Bitwise Manipulation:

	128	64	32	16	8	4	2	1	Worth
205									Byte
	7	6	5	4	3	2	1	0	Position

AND To reset (Set a bit to zero)
OR To set (Set a bit to one)
XOR To inverse (Set to 1 if it is 0, to 0 if it is 1)

AND 0 reset , 1 no change
OR 0 no change , 1 set
XOR 0 no change, 1 inverse.
AND 1 to check, what is in, the bit.

SHIFTS (Binary Shifting).

Logical Shift:

LSL Left Shift

- It multiplies numbers by 2.
- Only until result is in range.

	128	64	32	16	8	4	2	1	
drop ←	0	1	1	0	0	1	0	0	100
	1	1	0	0	1	0	0	0 ← Zero moves in.	200

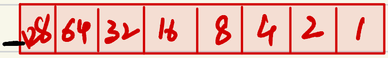
LSR Right Shift

- It is integer value so 0.5 can't be held and will be lost.

	128	64	32	16	8	4	2	1		
	0	0	0	0	1	1	1	0	14 Even	
→	0	0	0	0	0	1	1	1 →	7 Odd	7/2 = 3.5
	0	0	0	0	0	0	1	1	3	

Arithmetic Shifts:

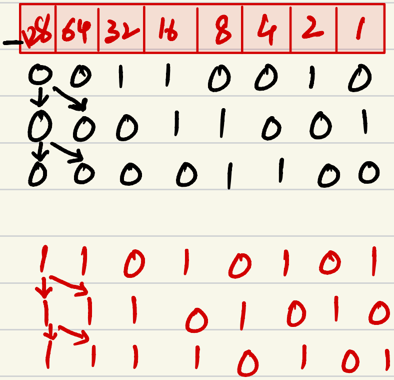
Left:



128 | 64 | 32 | 16 | 8 | 4 | 2 | 1

Similar to Logical Left Shift

Right:



128 | 64 | 32 | 16 | 8 | 4 | 2 | 1

+ve

0 0 1 1 0 0 1 0

0 0 0 1 1 0 0 1

0 0 0 0 1 1 0 0

-ve

1 1 0 1 0 1 0 1

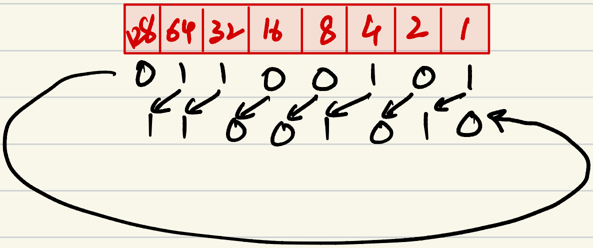
1 1 1 0 1 0 1 0

1 1 1 1 0 1 0 1

- Mainly used to signed numbers.
- It shifts to the right but leftmost is retained

Cyclic Shift

Left

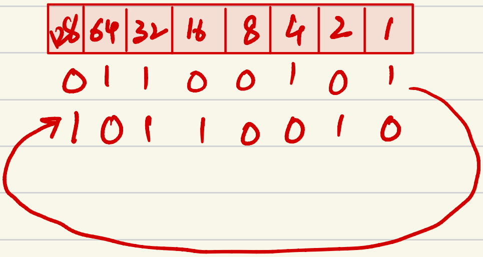


128 | 64 | 32 | 16 | 8 | 4 | 2 | 1

0 1 1 0 0 1 0 1

1 1 0 0 1 0 1 0

Right



128 | 64 | 32 | 16 | 8 | 4 | 2 | 1

0 1 1 0 0 1 0 1

1 0 1 1 0 0 1 0