VERIFICATION OF AMBA AHB2APB BRIDGE USING UNIVERSAL VERIFICATION METHODOLOGY (UVM)

ANKEM KIRAN[1], V THRIMURTHULU[2]
[1]PG Scholar, M.Tech(VLSISD), Dept. of ECE, CR Engineering College, Tirupati, AP, India
[2]Professor & HOD, Dept of ECE, CR Engineering College, Tirupati, Chittor, AP, India

Abstract–

With the progress in technology, tools and methodologies need to be improved to meet the challenges of transforming verification environment. The adoption of System Verilog[9] based Universal Verification Methodology[7] (UVM) bridges the gap between high-level proposition and low-level details of the design under verification. The intent of this paper is to throw light into benefits associated with AHB2APB verification using Universal Verification Methodology (UVM). An important feature for any of the SoC is based on how they interconnect. The Advanced Microcontroller Bus Architecture (AMBA) is on-chip bus architecture used to strengthen the reusability of IP core and widely used interconnection standard for system on chip (SOC).

Keyword: UVM, AHB, APB, AMBA, SOC

I. Introduction

This paper is all about using Universal Verification Methodology for verifying AMBA [3] based AHB2APB Bridge's functionality. In our work we intend to use Verilog HDL[8] (Hardware Description Language) based RTL (Register Transfer Level) design code and UVM[7] for verification. Synthesis is done using Xilinx & Verification is done using Rivera pro.

II. TYPICAL AMBA BASED MICROCONTROLLER

An AMBA-based microcontroller [Fig.1] typically consists of a high-performance system backbone bus[Fig.1] (AMBA AHB or AMBA ASB), able to sustain the external memory bandwidth on which the CPU, on-chip memory and other Client devices reside. This bus provides a high-bandwidth interface between the elements that are involved in the majority of transfers. Also located on the high performance bus is a bridge to the lower bandwidth APB client, where most of the client devices in the system are located. AMBA APB provides the basic peripheral macro cell communications infrastructure as a secondary bus from the higher bandwidth pipelined main system bus.

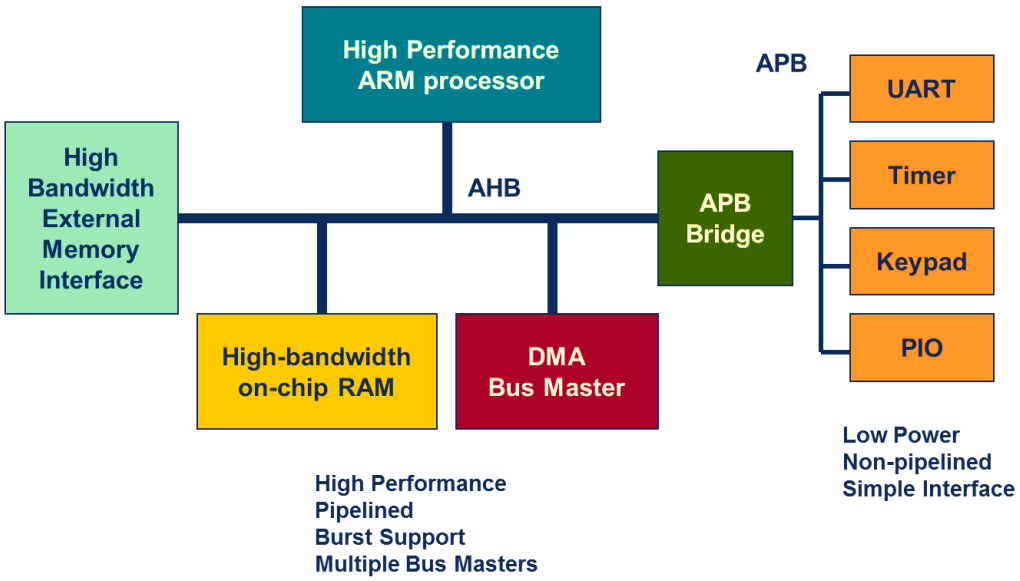

Fig.1.STANDARD AMBA BUS

II.A. Advanced high-performance bus (AHB)

AHB is a new generation of AMBA bus[2], which is intended to address the requirements of high-performance synthesizable designs. It is a high-performance system bus that supports multiple bus masters and provides

High - bandwidth operation. AMBA AHB implements the features required for high-performance, high clock frequency systems including:

- High performance
- Pipelined operation
- Multiple bus masters
- Burst transfers
- Single-cycle bus master handover
- Non-tri state implementation
- Wider data bus configurations (64/128bits).

Bridging [Fig.2] between this higher level of bus and the current AHB/APB can be done efficiently to ensure that any existing designs can be easily integrated

II.B. Advanced peripheral bus (APB)

The Advanced Peripheral Bus (APB) is part of the Advanced Microcontroller Bus Architecture (AMBA) hierarchy of buses and is optimized for speed and reduced interface complexity. The AMBA APB should be used to interface to any peripherals which are low-bandwidth and do not require the high performance of a pipelined bus interface. The latest revision of the APB ensures that all signal transitions are only related to the rising edge of the clock[Fig.7].

This improvement means the APB peripherals can be integrated easily into any design flow.
Features of APB[2]:

- Low power
- Latched address and control
- Simple interface
- Suitable for many peripherals

These changes to the APB also make it simpler to interface it to the new Advanced High-performance Bus (AHB) on cloud side.

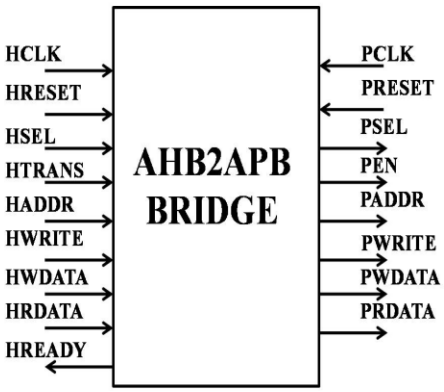

Fig.2 AHB2APB Pin Diagram

II.C. Features of AHB2APB Bridge

Interface between AMBA high performance bus (AHB) and AMBA peripheral bus (APB) , provides latching of address, controls and data signals for APB peripherals. Also synchronization is necessary in between client on server (AHB and APB) for optimal performance.

III. Pin details of AHB2APB Bridge

Systems are considered asynchronous to each other:

- When they operate at two different frequency
- When they operate at same frequency, but at twodifferent clock phase angles

This interfacing is difficult in the sense that design becomes asynchronous at the boundary of interface, which results in setup and hold time violation, Meta stability and unreliable data transfers. Hence we need to go out for special design and interfacing techniques. In such a case if we need to do data transfer, there are very few methods to achieve this namely:

- Handshake signalling method
- Asynchronous FIFO

Both have its own advantages and disadvantages. In our paper we have used Handshake signaling Method. In Handshake signalling method the AHB interface sends data to APB interface based on the handshake signals[Fig.8] PENDWR (or PENDRD) and PDONE signals. The protocol for this uses the same method that is found with 8155 chip used with 8085 based on handshake signals Request and Acknowledge

III.A. AHB Response

The sub-module AHB[4] Response sequences the way that the AHB2APB responds to AHB requests. Valid commands are forwarded to control transfer for action. Invalid commands are not forwarded and an error message is generated. It operates on AHB CLOCK and RESET.

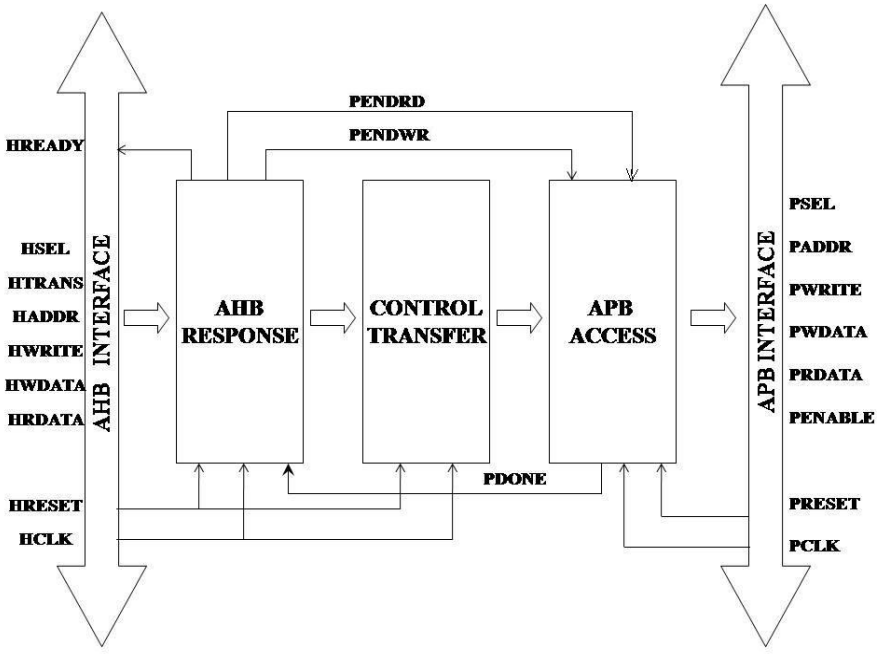

Fig.3. Internal architecture of bus

The control Transfers block transfers AHB control signal to the APB access with appropriate delays inserted to map the pipelined AHB protocol to the two cycle APB protocol. It ensures that only one request is presented to the APB access while it is processing a request. It operates on AHB CLOCK and RESET.

III.B. APB Access

The APB[1] access generates the control signals on the APB for read and writes cycles. It operates on APB CLOCK and RESET. The APB Bridge is the only bus master on the AMBA APB. In addition, the APB Bridge is also a slave on the higher-level system bus. The bridge unit converts system bus transfers into APB transfers and performs the following functions:

- Latches the address and holds it valid throughout the transfer.
- Decodes the address and generates a peripheral select (PSEL). Only one select signal can be active during a transfer.
- Drives the data onto the APB for a write transfer.
- Drives the APB data onto the system bus for a read transfer.
- Generates a timing strobe, PENABLE, for the transfer

III.C. Design of top module

This module is simplest and also very prominent. All the signals are taken as wire to interconnect various modules present under this top module.

In this module all the three modules namely:

- AHB Driver/Monitor
- AHB2APB bridge
- APB Driver/Monitor

These modules are all instantiated using Positional assignments which is again simple compared to naming assignment which is little tedious.

IV. PROPOSED VERIFICATION ENVIRONMENT

The AHB2APB Bridge has been verified. The design has the bidirectional channels as the Input/ Output. The design is based on the finite state machine approach. The verification methodology adopted covers the principles like constrained-Random stimulus[7], Functional Coverage, Common test bench for all tests. Test-specific code is kept separate from test bench. Automation Scripts support various switches so that we can re-use the test cases for many different scenarios. The Verification Environment[Fig.4] is as shown below

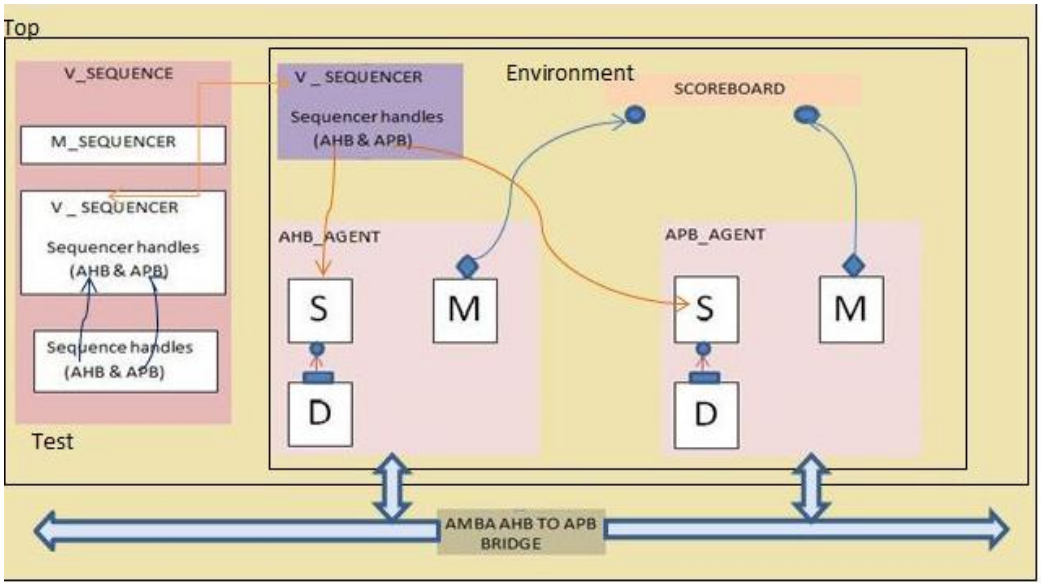

Fig.4..UVM Architecture of AMBA AHB2APB Bridge Verification

The basic components of the verification environment are described below:

1. **Test bench:** Test bench mimics the environment in which the design resides. It checks whether the RTL implementation meets the design specification or not. This environment creates invalid and unexpected as well as valid and expected conditions to test the design.

2. **Test Cases:**Test cases are written for different scenarios, which cover the functionality, corner cases. Basic read write transactions, various burst modes are verified. Parameterization will be randomized for different test cases and are written to check all the possible scenarios. These test cases are run in regression with multiple seeds.

3. **DUT:**The verification environment is organized in a hierarchical layered structure which helps to maintain and reuse it with the DUT.

4. **Virtual Interface:**Virtual interfaces[7] provide a mechanism for separating abstract models and test programs from the actual signals that make up the design. It allows the same subprogram to operate on different portions of design, and to dynamically control the set of signals associated with the subprogram. Instead of referring to the actual set of signals directly, users are able to manipulate a set of virtual signals.

5. Scoreboard: The scoreboard[7] collects write address and control information on address channel, and data on write data channel at the output of BRIDGE and compares it with the address, data and control information at the output of DUT.

V. RESULT & ANALYSIS

The Synthesized Netlist of Bridge Module is shown in fig.5

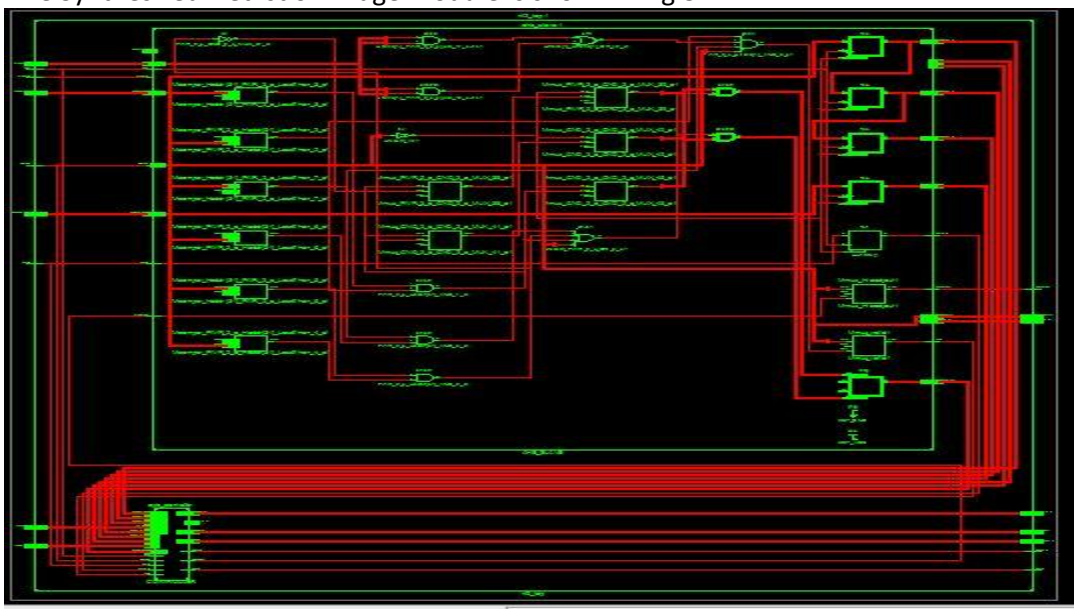

Fig.5. Synthesized Netlist of Bridge Module

V.A. SIMULATION RESULTS

The fallowing simulation results of read transfer results is shown fig.6, write transfer results is shown in fi.7, and write and read back to back results is shown in fig.8.

Read Transfer

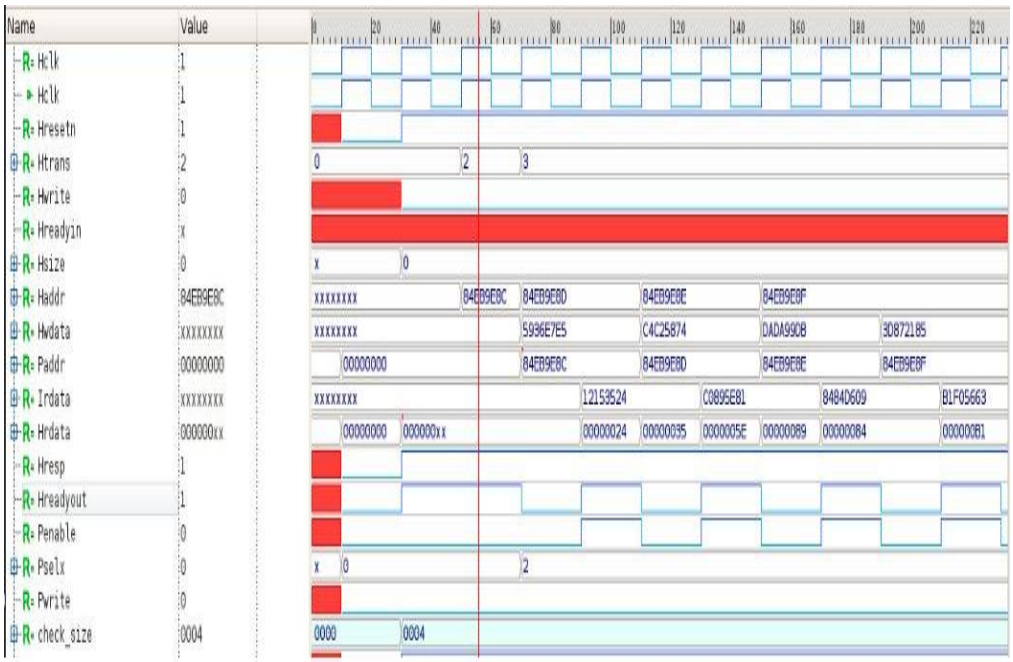

Fig6.Simulation of Read Transfer

Write transfer

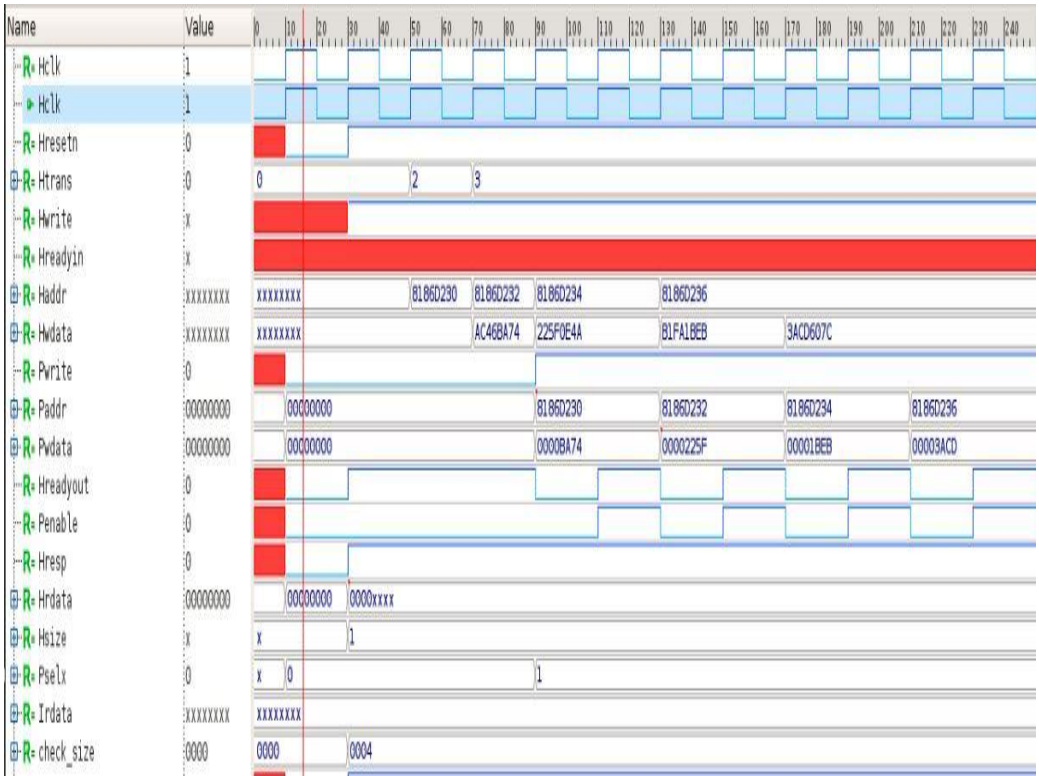

Fig.7. Simulation of Write Transfer

Write and read back to back transfer

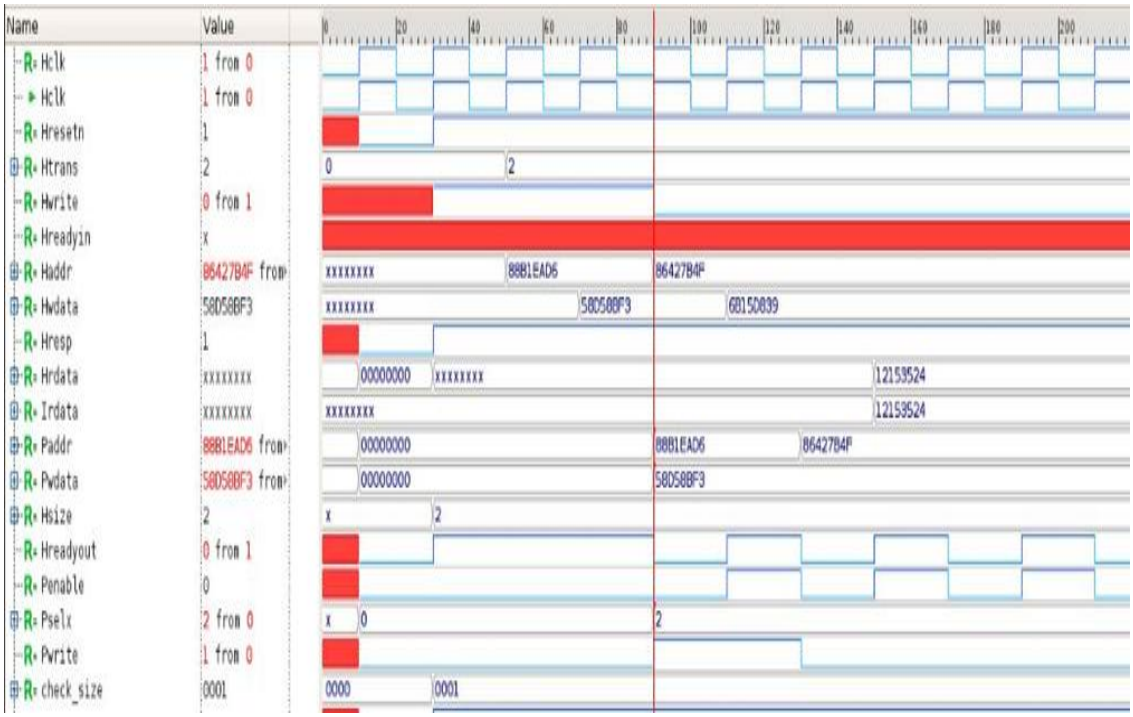

Fig.8. Simulation of Write and read back to back transfer with handshaking.

COVERAGE REPORT

Covergroup	Hits	Goal / At Least	Status
TYPE /\package pckg\/sc_b/master	100.000%	100.000%	Covered
INSTANCE <UNNAMED1>	100.000%	100.000%	Covered
COVERPOINT <UNNAMED1>::HTRANS	100.000%	100.000%	Covered
bin a	4	1	Covered
bin b	43	1	Covered
bin c	145	1	Covered
COVERPOINT <UNNAMED1>::HSIZE	100.000%	100.000%	Covered
bin a	57	1	Covered
bin b	62	1	Covered
bin c	73	1	Covered
COVERPOINT <UNNAMED1>::HADDR	100.000%	100.000%	Covered
COVERPOINT <UNNAMED1>::HWRITE	100.000%	100.000%	Covered
COVERPOINT <UNNAMED1>::HWDATA	100.000%	100.000%	Covered
bin a	28	1	Covered
bin b	12	1	Covered
bin c	19	1	Covered
bin d	15	1	Covered
bin e	10	1	Covered
TYPE /\package pckg\/sc_b/slave	100.000%	100.000%	Covered
INSTANCE <UNNAMED1>	100.000%	100.000%	Covered
COVERPOINT <UNNAMED1>::PRDATA	100.000%	100.000%	Covered
bin a	31	1	Covered
bin b	29	1	Covered
bin c	28	1	Covered
bin d	14	1	Covered
bin e	100	1	Covered

Fig.9. Functional coverage report

V.B. RESULT ANALYSIS

UVM architecture for AMBA AHB to APB Bridge is designed in Aldhec's Rivera pro-environment as shown in Fig.4 which is configurable as compared to traditional testbench, the coverage report is obtained with 100% coverage which conforms the certainty of each & every function of a design & also exhaustiveness of the test case's. The waveforms in the Fig.6,7,8 depicts the simulated read , write & back to back read write transfers in the UVM environment with different Hburst and Hsize's with respect to Hclk and penable. For read operations, it can be observed that Hwrite signal in Fig 6 determines the write/read operations. Hwrite low indicates the read operation and for write operation in Fig.7 Hwrite signal high determines the write operation.

VI. CONCLUSION

This paper gives an overview of UVM based AMBA ahb to apb bridge architecture and discusses the ahb to APB bridge in detail. The Bridge design is verified using Universal Verification Methodology. The simulation results [fig.6,7,8]show that the data read from a particular memory location is same as the data written to the given memory location. Hence, the design is functionally correct. The UVM report summary also ensures the functional correctness of the design.

Finally functional verification is done using Universal Verification Methodology (UVM) without any failure & 100% Functional Coverage [Fig.9] is achieved.

VII. Acknowledgement

I express my sincere thanks to my guide V.Thrimurthulu**. Professor & Head of ECE Dept. CREC, Tirupathi**, for his valuable guidance and useful suggestions, which helped me in the project work.

REFERNCES

[1]. Kiran Rawat., KanikaSahni, sujataPandey., RTL Implementation For AMBA APB ASB Protocol at System *On-Chip*, IEEE Trans. on Signal Processing and Integrated Networks (SPIN), June 2015, pp. 927 – 930.

[2]. AMBA Specification (Rev 2.0). http://www.arm.com

[3]. ARM, "AMBA Specification Overview", http://www. arm. com/. .

[4]. ARM, "AMBA APB3 Specification Overview", http://www. arm. com/

[5]. Akhilesh Kumar, RichaSinha, "Design and Verification analysis of APB3 Protocol with Coverage," IJAET, Nov 2011.

[6]. SanthiPriyaSarekokku, K. Rajasekhar, "Design and Implementation of APB Bridge based on AMBA AXI 4. 0," IJERT, Vol. 1, Issue 9, Nov 2012.

[7]. UVM Reference Manual, http://www. accellera. Com

[8]. Samir Palnitkar, "Verilog HDL: A guide to Digital Design and Synthesis (2nd Edition), Pearson, 2008.

[9].Chris Spear, "SystemVerilog for verification (2nd Edition): A guide to learning the testbench features, Springer, 2008.

[10]. URL:http://www. testbench. com.

[11]. Bergeron, "Writing testbenches using SystemVerilog," Springer, 2009.

[12]. Vanessa R. Cooper, "Getting Started with UVM: A Beginner's Guide," Verilab, 2013.

Authors

Ankem Kiran received his BE degree in Electronics & Communication Engineering from chaitanya bharathi institute of technology, gandipet, Hyderabad, telangana in the year 2013.currently pursuing his M.Tech degree in VLSI System Design at CR Engineering College, Tirupathi, India. His area of research includes VLSI design and Verification.

Dr. V. Thrimurthulu is currently working as Professor and Head of ECE Department at CR Engineering College, Tirupathi, India. He received his Graduation in Electronics & Communication Engineering from Institute of Electronics & Telecommunication Engineering, New Delhi and M.E (Microwaves and Radar Engineering) from University College of Engineering, Osmania University, Hyderabad. He has done his Ph.D in Philosophy from Central University. He has presented and published more than 40 papers in various National & International Journals.