CSE 471: Computer Design & Organization
Assignment 3
Due: Thursday, May 8

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 area. This section explains the out-of-order architecture that SimpleScalar simulates, namely a 6-stage pipeline with the “ruu” being a combination of a register mapping device and a reorder buffer. (A reorder buffer is the Pentium’s version of an active list, but it also provides space to store values before they are put into the register file. You can read about it in the context of Tomasulo on p. 106 and the Itanium on G-31 and G-32 in our text.) In the web page you can also see how to set various parameters, such as fetch, decode and issue widths, number of functional units, etc.

The main question that you should try to answer is how much out-of-order execution “buys” in terms of CPU performance. Another way to phrase this is -- is it possible to design an in-order CPU that performs just as well the simplest out-of-order processor you can implement? To address these questions, you are going to perform a study of in-order processors, increasing the widths and hardware capabiltiy of both the front-end and back-end of its pipeline.

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. Run your simulations using *gzip* with the same input as in Assignment #2.

For both in-order and out-of-order processors, begin with the simple configuration of the last assignment 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 never change), 1 integer ALU, 1 integer multiplier/divider, 1 memory system port, 1 floating-point ALU and 1 floating point multiplier/divider (as far as we know *gzip* does not use floating point, so you should not have to worry about these last two parameters). You should then begin increasing the front-end width and at the same time some of the other parameters for the in-order processor. For example, you might see what a 2 issue in-order processor that fetches 2 (and also maybe 4) instructions at a time and does 2 decodes/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.

How you answer this question is up to you. Whatever tack you decide to take, follow the good experimental methodology you used in the branch prediction assignment. Think about what parts of the in-order processor you plan to change and develop a good hypothesis as to why the new design might execute faster than the out-of-order version or the previous in-order design. Base new changes on the results of previous experiments. Also, try to keep other factors constant in your comparisions, 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 Andrei beforehand. He'll ward you off from ideas that will take you until next year to implement.

Depending on your new design, Andrei may assign you a new cycle time relative to that used by the out-of-order processor. Use this new cycle time to report your bottom line performance.

Most credit for the most clever and sucessful design changes.

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