

The background is a blue gradient with white circuit-like lines and circles in the corners. The text is centered in a white serif font.

# Chapter No:3

## Logic Gates

# *CHAPTER OUTLINE*

- The Inverter
- The AND Gate
- The OR Gate
- The NAND Gate
- The NOR Gate
- The Exclusive-OR and Exclusive-NOR Gates

# NOT OPERATION

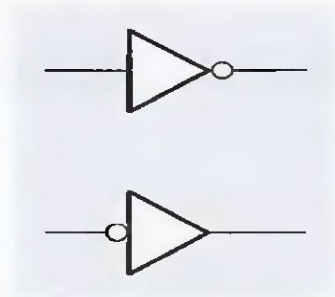
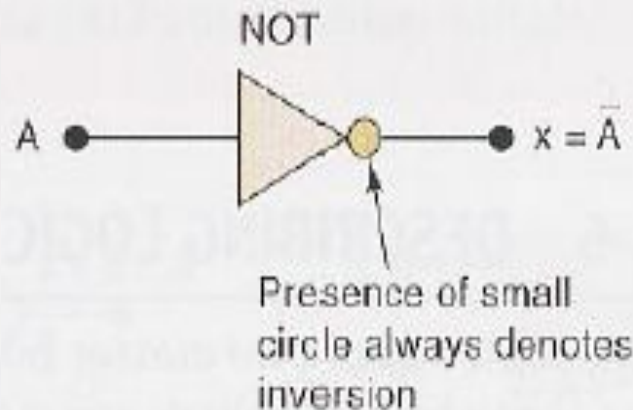
Boolean Expression :

$$x = \bar{A}$$

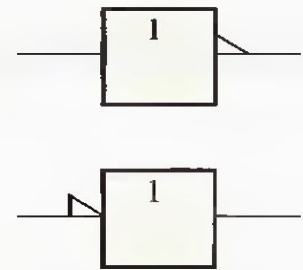
$0 = \bar{1}$  because 0 is not 1

$1 = \bar{0}$  because 1 is not 0

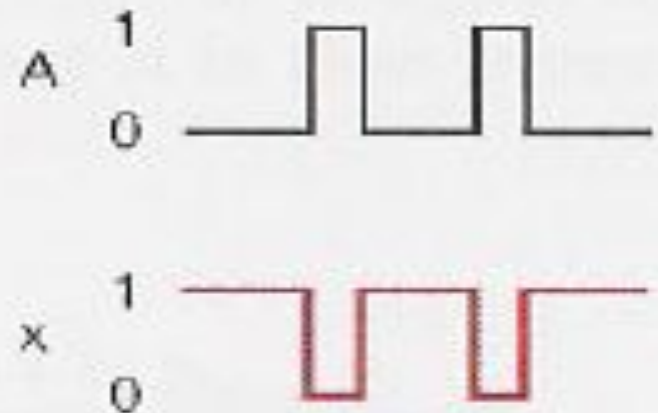
NOT		
A		$x = \bar{A}$
0		1
1		0



(a) Distinctive shape symbols with negation indicators

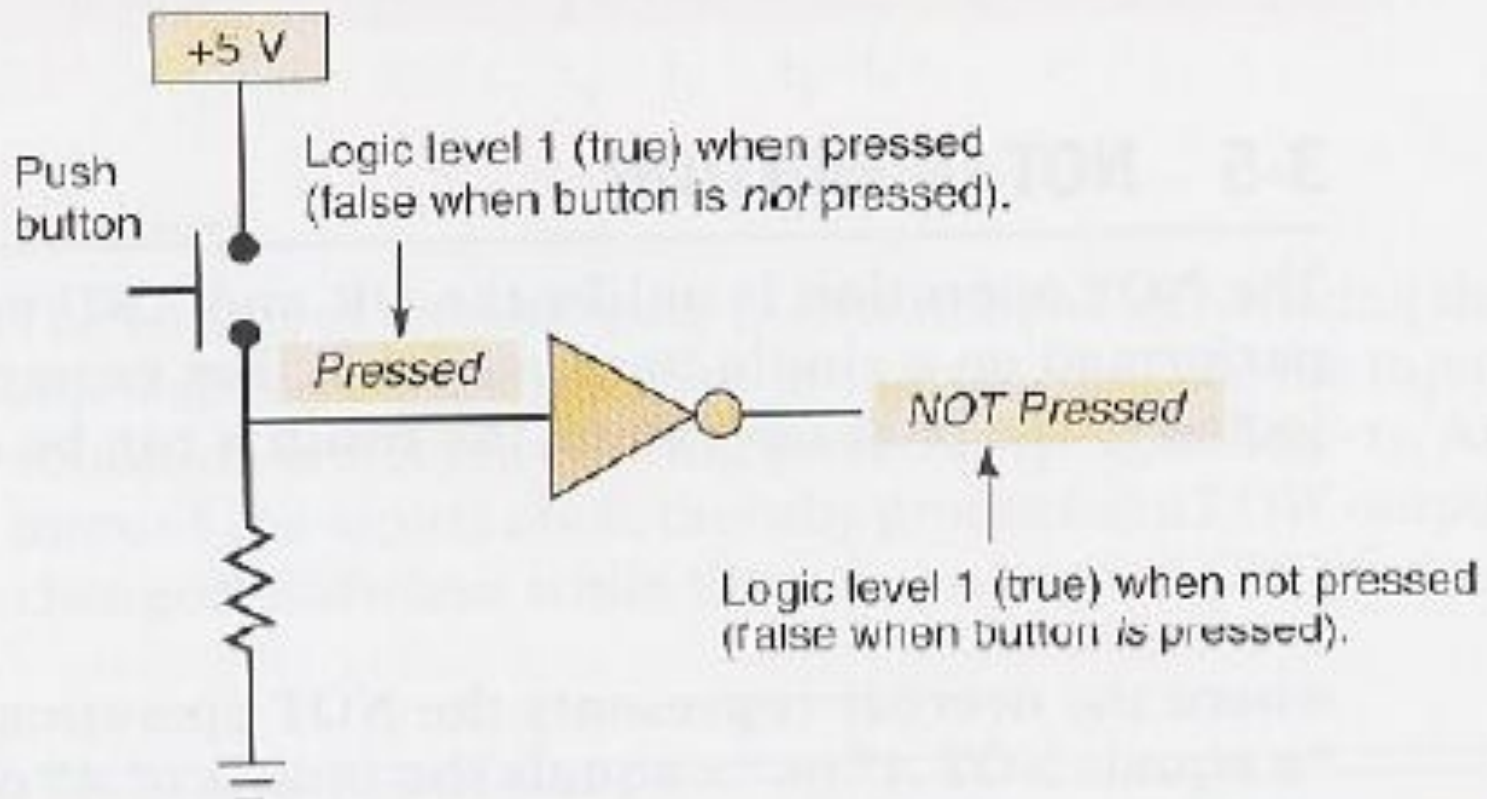


(b) Rectangular outline symbols with polarity indicators



# EXAMPLE

Figure 3-12 shows a typical application of the NOT gate. The push button is wired to produce a logic 1 (true) when it is pressed. Sometimes we want to know if the push button is not being pressed, and so this circuit provides an expression that is true when the button is not pressed.



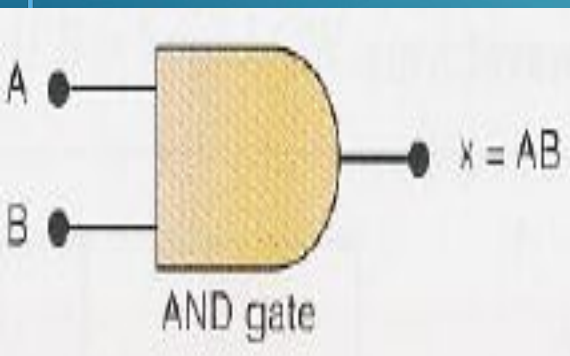
# AND OPERATION WITH AND GATES

**For a 2-input AND gate, output  $X$  is HIGH only when inputs  $A$  and  $B$  are HIGH;  $X$  is LOW when either  $A$  or  $B$  is LOW, or when both  $A$  and  $B$  are LOW.**

*The Boolean expression for AND operation :*

$$X = A.B$$

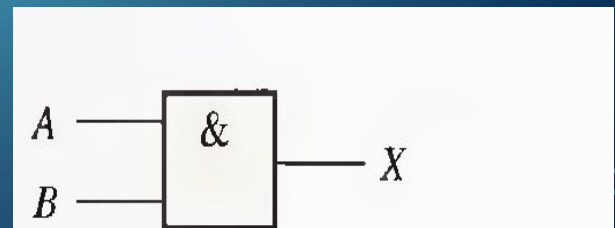
Circuit symbol



Truth Table

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

1 = HIGH, 0 = LOW



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



# Possible Combinations of binary input to a gate

The total number of possible combinations of binary inputs to a gate is determined by the following formula:

$$N = 2^n$$

where  $N$  is the number of possible input combinations and  $n$  is the number of input variables. To illustrate,

For two input variables:  $N = 2^2 = 4$  combinations

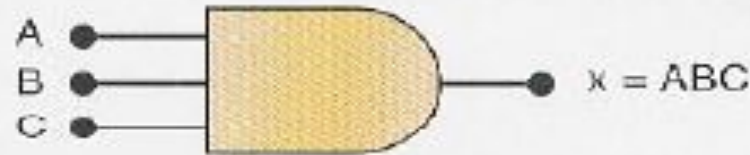
For three input variables:  $N = 2^3 = 8$  combinations

For four input variables:  $N = 2^4 = 16$  combinations

# AND OPERATION FOR THREE INPUT

$$X = A.B.C$$

A	B	C	$x = ABC$
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

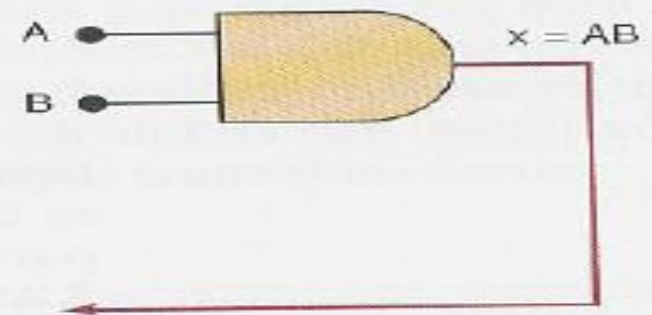
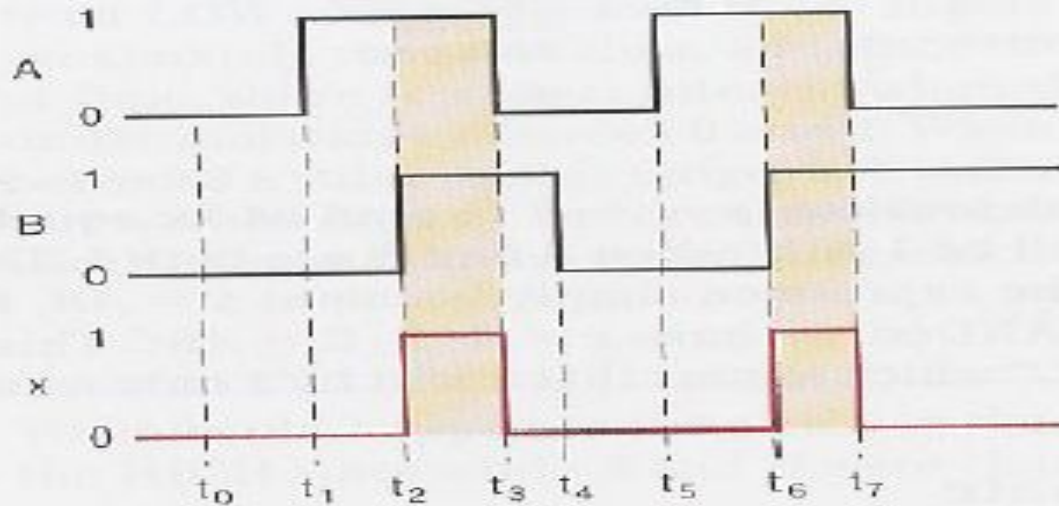


## Important point :

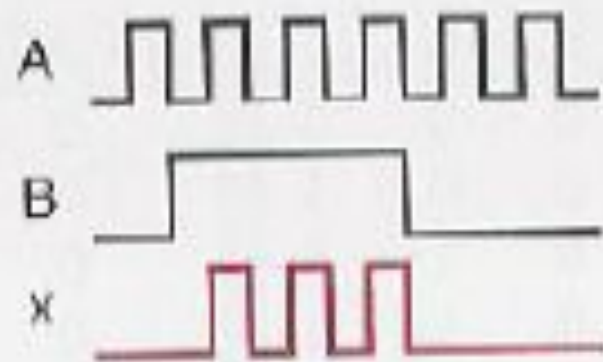
1. The AND operation is performed the same as ordinary multiplication of 1s and 0s.
2. An AND gate is a logic circuit that performs the AND operation on the circuit's inputs.
3. An AND gate output will be 1 *only* for the case when *all* inputs are 1; for all other cases, the output will be 0.
4. The expression  $x = AB$  is read as "x equals A AND B."

# EXAMPLES

Determine the output  $x$  from the AND gate in Figure 3-9 for the given input waveforms.



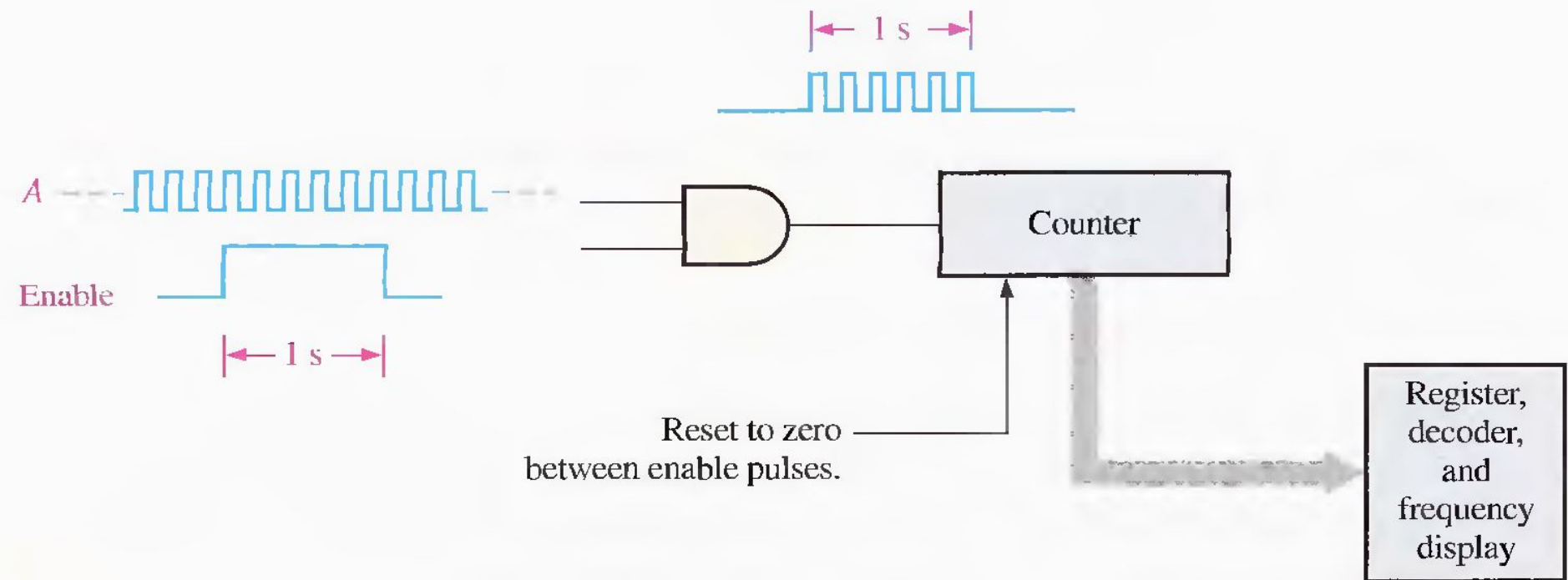
Determine the output waveform for the AND gate shown in Figure 3-10.





# Applications

## *The AND Gate as an Enable/Inhibit Device*



An AND gate performing an enable/inhibit function for a frequency counter.

# Applications

## *A Seat Belt Alarm System*

HIGH = On  
LOW = Off

HIGH = Unbuckled  
LOW = Buckled

Ignition  
switch

A

Seat  
belt

B

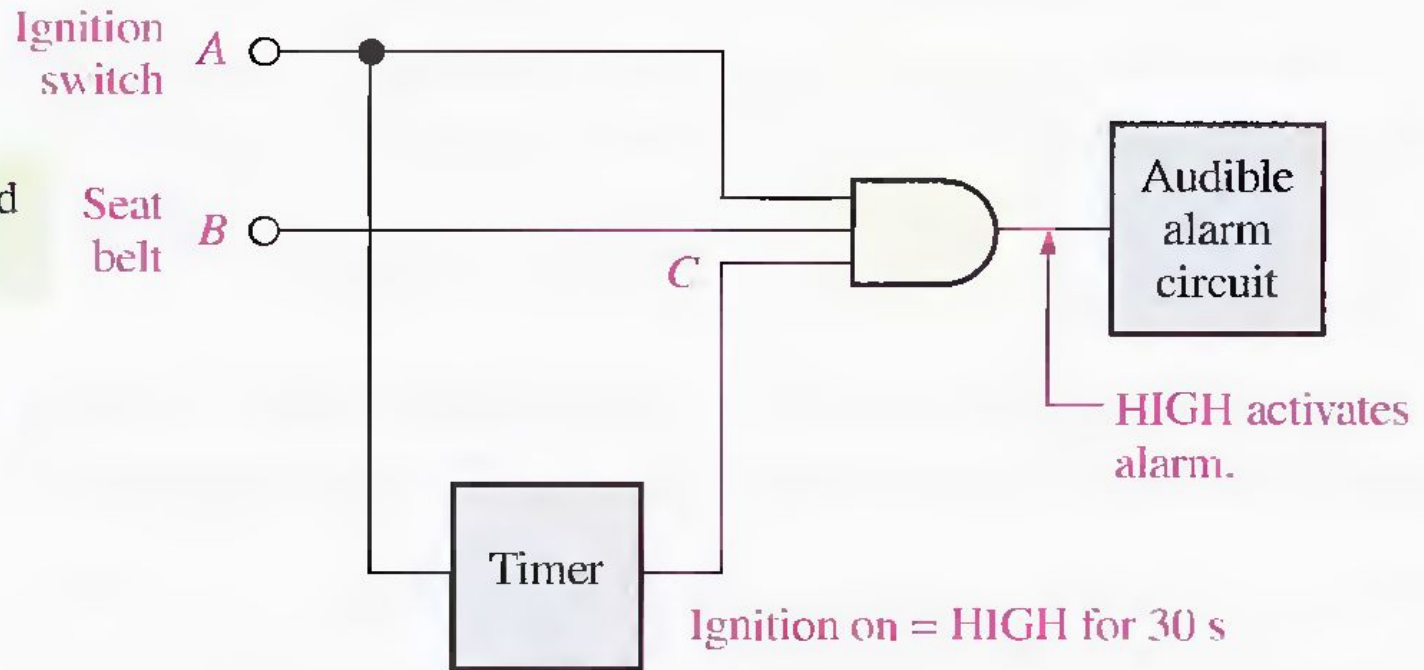
C

Timer

Audible  
alarm  
circuit

HIGH activates  
alarm.

Ignition on = HIGH for 30 s



# OR OPERATION WITH OR GATES

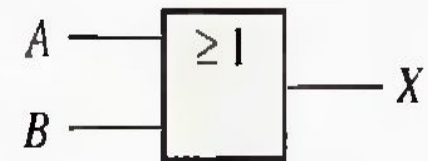
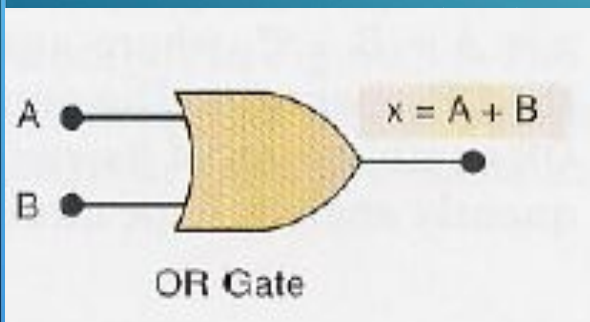
*The Boolean expression for OR operation :*

$$X = A + B$$

*The Truth Table*

INPUTS		OUTPUT
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

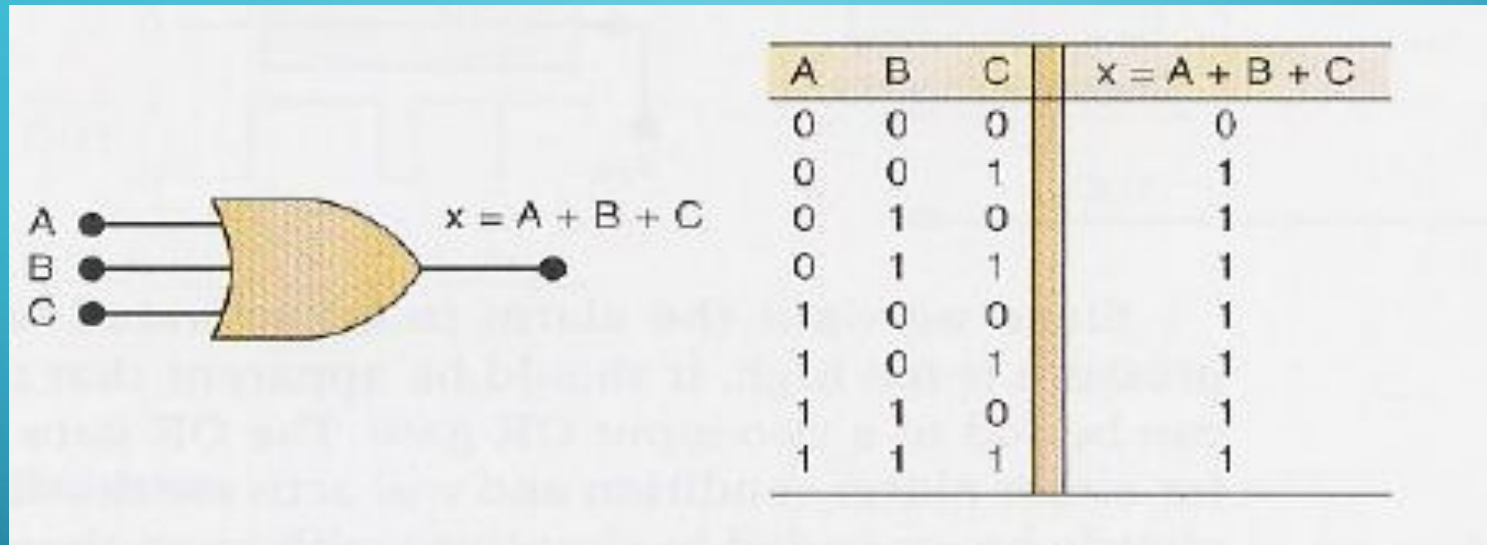
1 = HIGH, 0 = LOW



(b) Rectangular outline with the OR ( $\geq 1$ ) qualifying symbol

# OR OPERATION FOR THREE INPUT

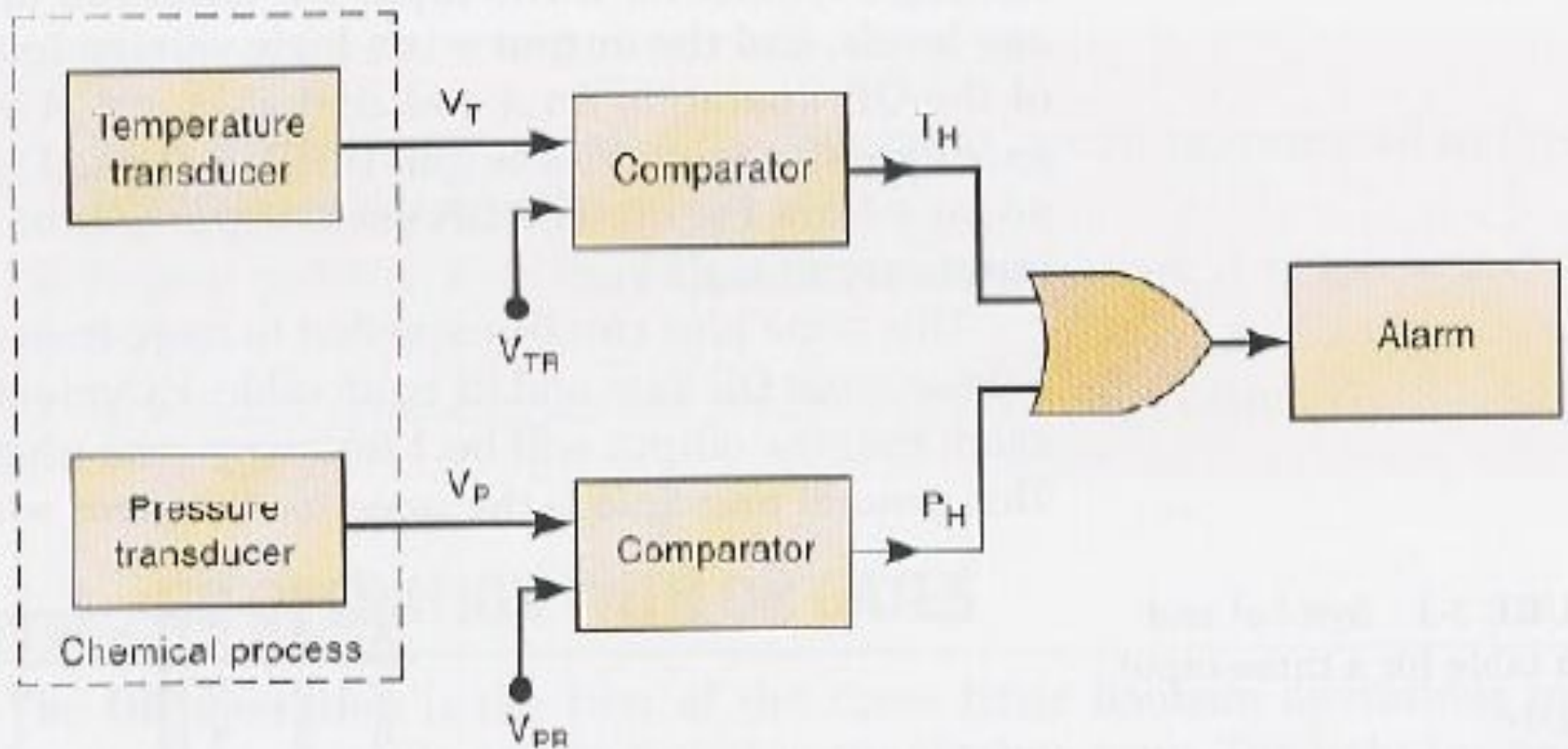
$$X = A + B + C$$



## Important point :

1. The OR operation produces a result (output) of 1 whenever *any* input is a 1. Otherwise the output is 0.
2. An OR gate is a logic circuit that performs an OR operation on the circuit's inputs.
3. The expression  $x = A + B$  is read as "x equals A OR B."

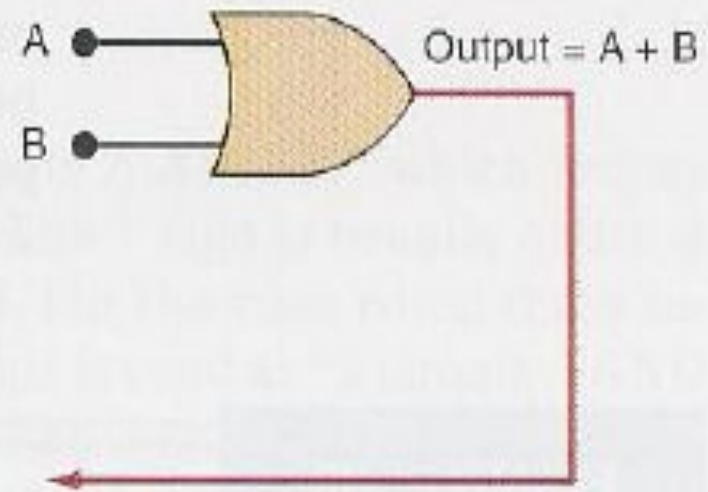
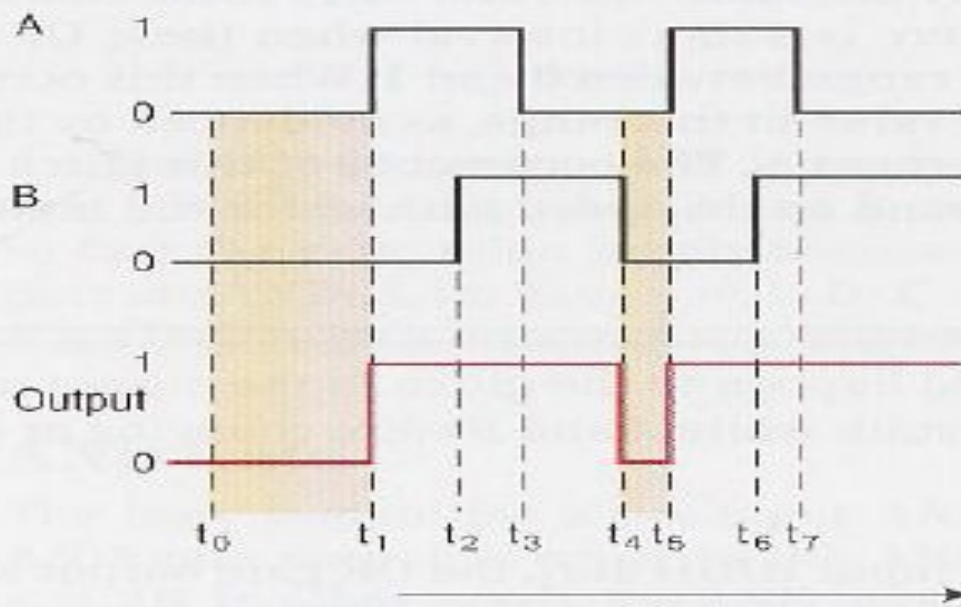
# APPLICATION OF USE OF OR GATE IN AN ALARM SYSTEM





# EXAMPLES

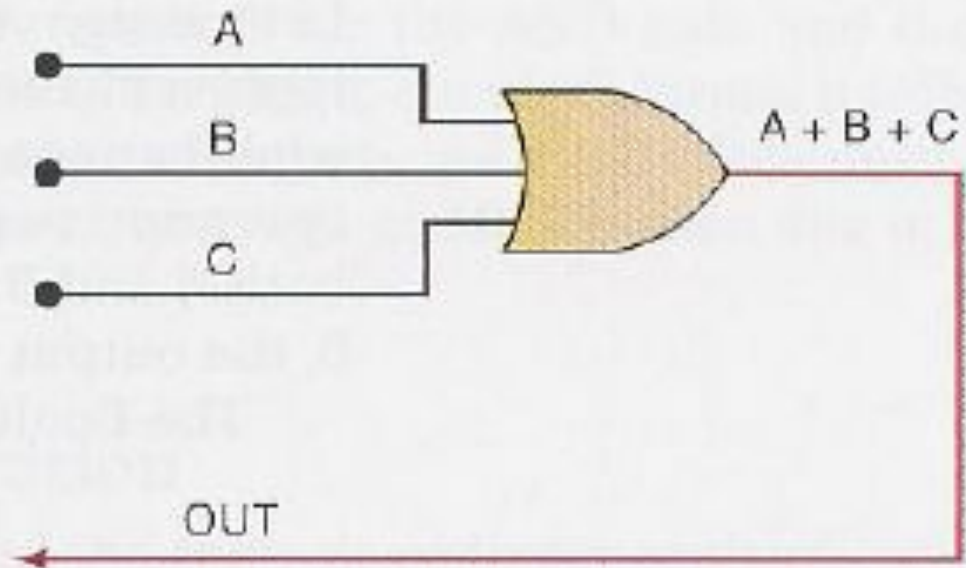
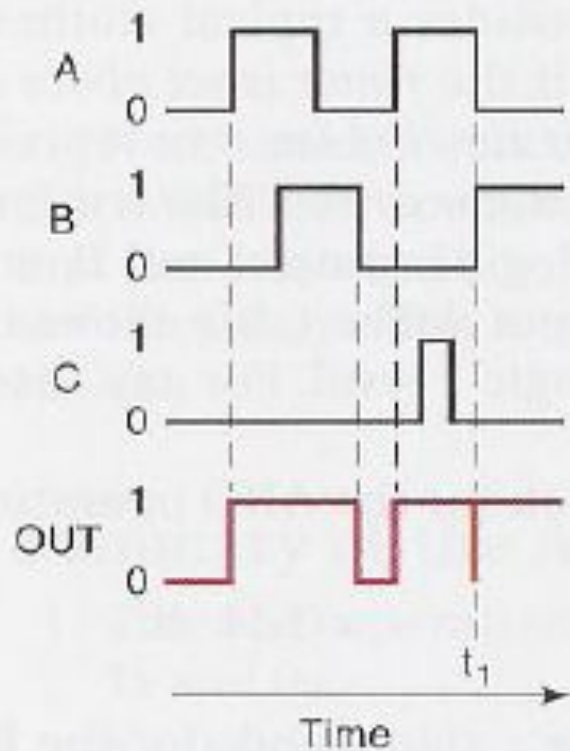
Determine the OR gate output in Figure 3-5. The OR gate inputs *A* and *B* are varying according to the timing diagrams shown. For example, *A* starts out LOW at time  $t_0$ , goes HIGH at  $t_1$ , back to LOW at  $t_3$ , and so on.



The OR gate output will be HIGH whenever *any* input is HIGH. Between time  $t_0$  and  $t_1$ , both inputs are LOW, so OUTPUT = LOW. At  $t_1$ , input *A* goes HIGH while *B* remains LOW. This causes OUTPUT to go HIGH at  $t_1$  and stay HIGH until  $t_4$  because, during this interval, one or both inputs are HIGH. At  $t_4$ , input *B* goes from 1 to 0 so that now both inputs are LOW, and this drives OUTPUT back to LOW. At  $t_5$ , *A* goes HIGH, sending OUTPUT back HIGH, where it stays for the rest of the shown time span.

# EXAMPLES

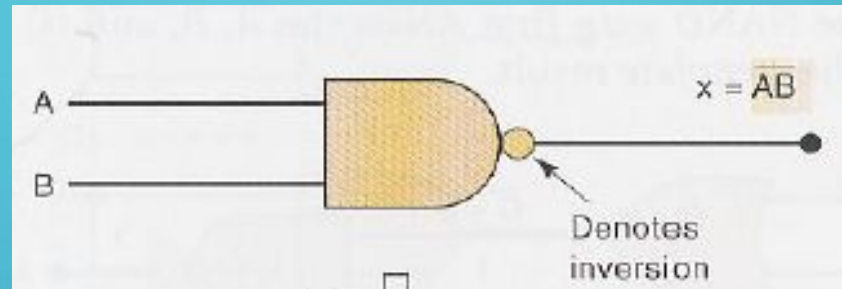
For the situation depicted in Figure 3-6, determine the waveform at the OR gate output.



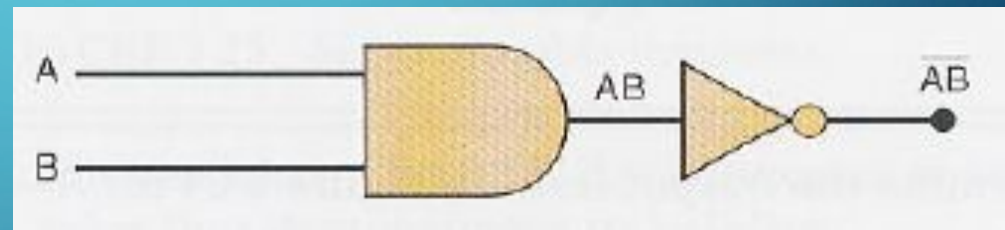
# NAND GATE:

The NAND gate is a popular logic element because it can be used as a universal gate; that is, NAND gates can be used in combination to perform the AND, OR, and inverter

*Symbol :*



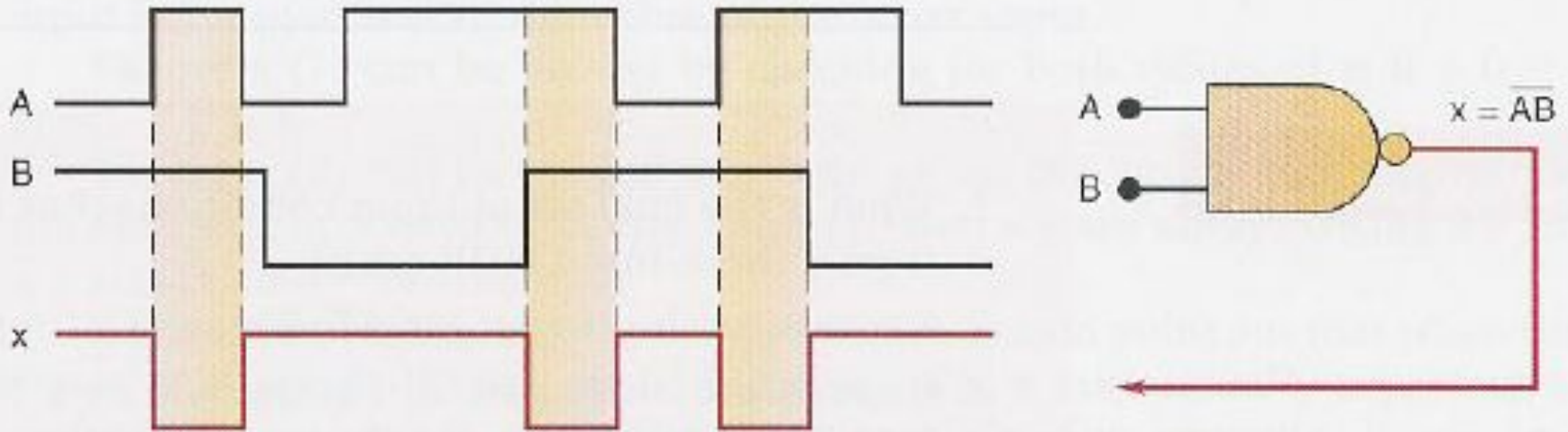
*Equivalent Circuit :*



*Truth Table :*

		AND		NAND	
A	B	AB		AB	
0	0	0		1	
0	1	0		1	
1	0	0		1	
1	1	1		0	

# WAVE FORM AT THE OUT PUT OF NAND GATE



## Operation of NAND Gate

**For a 2-input NAND gate, output  $X$  is LOW if inputs  $A$  and  $B$  are HIGH;  $X$  is HIGH if either  $A$  or  $B$  is LOW, or if both  $A$  and  $B$  are LOW.**



# NEGATIVE -OR EQUIVALENT OPERATION OF A NAND GATE

**For a 2-input NAND gate performing a negative-OR operation, output  $X$  is HIGH if either input  $A$  or input  $B$  is LOW, or if both  $A$  and  $B$  are LOW.**

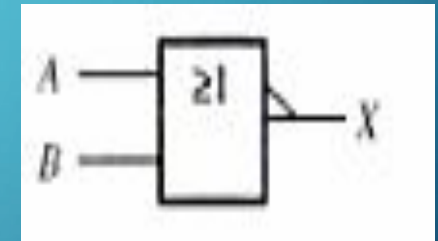
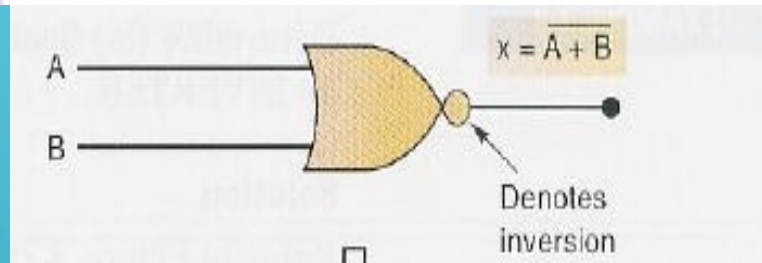




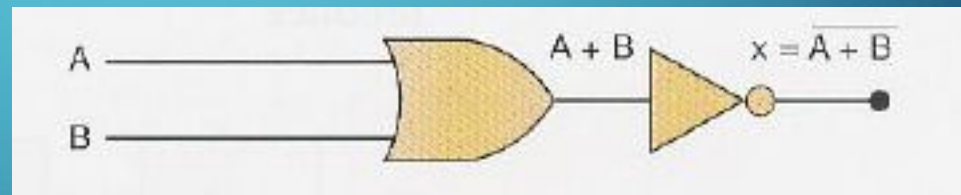
# NOR GATE

The NOR gate, like the NAND gate, is a useful logic element because it can also be used as a universal gate; that is, NOR gates can be used in combination to perform the AND, OR, and inverter operations.

*Symbol :*



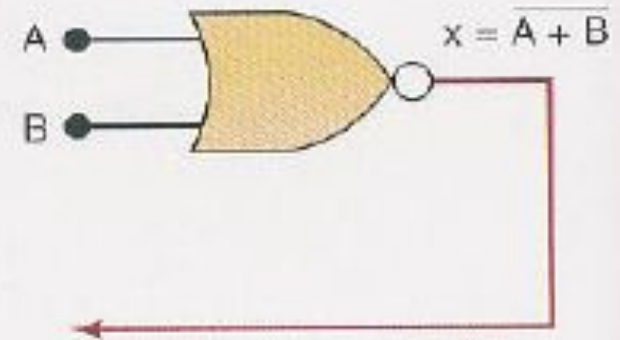
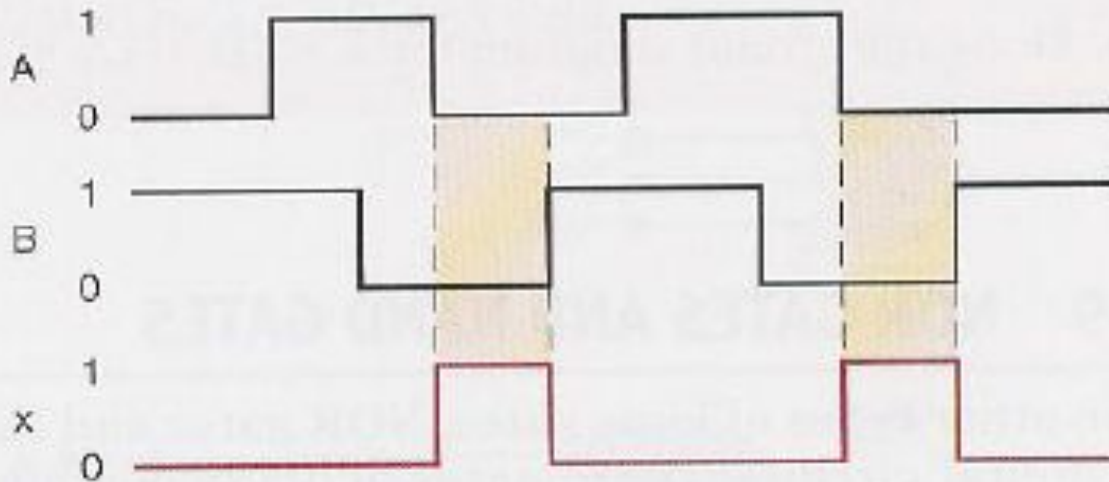
*Equivalent Circuit :*



*Truth Table :*

		OR		NOR	
A	B	$A + B$		$\overline{A + B}$	
0	0	0		1	
0	1	1		0	
1	0	1		0	
1	1	1		0	

# WAVE FORM AT THE OUT PUT OF NOR GATE



## Operation of NOR Gate

**For a 2-input NOR gate, output X is LOW if either input A or input B is HIGH, or if both A and B are HIGH; X is HIGH if both A and B are LOW.**

## EXAMPLE

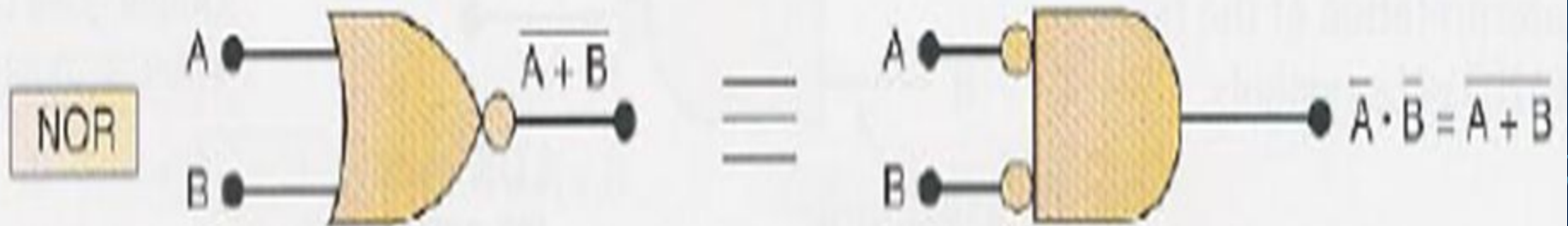
Show the output waveform for the 3-input NOR gate in Figure 3-35 with the proper time relation to the inputs.



The output X is LOW when any input is HIGH as shown by the output waveform X in the timing diagram.

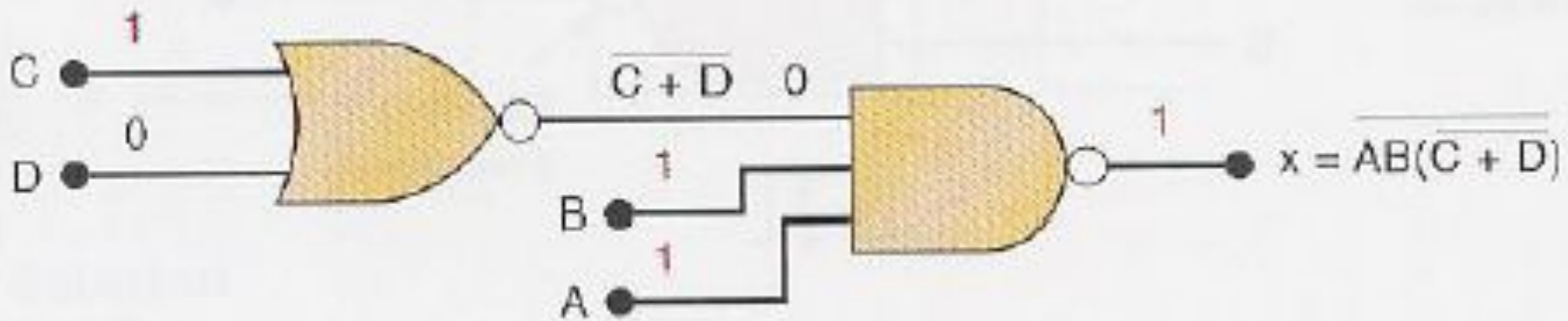
# NEGATIVE -AND EQUIVALENT OPERATION OF THE NOR GATE

**For a 2-input NOR gate performing a negative-AND operation, output  $X$  is HIGH if both inputs  $A$  and  $B$  are LOW.**



# EXAMPLE

Implement the logic circuit that has the expression  $x = \overline{AB \cdot (\overline{C + D})}$  using only NOR and NAND gates.

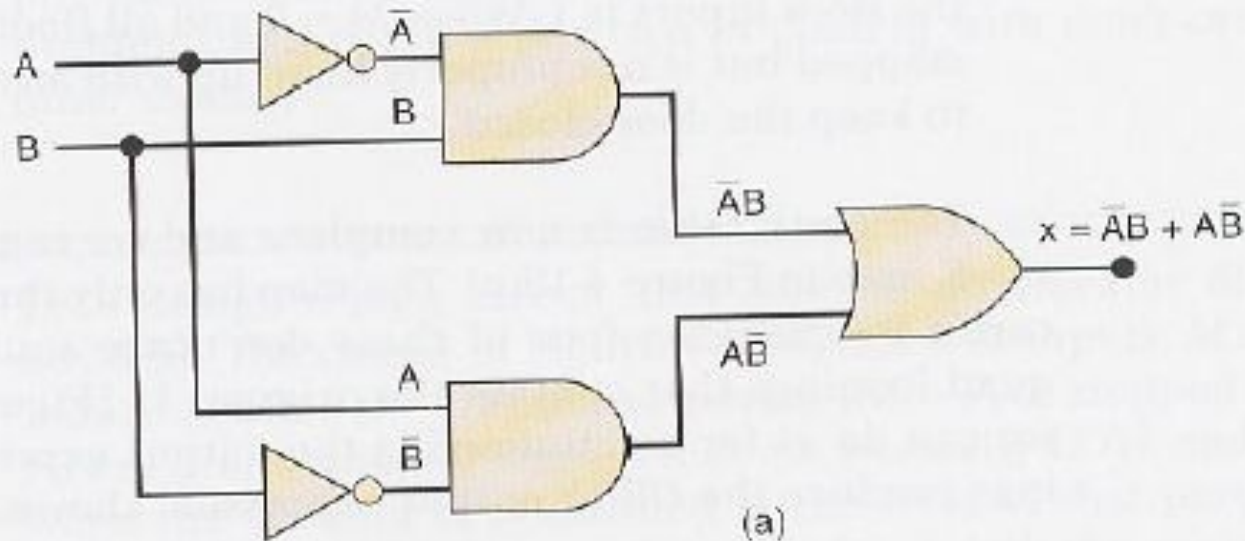




# EXCLUSIVE - OR

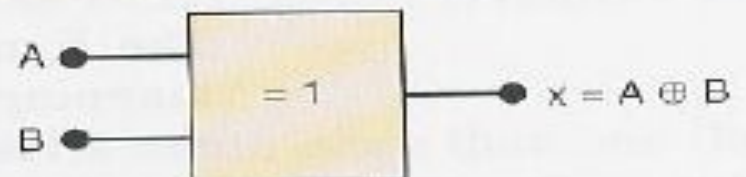
This circuit produces a **HIGH** output whenever the two inputs are at opposite levels.

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



A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

XOR gate symbols



# EXCLUSIVE - OR

An XOR gate has only *two* inputs; there are no three-input or four-input XOR gates. The two inputs are combined so that  $x = \overline{A}B + A\overline{B}$ . A shorthand way that is sometimes used to indicate the XOR output expression is

$$x = A \oplus B$$

where the symbol  $\oplus$  represents the XOR gate operation.

The characteristics of an XOR gate are summarized as follows:

1. It has only two inputs and its output is

$$x = \overline{A}B + A\overline{B} = A \oplus B$$

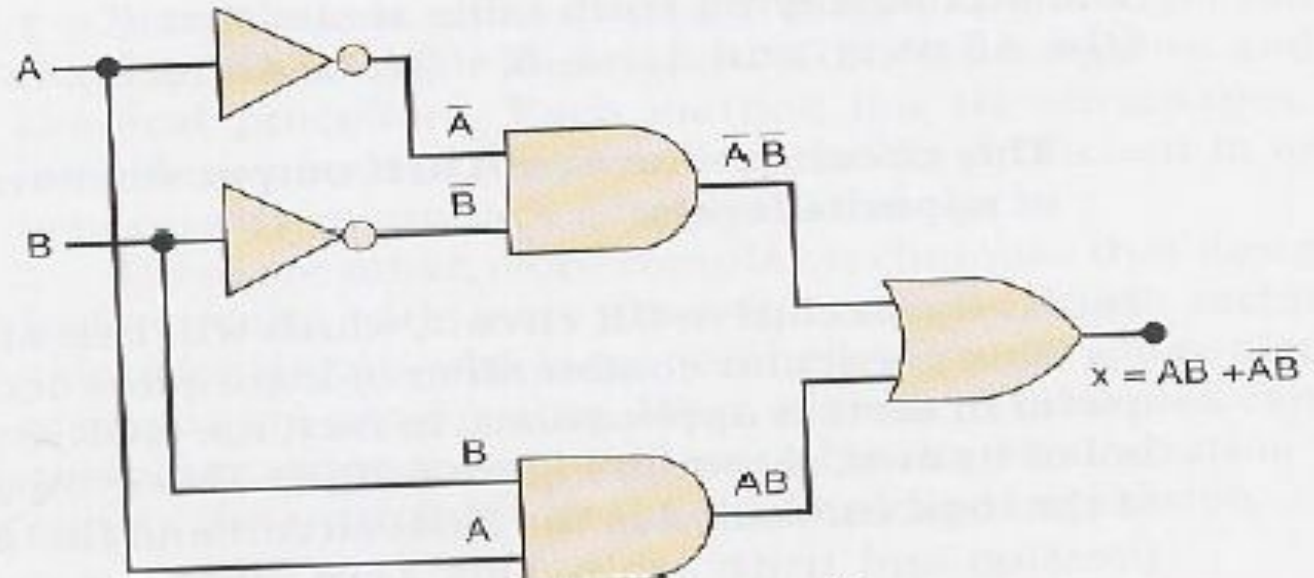
2. Its output is *HIGH* only when the two inputs are at *different* levels.

# Exclusive - NOR

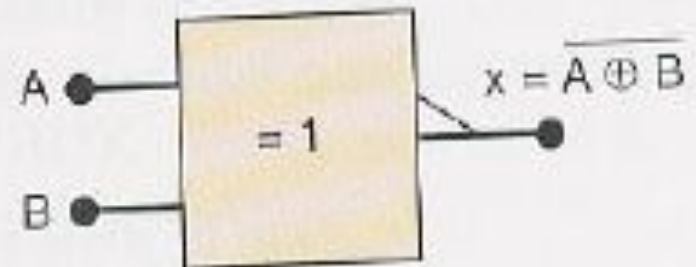
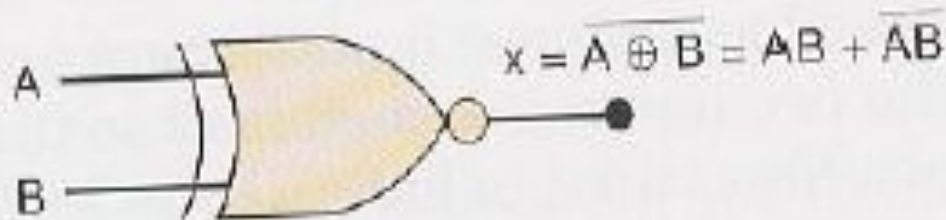
The **XNOR** produces a **HIGH** output whenever the two inputs are at the same level.

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

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1



XNOR gate symbols



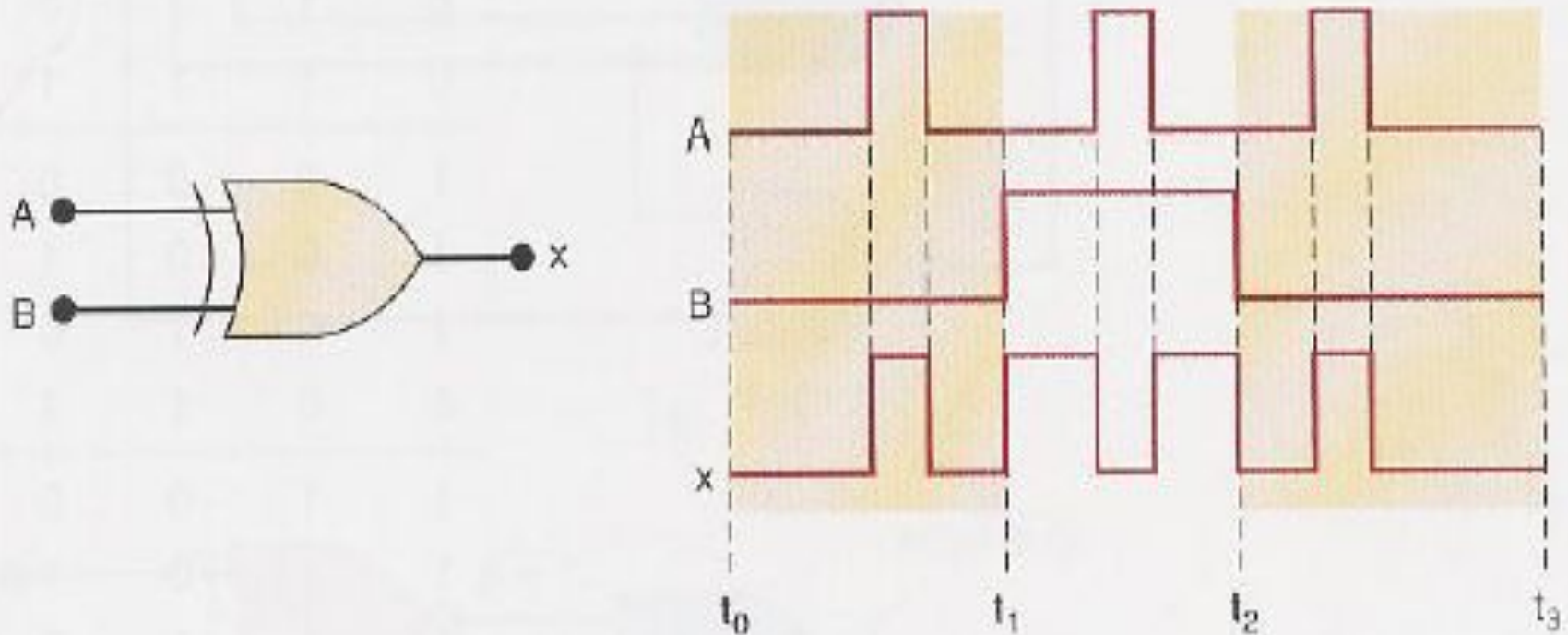
# Exclusive - OR

1. It has only two inputs and its output is

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

2. Its output is HIGH only when the two inputs are at the *same* level.

Determine the output waveform for the input waveforms given in Figure

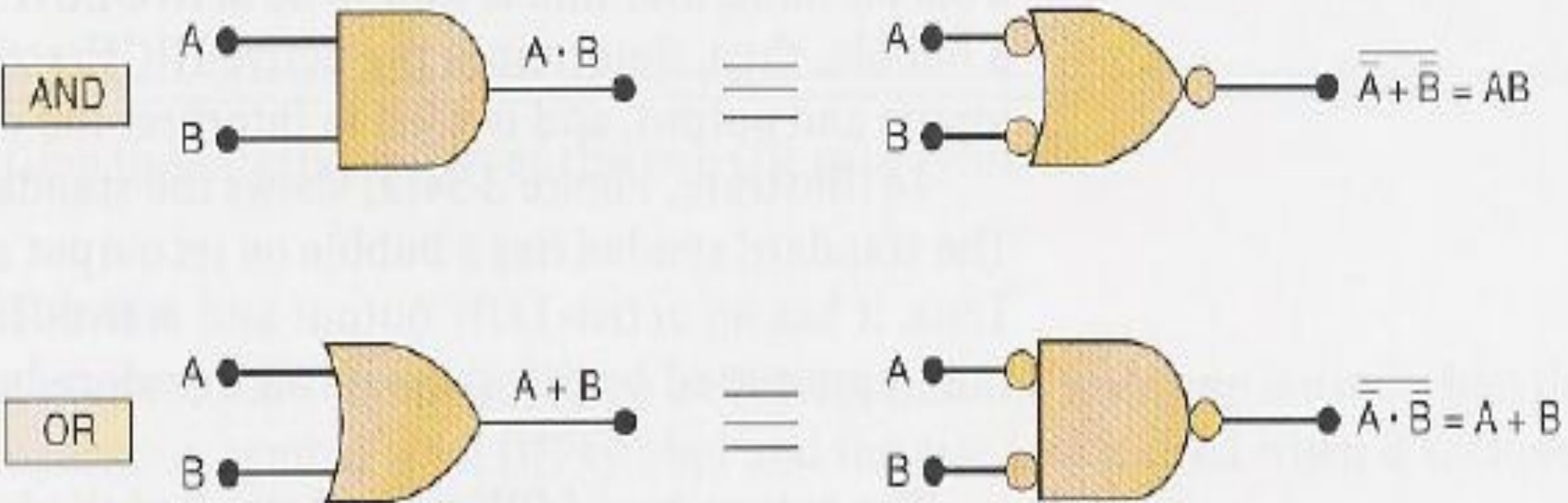




# ALTERNATE LOGIC GATE REPRESENTATIONS

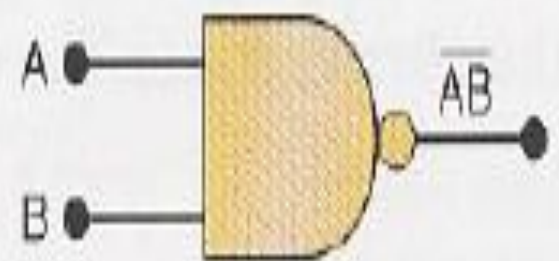
*The alternate symbol for each gate is obtained from the standard*

1. Invert each input and output of the standard symbol. This is done by adding bubbles (small circles) on input and output lines that do not have bubbles and by removing bubbles that are already there.
2. Change the operation symbol from AND to OR, or from OR to AND. (In the special case of the INVERTER, the operation symbol is not changed.)

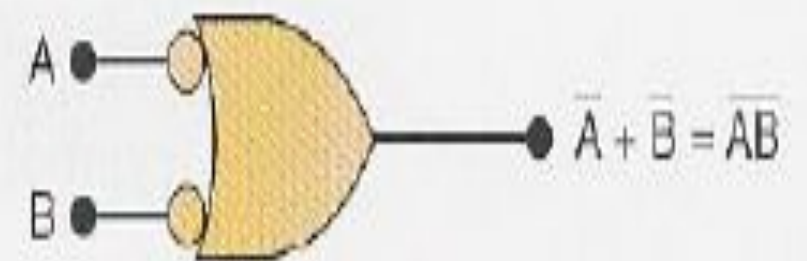




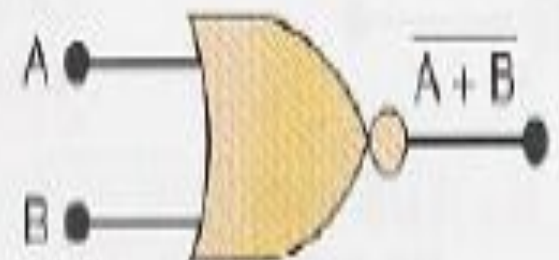
NAND



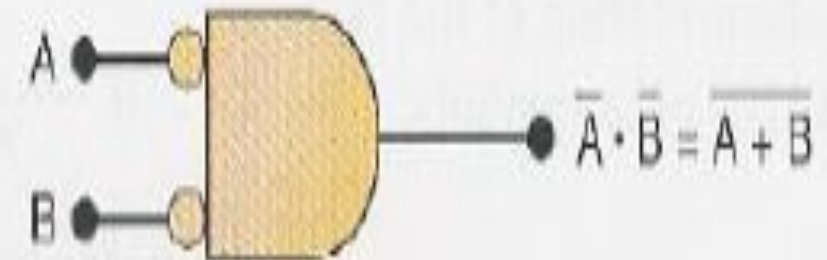
$\equiv$



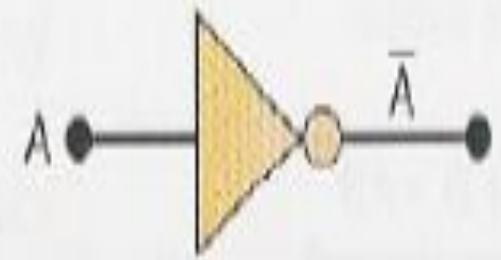
NOR



$\equiv$



INV



$\equiv$

