🚀 LLM4ChipDesign

Generative AI for Chip Design: Stackable Modules

Three main topics: 🔧 LLM4Verilog | 🛡️ LLM4Validation | ⚠️ LLM4Security

📦 **Github: https://github.com/FCHXWH823/LLM4Hardware**

🧐 Who is the target audience

This collection of training modules is designed for those from relevant backgrounds who want to understand how artificial intelligence is revolutionizing computer chip design. Whether you are a high school student exploring STEM fields, an undergraduate student in EE/CS/CE or any other STEM field, an educator in the relevant fields, a professional chip designer interested in AI applications, these materials will guide you.

The teaching duration for each module is approximately 4 hours.

🔍 What you will discover

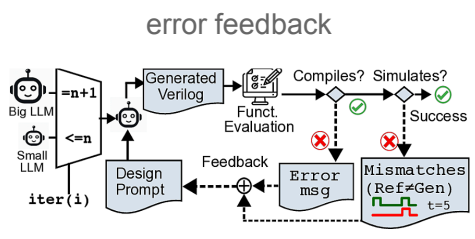
This project explores the application of large language models (LLMs) in the field of chip design with three main topics:

- **LLM4Verilog: Generate functional Verilog modules from design prompts using LLMs.**
- **LLM4Validation: LLM-generated SystemVerilog assertions, testbench files for hardware validation.**
- **LLM4Security: LLM-aided detection of hardware security risks.**

🌟 Featured Modules

🔧 LLM4Verilog

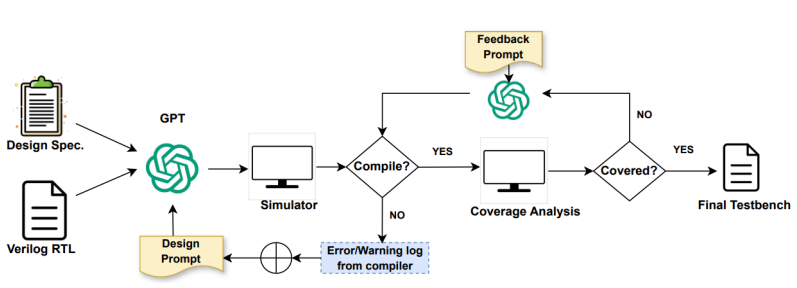
AutoChip: Generate functional Verilog modules from design prompts using LLMs with iterative error feedback



Paper

🛡️ LLM4Validation

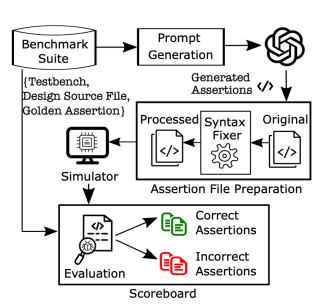
FSM Testbench Generation: LLM-generated testbench files for hardware validation



Paper

⚠️ LLM4Security

Security Assertions: LLM-generated security assertions for detecting security risks



Paper

🔧 LLM4Verilog

Functional Verilog generation with LLM feedback (AutoChip)

Paper Code Slides

LLM Reasoning-based Verilog code generation (VeriThoughts)

Paper Code

Hierarchical LLM prompting for complex chip design (ROME)

Paper Code Slides

Deterministic LLM-based Verilog synthesis from Conjunctive Normal Form (Veritas)

Paper Code Slides

LLM-aided prefix adder circuit design (PrefixLLM)

Paper Code

🛡️ LLM4Validation

LLM-based Natural Language to SystemVerilog Assertion (Hybrid-NL2SVA)

Paper Code Slides

LLM-based testbench generation and bug detection (FSM Testbench Generation)

Paper Code

Retrieval-Augmented Generation (RAG)-based SystemVerilog assertion using LLM

Code

LLM-based hardware assertion generation for security (Security Assertions)

Paper Slides

⚠️ LLM4Security

LLM-based testbench generation and bug detection (FSM Testbench Generation)

Paper Code

LLM-based hardware assertion generation for security (Security Assertions)

Paper Slides

LLMs for black-box hardware IP piracy (LLMPirate)

Paper Slides

⚡ LLM4HLS

Bridge software-to-hardware design gap using LLMs to refactor C code for High-Level Synthesis (C2HLS)

Paper Code Slides

LLM4Analog

LLM-generated analog circuit netlists for circuit schematic images (Masala-CHAI)

Paper Code

📊 Repository Statistics

12+	9	10+	5
Research Projects	Git Submodules	Research Papers	Main Focus Areas

🎯 **Mission:** Advancing the intersection of Large Language Models and Chip Design

🔬 **Research Areas:** Verilog Generation • Validation • Security • High-Level Synthesis • Analog Circuit Generation

🌐 **Explore:** Slides, Colab Scripts, and Reference Papers