iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 **Issue:** XI **Month of publication:** November 2025

DOI: https://doi.org/10.22214/ijraset.2025.74959

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: 7.538
Volume 13 Issue XI Nov 2025- Available at www.ijraset.com

UVM Based RISC-V Verification Using Arduino Uno

Aarya S. Kutwal[1], Pranav D. Yadav[2], Ajit A. Ware[3]
Department of Electronics and Telecommunication Engineering, Zeal College of Engineering and Research, Pune, India

Abstract: ***This paper presents the design and implementation of a Universal Verification Methodology (UVM) based verification environment for a RISC-V processor core integrated with an Arduino Uno interface. The work demonstrates a practical approach to processor verification where testbench results are interfaced through Arduino for display and serial communication. The SystemVerilog-based UVM environment automates constrained random testing, coverage analysis, and assertion-based verification. Results indicate over 90% instruction coverage, validating the processor's functional correctness and proving UVM's efficiency for academic and industrial verification projects.***
Keywords: UVM, RISC-V, SystemVerilog, Arduino Uno, Functional Verification, Constrained Random Testing.

I. INTRODUCTION

The RISC-V architecture has become one of the most widely adopted open-source Instruction Set Architectures (ISA), enabling academic and industrial users to design and verify processors without licensing restrictions. As designs grow in complexity, the verification effort accounts for nearly 70% of total design time. This has driven the adoption of structured verification methodologies like the Universal Verification Methodology (UVM). In this paper, we present a UVM-based verification framework for a 32-bit RISC-V processor core, augmented with Arduino Uno for output visualization and serial communication.

II. LITERATURE REVIEW

Several researchers have proposed RISC-V verification environments using SystemVerilog and UVM frameworks. RISCV-DV, developed by Chips Alliance, provides a reusable instruction generator integrated with UVM agents. riscv-formal uses formal verification through the RISC-V Formal Interface (RVFI). OpenHW's CORE-V framework integrates both formal and simulation-based verification. However, very few works demonstrate integration with microcontrollers like Arduino for educational or experimental outputs.

III. METHODOLOGY

The proposed system consists of two primary domains: the Design Domain and the Verification Domain. The Design Domain includes the RISC-V core RTL, memory models, and interface modules. The Verification Domain implements UVM components including sequencer, driver, monitor, scoreboard, and coverage collector. The Arduino Uno board is used to display verification results and communicate key signals via serial output. The testbench generates constrained-random instruction sequences, drives them into the DUT, and compares results with a golden model.

IV. RESULTS AND DISCUSSION

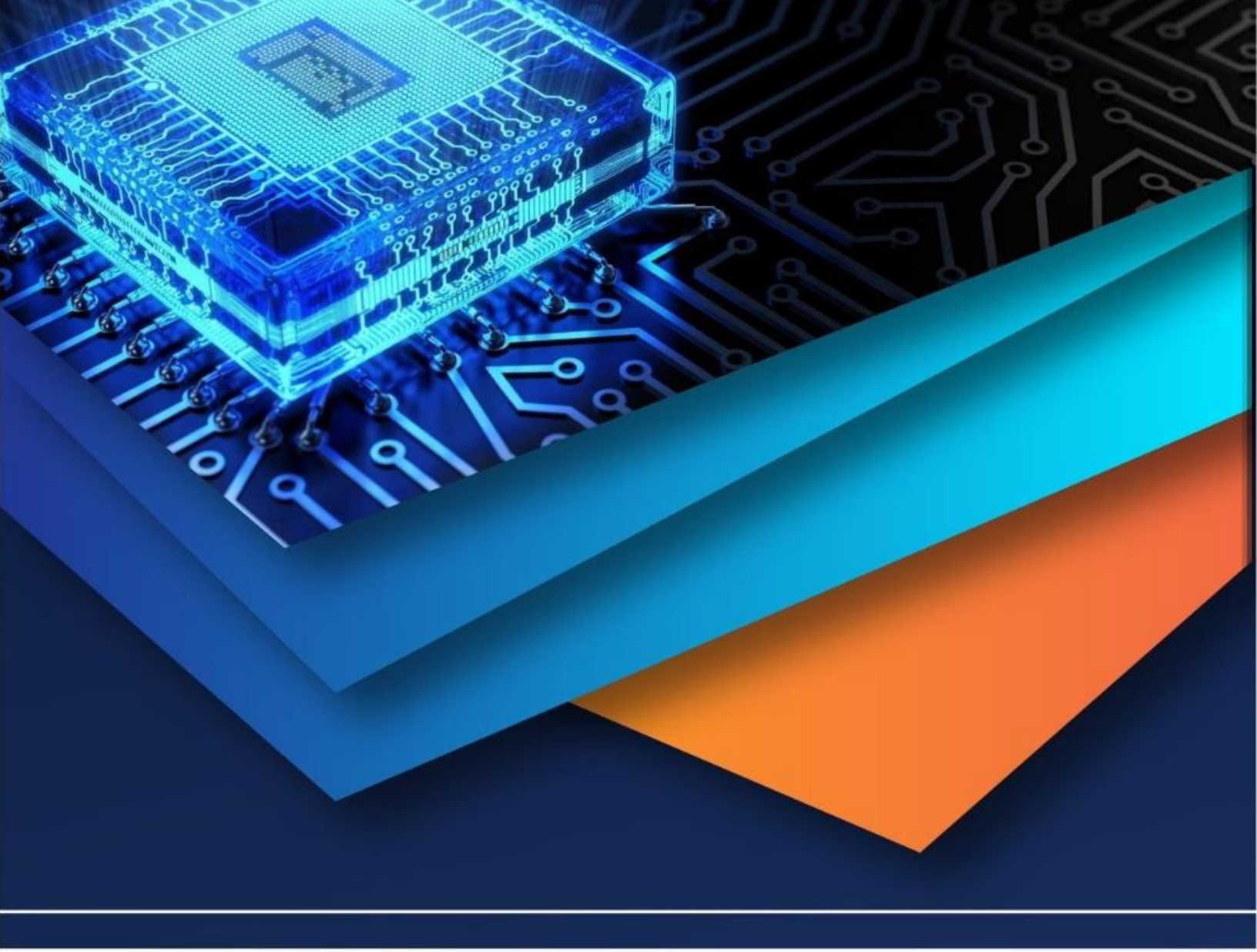
The UVM-based verification testbench successfully verified the functional correctness of the RISC-V core. Simulation was performed using Synopsys VCS, generating both random and directed tests. Functional coverage achieved over 90%, indicating high verification completeness. Arduino Uno displayed instruction execution results in real-time through serial communication, providing a physical output layer for educational demonstration. Waveforms in QuestaSim validated proper instruction decoding, ALU operations, and register updates.

V. CONCLUSION

This paper demonstrated a complete UVM-based verification environment for a RISC-V processor core integrated with Arduino Uno for result visualization. The developed testbench achieved high coverage and proved effective in detecting design bugs. The methodology can be extended to verify advanced RISC-V extensions and implement hardware testing using FPGA platforms. Future work includes integrating formal verification and machine learning-based test generation for intelligent verification flows.

REFERENCES

[1] Chips Alliance, “RISCV-DV: Open-source Random Instruction Generator for RISC-V Processor Verification,” GitHub, 2024.
[2] YosysHQ, “riscv-formal: Formal Verification Framework for RISC-V Cores,” ReadTheDocs, 2024.
[3] OpenHW Group, “CORE-V Verification Environment (CV32E40P),” OpenHW Documentation, 2024.
[4] Synopsys Inc., “VCS User Guide: UVM Simulation and Verification,” Synopsys, 2024.
[5] M. Zachariášová et al., “UVM-Based Verification of a RISC-V Processor Core,” DVCon Europe, 2023.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY