International Journal of Research and Review
Vol. 9; Issue: 11; November 2022
Website: www.ijrrjournal.com
Research Paper *E-ISSN: 2349-9788; P-ISSN: 2454-2237*

Testing of AMBA AXI Protocol

Tanupriya A G[1], Kiran V[2]

[1]Student, Department of Electronics and Communication Engineering, R V College of Engineering, Mysore Road, RV Vidyaniketan, Post, Bengaluru, Karnataka, India-560059
[2]Associate Professor, Department of Electronics and Communication Engineering, R V College of Engineering, Mysore Road, RV Vidyaniketan, Post, Bengaluru, India-560059

Corresponding Author: Tanupriya A G

DOI: https://doi.org/10.52403/ijrr.20221114

ABSTRACT

The rapidly expanding VLSI sector has a great need for a very robust verification mechanism that can be developed in a short amount of time in the constantly changing environment where new intellectual property and chips are being designed every day having reduced time-to-market. (SoC) System on Chip design emerged as a primary integrated technique for reducing the design time of the entire system with the introduction of a standardized signal bus architecture used for the connectivity of different modules of the system. One of the major difficulties at hand is how to check these on chip bus protocols, since traditional old methods are ineffective in certifying large SoCs. In this paper, we have verified AMBA AXI protocol which shows write as well as read operations using System Verilog.

Keywords: *AMBA AXI, APB, AHB.*

I. INTRODUCTION

The correct synchronization of all of the intellectual property (IP)cores that are built into today's SOCs during data communication is a challenging issue. Modern SOCs mostly use communication protocols from the AMBA family to create synchronized communication, including the advanced extensible interface (AXI) and advanced peripheral bus (APB), Advanced High-performance Bus (AHB). Because verification is often projected to take up 70% of the whole project time, compared to the design stage's 30%, it becomes a very difficult work to complete throughout the creation of these sorts of SOCs. So, with the aid of a special testing environment having known as Verification-IP. Today's SOC communication bus protocols are essentially categorized according to how much power they utilize and how well they perform [1]. APB bus protocol structure is the simplest to create and the least power-hungry of the three AMBA protocols when compared to AHB bus protocol structure, although it performs poorly it is being compared to the AXI [4]. AMBA-AXI protocol significantly consumes the moderate power with better performance than AHB and APB protocols. Therefore, the AXI bus protocol structure has been chosen for the SOC designs if high performance is needed. AHB is a shared-bus architecture, hence arbitration schemes only allow one master to use the bus at a time, but AXI is an interconnect-based bus, allowing (multiple) masters and (multiple) slaves to communicate simultaneously, making it far more efficient. Since AXI is actually an interconnect-based bus, it is much more effective because it allows for simultaneous communication between numerous masters and multiple slaves. In this paper, we discuss the architecture of an AMBA AXI protocol verification built on System Verilog.

II. DESIGN

The central processing unit on-chip memories and Direct Memory Access devices are located on the high-performance

system bus (like AHB) of microcontroller that is based on AMBA and is capable of supporting the external memory band-width. Primary usage of the AMBA bus protocol is for data transfer between masters and slaves [1][2][3]. A high-bandwidth connectivity between the components involved in the majority of transfers is provided by this communication bus.

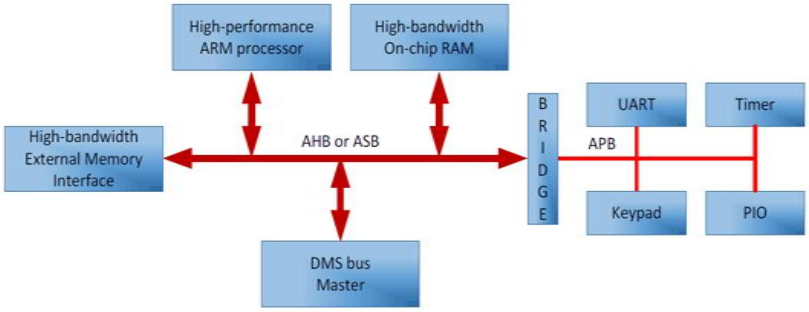

Fig. 1. AMBA Block Diagram

High frequency with high-performance system designs is being supported by the AXI protocol [3]. The AXI protocol is being appropriate for applications requiring extremely high-bandwidth and low latency design. Additionally, it allows for frequency (High) operation without elaborate bridges. Memory controllers usually with high initial access latency should use AXI. We are able to connect many masters with numerous slaves simultaneously using AXI connection, which is not possible with AHB protocol. Only the first address in AXI is initialized by the driver; the slave is responsible for the other addresses. In order to handle several transactions at once, AXI introduces the idea of ID.

AMBA AXI feature is as follows:

- Support burst write as well as read transactions, which allows us to send a certain number of bytes per transfer in single cycle (for receiving increasing performance).
- Support out of order transactions: In order to comprehend out of order, we must first comprehend in-order transactions.
- •When the request and response happen in the same order, the transaction is sent in the same order that we receive the response and is referred to as an in-order transaction.
- When an order-transactions is being sent along with it an order response is being received, the two events don't happen in the same sequence.
- Support for write and read transactions that occur simultaneously.
- Support overlapped transaction: Let's say we insert the address in the overlapped transaction at time t1 so that it can't reach me at the same time.
- Allows for protected and locked transfers.
- Supports transactions that are aligned and those that are not.

(AXI protocol) defines the following 5 transaction independent channels:

- Write address: The address and control information carried by this channel indicates the type of data that will be sent.
- Write data: Transactions between the (master-slave) masters and slaves are made through this Write data channel.
- Write response: Channel is being used in order to provide feedback from the slave to the master. shows that the master transfer has been finished.
- Read address: Channel actually carries control along with addresses information

for the read path. Which in turn identifies the type of the data being sent.
- Read data: Transactions are transferred from slave to master using this channel.

In order to handle several transactions at once, it introduces the idea of ID. Write address channel signals begin with AW. W is the signal on the Write data channel. Write response channel signals begin with the letter B. The Read Address Channel signal begins with AR. R is the signal of the read data channel.

Handshaking signals, READY and VALID, are present on each channel as AXI channel has its own unique set-of-information signals and handshaking protocol (two way). The signals VALID and READY enable a two-way handshaking mechanism. AWVALID=1 (which is driven by master) at the edge of the clk indicates that master is the driving valid information on the write address channel signals. Each channel has VALID & READY signals. If AWREADY=1 (driven by the slave), the slave is prepared to accept the address and control information from the master. When AWVALID and AWREADY are 1 or (high) at the edge of the clock, the particular transfer has been completed.

AXI channels	Direction of handshaking signals
Write Address channel	AWVALID(master->slave) AWREADY(slave->master)
Write Data channel	RVALID(master->slave) RREADY(slave->master)
Write Response channel	BVALID(slave->master) BREADY(master->slave)
Read Address channel	ARVALID(master->slave) ARREADY(slave->master)
Read Data channel	RVALID(slave->master) RREADY(master->slave)

Fig. 2. Various channels with its Signals

The following dependencies must be maintained by the AXI Protocol: • A write response happens only after the particular write transaction's last write transfer.
- Following the read data is always the address to which the data pertains.
- Channel handshaking signals have dependencies: Read and write dependencies.

The following describes the Write transaction in further detail:
- The master provides the address along with control information for write address request.
- Master sends the following write data request along with it.
- Slave replies with a state of write response.

The following describes the read transaction in detail:
- The address and control information are provided with the read address request by the master.
- The read data phase begins when the slave provides the read data.

III. METHODOLOGY

The verification process has become significantly more difficult due to the growing need for trustworthy systems with faster time to market. Verilog was once used largely for design verification, but because of its lack of flexibility, businesses have switched to System Verilog is used because of its simplicity, efficiency, and higher design reusability

IV. RESULTS

The simulation is performed in EDA playground and waveform of AMBA AXI is shown in fig 9. Write address channel signals begin with AW. W is the signal on the Write data channel. Write response channel signals begin with the letter B. The Read Address Channel signal begins with AR. R is the signal of the read data channel can be analyzed.

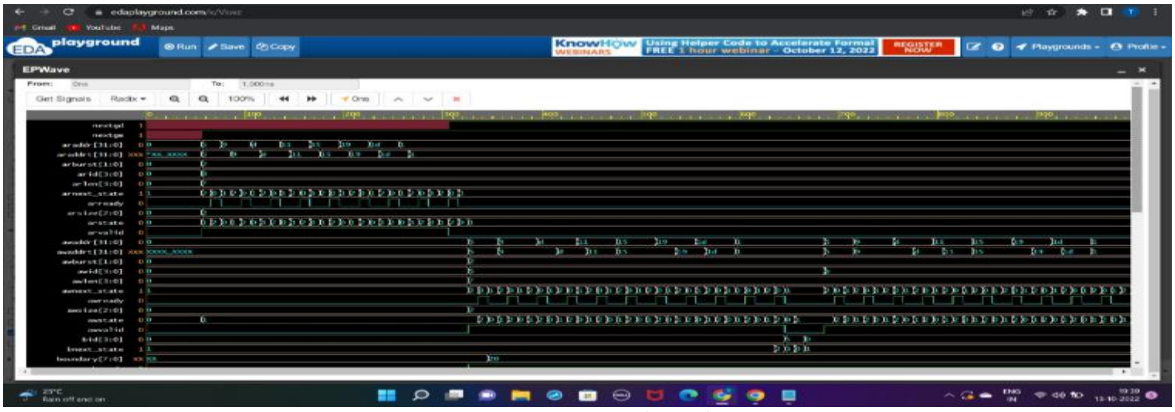

Fig. 9. Simulation result waveform of AMBA

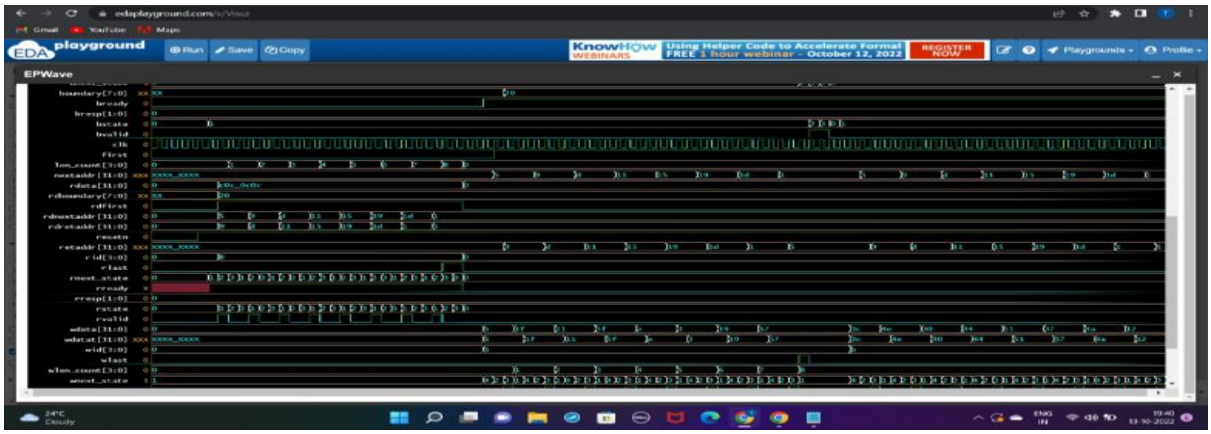

Fig. 10. Simulation result waveform of AMBA (continued)

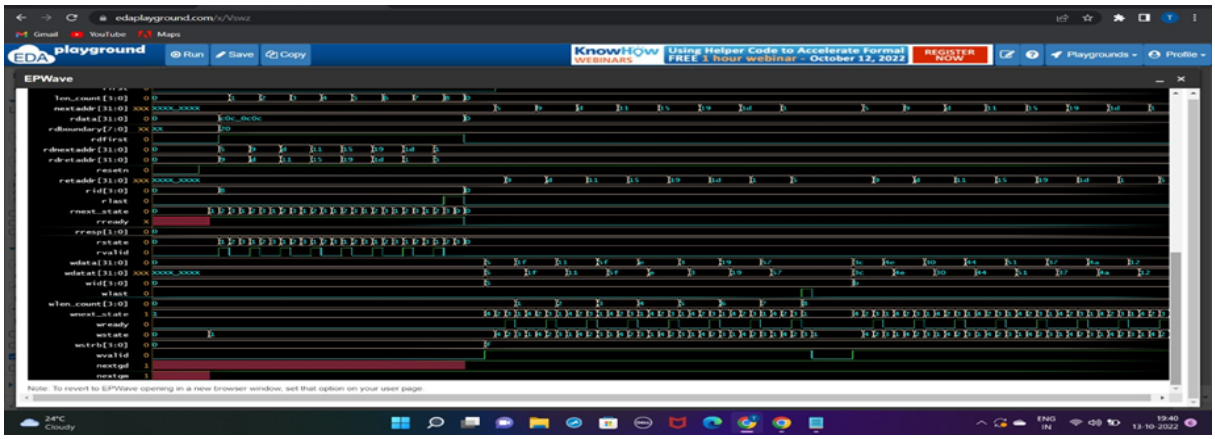

Fig. 11. Simulation result waveform of AMBA (continued)

V. CONCLUSION

Therefore, the AMBA AXI protocol was checked using System Verilog and got simulation results has been seen in fig (9,10,11). This main idea is to see the read and write operation happening.

Acknowledgement: None

Conflict of Interest: None

REFERENCES

1. Nikhil Gaikwad and Vijay.N.Patil , Verification of AMBA AXI on-chip Communication Protocol ,2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), DOI: 10.1109/ICCUBEA.2018.8697587
2. ARM, ”AMBA™ Specification(Rev 2.0)”.
3. ARM, ”AMBA AXI™ Protocol Specification”.
4. A Harsha Garua. “Verification Of Amba Axi Bus Protocol Implementing Incr And Wrap Burst Using System Verilog.” International Journal of Research in Engineering and Technology, vol. 05, no. 03, 2016, pp. 201–206. doi:10.15623/ijret.2016.0503041.
5. Renduchinthala H. H. S. S. Prasad, and ChSanthi Rani.“Development of VIP for AMBA AXI-4.0 Protocol.” IndianJournal of Science and Technology, vol. 9, no. 48, 2016, doi:10.17485/ijst/2016/v9i48/101912.
6. Patil, Rohita P., and Pratima V. Sangamkar“A Review of System-On-Chip Bus Protocols.” International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, vol. 04, no. 01, 2015, pp. 271–281., doi:10.15662/ijareeie.2015.0401042.
7. Spear, Chris. Systemverilog for Verification: a Guide to Learning the Testbench Language Features. Springer, 2014.
8. github.com and www.freepatents.com,internet source
9. Shivani Malhotra, Neelam Rup Prakash "Chapter 57 UVM-Based Verification IP of AMBA AXI Protocol Showing Multiple Transactions and Data Interconnect", Springer Nature, 2019.
10. Anurag Shrivastav, G.S. Tomar, Ashutosh Kumar Singh. "Performance Comparison of AMBA Bus-Based System-On-Chip Communication Protocol", 2011 International Conference on Communication Systems and Network Technologies , 2011
11. Chien-Hung Chen, Jiun-Cheng Ju, and Ing-Jer Huang A Synthesizable AXI Protocol Checker for SoC Integration, Conference, DOI**:** 10.1109/SOCDC.2010.5682961.

How to cite this article: Tanupriya A G, Kiran V. Testing of AMBA AXI protocol. *International Journal of Research and Review*. 2022; 9(11): 86-90.
DOI: *https://doi.org/10.52403/ijrr.20221114*

******