Qeios

Peer Review

Review of: "DRC-Coder: Automated DRC Checker Code Generation Using LLM Autonomous Agent"

Bei Yu[1]

1. The Chinese University of Hong Kong, Hong Kong, Hong Kong

The article presents a significant advancement in the field of electronic design automation (EDA) by introducing DRC-Coder, a multi-agent framework designed to automate the generation of Design Rule Check (DRC) codes. The abstract highlights the challenges faced in the current process of implementing integrated DRC checkers, particularly the labor-intensive nature of interpreting foundry specifications, analyzing layouts, and debugging code, which is exacerbated by frequent updates in technology nodes.

Key Contributions:

Automation of DRC Code Generation: The authors propose DRC-Coder, which leverages vision language models (VLMs) and large language models (LLMs) to interpret design rules and generate DRC codes automatically. This is a notable improvement over traditional methods that require extensive human intervention.

Multi-Agent Framework: The framework employs two specialized LLMs to handle different aspects of the DRC code generation process—rule interpretation and coding. This division of labor between agents is a clever approach to managing the complexity of the task.

Auto-Evaluation Function: The inclusion of an auto-evaluation function for debugging DRC codes is a critical feature that enhances the robustness of the system. This function allows for iterative refinement of the generated codes, ensuring they meet the stringent standards of commercial DRC tools.

Performance Metrics: The experimental results are impressive, with DRC-Coder achieving a perfect F1 score of 1.000 in generating DRC codes for a sub-3nm technology node. This performance

 doi.org/10.32388/1MMGBB

significantly outperforms standard prompting techniques, which achieved an F1 score of 0.631. Additionally, the system can generate code for each design rule in an average of four minutes, which is a substantial reduction in turnaround time compared to manual methods.

Strengths:

Efficiency: The framework drastically reduces the time required to generate DRC codes, which is crucial for keeping pace with the rapid advancement of technology nodes.

Accuracy: The perfect F1 score indicates that the system is highly reliable in producing accurate DRC codes that meet commercial standards.

Cost-Effectiveness: By automating a process that traditionally requires significant human expertise, DRC-Coder has the potential to reduce engineering costs substantially.

Potential Limitations:

Generalizability: While the results are impressive for a sub-3nm technology node, it would be beneficial to see how well DRC-Coder performs across a broader range of technology nodes and design rules.

Complexity of Integration: The abstract does not discuss the ease of integrating DRC-Coder into existing EDA workflows. The practical implementation of such a system in a real-world design environment could pose challenges.

Dependence on LLMs: The framework's reliance on LLMs and VLMs might raise concerns about the interpretability and transparency of the generated codes, especially in critical applications where errors could have significant consequences.

Overall, the article presents a groundbreaking approach to automating DRC code generation, which could have a transformative impact on the EDA industry. The combination of vision and language models, along with the auto-evaluation function, makes DRC-Coder a powerful tool for accelerating technology advancement and reducing engineering costs. However, further research and real-world testing would be necessary to fully understand the system's limitations and potential areas for

improvement. This work is a promising step forward in the quest for more efficient and automated design processes in the semiconductor industry.

Declarations

Potential competing interests: No potential competing interests to declare.