DESIGN OF SECURE DIGITAL HOST CONTROLLER CARD

T.Abinaya[1], P.Priyadarshini[2], M.Ramya[3]

[1,2,3] *PG students / VLSI Design, SNS College of Technology, Coimbatore, (India)*

ABSTRACT

With the increasing consumer digital content, demand for high capacity digital storage is increasing rapidly. Today, portable storage media's are widely used in all mobile phones, digital Cameras, camcorders, and in many multimedia devices. Different memory formats like Flash, Secure Digital (SD), Compact Flash, Universal Serial Bus (USB), and Multimedia Card (MMC) are available in the market to store the digital contents. Of all these formats, SD provides many advantages over other formats. SD cards provide high storage capacity, higher transfer speed, and interoperability with Personal Computer (PC) related devices and multimedia products. Portable devices are battery operated, so they need to be power efficient. The goal of the Project is to design Secure Digital card which consumes low power and transfers data with greater speed.

Keywords*: *Secure Digital, Host controller, ADMA, FIFO

I. INTRODUCTION

The SDHC standard was developed by the Secure Digital Association (SDA) to address the requirement to support the growth in storage capacity of memory cards. The Secure Digital name was chosen because of the ability to protect copyright content through Digital Rights Management (DRM). SDHC cards are memory cards used to store files, music, pictures, video and any other data. It is defined by memory card specifications such as size, capacity, speed, security and power. It ensures interoperability among devices. SDHC cards are supported by many small electronic devices that use memory cards for storage. A regular SD card uses the first version of the standard, which limits its maximum storage capacity. SDHC cards, which use a later version of the standard, can store more data and may also support ultra high-speed data transfer rates that a regular SD card does not support. Secure digital is a non-volatile memory card format developed and managed by Secure Digital Association. SD Card is a semiconductor flash based memory device which is well known for its simple interface, high bandwidth, low cost, greater security, low power etc. The SD Card can easily be connected to a personal computer also.SD Host Controller implements the SD Host Controller standard specification version 3.0. It enables the host to access SD Devices such as SD Memory Cards, SDIO devices, SD Combo devices etc. The SD Protocol operates a Master-Slave communication model. It employs a command-response mechanism. Commands are always initiated by the Host Controller and responded to by the Card. The Host Controller has two interfaces: the System side interface and the SD Bus

interface. The Host Controller assumes that both these interfaces are asynchronous. The Host Driver is on system bus time and the SD Card is on SD Bus time.

II. ARCHITECTURE OF SD HOST CONTROLLER

The Host Controller provides a "programmed I/O" method for the Host Driver to transfer data using the Buffer Data Port register. Optionally, Host Controller implementers may support data transfer using DMA. The DMA algorithm defined in the SD Host Controller Standard Specification Version 1.00 is called SDMA (Single Operation DMA). Only one SD command transaction can be executed per SDMA operation. Support of SDMA can be checked by the SDMA Support in the Capabilities register. This specification defines a new DMA transfer algorithm called ADMA (Advanced DMA). A DMA provides data transfer between system memory and SD card without the interruption of CPU execution. Support of ADMA can be checked by the Capabilities register. When the term "DMA" is used in this document, it applies to both SDMA and ADMA.DMA shall support both single block and multiple-block transfers but does not support infinite transfers. Host Controller registers shall remain accessible for issuing non-DAT line commands during a DMA transfer execution.

The Host Driver can stop and restart a DMA operation by the control bits in the Block Gap Control register. By setting Stop at Block Gap Request, a DMA operation can be stopped at block gap. By setting Continue Request, DMA operation can be restarted. If an error occurs, DMA operation shall be stopped. To abort DMA transfer, Host driver shall reset the Host Controller by the Software Reset for DAT Line in the Software Reset register and issue CMD12 if multiple-block read / write command is executing.

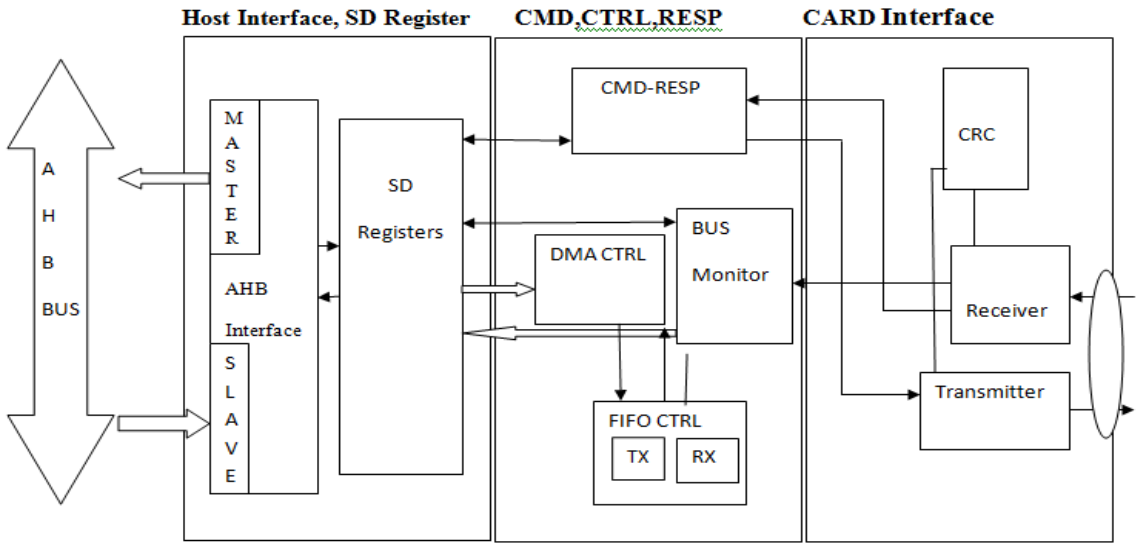

Fig 1: SD Host Controller Block Diagram

The design shown in Fig 1 is partitioned into following sections as

- Host Interface:

Host interface connects the AHB Master and Slave to AHB bus. In order to setup a command the host communicates with SD Register through slave interface. The SD controller Master interface is used for DMA Data to/from host memory.

- Controller:

The Command and Response block initiates Command to the card and receives response from the card and update SD registers.DMA Control block controls the DMA transfer to/from Host memory. The BUS Monitor looks for protocol violation and update SD status registers. FIFO (First-In First- Out) control manages the TX and RX FIFO's during transmission and reception and act as a flow control during read and write data transfers.

- Card Interface:

Card Interface consists of transmitter and receiver block that interacts with the physical Interface of the SD card. Command and Data uses CRC checker (CRC7 and CRC16).

The SDIO Host Controller is a Host Controller with an ARM processor interface. This product conforms to SD Host Controller Standard Specification Version 3.00. The Host Controller handles SDIO/SD Protocol at transmission level, packing data, adding cyclic redundancy check (CRC), Start/End bit, and checking for transaction format correctness. The Host Controller provides Programmed IO method and DMA data transfer method. In programmed IO method, the ARM processor transfers data using the Buffer Data Port Register.

Host controller support for DMA can be determined by checking the DMA support in the Capabilities register. DMA allows a peripheral to read or write memory without the intervention from the CPU. The Host Controller's AHB Host Controller system address register points to the first data address, and data is then accessed sequentially from that address.

AHB master and slave interface, enable devices to access the external memories. ADMA2 refers to the ability of an I/O device to transfer data directly to and from memory without the intervention of CPU. The Clock Manager block generates various clock frequencies for several modes of data transfer in the SD Host Controller. The initialization is carried out at 300 KHz. Interrupt generator enable only when interruption occurs during the transfer of data.

2.1 Module Description

The SDIO Host Controller consists of the following components:

- Host Registers
- Data FIFO
- CMD Control
- DAT Control
- Clock Management
- Power Management and Card Detection
- Synchronization Block

2.1.1 Data FIFO

There are two Data FIFOs:

- TX_FIFO—for buffering transmit data

• RX_FIFO—for receiving data

The TX_FIFO provides Buffer Write Enable flag to the Host Registers block. When the TX_FIFO is empty, Buffer Write Enable is asserted to indicate that the TX_FIFO is ready to accept the next block of data.

The RX_FIFO provides a Buffer Read Enable flag to the Host Registers block. When the RX_FIFO receives a block of data from the card. Buffer Read Enable is asserted to indicate the processor that data is ready for pickup.

2.1.2 Host Registers

The SD Host Controller is fully compliant to SD Host Controller Specification version 3.0 and Physical Layer Specification version 3.01. The standard register set is implemented. Internal FIFOs are provided for temporary buffering of ingress and egress data. The Host Processor accesses the various registers and FIFOs in the Host Controller to transfer data between Host and SD Card. Every data transfer is preceded by corresponding command and response. Command is send by the Host Controller to the Card and Response is send back by the Card to the Host Controller.

The Host Registers block consists of the following components:

• Standard SD/SDIO host register set

• Interrupt logic

• Auto CMD12 logic

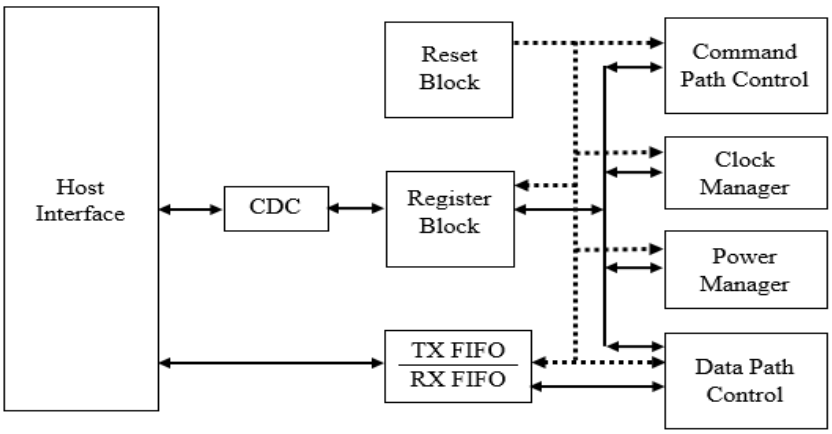

Fig 2: Block diagram of Host controller

2.1.3 Command Path Control

Command Path Control deals with the process of command transmission and response reception. All commands are 48 bits long with a 6 bit command index, 32-bit argument and a 7-bit CRC field. Each command is preceded by a start bit '0' and a transmission bit '1' and succeeded by an end bit '1'.

The Host Processor writes argument into the argument register and command into the command register in the Host Controller. Having received a command, the card checks for errors and replies with a response with the same index

field. The transmission field will be '0' and the argument field will contain status of previous command. The command path control block checks the response for CRC or any other error and stores in the Response register.

2.1.4 Data Path Control

The data path control block manages the process of writing and reading data between Host and SD Card. Transmit and receive buffers help in temporary buffering. Data is accessed as a block, in an SD card. In SDSC cards, the block size is not fixed, but default size is 512 bytes. In SDHC and SDXC, the block size is fixed 512 bytes. Every data block is preceded by a start bit '0' and succeeded by 16-bit CRC and stop bit '1'.

For write and read, the SD Host Controller supports both single block and multiple block access. In single block read and write, only one block of data is transferred with each read or writes command. In multiple block access method, infinite number of blocks can be transferred by sending one command. For stopping a multiple block operation, a transmission stop command (CMD12) is sent by the Host Controller.

To perform a write operation, the data to be transferred is written to the Transmit FIFO through the Host Interface and data write command is issued. When the write command is sent by the Command Path Control block, the Data path control block sends the data along with CRC and end bit.

To perform a read operation, the read command is sent. After receiving the read command, if no error is found, the SD Card sends the requested data through the data lines. It will be checked for any errors and stored in the Receive FIFO. The Host Processor can read it through the Host Interface.

2.1.5 Clock Management

The Clock Management block is used to divide and enable/disable the SD clock. It consists of a Clock Divider and SD Clock Control Logic. The Clock Divider is a loadable counter. When a new value is written to the Clock Control register, the Host Register block loads the new value to the Clock Divider. Whenever the internal clock enable bit in the Clock Control register is set, the Clock Divider starts to count down from the loaded value to '0'. The SD Clock Control Logic controls SD clock toggling. When the Clock Divider reaches zero and SD clock enable bit in the Clock Control register is set, the SD clock toggles.

III ADVANCED DIRECT MEMORY ACCESS

In the SD Host Controller Standard Specification Version 3.00, new DMA transfer algorithm called ADMA (Advanced DMA) is defined. The DMA algorithm defined in the SD Host Controller Standard Specification Version 3.00 is called SDMA (Single Operation DMA). SDMA had disadvantage that DMA Interrupt generated at every page boundary disturbs CPU to reprogram the new system address. This SDMA algorithm forms a performance bottleneck by interruption at every page boundary. ADMA adopts scatter gather DMA algorithm so that higher data transfer speed is available. The Host Driver can program a list of data transfers between system memory and SD

card to the Descriptor Table before executing ADMA. It enables ADMA to operate without interrupting the Host Driver. Furthermore, ADMA can support not only 32-bit system memory addressing but also 64-bit system memory addressing. The 32-bit system memory addressing uses lower 32-bit field of 64-bit address registers. Support of SDMA and ADMA are optional for the Host Controller.

There are two types of ADMA; ADMA1 and ADMA2. ADMA1 can support data transfer of only 4KB aligned data in system memory. ADMA2 improves the restriction so that data of any location and any size can be transferred in system memory. The format of Descriptor Table is different between them. The Host Controller Specification Version 3.00 defines ADMA2 as standard ADMA and recommends supporting ADMA2 rather than ADMA1. "ADMA" is used in this document; it applies to both ADMA1 and ADMA2.

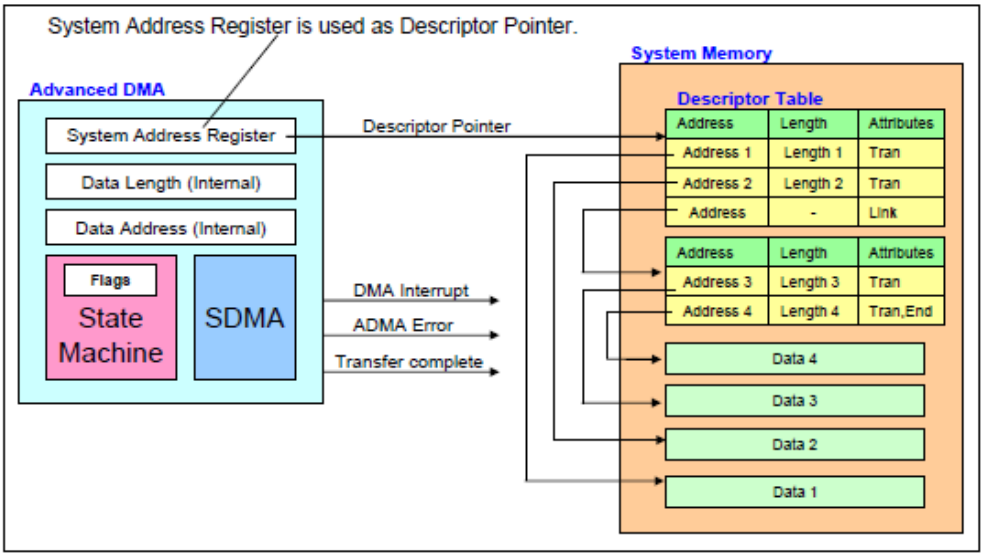

Fig 3: Block diagram of ADMA2

The Descriptor Table shown in Fig 3 is created in system memory by the Host Driver. A 32-bit Address Descriptor Table is used for the system with 32-bit addressing and 64-bit Address Descriptor Table is used for the system with 64-bit addressing. Each descriptor line (one executable unit) consists of address, length and attribute field. The attribute specifies operation of the descriptor line. ADMA2 includes SDMA, State Machine and Registers circuits.

ADMA2 does not use 32-bit SDMA System Address Register (offset 0) but uses the 64-bit Advanced DMA System Address register for descriptor pointer. Writing Command register triggers off ADMA2 transfer. ADMA2 fetches one descriptor line and execute it.

3.1. ADMA2 States

Four states are defined; Fetch Descriptor state, Change Address state, Transfer Data state, and Stop ADMA state. The state operation of ADMA2 consists of four states as Fetch Descriptor state, Change Address state, Transfer Data state, Stop ADMA state and these four state description is shown below in Table1

STATE NAME	OPERATION
ST_FDS (Fetch descriptor)	ADMA2 fetches a descriptor line and set Parameters in internal registers
ST_CADR (Change Address)	Link operation loads another descriptor address to ADMA system Address register
ST_TFR (Transfer data)	Data transfer of one descriptor line is executed between system memory and SD card. If data transfer continues go to ST_FDS state. If data transfer completes, go to ST_STOP state.
ST_STOP (Stop DMA)	ADMA2 stays in this state in following cases: • After power on reset or software reset • All descriptor data transfer are completed. If new ADMA2 operation is started by writing command register, go to ST_FDS state.

Table 1: ADMA2 states

IV. SD HOST STANDARD REGISTER MAP

4.1. Summary of Register set

As per SD specification 3.0, registers are classified into 9 categories as shown below:

- SD Command Generation: Register to support Command Generation
- Response: Register to store Response from the Device
- Buffer Data Port: Registers to access Internal Data Buffer
- Host controls: Register to Store state of the Controller (status)
- Interrupt controls: Interrupt control Register (Masks, Enable, Status)
- Capabilities: Register for vendor specific capabilities of the controller
- Force Event: Software Interface registers to generate events
- ADMA: Advance DMA Control Register
- Common Area: Common Information Area (for multiple card support)

4.2. Host Registers

The Host Registers block consists of the following components:

• Standard SD/SDIO host register set
• Interrupt logic
• Auto CMD12 logic

- **Standard SD/SDIO Host Register set**

The standard SDIO host register consists of seven parts:

- SD command generation — parameters to generate SD commands
- Response — response value from the card
- Data port — data access port to the internal buffers
- Host controls — host mode control and status
- Interrupt controls — interrupt status and enables
- Capabilities — host controller capability information

- CMD12 Logic

CMD12 command is issued during multiple block transmission. To stopping a multiple block operation, a transmission stop command (CMD12) is sent by the Host Controller. If the Auto CMD12 Enable bit in the Transfer Mode register is set, then CMD12 command is issued automatically.

4.2.1. Types Of Host Register

Host registers are of 5 types as

- RO – Read Only Register

Register bits are of read-only type and it cannot be altered by any software or by reset operation.

- ROC – Read Only Status

Bits are initialized to zero at reset. Writes to these bits are not allowed.

- RW – Read Write Register

Register bits are read-write and can be set or cleared by software.

- RW1C – Write-1-to-clear

Register bits indicate status when read, a set bit indicating a status event may be cleared by writing a '1'. Writing a '0' to RW1C bits has no effect.

- RSVD – Reserved

Bit can be defined for future use and is currently set to '0'.

4.3. SD Card Operation Modes

To get the card to "boot" into SD mode the host has to pull the command line high for at least 74 clock cycles. The SD card has 10 predefined states, and the host operates in 3 modes.

CARD STATE	OPERATION MODE
Inactive State	Inactive
Idle State	Card Identification Mode
Ready State	
Identification State	
Stand- by State	Data Transfer mode
Transfer State	
Sending data state	
Receive- data state	
Programming State	
Disconnect state	

Table 2. SD Card Operation Modes

In card identification mode, the host initializes the card, the goal of this state is to validate if the card and host are compatible. Also the host will know what kind of card it is communicating with SD, SDHC, SDIO or MMC card. The differences between the cards are in how they respond to the commands in the identification mode. All the communication in this mode is performed on the command line. At start of the identification process the card should first be reset, which is done by sending it a reset command. This is done to be sure that the card is in idle state. When the card is in idle state, then it is ready for receiving commands. In the next step the interface condition is validated. To validate the interface condition, the host sends its voltage operation range to the card. The card responds only if it is specification 3.0 compatible and if the voltage operation is valid for the card.

If no response is received the host realizes that it is either a 1.0 card or an operation condition mismatch, with an operation mismatch to be discovered with the next command in the chain. However, if a response is received the host knows that it is a 3.0 card and additional features can be enabled. A received response is validated with CRC check and bit matching. An invalid response will put the card to inactive state.

Next command specifies the operation condition to the card, this command has two purposes: the first is to specify the required operation voltage window; the second is to check if the card is busy. A card with an incompatible voltage window should discard itself from further bus operations and no response will be delivered. When a response of the operation condition is received it checks for a busy bit in the response data field. If the card is in busy state, the operation command is resent until the busy bit is cleared, then the host knows the card has finished its start-up operations.

4.4. Data Transfer Mode

After the identification mode the card will be in the idle state. To set the card to transfer state one additional command containing the response number is sent, card with matching response number will then be put in transfer

state. Before transmitting data there are some settings that should be set, like the data width which can be set to 1 or 4 bit and the block length that can be set to a value between 1 and 2048?

- DATA WRITE

To write data to the card either a single block or a multiple blocks command can be used. First the command is sent, and a response is received. When the host has received the response, the data transfer can begin.

After a data transmission, the card replies with a 3-bit CRC status token, this always occurs 2 cycles after the end bit. "010" indicates a successful transfer (no CRC error). In case of CRC error the card response with "101" and no data will be written to the card. After a data write the card can become busy if its write buffer is full.

- DATA READ

The reading of data is performed in a similar manner. First a read block command is sent, the host should then be prepared that data can arrive at any moment. The card starts to send data after receiving a data read command

4.5. Multiple Block Operation

There are also one transfer mode where there is no need to issue a command for each block to be sent. Instead just the address of the first block is sent, and then the card transfer data until a special stop command is received.

V. SD BUS INTERFACE

SD Bus Interface is an advanced 9-pin bus is shown in Fig 4 consisting of one clock line, one bi-directional command line, 4 bi-directional data line and 3 power lines. The bus is designated to operate at a maximum operating frequency of 200MHz depending on the capabilities of the Card.

The Data transfers to and from the SD Memory Card are done in blocks. Data blocks are always succeeded by CRC bits. Single and Multiple block operations are defined. Data can be transferred using single or multiple data lines

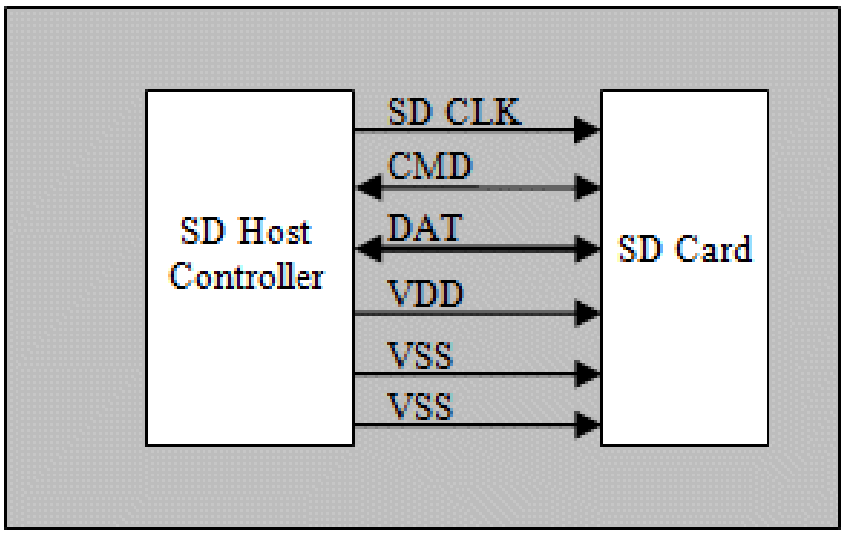

Fig 4: Interfacing of Bus

VI. SIMULATION RESULTS AND DISCUSSION

6.1 Output Waveform of Initial State (00)

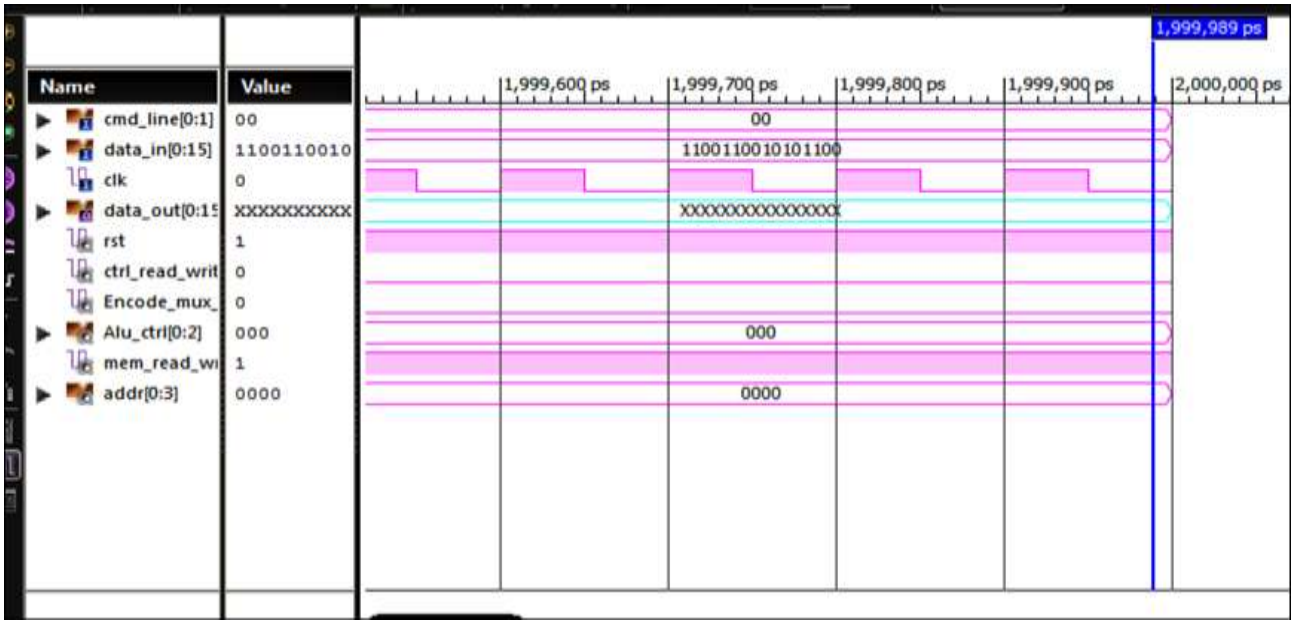

Fig 5: Waveform of the Initial State (00)

At this state, Data is received from the memory card and it is said to be in wait mode is shown in Figure 6.1. Here only reset operation is performed

6.2 Output Waveform of Next State (01)

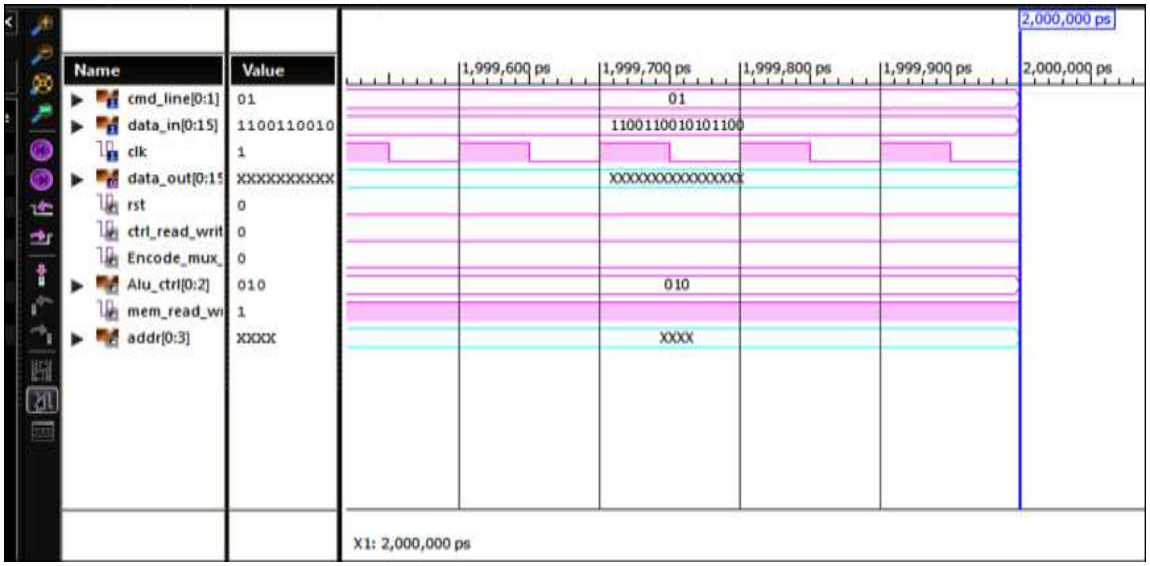

Fig 6: Waveform of the Next State (01)

As shown in Fig 6, When the command line input is changes to next state, only read operation is performed and the counter gets incremented.

6.3 Output Waveform of Next State (10)

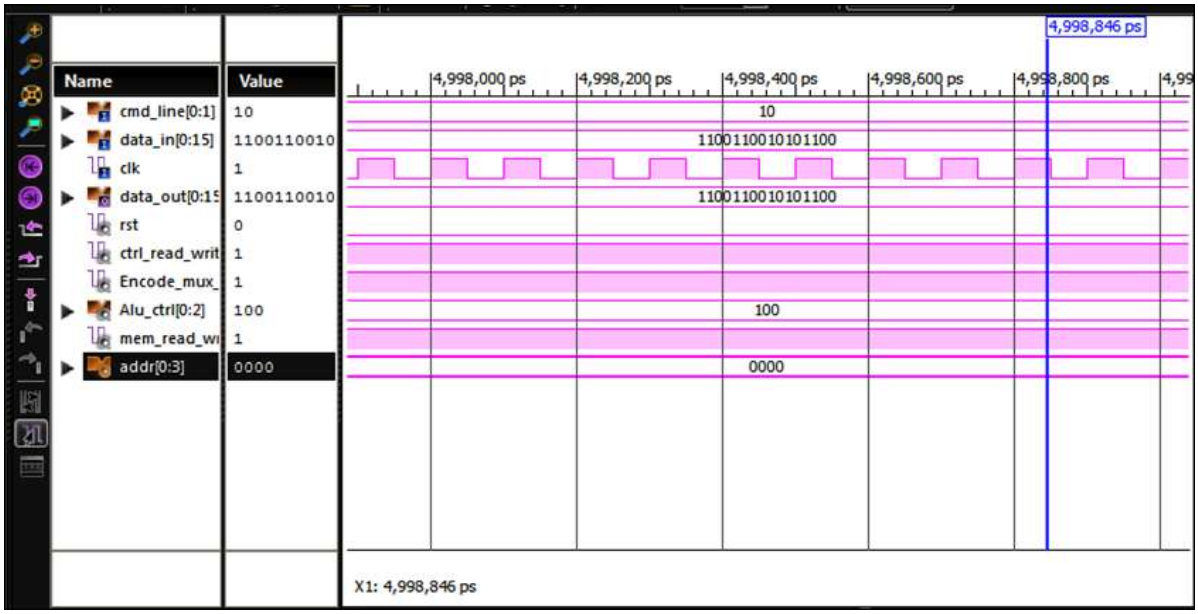

Fig 7: Waveform of the State (10)

When the command line input changes, the data which is read during previous state will be written from the host to the memory card and also counter gets incremented as shown in Fig 7.

6.4 Output Waveform of Next State (11)

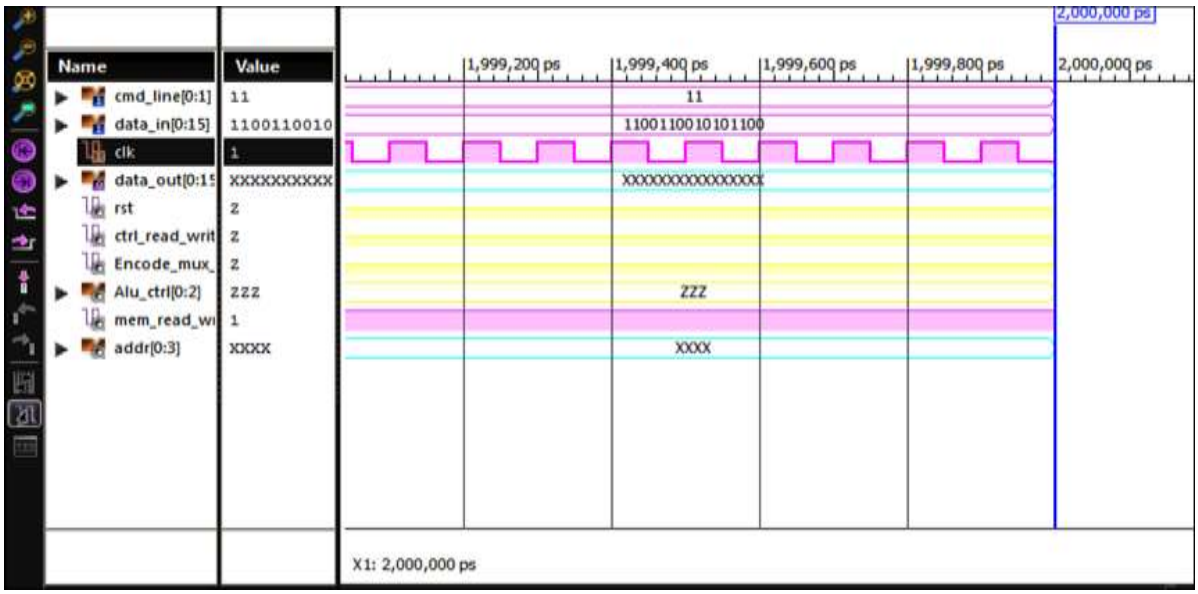

Fig 8: Waveform of the State (11)

When a command line input is applied as 11, the result will not be reflected and it goes to high impedance (ie) undefined state is shown in Fig 8.

6.5 Output Waveform of All Four Combined States

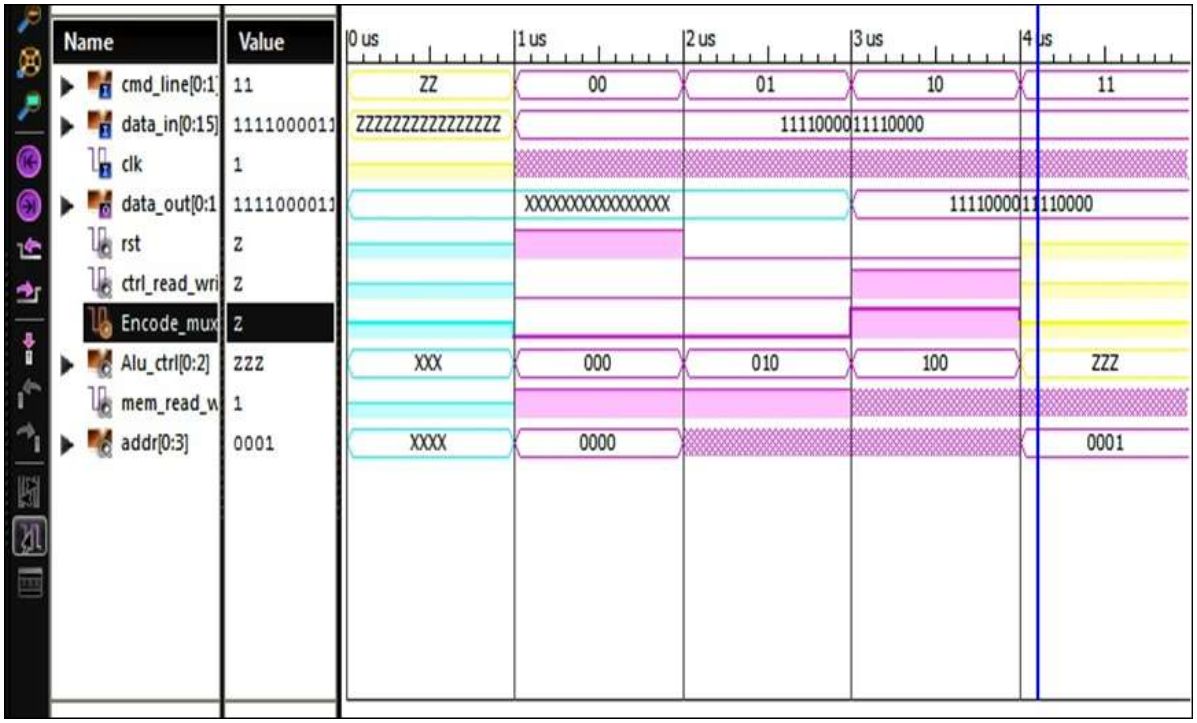

Fig 9: Waveform of FIFO Data Transfer

The Fig 9 shown above represents the data transfer in four different states. During initial state, no read or write operation is performed and in second state, the data is only read from the card to the host and in next state, the data which is read in previous state will displayed and the last state is the undefined state and no operation will be performed.

VII. CONCLUSION

Thus, the operation and the features of Secure Digital card were studied. Finally, the data transfer is performed by using ADMA2 mode and test cases covering various scenarios are verified under simulation in Xilinx design suite 13.2 using verilog codings. Future Improvement is to design the Secure Digital Host controller card to compare the data transfer speed with the existing card and also to verify the functionalities of the design using system verilog.

REFERENCES

[1] SD Specifications part A2 SD host Controller simplified- Specification, Technical Committee SD Association- 2013.

[2] SD Specifications part 1 physical layer Simplified Specification, Technical committee SD Association- 2012.

[3] SD Specifications part 1 Ultra high Speed(UHS) Interface Simplified addendum version 1.00, Technical Committee SD Association- 2011.

[4] Jaikiranl, Prakash biswagar, "Functional Verification of Secure Digital Host Controller", International Journal of Electronics Signals and Systems (IJESS), ISSN: 2231- 5969, VOL-3, ISS-1, 2013.

[5] Jose Simon, Deepu K. Krishnan, S. Krishnakumar Rao, Biju C. Oommen, R. Ravindrakumar, "Design and Implementation of SD Host Controller IP Core", July 2012.