課程: **Computer Organization,**
國立中山大學資訊工程學系, 教師：黃英哲

Chapter Exam

Chapter 2.1~2.9-instructions:language of the computer

2019/04/30

*Please write the answer on your answer sheet

1. Please use a 4 bytes data ***0x1234abcd*** for example to describe what would the data stored in the following Little-Endian memory architecture and Big-Endian memory architecture? **(20%)**

Memory address	Little-Endian	Big-Endian
00		
01		
02		
03		

2. Assume variable h is associated with register $s2 and the base address of the Array A is in $s3. Now the C assignment statement is as below：

A[8] = h +A[2]

Write the answers of (A)、(B)、(C) according to the following MIPS assembly code. **(15%)**

```
lw   $t0 , __(A)__ ($s3)
add  __(B)__, __(C)__, $t0
sw   $t0, 32($s3)
```

3. A recursive C language statement is as:

```
Void swap(int v[], int k)
{
    Int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1] = temp;
}
```

課程: Computer Organization,
國立中山大學資訊工程學系, 教師：黃英哲

The corresponding MIPS assembly code is as:

```
swap : muli   $t1, $a1, _(A)_
       add    $t1, $a0, $t1
       lw     $t0, _(B)_
       lw     $t2, _(C)_
       sw     $t2, _(D)_
       sw     $t0, _(E)_
```

```
void swap(int v[], int k){
    int temp;
    temp = v[k];
    v[k] = v[k+1];
    v[k+1]= temp;
}
```

Now you have three temporal registers, **$t0, $t1, $t2**, $a1 for **k** , the base address of array **v[ ]** is $a0 . Write the answers of (A)~(E). **(20%)**

4. What is the MIPS assembly code to load the following 32-bit constant into register $s0? **(10%)**

32-bit constant: 0000 0000 1100 1010 0000 0000 0001 1000

10	11	12	13	14	15
a	b	c	d	e	f

5. Converting the following MIPS codes to C codes. The argument register $a0 corresponds to the parameter variable n. **(20%)**

```
Test:  addi $sp, $sp, -8
       sw   $ra, 4($sp)
       sw   $a0, 0($sp)
       slti $t0, $a0, 1
       beq  $t0, $zero, L1
       addi $v0, $zero, 1
       addi $sp, $sp, 8
       jr   $ra
L1:    addi $a0, $a0, -1
       jal  fact
       lw   $a0, 0($sp)
       lw   $ra, 4($sp)
       addi $sp, $sp, 8
       mul  $v0, $a0, $v0
       jr   ~~$ra~~
            Test
```

```
sp -= 8 ;
ra = sp[1] ;
n = sp[0] ;
t0 = n ≤ 1
if t0 == 0 ⇒ L1
```

Category	Instruction	Example
Arithmetic	add	add $s1.$s2.$s3
	subtract	sub $s1.$s2.$s3
	add immediate	addi $s1.$s2.20
Data transfer	load word	lw $s1.20($s2)
	store word	sw $s1.20($s2)
	load half	lh $s1.20($s2)
	load half unsigned	lhu $s1.20($s2)
	store half	sh $s1.20($s2)
	load byte	lb $s1.20($s2)
	load byte unsigned	lbu $s1.20($s2)
	store byte	sb $s1.20($s2)
	load linked word	ll $s1.20($s2)
	store condition. word	sc $s1.20($s2)
	load upper immed.	lui $s1.20
Logical	and	and $s1,$s2,$s3
	or	or $s1,$s2,$s3
	nor	nor $s1,$s2,$s3
	and immediate	andi $s1,$s2,20
	or immediate	ori $s1,$s2,20
	shift left logical	sll $s1,$s2,10
	shift right logical	srl $s1,$s2,10

Conditional branch	branch on equal	beq $s1.$s2.25
	branch on not equal	bne $s1.$s2.25
	set on less than	slt $s1.$s2.$s3
	set on less than unsigned	sltu $s1.$s2.$s3
	set less than immediate	slti $s1.$s2.20
	set less than immediate unsigned	sltiu $s1.$s2.20
Unconditional jump	jump	j 2500
	jump register	jr $ra
	jump and link	jal 2500

```
L1 :  i--n
      jal fact
      a0 = sp[0]
```

```
v0 = 1
sp += 8
return sp[1] .
```

```
v0 *= a0.
```

6. Converting the following C codes to MIPS codes. In the following code segment, ***f***, ***g***, ***h***, ***i***, and ***j*** are variables. The five variables ***f*** through ***j*** corresponds to the five registers $s0 through $s4. **(25%)**

```
if (i == j)
    f = g + h;
else
    f = g - h;
```