



مدونة المناهج السعودية

<https://eduschool40.blog>

الموقع التعليمي لجميع المراحل الدراسية

في المملكة العربية السعودية

الشامل و الشفاف

# Digital Design

ابن العلاء  
أحمد رمضان العجماني

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

# المقدمة

بسم الله الرحمن الرحيم

الحمد لله رب العالمين والصلوة والسلام على أشرف الأنبياء وسيد المرسلين

سيدنا ونبينا محمد عليه أفضل الصلاة وأتم التسليم      وبعد :

فقد وضعت نصب عيني أثناء كتابتي هذا الكتاب أن يكون شاملًا لكل ما تم دراسته في هذه المادة لكي يستفيد منه كل طالب وطالبة سواء سبق لهم دراسة هذه المادة أم بعد في جامعة أم القرى أو باقي الجامعات داخل وخارج المملكة العربية السعودية.

وكذلك ممكن أن يستفيد منه طلاب مادة مدخل علوم الحاسوب الآلي ومادة عمارة الحاسوب الآلي وذلك لأن هناك بعض المواضيع المقررة عليهم يتحدث عنها ويناقشها ويشرحها هذا الكتاب.

هذا العمل إجتهاد شخصي وأعتمدت فيه بعد الله عز وجل على فهمي لما درسته في هذه مادة  
لذا المهندس / ماهر الشقيري الذي كان خير معلم لها .

وقد كان مرجعى لهذا العمل الكتاب المقرر علي دراسته في هذه المادة

M. Morris Mano **Digital Design**

يستغرق هذا العمل مني قرابة الأربعة أشهر وقد واجهت بعض الصعوبات خصوصاً الرسم حيث أنه كان مجهاً  
للغاية .

قمت بهذا المجهود من أجلي وأجلكم ولكي يكون عونا لكم في دراسة وفهم هذه المادة.

أسأل الله العلي القدير الأجر والمثوبة على هذا العمل وأن يكون خالصاً لوجهه الكريم إنه ولـي ذلك وال قادر  
عليه وصل الله على سيدنا محمد وعلى آله وصحبه وسلم .

أحمد رمضان الزهراني

طالب جامعة أم القرى

قسم علوم الحاسوب الآلي

Ahmad\_911@hotmail.com

# الإهداء

أهدى هذا الكتاب كلية الحاسوب الآلي ونظم المعلومات بجامعة أم القرى  
ممثلة في عميدتها سعادة الدكتور / محمد الصالح  
ووكيلها سعادة الدكتور / سعود مغربي  
ورئيس قسم علوم الحاسوب الآلي سعادة الدكتور / عدنان أبو عرفة  
وجميع أعضاء هيئة التدريس وأخص بالذكر معلمي لهذه المادة المهندس / ماهر الشقيري  
وطلابها وطالباتها  
وكل من يقرأ هذا الكتاب سوف يستفيد منه بإذن الله تعالى

كما أهدي هذا الكتاب لمنتديات كلية الحاسوب ونظم المعلومات [www.uqucs.com](http://www.uqucs.com)  
المنتدى الغالي والعزيز على قلبي الذي سوف يحتضن كتابي وسيحظى بشرف رعاية المنتدى له

# الفهرس

## الفصل الأول Binary Systems

٨	.....Binary Systems
٩	١-١ مقدمة Introduction
٩	١-٢ أنظمة الأعداد Digital Systems
١٠	١-٣ التحويل بين الأنظمة The Conversion Between Numbering Systems
١٠	- من النظام الثنائي إلى النظام العشري From Binary To Decimal
١١	- من النظام الثمانى إلى النظام العشري Octet To Decimal
١٢	- من النظام ست عشري إلى النظام العشري Hexadecimal To Decimal
١٣	- من النظام العشري إلى النظام الثنائى Decimal To Binary
١٥	- من النظام العشري إلى النظام الثمانى Decimal To Octet
١٦	- من النظام العشري إلى النظام ست عشري Decimal To Hexadecimal
١٧	- من النظام الثنائى إلى النظام الثمانى Binary To Octet
١٩	- من النظام الثمانى إلى النظام الثنائى Octet To Binary
٢٠	- من النظام الثنائى إلى النظام ست عشري Binary To Hexadecimal
٢١	- من النظام ست عشري إلى النظام الثنائى Hexadecimal To Binary
٢٢	٤- العمليات على الأعداد الثنائية Operations on Binary Numbers
٢٢	- المتممة الأولى One's Complement
٢٣	- المتممة الثانية Two's Complement
٢٥	- الجمع والطرح Adding & Subtraction

## الفصل الثاني Boolean Algebra And Logic Gates

٢٧	.....Boolean Algebra And Logic Gates
٢٨	٢-١ مقدمة Introduction
٢٨	٢-٢ المنطق الثنائى Binary Logic
٢٨	٢-٣ القواعد Grammars
٢٩	٢-٤ Logics Gates
٣١	- متممة الدالة Complement of a Function
٣٢	-Canonical and Standard Forms
٣٦	-Digital Logic Gates

٣٩	٣-١ مقدمة	Introduction
٣٩	٣-٢ طريقة الخريطة	Map Method
٤٢	٣-٣ خريطة ثلاث متغيرات	Three Variable Map
٤٥	٣-٤ خريطة أربع متغيرات	Four Variable Map
٤٦	٣-٥ Don't Care Conditions	

## الفصل الرابع Combinational Logic

٤٩	٤-١ مقدمة	Introduction
٤٩	٤-٢ إجراء التحليل	Analysis Procedure
٥٠	٤-٣ إجراء التصميم	Design Procedure
٥٤	٤-٤ Half Adder	
٥٥	٤-٥ Full Adder	
٦٢	٤-٦ Decoder	
٦٣	Decoder 2 * 4 -١	
٦٤	Decoder 3 * 8 -٢	
٦٤	Decoder 4 * 16 -٣	
٦٤	Decoder With Enabel	٤-٧
٦٧	Multiplexer	٤-٨
٦٧	Multiplexer 2*1-١	
٦٩	Multiplexer 4*1-٢	
٦٩	Multiplexer 8*1-٣	

## الفصل الخامس .....Synchronous Sequential Logic

٧٥	١-٥ المقدمة
٧٦	.....Introduction
٧٦	.....Types of Flip Flop
٧٦	-١ D Flip Flop
٧٧	-٢ J K Flip Flop
٧٨	-٣ T Flip Flop
٨١	٣-٥ التحليل المؤقت للتسلسل الدائري Analysis of clocked sequential circuits
٨٣	.....State Table
٨٤	.....State Diagram
٨٩	.....State Reduction and Assignment
٩٢	.....Design Procedure

## الفصل السادس .....Registers and Counters

٩٨	٦-١ مقدمة
٩٨	.....Introduction
٩٨	.....Register
٩٨	-١ Shift Register
٩٩	-٢ Rotate Register
١٠٢	.....Counter
١٠٢	-٣ العداد

## الخاتمة

الفضل لله ون

Binary Systems

## ١-١ مقدمة : Introduction

في هذا الفصل سوف نتحدث عن أنظمة الأعداد (Digital systems) وهي : النظام العشري (Decimal system) والنظام الثنائي (Binary system) والنظام الثمانى (Octet system) والنظام الست عشري (Hexadecimal system) والتحويل فيما بين هذه الأنظمة .

كذلك سوف نتحدث عن بعض العمليات التي تتم على الأعداد الثنائية (Operations on binary numbers) وهي: المتممة الأولى (One's complement) والمتممة الثانية (two's complement) والجمع والطرح (Adding & Subtraction) .

## ١-٢ أنظمة الأعداد : Digital Systems

System النظام	Digits الأعداد	Base الأساس
Decimal System النظام العشري	0,1,2,3,4,5,6,7,8,9	10
Binary System النظام الثنائي	0,1	2
Octet System النظام الثمانى	0,1,2,3,4,5,6,7	8
Hexadecimal System النظام الست عشري	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F	16

### ١-٣ التحويل بين الأنظمة : The Conversion Between Numbering Systems

#### ١- من النظام الثنائي إلى النظام العشري : From Binary To Decimal

مثال : Example

حول العدد الثنائي التالي إلى النظام العشري :

$(1011)_2$

الحل : Selution

$$\begin{aligned}(1011)_2 &= 1*2^0 + 1*2^1 + 0*2^2 + 1*2^3 \\&= 1 + 2 + 0 + 8 \\&= (11)_{10}\end{aligned}$$

الشرح : explain

في هذا المثال قمنا بتحويل العدد  $(1011)_2$  من النظام الثنائي إلى النظام العشري حيث قمنا بضرب العدد الأول 1 في 2 ثم نضرب العدد الثاني 1 في  $2^1$  ثم نضرب العدد الثالث 0 في  $2^2$  وأخيراً نضرب العدد الرابع 1 في  $2^3$  ثم نقوم بجمع حاصل ضرب الأعداد السابقة . حاصل عملية الجمع يمثل العدد في النظام العشري = 11

مثال : Example

حول العدد الثنائي التالي إلى النظام العشري :

$(110.1)_2$

الحل : Selution

$$\begin{aligned}(110.1)_2 &= 0*2^{-1} + 1*2^0 + 1*2^1 + 1*2^2 \\&= 0 + 2 + 4 + 0.5 \\&= (6.5)_{10}\end{aligned}$$

الشرح : explain

في هذا المثال العدد الثنائي مكون من جزأين الجزء الأول صحيح والجزء الثاني كسري وعند التحويل نعامل كل جزء على حده .

العدد الصحيح نعمل معه كما تعلمنا في المثال السابق .

أما العدد الكسري يختلف عن العدد الصحيح حيث نقوم بضربه في 2 مرفوعاً للأس السالب .

نرجع للمثال ونقوم بضرب العدد الكسري 1 في  $2^{-1}$  ونجمعه على العدد الصحيح الذي أوجدناه .

أي أننا سوف نعامل العدد على أنه عدد واحد نقوم بضرب العدد الصحيح في العدد 2 مرفوعاً للأس ..3,..2,..1 . والعدد الكسري مرفوعاً للأس ..3,-2,-1 . ثم نجمع العدد كاملاً .

مثال : Example

حول العدد الثنائي التالي إلى النظام العشري :

(1100.101)<sub>2</sub>

الحل : Selution

$$\begin{array}{rcl} \overleftarrow{\overrightarrow{(1100.101)_2}} & = 0*2^0 + 0*2^1 + 1*2^2 + 1*2^3 + 1*2^{-1} + 0*2^{-2} + 1*2^{-3} \\ & = 0 + 0 + 4 + 8 + 0.5 + 0 + 0.125 \\ & = (12.625)_{10} \end{array}$$

٢- من النظام الثماني إلى النظام العشري : Octet To Decimal

مثل طريقة تحويل العدد من النظام الثنائي للعشري ولكن الإختلاف فقط أننا سوف تضرب العدد في العدد 8 الذي يمثل أساس النظام المحول منه بدلاً من العدد 2

مثال : Example

حول العدد الثنائي التالي إلى النظام العشري :

(752)<sub>8</sub>

الحل : Selution

$$\begin{array}{rcl} (752)_8 & = 7*8^0 + 5*8^1 + 2*8^2 \\ & = 7 + 40 + 448 \\ & = (490)_{10} \end{array}$$

مثال : Example

حول العدد الثنائي التالي إلى النظام العشري :

(35.6)<sub>8</sub>

الحل : Selution

$$\begin{array}{rcl} (35.6)_8 & = 5*8^0 + 3*8^1 + 6*8^{-1} \\ & = 5 + 24 + 0.75 \\ & = (29.75)_{10} \end{array}$$

### ٣- من النظام الست عشرى إلى النظام العشري : Hexadecimal To Decimal

لا يختلف هذا التحويل عن التحويل إلى النظائر السابقين إلا في الأساس الذي سوف نضرب العدد فيه وهو 16

مثال : Example

حول العدد الست عشرى التالي إلى النظام العشري :

(ABC)<sub>16</sub>

الحل : Solution

$$(ABC)_{16} = 12 \cdot 16^2 + 11 \cdot 16^1 + 10 \cdot 16^0$$

$$= 12 + 176 + 2560$$

$$= (2748)_{10}$$

مثال : Example

حول العدد الست عشرى التالي إلى النظام العشري :

(2F.8)<sub>16</sub>

الحل : Solution

$$(2F.8)_{16} = 15 \cdot 16^1 + 2 \cdot 16^0 + 8 \cdot 16^{-1}$$

$$= 15 + 32 + 0.5$$

$$= (47.5)_{10}$$

الخلاصة :

عند التحويل من أي نظام (الثنائي أو الثمانى أو الست عشرى) إلى النظام (العشري) فإننا نضرب في أساس النظام المحول منه .

## ٤- من النظام العشري إلى النظام الثنائي : Decimal To Binary

مثال : Example  
حول العدد العشري التالي إلى النظام الثنائي :  
 $(59)_{10}$

الحل : Solution

$$\begin{array}{r} 2 \\ 59 \Big| \\ 29 \quad 1 \\ 14 \Big| \\ 7 \quad 1 \\ 3 \Big| \\ 1 \quad 1 \\ 0 \quad 1 \end{array}$$

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

الشرح : explain

في هذا المثال قمنا بتحويل العدد 59 من النظام العشري إلى الثنائي والطريقة هي القسمة على أساس النظام المحول إليه .

حيث قمنا بقسمة العدد 59 على أساس النظام المحول إليه وهو العدد 2 وكان ناتج عملية القسمة = 29.5

ولكن نحن نريد عدد صحيح فقط بدون كسور نقوم بعد ذلك بضرب العدد الكسري الناتج من عملية القسمة وهو العدد 0.5 في أساس النظام المحول إليه وهو العدد 2 وناتج عملية الضرب وهو العدد 1 سوف يكون باقي عملية القسمة ويكتب في الطرف الثاني على يمين الأعداد ثم نقوم بقسمة ناتج عملية القسمة الأولى 29 على 2 وهكذا مع بواقي نواتج عملية القسمة إلى أن نصل إلى أن يكون ناتج القسمة = 0 والباقي = 1 بواقي عمليات القسمة وهو العدد (111011) يمثل العدد 59 في النظام العشري .

- يوجد بعض الملاحظات التي لابد أن نراعيها أثناء عملية التحويل وهي :
- ١- لو كان ناتج عملية القسمة عدد صحيح بدون كسر كما في المثال السابق
  - $7 = 2 / 14$  ولا يوجد باقي يكون الباقى = 0
  - ٢- عند كتابة ناتج عملية التحويل يكتب العدد من الأسفل إلى الأعلى كما فعلنا في مثالنا السابق .

: Example مثال

حول العدد العشري التالي إلى النظام الثنائي :

$$(0.78125)_{10}$$

: Solution الحل

$$0.78125 * 2 = 1.5625 \rightarrow 1$$

$$0.5625 * 2 = 1.125 \rightarrow 1$$

$$0.125 * 2 = 0.25 \rightarrow 0$$

$$0.25 * 2 = 0.5 \rightarrow 0$$

$$0.5 * 2 = 1 \rightarrow 1$$

$$(0.78125)_{10} = (0.11001)_2$$

: explain الشرح

في هذا المثال قمنا بتحويل العدد (0.78125) من النظام العشري إلى النظام الثنائي طريقة التحويل هي كالتالي :  
كما تلاحظون أن العدد عدد كسري وليس صحيح إذا سوف تتغير طريقة تعاملنا معه خلافاً للمثال السابق  
عند تحويل العدد الكسري نقوم بضرب العدد الكسري في أساس النظام المحول إليه وهو العدد 2  
نضرب العدد (0.78125) في العدد 2 ينتج من عملية الضرب العدد (1.5625) نأخذ الجزء الصحيح 1 ويعتبر هو  
أول عدد ناتج من عملية التحويل ثم نأخذ الجزء الكسري (0.5625) ونكرر معه نفس الخطوات السابقة وهكذا  
إلى أن نصل أن يكون ناتج عملية الضرب عدد صحيح فقط بدون كسور عند ذلك تنتهي عملية التحويل .

يوجد بعض الملاحظات التي لابد أن نراعيها أثناء عملية التحويل وهي :

١ - عند كتابة ناتج عملية التحويل يكتب العدد من الأعلى إلى الأسفل كما فعلنا في مثالنا السابق  
عكس طريقة كتابة تحويل العدد الصحيح .

٢ - نكتب العدد الناتج بعد الفاصلة لأن العدد الذي قمنا بتحويله عدد كسري ولا بد من أن يكون العدد بعد التحويل  
عدد كسري وكما نعلم أن العدد الكسري يكتب بعد الفاصلة .

: Example مثال

حول العدد العشري التالي إلى النظام الثمانى :

$$(35.375)_{10}$$

: Solution الحل

هذا العدد مكون من جزئين صحيح والآخر كسري وعند عملية تحويل إلى النظام الثنائى نعامل كل جزء على  
حدة أي نأخذ الجزء الصحيح ونحوله ثم نأخذ الجزء الكسري ونحوله ونكتب الناتج كما فعلنا في الأمثلة السابقة .

2	
35	0.375 * 2 = 0.75 → 0
17 1	0.75 * 2 = 1.5 → 1
8 1	0.5 * 2 = 1 → 1
4 0	(0.011)
2 0	
1 0	
0 1	
(100011)	

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

## ٥- من النظام العشري إلى النظام الثنائي : Decimal To Octet

مثال : Example  
حول العدد العشري التالي إلى النظام الثنائي :  
 $(153.6875)_{10}$   
الحل : Solution

$$\begin{array}{r}
 & 8 \\
 153 & | \\
 19 & 1 \\
 2 & 3 \\
 0 & 2 \\
 \hline
 (231)
 \end{array}
 \quad
 \begin{array}{l}
 0.6875 * 8 = 5.5 \rightarrow 5 \\
 0.5 * 8 = 4 \rightarrow 4 \\
 (0.54)
 \end{array}
 \quad
 (153.6875)_{10} = (231.54)_8$$

### الشرح : explain

لا تختلف طريقة التحويل العدد من النظام العشري إلى النظام الثنائي عن طريقة تحوليه إلى النظام الثنائي سواء كان العدد صحيح أم كسري ولكن الإختلاف يكون فقط في الأساس الذي سوف نقسم أو نضرب العدد العشري فيه وهو 8 ولكن هناك نقطة سبق وأن تكلمت عنها في طريقة تحويل العدد العشري الصحيح إلى النظام الثنائي ولكن أحببت أن أعيدها لأهميتها وهي :

عند قسمت العدد (153) على 8 ينتج عن عملية القسمة العدد (19.125) نأخذ الجزء الكسري من الناتج (0.125) ونضربه في أساس النظام المحوول إليه 8 ينتج من عملية الضرب العدد 1 ويكون هذا العدد باقي أول عملية قسمة ثم نأخذ الجزء الصحيح وهو العدد (19) ونكرر معه الخطوات السابقة وتنتهي عملية التحويل عندما يكون ناتج القسمة = 0 والباقي = عدد صحيح .

أما طريقة تحويل العدد الكسري العشري إلى ثمانى فهي مثل طريقة تحويله إلى النظام الثنائي تماماً .

## ٦- من النظام العشري إلى النظام ست عشرى Decimal To Hexadecimal

مثل طريقة التحويل إلى الأنظمة السابقة والإختلاف كما وسبق أن ذكرنا في الأساس الذي سوف نضرب أو نقسم عليه العدد العشري وهو العدد 16 وهناك نقطة بسيطة وهي كما سوف تلاحظون في المثال أن باقي عملية قسمة العدد  $= 13$  وهذا العدد لا يكتب بصورته المعروفة وإنما يكتب بصورته في النظام ست عشرى أي نكتب (D).

مثال : Example  
حول العدد العشري التالي إلى النظام ست عشرى :

$(125.34375)_{10}$

الحل : Solution

$$\begin{array}{r} 16 \\ \overline{)125} \\ 7 \quad 13 \\ \overline{)0 \quad 7} \\ (7D) \end{array} \qquad \begin{array}{l} 0.34375 * 16 = 5.5 \rightarrow 5 \\ 0.5 \quad * 16 = 8 \rightarrow 8 \\ (0.58) \end{array} \qquad (125.34375)_{10} = (7D.58)_{16}$$

### الخلاصة :

عند التحويل من النظام العشري إلى أي نظام (الثنائي أو الثماني أو ست عشرى) نتبع الآتي :  
إذا كان العدد العشري الذي نريد تحويله صحيح فإننا نقسم على أساس النظام المحول إليه .  
أما إذا كان العدد عدد كسري فإننا نضرب في أساس النظام المحول إليه .

## ٧- من النظام الثنائي إلى النظام الثماني : Binary To Octet

التحويل من النظام الثنائي إلى النظام الثماني يعتمد على الجدول التالي :

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

مثال : Example

حول العدد الثنائي التالي إلى النظام الثماني :

$(10011101110)_2$

الحل : Solution

$$\begin{array}{cccc} 010 & 011 & 101 & 110 \\ 2 & 3 & 5 & 6 \\ (10011101110)_2 = (2356)_8 \end{array}$$

الشرح : explain

عند التحويل من النظام الثنائي إلى النظام الثماني نقوم بأخذ كل ثلاثة أعداد ثنائية من الناحية اليمنى وذلك لأنه في بعض الأحيان يبقى عدد أو عددين ثنائيين بمفردهما دون الثالث نقوم نحن بإضافة صفر أو صفرتين على يسار الأعداد وكما نعلم أن الصفر لو كان في الخانة اليسرى لا يكون قيمة ونضيف الأصفار لكي يكتمل العدد ويصبح مكون من ثلاثة أعداد كما فعلنا في المثال السابق فلما بإضافة صفر على آخر عددين .

بعد ذلك ننظر في الجدول وننظر للعدد المكافئ لكل ثلاثة أعداد ثنائية من النظام الثماني .

**مثال : Example**  
**حول العدد الثنائي التالي إلى النظام الثمانى :**  
 $(.0101111)_2 = (.234)_8$

**الحل : Solution**

010	111	100
2	3	4
$(.0101111)_2 = (.234)_8$		

**الشرح : explain**

في هذا المثال العدد الثنائي الذي قمنا بتحويله عدد كسري والفرق بين تحويل العدد الكسري عن العدد الصحيح أننا في العدد الصحيح نأخذ كل ثلاثة أعداد من جهة اليمين ونضيف الأصفار على العدد من جهة اليسار.  
 أما في العدد الكسري فإننا نعمل العكس تماماً نأخذ كل ثلاثة أعداد ثنائية من اليسار (أول عدد بعد الفاصله) ونضيف الأصفار على العدد من جهة اليمين .

**مثال : Example**  
**حول العدد الثنائي التالي إلى النظام الثمانى :**  
 $(11001.01)_2 = (31.2)_8$

**الحل : Solution**

011	001	.010
3	1	2
$(11001.01)_2 = (31.2)_8$		

**الشرح : explain**

في هذا المثال العدد الثنائي مكون من جزئين جزء صحيح والأخر كسري ونعامل كل جزء كما تعلمنا في الأمثلة السابقة .

## ٨- من النظام الثماني إلى النظام الثنائي : Octet To Binary

التحويل من النظام الثنائي إلى النظام الثنائي هو عملية عكسية للتحويل من النظام الثنائي إلى الثنائي أي أننا سوف نعتمد على الجدول السابق ونطبق نفس الخطوات السابقة سواءً كان العدد صحيح أو كسري .

مثال : Example

حول العدد الثنائي التالي إلى النظام الثنائي :

(62.7)<sub>8</sub>

الحل : Solution

$$(62.7)_8 = (110\ 010\ .\ 111)_2$$

مثال : Example

حول العدد الثنائي التالي إلى النظام الثنائي :

(35.41)<sub>8</sub>

الحل : Solution

$$(35.41)_8 = (011\ 101\ .\ 100\ 001)_2$$

## ٩ - من النظام الثنائي إلى النظام الست عشري : Binary To Hexadecimal

هي نفس طريقة التحويل السابقة من النظام الثنائي إلى النظام الثمانى ولكن تختلف فقط في أن كل عدد ست عشري يكافئ أربع أعداد ثنائية معتمدين على الجدول التالي :

0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111

8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

مثال : Example  
حول العدد الثنائي التالي إلى النظام الست عشري :

$$(0010\ 1110.\ 1010)_2$$

الحل : Solution  
 $(0010\ 1110.1010)_2 = (2E.A)_{16}$

مثال : Example  
حول العدد الثنائي التالي إلى النظام الست عشري :

$$(1111\ 1100.\ 0101\ 1011)_2$$

الحل : Solution  
 $(1111\ 1100.0101\ 1011)_2 = (FC.5B)_{16}$

## ١٠ - من النظام الست عشرى إلى النظام الثنائى : Hexadecimal To Binary

مثال : Example  
حول العدد الست عشرى التالي إلى النظام الثنائى :

(AB.6D)<sub>16</sub>  
الحل : Selution

$$(AB.6D)_{16} = (1010\ 1011.0110\ 1101)_2$$

مثال : Example  
حول العدد الست عشرى التالي إلى النظام الثنائى :

(9C.8F3)<sub>16</sub>  
الحل : Selution

$$(9C.8F3)_{16} = (1001\ 1100.1000\ 1111\ 0011)_2$$

## ٤- العمليات على الأعداد الثنائية : Operations on Binary Numbers

### ١- المتممة الأولى : One's Complement

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

مثال : Example

أوجد المتممة الأولى للعدد الثنائي التالي :

(1100101001)<sub>2</sub>

الحل : Solution

المتممة الأولى للعدد الثنائي السابق = 0011010110

مثال : Example

أوجد المتممة الأولى للعدد الثنائي التالي :

(1000000000)<sub>2</sub>

الحل : Solution

المتممة الأولى للعدد الثنائي السابق = 0111111111

## ٢- المتممة الثانية : Two's Complement

هذه العملية من أهم العمليات للتي تتم على الأعداد الثنائية ومن خلالها نستطيع أن نقوم بعملية طرح الأعداد الثنائية وغيرها من العمليات .  
المتممة الثانية تقوم بتحويل العدد السالب إلى عدد موجب والعكس وبالتالي نستطيع إجراء عملية الجمع على الأعداد الثنائية إذا قمنا بتحويلها إلى موجبه .

يتم إيجاد المتممة الثانية بإحدى الطرقتين :

### الطريقة الأولى :

وذلك بأن توجد المتممة الأولى للعدد الثنائي ثم تجمع على المتممة الأولى الرقم 1

### مثال : Example

أوجد المتممة الثانية للعدد الثنائي التالي :

(1100101001)<sub>2</sub>

### الحل : Solution

أولاً : نوجد المتممة الأولى للعدد الثنائي السابق = 0011010110

ثانياً : نقوم بجمع الرقم 1 على المتممة الأولى للعدد الثنائي

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

المتممة الثانية للعدد الثنائي السابق = 0011010111

### مثال : Example

أوجد المتممة الثانية للعدد الثنائي التالي :

(1111000000)<sub>2</sub>

### الحل : Solution

المتممة الأولى = 0000111111

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

## الطريقة الثانية:

هذه الطريقة أسهل وأفضل من الطريقة السابقة ولا نحتاج أن نوجد المتممة الأولى للعدد الثنائي وإنما يتم إيجادها بالشكل التالي :

ننظر في العدد الثنائي ونكتبه في الناتج كما هو إلى أن نصل إلى أول رقم 1 في العدد نقوم بكتابته في الناتج كما هو ثم من بعد هذا العدد نقوم بتغيير كل عدد بعده من 0 إلى 1 ومن 1 إلى 0 وسوف نقوم بإيجاد المتممة الثانية للأمثلة السابقة بهذه الطريقة .

مثال : Example  
أوجد المتممة الثانية للعدد الثنائي التالي :

<sup>٢</sup>(1100101001)

الحل : Selution

$$\begin{array}{r} 1100101001 \\ \hline 0011010111 \end{array}$$

المتممة الثانية للعدد الثنائي السابق = 0011010111

## الشرح : explain

في المثال السابق قمنا بإيجاد المتممة الثانية للعدد الثنائي السابق ولأن أول رقم في العدد الثنائي 1 قمنا بكتابته في الناتج كما هو وقمنا بتغيير كل عدد بعده من 0 إلى 1 ومن 1 إلى 0

مثال : Example

أوجد المتممة الثانية للعدد الثنائي التالي :

<sup>٢</sup>(1111000000)

الحل : Selution

$$\begin{array}{r} 1111000000 \\ \hline 0001000000 \end{array}$$

المتممة الثانية للعدد الثنائي السابق = 0001000000

## الشرح : explain

كما تلاحظون قمنا بكتابة أول ستة أرقام في الناتج كما هي إلى أن وصلنا للرقم السابع وهو أول رقم 1 في العدد قمنا بكتابته في الناتج كما هو وغيرنا باقي العدد من بعده كما فعلنا في المثال السابق .

### ٣- الجمع والطرح : Adding & Subtraction

في هذه الجزئية لن أتحدث عن عملية الجمع وإنما سوف يكون اهتمامي فقط بعملية الطرح وذلك لأن عملية الطرح في الأساس ما هي إلا عملية جمع عدد موجب مع عدد سالب . في النظام الثنائي لا يمكن إجراء عملية الطرح مباشرة كما نفعل في النظام العشري بل نقوم بتحويل عملية الطرح إلى جمع باستخدام المتممة الثانية .

مثال : Example  
أجري عملية الطرح التالية :

$$1101 - 0100$$

الحل : Solution

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

$$1101 - 1100 = +1001$$

الشرح : explain

في هذا المثال قمنا بإجراء عملية الطرح على العددين الثنائيين السابقين . العدد الأول 1101 موجب لذلك نقيه كما هو أما العدد الثاني 0100 سالب لذلك نقوم بإيجاد المتممة الثانية له = 1100 ثم بعد ذلك نقوم بعملية الجمع .

ناتج عملية الجمع = 11001 ولكن يوجد به (Overflow) وذلك لأن العدد الأول يمثل في 4 بait والعدد الثاني يمثل في 4 بait والناتج 5 بait ونحن نريد أن يتمثل الناتج في 4 بait لذلك نقوم بحذف آخر عدد في الناتج  $\cancel{1001}$  ونضع إشارة + أمام الناتج .

مثال :Example

أجري عملية الطرح التالية :

0110 - 1100

: Solution الحل

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

0110 - 1100 = -0110

: explain الشرح

هذا المثال لا يختلف عن السابق إلا في أن الناتج لا يوجد به (Overflow) أي أنه يتمثل في 4 بait مثلا العددان لذلك نقوم بإيجاد المتممة الثانية للناتج  $1010 = 0110$  ونضع أمام الناتج إشارة -

الخلاصة :

إذا وجد في الناتج (Overflow) نحذف آخر عدد من الناتج ونضع إشارة +  
إذا لم يوجد في الناتج (Overflow) نأتي بالمتممة الثانية للناتج ونضع إشارة -

الفصل الثاني

Boolean Algebra  
and  
Logic Gates

## ٢-١ مقدمة : Introduction

في هذا الفصل سوف نتحدث عن الدوال (Functions) وكيفية تبسيطها بواسطة الجبر البوليني (Boolean algebra) ولكن لن ننبعق في تبسيط الدوال بهذه الطريقة لأن هنالك طريقة أسهل وأوضحتدعى (karnaugh map) سوف نتعرف عليها في الفصل الثالث بإذن الله تعالى . كما سوف نتحدث عن كيفية رسم الدوال .

وكذلك سوف نتحدث عن روابط (AND , OR , NOT.....) أشكالها وعملها وأيضاً سوف نتحدث عن (Maxterms) و (Minterms) وكيفية إيجادها .

## ٢-٢ المنطق الثنائي : Binary Logic

		AND	OR	NOT	
X	Y	X.Y	X + Y	X	Y
0	0	0	0	1	1
0	1	0	1	1	0
1	0	0	1	0	1
1	1	1	1	0	0

الجدول السابق يوضح أهم الروابط المستخدمة لإنشاء الدوال (Functions) ويوضح (Truth table) لها

## ٢-٣ القواعد : Grammars

OR	
1	$x+1 = 1$
2	$x+x' = 1$
3	$x+x = x$
4	$x+0 = x$
5	$(x')' = x$
6	$x+y = y+x$
7	$x+(y+z) = (x+y)+z$
8	$x.(y+z) = x.y+x.z$
9	$(x+y)' = x'.y'$
10	$x+(x.y) = x$

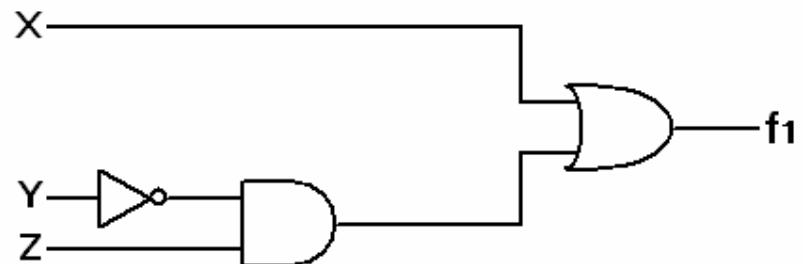
AND	
1	$x.1 = x$
2	$x.x' = 0$
3	$x.x = x$
4	$x.0 = 0$
5	$(x')' = x$
6	$x.y = y.x$
7	$x.(y.z) = (x.y).z$
8	$x+y.z = (x+y).(x+z)$
9	$(x.y)' = x'+y'$
10	$x.(x+y) = x$

هذه القواعد مهمة وتحاجها لتبسيط الدوال (Functions) بواسطة الجبر البوليني (Boolean Algebra) وأهم هذه القواعد القاعدة 9 وهي ما يسمى بقاعدة (De Morgan) .

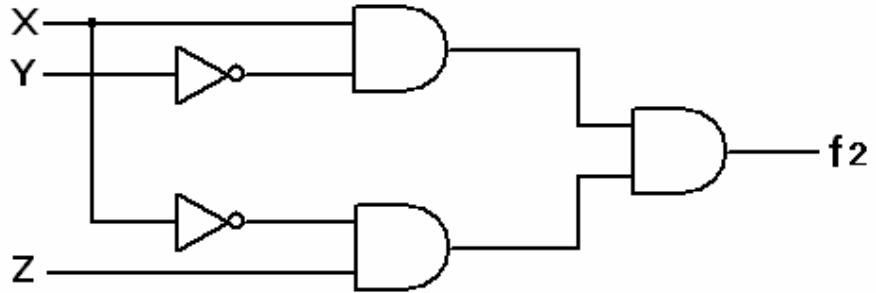
Name	Graphic Symbol	Algebraic Function
AND		$F = xy$
OR		$F = x+y$
Inverter		$F = x'$

الجدول الشكل السابق يوضح كيفية تمثيل الروابط بالرسم وذلك لكي نتمكن من تمثيل الدوال بالرسم .

: Example مثال  
أرسم الدالة التالية :  
 $F_1 = x + y'z$   
: Solution الحل



: Example مثال  
أرسم الدالة التالية :  
 $F_1 = xy' + x'z$   
الحل : Solution



: Example مثال  
بسط الدوال المنطقية التالية  
: Simplify the following Boolean functions

- ١  $x(x' + y)$
- ٢  $x + x'y$
- ٣  $(x + y).(x + y')$
- ٤  $xy + x'z + yz$

الحل : Solution

سوف نقوم بتبسيط هذه الدوال بطريقة الجبر البوليني (Boolean Algebra) معتمدين على قواعد الروابط (AND) و (OR) والتي سبق وأن مرت بنا في هذا الفصل .

$$\begin{aligned} 1- x(x' + y) &= xx' + xy \\ &= 0 + xy \\ &= xy \end{aligned}$$

$$\begin{aligned} 2- x + x'y &= (x+x').(x+y) \\ &= 1.(x+y) \\ &= x + y \end{aligned}$$

$$\begin{aligned} 3- (x+y)(x+y') &= x(x+y') + y(x+y') \\ &= xx + xy' + xy + yy' \\ &= x+xy' + xy + 0 \\ &= x(1+y'+y) \\ &= x1 \\ &= x \end{aligned}$$

$$\begin{aligned} 4- xy + x'z + yz &= xy + x'z + yz.(x+x') \\ &= xy + x'z + xyz + x'y y \\ &= xy(1+z) + x'z(1+y) \\ &= xy + x'z \end{aligned}$$

## ٢-٥ متممة الدالة : Complement of a Function

$$(A + B + C + D)' = A'B'C'D'$$
$$(ABCD)' = A' + B' + C' + D'$$

- سوف نعتمد على القاعدتين السابقتين لإيجاد متممة الدالة وهي ما يعرف بقاعدة (De Morgan) .  
تتم هذه القاعدة بنفي الدالة كاملة وعند نفيها يحدث ما يلي :  
١- تتحول الروابط بين عناصر الدالة من (AND) إلى (OR) والعكس .  
٢- كل عنصر غير منفي يصبح منفي والعكس .

**مثال : Example**

**أوجد متممة الدالة التالية** : find the complement of the following functions

$$F1 = x'yz' + x'y'z$$

**الحل : Seloution**

$$\begin{aligned} F1 &= (x'yz' + x'y'z)' \\ &= (x'yz')' \cdot (x'y'z)' \\ &= (x+y'+z) \cdot (x+y+z') \end{aligned}$$

**الشرح : explain**

كما تلاحظون أن المثال السابق مكون من حدين .

خطوات الحل التالي :

- ١- تغيير الرابط بين الحدين من (OR) إلى (AND) .
- ٢- نفي كل حد على حدة .
- ٣- تغيير الرابط بين عناصر كل حد من (AND) إلى (OR) .
- ٤- نفي كل عنصر مثبت وإثبات كل عنصر منفي .

**مثال : Example**

**أوجد متممة الدالة التالية** : find the complement of the following functions

$$F1 = (x+y'+z') \cdot (x'+y+z) \cdot (x'+y'+z')$$

**الحل : Seloution**

$$\begin{aligned} F1 &= ((x+y'+z') \cdot (x'+y+z) \cdot (x'+y'+z'))' \\ &= (x+y'+z')' + (x'+y+z)' + (x'+y'+z')' \\ &= (x'yz) + (xy'z') + (xyz) \end{aligned}$$

			Minterms		Maxterms	
X	Y	Z	Term	deaignation	Term	deaignation
0	0	0	$x'y'z'$	m0	$x+y+z$	M0
0	0	1	$x'y'z$	m1	$x+y+z'$	M1
0	1	0	$x'yz'$	m2	$x+y'+z$	M2
0	1	1	$x'yz$	m3	$x+y'+z'$	M3
1	0	0	$xy'z'$	m4	$x'+y+z$	M4
1	0	1	$xy'z$	m5	$x'+y+z'$	M5
1	1	0	$xyz'$	m6	$x'+y'+z$	M6
1	1	1	$xyz$	m7	$x+y'+z'$	M7

الشرح : explain

الجدول السابق يوضح كيفية كتابة (Maxterms) (Minterms) لـ (X,Y,Z) بواسطة (Truth table) . و سوف أقوم بعمل مقارنة في الجدول التالي أبين فيها الفرق بين (Minterms) و (Maxterms) .

Maxterms	Minterms
الرابط (OR) هو الذي يربط بين عناصر الدالة	الرابط (AND) هو الذي يربط بين عناصر الدالة ١
عند كتابة عناصر الدالة يكتب المثبت منفي والمنفي مثبت كما هو موضح في الجدول مثل من الجدول (010) كتبت ( $x+y'+z$ ) أي نعمنا بقلب كل 0 إلى 1 والعكس	عند كتابة عناصر الدالة يكتب المثبت مثبت والمنفي منفي كما هو موضح في الجدول مثل من الجدول (010) كتبت ( $x'yz'$ ) أي لم نقم بأي تغيير أثناء كتابتها ٢
يمثل قيمة الدالة عندما تكون قيمة الدالة = 0	يمثل قيمة الدالة عندما تكون قيمة الدالة = 1 ٣
الخلاصة : عند كتابة (Minterms) للدالة يكتب كما هو و عند إيجاد (Maxterms) نعمل نفي (Minterms) لـ (De Morgan)	٤

مثال : Example  
أوجد الجدول التالي Products of Sum و Sum of Products لدوال :

X	Y	Z	F1	F2
0	0	0	0	0
0	0	1	1	0
0	1	0	0	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

الحل : Solution

Sum of Products  $\Leftrightarrow$  Minterms  
Products of Sum  $\Leftrightarrow$  Maxterms

(جمع كميات مضروبة) : Sum of Products

$$\begin{aligned} F1 &= x'y'z + xy'z' + xyz = m1 + m4 + m7 \rightarrow \Sigma(1,4,7) \\ F2 &= x'yz + xy'z + xyz' + xyz = m3 + m5 + m6 + m7 \rightarrow \Sigma(3,5,6,7) \end{aligned}$$

(ضرب كميات مجموعية) : Products of Sum

$$\begin{aligned} F1 &= (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z)(x'+y'+z') = M0.M2.M3.M4 \rightarrow \prod(0,2,3,5,6) \\ F2 &= (x+y+z)(x+y+z')(x+y'+z)(x'+y+z) = M0.M1.M2.M4 \rightarrow \prod(0,1,2,4) \end{aligned}$$

مثال : Example

: Express the Boolean function  $F = A + B'C$  in a sum of minterms

الحل : Selution

هذا طريقة الحل ولكن أرى أن الطريقة الثانية هي الأسهل .

الطريقة الأولى :

$$\begin{aligned} F &= A + B'C \\ &= A(B+B') + B'C \\ &= AB + AB' + B'C \\ &= AB(C+C') + AB'(C+C') + B'C(A+A') \\ &= ABC + ABC' + AB'C + AB'C' + AB'C + A'B'C \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

الشرح : explain

في المثال السابق قمنا بإيجاد (sum of minterms) للدالة  $F = A + B'C$  تلاحظون أن الدالة تحتوي على ثلاثة متغيرات بغض النظر عن أن المتغير مثبت أم منفي وهي : A,B,C وكذلك تلاحظون أن الدالة مكونة من حدين الحد الأول (A) والحد الثاني (B'C) وعندما نريد أن نوجد (sum of minterms) للدالة نعامل كل حد على حدة .

الحد الأول (A) ينقصه المتغيرين (C) و (B) نقوم بإضافة هذه المتغيرات الناقصة للحد ولكن لا نستطيع أن نغیر في المسألة من تلقاء أنفسنا ونضيف الحدود الناقصة مباشرة لذلك نقوم نحن بالتحليل على المسألة ونضرب في 1 ولنا الحق في اختيار القيمة التي = 1 حسب ما نحتاج للتوصيل لحل المسألة لذلك نضرب المتغير A في  $(B+B')$  في  $(C+C')$  ينتج من عملية الضرب حدين وهما  $(AB)$  و  $(A'B)$  نقوم بضربها في  $(C+C')$  وسوف ينتج من عملية الضرب أربعة حدود .

ثم ننتقل للحد الثاني  $(B'C)$  وكما تلاحظون ينقص هذا الحد المتغير (A) نعمل نفس الخطوات السابقة مع الحد الأول بعد ذلك نكتب كامل الحدود الناتجة ونختصر الحدود المتكررة كما فعلنا مع الحدود الملونة بالأحمر .

الطريقة الثانية :

$$\begin{aligned} F &= A + B'C \\ &= [ ABC + ABC' + AB'C + AB'C' ] + [ AB'C + A'B'C ] \\ &= ABC + ABC' + AB'C + AB'C' + A'B'C \end{aligned}$$

الشرح : explain

هي ليست طريقة مختلفة عن الطريقة المتبعة في المثال السابق وإنما هي فقط اختصار لخطوات الحل .  
الحد الأول (A) ينقصه المتغيرين (C) و (B) ونقوم بضربه في هذه الحدود مباشرة بكل احتمالاتهما أقصد أن يكونا مرة مثبتين ومرة المتغير الأول مثبت والآخر منفي ومرة المتغير الأول منفي والآخر مثبت وأخيرا كلاهما منفيين  
الحد الثاني  $(B'C)$  ينقصه المتغير (A) نعمل نفس الخطوات التي قمنا بعملها مع الحد الأول .  
بعد ذلك نكمل بقية الحل كما فعلنا في الطريقة الأولى .

: Example مثال

: Express the Boolean function  $F = xy + x'z$  in a product of maxterms from

: Selution الحل

الطريقة الأولى :

$$\begin{aligned}
 F &= xy + x'z \\
 &= (x+x'z)(y+x'z) \\
 &= (\textcolor{red}{x+x'})(x+z)(y+x')(y+z) \\
 &= \textcolor{red}{1}(x+z)(y+x')(y+z) \\
 &= (x+z)(y+x')(y+z) \\
 &= (\textcolor{red}{x+y+z})(x+y'+z)(\textcolor{blue}{x'+y+z})(x'+y+z')(\textcolor{red}{x+y+z})(\textcolor{blue}{x'+y+z}) \\
 &= (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z') \\
 &= M_0 \cdot M_2 \cdot M_3 \cdot M_4 = \prod(0,2,4,5) = \Sigma(1,3,6,7)
 \end{aligned}$$

: explain الشرح

في المثال السابق قمنا بإيجاد (product of maxterms) للدالة  $F = xy + x'z$  لحل هذه المسألة سوف نعتمد على القواعد السابقة للروابط (AND) و(OR) والتي سبق أن مرت بنا في هذا الفصل وذلك لكي نتمكن من توزيع (AND) و(OR) والعكس .

عند حل هذه المسألة قمنا بتوزيع الحد الأول ( $xy$ ) على الحد الثاني ( $x'z$ ) نتج عن عملية التوزيع توزيع الحدود التالية  $(x+x'z)(y+x'z)$  نعامل كل حد على حدة .

نأخذ الحد الأول ونقوم بتوزيعه ثم الحد الثاني .  
تنتج حدود أخرى من عملية التوزيع ونستمر في توزيع الحدود إلى أن نصل أن يكون كل حد مكون من ثلاثة متغيرات بعد ذلك نقوم باختصار الحدود المتكررة كما فعلنا مع الحدود الملونة .

الطريقة الثانية :

$$\begin{aligned}
 F &= xy + x'z \\
 &= xyz + xyz' + x'yz + x'y'z \\
 &= m_7 + m_6 + m_3 + m_1 = \Sigma(1,3,6,7) = \prod(0,2,4,5)
 \end{aligned}$$

: explain الشرح

كما سبق أن ذكرت أن (Minterms) يمثل قيمة الدالة عندما تكون قيمة الدالة = 1  
(Maxterms) يمثل قيمة الدالة عندما تكون قيمة الدالة = 0 طبعا لنفس الدالة  
إيجاد (sum of minterms) أسهل من إيجاد (product of maxterms)

أقصد بذلك أنه لو طلب منك إيجاد (product of maxterms) قم بإيجاد (sum of minterms)  
وعند إيجادي له (sum of minterms) أكون قد أوجدت الحدود التي تكون عندها الدالة = 1 وهذا ما قمت  
بعملة في هذا المثال حيث نتجت هذه الحدود التي تعطي مجموعة الحل هذه  $\Sigma(1,3,6,7)$   
أذن متممة هذا الحل وهي مجموعة الحل هذه  $\prod(0,2,4,5)$  تعطي (product of maxterms)  
بعد ذلك أكتب هذه المجموعة للحل بطريقة (Maxterms) .

Name	Graphic Symbol	Algebraic Function	Truth Table															
AND		$F = xy$	<table border="1"> <tr> <td>X</td><td>Y</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
OR		$F = x+y$	<table border="1"> <tr> <td>X</td><td>Y</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	1
X	Y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
Inverter		$F = x'$	<table border="1"> <tr> <td>X</td><td>F</td></tr> <tr> <td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td></tr> </table>	X	F	0	1	1	0									
X	F																	
0	1																	
1	0																	
Buffer		$F = x$	<table border="1"> <tr> <td>X</td><td>F</td></tr> <tr> <td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td></tr> </table>	X	F	0	0	1	1									
X	F																	
0	0																	
1	1																	
NAND		$F = (xy)'$	<table border="1"> <tr> <td>X</td><td>Y</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	F	0	0	0	0	1	1	1	0	1	1	1	1
X	Y	F																
0	0	0																
0	1	1																
1	0	1																
1	1	1																
NOR		$F = (x+y)'$	<table border="1"> <tr> <td>X</td><td>Y</td><td>F</td></tr> <tr> <td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td></tr> </table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

XOR	 A logic symbol for XOR. It consists of two input lines labeled 'x' and 'y' entering a single OR gate-like symbol. The output line is labeled 'f'.	$F = xy' + x'y$ $= x \oplus y$	<table border="1" data-bbox="1215 73 1428 327"><thead><tr><th>X</th><th>Y</th><th>F</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																
XNOR	 A logic symbol for XNOR. It consists of two input lines labeled 'x' and 'y' entering a single AND gate-like symbol. The output line is labeled 'f'. A small circle is placed at the output terminal 'f'.	$F = xy' + x'y$ $= x \oplus y$	<table border="1" data-bbox="1215 390 1428 644"><thead><tr><th>X</th><th>Y</th><th>F</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table>	X	Y	F	0	0	0	0	1	0	1	0	0	1	1	1
X	Y	F																
0	0	0																
0	1	0																
1	0	0																
1	1	1																

الفصل الثالث

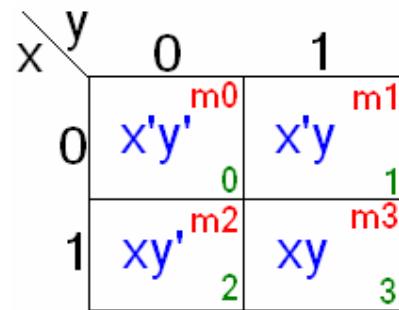
Gate . Level  
Minimization

### ٣-١ مقدمة : Introduction

في هذا الفصل سوف نتحدث عن كيفية تبسيط الدوال (Functions) بواسطة (karnaugh map) وسوف نتعرف عن أشكال (karnaugh map) وكيفية استخدامها .

### ٣-٢ طريقة الخريطة : Map Method

X	Y	Minterms	
0	0	$x'y'$	m0
0	1	$x'y$	m1
1	0	$xy'$	m2
1	1	$xy$	m3



الشكل السابق يوضح الشكل العام لطريقة (karnaugh map) لمتغيرين .  
 تلاحظون أن (X) لها قيمتين 0,1 وكذلك (Y) لها نفس القيم .  
 تقاطع قيم (X) مع قيم (Y) تكون قيم هذه الخريطة .  
 (X) يمثل الجانب العمودي بينما (Y) يمثل الجانب الأفقي .

مثال : Example

بسط الدالة المنطقية التالية : Simplify the following Boolean function

$$F(x,y) = x'y + x'y'$$

الحل : Selution

	y	0	1
x	0	1	1
	1		

$$F(x,y) = x'$$

الشرح : explain

في المثال السابق فلما بتبسيط الدالة  $x'y + x'y'$  وكما تلاحظون الدالة مكونة من حددين . نأخذ الحد الأول ( $x'y$ ) ونلاحظ أنه مكون من تقاطع قيمة (X) وهي القيمة 0 مع قيمة (Y) وهي القيمة 1 نضع في مربع تقاطع قيمة (X) مع قيمة (Y) وهو المربع رقم 1 نضع فيه القيمة 1

الحد الثاني ( $x'y$ ) مكون من تقاطع قيمة (X) وهي 0 مع قيمة (Y) وهي 0 ، نضع في المربع رقم 0 القيمة 1 في الخطوة السابقة فلما بتحديد المربعات التي تمثل قيمة الدالة .

والآن نريد اختصار الدالة :

لدينا مربعين تحتوي على القيمة 1 وبما أن المربعين بجانب بعضهما نأخذهما مع بعضهما لنقوم بإختصار هما ولا نأخذ كل مربع بمفرده لأن الأولوية نأخذ 16 مربع إن لم نستطع نأخذ 8 إن لم نستطع نأخذ 4 إن لم نستطع نأخذ مربعين وأخيراً إن لم نستطع نأخذ مربع واحد

تدرج على هذا التسلسل ولا ننتقل من أولوية إلى الأخرى إلى إذا عجزنا تماماً وإلا سوف يكون تبسيطنا للدالة خاطئ.

نرجع للدالة والإختصار ها أولاً ننظر ماذا تمثل بالنسبة للجانب العمودي وفي هذا المثال لدينا عمودياً فقط (X) ولـ (X) قيمتين 0 وتمثل قيمة 'X' و 1 وتمثل قيمة X وعند الإختصار لابد أن تكون قيم (X) متشابهة أقصد إما أن تكون قيم (X) كلها 0 أو 1 لكي تكتب في الناتج أما إذا كانت مختلفة فإننا نشطب (X) من الناتج تماماً مثل ما حدث مع المتغير (Y) وذلك لأن المربعات واقعة تحت قيمتين مختلفتين للمتغير (Y). ناتج عملية الإختصار = 'X' وذلك لأن المربعات المحتوية على الرقم 1 واقعة تحت قيمة متشابهه للمتغير (X) وهي القيمة 0 ولم نكتب قيمة للجانب الأفقي وهو (Y) وذلك لأن قيمة (Y) مختلفة .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(x,y) = xy + x'y$$

: Selution الحل

	y	0	1
x	0		1
	1		1

$$F(x,y) = y$$

: explain الشرح

في المثال السابق قمنا في البداية بتعبئة المربعات كما تعلمنا في المثال السابق أما بالنسبة لاختصار فقد حدث العكس تماماً للمثال الذي قبله حيث أن قيمة (X) مختلفة مرة 0 ومرة 1 بالنسبة للربعات لذلك لم نكتب قيمة (X) في الناتج أما بالنسبة للمتغير (Y) فإن المربعات واقعة تحت قيمة واحدة له وهي القيمة 1 لذلك كتبنا في الناتج (Y) .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(x,y) = x'y' + xy' + xy$$

: Selution الحل

	y	0	1
x	0	1	
	1	1	1

$$F(x,y) = x + y'$$

: explain الشرح

في المثال السابق قمنا في البداية بتعبئة المربعات حيث قمنا بتعبئة ثلاثة مربعات بالقيمة 1 وذلك لأن الدالة مكونة من ثلاثة حدود وعند إختصارنا لهذه الدالة في البداية ننظر لـ 1 البعيد والذي لا يمكن إختياره إلا مع 1 آخر فقط .

كما تلاحظون 1 الواقع في المربع رقم 2 نستطيع اختياره مع 1 الواقع في المربع رقم 0 وكذلك مع 1 الواقع في المربع رقم 3 لذلك لانننظر إليه الآن وإنما ننظر لـ 1 لأننستطيع أخذه إلا مع 1 آخر وينطبق الشرط السابق على 1 المتواجد في المربع رقم 0 وكذلك المربع رقم 3 طبعاً أنا حبيت ذكر هذه النقطة لأن عند تعاملنا مع ثلاثة وأربع متغيرات سوف تكون العملية أكثر تعقيد في إختيار 1s

بعد إختيار 1s نقوم بعملية التبسيط كما فعلنا في الأمثلة السابقة ونستطيع استخدام 1 أكثر من مرة إذا احتجنا لذلك ولا نختار 1 بمفردة إلا إذا عجزنا تماماً عن وجود 1 بجانبة يمكن اختياره معه .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(x,y) = xy + x'y + xy' + x'y'$$

: Solution الحل

	y	0	1
x	0	1	1
	1	1	1

$$F(x,y) = 1$$

: explain الشرح

في المثال السابق قمنا بتبسيط جميع المربعات الموجودة وعند تبسيطنا لها سوف نقوم بإختيار 4 مع بعضها حسب أولوية اختيار المربعات التي سبق وإن مرت بنا.  
وعند إختيارنا لجميع المربعات يكون ناتج عملية التبسيط = 1

### ٣- خريطة ثلاثة متغيرات :Three Variables Map

x\yz	00	01	11	10
0	$x'y'z'$ 0	$x'y'z$ 1	$x'yz$ 3	$x'yz'$ 2
1	$xy'z'$ 4	$xy'z$ 5	$xyz$ 7	$xyz'$ 6

الشكل السابق يوضح الشكل العام لطريقة (karnaugh map) لثلاثة متغيرات .

(X) يمثل الجانب العمودي بينما (YZ) تمثل الجانب الأفقي .

هناك ملاحظة مهمة وهي :

١- ترتيب المربعات مختلف عن المعتمد حيث أن بعد المربع رقم 1 يأتي المربع رقم 3 ثم المربع رقم 2 وكذلك المربع رقم 5 ثم المربع رقم 7 ثم المربع رقم 6 ، أي الترتيب غير تسلسلي .

٢- شكل (karnaugh map) ليس مستطيلاً يشبه الإسطوانة وأقصد أن المربع رقم 0 ملاصق للمربع رقم 2 وكذلك المربع رقم 4 مع المربع رقم 6 وهذا يعني أن لو كان هنالك 1s في المربعين 0 و 2 نستطيع أن تأخذهما مع بعضهما وذلك لأنهما بجانب بعضهما ، وكذلك المربعين 4 و 6 ينطبق عليهما نفس الكلام السابق .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(x,y,z) = \Sigma(3,4,6,7)$$

: Solution الحل

	$y \backslash z$	00	01	11	10
x	0			1	
	1	1		1	1

$$F(x,y,z) = xz' + yz$$

: explain الشرح

في المثال السابق تغيرت علينا صيغة السؤال حيث قام بإعطائنا أرقام المربعات التي تحتوي على 1s ولم يعطينا حدود كما في الأمثلة السابقة والطريقة الجديدة للسؤال أسهل بكثير عن السابقة .

بعد تعابتنا للمربعات بالقيمة 1 وإختيارنا للمربعات التي سوف تختصرها مع بعضها أريد أن أعيد وأذكر بطريقة الإختصار وذلك لأن أصبح لدينا متغير جديد وهو (Z) .

نأخذ المستطيل الأول المكون من المربعين 4,6 وننظر في البداية ماذا يمثل بالنسبة للجانب العمودي وهو (X) ونلاحظ أن قيمة (X) لا تتغير مع هذا المستطيل حيث أن قيمته = 1 ونكتب في الناتج (X) .

ثم ننظر ماذا يمثل المستطيل بالنسبة للجانب الأفقي المكون (Z , Y ) ونتعامل مع كل متغير على حدة ننظر أولاً ماذا يمثل بالنسبة للمتغير (Y)

ونلاحظ أن قيمة (Y)ختلفت مرة 0 ومرة 1 لذلك نشطب (Y) ولا نكتبه في الناتج لأن قيمته مختلفة ثم ننظر للمتغير (Z) ماذا يمثل هذا المتغير بالنسبة للمستطيل السابقة ونلاحظ أن قيمة (Z) مع المستطيل متشابهة حيث أنها = 0 لذلك نكتب في الناتج (Z') وبذلك يكون قد إنتهينا من أول مستطيل

ثم نأخذ المستطيل الثاني المكون من المربعين 3,7 ونعمل معه مثل ما عملنا مع المستطيل السابق تماماً .

ثم بعد ذلك نكتب ناتج عملية التبسيط ويربط بين كل مستطيل تم تبسيطه مع الآخر علامة +  
الناتج  $xz' + yz$

الحد الأول يمثل إختصار المستطيل الأول والحد الثاني إختصار المستطيل الثاني  
وعلامة + تربط بين الحدود الناتجة عن عملية تبسيط المستطيلين السابقة .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(x,y,z) = \Sigma(0,1,2,4,5,6)$$

: Selution الحل

	$y\backslash z$	00	01	11	10
x	0	1	1		1
	1	1	1		1

$$F(x,y,z) = y' + z'$$

: Example مثال

: Given the Boolean function

$$F(x,y,z) = A'C + A'B + AB'C + BC$$

Express it in sum of minterms - ١

Find the minimal sum of products expression - ٢

: Selution الحل

	$y\backslash z$	00	01	11	10
x	0		1	1	1
	1		1	1	

$$F(x,y,z) = \Sigma(1,2,3,5,7) - ١$$

$$F(x,y,z) = C + A'B - ٢$$

: explain الشرح

في المثال السابق إختلفت علينا صيغة السؤال والمطلوب إيجاده، أعطانا في المثال الدالة وقام بكتابتها على شكل حدود وطلب منك أولاً بعد تعبئة المربعات إيجاد (sum of products) بمعنى آخر كتابة أرقام المربعات التي تحتوي على

القيمة 1s

المطلوب الثاني تبسيط الدالة كما مر بنا سابقاً.

#### ٤-٣ خريطة أربع متغيرات

		yz	00	01	11	10
		wx	m0 w'x'y'z' 0	m1 w'x'y'z 1	m3 w'x'yz 3	m2 w'x'yz' 2
			m4 w'xy'z' 4	m5 w'xy'z 5	m6 w'xyz 7	m7 w'xyz' 6
			m12 wxy'z' 12	m13 wxy'z 13	m15 wxyz 15	m14 wxyz' 14
			m8 wx'y'z' 8	m9 wx'y'z 9	m10 wx'yz 10	m11 wx'yz' 11

الشكل السابق يوضح الشكل العام لطريقة (karnaugh map) لأربع متغيرات .  
 الجديد في الأمر زيادة متغير جديد وهو (W) ليمثل مع (X) الجانب العمودي .  
 نلاحظ أيضاً ترتيب المربعات وكذلك إتصال المربعات مع بعضها البعض من الجانبين مثل (karnaugh map) للأربع متغيرات وكذلك إتصال المربعات من الأعلى والأسفل .  
 بمعنى آخر أن شكل (karnaugh map) لأربع متغيرات ليس مربعاً وإنما تقريباً يشبه المكعب وذلك لإتصال مع وبعضها البعض .  
 إذن (karnaugh map) لأربع متغيرات ليست إلا شكل مكبر لـ (karnaugh map) لثلاث متغيرات .

: Example  
 بسط الدالة المنطقية التالية : Simplify the following Boolean function  
 $F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$   
 : Solution

		yz	00	01	11	10
		wx	1	1		1
			1	1		1
			1	1		1
			1	1		

$$F(w,x,y,z) = y' + w'z' + xz'$$

: explain

المثال السابق لا يختلف عن الأمثلة السابقة وتعاملنا مع (karnaugh map) لمتغيرين أو ثلاثة متغيرات .  
 طريقة الحل واحدة ولا تختلف الجديدة في الأمر زيادة عدد المتغيرات وكبر الحجم ونعاملها مثل ما تعلمنا سابقاً من تعبئة المربعات وطرق اختيار المستويات وكيفية الاختصار .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(w,x,y,z) = \Sigma(0,2,3,5,7,8,9,10,11,13,15)$$

: Seloution الحل

wx \ yz	00	01	11	10
00	1		1	1
01		1	1	
11		1	1	
10	1	1	1	1

$$F(w,x,y,z) = wx' + yz + xz + x'z'$$

لاحظ أنه يمكننا اختيار المربعات الموجودة في الأطراف و اختصارها مع بعضها وذلك لأنها تعتبر مربعات متجاورة

: Don't Care Conditions ٣-٥

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

$$F(w,x,y,z) = \Sigma(0,3,7,11,15)$$

Which has the don't care conditions

$$d(w,x,y,z) = \Sigma(0,2,5,8)$$

: Seloution الحل

wx \ yz	00	01	11	10
00	x	1	1	x
01		x	1	
11			1	
10	x		1	

$$F(w,x,y,z) = w'x' + yz$$

## الشرح : explain

في المثال السابق نلاحظ شيء جديد وهو (Don't care) ويرمز له بالرمز (x).  
نستفيد من (Don't care) أنها تساعدنا في الحل ولكن لا يتوجب علينا تغطية المربعات التي تحتوي على قيمة (Don't care) بالكامل ولكن إذا احتجنا لاستخدامها نستخدمها عكس 1s فإنك إذا لم تقوم بتغطيتها بالكامل فإن حلاً يعتبر خاطئاً.

نرجع للمثال لقد قمنا بتبعدة مربعات (karnaugh map) بقيم 1s ثم تعبيتها بقيم (Don't care) بعد ذلك نقوم بإختيار المربعات ونقوم بعملية الإختصار.  
المربعات نتج لدينا مستطيلين المستطيل الأول المستطيل العمودي المكون من 11,15,15,3 وهذا المستطيل طريقة حله مثل السابق ولن نتطرق إليه.

المستطيل الآخر المستطيل الأفقي المكون من المربعات 3,2,1,0 وهو محور حديثنا  
لو أنه لم يوجد (Don't care) كنا أخذنا المربعين 1,3 وأصبح الحل معقد بعض الشيء  
وبما أنه يوجد (Don't care) فإنها سوف تساعدنا في الحصول على مستطيل أكبر  
وكلما كبر المستطيل كان الحل أكثر اختصاراً.

لكن عملية اختيار المربعات وتكوين مستطيل الإختصار ليست اختيارية وأقصد أن نمشي على التسلسل الذي مر بنا في أولوية اختيار المربعات وإلا فإن الحل خاطئ.

عند وجود (Don't care) يتوجب عليك استخدامها إذا احتجت إليها ولا يجب عليك تغطيتها بالكامل وكما تلاحظون تجاهلنا (Don't care) الموجودة في المربعين 5,8 وذلك لعدم حاجتنا إليها.  
بعد ذلك نكمل عملية الإختصار كما تعلمنا سابقاً.

الفصل الرابع

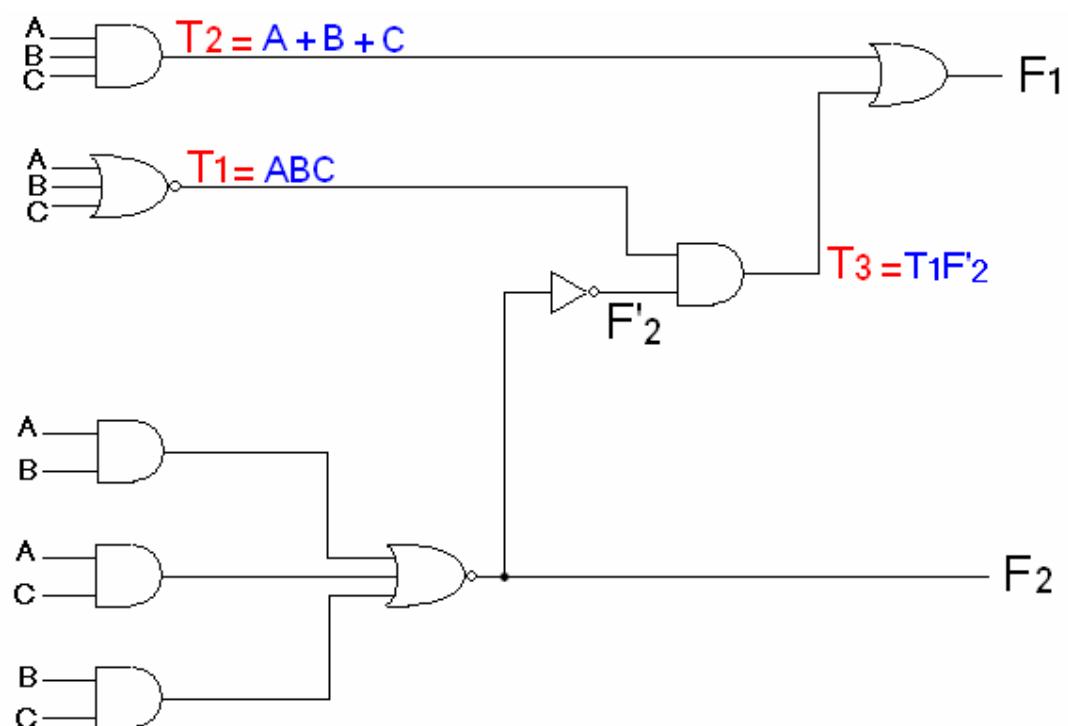
Combinational

Logic

في هذا الفصل سوف نتحدث عن تحليل الدوال (Analysis) وإجاد مخرجاتها . وكذلك سوف نتحدث عن تصميم الدوال (Design) وحل مسائل التصميم . كما سوف نتعرف على بعض الدوال وهي :

(Multiplexer) ، (Decoder) ، (Full Adder) ، (Half Adder) .  
وسوف تحدث عن تعريفها وأشكالها وأنواعها أقسامها وعملها و حل المسائل بواسطتها .

## ٤-٢ إجراء التحليل : Analysis Procedure



$$T_1 = A + B + C$$

$$T_2 = ABC$$

$$F_2 = AB + AC + BC$$

$$T_3 = T_1 F'_2$$

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

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

$$F_1 = T_2 + T_3$$

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

الشرح : explain

الشكل السابق يبيّن شكل دالتين  $F_1$ ,  $F_2$  بالرسم وطلب منك إيجاد مدخلات وحدود كل دالة .  
وكما تلاحظون أن عدد المدخلات كثير وبالتالي قد تحصل هنالك بعض الخبطه ونسیان متغير وبالتالي تضييع درجة سؤال سهل وأنت في غنى عن ضياع درجات على أسئلة سهلة .  
ولتفادي ذلك نقوم بترميز كل رابط يربط حدود الدالة برمز يغنينا عن كتابة كامل مدخلات الرابط  
يفضل أن نقوم بهذه العملية إذا كانت مدخلات الرابط كثيرة بمعنى آخر إذا احتجنا لذلك وخشينا من خلط بين  
المتغيرات أو نسيان متغير .

### ٤-٣: إجراء التصميم : Design Procedure

#### **: خطوات حل مسائل التصميم (Design)**

- ١- معرفة المدخلات والمخرجات وتحديد عددها .
  - ٢- إنشاء (Truth table) لمدخلات ومخرجات الدالة .
  - ٣- تبسيط مخرجات الدالة بأحد الطرق السابقة .
  - ٤- رسم الدالة الناتجة بعد عملية التبسيط .

## مثال : Example

Design a combinational circuit that converts the binary coded decimal (BCD) to the excess-3 code for the decimal digit.

## الحل : Selution

المطلوب من السؤال تصميم دائرة تحول التشفير الثنائي إلى عشري (BCD) بزيدة 3 للرقم العشري .  
بمعنى آخر نقوم بإنشاء (Truth table) لـ 4 مدخلات بعد ذلك ننظر لكل صف مكون من قيم المتغيرات A,B,C,D ونضيف (نجمع) عليه الرقم 3 ليخرج لنا أول صف من المخرجات لكل من قيم المتغيرات W,X,Y,Z مثل :

- ١- أول صف من المدخلات قيمته (0 0 0 0 0) قمنا بإضافة الرقم 3 عليه ليعطينا المخرجات (1 0 0 1 1)

٢- الصف الخامس من المدخلات قيمته (0 1 0 0 1) قمنا بإضافة الرقم 3 عليه ليعطينا المخرجات (7 0 1 1 1)

## الخطوة الأولى والثانية :

وفي هذه الخطوتين أولاً قمنا بتحديد المدخلات والمخرجات ومن ثم إنشاء (Truth table) لاحظ أن المخرجات أعداد عشرية (BCD) وكما نعلم أن آخر عدد في النظام العشري هو 9 وأي عدد ينتج قيمة أكبر من الرقم 9 لا يكتب في الناتج لأنه أكبر من أعداد النظام العشري ونكتب عوضاً عن الأعداد الناتجة الأكبر من الرقم 9 (Don't care) كما تشاهدون على آخر 6 صفوف من الجدول التالي :

InPuts				OutPuts			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	1	0	1	0
1	0	0	0	1	0	1	1
1	0	0	1	1	1	0	0
1	0	1	0	x	x	x	x
1	0	1	1	x	x	x	x
1	1	0	0	x	x	x	x
1	1	0	1	x	x	x	x
1	1	1	0	x	x	x	x
1	1	1	1	x	x	x	x

### الخطوة الثالثة :

وفيها نبسط المخرجات الناتجة من الجدول السابق.

W				X					
wx \ yz	00	01	11	10	wx \ yz	00	01	11	10
00					00		1	1	1
01		1	1	1	01	1			
11	X	X	X	X	11	X	X	X	X
10	1	1	X	X	10	1	X	X	X

$W = A + BD + BC$

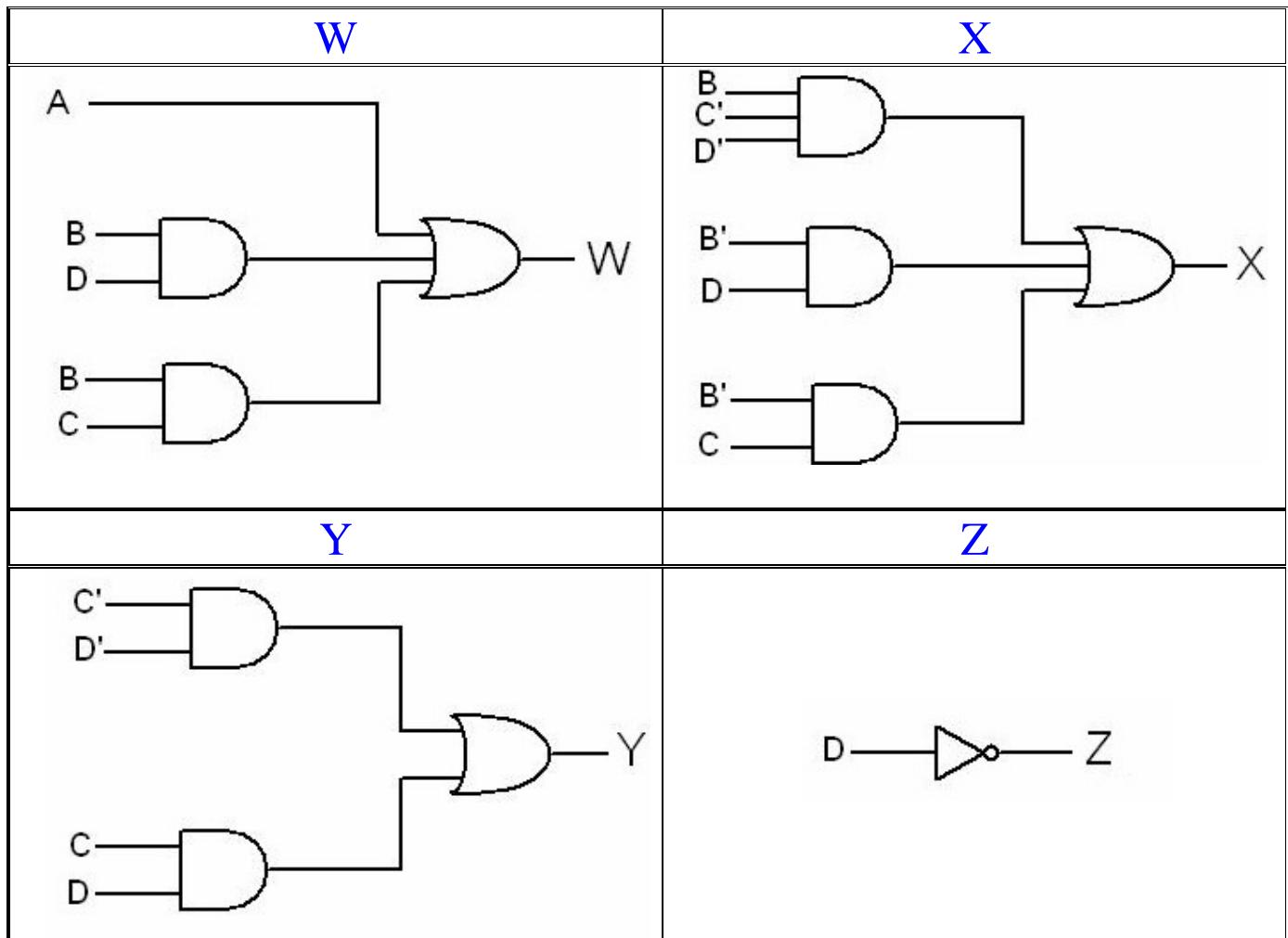
Y				Z					
wx \ yz	00	01	11	10	wx \ yz	00	01	11	10
00	1				00	1			1
01	1				01	1			1
11	X	X	X	X	11	X	X	X	X
10	1				10	1			X

$Y = C'D' + CD$

$Z = D'$

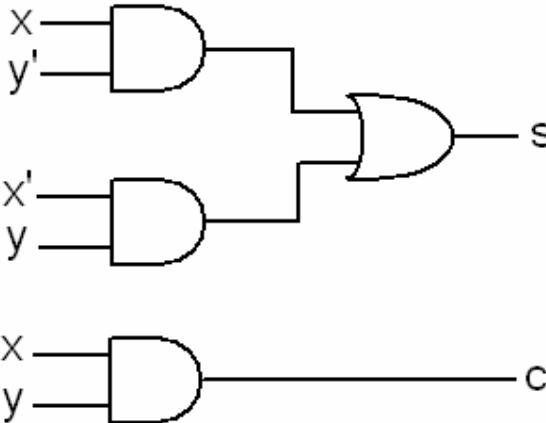
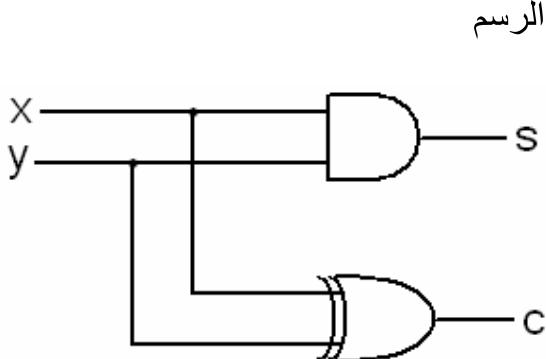
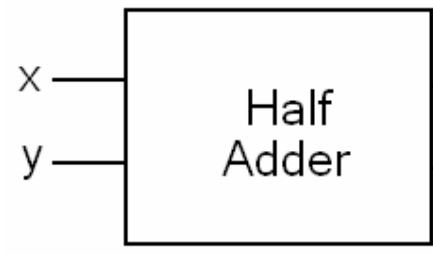
#### الخطوة الرابعة :

نرسم المخرجات التي قمنا بتبسيطها في الخطوة التي قبلها .  
وهذا يدل على أن كل خطوة مرتبطة بالتي قبلها لذلك كن دقيق وحذر أثناء الحل .



## ٤-٤ : Half Adder

سوف نقوم بتعريفها في الجدول التالي :

<b>Truth Table</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2" style="text-align: center;">InPuts</th> <th colspan="2" style="text-align: center;">OutPuts</th> </tr> <tr> <th style="text-align: center;"><b>X</b></th><th style="text-align: center;"><b>Y</b></th><th style="text-align: center;"><b>C</b></th><th style="text-align: center;"><b>S</b></th></tr> </thead> <tbody> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td></tr> <tr> <td style="text-align: center;">0</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td></tr> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">0</td><td style="text-align: center;">0</td><td style="text-align: center;">1</td></tr> <tr> <td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">1</td><td style="text-align: center;">0</td></tr> </tbody> </table>	InPuts		OutPuts		<b>X</b>	<b>Y</b>	<b>C</b>	<b>S</b>	0	0	0	0	0	1	0	1	1	0	0	1	1	1	1	0
InPuts		OutPuts																							
<b>X</b>	<b>Y</b>	<b>C</b>	<b>S</b>																						
0	0	0	0																						
0	1	0	1																						
1	0	0	1																						
1	1	1	0																						
	(S) اختصار لكلمة (Sum) وهي هذه الخانة من الجدول نضع قيمة جمع كل صف من المدخلات أي $X+Y$ (C) اختصار لكلمة (Carry) إذا وجد (Carry) نضع في هذه الخانة 1 وإذا لم يوجد نضع 0 ويوجد (Carry) إذا كان مجموع عملية جمع $X+Y$ أكبر من 1 كما في آخر صف في الجدول																								
<b>Algebraic Function</b>	$S = xy' + x'y$ $C = xy$ <span style="text-align: center;">(A)</span>	$S = x \oplus y$ $C = xy$ <span style="text-align: center;">(B)</span>																							
<b>Graphic Symbol</b>	 <span style="text-align: center;">شكل(A)</span>	 <span style="text-align: center;">الرسم</span> <span style="text-align: center;">شكل(B)</span>																							
		هذا الشكل الذي نحتاجه لحل المسائل بواسطة (Half adder)																							

## : Full Adder ٤-٥

هي شكل مكبر (Half adder) وسوف نتعرف عليها أكثر من خلال الجدول التالي :

**Truth Table**

InPuts			OutPuts	
X	Y	Ci	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

(Carry out) (Co) ، (Carry in) (Ci) اختصار لكلمة

**Algebraic Function**

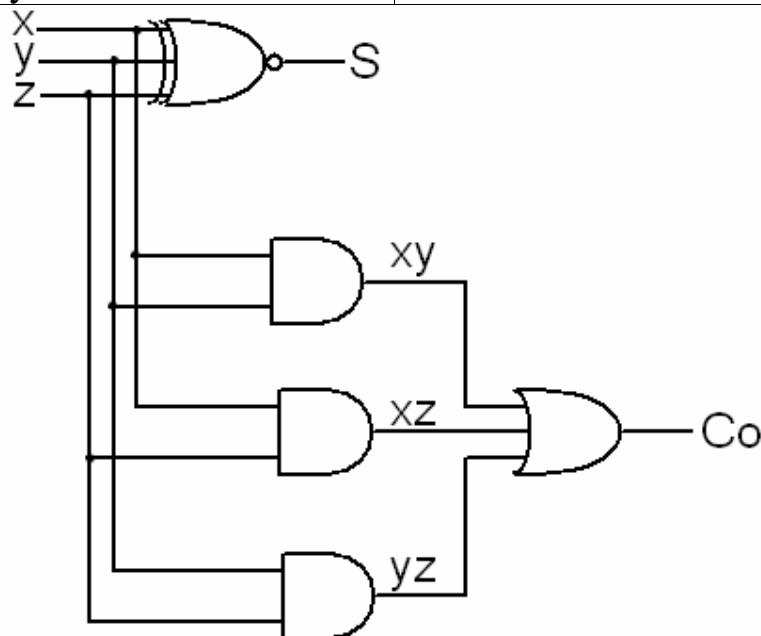
x \ yz	00	01	11	10
0	1			1
1	1		1	

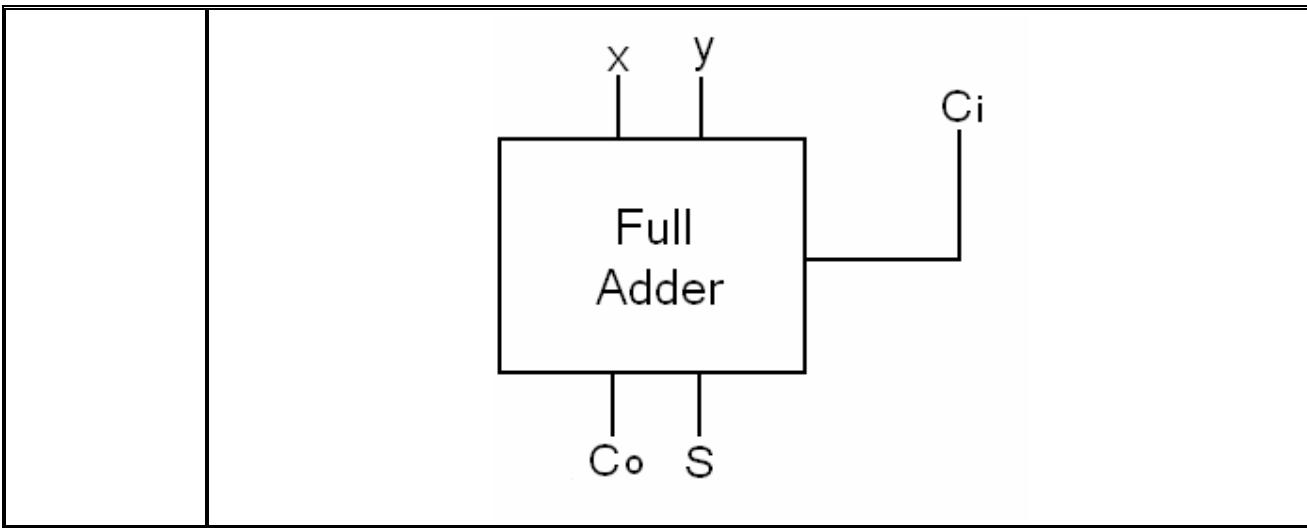
$$\begin{aligned} S &= x'y'z + x'yz' + xy'z' + xyz \\ &= x \oplus y \oplus z \end{aligned}$$

x \ yz	00	01	11	10
0			1	
1		1	1	1

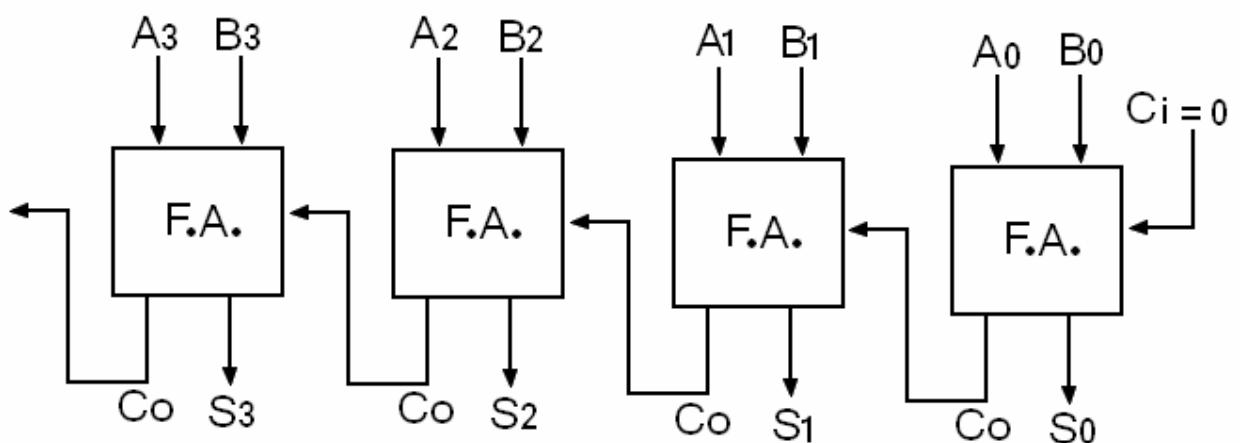
$$C = xy + xz + yz$$

**Graphic Symbol**





**مثال : Example**  
**: Design a 4 - bit full adder**



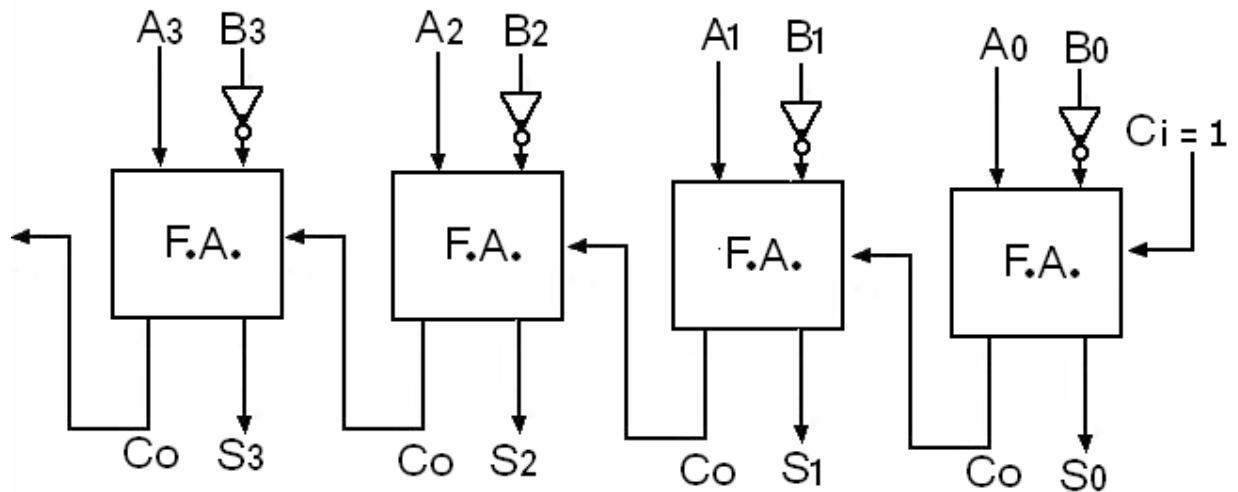
**الشرح : explain**  
في المثال السابق طلب منك تصميم 4 (full adder) وعملها تقوم بعملية الجمع وكما تلاحظون قيمة  $C_i = 0$  شكل مبسط يوضح ما الذي يحدث :

$$\begin{array}{r}
 & 0 \\
 & A_3 \ A_2 \ A_1 \ A_0 \\
 & B_3 \ B_2 \ B_1 \ B_0 + \\
 \hline
 \end{array}$$

: Example مثال

: Design a 4 - bit full subtractor using full adder and additional gates

: Solution الحل



: explain الشرح

في المثال السابق طلب منك تصميم 4 (full adder) وعملها تقوم بعملية الطرح وكما تلاحظون قيمة  $C_i = 1$  (Ci) للقيام بعملية الطرح نحتاج لـ (Two's Complement) .

العلمية التالية توضح كيف تمكنا من تصميم (Two's Complement) لكي تعمل الدالة بشكل صحيح :

$$\begin{aligned}
 &= A + (B' + 1) \\
 &= A + (2's \text{ Comp of } B) \\
 &= A - B
 \end{aligned}$$

: Example مثال

: Design a 2 - bit binary multiplier : تصميم دالة ضرب بعمليه الضرب لقيم المدخلات ووضع الناتج في المخرجات.

: Selution الحل

المطلوب تصميم دالة (multiplier) عمل هذه الدالة تقوم بعمليه الضرب لقيم المدخلات ووضع الناتج في المخرجات.

نرجع للمثال ونلاحظ أن المدخلات متغيرين (A وB)

(A0 و A1) تعتبر رقم واحد وكذلك (B0 و B1)

هذه الدالة تقوم بضرب قيمة  $A * B$  ووضع الناتج في خانة المخرجات ويكتب الناتج طبعاً بالنظام الثنائي مثل :

١- ثالث صف من المدخلات قيمة (0)  $A = 0$  مضروبة في قيمة (2)  $B = 10$  وكان الناتج عملية الضرب = 0

لذلك نكتب في خانة المخرجات الرقم 0

٢- ثامن صف من المدخلات قيمة (1)  $A = 01$  مضروبة في قيمة (3)  $B = 11$  وكان الناتج عملية الضرب = 3

لذلك نكتب الرقم 3 في خانة المخرجات وطبعاً يكتب الرقم بالنظام الثنائي .

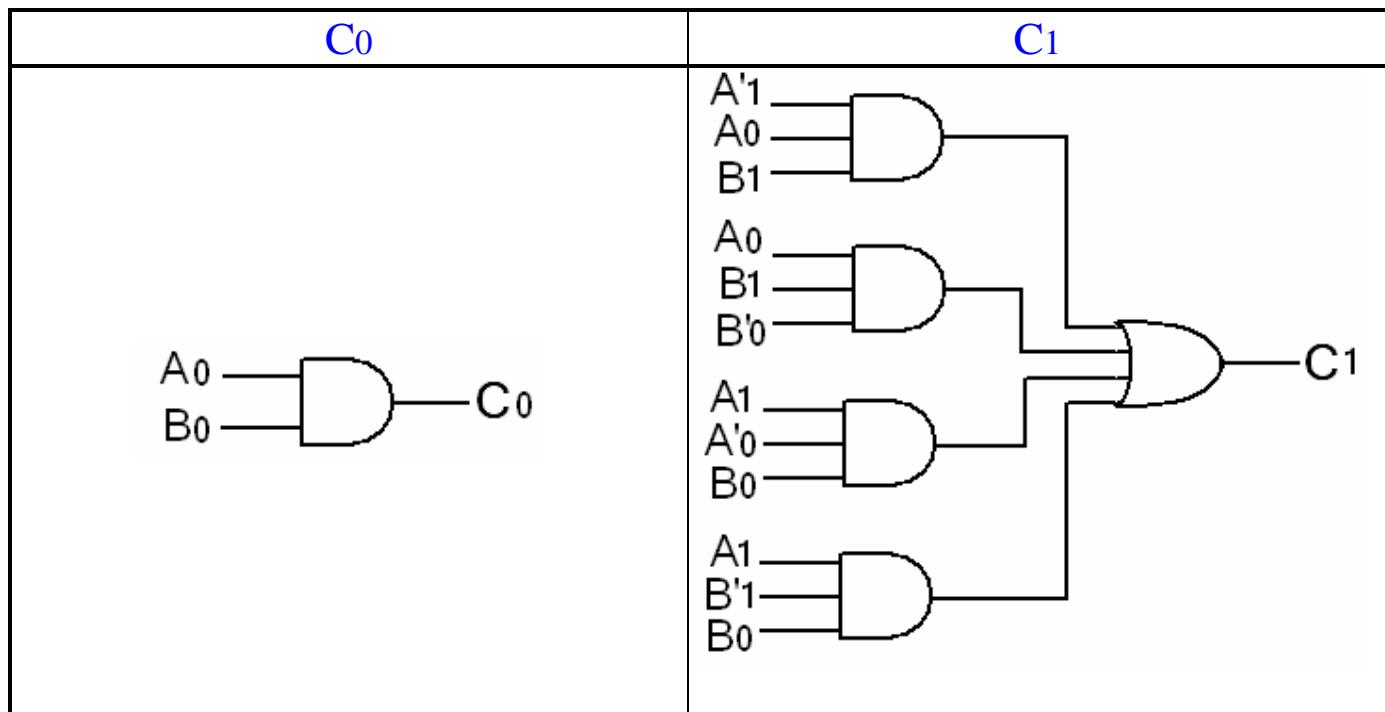
بعد ذلك نكمل الحل كما تعلمنا من تبسيط المخرجات والرسم .

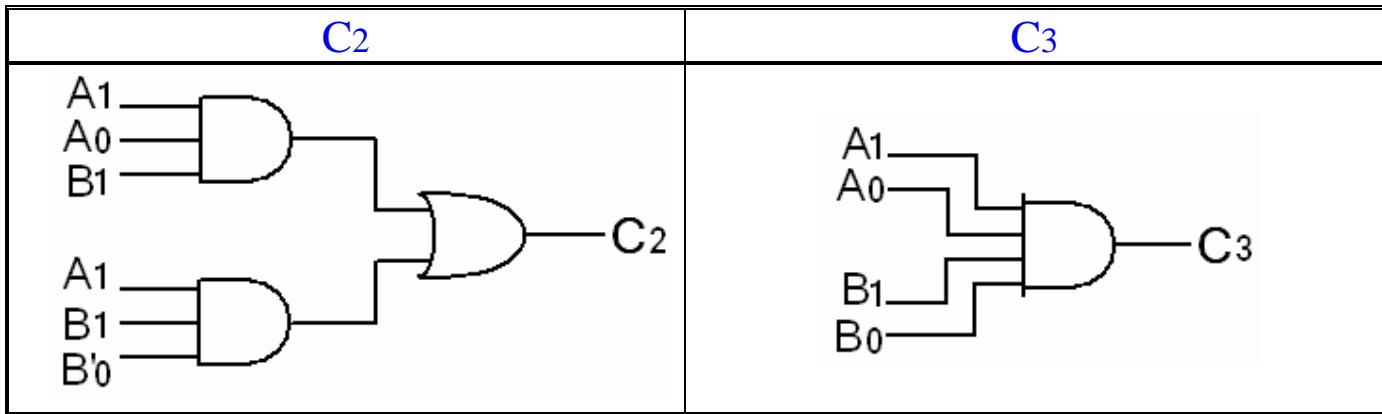
InPuts		OutPuts					
A1	A0	B1	B0	C3	C2	C1	C0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

		C0						C1			
		B1B0	00	01	11	10	A1A0	00	01	11	10
		A1A0	00				B1B0	00			
		00					00				
		01		1	1		01			1	1
		11		1	1		11		1		1
		10					10		1	1	

		C2						C3			
		B1B0	00	01	11	10	A1A0	00	01	11	10
		A1A0	00				B1B0	00			
		00					00				
		01					01				
		11				1	11			1	
		10			1	1	10				





: Example مثال

: Design a 2 - bit magnitude comparator

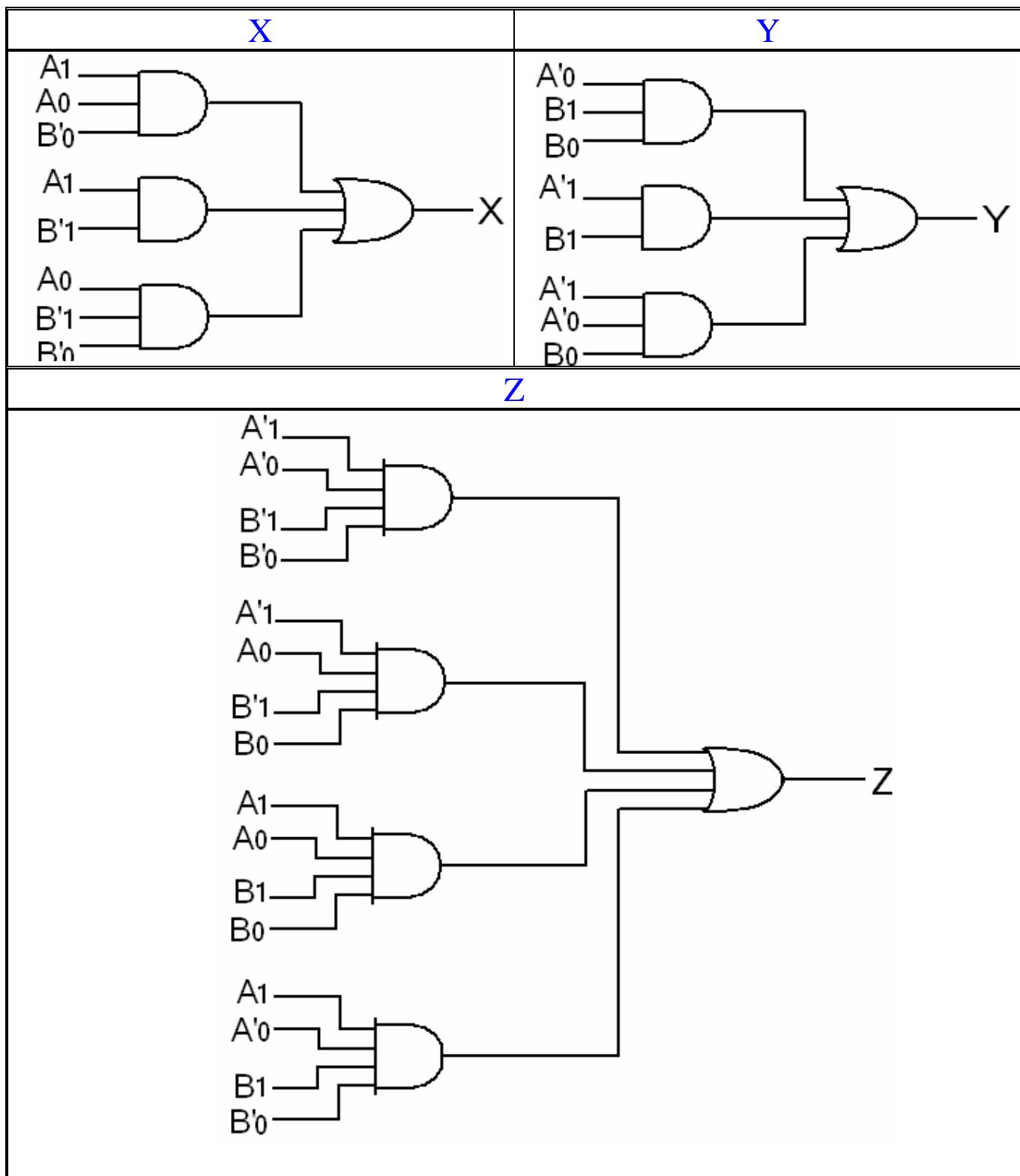
: Solution الحل

عملها تقوم بمقارنة بين قيم المدخلات.

المطلوب تصميم دالة (magnitude comparator)

InPuts		OutPuts				
A1	A0	B1	B0	X (A>B)	Y (A<B)	Z (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

		X						Y					
		B1B0	00	01	11	10	A1A0	B1B0	00	01	11	10	
A1A0		00					00		1	1		1	
	01		1							1		1	
	11		1	1									
	10		1	1							1		
$X = A_1B'_1 + A_1A_0B'_0 + A_0B'_1B_0$													
		Z											
		B1B0	00	01	11	10	A1A0	B1B0	00	01	11	10	
A1A0		00	1				00						
	01			1									
	11				1								
	10					1							
$Z = A'_1A'_0B'_1B'_0 + A'_1A_0B'_1B_0 + A_1A_0B_1B_0 + A_1A'_0B_1B'_0$													



: Decoder ٤-٦

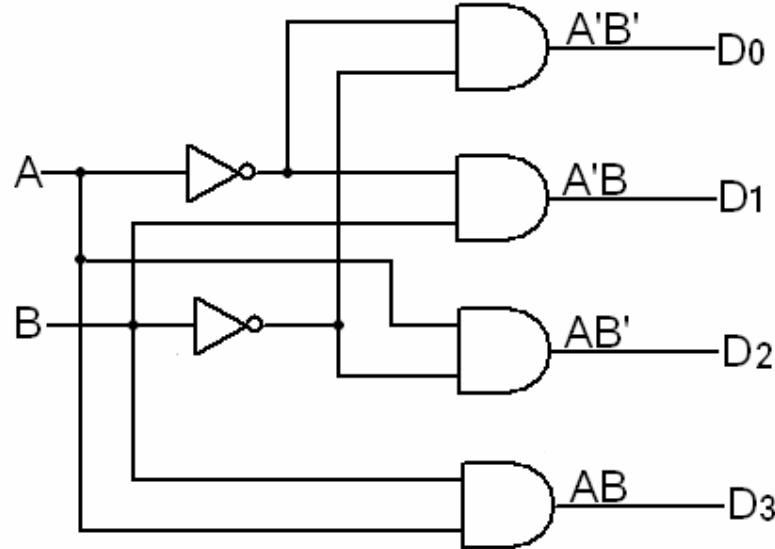
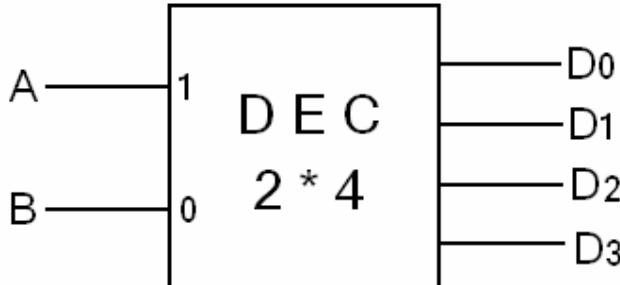
A decoder has  $n$  inputs and  $2^n$  outputs

هذه الدالة مدخلاتها =  $n$  و مخرجاتها =  $2^n$  أي  $2$  أ<sub>س(قوى)</sub> عدد المدخلات

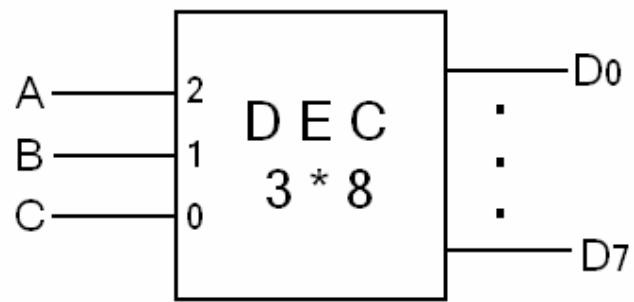
لو كان عدد المدخلات = 2 فإن عدد المخرجات =  $2^2 = 4$

لو كان عدد المدخلات = 3 فإن عدد المخرجات =  $8 = 2^3$   
وبالتالي فإن أنواع وأشكال (Decoder) يعتمد على عدد المدخلات.

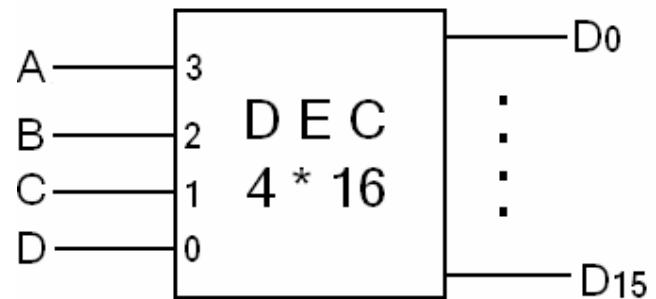
١ - ) : Decoder 2 \* 4 ( Has 2 inputs and 4 outputs  
هو أول أنواع وسوف نتعرف عليه أكثر من خلال الجدول التالي :

Truth Table	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th colspan="2">InPuts</th><th colspan="4">OutPuts</th></tr> <tr> <th>A</th><th>B</th><th>D<sub>3</sub></th><th>D<sub>2</sub></th><th>D<sub>1</sub></th><th>D<sub>0</sub></th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	InPuts		OutPuts				A	B	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>	0	0	0	0	0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	1	1	0	0	0
InPuts		OutPuts																																			
A	B	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>																																
0	0	0	0	0	1																																
0	1	0	0	1	0																																
1	0	0	1	0	0																																
1	1	1	0	0	0																																
كل صف من المدخلات وأقصد قيمة A,B معاً تشكل رقم وإذا كان الرقم الناتج = 0 نضع في الخانة D <sub>0</sub> الرقم 1 وبباقي الخانات نضع فيها الرقم 0 ولو كان الرقم الناتج = 1 نضع في الخانة D <sub>1</sub> الرقم 1 وبباقي الخانات نضع فيها الرقم 0 وهكذا .																																					
Graphic Symbol																																					
																																					
هذا الشكل الذي نحتاجه لحل المسائل بواسطة (Decoder)																																					

: Decoder 3 \* 8 -٢



: Decoder 4 \* 16 -٣



: Decoder With Enabel ٤-٧

في هذه الجزيئية سوف نتعرف على شيء جديد وهو (Enabel) و عمله بإختصار كالتالي :  
إذا كانت قيمة (Enabel) = 0 فلن (Decoder) ي العمل .  
أما إذا كانت قيمة (Enabel) = 1 فلن (Decoder) يعمل .  
إذن (Enabel) ليس إلا مفتاح للدالة .

InPuts			OutPuts			
E	A	B	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

الجدول السابق يوضح عمل (Enabel) وكما تلاحظون عندما كانت قيمة (Enabel) = 0 في الصفوف الأربع الأولى كان (Decoder) لا يعمل وبالتالي فإن قيم المخرجات = 0 وعندما كانت قيمة 1 = (Enabel) في الصفوف الأربع الأخيرة كان (Decoder) يعمل وظهرت نتيجة عملة على المخرجات .

InPuts			OutPuts			
E	A	B	D <sub>3</sub>	D <sub>2</sub>	D <sub>1</sub>	D <sub>0</sub>
0	X	X	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

الجدول السابق ليس إلا اختصار للجدول الذي قبله . وكما تلاحظون على الجدول الأول في الصفوف الأربع الأولى أن وجودها ليس له أي فائدة ووجودها زائد فقط لذلك قمنا بإختصارها في صف واحد وهو الصف الأول من الجدول الجديد ولم نلغى وجودها تماماً .

الصف الأول من الجدول الجديد تلاحظون فيه أن قيمة X = (B) و هذه (X) ليست (Don't care) وكما تعلمون أن (Don't care) توجد فقط في المخرجات ولا يمكن أن توجد في المدخلات إذن هي ليست (Don't care) وإنما هي اختصار ينوب عن كتابة قيمة (B) و (A) في المدخلات

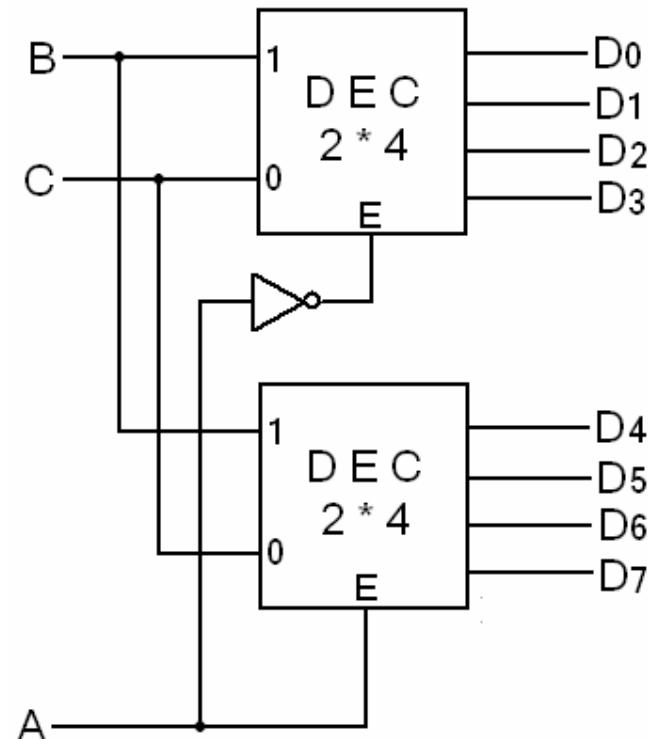
أي أن (X) مرّة = 0 و مرّة أخرى = 1 و كتبناها بهذه الطريقة للاختصار فقط و يمكنك تجاهلها و كتابة قيمة (B) و (A) كاملة إذا أردت .

: Example مثال

: Desing a Decoder 3\*8 using a Decoder 2\*4 with Enable and additional gate

: Solution الحل

المطلوب تصميم (Decoder 3\*8) بإستخدام (Decoder 2\*4) مع (Enable)



: explain الشرح

نلاحظ في الحل أن (A) يمثل قيمة (Enabel)

الجدول التالي ليس من الحل وإنما يصف فقط الذي يحدث في الشكل السابق .

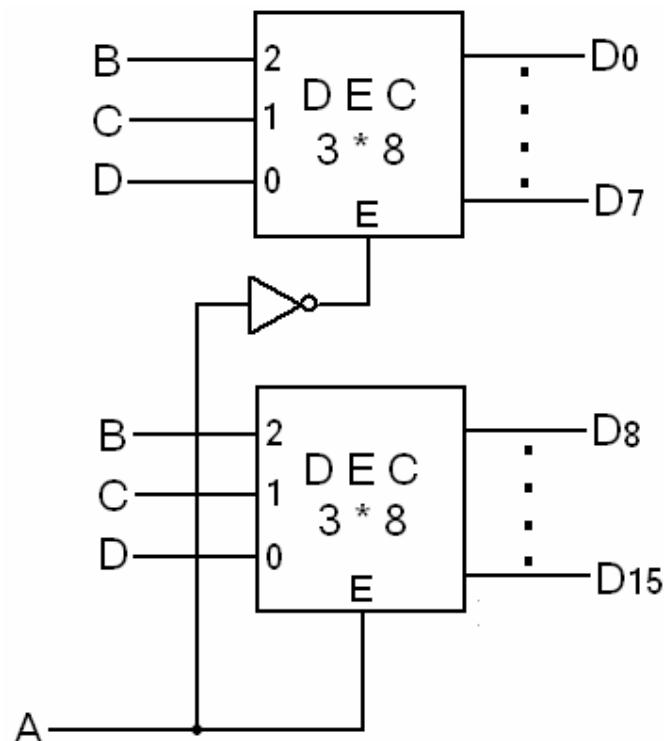
InPuts			OutPuts								
A	B	C	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	0	0	1	
0	0	1	0	0	0	0	0	0	1	0	
0	1	0	0	0	0	0	0	1	0	0	
0	1	1	0	0	0	0	1	0	0	0	
1	0	0	0	0	0	1	0	0	0	0	
1	0	1	0	0	1	0	0	0	0	0	
1	1	0	0	1	0	0	0	0	0	0	
1	1	1	1	0	0	0	0	0	0	0	

: Example مثال

: Desing a Decoder 4\*16 using a Decoder 3\*8 with Enable and additional gate

: Selution الحل

المطلوب تصميم (Decoder 4\*16) بـاستخدام (Decoder 3\*8) مع (Enable) وطريقة الحل هي نفس طريقة حل المثال الذي قبله تماماً.



: Multiplexer ٤-٨

A Multiplexer has  $2^n$  inputs , 1outputs and  $n$  selections

هذه الدالة مدخلاتها =  $2^n$  ومخرجاتها = 1 وعدد الإختيارات =  $n$  (Selections)

لو كان عدد المدخلات =  $2^2 = 4$  فإن عدد المخرجات = 1 وعدد الإختيارات = 2

لو كان عدد المدخلات =  $2^3 = 8$  فإن عدد المخرجات = 1 وعدد الإختيارات = 3

: Multiplexer 2\*1 - ١

A Multiplexer has 2 inputs , 1 outputs and 1 selections

هو أول أنواع وسوف نتعرف عليه أكثر من خلال الجدول التالي :

Truth Table

Selection	InPuts	OutPuts	
S	X	Y	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

قيمة المخرجات تكون بالشكل التالي :

إذا كانت  $S = 0$  فإن قيمة المخرجات هي قيمة  $Y$  ,  $Y$  هي قيمة

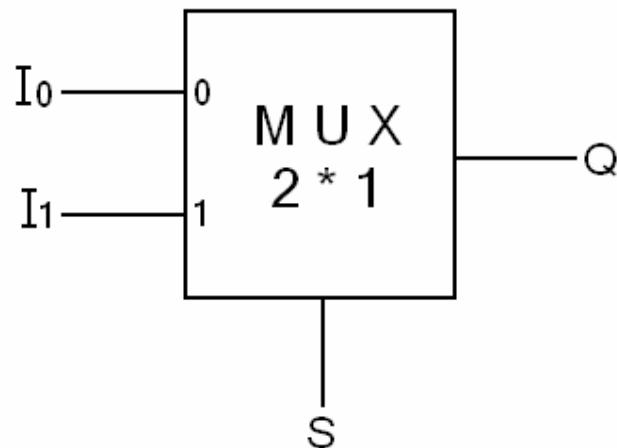
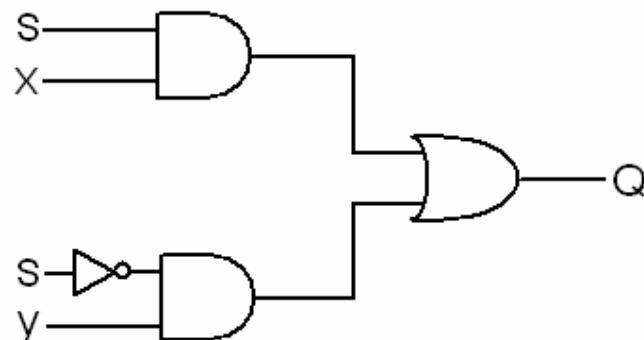
وإذا كانت  $S = 1$  فإن قيمة المخرجات هي قيمة  $X$  ,  $X$  هي قيمة

Algebraic Function

S	xy	00	01	11	10
0			1	1	
1				1	1

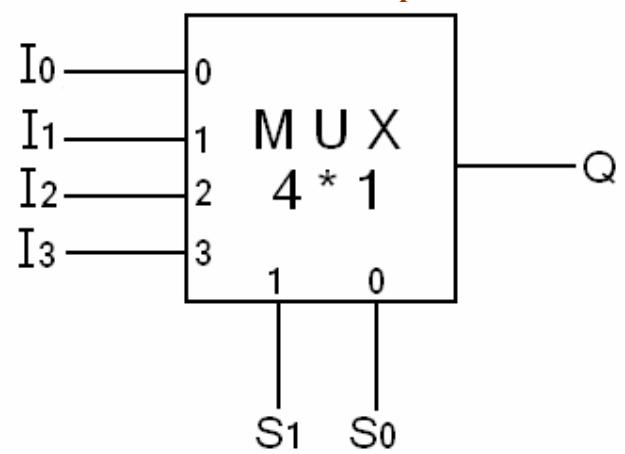
$$Q = sy + sx$$

Graphic Symbol

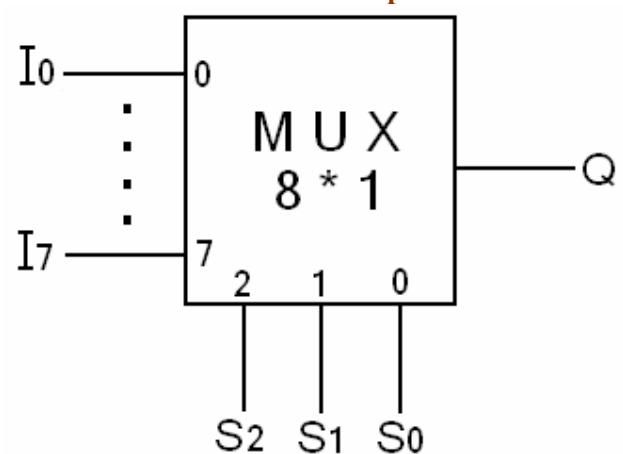


هذا الشكل الذي نحتاجه لحل المسائل بواسطة (Multiplexer)

: Multiplexer 4\*1 - ४



: Multiplexer 8\*1 - ८

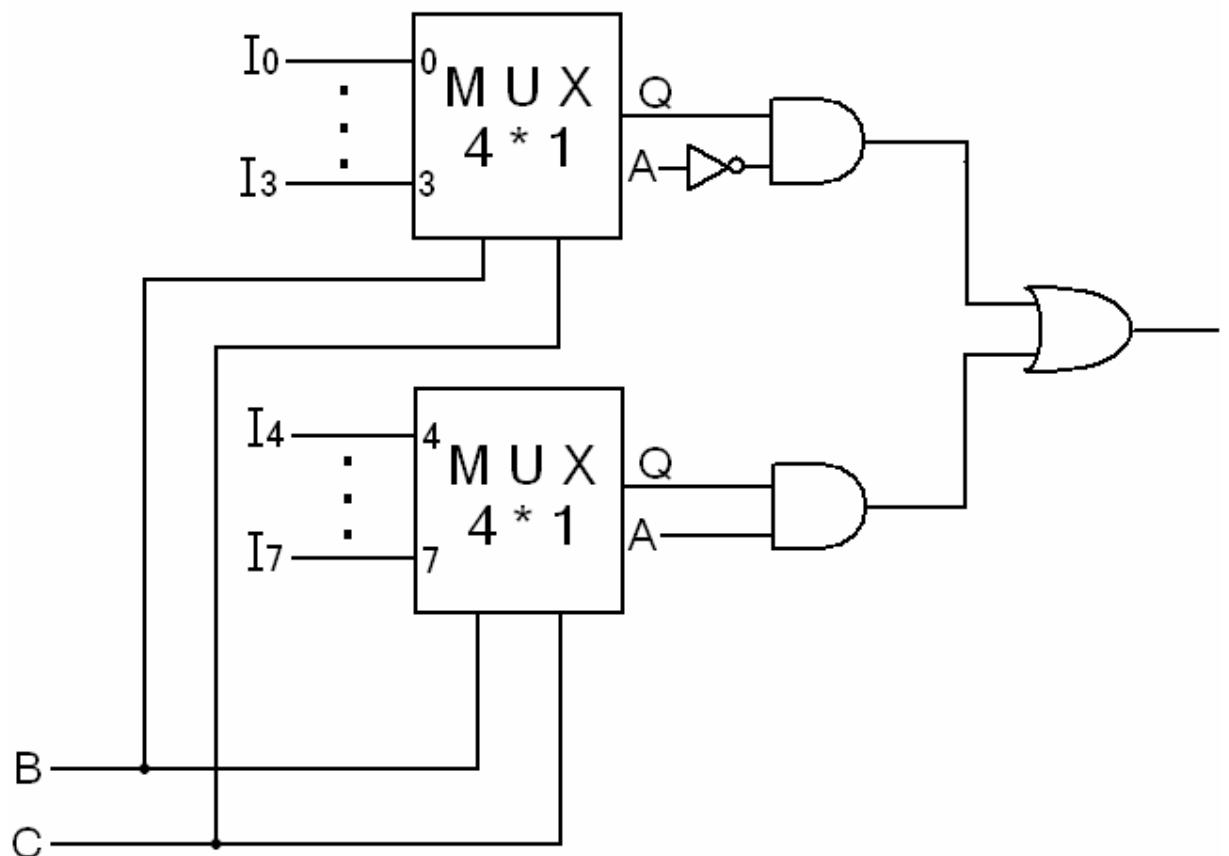


: Example مثال

: Construct an Multiplexer 8\*1 with two Multiplexer 4\*1 and additional gate

: Selution الحل

المطلوب تركيب(بناء) (Multiplexer 4\*1) 2 بـ (Multiplexer 8\*1) و دوال إضافية .

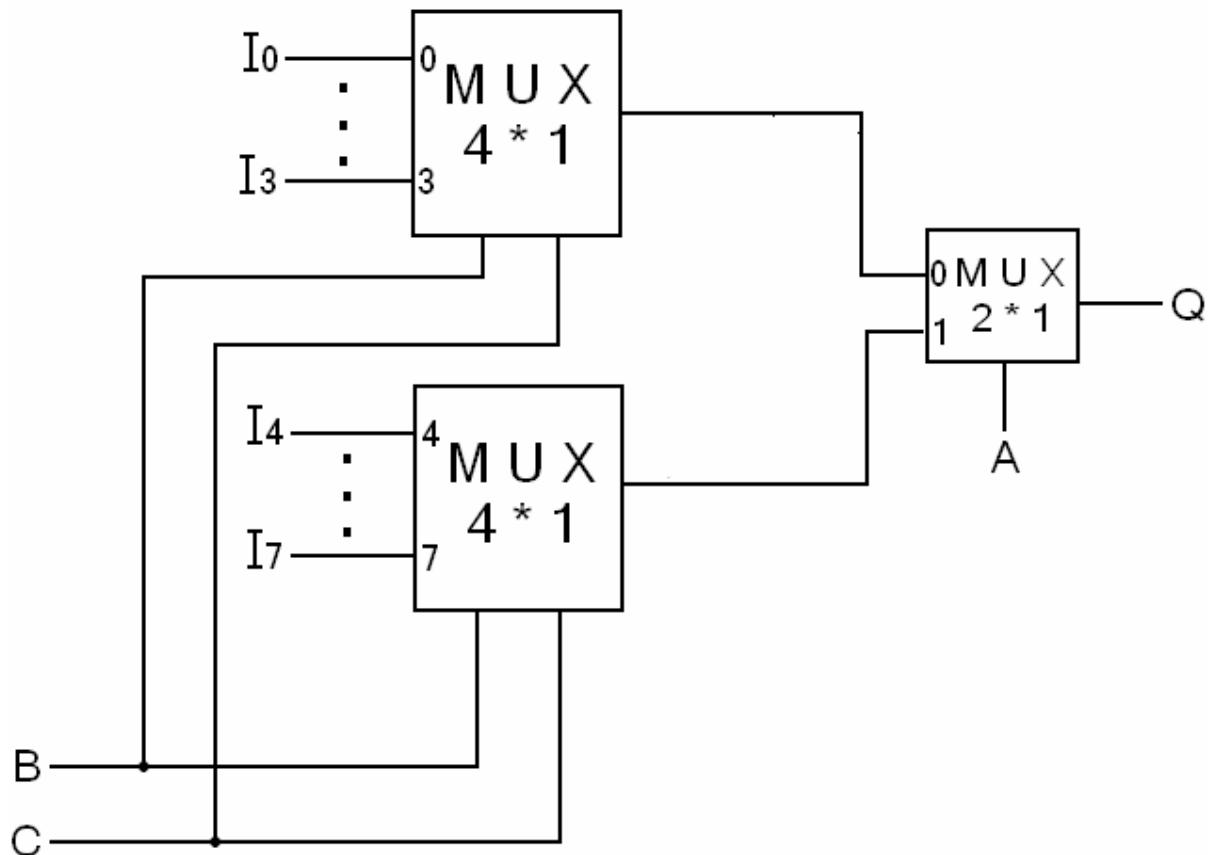


: Example مثال

: Construt an Multiplexer 8\*1 with two Multiplexer 4\*1 and one Multiplexer 2\*1

: Selution الحل

المطلوب تركيب(بناء)(Multiplexer 8\*1)  $\rightarrow$  2 (Multiplexer 4\*1) و 1 (Multiplexer 4\*1)

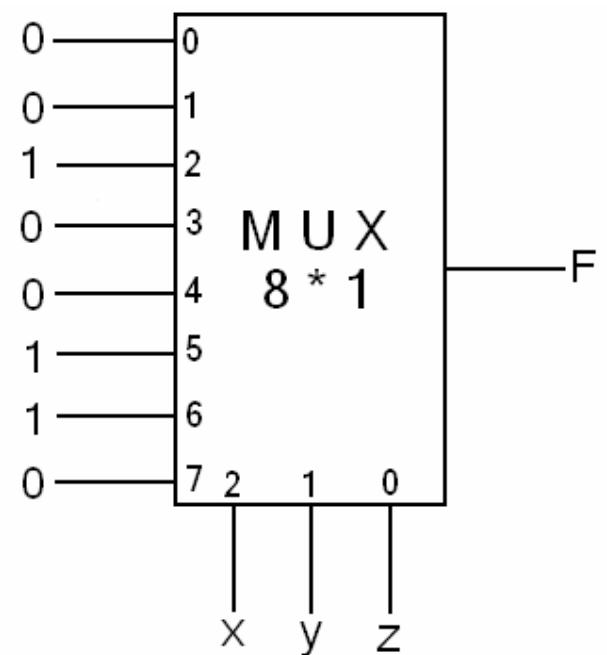


: Example مثال

: Implement the following Boolean function  $F(x,y,z) = \Sigma(2,5,6)$  using an 8\*1 Multiplexer

: Selution الحل

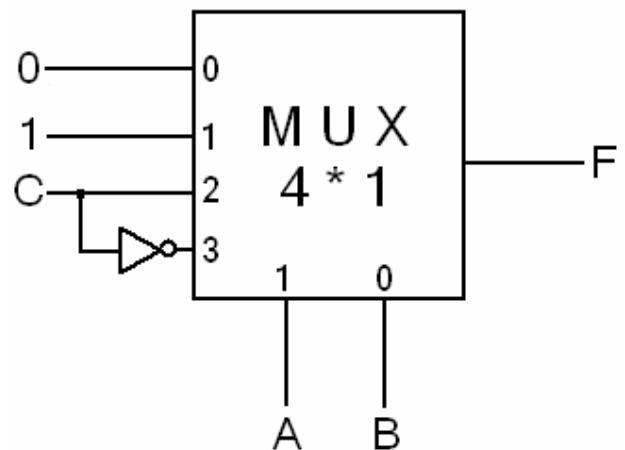
المطلوب تفیذ الدالة المنطقیة التالیة بایستخدام (8\*1 Multiplexer)



: Example مثال

: Implement the following Boolean function  $F(A,B,C) = \Sigma(2,3,5,6)$  using an 4\*1 Multiplexer  
الحل : Selution

المطلوب تفويذ الدالة المنطقية التالية بإستخدام (4\*1 Multiplexer)



: explain الشرح

يفترض أن نحل هذا المثال بإستخدام (8\*1 Multiplexer)  
ولأن كل قيمتين من المدخلات نستطيع أن نعرض عنها بقيمة واحدة نستطيع أن نحل المثال بـ (4\*1 Multiplexer)

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

الفصل السادس

Synchronous  
Sequential Logic

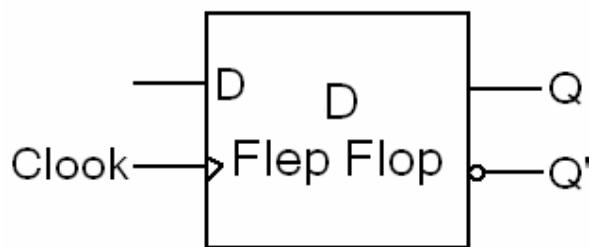
## ٥-١ المقدمة : Introduction

في هذا الفصل سوف نتعرف على دالة جديدة وهي (Flip Flop) .  
وسوف تحدث عن تعريفها وأشكالها وأنواعها أقسامها و عملها و حل المسائل بواسطتها والتحويل فيما بين أشكالها .  
كما سوف نتحدث التحليل المؤقت للسلسل الدائري (Analysis of clocked sequential circuits)  
. وإيجاد جدول (State diagram) ولدوائر (State table) .

## ٥-٢ أنواع الـ : Types of (Flip Flop)

### ١- D Flip Flop

هو أول أنواع (Flip Flop) وسوف نتعرف عليه أكثر من خلال الجدول التالي :



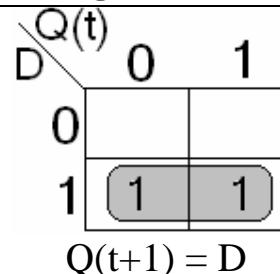
هذا الشكل الذي نحتاجة لحل المسائل بواسطة (D Flip flop).

D	$Q(t+1)$
0	0
1	1

الجدول السابق يوضح كيفية تكوين قيمة  $Q(t+1)$  وكما تلاحظون قيمة  $Q(t+1)$  ليست إلا نسخة من قيمة (D).

Present State		Next State
D	$Q(t)$	$Q(t+1)$
0	0	0
0	1	0
1	0	1
1	1	1

يعتمد تكوين هذا الجدول على الجدول الذي قبلة



$Q(t)$	$Q(t+1)$	D
0	0	0
0	1	0
1	0	1
1	1	1

الجدول يوضح كيفية تكوين قيمة (D) من خلال إعطائنا قيمة  $Q(t+1), Q(t)$  وهذا الجدول ليس إلا صورة عكسية لـ (Truth Table) الدالة.

## : J K Flip Flop - ٢

ثاني أنواع (Flip Flop) وسوف نتعرف عليه أكثر من خلال الجدول التالي :

Characteristic Table	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>J</th><th>K</th><th>Q(t+1)</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>Q(t)</td></tr> <tr> <td>0</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>Q'(t)</td></tr> </tbody> </table>	J	K	Q(t+1)	0	0	Q(t)	0	1	0	1	0	1	1	1	Q'(t)																									
J	K	Q(t+1)																																							
0	0	Q(t)																																							
0	1	0																																							
1	0	1																																							
1	1	Q'(t)																																							
Truth Table	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="3">Present State</th><th>Next State</th></tr> <tr> <th>J</th><th>K</th><th>Q(t)</th><th>Q(t+1)</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td></tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>0</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>0</td></tr> </tbody> </table>	Present State			Next State	J	K	Q(t)	Q(t+1)	0	0	0	0	0	0	1	1	0	1	0	0	0	1	1	0	1	0	0	1	1	0	1	1	1	1	0	1	1	1	1	0
Present State			Next State																																						
J	K	Q(t)	Q(t+1)																																						
0	0	0	0																																						
0	0	1	1																																						
0	1	0	0																																						
0	1	1	0																																						
1	0	0	1																																						
1	0	1	1																																						
1	1	0	1																																						
1	1	1	0																																						
Characteristic Equation	$Q(t+1) = JQ'(t) + K'Q(t)$																																								
Excitation Table	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Q(t)</th><th>Q(t+1)</th><th>J</th><th>K</th></tr> </thead> <tbody> <tr> <td>0</td><td>0</td><td>0</td><td>X</td></tr> <tr> <td>0</td><td>1</td><td>1</td><td>X</td></tr> <tr> <td>1</td><td>0</td><td>X</td><td>1</td></tr> <tr> <td>1</td><td>1</td><td>X</td><td>1</td></tr> </tbody> </table>	Q(t)	Q(t+1)	J	K	0	0	0	X	0	1	1	X	1	0	X	1	1	1	X	1																				
Q(t)	Q(t+1)	J	K																																						
0	0	0	X																																						
0	1	1	X																																						
1	0	X	1																																						
1	1	X	1																																						

### : T Flip Flop -٣

ثالث أنواع (Flip Flop) وسوف نتعرّف عليه أكثر من خلال الجدول التالي :

Characteristic Table	<table border="1"> <thead> <tr> <th>T</th> <th>Q(t+1)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Q(t)</td> </tr> <tr> <td>1</td> <td>Q'(t)</td> </tr> </tbody> </table>	T	Q(t+1)	0	Q(t)	1	Q'(t)												
T	Q(t+1)																		
0	Q(t)																		
1	Q'(t)																		
Truth Table	<table border="1"> <thead> <tr> <th colspan="2">Presnt State</th> <th>Next State</th> </tr> <tr> <th>T</th> <th>Q(t)</th> <th>Q(t+1)</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Presnt State		Next State	T	Q(t)	Q(t+1)	0	0	0	0	1	1	1	0	1	1	1	0
Presnt State		Next State																	
T	Q(t)	Q(t+1)																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	
Characteristic Equation	$Q(t+1) = T'Q(t) + tq'(t) \Rightarrow T \oplus Q(t)$																		
Excitation Table	<table border="1"> <thead> <tr> <th>Q(t)</th> <th>Q(t+1)</th> <th>T</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> </tr> </tbody> </table>	Q(t)	Q(t+1)	T	0	0	0	0	1	1	1	0	1	1	1	0			
Q(t)	Q(t+1)	T																	
0	0	0																	
0	1	1																	
1	0	1																	
1	1	0																	

: Example مثال

: Desing a JK Flip Flop using D Flip Flop

: Selution الحل

المطلوب تصميم (D Flip Flop) JK Flip Flop (JK) باستخدام (Truth table) ولكي تقوم بعملية التصميم يتوجب عليك أولاً عمل خلايا (Truth table) تقسم بالشكل التالي :

- ١ - (Present state) ويحتوي على قيمة (Flip Flop) الذي نريد تصميمه وفي هذا المثال (Flip Flop) الذي نريد تصميمه (JK)، ويحتوي أيضاً على قيمة  $Q(t)$
- ٢ - (Next state) ويحتوي على قيمة (Flip Flop) الذي نريد استخدامه في عملية التصميم وفي هذا المثال هو (D) ويحتوي أيضاً على قيمة  $Q(t+1)$

لإيجاد قيمة  $Q(t+1)$  نرجع لـ (Characteristic table) الخاص بجدول (D Flip Flop) الخاص بجدول (Excitation table) أما إيجاد قيمة (D) نرجع لـ

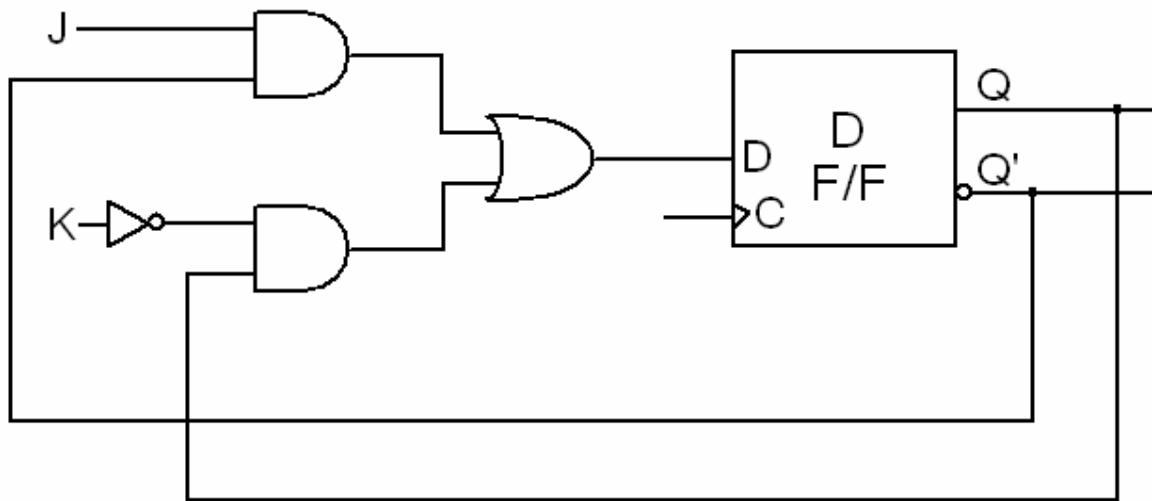
J	K	$Q(t)$	$Q(t+1)$	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

الخطوة الثانية نقوم بعملية التبسيط للمخرجات والذي يهمنا تبسيطه هو عمود (Flip Flop) المستخدم في عملية التصميم فقط وهو (D) ولسنا بحاجة لتبسيط  $Q(t+1)$

		00	01	11	10
		0	1		
J	K	00			
		0	1		
J	K	01	1		
		1	1		
J	K	11			
		1			
J	K	10			
		1			

$$D = JQ'(t) + K'Q(t)$$

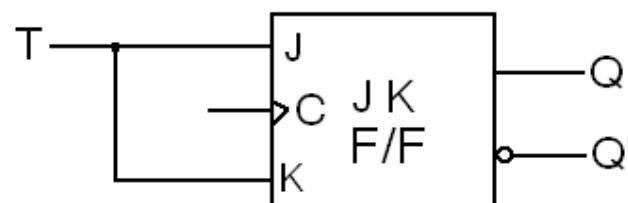
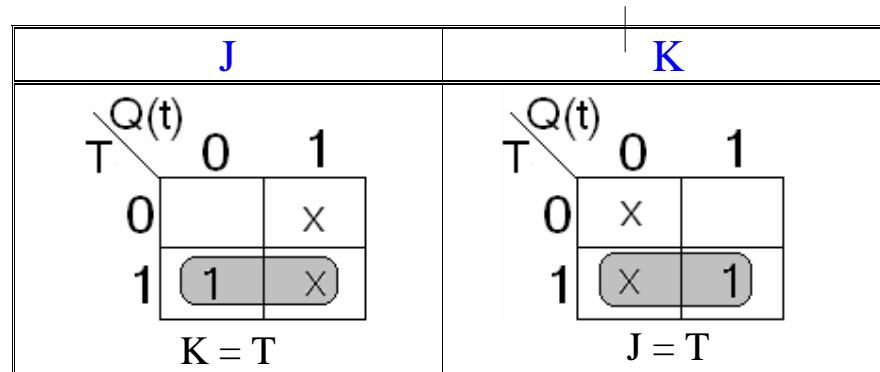
الخطوة الثالثة والأخيرة رسم الدالة الناتجة من عملية التبسيط .  
وبالتالي نكون قد إنتهينا من عملية التصميم .



**مثال** : Example  
**: Desing a T Flip flop using J K Flip flop**  
**الحل** : Selution

المطلوب تصميم (T Flip Flop) بإستخدام (J K Flip Flop) وطريقة الحل هي نفس الطريقة المتبعة في المثال الذي قبله .

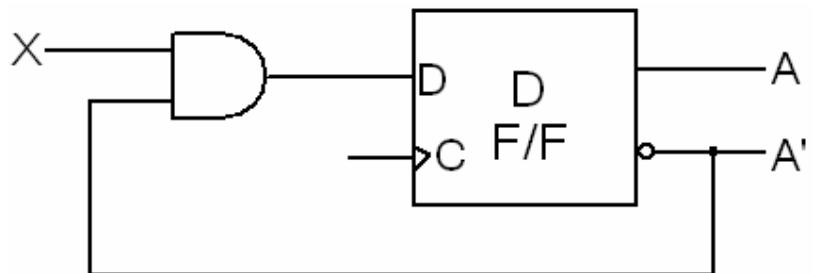
T	Q(t)	Q(t+1)	J	K
0	0	0	0	X
0	1	1	X	0
1	0	1	1	X
1	1	0	X	1



## ٣-٥ التحليل المؤقت للتسلسل الدائري : Analysis of clocked sequential circuits

مثال : Example

: Analysis of clocked sequential circuits



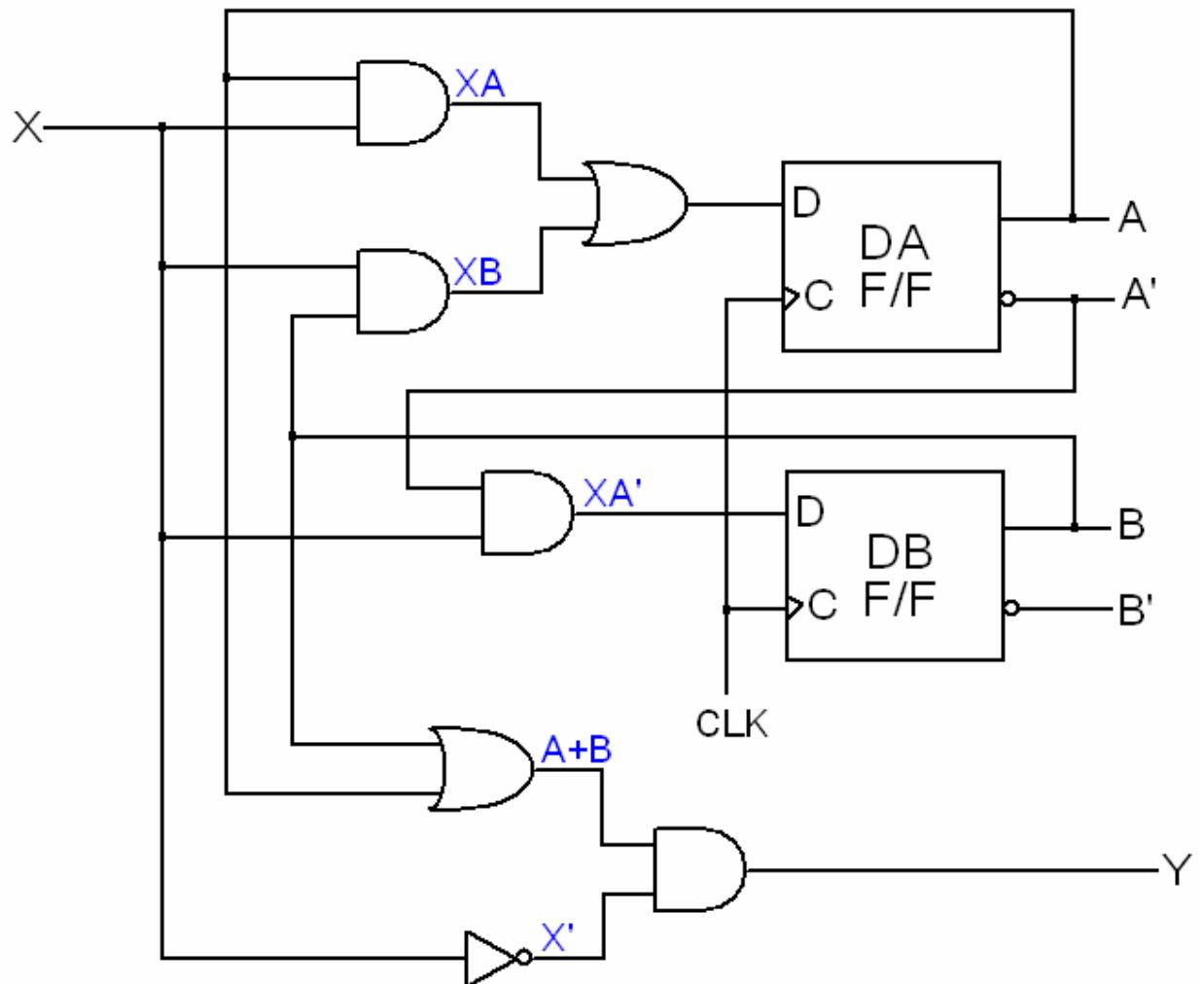
الحل : Selution

المطلوب تحليل الشكل السابق للوصول لمخرجات الدالة .

نرجع للشكل ونلاحظ أنه يحتوي على 1 (Flip Flop) وهو (A) وكم نعلم أنه ينتج من كل قيمه واحدة لـ (N.S) أي القيمة (Flip Flop)

$$A(t+1) = D$$

$$A(t+1) = XA'(t)$$



### الحل : Solution

المطلوب تحليل الشكل السابق للوصول مخرجات الدالة أي الوصول لقيمة (O/P) و (N.S)

نرجع للرسم ونلاحظ أنه يحتوي 2 (Flip Flop)

الأول (A Flip Flop) وسوف تنتج منه القيمة الأولى لـ (N.S) وهي (A(t+1))

الثاني (A Flip Flop) وسوف تنتج منه القيمة الثانية لـ (N.S) وهي (B(t+1))

أما (Y) فتمثل قيمة (O/P)

نرجع للشكل ونقوم بتحليله وتتبع مدخلاته ومن ثم نصل للقيم المطلوب إيجادها وهي كالتالي :

$$A(t+1) = DA \rightarrow XA + XB$$

$$B(t+1) = DB \rightarrow XA'$$

$$Y = (A+B)X' \rightarrow X'A + X'B$$

## ٥-٤ : State Table

سوف نقوم بإنشاء (State table) للمثال السابق .

P.S		I/P	N.S		O/P
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	0	1
0	1	1	1	1	0
1	0	0	0	0	1
1	0	1	1	0	0
1	1	0	0	0	1
1	1	1	1	0	0

### الشرح : explain

لإجاد جدول (State table) للمثال السابق يتوجب علينا أولاً تحليل الشكل السابق وإجاد قيم (O/P), (N.S) ، (A) و (B) ومن خلال المعادلات الثلاث التي أوجدناها وأقصد قيم  $(A(t+1), B(t+1), Y)$  نستطيع إجاد (State table) للمثال السابق .

وإليكم الآن شرح يصف لنا كيف تمكنا من تعبئة الجدول السابق :

بالنسبة لـ (P.S) و (I/P) نقوم بتباعية حقولها كما تعلمنا سابقاً .

أما بالنسبة لـ (N.S) و (O/P) سوف تعتمد على المعادلات السابقة لإجاد قيمها

من معادلة  $A(t+1) = XA + XB$  نوجد قيمة العمود (A)

ومن معادلة  $B(t+1) = XA'$  نوجد قيمة العمود (B)

أما من معادلة  $Y = XB$  نوجد قيمة العمود (Y)

على سبيل المثال لو أخذنا الصيغ الرابع من الجدول وأردنا أن نبين كيف قمنا بتباعية حقوله

قيم (N.S) و (O/P) في هذا الصيغ كالتالي :  $(A = 0)$  و  $(B = 1)$  و  $(X = 1)$

$$\begin{aligned} \text{نعرض في المعادلة } A(t+1) &= XA + XB \text{ لكي نوجد قيمة (A)} \\ A &= XA + XB \\ &= 1*0 + 1*1 \\ &= 0 + 1 \\ &= 1 \end{aligned}$$

$$\begin{aligned} \text{ونعرض في المعادلة } B(t+1) &= XA' \text{ لكي نوجد قيمة (B)} \\ B &= XA' \\ &= 1*1 \\ &= 1 \end{aligned}$$

ونعرض في المعادلة  $Y = X'A + X'B$  لكي يوجد قيمة (Y)

$$Y = X'A + X'B$$

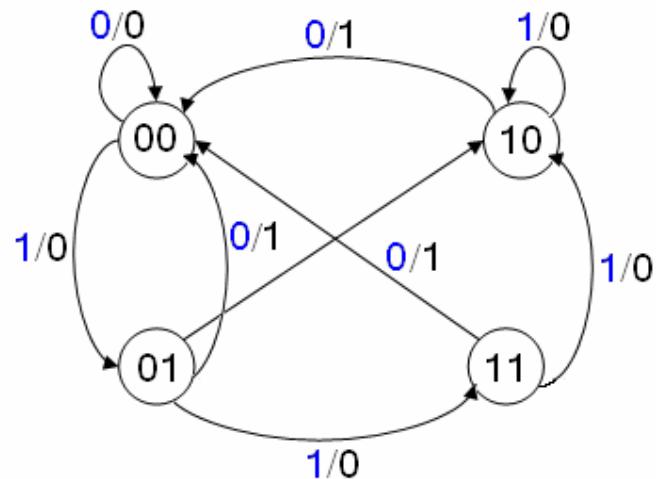
$$= 0*1 + 0*1$$

$$= 0 + 0$$

وبالتالي تكون قد إنتهينا من تعبئة حقول الصف الرابع ونعمل هذه الطريقة مع باقي الصفوف لتعبئة الجدول كاملاً.

## ٥-٥ : State Diagram

سوف نقوم بإنشاء (State diagram) للمثال السابق



### الشرح : explain

(State diagram) ليس إلا وصف لجدول (State table) بالرسم .  
وعند وجود أحدهما نستطيع إجاد الآخر منه .

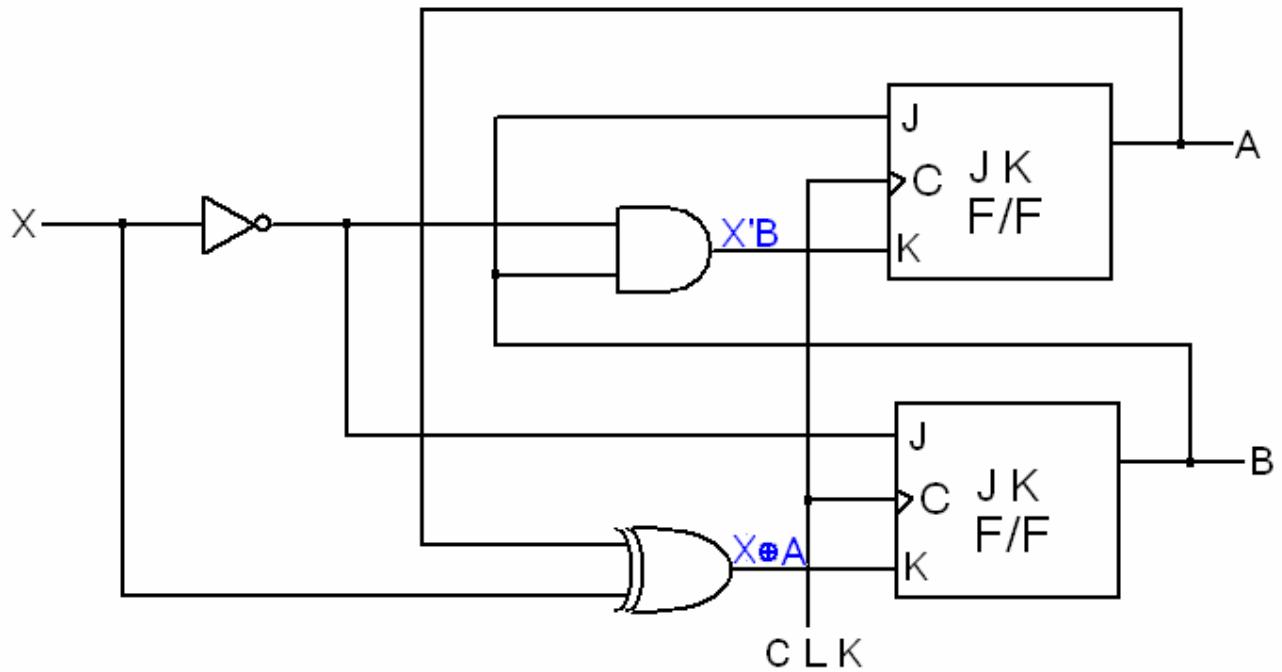
وإليكم الآن شرح يصف لنا الرسمة السابقة :

القيم التي داخل الدوائر تمثل قيم (A و B) التي تمثل قيم [(P.S),(N.S)]  
أما القيم التي على الأسهم تمثل قيم (Y و X) التي تمثل قيم [(I/P),(O/P)]  
X تمثل قيمة (I/P) أما Y فتمثل قيمة (O/P)

نرجع لجدول (State table) ونريد الآن أن نوضح كيفية تمثيله بـ (State diagram)  
قيم الصف الأول كالتالي : (P.S) = 00 ، (N.S) = 00 ، (I/P) = 0 ، (O/P) = 0  
وعند تمثيل الصف الأول من جدول (State table) تقوم وبالتالي :  
نتجه للدائرة التي تحتوي على قيمة (P.S) وهي مثانا = 00 ، ثم ننظر في قيمة (N.S) وهي مثانا = 00 ، ثم نرسم سهم من قيمة (P.S) يتوجه إلى قيمة (N.S) أي من القيمة (00) إلى القيمة (00) ونضع على السهم قيمة (I/P)  
(O/P) حيث أن القيمة التي باللون الأزرق تمثل 0 (I/P) والقيمة التي باللون الأسود تمثل 0 (O/P)

الصف الرابع (P.S) = 10 ، (N.S) = 11 ، (I/P) = 1 ، (O/P) = 0  
نرسم سهم من القيمة (10) يتوجه إلى القيمة (11) ونضع على السهم القيمة (1/0) وهكذا نعمل مع باقي القيم .

: Example مثال  
: Analysis with J K Flip flop



**الحل : Selution**

طريقة حل هذا المثال هي نفس الطريقة المتتبعة مع المثال الذي قبله .

$$JA = B$$

$$KA = X'B$$

$$JB = X'$$

$$KB = A'X + X'A \Rightarrow X \oplus A$$

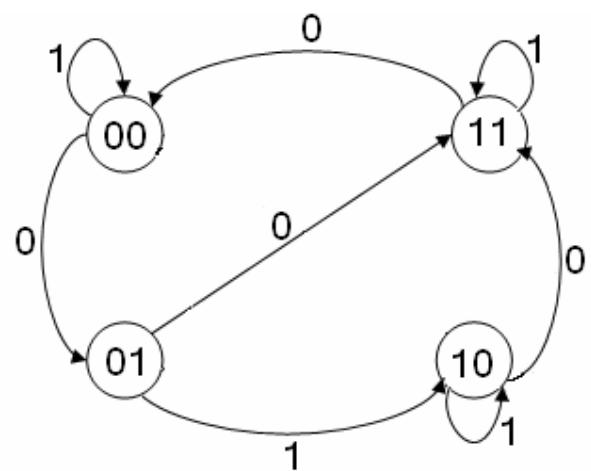
$$\begin{aligned} A(t+1) &= JA \cdot A' + KA \cdot A \\ &= BA' + (X'B)A \\ &= A'B + (X+B')A \\ &= A'B + XA + AB' \end{aligned}$$

$$\begin{aligned} B(t+1) &= JB \cdot B' + KB \cdot B \\ &= X'B' + (XA' + X'A)B \\ &= X'B' + X'A'B + XAB \end{aligned}$$

## : State Table

P.S		I/P	N.S	
A	B	X	A	B
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

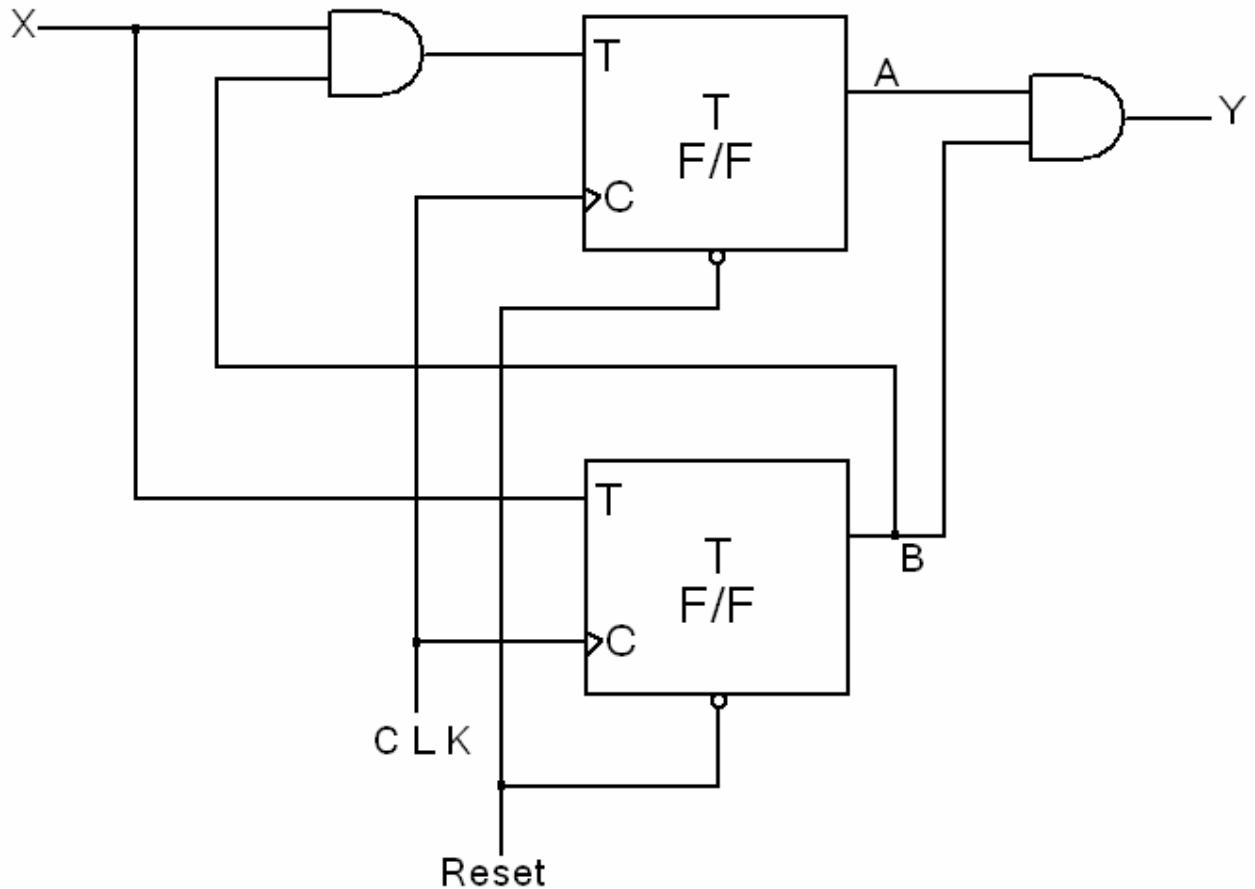
## : State Diagram



الشرح : explain

لاحظوا أنه يوجد قيمة واحدة على الأسماء وهي قيمة (X) وذلك لأنه لا يوجد قيمة لـ (Y) في هذا المثال .

: Example  
: Analysis with T Flip flop



**الحل : Solution**

طريقة حل هذا المثال هي نفس الطريقة المتبعة مع المثال السابق .

$$TA = XB$$

$$TB = X$$

$$\begin{aligned} A(t+1) &= TA \oplus A' \\ &= XB \oplus A \end{aligned}$$

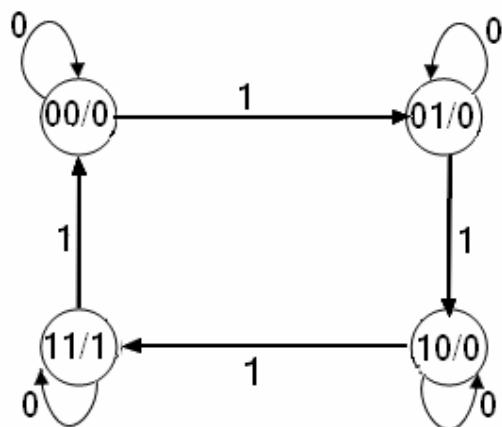
$$\begin{aligned} B(t+1) &= TB \oplus B' \\ &= X \oplus B \end{aligned}$$

$$Y = AB$$

## : State Table

P.S		I/P	N.S		O/P
A	B	X	A	B	Y
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1

## : State Diagram



## الشرح : explain

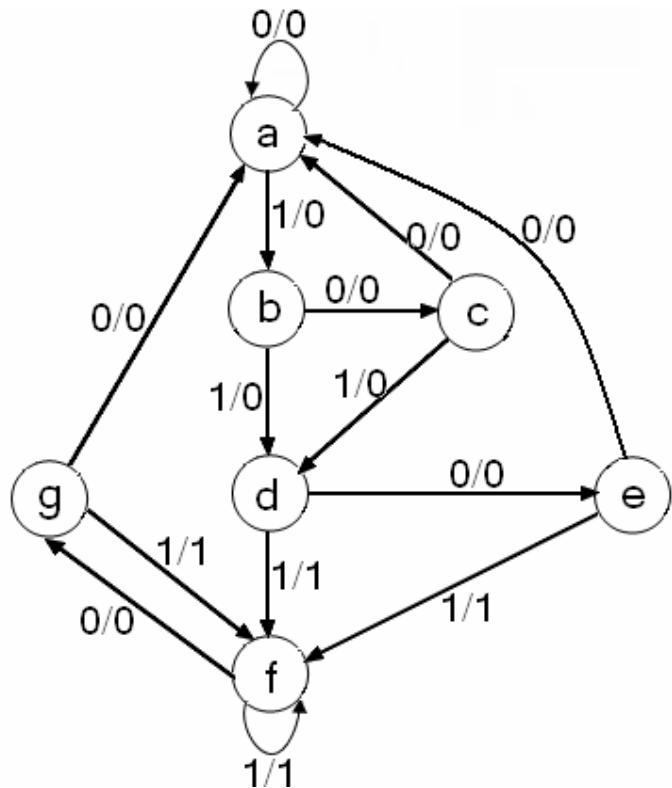
لاحظوا على رسمة (State diagram) أن قيمة (Y) لم تكتب على السهم كما تعلمنا سابقاً وإنما كتبت داخل الدوائر مع قيم (N.S) أي مع قيم (A,B) والسبب أن قيمة (Y) تعتمد في تكوينها على قيمة (N.S) دون قيمة (X) مباشرة أما في الأمثلة السابقة فإن (X) كان يدخل في تكوين قيمة (Y) لذلك كانت نضع قيمة (Y) على الأسهوم مع قيمة (X) أما في هذا المثال فإن قيمة (Y) لا تعتمد على قيمة (X) .

راجع رسومات جميع الأمثلة السابقة ولاحظ الفرق بينها وبين رسمة هذا المثال من حيث تمثيل الدالة (Y) بالرسم .

## ٥-٦ : State Reduction and Assignment

في هذه الجزئية سوف تختلف علينا صيغة السؤال عن الأمثلة السابقة وكذلك تكون جدول (State table) وكذلك اختلاف المطلوب إجاده ، حيث أنك سوف تقوم بإجاد جدول (State table) من رسمة (State diagram) بعد ذلك تقوم بإختصار لقيم المتشابهة في جدول (State table) إلى أن تصل إلى أبسط صورة ممكنة .

### مثال : Example



### الحل : Seloution

المطلوب إنشاء جدول (State table) من رسمة (State diagram) ومن ثم تبسيط جدول (State table) إلى أن نصل إلى أبسط صورة ممكنة .

وإليكم الآن رشح يصف لنا الرسمة السابقة :

القيم التي داخل الدوائر تمثل قيم (P.S) .

أما القيم التي على الأسهم تمثل قيم (X) .

حيث أن القيمة التي على يسار الخط تمثل قيمة (X) في حالة (N.S) .

أما القيمة التي على يمين الخط تمثل قيمة (X) في حالة (O/P) .

ولإنشاء جدول (State table) من رسمة (State diagram) السابقة لا عليك إلا فقط تتبع الأسهوم من أين إنطلقت وإلى أين اتجهت .

P.S	N.S		O/P	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	f	0	1
e	a	f	0	1
f	g	f	0	1
g	a	f	0	1

### الشرح : explain

في هذا المثال قمنا بإنشاء جدول (State diagram) لرسمة (State table) السابقة .

وإليكم الآن شرح يصف لنا تكوين الجدول السابق :

نضع في العمود (P.S) القيم التي سوف تنطلق منها السهم .

ونضع في العمودين (N.S) القيم التي سوف تصل إليها الأسهم .

حيث أنه إذا كانت قيمة (X) التي على يسار الخط = 0 نضع في العمود (X = 0) الذي يمثل الجزء الأول من عمود (N.S) القيمة التي وصل إليها السهم ، ونضع في العمود (X = 0) الذي يمثل الجزء الأول من عمود (O/P) القيمة التي على يمين الخط الفاصل بين قيمتي (X)

وإذا كانت قيمة (X) التي على يسار الخط = 1 نضع في العمود (X = 1) الذي يمثل الجزء الثاني من عمود (N.S) القيمة التي وصل إليها السهم ، ونضع في العمود (X = 1) الذي يمثل الجزء الثاني من عمود (O/P) القيمة التي على يمين الخط الفاصل بين قيمتي (X)

على سبيل المثال لوأخذنا الصف الأول من الجدول لنبين كيف قمنا بتبنته :

نلاحظ على الجدول أن قيمة (P.S = a) بعد ذلك نرجع لرسمة (State diagram) ونتجه للدائرة التي تحتوي

على القيمة (a) ون تتبع الأسهم التي إنطلقت من هذه القيمة ونجد أنه إنطلق من هذه القيمة سهمان

الأول عاد إلى نفس القيمة التي إنطلق منها وهي القيمة (a) وعلى هذا السهم توجد القيمة (0/0) لذلك نضع في العمود (X = 0) الذي يمثل الجزء الأول من عمود (N.S) القيمة التي وصل إليها السهم وهي القيمة (a)

ونضع في العمود (X = 0) الذي يمثل الجزء الأول من عمود (O/P) القيمة التي على يمين الخط الفاصل بين قيمتي وهي القيمة (X) وهي القيمة (0) وبالتالي تكون قد إنتهينا من السهم الأول .

ثم نأخذ السهم الثاني ونلاحظ أنه إتجه إلى الدائرة التي تحتوي على القيمة (b)

ونعمل معه مثل ما عملنا مع السهم الأول .

وبالتالي تكون قد إنتهينا من تبنة حقول الصف الأول من الجدول .

ونعمل مع باقي الصفوف مثل ما عملنا مع الصف الأول .

وبالتالي تكون قد إنتهينا من الخطوة الأولى وهي تبنة الجدول .

ثم ننتقل للخطوة الثانية وهي تبسيط الجدول

ننظر في قيم الجدول ونبحث إذا كان هنالك يوجد صفوف متشابهه لكي نختصرها معاً.

ونلاحظ أن الصفيين (e) و (g) متشابهان أي لهما نفس القيم ولذلك سوف نقوم بالغاء أحدهما ولتكن الصف (g) وبعد إلغائنا لهذا الصف نقوم بخطوة ثانية وهي إستبدال كل (g) في الجدول بالقيمة الجديدة لها وهي القيمة (e) وكما تلاحظون توجد القيمة (g) في الصف (f) قمنا بإستبدالها بالقيمة الجديدة وهي القيمة (e).

P.S	N.S		O/P	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
<b>d</b>	<b>e</b>	<b>f</b>	<b>0</b>	<b>1</b>
e	a	f	0	1
<b>f</b>	<b>e</b>	<b>f</b>	<b>0</b>	<b>1</b>

بعد إلغائنا للصف (g) وإستبدال كل (g) بالقيمة (e) أصبح لدينا صفين آخرين متشابهين وهما (d) و (f) تقوم بالغاء الصف (f) وإستبدال كل (f) بالقيمة (d).

P.S	N.S		O/P	
	X = 0	X = 1	X = 0	X = 1
a	a	b	0	0
b	c	d	0	0
c	a	d	0	0
d	e	<b>d</b>	0	1
e	a	<b>d</b>	0	1

بعد إلغائنا الصف (f) وإستبدال كل (f) بالقيمة (d) تكون قد إنتهينا من تبسيط جدول (State table) وبالتالي تكون توصلنا للحل النهائي للمثال.

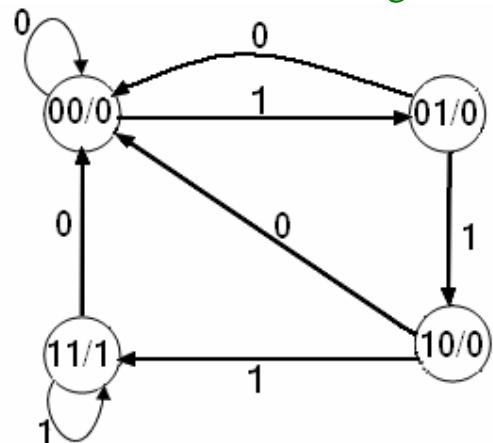
## ٥-٧ إجراء التصميم : Design Procedure

هذه الجزئية تمثل صورة عكسية لجزئية (Analysis of clocked sequential circuits) حيث أنه في جزئية (Analysis of clocked sequential circuits) كان يعطينا الرسمة ونقوم بتحليلها ونوجد قيم (N.S) و (O/P) ومن هذه القيم نوجد جدول (State table) ومن هذا الجدول نوجد رسمة (State diagram).

أما في جزئية (Design Procedure) فسوف يحدث العكس تماماً حيث أنه سوف يعطيك (N.S) ومنها سوف تقوم بإجاد جدول (State table) بعد ذلك سوف تقوم بتبسيط مخرجات الجدول وإجاد قيم (O/P) وأخيراً نقوم برسم الدوال الناتجة.

### مثال : Example

: Design circuit that detects three or more consecutive 1'S in a string of bits

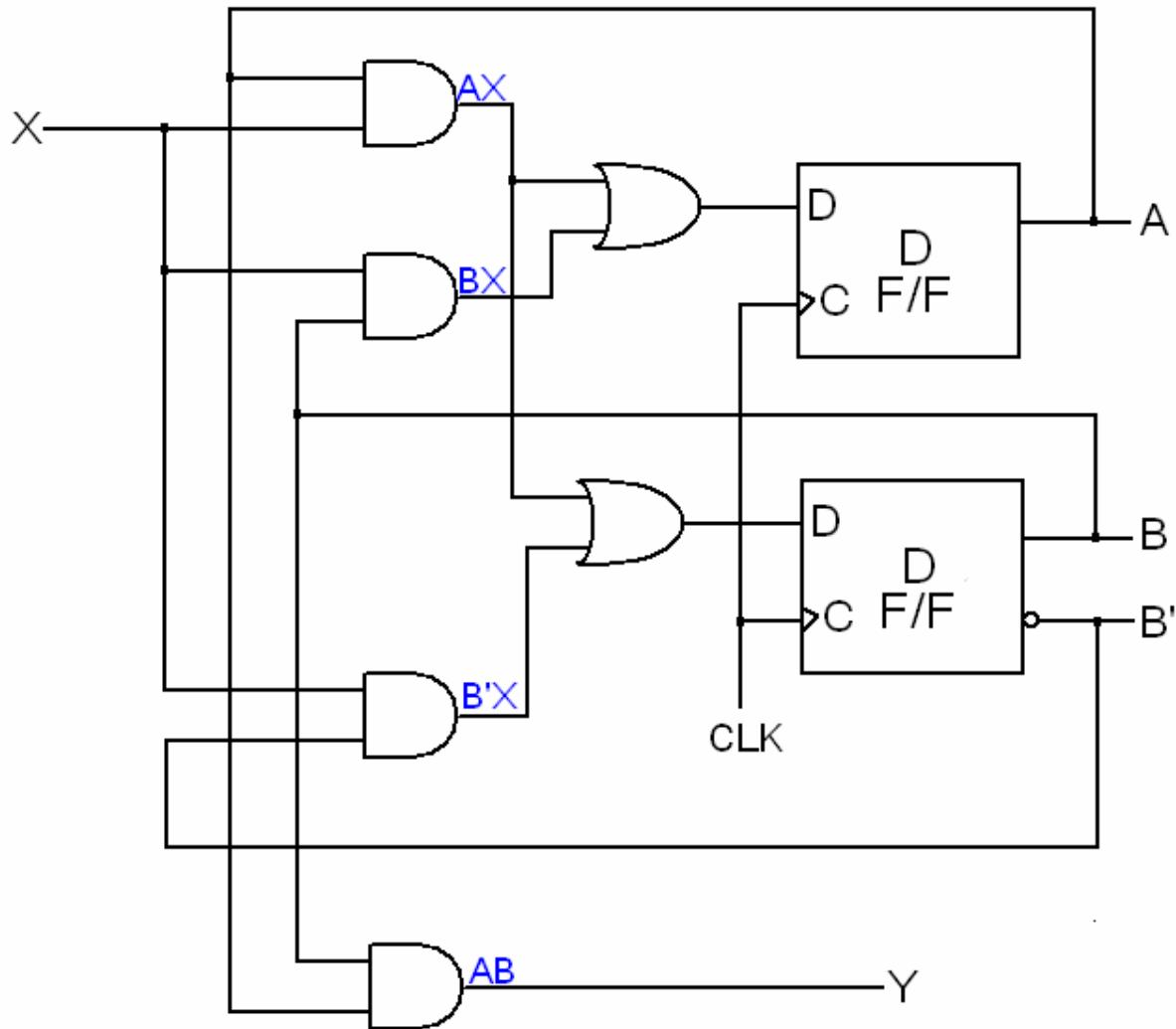


### الحل : Solution

: D Flip flop حل المثال بـ

P.S			I/P	N.S			O/P		
A	B	X		A	B	Y	DA	DB	
0	0	0		0	0	0	0	0	
0	0	1		0	0	0	0	1	
0	1	0		0	0	0	0	0	
0	1	1		1	0	0	1	0	
1	0	0		0	0	0	0	0	
1	0	1		1	0	0	1	1	
1	1	0		0	1	1	0	0	
1	1	1		1	1	1	1	1	

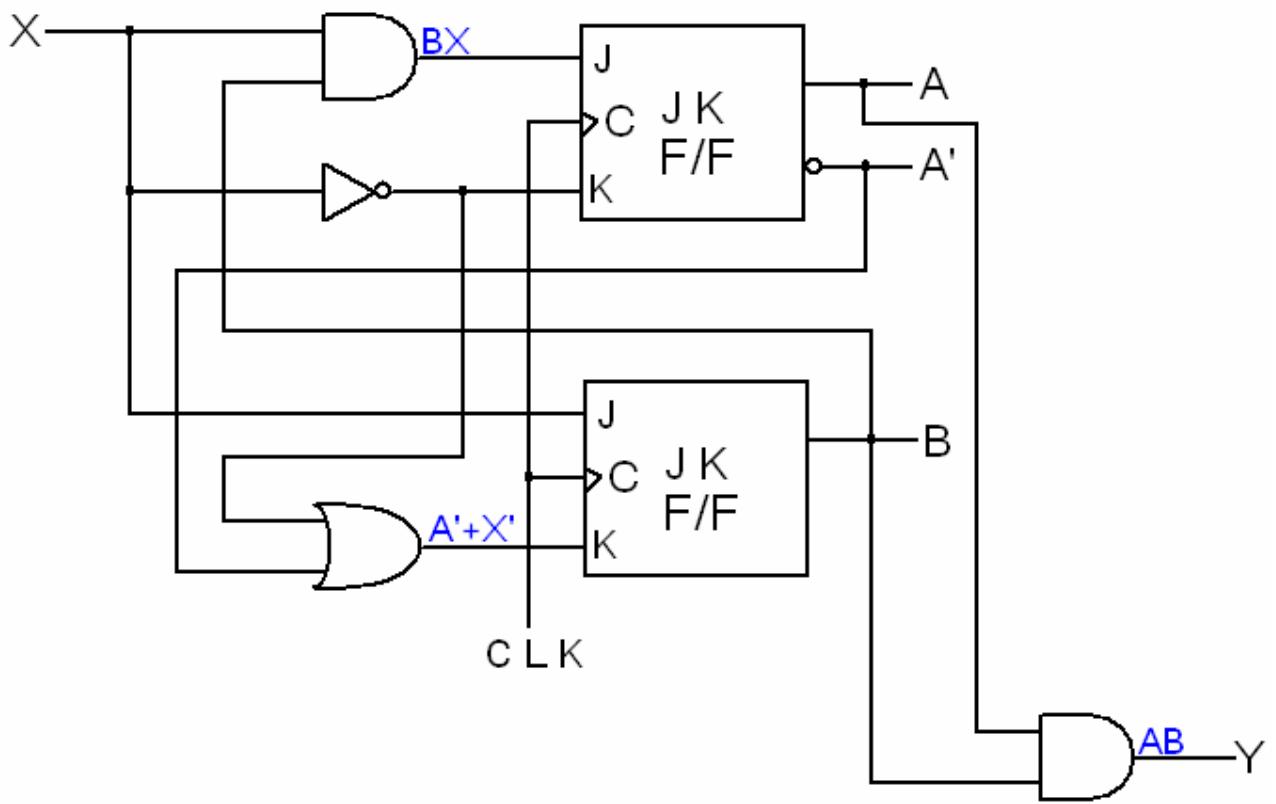
DA	DB																
$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td></td><td>1</td></tr> <tr> <td>1</td><td></td><td>1</td><td>1</td></tr> </table> $DA = AX + BX$	0			1	1		1	1	$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td>1</td><td></td></tr> <tr> <td>1</td><td></td><td>1</td><td>1</td></tr> </table> $DB = AX + B'X$	0		1		1		1	1
0			1														
1		1	1														
0		1															
1		1	1														
<b>Y</b>																	
$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td></td><td></td></tr> <tr> <td>1</td><td></td><td></td><td>1</td></tr> </table> $Y = AB$	0				1			1									
0																	
1			1														



## حل المثال بـ J K Flip flop

P.S			I/P	N.S			O/P				
A	B	X		A	B	Y		JA	KB	JA	KB
0	0	0		0	0	0		0	X	0	X
0	0	1		0	0	0		0	X	1	X
0	1	0		0	0	0		0	X	X	1
0	1	1		1	0	0		1	X	X	1
1	0	0		0	0	0		X	1	0	X
1	0	1		1	0	0		X	0	1	X
1	1	0		0	1	1		X	1	X	1
1	1	1		1	1	1		X	0	X	0

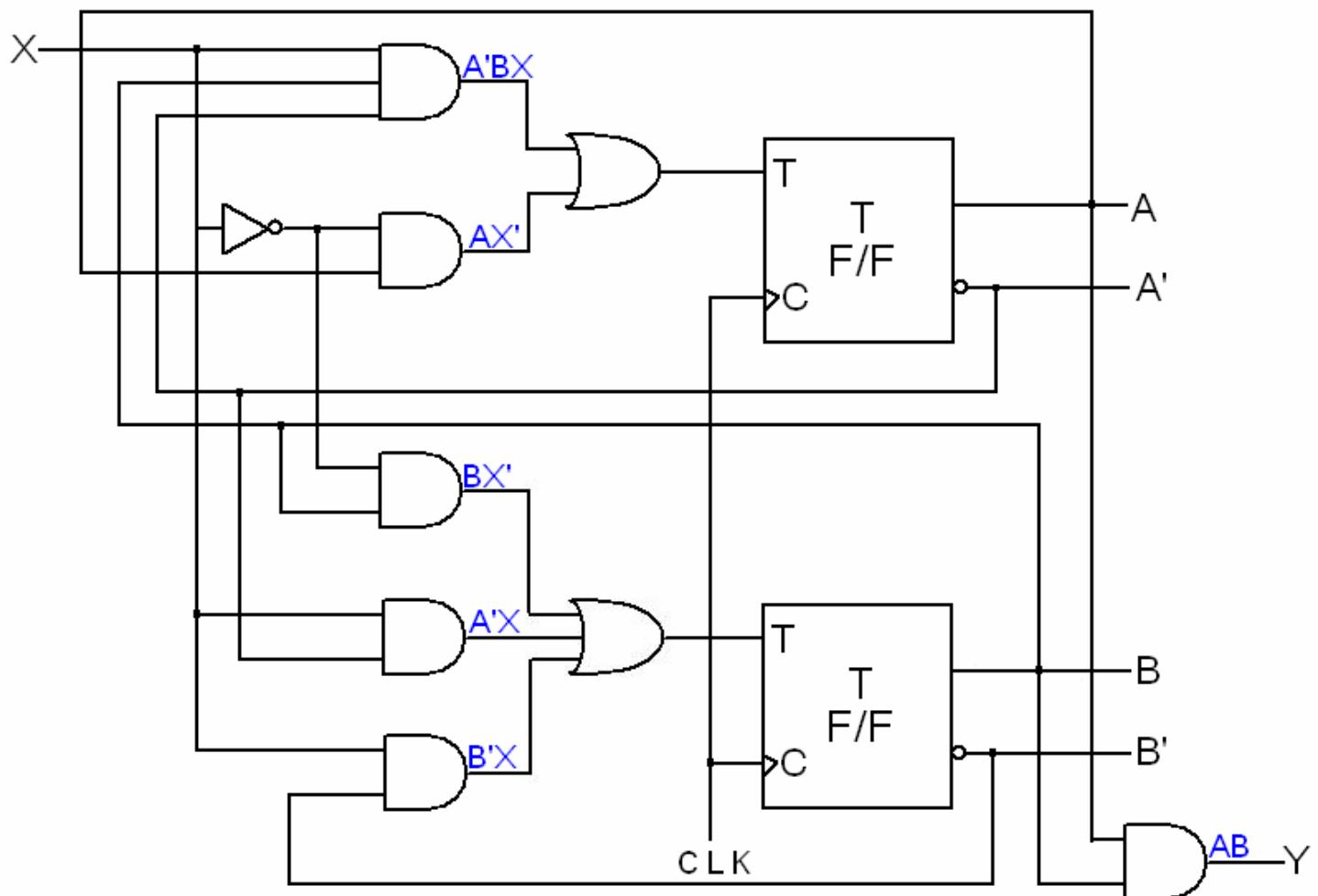
JA				KA							
$A \setminus BX$				$A \setminus BX$							
00 01 11 10				00 01 11 10							
0				0							
1				1							
JA = BX				KA = X'							
JB				KB							
$A \setminus BX$				$A \setminus BX$							
00 01 11 10				00 01 11 10							
0				0							
1				1							
JB = X				KB = A' + X'							
Y											
$A \setminus BX$											
00 01 11 10											
0											
1											
Y = AB											



: حل المثال بـ T Flip flop

P.S		I/P	N.S		O/P		
A	B	X	A	B	Y	TA	TB
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	1
0	1	1	1	0	0	1	1
1	0	0	0	0	0	1	0
1	0	1	1	0	0	0	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	0	0

TA	TB																				
$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td></td><td>1</td><td></td></tr> <tr> <td>1</td><td>1</td><td></td><td></td><td>1</td></tr> </table> $TA = A'BX + AX'$	0			1		1	1			1	$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td>1</td><td>1</td><td></td></tr> <tr> <td>1</td><td></td><td>1</td><td></td><td>1</td></tr> </table> $TA = B'X + A'X + BX'$	0		1	1		1		1		1
0			1																		
1	1			1																	
0		1	1																		
1		1		1																	
<b>Y</b>																					
$A \begin{matrix} BX \\ 00 & 01 & 11 & 10 \end{matrix}$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td></td><td></td><td></td></tr> <tr> <td>1</td><td></td><td></td><td>1</td><td>1</td></tr> </table> $Y = AB$	0					1			1	1											
0																					
1			1	1																	



الفصل السادس

Registers

and

Counters

## ٦-١ مقدمة : Introduction

في هذا الفصل سوف نتعرف على المسجل (Register) ونتعرف على بعض العمليات التي تتم عليه وهي : (Shift) ، (Rotate) ، (Counter) وكيفية تصميمه .

## ٦-٢ المسجل : Register

A7	A6	A5	A4	A3	A2	A1	A0
----	----	----	----	----	----	----	----

الشكل السابق يوضح الشكل العام لـ (Register)

### ١ - Shift Register :

أول العمليات التي تتم على (Register) هي عملية (Shift) وهذه العملية ليست إلا عملية إزاحة لقيم (Register) وبعد عملية الإزاحة إلى جهة كانت اليمين أو اليسار سوف تطرد آخر قيمة في (Register) خارج (Register) وتحل محلها القيمة التي قبلها وتحل محل القيمة قبل الأخيرة القيمة التي قبلها إلى أن نصل إلى أول خانة في (Register) وقد أصبحت خالية من القيم ونضع فيها القيمة (0)

#### مثال : Example

: Shift left R

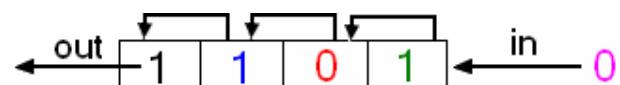
R	1	1	0	1
---	---	---	---	---

### الحل : Solution

المطلوب عمل (Shift) إلى جهة اليمين لقيم (R Register)

تبعد قيم (Register) قبل وبعد عملية (Shift) وتتابع القيم وقد قمت بتلويين القيم لتلاحظ إنطلاق كل قيمة للخانة التي بعدها وإضافة قيمة جديدة وهي القيمة (0)

الشكل التالي يوضح لك بشكل مفصل حل المثال :



وهذا الشكل يوضح الحل النهائي المطلوب إجاده :

R	1	0	1	0
---	---	---	---	---

: Example  
: Shift Right R

R
0   1   0   1

الحل : Solution

المطلوب عمل (Shift) إلى جهة اليمين لقيم (R Register)

R
0   0   1   0

: Rotate Register - ٢

هي العملية الثانية التي تتم على (Register) ولا تختلف عن عملية (Shift) إلا في أن القيمة الموجودة في الخانة الأخيرة والتي كنا نلغيها في عملية (Shift) لن نقوم بإلغائها هنا بل سوف تصبح القيمة الأولى في (Register)

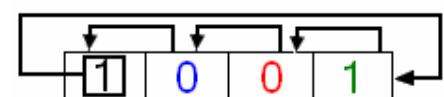
: Example  
: Rotate left R

R
1   0   0   1

الحل : Solution

المطلوب عمل (Rotate) إلى جهة اليسار لقيم (R Register)

الشكل التالي يوضح لك بشكل مفصل حل المثال :



وهذا الشكل يوضح الحل النهائي المطلوب إجاده :

R
0   0   1   1

: Example مثال  
: Rotate Right R

R
0   1   0   1

: Solution الحل  
المطلوب عمل (Rotate Register) إلى جهة اليمين لقيمة R

R
1   0   1   0

: Example مثال  
: Rotate Right R 3 times

R
1   1   0   0   0   1   0   1

: Solution الحل  
المطلوب عمل (Rotate Register) إلى جهة اليمين 3 مرات لقيمة R

R
1   1   1   1   0   0   0   1   0
2   0   1   1   1   0   0   0   1
3   1   0   1   1   1   0   0   0

مثال : Example

: Content of Register A(11010100) shift Register a 4 times or the left with serial input 101100

A							
1	1	0	1	0	1	0	0

الحل : Seloution

المطلوب عمل (Shift) إلى جهة اليسار 4 مرات لقيمة (A Register).

ولكن في هذا المثال لن نضيف صفر في الخانة الأخيرة كما كنا نفعل وذلك لأنه أعطانا في المثال (Serial) ومنه سوف نقوم بتعبئة الخانة الأخيرة.

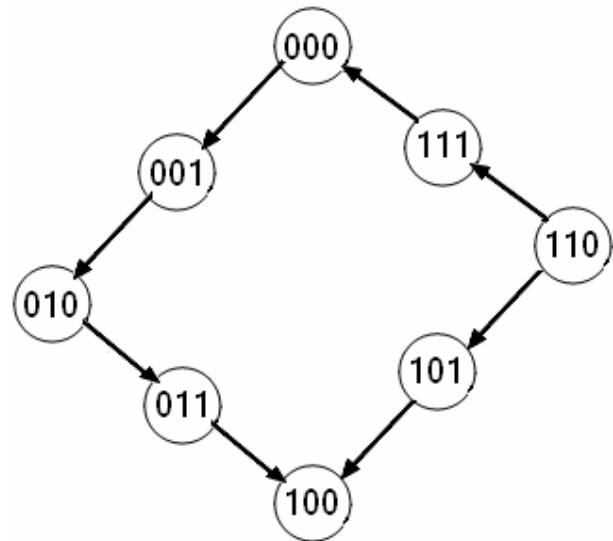
طلب منك أن تقوم بعملية (Shift) 4 مرات وفي كل مرة نأخذ رقم من (Serial) ونضعه في الخانة الأخيرة ونختار أرقام (Serial) من جهة اليسار إلى اليمين.

في المرة الأولى نأخذ آخر رقم (Serial) وهو الرقم 1 ونضعه في الخانة التي أصبحت فارغة من (Register) وهي الخانة الأولى من جهة اليمين.

وفي المرة الثانية نأخذ الرقم قبل الأخير من (Serial) وهو الرقم 0 ونعمل مع مثل ما عاملنا في المرة الأولى وهكذا إلى أن ننتهي من عمل (Shift) 4 مرات وبالتالي تكون قد إنتهينا من حل هذا المثال.

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

مثال : Example  
 : Design a 3-bit Counter using T Flip flop



الحل : Selution  
 المطلوب تصميم عداد يستقبل (3-bit) بإستخدام (T Flip flop).

P.S			N.S					
A2	A1	A0	A2	A1	A0	TA2	TA1	TA0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1
0	1	1	1	0	0	1	1	1
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	1	1	1	0	0	1
1	1	1	0	0	0	1	1	1

## الشرح : explain

أريد أن أبين كيف قمنا بإنشاء جدول (State table) للعداد الموضع برسمة (State diagram) السابقة كل صف مكون 3 أعداد ، وسوف يكون لدينا 8 صفوف .

وذلك حسب العملية الحسابية التالية :

$8 = 2^3$  حيث أن :

الأس وهو العدد 3 يمثل العدد الذي يستقبله العداد وهو (3-bit) الذي يمثل العدد المكون منه كل صف . ونتائج العملية الحسابية السابقة وهو العدد 8 يمثل عدد الصفوف .

لو أن العداد يستقبل (4-bit)  
سوف تنتج لدينا العملية الحسابية التالية :

$$16 = 2^4 \text{ منها نوجد}$$

أن كل صف مكون من 4 قيم ، وأن عدد الصفوف = 16 صف

نرجع لمثالنا وأريد أن أبين أن الطريقة والخطوات المتتبعة في تصميم (Counter) هي نفس الطريقة ونفس الخطوات المتتبعة لحل مسائل (Design Procedure) والتي سبقت أن مرت بنا في الباب الخامس .

حيث أن خطوات الحل سوف تكون كالتالي :

أعطانا في السؤال رسماً (State diagram) لـ (Counter) ومنها سوف تقوم بإجاد جدول (State table) ونقوم بتعبئته خلايا الجدول كما تعلمنا حيث :  
نضع في الأعمدة (P.S) القيم التي سوف تتطرق منها السهم .  
ونضع في الأعمدة (N.S) القيم التي سوف تصل إليها الأسم .

أما أعمدة (T Flip Flop) فنقوم بتعبئتها من خلال التعويض في جدول (Excitation table) الخاص بجدول (T Flip Flop) والذي سبق وأن مر بنا في الباب الخامس وسوف أعيد كتابته في هذه الجزئية لسهولة إجاده عملية تعبئة أعمدة (T Flip Flop) من جدول (State table) المطلوب إجاده .

### Excitation Table

Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

وكما نعلم  $Q(t)$  تمثل قيمة تمثل قيمة (N.S) و  $Q(t+1)$  تمثل قيمة (P.S) :

وأريد أن أبين كيف قمنا بتعبئة العمود (TA2) :  
لإجاد العمود (TA2) يجب علينا أولاً إيجاد العمود (A2) الذي هو جزء من أعمدة (P.S)  
وكذلك إيجاد العمود (A2) الذي هو جزء من أعمدة (N.S)  
وقد قمنا بإيجادها .

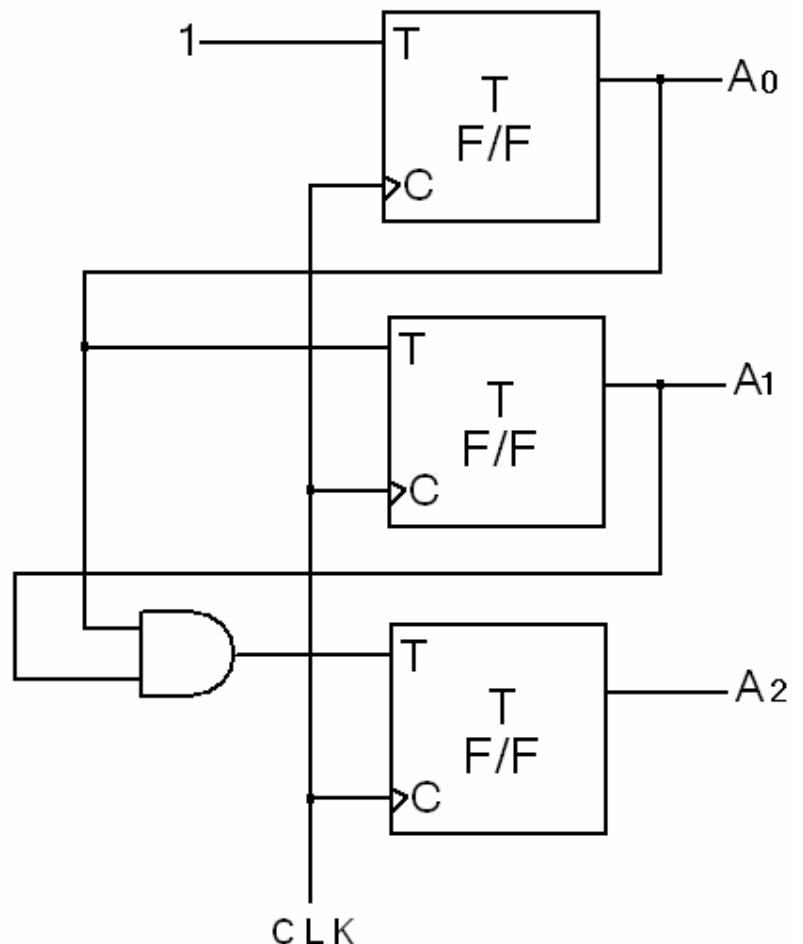
بعد ذلك نقوم بالتعويض في جدول (Excitation table) للعداد الموضع برسمة (State diagram) السابقة .

TA2				TA1			
A <sub>1</sub> A <sub>0</sub>		A <sub>2</sub>		A <sub>1</sub> A <sub>0</sub>		A <sub>2</sub>	
00	01	11	10	00	01	11	10
0			1		1	1	
1			1		1	1	
TA2 = A <sub>1</sub> A <sub>0</sub>				TA1 = A <sub>0</sub>			

TA0							
A <sub>1</sub> A <sub>0</sub>		A <sub>2</sub>					
00	01	11	10	0	1	1	1
0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
TA0 = 1							

. الخطوة التالية نقوم بتبسيط المخرجات وهي أعمدة (T Flip Flop) من جدول (State table).



الخطوة الأخيرة وهي رسم الدوال الناتجة بعد أن قمنا بتبسيطها التي رسمة العداد (Counter) المطلوب تصميمه وبالتالي تكون قد أوجدنا حل المثال السابق.

: Example مثال

: Design a Counter that goes through the following binary repeated sequence : 0,1,2,4,5,6  
using T Flip flop  
الحل : Solution

المطلوب تصميم عداد بإستخدام (T Flip flop)  
حسب تسلسل القيم التالية :

$0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6$

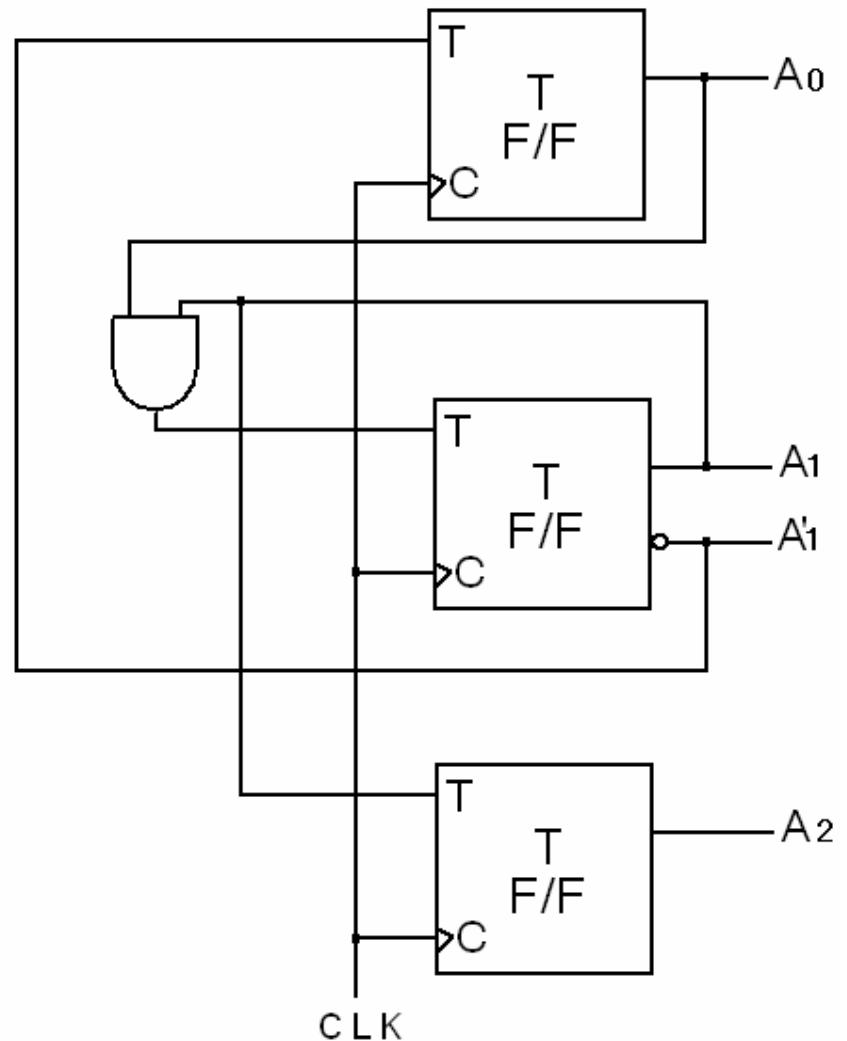
وعند وصولنا للعدد 6 سوف نعود مجدداً إلى أول قيمة وهي العدد 0  
أي أن تسلسل القيم السابقة مثل الحلقة المتصلة وسوف يصبح التسلسل بالشكل التالي :  
 $0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0$

توجد بعض القيم الموجودة في (P.S) ولكن لا يوجد لها قيمة تتجه إليها في (N.S) وهي القيمتين 7 و 3  
أي قيمة في (P.S) لا يوجد لها قيمة تتجه إليها في (N.S) يكون إتجاهها مباشرة إلى (Don't care)

بعد ذلك نقوم بإكمال الحل كما تعلمنا سابقاً .

P.S			N.S					
A2	A1	A0	A2	A1	A0	TA2	TA1	TA0
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	1
0	1	0	1	0	0	1	0	0
0	1	1	X	X	X	X	X	X
1	0	0	1	0	1	0	0	1
1	0	1	1	1	0	0	1	1
1	1	0	0	0	0	1	1	0
1	1	1	X	X	X	X	X	X

TA2	TA1																		
$A_1 A_0$ $A_2 \quad 00 \quad 01 \quad 11 \quad 10$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td>X</td><td>1</td></tr> <tr> <td>1</td><td></td><td>X</td><td>1</td></tr> </table>	0		X	1	1		X	1	$A_1 A_0$ $A_2 \quad 00 \quad 01 \quad 11 \quad 10$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td></td><td>1</td><td>X</td><td>1</td></tr> <tr> <td>1</td><td></td><td>1</td><td>X</td><td>1</td></tr> </table>	0		1	X	1	1		1	X	1
0		X	1																
1		X	1																
0		1	X	1															
1		1	X	1															
$TA_0 = 1$	$TA_1 = A_0$																		
<b>TA0</b>																			
$A_1 A_0$ $A_2 \quad 00 \quad 01 \quad 11 \quad 10$ <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>0</td><td>1</td><td>1</td><td>X</td><td></td></tr> <tr> <td>1</td><td>1</td><td>1</td><td>X</td><td></td></tr> </table>		0	1	1	X		1	1	1	X									
0	1	1	X																
1	1	1	X																
$TA_2 = A_1 A_0$																			



# الخاتمة

أسأل الله العلي القدير أن أكون وفقت في عمل هذا الكتاب وأعطي ثماره المرجوة منه  
أخواني وأخواتي

كل إنسان معرض للخطأ وليس عيباً أن نخطئ بل العيب أن نستمر في الخطأ  
هذا الكتاب أمامكم لا أظمن أن يكون صحيحاً ١٠٠ % ولكن ربما غفلت عن بعض الأخطاء  
أو نسيت بعض الجزئيات

فأرجو منكم جميعاً أن لا تترددوا في إخباري عنها ومراسلي عبر البريد الإلكتروني المذكور في الكتاب  
وكذلك إذا كان لديكم أي ملاحظات وإقتراحات حول العمل أرجو أن لا تخلوا علي بها أيضاً  
وذلك لكي تكون النسخة القادمة بمشيئة الله تعالى من هذا الكتاب أكثر شمولاً وأعم فائدة

وختاماً أسائل الله العلي القدير التوفيق لي ولكم  
وآخر دعونا أن الحمد لله رب العالمين وصلى الله على سيدنا محمد وعلى آله وصحبه وسلم  
والسلام عليكم ورحمة الله وبركاته

أحمد رمضان الزهراني  
طالب جامعة أم القرى  
قسم علوم الحاسوب الآلي  
[Ahamd\\_911@hotmail.com](mailto:Ahamd_911@hotmail.com)