

16.482 / 16.561: Computer Architecture and Design

Dynamic Branch Prediction Practice Problems

1. Branch history tables: Say you are executing a program that contains the following high-level code snippet:

```
A[8] = {3, 7, 4, 9, 2, 1, 8, 4};
for (i = 0; i < 8; i++) {
    if (A[i] < 5)      { <fall-through code>  }
    else               { <branch taken code>  }
}
```

When compiled, this code contains two branches, as shown below. The BNE is part of the `if` statement above—if the condition is true, the branch is not taken; if the condition is false, the branch is taken. The BEQ controls the end of the loop.

<u>Address</u>			
<u>Decimal</u>	<u>Hex</u>		
20	0x14	loop	...
			...
40	0x28	BNE R4, R0, else	
			...
52	0x34	BEQ R7, R8, loop	

Your processor contains an eight-entry, 2-bit branch history table; its state when the processor reaches this code is as follows:

<u>Entry #</u>	<u>Value</u>
0	10
1	11
2	01
3	00
4	01
5	00
6	11
7	10

Determine the overall misprediction rate of the branch predictor for this code.

2. Correlating branch predictors: Now assume you have a 4-line, (2,2) correlating branch predictor, with all entries initially set to 11. Assume the initial global history is 11. Determine the overall accuracy of this predictor using the same code as in Problem 1.