Logic Type Rock
MP4 CP1 Progress Report and Road Map
dbycul2, danielp7, jayhp2

## Progress Report

Basic RV32I Pipelined Design:

- Control:
  - Design (dbycul2): Design of our control ROM was inspired from the MP2 control FSM except with an interface for the signals in and out. Instead of cycling through states, we assigned the values of the control word that was the output. We
  - Verification (dbycul2): Verification of our control unit consisted of a separate testbench to verify that the signals were being generated correctly with respect to the expected output.
- Datapath:
  - Design (All): To start our datapath design, Dan instantiated and planned out the connections for most of the muxes and registers within the stages. He also wrote the module files for each general mux needed along with the extra features. Jay finished the datapath connections in each of the separate pipeline sections. Dawid then instantiated the between stage registers and connected the IF and ID portion of the datapath. Jay and Dan then finished the EX, MEM, and WB connections. We also accounted for memory signals towards the end.
  - Verification (jayhp2): Jay verified the datapath by checking each input and output signal to make sure the datapath gave the correct values. Jay also compared the register values during each stage of the pipeline and compared to expected values from mp2.
- Between Stage Registers:
  - Design (dbycul2): Dawid used structs as a way to represent the signals that would be held in registers. He then fed those structs into our register modules as inputs and outputs which will change every clock cycle.
  - Verification (All): The verification for the registers was a visual inspection of the values coming in and out of the registers. We were able to confirm that the input was assigned to the output after one clock cycle.
- Top Level
  - Design (All): Dan started the top level connections between different modules. Jay finished connecting the top level modules by creating structs with their respective signals for easier readability. Dawid connected the given memory files to our datapath.

## Road Map

1. Hook up RVFI monitor (Implementation and Verification = dbycul2)
2. Hook up shadow memory (Implementation and Verification = dbycul2)
3. Integrate the cache (Implementation = danielp7, Verification = dbycul2)
4. Implement arbiter for caches (Implementation = danielp7 , Verification = dbycul2)
5. Implement hazard detection (Implementation = jayhp2, Verification = jayhp2)
6. Implement forwarding (Implementation = jayhp2, Verification = danielp7)
7. Proposal for advanced features (All)

a. Branch target buffer and branch predictor (4 if 4-way set associative or higher + up to 8)
b. Pipelined L1 Cache (6)
c. Basic Hardware Prefetching (4)
d. Memory Stage Leapfrogging (12)