Name: Dhirajzen Bagawath Geetha Kumaravel
NetID: db5309

Veritas Colab Lab Report

Introduction

The Veritas lab demonstrates the integration of Large Language Models (LLMs) with hardware design verification. While tools such as ChipChat, AutoChip, and Verigen show LLMs' potential for Verilog code generation, the generated outputs often contain syntax errors and logic inaccuracies. Veritas aims to address this challenge by embedding verification and testing directly into the generation process.

Methodology

The methodology involves representing circuits in Conjunctive Normal Form (CNF). Each logical gate (e.g., AND, NAND, NOR, XNOR) has an equivalent CNF representation. By combining CNFs of gates, a full circuit can be represented and tested exhaustively. This iterative approach allows the framework to learn from smaller circuits and progressively generate larger, more complex designs.

Experiment Details

For this lab session, a 3-bit adder design was generated and verified. The following artifacts were produced:
- Verilog implementation of the 3-bit adder (`adder_3-bit.v`)
- CNF representation of the circuit (`adder_3-bit.cnf`)
- Bench file for equivalence checking (`adder_3-bit.bench`)
- Truth table in CSV format (`adder_3-bit_tab.csv`)

Generated Artifacts

Artifact	Description
adder_3-bit.v	Verilog code implementation of the 3-bit adder
adder_3-bit.cnf	CNF clauses representing the logic gates
adder_3-bit.bench	Benchmark circuit description for verification
adder_3-bit_tab.csv	Exhaustive truth table of inputs/outputs

Name: Dhirajzen Bagawath Geetha Kumaravel
NetID: db5309

Observations

The Veritas framework successfully demonstrated the integration of CNF-based verification within LLM-guided hardware generation. The 3-bit adder design was not only generated but also validated against its truth table and CNF representation. This indicates a reliable flow for scaling towards more complex arithmetic circuits.

Error Iterations & Debugging

During the CNF generation process, the script required multiple iterations (8 restarts) to arrive at a correct and valid CNF file. The major issues encountered were:

- Incorrect CNF Symbols: The initial outputs used non-standard or missing symbols, which made the CNF representation invalid for automated verification.
- Improper Variable Naming: Variables were generated with underscores and inconsistent formats, leading to incompatibility with verification tools.

These errors were corrected progressively in subsequent runs, with each iteration fixing one class of errors. By the final iteration, the CNF file (`adder_3-bit.cnf`) had the proper syntax, valid symbols, and correctly named variables, making it usable for benchmarking and truth table validation.

Conclusion

Veritas bridges the gap between generation and verification in AI-driven hardware design. By coupling CNF-based correctness checks with Verilog synthesis, the lab illustrates how future design flows can leverage AI not just for code generation, but also for integrated validation and error reduction.