Name _____

Computer Architecture
EE 4720
Midterm Examination

Friday, 31 October 2008, 10:40–11:30 CDT

Problem 1	_____	(50 pts)
Problem 2	_____	(10 pts)
Problem 3	_____	(20 pts)
Problem 4	_____	(20 pts)
Exam Total	_____	(100 pts)

Alias _____

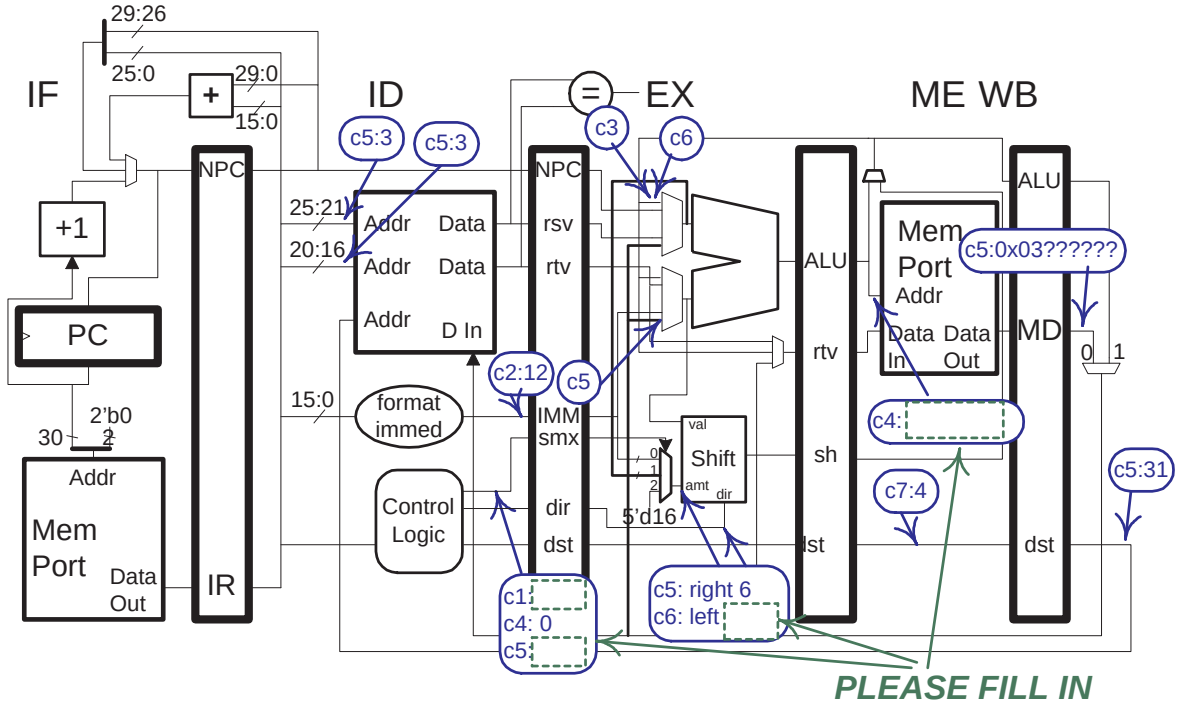
Good Luck!

Problem 1: In the MIPS implementation below some wires are labeled with cycle numbers and values that will then be present. For example, c5:3 indicates that at cycle 5 the wire will hold a 3. Other wires are labeled just with cycle numbers, indicating that the wire is used at that cycle. If a value on any labeled wire is changed the code would execute incorrectly. Instruction addresses and the first instruction have been provided. [50 pts]

(*a*) Finish a program consistent with these labels.

☐ All register numbers and immediate values can be determined.

☐ Be sure to fill the **three** blocks marked *PLEASE FILL IN*.



	Cycle: 0	1	2	3	4	5	6	7	8
`0x0b0000 lui r2, 0xb`	IF	ID	EX	ME	WB				
`0x0b0004`		IF	ID	EX	ME	WB			
`0x0b0008`			IF	ID	EX	ME	WB		
`0x0b000c`				IF	ID	EX	ME	WB	
`0xedd900`					IF	ID	EX	ME	WB

Problem 1, continued: Continue referring to the implementation on the previous page.

(*b*) Explain whether each instruction below could be the instruction at address `0xb0008` (on the previous page). If it could be the instruction write "Possible" otherwise write "Impossible because ..." (The grade will be based on the reason.)

☐ `add`

☐ `j`

☐ `jal`

☐ `beq`

Problem 2: [10 pts] Based on the experience of preparing SPECcpu benchmark runs manufacturers A and B each release a new compiler that improves their respective SPECcpu scores.

(a) In the course of preparing a SPECcpu benchmark run manufacturer A discovers a new optimization technique that improves the performance of most of the SPECcpu programs, and other programs. This optimization technique is added to the compiler for use at the `-O1` and higher optimization levels. The manufacturer sells the compiler, proudly boasting about the performance benefits.

☐ Is it in the spirit of the SPECcpu rules to use this new optimization technique for the base results? Explain.

(b) To prepare a SPECcpu benchmark run manufacturer B has its best programmers prepare hand-written assembly code for the most time consuming portion of each benchmark. The compiler will recognize each benchmark based on the source code and substitute the hand-written routines where needed; the hand-written code will only work for these benchmarks.

These optimizations are included in manufacturer B's compiler and used at `-O1` and higher levels. Manufacturer B sells this new compiler.

☐ Is it in the spirit of the SPECcpu rules to use this new optimization technique for the peak results? Explain.

☐ Explain why it is not in the spirit of SPECcpu rules to use such optimizations for base results.

☐ Assuming that the compiler can not be reverse engineered (there is no way to inspect the compiler itself) and assuming that manufacturer B can keep secrets, how might this cheating be discovered?

Problem 3: Answer the following ISA questions.

(*a*) [10 pts] A RISC advocate claims that by having fixed-length instructions and alignment restrictions a branch can reach twice as far for a given displacement field size than would be possible in CISC ISAs.

☐ Explain why.

A CISC advocate responds that branches in CISC programs would take less space anyway.

☐ Provide a reason for small-displacement branches.

☐ Provide a reason for large-displacement branches.

(*b*) [10 pts] Indicate whether each item below is usually an ISA feature or an implementation feature.

☐ Number of bits in immediate.

☐ Clock frequency.

☐ Number of branch delay slots (if any).

☐ Floating-point format.

☐ Minimum distance between load instruction and a dependent instruction to avoid a stall.

Problem 4: Equivalent MIPS and SPARC code fragments appear below.

(*a*) [10 pts] Taking advantage of SPARC's condition code features, modify the SPARC code to use one fewer instruction (without changing what it does).

```
# MIPS Branch Example
 addi $t1, $t1, -1
 bne  $t1, 0  LOOP
 add  $t2, $t2, $t3
...

! Equivalent SPARC Branch Example
 add l1, -1, l1      ! l1 = l1 - 1
 subcc l1, 0, g0     ! g0 = l1 - 0
 bne LOOP
 add l2, l3, l2
...
```

(*b*) [10 pts] For branch-in-ID implementations, why might it be possible to attain higher clock frequencies for condition-code ISAs, like SPARC, than for register-test branche ISAs, like MIPS.

☐ Condition code performance advantage.