Project 2 Branch Prediction Simulator Report

Alexander Bagherzadeh

alexbag@knights.ucf.edu

EEL4768: Comp Architecture

Due Date: 11-30-21

1.0 Project Description

The objective of this project is to be able to simulate the process of branch prediction using GShare. In this simulation, three arguments are used to change the miss prediction rate of the branch prediction software. You can change the size of the index of the branch prediction table, the size of the global history buffer, and you can provide a trace file to read simulate input addresses and results. This simulation can then help us figure out the relationship between each variable and how it affects the miss prediction rate.

2.0 Program Design

I wrote my program in c code. I started my program by defining some structs to store my data to use throughout the program. The one labeled predictionTableData held the informadion regarding the index size of the branch prediction table and the information for the global history buffer like the bits and the actual value. The struct labeled predictionData holds the expected path to take and the actual path that was taken during the parsing of each line in the trace file. MaskData holds the information regarding M or the index of the prediction table as well as the calculated mask to use over the address to get the index. The one labeled HitData stores all the counting information for hits and misses. Finally, addressData stores all the parsed information from the file as well as the file pointer to the file.

After the structures, I initialized everything in the start of my main function. Before cycling through the file, I get the index using the m bits given in the argument and generate my prediction table while initializing all spots as weakly taken. Then, I calculate the index mask and start looping through the file. For each line in the file, I parse the actual path taken as well as the address and calculate the current index. Once at the appropriate index I determine the expected path and see if it’s a hit or a miss. Then I update the GHB and the slot of the prediction table. Once each line in the file is read, the loop ends, and I print out the results for the miss prediction rate.

3.0 Data Tables from Tests

The tables holding the results for all tests to determine requested results shown below.

Test A: M = 4 Bits	MCF	GOBMK
N Bits	Miss Prediction Rate	Miss Prediction Rate
1	24.71%	0.77%
2	26.86%	0.87%
3	29.36%	0.86%
4	31.72%	0.82%

Test B: N = 4 Bits	MCF	GOBMK
M Bits	Miss Prediction Rate	Miss Prediction Rate
4	31.72%	0.82%
5	26.56%	0.67%
6	19.81%	0.60%
7	12.4%	0.58%

Test A: N = 0 Bits	MCF	GOBMK
M Bits	Miss Prediction Rate	Miss Prediction Rate
4	23.76%	0.69%
5	20.83%	0.66%
6	15.07%	0.67%
7	10.63%	0.60%

4.0 Learning Coverage

There are many technical topics that were covered and learned in this project.

- Understanding the concepts of Branch prediction.
- Using a high-level programming language to simulate and test.
- Using an array and bitwise operations to cycle through the array.
- Implementing GShare properties.

5.0 Program Code

The code will be turned in in tandem to this report.

6.0 Program Tests

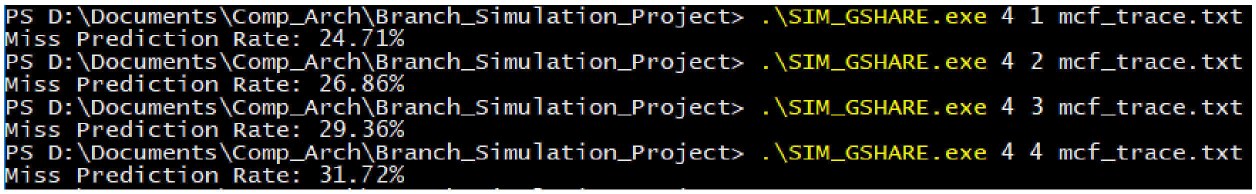
The program was used to determine the results requested in the project instructions. (Paraphrased below)

- A- Keep M at 4 bits and vary N from 1 to 4 bits. Determine miss prediction rate for both MCF and GOBMK trace files.
- B- Keep N at 4 bits and vary M from 4 to 7 bits. Determine miss prediction rate for both MCF and GOBMK trace files.
- C- Keep N at 0 bits and vary M from 4 to 7 bits. Determine miss prediction rate for both MCF and GOBMK trace files.

7.0 Test Results

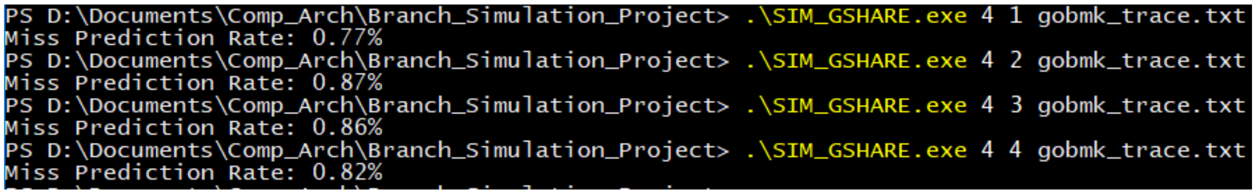
Test A: ---

- MCF Results in Command Prompt:



```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 1 mcf_trace.txt
Miss Prediction Rate: 24.71%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 2 mcf_trace.txt
Miss Prediction Rate: 26.86%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 3 mcf_trace.txt
Miss Prediction Rate: 29.36%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 4 mcf_trace.txt
Miss Prediction Rate: 31.72%
```

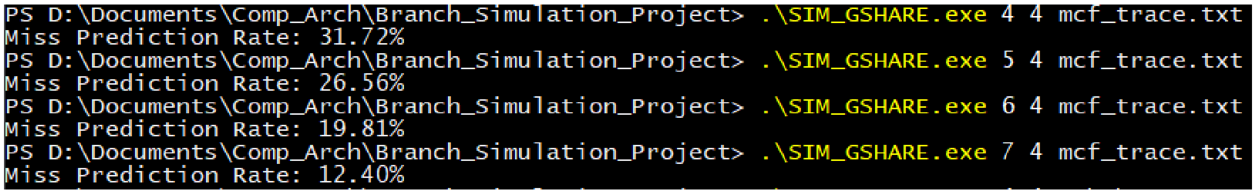
- GOBMK Results in Command Prompt:



```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 1 gobmk_trace.txt
Miss Prediction Rate: 0.77%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 2 gobmk_trace.txt
Miss Prediction Rate: 0.87%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 3 gobmk_trace.txt
Miss Prediction Rate: 0.86%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 4 gobmk_trace.txt
Miss Prediction Rate: 0.82%
```

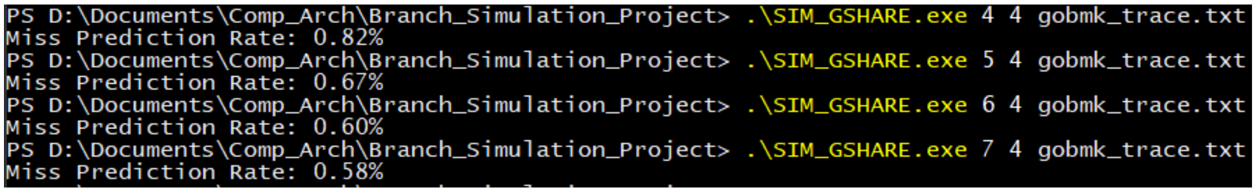
Test B: --

- MCF Results in Command Prompt:



```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 4 mcf_trace.txt
Miss Prediction Rate: 31.72%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 5 4 mcf_trace.txt
Miss Prediction Rate: 26.56%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 6 4 mcf_trace.txt
Miss Prediction Rate: 19.81%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 7 4 mcf_trace.txt
Miss Prediction Rate: 12.40%
```

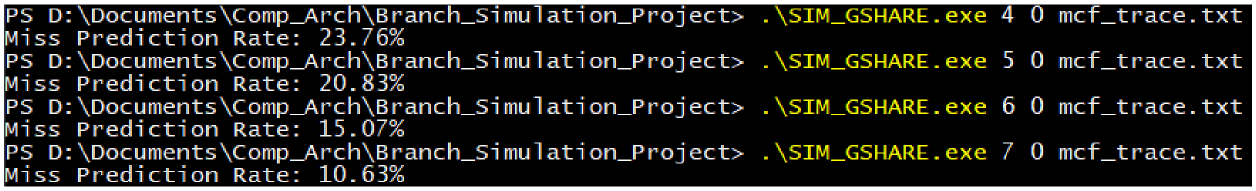
- GOBMK Results in Command Prompt:



```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 4 gobmk_trace.txt
Miss Prediction Rate: 0.82%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 5 4 gobmk_trace.txt
Miss Prediction Rate: 0.67%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 6 4 gobmk_trace.txt
Miss Prediction Rate: 0.60%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 7 4 gobmk_trace.txt
Miss Prediction Rate: 0.58%
```

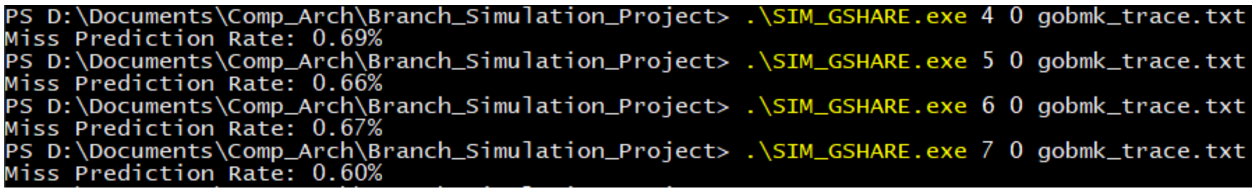
Test C: --

- MCF Results in Command Prompt:



```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 0 mcf_trace.txt
Miss Prediction Rate: 23.76%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 5 0 mcf_trace.txt
Miss Prediction Rate: 20.83%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 6 0 mcf_trace.txt
Miss Prediction Rate: 15.07%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 7 0 mcf_trace.txt
Miss Prediction Rate: 10.63%
```

- GOBMK Results in Command Prompt:

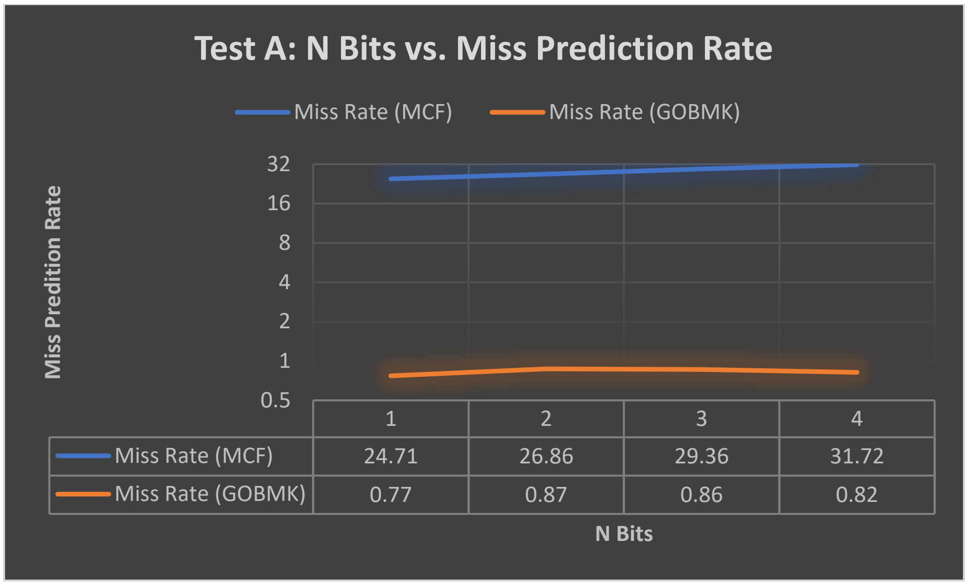


```
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 4 0 gobmk_trace.txt
Miss Prediction Rate: 0.69%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 5 0 gobmk_trace.txt
Miss Prediction Rate: 0.66%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 6 0 gobmk_trace.txt
Miss Prediction Rate: 0.67%
PS D:\Documents\Comp_Arch\Branch_Simulation_Project> .\SIM_GSHARE.exe 7 0 gobmk_trace.txt
Miss Prediction Rate: 0.60%
```

8.0 Data Analysis

The results for all tests are plotted and analyzed below.

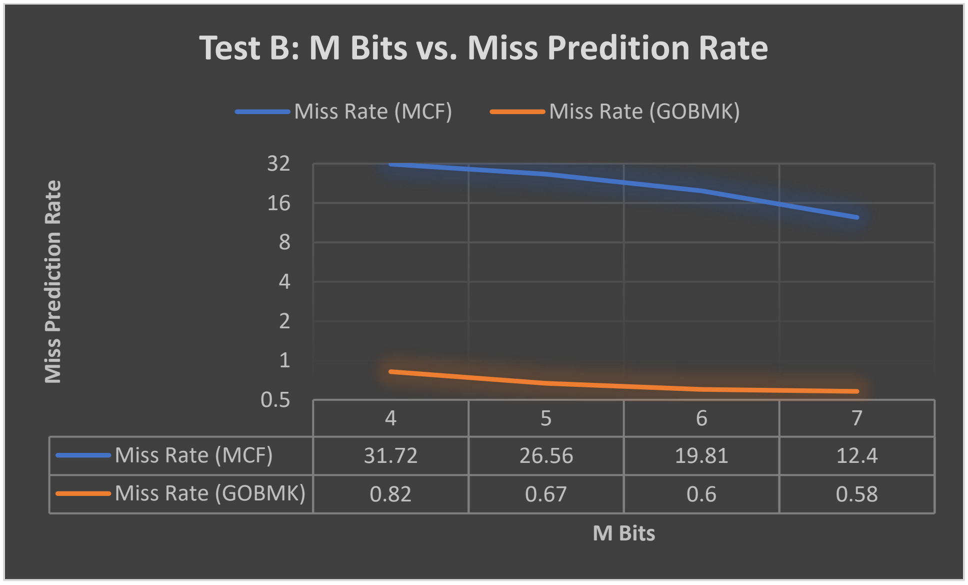
Test A: --



	1	2	3	4
Miss Rate (MCF)	24.71	26.86	29.36	31.72
Miss Rate (GOBMK)	0.77	0.87	0.86	0.82

- We can see that ass we increase the size of the global history buffer; The miss rate increases slightly for the MCF file and starts to increase, then decreases for the GOBMK file. Overall, there is not much of a change in the miss prediction rate for either file. This shows us that changing the size of the global history buffer doesn't have much of an affect on the overall performance of the branch prediction.

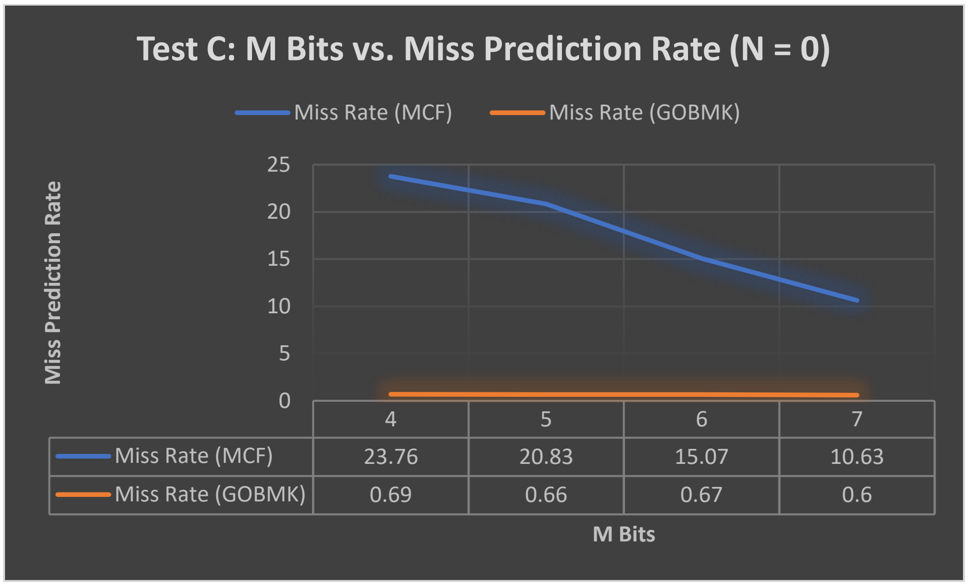
Test B: --



	4	5	6	7
Miss Rate (MCF)	31.72	26.56	19.81	12.4
Miss Rate (GOBMK)	0.82	0.67	0.6	0.58

- When changing the size of the branch prediction table, we can see for both files there was a decrease in miss rate. Especially for the MCF file, the decrease was a little more extreme when changing from 5 to 7 bits. This tells us that increasing the size of the branch prediction table is effective in lowing miss prediction rate.

Test C: ---



	4	5	6	7
Miss Rate (MCF)	23.76	20.83	15.07	10.63
Miss Rate (GOBMK)	0.69	0.66	0.67	0.6

- This test was like the previous test be, but the size of the global history buffer was 0 instead of 1. This, in other words, removed the function of the global history buffer. We can see in the graph that the miss prediction rate for the GOBMK file was almost constant and the miss prediction rate for the MCF file decreased significantly as we increased the size of the prediction table. We can see from these results, that depending on specific applications, having no GHB and increasing the size of the branch prediction table can have a serious affect in lowering the miss prediction rate.

9.0 References

9.1 Project Instructions and Text file containing validation data.

The project instructions and text file can be found under this assignment's description in the EEL 4768 webcourses.

9.2 Textbook

Computer Organization and Design textbook written by Patterson and Hennessy which can be found in the course syllabus on the home page.

9.3 C Code Libraries

C Code libraries to use built in functions including ones to read the files