

## Homework Assignment 2

(Due 2:10pm, Oct. 5, email to [daehyun.kim@wsu.edu](mailto:daehyun.kim@wsu.edu) or submit a hardcopy)

You can use the following instructions only for this homework.

- Instructions
  - ADD R\$, R%, R&
  - ADD R\$, R%, #imm
  - SUB R\$, R%, R&
  - SUB R\$, R%, #imm
  - AND R\$, R%, R& // logical AND
  - AND R\$, R%, #imm
  - ORR R\$, R%, R& // logical OR
  - ORR R\$, R%, #imm
  - EOR R\$, R%, R& // logical XOR
  - EOR R\$, R%, #imm
  - CMP R\$, R%
  - CMP R\$, #imm
  - BGE, BLT, BGT, BLE, BEQ, BNE, B
  - MOV R\$, R% // R\$ = R%
  - MOV R\$, #imm

1. (30 points) Write an assembly code for the following C code.

```
int a, b, c;

switch ( a ) {
    case 0: b++; break;
    case 1: b--; break;
    case 2: c++; break;
    default: b = 0; c = 1; break;
}
```

- Assume that a is in R0, b is in R1, and c is in R2.
- The exit point (the end of the code) could be just an address label.

```
CMP R0, #0
BNE check_case1
ADD R1, R1, #1
B switch_end
check_case1:
CMP R0, #1
BNE check_case2
SUB R1, R1, #1
```

```

    B switch_end
check_case2:
    CMP R0, #2
    BNE check_case3
    ADD R2, R2, #1
    B switch_end
default_case:
    MOV R1, #0
    MOV R2, #1
switch_end:

```

2. (40 points) Write an assembly code for the following C code.

```

int a, b, c;

for ( a = 0 ; (a / 4) < 10 ; a = a + 2 ) {
    b++;

    if ( b < c )
        b++;
    else
        c++;

    if ( (c % 4) == 1 )
        break;
}

```

- Assume that a is in R0, b is in R1, and c is in R2.
- The exit point (the end of the code) could be just an address label.

```

MOV R0, #0 // a = 0
check_for:
MOV R4, R0, LSR #2 // R4 = a/4
CMP R0, #10
BGE end_for
ADD R1, R1, #1 // b++
CMP R1, R2
BGE if_else
ADD R1, R1, #1 // b++
B next_if
if_else:
    ADD R2, R2, #1 // c++
next_if:

```

```

AND R4, R4, #0x03 // c % 4
CMP R4, #1
BEQ end_for
ADD R0, R0, #2 // a = a + 2
B check_for
end_for:

```

3. (50 points) Write an assembly code for the following C code.

```

int a, b, c;

a = 1;
b = 2;
c = 3;

while ( a < 10 ) {
    while ( b < 20 ) {
        if ( c < 30 )
            c++;
        else
            c += 2;

        b++;
    }
    a++;
}

```

- Assume that a is in R0, b is in R1, c is in R2, and n is in R3.
- The exit point (the end of the code) could be just an address label.

```

MOV R0, #1 // a = 1
MOV R1, #2 // b = 2
MOV R2, #3 // c = 3
check_while1:
    CMP R0, #10
    BGE end_while1
check_while2:
    CMP R1, #20
    BGE end_while2
check_if:
    CMP R2, #30
    BLT exe_if
    ADD R2, R2, #2 // c += 2
    B end_if
exe_if:

```

```
    ADD R2, R2, #1 // c++
end_if:
    ADD R1, R1, #1 // b++
    B check_while2
end_while2:
    ADD R0, R0, #1 // a++;
    B check_while1
end_while1:
```