Configurable Environment of AHB-QSPI

Ms Nikita Vadera[1], Mr Gardas Naresh Kumar[2], Mr. Prashant D.Karandikar[3]

[1] Research Scholar, GTU PG School,Gujarat,India
[2] Course Co-Ordinator, CDAC ACTS, Maharashtra, India
[3] Aurora Copper System LLP,Pune

ABSTRACT

SoC designers mostly use AMBA bus protocol which is developed by ARM which supports the efficient link between processors, off-chip external memory interfaces and on-chip memories. AMBA came up with many standards like AHB, APB and ASB. Working with different methodology till UVM comes into picture, which has a rich class library. This paper presents UVM based verification environment between the AHB protocols to QSPI protocol. Comparative study of AHB-QSPI bridge and add read-write testcase.

Keywords: - *AHB, QSPI, verification methodology, UVM.*

1. Introduction

This paper explains generic test bench architecture based on UVM. As per this methodology significance of different verification components and how to connect them .Section II presents detail information of UVM environment structure and transaction level modeling. Section III presents how to communicate two protocols in UVM environment. Section IV presents simulation result of test case and conclusion is given in section V.

2. Universal Verification Methodology

2.1 Base class Library

The basic building block is provided which are used to create verification environment by the base class of UVM.

2.2 UVM Test bench architecture

UVM based test bench architecture (fig) consist of different verification component.

1. **Test:** The UVN test bench mainly performs two tasks: Instantiates UVM environment, Driver the signal through the environment to the DUT.
2. **Environment:** The environment is the top-level component of the verification component. It contains all component and one or more agents, the environment contains configuration properties that make it reusable.

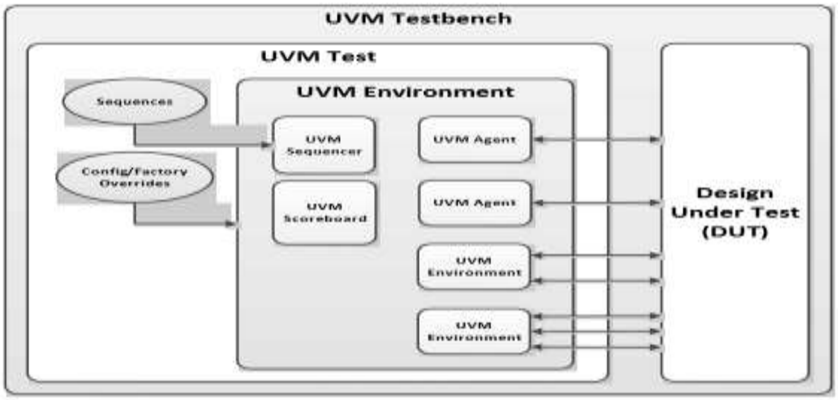

Fig 1: UVM test bench architecture [12]

3. **Agent**: Sequencers, drivers, and monitors are part of the agent. To reduce the amount of work and knowledge required by the test writer, UVM recommends that environment developers create a more abstract container called an agent.
4. **Sequencer**: A sequencer is bunch of signals that controls the items that are applied to the driver. Sequencer has an ability to react to the current state of the DUT and provide synchronization and control of multiple interfaces.
5. **Monitor:** A monitor which samples DUT signals and display the message .Monitor get the information through transaction and pass the signals to score board. Also Monitors collect coverage information and perform checking.
6. **Driver**: Driver as the name suggest, drive signals. It receives the signals from the sequencer and converts it in to the pin level activity.
7. **Scoreboard:** The UVM Scoreboard’s important function is to check the behavior of a certain DUT. The UVM Scoreboard usually receives transactions through UVM Agent and compare with reference or golden model which produce expected output.
8. Protocols which is set of regulation and rules to perform any task effectively. AMBA AHB and QSPI protocol are creating interface in UVM environment.

3. AHB-QSPI Interface

3.1 AMBA-AHB PROTOCOL

AHB bus which is planned to address the requirement of high performance Synthesizable designs. Also AHB High performance pipelined operation, multiple masters and burst transfers.

3.2 QSPI PROTOCOL

Quad Serial Peripheral Interface (SPI) module that either controls a serial data link as a master, or reacts to a serial data link as a slave.

3.3 AHB-QSPI BRIDGE

The AHB2QSPI interfaces AHB and QSPI. It buffers address, controls and data from the AHB, drives to the QSPI. The bridge performs transfer of signals from AHB to QSPI for write cycle and QSPI to AHB for Read cycle.

3.4 UVM based AHB-QSPI Environment

To achieve this, we will need a sequencer that generates sequences of signals to be transmitted in design under test. Sequences of signal pass to the driver which takes care of the communication with the DUT.

Then monitor sample the input and output of the DUT. They try to make a prediction of expected result and send the prediction and result of the DUT to scoreboard which compare the actual output with the DUT's output and evaluated.

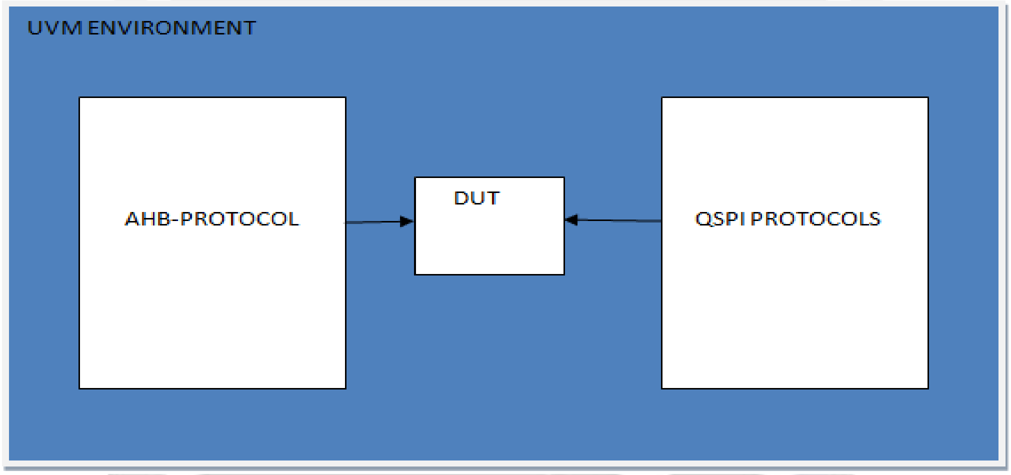

Fig 3: AHB-QSPI Environment

4 Simulation

Here one test case is presented which is performed by modelsim simulator. One of them is named as register read write test.

A. Register read write test

For register read-write test create reg model; which is itself is a set of DUT-specific files that extend the UVM base classes. In this test we are Appling different value which is specified in specification and then doing test and get the specified value.

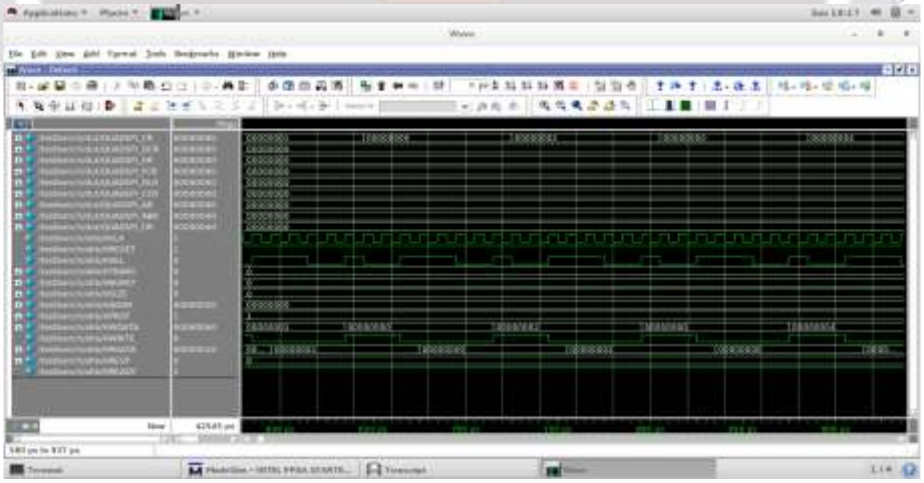

Fig 4 : Register Read-Write Test-1

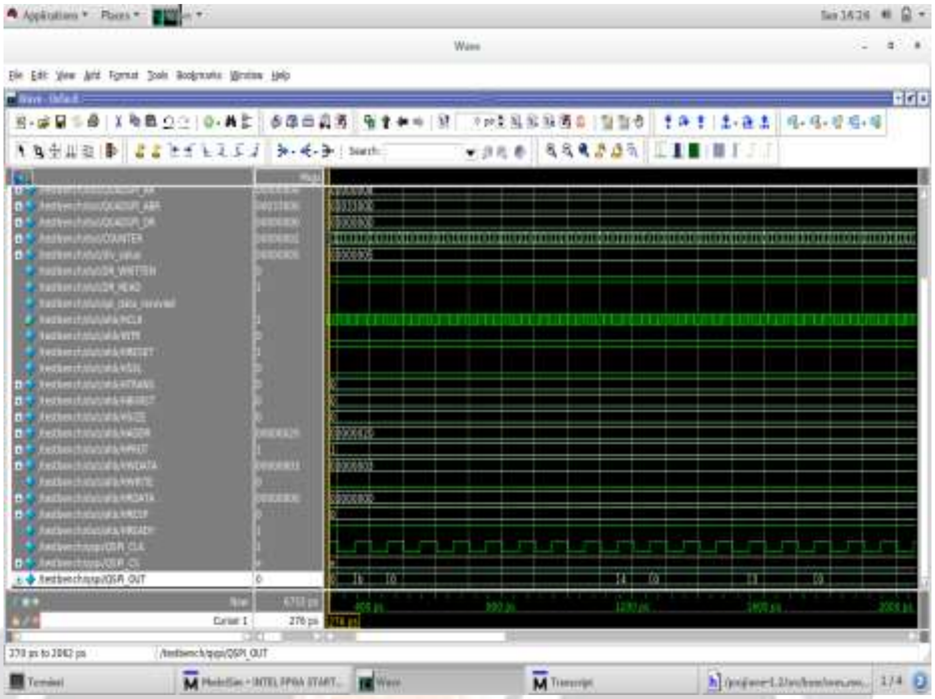

Fig 5 : Register Read-Write Test-2

5 Conclusion

we can conclude that environment for AHB-QSPI interface is developed with Universal verification methodology and register read write test is created to verify the bugs such the design register reset value.

REFERENCES

[1] Zhili Zhou, Zheng Xie, Xin'an Wang* and Teng Wang "Development of Verification Environment for SPI Master Interface Using SystemVerilog"2012 IEEE

[2] A.K. Oudjida, M.L. Berrandjia, A. Liacha, R. Tiar, K. Tahraoui "Design and test of general purpose SPI master/slave IPs on OPB" 2010 7th international multi-conference on system, signal and devices

[3]Jonathan Bromley, Verilab Ltd, Edinburgh, Scotland "If System Verilog Is So Good, Why Do We Need the UVM? Sharing Responsibilities between Libraries and the Core Language"- Specification & Design Languages (FDL), 2013 Forum – Sept 2013 IEEE.

[4] Khaled Salah "A UVM-Based Smart Functional Verification Platform: Concepts, Pros, Cons, and Opportunities"- 2014 9th International Design and Test Symposium

[5] Yingke Gao1,2*, Diancheng Wu1, Quanquan Li1,2, Tiejun Zhang1, Chaohuan Hou1 "Design and Implementation of Transaction Level Processor based on UVM"- 2013 IEEE

[6] Naveen Sudhish, Raghavendra BR, Harish Yagain "An efficient method for using transaction level assertions in a class based verification environment" 2011 International Symposium on Electronic System Design

[7] Mark Litterick, Marcus Harnisch Verilab GmbH "Advanced UVM Register Modeling" 2014 Verilab & DVCon

[8] Gayathri M, Rini Sebastian, Silpa Rose Mary, Anoop Thomas "A SV-UVM framework for Verification of SGMII IP Core with reusable AXI to WB Bridge UVC" ICACCS -2016

[9] AMBA specification [Rev 2.0]

[10] QUAD SPI specification

[11] uvm_users_guide_1.2

[12]Class Reference June 2011.pdf"

[13]"SYSTEMVERILOG FOR VERIFICATION A Guide to Learning the Testbench Language Features, by Chris Spear"

[14] SystemVerilog for Verification by Chris Spear (book)