Generative AI based Chip Design

(Developed by Prof. R. Karri, S. Garg, NYU; JV Rajendran, TAMU; O Sinanoglu, NYU-AD; K. Basu, RPI)

Lectures/Lab Fall 2025: Friday 2:00-4:30

Instructor: Prof. Ramesh Karri; Co-Instructor: Dr. Weihua Xiao

Slack Link: https://rameshnyu.slack.com/archives/C09EEMA6UJU

Course Github Repository: https://github.com/FCHXWH823/LLM4ChipDesign

Classroom: **Jacobs Hall, 6 Metrotech Room 207**

Who This Is For: This collection of **Generative AI based Chip Design** modules are designed for those with relevant backgrounds who want to understand how artificial intelligence (AI) is revolutionizing computer chip design. Whether you are an undergraduate student in EE/CS/CE or other STEM-adjacent fields, an educator in relevant fields, a professional chip designer interested in AI applications, these modules are for you.

What You'll Discover: Explore how Large Language Models (LLMs) are used to automatically design computer chips. You will learn about how to convert software code to hardware designs, generating computer circuits from natural language descriptions, creating tests for chip validation, and more.

Featured Topics

LLM4Verilog: Automatic generation of Verilog using LLM

LLM4Validation: Automatic generation of SystemVerilog assertions & testbench files using LLM

LLM4Security: Automatic detection of hardware security risks using LLM

LLM4HLS: Automatic High-level Synthesis for C codes using LLM

LLM4Analog: Automatic generation of analog circuit netlists from schematics.

Getting Started: Begin with the interactive tutorials in the 'colab-scripts' folder. These are designed to run in your web browser without any software installation. Each tutorial includes step-by-step explanations and real examples. The README.md file provides links to all papers, code repositories, and additional resources. Don't worry if some technical terms are unfamiliar at first - the materials are designed to build your understanding progressively.

Course Structure: Each module is taught over 2 weeks following a consistent pattern: Week 1: Introduce basic concepts and theory, assign homework/projects. Week 2: Hands-on problem solving using GenAI based tools.

Grading (Tentative): About 8 Labs: Five selected labs will be Graded Lab points 16 * 5 – 80 points; Project 20 points (Preliminary Proposal -5 points); Final Project including report 15 points) To. —Total 100 points

Code of Conduct: All students must adhere to the NYU Tandon Student Code of Conduct (https://engineering.nyu.edu/life-tandon/student-life/student-advocacy/student-code-conduct)

Draft weekly outline

Week	Module	Activities
1	AutoChip: Generate Functional Verilog	Intro to LLM-based Verilog Generation
2		Colab: Hands-on AutoChip
3	LLM-aided Testbench Generation	LLM-aided Testing Methodologies
4		Colab: FSM Testbench Generation
5	VeriThoughts: Verilog Code Gen using Reasoning and Formal Verification	Formal Verification in Hardware Design & Integrate into LLM-aided Verilog Generation
6		Colab: Use VeriThoughts to Generate Verilog
7	GenAI-based Hierarchical Verilog	Hierarchical Design Lab
8		Colab: Hierarchical Verilog & Project Proposal
9	GenAI-aided Verilog for Prefix Adders	Design Space Exploration Prefix Adders
10		Colab: PrefixLLM to Generate Adder Ckt
11	Natural Language2SystemVerilog Assertion	Translate Natural Language 2 SystemVerilog Assertions
12		Colab: NL2SVA Tools& Project Presentation
13	Bridge Software-to-H/WDesign Gap	High-Level Synthesis with LLMs
14		Colab: C→HLS Translation;Submit Final Project
15	GenAI-aided H/W Security	IP Piracy

16		Colab: LLM for IP Piracy
15	Masala-CHAI: Large-Scale SPICE Netlist Dataset for Analog Circuits	GenAI to Generate Analog Ckts from Schematics
16		Colab: Masala-CHAI to Generate SPICE Netlists