NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #2: Branch Prediction

by

SRI HARSHA TEJ PATNALA

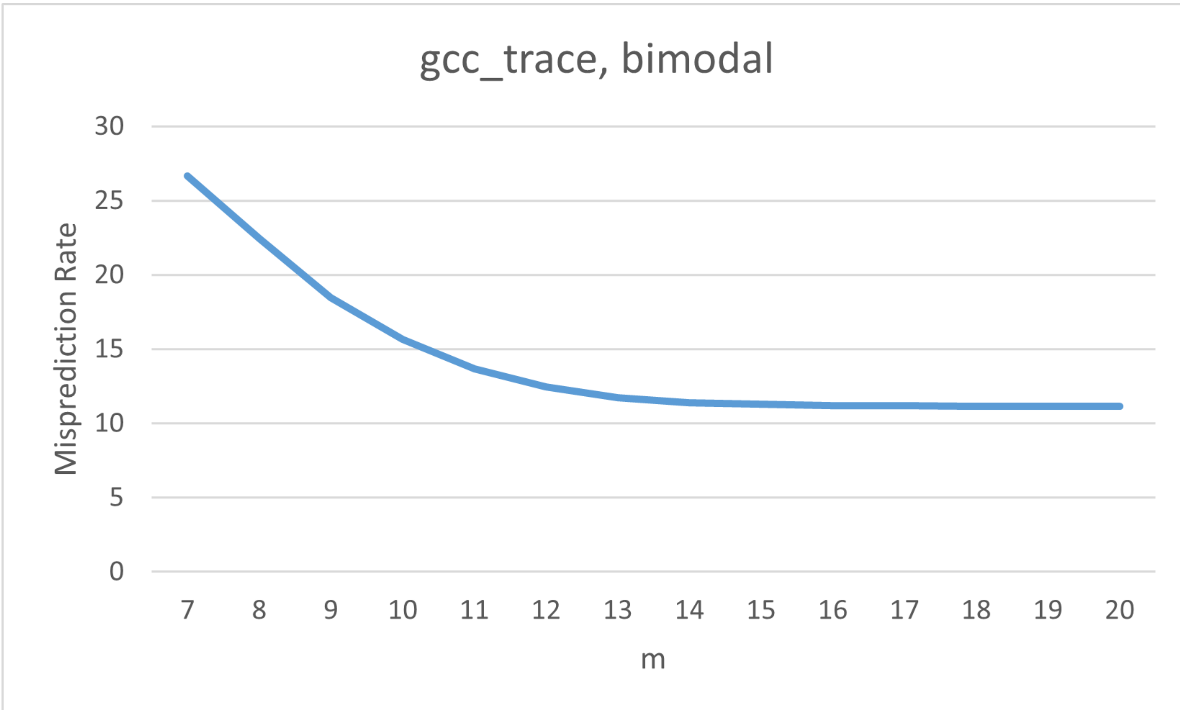
NCSU Honor Pledge: "I have neither given nor received unauthorized aid on this project."

Student's electronic signature: ___________Sri Harsha Patnala____________________
(Sign by typing your name)

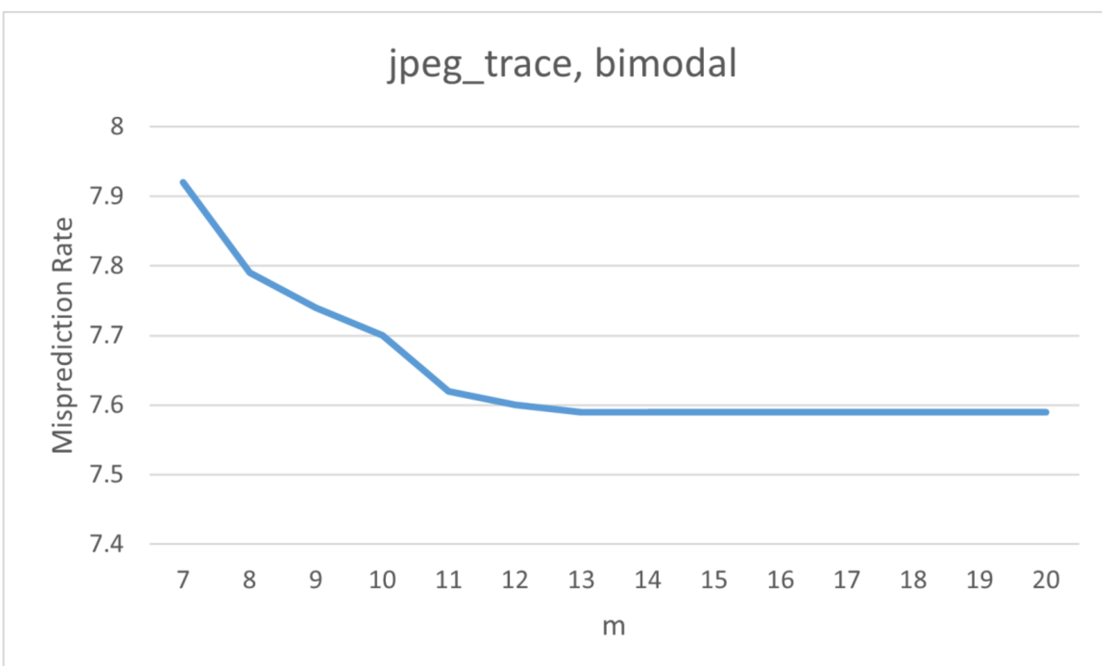
Course number: _563____________
(463 or 563?)

Part 1: Bimodal Predictor

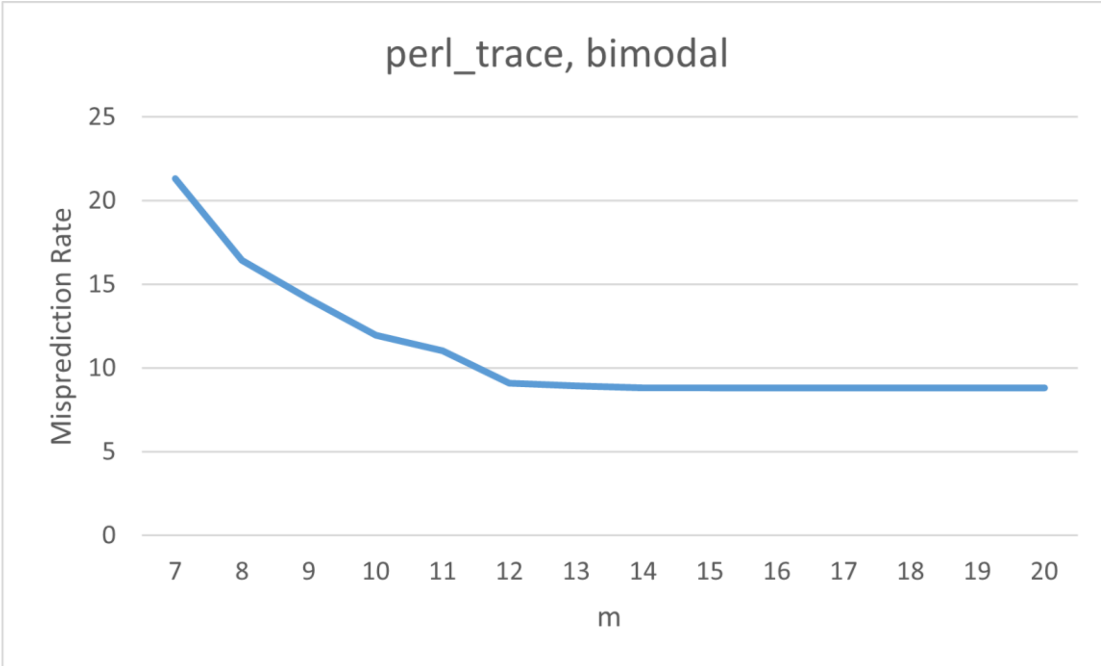
Graph for gcc trace



Graph for jpeg trace



Graph for Perl trace



From the graphs above, we can see that for all the benchmark traces, the misprediction rate gradually decreases as we increase the number of m bits. And eventually, the misprediction rate saturates at around 13 m bits.

This is as expected because, as we increase the number of bits that are indexed into the BTB, we have a larger BTB. And hence having a larger BTB means that more unique PC address can map into different locations, thereby reducing the conflicts in the BTB.

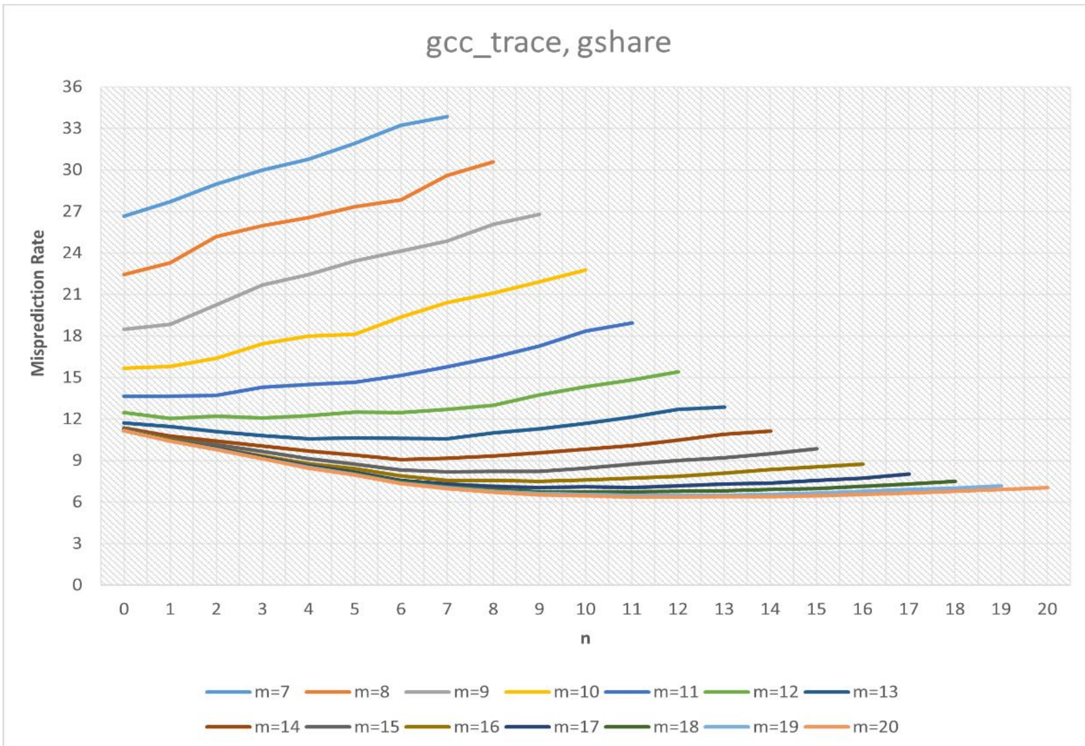
So, for more BTB locations, we don't have to blindly take the prediction for PC values that map to the same location of the BTB and avoid updating the branch counters unnecessarily.

Looking the graphs for different traces, we observe that for the jpeg and Perl trace, the outcomes are quite similar with a few roughs at certain m bits. But for the gcc trace, the curve tends to be a lot smoother and appears that the misprediction rate decreases exponentially, eventually saturating at a certain m bit value.

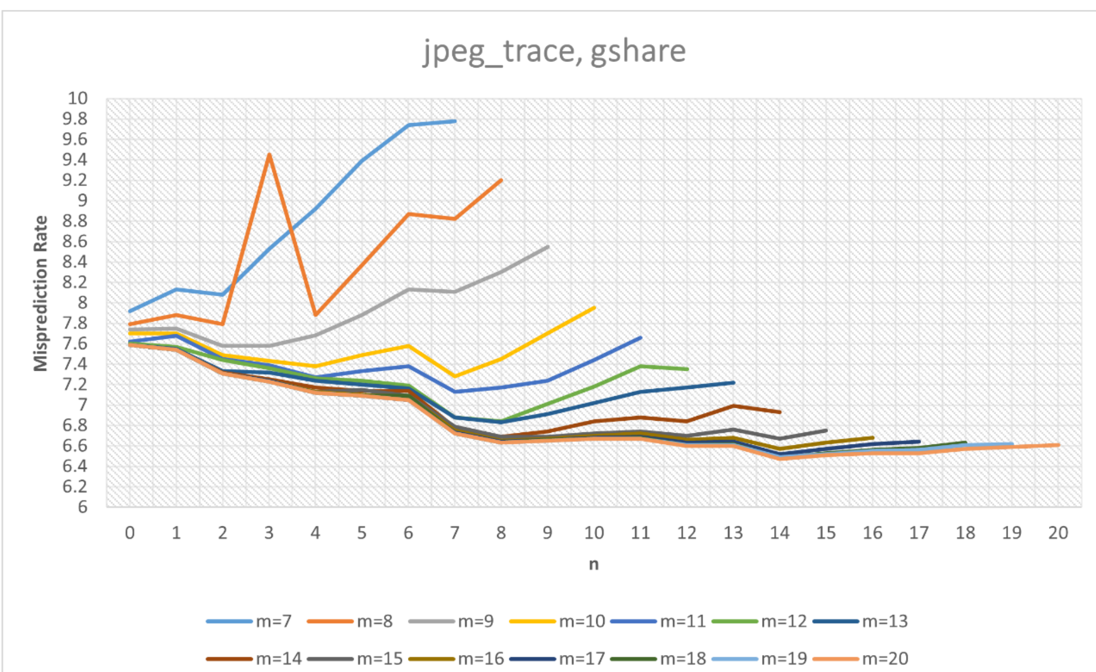
Also, the trends in the trace could also affect the misprediction rate. Since bimodal predictors are best when a branch largely biased towards not taken or taken, having a lesser number of bits to index an entry could possibly introduce conflicts for other branches that map to same entry and could ruin the biased trends observed for the previous branch PC.

Part 2: Gshare Predictor

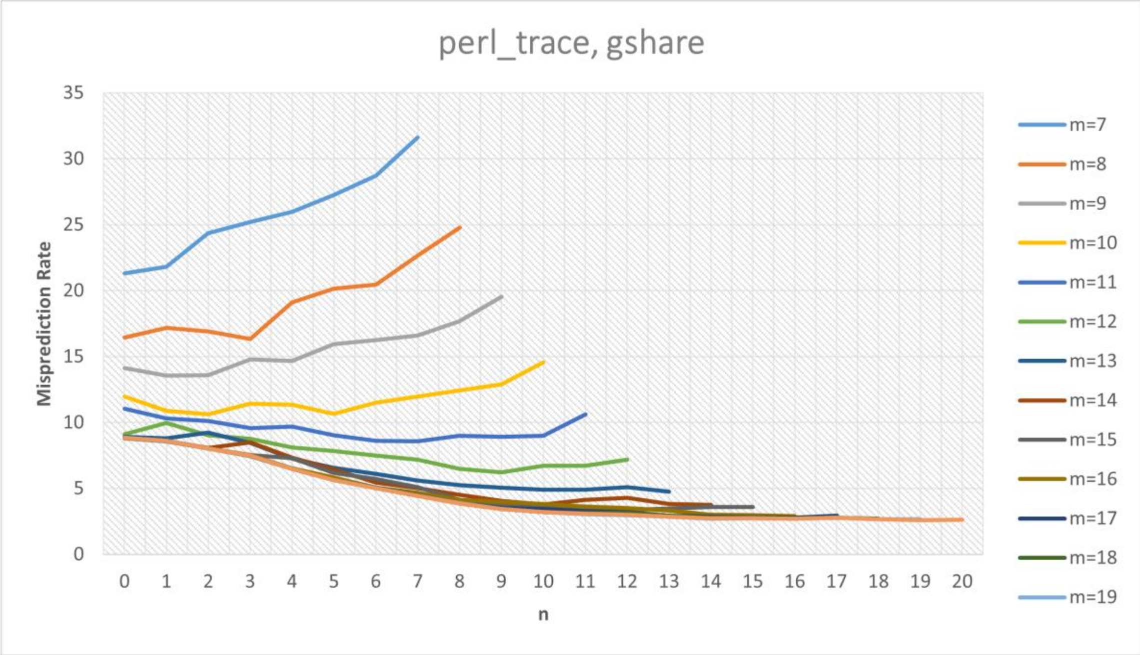
Graph for gcc trace



Graph for jpeg trace



Graph for Perl trace



Based on the graphs above, we can see that for fewer number of m bits, increasing the number of n bits has increased the misprediction rate and thus the performance. But for a larger number of m bits, increasing the number of n bits has improved the misprediction rate and thus improved the performance.

Since a gshare predictor uses both the global history register bits and the lower order bits from the PC to index an entry, it means that we are looking at the history to predict the branch based on a trend observed so far. So, if there are fewer number of m bits, then this would inhibit the record of past trends, as we have fewer global history register bits that are XORed with the PC bits, and thereby reducing the chances to correctly predict the branch. And that is visible from the graph.

However, even though there are a greater number of n bits i.e., the number of PC bits, we still are using fewer bits from the GBHR to index an entry, implying that we have not much information about the past. So that could increase the misprediction rate. For larger number of m bits, we have enough information about the past trends, and thus increasing the number of n bits could reduce the contention where two branch instructions could map to same entry.

Overall, the gcc trace has the highest misprediction rates compared to other traces, and jpeg trace has an overall best misprediction rates among the traces. This also indicates that how a predictor's performance is largely dependent on the trace, as an occurrence of a particular branch many times, improves the training for future predictions. So, there could be possibility in the jpeg and perl traces that a branch has occurred quite often which proved to provide sufficient training for the predictor, to accurately predict outcome.