CSCI 4717 – Computer Architecture, Fall 2006
Pipeline Homework
Due: Tuesday, November 21, 2005

1.) Assume that for a set of instructions, there is a 12% chance that any particular instruction is a conditional branch instruction. Of that 12%, one fourth of them result in a branch to a nonconsecutive address, i.e., the pipeline will have to be flushed. Ignoring any branch prediction algorithm, and using τ to represent the time it takes to execute a single stage of the pipeline, predict how long it will take to execute 1200 instructions. Assume the pipeline has 5-stages.

2.) Consider the following section of code.

```
for (i=0; i<10; i++)
{
     for (j=0; j<5; j++)
     {
          <code containing no conditional jumps>
     }
}
```

a.) Once compiled, how many conditional jumps would be contained in the above section of code? (static occurrence)
b.) After fully executing the above section of code, how many conditional jumps would the CPU have encountered? (dynamic occurrence)
c.) Using the static branch prediction algorithm “branch always,” how many of the conditional jumps calculated in the previous problem would have been predicted incorrectly?
d.) Using the branch prediction algorithm described in figures 12.16 and 12.17, how many of the conditional jumps calculated in part b would have been predicted correctly assuming an initial state of "predict taken"?

3.) Modify the following piece of code in order to support delayed branching and delayed loading. Assume a load from memory will force a subsequent instruction to stall in the pipeline if it uses the same register.

```
    MOV  CL,0           ;Initialize counter CL
L1: MOV  AL,[BX:1000]   ;Retrieve data from address 1000
    ADD  AL,23          ;Add immediate value
    MOV  [BX:1000],AL   ;Store result back to address 1000
    INC  CL             ;Increment CL
    CMP  CL, 100        ;Check to see if CL equals 100
    JNE  L1             ;If not, continued looping
    MOV  AL,1           ;Storing 1 at address 1001...
    MOV  [BX:1001],AL   ;...indicates we're done
```