CS 6143 COMPUTER ARCHITECTURE II FALL 2011

EXAM I

150 MINUTES

WORK ALL PROBLEMS ***OPEN BOOK***

(25 pts)

1) Consider the following piece of **old** MIPS code for the unpipelined MIPS processor (*machine model number 0*) :

```
loop :  L.D     F1, 0(R1)        ; Load from vector A
        L.D     F2, 2000(R1)     ; Load from vector B
        ADD.D   F3, F1, F2
        MUL.D   F3, F3, F0       ; F0 is already initialized
        S.D     F3, 4000(R1)     ; Store to vector C
        DIV.D   F1, F1, F0
        S.D     F1, 6000(R1)     ; Store to vector D
        DADDI   R1, R1, #8
        DADDI   R2, R2, #(-1)10
        BNEZ    R2, loop
```

Assume that this is *machine model number 2* (the MIPS Int+FP pipeline).

The functional unit timings are as follows : FP operations ADD, SUB, MUL and DIV take 4, 4, 7 and 10 clock periods in EX, respectively.

The L1 cache memories take **one** clock period each and there are **no** cache misses.

Assume also that R2 is **two** (**2**) initially. In which clock period, will the execution of the code be completed ?

Clearly show in which clock period the execution ends as done in class, i.e. together with all the **necessary** forwarding and write-in-the-first-half-read-in-the-second-half cases.

Make sure you order the instructions so that the new code can run on this model correctly and with a minimum number of stall cycles. But, do **not** unroll the code !

(25 pts)
2) Consider the old **original** MIPS code in Question 1 again. Assume that the MIPS is implemented as the **superscalar hardware-speculative** Tomasulo algorithm machine as discussed in class. That is, this is *machine model number 5*. **Four (4)** instructions can be committed per cycle.

Assume that the functional unit timings are as given in Question 1 above ; the number of FP reservation station buffers is as given in class. Assume again that R2 is **two** (**2**) initially.

The L1 cache memories take **one** clock period each and there are **no** cache misses.

In which clock period, will the execution of the code be completed ? That is, what is the last clock period in which the Commit stage of the last instruction is done ? Show all the **necessary** forwarding and write-in-the-first-half-read-in-the-second-half cases. But, do **not** show the flushed out instructions.

(30 pts)
3) Consider the old **original** MIPS code in Question 1 again. Assume that the MIPS is implemented as the **VLIW MIPS** processor : That is, this is *machine model number 6.* Assume that the functional unit timings are as given in Question 1 above.

Show how you would unroll and issue the instructions as done in class. Assume that there are as many iterations as needed for your unrolling.

(20 pts)
4) Consider the old **original** MIPS code in Question 1 again. Assume that the MIPS is implemented as the **VMIPS** processor : That is, this is *machine model number 7*.

a) ***Vectorize*** the old code in terms of VMIPS instructions as much as possible. Add comments to your code. Assume that the loop has **64** iterations and all the registers needed have been appropriately initialized. For example, the R3, R4 and R5 registers are initialized to (R1 + 2000), (R1 + 4000) and (R1 + 6000), respectively.

If you **cannot** vectorize a part of the **original** code in Question 1, indicate which part and why !

b) ***Show*** the execution timings of the vectorized code as discussed in class. Assume that there is **chaining**. Also assume that there are **two (2)** memory pipelines.