

# 大语言模型在电子设计自动化中的应用综述

王 静<sup>1</sup>, 林立宇<sup>2</sup>, 李 磊<sup>3</sup>, 陈霖谊<sup>1</sup>, 李 征<sup>2</sup>, 来 耀<sup>3</sup>,  
李 炎<sup>2</sup>, 姚佳飞<sup>1</sup>, 曾晓洋<sup>2</sup>, 郭宇锋<sup>1\*</sup>

(1. 南京邮电大学集成电路科学与工程学院, 江苏南京 210023; 2. 复旦大学集成电路与微纳电子创新学院, 上海 200433;  
3. 香港大学计算与数据科学学院, 香港 999077)

**摘 要:** 随着集成电路设计复杂度呈指数级增长, 传统电子设计自动化(Electronic Design Automation, EDA)工具面临着前所未有的挑战。近年来, 大语言模型(Large Language Models, LLM)在自然语言处理领域取得的突破性进展为EDA领域带来了新的机遇。本文系统总结了LLM在EDA领域的应用研究, 将其归纳为5个主要方向: 寄存器传输级(Register Transfer Level, RTL)代码生成与优化、高层次综合与逻辑优化、后端实现验证与安全、模拟电路设计自动化, 以及EDA工具流程自动化与辅助。本文同时总结了LLM在EDA领域所面临的挑战, 包括高质量领域数据的稀缺性与多模态信息处理能力不足、EDA工具链的集成障碍等技术问题, 以及知识产权风险、数据隐私保护和伦理合规性等应用层面的制约因素。未来研究方向主要集中在领域专用语料库的构建、跨模态信息融合的模式架构设计、标准化工具接口协议的制定, 以及行业规范与伦理框架的建立等方面。

**关键词:** 大语言模型; 电子设计自动化; 硬件描述语言; 代码生成; 检索增强生成

**基金项目:** 国家自然科学基金(No.U23B2042, No.62334003); 南京市科技计划(No.202309003); 南京邮电大学引进人才自然科学研究启动基金(No.NY225066)

**中图分类号:** TN402; TN391

**文献标识码:** A

**文章编号:** 0372-2112(2026)06-2377-17

**电子学报URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20250971

## A Survey on the Applications of Large Language Models in Electronic Design Automation

WANG Jing<sup>1</sup>, LIN Liyu<sup>2</sup>, LI Lei<sup>3</sup>, CHEN Linyi<sup>1</sup>, LI Zheng<sup>2</sup>, LAI Yao<sup>3</sup>,  
LI Yan<sup>2</sup>, YAO Jiafei<sup>1</sup>, ZENG Xiaoyang<sup>2</sup>, GUO Yufeng<sup>1\*</sup>

(1. School of Integrated Circuit Science and Engineering, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China;  
2. School of Integrated Circuits and Microelectronic Innovation, Fudan University, Shanghai 200433, China;  
3. School of Computing and Data Science, The University of Hong Kong, Hong Kong 999077, China)

**Abstract:** As the complexity of integrated circuit design grows exponentially, traditional electronic design automation (EDA) tools face unprecedented challenges. In recent years, breakthrough advances in large language models (LLMs) in the field of natural language processing have brought new opportunities to the EDA domain. This paper systematically reviews the application of LLMs in EDA, categorizing them into five main directions: register transfer level (RTL) code generation and optimization, high-level synthesis and logic optimization, physical implementation, verification and security, analog circuit design automation, and EDA tool flow automation and assistance. This paper also analyzes the main challenges faced by LLMs in the EDA domain, including technical issues such as scarcity of high-quality domain data and insufficient multi-modal processing capabilities, heterogeneous tool chain integration barriers, and verification of generated content correctness and security, as well as application constraints including intellectual property risks, data privacy protection, and ethical compliance. Future research directions mainly focus on domain-specific corpus construction, cross-modal fusion model architecture, standardized tool interface protocols, and industry standards and ethical frameworks.

**Keywords:** large language models; electronic design automation; hardware description language; code generation; retrieval-augmented generation

**Foundation Item(s):** National Natural Science Foundation of China (No.U23B2042, No.62334003); Nanjing Science and Technology Plan (No.202309003); Natural Science Research Start-up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (No.NY225066)

## 0 引言

电子设计自动化(Electronic Design Automation, EDA)是现代集成电路(Integrated Circuit, IC)设计的核心技术支柱,通过自动化工具和算法实现从概念设计到物理版图的全流程数字化处理,简化了电路设计流程并提升了设计效率和质量。然而,随着摩尔定律逼近物理极限、芯片设计复杂度呈指数级增长以及先进封装与三维集成等新兴技术的涌现,传统EDA方法面临设计规模爆炸、工艺约束复杂化、设计空间探索困难、抽象层级语义鸿沟等挑战,亟需新的技术范式突破现有瓶颈<sup>[1]</sup>。

近年来,以GPT(Generative Pre-trained Transformer)、Claude等为代表的大语言模型(Large Language Model, LLM)在自然语言处理(Natural Language Processing, NLP)领域取得了突破性进展,国内也涌现出DeepSeek、Qwen(通义千问)等具有竞争力的大模型<sup>[2]</sup>。此类模型通过在海量文本和代码语料上进行预训练,习得了语言理解、代码生成和逻辑推理等能力,为EDA领域的智能化变革提供了技术基础。通过对近年相关文献的系统梳理,LLM在EDA领域的应用主要体现在以下5个方向:(1)寄存器传输级(Register Transfer Level, RTL)代码生成与优化,将自然语言设计规范转换为硬件描述语言(Hardware Description Language, HDL)代码;(2)高层次综合与逻辑优化,辅助算法硬件映射及逻辑电路的面积、功耗、时序优化;(3)后端实现验证与安全,支持布局布线验证、时序分析及硬件安全检测;(4)模拟电路设计自动化,实现电路拓扑生成与参数优化;(5)EDA工具流程自动化与辅助,通过自然语言交互提升设计流程自动化。

尽管LLM在EDA领域的应用探索中取得了初步

进展,但其与EDA工作流程的系统性集成仍面临诸多挑战。从技术层面来看,主要挑战集中于3个方面:一是领域专业知识的有效获取,通用预训练语料难以涵盖半导体工艺、电路设计和EDA工具的专业知识;二是多模态信息的协同处理,需要融合电路图、时序图、版图等非文本设计信息;三是工业级精度与可靠性的保障,生成结果需满足芯片设计的正确性要求。现有研究主要采用检索增强生成(Retrieval-Augmented Generation, RAG)技术、多模态理解方法和多阶段验证机制予以应对,但在知识覆盖的全面性、跨模态语义一致性及系统性可靠性保障方面仍存在不足。从实践层面来看,工具链集成的复杂性、计算资源的开销、用户接受度的提升以及知识产权的保护等问题,进一步制约了LLM技术在实际EDA工业环境中的规模化部署。

基于上述背景,本文旨在系统分析大语言模型在集成电路设计领域的应用现状与发展前景,为未来研究提供指引。本文主要贡献包括:(1)梳理LLM在EDA 5个应用方向的技术现状,涵盖从RTL设计到物理实现的完整流程;(2)分析当前研究的关键技术方法和实现路径;(3)评估LLM在EDA应用中的技术局限性和工程化挑战;(4)提出未来研究的重点方向和发展路径。

## 1 技术背景

### 1.1 电子设计自动化概述

集成电路设计是一个系统化、层次化的复杂工程过程,涵盖从初始概念到最终制造的多个关键阶段<sup>[3]</sup>。EDA工具通过软件自动化实现设计流程的各个环节,极大地提升了设计效率和质量,使得复杂芯片的设计成为可能。如图1所示,现代集成电路设计流程主要包含以下几个阶段。

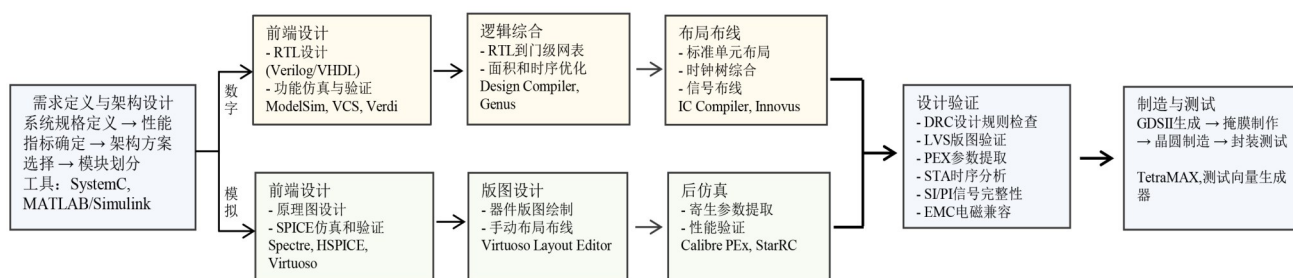


图1 集成电路设计流程

Figure 1 Integrated circuit design flow

(1) 需求定义与架构设计:首先确定系统规格定义,明确性能指标(如功耗、面积、速度)和功能需求,建立芯片级的技术指标体系。随后进行系统架构规

划和模块划分,决定芯片的整体结构、功能分配和接口定义,为后续设计奠定基础。这一阶段主要依赖系统级设计工具和建模验证平台。

(2) 前端设计:根据架构设计分为数字电路和模拟电路两条并行路径。数字电路设计使用 VHDL 或 Verilog 等 HDL 进行 RTL 描述,通过功能仿真与验证确保逻辑正确性,主要工具包括 ModelSim、VCS 等仿真器;模拟电路设计采用原理图设计方式,使用 Spectre、HSPICE 等 SPICE 仿真器进行电路行为验证,确保电路性能满足指标要求。

(3) 逻辑综合:数字电路通过 Design Compiler 等逻辑综合工具将 RTL 代码自动转换为门级网表,同时进行时序约束和面积优化;模拟电路则直接进入版图设计阶段,使用 Virtuoso 等工具精确地进行手动布局布线,确保电路性能和匹配特性。

(4) 后端物理设计:数字电路使用 IC Compiler、Innovus 等工具进行自动化布局布线,包括布图规划、标准单元布局、时钟树综合和信号布线等步骤;模拟电路完成版图设计后需要进行后仿真验证,包括 PEX (Parasitic Extraction) 和版图后仿真,确保版图实现与电路设计的一致性。

(5) 设计验证:数字和模拟电路汇合后进行全面的设计验证,包括 DRC (Design Rule Check) 验证设计是否满足制造工艺规则、LVS (Layout Versus Schematic) 验证版图与原理图的一致性、PEX 提取寄生参数进行精确的时序和功耗分析。此外还需要进行 EMC (ElectroMagnetic Compatibility) 分析、SI (Signal Integrity) 和 PI (Power Integrity) 验证。

(6) 制造与测试:通过所有验证后,生成制造所需的 GDSII 版图数据,进行光刻掩膜制作。同时使用 TetraMAX 等工具生成测试向量,为后续的晶圆级测试和封装测试作准备。最终完成晶圆制造、封装与测试,验证芯片功能与性能是否达到设计目标。

传统 EDA 工具主要基于确定性算法和专家规则系统,依赖预定义的设计规则和工艺约束进行自动化处理,通过标准化工具链实现设计流程自动化。这类工具专业门槛较高,要求工程师具备丰富的领域知识,且不同设计阶段需要专门工具协作,涉及复杂的数据交换和流程管理。

## 1.2 大语言模型概述

LLM 是基于 Transformer 架构的深度学习模型,通过自监督学习在大规模文本语料上进行预训练,习得语言统计规律和知识表示<sup>[4]</sup>。自 2017 年 Vaswani 等人<sup>[4]</sup>提出 Transformer 架构以来,LLM 经历了从编码器-解码器到纯解码器的架构演进,模型规模从早期 BERT 的 3.3 亿参数发展至 GPT-3 的 1 750 亿参数,参数规模增长超过 500 倍,并开始展现出涌现能力<sup>[5]</sup>。

如图 2 所示,LLM 通常采用“预训练+微调+对齐”

的三阶段训练范式:预训练阶段在海量词元(token)语料库上进行自监督学习;监督微调(Supervised Fine-Tuning, SFT)阶段通过指令-响应数据对增强模型的指令遵循能力;偏好对齐阶段通过人类反馈强化学习(Reinforcement Learning from Human Feedback, RLHF)或直接偏好优化(Direct Preference Optimization, DPO)等技术使模型输出符合人类偏好和安全要求<sup>[6-7]</sup>。这一训练流程使 LLM 形成了四项核心能力:一是上下文学习(In-Context Learning, ICL)能力,使模型无需参数更新即可适应新任务;二是大规模知识存储,形成跨领域的分布式知识表示;三是复杂推理能力,支持多步骤逻辑推理和问题分解;四是多模态扩展潜力,大规模预训练形成的语义空间为跨模态对齐处理了奠定基础,并催生了一系列多模态大模型(Multimodal Language Language Model, MLLM)。围绕这些核心能力,近年来涌现出思维链提示、提示工程、检索增强生成和智能体架构等重要技术,进一步增强了 LLM 的实用性和适用范围。

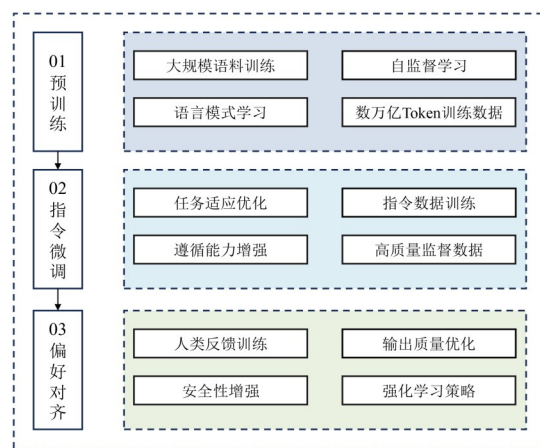


图 2 LLM 技术架构

Figure 2 LLM technical architecture

LLM 的核心能力与集成电路设计需求形成良好匹配。上下文学习能力使 LLM 能够从设计示例中快速掌握设计模式和约束规则,无需大规模标注数据;大规模知识存储整合了电路设计、工艺技术、系统架构等多领域专业知识,支持跨领域协同设计;复杂推理能力可辅助电路逻辑关系分析、时序约束处理和性能优化决策;多模态扩展潜力为电路图、时序图、仿真波形等多种设计表示形式的统一理解提供了可能。同时,LLM 的自然语言理解能力可解析设计规范和技术文档,为需求到实现的自动化转换提供语义基础;代码生成能力则直接支持 RTL 代码自动生成和逻辑分析。这些特性的协同作用有望降低 EDA 工具的使用门槛,提升设计效

率,推动集成电路设计从专家驱动模式向智能化辅助模式转变。

## 2 LLM在EDA领域的应用

如图3所示,通过对近年发表的相关研究文

献的系统梳理分析,本文将 LLM 在 EDA 领域的应用归纳为 5 个主要方向:RTL 代码生成与优化、高层次综合与逻辑优化、后端实现验证与安全、模拟电路设计自动化以及 EDA 工具流程自动化与辅助。

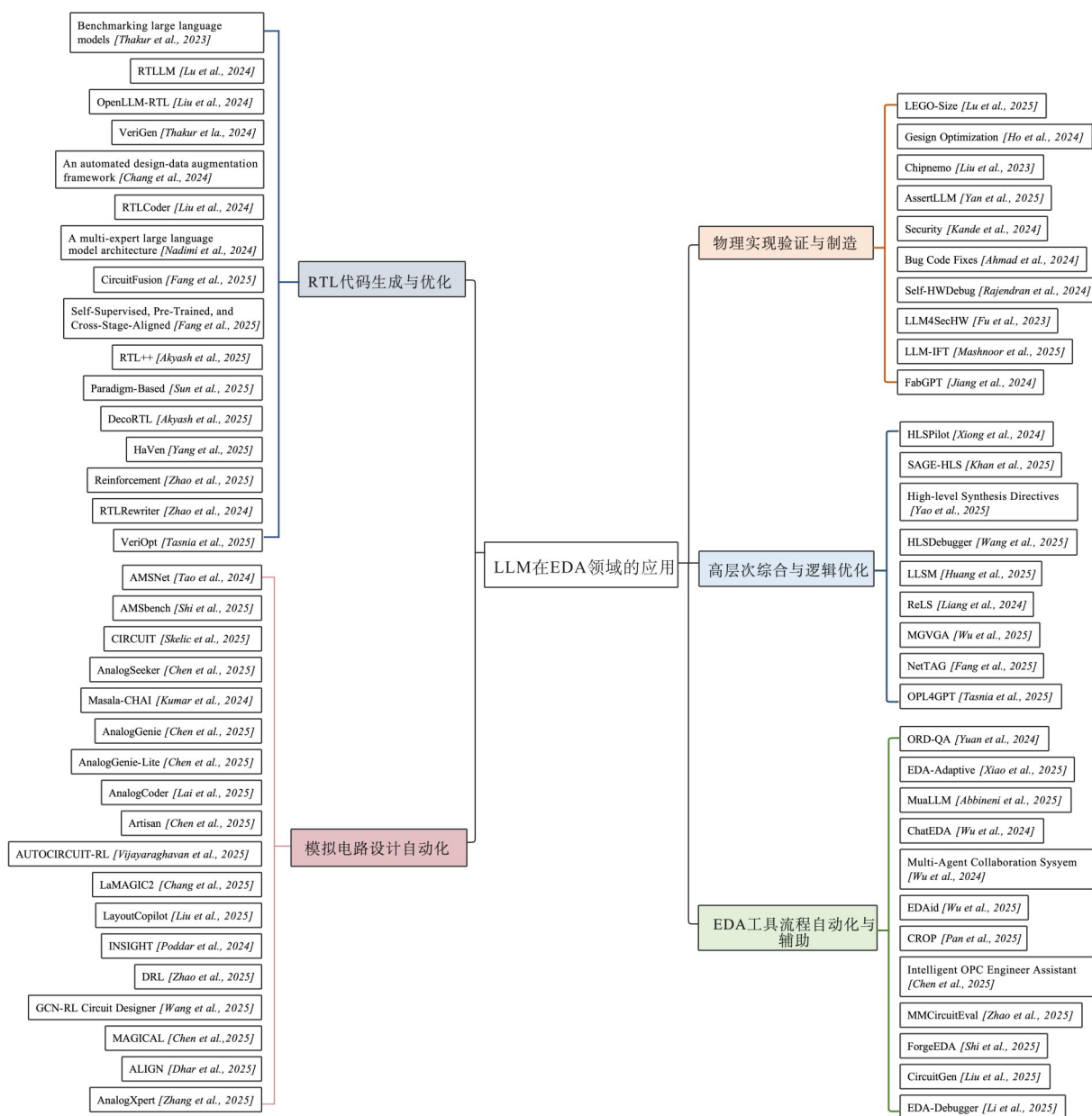


图3 LLM在EDA领域的应用

Figure 3 Applications of LLM in EDA

### 2.1 RTL代码生成与优化

大语言模型在硬件描述语言代码生成和优化领域的应用研究聚焦于将自然语言形式的设计规范转化为功能正确的 Verilog/VHDL 代码。相关研究涵盖数据集构建、模型架构创新和代码优化 3 个方面。

在数据集构建与专用模型发展方面,早期硬件设计领域缺乏标准化的 LLM 评估方法。针对这一问题,Thakur 等人<sup>[8]</sup>首次构建了包含不同复杂度 RTL 设计任务的评估基准,系统性评估结果显示模型在处理复杂时序逻辑和设计约束方面存在局限性。为弥补

评估基准规模较小的不足,RTL<sup>[9]</sup>在此基础上提供了涵盖从简单组合逻辑到复杂状态机设计的开源基准数据集,建立了标准化的质量评估指标体系。随着评估体系完善,数据质量问题逐渐凸显,Liu 等人<sup>[10]</sup>发布了 OpenLLM-RTL 框架——包含 7 000 个经过验证的高质量样本的数据集和基准测试平台,扩展了该领域的数据资源。但该方法依赖 LLM 本身进行验证,可能误判正确样本导致高质量数据受限。此外,为提高数据集质量,Thakur 等人<sup>[11]</sup>创建了大规模 Verilog 代码训练语料库——VeriGen,并且基于该语料库对多个预训练模型进行领域特定微调,验证了领域专门化训练的有效性。然而,自然语言与 Verilog 代码不对齐、训练数据质量参差以及高昂人力成本仍是关键瓶颈。Chang 等人<sup>[12]</sup>进一步提出自动化设计数据增强框架,采用 Verilog 抽象语法树增强、代码自动分割和 EDA 工具脚本数据增强等技术自动化地提升数据质量。

在模型架构创新方面,研究者探索了多专家混合、多模态融合和结构化表示等技术路线,技术经历了从传统监督学习到自监督表示学习再到 LLM 直接生成的演变,当前主流是大语言模型结合领域知识的提示工程或轻量微调框架。由于 LLM 难以同时处理多样化的设计需求,Nadimi 等人<sup>[13]</sup>提出的 MEV-LLM 架构采用多专家混合策略,集成多个针对不同专门数据集微调的模型以增强代码生成的针对性。然而该框架依赖纯文本表示,缺乏电路结构信息分析,导致模型难以理解模块间的层次关系。为解决这一问题,CircuitFusion<sup>[14]</sup>作为多模态、实现感知的电路编码器,融合硬件代码、结构图和功能总结三种模态信息,将电路设计编码为支持多种下游任务的通用表示。CircuitEncoder<sup>[15]</sup>在此基础上提出自监督、预训练、跨设计阶段对齐的通用电路编码器架构,通过在寄存器传输级和网表两个阶段采用多阶段对比学习实现电路嵌入对齐,相较于 CircuitFusion 更加简化通用。为提升 LLM 对硬件内在层次模块关系及控制逻辑的理解,RTL++<sup>[16]</sup>将 RTL 代码编码为文本化的控制流图和数据流图表示,增强了模型对结构化上下文信息的处理能力。

在代码质量保障方面,研究者针对生成过程中的幻觉和错误问题提出了多种解决方案。为抑制 LLM 幻觉问题,基于范式块的 HDL 代码生成方法利用大语言模型对电路功能块进行分类,通过与预定义范式块的匹配机制减少代码生成中的幻觉<sup>[17]</sup>。但该方法依赖预定义的范式模板,泛化能力有限。从解码策略角度,Akyash 等人<sup>[18]</sup>提出的 DecoRTL 采用无需额外模型微调的运行时解码优化策略,通过结合自洽性采

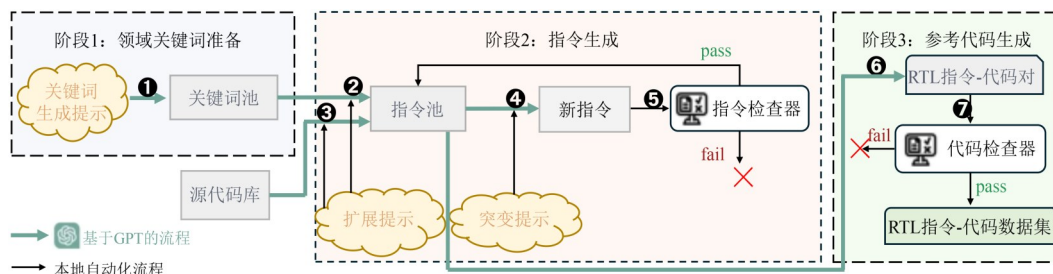
样和语法感知的温度调节机制解决幻觉、重复和无效输出等问题。相较于在推理时优化,HaVen<sup>[19]</sup>框架通过思维链推理机制将符号化的设计模式转换为自然语言表述,使代码生成过程与硬件描述语言工程师的设计实践保持一致。进一步地,Zhao 等人<sup>[20]</sup>提出的基于强化学习的优化框架通过集成抽象语法树和数据流图的结构化信息,提高生成代码的结构准确性和功能正确性。

在代码优化与性能提升方面,研究者关注生成代码的电路性能指标优化。由于传统方法依赖专家经验且编辑器能力有限,Yao 等人<sup>[6]</sup>提出了 RTL Rewriter——一种基于大语言模型构建 RTL 代码优化框架,通过应用检索增强生成和示例学习技术,设计电路分区流水线 and 成本感知的蒙特卡罗树搜索算法,RTL Rewriter 可以在复杂电路优化任务中的性能超过传统编译器优化方法。与 RTL Rewriter 优化已有代码不同,VeriOpt<sup>[21]</sup>是一个端到端生成框架,从自然语言描述到生成优化后的 RTL,通过将大语言模型的交互结构分解为多个专门的功能模块,在工业级设计项目中降低了功耗消耗、减少了芯片面积并改善了时序性能。除了优化电路性能指标,模型自身的效率也是值得关注的研究方向。RTLCoder<sup>[22]</sup>通过定制化训练算法开发了 70 亿参数规模的专用模型,在 VerilogEval Machine 基准测试中的性能超越 GPT-3.5 并接近 GPT-4,并通过 4 比特量化技术将模型压缩至 4 GB,实现了在普通笔记本电脑上的部署。如图 4 所示,RTLCoder 采用了三阶段自动化训练数据生成流程:首先准备领域关键词池,然后通过扩展和变异策略生成多样化的设计指令并经指令检查器验证,最后为合格指令生成 RTL 参考代码并通过代码检查器验证,最终构建出包含超过 27 000 个高质量指令-代码对的训练数据集。整个流程结合 GPT 驱动和本地自动化机制,有效解决了 IC 设计领域训练数据稀缺的问题,这种自动化数据生成方法也为专业领域的大语言模型训练提供了可借鉴的范式。

综合来看,RTL 代码生成领域呈现出数据、模型与优化三条协同演进的技术路线。数据集构建从规模扩张转向质量提升与自动化增强,缓解了人工标注的成本约束;模型架构从纯文本处理演进至多模态融合,对电路层次关系的建模能力逐步增强;代码质量与性能优化则经历了从功能正确性保障到电路性能指标提升。这三条路线相互支撑:数据集质量为模型训练提供基础支撑,而模型能力提升则为复杂优化目标的实现提供技术前提。

## 2.2 高层次综合与逻辑优化

高层次综合与逻辑优化是硬件设计流程中的核

图4 RTLCode自动化训练数据集生成流程<sup>[22]</sup>Figure 4 RTLCode automated training dataset generation flow<sup>[22]</sup>

心环节,直接影响最终电路的性能、面积和功耗等关键指标。大语言模型为该领域的自动化综合优化与质量提升提供了新的技术手段。

高层次综合通过 C/C++ 等高级语言作为中间表示,将设计抽象层次提升,相比直接生成 RTL 代码降低了自然语言与硬件实现之间的转换难度。2024 年, Xiong 等人<sup>[23]</sup>在 ICCAD 上提出了 HLSPilot 框架,将 LLM 应用于从自然语言到 HLS 代码的转换流程。该框架建立了覆盖多种代码模式的优化策略库,使 LLM 能够通过示例学习完成代码转换;同时引入自动化设计空间探索补充 LLM 在优化参数确定上的不足,实现了从应用程序到硬件加速器的端到端流程。尽管 HLSPilot 等工作开创了 LLM 辅助 HLS 设计的新方向,但仍主要依赖通用模型结合提示工程技术。2025 年, Khan 等人<sup>[24]</sup>提出的 SAGE-HLS 框架标志着从提示工程向领域专用模型微调范式的转变。该工作通过 Verilog-to-C/C++ 移植方法构建了包含 16 700 个样本的 HLS 代码数据集,利用抽象语法树指导的指令微调使模型学习代码结构化特征,并开发了基于 VerilogEval 的半自动化验证系统。实验结果显示,基于 QwenCoder-2.5 (7 B) 微调的模型在可综合性和功能正确性上分别达到接近 100% 和 75%。除代码生成外, HLS 性能优化同样是设计流程的重要环节。Yao 等人<sup>[25]</sup>将 LLM 应用于 HLS 指令的设计空间探索,将其作为特征提取器和决策智能体,通过微调使其学习 HLS 指令的潜在特征表示,用于指导设计空间探索策略。实验结果表明,该方法在归一化 ADRS 指标上相比现有算法提升 15%,在有限采样步数下改善了 Pareto 前沿的优化效率。此外,调试环节同样是 HLS 设计流程的关键瓶颈。2025 年, Wang 等人<sup>[26]</sup>提出的 HLSDebugger 则专注于 HLS 逻辑错误的自动化识别与修正。该工作构建了包含 300 000 样本的逻辑错误标注数据集,采用编码器-解码器架构实现错误定位、类型预测和代码修正的多任务联合学习。实验表明, HLSDebugger 在错误识别上优于 GPT-4,在错误修正上性能提升超过 3 倍。

逻辑综合与电路优化环节则是将 RTL 描述转换为门级网表,其综合质量直接决定电路的最终性能。逻辑综合过程复杂且耗时,设计人员难以在综合前准确评估设计质量,导致设计空间探索效率低下。针对这一问题, Huang 等人<sup>[27]</sup>提出的 LLSM 模型通过 EDA 引导的思维链提示和文本-电路混合嵌入技术,在延迟与面积预测任务中分别实现 21.53% 和 19.27% 的损失降低,提升了综合质量评估的准确性。然而,仅预测综合质量无法主动改善综合结果,逻辑综合过程本身的优化仍依赖人工经验调整综合策略。Liang 等人<sup>[28]</sup>提出的 ReLS 框架引入检索增强机制,使 LLM 能够从历史综合经验中提取和迁移优化知识,提升了综合策略的自动化选择能力和综合质量。值得注意的是,上述技术的有效性依赖于电路的准确表示,但现有电路表示方法还面临着逻辑等效性保持和功能抽象不足的问题。因此, Wu 等人<sup>[29]</sup>提出的 MGVGA 模型采用约束掩码建模与 Verilog-AIG 对齐技术,构建了多粒度图表示,增强了电路的结构化语义学习能力,在 QoR 预测和逻辑等效识别任务中取得了更优表现。然而,现有图表示方法难以同时编码复杂门功能语义和全局结构信息。针对这一问题, Fang 等人<sup>[30]</sup>提出的 NetTAG 框架将门级网表建模为文本属性图,通过符号逻辑表达式和物理特征作为文本属性标注门节点,采用 LLM 文本编码器捕获门语义,结合图 Transformer 编码全局结构。该框架通过门级和图级自监督目标进行预训练,并与 RTL 和版图阶段对齐,使其能够处理多种门类型并支持功能和物理任务。实验表明, NetTAG 在四类不同任务上均优于任务特定方法和现有 AIG 编码器,展现了跨 RTL-综合-物理设计阶段的通用性。此外,在综合流程之前的设计阶段,硬件描述语言和编程范式的选择同样影响后续综合效果。Tasnia 等人<sup>[31]</sup>基于 GPT-4o 构建了自动化推荐框架,能够根据设计需求推荐最优的硬件描述语言和编程范式,为后续综合流程提供决策支持。

综上所述,大语言模型在高层次综合与逻辑优化

领域的应用涵盖了从高级语言到硬件实现的多个环节,包括 HLS 代码转换、设计空间探索、综合质量预测、综合策略优化以及电路表示学习等方面。技术路径呈现从通用模型提示工程向领域专用模型微调演进的趋势,并通过检索增强、多任务学习和多粒度表示等机制提升了设计流程的自动化水平。然而,电路语义的深层理解和跨设计场景的泛化能力仍是该领域需要进一步探索的关键问题。

### 2.3 物理实现验证与制造

后端实现验证与安全环节涵盖了从物理设计优化到功能验证测试,再到安全分析和制造检测的完整链条。该阶段需要处理复杂的物理约束、时序关系和安全属性,对自动化技术的精确性和可靠性提出了更高要求。大语言模型在该领域的应用探索了多个关键技术方向。

物理设计阶段需要在满足时序、功耗和面积约束的前提下完成布局布线和门级优化,传统方法依赖专家经验和启发式算法。门尺寸调整是物理设计中的关键优化步骤,但在先进工艺节点(如 3 nm),商业工具面临标准单元库规模庞大和片上变异感知时序分析复杂度高的挑战,导致优化过程需要多次迭代且耗时严重。Lu 等人<sup>[32]</sup>提出的 LEGO-Size 方法将时序路径表示为标记化标准单元序列,采用 Transformer 编码器架构从全局视角进行尺寸决策,通过自监督学习和有监督微调提升泛化能力,并集成可微分静态时序分析引擎优化时序指标。实验结果表明,该方法在 5 个百万门级工业设计上相比商业签核工具实现了 125 倍加速和 37% 的 TNS 改进。除门尺寸优化外,标准单元版图设计同样是物理实现阶段的核心问题。随着工艺技术向 2 nm 演进,由于布线资源减少、设计规则复杂度提升和严格的图形化约束,可布线性问题日益突出。传统自动化框架难以为复杂时序单元生成兼具优异 PPA 和可布线性的版图设计。NVIDIA 研究团队的 Ho 等人<sup>[33]</sup>利用 LLM 的自然语言理解和推理能力,通过 ReAct 提示机制逐步生成高质量的器件聚类约束,以优化单元版图 PPA 并修复可布线性问题。该方法考虑了网表拓扑、扩散区共享/断开和版图可布线性,在 2 nm 工艺节点的时序标准单元测试集上,相比现有方法实现了最高 19.4% 的单元面积缩减,并使 LVS/DRC 干净版图的生成率提升 23.5%,平均单元面积减少 4.65%。

在功能验证环节,传统方法依赖人工编写断言代码和测试用例以确保设计正确性,但编写 SystemVerilog 断言需要深入理解设计规范和时序行为,效率受限。针对这一问题,Yan 等人<sup>[34]</sup>提出的 AssertLLM 框架采用多模型协同机制,将自然语言规范和波形信号

转换为 SystemVerilog 断言,提升了基于断言验证的自动化水平。该方向的研究进一步拓展至 SVA 断言和 UVM 测试平台的自动生成,以及错误日志和波形的智能解析,为验证流程的端到端自动化提供了技术基础。

硬件安全漏洞的检测与修复是芯片设计中的关键环节,传统安全分析方法难以覆盖复杂的安全属性,且漏洞修复高度依赖专家经验。Kande 等人<sup>[35]</sup>首先验证了 LLM 在硬件安全断言生成中的可行性,使模型能够理解安全规范并自动生成检测代码。然而,仅检测漏洞无法主动提升设计安全性,漏洞修复能力同样重要。Ahmad 等人<sup>[36]</sup>就针对这一需求建立了定量评估 LLM 漏洞修复性能的框架,为衡量模型的修复能力提供了基准。在此基础上,Akyash 等人<sup>[37]</sup>提出的 Self-HWDebug 框架通过自动生成调试指令减少了人工介入需求,而 Fu 等人<sup>[38]</sup>的 LLM4SECHW 框架利用开源硬件缺陷数据集缓解了训练数据不足的问题。此外,信息流跟踪是安全分析的重要手段,但传统方法在处理大规模设计时面临可扩展性挑战。LLM-IFT<sup>[39]</sup>框架将结构化推理引入信息流跟踪,通过 LLM 的推理能力提升了安全分析在复杂设计中的适用性。

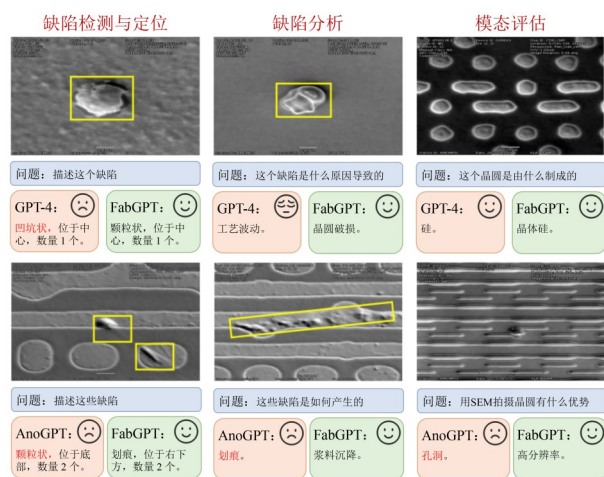
在制造环节,随着工艺节点推进至 3 nm 及以下,设计-工艺协同优化(DTCO)面临多尺度仿真、跨层次优化和海量参数空间探索等挑战。晶圆缺陷检测作为良率提升的关键环节,传统方法难以充分利用扫描电镜图像、工艺日志等多模态数据,且缺乏对缺陷成因的知识推理能力。针对这一问题,浙江大学卓成团队提出的 FabGPT 框架<sup>[40]</sup>构建了面向晶圆缺陷分析的多模态大模型。如图 5 所示,该框架采用三阶段训练策略,通过交互式语料库训练缓解模态偏差问题,使模型不仅能检测、定位和分割缺陷区域,还能基于领域知识进行缺陷成因分析。

相比前端设计环节,物理实现验证与制造领域的大语言模型应用研究仍处于探索阶段。这主要源于该阶段涉及的几何约束、工艺物理特性、时序分析和安全验证等专业知识具有更高的复杂性,同时高质量的训练数据更为稀缺,因此对模型的鲁棒性、可靠性和可解释性提出了更严格的要求。

### 2.4 模拟电路设计自动化

模拟与混合信号电路设计因其复杂的物理特性和对设计经验的高度依赖,长期以来自动化程度较低。大语言模型的引入为该领域带来了新的技术路径,相关研究从数据资源构建、模拟电路设计自动化到版图生成形成了较为完整的技术体系。

数据资源与评估体系的建立是支撑大语言模型

图5 FabGPT与GPT-4、AnoGPT的对比<sup>[40]</sup>Figure 5 Comparison of FabGPT with GPT-4 and AnoGPT<sup>[40]</sup>

应用的重要前提。在数据构建方面,技术路线呈现出从实际设计数据收集到知识蒸馏提取的演进。Tao等人<sup>[41]</sup>开发的AMSNet数据集包含晶体管级原理图及其SPICE网表,并提供了自动转换工具,为多模态模型训练提供了数据基础。但其收录的电路多为基础电路且网表信息不完整,在数据规模和完整性方面仍有提升空间。针对实际设计数据获取困难的问题,AnalogSeeker<sup>[42]</sup>通过知识蒸馏方法将教科书中的设计知识转化为结构化问答数据,构建了面向模拟电路的开源基础模型。这种基于知识提取的范式降低了对实际设计数据的依赖,但其生成的问答数据侧重概念性知识而较少涵盖实际设计经验。在评估基准方面,Skelic等人<sup>[43]</sup>提出CIRCUIT基准引入了单元测试式评估方法,涵盖510个不同难度的问答对。进一步地,Shi等人<sup>[44]</sup>开发了AMSbench,包含约8 000个跨难度级别的问题,系统性评估模型在电路感知、分析和设计方面的能力。这些工作为模拟电路领域的大语言模型研究提供了数据支撑和评估标准。

模拟电路设计自动化研究聚焦于从规格到拓扑再到参数的完整设计流程<sup>[45-49]</sup>。在拓扑生成阶段,传统方法依赖人工经验选择电路拓扑,难以系统化探索设计空间针对特定电路类型的优化需求,Chen等人<sup>[45]</sup>的Artisan框架专注于运算放大器设计,作为首个LLM模拟电路设计工作,采用双向电路表示对齐拓扑与语义,利用多智能体机制实现层次化设计流程,在成功率和性能指标上优于优化方法及通用LLM,设计加速达50.1倍。针对如何提升拓扑生成的有效性并降低对训练数据规模的依赖,Vijayaraghavan等人<sup>[46]</sup>提出的AUTOCIRCUIT-RL框架引入强化学习机制,采用指令调优使LLM从结构化提示生成电路拓扑,并通过奖励模型精炼以优化有效性和效率,相

比基线方法有效电路生成提升12%、重复率降低38%,在有限训练数据下成功率超60%。Gao等人<sup>[50]</sup>提出的AnalogGenie作为生成引擎,针对模拟电路领域数据集缺乏和表示方法不足的问题,构建了综合性的拓扑数据集,并开发了基于序列的图表示方法和基于欧拉回路的拓扑生成机制,实验结果显示其能够扩展电路多样性、增加设计中的器件数量并发现未见过的拓扑结构,其轻量级版本<sup>[51]</sup>进一步提升了可扩展性。Bhandari等人<sup>[52]</sup>的Masala-CHAI框架利用大语言模型的多模态能力解决了从原理图图像到SPICE网表的自动生成难题,通过标注、提示词调优和网表验证的三步流程实现端到端转换,该框架收集了涵盖10本教科书中7 500个不同复杂度原理图的语料库,在其上微调的模型应用于智能体框架时可使Pass@1分数提升46%。

此外,免训练方法探索为降低训练数据需求提供了新思路,Lai等人<sup>[53]</sup>提出的AnalogCoder是首个通过Python代码生成设计模拟电路的免训练LLM智能体,采用定制化领域特定提示词的反馈增强流程实现自动化设计和自我纠错,并提出电路工具库将成功设计归档为可重用的模块化子电路以简化复合电路创建,在涵盖广泛模拟电路任务的基准上成功设计20个电路,比标准GPT-4o多5个。其增强版本AnalogCoder-Pro<sup>[54]</sup>统一了生成与优化技术,引入利用仿真错误信息和波形图像的多模态诊断修复反馈循环实现设计错误的自主纠正,并通过从目标规格生成拓扑、提取关键参数、应用贝叶斯优化进行器件尺寸优化的流程实现端到端自动化,在涵盖13种电路类型的基准上成功设计28个电路。

版图生成与性能优化环节需要将电路拓扑转化为满足工艺规则和性能指标的物理版图,传统流程依赖人工调整且难以处理设计-性能的复杂映射关系。LayoutCopilot<sup>[55]</sup>构建了多智能体协作框架,利用自然语言理解能力将设计意图转化为可执行脚本,实现了交互式版图设计。然而,版图优化依赖SPICE仿真进行性能评估,仿真耗时成为制约效率优化的关键瓶颈。INSIGHT<sup>[56]</sup>基于自回归Transformer将性能预测建模为序列生成任务,加速了仿真流程。在此基础上,Gao等人<sup>[57]</sup>提出的仿真驱动框架将版图生成集成到尺寸优化循环中,利用贝叶斯优化直接优化后仿真性能,形成了从拓扑到版图的闭环优化。此外,传统方法在该领域也取得了重要进展,MAGICAL<sup>[58]</sup>和ALIGN<sup>[59]</sup>作为开源的全自动化版图系统,分别通过层次化策略和自动约束推断实现了从网表到版图的自动生成,为大语言模型方法提供了技术参照。

相比数字电路设计,模拟电路领域的大语言模型应用研究仍处于探索阶段。这主要源于模拟电路设计中涉及的连续参数空间、复杂的非线性关系、对工艺特性的敏感性以及性能指标的多维权衡,对模型的物理理解能力、优化精度和设计可靠性提出了更高要求,同时高质量的实际设计数据更为稀缺,限制了模型的泛化能力。

## 2.5 EDA 工具流程自动化与辅助

大语言模型在 EDA 工具流程自动化与辅助方面的应用涵盖知识获取、工具集成和数据支撑 3 个层面,相关研究通过智能化手段简化从 RTL 到 GDSII 的设计流程,降低专业知识门槛。

在知识获取层面,研究者基于检索增强生成技术构建智能问答系统以降低专业知识门槛。Xiao 等人<sup>[60]</sup>的 EDA-Copilot 通过构建大规模 RTL-to-GDSII 流程语料库,采用查询分类与检索增强的两阶段策略处理 EDA 数据的隐式语义关系,在答案稳定性和代码质量方面优于基线方法。针对通用 RAG 框架在 EDA 垂直领域适配不足的问题,Pu 等人<sup>[61]</sup>提出定制化 RAG 框架,通过对比学习微调文本嵌入模型、蒸馏重排序器和领域语料微调生成模型三项技术提升检索与生成性能,并发布了针对 OpenROAD 文档的问答评估基准 ORD-QA。随着设计查询复杂度提升,单次检索难以满足多步推理需求,MuaLLM<sup>[62]</sup>采用 Re-Act 工作流实现迭代推理与多步检索,通过检索与推理耦合机制支持任意规模语料库的可扩展推理,结合自适应向量数据库和实时更新能力处理复杂多模态查询,在成本和速度上分别优于传统方法 10 倍和 1.6 倍,标志着该方向从单一文本检索向多模态知识获取演进。

在工具集成方面,研究者探索了从单代理到多代理协作再到专业化应用的技术演进路线。He 等人<sup>[63]</sup>的 ChatEDA 构建了任务分解、脚本生成和执行的流程框架,将自然语言指令转化为可执行的工具操作,其微调的 AutoMage 模型在 EDA 任务处理方面优于通用大语言模型。但单代理系统在处理复杂设计任务时难以兼顾多环节的专业性需求多代理协作机制成为解决这一问题的新方向。相关研究通过多个专业化代理分别负责设计、验证等不同环节以提高系统可靠性<sup>[64]</sup>,EDAid 的实验结果表明多智能体系统在自动化成功率和跨平台通用性方面明显优于单代理架构。在通用流程自动化框架基础上,研究者进一步将大语言模型应用于特定设计环节的深度优化。Pan 等人<sup>[65]</sup>的 CROP 框架通过设计嵌入实现相似设计检索以优化工业设计质量。Chen 等人<sup>[66]</sup>将大语言模型应用于光学邻近校正问题,实现针对对不同设计拓扑的

OPC 配方构建。Li 等人<sup>[67]</sup>的 EDA-Debugger 聚焦于运行时问题的自动诊断与解决,呈现出从通用自动化向专业化纵深发展的趋势。

在数据支撑方面,评估基准和数据集的构建为上述应用提供了发展基础。由于缺乏跨领域的综合性评估体系,Zhao 等人<sup>[68]</sup>的 MMCircuitEval 构建了涵盖数字、模拟和物理设计等多个领域的综合性多模态评估基准,结合文本和图像等多种模态评估模型能力。为弥补训练数据规模和类型的不足,ForgeEDA<sup>[69]</sup>作为开源多模态数据集,也涵盖了从 RTL 代码到布局后网表等多种电路表示形式,在规模和类型上扩展了现有数据资源。随着数据需求持续增长,人工构建数据集的成本约束逐渐凸显,CircuitGen<sup>[70]</sup>提出基于深度学习的电路生成模型,通过可控方式自动生成新设计,为数据获取提供了自动化途径。

## 3 核心技术方法

为实现 LLM 在 EDA 领域的有效应用,研究者发展了模型适配与优化、知识增强与推理以及系统集成与协同三类核心技术方法。表 1 对比了当前主要的基于大语言模型的 EDA 工具在应用领域、测试基准与数据集、技术方法以及性能表现等方面的差异。

### 3.1 模型适配与优化技术

针对 EDA 领域的专业特点,研究者采用多种技术手段提升 LLM 在电子设计任务中的适用性。在领域专用微调方面,Chang 等人<sup>[12]</sup>和 Fang 等人<sup>[71]</sup>提出数据增强框架,通过 Verilog 抽象语法树增强、代码自动分割和 EDA 工具脚本数据增强等技术生成训练数据用于模型微调。RTLCoder<sup>[22]</sup>通过构建专门的 RTL 代码数据集和定制化训练方法,开发了 70 亿参数的轻量级模型用于 RTL 代码生成任务。

在多模态表示学习方面,LLSM<sup>[27]</sup>提出文本-电路混合嵌入技术,将 RTL 代码的文本表示和电路的结构表示进行融合。该技术在逻辑综合的延迟预测和面积预测任务中分别实现 21.53% 和 19.27% 的性能改进。

### 3.2 知识增强与推理技术

检索增强生成技术用于扩展 LLM 的 EDA 知识范围。检索增强生成技术用于扩展 LLM 的 EDA 知识范围。Xiao 等人<sup>[60]</sup>提出 EDA-Adaptive RAG 框架,采用查询分类器和增强检索优化器的两阶段策略。如图 6 所示,该框架的工作流程包括:首先通过查询分类器识别查询类型(术语解释、脚本检索、脚本注释、知识建议),将检索范围缩小至对应的专用语料库子集;随后采用混合检索策略,结合密集向量检索和稀疏检索进行加权融合,并利用交叉编码器对检索结果重排

表 1 基于大语言模型的 EDA 工具对比:测试基准、数据集与方法、效果及运行性能

Table 1 Comparison of LLM-based EDA tools: test benchmarks, datasets and methods, effectiveness and runtime performance

| 名称                            | 应用领域      | 测试基准及数据集                                                                                                              | 方法                                                                                  | 效果及运行性能                                                                    |
|-------------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|----------------------------------------------------------------------------|
| RTLCoder <sup>[22]</sup>      | 数字 RTL 设计 | - 基准: VerilogEval、RTL2V1.1<br>- 训练数据集: RTLCoder-27K (27 000+ 个指令-代码对, GPT-3.5 自动生成并语法检查过滤)                            | 三阶段自动化数据生成(关键词→指令→代码)+基础模型微调(Mistral-7B/DeepSeek-6.7B)+代码质量反馈训练+梯度累积                | Pass@1 达 41.6%~62.5%, 优于 GPT-3.5, 部分超越 GPT-4; 7B 参数, 4-bit 量化后 4 GB, 可本地部署 |
| HLSPilot <sup>[23]</sup>      | 高层次综合     | - 基准: 修改版 Rosetta, 每个含未优化/优化版本+L-BFGS 案例<br>- 知识库 (training-free): Xilinx HLS 手册                                      | 5 阶段框架(Profiling→程序树任务流水线→自动 HLS 优化→DSE→部署)+程序树分解+RAG+DSE+GPT-4+8 类优化策略             | 硬件运行速度提升 1.8~188 倍; 77% 场景达到或超越专家水平; 完全自动化, 但精细优化任务仍有挑战                    |
| AnalogCoder <sup>[53]</sup>   | 模拟电路设计    | - 自建基准: 24 个模拟电路设计任务, (基础电路 15 个、复合电路 9 个)<br>- 数据集: GPT-3.5、GPT-4o 和 Llama-3 生成电路样本                                  | 无需训练+Python 代码生成+提示工程+四阶段反馈增强流程(需求/仿真/DC 扫描/功能检查)+电路工具库(模块化子电路)                     | Pass@1 为 66.1%、Pass@5 为 75.9%, 优于 GPT-4o; 成功完成 83% 任务, 复杂电路仍有提升空间          |
| LayoutCopilot <sup>[55]</sup> | 模拟版图设计    | - Concrete 测试 (1 250 案例)+分类测试 (2000 案例)+功能检验+真实案例;<br>- 知识库 (training-free): 模拟设计书籍+工具手册                              | 多智能体 (6 代理)+7 模块提示工程(角色扮演/CoT/自我精炼/RAG)+GPT-3.5/4/Claude-3+交互编辑器 (15 命令)            | 多智能体+指令配置准确率达 93.75% (Claude-3 达 96.80%); 成功修复功能失效, 能权衡多项性能指标              |
| ChatEDA <sup>[63]</sup>       | EDA 脚本生成  | - 数据集: AutoMage (利用 GPT 生成约 1 500 条实例)+AutoMage2 (约 11 万条通用代码指令)<br>- 基准: ChatEDA-Bench (50 任务); 对比 GPT-3.5/4、Claude2 | Llama2+ 自指令范式 (GPT-3.5/4)+指令调优 (解释式)+QLoRA (4-bit 量化)+零样本 CoT+三阶段工作流 (任务分解→脚本生成→执行) | AutoMage2 A 级成功率 82%, 优于 GPT-4 (62%); 实现 RTL 到 GDSII 全流程自动化, 使用门槛较低        |

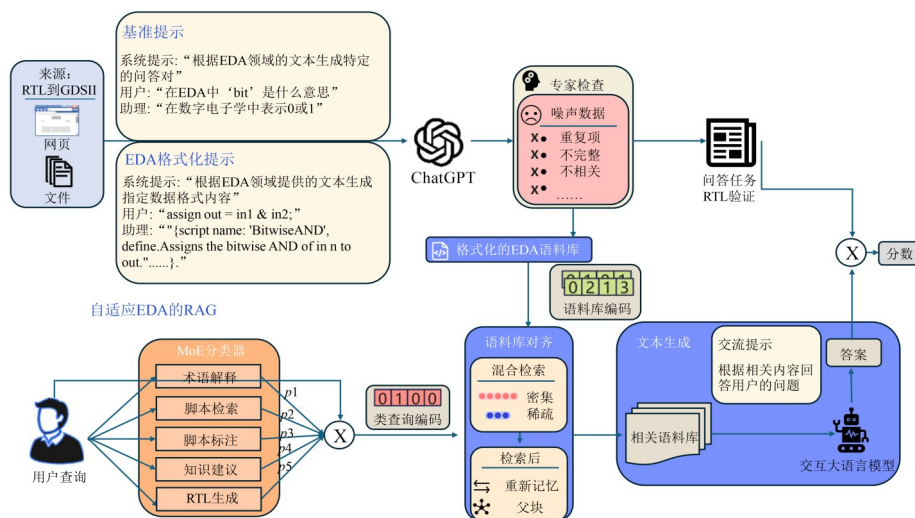
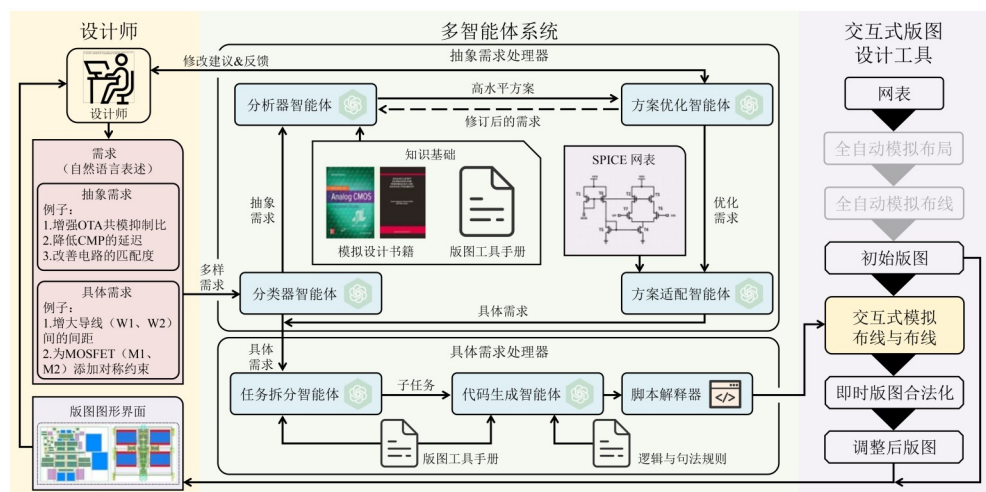
序;最后通过上下文集成和响应生成模块输出答案。实验表明,针对高难度 EDA 查询,该方法相比标准 RAG 在准确率上平均提升 12.67%~16.1%;在 RTL 代码生成任务中,语法正确率从 68% 提升至 92%,功能正确率从 52% 提升至 88%。该框架无需预训练或微调,仅通过检索增强即可实现性能提升,适用于资源受限的工业场景。

推理机制的改进主要体现在思维链技术的应用。EDA 引导思维链提示方法通过设计特定提示模板,引导 LLM 按照专家分析流程进行推理,从 RTL 代码中提取关键信息并进行决策。RTL Rewriter<sup>[6]</sup>采用检索增强技术,开发搜索引擎用于识别优化指南和代码片段,为代码重写提供知识支持。

3.3 系统集成与协同技术

多代理协作系统通过任务分解和专业化分工处理复杂的 EDA 问题。相关研究<sup>[64]</sup>提出的多代理协作框架包含专业化代理、协作机制、中央协调器和反馈

循环等组件,各代理分别负责系统设计、RTL 设计、验证等特定环节。实验结果表明,多代理系统在处理需要多领域知识和多步骤推理的 EDA 任务时相比单一 LLM 具有优势。LayoutCopilot<sup>[55]</sup>采用多代理系统架构支持交互式模拟电路版图设计。图 7 展示了该框架的整体架构,包含抽象请求处理器和具体请求处理器两大核心组件。抽象请求处理器由分类器、分析器、方案精炼和方案适配器 4 个智能体协同工作:分类器智能体区分具体请求和抽象请求;分析器智能体结合领域知识库生成高层解决方案;方案精炼智能体与设计师交互优化方案;方案适配器智能体将精炼后的方案转化为具体的可执行请求。具体请求处理器通过任务分解和代码生成将自然语言描述转换为版图工具可执行的脚本命令。多智能体分工避免了提示词稀释和注意力分散问题,使每个智能体专注于特定任务。实验显示,引入提示词工程后,基于 Claude-3 的多智能体系统总体准确率达 96.80%,相较单智能体系统准确率提升 5.89%。在实际应用

图6 EDA-Adaptive RAG方法的框架<sup>[60]</sup>Figure 6 Framework of EDA-Adaptive RAG method<sup>[60]</sup>图7 LayoutCopilot概述<sup>[55]</sup>Figure 7 Overview of LayoutCopilot<sup>[55]</sup>

中,该框架优化运算跨放放大器(OTA)电路后,增益从 $-8.75$  dB提升至 $38.26$  dB,共模抑制比从 $27.3$  dB提升至 $58.7$  dB,版图面积减小 $34\%$ ;对比较器电路噪声从 $50.3$   $\mu$ V降低至 $30.9$   $\mu$ V。

可微分优化技术将LLM与数值优化方法结合。LEGO-Size<sup>[32]</sup>提出LLM增强的GPU优化可微分VLSI门尺寸调整方法,通过LLM知识提取、可微分电路模型、GPU并行加速和多目标优化等技术,适用于先进工艺节点的门尺寸调整。HLSPilot<sup>[23]</sup>将可微分优化技术应用于高层次综合,构建基于性能分析的加速机制,实现在混合CPU-FPGA架构上的部署。

表2总结了上述核心技术在不同EDA应用场景中的代表性工作,系统展现了LLM在集成电路设计全流程中的技术实现路径。

## 4 挑战与未来研究方向

尽管LLM在EDA领域展现出应用潜力,但其实际应用仍面临技术能力、工程实践和伦理安全三个层面的挑战。本节分析制约LLM在EDA中应用的关键挑战,并提出未来研究方向。

### 4.1 领域数据构建与专用模型能力提升

高质量领域数据的稀缺性是制约LLM在EDA领域应用的核心挑战。企业数据因商业机密限制共享,公开数据质量参差不齐。研究表明,通用LLM在HDL代码生成任务中的准确率通常低于 $50\%$ ,而经过领域微调的模型可达到 $60\%$ 以上<sup>[22]</sup>,说明专业知识表征对性能的关键影响。此外,电子设计自动化过程涉及代码、电路图、布局图、时序波形等多种信息形式,这些数据的收集与整合难度大,进一步加剧了数

表2 核心技术与应用的关系及相关代表性工作  
Table 2 Relationship between core technologies and applications, and representative works

| 核心技术    | 应用类别       |            |           |           |            | 代表性工作                            |
|---------|------------|------------|-----------|-----------|------------|----------------------------------|
|         | RTL代码生成与优化 | 高层次综合与逻辑优化 | 后端实现验证与安全 | 模拟电路设计自动化 | EDA工具流程自动化 |                                  |
| 模型适配与优化 |            | √          |           |           |            | LLSM <sup>[27]</sup>             |
|         | √          |            |           |           |            | RTLCoder <sup>[22]</sup>         |
|         | √          | √          |           |           |            | MasterRTL <sup>[71]</sup>        |
| 知识增强与推理 | √          |            |           |           |            | RTL Rewriter <sup>[6]</sup>      |
|         | √          |            |           |           | √          | EDA-Adaptive RAG <sup>[60]</sup> |
|         |            | √          |           |           |            | HLSPIlot <sup>[23]</sup>         |
| 系统集成与协同 |            |            | √         |           |            | LEGO-Size <sup>[32]</sup>        |
|         |            |            |           | √         |            | LayoutCopilot <sup>[55]</sup>    |
|         |            |            |           |           | √          | ChatEDA <sup>[63]</sup>          |
|         |            |            |           | √         |            | AnalogCoder <sup>[53]</sup>      |

据稀缺问题。

如第3.1节所述,RTLCoder<sup>[22]</sup>、LLSM<sup>[27]</sup>等研究通过领域数据集构建、专用预训练和模型量化等技术,在特定任务上实现了性能提升。然而,这些尝试仍面临局限:现有数据集规模有限且覆盖范围窄,难以支撑EDA全流程的多样化任务;模型在特定任务上的过度专用化可能损害其在其他子任务上的泛化性能。此外,虽然AMSNet<sup>[41]</sup>和Masala-CHAI<sup>[49]</sup>等研究探索了从图像到网表的转换,但当前多数LLM在处理电路图和波形等非文本数据方面的能力仍然有限,特别是在理解专业图形的深层语义方面存在不足。

基于上述局限,未来研究应重点关注以下方向。

(1)构建高质量EDA领域语料库:系统收集和整合公开的开源硬件设计代码、EDA工具文档、设计规范手册(PDK文档、设计规则)以及学术论文中的方法学知识。鉴于电路设计本质上是多模态的,设计信息分散体现在代码、原理图、版图和波形等不同模态中,需要特别强化多模态数据的整合。这包括构建跨模态对齐的语料库,开发适合电路特征的多模态嵌入方法以实现不同模态信息的统一表示,从而增强模型对电路的不同表达形式的理解能力。

(2)提升有限数据的利用效率:发展数据增强技术(如通过代码变换生成语义等价的RTL变体)、少样本学习(使模型能从少量示例中快速适应新任务),以及主动学习(利用人类专家反馈迭代改进模型)等方法,在数据有限约束下提升模型性能。

(3)发展EDA任务特定的模型架构与训练范式:探索将硬件设计的结构化特性(如抽象语法树、数据流图)融入模型架构,设计电路结构感知的预训练任务(如信号依赖关系建模、设计意图推理),开发层次

化的token表示方法以保留代码的语义结构。针对多模态处理需求,需要研究跨模态注意力机制以捕捉文本、图形和波形等不同模态间的协同关系,提升模型在复杂场景下的综合理解能力。

4.2 系统集成化:克服工程实践障碍

将LLM技术集成到EDA工作流程中面临多重障碍。现有EDA工具具有复杂的架构和专有工作流程,不同厂商工具间的互操作性有限,实现LLM与这些异构工具系统的深度集成需要开发适配层、API接口和数据转换机制。部分商业EDA工具采用封闭架构设计,缺乏开放接口,进一步增加了集成难度。此外,芯片设计工程师通常具有固定的工作习惯和成熟的工具链使用经验,对新技术的采用往往较为谨慎,对AI工具可能引入不可预见的错误或设计质量产生负面影响存在担忧。

如第3.3节所述,系统集成与协同是将LLM融入EDA工具链的关键技术。LLSM<sup>[27]</sup>利用大语言模型自动生成逻辑综合工具调用序列,实现了从“工具驱动”到“任务驱动”的范式转变。HLSPIlot<sup>[31]</sup>构建了与HLS工具交互的闭环系统,能够自动解析工具报告并优化设计参数。ChatEDA<sup>[63]</sup>通过任务分解、脚本生成和执行反馈的闭环框架,实现了数字后端设计中复杂工具链的自动化编排。LayoutCopilot<sup>[53]</sup>采用多代理协作生成控制版图工具脚本,提升了设计效率。

然而,现有EDA工具采用不同的输入输出格式和调用方式,LLM与之交互需要大量定制化开发,每增加一个新工具都需要重新开发适配层,缺乏通用的集成框架;跨工具的信息传递和状态管理缺乏标准化方案,例如从综合工具到布局布线工具的数据流转往往需要人工干预和格式转换。为解决实际应用中的

系统集成障碍,需要开发 LLM 与现有 EDA 工具融合的技术方案。研究重点包括设计标准化的接口协议和中间表示层,解决不同厂商工具间的互操作性问题;构建能够执行 EDA 流程的代理系统;开发与设计工程师交互的助手系统。未来的 EDA 生态系统将通过 LLM 驱动的统一接口层实现无缝集成。该接口层能够自动理解不同工具(如 Cadence、Synopsys、Mentor 等)的语义规范,实时翻译数据格式,并生成最优的工具调用序列。通过建立工具语义的中间表示,可实现从“工具驱动”向“任务驱动”的转变,使工程师能够以自然语言描述设计目标,由系统自动选择和编排相应工具。

### 4.3 安全性、可靠性与伦理规范

集成电路设计对精确性有严格要求,以寄存器传输级代码生成为例,即使 1% 的语法错误率也可能导致功能失效。由于 LLM 作为概率生成模型在内容生成过程中存在不确定性和错误风险,直接生成的 RTL 代码首次通过仿真验证的比例较低,需要多轮迭代修正。此外,LLM 在自动生成代码或优化设计方案时,可能引入具有隐蔽性的安全漏洞,此类漏洞一旦被恶意利用,可能导致系统功能异常或敏感信息泄露。在伦理层面,LLM 预训练阶段学习的海量公开数据中不可避免地包含受知识产权保护的设计模式、电路拓扑结构和技术规范等敏感信息,其生成内容可能无意中重现或重组这些受保护的知识要素,引发知识产权争议。企业在使用基于云端的 LLM 服务时,专有设计数据和核心技术信息可能面临意外泄露风险。在我国《生成式人工智能服务管理暂行办法》框架下,使用 LLM 生成的电路设计需要满足内容安全、数据安全等合规要求。

现有研究已在验证与安全保障方面开展探索。如第 2.1 节和第 2.3 节所述,RTL Rewriter<sup>[6]</sup>在代码重写优化过程中引入验证闭环,通过形式化等价性检查确保重写后的代码功能与原设计一致。AssertLLM<sup>[37]</sup>利用 LLM 自动生成系统级断言,增强了形式化验证能力。LLM-IFT<sup>[39]</sup>通过 4 步信息流追踪检测硬件设计中的信息泄露。然而,这些方案存在局限:验证机制多为事后检查,未从生成过程上根本保证输出正确性。

为应对上述挑战,未来研究应重点关注以下方向。首先,发展完善的验证与测试技术。可将 RTL Rewriter 的验证闭环思想推广到更广泛的代码生成场景,结合 AssertLLM 的断言生成能力,形成“生成-验证-断言”的安全保障体系;开发基于形式化方法的代码正确性验证工具,不仅检查功能等价性,还验证时序约束、资源占用等非功能性属性。其次,建立知识产权保护与数据隐私机制。可开发模型水印技术,在

LLM 训练阶段嵌入不可见的标识信息,用于追溯模型生成内容的来源;开发本地化的轻量级 LLM 部署方案,从根本上避免数据外泄风险。此外,制定伦理规范与审查流程。可在模型训练阶段引入内容过滤和偏见消除机制,避免模型学习和复现受保护的设计知识;在模型部署阶段嵌入安全性检查模块,对生成内容进行审查;联合学术界、工业界和监管部门,制定 LLM 在 EDA 领域应用的行业标准和伦理规范,明确数据使用、模型训练、内容生成等环节的合规要求。随着相关技术的持续改进、行业标准的逐步完善和监管规范的不断建立,这些挑战有望通过产学研协作得到有效解决,推动 LLM 技术在确保正确性、安全性和合规性的前提下为 EDA 领域提供技术支撑。

本节从领域数据构建、系统集成和安全伦理 3 个层面系统分析了 LLM 在 EDA 领域应用面临的关键挑战。领域数据的稀缺性和多模态特性限制了模型的专业能力,需要通过构建高质量语料库、发展数据增强技术和设计领域特定的模型架构来提升性能。工程实践中的系统集成障碍源于 EDA 工具链的异构性和封闭性,要求开发标准化接口协议和智能代理系统,实现从“工具驱动”向“任务驱动”的范式转变。安全性与可靠性方面,LLM 生成内容的不确定性和潜在的知识产权风险要求建立完善的验证机制、隐私保护方案和伦理规范体系。

## 5 结束语

本文对大语言模型在电子设计自动化领域的应用进行了综述。从 RTL 代码生成与优化、高层次综合与逻辑优化、后端实现验证与安全、模拟电路设计自动化,到 EDA 工具流程自动化与辅助等 5 个应用方向,LLM 技术展现出为传统 EDA 方法带来智能化变革的潜力。当前研究通过检索增强生成、多模态融合、多智能体协作等技术手段,在代码生成、设计优化、验证辅助等环节取得了初步进展。然而,LLM 在 EDA 领域仍面临领域知识获取、多模态信息处理、工业级精度保障、工具链集成以及计算资源限制等挑战。实现工业应用需要在领域专用化模型架构、多模态信息融合、系统集成自动化以及安全可靠保障等方向取得突破,并通过学术界与产业界协作建立开放数据集、基准测试和技术标准。展望未来,LLM 有望在降低设计门槛、提升设计效率方面发挥重要作用,为应对摩尔定律放缓背景下日益复杂的集成电路设计挑战提供新的技术路径。

### 参考文献

- [1] Pan Jingyu, Zhou Guanglei, Chang C C, et al. A survey of

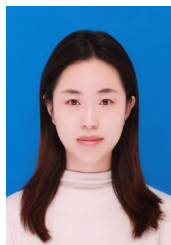
- research in large language models for electronic design automation[J]. *ACM Transactions on Design Automation of Electronic Systems*, 2025, 30(3): 34.
- [2] Touvron H, Lavril T, Izacard G, et al. LLaMA: Open and efficient foundation language models[PP/OL]. V1. arXiv (2023-02-27) [2025-10-15]. <https://arXiv.org/abs/2302.13971>.
- [3] Kahng A B. Machine learning applications in physical design: Recent results and directions[C]//*Proceedings of the 2018 International Symposium on Physical Design*. Monterey: ACM, 2018: 68-73.
- [4] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//*Proceedings of the 31st International Conference on Neural Information Processing Systems*. Long Beach: Curran Associates Inc., 2017: 6000-6010.
- [5] Zhao W X, Zhou Kun, Li Junyi, et al. A survey of large language models[PP/OL]. V18. arXiv (2026-03-08) [2025-10-15]. <https://arXiv.org/abs/2303.18223>.
- [6] Yao Xufeng, Wang Yiwen, Li Xing, et al. RTLRewriter: Methodologies for large models aided RTL code optimization[C]//*Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. New York: ACM, 2025: 98.
- [7] Wang L T, Chang Y W, Cheng K T. Electronic design automation: Synthesis, verification, and test[M]. San Francisco: Morgan Kaufmann, 2009.
- [8] Thakur S, Ahmad B, Fan Zhenxing, et al. Benchmarking large language models for automated verilog RTL code generation[C]//2023 Design, Automation & Test in Europe Conference & Exhibition. Antwerp: IEEE, 2023: 1-6.
- [9] Lu Yao, Liu Shang, Zhang Qijun, et al. RTLLM: An open-source benchmark for design RTL generation with large language model[C]//2024 29th Asia and South Pacific Design Automation Conference. Incheon: IEEE, 2024: 722-727.
- [10] Liu Shang, Lu Yao, Fang Wenji, et al. OpenLLM-RTL: Open dataset and benchmark for LLM-aided design RTL generation: Invited paper[C]//*Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. Newark: ACM, 2025: 1-9.
- [11] Thakur S, Ahmad B, Pearce H, et al. VeriGen: A large language model for verilog code generation[J]. *ACM Transactions on Design Automation of Electronic Systems*, 2024, 29(3): 46.
- [12] Chang Kaiyan, Wang Kun, Yang Nan, et al. Data is all you need: Finetuning LLMs for Chip Design via an Automated design-data augmentation framework[C]//*Proceedings of the 61st ACM/IEEE Design Automation Conference*. San Francisco: ACM, 2024: 60.
- [13] Nadimi B, Zheng Hao. A multi-expert large language model architecture for verilog code generation[C]//2024 IEEE LLM Aided Design Workshop. San Jose: IEEE, 2024: 1-5.
- [14] Fang Wenji, Liu Shang, Wang Jing, et al. CircuitFusion: Multimodal circuit representation learning for agile chip design[PP/OL]. V1. arXiv (2025-05-04) [2025-10-15]. <https://arXiv.org/abs/2505.02168>.
- [15] Fang Wenji, Liu Shang, Zhang Hongce, et al. A self-supervised, pre-trained, and cross-stage-aligned circuit encoder provides a foundation for various design tasks[C]//*Proceedings of the 30th Asia and South Pacific Design Automation Conference*. Tokyo: ACM, 2025: 505-512.
- [16] Akyash M, Azar K, Kamali H. RTL++: Graph-enhanced LLM for RTL code generation[PP/OL]. V1. arXiv (2025-05-11) [2025-10-15]. <https://arXiv.org/abs/2505.13479>.
- [17] Sun Wenhao, Li Bing, Zhang G L, et al. Paradigm-based automatic HDL code generation using LLMs[C]//2025 26th International Symposium on Quality Electronic Design. San Francisco: IEEE, 2025: 1-8.
- [18] Akyash M, Azar K, Kamali H. DecoRTL: A run-time decoding framework for RTL code generation with LLMs[PP/OL]. V1. arXiv (2025-07-03) [2025-10-15]. <https://arXiv.org/abs/2507.02226>.
- [19] Yang Yiyao, Teng Fu, Liu Pengju, et al. HaVen: Hallucination-mitigated LLM for verilog code generation aligned with HDL engineers[C]//2025 Design, Automation & Test in Europe Conference. Lyon: IEEE, 2025: 1-7.
- [20] Zhao Yifang, Fu Weimin, Li Shijie, et al. Enhancing LLM performance on hardware design generation task via reinforcement learning[C]//2025 IEEE International Symposium on Circuits and Systems. London: IEEE, 2025: 1-5.
- [21] Tasnia K, Garcia A, Farheen T, et al. VeriOpt: PPA-aware high-quality verilog generation via multi-role LLMs[PP/OL]. V1. arXiv (2025-07-20) [2025-10-15]. <https://arXiv.org/abs/2507.14776>.
- [22] Liu Shang, Fang Wenji, Lu Yao, et al. RTLCoder: Fully open-source and efficient LLM-assisted RTL code generation technique[J]. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2025, 44(4): 1448-1461.
- [23] Xiong Chenwei, Liu Cheng, Li Huawei, et al. HLSPilot: LLM-based high-level synthesis[C]//*Proceedings of the 43rd IEEE/ACM International Conference on Computer-Aided Design*. New York: ACM, 2025: 226.

- [24] Khan M Z S, Mashnoor N, Akyash M, et al. SAGE-HLS: Syntax-aware AST-guided LLM for high-level synthesis code generation[PP/OL]. V1. arXiv (2025-08-05) [2025-10-15]. <https://arXiv.org/abs/2508.03558>.
- [25] Yao X F, Zhao W Q, Sun Q, et al. High-level synthesis directives design optimization via large language model[J]. ACM Transactions on Design Automation of Electronic Systems, 2025, 30(5): 78.
- [26] Wang Jing, Liu Shang, Lu Yao, et al. HLSDebugger: Identification and correction of logic bugs in HLS code with LLM solutions[PP/OL]. V1. arXiv (2025-07-29) [2025-10-15]. <https://arxiv.org/abs/2507.21485>.
- [27] Huang Shan, Li Jinhao, Yu Zhen, et al. LLSM: LLM-enhanced logic synthesis model with EDA-guided CoT prompting, hybrid embedding and AIG-tailored acceleration[C]//Proceedings of the 30th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2025: 974-980.
- [28] Liang Rongjian, Ho C T, Agnesina A, et al. ReLS: Retrieval is efficient knowledge transfer for logic synthesis[C]//2024 ACM/IEEE 6th Symposium on Machine Learning for CAD. Salt Lake City: IEEE, 2024: 1-7.
- [29] Wu Haoyuan, Zheng Haisheng, Pu Yuan, et al. Circuit representation learning with masked gate modeling and verilog-AIG alignment[PP/OL]. V1. arXiv (2025-02-18) [2025-10-15]. <https://arXiv.org/abs/2502.12732>.
- [30] Fang Wenji, Li Wenkai, Liu Shang, et al. NetTAG: A multimodal RTL-and-layout-aligned netlist foundation model via text-attributed graph[PP/OL]. V1. arXiv (2025-04-12) [2025-10-15]. <https://arXiv.org/abs/2504.09260>.
- [31] Tasnia K, Rahman S. OPL4GPT: An application space exploration of optimal programming language for hardware design by LLM[C]//Proceedings of the 30th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2025: 981-987.
- [32] Lu Yichen, Kunal K, Pradipta G, et al. LEGO-size: LLM-enhanced GPU-optimized signoff-accurate differentiable VLSI gate sizing in advanced nodes[C]//Proceedings of the 2025 International Symposium on Physical Design. Austin: ACM, 2025: 152-162.
- [33] Ho C T, Ren Haoxing. Large language model (LLM) for standard cell layout design optimization[C]//2024 IEEE LLM Aided Design Workshop. San Jose: IEEE, 2024: 1-6.
- [34] Yan Zhiyuan, Fang Wenji, Li Mengming, et al. AssertLLM: Generating hardware verification assertions from design specifications via multi-LLMs[C]//Proceedings of the 30th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2025: 614-621.
- [35] Kande R, Pearce H, Tan B, et al. (Security) assertions by large language models[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4374-4389.
- [36] Ahmad B, Thakur S, Tan B, et al. On hardware security bug code fixes by prompting large language models[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 4043-4057.
- [37] Akyash M, Kamali H M. Self-HWDebug: Automation of LLM self-instructing for hardware security verification[C]//2024 IEEE Computer Society Annual Symposium on VLSI. Knoxville: IEEE, 2024: 391-396.
- [38] Fu Weimin, Yang Kaichen, Dutta R G, et al. LLM4 SecHW: Leveraging domain-specific large language model for hardware debugging[C]//2023 Asian Hardware Oriented Security and Trust Symposium. Tianjin: IEEE, 2024: 1-6.
- [39] Mashnoor N, Akyash M, Kamali H, et al. LLM-IFT: LLM-powered information flow tracking for secure hardware[C]//2025 IEEE 43rd VLSI test symposium. Tempe: IEEE, 2025: 1-5.
- [40] Jiang Yuqi, Jin Qian, Lu Xudong, et al. FabGPT: An efficient large multimodal model for complex wafer defect knowledge queries[PP/OL]. V1. arXiv (2024-07-15) [2025-10-15]. <https://arXiv.org/abs/2407.10810>.
- [41] Tao Zhuofu, Shi Yichen, Huo Yiru, et al. AMSNet: Netlist dataset for AMS circuits[C]//2024 IEEE LLM Aided Design Workshop. San Jose: IEEE, 2024: 1-5.
- [42] Chen Zihao, Zhuang Ji, Shen Jinyi, et al. AnalogSeeker: An open-source foundation language model for analog circuit design[PP/OL]. V2. arXiv (2025-11-05) [2025-10-15]. <https://arXiv.org/abs/2508.10409>.
- [43] Skelic L, Xu Yan, Cox M, et al. CIRCUIT: A benchmark for circuit interpretation and reasoning capabilities of LLMs[PP/OL]. V1. arXiv (2025-02-11) [2025-10-15]. <https://arXiv.org/abs/2502.07980>.
- [44] Shi Yichen, Zhang Ze, Wang Hongyang, et al. AMSbench: A comprehensive benchmark for evaluating MLLM capabilities in AMS circuits[PP/OL]. V2. arXiv (2025-10-13) [2025-10-15]. <https://arXiv.org/abs/2505.24138>.
- [45] Chen Zihao, Huang Jiangli, Liu Yiting, et al. Artisan: Automated operational amplifier design via domain-specific large language model[C]//Proceedings of the 61st ACM/IEEE Design Automation Conference. San Francisco: ACM,

- 2024: 39.
- [46] Vijayaraghavan P, Shi Luyao, Degan E, et al. AUTOCIR-CUIT-RL: Reinforcement learning-driven LLM for automated circuit topology generation[PP/OL]. V1. arXiv (2025-06-03) [2025-10-15]. <https://arXiv.org/abs/2506.03122>.
- [47] Chang C C, Lin W H, Shen Yikang, et al. LaMAGIC2: Advanced circuit formulations for language model-based analog topology generation[PP/OL]. V2. arXiv (2025-07-26) [2025-10-15]. <https://arXiv.org/abs/2506.10235>.
- [48] Zhang Haoyi, Sun Shizhao, Lin Yibo, et al. AnalogXpert: Automating analog topology synthesis by incorporating circuit design expertise into large language models[C]//2025 International Symposium of Electronics Design Automation. Hong Kong, China: IEEE, 2025: 772-777.
- [49] Liu Chang, Olowe E A, Chitnis D. LLM-based AI agent for sizing of analog and mixed signal circuit[C]//2025 23rd IEEE Interregional NEWCAS Conference. Paris: IEEE, 2025: 90-94.
- [50] Gao Jian, Cao Weidong, Yang Junyi, et al. AnalogGenie: A generative engine for automatic discovery of analog circuit topologies[PP/OL]. V1. arXiv (2025-02-28) [2025-10-15]. <https://arXiv.org/abs/2503.00205>.
- [51] Gao Jian, Cao Weidong, Zhang Xuan. AnalogGenie-Lite: Enhancing scalability and precision in circuit topology discovery through lightweight graph modeling[C]//Proceedings of the 42nd International Conference on Machine Learning. Vancouver, 2025: 18277-18289.
- [52] Bhandari J, Bhat V, He Yuheng, et al. Masala-CHAI: A large-scale SPICE netlist dataset for analog circuits by harnessing AI[PP/OL]. V5. arXiv (2025-03-23) [2025-10-15]. <https://arXiv.org/abs/2411.14299>.
- [53] Lai Yao, Lee S, Chen Guojin, et al. AnalogCoder: Analog circuit design via training-free code generation[C]//Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia: AAAI, 2025: 379-387.
- [54] Lai Yao, Poddar S, Lee S, et al. AnalogCoder-pro: Unifying analog circuit generation and optimization via multi-modal LLMs[PP/OL]. V2. arXiv (2025-08-31) [2025-10-15]. <https://arXiv.org/abs/2508.02518>.
- [55] Liu Bingyang, Zhang Haoyi, Gao Xiaohan, et al. Layout-Copilot: An LLM-powered multiagent collaborative framework for interactive analog layout design[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(8): 3126-3139.
- [56] Poddar S, Oh Y, Lai Yao, et al. INSIGHT: Universal neural simulator for analog circuits harnessing autoregressive transformers[PP/OL]. V3. arXiv (2024-08-06) [2025-10-15]. <https://arXiv.org/abs/2407.07346>.
- [57] Gao Xiaohan, Zhang Haoyi, Ye Siyuan, et al. Post-layout simulation driven analog circuit sizing[J]. Science China Information Sciences, 2024, 67(4): 142401.
- [58] Chen Hao, Liu Mingjie, Xu Biying, et al. MAGICAL: An open-source fully automated analog IC layout system from netlist to GDSII[J]. IEEE Design & Test, 2021, 38(2): 19-26.
- [59] Dhar T, Kunal K, Li Yaguang, et al. The ALIGN open-source analog layout generator: V1.0 and beyond[C]//Proceedings of the 39th International Conference on Computer-Aided Design. Virtual Event: ACM, 2020: 54.
- [60] Xiao Zhe, He Xu, Wu Haoying, et al. EDA-copilot: A RAG-powered intelligent assistant for EDA tools[J]. ACM Transactions on Design Automation of Electronic Systems, 2025, 30(6): 99.
- [61] Pu Yuan, He Zhuolun, Jiang Yuqi, et al. Customized retrieval augmented generation and benchmarking for EDA tool documentation QA[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(12): 4615-4628.
- [62] Abbineni P, Aldowais S, Liechty C, et al. MuaLLM: A multimodal large language model agent for circuit design assistance with hybrid contextual retrieval-augmented generation[PP/OL]. V1. arXiv (2025-08-11) [2025-10-15]. <https://arXiv.org/abs/2508.08137>.
- [63] He Zhuolun, Wu Haoyuan, Zhang Xinyun, et al. ChatEDA: A large language model powered autonomous agent for EDA[C]//2023 ACM/IEEE 5th Workshop on Machine Learning for CAD. Snowbird: IEEE, 2023: 1-6.
- [64] Wu Haoyuan, Zheng Haisheng, He Zhuolun, et al. Divergent thoughts toward one goal: LLM-based multi-agent collaboration system for electronic design automation[PP/OL]. V1. arXiv (2025-02-15) [2025-10-15]. <https://arXiv.org/abs/2502.10857>.
- [65] Pan Jingyu, Jacobson I, Zhao Zheng, et al. CROP: Circuit retrieval and optimization with parameter guidance using LLMs[PP/OL]. V2. arXiv (2025-08-22) [2025-10-15]. <https://arXiv.org/abs/2507.02128>.
- [66] Chen Guojin, Yang Haoyu, Yu Bei, et al. Intelligent OPC engineer assistant for semiconductor manufacturing[C]//Proceedings of the 39th AAAI Conference on Artificial Intelligence. Philadelphia: AAAI, 2025: 23144-23151.
- [67] Li Junyan, Wong S Z, Wan G W, et al. EDA-debugger: An LLM-based framework for automated EDA runtime issue resolution[C]//2025 26th International Symposium on

- Quality Electronic Design. San Francisco: IEEE, 2025: 1-7.
- [68] Zhao Chenchun, Shi Zhengyuan, Wen Xiangyu, et al. MMCircuitEval: A comprehensive multimodal circuit-focused benchmark for evaluating LLMs[PP/OL]. V1. arXiv (2025-07-20) [2025-10-15]. <https://arXiv.org/abs/2507.19525>.
- [69] Shi Zhengyuan, Li Zeju, Ma Chengyu, et al. ForgeEDA: A comprehensive multimodal dataset for advancing EDA[PP/OL]. V1. arXiv (2025-05-04) [2025-10-15]. <https://arXiv.org/abs/2505.02016>.
- [70] Liu Shang, Fang Wenji, Lu Yao, et al. Towards big data in AI for EDA research: Generation of new pseudo circuits at RTL stage[C]//Proceedings of the 30th Asia and South Pacific Design Automation Conference. Tokyo: ACM, 2025: 527-533.
- [71] Fang W J, Lu Y, Liu S, et al. Transferable presynthesis PPA estimation for RTL designs with data augmentation techniques[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025, 44(1): 200-213.

## 作者简介



**王 静** 女,1997年11月出生于陕西省榆林市。博士毕业于复旦大学集成电路与微纳创新学院。现为南京邮电大学集成电路科学与工程学院讲师。主要研究方向为模拟集成电路设计及人工智能在设计自动化中的应用。中国电子学会会员编号:E190181919M。  
E-mail: wangjing25@njupt.edu.cn



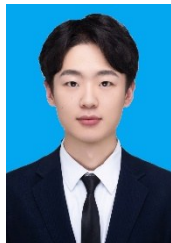
**林立宇** 男,1995年11月出生于广东省揭阳市。现为复旦大学集成电路与微纳创新学院博士后。主要研究方向为数字集成电路设计。  
E-mail: lylin@fudan.edu.cn



**李 磊** 男,1998年7月出生于江西省上饶市。现为香港大学计算与数据科学学院博士研究生。主要研究方向为多模态大模型。中国电子学会会员编号:E190160089A。  
E-mail: lilei98@connect.hku.hk



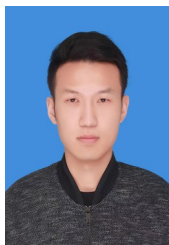
**陈霖谊** 女,2001年12月出生于四川省成都市。现为南京邮电大学集成电路科学与工程学院硕士研究生。主要研究方向为模拟集成电路设计。  
E-mail: 1225228011@njupt.edu.cn



**李 征** 男,2001年4月出生于山东省临沂市。现为复旦大学集成电路与微纳电子创新学院硕士研究生。主要研究方向为近似计算与AI加速器设计。  
E-mail: 24212020117@m.fudan.edu.cn



**来 耀** 男,1994年9月出生于安徽省芜湖市。博士毕业于香港大学计算与数据科学学院。现为剑桥大学博士后。主要研究方向为人工智能在设计自动化中的应用。  
E-mail: yl2204@cam.ac.uk



**李 炎** 男,1993年8月出生于江苏省无锡市。现为复旦大学集成电路与微纳创新学院青年研究员、博士生导师。主要研究方向为高可靠(抗辐射)/高性能(低温77 K)集成电路设计与方法学研究,在该领域已经发表论文20余篇。中国电子学会会员编号:E190029996S。  
E-mail: liyan@fudan.edu.cn



**姚佳飞** 男,1988年2月出生于江苏省常熟市。现为南京邮电大学副教授、博士生导师。主要研究方向为功率器件及集成封装技术。发表学术论文100余篇,授权中国发明专利60余项。中国电子学会会员编号:E190016854S。  
E-mail: jfyao@njupt.edu.cn



**曾晓洋** 男,1972年出生于湖南省衡阳市。现为复旦大学特聘教授、科学技术研究院副院长。主要研究方向为高能系统芯片研究与开发。在该领域已经发表论文300余篇,申请发明专利100余项。中国电子学会会员编号:E190005669F。  
E-mail: xyzeng@fudan.edu.cn



**郭宇锋** 男,1974年4月出生于河南省洛阳市。现为南京邮电大学教授、博士生导师。主要研究方向为功率器件及其集成技术、宽禁带半导体技术、射频集成电路设计和智能EDA技术。发表学术论文400余篇,授权中国发明专利120余项。中国电子学会会员编号:E190004403F。  
E-mail: yfguo@njup.edu.cn