
Your assignment is to evaluate the impact of branch prediction. You will again run the familiar program of sorting for quicksort, mergesort and bubblesort. You can treat the branch predictor as

Download:
http://cs.nyu.edu/courses/fall11/CSCI-GA.2233-001/labs/lab3/msort.cpp

Compile as follows:
sslittle-na-sstrix-gcc -O3 msort.cpp -o msort

The simulation programs in this lab used will be sim-profile and sim-outorder. These programs are discussed in the materials pointed at in Homework (09/21/2011 – Problem 5). If not please read up on it again (particular the presentation referenced).

Using sim-profile you first establish the distribution of instructions.

"./part1.sh 100" will give you for all three algos (q,m,b) the instruction mix.

NUM 100
load  store  uncond  cond  integer  float  trap
q: 15.61  21.83   6.67  14.01  41.84  0.00  0.03
m: 16.05  20.08   4.38  13.23  46.23  0.00  0.02
b: 18.74  14.39   8.97  18.75  39.14  0.00  0.01

Look at the script and familiarize yourself with individual output of each simulation in particular the –iclass option of sim-profile that creates detailed outputs on the simulation and from which the above data is generated.

We now will investigate different branch predictors.
run the script: http://cs.nyu.edu/courses/fall11/CSCI-GA.2233-001/labs/lab3/part2.sh

"./part2.sh 100" will give you hit rates on various predictors and targets

Again look at the script and familiarize yourself with the individual output of each simulation in particular the –bpred option of sim-outorder that creates detailed outputs on the simulation and from which the above data is generated. We are only using sim-outorder to generate statistics on the branch prediction. Here is an example of what data is used to create the part2.sh output. You
should be able to follow that from the script. We ignore the jr_non_ras_rate as there are essentially no function calls.

bpred_comb.bpred_addr_rate 0.8880 # branch address-prediction rate (i.e., addr-hits/updates)
bpred_comb.bpred_dir_rate 0.8947 # branch direction-prediction rate (i.e., all-hits/updates)
bpred_comb.bpred_jr_rate 0.9665 # JR address-prediction rate (i.e., JR addr-hits/JRs seen)

These three rates correspond to <addr> <dir> and <jr>. Branch prediction until described in more detail in the class can be treated as a black box.

A branch target buffer looks up the executing branch instruction address in the branch target buffer (BTB) and identifies the predicted address. A branch direction Table is indexed by the address of the conditional branch and predicts taken or not taken. If taken, the next predicted instruction to execute will be taken from the BTB otherwise its the next instruction (PC+4).
When a branch is executed, the BTB is updated (PC, branch target). When no entry is found in the BTB we assume a garbage prediction which reflect itself in the <addr> rate.
<dir> reflects the correct prediction rate of the whether the branch was taken or not.
JR prediction is another predictor that for jmp-register instructions.
Through experimentation of a small program you shall write, figure out how these counters correlate to each other and detangle their prediction rates by running sim-outorder and examining the output. If you had to write such a program, attach your simple program(s) to your report and with the logic explaining how you got to your conclusion.

Build a spreadsheet to integrate the part1.sh and part2.sh in order to create a CPI calculation. We assume a simple in-order model.

Assume that the AMAT of load instructions is 3 cycles, store instructions is 2 cycles and that on average uncond branches take 2 cycles and conditional branches take 4 cycles if correctly predicted and 10 cycles if not correctly predicted. Traps take 1 cycle (not really true).
Also integrate into your spreadsheet the “perfect” branch predictor.

When you are ready with your spreadsheet and CPI calculation, repeat the experiment for 300 500 700 and 900 (i.e. rerun part1.sh and part2.sh with those arguments) and integrate into your spreadsheet. Write a report with graphical representation (graph) on the relative impact on CPI of these branch predictors wrt to the perfect predictor. Your typed report should clearly describe the formula you are using for calculating the CPI.

In an additional report calculate for the NUM=500 case what will happen if the BTB prediction accuracy deteriorates to 85,80,75,70,50 percent, while branch direction prediction stays constant. What will happen if the BTB prediction remains as is, but the branch direction prediction deteriorates to 85,80,75,70,50. Utilize your spreadsheet for this.