NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2021 (Rotenberg)

Project #2: Branch Prediction

by

PRAJAKTA KESHAVRAO JADHAV

Unity ID: pkjadhav

Student ID: 200375352

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

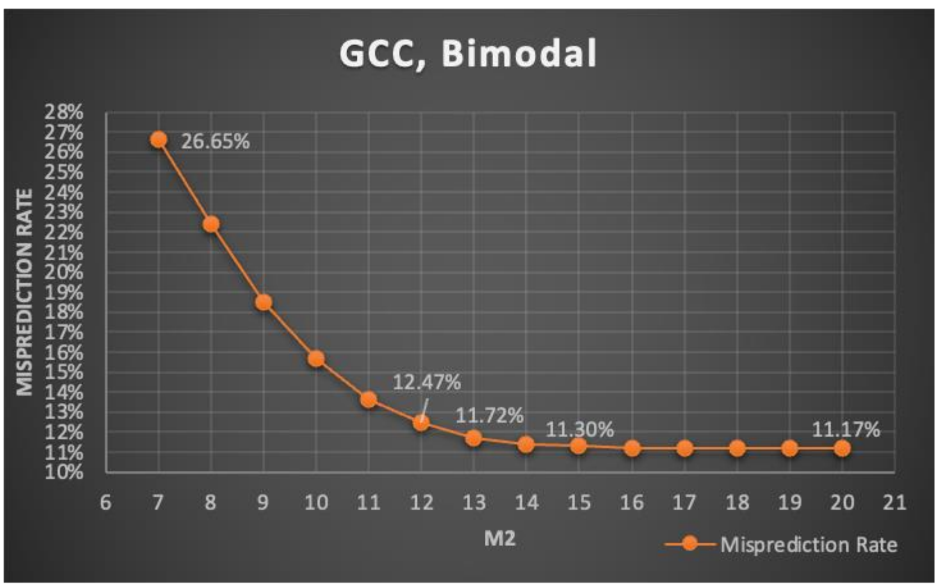
Student's electronic signature: ___PRAJAKTA KESHAVRAO JADHAV__
(sign by typing your name)

Course number: __ECE563__

Q1: Simulate BIMODAL PREDICTOR for different sizes (7 ≤ *m2* ≤ 20). Use the traces *gcc*, *jpeg*, and *perl* in the trace directory.

BIMODAL BP			
	Misprediction Rate in %		
m2	GCC	JPEG	PERL
7	26.65	7.92	21.31
8	22.43	7.79	16.45
9	18.49	7.74	14.14
10	15.67	7.70	11.95
11	13.65	7.62	11.05
12	12.47	7.60	9.09
13	11.72	7.59	8.92
14	11.37	7.59	8.82
15	11.30	7.59	8.82
16	11.21	7.59	8.83
17	11.19	7.59	8.83
18	11.17	7.59	8.83
19	11.17	7.59	8.83
20	11.17	7.59	8.83

1. **GCC**

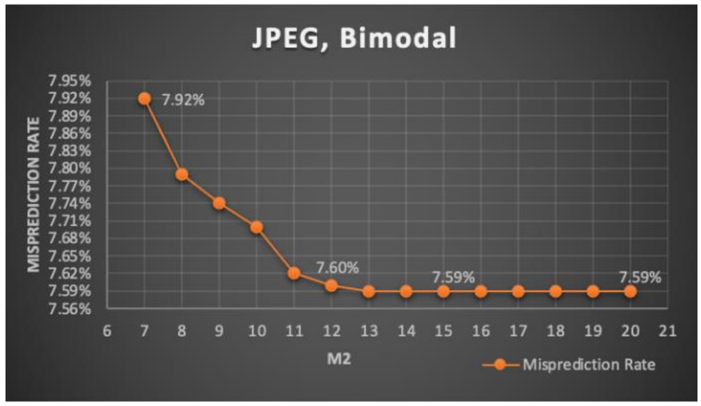


For GCC benchmark, bimodal predictor with m2=7, predicted 26.65 % of 2,000,000 branches incorrectly. As the number of m2 bits of BHT increases prediction rate decreased with –(5to1)% change factor from 7 to 13.

Observations:

- Misprediction rate range covered from 26.65% to 11.17% is 15.48%
- Bimodal performed good enough up to m2=17; BHT size of 2*2^17 = 2^8*2^10 = 256KB with 11.19% misprediction rate
- But, with m2=16, size of BHT will be 128KB giving 11.21% misprediction rate. That is for the 0.02% improvement BHT size got doubled. Here**, m2=16; BHT Size 128KB** will be preferable over m2=17.
- From m2=16 onwards bimodal performance doesn't improve much which shows even though size of BHT keep doubling, after this point Bimodal will not perform any better.

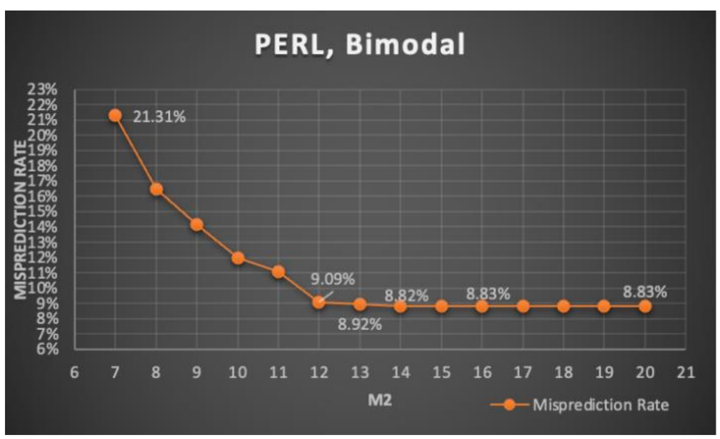
1. **JPEG**



Observations:

- Misprediction rate range covered from 7.92%;m2=7 to 7.59%;m2=13 is 0.33%. It shows for JPEG, bimodal resulted in better misprediction rate compared to the bimodal. But the misprediction rate variation is poor i.e. just for 0.33% of performance improvement size of BHT was doubled from 256B to 16KB
- JPEG result also shows m2=13 onwards misprediction rate remained exact same i.e. 7.59% and for m2=12 it is 7.60%. Here, doubling the size of BHT resulted into negligible improvement of 0.01%. Here, **m=12 i.e. BHT size= 8KB is preferable** over m=13
- Similar to GCC for JPEG as well, after certain point doubling the size of BHT (here from m2=13 onwards) didn't result in any performance improvement.

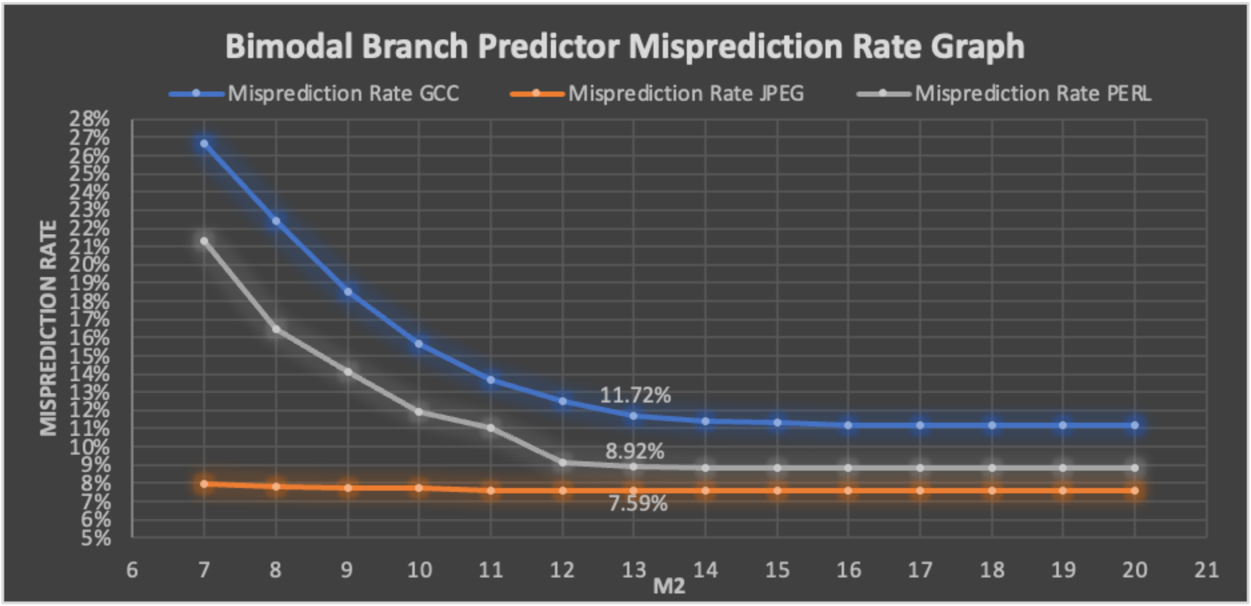
2. **PERL**



Observations:

- Misprediction rate range covered from 21.31%;m2=7 to 8.82%;m2=14 is 12.49%. It shows for PERL, bimodal resulted in better misprediction rate compared to the bimodal.
- For **best misprediction rate of 8.82% here, m2 is 14**, i.e. **BHT size = 32KB**. Here, doubling size to 64KB didn't result in any improvement in misprediction rate. While from m2=13, 8.92% misprediction rate improved by 0.1%
- PERL result also shows m2=14 onwards misprediction rate remained mostly similar with additional poor performance decrease of 0.01% from m=16 onwards.
- Similar to GCC & JPEG, for PERL as well, after certain point doubling the size of BHT (here from m2=14 onwards) didn't result in any performance improvement.

Conclusive observations for Bimodal BP:



- A performance comparison of all three benchmarks leads to the selection of m=13, i.e. BHT size 16KB for Bimodal BP.
- Even when the size of the BHT is increased, the accuracy of the bimodal branch predictor stopped improving after a certain point(the curve flattens out)
- Here, conflicts in the table isn't the problem as the increasing bits per entry also does not improve accuracy i.e. 1bit counter to 2bit counter
- Reason is Bimodal's BHT uses only recent local history of a branch to predict future
- Hence the solution GShare predictor which looks at global history of other branches in making a prediction about the current one.

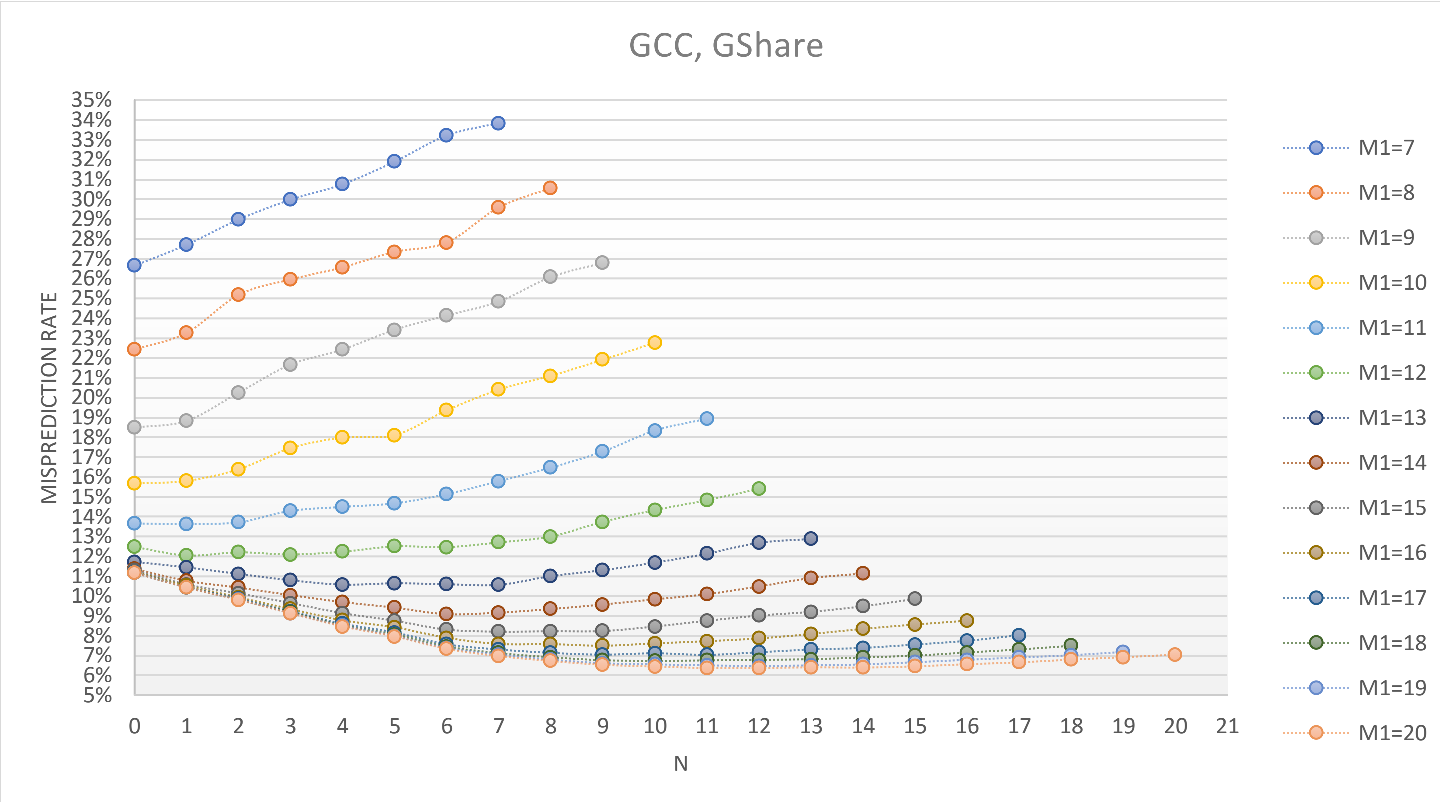
Q2: Simulate GSHARE PREDICTOR for different sizes (7 ≤ *m* ≤ 20), and for each size, *i.e.*, for each value of *m*, sweep the global history length *n* from 0 to *m*. Use the traces *gcc*, *jpeg*, and *perl* in the trace directory.

In case of Gshare, the N bits of global history are xor-ed with the N higher-order bits of the M1 low-order bits of the PC (after discarding the 2 lowest bits) to generate the index into the BranchHistoryTable. The BHT consists of 2n two-bit saturating counters initialized to weakly taken(2).

To understand the effect of global history register(GHR) length on gshare's prediction accuracy, gshare was simulated on *gcc*, *jpeg* & *perl* benchmarks where BHTs indexed(M1) using bits ranging fromM1=7 to M1=20 and history lengths(N) bits ranging from 0 to the number of index bits.

The misprediction rate on each benchmark is presented in graph 5,6,7 in which each curve represents a particular predictor size based on number of M1 bits i.e SIZE = 2*2^M1 bits.

1. GCC

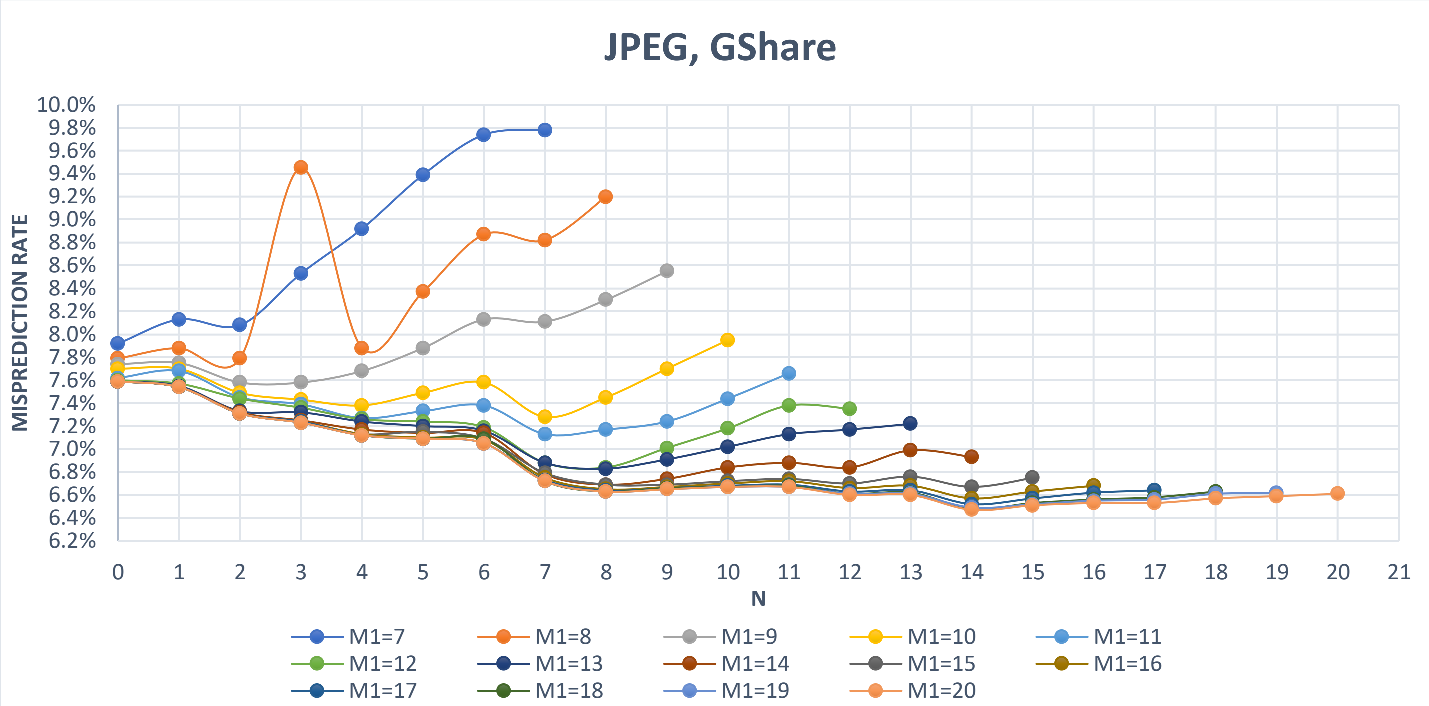


Graph 5: GShare Performance for GCC

Observations:

- With M1 bits ranging from 7 to 12, the rate of misprediction rose as GHR length increased from 0 onwards. While when M1 bits increased from 13 to 20 and N was varied from 0 to 7, the rate of misprediction reduced with slight changing factor. Notably, for all M1 7 to 20, when N-bits are changed from 8 to 20 , the rate of misprediction continuously increased with minor differences each time.
- For N=7 bits GHR length, misprediction rate varied from 6.97% to 33.84% w.r.t M1=20 to M1=7. But the least of 6.97% achieved here is for M1=20 which yields 2GB BHT size.
- Worst for BHT Size: For M1=20, variation GHR length(N) from 1 to 20 showed lesser/no effect on misprediction rate. Least misprediction rate achieved here is 6.37% for n=11 and maximum reached is 10.42% for n=1.
- Best for BHT Size: For M1=14 variation GHR length(N) from 0 to 6 resulted in good prediction accuracy, precisely the with n=6 where misprediction rate is 9.08% and increasing the N upto 14 will increase misprediction rate maximum to11.13%.
- <u>In particular, here M1=14 showed less variation in prediction accuracy as GHR length increased from 1 to 14, covering range of 11.13-9.08=2.05% of misprediction rate.</u>
- If worst and best choices are compared from BHT size's point of view, difference of misprediction rate is 9.08-6.37=2.71%. And respective sizes differ from 2GB to 32KB. **Hence M1=14 with N=6 can be preferred as the ideal choice for GCC.**

2. **JPEG**

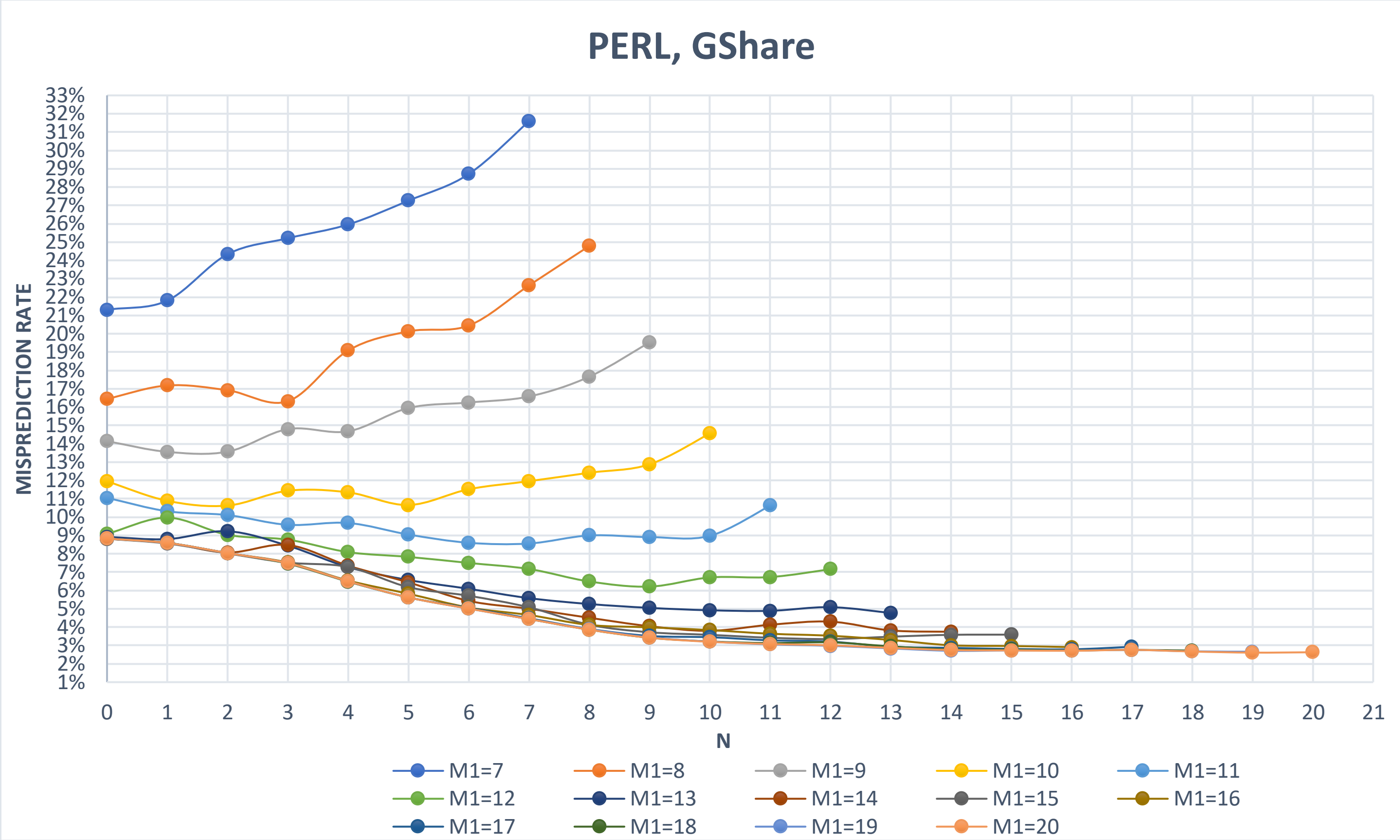


Graph 6: GShare Performance for JPEG

Observations:

- Compared to GCC, JPEG's misprediction rate range is short as well as has better prediction accuracy.
- For N=7 bit of history register length, misprediction rate varied from 7.05% to 9.78%. But the least of 7.05% is achieved for M1=20 which yields 2GB BHT size.
- For M1=20 increasing branch history length gives best accuracy. Least misprediction rate achieved here is 6.47%;N=14 and maximum is 7.54% for N=1.
- For M1=14, with GHR length of N=8bits, the least misprediction rate attained is 6.69%, while the most misprediction rate reached is 7.54 % with N=1. The misprediction range covered is 0.85%.
- In particular, within 32KB BHT Size, it displays less degradation in prediction accuracy as branch history length increases, and hence **the M=14, N=8 combination can be considered the ideal choice for JPEG**.
- Notably, JPEG with M1=8, BHT Size 512KB for N=3 showed spike giving misprediction rate of 9.45% which is 9.45-7.79=1.66% sudden increase from N=2.

3. PERL



Graph 7: GShare Performance for PERL

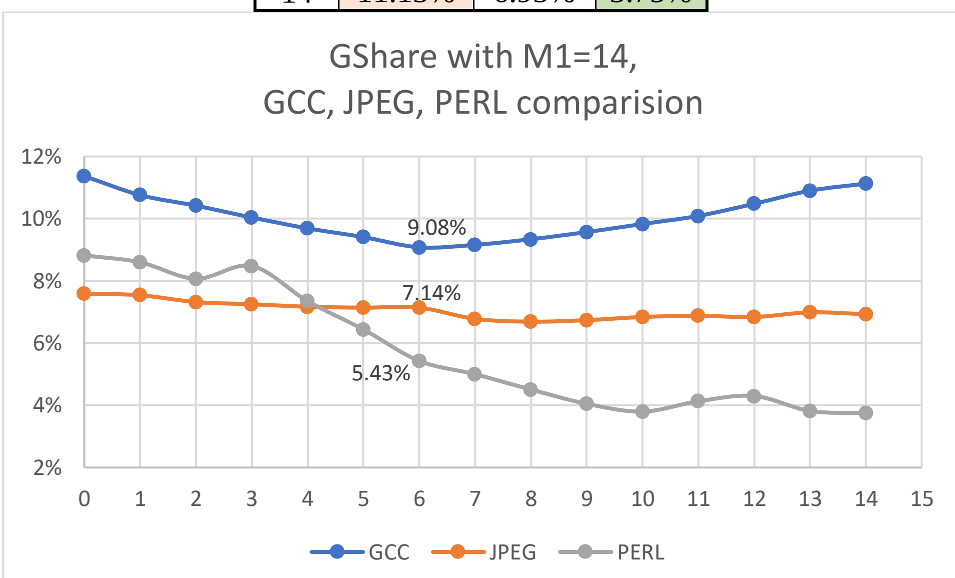
Observations:

- Compared to GCC, PERL gives better prediction accuracy range but not as good as JPEG's range.
- For N=7 bit of global history register length, misprediction rate varied from 4.44% to 31.60% w.r.t M1=20 to M1=7. But the least of 4.44% achieved here is for M1=20 which yields 2GB BHT size.
- Worst for BHT Size: For M1=20 increasing GHR length gave best accuracy. Least misprediction rate achieved here is 2.63% for N=20 and maximum reached is 8.59% for N=1.
- Best for BHT Size: For M1=14, with GHR length n=8bits, the least misprediction rate attained is 3.75%; N=14, while the most misprediction rate reached is 8.60 %; N=1. <u>Also, this combination, displays less variation in prediction accuracy when GHR length increased from 1 to 14</u>
- If worst and best choices are compared from BHT size's point of view, difference of misprediction rate is 3.75-2.63=1.12%. And respective sizes differ from 2GB to 32KB. **Hence M1=14 with N=8 can be considered the ideal choice for PERL.**
- Notably, PERL with M1=8, BHT Size 512KB for N=3 showed short drop of misprediction rate i.e. 16.32% from 16.92%;N=2 but again sharply increased to 19.09% for N=4.

Conclusive observations for GShare

- GCC and PERL misprediction rates varies from 0% to 35% while for JPEG it varies within range of 6% to 10%.
- It's identified by comparison of above three different benchmarks behaviors that how the code predictability evolves when the history length is increased: the predictability for GCC & PERL improves as more history is used, similarly JPEG with some irregularities shows improvements as well.
- JPEG and PERL show some irregular short spikes & drops for different history lengths.
- Considering overall performance for all three benchmarks, M1=14 i.e. BHT size 32KB can be ideal choice for GShare.

	M1=14		
N	GCC	JPEG	PERL
0	11.37%	7.59%	8.82%
1	10.76%	7.54%	8.60%
2	10.42%	7.32%	8.07%
3	10.04%	7.25%	8.48%
4	9.69%	7.17%	7.35%
5	9.41%	7.14%	6.44%
6	9.08%	7.14%	5.43%
7	9.16%	6.78%	5.00%
8	9.34%	6.69%	4.51%
9	9.57%	6.74%	4.05%
10	9.83%	6.84%	3.80%
11	10.09%	6.88%	4.13%
12	10.48%	6.84%	4.30%
13	10.90%	6.99%	3.82%
14	11.13%	6.93%	3.75%



CONCLUSION:

From Branch predictor project work following are the conclusive remarks:

Hybrid Predictor implements both Bimodal and GShare algorithms. It adapts to the particular branch prediction method by way of the predictor selection mechanism. But it needs additional chooser policy which might take resources away from Bimodal/GShare. Generally, branches tend to show either local or global history. So the Hybrid Predictor

- Selects Bimodal when local history is beneficial
- Selects Gshare when global history is beneficial

Bimodal uses only M2 bits of PC , when M2 bit count is less, the interreferences between unique branches is more. Also notably, for bimodal the misprediction rates completely flattens out after certain value of M2 . That shows, even after continues doubling of BHT size where every branch will get its own counter, prediction accuracy won't improve.

While the GShare uses that additional 2bit counter of each branch and specialize it with different counter values & use of global history bits and hence gives better prediction accuracy. But here sometimes global history bits xor-ing with each branch's PC bits may result in same Index bits to BHT. Hence the interferences still exist in GShare as well.