

Проектирование сверхбольших интегральных схем с использованием больших языковых моделей**Колонин А. Г.¹**

¹ Новосибирский Государственный Университет
630090, Россия, г. Новосибирск, ул. Пирогова, 1
anton.kolonin@yandex.ru

Аннотация

Показаны варианты использования больших языковых моделей для проектирования сверхбольших интегральных схем, включая генерацию кода Verilog/VHDL и тестов по спецификациям, а также верификацию кода. Представлены результаты базовой апробации с использованием набора больших языковых моделей, предложены вычислительные архитектуры ансаблированной конкурентной генерации и верификации с арбитражом, указаны возможные риски и проблемы, требующие решения.

Ключевые слова: LLM; БЯМ; VLSI; СБИС; Verilog; VHDL; большая языковая модель; проектирование сверхбольших интегральных схем.

На сегодняшний день важнейшим этапом проектирования сверхбольших интегральных схем (СБИС, very large scale integration, VLSI) является логическое проектирование, когда на основе текстовых спецификаций создается программное описание логики на языках Verilog или VHDL. Также, на этом этапе может осуществляться дополнительная верификация полученных кодов, со сверкой их на предмет соответствия исходным спецификациям и отсутствия не предусмотренной спецификациями логики, что особо важно для обеспечения «доверенности» создаваемой аппаратной базы. Кроме того, на основе спецификаций могут создаваться тесты для программного тестирования верифицируемого кода СБИС. Как показано в работах [1,2,3] на этих этапах могут быть использованы современные большие языковые модели (БЯМ, или large language models, LLM).

Мы исследовали принципиальную применимость, для задач генерации и верификации кода Verilog, ряда доступных БЯМ/LLM, как доступных на зарубежном «облачном» развертывании (ChatCPT и Grok), так и развернутых локально (Llama и Qwen2) на доступном штатном оборудовании - MSI Raider GE77HX 12UGS notebook with 12th Gen Intel(R) Core(TM) i7-12800HX 2.00 GHz, 32.0 GB RAM, 23.9 GB GPU NVIDIA GeForce RTX 3070 Ti Laptop GPU; 2) MacBook Pro with 2.9 GHz 6-Core Intel Core i9, Radeon Pro 560X 4GB Intel UHD Graphics 630 1536 MB, 32 GB 2400 MHz DDR4. При экспериментах всегда одна модель использовалась для генерации кода Verilog по спецификации, а другая — для его верификации как такового, и затем — относительно исходной спецификации. Эксперимент показал принципиальную возможность решений обоих поставленных задач в обоих («облачном» и локальном) вариантах развертывания.

Возможность использования различных моделей позволяет рассматривать возможность реализации вычислительных архитектур «доверенного» проектирования СБИС на основе ансаблированной конкурентной генерации и верификации с арбитражом, когда целевой код генерируется различными моделями на основе одной и той-же спецификации, а затем альтернативные варианты генерации верифицируются другими различными моделями с арбитражом делаемых выводов. Кроме того, имеющиеся публикации [1,2,3] указывают на возможность обучения собственных БЯМ/LLM для решения указанных задач на основе как публичных, так и собственных закрытых данных.

В качестве проблем требующих решения, мы указываем необходимость изучения вопроса о возможности минимизации «галлюцинаций», проработки возможности компактизации и структуризации результатов верификации для оперативного анализа, возможности формализации и автоматизации самого этого анализа, ответственности за принятие решений на основе автоматической верификации, а также технические и методические возможности до-обучения моделей на собственных данных.

Литература

1. RTLCode: Fully Open-Source and Efficient LLM-Assisted RTL Code Generation Technique / Liu, Shang and Fang, Wenji and Lu, Yao and Wang, Jing and Zhang, Qijun and Zhang, Hongce and Xie, Zhiyao // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, IEEE.
2. RTLCode: Outperforming GPT-3.5 in Design RTL Generation with Our Open-Source Dataset and Lightweight Solution / Liu, Shang and Fang, Wenji and Lu, Yao and Zhang, Qijun and Zhang, Hongce and Xie, Zhiyao // 2024 IEEE International Workshop on LLM-Aided Design, 2024, IEEE.
3. OpenLLM-RTL: Open Dataset and Benchmark for LLM-Aided Design RTL Generation / Liu, Shang and Lu, Yao and Fang, Wenji and Li, Mengming and Xie, Zhiyao // 2024 IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2024, IEEE/ACM

Very large scale integration circuits design using large language models

Kolonin A. G.¹

¹ *Novosibirsk State University
630090, Russia, Novosibirsk, Pirogova st., 1
anton.kolonin@yandex.ru*

Abstract

The thesis presents the use cases of large language models for designing very large scale integration circuits, including generation of Verilog/VHDL code and tests according to specifications, as well as code verification. The article presents the results of basic testing using a set of large language models, proposes computational architectures for competitive ensemble-based generation and verification with arbitration, and indicates possible risks and problems that require solutions.

Keywords: LLM; VLSI; Verilog; VHDL; large language model; design of very large scale integration circuits.

Today, the most important stage of designing very large scale integrated circuits (VLSI) is logical design, when a software description of logic in Verilog or VHDL is created based on text specifications. Also, at this stage, additional verification of the obtained codes can be carried out, checking them for compliance with the original initial specifications and the absence of logic not provided for by the specifications, which is especially important for ensuring the "trusted" of the created hardware base. In addition, tests for software testing of the verified VLSI code can be created based on the same specifications. As shown in the works [1,2,3], modern large language models (LLM) can be used for the specified purposes at this stage.

We investigated the fundamental applicability, for the tasks of generating and verifying Verilog code, of a number of available LLMs, both available on foreign "cloud" deployment (ChatCPT and Grok) and deployed locally (Llama and Qwen2) on available standard equipment - MSI Raider GE77HX 12UGS notebook with 12th Gen Intel(R) Core(TM) i7-12800HX 2.00 GHz, 32.0 GB RAM, 23.9 GB GPU NVIDIA GeForce RTX 3070 Ti Laptop GPU; 2) MacBook Pro with 2.9 GHz 6-Core Intel Core i9, Radeon Pro 560X 4GB Intel UHD Graphics 630 1536 MB, 32 GB 2400 MHz DDR4. In the experiments, one model was always used to generate Verilog code according to the specification, and the other one was used to verify it as such first, and next verified relative to the original specification. The experiment showed the fundamental possibility of solving both tasks in both ("cloud" and local) deployment options.

The possibility of using different models allows us to consider the possibility of implementing computing architectures of "trusted" VLSI design based on ensembled competitive generation and verification with arbitration, when the target code is generated by different models based on the same specification, and then alternative generation options are verified by other different models with arbitration of the conclusions made. In addition, existing publications [1,2,3] indicate the possibility of training proprietary LLMs to solve the specified problems based on both public and proprietary closed data.

As problems requiring solutions, we indicate the need to study the issue of the possibility of minimizing "hallucinations", the development of the possibility of compacting and structuring the results of verification for operational analysis, the possibility of formalizing and automating this analysis itself, responsibility for making decisions based on automatic verification, as well as technical and methodological capabilities for pre-training models on the proprietary data.

References

1. RTLCode: Fully Open-Source and Efficient LLM-Assisted RTL Code Generation Technique / Liu, Shang and Fang, Wenji and Lu, Yao and Wang, Jing and Zhang, Qijun and Zhang, Hongce and Xie, Zhiyao // IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2024, IEEE.
2. RTLCode: Outperforming GPT-3.5 in Design RTL Generation with Our Open-Source Dataset and Lightweight Solution / Liu, Shang and Fang, Wenji and Lu, Yao and Zhang, Qijun and Zhang, Hongce and Xie, Zhiyao // 2024 IEEE International Workshop on LLM-Aided Design, 2024, IEEE.
3. OpenLLM-RTL: Open Dataset and Benchmark for LLM-Aided Design RTL Generation / Liu, Shang and Lu, Yao and Fang, Wenji and Li, Mengming and Xie, Zhiyao // 2024 IEEE/ACM International Conference on Computer Aided Design (ICCAD), 2024, IEEE/ACM