اسم المادة	النظام السنوي	الساعات الاسبوعية		
		نظري	عملي	مجموع
تركيب الحاسبة والدوائر الرقمية	(30) اسبوع	2	3	5
لغة التدريس: الإنكليزية				

Objective: Teaching students about basic hardware components, combination of electronic components, foundations of logic circuits in electronic computers, and their operation principle.

week	Theoretical Vocabulary Details
1	Introduction to the electronic computer, Generations of computers, Classification of Computing, Computer Networks.
2	Numerical systems (Decimal, Binary, Octal, Hexadecimal), transfers between numerical systems.
3	Calculations in the Binary system, complements in Binary system, Use of complements in Binary system.
4	Numbers representation in electronic computer, Binary Coded Decimal code (BCD), 4Bit BCD, American Standard Code for Information Interchange code (ASCII), Extended BCD Interchange Code (EBCDIC).
5	Parity checker, Parity bit, - Representation of Integers - Representation of real numbers.
6	Logic gates AND , OR , NOT , NOR , NAND , XNOR , XOR.
7	Laws of Boolean Algebra, De Morgan theorems, Simplifying logical equations.
8	Representation of the different logical circuits using universal gates (NAND and NOR).
9	Methods of extracting logical equations from the truth table: - Sum of product (SOP), - Product of sum (POS).
10-11	Karnaugh map (K-map), Two variables K-map, Three and four variables K-map, Simplification of logical equations using K-map.
12	Combinational logical circuits: half-adder, full-adder, parallel adder circuits.
13	Combinational logical circuits: half subtractor, full subtractor, subtraction using adder logical circuits, subtraction using 1's complement.
14	Decoder logical circuit, Encoder logical circuit.
15	Comparator logical circuit: one bit comparator, two bit comparator.
16	Flip-Flops: General idea, SR Flip-Flop, Timing clock pulses, Synchronous SR Flip-Flop.
17	J-K Flip-Flop, Master/Slave (M / S) JK Flip-Flop, D Flip-Flop, T Flip-Flop.
18	Shift register, Bidirectional shift register.

week	Theoretical Vocabulary Details
19	Counters - Asynchronous counters, Ascending counter, Descending counter, Ascending/ Descending counter.
20	Synchronous counters, Synchronous series counter, Synchronous parallel counters.
21	Computer structure, General definitions - Computer components and their function, Central processing unit (CPU).
22	Memories, Semiconductor memories (EEPROM, EPROM, PROM, ROM), Memory buses.
23	Secondary storage devices (Magnetic tape, reel and cassette, Magnetic disk – Compact disk CD).
24	Input and output devices.
25	Digital to analogue conversion, Digital-to-analog convertor DAC (Flash type).
26	Digital-to-analog convertor DAC (resistors' network type), Accuracy and precision.
27	Analog to digital Conversion ADC using the simultaneous method and comparator method.
28	ADC using ascending counter method.
29	Integrated circuit (IC), General idea, Types of integrated circuits, Difference between them, logic gates IC's, Flip-Flop IC's.
30	Introduction to microcomputers and microprocessors.

week	Practical Vocabulary Details
1-2	General introduction about logic circuits, explain the laboratory board and how to use it, guide students on how to take practical results and the creation of the weekly experimental report.
3	Verification of basic logic gates functioning (AND, OR, NOT)
4	Verification of universal logic gates functioning (NAND, NOR)
5	Building logic circuits using different logic gates.
6	Verification of exclusive logic gates functioning (XOR, XNOR).
7	Design of logic circuits using universal logic gates (NAND, NOR).
8	Verification of Boolean Algebra laws.
9	Verification of De Morgan's laws.
10	Design, building, and verification of the Half Adder logic circuit.
11	Design, building, and verification of the Half Subtractor logic circuit.
21	Design, building, and verification of the Full Adder logic circuit.
13	Design, building, and verification of the Half Subtractor logic circuit.
14	Design, building, and verification of the Decoder logic circuit.
15	Design, building, and verification of the Encoder logic circuit.
16	Design, building, and verification of 1-Bit Digital Comparator logic circuit.
17	Design, building, and verification of 2-Bit Digital Comparator logic circuit.

week	Practical Vocabulary Details
18	Design, building, and verification of S-R Flip Flop logic circuit.
19	Design, building, and verification of J-K Flip Flop logic circuit.
20	Design, building, and verification of D Flip Flop logic circuit.
21	Design, building, and verification of T Flip Flop logic circuit.
22	Design, building, and verification of Master-Slave JK Flip Flop logic circuit.
23	Design, building, and verification of Shift Register circuit.
24	Design, building, and verification of Asynchronous (Ripple) Descending Counter circuit.
25	Design, building, and verification of Ripple Bidirectional Mod-8 Counter circuit.
26	Design, building, and verification of Synchronous Binary Counter circuit.
27	Square wave generation using Flip-Flops.
28	Digital to analogue conversion using Binary Ladder DAC.
29	Analogue to Digital conversion using Counter-type ADC.
30	Introduce integrated circuits (IC's) practically, types of integrated circuits, difference between them, logic gates IC's, Flip-Flop IC's.

الإسبوع الأول

الـفـصـل الأول

الأنـظـمـة الـعـدديـة (Numerical Systems)

1.1 مقـدمـة :

يعد إستخدام الأرقام كوسيلة للعد والحساب من الإنجازات الهامة التي حققها الإنسان عبر التاريخ والتي ساهمت في تسهيل كافة العمليات الحسابية وتسريعها. فقد إستخدام الإنسان منذ القدم الكثير من الأدوات لتمثيل عمليات العد والحساب ومنها استخدامه لأصابع يده العشرة والتي كانت الأساس للنظام العددي والذي لا يزال معمول به حتى يومنا هذا والمسمى **بالنظام العشري** (Decimal System).

في المراحل الدراسية السابقة وعند دراستك للنظام العشري لابد أنك لاحظت أن القيمة الحقيقية للرقم تعتمد على قيمته المكانية في العدد , وهذا يعني أن الرقم يمكن أن يأخذ أكثر من قيمة والذي يحدد ذلك مكانه داخل العدد (والذي يسمى بالمرتبة), تزداد قيمة العدد إذا حركتهُ باتجاه اليسار وتقل قيمته إذا حركيه باتجاه اليمين. فمثلاً العدد (937) نجد أن القيمة الحقيقية للرقم 7 هي سبعة فقط أما قيمة الرقم 3 فهي (30) وقيمة الرقم 9 هي (900).

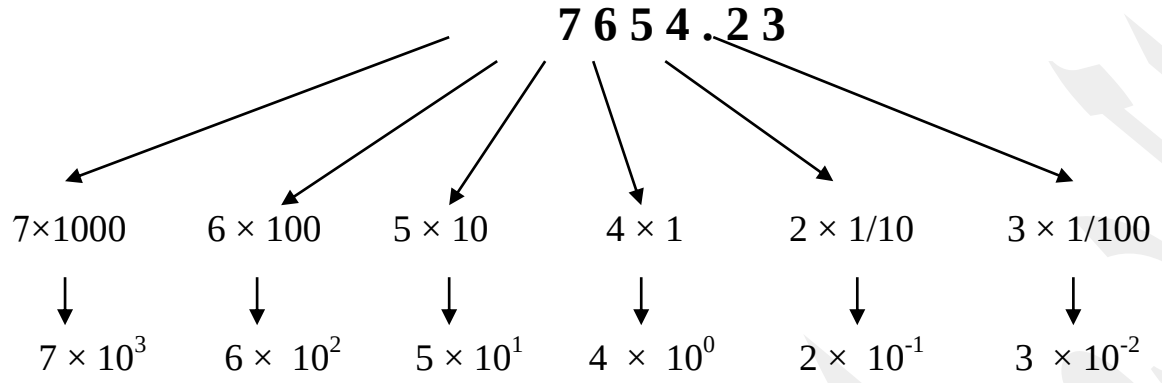
وهنالك أنظمة عددية أخرى غير النظام العشري , وأكثرها شيوعاً هي **النظام الثنائي, النظام الثماني, النظام السادس عشري**. وتكون هذه الأنظمة مفيدة في الأنظمة الرقمية مثل الحاسبات الالكترونية , المعالجات الدقيقة , وغيرها من الأنظمة الرقمية. ولهذا السبب فانه من الضروري الإطلاع على كل من هذه الأنظمة العددية لغرض استخدامها في دراستنا للأنظمة الرقمية.

2.1 الـنـظـام الـعـشـري : Decimal System

وهو النظام العددي المتعارف عليه والمستخدم في كافة المجالات وفي كل انحاء العالم وجاءت تسمية النظام ب(العشري) لان عدد الرموز الداخلة في تركيبة أي عدد في هذا النظام هي عشرة رموز وهي (0 , 1 , 2 , 3 , 4 , 5 , 6 , 7 , 8 , 9) وفي حالة استخدام اكثر من رمز فان القيمة العددية تعتمد على موقع الرمز ضمن سلسلة

الرموز , ان عدد الرموز الداخلة في تركيب النظام العددي تسمى **بأساس النظام** , لذلك فان اساس النظام العشري هو العدد (10) وسمي بأساس العدد لان كل عدد مكتوب بهذا النظام يعتمد بالاساس على هذا العدد .

مثال : العدد العشري 7654.23 يمكن تحليله إلى المراتب التالية



3.1 النظام الثنائي : Binary System

وهو نظام عددي أساسه العدد (2) مقارنة بالنظام العشري الذي أساسه العدد (10) , أي ان عدد الرموز المستخدمة في النظام هي رمزين فقط وهي (0 , 1) لتمثيل كافة الاعداد . ويعتبر النظام الثنائي اساس اللغة التي تتعامل بها الحاسبة الالكترونية والأنظمة الرقمية , مثال على اعداد في هذا النظام :

1001 , 10111.101 , 10.1101 , 0.1011

من خلال ملاحظتنا الاعداد اعلاه نلاحظ بان الاعداد هي بالنظام الثنائي ولكن توجد لها أعداد شبيهه بها في النظام العشري . لتمييز العدد المكتوب بالنظام المعين , يكتب الاعداد المطلوب داخل اقواس مع كتابة رمز اسفل القوس يمثل اساس النظام المكتوب به العدد .

فمثلا : العدد 110 يكتب بالثنائي $(110)_2$ وبالعشري $(110)_{10}$

مثال : لتحليل العدد $(110.101)_2$ الى مراتبه :

$$(110.101)_2 = 0\times2^0 + 1\times2^1 + 1\times2^2 + 1\times2^{-1} + 0\times2^{-2} + 1\times2^{-3}$$

4.1 النظام الثماني : Octal System

وهو من الانظمة المستخدمة في الحاسبات الالكترونية أساسه العدد (8) , الرموز المستخدمة في هذا النظام هي (0 , 1 , 2 , 3 , 4 , 5 , 6 , 7) , ومثال على إعداد النظام الثماني :

$(110.013)_8$, $(203.62)_8$, $(721.5)_8$, $(0.513)_8$

مثال : حلل العدد $(203.65)_8$ الى مراتبه

$$(203.65)_8 = 3 \times 8^0 + 0 \times 8^1 + 2 \times 8^2 + 6 \times 8^{-1} + 5 \times 8^{-2}$$

5.1 النظام السادس عشري : Hexadecimal System

وهو من الانظمة المهمة المستخدمة في الحاسبات الالكترونية أساسه العدد (16) أي إن عدد الرموز المستخدمة في تشكيل أعداد النظام هي 16 رمز وهي :

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

ومثال على أعداد بالنظام السادس عشري :

$(2D6.F3)_{16}$, $(10011.1)_{16}$, $(FFF)_{16}$, $(0.257)_{16}$

مثال :: حلل العدد $(3A1.7F)_{16}$ إلى مراتبه :

$$(3A1.7F)_{16} = 1 \times 16^0 + 10 \times 16^1 + 3 \times 16^2 + 7 \times 16^{-1} + 15 \times 16^{-2}$$

ملاحظة : عند مقارنة الرموز السادس عشرية بالنظام العشري فان الرموز (A ← F) تساوي في النظام العشري (10 ← 15).

6.1 التحويلات بين الأنظمة العددية

أن عملية التحويل بين الأنظمة العددية من العمليات المهمة والتي يجب إن يتعرف عليها الشخص الذي يدرس عملية تصميم الأنظمة الرقمية . ولتسهيل عملية فهم هذه التحويلات سيتم تقسيمها إلى مجاميع كل مجموعة تتشابه بطريقة التحويل .

1.6.1 التحويل من الأنظمة (غير العشرية) إلى النظام العشرى :

لتحويل أي عدد من أي نظام عددي إلى نظام العشري يتم تحليل العدد إلى مراتبه اعتمادا على أساس ذلك النظام ثم إيجاد ناتج جمع الحدود ، والعدد الناتج من الجمع سيكون هو العدد في النظام العشري .

مثال : حول العدد $(1101.01)_2$ إلى النظام العشري :

$$(1101.01)_2 = 1\times 2^0 + 0\times 2^1 + 1\times 2^2 + 1\times 2^3 + 0\times 2^{-1} + 1\times 2^{-2}$$
$$= 1\times 1 + 0\times 2 + 1\times 4 + 1\times 8 + 0\times 1/2 + 1\times 1/4$$
$$= 1 + 0 + 4 + 8 + 0 + 0.25$$
$$= (13.25)_{10}$$

مثال : حول العدد $(125.4)_8$ إلى النظام العشري :

$$(125.4)_8 = 5\times 8^0 + 2\times 8^1 + 1\times 8^2 + 4\times 8^{-1}$$
$$= 5\times 1 + 2\times 8 + 1\times 64 + 4\times 1/8$$
$$= 5 + 16 + 64 + 0.5$$
$$= (85.5)_{10}$$

مثال : حول العدد $(A15.C)_{16}$ إلى النظام العشري :

$$(A15.C)_{16} = 5\times 16^0 + 1\times 16^1 + 10\times 16^2 + 12\times 16^{-1}$$
$$= 5\times 1 + 1\times 16 + 10\times 256 + 12\times 1/16$$
$$= 5 + 16 + 2560 + 0.75$$
$$= (2581.75)_{10}$$

الإسبوع 2

2.6.1 التحويل من النظام العشرى إلى الأنظمة الأخرى :

لتحويل أي عدد عشري إلى أي نظام أخر يجب تجزئته إلى جزء صحيح وجزء كسري وتحويل كل جزء بطريقة خاصة ثم جمع ناتج التحويل للجزئين للحصول على الناتج النهائي .

أولاً: تحويل الجزء الصحيح :

لتحويل الجزء الصحيح للعدد العشري لأي نظام نقوم بتقسيم العدد العشري على أساس النظام المطلوب التحويل إليه ونحتفظ بباقي القسمة ، ثم نأخذ ناتج القسمة ونقسمه مرة أخرى على أساس النظام ونحتفظ بالباقي وهكذا نستمر بتكرار العملية إلى أن نحصل على ناتج قسمة يساوي صفر . فيكون ناتج التحويل في عمود باقي القسمة بقراته من الأسفل إلى الأعلى وكتابته من اليسار إلى اليمين

ثانياً: تحويل الجزء الكسري :

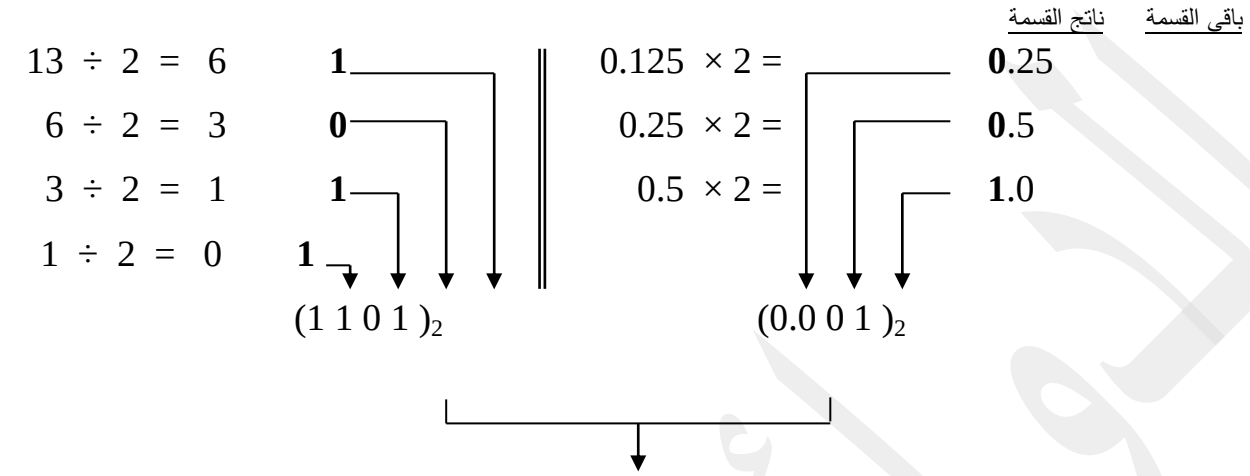
لتحويل الجزء الكسري من العدد العشري إلى نظيره في الأنظمة الأخرى نقوم بضرب العدد الكسري في أساس النظام المطلوب التحويل إليه ثم اخذ الجزء ألكسري فقط من ناتج الضرب وضربه

مرة أخرى في الأساس وهكذا تستمر عملية الضرب إلى أن نتوقف في إحدى الحالات التالية :

- إما أن يكون الجزء ألكسري الناتج في الضرب **يساوي صفر** .
- تكرار الجزء ألكسري أكثر من مرة , (في هذه الحالة يكون ناتج التحويل تقريبي).
- تعقيد الجزء ألكسري أكثر مع استمرار عملية الضرب, (أيضاً في هذه الحالة يكون ناتج التحويل تقريبي) .

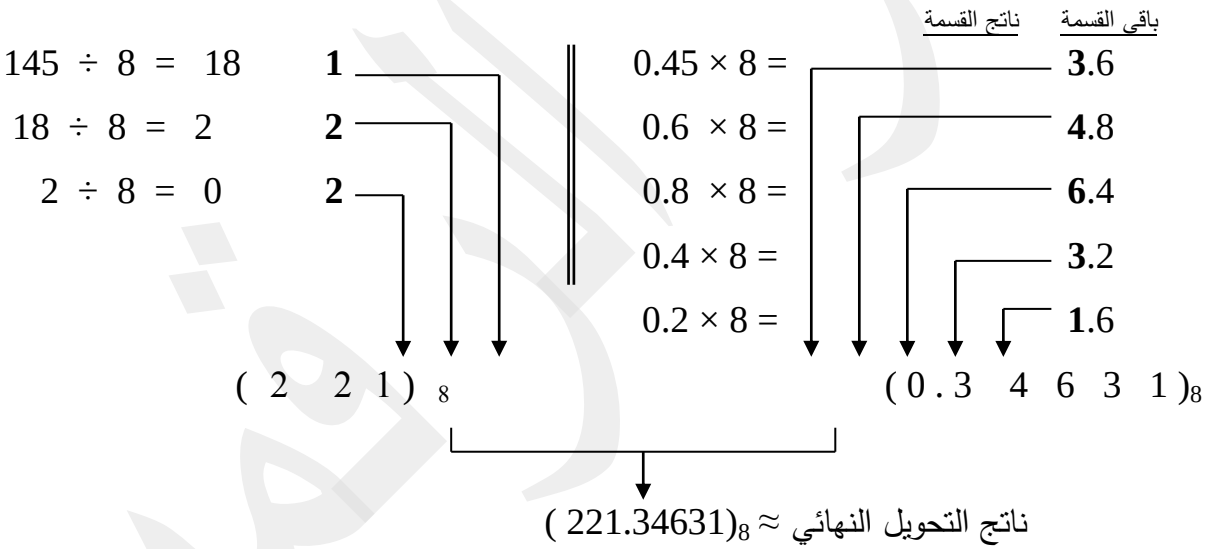
بعد توقف عملية الضرب يتم قراءة ناتج التحويل في عمود الجزء الصحيح من الضرب بقراءته من الأعلى إلى الأسفل وكتابته بعد الفارزة من اليسار إلى اليمين .

مـثـال: حول العدد $(13.125)_{10}$ إلى النظام الثنائي :



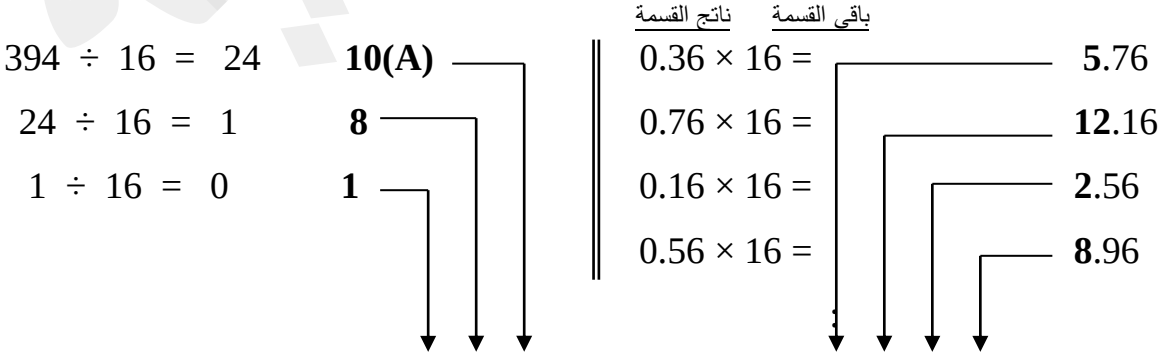
ناتج التحويل النهائي = $(1101.001)_2$

مـثـال: حول العدد $(145.45)_{10}$ إلى النظام الثماني :



ناتج التحويل النهائي $\approx (221.34631)_8$

مـثـال: حول العدد $(394.36)_{10}$ إلى النظام السادس عشري :



$(1\ 8\ A)_{16}$ $(0.5\ C\ 2\ 8)_{16}$

ناتج التحويل النهائي ≈ $(18A.5C28)_{16}$

3.6.1 التحويل من النظام الثنائي إلى الثماني وبالعكس :

لتحويل العدد من النظام الثنائي إلى الثماني يقسم العدد الثنائي إلى مجاميع من ثلاثة مراتب ابتدءا من الفارزة باتجاه اليسار للجزء الصحيح وباتجاه اليمين للجزء ألكسري ,وإذا انتهت الأطراف بمراتب اقل من ثلاثة تكمل باصفار , ثم تحول كل مجموعة ثلاثية في النظام الثنائي إلى ما يقابلها في النظام الثماني كما في الجدول أدناه , والعدد الناتج هو العدد بالنظام الثماني .

الثماني	الثنائي		
	2^2	2^1	2^0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

مـثـال: حول العدد $(11010111.1101)_2$ إلى النظام الثماني :

011 010 111 . 110 100

3 2 7 . 6 4 ←← $(11010111.1101)_2 = (327.64)_8$

ولتحويل أي عدد من النظام الثماني إلى الثنائي فتكون العملية عكسية نسبة للتحويل السابق حيث يحول كل رمز ثماني إلى ما يعادله في النظام الثنائي من ثلاثة رموز وحسب الجدول السابق , ثم نحذف الاصفار التي في الطرف الأيمن من الجزء الكسري والطرف الأيسر من الجزء الصحيح من التحويل إن وجدت والعدد الباقي هو ناتج التحويل .

مـثـال: حول العدد $(150.26)_8$ إلى النظام الثنائي :

1 5 0 . 2 6

001 101 000 . 010 110 ➔➔ $(150.26)_8 = (1101000.01011)_2$

4.6.1 التحويل من النظام الثنائي إلى النظام السادس عشرى وبالعكس :

إن التحويل بين النظام السادس عشري و الثنائي هو شبيه بطريقة التحويل الثنائي والثماني الفرق فقط هو إن المجاميع الثنائية في التحويل هي أربعة مراتب وجدول التحويل هو المبين أدناه

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

مثال : حول العدد $(1111011.10101)_2$ إلى النظام السادس عشري :

0111 1011 . 1010 1000

↓ ↓ ↓ ↓

7 B . A 8 ➔➔ $(1111011.10101)_2 = (\mathbf{7B.A8})_{16}$

مـثـال : حول العدد $(8D.9)_{16}$ إلى النظام الثنائي :

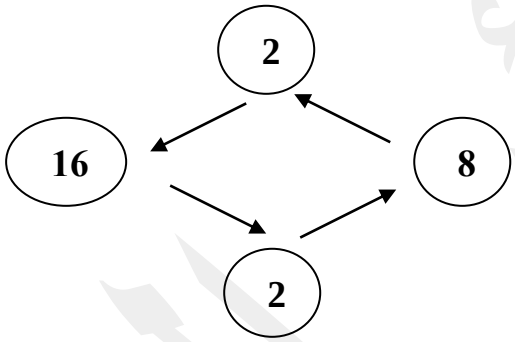
8 D . 9

↓ ↓ ↓

1000 1101 . 1001 ➔ ➔ $(8D.9)_{16} = (\mathbf{10001101.1001})_2$

5.6.1 التحويل من النظام الثماني إلى السادس عشرى وبالعكس :

للتحويل بين النظام الثماني و السادس عشري يتم الاستفادة من التحويلات السابقة لانجاز التحويل النهائي , مثلا إذا أردنا التحويل من الثماني إلى السادس عشري , يتم تحويل الثماني الى الثنائي ومن ثم تحويل الثنائي (الناتج) إلى السادس عشري , والعكس صحيح .



مـثـال : حول العدد $(670.25)_8$ إلى النظام السادس العشري :

6 7 0 . 2 5 الثماني

↓ ↓ ↓ ↓ ↓ ↓

0001 10 11 1 000 . 010 1 0100 الثنائي

↓ ↓ ↓ ↓ ↓

1 B 8 . 5 4 السادس عشري

$(670.25)_8 = (1B8.54)_{16}$

تـمـاريـن :

1. حول العدد $(82.01)_{10}$ إلى النظام الثنائي ؟
2. حول العدد $(540.12)_{10}$ إلى النظام الثماني ؟
3. حول العدد $(260.42)_{10}$ إلى النظام السادس عشري ؟
4. حول العدد $(101101.001)_2$ إلى النظام العشري ؟
5. حول العدد $(17E.2A)_{16}$ إلى النظام الثماني ؟

6. اوجد قيمة X في كل مما يأتي :

$(X)_8 = (35.875)_{10}$, $(X)_{16} = (10001010.101)_2$, $(X)_{10} = (804.1C)_{16}$

الإسبوع الثالث

الفصل الثاني
البوابات المنطقية و الجبر البوليني
(Logic Gates and Boolean Algebra)

يمكن تعريف **البوابة المنطقية** (Logic gate) : بأنها دوائر الكترونية لها إدخال واحد أو أكثر وإخراج واحد . تستخدم لتمثيل بعض الدوال المنطقية التي يمكن صياغتها باستخدام قوانين **الجبر البوليني** (الجبر المنطقي) .

1.2 البوابات الأساسية : (Basic Gates)

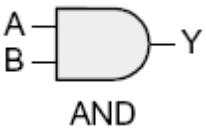
ا. بوابة AND : (AND Gate)

وهي بوابة منطقية لها ادخالين أو أكثر وإخراج واحد ، تعمل على أساس إن إخراج البوابة يكون على منطق (1) عندما تكون جميع الإدخالات على منطق (1) وفي خلاف ذلك يكون الإخراج على منطق (0) .

ملاحظة : منطق (0) ومنطق (1) هي قيم منطقية يمكن إعطائها إلى المتغيرات المنطقية في الحسابات النظرية لكن في الحالة العملية يمكن تمثيلها بشكل فولتيات مثلا :

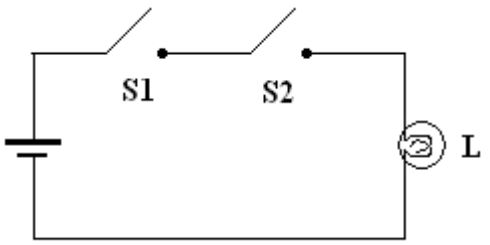
منطق (0) = 0V ، منطق (1) = +5V ، +12V

الرمز المنطقي للبوابة | الصيغة المنطقية | جدول الحقيقة



$Y = A \cdot B$

Inputs		Output
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1



S_1	S_2	L
OFF	OFF	OFF
OFF	ON	OFF
ON	OFF	OFF
ON	ON	ON

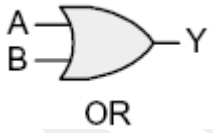
تمثيل بوابة AND بشكل دائرة كهربائية

ب. بوابة OR: (OR Gate)

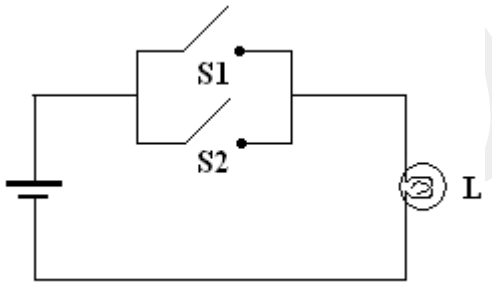
وهي بوابة منطقية لها ادخالين أو أكثر وإخراج واحد ، تعمل على أساس إن إخراج البوابة يكون على منطق (1) عندما يكون احد الإدخالات على الأقل أو جميعها على منطق (1) وبخلاف ذلك يكون الإخراج على منطق (0) .

الرمز المنطقي الصيغة المنطقية جدول الحقيقة

$Y = A + B$



Inputs		Output
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1



S_1	S_2	L
OFF	OFF	OFF
OFF	ON	ON
ON	OFF	ON
ON	ON	ON

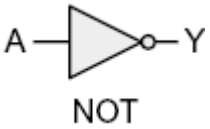
لتمثيل عمل البوابة OR بشكل دائرة كهربائية

جـ. بوابة NOT: (Inverter)

وهي بوابة منطقية لها إدخال واحد وإخراج واحد تعمل على أساس إن منطق الإخراج دائما عكس منطق الإدخال .

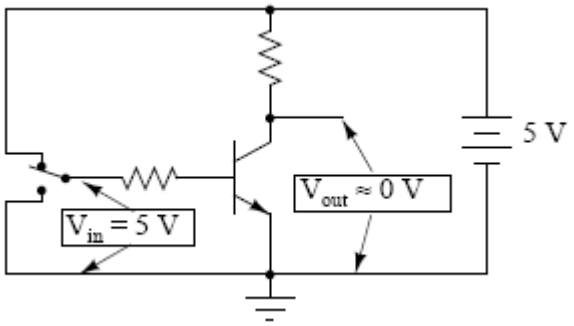
الرمز المنطقي الصيغة المنطقية جدول الحقيقة

$\mathbf{Y = \bar{A}}$



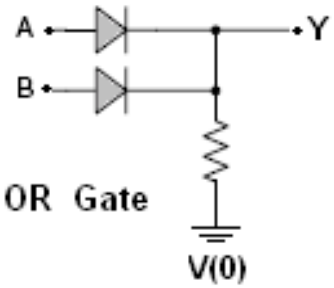
Input	Output
A	Y
0	1
1	0

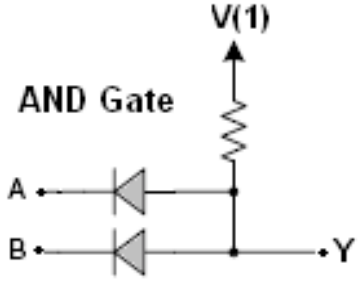
يمكن تمثيل بوابة NOT بإستخدام الترانزستور وكما يلي



V_{in}	V_{out}
0V	5V
5V	0V

ويمكن تمثيل بوابة AND وبوابة OR بإستخدام الدايودات والمقاومات وكما مبين أدناه, حيث إن الفولتية المسلطة على الإدخالات إما فولتية V(0) والتي تمثل منطق 0 أو فولتية V(1) والتي تمثل منطق 1 .





A	B	Y
V(0)	V(0)	V(0)
V(0)	V(1)	V(1)
V(1)	V(0)	V(1)
V(1)	V(1)	V(1)

A	B	Y
V(0)	V(0)	V(0)
V(0)	V(1)	V(0)
V(1)	V(0)	V(0)
V(1)	V(1)	V(1)

2.2 البوابات المركبة :

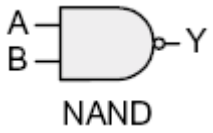
تسمى المجموعة الثانية من البوابات المنطقية بالبوابات المركبة لأنها تعتمد في عملها وتركيبها على البوابات الأساسية , أو لأنها مشتقة من عمل البوابات الأساسية . يشمل هذا النوع من البوابات أربعة بوابات وهي كما يلي:

ا. بوابة NAND : (NAND Gate)

إن تسمية هذه البوابة مشتق من (NOT +AND) وملخص عملها هو عكس عمل بوابة AND.

الرمز المنطقي للبوابة الصيغة المنطقية جدول الحقيقة

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

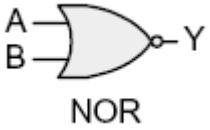


ب. بوابة NOR : (NOR Gate)

إن تسمية هذه البوابة مشتق من (NOT +OR) وملخص عملها هو عكس عمل بوابة OR.

الرمز المنطقي للبوابة الصيغة المنطقية جدول الحقيقة

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

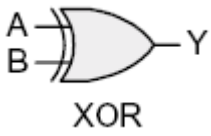


جـ. بوابة XOR : (Exclusive OR Gate)

هذه البوابة تسمى ببوابة OR الاستثنائية ويمكن تلخيص عملها بأنه نفس عمل بوابة OR باستثناء الاحتمال الأخير (عندما تكون الإدخالات 1 , 1) فإن الإخراج يكون على منطق (0) بدل منطق (1). كما يمكن وصف عمل XOR بأن منطق الإخراج يكون (0)عندما تكون الإدخالات متشابهة و(1) عندما تكون الإدخالات مختلفة.

الرمز المنطقي للبوابة الصيغة المنطقية جدول الحقيقة

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



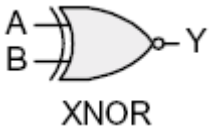
$$Y = A \oplus B$$
$$Y = \bar{A} \cdot B + A \cdot \bar{B}$$

د. بوابة XNOR : (Exclusive OR Gate)

هذه البوابة تسمى ببوابة NOR الاستثنائية ويمكن تلخيص عملها بأنه نفس عمل بوابة NOR باستثناء الاحتمال الأخير (عندما تكون الإدخالات 1 , 1) فإن الإخراج يكون على منطق (1) بدل منطق (0) أو يمكن

القول أن عملها عكس عمل بوابة XOR .

الرمز المنطقي للبوابة الصيغة المنطقية جدول الحقيقة



$$Y = A \odot B$$
$$Y = \overline{A} \cdot \overline{B} + A \cdot B$$
$$Y = \overline{A \oplus B}$$

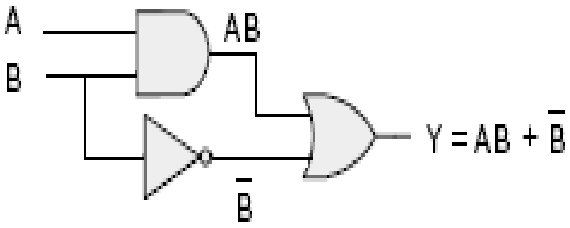
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

الأسبوع الرابع

3.2 الدوائر المنطقية : (Logic Circuits)

الدائرة المنطقية هي عبارة عن مجموعة من البوابات المنطقية المتشابهة أو المختلفة والمرتبطة مع بعضها لتنفيذ فكرة معينة . هذه البوابات يمكن أن تكون أساسية أو مركبة أو خليط منها . ويأتي تصميم هذه الدائرة بعد المرور بمجموعة من الخطوات والتي سنذكر تفاصيلها في المواضيع القادمة ضمن هذا الفصل.

و**مثال** على دائرة منطقية بسيطة مبين في أدناه :

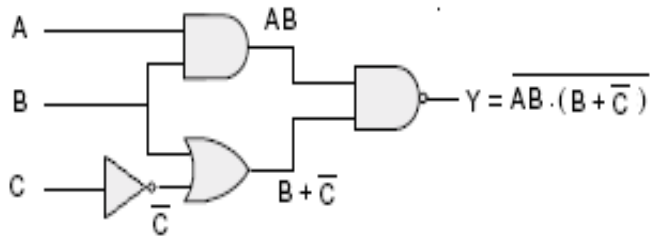


A	B	AB	$\overline{B}$	Y
0	0	0	1	1
0	1	0	0	0
1	0	0	1	1
1	1	1	0	1

يمكن دراسة منطق الإخراج لكل دائرة منطقية بتتبع منطق الإخراج لكل بوابة في الدائرة حتى نصل إلى الإخراج النهائي للدائرة . وهذا يتم بكتابة جدول الحقيقة التفصيلي (جدول المتابعة) للدائرة, وكما مبين في المثال السابق والمثال التالي .

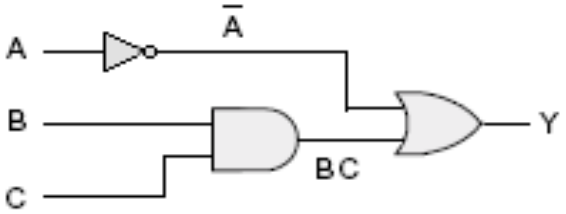
A	B	C	AB	$\overline{C}$	$B+\overline{C}$	Y
0	0	0	0	1	1	1
0	0	1	0	0	0	1
0	1	0	0	1	1	1

0	1	1	0	0	1	1
1	0	0	0	1	1	1
1	0	1	0	0	0	1
1	1	0	1	1	1	0
1	1	1	1	0	1	0



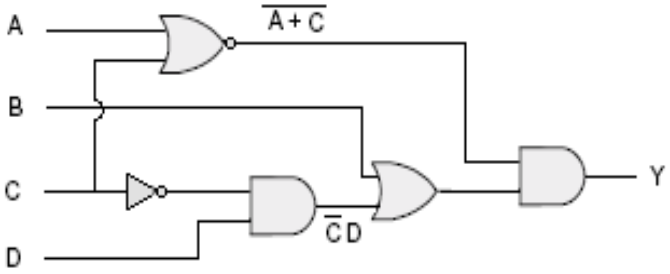
مـثـال إرسم المعادلة المنطقية التالية :

$$\mathbf{Y = \bar{A} + B \cdot C}$$



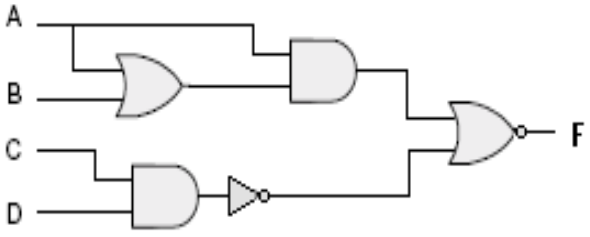
مـثـال إرسم المعادلة المنطقية التالية :

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



تـمـار

1- أكتب المعادلة المنطقية لكل إخراج بوابة للدائرة المنطقية التالية , ثم أكتب جدول الحقيقة التفصيلي للدائرة .



2- إرسم المعادلة المنطقية للصيغ المنطقية التالية :

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

$$R = \overline{XY + Z} \cdot \overline{XY}$$

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

3- بإستخدام جدول الحقيقة برهن إن :

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

الإسبوع الخامس

5.2 قوانين وعلاقات الجبر البوليني (Boolean Algebra)

وضع عالِم الرياضيات الإنكليزي جورج بول في أواسط القرن السابع عشر قواعد لمعالجة الأعداد الثنائية وهذه القواعد سميت **بالجبر البولييني** والتي تعتبر أساس كل الأنظمة الرقمية. ومن الأمور المهمة التي نحتاج فيها تلك القوانين هي عملية تبسيط المعادلات المنطقية وتحويل الصيغ المنطقية من شكل إلى آخر والتي هي خطوة مهمة من خطوات تصميم الدوائر المنطقية حيث يتم فيها اختصار عدد البوابات الداخلة في التصميم.

من أهم علاقات وقوانين الجبر البولييني هي المبينة في أدناه :

علاقة NOT	علاقات OR	علاقات AND
(9) $\overline{\overline{A}} = A$	(5) $A + 0 = A$	(1) $A \cdot 0 = 0$
	(6) $A + 1 = 1$	(2) $A \cdot 1 = A$
	(7) $A + A = A$	(3) $A \cdot A = A$
	(8) $A + \overline{A} = 1$	(4) $A \cdot \overline{A} = 0$

(10) $A.B = B.A$
(11) $A + B = B + A$
} قانوني الإبدال (Commutative laws)

(12) $A.(B.C) = (A.B).C$
(13) $A + (B + C) = (A + B) + C$
} قانوني التجميع (Associative laws)

(14) $A.(B + C) = A.B + A.C$
(15) $A + B.C = (A + B).(A + C)$
} قانوني التوزيع (Distributive laws)

1.5.2 قانون دي موركان (DeMorgan's Law)

أوكستس دي موركان عالم رياضيات إنكليزي آخر في القرن التاسع عشر قام بوضع قانون تحويل منطقي سمي **بقانون دي موركان** والذي كان له منفعة عظيمة في عملية تبسيط وتحوير الصيغ المنطقية. ينص قانون دي موركان على التالي :

قانون دي موركان الأول ..... $\overline{A + B} = \overline{A} \cdot \overline{B}$

قانون دي موركان الثاني ..... $\overline{A \cdot B} = \overline{A} + \overline{B}$

يمكن برهنة أي واحدة من العلاقات والقوانين أعلاه بإستخدام جدول الحقيقة وذلك بأخذ إحتمالات الطرفين.

مـثـال برهن على صحة قانون التوزيع الأول بإستخدام جدول الحقيقة :

A	B	C	B+C	الطرف الأيسر A.(B+C)	A.B	A.C	الطرف الأيمن A.B+A.C
0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	0
1	0	1	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

من الجدول المجاور نلاحظ أن إحتمالات الطرف الأيسر تساوي إحتمالات الطرف الأيمن لكل قيم المتغيرات المنطقية A , B , C .

∴ القانون صحيح.

تـمـار

1- بإستخدام جدول الحقيقة برهن على صحة قانون التوزيع الثاني.

2- بإستخدام جدول الحقيقة برهن على صحة قانون دي موركان.

6.2 تبسيط المعادلات المنطقية بإستخدام قوانين وعلاقات الجبر البوليني

كما ذكرنا سابقا, من الفوائد الأساسية من دراستنا لقوانين وعلاقات الجبر البوليني هي تبسيط المعادلات المنطقية, وهذه العملية مهمة جداً في تصميم الدوائر الرقمية لأنها تؤدي إلى إختصار عدد البوابات المستخدمة في التصميم والذي بدوره يؤدي إلى تقليص حجم الدائرة وتقليل الكلفة المطلوبة.

لا توجد خطوات ثابتة لتبسيط المعادلات المنطقية بهذه الطريقة , والأمثلة أدناه تبين بعض أشكال التبسيط :

مـثـال بسط المعادلة التالية بإستخدام قوانين الجبر البوليني .

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

$$= A.\overline{A} + A.B \qquad \text{(قانون التوزيع 14)}$$

$$= 0 + A.B \qquad \text{(علاقة 4)}$$

$$Y = A.B \qquad \text{(علاقة 5)}$$

نلاحظ أن المعادلة قبل التبسيط تحتاج إلى ثلاثة بوابات , بينما يمكن تحقيق نفس المنطق بالمعادلة بعد التبسيط وببوابة واحدة فقط.

مـثـال بسط المعادلة التالية بإستخدام قوانين الجبر البوليييني .

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

$= A\overline{C}(B + \overline{B}) + A\overline{B}C$ (قانون التوزيع 14)

$= A\overline{C}.1 + A\overline{B}C$ (علاقة 8)

$= A\overline{C} + A\overline{B}C$ (علاقة 2)

$= A(\overline{C} + \overline{B}C)$ (قانون التوزيع 14)

$= A(\overline{C} + \overline{B})(\overline{C} + C)$ (قانون التوزيع 15)

$= A(\overline{C} + \overline{B}).1$ (علاقة 8)

$= A(\overline{C} + \overline{B})$ (علاقة 2)

$Y = A(\overline{C \cdot B})$ (قانون دي موركان الأول)

مـثـال بسط المعادلة التالية بإستخدام قوانين الجبر البوليييني .

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

$= \overline{X + Y} \cdot \overline{\overline{XY + \overline{X}}}$ (دي موركان)

$= \overline{X} \cdot \overline{Y} \cdot (XY + \overline{X})$ (علاقة 16 + 9)

$= \overline{X} \cdot \overline{Y} \cdot XY + \overline{X} \cdot \overline{Y}\,\overline{X}$ (قانون التوزيع 14)

$= \overline{X} \cdot X\ \overline{Y} \cdot Y + \overline{X}\,\overline{X}\,\overline{Y}$ (قانون الإبدال 10)

$= 0.0 + \overline{X}\,\overline{Y}$ (علاقة 4 + 3)

$= 0 + \overline{X}\,\overline{Y}$ (علاقة AND)

$= \overline{X}\,\overline{Y}$ (علاقة 5)

تـمـار

1- بسط المعادلات المنطقية التالية بإستخدام قوانين وعلاقات الجبر البوليييني وقانون دي موركان إلى أبسط صورة ممكنة.

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

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

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

2- بإستخدام قوانين الجبر البوليييني إثبت أن

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

الإسبوع السادس

6.2 تبسيط المعادلات المنطقية بإستخدام قوانين وعلاقات الجبر البوليبيني

كما ذكرنا سابقا, من الفوائد الأساسية من دراستنا لقوانين وعلاقات الجبر البوليبيني هي تبسيط المعادلات المنطقية, وهذه العملية مهمة جداً في تصميم الدوائر الرقمية لأنها تؤدي إلى إختصار عدد البوابات المستخدمة في التصميم والذي بدوره يؤدي إلى تقليص حجم الدائرة وتقليل الكلفة المطلوبة.

لا توجد خطوات ثابتة لتبسيط المعادلات المنطقية بهذه الطريقة , والأمثلة أدناه تبين بعض أشكال التبسيط :

مثال بسط المعادلة التالية بإستخدام قوانين الجبر البوليبيني .

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

$$= A.\overline{A} + A.B \qquad \text{(قانون التوزيع 14)}$$

$$= 0 + A.B \qquad \text{(علاقة 4)}$$

$$Y = A.B \qquad \text{(علاقة 5)}$$

نلاحظ أن المعادلة قبل التبسيط تحتاج إلى ثلاثة بوابات , بينما يمكن تحقيق نفس المنطق بالمعادلة بعد التبسيط وببوابة واحدة فقط.

مثال بسط المعادلة التالية بإستخدام قوانين الجبر البوليبيني .

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

$$= A\overline{C}(B + \overline{B}) + A\overline{B}C \qquad \text{(قانون التوزيع 14)}$$

$$= A\overline{C}.1 + A\overline{B}C \qquad \text{(علاقة 8)}$$

$$= A\overline{C} + A\overline{B}C \qquad \text{(علاقة 2)}$$

$$= A(\overline{C} + \overline{B}C) \qquad \text{(قانون التوزيع 14)}$$

$$= A(\overline{C} + \overline{B})(\overline{C} + C) \qquad \text{(قانون التوزيع 15)}$$

$$= A(\overline{C} + \overline{B}).1 \qquad \text{(علاقة 8)}$$

$$= A(\overline{C} + \overline{B}) \qquad \text{(علاقة 2)}$$

$$Y = A(\overline{C \cdot B}) \qquad \text{(قانون دي موركان الأول)}$$

مثال بسط المعادلة التالية بإستخدام قوانين الجبر البوليبيني .

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

$$= \overline{\overline{X+Y}} \cdot \overline{\overline{\overline{XY+\overline{X}}}} \qquad \text{(دي موركان)}$$

$$= \overline{X} \cdot \overline{Y} \cdot (XY + \overline{X}) \qquad \text{(علاقة 16 + 9)}$$

$= \overline{X}\cdot\overline{Y}\cdot XY + \overline{X}\cdot\overline{Y}\,\overline{X}$　　(قانون التوزيع 14)

$= \overline{X}\cdot X\ \overline{Y}\cdot Y + \overline{X}\,\overline{X}\,\overline{Y}$　　(قانون الإبدال 10)

$= 0 . 0 + \overline{X}\,\overline{Y}$　　(علاقة 4 + 3)

$= 0 + \overline{X}\,\overline{Y}$　　(علاقة AND)

$= \overline{X}\,\overline{Y}$　　(علاقة 5)

تمارين:

1- بسط المعادلات المنطقية التالية بإستخدام قوانين وعلاقات الجبر البوليبيني وقانون دي موركان إلى أبسط صورة ممكنة.

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

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

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

2- بإستخدام قوانين الجبر البوليبيني إثبت أن

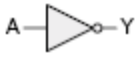
$$(\overline{A} + B)(\overline{B} + \overline{C}) = B\overline{C} + \overline{A}\,\overline{B}$$

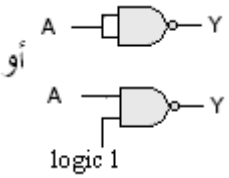
الإسبوع السابع

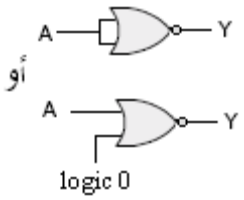
4.2 تصميم الدوائر المنطقية بإستخدام البوابات الشاملة NAND و NOR

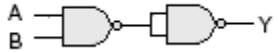
يمكن تحقيق أي معادلة منطقية بإستخدام البوابات الأساسية , كما يمكن تحقيق أي معادلة منطقية بإستخدام بوابات **NAND فقط** أو بوابات **NOR فقط** لذلك سميت هاتين البوابتين بالبوابات الشاملة, يبقى أن نتعرف على كيفية تحويل كل بوابة منطقية إلى ما يكافئها من بوابات NAND أو بوابات NOR وكما مبين في أدناه .

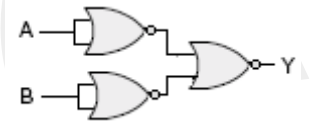
أسم البوابة	الرمز المنطقي	تحقيق البوابة بإستخدام NAND	تحقيق البوابة بإستخدام NOR
NOT			
AND			

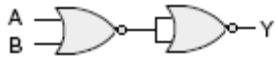
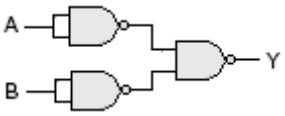






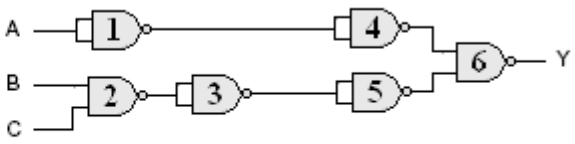




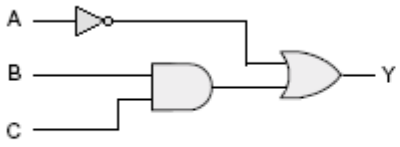
			**OR**

مـثـال حقق المعادلة التالية بإستخدام بوابات NAND فقط وبأقل عدد ممكن من البوابات .

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



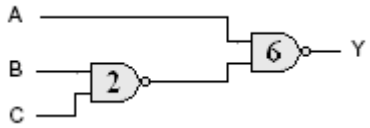
الرسم بإستخدام NAND فقط



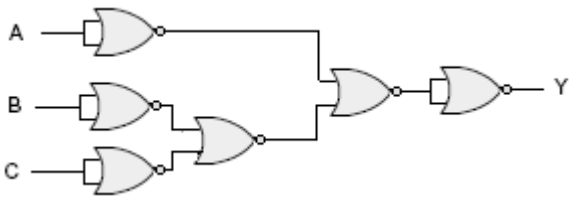
الرسم بإستخدام البوابات الأساسية

مـلاحظة مـهمـة في تصميم الدوائر المنطقية إذا كان هنالك بوابتي NOT متواليتين في التصميم , فيمكن حذفهما والتعويض عنهما بخط توصيل مباشر.

ففي الدائرة أعلاه يمكن ملاحظة البوابتين 1 و4 متوالية وكذلك البوابتين 3 و5 , وعليه يمكن حذف هذه البوابات الأربعة والتعويض بدلاً عنها بخطوط توصيل مباشرة فتصبح الدائرة المختصرة النهائية كما يلي :



مـثـال حقق المعادلة في المثال السابق بإستخدام بوابات NOR فقط وبأقل عدد ممكن من البوابات .



نلاحظ عدم وجود أي اختصار في الرسم فيكون الرسم أعلاه هو الرسم النهائي.

تمارين:

1- حقق كل مما يأتي بإستخدام بوابات NAND فقط وبأقل عدد ممكن من البوابات :

(أ) بوابة NOR (ب) بوابة XOR (جـ) بوابة XNOR

2- حقق كل مما يأتي بإستخدام بوابات NOR فقط وبأقل عدد ممكن من البوابات :

(أ) بوابة NAND (ب) بوابة XOR (جـ) بوابة XNOR

3- بإستخدام بوابات NAND فقط وبأقل عدد ممكن من البوابات حقق المعادلة المنطقية التالية .

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

4- بإستخدام بوابات NOR فقط وبأقل عدد ممكن من البوابات حقق المعادلة المنطقية التالية .

$$F = \overline{AB(\bar{C} + D)} + C$$

الإسبوع الثامن

7.2 كتابة المعادلات المنطقية من جدول الحقيقة

من الخطوات المهمة في عملية تصميم الدوائر الرقمية هي كتابة المعادلة المنطقية من جدول الحقيقة حيث أن جدول الحقيقة يمثل كافة إحتمالات عمل الدائرة والذي تم استنتاجه من فكرة عمل الدائرة.

يمكن كتابة المعادلة المنطقية من جدول الحقيقة بطريقتين طريقة جمع النتاجات وطريقة نتاج المجاميع

1.7.2- طريقة جمع النتاجات Sum of Products (SOP)

في هذه الطريقة يتم الآخذ بنظر الاعتبار إحتمالات الإخراج التي يكون فيها الإخراج على منطق (1) حيث نقوم بكتابة نتاج لكل إحتمال ثم جمعها مع بعضها لتكوين المعادلة المنطقية.

مثال أكتب معادلة Y من جدول الحقيقة التالي بطريقة جمع النتاجات (SOP) :

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

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

$\bar{A} \cdot B$ ←

$A \cdot B$ ←

ملاحظة يمكن كتابة المعادلة أعلاه بصيغة أخرى وهي

$$Y(A,B) = \sum m(1,3)$$

مثال أكتب معادلة Y من جدول الحقيقة التالي بطريقة جمع النتاجات (SOP) :

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

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

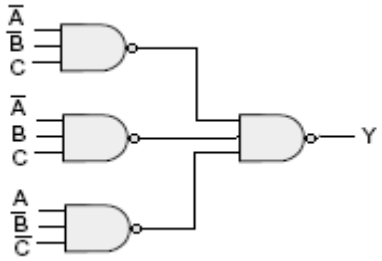
$\bar{A}\bar{B}C$ ←

$\bar{A}BC$ ←

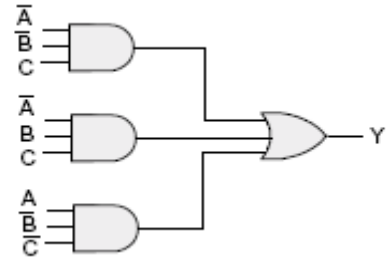
$A\bar{B}\bar{C}$ ←

$$Y(A,B,C) = \sum(1,3,4)$$

نلاحظ في المثالين السابقين أن المعادلة المستخرجة بطريقة جمع النتاجات يمكن رسمها بإستخدام بوابات AND كمستوى أول ثم بوابة OR كمستوى ثاني , أو بإستخدام مستويين من بوابات NAND فقط فمثلا في المعادلة للمثال الأخير يكون الرسم كما يلي :



بإستخدام طريقة NAND - NAND



بإستخدام طريقة AND – OR

2.7.2- طريقة نتاج المجاميع Product of Sums (POS)

في هذه الطريقة يتم الاعتماد على إحتمالات الإخراج التي يكون فيها الإخراج على منطق (0) حيث نقوم بكتابة مجموع لكل إحتمال ثم وضع علاقة AND بين المجاميع المستخرجة لتكوين المعادلة المنطقية النهائية.

مثال أكتب معادلة الإخراج Y من جدول الحقيقة التالي بطريقة نتاج المجاميع (POS) :

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

$(A + B)$

$(\bar{A} + B)$

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

ملاحظة يمكن كتابة المعادلة أعلاه بصيغة أخرى وهي

$$Y(A,B) = \Pi\, M(0,2)$$

مثال أكتب معادلة الإخراج Y من جدول الحقيقة التالي بطريقة نتاج المجاميع (POS) ::

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

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

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

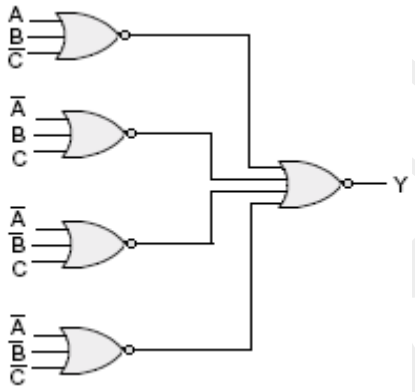
$(\bar{A} + \bar{B} + C)$

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

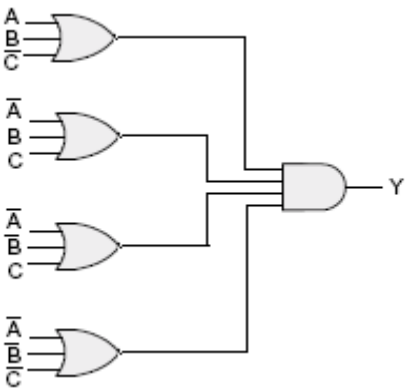
$$Y = (A + B + \bar{C})(\bar{A} + B + C)(\bar{A} + \bar{B} + C)(\bar{A} + \bar{B} + \bar{C})$$

أو بالصيغة $Y(A,B,C) = \Pi M(1,4,6,7)$

ولرسم المعادلات المستخرجة بطريقة نتاج المجاميع(POS) يمكن إستخدام بوابات OR كمستوى أول ثم بوابة ANDكمستوى ثاني , أو بإستخدام مستويين من بوابات NOR فقط فمثلا في المعادلة للمثال الأخير يكون الرسم كما يلي :



بإستخدام طريقة NOR – NOR



بإستخدام طريقة OR – AND

تمارين:

1- أعد كتابة المعادلة التالية بطريقة (SOP) مرة وبطريقة (POS) مرة أخرى .

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

2- إرسم الصيغة التالية بإستخدام بوابات NOR فقط .

$$Y(A,B,C) = \sum m(2,5,6)$$

الإسبوع التاسع

8.2 مخطط كارنو (كارنوف) (Karnaugh Map or K-map)

لقد بيّنا سابقاً الحاجة لتبسيط المعادلات المنطقية وشرحنا الطريقة الجبرية بإستخدام قوانين الجبر البوليبني والتي يكون فيها من الصعوبة التأكد في بعض الأحيان من إمكانية تبسيط المعادلة , وتتطلب أيضاً إلى معرفة كاملة لعلاقات وقوانين الجبر البوليبني. لقد تم اكتشاف طريقة جديدة لتحويل جدول الحقيقة إلى المعادلة المنطقية وتبسيطها من قبل موريك كارنو في عام 1950 والتي سميت لاحقاً **بمخطط كارنو** لإعتمادها طريقة الرسم والتخطيط لتبسيط ومعالجة المعادلات المنطقية.

يتكون مخطط كارنو من عدد من الخلايا (أو المربعات) يعتمد عددها على عدد المتغيرات الداخلة في تركيبة المعادلة المنطقية. فلو فرضنا أن N هو عدد الخلايا و n هو عدد المتغيرات فستكون العلاقة بينهما هي ($N = 2^n$) .

1- مخطط كارنو لمتغيرين :

لو فرضنا أن الإخراج F يتصرف منطقياً كما في جدول الحقيقة التالي :

A	B	F
0	0	0
0	1	1
1	0	0
1	1	1

F	$\bar{A}$	A
$\bar{B}$	0	0
B	1	1

$$F(A,B) = \sum m(1,3)$$

2- مخطط كارنو لثلاثة متغيرات :

لو فرضنا أن الإخراج Y يتصرف منطقياً كما في جدول الحقيقة التالي :

A	B	C	Y
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	1
1	1	1	1

$$Y(A,B,C) = \sum m(3,6,7)$$

Y	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	0	0	1	0
C	0	1	1	0

3- مخطط كارنو لأربعة متغيرات :

لو فرضنا أن الإخراج F يتصرف منطقياً كما في الصيغة المنطقية التالية :

$$F(A,B,C,D) = \sum m(1,6,7,12,15)$$

$1 = 0001 = \bar{A}\bar{B}\bar{C}D$
$6 = 0110 = \bar{A}BC\bar{D}$
$7 = 0111 = \bar{A}BCD$
$12 = 1100 = AB\bar{C}\bar{D}$
$15 = 1111 = ABCD$

F	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}\bar{D}$	0	0	1	0
$\bar{C}D$	1	0	0	0
CD	0	1	1	0
$C\bar{D}$	0	1	0	0

الإسبوع العاشر

1.8.2 تبسيط المعادلات المنطقية بإستخدام مخطط كارنو

لتبسيط المعادلات المنطقية بإستخدام مخطط كارنو نتبع الخطوات التالية :

1- نبحث عن المجاميع (الزوجية أو الرباعية أو الثمانية .....) إن وجدت وإبتداءاً من المجموعة الأكبر, مع الأخذ بنظر الاعتبار خاصية اللف (وهي تصور المخطط وكأنه متصل الأطراف أو بشكل كرة).

2- يُسمح بتداخل المجاميع لغرض إدخال أكثر عدد من الواحدات بدلاً من تركها منفردة .

3- كتابة الصيغ المنطقية للمجاميع مع ترك أي مجموعة اشتركت جميع عناصرها مع مجاميع أخرى .

4- كتابة المعادلة المنطقية النهائية المبسطة بإستخدام طريقة جمع النتاجات.

مـثـال أكتب المعادلة المبسطة لـ Y من الصيغة التالية بإستخدام مخطط كارنو:

$$Y(A,B,C) = \sum m(0,1,2,3,5)$$

000	001	010	011	101
$\bar{A}\bar{B}\bar{C}$	$\bar{A}\bar{B}C$	$\bar{A}B\bar{C}$	$\bar{A}BC$	$A\bar{B}C$

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

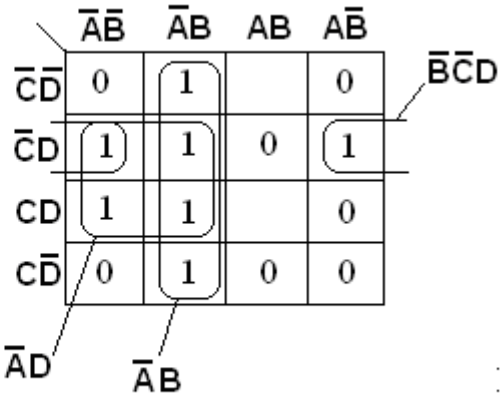
Y	$\bar{A}\bar{B}$	$\bar{A}B$	AB	$A\bar{B}$
$\bar{C}$	1	1	0	0
C	1	1	0	1

$\bar{A}$ $\bar{B}C$

مـثـال أكتب المعادلة المبسطة لـ F من الصيغة التالية بإستخدام مخطط كارنو:

$$F(A,B,C,D) = \sum m(1,3,4,5,6,7,9)$$

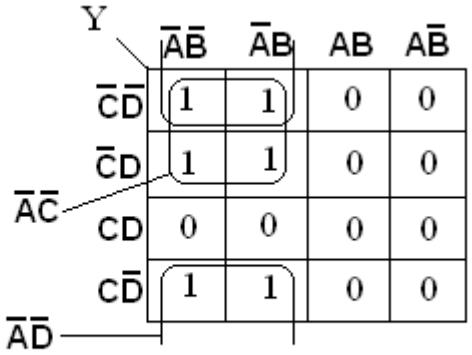
$1 = 0001 = \bar{A}\bar{B}\bar{C}D$
$3 = 0011 = \bar{A}\bar{B}CD$
$4 = 0100 = \bar{A}B\bar{C}\bar{D}$
$5 = 0101 = \bar{A}B\bar{C}D$
$6 = 0110 = \bar{A}BC\bar{D}$
$7 = 0111 = \bar{A}BCD$
$9 = 1001 = A\bar{B}\bar{C}D$



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

مـثـال بسط المعادلة التالية بإستخدام مخطط كارنو .

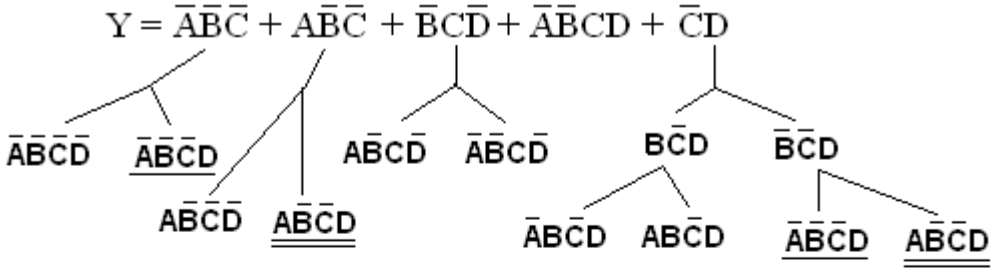
$$Y = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + \bar{A}B\bar{C}D + \bar{A}BC\bar{D}$$

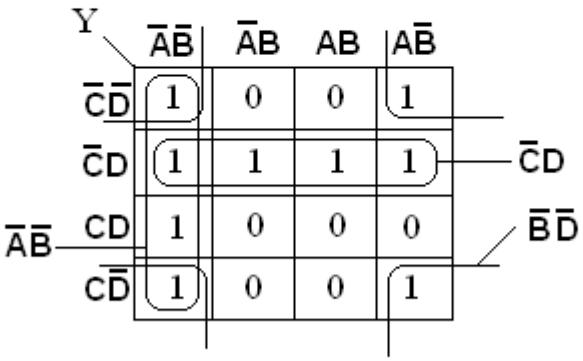


$$Y = \bar{A}\bar{C} + \bar{A}\bar{D}$$

مـثـال بسط المعادلة التالية بإستخدام مخطط كارنو .

$$Y = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{C}D$$



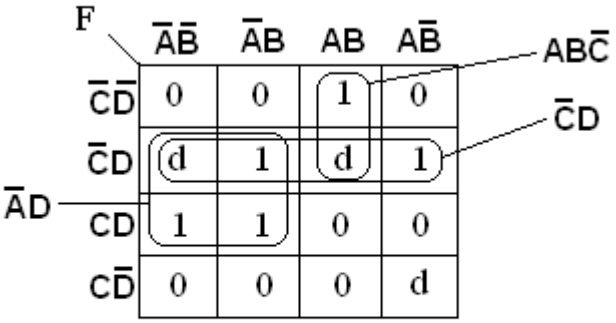


$$Y = \bar{A}\bar{B} + \bar{C}D + \bar{B}\bar{D}$$

مـثـال أكتب المعادلة المبسطة لـ F من الصيغة التالية بإستخدام مخطط كارنو:

$$F(A,B,C,D) = \sum m(3,5,7,9,12) + d(1,10,13)$$

مـلاحظة تعني حالات d حالات (don't care) أو (لا تهتم) وهي إحتمالات لا تمر بها الدالة F فلذلك يمكن إعتبارها (0) أو (1) حسب حاجتنا أليها في المخطط بحيث تعطينا تبسيط أكثر.



$$F = \bar{A}D + \bar{C}D + AB\bar{C}$$

تمارين:

1- أكتب المعادلة المنطقية المبسطة لمخططات كارنو التالية بعد إضافة المتغيرات :

0	0	1	1
0	1	1	0

1	1	0	1
1	1	0	1
1	0	0	1
1	1	0	1

0	1	0	0
0	1	1	1
1	1	1	0
0	0	1	0

1	0	0	1
0	1	0	0
0	0	0	0
1	0	0	1

2- أكتب المعادلة المبسطة لـ Y لكل من الصيغ المنطقية التالية :

$$Y(A,B,C,D) = \sum m(0,5,6,7,11,12,13,15) \quad , \quad Y(A,B,C) = \Pi(0,4,6,7)$$

3- بسط المعادلة التالية بإستخدام مخطط كارنو :

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

4- أكتب المعادلة المنطقية المبسطة لـ X من الصيغة التالية :

$$X(A,B,C,D) = \sum m(3,6,7,10,11,13,15) + d(2,4,8,14)$$

الإسبوع الحادي عشر

7.1 العمليات الحسابية في النظام الثنائي

كلنا تعلّم العمليات الحسابية التي تتم باستخدام الأعداد العشرية مثل الجمع والطرح والضرب والقسمة في المراحل الدراسية السابقة ، كل هذه العمليات يمكن اجرائها أيضاً في الأنظمة العددية الأخرى ، **ولأهمية النظام الثنائي** في دراستنا لموضوع الدوائر الرقمية , فسنقوم بدراسة تلك العمليات الحسابية في النظام الثنائي .

1 . 7 . 1 الجمع في النظام الثنائي : Binary Addition

إن ابسط عملية جمع في النظام الثنائي هي التي تتم بين عددين كل عدد يتكون من رمز (مرتبة) ثنائي واحد . ولو أخذنا كافة الاحتمالات لهذه العملية فستكون الاحتمالات المبينة في أدناه . وبالاعتماد على هذه الاحتمالات يمكن تنفيذ أي عملية جمع ثنائية لأي عدد من المراتب.

0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 ⟶ 1 محمل (Carry)

مثال : . اجمع العددين $(11010.1)_2$, $(1011.01)_2$:

$$\begin{array}{r} 1\,1\,0\,1\,0\,.\,1\,0 \\ 0\,1\,0\,1\,1\,.\,0\,1 \quad + \\ \hline 1\,0\,0\,1\,0\,1\,.\,1\,1 \end{array}$$

مثال : ما ناتج جمع العددين $(11011.101)_2$, $(1110.11)_2$:

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

ملاحظة : ناتج جمع 1 + 1 + 1 = 1 ⟵ 1 محمل

2.7.1 الطرح في النظام الثنائي(الطريقة المباشرة) : Binary Subtraction

كما في عملية الجمع , تكون احتمالات ابسط عملية طرح بين عددين ثنائيين , وهي أربع احتمالات, وكما مبينة :

0 - 0 = 0
0 - 1 = 1 ⟶ **1** استعارة(Borrow)
1 - 0 = 1
1 - 1 = 0

مثال : اطرح العدد $(1011)_2$ من العدد $(1101.1)_2$:

```
  1 1 0 1 . 1
  1 0 1 1 . 0 -
---------------
  0 0 1 0 . 1
```

تمرين / اطرح العدد $(110.1)_2$ من العدد $(1000.01)_2$.

3.7.1 الضرب في النظام الثنائي : Binary Multiplication

إن احتمالات عملية الضرب في النظام الثنائي هي :

0 × 0 = 0
0 × 1 = 0
1 × 0 = 0
1 × 1 = 1

مثال : اوجد ناتج ضرب العددين $(101)_2$, $(1010)_2$:

```
   1 0 1 0
     1 0 1  ×
  ---------
   1 0 1 0
 0 0 0 0
1 0 1 0
-----------
1 1 0 0 1 0
```

4.7.1 القسمة في النظام الثنائي : Binary Division

إن احتمالات عملية القسمة في النظام الثنائي هي :

$$0 \div 0 = ?$$

$$0 \div 1 = 0$$

$$1 \div 0 = ?$$

$$1 \div 1 = 1$$

مثال: اوجد ناتج قسمة العدد $(11000)_2$ على العدد $(100)_2$

```
            110
        ---------
      100 | 11000
            100
            ----
            0100
             100
            ----
            0000
```

8.1 المتممات Complements

يستخدم مفهوم المتممات في الحاسبة في خزن الاعداد السالبة وسنبين ذلك في المواضيع القادمة، والاستخدام الثاني هو للتعويض عن عملية الطرح بعملية جمع متكرر والذي يؤدي بدوره إلى جعل الدوائر الالكترونية المسؤولة عن عملية الجمع بتنفيذ عملية الطرح مع بعض الإضافات للدائرة .

1.8.1 المتممات في النظام الثنائي :

هنالك نوعان من المتممات في النظام الثنائي .

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

2. **المتمم لـ2** (2's Complement) : هو المتمم لـ 1 مضافا إليه 1 .

مـثـال : العدد	المتتم لـ 1	المتتم لـ 2
110111	001000	001001
10010	01101	01110

2.8.1 الطرح الثنائي باستخدام المتممات :

أولا . الطرح باستخدام المتمم لـ 1 :

لطرح عددين ثنائيين باستخدام المتمم **لـ 1** نتبع الخطوات التالية :

1. إكمال مراتب العدد الأقل عددا بالمراتب (المطروح أو المطروح منه) .
2. إيجاد المتمم **لـ 1** للعدد المطروح .
3. جمع المتمم **لـ 1** للمطروح مع المطروح منه .
4. نلاحظ نتيجة الجمع للخطوة 3 وكما يلي :

أ. إذا كان هنالك واحد ظاهر في المرتبة الإضافية ، فنقوم بجمعه مع بقية العدد والناتج من عملية الجمع هو ناتج الطرح ويكون موجب .

ب. إذا لم يظهر واحد في المرتبة الإضافية (وهو دلالة إن ناتج الطرح سالب) ويكون ناتج الطرح بأخذ المتمم **لـ 1** لناتج الجمع للخطوة 3 ويكون ناتج العملية هو ناتج الطرح ويكون سالب.

مـثـال : اطرح العدد $(110)_2$ من العدد $(1010)_2$ باستخدام طريقة المتمم **لـ 1** :

	1 0 1 0	المطروح منه
	− 1 1 0	المطروح
الخطوة 1	0 1 1 0	تكملة مراتب المطروح
الخطوة 2	1 0 0 1	المتمم **لـ 1** للمطروح

الخطوة 3

1 0 0 1 المتمم لـ **1** للمطروح

\+ 1 0 1 0 المطروح منه

الخطوة 4

1 0 0 1 1 ← المرتبة الإضافية

\+ 1 ←

0 1 0 0 ← ناتج الطرح

مـثـال : اطرح العدد $(10101)_2$ من العدد $(1011)_2$ باستخدام المتمم لـ **1** :

0 1 0 1 1 المطروح منه

— 1 0 1 0 1 المطروح

0 1 0 1 0 المتمم لـ **1** للمطروح

\+ 0 1 0 1 1 المطروح منه

1 0 1 0 1 ? ← المرتبة الإضافية خالية إذن النتيجة سالبة

— **0 1 0 1 0** ناتج الطرح

ثانياً. الطرح باستخدام المتمم لـ 2 :

لطرح عدديين ثنائيين باستخدام المتمم لـ **2** تتبع الخطوات التالية :

1. إكمال مراتب العدد الأقل مراتب .
2. إيجاد المتمم لـ **2** للعدد المطروح .
3. جمع المتمم لـ **2** للعدد المطروح مع المطروح منه .
4. نلاحظ نتيجة الجمع للخطوة 3 :

أ. إذا كان هنالك واحد ظاهر في المرتبة الإضافية ، فنقوم بحذف هذا الواحد والباقي هو ناتج الطرح (موجب) .

ب. إذا لم يظهر واحد في المرتبة الإضافية ، فنقوم بأخذ المتمم لـ **2** لناتج الجمع ويكون هو ناتج الطرح (سالب)

مـثـال : اطرح العدد $(110)_2$ من العدد $(1110)_2$ باستخدام المتمم لـ **2** :

1110 المطروح منه

0110 − المطروح

1001 المتمم لـ **1** للمطروح

1 +

1010 المتمم لـ **2** للمطروح

1110 + المطروح منه

1 **1000** ← 1 المرتبة الإضافية تحذف

ناتج الطرح (موجب)

مـثـال : اطرح العدد $(11101)_2$ من العدد $(1011)_2$ باستخدام المتمم لـ **2** :

01011 المطروح منه

11101 − المطروح

00010 المتمم لـ **1** للمطروح

1 +

00011 المتمم لـ **2** للمطروح

01011 + المطروح منه

01110 ? ← المرتبة الإضافية خالية إذن النتيجة سالبة

ونتيجة الطرح هو بأخذ **المتمم لـ 2** لآخر نتيجة

10001

1 +

1 0010

إذن ناتج الطرح هو العدد (**10010 −**)

الإسبوع الثاني عشر

الفصل الثالث
تطبيقات على دوائر المنطق الإئتلافي
Applications on Combinational Logic Circuits

1.3 مقدمة :

ناقشنا في الفصل الثاني أهم البوابات المنطقية وكيفية كتابة الصيغ المنطقية لإخراج البوابة أو إخراج الدائرة المنطقية , أيضا بينّا أهم الخطوات المطلوبة لتصميم أي دائرة منطقية والتي يجب أن تكون بأقل عدد ممكن من البوابات المنطقية لتقليل حجم الدائرة وتقليل القدرة المطلوبة لتشغيل الدائرة.

لذلك يمكن تلخيص أهم الخطوات لتصميم أي دائرة أو منظومة منطقية كما يلي :

- استيعاب فكرة عمل الدائرة وتحديد إدخالات وإخراجات الدائرة.
- تحويل فكرة عمل الدائرة الى جدول حقيقة يحوي كافة احتمالات عمل الدائرة.
- كتابة الصيغ(المعادلات)المنطقية لكل إخراج بطريقة جمع النتاجات (SOP)أو نتاج المجاميع (POS).
- تبسيط المعادلات المستخرجة باستخدام علاقات وقوانين الجبر المنطقي أو بطريقة مخطط كارنو.
- رسم المعادلات المبسطة باستخدام البوابات المتوفرة .
- بناء الدائرة عملياً حسب الرسم المستخرج.

تقسم الدوائر الرقمية الى صنفين رئيسيين :

1- الدوائر الائتلافية Combinational Circuits .

2- الدوائر المتعاقبة Sequential Circuits .

بالنسبة للدوائر الائتلافية يعتمد الإخراج في أي وقت على وجود الإدخالات في نفس الوقت ويعني ذلك عدم وجود ذاكرة في هذه الدوائر. أما الدوائر المتعاقبة فيعتمد فيها الإخراج على الإدخالات الموجودة في نفس الوقت إضافة إلى الإدخلات والإخراجات التي كانت سابقاً ويعني ذلك وجود عناصر تستخدم لخزن المعاومات السابقة تسمى هذه العناصر بالذاكرة.

في هذا الفصل سيتم تناول خطوات تصميم بعض الدوائر الإئتلافية المستخدمة في الدوائرالمنطقية العملية مثل الدوائر الحسابية ودائرة مفك الشفرة ودائرة الترميز ودائرة المقارن الرقمي , أما الدوائر المتعاقبة فسيتم تناولها في الفصول القادمة.

2.3 الدوائر الحسابية الرقمية : (Arithmetic Logic Circuits)

1.2.3 دوائر الجمع الثنائية : (Binary Adders)

وهي الدوائر التي تقوم بعملية الجمع بين عددين ثنائيين وهذه الدوائر هي :

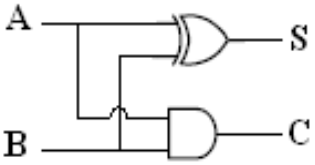
أ- دائرة نصف الجامع : (Half Adder)

هي دائرة تقوم بجمع رقمين ثنائيين كل منهما يتكون من رمز ثنائي واحد (1-bit) . الإدخالات لهذه الدائرة هي العددين الثنائيين (A) و (B) ولها إخراجين وهي الإخراج (S) الذي يمثل ناتج الجمع "Sum" والإخراج (C) الذي يمثل المحمل من عملية الجمع "Carry" .

INPUTS		OUTPUTS	
A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = \bar{A}B + A\bar{B}$$
$$= A \oplus B$$
$$C = AB$$

الرمز المنطقي



الدائرة المنطقية

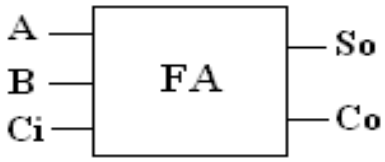
تمارين:

1- مثل دائرة نصف الجامع باستخدام البوابات الأساسية.
2- مثل دائرة نصف الجامع بإستخدام بوابات NAND فقط .
3- مثل دائرة نصف الجامع بإستخدام بوابات NOR فقط .

ب- دائرة الجامع التام : (Full Adder)

بالنظر لكون دائرة نصف الجامع تنجز عملية الجمع لرقمين ثنائيين فقط ولا تأخذ بنظر الاعتبار المحمل من عملية الجمع للمرتبة السابقة ولأجل إنجاز عملية جمع تامة يجب تصميم دائرة جمع تقوم بجمع المحمل من المرتبة السابقة مع عملية الجمع للمرتبة الجديدة , وتسمى هذه الدائرة بدائرة **الجامع التام** والمبينة خطوات تصميمها كما يلي :

INPUTS			OUTPUTS	
A	B	C_i	S_O	C_O
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1



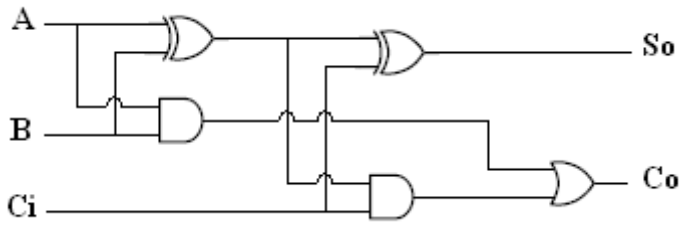
الرمز المنطقي

من جدول الحقيقة :

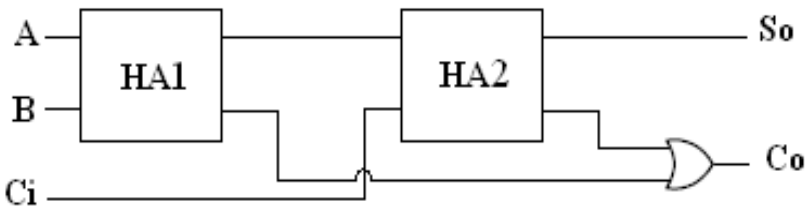
$$S_o = \bar{A}\bar{B}C_i + \bar{A}B\bar{C_i} + A\bar{B}\bar{C_i} + ABC_i$$
$$= \bar{C_i}(\bar{A}\,B + A\,\bar{B}) + C_i(\bar{A}\,\bar{B} + A\,B)$$
$$= \bar{C_i}(A \oplus B) + C_i\overline{(A \oplus B)}$$
$$\mathbf{S_o = (A \oplus B) \oplus C_i} \quad \text{..... (1)}$$

$$C_o = \bar{A}BC_i + A\bar{B}C_i + AB\bar{C_i} + ABC_i$$
$$= C_i(\bar{A}\,B + A\,\bar{B}) + AB(\bar{C_i} + C_i)$$
$$\mathbf{C_o = C_i(A \oplus B) + AB} \quad \text{..... (2)}$$

يمكن إستخدام معادلة (1) و معادلة (2) لرسم الدائرة المنطقية للجامع التام :

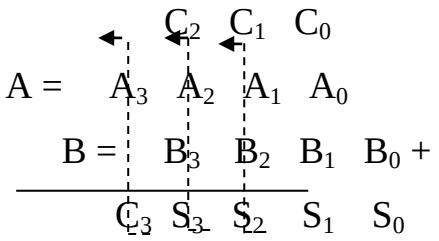


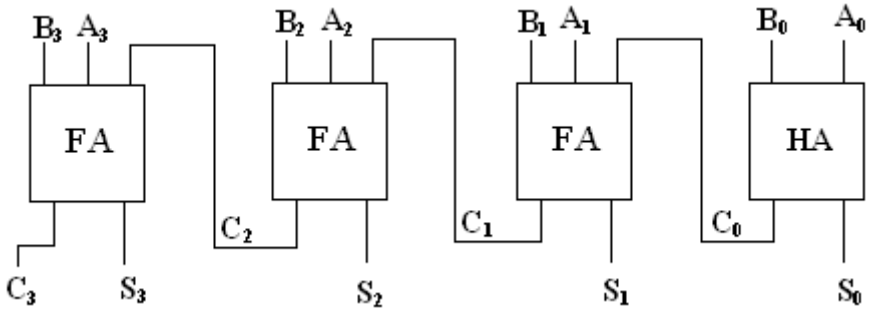
وكذلك يمكن بناء دائرة الجامع التام بإستخدام دائرتين نصف جامع وكما يلي :



جـ – **دائرة الجمع المتوازية : (Parallel Adder)**

لجمع عددين كل عدد يتكون من أكثر من مرتبة ثنائية سوف نحتاج الى سلسلة من دوائر الجمع , دائرة جمع لكل مرتبة هذه الدائرة تسمى بدائرة الجمع المتوازية. فمثلاً لتصميم دائرة لجمع العددين (A) و(B) كل منهما يتكون من أربعة مراتب هو كما يلي :





ألأسبوع الثالث عشر

2.2.3 دوائر الطرح الثنائية : (Binary Subtractors)

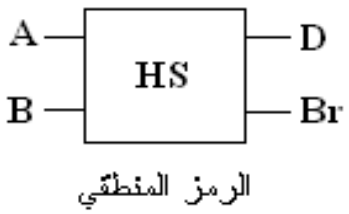
وهي الدوائر التي تقوم بعملية الطرح بين عددين ثنائيين , وطريقة تصميم هذه الدوائر هي طريقة مشابهة لطريقة تصميم دوائر الجمع السابقة, وهذه الدوائر هي :

أ– دائرة نصف الطارح : (Half Subtractor)

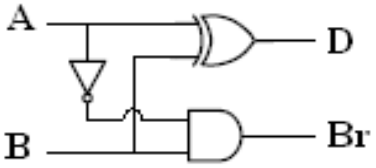
هي دائرة تقوم بطرح رقمين ثنائيين كل منهما يتكون من رمز ثنائي واحد (1-bit) . الإدخالات لهذه الدائرة هي العددين الثنائيين (A) و(B) ولها إخراجين وهي الإخراج (D) الذي يمثل ناتج الطرح "Difference" والإخراج (Br) الذي يمثل عملية الإستعارة "Borrow" .

INPUTS		OUTPUTS	
A	**B**	**D**	**Br**
0	**0**	**0**	**0**
0	**1**	**1**	**1**
1	**0**	**1**	**0**
1	**1**	**0**	**0**

$$D = \bar{A}B + A\bar{B}$$
$$= A \oplus B$$
$$Br = \bar{A}B$$



الرمز المنطقي



الدائرة المنطقية

تمارين:

1- مثل دائرة نصف الطارح بإستخدام البوابات الأساسية.

2- مثل دائرة نصف الطارح بإستخدام بوابات NAND فقط .

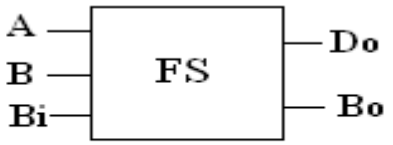
3- مثل دائرة نصف الطارح بإستخدام بوابات NOR فقط .

4- مثل دائرة نصف الطارح بإستخدام بوابة XOR وبوابة AND فقط .

ب- **دائرة الطارح التام : (Full Subtractor)**

نحتاج لدائرة الطارح التام مثل ماهو في دائرة الجامع التام , لغرض إجراء عملية الطرح للأرقام الثنائية المتعددة المراتب , إذ يمكن وجود إستعارة من الرقم الثنائي في المرتبة السابقة . تحتوي دائرة الطارح التام على ثلاثة إدخالات وهي العددين (A) و(B) والإدخال (Bi) الذي يمثل الإستعارة من المرتبة السابقة , وإخراجين وهي الإخراج (Do) الذي يمثل ناتج الطرح "Difference" والإخراج (Bo) الذي يمثل عملية الإستعارة النهائية "Borrow" .

INPUTS			OUTPUTS	
A	**B**	**B_i**	**D_O**	**B_O**
0	**0**	**0**	**0**	**0**
0	**0**	**1**	**1**	**1**
0	**1**	**0**	**1**	**1**
0	**1**	**1**	**0**	**1**
1	**0**	**0**	**1**	**0**
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**0**	**0**
1	**1**	**1**	**1**	**1**



الرمز المنطقي

من جدول الحقيقة :

وبنفس طريقة الجامع التام يمكن إيجاد معادلات الإخراج وكما يلي:

$$D_o = \overline{A}\,\overline{B}B_i + \overline{A}B\overline{B_i} + A\overline{B}\,\overline{B_i} + ABB_i$$

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

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

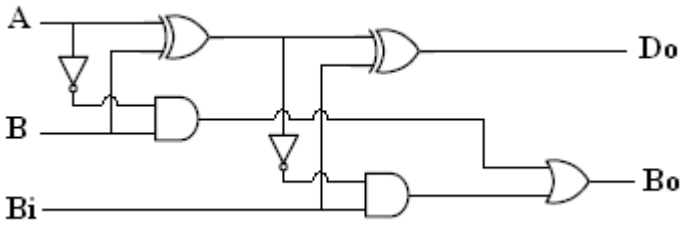
$$\mathbf{D_o = (A \oplus B) \oplus B_i} \qquad \text{..... (1)}$$

$$B_o = \overline{A}\,\overline{B}B_i + \overline{A}B\overline{B_i} + \overline{A}BB_i + ABB_i$$

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

$$\mathbf{B_o = B_i\overline{(A \oplus B)} + \overline{A}B} \qquad \text{..... (2)}$$

يمكن إستخدام معادلة (1) و معادلة (2) لرسم الدائرة المنطقية للطارح التام :



تمارين:

1− صمم دائرة الطارح التام بإستخدام دائرتين نصف طارح .

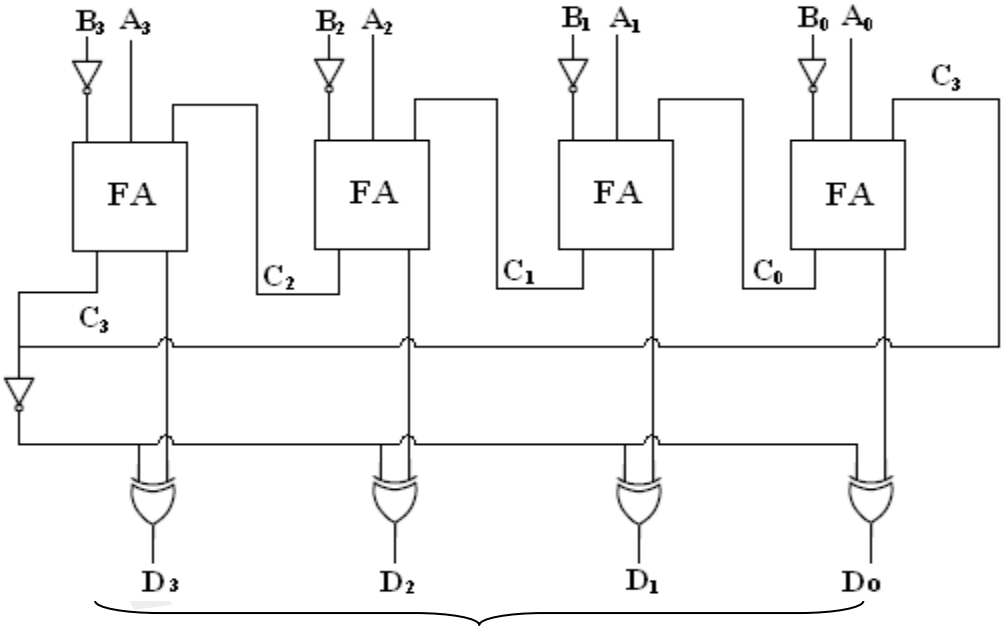
2- صمم دائرة الطارح التام بإستخدام بوابات NAND فقط .

جـ− دائرة الطرح المتوازية باستخدام دوائر الجمع المتوازية(بطريقة المتمم لـ1): (Parallel Subtractor)

هذه الدائرة هي الدائرة التطبيقية الفعلية التي تنفذ عملية الطرح في وحدة الحساب والمنطق في الحاسبة الألكترونية حيث يتم إستخدام دوائر الجمع لتنفيذ عملية الطرح بالإستفادة من مفهوم المتممات . والدائرة أدناه تقوم بطرح عددين كل منهما يتكون من أربعة مراتب ثنائية (بطريقة المتمم لـ1) :

لطرح العددين : $\mathbf{A} = A_3 \; A_2 \; A_1 \; A_0$

$\mathbf{B} = B_3 \; B_2 \; B_1 \; B_0$



ناتج الطرح

تمرين:

أرسم دائرة لطرح العدد $(11010)_2$ من العدد $(10010)_2$ بطريقة المتمم لـ1 باستخدام دوائر الجمع ثم بين النواتج على الرسم .

الأسبوع الرابع عشر

5.3 دائرة المقارن الرقمي : (Digital Comparator Circuit)

تستخدم هذه الدائرة في الأنظمة الرقمية والحاسبات الألكترونية للمقارنة بين عددين ثنائيين ويكون ناتج المقارنة أما أحد العددين أكبر أو أصغر أو يساوي العدد الآخر.

1.5.3 دائرة المقارن الرقمي لمرتبة ثنائية واحدة : (1-bit Digital Comparator Circuit)

وهي دائرة مقارنة بين عددين **A** و **B** كل منهمامؤلف من مرتبة واحدة تمثل الإدخالات للدائرة . أما إخراجات الدائرة فهي X_1 و X_2 و X_3 والتي تمثل حالات المقارنة الثلاث **A>B** و **A<B** و **A=B** على التوالي ، حيث أن في كل حالة من حالات العمل يكون أحد الإخراجات فعّال نسبة إلى حالة المقارنة بين الإدخالات .

INPUTS		OUTPUTS		
A	**B**	X_1 (A>B)	X_2 (A<B)	X_3 (A=B)
0	**0**	**0**	**0**	**1**
0	**1**	**0**	**1**	**0**
1	**0**	**1**	**0**	**0**
1	**1**	**0**	**0**	**1**

$$X_1 = A\overline{B} \quad .....(1)$$

$$X_2 = \overline{A}B \quad .....(2)$$

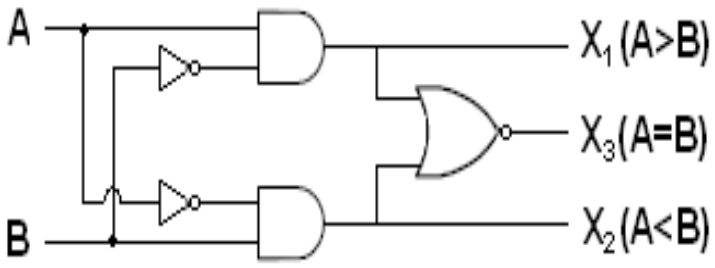
$$X_3 = \overline{A}\,\overline{B} + AB$$

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

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

$$X_3 = \overline{X_1 + X_2} \quad .....(3)$$

بإستخدام المعادلات (1) و(2) و(3) سيكون رسم الدائرة لمقارن رقمي لمرتبة واحدة كما مبين أدناه :



تمرين:

صمم دائرة المقارن الرقمي لمرتبة واحدة باستخدام بوابات NAND فقط .

2.5.3 دائرة المقارن الرقمي لمرتبتين ثنائيتين: (2-bit Digital Comparator Circuit)

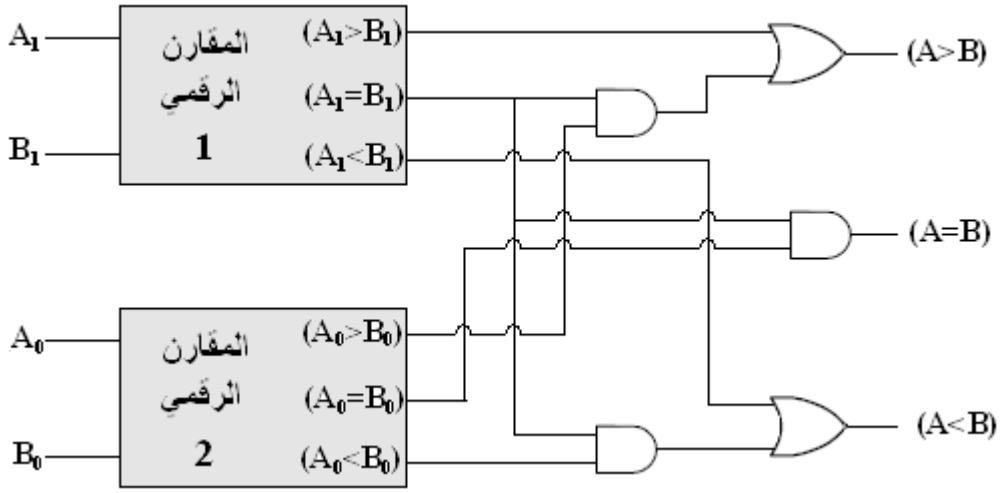
لمقارنة عددين ثنائيين كل منهما يتكون من مرتبتين مثل العدد $A = A_1A_0$ والعدد $B = B_1B_0$, فسيتم مقارنة المراتب العليا في البداية ثم مقارنة المراتب الدنيا في حالة تساوي المراتب العليا وبالتالي يكون منطق الإخراجات الثلاث للمقارن كما يلي:

$(\mathbf{A>B}) = (A_1>B_1) + (A_1=B_1).(A_0>B_0)$

$(\mathbf{A<B}) = (A_1<B_1) + (A_1=B_1).(A_0<B_0)$

$(\mathbf{A=B}) = (A_1=B_1).(A_0=B_0)$

وبذلك ستكون دائرة المقارن الرقمي لمرتبتين وجدول الحقيقة لها كما مبين أدناه :

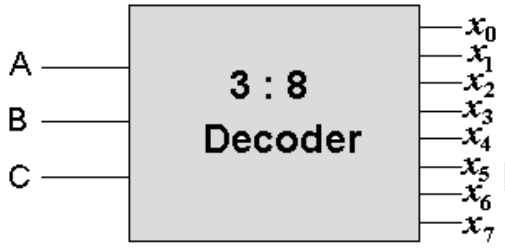


INPUTS				OUTPUTS		
A_1	A_0	B_1	B_0	X_1 (A>B)	X_2 (A<B)	X_3 (A=B)
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

الإسبوع الخامس عشر

3.3 دائرة مفك الشفرة : (Decoder Circuit)

دائرة مفك الشفرة (Decoder) دائرة تقوم بتفعيل واحد من الإخراجات التي عددها (2^n) إعتماداً على القيمة الثنائية لإدخالات البيانات التي عددها (n) . عمل دائرة مفك الشفرة على أساس إن الإخراج الذي تسلسله مساوياً للقيمة الثنائية لإدخالات البيانات هو الفعال (قيمته 1) وباقي الإخراجات غير فعاله (قيمتها 0). فمثلاً في دائرة مفك الشفرة (3:8) سيكون عدد الإدخلات ثلاثة هي (A , B , C) بينما عدد الإخراجات ($2^3 = 8$) وهي (x_0 , x_1, x_2, x_3, x_4, x_5, x_6, x_7) والمبينة في المخطط أدناه :



فعند الإحتمال الذي تكون فيه قيمة الإدخالات الثنائية (101) فإن الإخراج x_5 هو الذي سيكون فعال أما بقية الإخراجات ستكون غير فعالة. بقية إحتمالات عمل الدائرة مبينة في الجدول التالي :

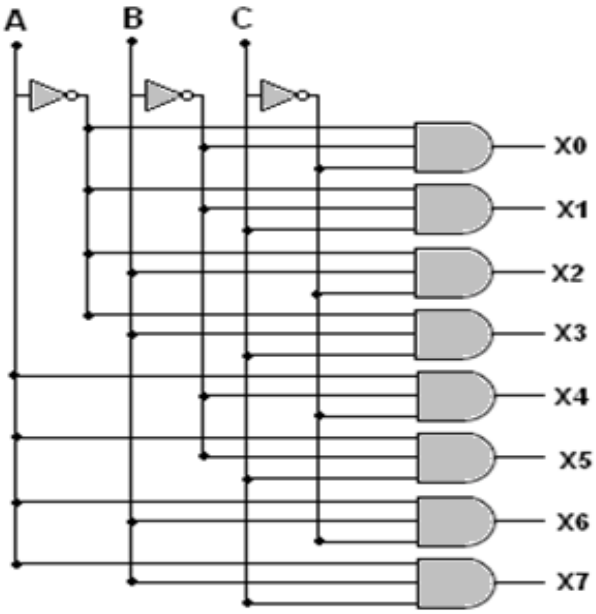
INPUTS			OUTPUTS							
A	B	C	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

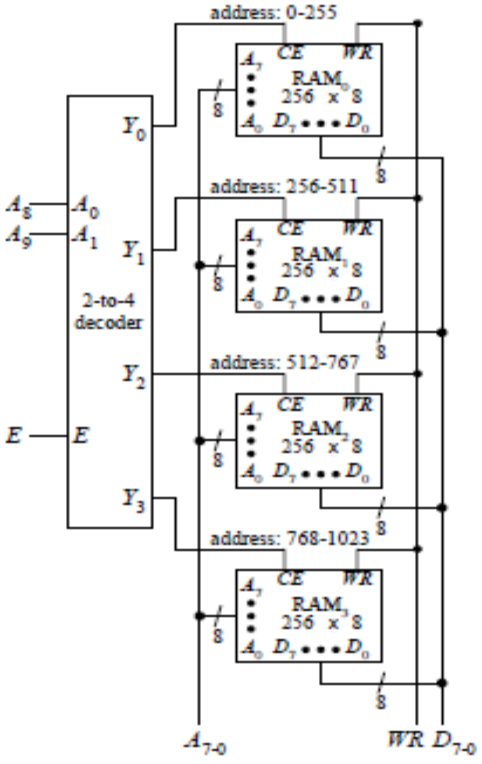
من جدول الحقيقة يمكن أستنتاج معادلات الإخراجات وستكون كما يلي :

$$x_0 = \bar{A}\bar{B}\bar{C} \quad , \quad x_1 = \bar{A}\bar{B}C \quad , \quad x_2 = \bar{A}B\bar{C} \quad , \quad x_3 = \bar{A}BC \quad ,$$

$$x_4 = A\bar{B}\bar{C} \quad , \quad x_5 = A\bar{B}C \quad , \quad x_6 = AB\bar{C} \quad , \quad x_7 = ABC$$

وعند رسم تلك المعادلات سنحصل على الدائرة التالية والتي تمثل دائرة مفك شفرة (3:8) .





دائرة مفك الشفرة تستخدم في المنظومة التي تحتوي مجموعة من المكونات ونحتاج الى واحدة من المكونات لإختيارها لغرض التشغيل في وقت معين. مثلاً منظومة ذاكرة تحتوي على عدة دوائر متكاملة , ودائرة واحدة يجب أن تعمل في كل مرة , فيتم ربط كل إخراج من إخراجات مفك الشفرة الى إدخال التمكين لكل دائرة من الدوائر المتكاملة للذاكرة , وكما مبينة في الشكل.

تمارين:

1- صمم دائرة مفك الشفرة (2:4).

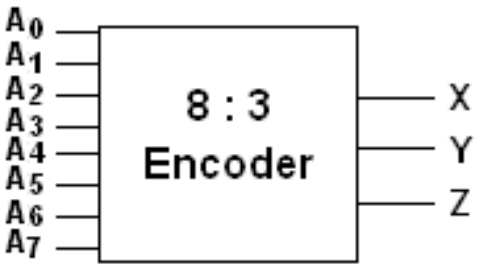
2- صمم دائرة مفك الشفرة (4:10).

3- صمم دائرة مفك الشفرة (2:4) باستخدام بوابات NAND فقط .

الإسبوع السادس عشر

4.3 دائرة الترميز : (Encoder Circuit)

دائرة الترميز عملها تقريبا عكس عمل دائرة مفك الشفرة , حيث تقوم بتحويل عدد 2^n من الإدخالات الى عدد n من الإخراجات الثنائية . الرمز المنطقي لدائرة ترميز لقيمة n = 3 مبينة في الشكل أدناه حيث يكون عدد خطوط الإدخال ثمانية هي $(A_0\ ,\ A_1,\ A_2,\ A_3,\ A_4,\ A_5,\ A_6,\ A_7)$ وخطوط الأخراج الثنائية هي ثلاثة $(X\ ,\ Y\ ,\ Z)$.



تعمل دائرة الترميز بطريقة بحيث يكون واحد فقط من الإدخالات يحمل منطق (1) بينما تكون بقية الإدخالات (0). قيمة الإخراجات الثنائية عبارة عن تسلسل الإدخال الذي قيمته (1) بالثنائي. جدول الحقيقة لدائرة ترميز (8: 3) مبينة أدناه.

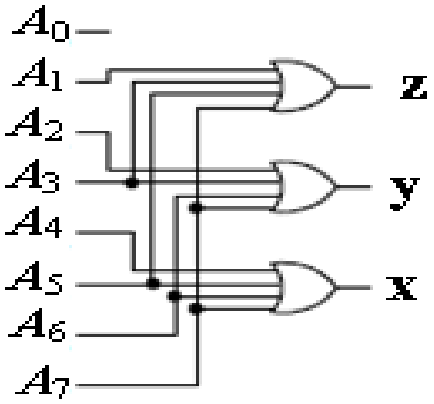
INPUTS								OUTPUTS		
A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	X	Y	Z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

فمثلاً عندما يكون الإدخال A_3 على منطق (1) , فإن قيمة الإخراجات ستكون (011) والذي يمثل القيمة الثنائية للتسلسل (3). إحتمالات الإدخالات التي فيها أكثر من إدخال فعال سوف تهمل حيث إن الدائرة لاتعمل بها.

عند النظر الى احتمالات الإخراجات الثلاثة في جدول الحقيقة سنرى أن معادلات الإخراجات الثلاثة هي

$$x = A_4 + A_5 + A_6 + A_7$$
$$y = A_2 + A_3 + A_6 + A_7$$
$$z = A_1 + A_3 + A_5 + A_7$$

وبذلك ستكون الدائرة كما يلي:



دائرة الترميز تستخدم لتقليل عدد البتات المطلوبة لتمثيل بعض البيانات المعطات لغرض الخزن أو لغرض الإرسال. كذلك تستخدم دائرة الترميز في المنظومة التي تحتوي 2^n من أجهزة الإدخال وكل منها يحتاج الإستجابة للعمل. جهاز الإدخال الذي يحصل على إستجابة للعمل يُفَعّل خط الإدخال المتصل به وبذلك سيتم توليد شفرة ثنائية على الإخراجات التي تعطي دلالة للنظام أي من أجهزة الإدخال إستجاب للعمل.

تمارين:

1- صمم دائرة ترميز (2 :4) .

2- صمم دائرة ترميز (4 :10) .

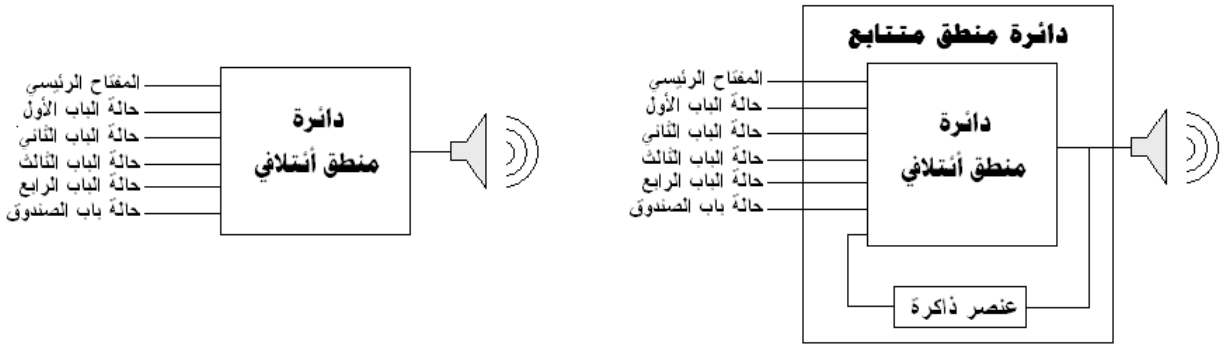
الإسبوع السابع عشر

الفصل الرابع
تصميم دوائر المنطق المتتابع
Sequentional Logic Circuits Design

1.4 مقدمة :

لقد بيّنا في دراستنا للفصل الثالث تحليل وتصميم دوائر المنطق الإئتلافي , وبرغم أهميتها الكبيرة فإنها لاتشكل إلا جزءاً من الأنظمة الرقمية أما الجزء الآخر من الأنظمة الرقمية فهو يقع ضمن تصميم دوائر المنطق المتتابع . وكما بيّنا سابقاً فإن إخراج دوائر المنطق الإئتلافي يعتمد كلياً على قيم الإدخالات الآنية للدائرة , بينما يعتمد الإخراج لدوائر المنطق المتتابع على حالة الإدخالات الآنية وعلى الحالة السابقة للإدخالات.

ولتوضيح هذه الفكرة أكثر نأخذ كمثال منظومة حماية السيارة من السرقة بشكل مبسط (كما مبين في الشكل أدناه) . في هذه المنظومة يتم تشغيل منبه الإنذارعندما يكون المفتاح الرئيسي لتشغيل المنظومة في حالة تشغيل وقام أحدهم بفتح أحد أبواب السيارة , وإذا قمت بإغلاق الباب فإن المنبه سوف يتوقف مباشرة . ولغرض جعل منظومة الحماية أكثر واقعية وكفاءة يجب جعل منبه الإنذار يستمر بالإشتغال حتى بعد إغلاق باب السيارة بعد فتحها للمرة الأولى . ولجعل هذه الدائرة تعمل بهذه الخاصية الإضافية , يجب جعل المنبه يعمل ليس فقط لتشغيل المفتاح الرئيسي وفتح أحد الأبواب ولكن أيضاً في كون حالة المنبه السابقة كانت في حالة تشغيل (أي يجب على الدائرة تذكر حالة الإخراج السابقة), وبعبارة أخرى إن هذا التطوير في عمل الدائرة هو دائرة منطق متتابع أما قبل التطوير فالدائرة هي دائرة منطق أئتلافي.



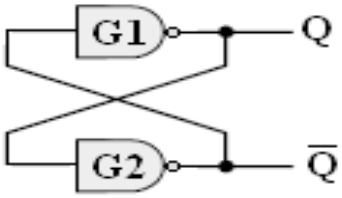
إن الإعتماد على حالة الإدخالات السابقة يتطلب وجود عنصر ذاكرة لكي يتذكر حالة الإدخالات السابقة. ولتصميم دائرة تتذكر حالتها السابقة يجب ربط خارج الدائرة بصورة مباشرة أو غير مباشرة رجوعا الى إدخالات نفس الدائرة , وجزء الدائرة الراجع هذا يسمى بالتغذية الخلفية (Feed Back) والذي يشكل الأساس في تركيبة عناصر الذاكرة . ولو لاحظنا دوائر المنطق الأئتلافي السابقة نراها لاتحتوي على خط التغذية الخلفية.

إن الماسكات (Latches) والنطاطات (Flip-Flops) هي العناصر الأساسية لخزن المعلومات وبالتالي فهي التراكيب الأساسية لبناء دوائر المنطق المتتابع . وبشكل عام يوجد أربعة أنواع أساسية من النطاطات هي النطاطات (SR, D , JK , T) التي سيتم شرحها في هذا الفصل . الإختلاف الأساسي بين تلك النطاطات هو عدد الإدخالات للدائرة وكيفية تغيير محتوياتها . أي دائرة منطق متتابع يمكن أن تحتوي على أي من هذه الأنواع أو مجموعة منها.

2.4 النطاطات : (Flip-Flops)

في الدوائر الرقمية النطاط هو نوع من المذبذبات ثنائية الإستقرار(Bi-stable Multivibrator) وهي دائرة الكترونية لها حالتين مستقرتين لذلك يمكن إستخدامها كخلية ذاكرة واحدة (1-bit memory cell) قادرة على خزن المعلومات الثنائية. يمكن تصميمها باستخدام بوابات NAND أو بوابات NOR .

الدائرة الأساسية لتركيب النطاطات هي المبينة في الشكل أدناه. تتألف من عاكسين G1 و G2 (أستعملت بوابات NAND كعاكس) وقد تم ربط إخراج G1 الى إدخال G2 كما ربط إخراج G2 الى أدخال G1 .



لنفرض أن إخراج G1 كان (Q = 1) وهو نفسه إدخال لـ G2, لذلك فإن إخراج G2 سيكون ($\overline{Q}$ = 0) ويكون هذا الأخير كإدخال لـ G1 مما يؤدي الى أن يكون الإخراج (Q = 1) وتستقر الدائرة على هذه الحالة. وبنفس الطريقة إذا إفترضنا أن إخراج G1 كان (Q = 0) والذي يؤدي الى أن إخراج G2 سيكون ($\overline{Q}$ = 0) وتستقر الدائرة على هذه الحالة.

من خلال ماتم توضيحه إعلاه يمكن إستخلاص الملاحظات التالية :

1- الإخراجان Q و $\overline{Q}$ يكونان متتامين دائماً.

2- تمتلك الدائرة إحى الحالتين المستقرتين التاليتين :

الحالة الأولى : تكون (Q = 1) , ($\overline{Q}$ = 0) وتسمى حالة الوضع (Set state) .

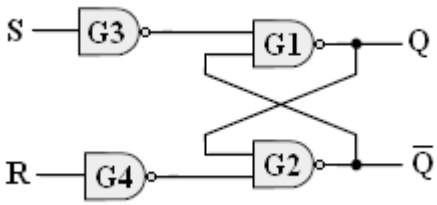
الحالة الثانية : تكون (Q = 0) , ($\overline{Q}$ = 1) وتسمى حالة التصفير (Reset state) .

لذلك تسمى هذه الدائرة في بعض الأحيان بدائرة **ثنائي الإستقرار** (Bi-stable circuit) .

3- إذا كان خارج الدائرة في حالة (Q = 1) فإن الدائرة ستستمر على هذه الحالة , وكذلك إذا كان خارج الدائرة في حالة (Q = 0) فإنها ستستمر على هذه الحالة. هذه الصفة المميزة تعطي الدائرة إمكانية خزن معلومة ثنائية واحدة , وبسبب حجز تلك المعلومة أو الإمساك بها فإن هذه الدائرة تسمى **الماسِك** (Latch) .

1.2.4 النطاط نوع S - R : (S – R Flip-Flop)

في دائرة الماسك في الفقرة السابقة نلاحظ عدم وجود طريقة لإدخال المعلومات الرقمية التي نرغب بتخزينها فيه. عند توصيل القدرة, ستتحول الدائرة الى أحد حالات الإستقرار (Q = 0) أو (Q = 1), وبالحقيقة , لايمكن التنبؤ بالحالة . وإذا إستبدلنا العواكس G1 و G2 ببوابات NAND ثنائية الإدخال, فنستطيع إستعمال طرف الإدخال الثاني لبوابات NAND لإدخال المعلومات الرقمية التي نرغب بخزنها.وبعد إضافة عاكسان إضافيان G3 وG4 الى الإدخالات تصبح الدائرة المحورة والمبينة في الشكل أدناه تسمى **بالنطاط S-R** .



في هذه الدائرة هنالك أربعة إحتمالات لعمل الدائرة :

<u>الإحتمال الأول:</u> عندما يكون (S = R = 0) : في هذا الإحتمال سيكون عمل الدائرة مشابه الى الدائرة قبل التحوير , حيث سيبقى الإخراج مساوياً الى حالة الإخراج السابقة. يسمى هذا الإحتمال **بإحتمال عدم التغيير(No Change)** .

<u>الإحتمال الثاني:</u> عندما يكون (S =1) و(R = 0) : في هذه الحالة سيكون إخراج G3 يساوي 0 و إخراج G4 يساوي 1, وبما أن أحد إدخالات G1 أصبح يساوي 0 فإن إخراجها سيكون (Q = 1) بالتأكيد, وبالتالي كلا ألإدخالين لـ G2 سيكونان 1 والذي ينتج عنه إخراجاً ($\overline{Q}$ = 0)يسمى هذا الإحتمال **بإحتمال الوضع(Set)** **<u>الإحتمال الثالث:</u>** عندما يكون (S =0) و(R = 1) : في هذه الحالة سيكون إخراج G3 يساوي 1 و إخراج G4 يساوي 0, وبما أن أحد إدخالات G2 أصبح يساوي 0 فإن إخراجها سيكون ($\overline{Q}$ = 1) بالتأكيد, وبالتالي كلا ألإدخالين لـG1 سيكونان 1 والذي ينتج عنه إخراجاً(Q = 0)يسمى هذا الإحتمال **بإحتمال التصفير(Reset)** **(ملاحظة مهمة** إن سبب تسمية طرفا الإدخال بـ S و R وذلك لأن جعل (S=1) يضع الدائرة في حالة وضع(Set) وجعل (R=1) يجعل الدائرة في حالة تصفير(Reset)).

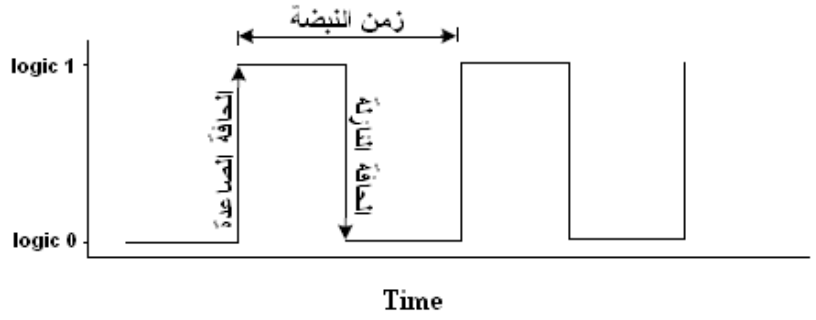
<u>الإحتمال الرابع:</u> عندما يكون (S =1) و(R = 1) : في هذه الحالة سيحاول كلا الإخراجين Q و $\overline{Q}$ **أن** يصبحا على منطق 1 وهذه حالة غير مسموح بها , لذلك يحضر شرط الإدخال هذا في النطاط نوع S-R .

يمكن تلخيص حالات العمل الأربعة للنطاط نوع S-R بالجدول التالي :

S	R	حالةQ الجديدة
0	0	نفس الحالة السابقة
0	1	0
1	0	1
1	1	حالة غير مسموحة

2.2.4 <u>نبضات الساعة :</u> (Clocke Pulses)

بعض دوائر المنطق المتتابع يتحدد تصرفها أوحالة إخراجها عند لحظة محددة من الزمن يعرف هذا النوع من الدوائر **بالدوائر المتتابعة التزامنية** (Synchronous Sequentional Circuits) . يتم إنجاز التزامن بأداة تسمى **نبضات الساعة** (Clock pulses) , وهي عبارة عن مجموعة من النبضات المربعة المنتظمة المتوالية والتي تتناوب بين منطق 0 ومنطق 1 , كما مبينة أدناه, يختلف عرض أو زمن النبضة من تطبيق الى آخر. يسمى جزء النبضة عندماتنتقل الإشارة من 0 الى 1 **بالحافة الصاعدة أو الحافة الموجبة**(Rising or Positive edge) , وعندما تنتقل الإشارة من 1 الى 0 **بالحافة النازلة أو الحافة السالبة**(Falling or Negetive edge), هذه الحافات تعتبر المناطق الفعالة في حساب الحالة الجديدة للدائرة.

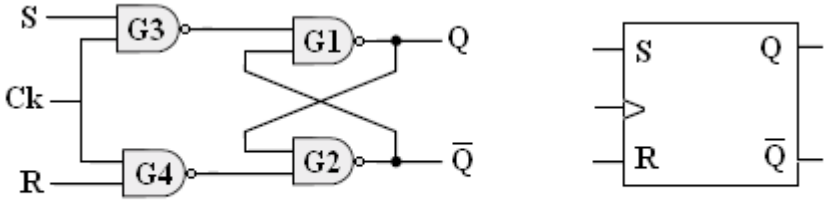


تقوم نبضات الساعة بتنظيم وتوقيت العمليات في الأنظمة الرقمية بإعطاء فترة سماحية للنتائج الجديدة لغرض حسابها من قبل البوابات المنطقية ومن ثم مسك هذه النتائج في النطاطات عند حافات نبضة الساعة.

إن فترة نبضة الساعة الواحدة يتم إحتسابها من بداية الحافة الصاعدة الى الحافة الصاعدة التالية (أو من بداية الحافة النازلة الى الحافة النازلة التالية). سرعة نبضة الساعة تقاس بالهيرتز (Hz) والذي يمثل عدد النبضات في الثانية الواحدة, فمثلاً سرعة نبضة الساعة في المعالجات والتي تعمل كجزء في منظومة رقمية تتراوح بين 20 – 4) (Hz,بينما المعالجات الدقيقة في الحاسبات الشخصية الحديثة فتعمل بأكثرمن (2GHz) .

3.2.4 النطاط نوع S – R المتزامن : (Clocked S – R Flip-Flop)

إن مصطلح النطاط المتزامن يعني إن الإخراج للنطاط يغير حالته المنطقية فقط عند زمن محدد تبعاً لحالة الإدخالات. يمكن جعل النطاط متزامناً بإضافة إدخال نبضات الساعة يصمم كإدخال سيطرة بحيث إن التغير في الإخراج يظهر بالتزامن مع نبضة الساعة.لتحويل النطاط S-R الى النوع المتزامن يتم إضافة نبضات الساعة(Ck) الى إدخال النطاط عند إدخال البوابات G3 وG4 وكما مبين في الدائرة أدناه.



دائرة النطاط نوع S – R المتزامن الرمز المنطقي

عمل هذه الدائرة أصبح مسيطر عليه من قبل إدخال نبضات الساعة (Ck) , وكما يلي :

1-<u>عندما تكون (Ck =0)</u> فإن إخراج كل من البوابات G3 و G4 يصبح 1 بغض النظر عن حالة الإدخالات S و R وبالتالي يبقى إخراج النطاط على نفس القيمة السابقة , أي إن تأثير إدخالات الدائرة على الإخراج يصبح غير فعال .

2-**<u>عندما تكون (Ck =1)</u>** فإن إخراج كل من البوابات G3 وG4 سيعتمد على حالة الإدخالات S و R وبالتالي ستعتمد حالة الإخراج النهائية على حالة الإدخالات S و R , أي أن تأثير إدخالات الدائرة على الإخراج يصبح فعال وعمل الدائرة يصبح مشابهاً الى عمل دائرة النطاط S-R غير المتزامن .

يمكن تلخيص عمل دائرة النطاط S-R المتزامن عندما تكون (Ck = 1) بالجدول التالي حيث أن :

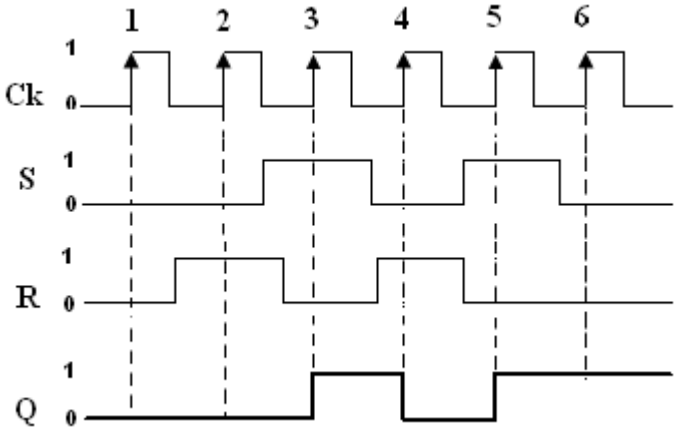
S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	?

Q_n : تمثل حالة الإخراج Q الحالية عند نبضة الساعة n .

Q_{n+1} : تمثل حالة الإخراج Q الجديدة بعد النبضة الساعة n .

? : تعني أن حالة إخراج الدائرة غير محددة.

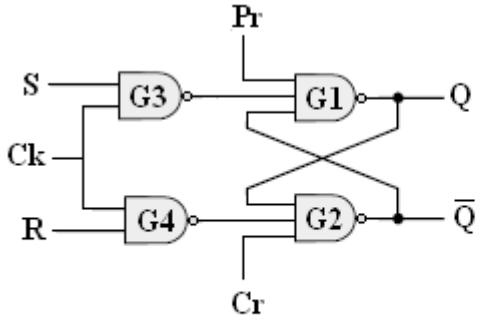
مـثـال إذاكانت حالة الإدخالات للنطاط S-R لفترة ستة نبضات للساعة كما مبينة في الشكل أدناه.بين حالة الإخراج Q خلال هذه الفترة,علماً إن النطاط يتأثر بالحافة الصاعدة لنبضات الساعة.



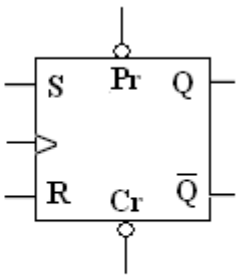
الإسبوع الثامن عشر

4.2.4 الإدخالات المباشرة (أو غير المتزامنة) (Direct (Asynchronous) Inputs)

في النطاط S-R المتزامن تكون حالة الدائرة غير محددة عند توصيل القدرة . فقد تصبح الدائرة في حالة الوضع (Q = 1) أو في حالة التصفير (Q = 0) . ولكن في كثير من التطبيقات نحتاج الى تحديد الحالة الإبتدائية للنطاط على حالة الوضع أو التصفير قبل البدأ بتسليط نبضات الساعة أو البدأ بعمل الدائرة. يمكن إنجاز ذلك باستخدام الإدخالات المباشرة أو اللاتزامنية التي تسمى بإدخال الوضع المسبق Pr (Preset) وإدخال التصفير Cr (Clear) والتي يمكن إضافتها الى النطاط S-R عند إدخالات البوابات G1 و G2 على التوالى لتصبح الدائرة كما مبينة أدناه.



دائرة النطاط نوع S-R بوضع مسبق وتصفير



الرمز المنطقي

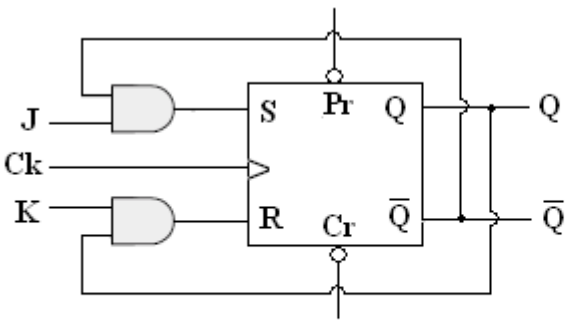
يمكن في أي وقت لتحديد حالة النطاط الإبتدائية أو بين نبضات التوقيت . تكون فعالية هذه الإدخالات عندما يكون الإدخال على منطق 0 (لهذا السبب تم إستخدام الفقاعات (Bubbles) لإدخالات Pr و Cr في الرمز المنطقي), لذلك عندما **تكون (Pr = Cr = 1)** سيكون كلا الإدخالين غير فعال وستعمل الدائرة كما في جدول الحقيقة للنطاط S-R وبتأثير نبضات الساعة . أما إذا **كانت (Cr = 1 , Pr = 0)** فإن الإخراج الدائرة سوف تستقر على (Q = 1) , وإذا **كانت (Cr = 0 , Pr = 1)** فإن الإخراج الدائرة سوف تستقر على (Q = 0) .يجب عدم إستخدام الحالة **(Pr = Cr = 0)** لأنها ستؤدي الى حالة إخراج غير محددة.

يمكن تلخيص حالات عمل الإدخالات المباشرة بالجدول التالي:

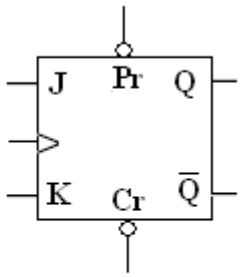
Cr	Pr	حالة النطاط
0	**0**	**لاتستخدم**
0	**1**	**تصفير Q=0**
1	**0**	**إعادة وضع Q=1**
1	**1**	**نطاط إعتيادي**

5.2.4 النطاط نوع J – K : (J – K Flip-Flop)

إن النطاط نوع S-R يحتوي على حالة غير معرفة عندما تكون الإدخالات (S = R =1) وهذا يعتبر خسارة كبيرة في عمل دائرة النطاط. ولغرض معالجة هذه الحالة وتحويلها الى حالة مفيدة تم تصميم **النطاط نوع J-K** وذلك بإضافة بوابتين AND الى دائرة النطاط S-R وعمل تغذية عكسية من إخراجات النطاط الى إدخالاته عبر بوابات AND أي إن(S = J.$\overline{Q}$) و (R = K.Q) وكما مبينة في الشكل أدناه :



دائرة النطاط نوع J – K



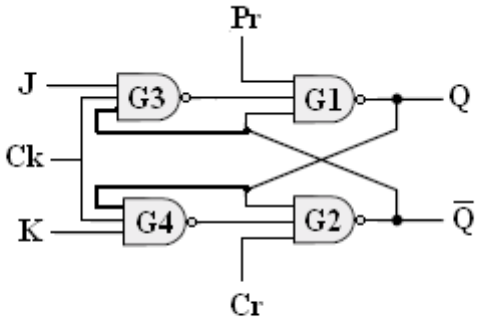
الرمز المنطقي

إن عمل النطاط نوع J-K مشابه الى حد كبيرلعمل النطاط نوع S-R . فالإدخال J مشابه تماماً للإدخال S عندما يُفعَّل يؤدي الى جعل النطاط في حالة وضع. كذلك، الإدخال K عندما يُفعَّل يؤدي الى جعل النطاط في حالة

تصفير. الإختلاف الوحيد هو **عندما يكون (J = K = 1) تكون الحالة الجديدة للإخراج Q عكس الحالة السابقة** وتسمى تلك الحالة بحالة **الإقلاب** (Toggle) . وبذلك يمكن تلخيص عمل النطاط نوع J-K بالجدول أدناه.

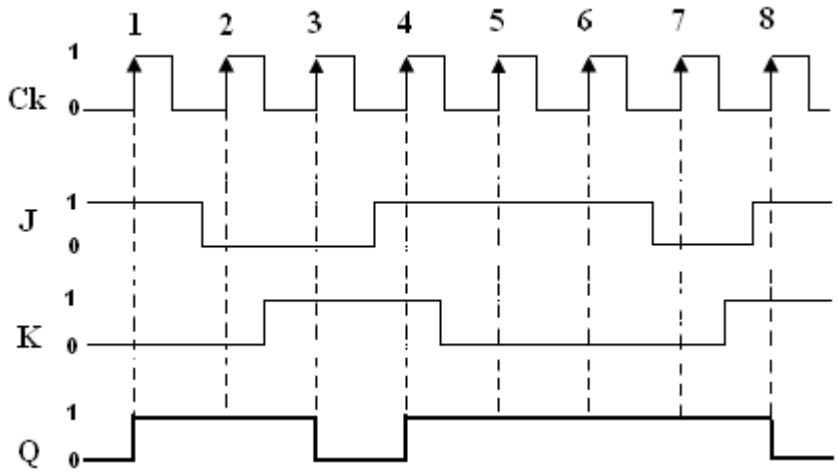
ليس من الضروري تصميم النطاط نوع J-K من النطاط نوع S-R ولكن نستطيع تصميم النطاط نوع J-K باستخدام بوابات NAND فقط مع بقاء عمل الدائرة بنفس جدول الحقيقة , وكما مبين في الدائرة أدناه :

J	K	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	$\overline{Q_n}$



دائرة النطاط نوع J-K بإستخدام بوابات NAND

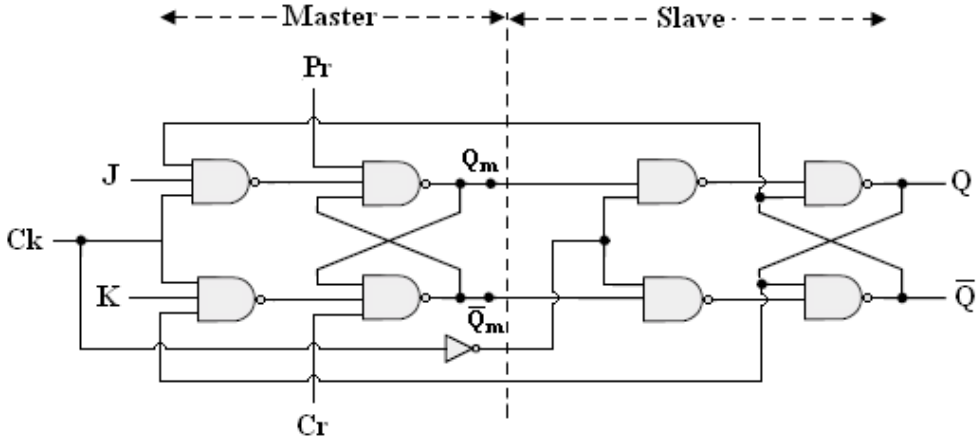
مـثـال إذاكانت حالة الإدخالات للنطاط نوع J-K لفترة ثمانية نبضات للساعة كما مبينة في الشكل أدناه . بين حالة الإخراج Q خلال هذه الفترة , علماً إن النطاط يتأثر بالحافة الصاعدة لنبضات الساعة.



6.2.4 النطاط نوع J – K السيد – العبد : (Master – Slave J – K Flip-Flop)

بالرغم من أن النطاط نوع J-K لايحتوي على حالة غير معرفة ولكنه يعاني من مشكلة تسمى **بالتسابق الزمني** (Time racing) هذه المشكلة تحدث عندما تكون الإدخالات (J= K = 1) وحالة نبضات الساعة (Ck =1) وعرض النبضة أكبر بكثير من زمن الإنتشار للنطاط , حيث يقوم إخراج النطاط Q بالتذبذب بين 0 و 1 لحين إنتهاء النبضة . وفي هذه الحالة ستعتمد الحالة التي يستقر عليها النطاط على عرض النبضة وليس على حالة Q السابقة.

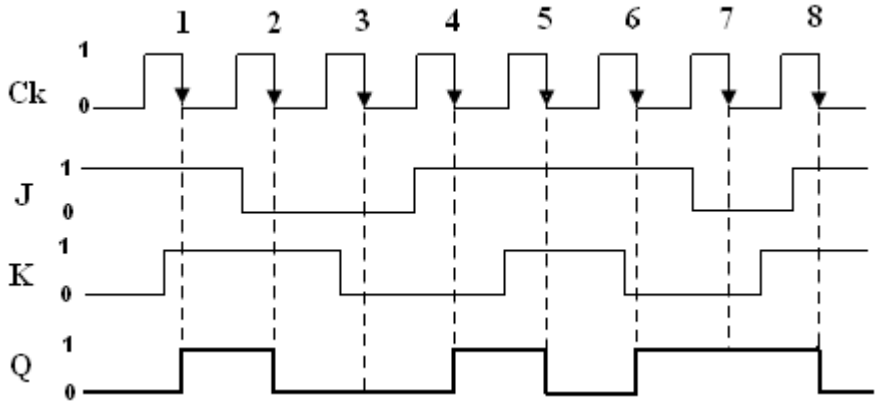
يمكن تلافي مشكلة التسابق الزمني نظرياً بجعل عرض نبضة الساعة أقل من زمن الإنتشار للنطاط , وهذه العملية صعبة التطبيق عملياً. الحل العملي لهذه المشكلة هو بتصميم نوع أخر من النطاطات يسمى **النطاط J-K السيد– العبد** . يتكون هذا النطاط من نطاطين نوع S-R مع عمل تغذية خلفية عكسية من إخراجات النطاط الأول الى إدخالات النطاط الثاني وجعل نبضات الساعة للنطاط الثاني عكس نبضات الساعة للنطاط الثاني, وكما مبين في الشكل أدناه.



دائرة النطاط نوع J-K السيد/العبد الرمز المنطقي

فعندما تكون (Ck = 1) فإن النطاط الأول سيعمل ويكون منطق النقاط Qm و $\overline{Q}$m حسب قيمة الإدخالات J و K والحالة السابقة للإخراج Q , وخلال هذا الوقت يكون النطاط الثاني متوقف عن العمل. في النصف الثاني من نبضة الساعة وعندما تكون (Ck = 0) فإن هذا سيؤدي الى توقف النطاط الأول عن العمل وعمل النطاط الثاني , حيث ستعتمد قيمة الإخراجات النهائية Q و $\overline{Q}$ على منطق النقاط Qm و $\overline{Q}$m في النصف الأول من النبضة. وبسبب إعتماد منطق النطاط الثاني على إخراج النطاط الأول سمي الأول بالسيد والثاني بالعبد. جدول الحقيقة للنطاط J-K السيد– العبد هو نفس جدول الحقيقة للنطاط J-K الإعتيادي ولكن بدون مشكلة التسابق الزمني. ويكون تأثر النطاط بالحافة السالبة لنبضات الساعة.

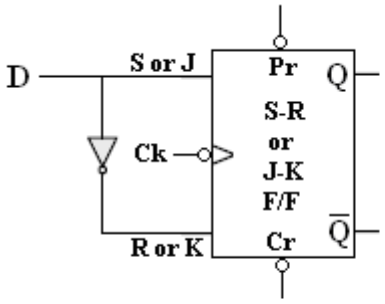
مـثـال إذاكانت حالة الإدخالات للنطاط نوع J-K السيد/العبد لفترة ثمانية نبضات للساعة كما مبينة في الشكل أدناه . بين حالة الإخراج Q خلال هذه الفترة , علماً إن النطاط يتأثر بالحافة النازلة لنبضات الساعة.



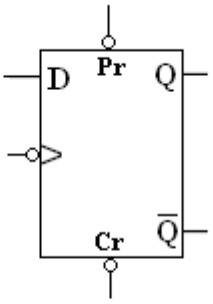
الإسبوع التاسع عشر

7.2.4 النطاط نوع – D : (D – type Flip-Flop)

يمكن الحصول على **النطاط نوع – D** من النطاط نوع S-R أو النطاط نوع J-K وذلك بربط عاكس بين الإدخال الأول والإدخال الثاني ثم جعل إدخال العاكس هو إدخال النطاط D وكما مبينة في الشكل أدناه . جدول الحقيقة للنطاط يبين أن إخراج النطاط Q بعد نبضة الساعة يساوي إدخال النطاط D قبل النبضة.



دائرة النطاط نوع – D



الرمز المنطقي

D	Q_{n+1}
0	0
1	1

جدول الحقيقة

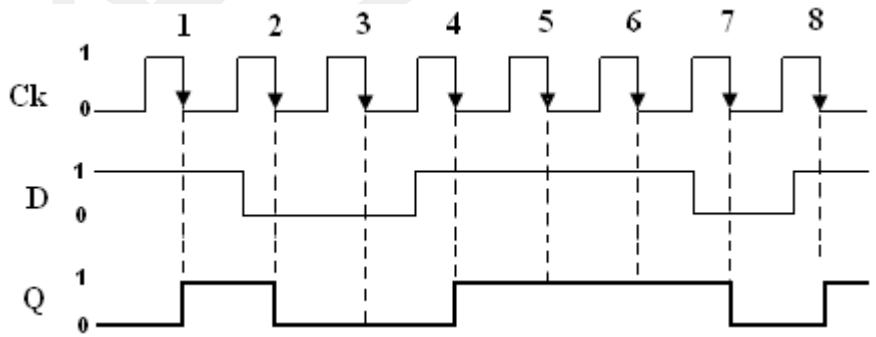
من أهم تطبيقات النطاط نوع – D هي :

1- **لإحداث تأخير زمني في الإشارة الرقمية** , ولهذا السبب سمي هذا النطاط بنوع D نسبة الى كلمة تأخير الإنكليزي (Delay) .

2- **لبناء دوائر سجلات الإزاحة** (Shift Registers), والتي تعتبرمن الدوائرالمهمة في تصميم الحاسبة الألكترونية والأنظمة الرقمية.

3- **يعتبر العنصر الأساسي في تركيب دوائر الذاكرات** (Memories).

مـثـال إذاكانت حالة الإدخالات للنطاط نوع -D لفترة ثمانية نبضات للساعة كما مبينة في الشكل أدناه. بين حالة الإخراج Q خلال هذه الفترة , علماً إن النطاط يتأثر بالحافة النازلة لنبضات الساعة.

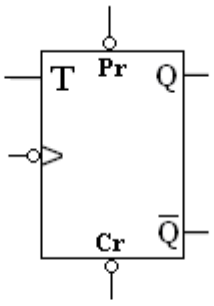


8.2.4 النطاط نوع - T : (T – type Flip-Flop)

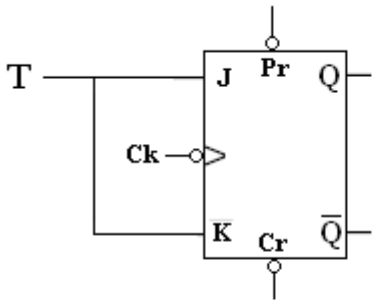
يمكن الحصول على **النطاط نوع – T** من النطاط نوع J-K فقط وذلك بربط كلا الإدخالين مع بعضهما لتكوين إدخال واحد هو الإدخال T. جدول الحقيقة للنطاط يبين أن إخراج النطاط Q بعد نبضة الساعة يساوي الحالة السابقة للإخراج إذا كان إدخال النطاط (T = 0) , وعكس الحالة السابقة للإخراج إذا كان إدخال النطاط (T = 1) . دائرة هذا النوع وجدول الحقيقة مبينة في أدناه :

T	Q_{n+1}
0	Q_n
1	$\overline{Q_n}$

جدول الحقيقة



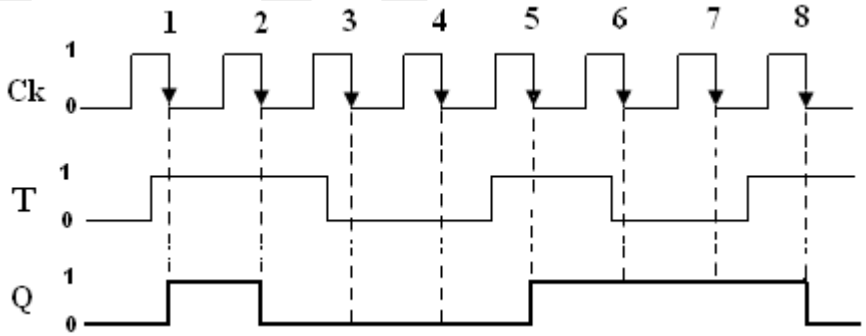
الرمز المنطقي



دائرة النطاط نوع - T

من أهم التطبيقات العملية للنطاط نوع – T هو في **بناء دوائر العدادات** (Counters) والتي تعتمد في أكثر الأحيان على حالة الإقلاب(Toggle) في جدول الحقيقة لذلك سمي هذا النطاط بالنوع T .

مثال إذاكانت حالة الإدخالات للنطاط نوع -T لفترة ثمانية نبضات للساعة كما مبينة في الشكل أدناه. بين حالة الإخراج Q خلال هذه الفترة , علماً إن النطاط يتأثر بالحافة النازلة لنبضات الساعة.



الأسبوع العشرون

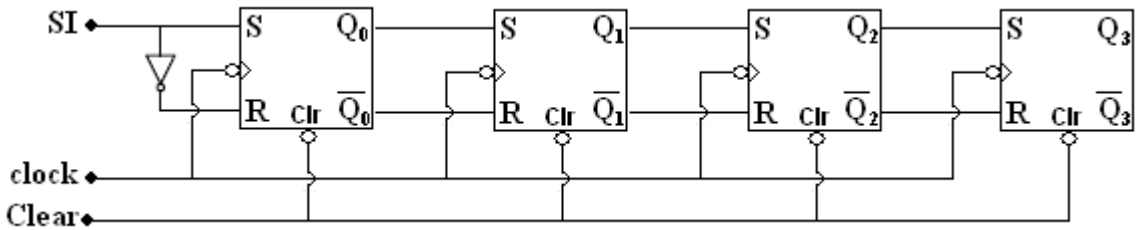
الفصل الخامس
تطبيقات على دوائر المنطق المتتابع
Sequentional Logic Circuits Applications

1.5 مقدمة :

من أهم التطبيقات لدوائر المنطق المتتابع والتي تمت دراستها في الفصل الرابع في الدوائر الرقمية والحاسبات الدقيقة هي دوائر السجلات والعدادات والتي يعتبر النطاط العنصر الرئيسي الداخل في تصميمها . سيتم في هذا الفصل التطرق الى طرق تصميم هذه الأنواع من الدوائر بشكل مبسط وسريع .

2.5 سجلات الإزاحة: (Shift Registers)

وهي عبارة عن مجموعة من دوائر النطاط S-R أو نوع – D المربوطة بشكل متوالي لغرض خزن معلومة ثنائية تتألف من عدة مراتب. تعمل هذه الدائرة على أساس إزاحة البيانات في داخلها بإتجاه اليمين أو بإتجاه اليسار. يكون عدد دوائر النطاط التي يتألف منها السجل يعتمد على طول المعلومة المطلوب المطلوب خزنها أوعدد المراتب الثنائية. الشكل أدناه يمثل دائرة سجل إزاحة نحو اليمين لأربعة مراتب ثنائية باستخدام المرجاح نوع S-R .

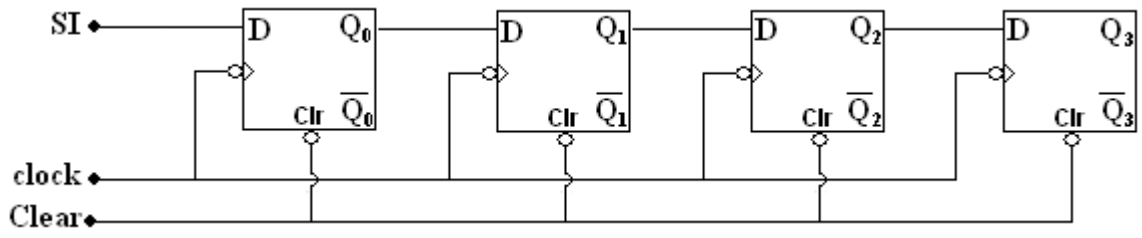


لإدخال المعلومة $(1011)_2$ وخزنها داخل السجل أعلاه , في البداية نقوم بتصفير السجل بوضع منطق(0) على إدخال التصفير(clear) ثم إعادته الى منطق(1) والبدأ بإدخال المعلومة من خلال إدخال التوالي (SI) وإبتداً من أول مرتبة من اليمين وإعطاء نبضة الساعة على الإدخال (clock) ثم نكرر العملية لبقية المراتب , أي نحتاج الى أربعة نبضات لإدخال المعلومة الى داخل السجل.

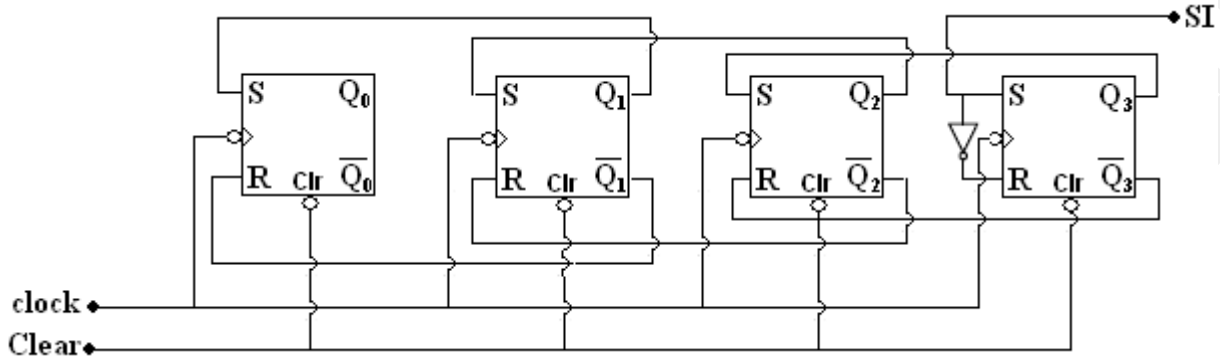
وكما مبين في الجدول أدناه , نلاحظ في نهاية النبضة الرابعة المعلومة مستقرة في داخل السجل .

Clear	Ck	SI	Q_0	Q_1	Q_2	Q_3
0	X	X	0	0	0	0
1	1	1	1	0	0	0
1	2	1	1	1	0	0
1	3	0	0	1	1	0
1	4	1	1	0	1	1

كما يمكن إستخدام النطاط نوع – D في تصميم السجل وكما موضح في الشكل أدناه .



و يمكن تصميم سجل إزاحة نحو اليسار باستخدام النطاط S-R وكما مبين أدناه.



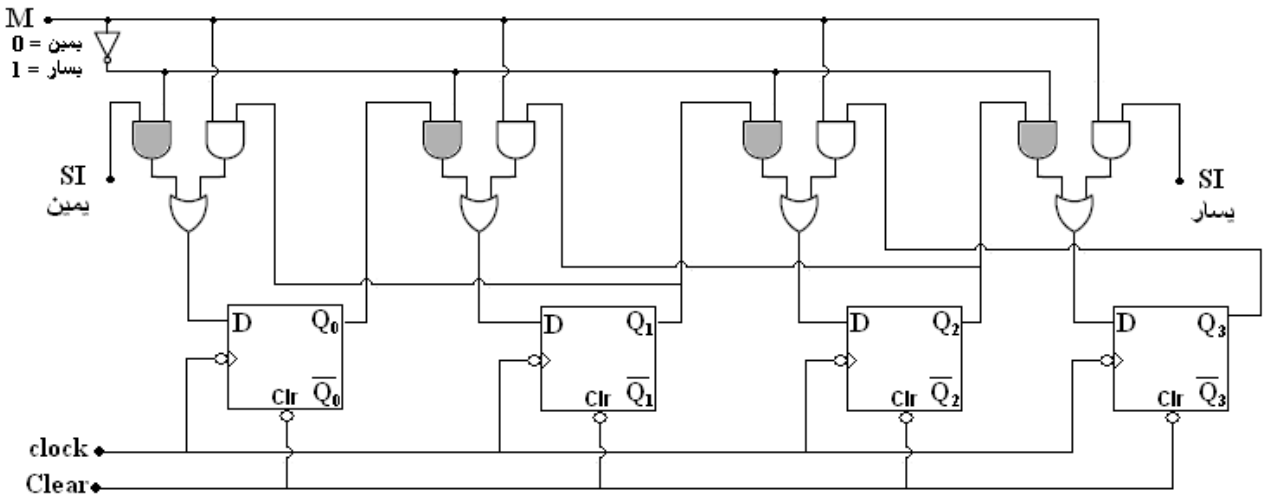
و لإدخال نفس المعلومة السابقة (1011) في سجل إزاحة نحو اليسار, سيكون إدخال المعلومة عن طريق النطاط Q_3 و إبتداءاً من المرتبة الأولى من اليسار وكما في الجدول أدناه .

Clear	Ck	SI	Q_0	Q_1	Q_2	Q_3
0	X	X	0	0	0	0
1	1	1	0	0	0	1
1	2	0	0	0	1	0
1	3	1	0	1	0	1
1	4	1	1	0	1	1

3.5 سجل إزاحة ثنائي الإتجاه: (Bidirectional Shift Registers)

في بعض التطبيقات نحتاج الى سجل إزاحة يعمل بإتجاهين (نحو اليمين ونحو اليسار)وهو المبين في الشكل أدناه , حيث ان المنطق على الإدخال M هو الذي يحدد إتجاه إزاحة المعلومة داخل السجل .

عندما تكون (M = 0) سيؤدي ذلك الى عمل بوابات AND المظللة وإغلاق باقي بوابات AND مما يؤدي الى توصيل التوصيلات الخاصة بسجل إزاحة اليمين بين إدخالات وإخراجات دوائر النطاطات. ويكون العكس عند وضع الإدخال (M = 1) حيث سيتحول السجل الى سجل إزاحة نحو اليسار.



تمارين:

صمم دائرة سجل إزاحة نحو اليسار باستخدام دوائر النطاط نوع – D ولأربعة مراتب.

الأسبوع الحادي والعشرون

1.4.5 العدادات غير المتزامنة(التموجية): (Asynchronous OR Ripple Counters)

في هذا النوع من العدادات يتم ربط إدخال كل نطاط نوع – T الى منطق(1) وتكون نبضات الساعة (clock) داخلة لأول نطاط فقط وربط إخراج كل نطاط من النطاطات الى إدخال نبضة الساعة للنطاط الذي يليه. تكون عدد حالات العد المختلفة للعداد تساوي أقل أو تساوي (2^N) حيث N تمثل عدد النطاطات الداخلة في تركيب العداد , وتسمى عدد الحالات المختلفة التي يمر بها العداد **بالمُعامِل (Modulus)** وبشكل مختصر كلمة **(Mod)** .

من مميزات هذا النوع من العدادات :

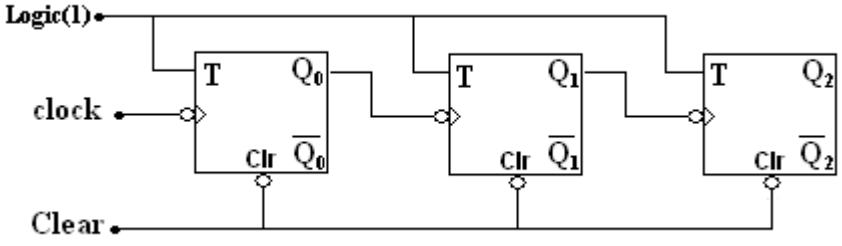
1- سهولة التصميم.

2- حالات العد تكون بالثنائي المتسلسل.

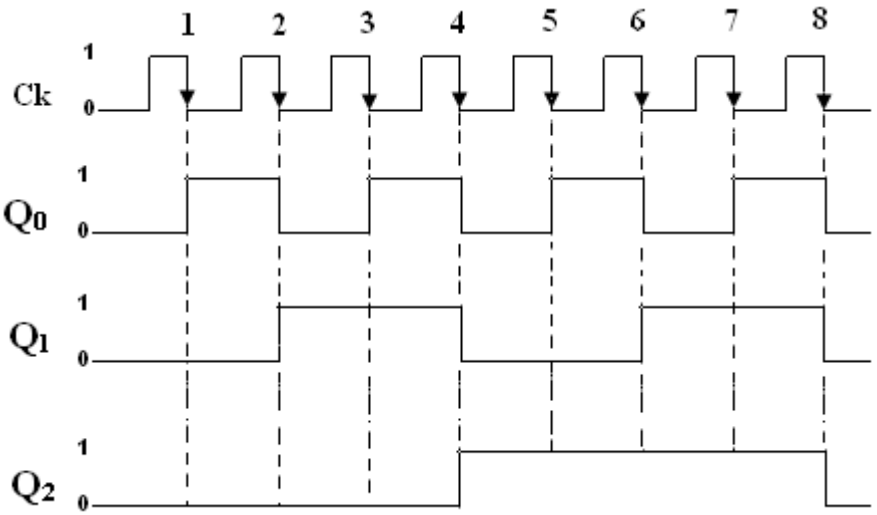
3- دوائر النطاط لاتعمل في وقت واحد حيث يقدح أحدهما الآخر.

4- بطيء العد مقارنة بالأنواع الأخرى.

مـثـال لتصميم عداد غير متزامن تصاعدي معامل – 8 .



Clear	Ck	Q_0	Q_1	Q_2
0	X	0	0	0
1	1	1	0	0
1	2	0	1	0
1	3	1	1	0
1	4	0	0	1
1	5	1	0	1
1	6	0	1	1
1	7	1	1	1
1	8	0	0	0



(جدول العد للعداد) (مخطط الإشارة للعداد)

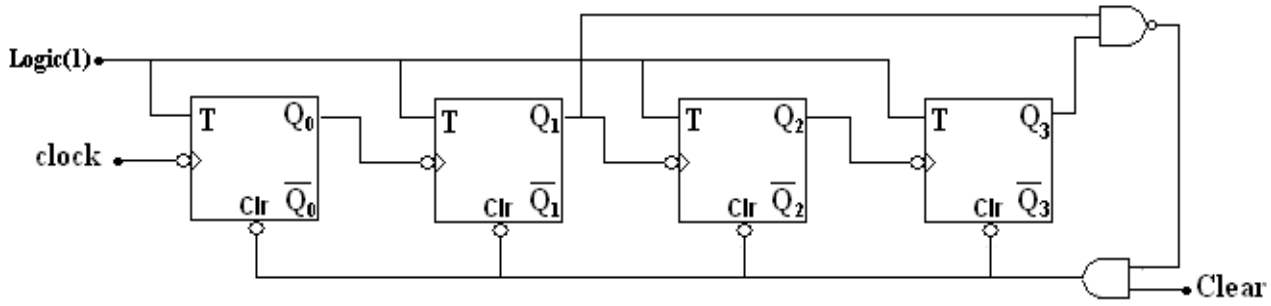
من خلال النظر الى مخطط الإشارة للعداد نلاحظ أن خلال الفترة الزمنية التي تم رسمها تكون عدد نبضات الساعة الداخلة ثمانية نبضات, بينما عدد نبضات الإخراج للنطاط Q_0 هي أربعة, وإخراج النطاط Q_1 نبضتين و Q_2 نبضة واحدة, لذلك يمكن إستخدام العداد كمقسم للتردد , حيث يعمل النطاط نوع – T كمقسم على 2 لتردد الإشارة الداخلة الى مدخل (Ck) للنطاط .

تمارين:

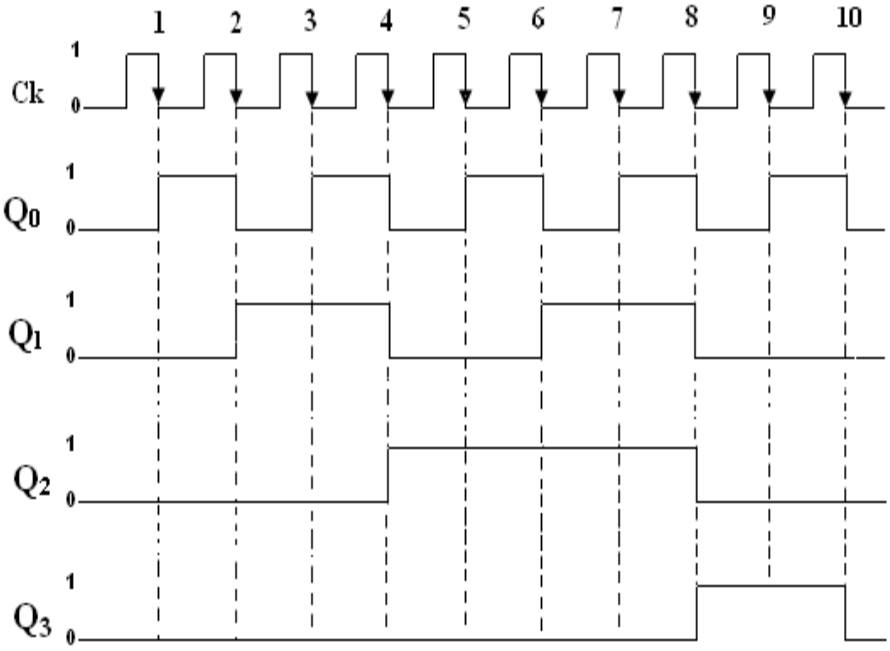
صمم عداد غير متزامن تصاعدي معامل – 16 باستخدام النطاط نوع – T ثم أكتب جدول العد للعداد وإرسم مخطط الإشارة لإخراج كل نطاط .

مـثـال صمم عداد غير متزامن تصاعدي معامل – 10 .

الرقم 10 ليس من الأرقام المتكونة من 2^N وهي بين الرقم 8 والرقم 16 لذلك سوف نقوم بتصميم دائرة عداد معامل – 16 ثم نضيف دائرة لتصفير العداد بعد حالة العد العاشرة وكما مبين في الدائرة أدناه .



Ck	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0
1	1	0	0	0
2	0	1	0	0
3	1	1	0	0
4	0	0	1	0
5	1	0	1	0
6	0	1	1	0
7	1	1	1	0
8	0	0	0	1
9	1	0	0	1
10	0	0	0	0



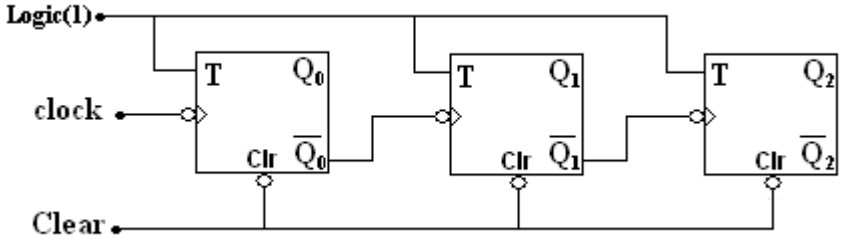
تمارين:

1- صمم عداد غير متزامن تصاعدي معامل – 21 .

2- صمم عداد غير متزامن تصاعدي معامل – 60 .

مـثـال لتصميم عداد غير متزامن تنازلي معامل – 8

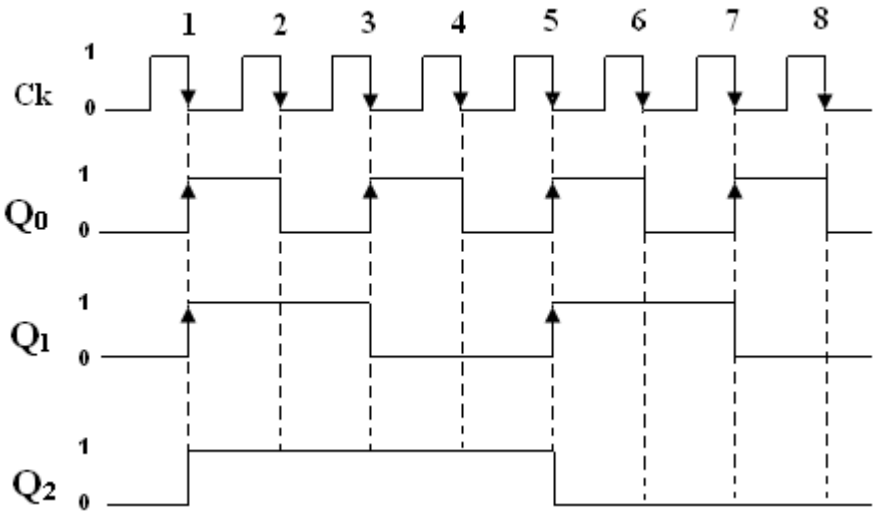
إن الفرق بين تصميم العداد غير المتزامن التصاعدي والتنازلي هو إن ربط إدخالات نبضات الساعة لكل نطاط الى إخراج $\overline{Q}$ بدل **Q** للنطاط الذي يسبقه.



.

Clear	Ck	Q_0	Q_1	Q_2
0	X	0	0	0
1	1	1	1	1
1	2	0	1	1
1	3	1	0	1
1	4	0	0	1
1	5	1	1	0
1	6	0	1	0
1	7	1	0	0
1	8	0	0	0

(جدول العد للعداد)

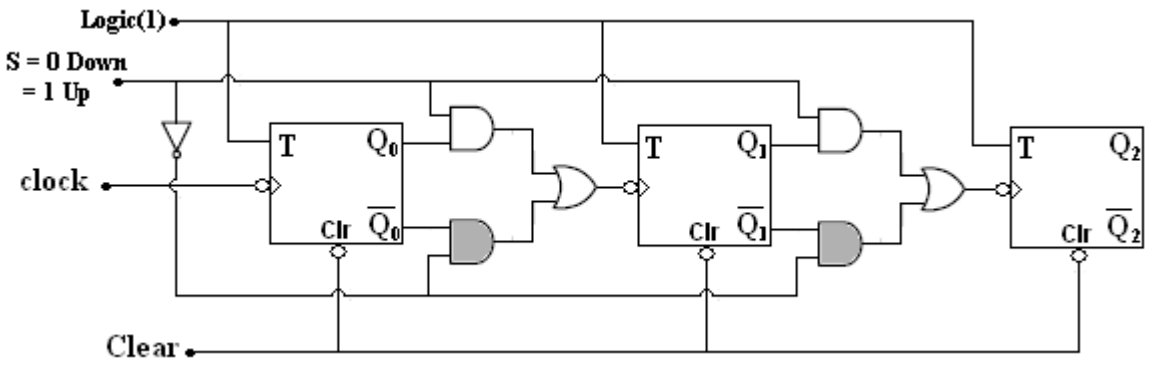


(مخطط الإشارة للعداد التنازلي)

2.4.5 العداد غير المتزامن تصاعدي/ تنازلي: (Asynchronous Up/ DownCounter)

في بعض التطبيقات نحتاج الى عداد يعد بالإتجاه التصاعدي والتنازلي في نفس الوقت. في مثل هذه الحالة نحتاج الى وضع دائرة إختيار (Multiplexer) بين كل نطاطين لربط إدخال نبضات النطاطات مرة مع **Q** ليعمل العداد بشكل تصاعدي ومع $\overline{\mathbf{Q}}$ مرة أخرى ليعمل بشكل تنازلي. دائرة الإختيار هذه تتكون من بوابتين AND مربوطة إخراجاتها الى بوابة OR . بوابات AND يسيطر عليها خط (S) , بحيث عندما يكون (S = 0) يعمل العداد بشكل تنازلي و عندما يكون (S = 1) يعمل العداد بشكل تصاعدي.

مـثـال لتصميم عداد غير متزامن تصاعدي / تنازلي معامل – 8.



Clear	Ck	S = 1 تصاعدي			S = 0 تنازلي		
		Q_0	Q_1	Q_2	Q_0	Q_1	Q_2
0	X	0	0	0	0	0	0
1	1	1	0	0	1	1	1
1	2	0	1	0	0	1	1
1	3	1	1	0	1	0	1
1	4	0	0	1	0	0	1
1	5	1	0	1	1	1	0
1	6	0	1	1	0	1	0
1	7	1	1	1	1	0	0
1	8	0	0	0	0	0	0

1.4.5 العدادات المتزامنة: (Synchronous Counters)

كما بينا سابقاً إن من مميزات العدادات غير المتزامنة هو بطئ العد بسبب عدم تزامن دوائر النطاطات المكونة لها في العمل , لذلك فإن العدادات غير المتزامنة سوف لاتعمل في التطبيقات التي تحتاج الى عملية عد سريعة جداً. بينما تمتاز العدادات المتزامنة بسرعة العد وذلك لأن دوائر النطاطات فيها تعمل جميعها في وقت واحد , حيث أن إدخال النبضات (Clock) سيوصل بكل النطاطات .

من مميزات العدادات المتزامنة :

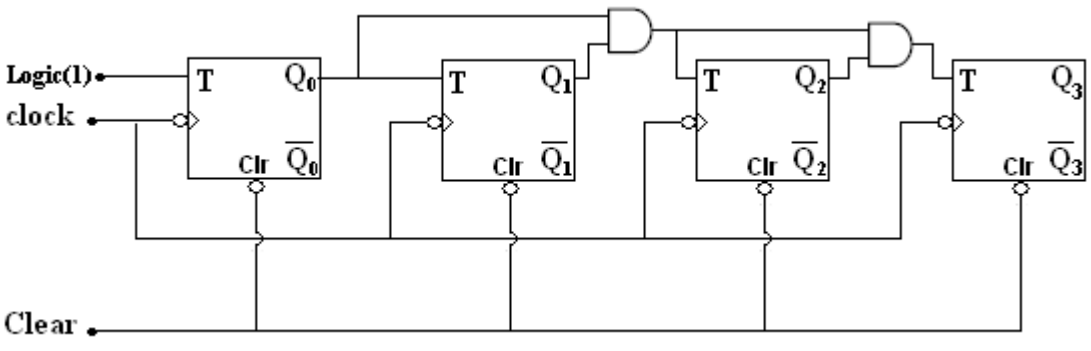
1- صعوبة التصميم.

2- يمكن تصميم عداد يعد أي حالات عد حتى لو كانت ثنائية غير متسلسلة.

3- دوائر النطاط تعمل جميعها في وقت واحد.

4- سريعة العد مقارنة بالأنواع الأخرى.

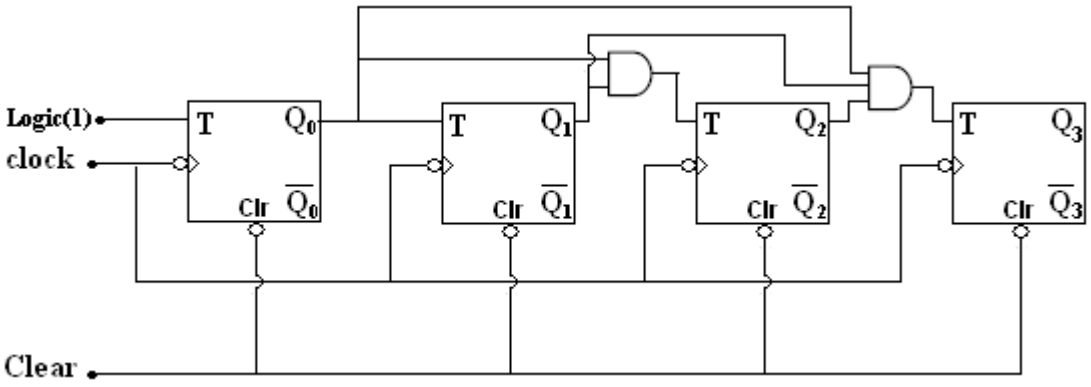
مثال لتصميم عداد متزامن توالي معامل – 16.



كما نلاحظ من الدائرة أن دوائر النطاطات تعمل جميعها على نفس نبضات الساعة وفي نفس الوقت ، ولكن إخراج النطاط لايمكن أن يتغير الا إذا أصبح إدخال T لذلك النطاط يساوي (1) وفي خلاف تلك الحالة يبقى اخراج النطاط على نفس الحالة السابقة أو بدون تغيير.

Clear	Ck	Q_0	Q_1	Q_2	Q_3
0	0	0	0	0	0
1	1	1	0	0	0
1	2	0	1	0	0
1	3	1	1	0	0
1	4	0	0	1	0
1	5	1	0	1	0
1	6	0	1	1	0
1	7	1	1	1	0
1	8	0	0	0	1
1	9	1	0	0	1
1	10	0	1	0	1
1	11	1	1	0	1
1	12	0	0	1	1
1	13	1	0	1	1
1	14	0	1	1	1
1	15	1	1	1	1
1	16	0	0	0	0

مـثـال لتصميم عداد متزامن توازي معامل – 16.



مـلاحظة : جدول العد للعداد المتزامن التوازي هو نفس جدول العد للعداد التوالي ولكن الفرق هو إن عد العداد التوازي أسرع من عد العداد التوالي وذلك لإن إدخالات T لكل نطاط تأخذ الإشارة مباشرة من إخراجات النطاطات التي تسبقها.

تمرين:

صمم عداد متزامن توازي معامل – 32 .

الفصل السادس
محولات الرقمي الى تناظري
و محولات التناظري الى رقمي
Digital to Analog & Analog to Digital Converters

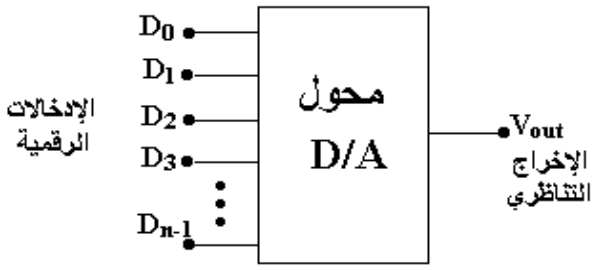
1.6 مقدمة :

تعد جميع الظواهر الفيزياوية ذات طبيعة تناظرية (مثل الجهد , الحرارة , الضغط ... الخ), ولما كان التعامل مع هذه الظواهرضمن المبادئ الرقمية له فائدة كبيرة , برزت الحاجة الى عملية تحويل هذه المتغيرات من شكلها التناظري الى الرقمي (A/D) وعملية التحويل من الشكل الرقمي الى التناظري(D/A) .

ومن الأمثلة على الأنظمة التي تستخدم هذا النوع من التحويل , نظام رقمي لعرض والسيطرة على الحرارة , جهاز الفولتميتر الرقمي .

2.6 التحويل الرقمي الى التناظري: "Digital to Analoge Conversion D/A"

يكون الإدخال للمحول الرقمي الى التناظري (D/A) هو عدد N من الإدخالات الرقمية على شكل إشارات ثنائية وتكون على شكل الفولتية المساوية لمنطق (0) أو منطق (1) , وتكون هذه الفولتية غير قادرة على سوق المحول , لذلك تستخدم تستخدم هذه الفولتية لتشغيل مفاتيح سيطرة رقمية . والإخراج للدائرة تكون فولتية تناظرية يمكن تحويلها الى شكل المتغير الفيزياوي المطلوب بدوائر تحويل أخرى.



يمكن حساب الفولتية التناظرية V_{out} لمحول D/A ذات عدد N من الإدخالات الرقمية كما في المعادلة التالية :

$$V_{out} = K(2^{N-1}b_{N-1} + 2^{N-2}b_{N-2} + \cdot\cdot\cdot\cdot\cdot\ 2^{1}b_{1} + 2^{0}b_{0}) \qquad .... (1)$$

حيث :

K = هو ثابت التناسب .

N = عدد الإدخالات الرقمية .

$b_n = 1$ عندما تكون قيمة المرتبة n تساوي 1 .

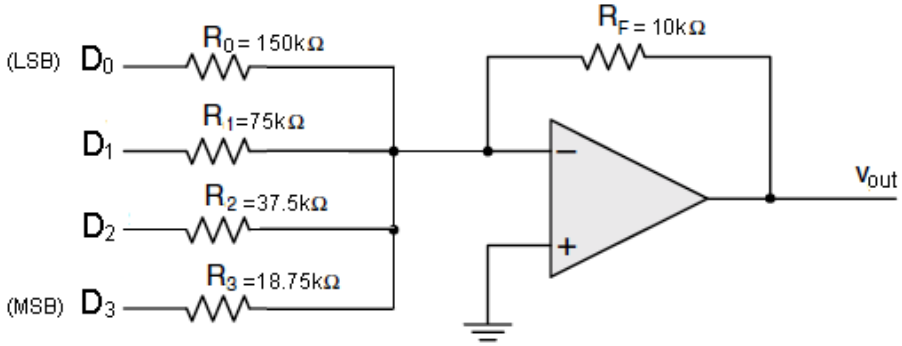
$b_n = 0$ عندما تكون قيمة المرتبة n تساوي 0 .

هنالك نوعان شائعان من محولات D/A هما :

أ - **محول D/A ذات المقاومة الموزونة** " Weighted – Resistor D/A Converter"

الشكل أدناه يمثل محول D/A ذات أربعة إدخالات رقمية نوع المقاومة الموزونة حيث أن الشبكة للمقاومات يجب أن تحوي على مقاومات ذات وزن وكما يلي :

$$R_1 = \frac{1}{2}R_0 \qquad R_2 = \frac{1}{2}R_1 \qquad R_3 = \frac{1}{2}R_2$$



وترتبط المقاومة R_0 بالمرتبة الأقل قيمة (LSB) للإدخالات الرقمية. التيار المتولد من هذه الشبكة يتناسب مع الإدخال الرقمي , هذا التيار يمكن تحويله الى فولتية مناسبة بإستخدام مكبر العمليات.

يمكن حساب الفولتية الخارجة V_{out} لهذا المحول بإستخدام المعادلة (1) بعد حساب قيمة K من المعادلة التالية :

$$K = -\frac{R_f}{R_0}V(1)$$

حيث : V(1) = تعني فولتية منطق (1).

مثال:

في دائرة محول D/A نوع المقاومة الموزونة المبينة في الشكل أعلاه إذا كانت قيمة V(1) = 5V , $R_0 = 150k\Omega$, $R_f = 10k\Omega$. إحسب قيمة V_{out} عند الإدخال الرقمي (1011) , ثم إحسب أعلى فولتية خارجة للمحول.

$$V_{out} = -K(2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0)$$

$$= \frac{10k\Omega}{150k\Omega} \times 5(8 \times 1 + 4 \times 0 + 2 \times 1 + 1 \times 1)$$

$$= -\frac{1}{3}(8 + 0 + 2 + 1)$$

$$= -\frac{1}{3}(11) = \mathbf{-3.66\ V}$$

$$V_{out_{max}} = -\frac{1}{3}(15) = \mathbf{-5V}$$

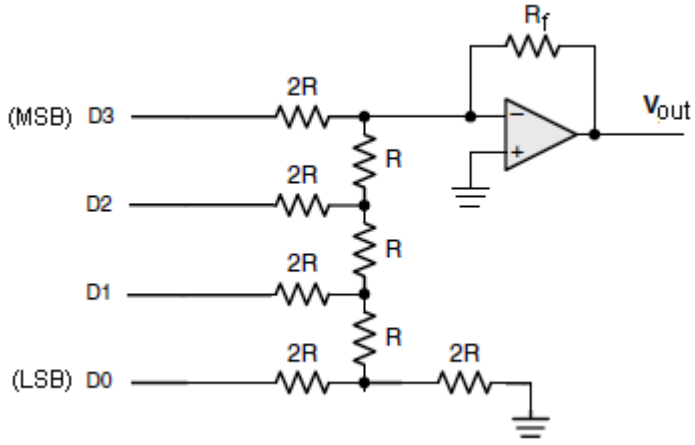
تمرين:

أكتب جدول يبين كافة إحتمالات العمل للدائرة في المثال السابق مع قيم V_{out} لكل إحتمال.

ب - محول D/A نوع السلم الثنائي "Binary Ladder D/A Converter"

في النوع المقاومة الموزونة من محولات D/A , رغم سهولة تصميمه الا إنه غير مرغوب في التطبيقات العملية التي تحتاج الى قيم إخراج دقيقة , وذلك بسبب الصعوبة في الحصول على قيم مقاومات دقيقة كما في الحسابات النظرية , وكذلك يجب أن تكون نسبة الخطأ في هذه المقاومات قليلة جداً إذا أردنا الحصول على نتائج دقيقة وموثوقة.

يمكن تلافي سلبيات محول المقاومة الموزونة بإستخدام محول نوع السلم الثنائي. الشكل أدناه يبين هذا النوع من المحولات ذات أربعة إدخالات رقمية. حيث نلاحظ أن شبكة المقاومات تحتوي على قيمتين فقط للمقاومات , المقاومات العمودية R والمقاومات الأفقية 2R .



ويمكن حساب الفولتية الخارجة للمحول V_{out} من المعادلة (1) أيضاً بعد حساب قيمة الثابت K وكما يلي :

$$K = \frac{V(1)}{(2^n - 1)}$$

شرط أن يكون $R_f = 2R$, $V(0) = 0V$

مثال:

دائرة محول D/A نوع السلم الثنائي ذات ستة إدخالات رقمية إذا كانت قيمة $V(1) = +5V$ و $V(0) = 0V$, $R_f = 2R$. إحسب قيمة V_{out} عند الإدخال الرقمي (101100).

$$V_{out} = -K(2^5 b_5 + 2^4 b_4 + 2^3 b_3 + 2^2 b_2 + 2^1 b_1 + 2^0 b_0)$$

$$= -\frac{5}{2^6 - 1}(32 \times 1 + 16 \times 0 + 8 \times 1 + 4 \times 1 + 2 \times 0 + 1 \times 0)$$

$$= -\frac{5}{63}(32 + 0 + 8 + 4 + 0 + 0)$$

$$= -\frac{5}{63}(44) = \mathbf{-3.49\ V}$$

3.6 الدقة والوضوح في محولات D/A "Accuracy and Clearness in D/A Convertors"

الدقة : هي مقياس مدى قرب فولتية الإخراج الحقيقية من الفولتية المحسوبة نظرياً , وتعتمد الدقة على قيم المقاومات المستخدمة وكذلك على دقة فولتية المصدر . و لحساب الدقة نستخدم المعادلة التالية :

$$\text{الدقة} = \frac{\text{القيمة النظرية} - \text{القيمة العملية}}{\text{القيمة العملية}} \times 100\%$$

فلو فرضنا أن القيمة النظرية هي 5V والقيمة العملية 4.5V فإن الدقة $= \frac{5-4.5}{5} \times 100\% = 10\%$

الوضوح :هي أقل زيادة في فولتية المحول التي يمكن تمييزها . ويعتمد الوضوح بصورة رئيسية على عدد الإدخالات الرقمية للمحول . يمكن حساب الوضوح بشكلين

أما الوضوح $= \frac{1}{2^N}$ حيث N يمثل عدد الإدخالات الرقمية للمحول.

أو الوضوح بالفولت $= \frac{\text{أقصى فولتية إخراج}}{2^N}$

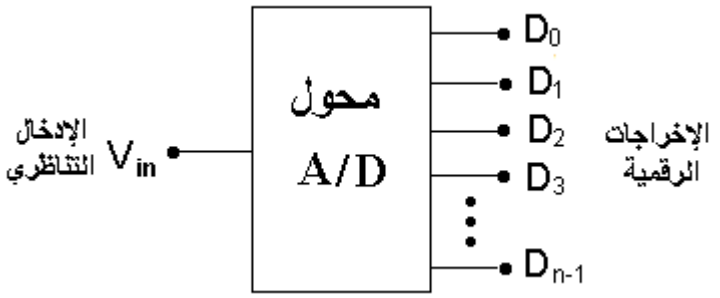
فلو فرضنا أن عدد الإدخالات الرقمية للمحول هي 8 وإن أقصى فولتية إخراج هي 4.5V فسيكون

الوضوح $= \frac{1}{256} = \frac{1}{2^8} = 0.0093$

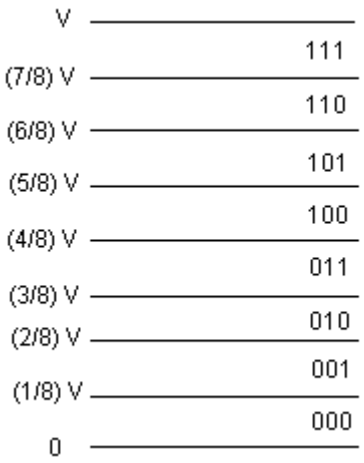
الوضوح بالفولت $= \frac{4.5}{256} = \frac{4.5}{2^8} = 17.55\ \text{mV}$

3.6 التحويل التناظري الى الرقمي: "Analoge to Digital Conversion A/D"

في محولات A/D تكون قيمة الإدخال التناظري أي قيمة ضمن وسط التحويل , لذلك يجب تحويل هذه القيمة في البداية الى فولتية مناسبة لكي تتمكن هذه الفولتية من تشغيل أو سوق دائرة المحول . أما خارج المحول A/D فسيكون على شكل قيم رقمية منفصلة كل قيمة تمثل إخراجا رقمياً .



فلو فرضنا أن القيمة التناظرية محصورة بين قيم (V - 0) ولدينا إخراجاً رقمياً ذات ثلاثة مراتب , فهذا يعني أن المدى الكلي يجب أن يقسم على 8 بين كل قيمة وأخرى $\frac{V}{8}$ وكما يلي :



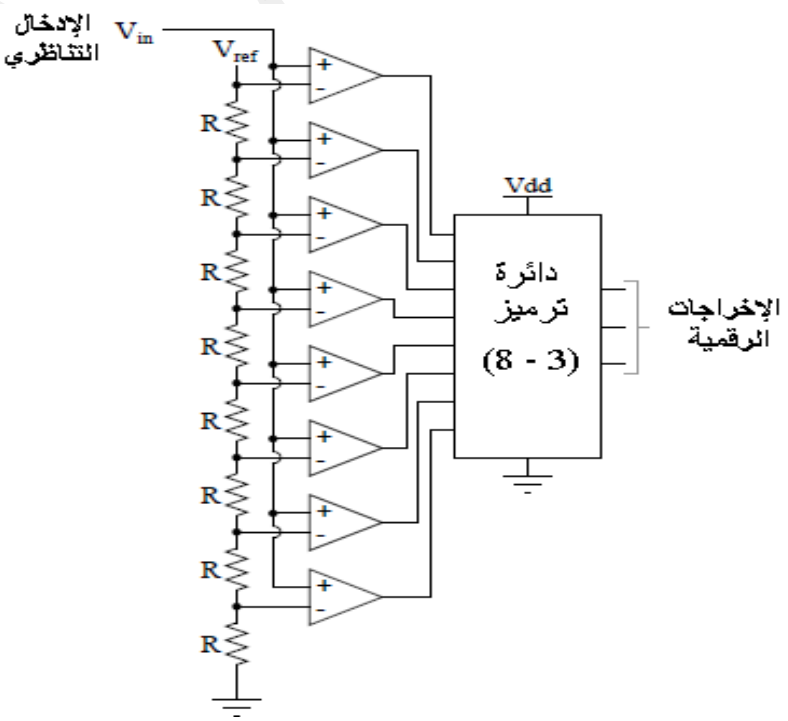
كلما كان الإخراج الرقمي ذات مراتب أكثر يجب أن تكون القراءة أدق ونسبة الخطأ أقل.

هنالك العديد من محولات A/D ولكن سنتطرق الى نوعين أساسيين وهما :

أ- **محول A/D اللحظي** " Flash A/D Converter"

سمي هذا النوع باللحظي لسرعته في التحويل , أيضاً يسمى هذا النوع بالمحول المتوازي , حيث إن دائرة هذا المحول هي الأبسط لفهم عمل محول A/D. تتكون الدائرة من سلسلة من مكبرات العمليات , كل واحد منها يقوم بمقارنة فولتية الإدخال (V_{in}) مع قيمة فولتية محددة , ويكون إخراج هذا المقارن فعالاً إذا كانت فولتية الإدخال مساوية للفولتية المحددة. إخراجات مكبرات العمليات هذه تربط الى إدخال دائرة ترميز (Encoder) والتي بدورها تقوم بإخراج القيمة الرقمية المطلوبة. عدد مكبرات العمليات وحجم دائرة الترميز يعتمد على عدد المراتب الرقمية المطلوبة للإخراج.

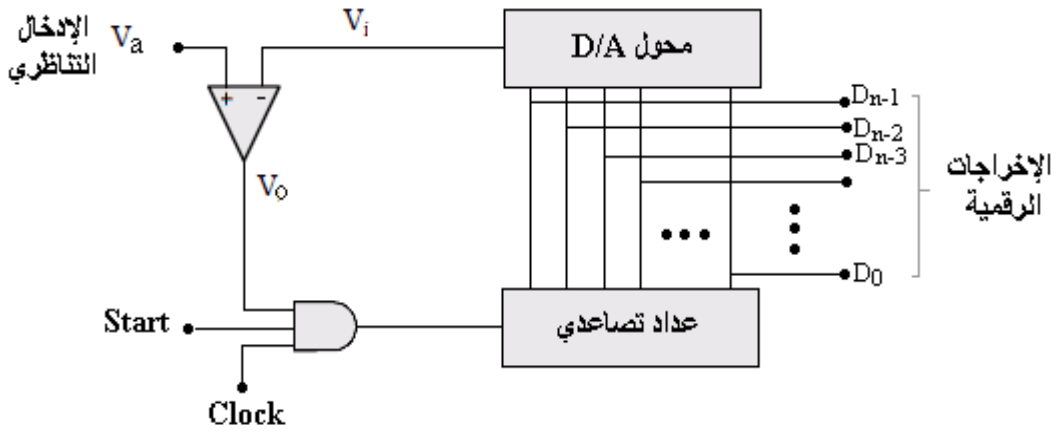
الشكل أدناه يبين محول A/D نوع اللحظي ذات ثلاثة إخراجات رقمية.



ب- محول A/D نوع العداد "Counter Type A/D Converter"

يتكون هذا المحول من عداد تصاعدي ومحول D/A ومقارن مربوطة كما مبينة في الشكل أدناه.

في البداية تكون قيمة خارج المقارن $V_o = 1$ عند وضع الإدخال Start على منطق (0) ثم على منطق(1) حيث تبدأ نبضات المؤقت الدخول الى العداد ويبدأ العداد بالعد بالإتجاه التصاعدي. في كل مرة يعد العداد قيمة معينة , تحول هذه القيمة الرقمية الى فولتية V_i عن طريق المحول D/A . قيمة V_i الناتجة تقارن مع فولتية الإدخال التناظرية V_a المطلوب تحويلها.



فإذا كانت قيمة $(V_a > V_i)$ فإن خارج المقارن $V_o = 1$ لذلك تستمر نبضات المؤقت بالدخول الى العداد ليقوم بعد قيمة جديدة أكبر , وهكذا حتى تصل قيمة $(V_a \leq V_i)$ عندها يتغير إخراج المقارن الى $V_o = 0$ مما يؤدي الى منع دخول نبضات المؤقت الى العداد ويتوقف العد . عندها يمكن قراءة القيمة الرقمية على خطوط الإخراج الرقمي والتي تعادل قيمة الإدخال التناظري V_a .

يمتاز هذا النوع من محولات A/D بالدقة العالية , إلا إنه يستغرق وقت أطول من الأنواع الأخرى في إنجاز عملية التحويل . ويمكن حساب زمن التحويل كما يلي :

$$\text{أعظم زمن تحويل } (T_{max}) = \frac{1}{f} \times 2^n$$

حيث : n = عدد الإخراجات الرقمية.

f = تردد نبضات المؤقت.

مثال:

محول A/D نوع العداد ذات أربعة إخراجات رقمية وتردد نبضات التوقيت 200kHz ، إحسب أعظم زمن تحويل .

$$T_{max} = \frac{1}{f} \times 2^n$$

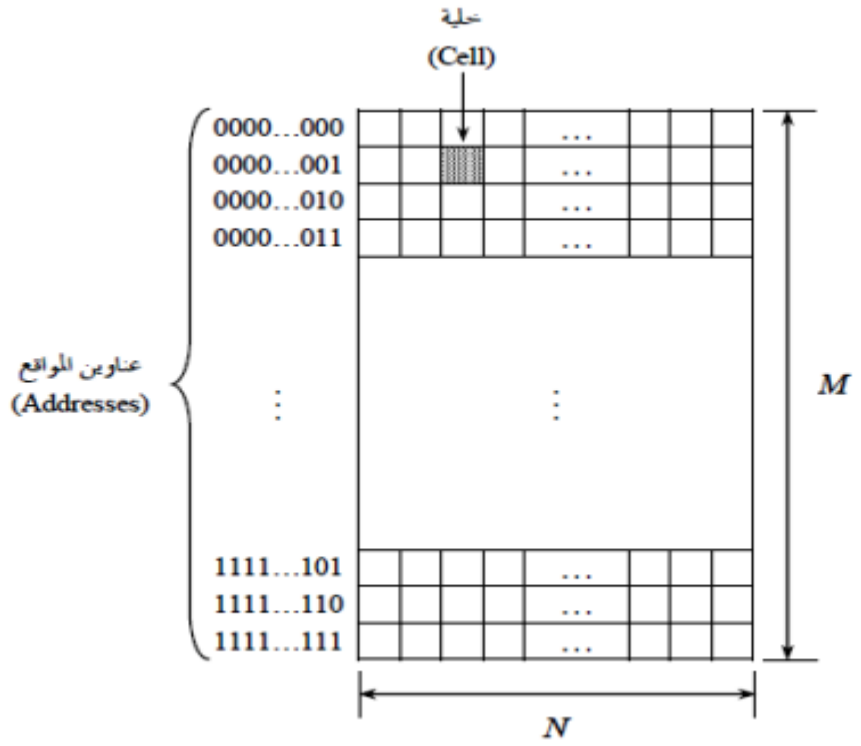
$$T_{max} = \frac{1}{200 \times 10^3} \times 2^4 = 800\ \mu sec$$

الـفـصل الـسـادس
الـذاكـرات و أنـواعـهـا
Memories and There Types

1.7 التنظيم المنطقي للذاكرة :

المقصود بالتنظيم المنطقي للذاكرة هنا هو صورة الذاكرة كما يراها مبرمج النظام الرقمي(Programmer) أي مجموعة من المواقع التخزينية المتتالية المتساوية في الطول، كل موقع منها يتكون من عدد من الخلايا التخزينية(Cells) ، كل خلية منها تستطيع تخزين bit واحد فقط من البيانات و لكل موقع من هذه المواقع عنوان (Address) فريد يميزه عن سواه من المواقع .يمكن الوصول لأي موقع من مواقع الذاكرة عن طريق عنوانه، و ذلك إما لإجراء عملية قراءة(Read) منه أي إسترجاع للبيانات المخزنة فيه، أو عملية كتابة(Write) عليه ، أي تخزين لبيانات فيه . وعملية القراءة غير مدمرة(Nondestructive) لمحتويات الموقع، أي أن الموقع يظل محتفظاً بالبيانات المخزنة فيه كما هي بعدعملية القراءة أما عملية الكتابة فهي مدمرة(Destructive) للمحتويات السابقة للموقع حيث تحل البيانات الجديدة التي تمت كتابتها على الموقع محل البيانات السابقة التي كانت مخزنة فيه.

الشكل التالي يوضح التنظيم المنطقي لذاكرة من نوع $M \times N$ حيث Mهنا هو عبارة عن عدد صحيح يمثل عدد مواقع الذاكرة ويطلق عليه أيضاً طول الذاكرة (Length) و N عبارة عن عدد صحيح يمثل طول الموقع الواحد، أي عدد خاناته الثنائية (bits) أو خلاياه التخزينية ويطلق عليه أيضاً عرض الذاكرة(Width).



2.7 الذاكرة (تعريفها وأنواعها):

تتكون الذاكرة من مجموعة من الدوائرالالكترونية المعقدة، وظيفتها تخزين البرامج والمعطيات التي يستعملها الحاسوب، تصنف الذاكرات من حيث الكتابة والقراءة في نوعين رئيسين هما :

1- ذاكرة القراءة فقط ROM (Read Only Memory)

هذا النوع من الذاكرات كما تدل التسمية يمكن قراءة محتوياتها فقط ولايمكن الكتابة عليها (يقصد بالكتابة خزن المعلومات فيها) تبقى محتفظة بالمعلومات بشكل دائمي حتى بعد فصل التغذية الكهربائية عن الحاسوب. تشحن فيها البرامج من قبل مُصَنِّع الحاسوب ولا يستطيع المستخدم من محو أو تعديل المعلومات المخزونة فيها. تسمح البرامج في الذاكرة ROM بتحقيق أولى مراحل تشغيل الحاسوب عند تزويده بالتيار الكهربائي .

هناك أنواع من ذاكرات القراءة فقط ، يمكن برمجتها مرة واحدة لدى المستخدم تسمى Programmable ROM أو اختصاراً PROM وتحتاج إلى أداة خاصة لبرمجتها . وهناك أيضاً ذاكرات قراءة فقط يمكن مَحو محتواها وإعادة برمجتها عدداً كبير من المرات لدى المستخدم تسمى Erasable PROM أو اختصاراً EPROM وتحتاج إلى أداة خاصة لِمَحو محتواها وبرمجتها (يجري المحو باستخدام الأشعة فوق البنفسجية). وهناك أيضاً ذاكرات قراءة فقط يمكن إعادة برمجتها كهربائياً على نفس الدارة الكهربائية الموضوعة ضمنها خلال عملها وتسمى Electrically EPROM أو اختصاراً EEPROM. وأخيراً هناك الذاكرات الوامضة Flash EEPROM التي يمكن إعادة برمجتها كهربائياً على نفس الدارة الكهربائية ويتم هذا بشكل كتل من المعلومات (كل كتلة 256 بايت والبايت هو ثمانية مواقع خزن).

2- ذاكرات من نوع RAM (Random Access Memory)

هذا النوع من الذاكرات يمكن الكتابة فيها والقراءة منها،وسميت بهذا الإسم دلالة على طريقة الوصول إلى مواقعها المختلفة بطريقة الوصول العشوائي (Random Access)، ولا يشير إلى كونها قابلة للكتابة والقراءة . وتفقد هذه الذاكرات محتواها بمجرد قطع التغذية الكهربائية عنها . ويوجد منها صنفان : ذاكرة ساكنة Static RAM ويرمز إليها بـ SRAM وذاكرات ديناميكية Dynamic RAM ويرمز إليها بـ DRAM . الذاكرات الديناميكية أرخص من الساكنة وذات زمن نفاذ أطول نسبياً . تحافظ الذاكرة الساكنة على محتواها مادامت هناك تغذية كهربائية مطبقة عليها، على حين تفقد الذاكرة الديناميكية محتواها بعد مدة قصيرة ، ولابد من إجراء عملية إنعاش (refreshing) لمحتوياتها من خلال قراءة المحتوى وإعادة كتابته دورياً .