CS/ECE472 Midterm #1
SPRING 2011

NAME: ______________________
Student ID#: __________________

THIS IS NOT OPEN BOOK. YOU GET 1 page (front/back) of NOTES. You also get a
print-out of the Green MIPS pull-out sheet.

NO COMPUTERS, OR SOLVING PROBLEMS DIRECTLY USING
CALCULATORS.

Your signature is your promise that you have not cheated and will not cheat on this exam,
nor will you help others to cheat on this exam.

Signature: ________________________________

Question 1 _____________ (10 points)
Question 2 _____________ (10 points)
Question 3 _____________ (10 points)
Question 4 _____________ (10 points)
Question 5 _____________ (10 points)
Question 6 _____________ (20 points)
Question 7 _____________ (30 points)

TOTAL _____________
1) (10 points) Floating Point

Write the decimal number -13.375 in single precision IEEE 754 floating pt. I want to see all 32 bits. Please group your exponent bits together in groups of 4 and your significand in groups of 4.

2) (10 points) Multiplication

Multiply a times b using the 2's Complement multiplication algorithm, using the 1st hardware algorithm described in the notes (the one that was described in the lecture). You will get points for each intermediate step so show ALL your work. Circle the results of your shifts. (4 circled lines). Assume negative numbers, and that overflow is possible. (For example, show the product accumulation register after each cycle, the multiplicand and multiplier values, along with the corresponding algorithm).

a = 1001 (multiplicand), b = 1111 (multiplier)

Circle your 8-bit answer
3) (10 points) Instruction Decoding

Consider the instruction: addi $t7, $t3, 0xF001 through the implementation on page 7 of this test. After the instructions is decoded:
What are Instruction bits 25-21 __ __ __ __ __
What are Instruction bits 20-16 __ __ __ __ __ __
What is ALUSrc __ ( 0, 1, or X <- don't care )
What is the ALU operation to be performed? (i.e. AND, OR, add, sub, SLT, NOR)

4) (10 points) C-to-MIPS Translation

Read the MIPS code below and write the C function it is associated with. Write the function header as well as the C statements inside the function.

```c
foo:     
    srl $a0, $a0, 4
    srl $a1, $a1, 8
    add $v0, $a0, $a1
    jr $ra
```

5) (10 points) For each MIPS instruction, indicate the addressing mode and the instruction format that it uses.

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Addr. Mode</th>
<th>Instr. format</th>
</tr>
</thead>
<tbody>
<tr>
<td>j</td>
<td>Loop</td>
<td>1 = Register addressing</td>
</tr>
<tr>
<td>or</td>
<td>$t0, $t1, $t2</td>
<td>2 = Base or displacement addressing</td>
</tr>
<tr>
<td>beq</td>
<td>$t1, $t2, labl2</td>
<td>3 = Immediate addressing</td>
</tr>
<tr>
<td>sll</td>
<td>$t1, $t2, 2</td>
<td>4 = PC-relative addressing</td>
</tr>
<tr>
<td>jal</td>
<td>addThese</td>
<td>5 = Pseudodirect addressing</td>
</tr>
</tbody>
</table>
| addi        | $s0, $s1, 10 | R }
| lw          | $t0, 0($gp) | I } Instruction formats |
| slt         | $t0, $t1, $t2 | J |


Problem #6: Speedup (20 points)

6. In your favorite program which takes 10 seconds to run (assume single cycle):
   Swaps account for 20% of the clock cycles,
   memory access instructions take 70% of the cycles,
   floating point instructions take 10% of the cycles.

   You can either: improve memory access instructions by a factor of 8, or you can improve
   swaps by a factor of 4 and floating point instructions by a factor of 2.

   What is the execution time of each choice. Show your work.

Problem 7: Verilog (30 points) YOUR QUESTION, ROBERT!

6a.) Write a Verilog module that describes the following circuit in terms of its logic. Minor syntax errors will not be penalized. Keep in mind that this circuit takes 3 inputs and has 1 output. Name the module test1.

6b.) What common element does this circuit represent? (Hint: you have used one of these in lab)

6c.) How else can you describe the behavior of this circuit? Write a module that implements this circuit using a Verilog behavioral statement. Once again, minor syntax errors will not be penalized. Name this module test2.