Computer Organization
Chapter 2

ID: Name: 2012/3/14

1. (24%)
(1) For the values in the following table, what is the value of $t2 for the following sequence of instructions:

```
sll  $t2, $t0, 4
or   $t2, $t2, $t1
```

(2) For the values in the following table, what is the value of $t2 for the following sequence of instructions:

```
sll  $t2, $t0, 4
andi  $t2, $t2, -1
```

(3) For the values in the following table, what is the value of $t2 for the following sequence of instructions:

```
srl  $t2, $t0, 3
andi  $t2, $t2, 0Xffef
```

a.	$t0=0x55555555, $t1=0x12345678
b.	$t0=0xBEADFEED, $t1=0xDEADFADE

ANS:

	(1)	(2)	(3)
a.	0x57755778	0x00005550	0x0000AAAA
b.	0xFEFFFEDE	0x0000EED0	0X0000BFCD

2. (15%)An example of MIPS machine assembly language notation, op a, b, c means an instruction with an operation op on the two variables b and c, and to put their result in a. Now a C language statement is as:

```
f = (g + h) – (i + j);
```

The variables f, g, h, i, and j can be assigned to the registers $s0, $s1, $s2, $s3, and $s4, respectively. Now we use two temporary registers $t0 and $t1 to write the compiled MIPS assembly code as follows:

```
add  $t0,  $s1,  (A)
(C)  $t1   $s3,  (B)
sub  $s0,  (D)   (E)
```

Please fill the results on the blank (A), (B), (C), (D), and (E).

ANS:

(A)	(B)	(C)	(D)	(E)
$s2	$s4	add	$t0	$t1

3. (8%)Suppose register $s0 has the binary number
1111 1111 1111 1111 1111 1111 1111 1111 two
and that register $s1 has the binary number
0000 0000 0000 0000 0000 0000 0000 0000 two
What are the values of registers $t0 and $t1 after these two instructions?

```
slt   $t0, $s0, $s1    #set on less than signed comparison
sltu  $t1, $s0, $s1    #set on less than unsigned comparison
```

ANS: (1) $t0 = 1
(2) $t1 = 0

4. (3%)Fill in the appropriate term or terminology for the underline fields:
Move $s1, $zero = addi __, __, ___

ANS : $s1, $zero, 0

5. (10%)While executing the MIPS code shown below, what is the target address of the branch instruction if it is taken?(Assume the starting address of this code segment is 28(dec.))

```
lw      $4, 50($7)
beq     $1, $4, 3
add     $5, $3, $4
sub     $6, $4, $3
or      $7, $5, $2
slt     $8, $5, $6
```

Ans: 32+ 4 + 4*3 = 48(dec.)

6. (10%)Please show the five MIPS addressing modes.

Ans:
(1) Register addressing
(2) Base or displacement addressing
(3) Immediate addressing
(4) PC-relative addressing
(5) Pseudodirect addressing

7. (10%) 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[12] = h + A[8]
```

write the compiled MIPS assembly code by filling the blanks (A) ,(B) ,(C).

```
lw    $t0, ___(A)___($s3)
add   ___(B)___ , ___(C)___ , $t0
sw    $t0, 48($s3)
```

Ans:

(A): 32
(B): $t0
(C): $s2

8. (10%)
(1) Which of the following statements confirming to design principle: simplicity favors regularity?
(a) Keeping all instructions in a single size
(b) Always requiring three operands in arithmetic instruction
(c) Keeping the register fields in the same place in each instruction format
(d) Having the same opcode field in the same place in each instruction format

Ans: (a),(b),(c)

(2) About the 32-bit MIPS instructions, which description is correct?
(a) MIPS has 32 registers inside CPU because it is a 32-bit CPU
(b) *add* instruction can not directly store the addition result to memory.
(c) Since memory structure is byte-addressing, the address offset in *beq* instruction is referred to as byte
(d) In MIPS, "branch-if-less-than" is realize using *slt* and *beq/bne*, since its design principle is two faster instructions are more useful than one slow and complicated instruction

Ans: (b)

9. (10%)Please translate following binary code to MIPS instruction.
(a) 00000001 01101010 10100000 00100000
(b) 10001101 00110000 00000000 00110000

Ans:
(a) add $s4,$t3,$t2
(b) lw $0, 48($t1)

$ s0

MIPS register conventions

Name	Register number	Usage	Preserved on call?
$zero	0	The constant value 0	n.a.
$v0-$v1	2-3	Values for results and expression evaluation	No
$a0-$a3	4-7	Arguments	No
$t0-$t7	8-15	Temporaries	No
$s0-$s7	16-23	Saved	Yes
$t8-$t9	24-25	More temporaries	No
$gp	28	Global pointer	Yes
$sp	29	Stack pointer	Yes
$fp	30	Frame pointer	Yes
$ra	31	Return address	Yes

MIPS instruction encoding

op(31:26)								
28-26 31-29	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	R-format	Bltz/gez	jump	Jump&link	Branch eq	Branch ne	blez	bgtz
1(001)	Add immediate	addiu	Set less than imm.	sltiu	andi	ori	xori	Load upper imm
2(010)	TLB	FLPt						
3(011)								
4(100)	load byte	Load half	lwl	load word	lbu	lhu	lwr	
5(101)	store byte	Store half	swl	store word			swr	
6(110)	Lwc0	Lwc0						
7(111)	Swc0	Swc1						

2-0
5-3

op(31:26) =000000 (R-format),func(5:0)								
28-26 31-29	0(000)	1(001)	2(010)	3(011)	4(100)	5(101)	6(110)	7(111)
0(000)	Shift left logical		Shift right logical	sra	sllv		srlv	srav
1(001)	Jump reg.	jalr			syscall	Break		
2(010)	mfhi	mthi	mflo	Mtlo				
3(011)	Mult	multu	div	divu				
4(100)	add	addu	subtract	subu	and	or	xor	Not or(nor)
5(101)			Set l. t.	sltu				
6(110)								
7(111)								