HY425

Machine Assignment 1

Assignment: 4/3/2008

Due Date: 24/3/2008

Instructions: For this assignment you will need to submit a tarball with your branch predictor modeling program, and a well-written report (in .pdf) with your answers to the questions given in the assignment, including your qualitative conclusions. Send your tarball to Stamatis Kavadias (kavadias@ics.forth.gr), with a copy to the instructor (dsn@ics.forth.gr). Use the following subject in your e-mail: **HY425: Machine Assignment 1 Submission**. Please the aforementioned subject only, so that your homework is read and graded. Last, but not least, **start early!**

Assignment

Write a C or Java program to model a (m,n) branch predictor. Your program will read a series of lines from a file named history.txt, which is available through the course web site at: http://www.csd.uoc.gr/~hy425/projects/history.txt. We are providing you with a template of a C program which implements a (1,2) predictor, at: http://www.csd.uoc.gr/~hy425/projects/ph4e_branch.c and a sample of the output of this program at: http://www.csd.uoc.gr/~hy425/projects/ph4e_out.txt. You will need to generalize and parameterize the branch predictor model in the program, to model a (m,n) predictor, where m and n are program inputs. Note that for the second parameter (n), you are required to write the code for a 2^n-state model of prediction.

Each line in the history file has three data items, separated by tabs. The first datum of each line is the address of the branch instruction in hex. The second datum is the branch target address in hex. The third datum is a 0 or 1; 1 indicates a taken branch, 0 a not-taken branch. The total number of branches your model will consider is, of course, equal to the number of lines in the file. Assume a direct-mapped Branch Target Buffer (BTB), and don't worry about instruction lengths and alignment (i.e., if your BTB has four entries, then branch instructions 0x0, 0x1, 0x2 and 0x3 will reside in those four entries, but a branch instruction 0x4 will overwrite BTB[0]). For each line in the input file, your model will read the pair of data values, adjust the various tables per the branch predictor being modeled, and collect key performance statistics. The final output of your program should look as follows:

```
0x40074cdb    0x40074cdf    1
0x40074ce2    0x40078d12    0
0x4009a247    0x4009a2bb    0
0x4009a259    0x4009a2c8    0
0x4009a267    0x4009a2ac    1
0x4009a2b4    0x4009a2ac    1
...
```

The first address is the address of the branch, whereas the second address is the address of the branch target. The last bit indicates if the branch is not taken (0) or taken (1).

Make the number of entries of the Branch Target Buffer a command-line option. Make also, the number of bits of global history, and the number of bits of per-branch history a command-line option.

Part 1 (20%)

Write a (m,n) model of a branch predictor with branch target buffer with 64 entries. What is the overall hit rate of the BTB (the fraction of times a branch was looked up in the BTB and found present)? Note that this should be independent of (m,n).

Part 2 (40%)

Experiment with m=1,2,3 and n=1,2,3 and measure branch misprediction rates. Plot a chart with these rates. What conclusions can you draw from this experiment?

Part 3 (20%)

Investigate the difference between a cold-start predictor and a warm-start predictor. In the cold-start predictor, initially, the branch predictor tables have no information about the branches of the program (i.e. they are «empty»). In a warm-start predictor, you run the program once. In the end of the run, the branch predictor is loaded with the history of several branches executed by the program. You run the program again, but this time you assume that the branch predictor starts «warm», with the information of the branches of the previous execution. Does a warm-start make a difference in the performance of the branch predictor? To answer this question, pick the best predictor from your previous experiment.

Part 4 (20%)

Perform a sensitivity study of the predictor, where you change the size of the BTB and the BHT between 16, 32, 64, and 128. What conclusions do you draw from this experiment? Use a chart of misprediction rates with different BTB/BHT sizes, for comparisons.