Week 4

Codefest: Let's cook up some hardware

ECE 410/510
Spring 2025

Challenge #15

Overview and context:
The goals of last week's project challenge #13 were to (1) make a first attempt to decide your SW/HW boundary and to (2) test the entire design workflow by implementing a toy example. The goal for this week is to implement and simulate (in a high level HW description language) a version of your design.

Learning goals:

- Generate a first hardware description (in Verilog, SystemVerilog, or VHDL) of your accelerator chip (or of a toy example).
- Learn how to write HDL (or use vibe coding).
- Learn how to write a testbench.
- Learn how to simulate HDL.

Suggested tasks:

1. Based on your HW/SW boundary (which may be tentative) from last week, write an HDL (= hardware description language) description of your HW part.
2. If you have never done Verilog, SystemVerilog, or VHDL, here are some resources:
    - Verilog: https://www.geeksforgeeks.org/getting-started-with-verilog
    - SystemVerilog: https://verificationguide.com/systemverilog/systemverilog-tutorial
    - VHDL: https://nandland.com/learn-vhdl
3. Alternatively, take your Python or C/C++ high-level code and ask your favorite LLM to translate it into SystemVerilog.
4. Alternatively, use toy example (e.g., Frozen Lake Q table update formula) and convert that into a HW design (manually or with an LLM).
5. Write (or have your LLM write) a basic testbench for your design: https://fpgatutorial.com/how-to-write-a-basic-verilog-testbench
6. Simulate and test the design.
7. List of HDL simulators: https://en.wikipedia.org/wiki/List_of_HDL_simulators
8. Unless you know Cadence & Synopsis tools, I'd recommend you use verilator:
    - Verilator is a free Verilog/SystemVerilog simulator.
    - https://github.com/verilator/verilator
9. Alternatively, you can use one of the tools from the week 3 codefest (challenge #13) if you are able to write Verilog RTL code.