CSE 471 Autumn 2001

Computer Design and Organization

Assignment #2

Due: Wednesday October 24

The purpose of this second assignment is to look at various branch predictors and reason about their performance. You should still be in teams of two but choose a different partner this time.

You will use SimpleScalar but the simulator for this assignment has been modified so that it now includes another option for branch prediction as well as printing of extra statistics about branches. You will use, however, a smaller data set than in Assignment #1 so that the simulations will be shorter, by about a factor of 8. However, as you shall see you'll have to run several experiments. The command line (all in one line, sorry about the formatting) that you should use is something like:

```
HW2/simplesim-2.0/sim-outorder -config ~/yourfile.cfg spec95-little/perl.ss
inputs/primes.pl < inputs/primes.in.small |& less > ~/whatever
```

1. Simulate the same microarchitecture as in Assignment #1 but vary the prediction strategy between:

   - A static BTFNT (Backward Taken Forward Not-Taken) predictor (option *static_comb* in the configuration file; don't worry if it appears it's not one of the choices – just enter *static_comb* as the option)
   - A 2-bit saturating counter scheme with 1024 entries in the BPB (option *bimod* in the configuration file; be sure to set the "right" number of entries)
   - A 2-level GAg correlated branch predictor with 10 bits of history (option *2lev* in the configuration file; be sure to set the "right" number of entries; do not use the "xor" feature)

   This means that you need to run 3 simulations for the first part of the assignment.

2. **Answer the following questions in the form of a report** (cf. the "report Handout"; you might also want to look at H&P pp 323-326 to give you an idea on how to present part of your data).

   - How did you set-up the parameters for the 3 simulations? (this should be part of the "Methodology" section.)

- What is the frequency of branches for the application you simulated? Is this consistent with the "conventional wisdom" found, for example, in your textbook? If not, can you conjecture why not?
- For each branch predictor, record the frequencies of each element of the Cartesian product: $(forward,\ backward) \times (taken,\ not-taken)$. A convenient way to display these statistics is with a table such as:

	taken	not-taken
forward	#	#
backward	#	#

- For each branch predictor, compute the branch prediction accuracy.
- What conclusions can you draw *from this single experiment* about the efficiency of each branch predictor for *this particular application* (i.e., do not believe that what you uncover here is necessarily the general trend). How does this efficiency impact the CPI?
- Why is there a difference between the number of instructions committed and the number of instructions fetched?
- Sensitivity analysis.
  The purpose of "sensitivity analysis" is to ascertain how the performance of a particular scheme, here a branch predictor, is affected when a single factor in the design is varied over a range of values while leaving all other parameters unchanged.
  - For the *bimod* predictor run simulations with a BPB half the size and with a BPB twice the size of the one used before, What do your results tell you?
  - For the *2lev* predictor run simulations with a history register 9 bits and a history register of 11 bits (varying the size of the PHT accordingly). What do your results tell you?
  - For the *2lev* predictor run a simulation with the "xor" option on (with the same configuration for all other parameters as the original one). What do your results tell you?