

## CS 204 LAB ASSIGNMENT: 6

### Understanding of Branch Prediction Accuracy

Member1: Manik Gupta (2021MCB1187)

Member2: Anany Dhamija (2021MCB1231)

#### Observations:

##### For Bubble\_Test\_Lab

Parameter	Always Taken	Always Not Taken	1 Bit Predictor	2 Bit Predictor
Miss	60892	67181	8392	3525
Hit	67181	60892	119681	124548
Accuracy	52.45%	47.54%	93.44%	97.24%

##### For Fac\_Test\_Lab

Parameter	Always Taken	Always Not Taken	1 Bit Predictor	2 Bit Predictor
Miss	27379	48151	13425	7444
Hit	48151	27379	62105	68086
Accuracy	63.75%	36.24%	82.22%	90.14%

##### For Qsort\_Test\_Lab

Parameter	Always Taken	Always Not Taken	1 Bit Predictor	2 Bit Predictor
Miss	26381	56657	10971	4447
Hit	56657	26381	72067	78591
Accuracy	68.23%	31.77%	86.78%	94.64%

##### For Sqrt\_Test\_Lab

Parameter	Always Taken	Always Not Taken	1 Bit Predictor	2 Bit Predictor
-----------	--------------	------------------	-----------------	-----------------

Miss	44467	71676	12660	4811
Hit	71676	44467	103483	111332
Accuracy	61.73%	38.28%	89.09%	95.85%

### **For recursion\_Test\_Lab**

Parameter	Always Taken	Always Not Taken	1 Bit Predictor	2 Bit Predictor
Miss	23347	60131	7835	2822
Hit	60131	23347	75643	80656
Accuracy	72.03%	27.96%	90.61%	96.61%

### **Always Taken Case:**

The branch predictor always assumes that a conditional branch will be taken. This means that whenever the processor encounters a branch instruction, it predicts that the branch will be taken, regardless of the actual outcome. While this strategy eliminates the need for making a prediction.

### **Always Not Taken Case:**

The branch predictor always assumes that a conditional branch will not be taken.

### **1 Bit Dynamic Branch Predictor:**

This predictor uses a single bit to remember the outcome of the last branch instruction. Initially, it may start with a prediction of "not taken" or "taken" and then update its prediction based on the actual outcome of the branch. If the prediction was correct, the predictor continues with the same prediction. If the prediction was incorrect, it updates the prediction to match the actual outcome. In our code we initially predicted that branch is “not taken”.

### **2 Bit Dynamic Branch Predictor:**

This predictor extends the 1-bit predictor by using two bits to remember the history of the last few outcomes of the branch. It can have four states: strongly taken, weakly taken, weakly not taken, and strongly not taken. Initially, it starts with a

prediction based on the history of the last few outcomes and then updates its prediction similar to the 1-bit predictor. The advantage of the 2-bit predictor is that it can provide more accurate predictions and adapt to changing patterns in the branch behavior.

### **Our Observations:**

We observed for all the five test cases we run our code on, that prediction accuracy for always taken case is inversely proportional to the always not taken case. Also accuracy for always taken case lies in the range of 50-70 %. Now for 1 bit branch predictor accuracy comes to about 90 %. And finally the 2 bit branch predictor gives the maximum accuracy of about 95 %.

### **Important Points:**

- We run our code till the end of file or till the first million instructions.
- We have done branch prediction for conditional branch statements only as unconditional branch instructions will always be taken.
- We have maintained a Branch Buffer Table containing Current PC and Target Address (can see for some PC)

BRANCH TARGET BUFFER	
CURRENT PC	TARGET ADDRESS
0x00010078	0x00010088
0x000100b0	0x000100c0
0x000100dc	0x00010120
0x000100f4	0x00010108
0x00010128	0x00010140
0x00010160	0x0001016c
0x00010208	0x00010210
0x00010244	0x00010264

- We have also maintained a Branch History Table containing PC, Predicted state and Actual state (can see for a PC)

#### BRANCH HISTORY TABLE

BHT for PC 0x00010078

Predicted : T

Actual : T

BHT for PC 0x000100b0

Predicted : T

Actual : NT

BHT for PC 0x000100dc

Predicted : T

Actual : NT

BHT for PC 0x000100f4

Predicted : T

Actual : T

BHT for PC 0x00010128

Predicted : T

Actual : T

BHT for PC 0x00010160

Predicted : T | T | T | T | T | T | T | T | T | T | T

Actual : T | T | T | T | T | T | T | T | T | T | NT