Bubble Sort C-Code:

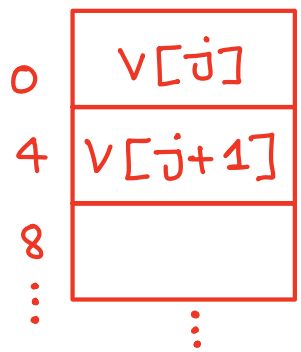
```
void sort(int v[], int n){
    int i,j;
    for (int i=0; i<n; i++){
        for(j=i-1; j>=0 && v[j+1]>v[j]; j--){
            swap(v, j);
        }
    }
}
void swap(int v[], int k){
    int temp;
    temp = v[k];
    v[k]=v[k+1];
    v[k]= temp;
}
```

Given, Base address of v in $a0, n in $a1, i in $s0, and j in $s1. temp is in $t0.

Bubble Sort MIPS-Code:

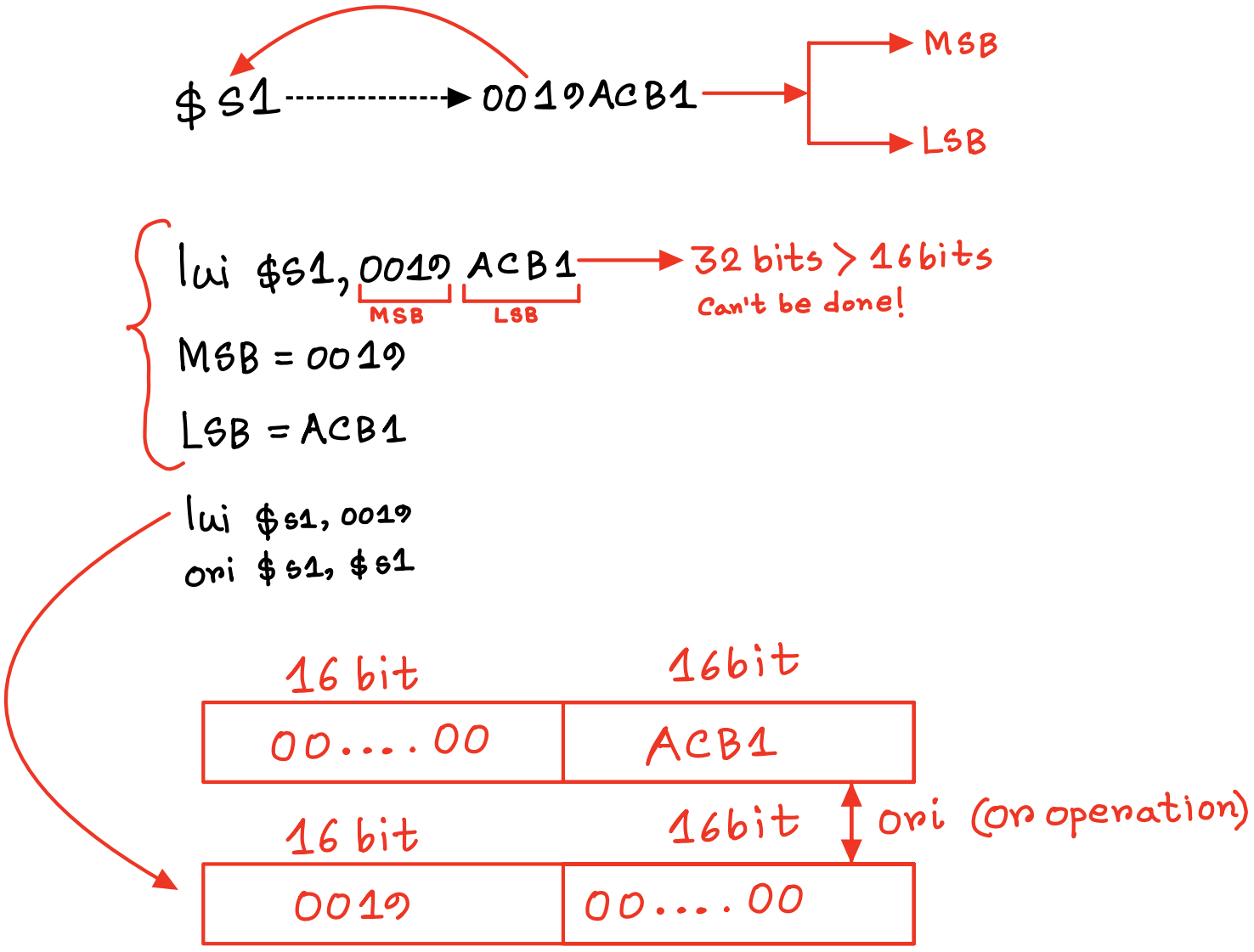
```
add $s2, $a0, $zero  ---> v
add $s3, $a1, $zero  ---> n
add $s0, $zero, $zero ---> i=0
Loop1:
slt $t0, $s0, $s3 ---> i>n
```

```
beq $t0, $zero, Loop1Exit
addi $s1, $s0, -1 ──────► j = i - 1
Loop2:
slti $t0, $s1, 0 ──────► j<0, $t0 = 1
bne $t0, $zero, Loop2Exit
sll $t1, $s1, 2 ──────► $t1 = 4 * j
add $t2, $s2, $t1 ──────► $t2 = 4 * j + Base Address of v[]
lw $t3, 0($t2) ──────► v[j]
lw $t4, 4($t2) ──────► v[j+1]
slt $t0, $t4, $t3 ──────► v[j+1] > v[j], $t0=0
beq $t0, $zero, Loop2Exit
add $a0, $s2, $zero
add $a1, $s1, $zero
jal swap
addi $s1, $s1, -1
j Loop2
Loop2Exit:
addi $s0, $s0, 1
j Loop1
Swap:
sll $t1, $a1, 2 ──────► k * 4
add $t1, $a0, $t1
lw $t0, 0($t1)
lw $t2, 4($t1)
sw $t2, 0($t1)
sw $t2, 4($t1)
jr $ra
```



k ──► a1

v ──► a0



Class 11 and 12: Solution of Quizes and Assignments Check the google drive links.