SDC/MMC Controller

Author: [Adam Edvardsson]
[adam@orsoc.se]

Rev. [0.1]
April 27, 2009

This page has been intentionally left blank.

Revision History

Rev.	Date	Author	Description
0.1	24/04/2009	Adam E	First Draft
0.2	27/04/2009	Adam E	Added: List of content SD Bus operation Note about the data FIFO Programing Example

Contents

1. INTRODUCTION....1
Glossary....2
2. ARCHITECTURE....3
2.1 SD controller top....4
2.2 SD CMD Master....4
2.3 SD CMD Host....4
2.4 SD (Bd) Buffer descriptor....4
2.5 SD DATA Master....4
2.6 SD Data Host....5
2.7 SD FIFO Tx/Rx Filer....5
2.8 SD Clock Divider....5
3. OPERATION....6
3.1 SD/MMC Operation....6
3.1.1 Bus protocol....6
3.1.2 Command transfer....6
3.1.3 Data transfer....6
3.2 Core Operation....7
3.2.1 Reseting the core....7
3.2.2 Setting up the core....7
3.2.3 Host Interface Operation....8
3.2.3.1 Configuration Registers....8
3.2.3.2 Sending commando....8
3.2.3.3 Buffer descriptors....8
3.2.3.4 Data block transmission....8
4. REGISTERS....9
List of Registers....9
4.1 Argument Register....10
4.2 Command setting Register....10
4.3 Status Register....10
4.4 Response Register....11
4.5 Controller settings....11
4.6 Block Size....11
4.7 Power control....11
4.8 Software Reset....12
4.9 Timeout Register....12
4.10 Normal Interrupt Status register....12
4.11 Error Interrupt Status register....12
4.12 Normal Interrupt Enable register....13
4.13 Error Interrupt Enable register....13
4.14 Capability register....13
4.15 Clock divider register....14
4.16 BD buffer status register....14
4.17 Data Interrupt status register....14
4.18 Data Interrupt enable egister....15

4.19 BD RX 15
4.20 BD TX 15

5. CLOCKS....16

6. IO PORTS....17

6.1 WISHBONE IO....17
6.2 SDC IO 18

7. PROGRAMING EXAMPLE....19

EXAMPLE 1. INITIATE CORE....19
EXAMPLE 2. SEND A COMMAND WITH NORMAL RESPONSE SIZE....19
EXAMPLE 3. SEND DATA....20

8. REFERENCES....21

1. Introduction

The "sd card controller" is SD/MMC communication controller which main focus is to provide fast and simple interface to SD/MMC cards.
The core is configurable depending on how your surrounding system looks like.

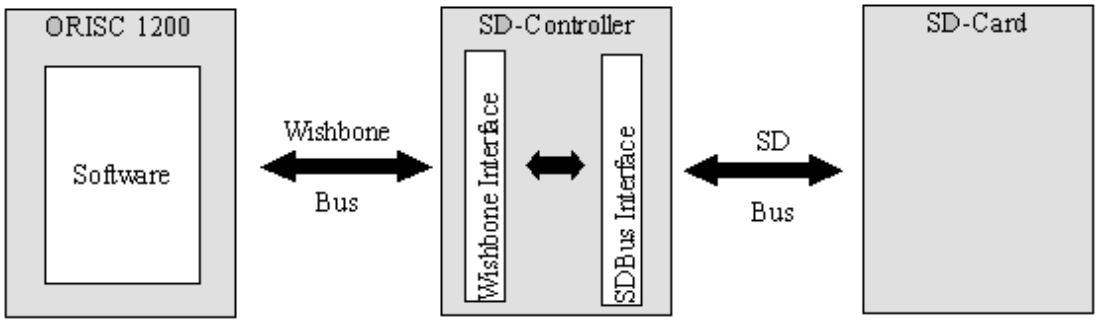

Fig 1: The controllers place in a system

Features

- 32-bit Wishbone Interface
- DMA
- Buffer Descriptor
- Compliant with SD Host Controller Spec version 2.0
- Support SD 4-bit mode
- Interrupt-on-completion of Data and Command transmission
- Write/Read FIFO with variable size
- Internal implementation of CRC16 for data lines and CRC7 for command line

1.1 Glossary

Block: The basic unit of data transfer, Its size is the number of bytes that are transferred when one block command is sent by the host. The size of a block is either programmable or fixed.

DMA: Direct Memory Access

CMD: Command**,** A command is a token that starts an operation. A command is sent from the host to a card . A command 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 lines.

Response: A response is a token that is sent from an addressed card, to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Sector: The unit that is related to the erase commands. Its size is the number of blocks that are erased in one portion. The size of a sector is fixed for each device.

SD: Secure Digital

SDHC: Secure Digital High Capacity

WB. Wishbone

2. Architecture

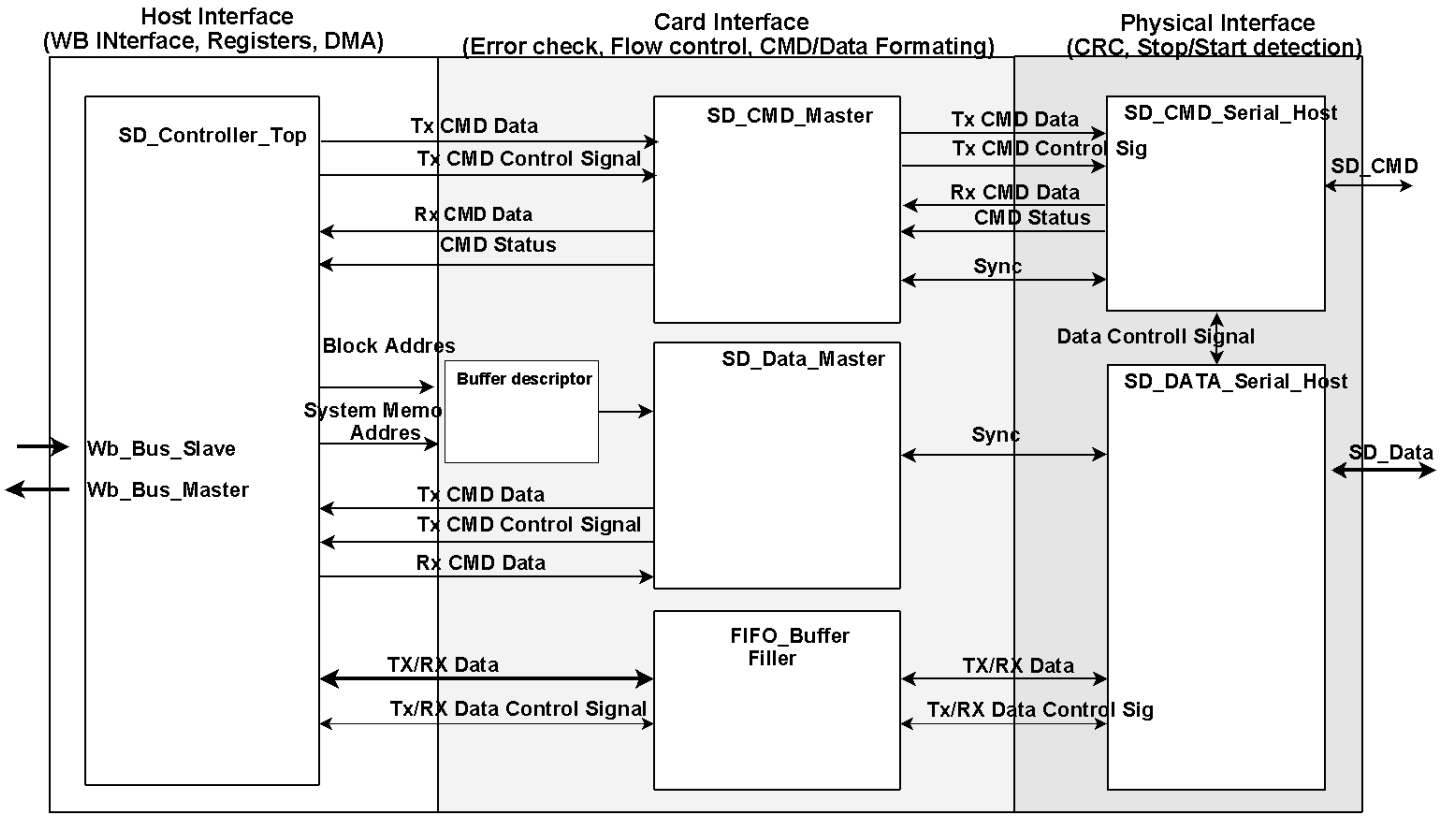

Fig 2: Simplified internal architecture

The Ethernet IP Core consists of 7 Major modules:

- Host Interface
- BD structure
- CMD Master
- CMD Host
- Data Master
- Data Host
- FIFO buffer filler

2.1 SD controller top

This module is the interface between the SDC Core and the bus . Two WB interfaces (slave and master) are used for this. The internal registers and buffer descriptors (BD) are all accessed through the same
WB Slave Interface. The master interface is used for the internal DMA to fetch and store data to and from an external memory. The module contains the setting and status register accessible by user from the WB slave, and the required logic to access this.

2.2 SD CMD Master

The SD CMD Master module synchronize the communication from the host interface with the physical interface . perform has three main tasks:
•Read a set of register from the user accessible register in the SD Controller Top to compose a 40 bit command messages to pass to the SD CMD
•Read response messages from the SD CMD Host and forward it to the user accessible register in the SD Controller Top module.
•Keep track of the status of the CMD Host module.

2.3 SD CMD Host

This module is the interface towards physical SD/MMC cards command pin. This module takes care of the physical sending and receiving of the messages, adding start bits, stop bits and CRC checksum.

2.4 SD (Bd) Buffer descriptor

The transmission and the reception processes are based on the descriptors. Two sequential wrings to this module is required to create one buffer descriptor. First the source address (Memory location) of the data is written then the card block address is written.

2.5 SD DATA Master

Starts to check if there are any new BD thats need to be processed if so the module generate a command and set up the DMA to read/write to correct address. If the command line is free the module send the command and wait fore response. If response is valid the module starts the DMA if not valid the CMD is resent again.

During transmission the module keep track for FIFO buffet underflow or overflows, when the transmission is completed it check for valid CRC. If anything goes wrong during a transmission a stop command is sent and the module try to restart the transmission n times before giving up.

2.6 SD Data Host

This module is the interface towards physical SD card device Data port. The interface consist of only 5 signals, one clock SDCLK, and the 1-4 bit bi-direction Data signal DAT.

The module perform the following actions.
• Synchronized request for write and read data and .
• Adding a CRC-16 checksum on sent data and check for correct CRC-16 on received commands.

2.7 SD FIFO Tx/Rx Filer

This module works as the DMA it manager the receive and transceiver FIFO buffer for the data stream. It keeps track of the status of the FIFO:s if somethings goes wrong, like full receiver FIFO or empty transfer buffer it signals this.

2.8 SD Clock Divider

Divide the input clock with 2 4 6 etc..

3. Operation

This section describes the SDC IP Core operation.

3.1 SD/MMC Operation

The SD/MMC cards bus includes the following,a serial data line (DAT0-DAT3), one command line (CMD) and a clock line (CLK). All signals are operating in push pull mode, all signals except CLK shall have a pull up resistor of a recommended size of 10-100kΩ

3.1.1 Bus protocol

Communication over the SD bus is based on command and data bit streams that are initiated by a start bit and terminated by a stop bit. A command is used start an operation in the card, most command gives a response token as reply.

3.1.2 Command transfer

When the bus is free/idle, meaning command line is high, an command transfer can be initiated by sending a start signal. A start signal, usually referred to as the S-bit, which is defined as a high-to-low transition of the CMD line. The command bit is sent in a MSB to LSB fashion.

3.1.3 Data transfer

To start a data transmission first a command request for data has to be sent to the card. A data then transmission starts with a S-bit on DAT0 line. The data is sent in a lowest byte first, highest byte last manner, with MSB to LSB manner for each byte.

(For further details see section 3.6 in " Physical Layer Simplified Specification Version 2.00")

3.2 Core Operation

Before starting to build the core the SD_Defines.v should be setup properly. The available options that can be modified are.:

Name	Valid Values	Description
BIG_ENDIAN		Big Endian System
LITLE_ENDIAN		Little Endian System
SIM		To ease up simulation
SYN		For Synthesizing
IRQ_ENABLE		Three extra Interrupt pin will be added
ACTEL		Get some predefine settings
BD_WIDTH	<= 8	2nlog2 BD-SIZE
BD_SIZE	<= 255	Size of the BD buffer
RAM_MEM_WIDTH_16		Width of block Ram == 16
RAM_MEM_WIDTH 16	16,32	Width of block Ram
RESEND_MAX_CNT	<255	How many retry to resend data
MEM_OFFSET		Memory address offset between 2 word
RESET_CLK_DIV	<255	In clock divider
SD_CLK_BUS_CLK		Use the same clock as the WB-Bus
SD_CLK_SEP		Use sd_clk_i_pad as SD CLK
SD_CLK_STATIC		SD CLK = IN clock
SD_CLK_DYNAMIC		SD CLK = CLK from ClkDivider
BLOCK_SIZE	512	Block Size
SD_BUS_WIDTH_4		Only support for 4 Bit
SD_BUS_W	4	Only support for 4 Bit
FIFO_RX_MEM_DEPTH		Width of RX Fifo
FIFO_RX_MEM_ADR_SIZE		FIFO_RX_MEM_DEPTH 2nlog2 +1
FIFO_TX_MEM_DEPTH		Width of TX Fifo
FIFO_TX_MEM_ADR_SIZE		FIFO_TX_MEM_DEPTH 2nlog2 +1

Table 1: Core define options

Note that the Rx and Tx FIFO is not intended to be implemented as RAM-block and therefore require a lots of logic if setting to high. If FIFO overflow or underflow a option is instead to lower the clock speed to the card or or enchant the performance of the system around to not use the memory as much leaving more free bus access cycler to the SD core.

3.2.1 Reseting the core

The RST_I signal is used for resetting all modules. This can also be done by setting the SRST bit in the Software reset register to 1 .

3.2.2 Setting up the core

1. Reset the core
2. Set the timeout register

3. Assert Software reset

3.2.3 Host Interface Operation

The host interface connects the IP Core to the rest of the system (RISC, memory) via the WISHBONE bus. The WISHBONE serves to access the configuration registers and the memory. Currently, only DMA transfers are supported for transferring the data from/to the memory

3.2.3.1 Configuration Registers

The function of the configuration registers is transparent and can be easily understood by reading the Registers section (Chapter 4).

3.2.3.2 Sending commando

The sending of a command to the SDC/MMC card is performed in two step. First the command index and transmission settings for the command to be, sent is written into the Command setting Register. Next the commands argument bits of the command is be written to the argument register, which then initiate the transfer. Upon response bit 0 in the Normal interrupt status register is set to 1 and the response is available in response register. If any of the requested error check fails will this be visible in the Error interrupt status register.

3.2.3.3 Buffer descriptors

The transmission and the reception processes are based on the descriptors. The Transmit Descriptors (Tx) are used for transmission while the Receive Descriptors (RxD) are used for reception. The buffer descriptors are 64 bits long. The first 32 bits contain the pointer to the associated buffer (where data is stored) while the last 32 bits contain the card block address to read or write from. The core has a internal ram that can store up to 255 Tx and Rx BD.

3.2.3.4 Data block transmission

To transmit a block of data, the RISC has to perform several steps. First make sure the card is initiated correctly and is ready for data with the block size of 512 byte with all 4 data bits enabled. If interrupt is used, those associated with the command line should be disabled to not receive unnecessary interrupts when the data module use the CMD line. Enabling interrupts generated by the data module should instead be activated.

Then it has to check the BD status register to see if there are any free BD. If so it store what to be sent in the memory after that it writes the start address of the stored data to the the TX Buffer descriptor register and the destination block address.

The core continuously reads the first BD, where it reads the pointer to the memory storing the associated data and starts then reading data to the internal FIFO. At the end of the transmission, the transmit status is written to the data interrupt status register and interrupt might be generated (when enabled).
The next descriptor is then loaded if more is qued up, and the process starts all over again.

4. Registers

List of Registers

Name	Address	Width	Access	Description
Argument	0x00	32	RW	Command Argument Reg
Command Setting	0x04	16	RW	Command Setting Reg
Card Status	0x08	16	R	Card Status Reg
Response	0x0c	32	R	Command Response
Controller Setting	0x1c	16	R	Controller Setting
Block Size	0x20	16	R	Block Size Reg
Power Control	0x24	8	R	Power Control Reg
Software reset	0x28	8	RW	Software reset Reg
Timeout	0x2c	16	RW	Timeout Reg
Normal Int Status	0x30	16	RW	Normal Interrupt Status Reg
Error Int Status	0x34	16	RW	Error Interrupt Status Reg
Normal Int Enable	0x38	16	RW	Normal Interrupt Enable
Error Int Enable	0x3c	16	RW	Error Interrupt Enable Reg
Capability	0x48	16	R	Capability Reg
Clock Divider	0x4c	8	RW	Clock Divider Reg
BD buffer Status	0x50	16	RW	BD Status Reg
Dat Int Status	0x54	16	RW	Data Interrupt Status Reg
Dat Int Enable	0x58	16	RW	Data Interrupt Enable Reg
BD RX	0x60	64	W	BD RX
BD TX	0x80	64	W	BD TX

Table 2: List of registers

4.1 Argument Register

Bit #	Access	Description
[31:0]	W	CMDA - Command Argument Command Data, when writing to this register the transmission st arts

Table 3: Argument Register

Reset Value:: 0000h

4.2 Command setting Register

Bit #	Access	Description
[15:14]		Reserved
[13:8]	RW	CMDI - Command Index Index of the next command
[7:6]	RW	CMDW – Command Word Select Word to read when response is > 48 Bit
5		Reserved
4	RW	CICE - Command index check 0 : Do not perform index check on response CMD 1 : Perform index check on response CMD
3	RW	CIRC - Command CRC check 0 : Do not perform CRC check on response CMD 1 : Perform CRC check on response CMD
2		Reserved
[1:0]	RW	RTS – Response type 0: No response 01: Response length 136 10: Response length 48 11: Response length 48

Table 4: Command Setting Register

Reset Value: 0000h

4.3 Status Register

Bit #	Access	Description
[15:12]	R	CST – CMD Host Serial Status 0 : Reset 1 : Write Only sate 2: Write to Read State 3: Delay after write to read 4: Delay after write only state 5: Read CMD

Bit #	Access	Description
		6: Delay after read
[11:1]		Reserved
0	R	CICMD – Command Inhibit 0 : Busy 1 : Ready

Table 5: Status Register

Reset Value: ***0001h***

4.4 Response Register

Bit #	Access	Description
[31:0]	R	CRSP – Command Response Response of last command

Table 6: Response Register

Reset Value:0000h

4.5 Controller settings

Bit #	Access	Description
[15:0]		Not in use

Table 7: Controller settings

Reset Value: 0000h

4.6 Block Size

Bit #	Access	Description
[15:12]		Reserved
[11:0]	R	BS - Block Size Currently Hard coded to 512. Value has no effect on the operation

Table 8: Block Size

Reset Value: 0200h

4.7 Power control

Bit #	Access	Description
[7:3]		Reserved
[3:0]	R	SDBP – SD Bus power Voltage provided to bus 111 : 3.3 V 110 : 3 V

Bit #	Access	Description
		101 : 1.8 V

Table 9: Power control register

Reset Value: 0007h

4.8 Software Reset

Bit #	Access	Description
[7:1]		Reserved
[0]	R	SRST – Software reset 0: 1: Reset the hardware

Table 10: Software Reset

Reset Value: 0000h

4.9 Timeout Register

Bit #	Access	Description
[15:0]	RW	CTO – Command timeout Time before a timeout signal is generated when sending, counted with the system clock.

Table 11: Timeout Register

Reset Value: 0000h

4.10 Normal Interrupt Status Reg

Writing any value to this register reset it 0000h

Bit #	Access	Description
15	RW	EI – Error interrupt If any of the bits in the *Error Interrupt Status* register are set, then this bit is
[14:1]		Reserved
0	RW	CC – Command Complete This bit is set when get the end bit of the command response.

Table 12: Normal Interrupt Status Reg

Reset Value: 0000h

4.11 Error Interrupt Status Reg

Writing any value to this register reset it 0000h

Bit #	Access	Description
[15:4]		Reserved
3	RW	CIE – Command index error

Bit #	Access	Description
		This bit is set if a Command Index error occurs in the command response.
2		Reserved
1	RW	CCRC – Command CRC error This bit is set when a command CRC check fail
0	RW	CTE – Command Timeout This bit is set when a command sequence timeout occur

Table 13: Normal Interrupt Status Reg

Reset Value: 0000h

4.12 Normal Interrupt Enable Reg

Bit #	Access	Description
15	RW	EEI – Enable Error interrupt 1 : Enable interrupt generation on EI 0 : Disable interrupt generation on EI
[14:1]		Reserved
0	RW	ECC – Enable Command Complete 1 : Enable interrupt generation on ECC 0 : Disable interrupt generation on ECC

Table 14: Normal Interrupt Enable

Reset Value: 0000h

4.13 Error Interrupt Enable Reg

Bit #	Access	Description
[15:4]		Reserved
3	RW	ECIE – Command index error 1 : Enable interrupt generation on CIE 0 : Disable interrupt generation on CIE
2		Reserved
1	RW	ECCRC – Command CRC error 1 : Enable interrupt generation on CCRC 0 : Disable interrupt generation on CCRC
0	RW	ECTE – Command Timeout 1 : Enable interrupt generation on CTE 0 : Disable interrupt generation on CTE

Table 15: Normal Interrupt Status Reg

Reset Value: 0000h

4.14 Capability register

Bit #	Access	Description
[15:0]		Reserved

Table 16: Capability Register

Reset Value: 0000h

4.15 Clock divider register

Bit #	Access	Description
[7:0]		CLKD – Clock Divider Divide the modules SD input clock 0: Divided by 2 1: Divided by 4 2: Divided by 6

Table 17: Clock divider Register

Reset Value: 0000h

4.16 BD buffer status register

Bit #	Access	Description
[15:8]	R	FBRX – Free RX Buffer Descriptors NO free receiving buffer Descriptors
[7:0]	R	FBTX – Free TX Buffer Descriptors NO free transmission buffer Descriptors

Table 18: BD Status Reg

Reset Value: 0404h

4.17 Data Interrupt status register

Bit #	Access	Description
[7:6]		Reserved
5	RW	TRE – Transmission Error 1 : CRC check failed during transmission 0 :
4	RW	CMDE – Command error 1 : Error in the command response 0 :
2	RW	FIFOE - FIFO error 1 : FIFO underflow/overflow 0 :
1	RW	MRC – Max Retry Attempts reach 1 : Unable to send after N attempts 0 :
0	RW	TRS – Transmission successful 1 : One data block has been sent/received 0 :

Table 19: Data Interrupt Status Reg

Reset Value: 0000h

4.18 Data Interrupt enable reg

Bit #	Access	Description
[7:6]		Reserved
5	RW	ETRE – Transmission Error 1 : Enable interrupt generation on TRE 0 : Disable interrupt generation on TRE
4	RW	ECMDE – Command error 1 : Enable interrupt generation on CMDE 0 : Disable interrupt generation on CMDE
2	RW	EFIFOE - FIFO error 1 : Enable interrupt generation on FIFOE 0 : Disable interrupt generation on FIFOE
1	RW	EMRC – Max Retry Attempts reach 1 : Enable interrupt generation on MRC 0 : Disable interrupt generation on MRC
0	RW	ETRS – Transmission successful 1 : Enable interrupt generation on TRS 0 : Disable interrupt generation on TRS

Table 20: Data Interrupt Status Reg

Reset Value: 0000h

4.19 BD RX

Writing any value to this register reset it 0000h

Bit #	Access	Description
[63:32]	W	Memory location were the data should be stored
[31:0]	W	Block address to read from

Table 21: BD RX

Reset Value: 0000h

4.20 BD TX

Writing any value to this register reset it 0000h

Bit #	Access	Description
[63:32]	W	Memory location were the data should be read from
[31:0]	W	Block address to write to

Table 22: BD RX

Reset Value: 0000h

5. Clocks

List of Registers

Name	Source	Rates (MHz)			Remarks	Description
		Max	Min	Resolution		
sd_clk_i_pad	Input Pad	-	0.8	0		If external clock is defined
wb_clk_I	PLL	200	-	-		System clock.
sd_clk_o	Internal generated	25	0.8	-		Clock SD and module use

Table 23: List of clocks

6 IO Ports

6.1 Wishbone IO

Port	Width	Direction	Description
wb_clk_i	1	Input	Slave WISHBONE Clock Input
wb_rst_i	1	Input	Slave WISHBONE Reset Input
wb_sel_i	4	Input	Slave WISHBONE Select Inputs
wb_dat_i	32	Input	Slave WISHBONE Data Inputs
wb_dat_o	32	Output	Slave WISHBONE Data Output
wb_adr_i	8	Input	Slave WISHBONE Address Input
wb_we_i	1	Input	Slave WISHBONE Write Enable
wb_cyc_i	1	Input	Slave WISHBONE Cycle
wb_stb_i	1	Input	Slave WISHBONE Strobe
wb_ack_o,	1	Output	Slave WISHBONE Acknowledgment
m_wb_adr_o	32	Output	Master WISHBONE Address
m_wb_sel_o	1	Output	Master WISHBONE Select
m_wb_we_o	1	Output	Master WISHBONE Write Enable
m_wb_dat_o	32	Output	Master WISHBONE Data Output
m_wb_dat_i	31	Input	Master WISHBONE Data Input
m_wb_cyc_o	1	Output	Master WISHBONE Cycle
m_wb_ack_i	1	Input	Master WISHBONE Acknowledgment Input
m_wb_cti_o	1	Output	Master WISHBONE Cti
m_wb_bte_o	1	Output	Master WISHBONE Bte

Table 24: List of IO ports

6.2 SDC IO

Port	Width	Direction	Description
sd_cmd_dat_i	1	Input	SDC/MMC CMD Input
sd_cmd_out_o,	1	Output	SDC/MMC CMD Output
sd_cmd_oe_o	1	Output	SDC/MMC CMD Output enable
sd_dat_dat_i	4	Input	SDC/MMC Data Input
sd_dat_out_o	4	Output	SDC/MMC Data Output
sd_dat_oe_o	1	Output	SDC/MMC Data Output enable
sd_clk_o_pad	1	Output	SDC/MMC CLK Output
sd_clk_i_pad	1	Input	SDCLK input
int_a,	1	Output	Interrupt A Output
int_b	1	Output	Interrupt B Output
int_c	1	Output	Interrupt C Output

Table 25: List SDC IO ports

7

Programing example

Example 1. Initiate core

Set up the core, to have a command response timeout off 2500 bus cycler and with a SD clock 6 times lower then the provided one.

Sequence:

1. Set timeout value (0x9c4)
2. Disable core
3. Set clock divider (0x2)
4. Enable core

Commands:

1) Write 0x9c4 to Timeout register
2) Write 0x1 to Software reset
3) Write 0x2 to Clock divider register
4) Write 0x0 to Software reset

Example 2. Send a command with normal response size

Send a command (i.e CMD 8) with response size of 48 bits, with CRC check and command index error check enabled on the incoming response message.

Sequence:

1. Setup command register
2. Setup argument register (1AA) 2.7-3.6V + Check pattern
3. Wait for replay (or error)

4. Check response

Commands:

1) Write 0x8 to the CMDI slice, 0x0 to word select 0x1 to the CICE and CIRC bit and 0x2 to the RTS = 0x81A
2) Write 0x1AA to Argument register
3) Read the Normal interupt register. If CC bit is 1 the command the sequence is completed if EI bit is set a error ouccred. If EI is not set goto step 5.
4) Read Error interup register if you wanna see what kind of error occured.
5) Read response register

Example 3. Send data

Prereq: The card has to be in data transfer state.
Send a block of data to the SD card.

Sequence:

1. Check how many free TxBD there are avaible
2. Write the memory adress
3. Write the block destination
4. Wait for transmission done

Commands:

1) Read the BD buffer Status check the FBTX bits, wait until >0
2) Write the memory location of the data to the BD TX register
3) Write the blockaddres for the destination of the data to the BD TX register
4) Either wait for Data Interrupt Status Reg TRS bit to turn 1 or keep track on the number of free BD and then check for errors.

8

References

SD Specifications Part 1
Physical Layer Simplified Specification
Version 2.00
September 25, 2006

SanDisk Secure Digital Card
Product Manual
Version 1.9
Document No. 80-13-00169
December 2003

Wishbone specification b3