

# CS6600: COMPUTER ARCHITECTURE

SEMESTER: JUL-NOV 2024

---

## Assignment 2 Report

Analysis of Branch Predictors

---

*Name:* Snehadeep Gayen

*Roll:* CS21B078

*Submission Date:* 23rd Oct 2024

# 1 Project Description

This project simulates and analyses bimodal and gshare branch predictors on various benchmark programs.

## 2 Results

### 2.1 Bimodal Predictor

(a)

The adjoining plot depicts the effect of number of bits, ( $m$ ), used for prediction, on the misprediction rate for two different traces.

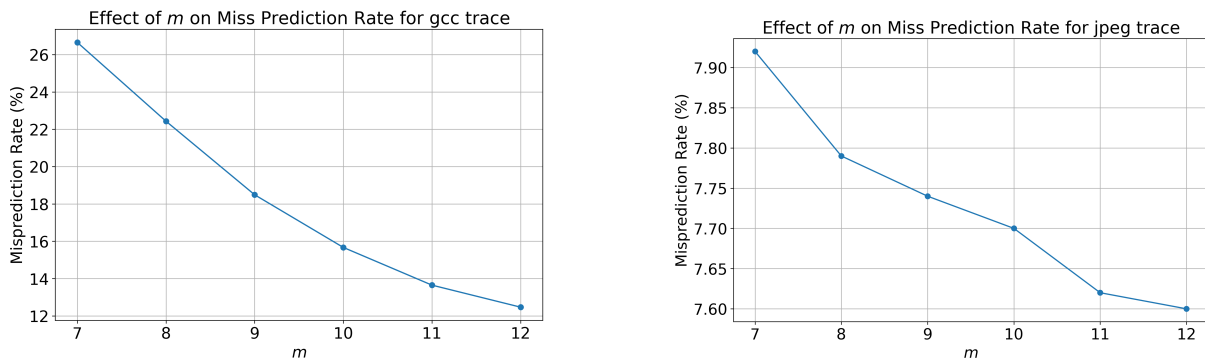


Figure 1: Miss prediction rate for varying number of prediction bits  $m$

Some general observations and similarities between the plots is explained below:

- Clearly the misprediction rate decreases as we increase the number of bits  $m$  used for prediction. With lesser  $m$ , there are collisions that happen among different branches having identical last  $m + 2$  bits. This mixes and garbles up the counters, leading to higher mispredictions. Having more number of bits decreases the number of collisions and thus, the misprediction rate.
- Clearly, there is a case of diminishing returns for both the graphs. In the left figure, we see that initially, increasing  $m$  by 2 decreases the miss prediction rate by 4%. However, at higher  $m$ , the same increase causes a decrease of less than 2% for miss prediction rate. Similarly, miss prediction decrease falls from 0.1% initially to about 0.02% at higher  $m$ .

Some differences between the two plots are:

- Firstly, it is noteworthy that the `gcc` trace has misprediction rate in the range of 10-30%, while the `jpeg` trace, has only 7-8% mispredictions. This means that the `gcc` trace inherently has more *unpredictable* branches compared to `jpeg` for the same number of prediction bits used for bimodal predictor.
- Secondly, the fractional decrease in the misprediction rate, by increasing  $m$ , is much higher in `gcc` (around 15%) than in `jpeg` (5-6%). This proves that despite having more unpredictable branches than `jpeg`, the improvement in misprediction rate with increasing  $m$  is much more for `gcc`.

(b)

Given  $m$  number of bits used for indexing into the predictor table, number of storage bits required for the predictor table is  $2^{m+2}$ . This is because each entry has to be a 2-bit counter. Thus, with 16KB budget, one can have  $m$  as atmost 16. But from the graphs we see that the improvement from  $m = 11$  to  $m = 12$  is not appreciable. However in the case of `jpeg` trace the increase in accuracy from 10 to 11 is sizeable. Further, with  $m = 11$  only 1KB of memory is used. Thus,  **$m = 11$**  is a reasonable choice balancing misprediction rate, storage and power.

## 2.2 Gshare Predictor

(a)

The figure below shows the misprediction rates of the Gshare branch predictor for different values of  $m$  (lookup bits from the program counter address) and  $n$  (global branch history register).

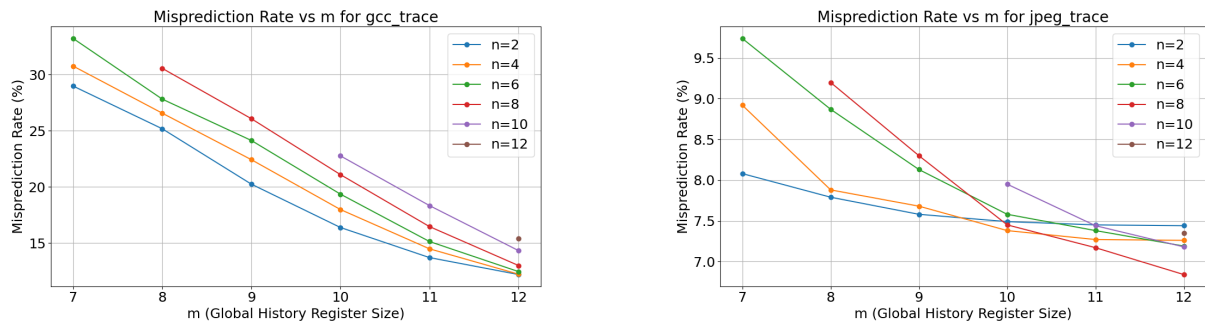


Figure 2: Miss prediction rate for varying number of prediction bits  $m$  and global branch register size  $n$

Some general observations, similarities and differences from the above plots are described below:

- For a given size  $n$  of the global branch history register, we see that the misprediction rate decreases if we increase the number of prediction bits  $m$  considered from the address. As explained in the previous plot, this is because of the reduced number of address collisions in the branch prediction buffer.
- For a given  $m$ , it is expected that increasing the global history register size  $m$ , decreases the misprediction rate by better utilising the global branch history. While this trend is clearly false in the case of `gcc` trace, it somewhat holds for `jpeg` trace. This is probably because, the initial setup of the global branch history takes time for larger values of  $n$ . Thus, there may be a lot of initial mispredictions. This effect would probably decrease with a larger trace.

(b)

One important observation here is that, for Gshare predictor, increasing  $n$  does not increase the storage required for the predictor table. Further, it has negligible increase on the power also. Thus, a larger value of  $n$  is preferred. From the figure we observe that  $m = 11$  gives a reasonable misprediction rate. Any lesser value of  $m$  would increase the mispredictions by atleast 35% and an increase in  $m$  for `jpeg` is observed to, unintuitively, increase the mispredictions. For this  $m$ , considering both the plots we observe that  $n = 4$  is a good choice as it has second best misprediction rate for both the traces. Thus,  $\mathbf{m = 11}$  and  $\mathbf{n = 4}$ , is the preferred choice. This choice has reasonable storage requirement of  $1KB$  and achieves a misprediction rate of about 14% and 7.25% for the `gcc` and `jpeg` traces respectively.