CSE 471: Computer Design & Organization
Assignment 3
Due: Thursday, May 3

The purpose of this assignment is to develop your intuition about the effects on performance of different parts of the CPU hardware that are designed to improve program execution. This assignment is much more open-ended than the previous two and will require more planning on your part in designing your experiments.

First you should start by (re)reading Section 4.4. of "The SimpleScalar Toolset" whose link is on the Simulator Information page of the CSE471 Web page. This Section explains the OOO architecture that SimpleScalar simulates, namely a 6-stage pipeline with the "ruu" being a combination of a register mapping device and a reorder buffer. In this section you can also see how to set various parameters such as, for example, fetch, decode and issue widths, number of functional units etc.

The main question that you should try and answer on a particular example is how much out-of-order execution "buys" in terms of CPU execution time. In order to do so, you are going to perform a study of in-order processors, increasing the widths of both the front-end and back-end of the pipeline. You will then attempt to see how you can match the performance of the "best" in-order processor with the least expensive (in terms of instruction issue and related widths and execution units) out-of-order machine possible.

In order to try and reduce the impact of parts of the architecture that are not totally relevant to this study, you should have "large" caches (e.g., 64KB L1 I and D caches and a 1 MB L2 cache) and TLBs (e.g, 64 sets). You should also keep the same branch predictor for all experiments (e.g., gshare with 10 history bits). Finally to somewhat reduce the impact of initialization, you should "fast forward" the first 25 Million instructions before recording statistics for the next 100 Million instructions. You should run your simulations using *gzip* with the same input as in Assignment #2.

For your limit study of in-order processor, you should start with the simple configuration of the previous assignments modulo the changes given in the previous paragraph, namely single-issue (and fetch and decode), a ratio of 1 for the speed of the front-end relative to the execution core (this is a parameter that you should not change), 1 Integer ALU, 1 Integer multipliers/dividers, 1 Memory system port, 1 Floating-point ALU and 1 Floating point multipliers/dividers (as far as we know *gzip* does not use f-p so you should not have to worry about these last two parameters). You should then increase the front-end width and at the same time some of the other parameters. For example, you could see what a 2 issue in-order processor with fetching of 2 (and also maybe 4) instructions at a time and 2 decode/cycle buys you with some or no increase in the number of integer units and memory ports. You should not go over an issue width of 4 instructions/cycle.

For finding the "cheapest" OOO processor, you might want to start in reverse and take as your initial configuration the one from the best in-order processor plus an rru size that you'll have to determine. Then you might want to pare down some of the parameters.

The report you'll write should have a strong methodological component explaining your approach. Of course graphs and/or tables will be necessary to report your results. Whatever tack you decide to take, think about what parts of the in-order processor (respectively out-of-order processor) you plan to change and develop a good hypothesis as to why it might execute faster (respectively almost as fast) than the previous ones. Also, try to keep other factors constant in your comparisons, so your results don't reflect more than one design change. In other words, don't just run a bizillion arbitrary simulations and pick the best. Think carefully about what parts of the design you want to change and why.

You have several means to change your CPU designs: parameter values, the def files and changing the SimpleScalar code. If you do the latter, it would behoove you to check with Jacob beforehand. He'll ward you off from ideas that will take you until next year to implement.

You are expected to work in teams of 2 people; again try to choose a different person.