CS/ECE 354 Examination #3
19 May 1994
Open Book/Note
No Calculators

Name: ______________________________________ Signature: ____

Student ID: ____

Instructor: ____

(1) This exam is open book, open notes. NO calculators.
(2) Write all answers on the exam.
(3) Partial credit may be given if work is shown.

Page	Score	Possible
1	(this page)	
2		18
3		16
4		12
5		18
6		14
7		8
8		14
Total		100

.

(1) (6 points) Consider a MIPS computer consisting of just a CPU and memory (no cache) where the CPU fetches instructions and data from memory. Consider the following code fragment:

```
        li $2, 2
loop:   sw $2, 1000($0)
        addi  $2, $2, -1
        bgtz $2, loop
```

(a) How many instructions will the CPU execute?

(b) How many words will the CPU read from memory?

(c) How many words will the CPU write to memory?

(2) (4 points) Please add numbers in the following 8-bit representations. Write `OVERFLOW` if the answer cannot be represented in 8 bits.

```
  Unsigned            Two’s Complement

  11001011              11001011
+ 01110011            + 01110011
----------            ----------
```

(3) (8 points)

(a) Translate 116 (base ten) into binary (base two).

(b) Translate your answer to part (a) into base 32. Assume that base 32 numbers have the digits with the following decimal values:

```
digit  0 1 2 3 4 5 6 7 8 9  a  b  c  d  e  f  g  h  i  j  k  l  m  n  o  p  q  r  s  t  u  v
value  0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
```

(4) (6 points) Assume that the bit pattern 11000000 is stored in a computer's memory. What number does the pattern represent in each of the following representations?

(a) 8-bit two's complement

(b) 8-bit sign magnitude

(c) 8-bit biased-128

(5) (6 points) The following are binary representations of IEEE single precision floating point numbers which are being added. Find the IEEE single precision floating point representation of the sum. Express the answer in both binary and hexadecimal.

```
     0 1000 0110   11000000100000000000000
 +   0 1000 0110   01011111000000000000000
 -----------------------------------------
```

(6) (4 points) What is the contents of `$10` when this MAL program terminates?

```
__start: li    $10, 0
         jal   kall
         add   $10, $10, 1
         j almost
kall:    jal   proc
         add   $10, $10, 10
         jr    $31
proc:    add   $10, $10, 100
         sub   $31, $31, 8
         jr    $31
almost:  done
```

The contents of `$10`:

(7) (6 points) A MIPS RISC assembler allows the following addressing mode for a MAL store instruction:

```
sw  $8, label($20)
```

An effective address is computed by adding the 32 bit address, given by `label`, to the contents of register $20. Give a TAL instruction sequence for this MAL instruction if `label` has been assigned the address `0x0ade1234`.

(8) (6 points) Assume that the contents of the memory and registers are as follows:

```
Memory Address     Value
   -------        -------
   0x10              2
   0x11            0x10
   0x12            0x13
   0x13            0x13
   0x14            0x10
   0x15             23


   Register        Value
   --------       -------
     $4              4
     $6            0x14
     $7            0x10
```

What is the value stored in `$8` (and `$6` for part 5.) after each the following is executed starting with the above memory and register values?

```
1. immediate
        move_byte $8, 0x12            $8 =
2. register
        move_byte $8, $7              $8 =
3. register direct
        move_byte $8, $6              $8 =
4. base displacement
        move_byte $8, 0x11($4)        $8 =
5. Autoincrement or Postincrement (where the increment
   happens after the memory access)
        move_byte $8, ($6)+           $8 =        $6 =
```

(9) (6 points) If the byte address of the element [5, 2] of a 8-by-10, two-dimensional array of integer is 1080, what is the base address of this array? Assuming the array is stored in row-major order, and the first element of each row and column is numbered 1.

(10) (4 points) Consider the following code for handling I/O from the keyboard.

```
WaitLoop:  lw    $2, KeyboardData
           lw    $14, KeyboardStatus
           bltz  $14, WaitLoop
```

This spin-waiting implementation of I/O will not guarantee that $2 will contain the correct character (the character whose key was pressed) upon exiting the loop. Describe what can go wrong with this implementation.

(11) (4 points) Suppose the exception handler provided in chapter 12 is implemented in your computer and you want to do a `puts` on the string `aBcDe`. Assume the display is initially idle and will remain idle after the `puts`.

How many interrupts will this task generate? Be sure to include the final interrupt when the display goes idle after printing `e`. Explain your answer.

(12) (4 points) Assume that a base computer design spends one-third of its time executing branch instructions. You come up with an enhanced design that cuts branch execution time in half. Write an expression that gives the speedup for this enhancement (that is, the enhanced computer's speed divided by the base computer's speed). You do NOT need to solve the expression, since calculators are not allowed.

(13) (4 points) How many total bytes are there in the Cray-1 vector registers (the V-file)? Show your work. Recall that the Cray-1 has eight vector registers and uses 64 bits to store a single floating-point number.

(14) (4 points) Machine U is an unpipelined implementation of MIPS, while machine P uses the same technology to implement a six-stage pipeline.

(a) What is the maximum speed difference between Machine P and Machine U? Why?

(b) In practice, the speed difference will usually be less than the maximum possible. Give two different reasons why.

(15) (6 points) Consider a memory organization that puts a cache between the registers and the main memory. Consider data loads only and ignore instruction fetches. The following statistics were taken from a computer that has this memory organization and is running the program Iguana:

(i) number of cycles required to load data from cache memory to a register: 2 cycles

(ii) number of additional cycles required to load data from main memory to a register in the event of a cache miss: 15 cycles

(iii) number of loads attempted while running program Iguana: 200

(iv) number of loads in (iii) that hit in the cache: 160

Use these statistics to answer the following two questions.

(a) For program Iguana, what is the miss ratio for loads?

(b) For program Iguana, what is the average number of cycles to load a register?

(16) (8 points) You are part of a team of programmers writing a large application in MAL. You must add code to a procedure named chocolate which is called by the main program and calls a procedure named vanilla. Both main and vanilla are written by someone else. Procedure chocolate accepts an input parameter in $4 and returns an output value in $2. Procedure vanilla accepts an input parameter in $6 and returns an output value in $3. Otherwise, you know nothing about procedure vanilla.

The code for procedure chocolate is given below. Code needs to be added to chocolate in order to implement a convention to be used for preserving registers across procedure calls that is adopted by the entire project team. Otherwise, consider procedure chocolate to be complete.

```
chocolate:  sub   $sp, $sp, 4    # save return address on system stack
            sw    $31, 4($sp)

            move  $19, $4
            li    $12, 15
            move  $6, $19
            jal   vanilla
            sub   $2, $19, $3

            lw    $31, 4($sp)    # restore return address and return
            add   $sp, $sp, 4
            jr    $31
```

(a) Assume the project team adopts the convention that the calling program is responsible for preserving registers $8 through $25, so that you add code to chocolate to implement this convention. Considering only registers $8 through $25, list those registers whose values must be saved and later restored in the code you add to chocolate. Do not list any registers whose values do not need to be preserved. If no registers need to be preserved, write NONE as your answer.

(b) Alternatively, assume the project team adopts the convention that the called procedure is responsible for preserving registers $8 through $25, so that you add code to chocolate to implement this convention. Considering only registers $8 through $25, list those registers whose values must be saved and later restored in the code you add to chocolate. Do not list any registers whose values do not need to be preserved. If no registers need to be preserved, write NONE as your answer.

(17) (6 points) Consider an architecture that does not support memory-mapped I/O. Instead, special instructions for accessing the I/O channels are provided. Let the instructions for accessing the display I/O channel be defined as follows.

```
pdsp   R         puts the character contained in register R
             onto the display data I/O channel

bdnr   addr      conditional branch instruction that branches
                 to addr if the display is not ready to accept
                 another character
```

Write a few lines of MAL code that includes the above instructions to perform a spin waiting loop for putting the character in register $4 onto the display monitor.

(18) (8 points) Recall that bits 3-0 of the MIPS Status register are:

Bit	Function
0	1 if interrupts are currently enabled (0 if disabled)
1	1 if current mode is "user" (0 if "kernel")
2	1 if interrupts were enabled before most-recent exception (0 if were disabled)
3	1 if mode was "user" before most-recent exception (0 if was "kernel")

Assume that the exception handler provided in chapter 12 has just been invoked because of an interrupt. What are the values of the bits 0-3 of the Status register? Why? State any additional assumptions that you need to make.

```
Bit   Value   Explanation
---   -----   -----------

0

1

2

3
```