Features

- Compatible with an Embedded ARM® Processor
- Supports MultiMediaCard Specification Version 2.2
- Supports SD Memory Card Specification Version 1.0
- Cards Clock Rate up to System Clock Divided by 2
- Embedded Power Management to Slow Down Clock Rate when Not Used
- Supports up to Sixteen Slots (through Multiplexing) – One Slot for One MultiMediaCard Bus (up to 30 Cards) or One SD Memory Card
- Support for Stream, Block and Multi-block Data Read and Write
- Compatible with PDC and PDC2 Minimizing CPU Intervention for Large Buffer Transfers
- Fully Scan Testable up to 98%
- Can Be Directly Connected to Atmel Implementation of the AMBA™ Peripheral Bus (APB)

Description

The MultiMedia Card Interface (MCI) supports the MultiMediaCard (MMC) Specification V2.2 and the SD Memory Card Specification V1.0.

The MCI includes a command register, response registers, data registers, timeout counters and error detection logic that automatically handle the transmission of commands and, when required, the reception of the associated responses and data with a limited CPU overhead. It supports stream, block and multi-block data read and write, and is compatible with PDC and PDC2 DMA channels, minimizing CPU intervention for large buffer transfers.

It can operate at a rate of up to Master Clock divided by 2 and supports the interfacing of up to 16 slots.

Each slot may be used to interface with a MultiMediaCard bus (up to 30 Cards) or with an SD Memory Card.

Only one slot can be selected at a time (slots are multiplexed). The bit SDCSEL of the MCI_SDCR register performs this selection.

The SD Memory Card communication is based on a 9-pin interface (clock, command, four data and three power lines) and the MultiMediaCard on a 7-pin interface (clock, command, one data and three power lines).

The SD Memory Card interface also supports MultiMediaCard operations; the main differences between SD and MultiMediaCards are the initialization process and the bus topology.

32-bit Embedded ASIC Core Peripheral

MultiMedia Card Interface (MCI)

Rev. 1764A–12/01

Figure 1. MultiMedia Card Interface Symbol

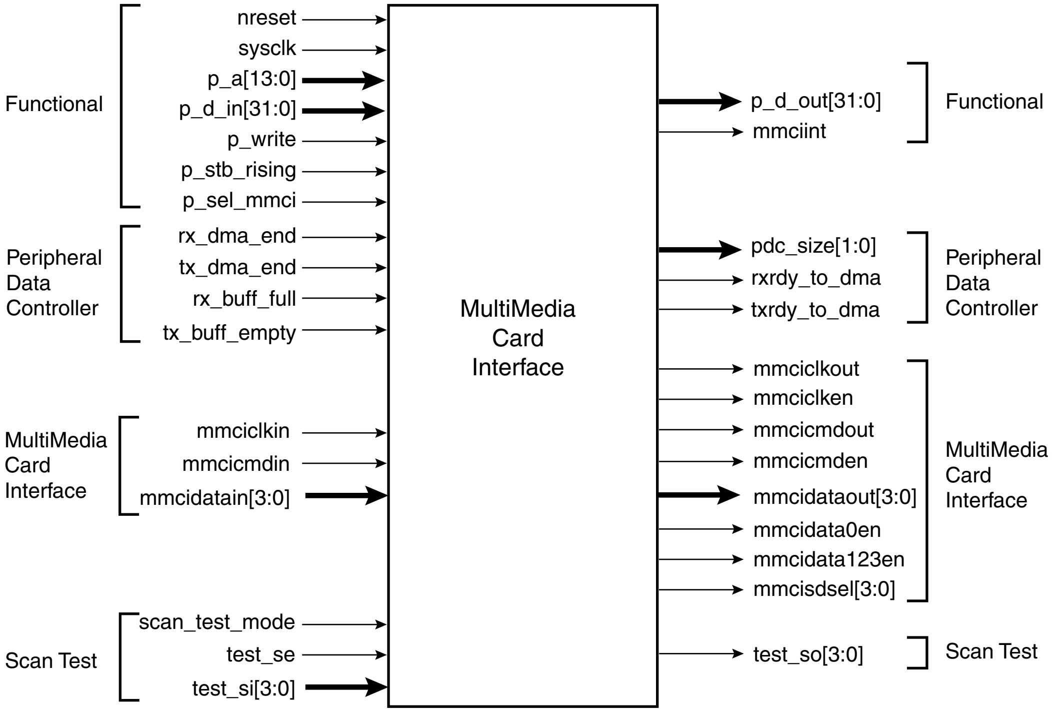


Table 1. MCI Signal Description

Signal	Description	Type	Active Level	Comments
		Functional		
nreset	System Reset	Input	Low	Asynchronous Reset
sysclk	System Clock	Input	–	System Clock for MCI internal registers
p_a[13:0]	Address Bus	Input	–	The address takes into account the two LSBs [1:0], but the MCI does not take these into account (left unconnected).
p_d_in[31:0]	Input Data Bus	Input	–	From Host (Bridge)
p_d_out[31:0]	Output Data Bus	Output	–	To Host (Bridge)
p_write	Write Enable	Input	High	From Host (Bridge)
p_stb_rising	User Interface Clock Signal	Input	–	From Host (Bridge). Clock for all DFFs controlling the configuration registers.
p_sel_mmci	Peripheral Select	Input	High	From Host (Bridge)
mmciint	Interrupt Signal to AIC	Output	High	To AIC
		PDC		
pdc_size[1:0]	Size of Transfer	Output	–	To PDC
rxrdy_to_dma	Output Signal to DMA	Output	High	Byte available in Receiver Data Register (RDR). This signal connects to the PDC.
txrdy_to_dma	Output Signal to DMA	Output	High	There are no more characters in the Transmit Data Register (TDR). This signal connects to the PDC.

Table 1. MCI Signal Description (Continued)

Signal	Description	Type	Active Level	Comments
rx_dma_end	End of Receive DMA Transfer	Input	High	Generated by PDC
tx_dma_end	End of Transmit DMA Transfer	Input	High	Generated by PDC
rx_buff_full	Input Signal from DMA Channel	Input	High	Generated by PDC
tx_buff_empty	Input Signal from DMA Channel	Input	High	Generated by PDC
MCI				
mmciclkin	MMC-SD Clock Input	Input	–	MMC or SD card clock line feedback input
mmciclkout	MMC-SD Clock Output	Output	–	MMC or SD card clock line output
mmciclken	MMC-SD Clock Enable	Output	Low	MMC or SD card clock line active low output enable
mmcicmdin	MMC-SD Command Input	Input	–	MMC or SD card command line input
mmcicmdout	MMC-SD Command Output	Output	–	MMC or SD card command line output
mmcicmden	MMC-SD Command Enable	Output	Low	MMC or SD card command line active low output enable
mmcidatain [3:0]	MMC-SD Data Input	Input	–	MMC or SD card data lines input
mmcidataout [3:0]	MMC-SD Data Output	Output	–	MMC or SD card data lines output
mmcidata0en	MMC-SD Data[0] Enable	Output	Low	MMC or SD card data[0] active low output enable
mmcidata123 en	MMC-SD Data[3:1] Enable	Output	Low	MMC or SD card data[3:1] active low output enable
mmcisdsel[3:0]	SD Card Selector	Output	–	MMC or SD card selector. If two or more SD cards are used, this output must be connected to the sdmux module.
Scan Test				
scan_test_ mode	Scan Test Mode	Input	High	Active high during scan. Must be set when running the scan vectors. During test mmciclkin is used as clock input.
test_se	Scan Test Enable	Input	High	Scan Test Enable
test_si[3:0]	Scan Test Inputs	Input	–	Scan Chain Inputs
test_so[3:0]	Scan Test Outputs	Output	–	Scan Chain Outputs

Slots and Bus Topology

Slots are generated by the SDMUX module which multiplexes/demultiplexes the different slot signals to/from the unique interface of the MCI.

The available configurations for two slots are shown in Table 2.

Table 2. MCI Slot Configurations

MCI Slot A	MCI Slot B
SD Card	SD Card
SD Card	MMC Bus
MMC Bus	SD Card
MMC Bus	MMC Bus

MultiMediaCard Bus Topology

Each MCI slot can be used as a MultiMediaCard bus.

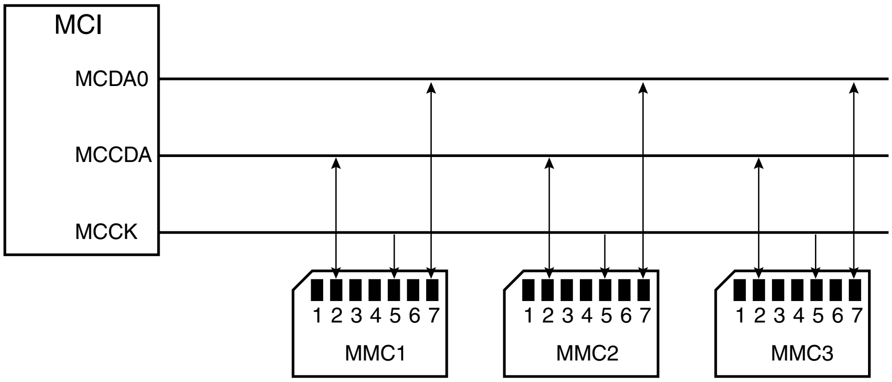
The MultiMediaCard bus has three communication lines and four supply lines, as shown in Table 3.

Table 3. MultiMediaCard Bus

Pin No.	Name	Type[(1)]	Description	MCI Pin Name
1	RSV	NC	Not connected	
2	CMD	I/O/PP/OD	Command/Response	MCCDA
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data 0	MCDA0

Note: 1. S: Power Supply, I: Input, O: Output, PP: Push-pull, OD: Open Drain, NC: Not Connected

Figure 2. MMC Bus Connections



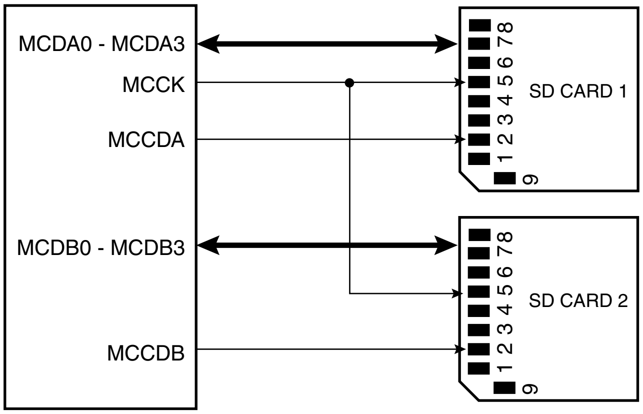
SD Memory Card Bus Topology

One SD Memory Card can be mounted on each MCI slot. The SD Memory Card includes the signals listed in Table 4.

Table 4. SD Memory Card Bus

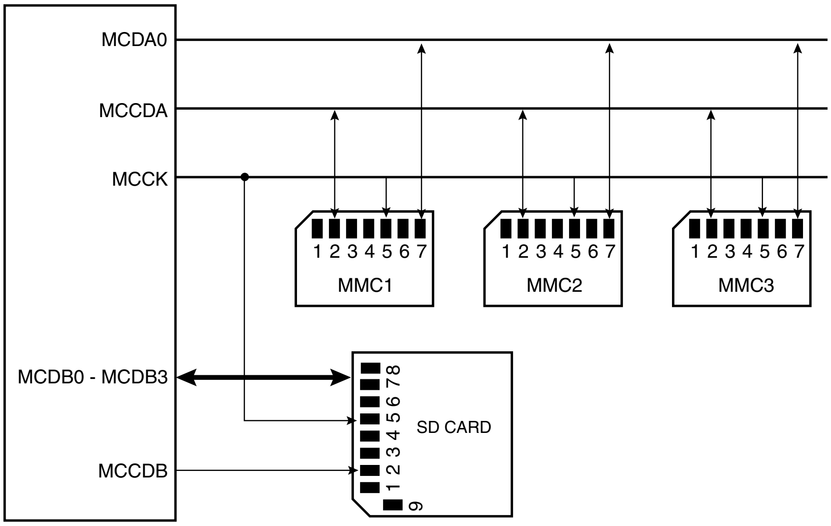
Pin No.	Name	Type	Description	MCI Pin Name
1	CD/DAT[3]	I/O/PP	Card detect/Data line bit 3	MCDA3/MCDB3
2	CMD	PP	Command/Response	MCCDA/MCCDB
3	VSS1	S	Supply voltage ground	VSS
4	VDD	S	Supply voltage	VDD
5	CLK	I	Clock	MCCK
6	VSS2	S	Supply voltage ground	VSS
7	DAT[0]	I/O/PP	Data line bit 0	MCDA0/MCDB0
8	DAT[1]	I/O/PP	Data line bit 1	MCDA1/MCBD1
9	DAT[2]	I/O/PP	Data line bit 2	MCDA2/MCDB2

Figure 3. SD Bus Connections



Mixing MultiMediaCards and SD Memory Cards

Figure 4. Mixed MMC and SD Bus Connections



MCI Configuration

After a hardware reset, the MCI clock is disabled and the user must configure the Power Management Controller before accessing the MCI user interface.

By default, the MCI pins are deselected and the user must configure the PIO Controller to assign PIOs to MCI peripheral functions. For details, refer to the PIO2 datasheet, literature number 1725.

When the MCI is configured to operate with SD memory cards, the width of the data bus can be selected in the MCI_SDCR register. Clearing the SDCBUS bit in this register means that the width is 1 bit and setting it means that the width is 4 bits. In the case of multimedia cards, only the data line 0 is used. The other data lines can be used as independent PIOs.

MMC Operations

After a power-on reset, the cards are initialized by a special message-based MultiMediaCard bus protocol. Each message is represented by one of the following tokens:

- Command: A command is a token that starts an operation. A command is sent from the host either to a single card (addressed command) or to all connected cards (broadcast command). A command is transferred serially on the CMD line.
- Response: A response is a token which is sent from an addressed card or (synchronously) from all connected cards to the host as an answer to a previously received command. A response is transferred serially on the CMD line.
- Data: Data can be transferred from the card to the host or vice versa. Data is transferred via the data line.

Card addressing is implemented using a session address assigned during the initialization phase by the bus controller to all currently connected cards. Their unique CID number identifies individual cards.

The structure of commands, responses and data blocks is described in the MultiMediaCard System Specification Version 2.2. See also Table 7 on page 8.

MultiMediaCard bus data transfers are composed of these tokens.

There are different types of operations. Addressed operations always contain a command and a response token. In addition, some operations have a data token, the others transfer their information directly within the command or response structure. In this case, no data token is present in an operation. The bits on the DAT and the CMD lines are transferred synchronous to the clock MCCK.

Two types of data transfer commands are defined:

- Sequential commands: These commands initiate a continuous data stream. They are terminated only when a stop command follows on the CMD line. This mode reduces the command overhead to an absolute minimum.
- Block-oriented commands: These commands send a data block succeeded by CRC bits. Both read and write operations allow either single or multiple block transmission. A multiple block transmission is terminated when a stop command follows on the CMD line similarly to the sequential read.

The MCI provides a set of registers to perform the entire range of MultiMediaCard operations. After reset the MCI is disabled and becomes valid after setting the MCIEN bit in the MCI_CR Control Register. The command and the response of the card are clocked out with the rising edge of the MCCK.

All the timings for MultiMedia cards are defined in the MultiMediaCard System Specification Version 2.2.

The two bus modes (open drain and push/pull) needed to process all the operations are defined in the MCI command register. The MCI_CMDR allows a command to be carried out. For example, to perform an ALL_SEND_CID command:

	Host Command					N_{ID} Cycles					CID or OCR			
CMD	S	T	Content	CRC	E	Z	******	Z	S	T	Content	Z	Z	Z

The command ALL_SEND_CID and the fields and values for the MC_CR Control Register are described in Table 5 and Table 6.

Table 5. ALL_SEND_CID Command Description

CMD Index	Type	Argument	Resp	Abbreviation	Command Description
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.

Table 6. Fields and Values for MC_CR Command Register

Field	Value
CMDNB (command number)	2 (CMD2)
RSPTYP (response type)	2 (R2: 136-bit response)
SPCMD (special command)	0 (not a special command)
OPCMD (open drain command)	1 (open drain)
MAXLAT (max latency for command to response)	0 (N_{ID} cycles ==> 5 cycles)
TRCMD (transfer command)	0 (No data transfer)
TRDIR (transfer direction)	X (available only in transfer command)
TRTYP (transfer type)	X (available only in transfer command)

The MCI Argument Register MCI_ARGR contains the argument field of the command.

The MultiMedia card allows several read/write operations (single block, stream, etc.). These operations can be done using the PDC features. If the PDCMODE bit is set in the MCI_MR then all reads and writes use the PDC. In all cases, the block length must be defined in the mode register.

MCI User Interface

Table 7. MCI Memory Map

Offset	Register Name	Register Code	Read/Write	Reset
0x00	Control Register	MCI_CR	Write	–
0x04	Mode Register	MCI_MR	Read/Write	0x0
0x08	Data Timeout Register	MCI_DTOR	Read/Write	0x0
0x0C	SD Card Register	MCI_SDCR	Read/Write	–
0x10	Argument Register	MCI_ARGR	Read/Write	0x0
0x14	Command Register	MCI_CMDR	Write	–
0x18	Reserved			
0x1C	Reserved			
0x20	Response Register[1]	MCI_RSPR	Read	0x0
0x24	Response Register[1]	MCI_RSPR	Read	0x0
0x28	Response Register[1]	MCI_RSPR	Read	0x0
0x2C	Response Register[1]	MCI_RSPR	Read	0x0
0x30	Receive Data register	MCI_RDR	Read	0x0
0x34	Transmit Data register	MCI_TDR	Write	–
0x38	Reserved			
0x3C	Reserved			
0x40	Status Register	MCI_SR	Read	0x0
0x44	Interrupt Enable Register	MCI_IER	Write	–
0x48	Interrupt Disable Register	MCI_IDR	Write	–
0x4C	Interrupt Mask Register	MCI_IMR	Read	0x0
0x50 - 0xFC	Reserved			
0x100 - 0x13F	Reserved for PDC2 Registers			

Note: 1. The response can be read by N accesses at the same MCI_RSPR register or at consecutive addresses (0x20 to 0x2C0). N depends on the size of the response.

MCI Control Register

Register Name: MCI_CR

Access Type: Write-only

Offset: 0x0

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	–	–	–	PWSDIS	PWSEN	MCIDIS	MCIEN

- **MCIEN: MultiMedia Interface Enable**

0: No effect.

1: Enables the MultiMedia Interface if MCIDIS is 0.

- **MCIDIS: MultiMedia Interface Disable**

0: No effect.

1: Disables the MultiMedia Interface.

- **PWSEN: Power Save Mode Enable**

0: No effect.

1: Enables the Power-saving mode if PWSDIS is 0.

- **PWSDIS: Power Save Mode Disable**

0: No effect.

1: Disables the Power-saving mode.

MCI Mode Register

Register Name:: MCI_MR

Access Type:: Read/Write

Offset: 0x04

31	30	29	28	27	26	25	24
BLKLEN							
23	22	21	20	19	18	17	16
BLKLEN						0	0
15	14	13	12	11	10	9	8
PDCMODE	PDCPADV	–	–	–	PWSDIV		
7	6	5	4	3	2	1	0
CLKDIV							

- **CLKDIV: Clock Divider**

MultiMedia Card Interface clock is Master Clock (MCK) divided by (2*(CLKDIV+1)).

- **PWSDIV: Power Saving Divider**

MultiMedia Card Interface clock is divided by 2 power PWSDIV when entering Power-saving mode. If PWSDIV is 0x00 then MultiMedia Card Interface clock is stopped when entering Power-saving mode.

- **PDCPADV: PDC Padding Value**

0: 0x00 value is used when padding data in PDC write transfer (non-multiple block size PDC write)

1: 0xFF value is used when padding data in PDC write transfer (non-multiple block size PDC write)

- **PDCMODE: PDC Oriented Mode**

0: Disables PDC transfer

1: Enables PDC transfer

- **BLKLEN: Data Block Length**

This field determines the size of the data block that is a multiple of four bytes. Therefore, bits 16 and 17 must be 0.

MCI Data Timeout Register

Register Name: MCI_DTOR

Access Type: Read/Write

Offset: 0x08

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–

15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–

7	6	5	4	3	2	1	0
–	DTOMUL			DTOCYC			

- **DTOCYC: Data Timeout Cycle Number**
- **DTOMUL: Data Timeout Multiplier**

These fields determine the maximum number of clock cycles that the MCI waits between 2 data block transfers. It equals (DTOCYC x Multiplier).

Multiplier is defined by the following table:

DTOMUL			Multiplier
0	0	0	1
0	0	1	16
0	1	0	128
0	1	1	256
1	0	0	1024
1	0	1	4096
1	1	0	65536
1	1	1	1048576

MCI SD Card Register

Register Name: MCI_SDCR

Access Type: Read/Write

Offset: 0x0C

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
–	–	–	–	–	–	–	–
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
–	–	–	–	–	–	–	–
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
SCDBUS	–	–	–	SCDSEL			

- **SDCSEL: SD Card Selector**

0: SD card A selected

1: SD card B selected

- **SDCBUS: SD Card Bus Width**

0: 1-bit data bus

1: 4-bit data bus

MCI Argument Register

Register Name: MCI_ARGR

Access Type: Read/Write

Offset: 0x10

31	30	29	28	27	26	25	24
ARG							
23	**22**	**21**	**20**	**19**	**18**	**17**	**16**
ARG							
15	**14**	**13**	**12**	**11**	**10**	**9**	**8**
ARG							
7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
ARG							

- **ARG: Command Argument**

MCI Command Register

Register Name: MCI_CMDR

Access Type: Write-only

Offset: 0x14

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	–	–	TRTYP		TRDIR	TRCMD	

15	14	13	12	11	10	9	8
–	–	–	MAXLAT	OPDCMD	SPCMD		

7	6	5	4	3	2	1	0
RSPTYP		CMDNB					

This register is write-protected while CMDRDY is 0 in MCI_SR. This means that the current command execution cannot be interrupted or modified.

- **CMDNB: Command Number**
- **RSPTYP: Response Type**

RSP		Response Type
0	0	No response
0	1	48-bit response
1	0	136-bit response
1	1	Reserved

- **SPCMD: Special CMD**

SPCMD			CMD
0	0	0	Not a special CMD
0	0	1	Initialization CMD
0	1	0	Synchronized CMD
0	1	1	Reserved
1	0	0	Interrupt command
1	0	1	Interrupt response

- **OPDCMD: Open Drain Command**

0: Push/pull command

1: Open drain command

- **MAXLAT: Maximum Latency for Command to Respond**

0: 5 cycles maximum latency

1: 64 cycles maximum latency

- **TRCMD: Transfer Command**

TRCMD		Transfer Type
0	0	No transfer
0	1	Start transfer
1	0	Stop transfer
1	1	Reserved

- **TRDIR: Transfer Direction**

0: Write

1: Read

- **TRTYP: Transfer Type**

TRTYP		Transfer Type
0	0	Block
0	1	Multiple Block
1	0	Stream
1	1	Reserved

MCI Response Register

Register Name: MCI_RSPR

Access Type: Read-only

Offset: 0x20-0x2C

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">RSP</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">RSP</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">RSP</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">RSP</td></tr>
</table>

- **RSP: Bit 31..0: Response**

MCI Receive Data Register

Register Name: MCI_RDR

Access Type: Read-only

Offset: 0x30

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DATA</td></tr>
</table>

- **DATA: Bit 31..0: Data to Read**

MCI Transmit Data Register

Register Name: MCI_TDR

Access Type: Write-only

Offset: 0x34

<table>
<tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td></tr>
<tr><td colspan="8">DATA</td></tr>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td colspan="8">DATA</td></tr>
</table>

- **DATA: Bit 31..0: Data to Write**

MCI Status Register

Register Name: MCI_SR

Access Type: Read-only

Offset: 0x40

31	30	29	28	27	26	25	24
UNRE	OVRE	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE
15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

- **CMDRDY: Command Ready Status**
- **RXRDY: RX Ready Status**
- **TXRDY: TX Ready Status**
- **BLKE: Data Block Transfer Ended Status**
- **DTIP: Data Transfer in Progress Status**
- **NOTBUSY: Data Line Not Busy Status**
- **ENDRX: End of RX Buffer Status**
- **ENDTX: End of TX Buffer Status**
- **RXBUFF: RX Buffer Full Status**
- **TXBUFE: TX Buffer Empty Status**
- **RINDE: Response Index Error Status**
- **RDIRE: Response Direction Error Status**
- **RCRCE: Response CRC Error Status**
- **RENDE: Response End Bit Error Status**
- **RTOE: Response Timeout Error Status**
- **DCRCE: Data CRC Error Status**
- **DTOE: Data Timeout Error Status**
- **OVRE: Overrun Status**
- **UNRE: Underrun Status**

MCI Interrupt Enable Register

Register Name: MCI_IER

Access Type: Write-only

Offset: 0x44

31	30	29	28	27	26	25	24
UNRE	OVRE	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE
15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

- **Interrupt Enable Register**

0: No effect.

1: Enables the interrupt.

See “MCI Interrupt Mask Register” on page 18 for definitions of fields.

MCI Interrupt Disable Register

Register Name: MCI_IDR

Access Type: Write-only

Offset: 0x48

31	30	29	28	27	26	25	24
UNRE	OVRE	–	–	–	–	–	–
23	22	21	20	19	18	17	16
–	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE
15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	–	–	–	–	–	–
7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

- **Interrupt Disable Register**

0: No effect.

1: Disables the interrupt.

See “MCI Interrupt Mask Register” on page 18 for definitions of fields.

MCI Interrupt Mask Register

Register Name: MCI_IMR

Access Type: Read-only

Offset: 0x4C

31	30	29	28	27	26	25	24
UNRE	OVRE	–	–	–	–	–	–

23	22	21	20	19	18	17	16
–	DTOE	DCRCE	RTOE	RENDE	RCRCE	RDIRE	RINDE

15	14	13	12	11	10	9	8
TXBUFE	RXBUFF	–	–	–	–	–	–

7	6	5	4	3	2	1	0
ENDTX	ENDRX	NOTBUSY	DTIP	BLKE	TXRDY	RXRDY	CMDRDY

This register shows which interrupt is masked.

0: Interrupt is disabled.

1: Interrupt is enabled.

- **CMDRDY: Command Ready Interrupt Mask**
- **RXRDY: RX Ready Interrupt Mask**
- **TXRDY: TX Ready Interrupt Mask**
- **BLKE: Data Block Ended Interrupt Mask**
- **DTIP: Data Transfer in Progress Interrupt Mask**
- **NOTBUSY: Data Not Busy Interrupt Mask**
- **ENDRX: End of RX Buffer Interrupt Mask**
- **ENDTX: End of TX Buffer Interrupt Mask**
- **RXBUFF: RX Buffer Full Interrupt Mask**
- **TXBUFE: TX Buffer Empty Interrupt Mask**
- **RINDE: Response Index Error Interrupt Mask**
- **RDIRE: Response Direction Error Interrupt Mask**
- **RCRCE: Response CRC Error Interrupt Mask**
- **RENDE: Response End Bit Error Interrupt Mask**
- **RTOE: Response Timeout Error Interrupt Mask**
- **DCRCE: Data CRC Error Interrupt Mask**
- **DTOE: Data Timeout Error Interrupt Mask**
- **OVRE: Overrun Interrupt Mask**
- **UNRE: Underrun Interrupt Mask**

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel SarL
Route des Arsenaux 41
Casa Postale 80
CH-1705 Fribourg
Switzerland
TEL (41) 26-426-5555
FAX (41) 26-426-5500

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Memory
Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 436-4270
FAX (408) 436-4314

Microcontrollers
Atmel Corporate
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 436-4270
FAX (408) 436-4314

Atmel Nantes
La Chantrerie
BP 70602
44306 Nantes Cedex 3, France
TEL (33) 2-40-18-18-50
FAX (33) 2-40-28-19-60

ASIC/ASSP/Smart Cards
Atmel Rousset
Zone Industrielle
13106 Rousset Cedex, France
TEL (33) 4-42-53-64-21
FAX (33) 4-42-53-62-88

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Smart Card ICs
Scottish Enterprise Technology Park
Maxwell Building
East Kilbride G75 0QR, Scotland
TEL (44) 1355-803-015
FAX (44) 1355-242-743

RF/Automotive
Atmel Heilbronn
Theresienstrasse 2
Postfach 3535
74025 Heilbronn, Germany
TEL (49) 71-31-67-0
FAX (49) 71-31-67-2340

Atmel Colorado Springs
1150 East Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Biometrics/Imaging/Hi-Rel MPU/
High Speed Converters/RF Datacom
Atmel Grenoble
Avenue de Rochepleine
BP 123
38521 Saint-Egreve Cedex, France
TEL (33) 4-76-58-30-00
FAX (33) 4-76-58-34-80

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical

Printed on recycled paper.