P3360 / AEP 3630

Lecture 24

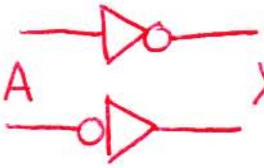
Boolean operations & gates

All digital operations can be reduced to operations on "0"'s and "1"'s = boolean operations

Gates = basic building blocks that perform simple boolean operations, e.g. AND, OR, NOT, etc.

- gates can still be quite complicated inside (transistors, diodes, caps, etc.)
- no need to worry about their internals
  ⇒ higher level abstraction
- see spec. sheets for pinout, etc.

NOT (compliment, inverter)



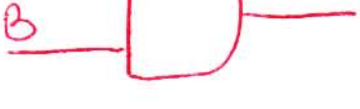
A

$X = \overline{A}$

truth table

A	X
1	0
0	1

AND binary or more inputs



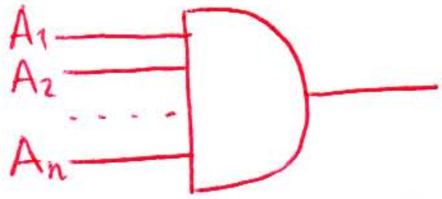
A

B

$X = A \cdot B$

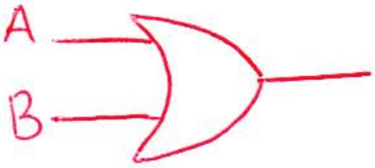
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

Multiple inputs AND



$X = A_1 \cdot A_2 \cdot \ldots \cdot A_n \iff$ TRUE iff <u>all</u> inputs TRUE

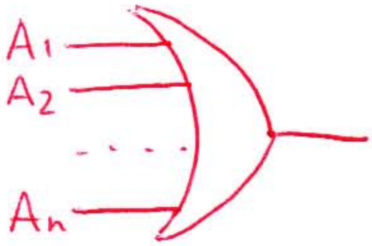
OR



$X = A + B$

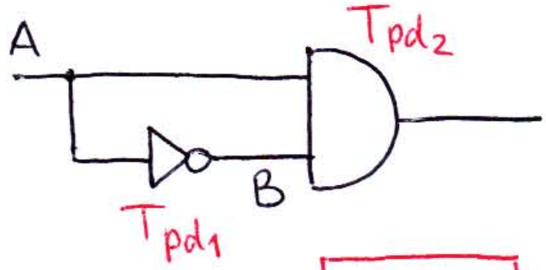
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

Multiple inputs OR



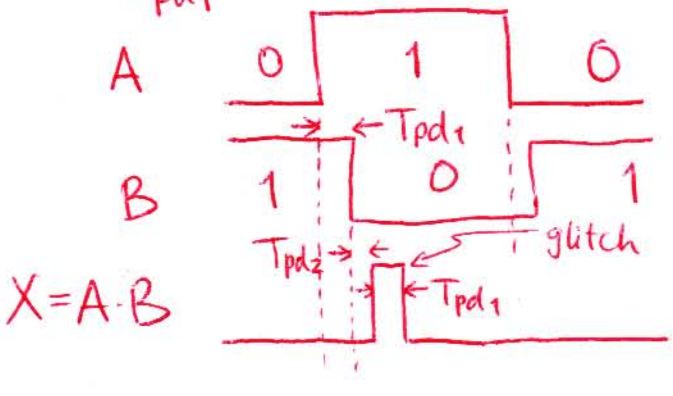
$X = A_1 + A_2 + \ldots + A_n \iff$ TRUE iff <u>any</u> input TRUE

Application of time dependent signal



$X = A \cdot B = A \cdot \bar{A} = 0$ always?

No, b/c of propagation delay



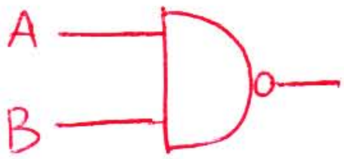
Refer to LTspice example simple_digital.asc

- any complicated fcn can be implemented using the basic gates
- IC's usually come with several copies of gates, e.g. quad 2-in AND's on a single chip
- AND, OR, NOT are not very efficient b/c typically one needs to reuse (mix) different IC's with spare/unused gates

Universal gates

- single type allows to implement arbitrary fcn
- more efficient (fewer chips) ⇒ more popular
- two basic types: NAND and NOR

NAND



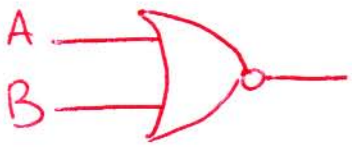
$X = \overline{A \cdot B}$

$\neq \overline{A} \cdot \overline{B}$

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\overline{A} \cdot \overline{B}$
0	0	1
0	1	0
1	0	0
1	1	0

NOR



$X = \overline{A + B}$

$\neq \overline{A} + \overline{B}$

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

A	B	$\overline{A} + \overline{B}$
0	0	1
0	1	1
1	0	1
1	1	0

De Morgan Theorem

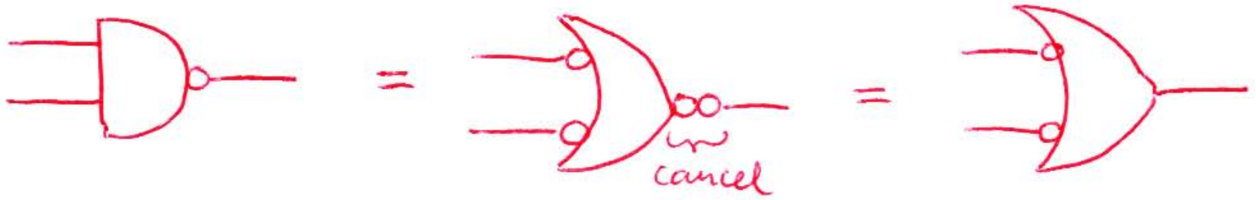
$\overline{A+B} = \bar{A}\cdot\bar{B}$ or $\overline{A\cdot B} = \bar{A}+\bar{B}$

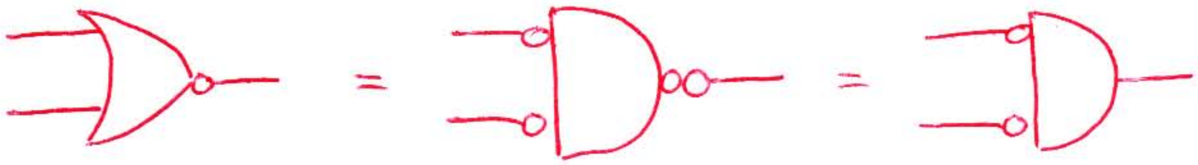
Alternatively $\overline{\bar{A}\cdot\bar{B}} = A+B$ or $\overline{\bar{A}+\bar{B}} = A\cdot B$

two steps to transform digital circuit into an equivalent

1) swap AND with OR and vice versa
2) invert all inputs and outputs

Examples





– use to convert one gate type into the other

Rules of Boolean algebra

$A\cdot\bar{A} = 0$	$A+\bar{A} = 1$	$A\cdot B = B\cdot A$	$\bar{\bar{A}} = A$
$A\cdot 0 = 0$	$A+0 = A$	$A+B = B+A$	$A(B+C) = A\cdot B + A\cdot C$
$A\cdot 1 = A$	$A+1 = 1$	$A(BC) = (AB)C$	$A+B\cdot C = (A+B)(A+C)$
$A\cdot A = A$	$A+A = A$	$A+(B+C) = (A+B)+C$	

prove it using Boolean algebra rules