CSE471: Computer Design & Organization
Assignment 2
Due: Thursday, April 15

The purpose of this assignment is to supplement your knowledge of the design of different dynamic branch prediction schemes with intuition about their relative performance.

You can work in teams of 2 people.

1. For this assignment your applications should be (1) `go` with an input Fran will specify or `gcc2000` with the input *integrate.i* and (2) `compress` with the same input as for homework 1 or `gzip2000` with *input.source* as the input. The new applications are from SPEC2000 rather than SPEC95.
2. Simulate the same microarchitecture as in the first assignment, but vary the branch prediction strategy between:
   - 2-bit dynamic branch prediction with 1K counters
   - correlated branch prediction, using a number of history bits and pattern history table (PHT) size, such that the PHT requires the same amount of storage as the 2-bit predictor above. All structures should be global. The point here is that you're incurring the same hardware cost, but using it to implement a different branch prediction strategy.
   - a gshare predictor using also the same amount of storage (setting the xor bit in the configuration parameters of the predictor should do the trick). Fran will explain in discussion what gshare means.

   This means you will have separate simulations for the three branch prediction techniques.

   FYI, SimpleScalar determines the branch target address for both branch prediction schemes with a branch target buffer.
3. Generate and analyze your results. In particular, address the following issues:
   - Branch frequency.
     - How often are branches executed in your programs?
     - If there is a difference between the two programs, do you have a hypothesis to explain it?
     - How does this compare with the "average" frequency of every 4-6 instructions?
     - If there is a difference from the average, do you have a hypothesis to explain it?
   - Branch characterization.
     - For conditional branches, record the frequencies of each element of the Cartesian product: (*forward*, *backward*) x (*taken*, *not taken*). You can express your answer in a

table like this one:

	taken	not taken
forward	#	#
backward	#	#

Do you need SimpleScalar to generate additional stats to do this? If so, write the code to do so.

- Prediction accuracy.
  - Record the number of correct predictions for all branch prediction schemes, including what a static scheme would have achieved. (For the static scheme, you don't have to run a simulation; just use the branch characterization results.) In particular, discuss whether or not your results support the use of the traditional static scheme of backward branches taken and forward branches not taken
  - Which prediction scheme has the most correct predictions for each program? Can you hypothesize why? If you need to run additional simulations to justify your analysis and conclusion, do so. (Don't be overly surprised if your results do not coincide with conventional wisdom since you are executing a limited number of instructions at the beginning of an application).
  - This question is only applicable if each of your programs performed best with a different branch prediction scheme. If you had to pick only one prediction scheme that would be used for both programs, which would it be and why?

4. Is there a difference between the number of instructions fetched and the number of instructions committed for each scheme? Why or why not?
5. Is there a limit to the number of useful history bits, or is more history always better? Perform simulations to determine what the situation really is, considering all three branch prediction strategies. But limit yourself to about one order of magnitude more storage than the base case. What do you think explains your data?

   Keep in mind that when doing a sensitivity analysis of this kind, you should keep all factors constant except the one you are varying.
6. Write up your experiments, the results and an analysis of the results in a report, as outlined in the report handout and illustrated in the sample reports. In the results section, devote a different subsection to each of the issues listed above in items 3 through 5. Use tables and graphs to illustrate the results.