& | ! ^

bitwise operation

a = 1 1 0 1

b = 1 0 1

a & b =

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

2's complement Method

Container Size → 8 bit

a = 5

-5

-5 = (!a) + 1

a = 00000101 !a = 11111010

$!a+1 = 11111011 = -5$

$!a+1 = 00000100+1$

$= 00000101$

$!a$ → 1's complement of a

$!a+1$ → 2's complement of a.

$(5)f(-3)$

00000101 $\quad 00000011 = 3$

$11111101 = -3$

f

00000101

11111101 → ? -3

overflow ← 100000010

result = 0 0 0 0 0 0 0 | 0

2^1 2^0

MSB = 1 → negative

1 2 3

$3 \times 10^0 = 3$

$2 \times 10^1 = 20$

$1 \times 10^2 = 100$

MSB = 1 ⟶ negative else positive

a = [1 1 1 1 1 1 0] = -3

|a+1| = 0 0 0 0 0 0 1 1 = 3

Addition → -a + → Conversion

Number System

Base

10 2 8 16

Unsigned Number

Signed Number

Signed Bit Method

2's Complement Method

5 + (-3)

Sign

5 - 3

Subtraction

5 + 3

addition

2's Complement

5 + (-3)

addition

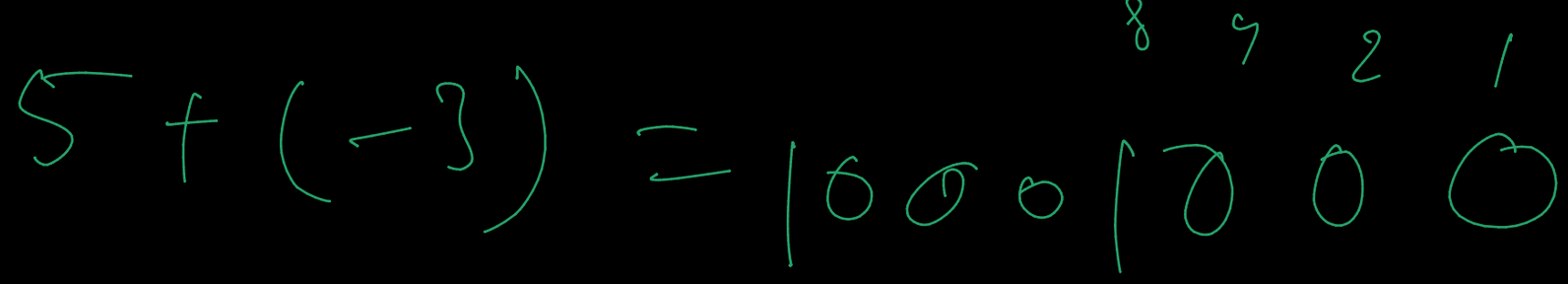
5 + 3

addition

Signed Bit

$5 = 00000101$

$-3 = 10000011$



$5 + (-3) = 10001000$

$= -8$

2's complement

$5 = 00000101$

$3 = 00000011$

$-3 = 11111101$

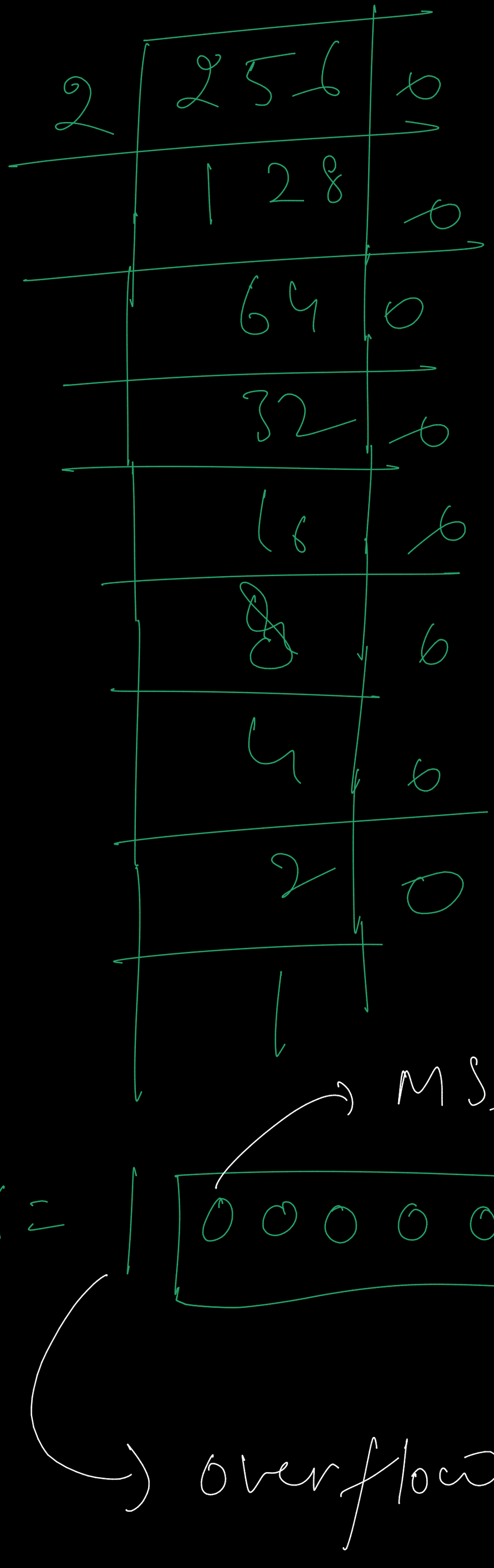
$5 + (-3) = 00000101$

$+\ 11111101$

100000010

00000010 → 2

1 byte → 256



	Size	Limit
char	1 byte	−128 - 127
{ bool	1 byte	0 / 1
short	2 byte	
int	4 byte	
long	4 byte - 8 byte	
{ long long long long int	8 byte	

unsigned char }
unsigned short
unsigned int
unsigned long
unsigned long long
unsigned long long int

%c

character print

%c 65

%d 65

%b 65

binary

%o

octal

%x → hexadecimal

65

%d → important

255

float → 4 byte

double → 8 byte

523.34

5.2334 x 10^2

$1\overline{10.011}$ $\overline{2}$ → exponent

1.10011×2 → mantissa

0.0011

1.1×2^{-3}