VERIFICATION OF DRIVER LOGIC USING AMBA-AXI UVM

Bijal Thakkar[1] and V Jayashree[2]

[1]Research Scholar, Electronics Dept., D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji, Maharashtra, India.
[2]Professor, Electronics Dept ., D.K.T.E. Society's Textile and Engineering Institute, Ichalkaranji, Maharashtra, India.

ABSTRACT

Advanced Extensible Interface (AXI) is the most commonly used bus protocols in the day-to-day because of its high performance and high-frequency operation without using complex bridges. AXI is also backward-compatible with existing AHB and APB interfaces. So verification of driver logic using AMBA-AXI UVM is presented in this paper. The AXI is used for multiple outstanding operations which is only possible in the other protocol but it is possible in AXI because it contains different write address and data channels and AXI also supports out of order transfer based on the transaction ID which is generated at the start of the transfer. The driver logic for the AXI has been designed and implemented using the Universal Verification Methodology (UVM).The signaling of the five channels such as write address, write data, write response, read address, read data channel of AXI protocol are considered for verification. According to the AXI protocol,the signals of these channels are driven to the interconnect and results are observed for single master and single slave. The driver logic has been implemented and verified successfully according to AXI protocol using the Rivera Pro. The results observed for single master and single slave have shown the correctness of AMBA-AXI design in Verilog.

KEYWORDS

AMBA(Advance Microcontroller Bus Architecture),AXI(Advanced Extensible Interface),UVM(Universal Verification Methodology),channel.

1. INTRODUCTION

AXI stands for (Advanced Extensible Interface) and it is an On-Chip communication protocol. It is a part of the Advanced Microcontroller Bus Architecture (AMBA) developed by ARM (Advanced RISC Machines) company. It provides high-frequency operation without using complex bridges to meet the interface requirements of a wide range of components. It is suitable for memory controllers with high initial access latency. AXI-UVM has separate address/control and data phases. It supports for unaligned data transfers using byte strobes instead of supporting burst based transactions with only start address issued. Also supports multiple outstanding addresses without order response. Because of these features, AXI is the most commonly used on chip bus protocols in the day-to-day high performance System On Chip (SOC's).

In reference on AMBA AXI protocol specification, the signalling information and handshaking signals and working of protocol from AMBA-AXI protocol has been reported[1].Gayathri M, and RSA have reported, an efficient SV-UVM framework for the verification of Serial Gigabit Media Independent Interface (SGMII) IP core, a single lane 1.25 Gbps data rate interface between Ethernet Media Access Control (MAC) and Physical (PRY) layer[2].Rini Sebastian ,SGA reported on benefits associated with Assertion Based Verification (ABV) which was successfully

applied to multiple levels of design abstraction . All simulations were carried out in NCsim and waveforms were analysed in Simvision. All the assertions carried out were ensured without any failure and shown the coverage capability of ABV through a case study on Serial Gigabit Media Independent Interface (SGMII) IP core[3]. Golla Mahesh, Sakthivel.S.M reported on Verification IP for an AMBA-AXI Protocol using System Verilog using the questa sim tool. In the verification environment mailbox are used for communication between the different classes have presented the coverage driven verification environment with functional coverage of 100%[4]. Mahendra.B.M,Ramachandra.A.C,Bus Functional Model Verification IP Development of AXI Protocol have verified the various features of the AXI such as out of burst transactions and out of order transactions and they are verified using the questa sim simulator [5]. AnushaRanga, L. HariVenkatesh, Venkanna, Design and Implementation of AMBA-AXI Protocol Using VHDL for SOC Integration have presented the rules for verifying the SOC as SOC is the complex design and contains various components and in that they have verified the AXI protocol using the model sim simulator[6].

Taking the literature review into account we have attempted to implemented the reusable verification environment UVM (Universal Verification Methodology) for testing slave agent of AXI protocol using AMBA bus. Also coverage analysis has been used as a parameter for testing and also to demonstrate its usefulness using various assertions. Accordingly in Section II, the architecture of the AXI-UVM is described where as in section III the methodology implemented for AMBA-AXI with address calculation formulae is presented. Results and observations are presented in sectionIV. Conclusion and future scope is explained in section V.

2. THEORETICAL BACKGROUND:ARCHITECTURE OF AXI-UVM

AXI provide flexibility in the implementation of interconnect architectures and it is also backward-compatible with existing AHB and APB interfaces because of this main features AXI protocol is efficient protocol because of its ultra-high-performance. It contains of five channels, viz.; write address, write data,write response, read address, read data channel of AXI protocol. These are considered for verification. The implemented AMBA AXI-UVM block diagram as shown in Figure1. Universal Verification Methodology consist three blocks. 2.1.Top block having test. 2.2. Testblock having environment with virtual sequence and 2.3. Environment and design under test or interconnect

2.1 TOP BLOCK HAVING TEST

The top block creates instances of the Device Under Test (DUT) of the test bench. Top interface module holds all the signals of the DUT. This acts as a link to the monitor, the driver and the DUT.

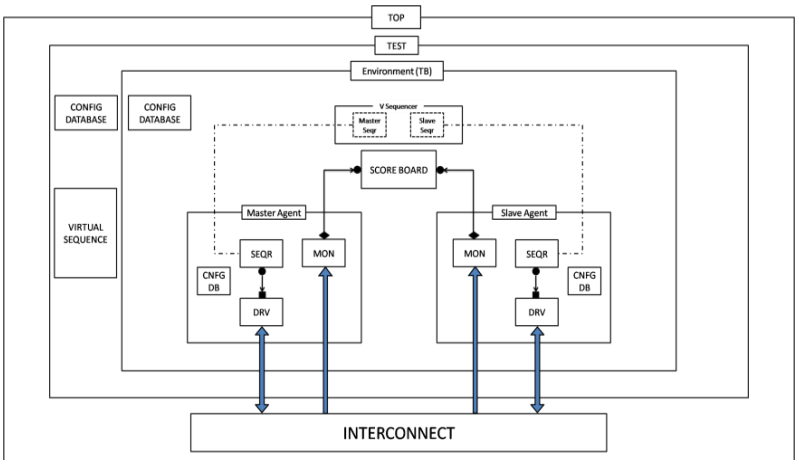

Figure. 1. Block diagram of AMBA AXI-UVM

2.2 TEST BLOCK HAVING ENVIRONMENT WITH VIRTUAL SEQUENCE

Test block is a top level block in UVM. It has two purposes first one is to create the environment block and then to connect the sequencer to the sequence.

2.3 ENVIRONMENT AND DESIGN UNDER TEST OR INTERCONNECT

Environment consist of the five blocks viz, 2.3.1. Master agent 2.3.2 Slave agent
2.3.3 Scoreboard 2.3.4. Coverage 2.3.5. Virtual sequencer.
The environment is top most UVM verification environment.

2.3.1 MASTER AGENT

Master agent holds drivers, sequencer and monitors.Sequences signify the input to the DUT, such as instructions, networking packets and bus transactions.

2.3.2 SLAVE AGENT

Slave agent holds drivers, sequencer and monitors. Monitors, sequencers and drivers can be used independently.

2.3.3 SCOREBOARD

Scoreboard is built to check the response from the DUT against the expected response. It is done by comparing them to the Reference Model.

2.3.4 COVERAGE

Functional coverage is the determination of how much functionality of the design has been exercised by the verification environment.

2.3.5 VIRTUAL SEQUENCER

The virtual sequencer has the handles of physical sequencers which are pointed to physical sequencers in the environment.

3. METHODOLOGY OF IMPLEMENTATION

Implementation of AXI-UVM is carried out using Rivera-Pro software. Rivera-Pro is tool provided by aldec to deliver its customer innovative products in shorter time with high performance simulation and containing supporting features that support verification libraries such as UVM and also support verilog,system C and many more.The driver logic channels to be verified for AXI-UVM are five which are mentioned as further.

3.1 Signals of write address channel
3.2 Signals of write data channel
3.3 Signals of write response channel
3.4 Signals of read address and control channel
3.5 Signals of read data and control channel

3.1. SIGNALS OF WRITE ADDRESS CHANNEL

During write logic transaction, the master asserts the AWVALID (Write address valid signifies availability of valid write address and control information) signal only when it drives valid address and control information. When asserted, AWVALID must remain asserted until the rising clock edge after the slave asserts of AWREADY(Write address ready indicates that slave is ready to accept an address). The default state of AWREADY can be either HIGH or LOW. When AWREADY is HIGH the slave accepts any valid address that is presented to it. The flowchart for implementation of Signals for write address channel is as shown in figure 2.

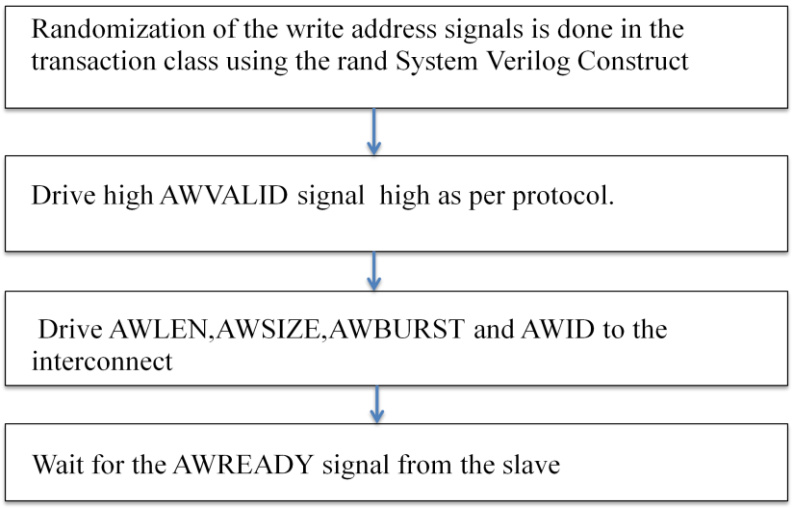

Figure.2.Flow chart of driver logic of write address channel

Write Address calculation protocol is as shown below

i. Start Address = AxADDR
ii. Number of Bytes = 2 ^ AxSIZE
iii. Burst Length = AxLEN + 1
Here A stands for the AXI protocol and x stands for W for write channels and R for read channels.
iv.

$$Aligned\ address = \text{INT}\left(\frac{Start\ address}{No.of\ Bytes}\right) xNo.of\ bytes \quad (1)$$

v. The equation used to determine the address of the first transfer in a burst:
Address_1 = Start Address. (2)
vi. For an INCR burst, and for a WRAP burst for which the address has not wrapped, this equation (3) determines the address of any transfer after the first transfer in a burst:

$$AddressN = \text{AlignedAddress} + (N-1)x\ No.Of\ Bytes \quad (3)$$

vii. For a WRAP burst, the Wrap Boundary variable defines the wrapping boundary:

$$WarpBoudary = (\text{INT}(\frac{Start\ address}{No.of\ Bytes})\text{x}\ Burstlength))$$

x(No.ofBytes)x (Burst length) (4)

3.2. SIGNALS OF WRITE DATA CHANNEL

The flow chart for the write data channel is similar to the write address channel shown in Fig 2.The difference is AWVALID,AWREADY,AWID(write address ID signal is an identification tag for write data) in the address channel are WVALID(write valid signal indicates that a valid write data is available),WREADY (write ready signal signifies that slave can accept valid write data) and WID (write ID tag signal is an ID tag of write data transfer) signals in the write data channel. The master asserts the WVALID signal only when it drives valid write data. When asserted, WVALID remains high until the rising clock edge after the slave asserts WREADY. WLAST (write last this indicates the last transfer in burst) signal while it is driving the final write transfer in the burst.

3.3 SIGNALS OF WRITE RESPONSE CHANNEL

As per the AXI protocol of driver logic, during write logic transaction,the slave asserts the BVALID(write response valid,this signal indicates that the channel signaling a valid response) signal only when it drives valid write response. When asserted, BVALID remains asserted until the rising clock edge after the master asserts BREADY (Response ready, signal indicates that master can accept a write response). The default state of BREADY can be HIGH, but only if the master can always accepts a write response in a single cycle. The implementation of Signals of write response channel is as shown in flowchart for of Figure 3.

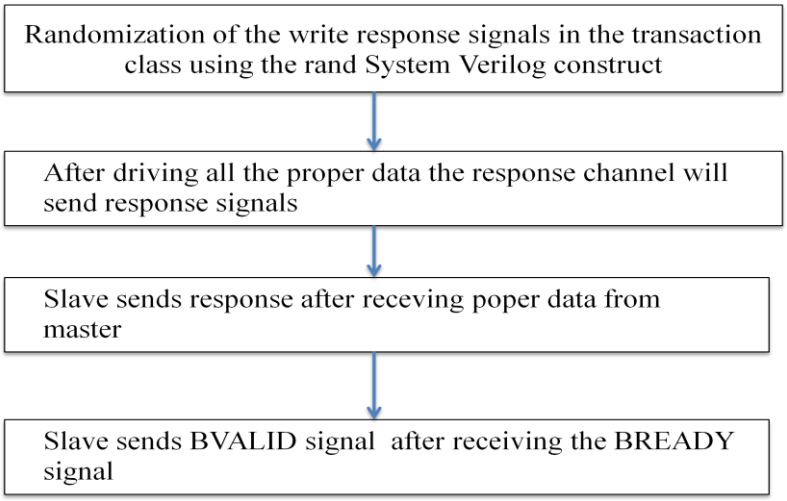

Figure.3.Flow chart of driver logic of write response channel

3.4 SIGNALS OF READ ADDRESS CHANNEL

The flowchart of the read address channel is similar to the write address channel,the only difference is the signals AWVALID,AWREADY,AWID in the write address channel is replaced by the ARVALID(read address valid signal indicates that slave is ready to accept an address),ARREADY(Read address ready signal indicates that slave is ready to accept an address) and ARID (Read address ID signal is the identification tag for read address signals) as shown in figure 4. As per AXI protocol of driver logic, during read logic transaction the master asserts the ARVALID signal only when it drives valid address and control information. When asserted,

ARVALID must remain asserted until the rising clock edge after the slave asserts the ARREADY signal. The default state of ARREADY can be either HIGH or LOW. This specification recommends a default state of HIGH, provided the slave must be able to accept any valid address that is presented to it. The implementation of Signals of read address and control channel is as shown in flowchart of Fig.4.

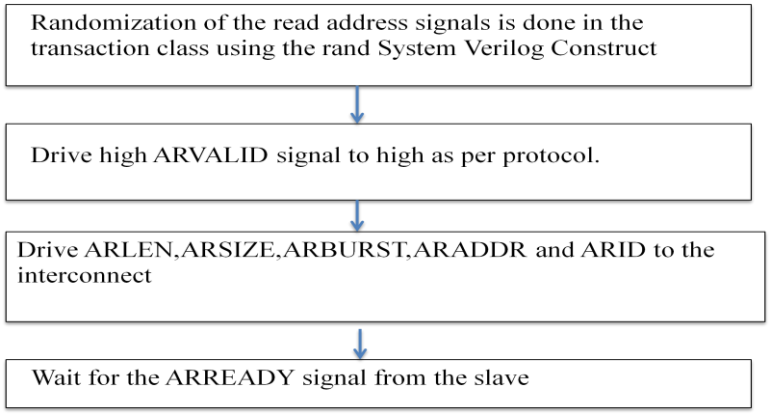

Figure.4.Flow chart of driver logic of read address channel.

3.5 SIGNALS OF READ DATA AND CONTROL CHANNEL

The read data and control channel is similar to the write data channel as shown in the flowchart 5,The slave assert the RVALID(Read valid signal indicates that the channel is signaling read data) signal only when it drives valid read data. When asserted, RVALID must remain asserted until the rising clock edge after the master asserts RREADY (Read ready signal indicates master can accept read data and response information). Even if a slave has only one source of read data, it asserts the RVALID signal only in response to a request for data. The master interface uses the RREADY signal to indicate that it accepts the data. The implemented driver logic of read data channel is as shown in flowchart of Fig.5

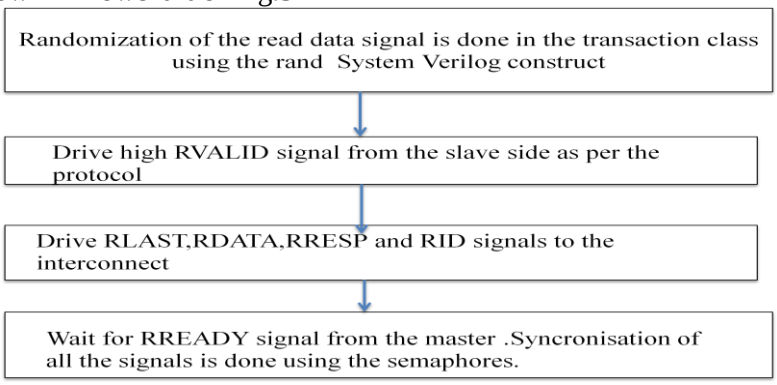

Figure.5.Flow chart of driver logic of read data channel

4 RESULTS AND OBSERVATION

Result for verification of AMBA-AXI for all five channels obtained from UVM environment of Rivera-Pro is presented as below

4.1. OUTPUT OF DRIVER LOGIC OF WRITE ADDRESS CHANNEL

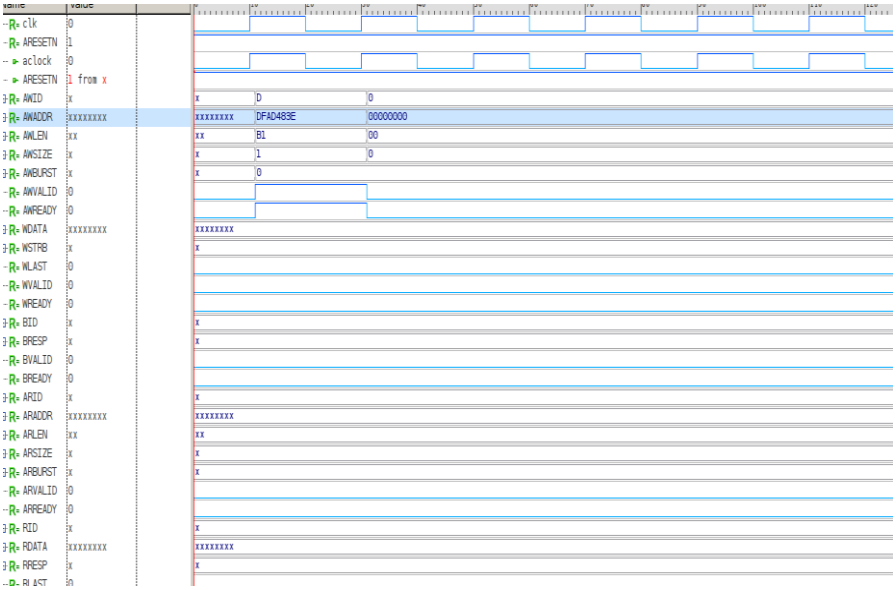

Figure.6.Output of driver logic of write address channel

It is observed from Figure 6,for output of driver logic of write address channel that, when AWVALID and AWREADY signal is high, then AWID and 32 bit AWADDR is generated as per address calculation in the protocol.

4.2 OUTPUT OF DRIVER LOGIC OF WRITE DATA CHANNEL

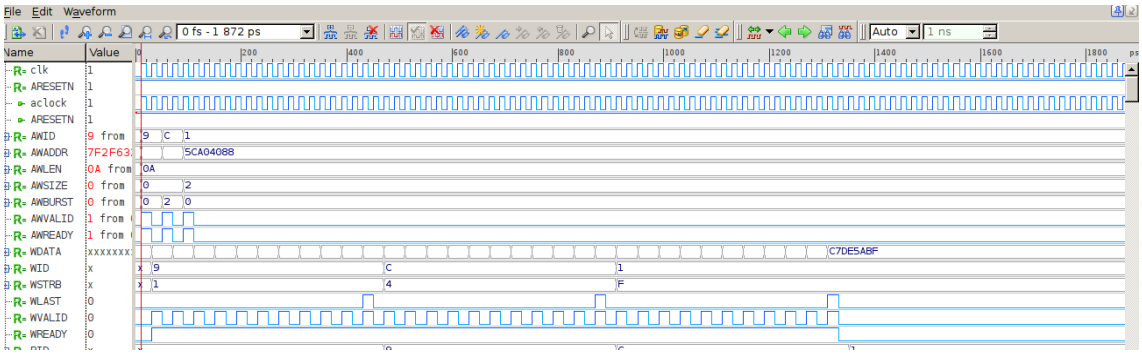

Figure.7.Output of write data channel driver logic

It is observed from the waveform of Figure 7,that write identification match, WID match is found with AWID as per the requirement .When WVALID signal is asserted,then WDATA is sent according to the WID and WLAST signal is seen to be asserted after receiving last signal in the burst.Here length of the data received is decided on the value of signal AWLEN

4.3 OUTPUT OF DRIVER LOGIC OF WRITE RESPONSE CHANNEL

Figure.8. Output of driver logic of write response channel

It is observed from the waveform of Figure 8, on output driver logic of write response channel that, after receiving all the data from the WDATA correctly, then BVALID signal has been asserted. Though BREADY can be high or low, BREADY signal is found high as per requirement of the protocol. Also match between BID, AWID and WID is seen as per requirement after BVALID signal has arrived and the BRESP signal has gone low.

4.4 OUTPUT OF DRIVER LOGIC OF READ ADDRESS CHANNEL

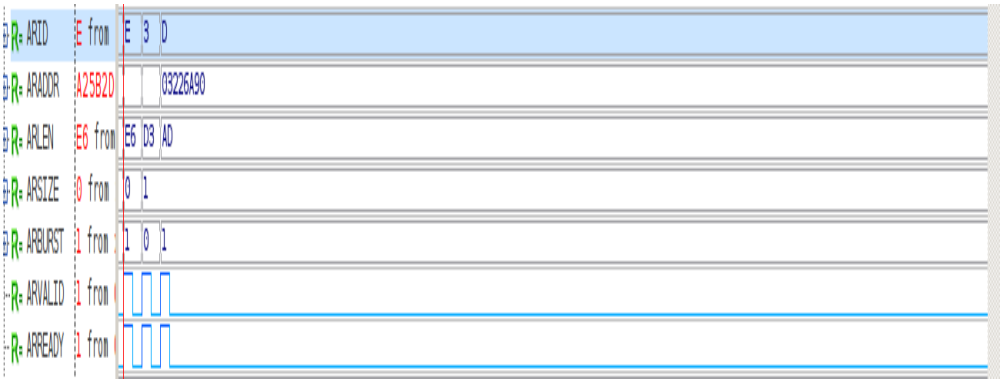

Figure.9.Output of driver logic of read address channel

It is observed from the waveform of Figure 9 that, the read address channel operations are same as the write address channel. ARADDR and ARID are found to be generated when ARVALID and ARREADY signal is high. The ARREADY can come before ARVALID or it can come after ARVALID or both ARREADY and ARVALID both can occur at same time. ARVALID and ARREADY are seen to occur at the same time in the verification of read address channel as an effect of randomization .

4.5 OUTPUT OF DRIVER LOGIC READ DATA AND CONTROL CHANNEL

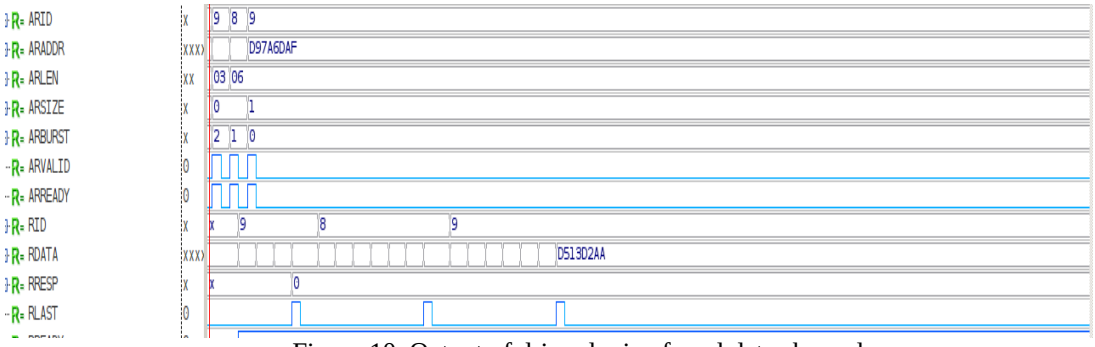

Figure.10. Output of driver logic of read data channel

It is seen from the waveform of Figure 10 that, according to the value of ARLEN signal the length of the data is decided, the RID and ARID match is found as per protocol. RDATA signal has come in output after receiving RVALID signal. Since RREADY signal can be high or it can be low this RREADY signal is found to be high in the output as per required in the protocol also RLAST(read last signal indicates the last transfer in read burst) signal is asserted after receiving the last data in the burst and as decided according to the burst length.

5 CONCLUSION AND FUTURE SCOPE

In this paper the driver logic has been implemented and verified successfully according to the protocol using the Rivera Pro and results are observed for single master and single slave. The results have shown the correctness of AMBA-AXI protocol. Testing the design using coverage as performance parameter and making it cacheable and bufferable are the future future scopes this implemented design on AMBA AXI. This design can be used to implement the monitor logic in which the signals driven by the channels can be collected in the monitor logic and can be compared in the scoreboard in order to ensure the data driven and collected are exactly the same. The axi can be used to observe the out of order transactions and out of burst transactions.

ACKNOWLEDGMENT

I would like to express my sincere gratitude towards Prof. Dr. Mrs.V.Jayashree Professor in Electronics Engineering of DKTE Society's Textile & Engineering Institute, Ichalkaranji (An Autonomous Institute) , who has been my supervisor. She has been my philosopher, mentor and guide. She provided me with many helpful suggestions, important advice and constant encouragement in this work.

REFERENCES

[1] Amba axi protocol specification, arm, 2011.
[2] Gayathri m, rini sebastian, silpa rose mary, anoop thomas, “a sv-uvm framework verification of sgmii ip core with reusable axi to wb bridge uvc”, ieee ,3rd international on advance computing and communication systems,jan 22 & 23 ,2016 ,coimbatore, india.
[3] Rini sebastian, silpa rose mary, gayathri m, anoop thomas, “assertion based verification of sgmii ip core incorporating axi transaction verification model”, ieee,international conference on control, communication & computing india (iccc), 19-21 november 2015 , trivandrum.
[4] Golla mahesh, sakthivel.s.m, verification ip for an amba-axi protocol using system verilog, international journal ofengineering research and general science, volume 3, issue 1, january-february, 2015.
[5] Mahendra.b.m,ramachandra.a.c, bus functional model verification ip development of axi protocol, international journal ofengineeringresearch andgeneral science, volume 3, special issue 1, february, 2014

[6] Anusharanga, l. harivenkatesh, venkanna, design and implementation of amba-axi protocol using vhdl for soc integration, international journal of engineering research and general science, vol. 2, issue4, july-august 20,2012.

AUTHORS

Bijal Thakkar is currently pursuing her M.Tech in Electronics from Shivaji University, Ichalkarnji .Doing project in DKTE Society's Textile & Engineering Institute. She Obtained her bachelor's degree in Electronics and communication engineering from Shivaji University,Kolhapur.Her interest includes,digital circuits design and Verification.