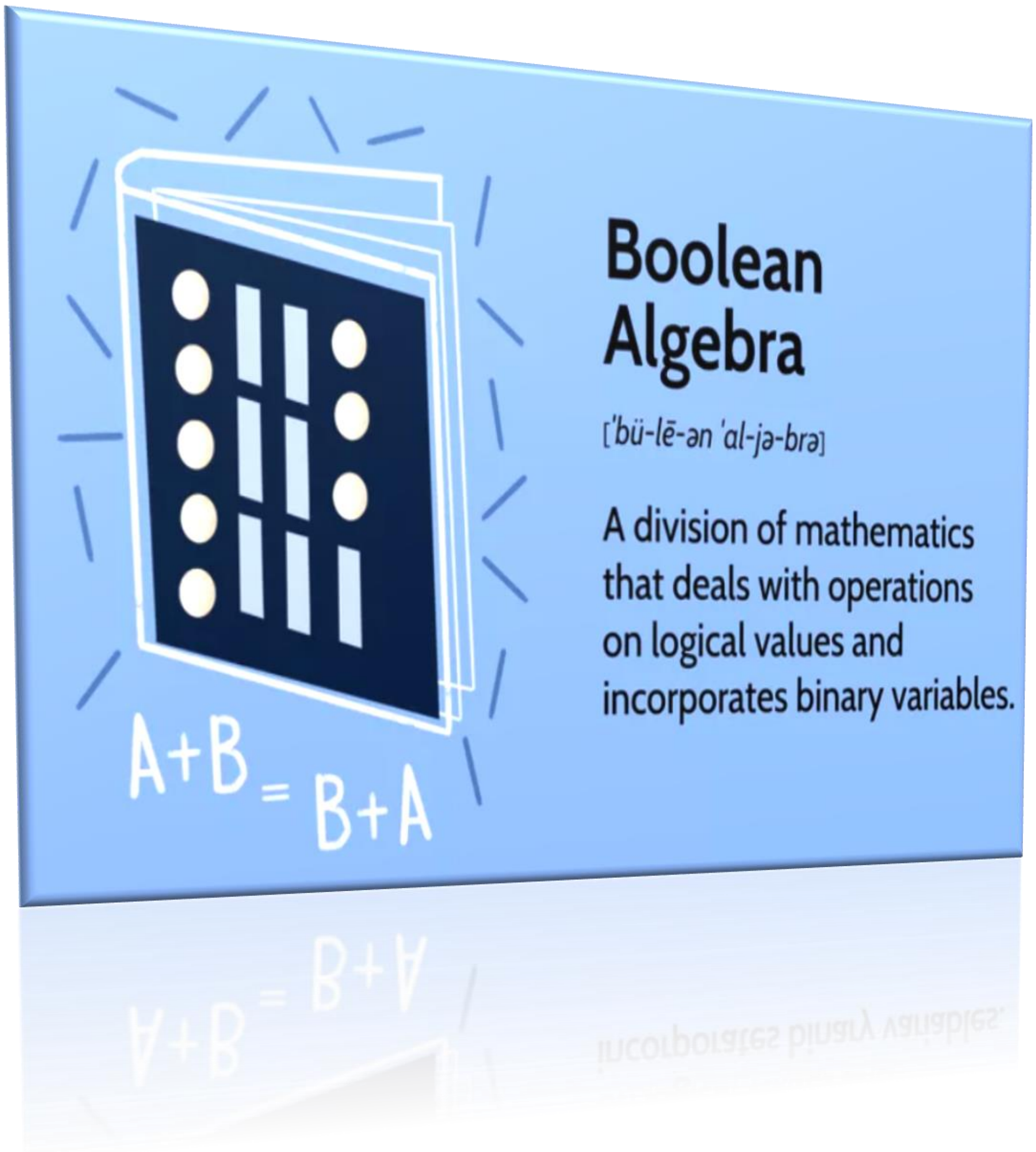


DIGITAL TECHNIQUE

FIRST STAGE/ LECTURE FOUR

NOOR ALHUDA S. ALJAMAAS
ELECTRONIC ENG. DEPT.

Boolean Algebra Laws:

(The mathematics of logic circuits)

Commutative laws: are applied to addition and multiplication. For addition, the commutative law states In terms of the result, the order in which variables are ORed makes no difference.

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

For multiplication, the commutative law states In terms of the result, the order in which variables are ANDed makes no difference.

$$\boxed{AB = BA}$$

Associative laws: are also applied to addition and multiplication. For addition, the associative law states When ORing more than two variables, the result is the same regardless of the grouping of the variables.

$$\boxed{A + (B + C) = (A + B) + C}$$

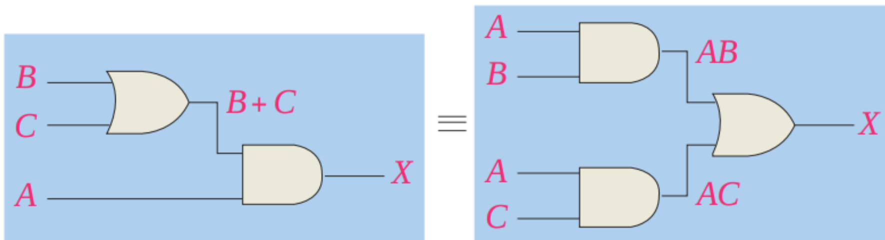
For multiplication, the associative law states When ANDing more than two variables, the result is the same regardless of the grouping of the variables.

$$\boxed{A(BC) = (AB)C}$$

Distributive law: is the factoring law. A common variable can be factored from an expression just as in ordinary algebra. That is

$$\boxed{AB + AC = A(B + C)}$$

The distributive law can be illustrated with equivalent circuits:



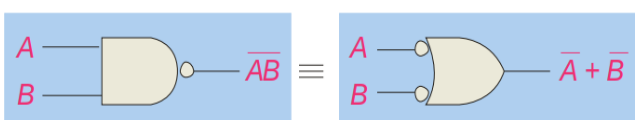
$A(B+C)$ $AB + AC$

DeMorgan's Theorem:

1'st Theorem: The complement of a product of variables is equal to the sum of the complemented variable.

$$\boxed{\overline{A.B} = \overline{A} + \overline{B}}$$

Applying DeMorgan's first theorem to gates:



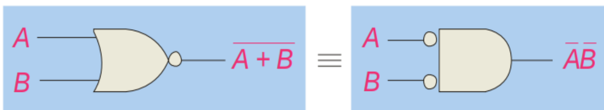
NAND Negative-OR

Inputs		Output	
A	B	$\overline{AB}$	$\overline{A}+\overline{B}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

2'nd Theorem: The complement of a sum of variables is equal to the product of the complemented variables.

$$\boxed{\overline{A + B} = \overline{A} . \overline{B}}$$

Applying DeMorgan's second theorem to gates:



NOR Negative-AND

Inputs		Output	
A	B	$\overline{A+B}$	$\overline{A}\,\overline{B}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

No.	Theorem	Name
1a	$A + B = B + A$	Commutative التبادل
1b	$A \cdot B = B \cdot A$	
2a	$(A + B) + C = A + (B + C)$	Associative الترابط
2b	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	
3a	$A \cdot (B + C) = A \cdot B + A \cdot C$	Distributive التوزيع
3b	$A + (B \cdot C) = (A + B) \cdot (A + C)$	
4a	$A + A = A$	Identity
4b	$A \cdot A = A$	
5a	$(\overline{A}) = \overline{A}$	Negation
5b	$\overline{\overline{A}} = A$	
6a	$A + A \cdot B = A$	Redundancy

No.	Theorem	Name
6b	$A \bullet (A + B) = A$	
7a	$0 + A = A$	
7b	$1 \bullet A = A$	
7c	$1 + A = 1$	
7d	$0 \bullet A = 0$	
8a	$\overline{A} + A = 1$	
8b	$\overline{A} \bullet A = 0$	
9a	$A + \overline{A} \bullet B = A + B$	
9b	$A \bullet (\overline{A} + B) = A \bullet B$	
10a	$\overline{A + B} = \overline{A} \bullet \overline{B}$	De Morgan's theorem
10b	$\overline{A \bullet B} = \overline{A} + \overline{B}$	

Proof by using Boolean Algebra Laws:

Prove: A+A.B =A

=A.(1+B)=A.(1)=A

Prove: A.(A+B) =A

=A.A+A.B=A+A.B=A.(1+B)=A

Prove: $A.(\overline{A}+B)=A.B$

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

Theorem 3b. $(A+B).(A+C)=A+B.C$ **[A + (B • C) = (A + B) • (A + C)]**

(A+B).(A+C)

(A+B).A+(A+B).C

A.A+A.B+A.C+B.C

A+AB+AC+BC

A.(1+B+C)+BC

A.1+BC

A+B.C

Additional Examples on Using Boolean Algebra

$$
\begin{aligned}
1.\quad F &= \overline{\overline{(\overline{AB})A} \cdot \overline{(\overline{AB})B}} \\
&= (\overline{AB})A + (\overline{AB})B \\
&= (\overline{A} + \overline{B})A + (\overline{A} + \overline{B})B \\
&= A\overline{A} + A\overline{B} + \overline{A}B + \overline{B}B \\
&= A\overline{B} + \overline{A}B \\
&= A \oplus B
\end{aligned}
$$

$$2.\ F=\bar{A}\,\bar{B}\,\bar{D}+A\,\bar{B}\,\bar{C}\,\bar{D}+A\,\bar{B}\,C\,\bar{D}$$

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

$$=\bar{B}\,\bar{D}(\bar{A}+A)$$

$$=\bar{B}\,\bar{D}$$

NOTE: Any equations can be proved using Truth table (T.T.)

Proof by Truth Table theorem 6a. A+A.B=A

A	B	$A\bullet B$	$A+A\bullet B$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Proof by Truth Table theorem 9a. $A+\bar{A}.B=A+B$

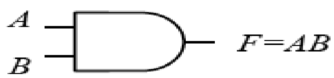
A	B	$A+B$	$\bar{A}$	$\bar{A}\bullet B$	$A+\bar{A}\bullet B$
0	0	0	1	0	0
0	1	1	1	1	1
1	0	1	0	0	1
1	1	1	0	0	1

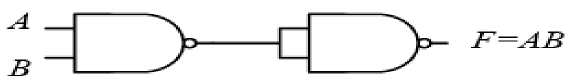
Example: Proof of Demorgan theory: $\overline{A+B} = \overline{A} \cdot \overline{B}$ using (T.T.)

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

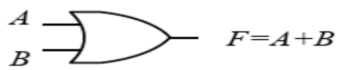
Designing Logic Circuits Using NAND Gates only :

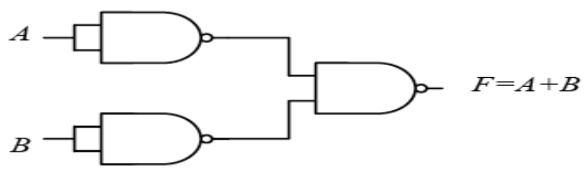
1. AND GATE

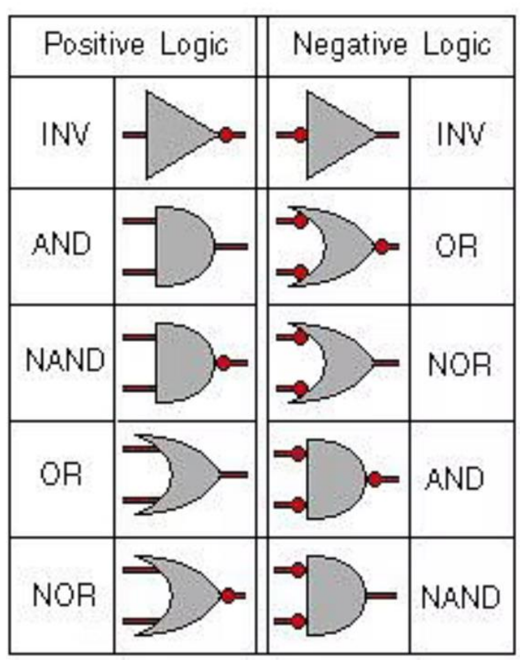


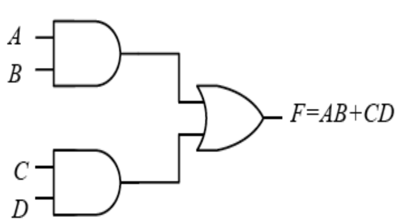


2. OR GATE

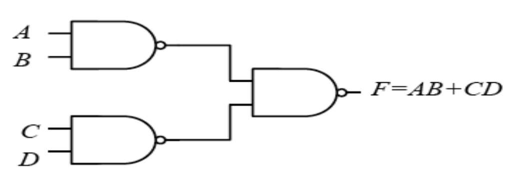

A
B
F=A+B

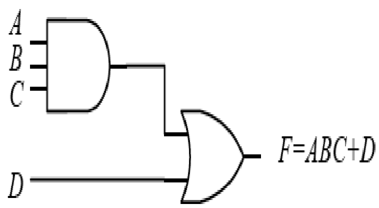

A
B
F=A+B


Positive Logic
Negative Logic
INV
INV
AND
OR
NAND
NOR
OR
AND
NOR
NAND

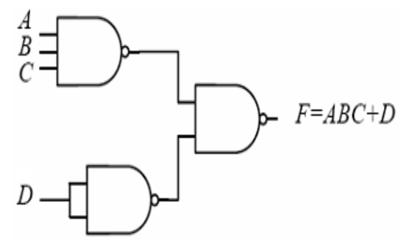

A
B
C
D
F=AB+CD

=


A
B
C
D
F=AB+CD

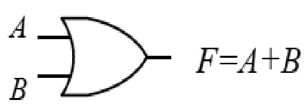


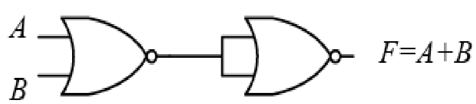
=



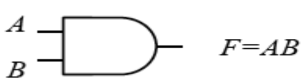
Designing Logic Circuits Using NOR Gates only:

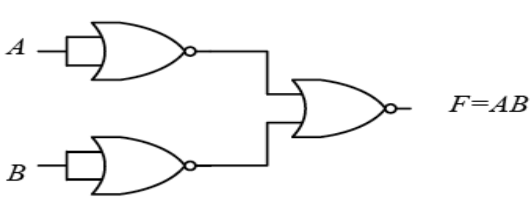
1. OR GATE

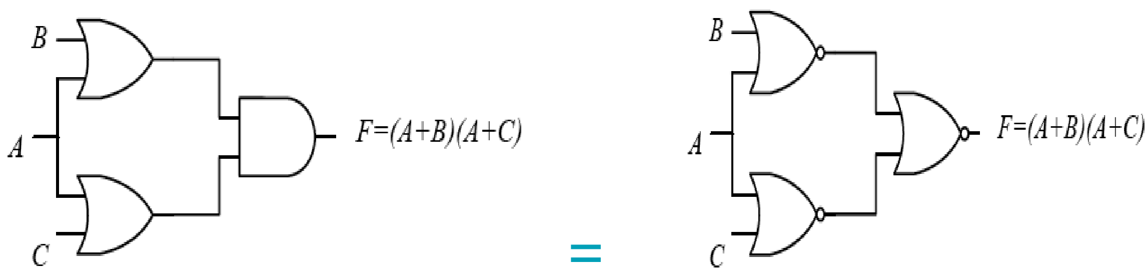




2. AND GATE

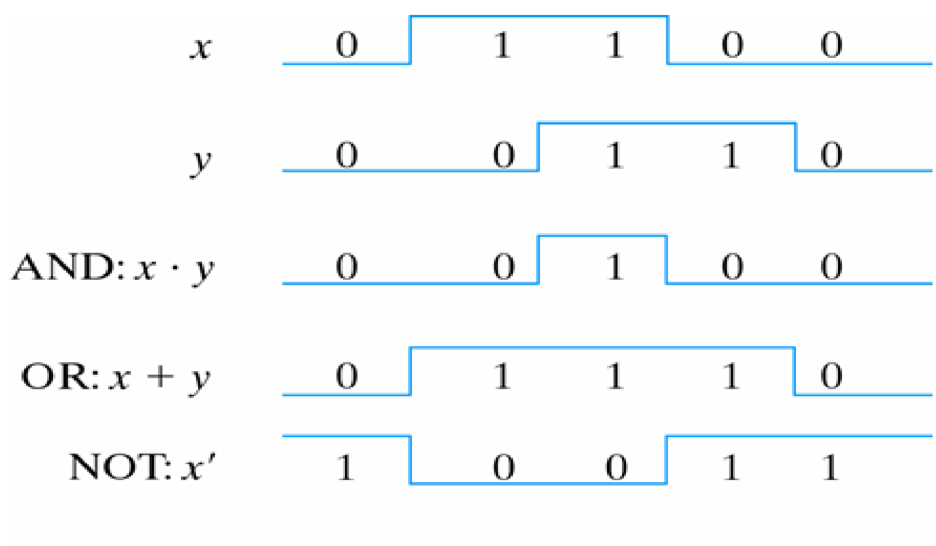




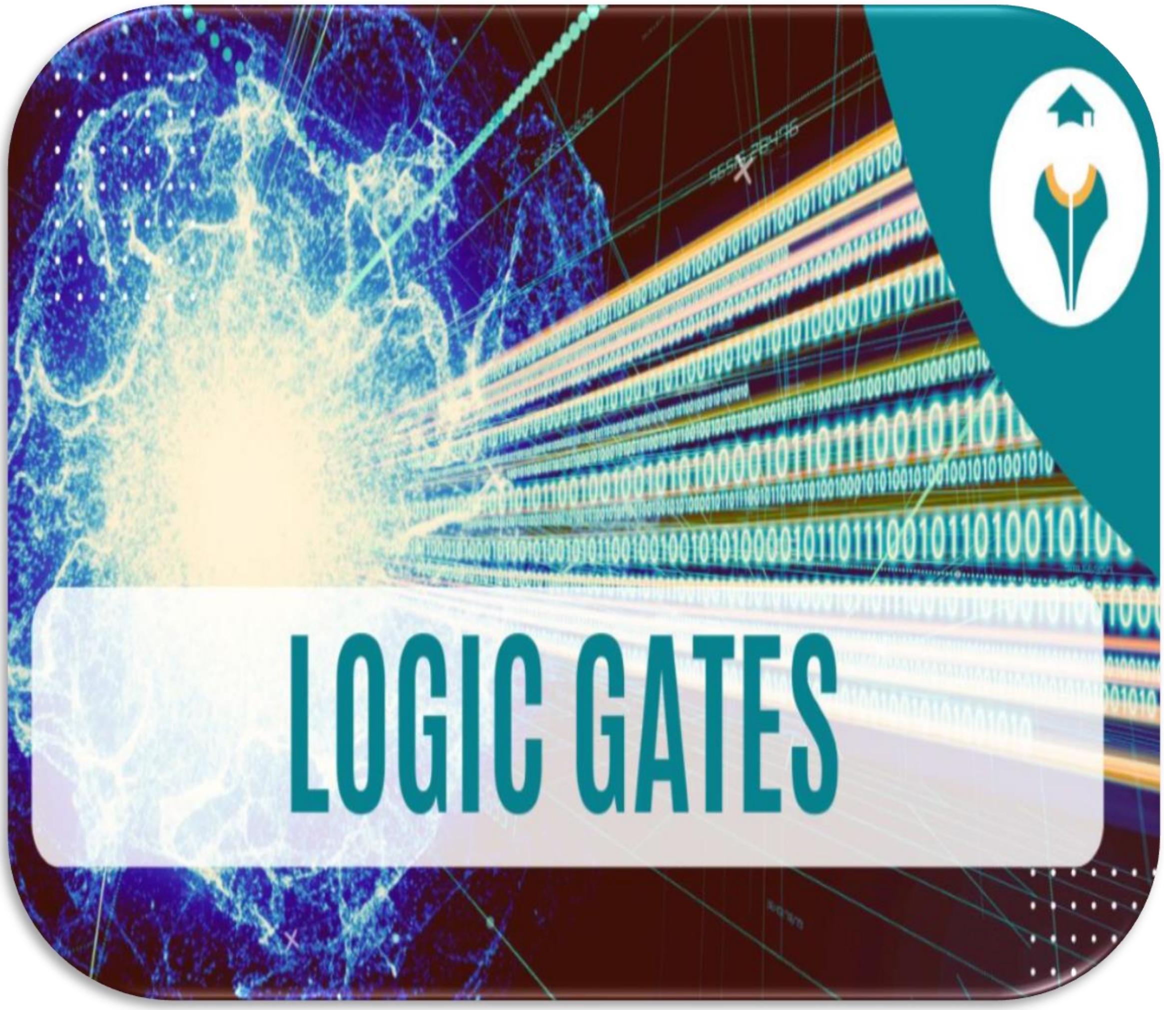


Timing diagram:

Sketch (draw the output waveforms of and ,or & invert):



Input-output signals for gates



DIGITAL TECHNIQUE

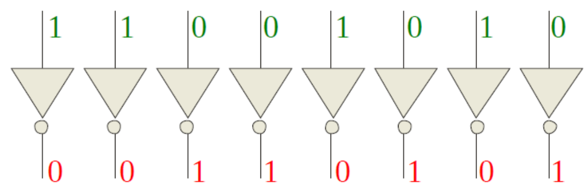
FIRST STAGE/ LECTURE THREE

NOOR ALHUDA S. ALJAMAAS ELECTRONIC ENG. DEPT.

Complements of Binary Numbers

1's Complement:

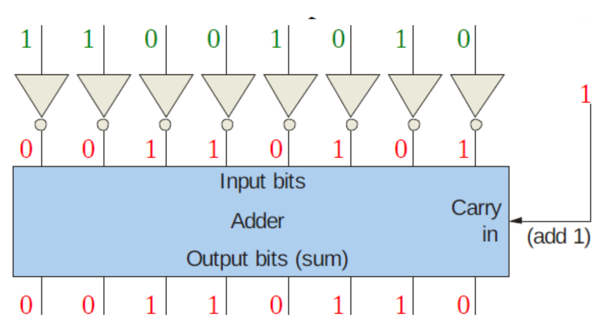
- The 1's complement of a binary number is just the inverse of the digits. To form the 1's complement, change all 0's to 1's and all 1's to 0's.
- For example, the 1's complement of 11001010 is 00110101.
- In digital circuits, the 1's complement is formed by using inverters:



2's Complement:

- The 2's complement of a binary number is found by adding 1 to the LSB of the 1's complement.
- Recall that the 1's complement of 11001010 is 00110101 (1's complement).
- To form the 2's complement, add 1: +1

00110110 (2's complement)



Signed Binary Numbers:

- There are several ways to represent signed binary numbers.
- In all cases, the MSB in a signed number is the sign bit, that tells you if the number is positive or negative.
- Computers use a modified 2's complement for signed numbers. Positive numbers are stored in true form (with a 0 for the sign bit) and negative numbers are stored in complement form (with a 1 for the sign bit).

For example, the positive number 58 is written using 8-bits

as 00111010 (true form)

Sign bit Magnitude bits

Example: 10011out of five bits (sign number machine) = -3_{10}

1011out of four bits (sign number machine) = -3_{10}

And so on. While 1011 out of standard machine $=11_{10}$.

Sign 1's compliment:

- ❖ This type of machine works on other type of negative number.
- ❖ To convert any binary number to a negative each bit must be inverted from 0 to 1 and from 1 to 0.

Example: $00011=3_{10}$ $11100=-3_{10}$

Example: perform this mathematic operation 5-3 on sign 1's compliment machine:

$$\begin{array}{r} 101 \\ -\ 011 \\ \hline \end{array} \qquad \begin{array}{r} 101 \\ +\ 100 \\ \hline 1\ 001 \\ \longrightarrow +\quad 1 \\ \hline 010 \end{array}$$

Look at the result if the far left bit is 0 the result is the same as it appeared.

If the far left bit is 1 the result is the 1's compliment of the number and it is negative.

Sign 2's compliment:

- ❖ This type of machine works on other type of negative number.
- ❖ To convert any binary number to a negative each bit must be inverted from 0 to 1 and from 1 to 0 after that add 1 to the result.

Example: $00011=3_{10}$ $\xrightarrow{\text{2's compliment}}$ $11101=-3_{10}$

Example: perform this mathematic operation 5-3 on sign 2's compliment machine:

$$\begin{array}{r} 101 \\ -\ 011 \\ \hline \end{array} \qquad \begin{array}{r} 101 \\ +\ 101 \\ \hline \cancel{1}\ 010 \end{array}$$

if there is a carry discard it.

Look at the result if the far left bit is 0 the result is the same as it appeared.

If the far left bit is 1 the result is the 2's compliment of the number and it is negative.

Example: perform this mathematic operation 3-6 on sign 2's compliment machine:

$$\begin{array}{r} 011 \\ -\ 110 \\ \hline \end{array} \xrightarrow{2'S} \begin{array}{r} 011 \\ +\ 010 \\ \hline 101 \end{array}$$

if there is a carry discard it.

The far left bit is 1 then the result is negative and the number is the 2's compliment of the result 101 $\xrightarrow{\text{2's compliment}}$ 011=-3_{10}.

Sign 2's compliment (+/-):

Procedures:

- 2's compliment can be used for any signed number using at least n+1 of the number of the digits including the sign bit.
- the far left bit is 0 the result is the same as it appeared.
- If the far left bit is 1 the result is the 2's compliment of the number and it is negative.

Example:

$$\begin{array}{lllll} -3 & \longrightarrow -00011 & \xrightarrow{2's} 11101 & & 11101 \\ -(-5) & \longrightarrow -(-00101) & \xrightarrow{2's} -(11011) & \xrightarrow{2's} & +\ 00101 \\ \hline & & & & \cancel{1}\ 00010 \end{array}$$

Example:

$$\begin{array}{llll} -3 & \longrightarrow -00011 & \xrightarrow{2's} 11101 & 11101 \\ -5 & \longrightarrow -00101 & \xrightarrow{2's} 11011 & +\ 11011 \\ \hline & & & \cancel{1}\ 11000 \xrightarrow{2's} -01000 \end{array}$$

How the numbers appeared om different machines:

- +7 = +111 000111 on signed magnitude machine of 6-bits.
- -7 100111 on signed magnitude machine of 6-bits.
- -7 111000 on sign 1's compliment.
- -7 111001 on sign 2's compliment.

Binary Logic and Gates:

Binary variables take on one of two values. Logical operators operate on binary values and binary variables.

Basic logical operators are the logic functions **AND**, **OR** and **NOT**. Logic gates implement logic functions.

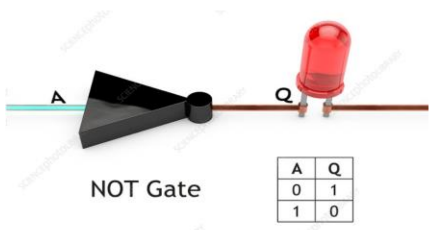
Boolean Algebra: a useful mathematical system for specifying and transforming logic functions. We study Boolean algebra as a foundation for designing and analyzing digital systems.

Truth Table:

The most elementary specification of the function of a Digital Logic Circuit is the Truth Table.

Table that describes the Output Values for all the combinations of the Input Values, called *MINTERMS.*

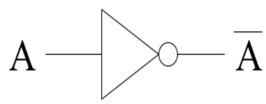
Switching Algebra Operations – Not:



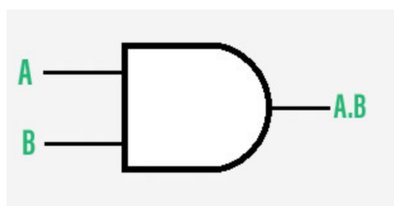
A	Q
0	1
1	0

NOT: This operation is represented by a prime (sometimes by an overbar). For example, $\overline{X}$ = Q (or X' = Q) is read "not X is equal to Q," meaning that Q is what Q is not. In other words, if X = 1, then Q = 0, but if X = 0, then Q = 1. The NOT operation is also referred to as the complement operation, since it changes a 1 to 0 and a 0 to 1, i.e., the result of complementing 1 is 0, and vice versa.

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

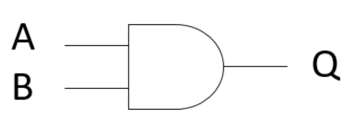


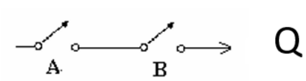
Switching Algebra Operations – AND:



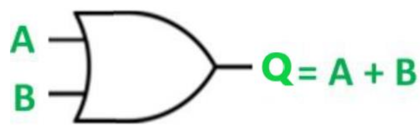
AND: This operation is represented by a dot or by the absence of an operator. For example, A.B = Q or AB = Q is read "A AND B is equal to Q." The logical operation AND is interpreted to mean that Q = 1 if and only if A = 1 and B = 1; otherwise Q = 0. (Remember that A, B, and Q are binary variables and can be equal either to 1 or 0, and nothing else.) The result of the operation A.B is Q. Also known as the conjunction operation; output is true (1) only if all inputs are true.

A	B	Q
0	0	0
0	1	0
1	0	0
1	1	1



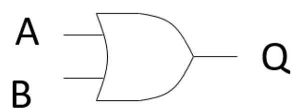


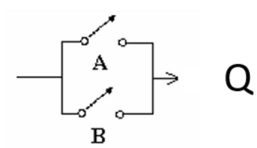
Switching Algebra Operations – OR:



OR: This operation is represented by a plus sign. For example, A+B = Q is read "A OR B is equal to Q," meaning that Q = 1 if A = 1 or if B = 1 or if both A = 1 and B = 1. If both A = 0 and B = 0, then Q = 0.

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

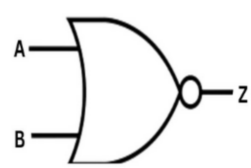




COMBINATIONAL GATES:

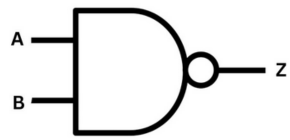
NOR: gate has a single output and two inputs. The output is 1 only when all of the inputs are 0. For all other combinations, the output is 0. The NOR gate's Boolean logic is Z= (A+B)' for inputs A and B.

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0



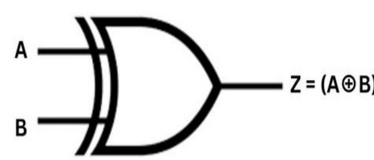
NAND: has a single output and two inputs. The output is 0 only when all of the inputs are 1. For all other combinations, the output is 1. The NAND gate's Boolean logic is Z= (A.B)' for inputs A and B.

A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0



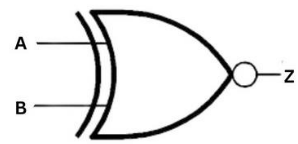
XOR: an abbreviation for "Exclusively-OR." The simplest XOR gate is a two-input digital circuit that outputs a logical "1" if the two input values differ, i.e., its output is a logical "1" if either of its inputs are 1, but not at the same time (exclusively). The Boolean expression for a two-input XOR gate, with inputs A and B and output X.

A	B	Output ($A \oplus B$)
0	0	0
0	1	1
1	0	1
1	1	0

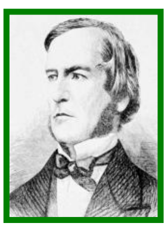


XNOR : has a single output and two inputs and it is called exclusive NOR. The output is 1 only when an even number of inputs are 1. The XNOR gate's Boolean logic for inputs A and B is Z= $(A \oplus B)'$ = A.B+ A'. B'

A	B	Output
0	0	1
0	1	0
1	0	0
1	1	1



Boolean Algebr:

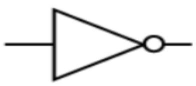


(Developed by George Boole in 1854)

George
Boole
1815-1864

- Formal way to describe logic statements and determine truth of statements.
- Only has two-values domain (0 and 1)
- Algebra with Binary (Boolean) Variable and Logic Operations.
- Boolean Algebra is useful in Analysis and Synthesis of Digital Logic Circuits.

- Input and Output signals can be represented by Boolean Variables, and Function of the Digital Logic Circuits can be represented by Logic Operations, i.e., Boolean Functions.
- From a Boolean function, a logic diagram can be constructed using AND, OR, and I .



NOT

Input	Output
I	F
0	1
1	0

AND

Inputs		Output
A	B	F
0	0	0
1	0	0
0	1	0
1	1	1

NAND

Inputs		Output
A	B	F
0	0	1
1	0	1
0	1	1
1	1	0

OR

Inputs		Output
A	B	F
0	0	0
1	0	1
0	1	1
1	1	1

NOR

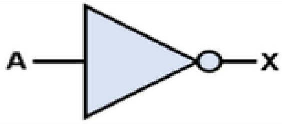
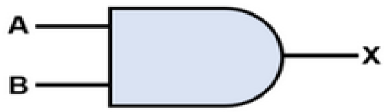
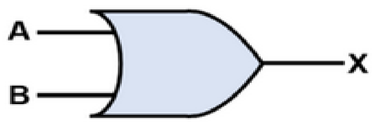
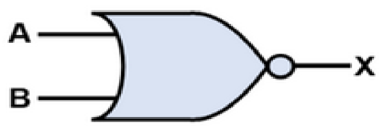
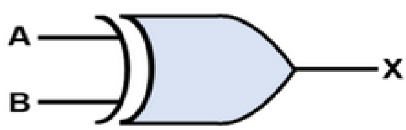
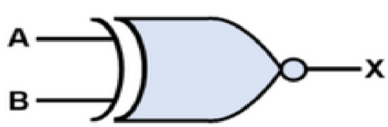
Inputs		Output
A	B	F
0	0	1
1	0	0
0	1	0
1	1	0

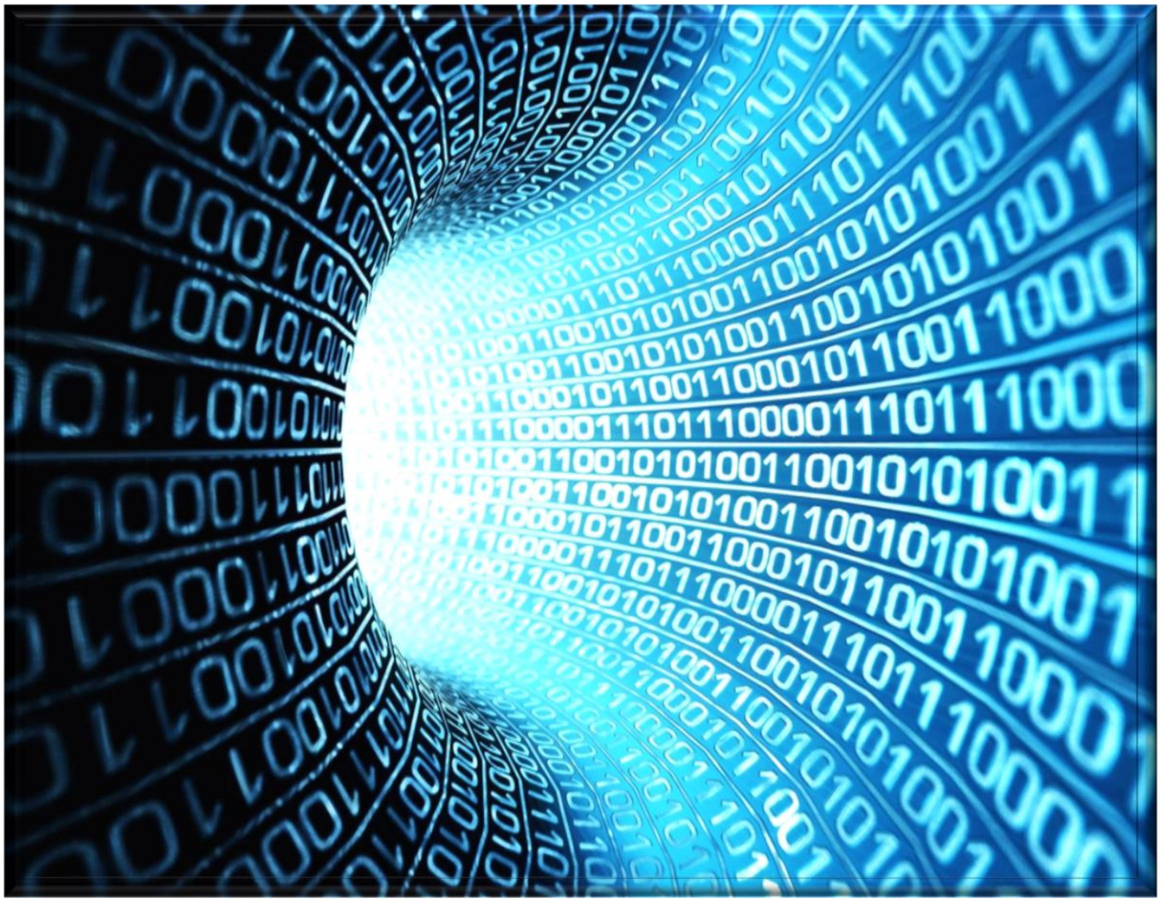
Inputs		Output
A	B	F
0	0	0
0	1	1
1	0	1
1	1	0

EXCLUSIVE OR

EXCLUSIVE NOR

Inputs		Output
A	B	F
0	0	1
0	1	0
1	0	0
1	1	1

Gate	Symbol	Rule
NOT		$0 \rightarrow 1$ $1 \rightarrow 0$
AND		$00 \rightarrow 0$ $01 \rightarrow 0$ $10 \rightarrow 0$ $11 \rightarrow 1$
OR		$00 \rightarrow 0$ $01 \rightarrow 1$ $10 \rightarrow 1$ $11 \rightarrow 1$
NAND		$00 \rightarrow 1$ $01 \rightarrow 1$ $10 \rightarrow 1$ $11 \rightarrow 0$
NOR		$00 \rightarrow 1$ $01 \rightarrow 0$ $10 \rightarrow 0$ $11 \rightarrow 0$
XOR		$00 \rightarrow 0$ $01 \rightarrow 1$ $10 \rightarrow 1$ $11 \rightarrow 0$
XNOR		$00 \rightarrow 1$ $01 \rightarrow 0$ $10 \rightarrow 0$ $11 \rightarrow 1$



DIGITAL TECHNIQUE

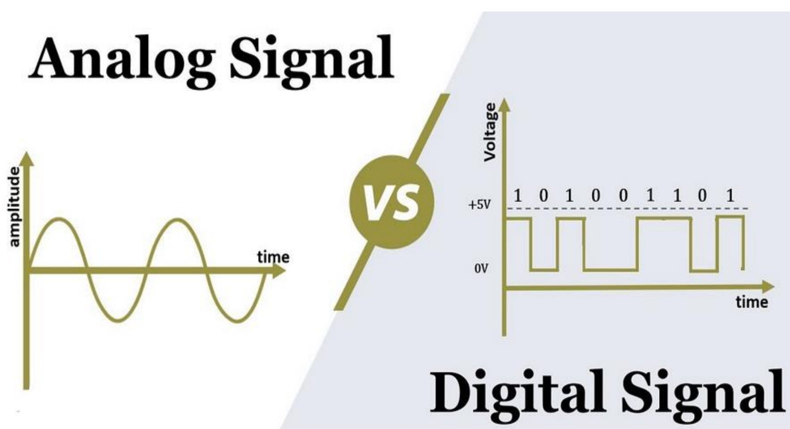
FIRST STAGE/ LECTURE ONE

NOOR ALHUDA S. ALJAMAAS ELECTRONIC ENG. DEPT.

INTRODUCTION

ANALOG SIGNAL VS DIGITAL SIGNAL

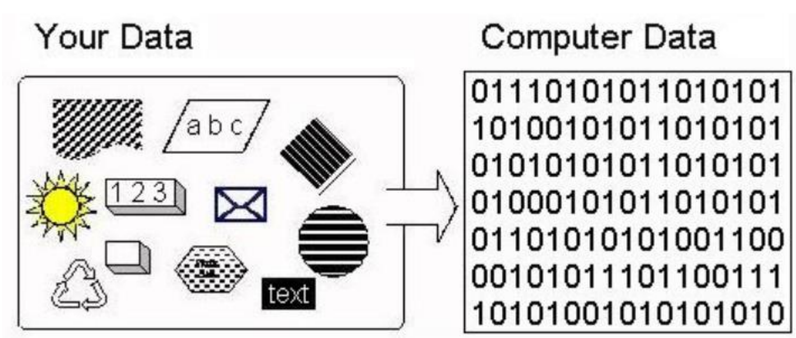
Analog signals were used in many systems to produce signals to carry information. These signals are continuous in both values and time. The use of analog signals has declined with the arrival of digital signals. In short, to understand analog signals – all signals that are natural or come naturally are analog signals. Unlike analog signals, digital signals are not continuous, but signals are discrete in value and time. These signals are represented by binary numbers and consist of different voltage values.



They used to transmit information, usually through electric signals. In both these technologies, the information, such as any audio or video, is transformed into electric signals. The difference between analog and digital technologies is that in analog technology, information is translated into electric pulses of varying amplitude. In digital technology, translation of information is into binary format (zero or one) where each bit is representative of two distinct amplitudes. The analog signal is a varied its amplitude with the time such as 0 volt to 40 volt linearly, while the digital signal has two values of amplitude 0 volt and 5 volt.

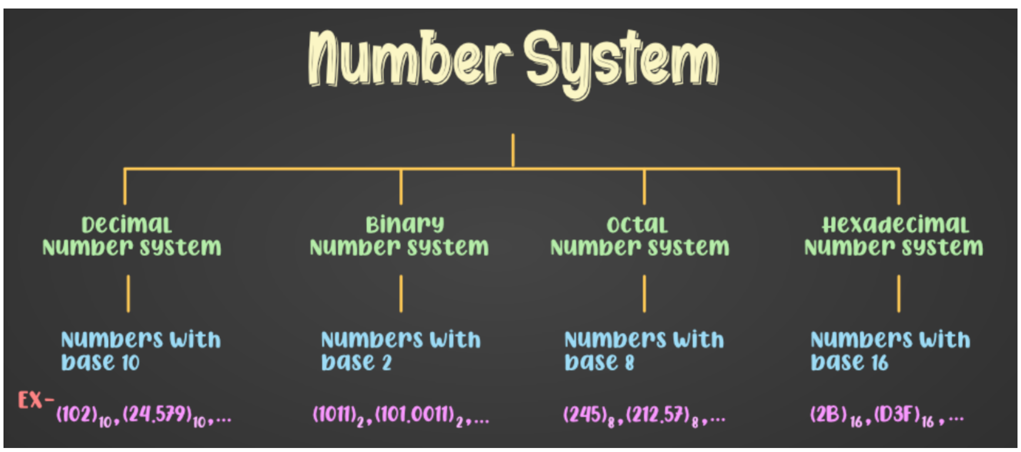
A digital computer stores data in terms of digits (numbers) and proceeds in discrete steps from one state to the next. The states of a digital computer typically involve binary digits which may take the form of the presence or absence of magnetic markers in a storage medium, on-off switches or relays. In digital computers, even letters, words and whole texts are represented digitally.

Digital Logic is the basis of electronic systems, such as computers and cell phones. Digital Logic is rooted in binary code, a series of zeroes and ones each having an opposite value. This system facilitates the design of electronic circuits that convey information, computing, robotics and other electronic applications.



Numeric Systems

- Decimal number system (base 10)
- Binary number system (base 2)
- Octal number system (base 8)
- Hexadecimal number system (base 16)



Decimal number system (Represent any number using 10 digits [0–9])

The numeric system we use daily is the decimal system, but this system is not convenient for machines since the information is handled codified in the shape of on or off bits; this way of codifying takes us to the necessity of knowing the positional calculation which will allow us to express a number in any base where we need it. A base of a number system or radix defines the range of values that a digit may have.

In the decimal system or base 10, there are ten different values for each digit of a number: "0", "1", "2", "3", "4", "5", "6", "7", "8", "9".

$$\ldots\ 10^5\ 10^4\ 10^3\ 10^2\ 10^1\ 10^0$$

$$7*100+3*10+5*1=735_{10}$$
$$1*16+0*8+1*4+0*2+1*1=21_{10}$$

Binary number system (Represent any number using 2 digits [0–1])

The binary number system has a base of 2. The position of a 1 or 0 in a binary number indicates its weight, or value within the number. The weights in a binary number are based on powers of two. Each digit in this system is said to be a bit.

$$\ldots\ 2^3\ 2^2\ 2^1\ 2^0$$

In binary number system we have only two digit 0 and 1. Let us count as before :

Zero: yes we have this symbol 0

One: yes we have this symbol 1

Two: the symbols are finished! What we will do? As we do with decimal number we will put a zero symbol in the first digit and put 1 in the second digit

Two yes we will use two symbols 10

Three increase the 1st digit by one 11

1 1

1 1

+ 1

----

1 0 0 Four

As you can see 1 + 1 equal to 2 and two in binary (10) zero will be in the first digit and one is added to the one of the second digit. The same process is again happen so

Four 100

Five 101

Six 110

And so on up to millions.

Some of you will ask what is this number 100 is it one hundred or four? What is the answer?

Yes you will confuse between the two systems. Accordingly we will use the following:

100_{10} The sub number 10 mean this number is decimal, it is one hundred.

100_2 The sub number 2 mean this number is binary, it is four in binary.

10^3	10^2	10^1	10^0	***Decimal***	2^4	2^3	2^2	2^1	2^0	***Binary***
1000	100	10	1		16	8	4	2	1	

Decimal النظام العشري	Binaryالنظام الثنائي
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001

Fractional numbers can also be represented in binary by placing bits to the right of the binary point, just as fractional decimal digits are placed to the right of the decimal point. The left-most bit is the MSB in a binary fractional number and has a weight of $2^{-1} = 0.5$.

The fractional weights decrease from left to right by a negative power of two for each bit.

Binary weights.

Positive Powers of Two (Whole Numbers)									Negative Powers of Two (Fractional Number)					
2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}
256	128	64	32	16	8	4	2	1	1/2 0.5	1/4 0.25	1/8 0.125	1/16 0.625	1/32 0.03125	1/64 0.015625

How to know the value of a number:

Decimal				Binary				
10^3	10^2	10^1	10^0	2^4	2^3	2^2	2^1	2^0
1000	100	10	1	16	8	4	2	1

Example1: Convert 9_{10} to binary

المتبقي من القسمة على ٢

/2	Result	Remain
9	4	1
4	2	0
2	1	0
1	0	1

$\xrightarrow{1001}$

Example 2: Convert 21_{10} to binary

المتبقي من القسمة على ٢

/2	Result	Remain
21	10	1
10	5	0
5	2	1
2	1	0
1	0	1

$\xrightarrow{10101} = 21_{10}$

10^2	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	power الأس	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
100	1000	1	0.1	0.01	0.001	digit المرتبه	4	2	1	0.5	0.25	0.125	0.0625

Example 3: 21.36_{10} and 101.101_2

21.36_{10}

2*10+1*1+3*0.1+6*0.01

20 + 1 + 0.3 + 0.06

101.101_2

1*4+0*2+1*1+1*1/2+0*1/4+1*1/8

4 + 0 + 1 +1/2+ 0 + 1/8

4 + 0 + 1 + 0.5+ 0 + 0.125

So 101.1012 =5.62510

Example 4: Convert 5.625_{10} to binary

/2	Result	Remain
5	2	1
2	1	0
1	0	1

101 →

0.625 *2	= 1.250	1	0.25
0.25 *2	= 0.5	0	0.5
0.5 *2	= 1	1	0

101.101 →

Addition and Subtraction in binary number system:

Example 5: Add 2 to 3 in binary system.

Decimal	Binary	Addition
$\begin{array}{r} 2 \\ +\ 3 \\ \hline 5 \end{array}$	$\begin{array}{r} 10_2 \\ +\ 11_2 \\ \hline \end{array}$	$\begin{array}{r} 1\ 0 \\ +\ 1\ 1 \\ \hline 1\ 0\ 1 \end{array}$

10

Example 6: Add 7 to 9 in binary system.

10

Decimal	Binary	Addition
$\begin{array}{r} 7 \\ +\ 9 \\ \hline 16 \end{array}$	$\begin{array}{r} 111_2 \\ +\ 1001_2 \\ \hline \end{array}$	$\begin{array}{r} {\scriptstyle 1}\ \ {\scriptstyle 1}\ \ {\scriptstyle 1}\ \ \ \\ 1\ 1\ 1 \\ +\ 1\ 0\ 0\ 1 \\ \hline 1\ 0\ 0\ 0\ 0 \end{array}$

10

Example 7: Subtact 2 from 3 in binary system.

$$\begin{array}{r} 3 \\ -\ 2 \\ \hline 1 \end{array} \qquad \begin{array}{r} 11_2 \\ -\ 10_2 \\ \hline \end{array} \qquad \begin{array}{r} 1\ \ 1 \\ -\ 1\ \ 0 \\ \hline 0\ \ 1 \end{array}$$

Example 8: Subtact 4 to 3 in binary system.

$$\begin{array}{r} 4 \\ -\ 3 \\ \hline 1 \end{array} \qquad \begin{array}{r} 100_2 \\ -\ 11_2 \\ \hline \end{array} \qquad \begin{array}{r} \cancel{1}\ \ 10 \\ 0\ \ \cancel{10}\ \ \ \\ \cancel{1}\ \ 0\ \ 0 \\ -\ \ 1\ \ 1 \\ \hline 0\ \ 0\ \ 1 \end{array}$$

Multiplying and Division in binary system:

Example 9: Multiply 2 by 3 in binary system.

Decimal	Binary	Multiplication
2	10_2	1 0
* 3	* 11_2	* 1 1
		1 0
		+ 1 0 .
		1 1 0

Example 10: Multiply 9 by 5 in binary system.

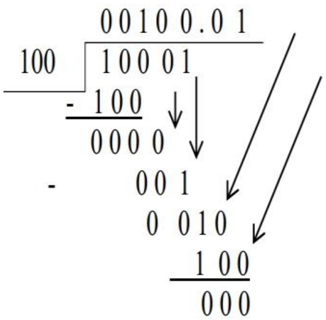
Decimal	Binary	Multiplication
9	1001_2	1 0 0 1
* 5	* 101_2	* 1 0 1
45		1 0 0 1
		0 0 0 0
		+ 1 0 0 1 .
		1 0 1 1 0 1 = 45

32*1+0*16+1*8+1*4+0*2+1*1=45

Example 11: Divide 9 by 3 in binary system.

```
        0 0 1 1
     ┌─────────
 11  │  1 0 0 1
      - 1 1   ↓
          0 1 1
      -     1 1
          0   0
```

Example 12: Divide 1710 by 410 in binary system.



Answer 17 / 4 = $100.01_2 = 4.25_{10}$

DIGITAL TECHNIQUE

FIRST STAGE/ LECTURE TWO

NOOR ALHUDA S. ALJAMAAS ELECTRONIC ENG. DEPT.

Some Definitions in Binary system

- Binary value: two values only ‘0’, ‘1’.
- Bit: single binary digit.
- Nibble: A group of four bits.(0110)
- Byte: A group of 8 bits(11010100).
- Word: Depends on processor (8,16,32 or 64 bits).
- LSB: Least significant bit (on the right).
- MSB: Most significant bit (on the left).

(10000100)

MSB **LSB**

Why the binary system is used with computer?

It is used because the electrical signals used by the electronic circuits have a binary or digital nature.

Advantage of binary systems:

- Rigorous mathematical foundation based on logic.
- It’s easy to implement.

Powers of 2:

$2^0 = 1$	$2^4 = 16$	$2^8 = 256$
$2^1 = 2$	$2^5 = 32$	$2^9 = 512$
$2^2 = 4$	$2^6 = 64$	$2^{10} = 1024$
$2^3 = 8$	$2^7 = 128$	

Octal system number:

Octal system is used in software and in printing data. Octal number system consists of 8 symbols They are 0, 1, 2, 3, 4, 5, 6, 7.

Let us recall some information:

In Decimal number system the base is 10 or we can call it **(radix).** In binary number system the base or the radix is 2. So what is the base for this system?

Octal radix=8 and $8=2^3$ so each digit in Octal can be represented by 3 bits of binary number.

The conversion from binary to octal is easily accomplished by partitioning the binary number into groups of three digits each, starting from the binary point and proceeding to the left and to the right. The corresponding octal digit is then assigned to each group. The following example illustrates the procedure:

$$\begin{array}{cccccccccc} (10 & 110 & 001 & 101 & 011 & \cdot & 111 & 100 & 000 & 110)_2 = (26153.7406)_8 \\ 2 & 6 & 1 & 5 & 3 & & 7 & 4 & 0 & 6 \end{array}$$

Example 1: Convert $(127.4)_8$ to decimal number.

$(127.4)_8 = (1 \text{ x } 8^2) + (2 \text{ x } 8^1) + (7 \text{ x } 8^0) + (4 \text{ x } 8^{-1})$

$= 64 + 16 + 7 + 0.5$

$= (87.5)_{10}$

Example 2: Convert 517_8 to binary number.

5 1 7

101 001 111

$517_8 = 101001111_2$

Example 3: Convert $(153.513)_{10}$ to octal number.

Integer	/8	Reminder
153	19	1
19	2	3
2	0	2

Fractional

$0.513 \times 8 = 4.104$
$0.104 \times 8 = 0.832$
$0.832 \times 8 = 6.656$
$0.656 \times 8 = 5.248$

Thus $(153.513)_{10} = (231.4065\ldots)_8$

Hexadecimal System Number:

Hexadecimal number system is used in software and in printing data. It consists of 16 symbols. They are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F Where A=10, B=11, C=12, ……F=15. Any number in hexadecimal number may be written in this form 169_{16} or 169H.

For Hexadecimal radix=16 and $16=2^4$ so each digit in Octal can be represented by 4 bits of binary number.

Conversion from binary to hexadecimal is similar, except that the binary number is divided into groups of four digits:

$$\begin{array}{cccccccc} (10 & 1100 & 0110 & 1011 & \cdot & 1111 & 0010)_2 & = (2C6B.F2)_{16} \\ 2 & C & 6 & B & & F & 2 & \end{array}$$

Each octal digit is converted to its three-digit binary equivalent. Similarly, each hexadecimal digit is converted to its four-digit binary equivalent. The procedure is illustrated in the following examples:

$$\begin{array}{ccccccccc} (673.124)_8 = & (110 & 111 & 011 & \cdot & 001 & 010 & 100)_2 \\ & 6 & 7 & 3 & & 1 & 2 & 4 \end{array}$$

$$\begin{array}{cccccc} (306.D)_{16} = & (0011 & 0000 & 0110 & \cdot & 1101)_2 \\ & 3 & 0 & 6 & & D \end{array}$$

Example 4: Convert $(5F)_{16}$ to decimal.

$(5F)_{16} = 5*16^1+15*16^0=80+15=(95)_{10}$

Example 5: Convert F5E2H to binary number.

F 5 E 2

1111 0101 1110 0010 $F5E2_{16} = 1111010111100010_2$

To convert from binary to hex, make groups of 4 bits, starting from the binary point. Add 0s to the ends of the number if needed. Then, just convert each bit group to its corresponding hex digit.

$(10110100.001011)_2 \quad = \quad (1011 \;\; 0100 \;.\; 0010 \;\; 1100)_2$

$\quad = \quad (\,B \quad 4 \quad . \quad 2 \quad C\,)_{16}$

Conversions between binary, octal, and hexadecimal can be done in a simpler way.

Example 6: Convert $(10111010011)_2$ to octal and hexadecimal.

010 111 010 011 $(2723)_8$

0101 1101 0011 $(5D3)_{16}$

Example 7: Convert $(12A7F)_{16}$ to binary and octal.

1 2 A 7 F

0001 0010 1010 0111 1111 binary

00 010 010 101 001 111 111 octal

Binary Coded Decimal: (BCD or 8421 code)

Although the binary number system is the most natural system for computer, most people are more accustomed to the decimal system.

One way to resolve this difference between (binary& decimal) is to convert the decimal numbers to binary.

Since the computer can accept only binary values, we must represent the decimal digits by means of a code that contains 1s and 0s.

This type of number system is a decimal number but it is written with binary symbols. Each digit of decimal must be written with 4 symbol of binary number. Why? That is because the largest symbol in decimal is 9 and this number in binary is 1001 so its four.

Example 8: Convert $(7\ 2\ 1_{10})$ to BCD.

$7\ 2\ 1_{10}$ = 0111 0010 0001 the space between the digits must be exist. If you write 721= ~~011100100001~~ this is wrong because there are no space between digits.

- The digits 0-9 are represented with their 4-bit equivalents 0000-1001 (same as hex digits 0-9).
- The values of the digit positions are taken as powers of 10, rather than 16.
- There is an advantage in the use of decimal numbers because computer input and output data are generated by people that use the decimal system.
- The BCD numbers are decimal numbers and not binary numbers although they use bits in their representation.
- BCD is more human-readable than hex or binary, but arithmetic hardware for plain binary is simpler.

$$(358)_{10} = (0011,0101,1000) \text{ in BCD.}$$

Decimal	Binary	Octal	Hexadecimal	BCD
0	0	0	0	0000
1	1	1	1	0001
2	10	2	2	0010
3	11	3	3	0011
4	100	4	4	0100
5	101	5	5	0101
6	110	6	6	0110
7	111	7	7	0111
8	1000	10	8	1000
9	1001	11	9	1001
10	1010	12	A	0001 0000
11	1011	13	B	0001 0001
12	1100	14	C	0001 0010
13	1101	15	D	0001 0011
14	1110	16	E	0001 0100
15	1111	17	F	0001 0101

Gray Code system:

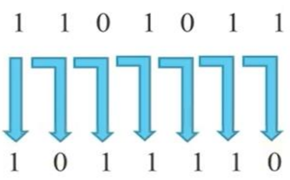
Reflected binary code or Gray code orders the binary numeral system.

- Successive Gray code values differ in only one binary digit. It is non weighted code and not arithmetic code.
- Gray codes are beneficial in hardware-generated binary sequences to prevent errors or ambiguity during transitions.
- Gray code ensures that only one bit changes value during any transition between two numbers, resolving potential problems efficiently (reduce the switching operation).

The most significant bit (left-most) in the Gray code is the same as the corresponding MSB in the binary number.

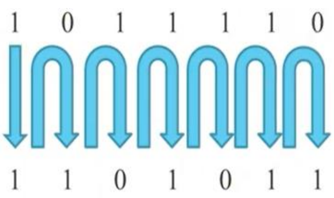
Going from left to right, add each adjacent pair of binary code bits to get the next Grey code bit. Discard carry.

Example 10: Convert 1101011 to Gray code.



On likewise number put 0 otherwise 1

Example 11: Convert this Gray code 1011110 to binary.



On likewise number put 0 otherwise 1

Gray Code

Gray Code	Decimal Equivalent
0000	0
0001	1
0011	2
0010	3
0110	4
0111	5
0101	6
0100	7
1100	8
1101	9
1111	10
1110	11
1010	12
1011	13
1001	14
1000	15

ASCII Codes:

ASCII, stands for American Standard Code for Information Interchange. It is a 7-bit character code where each individual bit represents a unique character.

Many applications of digital computers require the handling not only of numbers, but also of other characters or symbols, such as the letters of the alphabet.

The standard binary code for the alphanumeric characters is the American Standard Code for Information Interchange (ASCII), which uses seven bits to code 128 characters . The seven bits of the code are designated by b1 through b7, with b7 the most significant bit. The letter A, for example, is represented in ASCII as 1000001.The ASCII code also contains 94 graphic characters that can be printed and 34 nonprinting characters used for various control functions. The graphic characters consist of the 26 uppercase letters (A through Z), the 26 lower case letters (a through z), the 10 numerals (0 through 9), and 32 special printable characters, such as %, *, and $.

Control Characters				Graphic Symbols											
Name	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex	Symbol	Dec	Binary	Hex
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(	40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09	)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	–	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[	91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C	|	124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D	]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F

الفصل الأول

أنظمة الأعداد

Number Systems

الفصل الأول

أنظمة الأعداد Number Systems

1-1 مقدمة:

في هذا الفصل سنتطرق إلى **أنظمة الأعداد**؛ وتعرف بأنها طرق تمثيل الأعداد وكتابتها، وهي: النظام العشري والنظام الثنائي والنظام الثماني والنظام الست عشري، والتحويل فيما بين هذه الأنظمة، كذلك سوف نتحدث عن بعض العمليات التي تتم على الأعداد الثنائية وهي المتممة الأولى والمتممة الثنائية والجمع والطرح.

وتستخدم هذه الأنظمة في الحاسوب الآلي، وتستطيع بعضها التحكم في عمل المسجلات Registers، فهي السبيل للكتابة أو القراءة من المسجلات وخاصة النظام السادس عشر.

2-1 أساس أنظمة الأعداد:

تقوم فكرة أي نظام عد على مبدأين أساسين هما: **أساس النظام Radix** (عدد صحيح موجب يعبر عن نظام العد) **وعدد رموز** هذا النظام، فالأساس:

$$(Y)_X \xleftarrow{\text{الأساس}}$$

X= 2 في النظام الثنائي،

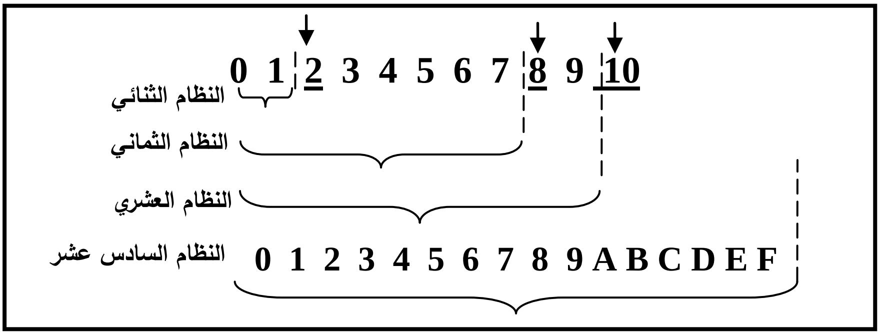
= 8 في النظام الثماني،

= 10 في النظام العشري،

= 16 في النظام السادس العشر.

ويأخذ Y عدد أو مجموعة من الأعداد والرموز حسب النظام، وبما ان أساس النظام الثنائي هو العدد (2)، فان هذا النظام يضم عددان فقط هما (0 و1)، وان أساس النظام الثماني هو العدد (8)، فان اكبر رقم في هذا النظام هو (7). وان أساس النظام العشري هو العدد (10)، فان اكبر رقم في هذا النظام هو (9). وان أساس النظام السادس عشر

هو العدد (16)، إذ أن هذا النظام يتكون من 15 رمز تتكون من تسعة أرقام أكبرها العدد (9) ومن ستة أحرف تكتب بصورة كبيرة هي (A→F). الشكل (1-1).



الشكل (1-1)

مثال (1): عين أنظمة الأعداد الأتية:

$(341)_{10}$: عشري (لان الأساس 10)

$(10)_2$: ثنائي (لان الأساس 2)

$(152)_8$: ثماني (لان الأساس 8).

وفيما يأتي شرح لكل نظام من أنظمة الأعداد.

3-1 **النظام العشري (Decimal System)**:

هو النظام العادي الذي نستخدمه في حياتنا اليومية وفي اغلب أمورنا، ويمثل الرقم (10) الأساس، ويتكون من الأرقام: **0 1 2 3 4 5 6 7 8 9**. وعدد مكونات النظام العشري هو عشرة أرقام، وهذا هو سبب تسميته بهذا الاسم إذ انه يكبر بعد كل عشرة أرقام(*).

(*) توضيح بسيط:

0 1 2 3 4 5 6 7 8 9 10 →11

عند الانتهاء من الأرقام (أخرها الرقم 9) رجعنا للرقم الأول وهو صفر وأضفنا واحد بجواره، ولو واصلنا العد لوصلنا إلى 19 وثم نرجع الرقم 9 إلى صفر ونضيف واحد إلى الرقم 1 فيصبح الرقم 20 وهكذا.

1-4 النظام الثنائي (Binary System):

هو نظام يعتمد على أول رقمين في العد وهم (0 و1)، وكما ذكرنا سابقاً فان الأساس هذا النظام هو العدد (2)، ويكون العد كالآتي:

$$(0000)_2 = (0)_{10}$$
$$(0001)_2 = (1)_{10}$$
$$(0010)_2 = (2)_{10}$$
$$(0011)_2 = (3)_{10}$$

$(0101)_2 = (4)_{10}$، وهكذا ...

1-4-1 تحويل الأعداد من النظام العشري إلى النظام الثنائي:

نستخدم الجدول الآتي للحصول على قيمة العدد بالنظام الثنائي من النظام العشري.

n	...	11	10	9	8	7	6	5	4	3	2	1	المرتبة	
	...	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		
	...	1024	512	256	128	64	32	16	8	4	2	1		
الأعداد بالنظام الثنائي $(Y)_2$													0	
												1	1	الأعداد بالنظام العشري
											1	0	2	
											1	1	3	
										1	0	0	4	
										1	0	1	5	
										1	1	0	6	
										1	1	1	7	
									1	0	0	0	8	
									1	0	0	1	9	
									1	0	1	0	10	
								...					...	
								1	0	1	0	1	21	
													...	
							1	0	1	1	1	1	47	
...						...							...	
					1	1	0	1	1	1	0	0	200	
...						...							...	
			1	0	0	0	0	0	0	1	1	0	521	
													...	
		1	0	0	0	0	0	0	0	0	0	0	1024	

وهكذا إلى n من المراتب أو لأي قيمة لـ Y.

ملاحظة: الرقم الموجود ضع عنده الرقم (1) واذا كان غير موجود ضع عنده (0).

مثال (2): حول الأعداد الأتية إلى النظام العشري: $(1001)_2$، $(1001)_2$

ملاحظة: للتحويل من النظام الثنائي إلى النظام العشري يتم كالاتي:

– ضرب كل رقم من الأرقام في الأساس 2.

– رفع العدد (2) الذي يبدأ من (n-1) إلى (0) (هنا n=4)، أي بصورة عامة:

$$101010001\ldots01 = 1\times2^{n-1}+0\times2^{n-2}+\ldots+0\times2^{1}+1\times2^{0}$$

a) العدد $(1001)_2$ مكتوب بالنظام الثنائي:

$$\begin{aligned} 1001 &= 1\times2^3 + 0\times2^2 + 0\times2^1 + 1\times2^0 \\ &= 1\times8 + 0\times4 + 0\times2 + 1\times1 \\ &= 8 + 0 + 0 + 1 \\ &= 9 \end{aligned}$$

$\therefore (1001)_2 = (9)_{10}$

b) العدد $(111)_2$

$$\begin{aligned} 111 &= 1\times2^2 + 1\times2^1 + 1\times2^0 \\ &= 1\times4 + 1\times2 + 1\times1 \\ &= 4 + 2 + 1 \\ &= 7 \end{aligned}$$

$\therefore (111)_2 = (7)_{10}$

مثال (3): حول الأعداد الأتية إلى ما يقابلها بالنظام الثنائي:

$(17)_{10}$، $(35)_{10}$، $(138)_{10}$، $(500)_{10}$

الحل:

$(17)_{10} \Rightarrow 17 = 16 + 1$

$\therefore (17)_{10} = (10001)_2$

$(35)_{10} \Rightarrow 35 = 32 + 2 + 1$

$\therefore (35)_{10} = (100011)_2$

$(138)_{10} \Rightarrow 138 = 128 + 8 + 2$

$\therefore (138)_{10} = (10001010)_2$

$(500)_{10} \Rightarrow 500 = 256 + 128 + 64 + 32 + 16 + 4$

$\therefore (500)_{10} = (111110100)_2$

طريقة أخرى: نقوم بقسمة العدد على 2 وتسجيل الباقي (أما 1 أو 0). ونستمر بالقسمة على 2 إلى أن يصبح الناتج 1 وهذا لا يقبل القسمة على 2 فيحول إلى جانب الباقي، كما في الأمثلة الأتية.

مثال (4): $(35.375)_{10}$, $(41)_{10}$, $(35)_{10}$, $(13)_{10}$, $(59)_{10}$.

a) $(59)_{10} = (?)_2$

2	
59	
29	1
14	1
7	0
3	1
1	1
0	1

$\therefore (59)_{10} = (111011)_2$

تم تحويل العدد 59 من النظام العشري إلى الثنائي والطريقة هي القسمة على أساس النظام المحول اليه. إذ تم قسمة العدد 59 على أساس النظام المحول اليه وهو العدد 2 وكان ناتج عملية القسمة =29.5.

وهنا المطلوب عدد صحيح بدون كسور نقوم بعد ذلك بضرب العدد الكسري الناتج من عملية القسمة وهو العدد 0.5 في أساس النظام المحول اليه وهو العدد 2 وناتج عملية الضرب هو العدد 1 سوف يكون باقي عملية القسمة ويكتب في الطرف الثاني على يمين الأعداد ثم نقوم بقسمة ناتج عملية القسمة الأول 29 على 2 وهكذا مع باقي نواتج عملية القسمة إلى ان يكون ناتج القسمة يساوي 0 والباقي يساوي 1 (باقي عمليات القسمة) والعدد (111011) يمثل العدد 59 في النظام العشري.

ملاحظات: لابد مراعاة الخطوات الأتية أثناء عملية التحويل:

1- لو كان ناتج عملية القسمة عدد صحيح بدون كسر كما في المثال السابق.

$14/2 = 7$

ولا يوجد باقي أي يساوي 0.

2- عند كتابة ناتج عملية التحويل يكتب العدد من الأسفل إلى الأعلى.

b) $\mathbf{(13)_{10} = (?)_2}$

2	13	الباقي
2	6	1
2	3	0
2	1	1
	0	1

$\therefore (13)_{10} = (1101)_2$

c) $\mathbf{\therefore (35)_{10} = (?)_2}$

2	35	الباقي
2	17	1
2	8	1
2	4	0
2	2	0
2	1	0
	0	1

$\therefore (35)_{10} = (100011)_2$ (يطابق الحل السابق)

d) $\mathbf{(41)_{10} = (?)_2}$

2	41	الباقي
2	20	1
2	10	0
2	5	0
2	2	1
2	1	0
	0	1

$\therefore (41)_{10} = (101001)_2$

e) $\mathbf{(35.375)_{10} = (?)_2}$

العدد مكون من جزئيين: صحيح وكسري، وعند عملية التحويل إلى الثنائي نعامل كل جزء لوحده.

$$\begin{array}{r|l} 2 & \\ 35 & \\ 17 & 1 \\ 8 & 1 \\ 4 & 0 \\ 2 & 0 \\ 1 & 0 \\ 0 & 1 \end{array}$$

$$0.375x2 = 0.75 \rightarrow 0$$
$$0.75x2 = 1.5 \rightarrow 1$$
$$0.5x2 = 1 \rightarrow 1$$
$$(0.011)$$

$\therefore (35.375)_{10} = (100011.011)_2$

تمرين (1): حول الأعداد الأتية إلى ما يقابلها بالنظام الثنائي:

$(117)_{10}$ ،$(335)_{10}$ ،$(1138)_{10}$ ،$(2113)_{10}$ ،$(5211)_{10}$

2-4-1 العمليات الحسابية في النظام الثنائي:

A) **عملية الجمع Addition:** تتم باعتماد الأسس الأتية:

$$\begin{array}{r} 0 \\ +\ 0 \\ \hline 0 \end{array} \qquad \begin{array}{r} 0 \\ +\ 1 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 0 \\ \hline 1 \end{array} \qquad \begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array} \qquad \begin{array}{r} 1 \\ 1 \\ +\ 1 \\ \hline 11 \end{array}$$

ملاحظة: لاحظ ان: a) $(1 + 1 = 10)$ أي 2 بالنظام الثنائي.

b) $(1+1+1= 11)$ أي 3.

مثال (5): الأمثلة على عملية جمع العددين:

1) $\begin{array}{r} (1001)_2 \\ +\ (110)_2 \\ \hline (1111)_2 \end{array}$ 2) $\begin{array}{r} (10011)_2 \\ +\ (101)_2 \\ \hline (11000)_2 \end{array}$ 3) $\begin{array}{r} (1011)_2 \\ +\ (11)_2 \\ \hline (1110)_2 \end{array}$

4) $\begin{array}{r} (110)_2 \\ +\underline{(100)_2} \\ (1010)_2 \end{array}$ 5) $\begin{array}{r} {\scriptstyle 1\,1} \\ (111)_2 \\ \underline{(11)_2} \\ (1010)_2 \end{array}$ 6) $\begin{array}{r} (100)_2 \\ +\underline{(10)_2} \\ (110)_2 \end{array}$ 7) $\begin{array}{r} {\scriptstyle 1} \\ (11)_2 \\ +\underline{(11)_2} \\ (110)_2 \end{array}$

B) **عملية الطرح Subtraction**: تتم باعتماد الأسس الأتية:

$\begin{array}{r} 1 \\ -\underline{\;1} \\ 0 \end{array}$ $\begin{array}{r} 0 \\ -\underline{\;0} \\ 0 \end{array}$ $\begin{array}{r} 1 \\ -\underline{\;0} \\ 1 \end{array}$ $\begin{array}{r} 0 \\ -\underline{\;1} \\ \text{غير ممكن} \end{array}$ $\begin{array}{r} 10 \\ -\underline{\;1} \\ 1 \end{array}$

ملاحظة: ان عملية (0-1) لا يمكن أجراؤها، لهذا نستعير 1 من الرتبة المجاورة، إذ ان المرتبة التالية هي المرتبة (الثانية) وعندما يتم نقلها للمرتبة الأولى تصبح (10) ويبقى صفر في مكانها الأصلي.

مثال (6): الأمثلة على عملية طرح العددين:

1) $\begin{array}{r} 101 \\ -\underline{\;\;11} \\ 10 \end{array}$ 2) $\begin{array}{r} 100 \\ -\underline{\;\;1} \\ 11 \end{array}$ 3) $\begin{array}{r} 100101 \\ -\underline{\;\;1011} \\ 11010 \end{array}$

ملاحظة: يمكن تحويل العددين إلى **النظام العشري** ثم يتم طرحهما، بعد ذلك يحول الناتج إلى النظام الثنائي. مثلاً في المثال (5) فقرة 2 نلاحظ أن العدد $(100)_2 = (4)_{10}$، والعدد $(1)_2 = (1)_{10}$. إذ 4-1 =3 بالنظام العشري أي ان الناتج بالنظام الثنائي هو (11).

مثال (7):

$\begin{array}{r} {\scriptstyle 10} \\ 1\cancel{1}0 \\ \underline{-101} \\ 001 \end{array}$ $\begin{array}{r} {\scriptstyle 10} \\ 1\cancel{0}1 \\ \underline{-011} \\ 010 \end{array}$ $\begin{array}{r} 11 \\ \underline{-10} \\ 01 \end{array}$ $\begin{array}{r} 11 \\ \underline{-01} \\ 10 \end{array}$

تمرين (2): جد نواتج العمليات الأتية؟

(1) $(1000)_2 - (11)_2$

(2) $11011 + 0111 + 100$

(3) $11011 - 101 - 11$

(4) $1000 + 101 - 1101$

C) عملية الضرب Multiplication: تتم باعتماد الأسس الأتية:

0 x 0 = 0

0 x 1 = 0

1 x 0 = 0

1 x 1 = 1

مثال (8):

1)

$$\begin{array}{r} 11 \\ \times\ 11 \\ \hline 11 \\ +\ 11 \\ \hline 1001 \end{array}$$

2)

$$\begin{array}{r} 111 \\ \times\ 101 \\ \hline 111 \\ +\ 000 \\ 111 \\ \hline 100011 \end{array}$$

3-4-1 الكسور الثنائية (Binary Fractions):

إذا أردنا تحويل عدد كسري من النظام الثنائي للنظام العشري. بالنسبة للعدد الصحيح يعامل كما مر سابقاً، لكن الأرقام بعد الفارزة فيتم استخدام الإشارة السالبة مع الأس.

مثال (9): سيتم حل هذا المثال بعدد أساليب.

a) $(101.11)_2 = (?)_{10}$

	1	0	1	.	1	1		
…	$(2)_2$	$(1)_2$	$(0)_2$	.	$(-1)_2$	$(-2)_2$	…	الرتبة

$$101.11 = 1\times2^2+0\times2^1+1\times2^1+1\times2^{-1}+1\times2^{-2}$$

$$= 4+0+1 \qquad + 0.5 + 0.25$$

$$= 5 \qquad + 0.75$$

$$\therefore (101.11)_2 = (5.75)_{10}$$

ملاحظة: الخط (تحت الجزء الصحيح من العدد) والخط المزدوج (تحت الجزء الكسري) للتوضيح ولا يوجد ضرورة وضعه عند الحل.

b) $(11001.011)_2$

4	3	2	1	0	-1	-2	-3
1	1	0	0	1	0	1	1

$=1x2^4 +1x2^3 + 0x2^2 + 0x2^1 + 1x2^0 +0x2^{-1} + 1x2^{-2} + 1x2^{-3}$

$= 1x16 + 1x8 + 1x1 + ¼+ 1/8 = 25.625$

$\therefore (11001.011)_2 = (25.625)_{10}$

c) $(1111.101)_2$

$\underline{1111}.\underline{\underline{101}}=\underline{1\times2^3+1\times2^2+1\times2^1+1\times2^0}+\underline{\underline{1\times2^{-1}+0\times2^{-2}+1\times2^{-3}}}$

$= \underline{8+4+2+1} \qquad + \underline{\underline{0.5 +0.0 +0.125}}$

$= \underline{15} \qquad + \underline{\underline{0.625}}$

$\therefore (\underline{1111}.\underline{\underline{101}})_2= (\underline{15}.\underline{\underline{625}})_{10}$

مثال (10): حول الأعداد الكسرية العشرية الأتية إلى ما يقابلها في النظام الثنائي:

0.2 ،8.1875 ،0.25

<u>ملاحظة:</u> في هذه الحالة يتم ضرب الكسر العشري × الأساس الذي يراد التحويل اليه (وفي المثال الأساس 2) مع تسجيل العدد الصحيح الناتج من كل عملية إلى ان تصبح قيمة الكسر العشري **<u>تساوي صفر</u>**.

– كما في الحالة السابقة الجزء الصحيح (X) يقابله العدد الصحيح بعد التحويل (X')، وهكذا بالنسبة للجزء الكسري، أي: $(\underline{X}.\underline{\underline{Y}})_2 =(\underline{X'}.\underline{\underline{Y'}})_{10}$.

الحل:

$$\begin{array}{r} 0.25 \\ \times\quad 2 \\ \hline 0.5 \end{array}$$

1) العدد الصحيح هنا (0)

$$\begin{array}{r} 0.5 \\ \times\quad 2 \\ \hline 1.00 \end{array}$$

⇐ العدد الصحيح 1 والكسر العشري صفر.

$\therefore\ (\underline{0}.\underline{25})_{10} = (\underline{0}.\underline{01})_2$

2) يتم أولاً تحويل الجزء الصحيح من العدد وكما مر سابقاً، أي:

$(8)_{10} = (1000)_2$

$\begin{array}{r} 0.1875 \\ \times\quad 2 \\ \hline 0.3750 \end{array}$ العدد الصحيح هنا 0، $\begin{array}{r} 0.3750 \\ \times\quad 2 \\ \hline 0.750 \end{array}$ ⇐ العدد الصحيح 0

$\begin{array}{r} 0.75 \\ \times\quad 2 \\ \hline 1.5 \end{array}$ ⇐ العدد الصحيح (1)، $\begin{array}{r} 0.5 \\ \times\quad 2 \\ \hline 1.00 \end{array}$ ⇐ العدد الصحيح (1).

(نلاحظ انه عملية الضرب بـ2 فقط للجزء الكسري).

$\therefore\ (\underline{8}.\underline{1875})_{10} = (\underline{1000}.\underline{0011})_2$

3)

$\begin{array}{r} 0.2 \\ \times\quad 2 \\ \hline 0.4 \end{array}$ العدد الصحيح هنا 0 ⇐ $\begin{array}{r} 0.4 \\ \times\quad 2 \\ \hline 0.8 \end{array}$ العدد الصحيح هنا 0

⇐ $\begin{array}{r} 0.8 \\ \times\quad 2 \\ \hline 1.6 \end{array}$ العدد الصحيح هنا (1)، ⇐ $\begin{array}{r} 0.6 \\ \times\quad 2 \\ \hline 1.2 \end{array}$ العدد الصحيح هنا (1)

⇐ $\begin{array}{r} 0.2 \\ \times\quad 2 \\ \hline 0.4 \end{array}$ نلاحظ إن الكسر أصبح دورياً، اي:

$\therefore\ (\underline{0}.\underline{2})_{10} = (\underline{0}.\underline{0011001100110011})_2$

4-4-1 العمليات على الأعداد الثنائية Operations on Binary Numbers:

1- المتممة الأولى One's Complement:

تتم هذه العملية بتغيير كل 0 إلى 1 والعكس على الرقم الثنائي بأكمله.

مثال (11): أوجد المتممة الأولى للأعداد الثنائية الأتية:

a) $(1100101001)_2$

الحل: المتممة الأولى للعدد الثنائي السابق = 0011010110

b) $(10000000000)_2$

الحل: المتممة الأولى للعدد الثنائي السابق = 01111111111

2- المتممة الثانية Two's Complement:

هذه العملية من اهم العمليات التي تتم على الأعداد الثنائية ومن خلالها نستطيع ان نقوم بعملية طرح الأعداد الثنائية وغيرها من العمليات. وتقوم المتممة الثانية بتحويل العدد السالب إلى عدد موجب والعكس وبالتالي نستطيع إجراء عملية الجمع على الأعداد الثنائية إذا قمنا بتحويلها إلى موجبة. ويتم إيجاد المتممة الثانية وذلك بأن توجد المتممة الأولى للعدد الثنائي ثم تجمع على المتممة الأولى الرقم 1.

مثال (12): أوجد المتممة الثانية للأعداد الثنائية الأتية:

a) $(1100101001)_2$

الحل: 1- توجد المتممة الأولى للعدد الثنائي السابق = 0011010110

2- نقوم بجمع الرقم 1 على المتممة الأولى للعدد الثنائي.

$$\begin{array}{r} 0011010110 \\ 1+ \\ \hline 0011010111 \end{array}$$

b) $(11110000000)_2$

الحل: المتممة الأولى = 00001111111

$$\begin{array}{r} 0000111111 \\ 1+ \\ \hline 0001000000 \end{array}$$

3- الجمع والطرح Adding & Subtraction:

تعد عملية الطرح في الأساس عملية جمع عدد موجب مع عدد سالب. في النظام الثنائي لا يمكن إجراء عملية الطرح مباشرة (كما في النظام العشري)، وإنما نقوم بتحويل عملية الطرح إلى جمع باستخدام المتممة الثانية.

مثال (13): جد ناتج عملية الطرح الأتية: 1101 – 0100

$$\begin{array}{r} 1101 \\ 0100- \\ \hline \end{array} \rightarrow \begin{array}{r} 1101 \\ 1100+ \\ \hline \end{array} \rightarrow \begin{array}{r} 1 \\ 1101 \\ 1100+ \\ \hline 11001 \end{array}$$

1101 – 0100 = + 1001

في هذا المثال تم إجراء عملية الطرح على العددين الثنائيين السابقين. العدد الأول 1101 موجب فيبقى كما هو، أما العدد الثاني 0100 سالب لذلك يتم إيجاد المتممة الثانية له = 1100 ثم بعد ذلك تجرى عملية الجمع.

ناتج عملية الجمع = 11001 ولكن يوجد به (Overflow) وذلك لان العدد الأول يمثل في 4 bit والعدد الثاني يمثل في 4 bit والناتج 5 bit ونحن نريد ان يتمثل الناتج في 4 bit لذلك نقوم بحذف أخر عدد في الناتج 1001 ونضع إشارة (+) أمام الناتج.

مثال (14): جد ناتج عملية الطرح الأتية: 0110 – 1100

$$\begin{array}{r} 0110 \\ 1100- \\ \hline \end{array} \rightarrow \begin{array}{r} 0110 \\ 0100+ \\ \hline \end{array} \rightarrow \begin{array}{r} 1 \\ 0110 \\ 0100+ \\ \hline 1010 \\ 0101 \\ 1\ + \\ \hline 0110 \end{array}$$

الناتج في هذا المثال لا يوجد به (Overflow) اي انه يتمثل في 4 bit مثل العددين لذلك نقوم بإيجاد المتممة الثانية للناتج 1010 يساوي 0110 ونضع أمام الناتج إشارة –.
أي بمعنى أخر، إذا وجد في الناتج (Overflow) نحذف أخر عدد من الناتج ونضع إشارة +، إذا لم يوجد في الناتج (Overflow) نأتي بالمتممة الثانية للناتج ونضع (–).

5-1 النظام الثماني (Octal System):

أساس النظام هو (8)، ويتكون من ثمانية أعداد وهي (0-7)، وفيما يأتي جدولاً لقيمة المرتبة وقوتها في هذا النظام ومقارنتها بالنظامين العشري والثنائي:

المرتبة	1	2	3	4	5	6	7	...	n
	8^0	8^1	8^2	8^3	8^4	8^5	8^6	...	
	1	8	64	512	4096	262144	2097152	...	

الأعداد بالنظام		
العشري	الثنائي	الثماني
0	000	0
1	001	1
2	010	2
3	011	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	10
9	1001	11
10	1010	12

1-5-1 تحويل الأعداد من النظام الثماني إلى النظام العشري:

مثال (15):

a) $(120)_8 = (?)_{10}$

$$120 = 1\times8^2 + 2\times8^1 + 0\times8^0$$
$$= 1\times64 + 2\times8 + 0\times1$$
$$= 64 + 16 + 0$$
$$= 80$$

$\therefore (120)_8 = (80)_{10}$

b) $(1000)_8 = (512)_{10}$

$$1000 = 1\times8^3 + 0\times8^2 + 0\times8^1 + 0\times8^0$$
$$= 1\times512 + 0\times64 + 0\times8 + 0\times1$$
$$= 512 + 0 + 0 + 0$$
$$= 512$$

$\therefore (1000)_8 = (512)_{10}$

➢ **تحويل الأعداد من النظام العشري إلى الثماني:**

- نقسم العدد العشري على (8) ونسجل الباقي.
- نستمر بالقسمة على (8) إلى ان يصل الناتج إلى اقل من (8).

مثال (16):

a) $\mathbf{(153.6875)_{10}} = (?)_8$

8	
153	
19	1
2	3
0	2

(231)

$0.6875 \times 8 = 5.5 \rightarrow 5$

$0.5 \times 8 = 4 \rightarrow 4$

(0.54)

$(153.6875)_{10} = (231.54)_8$

عند قسمة العدد (153) على 8 ينتج من عملية القسمة العدد (19.125) نأخذ الجزء الكسري من الناتج (0.125) ونضربه في أساس النظام المحول اليه 8 ينتج من عملية

الضرب 1 ويكون هذا العدد باقي أول عملية قسمة ثم نأخذ الجزء الصحيح وهو العدد (19) ونكرر معه الخطوات السابقة وتنتهي عملية التحويل عندما يكون ناتج القسمة هو 0 والباقي يساوي عدد صحيح.

b) $(1967)_{10} = =(?)_8$

8	1967	
8	245	7
8	30	5
8	3	6
	0	3

$(1967)_{10} = (3657)_8$

وللتأكد من ذلك:

$$3657 = 3\times8^3 + 6\times8^2 + 5\times8^1 + 7\times8^0$$
$$= 3\times512+6\times64 + 5\times8 + 7\times1$$
$$= 1536 + 384 + 40 + 7 = 1967$$

2-5-1 تحويل النظام الثنائي إلى الثماني:

– نجزئ العدد الثنائي (من جهة اليمين) إلى مجاميع كل مجموعة تحوي على ثلاثة أرقام، وكتابة ما يقابلها بالنظام الثماني.

مثال (17): حول $(1101011100011111)_2$ إلى النظام الثماني:

1101011100011111										
			001	101	011	100	011	111	⇐	ثنائي
			1	5	3	4	3	7		ثماني

$\therefore (1101011100011111)_2 = (153437)_8$

ملاحظة: نلاحظ تم أضافة 00 في الخلية الأخيرة فقط لجعل أرقام الخلية ثلاثة، وليس لها أي تأثير على الحل.

➢ تحويل النظام الثماني إلى النظام الثنائي:

– يتم عكس العملية السابقة، إذ نجزئ العدد الثماني (من جهة اليمين) إلى مجاميع كل مجموعة تحتوي على رقم واحد، وكتابة ما يقابلها بالنظام الثنائي وعلى هيئة ثلاث أرقام.

مثال (18): حول العدد $(14732)_8$ إلى النظام الثنائي:

14732								
ثماني			2	3	7	4	1		
ثنائي	⇐		010	011	111	100	001		

$\therefore (14732)_8 = (1100111011010)_2$

6-1 النظام السادس عشر (Hexadecimal System):

بعد الرقم 9 يأتي الرقم 10 هو رقم مركب من 0 و 1 وهذا تكرار، لذا استخدمت الحروف فقط في النظام السادس لأن الأرقام لا تتكرر في الأنظمة الأخرى، إذ في النظام الثنائي 0 و 1 الثماني 0، 1...7 والعشري 0، 1،...، 9 أما السادس عشر فلا يوجد أكثر من عشرة رموز مستقلة ومختلفة هي 0 - 9 فجاءت الحاجة إلى الأحرف الستة الباقية (A-F) للوصول إلى 16 رمز مختلف وتوسيع قاعدة العد. والأرقام الأحرف هي:

0 1 2 3 4 5 6 7 8 9 A B C D E F

تكتب الأحرف Capital فقط وتقابل:

A = 10, B = 11, C = 12, D = 13, E = 14, F = 15

المرتبة	1	2	3	4	5	6	7	...	n
	16^0	16^1	16^2	16^3	16^4	16^5	16^6	...	
	1	16	256	4096	65536	1048576	16777216		

الاعداد بالنظام			
العشري	الثنائي	الثماني	السادس عشر
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

1-6-1 تحويل النظام الثنائي إلى النظام السادس عشر وبالعكس:

مثال (19): لتحويل $(1101100111110101)_2$ إلى ما يقابله في النظام السادس عشر نقوم بتجزئته إلى مجاميع تحتوي كل منها **أربعة** مراتب، كما يأتي:

1101100111110101					
1101	1001	1111	0101	$\Leftarrow$	ثنائي
D	9	F	5	السادس عشري	

$\therefore (1101100111110101)_2 = (D9F5)_{16}$

ونفس الملاحظة السابقة إذا لم تكتمل الخلية الأخيرة بأربعة رموز نكملها بالأصفار.

وبالعكس، لتحويل $(3DA7)_{16}$ إلى النظام الثنائي:

3DA7					
3	D	A	7	السادس عشري	
0011	1101	1010	0111	$\Leftarrow$	ثنائي

و(00) الأخيرة لا يتم وضعها بالمقدار في الجواب الأخير وكالاتي:

$\therefore (3DA7)_{16} = (11110110100111)_2$

2-6-1 تحويل النظام السادس عشر إلى النظام الثماني وبالعكس:

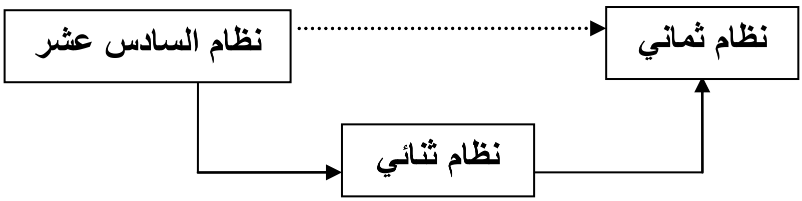
مثال (20): لتحويل $(5B)_{16}$ إلى ما يقابله في النظام الثماني.

– يحول أولاً إلى النظام الثنائي من ثم يحول إلى ما يقابله إلى النظام الثماني، كما في الخطوات الأتية:

5B				
	5	B		1- السادس عشري
	101	1011		2- ثنائي
1011011				الناتج
1	011	011		3- ثنائي
1	3	3		4- ثماني

$\therefore (5B)_{16} = (133)_8$

أي نلاحظ ان عملية التحويل: $(133)_8 \Rightarrow (1011011)_2 \Rightarrow (5B)_{16}$ أو بالعكس.



– بنفس الطريقة يحول النظام الثماني إلى النظام السادس عشر، أي:

$$(?)_8 \Rightarrow (?)_2 \Rightarrow (?)_{16}$$

<u>3-6-1 تحويل النظام السادس عشر إلى النظام العشري وبالعكس:</u>

مثال (21): $(A2D)_{16} \Rightarrow (?)_{10}$

$$\begin{aligned} A2D &= 10\times16^2 + 2\times16^1 + 13\times16^0 \\ &= 2560 + 32 + 13 \\ &= 2605 \end{aligned}$$

نلاحظ ان قيمة $A=10$ و$D=13$ عند تحويلها إلى نظام أخر، وهكذا مع بقية الأحرف.

$\therefore (A2D)_{16} = (2605)_{10}$

وللتحويل من **النظام العشري** إلى ما يقابله في **النظام السادس عشر** نتبع نفس الخطوات في حالة النظامين الثنائي والثماني لكن نقسم على (16).

مثال (22): حول العدد $(90)_{10}$ إلى نظيره بالنظام السادس عشر.

الحل:

القسمة $\frac{90}{16}$ تعطي 5 والباقي A يمين

القسمة $\frac{5}{16}$ تعطي 0 والباقي 5 يسار

$\therefore (90)_{10} = (5A)_{16}$

ملاحظة: يمكن ان نحول من النظام السادس عشر إلى النظام الثنائي من ثم إلى النظام العشري. والعكس صحيح (بالضبط مثل الطريقة المتبعة في تحويل النظام السادس عشر إلى النظام الثماني).

س: ما أهمية النظام الست عشري؟

يسهل التعامل مع الأعداد الكبيرة التي يصعب تمثيلها بالنظام الثنائي من قبل مستخدمي الحاسوب والمبرمجين.

مثال (23): تحويل من نظام (ثنائي، ثماني وسادس عشري) إلى نظام عشري:

1. $(11010.101)_2 = 1(2)^4+1(2)^3+ 0(2)^2+ 1(2)^1+ 0(2)^0+ 1(2)^{-1} + 0(2)^{-2}+ 1(2)^{-3} = 16+8+2+0.5+0.125 = (26.625)_{10}$

2. $(613.24)_8 = 6(8)^2 +1(8)^1+3(8)^0+2(8)^{-1} + 4(8)^{-2} = 384 + 8 + 3 + 0.25 + 0.0625 = (395.3125)_{10}$

3. $(5A.E)_{16} =5(16)^1+ A(16)^0+ E(16)^{-1} = 5(16)+10(16)+ 14(16)^{-1} = 80+10+0.875 = (90.875)_{10}$

مثال (24): تحويل من نظام عشري إلى نظام ثنائي:

1. $(353)_{10} = (?)_2$

		الباقي	
353	2	1	← **LSB** يمين
176	2	0	
88	2	0	
44	2	0	
22	2	0	
11	2	1	
5	2	1	
2	2	0	
1	2	1	← **MSB** يسار
0			

$(353)_{10} = (101100001)_2$

2. $(0.65625)_{10} = (?)_2$

$$
\begin{array}{lr}
 & 0.65625 \\
 & \underline{\quad\quad 2\ x} \\
\boxed{1} \leftarrow & 1.31250
\end{array}
$$

أول مرتبه بعد الفارزة.

$$
\begin{array}{lr}
 & 0.31250 \\
 & \underline{\quad\quad 2\ x} \\
\boxed{0} \leftarrow & 0.62500 \\
 & \underline{\quad\quad 2\ x} \\
\boxed{1} \leftarrow & 1.250 \\
 & \underline{\quad\quad 2\ x} \\
\boxed{0} \leftarrow & 0.50 \\
 & \underline{\quad\quad 2\ x} \\
\boxed{1} \leftarrow & 1.0
\end{array}
$$

$(0.65625)_{10} = (0.10101)_2$

3. $(254.75)_{10} = (?)_8$

		الباقي	
254	8	6 →	**LSB**
31	8	7	
3	8	3	
0			

$$
\begin{array}{lr}
 & 0.75 \\
 & \underline{\quad 8\ x} \\
\boxed{6} \leftarrow & 6.00
\end{array}
$$

$(254.75)_{10} = (376.6)_8$

4. $(567.1875)_{10} = (?)_{16}$

567	16	7 ←	**LSB**
35	16	3	
2	16	2	
0			

0.1875

$$\begin{array}{r} \underline{\quad 16\ x} \\ \boxed{3} \longleftarrow 3.0000 \end{array}$$

$(567.1875)_{10} = (237.3)_{16}$

مثال (25): التحويل بين النظام الثنائي، والثمائي والسادس عشري (طريقة سريعة):

1.

1 2 7 5 4 3	Octal
1 0 1 0 1 1 1 1 0 1 1 0 0 0 1 1	Binary
A F 6 3	Hexadecimal

$= (44899)_{decimal}$

2.

2 1 4 4 3 5	Octal
1 0 0 0 1 1 0 0 1 0 0 0 1 1 1 0 1	Binary
1 1 9 1 D	Hexadecimal

مثال (26): الجمع في النظام السادس عشر Hexadecimal Addition

$$\begin{array}{r} 23 \\ \underline{+16} \\ 39_H \end{array} \qquad \begin{array}{r} 58 \\ \underline{+22} \\ 7A_H \end{array} \qquad \begin{array}{r} 2B \\ \underline{+84} \\ AF_H \end{array}$$

مثال (27):

$$\begin{array}{r} {}^{1}\ \ \\ D\ F \\ \underline{+A\ C} \\ 18\ B \end{array}$$

$F \rightarrow 15_D + 12_D = 27_D \rightarrow 27_D - 16_D = 11_D = B_H$ with 1 carry

$\rightarrow 1 + 13_D + 10_D = 24_D \rightarrow 24_D - 16_D = 8_D = 8_H$ with 1 carry

مثال (28): الجمع في النظام الثماني Octal Addition.

1-
$$\begin{array}{r} 14 \\ \underline{+23} \\ 37_8 \end{array}$$

2-
$$\begin{array}{r} 3\ 7 \\ \underline{5\ 3} \\ 11\ 2 \end{array}$$

$7_O + 3_O = 10_D - 8_D = 2_D = 2_O \quad + 1 \quad \text{carry}$

$1 + 3_O + 5_O = 9_D - 8_D = 1_D = 1_O + 1 \text{ carry}$

مثال (29): المتممة في النظام الثماني: 7777_8-2415_8

ملاحظة:

7´s = 7 – each digit
8´s = 7´s + 1

الحل:

$$\begin{array}{r} 7777 \\ \underline{-\ 2415} \\ 5362 \\ \underline{+1} \\ 5363 \end{array}$$

5362 متممة الـ7

5363 متممة الـ8

مثال (30): جد ناتج 7526_8 -3142_8 باستخدام متممة النظام الثماني.

$$\begin{array}{r} 7777 \\ \underline{-3142} \end{array}$$

4635 + 1=4636

$$\begin{array}{r} {}^{1\ \ 1} \\ 7526 \\ \underline{+\ 4636} \end{array}$$

12 – 8 = 4

11 - 8 = 3

12 - 8 = 4

= 14364

4-6-1 المتممة Complements في النظام السادس عشر

مثال (31): جد متممة 15 و16 لـ$(1FAD)_{16}$.

15	15	15	15	
- 1	F	A	D	
E	0	5	2	← 15´s comp.
			1 +	
E	0	5	3	← 16´s comp.

مثال (32): جد الناتج باستخدام متممة النظام السادس عشر.

a) ABED – 1FAD

```
     1 1
    A B E D
  + E 0 5 3
  1 8 C 4 0      16 – 16 = 0
                 20 – 16 = 4
                 24 – 16 = 8
```

الجواب: 1 8C40

b) $FEED_{16} – DAF3_{16} = ?$

الجواب: $23FA_{16}$

أمثلة متنوعة (33):

1- $(1011)_2 = 1x\ 2^0 + 1x\ 2^1 + 0x\ 2^2 + 1x\ 2^3$
$= 1 + 2 + 0 + 8$
$= (11)_{10}$

2- $(110.1)_2 = 0x\ 2^0 + 1x\ 2^1 + 1x\ 2^2 + 1x\ 2^{-1}$
$= 0 + 2 + 4 + 0.5$
$= (6.5)_{10}$

3- $(1100.101)_2 = 0\times2^0 + 0x2^1 + 1x2^2 + 1x2^3 + 1x2^{-1} + 0x2^{-2} + 1x2^{-3}$
$= 0 + 0 + 4 + 8 + 0.5 + 0 + 0.125$
$= (12.625)_{10}$

4- $(752)_8 = 7x\ 8^0 + 5x\ 8^1 + 2x\ 8^2$
$= 7 + 40 + 448$
$= (490)_{10}$

5- $(ABC)_{16} = 12x\ 16^0 + 11x\ 16^1 + 10x\ 16^2$
$= 12 + 176 + 2560$
$= (2748)_{10}$

6- $(2F.8)_{16} = 15x\ 16^{0} + 2x\ 16^{1} + 8x\ 16^{-1}$
$= 15 + 32 + 0.5$
$= (47.5)_{10}$

7- $(62.7)_8 = (110010.111)_2$

8- $(35.41)_8 = (011101.100001)_2$

9- $(00101110.1010)_2 = (2E.A)_{16}$

10- $(1111100.01011011)_2 = (FC.5B)_{16}$

11 - $(AB.6D)_{16} = (10101011.01101101)_2$

12- $(9C.8F3)_{16} = (10011100.100011110011)_2$

مثال (34): حول العدد العشري $(125.34375)_{10}$ إلى النظام الست عشري.

16	
125	
7	13
0	7
(7D)	

$0.34375 \times 16 = 5.5 \rightarrow 5$
$0.5 \quad \times 16 = 8 \rightarrow 8$
(0.58)

$\therefore \ (125.34375)_{10} = (7D.58)_{16}$

مثال (35): حول العدد الثنائي إلى النظام الثماني:

1) $(10011101110)_2$

010	011	101	110
2	3	5	6

$\therefore (10011101110)_2 = (2356)_8$

2) $(0101111)_2$

010	111	100
2	3	4

$\therefore (0101111)_2 = (234)_8$

3) $(11001.01)_2$

011	001	010
3	1	2

$\therefore (11001.01)_2 = (31.2)$

تمارين الفصل الأول

1) حول الاتي:

a) $(43)_{10} \rightarrow (\ \ )_2$

b) $(0.375)_{10} \rightarrow (\ \ )_2$

c) $(2048.0625)_{10} \rightarrow (\ \ )_2$

d) $(0.0011011)_2 \rightarrow (\ \ )_{10}$

e) $(75102.72)_8 \rightarrow (\ \ )_{10}$

f) $(1101010.011)_{10} \rightarrow (\ \ )_{16}$

g) $(5FF3.A1)_{16} \rightarrow (\ \ )_8$

2) باستخدام متممة النظام الثماني، جد:

$(545)_8 - (14)_8 = ?$

$(6776)_8 - (4337)_8 = ?$

3) باستخدام متممة النظام السادس عشر، جد:

$(98A\ E)_{16} - (1FEE)_{16} = ?$

4) جد نواتج:

a) $(7152)_8 - (1010)_2 = (\ \ )_{10}$

b) $(1039)_{10} - (3E4B)_{16} = (\ \ )_8$

c) $(6022)_{10} - (4352)_{10} = (\ \ )_2$

الفصل الثاني

الجبر البولييني

Boolean Algebra

الفصل الثاني

الجبر البولييني Boolean Algebra

2–1 مقدمة:

سمي **الجبر البولييني** (Boolean Algebra)، نسبة إلى الإنجليزي جورج بوليين (1815–1864م) وهو عالم منطق ورياضيات، وهي طريقة رياضيّة تُستعمل لحلِّ مسائل المنطق والاحتمالات الهندسية، ويعتبر الجبر البولييني أحد المرتكزات الأساسية المستخدمة في تصميم وتركيب الحاسوب. ويعود الفضل في وضع الأسس النظرية للجبر البولييني، والذي يسمى أيضاً بالجبر المنطقي للعالم بوليين. وقد نشر هذا العالم نظرياته في منتصف القرن التاسع عشر لتصبح فيما بعد الأساس في تصميم الدوائر المنطقية التي يتكون منها الحاسوب. وقام بوليين بنشر كتابة "استقراء قوانين التفكير" في 1854 الذي وضع فيه وفي أعماله اللاحقة أسس الجبر المنطقي الذي يعد لبنة هامة في تصميم العمليات المنطقية للحاسوب الحديث.

وقد طوّر بول طريقة لتكوين العبارات المنطقيّة بالرموز. ويمكن كتابة هذه العبارات وإثباتها بطريقة مماثلة للطريقة المستعملة في الجبر العادي. ويستعمل الجبر المنطقي أيضاً في المسائل الهندسية مثل تصميم دوائر المفاتيح الكهربائية، وبخاصة الدوائر التي تؤدي عمليات حسابية في الآلات الحاسبة والحواسيب. ويتناول الجبر البوليني العلاقات بين المجموعات (مجموعات الأفكار أو الأشياء). مثال مجموعات الأرقام الأقل من مائة؛ فاكهة حمراء اللون؛ ... . وفي الجبر البوليني يتم التمثيل لهذه المجموعات بالحروف a, b c وهكذا. وتتبع ثلاث من العمليات البولينية قوانين مشابهة لقوانين الجبر العادية. ورموز هذه العمليات هي $\cap$,$\cup$ (تقاطع أو اتحاد). فمثلاً العملية $a\cup b$ تمثل مجموعة العناصر الموجودة في كلتا المجموعتين a وb.

2-2 العمليات البولينية الأساسية:

1. عملية "**و**" (**AND** Operation).
2. عملية "**أو**" (**OR** Operation).
3. عملية "**لا**" (**NOT** Operation).

تسمى العمليتان الأولى والثانية **عمليتان ثنائيتان (Binary Operations)** لأن كلاً منها تحتاج إلى متغيرين على الأقل، بينما تسمى عملية "لا" NOT **عملية أحادية** لأن لها متغيراً واحداً أو مدخلاً واحداً فقط، ويمكن استخدام الإشارات الجبرية التالية لتمثيل العمليات الأساسية. مع الافتراض أن المتغيرات هي X ،Y.

هذا ويمكن وصف العمليتين "و"، "أو" بأكثر من متغيرين ولكننا في معظم الحالات سنتكلم عنهما مستخدمين فقط متغيرين للتسهيل ليس إلا. وبالتعبير عن هذه العمليات بالنظام الثنائي باعتبار أن الرقم "1" يمثل الحالة الصحيحة والرقم "0" يمثل الحالة الخاطئة فيمكن تعريف هذه العمليات كما يأتي:

عملية AND
$$X \cdot Y = \begin{cases} 1 & \text{إذا كان كل من X و Y يساوي 1} \\ 0 & \text{إذا كان كل من X و / أو Y يساوي 0} \end{cases}$$

عملية OR
$$X + Y = \begin{cases} 0 & \text{إذا كان كل من X و Y يساوي 1} \\ 1 & \text{إذا كان كل من X و / أو Y يساوي 0} \end{cases}$$

عملية NOT
$$\overline{X} = \begin{cases} 1 & \text{إذا كان X=0} \\ 0 & \text{إذا كان X=1} \end{cases}$$

تستخدم عادة جداول لوصف العمليات المنطقية تسمى **جداول الحقيقة أو الصواب Truth Tables**، إذ تحتوي على كل الحالات التي تقع فيها المتغيرات وعلى ناتج العملية لكل حالة. من السهل ملاحظة أنه إذا كان عدد المتغيرات يساوي n فإن عدد الحالات الممكنة هي 2^n. وجدول العملية "و" ذات المتغيرين موضحة في الجدول (1−2).

الجدول (2−1) جدول الحقيقة لعملية "و" AND

X	Y	F = X.Y
0	0	0
0	1	0
1	0	0
1	1	1

أي أن تكون في الحالة الصحيحة فقط إذا كانت جميع المتغيرات في الحالة الصحيحة.

الجدول (2−2) يبين عملية "أو" ذات المتغيرين.

الجدول (2−2) جدول الحقيقة لعملية "أو" OR

X	Y	F = X+Y
0	0	0
0	1	1
1	0	1
1	1	1

أي أن عملية "أو" OR تكون في الحالة الصحيحة إذا كان أي من متغيراتها في الحالة الصحيحة وتكون في الحالة الخاطئة إذا كانت كل متغيراتها في الحالة الخاطئة. الجدول (2−3) يبين عملية "لا" NOT.

الجدول (2−3) جدول الحقيقة لعملية "لا" NOT

X	$F = \overline{X}$
0	1
1	0

2-3 البوابات المنطقية (Logic Gates):

هي دائرة إلكترونية بسيطة تقوم بعملية منطقية على مدخل واحد أو أكثر وتنتج مخرجاً منطقياً واحداً، وتستخدم في بناء معالجات الأجهزة الالكترونية والحواسيب. لأن مخرج البوابة الرقمية هو أيضاً قيمة منطقية، فإنه يمكن استخدام مخرج أحد البوابات المنطقية كمدخل لبوابة أخرى. المنطق المستخدم غالباً هو المنطق البوليني (Boolean logic)، وهو المنطق الذي يعمل في الدوائر الرقمية. يتم صناعة الدائرة الالكترونية للبوابة الرقمية

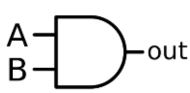
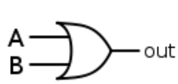
باستخدام دايودات وترانزستور، ولكن يمكن أيضاً بناؤها من مبدّلات إلكترونية، إشارات ضوئية، جزيئات، وحتى من أجزاء ميكانيكية.

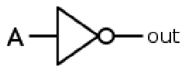
الدوائر المنطقية عبارة عن هياكل مصممة من عدد من الدوائر الأولية تسمى **بوابات منطقية**، وكل واحد من هذه الدوائر المنطقية يمكن النظر إليها كماكينة L تحتوي على جهاز أو أكثر للإدخال وجهاز إخراج واحد فقط، وفي أي لحظة يستطيع كل جهاز إدخال في L استيعاب وحدة أساسية واحدة من المعلومات هي 0 أو 1 ثم تعالج هذه البيانات بالدائرة لإعطاء ناتج هو 0 أو 1 على جهاز الإخراج، وبالتالي يمكن تخصيص سلسلة من الأصفار 0 أو الوحدات 1، إذ تعالج L وحدة أساسية في كل مرة.

4-2 أنواع البوابات المنطقية:

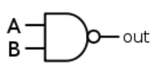
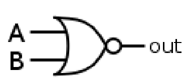
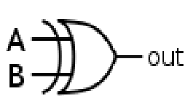
الجدول (2-4) يبين أنواع البوابات المنطقية والرموز القياسية المعبرة عنها:

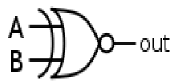
الجدول (2-4)

<table>
<tr><th>نوع</th><th>شكل مميز</th><th>الجبر البوليني بين B, A</th><th>جدول</th></tr>
<tr><td>AND</td><td>
</td><td>A.B</td><td>
<table>
<tr><th colspan="2">مدخل</th><th>مخرج</th></tr>
<tr><td>A</td><td>B</td><td>A AND B</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td></tr>
<tr><td>OR</td><td>
</td><td>A+B</td><td>
<table>
<tr><th colspan="2">مدخل</th><th>مخرج</th></tr>
<tr><td>A</td><td>B</td><td>A OR B</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td></td><td></td><td><table><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>NOT</td><td>
</td><td>$\overline{A}$</td><td><table><tr><td>مدخل</td><td>مخرج</td></tr><tr><td>A</td><td>NOT A</td></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table></td></tr>
</table>

في الإلكترونيات، غالباً ما تسمى بوابة NOT بالعاكس (Inverter). الدائرة المرسومة أمام البوابة تدعى الفقاعة (Bubble). وترسم الفقاعة أحياناً أمام أي دائرة منطقية لبيان أنّها معكوسة.

<table>
<tr><td>NAND</td><td>
</td><td>$\overline{A.B}$</td><td><table><tr><td>INPUT</td><td></td><td>OUTPUT</td></tr><tr><td>A</td><td>B</td><td>A NAND B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>NOR</td><td>
</td><td>$\overline{A+B}$</td><td><table><tr><td>INPUT</td><td></td><td>OUTPUT</td></tr><tr><td>A</td><td>B</td><td>A NOR B</td></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
<tr><td>XOR</td><td>
</td><td>$A \oplus B$</td><td><table><tr><td>INPUT</td><td></td><td>OUTPUT</td></tr><tr><td>A</td><td>B</td><td>A XOR B</td></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table></td></tr>
</table>

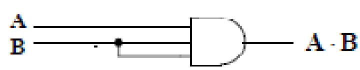
<table>
<tr>
<td>XNOR</td>
<td>
</td>
<td>$\overline{A\oplus B}$</td>
<td>
<table>
<tr><th colspan="2">INPUT</th><th>OUTPUT</th></tr>
<tr><td>A</td><td>B</td><td>A XNOR B</td></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

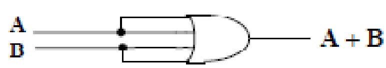
2-4-1 تغير عدد اطراف الدخل (Fan-In) للبوابة المنطقية:

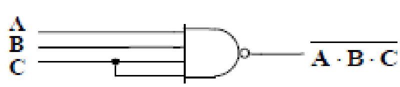
في كثير من الاحيان قد يتوفر بوابات منطقية بعدد من اطراف الدخل (Fan-In) اكبر أو اقل مما يحتاج اليه. وفي هذه الفقرة سنبين الاساليب المختلفة التي يمكن اتباعها لتغيير عدد اطراف الدخل للبوابة المنطقية بالزيادة أو بالنقصان.

- **تقليل عدد اطراف الدخل:**

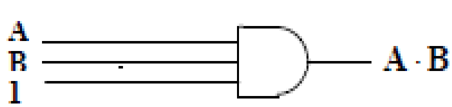
يتم ذلك بربط طرف الدخل الزائد بأحد اطراف الدخل المستخدمة، مثلاً:

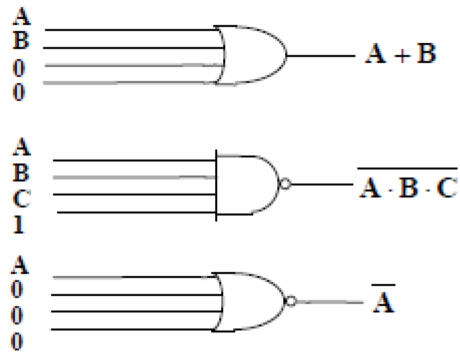






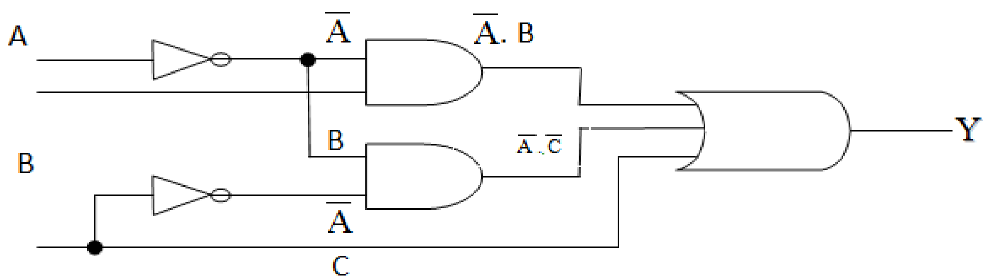
في الحالة الأولى أُستخدم بوابة AND بثلاثة مداخل كبوابة AND بمدخلين، وذلك بالتخلص من طرف الدخل الثالث غير المرغوب فيه بربطه باحد اطراف الدخل. وفي الحالة الثانية استخدم بوابة OR بأربعة مداخل كبوابة OR بمدخلين. كما يمكن التخلص من طرف الدخل الزائد بوضع القيمة المنطقية 1 في طرف الدخل الزائد في بوابات AND وNAND، ووضع القيمة المنطقية 0 في طرف الدخل الزائد في بوابات OR وNOR، مثلاً:





2-4-2 الدوائر المنطقية المركبة البسيطة:

مثال (1): للدائرة الموضحة في الشكل الاتي، أوجد التعبير البوليني وجدول الحقيقة لها:



الخرج عند:

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

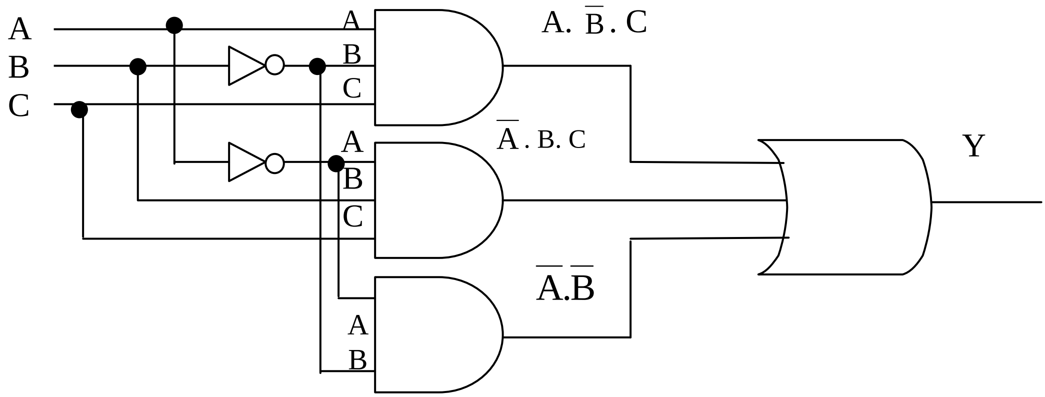
وجدول الحقيقة:

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

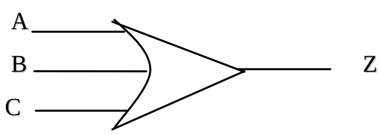
مثال (2): ارسم الدائرة التي تنفذ الصيغة البولينية الأتية:

$$Y = (A.\bar{B}.C) + (\bar{A}.B.C) + (\bar{A}.\bar{B})$$

الحل:

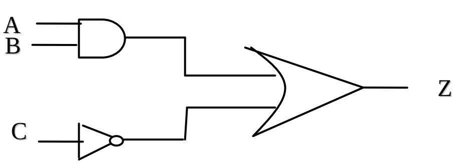


مثال (3): صف جدول الحقيقة لبوابة OR ذات ثلاثة مدخلات:



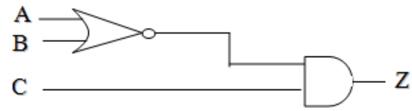
ABC	Z
000	0
001	1
010	1
011	1
100	1
101	1
110	1
111	1

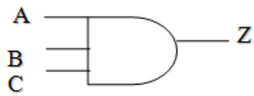
مثال (4): صف جدول الحقيقة للدائرة المنطقية الأتية:

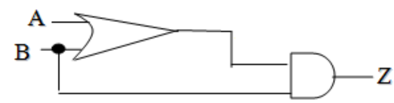


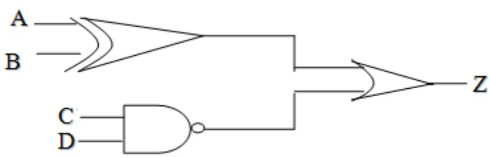
ABC	Z
000	1
001	0
010	1
011	0
100	1
101	0
110	1
111	1

تمرين (1): صف جدول الحقيقة للدوائر المنطقية الأتية:

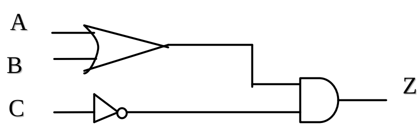






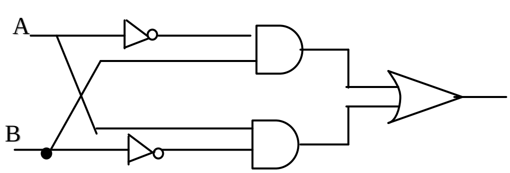


مثال (5): اكتب الصيغة البولينية للدائرة المنطقية الأتية:



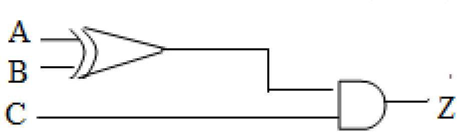
$$Z = (A + B).\overline{C}$$

مثال (6): اكتب الصيغة البوليينية للدائرة المنطقية الأتية:



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

مثال (7): اكتب الصيغة البوليينية للدائرة المنطقية الأتية:

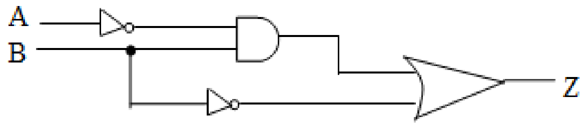


$$Z = (A \oplus B).C$$

مثال (8): كون دائرة منطقية للصيغة البوليينية الأتية:

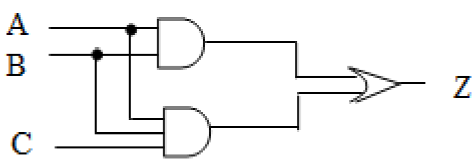
$$Z = \overline{A}.B + \overline{B}$$

نحتاج إلى بوابة AND عدد 2، وبوابة OR عدد 2، و2 عاكس (2-inverters).



مثال (9): كون الدائرة المنطقية واكتب جدول الحقيقة للصيغة الأتية:

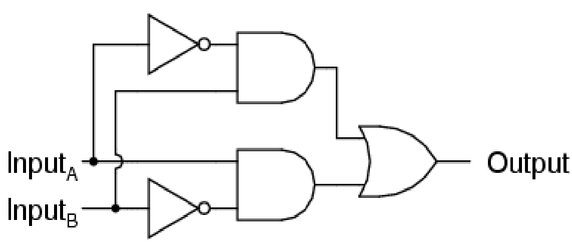
$Z = AB+ABC$



ABC	Z
000	0
001	0
010	0
011	0
100	0
101	1
110	0
111	1

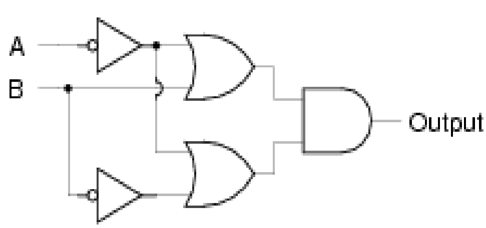
مثال (10): اكتب جدول الحقيقة للدوائر المنطقية الاتية:

a)



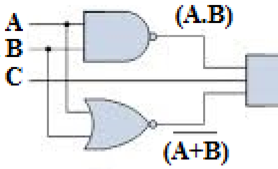
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	0

b)



A	B	Output
0	0	1
0	1	1
1	0	0
1	1	0

c)

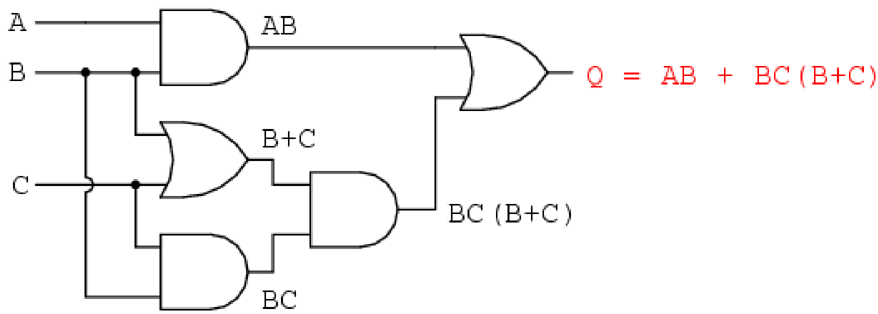


$Q=\overline{(A.B)}\ \overline{(A+B)}.C$

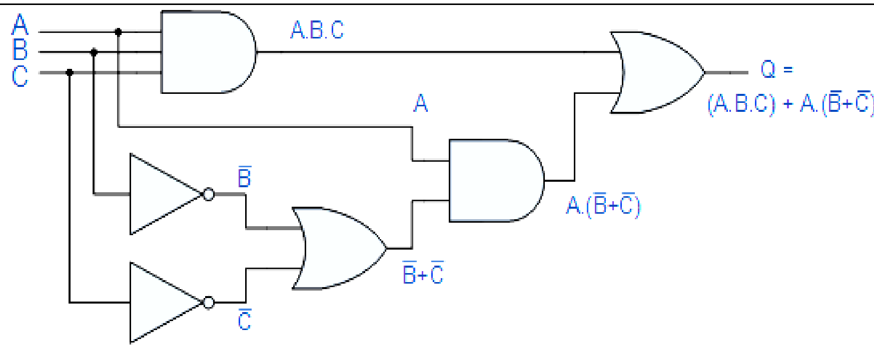
C	B	A	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
			0

مثال (11): اكتب الصيغة المنطقية للدوائر المنطقية الاتية:

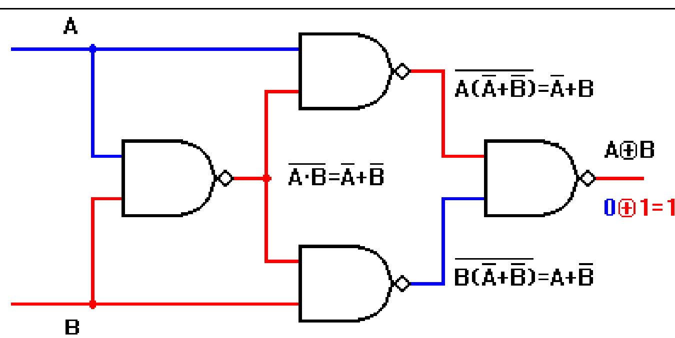
a)

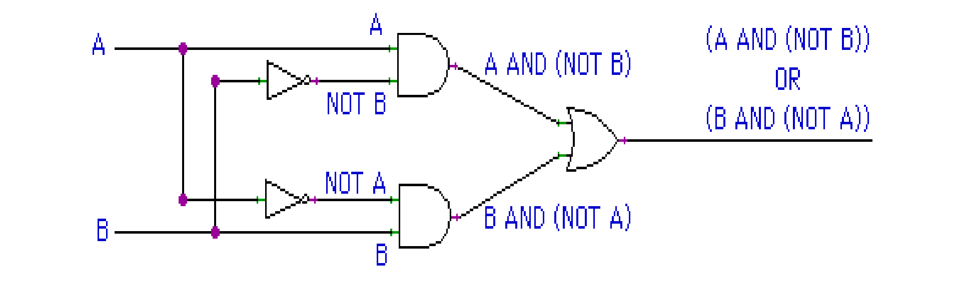


b)



c)



d)	

5-2 **قوانين الجبر البوليني:**

اشتقت من العمليات الأساسية الثلاثة مجموعة قوانين هامة جداً في عمل الدوائر المنطقية، وفيما يأتي ملخص لهذه القوانين:

(1) قانون الانفراد (Uniqueness) للمتغير البوليني:

– إذا كانت $X=1$ فان $X\neq 0$

– إذا كانت $X=0$ فان $X\neq 1$

(2) قانون عمليات "الصفر":

$X + 0 = X$

$X . 0 = 0$

– **إثبات القانون:**

بما أن X متغير ثنائي فإن له حالتين إما الصفر أو الواحد.

ففي حالة كون $X=0$ فأن:

0 = 0 OR 0

0 = 0 AND 0

وفي حالة $X=1$ فأن:

1= 0 OR 1

1 = 1 AND 1

يبين الجدول (2–5) أثبات قانون (2).

الجدول (2-5)

X	X+0	X.1
0	0	0
1	1	1

(3) قانون عمليات "الواحد":

- $X + 1 = 1$
- $X \, . \, 1 = X$

(4) قانون عمليات النفي (Complementation):

- $x + \overline{x} = 1$
- $x . \overline{x} = 0$

جدول (2-6) يوضح إثبات هذا القانون.

الجدول (2-6)

X	$\overline{X}$	$X + \overline{X}$	$X.\overline{X}$
0	**1**	**1**	**0**
1	**0**	**1**	**0**

(5) قانون النفي المزدوج (Double Negation)

$\overline{\overline{X}} = X$

(6) قانون التشابه (Similarity):

- $X + X = X$
- $X \, . \, X = X$

(7) قانون الاختزال (Absorption Law):

- $X + XY = X$
- $X \, (X + Y) = X$
- $X + \overline{X} Y = X + Y$
- $X \, .(\overline{X} + Y) = XY$

الجدول (2-7) يوضح إثبات هذا القانون بشقيه.

الجدول (2−7) جدول الحقيقة لقانون الاختزال

X	Y	X.Y	X+XY	X+Y	X. (X+Y)
0	0	0	0	0	0
0	1	0	0	1	0
1	0	0	1	1	1
1	1	1	1	1	1

(8) قانون التبديل (Commutative Law):

- $X + Y = Y + X$
- $X . Y = Y . X$

(9) قانون الاقتران (Associative Law):

- $X + Y + Z = X + (Y + Z) = (X + Y) + Z$
- $X . Y . Z = X . (Y . Z) = (X . Y) . Z$

(10) قانون التوزيع (Distributive Law):

- $X (Y + Z) = XY + XZ$
- $X + YZ = (X + Y) (X + Z)$

الجدول (2−8) يوضح إثبات القوانين السابقة:

الجدول (2−8)

X	Y	$\overline{X}$	$\overline{X}$ Y	X+ $\overline{X}$ Y	X+Y	XY	$\overline{X}$ +Y	X($\overline{X}$ +Y)
0	0	1	0	0	0	0	1	0
0	1	1	1	1	1	0	1	0
1	0	0	0	1	1	0	0	0
1	1	1	0	1	1	1	1	1

2−5−1 قانونا دي مورجان (De Morgan Laws)

ساهم دي مورجان (عالم رياضيات ومنطق) فضلاً عن بول في وضع القوانين المنطقية وخاصة القانونين الآتيين:

$$1)\ \overline{(X_1 + X_2 + X_3 \ldots + X_n)} = \overline{X_1}.\overline{X_2}.\overline{X_3}\ldots\overline{X_n}$$

أي أن مكمل المجموع (لمتغيرات منطقية) يساوي حاصل ضرب مكملات المتغيرات.

$$2)\ \overline{(X_1.X_2.X_3 \ldots X_n)} = \overline{X_1} + \overline{X_2} + \overline{X_3} \ldots + \overline{X_n}$$

أي أن مكمل حاصل الضرب يساوي مجموع مكملات المتغيرات (المقصود المجموع المنطقي وحاصل الضرب المنطقي).

الجدول (2-9) يثبت قانون دي مورجان الأول لثلاث متغيرات $\overline{X+Y+Z}=\overline{X}.\overline{Y}.\overline{Z}$.

الجدول (2-9)

X	Y	Z	X+Y+Z	$\overline{X+Y+Z}$	$\overline{X}$	$\overline{Y}$	$\overline{Z}$	$\overline{X}.\overline{Y}.\overline{Z}$
0	0	0	0	1	1	1	1	1
0	0	1	1	0	1	1	0	0
0	1	0	1	0	1	0	1	0
0	1	1	1	0	1	0	0	0
1	0	0	1	0	0	1	1	0
1	0	1	1	0	0	1	0	0
1	1	0	1	0	0	0	1	0
1	1	1	1	0	0	0	0	0

الجدول (2-10) يثبت قانون دي مورجان الثاني لثلاث متغيرات: $\overline{X.Y.Z}=\overline{X}+\overline{Y}+\overline{Z}$.

الجدول (2-10)

X	Y	Z	X.Y.Z	X.Y.Z	X	Y	Z	X + Y + Z
0	0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0	1
0	1	0	0	1	1	0	1	1
0	1	1	0	1	1	0	0	1
1	0	0	0	1	0	1	1	1
1	0	1	0	1	0	1	0	1
1	1	0	0	1	0	0	1	1
1	1	1	1	0	0	0	0	0

تستخدم هذه القوانين لتبسيط التعابير البوليينية للحصول على أبسط صيغة ممكنه حتى يتم بناؤها كدوائر الكترونية بأقل تكلفة.

مثال (12): اثبت ان: $\overline{\overline{A}.B}=A+\overline{B}$

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

مثال (13): بسط: Z=A(A+B).

$$Z = AA+AB = A+AB = A$$

مثال (14): استخدم نظريات الجبر البوليني في تبسيط التعبير المنطقي:

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

ثم ارسم الدائرة المنطقية قبل التبسيط وبعده.

الحل:

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

$Y = (\bar{\bar{A}} + \bar{\bar{B}} + \bar{C}).(\bar{\bar{A}} + \bar{B})$ دي مورجان

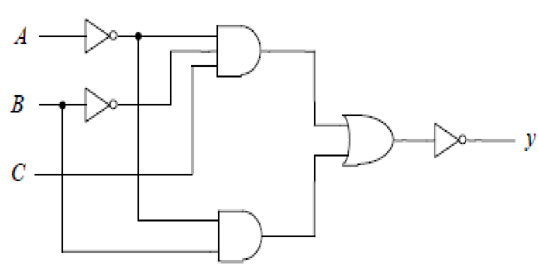
$Y = (A + B + \bar{C}).(A + \bar{B})$ عكس العكس

$Y = AA + A\bar{B} + BA + B\bar{B} + \bar{C}A + C\bar{B}$ التوزيعية

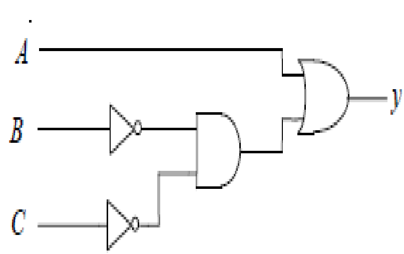
$Y = A(1 + \bar{B} + B + \bar{C}) + \bar{C}\,\bar{B}$ اختزال

$Y = A + \bar{C}\,\bar{B}$ اختزال

الدائرة قبل التبسيط:



الدائرة بعد التبسيط:



نلاحظ ان الدائرة قبل التسبيط مكونة من 6 بوابات واصبحت مكونة من 4 فقط بعد التبسيط.

مثال (15): استخدم الجبر البوليني في تبسيط التعبير المنطقي:

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$$

الحل: لتبسيط هذا النوع من التعبيرات نبحث عن التشابهات ما بين الحدود. والحدان المتشابهان هما حدين يتفقان في كل شي عدا متغير واحد يظهر في احدهما معكوساً وفي الأخر بدون عكس. مثلاً، في التعبير اعلاه الحد الأول $\overline{ABC}$ يشبه الحد الثاني يشبه الحد الثاني $\overline{AB}C$، إذ يتفق الحدان في كل شي عدا المتغير C الذي يظهر في الحد الأول معكوساً وفي الحد الثاني بدون عكس. وبنفس الطريقة يتشابه الحدان الثالث $\bar{A}BC$ والرابع ABC، إذ يتفقان في كل شي عدا المتغير A الذي يظهر في الحد الثالث معكوساً وفي الحد الرابع بدون عكس. مع ملاحظة ان الاختلاف ما بين الحدين المتشابهين يجب ان يكون في متغير واحد فقط.

$$y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$$

بعد ايجاد التشابهات ما بين الحدود نقوم بجمع كل حدين متشابهين في حد واحد هو عبارة عن العامل المشترك ما بين الحدين، اما المتغير المختلف فيتم اختصاره.

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + ABC$$

$Y = \bar{A}\,\bar{B}(\bar{C} + C) + BC(\bar{A} + A)$ بإخراج العامل المشترك في كل حدين متشابهين

$Y = \bar{A}\,\bar{B}(1) + BC(1)$ بجمع المتغير مع عكسه

$Y = \bar{A}\,\bar{B} + BC$ بالعمليات مع 1

نلاحظ في المثال وجود تشابه اضافي بين الحدود، إذ ان الحد الثاني $\overline{ABC}$ يشبه الحد الثالث $\bar{A}BC$، ولكن لم تكن في حاجة لاستخدام هذا التشابه في عملية التبسيط.

مثال (16): استخدم نظريات الجبر البوليبيني في تبسيط التعبير المنطقي:

$$Y = \bar{A}\,\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C$$

الحل: نبحث عن التشابهات ما بين الحدود. الحد الأول يشبه الحد الثاني، والحد الرابع يشبه الحد الخامس، والحد الثالث يشبه الحد الأول.

$$y = \bar A\bar B\bar C + \bar A\bar B C + \bar A B\bar C + A\bar B\bar C + A\bar B C$$

نلاحظ هنا وجود مشكلة تتمثل في ان الحد الأول يتشابه في نفس الوقت مع كل من الحدين الثاني والثالث. في مثل هذه الحالات نقوم بتكرار الحد الأول (مستخدمين نظرية المتغير مع نفسه) إذ يتم جمعه مع كلا الحدين الثاني والثالث.

$$Y = \bar A\bar B\bar C + \bar A\bar B C + \bar A B\bar C + A\bar B\bar C + A\bar B C$$

$$Y = \bar A\bar B\bar C + \bar A\bar B C + \bar A\bar B\bar C + \bar A B\bar C + A\bar B\bar C + A\bar B C$$ بتكرار الحد الأول

$$Y = \bar A\bar B + \bar A\bar C + A\bar B$$ بجمع كل حدين متشابهين

$$Y = \bar A\bar B + A\bar B + \bar A\bar C$$ بالنظرية الإبدالية

$$Y = \bar B + \bar A\bar C$$ بجمع الحدين المتشابهين

مثال (17): استخدم نظريات الجبر البولييني في تبسيط التعبير المنطقي:

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

الحل: نبحث عن التشابهات ما بين الحدود.

$$Y = \overline{\underline{\bar A B\bar C} + A\underline{\bar B\,\bar C} + \underline{A\bar B C} + \underline{A B\bar C}}$$

$$Y = \overline{B\bar C + A\bar B}$$ مجموع كل حدين متشابهين

$$Y = \overline{(B\bar C)}\,.\,\overline{(A\bar B)}$$ من نظرية مورغان

$$Y = (\bar B + C).\,(\bar A + B)$$ من نظرية مورغان

مثال (18): استخدم نظريات الجبر البولييني في تبسيط التعبير المنطقي:

$$Y = \overline{\bar A BC + AB\bar C + ABC}$$

الحل: نبحث عن التشابهات بين الحدود.

$$y = \overline{\bar A BC + AB\bar C + ABC}$$

$$Y = \overline{\bar{A}BC + AB\bar{C} + ABC}$$

$$Y = \overline{(\bar{A}BC + ABC) + (AB\bar{C} + ABC)}$$ بتكرار الحد الثالث

$$Y = \overline{BC + AB}$$ نجمع كل حدين متشابهين

$$Y = \bar{B} + \overline{(C + A)}$$ من نظرية مورغان

مثال (19): بسط الدالة البوليينية الأتية: $F = \bar{X} + \bar{X}\bar{Y}Z + \bar{X}\bar{Y}\bar{Z} + \bar{Y}\bar{Z} + \bar{Y}$.

الحل:

$$F = \bar{X} + \bar{X}\bar{Y}Z + \bar{X}\bar{Y}\bar{Z} + \bar{Y}\bar{Z} + \bar{Y}$$

$$F = \bar{X}(1 + \bar{Y}Z) + \bar{Y}\bar{Z}(\bar{X} + 1) + \bar{Y} \qquad \{X + 1 = 1\}$$

$$F = \bar{X}.1 + \bar{Y}\bar{Z}.1 + \bar{Y} \qquad \{X.1 = 1\}$$

$$F = \bar{X} + \bar{Y}\bar{Z} + \bar{Y}$$

$$F = \bar{X} + \bar{Y}(\bar{Z} + 1)$$

$$F = \bar{X} + \bar{Y}$$

مثال (20): اختصر الدالة البولينية الأتية لأبسط صيغة ممكنه:

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

الحل:

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

$$F = \overline{\overline{(\bar{Y} + \bar{Z})}} + \overline{\overline{(X + \bar{Y} + Z)}} \qquad \{\overline{\overline{X}} = X\}$$

$$F = (\bar{Y} + \bar{Z}) + (X + \bar{Y} + Z)$$

$$F = \bar{Y} + \bar{Z} + X + \bar{Y} + Z$$

$$F = X + \bar{Y} + \bar{Y} + Z + \bar{Z} \qquad \{\bar{Y} + \bar{Y} = \bar{Y},\ Z + \bar{Z} = 1\}$$

$$F = X + \bar{Y} + 1$$

$$F = 1$$

مثال (21): بسط الصيغة الأتية:

$$A = \bar{X}.\bar{Z} + X.\bar{Z}$$

$$= \bar{Z}(\bar{X} + X) = \bar{Z}$$

مثال (22): بسط الصيغة الأتية:

$$
\begin{aligned}
Z &= AB + A(B+C) + B(B+C) \\
&= AB + AB + AC + BB + BC \\
&= AB + AC + B + BC \\
&= AB + AC + B \\
&= B + AC
\end{aligned}
$$

مثال (23): بسط الصيغة الأتية:

$$
\begin{aligned}
Z &= A\overline{B}C + \underline{\overline{A}BC} + \underline{\overline{A}\overline{B}C} \\
&= A\overline{B}C + \overline{A}C[B + \overline{B}] \\
&= A\overline{B}C + \overline{A}C \\
&= C[A\overline{B} + \overline{A}] \\
&= C[\overline{A} + \overline{B}] \\
&= C\overline{A} + C\overline{B}
\end{aligned}
$$

مثال (24): بسط الصيغة الأتية:

$$
\begin{aligned}
F &= A\{ BC(A+B+C+D)\} \\
&= ABC(A+B+C+D) \\
&= ABCA + ABCB + ABCC + ABCD \\
&= ABC+ABC+ABC+ABCD \\
&= ABC+ABCD \\
&= ABC(1+D) \\
&= ABC
\end{aligned}
$$

مثال (25): بسط الصيغة الأتية:

$$
\begin{aligned}
F &= \underline{A\overline{C}} + \underline{ABC} + AC\overline{D} + CD \\
&= A(\overline{C} + BC) + C(A\overline{D} + D) \\
&= A(\overline{C} + B) + C(A+D) \\
&= A\overline{C} + AB + CA + CD \\
&= A(\overline{C} + C) + AB + CD \\
&= A + AB + CD \\
&= A(1+B) + CD \\
&= A + CD
\end{aligned}
$$

مثال (26): اثبت الأتي:

$$F= ABC + A\overline{B}C + AB\overline{C} = A(B+C)$$
$$= AC(B+\overline{B}) + AB\overline{C}$$
$$= AC + AB\overline{C}$$
$$= A(C+B\overline{C})$$
$$= A(C+B)$$

مثال (27): بسط الصيغة الأتية:

$$X = AB + ABC + \overline{A}B + A\overline{B}C$$
$$= AB(1+C) + \overline{A}B + A\overline{B}C$$
$$= AB + \overline{A}B + A\overline{B}C$$
$$= B + A\overline{B}C$$
$$= B + \overline{B}AC$$
$$= B + AC$$

مثال (28): ثلاثة مدخلات **لدائرة منطقية**. والناتج هو (1) عندما تكون الأغلبية من المدخلات هو (1)، وما عدا ذلك صفر، فان جدول الحقيقة هو:

ABC	Z
000	0
001	0
010	0
011	1
100	0
101	1
110	1
111	1

6-2 صورة مجموع الحدود الصغرى (Sum of Minterms):

الحد الاصغر (minterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات AND، وقد يظهر متغير معين في الحد الاصغر معكوسا أو بدون عكس. يوجد عدد من الحدود الصغرى يساوي عدد احتمالات الدخل، اي عدد اسطر جدول الحقيقة. لأيجاد الحد الاصغر المقابل لسطر معين من اسطر جدول

الحقيقة ننظر إلى قيم متغيرات الدخل في ذلك السطر، فاذا كانت قيمة المتغير هي 0 يظهر ذلك المتغير في الحد الاصغر معكوسا، اما إذا كانت قيمة المتغير هي 1 يظهر ذلك المتغير في الحد الاصغر بدون عكس. كما في الجدول (2−11) الذي يوضح جميع الحدود الصغرى لثلاثة متغيرات دخل A وB وC.

الجدول (2−11)

A	B	C	Minterm
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	$\overline{A}\,\overline{B}C$
0	1	0	$\overline{A}B\overline{C}$
0	1	1	$\overline{A}BC$
1	0	0	$A\overline{B}\,\overline{C}$
1	0	1	$A\overline{B}C$
1	1	0	$AB\overline{C}$
1	1	1	ABC

نلاحظ ان اي حد اصغر يساوي 1 فقط لاحتمال الدخل المقابل له في جدول الحقيقة، ويساوي 0 خلاف ذلك، اي يساوي 0 لجميع احتمالات الدخل الأخرى في جدول الحقيقة. لكتابة التعبير المنطقي لمتغير معين من متغيرات الخرج في صورة مجموع الحدود الصغرى ننظر إلى قيم ذلك المتغير في جدول الحقيقة ونبحث عن الـ1‘s، ثم نقوم بأخذ الحدود الصغرى المقابلة لهذه الـ1‘s ونقوم بجمعها، اي نربط بينها بعمليات OR.

مثال (29): اكتب التعبيرين المنطقين لمتغيري الخرج Y وX في جدول الحقيقة الاتي في صورة مجموع الحدود الصغرى.

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

الحل:

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

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

ويمكن بسهولة التحقق من ان التعبير المنطقي المكتوب في صورة مجموع الحدود الصغرى يعطي فعلاً قيم الخرج المطلوبة الموضحة في جدول الحقيقة عند تعويض احتمالات الدخل فيه. إذ تم اختيار الحدود الصغرى المقابلة للـ1‘s فقط في جدول الحقيقة وربطها بعمليات OR، إذ ان الحد الاصغر يساوي 1 فقط لاحتمال الدخل المقابل له في جدول الحقيقة ويساوي 0 خلاف ذلك، فأنه عند تعويض اي احتمال من احتمالات الدخل المقابلة للـ1‘s في التعبير المنطقي فأن الحد الاصغر المقابل لذلك الاحتمال يساوي 1، وبالتالي فان التعبير باكمله يساوي 1 أيضاً. اي ان كل حد من الحدود الصغرى المضمنة في التعبير المنطقي يضمن ظهور الـ1 المقابل له في جدول الحقيقة. اما عند تعويض احد احتمالات الدخل المقابلة للـ0‘s في التعبير المنطقي فان اي من الحدود الصغرى المضمنة في التعبير المنطقي لن يكون مساويا 1 بل ستكون جميعا مساوية للـ0 وبالتالي فان التعبير المنطقي باكمله يساوي 0.

لتسهيل كتابة التعبيرات المنطقية في صورة مجموع الحدود الصغرى يتم ترقيم اسطر جدول الحقيقة (ابتداءا بالقيمة 0)، واستخدام الرمز m_k للحد الاصغر المقابل للسطر k من جدول الحقيقة. كما في الجدول الاتي:

#	A	B	C	Minterm	
0	0	0	0	$\overline{ABC}$	m_0
1	0	0	1	$\overline{A}\,\overline{B}C$	m_1
2	0	1	0	$\overline{A}B\overline{C}$	m_2
3	0	1	1	$\overline{A}BC$	m_3
4	1	0	0	$A\overline{BC}$	m_4
5	1	0	1	$A\overline{B}C$	m_5
6	1	1	0	$AB\overline{C}$	m_6
7	1	1	1	ABC	m_7

وعليه يمكن كتابة التعبيريين المنطقين لمتغيري الخرج x و y في المثال السابق بصورة مختصرة، باستخدام رموز الحدود الصغرى، كالاتي:

$X = m_o + m_1 + m_3 + m_7$
$Y = m_o + m_1 + m_2 + m_4 + m_5$

كما يمكن تسهيل كتابة التعبير باستخدام رمز المجموع ∑، كالاتي:

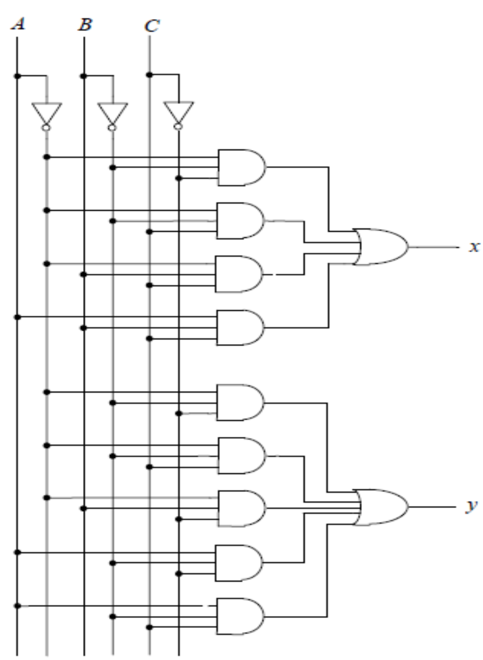
$X = \sum m\,(0,1,3,7)$
$Y = \sum m\,(0, 1,2,4,5)$

اي ان التعبير المنطقي في صورة مجموع الحدود الصغرى يمكن ان يكتب كاملاً، أو مختصرا باستخدام رموز الحدود الصغرى، أو مختصراً باستخدام رمز المجموع $\sum$ وارقام الحدود الصغرى (اي ارقام اسطر جدول الحقيقة الحاوية على القيمة 1 للمتغير الذي يتم كتابة التعبير له). وبناء عليه فانه في المثال السابق:

$$X = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + ABC$$
$$= m_0 + m_1 + m_3 + m_7$$
$$= \sum m\,(0,1,3,7) \ldots (1)$$
$$Y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$
$$= m_0 + m_1 + m_2 + m_4 + m_5$$
$$= \sum m\,(0,1,2,4,5) \ldots (2)$$

نلاحظ هنا اهمية مراعاة استخدام الحرف m الصغير (Small Letter) للرمز للحدود الصغرى (Minterms)، حتى لا يحدث خلط بينها وبين الحدود الكبرى (Maxterms)، إذ تم استخدام الحرف M الكبير (Capital Letter) للرمز للحدود الكبرى.

إذا تم رسم الدائرة المنطقية للتعبيرين (x، y) المكتوبين في صورة مجموع الحدود الصغرى نحصل على:



7-2 صورة مضروب الحدود الكبرى (Product of Maxterms):

الحد الاكبر (maxterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات OR، وقد يظهر متغير معين في الحد الاكبر معكوسا أو بدون عكس. يوجد عدد من الحدود الكبرى يساوي عدد احتمالات الدخل، اي عدد اسطر جدول الحقيقة، إذ يكون هناك حد اكبر مقابل كل سطر من اسطر جدول الحقيقة. لأيجاد الحد الاكبر المقابل لسطر معين من اسطر جدول الحقيقة ننظر إلى قيم متغيرات الدخل في ذلك السطر، فاذا كانت قيمة المتغير هي 0 يظهر ذلك المتغير في الحد الاكبر بدون عكس، اما إذا كانت قيمة المتغير هي 1 يظهر ذلك المتغير في الحد الاكبر معكوسا. كما موضح في الجدول الاتي، الذي يوضح جميع الحدود الكبرى لثلاثة متغيرات دخل A و B و C.

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

نلاحظ ان اي حد اكبر يساوي 0 فقط لاحتمال الدخل المقابل له في جدول الحقيقة، ويساوي 1 خلاف ذلك، اي يساوي 1 لجميع احتمالات الدخل في جدول الحقيقة.

لكتابة التعبير المنطقي لمتغير معين من متغيرات الخرج في صورة مضروب الحدود الكبرى ننظر إلى قيم ذلك المتغير في جدول الحقيقة ونبحث عن الـ0's، ثم نقوم بأخذ الحدود الكبرى المقابلة لهذه الـ0's ونقوم بضربها، اي نربط بينها بعمليات AND.

مثال (30): اكتب التعبيريين المنطقين لمتغيري الخرج X و Y بصيغة مضروب الحدود الكبرى في جدول الحقيقة الاتي:

A	*B*	*C*	*x*	*y*
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

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

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

ويمكن التحقق من ان التعبير المنطقي المكتوب في صورة مضروب الحدود الكبرى يعطي قيم الخرج المطلوبة الموضحة في جدول الحقيقة عند تعويض احتمالات الدخل فيه. إذ تم اختيار الحدود الكبرى المقابلة للـ0's فقط في جدول الحقيقة وربطها بعمليات AND، إذ ان الحد الاكبر يساوي 0 فقط لاحتمال الدخل المقابل له في جدول الحقيقة ويساوي 1 خلاف ذلك، فعند تعويض أي احتمال من احتمالات الدخل المقابلة للـ0's في التعبير المنطقي فان الحد الاكبر المقابل لذلك الاحتمال يساوي 0، وبالتالي فان التعبير باكمله يساوي 0 أيضاً. اي ان كل حد من الحدود الكبرى المضمنة في التعبير المنطقي يضمن لنا ظهور الـ0 المقابل له في جدول الحقيقة. اما عند تعويض احد احتمالات الدخل المقابلة للـ1's في التعبير المنطقي فان اي من الحدود الكبرى المضمنة في التعبير المنطقي لن يكون مساوياً 0 بل ستكون جميعا مساوية للـ1 وبالتالي فان التعبير المنطقي باكمله يساوي 1.

لتسهيل كتابة التعبيرات المنطقية في صورة مضروب الحدود الكبرى يتم ترقيم اسطر جدول الحقيقة (ابتداءاً بالقيمة 0)، واستخدام الرمز M_K للحد الاكبر المقابل للسطر K من جدول الحقيقة. كما في الجدول الاتي:

#	A	B	C	Maxterm	
0	0	0	0	$A+B+C$	M_0
1	0	0	1	$A+B+\overline{C}$	M_1
2	0	1	0	$A+\overline{B}+C$	M_2
3	0	1	1	$A+\overline{B}+\overline{C}$	M_3
4	1	0	0	$\overline{A}+B+C$	M_4
5	1	0	1	$\overline{A}+B+\overline{C}$	M_5
6	1	1	0	$\overline{A}+\overline{B}+C$	M_6
7	1	1	1	$\overline{A}+\overline{B}+\overline{C}$	M_7

وعليه يمكن كتابة التعبيريين المنطقين لمتغيري الخرج X و Y في المثال السابق بصورة مختصرة، باستخدام رموز الحدود الكبرى كالاتي:

$X = M_2 . M_4 . M_5 . M_6$
$Y = M_3 . M_6 . M_7$

كما يمكن تسهيل كتابة التعبيرات اكثر من ذلك باستخدام رمز المضروب $\prod$، وذلك كالاتي:

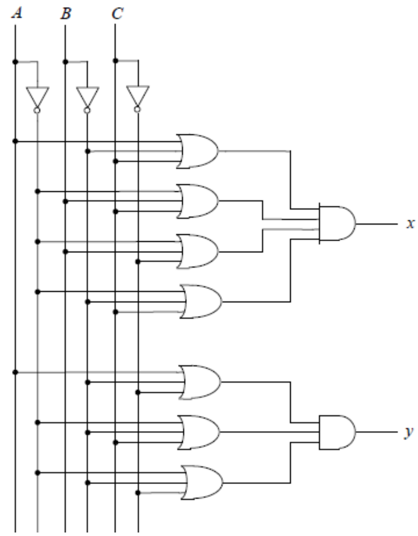
$X = \prod M(2,4,5,6)$
$Y = \prod M(3,6,7)$

اي ان التعبير المنطقي في صورة مضروب الحدود الكبرى يمكن ان يكتب كاملا، أو مختصرا باستخدام رموز الحدود الكبرى، أو مختصرا باستخدام رمز المضروب $\prod$ وارقام الحدود الكبرى (اي ارقام اسطر جدول الحقيقة الحاوية على القيمة 0 للمتغير الذي يتم كتابة التعبير له). وبناء عليه فانه في المثال السابق:

$$\begin{aligned} X &= (A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C) \\ &= M2.M4.M5.M6 \\ &= \prod M(2,4,5,6) \end{aligned}$$

$$\begin{aligned} Y &= (A+\overline{B}+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C}) \\ &= M3.M6.M7 \\ &= \prod M(3,6,7) \end{aligned}$$

واذا تم رسم الدائرة المنطقية للتعبيريين أعلاه المكتوبين في صورة مضروب الحدود الكبرى نحصل على:



مثال (31): اكتب الصيغة المنطقية باستخدام طريقة جمع الضرب (sum of product) لجدول الحقيقة الأتية؟

AB	Z	P terms	Z
00	1	$\overline{A}\,\overline{B}$	
01	0	$\overline{A}\,\overline{B}$	$Z = \overline{A}\,\overline{B} + AB$
10	0	$A\overline{B}$	
11	1	AB	

مثال (32): اكتب الصيغة المنطقية باستخدام طريقة جمع الضرب (sum of product) لجدول الحقيقة الأتية؟ مع تبسيط المقدار.

ABC	Z	→	P terms	min terms
000	0		$\overline{A}\,\overline{B}\,\overline{C}$	m_0
001	0		$\overline{A}\,\overline{B}\,C$	m_1
010	0		$\overline{A}\,B\,\overline{C}$	m_2
011	1		$\overline{A}\,B\,C$	m_3
100	0		$A\,\overline{B}\,\overline{C}$	m_4
101	1		$A\,\overline{B}\,C$	m_5
110	1		$A\,B\overline{C}$	m_6
111	1		$AB\,C$	m_7

$$
\begin{aligned}
Z &= m_3 + m_5 + m_6 + m_7 \\
&= \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC \\
&= BC(\bar{A} + A) + A\bar{B}C + AB\bar{C} \\
&= BC + A\bar{B}C + AB\bar{C} \\
&= C(B+\bar{B}A) + AB\bar{C} \;= C(B+A) + AB\bar{C} \\
&= CB + CA + AB\bar{C} = CB + A(C+B\bar{C}) \\
&= CB + A(C+B) \\
&= CB + AC + AB
\end{aligned}
$$

مثال (33): اكتب الصيغة المنطقية باستخدام طريقة ضرب الجمع (product of sum) لجدول الحقيقة الآتي؟

Z	S. treams ⟹	Max terms	AB
00	1	$(A+B)$	M_0
01	0	$(A+\bar{B})$	← M_1
10	0	$(\bar{A}+B)$	← M_2
11	0	$(\bar{A}+\bar{B})$	← M_3

$$
\begin{aligned}
Z &= M_1 \,.\, M_2 \,.\, M_3 \\
&= (A+\bar{B})(\bar{A}+B)(\bar{A}+\bar{B})
\end{aligned}
$$

مثال (34): بسط الصيغة الأتية باستخدام طريقة جمع الضرب (sum of product) وضرب الجمع (product of sum):

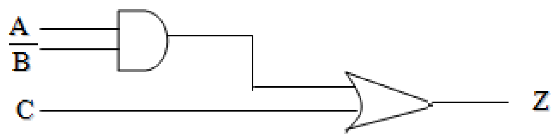
$$F(A,B,C) = \pi(M_2, M_3, M_6)$$

الحل:

ABC	Z	
000	1	m_0
001	1	m_1
010	0	m_2
011	0	m_3
100	1	m_4
101	1	m_5
110	0	m_6
111	1	m_7

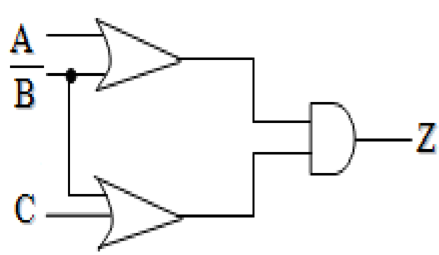
1. بطريقة جمع الضرب:

$$Z = m_o + m_1 + m_4 + m_5 + m_7$$
$$= \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + A\bar{B}\bar{C} + A\bar{B}C + ABC$$
$$= \bar{A}\,\bar{B}\,(C+\bar{C}) + A\bar{B}\,(C+\bar{C}) + ABC$$
$$= \bar{A}\,\bar{B} + A\bar{B} + ABC$$
$$= \bar{B}\,(\bar{A}+A) + ABC = \bar{B} + BAC = \bar{B} + AC$$



2. بطريقة ضرب الجمع:

$$Z = M_2 . M_3 . M_6$$
$$=(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)$$
$$=(A+\bar{B}+C)(A+\bar{B}+\bar{C})(\bar{A}+\bar{B}+C)(A+\bar{B}+C)$$
$$=(AA+\bar{B}A+CA+A\bar{B}+\bar{B}\bar{B}+C\bar{B}+A\bar{C}+\bar{B}\bar{C}+C\bar{C})$$
$$=(A+\bar{B}A+CA+A\bar{B}+\bar{B}+C\bar{B}+A\bar{C}+\bar{B}\bar{C})$$
$$=(A(1+\bar{B}+C+\bar{B}+\bar{C})+\bar{B}(1+\bar{C}+C))$$
$$=(A+\bar{B})(\bar{A}A+\bar{B}A+CA+\bar{A}\bar{B}+\bar{B}\bar{B}+C\bar{B}+\bar{A}C+\bar{B}C+CC)$$
$$=(A+\bar{B})(\bar{B}A+CA+\bar{A}\bar{B}+\bar{B}+C\bar{B}+\bar{A}C+\bar{B}C+C)$$
$$=(A+\bar{B})(\bar{B}(A+\bar{A}+1+C+C)+C(A+\bar{A}+1))$$
$$=(A+\bar{B})(\bar{B}+C)$$



يتطلب طريقة **ضرب الجمع بوابة** واحدة أكثر مقارنة بـ **جمع الضرب**.

تمارين الفصل الثاني

1) اكتب جدول الحقيقة للصيغ الأتية، مع رسم الدائرة المنطقية لكل منها.

a) $XYZ + \overline{X}\,\overline{Y}\,\overline{Z}$

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

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

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

2) طبق قانون دي مورجان للصيغ الأتية:

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

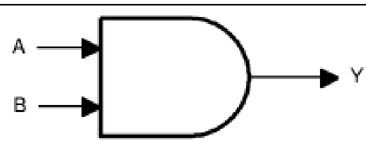
b) $\overline{A\overline{B}(C + \overline{D})}$

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

d) $A + B[AC + \overline{(A + \overline{C})}$

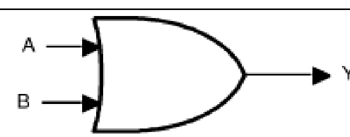
3) اكتب جدول الحقيقة للدوائر المنطقية الاتية:

1



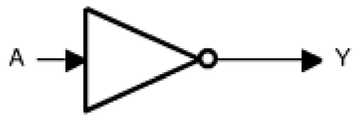
A	B	Y

3



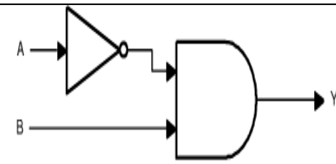
A	B	Y

2

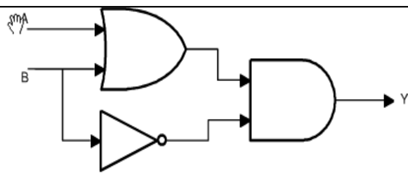


A	Y

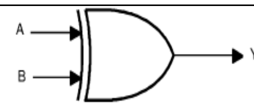
4



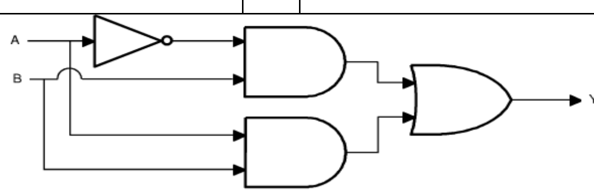
5



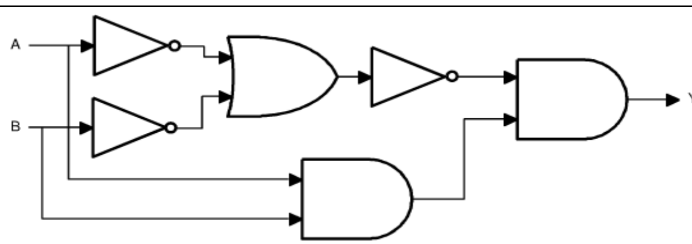
6

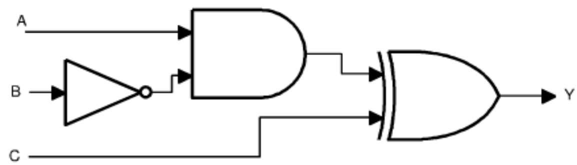


7



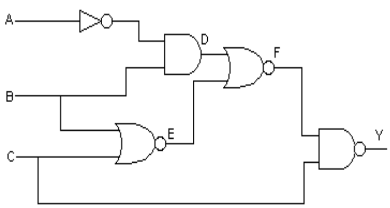
8

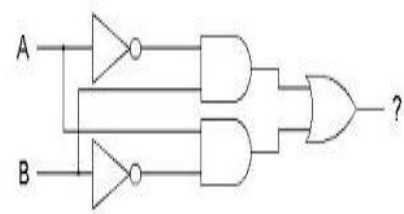


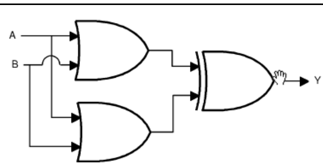
<table>
<tr><td>9</td><td>


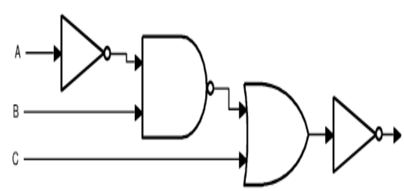
</td></tr>
</table>

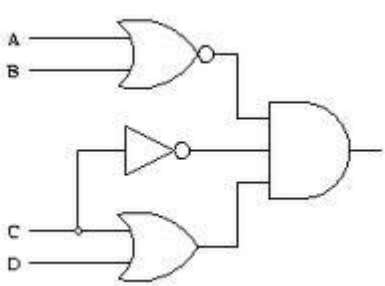
4) اكتب جدول الحقيقة للدوائر المنطقية الاتية:

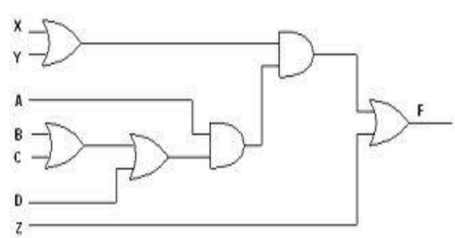
<table>
<tr><td>1.</td><td>


</td><td>4.</td><td>


</td></tr>
<tr><td>2.</td><td>


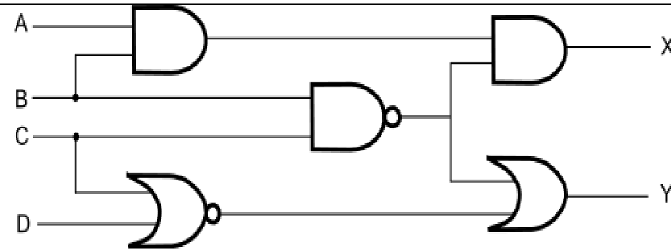
</td><td>5.</td><td>


</td></tr>
<tr><td>3.</td><td>


</td><td>6.</td><td>


</td></tr>
</table>

7.



5) حول الصيغ الأتية إلى جمع الضرب:

a) (A+B) (C+D)
b) (A+C) (ABC+ACD)
c) $A+B\,[AC+(B+\overline{C})D)$

6) ارسم الدائرة المنطقية للصيغ البوليينية الأتية:

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

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

c) $X=A[BC(A+B+C+D)$

d) $X=\overline{ABC}+B(EF+\overline{G})$

e) $X=B(C\overline{D}E+\overline{E}FG)(AB+C)$

7) بسط الصيغ البوليينية الأتية:

a) $ABCD+AB(\overline{CD})+\overline{AB}CD$

b) $XY(\overline{X}Y\overline{Z}+X\overline{Y}\overline{Z}+\overline{X}\overline{Y}\overline{Z})$

8) اكتب الصيغة البوليينية للعبارات الأتية:

a) X is 1 only if A is 1 and B is 1 or if A=0 and B=0.
b) X =0 if any three variables A, B, C are 1’s.

9) بسط الصيغ البوليينية الأتية:

a) $F(x,y,z) = \Sigma(2,3,6,7)$
b) $F(A,B,C,D,E) = \Sigma(4,15,20,23,31)$
c) $F(A,B,C) = \prod(1,4,5,7)$
d) $F(A,B,C,D) = \Sigma(0,2,5,7,8,10,13,15)$
e) $F(A,B,C) = \prod(1,4,7)$