اسم المادة : تقنيات رقمية
اسم التدريسي :م.م علياء محمد جواد
المرحلة : الثانية
السنة الدراسية :2023_2024
عنوان المحاظرة:Logic Gates

1

Logic Gates

- The inverter (NOT circuit) performs the operation called *inversion* or *complementation.*
- The inverter changes one logic level to the opposite level. In terms of bits, it changes a 1 to a 0 and a 0 to a 1.
- Standard logic symbols for the **inverter** are shown below:

(a) Distinctive shape symbols with negation indicators

(b) Rectangular outline symbols with polarity indicators

- The "bubble" indicates negation (**inversion** or *complementation*) when it appears on the input or output of any logic element.

2

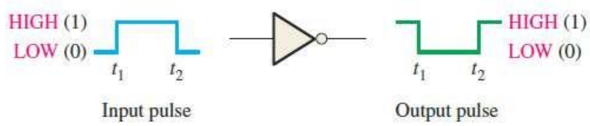
Logic Gates
The inverter (NOT circuit)

- Inverter Truth Table

Inverter truth table.

Input	Output
LOW (0)	HIGH (1)
HIGH (1)	LOW (0)

- A truth table shows the output for each possible input in terms of levels and corresponding bits.
- Inverter Operation

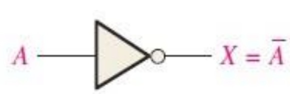


3

Logic Gates
The inverter (NOT circuit)

- Logic Expression for an Inverter
- In **Boolean algebra**, which is the mathematics of logic circuits, a variable is generally designated by one or two letters although there can be more.
- The **complement** of a variable is designated by a bar over the letter.
- The operation of an inverter (NOT circuit) can be expressed as follows: If the input variable is called *A* and the output variable is called *X*, then

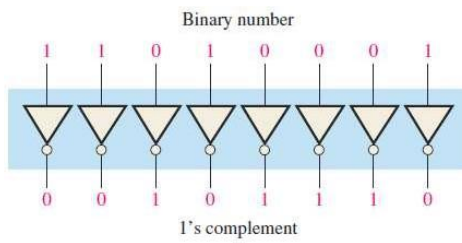
$$X = \overline{A}$$



4

The inverter (NOT circuit)
An Application

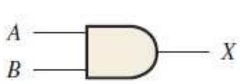
- The figure below shows a circuit for producing the 1's complement of an 8-bit binary number.



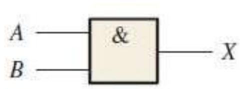
- The bits of the binary number are applied to the inverter inputs and the 1's complement of the number appears on the outputs.

The AND Gate

- The term *gate* is used to describe a circuit that performs a basic logic operation.
- The AND gate is one of the basic gates that can be combined to form any logic function.
- An AND gate can have two or more inputs and performs what is known as logical multiplication.



(a) Distinctive shape



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

Truth table for a 2-input AND gate.

Inputs		Output
A	*B*	*X*
0	0	0
0	1	0
1	0	0
1	1	1

1 = HIGH, 0 = LOW

Logic Gates

- 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.

Example: Develop the truth table for a 3-input AND gate

Solution:

Inputs			Output
A	*B*	*C*	*X*
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

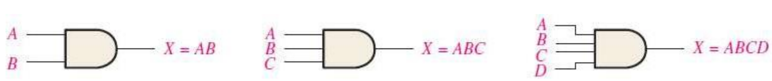
7

AND Gate

- Logic Expressions for an AND Gate
- The logical AND function of two variables is represented mathematically either by placing a dot between the two variables, as $A \cdot B$, or by simply writing the adjacent letters without the dot, as AB.
- **Boolean multiplication** follows the same basic rules governing binary multiplication, and are as follows:

$$0 \cdot 0 = 0$$
$$0 \cdot 1 = 0$$
$$1 \cdot 0 = 0$$
$$1 \cdot 1 = 1$$

Boolean multiplication is the same as the AND function.



Boolean expressions for AND gates with two, three, and four inputs.

8

The OR Gate

- An OR gate performs what is known as logical addition.
- An **OR gate** can have two or more inputs and one output.

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

- OR Gate Truth Table

Truth table for a 2-input OR gate.

Inputs		Output
A	*B*	*X*
0	0	0
0	1	1
1	0	1
1	1	1

9

OR Gate

- Logic Expressions for an OR Gate
- The logical OR function of two variables is represented mathematically by a + between the two variables, e.g., $X = A + B$. The plus sign is read as “OR.”
- The basic rules for **Boolean addition** are as follows:

$$0 + 0 = 0$$
$$0 + 1 = 1$$
$$1 + 0 = 1$$
$$1 + 1 = 1$$

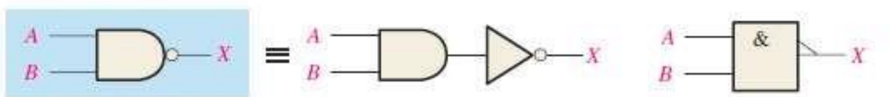
- **Boolean addition is the same as the OR function.**

10

NAND Gate

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

- The NAND gate is the same as the AND gate except the output is
- inverted.



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

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

NAND Gate

- A **NAND gate** produces a LOW output only when all the inputs are HIGH. When any of the inputs is LOW, the output will be HIGH.
- The table below is for the specific case of a 2-input NAND gate

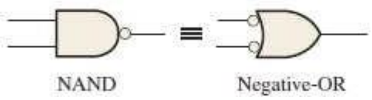
Truth table for a 2-input NAND gate.

Inputs		Output
A	*B*	*X*
0	0	1
0	1	1
1	0	1
1	1	0

1 = HIGH, 0 = LOW.

NAND Gate

- **Negative-OR Equivalent Operation of a NAND Gate**
- For a 2-input NAND gate performing a negative-OR operation, output *X* is HIGH when either input *A* or input *B* is LOW, or when both *A* and *B* are LOW.



A	B	$\overline{AB} = X$
0	0	$\overline{0 \cdot 0} = \overline{0} = 1$
0	1	$\overline{0 \cdot 1} = \overline{0} = 1$
1	0	$\overline{1 \cdot 0} = \overline{0} = 1$
1	1	$\overline{1 \cdot 1} = \overline{1} = 0$

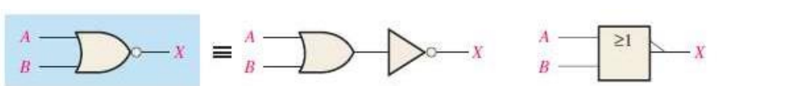
- Logic Expressions for a NAND Gate
- The Boolean expression for the output of a 2-input NAND gate is

$X = \overline{AB}$ *where* a bar over a variable or variables indicates an inversion.

13

The NOR Gate

- The NOR gate, like the NAND gate, is a useful logic element because it can also be used as a universal gate.
- The NOR is the same as the OR except the output is inverted.



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

(b) Rectangular outline, 2-input NOR gate with polarity indicator

- Operation of a NOR Gate

Truth table for a 2-input NOR gate.

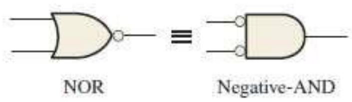
Inputs		Output
A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

1 = HIGH, 0 = LOW.

14

The NOR Gate

- **Negative-AND Equivalent Operation of the NOR Gate**
- For a 2-input NOR gate performing a negative-AND operation, output *X* is HIGH only when both inputs *A* and *B* are LOW.



- **Logic Expressions for a NOR Gate**

The Boolean expression for the output of a 2-input NOR gate can be written as

$X = \overline{A + B}$

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

The Exclusive-OR Gate

- The output of an exclusive-OR gate is HIGH *only* when the two inputs are at opposite logic levels.

Truth table for an exclusive-OR gate.

Inputs		Output
A	B	X
0	0	0
0	1	1
1	0	1
1	1	0

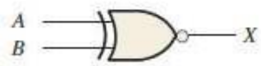
(a) Distinctive shape (b) Rectangular outline

The Exclusive-NOR Gate

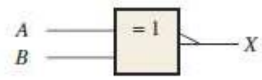
- For an exclusive-NOR gate, output *X* is LOW when input *A* is LOW and input *B* is HIGH, or when *A* is HIGH and *B* is LOW; *X* is HIGH when *A* and *B* are both HIGH or both LOW.

Truth table for an exclusive-NOR gate.

Inputs		Output
A	*B*	*X*
0	0	1
0	1	0
1	0	0
1	1	1



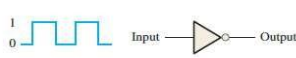
(a) Distinctive shape



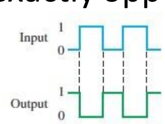
(b) Rectangular outline

Timing Diagrams

- a *timing diagram* is basically a graph that accurately displays the relationship of two or more waveforms with respect to each other on a time basis.
- Example:
- A waveform is applied to an inverter in the figure below.



- Determine the output waveform corresponding to the input and show the timing diagram.
- Solution:
- The output waveform is exactly opposite to the input (inverted).



Boolean Algebra and Logic Simplification

- Boolean algebra is the mathematics of digital logic.
- A **variable** is a symbol (usually an italic uppercase letter or word) used to represent an action, a condition, or data. In Boolean algebra, any single variable can have only a 1 or a 0 value.
- The **complement** is the inverse of a variable and it is indicated by a bar over the variable (overbar). For example, the complement of the variable *A* is *A*. If *A* = 1, then $\overline{A}$ = 0. If *A* = 0, then $\overline{A}$ = 1.
- The **complement** of the variable *A* is read as "not *A*" or "*A* bar." Sometimes a prime symbol rather than an overbar is used to denote the complement of a variable; for example, *A′* indicates a complement of *A*.
- A **literal** is a variable or the complement of a variable.

19

Boolean Addition

- The **Boolean addition** is equivalent to the OR operation.
- In Boolean algebra, a **sum term** is a sum of literals.
- Some examples of sum terms are *A* + *B*, *A* + $\overline{B}$, *A* + *B* + $\overline{C}$, and $\overline{A}$ + *B* + *C* + *D*.
- A **sum term** is equal to 1 when one or more of the literals in the term are 1. A sum term is equal to 0 only if each of the literals is 0.
- Example: Determine the values of *A*, *B*, *C*, and *D* that make the sum term *A* + *B* + *C* + *D* equal to 0.
- Solution:
- For the sum term to be 0, each of the literals in the term must be 0. Therefore, *A* = **0**, *B* = **1** so that $\overline{B}$ = 0, *C* = **0**, and *D* = **1** so that $\overline{D}$ = 0.
- *A* + $\overline{B}$ + *C* + $\overline{D}$ = 0 + 1 + 0 + 1 = 0 + 0 + 0 + 0 = 0

20

Boolean Multiplication

- **Boolean multiplication** is equivalent to the AND operation.
- In Boolean algebra, a **product term** is the product of literals.
- Some examples of product terms are $A\overline{B}$, AB, ABC, and $A\overline{B}C\overline{D}$.
- A product term is equal to 1 only if each of the literals in the term is 1. A product term is equal to 0 when one or more of the literals are 0.
- Example: Determine the values of A, B, C, and D that make the product term $ABCD$ equal to 1.
- Solution:
- For the product term to be 1, each of the literals in the term must be 1. Therefore, A = **1**, B = **0** so that $\overline{B}$ = 1, C = **1**, and D = **0** so that $\overline{D}$ = 1.
- $ABCD = 1 \cdot \overline{0} \cdot 1 \cdot \overline{0} = 1 \cdot 1 \cdot 1 \cdot 1 = 1$

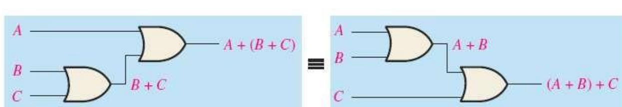
Laws and Rules of Boolean Algebra

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

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



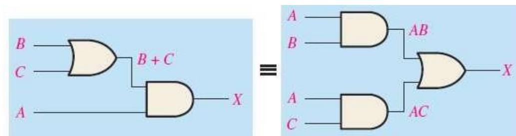
The *associative law of multiplication* is written as follows for three variables:

A(BC)=(AB)C

Laws and Rules of Boolean Algebra

- **Distributive Law**
- The distributive law is written for three variables as follows:

A(B+C)=AB+AC



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

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.

Rules of Boolean Algebra

- Rule 1: $A + 0 = A$. A variable ORed with 0 is always equal to the variable.
- Rule 2: $A + 1 = 1$. A variable ORed with 1 is always equal to 1.
- Rule 3: $A.0 = 0$. A variable ANDed with 0 is always equal to 0.
- Rule 4: $A.1 = A$. A variable ANDed with 1 is always equal to the variable.
- Rule 5: $A + A = A$. A variable ORed with itself is always equal to the variable.
- Rule 6: $A + \overline{A} = 1$. A variable ORed with its complement is always equal to 1.
- Rule 7: $A.A = A$. A variable ANDed with itself is always equal to the variable.
- Rule 8: $A.\overline{A} = 0$. A variable ANDed with its complement is always equal to 0.
- Rule 9: $\overline{\overline{A}} = A$. The double complement of a variable is always equal to the variable.

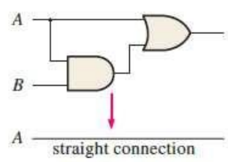
Rules of Boolean Algebra

- Rule 10: $A + AB = A$. This rule can be proved by applying the distributive law, rule 2, and rule 4 as follows:

$A + AB = A \cdot 1 + AB = A(1 + B)$	Factoring (distributive law)
$= A \cdot 1$	Rule 2: $(1 + B) = 1$
$= A$	Rule 4: $A \cdot 1 = A$

A	B	AB	$A + AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

equal (A and $A + AB$ columns)



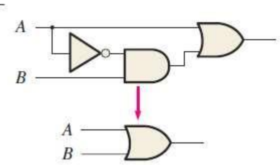
Rules of Boolean Algebra

- Rule 11: $A + \overline{A}B = A + B$. This rule can be proved as follows:

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

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

equal ($A + \overline{A}B$ and $A + B$ columns)



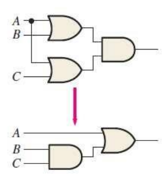
Rules of Boolean Algebra

- Rule 12: $(A + B)(A + C) = A + BC$. This rule can be proved as follows:

$(A + B)(A + C) = AA + AC + AB + BC$		Distributive law
	$= A + AC + AB + BC$	Rule 7: $AA = A$
	$= A(1 + C) + AB + BC$	Factoring (distributive law)
	$= A \cdot 1 + AB + BC$	Rule 2: $1 + C = 1$
	$= A(1 + B) + BC$	Factoring (distributive law)
	$= A \cdot 1 + BC$	Rule 2: $1 + B = 1$
	$= A + BC$	Rule 4: $A \cdot 1 = A$

A	B	C	A + B	A + C	(A + B)(A + C)	BC	A + BC
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

(A + B)(A + C) and A + BC: equal



27

DeMorgan's Theorems

- DeMorgan's first theorem is stated as follows:
- **The complement of a product of variables is equal to the sum of the complements of the variables.**

The formula for expressing this theorem for two variables is

$$\overline{XY} = \overline{X} + \overline{Y}$$

DeMorgan's second theorem is stated as follows:

- **The complement of a sum of variables is equal to the product of the complements of the variables.**

The formula for expressing this theorem for two variables is

$$\overline{X + Y} = \overline{X}\,\overline{Y}$$

28

Applying DeMorgan's Theorems

- The following procedure illustrates the application of DeMorgan's theorems and Boolean algebra to the specific expression

$$\overline{\overline{A + B\overline{C}} + D(\overline{E + \overline{F}})}$$

Step 1: Identify the terms to which you can apply DeMorgan's theorems, and think of each term as a single variable. Let $\overline{A + B\overline{C}} = X$ and $D(\overline{E + \overline{F}}) = Y$.

Step 2: Since $\overline{X + Y} = \overline{X}\,\overline{Y}$,

$$\overline{(\overline{A + B\overline{C}}) + (D(\overline{E + \overline{F}}))} = (\overline{\overline{A + B\overline{C}}})(\overline{D(\overline{E + \overline{F}})})$$

Step 3: Use rule 9 ($\overline{\overline{A}} = A$) to cancel the double bars over the left term (this is not part of DeMorgan's theorem).

$$(\overline{\overline{A + B\overline{C}}})(\overline{D(\overline{E + \overline{F}})}) = (A + B\overline{C})(\overline{D(\overline{E + \overline{F}})})$$

Step 4: Apply DeMorgan's theorem to the second term.

$$(A + B\overline{C})(\overline{D(\overline{E + \overline{F}})}) = (A + B\overline{C})(\overline{D} + (\overline{\overline{E + \overline{F}}}))$$

Step 5: Use rule 9 ($\overline{\overline{A}} = A$) to cancel the double bars over the $E + \overline{F}$ part of the term.

$$(A + B\overline{C})(\overline{D} + \overline{\overline{E + \overline{F}}}) = (A + B\overline{C})(\overline{D} + E + \overline{F})$$

29

Applying DeMorgan's Theorems

- Example: Apply DeMorgan's theorems to the following expression:

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

- Solution:

Let $A + B + C = X$ and $D = Y$. The expression $\overline{(A + B + C)D}$ is of the form $\overline{XY} = \overline{X} + \overline{Y}$ and can be rewritten as

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

Next, apply DeMorgan's theorem to the term $\overline{A + B + C}$.

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

30

Applying DeMorgan's Theorems

- Example: Apply DeMorgan's theorems to the following expression:

$\overline{ABC + DEF}$

Solution

Let $ABC = X$ and $DEF = Y$. The expression $\overline{ABC + DEF}$ is of the form $\overline{X + Y} = \overline{X}\,\overline{Y}$ and can be rewritten as

$$\overline{ABC + DEF} = (\overline{ABC})(\overline{DEF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{ABC}$ and $\overline{DEF}$.

$$(\overline{ABC})(\overline{DEF}) = (\overline{A} + \overline{B} + \overline{C})(\overline{D} + \overline{E} + \overline{F})$$

31

Applying DeMorgan's Theorems

- Example: Apply DeMorgan's theorems to the following expression:

$\overline{A\overline{B} + \overline{C}D + EF}$

- Solution:

Let $A\overline{B} = X$, $\overline{C}D = Y$, and $EF = Z$. The expression $\overline{A\overline{B} + \overline{C}D + EF}$ is of the form $\overline{X + Y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$ and can be rewritten as

$$\overline{A\overline{B} + \overline{C}D + EF} = (\overline{A\overline{B}})(\overline{\overline{C}D})(\overline{EF})$$

Next, apply DeMorgan's theorem to each of the terms $\overline{A\overline{B}}$, $\overline{\overline{C}D}$, and $\overline{EF}$.

$$(\overline{A\overline{B}})(\overline{\overline{C}D})(\overline{EF}) = (\overline{A} + B)(C + \overline{D})(\overline{E} + \overline{F})$$

32

Applying DeMorgan's Theorems

- Example: Apply DeMorgan's theorems to each of the following expressions:

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

(b) $\overline{(\overline{A}+B)+CD}$

(c) $\overline{(A+B)\overline{C}\,\overline{D}+E+\overline{F}}$

- Solution

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

(b) $\overline{(\overline{A}+B)+CD} = (\overline{\overline{A}+B})\overline{CD} = (\overline{\overline{A}}\,\overline{B})(\overline{C}+\overline{D}) = A\overline{B}(\overline{C}+\overline{D})$

(c) $\overline{(A+B)\overline{C}\,\overline{D}+E+\overline{F}} = \overline{((A+B)\overline{C}\,\overline{D})}(\overline{E+\overline{F}}) = (\overline{A}\,\overline{B}+C+D)\overline{E}F$

33

Applying DeMorgan's Theorem

- Example:
- The Boolean expression for an exclusive-OR gate is $A\overline{B} + \overline{A}B$. With this as a starting point, use DeMorgan's theorems and any other rules or laws that are applicable to develop an expression for the exclusive-NOR gate.

- Solution:

Start by complementing the exclusive-OR expression and then applying DeMorgan's theorems as follows:

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

Next, apply the distributive law and rule 8 ($A \cdot \overline{A} = 0$).

$$(\overline{A}+B)(A+\overline{B}) = \overline{A}A + \overline{A}\,\overline{B} + AB + B\overline{B} = \overline{A}\,\overline{B} + AB$$

The final expression for the XNOR is $\overline{A}\,\overline{B} + AB$. Note that this expression equals 1 any time both variables are 0s or both variables are 1s.

34