# AI-Driven Back-End Design of Digital Integrated Circuits

Jiakang Li

College of science, Hubei University of Technology, Wuhan, Hubei, 430068, China
2210411202@hbut.edu.cn

**Abstract.** This paper investigates the critical applications of Artificial Intelligence (AI) in the back-end design of digital integrated circuits, with a specific focus on logic synthesis, placement and routing, and clock tree synthesis (CTS). As the integration density of digital circuits increases and process node advancements approach physical limitations, traditional back-end design faces challenges such as complex layout optimization, multi-dimensional power management, and difficulties in timing closure. Deep Reinforcement Learning (DRL) has seen rapid development in recent years, and AI technology has significantly improved design efficiency and quality through data-driven and intelligent automation methods. In placement and routing, DRL and Generative Adversarial Networks (GANs) have enabled rapid optimization, with Google's AI tools saving thousands of human hours in chip layout. In logic synthesis, reinforcement learning frameworks (such as A2C and ERL) and active learning strategies (such as Bulls-Eye) have improved circuit area and power optimization. In CTS, GAN-driven prediction models have reduced the number of clock buffers by 31%. AI technology promotes the intelligence and efficiency of digital chip back-end design through automation, global optimization, and efficient design space exploration, and it is expected to further integrate with EDA tools in the future.

**Keywords:** Logic Synthesis, Deep Reinforcement Learning, Placement, Clock Tree Synthesis

## 1 Introduction

According to the different processing signals, integrated circuit design is mainly divided into digital plate and analog plate. In the digital circuit design, it is divided into front-end and back-end. The front-end is mainly to achieve functional requirements, which is mainly divided into detailed RTL-level design and simulation verification, while the back-end process is more complicated. The main process is the translation of HDL code to gate level netlist to the final physical implementation, which includes logic synthesis, layout and routing, timing verification, etc. The back-end is a crucial part of the electronic design automation (EDA) process. It directly affects the chip's performance, power consumption, area (PPA) and other key indicators. With the progress of technology and process, the digital back-end is faced with great challenges,


J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_51

and its own design process is lengthy and the correlation between steps is strong, resulting in low iteration efficiency and slow design update. The most difficult is the complexity of the digital circuit back end, which is mainly reflected in the collaborative design challenge under the constraints of multiple physical dimensions. Under the nano-process node, the device functions increase, the design rules are complex, and the layout and routing optimization faces the problem of exponential growth. Designers need to balance the unit density, interconnect length and design rule constraints within the nano-level accuracy, while ensuring that the timing margin of the critical path, signal integrity and dynamic power density meet the chip reliability requirements. Power management has evolved from the traditional global static analysis to a dynamic multi-level control system, which requires the coupling relationship between power network impedance optimization, multi-threshold voltage unit hybrid layout and dynamic voltage frequency regulation (DVFS). With the application of FinFET and surround gate transistor (GAA) technology, the local leakage current difference can be more than 20% due to the fluctuation of process parameters, and local multidimensional power management becomes a problem. At the same time, timing convergence faces the challenge of non-ideal interconnection characteristics caused by deep submicron effects. Clock tree synthesis needs to optimize clock offset, insertion delay and power distribution synchronously, and adopt a hybrid architecture of multistage buffer insertion and clock grid to maintain timing robustness under multiple process angles. The existing physical realization tools are encountering the algorithm bottleneck of multi-objective optimization, and the traditional routing engine based on linear assumption is difficult to deal with the double graphical constraints and lithographic proximity effect in advanced processes. How to achieve global optimization becomes the key problem of digital back-end [1].

Traditional methods can trap us in locally optimal solutions when confronted with intricate designs. AI, particularly machine learning can analyze extensive design data, automatically learning the rules in the design space to find a better solution in a larger design space. Leveraging AI's data-driven characteristics allows for the customization of design automation algorithms, such as congestion assessment during the layout phase, for specific design flows and application scenarios. Furthermore, AI's intelligent support enables designers to swiftly adapt to evolving design requirements and technological challenges through automated intelligent tools. AI can provide real-time optimization suggestions to designers by training datasets, thereby enhancing design efficiency and accuracy. AI lowers the threshold for workers and reduces the demands on algorithms for workers [1]. AI technology is increasingly applied in the industry. Companies like Google and Synopsys have launched AI-EDA tools (such as DSO.ai and Circuit Training), verifying the effectiveness of AI in industrial scenarios. In chip placement, Google utilized AI technology to complete a task in less than six hours that would take human experts weeks in a loop [2]. Domestic and international EDA companies have integrated AI technology into EDA tools, such as X-EPIC's AmazeFP-ME, promoting the deep integration of AI and EDA.

This paper summarizes the recent applications of artificial intelligence technology in digital back-end design. The summary focuses on key steps including layout, routing, logic synthesis, and clock tree synthesis. The paper analyzes the trends of key cutting-edge artificial intelligence technologies. It also explores potential applications of these

technologies. This work provides a foundation for further in-depth research in the future.

## 2 AI in Digital Circuit Key Back-end Design

### 2.1 Placement and routing optimization

As Moore's Law slows down, the world is turning to specialized hardware development to meet the exponential growth in demand for computing. However, the problem is that the chip design cycle is lengthy, statistics show that the layout and wiring stage are the two most time-consuming steps in the EDA back-end design process, and the increasing complexity and scale, so that the layout has a huge challenge, the main problem is the layout and wiring congestion problem, which can seriously affect the performance, power consumption and area (PPA) indicators. also consider the design rules, otherwise it may directly lead to the inability to manufacture, increase the design cycle, so the use of ai technology to enable layout and wiring to shorten the cycle has become a huge direction in the future. Early decisions in the first layout phase (such as the placement of macro cells) have a critical impact on subsequent wiring quality and final chip performance. A poor early layout can lead to difficult final wiring or poor performance. They propose a chip layout planning method based on deep reinforcement learning. In less than 6 hours, the method can automatically generate the chip layout diagram, they use the chip layout planning as a reinforcement learning problem [2], they model the layout process as a Markov decision process (MDP), by confirming the layout area, knowing the design rules and size information of each unit, and then return to the layout strategy through reinforcement learning. By training the policy network with algorithms such as near-end Policy Optimization (PPO), predicting that location can earn the ultimate reward. This method has shown good performance in large-scale design, even beyond manual experience and traditional analysis. After the macro cell layout is complete, the initial layout of the standard cell is usually done using the force-directed method. This approach views the network list as a spring-loaded system, where tightly connected nodes attract each other and overlapping nodes repel each other [2].shi et al also propose a Wiremask-EA framework that uses the concept of wire mask to predict the potential increase in half-cycle line length (HPWL) after each macrocell placement. This guides black-box optimization algorithms (BBO) such as Evolutionary Algorithms (EA) to macro cell layout, demonstrating the versatility of this approach in handling complex optimization tasks in chip design [3].

In congestion prediction and optimization, the architecture of neural network is often used to predict congestion. Accurate congestion prediction can help identify potential congestion areas during the layout phase and guide the layout optimization algorithm to make adjustments, such as moving units out of the congestion area by adjusting the placement density. The artificial neural network (ANN) model is generally superior to the traditional machine learning model in congestion prediction. Beniwal et al. selected the key design attributes that affect cabling congestion by analyzing the correlation and variance inflation factors of various design features. The ANN model is trained using these features to significantly reduce the final routing congestion[4]. In order to reduce

the cost of layout and routing, advance is particularly important for the prediction and evaluation of deployable linearity. A novel variant of U-Net, the ibUNet model, is proposed by Hailiang Li et al. to more accurately predict cabling congestion and design rule checking (DRC) hot spots by inserting an Inception module at the bottleneck of U-Net to expand the global relationship between nodes in the sensing network list. Experiments on the CircuitNet dataset show that it outperforms traditional model methods in accuracy and efficiency [5]. The evaluation of these algorithms usually focuses on intermediate alternative indicators such as congestion, line length (WL), half-cycle line length (HPWL) and macro HPWL (mHPWL), which are easy to calculate. However, it often shows significant deviations from end-to-end performance (i.e. the final design PPA). To address this challenge, ChiPBench was introduced, which can effectively facilitate the study of chip layouts within the AI community. ChiPBench is a comprehensive benchmark specifically designed to evaluate the effectiveness of existing AI-based chip layout algorithms in improving final design PPA metrics. Specifically, 20 circuits from various fields (e.g., CPUs, Gpus, and microcontrollers) were collected. These designs are compiled by executing a workflow from the Verilog source code that retains the necessary physical implementation toolkit to be able to evaluate the impact of the layout algorithm on the final design PPA [3].

### 2.2 Logic synthesis

Logic synthesis is a key step in the chip design flow, aiming to translate high-level hardware descriptions into optimized gate-level circuits digital. However, traditional logic synthesis methods usually use a unified heuristic algorithm to optimize different circuits.LS relies on a series of optimization commands to perform the optimization, but the complexity of the synthesis optimization process grows exponentially as the number of commands used increases, and it is easy to fall into a local minimum, which is not able to maximize the quality of the results. Currently, the research frontiers are focusing on the innovative application of Artificial Intelligence (AI) technology in the field of logic synthesis, aiming to break through the performance boundary of traditional Electronic Design Automation (EDA) tools. The current research frontiers focus on the innovative application of artificial intelligence technology in the field of logic synthesis, which is committed to breaking through the traditional electronic design automation (performance boundaries of ) tools. The core research direction focuses on deeply integrating machine learning algorithms with circuit design principles. Researchers have constructed a timing-aware graph neural network model. This model systematically identifies redundant logic transformations. It accelerates the Boolean function optimization process. A key focus is improving the optimization efficiency of timing-critical paths. To achieve this, a gradient-enhanced heuristic search strategy has been developed. Addressing the challenge of design space generalization is critical. Researchers are building intelligent model systems with strong generalization capabilities. These systems utilize transfer learning to enhance the quality of results. To solve the challenge of design space generalization, the researchers are constructing an intelligent model system with strong generalization capability, adopting a migration learning framework and a meta-learning mechanism to enhance the model's ability to adapt to the unseen circuit topology, and to effectively deal with

the migration of process nodes and the EDA the challenges of out-of-domain generalization brought about by reuse of modules. In terms of optimization quality improvement, a multi-objective optimization framework based on deep reinforcement learning is introduced into IP the QoR prediction system, which by building a enables collaborative optimization of circuit performance metrics power-area-timing Pareto frontier model. More breakthrough progress has been achieved by combining active learning mechanisms and Monte Carlo tree search. This combination enables the construction of an intelligent synthesis engine with self-evolving strategy capabilities. The engine dynamically evaluates the potential gains of design space exploration trajectories. It also autonomously determines the optimal technology mapping sequences and process constraint combinations. These innovations significantly enhance the synthesis convergence efficiency for complex circuits.

In terms of power consumption optimization in logic synthesis, Yang Chenghao et al., in their research, transformed the logic synthesis problem into a deep reinforcement learning problem, used the Advantage Actor-Critic (A2C) algorithm to train agents capable of independently exploring logic synthesis optimization sequences, and composed a neural network architecture with a strategy network and a value network. 50 cycles were trained to minimize the dynamic power consumption of the circuit, and the power optimization results improved by 16.01% on average compared with the previous ABC algorithm [6]. In terms of reducing circuit area, DRL has been used to solve the problem of exploring large design Spaces, where the best sequences of applying different optimization functions can be found. Hosny et al proposed DRiLLS, which are logic synthesis methods for deep reinforcement learning based on A2C algorithm. Experimental reports on some open source benchmark circuits show that, The average area has been reduced by 13% [7]. In the latest study, A3C is an improvement of the Deep Q-Networks and A2C algorithms. Unlike previous deep reinforcement learning algorithms, in A3C, there are multiple agents each with its own network parameters and an on-board copy of the global neural network. These agents learn asynchronously by interacting independently with the environment, and then share the learned knowledge by updating the global network, reducing the area even more, for example, by 57.9% in the c432 circuit and 64.9% in the c1355 circuit [7].Chenyang Lv et al. proposed ERL-LS, a cloud-based framework using evolutionary reinforcement learning (ERL), which can significantly improve the efficiency of circuit design exploration without reducing the quality of solutions through parallel execution on multi-core processors and population control of agents. The average speed of finding the global optimal solution is 3.37 times faster than that of the original method, which greatly accelerates the generation of the original sequence and the logic comprehensive optimization process[8]. In terms of circuit optimization, Zhihai Wang has proposed PruneX framework, which is a data-driven logical synthesis operator paradigm aimed at reducing invalid circuit conversions and improving efficiency by combining learned classifiers. PruneX consists of two stages: offline, where the training data set is collected and the classification model is learned, and online, where the learned model is used to predict the effectiveness of the node. PruneX solves the OOD (out-of-distribution) generalization problem in logical synthesis operators. Experiments have shown that PruneX significantly improves the efficiency of Resub and Mfs2 operators, while maintaining comparable optimization performance in industrial and very large scale circuits. Up to 3.1 times faster[9]. The generation of sub-optimal synthetic

transformation sequences (" synthetic formulations ") is also an important problem in logic synthesis. Animesh Basak Chowdhury et al proposed the Bulls Eye framework, which is a method for fine-tuning pre-training models. The model, based on past synthesis data and fine-tuned with a small number of training samples from the new design, can accurately predict the quality of the synthesis recipe for previously unseen netlists. The method achieves a 2-30x improvement in run time and produces a synthetic formulation that achieves close to 95% QoR compared to traditional techniques, and by using active learning, Bulls Eye is able to select data points more intelligently, thereby reducing the cost of running on a limited integrated operating budget. To achieve higher performance [10]. Finally, some people have developed a new logic synthesis method inspired by deep learning model. Xihan Li et al adopted Circuit Transformer architecture to learn circuit structure by generating adversarial network (GAN) and improve the circuit by merging equivalent nodes. Design logic circuit. The authors address the challenges associated with strict logic constraints, achieve equivalent holding And Inverter chart (AIG) transformations via Circuit Transformer, and facilitate logic synthesis rewriting. This study uses techniques such as Monte Carlo tree search (MCTS) to enhance optimization strategies in logic synthesis. Experimental results show that this new rewriting method is effective in circuit optimization and opens the way for generative AI based logic synthesis methods. Future research directions focus on expanding the scale of circuit converters and improving the efficiency of models [11].

### 2.3 Clock tree synthesis

Clock Tree Synthesis (CTS) is a critical phase in the physical design of digital circuits. In essence, it is the routing of the clock. clock routing involves constructing an efficient and balanced clock distribution network to provide synchronous clock signals for each element of the design. This ensures timing integrity and minimizes clock skew. The physical realization of the clock signal distribution is often referred to as the clock tree. Given the critical role of the clock signal in digital chips, its distribution should be symmetrical, connecting each register unit from a common clock source with minimal delay differences. This has a significant impact on the total power consumption of the final chip. In particular, high-performance computing (HPC) designs face multiple challenges, primarily concerning timing convergence and power consumption control [12]. Traditionally, designers had to manually search through various candidate parameters to achieve design goals, a process that was both labor-intensive and time-consuming[13]. To automate this process and alleviate the burden on designers, current applications of artificial intelligence focus on two main directions: improving the prediction of Clock Tree Synthesis (CTS) metrics and directly optimizing CTS results. Melikyan et al. propose a novel method for selecting logic elements in CTS using parametric models and neural networks to enhance timing performance and reduce power consumption. This method has been validated on 14 nm and 32 nm technologies [14].

To optimize Clock Skew, minimize clock tree length, and reduce clock network power consumption, a common strategy is to place latches close to local clock buffers. Research on using decision tree (DT) models to reduce latch redundancy has led to an

optimized latch layout scheme, significantly reducing clock bias and positively impacting placement and power consumption [15].In the CTS elements prediction, A recent study developed an ML-driven clock tree construction algorithm that generates optimized clock trees based on a pre-trained CTS buffer prediction framework, estimating buffer usage. Building on this, a GAN-CTS framework was proposed, utilizing conditional generative adversarial networks (GANs) to predict and optimize CTS outcomes. This framework consists of three successive learning stages: transfer learning to extract netlist features directly from layout images, regression learning to predict target CTS outcomes, and multi-task learning, which achieved better accuracy than previous meta-modeling approaches. The framework can accurately predict key CTS metrics such as clock power consumption, clock line length, and maximum clock skew, providing input parameter recommendations. Regarding clock buffer arrangement, Koh et al. developed an artificial neural network (ANN) to estimate the number of buffers required for each clock gate and source in an ideal clock network. Experiments with a limited number of test circuits demonstrated high accuracy, with an average clock power estimation error of less than 5%. The proposed method also enables finding the minimum number of clock buffers possible with optimized clock parameters (e.g., target skew, clock conversion time). Using a binary search algorithm, the optimal clock parameters for a given number of buffers can be determined at each step. This optimization reduces the number of buffers in the clock network by an average of 31% [16]. Another study demonstrated that the clock tree engine serves as a helper model for business tools, showing that machine learning offers a promising solution for achieving desired CTS goals with minimal effort [13]. In summary, AI can construct a more balanced clock distribution network, ensure uniform clock signal propagation across the entire chip, reduce delay differences, improve system timing margins, and enhance the operating frequency of the chip.

## 3 Conclusion

In summary, AI through data-driven optimization, automated processes and global design exploration, significantly improve the efficiency and quality of the back-end design of digital integrated circuits, to solve the traditional method of local optimal and efficiency bottlenecks. However, the accuracy and generalization ability of the model still need to be improved. By increasing the scale and characteristics of the data set, the future research direction can jump out of the traditional design thought and algorithm research, and realize the application of ai in image processing, such as layout and wiring. With the deep integration of AI and EDA tools and the introduction of cross-domain technologies (such as image processing), the research direction can be improved. It will further promote the chip design to be intelligent and efficient, and open up a new path to deal with more complex process challenges and high-performance requirements.

## References


1. Lin, Y., Gao, X., Chen, T., Yu, B.: 'Machine learning-assisted backend design methods for digital integrated circuits', Micro/Nano Electronics and Intelligent Manufacturing, 2021, 3, (2), pp. 11–20
2. Mirhoseini, A., Goldie, A., Yazgan, M., et al.: 'A graph placement methodology for fast chip design', Nature, 2021, 594, (7862), pp. 207–212
3. Wang, Z., Geng, T., Wang, J., et al.: 'Benchmarking the end-to-end performance of chip placement algorithms', AI-based arXiv, 2024, abs/2407.15026
4. Beniwal, P., Saurabh, S.: 'Artificial Neural Network Based Prediction and Congestion Mitigation during Placement', 37th International Conference on Design and VLSI, 2024, pp. 300–305
5. Li, H., Huo, Y., Wang, Y., et al.: 'Lightweight initial boosting of for routability prediction U-Net neural networks', 2024 2nd International Symposium on Electronic Design Automation (ISEDA), 2024, pp. 648–653
6. Yang, C., Xia, Y.: 'Reinforcement Learning for Power Optimization in Logic Synthesis', 2021 IEEE 14th ASIC International Conference (ASICON), 2021, pp. 1–3
7. Pasandi, G., Prati, S., Forsyth, J.: 'aisyn: based a framework for AI-driven reinforcement learning logic synthesis', arXiv, 2023, abs/2302.06415
8. Lv, C., Zhang, B., Qian, W., He, Z.: 'ERL-LS: Accelerating Primitive Sequence Generation of Logic Synthesis with Evolutionary Reinforcement Learning', 2024 2nd International Symposium on Electronic Design Automation (ISEDA), 2024, pp. 672–677
9. Wang, Z., Chen, L., Wang, J., et al.: 'A Circuit Domain Generalization Framework for Efficient Logic Synthesis in Chip Design', arXiv, 2023, abs/2309.03208
10. Chowdhury, A.B., Tan, B., Carey, R., et al.: 'Bulls-Eye: Active Few-Shot Learning Guided Logic Synthesis', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2023, 42, pp. 2580–2590
11. Li, X., Li, X., Chen, L., et al.: 'Logic Synthesis with Generative Deep Neural Networks', arXiv, 2024, abs/2406.04699
12. Chou, Y.-C., Wu, C.-C., Tsai, C.-H.: 'Clocking for HPC Design: Challenges and Experience Sharing', 2019 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), 2019, pp. 1–2
13. Lu, Y.-C., Lee, J., Agnesina, A., et al.: 'A Clock Tree Prediction and Optimization Framework Using Generative Adversarial Learning', IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2022, 41, (9), pp. 3104–3117
14. Melikyan, V., Galstyan, A., Ghukasyan, S., et al.: 'Development of Parameterized Model of Logic Elements at Clock Tree Synthesis', Proceedings, 2024, 77, (1), pp. 58
15. Attaoui, Y., Chentouf, M., Ismaili, Z.E.A.A., El Mourabit, A.: 'Machine Learning in VLSI Design: A Comprehensive Review', Journal of Integrated Circuits and Systems, 2024, 19, (2), pp. 1–14
16. Koh, S., Kwon, Y., Shin, Y.: 'Pre-layout Clock Tree Estimation and Optimization Using Artificial Neural Network', Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design, 2020