



UNIVERSITI
TEKNOLOGI
MALAYSIA
www.utm.my

SCSR1013 DIGITAL LOGIC

MODULE 3: LOGIC GATES

FACULTY OF COMPUTING



UNIVERSITI
TEKNOLOGI
MALAYSIA
www.utm.my

SCSR1013 DIGITAL LOGIC

MODULE 3: LOGIC GATES

FACULTY OF COMPUTING



- For each gates, we will emphasize on the following:

- i. Characteristics
- ii. Symbols
- iii. Operator
- iv. Boolean expression
- v. Truth Table
- vi. Timing Diagram

Outline

- NOT Gate (Inverter)
- AND Gate
- OR Gate

Basic gates

- NAND Gate
- NOR Gate

Universal gate using
2 of the basic gates

- Exclusive-OR (XOR) Gate
- Exclusive-NOR (XNOR) Gate

Arithmetic gates (also known as
universal gates as above)

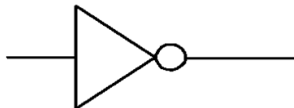
NOT Gate (Inverter)

- Characteristics**

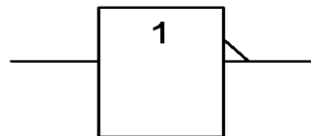
Performs inversion or complementation

- Changes a logic level to the opposite
- 0 (LOW) \rightarrow 1 (HIGH) ; 1 \rightarrow 0

Symbols :



(a) Distinctive shape symbols with negation indicators

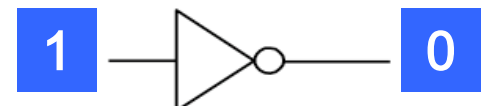
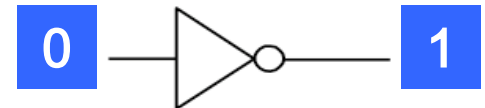


(b) Rectangular outline symbols with polarity indicators

Example :

"Output will be 0 if the input is 1"

"Output will be 1 if the input is 0"

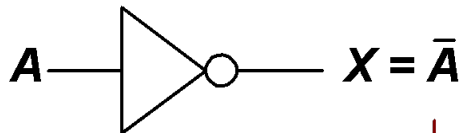


Continue...


- **Operator**

- NOT Gate is represented by overbar

- Logic expression



X is the complement of A
 X is the inverse of A
 X is NOT A

\bar{A} 
"A bar"
"not A"

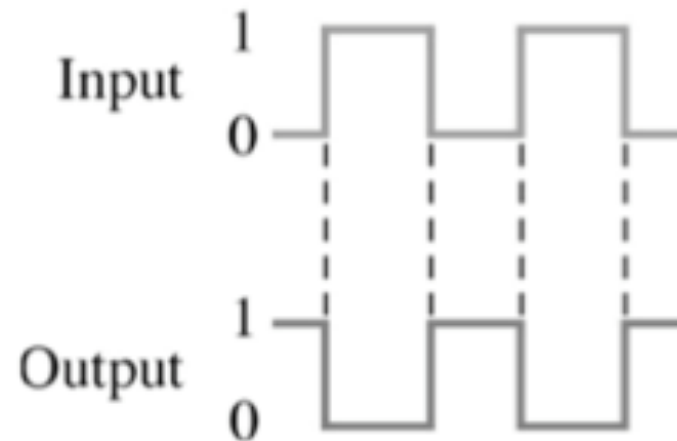
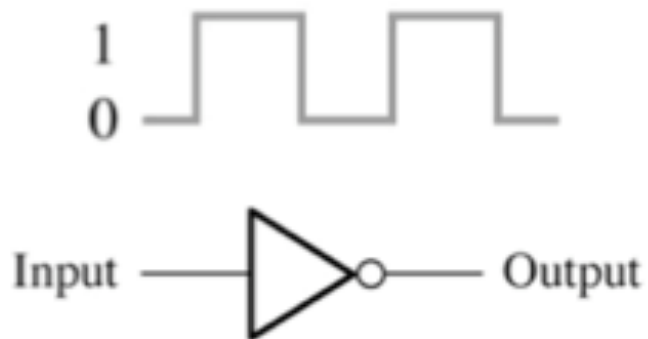
$X = \bar{A}$ or $X = A'$

Continue...

- Truth table

A	X
0	1
1	0

- Timing diagrams

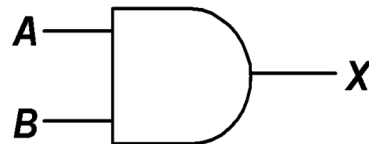


AND Gate

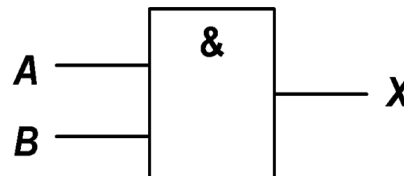
- **Characteristics**

- Performs 'logical multiplication'
 - If all of the input are HIGH, then the output is HIGH.
 - If any of the input are LOW, then the output is LOW.
 - *AND gate must at least have two (2) INPUTs, and must always have 1 (one) OUTPUT. The AND gate can have more than two INPUTs*

- **Symbols**



(a) Distinctive shape



(b) Rectangular outline with the AND (&) qualifying symbol

- Operator

$$0 \cdot 0 = 0$$

$$0 \cdot 1 = 0$$

$$1 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

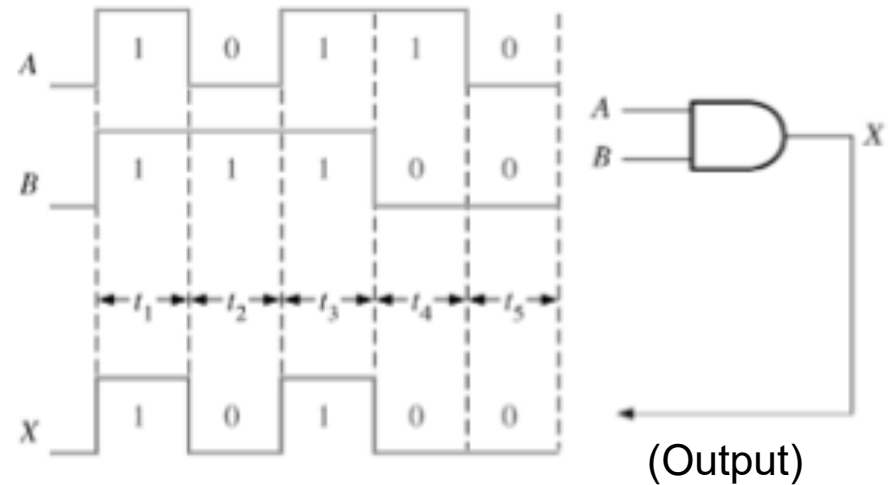
- Logic operation

$$X = A \cdot B$$

- Truth table

A	B	X = A . B
0	0	0
0	1	0
1	0	0
1	1	1

- Timing Diagram

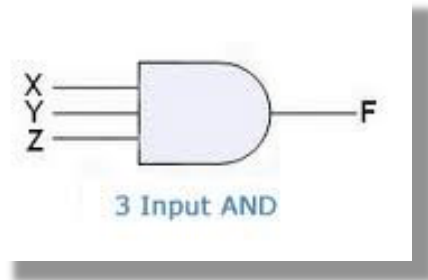


Exercise 3.1:

- Develop the truth table for a 3-input AND gate.
- Determine the total number of possible input combinations for a 4-input AND gate.

Solution :

(a) Possible input combinations = $2^3 = 8$



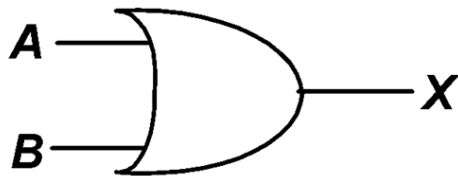
(b) Possible input combinations
= $2^4 = 16$

INPUTS			OUTPUT
X	Y	Z	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

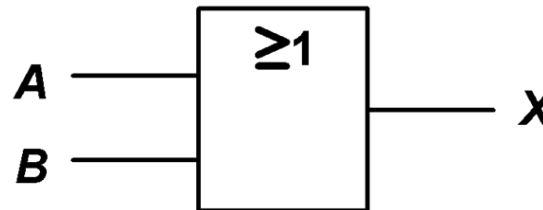
(Output 1 when all inputs 1)

- Performs '*logical addition*'
 - If any of the input are **HIGH**, then the output is **HIGH**.
 - If all of the input are **LOW**, then the output is **LOW**

- Symbols used:



(a) Distinctive shape



(b) Rectangular outline with the OR (≥ 1) qualifying symbol



Operator

OR operator can be represented by a plus sign “+”

OR gate performs Boolean addition

Boolean addition follows the basic rules as follows:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 1$$

Logical Expression

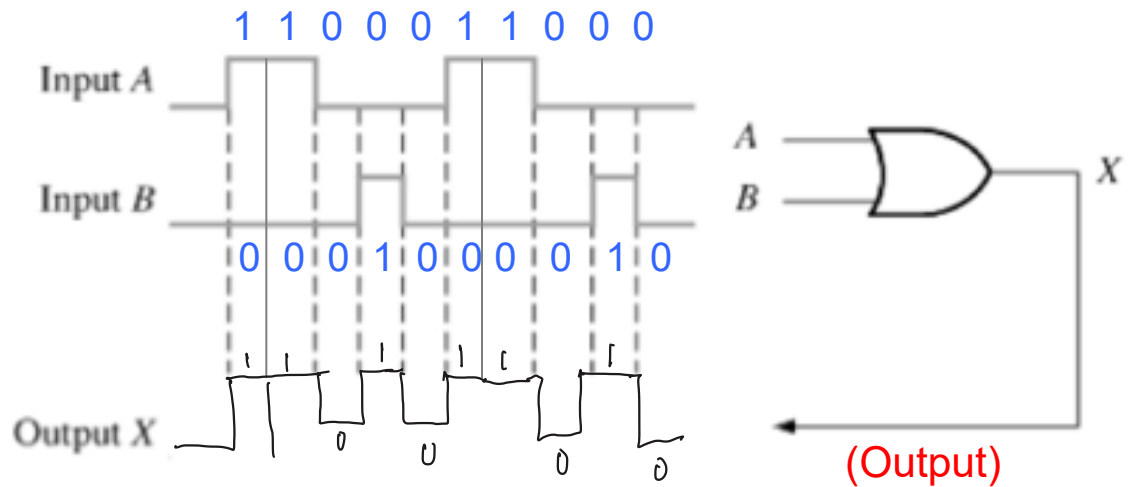
$$X = A + B$$

Continue...

Truth Table

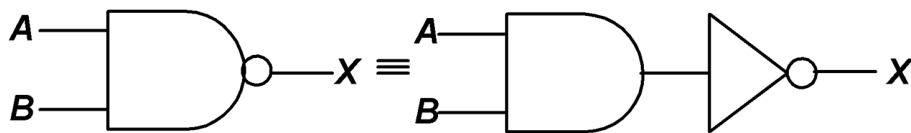
A	B	$X = A + B$
0	0	0
0	1	1
1	0	1
1	1	1

Timing Diagram

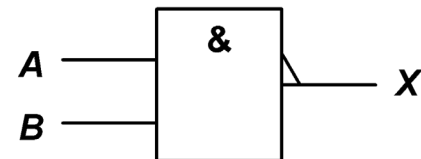


Exercise:
Draw the output waveform.

- **NAND** \equiv NOT-AND
 \Rightarrow combines the **AND** gate and an **inverter (NOT gate)**
- Used as a universal gate
 - Combinations of NAND gates can be used to perform AND, OR and inverter operations
 - If all or any of the input are **LOW**, then the output is **HIGH**.
 - If all of the input are **HIGH**, then the output is **LOW**
- Symbol used:



(a) Distinctive shape: 2 input NAND gate and its NOT/AND equivalent



(b) Rectangular outline: 2 input NAND gate with polarity indicator

Operator

NAND operator can be represented by a dot with an overbar symbol

$$\overline{0.0} = \overline{0} = 1$$

$$\overline{0.1} = \overline{0} = 1$$

$$\overline{1.0} = \overline{0} = 1$$

$$\overline{1.1} = \overline{1} = 0$$

Logical Expression

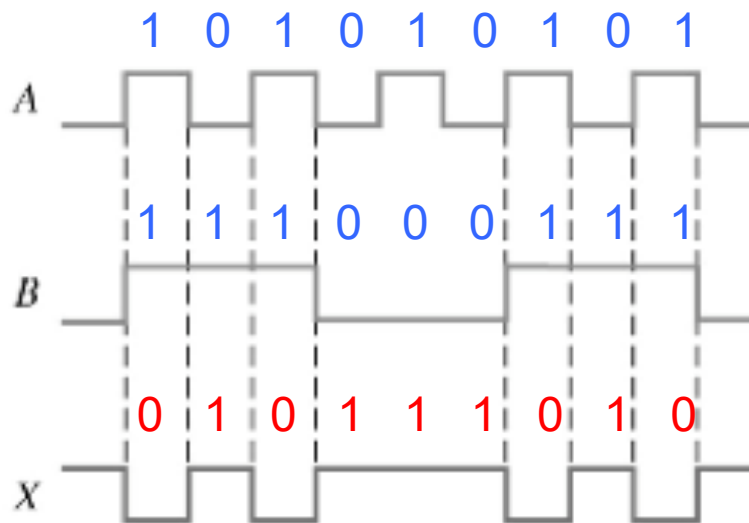
$$X = \overline{A \bullet B}$$

Truth Table

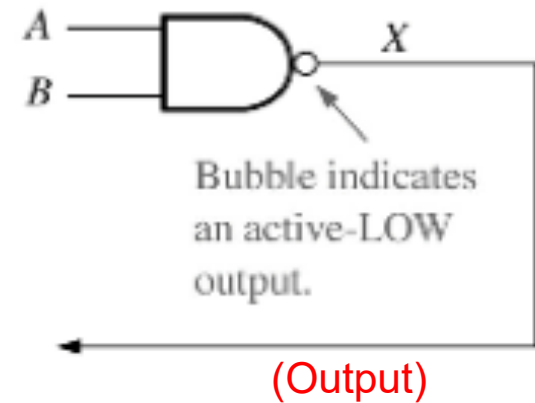
INPUTS		OUTPUT	
A	B	AB	$\overline{AB} = X$
0	0	0	$\overline{0.0} = \overline{0} = 1$
0	1	0	$\overline{0.1} = \overline{0} = 1$
1	0	0	$\overline{1.0} = \overline{0} = 1$
1	1	1	$\overline{1.1} = \overline{1} = 0$

Continue...

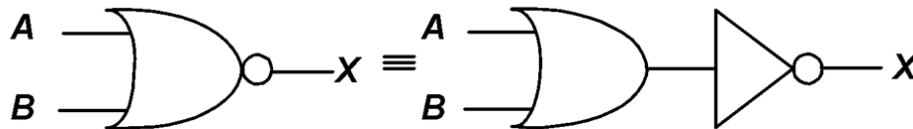
Timing Diagram



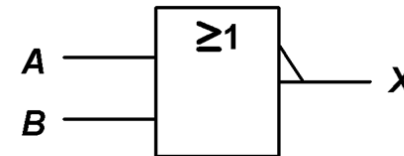
$$X = \overline{A \cdot B}$$



- **NOR** \equiv NOT-OR
 \Rightarrow combines the **OR** gate and an **inverter**
- Used as a universal gate
 - Combinations of NOR gates can be used to perform **AND**, **OR** and **inverter** operations
 - If all or any of the input are **HIGH**, then the output is **LOW**.
 - If all of the input are **LOW**, then the output is **HIGH**
- Symbol used:



(a) Distinctive shape: 2 input NOR gate and its NOT/OR equivalent



(b) Rectangular outline with the OR (≥ 1) qualifying symbol

Continue...

Operator

NOR operator can be represented by a plus with an overbar symbols

$$\overline{0+0} = \overline{0} = 1$$

$$\overline{0+1} = \overline{1} = 0$$

$$\overline{1+0} = \overline{1} = 0$$

$$\overline{1+1} = \overline{1} = 0$$

Logical Expression

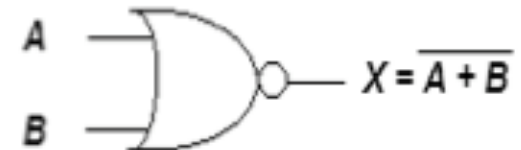
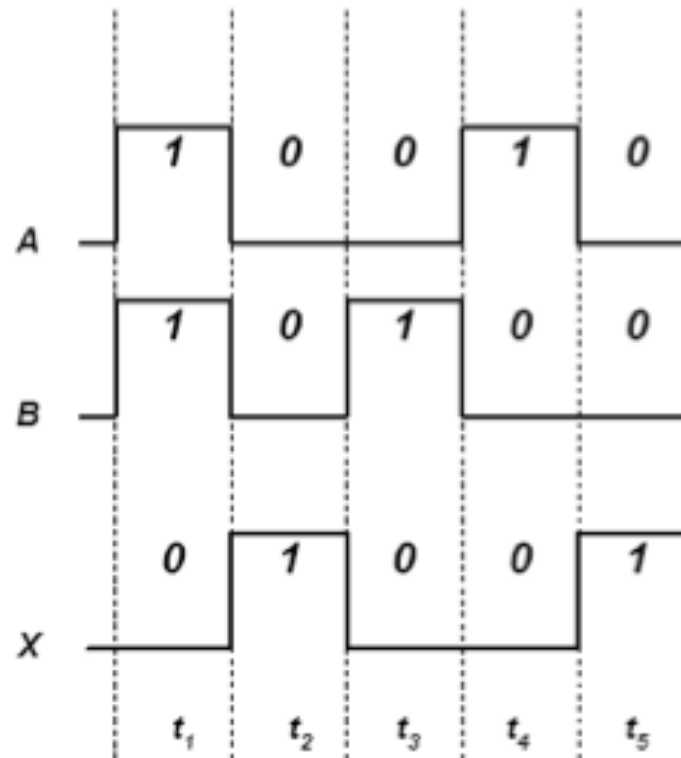
$$X = \overline{A+B}$$

Truth Table

INPUTS		OUTPUT	
A	B	A + B	$\overline{A+B} = X$
0	0	0	$\overline{0+0} = \overline{0} = 1$
0	1	1	$\overline{0+1} = \overline{1} = 0$
1	0	1	$\overline{1+0} = \overline{1} = 0$
1	1	1	$\overline{1+1} = \overline{1} = 0$

Continue...

Timing Diagram

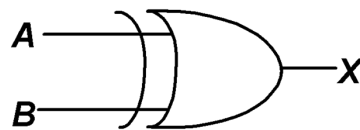


XOR Gate

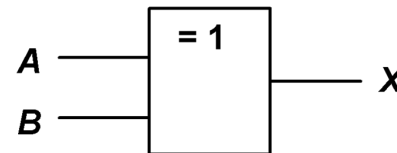
- **Characteristics**

- Combines basic logic circuits of AND, OR and Inverter.
- Has only 2 inputs
- Used as a universal gate
 - Can be connected to form an adder that allows a computer to do perform addition, subtraction, multiplication and division in ALU (Arithmetic and Logic Unit).
 - If both of the input are at the same logic level, then the output is LOW.
 - If both of the input are at opposite logic levels, then the output is HIGH

- **Symbol**



(a) Distinctive shape

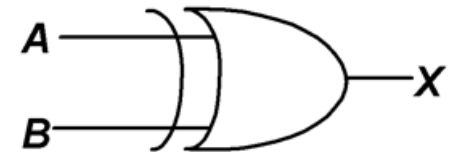


(b) Rectangular outline

Continue...

Operator

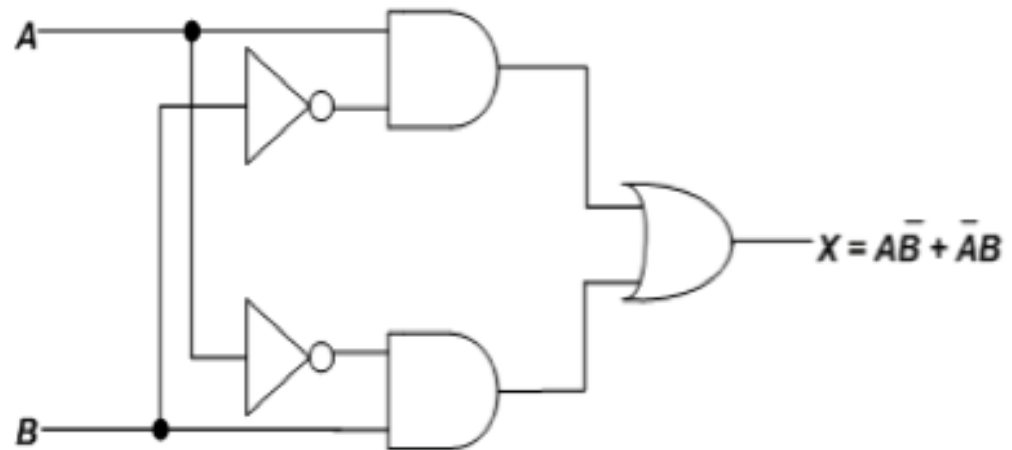
XOR operator can be represented by a \oplus symbol



Logical Expression

$$X = A \oplus B$$

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



XOR is a combination of 2 AND gates, 1 OR gate, and 2 NOT gates (inverter)

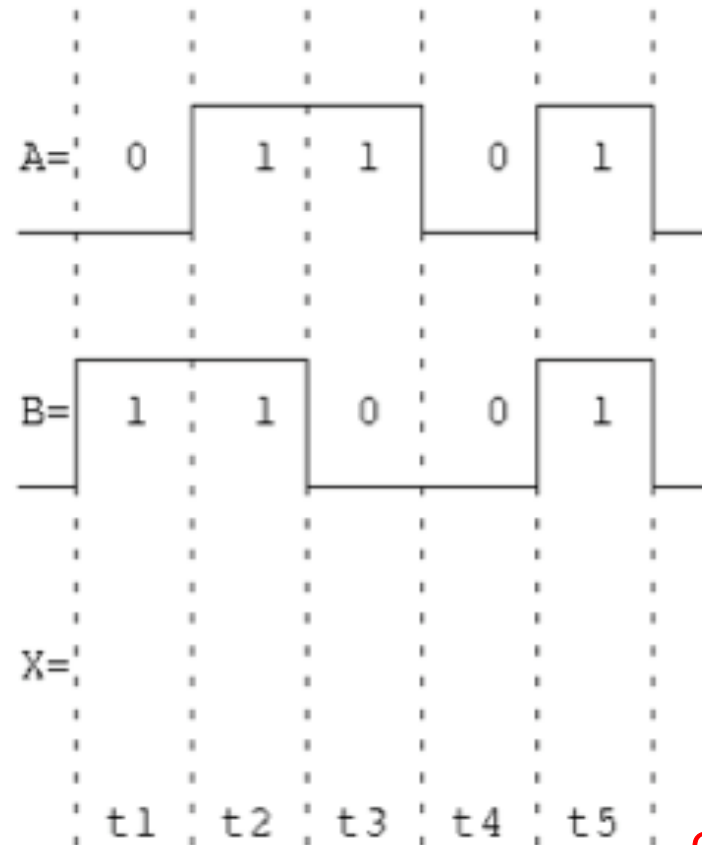
Continue...

Timing Diagram (DIY)

Truth Table

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

X-OR truth table

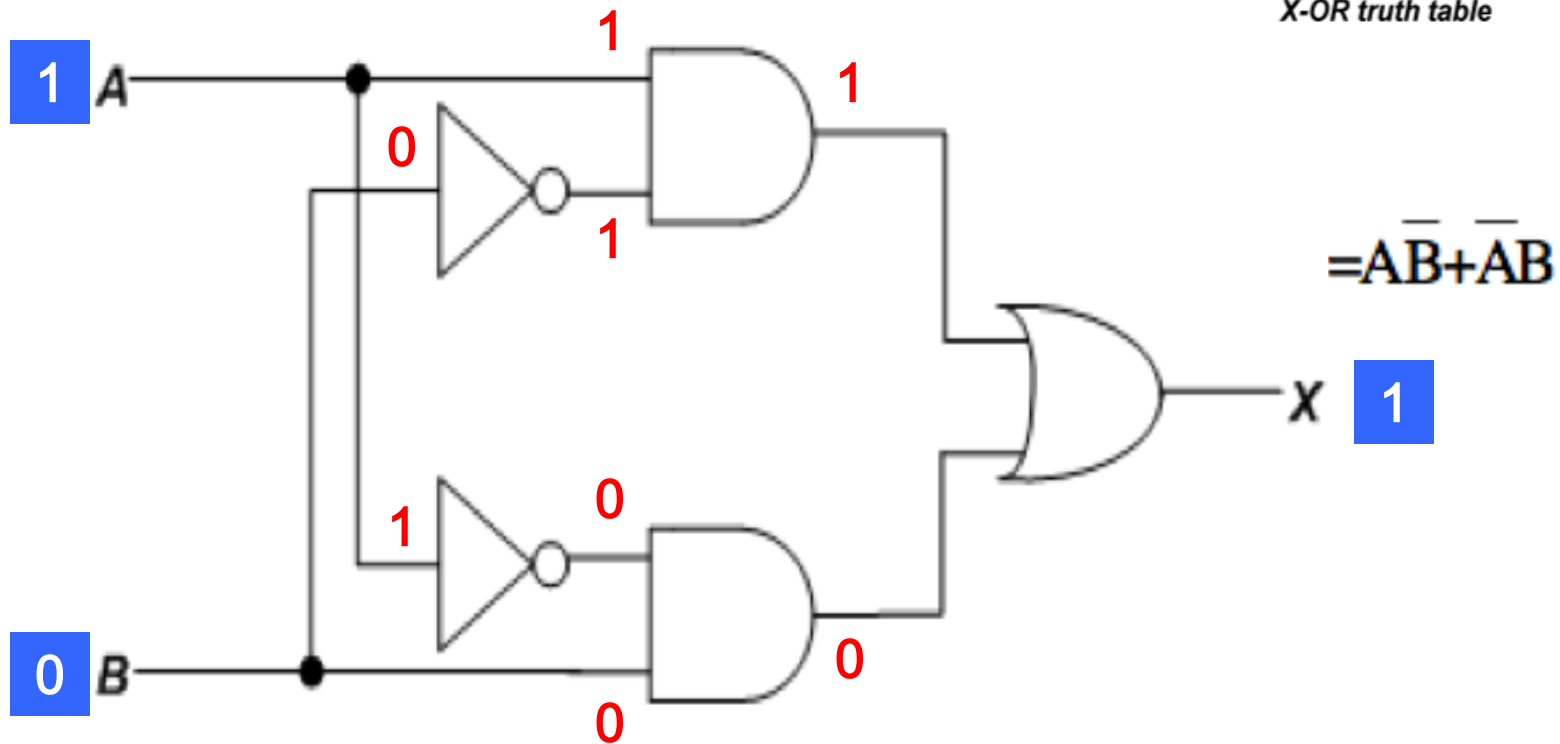


Continue...

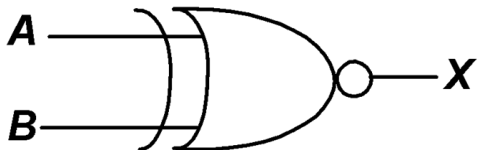
Truth Table

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

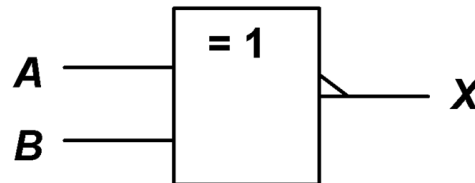
X-OR truth table



- Has only 2 inputs, but output of XNOR is the **opposite** of XOR
 - If both of the input are at the same logic level, then the output is **HIGH**.
 - If both of the input are at opposite logic levels, then the output is **LOW**.
- Symbol used:



(a) Distinctive shape



(b) Rectangular outline

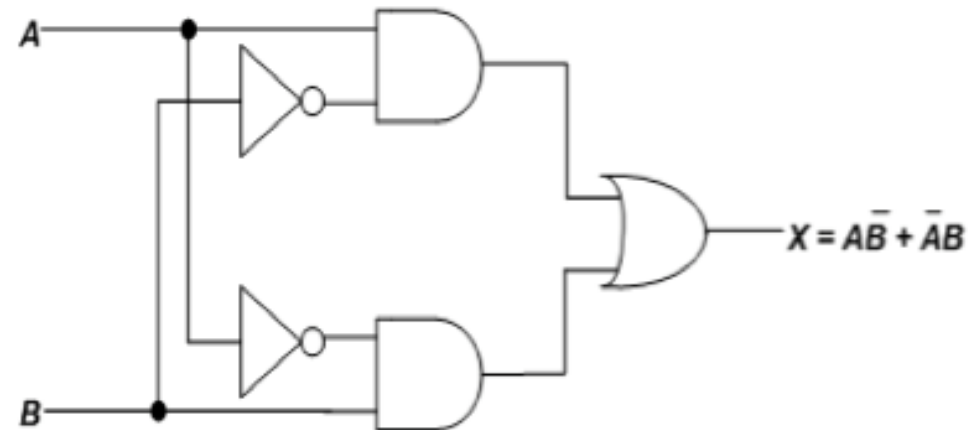
Continue...

Operator

XNOR operator can be represented by a \odot symbol

Logical Expression

$$\begin{aligned} X &= A \odot B \\ &= \overline{A \oplus B} \\ &= \overline{A\bar{B} + \bar{A}B} \\ &= \overline{AB + \bar{A}\bar{B}} \end{aligned}$$



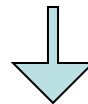
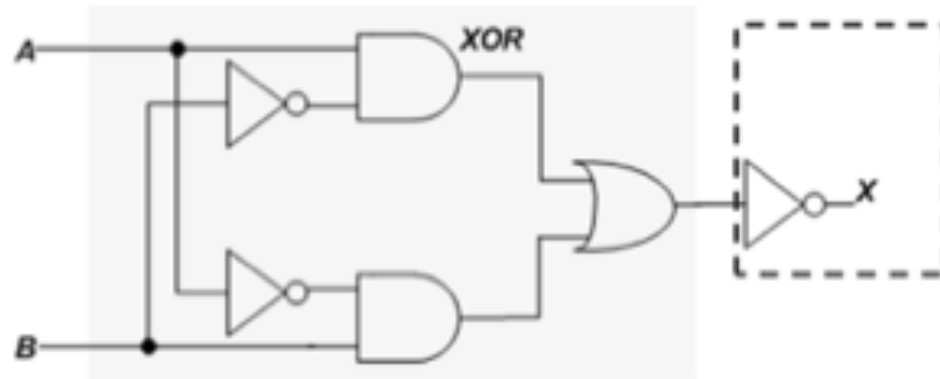
XOR is a combination of 2 AND gates, 1 OR gate and 2 inverters

Continue...

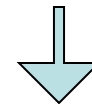
XNOR Implementation

- There are two ways in implementing XNOR Gate:
 - The first implementation by **complementing (inverting)** XOR circuit

1



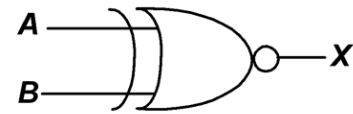
$$X = A\bar{B} + \bar{A}B$$



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

Continue...

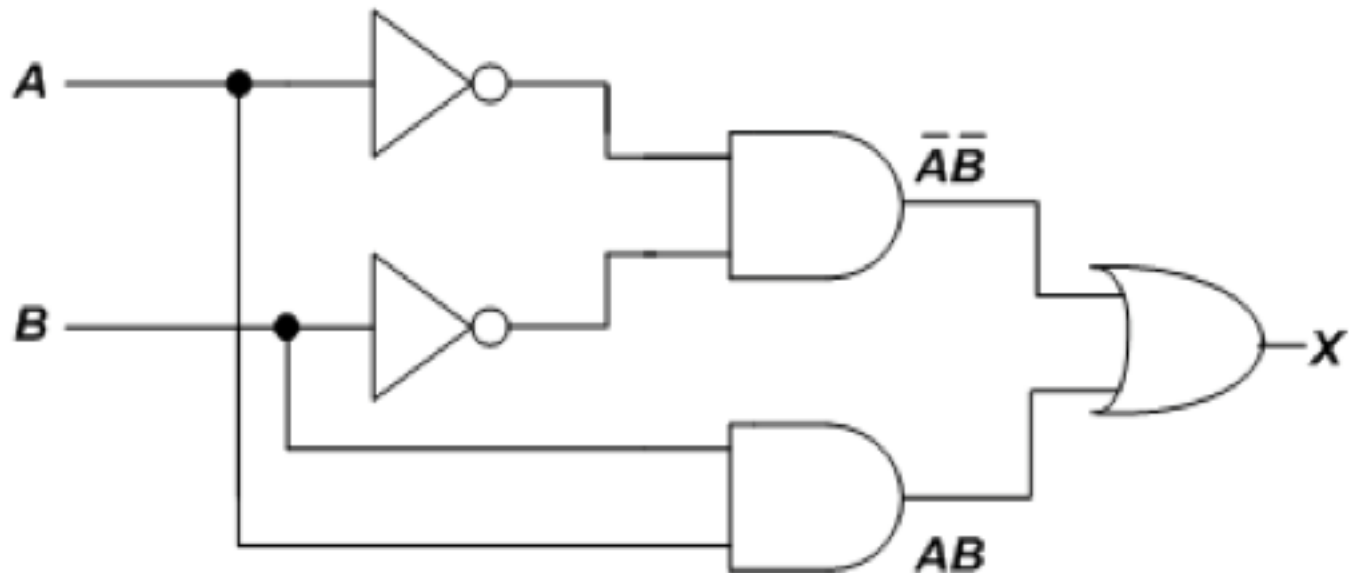
XNOR Implementation



Second implementation is by directly implementing the following expression:

2

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



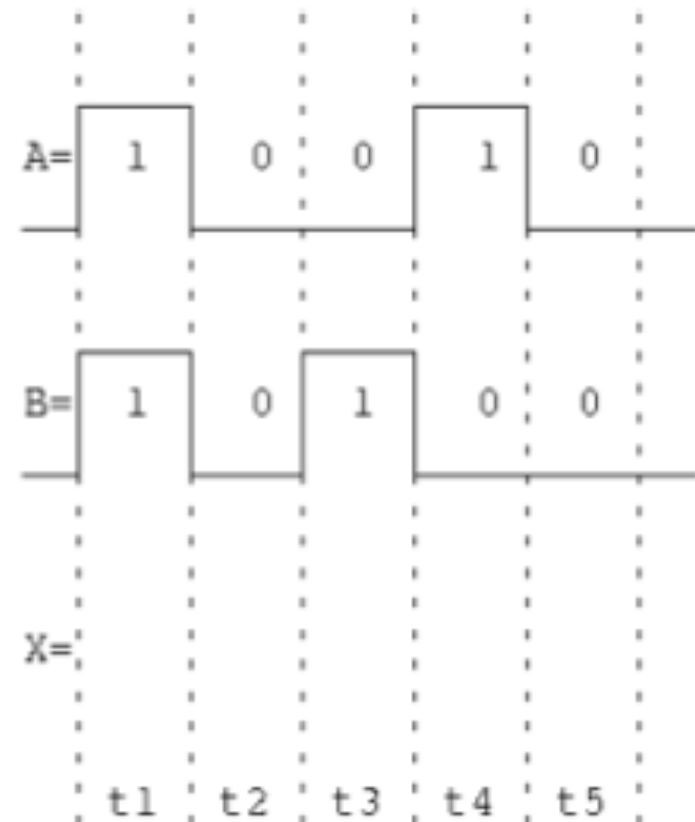
Continue...

Truth Table

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

$$X = AB + \overline{AB}$$

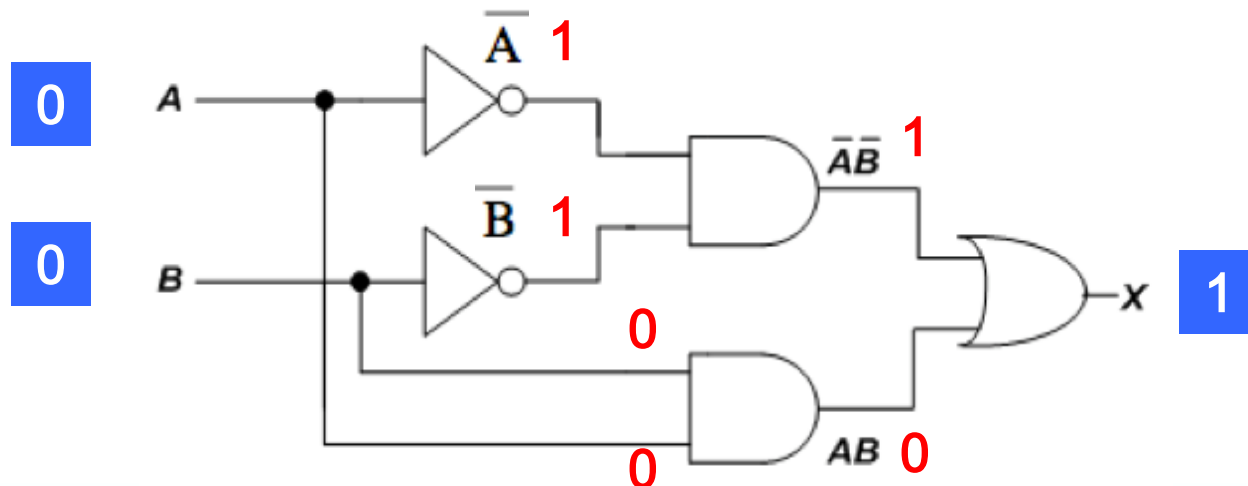
Timing Diagram










The characteristic of the above circuit can be proven by the following truth table

Example :

A	B	AB	$\bar{A} \bar{B}$	$X = AB + \bar{A} \bar{B}$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1



Summary

	NAND	Opposite of AND
	OR	Either is true (or both)
	XOR	Exactly one is true
	AND	Both are true
	NOT	Reverses the input
	NOR	Opposite of OR
	XNOR	Opposite of XOR