# Data Sheet For SD Memory Slave Controller

# DOCUMENT REVISION HISTORY

| Revision | Date | Change Description | Author |
|---|---|---|---|
| 1.0 | 31st May'11 | Initial Version | Preeti.N |
| REL 1.0 | 16th Aug'12 | Removed implementation results | VC |
| | | | |
| | | | |
| | | | |
| | | | |

# Table of Contents

# List Of Figures

# List Of Tables

# 1 Introduction

## 1.1 Purpose

This document describes the Technical Specification of the SD Memory Slave Controller. It includes the overall architectural description, detailed functional specifications and interface definitions for the SD Memory Slave Controller.

## 1.2 Features

Following are the main features of the SD Memory Slave Controller:

- Compliant with SD Physical Specification Version 2.00
- Supports 1-bit and 4-bit SD Mode
- Supports Standard and High Capacity operations
- Supports Default and High Speed Modes of operation
- Supports all mandatory slave registers set
- CID Register fields are configurable through header file
- Supports only Standard command set
- Supports all mandatory SD Command Classes
- Class 0(Basic Commands)
    - CMD0
    - CMD2
    - CMD3
    - CMD6
    - CMD7
    - CMD8
    - CMD9
    - CMD10
    - CMD12
    - CMD13
    - CMD15
- Class 2(Block Read Commands)
    - CMD16
    - CMD17
    - CMD18
- Class 4(Block Write Commands)
    - CMD16
    - CMD24
    - CMD25
- Class 5(Erase Commands)
    - CMD32
    - CMD33
    - CMD38
- Class 8(Application Specific)
    - CMD55

  - ACMD6
  - ACMD13
  - ACMD22
  - ACMD23
  - ACMD41
  - ACMD42
  - ACMD51
- Class 10(Switch Commands)
  - CMD6
- CRC7 checking/generation for Command/Response
- CRC16 checking/generation for Data transfer
- Support Maximum block length of 512 bytes
- Supports Single and Multiple block read and write data transfer
- Supports Partial and Misalign Block length option
- SD Memory only implementation
- IP provides simple and general-purpose 8-bit interface to user application
- Combo card features are not supported
- SPI Mode is not supported
- Card Lock/Unlock Operation is not supported.

## 1.3 Acronyms and Abbreviations

**Table 1: Acronyms & Abbreviations**

| Term | Meaning |
|---|---|
| CRC | Cyclic Redundancy Check |
| CMD | Command |
| FIFO | First In First Out |
| FSM | Finite State Machine |
| FPGA | Field Programmable Gate Array |
| SD | Secure Digital |
| SPI | Serial Peripheral Interface |
| TBD | To Be Done |
| WDT | Watch Dog Timer |

# 2 SD Memory Slave Controller

## 2.1 Block Diagram

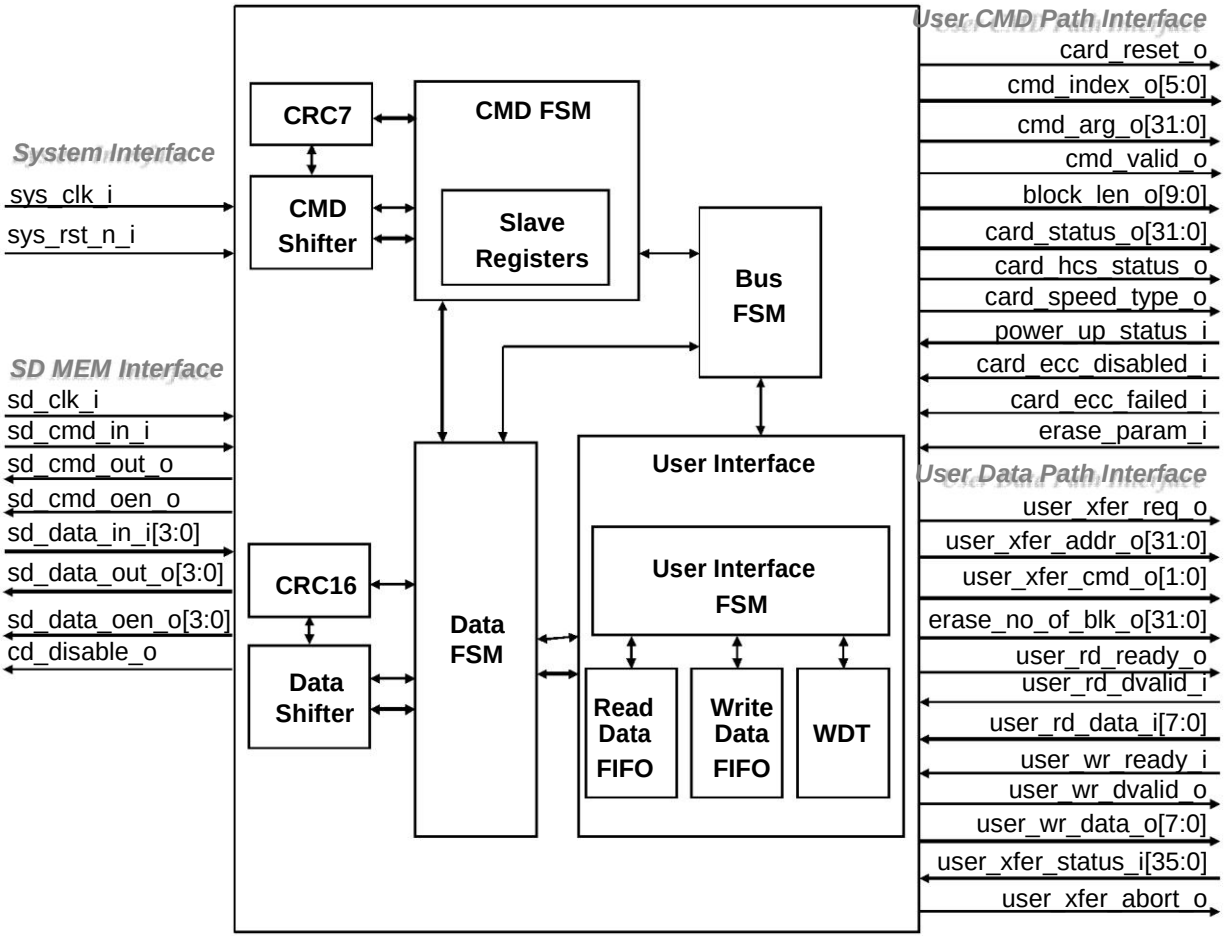


**Figure 1: SD Memory Slave Controller Block Diagram**

## 2.2 Description

### 2.2.1 Command FSM

The command path state machine controls the reception of Command and transmission of Response. Depending on the mode of operation, it generates appropriate control signals to the shifter and CRC7 block. It also parses the command received and generates suitable response output according to rules described in Specification, based on inputs from CRC7 block and Slave FSM. For every command, errors will be updated in the response flags of corresponding response.

### 2.2.2 Command Shifter

The Command Transmitter and Response Receiver block has a 128-bit “Shift Register”, it is used to receive and shift in the Command bits and transmit out the Response bits towards the Host. After it receives the complete command, it latches the value and passes it to the Command FSM for further processing. The incoming serial Command data is also passed to the CRC7 block to check for CRC. The serial Response data out is also passed to the CRC7 block to calculate CRC value. It appends the 7-bit CRC during response transmission.

### 2.2.3 CRC7

The CRC7 module generates CRC Checksum for the 48-bit response generated by the SD Response Block. It calculates the CRC checksum for the start bit, transmission bit, command index and command argument (or card status). The CRC value computed crc7_calc_o is then shifted out after the response data when the crc7_shift_i is asserted by the SD Command FSM.

### 2.2.4 Slave FSM

The SD slave Bus State Machine controls the overall operation of the SD memory controller operation. It defines the Bus States and their relations to SD Commands. The accepted commands indicated along with the individual states apply to SD Mode of operation.

### 2.2.5 Data FSM

The Data FSM block controls the process of data being transmitted from the SD card to host during read operations or data being received by the SD card from the host during write operations.

### 2.2.6 Data Shifter

The data Transmitter and Response Receiver block has a 8-bit “Shift Register”. For data path, since SD operates in 4-bit mode or 1-bit mode, during transmission the data stored in the shifter is shifted out on bit by bit or 4-bit by 4-bit on data line/s. Similarly during reception, the data comes to the shifter bit by bit or 4-bit by 4-bit is shifted in and accumulate on it.

### 2.2.7 CRC16

During data transmission, the CRC generator calculates CRC checksum for all data bits in a single block. When the Data FSM Module completes transmission of serial data it enables serial shifting of CRC16 value by asserting crc16_shift_i input. The same module acts as a CRC checker during data reception. CRC is calculated over the received data and CRC to result in a value of zero. For data in 4-bit mode, CRC16 is calculated separately for each data line.

### 2.2.8 User Interface

The user interface block interfaces SD slave controller and user application through a 8-bit general purpose user interface for data transfer. The user application has a separate FIFO for read and write data buffering. A WDT is implemented in the user logic to implement system timeout logic for user interface transaction.

## 2.3 I/O Signal Description

**Table 2: System Interface IO Signal Description**

| Signal | Type | Width | Description |
|---|---|---|---|
| sys_rst_n_i | I | 1 | System Reset.<br>Active Low Asynchronous reset input. |
| sys_clk_i | I | 1 | System Clock.<br>Clock input to the FPGA This clock is used for the user logic. |

**Table 3: SD Memory Interface IO Signal Description**

| Signal | Type | Width | Description |
|---|---|---|---|
| sd_clk_i | I | 1 | SDIO Bus Clock input. |
| sd_cmd_in_i | I | 1 | SDIO Command input |
| sd_cmd_out_o | O | 1 | SDIO Response output |
| sd_cmd_oen_o | O | 1 | SDIO Output enable |
| sd_data_in_i | I | 4 | SDIO Data input |
| sd_data_out_o | O | 4 | SDIO Data output |
| sd_data_oen_o | O | 4 | SDIO Data enable output |
| cd_disable_o | O | 1 | Connect/ Disconnect the 10K-90K ohm pull-up resistor The pull-up may be used for card detection.<br>0 → Connect<br>1 → Disconnect |

**Table 4: User Application Data Path Interface IO Signal Description**

| Signal | Type | Width | Description |
|---|---|---|---|

| Signal | Type | Width | Description |
|---|---|---|---|
| user_xfer_req_o | O | 1 | User Application Request signal.<br>For every new command user request is asserted for one clock cycle. |
| user_xfer_addr_o[31:0] | O | 32 | User Application Address Output<br>Standard Capacity Card -> It is in byte unit.<br>High Capacity Card -> It is block (512 Byte) unit |
| user_xfer_cmd_o[1:0] | O | 2 | User Application Command Output<br>00 -> Erase Command<br>01 -> Erase and then Write Command<br>10 -> Write Command<br>11 -> Read Command |
| erase_no_of_blk_o[31:0] | O | 32 | Indicates no. of Block to be erased.<br>This field is valid when user_xfer_cmd_o is “00” or “01” |
| user_rd_ready_o | O | 1 | User Application read ready signal.<br>This signal indicates SD slave controller is ready to receive the data. |
| user_rd_dvalid_i | I | 1 | User Application read data valid signal.<br>When valid data is available in user_rd_data_i bus, this is signal will be high. |
| user_rd_data_i[7:0] | I | 8 | User Application read data bus input<br>SD Slave controller latch the user read data when user_rd_ready_o signal and user_rd_dvalid_i signal is high. |
| user_wr_ready_i | I | 1 | User Application Write ready input signal<br>This signal indicates the user application is ready to receive the data. |
| user_wr_dvalid_o | O | 1 | User Application Write data valid output signal.<br>SD Slave controller drive high when valid write data available on user_wr_data_o bus. |
| user_wr_data_o[7:0] | O | 8 | User Application Write data output bus. |

| Signal | Type | Width | Description |
|---|---|---|---|
| user_txfer_status_i[35:0] | I | 36 | User transfer status input<br><br>[35] -> Erase successful<br>[34] -> Erase failed<br>[33] -> Write successful<br>[32] -> Write failed<br>[31:0] -> No. of blocks written successfully |
| user_xfer_abort_o | O | 1 | User transfer abort signal.<br><br>When user application is responding for long time then slave controller will abort the current transfer cycle. |

**Table 5: User Application Command Path Interface IO Signal Description**

| Signal | Type | Width | Description |
|---|---|---|---|
| card_reset_o | O | 1 | Active high card reset output signal |
| cmd_index_o[5:0] | O | 6 | Received host command output |
| cmd_arg_o[31:0] | O | 32 | Received host command argument output |
| cmd_valid_o | O | 1 | Received host command valid output |
| block_len_o[9:0] | O | 10 | SD Block length output |
| card_status_o[31:0] | O | 32 | SD Slave card status register output |
| card_hcs_status_o | O | 1 | SD Slave card capacity type output<br><br>0 – SD Slave configured as standard Capacity Card<br>1 - SD Slave configured as high Capacity Card |
| card_speed_type_o | O | 1 | SD Slave card speed type output<br><br>0 – SD Slave operating in default speed (25MHz)<br>1 – SD Slave operating in high speed mode (50 MHz) |
| power_up_status_i | I | 1 | Power Up Status input |

| Signal | Type | Width | Description |
|---|---|---|---|
| card_ecc_disabled_i | I | 1 | Card ECC Disable Status input |
| card_ecc_failed_i | I | 1 | Card ECC Failed Status input |
| erase_param_i | I | 1 | Invalid Selection of write blocks for erase occured |

# 3 Timing Waveforms

Below waveforms shows the timing diagram of SD Memory Slave Controller.

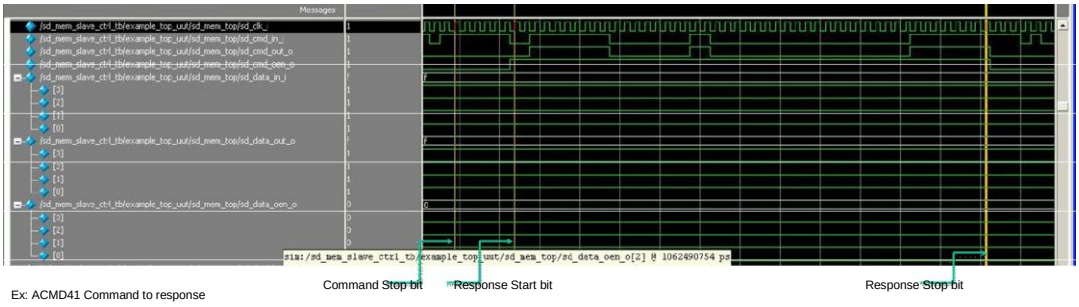


**Figure 2: Command to Response Waveform**

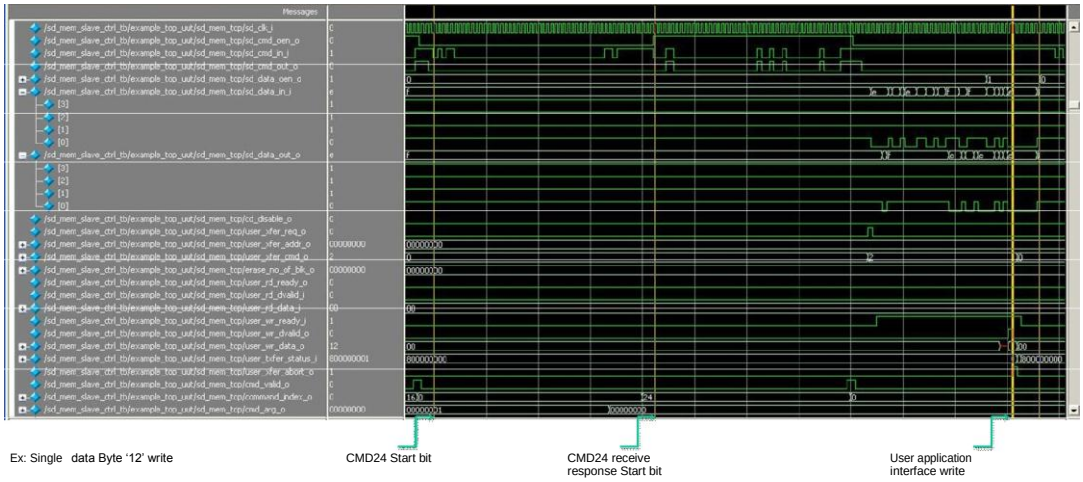


**Figure 3: Single Block Write 1-bit Mode Waveform**

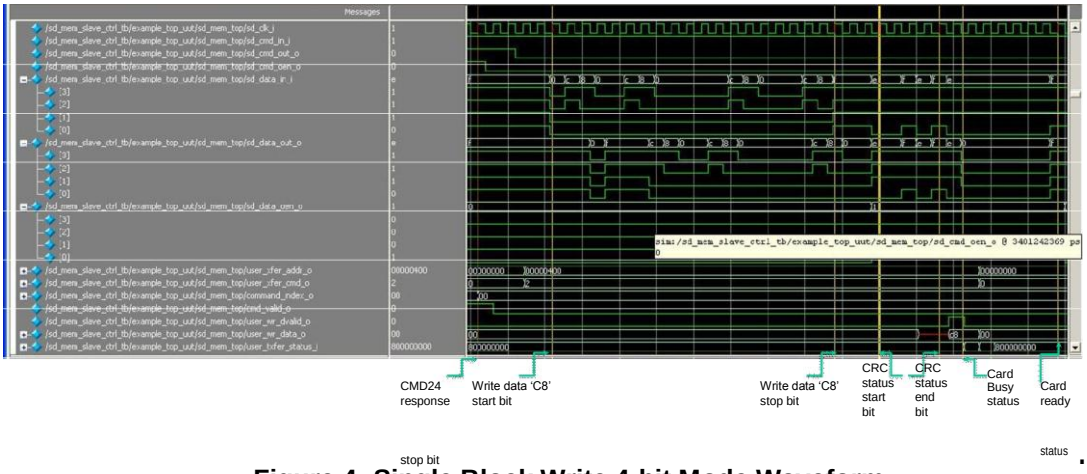


**Figure 4: Single Block Write 4-bit Mode Waveform**

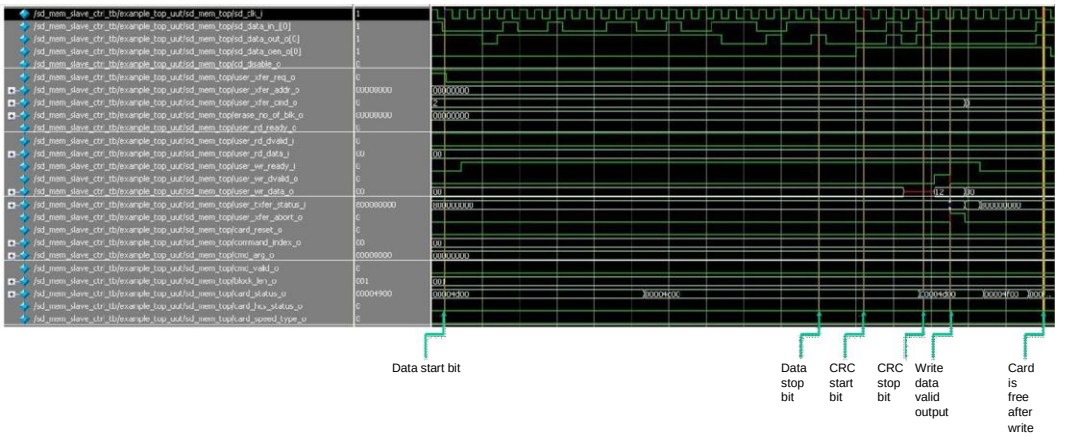


**Figure 5: Single Block Write 1-bit Mode Waveform(User Interface)**

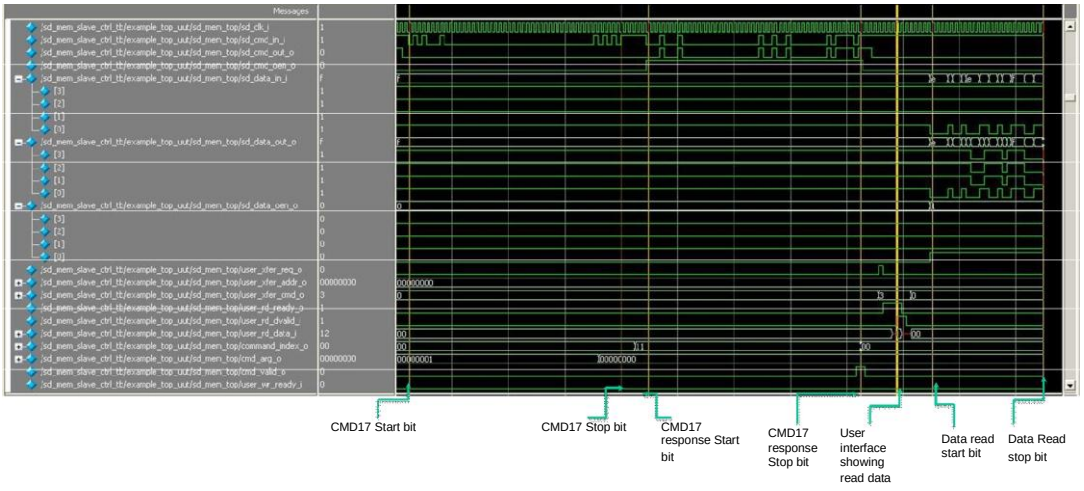


**Figure 6: Single Block Read Waveform**