CMPE 200 HW#3

Due: Monday, Oct. 21, 11:59pm
Total Score: /100

Instructor: Hyeran Jeon

Computer Engineering Department, San Jose State University

Branch Predictor (20 + 20 + 60 pts)

1. Consider the following code.
   a. Fill the following timing diagram of the code below by assuming that we have full forwarding paths (both EX/MEM to EXE and MEM/WB to EXE as well as forwarding path needed for the early branch determination) and we use early branch determination. The branch in I6 predicted not taken but actually taken always.

```
L1:  lw   $t4, 0($t1)          // I1
     lw   $t5, 4($t1)          // I2
     addi $t1, $t1, 4          // I3
     sll  $t1, $t1, 2          // I4
     add  $t2, $t4, $t5        // I5
     beq  $t2, $zero, L1       // I6
     sw   $t4, 0($t1)          // I7
     sw   $t5, -4($t1)         // I8
```

	CC1	CC2	CC3	CC4	CC5	CC6	CC7	CC8	CC9	CC10	CC11	CC12	CC13
I1													

b. We learned that branch delay slot helps reduce the stall cycles. In the above code, can you reduce the stall cycle by using branch delay slot? If so, explain how we can revise the code to use branch delay slots. Otherwise, explain why we cannot get advantage of branch delay slot in this code

2. Consider a branch instruction that has following outcomes:

NT, T, T, NT, NT, T, T, NT

a. What is the misprediction rate of always-taken and always-not-taken predictors for this sequence of branch outcomes?

b. What is the misprediction rate of one-bit branch predictor for this sequence of branch outcomes? The initial predictor state is 0.

3. Consider the following loop body:

```
if (x is even) then                  // (branch b1)
     run some instructions           // (b1 untaken)
if (x is multiple of 4) then         // (branch b2)
     run some instructions           // (b2 untaken)
```

Assume that the above four statements are executed over nine loop iterations. During the nine iterations, the following nine values are used as x:

2, 5, 6, 11, 13, 18, 5, 6, 11

a. Assume that both branches use their own private 2-bit predictor to predict the outcome. Show the predicted and actual branch directions of branch b1 and b2 in each iteration in the following table. Show the values of the predictors of the branches in the third and fourth rows. And then, show the predicted direction and actual branch outcome in the last four rows. Use NT for untaken and T for taken direction. For example, in the iteration 1, b1's predictor value is 00 and hence b1's predicted direction is NT. Based on the branch condition, you know that b1 will be actually NT as 2 is an even number. As a result, you fill 00 to the b1's predictor value in iteration 2, NT to the "Predicted" field and NT to "Actual" field. Assume that both predictors' initial states are 00 (i.e. untaken). What are the misprediction rates for b1 and b2?

Iteration		1	2	3	4	5	6	7	8	9
X		2	5	6	11	13	18	5	6	11
b1's predictor		00	00							
b2's predictor		00								
b1	Predicted	NT								
	Actual	NT								
b2	Predicted	NT								
	Actual									

b. Assume now that a two-level branch prediction scheme is used. There is a 2-bit global predictor and a 1-bit global history register as descripted as GAg in the class. For each iteration of the loop, show the value of the global history register, the predicted and the actual branch directions of both b1 and b2. For the global history, fill the initial state of each iteration and the updated state after executing b1. For example, if the iteration N began with 0 in the global history register and the global history register value is updated to 1 after the b1's execution, fill 0 to the "Initial" and 1 to "After b1". Note that in each iteration,

the initial state of global history register is the state updated by b2's execution in the previous iteration. The initial state of the predictor table and global history register is all zeros. Use the provided predictor buffers to keep track of each predictor buffer's value if needed; the value filled in the predictor buffers will be used for giving you partial credit. What are the misprediction rate of b1 and b2?

iteration		1	2	3	4	5	6	7	8	9
X		2	5	6	11	13	18	5	6	11
global history	Initial	0								
	After b1	0								
b1	Predicted	NT								
	Actual	NT								
b2	Predicted									
	Actual									

Predictor buffer 0	00 →00→
Predictor buffer 1	00 →