A Case Study: Verification of WISHBONE SD Card Controller using System Verilog

Pankaj Chavda[1]
[1]M. E.(VLSI and Embedded Systems Design)
[1]Electronics and Communications Engineering Department
[1]Gujarat Technological University

Abstract--- SD Memory Card is data storage device which is widely used in mobile phones, tablets, digital cameras, camcorders and in many electronics devices where high memory capacity with lower size is required. WISHBONE SD Card Controller is a host controller which connects the WISHBONE bus at one side and SD bus to another side. Here a verification environment for WISHBONE SD Card Controller IP Core which is written in Verilog is developed using System Verilog language. Constrained randomization and layered test bench approach is used for development of environment.

I. INTRODUCTION

SD Memory Card is a non-volatile digital data storage device developed by SD Card Association (SDA) which provides high capacity secured storage with lower size[1]. It is widely used in digital electronics devices where the purpose of high capacity storage with lower size is required like mobile phones, tablets, digital cameras, camcorders etc.

SD Memory Card comes with three different capacity standards: SDSC (Standard Capacity) with the maximum storage capacity of 2 GB, SDHC (High Capacity) with the maximum storage capacity of 32 GB and SDXC (eXtended Capacity) with the maximum storage capacity of 2 TB.

WISHBONE SD Card Controller is IP Core developed by OpenCores[5] which is used to connect the SD Memory Card with the processor[2].It connects 32-bit WISHBONE bus, which is interface with processor, at one side and 4-bit SD bus, which is interfaced with SD Memory Card, at another side as shown in Fig. 1.

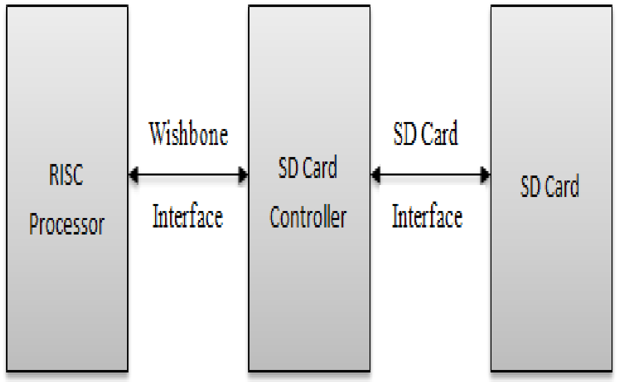

Fig. 1: WISHBONE SD Card Controller interface

II. ARCHITECTURE OF WISHBONE SD CARD CONTROLLER

Fig. 2 shows the architecture of WISHBONE SD Card Controller IP Core. The whole IP Core is divided in to 7 major modules according to their functionality[2].

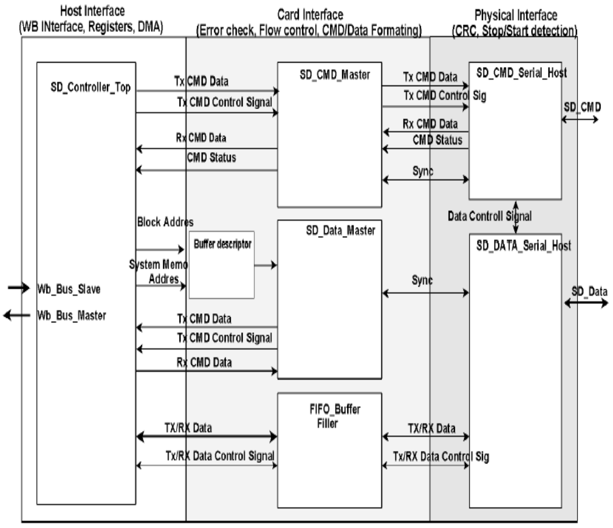

Fig. 2: Architecture of WISHBONE SD Card Controller (Source: OpenCores[5])

A. *Host Interface*

This module is the interface between the SD Card Core and the WISHBONE bus. Two WISHBONE interfaces (slave and master) are used for this. The internal registers and buffer descriptors (BD) are all accessed through the 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 registers accessible by user from the WISHBONE slave, and the required logic to access the same.

B. *SD Command Master*

The SD CMD Master module synchronizes the communication from the host interface with the physical interface. It performs three main tasks. (1) 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 Command Host. (2) Read response messages from the SD Command Host and forward it to the user accessible register in the SD Controller Top module. (3) Keep track of the status of the SD Command Host module.

C. *SD Command Host*

This module is the interface towards physical SD Memory card command pin. This module takes care of the physical sending and receiving of the messages, adding start bits, stops bits and CRC checksum.

D. *SD Data Master*

This module continuously checks for new BDs (Buffer Descriptors) that are needed to be processed. If so the module generates a command and set up the DMA to read/write to correct address. If the command line is free the module sends the command and waits for response. If response is valid the module starts the DMA if not valid the command is send again. During transmission the module keep track for FIFO buffer underflow or overflows. When the transmission is completed it checks for valid CRC. If anything goes wrong during a transmission a stop command is send and the module tries to restart the transmission *n* times before giving up.

E. SD Data Host

This module is the interface towards physical SD card device Data port. The interface consists of only 5 signals, one clock signal and the 1-4 bit bi-direction Data signal.
The module performs two actions. (1) Synchronizes request for write and read data. (2) Adding a CRC-16 checksum on sent data and check for correct CRC-16 on received commands.

F. Buffer Descriptors (BD) Structures

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

G. FIFO Fillers

This module works as the DMA. It manages receive and transceiver FIFO buffer for the data stream. It keeps track of the status of the FIFOs.If something goes wrong, like full receiver FIFO or empty transfer buffer it signals error.

III. SYSTEM VERILOG[4] LANGUAGE

System Verilog is a Hardware Verification Language (HVL).It is extended version of Verilog HDL[4].It has the advantage of Object Oriented Programming (OOP). The remarkable features of System Verilog are constrained randomization; inter process communication techniques, coverage, assertions etc. Using these features robust verification environment can be developed.

III. VERIFICATION ENVIRONMENT

Here WISHBONE SD Card Controller IP Core is referred as Design Under Test (DUT). The environment is developed using System Verilog language. Layered test bench concept is used to develop the verification environment. Since System Verilog is used it is possible to develop layered test bench.

Fig.3 shows the layered test bench environment. Here the test bench is divided in to different layers according to their functionality.

Layered test bench is divided in to different layers according to functionality. Different layers are shown in Fig.3. Test layer has all possible test cases according to specifications. Scenario generation and stimulus generation are performed at scenario layer. Score boarding and monitoring and checker functionality are performed at functional layer. Driving of stimuluses' are performed at command layer. DUT is kept at signal layer as we deal with all signals of DUT.

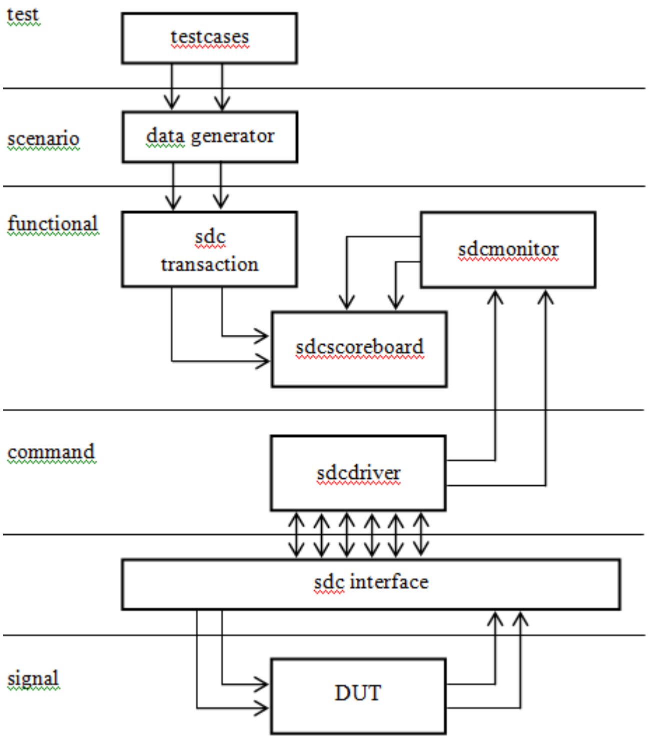

Fig. 3: Layered test bench for WISHBONE SD Card Controller

A. sdc_global class

In sdc_global class all global signals are defined. Signals that need to be randomized are defined with keyword *rand*. These global signals can be imported to any class for further use.

B. sdc_generator class

The function of this class is to generate the required data for verification. All randomization processes are done inside this class.Randomized data are exported using mailbox facility of SystemVerilog.

C. sdc_xactor class

The function of this class is to produce the transactions for read and write operation of DUT. All configuration of read and write configuration are done here.

D. sdc_driver

The function of this class is to drive all the transaction according to the requirement of DUT. Thus driving protocol is developed in this class. All input transactions are passed to the DUT through interface and all responses from DUT are stored for further operations.

E. sdc_monitor

The function of this class is to monitor the all input and output signals of the DUT. All outputs from the DUT are stored and forwarded to the scoreboard.

F. sdc_scoreboard

The functionality of this module is to check whether the transaction passes or fails. Thus this module keeps the track of passed and failed transactions. It has internal checker functionality which checks for the pass or failof transactions.

IV. SIMULATION RESULTS

The proposed verification environment is simulated using EDA tool Mentor Graphics QuestaSim 6.6d. To simulate the environment we have developed the functional model of SD Memory Card. DUT is exercised along with the verification environment and SD Memory Card functional model.

First of all DUT and SD Memory Card model need reset. After that DUT is configured for timeout operation and clock dividing operation. Then by using proper sequence as listed below the SD Memory Card is put in to transfer state. After the SD Memory Card is in transfer state read and write transactions can be performed.

Sequence to put SD Memory Card in to transfer state

(1) Send CMD0 command which resets the SD Memory card.
(2) Send CMD8 which asks the SD Memory Card for supported voltage range. In response to the CMD8 card provides supported voltage range in response register.
(3) Send CMD55 which indicates that the next command is application specific command.
(4) Send application specific command ACMD41 which send host capacity information and asks for the card operating condition. In response to ACMD41 card provides OCR (operating condition) in response register.
(5) Send CMD2 command which asks the card for CID (Cad Identification Number). In response card provide CID in response register.
(6) Send CMD3 command which asks the card to publish new RCA (Relative Card Address). In response card provide RCA in response register.
(7) Send CMD7 command which put the card in transfer state.
(8) Send CMD55 which indicates that the next command is application specific command.
(9) Send application specific command ACMD6 which defines the width of data bus which will be used for transactions.

Fig. 4 shows the simulation result for successful read transaction of WISHBONE SD Card Controller. For read transaction we need to provide the block address from where we need to read and WISHBONE address to where we need to write the same read data. According to WISHBONE protocol[3] DUT drives the signal and we get the output read data. This provides the evidence that the DUT performs the read transaction successfully.

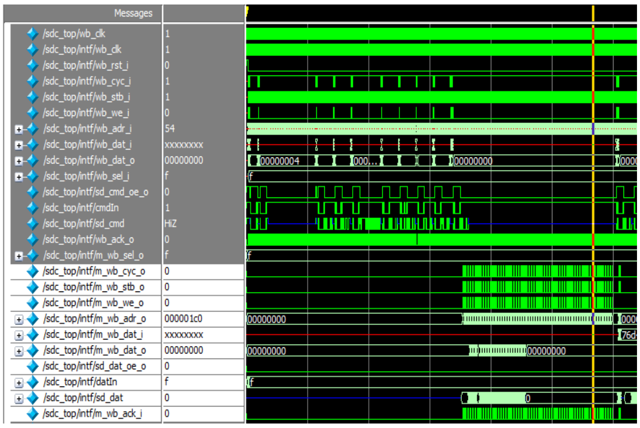
Messages
/sdc_top/wb_clk 1
/sdc_top/intf/wb_clk 1
/sdc_top/intf/wb_rst_i 0
/sdc_top/intf/wb_cyc_i 1
/sdc_top/intf/wb_stb_i 1
/sdc_top/intf/wb_we_i 0
/sdc_top/intf/wb_adr_i 54
/sdc_top/intf/wb_dat_i xxxxxxxx
/sdc_top/intf/wb_dat_o 00000000
/sdc_top/intf/wb_sel_i f
/sdc_top/intf/sd_cmd_oe_o 0
/sdc_top/intf/cmdIn 1
/sdc_top/intf/sd_cmd HiZ
/sdc_top/intf/wb_ack_o 0
/sdc_top/intf/m_wb_sel_o f
/sdc_top/intf/m_wb_cyc_o 0
/sdc_top/intf/m_wb_stb_o 0
/sdc_top/intf/m_wb_we_o 0
/sdc_top/intf/m_wb_adr_o 000001c0
/sdc_top/intf/m_wb_dat_i xxxxxxxx
/sdc_top/intf/m_wb_dat_o 00000000
/sdc_top/intf/sd_dat_oe_o 0
/sdc_top/intf/datIn f
/sdc_top/intf/sd_dat 0
/sdc_top/intf/m_wb_ack_i 0
00000004
000...
00000000

Fig. 4: Simulation result of read transaction

Fig. 5 shows the simulation result of successful write transaction of WISHBONE SD Card controller. For write transaction we need to provide the WISHBONE address from where we get the required data and block address to where we will write the data. According to WISHBONE protocol[3] DUT performs the write transactions.

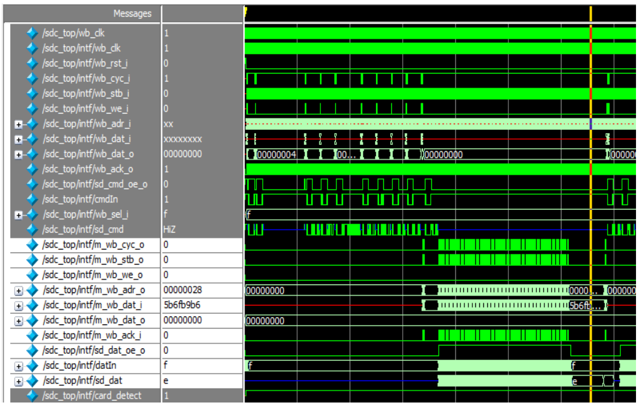
Messages
/sdc_top/wb_clk 1
/sdc_top/intf/wb_clk 1
/sdc_top/intf/wb_rst_i 0
/sdc_top/intf/wb_cyc_i 1
/sdc_top/intf/wb_stb_i 0
/sdc_top/intf/wb_we_i 0
/sdc_top/intf/wb_adr_i xx
/sdc_top/intf/wb_dat_i xxxxxxxx
/sdc_top/intf/wb_dat_o 00000000
/sdc_top/intf/wb_ack_o 1
/sdc_top/intf/sd_cmd_oe_o 0
/sdc_top/intf/cmdIn 1
/sdc_top/intf/wb_sel_i f
/sdc_top/intf/sd_cmd HiZ
/sdc_top/intf/m_wb_cyc_o 0
/sdc_top/intf/m_wb_stb_o 0
/sdc_top/intf/m_wb_we_o 0
/sdc_top/intf/m_wb_adr_o 00000028
/sdc_top/intf/m_wb_dat_i 5b6fb9b6
/sdc_top/intf/m_wb_dat_o 00000000
/sdc_top/intf/m_wb_ack_i 0
/sdc_top/intf/sd_dat_oe_o 0
/sdc_top/intf/datIn f
/sdc_top/intf/sd_dat e
/sdc_top/intf/card_detect 1
00000004
00...
00000000

Fig. 5: Simulation result of write transaction

V. CONCLUSION AND FUTURE WORK

In this paper a verification environment for WISHBONE SD Card Controller is developed using System Verilog language. The layered test bench architecture is developed using the facility of System Verilog language which comprise different classes with different functionality. Simulation results prove the proper functionality of WISHBONE SD Card controller.

In future DUT can be exercised with functional coverage plan and assertions which will prove the error free working of DUT.

REFERENCES

[1] SD Specifications, Part 1, Physical layer simplified specification, Version 4.10, January 22, 2013, by SD Group (Panasonic Corporation, SanDisk Corporation and Toshiba Corporation).
[2] SD/MMC Controller specifications, Adam Edvardsson, OpenCores, Rev. [0.1], May 23, 2009
[3] WISHBONE System-on-chip (SoC) Interconnect architecture for Portable IP Cores. Revision: B.3, Released: September 7, 2002
[4] Chris Spear. System Verilog for verification. A guide to learning the test bench language features. Second Edition. Springer publication
[5] OpenCores, Free IP Core Provider, www.opencores.org