Name: KEY **Id#**

ICS 233, Term 141

Computer Architecture & Assembly Language

Quiz# 3

Date: Sunday, Nov. 2, 2014

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

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

Array: .byte 1, -1, 2, -2, 3, -3, 4, -4

The content of register $t0 after executing the following sequence of instructions is 0xfffffe02

```
la $t0, Array
lh $t0, 2($t0)
```

(2) Assume that the instruction j NEXT is at address 0x00400094 in the text segment, and the label NEXT is at address 0x0040008C. Then, the address stored in the assembled instruction for the label NEXT is 0x0040008C/4=0x0100023.

(3) Assume that the instruction bne $t0, $t1, NEXT is at address 0x00400094 in the text segment, and the label NEXT is at address 0x0040008C. Then, the address stored in the assembled instruction for the label NEXT is (0x0040008C-0x00400098)/4=0xFFFFFFF4/4= 0xFFFD.

(4) Assuming that $a0 contains an Alphabetic character, the single instruction *xori $a0, $a0, 0x20* will convert an upper case character in $a0 to a lower case, and a lower case to an upper case. Note that the ASCII code of character 'A' is 0x41 while that of character 'a' is 0x61.

Q2. Write separate MIPS assembly code fragments to do the following minimizing the number of executed instructions (Pseudo instructions can be used):

i. Count the number of 0's in register $a0 and store the result in $v0. Assume that the number of 0's is larger than the number of 1's in register $a0. Assume that the content of register $a0 needs to be preserved.

```
            and $v0, $zero, $v0
            move $t0, $a0
    Next:
            andi $t1, $t0, 1
            add $v0, $v0, $t1
            srl  $t0, $t0, 1
            bne $t0, $0, Next
            li  $t0, 32
            sub $v0, $t0, $v0
```

ii. Implement the following high-level statement assuming that the registers contain signed numbers:

```
            if ( ($sl > 10  ||  $s1<=100) && $s1 > $s2) {$s3 = 1;}
```

```
            ble $s1, $s2, EndIf            # skip if false
            bgt $s1, 10, If                # yes, execute if part
            bgt $s1, 100, EndIf            # no: skip if part
    If:     li  $s3, 1                     # set $s3 to 1
    EndIf:
```