AI Agents for Automatic Chip Design from Specification to GDS

Aryan Rauniyar, Electrical Engineering
Mentor: Vidya Chhabria, Assistant Professor,
School of Electrical, Computer and Energy Engineering

Introduction & Question

- Application-Specific Integrated Circuit (ASIC) design flow is notoriously inaccessible, as modelling complex circuits are skills reserved for long-term experts.

- However, we want to bridge that gap, as this research aims to develop an AI agent that can effectively produce Verilog **[2]** and troubleshoot computer chip design in order to reduce the barrier to entry for electronic design automation tools.

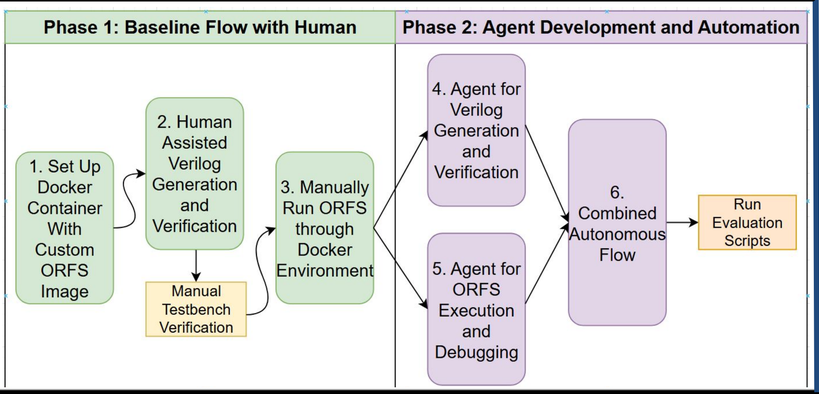
Research Question: Can an AI agent (in the form of a large language model) autonomously generate and optimize digital circuit designs from functional specifications, producing tapeout-ready layouts using open-source EDA tools?

Materials

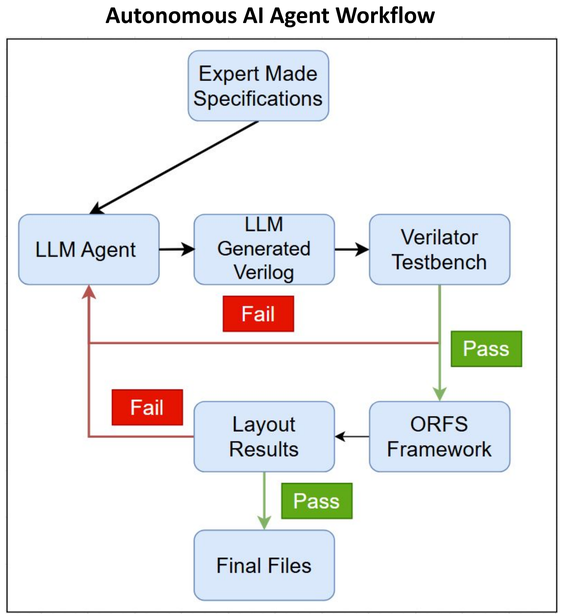
Materials: Problems and Specifications from the Spec2Tapeout Github **[3]**.

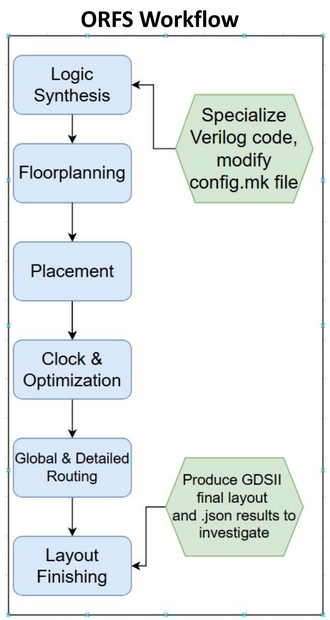
Software: A Chat-GPT API powered agent interacted with a custom Docker container that has on it loaded an OpenROAD-flow-scripts **[4]** image and Verilator.

Implementation



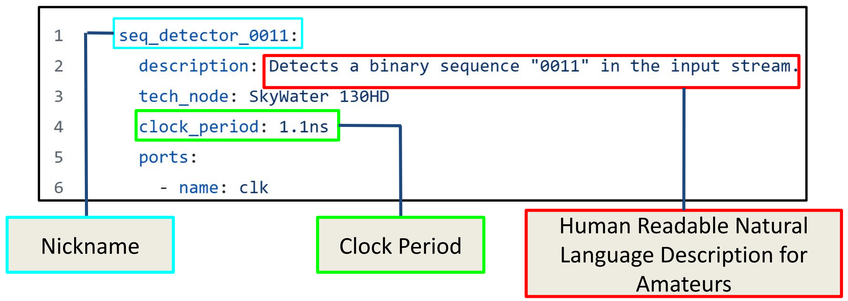
Flow Charts





Experimental Setup

A standardized format dictating how to create the specifications is necessary. Below is an excerpt from the .yaml specification for the sequence detector. The clock period is a necessary value for the [config.mk](#) input file in order to dictate the timing constraints of the circuit. Additionally, the constraint.sdc input pulls the formalized nickname it will use in all operations from here.



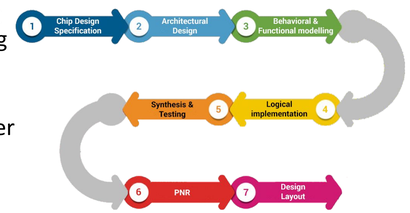
Results

Problems specification .yaml files and specifications taken from Spec2Tapeout repository

Problem Type	Pass/Fail Evaluation	Issue Type
0011 Sequence Detector	Pass	N/A
Dot Product	Pass	N/A
Exponential Function	Pass	N/A
16-Bit Multiplier	Fail	15 Set Up Violations
Finite Impulse Response (FIR) Filter	Fail	29 Set Up Violations

Conclusions/ Future Work

- Current LLM agents are capable of autonomously completing the ASIC design loop. But, timing and routing violations still occur within ORFS.
- Future Work will focus on further training and refining models in order to fix those violations
- The next step is to increase the number of circuit designs to test, while also creating an autonomous pipeline to physical tapeout, processing the GDSII files left behind.



[1] The complete cycle from specification to physical tapeout

References

[1] K. Chauhan, "ASIC Design Flow in VLSI Engineering Services – A Quick Guide," eInfochips Blog, Jun. 4, 2019. [Online]. Available: https://www.einfochips.com/blog/asic-design-flow-in-vlsi-engineering-services-a-quick-guide/. [Accessed: Apr. 4, 2026].
[2] "IEEE Standard for Verilog Hardware Description Language," in IEEE Std 1364-2005 (Revision of IEEE Std 1364-2001) , vol., no., pp.1-590, 7 April 2006, doi: 10.1109/IEEESTD.2006.99495.
[3] ICLAD-Hackathon, "ASU-Spec2Tapeout-ICLAD25-Hackathon," GitHub. https://github.com/ICLAD-Hackathon/ASU-Spec2Tapeout-ICLAD25-Hackathon.git (accessed Apr. 3, 2026).
[4] Ajayi, T., Chhabria, V.A., Fogaça, M., Hashemi, S., Hosny, A., Kahng, A.B., Kim, M., Lee, J., Mallappa, U., Neseem, M., Pradipta, G., Reda, S., Saligane, M., Sapatnekar, S.S., Sechen, C., Shalan, M., Swartz, W., Wang, L., Wang, Z., Woo, M., & Xu, B. (2019). INVITED: Toward an Open-Source Digital Flow: First Learnings from the OpenROAD Project. 2019 56th ACM/IEEE Design Automation Conference (DAC), 1-4.

FURI

Ira A. Fulton Schools of Engineering
Arizona State University