Checkpoint 1 Progress Report:

We have decided to implement a Tomasulo processor with superscalar, branch prediction, advanced multiplication, and potentially speculative loads.

Cameron designed the parameterizable queue. The queue maintains FIFO order and supports variable depth and width. A testbench to test filling and emptying the queue multiple times was used to verify that the design works.

Bobby and Zhihao created the design block diagram of the datapath. They modified the parameterizable queue to prepare for later superscalar integration. They also implemented the fetching logic with the queue integrated, which can constantly fetch instructions from some memory except when a stall signal is passed in. They tested the fetch logic using a simple testbench and checked the waveform manually. They have decided that manual testing was the easiest for this checkpoint since it's hard to drive an internal unconnected signal (instruction_queue_pop).

Checkpoint 1 Roadmap:

Main Feature:

Bobby & Zhihao:

- ROB (reorder buffer), used for in-order commit after out-of-order execution
- RS (reservation station), used for keeping track of input values and if they are ready
- CDB (common data bus), used for connecting output of execution units with ROB and RS

Cameron:

- Register file with scoreboard, used for keeping track of register values and current ROB instruction that will write new register values

All three of us: (pair programming session)

- ALU, or Execution Unit, used for calculating the actual output value
- Integration of all parts

Extra Feature (for checkpoint 3 and beyond):

- Load Store Queue, StoreBuffer, etc., used for supporting in-order store commits and dealing with hazards
- Advanced multiplier to speed up multiplication efficiency
- Branch Predictor, used for fewer occurrences of branch penalty
- Integration with mp_cache for faster memory access