ECE411 CP2 Progress Report

Progress

For checkpoint 2, we implemented the decode stage, dispatch stage, RAT, RRF, Free List, ROB, reservation stations, valid array, PRF, execute stage (for alu, mul, and div), and the common data bus. We tested the functionality of our processor by modifying the random testbench from mp_verif, as well as running additional tests for edge-cases.

Design Decisions

Our decode stage is similar in functionality to mp_pipeline as we take an incoming instruction packet from the instruction queue (populated by fetch) and fill out all of the relevant metadata for the instruction. This instruction packet gets passed along to the dispatch stage.

The dispatch stage then asks for a new physical register (if the instruction uses rd), invalidates that register in the valid array, and gets the operand mappings from the RAT. Afterwards, dispatch decides which RS to put the instruction into (depending on the type of instruction), here we chose to group BR/MUL/DIV, ALU, and MEM into their own separate groups.

For the reservation stations we went with a collapsing queue approach. Once the collapsing queue selects an instruction it latches it onto an external register, and sends operand addresses into the PRF. Next cycle, the operands of the instruction are available to the functional unit, which carries out the execution, and broadcasts the result to the valid array, ROB, and the PRF.

For the execute stage, we currently have the branch, mul, and div functional units grouped together with the incoming instruction packets coming from the branch, mul, div reservation station. The incoming instruction is decoded into the proper functional unit. The alu operations are handled by the alu functional unit with instructions being fed by the alu reservation station. The multiply and divide functionality are handled by imported IP blocks, and the alu functionality is kept similar to that in mp_pipeline. The resulting instruction packets are populated with the outputs from the functional units and are broadcasted to ROB, PRF, and valid array. Additionally, if any of the functional units are stalling, a stall signal is sent to the appropriate reservation station to tell it to stall.

The instruction being broadcasted on the cbd is fed into the ROB, which checks its entries to see if there is a matching instruction to mark as complete and ready to commit. Once a ready ROB instruction is at the head of the ROB queue, it is committed and the physical and architectural register info for the committed instruction are sent to the RRF.

Work Split

Ahmed worked on implementing and verifying the decode stage, execute stage (alu, mul, div functional units), cdb, and the random testbench.

Eddie worked on implementing and verifying the dispatch stage, Free List, ROB, PRF, reservation stations, and the random testbench.

Mahir worked on implementing and verifying the dispatch stage, RAT, RRF, ROB, and reservation stations.