

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

Multicore Enabled Verification of AMBA AHB Protocol using UVM

Pragati Agarwal

M. Tech (VLSI Student), Department of Electrical, Electronics & Communication Engineering, NCU University
(Formerly ITM University), Gurgaon, (Haryana), India

ABSTRACT: Intel cofounder Gordon E. Moore gave the Moore's law which states that the number of transistors double every two years. This statement cannot hold indefinitely, as the size of these transistors cannot be made infinitely small. On the other hand we are also reducing the chip area, which again has a limit. Since, the amount of data being processed is increasing at a really fast pace, so is the need of managing processor execution. Also, we have a greater number of processors coming up in the market. So, in order to meet the need of the hour, we have to come up with a solution and make full utilization of these multi-core processors. For the thesis work, a protocol is being worked upon which is the AMBA(Advanced Microprocessor Bus Architecture) AHB (Advanced High-performance Bus) protocol. This protocol would be designed for a high performance pipelined system. Following this, multi-threading concepts would be introduced using two different languages i.e. System Verilog as well as VLang. Since, SV does not support multi-threading on multiple cores so, the concept of multi-threading would be introduced using VLang which is a new language. Another advantage of VLang over SV is that VLang is a free and open source language, which makes it easier to access. Parallel processing in multi-core environment is the need of the hour, and the whole thesis work revolves around this. This brings novelty to the work. Hopefully, the readers find the work worthy, would help them in contributing to further growth and development of parallel processing.

KEYWORDS: System Verilog, UVM, VLANG, Multicore, Pipelining, Parallel Processing, AMBA AHB Protocol.

I. INTRODUCTION

ARM Ltd formed in 1990 as an Intellectual Property company that designs microprocessor technology. These microprocessors form the heart of digital products that vary from mobile phones and digital cameras to automotive systems and are widely used today. ARM provide standards in microprocessor architectures that are compatible with Windows and Linux and hence is widely accepted with a large network supporting design and development cycle[1]. ARM gave AMBA in the year 1996 which is an open-standard solution for connecting various functional blocks on an SoC. The first generation AMBA buses included ASB(Advanced System Bus) and APB(Advanced Peripheral Bus) followed by the second generation which introduced AHB(Advanced High-performance Bus). Also, for these standard protocols, we require verification IP's to get our design verified. VIPs ensure that the design under inspection is working as expected. There are various verification methodologies such as VMM, UVM, etc. UVM being the latest one. One of the most basic solution that ARM gave can be explained with the help of a diagram given in Fig. 1.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

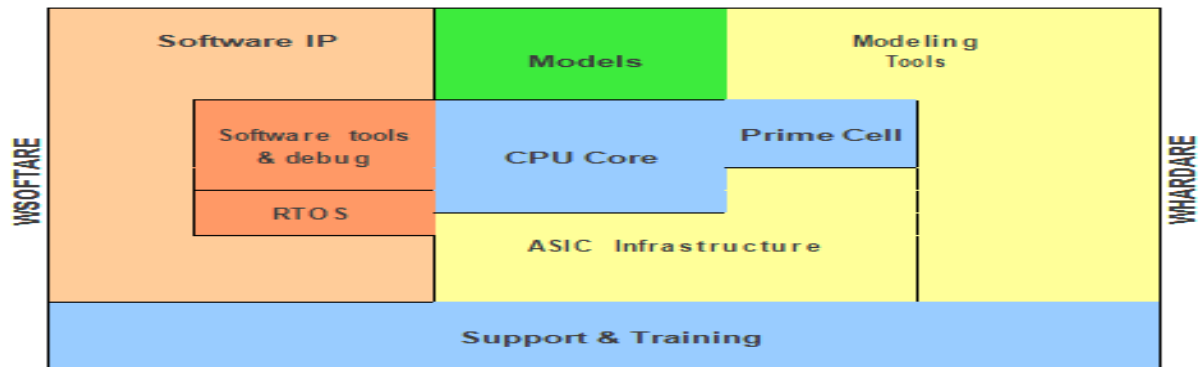


Fig. 1. ARM's Solution[1]

II. RELATED WORK

The complete paper revolves around how the Amdahl's law is applied in case of multicore architectures and its importance. At the beginning of the article, a comparison has been made on multicore chips made with asymmetric, symmetric and dynamic techniques with Amdahl's law applied to those. The comparison shows the upper bound on speedup for each one of these multicore chips. Following the comparison, an in-depth knowledge of Amdahl's law is given where speedup has been defined explicitly. This law has been explained using an equation which stands firm till today. A simple cost model for multicore chips has been taken and analyzed followed by a symmetric model for multicore chip. Eventually their results and implications have been discussed. Similarly, the results and implications of an asymmetric model for multicore chip has also been analyzed. A comparison between the symmetric as well as the asymmetric models has been made which is further supported by some graphs. Also, dynamic multicore chips are analyzed the same way. Concluding the paper, speedups for all the three chips has been evaluated and are supported with the help of three different equations [13].

III. AMBA PROTOCOL

As discussed earlier, ARM gave the world a standard for verification IP's called AMBA (Advanced Microcontroller Based Architecture). There are different generations of AMBA protocols namely AMBA AHB, AXI, ASB, APB et. al. Here, we are concentrating more on the second generation protocol called the Advanced High Performance Bus protocol. This chapter throws light on some of the important concepts required for understanding the research work and hence the VIP.

A. VARIOUS AMBA BUSES

The first generation AMBA bus contains ASB and APB bus protocols. The second generation contains AHB, while the third generation contains AXI. These buses may be explained in brief as under[18]:

- APB (Advanced Peripheral Bus): it is for low-power devices that are connected to the hardware. It keeps in mind minimum power consumption and also for reducing interface complexity.
- ASB (Advanced System Bus): it is used in system modules for a high-performance. It is an alternative for AHB only where its high-performance features are not needed.
- AXI (Advanced extensible Interface): it is a third generation bus protocol.

B. TYPICAL AMBA MICROCONTROLLER

This microcontroller contains basic peripherals connected to each other. These include internal and external memory devices, high-performance buses et. al. This can be seen with the help of a diagram.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

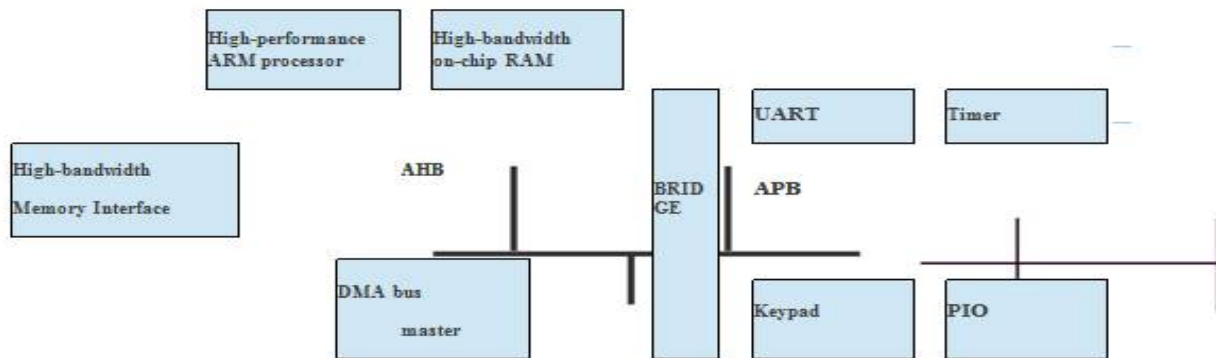


Fig. 2. AMBA microcontroller [18]

C. AMBA AHB

It is a second generation AMBA bus protocol which is used for synthesizable designs that requires high performance. It supports a high frequency of operation with multiple masters. The number of masters and slaves may vary, and data is transferred between the two using an arbitration mechanism. The main features of AMBA AHB includes burst transfer, split transfers, data buses with wider configuration et. al. [18].

A few of the components of AMBA AHB are:

- Master: it is used to send signals to the slave thereby initiating an operation which could be either a read or a write. These signals are address and control signals.
- Slave: after the execution of code, the response is sent back to the master by the slave via arbiter.
- Arbiter: it is used to ensure that the selected slave is sent the required signals via a proper mechanism.

D. AHB SIGNALS

The signal nomenclature contains an “H” at the starting of each signal name. Some of the signals used here are:

- HCLK : it is the bus clock. It is used to time all the data transfers. All the signals are made sensitive to the rising edge of the clock.
- HRESETn : it is an active low signal and is used to reset the bus.
- HADDR[31:0] : it is the address bus for the system.
- HTRANS[1:0] : it is used to recognize the type of transfer for a burst. The different types of transfer could be IDLE, BUSY, SEQUENTIAL and NON-SEQUENTIAL.
- HBURST : the different type of bursts are sequential, WRAP4, INCR4, WRAP8, INCR8, WRAP16, INCR16.
- HWRITE : this signal is used to determine the type of operation being performed. There could be two types of operation viz. read and write.
- HSIZE : this is used to indicate the transfer size which could be a byte, a word or a half-word.
- HWDATA[31:0] : it is generally a 32 bit data bus which is used to transfer data from a master to the respective slave. It is used for writing data.
- HRDATA[31:0] : it is generally a 32 bit data bus which is used to transfer data from a slave to the respective master. It is used for reading data.
- HREADY : it is used to check whether the transfer should start or not. If it is high then, a new transfer starts, and if low then no new transfer can start.
- HRESP : it is used to show the response from the slave after the complete data transfer.

E. SUMMARY

This chapter gave an overview of the AMBA AHB bus protocol where we learnt about the various signals that are used in creating the verification IP for AHB. Here, the features of AHB are also studied which revealed that it is a high-performance, high-frequency bus and hence is very useful in the hardware systems being used today.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

IV. VERSION CONTROL

As the technology is growing, the length of codes in the software world has also increased. So, in order to maintain these complex codes we need to have a proper system that can easily manage these lengthy codes. One of the tools that can be used to manage such work is the git version control tool. Here, in this chapter we will study about the advantages of managing our codes with this tool and would also learn some basics on how to do it.

A. VERSION CONTROL: AN INTRODUCTION

The first and foremost question that comes to our mind while reading something is what is it about? and why should it be used? Etc. The answer to these questions is that version control is a systematic way of managing the files over a period of time. If we make any changes inside the file, then these changes are recorded using this tool along with some basic information viz. the time of change, the places where the changes have been made, et. al. We can also get the difference between a previous file and a fresh file using a standard set of commands. Another advantage of using this tool is that the type of files that can be tracked could be of any type and need not be compulsarily a code file.

B. Various TYPES OF VCS (VERSION CONTROL SYSTEM)

In general, there could be three types of VCS viz. local version control systems, centralized version control systems, and distributed version control systems. These can be discussed in brief as under [19]:

– **Local version control systems (LVCS):** the most simple way of version-control that people usually prefer is to copy their files into another directory which could be a time-stamped directory. Though it is a simple method, it could prove to be extremely error prone, the reason being it is easier to make a mistake and write to a wrong directory. This system could be understood using the basic block diagram shown in Fig. 3.

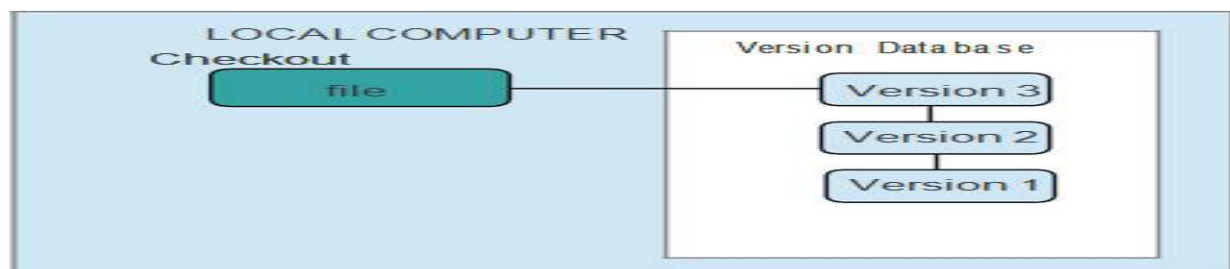


Fig. 3. Local Version Control System Block Diagram [19]

– **Centralized Version Control System (CVCS):** problem occurs when we have to share data with others which are on a remote device. The solution is to have a common server which contains all the files that are versioned and hence all the clients can checkout from that common server as in Fig. 4.

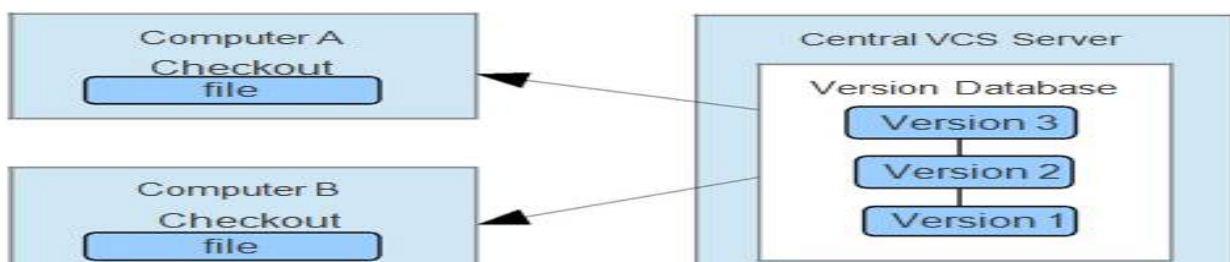


Fig. 4. Centralized Version Control System Block Diagram [19]

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

– **Distributed Version Control System Block Diagram (DVCS):** this is a more general approach which reaches a larger number of audience. Here, people can get the latest snapshots of any document and can even pull requests. Here, complete backups are available. See Fig. 5. for the block diagram.

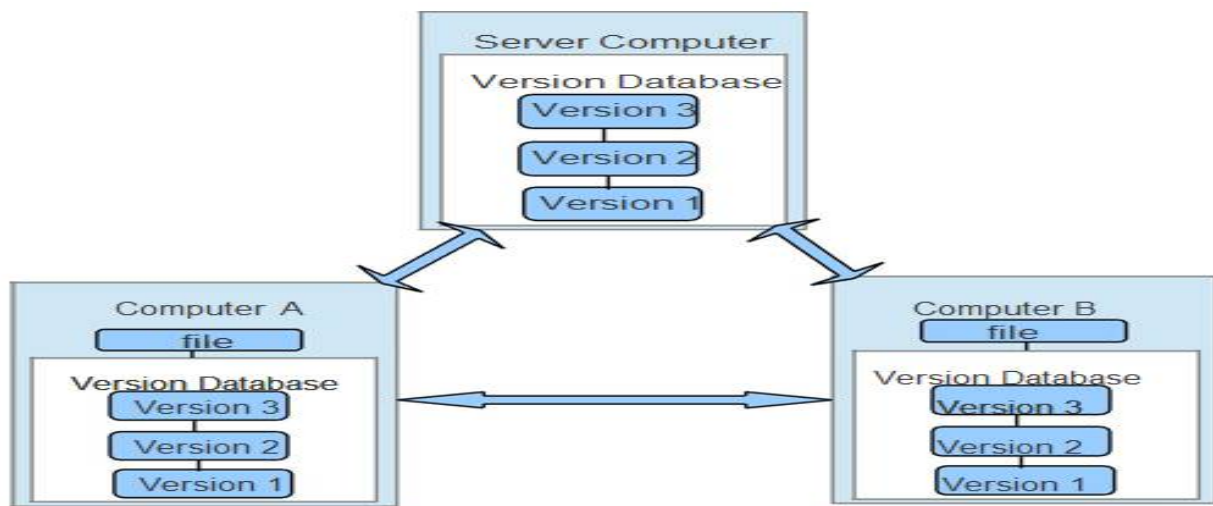


Fig. 5. Distributed Version Control System Block Diagram [19]

C. DIFFERENT STATES OF GIT

In order to understand the working of this tool, we must first understand how this tool requires three basic steps towards version control. These steps may be given as under [20]:

- **Commit** : it is used to store data in the local directory.
- **Modify** : it is used when we have made changes to the file but have not committed it yet.
- **Stage** : it means that the marked file has been marked which has to be committed further. This could also be understood by a simple diagram given in Fig. 6.

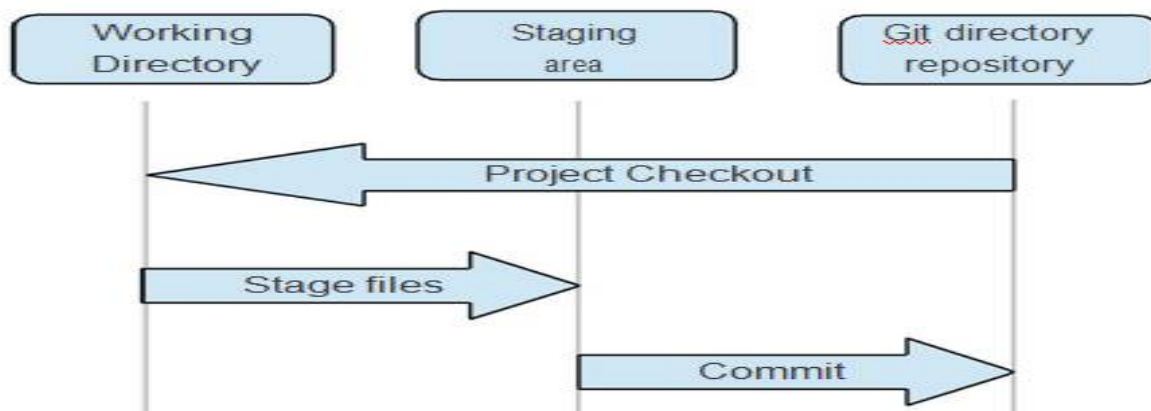


Fig. 6. Basic git states [20]

D. SUMMARY

Summarizing the chapter, I must say that we have learnt most of the basics that are required to know about this version control tool and how important it is to learn these tools. We also came to know of the most basic stages that are used in creating a new version of a file.

V. SIMULATION RESULTS

This chapter contains all the simulation results that are obtained after creating and running the verification IP for verifying the AMBA AHB protocol. Here, the outputs for both the verification IP's one being in SystemVerilog and the other one being in VLang.

A. SIMULATION RESULTS FOR AHB VIP – SYSTEM VERILOG AND VLANG

Here are the outputs after running the verification IP for the advanced high-performance AMBA bus.

The final output of the AHB SystemVerilog verification IP using UVM is obtained without having any errors and may be seen in Fig. 7. as given below

[illegible]

Fig. 7. Final AHB VIP in SystemVerilog using UVM

Also, after the successful creation of a verification IP for AHB written in System Verilog using UVM, the verification IP for the same has been created using a new language called as VLang that supports the multicore constructs.

```
-- UVM Report Summary --  
-- Report counts by severity  
Report counts by id  
[Severities]    0  
[Errors]        1  
[Warnings]      1  
[Infos]         6  
[Debugs]       298  
[TEST_DONE]
```

```
-- Shutting down all the active tasks ....  
Shutting down the Routine Threads ....  
Completed.  
UVM sequence controller phase is ready to run
```

Fig. 8. Different UVM phases in VLang for AHB

And, finally the complete output of the AHB VIP written and run in VLang simulator using UVM can be seen with the Fig. 9

Fig. 9. AHB VIP using VLang UVM

B. SUMMARY

Summarizing the chapter, I may say that the verification IP for AMBA AHB bus protocol using SystemVerilog UVM as well as VLang UVM has been successfully created and the outputs have been shown clearly.

International Journal of Innovative Research in Science, Engineering and Technology

(A High Impact Factor, Monthly Peer Reviewed Journal)

Vol. 5, Issue 2, February 2016

VI. CONCLUSION AND FUTURE SCOPE

This research work in which a verification IP for AMBA AHB bus protocol has been created and hence verified. This work was carried out using two different languages viz. SystemVerilog and VLang (both using UVM methodology). Multicore constructs have been enabled using the latter which is a free and an open source software and which supports multicore constructs. This is a novelty that the same two work were carried out on two different languages. Also, since the technology is going more towards the multicore side, the discussion and hence the research work has also been towards the multicore side.

Also, for the future, it may be said that this work could be taken to higher levels with a deeper study involved in understanding this novel language that supports multicore constructs. A further higher level of complexity may be added to the design by adding the concept of a Register Abstraction Layer called the RAL layer.

REFERENCES

- [1] Ioannis Skazikis, "The ARM Processor Architecture," Matr-Nr. 637868
- [2] Bergeron Janick, "What is Verification?," in *Writing Textbenches Using SystemVerilog*, Ed. New York: Springer, year, pp. 2-21
- [3] Bergeron Janick, "What is Verification?," in *Writing Textbenches Using SystemVerilog*, Ed. New York: Springer, year, pp. 38-39
- [4] <http://www.accellera.org/about>
- [5] Kathleen A Meade and Sharon Rosenberg, "A Practical Guide to Adopting the Universal Verification Methodology (UVM)," Second Edition
- [6] JamesT. Townsen d, "Searial vs. Parallel Processing: Sometimes they look like Tweedledum and Tweedledee but they can (and Should) be Distinguished," *Psychological Science*, Vol. 1, No. 1 (Jan., 1990), pp. 46-54 . URL: <http://www.jstor.org/stable/40062391>
- [7] <http://spectrum.ieee.org/computing/software/the-trouble-with-multicore>
- [8] verlang.org
- [9] Stuart Sutherland and Tom Fitzpatrick, "Keeping Up with Chip — the Proposed SystemVerilog 2012 Standard Makes Verifying Ever-increasing Design Complexity More Efficient,"
- [10] Alan J. Hu , "Formal Hardware Verification with BDDs: An Introduction,"
- [11] Abhishek Jain¹, Giuseppe Bonanno², Dr. Hima Gupta³ and Ajay Goyal ⁴, "Generic System Verilog Universal Verification Methodology Based Reusable Verification Environment For Efficient Verification Of Image Signal Processing Ips/Socs," *International Journal of VLSI design & Communication Systems (VLSICS)* Vol.3, No.6, December 2012 , DOI : 10.5121/vlsic.2012.3602
- [12] Geoffrey Blake, Ronald G. Dreslinski, and Trevor Mudge, "A Survey of Multicore Processors," *IEEE Signal Processing Magazine* [26] November 2009 , DOI : 10.1109/MSP.2009.934110
- [13] Mark D. Hill and Michael R. Marty, "Amdahl's Law in the Multicore Era ,"
- [14] Joseph Gasparakis and Peter P Waskiewicz, Jr., "Design considerations for efficient network applications with Intel® multi-core processor based systems on Linux," white paper, July 2010
- [15] "Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling," *ICSA'05 Proceedings of the 32nd annual international symposium on Computer Architecture*, IEEE Computer Society Washington, DC, USA ©2005, pp. 408-419, DOI : 10.1109/ISCA.2005.34
- [16] Shobana Sudhakar and Rohit K Jain , "The Need for Speed: Understanding design factors that make multi- core parallel simulations efficient ," DVCon
- [17] Vanessa R. Cooper, "Getting Started with UVM," First Edition, Austin, USA, © 2013, 2012 Verilab, Inc. - <http://www.verilab.com>
- [18] "AMBA™ Specification (Rev 2.0)," © Copyright ARM Limited 1999 , 13th May 1999, Issue A, First release
- [19] <http://git-scm.com/book/en/Getting-Started-About-Version-Control>