ECE472 Final Exam
FALL 2009

CS/ECE472 Midterm #2
Fall 2009

NAME: ____________________________
Student ID#: ______________________

OPEN BOOK, OPEN NOTES. OPEN COMPUTER IS OK. NOT SOLVING PROBLEMS
DIRECTLY USING CALCULATORS or GOOGLE, or SIMILAR ELECTRONIC COMPUTATION.

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: __________________________

Answer Sheet
Fall 2009

Question 1 ________ (10 points)
Question 2 ________ (10 points)
Question 3 ________ (5 points)
Question 4 ________ (10 points)
Question 5 ________ (5 points)
Question 6 ________ (5 points)
Question 7 ________ (15 points)
Question 8 ________ (10 points)
Question 9 ________ (10 points)
Question 10 ________ (20 points)

TOTAL ________
1a. IEEE Floating point number representation. Make sure your answer has the correct number of digits. Represent $-21.75$ in 32 bit IEEE single precision floating-point format. Show your answer in hexadecimal.

$$C1AE0000$$

1b. Multiply the integer $-29_{10}$ by $7_{10}$ using 2's complement. Use 32 bits. You may express your answer in binary, but it must have 32 bits.

2. Consider the following instruction through the implementation on page 9 of this test.

<table>
<thead>
<tr>
<th>Address of instr.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0040 0030</td>
<td>subi $18, $15, 1</td>
</tr>
</tbody>
</table>

After the instruction is decoded:

- What are Instruction bits 25-21
  - $0 1 1 1 1$

- What are Instruction bits 20-16
  - $1 0 0 1 0$

- What is ALUSrc
  - $0$ (0, 1, or X <= don't care)

- What is MemtoReg
  - $0$ (0, 1, or X <= don't care)

- What is RegDst
  - $0$ (0, 1, or X <= don't care)

- What is Branch
  - $0$ (0, 1, or X <= don't care)

Consider the following instruction through the implementation on page 10 of this test.

<table>
<thead>
<tr>
<th>Address of instr.</th>
<th>Instruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0040 0030</td>
<td>beq $18, $15, 9</td>
</tr>
</tbody>
</table>

- What is ALUSrcA during cycle 1
  - $0$ (0, 1, or X <= don't care)

- What is ALUSrcA during cycle 2
  - $x$ (0, 1, or X <= don't care)

- What is ALUSrcA during cycle 3
  - $0$ (0, 1, or X <= don't care)

E.C. Assuming the branch is taken, what address is loaded into the PC? $0x00400066$
3. Assume that you have a pipelined implementation of MIPS with a perfect cache. Draw the pipelining diagram for the code along the left side. I have started the diagram for you with the first statement. **Indicate all data dependencies by circling the registers in the code.** Indicate all necessary forwarding from the output of one functional unit to the input of another functional unit. Indicate all necessary stalls with a nop. **Keep everything that happens in the same clock cycle in a straight column.** You do not have to shade the active units.

```
sub $4, $2, $4
sw $4, 1($2)
lw $5, 100($4)
sub $3, $4, $5
```

4. The CPI of a machine with a perfect cache is 2.91 for a certain benchmark.

Now, assume a L1 instruction cache miss rate of 13% and a L1 data cache miss rate of 15%.
Assume the miss penalty (for either L1 caches) is 4 cycles. The hit time is 1 cycle (L1 cache).
Assume a L2 instruction cache miss rate of 3% and a L2 data cache miss rate of 5%.
Assume the miss penalty (for either L2 caches) is 400 cycles. The hit time is 10 cycles (L2 cache).

The frequency of data access instructions in this benchmark is 40%.

How much faster is the machine with the perfect cache than the one with the imperfect L1 and L2 caches? Otherwise the machines are identical and running the same code. Express your answer either as a fraction or as a decimal rounded to the hundredths place. (X.XX) Your answer should be > 1.
5. Assume we have a 8 KB cache direct mapped cache. (KB = $2^{10}$ bytes)
The block size is 16 words.
Given a 32-bit physical address, divide up all the bits and indicate what they are used for to find and access the requested word in the cache. Do NOT draw the cache entries, the mux, the AND gate, ...
Just indicate exactly how many bits and which bits of the address are used for each purpose (the tag, the index, ...). Label your groups of bits with their purpose.

6. Assume we have a 4 KB cache 4-way set associative cache. (KB = $2^{10}$ bytes)
The block size is 4 words.
Given a 32-bit physical address, divide up all the bits and indicate what they are used for to find and access the requested word in the cache. Do NOT draw the cache entries, the mux, the AND gate, ...
Just indicate exactly how many bits and which bits of the address are used for each purpose. Label your groups of bits with their purpose.
7. A) Write a loop in MIPS that first multiplies each int in an array size of 64 by the constant '-5', and then adds it to a running sum. Use my registers ($t0, $s1, $s2) as declared in the comments. Use only MIPS core instructions (and a new instruction 'mult'). Feel free to use registers $t1 - $t7.

    # $t0 holds the address of the first int in the array

LP:

B) Unroll your loop (just the code that starts at LP) so that it executes 16 times and has no data hazards. Write your code to the right of your code from part A).

C) Given the registers you see used in the starter code of A), what register(s) must be saved to the stack frame in order to follow MIPS calling conventions?
8. Assume a 46-bit virtual address and a 32-bit physical address. The page size is 12 KB. How many entries are there in the page table? Express your answer in powers of 2. Show your work for this problem. (KB = 2^10 bytes)

\[
\frac{2^{46}}{12 \cdot 2^{10}} = \frac{2^{36}}{12} = \frac{2^{36}}{3 \cdot 2^{10}}
\]

9. `beq $t0, $t1, 0` will branch forward 1 instruction if taken (because PC<--PC+4 already done).
What is the largest number of instructions we can branch forward with a `beq` instruction?

- Immediate is 16 bits.
- If `Immediate = 0`, branches 1 instruction.
- Largest instruction = all 1's.

\[
2^{16} - 1 = 32767
\]
\[
32767 + 1 \quad \text{(for effect == 0)}
\]
10. You are the CEO of a company building a real-time, continuous monitoring, heart-failure detector where you are most concerned about the battery life (and hence, power consumption) of your body sensor.

1) The algorithm for detecting heart failure requires 1,000,000 lines of code, with each instruction requiring exactly 4 cycles/instruction. Exactly how many cycles need to be executed to complete this algorithm? If the algorithm needs to be completed in at least 1 second, what is the minimum clock frequency allowable that will enable completion of the algorithm? What is the optimal execution time for this design? Assume the architecture is a conventional 5-stage pipeline, and there are no cache misses and data dependencies.

2) The total capacitance switching on the chip is 1nF. (10-9) Assuming the supply voltage (VDD) is 1V, the activity factor is 100% switching of all gates, what is the maximum power consumption (in Watts)?

3) Assume now that the algorithm is very parallelizable, and in fact—you can run simultaneously 10 instructions/cycle with NO data hazards, dependencies or loss in performance due to parallelization overhead. Assuming clock frequency is EXACTLY linearly proportional to Supply Voltage (VDD), what is the new, optimal execution time and power consumption for this parallel processor architecture?