Name: KEY **Id#**

ICS 233, Term 141

Computer Architecture & Assembly Language

Quiz# 2

Date: Sunday, Oct. 26, 2014

Q1. Fill in the blank in each of the following questions:

(1) Assuming variable Array is defined as shown below:

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

The content of register $t0 after executing the following sequence of instructions is 0x08070605.

```
la $t0, Array
lw $t0, 4($t0)
```

(2) Assume that the instruction j NEXT is at address 0x00400030 in the text segment, and the label NEXT is at address 0x004000a8. Then, the address stored in the assembled instruction for the label NEXT is 0x004000a8/4=0x010002a.

(3) Assume that the instruction bne $t0, $t1, NEXT is at address 0x00400030 in the text segment, and the label NEXT is at address 0x004000a8. Then, the address stored in the assembled instruction for the label NEXT is (0x004000a8-0x00400034)/4=0x00000074*/4*= 0x001d.

(4) Assuming that $a0 contains an Alphabetic character, the instruction *andi $a0, $a0, 0xDF* will guarantee that the character in $a0 is always an upper case character. Note that the ASCII code of character 'A' is 0x41 while that of character 'a' is 0x61.

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

1. *xori $t0, 0x12345678* #$t0 is xored with the 32-bit value 0x12345678

```
lui $at, 0x1234
ori $at, $at, 0x5678
xor $t0, $t0, $at
```

2. *ble $t0, $t1, Next* # branch to Next if $t0 is less than or equal to $t1

```
slt $at, $t1, $t0
beq $at, $0, Next
```

3. *bgt $t0, 100, Next* # branch to Next if $t0 is greater than 100

```
addi $at, $0, 100
slt    $at, $at, $t0
bne  $at, $0, Next
```

4. *neg $t0, $t1* #$t0 is loaded with the negative value of $t1

```
sub $t0, $0, $t1
```

5. *rol $t0, $t0, 12* #$t0 is rotated to the left by 12 bits and stored in $t0

```
srl $at, $t0, 20
sll $t0, $t0, 12
or $t0, $t0, $at
```

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

i. Determine the content of register **$s1** after executing the following code:

```
ori $s1, $zero, 12
sll $t0, $s1, 3
sub $t0, $t0, $s1
sra $t1, $s1, 2
add $s1, $t0, $t1
```

The content of $s1=87=0x57. The code computes the result of multiplying the content of $s1 by 7.25 = 12*7.25=87.

ii. Determine the content of register **$t2** after executing the following code:

```
        li $s1, 0x5A
        and $t2, $zero, $t2
Next:
        andi $t1, $s1, 1
        add $t2, $t2, $t1
        srl  $s1, $s1, 1
        bne $s1, $0, Next
```

The content of $t2=4=0x4. The code counts the number of 1's in register $s1.

iii. Given that **TABLE** is defined as: **TABLE: .word 7, 10, -4, 5, 20, 13**

Determine the content of register**s $t2** after executing the following code:

```
        la      $t0, TABLE
        li      $t1, 6
        lw      $t2, ($t0)
loop:   addi    $t0, $t0, 4
        lw      $t3, ($t0)
        bge     $t3, $t2, skip
        move    $t2, $t3
skip:   addi    $t1, $t1, -1
        bne     $t1, $0, loop
```

The content of $t2=-4=0xfffffffc. The code finds the minimum value in Table and stores it in $t2.