University of California, Riverside

Department of Computer Science & Engineering

Title: Advanced Computer Architecture Lab 2

Student Name:

Mahbod Afarin

Student ID:

862186340

Winter 2020

1. Assuming a simple 2-bit branch predictor that uses 8 bits from the PC to index into the BHT, how many entries does the BHT have? What is the size of the BHT (in bytes)?

$$Entries\ of\ BHT = 2^8 = 256$$

$$Size\ of\ BHT = 256 \times 2 = 512\ bits = 64\ bytes$$

2. Assuming a (4,2) predictor, how many total entries does this predictor have? What is the total size of this correlating predictor (in bits)?

$$Number\ of\ Entries = 2^8 \times 2^4 = 4096$$

$$Size = 4096 \times 2 = 8192\ bits$$

3- What is the misprediction rate of the given traces and predictor?

Trace	(0,1)	(0,2)	(6,1)	(6,2)
gcc-10K.txt	30.58%	27.95%	18.81%	26.74%
gcc-8M.txt	31.23%	27.28%	11.61%	7.72%

4- Is every entry in our branch predictor utilized? For our (6,1) predictor, how many entries are utilized?

No, our benchmarks did not utilize every entry in our branch predictor. For 10K benchmark it is 2571 and for 8M benchmark it is 8241.

5- How does the global branch history help improve branch prediction rates? Will all applications benefit from using global branch history?

Global branch history records last m branched to select between 2^m branch history tables. It improves the branch prediction rate by recording the history of the m most recently executed branches and in this way, it uses the global history of the branches to improve the prediction rate.

No, for some applications global history works better and for some application local history works better. Therefore, we cannot say that all application will benefit from global branch history. That lead to the idea of tournament predictor; tournament predictor has both local and global history together and based on their results, it decided which history is better and it will choose that history.

6- What is a local predictor? How does it help with branch prediction? Can it be combined with the predictors that we implemented in lab 2?

We can have a local history for every specific branch. For implementing that, we can use some of the program counter bits for showing the entry of every specific branch. For example, in figure 1, we used 4 bits of program counter for indexing the 10-bit local branch history and then we use that for indexing the 1k entries of 2-bit counters.

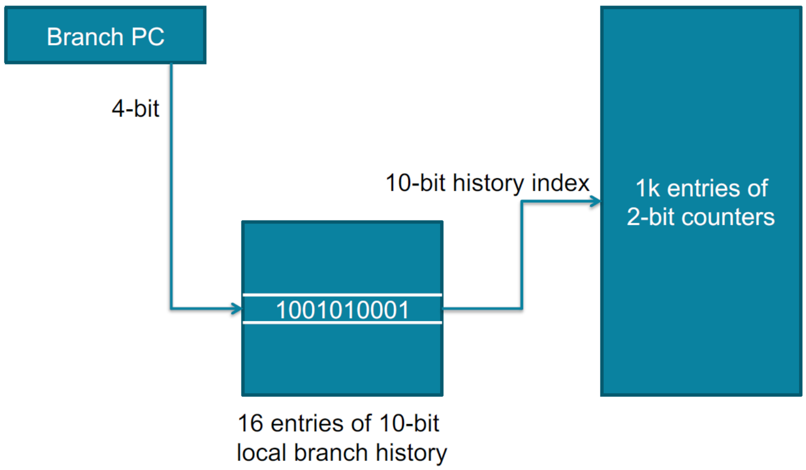

Figure 1: Local branch predictor

But the previous design has aliasing problem because it is possible to have various PC with the same history. As it has shown in figure 2, for removing the aliasing problem we use XOR to encode our information and our information will be encoded based on the program counter's bits and history bits. So, if various PCs has the same history, we do not have collision.

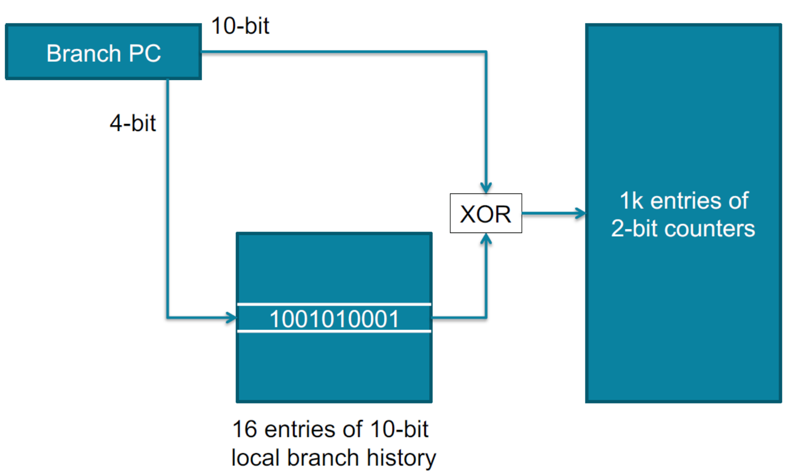

Figure 2: Local predictor with no collision problem

We can combine local predictor with the predictors that we implemented in Lab2 because some branches work well with local predictor while other branches work well with global predictor. So we can combine the 2 predictor and call it Tournament Predictor. As it has shown in figure 3, we have a multiplexer with 2 predictors and there are a Tournament unit and based on the program we can select which predictor is better to use.

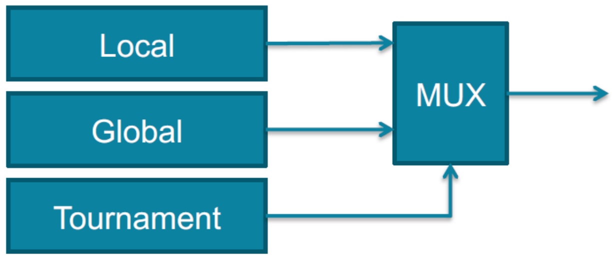

Figure 3: Tournament Predictor

The tournament unit select predictors based on the learning in the execution time. As it has shown in the figure 4, we have 4 states and we will switch between states based on accuracy of the predictors.

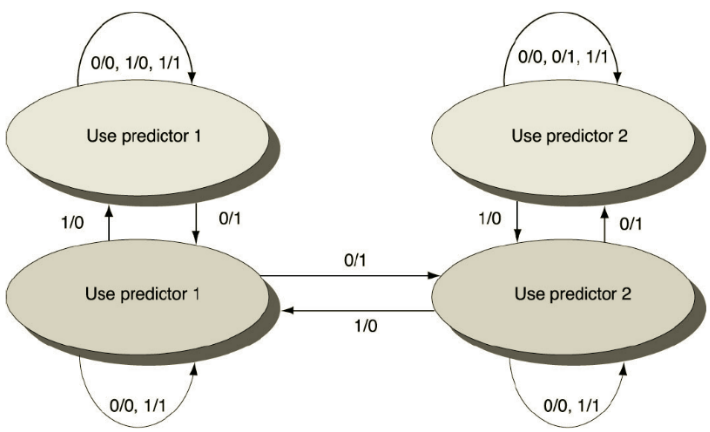

Figure 4: States of the Tournament Predictor