International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Verification of High Speed on Chip with VIP using System Verilog

Yoddha Shivani Kalpeshkumar[1], Prof. Suresh Bharwad[2], Er. Bhavinkumar R. Patel[3]

[1] *M.E (VLSI Design), School of Electronics and communication, L. D. College of Engineering Campus, Ahmedabad, India.*

[2] *Assistant Professor, School of Electronics and communication, L. D. College of Engineering Campus, Ahmedabad, India.*

[3] *Senior Technical Lead at Einfochips an Arrow company, Ahmedabad, India*

Abstract - The exploration work is addressing verification of High speed on chips protocol; we've used the system Verilog grounded test bench structure. I developed a system Verilog law and vindicating using Synopsys VCS tool & EDA playground (open source). Deployment of a layered test bench verification methodology, combined with the use of constrained random verification techniques, which are required to meet the challenge of verifying a subsystem which uses the AMBA on-chip Bus protocol. The verification of high- speed on- chip personality (Verification IP) protocol using System Verilog is a critical aspect of designing complex systems- on- chip (SoCs) that apply the Advanced extensible Interface (ON-CHIP) protocol. The ON-CHIP protocol is a extensively used interface standard for communication between different functional blocks within an SoC, and it plays a pivotal part in icing effective data transfer and synchronization between these blocks. The verification of the ON-CHIP protocol involves expansive testing of colorful functional scripts, including read and write deals, burst transfers, address and data range transformations, and protocol timing constraints. The verification process requires comprehensive testbench development, encompassing the generation of valid ON-CHIP deals, monitoring of protocol signals, and analysis of responses from the DUT (Design Under Test). Also, functional content analysis, assertion- grounded verification, and formal verification ways are employed to insure complete and robust verification of the ON-CHIP protocol. The challenge in vindicating high- speed on- chip personality protocol lies in the complexity and performance conditions of ultramodern SoCs, which demand effective and rigorous verification methodologies to achieve high quality and dependable designs. likewise, the adding demand for high- speed and low- quiescence communication in advanced SoCs necessitates the verification of ON-CHIP protocol at different timepiece frequentness and data transfer rates, posing fresh challenges in achieving accurate and effective verification. This abstract presents the need for vindicating the high- speed on- chip personality protocol using System Verilog, fastening on the challenges and methodologies involved in achieving comprehensive and dependable verification of the ON-CHIP protocol. It highlights the significance of advanced verification ways and methodologies to ensure error-free operation of the ON-CHIP protocol in complex SoCs, leading to robust and dependable system designs. A directed testing methodology cannot produce enough system stimulants to reach the needed content pretensions in the docked design cycle. Exercise and scalability are another crucial consideration with verification as well as with IP. The verification methodology must support exercise similar that tests at one hierarchical position can be reused at the coming. Position over, as well as with the coming design. With a layered verification approach, lower layers like protocol verification are reused in advanced situations. Also, to meet up with coming generation forthcoming 5G technology by probing on on-chip protocol is largely used to acclimatize reusability in perplexed design verification part. So, at the expansive position this protocol is extensively used in numerous diligences as a connecting block to meet the anticipated and factual labors.

Keywords: Verification-IP, on-chip Protocol, System Verilog, Verification Environment.

1. INTRODUCTION

In this chapter, the problem description, High speed on chip design, and prolusion to the provocation of the algorithm are curtly described. The Advanced Microcontroller Bus Architecture (AMBA), specs define an on-chip dispatches standard for designing high performance of 32- bit and 16- bit Embedded microcontrollers. It was introduced by ARM Ltd in 1996 and is considerably used as the ON- CHIP machine communication in System- on-a-chip (SOC) designs. The 1st AMBA buses was Advanced System Bus (ASB) and then Advanced Peripheral Bus (APB). In its 2nd interpretation, AMBA 2, ARM advanced AMBA High- performance machine (AHB) that is a single timer- edge protocol. In its 3rd generation, AMBA 3, ARM introduced including Advanced extensible Interface (ON- CHIP) to reach indeed advanced performance interconnects. These protocols are now the de facto standard for 32- bit bedded processors because they are well proved and can be used without royalties. AMBA on-chip is the bottommost addition to the AMBA family advancing three new interface protocols AX14 to minimize the overall performance and power effectiveness: AX14- Lite and AX14Stream ideal for performance in FPGA etc.

2. READ TRANSACTION PHASE

Parallel reads work in a similar way, but in this case, multiple read

transactions can occur simultaneously. The master can request multiple data transfers from a slave by specifying a burst length, data size, and starting address. The slave then sends back the requested data in a burst transfer.

Both parallel reads and writes are efficient ways to transfer large amounts of data over the on-chip bus. However, care must be taken to ensure that the bus is not overloaded, as this can lead to data corruption or loss. The protocol also includes mechanisms for error detection and correction to help ensure the reliability of data transfers. The entire architecture/operation of the read transaction is pictorially represented in fig.1.

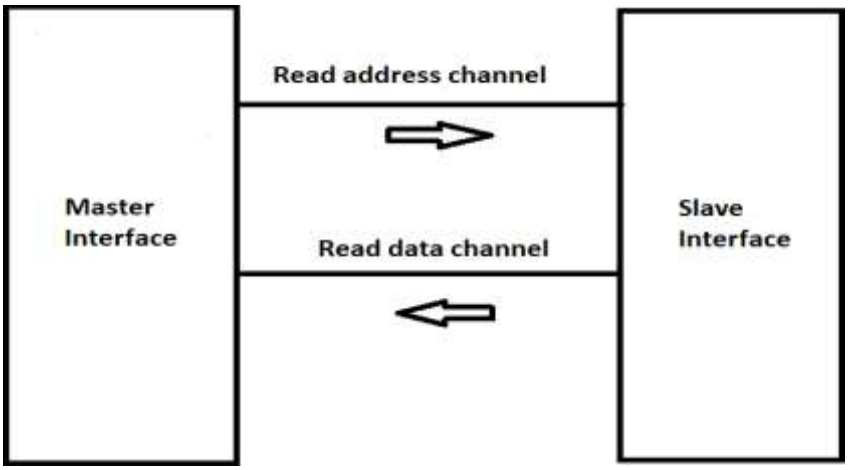

Figure 1: On-chip Parallel Read Transactions

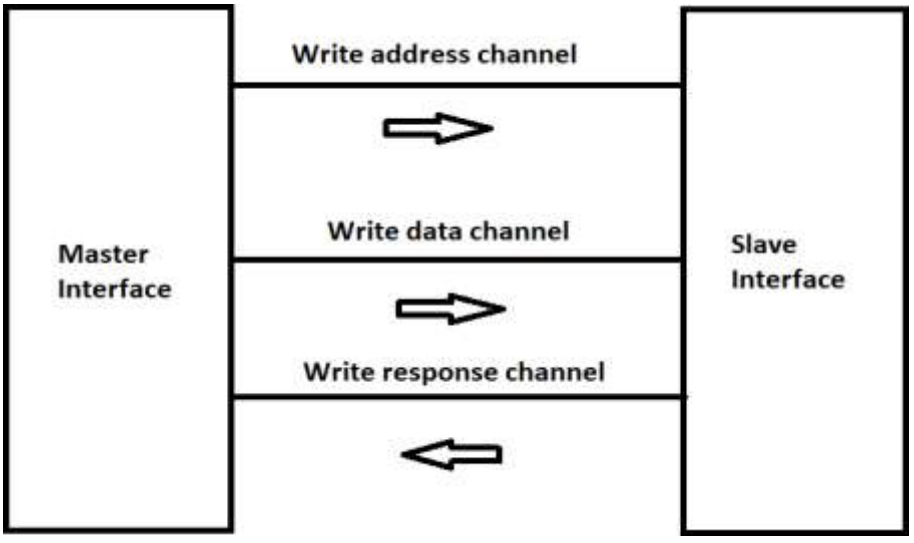

Figure 2: On-chip Parallel Write Transactions

3. WRITE TRANSACTION PHASE

In the case of resemblant writes, the on- chip machine allows for multiple write deals to do contemporaneously. This is achieved using burst deals, where a single master can shoot multiple data transfers to a slave without having torn-arbitrate for the machine. The master can specify the length of the burst, the size of the data transfer, and the address of the first transfer in the burst. The write deals are passing in the following three modes as i) Write address channel, ii) Write data channel and iii) Write response channel. In write address channel, the master will be starting the communication with the control signals AWVALID and after that Write channel access/ address to the slave (i.e it sends the valid address and channel control and address). incontinently the slave responds with a valid AW ready. In the same manner during the Write data channel mode, the master initializes the deals with a WVLAID and Write channel data, following to that the slave acknowledges by a valid WREADY signal. Like the data and address mode, the Write response channel mode involves the sale started by master with the control signals BVALID and write response channel. While entering the control signal, the slave will then establish the communication by valid BREADY signal. The operation of the write deals is shown in fig. 2.

4. PROPOSED VERIFICATION ENVIRONMENT

The verification environment is a standard method of verifying the DUV in code coverage mode. The following components are present in the proposed verification environment i) Test-suite, ii) Generator, iii)agent, iv) Driver, v) on-chip Monitor, vi) Protocol checker, vii) Bus Monitor and viii) Scoreboard.

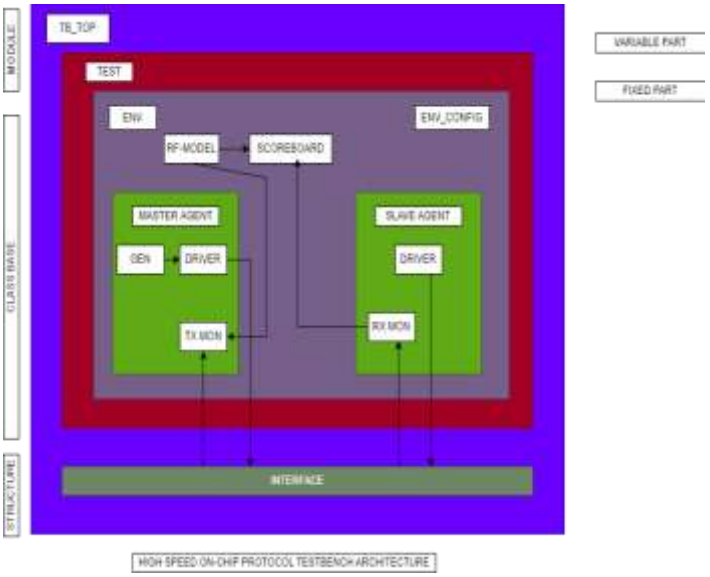

Figure - 3 Verification Environment

Test-Suite

Test-suite consists of all the test-cases for checking the specific key features of the DUV and the functionality in a constrained random mode. The test-case is of the form constrained random mode or direct random test-case. The test is regressively performed on the design and analyzed. It normally contains a set of stimuli for running the design.

Generator

Generally, the generator will generate the master transaction as the testbench is based on that so, the generator class will also generate the testcases in the same class. This component will then test the testcase which is to be test the sequence that is maintained for the entire test bench.

Agent

An agent is a component, which is either the active or passive agent. Well, in this case I must use two agents for master and the slave. Agent consisted of three other components which will then later be connected to further operations. To run the entire simulation.

Driver

Driver is the component which interfaces or forces all other components to respond accordingly to the feeder stimulus (i.e simply it drives the input to the corresponding blocks).

Monitor

The monitor watches all the data transactions of the all the components and reports the result to the coverage collector. It has a TLM port for analysis and virtual interface to handle the points of the DUV signals.

Protocol checker

In this module, the DUV functionality is verified, the interface compliance is also checked with the temporal behavior of the inputs. Also, the responses are monitored for all the combinations of the input sequences.

Bus monitor

Bus monitor screens the functional operation of individual components and connects the master/slave monitor. Finally, the transactions are collected in the coverage collector as database.

The above prescribed components are used in both themaster and slave modes in the AMBA ON-CHIP bus architecture. The individual modules are modeled in system Verilog and integrated as a main top module. The system Verilog instantiation for individual modules and top modules are

given as follows.

```
module tb_top;
  bit ACLK, ARESETn;              // Clock and reset signal

  always #5 ACLK = ~ACLK;         // Toggle the clock


  initial begin
    ARESETn = 0;                  // Reset condition (Active low
reset)
    #5 ARESETn =1;                // Reset 1
  end
……..……
  on-chip_interface pif(ACLK, ARESETn); // Interface instance
on-chip_test test_inst(pif);   // Passing interface instance in the
test file
……..……
  // Initial Block //
  initial begin
    $dumpfile("dump.vcd");   // dumpfile
    $dumpvars;
    ……..……
    ……..……
```

In the same manner the on-chip Master/Slave and the individual components of the verification environment are modeled using System Verilog. The instances of the on-chip top module pseudo-code are given partially as above. The above prescribed Verification Environment is used for the functional verification of the AMBA Bus protocol.

5. VERIFICATION PLAN

The verification plan tells the property must be vindicated and drives criteria to be satisfied. A good plan contains measurable criteria and a successful test- case to achieve check its functionality. The plan of verification tells what to be vindicated, how to corroborate it. In this AMBA on- chip verification the parcels to be vindicated are i) System connectivity during read and write cycle, ii) sale routing and iii) eventually the Data integrity. In the system connectivity checking test, the system integration is covered by the corresponding acknowledgement signals. Also, to check that creator generating the quantum of sale is going successful in the frame to match the data of read and write operation performed in the testbench itself. Parallelly it should run all the test cases which are to be enhanced during the run- phase. Also, during the routing stage, the machine data residency structure is vindicated by the machine application and effectiveness. Eventually, the data integrity is noted for every sale and the functionality check pass is carried out for each of the stages. In this on- chip protocol, the verifications plan focuses on the following scripts as Single read. Single write. Write & read. Multiple reads. Multiple writes. For the corresponding scripts the test- cases are generated, and the simulation is carried out for the functionality verification. Using the following FSM illustration also for understanding whether what all operation are to be done are satisfying the condition which are applied in the rendering pre, and post verification can be done fluently with the help of that illustration which matriculate all the testcases scripts which are enlisted as above. However, also the test- cases are modified consequently to reach the specific property checks If the machine isn't vindicated directly in the corresponding test- cases. These testcases are absolute and vindicated with colorful testing criteria using several other tools to confirm the correctness and scalable outgrowth within the time.

6. RESULTS & DISCUSSIONS

This section discusses the simulation results of the AMBA on-chip protocol for the five different test-scenarios and the performance assessment using VNC & EDA also Cent-OS. For carrying out the simulation analysis, first the individual modules on-chip Master/Slave are modeled using System Verilog and integrated with the modeled test- environment for the verification. In the verification environment, the DUV is integrated with generator, interface,driver and monitor to assess the performance in constrained mode. After the integration of the top module with DUV the simulation is carried out using Synopsys. Here the test-case is written for five different scenarios as i) Single Read, ii) Single Write, iii) Read & Write, iv) Multiple Read and Multiple Write, v) Slave and decode error checks etc.

Verification of Single Read

In a single read sale, the master sends a read address and control signals to the target, indicating that it wants to read data from a specific memory position. The target receives the read address and responds with the requested data. The data is also transferred back to the master over the data machine. The ON- CHIP protocol supports burst transfers, which allow for multiple data transfers to do without the need for fresh address signals. In a single read sale, the master can request a burst of one, meaning it only requests one data transfer. During a read sale, the master also sends signals to indicate the type of sale (normal, exclusive, or locked), the size of the data being read(8, 16, 32, or 64 bits), and any cache and memory attributes associated with the sale. Once the target receives the read address and control signals, it returns the requested data to the master over the data machine. The data transfer can be done in a single timepiece cycle or multiple timepiece cycles, depending on the speed of the system and the size of the data being transferred. Once the data transfer is complete, the target asserts a signal to indicate that the sale has finished. In this phase the read data deals of the AMBA on- chip protocol is vindicated with the Master/ Slave configuration mode. During the ready cycle deals, the signals ARVALID, ARREADY, RVALID, RREADY, RLAST, RDATA and ARSIZE are covered and vindicated. Then the functionality check primarily involves two modes of operation videlicet i) Read Address and ii) Read Response channel. For every positive edge of timepiece, the read address channel will cost the address at the high state values of ARVALID and ARREADY. Also, after a time detention, the response will be expressed to high mode with the high values of the signal on RVALID and RREADY. The last sale of the read operation is indicated by the high values of RLAST. The values of ARSIZE and ARLEN are measured and indicate the number of deals that passed in read mode. For vindicating the read phase, a test- case is generated and dissembled to test the functionality of the read operation. The working of the read mode includes two channels as explained over with the measure of the quality criteria

for the read address, read channel data with channel response. The entire operation of the read phase is pictorially represented as a waveform in fig. 4.

Figure 4: Read Phase Simulation Response

Verification of Single Write

In a single write transaction, the master sends a write address, control signals, and data to the target, indicating that it wants to write data to a specific memory location. The target receives the write address and data and writes the data to the specified memory location. The ON-CHIP protocol supports burst transfers, which allow for multiple data transfers to occur without the need for additional address signals. In a single write transaction, the master can request a burst of one, meaning it only requests one data transfer. During a write transaction, the master also sends signals to indicate the type of transaction (normal, exclusive, or locked), the size of the data being written (8, 16, 32, or 64 bits), and any cache and memory attributes associated with the transaction. Once the target receives the write address and data, it writes the data to the specified memory location. The data transfer can occur in a single clock cycle or multiple clock cycles, depending on the speed of the system and the size of the data being transferred. Once the data transfer is complete, the target asserts a signal to indicate that the transaction has finished. It is important to note that the ON-CHIP protocol supports a write-data response feature, which allows the target to send a response to the master indicating the status of the write operation (e.g., whether it was successful or not). This response is sent on a separate channel from the data bus and allows the master to verify that the write operation was successful. The Write Phase of the AMBA ON-CHIP protocol involves the verification of AWID, AWADDR, AWLEN, AWSIZE, and AWVALID AND AWREADY signals in Master/Slave configuration. The primary operation of the Write Phase involves three major operations as i) Write Address Channel, ii) Write Channel Data and iii) Finally Write response Channel. For every positive clock cycle, for fetching the high logic state the valid write address with an acknowledgement AWID the signals WVALID and WREADY take place. In the same manner, the write response operation will happen with the high transitions of the BVALID and BREADY. The control signal AWLEN holds the values 0000 in the starting and gets incremented for every transaction and the signal AWSIZE indicates the size of each transaction. The Write Transactions is pictorially represented in fig.5.

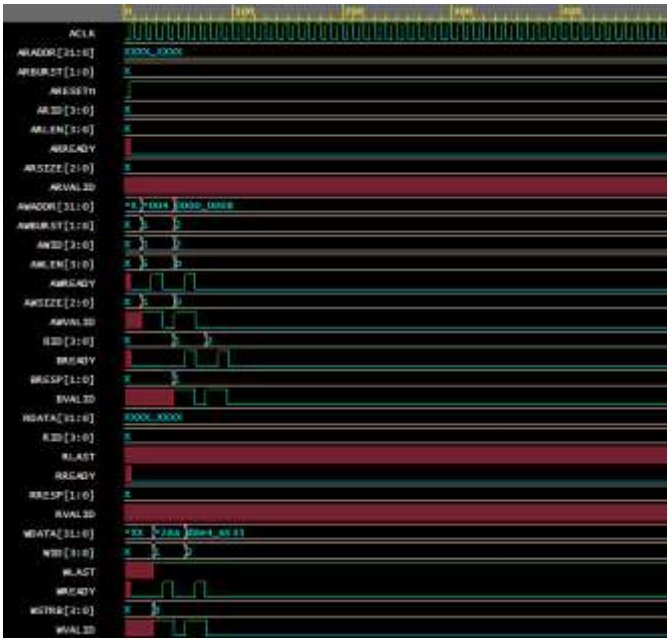

Figure 5: Write Phase Simulation Response

Verification of Read & Write

Both single read and write deals follow a similar protocol, with some differences in the signals transferred and entered. In a single read trade, the master sends a read address and control signals to the target, indicating that it wants to read data from a specific memory position. The target receives the read address and responds with the requested data. The data is also transferred back to the master over the data machine. During the trade, the master also sends signals to indicate the type of trade, the size of the data being read, and any cache and memory attributes associated with the trade. In a single write trade, the master sends a write address, control signals, and data to the target, indicating that it wants to write data to a specific memory position. The target receives the write address and data and writes the data to the specified memory position. During the trade, the master also sends signals to indicate the type of trade, the size of the data being written, and any cache and memory attributes associated with the trade. In both read and write deals, the ON- CHIP protocol supports burst transfers, which allow for multiple data transfers to do without the need for fresh address signals. In a single trade, the master can request a burst of one, meaning it only requests one data transfer. Once the target receives the applicable signals for either a read or write trade, it performs the operation and sends a response to the master indicating the status of the trade. This response includes a signal indicating whether the trade was successful or not. It's important to note that the ON- CHIP protocol supports various types of deals, including normal, exclusive, and locked deals. These different trade types are used to specify how the target should respond to different types of requests and allow for effective data transfer between different factors within the SoC. In the Read and Write phase from the same address position both the read and write deals do in the same address locales. The address having all the individual of the write and read operation that is

to be given by the signal AWADDR/ ARADDR with a size of 32- bit length in Hexadecimal format. The signal AWADDR indicates that the write address is the same for the read operation. The value of ARADDR is going to be same as the AWADDR (i.e., the read and write operation are parallelly passing in the same address position). The general value for the read data is used to get stored in the signal RDATA, whereas the value for the write operation is to get stored WDATA. With this effective monitoring of the signals in the test- case the read and write operation in the same address position is vindicated successfully for the AMBA ON- CHIP protocol. The entire verification of the Read & Write phase operation in the same address position is shown in fig. 6..

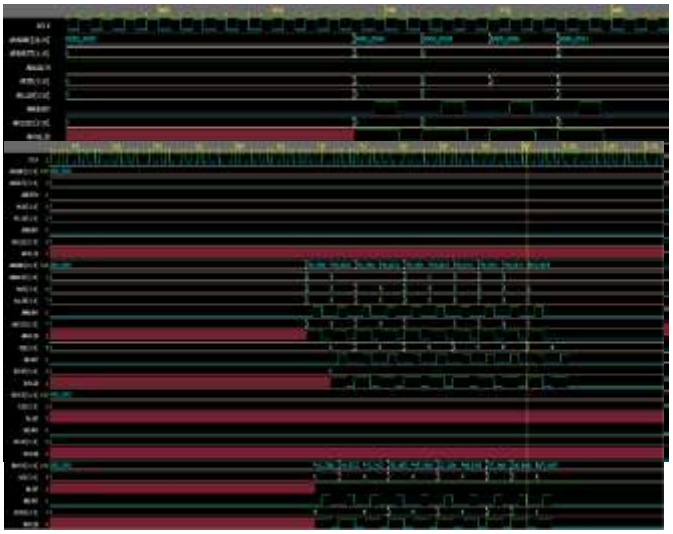

Figure 6: Read & Write

Verification of Multiple Read

The simulation waveform for the multiple reads at different locations is shown in fig.7.

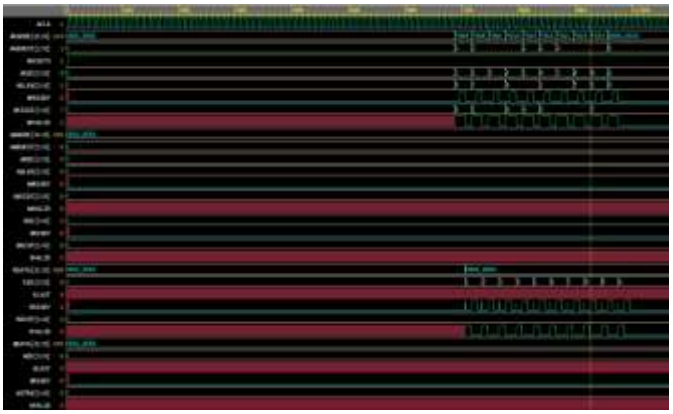

Figure 7: Multiple Read

The AMBA on-chip protocol supports multiple outstanding transactions, which means that multiple read transactions can be in progress at the same time. The following is a general description of an accurate ON-CHIP multiple read transaction: The master device initiates multiple read requests to the slave device, specifying the addresses of the data to be read and any additional control information. The slave device receives the read requests and starts processing them. The slave device checks each address and sends a read data response and the requested data for each read request. The master device receives the read data responses and the data. The master device then checks each response for errors and may perform additional actions, such as updating registers or initiating another transaction. Once all the read transactions are complete, the master device releases the ON-CHIP bus, allowing other transactions to take place. It's important to note that ON-CHIP also supports burst transfers, which can improve overall data transfer efficiency by reducing the number of individual transactions required. Burst transfers enable the master device to request a block of data in a single transaction, rather than issuing individual read requests for each data element. The slave device can then send the entire block of data in a single read data response, improving overall data throughput.

Verification of Multiple Write

The simulation waveform for the multiple writes at different locations is shown in fig.8.

Figure 8: Multiple Write

The AMBA on-chip protocol supports multiple outstanding transactions, which means that multiple write transactions can be in progress at the same time. The following is a general description of an accurate ON-CHIP multiple write transaction:

The master device initiates multiple write requests to the slave device, specifying the addresses where the data is to be written, the data itself, and any additional control information. The slave device receives the write requests and starts processing them. The slave device checks each address and data and acknowledges each write request. The master device receives the acknowledgements and may continue with other transactions. The slave device processes each write data and stores it at the specified address. Once all the write transactions are complete, the slave device sends a write response to the master device indicating that the data has been written. The master device receives the write responses and checks them for errors. Once all the write responses have been received and validated, the master device releases the ON-CHIP bus, allowing other transactions to take place. It's important to note that ON-CHIP also supports burst transfers, which can improve overall data transfer efficiency by reducing the number of individual transactions required. Burst transfers enable the master device to write a block of data in a single transaction, rather than issuing individual write requests for each data element. The slave device can then receive the entire block of data in a single transaction, improving overall data throughput.

7. ON CHIP OVERALL TEST FLOW USING FSM:

This protocol has a test flow which matches the internal hierarchical path, or we can say the design path which has the

8. CONCLUSION

In this research article, AMBA-ON-CHIP bus protocol is designed and implemented using System Verilog HDL. Then

actual and the expected scenario. Indeed, the testbench architecture is designed in such a way that it will be following the TOP to DOWN approach for generation of the testbench architecture.

To enlist the design path to assume that the flow of the state machine goes in a smoother way which matches the resultant by using case statements and other required equipment's for the easy and scalable flow before the execution.

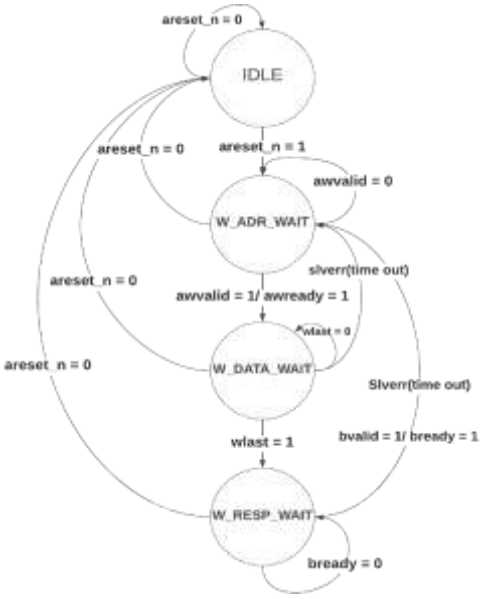

Figure 9: State Diagram for read operation.

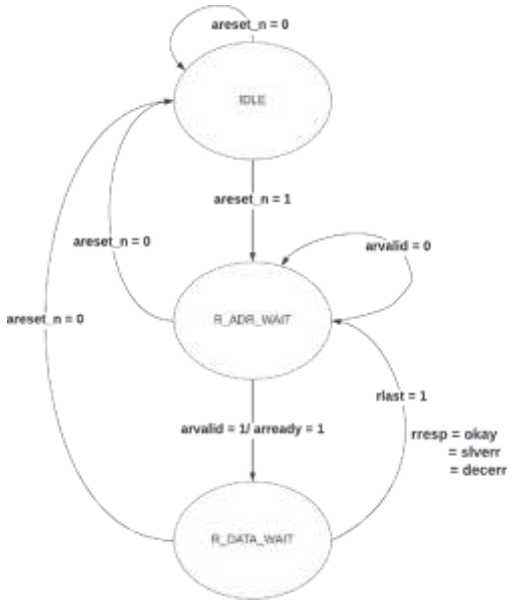

Figure 10: State Diagram for write operation.

the simulation is carried out for the functionality check using Synopsis EDA tool. Here the bus is verified for the five different test-case scenarios.

9. REFERENCES

[1]. K. Swaminathan, G. Lakshmi narayanan a, Seok- Bum Ko "Design and verification of an efficient WISHBONE-based network interface for network on chip". Computers and Electrical Engineering 40 (2014) 1838–1857

[2]. S.Saponara a, L. Fanucci a, M. Coppola "Design and coverage-driven verification of a novel network- interface IP macro cell for network-on- chip interconnects". Microprocessors and Microsystems 35 (2011) 579–592

[3]. Alan P. Su, Jiff Kuo, Kuen-Jong Lee, Ing-Jer Huang, Guo-An Jian" A Multi-core Software/Hardware Co- debug Platform with ARM CoreSightTM, On-chip Test Architecture and ON-CHIP/AHB Bus Monitor".

[4]. Attia B, Zitouni A, Tourki R. Design and implementation of network interface compatible OCP for packet based NOC. In: International conference on design & technology of integrated systems in nanoscale era; 2010. p. 1–8.

[5]. Technical Reference Manual of Prime Cell ON-CHIP Configurable Interconnect (PL300), ARM,Cambridge, U.K., 2010.

[6]. Singh Sanjay Pratap, Bhoj Shilpa, Balasubramanian Dheera, Nagda Tanvi, Bhatia Dinesh, Balsara Poras. "Generic network interfaces for plug and play NoC based architecture". In: Lecture notes in computer science Springerreconfigurable computing: architectures and applications; 2006. p. 287–98.

[7]. Chien-Hung Chen, Jiun-Cheng Ju, and Ing-Jer Huang, "A Synthesizable ON-CHIP Protocol Checker for SoC Integration", IEEE transl, ISOCC, Vol 8,pp.103- 106, 2010

[8]. Lai Yong-Long, Yang Shyue-Wen, Sheu Ming- Hwa, Hwang Yin- Tsung, Tang Hui-Yu, Huang Pin-Zhang. A high-speed network interfacedesign for packet- based NoC. In: IEEE ICCCAS;2006. p.2667–71.

[9]. J. Shao and B. T. Davis, "A burst scheduling access reordering mechanism, "in Proc. IEEE 13th Int. Symp. High Perform. Comput. Archit.,Feb. 2007, pp. 285–294.

[10]. Fattah M, Manian A, Rahimi A, Mohammadi S. "A high throughput low power FIFO used for GAL Chip",International conference on Computational Intelligence and communication networks,2010.

[12]. H.W wang,C.S lai,Hwang,and Y-H Lin,"on chip interconnection design and soc integration with open core protocol(ocp),"in proc.IEEE int.symp.VLSI Design 2008,pp. 25-28

[13]. X.Xiao and Lee,"A true parallel deadlock detection algorithm for single unit resource system and its hardware implementation,"IEEE trans.parallel Distrib.syst.,vol. 21,pp 4-19, jan 2010.

[14]. A.T.Tran and B.M.Bass,"Roshaq:High performance onchip router with shared queues ,"in proc.IEEE 29th int.conf.comput.design,oct,2011,pp.232-238.