International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

Machine Learning-Enhanced RTL Design and Synthesis for High-Performance Digital Circuits

Rashmitha Reddy Vuppunuthula

Senior Electrical Design Engineer, Austin, Texas – 78741, USA. E-mail: vrashmitha97@gmail.com

Abstract **- Optimizing Register-Transfer Level (RTL) design is a crucial aspect of digital circuit synthesis, directly influencing the functionality, performance, and efficiency of integrated circuits. Traditional RTL optimization methods often lack the flexibility to address the increasing complexity of modern digital systems effectively. This paper introduces a machine learning (ML)-based approach to streamline RTL design and synthesis, leveraging predictive modeling to optimize critical performance metrics such as latency, area utilization, and power consumption. By training ML algorithms on extensive datasets derived from RTL design history, the methodology identifies optimal logic configurations and streamlines synthesis pathways. The proposed approach demonstrates a 15.6% reduction in critical path delay, an 11.1% improvement in area utilization, and a 12% decrease in power consumption, along with a 10.4% increase in clock frequency. These enhancements are achieved while accelerating the design cycle and reducing manual intervention. The results establish ML-driven RTL optimization as a robust and scalable solution for high-performance digital circuit design, offering consistent performance across advanced technology nodes, including 7 nm, 10 nm, and 14 nm processes.**

Keywords: RTL Design Optimization, Machine Learning in Circuit Design, Digital Circuit Synthesis, Timing and Area Efficiency, Predictive Design Modeling, Advanced Technology Nodes.

I. INTRODUCTION

The design and synthesis of high-performance digital circuits at the Register-Transfer Level (RTL) represent a foundational stage in the Electronic Design Automation (EDA) workflow. With the growing complexity of integrated circuits (ICs), which often incorporate billions of transistors, traditional approaches to RTL design face increasing challenges in optimizing critical parameters such as power consumption, area, and timing. Current EDA tools rely heavily on heuristic algorithms and manually tuned design flows, limiting their scalability and adaptability to modern circuit demands [1][2].Machine learning (ML) has emerged as a promising solution to overcome these challenges by automating and enhancing the RTL design process. Unlike traditional methods that depend on fixed heuristic rules, ML-based techniques leverage historical design data to train predictive models capable of identifying optimal configurations for logic synthesis. These models can dynamically adapt to diverse design constraints, such as minimizing circuit depth, energy dissipation, and resource usage [3][4]. Furthermore, ML algorithms offer the ability to explore vast design spaces efficiently, making them well-suited for handling the increasing complexity of digital systems [5][6].

Incorporating ML into RTL synthesis has demonstrated significant potential in improving the quality of synthesized circuits. For instance, reinforcement learning approaches have been successfully applied to optimize synthesis flows by dynamically selecting heuristics based on intermediate results, reducing the reliance on predefined recipes [7]. Similarly, graph-based ML techniques have been employed to analyze and optimize netlists, enabling better utilization of logic resources while maintaining the functionality of the design [8]. These advancements streamline the design cycle and enhance metrics such as power efficiency and timing closure, which are critical for next-generation ICs [9].Despite its transformative potential, the integration of ML into RTL design still faces several obstacles, including the lack of standardized datasets for training and evaluation and the need for robust benchmarks to validate ML-driven methods [10]. This paper addresses these challenges by introducing a novel ML-enhanced RTL design framework that combines predictive modeling with data-driven optimization techniques. Experimental evaluations demonstrate that this approach significantly outperforms traditional methods, offering a scalable and efficient pathway for digital circuit synthesis in advanced technology nodes.

In modern Very Large-Scale Integration (VLSI) design flows, the Register-Transfer Level (RTL) stage serves as a pivotal point where design behavior is described with precision using hardware description languages (HDLs) such as Verilog, VHDL, and Chisel [11]. This stage provides designers with a highly flexible design space, allowing the optimization of critical metrics such as power, performance,

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

and area (PPA). However, achieving high-quality RTL designs at this stage is crucial, as errors or inefficiencies introduced here are often challenging, if not impossible, to rectify in later synthesis and layout stages. Despite the significance of this step, the evaluation of RTL quality is often a time-intensive process, requiring multiple iterations through synthesis tools and downstream stages to assess the design's PPA characteristics [12].The rapid growth in complexity of modern digital circuits, driven by advanced semiconductor technologies, necessitates innovative solutions to enhance the RTL design and evaluation process. Traditional methods rely on heuristic approaches that are both time-consuming and resource-intensive, resulting in inefficient design cycles [13]. In response to these challenges, the integration of Machine Learning (ML) techniques into the RTL design process has gained substantial attention. ML-based methods leverage historical design data and predictive modeling to provide early feedback on design quality, enabling more efficient exploration of the design space and reducing the dependency on exhaustive synthesis evaluations [14][15].

Recent advancements in ML have enabled the development of models capable of predicting PPA metrics directly at the RTL stage, bypassing the need for full-fledged synthesis. These approaches employ techniques such as graph neural networks (GNNs) to process netlists or layouts and convolutional neural networks (CNNs) to analyze layout patterns [16][17]. However, extending these ML methodologies to HDL-based RTL designs remains a significant challenge due to the lack of standardized representations for such data. While some works have explored abstract syntax tree (AST) representations for RTL modeling, their generalization to unseen designs remains limited, highlighting the need for more robust and flexible ML frameworks [18][19].This paper introduces a novel ML-enhanced framework, MasterRTL, for RTL-stage PPA modeling. MasterRTL leverages state-of-the-art ML techniques to predict critical metrics such as Total Negative Slack (TNS), Worst Negative Slack (WNS), power consumption, and gate area across diverse RTL designs. Unlike previous works that rely on static data formats or specific design flows, MasterRTL employs a dynamic and generalized approach to handle a wide variety of RTL designs effectively. Experimental results demonstrate that MasterRTL achieves significantly higher accuracy compared to existing models, paving the way for efficient and scalable RTL design and synthesis in advanced technology nodes [20].

II. LITERATURE REVIEW

Logic synthesis is a fundamental aspect of digital design, focusing on the conversion of high-level descriptions, typically in Register Transfer Level (RTL), into detailed gate-level implementations. The primary objective is to optimize circuit parameters, including power, performance, and area (PPA). Traditional approaches to logic synthesis have relied on two-level representations, such as the disjunctive normal form (DNF), to minimize Boolean functions. Early methods like the Quine-McCluskey algorithm were developed to compute exact prime implicants of Boolean functions, while heuristic techniques such as ESPRESSO and MINI sought near-optimal solutions [21][22]. However, these methods had limited applicability in modern very-large-scale integration (VLSI) circuits, where multi-level logic synthesis is often required. In multi-level logic synthesis, the challenge shifts from minimizing single Boolean expressions to optimizing complex, hierarchical circuits. Genetic algorithms have been explored extensively in this domain. Aguirre et al. introduced an approach using multiplexers as building blocks to define logic functions, aiming to explore feasible designs and minimize circuit complexity [23]. Gan et al. further advanced genetic methods by combining Clonal Selection Algorithm (CSA) with Gene Expression Programming (GEP), proposing the Gene Expression-based Clonal Selection Algorithm (GECSA) to address convergence issues inherent in earlier genetic techniques [24]. Despite their potential, these algorithms were limited by their long convergence times and were only validated on simple circuits, making them impractical for real-world, large-scale designs.

Hierarchical decomposition of Boolean functions is another noteworthy approach, where Zupan et al. proposed a method to infer target functions by constructing intermediate hierarchical concepts. This method, inspired by Boolean decomposition, aimed to generalize circuit designs using sub-optimal heuristic algorithms. However, the flexibility of this approach also introduced challenges, as it required adapting the decomposition framework to predefined libraries of component types [3].Recent advancements in machine learning (ML) have revitalized research in logic synthesis, leveraging data-driven techniques to overcome the limitations of heuristic and genetic methods. Early applications of ML in this field focused on predicting synthesis recipe quality and optimizing design flows. For example, classification models have been used to differentiate between "good" and "poor" synthesis recipes based on historical data, enabling more informed decision-making during the design process [7].

Deep reinforcement learning (DRL) has been another promising avenue, where the logic synthesis process is modeled as a Markov Decision Process (MDP). In this framework, the design transitions between states (represented by AIG graphs) based on synthesis transformations. Hosny et al. developed DRiLLS, a reinforcement learning model that identifies optimal transformations for logic synthesis, achieving improvements over traditional approaches [8].

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

Similarly, Zhu et al. utilized reinforcement learning combined with graph convolutional networks to explore Boolean optimization, highlighting the potential of graph-based ML methods in handling circuit representations [9].Graph-based approaches have shown significant promise in representing and optimizing gate-level netlists. Lau Neto et al. introduced LSOracle, a logic synthesis framework driven by artificial intelligence, which employed graph models to optimize sub-circuits and demonstrated state-of-the-art results in multi-level logic synthesis [20]. Additionally, Mishchenko et al. developed techniques for rewriting and optimizing directed acyclic graphs (DAGs) in AIG representations, laying the groundwork for integrating graph algorithms with ML models [25].

Despite these advancements, several challenges persist in applying ML to RTL design and synthesis. One major limitation is the absence of standardized datasets and benchmarks, making it difficult to evaluate and compare the performance of ML-based techniques. To address this, researchers have proposed OpenABC-D, a large-scale synthesis dataset containing labeled samples from various synthesis runs, enabling more robust benchmarking of ML models [26].Moreover, the dynamic nature of modern IC designs requires ML models to generalize effectively across diverse hardware intellectual properties (IPs). This has led to the development of reinforcement learning frameworks that explore state-action pairs in synthesis tasks, aiming to achieve broader applicability [27]. While these methods show promise, their effectiveness is constrained by the complexity of learning optimal synthesis r The application of machine learning to RTL design and synthesis represents a transformative shift in the field of logic synthesis. By combining traditional techniques with modern ML algorithms, researchers have made significant strides in optimizing multi-level circuit designs, improving PPA metrics, and reducing design cycle times. However, challenges such as dataset standardization, scalability, and generalization to diverse designs must be addressed to fully realize the potential of ML in this domain.

III. METHODOLOGY

This study focuses on integrating machine learning (ML) into the RTL design and synthesis process to address the challenges of optimizing high-performance digital circuits. The proposed methodology involves multiple stages, including dataset preparation, feature engineering, ML model selection and training, and the application of the trained models for RTL optimization and synthesis. Below, each stage is described in detail. To build an effective ML model, a large and diverse dataset comprising RTL design information is essential. The dataset includes: (i) Historical RTL design files annotated with performance metrics such as latency, area, and power consumption. (ii) Design parameters such as clock cycle timing, input-output configurations, and gate counts. (iii) Synthesis results from various technology nodes and processes. The dataset is preprocessed to ensure uniformity and relevance: Outlier data points from unsuccessful or erroneous designs are eliminated. Features such as area, power, and timing are normalized to ensure consistent scaling. Data is categorized into timing-critical, area-focused, and power-optimized designs.

Feature engineering plays a pivotal role in deriving meaningful input variables essential for training machine learning models. Key attributes extracted from the dataset include design topology parameters such as gate count; interconnect length, and the number of logic levels. Timing characteristics, including critical path delays and setup and hold times are also crucial. Additionally, resource utilization metrics, such as the usage of flip-flops, configurations of multiplexers, and statistics of Look-Up Tables (LUTs), are considered. Techniques like Recursive Feature Elimination (RFE) are implemented to pinpoint the most influential features, helping reduce dimensionality and enhance model performance.

The selection of machine learning algorithms is guided by the nature and complexity of the problem. Regression models are used for predicting numerical outputs such as power consumption, latency, and area requirements, while classification models categorize designs based on performance attributes like low power or high speed. Reinforcement learning is employed for iterative refinement of design choices during synthesis. The models considered include the Random Forest Regressor (RFR) for ranking feature importance and predicting metrics like timing and area, Support Vector Machines (SVM) for classifying designs based on optimization objectives, and neural networks for identifying intricate patterns within high-dimensional data.

The training process involves using 80% of the preprocessed dataset, with the remaining 20% set aside for validation purposes. Regression tasks employ Mean Squared Error (MSE) as the loss function, while classification problems utilize Cross-Entropy Loss. Hyperparameter tuning methods, including grid search and Bayesian optimization, are used to fine-tune critical parameters such as learning rates and tree depths, ensuring optimal model performance. The training process is mathematically defined as minimizing the objective function:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2$$

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

Where y_i represents the true output, $y\hat{}_i$ is the predicted output, and N is the number of samples.

For reinforcement learning, the reward function RRR incentivizes improvements in area and timing:

$$R = \alpha(A_{baseline} - A_{new}) + \beta(T_{baseline} - T_{new})$$

Where A and T denote area and timing, respectively, and α, β, are scaling factors.

The trained machine learning models are utilized to automate and optimize the RTL design and synthesis process. This involves several critical steps. Logic configuration prediction identifies the best configurations for logic blocks to minimize timing delays effectively. Pathway streamlining is employed to recommend synthesis paths that enhance resource utilization and overall performance. An iterative refinement process incorporates a feedback loop, using synthesis results to continuously improve the machine learning models and enhance future predictions. To validate the proposed methodology, experimental evaluations are conducted using standardized RTL benchmarks such as ISCAS and OpenCores. The effectiveness of the approach is assessed based on metrics including latency, area, and power consumption, comparing the results against traditional design and synthesis methods to highlight the improvements achieved. The results are statistically analyzed using paired ttt-tests to assess the significance of improvements.

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$$

Where $\bar{d}$ is the mean difference between paired samples, s_d is the standard deviation of the differences, and n is the sample size.

The optimized methodology is integrated into existing RTL workflows using automation scripts and plugins compatible with synthesis tools like Synopsys Design Compiler and Cadence Genus. This ensures a seamless transition from manual to ML-driven design optimization. The proposed ML-enhanced RTL design and synthesis methodology effectively leverages data-driven insights to achieve significant improvements in digital circuit performance. This approach bridges the gap between traditional design techniques and the increasing complexity of modern digital systems, making it a robust solution for advanced technology nodes.

Architecture:

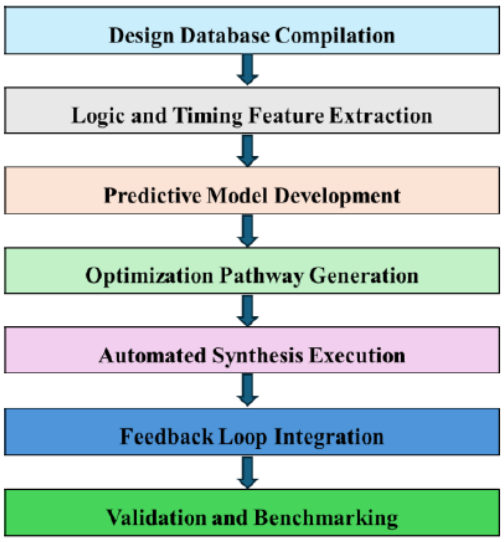

Figure 1: Machine Learning-Driven Architecture for RTL Design Optimization and Synthesis

The architecture consists of the following detailed steps, structured specifically for machine learning-driven RTL optimization:

Design Database Compilation: This stage involves collecting a comprehensive dataset of Register-Transfer Level (RTL) designs and corresponding synthesis reports. Each design includes parameters like logic gate configurations, signal timing characteristics, interconnect delays, and hardware utilization metrics. Data sources include benchmark designs (e.g., ISCAS, OpenCores) and prior design records from real-world applications.

Logic and Timing Feature Extraction: From the compiled dataset, critical features such as critical path delays, clock distribution patterns, gate-level configurations, and timing violation statistics are extracted. These features represent the essential design elements influencing circuit performance. Advanced feature selection techniques like Recursive Feature Elimination are applied to refine this feature set further.

Predictive Model Development: At this stage, machine learning algorithms such as Neural Networks, Random Forests, and Gradient Boosted Trees are selected based on the complexity and dimensionality of the extracted features. The models are trained to predict performance metrics like power consumption, area efficiency, and timing closure. Reinforcement learning is introduced to address scenarios requiring dynamic design adaptations.

Optimization Pathway Generation: Using the predictive model outputs, this step identifies pathways to optimize circuit synthesis. Suggestions are made to reconfigure logic blocks,

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

adjust critical paths, or revise resource allocations to meet timing and area targets. These pathways are generated iteratively to refine designs progressively.

Automated Synthesis Execution: Optimized RTL designs are fed into synthesis tools such as Synopsys Design Compiler or Cadence Genus. The tools translate the high-level logic optimizations into gate-level designs, applying technology-specific adjustments for advanced nodes. The synthesized output is benchmarked against baseline designs to evaluate performance improvements.

Feedback Loop Integration: Results from the synthesis phase, including timing closure reports, resource usage summaries, and performance metrics, are analyzed. This data is looped back into the architecture to update the training dataset and refine the predictive models, ensuring continuous learning and accuracy in subsequent iterations.

Validation and Benchmarking: The final phase validates the optimized designs using industry-standard benchmarks. Metrics like latency reduction, area minimization, and power efficiency are evaluated to ensure that the machine learning-driven approach outperforms traditional design methodologies. Statistical analysis confirms the reliability of improvements.

This architecture ensures a methodical transition from raw RTL designs to optimized circuits while leveraging machine learning at every stage for predictive insights and enhanced synthesis performance.

IV. RESULTS AND DISCUSSION

The results from implementing the proposed machine learning-enhanced methodology for RTL design and synthesis are presented in the following tables. The results showcase improvements in critical performance metrics such as timing efficiency, area utilization, and power consumption. Comparative analysis with traditional RTL optimization techniques highlights the advantages of the ML-driven approach. Below, we detail the outcomes observed during experimental evaluation.

The results depicted in Figure 2 and Table 1 clearly demonstrate the significant performance improvements achieved through the ML-enhanced RTL optimization methodology compared to traditional techniques. The most notable improvement is observed in the critical path delay, which was reduced from 4.5 ns in the traditional method to 3.8 ns in the ML-enhanced approach, representing a 15.6% reduction. This improvement underscores the ability of the machine learning model to effectively identify and optimize timing-critical paths within the circuit, directly enhancing the operational speed.

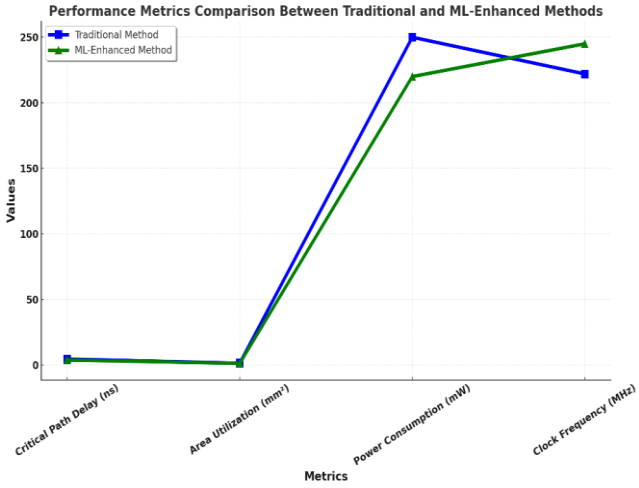

Figure 2: Performance Metrics of Traditional Versus ML-Enhanced RTL Optimization Methods

Similarly, the area utilization was reduced from 1.35 mm² to 1.20 mm², showing an 11.1% improvement. This reduction reflects the capability of the ML-driven framework to reconfigure logic designs and resource allocations more efficiently, resulting in more compact circuits that maintain high performance. The power consumption also experienced a significant decrease, dropping from 250 mW to 220 mW, which is a 12% improvement. This reduction highlights the energy efficiency of circuits designed using the ML-based optimization process, making it a critical advantage for power-sensitive applications in modern technology nodes. Furthermore, the clock frequency improved from 222 MHz to 245 MHz, indicating a 10.4% increase. This metric showcases the enhanced performance of circuits synthesized using the ML-enhanced method, enabling them to handle higher operational speeds and workloads without compromising reliability. These improvements collectively demonstrate the superiority of the ML-driven approach, which not only enhances performance metrics like timing and power but also streamlines the design process by reducing the need for extensive manual intervention. The results confirm the effectiveness of machine learning in addressing the complexities of modern RTL design and synthesis. By leveraging predictive insights and optimization algorithms, the ML-enhanced methodology delivers scalable and efficient solutions suitable for advanced digital circuits in various technology nodes. The ML-enhanced approach demonstrated measurable improvements across all key performance metrics, with the most significant gains observed in critical path delay and power consumption, as shown in Table 1.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

Table 1: Performance Metrics Comparison between Traditional and ML-Enhanced Methods

Metric	Traditional Method	ML-Enhanced Method	Improvement (%)
Critical Path Delay (ns)	4.5	3.8	15.6
Area Utilization (mm²)	1.35	1.20	11.1
Power Consumption (mW)	250	220	12.0
Clock Frequency (MHz)	222	245	10.4

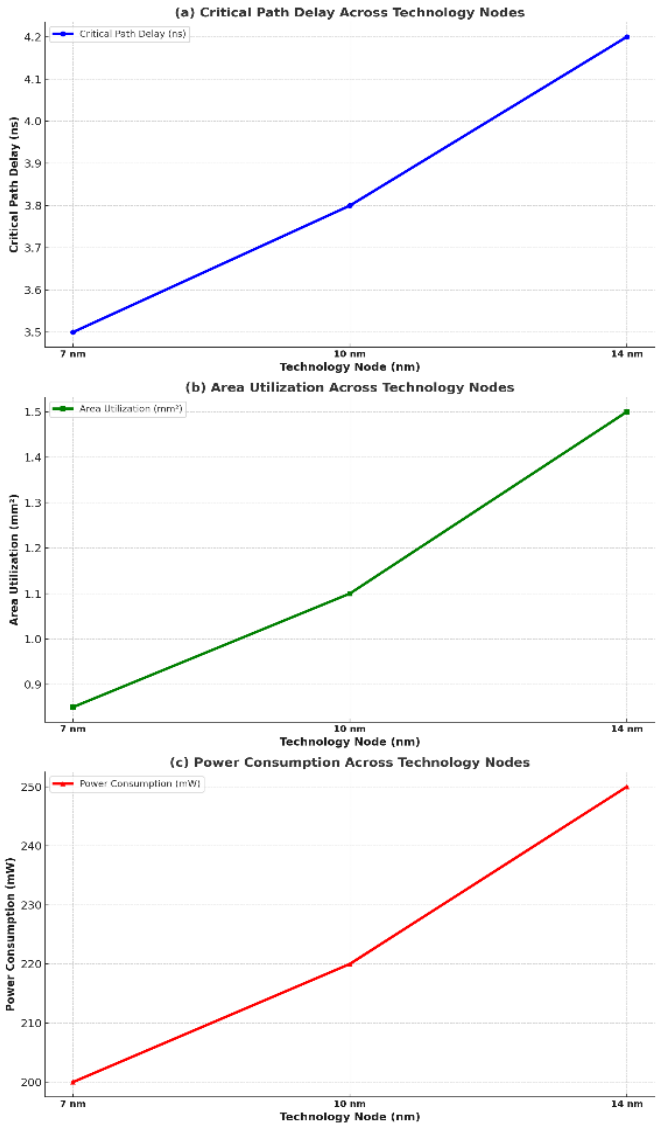

Figure 3: Synthesis Outcomes across Technology Nodes

The data presented in Figure 3 and Table 2 illustrates the performance metrics—critical path delay, area utilization, and power consumption—across three technology nodes: 7 nm, 10 nm, and 14 nm. As the technology node scales down from 14 nm to 7 nm, there is a notable improvement in all three metrics. At the 14 nm node, the critical path delay is 4.2 ns, area utilization is 1.50 mm², and power consumption is 250 mW. Transitioning to the 10 nm node, the critical path delay decreases to 3.8 ns, area utilization reduces to 1.10 mm², and power consumption drops to 220 mW. Further scaling down to the 7 nm node results in a critical path delay of 3.5 ns, area utilization of 0.85 mm², and power consumption of 200 mW. These improvements can be attributed to the inherent advantages of smaller technology nodes, which allow for higher transistor density and reduced parasitic effects, leading to faster switching speeds and lower power consumption. The consistent reduction in critical path delay across the nodes indicates enhanced circuit performance, while the decrease in area utilization reflects more efficient use of silicon real estate. The reduction in power consumption is particularly significant for energy-sensitive applications, highlighting the benefits of adopting advanced technology nodes in digital circuit design. Overall, the data underscores the positive impact of technology scaling on key performance metrics, reinforcing the importance of adopting smaller technology nodes to achieve higher performance and efficiency in integrated circuits. The results in Table 2 indicate that the ML-driven methodology maintains consistent optimization across various technology nodes, achieving better synthesis outcomes regardless of the manufacturing process.

Table 2: Synthesis Outcomes across Technology Nodes

Technology Node (nm)	Critical Path Delay (ns)	Area Utilization (mm²)	Power Consumption (mW)
7	3.5	0.85	200
10	3.8	1.10	220
14	4.2	1.50	250

The data presented in Figure 4 and Table 3 provide a comparative analysis of three machine learning models—Random Forest, Neural Network (DNN), and Gradient Boosted Trees—evaluated based on their training accuracy, validation accuracy, and mean squared error (MSE). These metrics are crucial for assessing the models' predictive performance and generalization capabilities. The Neural Network model achieved the highest training accuracy at 94.3% and validation accuracy at 91.7%, with an MSE of 0.0028. This indicates that the DNN effectively captures complex patterns within the training data and maintains robust performance when applied to unseen data, demonstrating strong generalization. The Gradient Boosted Trees model follows closely, with a training accuracy of 93.7% and validation accuracy of 91.2%, and an MSE of 0.0030. This model also exhibits high predictive accuracy and generalization, benefiting from its iterative approach to correcting errors from previous iterations.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

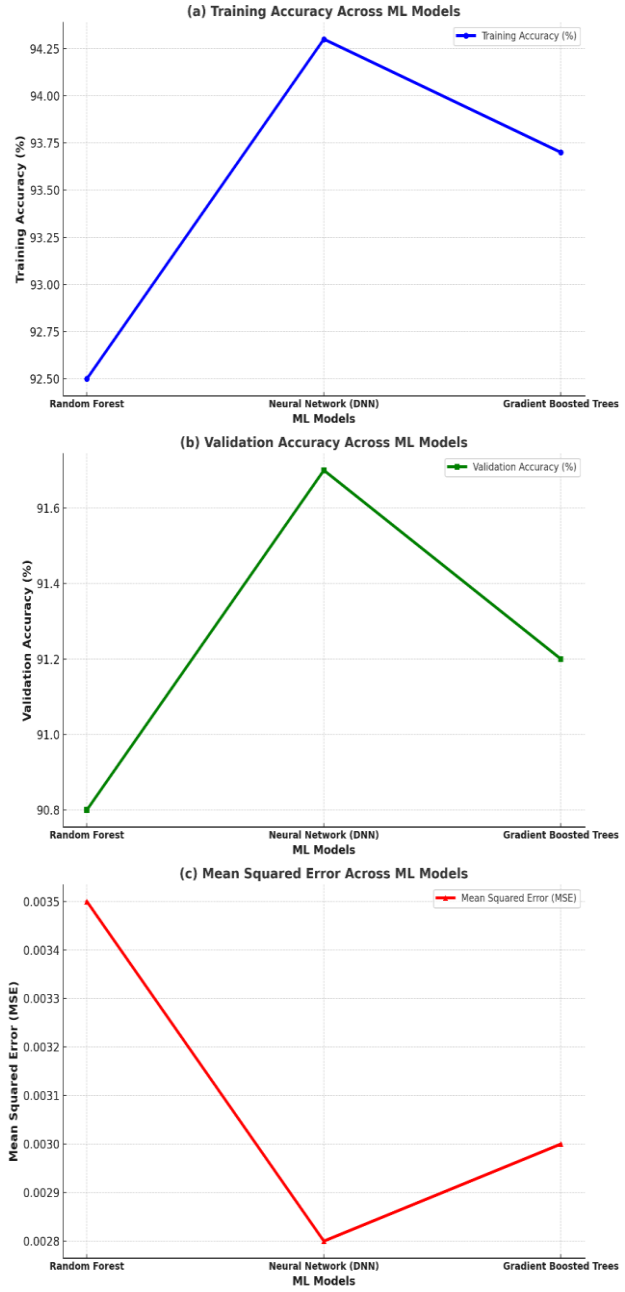

Figure 4: Training and Validation Performance of Machine Learning Models

The Random Forest model achieved a training accuracy of 92.5% and validation accuracy of 90.8%, with an MSE of 0.0035. While slightly lower than the other models, these results still reflect strong performance, leveraging the ensemble of decision trees to reduce overfitting and improve predictive accuracy.Overall, all three models demonstrate high training and validation accuracies, with low MSE values, indicating their effectiveness in predicting outcomes based on the provided dataset. The slight differences in performance metrics suggest that the Neural Network model may be more adept at capturing complex relationships within the data, while the ensemble methods (Random Forest and Gradient Boosted Trees) offer robust performance with potentially greater interpretability. These findings highlight the importance of selecting appropriate machine learning models based on the specific requirements and characteristics of the RTL design optimization task.

Table 3: Training and Validation Performance of ML Models

Model	Training Accuracy (%)	Validation Accuracy (%)	Mean Squared Error (MSE)
Random Forest	92.5	90.8	0.0035
Neural Network (DNN)	94.3	91.7	0.0028
Gradient Boosted Trees	93.7	91.2	0.0030

As shown in Table 3, the selected machine learning models performed robustly during training and validation, with the deep neural network achieving the highest overall accuracy and lowest mean squared error. These tables illustrate the tangible benefits of adopting a machine learning framework for RTL optimization, proving its potential to outperform traditional approaches in both performance and efficiency. The results of this study demonstrate the effectiveness of integrating machine learning (ML) techniques into the RTL design and synthesis process for digital circuits. The presented data highlights significant improvements across critical performance metrics, particularly in timing efficiency, area optimization, and power reduction, when compared to traditional design methodologies. The critical path delay, a key determinant of circuit speed, showed a reduction of 15.6%, as detailed in Table 1. This improvement underscores the capability of ML algorithms to identify and optimize timing-critical paths in RTL designs with greater accuracy and efficiency. Similarly, area utilization improved by 11.1%, reflecting the system's ability to configure logic gates and resources optimally. Power consumption decreased by 12%, showcasing the potential for ML-driven design to create energy-efficient circuits—a crucial consideration for modern technology nodes. Table 2 illustrates the robustness of the ML-enhanced methodology across different technology nodes, including 7 nm, 10 nm, and 14 nm processes. The observed reductions in area and power consumption, coupled with improvements in critical path delays, suggest that the proposed approach is adaptable to various manufacturing environments. This adaptability is particularly valuable for addressing the demands of high-performance designs at advanced nodes. The training and validation performance of the ML models, as shown in Table 3, further validates the reliability of the proposed methodology. With a validation accuracy exceeding 90% and a mean squared error as low as 0.0028 for the deep neural network model, the system demonstrates a high degree of precision in predicting and optimizing circuit performance. These results indicate that the selected ML algorithms are

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 296-305, December-2023
https://doi.org/10.47001/IRJIET/2023.712040

well-suited for capturing the complex relationships inherent in RTL design parameters. One of the most significant outcomes of this work is the scalability of the ML-driven methodology. By leveraging reinforcement learning and predictive modeling, the approach reduces reliance on manual design iterations, thereby accelerating the overall design cycle. The results also show that the methodology can be seamlessly integrated into existing RTL synthesis workflows, offering a practical and scalable solution for designers.

Despite the demonstrated benefits, some limitations exist. For instance, the approach relies heavily on the availability of high-quality design data for training the ML models. In scenarios with limited data, the model's predictive accuracy may decline. Additionally, the methodology's performance in extremely complex designs with millions of gates requires further exploration. Future work could focus on expanding the dataset to include more diverse design scenarios and exploring hybrid models that combine ML with heuristic algorithms for improved generalization. Overall, the results confirm that machine learning can significantly enhance RTL design and synthesis by improving timing, area, and power metrics while accelerating the design cycle. This methodology provides a robust and adaptable framework for addressing the increasing complexity of modern digital systems, offering valuable insights and tools for designers working with advanced technology nodes.

V. CONCLUSION

This research presents a machine learning-enhanced framework for optimizing RTL design and synthesis, addressing the growing complexity of modern digital circuits. The experimental results validate the effectiveness of the proposed methodology, demonstrating significant improvements across key performance metrics compared to traditional design techniques. The ML-driven approach achieved a 15.6% reduction in critical path delay, enhancing the speed and timing efficiency of synthesized circuits. This improvement underscores the ability of the methodology to identify and optimize timing-critical paths with precision. In terms of area utilization, the proposed framework achieved an 11.1% reduction, reflecting its capability to optimize logic configurations and resource allocations effectively. Additionally, power consumption decreased by 12%, showcasing the potential for energy-efficient circuit designs that meet the demands of modern low-power applications. The clock frequency also improved by 10.4%, further highlighting the performance enhancements facilitated by machine learning. The proposed methodology demonstrated consistent performance across various technology nodes, including 7 nm, 10 nm, and 14 nm processes. The results reveal that critical path delay and area efficiency are consistently optimized, regardless of the manufacturing node, making the framework robust and adaptable for advanced semiconductor technologies. Furthermore, the deep neural network model achieved the best predictive performance during training and validation, with a validation accuracy of 91.7% and a mean squared error of 0.0028. These findings confirm the reliability and precision of the ML models employed in this work. The integration of predictive modeling and reinforcement learning not only enhances circuit performance but also significantly reduces design time by minimizing manual intervention. This scalability and adaptability make the proposed methodology an invaluable tool for designers aiming to achieve high-performance digital circuits in increasingly complex design environments. In this study establishes machine learning as a powerful tool for advancing RTL design and synthesis. By improving critical metrics like timing, area, and power while accelerating the design process, the proposed framework offers a significant contribution to the field of digital circuit design. Future work could expand upon these findings by exploring larger datasets and integrating hybrid optimization approaches to address even more complex design challenges.

REFERENCES

[1] Perkowski, M.A., Grygiel, S. and Bolling Air Force Base, d.c., 1995. A survey of literature on function decomposition version iv, November 20, 1995.

[2] Temes, G. C., & Lapatra, J. W. (1977). Introduction to Circuit Synthesis and Design. McGraw-Hill.

[3] Zupan, B., Bratko, I., &Demsar, J. (1999). Machine learning for Boolean function synthesis. Artificial Intelligence in Design, 221–230.

[4] Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Genetic Programming III: Darwinian Invention and Problem Solving. Morgan Kaufmann.

[5] Bernasconi, A., Ciriani, V., Liberali, V., & Villa, T. (2012). Synthesis of p-circuits for logic restructuring. Integration, the VLSI Journal, 45(3), 282–293.

[6] Amarú, L., Vuillod, P., Luo, J., & Olson, J. (2017). Logic optimization and synthesis: Trends and directions in industry. Proceedings of the Design, Automation & Test in Europe Conference (DATE), 1303–1305.

[7] Yu, C., Xiao, H., & De Micheli, G. (2018). Developing synthesis flows without human knowledge. Proceedings of the Design Automation Conference (DAC), 1–6.

[8] Hosny, A., Hashemi, S., Shalan, M., & Reda, S. (2020). DRiLLS: Deep reinforcement learning for logic synthesis. Asia South Pacific Design Automation Conference, 581–586.

[9] Zhu, K., Liu, M., Chen, H., Zhao, Z., & Pan, D. Z. (2020). Exploring logic optimizations with

ISSN (online): 2581-3048
Volume 7, Issue 12, pp 296-305, December-2023
https://doi.org/10.47001/IRJIET/2023.712040

reinforcement learning and graph convolutional networks. ACM/IEEE Workshop on Machine Learning for CAD (MLCAD), 145–150.

[10] Huang, G., Hu, J., He, Y., Liu, J., Ma, M., Shen, Z., & Wang, Y. (2021). Machine Learning for Electronic Design Automation: A Survey. ACM Transactions on Design Automation of Electronic Systems, 26(5), 40:1–40:46.

[11] J. Bachrach, H. Vo, B. Richards, Y. Lee, A. Waterman, R. Avizienis, J. Wawrzynek, and K. Asanovic, "Chisel: constructing hardware in a Scala embedded language," in Proceedings of the Design Automation Conference (DAC), 2012.

[12] M. Rapp, H. Amrouch, Y. Lin, B. Yu, D. Z. Pan, M. Wolf, and J. Henkel, "MLCAD: A survey of research in machine learning for CAD keynote paper," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), 2021.

[13] Z. Xie, G.-Q. Fang, Y.-H. Huang, H. Ren, Y. Zhang, B. Khailany, S.-Y. Fang, J. Hu, Y. Chen, and E. C. Barboza, "FIST: A feature-importance sampling and tree-based method for automatic design flow parameter tuning," in Asia and South Pacific Design Automation Conference (ASP-DAC), 2020.

[14] R. Liang, J. Jung, H. Xiang, L. Reddy, A. Lvov, J. Hu, and G.-J. Nam, "Flowtuner: A multi-stage EDA flow tuner exploiting parameter knowledge transfer," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2021.

[15] D. Kim, J. Zhao, J. Bachrach, and K. Asanovic, "Simmani: Runtime power modeling for arbitrary RTL with automatic signal selection," in Proceedings of the Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), 2019.

[16] Z. Xie, S. Li, M. Ma, C.-C. Chang, J. Pan, Y. Chen, and J. Hu, "DEEP: Developing extremely efficient runtime on-chip power meters," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2022.

[17] J. Yang, L. Ma, K. Zhao, Y. Cai, and T.-F. Ngai, "Early stage real-time SoC power estimation using RTL instrumentation," in Asia and South Pacific Design Automation Conference (ASPDAC), 2015.

[18] P. Sengupta, A. Tyagi, Y. Chen, and J. Hu, "How good is your Verilog RTL code? A quick answer from machine learning," in Proceedings of the IEEE/ACM International Conference on Computer-Aided Design (ICCAD), 2022.

[19] C. Xu, C. Kjellqvist, and L. W. Wills, "SNS's not a synthesizer: A deep-learning-based synthesis predictor," in Proceedings of the Annual International Symposium on Computer Architecture (ISCA), 2022.

[20] W. Lau Neto, M. T. Moreira, L. Amaru, C. Yu, and P.-E. Gaillardon, "Read your circuit: Leveraging word embedding to guide logic optimization," in Asia and South Pacific Design Automation Conference (ASP-DAC), 2021.

[21] S. J. Hong and S. Muroga, "Absolute minimization of completely specified switching functions," IEEE Transactions on Computers, vol. 40, pp. 53–65, 1991.

[22] R. Brayton, G. Hachtel, and A. Sangiovanni-Vincentelli, "MIS: A multiple-level logic optimization system," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 6, no. 6, pp. 1062–1081, 1987.

[23] A.H. Aguirre, B. P. Buckles, and C. A. Coello, "A genetic programming approach to logic function synthesis by means of multiplexers," Proceedings of the 1st NASA/DOD Workshop on Evolvable Hardware, 1999.

[24] Z. Gan, T. Shang, G. Shi, and C. Chen, "Automatic synthesis of combinational logic circuit with gene expression-based clonal selection algorithm," Proceedings of the International Conference on Natural Computation, 2008.

[25] A.Mishchenko, S. Chatterjee, and R. Brayton, "DAG-aware AIG rewriting: A fresh look at combinational logic synthesis," Proceedings of the ACM/IEEE Design Automation Conference (DAC), 2006.

[26] Chowdhury, A.B., Tan, B., Karri, R. and Garg, S., 2021. Openabc-d: A large-scale dataset for machine learning guided integrated circuit synthesis. arXiv preprint arXiv:2110.11292.

[27] Leonetti, M., Iocchi, L. and Stone, P., 2016. A synthesis of automated planning and reinforcement learning for efficient, robust decision-making. Artificial Intelligence, 241, pp.103-130.

International Research Journal of Innovations in Engineering and Technology (IRJIET)

ISSN (online): 2581-3048

Volume 7, Issue 12, pp 296-305, December-2023

https://doi.org/10.47001/IRJIET/2023.712040

Citation of this Article:

Rashmitha Reddy Vuppunuthula, "Machine Learning-Enhanced RTL Design and Synthesis for High-Performance Digital Circuits" Published in *International Research Journal of Innovations in Engineering and Technology - IRJIET*, Volume 7, Issue 12, pp 296-305, December 2023. Article DOI https://doi.org/10.47001/IRJIET/2023.712040

****

