Nathan Beauchamp, Rishi Thakkar, Melissa Jin
Team name: Silver Pipelining
ECE 411 MP3, Checkpoint 3

Progress Report:

For Checkpoint 3, our group expanded upon the work done in Checkpoint 2 to create a pipelined processor design that can handle all LC3-b programs, including those with data dependencies and control hazards. Since we had already designed and integrated an 8-way, 8-set L2 cache in Checkpoint 2, our work for this checkpoint amounted to handling data and control hazards only. To solve the issue of data hazards, we implemented data forwarding, creating the EX->EX and MEM->EX forwarding paths. This is a simple yet effective strategy that enables us to avoid stalling the pipeline in the case of data dependencies, since the result of a previous instruction can be forwarded to an earlier pipeline stage before it is written back to the register file. To solve the issue of control hazards, we decided to stall the pipeline upon detection of an instruction that modifies the value of the PC (BR, JMP, JSR, JSRR) until its address calculation has been determined. Once this has occurred, we resume normal pipeline operation. In order to test our design, we wrote sample programs that included unpadded branches and arithmetic and memory instructions with many dependencies. We then ran these programs on our processor in ModelSim, and verified that the final register values produced were the same as those given by the LC3-b simulator. We also ran "sanity checks" on our cache memory lists to verify that the correspond to the values being read/written. This gave us confidence that our design would successfully run the CP3 test code.

As in previous checkpoints, work was divided fairly evenly for this checkpoint. Rishi and Melissa implemented data forwarding, while Nathan and Rishi implemented stalling upon branch detection. The three of us worked together to test and debug the design.