CSCI 401 Test 2

E. Gomez

All test questions will be on the following computer:

- 32 bit words, byte addressed
- 3 GHz, 5 stage pipeline, no out of order execution. There is a 2 bit branch predictor that is right 90% of the time. If correct and the branch is not followed then no penalty happens. If correct and the branch is followed then the predictor costs 1 cycle, if incorrect it costs 3. Assume the branch is followed (taken) 7 of every 9 times. Assume Instruction buffers that negate instruction cache misses.
- 128Kb L1 Data cache, 0.5ns access, 4 word blocks, 4-way associative, write-back, no write-allocate, 1 in 60 mr.
- 1Gb RAM, 70ns to transfers a block.

1. For this computer calculate the following:
   (a) What is the write penalty in cycles?
   (b) What is the read penalty in cycles?
   (c) What is the branch penalty in cycles?
   (d) Assume 25% loads, 10% stores, 20% branches, and everything else is ideal. What is the CPI?
   (e) Describe how tag, set location, offset in block are found in an address. Do we need a dirty bit?

2. For the computer described above, consider the following code, which does affine encryption. You may annote the code itself if you do so neatly (i.e. I have to be able to tell what you did).

```
affine: ld.b R2,0(R3)   ! load the byte at mem[R3] and place it in R2
        add  R3,R3,1    ! R3 = R3 + 1
        mul  R2,R2,R4   ! R2 = R2 * R4
        add  R2,R2,R5   ! R2 = R2 + R5
        mod  R2,R2,R6   ! R2 = R2 % R6
        st.b R2,0(R7)   ! store low order byte in R2 to mem[R7]
        add  R7,R7,1    ! R7 = R7 + 1
        sub  R2, R3, R8 ! R2 = R3 - R8
        bnez R2, affine ! if R2 != 0 then goto affine
```

   (a) Note any WAR, WAW, and RAW hazards in the code.
   (b) Which data hazards matter in MIPS and how are they handled?
   (c) Which data hazards in this cocde are resolved thorugh forwarding?