CSCI 401 TEST 2 PRACTICE

K.E. SCHUBERT

(1) Given a non-pipelined architecture running at 2.5GHz, that takes 5 cycle to finish an instruction. You want to make it pipelined with 5 stages. The increase in hardware forces you to run the machine at 2GHz. The only stalls are caused by

- memory - (30% of the total instructions) a stall of 50 cycles happens in 2% of the memory instructions
- branch - (20% of the total instructions) a stall of 2 cycles happens in 20% of the branch instructions

What is the speedup? Clearly show all work.

$$\begin{aligned}
S &= \frac{5}{1 + .3 \times .02 \times 50 + .2 \times .2 \times 2}\frac{2}{2.5} \\
&= \frac{5}{1 + .3 + .08}\frac{4}{5} \\
&= \frac{4}{1.38}
\end{aligned}$$

(2) For the following code indicate where and label the type of the hazard (RAW, WAW, WAR).

```
square: ld.b R2,0(R3)   ! load the byte at mem[R3] and place it in R2
        add  R3,R3,4    ! R3 = R3 + 4, Next Word
        mul  R2,R2,R2   ! R2 = R2 * R2, square
        st.b R2,0(R3)   ! store low order byte in R2 to mem[R3]
        sub  R2, R3, R4 ! R2 = R3 - R8
        bnez R2, square ! if R2 != 0 then goto square
```

```
square: ld.b R2,0(R3)   ! 1 -
        add  R3,R3,4    ! 2 - WAR with 1
        mul  R2,R2,R2   ! 3 - RAW with 1
        st.b R2,0(R3)   ! 4 - RAW with 2,3
        sub  R2, R3, R4 ! 5 - RAW with 2, WAR with 4
        bnez R2, square ! 6 - RAW with 5
```

(3) This question will be about the following computer:

- 1.5 GHz, No Pipeline, instruction buffers that negate instruction load penalties 80% of the time
- 32Kb L1 cache, 1ns access, 4-way associative, write-through, not write-allocate, 2% miss rate for Data and a miss rate of 1/60 for Instructions
- 1Mb L2 cache, 12ns access, 4-way associative, write-back, write-allocate, 30% miss rate, 20% dirty
- 1Gb RAM, 50ns access

(a) Does it make sense to have a branch delay slot, since there is no pipeline? Why or why not?

No, a branch delay slot is only useful to reduce the penalty of address jumps in a pipelined machine. In a pipelined machine, when the branch occurs, two or three instructions following the branch are typically already loaded. These instructions must be flushed, reducing the efficiency of the pipeline. The branch delay slot reduces the penalty by one.

(b) What would the ideal speedup be for this computer if a 5 stage pipeline were installed?
Ideal speedup equals the number of stages in the pipeline thus 5.

(c) What is the effective access time for loading instructions?

$$\begin{aligned} EAT_{loads} &= MR_{Buffers}(T_{L1} + MR_{L1}(T_{L2} + MR_{L2}(1 + Dirty)T_{RAM})) \\ &= .2(1 + 1/60(12 + .3(1 + .2)50)) \\ &= .3ns \end{aligned}$$

(d) What is the CPI penalty to each instruction due to loading instructions?

$$\begin{aligned} Penalty_{CPI} &= EAT_{loads} \times Clock \\ &= .3ns \times 1.5GHz \\ &= .45cycles \end{aligned}$$

(e) What would you recommend changing to make this faster?
Two-thirds of the penalty is due to the access time of level one cache. I would look at trying to reduce or negate this first.