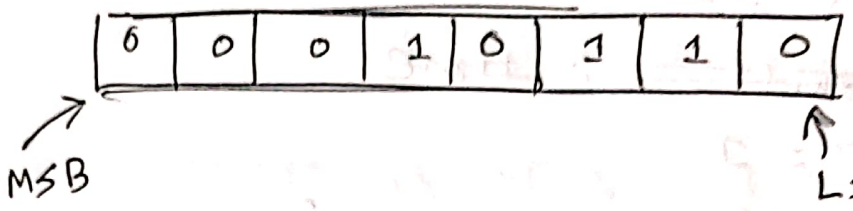
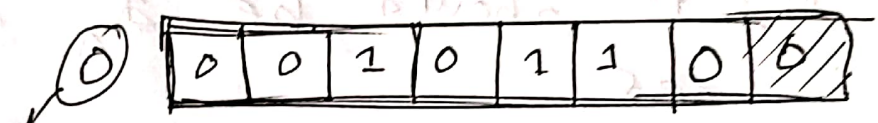
Binary Left and Right Shift

N = 22



← Left shift ⇒ N = N << 1 ⇒ N = N * (2^1)



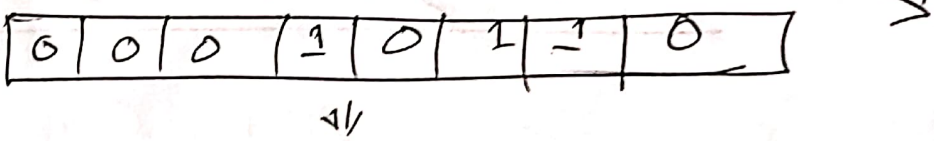
← Last bit will be filled by 0.

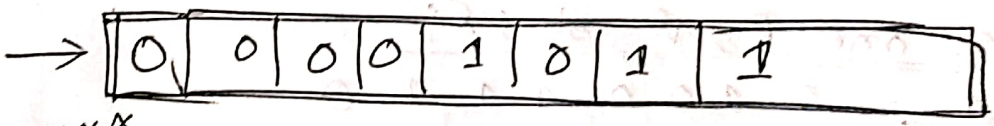
***

Left shift :-

N = N << i ⇒ N = N * (2^i)

Right shift :-





***

N = N >> i ⇒ N = N / (2^i)

check if i-th bit set or not

N = 12

0 0 0 0 1 1 0 0

~~;) Left shift i time~~

1. take F = 1

then Left shift i-time

2. Then N & F should be non-zero if i-th bit is set.

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

Counting number of set bits

N = 0 1 0 0 1 1 1 0

Approach one: Right shift :-

1. N & 1 =

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

2. Right shift if non-zero

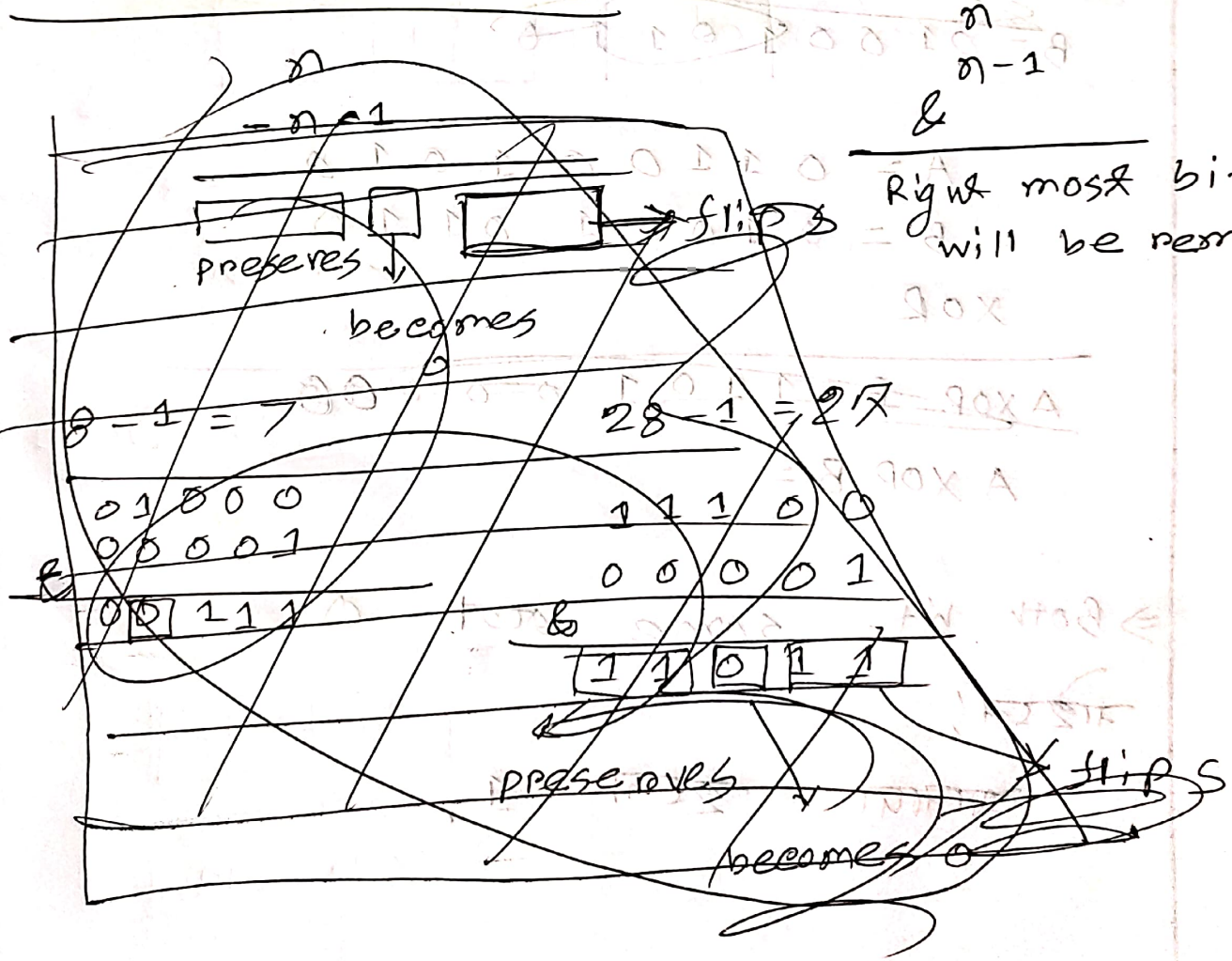
N & 1 = 0 0 1 0 0 1 1 1

0 0 0 0 0 0 0 0 1

0 0 0 0 0 0 0 0 1 → increment counter.

⇒ Complexity O(lg n)

Another approach:-



If a number is power of two then it will have only one set bit

XOR and It properties

A = 0 1 1 0 0 1 0 1

B = 0 1 0 0 1 0 1 0

A = 0 1 1 0 0 1 0 1 0

B = 0 1 0 0 1 0 1 1 0

XOR

A XOR B = 1 1 0 1

A XOR B =

⇒ Both bit same হলে 0

নাহলে,

আলাদা হলে 1.

A XOR B = A ^ B

Properties:-

1. Identity element:-

0 ^ A = A

~~A = 0~~

2. A ^ A = 0

3. ordering of XOR does not matter.