IJAREEIE

e-ISSN: 2278-8875
p-ISSN: 2320-3765

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

Volume 12, Issue 3, March 2023

INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.317

9940 572 462 ☎ 6381 907 438 ✉ ijareeie@gmail.com @ www.ijareeie.com

Testing AXI Protocol by Using UVM_Methodology

NIHARIKA SAHU, **CHANDRAHAS SAHU**

M. Tech (VLSI), Dept. of ET&T, SSTC, Bhilai, India

Asst. Professor, Dept. of ETC, SSTC, Bhilai, India

ABSTRACT: The design of System-on-a-Chip (SoC) devices has grown increasingly sophisticated. due to the integration of many functional components or IPs (Intellectual Property) into a chip. How to validate on-chip communication attributes is the integration difficulty. Even despite conventional They still lack a chip-level dynamic, despite simulation-based on-chip bus protocol evaluating bus signals to comply bus transaction behavior or not. verified to help with hardware troubleshooting. A rule-based synthesizable AMBA AXI protocol was put out by us. There are 44 rules to be checked in the AXI protocol. System on a Chip With the development of a standardized signal bus architecture used for connecting various system modules, chip design became a key integrated approach for cutting the design time of the complete system. Given that massive SoCs cannot be certified using conventional, antiquated methods, one of the main obstacles at hand is how to examine these on-chip bus protocols. The AMBA AXI protocol, which displays write, read, and write read operations, has been proven in this work using UVM.

KEYWORDS: AXI Protocol, UVM_METHODOLOGY, System Verilog, AMBA Family, Verification, VLSI.

I. INTRODUCTION

As a result of the integration of IP cores into today's SOCs that have been integrated into them, synchronizing all IP cores during the transmission of data is a very difficult task. Today's semiconductors use mainly communication protocols that are part of the AMBA family, namely the Advanced Peripheral Bus (APB), Advanced High-performance Bus (AHB) and ,Advanced Extensible Interface (AXI).As a result of the little time available in the design stage, verification becomes a problematic task when it comes to the production of these kinds of SOCs, since it is commonly estimated that it will take up 70 percent of the entire project time as opposed to 30 percent in the design phase. Thus, using a specialized testing environment called Verification-IP. One way to categorize modern System-on-Chip (SoC) communication bus protocols is based on their power consumption and performance efficiency. These protocols can be grouped into different categories depending on how effectively they function and how much power they consume. The specific group of communication bus protocols used in a given system is chosen based on the system's requirements for power consumption, performance, and functionality. While it performs badly when compared to the AXI, the APB bus protocol structure is the easiest to implement and the least power-hungry of the three AMBA protocols. Compared to AHB and APB protocols, the AMBA-AXI protocol considerably uses modest power while performing better. Consequently, if high performance is required, the AXI bus protocol structure has been adopted for the SOC designs. AHB has a shared-bus design, hence arbitration mechanisms only permit one master to utilize the bus at a time. In contrast, AXI uses an interconnect-based bus, which is far more effective since it enables simultaneous communication between (many) masters and (multiple) slaves. AXI is significantly more efficient since it enables concurrent communication between several masters and numerous slaves because it is essentially an interconnect-based bus. In this work, we explore the design of an AMBA AXI protocol verification built on System Verilog.

The following is a list of AMBA AXI features:

•Burst write and read transactions enable the transfer of a specified number of bytes per transfer in a single cycle, improving data transfer rates and performance. This is particularly useful in applications that require high-speed data transfers, such as video streaming and high-performance computing.

•Support out-of-order transactions: Understanding in-order transactions is a prerequisite for understanding out-of-order transactions.

•An in-order transaction is one that is submitted in the same order that we get responses when the request and response take place in the same order.

•Once an order-transaction is submitted and a request-response is just received, the two events do not occur in a sequential sequence.

• Support for simultaneous write and read transactions.

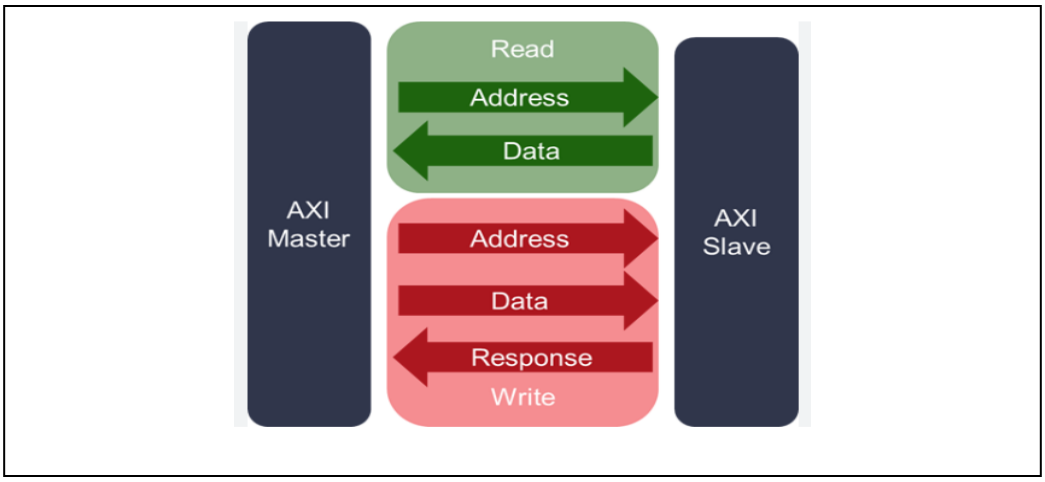
Signals in axi protocol:

Table 1 AXI SIGNALS

GLOBAL SIGNALS	**WRITE ADDRESS CHANNEL SIGNALS (AW)**	**WRITE DATA CHANNEL SIGNALS (W)**	**WRITE RESPONSE CHANNEL SIGNALS (B)**	**READ ADDRESS CHANNEL SIGNALS (AR)**	**READ DATA CHANNEL (R)**
ACLK	AWID [3:0]	WID [3:0]	BID [3:0]	ARID [3:0]	RID{3:0}
ARESETn	AWADDR [31:0]	WDATA [31:0]	BRESP [1:0]	ARADDE [31:0]	RDATA{31:0}
	AWLEN [3:0]	WSTRB [3:0]	BVALID	ARLEN [3:0]	RRESP{1:0}
	AWSIZE [2:0]	WLAST	BREADY	ARSIZE [2:0]	RLAST
	AWBRUST [1:0]	WVALID		ARBURST [1:0]	RVALID
	AWLOCK [1:0]	WREADY		ARLOCK [1:0]	RREADY
	AWCACHE [3:0]			ARCACHE [3:0]	
	AWPROT [2:0]			ARPROT [2:0]	
	AWVALID			ARVALID	
	AWREADY			ARREADY	

The following 5 transaction independent channels are included in the (AXI protocol):

Figure -1 CHANNELS IN AXI



1. Write address: The address and control information conveyed by this channel determines the type of data that will be transferred.
2. Write data: This Write data channel is used for transactions between (master-slave) masters and slaves.

3. Write response: The slave is using the channel to provide the master feedback. demonstrates that the master transfer has been completed.
4. Read address: The channel essentially contains control and address information for the read channel. This in turn determines the type of data being sent.
5. Read data: Through this channel, transactions are transmitted from the slave to the master.

Handshaking mechanism in axi:

Table 2 HANDSHAKING MECHANISM

AXI channels	Direction of handshaking signals

Write Address channel	AWVALID(master->slave)
	AWREADY(slave->master)

Write Data channel	RVALID(master->slave)
	RREADY(slave->master)

Write Response channel	BVALID(slave->master)
	BREADY(master->slave)

Read Address channel	ARVALID(master->slave)
	ARREADY(slave->master)

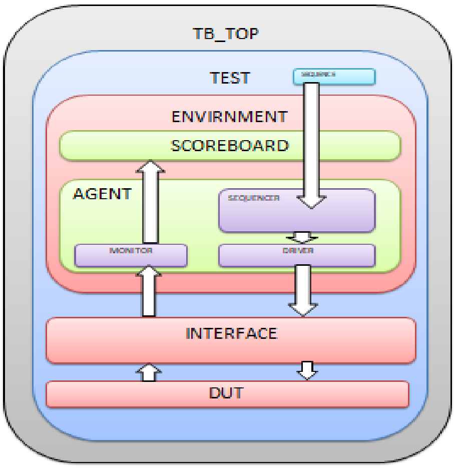
Read Data channel	RVALID(slave->master)
	RREADY(master->slave)

II. METHODOLOGY

Because of the growing industry's need for reliable systems with a shorter time to market, the verification process has become much more challenging.Verilog was once the only language used for design verification, but producers have switched to System Verilog due to its lack of modularity.in this paper we use UVM methodology because of its accessibility, effectiveness, and better level of design renewability.

UVM testbench:

Figure -2 UVM TESTBENCH



| e-ISSN: 2278 – 8875, p-ISSN: 2320 – 3765| www.ijareeie.com | Impact Factor: 8.317|

||Volume 12, Issue 3, March 2023||

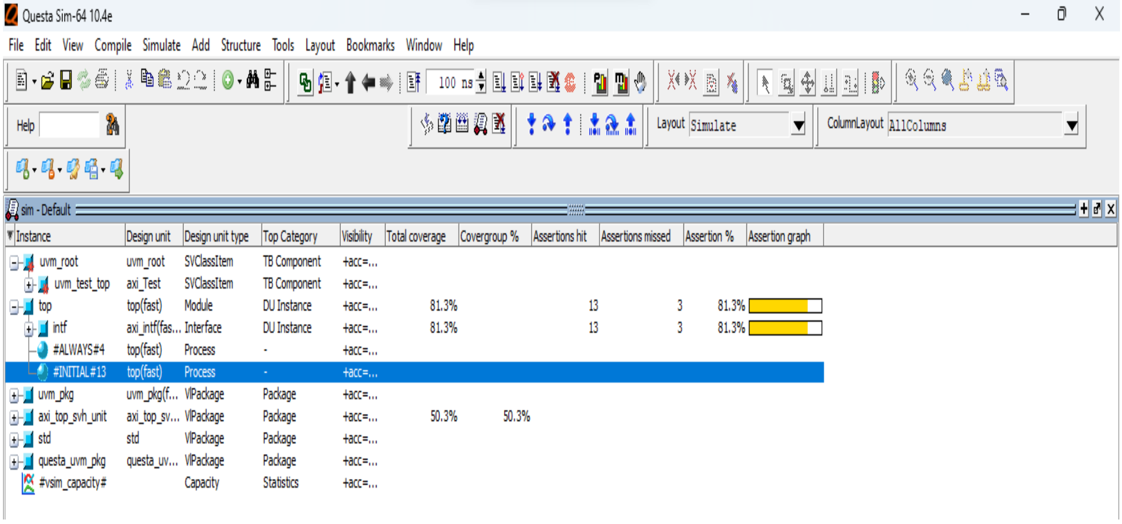
|DOI:10.15662/IJAREEIE.2022.1203004|

IV. RESULTS

The simulation is run in the QUESTASIM tool, and the Figure displays the AMBA AXI waveform. Signals on write address channels start with AW. W is the data channel's write signal. Signals on write response channels start with the letter B. AR denotes the Read Address Channel signal. R can be executed to evaluate the read data channel's signal.

COVERAGE

Figure -3 RESULT COVERAGE


Questa Sim-64 10.4e
File Edit View Compile Simulate Add Structure Tools Layout Bookmarks Window Help
Help
Layout Simulate
ColumnLayout AllColumns
sim - Default

Instance	Design unit	Design unit type	Top Category	Visibility	Total coverage	Covergroup %	Assertions hit	Assertions missed	Assertion %	Assertion graph
uvm_root	uvm_root	SVClassItem	TB Component	+acc=...						
uvm_test_top	axi_Test	SVClassItem	TB Component	+acc=...						
top	top(fast)	Module	DU Instance	+acc=...	81.3%		13	3	81.3%	
intf	axi_intf(fas...	Interface	DU Instance	+acc=...	81.3%		13	3	81.3%	
#ALWAYS#4	top(fast)	Process	-	+acc=...						
#INITIAL#13	top(fast)	Process	-	+acc=...						
uvm_pkg	uvm_pkg(f...	VlPackage	Package	+acc=...						
axi_top_svh_unit	axi_top_sv...	VlPackage	Package	+acc=...	50.3%	50.3%				
std	std	VlPackage	Package	+acc=...						
questa_uvm_pkg	questa_uv...	VlPackage	Package	+acc=...						
#vsim_capacity#		Capacity	Statistics	+acc=...						

WAVEFORM

Figure -4 RESULT WAVEFORM

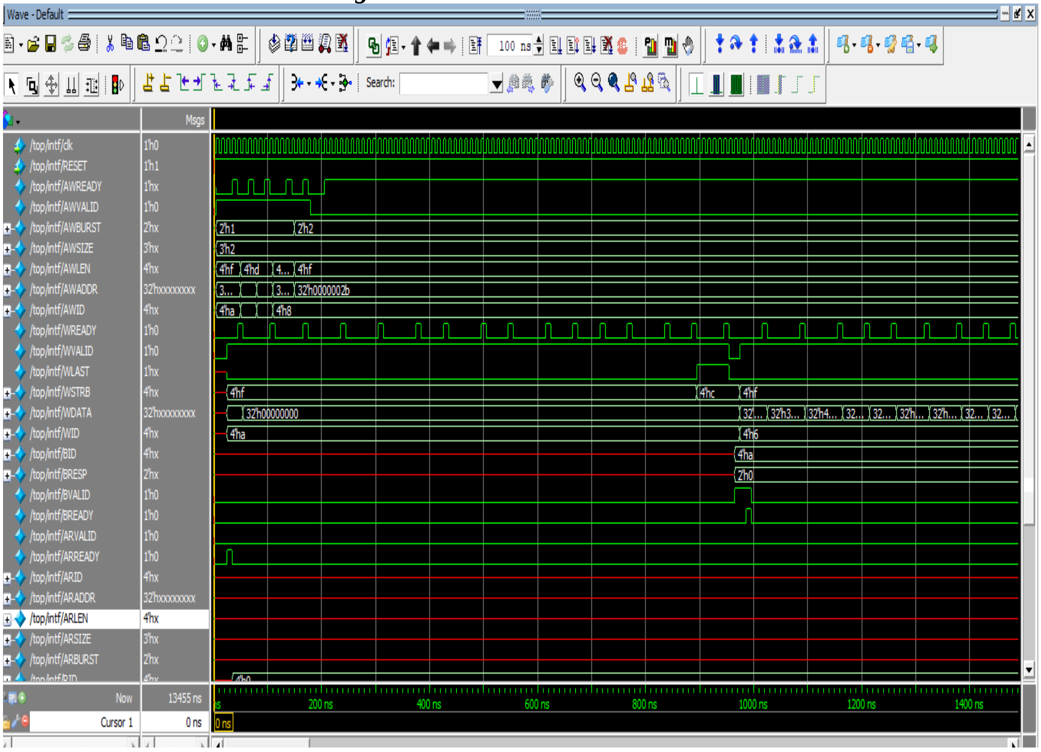
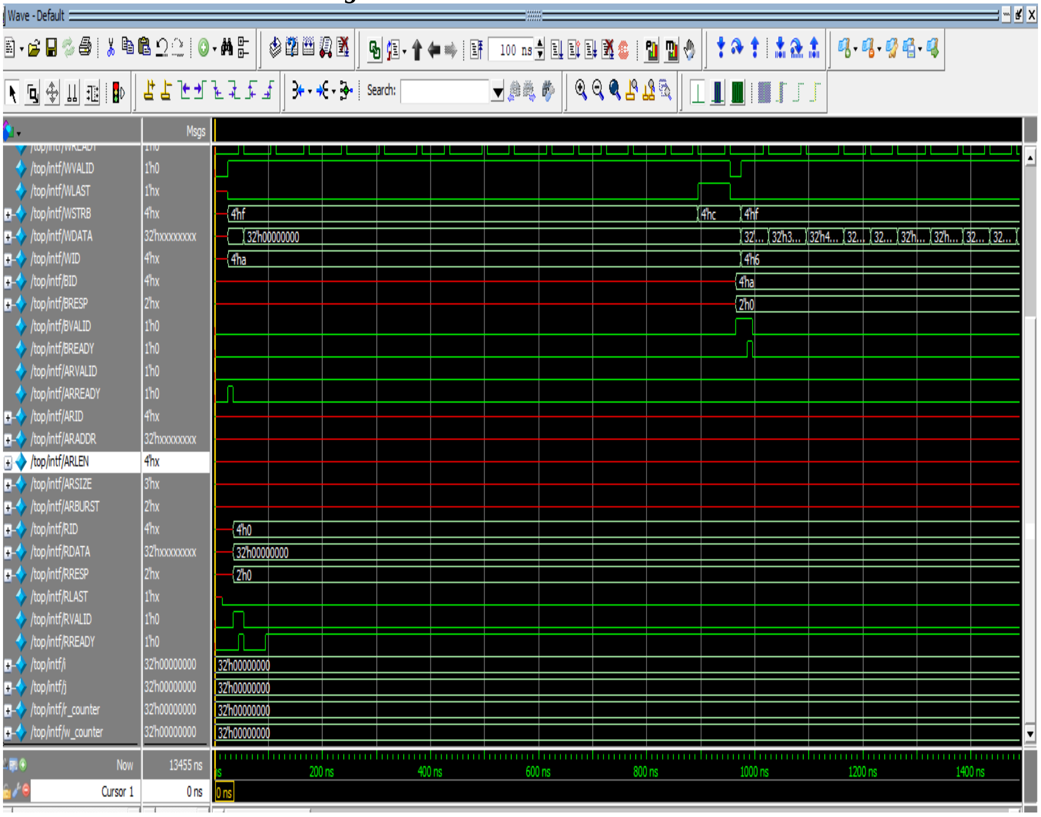
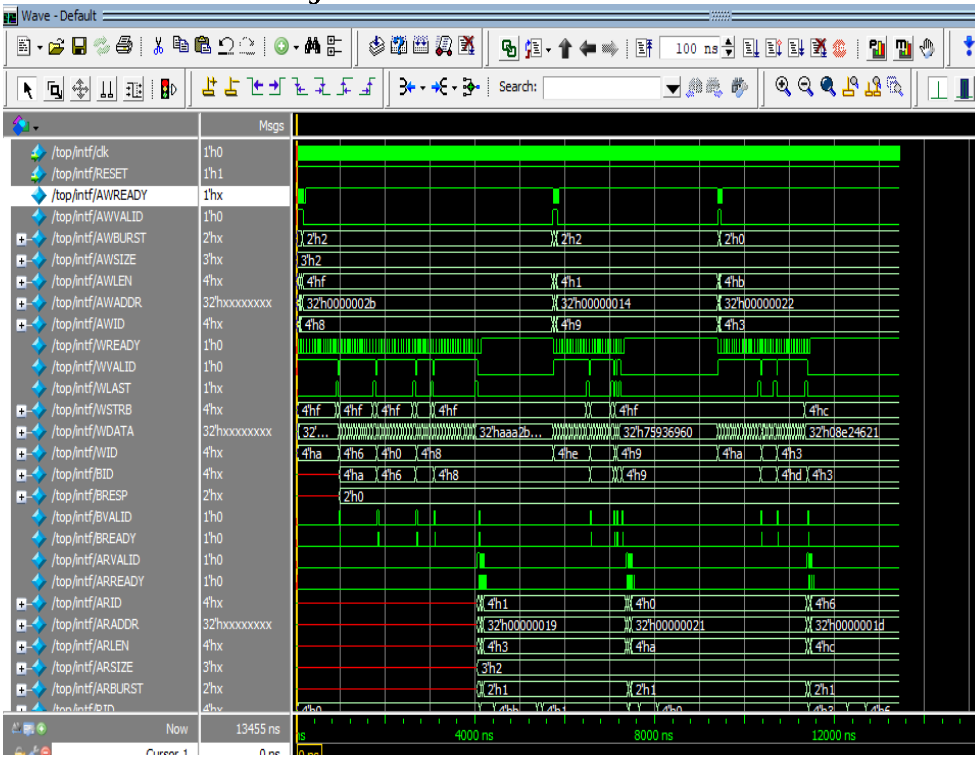


Figure -5 RESULT WAVEFORM 2



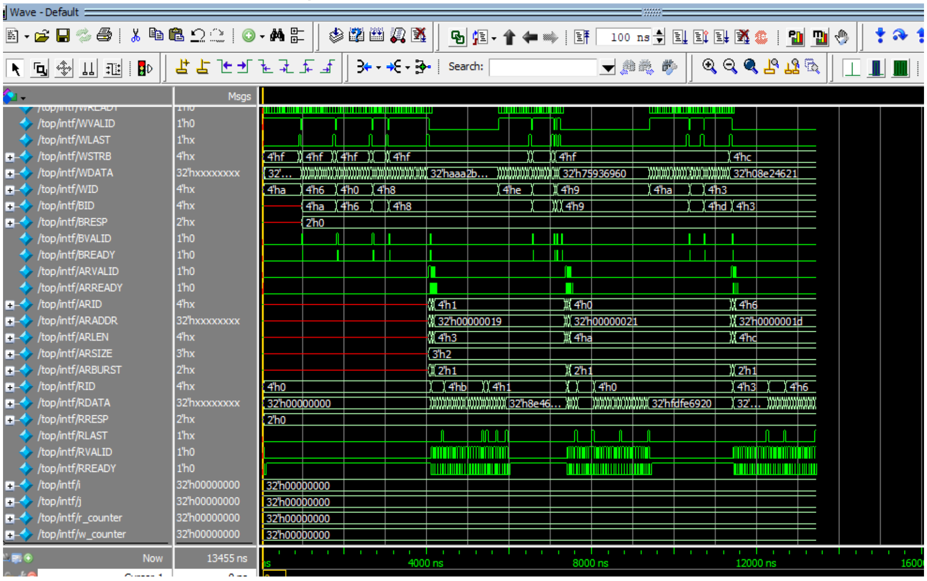
	Msgs
/top/intf/WVALID	1'h0
/top/intf/WLAST	1'hx
/top/intf/WSTRB	4'hx
/top/intf/WDATA	32'hxxxxxxxx
/top/intf/WID	4'hx
/top/intf/BID	4'hx
/top/intf/BRESP	2'hx
/top/intf/BVALID	1'h0
/top/intf/BREADY	1'h0
/top/intf/ARVALID	1'h0
/top/intf/ARREADY	1'h0
/top/intf/ARID	4'hx
/top/intf/ARADDR	32'hxxxxxxxx
/top/intf/ARLEN	4'hx
/top/intf/ARSIZE	3'hx
/top/intf/ARBURST	2'hx
/top/intf/RID	4'hx
/top/intf/RDATA	32'hxxxxxxxx
/top/intf/RRESP	2'hx
/top/intf/RLAST	1'hx
/top/intf/RVALID	1'h0
/top/intf/RREADY	1'h0
/top/intf/i	32'h00000000
/top/intf/j	32'h00000000
/top/intf/r_counter	32'h00000000
/top/intf/w_counter	32'h00000000
Now	13455 ns
Cursor 1	0 ns

Figure -6 RESULT WAVEFORM 3



	Msgs
/top/intf/clk	1'h0
/top/intf/RESET	1'h1
/top/intf/AWREADY	1'hx
/top/intf/AWVALID	1'h0
/top/intf/AWBURST	2'hx
/top/intf/AWSIZE	3'hx
/top/intf/AWLEN	4'hx
/top/intf/AWADDR	32'hxxxxxxxx
/top/intf/AWID	4'hx
/top/intf/WREADY	1'h0
/top/intf/WVALID	1'h0
/top/intf/WLAST	1'hx
/top/intf/WSTRB	4'hx
/top/intf/WDATA	32'hxxxxxxxx
/top/intf/WID	4'hx
/top/intf/BID	4'hx
/top/intf/BRESP	2'hx
/top/intf/BVALID	1'h0
/top/intf/BREADY	1'h0
/top/intf/ARVALID	1'h0
/top/intf/ARREADY	1'h0
/top/intf/ARID	4'hx
/top/intf/ARADDR	32'hxxxxxxxx
/top/intf/ARLEN	4'hx
/top/intf/ARSIZE	3'hx
/top/intf/ARBURST	2'hx
Now	13455 ns

Figure -7 RESULT WAVEFORM 4



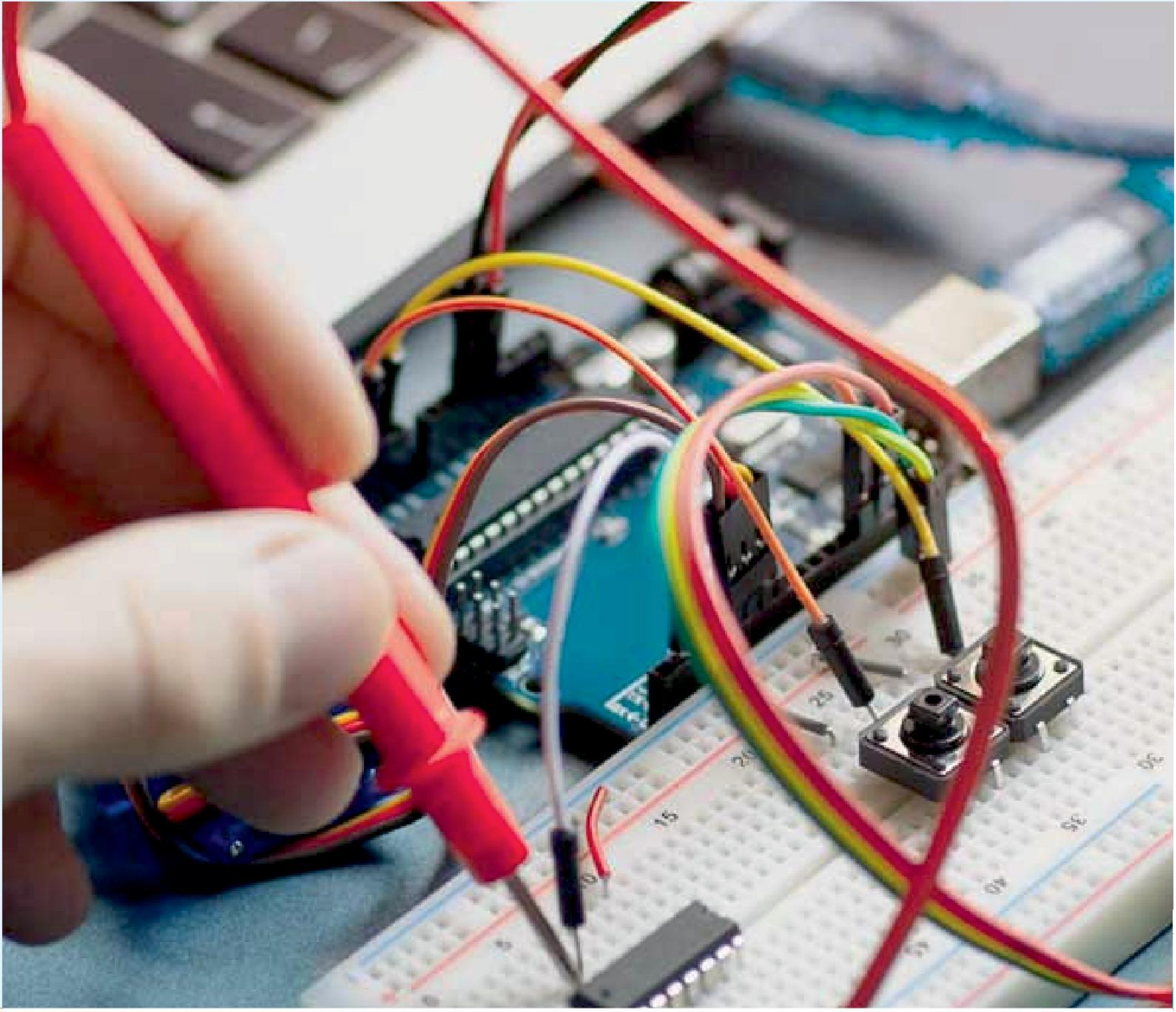
	Msgs
/top/intf/WVALID	1'h0
/top/intf/WLAST	1'hx
/top/intf/WSTRB	4'hx
/top/intf/WDATA	32'hxxxxxxxx
/top/intf/WID	4'hx
/top/intf/BID	4'hx
/top/intf/BRESP	2'hx
/top/intf/BVALID	1'h0
/top/intf/BREADY	1'h0
/top/intf/ARVALID	1'h0
/top/intf/ARREADY	1'h0
/top/intf/ARID	4'hx
/top/intf/ARADDR	32'hxxxxxxxx
/top/intf/ARLEN	4'hx
/top/intf/ARSIZE	3'hx
/top/intf/ARBURST	2'hx
/top/intf/RID	4'hx
/top/intf/RDATA	32'hxxxxxxxx
/top/intf/RRESP	2'hx
/top/intf/RLAST	1'hx
/top/intf/RVALID	1'h0
/top/intf/RREADY	1'h0
/top/intf/i	32'h00000000
/top/intf/j	32'h00000000
/top/intf/r_counter	32'h00000000
/top/intf/w_counter	32'h00000000
Now	13455 ns

V. CONCLUSION

Therefore, the AMBA AXI protocol was checked using UVM, and got simulation results has been seen in fig. This main idea is to add coverage in AXI protocol happening.

REFERENCES

1. "Development of VIP for AMBA AXI-4.0 Protocol," by Renduchinthala H. H. S. S. Prasad and ChSanthi Rani 10.17485/ijst/2016/v9i48/101912 Indian Journal of Science and Technology, vol. 9, no. 48, 2016.
2. ARM, ”AMBA AXI™ Protocol Specification”.
3. Verification of AMBA AXI on-chip Communication Protocol by Nikhil Gaikwad and Vijay N Patil, 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), DOI: 10.1109/ICCUBEA.2018.8697587
4. ARM,”AMBA™Specification(Rev 2.0)”.
5. “Verification OfAmbaAxi Bus Protocol Implementing Incr And Wrap Burst Using System Verilog.” By a HarshaGarua. International Journal of Research in Engineering and Technology, vol. 05, no. 03, 2016, pp. 201–206. doi:10.15623/ijret.2016.0503041.
6. "A synthesizable AXI protocol checker for SoC integration" .By Chen, Chien-Hung, Jiun-Cheng Ju, and Ing-Jer Huang. In 2010 International SoC Design Conference, 2010.
7. "Validation of transactions in AXI protocol using system verilog".By M. PrasannaDeepu, R. Dhanabal. 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), 2017



INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.317

doi crossref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

International Journal of Advanced Research

in Electrical, Electronics and Instrumentation Engineering

☎ 9940 572 462 6381 907 438 ✉ ijareeie@gmail.com

Scan to save the contact details

www.ijareeie.com