NC State University

Department of Electrical and Computer Engineering

ECE 463/563: Fall 2017

Project #1: Cache Design, Memory Hierarchy Design

by

<< KENNY KADAYIL PAUL >>

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

Student's electronic signature: KENNY KADAYIL PAUL

(sign by typing your name)

Course number: 563

(463 or 563 ?)

1. Abstract

For this project, a branch predictor simulator is implemented. simulator can simulate various types of branch predictor and configure various cache parameters like number of index bits for the BTB, number of bits for the global branch history register, size of BTB, and set associativity for the BTB. The branch predictors implemented are bimodal, gshare, hybrid enhancing these with a BTB. The replacement policy for all is kept as LRU for simplicity. The miss rates for each of the predictor type is tabulated and is studied for different BTB sizes. For each predictor careful analysis is done and the predictor size with the best peformance is selected keeping in mind the cost incurred in terms of bits to maintain a particular BTB size. The BTB size is fixed at 16 KB for calculations where the prediction is based on a set associative tag array.

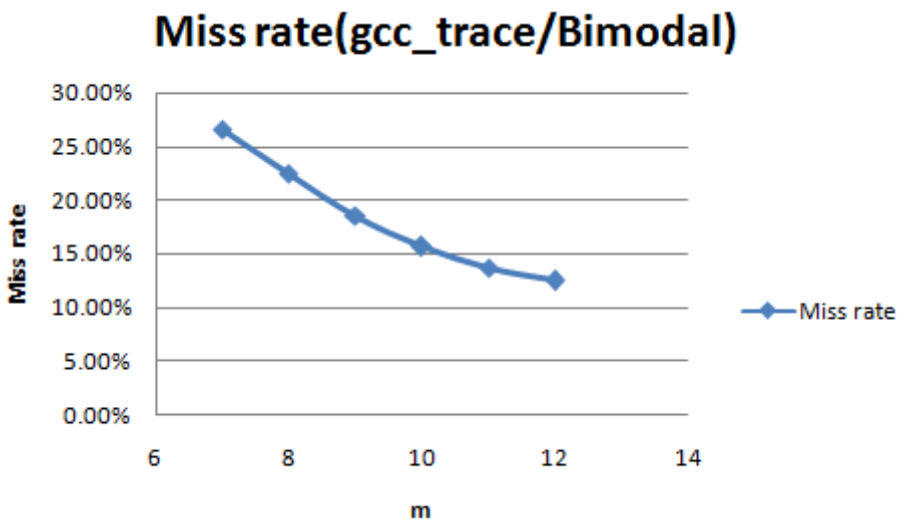
2. Experiments

2.1 Effect of BIMODAL on branch prediction performance

In the BIMODAL predictor, for each of the traces, it is seen that as the prediction table size is increased, the miss rate decreases. For smaller tables, multiple branches may share the same counter, resulting in degraded prediction accuracy. The miss rate saturates at 10% even when the size of the prediction table is increased. A set-associative predictor is also shown to follow the same trend as the simple BIMODAL predictor and seems to saturate at the same accuracy.

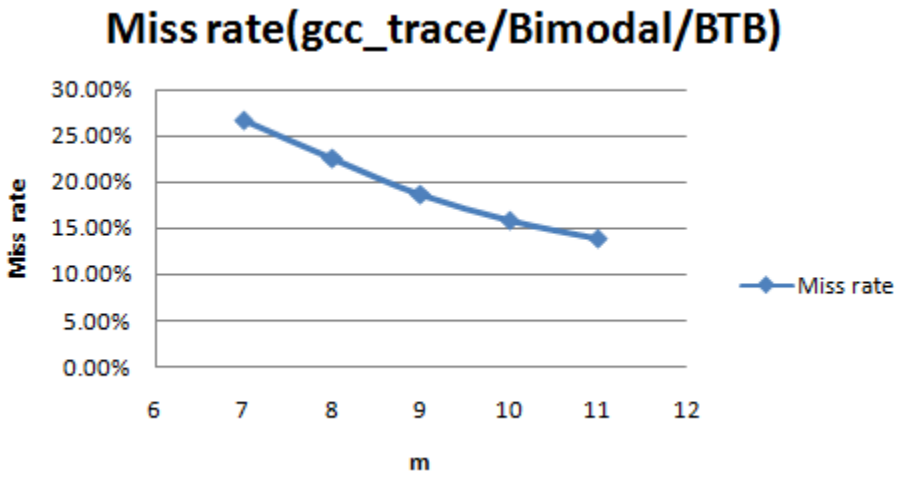
2.1.1.1 BIMODAL predictor for gcc_trace without tag store

No. of m bits	BTB size(B)	Miss rate
7	16	26.65%
8	32	22.43%
9	64	18.49%
10	128	15.67%
11	256	13.65%
12	512	12.47%



2.1.1.2 BIMODAL predictor for gcc_trace with tag store

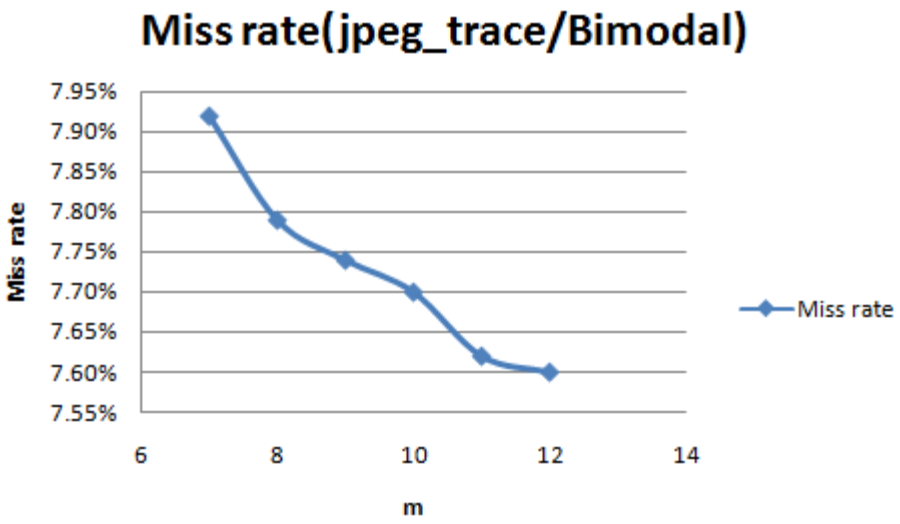
No. of m bits	BTB size(KB)	Set assoc	Miss rate
7	16	32	26.72%
8	16	16	22.52%
9	16	8	18.66%
10	16	4	15.86%
11	16	2	13.95%



Based on the above data, a **simple bimodal predictor** with **m bits equal to 10** gives the maximum performance for minimal cost for **gcc_trace**.

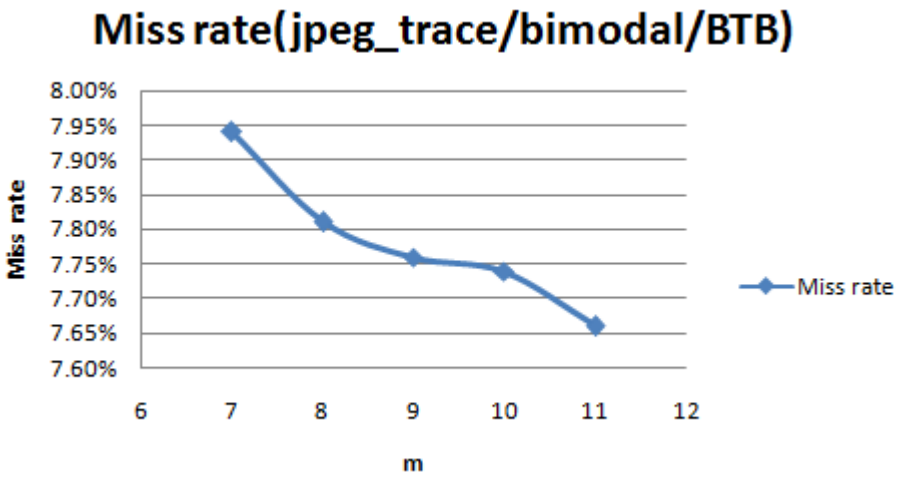
2.1.2.1 BIMODAL predictor for jpeg_trace without tag store

No. of m bits	BTB size(B)	Miss rate
7	16	7.92%
8	32	7.79%
9	64	7.74%
10	128	7.70%
11	256	7.62%
12	512	7.60%



2.1.2.2 BIMODAL predictor for jpeg_trace with tag store

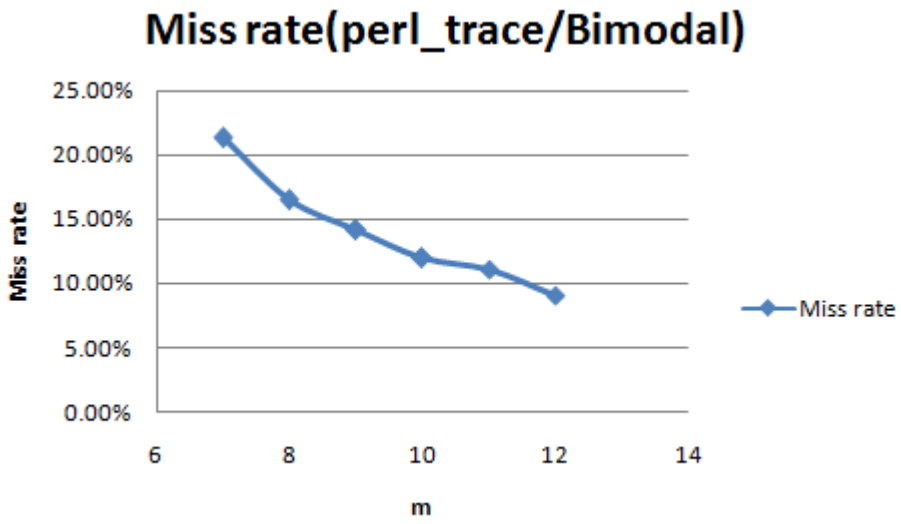
No. of m bits	BTB size(KB)	Set assoc	Miss rate
7	16	32	7.94%
8	16	16	7.81%
9	16	8	7.76%
10	16	4	7.74%
11	16	2	7.66%



Based on the above data, a **simple bimodal predictor** with **m bits equal to 11** gives the maximum performance for minimal cost for **jpeg_trace**.

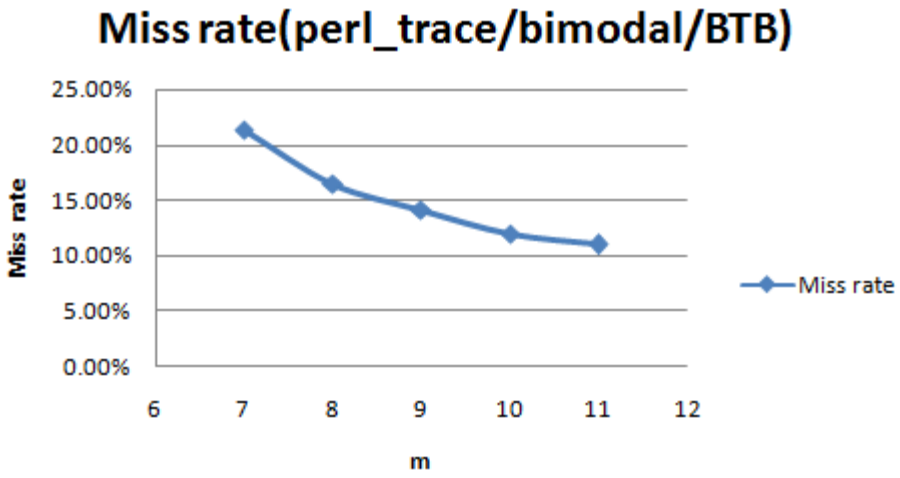
2.1.3.1 BIMODAL predictor for perl_trace without tag store

No. of m bits	BTB size(B)	Miss rate
7	16	21.31%
8	32	16.45%
9	64	14.14%
10	128	11.95%
11	256	11.05%
12	512	9.09%



2.1.3.2 BIMODAL predictor for perl_trace with tag store

No. of m bits	BTB size(KB)	Set assoc	Miss rate
7	16	32	21.29%
8	16	16	16.47%
9	16	8	14.11%
10	16	4	11.93%
11	16	2	10.98%



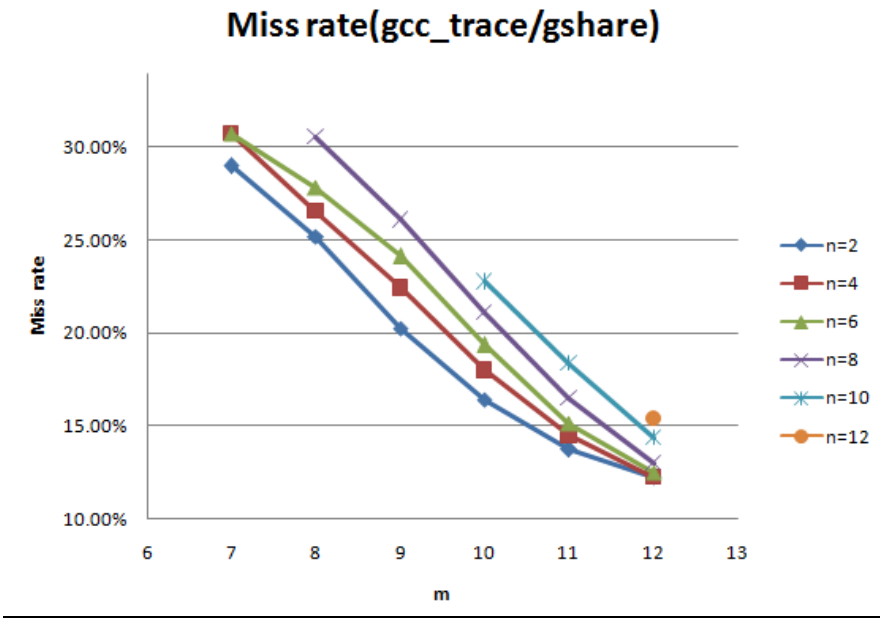
Based on the above data, a **simple bimodal predictor** with **m bits equal to 11** gives the maximum performance for minimal cost for **perl_trace**.

2.2 Effect of GSHARE on branch prediction performance performance

In the GSHARE predictor, for each of the traces, it is seen that as the prediction table size is increased, the miss rate decreases. For small predictors, the gshare scheme performs badly than the bimodal scheme. This trend inverts for higher value of m where gshare fairs better than the bimodal predictor. The bimodal predictor just identifies which branch is current and its behavior and thus it saturates at higher value of m. The n bits in the global history register helps to individually identify the different branches and their behaviour and hence it performs better for higher values of m.

2.2.1.1 GSHARE predictor for gcc_trace without tag store

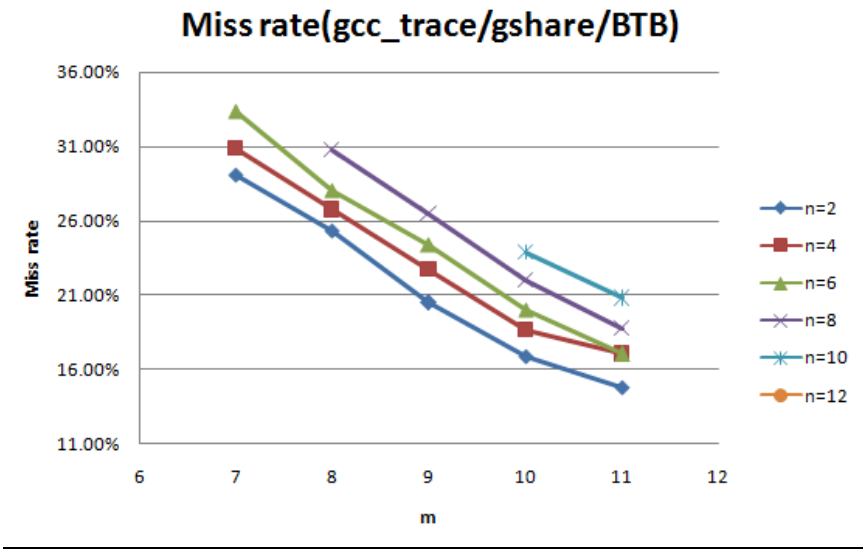
No. of m bits	BTB size(B)	No. of n bits	Miss rate
7	16	2	28.98%
		4	30.76%
		6	33.22%
8	32	2	25.18%
		4	26.57%
		6	27.82%
		8	30.56%
9	64	2	20.25%
		4	22.43%
		6	24.14%
		8	26.08%
10	128	2	16.39%
		4	17.99%
		6	19.36%
		8	21.10%
		10	22.77%
11	256	2	13.71%
		4	14.49%
		6	15.14%
		8	16.47%
		10	18.34%
12	512	2	12.20%
		4	12.23%
		6	12.46%
		8	13.00%
		10	14.33%
		12	15.40%



2.1.1.2 GSHARE predictor for gcc_trace with tag store

No. of m bits	BTB size(B)	Set associativity	No. of n bits	Miss rate
7	16348	32	2	29.07%
		32	4	30.86%
		32	6	33.33%
8	16348	16	2	25.31%
		16	4	26.74%
		16	6	28.01%
		16	8	30.77%
9	16348	8	2	20.47%
		8	4	22.70%
		8	6	24.40%
		8	8	26.50%
10	16348	4	2	16.85%
		4	4	18.66%
		4	6	19.99%
		4	8	21.97%
		4	10	23.87%
11	16348	2	2	14.77%
		2	4	16.00%

		2	6	17.01%
		2	8	18.73%
		2	10	20.76%

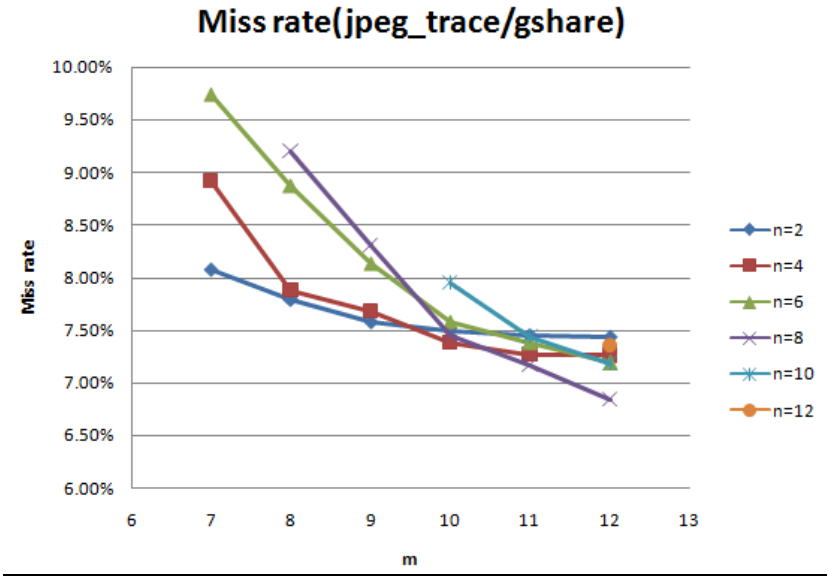


Based on the above data, a **simple gshare predictor** with **m bits equal to 11 and n bits equal to 6** gives the maximum performance for minimal cost for **gcc_trace**.

2.2.2.1 GSHARE predictor for jpeg_trace without tag store

No. of m bits	BTB size(B)	No. of n bits	Miss rate
7	16	2	8.08%
		4	8.92%
		6	9.74%
8	32	2	7.79%
		4	7.88%
		6	8.87%
		8	9.20%
9	64	2	7.58%
		4	7.68%
		6	8.13%

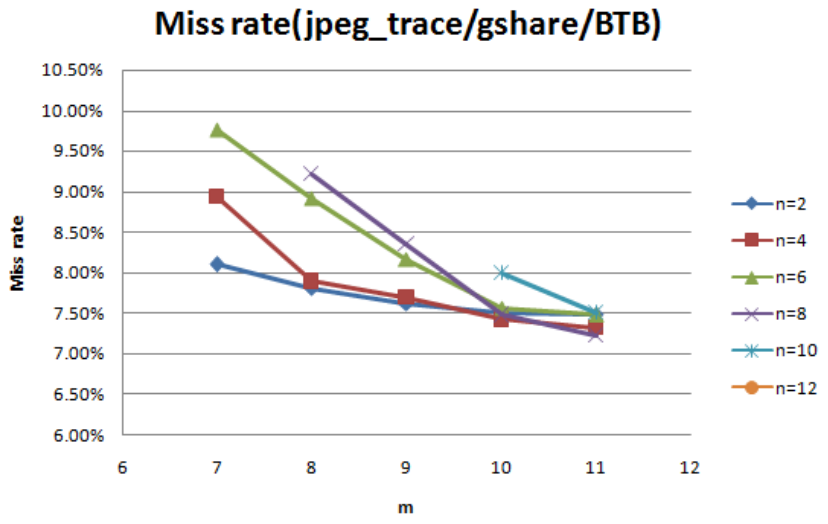
		8	8.30%
10	128	2	7.49%
		4	7.38%
		6	7.58%
		8	7.45%
		10	7.95%
11	256	2	7.45%
		4	7.27%
		6	7.38%
		8	7.17%
		10	7.44%
12	512	2	7.44%
		4	7.26%
		6	7.19%
		8	6.84%
		10	7.18%
		12	7.35%



2.2.2.2 GSHARE predictor for jpeg_trace with tag store

No. of m bits	BTB size(B)	Set associativity	No. of n bits	Miss rate

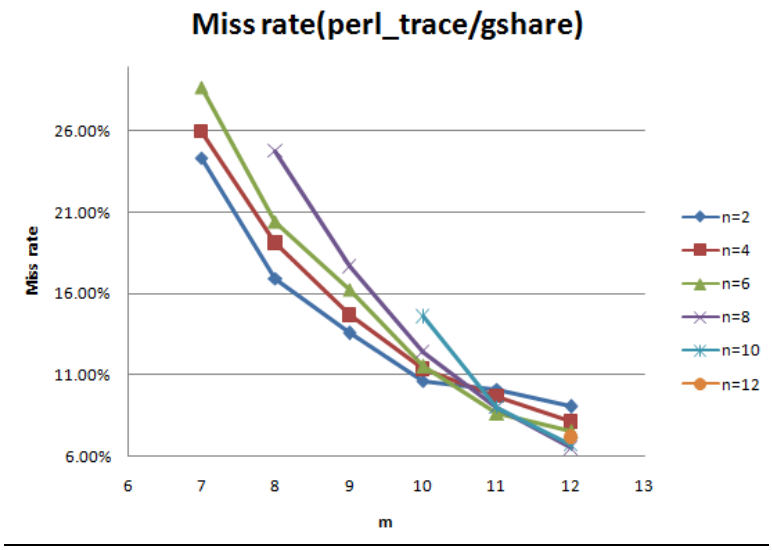
7	16348	32	2	8.10%
		32	4	8.93%
		32	6	9.76%
8	16348	16	2	7.81%
		16	4	7.89%
		16	6	8.92%
		16	8	9.22%
9	16348	8	2	7.62%
		8	4	7.70%
		8	6	8.16%
		8	8	8.35%
10	16348	4	2	7.51%
		4	4	7.42%
		4	6	7.56%
		4	8	7.48%
		4	10	7.99%
11	16348	2	2	7.49%
		2	4	7.32%
		2	6	7.48%
		2	8	7.22%
		2	10	7.50%



Based on the above data, a **simple gshare predictor** with **m bits equal to 10 and n bits equal to 4** gives the maximum performance for minimal cost for **jpeg_trace**.

2.1.1 GSHARE predictor for perl_trace without tag store

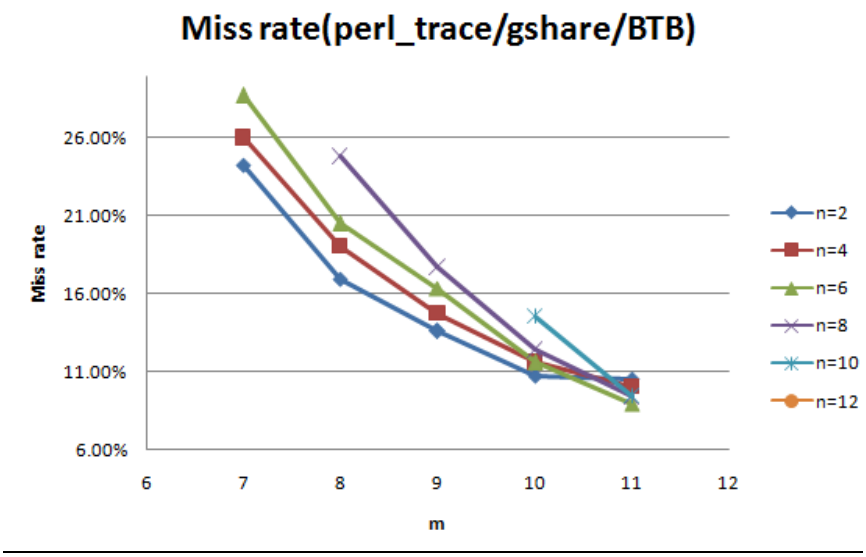
No. of m bits	BTB size(B)	No. of n bits	Miss rate
7	16	2	24.34%
		4	25.96%
		6	28.71%
8	32	2	16.92%
		4	19.09%
		6	20.45%
		8	24.79%
9	64	2	13.57%
		4	14.68%
		6	16.25%
		8	17.66%
10	128	2	10.63%
		4	11.35%
		6	11.52%
		8	12.42%
		10	14.57%
11	256	2	10.11%
		4	9.68%
		6	8.60%
		8	9.00%
		10	8.98%
12	512	2	9.03%
		4	8.09%
		6	7.50%
		8	6.49%
		10	6.71%
		12	7.16%



2.1.1 GSHARE predictor for perl_trace with tag store

No. of m bits	BTB size(B)	Set associativity	No. of n bits	Miss rate
7	16348	32	2	24.25%
		32	4	26.01%
		32	6	28.71%
8	16348	16	2	16.94%
		16	4	19.06%
		16	6	20.49%
		16	8	24.81%
9	16348	8	2	13.59%
		8	4	14.76%
		8	6	16.31%
		8	8	17.68%
10	16348	4	2	10.71%
		4	4	11.45%
		4	6	11.56%
		4	8	12.39%
		4	10	14.55%
11	16348	2	2	10.55%
		2	4	10.02%
		2	6	8.93%
		2	8	9.38%

		2	10	9.40%



Based on the above data, a **simple gshare predictor** with **m bits equal to 11 and n bits equal to 4** gives the maximum performance for minimal cost for **perl_trace**.