EEschematic: Multimodal-LLM Based AI Agent for Schematic Generation of Analog Circuit

Chang Liu
School of Engineering
The University of Edinburgh
Edinburgh, UK
C.Liu-134@sms.ed.ac.uk

Danial Chitnis
School of Engineering
The University of Edinburgh
Edinburgh, UK
d.chitnis@ed.ac.uk

Abstract—Circuit schematics play a crucial role in analog integrated circuit design, serving as the primary medium for human understanding and verification of circuit functionality. While recent large language model (LLM)-based approaches have shown promise in circuit topology generation and device sizing, most rely solely on textual representations such as SPICE netlists, which lack visual interpretability for circuit designers.

To address this limitation, we propose EEschematic, an AI agent for automatic analog schematic generation based on a Multimodal Large Language Model (MLLM). EEschematic integrates textual, visual, and symbolic modalities to translate SPICE netlists into schematic diagrams represented in a human-editable format. The framework uses six analog substructure examples for few-shot placement and a Visual Chain-of-Thought (VCoT) strategy to iteratively refine placement and wiring, enhancing schematic clarity and symmetry. Experimental results on representative analog circuits, including a CMOS inverter, a five-transistor operational transconductance amplifier (5T-OTA), and a telescopic cascode amplifier, demonstrate that EEschematic produces schematics with high visual quality and structural correctness.

Index Terms—Schematic, Analog, LLM, automation

I. INTRODUCTION

Machine learning based methods have significantly accelerated research in analog circuit design, particularly in topology generation and sizing. For example, research [1] develop an analog Predefined Building Block Library (PBBL) for analog sub-circuits and leverages Deep Reinforcement Learning (DRL) to generate building block sequences and decode as Python code to select topologies. AnalogCoder uses LLMs to generate Python code for PySpice, developing a feedback loop to enrich the circuit tool library to improve LLMs' ability in analog design [2]. AmpAgent [3] employs a Retrieval-Augmented Generation (RAG) method, manually converting the circuit images from research papers to SPICE netlists to enhance LLM's knowledge of analog topology. Most approaches represent circuits as SPICE netlists and Python code, describing connectivity in a textual format. However, analog circuit designers typically rely on schematic diagrams to understand signal paths and circuit topologies, rather than textual descriptions. Consequently, even when ML models generate optimal circuit netlists, they are often complex for

designers to interpret and verify, and converting them into schematics is time-consuming and prone to mistakes [4]. Therefore, an automatic approach for converting analog circuit netlists to visual schematics is desired.

Research on automated netlist-to-schematic generation is mainly focused on digital circuits. Existing synthesis and implementation tools such as Vivado [5] and Synopsys Design Compiler [6] support automated design and visualization. However, these gate-level approaches are not well-suited for transistor-level analog schematic generation, as analog schematics emphasise factors including symmetry and visual clarity, rather than sequential gate connections. Few studies have explored analog schematic generation. Early works used symmetry-based placement [7] and net-crossing minimization [8], but these methods failed to generalize across circuit types. RL-based approaches generate schematics using building block classification, successfully handling a few analog circuits, yet still rely on comprehensive libraries of subcircuits [9]. Schemato used domain-specific LLMs to convert netlists to schematics based solely on textual information, achieving a 76% compilation success rate on their dataset. However, it struggles to generate schematics for circuits with more than five components with accurate connectivity [4].

Recently, Multi-modal Large Language Models (MLLMs) such as Gemini 2, GPT 4, and Claude 4 have demonstrated outstanding abilities in integrating textual and visual understanding and reasoning. MLLMs allow flexible modelling of relationships across modalities, which suggests they have the potential to generate schematics by integrating multimodal information, such as textual descriptions, circuit diagrams, and symbolic netlists, into coherent circuit representations.

In this work, we propose EEschematic, a multimodal AI agent for automatic analog schematic generation. We reduce reliance on schematic libraries by introducing a few basic analog substrcuture examples for a few-shot guidance during the initial placement stage. Additionally, we integrate a Visual Chain-of-Thought (VCoT) prompting for placement and wiring, enabling iterative visual reasoning over schematic images and JSON representations. Experimental results on different analog circuits, including the 5T-OTA and the telescopic cascode amplifier, demonstrate that EEschematic can generate visually clear and structurally correct schematics.

Natural language

A CMOS inverter which have one PMOS and NMOS with Drain connected

Spice netlist

M1 out in vdd vdd pmos W=1u L=0.09u
M2 out in 0 0 nmos W=1u L=0.09u
Vin in 0 DC 0.9
vdd vdd 0 1.8

Schematic

JSON

```
{
  "componentInstances": [
  {
    "typeName": "pFET",
    "name": "M1",
    "value": "pmos W=1u L=0.09u",
    "origin": {
      "x": 0,
      "y": 20
    },
    "rotation": "0",
    "flip": "none"
  },
...]
}
"wires": [
  {
    "startLocation": {...},
    "relativePaths": [...],
    "endLocation": {...},
...]
}
```
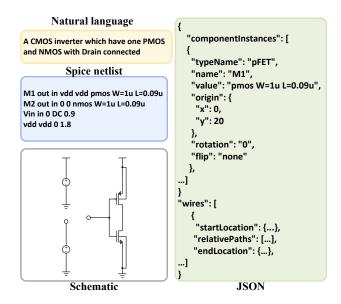
Fig. 1: Different modalities of circuit representation, including natural language description, SPICE netlist, circuit schematic diagram, and corresponding position and wiring information within a JSON schema.

TABLE I: Sub-circuit examples predefined as LLM's context for schematic generation. The JSON file is represented in images, which are generated from the schematic renderer to show the spatial structure of the circuits.

	Single Cascode	Single Current Source
JSON schema represented schematic		
Natural Language Description	Single transistor. The source and drain are connected...	Single transistor. The source should be connected to...
	Diode Connected	**Two-Transistor Cascode**
JSON schema represented schematic		
Natural Language Description	Single transistor with gate and drain shorted together...	Two transistors with gates connected together. The left one ...
	Differential Pair	**Current Mirror**
JSON schema represented schematic		
Natural Language Description	Three transistors with two sources connected to ...	Two PMOS transistors with gates connected together....

II. METHODOLOGY

A. Circuit Representations

In this work, circuit schematic generation is formulated as a multimodal task that involves constructing a natural language description, an SPICE netlist, a circuit schematic diagram, and corresponding position and wiring information within a JSON file for the schematic renderer, as shown in Fig. 1.

The SPICE netlist encodes the electrical connectivity and component parameters of the circuit in a machine-readable format. The natural language description provides a human-understandable explanation of circuit functionality, summarizing the roles of key components and their interactions. The schematic diagram offers a visual representation of the circuit topology, enabling intuitive analysis and refinement. The position and wiring data specify the spatial information and interconnections, ensuring consistent schematic rendering. Together, these modalities establish a shared context that enables MLLMs to associate textual, symbolic, and geometric information for more accurate schematic generation.

B. Sub-circuit Examples

LLMs' ability to reason from few-shot examples makes it possible to learn the underlying principles of analog structure from a limited set of examples and generalize this understanding to generate correct placement, demonstrating that it is not necessary to maintain a comprehensive example set containing every possible structure variant, such as both PMOS and NMOS-based implementations. In this work, we manually constructed a reduced analog substructure example set based on the building block library [1] to store instructions and examples for the placement process.

This set comprises three key collections: First, substructure placement descriptions, which serve as placement guidelines. Second, the corresponding SPICE netlists form the textual foundation for both structural recognition and simulation-driven reasoning by the LLM. Third, a JSON schema encodes positional information, which provides examples for reference. An overview of all the examples is presented in Table I.

C. Initial Placement and Wiring

The initial placement aims at establishing a foundational structure that ensures proper symmetry, flipping, and rotation of components. The LLM-guided process begins by analyzing the SPICE netlist to identify circuit substructures, with examples provided as context for the LLM. The LLM first analyzes the netlist to identify substructures, then determines their spatial relationships and orientations to generate an initial schematic placement that serves as the basis for subsequent refinement. The initial wiring between components is generated automatically by a custom-defined wiring algorithm, which reconstructs circuit connectivity from the SPICE netlist and node location data, automatically identifying and mapping electrical connections between device terminals. Terminals sharing the same net name are considered connected and grouped accordingly, forming the foundation for a hierarchical connection graph. The connection priority is assigned in the following order by their connection complexity: VDD, GND, Gate, Drain, Source, Bulk, and finally I/O, resistor, capacitor, and other pins. In the end, A post-processing step removes
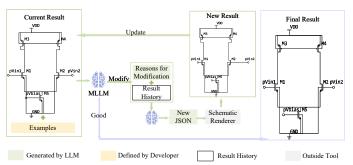
Fig. 2: The inner optimization loop for placement. The MLLM compares the current schematic with reference examples to determine whether modification is needed. If required, the model generates reasoning chains to guide schematic refinement. The updated schematic and its corresponding JSON data are then iteratively fed back into the inner optimization loop, enabling continuous improvement through multimodal reasoning.

conflicting connections, yielding start–end coordinate pairs that serve as the foundation for schematic visualization and further geometric refinement.

D. Optimization

In this work, the optimization of the initially generated schematic is divided into two stages: placement and wiring optimization. During these stages, Visual Chain-of-Thought (VCoT) prompting [10] is employed to construct an inner reasoning loop that guides the MLLM through multimodal information, thereby enhancing both the clarity and aesthetic quality of the schematic.

To implement VCoT in the placement optimization, a two-step interaction is established between EEschematic and the MLLM API. In the first step, the MLLM receives the current schematic in image format and determines whether modifications are necessary by comparing it against reference examples in its context. These reference examples include both good and bad examples: a poor placement case, illustrating misplaced devices and disorganized wiring, and a preferred placement case, demonstrating the desired spatial organization and clean connectivity pattern.

The MLLM evaluates the similarity between the current schematic and the reference examples to decide whether modifications are required. If the schematic is identified as suboptimal by the LLM, the VCoT prompt then guides the MLLM to generate reasoning chains that explain the necessary modifications. The model subsequently uses the result history, integrating schematic images and corresponding JSON data, as in-context information to produce an improved placement. This iterative reasoning process enables in-context learning, allowing the MLLM to better understand the relationship between spatial arrangement, wiring complexity, and schematic readability. Each newly generated result is then passed to the schematic renderer, and the outcomes are sent back for modification. The previous result is incorporated into the result

TABLE II: Correctness and aesthetics for different circuits. In this experiment, a maximum of 10 iterations is assigned to placement and 20 for wiring optimization, with 10 trials performed per circuit.

Circuit	Correctness	Aesthetics	Avg. Iter. for Placement.	Avg. Iter. for Wiring.
Inverter	9/10	9/10	3	3
5T-OTA	9/10	8/10	3	4
Telescope cascode	9/10	5/10	4	6

history, forming a self-improving inner optimization loop. The overall process of the proposed VCoT-based placement optimization is illustrated in Fig. 2. After placement optimization, the modified schematic is passed to the next stage for detailed wiring optimization, which follows a similar VCoT-guided methodology.

III. RESULTS

We conducted experiments on three representative analog circuits: an inverter, a 5T-OTA, and a telescopic cascode amplifier, using Gemini 2.0 Flash as the evaluation MLLM. To evaluate the performance of EEschematic, we introduced visual correctness and aesthetics as the key evaluation metrics. Correctness represents the structural validity of the schematic by verifying device connectivity and ensuring no overlaps occur in components and wiring. In contrast, aesthetics are evaluated subjectively by manually assessing visual quality for symmetry, alignment, wire compactness, and overall clarity.

The results are summarized in Table II. Simpler circuits, such as the inverter, achieve both high correctness and strong aesthetics with minimal iterations. In contrast, the telescopic cascode amplifier, with denser connectivity and more complex spatial relationships, maintains high correctness but exhibits lower aesthetic scores, reflecting a more challenging visually optimized schematics within limited iterations.

Fig. 3 illustrates the schematic generation process for the 5T-OTA. Starting from the netlist, the model performs initial placement and then refines alignment and connectivity through three placement and three wiring iterations, ultimately producing a coherent schematic. This demonstrates the model's multimodal reasoning capability to iteratively optimize based on JSON schema and visual information given.

Fig. 4 compares the best, worst, an example of a correct and aesthetic result and an example of a correct but not aesthetic result of the telescopic cascode amplifier. The best case shows strong symmetry and compact wiring, while the worst case contains incorrect and asymmetric connections. This highlights both the potential and variability of the LLM-based placement optimization process.

IV. DISCUSSION

This work found that the proposed EEschematic, a MLLM-based AI Agent, can effectively generate circuit schematics from input netlists for different circuit types. However, some limitations remain. Firstly, this work focused primarily on fundamental analog circuits. While these serve as representative
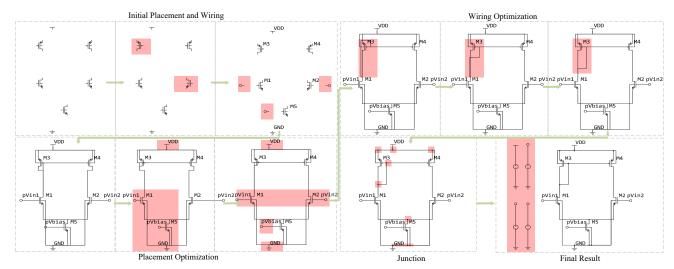
Fig. 3: The complete process of schematic generation for the 5T-OTA from initial placement to the final result. Changes from previous iterations are highlighted in red blocks.

(a) Best result

(b) Worst result

(c) Aesthetic result
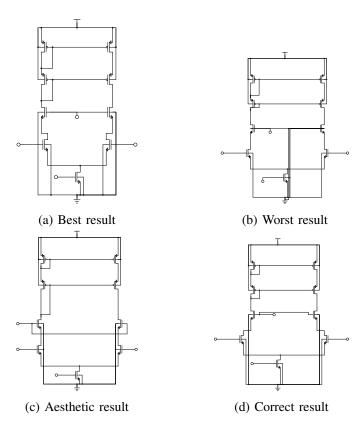
(d) Correct result

Fig. 4: Visualization of the JSON-based placement results of a telescopic cascode amplifier over 10 trials: (a) the best result, (b) the worst result, (c) an example of a correct and aesthetic result. (d) an example of a correct but not aesthetic result.

test cases for validating schematic reasoning and spatial organization, they only serve as a single block in more advanced circuit topologies, such as multi-stage amplifiers and mixed-signal systems. Future studies should extend the ability of these complex circuits to further assess the scalability and generalization of the proposed approach. Secondly, the subjective evaluation of the final schematic aesthetics introduces a degree of human bias. Future work could incorporate quantitative visual metrics or expert-validated benchmarks to ensure a more objective assessment. Thirdly, despite increasing the number of optimization iterations, a few generated schematics still exhibit structural or connectivity errors. Incorporating constraint-guided refinement strategies could help improve reliability and ensure convergence toward fully correct schematic representations.

V. CONCLUSION

We have developed EEschematic, an MLLM-based AI agent that automates the transformation from netlist to schematic. Using a few substructure examples for in-context learning and Visual Chain-of-Thought (VCoT) prompts to link textual and visual reasoning, the agent learns spatial organization and connectivity patterns, producing coherent and interpretable schematics. The proposed agent successfully generated schematics for three basic circuits: an inverter, a 5T-OTA, and a telescopic cascode amplifier, achieving a 90% correctness rate within an average of 10 iterations. The generated results exhibit diverse yet functionally valid schematics, showing that the model performs genuine reasoning rather than relying on fixed templates. Overall, this approach demonstrates strong potential for scaling schematic generation and optimization to more complex analog and mixed-signal systems. The source code is available on https://github.com/eelab-dev.

REFERENCES

[1] Z. Zhao and L. Zhang, "Analog integrated circuit topology synthesis with deep reinforcement learning," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 41, no. 12, pp. 5138–5151, 2022.

[2] Y. Lai, S. Lee, G. Chen, S. Poddar, M. Hu, D. Z. Pan, and P. Luo, "Analogcoder: Analog circuit design via training-free code generation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 1, 2025, pp. 379–387.

[3] C. Liu, W. Chen, A. Peng *et al.*, "Ampagent: An llm-based multi-agent system for multi-stage amplifier schematic design from literature for process and performance porting," *arXiv preprint arXiv:2409.14739*, 2024.

[4] R. Matsuo, S. Uhlich, A. Venkitaraman, A. Bonetti, C.-Y. Hsieh, A. Momeni, L. Mauch, A. Capone, E. Ohbuchi, and L. Servadei, "Schemato–an llm for netlist-to-schematic conversion," *arXiv preprint arXiv:2411.13899*, 2024.

[5] T. Feist, "Vivado design suite," *White Paper*, vol. 5, no. 30, p. 24, 2012.

[6] S. D. Compiler, "Synopsys design compiler," *Pages/default. aspx*, 2016.

[7] B. G. Arsintescu, "A method for analog circuits visualization," in *Proceedings International Conference on Computer Design. VLSI in Computers and Processors*. IEEE, 1996, pp. 454–459.

[8] Lee and McNamee, "Aesthetic routing for transistor schematics," in *1992 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 1992, pp. 35–38.

[9] H.-Y. Hsu and M. P.-H. Lin, "Automatic analog schematic diagram generation based on building block classification and reinforcement learning," in *Proceedings of the 2022 ACM/IEEE Workshop on Machine Learning for CAD*, 2022, pp. 43–48.

[10] H. Shao, S. Qian, H. Xiao, G. Song, Z. Zong, L. Wang, Y. Liu, and H. Li, "Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 8612–8642, 2024.