

CSE 775: Computer Systems Architecture
Autumn 2010
Homework #3

Instructor: Panda

Due: Monday, 15th November

1. (20+10=30 points) Consider the pipelined execution sequence of the following MIPS instructions. Assume 2 stages for floating-point addition, 5 stages for floating point multiplication, and 9 stages for floating-point division. Assume all functional units are pipelined. Assume interrupt for a floating-point operation occurs during the last stage of the EX step of that operation. Assume no interrupt occurring in the IF and ID steps.

LD	F4, 0(R10)
LD	F7, 0(R3)
MULTD	F6, F4, F7
ADDD	R4, R4, R7
LD	F5, 0(R4)
MULTD	F8, F4, F5
ADDD	F6, F6, F8
LD	F8, 0(R3)
DIVD	F7, F7, F8
ADDD	F6, F6, F7
SD	0(R3), F6

- (a) For each instruction, indicate all possible stages where *within* interrupts could occur. Identify possible causes behind these interrupts.
- (b) Show the pipelined sequence of the execution of the instructions and determine the percentage of clock cycles in which there is a provability of two or more interrupts occurring concurrently.
2. (35 points) Suppose the branch frequencies (as percentage of all instructions) are as follows: 1) Conditional branches - 20%, 2) Jumps and Calls - 5%, and 3) 60% of conditional branches are taken.

We are examining a four-deep pipeline (assume F, D, X and W) where the branch is resolved at the end of the second cycles for unconditional branches and at the end of the third cycle for conditional branches. Assuming that only the first pipe stage can always be done independent of whether the branch goes and ignoring other pipeline stalls, how much faster would the machine be without any branch hazards?

(Hints: Compute the number of stall cycles for different branch instructions (unconditional, conditional taken, conditional not-taken) on the new pipelined organization.)

3. (35 points) Assume the pipeline latencies as indicated in Figure 2.2 of the textbook (page 75). Unroll the following loop as many times as necessary to schedule it without any delays, collapsing the loop overhead instructions. Assume a one-cycle delayed branch. Show the schedule. FYI, the loop computes $Y[i]=a \times X[i] + Y[i]$, the key step in Gaussian elimination.

```
Loop: LD      F0, 0(R1)
      MULTD   F0, F0, F2
      LD      F4, 0(R2)
      ADDD    F0, F0, F4
      SD      0(R2), F0
      SUBI    R1, R1, #8
      SUBI    R2, R2, #8
      BNEQZ   R1, Loop
```