Computer Architecture & Organization HW#2 — 21700034 곽영훈

2.9 Translate the following C code to MIPS. Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3 and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively. Assume that the elements of the arrays A and B are 4-byte words:

B[8] = A[i] + A[j];

Explanation:

[op]	[rs]	[rt]	[rd]	[shamt]	[funct]
6bits	5..	5..	5..	5..	6..
opcode	first operand Register	Second	Destination Register	the amount of Shift	function code

the elements of the arrays A and B : 4-byte words (2^2)

=> variables i, j ($s3, $s4) (Shift left logical, sll) => sll $t0, $s3, 2
sll $t1, $s4, 2

Variables i, j ($t0, $t1) → base address of the array A ($s6) (A[i], A[j])

=> add $t0, $t0, $s6 , add $t1, $t1, $s6

Move words (A[i], A[j]) ($t0, $t1) from memory to register

=> lw $t0, 0($t0) , lw $t1, 0($t1)

(offset is 0 — no constant, register to access memory)

Add two arrays (A[i], A[j]) ($t0, $t1) to array B ($t2 → $s7)

=> add $t2, $t0, $t1

Store word ($t2) to B[8]

=> sw $t2, 32($s7)

32 ← B's Index (8) × 4 = 32

Answer :

```
sll $t0, $s3, 2
add $t0, $t0, $s6
lw $t0, 0($t0)
sll $t1, $s4, 2
add $t1, $t1, $s6
lw $t1, 0($t1)
add $t2, $t0, $t1
sw $t2, 32($s7)
```

2.14 Provide the type and assembly language instruction for the following binary value: 0000 0010 0001 0000 1000 0000 0010 0000_{two}

R-format

op	rs	rt	rd	shamt	funct	total
6bits	5..	5..	5..	5..	6..	32 bits

000000_2	10000_2	10000_2	10000_2	00000_2	100000_2
op	rs	rt	rd	shamt	funct
11	11	11	11	11	11
0	16	16	16	0	32

function value: 32 ⟹ add

rs, rt, rd value : 16 ⟹ $s0 (reg)

opCode, Shift amout value : 0

Answer : R-format / add $s0, $s0, $s0

2.17 Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS Fields:

op = 0x23, rs = 1, rt = 2, const = 0x4

I-format

op	rs	rt	Constant or address
6bits	5...	5..	16..

$op = 0x23_{hex} = 100011_2 = 35$ ⟹ lw

$rs = 1 = 00001_2 = \$at$

$rt = 2 = 00010_2 = \$v0$

$Const = 0x4_{hex} = 0000\ 0000\ 0000\ 0100_2 = 4$ = Memory value

Answer : I-format / lw $v0, 4($at)

$100\,11\ 0000\,1\ 0001\,0\ 0000\ 0000\ 0000\ 0100_2$

op rs rt Constant or address

2.16 Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS fields:

op = 0, rs = 3, rt = 2, rd = 3, Shamt = 0, funct = 34

R-format

op	rs	rt	rd	Shamt	funct
6bits	5..	5..	5..	5..	6..
000000_2	00011_2	00010_2	00011_2	00000_2	100010_2
op	rs	rt	rd	Shamt	funct

function value : 34 => Sub

rs, rd value : 3 => $V1

rt value : 2 => $V0

Shift amount : 0

Answer : R-format / Sub $V1, $V1, $V0

2.23 Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

```
        slt  $t2, $0, $t0
        bne  $t2, $0, ELSE
        j    DONE
ELSE:   addi $t2, $t2, 2
DONE:
```

slt : set on less than → if ($0 < $t0) ⇒ $t2 = 1;
else $t2 = 0;

bne : branch on not equal → if ($t2 != $0) ⇒ ELSE

j : jump → DONE

addi : add immediate → $t2 = $t2 + 2

$t0 > $0 ($t0 holds the value 0x00101000 (= 1052_{10}))

slt $t2, $0, $t0 ⇒ $t2 = 1

bne $t2, $t0, ELSE ⇒ go to ELSE

addi $t2, $t2, 2 ⇒ 1 + 2 = 3 = $t2

j DONE ⇒ end!

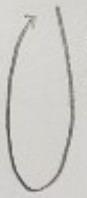
Answer: Value of $t2 = 3

2.26 Consider the following MIPS loop:

```
Loop: slt $t2, $0, $t1  # $t2=1 if ($0<$t1) else $t2=0
      beq $t2, $0, DONE # go DONE if ($t2==$0)
      subi $t1, $t1, 1  # $t1 = $t1-1
      addi $S2, $S2, 2  # $S2 = $S2+2
      j Loop            # Jump Loop
DONE:
```

1. Assume that the register $t1 is initialized to the value 10. What is the value in register $S2 assuming $S2 is initially zero?

$t1 = 10
slt → $t2=1
subi → $t1=9
addi → $S2=2



Loop 10 times (until $t1=0) =>

$t1=0 (10-1×10)
$S2=20 (2×10)

slt → $t2=0
beq → go DONE(end)

Answer: Value in register $S2 = 20

2. For each of the loops above, write the equivalent C code routine. Assume that the registers $S1, $S2, $t1, and $t2 are integers A, B, i, and temp, respectively.

$S1: A, $S2: B, $t1: i, $t2: temp

slt, beq are the conditions for the Loop exit.
subi, addi are the expressions calculated in Loop.

```
slt $t2, $0, $t1
beq $t2, $0, DONE
      :
j Loop
DONE:
```

=> While (i>0) {

```
i = i-1;
B = B+2;
}
```

Answer:

```
while(i>0)
{
   i-=1;
   B+=2;
}
```

3. For the loops written in MIPS assembly above, assume that the register $t1 is initialized to the value N. How MIPS instructions are executed?

# of iterations of the MIPS instructions in the loop state

⇒ $5 \times N$ ($t1)

slt, beq
Subi, addi, j
# of instructions

Needs slt & beq to exit the Loop
⇒ 2.

Answer: $5N+2$ times

2.27 Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the value of a, b, i and j are in registers $s0, $s1, $t0, and $t1, respectively. Also, assume that register $s2 holds the base address of the array D.

```
for (i=0; i<a; i++)
    for (j=0; j<b; j++)
        D[4*j] = i+j;
```

} ⇒ need 2 loop

a : $s0 i : $t0
b : $s1 j : $t1
array D's base address : $s2

Answer:

```
        move $t0, $0    # j = 0
        move $t1, $0    # j = 0
LOOP1:  slt  $t2, $t0, $s0  # if i≥a, t2=0
        beq  $t2, $0, EXIT1 # t2==0, jump to EXIT1
        j    LOOP2          # inner LOOP
LOOP2:  slt  $t3, $t1, $s1  # if j≥b, t3=0
        beq  $t3, $0, EXIT2 # t3==0, jump to EXIT2
        sll  $t4, $t1, 2    # t4 = j x 2^2
        add  $t5, $s2, $t4  # t5 = D + (j x 4)
        sll  $t5, $t5, 2    # t5 = {D + (j x 4)} x 2^2
        lw   $t6, 0($t5)    # t6 = D[4*j]
        add  $t7, $t0, $t1  # t7 = i + j
        sw   $t7, 0($t6)    # D[4*j] = i+j
        addi $t1, $t1, 1    # j++
        j LOOP2
EXIT2:  addi $t0, $t0, 1    # i++
        j LOOP1             # outter LOOP
EXIT1:                      # end
```