

Design and Development of AMBA-AXI4Bus Based System on Chip VIP Protocol Using UVM

Narra Venkata Krishna

M.Tech, (VLSI SYSTEM DESIGN), JNTUK, Kakinada. East Godavari dist., A.P, India.

ABSTRACT: The Advanced microcontroller bus architecture (AMBA) based AXI protocol supports high-performance, high-frequency system designs. It is suitable for high-bandwidth and low-latency designs and provides high-frequency operation without using complex bridges. The AXI protocol provides complexity, verification intellectual property (VIP), flexibility in the implementation of interconnect architectures and is backward-compatible with existing AHB and APB interfaces. The Design of the AXI4 Slave Interface and implementation using Verilog RTL coding. AXI slave interface can be used to connect different peripherals into AMBA based processors without using bridge. The developed slave interface can also be used to connect different peripherals like GPIO, SPI, I2C, UART etc., Advanced microcontroller bus architecture (AMBA) protocol provides verification of protocol compiling, enabling the comprehensive testing of interface verification intellectual property (VIP) blocks and system-on-chip (SoC) design. The AMBA advanced extensible interface 4 (AXI- 4) update to AMBA AXI3 includes: the support for burst lengths up to 1024 beats. It is updated write response requirements and removal of locked transactions. AXI verification has become the dominant cost in the design process. This paper proposes design environment, how to build up the verification environment of AXI bus using universal verification methodology is introduced. Functional coverage, code coverage, score-boarding and assertions are implemented with the proposed integrated verification environment.

KEYWORDS: Advanced Microcontroller Bus Architecture (AMBA), Advanced Extensible Interface (AXI), Verification Intellectual Property (VIP), Functional Coverage, IP, Multiple Outstanding Transaction, Out of Order Transaction Completion, Xilinx ISE, SoC, UVM, Questa sim.

I. INTRODUCTION

The IP protocol used many more SoC's today as Advanced Extensible Interface and is part of the ARM Advanced Microcontroller Bus Architecture (AMBA). It is especially prevalent Xilinx's Zynq devices, provides the interface between the processing system and programmable logic sections of the chip. The IP cores all shared a common interface like AXI. Known as the differences between these devices, I interested in why each IP Core was able to share this common interface. Reading more into the technology I found out just why AXI has become the most widespread AMBA AXI 4 interface.

The protocol simply sets up the rules for different modules on a chip communicates with each other, requiring a handshake-like procedure before all transmissions. Having a protocol such as this allows a true "system" rather than a "collection" of modules to be established as the protocol connects and provides an effective medium for transfers of data between the existing components on the chip.

The complexity inside of the chip has been increased with the advancements in Semiconductor industry. This led to the invention of SoC. SoC consists of Processor, Peripherals like Timers, Universal Asynchronous Receiver Transmitter (UART), I2C, Serial Peripheral Interface (SPI), General Purpose Input/output (GPIO), Analog to Digital Converter (ADC), Digital to Analog Converter (DAC), memory peripherals etc. To complete design in a less time, most of the SoC's are developed Verification IP's, which is used as a building block for ASIC and FPGA based designs. In a SoC based designs, these pre-designed IP cores are becoming popular. Chips must be verified for their functional correctness i.e., whether they are working Per specification given or not.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

II. AXI PROTOCOL

AMBA AXI protocol support's high-performance, high-frequency system design.

The AXI protocol:

- is suitable for high-bandwidth and low-latency design
- provide high-frequency operation without using complex bridges
- meet the interface requirements of a wide range of components
- is suitable for memory controllers with high initial access latency
- provide flexibility in the implementation of interconnect architectures
- is backward-compatible with existing AHB and APB interface.

The features of the AXI protocol are:

- separate address and control and data phases
- it supports for unaligned data transfers, using byte strobes
- uses burst-based transactions with only the start address issued
- separate read and write data channels, that can provide low-cost Direct Memory Access (DMA)
- it supports for issuing multiple outstanding addresses
- it supports for out-of-order transaction completion
- it permits easy addition of register stages to provide timing closure.

The AXI protocol includes the optional extensions that cover signalling for low-power operation.

III. AXI ARCHITECTURE

In this protocol is burst-based and defines the following independent transaction channels:

- read address
- read data
- write address
- write data
- write response

An address channel carries control information that describe the nature of the data can be transferred. The data can be transferred between master and slave channels:

- A write data channel to be transfer data from the master to the slave. In the write transaction, the slave uses the write response channel to be signal the completion of the transfer to the master.
- A read data channel to be transfer data from the slave to master.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

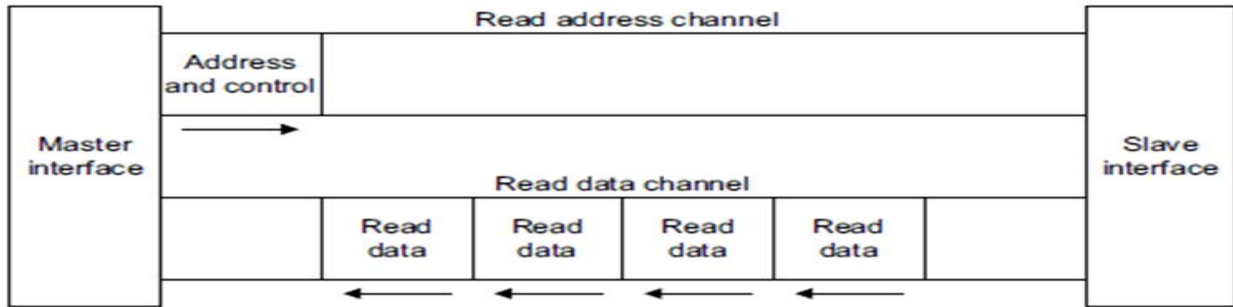


Fig: Read channel architecture.

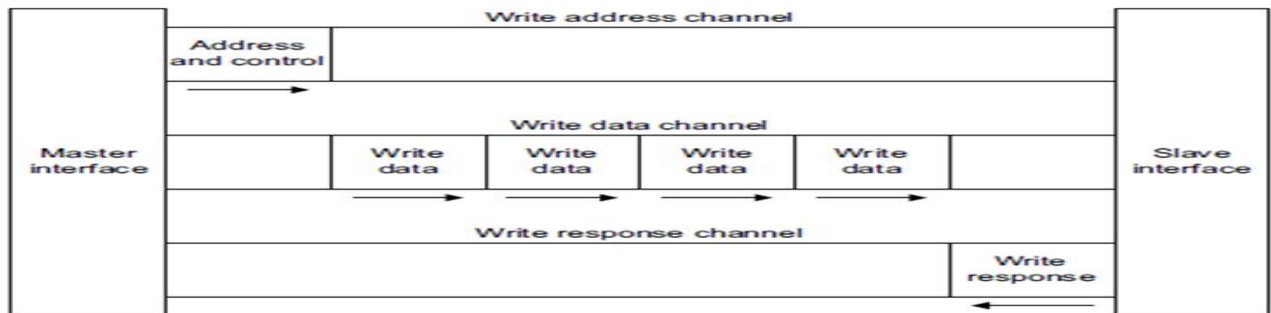


Fig: write channel architecture.

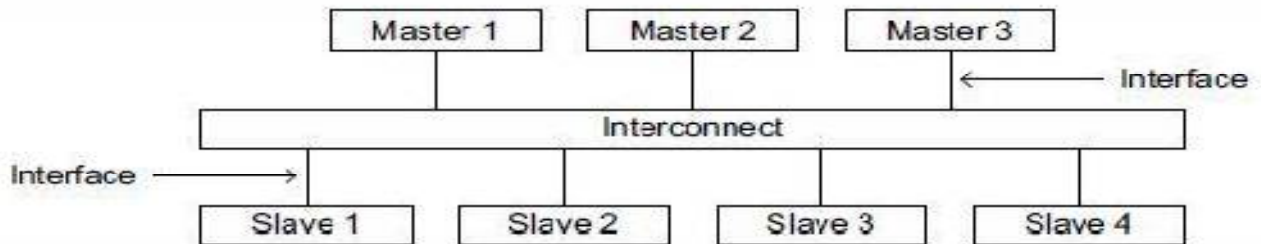


Fig: AXI System Topology

IV. DESIGN OF AXI PROTOCOL

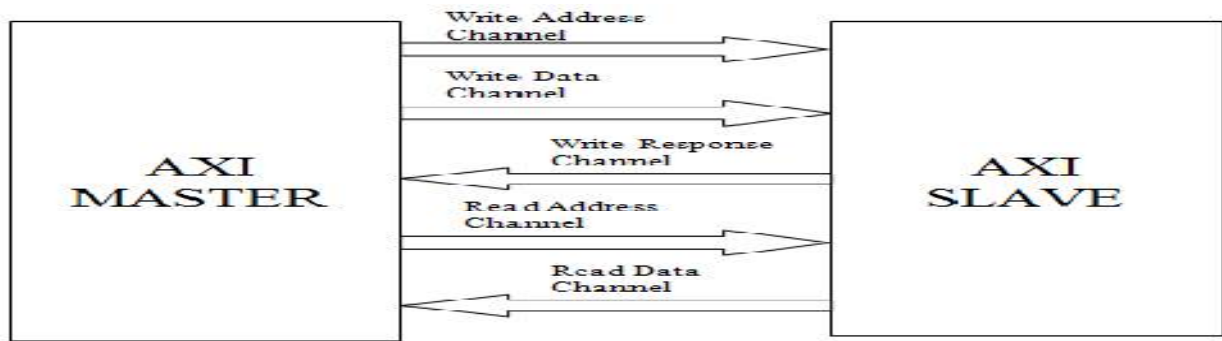
The AXI 4 slave is designed with operating frequency of 100MHz, which gives clock cycle of duration 10ns and it supports maximum of 1024 data transfers per burst. The AMBA AXI4 system component consists of a master and a slave. There are five different channels between AXI master and AXI slave write address channel, write data channel, read data channel, read address channel, and write response channel.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

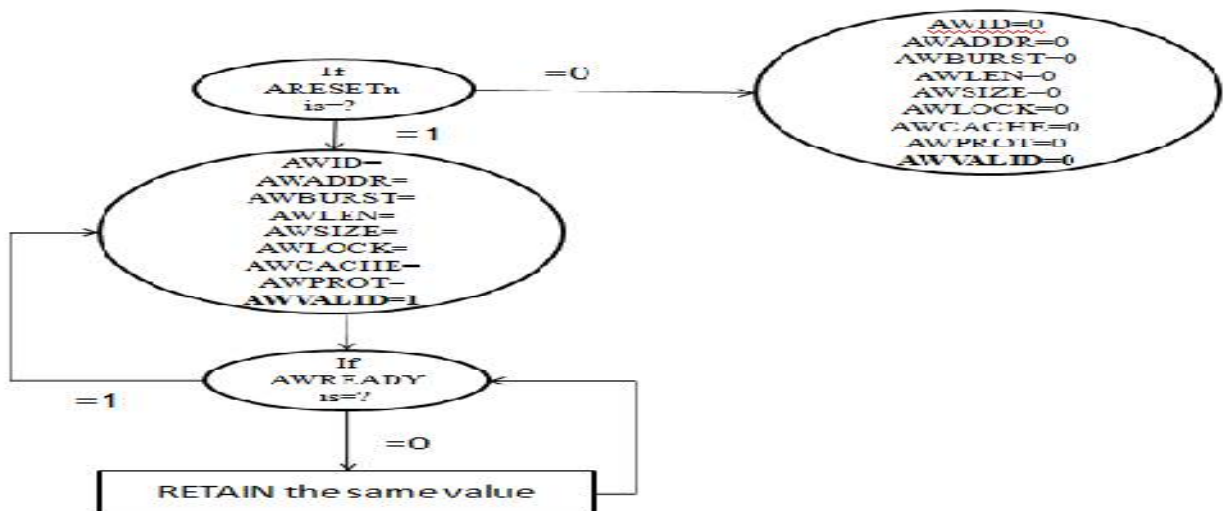


Block Diagram of a system

The AXI protocol, all transfers are done using hand shake mechanism. Each channel uses the same VALID/READY handshake to transfer control and data information. In this two-way flow control mechanism enables both the master and slave to control the rate at which the data and control information. Source generates the VALID signal to indicate the data or control information is available. The destination generates the READY signal to indicate that it accepts the data or control information. Transfer occurs only when both the VALID and READY signals are HIGH. There must be no combinatorial paths between input and output signals on both master and slave interface.

4.1 Address Write Channel (AW Channel)

The AXI_MASTER drives the write command signals only when ARESETn is HIGH, else it drives all signals as zero. The address write command signals driven by the AXI_MASTER is - AWID, AWADDR, AWBURST, AWLEN, AWSIZE, AWCACHE, AWLOCK, AWPROT, with AWVALID as HIGH indicating that the driven signals are valid. The AXI_MASTER does not drive AWVALID signal as LOW, until it receives AWREADY signal, which is driven by the DESTINATION_SLAVE, indicating that it has received the address write command signals. If AWREADY is LOW, then AXI_MASTER retains the same values.



State diagram of Address Write Channel

International Journal of Innovative Research in Science, Engineering and Technology

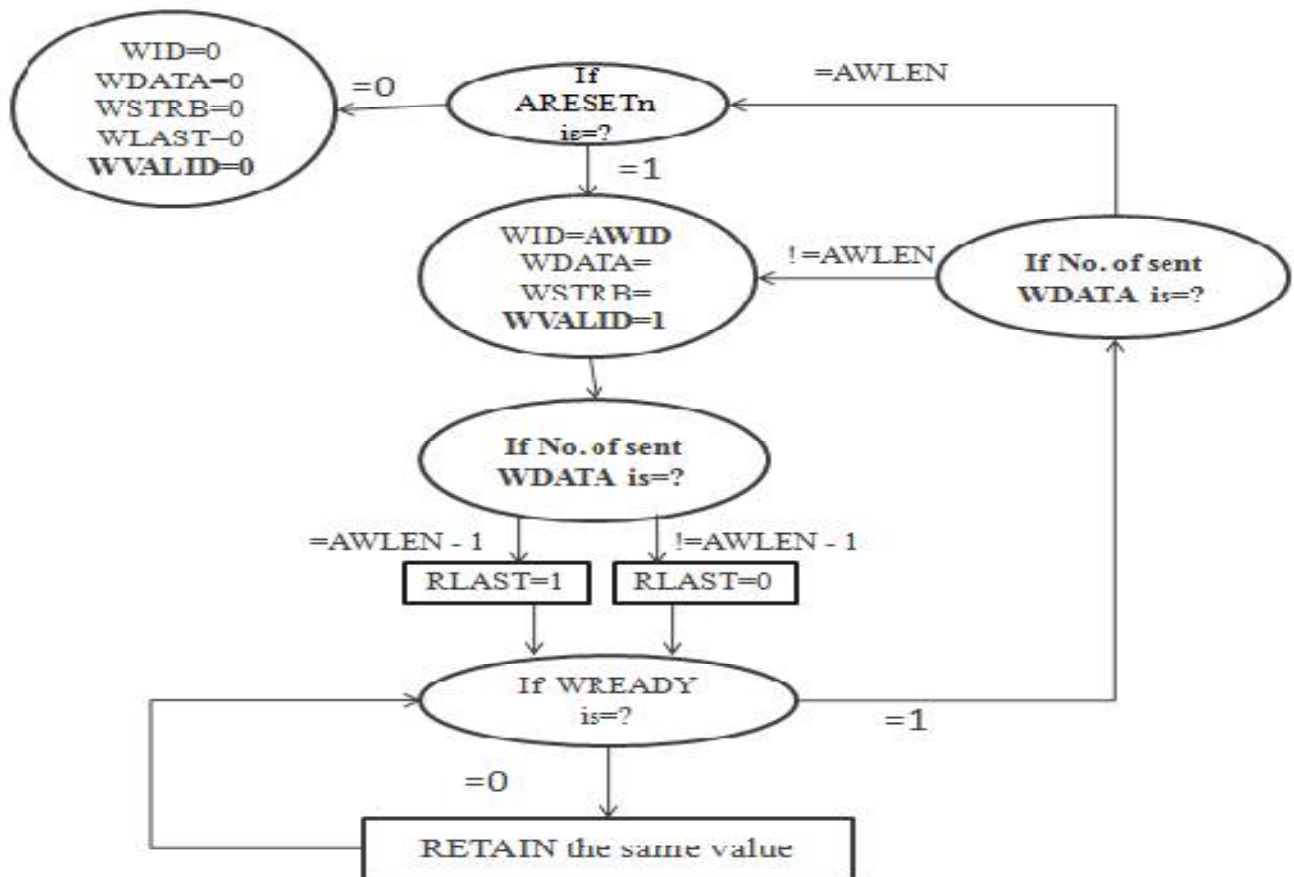
(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

4.2 Write Data Channel (W Channel)

AXI MASTER drives these Write Data signals, after sending the write address command signal. It drives these signals, only when ARESETn is HIGH, otherwise it drives all signals to zero. AXI MASTER drives the WDATA signal with WVALID as HIGH, it holds the same value until it receives the WREADY signal. If WREADY is HIGH, it drives the next WDATA. AXI MASTER drives the AWLEN No. of data. While driving the last data, it drives the WLAST as HIGH.



State diagram of Write Data Channel

4.3 Write Response Channel (B Channel)

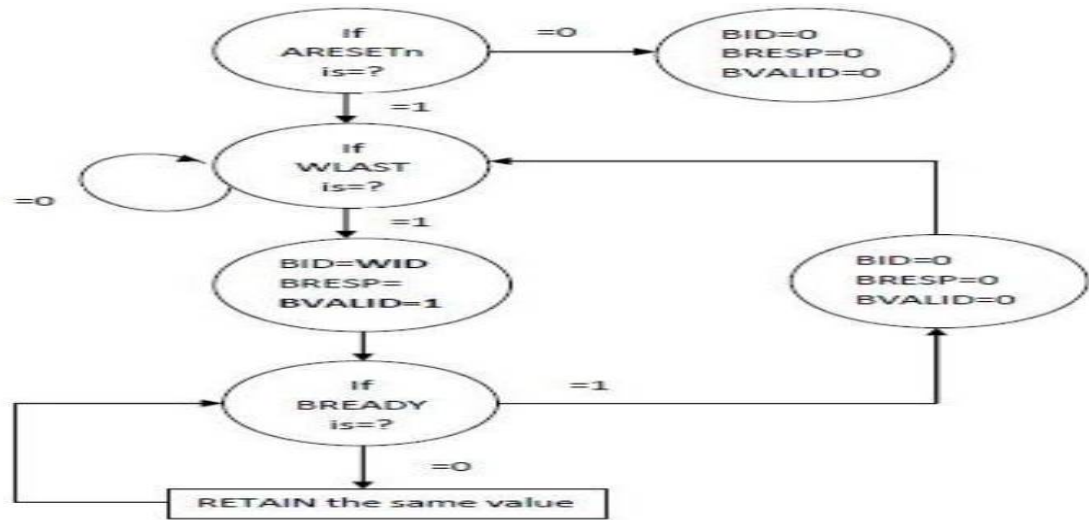
The AXI channel DESTINATION_SLAVE drives these Write Response signals, only when ARESETn is HIGH, otherwise it drives all signals as zero. DESTINATION_SLAVE waits for WLAST signal. After receiving the WLAST signal, it drives these response signals, with BVALID as HIGH. It holds the same value until it receives the BREADY signal from the AXI MASTER. If BREADY is HIGH, it drives all the signals as zero, at next positive edge of ACLK, otherwise it retains the same value.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

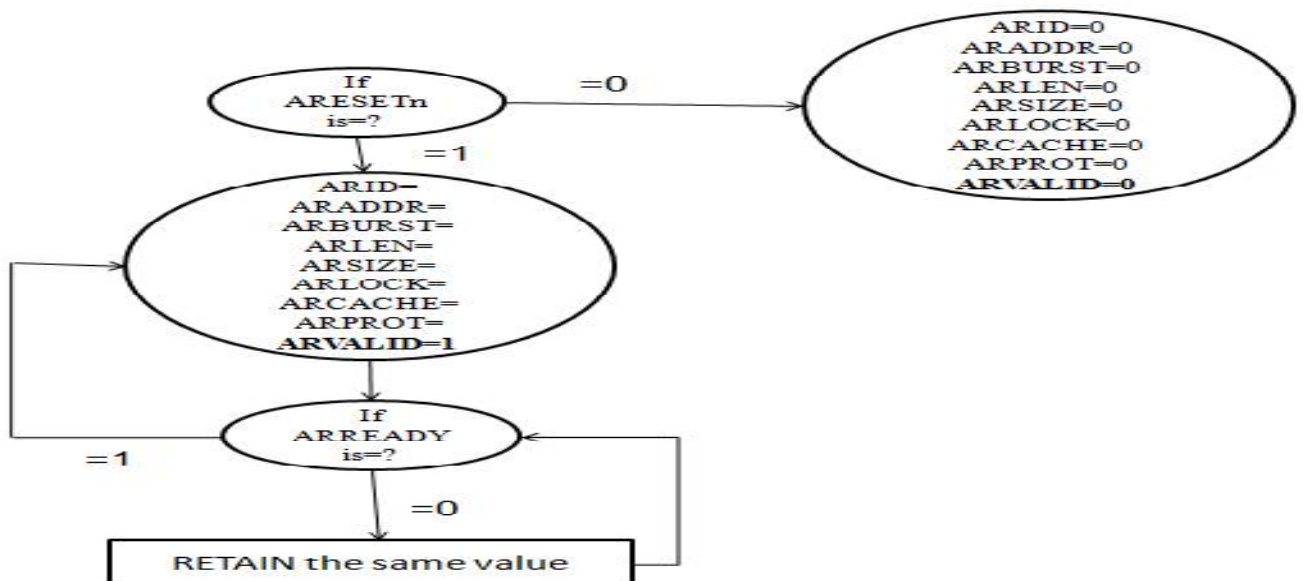
Vol. 6, Issue 4, April 2017



State diagram of Write Response Channel

4.4 Address Read Channel (AR Channel)

The AXI_MASTER drives the command signals only when ARESETn is HIGH, else it drives all signals as zero. The address read command signals driven by the AXI_MASTER is - ARID, ARADDR, ARBURST, ARLEN, ARSIZE, ARCACHE, ARLOCK, ARPROT, with ARVALID as HIGH indicating that the driven signals are valid. AXI_MASTER does not drive ARVALID signal is LOW, until it receives the ARREADY signals, which is driven by the SOURCE_SLAVE, indicating that, it has received the address read command signals. If ARREADY is LOW, then AXI_MASTER retains the same values.



State diagram of Address Read Channel

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

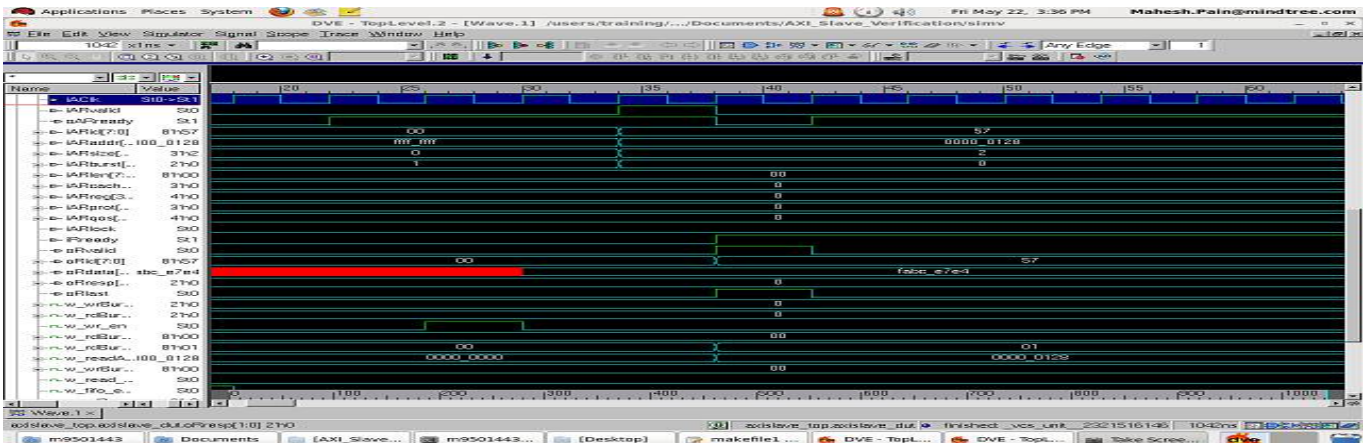


Fig: Fixed Type Read

VI. UVM BASED VERIFICATION

AXI Master device is one who initiates the transfers and one which has the decision taking capability. A Slave device is one which has no decision taking capability and it should follow the instructions of master. In AXI protocol, there are five channels which work independent to each other. So, there is no dependency between each channel. The AXI protocol is meant to offer high data transfer speed. The reasons for this are as follows. i.e.

1. The write data channel performs write transaction and concurrently can also perform read transaction by using read data channel, at the same time. So, both write and read operations can be performed individually and concurrently. So, this is one of the reason for which AXI offers more data transfer speed.
2. The AXI-4 protocol supports for issuing multiple outstanding transactions. It means that before the completion of one transaction, master can initiate one more transaction. These ways number of transactions, so that all these transactions which are initiated by master can be kept in a pipeline. By this way data transfer speed increases and it also improves the efficiency of the bus. So, this is the reason for which AXI is best suitable for memory devices which has high initial access latency. This will avoid a high initial access latency slave blocking the channel. By this way, the performance of the bus also will be increased.

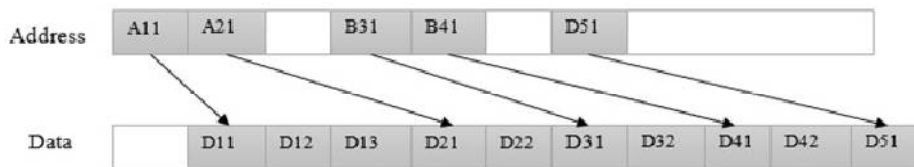


Diagram which explains Multiple Outstanding Addresses Transaction scenario

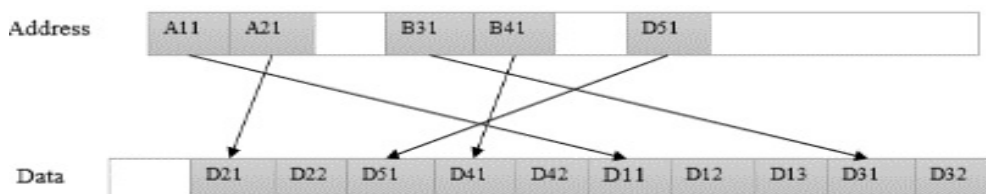


Diagram which explains Out of Order Transaction completion scenario

International Journal of Innovative Research in Science, Engineering and Technology

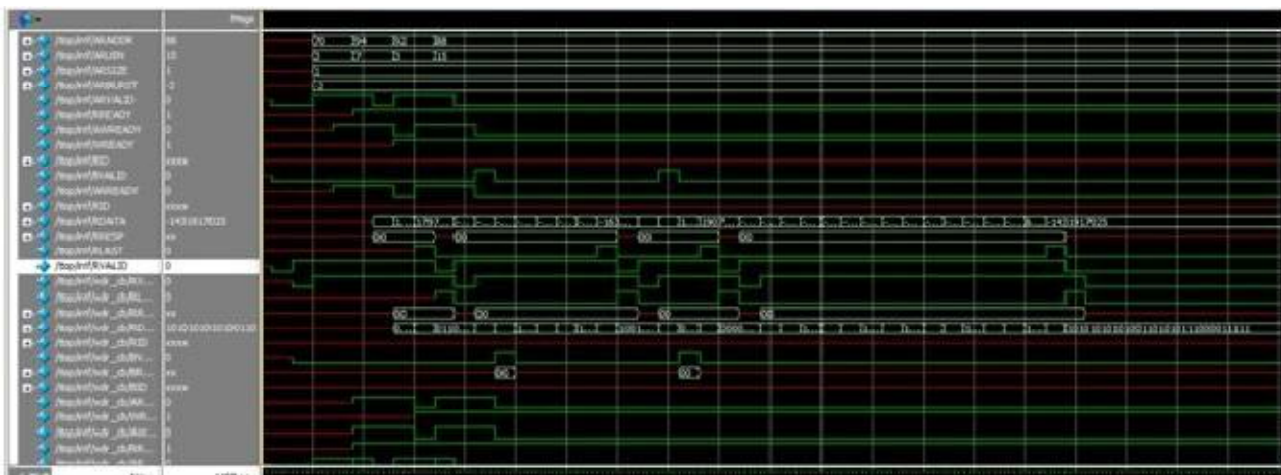
(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

control signals like ARSIZE, ARBURST, ARLEN etc. to Slave. The slave will calculate the next addresses per the control information sent by master. Now slave will be send read data to Master by read data channel. Slave must send acknowledgment for each read data. So, slave has the capability send both read data and acknowledgment by single channel i.e. read data channel. So, there is no need of read response channel for AXI.

In order to simplify the design generator in the slave, the address is allocated to a slave in 4KB unit. Bursts cannot cross 4KB boundary. In AXI, termination of early burst is not supported. WLAST signal is used to indicate last transfer in the write burst and RLAST signal is used to indicate the last transfer in the read burst.WSTRB signal is used to be indicate which byte lanes hold valid data in write burst. For every 8-bits of write data, there is one write strobe bit. Burst size will be intimate about the maximum number of data bytes to transfer in each beat of transfer within a burst. For incrementing type of the burst, the first transfer can have unaligned transfers, but the rest all transfers in a burst should be aligned transfers. For the wrapping type of the burst, all transfers in a burst should be of aligned transfers. For fixed type of burst, the address remains constant within a burst and every transfer should use the same byte lane. AXI also supports for aligned and unaligned transfers. For Wrapping type of bursts, the burst length should be 2,4,8,16 only. When data size is less than 32-bit, it is called as narrow transfer.



Waveform which explains Read channel architecture.

In this work, the VIP is developed without having any Device or Design Under test(DUT) for reference. So, this VIP is having one Master agent and one Slave agent and both of the agents are ACTIVE agents i.e. they have all the driver, monitor and sequencer. Each and every transactions initiated by Master agent are assumed that, they are going to a different slave. The number of slaves can be configured from test_case by over_writing the required number of slave's value into the "Number_of_slaves" parameter inside the configuration class. Here, the master agent exactly mimics the master DUT, and slave agent exactly mimics the slave DUT. If the VIP is ready, this VIP will work as per the AXI-4 protocol. Then, if master component (Master DUT) is given, then it should have a slave agent to have communication. Then make Master agent in the VIP as PASSIVE and keep the slave agent as ACTIVE only. Master DUT will drive the data, so only monitor is enough to just monitor those driven signals. If slave component is given, then make the slave agent in VIP as a PASSIVE agent and keep the master agent as ACTIVE only. If the master component and slave component are given, then make both master agent and slave agent as PASSIVE agents. We just monitor those signals and verify in the scoreboard.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

| Coverage Summary by Structure: | | Coverage Summary by Type: | | | | |
|----------------------------------|-----------------------|---------------------------|---------------|---------------|-----------------|-----------------------|
| Design Scope ▾ | Coverage (%) ▾ | Weighted Average: | | | | 100.00% |
| uvm_pkg | 100.00% | Coverage Type ▾ | Bins ▾ | Hits ▾ | Misses ▾ | Coverage (%) ▾ |
| uvm_callbacks | 100.00% | Covergroup | 9 | 9 | 0 | 100.00% |
| uvm_phase | 100.00% | Assertion Attempted | 28 | 28 | 0 | 100.00% |
| uvm_component | 100.00% | Assertion Failures | 28 | 0 | - | 0.00% |
| axi_pkg | 100.00% | Assertion Successes | 28 | 28 | 0 | 100.00% |
| master_seq_base | 100.00% | | | | | |
| slave_seq_base | 100.00% | | | | | |
| virtual_seq_base | 100.00% | | | | | |
| score_board | 100.00% | | | | | |

Functional Coverage report generated by Questa Sim.

VII. TEST CASES

This paper the working of all the five channels are verified. Basic Write and read scenarios for all possible burst types and burst sizes, Multiple outstanding addresses transaction and Out of Order transaction completion scenarios are also verified successfully by using UVM.

VIII. CONCLUSION

The VIP for AMBA AXI-4 protocol is developed and it can be configured and inserted into a test bench for verifying a design. This VIP is used to verify write and read transactions for all possible burst types and burst sizes, multiple outstanding addresses transactions, Out of order transaction completion successfully.

REFERENCES

1. AMBA AXI-4 Specification, Copyright ARM Limited. http://www.gstitt.ece.ufl.edu/courses/eel4720_5721/labs/refs/AXI4_specification.pdf.
2. Chen CH, Iu JC, Huang II. A Synthesizable AXI Protocol Checker for SoC Integration. IEEE Transl, ISOC. 2010; 8:103-6.
3. "AMBA Peripheral Bus Controller Data Sheet" Copyright © 1996 Advanced RISC Machines Ltd (ARM). Hu Yueli; Key Lab. of Adv. Display & Syst. Application., Shanghai Univ., Shanghai, China ; Yang Ben "Building an AMBA AHBC compliant Memory Controller", Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference, 6-7 Jan. 2011.
4. AMBA AXI 4 and ACE Protocol Specification" 28 October 2011 D Non-Confidential First release of AMBA AXI 4 and ACE Protocol Specification.
5. "Verilog-AMS Language Reference Manual". Release 2.4, Accellera Systems Initiative, 06-2014.
6. "Verilog-AMS Language Reference Manual", Release 2.3.1, Accellera Systems Initiative, 06-2009
7. "Verilog-A Language Reference Manual Analog Extensions to Verilog HDL", Version 1.0, Open Verilog International August 1, 1996
8. Manjula, R.B. ; Manvi, S.S. ; Kaunds, P. "Data transactions on system-on-chip bus using AXI4 protocol" Recent Advancements in Electrical, Electronics and Control Engineering (ICONRAEeCE), 2011 International Conference, 15-17 Dec. 2011.
9. AMBA AXI Specification (AR500-DA-10008)" 16 June, 2003-A, First release, Copyright ARM Limited 2003
10. "AMBA™ Specification (Rev 2.0)" 13th May 1999-A, First release, Copyright ARM Limited 1999.
11. "AMBA Peripheral Bus Controller Data Sheet" Copyright © 1996 Advanced RISC Machines Ltd (ARM).
12. ARM, AMBA AXI protocol specifications, Available at, <http://www.arm.com>, 2003.
13. AMBA AXI Protocol Version: 2.0 Specification, ARM Ltd, pp. 1-1.

International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: www.ijirset.com

Vol. 6, Issue 4, April 2017

14. Ref Shaila S Math, Manjula R B "Survey of system on chip buses based on industry standards," Conference on Evolutionary Trends in Information Technology(CETIT), Belgaum, Karnataka, India, pp. 52, May 2011.
15. Ms. Anusha Ranga, Mr. L. Hari Venkatesh, Mr.Venkanna, "Design and Implementation of AMBA-AXI Protocol using VHDL for SoC Integration," in International Journal of Engineering Research and Applications, Vol. 2, Issue4, July-August 2012, pp.1102-1106.
16. C. Spear," A Guide to Learning the Testbench Language Features",in System Verilog for verification, 2nd ed., Springer Publishing Company, Incorporated,2008,pp. 11-18.
17. L. Tao, X. Tong, Z. Yang, L. Huawei, and L. Xiaowei,"Bug analysis and corresponding error models in real designs", in IEEE International High Level Design Validation and Test Workshop,2007,pp. 59-64.
18. Samir Palnitkar, Verilog HDL: A Guide to Digital Design and synthesis, 2nd ed, Hall PTR Pub, 2003
19. Silicore Corporation, Wishbone system-on-chip (soc) interconnection Architecture for portable IP cores.
20. IBM, Core connect bus architecture. IBM Microelectronics[Online].Available: <http://www.ibm.com/chips/products/coreconnect>, 2000.