•

Computer Design & Organization
Assignment 2
Due: Wednesday, October 18

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

You are expected to work in teams of 3 people, and different partners than for the last assignment.

1. For this assignment your application should be `perl` with the same input as in homework 1.
2. Simulate the same microarchitecture as in the first assignment, but vary the branch prediction strategy between:
   - 2-bit dynamic branch prediction
   - 2-level correlated branch prediction, using the default number of history bits and pattern history table size
   - the traditional static scheme of backward branches taken and forward branches not taken.

   This means you will have separate simulations for the three branch prediction techniques.
3. Generate and analyze your results. In particular, address the following issues:
   - Branch frequency.
     - How often are branches executed in your programs? (Use the branches "total number of branches committed" metric.)
     - How does this compare with the "average" frequency of every 4-6 instructions?
     - If there is a difference, do you have an 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 the following table: Do we need additional stats generated to do this?

	taken	not taken
forward	#	#
backward	#	#

     - Discuss whether your results support the use of the static branch prediction algorithm or not.
   - Prediction accuracy.
     - Record the number of correct predictions for all branch prediction schemes.
     - Which prediction scheme has the most correct predictions? Why?
4. Is there a difference between the number of instructions fetched and the number of instructions committed? If so, why?

5. Comparing alternate 2-level correlated branch predictor implementations.

    - For the 2-level correlated branch predictor, vary the size of the pattern history table to gauge the sensitivity of branch prediction performance to prediction table size. What do your results tell you?
    - Vary the number of history bits. What do your results tell you?
    - Which has more effect on the quality of the predictions, the size of the prediction history table or the number of history bits? Any hypotheses why?

    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. In the results section, devote a different subsection to each of the issues listed above in item 3. Use tables and graphs to illustrate the results.

7. In addition to the experimental portion of the assignment, answer the following question.
    For a (3,2) correlated prediction scheme which has a global history register and one pattern history table, show the contents of the history register and the PHT after the following sequence of branch executions have taken place: 1 not-taken branch, 2 taken branches, 2 not-taken branches, 2 taken branches, and 2 not-taken branches.

    - The prediction bits in the pattern history table use 2-bit prediction, implemented with saturating counters. The counters are incremented when a branch is taken and decremented when a branch is not taken. Each counter is initialized to the weakly taken state, as illustrated on slide 4 from lecture. Assume this state has the value 10.
    - The global history register is initialized to zero.
    - The addresses of the ten branch instructions are: ffff0000, ffff0001, ffff0010, ffff0000, ffff0000, ffff0000, ffff0011, ffff0010, ffff0000.

Extra credit only:

After getting a sense of how prediction correctness changes as you change the size of the hardware data structures, the question that microarchitects usually face, is:

> If I have a certain amount of chip area that I can devote to branch prediction, how should I divide it among the two structures, the global history registers and the pattern history table.

For extra credit, you can try to answer that question for a bit budget of 2K bits. To focus your thinking, you could consider issues, such as the following:

- Given 2K bits, how many of those bits should you devote to the history registers and how many should be used for the pattern history table?
- Should there be one global history register that is used by all branches, or should there be several, each of which is used by fewer branches?
- If several, how many?
- How many entries should there be in the PHT?

Be sure that all your hardware structures have powers of two entries, so that they can be accessed easily by the PC or branch history bits. This will probably mean that you won't use up your entire bit budget.