

Bit Manipulation

0	→	0
1	→	1
2	→	10
3	→	11
4	→	100
5	→	101
⋮		

→ Binary

⋮
→ Decimal

* 1 bit → 2 values (0, 1)

2 bit → 4 values (0, 1, 2, 3)

3 bits → 0 → $2^3 - 1$ (0-7)

4 bits → 0 → $2^4 - 1$ (0-15)

⋮
⋮
⋮

* How to store negative in memory

1's complement

2's complement

This is best

E.g. $\rightarrow -5$

$+5 \rightarrow$ 0 0 0 0 0 1 0 1

\Downarrow

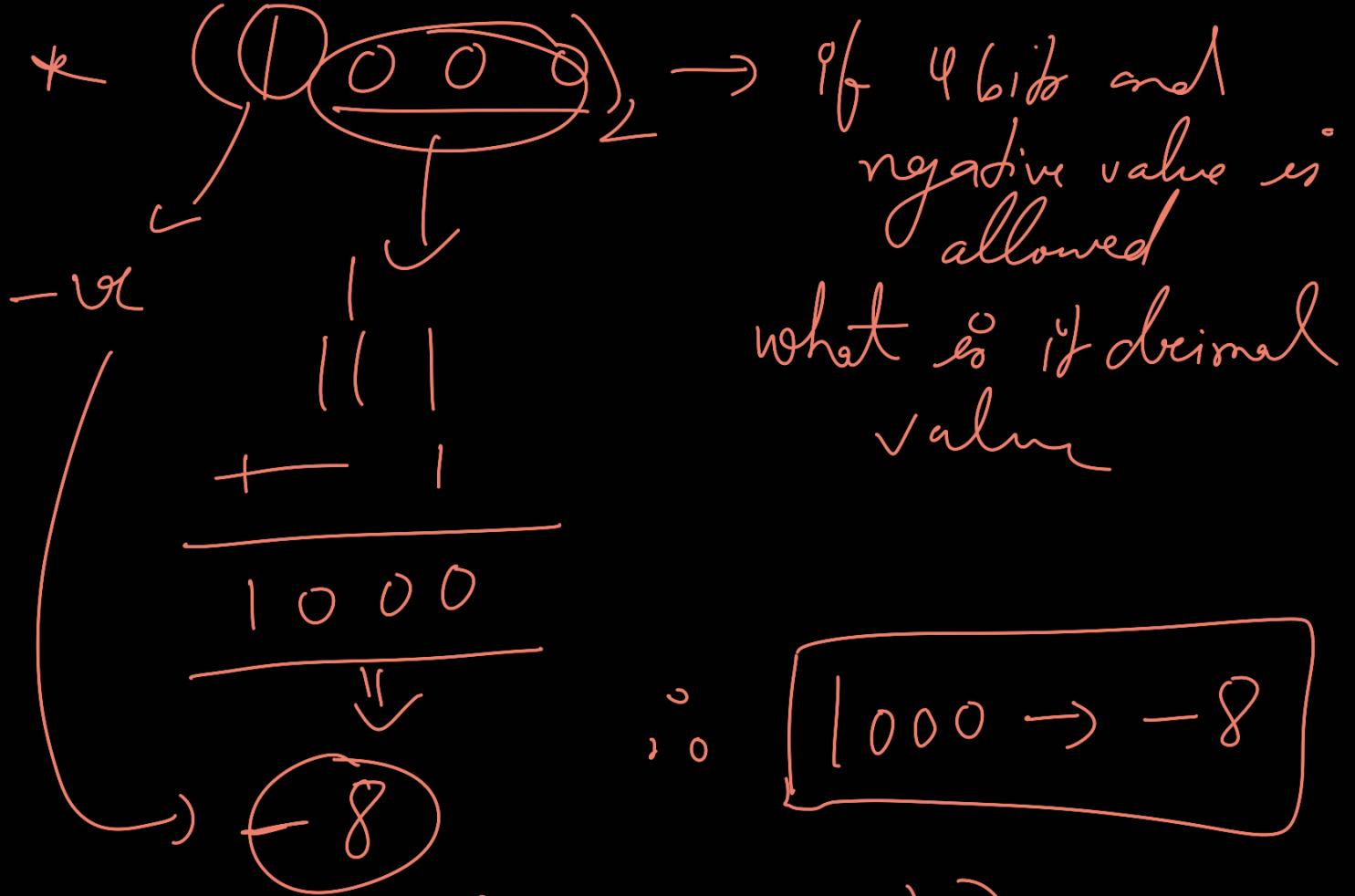
1 1 1 1 1 0 1 0

+

1 1 1 1 1 0 1 1

$-5 \leftarrow$

2's complement



* $(-2^{(n-1)} \rightarrow +2^{(n-1)} - 1)$

\hookrightarrow Range of 2's complement
 \hookrightarrow 1 bit used for storing

Sign

$1 \rightarrow OR$

$\gg \rightarrow$ Right shift

$\& \rightarrow AND$

$\ggg \rightarrow$ triple right shift

$\wedge \rightarrow XOR$

$\sim \rightarrow$ 1's complement

$\ll \rightarrow$ left shift

0	0	\rightarrow 0
0	1	\rightarrow 1
1	0	\rightarrow 1
1	1	\rightarrow 1

$\rightarrow OR$

0	0	\rightarrow 0
0	1	\rightarrow 0
1	0	\rightarrow 0
1	1	\rightarrow 1

$\rightarrow AND$

0	0	0
0	1	1
1	0	1
1	1	0

→ XOR

$0 \rightarrow 1$
 $1 \rightarrow 0$

→ ~ (complement)