Digital Logic

Lecture 4

2nd Stage

Computer Science Department

Faculty of Science

Soran University

Rafiq.maulud@soran.edu.iq

Fall 2022

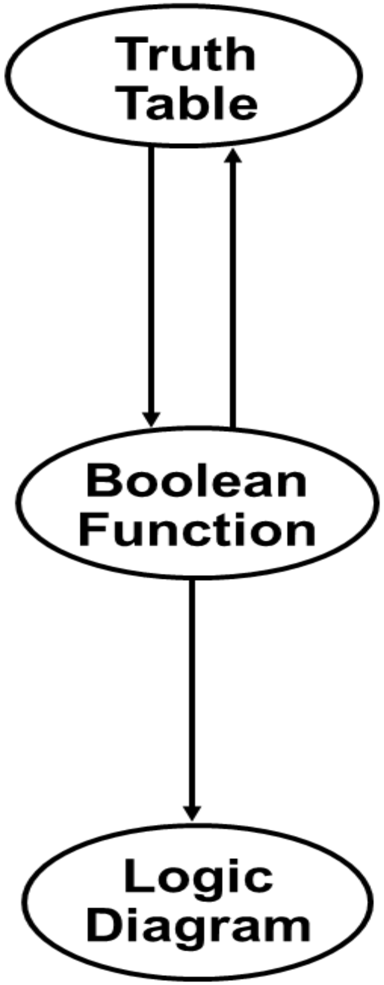
Topics covered

- Introduction
- Logic Gates
  - OR gate
  - AND gate
  - NOT gate
  - NAND gate
  - NOR gate
  - EXOR gate
  - EXNOR gates.
- Universal gates

Logic Gates

- A logic gate is an electronic circuit/device which makes logic decisions.
- Most logic gates are two inputs and one output.
- At any given moment, every terminal is in one of the two binary conditions low (0) or high(1), represented by different voltage levels.
- In most logic gates, the low state is approximately 0v, while the high state is approximately 5v.

Logic Circuit Design:

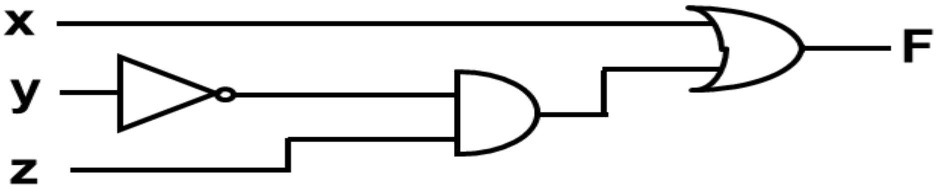

Truth Table

Boolean Function

Logic Diagram

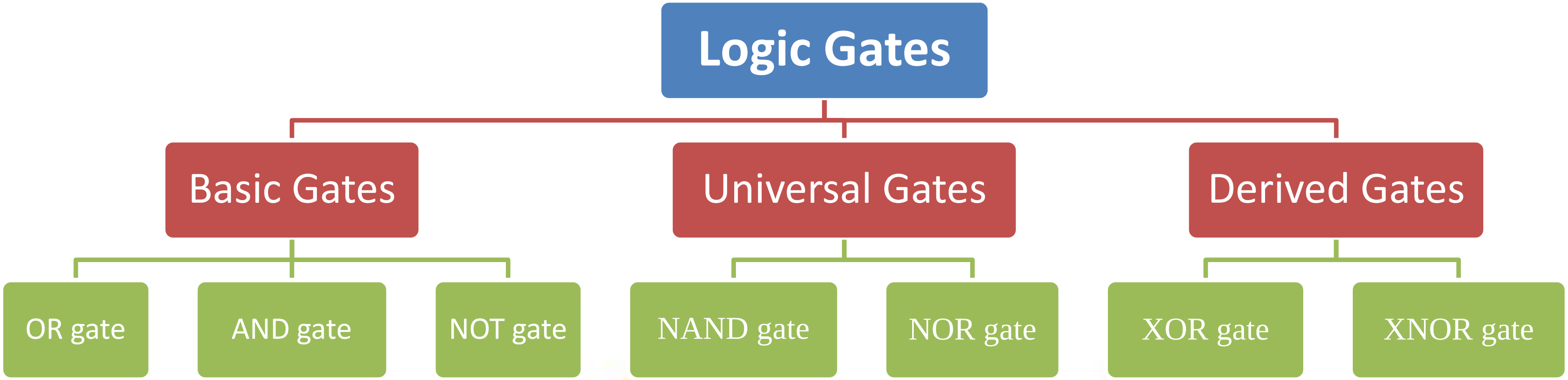
x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

$F = x + y'z$



Logic Gates

The most common logic gates used are:



Logic Gates

Basic Gates

Universal Gates

Derived Gates

OR gate

AND gate

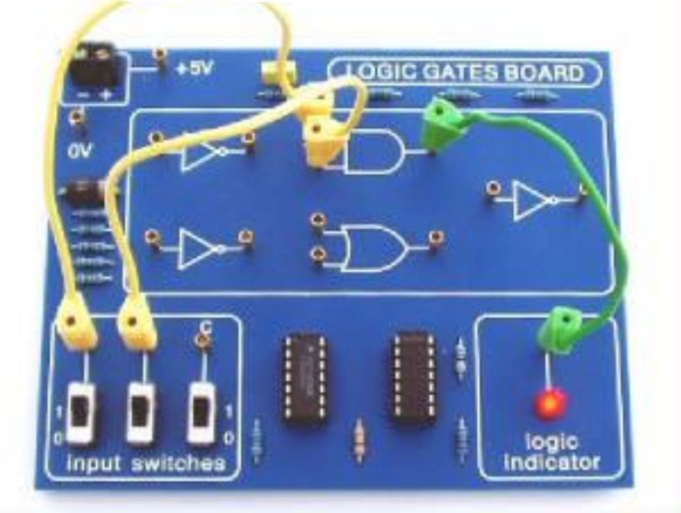
NOT gate

NAND gate

NOR gate

XOR gate

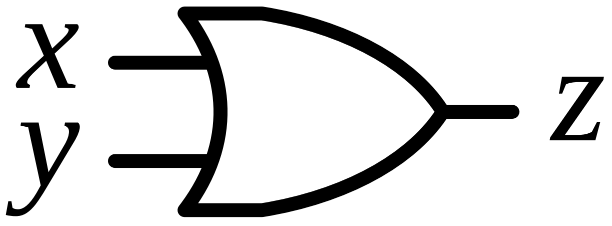
XNOR gate



1-Basic gates

OR gate

The OR gate is an electronic circuit that its output is true if at least one input is true. The symbol ‘+’ indicates the OR operation.



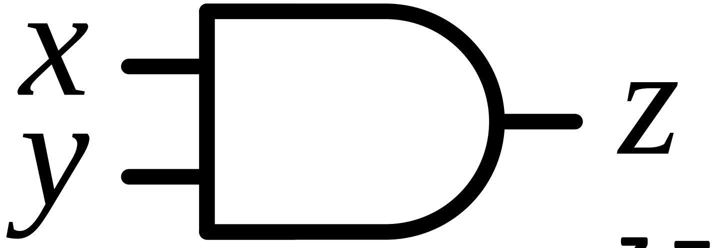
z = x + y

x	y	z
0	0	0
0	1	1
1	0	1
1	1	1

1-Basic gates

AND gate

The AND gate is also a basic kind of digital circuit. The output of the AND gate is one only when both inputs are one. A dot (.) is used to show the AND operation i.e. A.B.



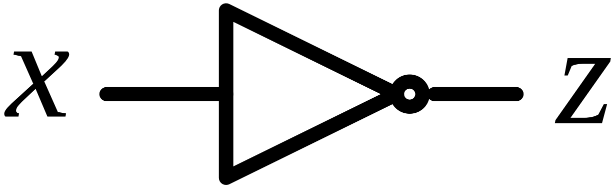
$$z = x \bullet y = x\,y$$

x	y	z
0	0	0
0	1	0
1	0	0
1	1	1

1-Basic gates

NOT gate

A NOT gate is a basic gate that has one input and one output that produces an inverted version of the input at its output. It is also known as an inverter. If the input variable is A, the inverted output is known as NOT A. This is also shown as A', or A with a bar over the top.

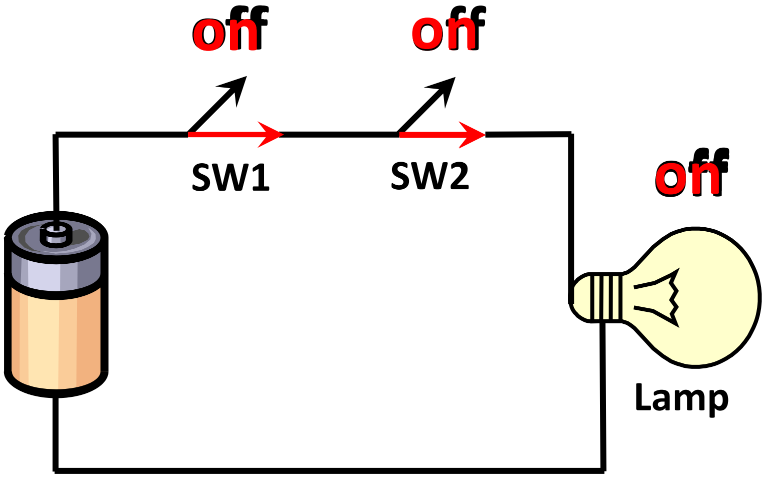


$$Z = \overline{X}$$

x	z
0	1
1	0

Switching Circuits

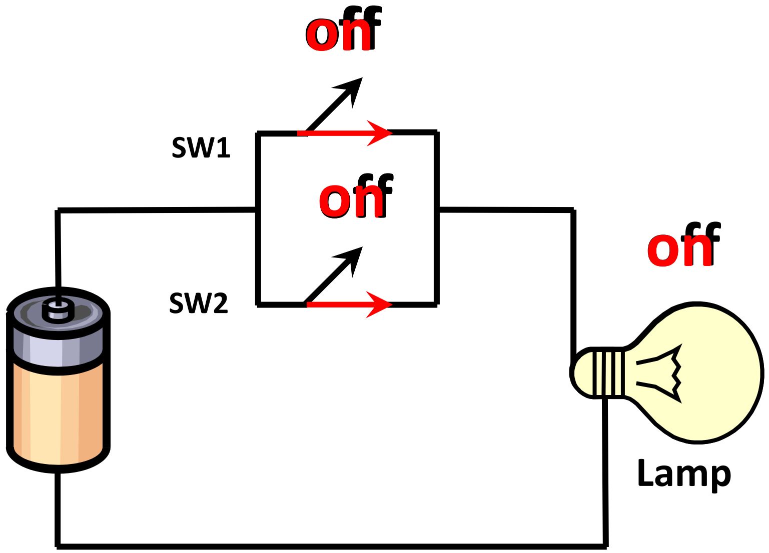
Representing **AND Gate** by Electric Circuit



SW1	SW2	Lamp

Switching Circuits

Representing OR Gate by Electric Circuit

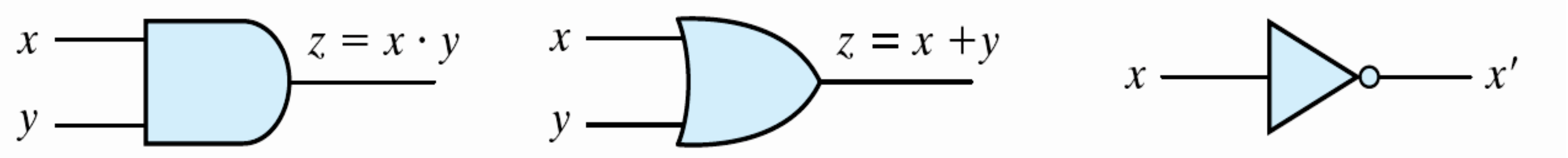


SW1	SW2	Lamp

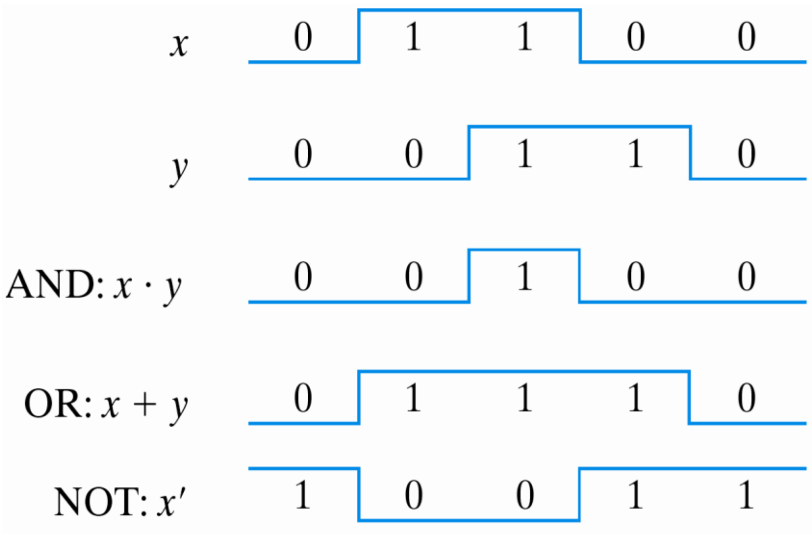
Binary Logic

✧ Logic gates

- Graphic Symbols and Input-Output Signals for Logic gates:



(a) Two-input AND gate (b) Two-input OR gate (c) NOT gate or inverter

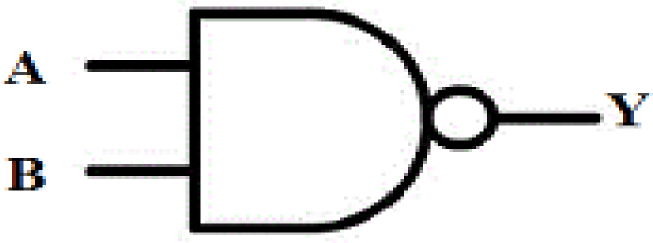


2- Universal gates

NAND Gate

The **NAND** gate represents the complement of the AND operation. Its name is an abbreviation of **N**OT **AND**.

Input		Output
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



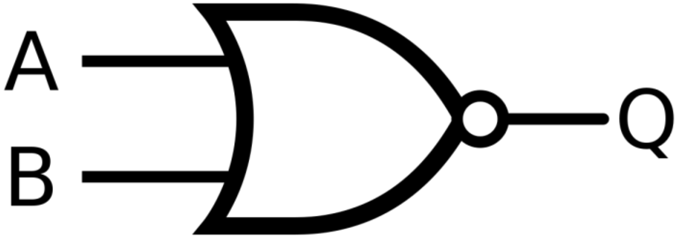
$$Y = \overline{A.B}$$

2- Universal gates

NOR Gate

The NOR gate represents the complement of the OR operation. Its name is an abbreviation of NOT OR.

Input		Output
A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0



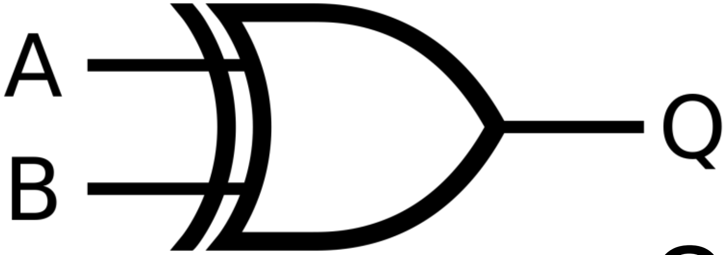
$$Q=\overline{A+B}$$

3- Derived gates

X-OR Gate

The output is high when either of inputs A or B is high, but not if both A and B are high. It is consists of a five gates (two AND, two NOT, and one OR)

Input		Output
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

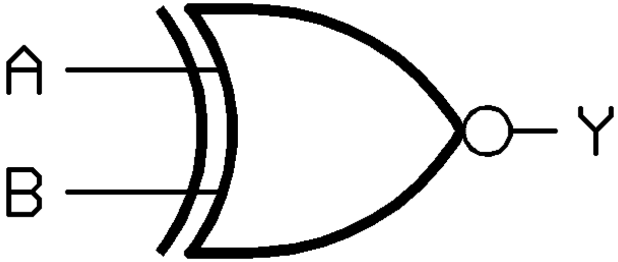


$$Q= A\oplus B$$

3- Derived gates

X-NOR Gate

This is an X-OR gate with the output inverted. It is consists of a five gates (two AND, two NOT, and one OR)



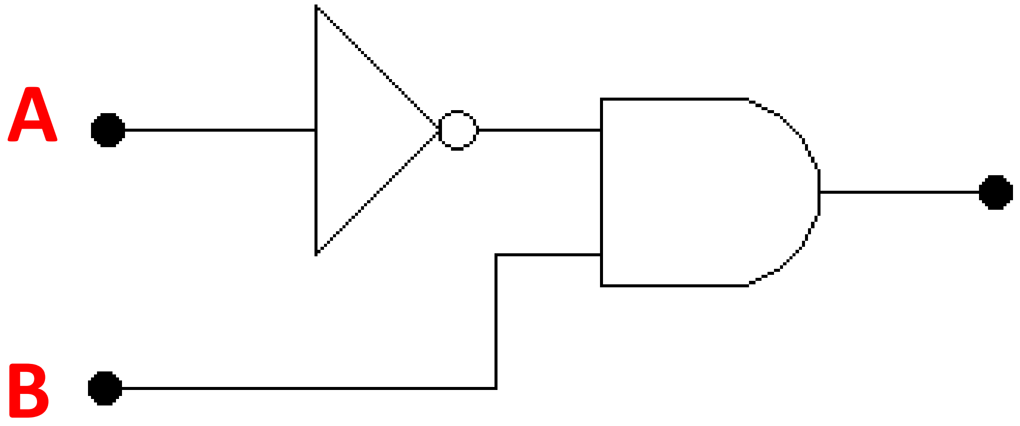
Input		Output
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

$$Y= A\otimes B$$

$$Y=\overline{A\oplus B}$$

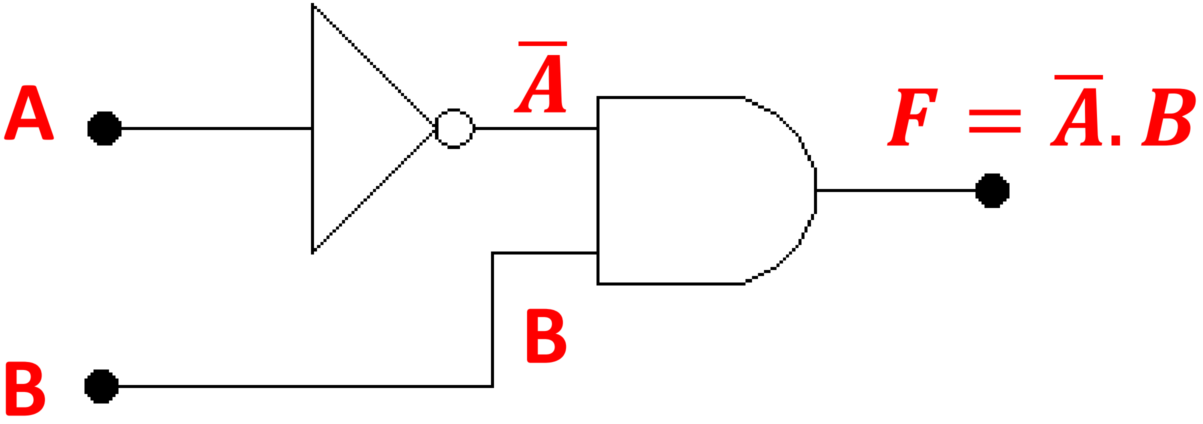
Conversion
logic diagram to logic function

Example1: **Convert the following diagram to logic function**



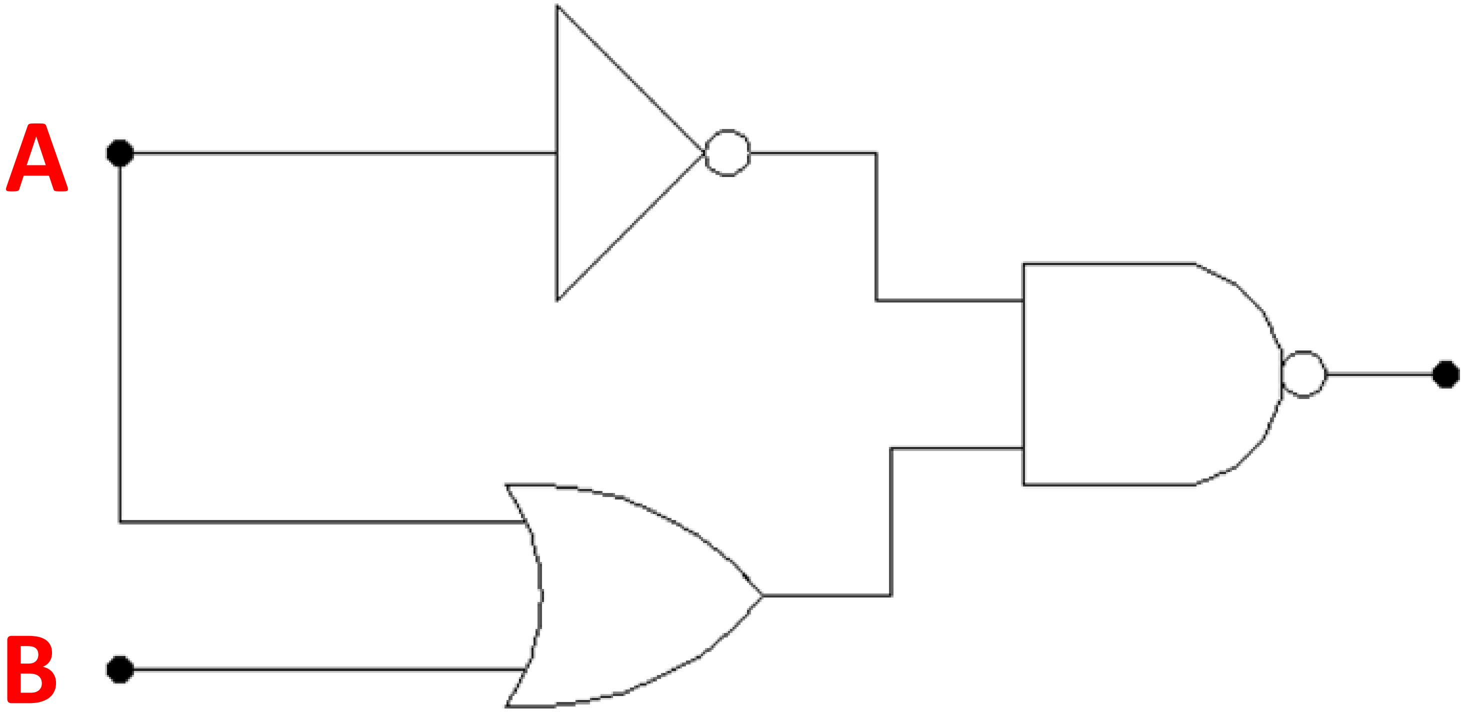
Conversion
logic diagram to logic function

Example1: **Convert the following diagram to logic function**



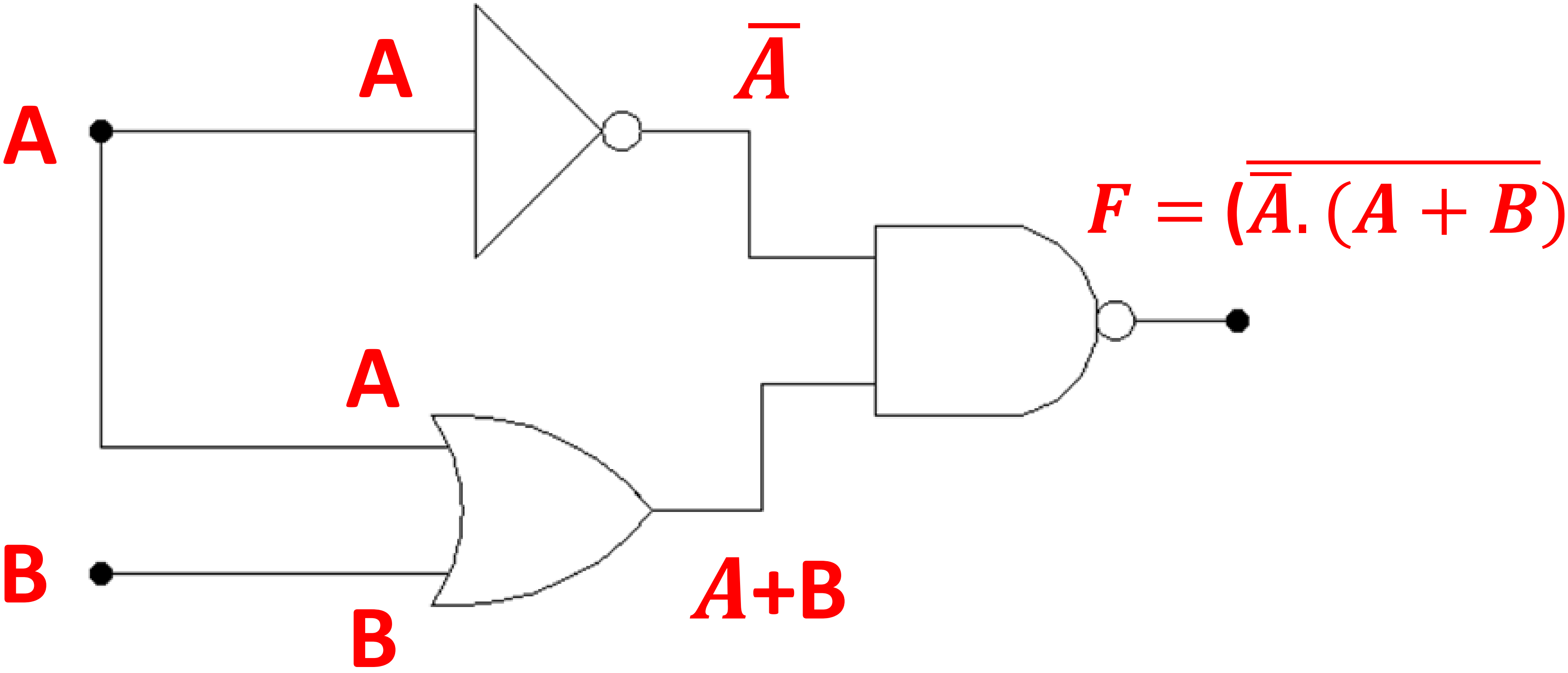
Conversion
logic diagram to logic function

Example2: Convert the following diagram to logic function



Conversion
logic diagram to logic function

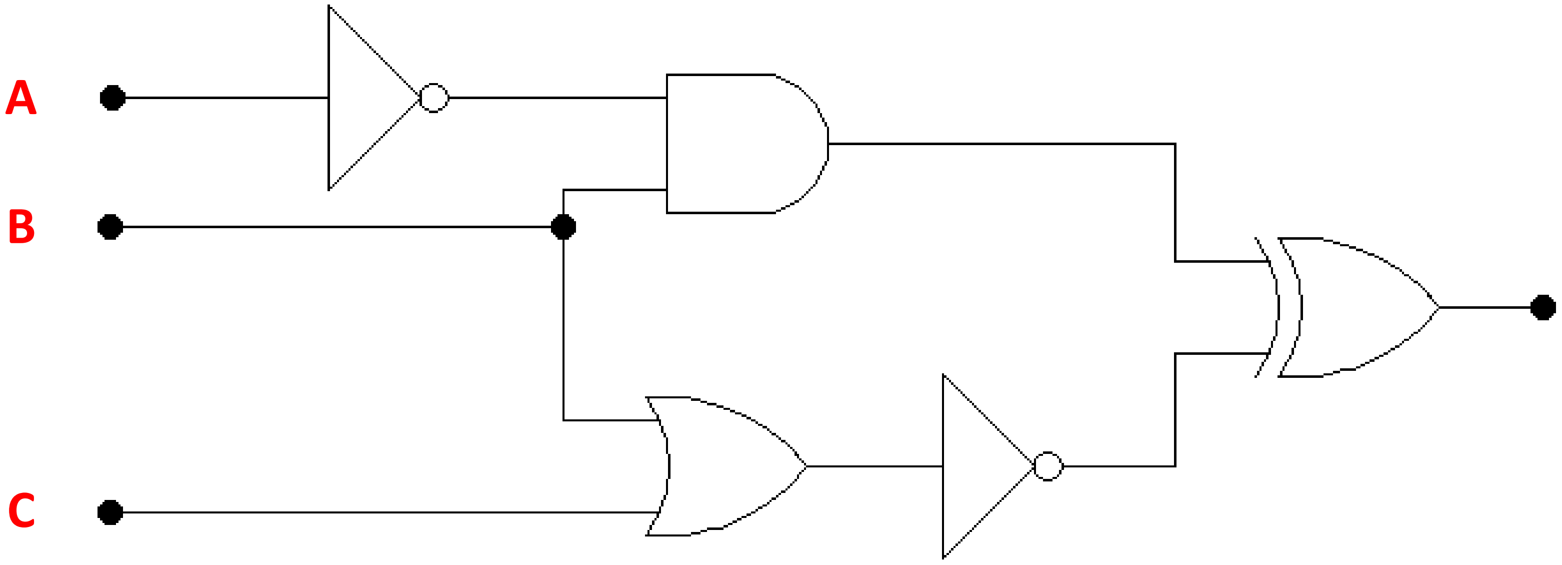
Example2: **Convert the following diagram to logic function**



$F = (\overline{\bar{A}.(A+B})$

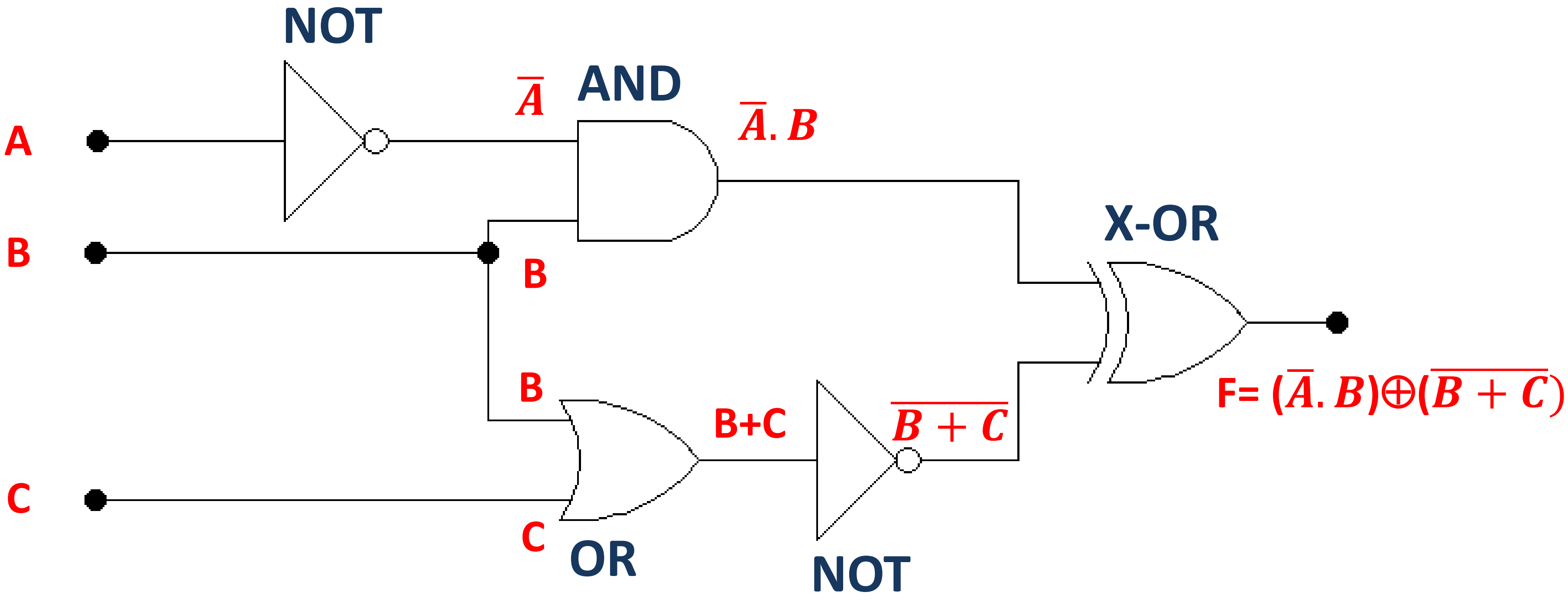
Conversion
logic diagram to logic function

Example3: Find the logic function for the following logic diagram.



Conversion
logic diagram to logic function

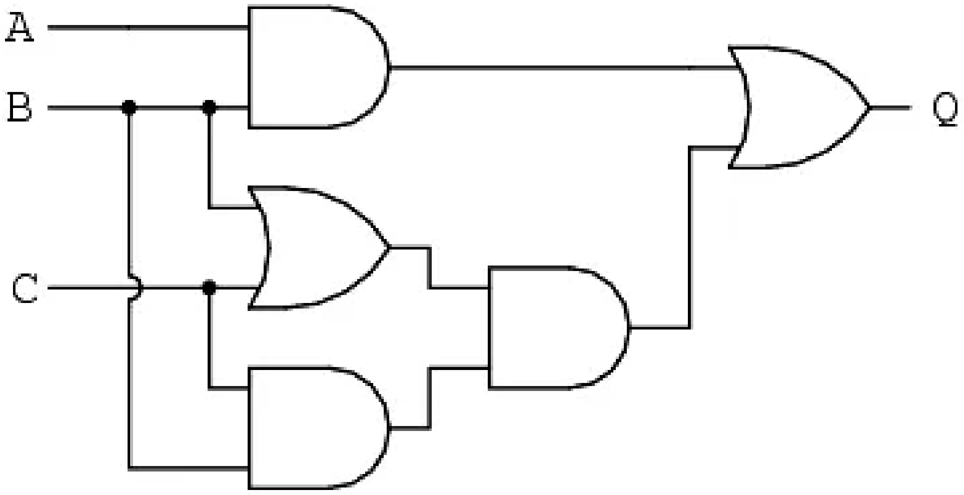
Example3: Find the logic function for the following logic diagram.



Conversion
logic diagram to logic function

Classwork 1:

Find the logic function for the following logic diagram.



Conversion
logic function to logic diagram

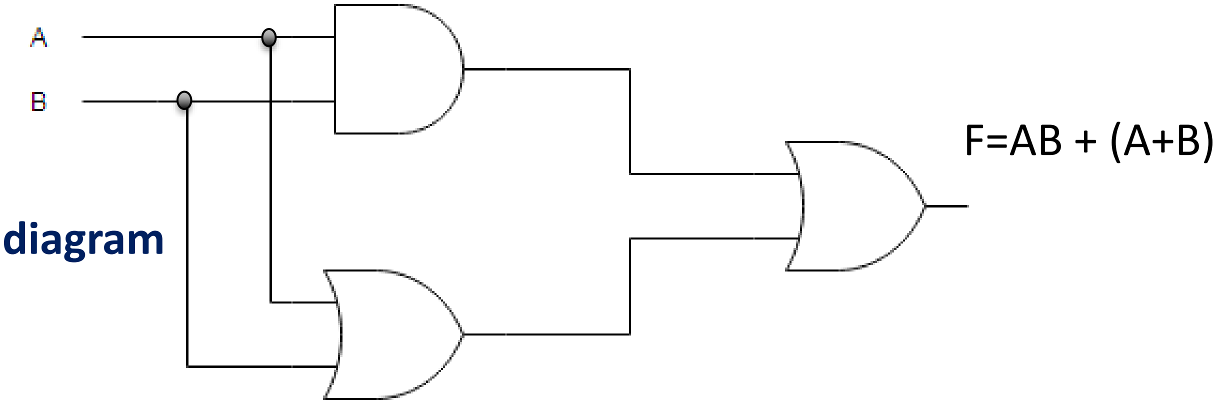
Example1:Consider the following **Boolean expression**:

(A.B) + (A+B)

Write the truth table and draw the logic diagram for it.

A	B	AB	A+B	AB + (A+B)
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

Truth Table



Logic diagram

Conversion
logic function to logic diagram

Example2: Consider the following **Boolean expression**:

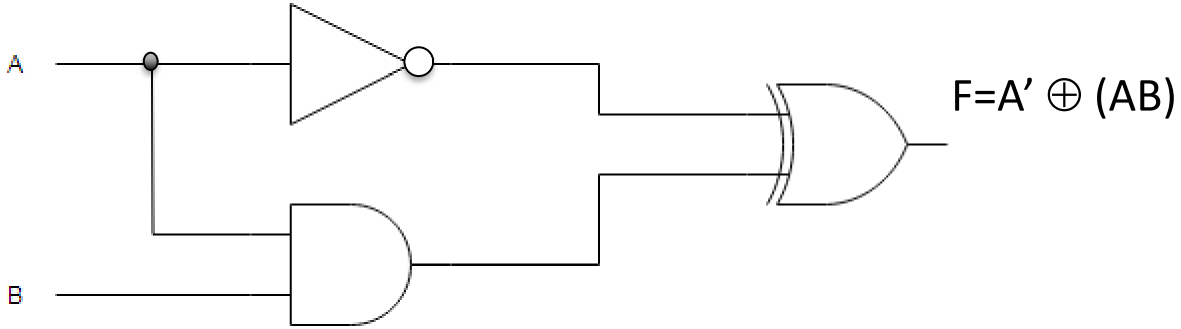
A’ ⊕ (AB)

Write the truth table and draw the logic diagram for it.

A	B	A’	AB	A’ ⊕ (AB)
0	0	1	0	1
0	1	1	0	1
1	0	0	0	0
1	1	0	1	1

Truth Table

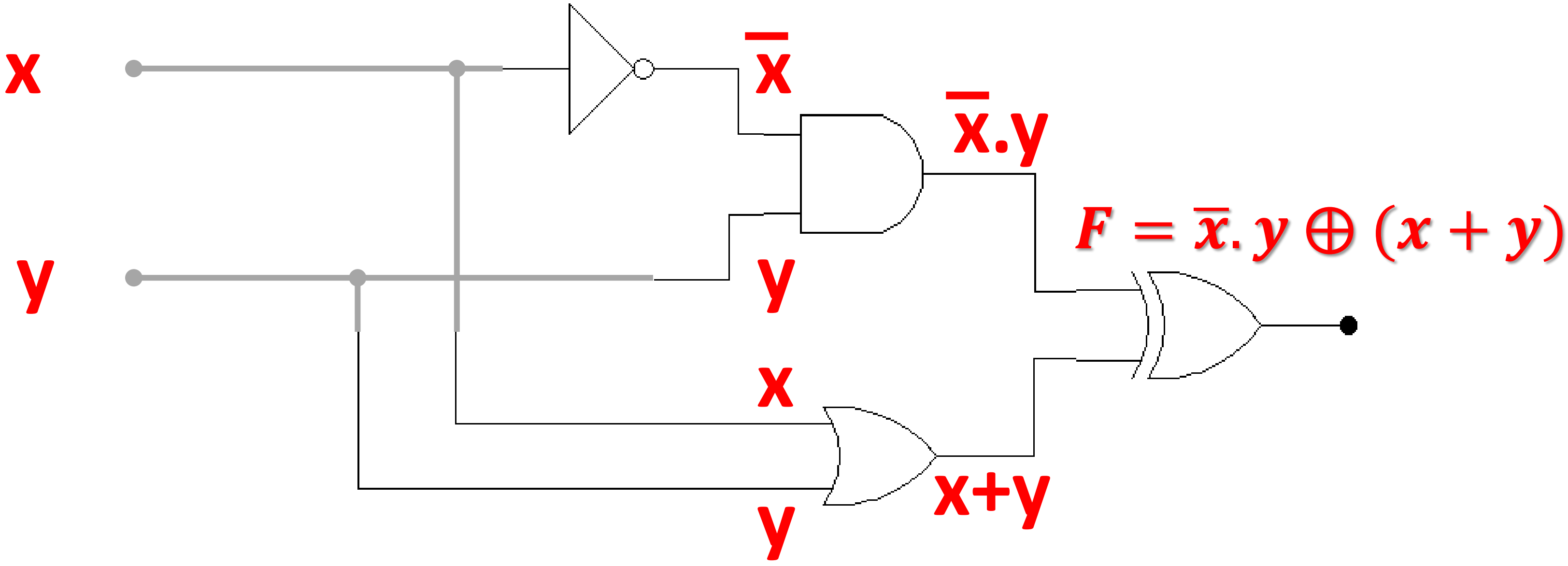
Logic diagram



Conversion
logic function to logic diagram

Example3: Convert the following logic function to diagram

$$F = \bar{x}.y \oplus (x + y)$$



Conversion
logic function to logic diagram

Classwork 2:

For the following logic function draw the logic diagram.

$$\boldsymbol{F(A,B,C) = A + \overline{(\overline{B} + C)} + \overline{(B.\overline{C}) + \overline{A}}}$$

Exercises-3

For the following logic function draw the logic diagram and write the truth table.

1) $\boldsymbol{F(X,Y,Z) = (Z + \overline{X})\otimes(\overline{Y} + X) + \overline{Y + \overline{X}})}$

2) $\boldsymbol{F(X,Y) = \overline{(\overline{X}\otimes\overline{Y}) + \overline{(X\oplus Y)}} + (X + \overline{Y})}$

Deadline: October 22, 2022 @ 11:59 PM

Homework 4

For the following logic function draw the logic diagram and write the truth table.

$$1)\ \boldsymbol{F = \overline{(X+Y)}.(\bar{X} \otimes \bar{Y}) + \overline{\overline{(X+Y)}}}$$

$$2)\ \boldsymbol{F = \overline{XYZ + (\bar{Z} + X)}.\overline{(Y.\bar{X}) + \bar{Y}}}$$

$$3)\ \boldsymbol{F = \overline{\overline{(X.\bar{Y}).}Z \oplus Y + (Z \otimes \bar{Z})}}$$

Deadline: October 28, 2022 @ 11:59 PM

Universal gates

- **NAND** and **NOR** gates are referred to as universal gates as the three basic gates can be constructed using either one of the two.

- This therefore implies that all logic circuits can be constructed using either of the gates.

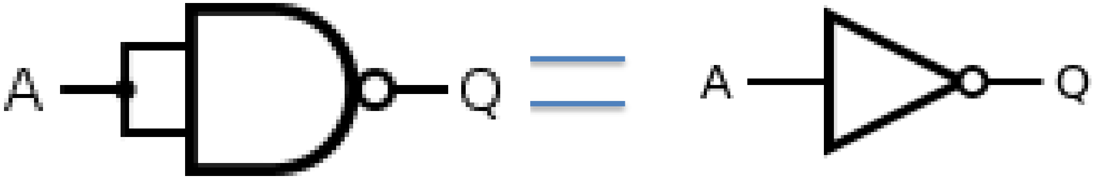
Universal gates

- **NOT** gate Using **NANDs** gate only

  - ✓ If we tie the inputs of a NAND together then we limit the possible input combinations to two, 1 1 and 0 0.

  - ✓ These are shown on the table, if the input is 0 the output is 1 and vice versa.

(AA)’ = A’



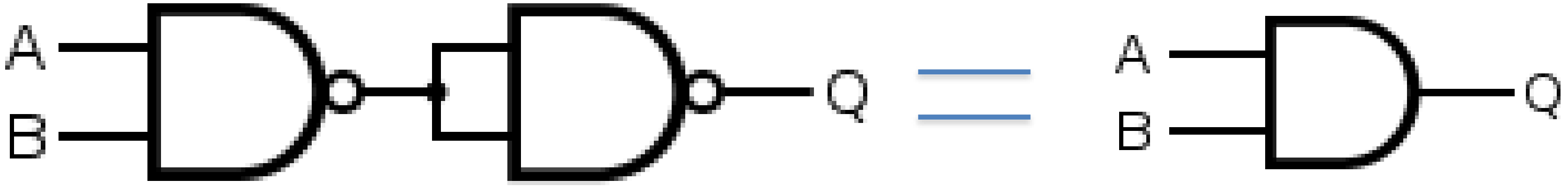
A	Q
0	1
1	0

Universal gates

- **AND** gate Using **NANDs** gate only

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

((A.B)')'= A.B



Universal gates

- **OR** Using **NAND**s Only

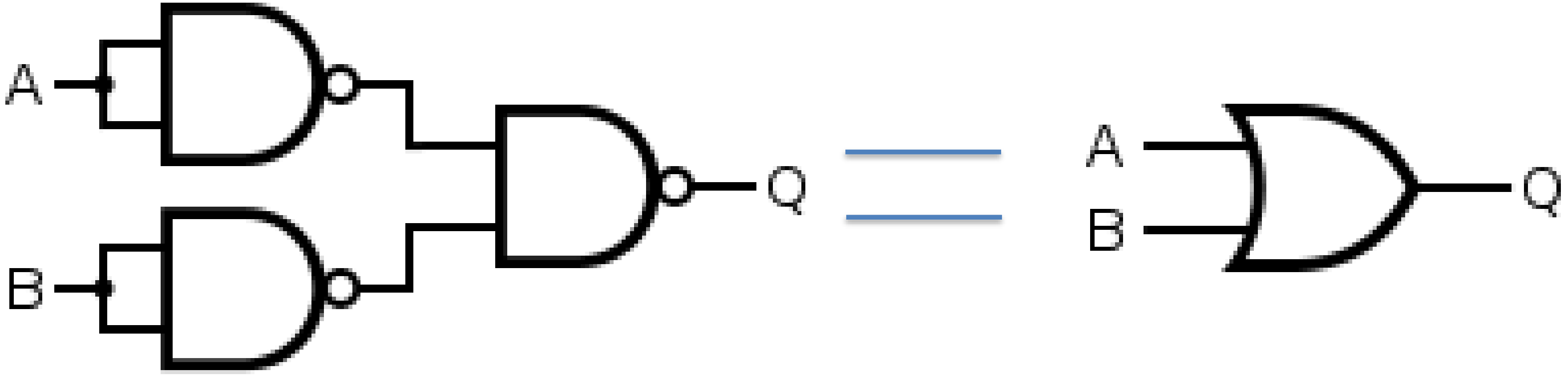
A	B	A + B	$\overline{A}$	$\overline{B}$	$\overline{A}\cdot\overline{B}$	$\overline{\overline{A}\cdot\overline{B}}$
0	0	0	1	1	1	0
0	1	1	1	0	0	1
1	0	1	0	1	0	1
1	1	1	0	0	0	1

✓ If we invert our result we see that the 3rd and 7th column are identical.

Universal gates

- **OR** Using **NAND**s Only

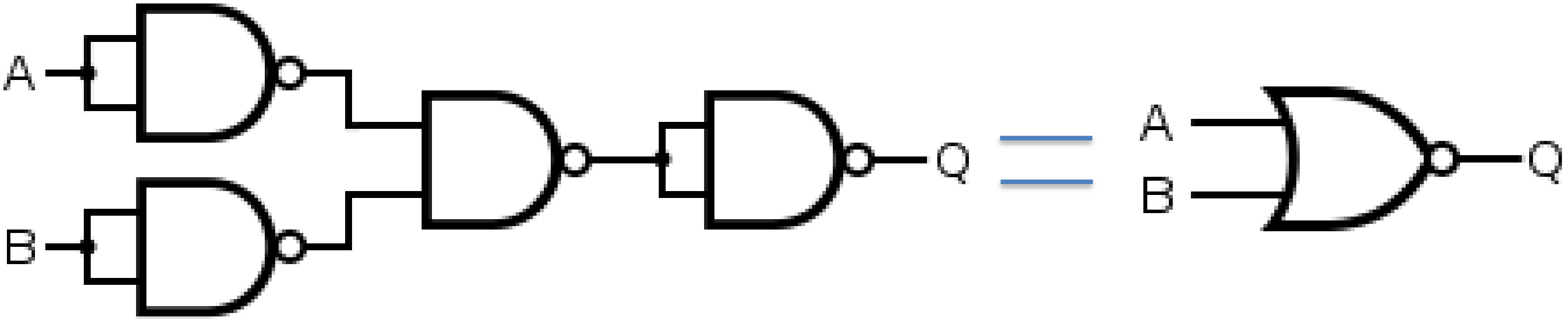
(A’B’)’= A+B



Universal gates

- **NOR** Using **NAND**s Only

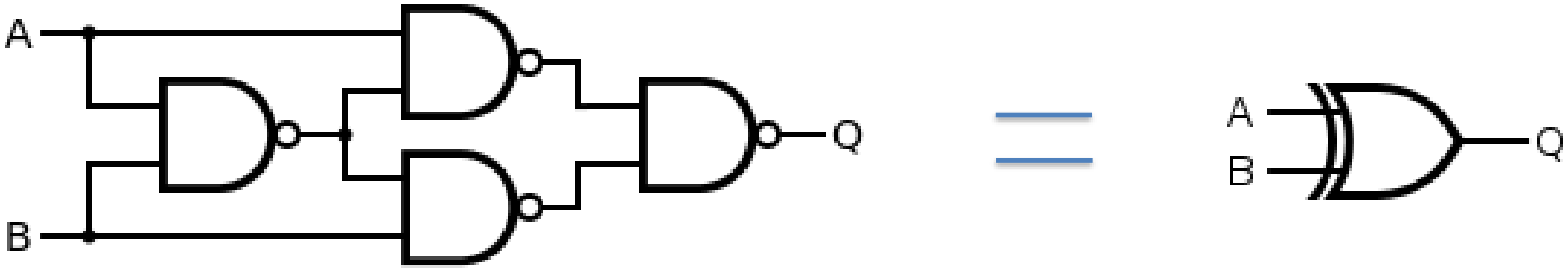
(A'B')')'= A+B



Universal gates

- **XOR** Using **NAND**s Only

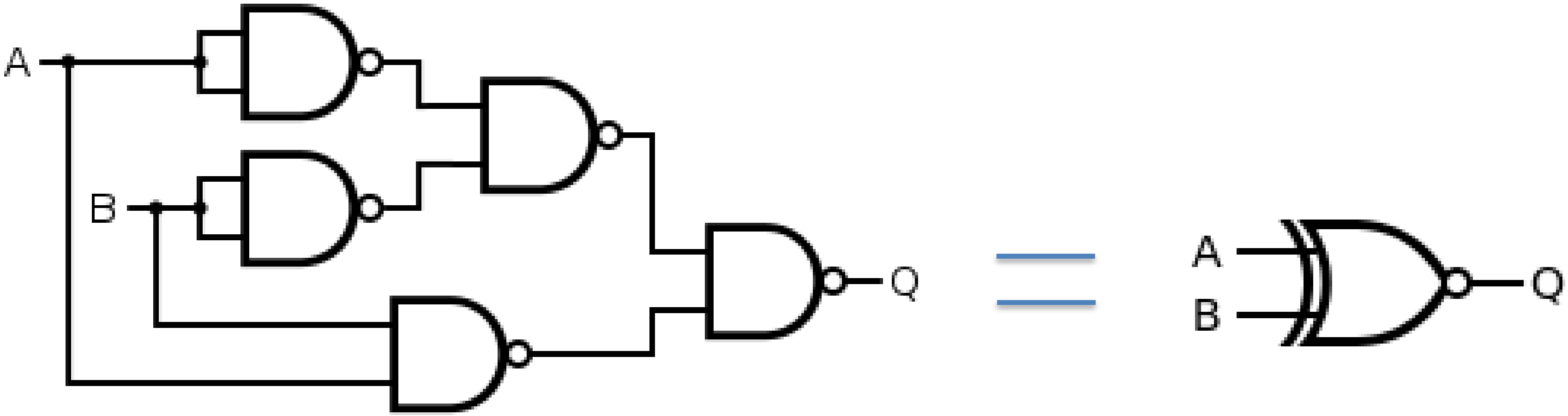
$$((A.(A.B)')'(B.(A.B)')')'= A\oplus C$$



Universal gates

- **XNOR** Using **NAND**s Only

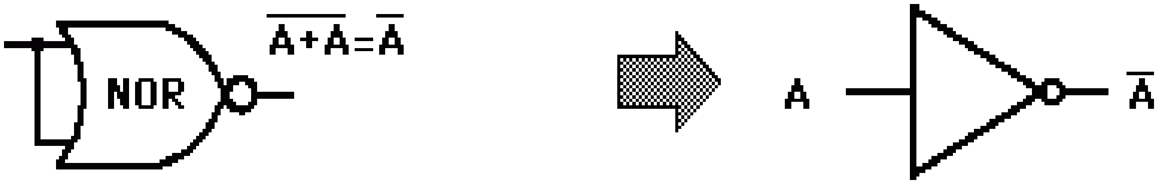
$$((A'.B')'(A.B)')'= A\otimes C$$



Universal gates

- **NOT** gate Using **NORs** gate only

$(A+A)' = A'$

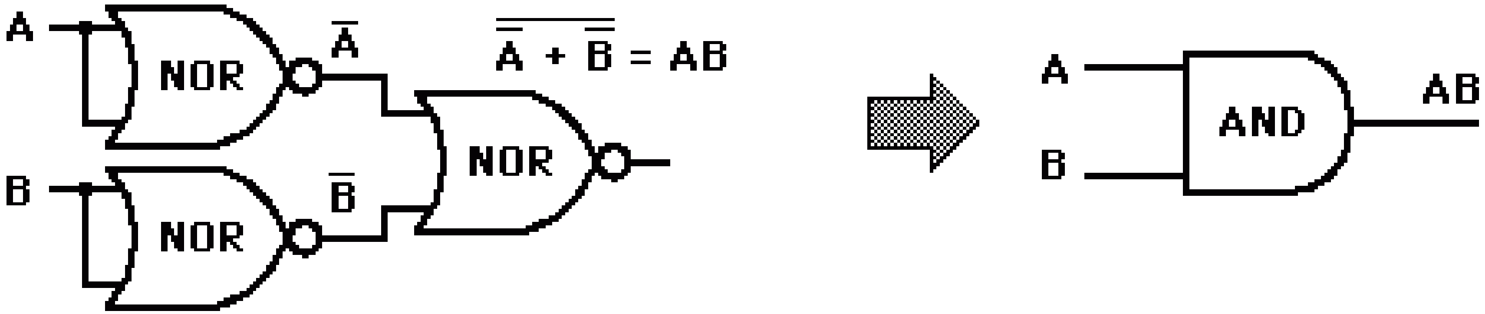


A	Y
0	1
1	0

Universal gates

- **AND** gate Using **NORs** gate only

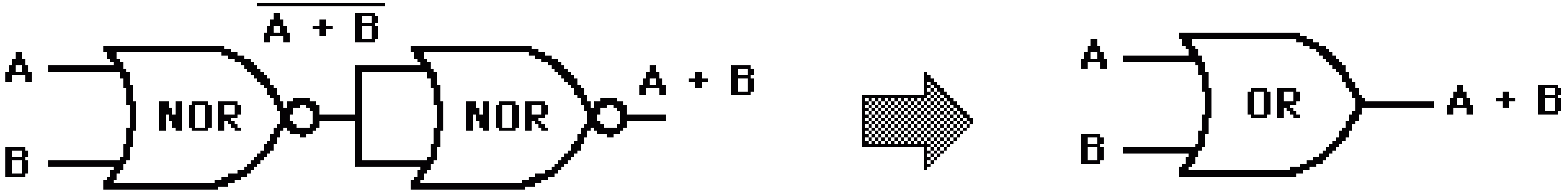
(A'+B')'= A.B



Universal gates

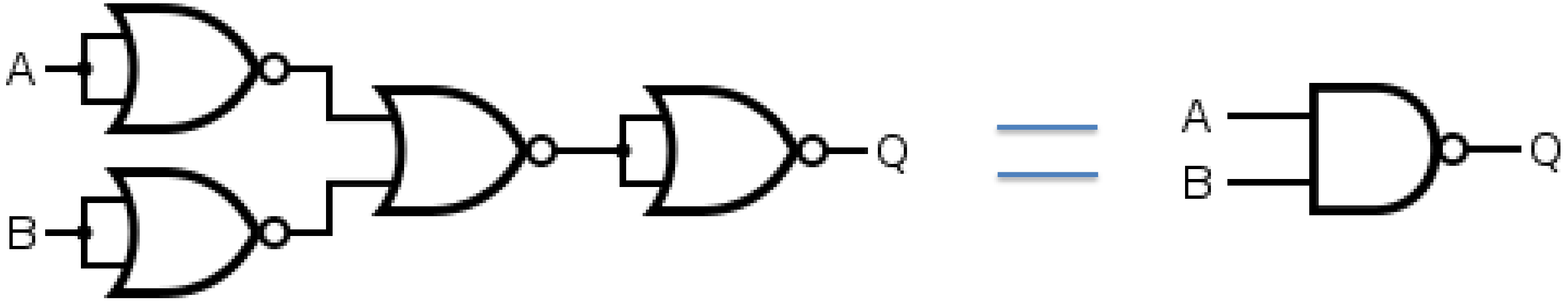
- **OR** Using **NOR**s Only

((A+B)’)’= A+B



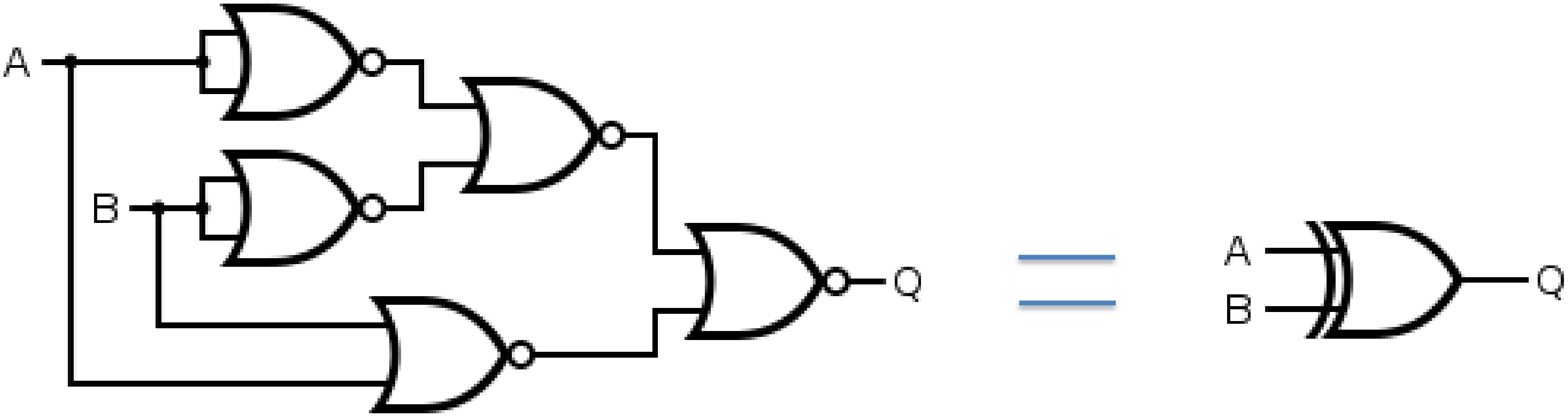
Universal gates

- **NAND** Using **NOR**s Only



Universal gates

- **XOR** Using **NOR**s Only



Universal gates

- **XNOR** Using **NOR**s Only