NC State University

Department of Electrical and Computer Engineering

ECE 463/521: Fall 2015 (Rotenberg)

Project #2: Branch Prediction

by

PARTH BHOGATE

(ID: 200108628)

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

Student's electronic signature: Parth Bhogate
(sign by typing your name)

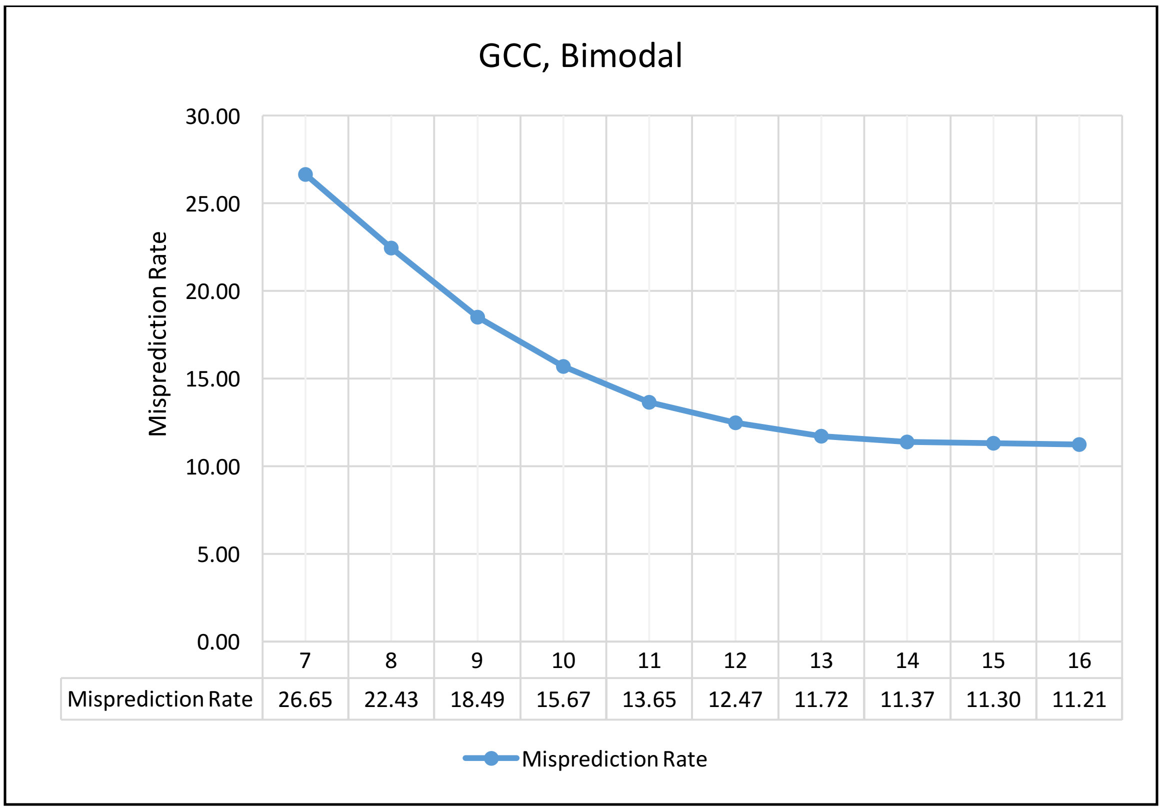
Course number: ECE 521

Bimodal Predictor

The Bimodal predictor uses 'm' bits of the PC to index the branch history table. Since only the PC bits are used to generate the index, Bimodal predictor is a special case of the GShare predictor with N=0.

The following graphs plot the misprediction rate for the Bimodal branch predictor for each of the three address traces:

1. GCC Trace



Design of best predictor for GCC Trace:

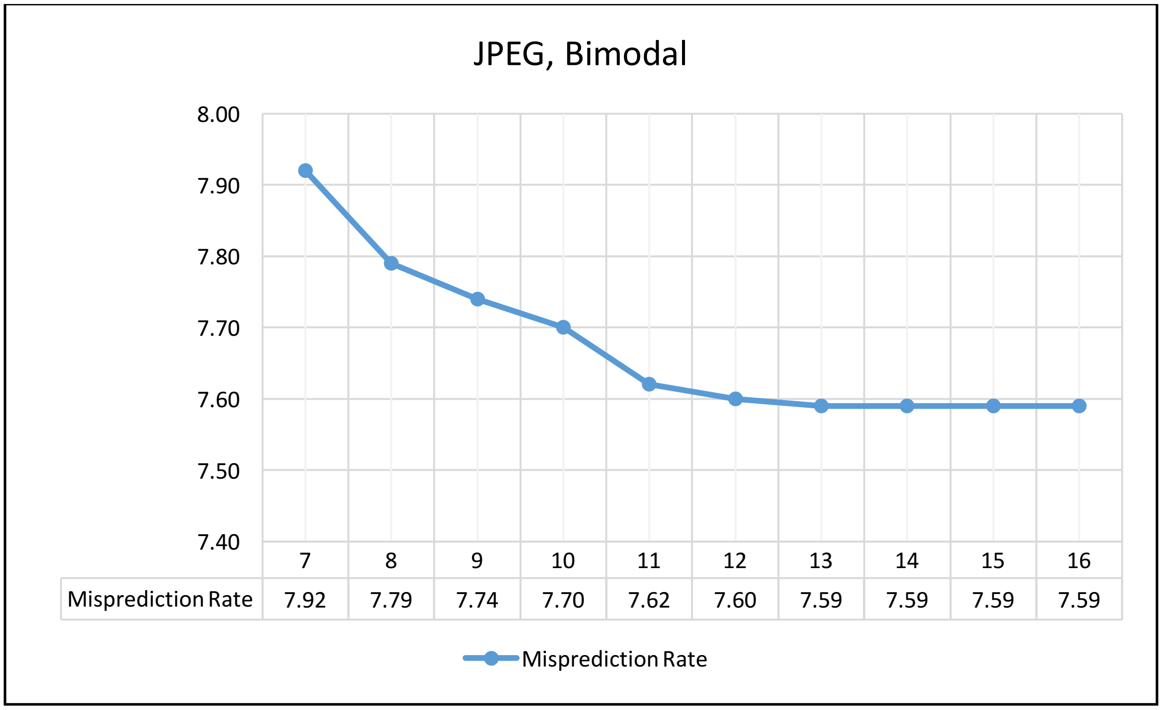
Value of m	Misprediction Rate	m * Misprediction Rate
7	26.65	186.55
8	22.43	179.44
9	18.49	166.41
10	15.67	156.7
11	13.65	150.15
12	12.47	149.64
13	11.72	152.36

14	11.37	159.18
15	11.30	169.5
16	11.21	179.36

In this table, the misprediction rates for each value of 'm' are shown. To minimize the value of 'm' as well as the value of the misprediction rate, their product has been considered. With a value of m=12, the size of the branch history table is 2*2^(12) = 2^(13). Thus, we are basically considering the $\log_2$(size) of the branch history table size along with the misprediction rate.

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

2. JPEG Trace



Design of best predictor for JPEG Trace:

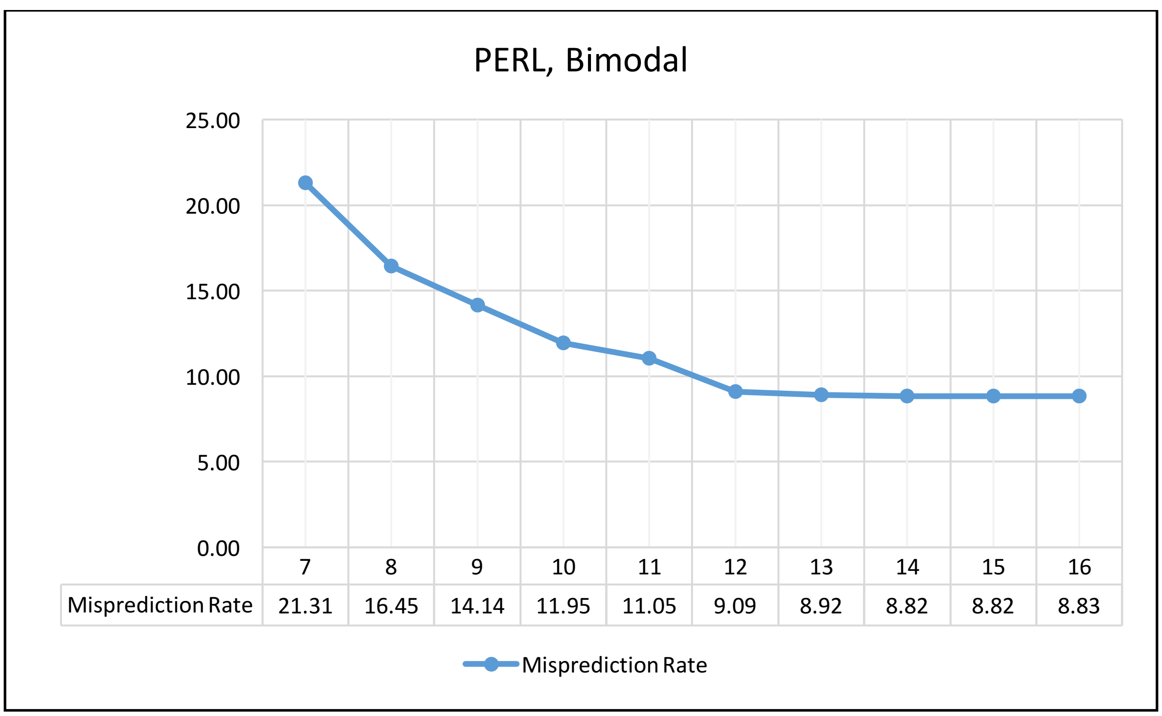
Value of m	Misprediction Rate	m * Misprediction Rate
7	7.92	55.44
8	7.79	62.32
9	7.74	69.66
10	7.70	77

11	7.62	83.82
12	7.60	91.2
13	7.59	98.67
14	7.59	106.26
15	7.59	113.85
16	7.59	121.44

In the case of the JPEG predictor, 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.

Thus, there are diminishing returns with increasing the value of 'm' from the initial value of m=7 itself. **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 due to increase in the number of index bits to justify the use of m=13 bits.

3. PERL Trace



Design of best predictor for PERL Trace:

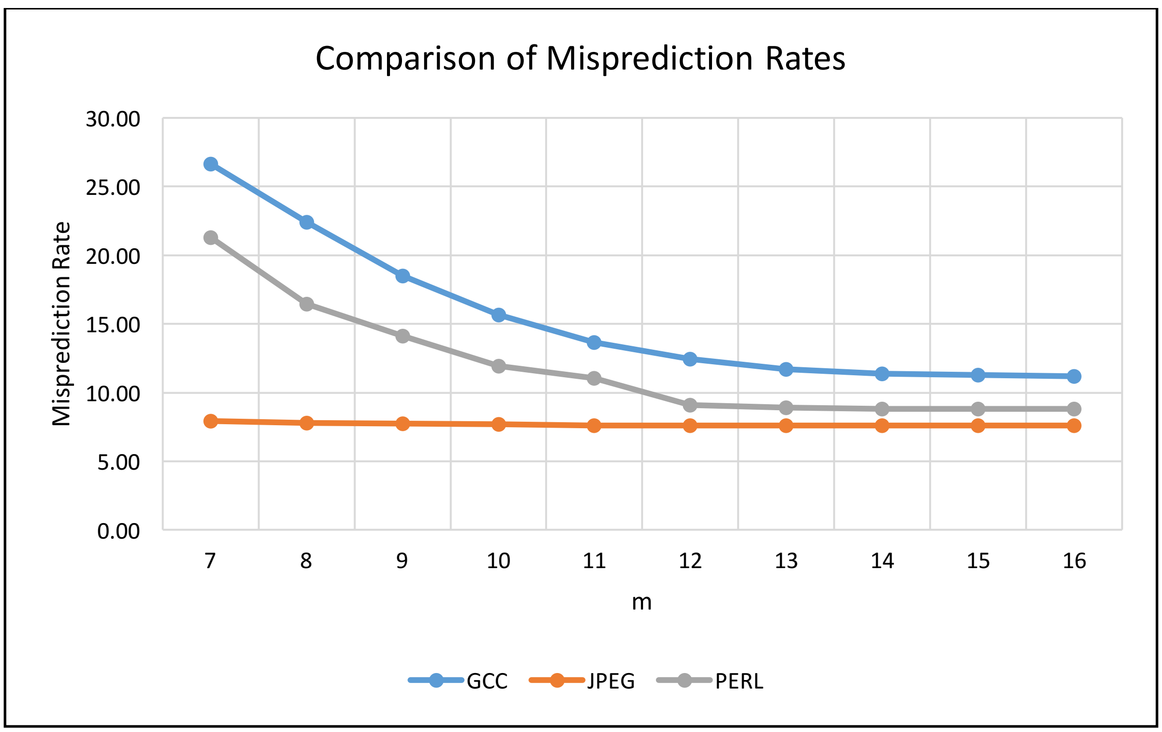
Value of m	Misprediction Rate	m * Misprediction Rate
7	21.31	149.17
8	16.45	131.6
9	14.14	127.26
10	11.95	119.5
11	11.05	121.55
12	9.09	109.08
13	8.92	115.96
14	8.82	123.48
15	8.82	132.3
16	8.83	141.28

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:

The following graph plots the misprediction rates for GCC, JPEG and PERL address traces against the value of 'm'.



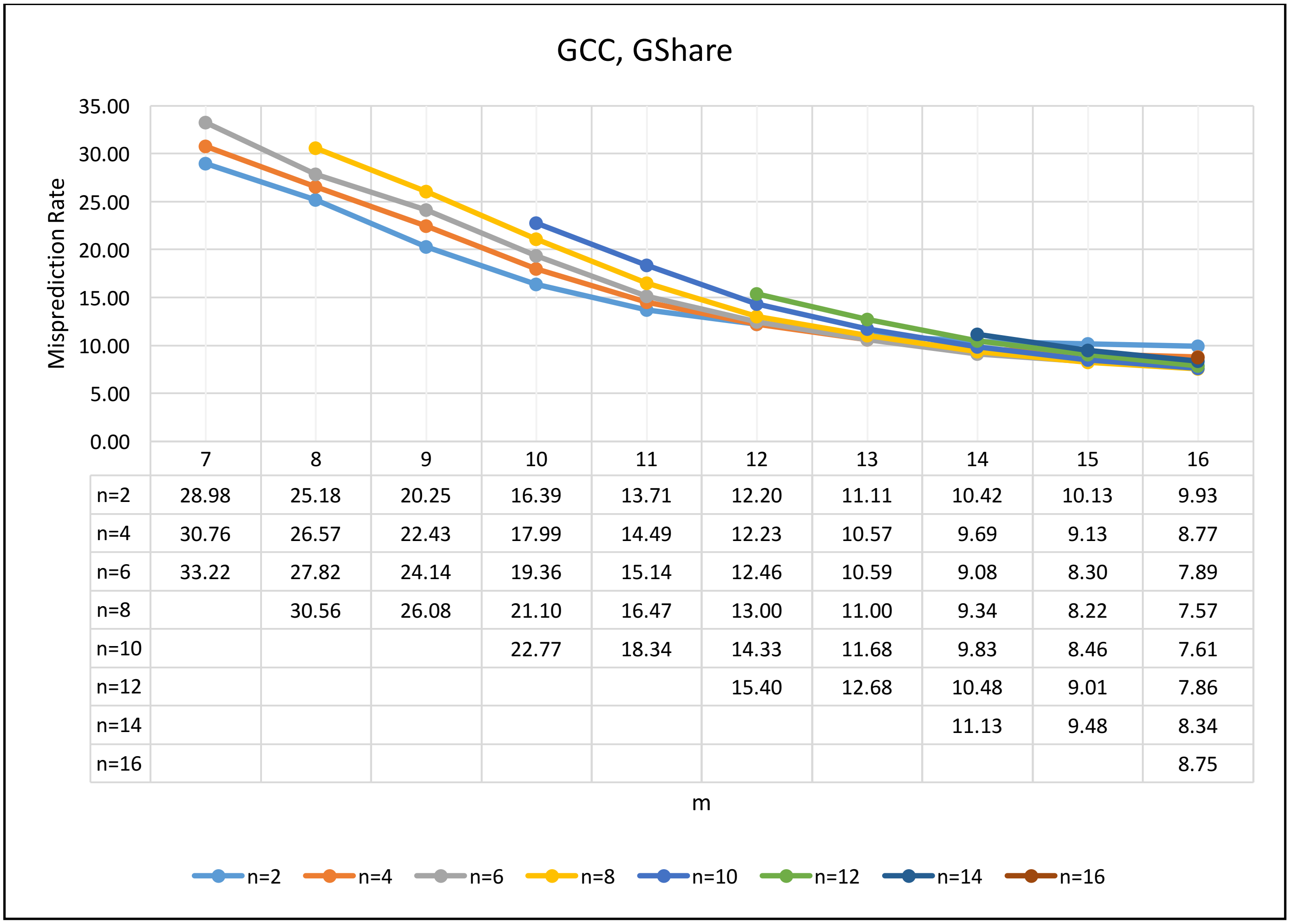
From the graph, we observe that the misprediction rate for all three address traces reduces as the number of PC bits used to index the branch prediction is increased. Using more number of bits to index the prediction table means there will be a greater number of counters in the branch history table. As a result, addresses will map to different rows in the branch history table, thus 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.

From the comparison graph, we can observe that for a size of m=12, the misprediction rate for all the three traces stabilizes, and there is not much improvement on account of increasing the value of 'm'. Thus, **m=12** gives the best configuration for these three address traces.

GShare Predictor

The GShare predictor XOR's '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



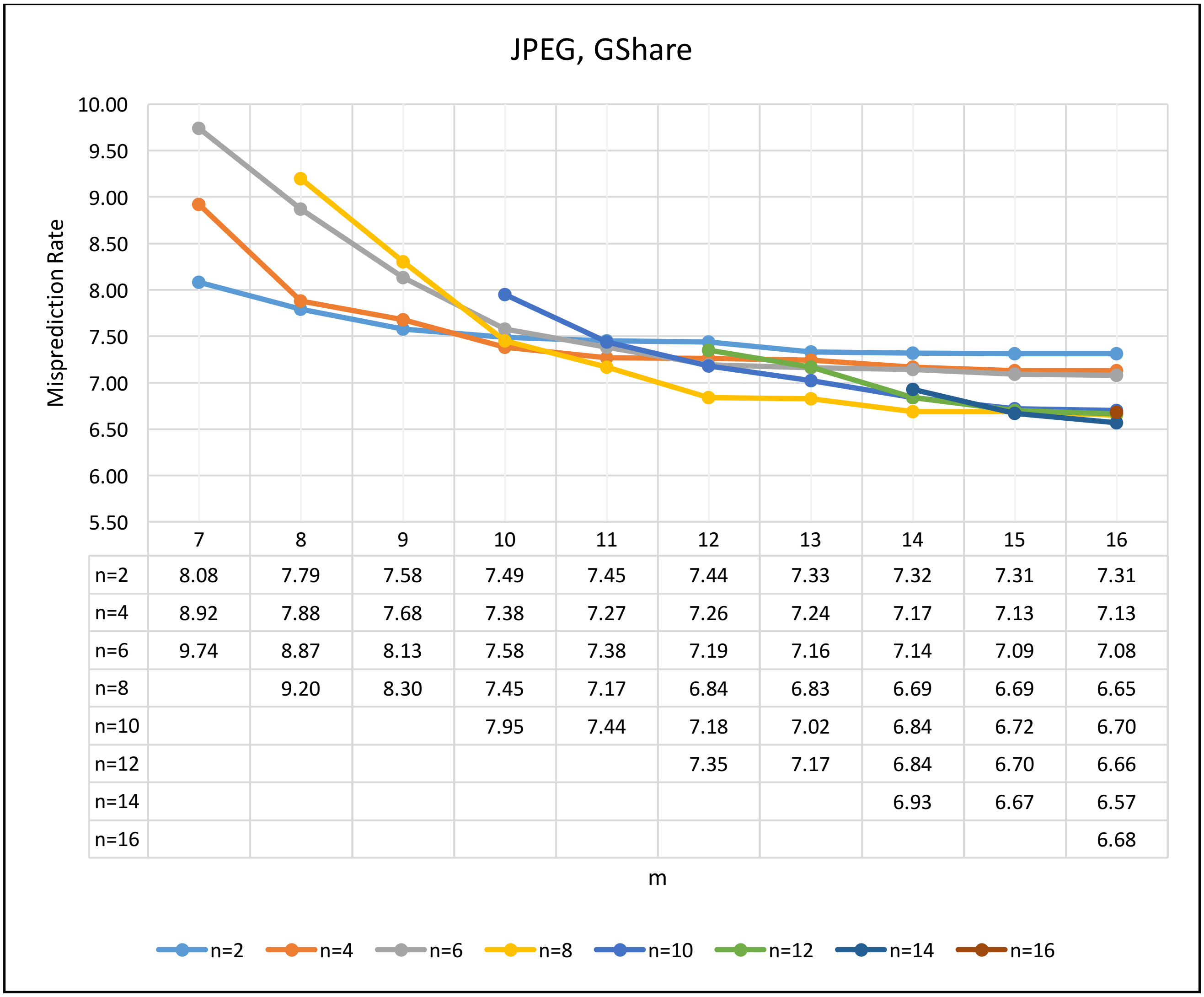
	7	8	9	10	11	12	13	14	15	16
n=2	28.98	25.18	20.25	16.39	13.71	12.20	11.11	10.42	10.13	9.93
n=4	30.76	26.57	22.43	17.99	14.49	12.23	10.57	9.69	9.13	8.77
n=6	33.22	27.82	24.14	19.36	15.14	12.46	10.59	9.08	8.30	7.89
n=8		30.56	26.08	21.10	16.47	13.00	11.00	9.34	8.22	7.57
n=10				22.77	18.34	14.33	11.68	9.83	8.46	7.61
n=12						15.40	12.68	10.48	9.01	7.86
n=14								11.13	9.48	8.34
n=16										8.75

Design of best predictor for GCC Trace:

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).
Thus, for deducing the best possible predictor configuration, we take the product of the $\log_2$(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%.

2. JPEG Trace

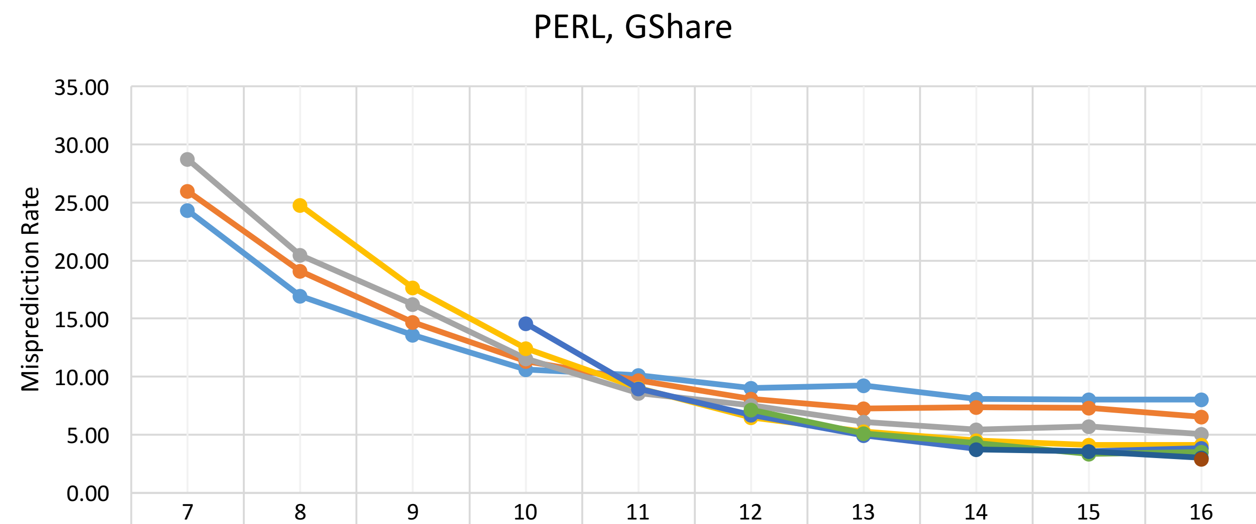


	7	8	9	10	11	12	13	14	15	16
n=2	8.08	7.79	7.58	7.49	7.45	7.44	7.33	7.32	7.31	7.31
n=4	8.92	7.88	7.68	7.38	7.27	7.26	7.24	7.17	7.13	7.13
n=6	9.74	8.87	8.13	7.58	7.38	7.19	7.16	7.14	7.09	7.08
n=8		9.20	8.30	7.45	7.17	6.84	6.83	6.69	6.69	6.65
n=10				7.95	7.44	7.18	7.02	6.84	6.72	6.70
n=12						7.35	7.17	6.84	6.70	6.66
n=14								6.93	6.67	6.57
n=16										6.68

Design of best predictor for JPEG Trace:

The observation for the GShare JPEG address trace is similar to the observation for the Bimodal trace. 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.

3. PERL Trace



	7	8	9	10	11	12	13	14	15	16
n=2	24.34	16.92	13.57	10.63	10.11	9.03	9.23	8.07	8.02	8.04
n=4	25.96	19.09	14.68	11.35	9.68	8.09	7.27	7.35	7.28	6.54
n=6	28.71	20.45	16.25	11.52	8.60	7.50	6.09	5.43	5.71	5.07
n=8		24.79	17.66	12.42	9.00	6.49	5.26	4.51	4.13	4.12
n=10				14.57	8.98	6.71	4.92	3.80	3.58	3.84
n=12						7.16	5.09	4.30	3.35	3.53
n=14								3.75	3.58	3.01
n=16										2.91

Design of best predictor for PERL Trace:

The PERL trace follows a similar trend to the GCC trace. Analyzing the trade-off between the predictor size and misprediction rates by looking at the product of the $\log_2$(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).

Analysis of GShare Predictor Performance:

Looking at the misprediction rates for the three benchmarks, we observe that 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