March 27, 2008

COMPUTER ENGINEERING DEPARTMENT

ICS 233

COMPUTER ARCHITECTURE & ASSEMBLY LANGUAGE

Major Exam I

Second Semester (072)

Time: 1:00-3:30 PM

Student Name : ____

Student ID. : ____

Question	Max Points	Score
Q1	30	
Q2	15	
Q3	15	
Q4	20	
Q5	20	
Total	100	

Dr. Aiman El-Maleh

[30 Points]

(Q1) Fill in the blank in each of the following questions:

(1) The smallest (negative) number that can be represented using 32-bit 2`s complement in hexadecimal is _________ and the largest positive number in hexadecimal is _________.

(2) Assuming 8-bit representation of numbers, the binary number 10101110 is equal to _________ in sign-magnitude representation, _________ in 1`s complement representation, and _________ in 2`s complement representation.

(3) Two advantages of programming in assembly language are ________________________________ and __________________________________.

(4) The advantage of dynamic RAM over static RAM is that ____ but the disadvantage is ____.

(5) __________________memory is used to bridge the widening speed gap between CPU and main memory.

(6) Memory hierarchy consists of the following from highest speed to lowest speed: __________________, _________________, ________________, ____________________ and _______________________.

(7) The following assembler directive allocates ______ words initialized by ______.

X: .word 5:20

(8) With a 36-bit address bus and 64-bit data bus, the maximum memory size than can be accessed by a processor is ________ Byte and the maximum number of bytes that can be read or written in a single cycle is ________ Bytes.

(9) Given a magnetic disk with the following properties:

- Rotation speed = 7200 RPM (rotations per minute)
- Average seek = 8 ms, Sector = 512 bytes, Track = 200 sectors

The average time to access a block of 100 consecutive sectors is ________________ ms.

(10) Assuming the following data segment, and assuming that the first variable X is given the address **0x10010000**, then the addresses for variable Y and Z will be ______________ and ______________.

```
.data
X:      .byte  1, 2, 3
Y:      .half  4, 5, 6
Z:      .word  7, 8, 9
```

(11) The code given below prints the statement: __________________________. *Note that the ASCII code for the line feed character is 10 and the ASCII code for the carriage return character is 13.*

```
MSG: .ascii  "Exam1",13
     .ascii " ICS 233"
     .ascii "is so easy !!",0

     li $v0, 4
     la $a0, MSG
     syscall
```

(12) Assume that the instruction j NEXT is at address 0x00401FC4 in the text segment, and the label NEXT is at address 0x0040003C. Then, the address stored in the assembled instruction for the label NEXT is _______________.

(13) Assume that the instruction beq $t0, $t1, NEXT is at address 0x00401FC4 in the text segment, and the label NEXT is at address 0x0040003C. Then, the address stored in the assembled instruction for the label NEXT is _____________.

(14) Assuming that $a0 contains an Alphabetic character, the instruction *ori $a0, $a0, 0x20* will guarantee that the character in $a0 is _________________. Note that the ASCII code of character ‘A’ is 0x41 while that of character ‘a’ is 0x61.

(15) Assume that you are in a company that will market a certain IC chip. The cost per wafer is $3000, and each wafer can be diced into 2000 dies. The cost per good die is $3. Then, the yield of this manufacturing process is ___________________.

[15 Points]

(Q2) Using only basic MIPS instructions, write the shortest sequence of instructions to implement each of the following pseudo instructions:

1. *sgt $t0, $t1, $t2* #$t0 is set to 1 if $t1 is greater than $t2

2. *move $t0, $t1* # $t0 = $t1

3. *ble $t0, 5, Next* # branch to Next if $t0 is less than or equal 5

4. *abs $t0, $t1* #$t0 is loaded with the absolute value of $t1

5. *ror $t0, $t0, 8* #$t0 is rotated to the right by 8 bits and stored in $t0

[15 Points]

(Q3) Answer the following questions. Show how you obtained your answer:

(i) Given that **TABLE** is defined as: **TABLE: .word 1, -1, 2, 50, -20, 16**

Determine the content of register**s $v0** and **$v1** after executing the following code:

```
        la      $a0, TABLE
        addi    $a1, $a0, 20
        move    $v0, $a0
        lw      $v1, 0($v0)
        move    $t0, $a0
loop:   addi    $t0, $t0, 4
        lw      $t1, 0($t0)
        bge     $t1, $v1, skip
        move    $v0, $t0
        move    $v1, $t1
skip:   bne     $t0, $a1, loop
```

(ii) Given that **TABLE** is defined as shown below, determine what will be printed by the following program:

TABLE: .ascii "0123456789ABCDEF"

```
        li $t0, 0x12EF67DC
        li $t3, 8
loop:
        rol  $t0, $t0, 4
        andi $a0,$t0, 15
        la $t1, TABLE
        addu $t1, $t1, $a0
        lb $t1, 0($t1)
        move $a0, $t1
        li $v0, 11
        syscall
        sub $t3, $t3, 1
        bne $t3, $zero, loop
```

(iii) Given that **Array** is defined as shown below, determine the content of **Array** after executing the following code:

Array: .byte 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12

```
    la $a0, Array
    li $a1, 4
    li $a2, 0
    li $a3, 2

    mul $t0, $a1, $a2
    add $t0, $t0, $a0
    mul $t1, $a1, $a3
    add $t1, $t1, $a0
Next:
    lb $t3, ($t0)
    lb $t4, ($t1)
    sb $t3, ($t1)
    sb $t4, ($t0)
    addi $t0, $t0, 1
    addi $t1, $t1, 1
    addi $a1, $a1, -1
    bnez $a1, Next
```

[20 Points]

(Q4) Write separate MIPS assembly programs to do each of the following using the smallest possible number of instructions.

(i) Multiply the content of register $s1 by 15.25.

(ii) Count the number of 1's in register $s1.

(iii) Ask the user to enter a character, c1. Then, in a new line ask the user to enter another character, c2, greater than the first character. Then, in a new line print the characters from character c1 until character c2 as shown in the format below. If the entered character is smaller than the first character ask the user to reenter the second character.

Enter a character: B
Enter another character greater than B: A
Enter another character greater than B: G
The range of entered characters is: B C D E F G

[20 Points]

(Q5) Write a MIPS assembly program, **BinarySearch,** to search an array which has been previously sorted in an ascending order. Each element in the array is a 32-bit signed integer. Assume that the address of the array to be searched in stored in \$a0, the size (number of elements) of the array is stored in \$a1, and the number to be searched is stored in \$a2. If the number is found then the program returns in \$v0 register the position of the number in the array. Otherwise, 0 is returned in \$v0.

The pseudocode for the **BinarySearch** algorithm is given below:

```
BinarySearch (array, size, number) {
        lower = 0;
        upper = size-1;
        while (lower <= upper) {
                middle = (lower + upper)/2;
                if (number == array[middle])
                        return middle;
                else if (number < array[middle])
                        upper = middle–1;
                else
                        lower = middle+1;
        }
        return 0;
}
```