Agentic AI Design Flow from Concept to GDS in ASIC Design

Aditya Patra
Department
School Name
Bay Area, California
email address

Saroj Rout
Dept. of Electronics Engineering
Silicon University, Odisha
Bhubaneswar, India
ORCID: 0000-0002-5191-8191

Arun Ravindran
Dept. of Electrical and Computer Engineering
University of North Carolina at Charlotte
Charlotte, USA
arun.ravindran@charlotte.edu

***Abstract*— Abstract here...**

***Index Terms*—Agentic Flow, Generative AI, Keyword Spotting (KWS)**

I. Introduction (Arun)

The development of Generative AI (GenAI) Large Language Models (LLMs) has brought significant advancements in artificial intelligence, allowing systems to generate human-like content across various mediums, including text, images, and code. Recent research has investigated the application of LLMs in digital system design, particularly in generating hardware description languages (HDLs) such as Verilog from natural language descriptions.

Simultaneously, agentic AI workflows have emerged, where autonomous AI agents perform specific tasks within defined parameters. These workflows are increasingly used in software design to automate tasks such as code generation, debugging, and testing, thereby improving development speed and minimizing human error. However, while agentic AI workflows have proven effective in software code generation, applying these techniques to hardware design presents additional complexities. This is due to the diverse range of tools required to meet functional, correctness, and physical constraints in hardware systems.

- Motivation
- Research questions
- Work done
- Key results
- Organization of paper
- Figure: Design flow overview

II. Related Work (Arun)

- Application of AI in code generation
- Application of AI in hardware

[1]

III. Background (Arun)

- Generative AI
- Agentic AI
- ASIC design flow

IV. System Design (Saroj)

- KWS design
- Figure

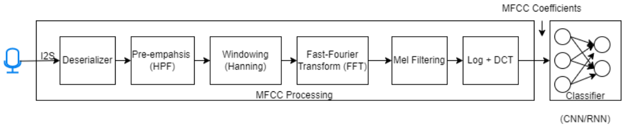

Fig. 1. Typical Implementation of a Keyword Spotter (KWS)

A. RTL-to-GDS Design Flow

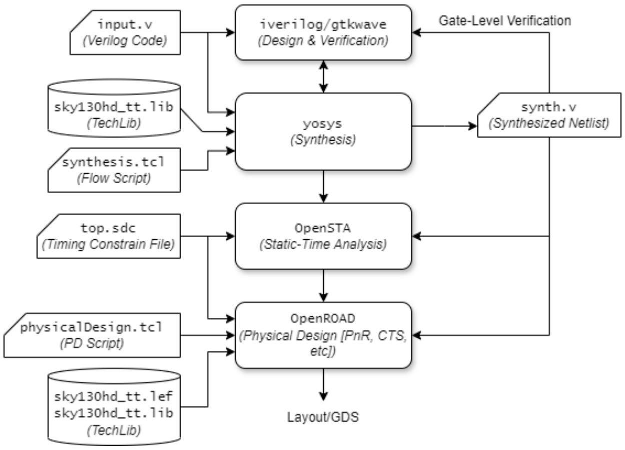

Fig. 2. Typical Implementation of a Keyword Spotter (KWS)

V. Agentic AI Design Flow (Arun)

- Agentic architecture
- Figure
- Prompt engineering strategies

VI. Evaluation (All)

- LLM model, libraries, technology library
- Fig - selected prompts
- Fig - Frequency domain ouptut
- Table - power, area, transistor count

VII. Discussion (Saroj and Arun)

- Impact
- Lessons learnt
- Limitations of approach
- Future improvements
- Other avenues of exploration

VIII. Conclusions (Arun)

- Summary of work
- Key results
- Expected results by final submission

Acknowledgment

References

[1] S. Thakur, B. Ahmad, H. Pearce, B. Tan, B. Dolan-Gavitt, R. Karri, and S. Garg, "Verigen: A large language model for verilog code generation," *ACM Transactions on Design Automation of Electronic Systems*, vol. 29, no. 3, pp. 1–31, 2024.