

UVM Based Master and Slave AMBA AXI Verification IP (VIP) For Verification of Soc Architecture

Dr. Neelam Rup Prakash, Shivani Malhotra

Dr. Neelam Rup Prakash is a professor in Electronics and Communication Department at PEC University of Technology, Sector 12 , Chandigarh , India.

Shivani Malhotra is pursuing Masters of Engineering in VLSI Design at PEC University of Technology, Sector 12, Chandigarh , India.(Phone : 750800228

Corresponding author: Dr. Neelam Rup Prakash

ABSTRACT

Advanced Microcontroller Bus Architecture(AMBA) Advanced eXtensible Interface (AXI) was developed by ARM Ltd and is extensively used in System-on-chip (SoC) designs as an on-chip bus. It is fit for the submicron interconnects which are very high in speed. In the verification strategy, Questa® Advanced Simulator has been used to verify AXI protocol under TSMC 0.18um Complementary Metal Oxide Semiconductor (CMOS) One-Poly-Six-Metal1P6M Technology[4]. A Verification Intellectual Property (VIP) is a pre-verified and reusable Intellectual Property that can generate tests for SoC verification due to its in-built verification infrastructure. Universal Verification Methodology (UVM) has been used for modelling the master and slave VIP.

KEYWORDS: Advanced Microcontroller Bus Architecture (AMBA), Advanced Extensible Interface (AXI),ONE-POLY-SIX-METAL Technology (1P6M) ,SoC (System-on-Chip),Verification Intellectual Property (VIP).

Date of Submission: 14-02-2018

Date of acceptance: 03-03-2018

I. INTRODUCTION

The AMBA 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 [5] 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. In the verification part [5], a testbench will be designed consisting of various components as shown in figure 1.

II. COMPARISON WITH PREVIOUS BUS NAMED ADVANCED HIGH PERFORMANCE BUS

Now if this advanced version with the previous one that is Advanced High Performance Bus [2] , many new features can be found present in it. They are :

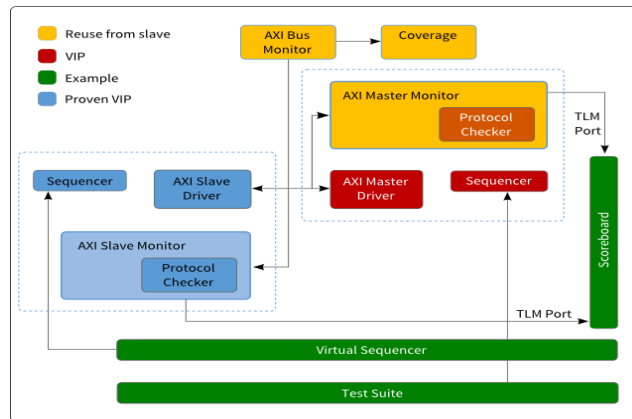


Figure 1. AXI VIP [6]

AMBA AHB has only one shared address bus and individual buses for reading and writing data whereas specifications of AMBA AXI are based on channels with five different channels for reading the address and data and writing the address and data.

In AMBA AHB, transmitting address of every entry of data is required whereas in AXI, the transmitting address of solely first entry of data on the bus is required. This is possible due to its burst mode feature.

In AHB, under the simple SPLIT and RETRY transaction scheme, transactions have to be completed in the original order whereas AXI has an outstanding feature in which multiple Out-of-Order transactions can be completed. [7]

There is no fixed burst mode in AHB but there is a fixed burst mode for input-output peripherals in AXI.

Simple protection unit and cache support is present in AHB but it is advanced in case of AXI.

No low-power interfaces are present in AHB but are there, in AXI.

III. AXI ARCHITECTURE

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. Now in this, the master and slave are interconnected by what is called an interconnect. [1]

In AXI protocol, interfaces are present as all possible mutual combinations among the master , slave and the interconnect.

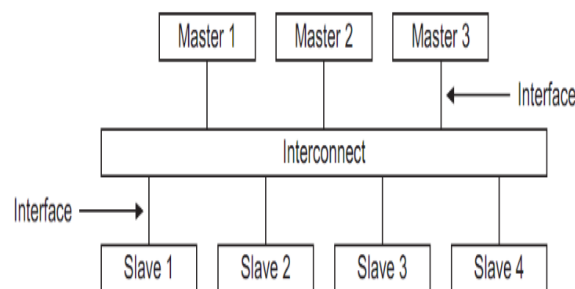


Figure 2. Master-Slave Interconnects [1]

We can make burst-based transactions in AXI and for that only start address needs to be issued whereas the ending address is not required for the same.

It includes independent read and write data channels that enables Direct Memory Access (DMA) which is lower in cost. The address information is issued before the transfer of actual data.

Regarding the transactions, the AMBA AXI protocol supports multiple outstanding transactions. All the transactions are in bursts. It supports bursts of variable-length and the transfer size of bursts ranges from 8-1024 bits. Bursts are of 3 types:

- Wrapping
- Incrementing
- Non-incrementing bursts

IV. ORDERING MODEL

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 AXI protocol, we can complete the out-of-order transactions. These features enhance the performance of interconnects and increase the efficiency of system. [1]

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. Besides the write data, the master sends a WID for matching the AWID of the corresponding address.

BID - This ID tag is used for the write response. A BID is transported by the slave for matching the AWID and WID of the transaction to which it has to acknowledge.

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. The slave transports an RID for matching the ARID of the transaction to which it has to acknowledge.

V. ATOMIC ACCESS

One atomic access type – address-lock type is proposed to support critical/non-critical data access simultaneously according to address range. [1] It also can improve bandwidth performance and reduce software cost. To allow the execution of atomic access primitives, 2 signals ARLOCK[1:0] and AWLOCK[1:0] are present, combinations of which provide various accesses.

Table 1. Atomic Access

ARLOCK[1:0] AWLOCK[1:0]		Access
0	0	Normal access
0	1	Exclusive access
1	0	Locked access
1	1	Reserved

Exclusive access: In this mechanism, semaphore type operations can be executed and there is no requirement that the bus remains engaged to a specific master during the operation. A master can carry out an exclusive read from an address location and subsequently can perform an exclusive write to the same address location in order to complete the exclusive operation.

Locked access: When the ARLOCK[1:0] or AWLOCK[1:0] signals indicate that it is a locked transfer and till the time an unlocked transfer from a particular master doesn't get completed, the interconnect must make sure that only that master is permitted to enter the slave region. [9]

VI. SYSTEM CACHE SUPPORT

In cache-support signal feature of the AXI protocol, the master allocates bufferable and cacheable attributes of a transaction to a system-level.[1]Cache information signals which are named as ARCACHE[3:0] and AWCACHE [3:0] provide support for system level caches and other elements which enhance the performance. Now these signals ARCACHE and AWCACHE are 4 bit signals and every bit is allocated some task particular as mentioned above. [3] The tasks are as follows:

BIT0 :Bufferable (B) bit

ARCACHE[0] AWCACHE[0] : When BIT0 is HIGH, it means that the transaction which is reaching its final destination can be delayed by interconnect or any component for an arbitrary number of cycles.

BIT1 : Cacheable (C) bit

ARCACHE[1] and AWCACHE[1] : When BIT1 is HIGH, it means that it is not necessary that transaction at the last destination should looklike the attributes of the original transaction. In case of write transactions, this means that any number of different write transactions can be clubbed together. And in case of read transactions, it means that we can fetch a location in advance for several read transactions.

BIT2 : Read Allocate (RA) bit

ARCACHE[2] and AWCACHE[2] : When BIT2 is HIGH, it means that if a read transfer takes place and it fails to hit the cache then it should be allotted. The RA bit must be LOW if the C bit is low.

BIT3 : Write Allocate (WA) bit

ARCACHE[3] and AWCACHE[3] : When BIT3 is HIGH, it means that if a write transfer takes place and it fails to hit the cache then it should be allotted. The WA bit must be LOW if the C bit is low.

For deciding if a transaction should be stored (cached) this bit should be synchronized with the second and third bit ,that is the Read Allocate (RA) and Write Allocate (WA) bits.

VII. PROTECTION UNIT SUPPORT

For assisting the system designs which are complex in nature, it is mandatory for all components in the system to protect it from unlicensed transactions. [1]

The signals named AWPROT[2:0] or ARPROT [2:0] give three separate levels of access protection. Now all three bits of these signals are assigned some functions separately in 3 different modes when these signals are used in conjunction with each other. They are as follows:

BIT0 : Normal or privileged

ARPROT[0] and AWPROT[0] : Bit0 if low specifies a normal access and if HIGH ,specifies a privileged access. This is used by some masters to specify their mode of processing. The privileged mode has considerable access within a system.

BIT1 : Secure or non-secure

ARPROT[1] and AWPROT[1] : Bit1 if low specifies a secure access and if HIGH , specifies a non-secure access. This is used in systems where an effective distinction between processing modes is required.

BIT2 : Instruction or data

ARPROT[2] and AWPROT[2]: Bit2 if low specifies a data access and if HIGH , specifies an instruction access. This bit specifies if the transaction is an instruction or a data access.

Table 2 : Access based on Protection bits

Bit0	Privileged access if HIGH
	Normal access if LOW
Bit1	Non-secure access if HIGH
	Secure access if LOW
Bit2	Instruction access if HIGH
	Data access if LOW

VIII. RESPONSE SIGNALING

The AXI protocol allows various signals for providing responses for both read and write transactions. For read transactions the response information from the slave is passed along with the read data only and for write response, the information is transmitted with the write response channel. Two signals named as RRESP[1:0] and BRESP[1:0] with four different combinations are there for giving four responses.

The AXI protocol responses are as follows :

Table 3.Slave Responses

Bit	Response
BIT0	OKAY
BIT1	EXOKAY
BIT2	SLVERR
BIT3	DECERR

OKAY: It is the simplest response specifying about the accomplishment of a normal access. It can also specify an exclusive access or its failure to a slave that does not carry the exclusive access. So these three conditions yield an OKAY response.

EXOKAY: It stands for Exclusive Access Okay and specifies that out of the read or write , only one part of this response has been accomplished.

SLVERR: This is known as slave error and is used when the ingress has hit the slave in a good time but the slave wants an error condition to recur to the master who is generating the signals.

DECERR: This is known as decode error which often gets triggered by an interconnect component to indicate the absence of slave at the transaction address.

One of the most important feature regarding the responses is that for a write transaction that there is no need to give response for each data transfer within the burst. Only one response is given for the whole burst regardless of the size of burst. [1]

However in case of read transaction, different responses for different transfers the can be given by the slave inside a burst which have been just discussed above.

IX. CONCLUSIONS

This generation of AMBA has lot many advanced features as compared to previous versions like APB, AHB but still it can be improved on grounds of coherency. Improvements can be made where behavior of individual components affect the overall performance and efficiency of the system.

REFERENCES

- [1]. ARM, AMBA AXI Protocol Specification [Online]. Available: <http://www.arm.com>ARM.
- [2]. AMBA AHB Protocol Specification [Online]. Available: <http://www.arm.com>.
- [3]. https://www.xilinx.com/products/intellectual-property/axi_systemcache.html.
- [4]. <https://www.cmc.ca/en/WhatWeOffer/Products/CMC-00200-00484.aspx>.
- [5]. R. Saleh,S. Wilton and S. Mirabbasi "System-on-Chip: Reuse and Integration," in Proceedings of the IEEE (Volume: 94, Issue: 6, June 2006).
- [6]. <http://www.testandverification.com/solutions/vip/axi-vip/>.
- [7]. Chin-Yao Chang, Kuen-Jong Lee, "On Deadlock Problem of On-Chip Buses Supporting Out-of-Order Transactions," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems (Volume: 22, Issue: 3, March 2014).
- [8]. Questa@ Advanced Simulator [ONLINE] Available: <http://go.mentor.com/4gvo9A>.
- [9]. Silberschatz "A Case for Non-Two-Phase Locking Protocols that Ensure Atomicity," in: IEEE Transactions on Software Engineering (Volume: SE-9, Issue: 4, July 1983).

Dr. Neelam Rup Prakash " UVM Based Master and Slave AMBA AXI Verification IP (VIP) For Verification of Soc Architecture" International Journal of Computational Engineering Research (IJCER), vol. 08, no. 02, 2018, pp. 19-23.