Analyzing AES Verification: A Comparative Study of UVM and Cocotb Approaches

N Susithra
Department of ECE
PSG Institute of Technology and Applied Research
Tamilmadu, India
susithra@psgitech.ac.in

Santhosh Kanna S
Department of ECE
PSG Institute of Technology and Applied Research
Tamilnadu, India
santhosh20102002@gmail.com

Sridhar Karthik K
Department of ECE
PSG Institute of Technology and Applied Research
Tamilnadu, India
sridharkarthik12@gmail.com

Naresh Raja T
Department of ECE
PSG Institute of Technology and Applied Research
Tamilnadu, India
naresshraja795@gmail.com

Yaswant V
Department of ECE
PSG Institute of Technology and Applied Research
Tamilnadu, India
yaswantv1110@gmail.com

Rajalakshmi K
Department of ECE
PSG College of Technology
Tamilnadu, India
krl.ece@psgtech.ac.in

Abstract—**The paper focuses on a comprehensive comparative study of two prominent verification methodologies: Universal Verification Methodology (UVM) and Cocotb. UVM is a well-established verification methodology based on System Verilog, which provides a structured framework and reusable components for verifying digital designs. It has been the industry standard for years and is known for its extensive support and mature ecosystem. On the other hand, Cocotb is a newer, Python-based verification framework that leverages the flexibility and ease of use of Python to create test benches and run simulations. This study analyzes the strengths and weaknesses of both methodologies in terms of productivity, simulation time, learning curve, and industry adoption. It also examines the potential of Cocotb to complement or replace traditional UVM-based verification in certain scenarios, considering the growing popularity of Python in the engineering community. AES is a symmetric encryption algorithm widely adopted as standard for securing electronic data. The choice of AES for verification is motivated by its robust security features and its significance in the industry. Cocotb takes longer to simulate than UVM, but its superior functional coverage makes it a strong choice for projects focused on thorough verification and testing.**

Keywords: **Universal Verification Methodology, Cocotb, AES.**

I. Introduction

In the real time of hardware verification methodologies, engineers face the critical task of selecting between Cocotb (Coroutine-based Cosimulation TestBench) and UVM (Universal Verification Methodology) to ensure the correctness and efficiency of complex hardware designs. This study aims to conduct a thorough comparative analysis of Cocotb and UVM, employing the widely-used Advanced Encryption Standard (AES) algorithm as a case study.

The AES algorithm, renowned for its importance in modern cryptography, poses a significant challenge for hardware verification due to its intricate design and stringent security requirements. Engineers rely on robust testing methodologies to validate AES encryption hardware implementations effectively. Cocotb characterized by its Python-based framework and open-source nature, offers an attractive alternative to conventional verification methodologies. Its simplicity, flexibility, and accessibility make it appealing for engineers engaged in rapid prototyping, experimentation, and agile development. Moreover, Cocotb's integration with Python's extensive library ecosystem enhances its capability to verify complex hardware designs, such as AES encryption circuits, with ease. On contrast, UVM stands as a pillar of traditional verification methodologies, recognized for its structured approach and comprehensive libraries tailored for ASIC and FPGA verification.

With standardized methodologies and strong industry support, UVM remains a preferred choice for large-scale verification projects, offering scalability, reusability, and robustness in verification environments. Through an in-depth examination of Cocotb and UVM in the context of AES encryption verification, this study seeks to highlight their respective strengths, weaknesses, and suitability for modern hardware verification challenges. By considering factors like usability, scalability, performance, and industry adoption, engineers can make informed decisions regarding the selection of the most appropriate verification methodology for their projects.

In the subsequent sections of this study, we delve into a comparative analysis of Cocotb and UVM, utilizing parameters such as time to build, code coverage, functional coverage, datatypes, inbuild libraries and automation. This paper aims to offer valuable guidance to hardware verification engineers navigating the complexities of verification methodology selection.

II. Literature survey

In article [1], the authors present the lightweight AES architecture with key length 256-bit for resource-constrained IoT devices by reducing the area and power consumption of the design. In paper [2], the authors introduce and investigate error detecting methods specifically designed for hardware implementation of AES. These techniques aim at detecting errors in encryption or decryption processes. In article[3], suggests enhancements to AES implementations to increase security and stability. They introduce new tricks or methods aimed at strengthening the resilience of AES against cryptographic attacks while at the same time tackling potential weaknesses and mistakes that could jeopardize the

979-8-3503-7817-7/24/$31.00 ©2024 IEEE

time to build is something to be taken note of. Analyzing the coverage is much easier.

Data types: UVM is written in system Verilog whereas pyuvm is written in python. Python surpasses SystemVerilog in data type richness and flexibility, boasting a wide array of built-in and library-defined data structures like lists, dictionaries, tuples, and sets. Unlike SV, Python's dynamic typing system enables variables to switch types dynamically, enhancing adaptability in handling diverse data scenarios.

Libraries: With its extensive libraries for numerical computation, data analysis, and visualization, Python emerges as a versatile choice for verification tasks, offering more than just hardware description capabilities.

Automation: Running multiple testcases at the same time is much easier in python than in UVM. Because, by writing a simple python script can push one testcase after another in to the DUT. In UVM, running testcases requires manual effort without a python automation script.

IV. Conclusion

The project findings highlight Cocotb's superiority over UVM in terms of testbench construction efficiency and coverage analysis. Cocotb's advanced features streamline event synchronization, reducing build time significantly. Achieving a coverage of 90.38% across 1000 test cases, Cocotb outperforms UVM 87.28%, showcasing its effectiveness and ease of coverage analysis. Python's data type richness and flexibility in pyuvm surpass SystemVerilog, enhancing adaptability in handling diverse scenarios. With extensive libraries for numerical computation and data analysis, Python proves to be a versatile choice for verification tasks, offering enhanced productivity beyond hardware description capabilities. Automation in Python simplifies running multiple test cases concurrently, providing a streamlined approach compared to manual efforts in UVM. By leveraging Python scripts, test case execution becomes seamless, improving efficiency and productivity in verification processes. Python's automation capabilities enhance the overall effectiveness of verification tasks.

References

[1] Karim Shahbazi, Seok-Bum Ko, ”Area-Efficient Nano-AES Implementation for Internet-of-Things Devices”, : IEEE Transactions on Very Large Scale Integration (VLSI) Systems (Volume: 29, Issue: 1, January 2021), dob: 10.1109/TVLSI.2020.3033928. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] C. H. Yen and B. F. Wu, "Simple Error Detection Methods for Hardware Implementation of Advanced Encryption Standard", IEEE Trans. Computers, vol. 55, no. 6, pp. 720-731, June 2006, dob: 10.1109/TC.2006.90.

[3] Mirajkar, Devyani Madhukar, "Design and Verification of a Pipelined Advanced Encryption Standard (AES) Encryption Algorithm with a 256-bit Cipher Key Using the UVM Methodology" (2018). Thesis. Rochester Institute of Technology.

[4] H. Liang, N. Tan, Y. Ren, W. Hu, J. He and J. Xia, "Python Based Testbench for Coverage Driven Functional Verification", 2022 7th International Conference on Integrated Circuits and Microsystems (ICICM), pp. 361-365, 2022.

[5] Lin Zhu1, Ligang Hou1, Qiuyun Xu1, Jingsong Zhi1, Jinhui Wang , “A UVM-based AES IP Verification Platform with Automatic Testcases Generation”, 2016 International Conference on Engineering and Advanced Technology (ICEAT-16).

[6] Aruna B, Naveen K B, Anandaraju M B, “Design and Verification of AES Algorithm using Verilog”, Journal of Fundamental & Comparative Research Vol. VII, No. 10(I): 2021.

[7] Ajinkya Sunil Naik, Dr.G.V.Maha Lakshmi, “Design and Verification of Encryption of AES Algorithm”, 2018 IJCRT | Volume 6, Issue 1 March 2018.

[8] H. Liang, N. Tan, Y. Ren, W. Hu, J. He and J. Xia, “Python Based Testbench for Coverage Driven Functional Verification”, 2022 7th 41 International Conference on Integrated Circuits and Microsystems (ICICM), pp. 361-365, 2022, doi: 10.1109/ICICM56102.2022.10011364.

[9] M. Trapaglia, R. Cayssials, L. De Pasquale and E. Ferro, "Flexible software to hardware migration methodology for FPGA design and verification", Proc. 10th Southern Conf. Program. Log. (SPL), pp. 39-44, Apr. 2019, doi: 10.1109/SPL.2019.8714377.

[10] Marek Cieplucha and Witold A. Pleskacz, “New Constrained Random and Metric-Driven Verification Methodology using Python”.

[11] Amr Moursi, Romaisaa Samhoud, Yaseen Kamal, Mazen Magdy, Sameh El-Ashry, Ahmed Shalaby, “Different Reference Models for UVM Environment to Speed Up the Verification Time”, 2018 19th International Workshop on Microprocessor and SOC Test and Verification (MTV), dob: 10.1109/MTV.2018.00023.

[12] Khaled Salah, A UVM-based smart functional verification platform: Concepts, pros, cons, and opportunities”, 2014 9th International Design and Test Symposium (IDT), dob: 10.1109/IDT.2014.7038594.

[13] N B Harshitha, Y G Praveen Kumar, M Z Kurian, “An Introduction to Universal Verification Methodology for the digital design of Integrated circuits (IC’s): A Review”, 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), dob: 10.1109/ICAIS50930.2021.9396034.

[14] Cong Liu, Xinyu Xu, Zhenjiao Chen, Binghao Wang, “A UniversalVerification-Methodology-Based testbench for the coverage driven functional verification of an instruction cache controller”.

[15] Esraa M. Hamed, Khaled Salah, Ahmed H. Madian, Ahmed G. Radwan, “An Automated Lightweight UVM Tool”, 2018 30th International Conference on Microelectronics (ICM), dob: 10.1109/ICM.2018.8704037.

[16] https://docs.cocotb.org/en/stable/index.html