BÀI TẬP CHƯƠNG 2

I. Assembly MIPS sang mã máy (Hex)

1) add $t0, $s1, $t1.

op	rs	rt	rd	shamt	funct.
000000.	10001	01001	01000	00000	32 (100000)

Mã nhị phân: 000000 10001 01001 01000 00000 100000.

Mã Hex: 0x02294020

2. sub $s0, $s1, $s2

op	rs	rt	rd	shamt	funct.
0	17	18	16.	0	34
000000	10001	10010	10000.	00000	100010.

Mã Hex: 0x02328022

3. addi $t0, $t1, 10.

op	rs	rt	Imm
8.	9	8.	10.
001000.	01001	01000.	0000 0000 0000 1010

Hex: 0x2128000A.

5. sw $t1, 32($s1)

op.	rs	rd	Imm
43	17	9	32
101011	10001	01001	00000 00000 100000

Hex: 0x AE290020.

6. lw $t0, -4($s1)

op	rs	rt	Imm
100011	17	8	-4
100011	10001	01000.	1111 1111 1111 1100

Hex: 0x8E28FFFC

7. j Label (Label = 0x0040001c)

op	Address
000010	00 0001 0000 0000 0000 0000 0111

Address = . Label >> 2 = 0000 0000 0100 0000 0000 0000

$0001\ 1100_2 >> 2$ = 0000 0001 0000.0000 0000 0000 0111_2

Hex: 0x08100007.

8. beq $t0, $s1, Skip. (offset = 3)

op	rs	rt	Imm
4	8	17.	3.
000100	01000	10001	0000 0000 0000 0011

Hex: 0x11110003.

9. beq $s2, $s1, Loop (offset = -3)

op	rs	rt	Imm
4	18	17	-3
000100	10010	10001	1111 1111 1111 1101

Hex: 0x1251FEFD.

10 bne $s1 $s2 , Label (Label = 0x 0040001c ; PC = 0x00400010)

12 = 4 + 4x offset → offset = 2

op	rs	rt	Imm
5	17	18.	2

000101. 10001 10010. 0000 0000 0000 0010.

Hex : 0x 1632 0002

II - Mã máy sang assembly MIPS

A. 0x 2109 FFFC = 0010 0001 0000 1001 1111 1111 1111 1100$_2$

addi $t1, $t0, -4

B. 0x 8D280000C = 1000 1101 0010 1000 0000 0000 0000 1100$_2$

lw $t0 , 12 ($t1)

C. 0x 025 38820. = 0000 0010 0101 0011 1000 1000 0010 0000$_2$

add $s1; $s2, $s3.

D 0x AE28 FFFC = .1010 1110 0010 1000 1111 1111 1111 1100.

sw $t0, -4 ($s1)

J 0x AD 3100 18. = 1010 1101 0011 0001 0000 0000 0001 1000.

sw. $s1 , 24 ($t1)

K. 0x 08100009. (PC = 0x 90400018)

0000 1000 0001 0000 0000 0000. 1001

j Label (Label = . 1001 0000 0100 0000. 0000. 0010 0100.

= 0x 90400024.)

$\overset{0000}{\downarrow}$

L. 0x 1109 fffd = 0001 0001 1001 1111 1111 1111 1101$_2$

beq $t0 , $t1 , -3

M 0x 1632 fff 9. = 0001 0110 0011 0010 1111 1111 1111 1001

bne $s1 , $s2 , -7.

II - Chuyển từ ngôn ngữ C sang assembly của MIPS

1)
```
    sll  $t0, $s3, 2
    add  $t0, $t0, $s6.
    lw   $t1, 0($t0)

    sll  $t2, $s4, 2
    add  $t2, $t2, $s6
    lw   $t3, 0($t2)
    add  $t3, $t1, $t3
    sw   $t3, 32($s7)
```

3)
```
    ori  $s3, $zero, 2
    beq  $s3, $s4, else
    add  $s0, $s1, $s2

else:
```

2)
```
    sub  $t0, $s3, $s4
    sll  $t0, $t0, 2
    add  $t0 ; $t0, $s6.
    lw   $t1, 0($t0)

    sw   $t1, 32($s7)
```

4)
```
    addi $s3, $s3, 1.
    slt  $t0, $s4, $s3
    beq  $t0, $zero, else
    add  $s0, $s1, $s2
    j exit
else: sub $s0, $s1, $s2
      sll $s0, $s1, 2
exit:
```

```
5. ori $s3, $zero, 0
while_start :
    slt $t1, $s3, 3.
    beq $t1, $zero, while-end
    addi $s3, $s3, 1.
    j   while_start
while-end:
```

```
6) ori $s3, $zero, 0
for_start :
    slt $t1, $s3, 4
    beq $t1, $zero, for_end
    add $s0, $s1, $s2
    addi $s3, $s3, 1
    j for_start
for_end :
```

```
7) ori $s3, $zero, 3
do_start :
    addi $s3, $s3, -1
    beq $s3, $zero, do_end
    j do_start.
do_end :
```

```
8) sll $t0, $s3, 2
   add $t0, $t0, $s6.
   lw $t1, 0($t0)
   beq $t1, $s0, else
   add $s0, $s1, $s2
   j exit
else: sub $s0, $s1, $s2
exit :
```

```
9)
   sll $t0, $s3, 2
   add $t0, $t0, $s6.
   lw $t1, 0($t0)
   slt $t2, $s4, $s3.
   bne $t2, $zero, else
   add $s0, $s1, $t1
   j exit
else: sub $s0, $s1, $t1
exit:
```

```
10)   sub $s0, $s1, $s2
      jal tinh
      or  $s0, $s1, $s2
      j exit
tinh:
      add $s0, $s1, $s2
      jr  $ra
exit:
```

```
12 tinh2:
      add $t0, $a0, $a1
      add $t1, $a2, $a3.
      sub $v0, $t0, $t0.
      jr $ra
```

```
14) slt $t0, $s3, 5
    ori $t1, $zero, 7
    slt $t2, $t1, $s3
    or  $t3, $t0, $t2.
    bne $t3, $zero, else
    addi $s0, $s0, -1
    j exit.
else: or $s0, $s1, $s2
exit:
```

```
11) ori $s3, $zero, 2
    ori $s4, $zero, 3
    or  $a0, $zero, $s3
    or  $a1, $zero, $s4
    jal tong
    or $s0, $zero, $v0.
tong: add $v0, $a0, $a1.
    jr $ra
```

```
13) or $a0, $zero, $s3
    or $a1, $zero, $s4
    jal tong
    or $t2, $zero, $v0.

    or $a0, $zero, $t2
    ori $a1, $zero, 3
    jal tong
    move $s0, $v0.

tong:
    add $v0, $a0, $a1
    jr $ra
```

15)
```
    ori $t0, $zero, 6.
    slt $t1, $t0, $s3
    slti $t2, $s3, 0.
    or $t0, $t1, $t2
    beq $t0, $zero, else
    and $s0, $s1, $s2
    j exit.
else:
    sub $s0, $s1, $s2
exit:
```

16)
```
b_gua_a_c:
    slt $t0, $a1, $a0
    slt $t1, $a2, $a1
    nor $v0, $t0, $t1
    jr $ra
```

17)
```
func:
    addi $sp, $sp, -4
    sw $ra, 0($sp).
    jal tong
    move $t0, $v0.
    sub $t1, $a2, $a3
    or $a0, $zero, $t0
    or $a1, $zero, $t1
    jal tong
    addi $sp, $sp, 4
    lw $ra, 0($sp)
    jr $ra
tong:
    add $v0, $a0, $a1
    jr $ra
exit:
```