Computer Architecture and Assembly Language

Second Midterm (07/05/2011)-30 Marks

BS-IV

S. M. Faisal Iradat

Name:_____

1) For the following Intel pseudo-instruction, choose a MIPS instruction that will perform the same functionality.

```
mov     ecx, edx        # ecx = edx (ecx and edx are both 32bits wide)
```

a. add $s0, $s0, $s1
b. add $s0, $s1, $zero
c. add $s0, $zero, $s1
d. **both B and C**
e. None of the above

(2 Marks)

2) For the following MIPS instruction, choose the right the hexadecimal value(s) given below for corresponding machine language instruction.

```
sw $t0, 4($s0)
```

a. 0x81100004
b. 0xAD100004
c. **0xAE080004**
d. 0x8E080004

(2 Marks)

3) For the following machine language instruction in hexadecimal choose the right MIPS instruction from the choices below.

0x02298020

a. addi $s0, $t0, 20
b. **add $s0, $s1, $t1**
c. slt $t0, $s0, 4
d. None of the above

(2 Marks)

4) Given the list design principles and their examples for the MIPS architecture, identify the principles and to each choose the right example.

~~a) The introduction of the I-format for data transfer instructions.~~
b) Make common case fast P3-c
~~c) Arithmetic Operations~~
d) Smaller is faster. P2-f
e) Simplicity favors regularity. P1-h
~~f) Limiting the size of the register set.~~
g) Good designs require good compromises. P4-a
~~h) Requiring that all MIPS assembly instructions be 32 bits in length.~~

(8 Marks)

5) For the following C code write a MIPS assembly code assuming $s1 has the base address of array 'arr' and 10th element of 'arr' is at the memory address 40. Also write the corresponding machine language code.

arr[11] = arr[10] + arr[12];

```
lw $t0, 40($s1)    8e280028
lw $t1, 48($s1)    8e290030
add $t0, $t0, $t1  01094020
sw $t0, 44($s1)    ae28002c
```

(8 Marks)

6) Determine the output of the following program

```
.data
        A:      .word   0, 0, 0, 0, 0, 0, 0, 0, 0, 0
.text
main:
        add     $s0, $zero, $zero
        addi    $s1, $zero, 11
        li $t1,11
        la      $s2, A
loop1:
        slt     $t0, $s0, $s1
        beq     $t0, $zero, end
        sw      $s0, 0($s2)
        addi    $s2, $s2, 4
        addi    $s0, $s0, 1
        blt $s0,$t1,loop1
        li $t1,11
        la      $s2, A
loop2:
        lw      $a0, 0($s2)
        addi    $s2, $s2, 4
        li      $v0,1
        syscall
        addi $t0,$t0,1
        blt     $t0,$t1,loop2
end:
        li $v0,10
```