iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** IV **Month of publication:** April 2020

DOI:

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Development of AHB-Lite System Verilog based VIP with Assertion

Kavisha Parikh[1], Mr. Kirit Patel[2], Mr. Vinod Ailsinghani[3]
[1]*M.E. Student, L.D. College Of Engineering, Ahmedabad, Gujarat, India*
[2]*Assistant Professor, L.D. College Of Engineering, Ahmedabad, Gujarat, India*
[3]*T.A. VLSI verification, eInfochips, Ahmedabad*

Abstract: ***Verification is the important part of SoC Development, it ensures DUT functions as per the specification or not. The SoC uses an on-chip bus protocols to exchange data. The most widely used on chip bus architecture is AMBA (Advanced Microcontroller Bus Architecture) bus which is introduced by the ARM. AMBA AHB (Advance High Performance) Lite bus addresses the requirements of high-performance and high-bandwidth synthesizable designs. AHB-Lite is verified by developing the reusable verification intellectual property (VIP) using hardware description and verification language Systemverilog by generating System Verilog environment. By writing assertions concurrently with the RTL design and keeping these assertions closely tied to the RTL code bring significant benefits in both the verification as well as design team processes for digital hardware. So, in this paper AMBA AHB-Lite bus verification environment is built which is verified by assertion. Synopsys VCS-MX Tool with gtkwave and dve waveform viewer will be used to develop AHB-Lite SV based reusable VIP.***

Keywords: AMBA (Advance Microcontroller Bus Architecture), AHB-Lite, System Verilog Assertion(SVA), System Verilog, Verification IP.

I. INTRODUCTION

Verification is very challenging as well as necessary task for the designer in the whole design because bugs which are uncovered in the earlier stages of the design will be carry on its next stages of the design and then it is too difficult to diagnose it.

Verification is mainly verifying whether our implementation matches with the micro-architectural specification or not. Any design is insufficient without proper verification of that designs. Almost 70% time of the overall project is consumed for developing verification environment. So, it is important to reduces the verification time, to improve the complete development process. But on the other side the complexity of the design makes it difficult to cover all the corner cases in least time.

The Advanced Microcontroller Bus Architecture (AMBA) bus protocols is a set of interconnect specifications from ARM that standardizes on chip communication mechanisms between various IPs or functional blocks for building high performance SOC designs. AMBA consists of verities of protocols like APB, ASB, AHB or AXI as per the system requirement. AHB (Advanced High-Performance Bus) is used for connecting component that need higher bandwidth on a shared bus. AHB-Lite is simplified version of AHB protocol. VIP (Verification Intellectual Property) is a type of reusable IP that can generate Complete tests for shortening SoC verification and increasing test coverage. VIP (Verification IP) is nothing but advanced layered testbench or a predefined functional block that can be inserted into the testbench. It can then be used to actually simulate the design (either an IP or an SOC) and verify the functional correctness of it. Assertion Based Verification is used to improve the design observability and to detect and interpret its faults. It is one of the recommended verification techniques to enhance the verification quality and reduce the debugging time of complex system-on-chip designs. Assertions helps to detect the failure and thus reduce the effort to establish the exact reason of the failure. By writing system Verilog assertions implemented on the design can speed up the verification process. Verification process debugging is divided into three stages: 1. error detection 2. error diagnosis and 3. error correction. The main advantages of keeping assertions in a separate file is that they can be independently verified without the need to control of RTL files.[3]

II. AMBA AHB-LITE PROTOCOL

AMBA AHB-Lite addresses the requirements of high-performance synthesizable designs. It is a bus interface that supports a single bus master with multiple slave which provides high-bandwidth operation.[1]

A. AHB-Lite Implements The Features Required For High Clock Frequency, High-Performance Systems That Including

1) Burst transfers
2) Single-clock edge operation
3) Non-tristate implementation
4) Wide data bus configurations, 64, 128, 256, 512, and 1024 bits.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

Figure 1 shows AHB-Lite system design with one AHB-Lite master and three AHB-Lite slaves. The bus interconnect logic consists of a slave-to-master multiplexor and an address decoder. The decoder continuously monitors the address from the master so that the relevant slave is selected and the multiplexor routes the Consonant slave output data back to the master.

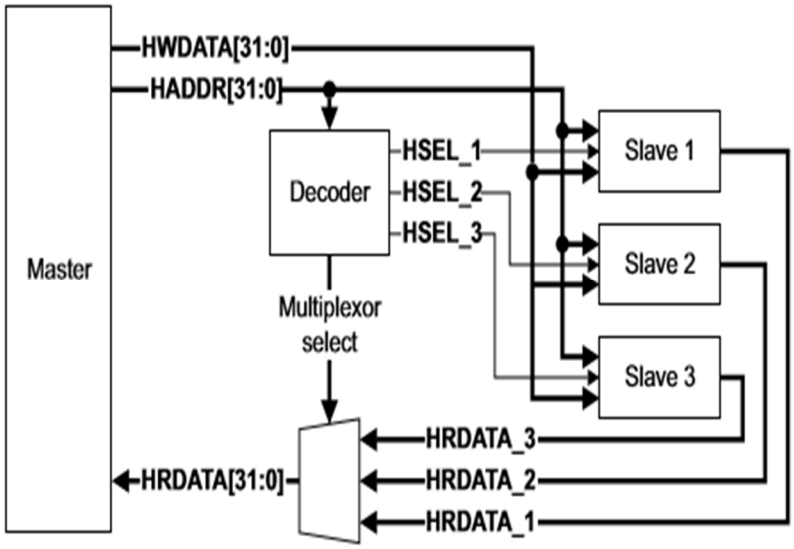


Fig 1. AHB Lite block diagram[1]

An AHB-Lite master starts a transfer by driving the address and control information to initiate read and write operations. Figure 2 shows an AHB-Lite master interface signals. These signals give information about the address, width of the transfer, direction, and also gives information about transfer whether it is parts of a burst or not.

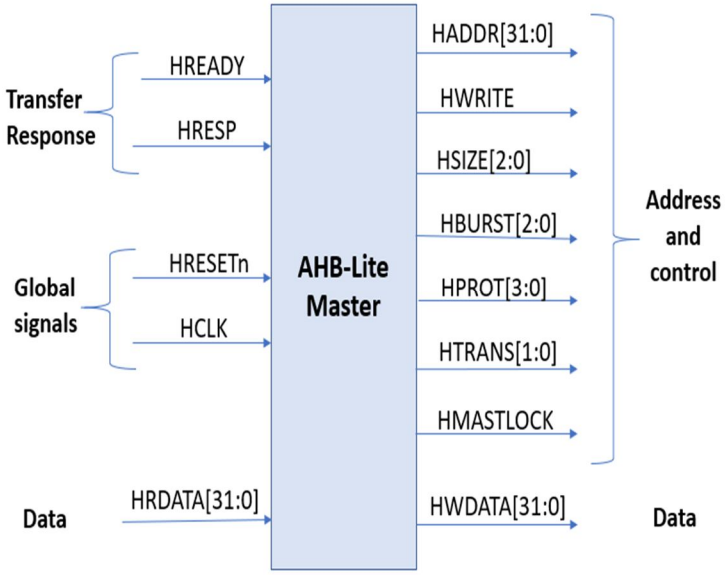


Fig 2. AHB Lite Master signals

Transfer can be

a) single
b) Incrementing burst
c) Wrapping burst that wrap at particular address boundary

Figure 3 shows an AHB-Lite slave interface signals. An AHB-Lite slave responds to transfers which is initiated by masters in the system by giving Hready and Hresp signals.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue IV Apr 2020- Available at www.ijraset.com

B. The slave Signals Respond back to the Master

1) Success
2) Failure
3) Or waiting of the data transfer.

When hwrite signal is 1 then write data bus moves data from the master to the slave and when hwrite is 0 then the read data bus moves data from a slave to the master.

The slave uses hresp signal to indicate the success or failure of a transfer. Slave can also request the master to extends the data phase by using hready signal, when hready goes low it causes wait states to be inserted into the transfer and so it enables the slave to have extra time to provide or sample data.

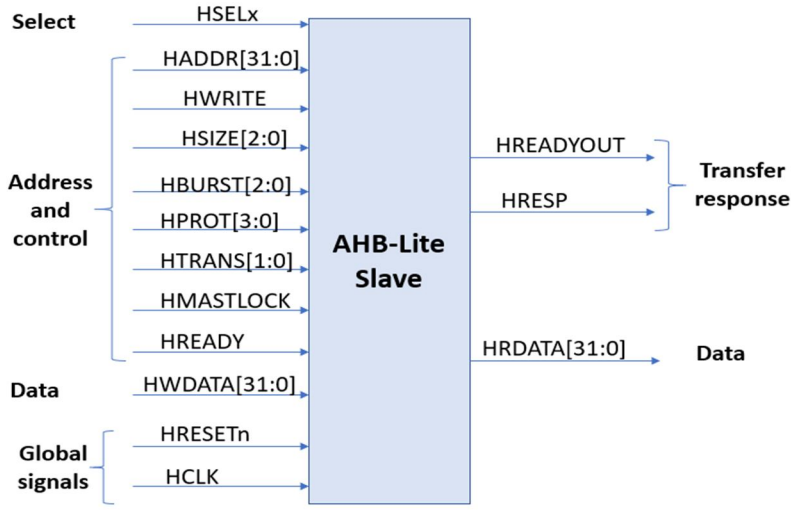


Fig 3. AHB Lite Slave signals

III. SYSTEM VERILOG TESTBENCH ENVIRONMENT

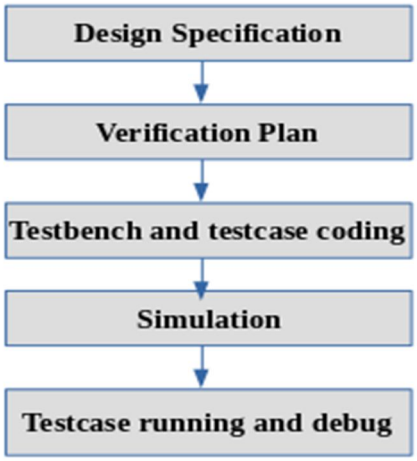


Fig 4. Verification Flow

As figure 4 shows verification flow for making Verification IP. First step is to read and understand design specification. Then we have to design verification test plan which contains list of testcases, list of features to be tested, approach, resources, risk and scheduling, entry and exit criteria. Test plan helps to understand how the verification should be done. Then In this testbench and testcase coding phase developed verification environment. After this step simulation done in Synopsys VCS-MX tool and shows waveform in dve or gtkwave. Also checking all testcases by running and debugging it.

Now SystemVerilog testbench architecture is as shown in figure 5 which contains Top, Test, Environment, Agent, Driver, Monitor, Generator, Interface, DUT. Testbench or Verification Environment is used to check the functional correctness of the DUT by generating and driving a predefined input sequence to a design, capturing the design output and comparing with-respect-to expected output.

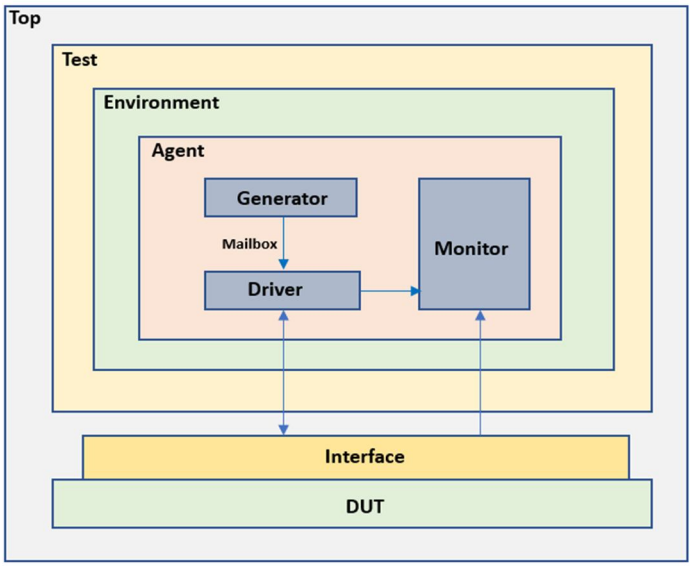


Fig 5. Systemverilog Testbench Architecture

Verification environment is a group of class's performing specific operation. Generator generate the packet and send it to driver using mailbox. Driver receives the packet from generator through mailbox and drive it to DUT through interface. Monitor monitors all the signal. Agent consists of generator, driver and monitor class. In environment class handle of agent class is created and gives all control to test. Test is responsible for configuring testbench and initiate the stimulus driving. The topmost file, which connects the DUT and Testbench. It consists of DUT, Test and interface instances, the interface connects the DUT and Testbench. The main purpose of using Systemverilog is reusability.

IV. SYSTEM VERILOG ASSERTION

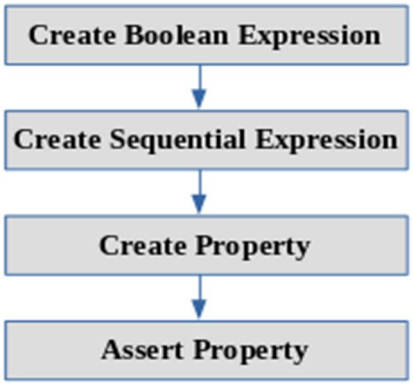


Fig 6. SVA Building Block

Different properties of AMBA AHB-Lite is verified by writing Assertion for the properties and verified it using Synopsys VCS tool and dve waveform. Fig. 6 shows the steps involved in the creation of an SVA checker. Boolean expression is the functionality which is represented by the combination of multiple logical events. Sequence is the Boolean expression events that evaluate over a period of time involving single or multiple clock cycles. A number of sequences can be combined sequentially or logically to create more complex sequences called property and this property is verified during simulation. So SVA provides “assert” keyword to check the property. The AMBA AHB-Lite is verified using different assertion property as shown in table 1.

Table 1. Assertion Property Description

No.	Assertion Property	Status
1.	Size should be constant until the burst has completed.	PASS
2.	Single burst followed by non-sequential or idle transfer but not followed by busy transfer or sequential transfer	PASS
3.	When reset comes all the signals like size, burst, transfer, write, write data and address goes in default state	PASS
4.	When ready signal is low then address, write and write data should remain in the same state until ready goes high.	PASS
5.	Checks for 4 beat wrapping or Incrementing bursts whether the state transitions is going on properly, i.e. non-sequential followed by 3 sequential states	PASS
6.	Checks for 8 beat wrapping or Incrementing bursts whether the state transitions is going on properly, i.e. non-sequential followed by 7 sequential states.	PASS

V. SIMULATION AND RESULT

For making VIP of AMBA AHB-Lite protocol in Systemverilog, we are taking different test case scenarios:

A. Single Transfer

As shown in figure 7 when global signals clock and reset is high then transfer is starts. In this hburst is 000 means single transfer and htrans is 00 means idle transfer means no transfer and when it is 10 then nonseq transfer is there. hwrite is 1 then write operation is performed and we get write data in hwdata after one cycle.

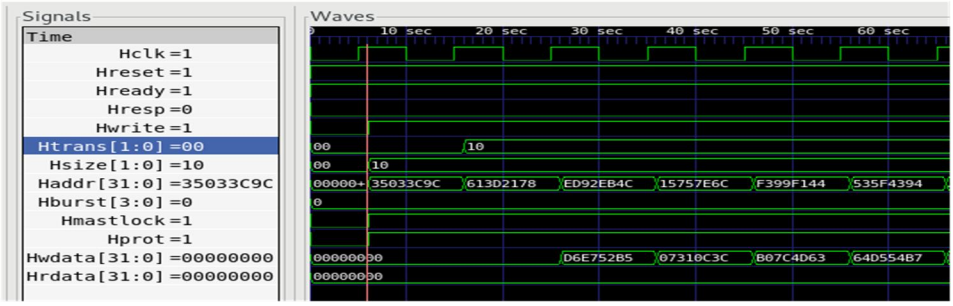


Fig. 7 Single Transfer with Okay response

B. Single Transfer with error Response

As hburst is 000 means single transfer so that it should be non-sequence transfer but as shown in figure 8. In that htrans signal is 01 and 11 which means busy and sequence transfer respectively.

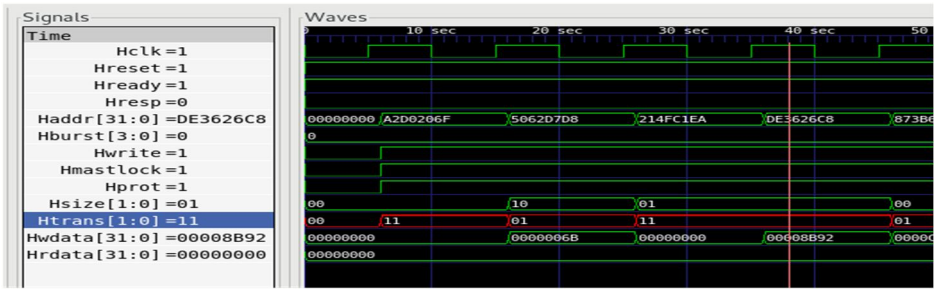


Fig. 8 Single Transfer with Error response

C. INCR 4 (4 Beat Incrementing Burst)

As hburst signal is 011 means INCR 4 is done. As shown in figure 9 size is 10 means word then address is incremented by 4 bytes and we get 4 back to back transfer 1st nonseq and rest of 3 beats are sequential.

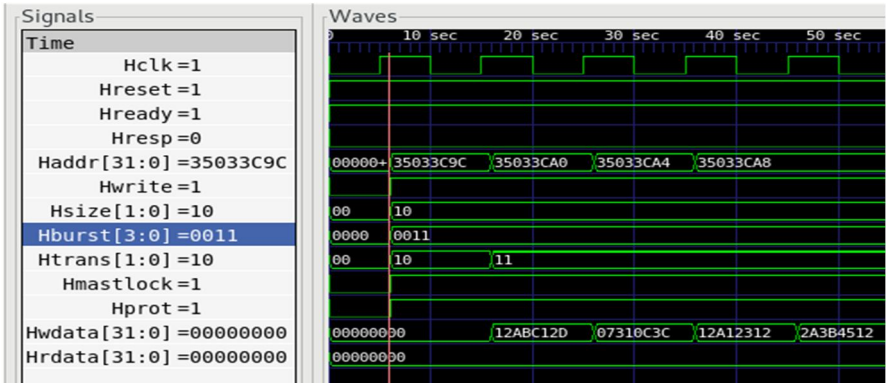


Fig. 9 INCR 4 (4 beat incrementing Burst)

D. INCR 8 (8 Beat Incrementing Burst)

As hburst signal is 101 means INCR 8 operation is done. As shown in figure 10 size is 01 means halfword then address is incremented by 2 byte and we get 8 beat means one nonseq and rest of 7 sequential transfer.

E. WRAP 4 (4 Beat Wrapping Burst)

When hburst is 010 then 4 beat wrapping bursts is done. As hsize is 10 then word transfers, so the address wraps at 16 bytes boundaries. The transfer to address 0xAC is followed by a transfer to address 0xA0 as shown in figure 11.

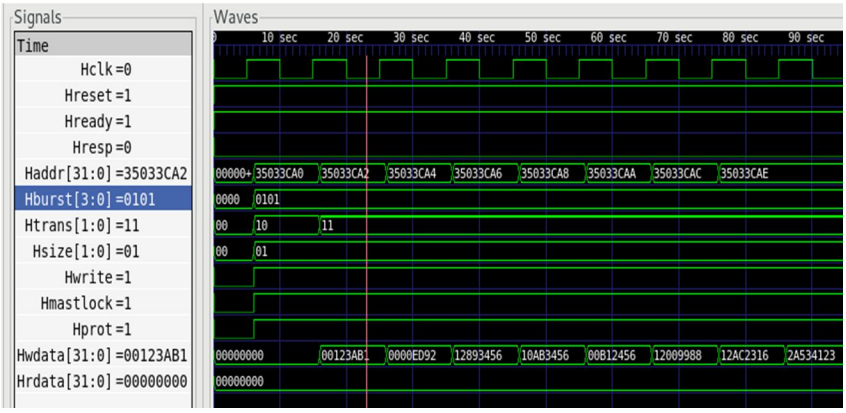


Fig. 10 INCR 8 (8 beat incrementing Burst)

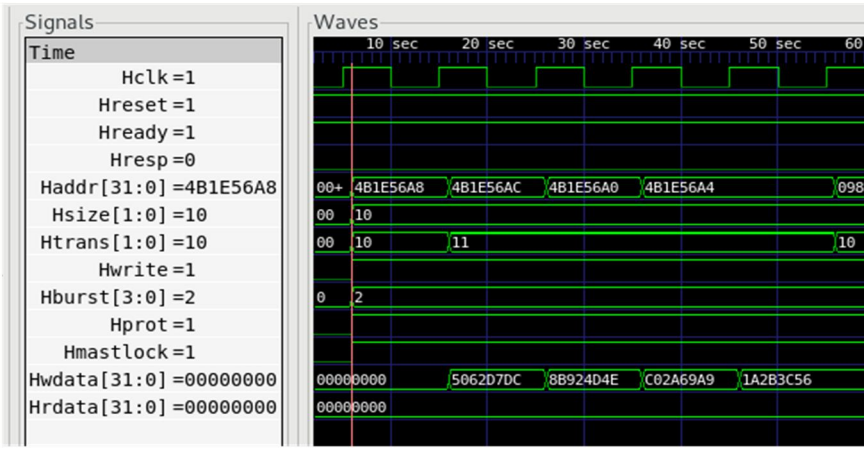


Fig. 11 WRAP 4 (4 beat Wrapping Burst)

F. Assertion Waveform

As shown in fig. 12 & 13 there is assertion waveform which gives success or failure of Testcases. When arrow is upper and green color then it is gives success and if it is lower and red color then it is failure of assertion property.

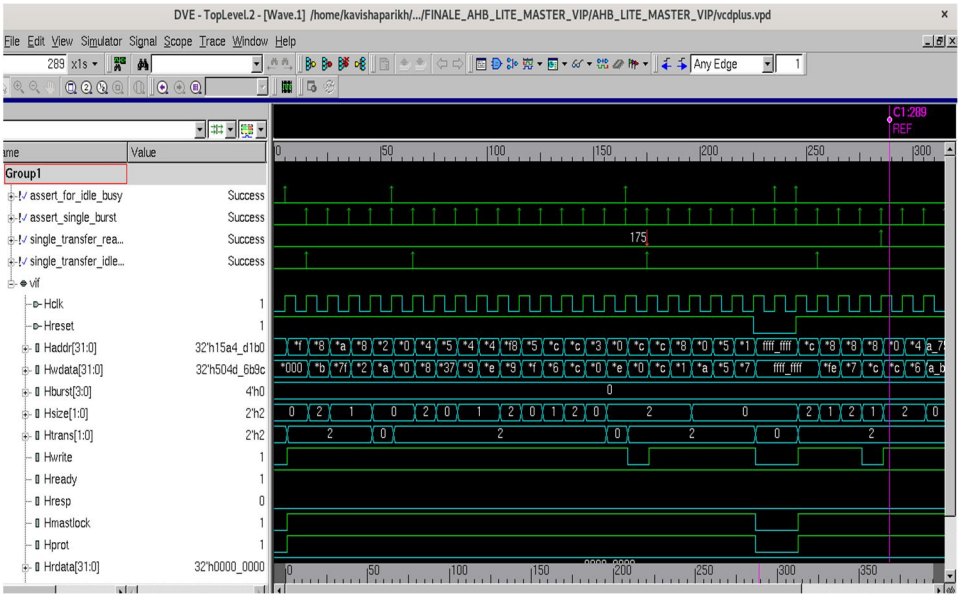


Fig. 12 Assertion Simulation 1

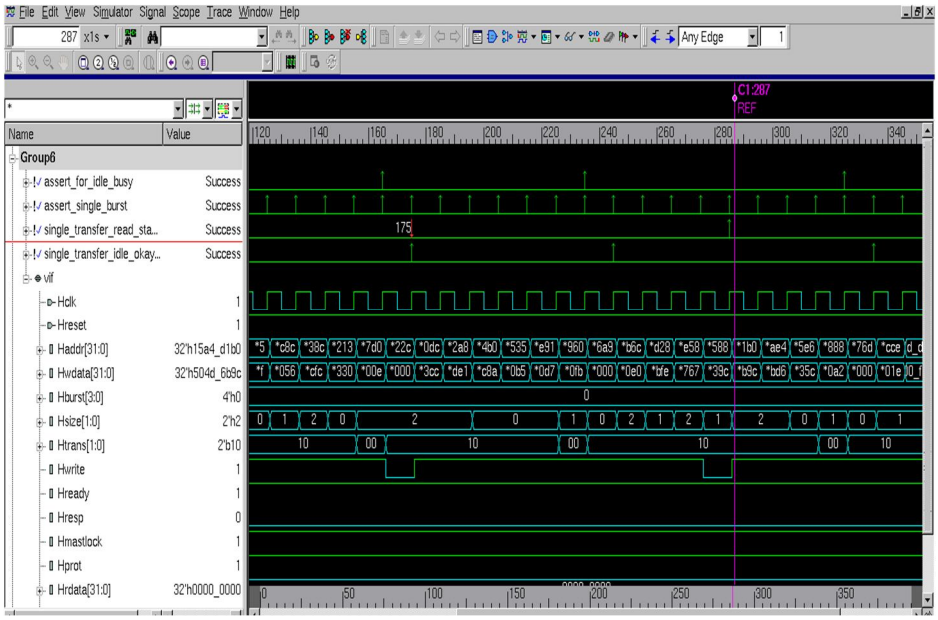


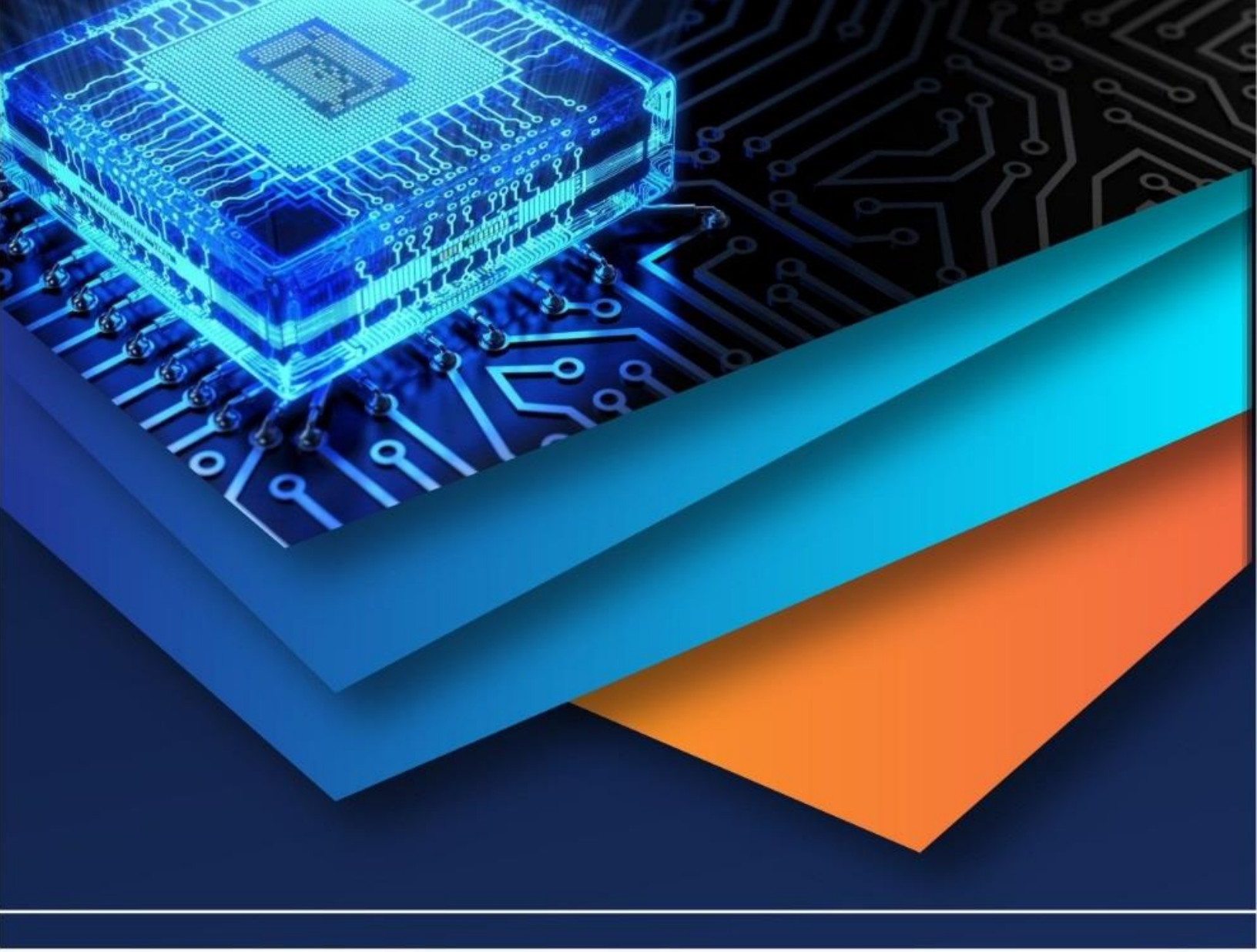
Fig. 12 Assertion Simulation 2

VI. CONCLUSION

Verification is mainly the process of inspecting, reviewing, testing, verifying and proven that the product behaves is as per the requirement of specification. The AMBA AHB-Lite protocol is designed, implemented and verified using Systemverilog along with assertion. It supports full features of AHB-Lite protocol as per the ARM specify it. It mainly supports this feature: 1. Single clock edge operation 2. Separate bus for read and write operation, 3. Pipelined operation, 4. Single transfer and 5. burst transfer like Incrementing burst and wrapping burst. Along with system Verilog Assertion, it enhances the verification quality and reduce the debugging time of complex SoC design by speed up the verification process. The different properties of AMBA-AHB and all its corner cases properties are verified by making verification environment using VCS-MX tool and shows its waveform in gtkwave and assertions waveforms are in dve waveform viewer.

REFERENCES

[1] Online] ARM AMBA AHB-Lite protocol specifications and design tools [www.arm.com]
[2] Soo Yun Hwang, Dong Soo Kang, Hyeong Jun Park, and Kyoung Son Jhang, "Implementation of a Self-Motivated Arbitration Scheme for the Multilayer AHB Bus matrix" IEEE transactions on very large-scale integration (VLSI) systems, vol. 18, no. 5, may 2010
[3] Prince Gurha, R. R. Khandelwal "SystemVerilog Assertion Based Verification of AMBA-AHB" IEEE 2016 International Conference on Micro-Electronics and Telecommunication Engineering.
[4] .Deeksha L Shivakumar B.R "Effective Design and Implementation of AMBA AHB Bus Protocol using Verilog" IEEE 2019 International Conference on Intelligent Sustainable Systems (ICISS 2019).
[5] Chris Spear, Greg Tumbush. "System Verilog for Verification: A Guide to Learning the Testbench Language Features": Springer, 2012.
[6] Nguyen Son Lam Duc Minh Nguyen "AHB-Master Controller Formal Compliance Verification" IEEE 2014
[7] Jain, G. Bonanno, H. Gupta and A. Goyal, 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 (2012).
[8] [online] system Verilog assertion by einfochips [www.einfochips.com/blog/systemverilog-assertion]
[9] "Introduction to SVA", A practical Guide for Systemverilog Assertions, 2005, Microprocessor Systems, 1995 & understanding behavioral synthesis, 1999 & ARM Ltd., AMBA specification (rev.2) 1999.
[10] [online] SystemVerilog testbench architecture by [https://verificationguide.com/systemverilogs/systemverilog-testbench] [online] SystemVerilog testbench architecture by [https://verificationguide.com/systemverilogs/systemverilog-testbench]
[11] [online] Tutorial for VCS [https://classes.engineering.wustl.edu/ese461/Tutorial/synopsys vcs tutorial.pdf]



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY