

CS422: Computer Architecture | PA2

1. AVERAGE MPPKI (BI-MODE[1]): 7.1196

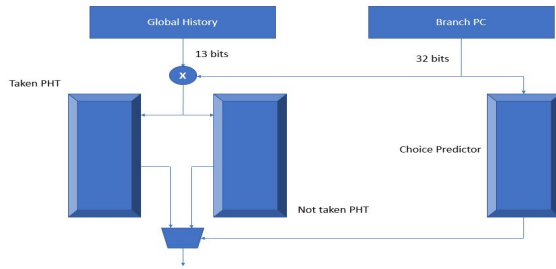


Figure 1: Bi-Mode predictor

This predictor operates similar to gshare predictor but with a twist of biasness. Here we divide the PHT into two different PHTs, i.e., Taken PHT and Not taken PHT. The choice predictor consists of 2 bit counters which decide which PHT to access. This division helps in decreasing the aliasing in the PHT. It helps to separate likely-to-be-taken branches from likely-to-be-not-taken branches.

- One entry of one PHT is updated on every branch outcome.
- Choice predictor is updated on every branch outcome except for the following cases:
 - Branch is taken, Not taken PHT is chosen and the prediction made is “taken”.
 - Branch is not taken, Taken PHT is chosen and the prediction made is “not taken”.

1.1 Hardware Cost

- Taken PHT and Not Taken PHT: 2^{13} entries each, each entry being a 2-bit counter. Total = $2^{13} \times 2 \times 2 = 2^{15}$ bits
- Choice Predictor = $2^{14} + 249$ entries, each entry a 2-bit counter. Total = $2^{15} + 498$ bits
- GHR: 13 bits

Total size = $2^{16} + 511$ bits

2. AVERAGE MPPKI (TOURNAMENT + BIAS): 7.6936

Tournament predictor consists of two predictors (Sag and Gshare) and a meta predictor to choose the best from them.

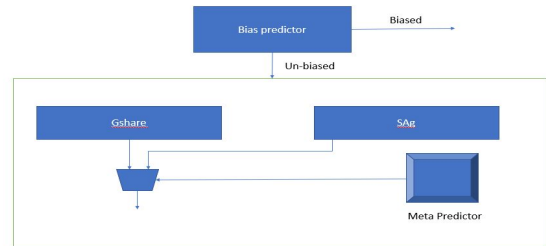


Figure 2: Tournament + Bias predictor

This tournament predictor is preceded by a bias predictor which checks if the branch is biased or not. If the branch is biased the branch is predicted directly according to the bias else it is predicted by the tournament predictor.

2.1 Hardware Cost

- Gshare 13 bit GHR, 2^{13} entry PHT each of 2 bits = $2^{14} + 13$
- Sag 6 bit history, 2^{12} sets in level 1 (GHT), 3 bit counter on level 2 (PHT) = $2^{12} \times 6 + 2^6 \times 3$
- Meta Predictor 2^{13} entries 2 bits each = 2^{14}
- Bias Predictor $2^{12} + 150$ entries of 2 bits each = $2^{13} + 300$

Total = $2^{16} + 505$

- Minor tweaks: Initialization of various counters change in Gshare¹.

3. AVERAGE MPPKI (GSHARE (BASELINE)): 8.2544

4. DIVISION OF LABOUR

50-50

5. REFERENCES

- [1] Chih-Chieh Lee, I-Cheng K. Chen, and Trevor N. Mudge, “The Bi-Mode Branch Predictor,” Proceedings of 30th Annual International Symposium on Microarchitecture, Research Triangle Park, NC, 1997, pp. 4-13.

¹Note: We have slightly modified the branch_record_c class to add one variable to account to predicted result (to be later compared with actual result).

Although this can be avoided by some workaround in the predictor.h code itself, but the former just seemed easier.