iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VI **Month of publication:** June 2021

DOI: https://doi.org/10.22214/ijraset.2021.35847

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 9 Issue VI Jun 2021- Available at www.ijraset.com

Verification of Dual Port RAM using System Verilog and UVM

Geethashree[1], Anand[2]
[1, 2]*ECE Department (VLSI Design and Embedded systems), PES College of Engineering Mandya, Karnataka (India)*

Abstract: ***Verification process place a prominent role in the field of SoC and ASIC design. Several verification methodologies are there apart from those Universal Verification Methodology (UVM) is advanced and it is widely used by the industries due to its special features. UVM provides reusable and well-structured verification components by using System Verilog class library. In this work, Dual Port RAM is considered as Design Under Test (DUT). System Verilog and UVM verification environments are developed to verify the DUT. Assertion and cover group coverage are set up with a goal of achieving 100% from both the environments.***

Keywords: ***Verification, System Verilog, Testbench, Environment, UVM.***

I. INTRODUCTION

In ASIC applications higher priorities are given to multi-port memories due to usage of on chip shared memory concept in the systems. To reduce the complexity of the design developments done over multi-port, as a result multi-port SRAM compiler are developed to achieve high speed and high level of integration [5]. Overall chip performance can be increased due to parallel operations done by using multi-port RAM. Recent works are done with multi-processor system come up with an added feature in which Dual Port RAM takes a higher priority. DPRAM have an advantage over single port RAM, with the same number of cycles it can perform multiple Read\Write operations at the same time [1]. Dual Port RAM consist of two ports port A and port B. Both ports have an ability to perform Read\Write or Read only or Write only, these operations are controlled by clock [10].

In simple words we can define a verification, it is the process of checking the design which is working correct or not with respect to the expected result. Verification plan has to be made to describe what needs to verify, how to verify and what are all the features to be covered in verification. System Verilog (SV) is a standard hardware description language and it is also used for verification. It is extended from improvement to IEEE 1364 verilog-2001 standard. It inherits the features from Verilog HDL, VHDL, C and C++. System Verilog is used to design and implement electronic circuits, they are going to verified by generating the testcases using OOPs concepts [11]. Rapid growth in the chip manufacturing process leads to integrate million number of transistors on a single chip. Difficulty in verification arises and overall, it slows down the chip design. Apart from the entire design cycle 70 to 80 percent of the time consumed by functional verification. To avoid the gap between production of product and time to market, it is necessary to use verification methodology with an effective environment. Reusability is one of the criteria of effective verification process. VMM\UMV and many of VIPs are some of verification methodologies for reuse, so that verification environment can be built quickly [19]. Most of the IC design manufacturing industries adopted UVM verification methodology to avoid complexity in verification and to achieve verification in an effective manner. UVM is extended version of OVM and it also supports some features of VMM. Reusability is the key factor in UVM, this can be achieved by using call-back function, overloading of the factory mechanism and inheritance property of UVM class [12]. Dual Port RAM is considered as DUT for verification.

This paper is divided into sections as fallows: In section 2, describing the DPRAM in a simple way and the objectives of the work. Section 3, is about System Verilog verification environment and also simulation results of DPRAM in System Verilog environment. Section 4, talk about UVM verification environment of DUT and its simulation results. Section 5, conclusion of the work is discussed.

II. DUAL PORT RAM

Dual Port RAM (DPRAM) is a type of random-access memory, it permits to perform multiple write or reads takes place at the same time on both ports or it almost near. Transaction can be done synchronously or asynchronously. Both ports have separate clock, enable and control signals for the transactions. To the single memory, they have ability to access locations. Each port has address, data input, data output, enable signals and control signals. Block diagram of Dual Port RAM is shown in figure 1.

A. Objectives

1) The proposed verification environment will be implemented by using OOP concepts.
2) The functionality of the design is verified using System Verilog and UVM.
3) Generation of multiple Test Cases to analyze the design.
4) Achieve 100% Coverage and Assertions.

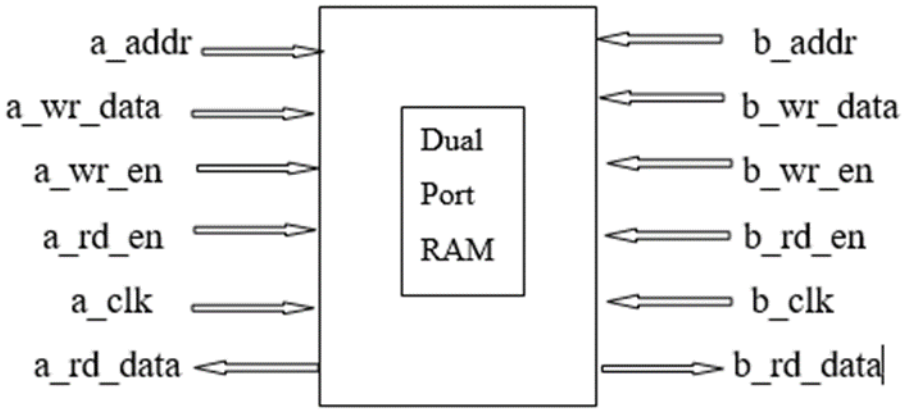


Fig.1. Block diagram of DPRAM

III. SYSTEM VERILOG ENVIRONMENT

A. System Verilog Verification Environment

System Verilog verification environment of DUT is illustrated in figure 2. We developed verification environment in such a way that to cover all the objectives.

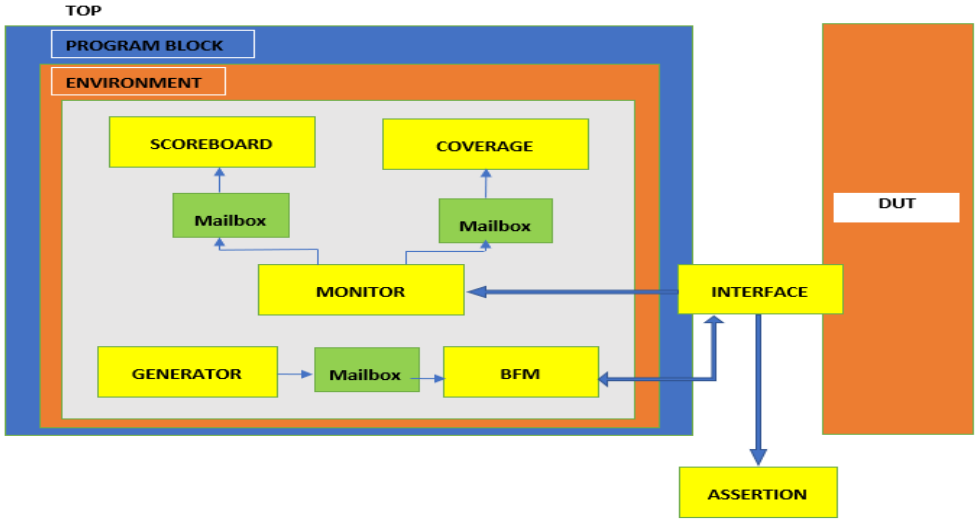


Fig.2. System Verilog verification environment.

1) Generator: Stimulus is generated randomly and these are driven by Bus Functional Model (BFM) it's nothing but a driver.
2) Functional coverage: Coverage is a measure of how much scenario\features covered. In SV coverage is achieved by using cover group, it consists of cover points and bins. Each variable is associated with bins if all the bins are covered then only, we can achieve complete coverage.
3) Assertion: Design behaviour is verified by using assertions and it can be used to check particular conditions. There are two types of assertions immediate assertion and concurrent assertion. In immediate assertion, at current simulation time only it will check for the condition. While in concurrent assertion, over multiple clock cycles it examines the sequence of conditions.

B. Testcases

We considered four testcases

1) Both ports are trying to write data to same location
2) Both ports reading data from same location
3) Mixed read and write
4) Writing data to both ports and reading data in a crossed version.

C. Simulation Results

All the testcases are verified results of Dual Port RAM of fourth testcase is shown below.

1) Transcript window results

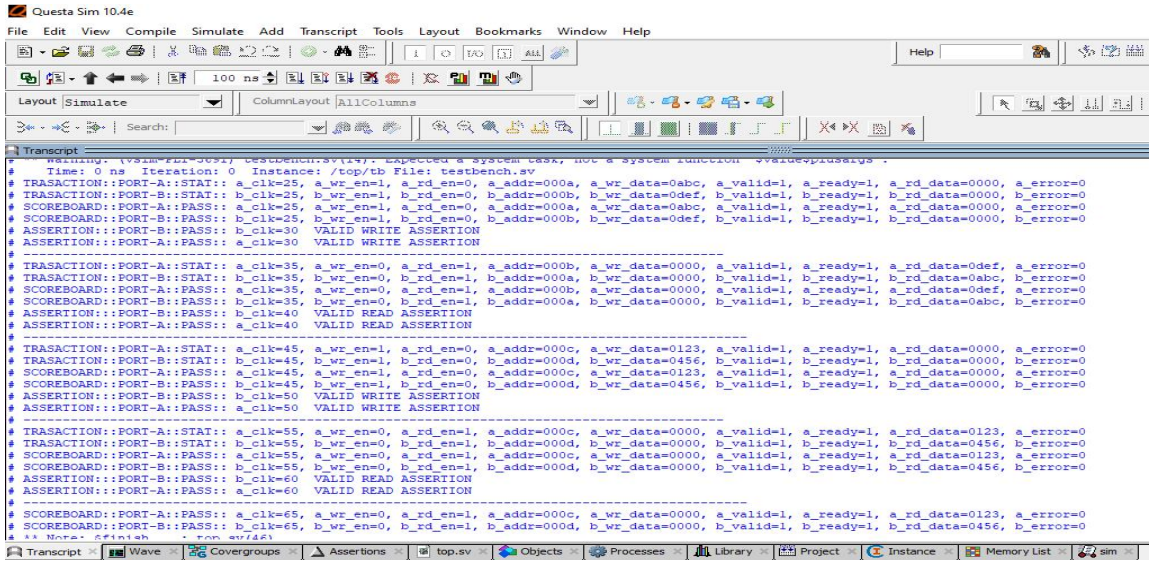


Fig.3. Transcript window results of DPRAM.

2) Waveform

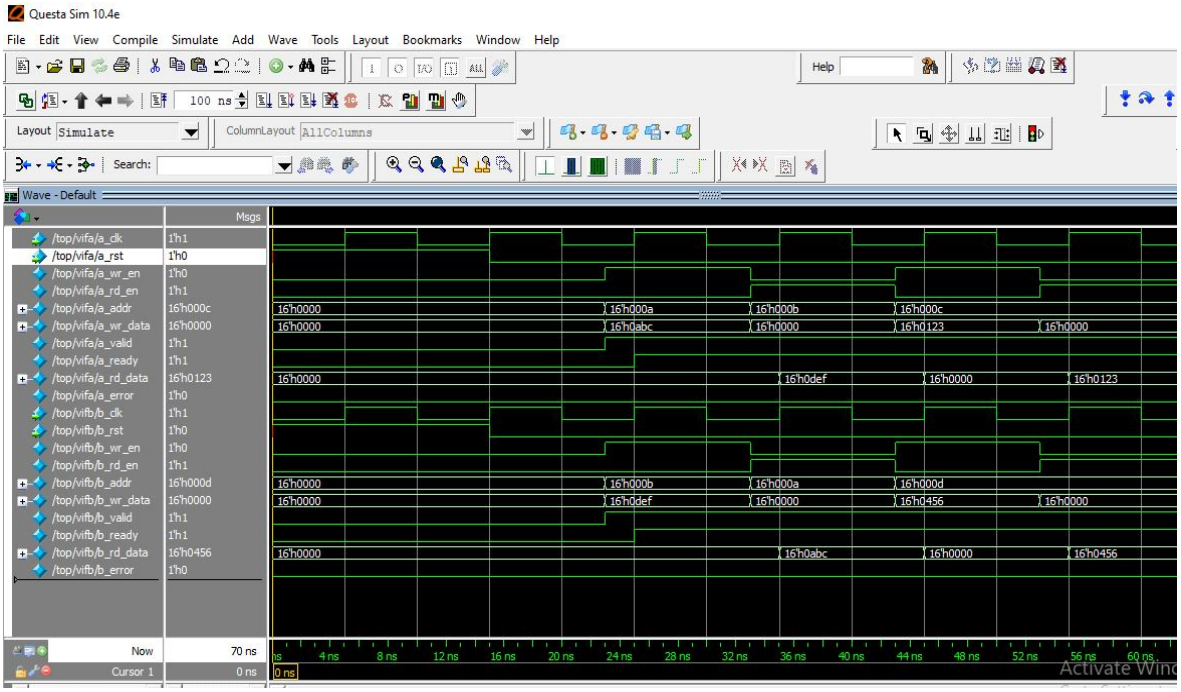


Fig.4. Waveform results.

3) Coverage output

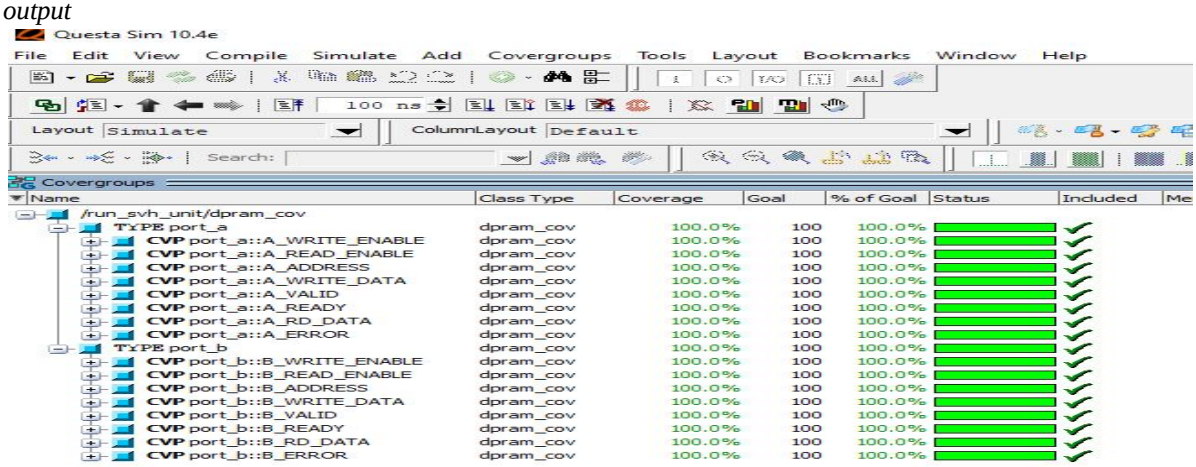


Fig.5. Results of coverage.

4) Result of total coverage and assertion

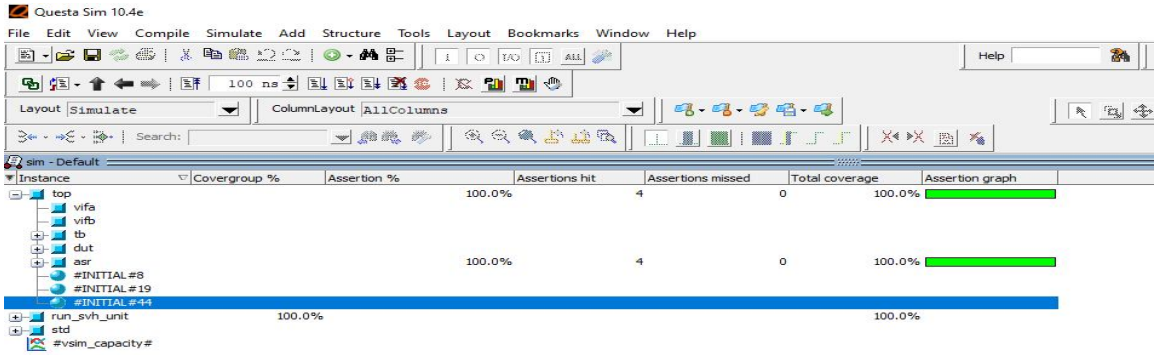


Fig.6. Total coverage and assertion result.

IV. UVM VERIFICATION ENVIRONMENT

UVM verification methodology is one of the advanced and broadly used verification methodologies. It claims reusability criteria standard which is achieved by its library classes. These classes can be inherited uvm_component, uvm_object, uvm_sequence is few of them. Verification environment is developed in such a way that to achieve complete coverage and assertion. UVM verification environment is shown in figure 7.

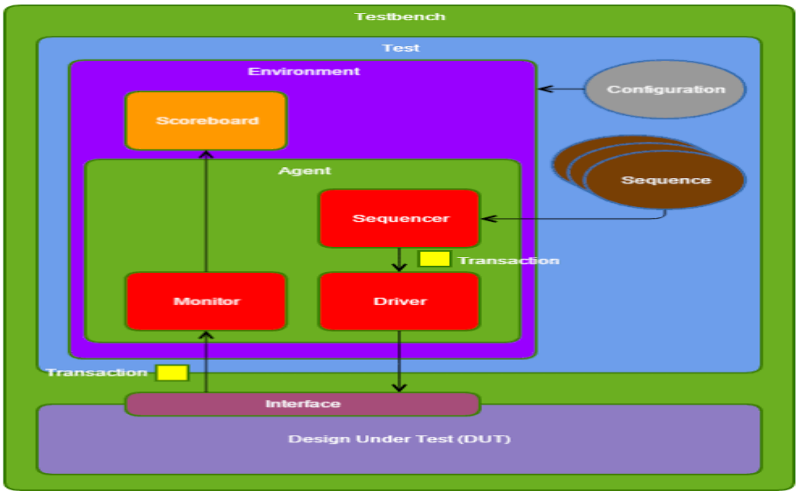


Fig.7. UVM verification environment

As stimulus is generated in the generator in SV, in UVM we have uvm_sequencer it consists of uvm_sequence_item and uvm_sequence. Agent generates the pin level activity which are defined by sequence item. These data packets are driven to\from the driver in a sequence through the routing mechanism of uvm_sequencer. TLM (Transaction Level Modeling) is meant for communication purpose; this can be done through implementing the methods which are transaction based. TLM interface class defines the methods which are defined for communication, those are put(), get(), peek(), try_put(), can_put(), try_get(), can_get(), transport(), nb_transport() and analysis.

A. Simulation Results

The results of Dual Port RAM are given below

1) Waveform results

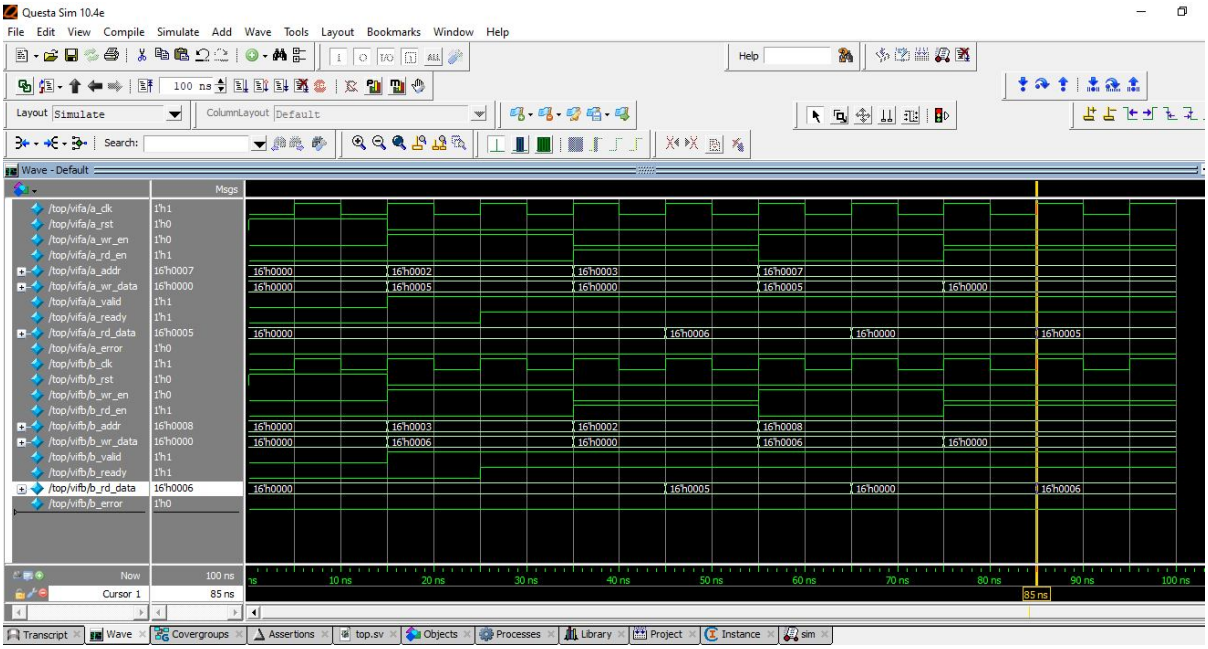


Fig.8. waveform results of DPRAM.

2) Total Coverage and Assertion Result

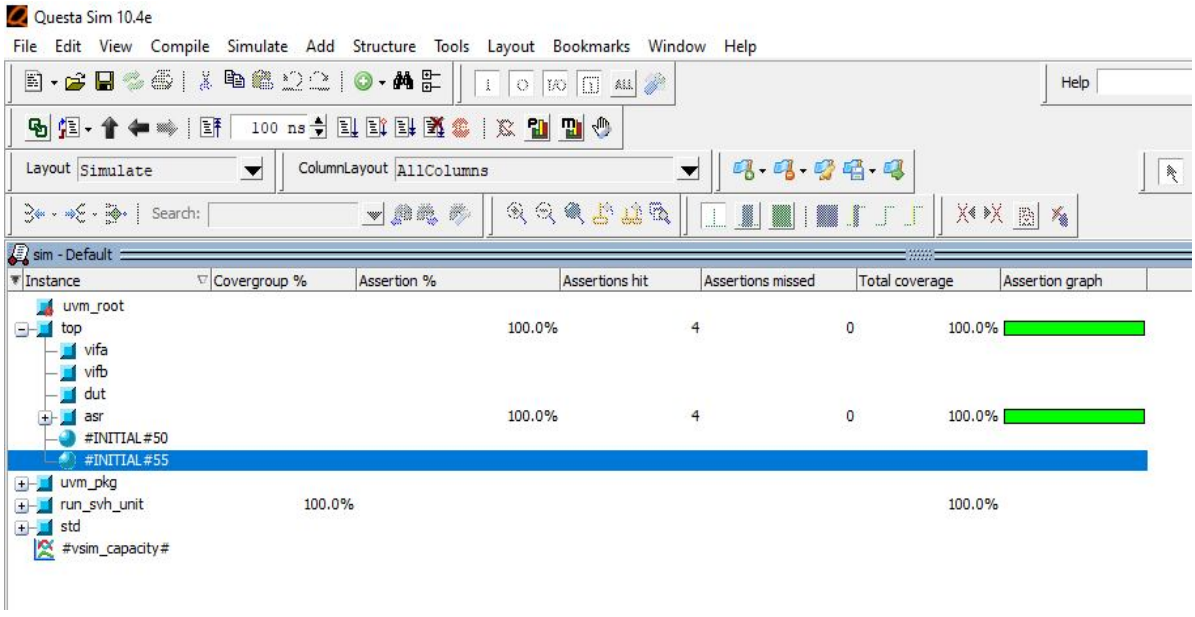


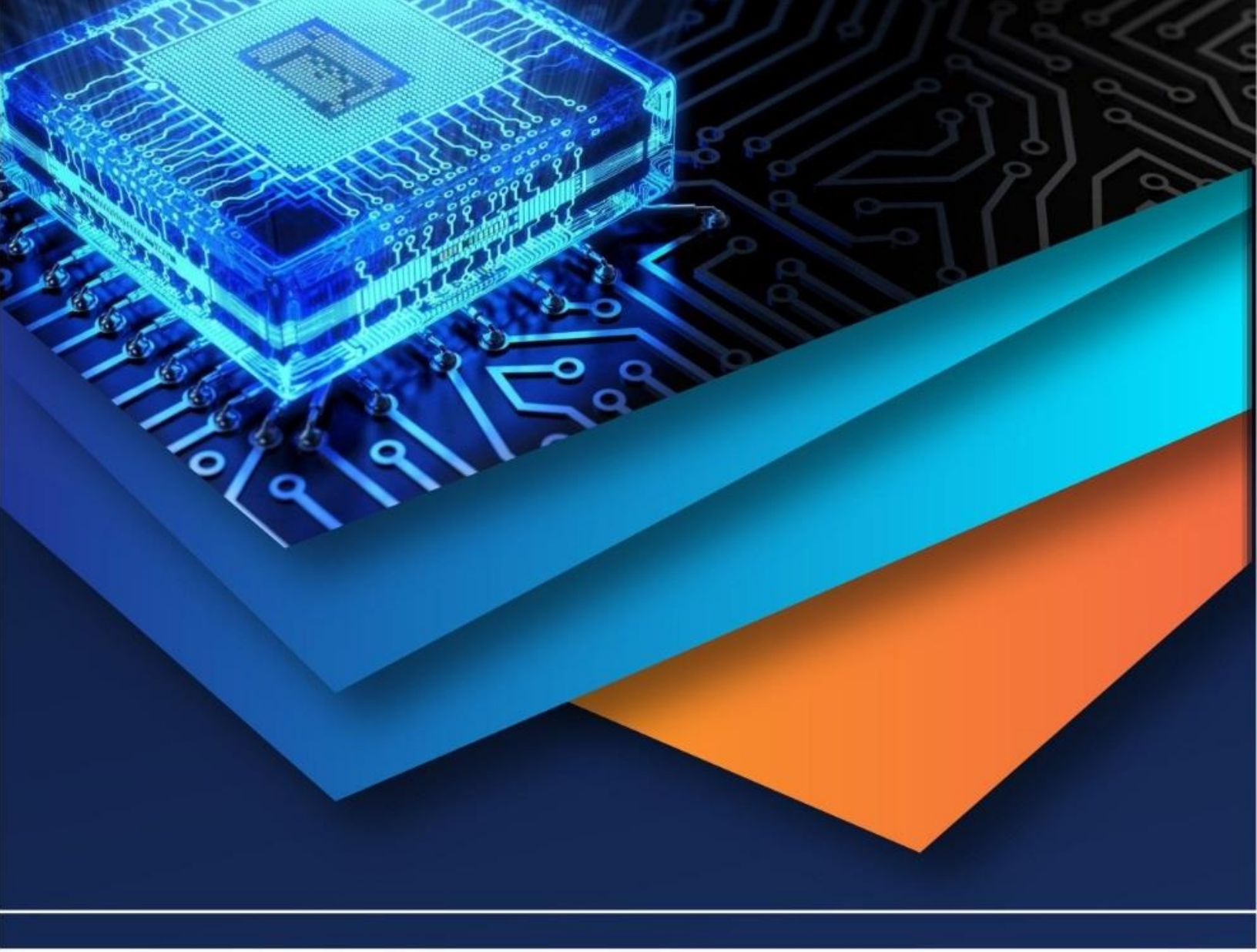
Fig.9. Result of total coverage and assertion

V. CONCLUSION

System Verilog and UVM Verification environments are developed by using OOPs concepts. Different classes are created for monitor, generator, scoreboard, driver and so on. All the testcases are tested in both the environments. Objectives are covered, from both the environments 100 % cover group coverage and 100% assertion is found for DPRAM. Therefore, the overall coverage achieved is 100% for verification of DPRAM which is considered as DUT in System Verilog and UVM verification environments.

REFERENCES

[1] Aditi, Pawan Kumar Dahiya. “Design and Verification of Dual Port RAM using System Verilog Methodology.” In International Journal for Research in Applied Science & Engineering Technology (IJRASET), IEEE, 2019.
[2] Lockhart, W. L., J. H. Greene, and D. T. Young. "A high-speed low-power spectrum accumulator using dual-port RAM and state machine control." IEEE Transactions on Nuclear Science36, no. 4 (1989): 1396-1403.
[3] Salah, Khaled, and Hassan Mostafa. "Constructing Effective UVM Testbench for DRAM Memory Controllers." In 2018 New Generation of CAS (NGCAS), pp. 178-181. IEEE, 2018.
[4] Karfa, Chandan, Chittaranjan Mandal, Dipankar Sarkar, S. R. Pentakota, and Chris Reade. "Verification of scheduling in high-level synthesis." In IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), pp. 6-pp. IEEE, 2006.
[5] Cristina Silvano, Giancarlo Sada, and Laura Populin. “Ramgen: A Dual Port Static RAM Generator.” In [1992] proceedings. Fifth annual IEEE International ASIC Conference and Exhibit, pp. 509-512. IEEE, 2002.
[6] Nangia, Rakhi, and Neeraj Kr Shukla. "Functional verification of I2C core using SystemVerilog." International Journal of Engineering, Science and Technology 6, no. 4 (2014): 31-44.
[7] Gao, Xinyan, Ning Zhou, Jinzhao Wu, and Dakui Li. "Wu’s characteristic set method for SystemVerilog assertions verification." Journal of Applied Mathematics 2013 (2013).
[8] Gao, Yingke, Diancheng Wu, Quanquan Li, Tiejun Zhang, and Chaohuan Hou. "Design and implementation of transaction level processor based on UVM." In 2013 IEEE 10th International Conference on ASIC, pp. 1-4. IEEE, 2013.
[9] Surepeddi, Ravi. "System Verilog for Quality of Results (QoR)." In 9th International Symposium on Quality Electronic Design (isqed 2008), pp. 460-464. IEEE, 2008.
[10] Muehlegg, Franz, and Alfred Schuetz. "A highly flexible dual-port-RAM compiler." In [Proceedings] EURO ASIC90, pp. 277-281. IEEE, 1990.
[11] Yadu Krishnan K, Ramesh Bhakthavatchalu. “Block Level SoC Verification Using System Verilog.” Proceedings of the Third International Conference on Electronics Communication and Aerospace Technology [ICECA 2019], pp. 878-887. IEEE, 2019.
[12] Ni, Wei, and Jichun Zhang. "Research of reusability based on UVM verification." In 2015 IEEE 11th International Conference on ASIC (ASICON), pp. 1-4. IEEE, 2015.
[13] Li, Yangyang, Wuchen Wu, Ligang Hou, and Hao Cheng. "A study on the assertion-based verification of digital IC." In 2009 Second International Conference on Information and Computing Science, vol. 2, pp. 25-28. IEEE, 2009.
[14] Nii, Koji, Yasumasa Tsukamoto, Makoto Yabuuchi, Yasuhiro Masuda, Susumu Imaoka, Keiichi Usui, Shigeki Ohbayashi, Hiroshi Makino, and Hirofumi Shinohara. "Synchronous ultra-high-density 2RW dual-port 8T-SRAM with circumvention of simultaneous common-row-access." IEEE Journal of Solid-State Circuits 44, no. 3 (2009): 977-986.
[15] Mulani, Purvi D. "SoC level verification using System Verilog." In 2009 Second International Conference on Emerging Trends in Engineering & Technology, pp. 378-380. IEEE, 2009.
[16] Pavithran, T. M., and Ramesh Bhakthavatchalu. "UVM based testbench architecture for logic sub-system verification." In 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), pp. 1-5. IEEE, 2017.
[17] Sukhanya, M., and K. Gavaskar. "Functional verification environment for I2C master controller using system verilog." In 2017 Fourth International Conference on Signal Processing, Communication and Networking (ICSCN), pp. 1-6. IEEE, 2017.
[18] Visalli, Giuseppe. "UVM-based verification of ECC module for flash memories." In 2017 European Conference on Circuit Theory and Design (ECCTD), pp. 1-4. IEEE, 2017.
[19] Lingling Chai, Zheng Xie, Xin’an Wang. “A Verification Methodology for Reusable Test Cases and Coverage Based on System Verilog.” In 2014 IEEE International Conference on electron devices and solid-state circuits, IEEE, 2014.
[20] Rejusha, M., and K. B. Jayanthi. "Implementation of VLSI-oriented FELICS algorithm using Pseudo Dual-Port RAM." In International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012), pp. 68-73. IEEE, 2012.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY