iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** VI **Month of publication:** June 2019

DOI: http://doi.org/10.22214/ijraset.2019.6363

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

System Verilog based AMBA AHB Protocol

Kumar Govind
Visveshvaraya technological university, PG Center, Belagavi, India

Abstract: ***Integrated circuits have emplane the era of a system-on-chip (SoC), which advent integrating all peripheral electronic devices into a single chip. In the proposed system we design and verify AMBA AHB protocol. The DUT is developed by considering AHB operating states IDLE, SETUP, ACCESS, when the AHB address is available with high HWRITE, HSEL, and HGRANT signals. The data is written to AHB write signal and the verification environment is developed with testbench components like test, driver, generator, monitor, and scoreboard. Verilog and SystemVerilog codes are simulated on the L-2016.06-SP2-5 Synopsys VCS tool. The code coverage obtained using Verilog is 70.10%, to improve the code coverage the same DUT is designed and simulated in the SystemVerilog environment, code coverage obtained is 89.33% and 100% functional coverage is achieved. We obtain a 19.23% improvement in the code coverage using SystemVerilog.***

Keywords: AMBA AHB, Verilog, SystemVerilog.

I. INTRODUCTION

Due to VLSI technology, semiconductor industries are enhanced a lot because it achieved very high density of components in a single chip. But the demand of market increased due to which complexity also increased. The time required for verifying the plan is getting to be monotonous as the complicacy of the chip configuration is increasing exponentially [1]. As the improvement of System on-chip lead to the change in processor designs even the addressing an AHB protocol must be configurable like a standard bus [2]. Our main concern is to improve the code coverage and the functional coverage of AMBA AHB Protocol which is improved by the AHB master SystemVerilog code [3]. Verification is a challenging task for a design engineer in the entire design and verification period, because error which are not covered in the earlier stages which of the design bring on in the next stage of the design and later it is too complex to analyze it [5]. Massive complexity of chip increase in recent years and integration of more number of transistors in a single SoC is very critical. The paper presents the overall AMBA AHB architecture in Section II. Section III and IV presents the detailed proposed method, simulation results of Verilog and SystemVerilog. Section V provides the proposed work of the AHB master [6]. The time required for verifying the design is getting to be monotonous ordinarily as the complexity of the chip configuration is expanding exponentially. These days, about 70% of the design time is required for developing the verification environment [13].

II. METHODOLOGY

AMBA is a wide specification protocol, which describes number of bus and interface. It is enrolled by trademark of ARM in 1996.AMBA has three buses that is show in below architecture [1].

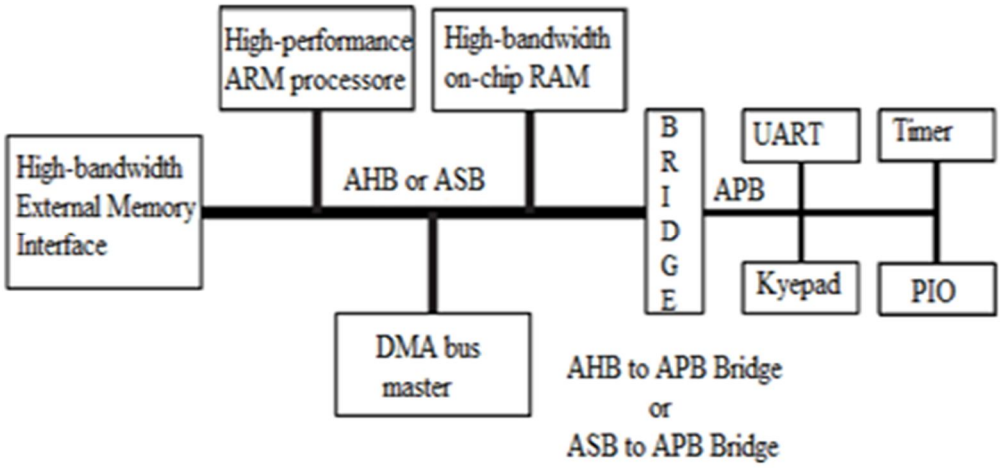


Figure 1 Advanced microcontroller bus architecture

Advanced High Performance Bus

AMBA AHB is the high-performance bus, high c1ock frequency system bus. The AHB features are

1) Burst transfer
2) Split transactions
3) Single cyc1e bus master handover
4) Single c1ock edge operation
5) Pipelined operation
6) Multiple bus master

A. AMBA AHB Description

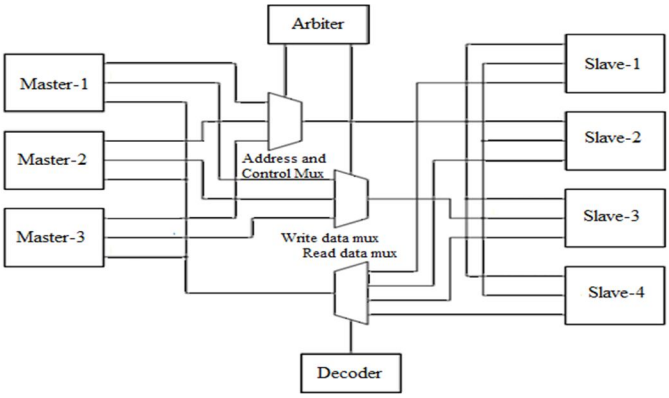


Figure 2 AMBA AHB Master Architecture

The AMBA AHB Architecture is shown in figure 2. The AHB is a high performance bus and it supports the high efficient network of processor, off-chip and on-chip memory interface with low-power peripheral microcell functions. AMBA aim is easing the component design by utilizing the combination of interchangeable components in the SoC design [11]. AHB Architecture which consist of four components they are Master, Slave, Decoder and Arbiter. Arbiter is the signal controlling component in the above architecture. Decoder is used for decode the each and every transferring data between slave and master. Master is act as input and slave is act as output and also perform write and read operation.

B. AMBA AHB master

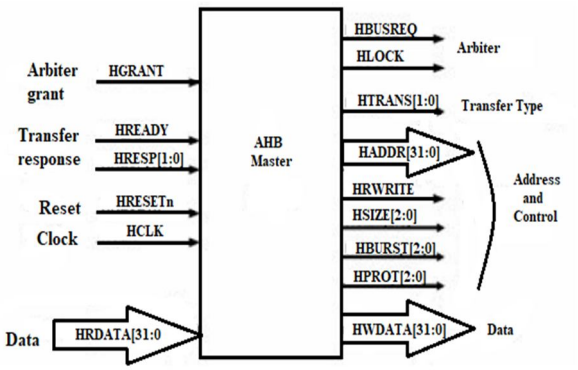


Figure 3 AHB Interface

C. Blocks in AHB

1) Master: The write and read operation between the master and slave should be done after driving the control and address signals but at one time only one master should be active
2) *Slave:* Slave device responds to communication requests from a master
3) *Arbiter:* AHB arbiter is to grant the particular master to access the data bus using arbitration techniques.
4) *Decoder:* It decodes the address of each data transfer to select the particular signal from the slave for read operation.

Table 1 Selection of Arbitration Algorithm

Arbitration[1:0]	Arbitration selection algorithm
00	High priority algorithm
01	Fair chance algorithm
10	Random access algorithm
11	Round robin algorithm

D. Round Robin

When the arbitration scheme selected signal ARBITRATION=01, arbiter goes into this state

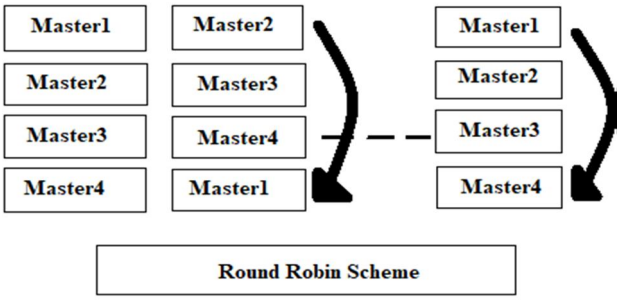


Figure 4 Round Robin algorithms

In Round Robin algorithm master1 as shown in the figure 4 have grant to access the share bus. In clock-1, after data done signal master1 move to the bottom and in next clock, grant is given to master-2, this process continues till all the master get grant to access to bus. After fourth clock master-1 is again in top position have grant to access the bus [1].

E. Verification Environment

The verification environment to verify the AMBA AHB protocol shown in figure 6. The different modules of the environment are explained.

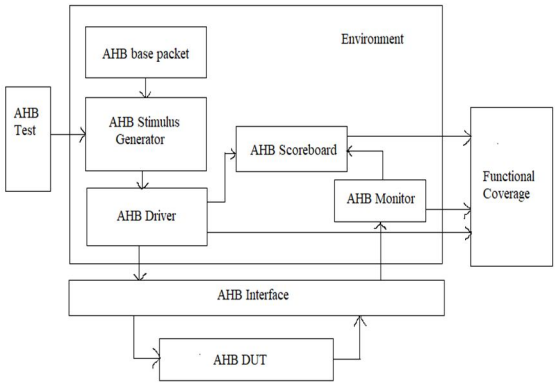


Figure 5 Verification Environment

Verification environment is the group of classes performing specific operation. Above figure 5 shows that in base packet we are declaring the variables and next in tx_gen we are generating the stimuli's for that input variables that are declared in base packet. From tx_gen the data is giving to driver. Then data transactions will drive through driver to DUT through interface. The monitor receives the data from DUT through the interface and it displays the various messages as per the operation. The monitor sends the actual input to Scoreboard and the driver sends the expected output Scoreboard to verify the actual and expected results are matched or not. The test is the top layer of verification specification and works as the functional block. Under the base packet, tx gen, driver, monitor and scoreboard are using to create the design of the modules to generate the signals and output is to check the monitor and scoreboard

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

III. PROPOSED METHOD

AMBA AHB Protocol designed by using Verilog that should be compile using Synopys VCS Tool, after compilation completed. Simulate the program and check the waveforms of design and check the address ,write data and read data. And the proposed method is to create the SystemVerilog verification environment for the AMBA AHB protocol compile and simulate it by using Mod Synopsys VCS tool after the simulation is succeed and verify the output results with verilog results.

IV. SIMULATION RESULTS

Simulation results of design and verification AMBA AHB protocol with Verilog and SystemVerilog.

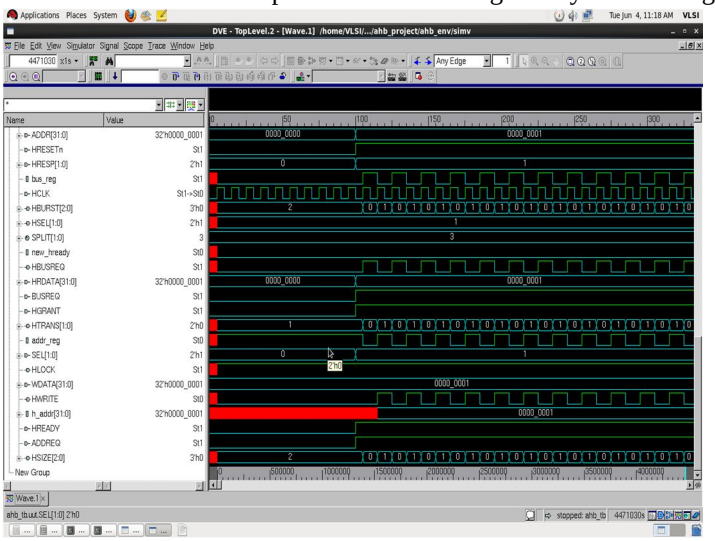


Figure 6 AHB DUT for simple Transfer

Simulation result consist of simple transfer of bit and Code coverage using Verilog is observed in figure 6, the simple transfer for AHB DUT and read/write operations means simple transfer initiates by master, then the slave acknowledge back by signal HREADY signal to master, means received successfully,

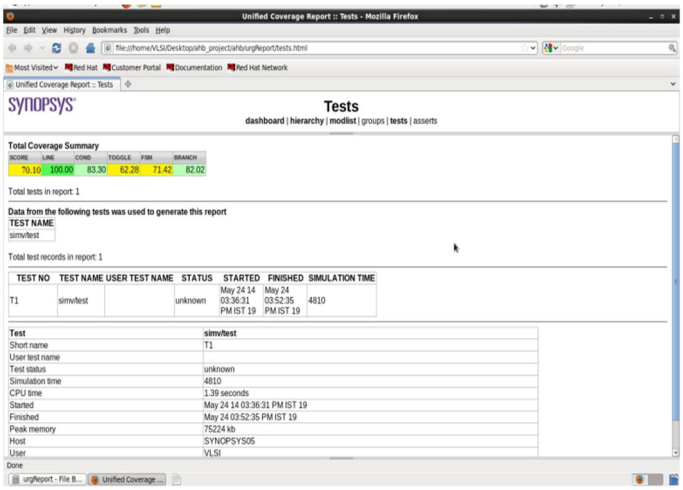


Figure 7 Coverage Report using Verilog

The total coverage report of the test using Verilog, by using Verilog 70.10% code coverage, all lines are covered means 100% line coverage, 83.30% conditional coverage, 62.28% toggle coverage, 71.42% FSM coverage and 82.02% branch coverage is obtained. The required simulation time is 4810 seconds and it has peak memory 75224 kb, host of this test is Synopsys and the total test records in report 1.

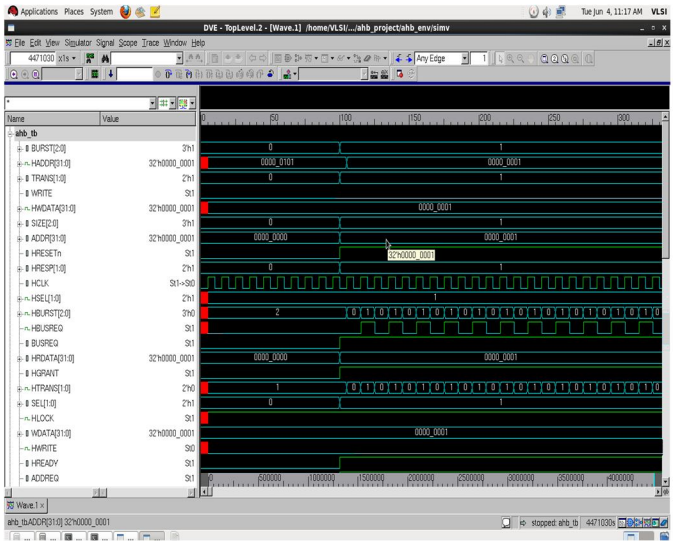


Figure 8 AHB TOP for simple Transfer

Simulation result consists of a simple transfer of bit and Code coverage and Functional coverage using SystemVerilog are observed. Read/write Architecture of SystemVerilog consists of Top module, so output produced at DUT, test, and interface also observed in figure 8, here we send the request from HBUSREQ signal to arbiter if the buses are free then the arbiter granted the signal using HGRANT signal, then HREADY signal is ready to transfer the data from master to slave means read/write operation performed

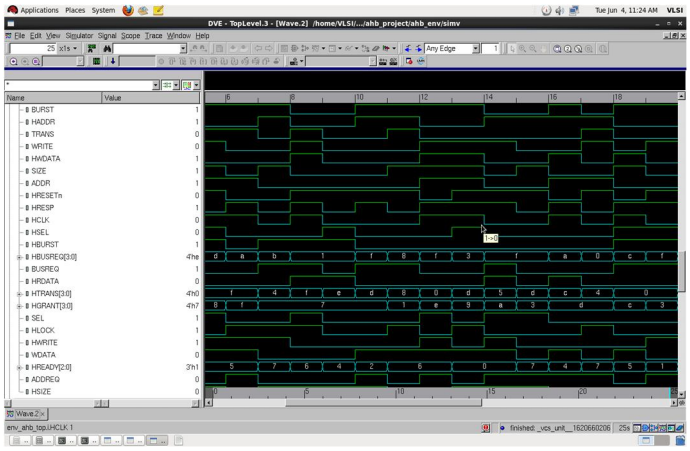


Figure 9 AHB Interface using SystemVerilog

The interface using SystemVerilog waveform is shown in figure 9, the AHB interface is the interfacing between the master and slave and it is a complete verification environment of the AHB protocol, which includes AHB components like master, Arbiter, and slave.

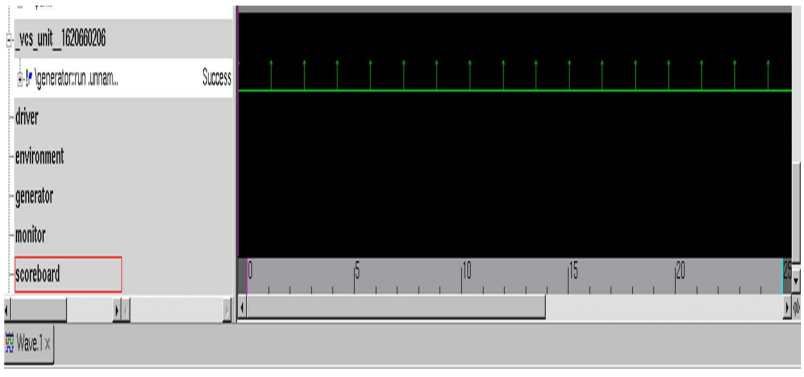


Figure 10 AHB VCS Generator units

The AHB VCS generator units are shown in Figure 10, the VCS generator units for AHB master means all Components driver, environment, generator, monitor and scoreboard are successfully executed.

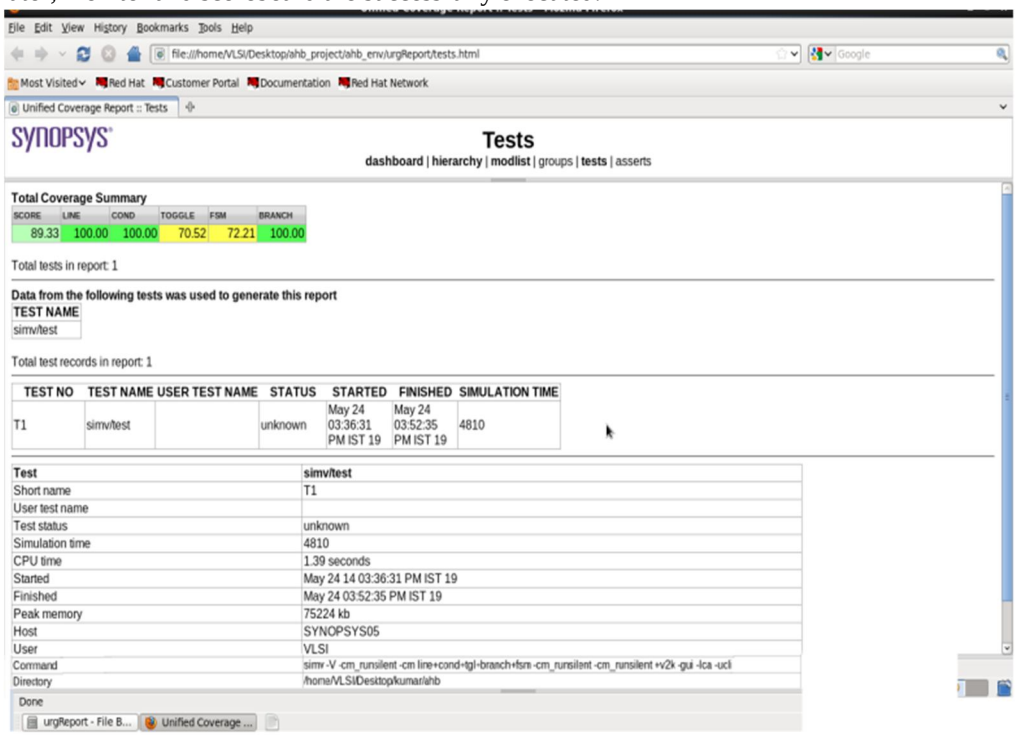


Figure 11 total coverage report using SystemVerilog

The total coverage report using SystemVerilog is shown in figure 11, the total coverage summary of the SystemVerilog Score is 89.33% obtained, line coverage 100%, conditional coverage 100%, toggle coverage 70.52%, FSM coverage 72.21% and branch coverage is 100% obtained.

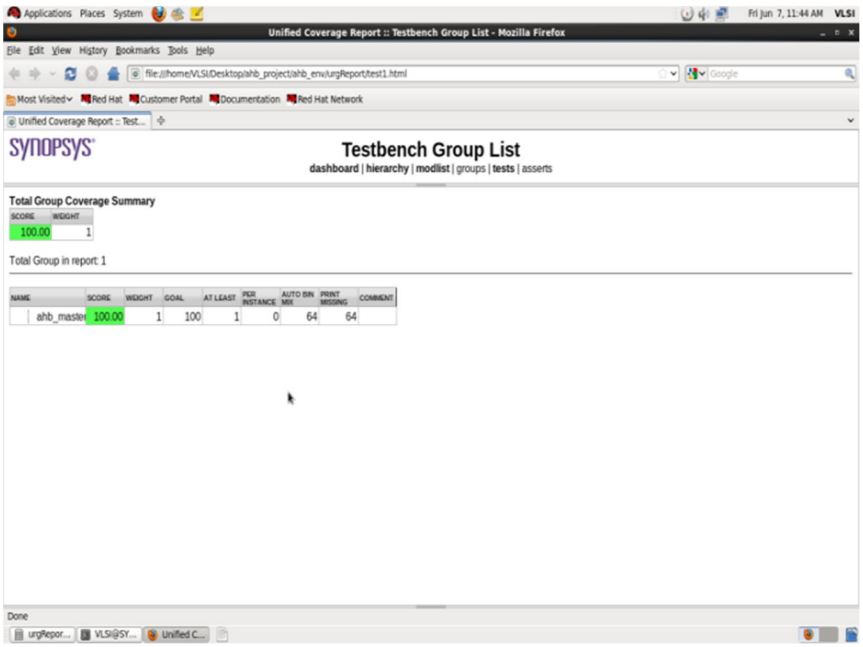


Figure 12 Functional Coverage Report Using SystemVerilog

The functional coverage report using SystemVerilog is shown in figure 12, the total functional coverage is 100% obtained.

V. PROPOSED WORK

Table 4.2 Comparison of implementation of AHB master using different platforms.

Parameters	AHB master core using Verilog (Proposed work)	AHB master core using SystemVerilog (Prposed work)
Tool used for simulation	Synopsys VCS compiler	Synopsys VCS compiler
Code coverage	70.10%	89.33%
Functional coverage	0	100%

The comparison of implementation of AHB master using different platform is shown in table 4.2, here 70.10% of code coverage is obtained by using Verilog and the tool is used for simulation is Synopsys VCS compiler. The code coverage is 89.33% and functional coverage 100% is obtained by using SystemVerilog, and the tool used for simulation is Synopsys VCS compiler.

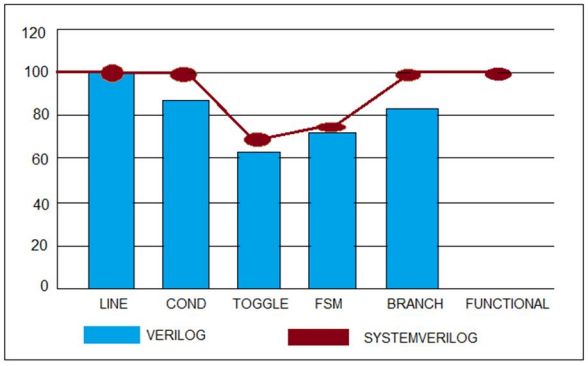


Figure 13 plot of coverage vs. code coverageby Verilog and SystemVerilog

The coverage versus code coverage by using Verilog and SystemVerilog is shown in figure 13, in the graph BLUE lines are shown Verilog coverage and RED lines are SystemVerilog coverage and this graph consist of line coverage, conditional coverage, Toggle coverage, FSM coverage, Branch coverage and Functional coverage. We obtained 100% functional coverage using SystemVerilog.

VI. FUNCTIONAL COVERAGE OF AHB MASTER

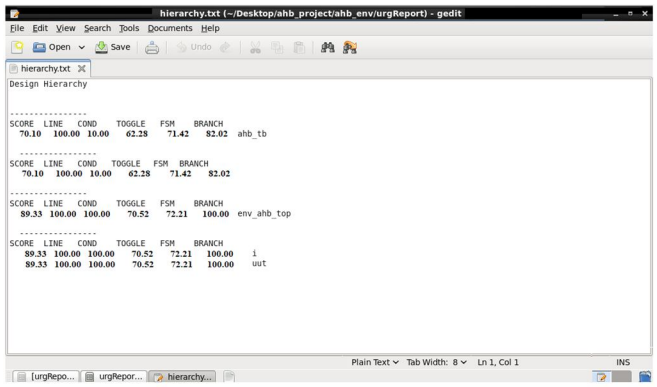


Figure 14 Overall functional coverage report of AHB master

The overall functional coverage of AHB master design by using SystemVerilog is shown in figure 14. All the coverage of AHB master such as line coverage 100%, conditional coverage 100%, toggle coverage 70.52%, FSM coverage 71.21% and branch coverage 100% obtained. The total functional coverage is 100% achieved.

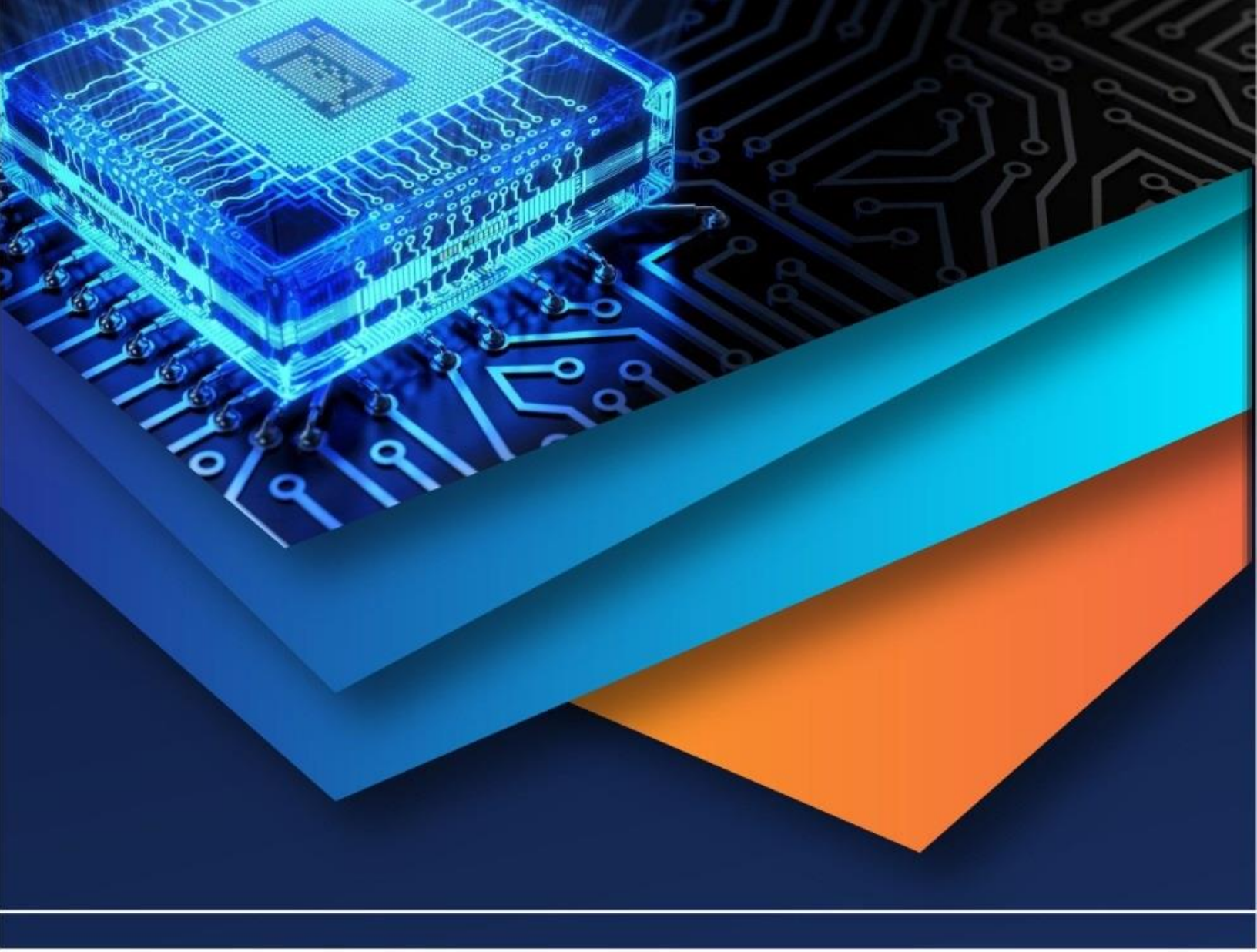
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue VI, June 2019- Available at www.ijraset.com

VII. CONCLUSION

In this paper, the AMBA AHB Master, is designed by using Verilog and verification by SystemVerilog, both the codes are simulated on the Synopsys VCS tool. The proposed method is verification of the entire design of AHB Master was created using SystemVerilog. The proposed verification environment comprised of the base packet, monitor, driver, DUT, interface, and the scoreboard was implemented by using OOP concepts, the functionality of the design verified by the SystemVerilog verification environment. The implementation of whole AMBA AHB master has been completed.

REFERENCES

[1] Keerthana Keerthu "The Design and Verification of AMBA AHB Protocol by using System Verilog" Volume 7,Issue 4, April 2018.
[2] Sweta Bevoor, Prashanth N Gowda, "Design of High Performance Master/Slave Memory Controller for Grey scale Image transfer" (RTEICT), May 19, 2017.
[3] Anshu Gaur, Piyush Sharma, Shiv Pratap Pandey, "HDL and Timing Analysis of AMBA AHB on FPGA Platform" (RDCAPE-2017).
[4] Pallavi Kumari Gautam , Naveen Upadhyay "A Novel Arbitration Technique of AMBA AHB" IJDACR, Volume 4, Issue 7, February 2016.
[5] Prince Gurha, R.R.Khandelwal, "SystemVerilog Assertion Based Verification of AMBA-AHB" (ICMETE-2016).
[6] Anurag Das, Rajat Arora, TVS RAM, "Efficient Baseband Processing System for DVB-RCS to DVB-S2 Onboard Processing Satellite. (Sept 21-24-2016).
[7] Prince Gurha, Dr.(Mrs.) R. R. Khandelwal, "Verification of Advanced High Performance Bus Arbiter using System Verilog Assertion" (IJETT) – Volume 37 Number 5 - July 2016
[8] Poorani.P, Ms. Vijayshree. BM. E "Implementation Of Ahb Bus Protocol For System On Chip Security" (ICECS '2015).
[9] Soiurin Shah, Nirav Patel. "SystemVerilog Based AMBA AHB Verification Environment" (May 2014).
[10] Sreehari S, Jaison Jacob, "AHB DDR SDRAM Enhanced Memory Controller" (ICACCS -2013), Dec. 19 – 21, 2013, Coimbatore, INDIA.
[11] Akshay mann, Ashwini kumar, "Assertion based verification of AMBA-AHB using Synopsys VCS" (November-2013).



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY