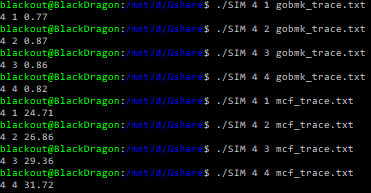
**Project 1: Gshare Simulator**

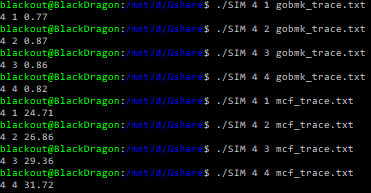
**Part A (M fixed at 4 bits, Vary N to 1, 2, 3 and 4):**

**gobmk\_trace.txt**

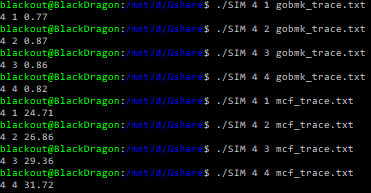
At N = 1:



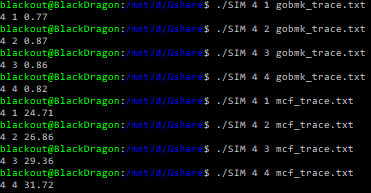
At N = 2:



At N = 3:



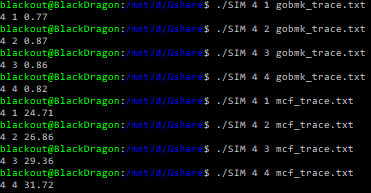
At N = 4:



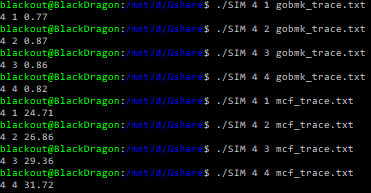
Plot:

**mcf\_trace.txt**

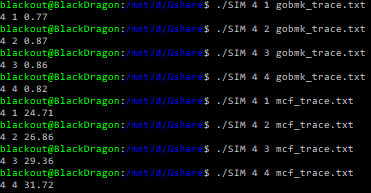
At N = 1:



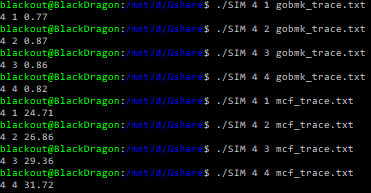
At N = 2:



At N = 3:



At N = 4:



Plot:

Explanation of Part A:

As you can see with M fixed at 4, each of the traces carry separate outcomes for misprediction rates. With mcf you have an increasing percentage for the higher increase in the global history bit, with gobmk you observe the opposite. With gobmk, you have many addresses that are duplicated looking for the same outcome as the previous line. This architecture causes you to reach the same global history bit while checking the same address you just recently checked, thus reducing randomness. Therefore, it explains why you see such a low misprediction rate. The reason for mcf being so high, is that you get much more randomness due to different addresses and outcomes. This explanation further explains why on all future parts, there is such a variation in misprediction rates between the two traces is.

Keeping all this in mind, when it comes to the increase in the misprediction rate of increasing global history bits (mcf trace), you are increasing the chance of randomness. In doing so, you cause the Gshare predictor to find a random address, which would give you a random outcome on prediction, usually resulting in a misprediction since the spontaneity is amplified. With gobmk, you have equivalent addresses resulting in a much more predictable index, thus give you a more accurate predictor.

**Part B (N fixed at 4 bits, Vary M to 4, 5, 6 and 7):**

**gobmk\_trace.txt**

At M = 4:



At M = 5:



At M = 6:



At M = 7:



Plot:

**mcf\_trace.txt**

At M = 4:



At M = 5:



At M = 6:



At M = 7:



Plot:

Explanation of Part B:

Much of part B and C is instinctive. Both graphs show decreasing misprediction rates as M (entries) increases. This is due to size of the table being used. When you use a smaller table and try to use the predictor, you tend to hit the same indexes with varying outcomes. This results in a higher misprediction rate. When you increase the size of your table, you have more of a variety in which to access, meaning less prevalent changes for each index as with a smaller table.

**Part C (N fixed at 0, Vary M to 4, 5, 6 and 7):**

**gobmk\_trace.txt**

At M = 4:



At M = 5:



At M = 6:



At M = 7:



Plot:

**mcf\_trace.txt**

At M = 4:



At M = 5:



At M = 6:



At M = 7:



Plot:

Explanation of Part C:

The explanation for part C takes some explanation from A and B. With 0 global history bits, your global history register is constant. It then reduces spontaneity of the entire process which in turn gives you better misprediction rates than what you gain in the previous parts. When you increase the size of the table, you help reduce change per index. These two factors cause a higher chance in reducing the misprediction rate.