DEVELOPMENT OF AMBA-AHB AND AXI PROTOCOLS FOR ADVANCED MICROCONTROLLER SYSTEMS

M.GURUNADHA BABU
RESEARCH SCHOLAR
OPJS UNIVERSITY, RAJASTHAN
mgurunadha babu@gmail.com

DR. VAIBHAV BANSAL
PROFESSOR
OPJS UNIVERSITY, RAJASTHAN

DR. S. BALAJI
PROFESSOR & DEAN
CMRIT, HYDERABAD

ABSTRACT

Coverage-driven Verification is a verification methodology in which coverage planning precedes the rest of the verification process. Coverage planning means defining a strategy for measuring verification progress – employing functional, code and assertion coverage – and the tactics that will be employed to implement it. This paper describes the use of coverage driven verification for AMBA systems. The first part contains a general discussion of the digital verification method as of today and the possible mapping to an AMBA design scenario. The second part describes a demo example that shows the coverage driven verification approach for a true AMBA AXI domain system. The system contains electrical as well as mechanical and software parts and is therefore a good representation of today‟s complex system designs.

Writing assertions concurrently with the RTL design and keeping these assertions closely tied to the RTL code has been found to bring significant benefits in both the design and verification processes for digital hardware. The primary benefit is that assertions help to detect more functional bugs, detect them earlier in the process and detect them closer to their original cause. A secondary benefit is that the very act of formulating and writing assertions can give the designer a better understanding of the design, and hence uncover bugs in the specification or else avoid introducing bugs into the design in the first place. This paper describes assertion based system Verilog verification environment with a robust and widely used AMBA AHB bus protocol master-slave architecture.

INTRODUCTION

This paper describes the verification architecture for the SV-OVM based AHB Master Slave and AXI Master Slave using concepts such as Randomization, class based concepts, SystemVerilog and this paper will be used as reference for building the test bench environment and writing the test scenarios.

This paper contains brief description about the

- AHB Protocol Overview
- AXI Protocol Overview
- SystemVerilog Overview
- OVM Overview
- Verification Architecture of AHB and AXI Bus Protocols

SCOPE OF ACTIVITY

The scope of the activity is to completely verify the AHB Master Slave and AXI Master Slave with SYSTEMVERILOG and OVM by developing AHB Master, AHB Slave, Monitors, Tests and Sequences, AXI BFM,. The key elements of the Paper include

- Building OVM test Bench.
- Listing the design features.
- Defining the scenarios.
- Defining Monitors.
- Building SYSTEMVERILOG test Bench.
- Identifying the design features.
- Defining the test cases.
- Defining Scoreboards

FUNCTIONAL DESCRIPTION

AMBA

A widely used Advanced Microprocessor Bus Architecture (AMBA) aims at easing the component design by using the combination of interchangeable components in the SOC design. It supports the reusability of intellectual property components, so that a least part of the design can become a composition.

A master initiates the read and write operations by providing the address and control information to the interconnected design. Only single master can access

the system bus. When access is granted to any of the masters, the address (HADDR) and control operations are performed. Also other signals like burst, HTRANS and HREADY signals are activated to initiate read and write operations in AHB protocol.

The slave responds to the read and write operations of the master within the given address space range. Also, it acknowledges to the master whether the read and write operations are successfully implemented. A particular slave is selected by HSEL signal and data write (HWDATA) and read (HRDATA) operations are performed. After data is read by the slave through read signal, it acknowledges to the master by respective signal.

AXI

AXI Master converts the AXI bus transfers to AXI SLAVE protocol. Details of DUV AXI Slave are as mentioned in AXI Protocol specification.

This Paper paper also serves as a means of Verification Plan for Verifying AXI Protocol using SystemVerilog Language. Various tests cases are written from Master to the Slave to prove that the test bench environment developed works as per standard AXI Protocol.

The environment (env) is the top-level component of the OVC. It contains one or more agents, as well as other components such as a bus monitor. The env contains configuration properties that enable you to customize the topology and behavior and make it reusable. For example, active agents can be changed into passive agents when the verification environment is reused in system verification. The structure is of a reusable verification environment. Notice that an OVC may contain an environment-level monitor. This bus-level monitor performs checking and coverage for activities that are not necessarily related to a single agent. An agent's monitors can leverage data and events collected by the global monitor.

OVM components communicate via standard TLM interfaces, which improve reuse. Using a SystemVerilog implementation of TLM in OVM, a component may communicate via its interface to any other component that implements that interface. Each TLM interface consists of one or more methods used to transport data. TLM specifies the required behavior (semantic) of each method but does not define their implementation. Classes inheriting a TLM interface must provide an implementation that meets the specified semantic.

The simulations are done using Mentor Graphics Modelsim and advanced Questasim simulator.

Acronyms

DUT

Device under test (DUT), also known as unit under test (UUT), Design under verification (DUV) is a term commonly used to refer to a manufactured product undergoing testing.

OVM

Open Verification Methodology is the newer verification methodology being used for verification environment development.

OVC

OVM Verification Components (OVC) sometimes referred to as verification IP (VIP). It is pre-verified, reusable, configurable, and plug-and-playable in verification environments.

AHB

AHB stands for Advanced High Performance Bus. It consists of an address phase and a subsequent data phase. Access to the target device is controlled through a MUX thereby admitting bus-access to one bus-master at a time. AHB-Lite is a subset of AHB formally defined in the AMBA 3 standard. This subset simplifies the design for a bus with a single master.

AXI

AMBA Advanced extensible Interface (AXI) Protocol - The ARM Advanced Microcontroller Bus Architecture (AMBA) is an open-standard, on-chip interconnect specification for the connection and management of functional blocks in system-on-a-chip (SOC) designs. AXI, the third generation of AMBA interface defined in the AMBA 3 specification, is targeted at high performance, high clock frequency system designs and includes features that make it suitable for high speed sub-micrometer interconnect.

- AXI BFM
  AXI Bus Functional Model
- AXI VIP

AXI Verification Intellectual Property

- Monitor

It monitors that the protocols at the input and output interfaces are not violated.

- SV

SystemVerilog is a Hardware verification language being used for verification environments recently.

- SVA

System Verilog Assertions used to model the protocol checks using simple syntax.

- AHB VIP

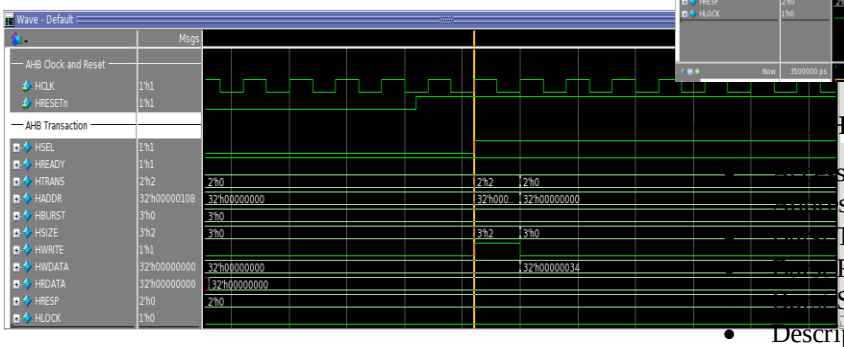
AHB Verification Intellectual Property

AHB Transactions

AHB Basic Write

AHB Write Transaction Actual Simulation Result is shown in below Figure. This transaction shows write and read signals of AHB with only write signals toggled.

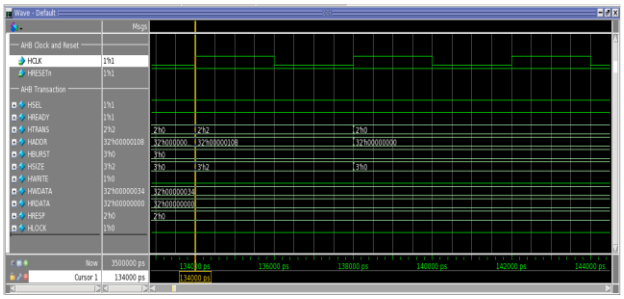
- Access type : Write -> Signals Write Address/Data/Response
- Burst Type : SINGLE –> HBURST =3'h0
- Burst Response : OKAY -> HRESP = 2'h0
- Burst Size : 32 -> HSIZE = 3'h2



AHB Basic Read

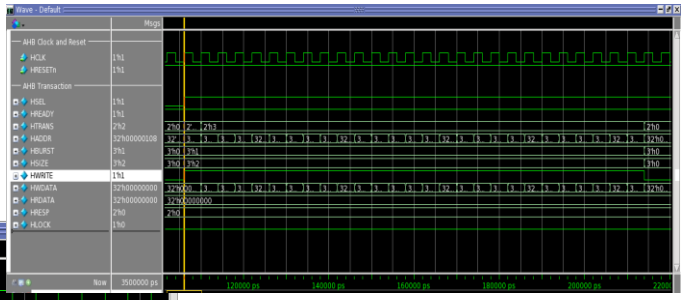
AHB Basic Read Transaction Actual Simulation Result is shown in below Figure. This transaction shows write and read signals of AHB with only read signals toggled.

- Access type : Read -> Signals Read Address/Data/Response
- Burst Type : SINGLE –> HBURST =3'h0
- Burst Response : OKAY -> HRESP = 2'h0
- Burst Size : 32 -> HSIZE = 3'h2



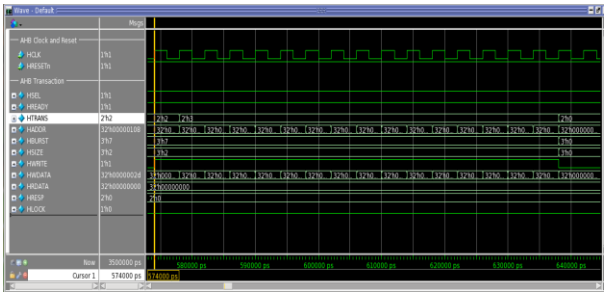
Test Case: AHB INCR Write of Undefined Length

- Access type : Write -> Signals Write Address/Data/Response
- Burst Type : INCR –> HBURST =3'h1
- Burst Response : OKAY -> HRESP = 2'h0
- Burst Size : 32 -> HSIZE = 3'h2
- Description: This test shows AHB INCR Write transaction for an undefined burst length. Also correct completion of the burst is indicated by HRESP signal. Slave is active indicated by HREADY.



HB INCR Write of Length 16

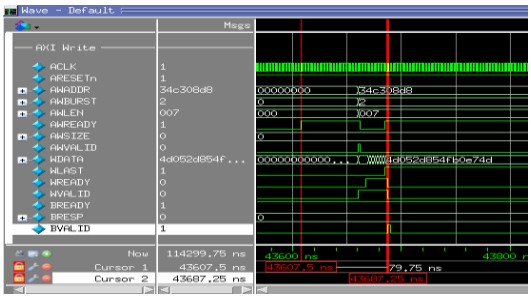
- s type : Write -> Signals Write ss/Data/Response
- Type : INCR –> HBURST =3'h7
- Response : OKAY -> HRESP = 2'h0
- Size : 32 -> HSIZE = 3'h2
- Description: This test shows AHB INCR Write transaction for burst length 16. Also correct completion of the burst is indicated by HRESP signal. Slave is active indicated by HREADY.



AXI Basic Transactions

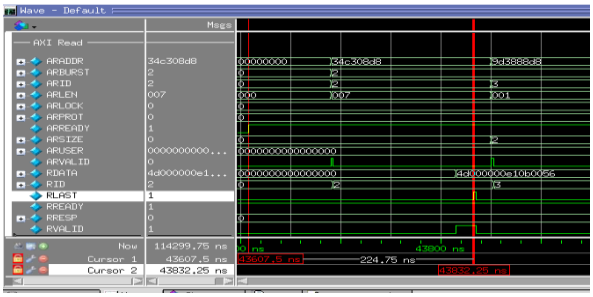
AXI Basic Write

AXI Write Transaction Actual Simulation Result is shown in below Fig. This transaction shows all the 3 channels of AXI.



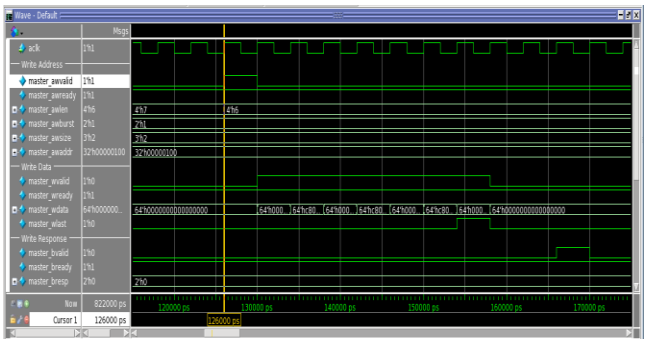
AXI Basic Read

AXI Basic Read Transaction Actual Simulation Result is shown in Fig. This transaction shows both the Read Address and Data Channel transactions.



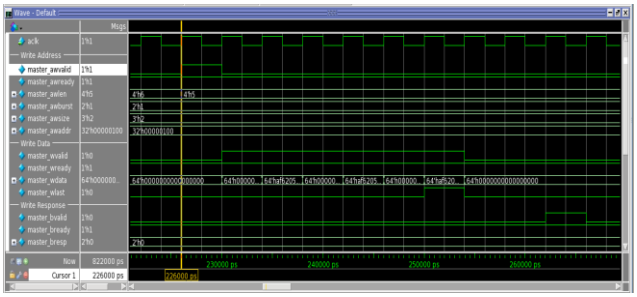
Test Case: AXI Single Write Length 7

- Access type : Write -> Channels Write Address/Data and Response
- Burst Type : INCR –> awburst=2'h1
- Burst Length : 7 -> awlen = 4'h6
- Burst Size : 32 -> awsize = 3'h2
- Description: This test shows AXI Write transaction with all 3 channels for a burst length of 7 where wvalid comes after wready. Also wlast indicates completion of last write transaction.



Test Case: AXI Single Write Length 6

- Access type : Write -> Channels Write Address/Data and Response
- Burst Type : INCR –> awburst=2'h1
- Burst Length : 6 -> awlen = 4'h5
- Burst Size : 32 -> awsize = 3'h2
- Description: This test shows AXI Write transaction with all 3 channels for a burst length of 6 where wvalid comes after wready. Also wlast indicates completion of last write transaction.



CONCLUSION

In this paper, functional verification of AMBA with AHB protocol is implemented using SV and OVM based scalable environment. All different components of AHB architecture i.e. master, slave, environment, tests and sequences are implemented. All major AHB scenarios are implemented and results are papered.

System-on-a-Chip (SOC) design has become more and more complexly. Verify a design effectively has become a serious challenge. In this paper, to build up the effective verification environment of AXI using SystemVerilog is introduced. Firstly, the design under verify (DUV) AXI bus is introduced. Then a comprehensive analysis of the verification plan has been made according to the protocol.
AMBA AXI4 is a plug and play IP protocol. It is released by ARM, defines both bus specification and a technology independent methodology for designing, implementing and testing customized high-integration embedded interfaces. The data is to be read or written to the slave is assumed to be given by the master and is read or written to a particular address location of slave.
In this Paper, an effective verification environment for AXI bus is developed with System Verilog. The multi-layer test bench is comprised of AXI master, AXI slave, checkers and score board. With the help of the components mentioned above, the verification environment can simulate most cases of the AXI signal, check all the transmitted data automatically and complete coverage analysis during the simulation.

This Verification environment in OVM has many advantages w.r.t to a complex AHB Protocol

1) It is a standardized verification methodogly has no confusion to choose a methodology to start a paper.
2) No tool dependency.
3) No more porting issues from one tool to other.
4) Less confusion for the engineers.
5) Easy to learn and maintain the skill set.
6) SOC level driver controller for AHB with other AMBA based bridges.
7) Verifying AHB protocol in switch environment.
8) Verifying AHB Master Modes for Single and Multi-Master.
9) Scalable to UVM-Universal Verification Methodology.

This AXI Verification environment in SystemVerilog can be used for

1) Verifying AXI protocol in switch environment
2) Verifying AXI Master modes for Single and Multi-Master
3) SoC level driver controller for AXI with other AMBA based bridges
4) Scalable to UVM-Universal Verification Methodology

SCOPE OF ENHANCEMENT

The AMBA AXI has limitations with reference to the burst and beats information to be transferred. The burst data must not cross the 4k boundary. Bursts longer than sixteen beats are only supported for the INCR burst type. The WRAP and FIXED burst types remain constrained to a maximum burst length of 16 beats. These are the measures of AMBA AXI system which need to be overcome.

The Verification environment already covers test development and inbuilt end to end scoreboard and also provides scope to add the below as enhancements or for future purposes

1) Code Coverage

2) Functional Coverage

3) Assertions

The Verification environment provides scope to add the below as enhancements or for future purposes.

1) Q-Formal.
2) Q-Check.
3) INFACT.
4) Additional SVA Assertions.
5) Coverage Code and Functional.

REFERENCES

[1] Yashdeep Godhal, Krishnendu Chatterjee, Thomas A. Henzinger, “Synpaper of AMBA AHB from Formal Specification: A Case Study”, in International Journal on Software Tools for Technology Transfer, July 2011.

[2] Yangyang Li, Wuchen Wu, LigangHou, Hao Cheng, “A Study on the Assertion-Based Verification of Digital IC”, in Proc. of Second International Conference on Information and Computing Science, 2009, pp. 25-28.

[3] A. Nandi, B. Pal, N. Chhetan, P. Dasgupta and P. P. Chakrabarti, “H-DBUG: A High-

level Debugging Framework for Protocol Verification using Assertions", in Proc. of IEEE Indicon Conference, 2005, Chennai, India, pp. 115-118.

[4] Roychoudhury, T. Mitra, S.R. Karri, "Using formal techniques to Debug the AMBA System-on-Chip Bus Protocol", Proc. of IEEE Computer Society of Design, Automation and Test in Europe, 2003,Munich, Germany, pp. 828 – 833.

[5] P. Chauhan, E. Clarke, Y. Lu, and D. Wang, "Verifying IP core based system-on-chip designs", in Proc. of 12th Annual IEEE ASIC SOC Conference, 1999, India, pp. 27-31.

[6] M. Benjamin, D. Geist, A. Hartman, G. Mas, R. Smeets, and Y. Wolfsthal, "A Study in Coverage-Driven Test Generation," in Proc. of 36th Design Automation Conference (DAC'99), 1999, Los Angeles, USA, pp. 970-975.

[7] Pradeep S R , Laxmi "DESIGN AND VERIFICATION ENVIRONMENT FOR AMBA AXI PROTOCOL FOR SOC INTEGRATION" IJRET: International Journal of Research in Engineering and Technology eISSN: 2319-1163 ISSN: 2321-7308,338-343

[8] A Survey of Three System-on-Chip Buses: AMBA, CoreConnect and Wishbone MilicaMitic and Mile Stojcev

[9] ARM Ltd., AMBA specification (rev. 2) 1999

[10] Development of Verification Environment for AXI Bus Using System Verilog Xu Chen, ZhengXie, and Xin-An Wang ,International Journal of Electronics and Electrical Engineering Vol. 1, No. 2, June 2013,112-114

[11] Walter Hartong, Nils Luetke-Steinhorst, Hannes Froehlich Coverage, Driven Verification for Mixed Signal Systems, Cadence Design Systems,2007

[12] A Survey of Three System-on-Chip Buses: AMBA, CoreConnect and Wishbone MilicaMitić and Mile Stojčev

[13] An Overview of System Verilog 3.1 by Stuart Sutherland published in EEdesign, May 23, 2003

[14] Akshay Mann, Ashwani Kuma, Assertion Based Verification of AMBA-AHB Using Synopsys VCS, International Journal of Scientific & Engineering Research, Volume 4, Issue 11, November-2013 ISSN 2229-5518,58-64

[15] Shankar, Dipti Girdhar, Neeraj Kr. Shukla, Design and Verification of AMBA APB Protocol, International Journal of Computer Applications (0975 – 8887) Volume 95– No.21, June 2014,29-34

[16] T.Ananth kumar, DR.S.Saraswathi Janaki, DESIGN OF AXI BUS FOR 32-BIT PROCESSOR USING BLUESPEC, International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May 2012, ISSN: 2278 – 1323,184-188