International Journal of Advanced Science and Technology
Vol. 29, No. 8, (2020), pp. 4968-4975

Development of Verification IP for AMBA AXI 5.0 Protocol

Praveen Chandrashekhar Metri
Department of Electronics and Communication Engineering
RV College of Engineering, Banglore, Karnataka, India
Email- praveen.metri7@gmail.com

Subrahmanya K N
Department of Electronics and Communication Engineering
RV College of Engineering, Banglore, Karnataka, India
Email- knsubrahmanya@rvce.edu.in

Abstract- With the recent development in IP based System on Chip (SoC) design, SoC design has become more complex with new IPs or modules used to integrate and chips are designed with goal of less time to market. So the interconnection of IPs on Soc is made with standardized signal bus architecture, standardized bus architecture has become major integration in Soc design which helps in minimize design time. In a chip design life cycle verification takes 70% -80% of time to market. It has become very complicated to verify SoC design with conventional methods of verification as they lack flexibility in environment for verification. Verification Intellectual Properties do come in handy while verifying the standard protocol. In this paper, work focuses on Developing Verification IP (VIP) for AMBA AXI 5.0 protocol achieving successful transactions of read and write according AXI specification. The VIPs are developed using recent methodology called Universal Verification Methodology(UVM). All simulations are done in Cadence Incisive and waveforms are observed and analyzed in Simvision. Incisive Metric Center(IMC) is used for coverage analysis.

Keywords –VIP, SoC, SV, UVM, AMBA, AXI, Cadence Incisive, Simvision, IMC, Functional Coverage.

I. INTRODUCTION

AMBA (Advanced Microcontroller Bus Architecture) is open standard for communication and management of the functional blocks in SoC, provide different on-chip communication protocols like CHI (Coherence Hub Interface), AXI (Advanced eXtensible Interface), ACE (Advanced Coherency Extension), AHB (Advanced High-performance Bus), APB (Advanced Peripheral Bus) developed by ARM (Advanced RISC Machine) [1]. Flexibility of AMBA protocols is IP reuse for different SoC design with different area, power and performance requirements. As the AMBA protocols are widely used open standards which ensures compatibility between IPs of different suppliers for the SoC , with compatibility it enables low friction integration and reuse of IP which catalyze the faster time to market.

The AMBA AXI protocol specification [3] is defined to implement high frequency, high bandwidth interface across wide variety of applications in embedded, automotive and cellphones. It does not require complex bridge implementation for different peripheral devices. AXI 5.0 protocol include some of new feature [3] which extends previous versions and is compatible to complement CHI[2].

Conventional verification methods involves writing test bench in verilog or VHDL by driving stimulus to them but as design complexity increases the methods become tedious and insufficient to verify the chip. System Verilog has advanced features[4] which helps in developing potential verification environment. To reduce verification effort and increase efficiency of verification need a methodological approach. In this paper we use UVM a SV based methodology approach which helps in developing verification environment which can be reused, reliable and robust. VIPs for AXI are developed using UVM methodology.[5], [6], [7], [8], [9], [10] have been reported verification of AXI prior version to AXI 5.0 using different methodologies. [11], [12], [13], [14], [15], [16] have been reported verification of peripheral bridges which bridges between AXI protocol and other peripheral protocols.

The remaining sections are discussed as follows. Section II discusses on development of VIPs based on UVM methodology and proposed UVM framework. Section III consists of simulation result and coverage report and the paper is concluded with section IV.

ISSN: 2005-4238 IJAST

II. AXI-UVM Testbench Development

Testbench development is creating verification environment for AXI Protocol independent of test cases. AXI testbench and VIPs developed using UVM methodology. This verification environment provide freedom for configuration according to the testcases.

2.1 Architecture of AXI Protocol

AXI protocol defines five channels.

1. Write Address Channel: The channel transfers address and associated control signal information for write transaction.
2. Write Data Channel: The channel carries data and bus which can 8, 16, 32, 64, 128, 256, 512, 1024 wide.
3. Write Response Channel: This channel ensures by slave to reply to write transaction by master to indicate the transfer completion
4. Read Address Channel: The channel transfers address and associated control signal information for read transaction.
5. Read Data Channel: The channel carries data / read response from the slave to master and data bus which can 8, 16, 32, 64, 128, 256, 512, 1024 wide.

5 channel which are independent defined by AXI architecture and concept of ID is introduced to have multiple transaction simultaneously. Figure 1 and Figure 2 describes the write and read channel architecture respectively. Each channel is independent and channel has two way handshaking mechanism has set several information signals and dependencies among handshaking signal, two way handshaking signals are VALID and READY. Figure 3 and Figure 4 describes the write transaction dependencies and read transaction dependencies respectively.

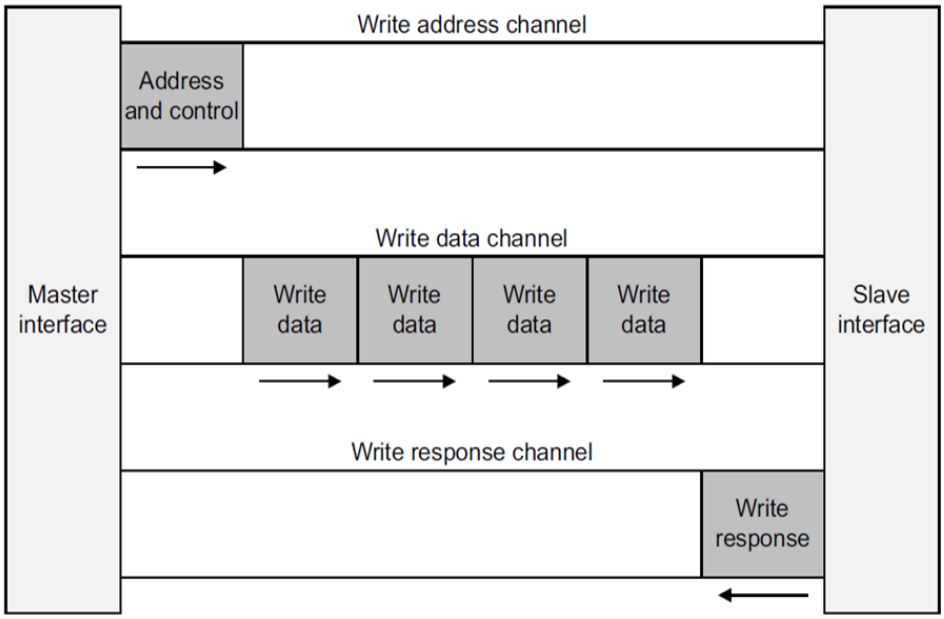

Figure 1. AXI Write Channel Architecutre

ISSN: 2005-4238 IJAST

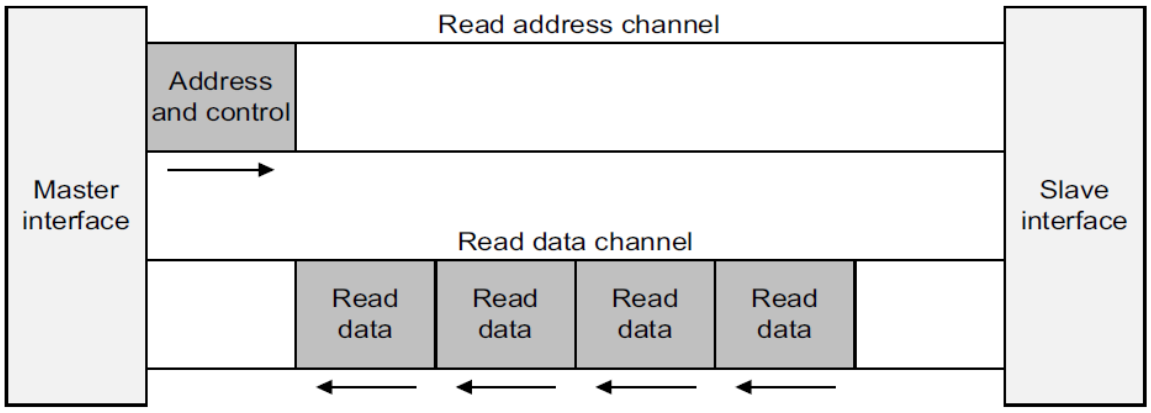

Figure 2. AXI Read channel Architecture

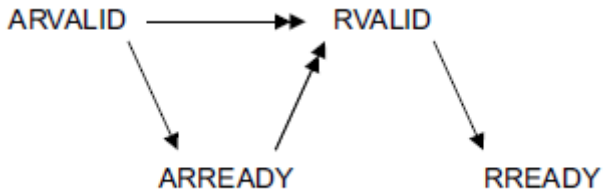

Figure 3. Read Transaction Handshake

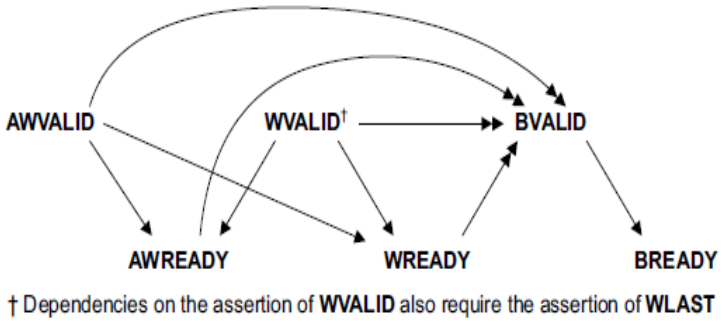

Figure 4. Write Transaction handshake

2.2. Methodology

The UVM is SV based standardized methodology for SoC functional verification that is used for creating testbench components and Transaction level model(TLM) [11] for interconnection between testbench components. The standardization of methodology for creating testbench enables users to implement modules that are portable and compatible thereby enforcing aspects of interpretability. Verification component is sub module or full system, ready to use configurable verification environment for an interface protocol. Figure 5 shows UVM methodology testbench with top module comprises of sequence item, sequencer, driver, monitor, agent, scoreboard, environment and test.

ISSN: 2005-4238 IJAST

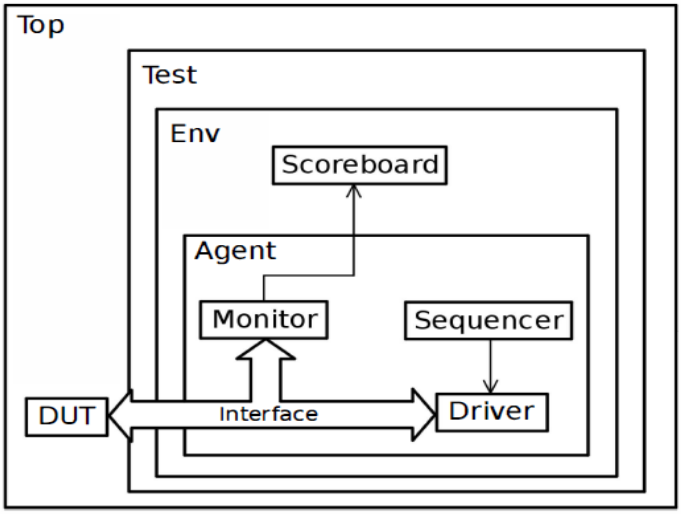

Figure 5. UVM Testbench Architecture

A. Agent

The component which connects a driver, sequencer and monitor. Data transfers take place at transaction level which are defined by sequence items and arrangement such items are called sequences. Sequencer transfers the sequences to the driver. Driver is a component repeatedly takes transactions form sequencer and drives on to DUT through interface by transforming transaction to signal level on to interconnect. Monitor is an entity which monitors signals which are driven into DUT send it analysis port for different component like coverage and scoreboard.

B. Scoreboard

It is testbench component which has self-checking mechanism to check correctness of DUT. It uses the response from DUT to compare against expected results of DUT called Golden Response Model(GRM).

C. Environment

This component is the top level which mimics the environment in which the DUT should be verified, this can be configured and model the behavior. Detailed description of UVM features can be found in [4].

2.3. Testbench Architecture for AXI.

Development of testbench for AXI protocol starts with defining interface signals, sequence items and then driver logic to implement the design. The driver logic which define write address calculation, write and read operations according to the specification.

Write Address calculation protocol for driver is as shown below,

1. Start Address = AxADDR
2. Number of Bytes = 2 ^ AxSIZE
3. Burst Length = AxLEN + 1
4. Here A stands for the AXI protocol and x stands for W for write and R for read channels.
5. Aligned address= INT(Start address / No.of bytes) * No. of bytes.
6. The equation used to determine the address of the first transfer in a burst
   Address_1 = Start Address.
7. For an INCR burst, and for a WRAP burst for which the address has not wrapped, the following equation determines the address of any transfer after the first transfer in a burst
   AddressN= Aligned Address+(N-1) *No. of bytes
8. For a WRAP burst, the Wrap Boundary variable defines the wrapping boundary
   wrapping boundary =(INT(Start address/No. of Bytes)*Burstlength)) *(No.ofBytes)* (Burst length)

ISSN: 2005-4238 IJAST

Figure 6 shows the proposed testbench developed for the verification of AXI Protocol, in which separate is agent developed for master and slave which drives, monitors, and collects coverage through the analyses port. Interface write and read transfers are achieved by constrained randomization coverage is calculated accordingly.

Write Transaction: The Write Transaction makes use 3 channels associated with it and transaction is done as per the Handshaking mechanism described in protocol.

Read Transaction: The Transaction make use of 2 channels dedicated for this purpose and read transaction is done as per the handshaking mechanism described in protocol

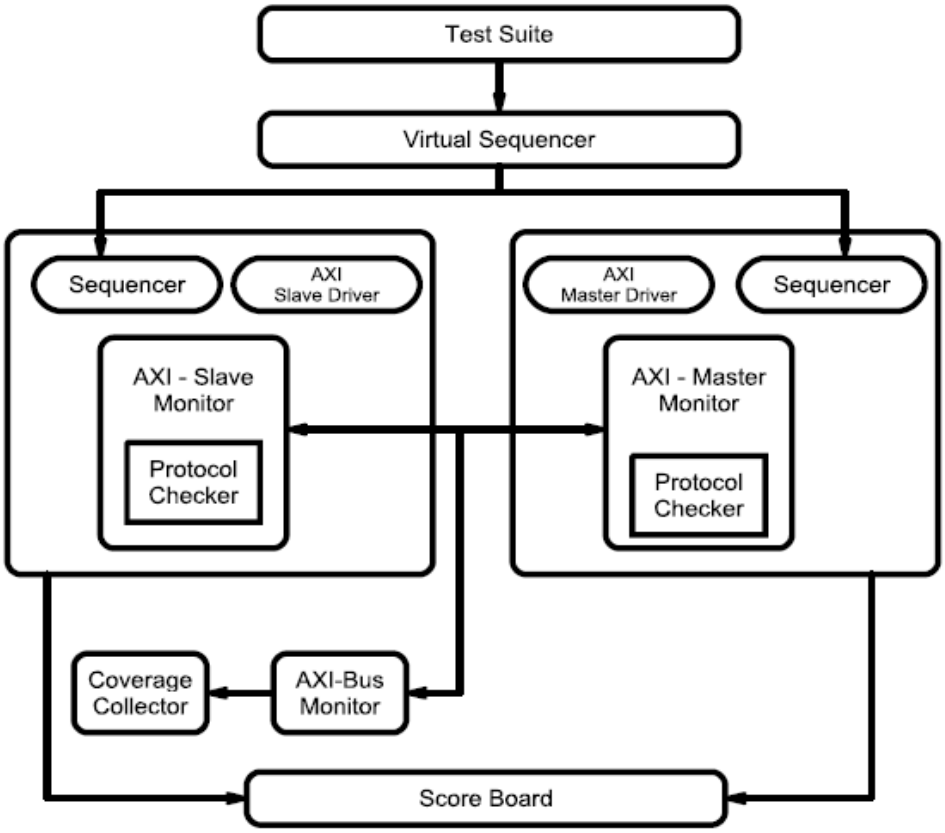

Figure 6. UVM Testbench Architecture

IV. RESULTS AND DISCUSSION

Assertion are used to validate the design's behavior. These are properties that should be true and they also provide information about functional coverage. Figure 7 shows Write channel transaction as master sends four different AWADDR, AWLEN, AWSIZE, AWBURST. As per the inputs given by master, WDATA and WSTRB are driven respectively and accordingly. After the last data of each burst has been driven, WLAST signal is driven high for one clock cycle to signify the end of the burst. Figure 8 shows read channel transaction as master will give ARADDR, ARLEN, ARSIZE, ARBURST. Depending on the these response is driven by the slave to master. In Figure 9, coverage report generated by Integrated Metrics Collector(IMC), which has calculated Functional coverage, code coverage, assertions.

ISSN: 2005-4238 IJAST

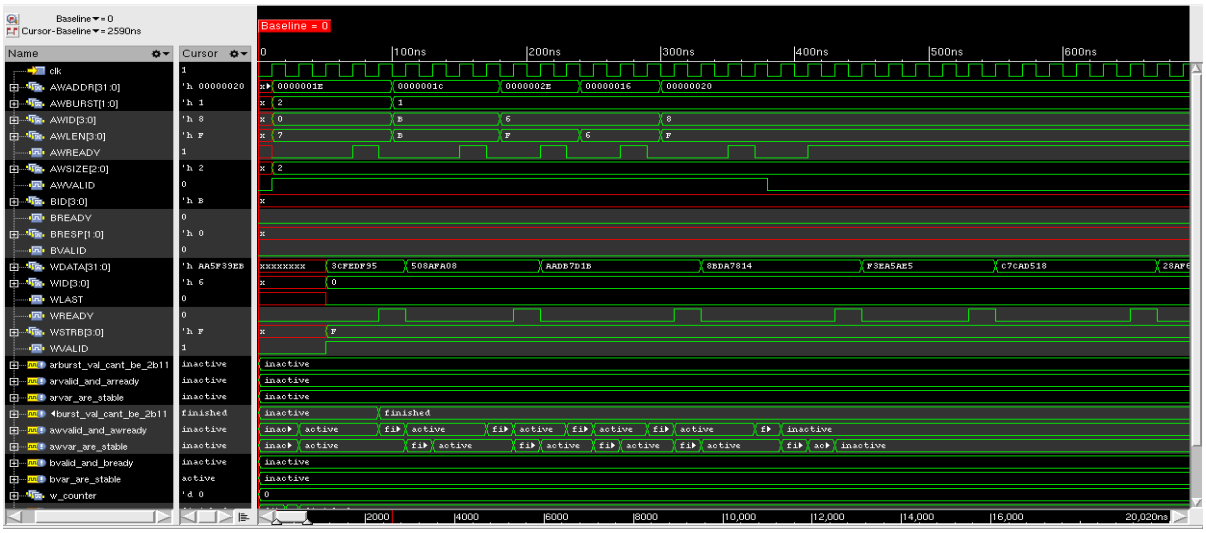

Figure 7. UVM Testbench Architecture

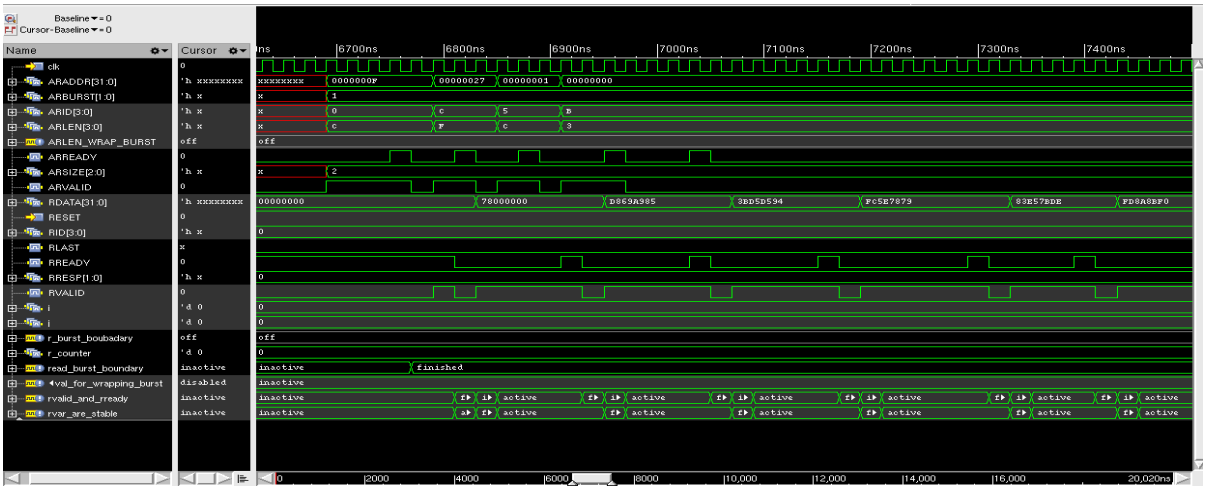

Figure 8. UVM Testbench Architecture

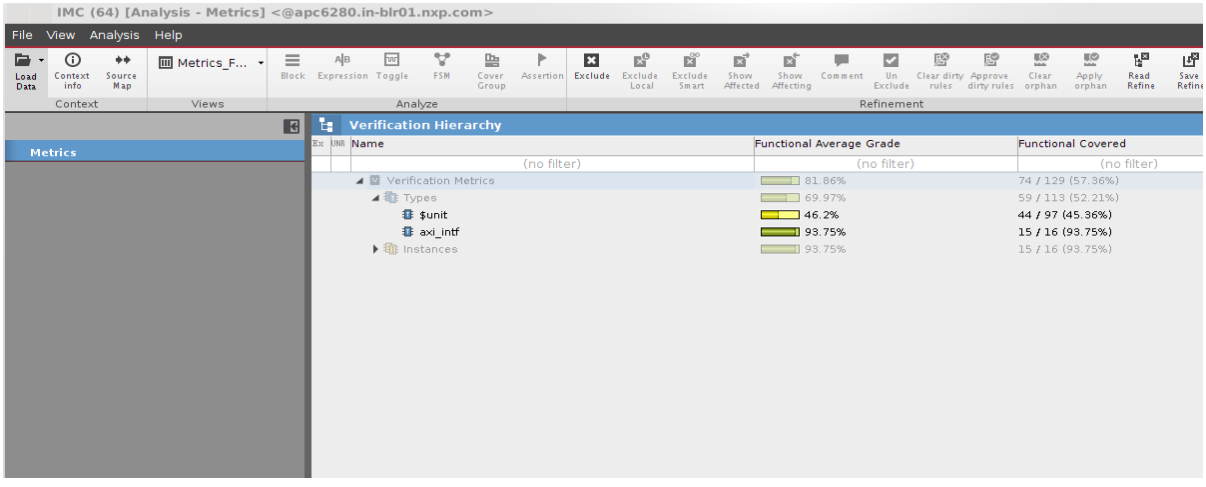

Figure 9. UVM Testbench Architecture

ISSN: 2005-4238 IJAST

Table -1 Coverage

Type	**Total Percentage**
Expression	**100**
Block	**100**
Toggle	**99**
Functional	**93.5**

Table-2 Coverage Parameters

Parameter	**Total Percentage**
Assertion	**30**
Cover points	**8**
Total bins	**52432**
Goal Average per bin	**590**

Table 1 shows the different type of coverage analysis done for the verification of AXI 5.0 Protocol. Table 2 Shows the parameters that are used to ensure proper design of protocol.

IV.CONCLUSION

As verification takes 70% to 80% of the time to market time of product, so if we could reduce this time by developing verification IPs for the standard designs or protocols that are frequently used in System On Chip (SOC) design. By incorporating these IPs while verification of system reduces total verification time by eliminating all the verification plan for design or protocol. As in this project we have taken standard protocol called AXI protocol which is used in all almost SoC design for the on chip communication.

This paper work achieved functional coverage of 94 % and bus utilization factor 95% with more number assertion nearly 30 when compared to previous work which had 10. All standard thresholds of Code Coverage and Functional Coverage defined in the verification plan were met, meaning that the present framework can be reuse or replicated in future analyzes.

As this project implement verification IP assuming direct AXI connection between master and slave. But Design includes low frequency peripherals such as APB or other on chip protocol as AHB, So consider these possibilities we can include verification IPs of bridges for all the AMBA protocols.

REFERENCES

[1] AXI, ”AMBA™ Specification(Rev 5.0)”.
[2] AXI, ”AMBA CHI Specification”.
[3] AXI, ”AMBA AXI™ Specification(Rev 5.0)”.
[4] MENTOR, “UVM (1.2) cookbook”
[5] Xiaokun Yang and Jean H. Andrian, “A High-Performance On-Chip Bus (MSBUS) Design and Verification,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* VOL. 23, NO. 7, JUL. 2015.
[6] J. Lee, “Design methodology for on-chip bus architectures using system on-chip network protocol,” *IET Circuits, Devices Syst.*, vol. 6, no. 2, pp. 85–94, Mar. 2012.
[7] Nikhil Gaikwad and Vijay.N.Patil “Verification of AMBA AXI on-chip Communication Protocol,” *Fourth International Conference on Computing Communication Control and Automation*, 2018.
[8] M Prasanna Deepu,” Validation of Transactions in AXI Protocol using System Verilog,” *International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM),*2015.
[9] Renduchinthala H, S. S. Prasad, and Ch Santhi Rani. “Development of VIP for AMBA AXI-4.0 Protocol.” *Indian Journal of Science and Technology*, vol. 9, no. 48, 2016.
[10] Rini Sebastian, Silpa Rose Mary, Gayathri M, Anoop Thomas,” Assertion BasedVerification of SGMII IP core incorporating AXI Transaction Verification Model”,*International Conference on Control, Communication & Computing India (ICCC)*,November 2015.
[11] Gayathri M, Rini Sebastian, Silpa Rose Mary, Anoop Thomas,” A SV-UVM framework for Verification of SGMII IP Core with reusable AXI to WB Bridge UVC”, *3rd International Conference on Advanced Computing and Communication Systems (ICACCS),* Jan. 22 , 2016.

ISSN: 2005-4238 IJAST

[12] Frank Plasencia-Balabarca, Edward Mitacc-Meza, Mario Raffo-Jara and Carlos Silva-Cárdenas," A Flexible UVM-Based Verification Framework Reusable with Avalon, AHB,AXI and Wishbone Bus Interfaces for an AES Encryption Module", *International Journal of VLSI design & Communication Systems (VLSICS)* Vol.3, No.2. 2012.

[13] Armando Sánchez-Peña, Pedro P. Carballo, Luz García and Antonio Núñez," VIPACES,Verification Interface Primitives for the Development of AXI Compliant Elements and Systems", *Proceedings of the 9th EUROMICRO Conference on Digital System Design (DSD'06),*2006.

[14] Nishit Gupta, Sunil Alag," NTRP: Novel Approach for DUT Testing based on Nonintrusive Timing Randomization Probes using SystemC Verification Library", *International conference on information Technology(InCiTe)-The Next Generation IT Summit,*2016.

[15] Shaila S Math, Manjula R. B, S.S. Manvi," Data Transactions on System-on-Chip Bus Using AXI4 Protocol", *International Conference on Recent Advancements in Electrical, Electronics and Control Engineering,*2011.

[16] Chenghai Ma, Zhijun Liu, Xiaoyue Ma," Design and Implementation of APB Bridge based on AMBA 4.0", *IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis*, October 2013.

ISSN: 2005-4238 IJAST