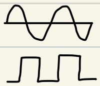
L1 + L3

Digital v/s Analog



- analog quantities vary continuously where as digital quantities vary in steps

- analog quantities can be reped in a digital format by sampling and quantisation

sampling → finding values of the analog quantity periodically [f(t)]

quantisation → a many-to-one mapping of the sampled data

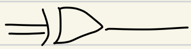
quantisation also results in a loss of precision

- digital representation of analog quantities results in a loss of precision
but digital techniques are more accurate and precise?

— x —

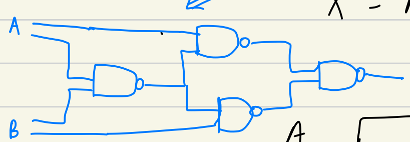
L3. exclusive OR / XOR

$A \oplus B \oplus C = (A \oplus B) \oplus C$
$= A \oplus (B \oplus C)$



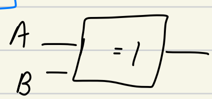
$\Rightarrow$ / $X = AB' + A'B$

$X = A \oplus B$



A	B	X
0	0	0
0	1	1
1	0	1
$\boxed{1}$	$\boxed{1}$	$\boxed{0}$

→ diff from OR



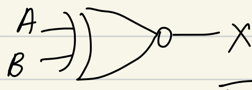
IEEE symbol

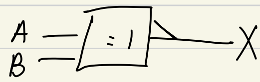
if XOR has multiple inputs, it is an odd-f^n generator
$\Rightarrow$ O/P = 1 if no of 1 I/P is odd.

A	B	$(A\oplus B)'$	$(A\oplus B)$	B'	$(A\oplus B')'$
0	0	1	0	1	0
0	1	0	1	0	1
1	0	0	1	1	1
1	1	1	0	0	0

exclusive - NOR / XNOR

$$X = AB + A'B'$$
$$X = (A \oplus B)'$$





IEEE symbol

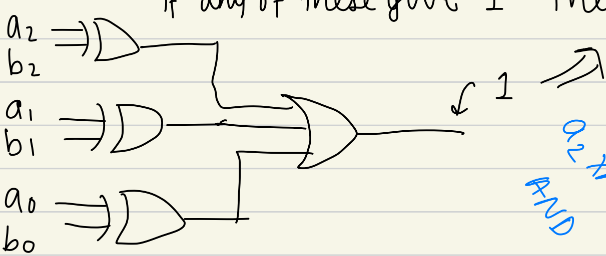
$$(A'B + B'A)' = (A + B')(B + A') = AB + B'A'$$

application of XOR

- bit-wise comparator ← parity detector
  - O/P = 1 if 2 multi-bit inputs are diff

let $a \Rightarrow a_2 a_1 a_0$ (3 bit)
$b \Rightarrow b_2 b_1 b_0$

if any of these give 1 then $a \neq b$



here we use $a_2 \oplus b_2$ OR $a_1 \oplus b_1$ OR $a_0 \oplus b_0$

but we can use a_2 XNOR b_2 AND a_1 XNOR b_1 AND a_0 XNOR b_0

- NAND identifies 0
- OR identifies 1
- AND identifies all 1s
- NOR identifies all 0s
- A (OR) 1 → forces value to be 1 "setting the bit"
- A (AND) 0 → forces value to be 0 "clearing the bit"
- XOR can be used as a parity detector. odd 1 → 1
- $A \oplus 1$ → changes parity
  aka
  "complent the bit"

A	1	$A \oplus 1$
0	1	1
1	1	0

↗ inverse

- "masking" the mask determines what of the original value is changed and kept unchange

eg. $x = 10110$ we want to clear x_2 & x_1. perform bitwise operation
AND
↳ 11001 → "mask"

x AND 11001 → 10000 x is unchanged wherever the mask is 1 and its 0 wherever the mask is 0

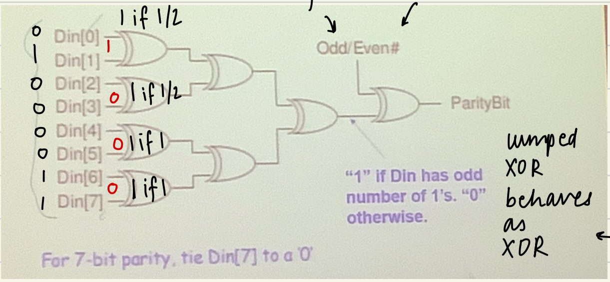
- $X = a \oplus \{ b \oplus (c \oplus (d \oplus e))$

  $d'e + e'd$

  $a'bcde + ab'cde + abcde' + abc'de + abcd'e$

- XOR as parity generator

  I/P depends on parity ← did not understand.



← v.v.v. imp.

if i am checking for odd parity
i will I/P = 1 so that O/P = 0 ⇒ parity bit is zero
& O/P XOR = 1
& O/P XOR = 0 ⇒ even no of ones
O/P = 1 ⇒ " is one

else if checking for even
I/P = 0 so that O/P = 1 ⇒ " is one
& O/P XOR = 1
& O/P XOR = 0 ⇒ odd no of ones.
O/P = 0 ⇒ " is zero