Simulation project Due: April 24

For the simulation project, you have the choice of simulating either Branch Prediction or Cache Coherence protocols:

Branch Prediction

If you choose to simulate branch prediction, you must choose one of two methods:

Method 1: Implement your own simulation in the language of your choice. You must simulate the following two branch prediction mechanisms:
* GAg global two-level prediction
* Perceptron

You may implement your simulation in any programming language. Your simulation framework must read in a sequence of branches with the syntax:

address_of_branch taken/not_taken

Ex:
0xA0F4 T
0xA0BC NT
…

You must create the sequence yourself. The sequence must contain at least 150 unique branches and a total of at least 10,000 executed branches (some branches will be executed multiple times during the program). Suggestions for creating the sequence is to write a program to output a sequence of branches or use a LLM to create the sequence.

Your simulator must output the total number of branches, the number of branches correctly predicted and the number of branches mispredicted. Your writeup should compare your results for GAg and Perceptron predictors. You should simulate a global history of 14 bits for the GAg predictor and a global history of 38 bits for the Perceptron. Your perceptron predictor should implement 128 perceptrons.

Method 2: You can extend the Branch Prediction Championship framework found at https://ericrotenberg.wordpress.ncsu.edu/cbp2025-simulator-framework/

This framework is written in C++. If you use this Method 2 you only need to implement the Perceptron predictor. You will use the instruction trace included with the Championship Branch Predictor framework code. You will also use one training trace that you select from the 105 training traces that are provided. You may choose any of the training traces that you like. You should utilize a global history of 38 bits for the Perceptron. Your perceptron predictor should use a total of 128 perceptrons.

Your writeup should evaluate the results of the TAGE predictor already implemented in the framework and the Perceptron predictor you you implement.

Cache Coherence

If you instead choose to evaluate cache coherence protocols, you must simulate the following two cache coherence protocols:

* MESI Invalidation Protocol
* Dragon Update Protocol

You may implement your simulation in any programming language. Your simulation framework must read in a sequence of reads and writes performed by 4 processors with the syntax:

P1 read
P2 read
P3 write
P1 write
P4 read
P2 write
…

Your simulator must output the number of each type of Bus Transaction that is required for the given sequence. Two sequences are provided for evaluation on Canvas. Your writeup should compare the numbers and type of Bus Transactions that are needed for each of the two protocols and for each of the two sequences.