

2. Boolean Laws, Realisation of Boolean Gates using Universal Gates

07 November 2023 14:46
BOOLEAN LAWS

Inversion law

$$\begin{aligned} \text{If } A = 1 &\longrightarrow A' = 0 \\ A = 0 &\longrightarrow A' = 1 \end{aligned}$$

AND law

$$A \cdot 0 = 0$$

$$A \cdot 1 = A$$

$$A \cdot A = A$$

$$A \cdot A' = 0$$

Double Inversion law

$$(A')' = A$$

OR law

$$A + 0 = A$$

$$A + 1 = 1$$

$$A + A = A$$

$$A + A' = 1$$

Principle of duality / Duality theorem

- Each AND sign is changed to an OR sign, 0s are changed to 1s and vice versa for both.

Dual of AND gate \Rightarrow OR gate

$$A \cdot B \Rightarrow A + B$$

Dual of NAND gate \Rightarrow NOR gate

$$\overline{A \cdot B} \Rightarrow \overline{A + B}$$

Dual of XOR gate \Rightarrow XNOR gate

Commutative law

$$A + B = B + A$$

$$A \cdot B = B \cdot A$$

Associative law

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

From principle of duality:

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

Distributive law

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

$$A + \bar{A} \cdot B = (A + \bar{A}) \cdot (A + B) = A + B$$

redundant

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

redundant

From principle of duality,
 $A \cdot (B+C) = A \cdot B + A \cdot C$

De Morgan's Theorem

"Break the line, change the sign"

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

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

Absorption Theorem

$$(i) \quad A + AB = A$$

$$\therefore A + AB = A \underbrace{(1+B)}_{=1}$$

$$\Rightarrow A + AB = A$$

$$(ii) \quad A(A+B) = A$$

$$\begin{aligned} &= A \underbrace{\bar{A} + A \cdot B}_{=1} \\ \Rightarrow A + AB &= A \underbrace{(1+B)}_{=1} = A \end{aligned}$$

Redundancy Laws

$$(i) \quad A + \bar{A}B = A + B$$

$$(ii) \quad A \cdot (\bar{A} + B) = AB$$

$$A + \bar{A} \cdot B = (A + \bar{A}) \cdot (A + B) = A + B$$

redundant
since $(A + \bar{A})$
is 1

Consensus Theorem

$$Y = AB + BC + \bar{A}C = AB + \bar{A}C$$

redundant

$$\begin{aligned} &AB + A\bar{C} + BC (A + \bar{A}) \\ &= AB + A\bar{C} + ABC + \bar{A}BC \\ &= AB + ABC + \bar{A}C + \bar{A}BC \\ &= AB(1+C) + \bar{A}C(1+B) \\ &= AB + \bar{A}C \end{aligned}$$

redundant
since both
equal one

Example

If you have an expression with no single variable:

- Check terms for any variable X and its complement \bar{X} .
- If such terms exist, then check their coefficients.
 Eg: $XA, \bar{X}B$
 Coefficients: A, B
- Check for terms consisting of only the

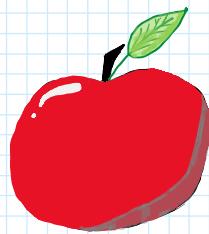
Example

$$\begin{aligned} \textcircled{1} \quad Y &= \bar{A}B + \bar{B}\bar{C} + \underline{\bar{B}C} + A\bar{B} + AC \\ &= \bar{A}B + \bar{B}\bar{C} + \underline{A\bar{B}} + AC \\ &= \bar{A}B + \bar{B}\bar{C} + AC \end{aligned}$$

$$\begin{aligned} \textcircled{2} \quad Y &= AB + A\bar{C} + \bar{A}\bar{B} + \underline{\bar{B}\bar{C}} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + ABC \\ &= \underline{AB} + \underline{A\bar{C}} + \underline{\bar{A}\bar{B}} + \underline{A\bar{B}\bar{C}} + \underline{\bar{A}\bar{B}C} + \underline{ABC} \\ &= AB(1+C) + \bar{A}\bar{B}(1+C) + A\bar{C}(1+\bar{B}) \\ &= AB + \bar{A}\bar{B} + A\bar{C} \end{aligned}$$

equal one

- Check for terms consisting of only the coefficient variables (eg: AB). Such terms are taken as redundant.



REMEMBER: A P P L E

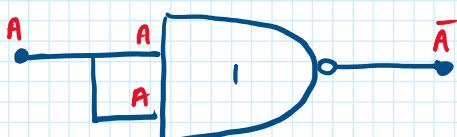
UNIVERSAL GATES



- Any digital logic circuit can be implemented using these

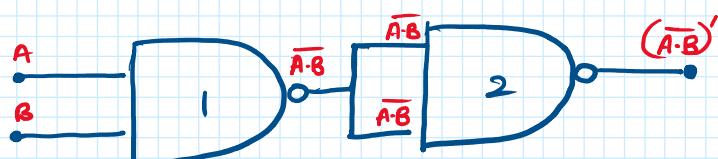
REALISATION OF LOGIC GATES USING NAND

① NOT GATE



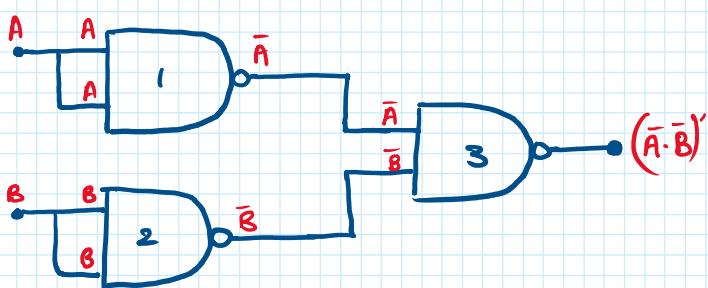
$$Y = \overline{A \cdot A} = \overline{A} + \overline{A} = \overline{A}$$

② AND GATE



$$Y = (\overline{A \cdot B})' = AB$$

③ OR GATE

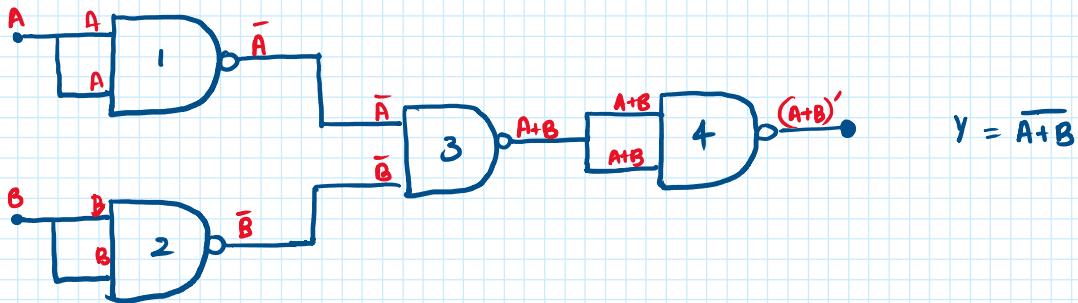


$$Y = (\overline{A \cdot B})' = (\overline{A} + \overline{B})'$$

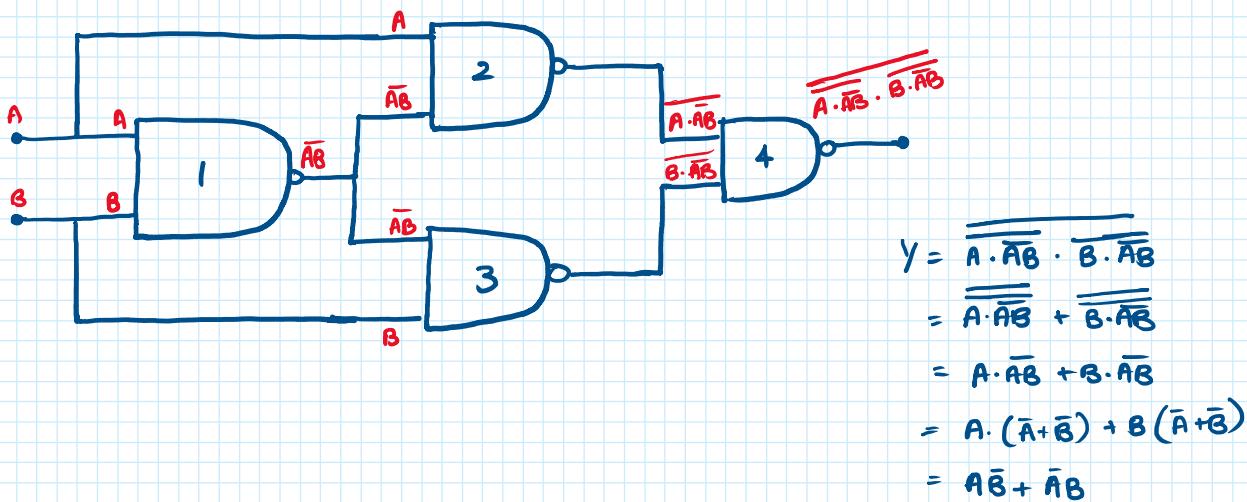
$$Y = A + B$$

④ NOR GATE



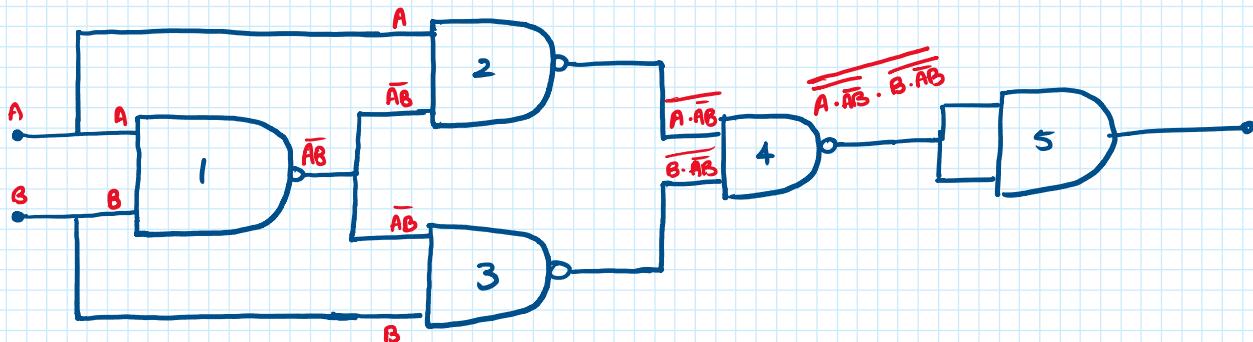


⑤ XOR GATE



$$\begin{aligned}
 Y &= \overline{\overline{A} \cdot \overline{B}} \cdot \overline{B \cdot \overline{A}} \\
 &= \overline{\overline{A} \cdot \overline{B}} + \overline{B \cdot \overline{A}} \\
 &= A \cdot \overline{B} + B \cdot \overline{A} \\
 &= A \cdot (\overline{A} + \overline{B}) + B \cdot (\overline{A} + \overline{B}) \\
 &= A\overline{B} + B\overline{A}
 \end{aligned}$$

⑥ XNOR GATE



REALISATION OF LOGIC GATES USING NOR

NOT - 1

OR - 2

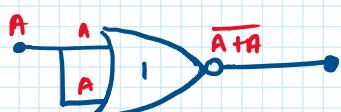
AND - 3

NAND - 4

XOR - 5

XNOR - 4

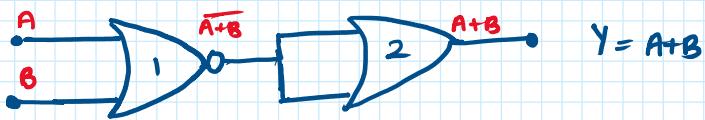
① NOT GATE



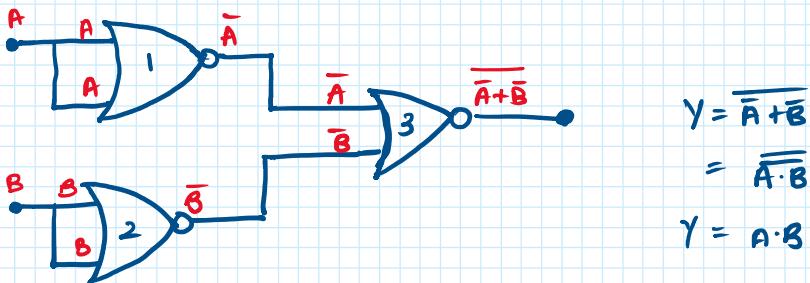
$$Y = \overline{A+A} = \overline{A} \cdot \overline{A}$$

$$Y = \overline{A}$$

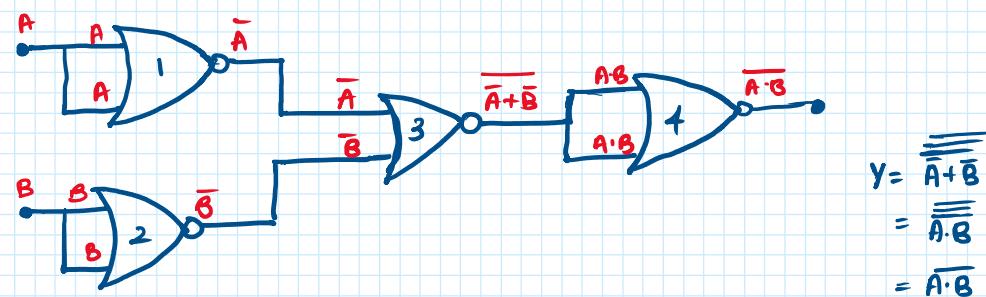
② OR GATE



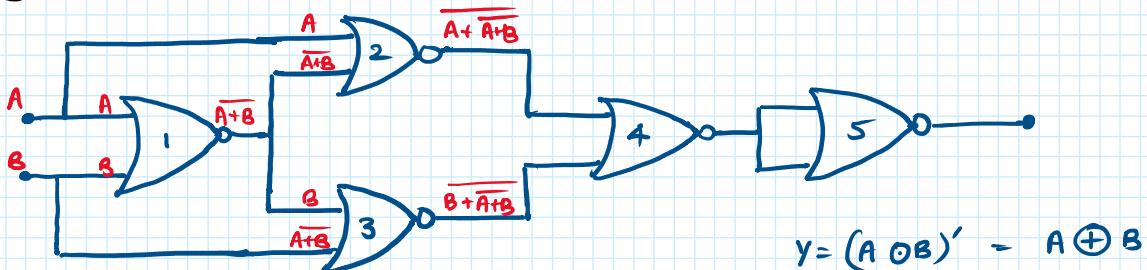
(3) AND GATE



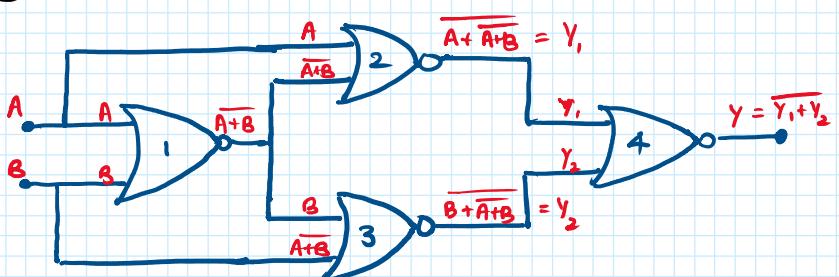
(4) NAND GATE



(5) XOR GATE



(6) XNOR GATE



$$\begin{aligned}
 Y_1 &= \overline{\bar{A} + \bar{B}} \\
 &= \bar{A} \cdot \bar{B} \\
 &= \bar{A}(\bar{A} + B) \\
 &= \bar{A}B
 \end{aligned}
 \quad \left| \quad \begin{aligned}
 Y_2 &= \overline{B + \bar{A} + \bar{B}} \\
 &= \bar{B} \cdot \bar{A} \\
 &= \bar{B}(\bar{A} + B) \\
 &= \bar{A}\bar{B}
 \end{aligned} \right.$$

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

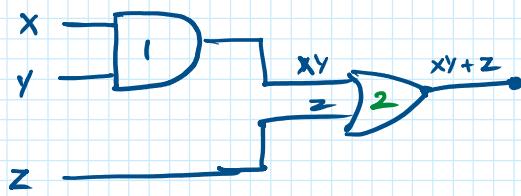
PROBLEMS

① Realise $F = XY + Z$ using only NAND gates

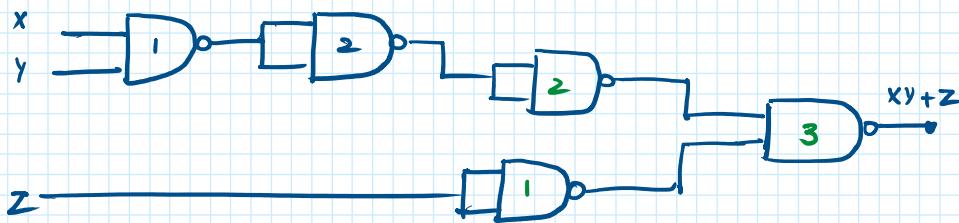
Solution:

Method ①

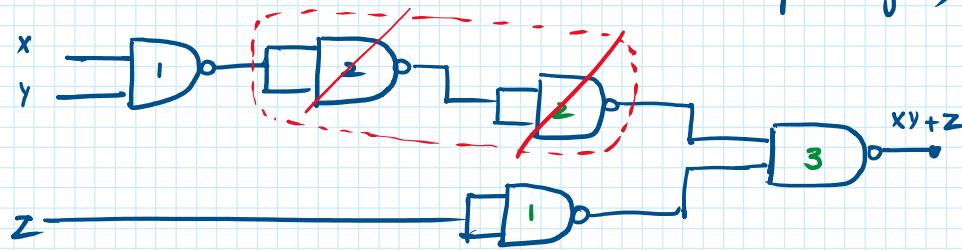
Step ①: Draw basic gate expression



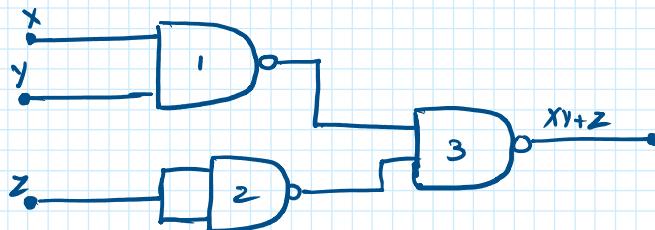
Step ②: Replace each gate by its NAND equivalent



Step ③: Remove double successive inversions (doubled up NOT gates)

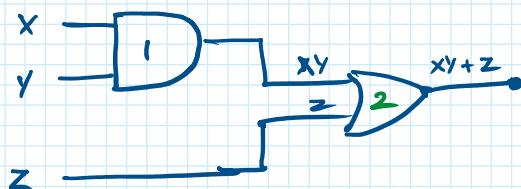


Solution:

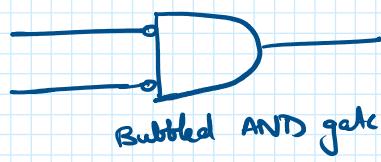
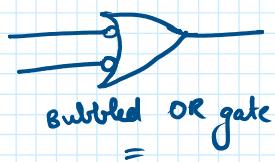


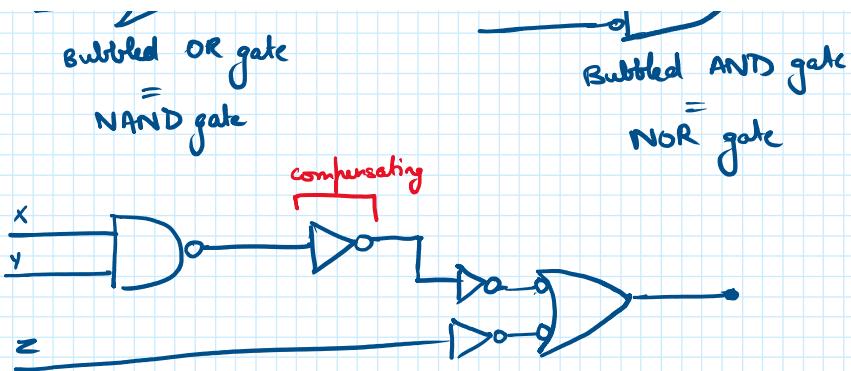
Method ②

Step ①: Draw basic gate expression



Step ②: Replace each gate symbol with NAND gate symbol





Step ③: Remove double successive inversions and ensure every diode is in terms of NAND