Binary system operations and representation of negative numbers.

objective:

1. Binary system operations.
2. Representation of negative numbers
3. Two's Complement Addition and subtraction
4. One's Complement Addition and subtraction.

① Binary system operations:

Ⓐ Binary Addition:

example:

```
   1 1 1 0 1     ← Carries.
   1 1 1 1 0 1
 + 1 0 1 1 0 1
 -------------
 1 1 0 1 0 1 0
```

addition table

	sum	carry
0 + 0 =	0	0
0 + 1 =	1	0
1 + 0 =	1	0
1 + 1 =	0	1

Ⓑ Binary subtraction:

B → borrow

```
                    0 0 1 1 1 1 1 0 0
first number   X    229      1 1 1 0 0 1 0 1
second number  Y   - 46    - 0 0 1 0 1 1 1 0
                   ----    -----------------
             X-Y =  183      1 0 1 1 0 1 1 1
```

subtraction table.

0 − 0 = 0

1 − 0 = 1

1 − 1 = 0

0 − 1 = 1 and the borrow = 1

After the first borrow, the new subtraction for this column is 0−1, so we must borrow again

must borrow 1, yeilding the new subtraction 10 − 1 = 1

```
                 0 10  1 1 10 10
X     229        1 1 1 0 0 1 0 1
Y    - 46      - 0 0 1 0 1 1 1 0
     ----      -----------------
X-Y   183                    1 1
```

The borrow goes through three columns to reach a borrowable 1.

100 = 011 (the modified bits) + 1 (the borrow)

© Binary multiplication :

example :

```
          1 0 1 1 1
       x    1 0 1 0
       -------------
          0 0 0 0 0
        1 0 1 1 1
+     0 0 0 0 0
    1 0 1 1 1
  -------------------
  1 1 1 0 0 1 1 0 (result)
```

multiplication table

0	×	0	=	0
0	×	1	=	0
1	×	0	=	0
1	×	1	=	1

② Representation of negative numbers

There are many ways to represent negative numbers

1. Signed - magnitude system
2. Complement number systems.

- **Signed - Magnitude Representation :**

in Signed- magnitude system, the number consists of a magnitude and a symbol indicating whether the magnitude is positive or negative.

in binary system : extra bit position to represent the sign (sign bit) : (MSB) is used.

Sign bit : → 0 = plus
→ 1 = minus

examples :

$\underset{\text{sign bit}}{0}\,\underbrace{1010101}_{\text{magnitude}}{}_2 = +85_{10}$, $11010101_2 = -85_{10}$

$01111111_2 = +127_{10}$ $11111111_2 = -127_{10}$

$0000\,0000_2 = +0_{10}$ $1000\,0000_2 = -0_{10}$

- **Complement Number System :**

Complement number system negates a number by talking its complement as defined by the system. There are two complement number systems that can be used : Two's complement system and One's complement system.

Two's complement system:

The complement of an n-digit number is obtained by:

Subtracting the number from r^n.

r- the base of the system.

In decimal system it's called the 10's complement. For binary numbers, it's called two's complement, the MSB of a number in this system is used as the sign bit.

examples:

① $17_{10} = 00010001_2$

⇓ Complement bits

$$\begin{array}{r} 11101110 \\ +\ 1 \\ \hline 11101111 \end{array} = -17_{10}$$

② $-99_{10} = 10011101_2$

⇓

Complement bits

$$\begin{array}{r} 01100010 \\ 1\ + \\ \hline 01100011_2 \end{array} = 99_{10}$$

$0_{10} = 0000\ 0000_2$

⇓

$$\begin{array}{r} 1111\ 1111 \\ 1\ + \\ \hline \times\ (1)\ 0000\ 0000_2 \end{array} = 0_{10}$$

for "0" → One representation

One's Complement Representation.

in One's complement representation the complement of an n-digit number D is obtained by

Subtracting the number from $r^n - 1$.

this can be accomplished by complementing the individual digits of D, without adding 1 as in Two's complement systems.

in decimal System it's called 9's complement and in binary system it's called one's complement in one's complement :

- MSB is used as sign digit.
- for 0 → there are two representations :

00000000 → positive zero
11111111 → negative zero.

examples : ⇒

① $17_{10} = 00010001_2$

⇓

$11101110_2 = -17_{10}$

② $-99_{10} = 10011100_2$

⇓

$01100011_2 = 99_{10}$

0_{10} = 0000 0000 (positive zero)

⇓

1111 1111 (negative zero)

3. Two's Complement Addition and Subtraction

examples :

a- two positive number adding : +15 +27 .

```
  0  0000 1111
  0  0001 1011
 --------------
  0  0010 1010 = (42)10 → (okay)
```

sign bit

b- two positive number adding with overflow :

+78 + 85 = $(163)_{10}$ → We need 8 bits + 1 for sign bit

```
   01001110
   01010101
  ----------
  [1]0100011
```

sign? or overflow?

example 3: 43 - 25 :⇒

```
     0 0 1 0 1 0 1 1
     1 1 1 0 0 1 1 1
     ---------------
(1)  0 0 0 1 0 0 1 0
ignore
```

example 4: 25 - 43 ⇒

```
  0  0 0 1 1  0 0 1
  1  1 0 1 0  1 0 1
  -----------------
  1  1 1 0 1  1 1 0
```

example 5: -15 - 27 :

```
         1  1 1 1 0 0 0  1
         1  1 1 0 0 1 0  1
         -----------------
ignore (1)  1  1 0 1 0 1 1  0
```

4. One's complement addition and subtraction

Note: the same as Two's complement addition and Subtraction, except that the carry value must be added to the result in LSB bit.

example1: 43 - 25 :

```
           0 0 1 0 1 0 1 1
           1 1 1 0 0 1 1 0
           ---------------
carry ← 1  0 0 0 1 0 0 0 1
                         1   (carry added)
           ---------------
           0 0 0 1 0 0 1 0
```

example 2: 25 - 43

```
0  0 0 1 1  0 0 1
1  1 0 1 0  1 0 0
-----------------
1  1 1 0 1  1 0 1
```