INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 3)
Available online at: www.ijariit.com

AHB design and verification AMBA 2.0 using System Verilog

Divya M
mdivya587@gmail.com
P. E. S. College of Engineering, Mandya, Karnataka

Dr. K. A. Radhakrishna Rao
hodec@pesce.ac.in
P. E. S. College of Engineering, Mandya, Karnataka

ABSTRACT

Bus protocol is important in the field of Intellectual Property (IP) reuse by standardizing the interface of the hardware component as they help in simplifying the task. The Bridge between Advance High Performance (AHB) master and AHB slave also play an important role in communication. Advance Microcontroller bus architecture (AMBA) is the solution for the block to interconnect with each other AMBA protocol family is used to replace complex bridges with the specific protocol block interface in SoC design. The new verification constructs can be easily reused for the objected-oriented feature of System Verilog. The paper also introduced how to design the AMBA (advanced microprocessors bus architecture) verification IP (intellectual property) by System Verilog, which include AHB (advanced high-performance bus) master and AHB monitor. The verification IP can be reused to verify any AMBA protocol based SoC. To reduce the time spending in the verification, a reference model designing method is also discussed in the paper. The approach is to design the core that contains the processing hardware and the minimal interface is adapted by specific protocol by using a wrapper. It is suitable for low latency and high bandwidth design and provides the flexibility in implementation of interconnect architecture, also provides high frequency of operation without using complex bridge Advanced Microcontroller Bus Architecture (AMBA) is the on-chip communication standard generally used for designing high performance embedded controller and the IP for verification can be reused to verify other AMBA protocol based System on chip (SoC). The tools used for this project is Modelsim.

Keywords: *System Verilog, AHBA (Advanced Microcontroller Bus Architecture), AHB (Advance high-performance bus), Verification environment.*

1. INTRODUCTION

In the recent years, due to the increasing market requirement for low power, low area, low cost and high-performance systems as well as improvement of the semiconductor process technology, Very Large Scale Integration (VLSI) demand extents to very high where an entire system was required to integrate on a single chip. A chip has different functionality intellectual properties, and probably all these IPs had finished their steps i.e., verification and design independently. The main and common problem is transaction error due to incompatibilities between two IP interfaces. Thus, for implementing System on Chip (SOC), standardized bus protocol based mostly on information science design interface became the de-facto integrated methodology.

Therefore on-chip communication employing a bus protocol, whose specification offer a standard interface that facilitates IPs integration became the premise of SoC design. To make sure reuse of IP and also to make the SoC integration fast, some of the communication design standards which are based on the bus have been proposed.

On a shared SoC bus multiple IPs might ask the bus at the similar time and hence may lead to contention. A component on a SOC bus which is shared and determines the priority of masters to get bus access among multiple master requests on same time is called an arbiter.

The arbitration scheme which the arbiter follows dictates the arbiter to use the criteria used to decide about the master. The designer decides the arbitration scheme and is application specific. The SOC performance mostly depends on inter-component communication and not on the pure processor speed. Arbiter is an important on-chip communication architecture element as every transfer has arbiter involved in it. Hence great care must be taken as far as the design of arbiter module is concerned.

1.1.1 Verification using System Verilog

In system Verilog, we can have a race-free condition by using program block as for the test bench which is executed at reactive region and also we can use clocking block in system Verilog but both program block and clocking block we can't use in Verilog.

In Verilog, we have a static task that means that values are get overridden at each time but in system Verilog, we have a default automatic task and function so it will override automatically. We have a limited randomization in Verilog but in system Verilog, we have huge scope for randomization because we have constraint and coverage driven randomization. We have an assertion and coverage in system Verilog, which improves debug time and verification and also which improves observability and controllability but in Verilog assertion and coverage is not possible.

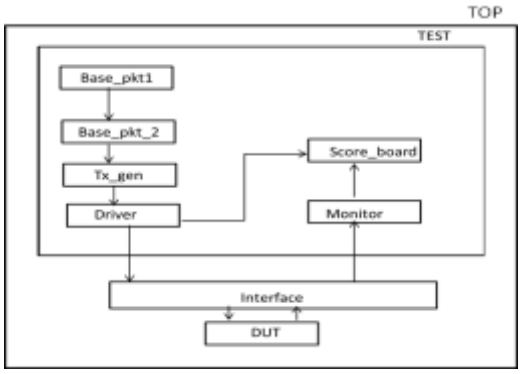


Fig. 1 AHB Environment Architecture

TOP

In the top module, we are generating clock and we are passing this clock to the interface as an argument and the connection for DUT, interface and test bench

TEST BENCH

Here we are creating the environment for the entire class handle and we are running the entire task inside the fork-join loop.

BASE PACKET

Here we are using two base packets, one for all input ports and another one for all output ports.

TRANS GENERATOR

In Trans generator, we are generating the stimulus and we are driving into the driver by calling the driver task instead of this we can also communicate driver and generator using the mailbox.

DRIVER

We all passing all the stimulus to the DUT through the virtual interface (virtual interface means it provides the same sub program to operate in different design and dynamically we can 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).

MONITOR

Here we are tacking all the output from DUT through the virtual interface and we are monitoring it in the monitor block.

SCORE BOARD

Here In score board, we are displaying all the outputs.

INTERFACE

The interface connects DUT and test bench and it gives the modport direction for the signals which are driven into the DUT and test bench.

DUT (design under test)

Here in DUT, we are instantiating our actual design.

2. PREVIOUS WORK

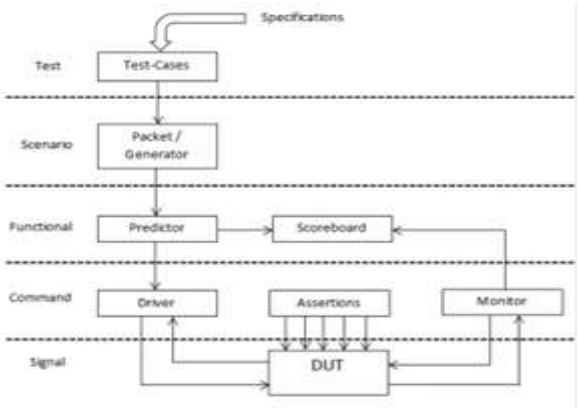


Fig.2 Shows a generalized verification platform

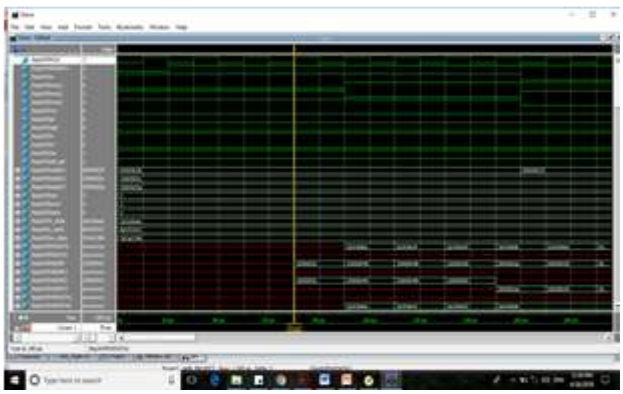
which includes test, packet/generator, predictor (agent), driver, monitor, Scoreboard, and assertions. Here the test is a program block. It is written after the top environment has been defined. The object of the top environment is called in the test file. It includes different kind of test-conditions to be applied for verification. The packet class is having all kinds of signals to be randomized. The generator generates random packets using randomize () function of SystemVerilog and send them to predictor using the mailbox. The predictor component receives these packets and does user requirement operations and transmits packets to driver as well scoreboard. The driver drives packets through a virtual interface. On the other side, monitor component receives packets from DUT and transmits these packets using the mailbox to the scoreboard. The scoreboard is nothing but storing the packets from driver and monitor. It has another part called Checker. It compares results from driver and monitor and checks correctness.

1.2 Simulation/Experimental Results

1.2.1 AHB Main Module

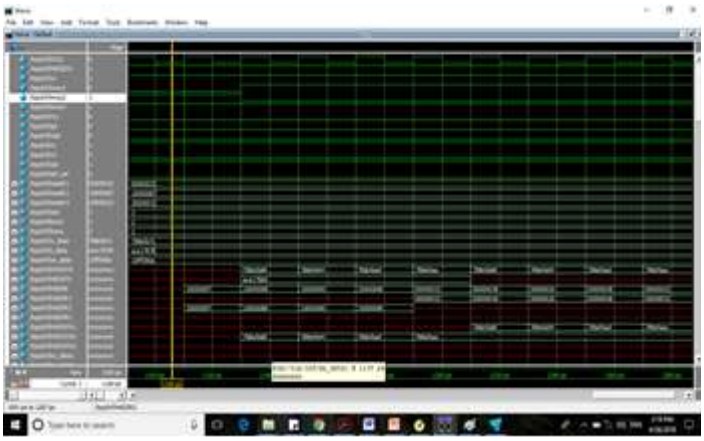
The result consists of simulated waveforms that are obtained after writing the code for the final module. Waveforms are used to verify the functionality of the master, slave, and their interconnection. It is found that the module of AHB give the desire functionality.

1.2.2 Priority Based Round Robin Algorithm



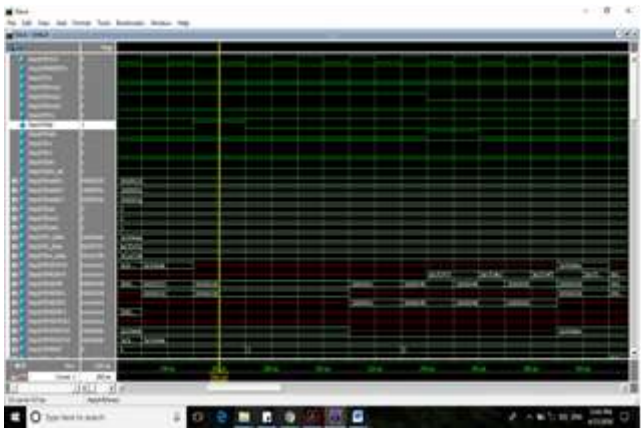
In the figure first mreq1,2 and 3 is 0,1,1 so that arbiter grant the bus to 2nd master, after some time we make mreq1,2 and 3 as 1,0,1 so according to priority based round robin algorithm it is given to the 3rd master.

1.2.3 Multimaster Higher Priority Based Algorithm



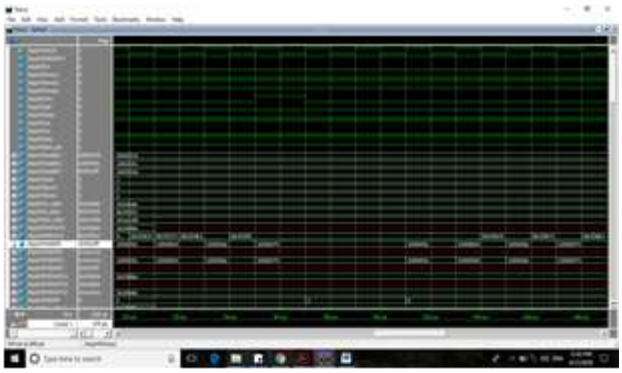
In the figure, first mreq1,2 and 3 is 0,1,1 so that arbiter grant the bus to 2nd master, after some time we make mreq1,2 and 3 as 1,0,1 so according to higher priority based algorithms it is given to the 1st master.

1.2.4 Split and Nonsplit Condition



Here mreq 1, 2 and 3 is 1,1,0 so arbiter grants the bus to the first master, but in between split is active hence according to round robin it is given to the second master again, here non split is active but second master completes the burst operation first then only it gives back the bus to the split master.

1.2.5 Retry Condition



Once retry signal is high it will stop the transaction after one clock cycle when retry goes low it will start the transaction from the initial address.

2. CONCLUSION

In this project, the Advanced Microcontroller Bus Architecture (AMBA) 2.0, which is an open System on Chip bus protocol for high-performance buses on low-power devices designed by ARM in 1999 is studied and designed the single master single slave[1], single master multi-slave and multi-master single slave configuration using Verilog HDL language in Modelsim10.5b. Also, design AHB Arbiter which monitors the AMBA [3] Bus for the request and chooses the master as the next AMBA bus transaction master by using highest priority Round Robin and Highest Priority Arbitration algorithms.

The single and burst transfer read operation is accomplished with zero wait states from the external ROM and the single and burst transfers write operation with zero states to the external RAM. By proposing the parallel communication in AMBA-AHB, the data transfer operation will be fast as compared to serial communication. It also provides the opportunity to use masters and slaves up to 16 numbers and the data of every master is read and written simultaneously.

Design and verification of single master multi-slave and multi master multi-slave AHB configuration and other two arbitration algorithms such as Random Access and Fair chance algorithms using Verilog Hardware Description Language and also the implementation of whole AHB device has been completed.

3. FUTURE SCOPES

Multimaster-Multislave AMBA2.0 AHB protocol should be verified using UVM (Universal verification method).

4. REFERENCES

[1] AMBA specification, version 2.0.
[2] Kumar, Akhilesh and Sinha, Richa, “design and verification analysis of ABP3 protocol with coverage “International journal of advance in engineering and Technology vol. 1 issue 5 pp.310-317, Nov 2011.
[3] Mitic, Milica and Stojcev, Mile, an Overview of On-Chip Buses SER.: ELEC. ENERG. vol. 19, no. 3, December 2006, 405-428.
[4] R. M., Vaniand M, Roopa “Design of AHB2APB Bridge for different phase and Frequency” International Journal of Computer and Electrical Engineering, Vol. 3, No.2, pp.April 2011.

