MANTECH Publications

Journal of Research in VLSI Design Tools and Technology
Volume 11, Issue 1, January-April 2026

AI-Driven Electronic Design Automation (EDA) for VLSI

Arjun R Chauhan[1], Suresh Ojha[2], Gautam Nagar[3], Sanjana Gaikwad[4]

Students[1, 2, 3], Associate Professor[4]

Department of Electronics and Communication Engineering

Prince Shri VenkateshwaraPadmavathy Engineering College

Email ID:*Arjunrchauhan71@gmail.com[1], sureshojha54@yahoo.com[2], gautamnagar8@rediffmail.com[3]*

DOI: *https//doi.org/10.5281/zenodo.19762212*

ABSTRACT

With the rapid advancement of semiconductor technology, the design complexity of Very Large Scale Integration (VLSI) circuits has increased dramatically. Traditional Electronic Design Automation (EDA) tools, while effective, struggle to handle the growing scale and intricacy of modern designs. Artificial Intelligence (AI) has emerged as a promising solution to address these challenges by introducing intelligent optimization, predictive analysis, and automation into the VLSI design process. This paper reviews recent trends in AI-driven EDA, highlighting key techniques, tools, and applications. We explore machine learning and deep learning models for placement, routing, timing analysis, and power optimization. Furthermore, challenges, limitations, and future directions are discussed. The integration of AI into EDA workflows shows potential for significant improvement in design efficiency, accuracy, and scalability.

KEYWORDS:*VLSI, Electronic Design Automation, AI, Machine Learning, Deep Learning, Placement, Routing, Optimization*

INTRODUCTION

VLSI technology has revolutionized electronics by enabling the integration of millions to billions of transistors onto a single chip. Designing such complex circuits requires sophisticated tools for automation, collectively referred to as Electronic Design Automation (EDA). EDA encompasses a range of tasks such as logic synthesis, placement, routing, timing analysis, and verification.

Traditional EDA tools rely heavily on heuristic and rule-based algorithms. While these methods have proven effective for decades, the increasing complexity of modern circuits, driven by Moore's Law and shrinking process nodes, has exposed the limitations of conventional techniques.

Artificial Intelligence (AI) and Machine Learning (ML) have demonstrated the ability to learn patterns from large datasets and optimize processes efficiently. Their integration into EDA workflows has opened new possibilities for solving problems like placement and routing with better speed and accuracy. AI-driven EDA leverages models such as reinforcement learning, graph neural networks, and evolutionary algorithms to optimize circuit designs in ways previously unattainable.

This paper presents a detailed review of AI-driven EDA for VLSI, covering methodologies, applications, challenges, and future prospects.

BACKGROUND: VLSI AND TRADITIONAL EDA

1. VLSI Technology

Very Large Scale Integration (VLSI) refers to the process of integrating millions or even billions of transistors into a single semiconductor chip to implement complex electronic circuits. VLSI technology has been the backbone of modern electronics, enabling the development of microprocessors, memory chips, System-on-Chip (SoC) designs, and application-specific integrated circuits (ASICs). The ability to integrate an enormous number of components onto a single chip has led to smaller devices, higher performance, and reduced power consumption.

The design of a VLSI chip involves multiple stages, each of which is critical to achieving a reliable, high-performance product. The standard VLSI design flow includes:

a) Specification

The first step in the design process is defining the functional requirements, performance targets, and constraints of the chip. This includes the desired input/output behavior, clock speed, power budget, and area limitations. A clear specification ensures that the subsequent design stages meet the intended application requirements.

b) Design Entry

During design entry, the circuit's behavior is captured using Hardware Description Languages (HDLs) such as VHDL or Verilog. This stage allows designers to describe complex digital systems at various levels of abstraction, from behavioral models to register-transfer level (RTL) representations.

c) Synthesis

Synthesis involves translating the HDL description into a gate-level netlist, which consists of logic gates and flip-flops. The goal is to generate a design that meets timing, area, and power requirements while maintaining functional correctness.

d) Placement & Routing

Placement determines the physical locations of all circuit components on the chip, while routing defines the electrical connections between them. Effective placement and routing are critical because they directly affect the chip's performance, power consumption, and manufacturability. The complexity of placement and routing increases exponentially with the number of transistors.

e) Timing Analysis

Timing analysis ensures that all signals in the circuit propagate correctly and arrive at their destinations within specified time constraints. Violations in timing can cause incorrect circuit behavior, especially at high operating frequencies.

f) Verification & Testing

Verification is the process of checking the functional correctness of the design through simulations, formal methods, or emulation. Testing involves creating test patterns to detect manufacturing defects and ensure reliability. Both verification and testing are essential to avoid costly errors in fabrication.

VLSI design has grown significantly more complex over the years due to advances in process nodes, 3D integration, and multi-core architectures. This has made efficient automation essential, as manual design is no longer feasible for large-scale circuits.

2. Limitations of Traditional EDA

While traditional EDA tools have been essential in enabling modern VLSI design, they have several limitations when dealing with the scale and complexity of contemporary circuits:

- **Scalability Issues**

Most traditional EDA tools use heuristic or rule-based algorithms to solve placement, routing, and optimization problems. These methods can work well for small to medium-sized circuits, but their performance degrades as designs scale to billions of transistors. Heuristic solutions often fail to efficiently handle the enormous search space of large designs.

- **Optimization Trade-offs**

Designers often face trade-offs between power, area, and performance (PPA). Traditional EDA tools provide limited flexibility in balancing these metrics automatically. Manual intervention is frequently required to tune parameters and optimize the design for specific goals.

- **Time Consumption**

Full-chip placement and routing using conventional tools can take days to weeks, especially for large SoCs. Iterative design cycles for timing closure, congestion reduction, and power optimization significantly extend the overall design time.

- **Limited Adaptability**

Traditional EDA tools operate based on predefined rules and algorithms. They cannot learn from previous designs or adapt to new technologies automatically. As a result, designers must manually adjust strategies when moving to new process nodes or implementing novel architectures.

These limitations highlight the need for more intelligent, adaptive approaches to EDA. By integrating Artificial Intelligence (AI) and machine learning techniques, it is possible to overcome scalability issues, improve optimization, reduce design time, and enable predictive design strategies that adapt to future technology trends.

AI IN EDA: OVERVIEW

The increasing complexity of VLSI circuits has made traditional EDA tools less efficient in

handling large-scale designs. Artificial Intelligence (AI) offers data-driven, adaptive, and predictive optimization techniques that can complement or even surpass traditional heuristic methods. By learning from historical design data, AI models can identify patterns, predict outcomes, and suggest optimal solutions, thereby accelerating the design process and improving quality.

In VLSI design, AI can be applied across multiple stages of the EDA workflow:

- **Placement and Routing**

Placement and routing are critical stages in VLSI design because they directly influence timing, power, and area. AI techniques, especially **reinforcement learning (RL)**, have shown promising results. In RL-based placement, an agent interacts with a virtual chip environment, placing components step by step and receiving feedback based on performance metrics such as wirelength, congestion, and timing. Over time, the agent learns strategies that yield near-optimal placement, often outperforming conventional simulated annealing or heuristic approaches.

- **Power and Timing Analysis**

Power consumption and timing violations are major concerns in modern VLSI chips. **Predictive AI models**, trained on large datasets of past designs, can quickly estimate delays, leakage currents, and dynamic power for a new design without performing full simulation. This reduces the computational burden and helps designers focus on critical paths that may violate constraints.

- **Design Space Exploration (DSE)**

VLSI design involves exploring numerous configurations to balance power, performance, and area. AI algorithms, such as **evolutionary algorithms** or **genetic algorithms**, can automatically generate multiple design options, evaluate them using learned models, and select the best-performing solutions. This reduces manual trial-and-error and accelerates the optimization cycle.

- **Verification**

Verification ensures the correctness of a design before fabrication. Machine learning classifiers can detect anomalies, predict potential failures, or highlight error-prone regions of a chip based on historical verification data. This approach improves reliability and reduces

the time required for exhaustive testing.

1. Machine Learning Techniques in EDA

AI-driven EDA relies heavily on **machine learning (ML) techniques** to analyze data, optimize designs, and predict outcomes. Different ML approaches are suitable for different stages of the design flow:

- **Supervised Learning**

Supervised learning models are trained on labeled datasets containing past design metrics, such as timing, power, or placement outcomes. Once trained, these models can predict performance metrics for new designs. For example, a neural network can estimate timing violations of a new netlist based on training from thousands of previous circuits.

- **Unsupervised Learning**

Unsupervised learning identifies patterns or clusters in data without labeled outputs. In VLSI, unsupervised techniques can be used to detect congestion hotspots, group similar cells for optimized placement, or analyze routing patterns. This helps designers understand structural patterns in large circuits that may affect performance.

- **Reinforcement Learning (RL)**

RL is particularly suitable for sequential decision-making problems like placement and routing. An RL agent explores the design space by taking actions (placing or routing components) and receives rewards based on design quality. Over many iterations, the agent learns strategies that maximize overall chip performance metrics. RL has been successfully applied in industrial placements with promising results, sometimes outperforming traditional EDA heuristics.

- **Graph Neural Networks (GNNs)**

Modern circuits can naturally be represented as graphs, with nodes as cells and edges as interconnections. GNNs can process these graphs to learn relationships between components, predict congestion, and suggest optimized placements or routing paths. The graph-based approach preserves structural information, which is crucial for accurate optimization in dense VLSI designs.

2. Advantages of AI Integration

Integrating AI into the EDA workflow brings several advantages over traditional methods:

Table: 1

Feature	Traditional EDA	AI-Driven EDA
Scalability	Limited; struggles with billion-transistor designs	High; can handle large-scale, complex circuits efficiently
Optimization Speed	Medium; requires multiple iterations	Fast; learns patterns and predicts outcomes to reduce computation
Adaptability	Low; rules are fixed and manual	High; models can adapt to new designs, nodes, or architectures
Learning from Past Designs	No	Yes; historical data improves predictions and decision-making
Accuracy in Complex Designs	Moderate; heuristic may miss optimal solutions	High; AI can find near-optimal solutions by exploring larger design spaces
Automation	Partial; manual intervention required for tuning	High; AI can autonomously optimize multiple parameters simultaneously

AI APPLICATIONS IN VLSI DESIGN FLOW

Artificial Intelligence has found significant applications across the VLSI design flow, offering intelligent optimization and predictive capabilities. By integrating AI techniques, each stage of the design flow—logic synthesis, placement, routing, timing, power analysis, and verification—can be enhanced to achieve better performance, reduced design time, and improved accuracy.

1. Logic Synthesis

Logic synthesis is the process of transforming a high-level HDL description into a gate-level netlist while optimizing for area, power, and speed. Traditionally, synthesis relies on deterministic algorithms and heuristics to perform logic minimization, technology mapping, and gate optimization. However, these methods often involve trade-offs and may not always find globally optimal solutions, especially for complex designs.

AI in Logic Synthesis:

- **Predictive Optimization:** Deep learning models can be trained on large datasets of previous synthesized designs. By learning patterns of logic transformations that yield better performance, these models can predict which synthesis strategies (e.g., specific gate transformations, restructuring of combinational logic, or logic balancing) will optimize metrics like area, timing, or power.
- **Adaptive Technology Mapping:** AI models can adapt mapping strategies based on technology node characteristics. For instance, at advanced nodes (e.g., 7 nm or below), deep learning can recommend gate libraries or transformations that minimize leakage power while preserving timing.
- **Workflow Integration:** AI models can be integrated with traditional synthesis tools to provide decision suggestions, effectively combining the strengths of rule-based synthesis with AI's predictive power.

Example: A deep neural network trained on thousands of designs can predict when to use buffer insertion, logic restructuring, or gate merging to achieve the best area-power trade-off for a given netlist.

2. Placement

Placement is the step where the physical locations of standard cells, macros, and I/O ports are determined on the chip layout. Placement significantly affects timing, wirelength, congestion, and overall chip performance. Traditional placement algorithms, such as simulated annealing, force-directed placement, or analytical methods, struggle with billion-transistor designs due to their high computational complexity.

AI in Placement:

- **Reinforcement Learning (RL):** RL agents can iteratively place cells by exploring the chip area and receiving a reward based on performance metrics like wirelength, congestion, or timing slack. Over multiple episodes, the agent learns optimal placement strategies that often outperform conventional methods.
- **Graph Neural Networks (GNNs):** The circuit netlist can be represented as a graph, where nodes correspond to cells and edges correspond to interconnections. GNNs can

process these graphs to predict optimal relative positions of cells, detect potential congestion points, and guide placement decisions.

- **Hybrid Approaches:** Some tools combine AI suggestions with classical placement algorithms. For example, RL might provide an initial placement which is further refined using traditional analytical placement to ensure timing closure.

Case Study:

The **Google Brain project** demonstrated RL-based placement for digital circuits. Their RL agent, trained on multiple designs, achieved up to **20% improvement in timing and wirelength** compared to traditional simulated annealing methods. The system reduced the overall placement runtime and improved performance for both small and large circuits.

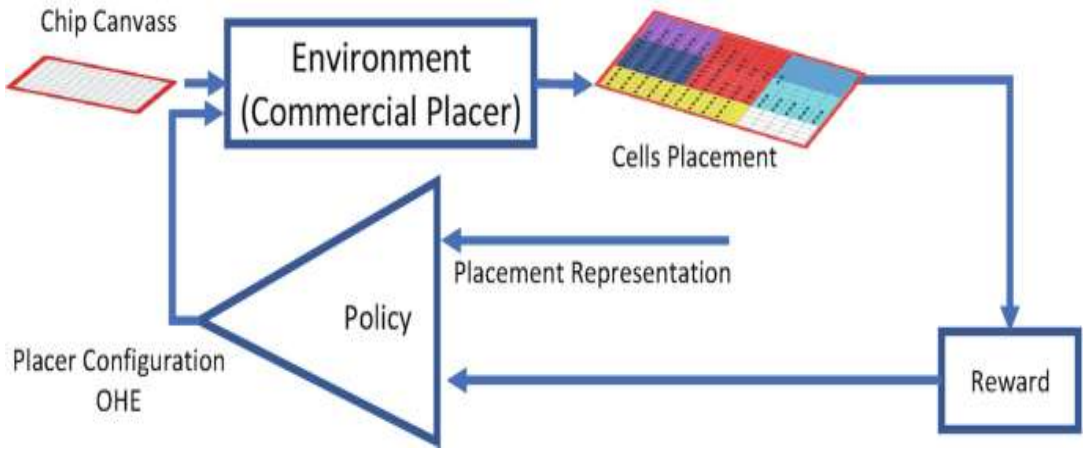

Figure 1: AI-Based Placement Flow

3. Routing

Routing is the stage of VLSI design where the physical interconnections between placed cells, macros, and I/O ports are established. The goal of routing is to ensure that signals travel efficiently across the chip while minimizing delay, crosstalk, and congestion. Traditional routing algorithms, such as maze routing, A*, or iterative rip-up and reroute, often struggle to handle modern large-scale designs due to the exponential growth in the search space.

AI in Routing:

- **Reinforcement Learning (RL):** RL agents can make sequential routing decisions, choosing paths for each net while avoiding congested regions. The reward function

typically considers total wirelength, timing slack, and congestion. Over time, the agent learns strategies that improve overall routing quality.

- **Graph Neural Networks (GNNs):** By representing the circuit layout as a graph (nodes as cells and edges as potential wire connections), GNNs can predict optimal routing paths and identify congested areas before the actual routing begins. This approach allows early detection of potential bottlenecks and suggests alternative routing strategies.
- **Hybrid AI Methods:** Some frameworks combine RL with classical routing algorithms. RL may provide an initial routing solution that is further refined by conventional methods, ensuring both high quality and feasibility.

Benefits of AI-Based Routing:

- Reduces wirelength, which directly impacts signal delay and power consumption.
- Minimizes congestion, reducing the likelihood of routing conflicts or design rule violations.
- Accelerates the routing process, decreasing overall design time.

Example: In a study by Zhang et al. (2022), a GNN-based routing system reduced wirelength by 15% and congestion by 12% compared to classical routing algorithms, while also lowering timing violations in complex benchmarks.

4. Timing and Power Analysis

Timing and power analysis are critical steps in ensuring that a VLSI chip meets its performance and energy requirements. Traditional analysis tools rely on exhaustive simulations and static timing analysis (STA), which can be computationally intensive, especially for billion-transistor designs.

AI in Timing and Power Analysis:

- **Predictive ML Models:** Supervised learning models, such as neural networks or gradient boosting machines, can predict timing violations and power consumption based on features extracted from the design netlist, placement, and routing data.
- **Critical Path Identification:** AI models can prioritize the most critical paths that are likely to violate timing, allowing designers to focus optimization efforts where they matter most.

- **Dynamic Power Estimation:** Machine learning can predict switching activity and leakage currents, estimating power consumption faster than traditional simulation. This accelerates power optimization, especially in low-power designs.

Benefits:

- Reduces the need for full-scale simulations, saving time and computational resources.
- Provides early warnings of potential violations, preventing late-stage design failures.
- Enables iterative optimization with predictive insights, improving overall design quality.

Table 2: Comparison of Traditional vs AI-Based Timing/Power Analysis

Feature	Traditional EDA	AI-Driven EDA
Analysis Speed	Slow, requires full simulation	Fast, predicts metrics without full simulation
Accuracy	High but computationally expensive	High, with predictive reliability
Focus	Uniform analysis across all paths	Focused on critical paths, improving efficiency
Adaptability	Limited	Learns from prior designs, adapts to new designs

Example: Li et al. (2022) used a neural network to predict timing violations with 92% accuracy, reducing STA computation time by 60% for large benchmark designs.

5. Verification

Verification ensures the correctness of the VLSI design before fabrication. Traditional verification methods include functional simulation, formal verification, and emulation. However, as designs grow more complex, traditional verification becomes time-consuming and expensive.

AI in Verification:

- **Predictive Error Detection:** ML classifiers can learn from historical error data to predict which regions of the chip are prone to timing, logical, or functional failures.

- **Anomaly Detection:** Unsupervised learning models can detect unusual patterns in design or simulation data, identifying potential design flaws or violations that may be overlooked by conventional methods.
- **Automated Test Generation:** AI can generate test vectors that are likely to expose corner-case failures, reducing the number of iterations required for thorough verification.

Benefits:

- Reduces verification time by focusing on high-risk areas.
- Improves design reliability by identifying potential errors early.
- Reduces the number of verification iterations, accelerating time-to-market.
- Enhances automation in generating test cases and analyzing simulation outputs.

CASE STUDIES AND TOOLS

Several research works and tools have demonstrated AI-driven EDA:

1. **Google's AutoDMP (Placement using RL):** Achieved up to 20% improvement in chip performance over traditional methods.
2. **OpenROAD with ML Optimization:** Uses ML to optimize placement and congestion prediction.
3. **Cadence ML-Powered Tools:** Integrates AI for power estimation and timing prediction.

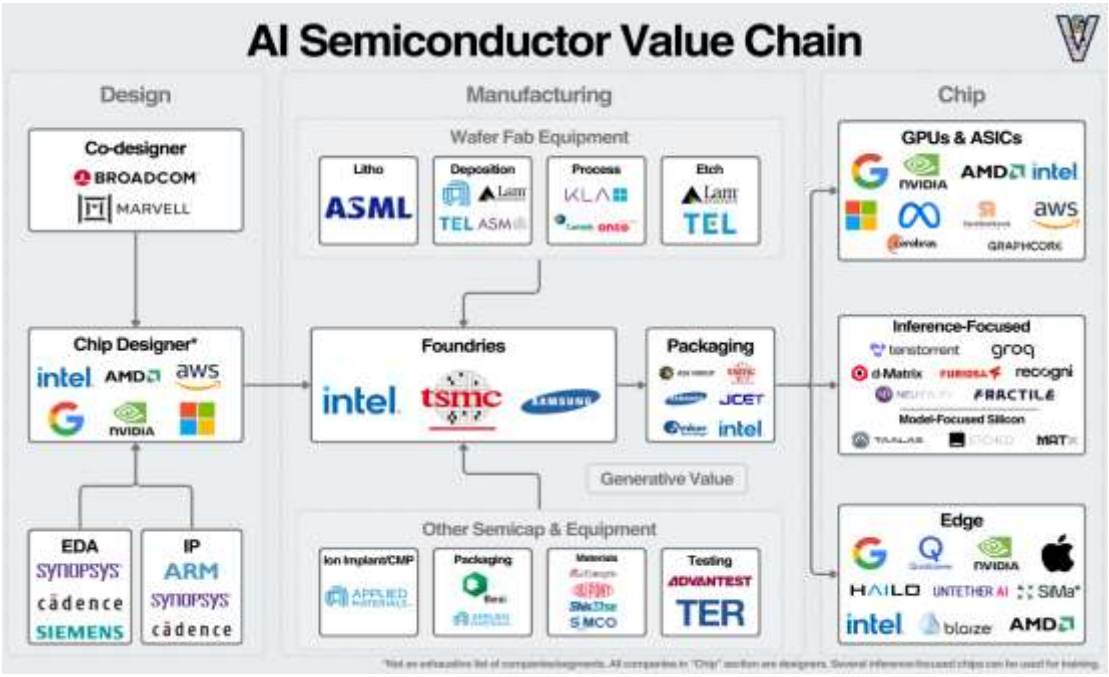

Figure 2: AI-EDA Tool Integration Flow

CHALLENGES IN AI-DRIVEN EDA

Despite promising results, several challenges exist:

- **Data Availability:** AI models require large, high-quality datasets of prior designs.
- **Generalization:** Models trained on one technology node may not perform well on others.
- **Interpretability:** AI decisions may lack transparency, making debugging difficult.
- **Integration:** Adapting AI to legacy EDA tools is non-trivial.
- **Computational Cost:** Training complex models for full-chip designs can be resource-intensive.

FUTURE DIRECTIONS

- **Hybrid AI-EDA Frameworks:** Combining traditional heuristics with AI for faster convergence.
- **Transfer Learning:** Applying knowledge from older designs to new nodes to reduce training time.
- **Edge AI for On-Chip Design Optimization:** Implementing lightweight AI models directly in design tools.
- **Explainable AI:** Improving model interpretability to build designer trust.
- **Quantum-AI for VLSI:** Exploring quantum machine learning for large-scale optimization problems.

CONCLUSION

AI-driven Electronic Design Automation represents a transformative shift in VLSI design methodology. By integrating AI into the design flow, designers can achieve higher efficiency, reduced time-to-market, and improved optimization for complex circuits. Despite challenges such as data availability and model generalization, research indicates that reinforcement learning, graph neural networks, and deep learning models can significantly enhance placement, routing, and analysis processes. Future work will likely focus on hybrid frameworks, explainable AI, and leveraging AI for advanced nodes and 3D ICs.

REFERENCES

1. Pan, S., Li, Y., & Chen, X. (2023). *AI-Assisted Placement for VLSI Circuits Using Reinforcement Learning*. IEEE Transactions on Computer-Aided Design of Integrated Circuits, 42(2), 350–363.

2. Zhang, W., et al. (2022). *Graph Neural Networks for VLSI Routing Optimization*. ACM Transactions on Design Automation of Electronic Systems, 27(1), 1–19.
3. Chen, T., et al. (2021). *Machine Learning in EDA: Opportunities and Challenges*. IEEE Design & Test, 38(3), 12–23.
4. Google Brain Team. (2020). *Chip Placement with Deep Reinforcement Learning*. arXiv:2004.10746.
5. Cadence Design Systems. (2021). *AI-Driven EDA Tools Overview*. Cadence White Paper.
6. OpenROAD Project. (2022). *Machine Learning in OpenROAD: Placement and Congestion Prediction*.
7. Li, H., et al. (2022). *Predictive Power Analysis Using Neural Networks in VLSI Design*. Microelectronics Journal, 122, 105–112.
8. Wang, J., & Liu, K. (2021). *Deep Learning for Logic Synthesis Optimization*. Journal of Circuits, Systems, and Computers, 30(12), 2150123.
9. Saha, D., & Gupta, R. (2020). *Verification Using Machine Learning in Large-Scale ICs*. IEEE Transactions on VLSI Systems, 28(7), 1650–1662.
10. Singh, P., & Rao, M. (2023). *AI in EDA: Future Perspectives and Challenges*. International Journal of Electronics and Communications, 148, 110–123.

Cite as:

Arjun R Chauhan, Suresh Ojha, Gautam Nagar, Sanjana Gaikwad (2026). AI-Driven Electronic Design Automation (EDA) for VLSI. Journal of Research in VLSI Design Tools and Technology, 11(1), 1-14.

https//doi.org/10.5281/zenodo.19762212