

Name \_\_\_\_\_

Computer Architecture  
EE 4720  
Midterm Examination  
Friday, 27 March 2009, 11:40–12:30 CDT

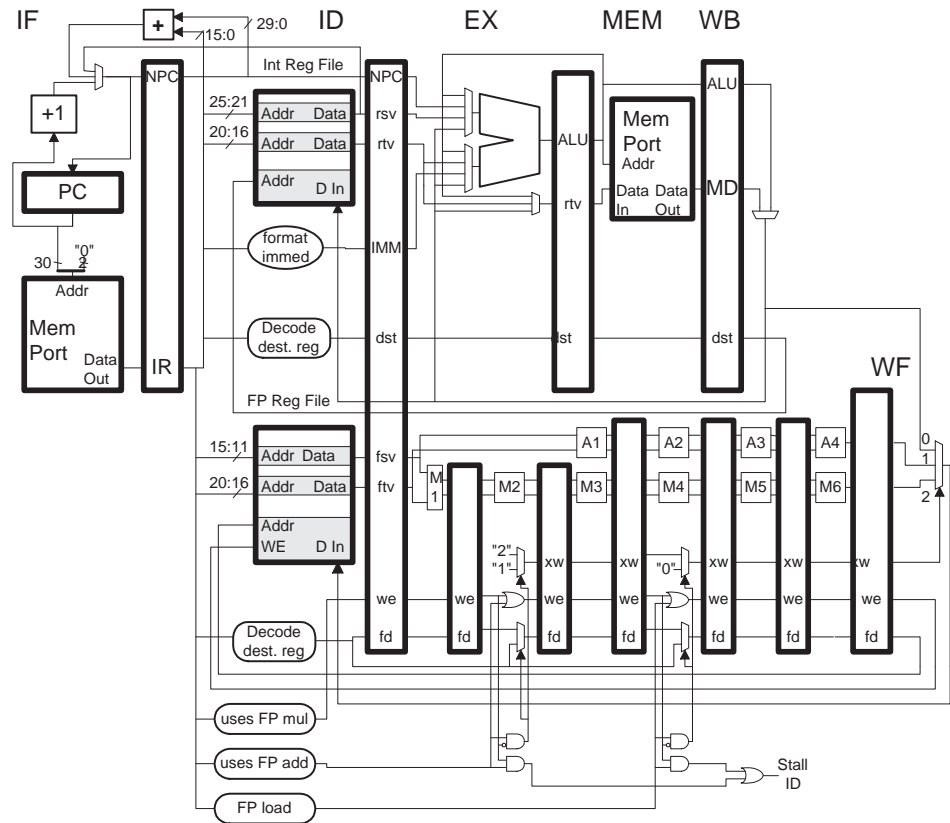
Problem 1 \_\_\_\_\_ (30 pts)  
Problem 2 \_\_\_\_\_ (30 pts)  
Problem 3 \_\_\_\_\_ (10 pts)  
Problem 4 \_\_\_\_\_ (20 pts)  
Problem 5 \_\_\_\_\_ (10 pts)

Alias \_\_\_\_\_

Exam Total \_\_\_\_\_ (100 pts)

*Good Luck!*

Problem 1: [30 pts] The code below executes on the illustrated FP pipeline.



```

LOOP:
ldc1 f0, 0(r1)

addi r1, r1, 8

mul.d f2, f0, f0

bneq r1, r2, LOOP

add.d f6, f6, f2
    
```

(a) Add reasonable bypass paths needed by the code above, for both the integer and FP pipelines.

Add reasonable bypass paths. Don't add unneeded bypass paths.

(b) Analyze the performance of the code using your bypasses:

Show a PED for the code above using your bypasses. (Use this page or next page.)

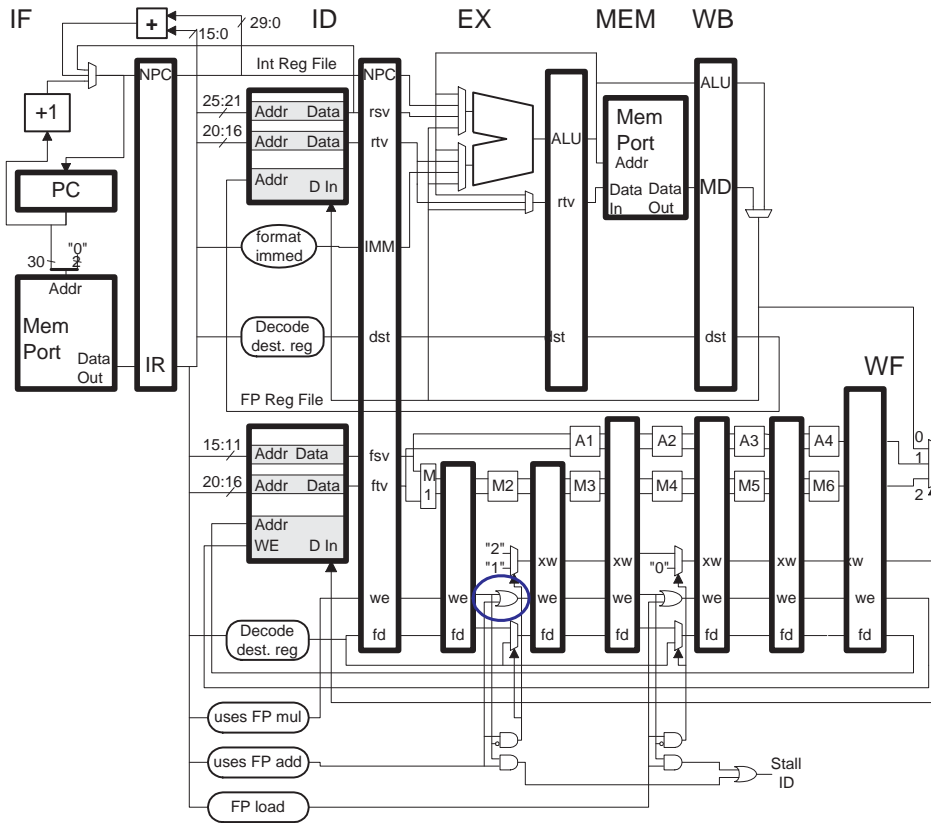
Compute the CPI of the code for a large number of iterations.

*Use this page for PED, if needed.*

```
LOOP:  
ldc1 f0, 0(r1)  
  
addi r1, r1, 8  
  
mul.d f2, f0, f0  
  
bneq r1, r2, LOOP  
  
add.d f6, f6, f2
```

Problem 1, continued:

(c) A component failure in the MIPS implementation below has changed the circled OR gate into an AND gate.



Is it still possible to perform a FP multiply? If yes, show how with PED, if no explain.

# Original Code. (For answer show modified code with PED.)  
mul.d f0, f2, f4

Is it still possible to perform a FP add? If yes, show how with PED, if no explain.

# Original Code. (For answer show modified code with PED.)  
add.d f0, f2, f4



Problem 2, continued:

(c) The four-stage MIPS-A implementation may be faster or slower than the five-stage MIPS-I.

- Provide a pair of equivalent code fragments, one for MIPS-I that runs on the five-stage implementation and one for MIPS-A that runs on the four-stage, in which the MIPS-A version is faster. *Hint: The MIPS-I code will have a familiar stall.*

- Provide a pair of equivalent code fragments, one for MIPS-I that runs on the five-stage implementation and one for MIPS-A that runs on the four-stage, in which the MIPS-A version is slower. *Hint: Think about the difference in the ISAs.*

(d) A company needs to decide whether to develop MIPS-I or MIPS-A. How does it decide? Assume that at this point software compatibility is not an issue.

- How should company decide which to develop?

- Will having skilled compiler writers tilt the decision towards MIPS-A or MIPS-I? Explain.

Problem 3: [10 pts] Answer the questions below.

(a) Why does MIPS have a `beq` but does not have a `blt` (branch less than), even though a `blt` instruction would be frequently used?

Why `beq` but no `blt`?

(b) Explain why a branch delay slot can be thought of as a short-sited feature in an ISA.

Delay slots are good when...

...but delay slots are bad when...

Problem 4: [20 pts] The doing-it-the-hard-way MIPS code below loads the constant  $\frac{1}{3}$  in IEEE 754 single-precision format, `0x3eaaaaab`, into register `f2`.

```
addi r10, r0, 1
addi r20, r0, 3
mtc1 f12, r10
mtc1 f22, r20
cvt.s.w f14, f12
cvt.s.w f24, f22
div.s f2, f14, f24
```

(a) Describe what the `mtc1` and `cvt.s.w` instructions do.

Explain `mtc1`

Explain `cvt.s.w`

(b) The code above uses more instructions than are necessary and also uses a wastefully time-consuming instruction.

What is the wastefully time-consuming instruction?

Re-write the code so it uses fewer instructions without using load instructions. *Hint: A correct answer uses three instructions and a piece of information slipped into the first sentence of the problem.*



Problem 5: [10 pts] Answer the following SPECcpu questions.

(a) In the SPECcpu2000 suite profiling was allowed for both base and peak results but in SPECcpu2006 profiling is allowed for peak results but not for base results.

- Why isn't profiling allowed for base results in SPECcpu2006? Your answer should say something about the difference between base and peak.

(b) Company *A* has a reputation for reliable compilers, company *B* has a reputation for buggy compilers. Both companies and their customers are okay with this. (Think Italian sports cars.)

Optimization *X* results in a substantial improvement in SPECcpu base scores. Company *B* has it in their shipping compilers (those sold to customers) but company *A* only has optimization *X* in their experimental compilers (not available to customers), but they are working hard on it. The optimization achieves the same results for company *A* and *B*. Company *B*'s compiler will not run on company *A*'s system (as though they'd want to!). *Grading Note: The last line was not in the original exam, but a student did see the possibility of using B's compiler for A's spec run.*

- Can Company *A* use optimization *X* to prepare SPECcpu2006? Explain.

- Can Company *B* use optimization *X* to prepare SPECcpu2006? Explain.

- Your answers above should reflect the letter of SPEC's rules. Do you think this achieves SPECcpu's goals or what you would like to see in benchmark results? Explain.