CSE 410 - Spring 2006
Midterm Exam

Closed book, no notes, no electronic devices

The MIPS Reference Data sheet is provided as the last page of the exam. Use it to help you answer the questions!

Name: _____

UWNetID: _____

This page is for use by the grader to record your scores.

page 1	_____	of 9
page 2	_____	of 12
page 3	_____	of 6
page 4	_____	of 8
page 5	_____	of 12
page 6	_____	of 8
Total	_____	of 55

CSE 410 Sp06 Midterm Exam

1. [4] In the MIPS architecture that we are studying, the expression *one word of memory* has a specific meaning.

a. How many bytes are there in one word of memory in this design?

b. How many bytes are there in a memory address?

c. Assume that the address of a particular word in memory is word-aligned. Is bit 0 (ie, the low order bit) of that address 0 or is it 1? Why?

2. [2] Give one example of the simplifications that make a RISC design less complicated than CISC and briefly describe why it makes it easier to design a faster or smaller implementation of the architecture.

3. [3] The expression *calling conventions* describes the way that procedures interface with each other during a procedure call, including managing the stack, expected register usage, and maintaining the return address. We have been using one consistent set of calling conventions in all the code that we have read or looked at in this class.

a. When returning from a call, a called procedure must guarantee that the values of some registers are unchanged from the values they had when the procedure was called. Is $ra one of these registers? Yes No

b. The stack pointer is set and restored when a procedure needs a stack frame for use during execution. What benefit is gained by maintaining a stack pointer that is a multiple of 8?

4. [4] Some procedures create a stack frame when they start executing and discard it when they finish, but other procedures do not. Describe a particular situation in which a procedure author would decide that a stack frame was necessary. What specifically would the stack be used for in your example?

5. [4] Translate the C language statement below into MIPS assembly language instructions. You can assume that `counter` is the label of an integer (one 32-bit word) stored in main memory. When your code snippet completes, the value in memory should be appropriately updated. Note: You are not writing a complete procedure, just the few instructions needed to implement this line of code.

```
counter++;
```

6. [4] Describe two differences between the representation used for `.asciiz` strings that we are using (also known as C strings) and the representation used for Java strings.

7. [2] There are several different instructions that can be used to change the control flow in a MIPS program. Selecting from `j, jal, jr, beq` and `bne`, which instruction has the greatest range? (In other words, which instruction can be used to go the furthest away relative to where the instruction is located?) Why is this true?

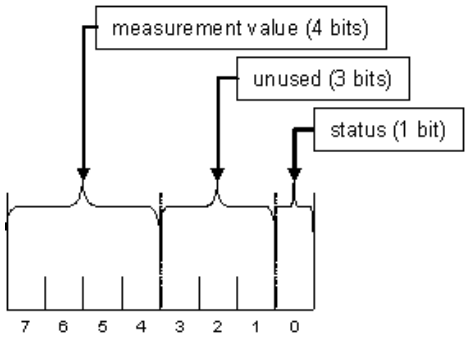
8. [2] Draw separation lines and label the sign bit, biased exponent, and mantissa fields in the following 32-bit floating point representation of 3.14159274 (the closest we can get to π with single-precision floating point).

```
0100 0000 0100 1001 0000 1111 1101 1011
```

9. [2] What is an important benefit of constructing single precision floating point numbers with the sign bit, biased exponent, and mantissa arranged the way they are in the word?

10. [3] Imagine that you have an electronic sensor that reports its measurements in an 8-bit byte, formatted as shown here.

The low order bit (bit 0) is the status bit. Bits 1, 2, and 3 are not used. Bits 4 through 7 contain the measurement that the device is reporting.



a. How many different measurement values can be reported by this device using this format?

b. If the measurement values are considered to be signed integer numbers represented in two's complement format, what is the largest possible positive measurement value that can be reported?

c. Again assuming two's complement notation, what 8-bit value would be returned when the device reported status=1, unused=0, and measurement = -1?

11. [5] Consider the hex value 0x20030007 as representing one MIPS machine language instruction.

a. Convert the hex value to a 32-bit binary value.

b. What is the opcode value for this instruction?

c. What is the name of the instruction with that opcode?

d. What is the format (R, I, or J) of the instruction?

e. What is the number of the destination register in this instruction?

12. [8] The figure shown here is copied from SPIM and shows part of the HW2 solution after it was assembled and loaded by SPIM. Refer to this figure when answering the following questions.

```
[0x00400090]   0x0008b021  addu $22, $0, $8      ; 81: move    $s6,$t0         # assume x is okay
[0x00400094]   0x0128082a  slt $1, $9, $8        ; 82: ble     $t0,$t1,skip6   # skip if okay
[0x00400098]   0x10200002  beq $1, $0, 8 [skip6-0x00400098]
[0x0040009c]   0x0009b021  addu $22, $0, $9      ; 83: move    $s6,$t1         # clamp to limit
[0x004000a0]   0x0000b821  addu $23, $0, $0      ; 88: move    $s7,$zero       # s7 is the sum
[0x004000a4]   0x3c011001  lui $1, 4097          ; 89: lw      $t0,limit       # t0 == loop index
[0x004000a8]   0x8c280040  lw $8, 64($1)
[0x004000ac]   0x02e8b820  add $23, $23, $8      ; 91: add     $s7,$s7,$t0     # $s7 = $s7 + index
[0x004000b0]   0x2108ffff  addi $8, $8, -1       ; 92: addi    $t0,$t0,-1      # index = index -1
```

a. Which one or more of the seven assembly language instructions written by the code's author are actually part of the core instruction set provided by the MIPS architecture?

b. When the CPU fetches the first instruction in this code, `move $s6, $t0`, what is the value of the PC?

c. The second line in the figure includes the instruction `slt $1, $9, $8`. What are the names (not numbers) of the three registers used in this instruction?

d. The last source instruction in this sequence is `addi $t0, $t0, -1`. The resulting machine language instruction is `0x2108ffff`. Which part of this hex value represents the -1?

13. [4] For the following sequence of MIPS instructions, identify all registers used and their values after the code has executed. The first column of the table is filled in as an example.

```
        li    $t0, 4
        li    $t1, 7
        li    $t2, 3
        sub   $t3, $t1, $t2
        beq   $t0, $t3, next
        add   $s0, $zero, $t3
        j     end
next:
        add   $s0, $t1, $t2
end:
```

Register name:	*t0*				
Register value:	*4*				

14. [4] In MIPS assembly language, there exists a pseudo-op called seq (set equal). It compares two source registers, and sets the destination register to 1 if equal, otherwise 0.

Example format: `seq $v0, $s2, $s3`

Write a short sequence of any valid MIPS assembly instructions (except seq itself) to compare the contents of the source registers `$s2` and `$s3`, and set the destination register `$v0` to 1 if equal, otherwise 0.

15. [4] Suppose $t0 contains the address of the 0^{th} element of an array of 8-bit bytes and $t1 holds the value of index n. Write a short sequence of MIPS instructions to store the value 1 into array[n].

```
la      $t0,byteValues  # address of byte values array
lw      $t1,n           # the index value
```