Large Language Models for EDA: Future or Mirage?

Zhuolun He
The Chinese University of Hong Kong
Hong Kong SAR

Bei Yu
The Chinese University of Hong Kong
Hong Kong SAR

ABSTRACT

In this paper, we explore the burgeoning intersection of Large Language Models (LLMs) and Electronic Design Automation (EDA). We critically assess whether LLMs represent a transformative future for EDA or merely a fleeting mirage. By analyzing current advancements, challenges, and potential applications, we dissect how LLMs can revolutionize EDA processes like design, verification, and optimization. Furthermore, we contemplate the ethical implications and feasibility of integrating these models into EDA workflows. Ultimately, this paper aims to provide a comprehensive, evidence-based perspective on the role of LLMs in shaping the future of EDA.

ACM Reference Format:
Zhuolun He and Bei Yu. 2024. Large Language Models for EDA: Future or Mirage? . In *Proceedings of the 2024 International Symposium on Physical Design (ISPD '24), March 12–15, 2024, Taipei, Taiwan.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3626184.3639700

1 INTRODUCTION

The advent of Large Language Models (LLMs) has ushered in a new era in the realm of artificial intelligence, redefining the boundaries of machine learning and its applications. One such field that stands on the cusp of potential transformation is Electronic Design Automation (EDA). EDA, a cornerstone in the semiconductor industry, encompasses a suite of software tools for designing electronic systems such as integrated circuits and printed circuit boards. The integration of LLMs into EDA promises to revolutionize this field, offering unprecedented capabilities in design automation, error detection, and process optimization.

This paper seeks to explore the intersection of LLMs with EDA, critically examining whether this integration heralds a new future or is merely an overhyped concept. The surge in LLM capabilities, characterized by models such as GPT-4, has demonstrated profound potential in understanding and generating human-like text. This capability, when applied to EDA, could offer transformative solutions in automating complex design processes, providing natural language interfaces for design tools, and enhancing the accuracy of predictive models used in circuit design and analysis.

However, the marriage of LLMs with EDA is not without its challenges. The complexity of EDA tasks, coupled with the specialized nature of electronic design languages, poses a significant hurdle. Moreover, the integration of LLMs raises pertinent ethical and practical concerns. Issues such as data privacy, model reliability, and the potential for automation-induced obsolescence in skilled professions are critical considerations that must be addressed.

In this paper, we delve into the nuances of applying LLMs to EDA. We analyze current trends, potential applications, and the challenges faced in actualizing this integration. By examining case studies and emerging research, we aim to provide a comprehensive overview of how LLMs could potentially reshape the future of EDA, evaluating whether this integration is a groundbreaking advancement or an overestimated prospect.

2 APPLICATIONS FOR EDA

There are many applications applying LLM into circuit design and the EDA flow. For example, ChipNeMo [1] aims to optimize the utilization of LLMs in chip design by employing domain adaptation techniques instead of relying solely on off-the-shelf LLMs. These techniques include custom tokenizers, domain-adaptive continued pretraining, supervised fine-tuning with domain-specific instructions, and domain-adapted retrieval models. The effectiveness of these techniques is evaluated through three selected LLM applications: an engineering assistant chatbot, EDA script generation, and bug summarization and analysis. By training on 128 A100 GPUs, ChipNeMo has demonstrated impressive results in the selected applications. However, the evaluation results indicate a considerable disparity compared to human expert performance. To bridge this performance gap, several approaches are being considered by the authors, including data collection with more internal proprietary data, integrating better code-based base model, conducting reinforcement learning from human feedback (RLHF) [2], and investigating better RAG methods.

The most popular application is to help circuit design with LLM: prior works [3–7], among many others, are all very recent attempts towards this direction. VerilogEval [5] introduces a benchmarking framework specifically designed to evaluate the performance of LLMs in generating Verilog code for hardware design and verification. The evaluation dataset covers a wide range of Verilog code generation tasks, varying from simple combinational circuits to complex finite state machines. To assess the functional correctness of the generated Verilog code, transient simulation outputs are compared with a golden solution, enabling automated testing. Similarly, Thakur et al. [3] fine-tune pre-trained LLMs using Verilog datasets sourced from GitHub and Verilog textbooks. They develop an evaluation framework that includes test-benches for functional analysis and a flow to test the syntax of the generated Verilog code across various problem scenarios. GPT4AIGChip [6] is a framework that leverages LLMs and human natural languages to democratize AI accelerator design. The framework features an automated demo-augmented prompt-generation pipeline that utilizes in-context learning to guide LLMs in generating high-quality AI accelerator designs. The authors highlight the insights and guidelines derived from understanding the limitations and exploitable capabilities of LLMs for AI accelerator

ISPD '24, March 12–15, 2024, Taipei, Taiwan

ACM ISBN 979-8-4007-0417-8/24/03.
https://doi.org/10.1145/3626184.3639700

design automation, such as decoupled hardware template design, prioritizing in-context learning given limited data, and proper prompt engineering. ChipChat [4] authors argue that when human feedback is incorporated into the more advanced ChatGPT-4 model or used for co-design, the language model acts as a 'force multiplier', enabling quick exploration and iteration of the design space.

The second interesting direction is to help the design flow with LLM. ChatEDA [8] addresses the challenge of integrating EDA tools to improve interoperability in circuit design, by leveraging LLM capabilities in natural language processing and comprehension. An autonomous agent for EDA that utilizes the AutoMage LLM in conjunction with EDA tools as executors is proposed. It offers users a conversational interface to interact with these tools. Users can communicate their requirements using natural language prompts, and ChatEDA's goal is to generate executable programs (scripts) that fulfill the user's specific needs. Acting as the controller, ChatEDA orchestrates the collaboration between these tools. It starts by creating a task list based on user requirements and then generates scripts for each task. The conversational interaction may inspire next-generation EDA tool evolution.

The third direction is debugging in Verilog: RTLFixer [9] is an automated framework that addresses syntax-related errors in Verilog code using LLMs. It consists of an LLM for code generation, RAG for accessing expert guidance, and ReAct for task decomposition and planning. The framework starts by formulating an input prompt combining a benchmark problem with a template. RAG and ReAct are then used to revise the code and resolve errors. Persistent syntax errors are addressed with feedback from compiler error logs and retrieved human guidance. This iterative debugging process continues until all errors are resolved. RTLFixer achieves a high success rate of 98.5% in fixing compilation errors and significantly improves pass rates in VerilogEval benchmarks. Specifically, it enhances the pass@1 success rates by 32.3% in VerilogEval-Machine and 10.1% in VerilogEval-Human benchmarks.

3 CONCLUSION AND FUTURE DIRECTIONS

This paper has embarked on a comprehensive exploration of the integration of LLMs into EDA. Our analysis reveals a landscape brimming with potential, yet punctuated by significant challenges and uncertainties. The promise of LLMs in enhancing EDA processes is undeniable, offering novel approaches to design automation, error detection, and efficiency improvements. However, the path to fully realizing this potential is fraught with technical, ethical, and practical obstacles.

The future of LLMs in EDA hinges on several key factors. Firstly, the development of more specialized LLMs tailored to understand the intricacies of electronic design languages and processes is crucial. The current generation of LLMs, while advanced, is predominantly trained on general language data. For LLMs to be truly transformative in EDA, they must evolve to comprehend the specialized syntax and semantics of electronic design, a process that requires extensive domain-specific data and training. Data generation techniques like self-instruct [10] are of great significance. Careful selection or combination between RAG and fine-tuning [11] is also crucial in achieving optimal results.

Secondly, addressing ethical and practical concerns is paramount. Issues such as data security, privacy, and the ethical use of AI in professional settings need rigorous standards and regulatory frameworks. Recent studies have demonstrate the vulnerabilities of LLMs when faced with malicious user interactions, and prompt injection techniques are proposed [12]. As LLMs become more integrated into EDA, it is vital to ensure that these tools are used responsibly, with a clear understanding of their limitations and potential biases.

Looking ahead, the future directions of LLM integration into EDA are manifold. One promising avenue is the development of AI-assisted design tools that seamlessly integrate with existing EDA software, offering enhanced capabilities in design optimization and error correction. Another area of exploration is the use of LLMs in educational settings, where they can aid in training the next generation of electronic designers, providing interactive learning experiences and personalized feedback.

Furthermore, there is substantial scope for collaborative research between AI experts and EDA professionals. Such collaboration can accelerate the development of customized LLMs that cater specifically to the needs of the EDA industry. Additionally, exploring the potential of LLMs in related fields like chip layout optimization and predictive maintenance of electronic systems could open new frontiers in the application of AI in electronics.

In conclusion, while the integration of LLMs into EDA presents a landscape rich with opportunities, it remains a complex and evolving field. Continued research, interdisciplinary collaboration, and mindful consideration of ethical implications are essential in steering this integration towards a future that is not only technologically advanced but also responsible and inclusive. As we stand at the cusp of this technological evolution, it is imperative to navigate this journey with a balanced perspective, embracing innovation while remaining cognizant of the challenges that lie ahead.

REFERENCES

[1] M. Liu, T.-D. Ene, R. Kirby, C. Cheng, N. Pinckney, R. Liang, J. Alben, H. Anand, S. Banerjee, I. Bayraktaroglu *et al.*, "ChipNeMo: Domain-Adapted LLMs for Chip Design," *arXiv preprint arXiv:2311.00176*, 2023.

[2] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Proc. NeurIPS*, vol. 35, pp. 27 730–27 744, 2022.

[3] S. Thakur, B. Ahmad, Z. Fan, H. Pearce, B. Tan, R. Karri, B. Dolan-Gavitt, and S. Garg, "Benchmarking Large Language Models for Automated Verilog RTL Code Generation," in *Proc. DATE*, 2023, pp. 1–6.

[4] J. Blocklove, S. Garg, R. Karri, and H. Pearce, "Chip-Chat: Challenges and Opportunities in Conversational Hardware Design," in *Proc. MLCAD*, 2023.

[5] M. Liu, N. Pinckney, B. Khailany, and H. Ren, "VerilogEval: Evaluating Large Language Models for Verilog Code Generation," in *Proc. ICCAD*, 2023.

[6] Y. Fu, Y. Zhang, Z. Yu, S. Li, Z. Ye, C. Li, C. Wan, and Y. C. Lin, "GPT4AIGChip: Towards next-generation AI accelerator design automation via large language models," in *Proc. ICCAD*, 2023.

[7] K. Chang, Y. Wang, H. Ren, M. Wang, S. Liang, Y. Han, H. Li, and X. Li, "ChipGPT: How far are we from natural language hardware design," *arXiv preprint arXiv:2305.14019*, 2023.

[8] Z. He, H. Wu, X. Zhang, X. Yao, S. Zheng, H. Zheng, and B. Yu, "ChatEDA: A Large Language Model Powered Autonomous Agent for EDA," in *Proc. MLCAD*, 2023.

[9] Y. Tsai, M. Liu, and H. Ren, "RTLFixer: Automatically Fixing RTL Syntax Errors with Large Language Models," *arXiv preprint arXiv:2311.16543*, 2023.

[10] Y. Wang, Y. Kordi, S. Mishra, A. Liu, N. A. Smith, D. Khashabi, and H. Hajishirzi, "Self-instruct: Aligning language model with self generated instructions," *arXiv preprint arXiv:2212.10560*, 2022.

[11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, and H. Wang, "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

[12] F. Perez and I. Ribeiro, "Ignore previous prompt: Attack techniques for language models," *arXiv preprint arXiv:2211.09527*, 2022.