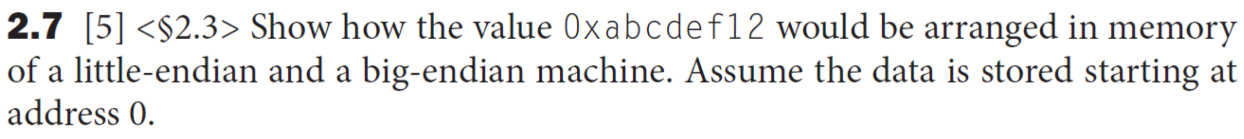
ECE 485/585 – Computer Organization and Design

HOMEWORK #2

Due date: Friday, September 23rd 2022, 11:59PM

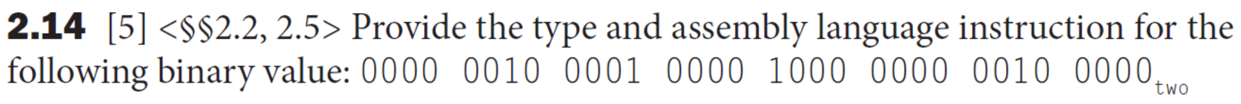
Solve the following exercises from the textbook (Chapter 2)

1. Exercise 2.7



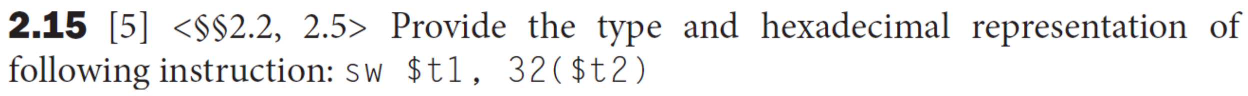
2.7 [5] <§2.3> Show how the value `0xabcdef12` would be arranged in memory of a little-endian and a big-endian machine. Assume the data is stored starting at address 0.

2. Exercise 2.14



2.14 [5] <§§2.2, 2.5> Provide the type and assembly language instruction for the following binary value: `0000 0010 0001 0000 1000 0000 0010 0000`$_{\text{two}}$

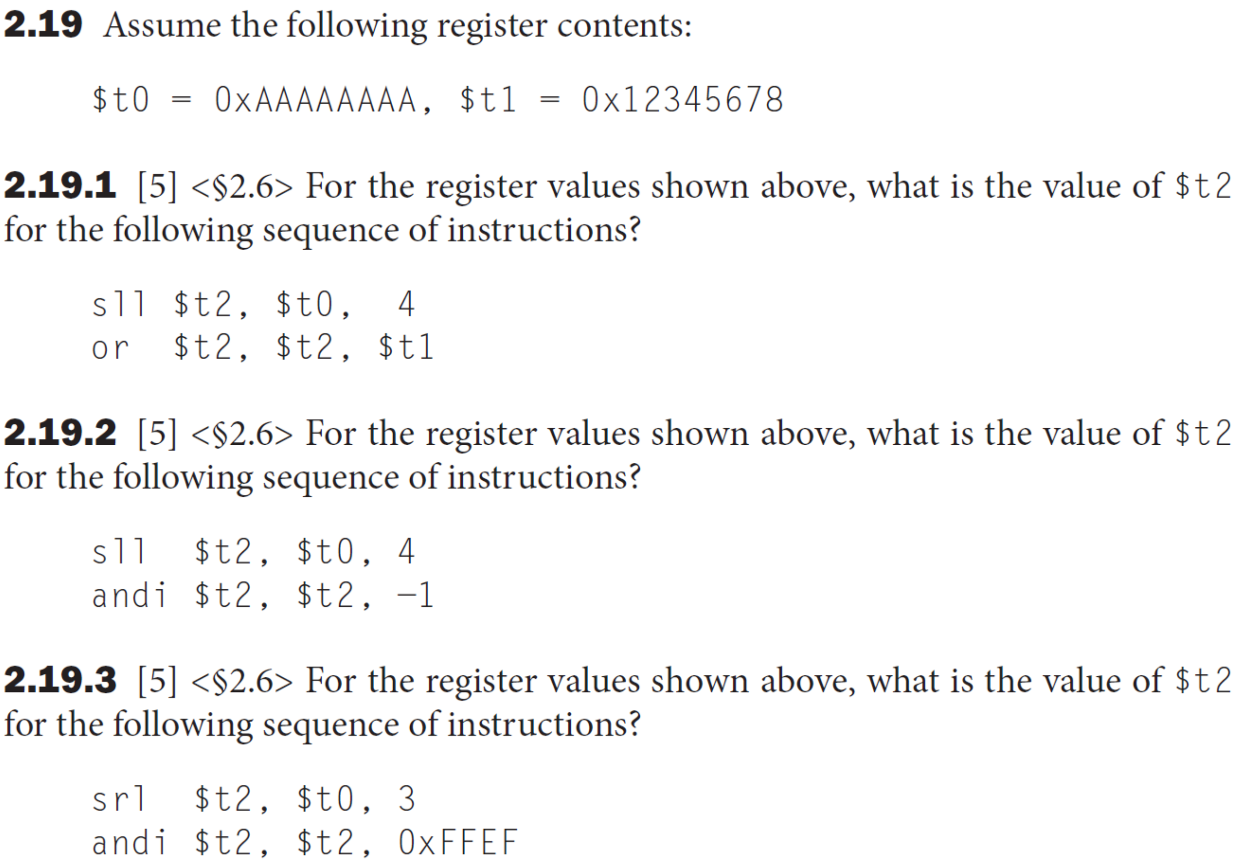
3. Exercise 2.15



2.15 [5] <§§2.2, 2.5> Provide the type and hexadecimal representation of following instruction: `sw $t1, 32($t2)`

(Continued on the next page)

4. Exercise 2.19



2.19 Assume the following register contents:

```
$t0 = 0xAAAAAAAA, $t1 = 0x12345678
```

2.19.1 [5] <§2.6> For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
sll $t2, $t0,  4
or  $t2, $t2, $t1
```

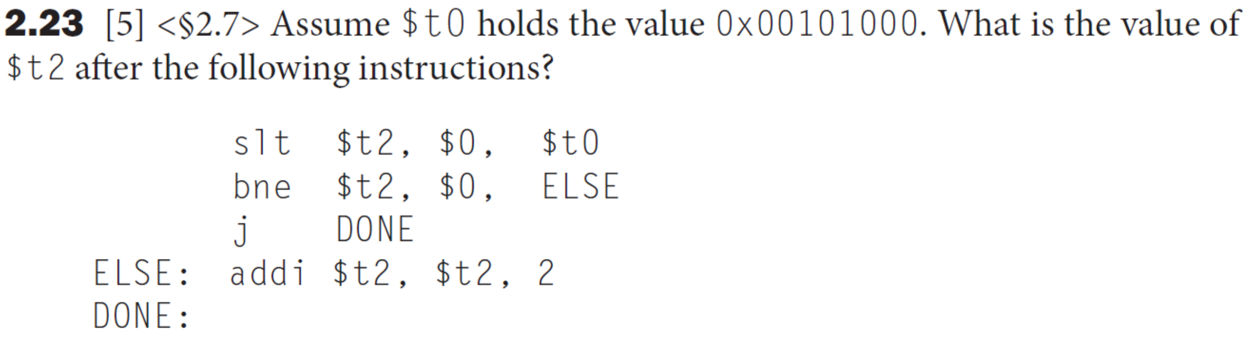
2.19.2 [5] <§2.6> For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
sll  $t2, $t0, 4
andi $t2, $t2, −1
```

2.19.3 [5] <§2.6> For the register values shown above, what is the value of $t2 for the following sequence of instructions?

```
srl  $t2, $t0, 3
andi $t2, $t2, 0xFFEF
```

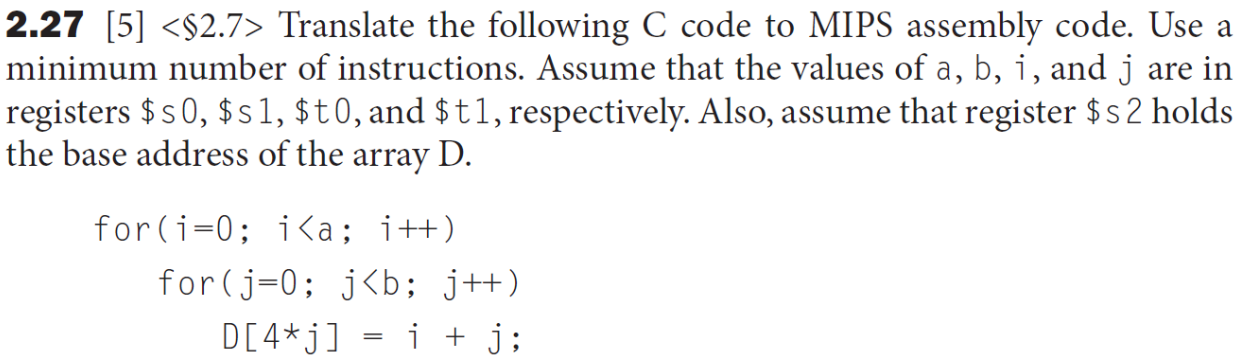
5. Exercise 2.23



2.23 [5] <§2.7> 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:
```

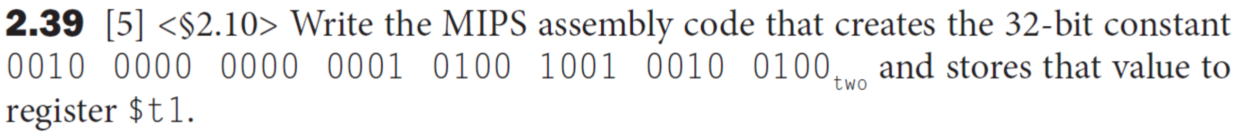
6. Exercise 2.27



2.27 [5] <§2.7> Translate the following C code to MIPS assembly code. Use a minimum number of instructions. Assume that the values 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.

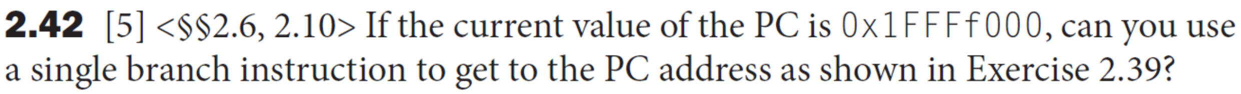
```
for(i=0; i<a; i++)
   for(j=0; j<b; j++)
      D[4*j] = i + j;
```

7. Exercise 2.39



2.39 [5] <§2.10> Write the MIPS assembly code that creates the 32-bit constant `0010 0000 0000 0001 0100 1001 0010 0100`$_{two}$ and stores that value to register `$t1`.

8. Exercise 2.42



2.42 [5] <§§2.6, 2.10> If the current value of the PC is `0x1FFFf000`, can you use a single branch instruction to get to the PC address as shown in Exercise 2.39?