컴퓨터구조 과제1

문항 수: 5 2022년 1학기

0[배점 없음]. 학번 및 이름을 쓰시오.

학번: B617065 이름: 신성우

1. (교재 연습문제 2.1)

For the following C statement, what is the corresponding MIPS assembly code? Assume that the variables f, g, h, and i are given and could be considered 32-bit integers as declared in a C program. Use a minimal number of MIPS assembly instructions. Assume that the variables f, g, and h are assigned to registers $s0, $s1, and $s2, respectively.

```
f = g + (h - 5);
```

```
addi $s0, $s2, -5
add  $s0, $s1, $s0
```

2. (교재 연습문제 2.3)

For the following C statement, what is the corresponding MIPS assembly code? 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.

```
B[8] = A[i-j];
```

```
sub  $t0, $s3, $s4    # i - j
sll  $t1, $t0, 2      # (i-j) x 4
add  $t1  $s6, $t1    # A[(i-j) x 4]
lw   $t1, 0($s6)
sw   $t1, 32($s7)
```

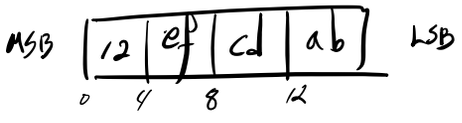
3. (교재 연습문제 2.4)
For the MIPS assembly instructions below, what is the corresponding C statement? Assume that the variables f, g, h, i, and j are assigned to registers $s0, $s1, $s2, $s3, and $s4, respectively. Assume that the base address of the arrays A and B are in registers $s6 and $s7, respectively.

```
sll  $t0, $s0, 2     # $t0 = f * 4
add  $t0, $s6, $t0   # $t0 = &A[f]
sll  $t1, $s1, 2     # $t1 = g * 4
add  $t1, $s7, $t1   # $t1 = &B[g]
lw   $s0, 0($t0)     # f = A[f]
addi $t2, $t0, 4     // $t2 = A[f+1]
lw   $t0, 0($t2)     // $t0 = A[f+1]
add  $t0, $t0, $s0   // $t0 = A[f+1] + A[f]
sw   $t0, 0($t1)     // B[g] = A[f+1] + A[f]
```

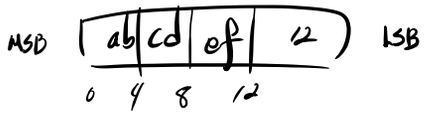
```
B[g] = A[f+1] + A[f];
```

4. (교재 연습문제 2.7)
Show how the value **0xabcdef12** would be arranged in memory of a little-endian and a big-endian machine. Assume the data is stored starting at address 0.

① little - endian



② big - endian



5. (교재 연습문제 2.29)

Translate the following loop into C. Assume that the C-level integer i is held in register $t1, $s2 holds the C-level integer called **result**, and $s0 holds the base address of the integer **MemArray**.

```
        addi $t1, $0, $0
LOOP:   lw   $s1, 0($s0)
        add  $s2, $s2, $s1
        addi $s0, $s0, 4
        addi $t1, $t1, 1
        slti $t2, $t1, 100
        bne  $t2, $0 , LOOP
```

```
for (i=0; i<100; i++)
{
    result t = MemArray[i];
}
```