IOSR Journal of VLSI and Signal Processing (IOSR-JVSP)
Volume 6, Issue 3, Ver. II (May. -Jun. 2016), PP 54-58
e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
www.iosrjournals.org

Verification IP of AMBA AXI v1.0 Using UVM

Priyanka M Shettar[1], Ashwin Kumar[2]
[1](Electronics and Communication Engineering, Sahyadri College of Engineering and Management, India)
[2](Electronics and Communication Engineering, Sahyadri College of Engineering and Management, India)

Abstract: *In this paper, AMBA AXI v1.0 is verified using UVM (Universal Verification Methodology) where verification environment is created. Verification is carried out by System Verilog language and UVM modelling approach. There are five independent channels in AXI and each channel is verified which includes write address, data and response channels and read address and data channels. Mentor Graphics Questa-Sim tool is used to perform the functional verification of AMBA AXI v1.0*

Keywords: *Verification IP; System Verilog; Transactions; Channels; Out of order transfer; UVM; Questa-Sim.*

I. Introduction

The interface IP blocks are tested extensively by the AMBA architecture hierarchy family and provides an exact solution for protocol verification [1]. Interface of AXI helps in design, implementation of highly integrated modular interfacing because it is a technology independent methodology [4]. This project aims at building the environment of verification for AXI using UVM. The same environment is used to implement the overall coverage, score-board and monitor components for the signals [3]. The modern System on Chip (SOC's) usually use the protocol buses which consumes, more power (APB), less power (AHB) and moderate power (AXI) for the synchronized communication between the hosts during data communication. When compared with AHB and APB structures, AXI gives better performance and also consumes moderate power. So for the modern design of SOC, AXI can be considered as the alternative standard [1]. The key points of this bus are

1. It has separate channels for data, response and address of both read and write transfer for which only start address is enough for the burst based transactions.
2. It uses byte strobe mechanism which supports the unaligned data transfer.
3. Only start address is enough for the burst based transactions.
4. Multiple outstanding addresses out of order transactions can be achieved using AXI protocol.

II. Objectives

1. To study the specification of AMBA AXI v1.0 including all scenarios.
2. To generate the Test Plan comprising of Test Cases meeting all specified scenarios in the specification.
3. To understand the development of verification environment for Master and Slave using UVM methodology. UVM methodology includes the Test Plan along with the topology of driver, sequencer and monitor.
4. To analyse, how driver drives a sequence item from sequencer, performs the operation and send it to scoreboard.
5. In the scoreboard, the obtained output is compared with the expected one. If the obtained output matches with the expected result, verification is completed successfully

III. Proposed Methodology

3.1 Architecture of AMBA Advanced Extensible Interface

The Advanced Extensible Interface belongs to burst based type of transaction. Each read or write transfer tells what and how the data has to be sent from master to slave. The work proposed tells that how master and slave establishes a communication between each other using different channels. This is achieved using System Verilog and then verified the design using Universal Verification Methodology. The AMBA AXI block diagram is shown in Fig-1. There are several channels which are used to establish the communication between master and slave. Those are address, data and response channels of both read and write respectively. Read channel does not have separate response channel instead response signal is included in data channel itself. Mentor Graphics Questa Sim tool is used to design the architecture of both AMBA AXI protocol and its VIP

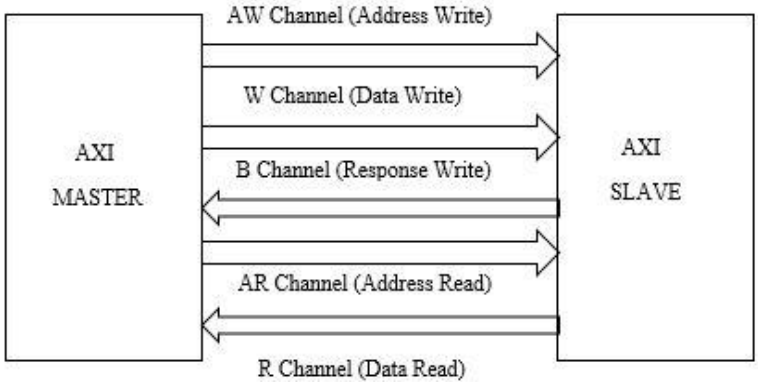

Fig. 1. Block diagram of AMBA AXI

3.2 Design each block of component environment in Universal Verification Methodology (UVM)

Each component of UVM has to be designed. In this project System Verilog is used to design each component. It consists of a master agent and a slave agent. Master agent again has two separate agents in which each agent deals with write and read channels separately. This agent is in environment in which different agents are connected to scoreboard. Each agent has a driver which drives all the signals and a sequencer which receives inputs from test case and provides it to driver. The test, environment and agents are connected through a virtual interface which is present in top as shown in Fig-2

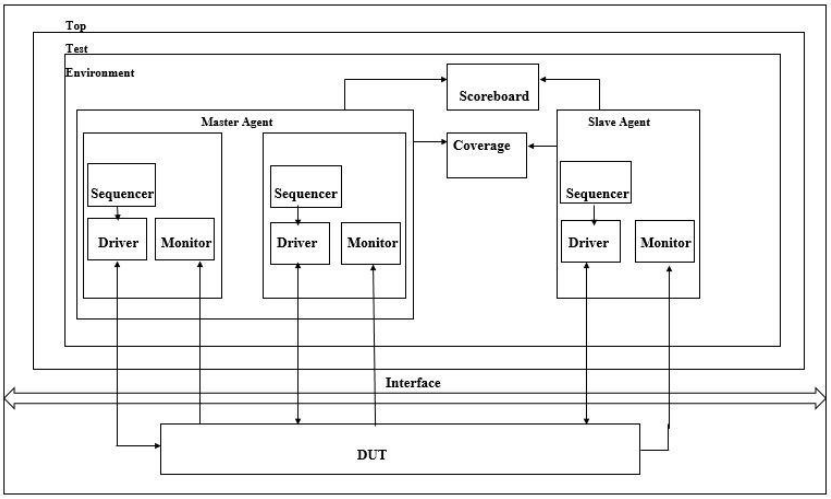

Fig. 2. Block diagram of AMBA AXI

IV. Verification of Channels

4.1 Verification of Write Channels

Write channels has three separate channels. All the three channels are verified together for each transaction in verification stage. Address channel has AWID, AWADDR, AWLEN, AWSIZE, AWBURST, AWVALID, AWREADY signals. Master sends these signals only when AWVALID is high and slave receives these signals only when AWREADY signal is high. AWID represents which transaction is going on. Data channel has WID, WDATA, WLAST, WSTRB, WVALID, WREADY signals. AWID should match WID and WDATA is written according to AWLEN, AWSIZE and AWBURST. WLAST signal represent the last data of the transaction. Response channel has BID, BVALID, BRESP, BREADY. Even here BID should match both AWID and WID. Slave sends BRESP only when master is ready with BREADY. Master write operation is completed only after response is received from the slave. Simple transaction is shown in Fig-3

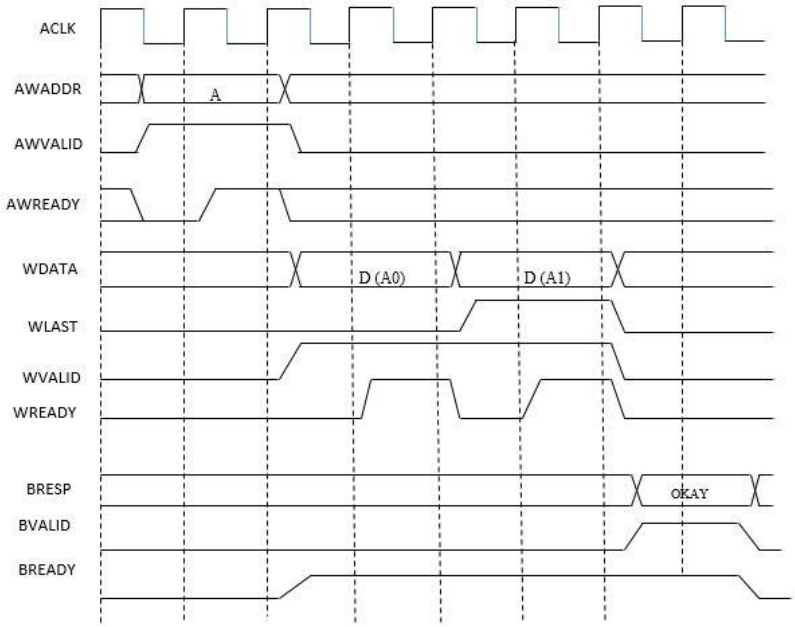

Fig. 3. Write Transaction

4.2 Unaligned Transfer

If start address is not completely divisible by 2 power size of the transaction i.e. (2** (AWSIZE or ARSIZE)) then that address is called Unaligned Address. If remainder is not zero after division, the start address is incremented by 1 till it is completely divisible by (2** (AWSIZE or ARSIZE)) to make it aligned address.

4.3 Verification of Read Channels

Read channels has two separate channels. The read response and any read data information from slave to the master is conveyed by read channels. Address channel has ARID, ARADDR, ARLEN, ARSIZE, ARBURST, ARVALID, ARREADY signals. Master sends these signals only when ARVALID is high and slave receives these signals only when ARREADY signal is high. ARID represents which transaction is going on. Data channel has RID, RDATA, RLAST, RRESP, RVALID, RREADY signals. ARID should match RID and RDATA is read according to ARLEN, ARSIZE and ARBURST. RLAST signal represent the last data of the transaction. Slave sends RRESP only when master is ready with RREADY. Simple transaction is shown in Fig-4.

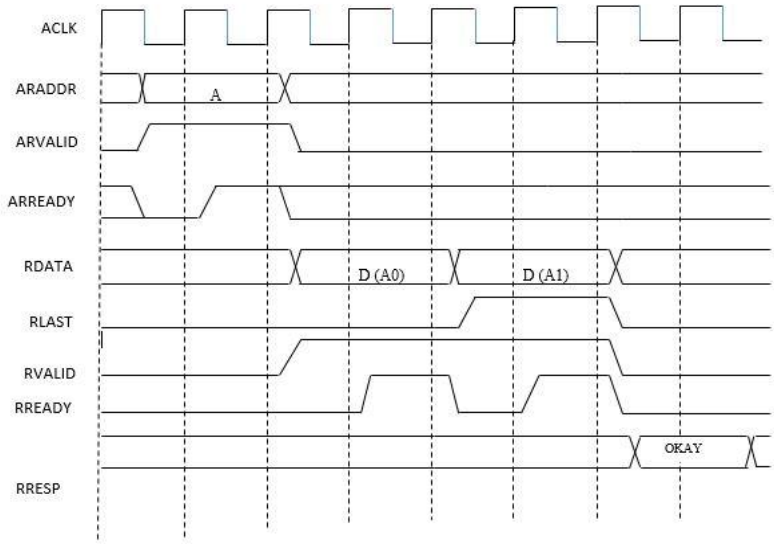

Fig. 4. Read Transaction

4.4 Multiple Outstanding Address

AXI supports multiple outstanding address. This means that without the completion of earlier transactions, the next transaction address can be issued by the master. This feature enables parallel processing of

transactions hence the system performance is also improved. The disadvantage of multiple outstanding address is slow slave or slave master. If one master which is slow has started a transaction then it takes long time to complete it which makes other master to wait which is fast. So in order to avoid this problem out of order transaction is supported by AMBA AXI.

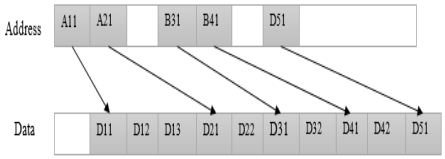

Fig. 5. Multiple Outstanding Address

Fig. 6. Slow Master or Slow Slave

4.5 Verification of Out of Order Transaction

When the master generates the same ID's, then those transactions have to be performed in-order i.e. the same order as it is issued by the master. If a master want to do out of order transaction, then the ID's must be different. Here the transactions can be processed any time irrespective of its order and can complete its transaction before any other slow transaction. This means a faster transaction which is issued late by the master can complete its transaction without waiting for a slow transaction which is issued first by the master.

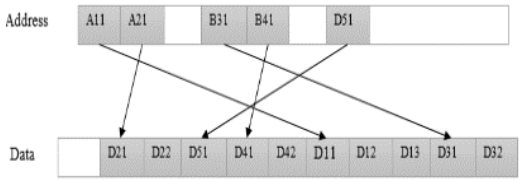

Fig. 7. Out of Order Transaction

V. Simulation Results

Simulation is done in Mentor Graphics Questa-Sim for random test case. Program is written in System Verilog language. This simulation results include Fixed, incrementing and wrapping burst. Write burst operation is shown in Fig-8. Read burst operation is shown in Fig-9 and out of order transaction is shown in Fig-10.

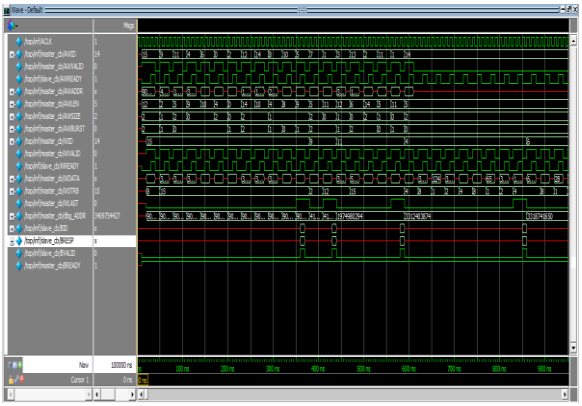

Fig. 8. Write Burst

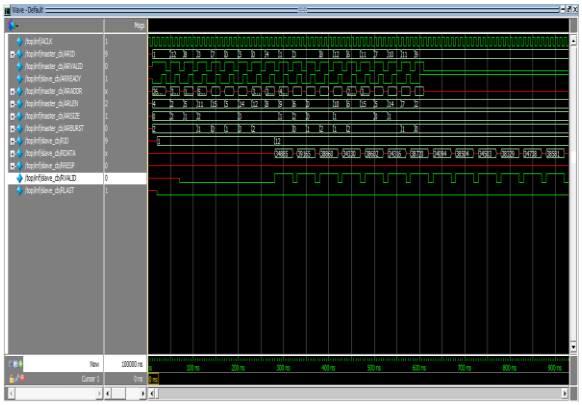

Fig. 9.Read Burst

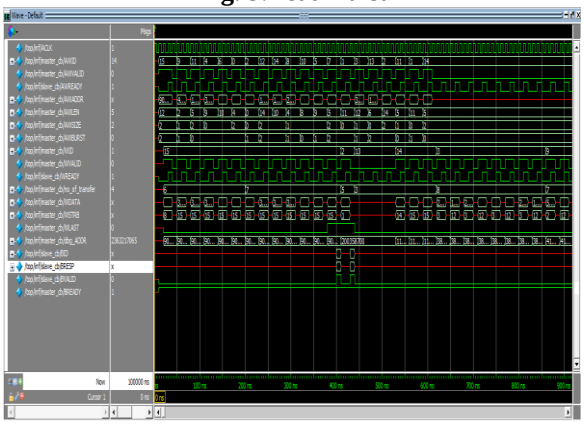

Fig. 10.Out of Order Transaction

VI. Conclusion

The project helped to achieve the verification of AMBA AXI bus. Here all the signals are verified using UVM. The master sends the address and data to the slave and gets the response effectively by the slave. In this paper, environment for verification of AXI is achieved and it checks for normal read and write transactions and out of order transaction. Since UVM is used, components are re-usable and it reduces the time span of verification.

Acknowledgements

I'm deeply thankful to Adeptchips Services Pvt Ltd., Bangalore for giving opportunity to be part of the organization and for wonderful guidelines and helping me carry out the project. I am thankful to my college Sahyadri College Engineering and Management for creating such an opportunity for the students to broaden their skills. I will take this opportunity to express sincere gratitude to Department of Electronics and Communication Engineering.

References

[1]. Golla Mahesh, Sakthivel.S.M, Verification IP for an AMBA-AXI Protocol using System Verilog, *International Journal of Engineering Research and General Science*, Volume 3, Issue 1, January-February, 2015

[2]. Karthik Ghanta and Srikanth Parikibandla, Design of a Two-Way Set-Associative Cache, *Advance in Electronic and Electric Engineering*, Volume 3, Number, 2013 Research India Publications.

[3]. Mahendra.B.M, Ramachandra.A.C, Bus Functional Model Verification IP Development of AXI Protocol, *International Journal of Engineering Research and General Science*, Volume 3, special Issue 1, February, 2014

[4]. Anusha Ranga, L. Hari Venkatesh, 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 2012

[5]. AMBA AXI Protocol Specification, ARM