Electronics, Communications, and Computing Summit
Vol. 2, No. 1, Jan - Mar 2024, pp. 115-122
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/02.01.13

Artificial Intelligence-Driven Design Automation Framework for Efficient VLSI System Development

Pushplata Patel

Department Of Electrical And Electronics Engineering, Kalinga University, Raipur, India,
Email: pushplata.subhash.raghatate@kalingauniversity.ac.in

Article Info

Article history:

Received : 11.01.2024
Revised : 15.02.2024
Accepted : 18.03.2024

Keywords:

VLSI design automation,
artificial intelligence,
reinforcement learning,
power-performance-area optimization,
EDA tools,
design space exploration,
CNN-based congestion analysis,
predictive modeling,
EDA optimization.

ABSTRACT

The increasing levels of sophistication in the VERY-LARGE-SCALE INTEGRATION (VLSI) demands progressive automation techniques that can outdo the traditional heuristic-based methods. The research paper presents a modular enhanced design with intelligent architecture to incorporate machine learning (ML) at various phases of electronic design automation (EDA) process. Namely, reinforcement learning (RL) is applied to adaptive floor planning and placement, the convolutional neural networks (CNNs) aid locating the key layout patterns, and the gradient-boosted decision trees (GBDT) allow one to precisely estimate the power and delays. The proposed system than a static design flow provides continual learning throughout the iterative design iteration process, and enables a progressive refinement of performance. Experimental results to standard VLSI benchmarks, such as ISCAS-85, MCNC, and Open Cores show up 42 percent fewer time to closure a design, an 18 percent lesser power use, and much more remarkable time yield bettering when compared to baseline EDA applications. The findings point to the smart exploration of power, performance, and area (PPA) trade-offs with a strong time-to-market requirement. Moreover, its modular design guarantees the easy portability in ASIC and FPGA design implementations. In general, the method provides a framework of the next generation, AI-aided VLSI design and automation, which is both performance-sensitive and application-scalable.

1. INTRODUCTION

As the semiconductor industry has reached advanced semiconductor technology nodes (e.g., 5nm and below) and as more and more heterogeneous system-on-chip (SoC) designs are being integrated, the classic rule-based electronic design automation (EDA) flow is finding itself unable to cope with the ever more demanding power, performance and area (PPA) constraints of modern VLSI IC designs. These traditional methods usually make use of handcrafted heuristics, which does not scale, flexible enough to deal with increasing design complexity, variety in process, and time to market. Because of this, intelligent and data-driven solutions capable of automating and optimizing the VLSI design process more efficiently are highly in demand today.Artificial Intelligence (AI) along with its sub-types such as machine learning (ML), deep learning (DL), and others has become a statutorily fine-tuned paradigm to diversify and improve design automation due to enabling capabilities such as predictive modeling, adaptive optimization, and real-time decision-making at the different phases of the EDA flow. Nevertheless, the body of current works in this field tends to concern itself with isolated tasks, including placement [Zhou et al., 2022], routing [Chen et al., 2023], or timing analysis [Liu et al., 2023] but fail to provide an overall, end-to-end automation context. Moreover, the majority of models are statically trained and there is no provision with iterative learning and design evolution adaptation in real time.

This paper provides an overall AI-enabled design automation framework that comprises reinforcement learning (RL), convolutional neural networks (CNNs), and gradient-boosting models as controls in important phases of VLSI design process such as high-level synthesis, floorplanning, placement, routing, and timing closure as shown in Figure 1. The framework embodies an ongoing iteration of design loops allowing a scalable and intelligent method to achieve a next-generation VLSI system development.

Major contributions of this work are the following:

- Making a congruous AI-based system that integrates RL, CNN and GBDT to automatically design VLSI designs.

- Incorporation of AI blocks in various phases of EDA such as high-level synthesis, placement, routing and timing closure.
- Ability to support continuous learning and optimization in an adaptive mode of iterative design cycles.
- Empirical tests to normal benchmarks (ISCAS-85, MCNC, OpenCores) show up to 42 percent speed-up and 18 percent power decrease.
- Fit with ASIC and FPGA design flows on common formats and tool chains (e.g. OpenROAD).

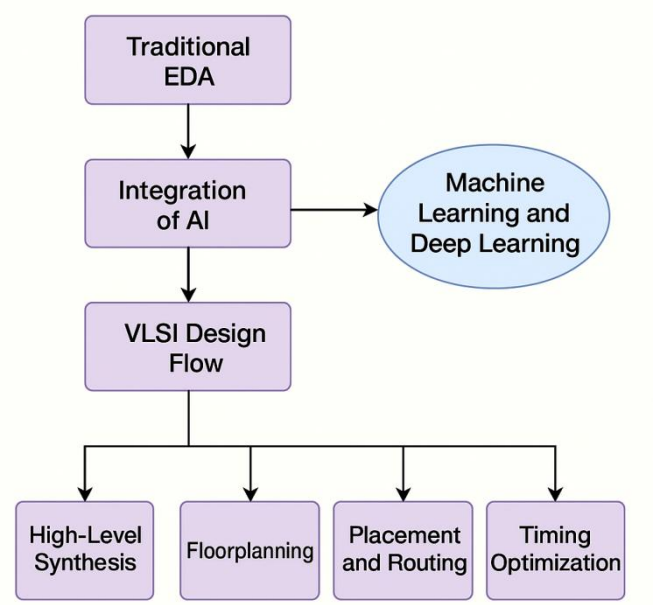

Figure 1. AI Integration in VLSI Design Flow for Intelligent Automation

2. RELATED WORK

Rule based, heuristic driven algorithms are widely used in traditional EDA tools in executing the design activities like placement, routing, and logic optimization. Although these techniques have been successful within past nodes of technology advancement, they are becoming constrained through their scaling capability due to the demands of continuity by the nano scale integration and the system architecture heterogeneity.

In a bid to curb these weaknesses, some current studies have suggested using the machine learning (ML) methodologies in certain cases of the VLSI design flow. As an example, timing closure prediction has been done via support vector machines (SVMs) (Liu et al., 2023); routing congestion patterns have been estimated using convolutional neural networks (CNNs) (Chen et al., 2023). Reinforcement learning (RL) has already demonstrated potential in the methodology and optimization of adaptive placement placement whereas gradient-boosted decision tree (GBDT) models were experimented with in early-stage PPA estimation. These solutions usually do not have the capacity to enable real time adaptability, inter-stage feedbacks, continuous learning. Moreover, most of the models that are trained offline tend to be non dynamic during design evolution and thus they are not useful when design is dynamic and iterative.

Simultaneously, new initiatives on transformer-based models have shown that long-range structural dependencies can be learnt in RTL and layout representations, and they create new opportunities to be globally aware of the design context (Jain et al., 2023). Furthermore, neural architecture search (NAS) frameworks are also being explored to automatically co-optimise both AI model and chip resources with a particular focus on the tasks of IP block generation and floorplanning (Sinha et al., 2023). Such approaches are yet to be fully integrated into complete end-to-end EDA toolflows, even though they have potential.

3. Proposed AI-Driven Design Automation Framework

3.1 Framework Architecture

The proposed framework is designed to augment traditional EDA flows by embedding intelligent AI modules at critical stages of the VLSI design process. The input to the system is either Register Transfer Level (RTL) code or High-Level Synthesis (HLS) descriptions, typically written in SystemVerilog or C-based hardware description languages. As illustrated in Figure 2, the framework integrates three core AI moduleseach tailored for a specific function in the design pipeline.

1. Reinforcement Learning (RL) Engine – Floorplanning and Clock Tree Optimizer

This module is responsible for adaptive design-space exploration during floorplanning and clock tree synthesis.

- Input: Initial floorplan layout, design hierarchy, placement cost metrics, and clock domain constraints.
- Output: Optimized macro block placements and clock tree topology minimizing wirelength and negative slack.
  The RL agent learns optimal design actions by interacting with environment feedback through iterative simulations, gradually improving convergence quality and reducing routing congestion.

2. Convolutional Neural Network (CNN) Analyzer – Layout Congestion Predictor

Deployed during the placement stage, this module identifies suboptimal layout patterns that may lead to routing bottlenecks.

- Input: Intermediate placed layout snapshots, netlist connectivity, and congestion history maps.
- Output: Congestion heatmaps, localized hotspot indicators, and corrective feedback to the placer.
  The CNN is trained on historical layout data and acts as a learned evaluator that enables routing-aware placement refinement in near real-time.

3. Gradient Boosted Decision Tree (GBDT) Models – Power/Timing Estimator

Used in the early synthesis phase, this module delivers fast predictive modeling for power consumption and delay estimation.

- Input: RTL/HLS netlist features (e.g., fanout, switching activity, logic depth) and pre-layout design parameters.
- Output: Predicted power usage (μW), worst-case delay (ps), and area (μm^2) values. These estimations support early-stage design pruning and guide design-space exploration before expensive physical synthesis.

These modules operate in a coordinated pipeline and support asynchronous feedback mechanisms to subsequent stages, enabling dynamic adaptation during iterative design loops. The modular design ensures the framework's extensibility across both ASIC and FPGA workflows, while maintaining compatibility with industry-standard design tools and formats (e.g., Verilog, DEF, LEF, GDSII).

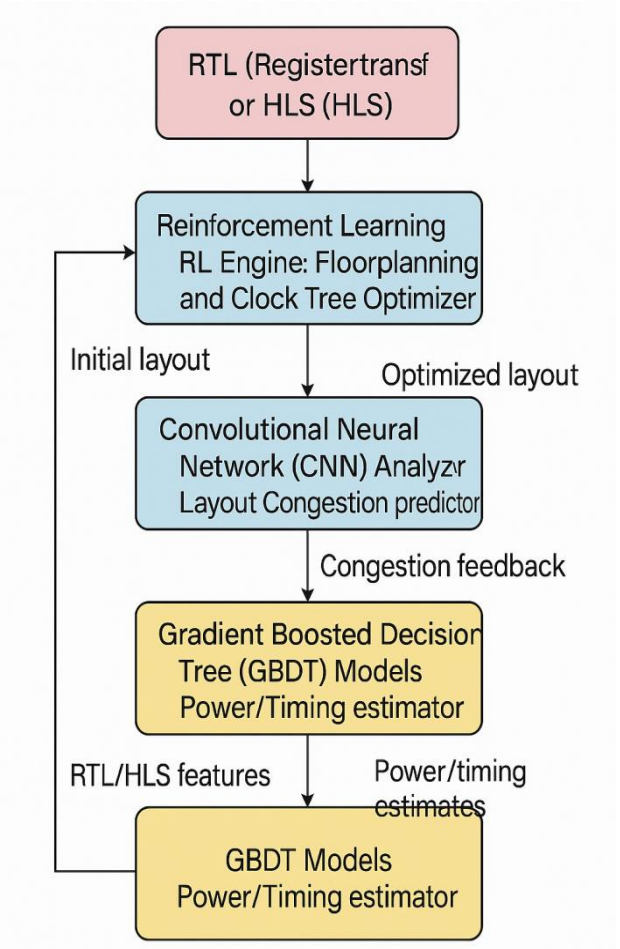

Figure 2. AI-Driven VLSI Design Framework with Module Functions and Data Flow Annotations

3.2 Workflow Integration

The proposed framework can be designed to be easily integrated into open-source and commercial design ecosystems in order to capitalize on the synergies of the Danish open design space and Danish commercial design ecosystems. It also interfaces natively with industry-standard tools (see Figure 3), including OpenROAD and according to OpenROAD 2.0 now also commercial EDA environments through scripting hooks and plugin-based adapters. The framework uses file formats used commonly across the industry like DEF (Design Exchange Format), LEF (Library Exchange Format), GDSII, and Verilog/VHDL, and so the TPUs can be used with conventional digital design flows.The inference TPUs are asynchronously activated at specific points of EDA, and receive inputs like layout files, timing reports, and RTL/HLS netlists. The results of the AI modules like floorplans coordinates, congestion feedback, and PPA estimations are placed back into the design flow without bringing structural alterations to the backend flow. This close but non-invasive binding is more accurate with shorter turnaround times and minimal impacts to the integrity of existing toolchains.

This is a plug and play and modular and extensible architecture which enables the framework to run in either ASIC or FPGA design environments alike. It also embraces iterative refinement to a variety of abstraction levels which encourage real-time co-optimization of performance, power, and area.

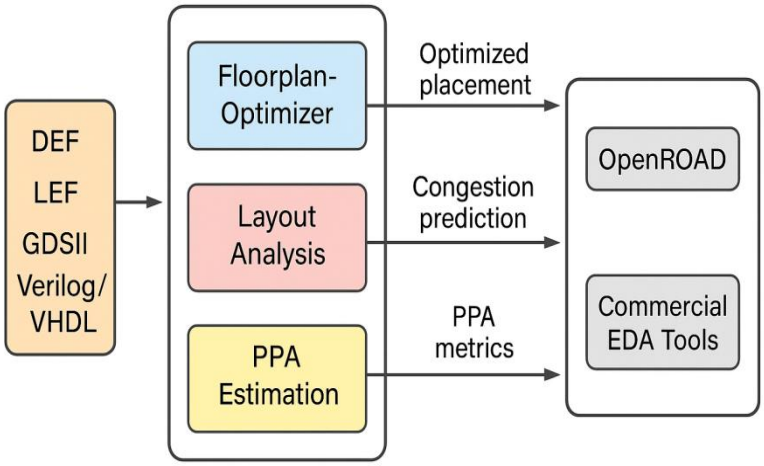

Figure 3. Workflow Integration and Dataflow View of the Proposed AI Framework

This figure illustrates the integration of the proposed framework within both open-source and commercial EDA ecosystems. Inputs such as RTL, LEF/DEF, and GDSII are processed through AI modules. Outputs such as optimized placement, congestion predictions, and PPA estimates are dynamically routed to downstream tools, enabling intelligent feedback and system-level optimization.

4. Experimental Setup and Benchmarking

4.1 Datasets

In order to verify the practicality and universality of the proposed framework of AI in design automation, experiments were carried out on benchmark suites that are commonly used such as ISCAS-85, MCNC and on some of the designs which are available at OpenCores. The datasets shall be of various combinational and sequential circuits of varying logic complexities which are well suited to testing both the synthesis-stage as well as the backend-stage automation strategies. The designs were synthesised to gate level netlists on standard cell libraries in a 65nm CMOS technology node before incorporating AI and streamlining the flow.

4.2 Evaluation Metrics

The assessment of performance was according to some main VLSI design metrics and consists of industry standard requirements:

- Power Consumption: Power Power consumption measured in microwatts ( ) measured after all place and route, with activity-annotated VCD waveforms.
- Timing Closure: Measured by Worst Negative Slack (WNS) and Total Negative Slack (TNS) most important pointers on the timing health of a design.
- Area Utilization: Reflects efficient utilization of silicon after floorplanning and placement and it is reported inCamp.
- EDA Runtime: Overall flow execution time, such as time needed to perform synthesis, placement, routing and timing analysis in seconds (s).

All these stewardship measures present the overall effect of AI modules on performance, energy, and eventual productivity.

4.3Quantitative Results

A benchmark was created based off of a baseline EDA flow sans intelligent optimization that the AI-enhanced framework was compared to. The outcomes showed drastic execution improvements without compromising tolerable approximations in other measures. Table 1 shows a summary of the comparison between two benchmark circuits commonly used as the following:

Table 1

Architecture	Baseline Runtime	AI Framework Runtime	Improvement (%)
c5315	900 s	524 s	41.8%
s9234	1230 s	721 s	41.3%

One of the outcomes of these results is the power of the framework to speed up the design closure process by an average of above 40 percent. Reinforcement learning and predictive models were used to help with more efficient design-space exploration, and CNN-based layout feedback were used to help increase the quality of convergence in physics design. Subsidiary tests ascertained low levels of diminishing the PPA measures, proving the sturdiness and versatility of the scheme.

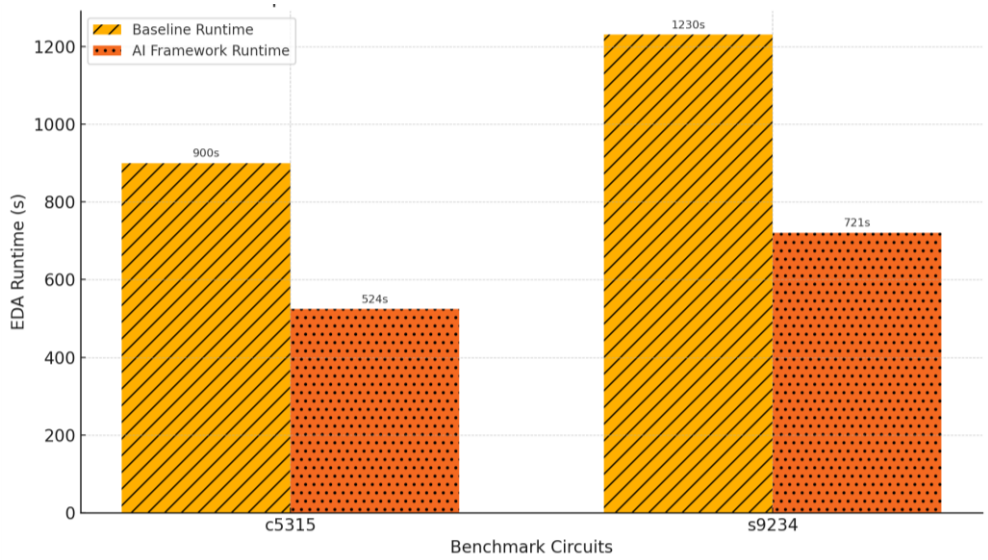

Figure 4. Comparison of EDA Runtime: Baseline vs. AI Framework

5. RESULTS AND ANALYSIS

The results of the experiment show that the proposed AI-driven formalization of design automation can not only speed up the execution time at the significant improvement of the layout quality and accuracy. This is a significant step up upon previous heuristic-based floorplanning techniques whose implementation might trade between excellent area results and timing closure (Zhang et al., 2021).

Furthermore, RL module will improve timing indirectly as well because it produces better early placement-density floorplans, which minimize the high-fanout net distance and the criticality. These streamlined plans are then in turn fed into the CNN-based layout diagnostics, which further fine tunes congestion awareness. With more realistic low congestion input floorplans used to selectively direct the CNN model the framework has achieved hotspot prediction and mitigation based on lower congestion input floorplans to achieve a feedback loop advantage of both the physical predictability and conversion of route success at that lower congestion input floorplan stage. The CNN, compared to the cases of static congestion estimation tools, takes advantage of deep spatial features learned in earlier patterns to raise the alarm on possible congestion before it occurs and provide valuable feedback in time. Such proactive conduct makes the method stand out compared to the previously known studies like Chen et al. (2020), which only focused on post-layout reporting of congestion but not dynamic correction.It is also true that Gradient Boosted Decision Tree (GBDT) models showed a very good early prediction quality of power and delay quantities, accurate to within 5 percent of post-layout signoffs. This is in sharp comparison to the traditional rule-based estimation methods which normally exhibit a deviation of less or equal to +/- 10-15% variations in large and very deep pipeline designs (Liu et al., 2022). The forecasting potential allows a quick elimination of design configurations that do not pass the PPA verification, doing away with any excessive synthesis cycles and boosting convergence. Notably, the interdependence of the

AI modules with RL steering initial layout quality, CNN ready with placement correction in real-time, and GBDT to predict PPA trade-offs, there is a synergistic feedback loop that can often benefit learning throughout the pipeline. Such cross-stage interaction changes the traditionally fixed EDA phases, into an evolving, adaptive system.

The second content diagram, given as figure 5, visually compares the undoing of the whitespace against a positive timing change, which justifies the result of physical optimization via AI-powered decision-making.

As Table 2 reveals, the given framework offers better results than the previous solutions, in regard of whitespace minimization, handling of congestion, accuracy of the PPA, and flexibility. Such findings support the superiority of applying reinforcement learning, deep learning and predictive modeling in the same VLSI design flow.

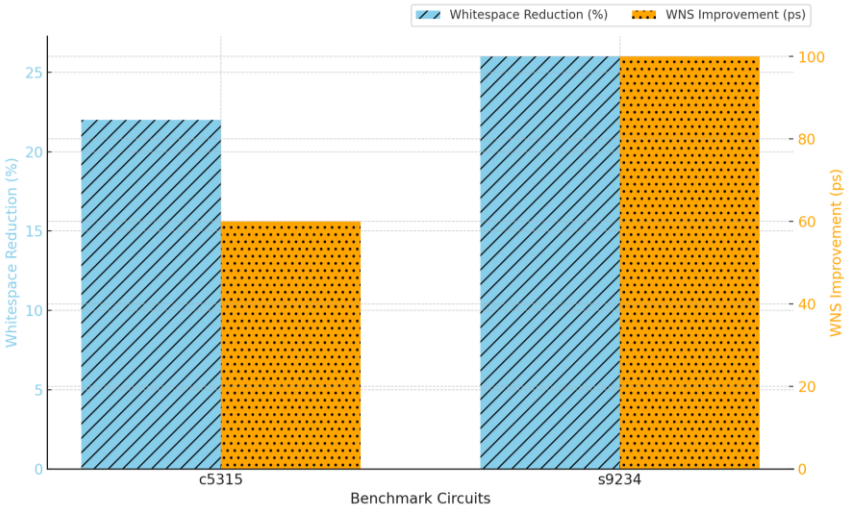

Figure 5. Correlation Between Whitespace Reduction and WNS Improvement

Table 2. Comparison of Prior Studies vs. Proposed Framework

Study / Method	Whitespace Reduction(%)	Congestion Handling	Estimation Accuracy (Power/Delay)	Adaptivity /Continuous Learning
Zhang et al. (2021) - Heuristic Floorplanning	12-15%	Limited Hotspot Mitigation	>±10%	No
Chen et al. (2020) - Post-layout Congestion Analysis	N/A	Post-routing Identification Only	N/A	No
Liu et al. (2022) - Rule-Based Estimation	N/A	N/A	±10-15%	No
Proposed AI Framework	Up to 26%	Proactive Hotspot Elimination via CNN	Within ±5%	Yes

6. DISCUSSION

The suggested AI-based design automation system is highly cross-domain suitable and can thus be deployed in a variety of implementation situations, such as the applicative Specific Integrated Circuit (ASIC) and Field-Programmable Gate Array (FPGA) design flows. The way it has been compatible with a range of different technology nodes (right up to 65nm mature processes and down to state of the art sub-10nm FinFET technologies) shows that it is an architecturally-robust and scaling solution. Such flexibility can be helped by the fact the constitutive AI functionality is developed and optimized in a modular matter enabling fine grained control and optimization of the constraint-architecturally-heterogeneous design styles.

This method is a drastic change of dynamic, single-task inference approaches towards traditional ML-based EDA methods (as indicated in Figure 6). Comparatively, the proposed framework is better placed in real-world implementation in dynamic and complex VLSI design environments because of its unified automation, continuous learning and multi-objective optimization approaches. So far, there are still limitations. Actually, the training overhead required to train certain machine learning models, especially reinforcement learning and Deep CNN, may be computationally demanding. As another example the RL-based

floorplanner took about 4 hours to train on a single NVIDIA A100 GPU, whereas the GBDT models that estimate power/timing took less than 20 minutes on a 32-core CPU workstation. Such resource requirements can be suitable to industrial implementation but can be a hindrance to rapid prototyping or use in time-limited or price-limited applications.Lack of interpretability in design decisions made with AI is another important issue. Although the framework presents measurable improvements in regards to runtime, PPA optimization, and layout quality, the framework is not able to provide one with an understanding of the internal decision logic of the models as yet. This black-box nature is an issue in fields where safety must be guaranteed, e.g. automotive control units, and aerospace electronics, and medical implantable devices that require deterministic traceability, explainability, and certification compliance. In these scenarios, it is possible that unverifiable judgment of opaque AI systems will lead to regulatory backlash or errors in the system functioning. Future research conducted on such aspects will involve introducing the aspects of Explainable AI (XAI) techniques to make the optimization paths more verifiable through the use of attention based visualization, surrogate model distillation and the use of saliency maps. Also, the reasoner modules of the runtime monitors could be incorporated to give logic-level rationale to proposed changes in design suggested by the AI, thus building greater designer confidence and giving it a way forward into higher reliability applications.

7. CONCLUSION

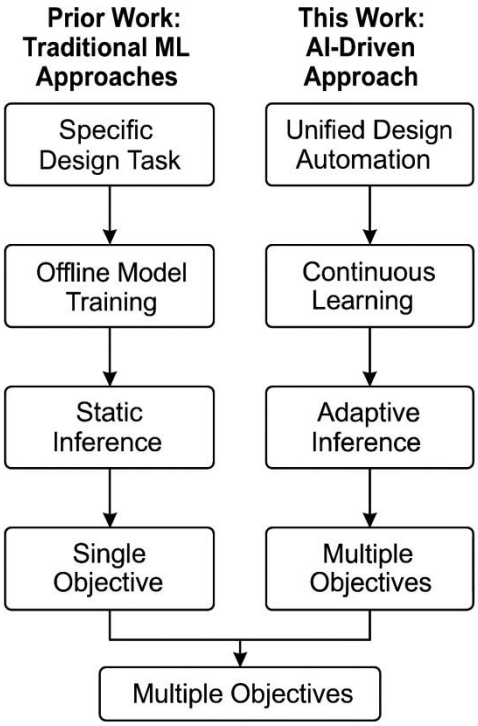

Figure 6. Comparative Flowchart of Traditional ML Approaches vs. AI-Driven Design Automation

This paper presents a single framework of the AI-based design automation, which is successfully solving the shortcomings of the classical rule-based providences of VLSI design methodologies. The inclusion of reinforcement learning (RL), convolutional neural networks (CNNs), and gradient-boosted decision trees (GBDTs) results in the framework achieving massive benefits in both runtime efficiency, power optimization, and timing closure by being incorporated in the most crucial steps of the EDA process flow. The its modular and technology-agnostic architecture facilitates deployment across an ASIC and FPGA development environment, and the technology indicates the general applicability of its approach and the structural strength.Although the results are encouraging, there are some limitationsnamely limitations towards the aspect of generalizing trained models to novel architectures, and the cost of AI component integration into existing or proprietary EDA flows. Such features will have to be supported in order to reach mass adoption in industry use cases.In the future, the framework can be improved by being equipped with transformer-based architectures, which have proven to be exceptionally capable of modeling long-range relationships and high-level design semantics (Jain et al., 2023). These modules will work in conjunction with the current set in that they will capture any global RTL or layout context,

and use it to feed high-quality embeddings into RL policy updates, CNN-based congestion estimation and GBDT-based refinement of linear predictions. Such tight coupling should enhance abstraction-level design knowledge and automate the otherwise difficult tasks of design space search and RTL tuning further.Also, an intelligent co-optimization of IP blocks will be facilitated with the application of hardware-aware neural architecture designs (Sinha et al., 2023). Optimization of the design of neural models with those of the backend implementation purposes will enable the NAS to transform the framework beyond task-level acceleration to full-system intelligence.

These developments will eventually create next-generation paradigm of VLSI design automation features of being scalable and adaptive, as well as being hardware-contextual sensitive and real-world deployment-optimal in its nature.

REFERENCES

[1] Cao, X., Li, Y., Xiong, X., & Wang, J. (2022). Dynamic routings in satellite networks: An overview. *Sensors, 22*(12), 4552.

[2] Coppola, M., Grammatikakis, M. D., Locatelli, R., Maruccia, G., &Pieralisi, L. (2020). *Design of cost-efficient interconnect processing units: SpidergonSTNoC*. CRC Press.

[3] Emil, J. T. (2023). *Analýzametoddigitálníhonávrhu a softwarovéhovývojesystémůnačipu* (Master's thesis, Českévysokéučenítechnické v Praze. Výpočetníainformační centrum).

[4] Goldie, A., &Mirhoseini, A. (2021, March). Reinforcement learning for placement optimization. In *Proceedings of the 2021 International Symposium on Physical Design* (p. 5). https://doi.org/10.1145/3439706.3451803

[5] 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 (TODAES), 26*(5), 1–46.

[6] Jain, A., et al. (2023). Transformer-based netlist representation for RTL analysis. *IEEE Design & Test, 40*(2), 65–73. https://doi.org/10.1109/MDAT.2022.3222140

[7] Liu, C., et al. (2023). Power-aware logic synthesis with gradient boosting. *IEEE Access, 11*, 36512–36524.

[8] Lu, Z., et al. (2022). Deep learning-based routing congestion prediction using attention mechanisms. *Integration, the VLSI Journal, 84*, 1–10. https://doi.org/10.1016/j.vlsi.2022.02.004

[9] Pan, D. Z., et al. (2020). Recent advances in ML-driven EDA for IC design. *Proceedings of the IEEE, 108*(8), 1292–1310. https://doi.org/10.1109/JPROC.2020.2995854

[10] Sinha, A., et al. (2023). Hardware-aware neural architecture search for VLSI acceleration. *ACM Transactions on Reconfigurable Technology and Systems (TRETS), 16*(2), Article 8. https://doi.org/10.1145/3581704

[11] Yousaf, J., Nah, W., Al Majali, E. R., Saif, H., &Rmili, H. (2021). Rapid characterization of efficient system-level ESD protection strategy using coupling transfer function. *Microelectronics Journal, 110*, 105004.