CSE 410 Midterm Exam 5/04/09

Name ________________________________

Do **not** write your id number or any other confidential information on this page.

There are 8 questions worth a total of 100 points. Please budget your time so you get to all of the questions. Keep your answers brief and to the point.

You will want to use a copy of the "green card" from the textbook. We have additional copies if you did not bring one with you. You may also have a sheet of hand-written notes if you brought them. Other than that, the exam is closed book, closed notes, closed calculators, closed laptops, closed twitter, closed telepathy, etc.

Please wait to turn the page until everyone is told to begin.

Score __________________ / 100

1. _______ / 14

2. _______ / 15

3. _______ / 12

4. _______ / 12

5. _______ / 12

6. _______ / 15

7. _______ / 12

8. _______ / 8

Question 1. (14 points) This question involves **8-bit signed, 2's complement** binary numbers.

(a) Give the 8-bit 2's complement binary and hexadecimal representations of the decimal number +76.

(b) Give the binary and decimal values of the 8-bit 2's complement hexadecimal number 0xE8.

Powers of 2 and 16 for reference.

Number	Hex	Decimal
2^0	16^0	1
2^1		2
2^2		4
2^3		8
2^4	16^1	16
2^5		32
2^6		64
2^7		128
2^8	16^2	256
2^9		512
2^{10}		1024
2^{11}		2048
2^{12}	16^3	4096

Question 2. (15 points) For each of the following MIPS assembly language instructions:

i. Circle whether the instruction is a real machine instruction or an an assembly-language pseudo instruction, and

ii. If the instruction is a pseudo-instruction, show the actual MIPS instruction(s) that the assembler would generate for the pseudo instruction. You should give the actual machine instructions in assembly language; do not translate the instructions to binary or hex. If there is more than one possible machine instruction sequence that implements the pseudo-instruction, we will give credit for a good answer that produces the correct result even if it is not exactly the same as that which SPIM would produce.

(a) `li   $s0, 17`

Circle: machine instruction assembler pseudo-instruction

Actual machine instruction(s) if this is a pseudo-instruction:

(b) `ori  $t3, $a0, 0xFF`

Circle: machine instruction assembler pseudo-instruction

Actual machine instruction(s) if this is a pseudo-instruction:

(c) `bge  $s0, $t0, there`

Circle: machine instruction assembler pseudo-instruction

Actual machine instruction(s) if this is a pseudo-instruction:

Question 3. (12 points) Suppose we have a 32-bit MIPS word containing the value `0x008A1021`. We would like to know what MIPS machine instruction this represents.

(a) Write this instruction word in binary. Leave enough spaces between the digits for part (c) of the question.

(b) What is the format of this instruction? (circle)

R I J

(c) In your answer to part (a), draw boxes around the bits that make up the different fields of the instruction and then label the instruction fields (opcode, rs, etc.)

(d) Translate this instruction to assembly language. Use symbolic register names like `$t8` instead of absolute register numbers like `$24`.

Question 4. (12 points) Suppose we execute the following MIPS instructions

```
   li   $t0, 2
   li   $t1, 5
   slt  $t2, $t1, $t0
   beq  $t2, $zero, skip
   addi $t1, $t2, 3
skip:
   li   $v0, 42
```

In the following table, write down each of the registers changed during execution and their values after the code has executed. The first column is filled in for you.

Register	`$t0`						
Value	2						

Question 5. (12 points) In a processor implementation, a *data hazard* can slow down the pipeline.

What is a *data hazard*? Give a short example using MIPS code that illustrates the problem and give a brief explanation of what the problem is.

Question 6. (15 points) A little programming. Implement a function in MIPS assembly language to return the larger of its two integer arguments. In a C-like programming language, this function would be specified as follows:

```
/* return the larger value of a or b */
int max(int a, int b) { ... }
```

You should use the standard MIPS calling conventions for this function. However, since this is a leaf function, you do not need to allocate a stack frame if your solution does not need one. You can use regular assembly language pseudo-instructions in your solution if you wish. Don’t worry if your solution is short – it might not take that much code to do the job. Comments are useful to help the grader understand your code.

```
# return the maximum value of the two integer arguments of
# this function.  If the arguments have the same value,
# return either one.

max:
```

Question 7. (12 points) One of the terms we encountered in discussing memory hierarchies and caches was *spatial locality*. Give a brief explanation of what this term means and an short example that gives one illustration of this property.

Question 8. (8 points). Suppose we interpret the following word of memory as a sequence of ASCII characters. What does it say? (i.e., translate from hex to ASCII). Hint: The MIPS reference card (green card) is your friend.

```
0x42796521
```