Vol. 16 (2026): Proceedings of the 2026 International Conference on AI and Technological Innovation Applications (AITIA 2026)
E-ISSN: 3105-0522 P-ISSN: 3105-0514
DOI: https://doi.org/10.70267/aitia.2026266270

A Review of the Applications of Artificial Intelligence in the Full-Process Design of Integrated Circuits

Yuqing Song*

School of Shanghai Dianji University, 300 Shuihua Road, Pudong New Area, Shanghai

**Corresponding author: Yuqing Song.*

Abstract

In the post-Moore era, traditional integrated circuit (IC) design faces bottlenecks such as long design cycles, high costs, and difficulty in PPA convergence. Artificial intelligence (AI), with its data-driven and global optimization capabilities, has penetrated the entire IC design process. This paper systematically reviews the technological evolution of AI in full-process IC design, with emphasis on algorithmic representation mechanisms—from how Graph Neural Networks (GNNs) model congestion as node and edge features, to how Circuit Transformers capture long-range dependencies in directed acyclic graphs. We also analyze engineering practices of AI-EDA integration in industry and discuss technical challenges, including generalization, interpretability, and physical consistency, providing a reference for related research.

Keywords

artificial intelligence, integrated circuit design, EDA, large language model, reinforcement learning, graph neural network

1. Introduction

Integrated circuits (ICs) are the foundational components of the global information industry, and their design complexity grows exponentially as manufacturing process nodes shrink. In the conventional IC design workflow, front-end functional verification accounts for 40%-60% of total design person-hours, back-end placement and routing typically requires several weeks of iterative work, and the design iteration cycle for analog/RF circuits can extend to 12-18 months. The rapid development of AI technologies—including machine learning, deep learning, reinforcement learning, and large language models (LLMs)—has promoted a paradigm shift from “empirical art” to “data-driven science.” For instance, in the design of 7nm-class digital SoCs with netlists exceeding one million cells, AI-driven methods have achieved approximately 50%-70% reduction in design cycle time; for analog/RF circuits at 28nm and above, first-pass tape-out success rates have improved by over 30%. However, these gains are highly context-dependent and vary significantly across process nodes, circuit topologies, and design scales.

Top-tier IEEE journals and conferences have published substantial research on AI-IC design integration, while EDA enterprises have achieved industrial-scale application. Unlike previous surveys that focus primarily on task-level performance metrics, this paper emphasizes the algorithmic representation layer—how AI

models encode circuit structure and design constraints—and provides a unified perspective across the full design process. We analyze the mathematical and architectural logic underlying key AI techniques, examine industrial engineering practices, and identify algorithmic roots of current challenges, thereby offering a reference distinct from existing reviews.

2. AI in the Full-Process Design of Integrated Circuits

2.1 Front-End Design: Architecture Exploration, RTL Generation, and Functional Verification

The core objective of front-end design is to convert requirements into synthesizable RTL code and conduct comprehensive functional verification. AI addresses the demands of automated generation, intelligent verification, and efficient design space exploration.

Architecture Exploration. Traditional IC architecture design relies on senior engineers' experience, limiting exploration to a small number of feasible schemes. Generative AI integrates LLMs for requirement parsing with GNNs for architecture performance prediction. The key insight is that architecture exploration can be formulated as a search problem over a combinatorial space: LLMs generate candidate microarchitectures from natural language specifications, while reinforcement learning (RL) agents navigate this space by treating PPA metrics as reward signals. Wang et al. [4] proposed ChipGPT, an end-to-end framework that automates the process from natural-language requirements to RTL generation, demonstrating a 10× improvement in design efficiency in industrial RISC-V processor projects.

RTL Code Generation. LLMs learn from massive open-source HDL code repositories and design specifications, automatically converting algorithm descriptions into synthesizable RTL code (Verilog/VHDL). The underlying mechanism relies on the transformer's self-attention to capture hierarchical syntactic structures in hardware description languages, where module instantiation and port mapping follow regular patterns amenable to sequence modeling. Sun et al. [8] proposed ChipMind, a multi-agent LLM framework that covers both digital and analog design, reducing verification cycles by 50% in industrial projects.

Functional Verification. AI technologies address slow test case generation and difficult error localization through generative test case synthesis and coverage-driven optimization. Deep learning models (CNNs, RNNs) can predict waveform behavior and automatically localize errors, while LLMs generate test stimuli that maximize coverage. These approaches can shorten verification cycles by 30%-50% and improve functional coverage to over 99% for medium-complexity digital designs. Zhuo et al. [5] systematically reviewed machine learning applications in IC design, classifying the roles of supervised, unsupervised, and reinforcement learning in verification fault diagnosis and discussing Explainable AI (XAI) compliance requirements for high-reliability chips.

2.2 Back-End Design: Logic Synthesis, Placement and Routing, and Optimization

Back-end design determines the final PPA performance, where AI technologies have become standard through global optimization and fast prediction capabilities.

Logic Synthesis. Traditional logic synthesis relies on fixed heuristic algorithms that struggle with global optimization. The Circuit Transformer proposed by Chen et al. [6] represents a significant advancement: unlike conventional GNNs that aggregate information through neighborhood message passing—which can lose long-range structural dependencies in deep circuits—the transformer's multi-head self-attention mechanism directly computes pairwise relationships between all nodes in the Boolean network, enabling it to capture critical path dependencies across distant logic stages. After tape-out verification in a 7nm process, this technology achieved up to 64.9% area reduction and 22% improvement in critical path delay. Li et al. [2] confirmed that AI-driven methods can reduce chip area by 25% and power consumption by 18% in VLSI physical design.

Placement and Routing (P&R). P&R is the most time-consuming back-end link. AI-driven P&R leverages Deep Reinforcement Learning (DRL) and GNNs for automatic global optimization. Critically, the representation of the circuit layout as a graph is central to GNN-based approaches: each routing region is modeled as a node whose features encode local utilization rate, timing slack, and pin density, while edges carry features representing inter-region routing demand, track availability, and estimated wirelength. Through

iterative message passing, each node aggregates neighborhood congestion information, enabling the GNN to produce globally congestion-aware embeddings that guide the RL agent's placement decisions. Liu et al. [7] proposed a DRL-driven router integrated into Cadence's tool chain, reducing routing time from weeks to hours and congestion by 25%.

Timing/Power Optimization. AI models predict timing and power with less than 2% error using deep learning (CNN, GNN), and automatically execute Engineering Change Orders (ECOs) using RL algorithms. Guo et al. [9] proposed a differentiable optimization fusion technology that coordinates global placement and gate sizing, increasing timing convergence speed by 3× and reducing power consumption by 15%.

2.3 Analog/RF Design: Topology Generation, Device Sizing, and Layout

Analog/RF circuit design involves a nonlinear, multi-constrained, and parasitic-sensitive design space that traditionally requires 6-18 months of iterative optimization. A fundamental challenge distinguishing analog from digital design is the non-continuity of the parameter space: device width-to-length ratios, bias currents, and compensation capacitances are typically selected from discrete grids dictated by foundry design rules, creating a combinatorial optimization landscape with numerous local optima. Bayesian optimization addresses this by constructing surrogate models (e.g., Gaussian Processes) that capture the non-smooth response surface and using acquisition functions (e.g., Expected Improvement) to balance exploitation and exploration across discontinuous regions. Reinforcement learning offers an alternative by learning policies that navigate the discrete parameter space through trial-and-error interaction with simulation environments. Evolutionary algorithms further complement these approaches by maintaining diverse solution populations that can escape local optima through crossover and mutation operations on circuit topology graphs.

In topology generation, evolutionary algorithms and RL can automatically generate new analog circuit topologies, achieving 20%-40% performance improvement. For layout automation, GNNs automatically match symmetric structures and avoid parasitic effects, increasing tape-out success rates by over 30%. Zhang et al. [3] published the first comprehensive review of generative AI for analog design, focusing on GNNs, LLMs, and VAEs for topology generation, sizing optimization, and layout automation, and proposing solutions to data scarcity and generalization challenges.

2.4 Physical Verification and Yield Improvement

Under advanced processes, physical verification (DRC/LVS) processes billions of layout patterns. AI addresses these by leveraging deep learning for design rule error identification, RL-based automatic repair, and yield prediction models, shortening DRC/LVS repair time by 40%-60% and increasing chip yield by 5%-15%. Islam et al. [1] reviewed AI applications in physical verification and yield improvement, proposing solutions to address data scarcity and improve PVT robustness.

3. Engineering Practices of AI-EDA Transformation in Industry

The integration of AI into industrial EDA workflows represents a fundamental engineering challenge: how to bridge the gap between AI's optimization objectives and the physical constraints of IC design. A central technical question is how to jointly model reinforcement learning reward functions with physical design rules to ensure that AI-optimized layouts remain manufacturable. In practice, this requires embedding design rule constraints (minimum spacing, metal density, via coverage) as penalty terms or constrained action spaces within the RL framework, rather than treating them as post-hoc verification checks.

At advanced process nodes (5nm and below), leading EDA vendors have demonstrated that RL-driven tools can achieve significant PPA improvements when the reward function is carefully aligned with foundry-specific design rules. For example, when the RL agent's state representation incorporates timing derating factors from the foundry's sign-off library, and the reward function penalizes design rule violations proportionally to their manufacturing severity, the resulting placements show both improved congestion metrics and reduced DRC violation counts compared to traditional heuristic approaches. The key engineering insight is that the fidelity of the reward signal—how accurately it reflects post-route timing and DRC outcomes—directly determines the quality of AI-generated solutions, making reward engineering a critical bottleneck in industrial AI-EDA deployment.

4. Technical Challenges and Development Prospects

4.1 Key Challenges

AI-driven IC design faces several interconnected challenges that stem from fundamental algorithmic limitations. The issue of insufficient generalization arises because current AI models learn process-specific representations that do not transfer across technology nodes: a GNN trained on 7nm placement data encodes density and congestion patterns specific to that node's metal stack and design rules, and these learned features become largely irrelevant when applied to 3nm designs with fundamentally different routing architectures. Overcoming this requires developing few-shot and zero-shot learning techniques that can rapidly adapt pre-trained models to new process contexts with minimal labeled data.

Data barriers compound the generalization problem. Foundry Process Design Kits (PDKs) are highly confidential, leading to a scarcity of high-quality training data, especially for analog/RF designs, where simulation data is expensive to generate. This data scarcity not only limits model accuracy but also biases training distributions toward well-represented design styles. Poor interpretability further constrains deployment in safety-critical domains: most AI models used in IC design are black-box systems whose optimization decisions cannot be traced to specific circuit properties, making it difficult to satisfy compliance requirements of automotive (AEC-Q100) and medical (ISO 26262) chip certification. Accelerating Explainable AI (XAI) research is essential to address this gap. Additionally, physical consistency remains problematic—some AI models lack effective integration of physical constraints such as layout design rules and parasitic effects, leading to optimization results that may be non-manufacturable. Embedding physical circuit equations into AI models using Physics-Informed Neural Networks (PINNs) offers a promising approach to improving physical consistency.

4.2 Development Prospects

Rather than predicting specific timelines, the development of AI in IC design can be understood through the lens of algorithmic breakthroughs needed at each challenge point. For generalization, the key breakthrough lies in developing transfer learning mechanisms that decouple process-specific features from universal circuit structure representations—circuit foundation models that pre-train on diverse design datasets and fine-tune with minimal process-specific data. For interpretability, attention visualization and concept-based explanations could make AI optimization decisions auditable, enabling compliance with safety certification frameworks.

The integration of PINN-style constraint embedding represents another critical direction: rather than applying physical rules as post-processing checks, embedding differentiable physics constraints directly into the loss function ensures that AI-generated designs remain within the feasible manufacturing region throughout the optimization process. Multi-agent collaboration frameworks, in which specialized agents handle topology generation, device sizing, layout, and verification using shared constraint representations, offer a pathway toward end-to-end automation. Zhou et al. [10] proposed the concept of Circuit Foundation Models (CFM), outlining how such pre-trained models could serve as a unified backbone for diverse IC design tasks, representing a fundamental shift from task-specific to foundation-model-driven design automation.

5. Conclusion

AI has become a core driving force in full-process IC design, reconstructing the traditional paradigm from front-end to back-end and from digital to analog/RF design. This paper has emphasized the algorithmic representation layer underlying AI-IC integration—how GNNs encode circuit structure, how transformers capture long-range dependencies, and how RL reward functions align with physical design rules. While challenges in generalization, data scarcity, interpretability, and physical consistency remain, the algorithmic paths toward resolution are becoming clearer. The convergence of circuit foundation models, physics-informed learning, and multi-agent frameworks points toward a future where AI-driven IC design transitions from point-tool optimization to end-to-end intelligent automation.

References

[1] Islam G, et al. AI-Driven Integrated Circuit Design: A Survey of Techniques, Challenges, and Opportunities[J]. IEEE Access, 2025, 13: 58043-58059.

[2] Li J, et al. Artificial Intelligence in VLSI Physical Design: A Survey[J]. IEEE TCAD, 2024, 43(12): 3689-3708.

[3] Zhang Y, et al. Generative AI for Analog Integrated Circuit Design: Methodologies and Applications[J]. IEEE Access, 2025, 13: 45123-45140.

[4] Wang X, et al. ChipGPT: Large Language Model for End-to-End Chip Design[J]. IEEE TVLSI, 2025, 33(5): 987-1000.

[5] Zhuo Z, et al. Machine Learning in Advanced IC Design: A Methodological Survey[J]. IEEE/CAA JAS, 2024, 11(3): 589-610.

[6] Chen L, et al. Circuit Transformer: A Graph Transformer for Logic Synthesis Optimization[C]//DAC 2024. IEEE, 2024: 1-6.

[7] Liu H, et al. Reinforcement Learning-Based Routing for Advanced Node Designs[C]//ICCAD 2024. IEEE, 2024: 1-8.

[8] Sun W, et al. ChipMind: LLMs for Agile Chip Design[C]//VTS 2025. IEEE, 2025: 1-6.

[9] Guo Y, et al. Fusion of Global Placement and Gate Sizing with Differentiable Optimization[C]//ICCAD 2024. IEEE, 2024: 1-7.

[10] Zhou J, et al. A Survey of Circuit Foundation Model: Foundation AI Models for VLSI Circuit Design[C]//ICCAD 2025. IEEE, 2025: 1-10.

Funding

This research received no external funding.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgment

This paper is an output of the science project.

Open Access