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.2019.5339

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: 7.177
Volume 7 Issue V, May 2019- Available at www.ijraset.com

Verification Environment of Dual Port RAM: A Review

Aditi[1], Rahul[2], Pawan Kumar Dahiya[3]
[1, 2, 3]*ECE Department (VLSI Design), Deenbandhu Chhotu Ram University of Science And Technology, Murthal, Sonepat, Haryana(India)*

Abstract: ***The analysis and verification of dual port RAM give an important role in complex designs. In this paper we discuss the altered alignment of analysis and their ambiance architecture of dual port memories. Traditional analysis techniques application verilog abridgement of adaptability of reusable ambiance of analysis the system verilog analysis overcomes the verilog results. It uses the assertion based analysis to optimized captivated ability of RTL designs. This type of analysis give accessible way to absolute the designs with the help of some steps. In this paper verification environment using system verilog accomplishing of UVM for DPRAM will altercate and again analyze the after-effects of environments.***
Keywords: system verilog, UVM, environment, TLM, assertion.

I. INTRODUCTION

On late plans of multiprocessors system dual port ram have been broadly utilized to the reason for correspondence furthermore information offering. Dual port RAM would unique in relation to single port ram by those characteristics for two units alternately ports could make accessed all the while [1].Today, for fast advancement of coordinated circuits, the unpredictability of the advanced IC designs may be increasing, furthermore an ever increasing amount was troublesome with check. The workload about confirmation need accounted to 70% to 80% of the whole configuration what's more improvement. Improvement from claiming confirmation technique or verification techniques makes it workable to complete what's more successfully move forward those confirmation effectiveness [2]. For profound submicron technology, System-On-Chip (SOC) results require a high-sounding with low power. The requests to multi-port ram keep on expand to suit high-sounding interchanges and image transforming. The multi-port ram will be suitableness for parallel operation what's more enhances those downright chip execution. In spite of the fact that those memory access velocity (the number about clock cycles) enhances for expanding amount forget ports of the multi port RAM, its region punishment likewise builds for the number for ports. Consequently, a multi-port ram will be utilized especially higher velocity register files to an information way or concerning illustration cushion memory for a feature picture processor motor. Inevitably examining extending DPRAM ability, the occupation about chip measure expands, something to that effect a higher thickness of the DPRAM might be distinctly required. Already, general, the unit-cell size of the double port RAM is seeing twofold also as gigantic in like manner that for single-port RAM to date. In spite of fact that those region punishment need been lessened toward the new design structure. This will be the reason unit-cell for DPRAM need eight transistors same time that about SPRAM need six transistors [3]. With advancement for IC manufacturing methodology technology, conventional outline strategy for processor can't fulfil the request for designing complex systems. It will be comprehensively acknowledged that pushing ahead the abstractive layer which might be all the more near the planner's reasoning is a creative extension among structure and production innovation [4]. Design complexity will be at any point expanding with multimode, factual timing analysis, multi-core execution built sort of outlines. The energy outline plan on will be brought will forthright of outline phase in view of low control with respect to chip. Whatever design procedure which facilitates alternately enhances configuration nature for outcomes may be dependably a help also welcome should design engineers[5]. Design verification may be recognized on a chance to be the a large portion vital procedure in chip design stream similarly as 70 percent of occasion when may be went through around plan confirmation. Since the verification methodology verifies accuracy, it may doesn't worry about whatever accessible additional design code that doesn't impact the arrangement exactness. This extra structure code may cause eating up extra additional wasted power concerning delineation this code will be not in the design judgments in any case it doesn't impact the setup precision. Therefore, architectures with low force utilization ended up the significant hopefuls to outline about microprocessors and system design parts [6]. In complex IC plan, check takes something like 70% of the structure make period. For fast IC structure, this bit will extend at present. In this circumstance, should build those effectiveness about verification, lessening the chance utilization for confirmation stage, is significant to speed those entirety improvement procedure up [11].

Functional verification is the methodology from claiming accepting those plan such-and-such every last one of configuration determinations. Here might be a few amount for test situations made to show that the expectation from claiming outline under test will be well-preserved done its employment. The constrained random verification are complimented with regular test bench with test difficult to compare test situations for random stimulus[12]. Likewise confirmation is currently recognized concerning illustration those bottleneck for any unpredictable VLSI outline. So moving forward the confirmation effectiveness will be must. There need help two dimensions about check, IP-level verification Also SOC-level verification. To IP level verification, we prerequisite should check reason. To SOC-level verification, we have will check the availability. A host controller is keen on right and accomplishment for correspondence with the gadget, high information through-place and in sparing force. [7]. Verification particular architects must guarantee every last one of properties of the IC get executed appropriately in front of creation stage. Functional verification gives a considerable measure from claiming reductions and is an incredible help in the field for IC plan .The functional verification methodologies includes VMM, OVM and UVM [8].

This paper is sorted out as pursues. In Section 2, we talk about the framework verilog condition of the double port RAM. In the ensuing area, we present the assertion of system verilog. In Section 3, we explain the UVM environment of verification. We also present the different features of UVM .In Section 4 we compare the both verification i.e system Verilog and UVM environment .we also discuss the both environment in the conclusion in section 5.

II. SYSTEM VERILOG ENVIRONMENT

System Verilog is the first unified hardware description and verification language, might have been created by Accellera furthermore could make seen as a development about verilog with included profits of assertions, coverage driven verification, constrained random verification. System Verilog may have been gotten similarly IEEE standard 1800-2005 initially in 2005, and the latest adjustment is IEEE standard 1800-2009, the present adjustment. System Verilog (SV) wide help exists in check gadgets viz. It might be conceived that SV will be used to arrangement tape outs rapidly concerning illustration a number configuration houses began utilizing SV particular RTL constructs for system outlines directing, including large amounts about plan information abstractions to different configuration requisition keeping in perspective of verification support. System Verilog will be a bound together configuration also confirmation language for System On Chip (SOC) outlines. On adapt up with expanding configuration complexity, SV need exactly exceptionally of service constructs should point at configuration [5].

SystemVerilog is an extraordinary fittings verification language will make utilized within capacity confirmation. It gives those high level information structures accessible in object-oriented languages, for example, C++. These information structures empower a larger amount from claiming reflection and displaying from claiming intricate information sorts. SystemVerilog additionally gives constructs vital to displaying equipment ideas, for example, cycles, tri-state values, wires, only similar to Verilog hardware language. With the goal SystemVerilog might be used to mimic the Hardware Description Language (HDL) outline also check them eventually large amount test case[13]. Layered test bench will be used for eccentric blueprint of the plans. System Verilog may be that's only the tip of the iceberg supportive to the verification transform. [14]. Figure 1 show the environmental view in System Verilog .

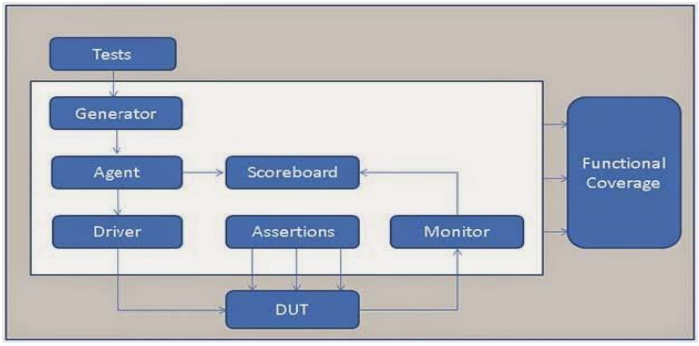


Fig.1. system Verilog environment of dual port RAM

1) *Top:* This is the top most file, which connects the DUT and test bench. It import all packages of the dual port RAM.
2) *Test:* It call the all functions without require DUT and it is responsible for the configure the test bench, initiate the test bench and initiate the stimulus driving.
3) *Environment:* All the classes are written in this and this is the container class for grouping higher level components.

4) *Generator:* It generate different types of packets (group of data) and transfer all transactions from transaction class to the drivers.
5) *Agent:* Agent include the generator, driver and monitor classes.
6) *Driver:* With the help of them data is transfer from packet level to pin level ad it receive the data from generator.
7) *Scoreboard:* It receive data from monitor and compares the data and give response from the DUT.
8) *Assertion:* It is basically statement of fact or claim of truth . They are the properties of design that are supposed to be true. They allow to communicate information to the test bench. It is basically a checker, that can be utilized in transient space with complex planning and to distinguish undesired conduct of DUT.
9) *Monitor:* It collect from response from DUT and monitors the inputs.
10) *Checker:* It check the DUT functions (coverage) and data output.

The data is transfer from pin level to packet level and vice versa with the help of interface and mailbox. Figure 2 show the respective design of mailbox and interface.

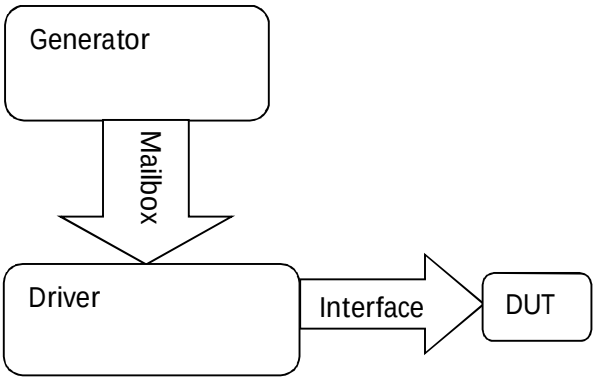


Fig.2. Packet, Interface and Mailbox

11) *Interface:* It transfer the pin level transactions from driver to DUT.
12) *Mailbox:* It transfer the packet level transactions from generator to driver. It is a medium to transfer memory locations.

A. System Verilog Assertion(SVA)

Attestation Based Verification (ABV) will be such a technique, which joins affirmation, entertainment moreover formal methodologies of the acknowledged limit assertion. A percentage properties for duration of the time prerequisites would not difficult should make depicted for verilog, There need aid a significant number institutionalized declaration languages for example, PSL, SVA what's more declaration library for which those attempt for describing a configuration property gets a great part less demanding and proficient. Recently, System Verilog Assertions (SVA) extensively utilized concerning illustration detail about assertions inside the design, empowering the test system with check those assertions throughout simulation [15]. Two distinct classes of assertions which are warning statements and fault declarations.

1) *Warning Assertions:* They are in charge of transferable included code which doesn't influence the design real however expends included waste power. These statements cautioning kneads which suit charge exhortation to be trailed by the craftsman to adjust and upgrade the engineering figure and subside the design devoured control.
2) *Fault Assertions:* They are in charge of transmittable the wrong code which abuses the engineering diagram and influence the engineering or structure accuracy. Accordingly, these declarations are utilized as guided investigation cases to confirm the definiteness or rightness of mood the design signals and genuine them

III. UVM ENVIRONMENT

Universal Verification Methodology (UVM) may be presently those The greater part prevalent verification technique. It might have been released with respect to 28 Feb 2011 with the exertion from the distinctive stakeholders in the business determined by Accellera. Its principle objective will be will move forward test bench reuse. UVM based test bench, created of reusable verification parts performs those verification during a speedier rate. A VIP (Verification IP) plan is created, which may be the foundation for the entirety verification transform. Test cases would composed will check those different purpose of the DUT. Principle highlights from guaranteeing which including:

1) Automatic test age with unique obliged randomized boost to recreate the genuine information input movement.
2) Automatic gathering attractive practical inclusion message to gauge advance and distinguish non-practiced usefulness.
3) Automatic looking at yield after effect of DUT with reference display, which has a similar capacity as DUT composed by other abnormal state language to check the remedy of the plan.

UVM gives a layer to track registers substance from claiming DUT, which is known as register model. The register model provides for a advantageous method for simulating the activity for composing furthermore perusing to registers [9]. UVM will be in a general sense produced using OVM (Open check Methodology) Also VMM 1.2 (Verification Methodology Manual) what's more it meets desires on article turned changing. As indicated on progression uvm_void and uvm_object may be guardian population to every last one of classes. Every last one of verification parts could make extends starting with uvm_component population. UVM base class library so give announcing offices to disappointment report and general exchange report. UVM additionally comprise of inherent focal factor which expedites more control object assignment [16].

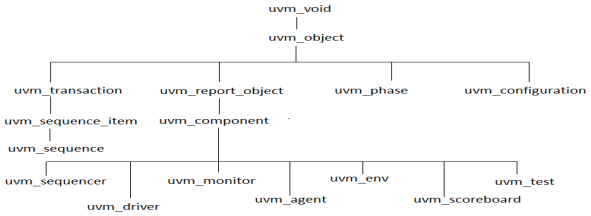


Fig.3. UVM hierarchy diagram

The main benefits of UVM are:

Wide-running form class library support, can perform inclusion driven interest irregular verification, completely bolstered by means of genuine gadget around sellers UVM based test seat serves compelled arbitrary info boosts and it additionally helps in achieving verification objectives by giving practical inclusion subtleties. Those main roles of a test situate are: - produce data improvements, drive the boosts to Design Under Verification (DUV), screen reaction, weigh to rightness, measure scope focuses to centre headway of verification [12]. UVM built test bench structural engineering is an arrangement for UVM parts for example, such that UVM test, sequencer, driver, monitor, agent, scoreboard, environment. These are the principle blocks of the test bench with enable UVM built verification [8][10].

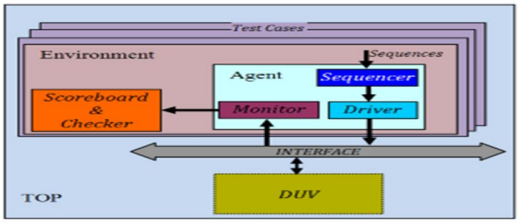


Fig.4. UVM environment of dual port RAM

a) *Sequencer:* The sequencer create data and driver sends it of the DUT. Sequencer part class may make created from base class.
b) *Scoreboard:* The scoreboard looks at the got information or reaction against the normal one. scoreboard is sufficiently conventional to permit all together looking at and of request checking. Besides, it underpins UVM get back to. UVM call backs are utilized to include the new abilities, without making a tremendous OOP chain of command.
c) *Monitor:* The screen tests the information and reactions.
d) *Test Bench Environment:* A test seat has been created with the parts and it can fit for being utilized again even with the expansion in multifaceted nature of plan.
e) *Driver:* The driver of a particular arrangement stretches out from an uvm_driver construct library. The driver part gets enter from the sequencer and executes the task.
f) *Agent:* An agenize part is comprised from claiming subcomponents for example, driver, monitor, sequencer. This agenize could be worked done whichever of the mode i.e. animated or indifferent modes.
g) *Top:* The top class contains the whole verification classes of UVM test bench.
h) *Test:* It tends to be stretches out from base class. Object of condition class and handle of interface module can be made in the fabricate period of the class.
i) *DUV:* It contains the design verification under the observations.

A. Test Bench Reusability

Reusable examination or verification stage can better decrease the analysis time and advance the productivity. UVM examination stage can naturally total the Transactions to be driven, quiet and looked at. Exchanges are produced and sent by Sequence. Succession, associated from sequence class , is modified from driver and included mechanical assembly. So grouping does not accord to mechanical assembly of UVM examination stage [2].

B. Test Case Reusability

The reusability may be reflected in the reusable test cases. On UVM platform, the reuse for test situations will be reflected in the reuse of grouping. There would two principle types in the following, you quit offering on that one may be inheritance reuse, and the different is combination reuse. Inheritance reuse alludes of the inheritance from claiming class, reusing the qualities and routines from claiming past classes. It implies that might over-burden a existing arrangement on produce another arrangement. It principally modifies existing constraints, alternately includes new control qualities will produce diverse classes[2].

C. Transaction Level Model(TLM)

TLM speaks to a great compromise the middle of displaying exactness furthermore recreation velocity. It can be a chance to be assembled for or without timing delay concerning illustration a programmer's view, or a chance to be based concerning illustration an exact cycle model. The principle reason for existing about TLM may be on unique out correspondence around the modules by alleged transactions: the information or off chance between two parts of a modelled and mimicked framework. TLM need been ended up being a quick also proficient manner for framework plan. It fills the hole between absolutely utilitarian portrayals and RTL models. TLM executes the correspondences the middle of module squares toward method for work alternately task calls[4].

D. UVM Phases

On UVM construction modelling built test benches, every last one of parts furthermore transactions happen to predefined stages. Those different stages over UVM are:

1) *Raise Phase:* The manufacture stage runs top down style. The larger amount segments can receive if should manufacture the small segments dependent upon random setup.
2) *Associate Phase:* The TLM ports need aid associated for interface period. It runs bottom dependent upon way.
3) *End of Elaboration Stage:* This stage is utilized to check relationship for test seat, to print topology. Fine change from asserting test situate is conveyed in this stage.
4) *Begin for Simulation Phase:* The TLM ports need aid joined for join stage. It runs bottom dependent upon way.
5) *Run Phase:* It may be the main period characterized similarly as an undertaking as it expends chance. Reproduction begins in this stage.
6) *Extract Phase:* The sum information need aid concentrated from the test bench segments.
7) *Check Stage:* This period checks to unforeseen states from test bench portions.
8) *Report Stage:* The scoreboard likewise extraordinary checkers report the multiplication comes about are outfitted in this period.
9) *Last Phase:* The simulation will be regarding should limit in this stage. It will be used to print last messages.

IV. COMPARISON BETWEEN BOTH ENVIRONMENTS

From above all discussion we can say that the UVM is a methodology and system verilog is a language. We can not compare both of them practically but in some features we can differ them. The table 1 shows the different features of the environments.

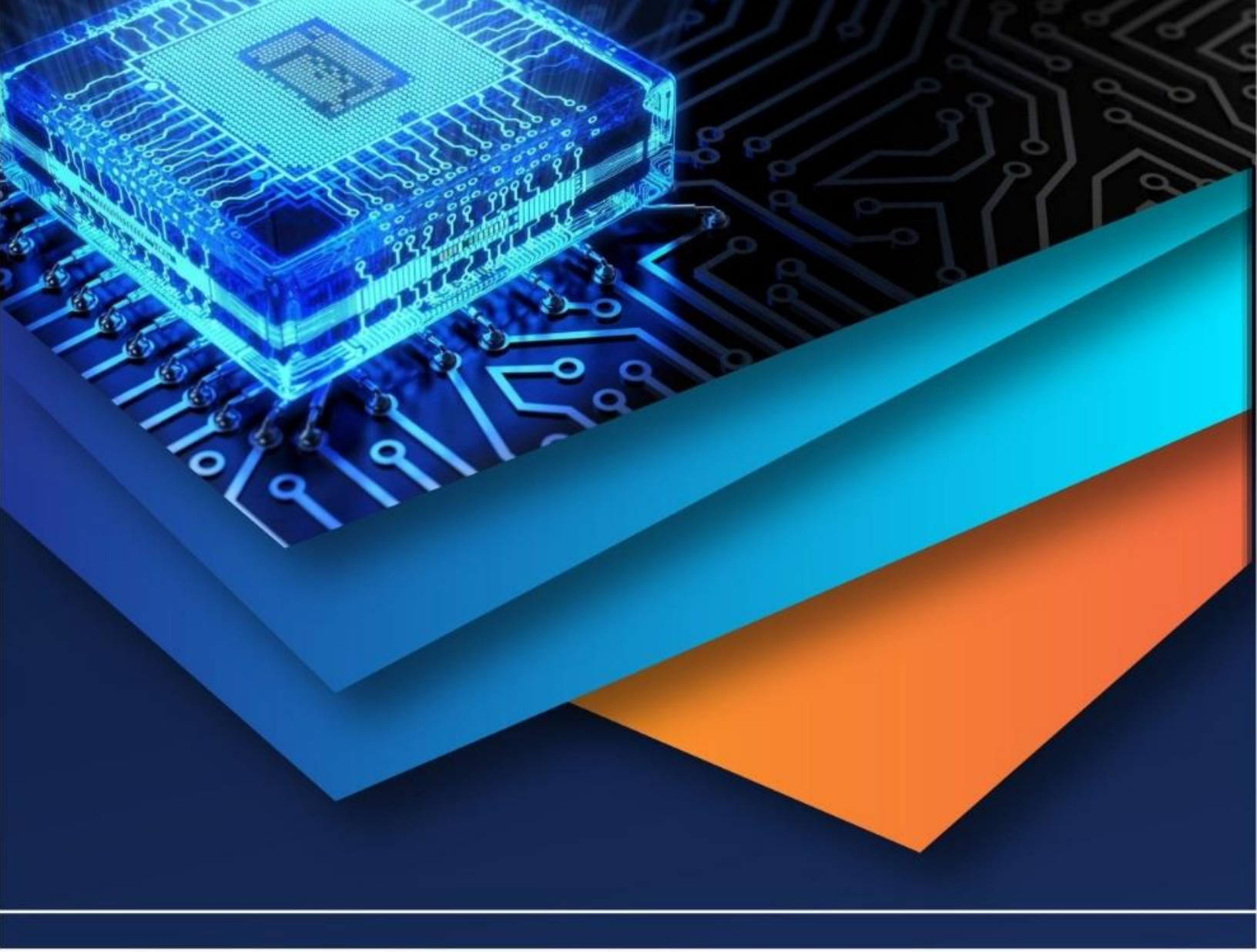
Features	System Verilog	UVM
Simulation statistics	More	Less
Debugging	No inbuilt debugging	Inbuilt simple debugging
Reusability	No reuse of test bench	Reuse of test bench due to flexibility of structure
Synchronization	there is no TLM in system Verilog	TLM is used in UVM to synchronize the design
Test bench structure	OOP is used and simple	OOP is used but complicated
Coding	Easy	Little difficult as compared with system Verilog
Dependency	Depend on Verilog language	No dependency

V. CONCLUSION

In this paper, system Verilog based verification and UVM based verification for dual port ram may be examined. Those examination demonstrate the features, verification process, done both methodologies. The test bench building design will be made dependent upon UVM technique for confirmation. The UVM consumed those preferences about OVM, VMM Furthermore other methodologies. Those UVM makes confirmation that's only the tip of the iceberg proficient furthermore most extreme reusability.

REFERENCES

[1] Yokoyama, H., H. Tamamoto, and Xiaoqing Wen. "Built-in random testing for dual-port RAMs." In Proceedings of IEEE International Workshop on Memory Technology, Design, and Test, pp. 2-6. IEEE, 1994.
[2] 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.
[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] 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.
[5] Surepeddi, Ravi. "System Verilog for Quality of Results (QoR)." In 9th International Symposium on Quality Electronic Design (isqed 2008), pp. 460-464. IEEE, 2008.
[6] 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.
[7] 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.
[8] Elakkiya, C., N. S. Murty, C. Babu, and Gaurav Jalan. "Functional Coverage-Driven UVM Based JTAG Verification." In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), pp. 1-7. IEEE, 2017.
[9] 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.
[10] Srinivas, M. B. R., and Sarada Musala. "Verification of AHB_LITE protocol for waited transfer responses using re-usable verification methodology." In 2016 International Conference on Inventive Computation Technologies (ICICT), vol. 3, pp. 1-3. IEEE, 2016.
[11] 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
[12] 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.
[13] 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.
[14] 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.
[15] 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.
[16] 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..
[17] 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).
[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] Trivedi, Jamdagni C., and Hardik Trivedi. "Verification environment for PCI target and Wishbone master interface module using system verilog." In 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 981-984. IEEE, 2016.
[20] Jia, Qingzhong, and Xingdou Wang. "Research on communication technology of complex system based on FPGA dual-port RAMs." In 2016 22nd International Conference on Automation and Computing (ICAC), pp. 32-36. IEEE, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY