CS/ECE472 Midterm #1
Spring 2010

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 0 ___________ (10 points)
Question 1 ___________ (10 points)
Question 2 ___________ (10 points)
Question 3 ___________ (10 points)
Question 4 ___________ (10 points)
Question 5 ___________ (15 points)
Question 6 ___________ (20 points)
Question 7 ___________ (15 points)

TOTAL _______________
3) (10 points)

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)

\[ a = 1001 \text{ (multiplicand) }, b = 1111 \text{ (multiplier)} \]

Circle your 8-bit answer

\[ \begin{array}{c}
2's \gg a = 0111 \\
b = 0001 \\
\text{Multiplier} \\
\text{add} \ 0001 \\
\text{Shift} \ 0000 \\
\text{Multiplicand} \\
1111 \\
1111 \\
111110 \\
\text{Product} \\
0000 \\
0000 \\
0000 \\
0000 \\
0000 \\
\text{Solution: } 0000 \\
\text{Overflow = 7, expect 7 so no overflow.} \\
\end{array} \]

4) (10 points)

Consider the instruction: addi $t3, t1, -17$ through the implementation on page 7 of this text. 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)
6) (20 points)
Assume the following latencies for logic blocks in the datapath:

<table>
<thead>
<tr>
<th></th>
<th>I-MEM</th>
<th>Add</th>
<th>Mux</th>
<th>ALU</th>
<th>Regs</th>
<th>D-Mem</th>
<th>Sign-Ext</th>
<th>Shift-left-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>200ps</td>
<td>200ps</td>
<td>100ps</td>
<td>300ps</td>
<td>200ps</td>
<td>200ps</td>
<td>100ps</td>
<td>100ps</td>
</tr>
<tr>
<td>b</td>
<td>300ps</td>
<td>50ps</td>
<td>50ps</td>
<td>100ps</td>
<td>100ps</td>
<td>800ps</td>
<td>50ps</td>
<td>50ps</td>
</tr>
</tbody>
</table>

Processor A: Built for fast memory; slow on compute
Processor B: Built for fast computation; slow on memory

a) What is the clock cycle time if the only type of instructions we need to support are ALU? (add, and, etc)
   
   \[
   \text{I-MEM} \rightarrow \text{Reg} \rightarrow \text{Mux} \rightarrow \text{ALU} \rightarrow \text{Mux} \rightarrow \text{Reg}
   \]
   
   \[A: 300\text{ps} + 100\text{ps} + 100\text{ps} + 300\text{ps} + 100\text{ps} + 100\text{ps} = 1100\text{ps}\]
   
   \[B: 300\text{ps} + 100\text{ps} + 50\text{ps} + 100\text{ps} + 800\text{ps} + 100\text{ps} = 1200\text{ps}\]

b) What is the clock cycle time if we only have to do lw?
   
   \[
   \text{I-mem} \rightarrow \text{Reg} \rightarrow \text{ALU} \rightarrow \text{D-MEM} \rightarrow \text{Mux} \rightarrow \text{Reg}
   \]
   
   \[A: 200\text{ps} + 200\text{ps} + 100\text{ps} + 100\text{ps} + 100\text{ps} = 700\text{ps}\]
   
   \[B: 200\text{ps} + 100\text{ps} + 100\text{ps} + 100\text{ps} + 100\text{ps} = 1400\text{ps}\]

c) What is the clock cycle time if we must support j instructions?
   
   \[
   \text{I-MEM} \rightarrow \text{Shift-left} \rightarrow \text{Mux}
   \]
   
   \[\text{(Add)}\]

d) For both processors (a, b), a new memory technology improves D-Mem by 10x.
   What is the speedup of processor b over a, with this new technology? What was the speedup of b over a, without this new technology?

   For lw, \((b)\) becomes 730
   \[ \text{Speed-up IS } \frac{1100}{730} \approx 1.4 \times \]
   \[ (a) \text{ becomes } 1020 \]

   Without, speedup IS \[ \frac{1200}{1480} = 0.83 \times \]
0) (10 points)

The quarter is exactly half-way over. What are the (2) things you would like to see improved in this class, to make it 'better' for you—from a “quality of learning” point of view.

1) (10 points)

Write the decimal number 11.125 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.

\[ 0 \quad \begin{array}{ccccc} 1 & 0 & 0 & 0 & 1 \end{array} \quad \begin{array}{ccccc} 0 & 1 & 1 & 1 & 0 \end{array} \]

2) (10 points)

Computer (2 GHz) has 3 instruction classes:  
- class A CPI=2  
- class B CPI=4  
- class C CPI=17  
The same program is compiled with 2 compilers.  
(Billion=10^9):

<table>
<thead>
<tr>
<th>Compiler</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compiler1</td>
<td>3</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>Compiler2</td>
<td>31</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

Determine the execution time for each version of the program.

- Compiler 1:  
  \[ \frac{233 \times 10^8 \text{ cycles}}{10^9 \text{ Hz}} = 233 \times 10^8 \text{ cycles} \]
  \[ \frac{116.5 \text{ seconds}}{10^8} = 0.001165 \text{ seconds} \]

- Compiler 2:  
  \[ \frac{66 \times 10^8 \text{ cycles}}{10^9 \text{ Hz}} = 66 \times 10^8 \text{ cycles} \]
  \[ \frac{33 \text{ seconds}}{10^8} = 0.00033 \text{ seconds} \]

Determine the MIPS figure for each version of the program.

\[ \text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^8} \]

- Compiler 1:  
  \[ \frac{233 \times 10^8}{116.5 \times 10^8} = \frac{233}{116.5} \approx 2.0 \]

- Compiler 2:  
  \[ \frac{37 \times 10^8}{33 \times 10^8} = \frac{37}{33} \approx 1.12 \]
5) (15 points)

A) Write a loop in MIPS that subtracts up all the ints stored in an array of size 5. I have started it for you. Use my registers ($t0, $s1, $s2) as declared in the comments. Use only MIPS core instructions. Feel free to use registers $t1 - $t7.

    # $t0 holds the address of the first int in the array
    addi $s1, $t0, 20     # first word not in the array
    addi $s2, $zero, 0     # sum initialized to zero

LP:   subi $s1, $s1, 4
      lw $t1, 0($s1)
      sub $s2, $s2, $t1
      bne $s1, $t0, LP

UNROLLED:  subi $s1, $s1, 4
            lw $t1, 0($s1)
            sub $s2, $s2, $t1
            subi $s1, $s1, 4
            lw $t1, 0($s1)
            sub $s2, $s2, $t1

B) Unroll your loop (just the code that starts at LP) so that there is no loops. Write your code to the right of your code from part A).

C) What is the difference in the # of instructions between (A) and (B)? What is the difference in the total execution time (assuming single-cycle processor)?

4 instructions in A, 15 in B

total execution time: 20 cycles in (A), 15 in (B)
7) (15 points) HINT: Since you might not have written this down:

\[ \text{Power} = \text{Capacitance} \times \text{Voltage}^2 \times \text{Clock\_Frequency} \] (p. 39 of book. You can ignore the \( \frac{1}{2} \) scaling coefficient for this class)

Suppose we have developed new versions of a microprocessor with the following characteristics:

<table>
<thead>
<tr>
<th>Version</th>
<th>Voltage</th>
<th>Clock Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor 1</td>
<td>5V</td>
<td>0.5GHz</td>
</tr>
<tr>
<td>Processor 2</td>
<td>3.3V</td>
<td>1 GHz</td>
</tr>
</tbody>
</table>

a) By how much has the capacitive load been reduced between versions if the dynamic power has been reduced by 10%?

\[
\frac{P_2}{P_1} = 0.9 \quad \Rightarrow \quad \frac{C_1 V_1^2 f_1}{C_2 V_2^2 f_2} = 0.9
\]

\[
\frac{C_1 V_1^2}{C_2 V_2^2} = 0.9 \quad \Rightarrow \quad \frac{C_1}{C_2} \cdot \frac{V_1^2}{V_2^2} \cdot \frac{f_2}{f_1} = 0.9
\]

Solve for \( C_2 \) in terms of \( C_1 \).

b) By how much has the dynamic power been reduced if the capacitive load does not change?

\[
\frac{P_2}{P_1} = \frac{C_1 V_1^2 f_1}{C_2 V_2^2 f_2}
\]

Solve for ratio \( \frac{f_2}{f_1} \).

c) Assuming that the capacitive load of version-2 is 80% the capacitive load of version-1, find the voltage for version-2 if the dynamic power of version-2 is reduced by 40% from version-1.

\[
C_2 = 0.8 C_1 \quad \Rightarrow \quad \frac{V_1^2 f_1}{V_2^2 f_2} = 0.6
\]

Using given \( V_1, f_1 \) and \( f_2 \), solve for \( V_2 \).