International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 6, June - 2014

Verification of AHB Protocol using UVM

Tejaswini H N [1]
Asst. Prof. Dept. of ECE,
Sambhram Inst of Technology,
Bangalore, India

Revati Bothe[2]
Team leader,
SmartPlay Technologies
Bangalore, India

Ravishankar C V [3]
Prof. Dept. of ECE
Sambhram Inst of Technology,
Bangalore, India

***Abstract* - This paper describes the verification of AHB Protocol using the methodology UVM(Universal Verification Methodology). AHB is an Advanced High performance system Bus that supports multiple masters and multiple slaves. It implements burst transfers, split transactions, single-cycle bus master handover, single-clock edge operation, wider data bus configuration (64/128bits). Verification IP is the one which provides a smart way to verify the AHB Components such as Master, Slave, Arbiter and Decoder. UVM is used for the verification of AHB Protocol which provides the best framework to achieve CDV (Coverage Driven Verification) which combines automatic test generation, self-checking test benches, and coverage metrics to significantly reduce the time spent on verifying a design. An UVM test bench is composed of reusable verification environments called VCs (verification Componenets). This paper examines the verification of VCs which are structured to work with Verilog, VHDL, System Verilog and System C.**

Keywords : AHB, CDV, UVM, TLM, VC, Test Bench, Sequencer

I. INTRODUCTION

This section describes the survey that was done and the verification methodology is described and an outline of the rest of this dissertation provided.

AHB (Advanced High Performance Bus) is a new generation of AMBA (Advanced Microcontroller Bus Architecture) bus which is intended to address the requirements of high-performance synthesizable designs. AMBA AHB is a new level of bus which sits above the APB and implements the features required for high-performance, high clock frequency systems. UVM is a complete verification methodology that codifies the best practices for development of verification environments targeted at verifying large gate-count, IP-based SoCs (System on chip). Verification productivity stems from the ability to quickly develop individual verification components, encapsulate them into larger reusable verification components (VCs), and reuse them in different configurations and at different levels of abstraction. UVM uses a System Verilog implementation of standard Transaction Level Modeling (TLM) interfaces for modular communication between AHB components such as Master and Slave . The System Verilog UVM Class Library also provides various utilities to simplify the development and use of verification environments. These utilities support debugging by providing a user- controllable messaging utility. The System Verilog UVM Class Library provides global messaging facilities that can be used for failure reporting and general reporting purposes.

A. Objective of the study

1) To study the specifications of AMBA AHB which include all the scenarios .
2) To generate the Test Plan comprises of Test Cases which meets the specified scenarios .
3) To understand the development of Verification Environment for Single Master -Single Slave which follows the UVM topology of Driver, Sequencer and a Monitor along with Test Plan.
4) Analysis on how the driver drives the sequences from the sequencer to the Score Board.
5) In the scoreboard, the actual output is compared with the expected one. If the obtained output matches with the expected result means verification is completed successfully

II. ARCHITECTURE OF UVM TEST BENCH

A. UVM Test Bench and Environment

An UVM test bench is composed of reusable verification environments called Verification Components (VCs). The VCs are applied to the device under test (DUT) to verify the implementation of the AHB protocol. Architecture of UVM Test Bench is shown in Figure.1

B. Building Blocks of Test Bench

The three main building blocks of a test bench in UVM based verification are

- Uvm_env:
  It is the top level component of the verification components. uvm_env is extended from uvm_component. It is used to create and connect the uvm_components like drivers, monitors, sequencers. It can also used as sub environment in another environment.

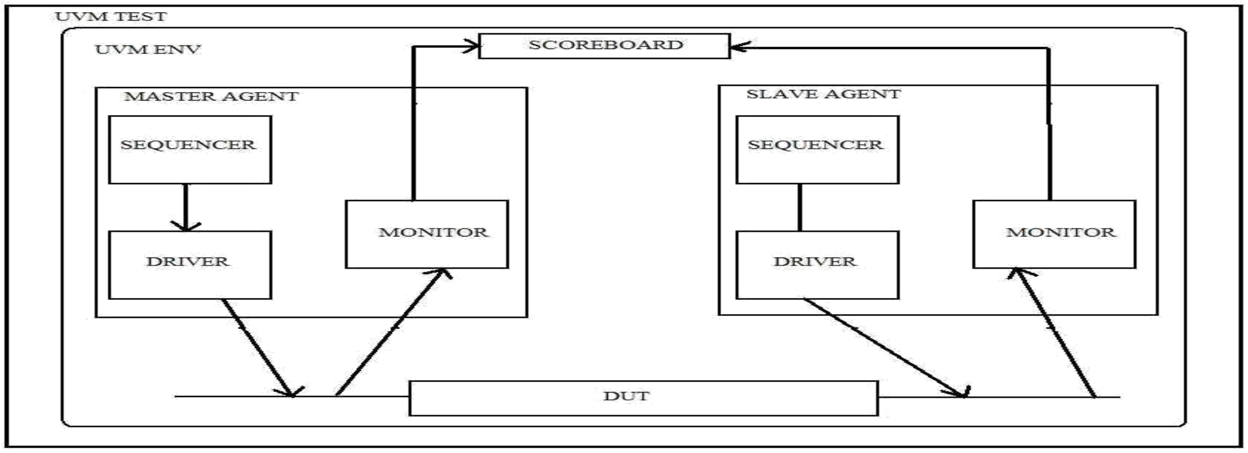


Figure.1. Typical UVM Test Bench Architecture

- UVM test:
  The uvm_test class defines the test scenario for the test bench specified in the test. The test class enables configuration of the test bench and verification components.

- UVM Verification Components:
  Sequencer (stimulus generator) : The sequencer generates stimulus data and passes it to a driver for execution. The uvm_sequencer is the base class of uvm class library contains all of the base functionality required to allow a sequence to communicate with a driver.

1. Driver : The driver drives data items to the bus following the interface protocol. The driver obtains the data items from the sequencer for execution. The UVM Class Library provides the uvm_driver base class.

2. Monitor: The monitor extract signal information from the bus and translates it into events, structs, and status information.

3. Agent : An agent has two basic operating modes

i. Active mode : In this mode, the agent emulates a device in the system and drives DUT signals. This mode requires that the agent instantiate a driver and sequencer. A monitor also is instantiated for checking and coverage.

ii. Passive mode : In this mode, the agent does not instantiate a driver or sequencer and operates passively. Only the monitor is instantiated and configured. This mode is used only when checking and coverage collection is desired.

III UVM CLASS HIERARCHY

The components in an UVM verification environment are derived either directly or indirectly from the uvm_component class as shown in Figure 2

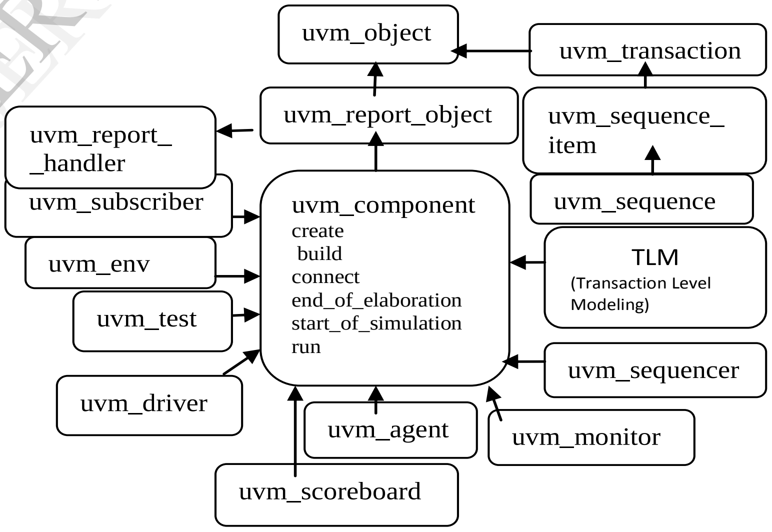


Figure.2 UVM Class Hierarchy

- Build() : This phase is used to construct various child components/ports/exports and configures them.
- Connect() : This phase is used for connecting the ports/exports of the components.
- End of elaboration() : This phase is used for con g-using the components if required.
- Start of simulation() : This phase is used to print the banners and topology.
- Run() : In this phase, main body of the test is executed where all threads are forked off.

IV. RESULTS AND DISCUSSION

The results of verification components such as Master Agent and the Slave Agent of the UVM Environment are presented. According to Test Plan, the test cases are verified by developing the Verification IP for the AHB Protocol. The Test Cases are written in the form of sequences in the Sequencer using System Verilog. The sequencer drives the sequences to the driver and thereby to Score Board.

In the scoreboard, the actual output is compared with the expected one. If the obtained output matches with the expected result then we conclude that the verification is completed successfully. By using the tool Questa, the Verification of AHB Components known as VC's such as Master Agent and Slave agent are done and the log files for the test cases are generated and simulated waveforms are achieved and shown in Figures 2 to 6

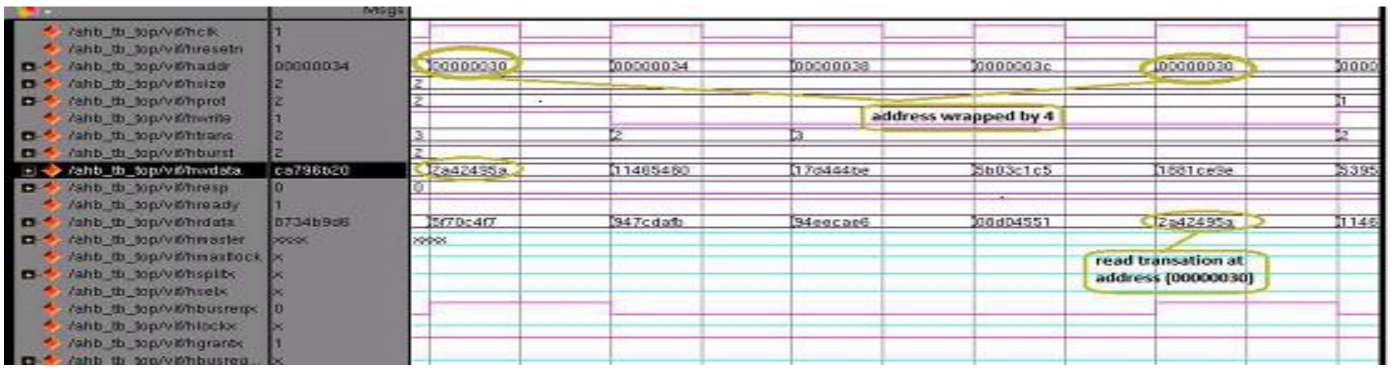
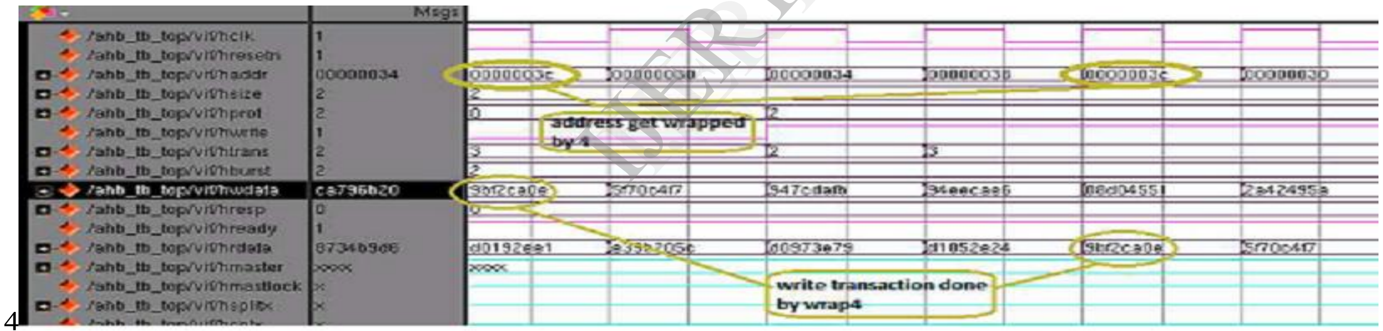


Figure 2: Read Transaction for wrapburst



4

Figure 3: Write Transaction for wrapburst 4

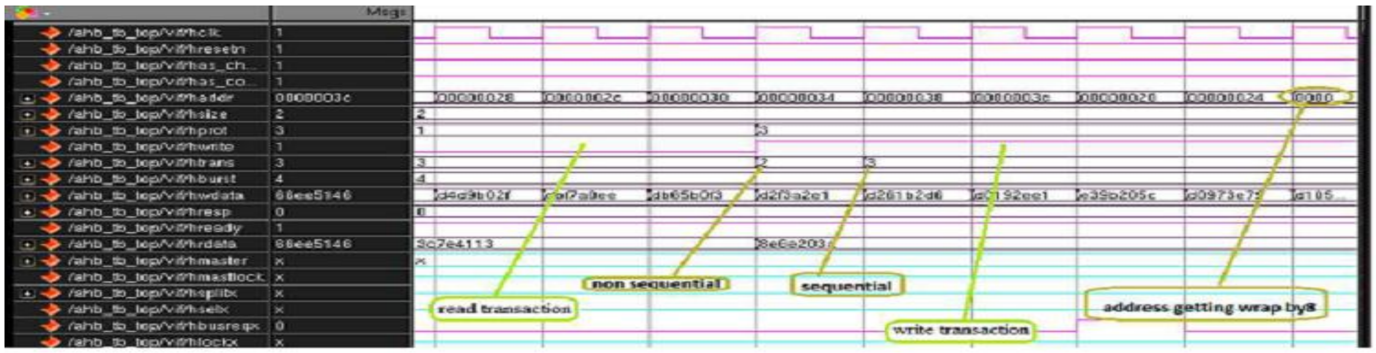


Figure 4: Read and Write Transaction for wrapburst 8

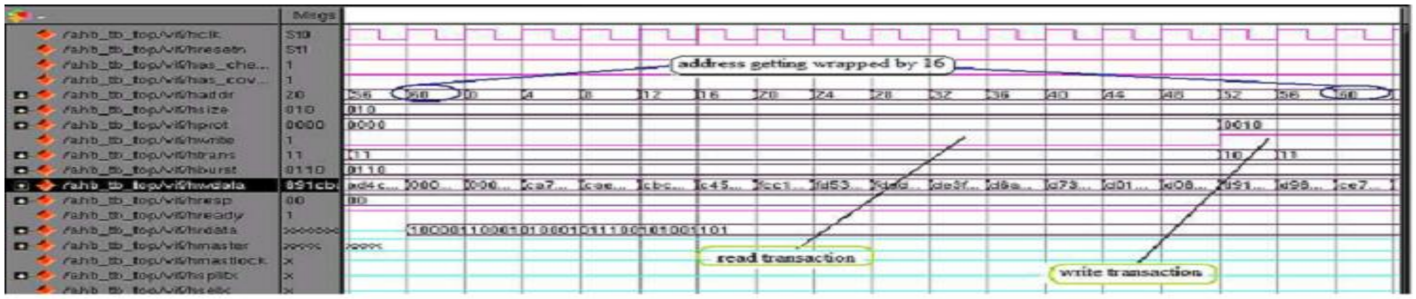


Figure 4: Read and Write Transaction for wrapburst 16

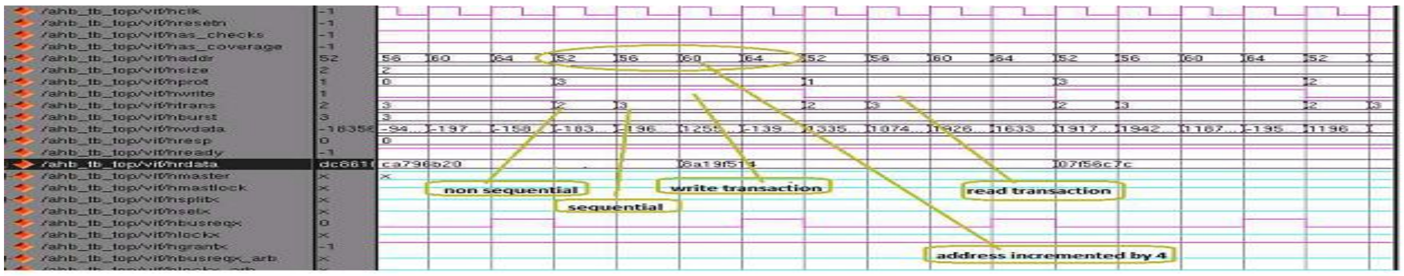


Figure 5: Read and Write Transaction for incrementing burst 4

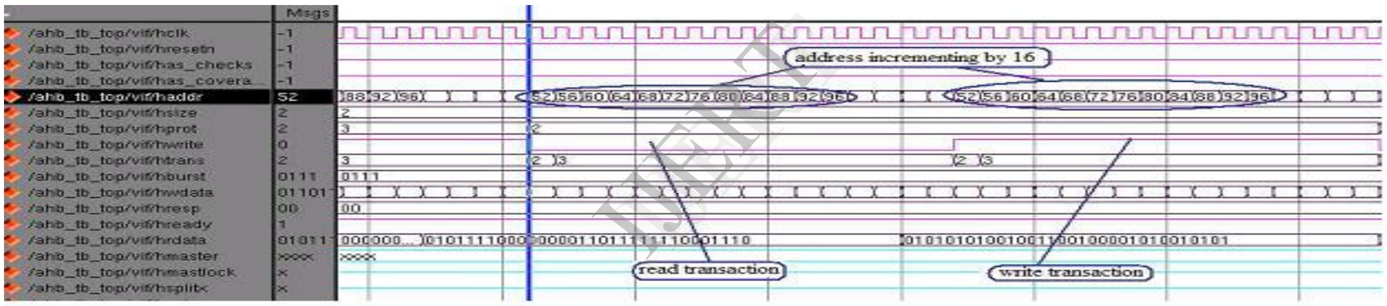


Figure 6: Read and Write Transaction for incrementing burst 16

V. CONCLUSION

Verification of AHB Protocol for Single Master-Single Slave has been verified by developing the Verification IP using the UVM methodology. UVM used System Verilog implementation of standard TLM interfaces for modular communication between Verification Components. Coding of AHB components such as Master Agent and Slave Agent has been done in System Verilog using UVM. The Verification IP s developed using tool Questa and it is sure that developed VIP is also compatible with other tools such as Cadence, Synopsys etc. The developed Verification IP can be used in the verification of SoCs.

VI. ACKNOWLEDGEMENT

We the authors of this paper would like to great fully acknowledge with thanks to Head of ECE department Prof. C V Ravishankar and Gaurav Jalan Senior Manager, SmartPlay Technologies (I) Pvt.Ltd.,Bangalore. for constant encouragement and support for enabling us to submit this paper.

VII. REFERENCES

[1] Jack Erickson , “TLM-Driven Design and Verification – Time For a Methodology Shift”, Cadence Design Systems, Inc.

[2] Rath A.W, Esen.V and Ecker.W , “A transaction –oriented UVM-based library for verification of analog behavior ” Publication Year: 2014 Page(s): 806 – 811

[3] Stuart Sutherland, Don Mills, “Synthesizing System Verilog Busting the myth that System Verilog is only for Verification”

[4] Soo-Yun Hwang and Kyoung-Sun Jhang , “An Improved Implementation Method Of AHB BusMatrix” SOC Conference 2005, Proceedings, IEEE International pp. 211-214

[5] Mulani, “ Level Verification Using SystemVerilog ” Emerging Trends in Engineering and Technology (ICETET), 2009 2nd International Conference on 16-18 Dec.2009 pp.378-380

[6] Pockrandt, M , Herber, P and Glesner, S, “ Model checking a System C/TLM design of the AMBA AHB Protocol ” Embedded Systems for Real-Time Multimedia (ESTIMedia), 2011 9th IEEE Symposium on 13-14 Oct.2011 pp.66 – 75

[7] IEEE Draft Standard for System Verilog - Unified Hardware Design, Specification and Verification Language, IEEE P1800/DS, February 2012 pp.1-1304

[8] Young-Nam Yun, Jae-Beom Kim, Nam-Do Kim and Byeong Min, “Beyond UVM for practical SoC verification” SoC Design Conference (ISOCC), 2011 International pp. 158-162

[9] Keaveney.Martin,McMahon.Antony, O’Keeffe.Niall, Keane.Kevin and O.Reilly James “The Development of advanced verification environments using System Verilog” Signals and Systems Conference,208.(ISSC 2008),IET Irish pp.325-330

[10] Universal Verification Methodology 1.1 User’s Guide May 18, 2011 http://www.accellera.org

AUTHOR'S PROFILE

1. Tejaswini H N
Currently working as Assistant Professor in ECE Dept of Sambhram Institute of Technology, Bangalore.
Specialization : VLSI Design and Testing.

2. Revati Bothe
Team Leader in SmartPlay Technologies (I) Pvt. Ltd , Bangalore

3. Prof.Ravishankar C V
Head of the Dept. Department of Electronics & Communication Engineering , Sambhram Institute of Technology, Bangalore.