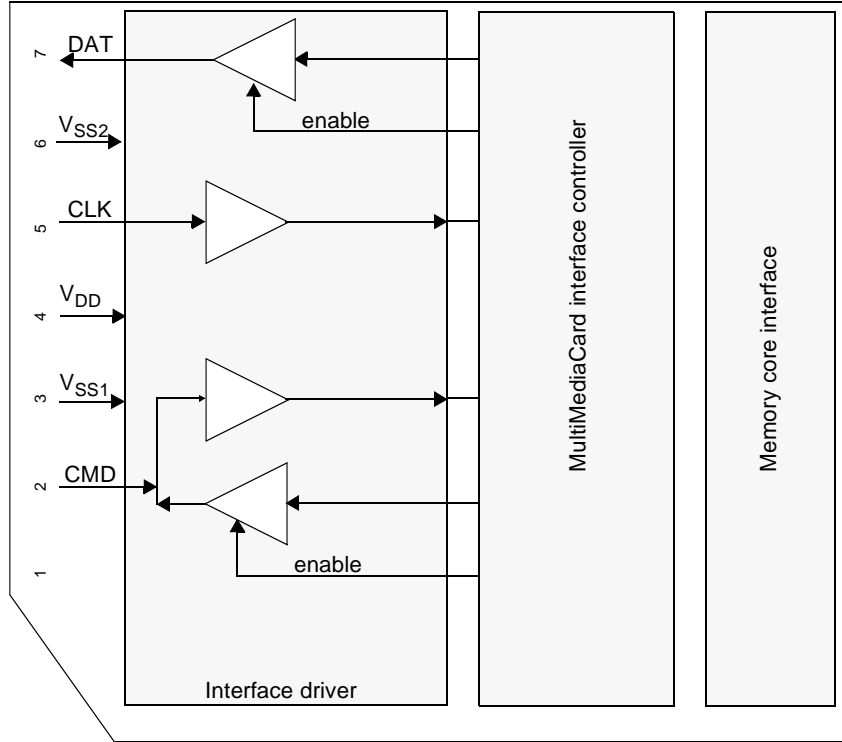
Figure 3: R0008 I/O-drivers

5   Registers

The R0008 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: R0008 registers

CID and RCA are used for identifying and addressing the R0008. The third register contains the card specific data record. This record is a set of information fields to define the operation conditions of the R0008.

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)

This registers content is the card identification information used during the card identification procedure. It is a 128 bit wide register. The CID is divided into three slices:

Name	Field	Width	CID-slice
Manufacturer ID	MID	24	[127:104]
Card individual number	CIN	96	[103:8]
CRC checksum	CRC	7	[7:1]
not used, always 1	-	1	[0:0]

Table 3: CID fields

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

$M(x) = CID[127] * x^{119} + ... + CID[8] * x^0$ $CRC[6...0] = \text{Remainder } [(M(x) * x^7)/G(x)]$

In the R0008 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 R0008.

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.

Name	Field	Width	CSD-slice	Value
CSD structure	CSD_STRUCTURE	2	[127:126]	1
MultiMediaCard protocol version	MMC_PROT	4	[125:122]	1
reserved	-	2	[121:120]	don't care
data read access-time-1	TAAC	8	[119:112]	0x6A (0.6 µs)
data read access-time-2 in CLK cycles (NSAC*100)	NSAC	8	[111:104]	0x03 (300 cycles)
max. data transfer rate	TRAN_SPEED	8	[103:96]	0x2A (20 Mbit/s)
card command classes	CCC	12	[95:84]	0x007 (class 0, 1, 2)
max. read data block length	READ_BLK_LEN	4	[83:80]	0xB (2048 bytes)
partial blocks for read allowed	READ_BLK_PARTIAL	1	[79:79]	'1'
write block misalignment	WRITE_BLK_MISALIGN	1	[78:78]	don't care
read block misalignment	READ_BLK_MISALIGN	1	[77:77]	'1'
DSR implemented	DSR_IMP	1	[76:76]	'0'
reserved	-	2	[75:74]	don't care
device size	C_SIZE	12	[73:62]	962
max. read current @V_{DD} min	VDD_R_CURR_MIN	3	[61:59]	0x3 (10mA)
max. read current @V_{DD} max	VDD_R_CURR_MAX	3	[58:56]	0x3 (25mA)
max. write current @V_{DD} min	VDD_W_CURR_MIN	3	[55:53]	don't care
max. write current @V_{DD} max	VDD_W_CURR_MAX	3	[52:50]	don't care
device size multiplier	C_SIZE_MULT	3	[49:47]	0
erase sector size	SECTOR_SIZE	5	[46:42]	don't care
erase group size	ERASE_GRP_SIZE	5	[41:37]	don't care
write protect group size	WP_GRP_SIZE	5	[36:32]	don't care
write protect group enable	WP_GRP_ENABLE	1	[31:31]	don't care
manufacturer default ECC	DEFAULT_ECC	2	[30:29]	don't care
write speed factor	R2W_FACTOR	3	[28:26]	don't care

Table 4: The CSD fields

Name	Field	Width	CSD-slice	Value
max. write data block length	WRITE_BLK_LEN	4	[25:22]	don't care
partial blocks for write allowed	WRITE_BLK_PARTIAL	1	[21:21]	don't care
reserved	-	5	[20:16]	don't care
reserved	-	1	[15:15]	don't care
copy flag (OTP)	COPY	1	[14:14]	don't care
permanent write protection	PERM_WRITE_PROTECT	1	[13:13]	1
temporary write protection	TMP_WRITE_PROTECT	1	[12:12]	1
reserved	-	2	[11:10]	don't care
ECC code	ECC	2	[9:8]	0
CRC	CRC	7	[7:1]	0x30
not used, always 1	-	1	[0:0]	1

Table 4: The CSD fields

All CSD fields are read only for the user. All don't care tagged fields are zero.

The following section describes the CSD fields and their values for the R0008:

- **CSD_STRUCTURE**

CSD_STRUCTURE	CSD register structure
"01"	CSD version No. 1.1

Table 5: CSD register structure

The CSD version of the R0008 is related to the CSD version 1.1 as defined in "MultiMediaCard system specification, Version 1.4". The parameter CSD_STRUCTURE is permanently assigned to the value 1.

- **MMC_PROT**

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.

MMC_PROT	MultiMediaCard protocol version
"0001"	MultiMediaCard protocol version 1.4

Table 6: MultiMediaCard protocol version

The MultiMediaCard protocol version of the R0008 is related to the "MultiMediaCard system specification, Version 1.4". The parameter MMC_PROT is permanently assigned to the value 1.

- **TAAC**

Defines the asynchronous data access time:

TAAC bit	Description	Values
2:0	time unit	0=1ns, 1=10ns, 2=100ns, 3=1µs, 4=10µs, 5=100µs, 6=1ms, 7=10ms

Table 7: TAAC access time definition

TAAC bit	Description	Values
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 7: TAAC access time definition

The value for the asynchronous delay for the R0008 is 0.3 μs. The coded TAAC value is 0x3A. 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 R0008 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 8: Maximum data transfer rate definition

The R0008 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 15.

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

Table 9: Supported card command classes

The R0008 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 10: 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 R0008 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 R0008 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 11: DSR implementation

The R0008 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:

$$BLOCKLEN = 2^{READBLKLEN} = 2048 \qquad (READBLKLEN < 12)$$

$$MULT = 2^{CSIZEMULT+2} = 4 \qquad (CSIZEMULT < 8)$$

$$BLOCKNR = (CSIZE + 1) \cdot MULT = 3852$$

$$SIZE = BLOCKNR \cdot BLOCKLEN = 7,888,896 Byte$$

The memory size of the R0008 is 8MByte. The parameter C_SIZE is 962, the parameter C_SIZE_MULT is 0 and the parameter READ_BLK_LEN is 0xB (see above).

- **VDD_R_CURR_MIN**

The maximum supply current at the minimal supply voltage V_{DD} (2.0 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 12: Supply current consumption @ V_{DD}=2.0V

The parameter **VDD_R_CURR_MIN** is permanently assigned to the value 3 (10 mA).

For more details see "Chapter 9.3: Power consumption".

- **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 13: Supply current consumption @ V_{DD} =3.6V

The parameter **VDD_R_CURR_MAX** is permanently assigned to the value 3 (25 mA).

For more details see "Chapter 9.3: Power consumption".

- **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'.

- **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

Table 14: ECC type

ECC	ECC type	Maximum number of correctable bits
1	BCH (542,512)	3
2-15	reserved	-

Table 14: ECC type

No external error correction is needed for the R0008. 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: $G(x) = x^7 + x^3 + 1$
$M(x) = CSD[127]*x^{119} +...+ CSD[8]*x^0$

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

This parameters value is 0x31 for the R0008.

6 Communication

All communication between host and cards is controlled by the host (master). The host sends commands and, depending on the command, receives a corresponding response from the selected card. In this chapter the commands to control the R0008, 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 R0008 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 15: R0008 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 µs @ 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 16 and Table 17). The CRC field is defined in "Chapter 7: Error handling".

The R0008 supports the following MultiMediaCard commands:

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 dis-connect 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 address[3]	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 16: 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 17: 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).
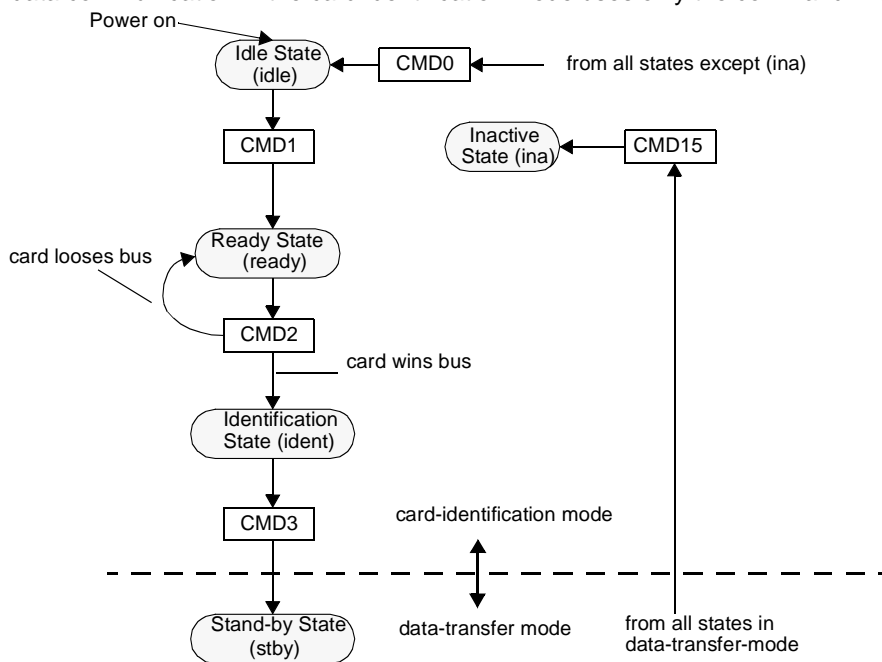

Figure 4: MultiMediaCard state diagram (card identification mode)

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 com-

mon 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 identification 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 R0008 supports the full range of 2.0 to 3.6V supply voltage. So the R0008 sends a R3 response to CMD1 which contains an OCR value of 0xFFFFFFFF (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.
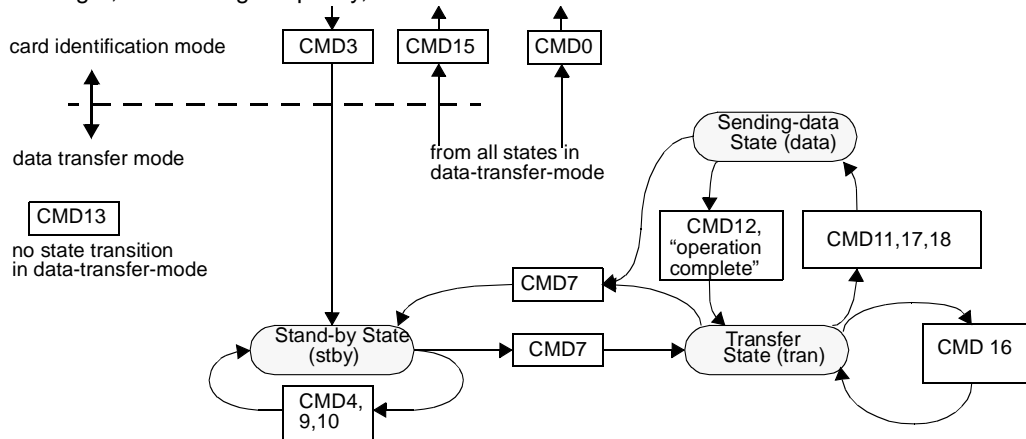

Figure 5: R0008 state diagram (data transfer mode)

The broadcast command SET_DSR (CMD4) configures the driver stages of all identified cards. It programs their DSR registers corresponding to the application bus layout (length) and the number of cards on the bus and the data transfer frequency. The clock rate is also switched from f_{OD} to f_{PP} at that point.

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 maximum 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 R0008.

- **State transition summary**

The Table 18 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 18: 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 R0008 supports the full range from 2.0 to 3.6 V. Respectively the value of all bits of the OCR field of the R0008 are always set to high (0xFFFFFFFF). The reserved bits are also high (0x3F and 0x7F). So the R3 frame of the R0008 contains always the value 0x3FFFFFFFFFFF.

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 following 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	E R	'0'= no error '1'= error	The commands argument was out of allowed range for this card.	C
30	Don't care		Permanently 0.		
29	BLOCK_LEN_ERROR	E R	'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	E R	'0'= no error '1'= error	The CRC check of the previous command failed.	B
22	ILLEGAL_COMMAND	E R	'0'= no error '1'= error	Command not legal for the current state	B
21	Don't care		Permanently 0.		
20	CC_ERROR	E R X	'0'= no error '1'= error	Internal card controller error	C
19	Don't care		Permanently 0.		
18	UNDERRUN	E X	'0'= no error '1'= error	The card could not sustain data transfer in stream read mode	C
17:13	Don't care		Permanently 0.		

Table 19: Status

Bits	Identifier	Type	Value	Description	Clear Condition
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 19: 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 20: 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 R0008 the following values are defined:

	Value [clock cycles]	Description
N_{CR}	3	Number of cycles between command and response
N_{ID}	5	Number of cycles between card identification or card operation conditions command and the corresponding response
N_{AC}	≥3[1]	

Table 21: Timing values

	Value [clock cycles]	Description
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 21: Timing values

[1]see "Chapter 10.2: Electrical characteristics" for more details about the access time

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:
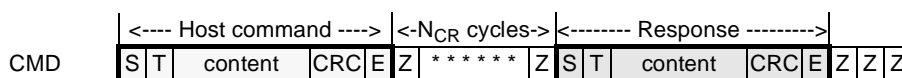

Figure 6: Command response timing (identification mode)

There is a two Z bit period followed by P bits pushed up by the responding card. The following timing 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 N_{ID} clock cycles.
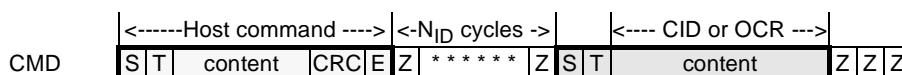

Figure 8: Identification timing (card identification mode)

- **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.

```
        <-------- Response --------> <-N_RC cycles -> <---- Host command ----->
CMD     |S|T|  content  |CRC|E| |Z| * * * * * * |Z| |S|T|  content  |CRC|E|
```
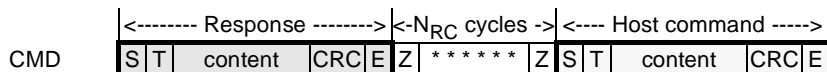
Figure 9: Timing response end to next CMD start (data transfer mode)

- **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.
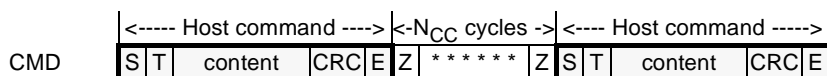
```
        <----- Host command ----> <-N_CC cycles -> <---- Host command ----->
CMD     |S|T|  content  |CRC|E| |Z| * * * * * * |Z| |S|T|  content  |CRC|E|
```

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.
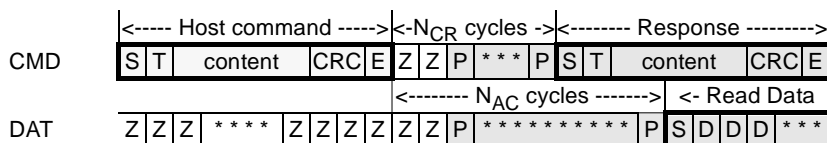
```
        <----- Host command -----> <-N_CR cycles -> <-------- Response --------->
CMD     |S|T|  content  |CRC|E| |Z|Z|P| * * * |P| |S|T|  content  |CRC|E|
                                 <-------- N_AC cycles -------> <- Read Data
DAT     |Z|Z|Z| * * * * |Z|Z|Z|Z| |Z|Z|P| * * * * * * * * * * |P| |S|D|D|D| * * *
```

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.
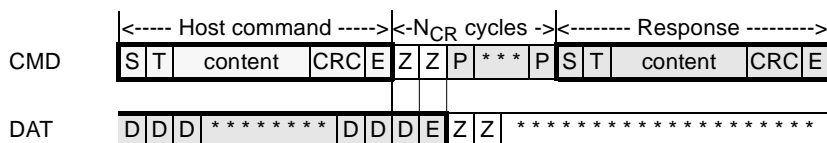
```
        <----- Host command -----> <-N_CR cycles -> <-------- Response --------->
CMD     |S|T|  content  |CRC|E| |Z|Z|P| * * * |P| |S|T|  content  |CRC|E|
DAT     |D|D|D| * * * * * * * |D|D|D|E| |Z|Z| * * * * * * * * * * * * * * * * * *
```

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

- **Data transfer stop - next data block transfer timing**

The data transmission stops immediately with the end bit of the host CMD. The next data block transmission starts with the access time delay t_{AC}, beginning from the end bit of the data address
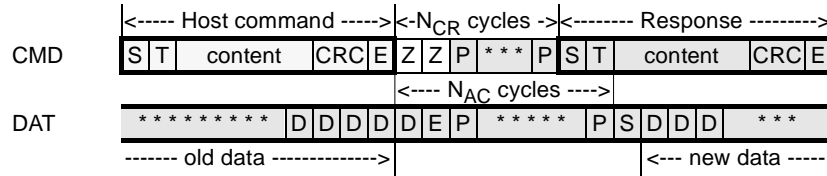
CMD.

Figure 13: Next data block transfer timing diagram

6.8 Reset

GO_IDLE_STATE (CMD0) is the software reset command, which sets the R0008 into the *Idle State* independently of the current state. In the *Inactive State* the R0008 is not affected by this command.

After power-on the R0008 is always in the *Idle State.* After power-on or command GO_IDLE_STATE (CMD0) all output bus drivers of the R0008 are in a high-impedance state and the card will be initialized with a default relative card address ("0x0001") The host runs the bus at the identification clock rate f_{OD} generated by a push-pull driver stage (see also "Chapter 9.2: Power on" for more details).

7 Error handling

The R0008 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).

7.1 CRC

Following the MultiMediaCard standard, the R0008 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 polynomial:

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

One CRC is checked in the R0008 for every command. For each response a CRC is generated in the R0008. On CRC failure the command will be ignored and a response is sent to initiate a repetition of the command by the host. Each data block read from the R0008 will be succeeded by redundancy bits generated with the second CRC. The code is usable for payload lengths of up to 2048 Bytes:

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

7.2 On card error correction

The R0008 includes a error correction unit which detects and corrects card internal errors. The correction is done just in time, when the data is read out. The correction is done by piping the data through the ECC unit, so that no detectable speed down is induced. The complete ECC is covered for the user, so that he must not care about handling statical errors in the device.

8   Programming mask format

The data format to deliver the mask data for the R0008 from the provider to the manufacturer is the "Intel Hex Object File Format".

- This format is built of ASCII characters. Valid characters are ':',"0-9" and "A-F".
- An object file contains an unlimited number of records.
- Each object file ends with an end of file record.
- Each record has the following format (spaces are skipped in real records):
  m ll oooo tt dd dd dd dd dd dd ... dd dd cc
- 'm' is always a colon ':'.
- 'll' is the hexadecimal coded two digit (= 8 bit) record length. Only "real" data bits are counted.
- 'oooo' is the hexadecimal coded four digit load offset for the record.
- 'tt' is the hexadecimal coded two digit record type:
  - 00-Data record
  - 01-End of file record
  - 04-Extended linear address record
- 'dd' are the hexadecimal coded two digit data bytes. The maximum number of bytes is given by the eight bit length coded in 'll' which leads up to 256 data bytes per record.
- Each record is closed by a hexadecimal coded two digit check sum. The sum over all data bits (modulo 256) of the entire record (including length, offset, type and data) and the check sum is equal to '00'.

8.1   Data record

The type field of a data record contains the value "00". A typical data record which contains the values from 0 to 9 looks like this:

	Length	Offset	Type	Data	Checksum
:	0A	0000	00	00010203040506070809	C9

8.2   End of file record

The end of file record is always the last record of a file. The type field of a data record contains the value "01":

	Length	Offset	Type	Data	Checksum
:	00	0000	01	--	FF

8.3   Extended linear address record

The type field of an extended segment address (ESA) record contains the value "04":

	Length	Offset	Type	Data	Checksum
:	02	0000	04	0001	F9

The number of data bytes is always two. The data field contains the bits 16 to 31 of the data address. The start address of record storage locations is calculated as the sum of the linear address record multiplied with 65536 and the 16 bit record offset (RO) coded in the data record (see "Chapter 8.1: Data record"): The result is a 32 bit address (ADDR):

$$ADDR = ESA \cdot 65536 + RO$$

The resulting address is the address of the first byte of the data record.

A typical file may then look like this:

```
:020000040001F9
:0A00000000010203040506070809C9
:00000001FF
```

In this example bytes with the values 0 to 9 are loaded into the locations from 0x00010000 to 0x00010009.

8.4   Address mapping

The memory contents are stored under addresses up to the highest valid address given by the device capacity defined in the CSD fields C_SIZE and C_SIZE_MULT (see "Chapter 5.3: Card specific data (CSD)"). The covered addresses for the R0008 are 0x00000000-0x00FFFFFF. The 64 kByte page starting with 0xFFFF0000 is reserved for the CID.

Description	Address	
	First element	**Last element**
Data	0x00000000	0x00FFFFFF
CID	0xFFFF0000	0xFFFF000F

Table 22: Object file addresses

The following example shows how to define the special register values for the R0008:

```
:02000004FFFFFC
:0F000000<CID><checksum>
:00000001FF
```

The CID field contains the complete CID as defined in "Chapter 5.1: Card identification (CID)" including bit zero which is always 1.

All memory locations which are not covered by a data record are filled with the default value. The default value for the R0008 is the logical '0'.

Each file must contain a CID record.

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 14.
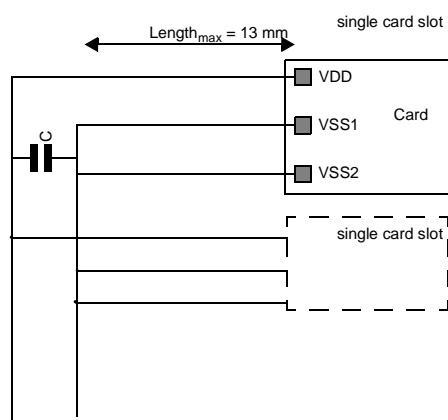

Figure 14: Power supply decoupling

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.
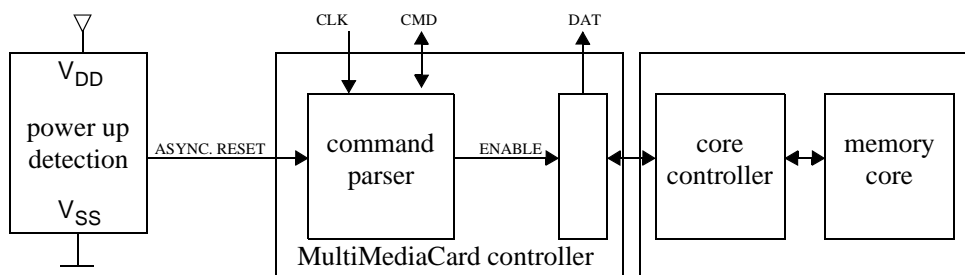

Figure 15: Power on detection

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 R0008 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 R0008 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 inter-
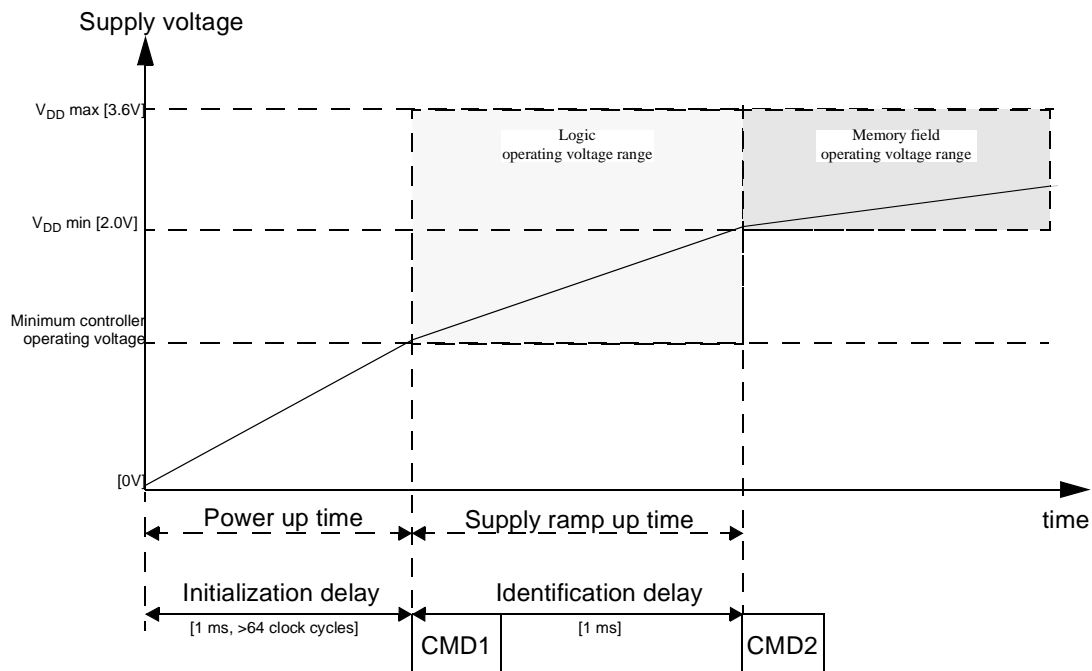
preted by the card:

Figure 16: Power up diagram

For the R0008 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 23: 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 R0008 ignores all commands until the sequence CMD1, CMD2 is received and the RCA of the card is initialized. 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 R0008 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 R0008 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	2V	3.6V
Clock off	$0\ Hz^{1}$	stby	<1µA <2µW	<1µA <3.6µW
		data, tran	100µA 200µW	150µA 540µW
Low speed	100 kHz	stby	20µA 40µW	35µA 126µW
		data, tran	100µA 200µW	150µA 540µW
Initialization	$400\ kHz^{2}$	idle, ready, ident, ina	60µA 120µW	100µA 360µW
High speed	20 MHz	stby	3mA 6mW	5.5mA 19.8mW
		data, tran	5mA 10mW	8.5mA 30.6mW

Table 24: Typical R0008 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 R0008 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.
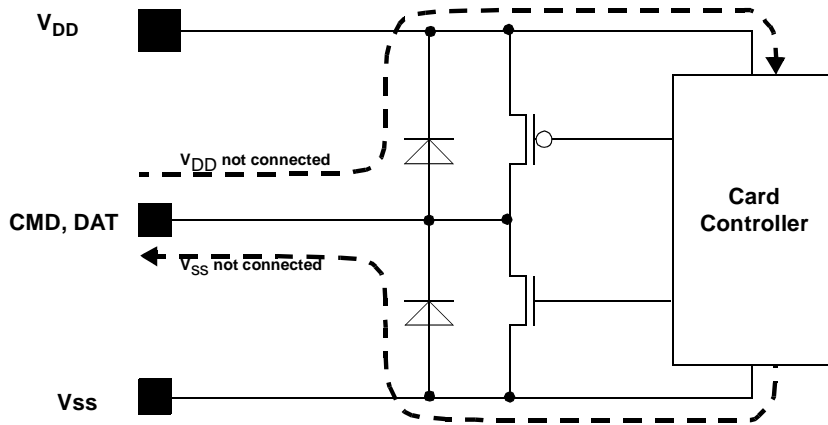

Figure 17: Improper power supply
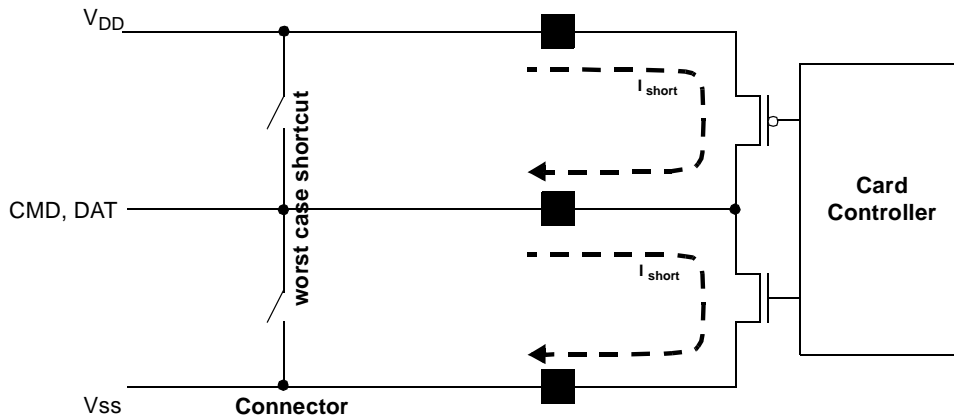
Every R0008 output withstands shortcuts to either supply.

Figure 18: 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	T_{STG}	-40	85	°C	
Operating temperature	T_A	-20	85	°C	

Table 25: Temperature characteristics

10.2   Electrical characteristics

In this chapter the electrical characteristics for the R0008 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 R0008, requirements for the operating characteristics

• Operating characteristics: properties of the R0008 measurable if the recommended operating conditions are considered

10.2.1   Pad characteristics

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

Table 26: Pad characteristics

10.2.2 Absolute maximum ratings

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

T_A -20 ... 85 $^{\circ}$C, 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	$|I_{OH}|$		100	mA	short cut protected
low-level output current	I_{OL}		150	mA	short cut protected

Table 27: Absolute Maximum Ratings

10.2.3 Recommended operating conditions

The recommended operating conditions define the parameter ranges for optimal performance and durability of the R0008.

T_A -20 ... 85°C, 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, CLmaxCMD.DAT = 100pF, Rpullup = 4.7k
clock frequency data transfer mode (pp)	f_{CLK}	0		5	MHz	<= 30 cards, CLmaxCMD.DAT = 250pF, Rpullup = 1.65k
clock frequency ident. mode (od)	f_{CLK}	0		400	kHz	<= 30 cards, CLmaxCMD.DAT = 250pF, Ropen drain = 1.65k
clock low time	t_{WL}	10			ns	S. Figure 20
clock high time	t_{WH}	10			ns	S. Figure 20

Table 28: Recommended Operating Conditions

Parameter	Symbol	Min	Typ	Max	Unit	Remark
clock input rise time	t_{LH}			10	ns	S. Figure 20
clock input fall time	t_{HL}			10	ns	S. Figure 20

Table 28: Recommended Operating Conditions

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_{DD}		10	µA	Standby state, Clock = 0Hz, V_{DD} = 2V, $V_{IH\ CMD,CLK}$ = V_{DD}
Supply Current	I_{DD}		200	µA	Transfer state, Clock = 400kHz, V_{DD} = 2V, $V_{IH\ CMD,CLK}$ = V_{DD} $V_{IL\ CMD,CLK}$ = GND
Supply Current	I_{DD}		4	mA	Transfer state, Clock = 20MHz, V_{DD} = 2V, $V_{IH\ CMD,CLK}$ = V_{DD} $V_{IL\ CMD,CLK}$ = GND
Supply Current	I_{DD}		8	mA	Transfer state, Clock = 20MHz, V_{DD} = 3.6V, $V_{IH\ CMD,CLK}$ = V_{DD} $V_{IL\ CMD,CLK}$ = GND

Table 29: 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:

$$C_{L10} = n \cdot C_{catd} + C_{bus} = 10 \cdot 7pF + 30pF = 100pF$$
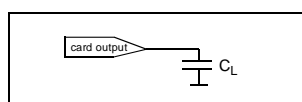

Figure 19: Test circuit

Parameter	Symbol	Min	Max	Unit	Remark
All Digital Inputs (Including I/O)					
Input Leakage Current	I_{LI}		10	µA	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}

Table 30: Operating characteristics

Parameter	Symbol	Min	Max	Unit	Remark
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 20
Input hold time	t_{IH}	3		ns	S. Figure 20
Outputs: CMD, DAT (Related to CLK)					
Output setup time	t_{OSU}	5		ns	S. Figure 20
Hold time	t_{OH}	5		ns	S. Figure 20

Table 30: Operating characteristics

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

$$C_{L30} = n \cdot C_{catd} + C_{bus} = 30 \cdot 7pF + 40pF = 250pF$$

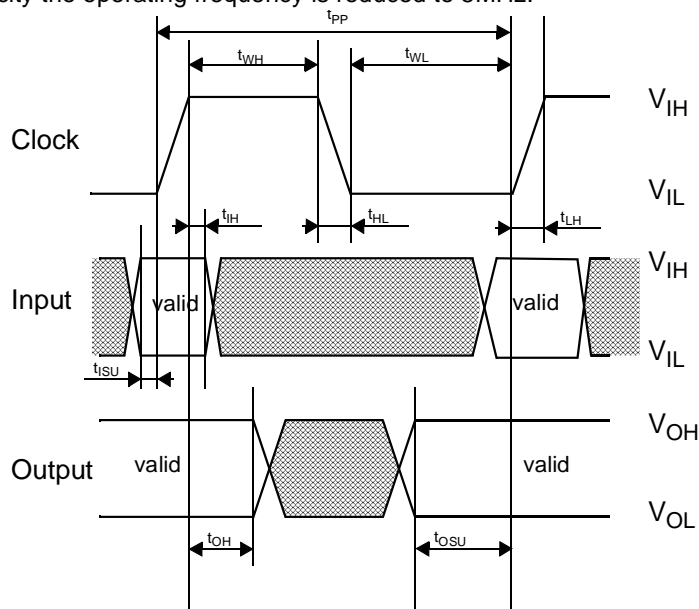With this load capacity the operating frequency is reduced to 5MHz.

Figure 20: 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 N_{SAD} the resulting frame time is $T_{SAD}=15$ μs. The asynchronous access delay of the R0008 is $T_{AAD}=0.6$ μs maximum. The resulting memory access time t_{AT} is equal to the sum of both parts:

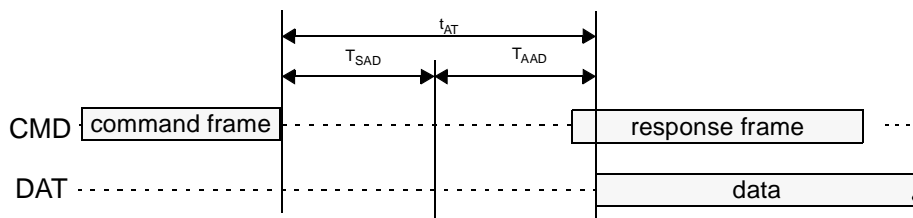$$t_{AT} = T_{AAD} + T_{SAD} \qquad with \qquad T_{SAD} = \frac{N_{SAD}}{f_{CLK}}$$

Figure 21: Access time

Parameter	Symbol	Max	Unit	Remark
Synchronous access delay cycles	N_{SAD}	300	cycles	
Synchronous access delay	T_{SAD}	15	µs	@ 20 MHZ clock frequency
Asynchronous access delay	T_{AAD}	0.6	µs	
Memory access time	t_{AT}	15.6	µs	@ 20 MHZ clock frequency

Table 31: 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 N_{SAD} (maximum: 300 cycles) by division with 100 and rounding up to the next integer:

$$NSAC = \left\lceil \frac{N_{SAD}}{100} \right\rceil = \left\lceil \frac{300}{100} \right\rceil$$

• NSAC=0x03

The value for the CSD field TAAC is rounded up to 0.6µs:

$$TAAC = \left\lceil T_{AAD} \right\rceil = 6 \cdot 100ns$$

• TAAC=0x6A

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

10.3   Mechanical characteristics

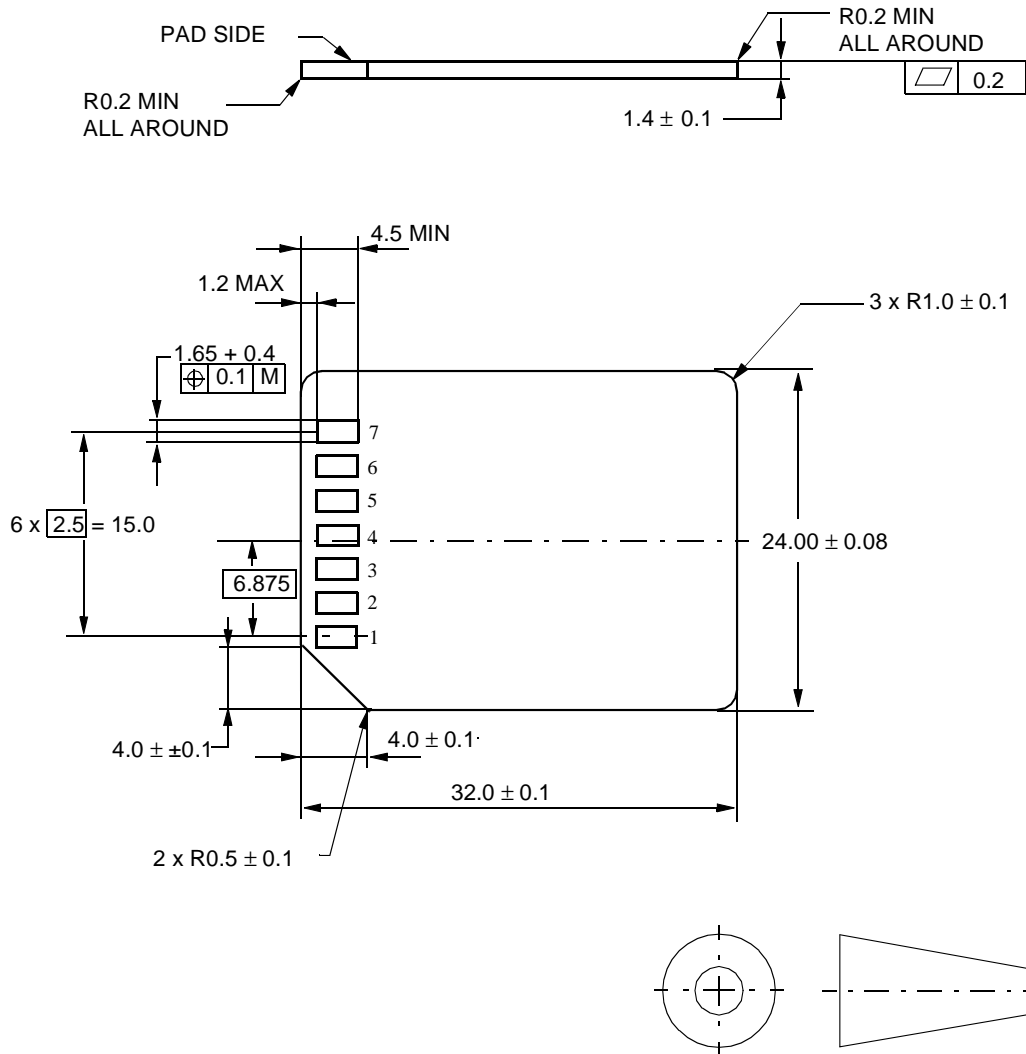• R0008 form factor: 24mm x 32mm x 1.4mm (WxLxH):

Figure 22: MultiMediaCard R008 shape

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

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.