ECEC 621
Simulation Project 2 – Branch Predictor Implementation

Due Tuesday, March 10thth, 12:00am

1 Branch Predictor Implementation

In this part of the project, you will implement your own branch predictor. The ∼/gem5/src/cpu/pred directory contains the code for all branch prediction classes in the simulator. This directory contains all the files for the branch predictors that you read and analysed in Part 1. Two files: ECEC621_pred.cc,hh describe an empty class derived from the base branch predictor class.

1.1 Predictors

Choose one of the following branch predictors and implement the lookup() and update() functions in ECEC621_pred.cc. Refer to tournament.cc and 2bit_local.cc to see how the other predictors were implemented. You may need to add configuration parameters into the constructor (or set values statically). Configuration is set in BranchPredictor.py.

1. Alloyed Predictor (Skadron, Martonosi & Clark: http://www.cs.virginia.edu/~skadron/Papers/alloy_p
2. 2-level adaptive predictor (Yeh & Patt: http://courses.engr.illinois.edu/ece512/Papers/Yeh.1991.M
3. G-share (described here: http://www.hpl.hp.com/techreports/Compaq-DEC/WRL-TN-36.pdf)
4. Other predictor: If you know of another predictor you want to implement, cite the paper and build an implementation.

1.2 Procedure

1. Edit src/cpu/pred/BranchPredictor.py, setting the predictor type = "ECEC621"
2. Rebuild the processor simulator by executing the following from your ∼/gem5 directory
   `[rb639@xunil-00 gem5] scons build/X86/gem5.opt`
3. Simulate the same two benchmarks you chose in part 1:
   `[rb639@xunil-00 gem5] ./build/X86/gem5.opt config/example/ECEC621.py -b $BENCHMARK`
4. Record the following statistics from the output file:

   IPC (system.cpu.ipc)

   Execution cycles (system.cpu.numCycles)

   Branch rate (system.cpu.fetch.branchRate)

   Branch mis-prediction rate ($\frac{\texttt{system.cpu.branchPred.condPredicted}}{\texttt{system.cpu.branchPred.condIncorrect}}$)

   % fetch-squashed ($\frac{\texttt{system.cpu.fetch.SquashCycles}}{\texttt{system.cpu.fetch.Cycles}}$)

1.3 Report

Write a report summarizing the design and results for your predictor. Your report should include *at least* the following sections: introduction, experimental setup, results and analysis, conclusion. Include schematics or diagrams showing how your branch predictor works, and answer the following in your report:

1. Describe the predictor you chose to implement (include the citation for the paper)
2. How large is the predictor in terms of storage?
3. How does the predictor work (include a diagram)
4. How did you implement the predictor (i.e. how does your code work?)
5. Compare the performance with the predictors from Part 1