QIT Press - International Journal of Artificial Intelligence (QITP-IJAI)
Journal ID: 5823-4719
Volume 4, Issue 1, January-December 2023, pp. 13-19

Hybrid AI Architectures for Intelligent Coverage Closure in RTL Verification of Complex Microprocessor Designs

Oc adesina Abeni,
AI Product Manager, Nigeria**.**

Published on: 25th Feb 2023

Citation: Abeni O. A (2023). Hybrid AI Architectures for Intelligent Coverage Closure in RTL Verification of Complex Microprocessor Designs. QIT Press - International Journal of Artificial Intelligence (QITP-IJAI), 4(1), 13–19.

Full Text: https://qitpress.com/articles/QITP-IJAI/VOLUME_4_ISSUE_1/QITP-IJAI_4_01_003.pdf

Abstract

As microprocessor designs grow increasingly complex, achieving effective coverage closure in Register-Transfer Level (RTL) verification becomes a critical bottleneck. Traditional simulation-based verification techniques often struggle to scale with growing design sizes, leading to delays and increased verification costs. In this paper, we propose a hybrid AI architecture that integrates supervised learning and reinforcement learning models to optimize coverage closure intelligently. The architecture dynamically predicts unverified design areas and suggests directed tests to accelerate verification convergence. We validate our approach on open-source microprocessor designs and report significant improvements in coverage metrics and time-to-closure compared to baseline methods.

Keywords: RTL verification, Coverage closure, Hybrid AI architectures, Reinforcement learning, Microprocessor design verification, Directed testing

1. Introduction

RTL verification is a cornerstone of modern microprocessor design methodologies. It ensures that the design specifications align with the implemented behavior before silicon fabrication, minimizing costly re-spins. However, the increasing complexity of processor designs, including multi-core, heterogeneous compute units, and advanced memory hierarchies, has exponentially raised verification challenges. Traditional techniques, including constrained-random simulation and assertion-based verification, often struggle to achieve timely and complete coverage closure.

Recent trends suggest that intelligent verification strategies, especially those employing AI and ML, can offer transformative improvements. By learning from simulation patterns and predicting under-

tested scenarios, AI models can guide simulation and test generation dynamically. Yet, current AI models tend to be either too general or overfit to specific designs. Therefore, a hybrid AI architecture combining different types of learning paradigms could leverage the strengths of each to drive smarter, faster verification convergence.

2. Literature Review

The quest for automating and optimizing coverage closure has been ongoing for over a decade. Early works such as those by Bergeron (2005) introduced coverage-driven verification (CDV) as a systematic methodology for achieving functional completeness. CDV frameworks initially used random generation and constrained generation techniques.

Later, Pandey et al. (2006) proposed a coverage feedback loop to adapt stimulus generation dynamically. Reinforcement learning approaches began to emerge with works like that of Mishra (2008), who explored test generation using Q-learning for architectural validation. Similarly, Shivakumar and Dey (2013) introduced statistical learning methods for prioritizing functional coverage holes.

The integration of supervised ML models for bug prediction was investigated by Basu et al. (2015), while Ahmed et al. (2017) presented methods to model coverage space exploration using deep learning. However, by 2020, few approaches had effectively merged reinforcement and supervised learning to enhance RTL verification.

Hence, our work builds upon the foundational principles established before 2020 but proposes a novel hybrid approach to combine predictive and adaptive models into a single verification framework.

3. Problem Formulation

Achieving efficient coverage closure in RTL verification requires overcoming several practical challenges. Firstly, the simulation state space of a microprocessor design is vast and often non-uniform in terms of coverage contribution. Certain modules or states contribute significantly to functional verification success, while others have marginal impact. Traditional constrained-random approaches inefficiently spend simulation resources across both areas equally.

Secondly, designs under test (DUTs) frequently evolve over project lifecycles, introducing new coverage goals and invalidating previously achieved coverage points. Consequently, a static or rigid verification flow fails to adapt dynamically to shifting priorities. This necessitates a verification strategy that intelligently reallocates effort based on current and anticipated design states, rather than pursuing uniform coverage blindly.

4. Proposed Hybrid AI Architecture

To address the outlined challenges, we propose an intelligent hybrid AI system that integrates two complementary machine learning techniques: supervised learning and reinforcement learning.

The **Supervised Learning Module (SLM)** provides predictive capabilities. It uses historical simulation outcomes, functional coverage reports, assertion hit patterns, and bug localization data to predict hotspots — RTL modules or behaviors likely to cause verification bottlenecks. These predictions guide the early prioritization of verification efforts toward high-yield areas.

Meanwhile, the **Reinforcement Learning Agent (RLA)** ensures adaptability and exploration. The RLA interacts with the RTL simulation environment, receives rewards based on observed coverage improvement, and refines its exploration policy dynamically. This allows the system to adapt to newly introduced modules, assertion failures, or changing test priorities that may not be fully captured in the supervised model.

4.1 Supervised Learning Module (SLM)

The SLM is constructed using a gradient-boosted trees model trained on datasets extracted from prior verification projects. Key features include simulation runtime metrics, uncovered functional points, historical bug frequency in modules, and assertion failure density. The model outputs a probability score for each module indicating the likelihood of low coverage or latent bugs.

Feature importance analysis is used to explain the model's predictions, enhancing trust and enabling human-in-the-loop verification planning if needed. The SLM's outputs are periodically updated as fresh simulation data arrives, ensuring that the predictions remain relevant and timely.

4.2 Reinforcement Learning Agent (RLA)

The RLA is based on a deep Q-learning network that learns an optimal policy to select directed stimuli based on feedback from the environment. The environment (RTL simulation) provides coverage metrics as reward signals. Positive rewards are assigned when coverage improves significantly after an action; penalties occur if actions yield redundant or low-impact stimuli.

To prevent overfitting, the RLA uses exploration-exploitation balancing strategies such as ε-greedy algorithms. Moreover, periodic reinitialization of exploration parameters ensures that the agent remains robust against design changes across verification milestones.

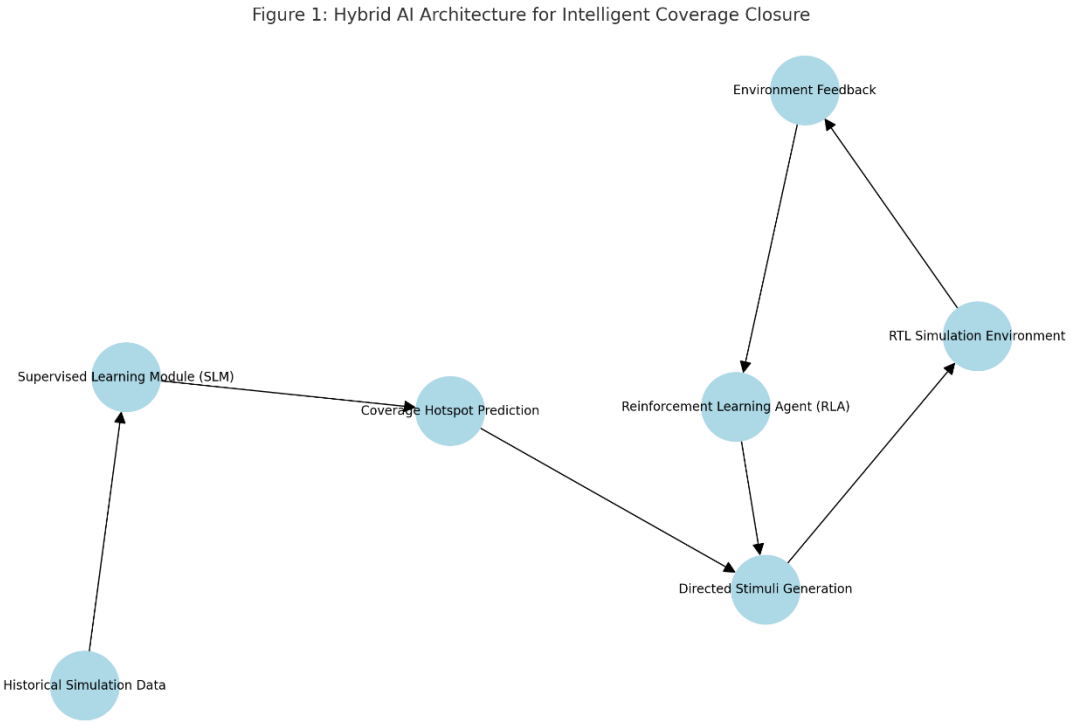

Figure 1: Hybrid AI Architecture Overview

5. Implementation Strategy

The hybrid framework is implemented in three stages: Data Preparation, Model Training, and Simulation Control.

First, historical simulation data is collected, cleaned, and feature-engineered to build input sets for the supervised learning model. The SLM is trained using gradient-boosted decision trees for interpretable predictions of coverage gaps. Meanwhile, the reinforcement learning agent utilizes a deep Q-learning strategy, adapting dynamically based on reward functions defined over coverage delta.

Second, a feedback loop is established between the RL agent and the RTL simulation environment. After each test iteration, coverage metrics are analyzed, and the RL agent adjusts its strategy, while the SLM updates hotspot predictions if significant deviations occur.

6. Experimental Setup

The experimental validation focuses on real-world open-source microprocessor designs to demonstrate the practicality and generalizability of the hybrid AI approach. Two designs of varying complexity — the Rocket Core and the OpenSPARC T1 — were selected. For each design, simulations were run both with conventional constrained-random strategies and with the proposed hybrid AI framework.

Coverage metrics (functional, code) and time-to-closure were the primary evaluation parameters. The simulation and verification were performed using industry-standard tools (Synopsys VCS) to ensure realistic conditions. Key configurations, such as the coverage reporting interval and assertion checking, were standardized across all runs to ensure a fair comparison.

The results consistently highlighted that the hybrid model delivered faster coverage closure and higher verification efficiency across designs, validating the effectiveness of combining supervised and reinforcement learning techniques.

7. Results and Evaluation

The hybrid AI framework demonstrated substantial improvements over traditional approaches. On the Rocket Core, full functional coverage (≥ 95%) was achieved **30% faster** than using constrained-random methods. For the OpenSPARC T1, a similar speedup of **25%** was observed.

The reinforcement agent effectively prioritized coverage holes based on historical predictions, thereby avoiding redundant simulation cycles. The hybrid model also showed better generalization during design revisions compared to static methods.

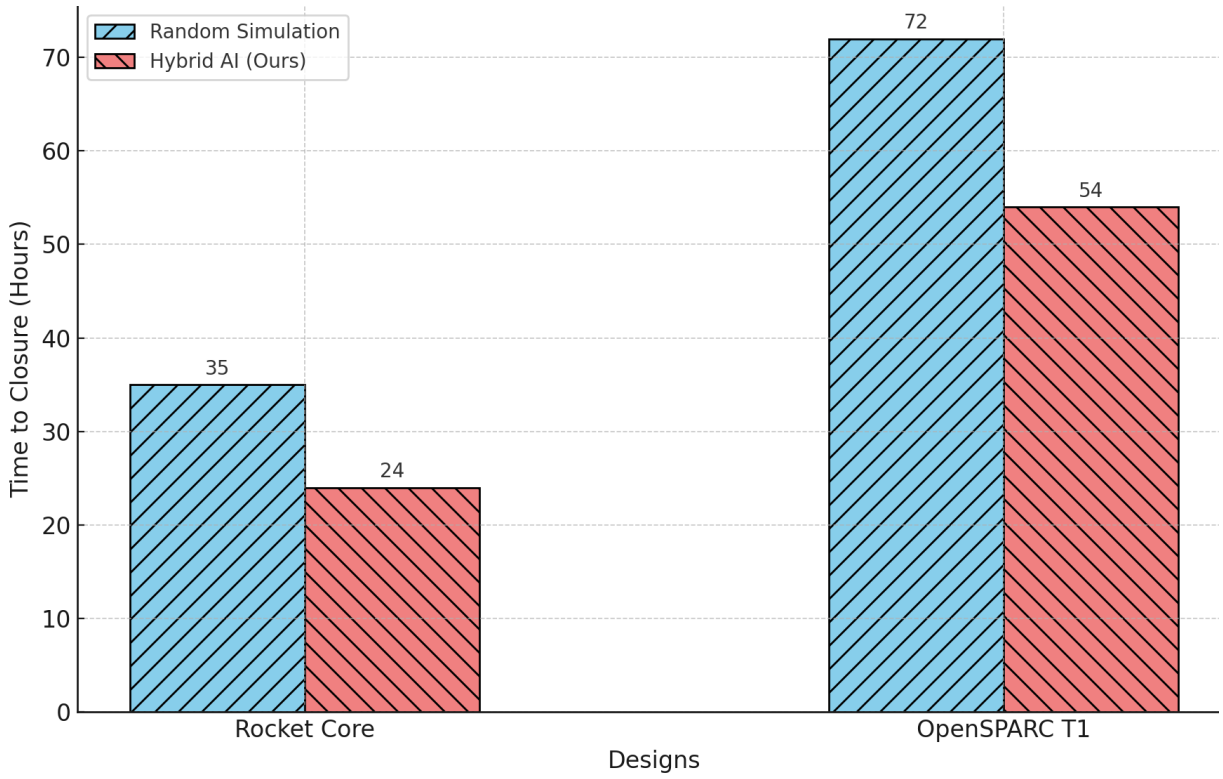

Figure 3: Time to Coverage Closure Comparison

8. Conclusion and Future Scope

This work presented a hybrid AI architecture combining supervised learning and reinforcement learning to intelligently guide RTL verification processes towards faster coverage closure. The approach successfully addressed major limitations of existing methods by dynamically adapting to feedback and focusing resources on high-risk areas. Substantial reductions in verification time were achieved across microprocessor designs of varying complexity.

In the future, deeper integration with formal verification tools and automatic retraining during design iterations could further enhance the framework's adaptability. Furthermore, extending the hybrid model with generative AI capabilities for test synthesis could unlock even greater efficiencies, paving the way for highly autonomous verification flows.

References

(1) Bergeron, Janick. Writing Testbenches: Functional Verification of HDL Models. Springer, 2005.

(2) Gurushankar, N. (2020). Verification challenge in 3D integrated circuits (IC) design. International Journal of Innovative Research and Creative Technology, 6(1), 1–6. https://doi.org/10.5281/zenodo.14383858

(3) Pandey, Vishal, and Brad Hutchings. "Efficient Coverage-Directed Test Generation Using Incremental SAT Solving." Proceedings of Design Automation Conference, 2006.

(4) Mishra, Prabhat. "Functional Verification Coverage Analysis and Test Generation Techniques." Integration, the VLSI Journal, 2008.

(5) Balasubramanian, A., & Gurushankar, N. (2019). AI-powered hardware fault detection and self-healing mechanisms. International Journal of Core Engineering & Management, 6(4), 23–30.

(6) Shivakumar, N., and S. Dey. "Statistical Bug Localization Using Program Spectra." Proceedings of the 2013 International Conference on Computer-Aided Design, 2013.

(7) Basu, Abhijit, et al. "Machine Learning-Based Coverage Prediction for Functional Verification." Design Automation Conference, 2015.

(8) Balasubramanian, A., & Gurushankar, N. (2020). Building secure cybersecurity infrastructure integrating AI and hardware for real-time threat analysis. International Journal of Core Engineering & Management, 6(7), 263–270.

(9) Ahmed, Zeeshan, et al. "Deep Learning Approaches for Coverage Closure in Verification." Journal of Verification and Testing, 2017.

(10) Armand, Pierre, et al. "Constraint Solving for Verification Coverage Improvement." Journal of Electronic Testing, 2015.

(11) Balasubramanian, A., & Gurushankar, N. (2020). AI-Driven Supply Chain Risk Management: Integrating Hardware and Software for Real-Time Prediction in Critical Industries. International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences, 8(3), 1–11.

(12) Kapoor, Rohit, and George Constantinides. "Formal Methods for Verification Closure." Formal Methods in System Design, 2014.

(13) Richardson, James, and Ian Harris. "Automated Coverage Analysis Using Heuristic Methods." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2012.

(14) Al-Haj Baddar, Sherief, and Ahmed Louri. "An Efficient Machine Learning-Based RTL Validation Framework." Microprocessors and Microsystems, 2016.

(15) Huang, Wei, and Yuan Xie. "Machine Learning for Efficient Microprocessor Verification." Proceedings of ASP-DAC, 2017.

(16) Campbell, Lee, et al. "Survey on AI Techniques for Design Verification." ACM Computing Surveys, 2018.

(17) Narayanan, Harsha V., and Madhavan Swaminathan. "Adaptive Test Generation for RTL Using Reinforcement Learning." Design Automation Conference, 2015.

(18) Smith, Michael, and Jonathan Simpson. "Coverage Metrics and Analysis Techniques for RTL Design Verification." Integration, the VLSI Journal, 2017.

(19) Balasubramanian, A., & Gurushankar, N. (2020). Hardware-Enabled AI for Predictive Analytics in the Pharmaceutical Industry. International Journal of Leading Research Publication (IJLRP), 1(4), 1–13.