Aaron Etheridge

8009326351 aetheri2@uncc.edu

ECGR 4181 Comp Arch

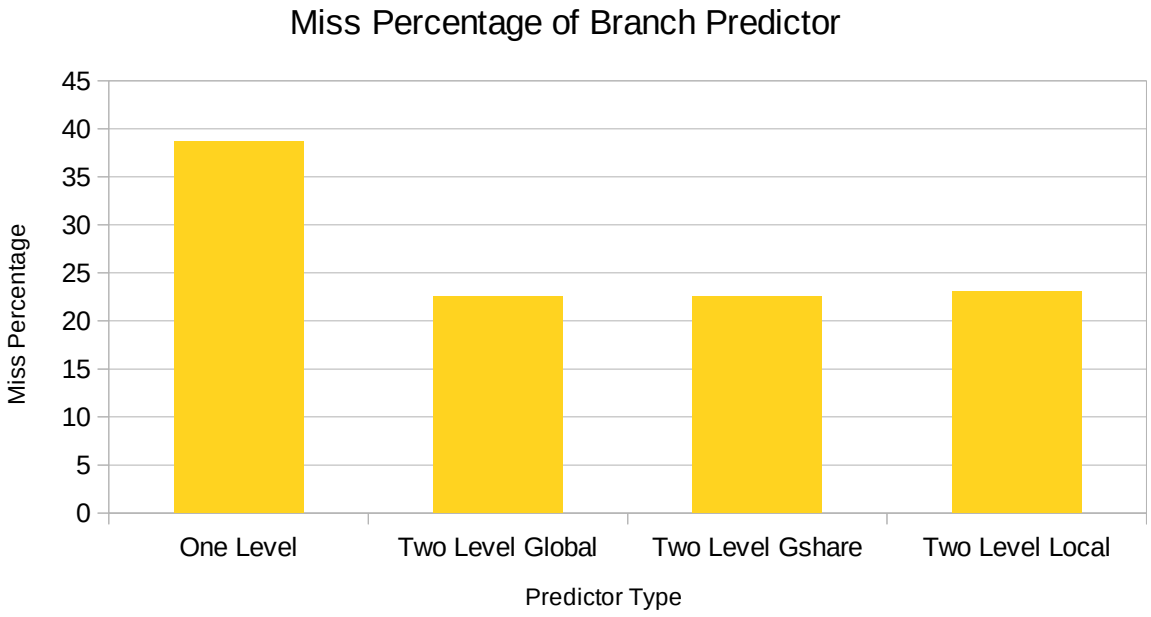
Home Work 4 Branch Predictors

IMPORTANT:

Please read the readme inside this file, there is an script to quickly compile all the codes.

Problem 1:

	2 Bits	4 Bits	6 Bits	8 Bits
One Level	38.6877			
Two level global	22.5672			
Gshare	22.5765	20.8707	22	23.8047
Two level Local	23.05	23.49	24.51	26.0873

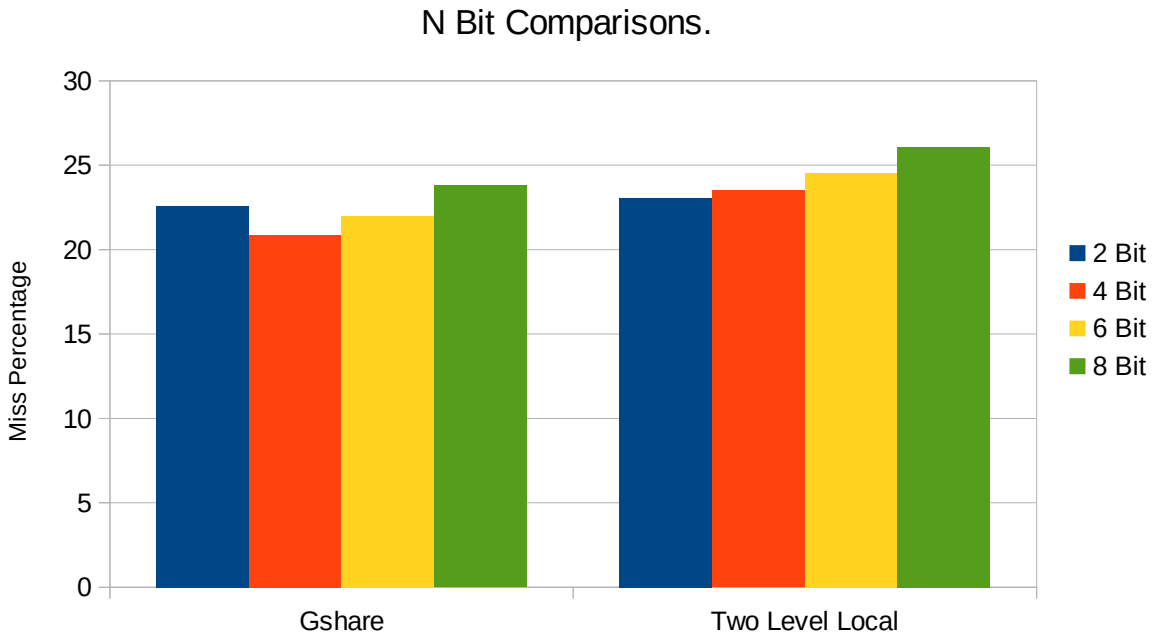


In the 2 bit saturating counter setting, all of the two level branch predictors outperformed the one level counter by a significant ammout. In my results with the provided trace file the two level global branch predictor only barley beat out gshare and the local branch predictors. Two Level local should have had the lowest miss rate but my results didn't show this. I believe this has to do with how I defined the part of the PC address that was used. I'm not entirly sure how to use the address because the parameters of

the address changed through out the trace file. This would have mad a significant difference how the tables mapped out, which would have improved performance.

Problem 2:

Gshare	22.5765	20.8707	22	23.8047
Two level Local	23.05	23.49	24.51	26.0873

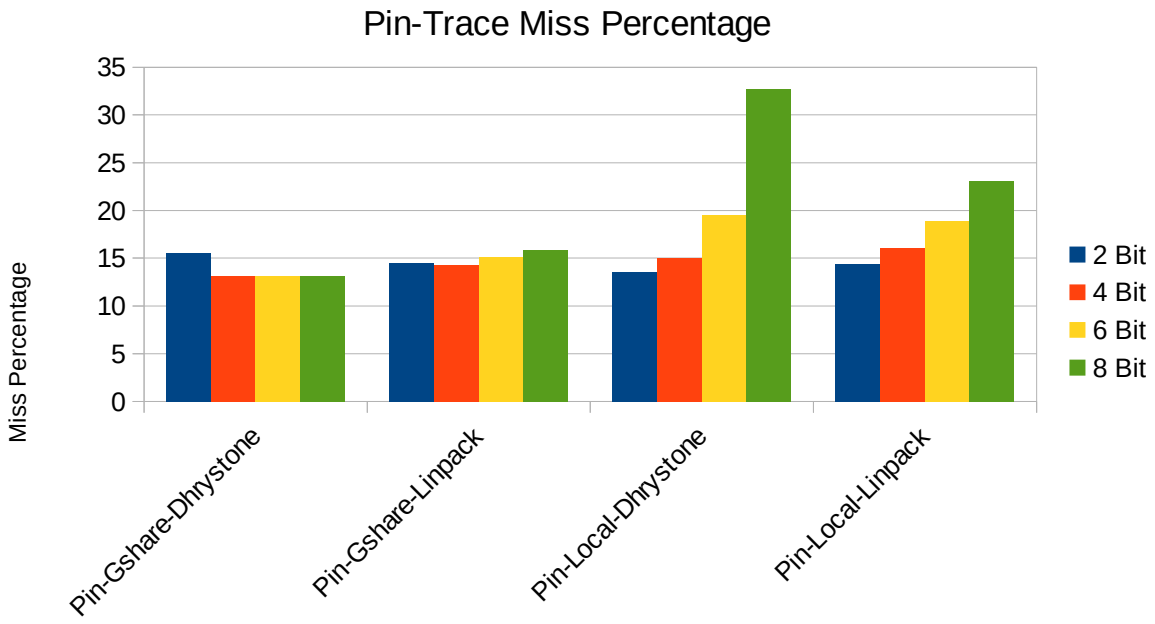


In my results gshare running with a 4 bit counter had the lowest miss rates. While for the two level local predictor, a 2 bit counter seemed to marginally outperform the 4 bit counter. The reason the larger bit counters didn't do well is because they have slow adjusting times. And cannot adapt to the predictions flip flopping quickly. The 2 bit and 4 bit counters are able to quickly change between taken and not take because they need far fewer cycles to change.

Problem 3

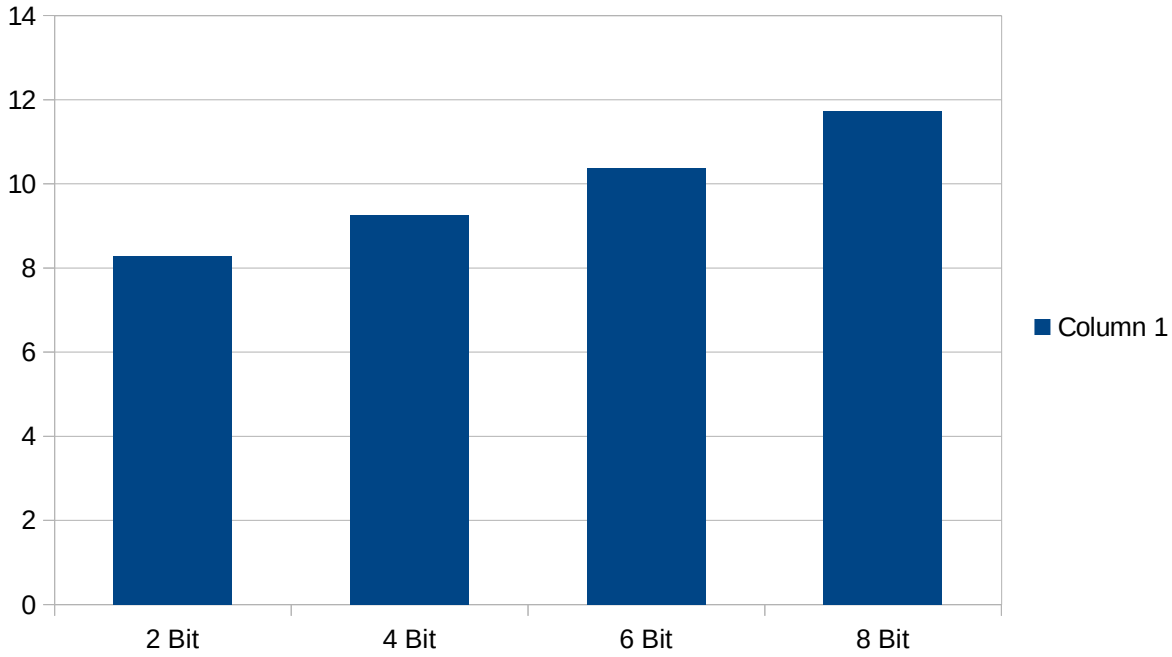
pin_gshare	2 bit	4 bit	6 bit	8 bit
dhrystone	15.54	13.17	13.17	13.17
Linpack	14.49	14.23	15.10	15.89

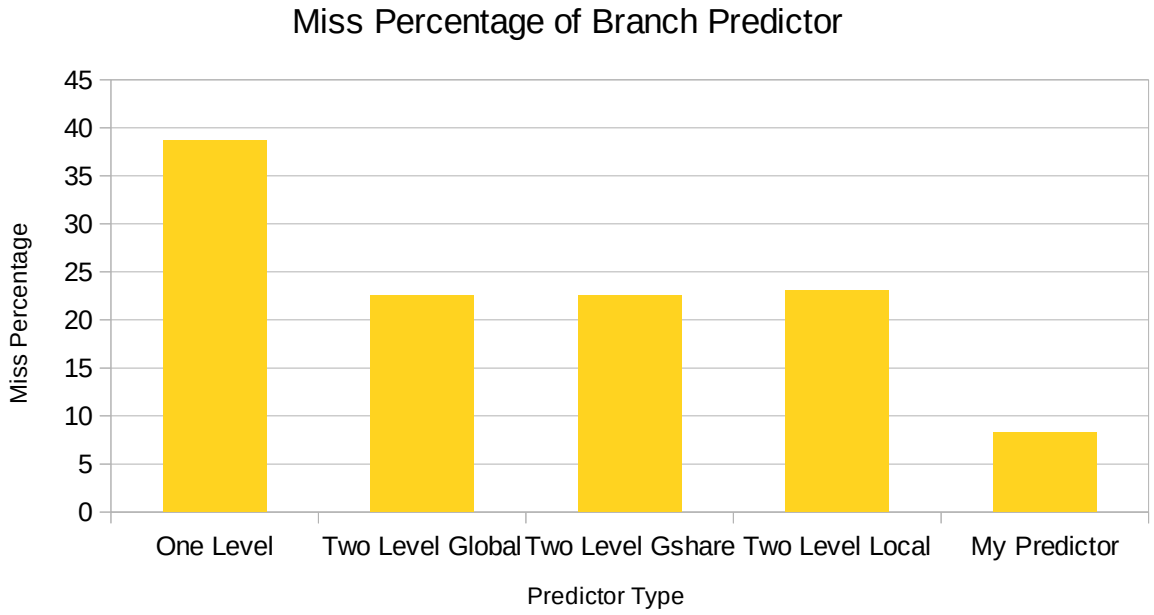
pin_local	2 Bit	4 Bitt	6 Bit	8 Bit
Dhrystone	13.59	15.04	19.55	32.67
Linpack	14.34	16.05	18.89	23.08



In my test traces for gshare, the higher bit counters did better but this is wrong. The lower bit amounts are supposed to out perform the larger bit counters. This is probably because of the way that I set up the PC in my gshare program. The input address changes a lot in the trace files. So what the PC correlates to changes making the pht slightly wrong. The two level local history tables performed as expected. The larger bits were supposed to be less accurate because the states are harder to change between taken and not taken. Based on my results I would have to say that the 2-bit counter would probably be the most optimal state, however I believe that 4 bit counters might be better suited to these particular tasks because they are able to adapt relatively quickly, but are less susceptible to random changes in the predictions.

Problem 4 Extra Credit





The branch predictor I created generated had about an 8.3% miss rate with a 2 bit saturating counter, which means it had about a 92% hit rate. It incorporated 4 different branch predictors and combined the outputs into a weighted table that would ultimately generate the final output. This predictor used aspects of both gshare and local level predictors but also incorporated other simpler predictors into the final method. Theses other predictors were incorporated in order to deter similar inputs from always aliasing to the same locations in the tables. My branch predictor outperformed both gshare and local level because it uses positive aspects from both of these predictors.