

Lab 2 – Due May 19th

In this lab, you will expand the sim-safe simulator you have used in the first lab project. Implement the following in sim-safe.c and turn in the hardcopy of the results by Tuesday, May 9th.

- The conditional branch hit rate with a 1-bit branch predictor.
- The conditional branch hit rate with a 2-bit branch predictor. For this predictor, use the MSB as the prediction (i.e. 1 = Taken, 0 = Not Taken). Decrement the 2-bit counter by 1 when the branch is not taken, increment the counter by 1 when the branch is taken.
- The conditional branch hit rate with a 3-bit pattern-based prediction. In this case, use only the global branch history register to make the prediction.
- The conditional branch hit rate with a 3-bit pattern-based prediction. In this case, implement a history register for each branch instruction (i.e., keep track of the last three execution of each branch instruction) and make your predictions accordingly.

Of course - you should use the predictor to make a prediction about an instance of a branch BEFORE updating the predictor based on the correct branch direction.

Use a 1024 entry table for the 1-bit and 2-bit predictors as well as the 3-bit pattern-based predictor with individual history registers.

Run sim-safe on the following benchmarks for 100 million instructions (set this with -max:inst):

- crafty
- gcc
- vortex
- gap

The input parameters to the applications are:

- crafty00.peak.ev6 < crafty.in > /dev/null
- gcc00.peak.ev6 expr.i -o expr.s > /dev/null
- vortex00.peak.ev6 lendian2.raw > /dev/null
- gap00.peak.ev6 -l ./all -q -m 128M < ref.in > /dev/null

Note that these are the applications and the input sets you have used for the first lab project.