iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 Issue: V Month of publication: May 2019

DOI: https://doi.org/10.22214/ijraset.2019.5340

Design and Verification of Dual Port RAM using System Verilog Methodology

Aditi[1], Pawan Kumar Dahiya[2]
[1, 2]ECE Department (VLSI Design), Deenbandhu Chhotu Ram University of Science and Technology, Murthal, Sonepat, Haryana (India)

Abstract: Verification ambiance may be able application System Verilog after application any accurate methodology but that will be different for every distortion of the design. There are assorted analysis methodologies out of which Universal Verification Methodology (UVM) is broadly adopted by the analysis industry worldwide, as the verification ambiance created application UVM is reusable, able and well structured. In this work we discuss the System Verilog and UVM verification environments. The Design Under Test (DUT) is the Dual Port RAM. The environments created application System Verilog and UVM, absolutely wrap the DUT. The assertion advantage begin is 100% and cover group advantage is begin 20.1% from SV environment. Therefore, the all-embracing advantage begin is 100% from developed system verilog environment.

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

I. INTRODUCTION

With respect to late arrangements for multiprocessors framework Dual port RAM have been comprehensively used of the purpose behind correspondence besides majority of the data advertising. Dual port RAM might interesting done connection to single port RAM by the individuals aspects for two units alternately ports might settle on accessed every last one of same time [1]. Today, for quick headway for facilitated circuits, the flightiness of the propelled IC outlines might be increasing, moreover at any point expanding adding up might have been troublesome with check. The workload over verification have accounted should 70% with 80% of the entire setup in addition change. Change from guaranteeing verification procedure or confirmation systems makes it workable on finish in addition effectively get up and go the individuals verification viability [12].

The multi-port RAM will be suitably to parallel operation in addition enhances the individuals absolute chip execution. Despite the certainty that the individual memory access speed (clock cycles) enhances to stretching measure overlook ports of the multi port RAM, its area discipline similarly manufactures for the individuals number to ports. Inevitably examining stretching DPRAM capability, the occupation regarding chip increases, something in that a higher thickness of the DPRAM might be unequivocally needed. Previously, general, the unit-cell extent of the dual-port ram may be in regards to dual comparatively concerning illustration enormous similarly that for single-port ram to date. Despite the certainty that the individuals area discipline requirement been decrease to the new outline structure [3]. Functional verification will be the procedure from guaranteeing tolerating the individuals want suchlike every single a standout amongst setup determinations. Here may a chance to be a couple measure for test circumstances constructed should indicate that the desire from asserting design under test will a chance to be well-preserved done its scenarios. The compelled random verification blankets very nearly 80% of the situations. They need aid complimented for control test cases with test hard-to compass test particular circumstances for random stimulus[15]. Similarly verification may be at present perceived concerning delineation the individuals bottleneck for whatever flighty VLSI framework. Thus pushing ahead the verification adequacy will a chance to be must. There have help two levels regarding verification, IP-level affirmation likewise SOC-level affirmation. To IP level verification, we prerequisite ought weigh the individuals reason. With SOC-level verification, we need will weigh connectivity. A group controller may be intrigued by right also triumph to correspondence with those device, secondary information through-put and in sparing energy [4]. Verification specific designers must assurance every single a standout amongst properties of the IC get executed suitably in front about making stage. Functional verification provides for respectable measure starting with asserting diminishments and is a staggering assistance in the field to IC plan. The functional verification methodologies incorporates VMM, OVM and UVM [11].

This paper is organized as follows. Section 2 has small description of the DUT that is DPRAM. Section 3, talk about the system verilog environment of the dual-port RAM and its simulation results .Section 4, discuss about the UVM environment of verification. We also present the different features of UVM .section 5 close with conclusion of the work.

II. DUAL PORT RAM

Dual-port ram (DPRAM) is a kind for arbitrary-access memory that permits various peruses or composes should happen during those same time, or About the same time, Dissimilar to single-ported ram which permits special case get at once. DPRAM might have been after the fact advancement about SPRAM, utilized within pc memory start to 2000. Resulting variants need aid numbered consecutively. DPRAMs respond asynchronously will area in addition control pin transforms. DPRAMs assistance three working modes: Pipelined, Flow-Through, Moreover impact. Clinched alongside DPRAMS, synchronous units similarly provide for synchronous passage capacity around whatever region in the memory. Potentially port might create alternately read majority of the data into/out-of any memory zone. Table 1 provide for the signals about signs under the DUT.

Table.1. Signals of DPRAM under DUT.

Reset	Read Enable	Write Enable	Operation	Comment
0			mem[1] = infinity	
If Else	0	0	Data_in <=’hx;	Unknown Value
	0	1	Mem[wrt_add]<=data_in;	Storing data
	1	0	Data_out<= mem[read_add]	Pass read address to memory
	1	1	Mem[wrt_add]<= Mem[read_add]	Both operations (read,write) performed.

III. SYSTEM VERILOG VERIFICATION ENVIRONMENT

A. Hierarchy Of Environment

The System Verilog code produced need the emulating segments. The progression of the code will be Concerning illustration indicated done figure 1.

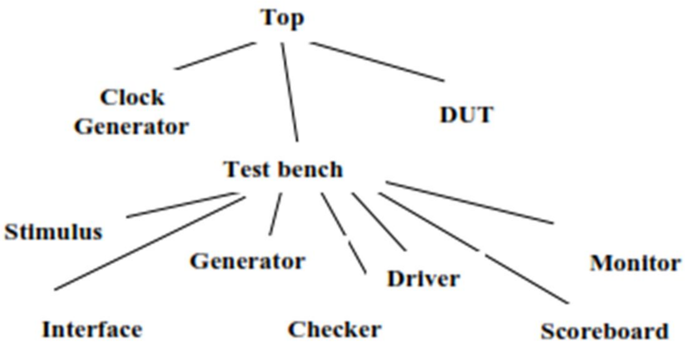


Fig.1. Hierarchy of SV.

The reason for testbench is with weigh the accuracy of the Design under test (DUT). For this emulating steps must make taken after :(1) Produce stimulus(2) Apply stimulus of the DUT(3)Catch reaction (4)Check for accuracy(5) Measure coverage .

1) Stimulus: It may be the provided for should different fields the individuals might be randomized for required constrained. It makes packet. The generator generates random stimulus starting with packet class and sends them with driver using mailbox.

```
Begin
case({we, re})//2'b00 :d_out <= 8'bz;//high impedance
2'b01 :   data_out <= mem[rd_addr];//memory from read add. will be written
2'b10 :   mem[wr_addr] <= data_in;//data will be written in mem
2'b11 :   begin
 data_out <= mem[rd_addr];
```

mem[wr_addr] <= data_in;
end

2) *Package:* Driver primary open package and interprets the operations processed toward the generator under the inputs for DUT. The driver sends these signs utilizing virtual interface on reset furthermore design DUT. Driver sends this package of the scoreboard utilizing mail box.

package pkg;
int no_of_tx = 50;
event DONE;
static int hit;
static int miss;
static int no_out;
`include "P1_transection.sv"
`include "P1_generator.sv"
`include "P1_wr_driver.sv"
`include "P1_rd_driver.sv"
`include "P1_wr_monitor.sv"
`include "P1_rd_monitor.sv"
`include "P1_scoreboard.sv"
`include "P1_env.sv"
`include "P1_test.sv"
Endpackage

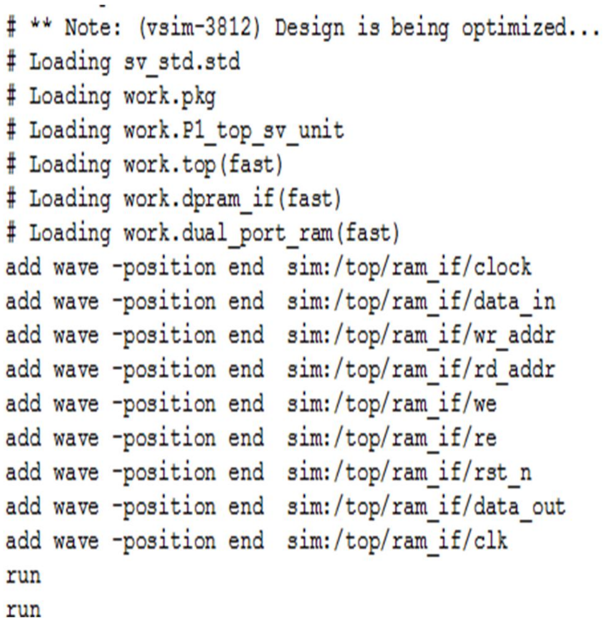
3) *Checker and Coverage:* Checker checks if information will be same as anticipated or not. Covergroup is coverpoints are embedded in the code to discover the scope.

covergroup mem_group2; // Place above function new.
option.per_instance = 1; //bins range hit once we get 100% coverage for a point. The at least column specifies how many hits are needed before a bin is considered covered.

B. Simulation Results

The results of dual port RAM is given below.

1) *Message view*



```
# ** Note: (vsim-3812) Design is being optimized...
# Loading sv_std.std
# Loading work.pkg
# Loading work.P1_top_sv_unit
# Loading work.top(fast)
# Loading work.dpram_if(fast)
# Loading work.dual_port_ram(fast)
add wave -position end  sim:/top/ram_if/clock
add wave -position end  sim:/top/ram_if/data_in
add wave -position end  sim:/top/ram_if/wr_addr
add wave -position end  sim:/top/ram_if/rd_addr
add wave -position end  sim:/top/ram_if/we
add wave -position end  sim:/top/ram_if/re
add wave -position end  sim:/top/ram_if/rst_n
add wave -position end  sim:/top/ram_if/data_out
add wave -position end  sim:/top/ram_if/clk
run
run
```

Fig.1. Message Result

2) *Waveforms obtained*

Figure.2 gives the waveform results.

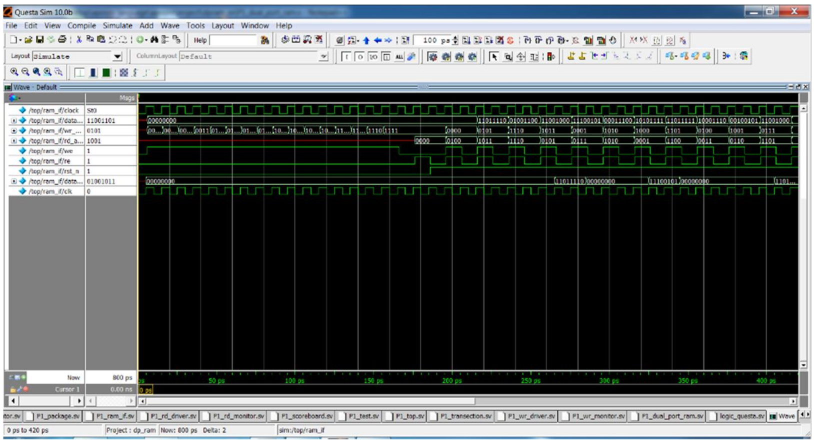


Fig.2. Simulation showing enabled DPRAM.

3) *RTL View*

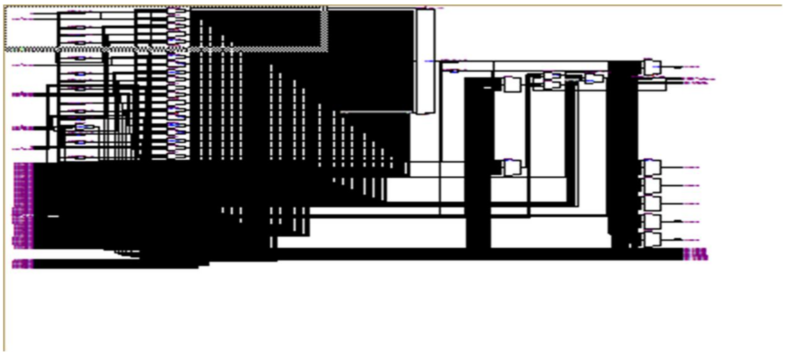
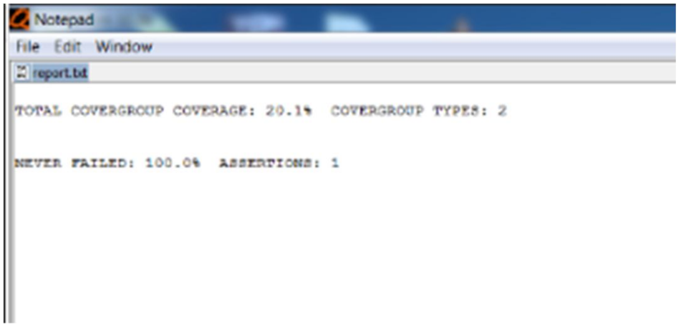


Fig.3. RTL view of DPRAM

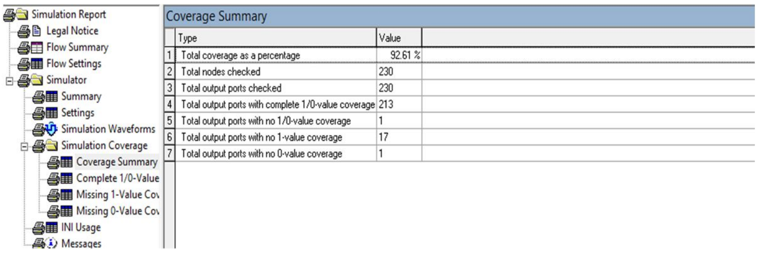
4) *Coverage Output:* Figure 4and 5 gives the coverage output.



```
TOTAL COVERGROUP COVERAGE: 20.1%   COVERGROUP TYPES: 2

NEVER FAILED: 100.0%   ASSERTIONS: 1
```

Figure 4:Coverage output.



Coverage Summary

	Type	Value
1	Total coverage as a percentage	92.61 %
2	Total nodes checked	230
3	Total output ports checked	230
4	Total output ports with complete 1/0-value coverage	213
5	Total output ports with no 1/0-value coverage	1
6	Total output ports with no 1-value coverage	17
7	Total output ports with no 0-value coverage	1

Figure 5: Coverage Result.

IV. UVM BASED VERIFICATION ENVIRONMENT

To adjust testbench should expanding configuration complexities UVM advanced should create a test bench, with the goal that a testbench could make produced in the same run through concerning illustration those outline itself. Hence we could say that due to a large number reasons such as unpredictable designs, time with market, superior correspondence around organizations and likewise inside the company, confirmation masters have characterized some as a relatable point classes what's more works which need aid certainly needed will manufacture test benches .UVM may be a confirmation procedure for utilitarian confirmation. UVM might have been made Accellera dependent upon Open Verification Methodology (OVM) form 2. 1. 1. UVM principally employments simulations with would practical confirmation of the advanced fittings. UVM environment need Usable Verification Component(UVC), it will be acknowledged as a technique of coverage-driven verification(CDV) . CDV combines self-checking testbench and test case generation, is scope measurements to decrease the period that is required to check plan .Verification part may be instantiated also arranged concerning illustration for every prerequisite. The UVM class library gives every last one of building obstructs we have should fast create well-constructed, reusable, confirmation segments. The UVM library comprises of build classes, a large number utilities.

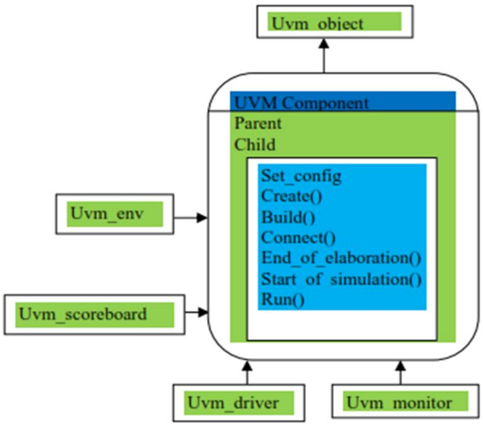


Figure.4. UVM Environment.

A. UVM Testbench Design.

A UVM based testbench need three fundamental parts

1) Main module which instantiate the DUT, testbench and the interfaces on convey the middle of testbench parts and the DUT.
2) Testbench holds every last bit UVM component UVCs. UVCs would reusable segments the individuals could make developed of the prerequisite. Testbench additionally holds register models
3) Test situation piece comprises of questions for UVM classes which need aid inherited starting with its base library. Progression for UVM earth that need been constructed may be likewise indicated clinched alongside figure. 5. The testbench need a number for population question that would associated hierarchically.

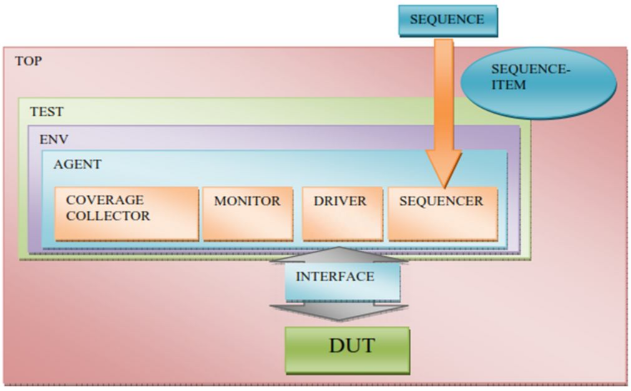


Fig.5. UVM Environment.

V. CONCLUSION

The SystemVerilog verification environment produced alongside complete stream about verification need been examined. The Different classes for driver monitor, stimulus, environment and so on. Also modules alternately projects aggravated bring been aggregated is simulate and the outputs watched need aid demonstrated. Environment made totally wraps the DUT is utilitarian more assertion based coverage need been found. Assertion coverage found will be 100% and covergroup coverage may be discovered as 20. 1%. The overall coverage may be discovered to make 100%. The SystemVerilog environment formed need been broadened should UVM toward calling the base class library also different confirmation segments. The UVM built nature's domain formed need been talked about for a little talk about UVM.

REFERENCES

[1] Muehlegg, Franz, and Alfred Schuetz. "A highly flexible dual-port-RAM compiler." In [Proceedings] EURO ASIC90, pp. 277-281. IEEE, 1990.

[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] 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.

[4] 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.

[5] Jain, Abhishek, Dr Gupta, Sandeep Jana, and Krishna Kumar. "Early development of UVM based verification environment of image signal processing designs using TLM reference model of RTL." arXiv preprint arXiv:1408.1150 (2014).

[6] 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.

[7] Amin, Alaaeldin A., Azzam A. Hamzah, and R. E. Abdel-Aal. "Generic DFT approach for pattern sensitive faults in word-oriented memories." IEE Proceedings-Computers and Digital Techniques 143, no. 3 (1996): 199-202

[8] 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.

[9] Park, Sungju, Donkyu Youn, Taehyung Kim, Sangwon Kang, Heekuk Oh, Kyunggoo Doh, and Young Shik Moon. "Microcode-based memory BIST implementing modified march algorithms." JOURNAL-KOREAN PHYSICAL SOCIETY 40 (2002): 749-753.

[10] Gao, Xinyan, Ning Zhou, Jinzhao Wu, and Dakui Li. "Wu’s characteristic set method for SystemVerilog assertions verification." Journal of Applied Mathematics 2013 (2013).

[11] Cheng, Li-Bo, Francis Anghinolfi, Ke Wang, Hong-Bo Zhu, Wei-Guo Lu, and Zhen-An Liu. "A testbench research based on UVM for ABCStar." In 2016 IEEE-NPSS Real Time Conference (RT), pp. 1-3. IEEE, 2016.

[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] 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.

[14] Surepeddi, Ravi. "System Verilog for Quality of Results (QoR)." In 9th International Symposium on Quality Electronic Design (isqed 2008), pp. 460-464. IEEE, 2008.

[15] Khalifa, Khaled, and Khaled Salah. "An RTL power optimization technique based on System Verilog assertions." In 2016 IEEE 7th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), pp. 1-4. IEEE, 2016.

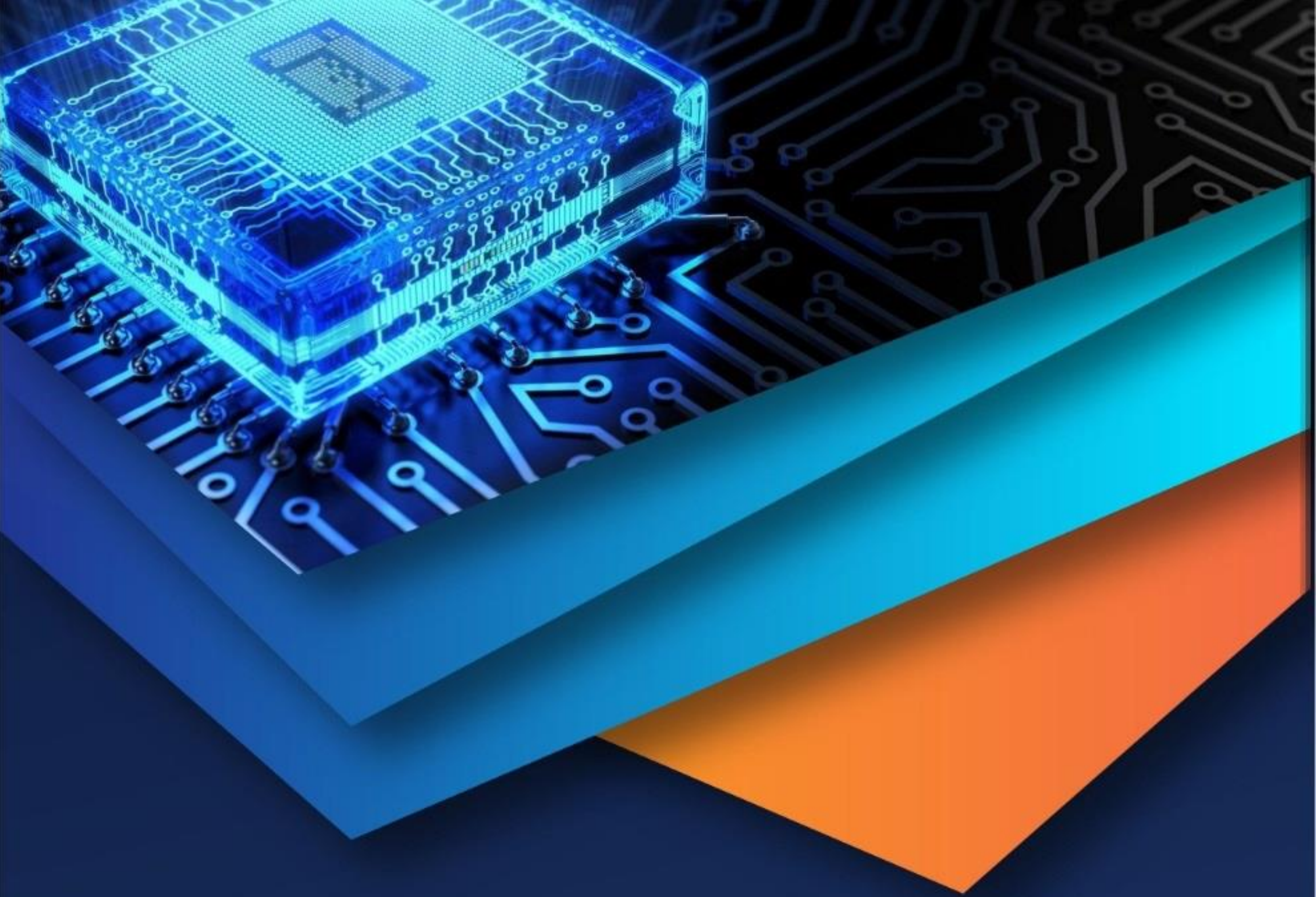
[16] Li, Yangyang, Wuchen Wu, Ligang Hou, and Hao Cheng. "A study on the assertion-based verification of digital IC." In 2009 Second International Conference on Information and Computing Science, vol. 2, pp. 25-28. IEEE, 2009.

[17] Doshi, NilayJayesh Kumar, Sheetal Suryawanshi, and Gardas Naresh Kumar. "Development of generic verification environment based on UVM with case study on HMC controller." In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 550-553. IEEE, 2016.

[18] 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.

[19] 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.

[20] 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.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

TOGETHER WE REACH THE GOAL

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY