EECS470 Project Proposal

Hsiang-Yang Fan (seanfan@umich.edu), Yan-Ru Jhou (yanruj@umich.edu), Hsin-Ling Lu (hsinling@umich.edu), Eli Muter (innitir@umich.edu), Daniel Yu (dddaniel@umich.edu)

Design Features

The main design features that we will add to our pipeline are:

- I and D Caches
- Multiple functional units with varying latencies
- Out-of-order execution
- Branch prediction with address prediction

Additional features that we are considering implementing are:

- 3-way superscalar execution*, because we would like to increase the degree of instruction level parallelism in a single microprocessor.
- Early branch resolution, because branch resolution time is a significant factor in performance. We want to save cycles by making branch resolution as soon as we find a branch misprediction instead of the branch hitting the head of ROB.
- Non-blocking L1 data cache because one of the major performance bottlenecks for microarchitecture design lies in the memory manipulation. We believe that if we add memory hierarchy to our microprocessor, we could boost our benchmark. We also have a team member doing research on formal verification on memory consistency models, and it would be a great opportunity to work on memory management at the microarchitecture level.

Team Logistics

We plan to use stack to communicate, as well as to have a weekly team meeting every Wednesday after class.

We will create a central repository with a production branch in order to manage our code. Each group member is going to fork the central repository for their own development. For each task completed by each member, they will create a pull request for the project leader's approval to merge into the production branch. We are also considering integrating CI/CD development workflow by taking the advantage of "pipelines" CI/CD support from Bitbucket.

To organize and manage our work, we'll add each task assignment to an ongoing Trello project. The assignments for our initial tasks are:

- Yan-Ru will create central git repository with a main/production branch
- Implementing multiple functional units with varying latencies
  - Hsiang-Yang Fan will split the existing integer ALU into multiple functional units
  - Hsin-Ling Lu will implement a functional unit for multiplication that uses a pipelined multiplier
- Implementing out-of-order execution
  - Yan-Ru Jhou will implement the Register Status and Functional Unit Status tables
  - Daniel Yu will implement the Dispatch Stage
  - Eli Muter will implement the Issue Stage

Schedule

Our project-long schedule is, tentatively: implementing the multiple function units and out-of-order execution by the first checkpoint on October 24th; the I and D caches and branch prediction by the second checkpoint on November 10th; and all additional features by the third checkpoint on December 2nd. We plan on writing our final project report after the third checkpoint, since all our pipeline's features should be fully implemented.