International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 3 Issue 5, May - 2014

Systemverilog Based AMBA AHB Verification Environment

Saurin Shah
VLSI & Embedded System Design
GTU PG School
Ahmedabad, India

Nirav Patel
VLSI & Embedded System Design
GTU PG School
Ahmedabad, India

***Abstract*--A verification environment of AMBA (Advanced Microprocessor Bus Architecture) AHB (Advanced High performance Bus)protocol using SystemVerilog language is presented in this paper.A verification based OOP technique increases the level of functional verification. This paper also shows Verification IP of AHB protocol including AHB Master, AHB Slave and AHB Arbiter. The Verification IP can be reused and easily managed to verify any AHB based design. Here, byapplying different test-cases on AHB protocol, shows the speed of design.**

Keyword: SystemVerilog, AMBA AHB, Verification environment, Verification intellectual property (VIP). A

I INTRODUCTION

Huge complexity of chip increases in recent years and integration of more number of transistors in a single SoC makes verification of any SoC based design very critical. About 75-80 percentage of the design cycle is spent in functional verification[5]. The use of OOP in verification process makes it bit easy. Due to high range of flexibility, the code of verification becomes very easy and simpler [6].

Currently there are many languages that are used as a verification language like, System C, SystemVerilog, Vera, etc. Among these SystemVerilog has a very high level data structure available in object-oriented language, like C++. These data structure has a very high level of modularity and reusability [1]. SystemVerilog can be used as a HDL (Hardware description language) like Verilog. It also provides some constructs similar to Verilog language, so it is very easy to understandable and easily usable.

In this paper, the problems regarding reusability and complexity were applied in the fashion of building verification environment. The different kinds of Test-cases were applied to the environment to verify functionality [7].

Here SystemVerilog is a Hardware Description and Verification Language based on Verilog. SystemVerilog provides one modelling language. You do not have to learn C++ and the Standard Template Library to create your models.

II AMBA AHB PROTOCOL

AMBA AHB addresses very high performance and high synthesizable designs. It supports multi-master and multi-slave. It has an Arbiter having different kinds of Arbitration strategy.

It has,

- burst transfers
- split transactions
- single cycle bus master handover
- single clock edge operation
- non-tristate implementation
- Wider data bus configurations (64/128 bits).

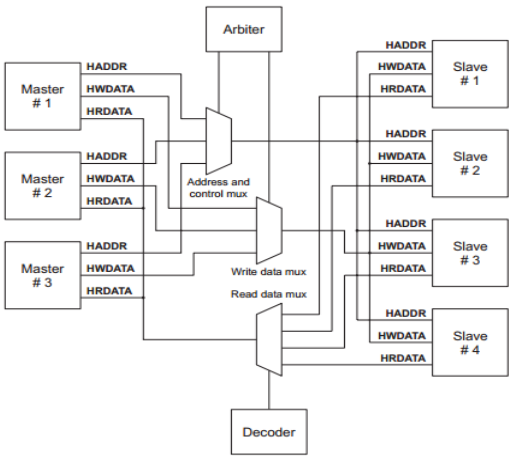

Fig. 1 AHB BUS configuration

Fig. (1) Shows AHB bus configuration, which includes AHB masters, slaves and Arbiter.

The AMBA AHB bus protocol is designed to be used with a central multiplexor interconnection scheme. Using this scheme all bus masters drive out the address and control signals indicating the transfer they wish to perform and the

arbiter determines which master has its address and control signals routed to all of the slaves. A central decoder is also required to control the read data and response signal multiplexor, which selects the appropriate signals from the slave that is involved in the transfer [11].

III GENERALIZED VERIFICATION PLATFORM

Fig. (2) Shows a generalized verification platform, which includes test, packet/generator, predictor (agent), driver, monitor, Scoreboard and assertions.

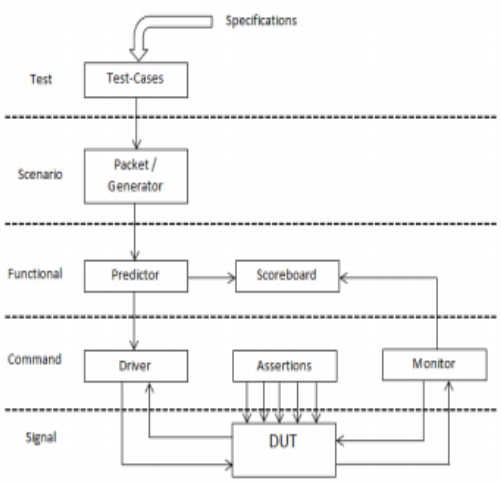

Fig. 2 generalized verification platform

Here test is a program block. It is written after top environment has been defined. The object of top environment is called in 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 generatorgenerates random packets using randomize () function of SystemVerilog and send them to predictor using 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 virtual interface. On the other side monitor component receives packets from DUT and transmits these packets using mailbox to 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.

IV VERIFICATION ENVIRONMENT OF AHB PROTOCOL

Fig. (3) Shows a complete verification environment of AHB protocol, which includes AHB components like, Master, Arbiter and Slave.

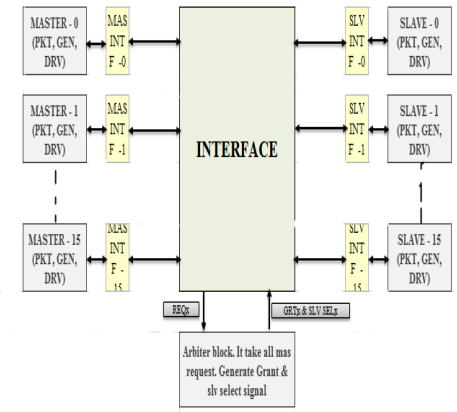

Fig. 3 Verification Environment of AHB

- Operation of Environment:

All 16 masters have packet, Generator, and Driver class. The Driver class transfer the all signal to local interface. Each master has its own local interface. They connected with main interface. Master receivesa Grant signal and some slave response signal in local interface from main interface. Only granted master transfers its signals to the main interface. Other masters wait for grant.

Similar all the 16 slaves have packet, Generator, and Driver classes. Each slave has its own local interface. Slave receives a select signal and master signal in its own local interface from main interface. It drives some response signal. Selected slave transfer response signal to main interface via local interface.

Arbiter takes all master requests and granted master addresses from main interface. According to arbiter algorithm a master is granted and according MSB bit of granted master address slave selection signal generate. These Grant and select signal transfer in main interface.

V TEST-CASES

1 Random test cases

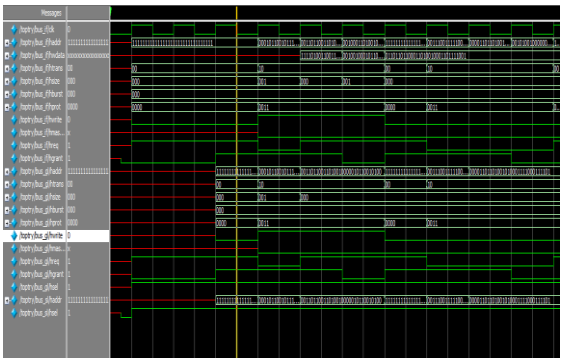

Fig. 4 random test-cases

2 Write operation

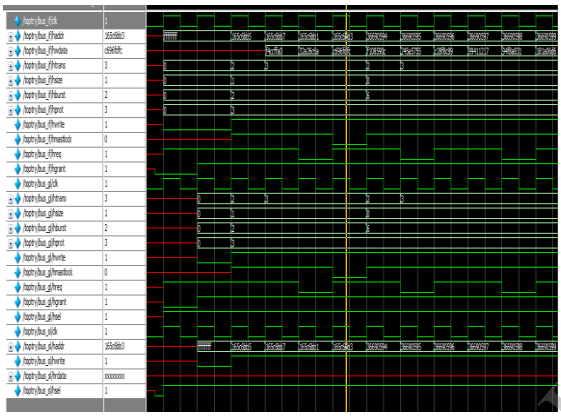

Fig. 5 write operation

3 Single burst transfer

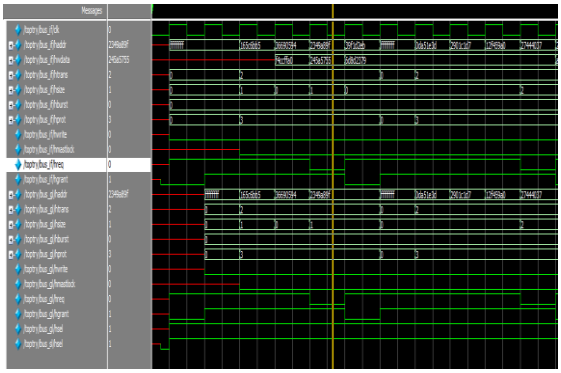

Fig. 6 Single burst transfer

I work on 3 different test-cases in this chapter. Randomization mode use in my 1st test cases. Fig (4) shows read and write both operation. Constrain random mode use in my 2nd test cases. Fig. (5) Shows write operation. When burst operation come at this time master lock signal high. It is saw in both Fig (4) and Fig (5). User mode use in 3rd test cases. Fig. (6) Shows single burst operation in read and write mode. Master lock signal is low(0) in this Fig. (6).

VI CONCLUSION

In this paper we use SystemVerilog language as a verification language. A planned verification environment has been done. Using test-Cases we can check the functionality of AHB based design. This environment is compatible with the all AHBA AHB specifications.

REFERENCES

[1] "Verification of AMBA Bus Model Using SystemVerilog" Han Ke, Deng Zhongliang, Shu Qiong2, The Eighth International Conference on Electronic Measurement and Instruments ICEMI'2007.
[2] "VERIFICATION ANALYSIS OF AHB-LITE PROTOCOL WITH"COVERAGE, Richa Sinha, Akhilesh Kumar and ArchanaKumari Sinha, International Journal of Advances in Engineering & Technology, Jan 2012.
[3] "Simple Hardware Verification Platform using SystemVerilog" Young-Jin Oh, Gi-Yong Song, TENCON 2011 - 2011 IEEE Region 10 Conference .
[4] "SoC Level Verification using System Verilog", Purvi D. Mulani, Second International Conference on Emerging Trends in Engineering and Technolog, ICETET-09.
[5] "A Functional Verification Environment", Sagahyroon, G. Lakkaraju, M. Karunaratne, circuit and system, 2005 48th Midwest Symposium.
[6] "Simple Hardware Verification Platform using SystemVerilog", Young-Jin Oh, Gi-Yong Song, TECON 2011 - 2011 IEEE region 10 conference
[7] "Promising Complex ASIC Design Verification Methodology", Mansour H Assaf, Sunil R Das, WaelHermas, Wen -B Jone, Instrumentation and Measurement Technology Conference Proceedings, 2007. IMTC 2007, IEEE.
[8] "Verification of I2C DUT using SystemVerilog", PurviMulani, JigneshPatoliya, Hitesh Patel, Dharmendra Chauhan, International Journal of Advanced Engineering Technology.
[9] Chris Spear, SystemVerilog for Verification A guide to Learning the Testbench Language Features, 2nd edition.
[10] ASICs to SOCs A Practical approach: by FarzadNekoogar and FaranakNekoogar
[11] "AMBA AHB"- Specification by ARM Limited.
[12] Questa SIM User's Manual, by Mentor Graphics.
[13] www.testbench.in
[14] www.systemverilog.in