Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Lecture Three

Logic gates

The **logic gate** is the basic building block in digital systems. Logic gates operate with binary numbers. Gates are therefore referred to as binary logic gates. All voltages used with logic gates will be either HIGH or LOW. In this lecture, a HIGH voltage will mean a binary 1, and a LOW voltage will mean a binary 0. Remember that logic gates are electronic circuits.

All digital systems are constructed by using only three basic logic gates.

These basic gates are:

1- **AND** gate,
2- **OR** gate,
3- **NOT** gate.

1- THE AND GATE

The AND gate is called the "all or nothing" gate. The schematic in Fig. l shows the idea of the AND gate. The lamp (Y) will light only when both input switches (A and B) are closed.

Input Switches		Output Light
B	A	Y
open	open	no
open	close	no
close	open	no
close	close	yes

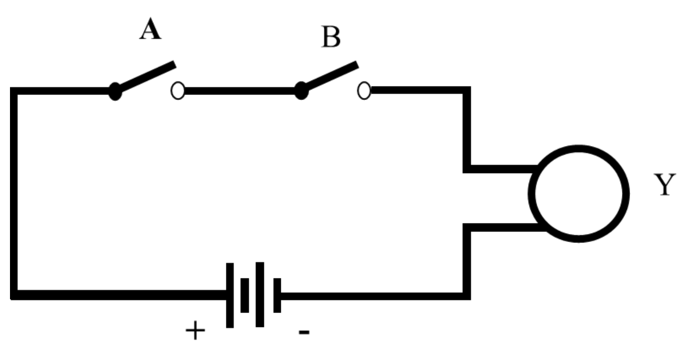


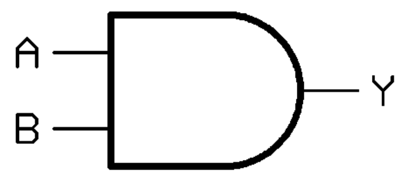
Figure 1.

The standard logic symbol for the AND gate is drawn in Fig. 2. This symbol shows the inputs as *A* and *B*. The output is shown as *Y*. This is the symbol for a 2-input **AND** gate. The truth table for the 2-input AND gate is shown in Fig. 2. Note that only when both input *A* and input *B* are 1 will the output be 1.

A truth table defines the logical outputs (0 or 1) of a logic gate for all possible combinations of logical inputs

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

(a) truth table



(b) AND-gate symbol

Figure 2

Boolean algebra is a form of symbolic logic that shows how logic gates operate. A Boolean expression is a “shorthand” method of showing what is happening in a logic circuit.

$$A \cdot B = Y$$

The Boolean expression is read as *A* AND (**.** means AND) *B* equals the output Y. The dot (**.**) means the logic function AND in Boolean algebra, not multiply as in regular algebra, Sometimes the dot (**.**) is left out of the Boolean expression. The Boolean expression for the 2-input AND gate is then:

$$A\,B = Y$$

The Boolean expression reads *A* AND *B* equals the output *Y*.

Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Example:

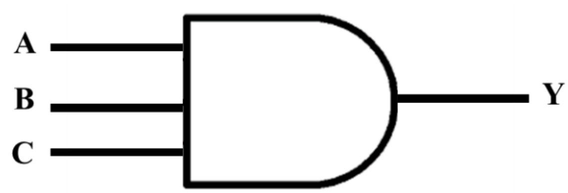
1- **Write the Boolean expression for a 3-input AND gate**

Solution:

$A \,.\, B \,.\, C = Y$ or $A\,B\,C = Y$

2- **Draw the logic symbol for a 3-input AND gate.**

Solution:



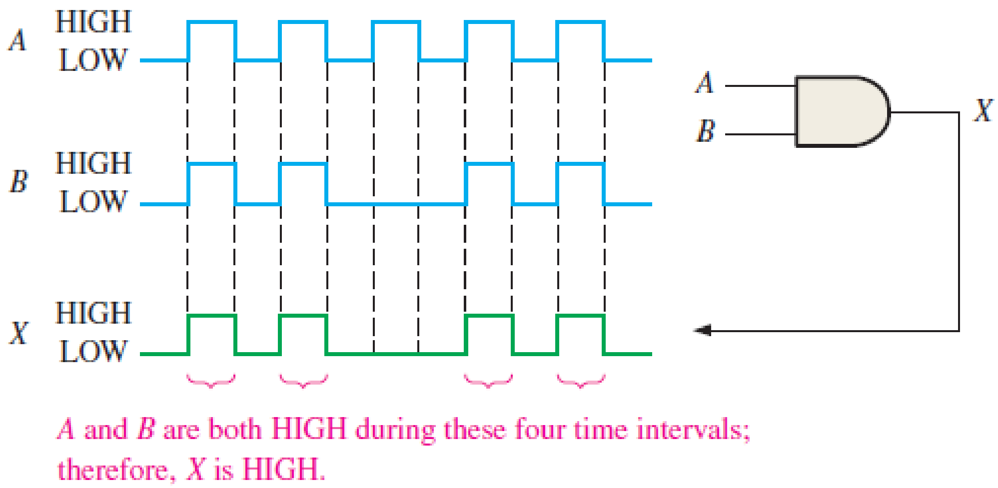
3- **Draw a truth table for a 3-input AND gate.**

Solution**:**

Input			***Output***
C	B	A	y
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

Example 2:

If the two waveforms A and B are applied to the AND gate as shown in the figure below, what is the resulting output waveform?

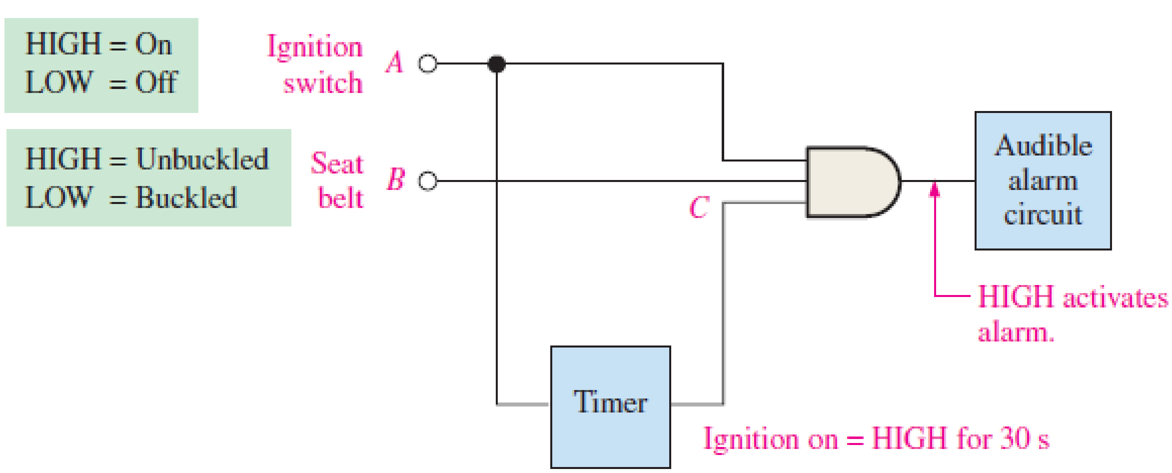


Solution

The output waveform *X* is **HIGH** only when both ***A*** and ***B*** waveforms are **HIGH**

Applications

A Seat Belt Alarm System



2- THE OR GATE

The OR gate is called the "any or all" gate. The schematic in Fig. ***3*** shows the idea of the OR gate. The lamp ***(Y)*** will glow when either switch ***A or*** switch ***B*** is closed. The lamp will also glow when both switches ***A*** and ***B*** are closed. The lamp *(Y)* will not glow when both switches ***(A*** and ***B)*** are open.

Input Switches		Output Light
B	A	Y
open	open	no
open	close	yes
close	open	yes
close	close	yes

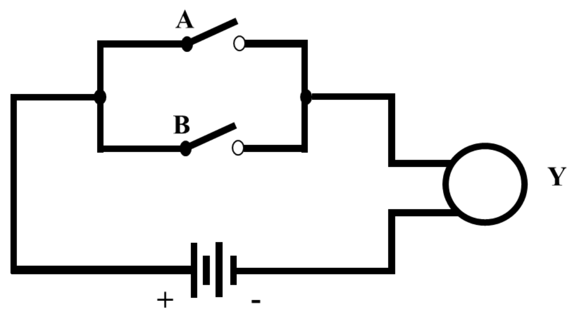
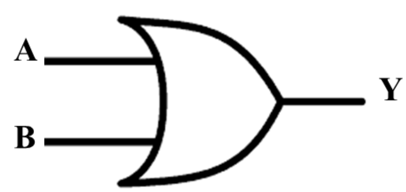


Figure 3

The standard logic symbol for an **OR** gate is drawn in Fig. 4. Note the different shape of the OR gate. The OR gate has two inputs labeled ***A*** and ***B***. The output is labeled ***Y***. The shorthand Boolean expression for this OR function is given as ***A + B = Y.*** Note that the plus (+) symbol means OR in Boolean algebra. The expression ***(A*** *+ B =* ***Y)*** is read as ***A*** OR (+ means OR) *B* equals output ***Y.***

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

(a) truth table



(b) OR-gate symbol

Figure 4

Example:

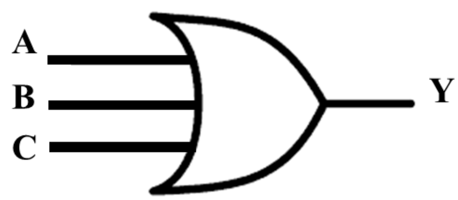
1- **Write the Boolean expression for a 3-input OR gate.**

Solution:

$$A + B + C = Y$$

2- **Draw the logic symbol for a 3-input OR gate.**

Solution



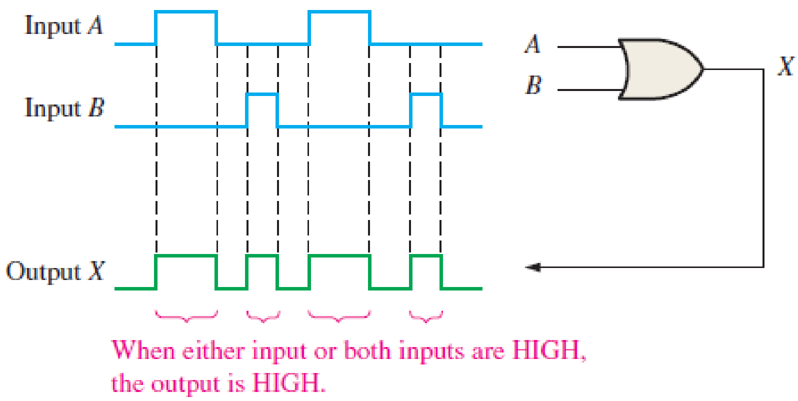
3- **Draw a truth table for a 3-input OR gate.**

Solution

Input			***Output***
C	B	A	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Example 2

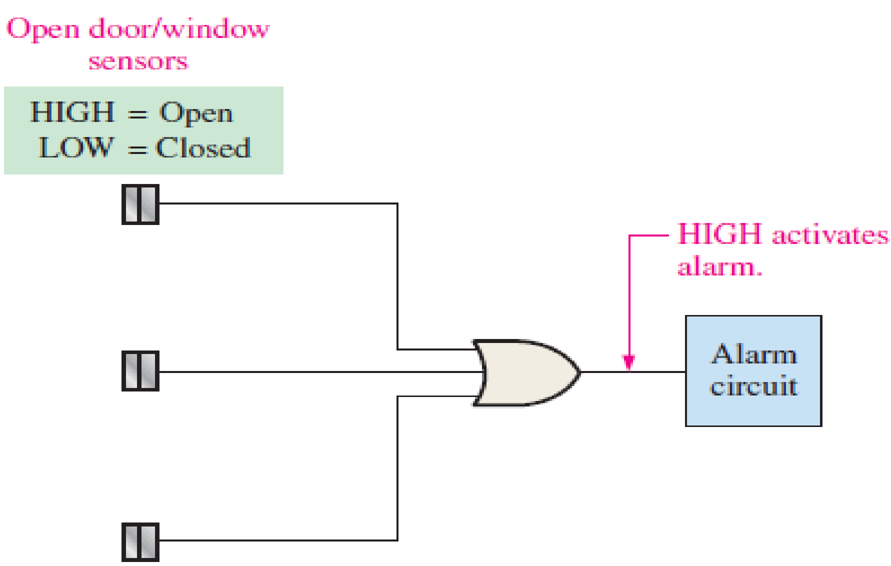
If the two waveforms A and B are applied to the OR gate as shown in the figure below, what is the resulting output waveform?



Solution

The output waveform *X* of a 2-input OR gate is HIGH when either or both input waveforms are HIGH.

Applications

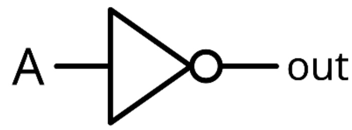


3- THE NOT GATE

A NOT gate is also called an inverter. The NOT gate has only one input and one output. Figure 5 illustrates the logic symbol for the inverter or NOT gate.

Input	Output
A	Y
0	1
1	0

(a) truth table



(b) NOT-gate symbol

Figure 4

The input is always changed to its opposite. **If** the input is 0, the NOT gate will give its ***complement,*** or opposite, which is 1. If the input to the NOT gate is a 1, the circuit will complement it to give a **0.** This inverting is also called ***complementing*** or ***negating.***

The Boolean expression for inverting is $A = \bar{A}$ reads as ***A*** equals the output ***not A.*** The bar over the ***A*** means to complement ***A.***

Universal Logic Gates

These gates can perform the functions of all other gates:

1- NAND Gate
2- NOR Gate

1- The NAND gate

The NAND gate is the complement of AND gate and the logic symbol of this gate is shown in fig. 5.

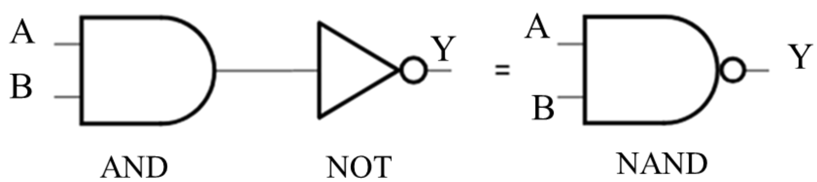


Figure 5

The Boolean expression for the entire circuit is $\overline{\boldsymbol{A.B}}$= Y. It is said that this is a not-AND or NAND circuit. The bubble is sometimes called an invert bubble. The invert bubble provides a simplified method of representing the NOT gate. The unique output from the NAND gate is a LOW when all inputs are HIGH.

The NAND gate is widely used in most digital systems.

Input		Output	
B	A	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

The AND- and NAND-gate truth tables

Examples:

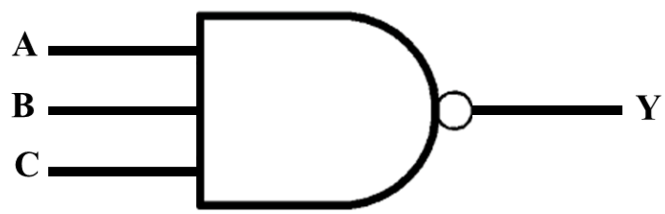
1- **Write the Boolean expression for a 3-input NAND gate.**

Solution:

$\overline{\boldsymbol{A}.\boldsymbol{B}.\boldsymbol{C}} = Y$ or $\overline{\boldsymbol{A}\,\boldsymbol{B}\,\boldsymbol{C}} = Y$

2- **Draw the logic symbol for a 3-input NAND gate**

Solution:



3- **Draw the truth table for a 3-input NAND gate.**

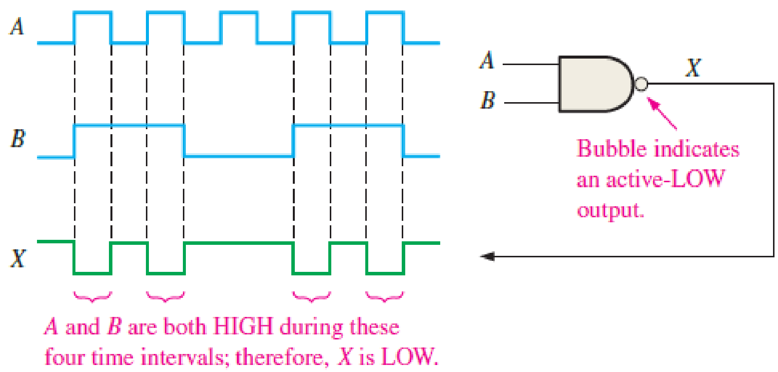
Solution:

Input			***Output***
C	B	A	y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Example

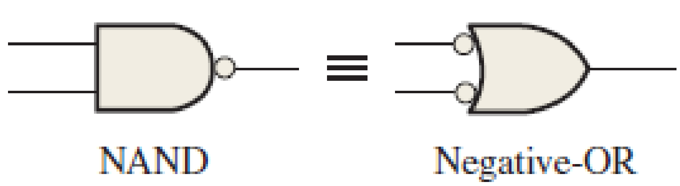
If the two waveforms A and B are applied to the NAND gate as shown in the figure below, what is the resulting output waveform?



Solution

The output waveform *X* is LOW only during the four-time interval when both input waveforms A and B are HIGH.

Negative-OR Equivalent Operation of a NAND Gate



Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

2- The NOR gate

The NOR gate represents the complement of the OR operation. An inverter has been connected to the output of an OR gate. Adding the overbar produces the Boolean expression $\overline{A+B} = Y$. The not-OR function can be drawn as a single logic symbol called a *NOR gate*. The standard symbol for the NOR gate is illustrated in Fig. 6. Note that a small invert bubble has been added to the OR symbol to form the NOR symbol.

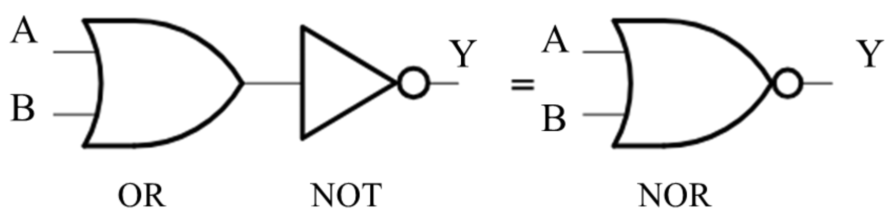


Fig. 6 The NOR gate

The truth table of the NOR gate is show below. Note that the output column of the NOR gate is the complement (has been inverted) of the shaded OR column.

Input		**Output**	
B	A	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

The ***unique output*** from the NOR gate is a **HIGH** when all inputs are LOW.

Examples:

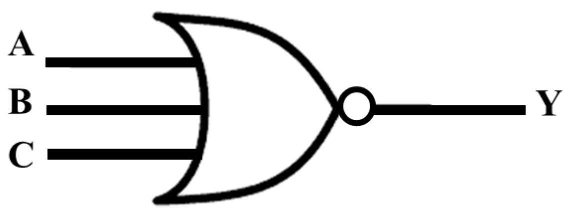
1- **Write the Boolean expression for a 3-input NOR gate.**

Solution:

$$\overline{\boldsymbol{A}+\boldsymbol{B}+\boldsymbol{C}}=Y$$

2- **Draw the logic symbol for a 3-input NOR gate**

Solution:



3- **What is the truth table for a 3-input NOR gate?**

Solution:

Input			***Output***
C	B	A	y
0	0	0	1
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	0

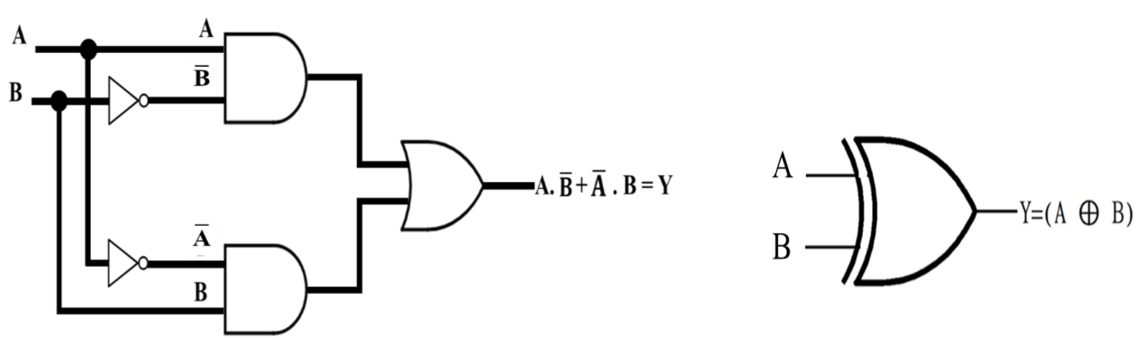
Department of Biomedical Engineering
Digital Electronics / Second stage
Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Derived Logic Gates

1- **The Exclusive-OR gate**
2- **The Exclusive-NOR gate**

1- The Exclusive-OR gate

The ***exclusive-OR gate*** is referred to as the "any but not all" gate. The exclusive-OR term is often shortened to read as ***XOR. A*** truth table for the **XOR** function is shown in Fig. **7.** Careful examination shows that this truth table is similar to the OR truth table except that, when both inputs are **1,** the **XOR** gate generates a 0. The **XOR** gate is enabled ***only when an odd number*** of *1s* ***appear at the inputs.***
Lines **1** and **4** of the truth table contain even numbers (0,2) of **1s,** and therefore the **XOR** gate is disabled and a 0 appears at the output. The XOR gate could be referred to as an odd-bits check circuit.
The Boolean expression of XOR is $\mathbf{A.\overline{B}+\overline{A}\,.\,B=A\oplus B}$



(a) Logic circuit that performs the XOR function

(b) Standard logic symbol for the XOR gate

Figure 7

Examples:

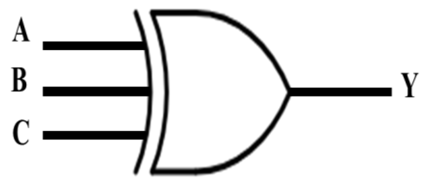
1- **Write the Boolean expression for a 3-input XOR gate.**

Solution:

$$A \oplus B \oplus C = Y$$

2- **Draw the logic symbol for a 3-input XOR gate.**

Solution:



3- **What is the truth table for a 3-input XOR gate? Remember that an odd number of 1s generates a 1 output.**

Solution:

Input			***Output***
C	B	A	y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

2- The Exclusive-NOR gate

X- NOR gate represent the complement of X-OR gate, logic symbol is shown in Fig. 8. The XOR gate produces the expression $\mathbf{A \oplus B}$. When this is inverted, it forms the Boolean expression for the XNOR gate, $\overline{\mathbf{A \oplus B}} = \mathbf{Y}$

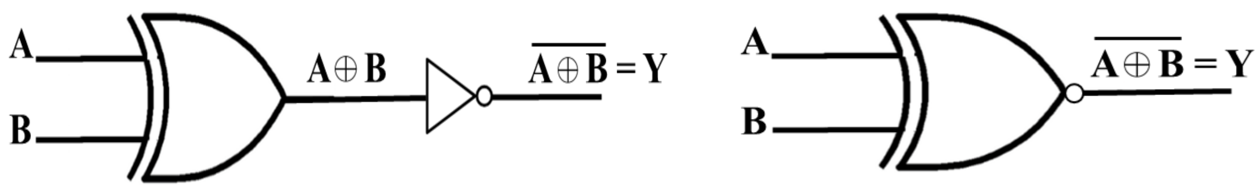


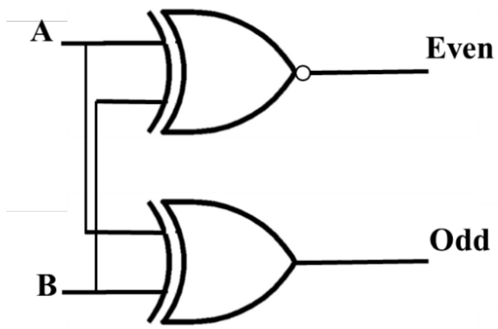
Figure 8

Note that all outputs of the XNOR gate are the complements of the XOR-gate outputs. While the XOR gate is an odd-number-of-1s detector, the XNOR gate detects *even numbers* of 1s. The XNOR gate will produce a 1 output when an *even number* of 1*s* appear at the inputs.

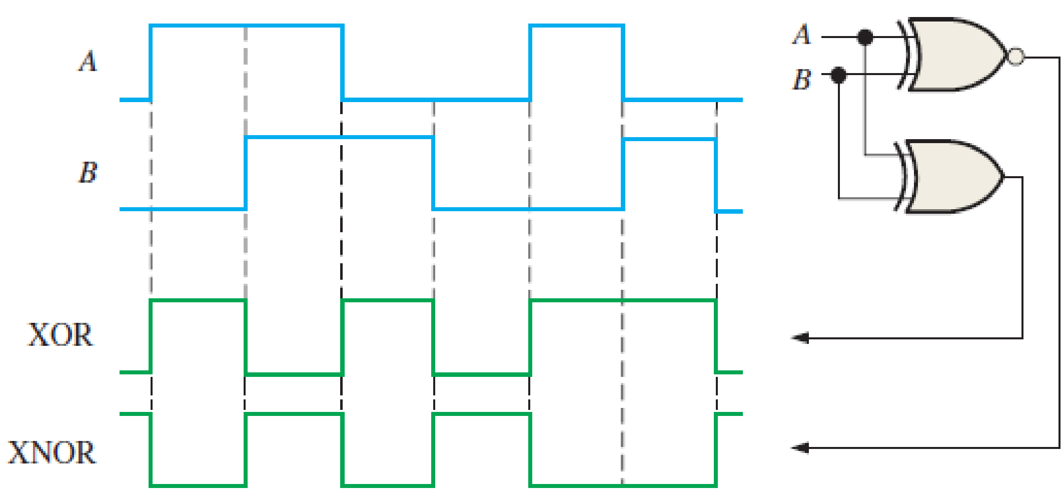
Input		**Output**	
B	A	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

The XOR and XNOR gate truth tables

Example: **For two input bits draw a logic circuit that finds the odd and even parity cheek for them.**



Input		Output	
B	A	Odd	Even
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1



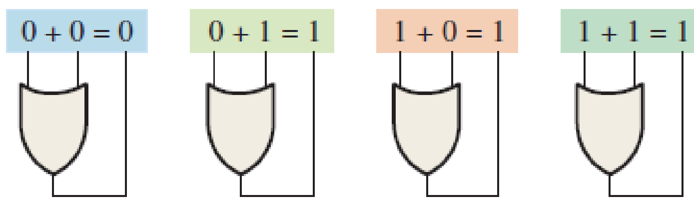
Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Boolean Algebra and Logic Simplification

Boolean algebra is the mathematics of digital logic.

Boolean Addition

Boolean addition is equivalent to the OR operation. The basic rules are illustrated with their relation to the OR gate in the figure below.



In Boolean algebra, a **sum term** is a sum of literals. In logic circuits, a sum term is produced by an OR operation. Some examples of sum terms are:

$$A + B, A + \overline{B}, A + B + \overline{C}, \text{ and } \overline{A} + B + C + \overline{D}.$$

Example:

Determine the values of A, B, C, and D that make the sum term $A + \overline{B} + C + \overline{D}$ equal to 0.

Solution

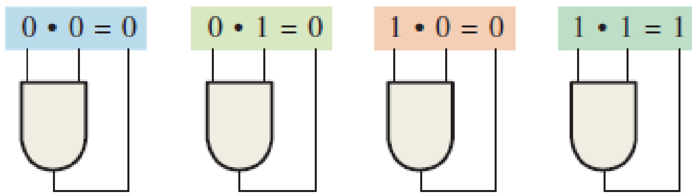
For the sum term to be 0, each of the literals in the term must be 0. Therefore, $A = \mathbf{0}$, $B = \mathbf{1}$ so that $\overline{B} = 0$, $C = \mathbf{0}$, and $D = \mathbf{1}$ so that $\overline{D} = 0$.

$$A + \overline{B} + C + \overline{D} = 0 + \overline{1} + 0 + \overline{1} = 0 + 0 + 0 + 0 = 0$$

Boolean Multiplication

Boolean multiplication is equivalent to the AND operation.

The basic rules are illustrated with their relation to the AND gate in the figure below.



Example:

Determine the values of A, B, C, and D that make the product term $A\overline{B}C\overline{D}$ equal to 1.

Solution

For the product term to be 1, each of the literals in the term must be 1. Therefore, $A = \mathbf{1}$, $B = \mathbf{0}$ so that $\overline{B} = 1$, $C = \mathbf{1}$, and $D = \mathbf{0}$ so that $\overline{D} = 1$.

$$A\overline{B}C\overline{D} = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$$

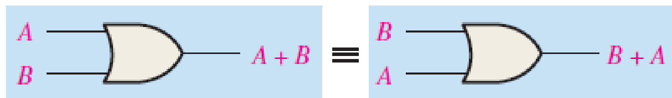
Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Laws and Rules of Boolean Algebra

1- Commutative Laws

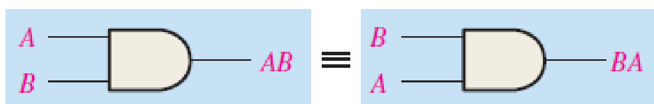
The *commutative law of addition* for two variables is written as

$$A + B = B + A$$



The *commutative law of multiplication* for two variables is

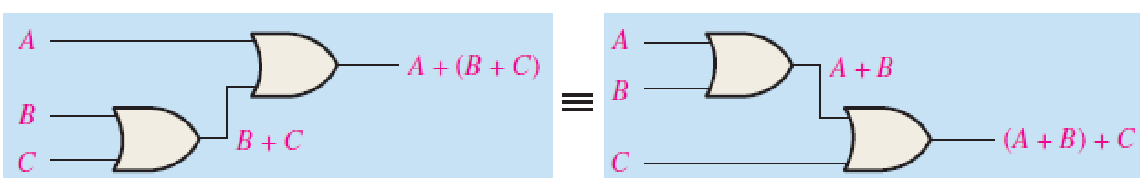
$$AB = BA$$



2- Associative Laws

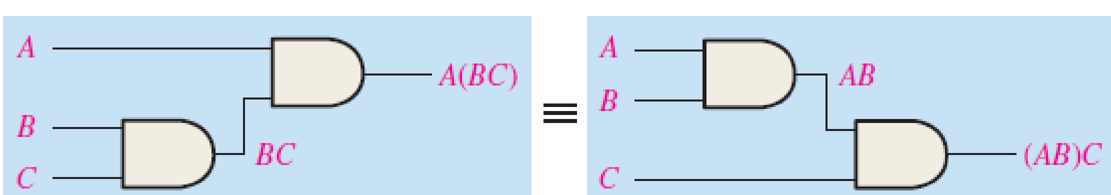
The *associative law of addition* is written as follows for three variables

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



The *associative law of multiplication* is written as follows for three variables:

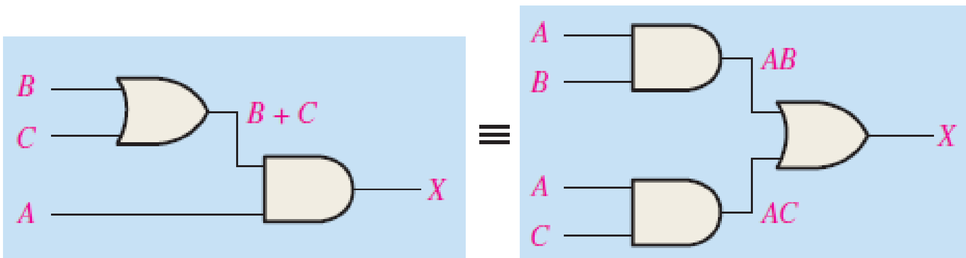
$$A(BC) = (AB)C$$



3- Distributive Law

The distributive law is written for three variables as follows:

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



$X = A(B + C)$ $\qquad$ $X = AB + AC$

Rules of Boolean Algebra

TABLE 4–1

Basic rules of Boolean algebra.

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	9. $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	12. $(A + B)(A + C) = A + BC$

A, *B*, or *C* can represent a single variable or a combination of variables.

***Rule 1*:** A + 0 = A

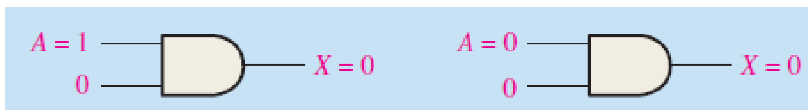
$X = A + 0 = A$

Lecturer: Dr. Hussam Jawad Kadhim AL_Janabi
Email: hussam.jawad@uomus.edu.iq

Rule 2: A + 1 = 1

$X = A + 1 = 1$

Rule 3: A . 0 = 0

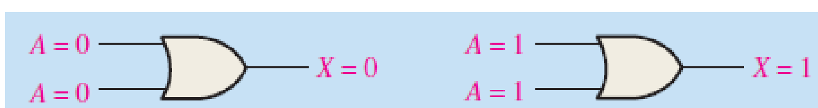


$X = A \cdot 0 = 0$

Rule 4: A . 1 = A

$X = A \cdot 1 = A$

Rule 5: A + A = A

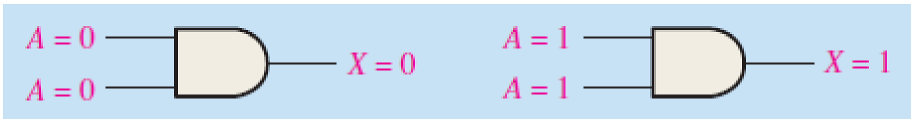


$X = A + A = A$

Rule 6: $A + \bar{A} = 1$

$X = A + \bar{A} = 1$

Rule 7: A . A = A

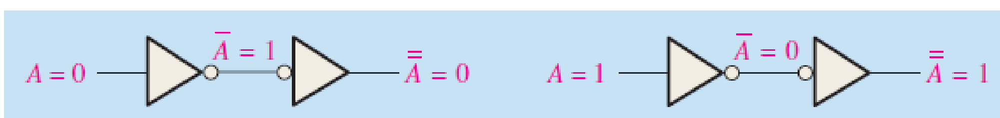


$X = A \cdot A = A$

Rule 8: A . Ā = 0

$X = A \cdot \overline{A} = 0$

Rule 9: $\overline{\overline{A}} = A$



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

Rule 10: A + AB = A

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

Rule 11: $A + \bar{A}B = A + B$

$$
\begin{aligned}
A + \overline{A}B &= (A + AB) + \overline{A}B && \text{Rule 10: } A = A + AB \\
&= (AA + AB) + \overline{A}B && \text{Rule 7: } A = AA \\
&= AA + AB + A\overline{A} + \overline{A}B && \text{Rule 8: adding } A\overline{A} = 0 \\
&= (A + \overline{A})(A + B) && \text{Factoring} \\
&= 1 \cdot (A + B) && \text{Rule 6: } A + \overline{A} = 1 \\
&= A + B && \text{Rule 4: drop the 1}
\end{aligned}
$$

Rule 12: $(A + B)(A + C) = A + BC$

$$
\begin{aligned}
(A + B)(A + C) &= AA + AC + AB + BC && \text{Distributive law} \\
&= A + AC + AB + BC && \text{Rule 7: } AA = A \\
&= A(1 + C) + AB + BC && \text{Factoring (distributive law)} \\
&= A \cdot 1 + AB + BC && \text{Rule 2: } 1 + C = 1 \\
&= A(1 + B) + BC && \text{Factoring (distributive law)} \\
&= A \cdot 1 + BC && \text{Rule 2: } 1 + B = 1 \\
&= A + BC && \text{Rule 4: } A \cdot 1 = A
\end{aligned}
$$

Logic Simplification Using Boolean Algebra

Example 1

Using Boolean algebra techniques, simplify this expression:

AB + A(B + C) + B(B + C)

Solution

The following is not necessarily the only approach.

Step 1: Apply the distributive law to the second and third terms in the expression, as follows:

$$AB + AB + AC + BB + BC$$

Step 2: Apply rule 7 ($BB = B$) to the fourth term.

$$AB + AB + AC + B + BC$$

Step 3: Apply rule 5 ($AB + AB = AB$) to the first two terms.

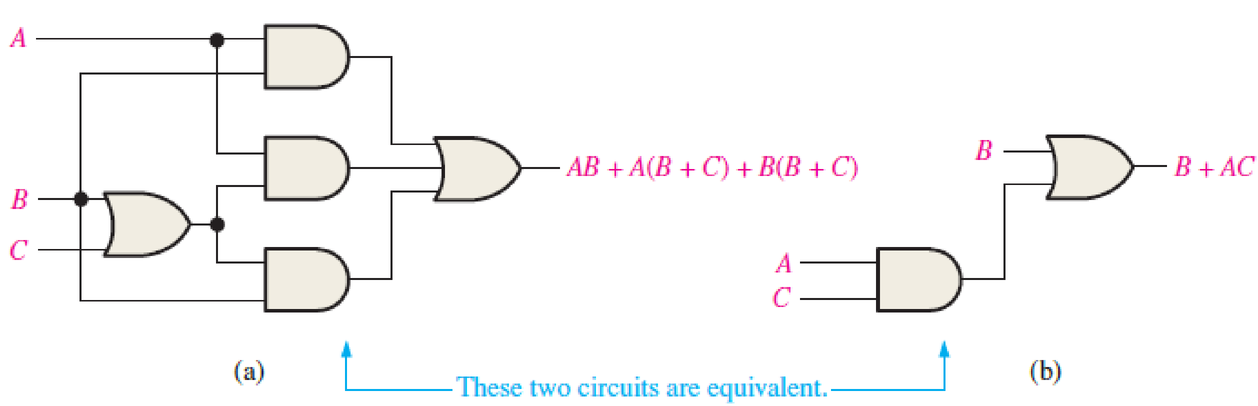
$$AB + AC + B + BC$$

Step 4: Apply rule 10 ($B + BC = B$) to the last two terms.

$$AB + AC + B$$

Step 5: Apply rule 10 ($AB + B = B$) to the first and third terms.

$$B + AC$$



Example 2

Simplify the following Boolean expression:

$$[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$$

Solution

Step 1: Apply the distributive law to the terms within the brackets.

$$(A\overline{B}C + A\overline{B}BD + \overline{A}\,\overline{B})C$$

Step 2: Apply rule 8 ($\overline{B}B = 0$) to the second term within the parentheses.

$$(A\overline{B}C + A \cdot 0 \cdot D + \overline{A}\,\overline{B})C$$

Step 3: Apply rule 3 ($A \cdot 0 \cdot D = 0$) to the second term within the parentheses.

$$(A\overline{B}C + 0 + \overline{A}\,\overline{B})C$$

Step 4: Apply rule 1 (drop the 0) within the parentheses.

$$(A\overline{B}C + \overline{A}\,\overline{B})C$$

Step 5: Apply the distributive law.

$$A\overline{B}CC + \overline{A}\,\overline{B}C$$

Step 6: Apply rule 7 ($CC = C$) to the first term.

$$A\overline{B}C + \overline{A}\,\overline{B}C$$

Step 7: Factor out $\overline{B}C$.

$$\overline{B}C(A + \overline{A})$$

Step 8: Apply rule 6 ($A + \overline{A} = 1$).

$$\overline{B}C \cdot 1$$

Step 9: Apply rule 4 (drop the 1).

$$\overline{B}C$$

H.W

Simplify the following Boolean expression:

$$\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}C + ABC$$