Computer Design & Organization
Assignment 4
Due: Friday, November 17

The purpose of this assignment is to help you gain intuition about how instruction width and the number and variety of functional units affect processor performance, to learn how to satisfy different applications needs with a fixed hardware budget, and to experience how much/little ILP exists in today's applications.

You are expected to work in teams of 3 people, with different partners than the last assignment. At the end of your report, say what part of the project each person worked on.

For this assignment, you should use the following inputs:

Integer: **perl charcount all_gre_words**
Floating point: **swim < swim.in**

To limit the running time of the experiment, you may use the "-max:insts" flag to limit executions to a reasonable number of instructions. Please document how you arrive at how many is a reasonable number.

Begin with the default configuration file that you used for homework 2, but specify a gshare 2-level branch prediction scheme that has 8 bit wide history, 1 history level, and 1024 entries in the local history table. In addition, start with two integer ALUs, two floating-point ALUs, and one integer and one floating-point multiplier.

1. Simulate both integer and floating point applications on an in-order machine while increasing issue widths from one to eight with the same number and type of functional units. Note that you may have to change more than one parameter to change the issue width. What do your results tell you about the ability of a superscalar processor to use functional units?

2. Increase the number of functional units to be a better match for each issue width. How do the results change and why? Are you using a reasonable number of functional units?

3. Design a hardware configuration for a 4-wide processor that gets the best performance on both workloads. Assume a hardware budget of **32** 'functional unit units', where:

a. an integer ALU is one unit
b. an integer multiplier/divider is ten units
c. a floating point ALU is two units
d. a floating point multiplier/divider is twelve units

Analyze the results, and explain why the configuration was optimal.

4. If you could pick one additional functional unit, what would it be and why? Do so. Does it solve your performance problems? Why or why not?

5. If you could make any change to the hardware, limiting yourself to the configuration parameters we have, what would it be and why? Do so. Does it solve your performance problems? Why or why not?

6. Change your processor from an in-order to an out-of-order implementation. How do the performance of difference issue widths, as well as the functional units needed for a 4-wide processor change, and why?