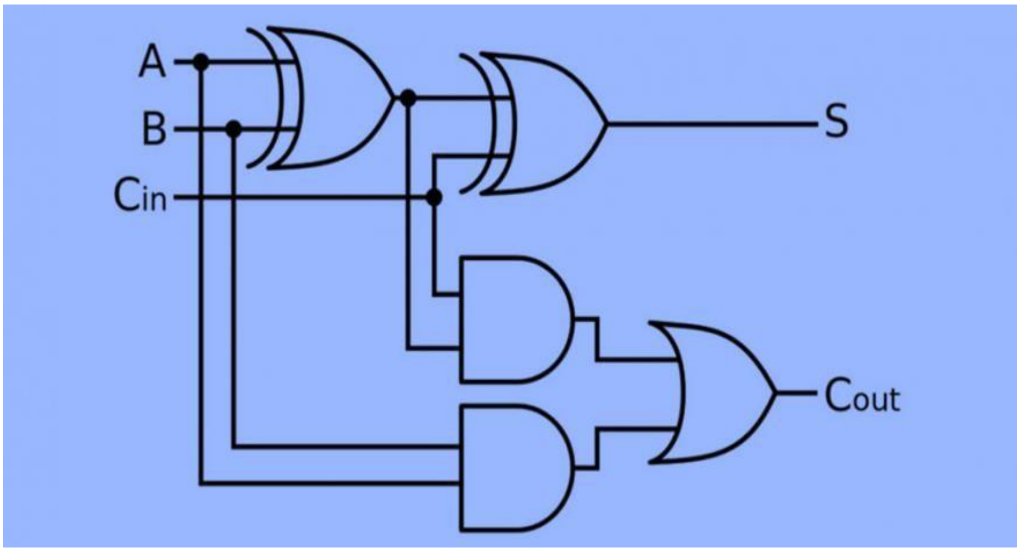
مباديء الالكترونيك الرقمي

((الجزء الأول))

Digital Electronic Principles

Part 1

الأول أجهزة طبية



مدرس المادة: أ.م. عامر محمد نوري النعيمي

للعام الدراسي

2025-2024

Chapter 1

(Numerical Systems) الأنظمة العددية

1- Decimal System: (النظام العشري)

وهو النظام العددي المتعارف عليه والمستخدم على مستوى العالم ، وسمي بهذا الأسم لان عدد الرموز الداخلة في تركيبه عددها (عشرة) وهي: (٠,١,٢,٣,٤,٥,٦,٧,٨,٩). وفي حالة استخدام أكثر من رمز فان القيمة العددية تعتمد على موقع الرمز .
ان عدد الرموز الداخلة في تركيب النظام العددي تسمى (أساس النظام) مثل:
أساس النظام العشري هو العدد ١٠ وأساس النظام الثنائي هو العدد ٢ وهكذا

وبالتالي فإنه يمكن تمثيل العدد 128 طبقاً لذلك كما يلي:

1		2		8	
مرتبة المئات		مرتبة العشرات		مرتبة الآحاد	
10^2		10^1		10^0	
1×10^2	+	2×10^1	+	8×10^0	
100	+	20	+	8	=128

Example 1.1: Analyze The decimal number $(7654)_{10}$

$$(7\ 6\ 5\ 4)_{10}$$

$$7 \times 10^3 \qquad 6 \times 10^2 \qquad 5 \times 10^1 \qquad 4 \times 10^0$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$7 \times 1000 + 6 \times 100 + 5 \times 10 + 4 \times 1 = 7654$$

Example 1.2: Analyze the following decimal number 5631

$(5631.01)_{10} = (5\times10^3) + (6\times10^2) + (3\times10^1) + (1\times10^0) =$

$(5000) + (600) + (30) + (1) = (5631)_{10}$

2- Binary System: (النظام الثنائي)

يطلق على النظام الثنائي اسم نظام الأساس اثنين (2) ويشار إليه بالأساس (2) لأنه يعتمد على رمزين اثنين فقط هما (1 و0). ومراتب الخانات في النظام الثنائي من اليمين إلى اليسار تمثل قوى العدد (2) أي أن:

$$\begin{array}{cccccc} \ldots\ldots & 2^4 & 2^3 & 2^2 & 2^1 & 2^0 \\ & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ & 16 & 8 & 4 & 2 & 1 \end{array}$$

لتحليل العدد $(110)_2$ الى مراتبه:

$$(110)_2 \rightarrow 1\times2^2 \quad 1\times2^1 \quad 0\times2^0$$

وهناك بعض المصطلحات المستخدمة مع هذا النظام الثنائي منها :

■ **الخانة الثنائية** *(Bit)*: الخانة الثنائية *(Bit)* هي اختصار لكلمتي *(Binary Digit)* والتي تعني الخانة الثنائية أو الرقم الثنائي. ويستخدم هذا المصطلح للتعبير عن عدد الأرقام (الخانات) التي يتكون منها العدد الثنائي، فمثلاً العدد $(1001)_2$ يتكون من (4-bits) أو أربع خانات ثنائية وكذلك العدد $(1101101)_2$ يتكون من (7-bits) أو سبع خانات ثنائية وهكذا.

■ **عدد التشكيلات الثنائية** *(Number of Binary Combinations)*: عدد التشكيلات الثنائية تعني عدد الاحتمالات التي يمكن الحصول عليها من عدد معين من الخانات *(bits)*. وهناك صيغة رياضية يمكن عن طريقها حساب هذا العدد من التشكيلات وهي :

$$N = 2^n$$

حيث: N = عدد التشكيلات الثنائية المحتملة

n = عدد الخانات (bits)

وبالتالي فإذا كان عدد الخانات يساوي (2) فإن عدد التشكيلات الثنائية هو :

$$N = 2^2 = 4$$

وإذا كان عدد الخانات يساوي (3) فإن عدد التشكيلات الثنائية هو :

$$N = 2^3 = 8$$

وإذا كان عدد الخانات يساوي (4) فإن عدد التشكيلات الثنائية هو :

$$N = 2^4 = 16$$

عدد ال bits (n)	عدد التشكيلات	التشكيلات
n= 1 bit	N= 2 values	0,1
n= 2 bits	N= 4 values	00, 01, 10, 11
n= 3 bits	N= 8 values	000, 001, 010, 011, 100, 101, 110, 111.
n = 4 bits	N= 16 values	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111.

Bit=2

Decimal	binary
0	00
1	01
2	10
3	11

Bit=3

Decimal	binary
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Bit=4

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

■ **أهمية رتبة الخانة الثنائية** (Bit): في أي تشكيلة من التشكيلات الثنائية المحتملة لأي عدد من الخانات نجد أن الخانة الأولى في اليمين تحت مرتبة 2^0 أي تساوي (1) أو يقال وزنها يساوي (1) وأن الخانة الثانية والتي على يسار الأولى تحت مرتبة 2^1 أي وزنها يساوي (2) والثالثة تحت مرتبة 2^2 أي وزنها يساوي (4) وهكذا. وبذلك نجد أن الخانة الثنائية الأولى التي في أقصى اليمين أقل وزناً وأن الخانة الأخيرة وهي آخر خانة على اليسار هي الأكبر وزناً، ولذلك يطلق على الخانة الثنائية الأولى، الخانة الأقل وزناً أو الأقل قيمة (***L**east **S**ignificant **B**it*) وتكتب اختصاراً (***LSB***) ويطلق على الخانة الثنائية الأخيرة في أقصى اليسار الخانة الأكبر وزناً أو الأعلى قيمة(***M**ost **S**ignificant **B**it*) وتكتب اختصاراً (***MSB***).

1001

MSB ← → **LSB**

Conversions between numerical systems

ان عملية التحويل بين الأنظمة العددية من العمليات المهمة في تصميم الانظمة الرقمية. ولتسهيل فهم هذه التحويلات ، سيتم تقسيمها الى مجاميع كل مجموعة تتشابه بطريقة التحويل.

A) *Binary to Decimal- Conversion*

لتحويل أي عدد في النظام الثنائي الى النظام العشري، يتم تحليل العدد الى مراتبه-اعتمادا على أساس النظام (2) ثم ايجاد ناتج جمع الحدود ، والعدد الناتج من الجمع سيكون هو العدد في النظام العشري.

Example 1.3: **Convert the following binary number 110101 to its decimal equivalent**

1 1 0 1 0 1

$1*2^0 = 1*1 = 1$

$0*2^1 = 0*2 = 0$

$1*2^2 = 1*4 = 4$

$0*2^3 = 0*8 = 0$

$1*2^4 = 1*16 = 16$

$1*2^5 = 1*32 = 32$

53

Example 1.4: **Convert the number $(1101)_2$ to the decimal system.**

$(1\,1\,0\,1)_2 = 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0$

$= 1\times8 + 1\times4 + 0\times2 + 1\times1$

$= 8 + 4 + 0 + 1$

$= (13)_{10}$

Example 1.5: **Convert the following binary number 110111011 to its decimal number**

$(110111011)_2 = 1*2^8 + 1*2^7 + 0*2^6 + 1*2^5 + 1*2^4 + 1*2^3 + 0*2^2 + 1*2^1 + 1*2^0$

$= 256 + 128 + 32 + 16 + 8 + 2 + 1$

$= (443)_{10}$

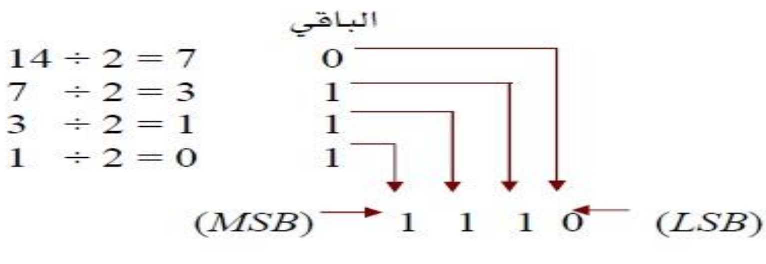
H.W1.1: **1- Convert the number $(11001)_2$ to the decimal system.**

2- Convert the number $(10001)_2$ to the decimal system.

B) Decimal-to-Binary Conversion

لتحويل العدد العشري الى النظام الثنائي نقوم بتقسيم العدد العشري على أساس النظام الثنائي(2) ونحتفظ بباقي القسمة ، ثم ناخذ ناتج القسمة ونقسمه مرة اخرى على الأساس (2) ونحتفظ بالباقي وهكذا نستمر بتكرار العملية الى ان نحصل على ناتج قسمة يساوي صفر ، فيكون ناتج التحويل في عمود باقي القسمة بقرائته من الأسفل الى الأعلى وكتابته من اليسار الى اليمين.

Example 1.6: **Convert the following decimal number $(14)_{10}$ to binary.**



وعلى ذلك يكون:

$(14)_{10} = (1110)_2$

Example 1.7: Convert the following decimal number $(87)_{10}$ to binary

	الباقي	
$87 \div 2 = 43$	1	(LSB)
$43 \div 2 = 21$	1	
$21 \div 2 = 10$	1	
$10 \div 2 = 5$	0	
$5 \div 2 = 2$	1	
$2 \div 2 = 1$	0	
$1 \div 2 = 0$	1	(MSB)

ويكون الناتج:

$(87)_{10} = (1010111)_2$

H.W1.2: Prove the following by using ***Decimal-to-Binary Conversion:***

- $(51)_{10} = (110011)_2$
- $(123)_{10} = (1111011)_2$
- $(101)_{10} = (1100101)_2$

3- Hexadecimal numeral system نظام العد السداسي عشر

ان اساس هذا النظام هو العدد 16 والجدول التالى يبين رموز (أرقام) هذا النظام والأعداد الثنائية والعشرية الى تكافئها

16^2	16^1	16^0
256	16	1

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

a- binary to hexadecimal conversion

١. نضيف أصفار على اليسار حتى يصبح عدد الخانات من مضاعفات الـ 4
٢. نجزيء العدد الى مجموعات مكونة من 4 خانات ابتداءا من اليمين.
٣. نحول كل مجموعة الى الرقم الموافق في النظام السداسي عشر

Example1.8: Convert the binary number 1011110011 to hexadecimal

0010	1111	0011
2	F	3

وسيكون الناتج النهائي هو الاتي: $(1011110011)_2 = (2F3)_{16}$

b- hexadecimal to binary conversion

١. نحول كل خانة من الرقم السداسي عشر الى الرقم الموافق لها في النظام الثنائي
٢. نقوم بتقريب هذه الارقام من بعضها البعض لتصبح سلسلة واحدة من الـ bits
٢. نحذف الأصفار (ان وجدت) جهة اليسار .

Example 1.9: Convert the hexadecimal number F83 to its binary equivalent

نقوم بتحويل كل خانة من الرقم السابق الى ما يوافقها من النظام الثنائي

F	8	3
1111	1000	0011

وبالتالي يصبح الناتج: $(F83)_{16} = (1111\ 1000\ 0011)_2$

Example 1.10: Convert the hexadecimal number AB2 to its binary equivalent

نقوم بتحويل كل خانة من الرقم السابق الى ما يوافقها من النظام الثنائي من الجدول السابق فيكون الناتج:

$$(AB2)_{16}$$

$$(1010\ 1011\ 0010)_2$$

$(AB2)_{16} = (101010110010)_2$

c- hexadecimal to Decimal conversion

Example1.11: Convert the hexadecimal number A2 to its Decimal equivalent

Solution:

A2 = $(A*16^1) + (2*16^0)$

= (10 * 16) + (2 * 1) = (160 + 2) = 162

H.W1.3: Find the equivalent number of each of

1- $(10011001)_2 = (\quad)_{16}$
2- $(CE1)_{16} = (\quad)_2$
3 - $(B3)_{16} = (\quad)_{10}$
4- $(A5)_{16} = (\quad)_2$
5- $(10101011011)_2 = (\quad)_{16}$

Arithmetic operations in the binary system

العمليات الحسابية التي تتم باستخدام الاعداد العشرية مثل الجمع والطرح والضرب والقسمة يمكن اجراؤها في الأنظمة العددية الأخرى, ولأهمية النظام الثنائي في الدوائر الرقمية , فاننا سنقوم بدراسة تلك العمليات في النظام الثنائي

Binary Addition الجمع في النظام الثنائي

ان أبسط عملية جمع في النظام الثنائي هي التي تتم بين عددين كل عدد يتكون من رمز ثنائي واحد. ولو أخذنا كافة الاحتمالات لهذه العملية فستكون الاحتمالات مبينة في أدناه:

$$0 + 0 = 0$$

$$0 + 1 = 1$$

$$1 + 0 = 1$$

$$1 + 1 = 0 \longrightarrow 1 \text{ محمل (Carry)}$$

$$1 + 1 + 1 = 1 \longrightarrow 1 \text{ (carry)}$$

Example1.12: Add the two numbers $(11010)_2$ ، $(1011)_2$

```
  1 1 0 1 0
  0 1 0 1 1    +
-------------
1 0 0 1 0 1
```

Example 1.13: Add the two numbers $(1110)_2$ ، $(11011)_2$

```
  1 1 0 1 1
  0 1 1 1 0    +
-------------
1 0 1 0 0 1
```

Binary Multiplication : الضرب في النظام الثنائي

ان احتمالات عملية الضرب في النظام الثنائي هي:

$$0 \times 0 = 0$$

$$0 \times 1 = 0$$

$$1 \times 0 = 0$$

$$1 \times 1 = 1$$

Example 1.14: Find the product of the two numbers $(1010)_2$ and $(101)_2$

$$\begin{array}{r} 1010 \\ \times \quad 101 \\ \hline 1010 \\ 00000 \\ 101000 \\ \hline 110010 \end{array}$$

Binary Subtraction : الطرح في النظام الثنائي

هناك عدة طرق لعملية الطرح:

1-الطريقة المباشرة

كما في عملية الجمع , هناك أربعة احتمالات كما مبينة ادناه:

$$0 - 0 = 0$$

$$0 - 1 = 1 \longrightarrow 1$$ استعارة (Borrow)

$$1 - 0 = 1$$

$$1 - 1 = 0$$

Example 1.15: Subtract the number $(1011)_2$ from the number $(1101)_2$

$$\begin{array}{r} 1101 \\ 1011 \; - \\ \hline 0010 \end{array}$$

H.W1.4: Find the results of mathematic operations

A) $(1101)_2 + (1101)_2$

B) $(1100)_2 - (1000)_2$ by using the direct method.

C) (10011) * (101)

** ملاحظة مهمة: بالامكان التحقق من الناتج النهائي للعمليات الحسابية عن طريق تحويلها من ثنائي الى عشري واجراء العمليات الحسابية عليها ثم مقارنتها مع النتيجة الثنائية.

-2 الطريقة الثانية للطرح: المتممات (المكملات) Complements

المتممات في النظام الثنائي :

هناك نوعان من المتممات في النظام الثنائي هما: .

1. المتمم الأحادي (1's Complement) مقلوب العدد (أي جعل كل واحد صفر وكل صفر واحد).

Example1.16: Find the 1's complement of 10110010

Solution:

10110010	Binary number
01001101	1 's complement

2. المتمم الثنائي (2's Complement) هو المتمم الأحادي مضافا اليه 1

Example1.17:

Find the 2's complement of 10110010.

Solution

10110010	Binary number
01001101	l's complement
+ 1	add 1
01001110	2's complement

Example1.18: Find the 2's complement of 10111000

Solution:

10111000	Binary number
01000111	1 's complement
+ 1	
01001000	2's complement

H.W1.5: Find the 1's complement and 2's complement of $(10011011)_2$

Chapter 2

Logical Gates (البوابات المنطقية)

معظم الأنظمة الرقمية مثل الحاسبات و أجهزة معالجة البيانات و أجهزة التحكم و أجهزة القياس- وأنظمة الاتصالات الرقمية، تحتوي على مجموعة من الدوائر المنطقية التي تؤدي بعض العمليات الأساسية والتي يتكرر تنفيذها كثيراً وبسرعة كبيرة جداً، وهذه العمليات الأساسية هي في الواقع مجموعة من العمليات المنطقية، ولذلك تسمى الدوائر البسيطة التي تقوم بهذه العمليات بالدوائر المنطقية.
وتمثل البوابات المنطقية حجر الأساس لبناء أي دائرة منطقية ومن ثم أي نظام رقمي أو منطقي، وحيث إن كلمة منطق ترمز إلى "عملية صنع القرار" لذا فإن بوابة المنطق هي البوابة التي تعطي خرجاً فقط عندما تتحقق شروط معينة على مدخلات هذه البوابة.
وفي هذه الوحدة سوف نتناول بالدراسة الأنواع المختلفة للبوابات المنطقية وسنبدأ بالبوابات الأساسية وهي بوابة *AND* و بوابة *OR* وبوابة *NOT* أو العاكس(*INVERTER*). ومن خلال التركيبات البسيطة لهذه البوابات الثلاث يمكننا الحصول على باقي أنواع البوابات الأخرى، ثم نقوم بعد ذلك بدراسة كيفية تجميع هذه البوابات لتمثيل الدوائر المنطقية البسيطة.

1- AND Gate:

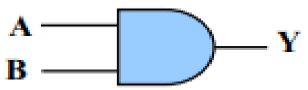
تعتبر البوابة *AND* واحدة من البوابات الأساسية والتي تدخل في بناء معظم الدوال المنطقية (*Logic Functions*). والبوابة *AND* لها مدخلان أو أكثر ولها خرج واحد، وتؤدي هذه البوابة ما يسمى بالضرب المنطقي (*Logical Multiplication*)

يوضح الشكل (2.1) الرمز المنطقي القياسي (standard) للبوابة (AND) حيث يظهر المدخلان (2 inputs) و اخراج واحد (1 output) و الجدول (2.1) يبين جدول الحقيقة (truth table) للبوابة AND بمدخليين . الجتول (2.1)

The truth table for 2- input AND Gate

INPUTS A	INPUTS B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

1 = HIGH, 0 = LOW



الشكل (2.1)

Logic symbol for the AND gate

يعتبر الجبر البوليني (*Boolean Algebra*) صيغة للمنطق الرمزي والذي يبين كيف تعمل البوابات المنطقية، والعبارة البولينية (*Boolean Expression*) هي طريقة مختصرة لإظهار ماذا يحدث في دائرة منطقية ما. والعبارة البولينية لبوابة *AND* ذات مدخلين هي:

$Y = A \bullet B$ ⟵ الصيغة الرياضية لبوابة AND باستخدام (الجبر البولي)

وتقرأ هذه العبارة كالتالي: الخرج Y يساوي A AND B (• تعني *AND*)، وأحيانا تحذف النقطة من العبارة البولينية وتصبح:

$Y = AB$

وتقرأ الخرج Y يساوي A AND B.

قواعد الجبر البوليني في بوابة AND:(حرف ال A مجرد رمز)

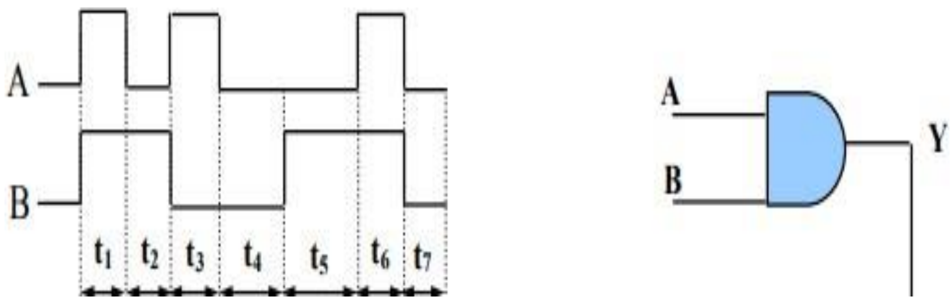
$A.0 = 0$ -a

$A.\bar{A} = 0$ -b

$A.1 = A$ -c

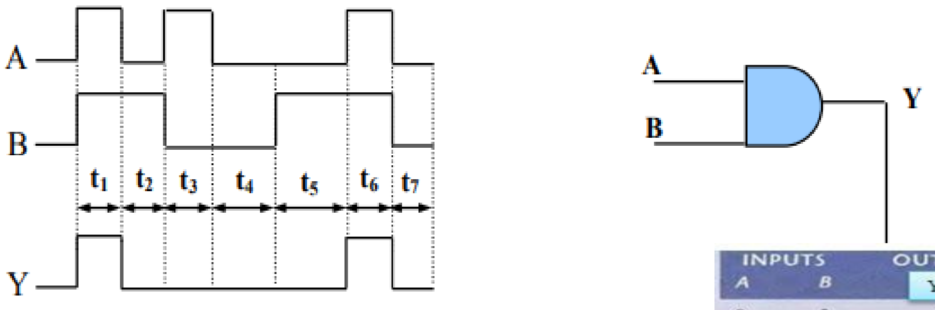
$A.A = A$ -d

Example 2.1: Find the output Y in timing diagram for the AND Gate has **2 input** (A,B) as shown in figure below



the timing diagram of AND Gate for 2- inputs

Solution:



INPUTS A	INPUTS B	OUTPUT Y
0	0	0
0	1	0
1	0	0
1	1	1

1 = HIGH, 0 = LOW

Y= A.B

H.W2.1: write the symbol and the truth table of **AND Gate** has **3- input** and **one output.**

2-OR Gate

تعتبر البوابة *OR* واحدة من البوابات الأساسية والتي تدخل في بناء معظم الدوال المنطقية. والبوابة *OR* لها مدخلان أو أكثر ولها خرج واحد ، وتؤدي هذه البوابة ما يسمى بالجمع المنطقي (*Logical Addition*)

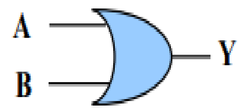
يوضح الشكل (2.2) الرمز المنطقي القياسي (standard) للبوابة (OR) حيث يظهر المدخلان (2 inputs) و اخراج واحد (1 output) و الجدول (2.2) يبين جدول الحقيقة (truth table) للبوابة OR بمدخلين.

الجدول (2.2)

The truth table for 2- input OR Gate

INPUTS A	INPUTS B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	1

1 = HIGH, 0 = LOW



الشكل (2.2)

Logic symbol for the OR gate

العبارة البولية لبوابة OR ذات مدخلان هي :

$$Y = A + B$$

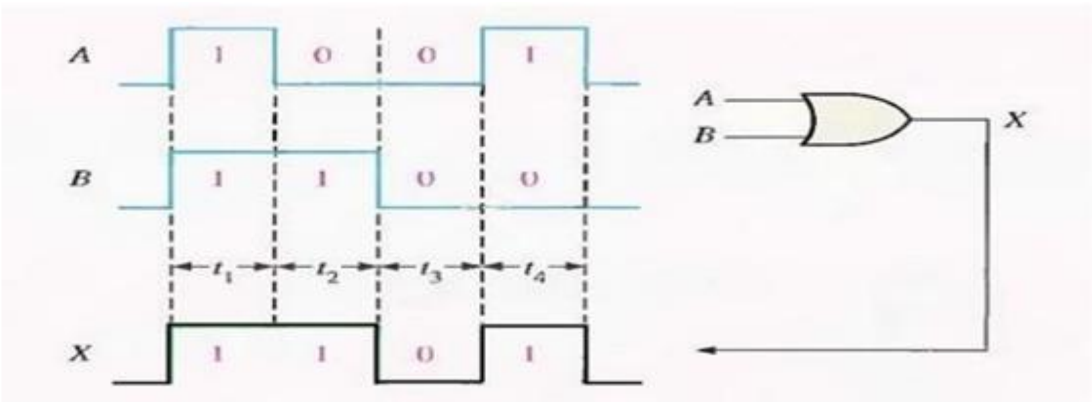
وتقرأ هذه العبارة كالتالي: الخرج Y يساوي A OR B (+ تعني OR).

قواعد الجبر البولى ى بوابةOR:

a- A+1=1
b- A+0=A
c- A+A=A
d- A+A=1

xample2.2 : Write the symbols of OR gate and Boolean expression which has 2 input (A,B). Then find the output X in timing diagram as shown in figure below**.**

Solution:



the timing diagram of AND Gate for 2- inputs

X= A+B

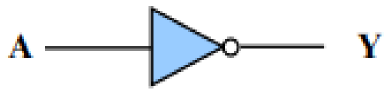
3-Inverter Gate:

العاكس أو بوابة NOT تؤدي عملية يطلق عليها العكس ($Inversion$) أو الإتمام ($Complementation$). والعاكس يغير المستوى المنطقي للدخل إلى عكسه، فإذا كان دخله (1) يغيره فى الخرج إلى (0)، وإذا كان دخله (0) يغيره إلى (1).

بوابة NOT تحتوى على مدخل واحد فقط ومخرج واحد، يوضح الشكل (2.3) الرمز المنطقي القياسي (standard) لبوابةNOT وجدول الحقيقة لها

The truth table for 1- input NOT Gate **(الجدول 2 . 3)**

A	Y
0	1
1	0



Logic symbol for the NOT gate **الشكل (2.3)**

-التعبير البوليني للبوابة NOT هي: $Y = \overline{A}$

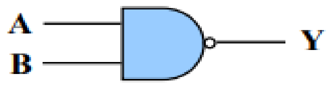
4- NAND Gate

كلمة $(NAND)$ هي اختصار لكلمتي $(NOT\ AND)$ وهي تعني عكس AND، وهذه البوابة يمكن الحصول عليها بتوصيل دخل بوابة العاكس مع خرج البوابة AND كما يبين ذلك شكل(2-4)، كما يبين الشكل الرمز المنطقي لهذه البوابة حيث إنه رمز بوابة AND ولكن مع دائرة صغيرة عند الخرج والتي ترمز إلى بوابة العاكس. جدول(2-4) يوضح جدول الحقيقة للبوابة $NAND$ بمدخلين.

الجدول (2.4)

The truth table for 2- input NAND Gate

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



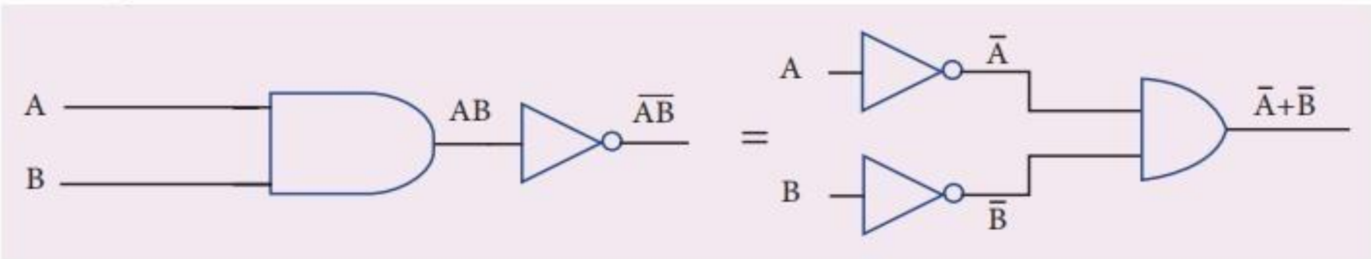
الشكل(2.4)

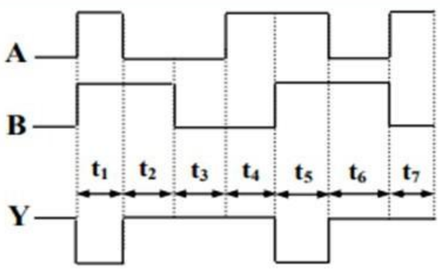
Logic symbol for the NAND gate

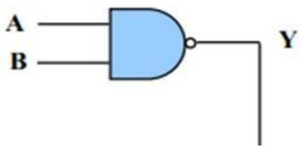
التعبير البوليني لبوابة NAND ذات المدخلين والاخراج الواحد هي:

$$Y = \overline{AB}$$

Example2.3: Write the symbol of **NAND** gate, the truth table and Boolean expression which has 2 input (A, B).







$Y = \overline{A.B}$

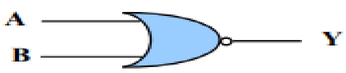
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

5-NOR Gate:

كلمة (NOR) هي أيضاً اختصار لكلمتي (NOT OR) وهي تعني عكس OR، وهذه البوابة يمكن الحصول عليها بتوصيل دخل بوابة العاكس (NOT gate) مع خرج البوابة OR كما هو موضح في شكل(2.5)، ويبين الشكل أيضا الرمز المنطقي للبوابة NOR. وجدول الحقيقة للبوابة NOR بمدخلين موضح في جدول(2.5).

الجدول (2.5)
The truth table for 2- input NOR Gate

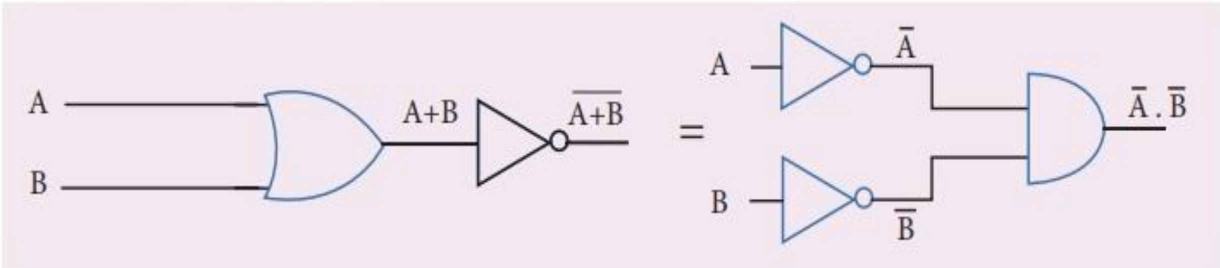
المدخلات		الخرج
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0



الشكل(2.5)
Logic symbol for the NOR gate

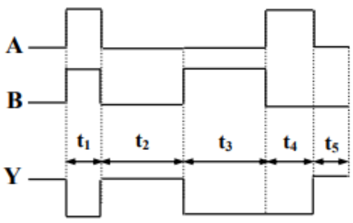
العبارة البولينية لبوابة NOR ذات مدخلين واخراج واحد هي:

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



Example 2.4: Write the symbols of NOR gate, the truth table and Boolean expression which has 2 inputs (A, B).

solution:



المدخلات		الخرج
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$Y = \overline{A+B}$

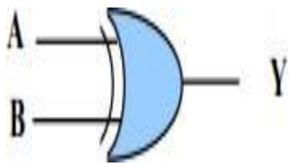
THE EXCLUSIVE-OR AND EXCLUSIVE-NOR GATES

1-The Exclusive-OR Gate

تسمى البوابة OR المنفردة باسم بوابة "أيهما وليس كلاهما" وتختصر إلى XOR-gate، ويوضح شكل(2.6) الرمز المنطقي للبوابة.

(2.6)الجدول The truth table for 2- input **XOR** Gate

INPUTS A	INPUTS B	OUTPUT Y
0	0	0
0	1	1
1	0	1
1	1	0



الشكل (2.6): Logic symbol for the XOR gate

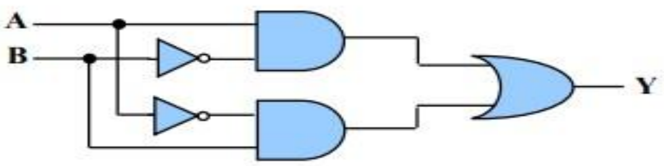
ومن جدول الحقيقة يمكن استنتاج التعبير البوليني لهذه البوابة وهو:

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

والذي يرمز إليه اختصاراً بالتعبير المنطقي:

$$Y = A \oplus B$$

والعلامة $\oplus$ تعني أن A منفردة أو B منفردة. ومن التعبير البوليني السابق للبوابة XOR يمكننا بناء البوابة باستخدام بوابات AND,OR,NOT، وهذا ما يبينه الشكل(**2.7**) حيث تقوم هذه الدائرة المنطقية بوظيفة البوابة XOR المنطقية.



الشكل(2.7) البوابة XOR ممثلة بالبوابات AND,OR,NOT.

2- The Exclusive-NOR Gate

البوابة ***NOR*** المنفردة وتختصر الى ***XNOR-gate*** ويوضح الشكل (**2.8**) الرمز المنطقي للبوابة .

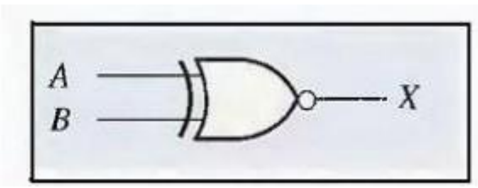
الجدول (2.7)

The truth table for 2- input Exclusive-NOR Gate

INPUTS A	INPUTS B	OUTPUT Y
0	0	1
0	1	0
1	0	0
1	1	1

الشكل (2.8)

Logic symbol for the NAND gate



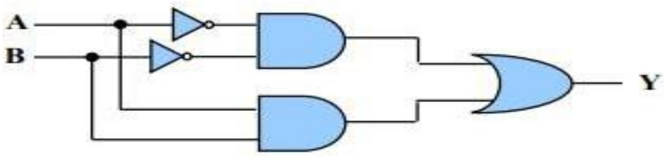
ومن جدول الحقيقة يمكن استنتاج التعبير البوليني لهذه البوابة وهو:

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

والذي يرمز إليه اختصاراً بالتعبير المنطقي:

$$Y = A \odot B$$

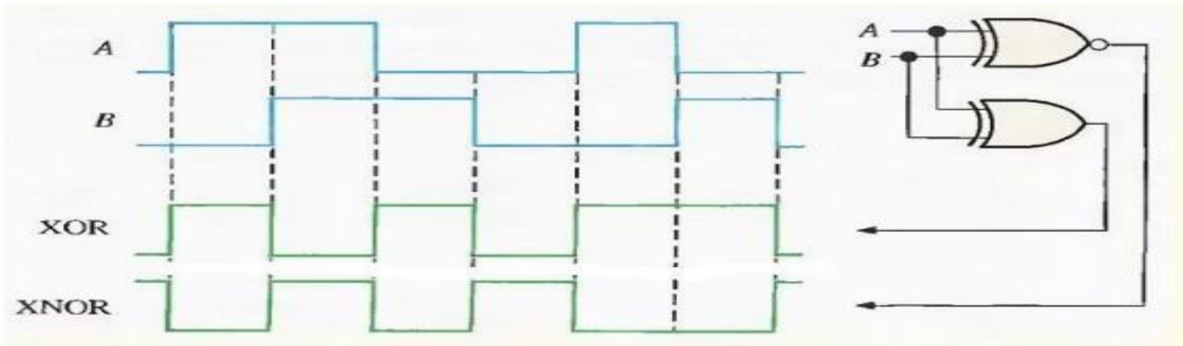
والعلامة $\odot$ تعني علامة التكافؤ. ومن التعبير البوليني السابق للبوابة $XNOR$ يمكننا بناء البوابة باستخدام بوابات AND,OR,NOT، وهذا ما يبينه الشكل(2.9) حيث تقوم هذه الدائرة المنطقية بوظيفة البوابة $XNOR$ المنطقية.



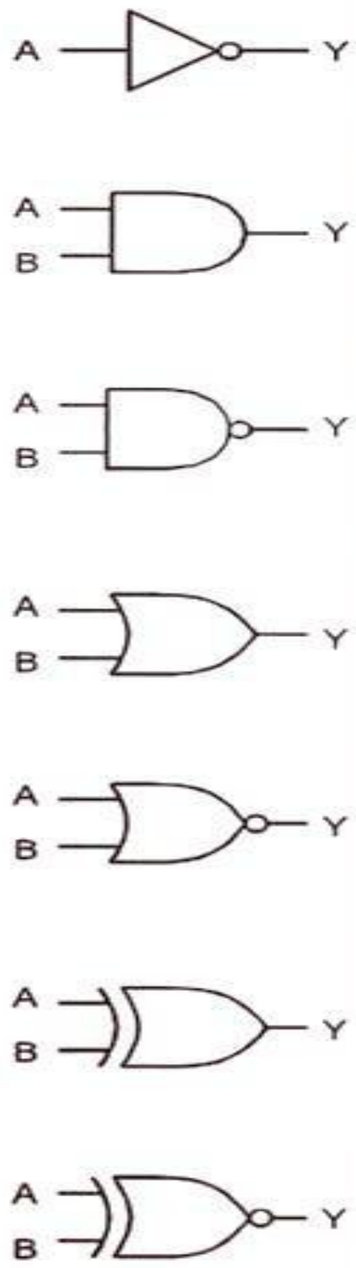
الشكل(2.9) البوابة $XNOR$ ممثلة بالبوابات AND, OR, NOT.

Example2.5: Determine the output waveforms for the XOR gate and for the XNOR gate, given the input waveforms, A and B, in Figure below:

Solution:



Summary



Inverter (NOT gate)

A	Y
0	1
1	0

$Y = \overline{A}$

2-input AND gate

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

$Y = A \bullet B$

2-input NAND gate

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

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

2-input OR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

$Y = A + B$

2-input NOR gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

$Y = \overline{A + B}$

2-input EX-OR gate

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

$Y = A \oplus B$

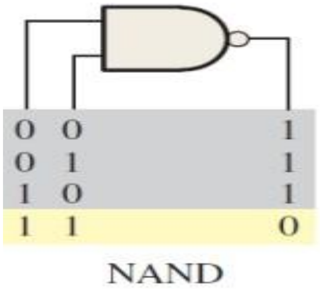
2-input EX-NOR gate

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

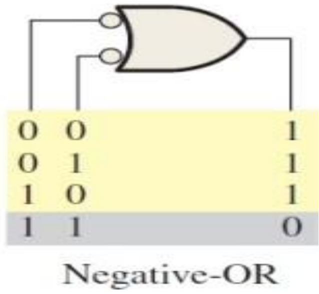
$Y = \overline{A \oplus B}$

الدائرة المكافئة لبعض البوابات

1- … 2- input NAND Gate

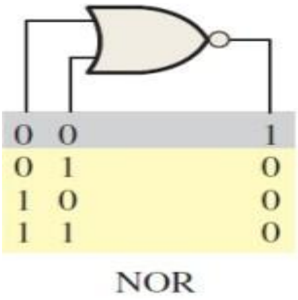


≡

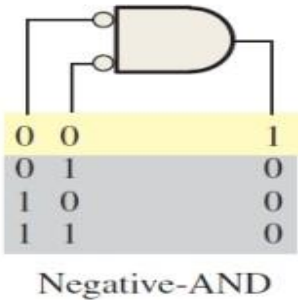


$Y = \overline{A.B}$ $Y = \bar{A}+\bar{B}$

2- … 2- input NOR Gate

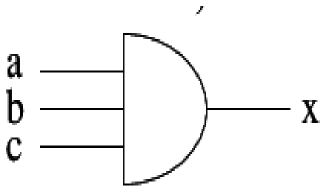


≡

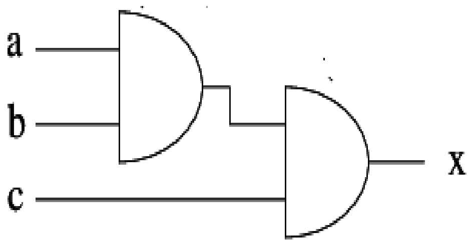


$Y = \overline{A + B}$ $Y = \bar{A}.\bar{B}$

3- … 3-input AND Gate



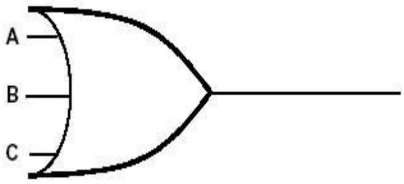
=

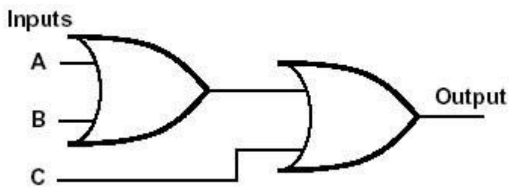


A B C	Y
0 0 0	0
0 0 1	0
0 1 0	0
0 1 1	0
1 0 0	0
1 0 1	0
1 1 0	0
1 1 1	1

X= a.b.c X= (a.b).c

4- … 3-input OR Gate





A B C	Y
0 0 0	0
0 0 1	1
0 1 0	1
0 1 1	1
1 0 0	1
1 0 1	1
1 1 0	1
1 1 1	1

X= a+b+c X= (a+b)+c