

## 9 Branch Prediction [45 points]

You are given the following piece of code that iterates through two large arrays, *j* and *k*, each populated with completely (i.e., truly) random positive integers. The code has five branches (labeled *B1*, *B2*, *B3*, *B4*, and *B5*). When we say that a branch is *taken*, we mean that the code inside the curly brackets is executed. Assume that the code is run to completion without any errors or interruptions (i.e., there are no exceptions). For the following questions, assume that this is the only block of code that will ever be run on the machines, and that the loop condition branch is resolved first in the iteration (i.e., the if statements execute only *after* resolving the loop condition branch).

---

```

1  for (int i = 0; i < 1000; i++) {    //B1
2                                     //TAKEN PATH for B1
3      if (i % 2 == 0) {                //B2
4          j[i] = k[i] * i;            //TAKEN PATH for B2
5      }
6      if (i < 250) {                  //B3
7          j[i] = k[i] - i;            //TAKEN PATH for B3
8      }
9      if (i < 500) {                  //B4
10         j[i] = k[i] + i;             //TAKEN PATH for B4
11     }
12     if (i >= 500) {                 //B5
13         j[i] = k[i] / i;             //TAKEN PATH for B5
14     }
15 }
```

---

Listing 1: Application to evaluate.

You are given three machines whose components are identical in every way, except for their branch predictors.

- Machine A uses an always-taken branch predictor.
- Machine B uses one single-level global two-bit saturating counter branch predictor *shared by all branches*, which starts at Weakly Taken (2'b10).
- Machine C uses a *per-branch* two-bit saturating counter as its branch predictor. All counters start at Weakly Not Taken (2'b01).

The saturating counter values are as follows:

- 2'b00 - Strongly Not Taken
- 2'b01 - Weakly Not Taken
- 2'b10 - Weakly Taken
- 2'b11 - Strongly Taken

Answer the following questions:

1. [15 points] What is the branch misprediction rate when the above piece of code runs on Machine A? Show your work.

$$45.01\% = \frac{2251}{5001}.$$

**Explanation:**

B1 will generate 1 misprediction out of 1001 iterations (B1 is not taken in the 1001th iteration and the loop body does not execute). B2 will generate 500 mispredictions out of 1000 iterations, B3 will generate 750 mispredictions out of 1000 iterations, and both B4 and B5 will generate 500 mispredictions out of 1000 iterations.

2. [15 points] What is the branch misprediction rate when the above piece of code runs on Machine B? Show your work.

$$59.97\% = \frac{2999}{5001}.$$

**Explanation:**

From (0-249): 375 mispredictions (125 for B2 and 250 for B5) for 1250 branches.  
From (250-499): 874 mispredictions (2 for iteration 250, 4 for every odd iteration, 3 for every even iteration except for iteration 250) for 1250 branches.  
From (500-1000): 1750 mispredictions (3 for odd iterations, 4 for even iterations, 0 for  $i = 1000$ ) for 2501 branches.

3. [15 points] What is the branch misprediction rate when the above piece of code runs on Machine C? Show your work.

$$20.20\% = \frac{1010}{5001}.$$

**Explanation:**

You can split this up by branch.

**B1:** mispredicts at  $i = 0$ , and  $i = 1000$  (2 mispredictions out of 1001).

**B2:** mispredicts every time since it oscillates between Weakly Not Taken and Weakly Taken (1000 mispredictions out of 1000).

**B3:** mispredicts at  $i = 0$ ,  $i = 250$ , and  $i = 251$  (3 mispredictions out of 1000).

**B4:** mispredicts at  $i = 0$ ,  $i = 500$ , and  $i = 501$  (3 mispredictions out of 1000).

**B5:** mispredicts at  $i = 500$ , and  $i = 501$  (2 mispredictions out of 1000).