arXiv:2601.13815v1 [cs.AR] 20 Jan 2026

从寄存器传输级到提示词编程：借助大语言模型赋能新一代芯片设计师

Lukas Krupp*、Matthew Venn† 和 Norbert Wehn*
*凯撒斯劳滕-兰道大学，德国凯撒斯劳滕
†Tiny Tapeout

摘要——本文介绍一种用于芯片设计教育的基于大语言模型的学习平台，旨在让初学者也能轻松接触芯片设计，而不被技术复杂性所困扰。这是首个在前端与后端设计两方面为学习者提供全方位辅助的教育平台。该方法将基于大语言模型的聊天代理集成到基于浏览器的工作流程中，该流程构建于 TinyTapeout 生态系统之上。工作流程引导用户从最初的设计构想出发，经过 RTL 代码生成，最终完成可流片芯片。为评估这一概念，研究人员与 18 名高中生开展了一项案例研究。在一堂 90分钟课程中，他们采用 130 纳米工艺开发了八款功能完备的 VGA 芯片设计。尽管此前没有芯片设计经验，所有小组都成功完成了可流片项目。研究结果证明了大语言模型辅助芯片设计的可行性及其教育影响，凸显了其吸引并启发初学者的潜力，并能显著拓宽这一领域的目标受众。

索引词—**芯片设计教育、大语言模型、智能体系统、RTL设计、Tiny Tapeout**

I. 引言

熟练芯片设计师短缺已成为一个亟待解决的问题，在欧洲尤为突出，在美国和中国等国家也同样如此。近期报告指出，行业需求与现有劳动力之间的差距正在不断扩大 [1], [2]。令人担忧的是，学生对微电子学的兴趣日益下降；若不采取行动，设计师短缺问题将进一步恶化 [3]。另一方面，过去几年间，人们对软件工程和人工智能（AI）的热情持续高涨。

这种分化的一个主要原因在于这两个领域的生产力发展动态不同。软件开发和人工智能受益于高度抽象以及极低的基础设施要求。相比之下，芯片设计仍是一门以效率为导向的学科，面积、功耗和时序等参数至关重要。因此，需要在所有设计层面进行优化，这阻碍了高层次抽象的实现。与软件不同，芯片设计受物理定律和芯片实现复杂性的制约。

因此，硬件领域的生产力提升在很大程度上落后于软件领域。例如，数字设计仍主要依赖使用 Verilog 和 VHDL 等硬件描述语言（HDL）进行寄存器传输级编码。HDL 能够对关键的物理实现参数进行细粒度控制，但会使 RTL 设计过程耗时且容易出错。此外，从系统级规范到物理布局的数字设计流程，需要一套专用工具，以及在设计方法学、电路和工艺方面的深厚专业知识。

软件开发提供了一条“快速成功之路”，凭借即时反馈吸引学习者；而芯片设计则让新手面临陡峭的学习曲线，令人望而却步。设计复杂性、对专用基础设施的需求，以及相应的“漫长成功之路”，都使吸引学习者并保持其学习动力变得困难。为了增加未来芯片设计师的数量，必须尽早激发好奇心，理想情况下，从学校阶段就开始 [4]。要吸引广大的非专业人士，芯片设计必须兼具易用性与吸引力。

Tiny Tapeout 倡议 [5]展现了降低入门门槛的潜力，使人们只需极少的基础设施即可进行芯片设计。只需一台电脑和互联网连接即可开始。其关键在于采用开源电子设计自动化（EDA）工具和工艺开发套件（PDK），这些工具和套件可提供不受限制的访问，并支持后端流程的云端部署。结合低成本制造，这些进展有助于应对芯片实现与制造方面的挑战。然而，前端设计知识，尤其是数字设计中的 RTL 编码，仍不可或缺，也继续构成初学者面临的一道门槛。迄今为止，还没有教育平台能够在前端与后端设计的各个环节为学习者提供全面支持，引导他们从构想到可流片芯片。

因此，本文旨在探索如何将大语言模型（LLM）应用于芯片设计教育，降低非专业人士学习 RTL 设计的门槛，同时仍使他们能够实现切实可行、可流片的设计。本文的主要贡献如下：

1）开发了一款基于大语言模型的聊天代理，旨在帮助没有或仅有少量 RTL 编码基础知识的学习者。2）提出了一种方法和基于浏览器的环境，借助聊天代理和开源 Tiny Tapeout 基础设施，支持在 130 纳米工艺下开发视频图形阵列（VGA）芯片设计，涵盖从构想到 GDSII 的全过程。3）通过一项针对 18 名高中生的案例研究，验证了该方法并展示其可行性。

案例研究中的设计、聊天代理访问权限以及实践工作流程指令均通过 GitHub[1] 提供。

本文由德国联邦研究、技术与航天部（BMFTR）资助，属于“Chipdesign Germany”项目，资助编号为 16ME0890。

[1]设计成果与工作流程演示：https://github.com/tukl-msd/prompt2silicon

本文已提交至 IEEE E，以供考虑发表。版权可能在不另行通知的情况下转让，届时此版本可能不再易于使用。

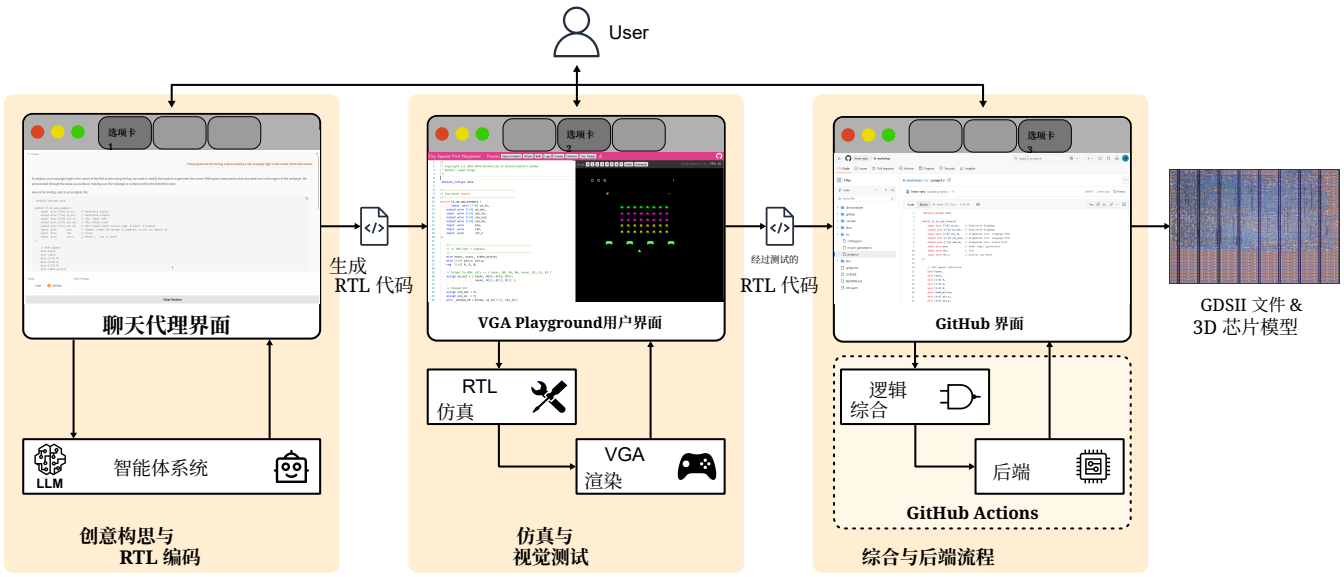

图1. 所提出的从构想到 GDSII学习工作流概览：集成了用于RTL实现的基于大语言模型的聊天代理、VGA仿真工具 ，
以及 GitHub驱动的后端流程。该 bro 图中展示了基于浏览器的用户界面（UI）和底层应用程序。

II. 背景与相关工作

在芯片设计教育中，我们区分三类目标群体。第一类是具有实践经验的高级学习者，例如博士生或硕士生，他们所占比例最小。第二类包括刚开始接触或对此感兴趣的学习者，他们出于内在动机，希望探索芯片设计基础。人数最多的是非专业受众，包括中小学生、职业教育学生、本科低年级学生或业余爱好者；他们可能听说过芯片设计，但此前从未接触过。

Tiny Tapeout计划大幅降低了参与定制芯片设计的门槛，尤其是对感兴趣的学习者和非专业人士而言。Tiny Tapeout将多项目晶圆这一成熟概念扩展至多项目裸片，使数百种设计能够共享单个芯片 [6]。多路复用器阵列允许用户一次选择一种设计，并将其连接到输入输出引脚。自2022年推出以来，Tiny Tapeout已利用 SkyWater [7] 和IHP [8] 提供的开源工艺设计套件，在130纳米技术节点上推动制造了2000多种设计。

此外，Tiny Tapeout 还提供了丰富的开源生态系统。该生态系统的核心是面向数字 [9] 和模拟设计 [10]的 GitHub 仓库模板。这些模板包含自动化综合和后端流程，通过 GitHub Actions 进行编排，并完全基于开源 EDA 工具。对于初学者，Wokwi [11] 和 SiliWiz [12]等图形化工具提供了设计和学习选项。其他资源还包括 VGA Playground [13] 和 GDS Viewer [14],等可视化工具，以及教程和视频。流片后，用户会收到制造完成的芯片，并可在评估板上使用。系统软件支持评估板开箱即用，并可与设计进行交互。

尽管进行了这些简化，数字设计仍然需要扎实的RTL编程知识，这对初学者构成了挑战。近期研究表明，大语言模型支持EDA工作流的潜力 [15], ，并在RTL代码生成、调试、优化、验证和理解方面展现出令人鼓舞的成果 [16]–[21]。然而，现有工作主要聚焦于面向业界的应用场景，即利用大语言模型提升经验丰富的设计人员的生产力。相比之下，大语言模型在编程 [22], [23] 和STEM学科[24], [25], 中的教育应用已得到广泛采用，在这些领域中，它们充当导师和学习加速器。

然而，迄今为止，在芯片设计领域，还没有推出基于大语言模型的教育平台，能够覆盖从构想到可流片芯片的完整流程。

III. 方法

A. 系统概述与学习工作流

如图 1 所示，人在回路设计流程使用户能够通过网页浏览器与系统交互，并从初步构想逐步推进到可流片芯片。工作流始于构想阶段，用户在基于 VGA 的设计范畴内定义希望实现的概念。例如，可以从复古游戏中汲取灵感，如《太空侵略者》或《俄罗斯方块》；这些游戏至今仍广受欢迎，尤其受到年轻群体的喜爱。利用输入引脚，设计可以涵盖从动画到交互式控制元件或游戏等多种形式。在此阶段，提示工程发挥着核心作用。用户必须足够精确地描述预期行为，以确保生成的 RTL 代码符合他们的预期。

接下来，用户通过聊天智能体用户界面（UI）与基于大语言模型的智能体系统交互。该界面是将设计构想转化为硬件描述的入口。

用户基于构思阶段形成的概念和行为规范提供提示词，以生成 RTL 代码。第三-B节将详细介绍底层智能体系统。

工作流程的第二阶段是仿真和视觉测试。为此，使用 VGA Playground 用户界面 [13] 。该应用结合基于开源工具 *Verilator* 的 RTL 仿真与 VGA渲染引擎，以生成交互式视觉输出。用户将 RTL 代码粘贴到工具中，随后工具会实时仿真设计。即时的视觉反馈使用户无需编写复杂的测试平台，即可验证正确性并调试其设计。用户可以返回聊天代理，描述预期行为与实际观察到的行为之间的差异，逐步优化提示词，并重新生成 RTL 代码。

工作流程的最后阶段是芯片实现。该步骤基于 GitHub 网页用户界面和 Tiny Tapeout Verilog模板 [9]。用户将其 RTL 代码插入指定文件，推送更改后，GitHub Actions 将启动逻辑综合和后端流程。借助开源工具 *Yosys* 和 *LibreLane*，该步骤最终会生成一个 GDSII 文件。生成的版图可以使用 Tiny Tapeout GDS查看器 [14]进行检查。

B. 基于大语言模型的聊天智能体系统

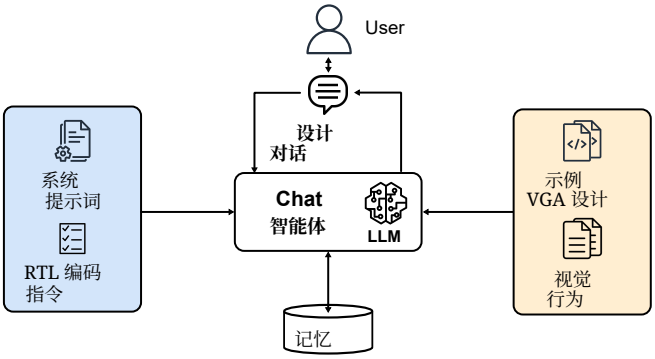

图2. 基于大语言模型的聊天智能体系统概览。

图2展示了智能体系统的详细配置。用户通过上一节介绍的 UI，以对话方式与聊天代理交互。该系统采用基于大语言模型的单智能体配置。这一设计选择反映了任务的聚焦性：任务仅涉及 Tiny Tapeout VGA 设计。相比之下，当搜索空间广阔且需求模糊时，多智能体系统通常更具优势。系统记忆会保存对话历史、中间设计结果和上下文信息，以确保多轮设计迭代的一致性。

为指导智能体的运行，我们提供了详细的系统提示词和 RTL 编码指令。系统提示词定义了智能体的角色和限制，并介绍了用于 VGA 芯片设计的 Verilog RTL 编码背景，描述了任务的教育导向，并说明用户属于初学者水平。此外，它还要求智能体用通俗易懂的语言解释概念，并生成带有大量注释的 RTL 代码。

RTL 编码指令为智能体提供了规范的工作流程，使其生成的设计符合 VGA Playground 和 Tiny Tapeout后端流程的要求。这些指令提供逐步的编码指导，并强制采用满足接口要求的固定模块和信号声明。Tiny Tapeout VGA 设计的核心在于实例化开源 VGA 时序控制器知识产权（IP）核。这些指令确保声明必要的连线，并完成信号路由和输出引脚映射。智能体需在此默认模块模板基础上，根据用户指定的行为进行扩展，并插入所需的控制逻辑和数据通路逻辑。

此外，通过示例 VGA 设计及其对应视觉行为的文本描述配对，将领域特定知识整合到智能体系统中。这些输入（描述）与预期输出（RTL 代码）配对内容可作为智能体系统的参考。示例是 VGA Playground 中提供的默认设计。

IV. 案例研究：高中生评估

本案例研究旨在评估所提出的大语言模型辅助设计方法在真实教育环境中的可行性。我们旨在评估非专业人士能否在有限时间内完成整个设计流程。主要评估指标为成功率，包括：（1）功能正确性，即通过行为仿真；（2）流片就绪度，即通过 Tiny Tapeout 后端流程成功生成 GDSII。此外，本案例研究还应深入了解在时间和经验限制下可实现的设计规模与复杂度。

A. 实验设置

本案例研究的参与者为 18 名来自德国高中的学生，年级从 10 年级到 12 年级不等。参与者被分为八组，每组两到三人。研究总时长为 90 分钟。前 15 分钟用于介绍芯片制造流程和芯片设计流程。剩余 75 分钟用于动手设计，采用图 1 所示的预配置工作流。

B. 结果与观察

在开始设计阶段之前，我们开展了一项调查，以评估学生的先前经验。调查内容包括参与者是否了解芯片设计，以及他们之前是否接触过电子学（例如，分立元件、焊接、自己动手制作项目）或编程。调查结果汇总于图3。

三名学生（17%）听说过芯片设计，四名学生（22%）曾接触过电子学，同样数量的学生（22%）曾接触过编程。学生提及的编程语言包括 Python、Java 和 HTML。

表I 案例研究期间创建的学生设计概览。

组	设计描述	类别	SLOC	图块	创造力	复杂度	功能正确?	可流片?
1	在街道上行驶的蓝色汽车	交互式动画	84	1×1	中等	中等	Yes	Yes
2	有游动的鱼的水族箱		89	1×1	High	中等	Yes	Yes
3	像素艺术猫咪形象	静态精灵	66	1×1	中等	Low	Yes	Yes
4	蓝色方块	静态精灵	41	1×1	Low	Low	Yes	Yes
5	跳跃的火柴人	交互式动画	76	2×2	High	中等	Yes	Yes
6	在街道上行驶的红色汽车		59	1×1	中等	中等	Yes	Yes
7	独角兽接住一根胡萝卜		126	1×1	High	High	Yes	Yes
8	飘落着树叶的树		97	1×2	High	High	Yes	Yes

结果表明，大多数学生在参与案例研究前，并未明显接触过硬件或编程。

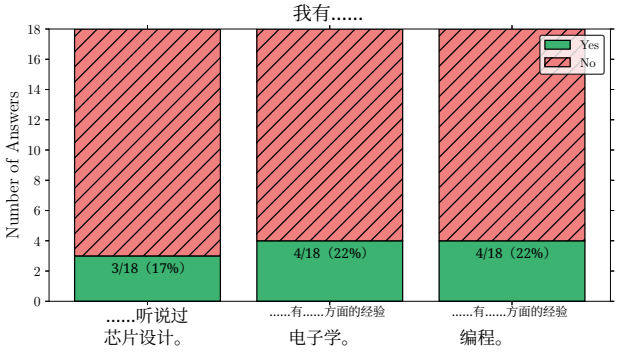

图3. 学生背景知识调查结果。

表I总结了学生项目的成果。设计分为三类：静态精灵图显示、非交互式动画，以及通过输入引脚控制的交互元素。为估算项目规模，我们报告了每项设计的有效源代码行数（SLOC），不包括注释和空行。此外，所需硅片面积以图块数量表示，图块是 Tiny Tapeout 多项目裸片中最小的集成单元。每个图块的尺寸约为$160 \times 100\,\mu$米。两位外部专家根据设计构想和 RTL实现评估了创造力和复杂度。最后，表中还注明了 VGA仿真和芯片实现是否成功。

八个学生小组完成了各种各样的设计。重要的是，所有设计均成功通过了 VGA仿真和 Tiny Tapeout后端流程。两个小组实现了静态精灵图显示，这是最简单的输出形式，源代码行数介于 41 至 66 行之间。三个小组开发了没有用户交互的动画，源代码长度从 89 行到 126 行不等。其余三个项目实现了交互元素，例如可控汽车或跳跃的火柴人，代码长度介于 59 至 84 行之间。在硬件使用方面，大多数设计都能容纳在单个图块内。较为复杂的交互式火柴人总共需要四个图块，而落树动画则占用了两个图块。创造力和复杂度差异显著。静态精灵的评级较低至中等，而动画和交互式应用的评级则达到中等至较高水平。这些结果表明，在较短时间内完成切实可行且可流片的项目是可行的。

除了表I总结的定量结果外，案例研究还揭示了若干定性见解。学生能够借助插入的注释理解生成的 RTL 代码。具备编程经验的学生则利用这种方法，验证生成的代码是否符合他们预期的设计行为。调试过程中自然形成了基本的调试策略；事实证明，这种迭代式、对话式设计流程能够有效帮助学生逐步构建和完善项目。例如，汽车项目从一个简单的方块逐步发展为可控车辆，而猫形图案则通过逐一添加视觉特征而不断完善。

对于更复杂的设计，挑战也随之而来。树木动画虽然始终通过仿真，但包含不可综合的 Verilog 代码，因此需要借助聊天代理，经过多轮后端错误报告和修正。同样，火柴人和树木设计的面积都超过了单个 tile 的容量。根据后端流程报告调整 Tiny Tapeout配置文件即可解决此问题。然而，仅靠聊天代理不足以完成这项任务，教师专业知识至关重要。这些经验既凸显了对话式方法的有效性，也揭示了其在处理高级后端问题方面的当前局限。

V. 结论与未来工作

本文探讨了如何利用大语言模型，让初学者更容易接触芯片设计。我们介绍了一种基于大语言模型的聊天代理，并将其集成到基于 Tiny Tapeout 生态系统构建的基于浏览器的学习工作流中。借助这种方法，用户可以在短时间内从最初的设计构想一路推进到RTL代码生成，最终完成可流片芯片。对18名高中生开展的案例研究表明，没有相关经验的参与者也能够实现功能性设计。在一节90分钟课程中，八个学生小组均成功使用130纳米工艺开发出VGA芯片。最终设计展现了该方法的创造潜力和技术可行性。Tiny Tapeout已在三年内支持了超过2,000项设计，而我们的方法进一步扩大了这一覆盖面，使更多非专业人士能够轻松进入芯片设计领域。未来的工作将着重于把聊天代理和工作流拓展至其他受众，并进一步分析用户的学习曲线。

参考文献

[1] L. Saint-Martin、O. Coulon 和 A. Barbosa，《2023年度监测报告》，微电子培训、产业与技能（METIS），2023年。访问日期：2025年10月2日。 [在线]。获取地址：https://www.metis4skills.eu/wp-content/uploads/2023/12/Yearly-Monitoring-Report-2023.pdf [2] M. Guthaus、C. Batten、E. Brunvand、P.E. Gaillardon、D. Harris、R. Manohar、P. Mazumder、L. Pileggi 和 J. Stine，《NSF集成电路研究、教育与人才队伍发展研讨会最终报告》，美国国家科学基金会，2023年11月3日。获取地址：https://arxiv.org/abs/2311.02055 。[3] R. Beaujeu、L. Saint-Martin 和 C. Lebon，《2024年技能战略——应对欧盟半导体生态系统的人才缺口》，欧洲芯片技能学院，2024年。访问日期：2025年10月2日。 [在线]。获取地址：https://chipsacademy.eu/news/ecsa-skills-strategy-2024/ [4] L. Krupp、I. O’Connor、L. Benini、C. Studer、J. Rodrigues、N. Wehn，《改善欧洲大学的芯片设计赋能——立场文件》，2025年欧洲设计、自动化与测试会议（DATE），2025，2508.14907 获取地址：https://arxiv.org/abs/ 。[5] Tiny Tapeout。《Tiny Tapeout》。访问日期：2025年10月2日。 [在线]。获取地址：https://tinytapeout.com/ [6] M. Venn，《Tiny Tapeout：人人皆可使用的共享硅芯片流片平台》，《IEEE固态电路杂志》，第16卷第2期，2024年，第20–29页。[7] 谷歌，《SkyWater-PDK：适用于 SkyWater 130 纳米工艺的开源工艺设计套件》，2025年。访问日期：2025年10月6日。 [在线]。获取地址：https://github.com/google/skywater-pdk [8] IHP GmbH，《IHP-Open-PDK：适用于 IHP 硅锗双极型互补金属氧化物半导体工艺的开源工艺设计套件》，2025年。访问日期：2025年10月6日。 [在线]。获取地址：https://github.com/IHP-GmbH/IHP-Open-PDK [9] Tiny Tapeout，《ttsky-verilog-template：Tiny Tapeout SKY130（ChipFoundry）穿梭项目的提交模板——Verilog硬件描述语言项目》，2025年。访问日期：2025年10月2日。 [在线]。获取地址：https://github.com/ TinyTapeout/ttsky-verilog-template [10] Tiny Tapeout，《ttsky-analog-template：Tiny Tapeout SKY130（ChipFoundry）穿梭项目的提交模板——模拟、混合信号和定制GDS项目》，2025年。访问日期：2025年10月2日。 [在线]。获取地址：https://github.com/TinyTapeout/ttsky-analog-template [11] Tiny Tapeout，《数字设计工具（Wokwi）入门》，2025年。访问日期：2025年10月2日。 [在线]。获取地址：https://tinytapeout. com/digital design/wokwi/ [12] Tiny Tapeout，《SiliWiz简介：更快捷、更轻松、更经济地制作自己的芯片》，2025年。访问日期：2025年10月2日。 [在线]。获取地址：https://tinytapeout.com/siliwiz/introduction/ [13] Tiny Tapeout，《VGA Playground》，2025年。访问日期：2025年10月2日。[在线]。获取地址：https://vga-playground.com/ [14] Tiny Tapeout，《GDS Viewer》，2025年。访问日期：2025年10月2日。 [在线]。获取地址：https://gds-viewer.tinytapeout.com/ [15] C.-T. Ho、J. Gong、Y. Bai、C. Deng、H. Ren 和 B. Khailany，《Marco：面向硬件设计的可配置基于图的任务求解与多AI智能体框架》，2025年。获取地址：https://arxiv.org/abs/2504. 01962 [16] N. Pinckney、C. Deng、C.-T. Ho、Y.-D. Tsai、M. Liu、W. Zhou、B. Khailany 和 H. Ren，《综合Verilog设计问题：用于评估大语言模型和智能体在RTL设计与验证方面表现的新一代基准数据集》，2025年。获取地址：https://arxiv.org/abs/2506.14074 。[17] C.-T. Ho、H. Ren 和 B. Khailany，《VerilogCoder：具备基于图的规划和基于抽象语法树（AST）的波形跟踪工具的自主Verilog编程智能体》，2024年。获取地址：https://arxiv. org/abs/2408.08927 。[18] H. Huang、Z. Lin、Z. Wang、X. Chen、K. Ding 和 J. Zhao，《迈向大语言模型驱动的Verilog RTL助手：自我验证与自我纠错》，2024，2406.00115 。获取地址：https://arxiv.org/abs/ 。[19] Y. Zhao、H. Zhang、H. Huang、Z. Yu 和 J. Zhao，《MAGE：用于自动化RTL代码生成的多智能体引擎》，2024年。获取地址：https://arxiv.org/abs/2412.07822 。[20] X. Yao、Y. Wang、X. Li、Y. Lian、R. Chen、L. Chen、M. Yuan、H. Xu 和 B. Yu，《RTLRewriter：大模型辅助RTL代码优化方法》，2024，2409.11414 。获取地址：https://arxiv.org/abs/ 。[21] B. Wang、Q. Xiong、Z. Xiang、L. Wang 和 R. Chen，《RTLSquad：基于多智能体的可解释RTL设计》，2025年。获取地址：https: // arxiv.org/abs/2501.05470 [22] J. Zhao, P. Gao, J. Cao, Z. Wen, C. Chen, J. Yin, R. Yang, B. Yuan，《CodeEdu：用于个性化编程教育的多智能体协作平台》，2025，2507.13814 。可查阅：https://arxiv.org/abs/。[23] Q. Ma, H. Shen, K. Koedinger, S. T. Wu，《如何在人工智能时代教授编程？将大语言模型用作调试的可教智能体》，载于教育中的人工智能，施普林格·自然瑞士，2024年，第265–279页。可查阅：https://arxiv.org/abs/2310.05292。[24] F. Neha, D. Bhati，《DeepSeek模型在STEM教育中的应用：能力、应用与挑战》，2025，[25] S. Due, S. Das, M. Andersen, B. P. L´opez, S. A. Nexø, L. Clemmensen，《大语言模型评估：STEM教育与性别刻板印象》，2024年。可查阅：https://arxiv.org/abs/2406.10133