ENGN 2910A Homework 05 (60 points) – Due Date: Oct 17th 2013

Professor: Sherief Reda

School of Engineering, Brown University

---

1. [Shen & Lipasti – 25 points] The code below steps through the elements of two arrays (A[] and B[]) concurrently, and for each element, it puts the larger of the two values into the corresponding element of a third array (C[]). The three arrays are of length N. NOTE: r0 is hardwired to 0.

```
     main:
1.          addi  r2, r0, A      %add immediate
2.          addi  r3, r0, B
3.          addi  r4, r0, C
4.          addi  r5, r0, N
5.          add   r10, r0, r0    %add register
6.          bge   r10, r5, end   %branch greater than or equal
     loop:
7.          lw    r20, 0(r2)
8.          lw    r21, 0(r3)
9.          bge   r20, r21, T1
10.         sw    r21, 0(r4)     % store word
11.         b     T2             % branch to T2
     T1:
12.         sw    r20, 0(r4)
     T2:
13          addi  r10, r10, 1
14          addi  r2, r2, 4
15          addi  r3, r3, 4
16          addi  r4, r4, 4
17          blt   r10, r5, loop  % branch less than
     end:
```

a. (5 points) Identify the basic blocks of this benchmark code by listing the static instructions belonging to each basic block in the following table. Number the basic blocks based on the lexical ordering of the code. Note: There may be more boxes than there are basic blocks.

b. (4 points) Draw the control flow graph for this benchmark.

c. (4 points) Now generate the instruction execution trace (i.e., the sequence of basic blocks executed). Use the following arrays as input to the program, and trace the code execution by recording the number of each basic block that is executed. N = 5; A[] = {8, 3, 2, 5, 9}; B[] = {4, 9, 8, 5, 1}.

d. (12 points) Fill in the following two tables based on the data you generated above.

Instr. Class	Static Number %	Dynamic Number %
ALU		
Load/Store		
Branch		

Table 1: Instruction mix.

	Static	Dynamic
Average basic block size (no. of instr.)		
Number of taken branches		
Number of not-taken branches		

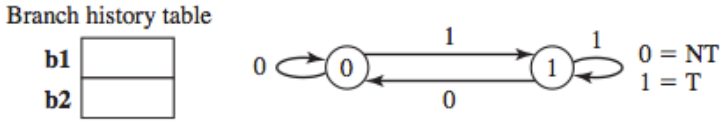
Table 2: Basic block/branch data.

2. [Shen & Lipasti – 35 points] Consider the following code segment within a loop body.

```
if (x is even) then increment a
if (x is a multiple of 10) then increment b
```

Assume that the following list of nine values of x is to be processed by nine iterations of this loop: 8, 9, 10, 11, 12, 20, 29, 30, 31. Note: assume that predictor entries are updated by each dynamic branch before the next dynamic branch accesses the predictor (i.e., there is no update delay).

a. [10 points] Assume that an one-bit (history bit) state machine (see below) is used as the prediction algorithm for predicting the execution of the two branches in this loop. Indicate the predicted and actual branch directions of the b1 and b2 branch instructions for each iteration of this loop. Assume ini- tial state of 0, i.e., NT, for the predictor.

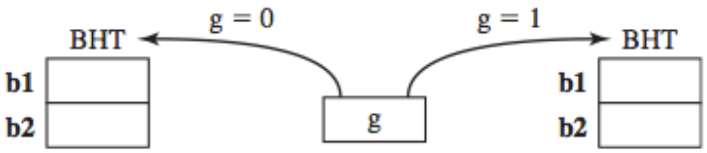


	8	9	10	11	12	20	29	30	31
b1 predicted									
b1 actual									
b2 predicted									
b2 actual									

b. [2 points] What are the prediction accuracies for b1 and b2?

c. [2 points] What is the overall prediction accuracy?

d. [15 points] Assume a two-level branch prediction scheme is used. In addition to the one-bit predictor, a one bit global register (g) is used. Register g stores the direction of the last branch executed (which may not be the same branch as the branch currently being predicted) and is used to index into two separate one-bit branch history tables (BHTs) as shown below.



Depending on the value of g, one of the two BHTs is selected and used to do the normal one-bit prediction. Again, fill in the predicted and actual branch directions of b1 and b2 for nine iterations of the loop. Assume the initial value of g = 0, i.e., NT. For each prediction, depending on the current value of g, only one of the two BHTs is accessed and updated. Hence, some of the entries below should be empty. Note: assume that predictor entries are updated by each dynamic branch before the next dynamic branch accesses the predictor (i.e. there is no update delay).

	8	9	10	11	12	20	29	30	31
For g=0: b1 predicted									
For g=0: b1 actual									
For g=0: b2 predicted									
For g=0: b2 actual									
For g=1: b1 predicted									
For g=1: b1 actual									
For g=1: b2 predicted									
For g=1: b2 actual									

e. [2 points] What are the prediction accuracies for b1and b2?

f. [2 points] What is the overall prediction accuracy?

g. [2 points] What is the prediction accuracy of b2 when g=0? Explain why.