NC State University
Department of Electrical and Computer Engineering
ECE 463/563: Fall 2019 (Dr. Huiyang Zhou)
Project #2: Branch Prediction

By
Vishnu Suresh Menon
(Your Name)

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

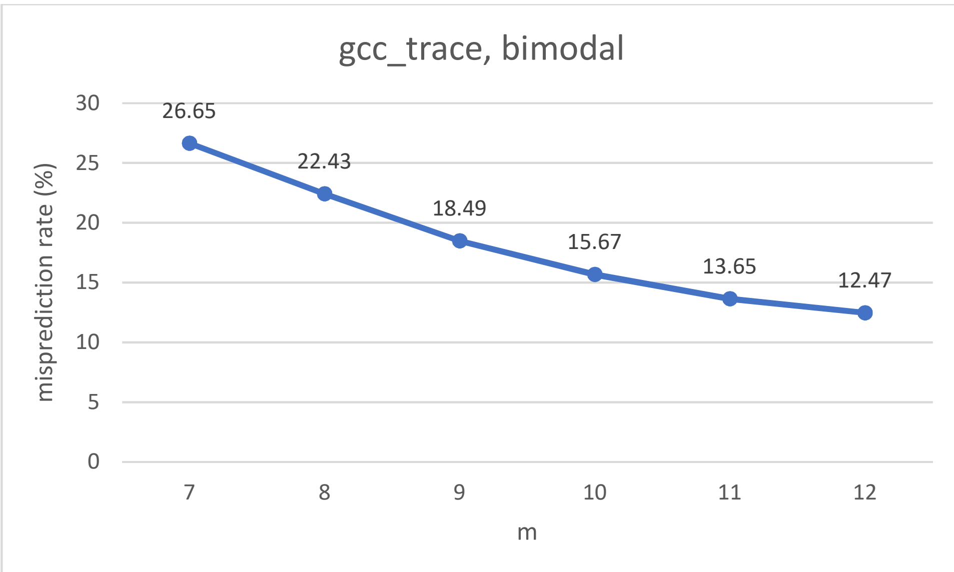
Student's Electronic Signature: **Vishnu Suresh Menon**
(Sign by typing your name)

Course Number: ______563_________
(463 or 563?)

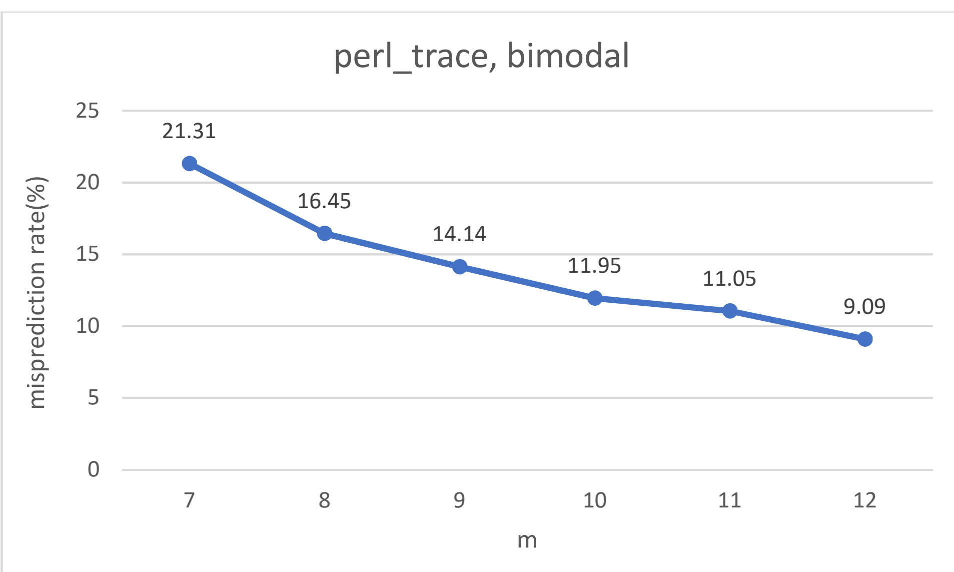
BIMODAL PREDICTOR

1. Graphs

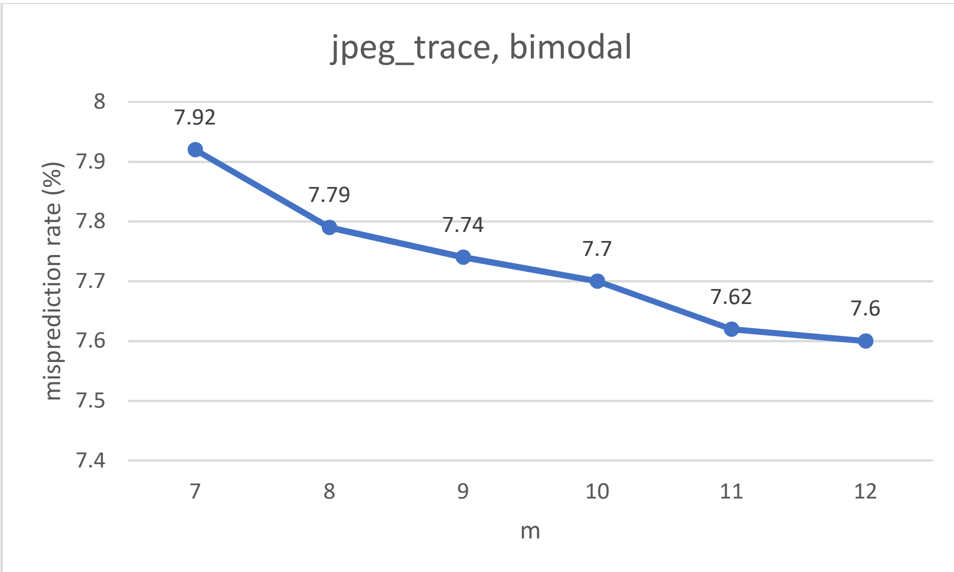
gcc_trace.txt						
m	7	8	9	10	11	12
misprediction rate (%)	26.65	22.43	18.49	15.67	13.65	12.47



perl_trace.txt						
m	7	8	9	10	11	12
misprediction rate (%)	21.31	16.45	14.14	11.95	11.05	9.09



jpeg_trace.txt						
m	7	8	9	10	11	12
misprediction rate(%)	7.92	7.79	7.74	7.7	7.62	7.6



2. Analysis

From the above data tables and graphs of the bimodal predictor, it is evident that the misprediction rate of the branch prediction unit decreases significantly and nearly stabilizes as the no: of PC bits are increased. This trend is observed and are similar among all the above-mentioned benchmarks (gcc_trace, perl_trace and jpeg_trace). From this its observed that as the amount of index bits increases the number of entries in the prediction table increases which helps to index to more entries making space for more accurate predictions and thus increasing the prediction accuracy.

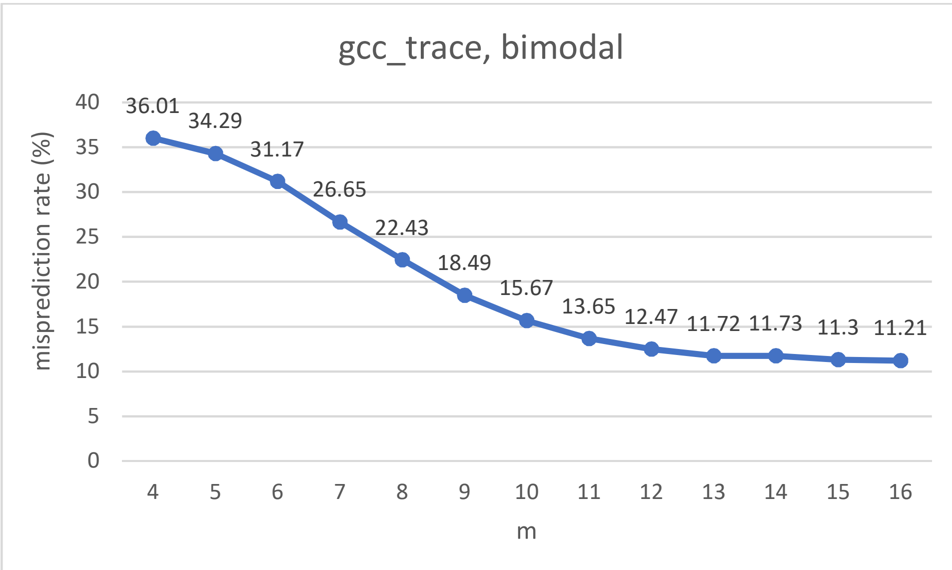
Since here, the prediction is based on a 2 bit saturating counter which is incremented on a taken branch and decremented on non-taken branch , the prediction pattern becomes constant at a particular size of the prediction table which is at the point of diminishing returns for 2 bit entries so there isn't much point in storing more.

Among benchmarks, its observed that the variation of prediction accuracy with size of prediction table (predictor cost of bits) is almost similar for all three benchmarks. For values of 'm' in the range of 7-12 the misprediction rate varies within the range of (26.65 – 12.47) and (21.31 – 9.09) for gcc and perl traces whereas for jpeg rate is almost stable at a range of 7.5, ie; size of hardware doesn't have significant influence on affecting the prediction accuracy.

3. Design

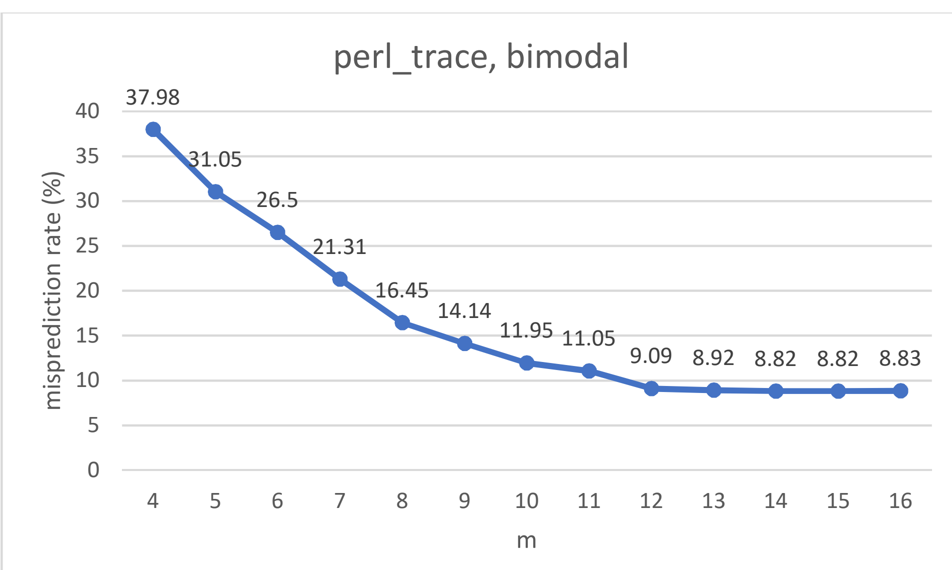
Maximum budget of storage : 16KB
Size of storage : 2^m * 2 where m = no: of PC bits and since we are using a 2-bit smith counter.

gcc-trace



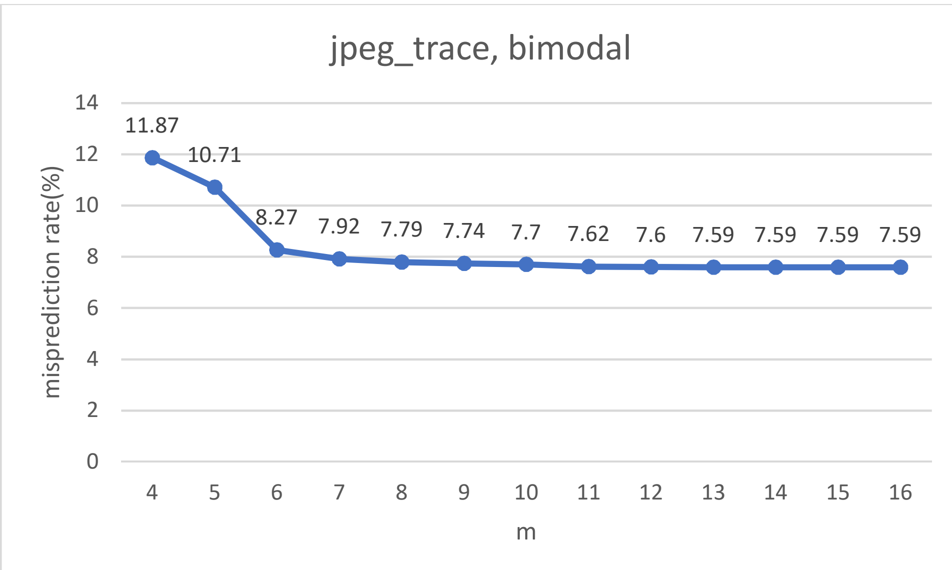
For the benchmark, gcc_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 3<m<17. The above trend reveals that at value of m = 13 the decrease in misprediction rate is almost equals to 0.01% which is highly insignificant. For values of 6<m<13 the rate difference is about 2.82% to 4.2% and for values m<6 rate difference is about 1.72%.
Therefore, at value m =13 the misprediction rate starts saturating to a minimal value of 0.01% ie; additional bits bring no use in the prediction accuracy rather increase the hardware area and predictor cost.

perl-trace



For the benchmark, perl_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 3<m<17. The above trend reveals that at values of 10<m<13 the decrease in misprediction rate varies in the range of (0 - 0.9)% ie; at m equals 10 and 12 the rate difference is 0.9 and 0.17 whereas at m equals 13 and 14 the rate difference is 0.1 and 0. So, therefore m=12 can be taken as design value. The reason behind that is for m = 10 the rate shoots up to value of 0.9 whereas at values m equals 13 and 14 doesn't provide a significant improvement in prediction accuracy and moreover the predictor cost increases.

jpeg-trace

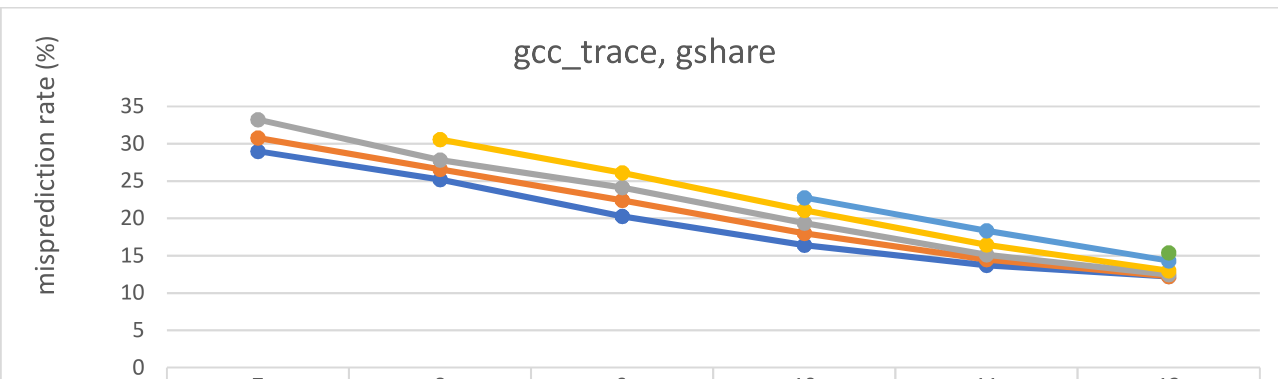


For the benchmark, perl_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 3<m<17. Here, the trend saturates to value of 7.7 range even after increasing the predictor cost bits. Therefore, at value m =8 is the ideal design configuration for the bimodal predictor since raising the predictor cost bits doesn't bring significant improvement prediction accuracy and the hardware area and cost shoots up unnecessarily.

GSHARE PREDICTOR

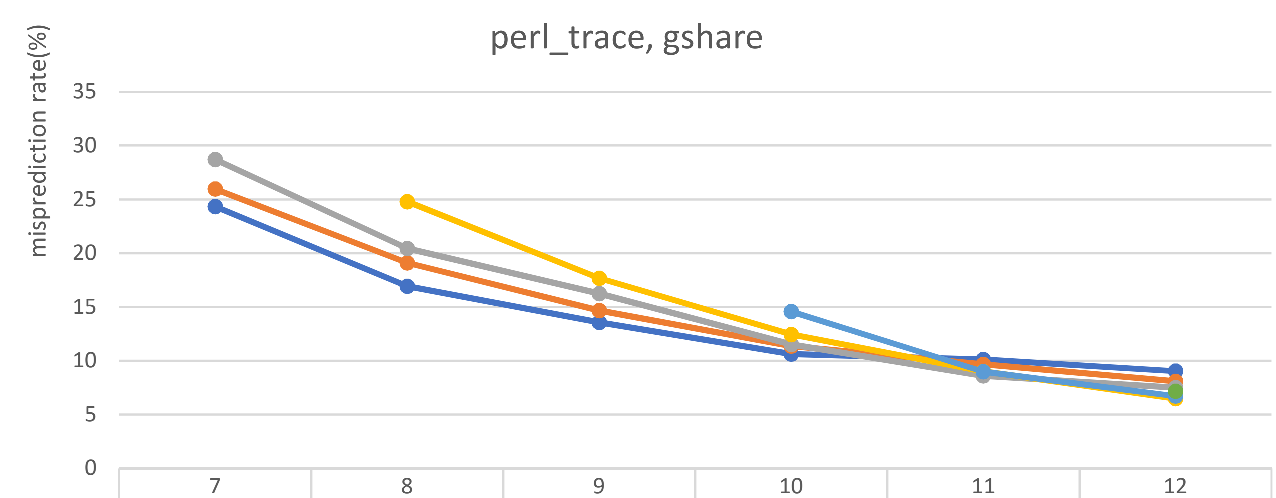
1. Graphs

gcc_trace.txt							
n=2	m	7	8	9	10	11	12
	misprediction rate (%)	28.98	25.18	20.25	16.39	13.71	12.2
n=4	m	7	8	9	10	11	12
	misprediction rate (%)	30.76	26.57	22.43	17.99	14.49	12.23
n=6	m	7	8	9	10	11	12
	misprediction rate (%)	33.22	27.82	24.14	19.36	15.14	12.46
n=8	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	30.56	26.08	21.1	16.47	13
n=10	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	22.77	18.34	14.33
n=12	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	NIL	NIL	15.4



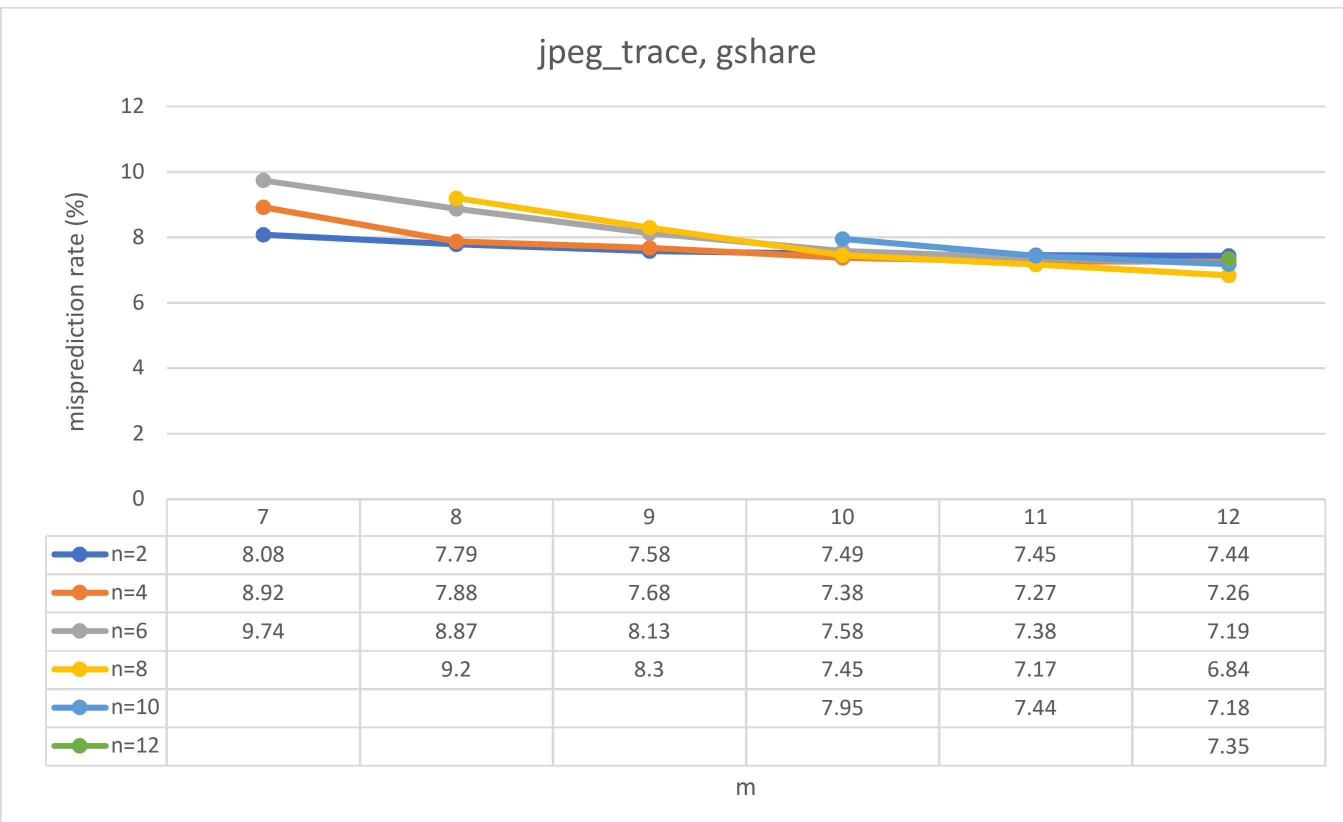
	7	8	9	10	11	12
n=2	28.98	25.18	20.25	16.39	13.71	12.2
n=4	30.76	26.57	22.43	17.99	14.49	12.23
n=6	33.22	27.82	24.14	19.36	15.14	12.46
n=8		30.56	26.08	21.1	16.47	13
n=10				22.77	18.34	14.33
n=12						15.4

perl_trace.txt							
n=2	m	7	8	9	10	11	12
	misprediction rate (%)	24.34	16.92	13.57	10.63	10.11	9.03
n=4	m	7	8	9	10	11	12
	misprediction rate (%)	25.96	19.09	14.68	11.35	9.68	8.09
n=6	m	7	8	9	10	11	12
	misprediction rate (%)	28.71	20.45	16.25	11.52	8.6	7.5
n=8	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	24.79	17.66	12.42	9	6.49
n=10	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	14.57	8.98	6.71
n=12	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	NIL	NIL	7.16



	7	8	9	10	11	12
n=2	24.34	16.92	13.57	10.63	10.11	9.03
n=4	25.96	19.09	14.68	11.35	9.68	8.09
n=6	28.71	20.45	16.25	11.52	8.6	7.5
n=8		24.79	17.66	12.42	9	6.49
n=10				14.57	8.98	6.71
n=12						7.16

jpeg_trace.txt							
n=2	m	7	8	9	10	11	12
	misprediction rate (%)	8.08	7.79	7.58	7.49	7.45	7.44
n=4	m	7	8	9	10	11	12
	misprediction rate (%)	8.92	7.88	7.68	7.38	7.27	7.26
n=6	m	7	8	9	10	11	12
	misprediction rate (%)	9.74	8.87	8.13	7.58	7.38	7.19
n=8	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	9.2	8.3	7.45	7.17	6.84
n=10	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	7.95	7.44	7.18
n=12	m	7	8	9	10	11	12
	misprediction rate (%)	NIL	NIL	NIL	NIL	NIL	7.35



2. Analysis

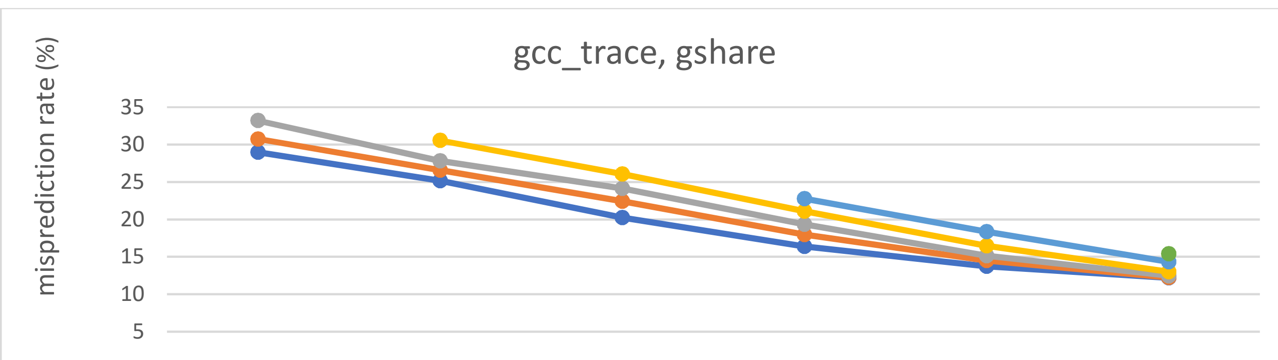
The above graphs and tables illustrate the gshare prediction unit for the above-mentioned benchmarks. Here, the misprediction rate decreases and saturates to a value as the value for m increases. But here the branch history register comes into picture in addition to the branch prediction table and this history is responsible enough to record the history of branches into the shift register and uses the address of the branch instruction and branch history to XOR together. So here the involvement of branch history register helps to reduce the misprediction tendency of the same branch instruction the PC encounters multiple times ie; the gshare scheme significantly provides better prediction accuracy than bimodal.

The above-mentioned benchmarks showcase a similar trend in the variation of misprediction rate vs m. But the influence of n bits of the BHR is slightly different in the three benchmarks. In gcc-trace the misprediction rate increases significantly as the number of bits(n) in the branch history register increases for the same fixed point of m value. But in the case of perl-trace benchmark at values of m beyond 10 the misprediction rate start diminishing slightly which is similar to the case of jpeg-trace file.

Therefore, for benchmarks perl and jpeg, at higher values of PC bits(m) the influence of the branch history register bits on the misprediction rate is evident. ie; for higher PC index bits configuration as the number of bits in the BHR register approaches the value of m the higher the influence of BHR on indexing the prediction table and therefore reduction in the misprediction rate.

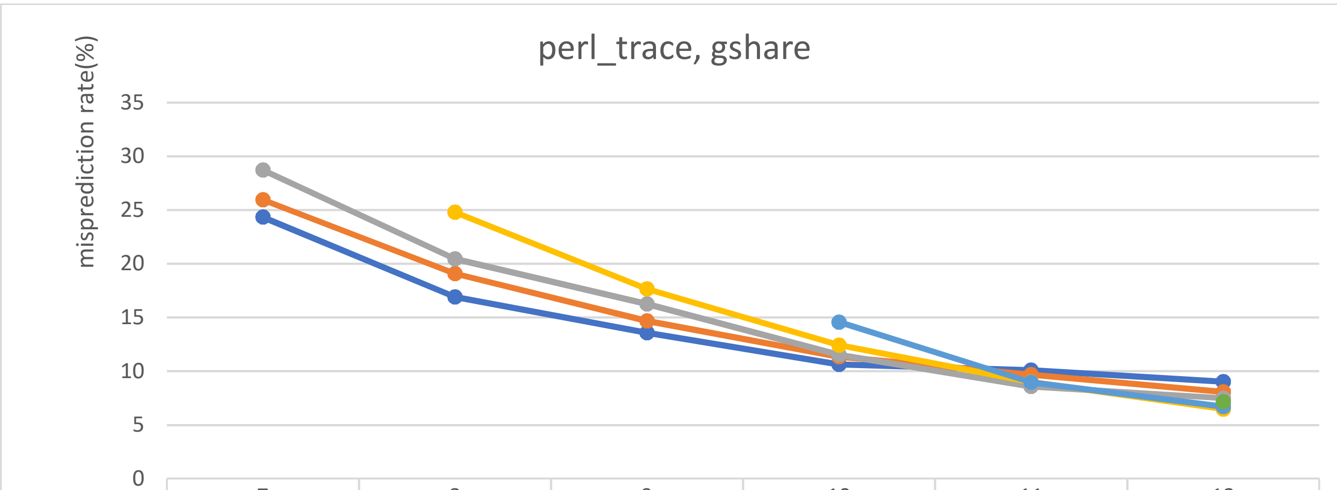
3. Design

Maximum budget of storage : 16KB
Size of storage : 2^m * 2 where m = no: of PC bits and since we are using a 2-bit smith counter.
Predictor cost of bits includes the m and n bits used for the BHT and BHR



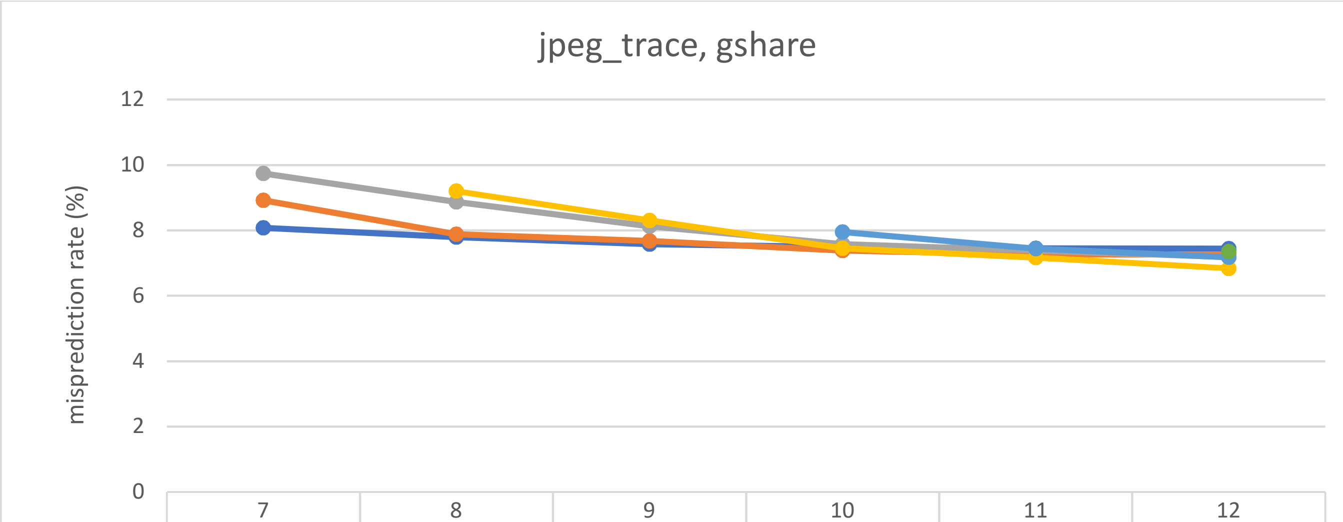
	7	8	9	10	11	12
n=2	28.98	25.18	20.25	16.39	13.71	12.2
n=4	30.76	26.57	22.43	17.99	14.49	12.23
n=6	33.22	27.82	24.14	19.36	15.14	12.46
n=8		30.56	26.08	21.1	16.47	13
n=10				22.77	18.34	14.33
n=12						15.4

For the benchmark, gcc_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 7<m<12. The above trend reveals that at value of m = 12 and n = 2 the misprediction rate is almost equals to highly insignificant. For other values of m, the rate difference is a significant value and higher rate than the value (12.2)% at (m,n) = (12,2).
Therefore, at value m =12 and n=2 the misprediction rate starts saturating to a minimal value of 12.2% ie; additional bits bring no use in the prediction accuracy rather increase the hardware area and predictor cost.



	7	8	9	10	11	12
n=2	24.34	16.92	13.57	10.63	10.11	9.03
n=4	25.96	19.09	14.68	11.35	9.68	8.09
n=6	28.71	20.45	16.25	11.52	8.6	7.5
n=8		24.79	17.66	12.42	9	6.49
n=10				14.57	8.98	6.71
n=12						7.16

For the benchmark, gcc_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 7<m<12. The above trend reveals that at value of m = 12 and n =8 the misprediction rate is almost equals to highly insignificant. For other values of m, the rate difference is a significant value and higher rate than the value (6.49)% at (m,n) = (12,8).
Therefore, at value m =12 and n=8 the misprediction rate starts saturating to a minimal value of 6.49% ie; additional bits bring no use in the prediction accuracy rather increase the hardware area and predictor cost.



	7	8	9	10	11	12
n=2	8.08	7.79	7.58	7.49	7.45	7.44
n=4	8.92	7.88	7.68	7.38	7.27	7.26
n=6	9.74	8.87	8.13	7.58	7.38	7.19
n=8		9.2	8.3	7.45	7.17	6.84
n=10				7.95	7.44	7.18
n=12						7.35

For the benchmark, gcc_trace the above graph is plotted to depict the variation of misprediction rate vs m for values 7<m<12. The above trend reveals that at value of m = 12 and n = 8 the misprediction rate is almost equals to highly insignificant. For other values of m, the rate difference is a significant value and higher rate than the value (6.84)% at (m,n) = (12,8).
Therefore, at value m =12 and n=8 the misprediction rate starts saturating to a minimal value of 6.84% ie; additional bits bring no use in the prediction accuracy rather increase the hardware area and predictor cost.