Digital Electronics

UNIT-04

Boolean Algebra

1. Number System
2. Data Representation
3. Logic Gates
4. Boolean Algebra

Boolean Algebra

↳ George Boole ⟶ Book ⟶ The Laws of thought (1854)

Boolean Algebra

↳ It is a mathematics of two variables – 'high' and 'low', that is used to analyse and simplify the digital circuit
↳ (logic)

→ Only Binary numbers are used, i.e. Value = {0, 1}

Following are the basic important rules used in Boolean Algebra

1) Variables used can have any two values, i.e. 0 or 1

2) Complement of a variable is represented by 'overbar' ($-$) or prime (')

Ex: $A = 0$, $\bar{A} = 1$, $A' = 1$

3) OR-ing: ORing of a variable is represented by (+) plus symbol.

Ex. $A + B \rightarrow$ A OR B

A OR B OR C $\rightarrow A + B + C$

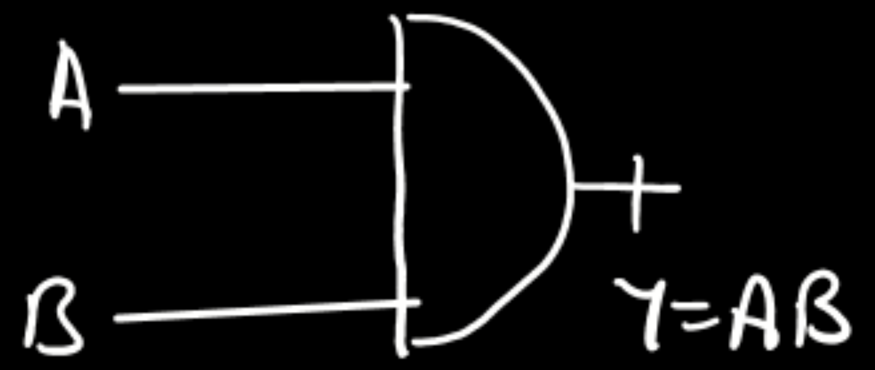
4) AND-ing: If two or more variables are represented by writing a dot ($\cdot$) between them such as $\underbrace{A \cdot B \cdot C}_{\text{Omitted}} \Rightarrow$ A AND B AND C

$\rightarrow ABC$

Omitted $\rightarrow$ Ignore

Boolean Laws

AND Law



A = B = High , O/P = high

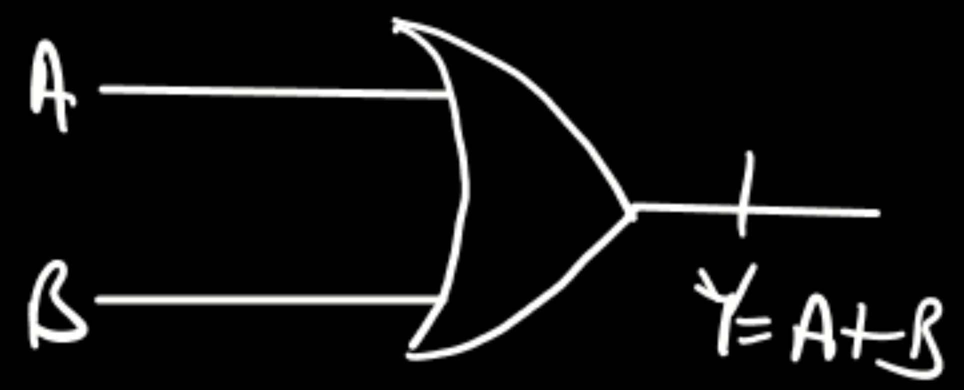
a) $A \cdot 0 = 0$ (Disabled)

b) $A \cdot 1 = A$ (Enabled)

c) $A \cdot A = A$ (Buffer)

d) $A \cdot \overline{A} = 0$ (Disabled)

OR Law



A / B = 1, O/P = 1

a) $A + 0 = A$ (Enabled)

b) $A + 1 = 1$ (Disabled)

c) $A + A = A$ (Buffer)

d) $A + \overline{A} = 1$ (Disabled)

Inversion Law

↳ The law uses NOT operation.

$$\overline{\overline{A}} = A$$

⇒ The double inversion of a variable results the original variable itself.

$A = 0$ $\overline{\overline{A}} \Rightarrow \overline{1} = 0$

$A = 1$, $\overline{\overline{A}} = \overline{0} = 1$

Commutative law :

↳ If we change the sequence of the variables it does not have any effect on the output of the logical Circuit

Ex $A+B = B+A$, $A \cdot B = B \cdot A$

Associative Law :

↳ This law states that the order in which the logic operation are performed is irrelevent as their effect is same.

↳ Unwanted / wrong / unexpected

$(A+B)+C = A+(B+C)$

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

followed by AND, OR, XOR, XNOR ✓

⇒ NAND & NOR gates do not follow this law

Distributive Law:

↳ It States the following Condition:

$$A\cdot(B+C) = AB + AC \checkmark \longleftarrow \text{Same as Normal algebra.}$$

De-Morgan's first Theoram: $\overline{AB} = \overline{A} + \overline{B}$

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

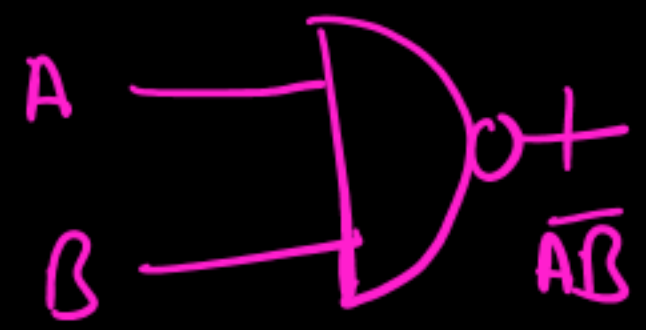
$$\overline{AB} = \overline{A} + \overline{B}$$

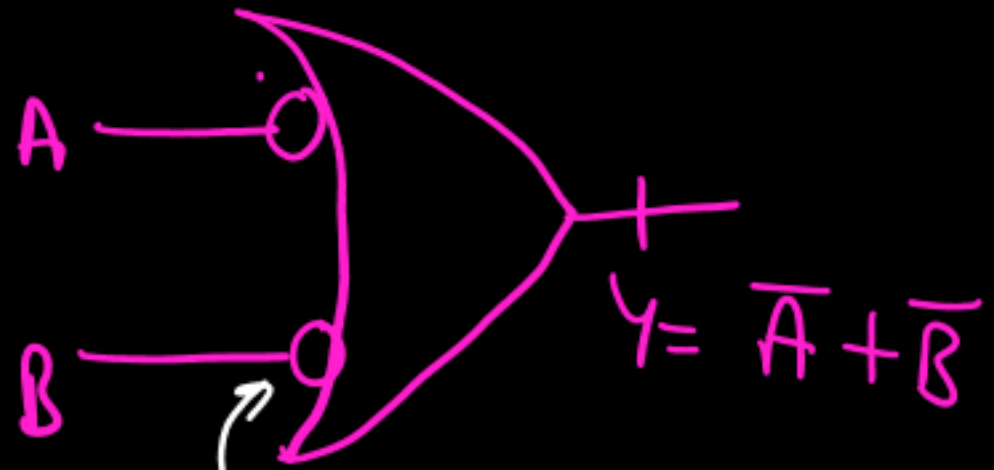
NAND gate

OR gate

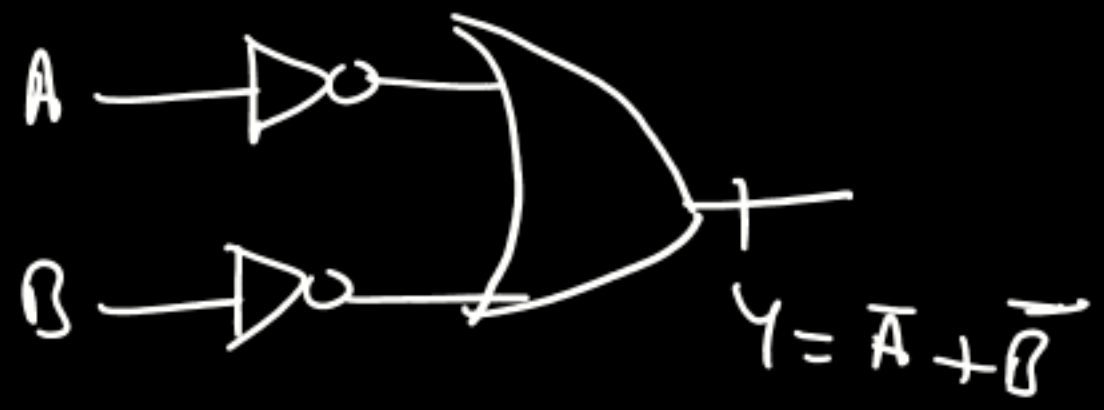
$$\overline{AB} = \overline{A} + \overline{B}$$

$\Downarrow$

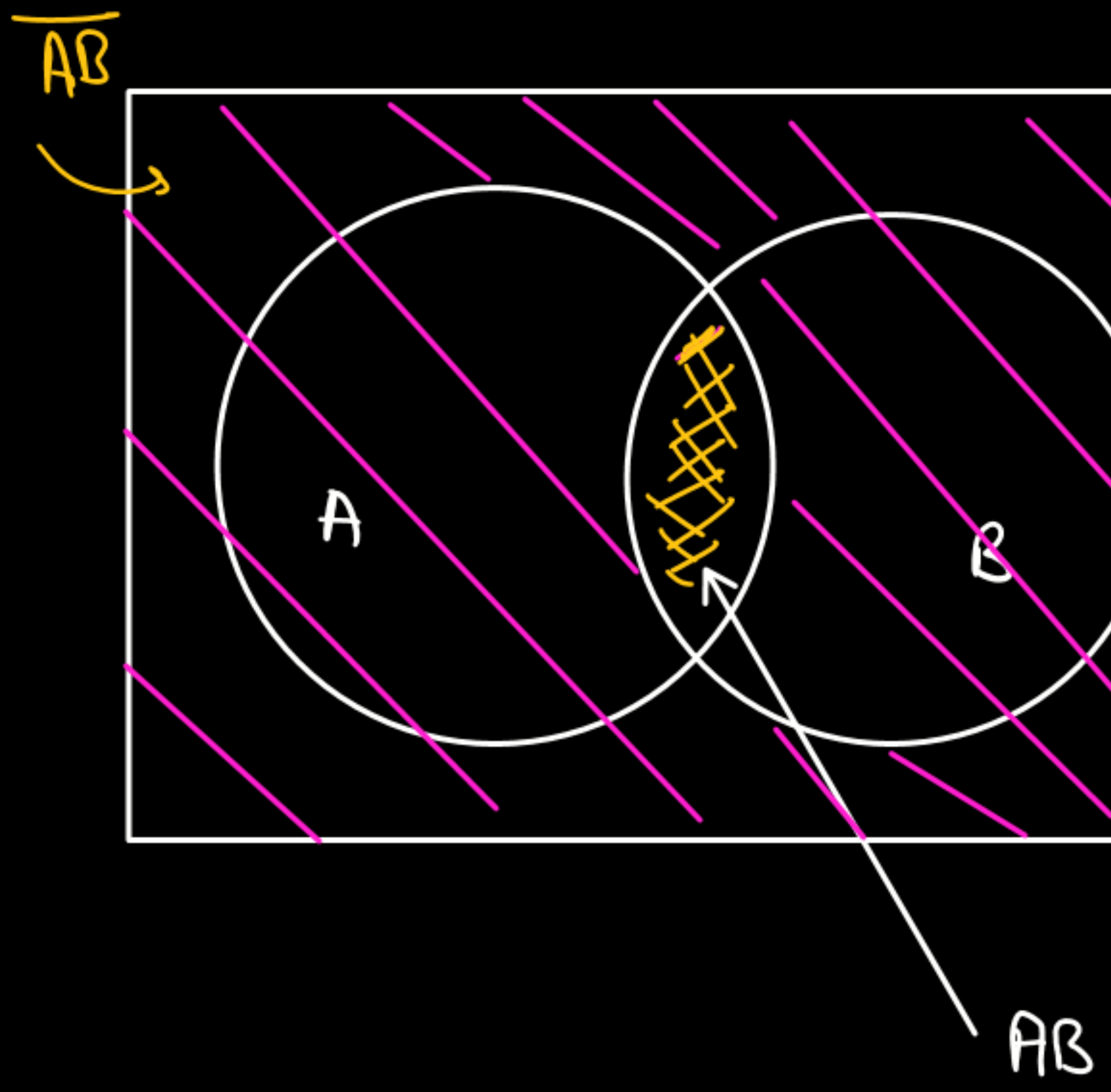


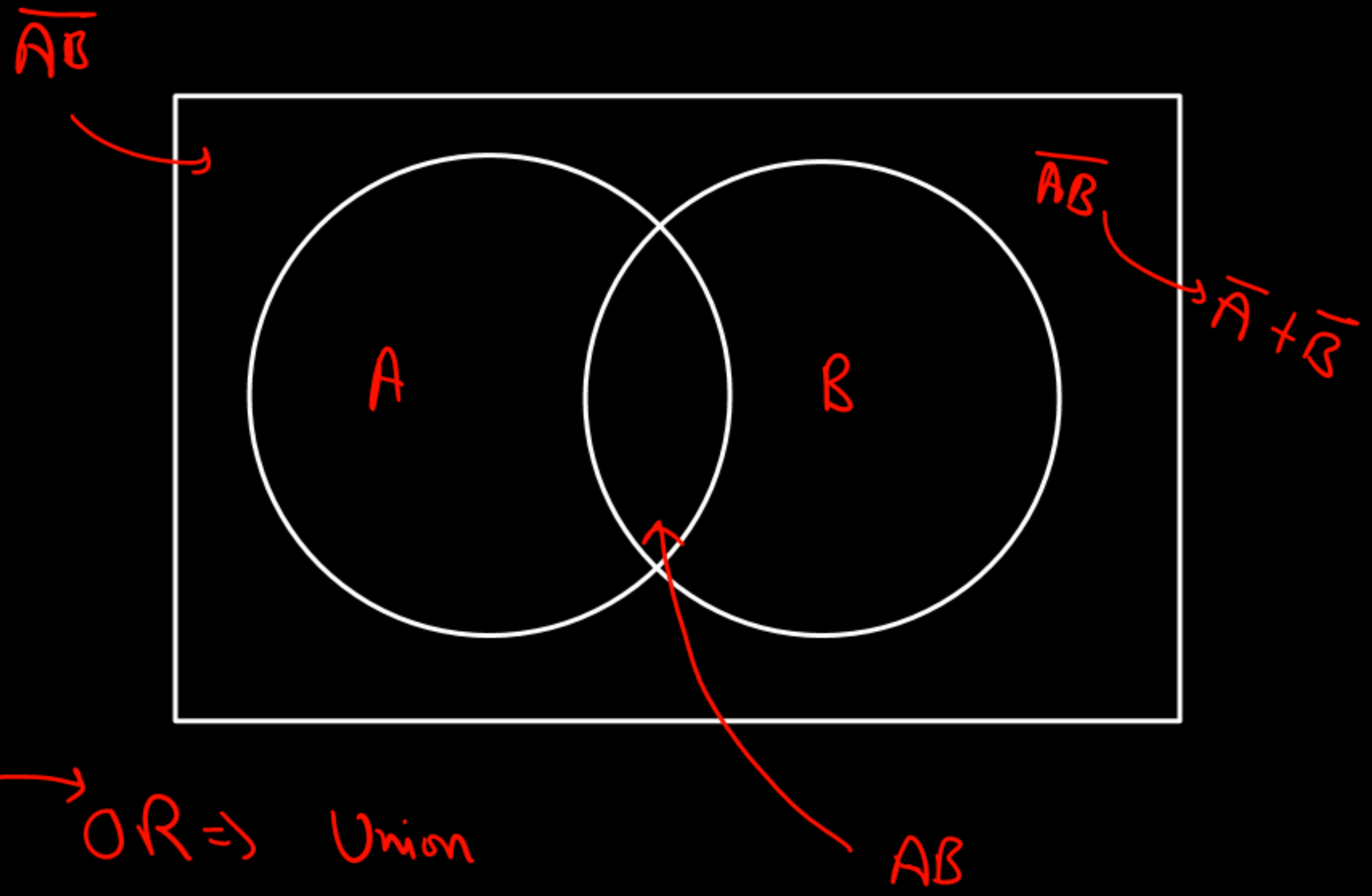


Bubble implies Inversion



Using Venn Diagram





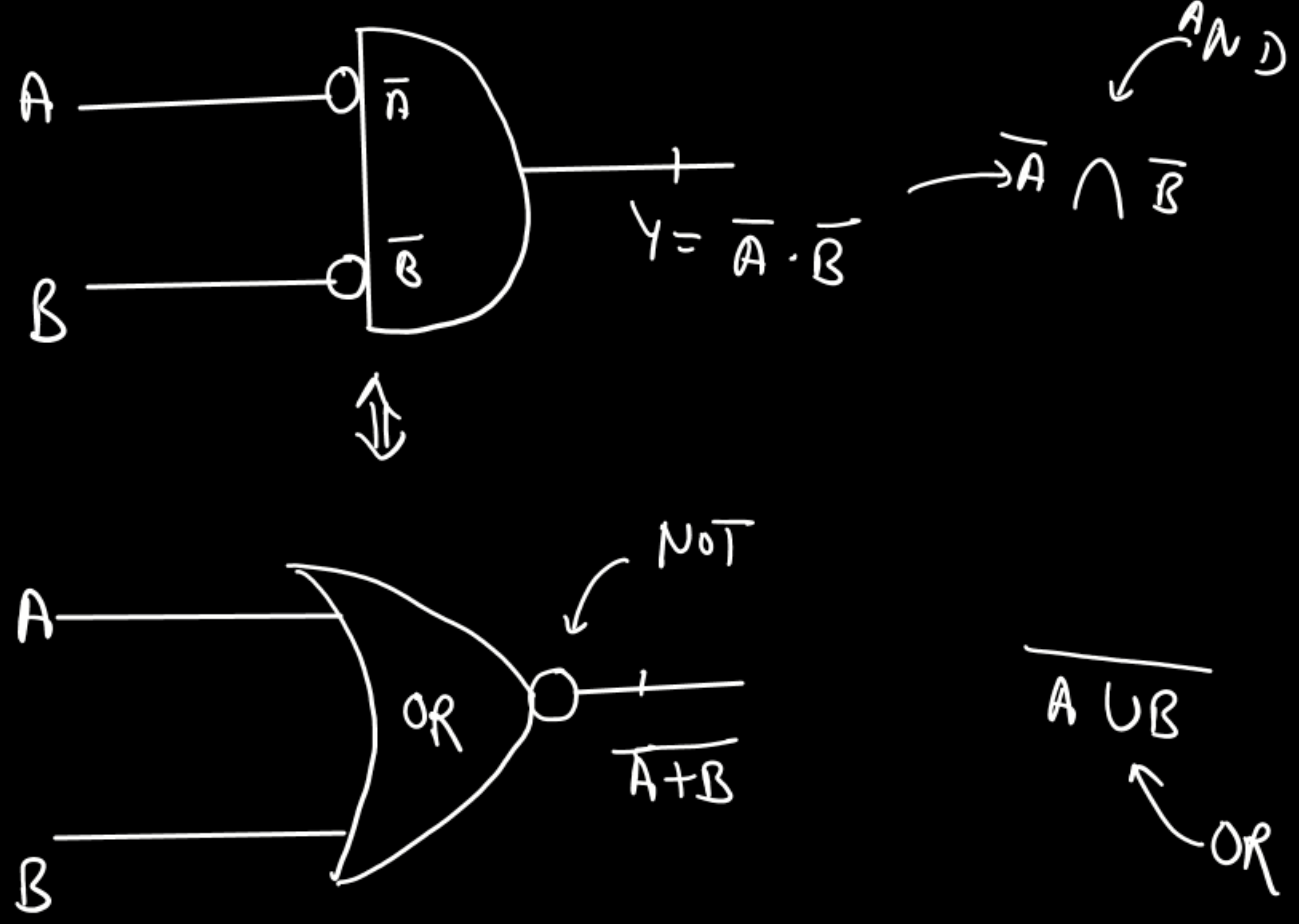
$A \cup B \rightarrow$ OR $\Rightarrow$ Union

$A \cap B \rightarrow$ AND = Intersection

De-Morgan's Second Theoram : $\overline{A+B} = \bar{A} \cdot \bar{B}$

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

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



Bubbled Input AND gate = NOR gate

$\downarrow$

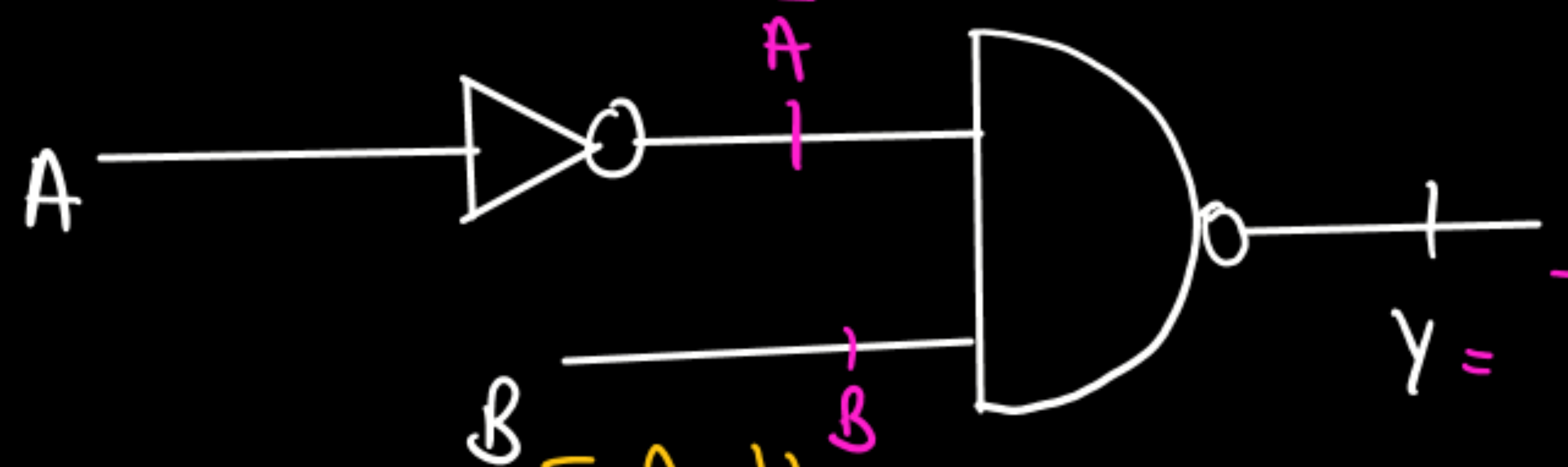
Inverter

Implication & Inhibition

$\overline{AB} = \bar{A} + \bar{B}$

Implication

NAND



$Y = \overline{\bar{A} \cdot B}$

$\bar{\bar{A}} + \bar{B}$ Inversion law

$\Rightarrow A + \bar{B}$

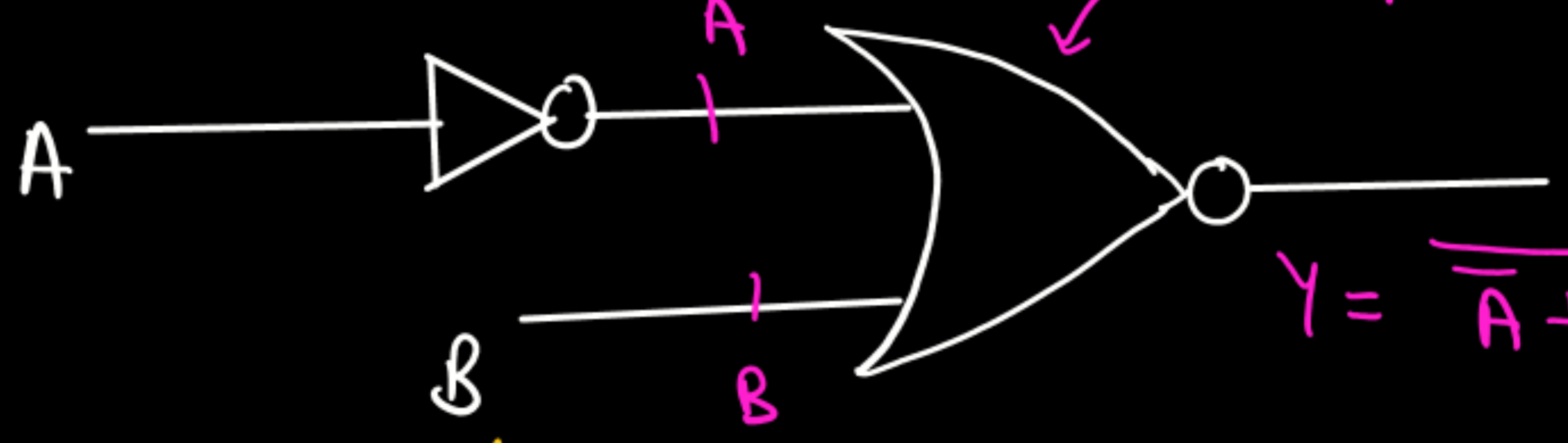
B ← Imply

Input → $\bar{A} \to A$

$B \to \bar{B}$

Inhibition

NOR



$Y = \overline{\bar{A} + B}$

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

$\overline{\bar{A} + B} = \bar{\bar{A}} \cdot \bar{B}$ Inversion law

$= A \cdot \bar{B}$

$\bar{\bar{A}} = A$

B ← Inhib

⇒ If Inhib = 1, then A will not reach to output

⇒ $A \cdot \bar{B}$, $B = 1$, ⇒ $A \cdot \bar{1} = A \cdot 0 = 0$

* Imply = 1, (Imply → B) $B = 1$, $Y = A + \bar{B}$

⇒ $A + \bar{1} = A + 0 = A$

↳ if Imply = 1, A → output

tomorrow

↳ Duality Theorem,

→ Redundancy Theorem, ← Rules

Rules