**NC State University**

**Department of Electrical and Computer Engineering**

**ECE 463/563: Fall 2017**

# Project #2: Branch Prediction

**by**

**Viplove Rakheja**

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

Student's electronic signature: __________**Viplove Rakheja**_________________
(sign by typing your name)

Course number: ______563__________

- ## Introduction:

As a part of this project, a branch predictor simulator was constructed and was used to design branch predictor well suited for SPECint95 benchmarks. There were total of 3 type of branch predictor that were designed which were: Bimodal, Gshare and Hybrid branch predictors. Apart from that, a BTB support is also made available which was n way set associative and used LRU replacement policy to evict the blocks.

- ## Formulas used :

There were few formulas used for implementing the branch predictor. The program counter always 4-byte aligned starting from 0. Since the program counter always increment by a count of 4, i.e. why the last two bits of the program counter will be always be zero and hence they are not considered for index calcultations.

  - **Index Calculation for bimodal branch predictor**:

    *programcounter >>= 2;*
    *index = m-bit lower order mask*
    *index = index & programcounter*

  - **Index Calculation for gshare branch predictor** :

    Gshare branch predictor consists of a GHR (Global History Register) of m bits which is used to calculate the index for the branch predictor.

    *programcounter >>= 2;*
    *index = m-bit lower order mask*
    *index = index & programcounter*

    *finalIndex = index ^ GHR;*

    After every branch whether its taken or not taken, GHR gets shifted by 1 bit and the MSB is appended with 0 (if not taken) or 1 (if taken).

  - **Miss Prediction Rate:**

    Misprediction rate is the ratio of number of branch mispredictions by the predictor to the total number of branches. The lower the misprediction rate, the better is the predictor's performance. This rate is one of the important attribute of a predictor which helps in understanding the characteristics of it.

$$Missprediction\ Rate = \frac{Number\ of\ Misspredictions}{Total\ number\ of\ Predictions}$$

- **Predictor Table size :**

  The predictor table size is determined by the number of bits that are available to index the table multiplied by the number of bits used in the counter:

$$Predictor\ Table\ Size = (\#\ of\ bits\ in\ counter) * (2\text{^}mbits)$$

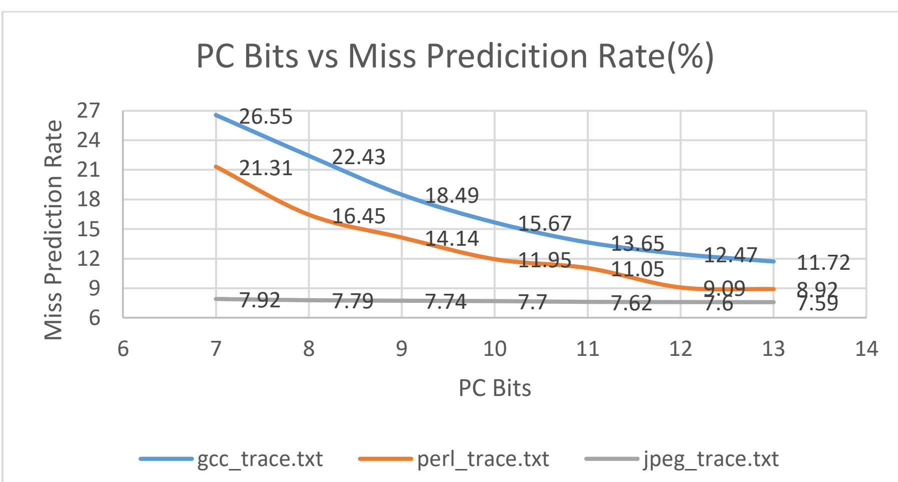
- ## Experimental Data:

  - **Bimodal Branch Predictor :**
    In order to simulate the Bimodal Branch Predictor, the input that followed was:

$$java\ sim\ bimodal\ PCBits\ BTBSize\ BTBAssoc < tracefilename >$$

Overall the simulation was made on three trace files i.e. gcc_trace.txt, jpeg_trace.txt and per_trace.txt and the miss prediction rate and number of PC Bits was taken into account as the performance metrics. The data that is calculated is presented in the table below:

| Miss Prediction Rate | | | |
|---|---|---|---|
| **PCBits** | **gcc_trace.txt** | **perl_trace.txt** | **jpeg_trace.txt** |
| 7 | 26.55 | 21.31 | 7.92 |
| 8 | 22.43 | 16.45 | 7.79 |
| 9 | 18.49 | 14.14 | 7.74 |
| 10 | 15.67 | 11.95 | 7.7 |
| 11 | 13.65 | 11.05 | 7.62 |
| 12 | 12.47 | 9.09 | 7.6 |
| 13 | 11.72 | 8.92 | 7.59 |



- ## Gshare Branch Predictor:

In order to simulate the Bimodal Branch Predictor, the input that followed was:

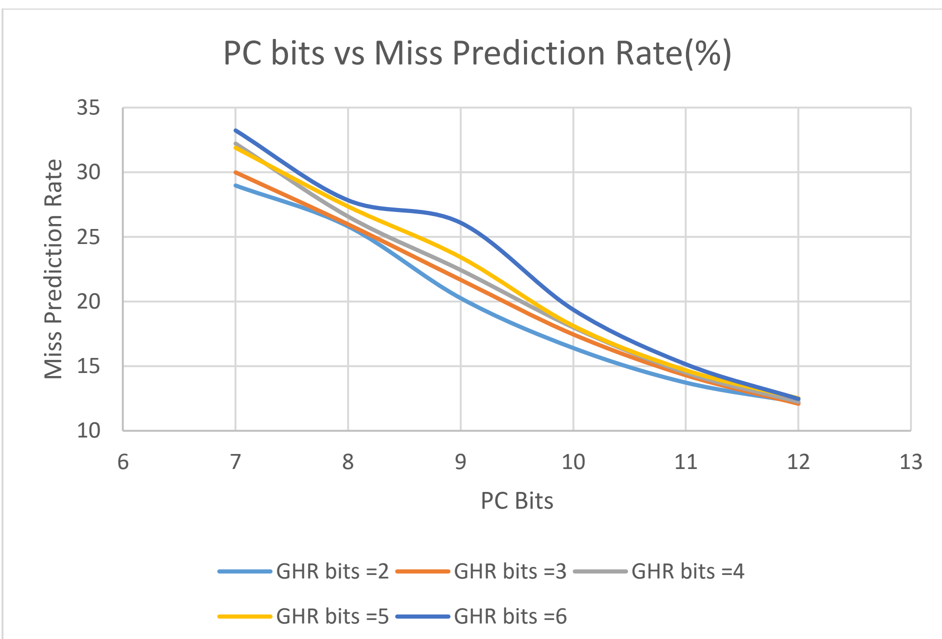
$java\ sim\ gshare\ PCBits\ BHRbits\ BTBSize\ BTBAssoc < tracefilename >$

BHRbits- the number of bits of Branch History Register

Overall the simulation was made on three trace files i.e. gcc_trace.txt, jpeg_trace.txt and per_trace.txt and the miss prediction rate, number of PC Bits along with various values of branch history register bits were taken into account as the performance metrics. The data that is calculated is presented in the table below:

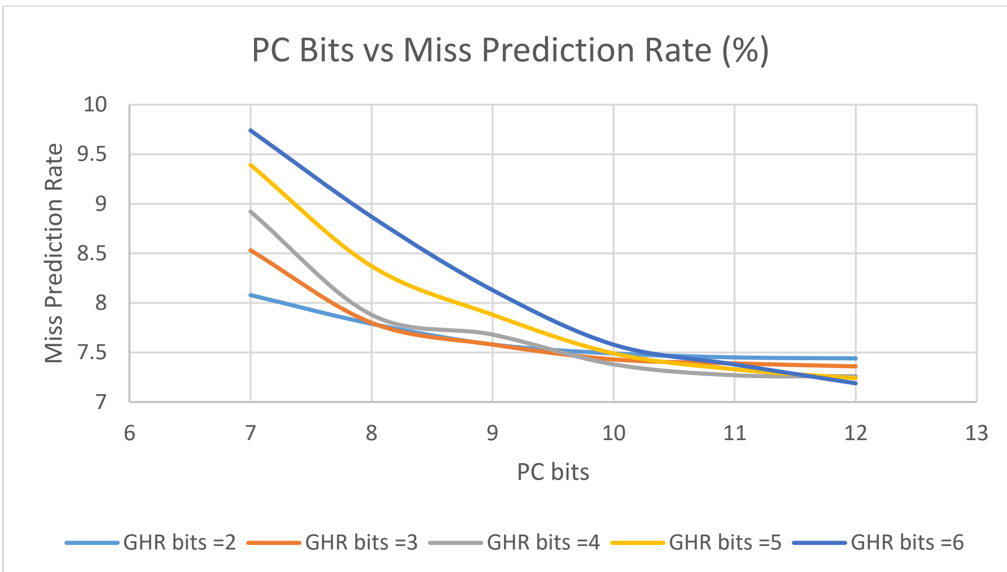
- **For gcc_trace.txt trace file :**

| Miss Prediction Rate | | | | | |
|---|---|---|---|---|---|
| **PCBits** | **GHR bits =2** | **GHR bits =3** | **GHR bits =4** | **GHR bits =5** | **GHR bits =6** |
| 7 | 28.98 | 29.98 | 32.22 | 31.9 | 33.22 |
| 8 | 25.81 | 25.96 | 26.57 | 27.35 | 27.82 |
| 9 | 20.25 | 21.67 | 22.43 | 23.42 | 26.08 |
| 10 | 16.39 | 17.45 | 17.99 | 18.11 | 19.36 |
| 11 | 13.71 | 14.29 | 14.49 | 14.67 | 15.14 |
| 12 | 12.2 | 12.09 | 12.23 | 12.5 | 12.46 |



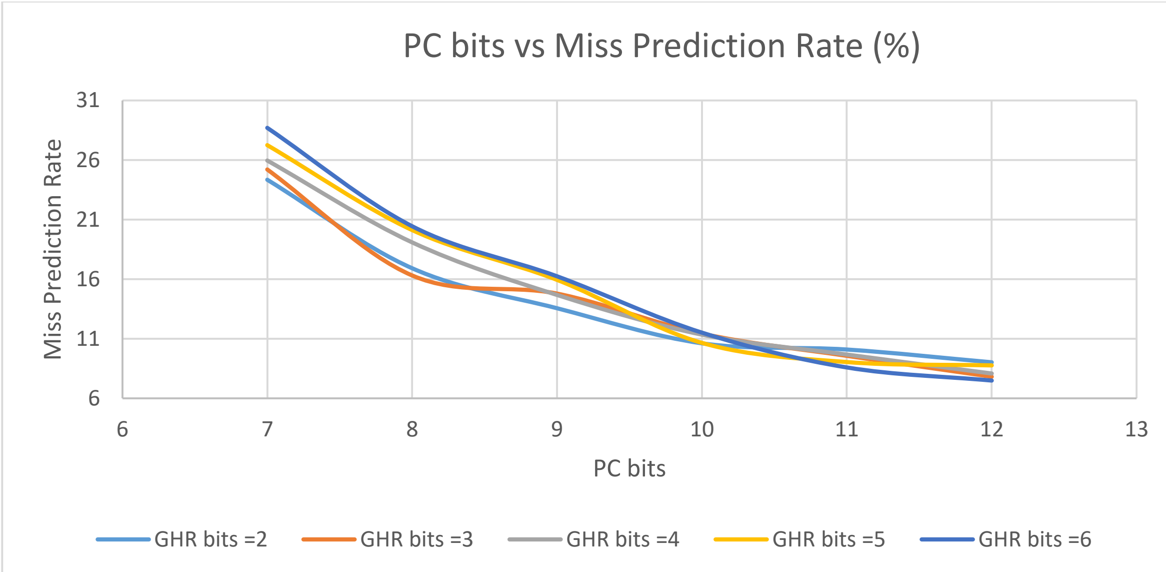
- **For jpeg_trace.txt trace file :**

| Miss Prediction Rate | | | | | |
|---|---|---|---|---|---|
| **PCBits** | **GHR bits =2** | **GHR bits =3** | **GHR bits =4** | **GHR bits =5** | **GHR bits =6** |
| 7 | 8.08 | 8.53 | 8.92 | 9.39 | 9.74 |
| 8 | 7.79 | 7.8 | 7.88 | 8.37 | 8.87 |
| 9 | 7.58 | 7.58 | 7.68 | 7.88 | 8.13 |
| 10 | 7.49 | 7.43 | 7.38 | 7.49 | 7.58 |
| 11 | 7.45 | 7.39 | 7.27 | 7.33 | 7.38 |
| 12 | 7.44 | 7.36 | 7.26 | 7.24 | 7.19 |




- **For perl_trace.txt trace file** :

| Miss Prediction Rate | | | | | |
|---|---|---|---|---|---|
| **PCBits** | **GHR bits =2** | **GHR bits =3** | **GHR bits =4** | **GHR bits =5** | **GHR bits =6** |
| 7 | 24.34 | 25.21 | 25.96 | 27.26 | 28.71 |
| 8 | 16.92 | 16.32 | 19.09 | 20.13 | 20.45 |
| 9 | 13.57 | 14.8 | 14.68 | 15.95 | 16.25 |
| 10 | 10.63 | 11.44 | 11.35 | 10.66 | 11.52 |
| 11 | 10.11 | 9.58 | 9.68 | 9.05 | 8.6 |
| 12 | 9.04 | 7.83 | 8.09 | 8.77 | 7.5 |




- ## Hybrid Branch Predictor:

In order to simulate the Bimodal Branch Predictor, the input that followed was:

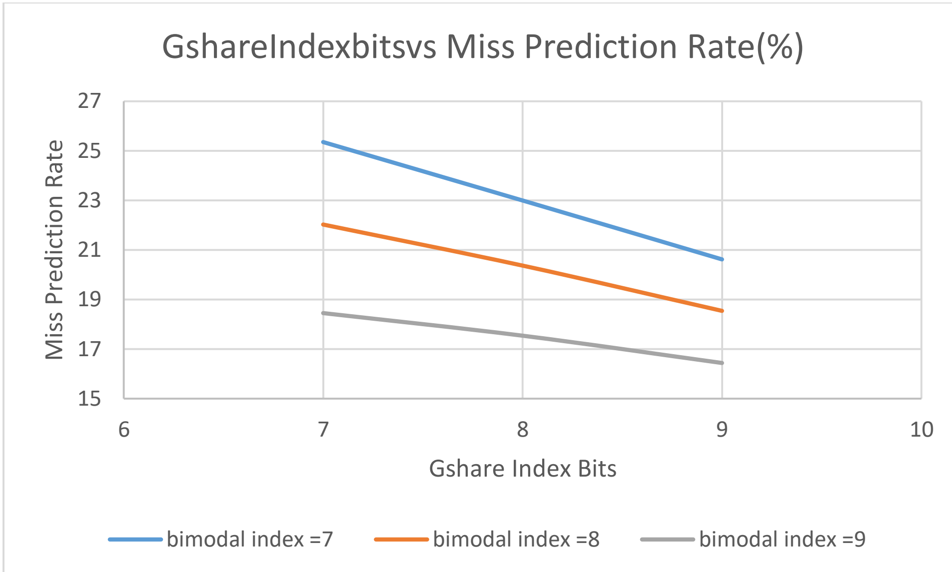
$$java\ sim\ hybrid\ PCBits\ gshareindexbits\ BHRbits\ bimodalindexbits\ BTBSize\ BTBAssoc < tracefilename >$$

BHRbits- the number of bits of Branch History Register

Overall the simulation result were fetched on a single trace file i.e. gcc_trace.txt file and the parameters BHR bits was kept constant and was kept 4 bits while other parameters like PC bits, gshare index bits, bimodal index bits and miss prediction rate was taken as performance metrics.

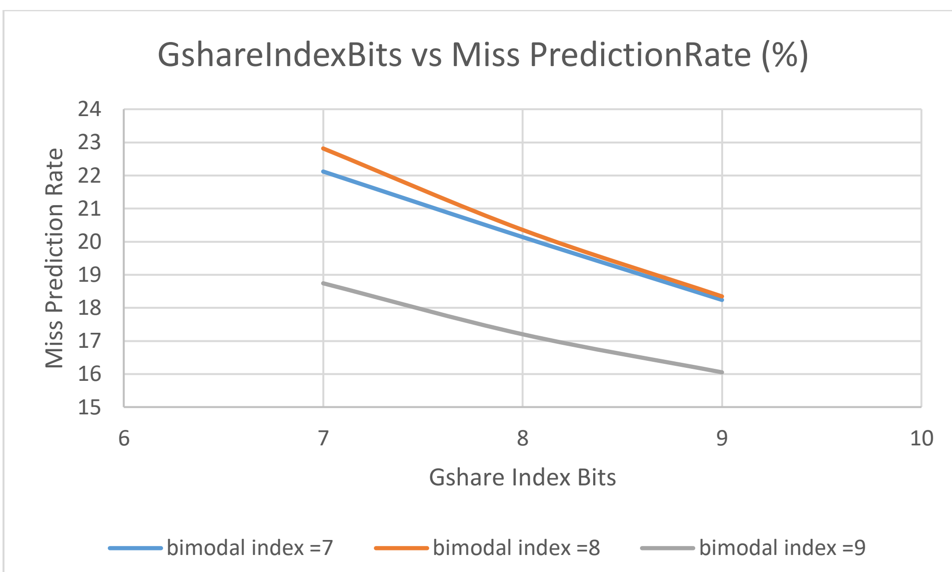
- **PC bits equal to 7**

| Miss Prediction Rate | | | |
|---|---|---|---|
| **gsharebits** | **bimodal index =7** | **bimodal index =8** | **bimodal index =9** |
| 7 | 25.36 | 22.03 | 18.45 |
| 8 | 23 | 20.37 | 17.54 |
| 9 | 20.62 | 18.54 | 16.44 |




- **PC bits equal to 8**

| Miss Prediction Rate | | | |
|---|---|---|---|
| **gsharebits** | **bimodal index =7** | **bimodal index =8** | **bimodal index =9** |
| 7 | 22.12 | 22.82 | 18.74 |
| 8 | 20.14 | 20.36 | 17.2 |
| 9 | 18.24 | 18.35 | 16.05 |



- ## Branch Predictor Performance Analysis :

  The performance of the Branch Predictor is highly dependent upon the parameters like number of bits available to store the history of predictions, size of the predictor table, size of the branch target buffer and the branch instruction patterns themselves Let us see below points in which a brief comparison of the above simulation results have been provided:

  - **Bimodal Branch Predictor :**

    Miss prediction rate is inversely proportional to the number of bits as it can be inferred from the above graphs that as the value of ‘m’ index increases, the miss prediction rate reduces significantly as it reduces the collision domain of the branch instructions in the predictor table.
    Also, if we have a look at the three different trace files i.e. jpeg_trace.txt, gcc_trace.txt, perl_trace.txt, we could see that jpeg_trace.txt had the minimum miss prediction rate for the same size of predictor table. This is due to the fact that jpeg_trace.txt has the branch instructions in a widely distributed manner thus reducing the collisions within the predictor table

    - **Best Design :**

      For different trace files, different results can be obtained,

      **Gcc_trace.txt :** For this particular trace file, the branch predictor will work best if the size of index table is as large as possible i.e. m=13.

      **perl_trace.txt :** For this particular trace file, the branch predictor will work best if the size of index table is as large as possible i.e. m=13

      **jpeg_trace.txt :** For this particular trace file, the branch predictor will work best if the size of index table is as large as possible i.e. m=10. Reason for selecting m=10 is due to the fact that as the m goes beyond 10, the change is miss prediction rate is not deviating a lot, so its better to go with the design having less size of Index table as it reduces the cost and time of prediction as well.

- **Gshare Branch Predictor :**

  Similar to bimodal branch predictor, as the value of gshare index bits increases i.e. the size of predictor table increases, the miss prediction rate decreases significantly. But as we increase the number of bits in the BHR, the overall miss prediction rate increases.

  However, there is a factor that is needed to be considered in case of Gshare Branch Predictor, i.e. Time. As the size of predictor table increases, the time complexity increases a lot making the system slow.

  - **Best Design:**

    **Gcc_trace.txt :** For the current trace file, the best suited design will be with the gshare index bits to be 12 and the number of bits for History register to be 3 since the above combination gives the minimum miss rate of 12.09. But since the size of index table increases the time complexity as well, so the second best design could be considered as : 11(gshare index bits.) and 2 (bits of history register)

    **Jpeg_trace.txt :** For the current trace file, the best suited design will be with the gshare index bits to be 12 and the number of bits for History register to be 6 since the above combination gives the minimum miss rate of 7.19.

    **Perl_trace.txt** : For the current trace file, the best suited design will be with the gshare index bits to be 12 and the number of bits for History register to be 3 since the above combination gives the minimum miss rate of 7.83.

- **Hybrid Branch Predictor :**

Hybrid branch predictor is a combination of both bimodal and gshare branch predictor. And as per the simulation results obtained, the minimum miss prediction rate that could be obtained was using the configuration as : Hybrid Index size : 8 bits, Gshare Index size : 9 , Bimodal Index size: 9 and the History Register size as 4 bits.

- ## **Conclusion**

The main motive of understanding the concepts of Branch Predictor and finding out the most effective and time saving and cost saving branch predictor was successfully achieved. Also, how the various performance metrics like Index table size, History Register Size affects the performance parameters are now well understood. The experiments and the associated plots show that the `m' value of a predictor plays an important role in bringing down the misprediction rate. Other factors include `n' value and branch target buffer size.