iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 9 **Issue:** VI **Month of publication:** June 2021

**DOI:** https://doi.org/10.22214/ijraset.2021.36203

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: 7.429*
*Volume 9 Issue VI June 2021- Available at www.ijraset.com*

# Verification of AMBA AHB Protocol using UVM

Janardhana S Y
*Student, Department of Electronics and communication Engineering, PES Collage of Engineering Mandya, India*

***Abstract— The project aims to verify the AMBA AHB protocol by using universal verification methodology is presented in this paper. Advanced high-performance(AHB) is used for communication of on chip bus which support single clock edge operation wider data 32/64/128 bit can be supported. The new verification constructs can be easily reused for the objected-oriented feature of universal verification methodology (UVM). Verification IP is the one which provides a smart way to verify the AHB Components. The advanced verification testbench incorporates the illustrations regarding simulation result are analysed to evaluate the effectiveness of the proposed testbench and functional coverage is to check functionality of the design. The self-checking mechanism using assertions improves the quality of UVM check by shortening time to debug and reducing time to cover for the in-depth understanding of test case output.***
***Keywords— AMBA, AHB, UVM, functional coverage, assertion***

## I. INTRODUCTION

The tremendous progress of the VLSI Technology enables the integration of more than several million transistors in a single chip to make a system-on-chip this has made verification the most critical bottleneck of the chip design flow roughly 70 to 75 percent of the design is spent in verification. verification is the process of determining whether a system conforms to its specification the process should provide evidence that a system behaves like intended by its functionality and like described by its design specification AHB protocol on chip bus uses universal verification methodology for verification it combines the advantages of OVM and VMM two verification methods it has become the latest verification standard in the chip verification industry UVM offers base class libraries for generating and organizing testbench methodology outlines set of rules and procedures for doing things in a systematic way the main benefit of UVM are wide-ranging base class library support and can perform coverage-driven constraint random verification fully supported by major tool vendors maintained by accellera.

## II. UVM HIERARCHY

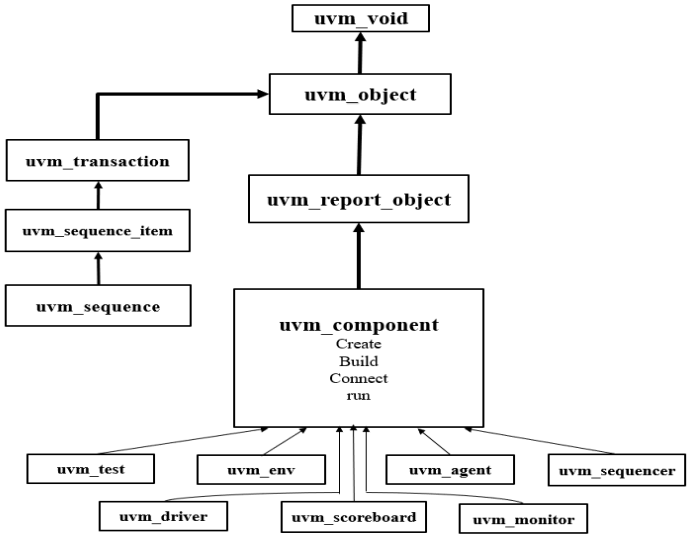



Fig.1. UVM Hierarchy

The purpose of building a UVM verification platform is to verify that the functionality of the design under test conforms to the design specifications. The UVM verification platform consists of a series of verification components. For all classes in UVM, there is a common base class, uvm_void, which is an empty shell with neither data members nor member functions. uvm_object is extended from the uvm_void class and serves as the base class for all entities in UVM. Then uvm_object is divided into two

©IJRASET: All Rights are Reserved

branches, which evolve into uvm_component through uvm_report_object along the way and various components thereafter uvm_sequence_item and uvm_sequence through uvm_transaction along the way. The classes evolved from these two lines are significantly different. uvm_component exists all the time in the entire simulation, while uvm_transaction can be transient in the entire platform. The two cooperate with each other, and the specific inheritance relationship is shown in Fig.1. All the nodes of uvm tree are composed of uvm_component. Only the derived class based on uvm_component can be the node of UVM tree. The leftmost branch or the class directly derived from uvm_object is not likely to reappear in the form of nodes on the UVM tree. Usually used verification components and their functions in the UVM verification platform are shown in UVM Testbench Architecture.

## III. UVM TESTBENCH ARCHITECTURE

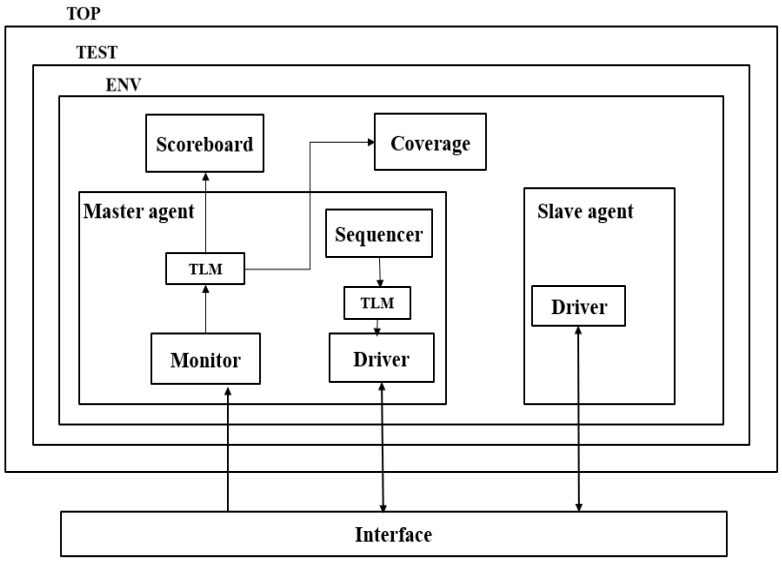



Fig. 2. UVM Testbench Architecture

The UVM testbench is developed for the verification and analysis components derived from the UVM base class libraries. The interface is used to define the signals of DUV. It is through this interface that Driver and DUV communicates each other. The virtual interface handle is set to interface instance in top harness module. The testbench generates random vectors and drives the DUV. The stimulus applied as well as DUV response is monitored for checking purposes to verify the DUV functionality. The testbench also uses the coverage mechanism to report functional coverage as a measure to evaluate the progress of verification process. The UVM components along with their functionality are described below:

### *A. Test*

The test instantiates the environment and achieve connection from sequence to sequencer by using start method.

### *B. Environment*

All the UVM verification components are grouped together inside a single container called environment. The verification components can be configured in any arbitrary way by defining an environment. A UVM testbench can have multiple such environments.

### *C. Agent*

The sequencer, driver and monitor are encapsulated into a single entity called agent. Agents can either be active or passive depending on whether it instantiates all the components and stimulate the DUV or only the monitor to track the DUV responses respectively.

### *D. Sequencer*

The sequencer runs the sequence. The sequences generated are send to the driver through sequencer whenever the driver demands for it.

©IJRASET: All Rights are Reserved

*E. Driver*

The driver converts the data fields of sequence items into pin level transactions to drive the DUV using a virtual interface.

*F. UVM Sequence Item*

The data properties such as control, payload, configuration and analysis information of the random stimuli to be applied to the DUV are defined by the sequence item. The input data properties are declared as random variables for constrained random testing.

*G. Sequence*

The sequence items are generated and randomized by the sequence before transmitting them to the driver. Sequences can be test specific.

*H. Monitor*

The monitor tracks the activity at pin level of DUV, converts them into transfer level transactions and send them to the scoreboard and coverage collector using the TLM analysis ports.

*I. Scoreboard*

The scoreboard is used to compare whether the DUV response matches with the expected behaviour. It decides whether a test case has passed or failed.

*J. Coverage*

The coverage keeps track of all hits occurring on the DUV signals that are specified as cover points in the coverage model during simulation.

*K. Assertion*

Validates the protocol behaviour at DUT interface and DUT inside contents.

## IV. OPERATION OF AHB

*A. AHB Master*

A master initiates the operation by transferring the address and control information. These control information gives the whether to perform write or read operation.

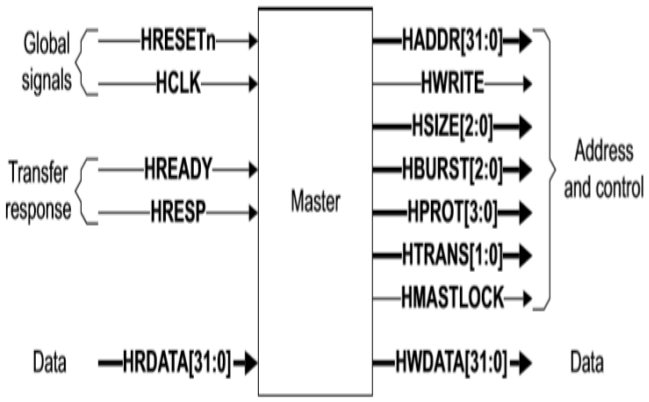



Fig. 3. AHB Master

*B. AHB Slave*

The slave responds write and read operation to the address and control information which is sent by master. When the HREADY signal is high it indicates the particular transfer is completed, if it is low it means that its performing operation so at this time slave can request the master to place the wait states in order to complete the operation. The RESP signal gives the information whether the operation is OKAY or ERROR/Wait, if the RESP is low it represents the Okay else Error/wait.

©IJRASET: All Rights are Reserved

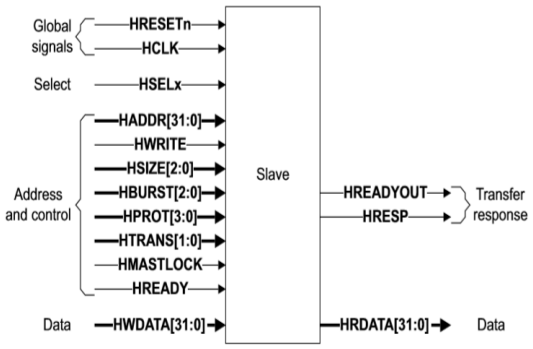



Fig. 4. AHB Slave

## V. RESULTS

Questa Sim 10.4e tool is used to simulate and verify the AHB protocol and the simulation results are analysed to evaluate the effectiveness of the proposed testbench. Functional coverage and assertion are achieved as per the verification plans.

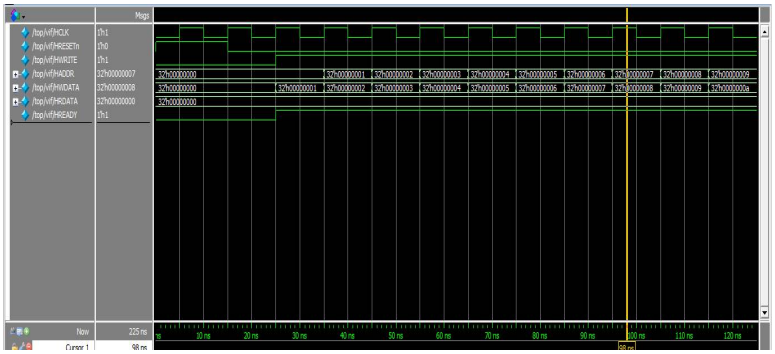


Fig. 5. Write transfer simulation result

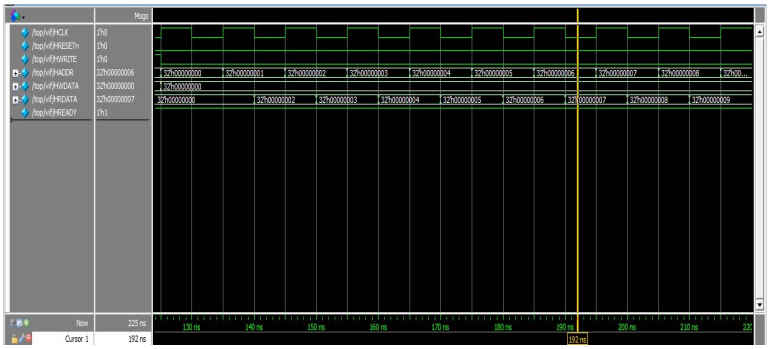


Fig. 6. Read transfer simulation result

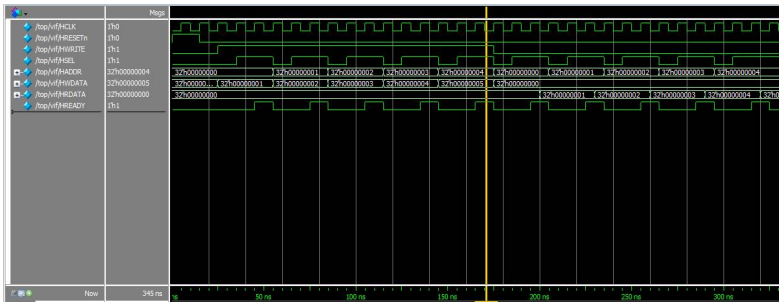


Fig. 7. Write and read transfer with wait simulation result

©IJRASET: All Rights are Reserved

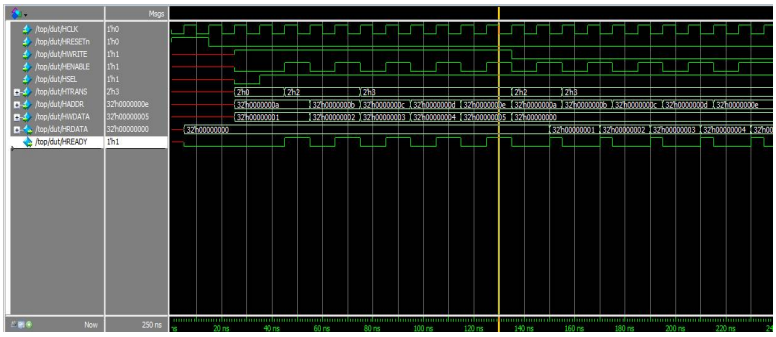


Fig. 8. Transfer type simulation result

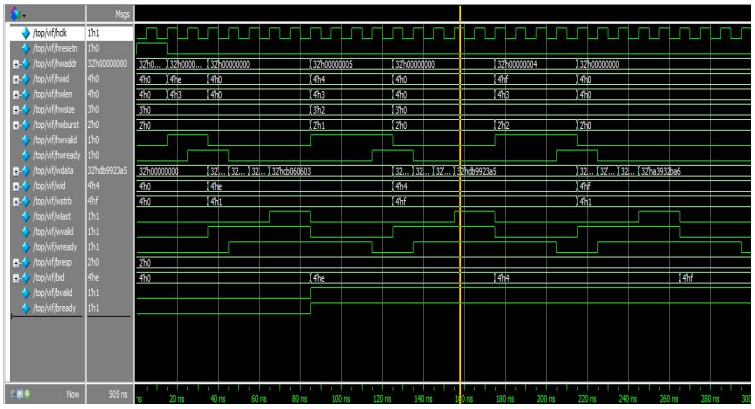


Fig. 9. Burst write transaction simulation result

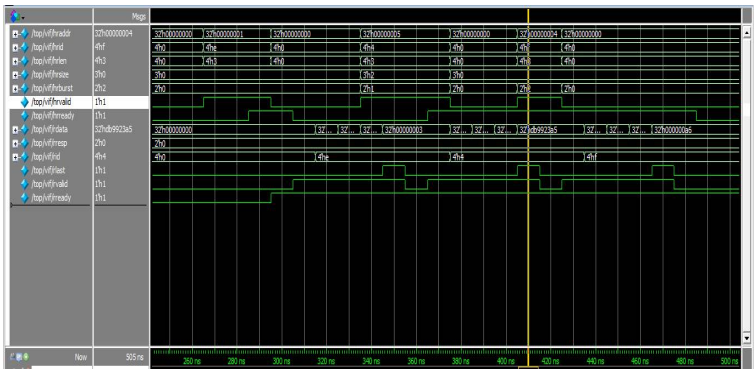


Fig. 10. Burst read transaction simulation result

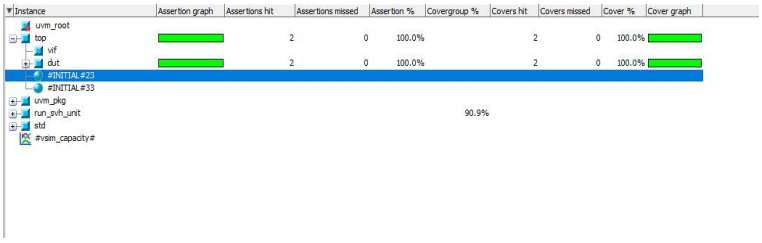


Fig. 11. Coverage and Assertion report display

## VI. CONCLUSIONS

In this project verification of AMBA AHB Protocol has been verified by using the UVM methodology. We have used UVM to put up a verification environment which focuses on functional verification in order to check the correctness of the AHB. The test bench accomplished complete functional verification of AHB and achieved 100% coverage through systematic execution of

©IJRASET: All Rights are Reserved

relevant testcases and assertion are achieved. The simulation waveform demonstrates the successful exchange of data between AHB master and AHB slave.

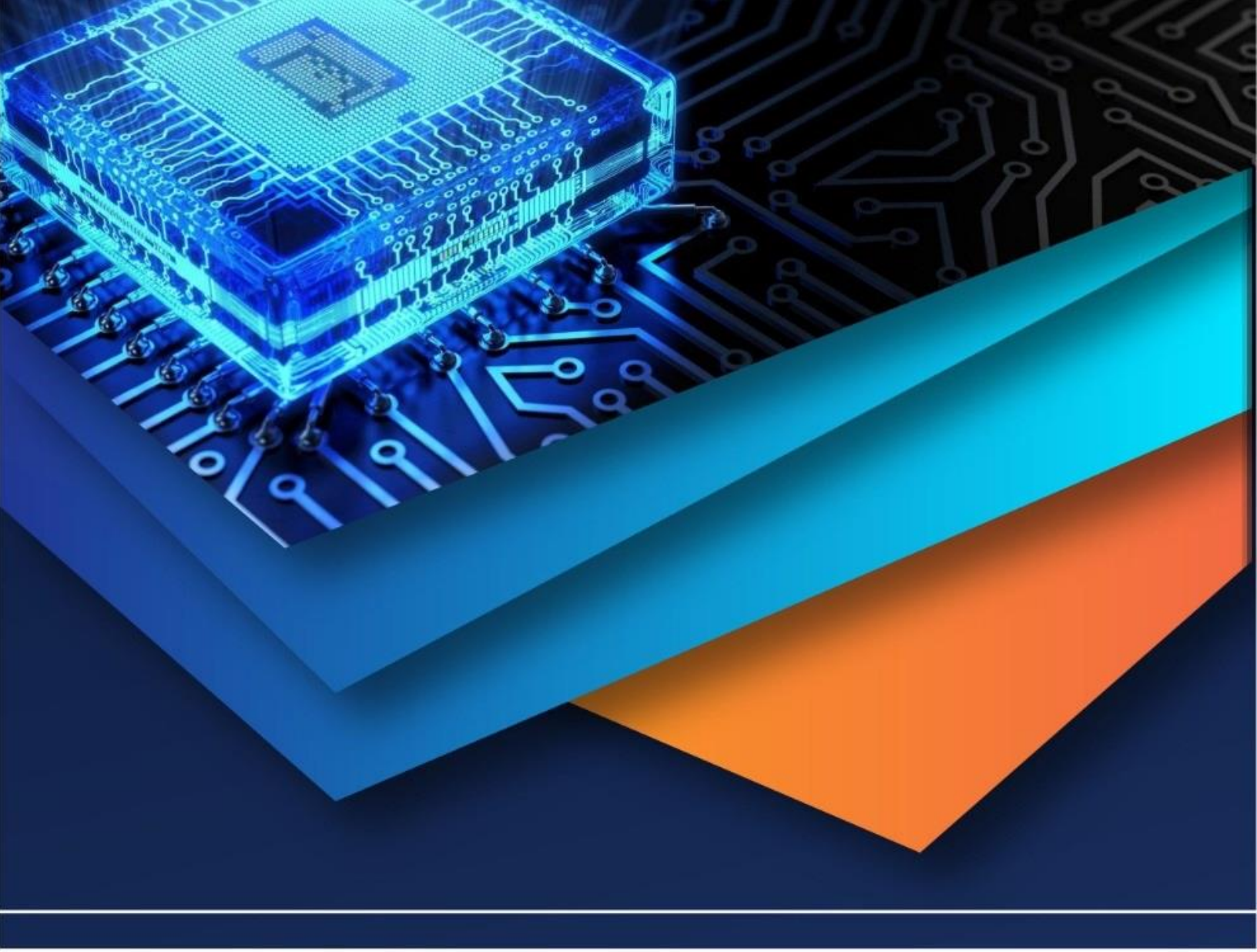
## ACKNOWLEDGMENT

I specially thank Mr. M Subramanyam, Associate Professor Department of Electronics and Communication Engineering, PES Collage of Engineering Mandya for guiding this project. And for his consistent technical guidelines and moral support. I also extend my Gratitude and thanks to all the Teaching and Non-Teaching Staff Dept. of ECE.

## REFERENCES

[1] Gurha, P. and Khandelwal, R.R., 2016, September. "System Verilog Assertion Based Verification of AMBA-AHB". In 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE) (pp. 641-645). IEEE.
[2] Han Ke,Deng Zhongliang,Shu Qiong."Verification of AMBA Bus Model Using System Verilog" The Eighth International Conference on Electronic Measurement and Instruments. 1-4244-1135-1/07/$25.00 ©2007 IEEE.
[3] M. You and G. Song, "System Verilog-based verification environment using System C custom hierarchical channel," 2009 IEEE 8th International Conference on ASIC, Changsha, Hunan, 2009, pp. 1310-1313.
[4] Ni, Wei, and Jichun Zhang. "Research of reusability based on UVM verification." In 2015 IEEE 11th International Conference on ASIC (ASICON), pp. 1-4. IEEE, 2015.
[5] Francesconi, J., Rodriguez, J.A. and Julian, P.M., 2014, July. "UVM based testbench architecture for unit verification". In 2014 Argentine Conference on Micro-Nanoelectronics, Technology and Applications (EAMTA) (pp. 89-94). IEEE.
[6] AMBA® 3 AHB Protocol Copyright © 2001, 2006 ARM Limited. All rights reserved. ARM IHI 0033A
[7] Mulani, "soc Level Verification Using System Verilog" Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on 16-18 Dec.2009 pp.378-380
[8] Li, Yangyang, Wuchen Wu, Ligang Hou, and Hao Cheng. "A study on the assertion-based verification of digital IC." In 2009 Second International Conference on Information and Computing Science, vol. 2, pp. 25-28. IEEE, 2009.
[9] Universal Verification Methodology 1.1 User's Guide May 18, 2011 http://www.accellera.org
[10] Vineeth B, Dr. B Bala Tripura Sundari"UVM Based Testbench Architecture for Coverage Driven Functional Verification of SPI Protocol"978-1-5386-5314-2/18/$31.00 ©2018 IEEE
[11] Pavithran, T. M., and Ramesh Bhakthavatchalu. "UVM based testbench architecture for logic sub-system verification." In 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), pp. 1-5. IEEE, 2017.

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)