# ChatLS: Multimodal Retrieval-Augmented Generation and Chain-of-Thought for Logic Synthesis Script Customization

Haisheng Zheng ♠ Haoyuan Wu ♡ Zhuolun He ♡♣

♠ Shanghai AI Lab ♡ CUHK ♣ ChatEDA Tech

上海人工智能实验室
Shanghai Artificial Intelligence Laboratory

## Highlights

- We propose **ChatLS**, which precisely customize synthesis scripts.
- We propose a graph-based method that enhances the ability of LLMs to analyze and understand circuit designs by transforming them into graph databases, with GNN supporting the analysis process.
- We introduce a domain-specific multimodal RAG framework that efficiently retrieves information to enhance LLMs for customizing logic synthesis scripts.
- We present a task-specific CoT mechanism, in tandem with the multimodal RAG framework, that enables iterative refinement of synthesis scripts by revising reasoning steps based on retrieved data.
- Our method has achieved superior performance in customizing synthesis scripts with a commercial logic synthesis tool.

## Challenges

- **Comprehensive Analysis of Design**
  ✓ Recent research (e.g., RTLRewriter [1]) demonstrates the capability of LLMs to understand RTL code snippets, potentially aiding in analyzing circuit design for script customization.
  ✗ LLMs have limitations with long-context inputs, affecting their effectiveness in analyzing large codebases [2] and executing lengthy reasoning sequences [3].
- **Selecting Suitable Synthesis Commands**
  - Logic synthesis tools (e.g., Design Compiler) offer a range of optional commands to improve design quality, which should be selected based on user preferences and the analysis of the design.

## Problem Formulation

**Problem** (Objective of ChatLS). *Design methodologies to enhance the capability of LLMs in customizing logic synthesis scripts to meet user requirements.*

## References

[1] Xufeng Yao, Yiwen Wang, Xing Li, Yingzhao Lian, Ran Chen, Lei Chen, Mingxuan Yuan, Hong Xu, and Bei Yu. RTLRewriter: Methodologies for Large Models aided RTL Code Optimization. In *Proc. ICCAD*, 2024.

[2] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can Language Models Resolve Real-world Github Issues? In *arXiv preprint*, 2024.

[3] Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the Middle: How Language Models Use Long Contexts. In *Transactions of the Association for Computational Linguistics*, 2024.

[4] Haoyuan Wu, Haisheng Zheng, Yuan Pu, and Bei Yu. Circuit Representation Learning with Masked Gate Modeling and Verilog-AIG Alignment. In *Proc. ICLR*, 2025.

[5] Justin J Miller. Graph Database Applications and Concepts with Neo4j. In *Proc. SAIS*, 2013.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive Representation Learning On Large Graphs. In *Proc. NIPS*, 2017.

[7] Nadime Francis, Alastair Green, Paolo Guagliardo, Leonid Libkin, Tobias Lindaaker, Victor Marsault, Stefan Plantikow, Mats Rydberg, Petra Selmer, and Andrés Taylor. Cypher: An Evolving Query Language for Property Graphs. In *Proc. SIGMOD*, 2018.

[8] Josh Achiam et al. GPT-4 Technical Report. *arXiv preprint*, 2023.

[9] Patrick Lewis et al. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Proc. NIPS*, 2020.

[10] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, H Chi, Quoc V Le, and Denny Zhou. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Proc. NIPS*, 2022.

[11] T Ajayi, D Blaauw, TB Chan, CK Cheng, VA Chhabria, DK Choo, M Coltella, S Dobre, R Dreslinski, M Fogaça, et al. OpenROAD: Toward a Self-Driving, Open-Source Digital Layout Implementation Tool Chain. *Proc. GOMACTECH*, 2019.

[12] Anthropic PBC. Introducing Claude 3.5 Sonnet, 2024.

62 THE CHIPS TO SYSTEMS CONFERENCE

## Methodology

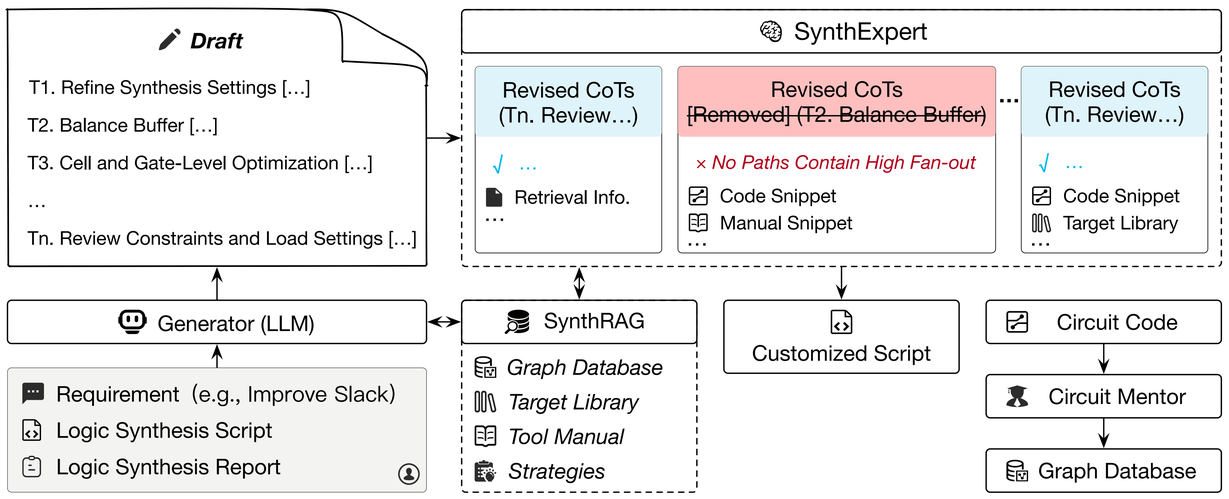


Figure 1. The Overall Flow of ChatLS.

- **CircuitMentor** is a circuit design analysis tool that transforms the circuit design into a graph representation and employs graph neural network (GNN) to extract high-level features from the circuit design.
- **SynthRAG** is a multi-modal RAG framework used to retrieve relevant information for LLMs when customizing logic synthesis scripts.

**ChatLS** begins with the user providing the requirements, circuit design code, associated logic synthesis scripts, and reports generated by the logic synthesis tool. Based on this input, the following steps are performed to customize logic synthesis script for the design:

- **Generator (LLM)** drafts a customized logic synthesis script based on user requirements, integrating synthesis tool reports, relevant data from **SynthRAG**, and high-level information matched using embeddings from **CircuitMentor**.
- **SynthExpert** is a Chain-of-Thought (CoT) reasoning mechanism that uses **SynthRAG** to fetch step-relevant information and iteratively evaluates the suitability of the drafted logic synthesis script for the circuit design.

## Circuit Mentor

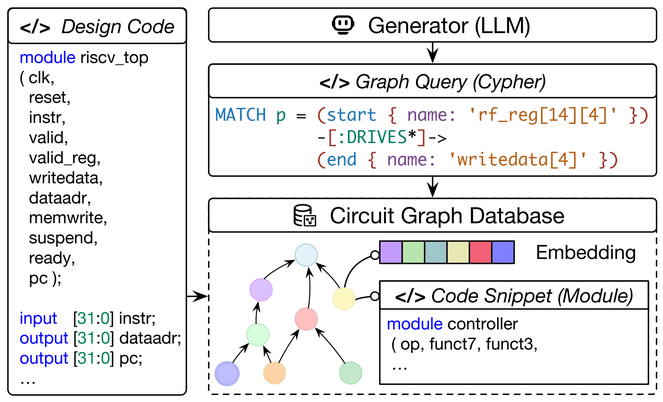


Figure 2. Workflow Visualization of CircuitMentor.

- **Motivation**
  - LLMs encounter significant challenges in comprehending complex circuit code.
  - Numerous studies (e.g., MGVGA [4]) to leverage GNNs to extract meaningful characteristics from complex circuits.
- **Graph Representation**
  - Transforming the circuit code into a graph and stored it in the Neo4j database [5].
- **Global Circuit Feature Extraction**
  - Hierarchical GNN based on GraphSAGE [6].
  - *Metric Learning* is employed to let similar designs be pulled closer together, while dissimilar ones be pushed farther apart.
- **Local Circuit Feature Extraction**
  - LLMs could generate query requests using Cypher to extract code snippets for analysis.

## SynthRAG

| Category | Representation | Query Method | Retrieval Content |
|---|---|---|---|
| High Level Info. of Circuit Design | Graph Embedding | Join | Compile Strategy Optimization Strategy |
| Code of Circuit Design | Graph Structure | Direct | The code of the module where the path is located |
| Target Library | Graph Structure | Direct | Gate Cell Information |
| Logic Synthesis Tool User Manual | LLM Embedding | Direct | Command Usage Command Requirement |

Table 1. Summary of Query Methods.

- **Graph Embedding-Based Retrieval**
  - *Circuit Embeddings Similarity-Based Retrieval*

$$\boldsymbol{z}_{Retrieved} = \arg\min_{\boldsymbol{z}_{db}} \left(\text{sim}(\boldsymbol{z}_{query}, \boldsymbol{z}_{db})\right) \quad (1)$$

  - *Circuit Performance-Based Reranking*

$$\text{Score}(\boldsymbol{z}_i) = \alpha \cdot \text{sim}(\boldsymbol{z}_{query}, \boldsymbol{z}_i) + \beta \cdot c_i \quad (2)$$

- **Graph Structure-Based Retrieval**:
  - Cypher queries [7] are used to retrieve information from circuit graphs.
- **LLM Embedding-Based Retrieval**:
  - The *text-embedding-3-large* model is used to convert the user manual into text embeddings.
  - *GPT-4o* [8] is employed as a reranker to improve retrieval accuracy.

## SynthExpert

- **Motivation**: LLMs can be effectively assisted in complex decision-making tasks with Retrieval-Augmented Generation (RAG) [9] and Chain-of-Thought (CoT) [10] reasoning.
- **SynthExpert** employs **SynthRAG** to retrieve information for step revision, ensuring command suitability in customized scripts:

$$T^{\star}_{1:i} = p_\theta(\cdot | I, T^{\star}_{1:i-1}, T_i, R_i) \quad (3)$$

## Evaluation

| Design | Baseline | | | | GPT-4o | | | | Claude 3.5 Sonet | | | | ChatLS (Ours) | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Timing (ns) | | | Area | Timing (ns) | | | Area | Timing (ns) | | | Area | Timing (ns) | | | Area |
| | WNS | CPS | TNS | ($\mu m^2$) | WNS | CPS | TNS | ($\mu m^2$) | WNS | CPS | TNS | ($\mu m^2$) | WNS | CPS | TNS | ($\mu m^2$) |
| aes | -0.16 | -0.16 | -31.64 | 16577.12 | -0.17 | -0.17 | -31.64 | 16408.21 | -0.17 | -0.17 | -30.70 | 16126.78 | 0.00 | 0.00 | 0.00 | 15919.04 |
| dynamic_node | -0.08 | -0.08 | -0.45 | 21155.51 | 0.00 | 0.01 | 0.00 | 16327.08 | 0.00 | 4.06 | 0.00 | 20048.95 | 0.00 | 4.85 | 0.00 | 18907.28 |
| ethmac | -0.54 | -0.54 | -76.55 | 80533.36 | -0.55 | -0.55 | -75.89 | 80502.77 | -0.54 | -0.54 | -128.99 | 81993.70 | -0.47 | -0.47 | -55.72 | 80349.56 |
| jpeg | -1.17 | -1.17 | -439.66 | 107612.16 | 0.00 | 0.00 | 0.00 | 57227.24 | 0.00 | 0.00 | 0.00 | 66106.85 | 0.00 | 0.00 | 0.00 | 67290.02 |
| risv32i | 0.00 | 0.59 | 0.00 | 10241.00 | 0.00 | 0.59 | 0.00 | 10241.00 | 0.00 | 0.60 | 0.00 | 9711.66 | 0.00 | 0.74 | 0.00 | 9329.95 |
| swerve | 0.00 | 0.81 | 0.00 | 161551.64 | 0.00 | 0.79 | 0.00 | 143557.54 | 0.00 | 1.86 | 0.00 | 158364.70 | 0.00 | 2.05 | 0.00 | 143545.30 |
| tinyRocket | -0.88 | -0.88 | -1057.89 | 44231.28 | -0.21 | -0.21 | -42.22 | 35960.81 | -0.33 | -0.33 | -484.71 | 41999.01 | -0.09 | -0.09 | -14.74 | 36070.66 |

Table 2. Performance Comparison for Logic Synthesis Script Customization *(Pass@5)*.

- **Evaluation Settings**
  - *Logic Synthesis Tool*
    Design Compiler
  - *Circuit Designs*
    Part of the OpenROAD [11] example
  - *Technology Library*
    *Nangate 45nm* with the *5K_heavy_1k* wireload model
  - *Synthesis Script*
    Configured according to the OpenROAD example.
  - *Baselines*
    GPT-4o [8] (Version 2024-08-06)
    Claude 3.5-Sonet [12] (Version 2024-10-22)
  - *Resynthesis Iterations*: One