NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2018 (Rotenberg)

Project #2: Branch Prediction

by

<< SALONI SHAMBHUWANI >>

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

Student's electronic signature: __Saloni Shambhuwani____
(sign by typing your name)

Course number: _____563_____
(463 or 563 ?)

Bimodal Predictor

In this experiment, the 2-bit bimodal predictor is tested against different values of m (the number of lower order bits of PC to be used as an index of the predictor table) and different benchmarks - gcc, jpeg and perl. The misprediction rate is calculated in each case and plotted against the m value.

Bimodal Predictor is a special case of gshare with N=0 as only the PC bits are used to generate the index.

1. GCC Trace

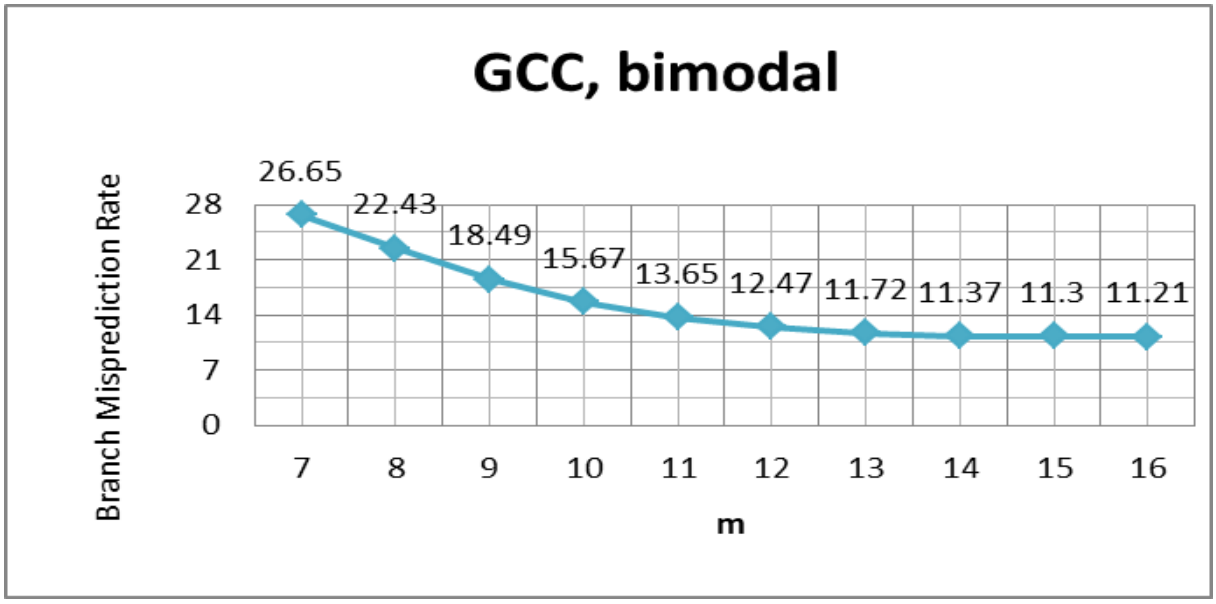


Figure: Bimodal predictor misprediction rate for gcc trace

m	Misprediction Rate(gcc)	m*Misprediction rate
7	26.65	186.55
8	22.43	179.44
9	18.49	166.41
10	15.67	156.7
11	13.65	150.15
12	**12.47**	**149.64**
13	11.72	152.36
14	11.37	159.18
15	11.3	169.5
16	11.21	179.36

For **m=12**, the trade-off between the branch history table size and the misprediction rate gives the best value.

To minimize the value of 'm' as well as the value of the misprediction rate, we consider their product. Thus, we are actually considering the log2(size) of the branch history table size along with the misprediction rate.

2. JPEG Trace

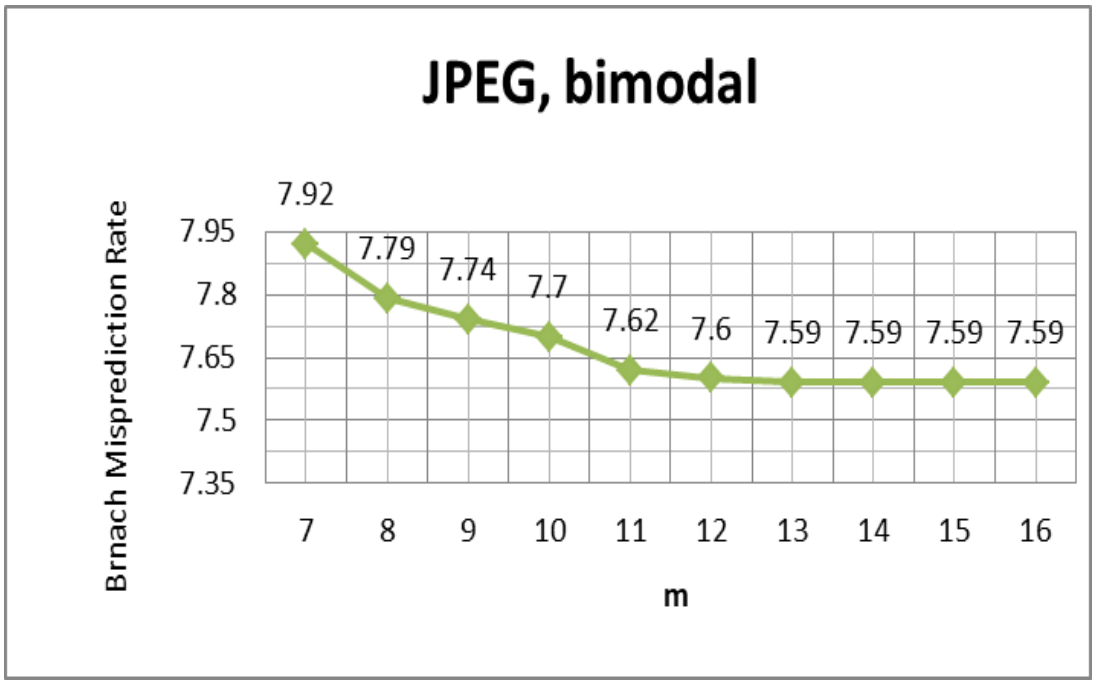
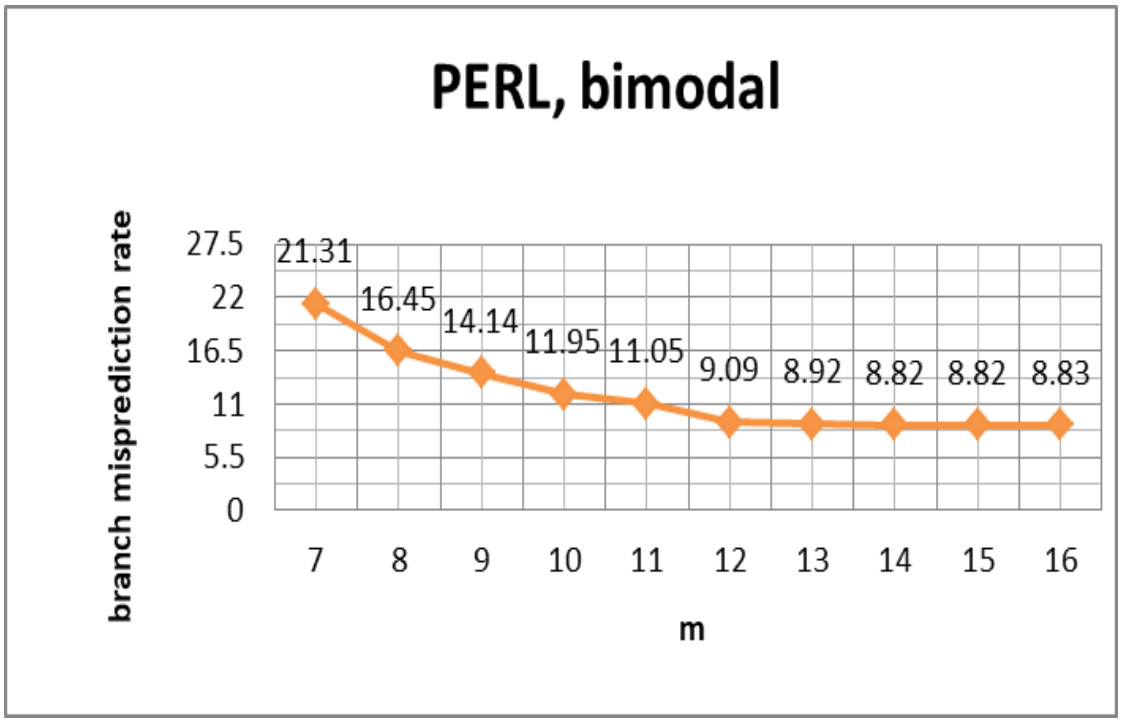


Figure: Bimodal predictor misprediction rate for jpeg trace

m	Misprediction Rate (jpeg)	m* Misprediction Rate
7	**7.92**	**55.44**
8	7.79	62.32
9	7.74	69.66
10	7.7	77
11	7.62	83.82
12	7.6	91.20
13	7.59	98.67
14	7.59	106.26
15	7.59	113.85
16	7.59	121.44

The minimum value for the trade-off is attained for a value of m=7. The lowest value of 'm' gives the best trade-off in this case since the value of the misprediction rate for the JPEG does not reduce much with increase in the branch history table size. Since the misprediction rate does not decrease, the address trace most probably does not have a large number of unique branch addresses, or it has a high-looping structure. This allows the predictor counters to be trained well and results in a low misprediction rate. As a result, once all the branches have been assigned a unique counter in the table, there is no added benefit of increasing the table size further.
m=7 gives the best configuration according to the metric used. Looking at the graph, the value of the misprediction rate stabilizes around m=13, but there is not a proportional decrement in the misprediction rate.

3. PERL Trace



m	Misprediction Rate(perl)	m* Misprediction
7	21.31	149.17
8	16.45	131.6
9	14.14	127.26
10	11.95	119.5
11	11.05	121.55
12	**9.09**	**109.08**
13	8.92	115.96
14	8.82	123.48
15	8.82	132.3

16	8.83	141.28

The PERL trace follows a similar trend to the GCC predictor. Initially, increasing the value of 'm' gives a good trade-off between the predictor size and the misprediction rate. However, after m=12, increasing the size further gives diminishing returns on the reduction in the misprediction rate.
Thus, **m=12** gives the best trade-off between the size and the misprediction rate.

Analysis of Bimodal Predictor Performance:

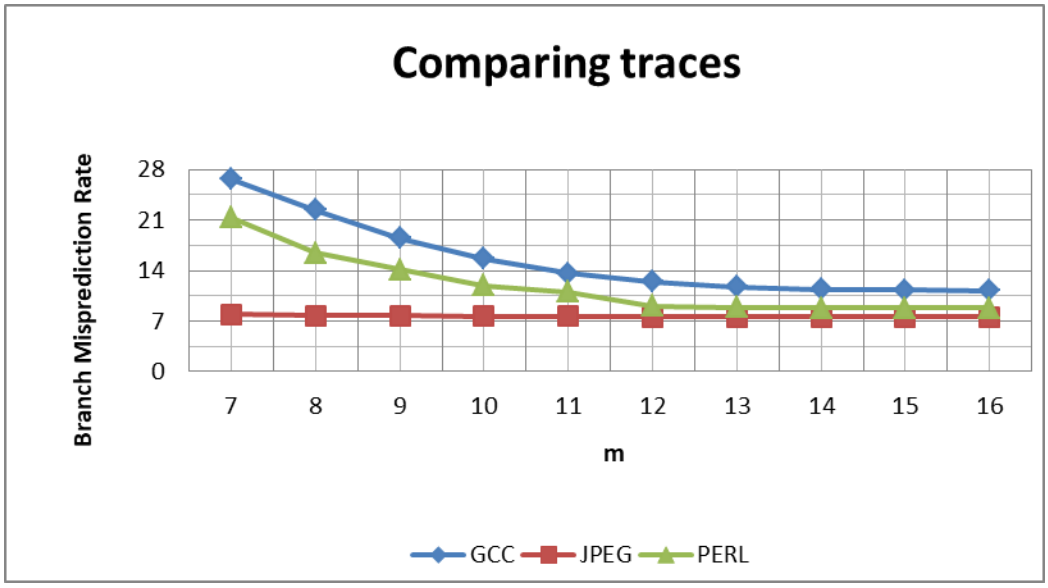


Figure: Misprediction rates for GCC, JPEG and PERL address traces against the value of 'm'.

As per the graph, the misprediction rate for all three address traces reduces as the number of PC bits used to index the branch prediction is increased. More the number of bits to index the prediction table, greater the number of counters in the branch history table. Thus addresses will map to different rows in the branch history table leading to better prediction.

Also, the misprediction rate for the JPEG file trace is lower than either GCC or PERL trace. Along with the branch predictor configuration, the misprediction rate also depends upon the nature of the address trace. The nature of some kinds of branches may be easily captured by a simple 2-bit counter predictor, while other might need a more sophisticated predictor.

At **m=12** it gives the best configuration for these three address traces.

GShare Predictor

For GShare predictor we XOR ‘n’ bits from the PC address with the global branch history register in order to generate the index for the branch history table. The following graphs plot the misprediction rate against various values of ‘m’ and ‘n’.

1. GCC Trace

<u>Design of best predictor for GCC Trace:</u>

In the case of a GShare predictor, the total storage overhead on account of the branch prediction circuitry is n + 2*2^(m). Since the value of ‘n’ is negligible compared to the exponential, we can assume the storage overhead to be 2*2^(m).

Therefore, to find the best possible predictor configuration, we take $\log_2$(branch table size)*Misprediction Rate. After analyzing these product values (i.e. the trade-off between predictor size and the misprediction rate), we find that the configuration **n=8, m=16** provides the best value. The misprediction rate for this configuration is 7.57%.

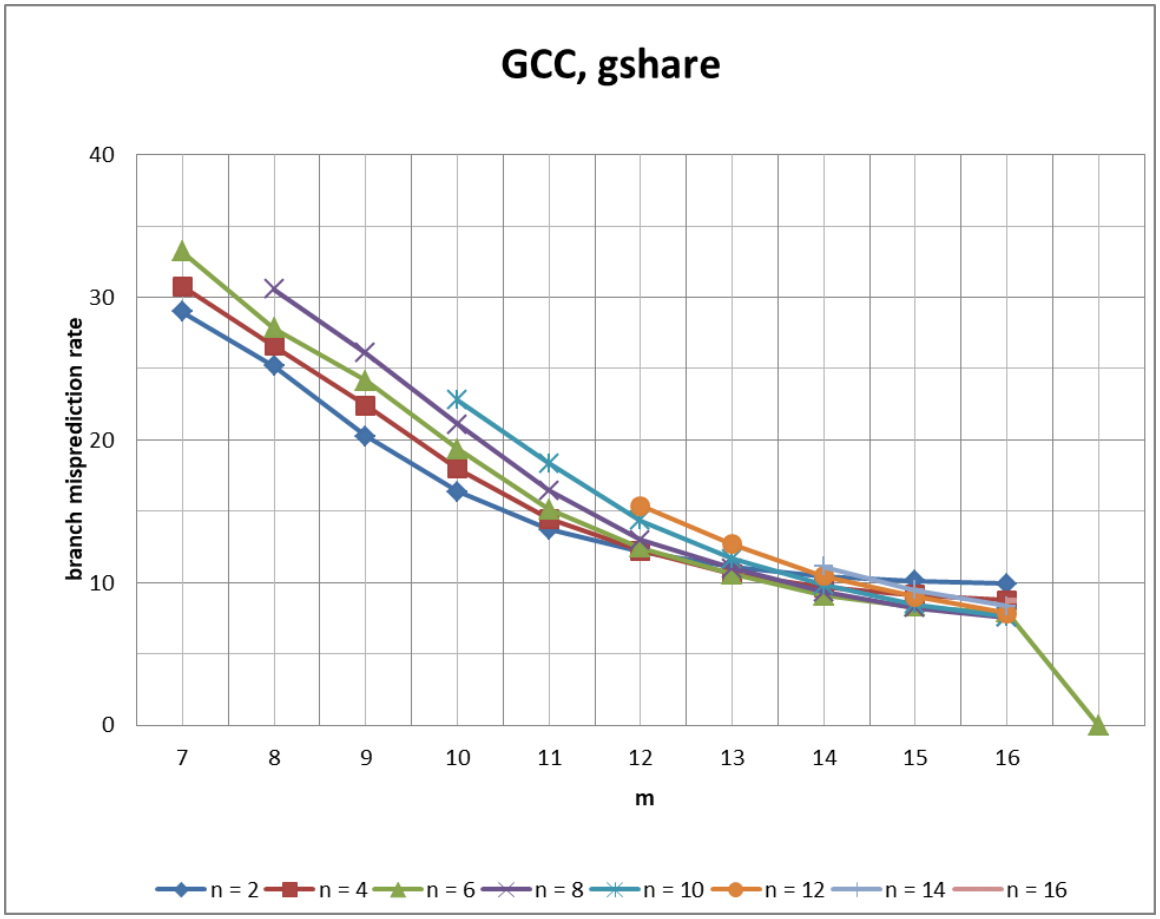


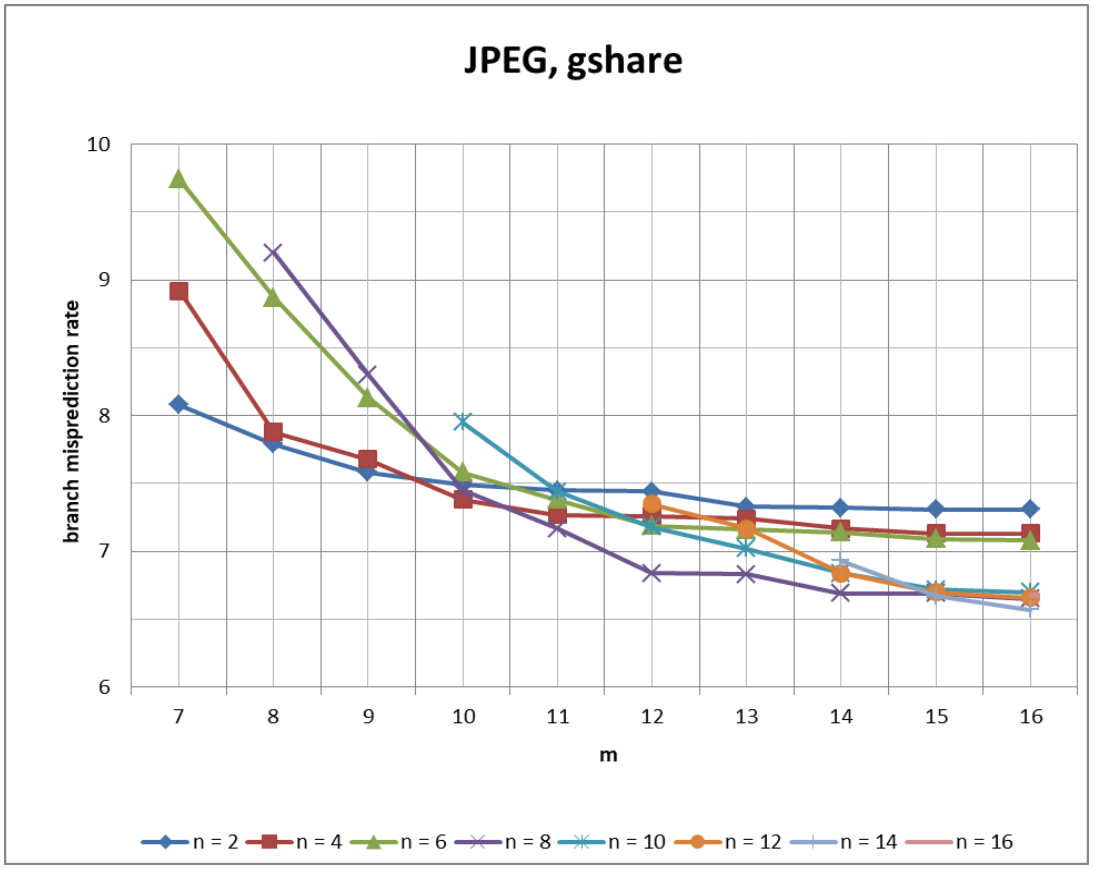
Figure: Mispridiction rate for gshare for diffent values of m and n

Misprediction Rate(gcc)										
n/m	7	8	9	10	11	12	13	14	15	16
2	28.98	25.18	20.25	16.39	13.71	12.2	11.11	10.42	10.13	9.93
4	30.76	26.57	22.43	17.99	14.49	12.23	10.57	9.69	9.13	8.77
6	33.22	27.82	24.14	19.36	15.14	12.46	10.59	9.08	8.3	7.89
8		30.56	26.08	21.1	16.47	13	11	9.34	8.22	**7.57**
10				22.77	18.34	14.33	11.68	9.83	8.46	7.61
12						15.4	12.68	10.48	9.01	7.86
14								11.13	9.48	8.34
16										8.75

2. JPEG Trace

<u>Design of best predictor for JPEG Trace:</u>

Increasing the number of bits used to index PC (m), or the number of bits in the global history table does not lead to a significant decrease in the misprediction rate. The least value of 'm' and 'n', which are **n=2, m=7** give the best trade-off while keeping the total storage size below the limit of 16KB.

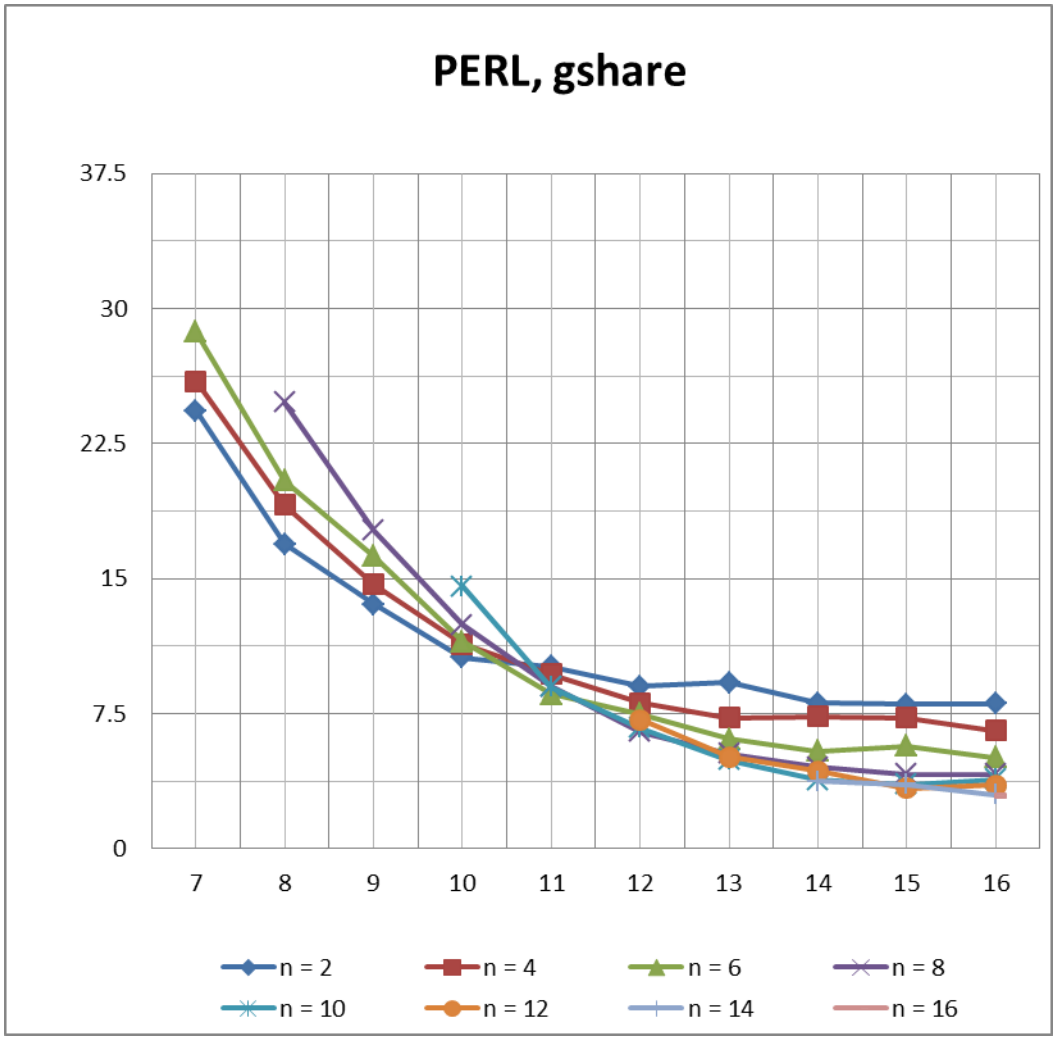


Misprediction Rate(jpeg)										
n/m	7	8	9	10	11	12	13	14	15	16
2	**8.08**	7.79	7.58	7.49	7.45	7.44	7.33	7.32	7.31	7.31
4	8.92	7.88	7.68	7.38	7.27	7.26	7.24	7.17	7.13	7.13
6	9.74	8.87	8.13	7.58	7.38	7.19	7.16	7.14	7.09	7.08
8		9.2	8.3	7.45	7.17	6.84	6.83	6.69	6.69	6.65
10				7.95	7.44	7.18	7.02	6.84	6.72	6.7
12						7.35	7.17	6.84	6.7	6.66
14								6.93	6.67	6.57
16										6.68

3. PERL Trace

Design of best predictor for PERL Trace:

Analyzing the trade-off between the predictor size and misprediction rates by looking at the product of the log2(branch table size)*misprediction rate, we find that the configuration **n=16, m=16** provides the best possible value. However, the size of the predictor for a configuration of n=16, m=16 exceeds the upper bound of 16KB for the predictor (by a tiny amount of 16 bits, which is negligible compared to 16KB used up by the branch prediction table).



PERL										
	7	8	9	10	11	12	13	14	15	16
2	24.34	16.92	13.57	10.63	10.11	9.03	9.23	8.07	8.02	8.04
4	25.96	19.09	14.68	11.35	9.68	8.09	7.27	7.35	7.28	6.54
6	28.71	20.45	16.25	11.52	8.6	7.5	6.09	5.43	5.71	5.07
8		24.79	17.66	12.42	9	6.49	5.26	4.51	4.13	4.12
10				14.57	8.98	6.71	4.92	3.8	3.58	3.84
12						7.16	5.09	4.3	3.35	3.53
14								3.75	3.58	3.01
16										2.91

Analysis of GShare Predictor Performance:

Observing misprediction rates for the three benchmarks, the GCC benchmark has a higher misprediction rate compared to JPEG and PERL. This observation is similar to the Bimodal predictor. This is quite expected since we have just changed the indexing strategy in GShare, and not the predictor counter size (2 bits) or the algorithm to set the counter. In GShare, the minimum achieved misprediction rate is lower than the minimum achieved in Bimodal. This is due to the better indexing strategy employed in GShare.

Looking at the overall trend across all the three address traces, the best trade-off between predictor size and the misprediction rate is attained at a value of **m=16**.

Conclusion

After carefully analyzing the trade-offs between the predictor size and misprediction rate, we can conclude that the following configurations provide the lowest misprediction rate for all three address traces while consuming a reasonable low storage overhead at the same time:

1. Bimodal Predictor: m=12; Predictor Size = 2*2^(12) = 8Kb = 1KB

2. GShare Predictor: m=16, n=8; Predictor Size = 8 + 2*2^(16) ~= 128Kb = 16KB