

→ Binary Numbers

33

$\rightarrow 0 \rightarrow 0$   
 $1 \rightarrow 1$   
 $2 \rightarrow 10$   
 $3 \rightarrow 11$   
 $4 \rightarrow 100$   
 $5 \rightarrow 101$   
 $6 \rightarrow 110$   
 $7 \rightarrow 111$

Here, we can observe  
 that last bit of every  
 even no. is 0.  
 $\&$  last bit of every  
 odd no. is 1.

		AND (4)	OR (1)	XOR (1)
$\rightarrow$	0 0	0	0	0
	0 1	0	1	1
	1 0	0	1	1
	1 1	1	1	0

$\rightarrow$  NOT (1)  $\Rightarrow 1 \rightarrow 0$   
 $0 \rightarrow 1$

$\rightarrow$  Left Shift (L.S) ( $\ll$ )  
 e.g:

①  $3 \ll 2 \rightarrow$  (Add 2 0's in the end  
 of binary of 3)  
 $11 \rightarrow 1100$

②  $3 \ll 1$   
 $\downarrow$   
 $11 \rightarrow 110$

$\rightarrow$  Right Shift (R.S) ( $\gg$ )  
 e.g:

①  $3 \gg 1 \rightarrow$  (Remove 1 element from the  
 end of binary of 3)  
 $11 \rightarrow 1$

②  $6 >> 1$

↓

110 → 11

→ Let's say we have a datatype of 4 bits.

0/1 0/1 0/1 0/1

$2 \times 2 \times 2 \times 2$

$\Rightarrow 2^4 \Rightarrow 16$

e.g. 0 0 0 0 → 0

0 0 0 1 → 1

0 0 1 0 → 2

0 0 1 1 → 3

⋮  
⊙

1 1 1 1 → 15

So, in every bit we can either place

1/0. So, the total possibility is

$2 \times 2 \times 2 \times 2$

$\Rightarrow 2^4 \Rightarrow 16$

(It means we can store 16 numbers in a 4 bit binary system)

So, the maximum no. we can produce using 4 bits is  $2^4 - 1 \Rightarrow 15$ .

NOTE: As, our int datatype in C/C++ is of 32 bits. So, its maximum value will be  $2^{32} - 1$ .