UVM-Based Verification IP of AMBA AXI Protocol Showing Multiple Transactions and Data Interconnect

Shivani Malhotra and Neelam Rup Prakash

Abstract Many advanced protocols are emerging and are being used widely in the industry these days. One such protocol is the AMBA Advanced Extensible Interface Protocol by ARM processors. This protocol provides significant attributes which did not exist in the previous Advanced High-Performance Protocol. This paper presents practical results of some of those attributes of this protocol. Now, it should be noted that verification of such protocols is a tedious task and consumes a lot of time. Hence, a Verification IP should be created which acts as a catalyst in the verification process. A Verification IP already contains the necessary mechanisms for testbench generation and can be easily integrated to other tools.

Keywords Advanced microcontroller bus architecture (AMBA) · Advanced RISC machine (ARM) · Advanced extensible interface (AXI) · Advanced high-performance bus (AHB) · Verification intellectual property (VIP) · Universal verification methodology (UVM)

1 Introduction to AMBA

The Advanced Microcontroller Bus Architecture was developed by ARM Ltd. in 1996 and is a registered trademark of ARM Ltd. It is actually a family of protocol guidelines which is open to the public and sets out a policy for the interlink and control the practically working structures that constitute a SoC. It also helps to design the embedded processors who possess one or more CPUs and several peripherals. It

S. Malhotra (✉) · N. R. Prakash
VLSI Design, PEC University of Technology, Sector 12, Chandigarh, India
e-mail: shivani.malhotra610@gmail.com

N. R. Prakash
e-mail: neelamrprakash@pec.ac.in

S. Malhotra · N. R. Prakash
Electronics and Communication Department, PEC University of Technology, Sector 12, Chandigarh, India

R. Bera et al. (eds.), *Advances in Communication, Devices and Networking*,
Lecture Notes in Electrical Engineering 537,
https://doi.org/10.1007/978-981-13-3450-4_57

defines a common backbone for Soc blocks that accolades a multipurpose planning approach [1].

Several buses are described in the AMBA specifications which are based on different generations:

- Advanced High-performance Bus (AHB)
- Advanced System Bus (ASB)
- Advanced Peripheral Bus (APB)
- Advanced Extensible Interface (AXI)
- Advanced Trace Bus (ATB)

The abovementioned protocols are, in fact, a predefined standard for embedded processors as they are registered skillfully and as such, no authorities are entailed to utilize them. These specifications have been obtained to fulfill the following significant conditions:

- It ensures that the growth of embedded microcontroller products is made smooth and easy with just one or sometimes more than one signal processors.
- Other necessity is to become technology maverick and make sure that multipurpose outlying libraries get drifted across a wide ranging of IC processes and be apt for various design implementation technologies [1].
- Also to promote customizable system structure to outdo the freedom of processor, issuing a progress enchiridion for improved cached CPU fundamentals plus the growth of outlying libraries.
- Another important condition is to reduce the necessary silicon substructure in order to carry out on-chip and off-chip transmission efficiently for both operation and manufacturing test.

2 Introduction to AXI Protocol

The latest generation of AMBA that is AXI protocol targets high performance and is used in high-frequency system designs. This latest generation protocol has an edge over previous generations like APB, AHB as it enables high-frequency operation without using complex bridges. And if backtracking is done, it would be compatible with the existing AHB and APB protocols. This VIP [2] can be used to verify the AXI bus in any kind of project and supports many simulators like Aldec Riviera PRO, Cadence Incisive Enterprise Simulator, Vivado Simulator, Mentor Graphics Questa Prime, and Synopsys VCS (Fig. 1).

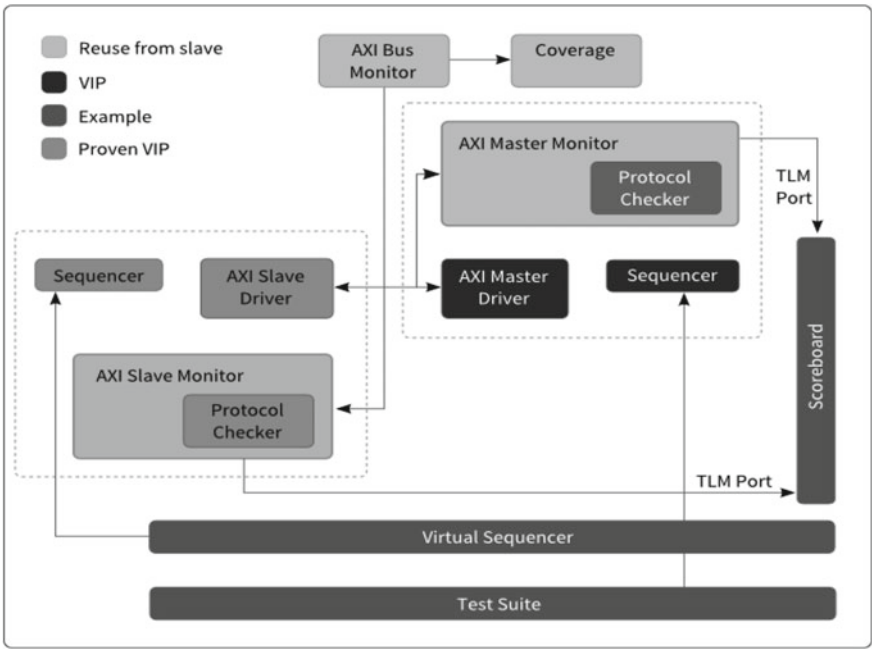

Fig. 1 AXI verification IP

3 Key Features of AXI Protocol

3.1 Individual Address/Control and Data Phases

The main features of AXI protocol is the presence of independent and discrete address and data buses for transmission of data between the master and slave. Several masters and slaves can be present in AXI.

3.2 Unaligned Transfers

For the bursts which are formed of data transfers exceeding one-byte boundary, it is not necessary that the initial bytes which have to be ingressed needs to be aligned with the built-in width threshold. It happens when the value of the address (AxADDR) is not a multiple of the size of the data being transferred (AxSIZE) [3].

3.3 Multiple Outstanding Addresses

AXI protocol utilizes transaction ID tags for delivering manifold outstanding addresses. There are no restrictions on order of data packets sent during the read and write transactions and can be completed in any order. Hence in the AXI protocol, we can complete the out-of-order transactions by permitting every port to work as multiple ordered ports. These features enhance the performance of interconnects and increase the efficiency of the system [4]. We can order only those transactions which have a valid ID and there is no restriction even if they have different IDs. There are five transaction IDs present in AXI which are:

AWID—This ID tag is used for the write address group of signals.
WID—This can be defined as the write ID tag for a write transaction.
BID—This ID tag is used for the write response.
ARID—This is the ID tag for the read address group of signals.
RID—This can be defined as the read ID tag for a read transaction.

3.4 Out-of-Order Transaction Completion

This protocol supports the accomplishment of transactions which are out of order. In this, an ID tag is provided to every transaction across the interface. It is necessary for the protocol to complete the transactions with the same ID tag in order but this does not apply for transactions with different ID tags, they can be completed in any order.

4 Analysis and Results

In this paper, simulation results of two basic yet most important attributes which clearly help this protocol to stand aside from previous generation protocols have been explained. These results have been obtained from DVE Synopsys tool. So, given below are those features along with their VPD as a result of DVE Synopsys which is basically a waveform viewer by Synopsys.

4.1 Separate Address/Control and Data Phases

We know that the identity of information that will be conveyed is explained by the transaction on address channel containing address and control data [2]. The interchange of data between master and slave is performed with help of a write data channel to the slave or a read data channel to the master. In write transactions,

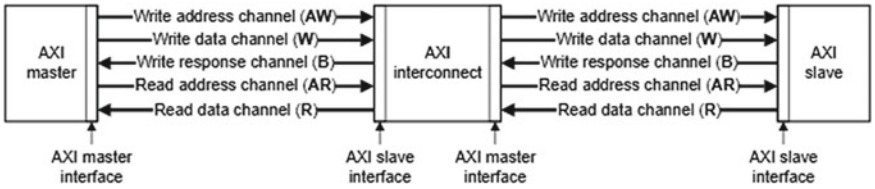

Fig. 2 AXI master slave interface

in which whole data get transferred from master to slave, a supplementary write response channel is there in AXI protocol that allows a write transaction completion signaling from slave to master.

All the five individual channels contain a set of data signals and utilize a two-way VALID and READY handshake process (Fig. 2).

Read/write address channels [5]

All the read and write transactions possess their individual address channels. The necessary address and control data is transferred by a suitable address channel.

Read data channel

The data pair comprising read data and its response is transmitted by the read channel from slave to the master. This channel comprises of:

- A data bus, which can be of the width ranging from 8 to 1024 bits.
- A read response signaling the fulfillment report of the read transaction.

Write data channel

The write data is transmitted from master to slave through this channel. It involves a data bus, which can be of the width ranging from 8 to 1024 bits.

Write response channel

This channel issues various procedures for the slave acknowledge the write transactions. All write transactions use accomplishment indications. These indications are in form of signals which occurs for single time only for individual bursts.

Following are the simulation results referring to individual address/control and data buses for read and write transactions (Figs. 3, 4, 5 and 6).

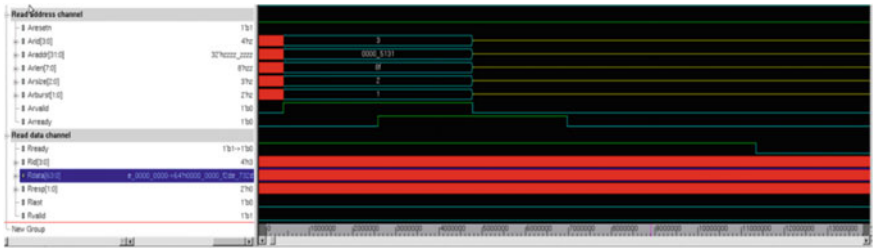

Fig. 3 Simulation result showing the read address/control signals and Arid value of 3 which is of 4 bits

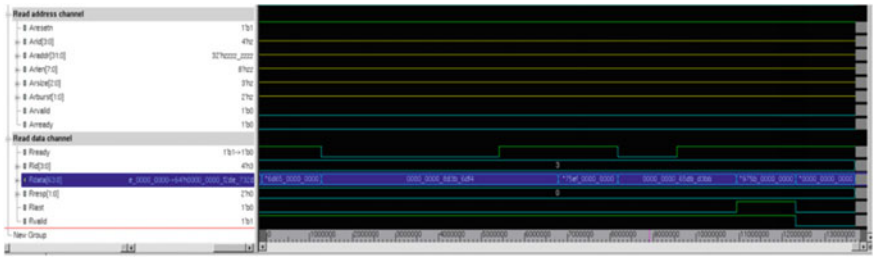

Fig. 4 Simulation result showing the read data of 64 bits with the same Arid value

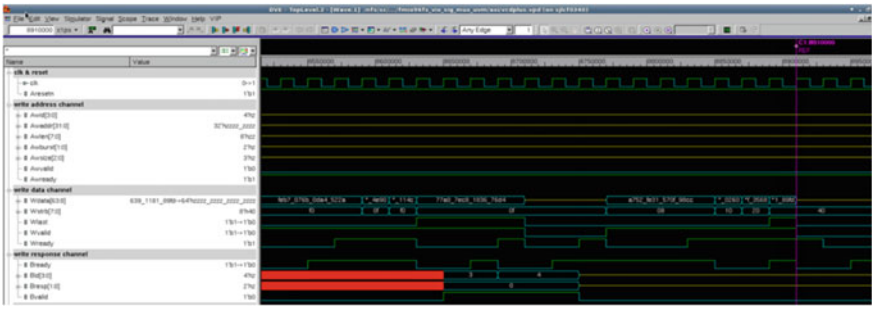

Fig. 5 Simulation result showing write data of 64 bits along with the strobe signals

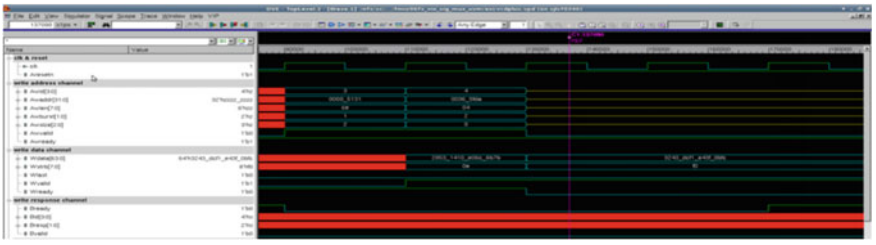

Fig. 6 Simulation result showing write data, strobe signals and the write response with Arid of 3 and 4 in write address channel

4.2 Out-of-Order Completion

The AXI Protocol supports the completion of transactions which are not in order. In this, the protocol provides an ID tag to every transaction present across the interface. This ID tag plays a great role in deciding the order of transactions. It is the requirement of this protocol that the transaction with same ID tags need to be accomplished in same sequence, whereas the ones with different ID tags can be accomplished out of order [6].

The inclusion of this feature has improved the system performance to a great extent in two ways:

(1) The transactions with fast-responding slaves can be enabled by the interconnect to move forward of the prior transactions having slow going slaves.
(2) Read data which is out of order can be returned by complex slaves. For example, a data item which has to be accessed later might be available before the data item which is to be accessed earlier [1].

Following are the simulation results showing the out-of-order completion attribute of AXI Protocol (Figs. 7 and 8).

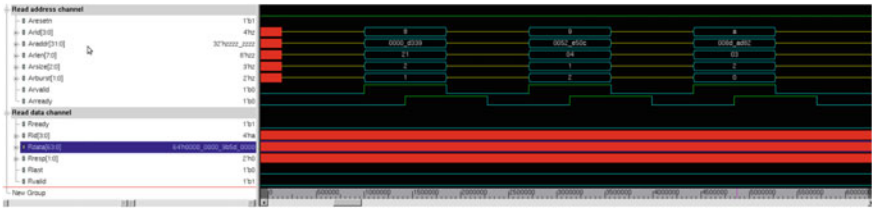

Fig. 7 Simulation result showing read data and Arid of decimal values 8, 9, and 10

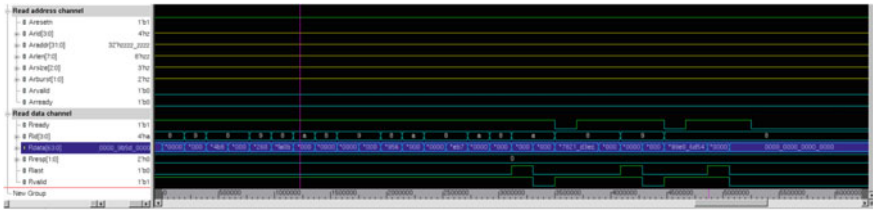

Fig. 8 Simulation result showing out-of-order completion

5 Conclusion

The verification IP has been successfully designed and verified by using DVE Synopsys tool. Test cases concerning Multiple Transactions and Data Interconnect were run and the results were obtained. In the future, more work can be carried out based on some other key features of AXI protocol such as unaligned transfers, atomic accesses, etc. Also, some efforts can be made in the field of AXI Bridge to add external connectivity.

References

1. ARM, AMBA AXI Protocol Specification [Online]. Available http://www.arm.comARM
2. Silberschatz A (1983) A case for non-two-phase locking protocols that ensure atomicity. IEEE Trans Softw Eng SE-9(4)
3. The AMBA SOC Platform. http://www.springer.com/in/book/9781461478638
4. https://www.xilinx.com/products/intellectual-property/axi_systemcache.html
5. Synopsys VCS https://www.synopsys.com/verification/simulation/vcs.html
6. Wang HW, Lai CS, Hwang SA, Lin Y-H (2008) On chip interconnection design and soc integration with open core protocol (OCP). In: Proceedings of the IEEE international symposium, VLSI Design 2008, pp 25–28
7. AMBA AHB Protocol Specification [Online]. Available http://www.arm.com
8. Chang C-Y, Lee K-J (2014) On deadlock problem of on-chip buses supporting out-of-order transactions. IEEE Trans Very Large Scale Integr Syst 22(3)
9. http://www.testandverification.com/solutions/vip/axi-vip/
10. Mahesh G, Sakthivel SM (2015) Verification of memory transactions in AXI protocol using system verilog approach. In: International conference on communication and signal processing (ICCSP) 2015, Melmaruvathur India