iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VII **Month of publication:** July 2020

DOI: https://doi.org/10.22214/ijraset.2020.30399

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 VII July 2020- Available at www.ijraset.com

AXI2OCP Bridge Verification

Sindhu J[1], N. M. Mahesh Gowda[2]
[1]*M. Tech Student,* [2]*Assistant Professor, Department of Electronics and Communication, PES College of Engineering, Mandya, Karnataka, India*

Abstract: ***AXI2OCP Bridge is a component which converts the AXI signals to OCP signals. Verification is a process of checking the design that is implemented is working according to the requirements specified. The Advanced eXtensible Interface (AXI) protocol and Open Core Protocol are the industry standard on-chip communication protocol. AXI2OCP Bridge helps to establish an effective communication between two components; one communicating in AXI protocol format and the other communicating in OCP format.***

The AXI2OCP Bridge acts as a Design Under Test (DUT) module for the verification environment and it is implemented using verilog, which is a hardware description language.

The verification environment is implemented with the help of System Verilog (SV). Verifying helps in increasing the efficiency of the design. With the use of AXI2OCP Bridge, it is easier to connect the AXI based CPU cores to other intellectual property which are OCP based.

Keywords: AXI2OCP Bridge, Assertion Based Verification, System Verilog, cache transaction, protected transaction.

I. INTRODUCTION

Today's System-on-Chip (SoC) has much Intellectual Property (IP) cores which are inbuilt in them and the correct synchronization between all of these cores during the communication of data is a big task [1].The increasing complexity in modern SoC designs leads to more number of IP blocks to be integrated on to a chip. Bus protocols have much importance in the field of SoC design.

There are various types of standard protocols available and are used in SoC which requires a bridge to pass the information from one type of protocol to other type of protocol safely and without any data loss [2]. One such bridge is AXI2OCP Bridge, which converts Advanced eXtensible Interface (AXI) signals to Open Core Protocol (OCP) signals by mapping. AXI was introduced by ARM Holdings.

AXI protocol is a standard architecture designed for supporting the high frequency and higher performance SoC designs and incorporates the features such as independently acknowledged address and data channels, separate address/control and data phases, support for unaligned data transfers using byte strobes, out of order transaction, burst based transactions, separate read and write data channels to enable low- cost Direct Memory Access (DMA) ability to issue multiple outstanding addresses, easy addition of register stages to provide timing closure [3].

OCP was introduced by OCP International Partnership. The Open Core Protocol (OCP) defines the only non-proprietary, openly licensed, core centric protocol with high performance, bus independent interface between IP cores that reduces design time, design risk, and manufacturing costs and promote IP core reusability for SOC designs [4]. OCP being an existing standard, it is capable enough in meeting all the future SOC requirements in terms of performance, design methodology, including verification and modeling tools. OCP is utilized by many pipelined signal processing applications such as MPEG2 decoding and multiband DRAM architectures [5].

Features of OCP include point- to-point synchronous interface, pipelining and bus independence, separation between requests and responses, support of bursts through annotation of transfers with burst information, support for transmission of in-band information, out-of-order request and response delivery using multiple threads and tags. The benefit of System Verilog (SV) is that it allows the user to construct reliable, reiteration, verification environments in a regular syntax, cross functional projects. Now that SV incorporates Object Oriented Programming (OOP), inter-process communication and dynamic threads, it can be used for system design [6].

II. RELATED WORK

The aim specified in [2] is to design and verify various AXI2OCP Bridge features using coverage closure criteria. Here, the AXI2OCP Bridge is implemented using verilog hardware description language. AXI2OCP bridge protocol verification environment is constructed by using System Verilog assertion based functional coverage and code coverage. Creating verification environment of AXI2OCP Bridge and checking the response from the design is one of the verification criteria developed in this paper. This proposed implemented environment improves the coverage and it reduces the time taken for verification. The simulation result of AXI2OCP Bridge for different test cases like write, read and write-read transactions are observed. Realistic waveforms are created in Questa Sim tool by Mentor Graphics. The effective bus utilization of 94 percent for the read transaction and 84 percent for the write transaction is obtained.

The main focus of [7] is about measuring the bus utilization parameter for the AXI 3.0 protocols generated test cases and functional verification of the AXI2OCP Bridge using system verilog language. Mapping the different set of signals of AXI protocol and OCP with the help of AXI2OCP Bridge is explained and analyzed. AXI2OCP Bridge is implemented using verilog, verification environment is created using System Verilog. The implementation of the bridge, creating the verification environment and checking the response from this design is the verification criteria developed in [2]. A bus utilization of 76.92 percent for the read phase, 81.25 percent for the write phase and 95.45 percent for write_read phase is obtained. The functional verification of the AXI2OCP Bridge with practical waveforms is generated by the Mentor Graphics Questa Sim tool.

III. METHODOLOGY

The verification environment of the AXI2OCP Bridge is as shown in Fig. 1. The components of the verification environment includes AXI generator, AXI coverage, AXI Bus Functional Model (BFM), AXI Monitor, Mailbox, OCP monitor, AXI assertion, OCP assertion, AXI2OCP assertion, AXI2OCP reference model, checker and the DUT which is the AXI2OCP Bridge. Testbench is implemented as a hierarchy of classes is as shown in Fig. 2. Class helps in keeping code modular by creating different components (class) for different testbench requirements. This also helps in Code reusability by reusing only required components.

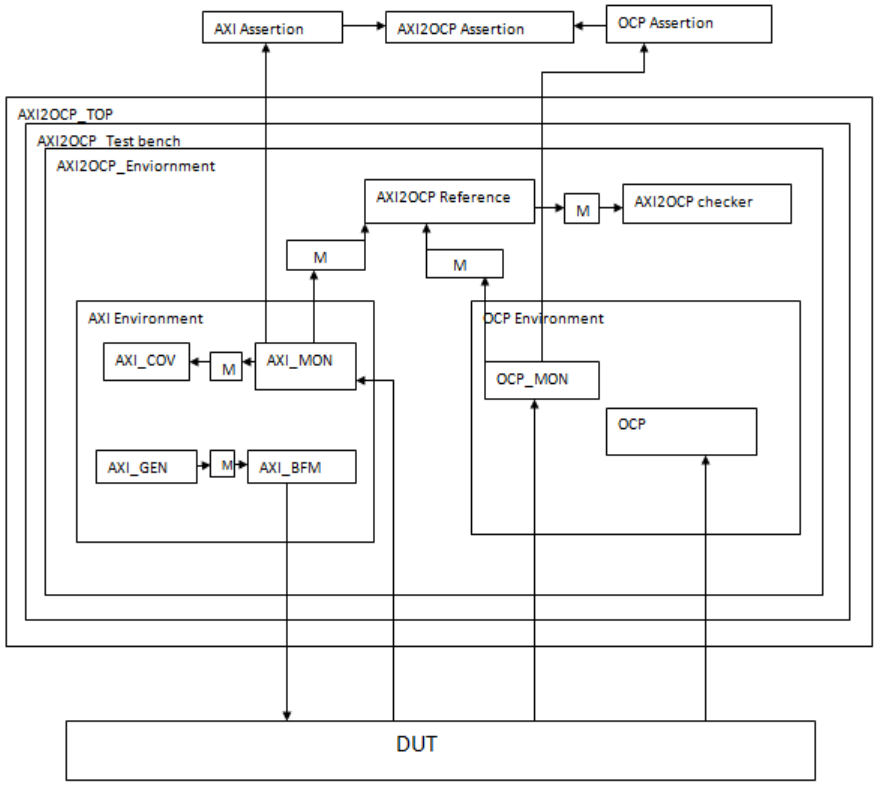


Fig. 1. AXI2OCP verification environment components

The generator generates test cases based on the specifications. BFM drives the data generated by the generator to the DUT. Mailbox allows different process to exchange data between them. Monitor is used to sample the interface signals and converts signal level to transaction level. The reference model reflects the behavior of the DUT. The checker model compares the results from DUT and Reference Model. The coverage module measures the functionality feature coverage, the line coverage and the conditional coverage of the design. Assertion model is used to check the occurrence of a specific condition or sequence of events, warnings or errors which are generated on the failure of specific condition or sequence of events. Thus Assertion Based Verification helps to identify the design bugs and helps to improve the coverage of the design.

The reference model and checker are used for self checking. The AXI2OCP Bridge is the Design Under Test (DUT) module for the verification environment, verilog hardware description language is used for its implementation. To verify the functional working of the bridge and to check its response, the System Verilog verification language is used. In the top module, all the modules of AXI2OCP Bridge environment are instantiated.

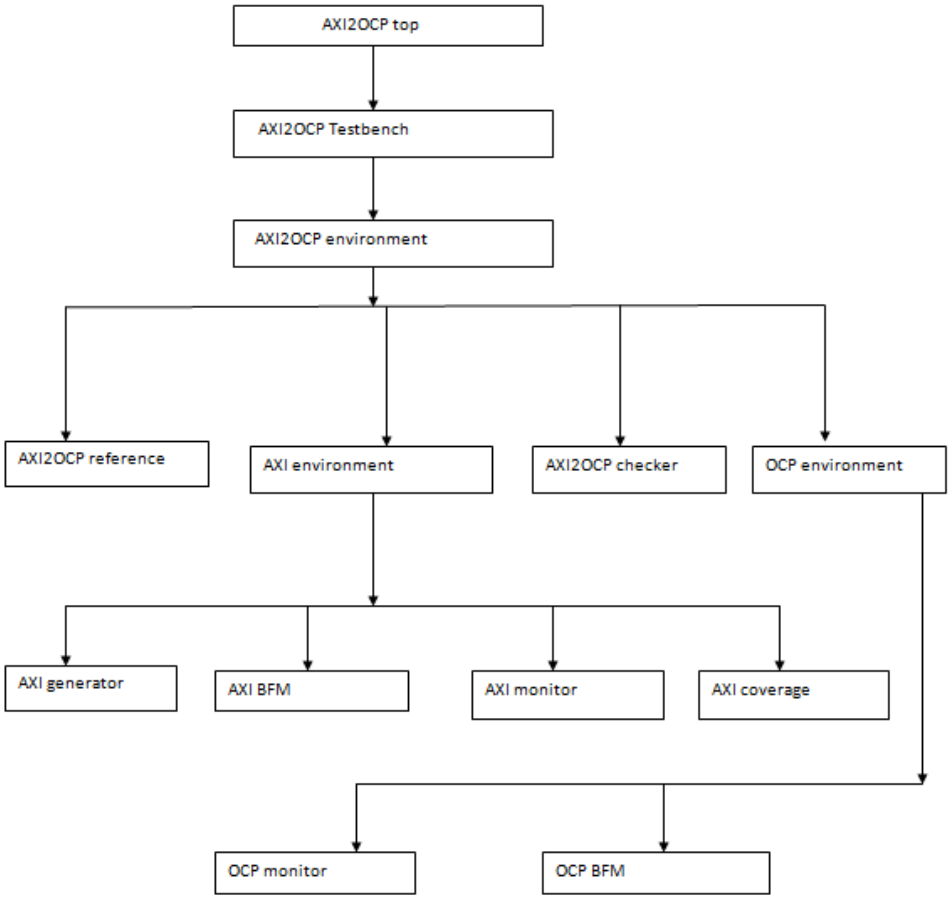


Fig. 2. AXI2OCP hierarchy using System Verilog

The AXI generator generates test cases such as single read transaction, single write transaction, burst transactions and unaligned transactions. These generated test cases are sent to the mailbox, which stores the data until it is sent to the BFM in order to avoid the loss of data. The BFM receives the data from the mailbox and drives it to the AXI2OCP Bridge. The received data is sent to AXI monitor who then sends the data to the AXI2OCP reference model and AXI coverage model via the mailbox and the coverage for the respective test case is obtained. After the mapping of AXI signals to OCP signals, the result is sent to the OCP monitor and then to the reference model via mailbox. Same input data is given to both DUT module and the reference model. Since the data given to DUT and reference model are same, the outputs from these two must match with one another. The comparison of the outputs is carried on by the checker module. If there is any mismatch in the output, it denotes that there is some functional error in the DUT.

IV. SIMULATION AND RESULT

All the simulations are carried out with the help of the tool Questa Sim 10.6c. Various test cases such as single read transaction, single write transaction, write transactions with varying burst length, checking for different burst types, one write transaction with aligned address, one write transaction with unaligned address. Similarly, different test cases are generated and the obtained coverage report for each test case is observed. Bus utilization is also calculated for every test case. Bus utilization is the percentage of number of valid count to number of busy count.Valid count is the actual number of clock cycles required for actual data transfer. Busy count is the number of clock cycles required for completing a transaction.

1) Testcase 1: Write transaction with varying AWLEN

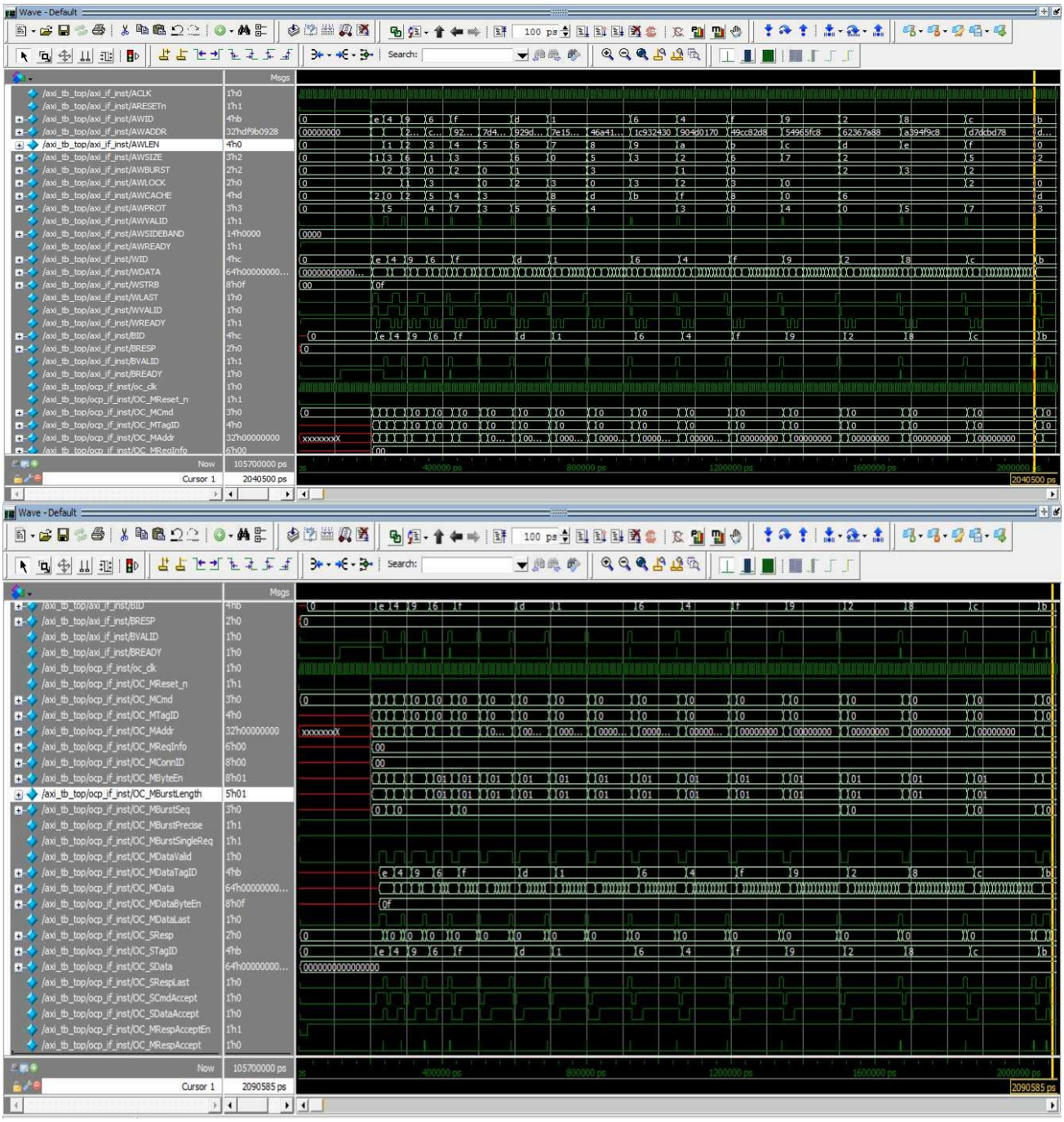
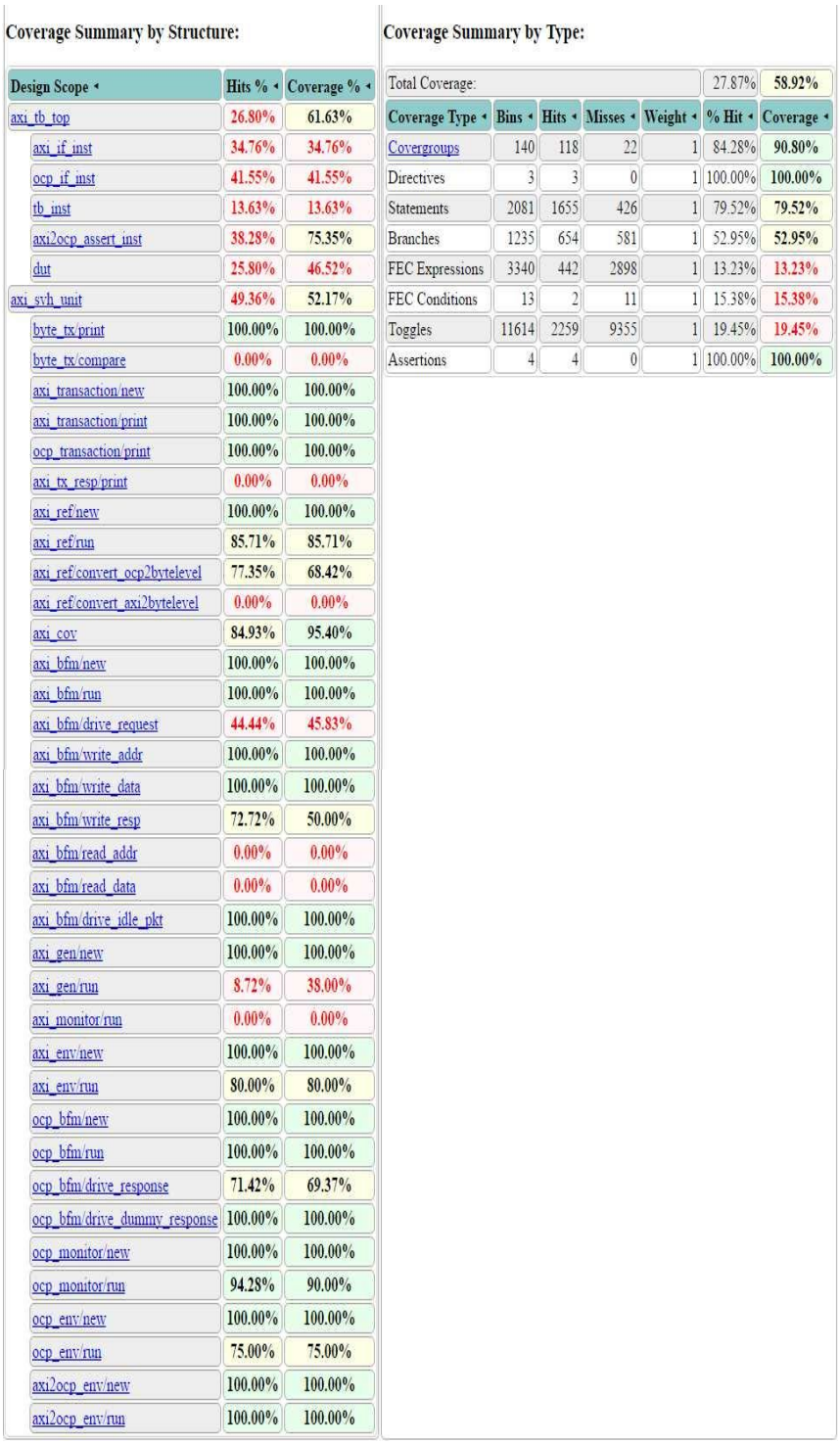


Fig. 3: Write transaction with varying AWLEN



Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	26.80%	61.63%
axi_if_inst	34.76%	34.76%
ocp_if_inst	41.55%	41.55%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	25.80%	46.52%
axi_svh_unit	49.36%	52.17%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	100.00%	100.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	77.35%	68.42%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	84.93%	95.40%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	44.44%	45.83%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	0.00%	0.00%
axi_bfm/read_data	0.00%	0.00%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	8.72%	38.00%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	71.42%	69.37%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					27.87%	58.92%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	118	22	1	84.28%	90.80%
Directives	3	3	0	1	100.00%	100.00%
Statements	2081	1655	426	1	79.52%	79.52%
Branches	1235	654	581	1	52.95%	52.95%
FEC Expressions	3340	442	2898	1	13.23%	13.23%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	2259	9355	1	19.45%	19.45%
Assertions	4	4	0	1	100.00%	100.00%

Fig. 4: Coverage report of write transaction with varying AWLEN

Testcase 1 is the verification of the write transaction. This write transaction is carried out for varying length of AWLEN. AWLEN is the signal indicating the burst length of the transaction. Here the burst length is varied from 0 to 15.The simulation waveform of this testcase is as shown in Fig. 3. The coverage report of the write transaction with varying AWLEN is given in Fig.4. Here we can see that there is 100 percent coverage of the assertions and 90.80 percent functional coverage.

2) Testcase 2: Testing different burst type

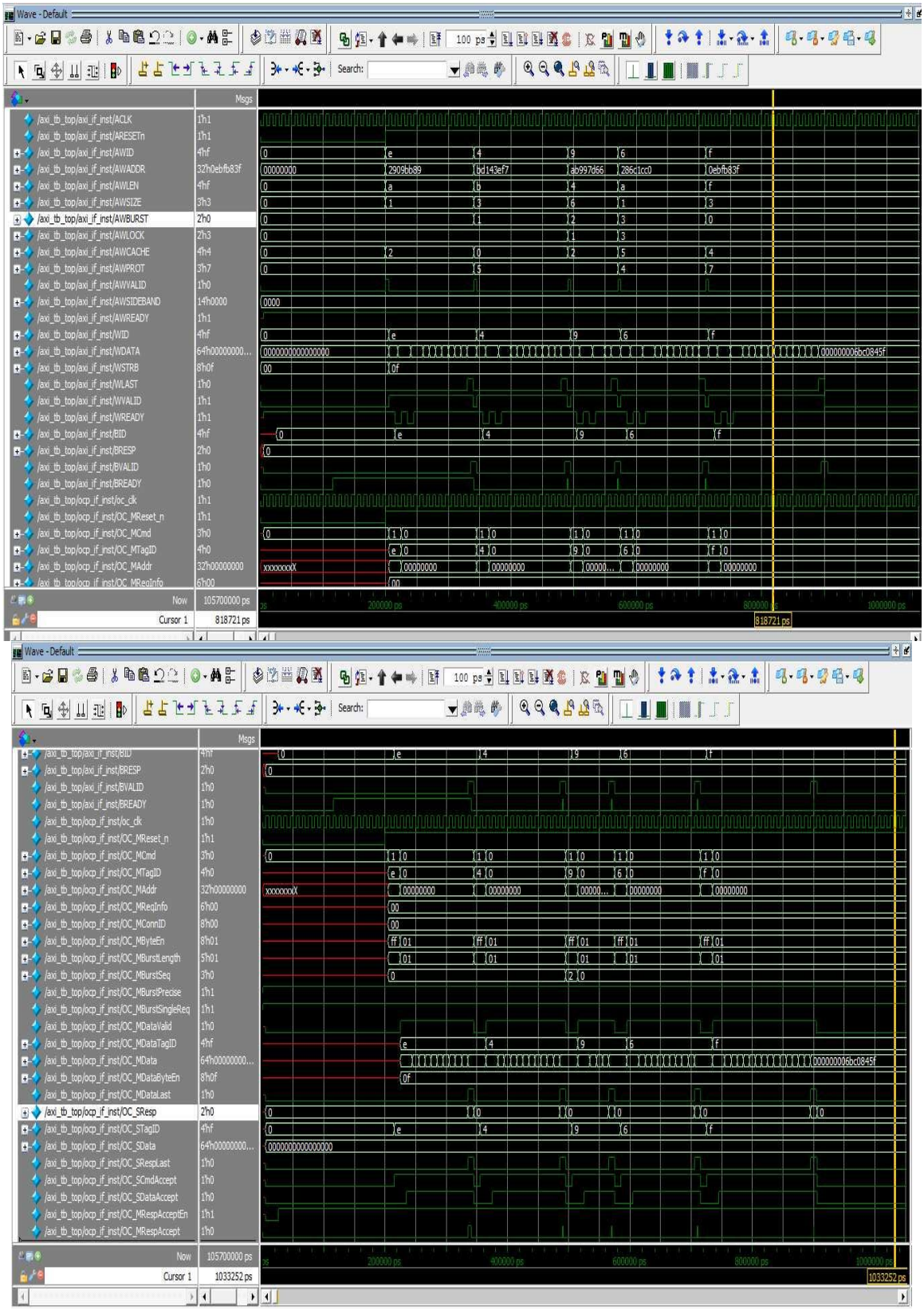
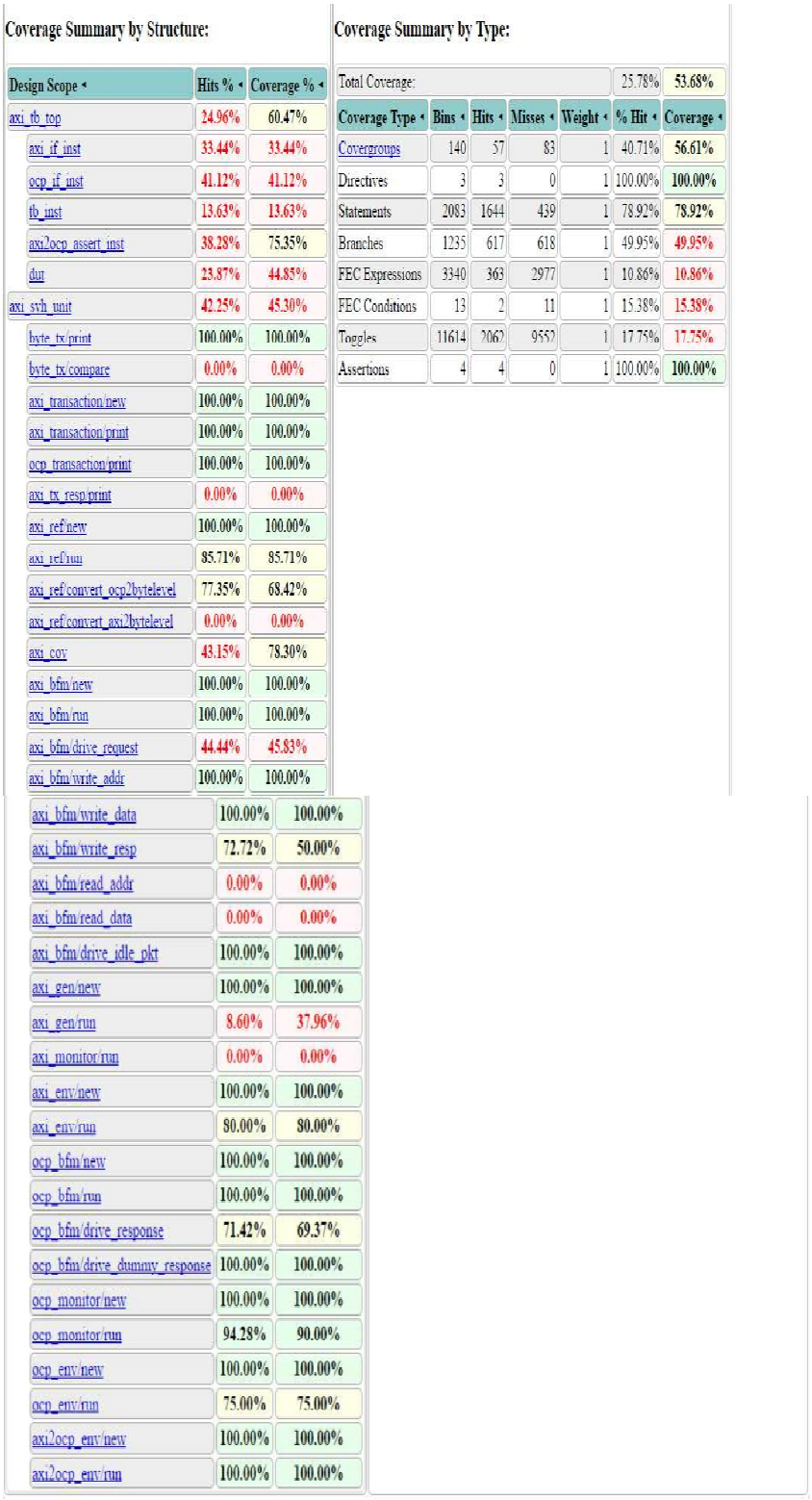


Fig. 5: Testing different burst type



Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	24.96%	60.47%
axi_if_inst	33.44%	33.44%
ocp_if_inst	41.12%	41.12%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	23.87%	44.85%
axi_svh_unit	42.25%	45.30%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	100.00%	100.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	77.35%	68.42%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	43.15%	78.30%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	44.44%	45.83%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	0.00%	0.00%
axi_bfm/read_data	0.00%	0.00%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	8.60%	37.96%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	71.42%	69.37%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					25.78%	53.68%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	57	83	1	40.71%	56.61%
Directives	3	3	0	1	100.00%	100.00%
Statements	2083	1644	439	1	78.92%	78.92%
Branches	1235	617	618	1	49.95%	49.95%
FEC Expressions	3340	363	2977	1	10.86%	10.86%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	2062	9552	1	17.75%	17.75%
Assertions	4	4	0	1	100.00%	100.00%

Fig. 6: Coverage report of testing different burst types

Testcase 2 is for testing the different burst types of the AXI protocol such as fixed burst type, wrap burst type, increment burst type and reserved burst type. Fig 5 shows the simulation waveform obtained for the different burst type test and the Fig. 6 gives the coverage report of this testcase. The coverage report indicates a functional coverage of 56.61 percent and 100 percent assertion coverage.

3) Testcase 3: Write_read transaction

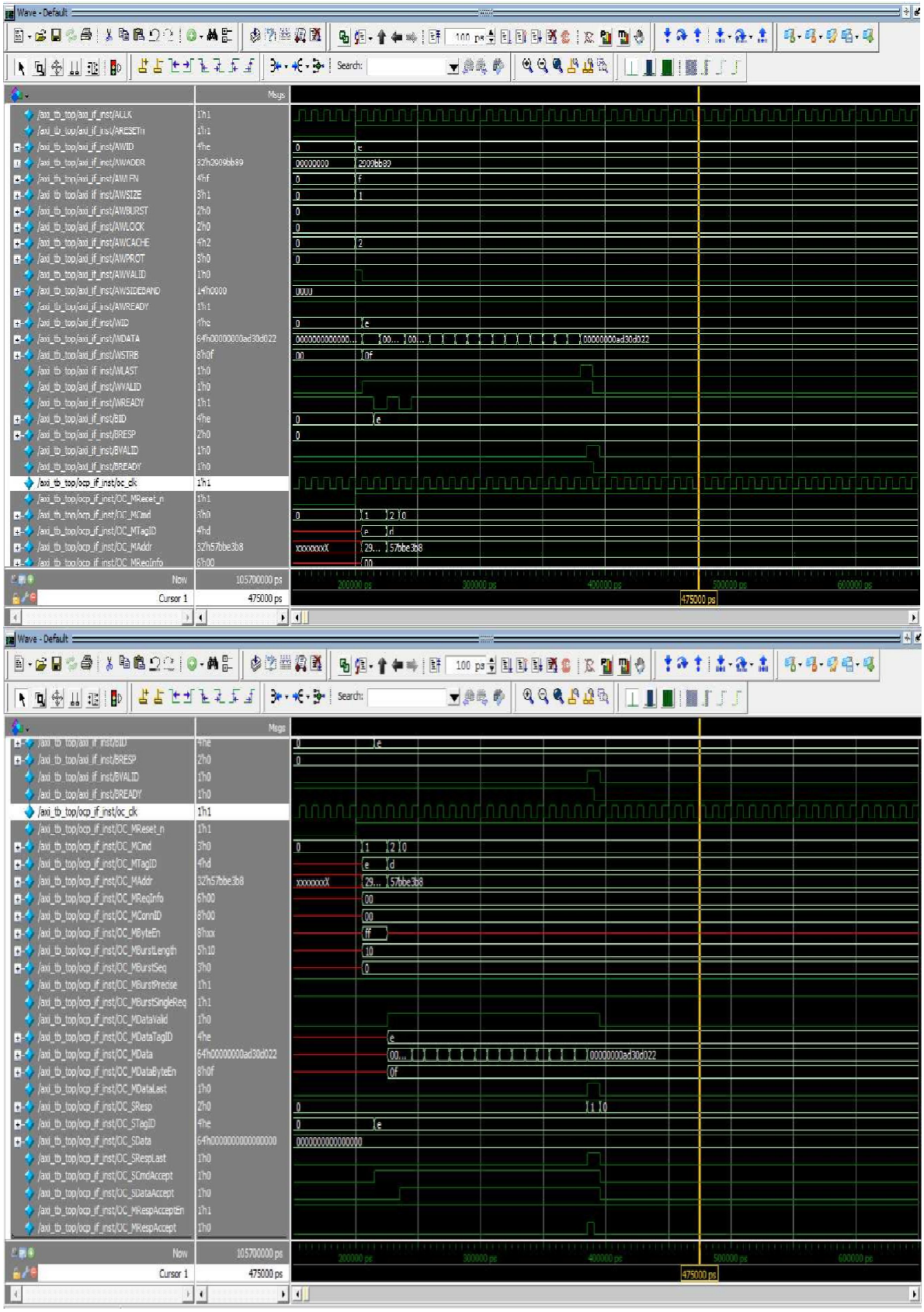


Fig. 7:Write_read transaction

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	20.77%	59.44%
axi_if_inst	25.66%	25.66%
ocp_if_inst	22.94%	22.94%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	20.15%	43.52%
axi_svh_unit	46.49%	49.09%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	100.00%	100.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	75.47%	65.78%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	52.73%	85.53%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	55.55%	54.16%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	94.11%	75.00%
axi_bfm/read_data	33.33%	22.22%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	8.60%	37.96%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	71.42%	69.37%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					21.98%	54.73%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	71	69	1	50.71%	71.07%
Directives	3	3	0	1	100.00%	100.00%
Statements	2083	1669	414	1	80.12%	80.12%
Branches	1235	609	626	1	49.31%	49.31%
FEC Expressions	3340	348	2992	1	10.41%	10.41%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	1347	10267	1	11.59%	11.59%
Assertions	4	4	0	1	100.00%	100.00%

Fig. 8: Coverage report of write_read transaction

Testcase 3 indicates the write and read transaction verification of the AXI2OCP Bridge. The AXI protocol writes a data to the memory and the read operation takes place to the same memory location by the OCP. Fig. 7 shows the simulation waveform of the write and write transaction and Fig. 8 gives the coverage report. The coverage report indicates a 71.07 percent functional coverage and 100 percent assertion coverage.

4) Testcase 4: Write_read loop test

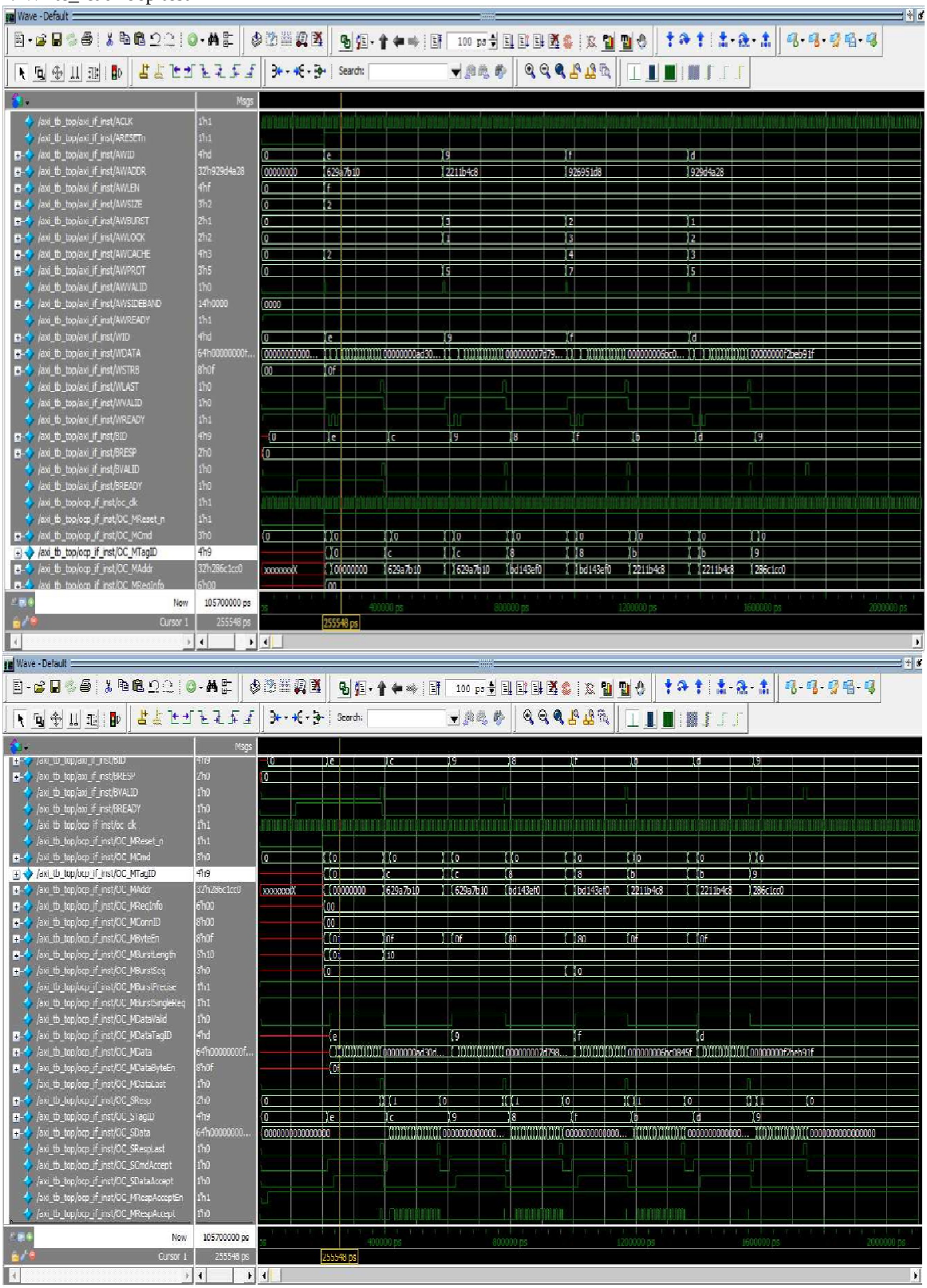
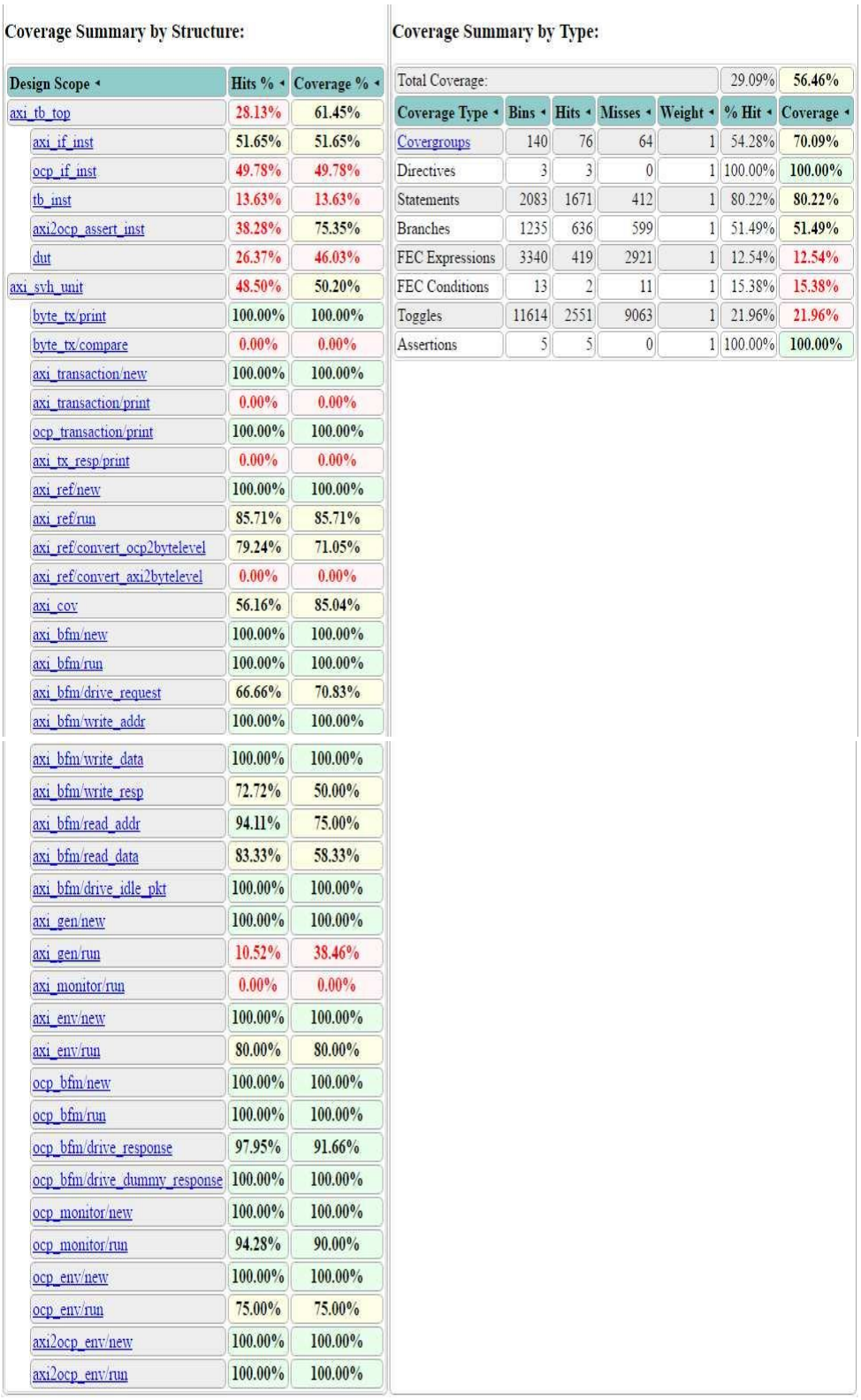


Fig. 9: Write_read loop test



Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	28.13%	61.45%
axi_if_inst	51.65%	51.65%
ocp_if_inst	49.78%	49.78%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	26.37%	46.03%
axi_svh_unit	48.50%	50.20%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	0.00%	0.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	79.24%	71.05%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	56.16%	85.04%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	66.66%	70.83%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	94.11%	75.00%
axi_bfm/read_data	83.33%	58.33%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	10.52%	38.46%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	97.95%	91.66%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					29.09%	56.46%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	76	64	1	54.28%	70.09%
Directives	3	3	0	1	100.00%	100.00%
Statements	2083	1671	412	1	80.22%	80.22%
Branches	1235	636	599	1	51.49%	51.49%
FEC Expressions	3340	419	2921	1	12.54%	12.54%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	2551	9063	1	21.96%	21.96%
Assertions	5	5	0	1	100.00%	100.00%

Fig. 10: Coverage report of write_read loop test

Testcase 4 is a write and read loop test. Fig. 9 indicates the simulation waveform of this testcase and Fig. 10 gives the coverage report. The obtained functional coverage is 70.09 percent and the assertion coverage is 100 percent.

5) Testcase 5: Write_read with varying AWLEN

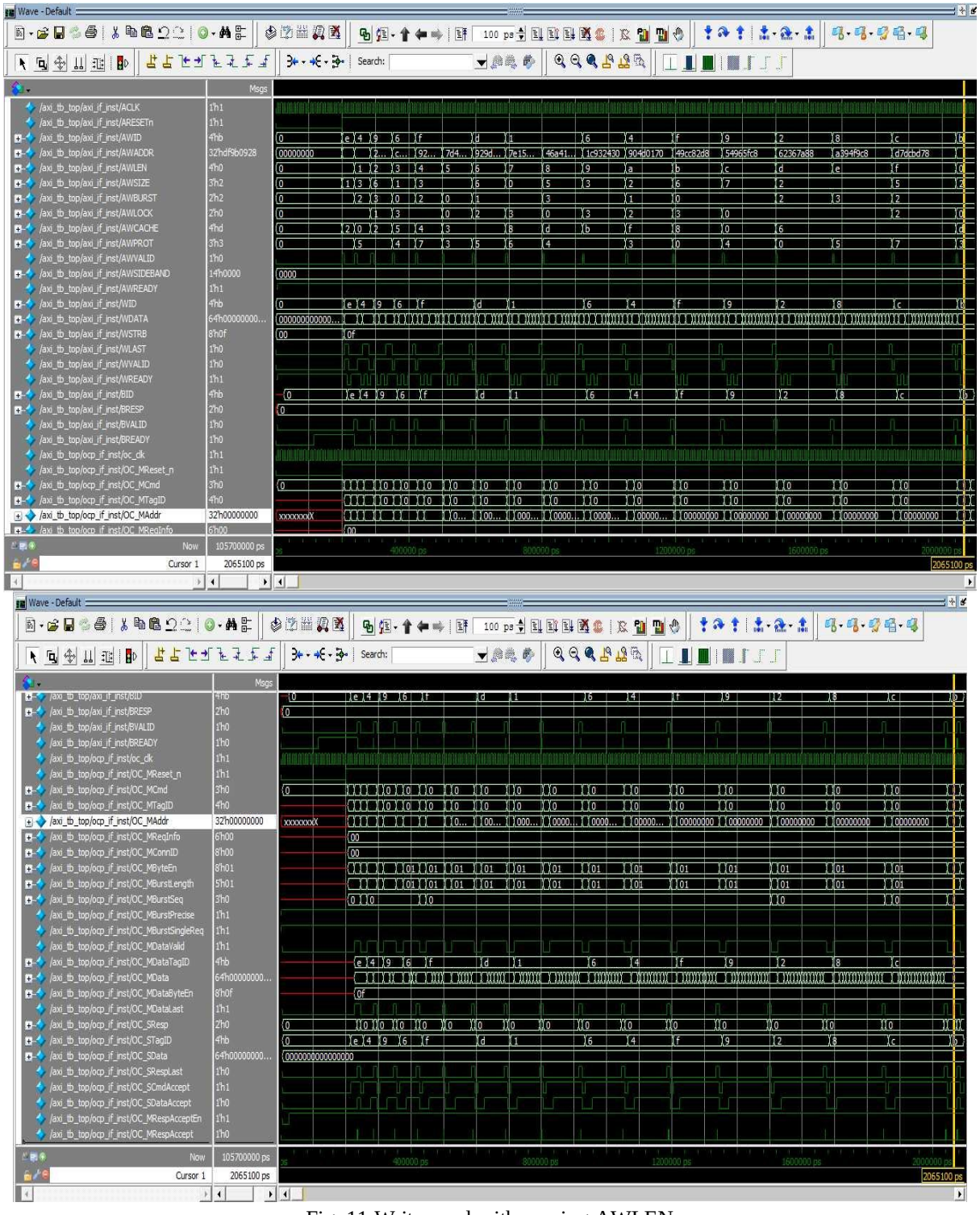


Fig. 11:Write_read with varying AWLEN

Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	28.96%	62.21%
axi_if_inst	47.01%	47.01%
ocp_if_inst	51.73%	51.73%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	27.38%	47.17%
axi_svh_unit	55.46%	55.37%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	100.00%	100.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	79.24%	71.05%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	93.83%	98.10%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	66.66%	70.83%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	94.11%	75.00%
axi_bfm/read_data	33.33%	22.22%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	13.77%	39.72%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	97.95%	91.66%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					30.23%	60.32%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	131	9	1	93.57%	96.20%
Directives	3	3	0	1	100.00%	100.00%
Statements	2098	1700	398	1	81.02%	81.02%
Branches	1235	664	571	1	53.76%	53.76%
FEC Expressions	3340	459	2881	1	13.74%	13.74%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	2614	9000	1	22.50%	22.50%
Assertions	5	5	0	1	100.00%	100.00%

Fig. 12: Coverage report of Write_read with varying AWLEN

Testcase 5 verifies the AXI2OCP Bridge for a write and read transaction with AWLEN varying from 0 to 15. Fig. 11 indicates the simulation waveform of this testcase and the Fig. 12 gives the coverage report for this testcase. The functional coverage obtained for this testcase is 96.20 percent and the assertion coverage obtained is 100 percent.

The bus utilization percentages of the above testcases are tabulated in Table.1. The average bus utilization obtained is 88.584percent.

Table. 1. Bus utilization

Testcase	Testcase name	Valid Count	Busy Count	Bus Utilization %= ———— *100
1	Write transaction with varying AWLEN	136	165	82.42%
2	Testing different burst type	39	43	90.69%
3	Write_read transaction	17	19	89.47%
4	Write_read loop test	16	18	88.89%
5	Write_read with varying AWLEN	139	152	91.45%

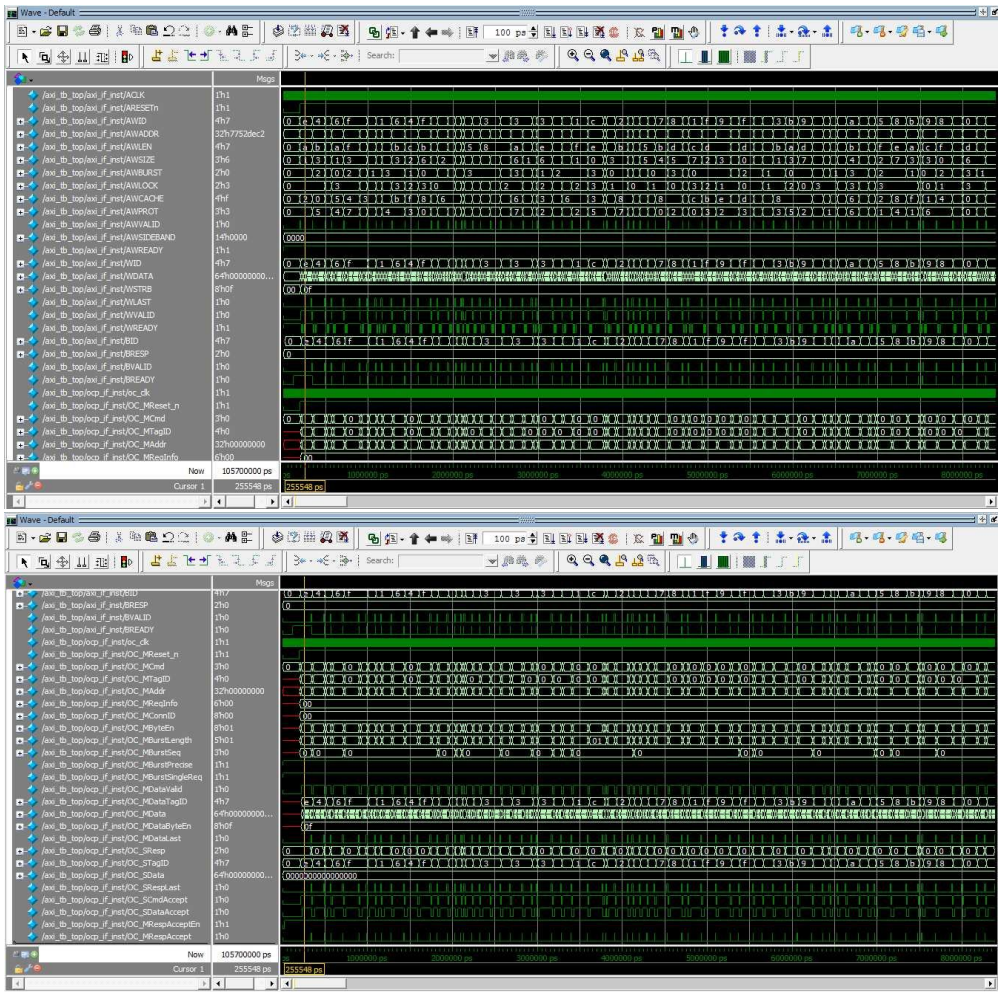
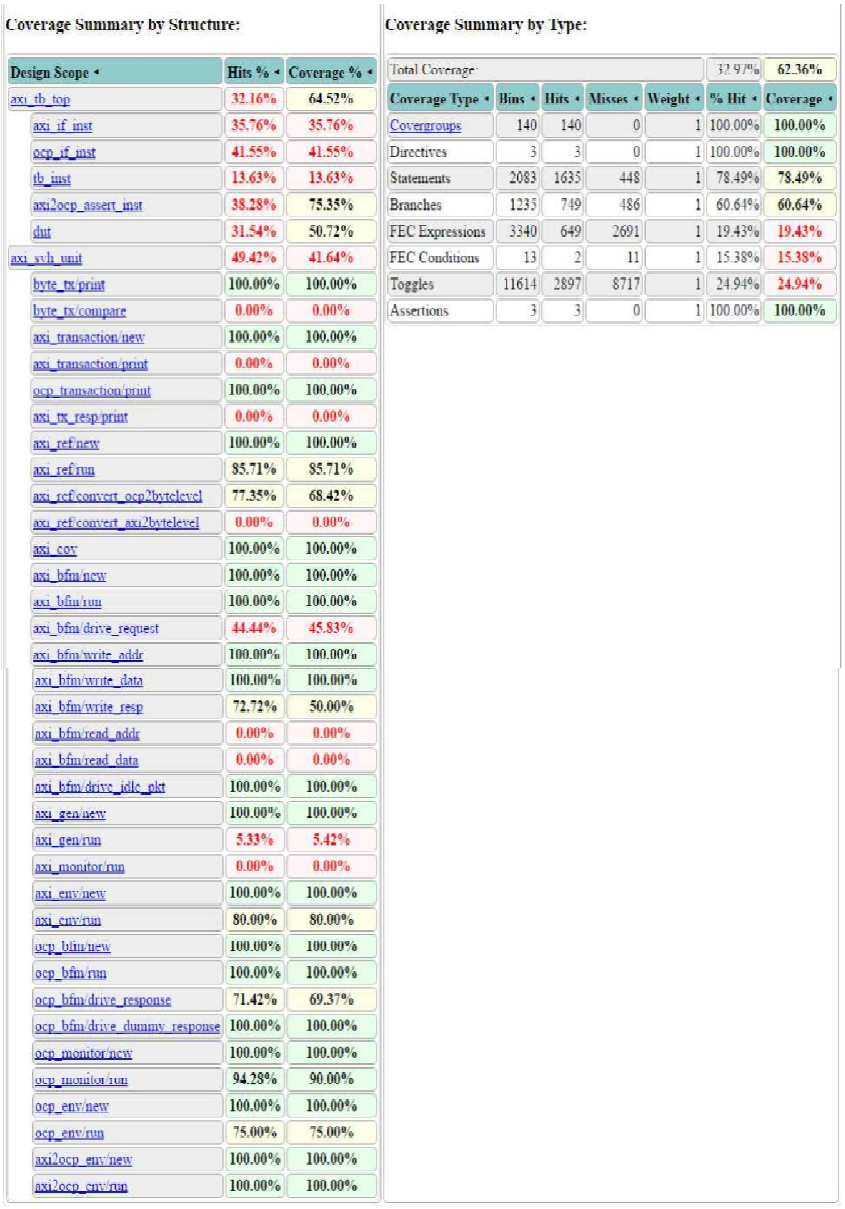


Fig. 13: Merged testcases



Coverage Summary by Structure:

Design Scope	Hits %	Coverage %
axi_tb_top	32.16%	64.52%
axi_if_inst	35.76%	35.76%
ocp_if_inst	41.55%	41.55%
tb_inst	13.63%	13.63%
axi2ocp_assert_inst	38.28%	75.35%
dut	31.54%	50.72%
axi_svh_unit	49.42%	41.64%
byte_tx/print	100.00%	100.00%
byte_tx/compare	0.00%	0.00%
axi_transaction/new	100.00%	100.00%
axi_transaction/print	0.00%	0.00%
ocp_transaction/print	100.00%	100.00%
axi_tx_resp/print	0.00%	0.00%
axi_ref/new	100.00%	100.00%
axi_ref/run	85.71%	85.71%
axi_ref/convert_ocp2bytelevel	77.35%	68.42%
axi_ref/convert_axi2bytelevel	0.00%	0.00%
axi_cov	100.00%	100.00%
axi_bfm/new	100.00%	100.00%
axi_bfm/run	100.00%	100.00%
axi_bfm/drive_request	44.44%	45.83%
axi_bfm/write_addr	100.00%	100.00%
axi_bfm/write_data	100.00%	100.00%
axi_bfm/write_resp	72.72%	50.00%
axi_bfm/read_addr	0.00%	0.00%
axi_bfm/read_data	0.00%	0.00%
axi_bfm/drive_idle_pkt	100.00%	100.00%
axi_gen/new	100.00%	100.00%
axi_gen/run	5.33%	5.42%
axi_monitor/run	0.00%	0.00%
axi_env/new	100.00%	100.00%
axi_env/run	80.00%	80.00%
ocp_bfm/new	100.00%	100.00%
ocp_bfm/run	100.00%	100.00%
ocp_bfm/drive_response	71.42%	69.37%
ocp_bfm/drive_dummy_response	100.00%	100.00%
ocp_monitor/new	100.00%	100.00%
ocp_monitor/run	94.28%	90.00%
ocp_env/new	100.00%	100.00%
ocp_env/run	75.00%	75.00%
axi2ocp_env/new	100.00%	100.00%
axi2ocp_env/run	100.00%	100.00%

Coverage Summary by Type:

Total Coverage:					32.97%	62.36%
Coverage Type	**Bins**	**Hits**	**Misses**	**Weight**	**% Hit**	**Coverage**
Covergroups	140	140	0	1	100.00%	100.00%
Directives	3	3	0	1	100.00%	100.00%
Statements	2083	1635	448	1	78.49%	78.49%
Branches	1235	749	486	1	60.64%	60.64%
FEC Expressions	3340	649	2691	1	19.43%	19.43%
FEC Conditions	13	2	11	1	15.38%	15.38%
Toggles	11614	2897	8717	1	24.94%	24.94%
Assertions	3	3	0	1	100.00%	100.00%

Fig. 14: Coverage report of merged testcases

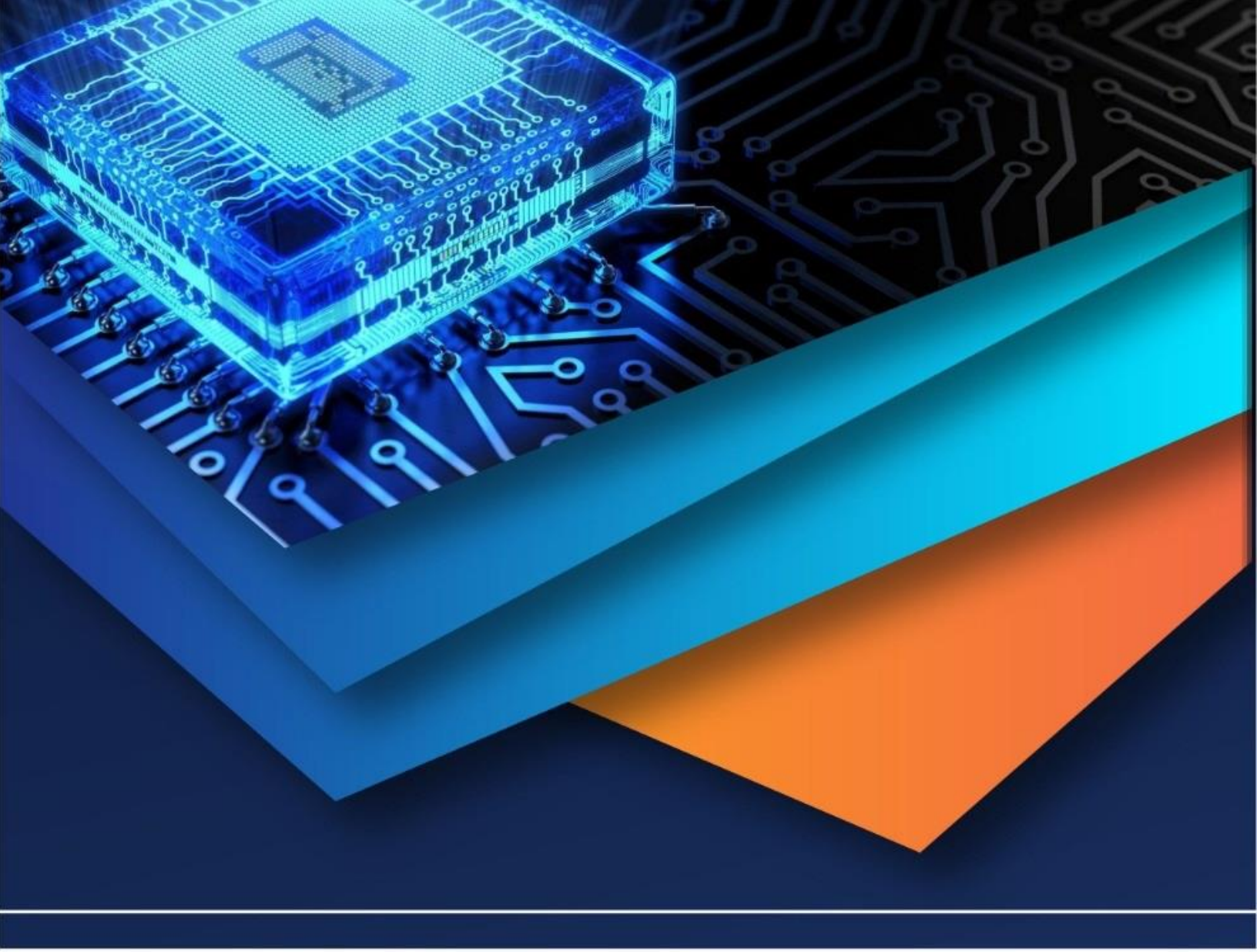
Fig. 13 indicates the simulation waveform when all the test cases are executed together or merged and the Fig. 14 gives the coverage report of the overall testcases. With inference from the above coverage report, we can see that 100 percent coverage is obtained for the functionality of the AXI2OCP Bridge and there is 100 percent coverage of assertions.

V. CONCLUSION

A verification environment is built for AXI2OCP Bridge using System Verilog. The reference model and checker are included in the verification environment for self checking. The simulation waveforms and coverage reports are generated using Questa sim 10.6c tool by Mentor Graphics. The AXI2OCP bridge is verified for various test cases like write transactions with varying AWLEN, testing various burst type, Write_read transaction, Write_read loop and Write_read transaction with varying AWLEN. Bus utilization is calculated for various test cases and obtained an average bus utilization of 88.584 percent. 100percent functional coverage is obtained indicating the efficient verification of the functionalities of the AXI2OCP Bridge. The coverage of the design has been improved through the Assertion Based Verification. The coverage reports show that 100 percent coverage is obtained for directives covergroups (functional coverage) and assertions.

REFERENCES

[1] N. Gaikwad and V. N. Patil, "Verification of AMBA AXI On-Chip Communication Protocol," 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), Pune, India, 2018, pp. 1-5.
[2] Limbanna. P. Rathod, Dr. Jayanthi. K. Murthy, Mr. Shaik Chand Basha, "Design and Coverage Driven Verification of AXI2OCP Bridge for Industrial SoC Designs", International Journal Of Advanced Research in Engineering & Management (IJAREM), 2017, PP. 07-13.
[3] C. Sarojini and J. Thangaraj, "Implementation and Optimization of Throughput in High Speed Memory Interface Using AXI Protocol," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bangalore, 2018, pp. 1-5.
[4] R. Bhakthavatchalu, G. R. Deepthy, S. Vidhya and V. Nisha, "Design and analysis of low power open core protocol compliant interface using VHDL," 2011 International Conference on Emerging Trends in Electrical and Computer Technology, Nagercoil, 2011, pp. 621-62.
[5] G. R. A. Pamarthy, M. D. Prakash and A. Yadlapati, "Extensions of open core protocol and their high level verification using system verilog and UVM," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, 2017, pp. 475-477.
[6] M. P. Deepu and R. Dhanabal, "Validation of transactions in AXI protocol using system verilog," 2017 International conference on Microelectronic Devices, Circuits and Systems (ICMDCS), Vellore, 2017, pp. 1-4
[7] G. Mahesh and S. M. Sakthivel, "Functional verification of the Axi2OCP bridge using system verilog and effective bus utilization calculation for AMBA AXI 3.0 protocol," 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, 2015, pp. 1-5.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY