التحليل والتصميم المنطقي

Digital Design

إعداد
أحمد رمضان الزهراني

بسم الله الرحمن الرحيم

المقدمة

بسم الله الرحمن الرحيم

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء وسيد المرسلين

سيدنا ونبينا محمدعليه أفضل الصلاة وأتم التسليم وبعد :

فقد وضعت نصب عيني أثناء كتابتي هذا الكتاب أن يكون شاملاً لكل ما تم دراسته في هذه المادة لكي يستفيد منه كل طالب وطالبة سواءً 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 الحاسب الآلي ومادة عمارة الحاسب الآلي وذلك لأن هنالك 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

لدا المهندس / ماهر الشقنقيري الذي كان خير معلم لها.

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

Digital Design للمؤلف M. Morris Mano

إستغرق هذا العمل مني قرابة الأربعة أشهر وقد واجهت بعض الصعوبات خصوصاً الرسم حيث أنه كان مجهداً 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 ولي ذلك والقادر عليه وصل الله على سيدنا محمد وعلى آله وصحبه وسلم.

أحمد رمضان الزهراني

طالب جامعة أم القرى

قسم علوم الحاسب الآلي

Ahmad_911@hotmail.com

الإهداء

أهدي هذا الكتاب لكلية الحاسب الآلي ونظم المعلومات بجامعة أم القرى

ممثلة في عميدها سعادة الدكتور / محمد الصالح

ووكيلها سعادة الدكتور / سعود مغربي

ورئيس قسم علوم الحاسب الآلي سعادة الدكتور / عدنان أبوعرفة

وجميع أعضاء هيئة التدريس وأخص بالذكر معلمي لهذه المادة المهندس / ماهر الشقنقيري

وطلابها وطالباتها

وكل من يقرأ هذا الكتاب وسوف يستفيد منه بإذن الله تعالى

كما أهدي هذا الكتاب لمنتديات كلية الحاسب ونظم المعلومات www.uqucs.com

المنتدى الغالي والعزيز على قلبي الذي سوف يحتضن كتابي وسيحظى بشرف رعاية المنتدى له

الفهرس

الفصل الأول

Binary Systems

١-١ مقدمة Introduction :

في هذا الفصل سوف نتحدث عن أنظمة الأعداد (Digital systems) وهي :
النظام العشري (Decimal system) و النظام الثنائي (Binary system) و النظام الثماني (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 الثماني	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

٣-١ 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 :

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

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 :

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

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 مرفوعا للأسس..0,1,2,3
والعدد الكسري مرفوعاً للأسس...3-,2-,1- ثم نجمع العدد كاملاً .

مثال Example :

حول العدد الثنائي التالي إلى النظام العشري :

$(1100.101)_2$

الحل Selution :

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

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

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

مثال Example :

حول العدد الثماني التالي إلى النظام العشري :

$(752)_8$

الحل Selution :

$$(752)_8 = 7*8 + 5*8^1 + 2*8^2$$
$$= 7 + 40 + 448$$
$$= (490)_{10}$$

مثال Example :

حول العدد الثماني التالي إلى النظام العشري :

$(35.6)_8$

الحل Selution :

$$(35.6)_8 = 5*8 + 3*8^1 + 6*8^{-1}$$
$$= 5 + 24 + 0.75$$
$$= (29.75)_{10}$$

٣- من النظام الست عشري إلى النظام العشري Hexadecimal To Decimal :

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

مثال Example :

حول العدد الست عشري التالي إلى النظام العشري :

$(ABC)_{16}$

الحل Selution :

$$(ABC)_{16} = 12*16 + 11*16^{1} + 10*16^{2}$$
$$= 12 + 176 + 2560$$
$$= (2748)_{10}$$

مثال Example :

حول العدد الست عشري التالي إلى النظام العشري :

$(2F.8)_{16}$

الحل Selution :

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

الخلاصة :

عند التحويل من أي نظام (الثنائي أو الثماني أو الست عشري) 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 Selution :

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

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

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 القسم 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

$14 / 2 = 7$ ولا يوجد باقي يكون الباقي = 0

٢- عند كتابة ناتج عملية التحويل يكتب العدد من الأسفل إلى الأعلى كما فعلنا في مثالنا السابق .

مثال Example :

حول العدد العشري التالي إلى النظام الثنائي :

$(0.78125)_{10}$

الحل Selution :

0.78125	* 2 = 1.5625	→ 1
0.5625	* 2 = 1.125	→ 1
0.125	* 2 = 0.25	→ 0
0.25	* 2 = 0.5	→ 0
0.5	* 2 = 1	→ 1

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

الشرح explain :

في هذا المثال قمنا بتحويل العدد (0.78125) من النظام العشري إلى النظام الثنائي طريقة التحويل هي كالآتي :
كما تلاحظون أن العدد عدد كسري وليس صحيح إذا سوف تتغير طريقة تعاملنا معه خلافا 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 عملة 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 كذلك 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 ولابد من أن يكون العدد بعد التحويل عدد كسري وكما نعلم أن العدد الكسري يكتب بعد الفاصلة .

مثال Example :

حول العدد العشري التالي إلى النظام الثماني :

$(35.375)_{10}$

الحل :

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

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

(100011)

0.375	* 2 = 0.75	→ 0
0.75	* 2 = 1.5	→ 1
0.5	* 2 = 1	→ 1

(0.011)

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

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

مثال Example :

حول العدد العشري التالي إلى النظام الثماني :

$(153.6875)_{10}$

الحل Selution :

8	
153	
19	1
2	3
0	2

(231)

0.6875 * 8 = 5.5 → 5

0.5 * 8 = 4 → 4

(0.54)

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

الشرح explain :

لا تختلف طريقة التحويل العدد من النظام العشري إلى النظام الثماني 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 هنالك 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 المحول 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 ثماني فهي مثل طريقة تحويله إلى النظام الثنائي تماماً .

٦- من النظام العشري إلى النظام الست عشري 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 وهنالك نقطة بسيطة وهي كما سوف تلاحظون في المثال أن باقي عملية قسمة العدد 125 = 13 وهذا العدد لا يكتب بصورته المعروفة وإنما يكتب بصورته في النظام الست عشري أي نكتب (D) .

مثال Example :

حول العدد العشري التالي إلى النظام الست عشري :

$(125.34375)_{10}$

الحل Selution :

	16	
125		
7	13	
0	7	
(7D)		

$0.34375 * 16 = 5.5 \rightarrow 5$

$0.5 * 16 = 8 \rightarrow 8$

(0.58)

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

الخلاصة :

عند التحويل من النظام العشري إلى أي نظام (الثنائي أو الثماني أو الست عشري) 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 Selution :

010	011	101	110
2	3	5	6

$(10011101110)_2 = (2356)_8$

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 أوعددين ثنائيين بمفردهما دون الثالث نقوم نحن بإضافة صفر أو صفرين 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 قمنا 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 الثماني :

$(.0101111)_2$

الحل Selution :

100	111	010
4	3	2

$(.0101111)_2 = (.234)$

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 الثماني :

$(11001.01)_2$

الحل Selution :

.010	001	011
2	1	3

$(11001.01)_2 = (31.2)_8$

الشرح explain :

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

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

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

مثال Example :

حول العدد الثماني التالي إلى النظام الثنائي :

$(62.7)_8$

الحل Selution :

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

مثال Example :

حول العدد الثماني التالي إلى النظام الثنائي :

$(35.41)_8$

الحل Selution :

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

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

هي نفس طريقة التحويل السابقة من النظام الثنائي إلى النظام الثماني 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 Selution :

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

مثال Example :

حول العدد الثنائي التالي إلى النظام الست عشري :

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

الحل Selution :

$(1111\ \ 1100.\ 0101\ 1011)_2 = (FC.5B)_{16}$

١٠- من النظام الست عشري إلى النظام الثنائي Hexadecimal To Binary :

مثال Example :

حول العدد الست عشري التالي إلى النظام الثنائي :

$(AB.6D)_{16}$

الحل Selution :

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

مثال Example :

حول العدد الست عشري التالي إلى النظام الثنائي :

$(9C.8F3)_{16}$

الحل Selution :

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

٤-١ 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 :

$(1100101001)_2$

الحل Selution :

المتممة الأولى للعدد الثنائي السابق = 0011010110

مثال Example:

أوجد المتممة الأولى للعدد الثنائي التالي :

$(10000000000)_2$

الحل Selution :

المتممة الأولى للعدد الثنائي السابق = 01111111111

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 الطريقتين :

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

مثال Example :
أوجد المتممة الثانية للعدد الثنائي التالي :
$(1100101001)_2$
الحل Selution :
أولا : نوجد المتممة الأولى للعدد الثنائي السابق = 0011010110
ثانيا : 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 :
$(1111000000)_2$
الحل Selution :
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 بكاتبته 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 :

$(1100101001)_2$

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 :

$(1111000000)_2$

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 بعملة 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 Selution :

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

1101 - 1100 = +1001

الشرح explain :

في هذا المثال قمنا بإجراء عملية الطرح على العددين الثنائيين السابقين.
العدد الأول 1101 موجب لذلك نبقيه 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 بايت والعدد الثاني يمثل في 4 بايت والناتج 5 بايت ونحن نريد أن يتمثل الناتج في 4 بايت لذلك نقوم بحذف آخر عدد في الناتج $\cancel{1}1001$ ونضع إشارة + أمام الناتج .

مثال Example:

أجري عملية الطرح التالية :

0110 - 1100

الحل Selution :

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

0110 - 1100 = -0110

الشرح explain :

هذا المثال لا يختلف عن السابق إلا في أن الناتج لا يوجد به (Overflow) أي أنه يتمثل في 4 بايت مثل العددين 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 نتعمق في تبسيط الدول بهذه الطريقة لأن هنالك طريقة أسهل و أوضح تدعى (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 و عملها
وأيضا سوف نتحدث عن (Minterms) و (Maxterms) 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

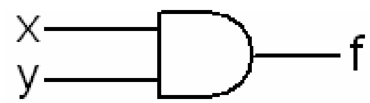
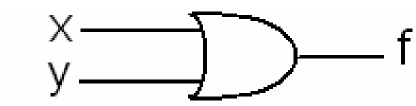
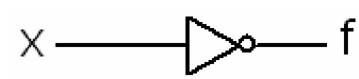
٢-٣ القواعد Grammars :

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

: Logic Gates ٤-٢

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

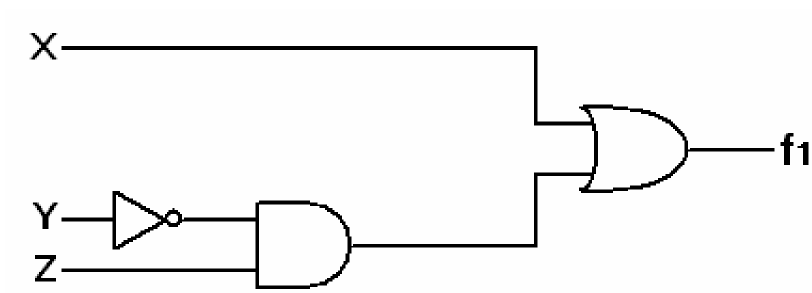
الجدول الشكل السابق يوضح كيفية تمثيل الروابط بالرسم وذلك لكي نتمكن من تمثيل الدوال بالرسم .

مثال Example :

أرسم الدالة التالية :

$F_1 = x + y'z$

الحل Selution :

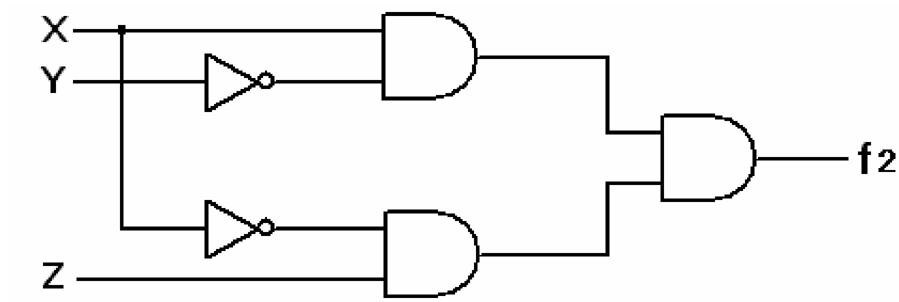


مثال Example :

أرسم الدالة التالية :

$F_1 = xy' + x'z$

الحل Selution :



مثال Example :

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 Selution :

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 .

1- $x(x' + y) = xx' + xy$ $= 0 + xy$ $= xy$	2- $x + x'y = (x+x').(x+y)$ $= 1 .(x+y)$ $= x + y$
3- $(x+y)(x+y') = x(x+y') + y(x+y')$ $= xx + xy' + xy + yy'$ $= x+xy' +xy +0$ $= x(1+y'+y)$ $= x1$ $= x$	4- $xy + x'z + yz = xy + x'z + yz.(x+x')$ $= xy + x'z + xyz + x'yy$ $= xy(1+z) + x'z(1+y)$ $=xy +x'z$

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 Selution :

$$F1 = (x'yz' + x'y'z)'$$
$$= (x'yz')' \,.\, (x'y'z)'$$
$$= (x+y'+z) \,.\, (x+y+z')$$

الشرح explain :
كما تلاحظون أن المثال السابق مكون من حدين .
وخطوات الحل التالي :
١- تغيير الرابط بين الحدين من (OR) إلى (AND) .
٢- نفي كل حد على حدة .
٣- تغيير الروابط 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 :

$$F1 = (x+y'+z').(x'+y+z).(x'+y'+z')$$

الحل Selution :

$$F1 = ((x+y'+z').(x'+y+z).(x'+y'+z'))'$$
$$=(x+y'+z')' + (x'+y+z)' + (x'+y'+z')'$$
$$= (x'yz) + (xy'z') + (xyz)$$

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

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

مثال Example :

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

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 Selution :

Sum of Products ⇔ Minterms

Products of Sum ⇔ Maxterms

(جمع كميات مضروبة)Sum of Products:

$F1 = x'y'z + xy'z' + xyz = m1 + m4 + m7 \rightarrow \Sigma(1,4,7)$

$F2 = x'yz + xy'z + xyz' + xyz = m3 + m5 + m6 + m7 \rightarrow \Sigma(3,5,6,7)$

(ضرب كميات مجموعة)Products of Sum :

$F1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y+'z) = M0.M2.M3.M4 \rightarrow \prod(0,2,3,5,6)$

$F2 = (x+y+z)(x+y+z')(x+y'+z)(x'+y+z) = M0.M1.M2.M4 \rightarrow \prod(0,1,2,4)$

مثال Example :

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

الحل Selution :

هنالك طريقتان للحل 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 نغير في المسألة من تلقاء أنفسنا ونضيف الحدود الناقصة مباشرة لذلك نقوم نحن بالتحايل على المسألة ونضرب في 1 ولنا الحق في اختيار القيمة التي =1 حسب ما نحتاج للتوصل لحل المسالة لذلك نضرب المتغير A في (B+B')
ينتج من عملية الضرب حدين وهما (AB) و(AB') نقوم بضربها في (C+C')
وسوف ينتج من عملية الضرب أربعة حدود .
ثم ننتقل للحد الثاني (B'C) وكما تلاحظون ينقص حذا 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 .

مثال Example :

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

الحل Selution :

الطريقة الأولى :

$$
\begin{aligned}
F &= xy + x'z \\
&= (x+x'z)(y+x'z) \\
&= (x+x')(x+z)(y+x')(y+z) \\
&= 1(x+z)(y+x')(y+z) \\
&= (x+z)(y+x')(y+z) \\
&= (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')(x+y+z)(x'+y+z) \\
&= (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z') \\
&= M0 \;.\; M2 \;.\; M3 \;.\; M4 = \prod(0,2,4,5) = \Sigma(1,3,6,7)
\end{aligned}
$$

الشرح explain :

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 توزع (AND) و(OR) والعكس .

عند حل هذه المسألة قمنا بتوزيع الحد الأول (xy) على الحد الثاني (x'z) نتج عن عملية التوزيع 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 .

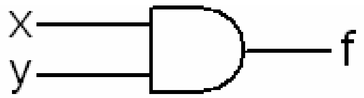
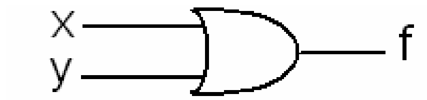
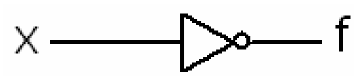
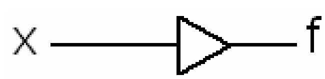
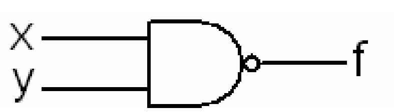
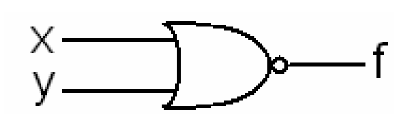
الطريقة الثانية :

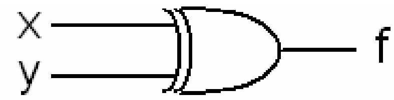
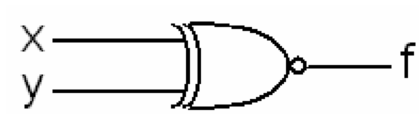
$$
\begin{aligned}
F &= xy + x'z \\
&= xyz + xyz' + x'yz + x'y'z \\
&= m7 + m6 + m3 + m1 = \Sigma(1,3,6,7) = \prod(0,2,4,5)
\end{aligned}
$$

الشرح explain :

كما سبق أن ذكرت أن (Minterms) يمثل قيمة الدالة عندما تكون قيمة الدالة = 1
و(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 لـ (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) .

٧-٢ : Digital Logic Gates

Name	Graphic Symbol	Algebraic Function	Truth Table
AND		F = xy	X Y F 0 0 0 0 1 0 1 0 0 1 1 1
OR		F = x+y	X Y F 0 0 0 0 1 1 1 0 1 1 1 1
Inverter		F = x'	X F 0 1 1 0
Buffer		F = x	X F 0 0 1 1
NAND		F = (xy)'	X Y F 0 0 0 0 1 1 1 0 1 1 1 1
NOR		F = (x+y)'	X Y F 0 0 0 0 1 0 1 0 0 1 1 1

<table>
<tr><td>XOR</td><td>
</td><td>F = xy' + x'y
= x ⊕ y</td><td><table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>XNOR</td><td>
</td><td>F = xy' + x'y
= x ⊕ y</td><td><table><tr><th>X</th><th>Y</th><th>F</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table></td></tr>
</table>

الفصل الثالث

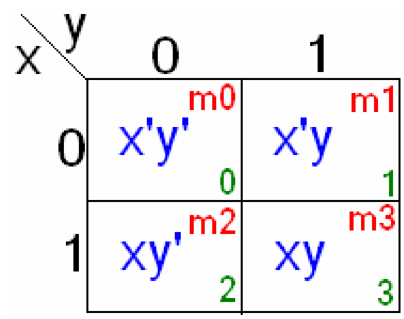
Gate _ Level Minimization

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



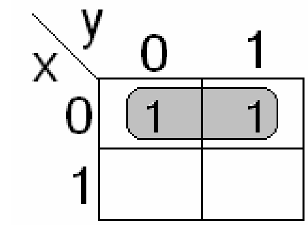
الشكل السابق يوضح الشكل العام لطريقة (karnaugh map) لمتغيرين .
تلاحظون أن (X) لها قيمتين 0,1 وكذلك (Y) لها نفس القيم .
تقاطع قيم (X) مع قيم (Y) تكون قيم هذه الخريطة .
(X) يمثل الجانب العمودي بينما (Y) يمثل الجانب الأفقي .

مثال Example :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل Selution :



$F(x,y) = x'$

الشرح explain :

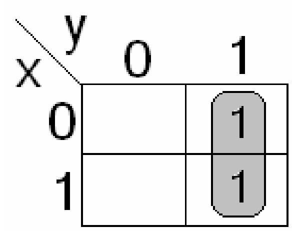
في المثال السابق قمنا بتبسيط الدالة $x'y + x'y'$ وكما تلاحظون الدالة مكونة من حدين .
نأخذ الحد الأول $(x'y)$ ونلاحظ أنه مكون من تقاطع قيمة (X) وهي القيمة 0 مع قيمة (Y) وهي القيمة 1
نضع في مربع تقاطع قيمة (X) مع قيمة (Y) وهو المربع رقم 1 نضع فيه القيمة 1

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 قمنا بتحديد المربعات التي تمثل قيمة الدالة .

والآن نريد إختصار الدالة :
لدينا مربعين تحتوي على القيمة 1 وبما أن المربعين بجانب بعضهما نأخذهما مع بعضهما لنقوم بإختصارهما
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
نتدرج على هذا التسلسل ولا ننتقل من أولوية إلى الأخرى إلى إذا عجزنا تماماً وإلا سوف يكون تبسيطنا للدالة خاطئ.

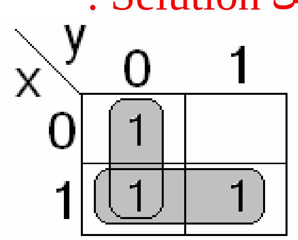
نرجع للدالة ولإختصارها أولا ننظر ماذا تمثل بالنسبة للجانب العمودي وفي هذا المثال لدينا عمودياً فقط (X)
ولـ (X) قيمتين 0 وتمثل قيمة X' و 1 وتمثل قيمة X وعند الإختصار لابد أن تكون قيم (X) متشابهة
أقصد إما أن تكون قيم (X) كلها 0 أو 1 لكي تكتب في الناتج أما إذا كانت مختلفة فإننا نشطب (X) من الناتج
تماماً 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 .

مثال Example :
بسط الدالة المنطقية التالية Simplify the following Boolean function :
F(x,y) = xy + x'y
الحل Selution :



F(x,y) = y
الشرح explain :
في المثال السابق قمنا في البداية بتعبئة المربعات كما تعلمنا في المثال السابق أما بالنسبة للإختصار فقد حدث العكس تماماً للمثال الذي قبله حيث أن قيمة (X) مختلفة مرة 0 ومرة 1 بالنسبة للمربعات 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) .

مثال Example :
بسط الدالة المنطقية التالية Simplify the following Boolean function :
F(x,y) = x'y' + xy' + xy
الحل Selution :



F(x,y) = x + y'
الشرح explain :
في المثال السابق قمنا في البداية بتعبئة المربعات حيث قمنا بتعبئة ثلاث 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 إختياره مع 1 الواقع في المربع رقم 0 وكذلك مع 1 الواقع في المربع رقم 3 لذلك لاننظر إليه الآن وإنما ننظر لـ 1 لانستطيع أخذه إلا مع 1 آخر وينطبق الشرط السابق على 1 المتواجد في المربع رقم 0 وكذلك المربع رقم 3 طبعا أنا حبيت أذكر هذه النقطة لأن عند تعاملنا مع ثلاث وأربع متغيرات سوف تكون العملية أكثر تعقيد في إختيار 1s

بعد إختيار 1s نقوم بعملية التبسيط كما فعلنا في الأمثلة السابقة
ونستطيع إستخدام 1 أكثر من مرة إذا إحتجنا لذلك ولا نختار 1بمفردة إلا إذا عجزنا تماما عن وجود 1 بجانبة يمكن إختياره معه .

مثال Example :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل Selution :

x \ y	0	1
0	1	1
1	1	1

F(x,y) = 1

الشرح explain :

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

٣- خريطة ثلاث متغيرات Three Variables Map:

x \ yz	00	01	11	10
0	m0 x'y'z' 0	m1 x'y'z 1	m3 x'yz 3	m2 x'yz' 2
1	m4 xy'z' 4	m5 xy'z 5	m6 xyz 7	m7 xyz' 6

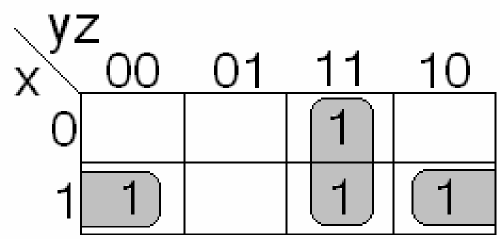
الشكل السابق يوضح الشكل العام لطريقة (karnaugh map) لثلاث متغيرات .
(X) يمثل الجانب العمودي بينما (YZ) تمثل الجانب الأفقي .
هنالك ملاحظة مهمة وهي :
١- ترتيب المربعات مختلف عن المعتاد حيث أن بعد المربع رقم 1 يأتي المربع رقم 3 ثم المربع رقم 2 وكذلك المربع رقم 5 ثم المربع رقم 7 ثم المربع رقم 6 ، أي الترتيب 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 و2 نستطيع أن تأخذهما مع بعضهما وذلك لأنهما بجانب بعضهما ، وكذلك المربعين 4 و 6 ينطبق عليهما نفس الكلام السابق .

مثال : Example

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل : Selution



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

الشرح : explain

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

بعد تعبأتنا 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 (Y, Z)
ونتعامل مع كل متغير على حدة ننظر أولاً ماذا يمثل بالنسبة للمتغير(Y)
ونلاحظ أن قيمة (Y) إختلفت 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 إتنهينا من أول مستطيل

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

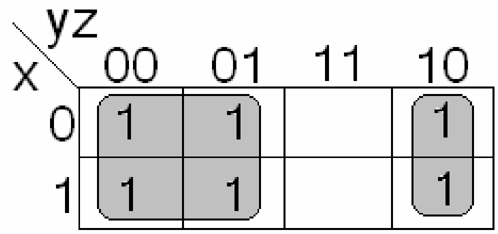
ثم بعد ذلك نكتب ناتج عملية التبسيط ويربط بين كل مستطيل تم تبسيطة مع الآخر علامة +
الناتج = xz' + yz
الحد الأول يمثل إختصار المستطيل الأول والحد الثاني إختصار المستطيل الثاني
وعلامة + تربط بين الحدود الناتجة عن عملية تبسيط المستطيلين السابقة .

مثال : Example

بسط الدالة المنطقية التالية Simplify the following Boolean function :

$F(x,y,z) = \Sigma(0,1,2,4.5,6)$

الحل : Selution



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

مثال : Example

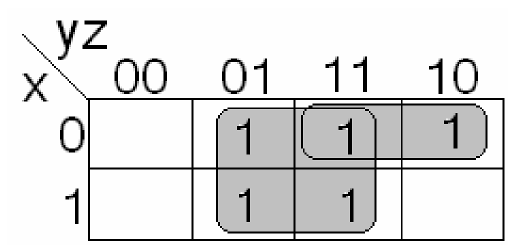
Given the Boolean function :

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

١- Express it in sum of minterms

٢- Find the minimal sum of products expression

الحل : Selution



١- $F(x,y,z) = \Sigma(1,2,3,5,7)$

٢- $F(x,y,z) = C + A'B$

الشرح : explain

في المثال السابق إختلفت علينا صغية السؤال والمطلوب إيجادة،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.

٣-٤ خريطة أربع متغيرات Four Variables Map

wx \ yz	00	01	11	10
00	m0 w'x'y'z' 0	m1 w'x'y'z 1	m3 w'x'yz 3	m2 w'x'yz' 2
01	m4 w'xy'z' 4	m5 w'xy'z 5	m6 w'xyz 7	m7 w'xyz' 6
11	m12 wxy'z' 12	m13 wxy'z 13	m15 wxyz 15	m14 wxyz' 14
10	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) لثلاث 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 .

مثال Example :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

$$F(w,x,y,z) = \Sigma(0,1,2,4,5,6,8,9,12,13,14)$$

الحل Selution :

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

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

الشرح explain :

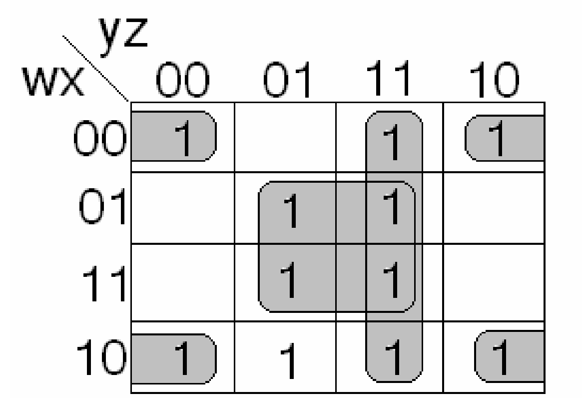
المثال السابق لا يختلف عن الأمثلة السابقة وتعاملنا مع (karnaugh map) لمتغيرين أو ثلاث متغيارت .
طريقة الحل واحدة ولا تختلف الجديد في الأمر زيادة عدد المتغيارت وكبر الحجم ونعاملها مثل ما تعلمنا سابقاً من تعبئة المربعات وطرق إختيار المستطيلات وكيفية الإختصار .

: Example مثال

: Simplify the following Boolean function بسط الدالة المنطقية التالية

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

: Selution الحل



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

لاحظ أنه يمكننا إختيار المربعات الموجودة في الأطراف وإختصاراها مع بعضها وذلك لأنها تعتبر مربعات متجاورة

: Don't Care Conditions ٥-٣

: Example مثال

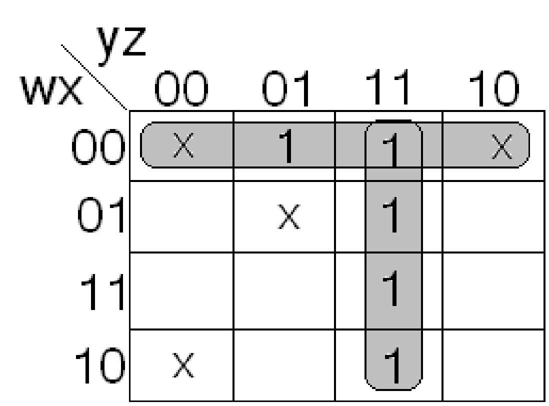
: Simplify the following Boolean function بسط الدالة المنطقية التالية

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

Which has the don't care conditions

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

: Selution الحل



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

الشرح explain :

في المثال السابق نلاحظ شي 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 إحتجنا لإستخدامها نستخدمها عكس 1s فإنك إذا لم تقوم بتغطيتها بالكامل فإن حلك يعتبر خاطئ .

نرجع للمثال لقد قمنا بتعبئة مربعات (karnaugh map) بقيم 1s ثم تعبئتها بقيم (Don't care) بعد ذلك نقوم بإختيار المربعات ونقوم بعملية الإختصار .
المربعات نتج لدينا مستطيلين المستطيل الأول المستطيل العمودي المكون من3,7,15,11
وهذا المستطيل طريقة حله مثل السابق ولن نتطرق إليه .

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

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

عند وجود (Don't care) يتوجب عليك إستخدامها إذا إحتجت 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 .

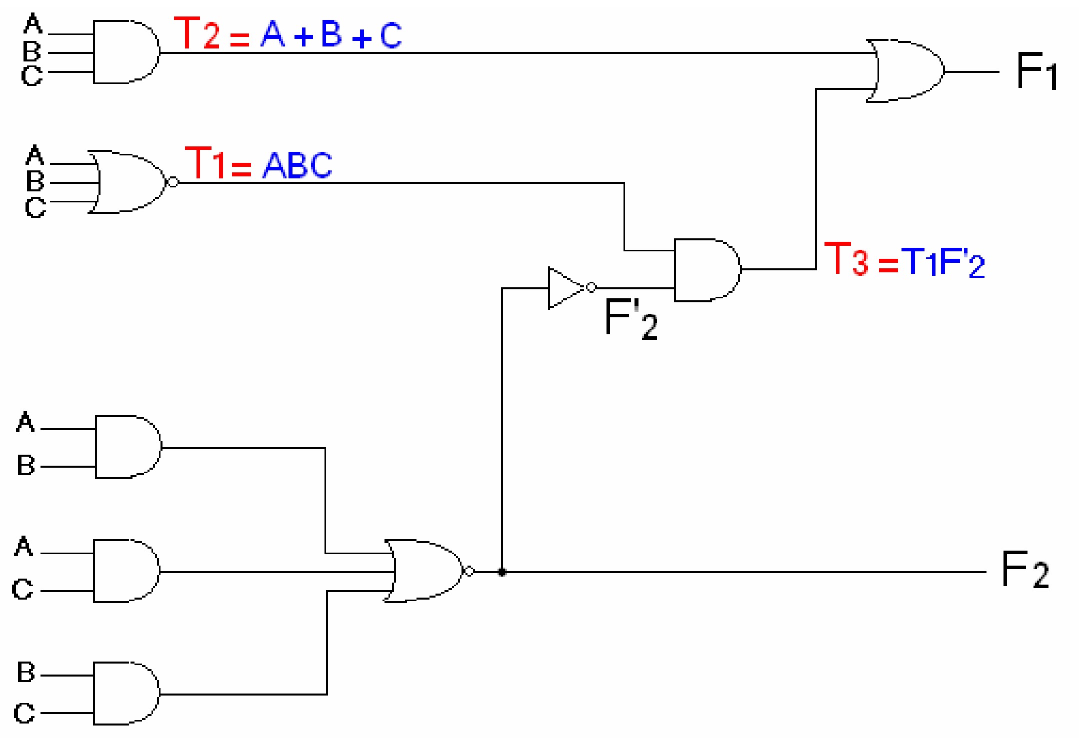
الفصل الرابع

Combinational Logic

٤-١ مقدمة Introduction :

في هذا الفصل سوف نتحدث عن تحليل الدوال (Analysis) وإجاد مخرجاتها .
وكذلك سوف نتحدث عن تصميم الدوال (Design) وحل مسائل التصميم (Design).
كما سوف نتعرف على بعض الدوال وهي :
(Half Adder) ، (Full Adder) ، (Decoder) ، (Multiplexer)
وسوف تحدث عن تعريفها وأشكالها وأنواعها أقسامها وعملها و حل المسائل بواستطها .

٤-٢ إجراء التحليل Analysis Procedure :



$T_1 = A + B + C$

$T_2 = ABC$

$F_2 = AB + AC + BC$

$T_3 = T_1F'$

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

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

$F_1 = T_2 + T_3$

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

الشرح explain :

الشكل السابق يبين شكل دالتين F1,F2 بالرسم وطلب منك إيجاد مدخلات وحدود كل دالة .
وكما تلاحظون أن عدد المدخلات كثير وبالتالي قد تحصل هنالك بعض اللخبطة ونسيان متغير
وبالتالي تضيع 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 إحتجنا لذلك وخشينا من خلط بين
المتغيارت أو نسيان متغير .

٣-٤ 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)
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) لـ4 مدخلات بعد ذلك ننظر لكل صف مكون من قيم المتغيرات A,B,C,D
ونضيف (نجمع)علية الرقم 3 لينتج لنا أول صف من المخرجات لكل من قيم المتغيارت W,X,Y,Z
مثل :

١- أول صف من المدخلات قيمته (0) 0 0 0 0 قمنا بإضافة الرقم 3 عليه ليعطينا المخرجات (3)1 1 0 0
٢- الصف الخامس من المخلات قيمته (4) 0 0 1 0 قمنا بإضافة الرقم 3عليه ليعطينا المخرجات (7) 1 1 1 0

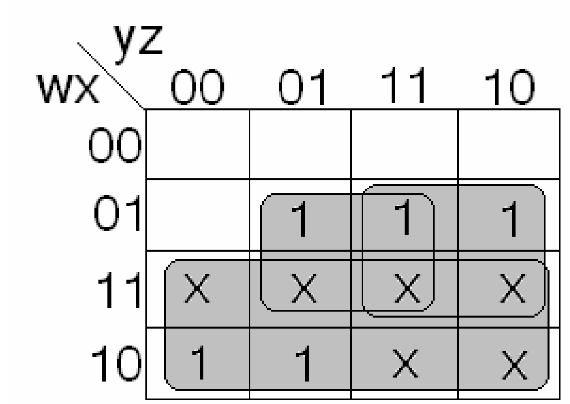
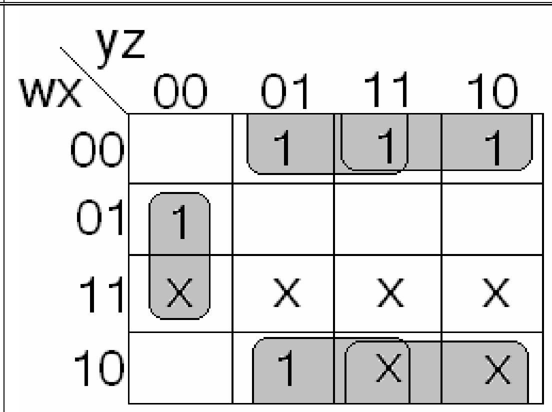
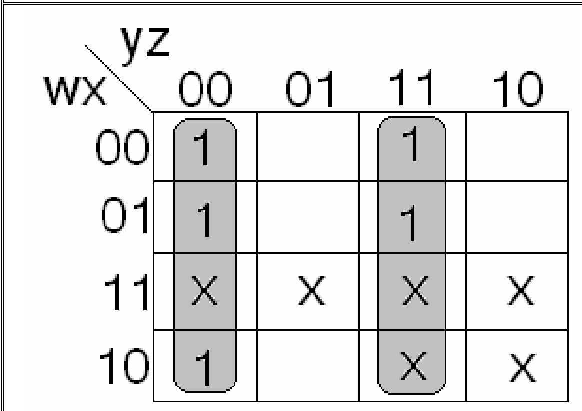
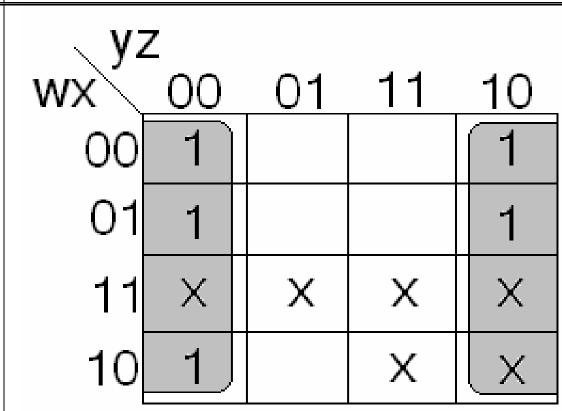
الخطوة الأولى والثانية :

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

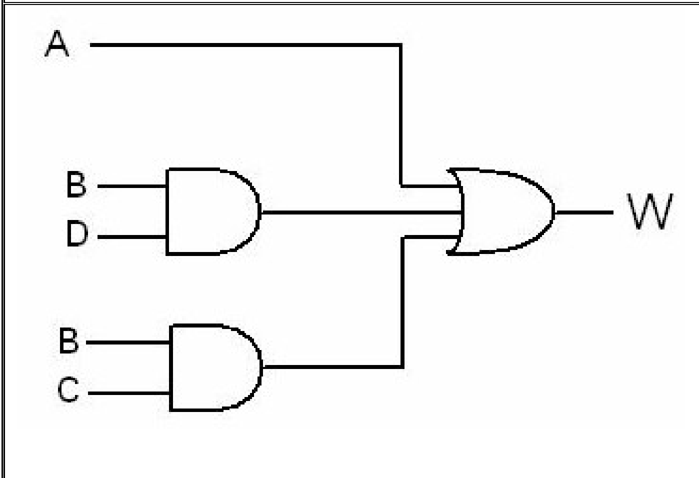
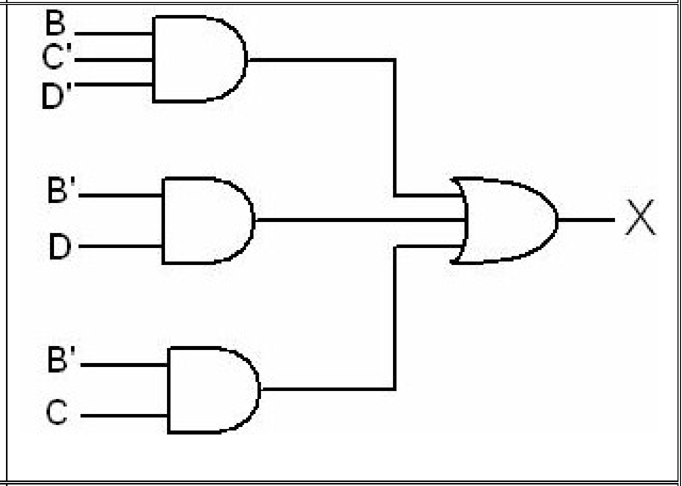
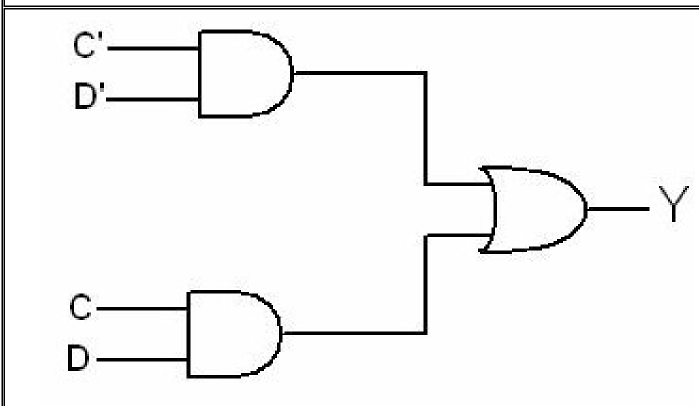
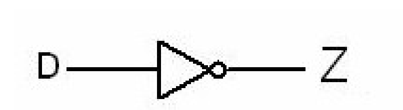
الخطوة الثالثة :

وفيها نبسط المخراجت الناتجة من الجدول السابق.

W	X
 W = A + BD + BC	 X = BC'D' + B'D + B'C
Y	**Z**
 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 .

<table>
<tr><th>W</th><th>X</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th>Y</th><th>Z</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

٤-٤ Half Adder :

سوف نقوم بتعريفها في الجدول التالي :

Truth Table

InPuts		OutPuts	
X	Y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

(S) إختصار لكلمة (Sum) وفي هذه الخانة من الجدول نضع قيمة جمع كل صف من المدخلات أي X+Y

(C) إختصار لكلمة (Carry) إذا وجد (Carry) نضع في هذه الخانة 1 وإذا لم يوجد نضع 0 ويوجد (Carry) إذا كان مجموع عملية جمعX+Y أكبر من 1 كما في آخر صف في الجدول

Algebraic Function

تبسيط مخرجات الجدول

(A)

$S = xy' + x'y$

$C = xy$

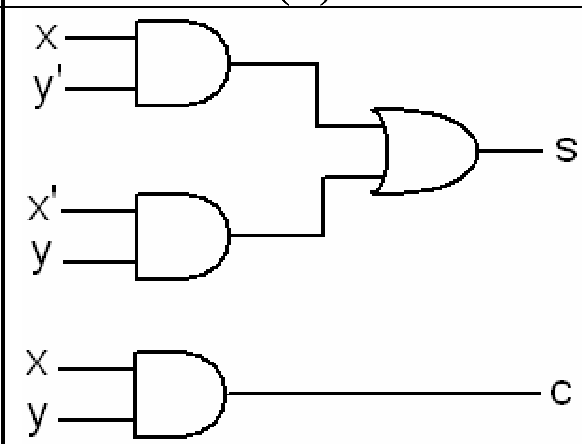
(B)

$S = x \oplus y$

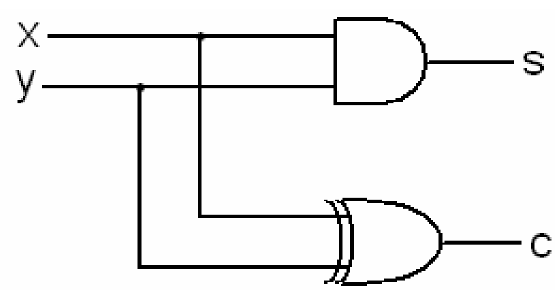
$C = xy$

Graphic Symbol

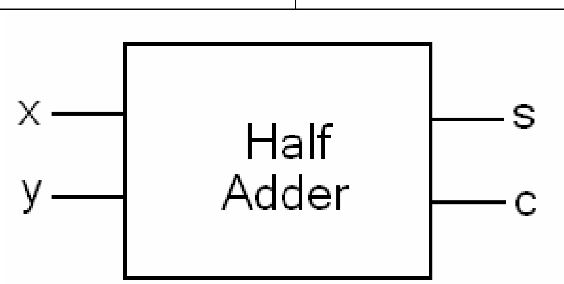
الرسم



شكل(A)



شكل(B)



هذا الشكل الذي نحتاجة لحل المسائل بواسطة (Half adder)

٤-٥ Full Adder :

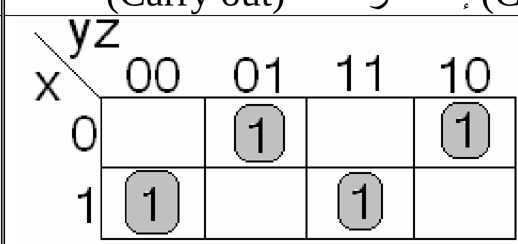
هي شكل مكبر (Half adder) وسوف نتعرف عليها أكثر من خلال الجدول التالي :

Truth Table

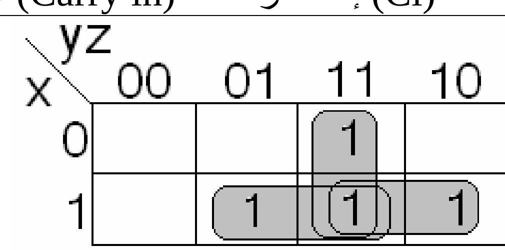
InPuts			OutPuts	
X	Y	Ci	Co	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

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

Algebraic Function

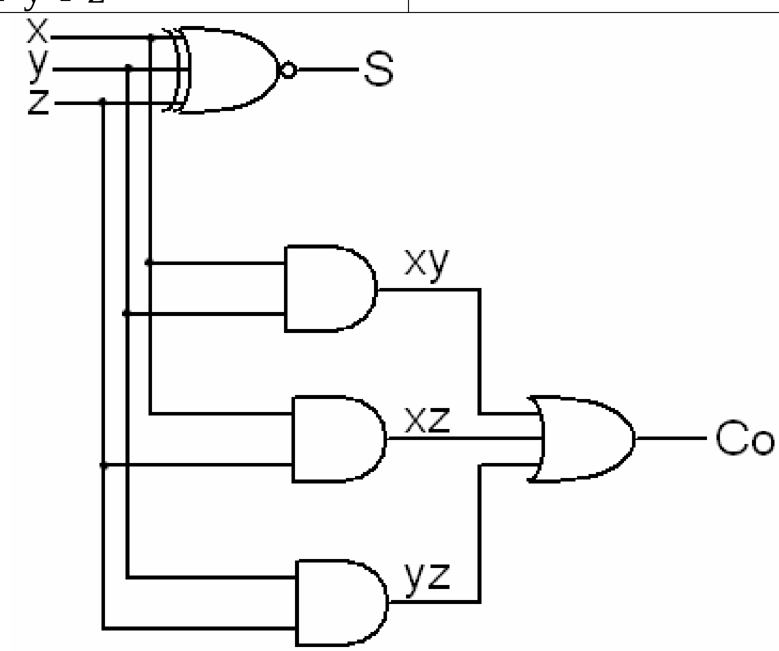


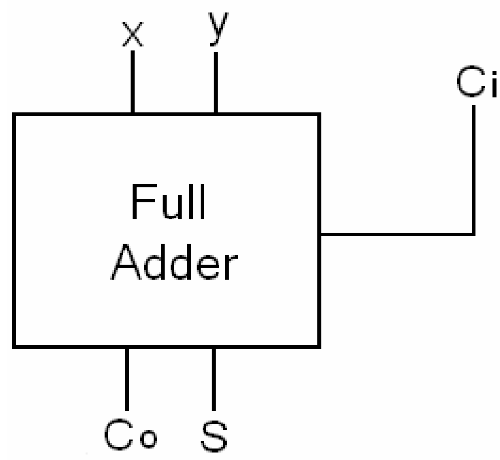
$S = x'y'z + x'yz' + xy'z' + xyz$
$= x \oplus y \oplus z$



$C = xy + xz + yz$

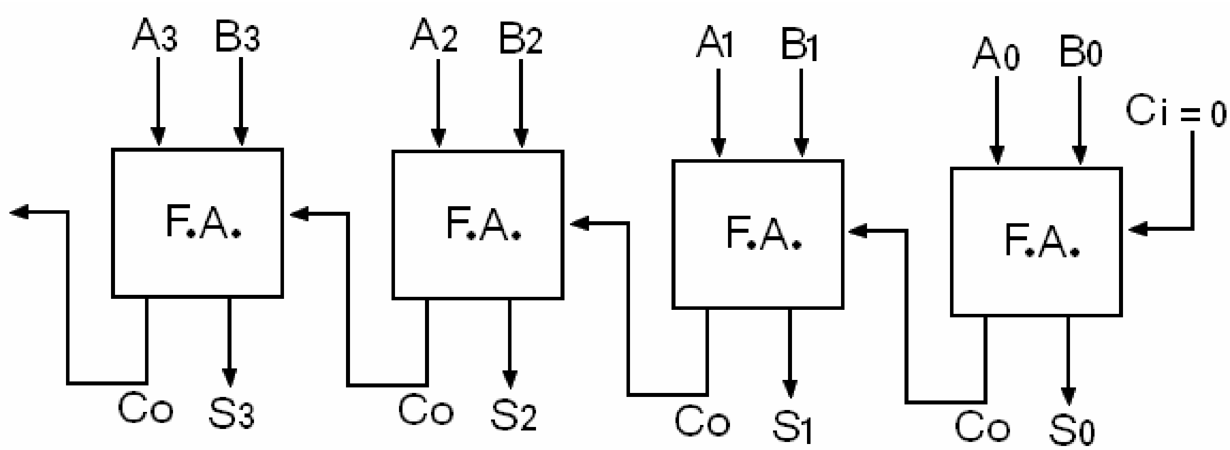
Graphic Symbol





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 0 = (Ci)

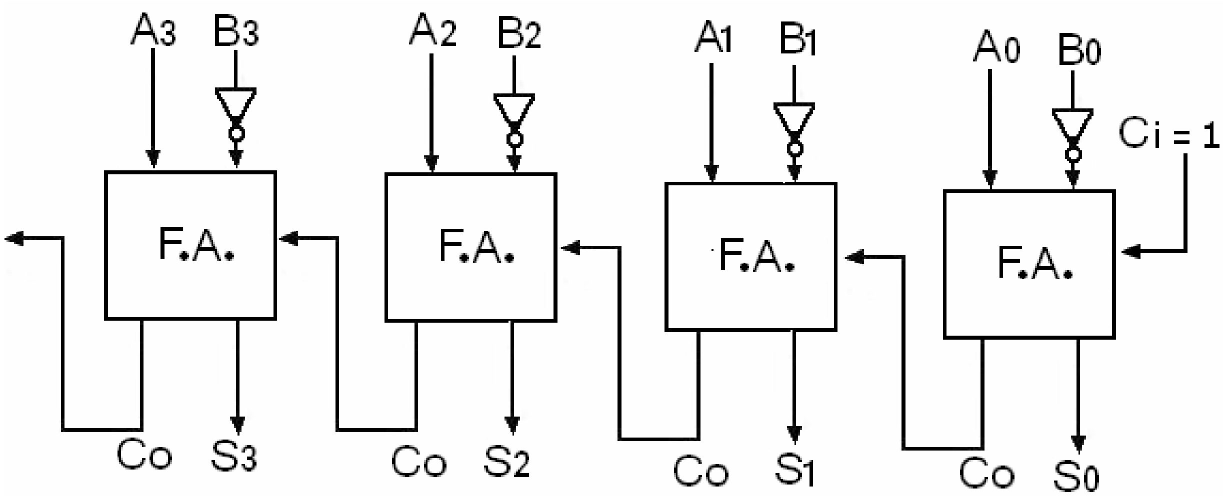
شكل مبسط يوضح ما الذي يحدث :

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

مثال Example :

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

الحل Selution :



الشرح explain :

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

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

$= A + (B' + 1)$

$= A + (2\text{'s Comp of } B)$

$= A - B$

مثال Example :

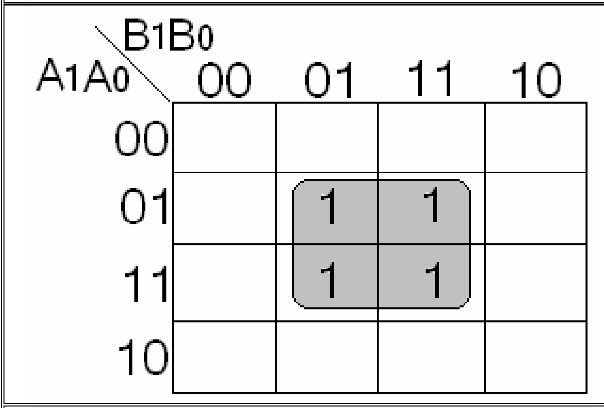
Design a 2 - bit binary multiplier :

الحل Selution :

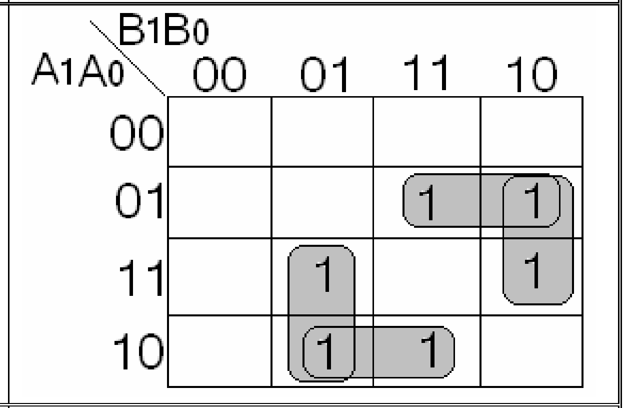
المطلوب تصيميم دالة (multiplier)عمل هذه الدالة تقوم بعملية الضرب لقيم المدخلات وتضع الناتج في المخرجات.
نرجع للمثال ونلاحظ أن المدخلات متغيرين (AوB)
(A1 و A0) تعتبر رقم واحد وكذلك (B1 و B0)
هذه الدالة تقوم بضرب قيمة A*B وتضع الناتج في خانة المخرجات ويكتب الناتج طبعاً بالنظام الثنائي مثل :
١- ثالث صف من المدخلات قيمة A = 0 0 (0) مضروبة في قيمة B = 1 0 (2) وكان الناتج عملية الضرب = 0 لذلك نكتب في خانة المخرجات الرقم 0
٢- ثامن صف من المدخلات قيمة A = 0 1 (1) مضروبة في قيمة B = 1 1 (3) 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 و الرسم .

InPuts		OutPuts			
A_1 A_0	B_1 B_0	C_3	C_2	C_1	C_0
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

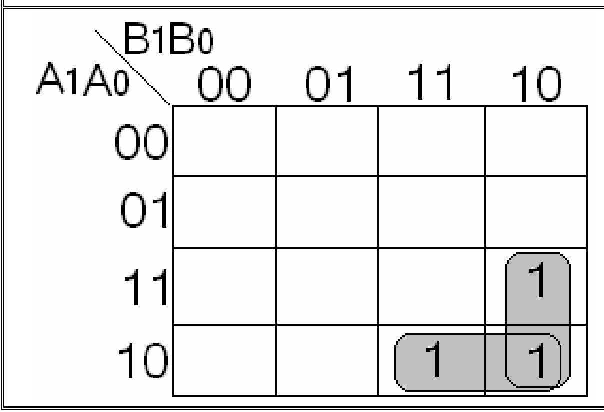
C0



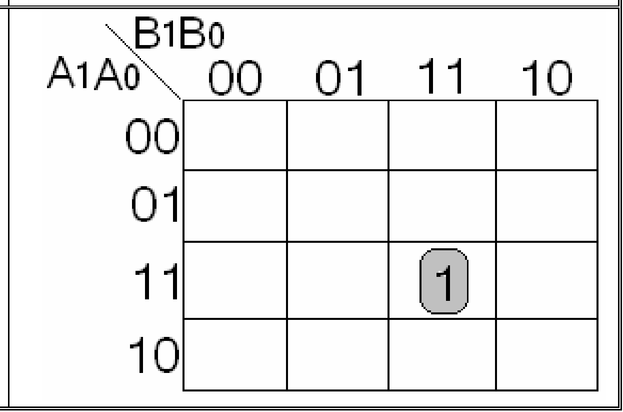
C1



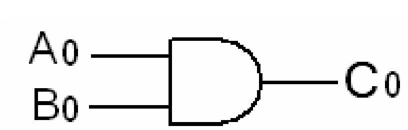
C2



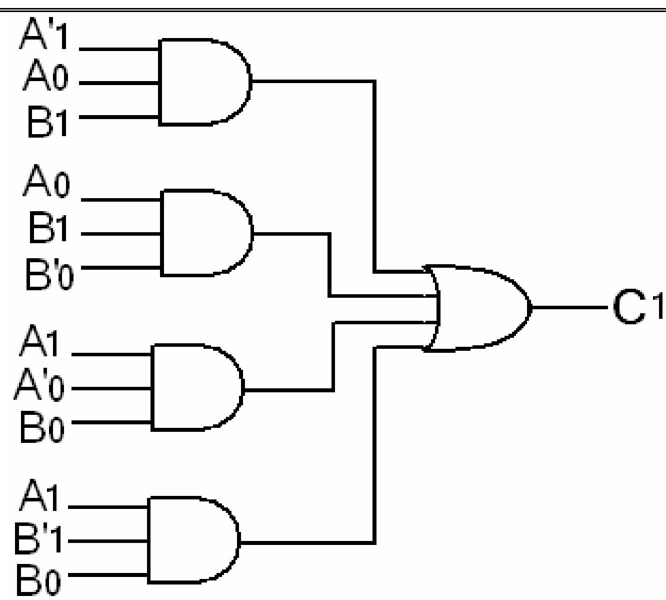
C3

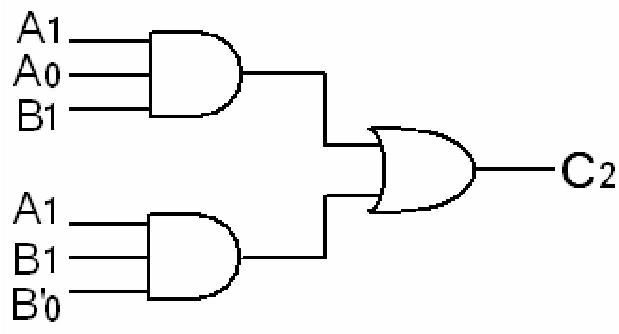
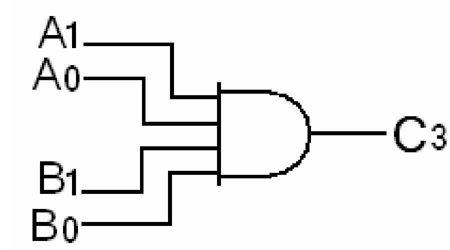


C0



C1



C2	C3
	

مثال Example :

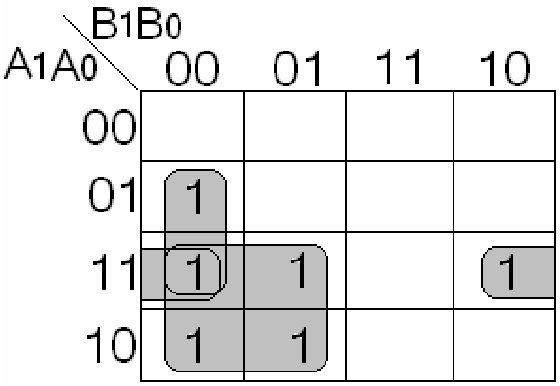
Design a 2 - bit magnitude comparator :

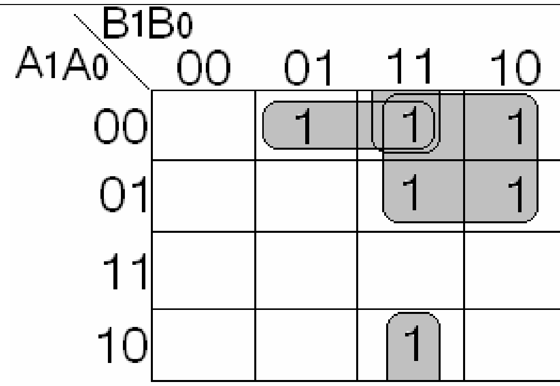
الحل Selution :

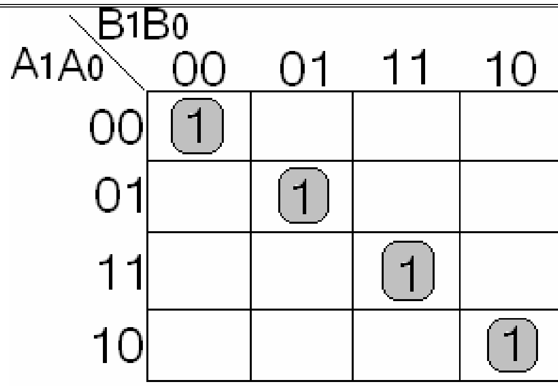
عملها تقوم بمقارنة بين قيم المدخلات.

المطلوب تصميم دالة (magnitude comparator)

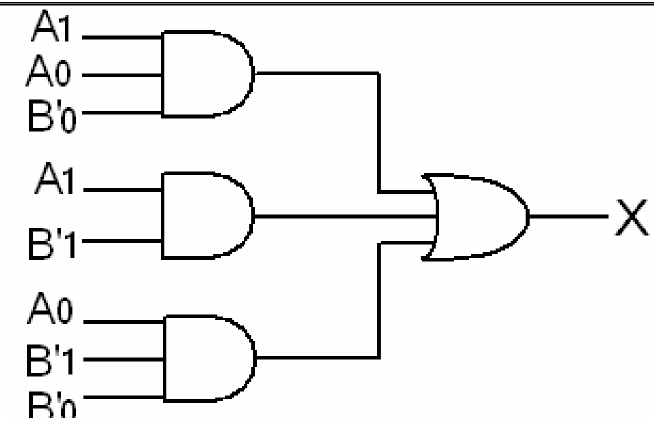
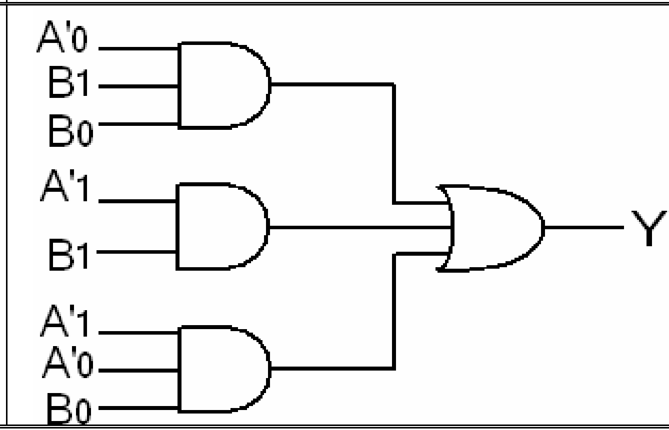
InPuts		OutPuts		
A_1 A_0	B_1 B_0	X (A>B)	Y (A<B)	Z (A=B)
0 0	0 0	0	0	1
0 0	0 1	0	1	0
0 0	1 0	0	1	0
0 0	1 1	0	1	0
0 1	0 0	1	0	0
0 1	0 1	0	0	1
0 1	1 0	0	1	0
0 1	1 1	0	1	0
1 0	0 0	1	0	0
1 0	0 1	1	0	0
1 0	1 0	0	0	1
1 0	1 1	0	1	0
1 1	0 0	1	0	0
1 1	0 1	1	0	0
1 1	1 0	1	0	0
1 1	1 1	0	0	1

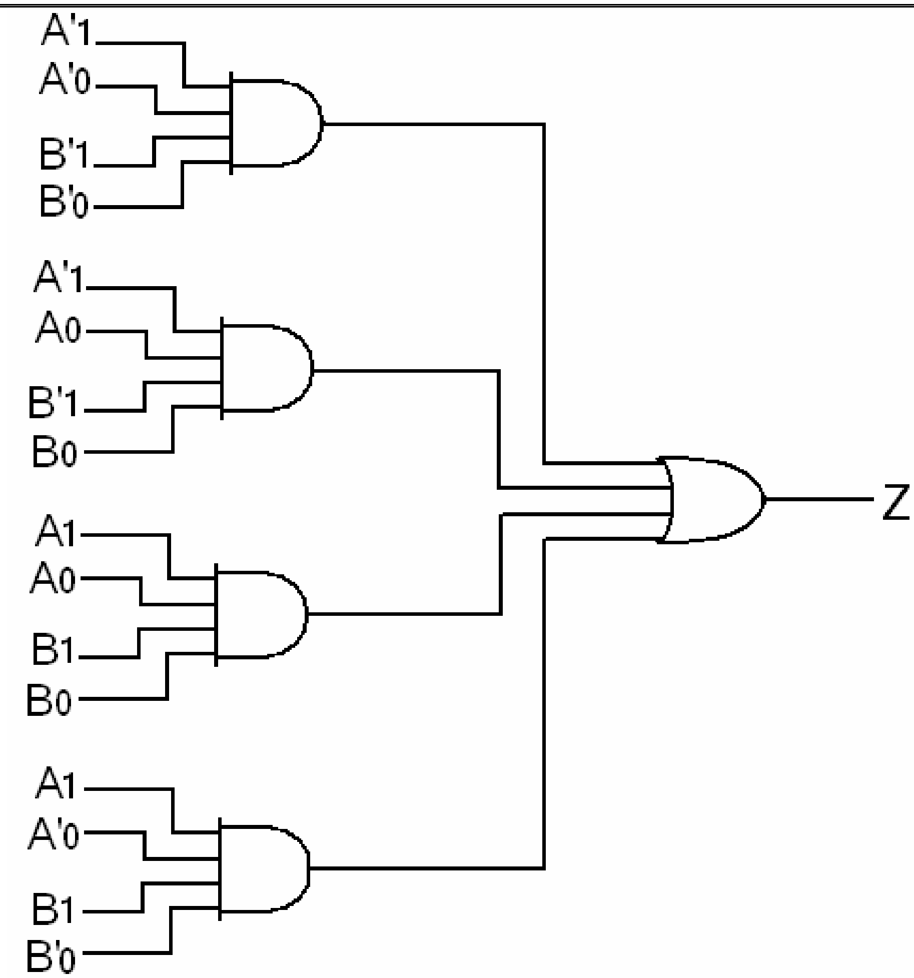
<table>
<tr><th>X</th><th>Y</th></tr>
<tr><td>


$X = A_1B'_1 + A_1A_0B'_0 + A_0B'_1B_0$
</td><td>


$Y = A'_1B_1 + A'_0B_1B_0 + A'_1A'_0B_0$
</td></tr>
<tr><th colspan="2">Z</th></tr>
<tr><td colspan="2">


$Z = A'_1A'_0B'_1B'_0 + A'_1A_0B'_1B_0 + A_1A_0B_1B_0 + A_1A'_0B_1B'_0$
</td></tr>
</table>

X	Y
	

Z


٦-٤ Decoder :

A decoder has n inputs and 2^n outputs

هذه الدالة مدخلاتها = n ومخرجاتها = 2^n أي 2 أس(قوى) عدد المدخلات

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

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

١- Decoder 2 * 4 (Has 2 inputs and 4 outputs) :

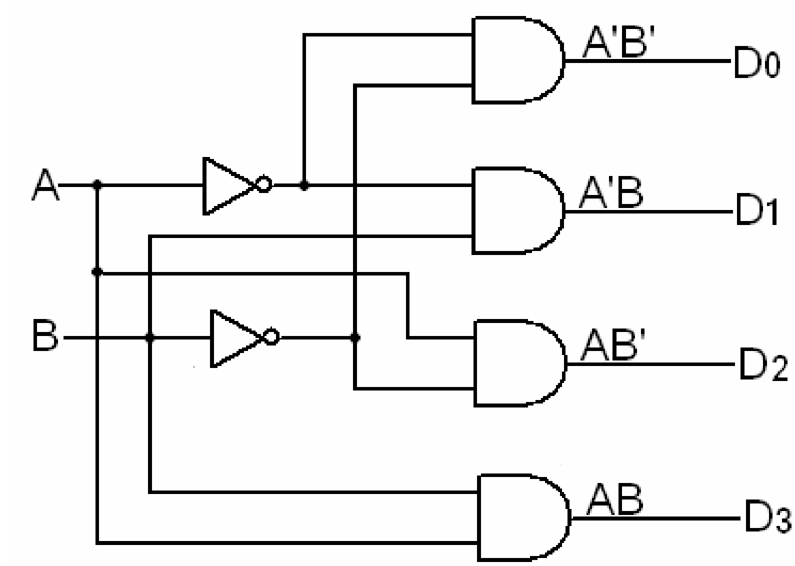
هو أول أنواع (Decoder) وسوف نتعرف عليه أكثر من خلال الجدول التالي :

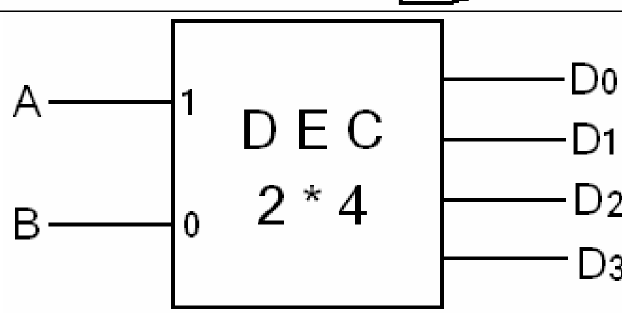
Truth Table

InPuts		OutPuts			
A	B	D_3	D_2	D_1	D_0
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 معا تشكل رقم وإذا كان الرقم الناتج = 0 نضع في الخانة D_0 الرقم 1 وباقي الخانات نضع فيها الرقم 0
ولو كان الرقم الناتج = 1 نضع في الخانة D_1 الرقم 1 وباقي الخانات نضع فيها الرقم 0 وهكذا .

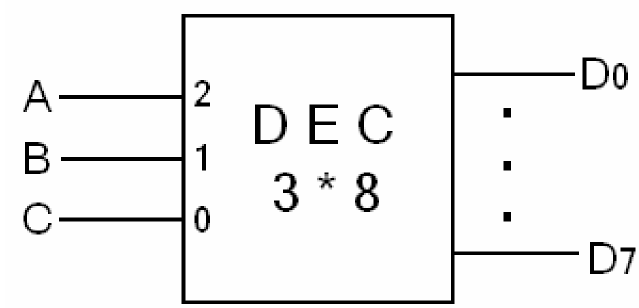
Graphic Symbol



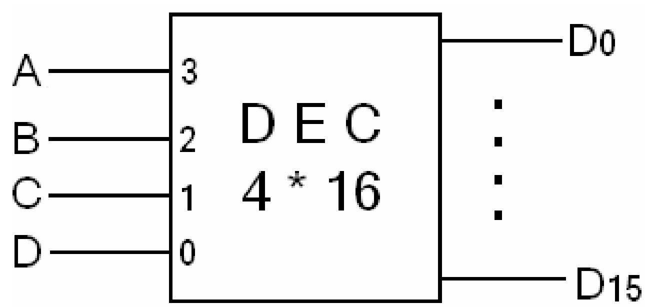


هذا الشكل الذي نحتاجة لحل المسائل بواسطة (Decoder)

٢- Decoder 3 * 8 :



٣- Decoder 4 * 16 :



٧-٤ Decoder With Enabel :

في هذه الجزئية سوف نتعرف على شيء جديد وهو(Enabel) وعمله بإختصار كالتالي :
إذا كانت قيمة (Enabel) = 0 فإن (Decoder) لن يعمل .
أما إذا كانت قيمة (Enabel) = 1 فإن (Decoder) يعمل .
إذن (Enabel) ليس إلا مفتاح للدالة .

InPuts			OutPuts			
E	A	B	D_3	D_2	D_1	D_0
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 0 = (Enabel) في الصفوف الأربعة الأولى
كان (Decoder) لا يعمل وبالتالي فإن قيم المخرجات = 0
وعندما كانت قيمة1 = (Enabel) في الصفوف الأربعة الأخيرة
كان (Decoder) يعمل وظهرت نتيجة عملة على المخرجات .

InPuts			OutPuts			
E	A	B	D_3	D_2	D_1	D_0
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 إختصار للجدول الذي قبلة .
وكما تلاحظون على الجدول الأول في الصفوف الأربعة الأولى .
أن وجودها ليس له أي فائدة ووجودها زائد فقط لذلك قمنا بإختصارها في صف واحد وهو الصف الأول من الجدول الجديد ولم نلغي وجودها تماماً.

الصف الأول من الجدول الجديد تلاحظون فيه أن قيم X = (B) و(A) وهذه (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 إختصار ينوب عن كتابة قيم (B) و(A) في المدخلات

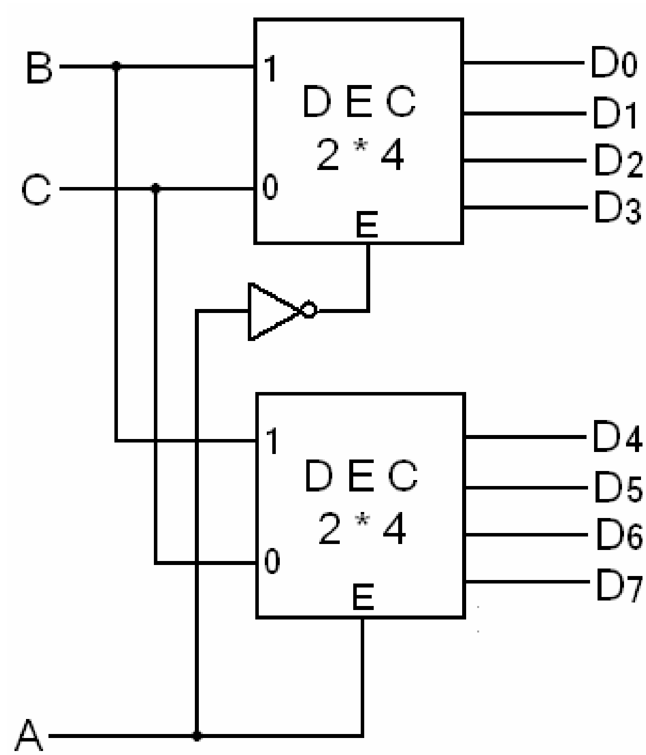
أي أن (X) مرة = 0 ومرة أخرى = 1 وكتبناها بهذه الطريقة للإختصار فقط
ويمكنك تجاهلها وكتابة قيم (B) و(A) كاملة إذا أردت .

مثال Example :

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

الحل Selution :

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



الشرح explain :

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

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

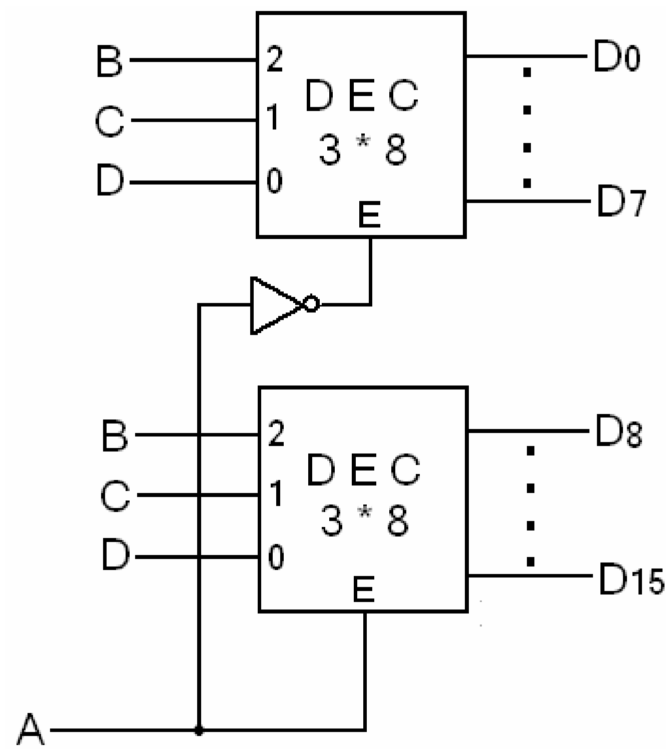
InPuts			OutPuts							
A	B	C	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

مثال Example :

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

الحل Selution :

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



٨-٤ Multiplexer :

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

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

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

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

١- Multiplexer 2*1 :

A Multiplexer has 2 inputs , 1 outputs and 1 selections

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

Truth Table

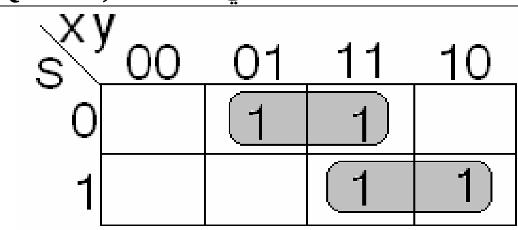
Selection	InPuts		OutPuts
S	X	Y	Q
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

قيمة المخرجات تكون بالشكل التالي :

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

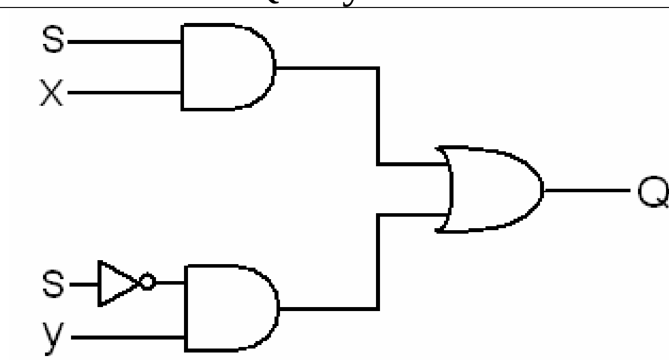
وإذا كانت1 = S فإن قيمة المخرجات هي قيمة X , Q = X

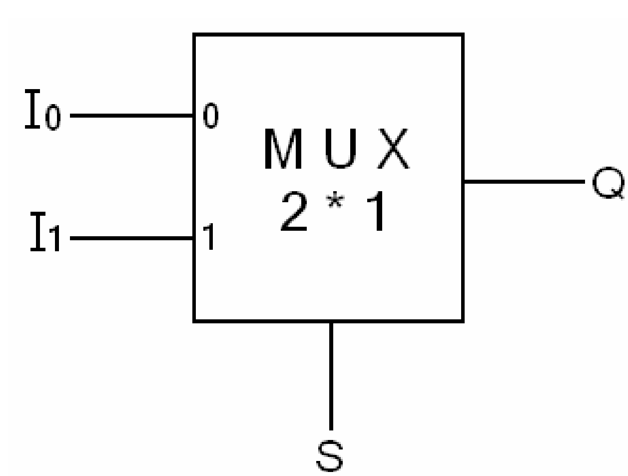
Algebraic Function



$Q = sy + sx$

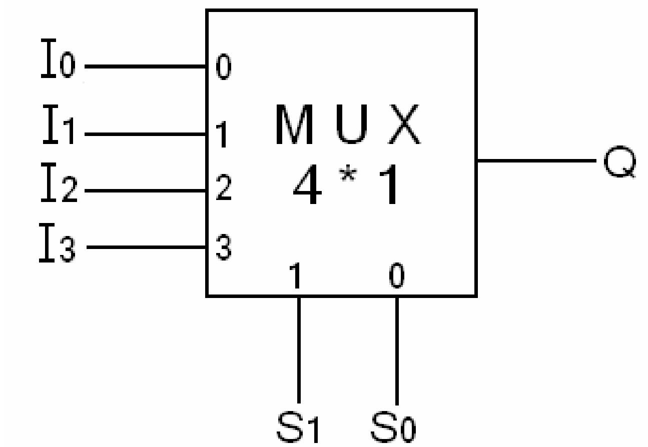
Graphic Symbol



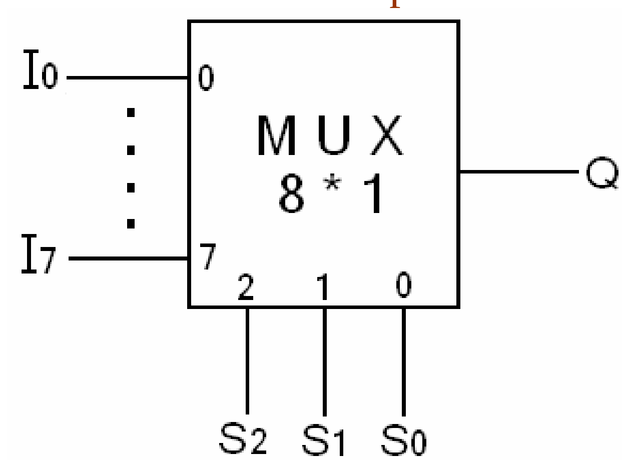


هذا الشكل الذي نحتاجة لحل المسائل بواسطة (Multiplexer)

٢- Multiplexer 4*1 :



٣- Multiplexer 8*1 :

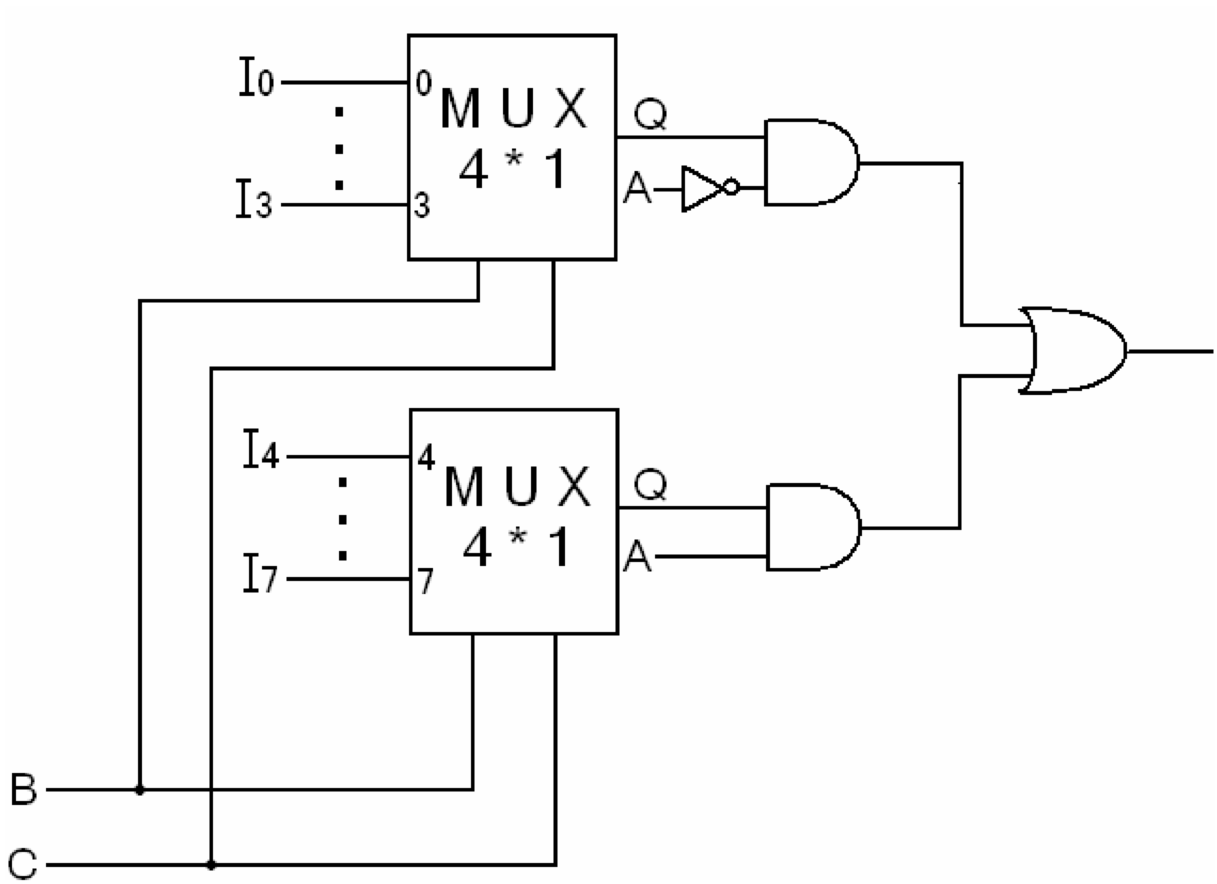


مثال : Example

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

الحل : Selution

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

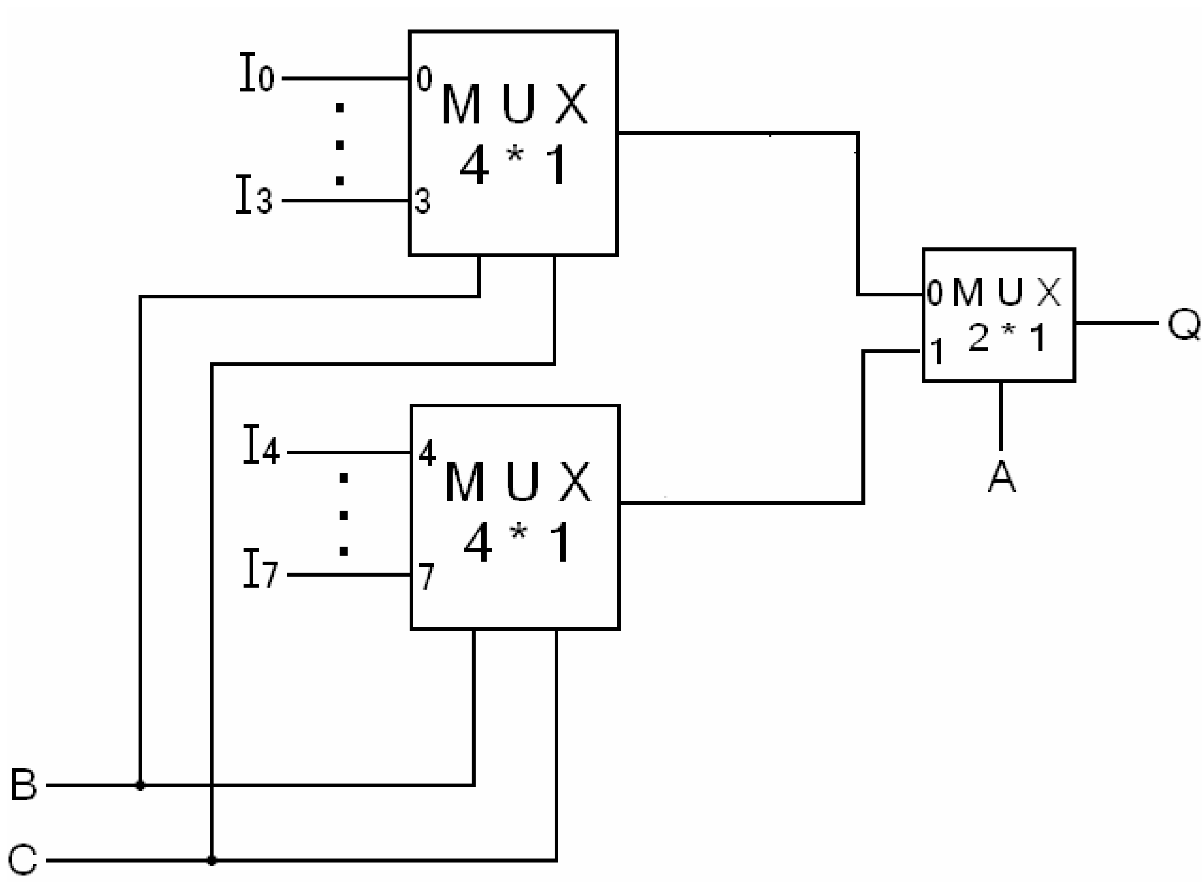


مثال Example :

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

الحل Selution :

المطلوب تركيب(بناء)(Multiplexer 8*1) بـ 2 (Multiplexer 4*1) و1 (Multiplexer 4*1)

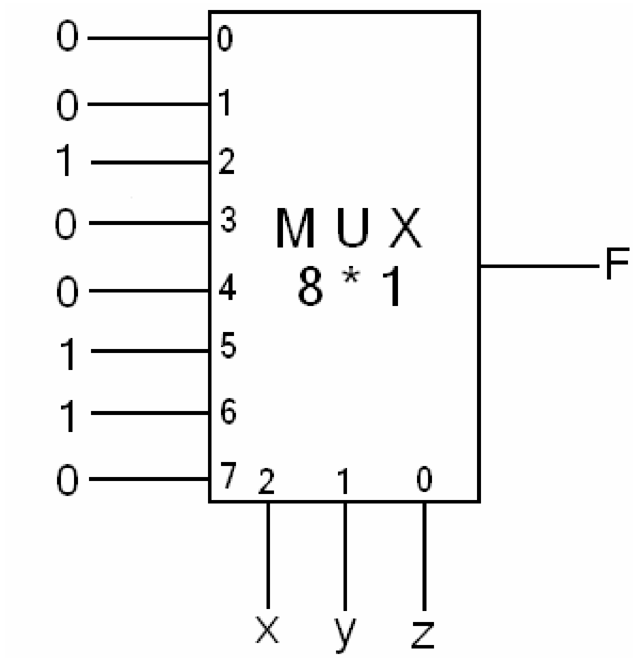


مثال Example :

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

الحل Selution :

المطلوب تنفيذ الدالة المنطقية التالية بإستخدام (8*1 Multiplexer)

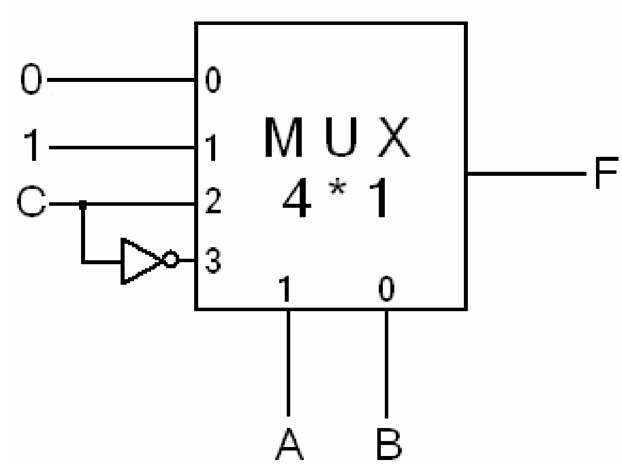


مثال Example :

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

الحل Selution :

المطلوب تنفيذ الدالة المنطقية التالية بإستخدام (4*1 Multiplexer)



الشرح explain :

يفترض أن نحل هذا المثال بإستخدام (8*1 Multiplexer)

ولأن كل قيمتين من المدخلات نستطيع أن نعوض 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 الخامس

Synchronous Sequentil Logic

٥-١ المقدمة Introduction :

في هذا الفصل سوف نتعرف على دالة جديدة وهي (Flip Flop) .

وسوف تحدث عن تعريفها وأشكالها وأنواعها أقسامها وعملها و حل المسائل بواسطتها والتحويل فيما بين أشكالها .

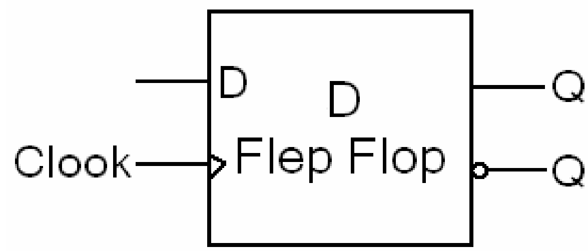
كما سوف نتحدث التحليل المؤقت للتسلسل الدائري (Analysis of clocked sequential circuits)

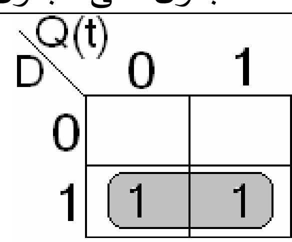
وإيجاد جدول (State table) و(State diagram) للدوائر(circuits) .

٢-٥ أنواع الـ (Flip Flop) Types of :

١- D Flip Flop :

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

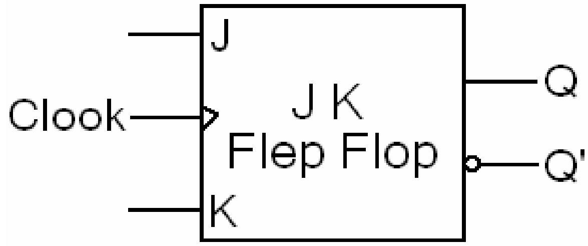
<table>
<tr>
<td></td>
<td>

هذا الشكل الذي نحتاجة لحل المسائل بواسطة (D Flip flop) .</td>
</tr>
<tr>
<td>Characteristic Table</td>
<td>
<table>
<tr><th>D</th><th>Q(t+1)</th></tr>
<tr><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>
الجدول السابق يوضح كيفيت تكوين قيمة Q(t+1) وكما تلاحظون قيمة Q(t+1) ليست إلا نسخة من قيمة (D).</td>
</tr>
<tr>
<td>Truth Table</td>
<td>
<table>
<tr><th colspan="2">Presnt State</th><th>Next State</th></tr>
<tr><th>D</th><th>Q(t)</th><th>Q(t+1)</th></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>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
يعتمد تكوين هذا الجدول على الجدول الذي قبلة</td>
</tr>
<tr>
<td>Characteristic Equation</td>
<td>

Q(t+1) = D</td>
</tr>
<tr>
<td>Excitation Table</td>
<td>
<table>
<tr><th>Q(t)</th><th>Q(t+1)</th><th>D</th></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>1</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
الجدول يوضح كيفيت تكوين قيمة(D) من خلال إعطائنا قيمة Q(t+1),Q(t)
و هذا الجدول ليس إلا صورة عكسية لـ (Truth Table) الدالة .</td>
</tr>
</table>

٢- J K Flip Flop :

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



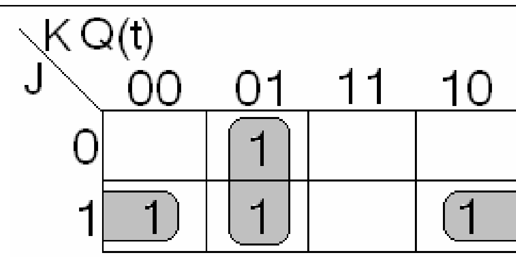
Characteristic Table

J	K	Q(t+1)
0	0	Q(t)
0	1	0
1	0	1
1	1	Q'(t)

Truth Table

Presnt 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



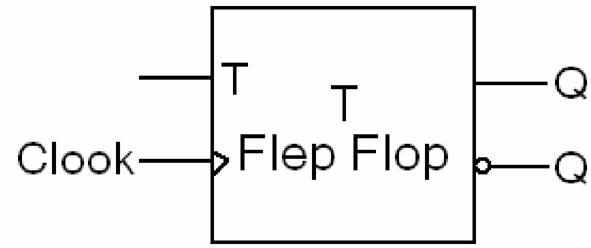
Q(t+1) = JQ'(t) + K'Q(t)

Excitation Table

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

٣- T Flip Flop :

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



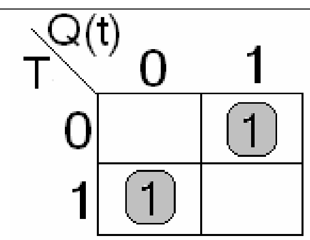
Characteristic Table

T	Q(t+1)
0	Q(t)
1	Q'(t)

Truth Table

Presnt State		Next State
T	Q(t)	Q(t+1)
0	0	0
0	1	1
1	0	1
1	1	0

Characteristic Equation



$$Q(t+1) = T'Q(t) + tq'(t) \quad \Rightarrow \quad T \oplus Q(t)$$

Excitation Table

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

مثال : Example

: Desing a J K Flip Flop using D Flip Flop

الحل : Selution

المطلوب تصميم (J K Flip Flop) بإستخدام (D Flip Flop)
ولكي تقوم بعملية التصميم يتوجب عليك أولا عمل (Truth table)
خلايا (Truth table) تقسم بالشكل التالي :
١- (Presnt state) ويحتوي على قيمة (Flip Flop) الذي نريد تصميمه وفي هذا المثال (Flip Flop) الذي نريد تصميمه (J K)،ويحتوي أيضا على قيمة Q(t)
٢- (Next state) ويحتوي على قيمة (Flip Flop) الذي نريد إستخدامه في عملية التصميم وفي هذا المثال هو(D) ويحتوي أيضا على قيمة Q(t+1)

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

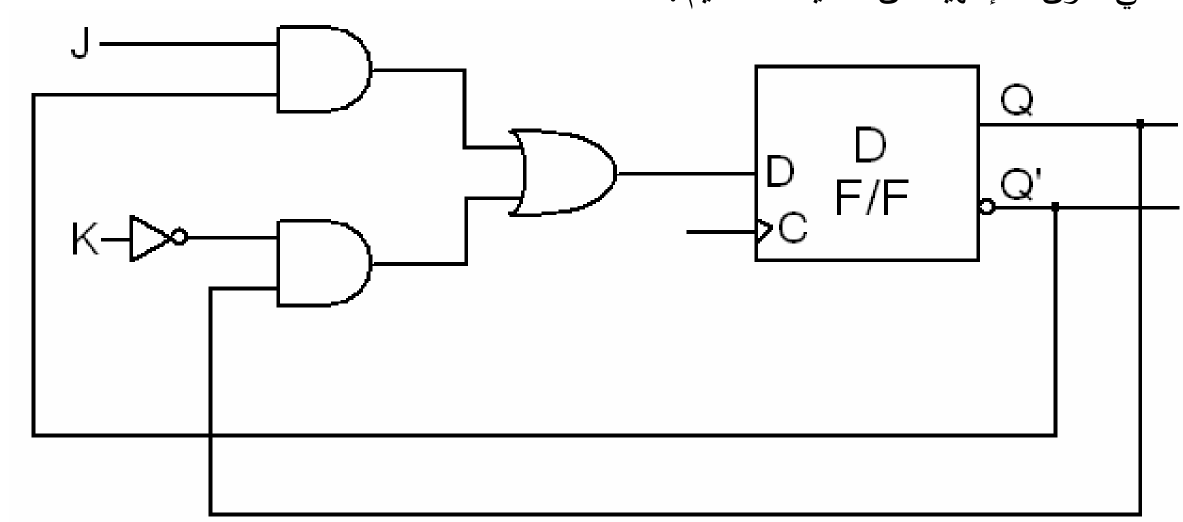
J	K	Q(t)	Q(t+1)	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

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

J \ KQ(t)	00	01	11	10
0		1		
1	1	1		1

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

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



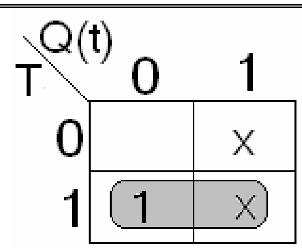
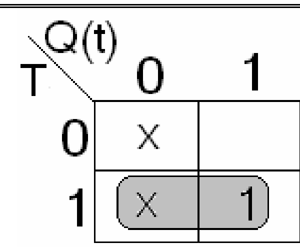
مثال Example :

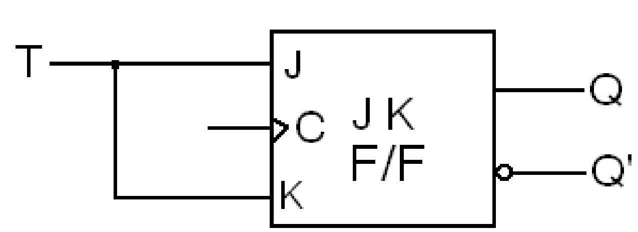
Desing a T Flip flop using J K Flip flop :

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

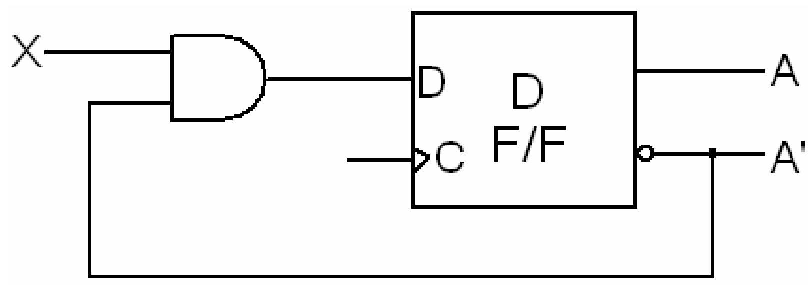
J	K
 K = T	 J = T



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

مثال Example :

Analysis of clocked sequential circuits :



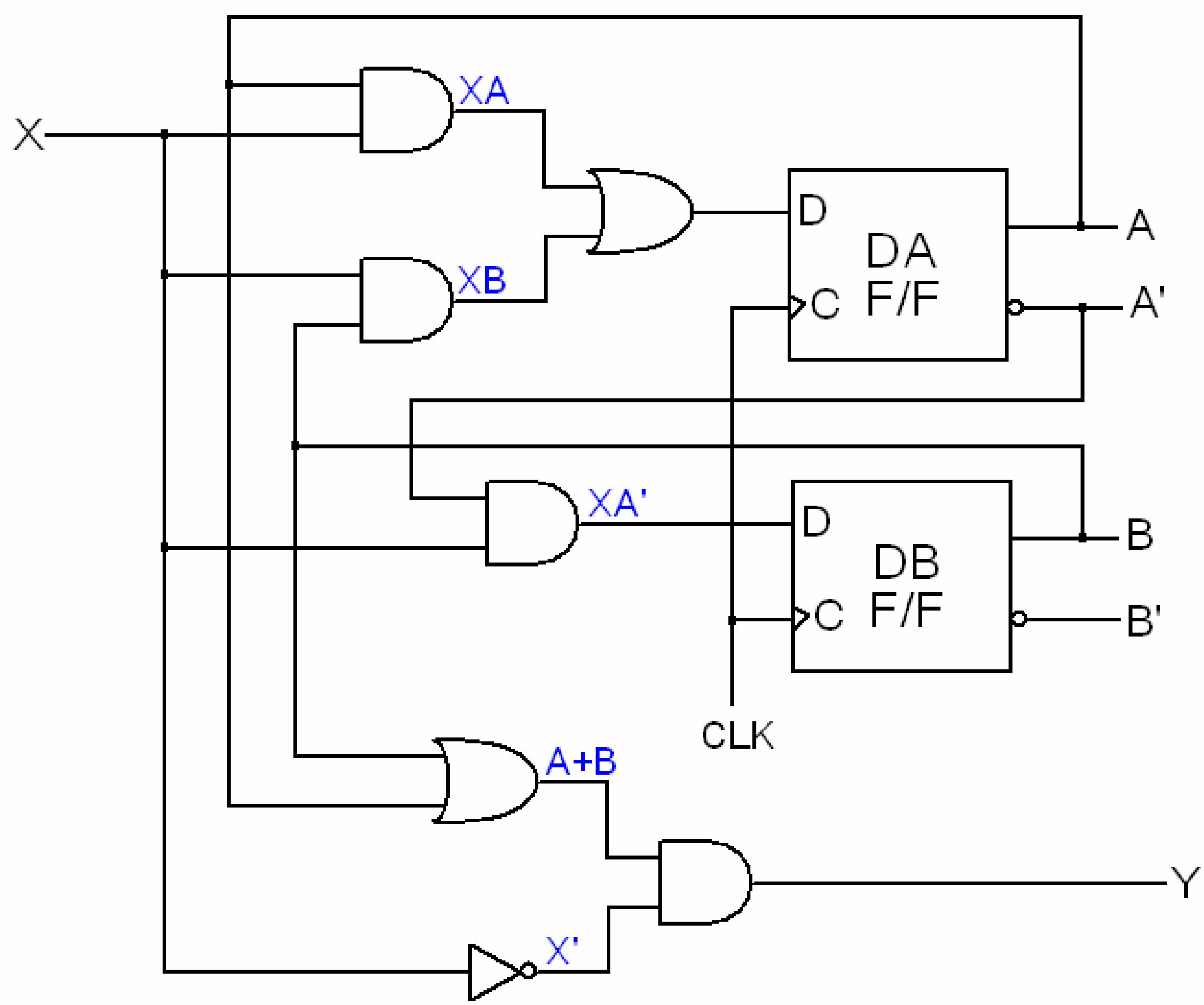
الحل Selution :

المطلوب تحليل الشكل السابق للوصول مخرجات الدالة .

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

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

مثال Example :
: Analysis of clocked sequential circuits



الحل Selution :

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

نرجع للرسم ونلاحظ أنه يحتوي 2 (Flip Flop)

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

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

أما Y فتمثل قيمة (O/P)

نرجع للشكل ونقوم بتحليه وتتبع مدخلاته ومن ثم نصل للقيم المطلوب إيجادها و هي كالتالي :

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

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

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

٤-٥ 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 (N.S), (O/P) 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 بتعبأة حقولها كما تعلمنا سابقاً .
أما بالنسبة لـ (N.S) و (O/P) سوف تعتمد على المعادلات السابقه لإجاد قيمها
من معادلة $A(t+1)$ نوجد قيم العمود (A)
ومن معادلة $B(t+1)$ نوجد قيم العمود (B)
أما من معادلة Y نوجد قيم العمود (Y)

على سبيل المثال لو أخذنا الصف الرابع من الجدول وأردنا أن نبين كيف قمنا بتعبئة حقوله قيم (N.S) و (O/P) في هذا الصف كالتالي : $(A = 0)$ و$(B = 1)$ و$(X = 1)$

نعوض في المعادلة $A(t+1) = XA + XB$ لكي نوجد قيمة (A)

$$A = XA + XB$$
$$=1*0 + 1*1$$
$$= 0 + 1$$
$$= 1$$

ونعوض في المعادلة $B(t+1) = XA'$ لكي نوجد قيمة (B)

$$B = XA'$$
$$= 1*1$$
$$= 1$$

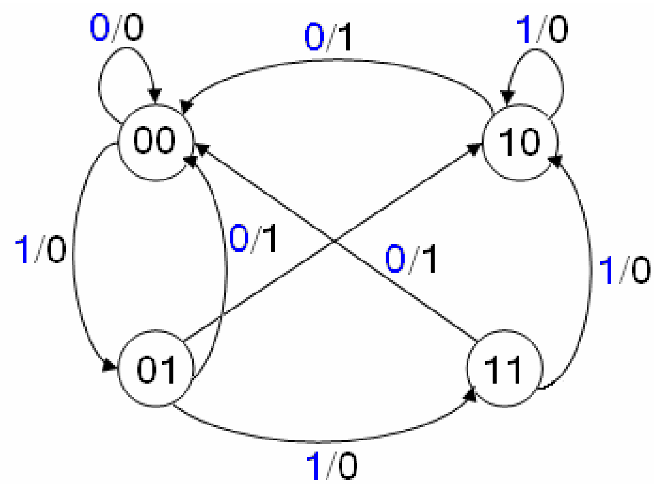
ونعوض في المعادلة $Y = X'A + X'B$ لكي نوجد قيمة (Y)

$$Y = X'A + X'B$$
$$= 0*1 + 0*1$$
$$= 0 + 0$$

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

٥-٥ State Diagram :

سوف نقوم بإنشاء (State diagram) للمثال السابق



الشرح explain :

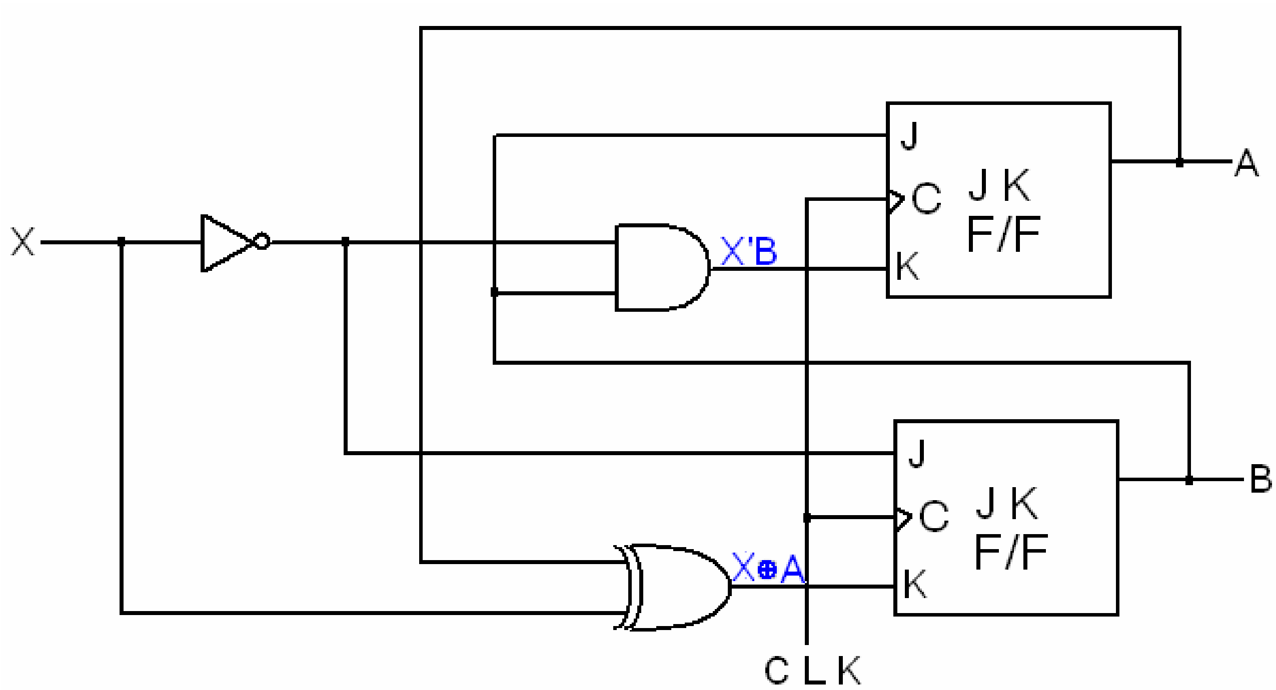
(State diagram) ليس إلا وصف لجدول (State table) بالرسم .
وعند وجود أحدمها نستطيع إجاد الآخر منه .

وإليكم الآن شرح يصف لنا الرسمة السابقة :
القيم التي داخل الدوائر تمثل قيم (A وB) التي تمثل قيم [(P.S),(N.S)]
أما القيم التي على الأسهم تمثل قيم (X وY) 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 بالتالي :
نتجه للدائرة التي تحتوي على قيمة (P.S) وفي مثالنا = 00 ، ثم ننظر في قيمة (N.S) وفي مثالنا = 00 ، ثم نرسم سهم من قيمة (P.S) يتجه 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 (I/P) = 0 والقيمة التي باللون الأسود تمثل (O/P) = 0

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

مثال Example :
Analysis with J K Flip flop :



الحل Selution :

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

$J_A = B$ $\quad\quad$ $K_A = X'B$

$J_B = X'$ $\quad\quad$ $K_B = A'X + X'A \Rightarrow X \oplus A$

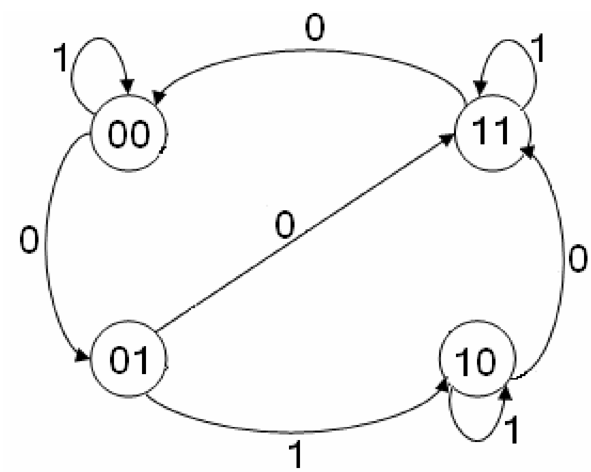
$$A(t+1) = J_A*A' + K'_A*A$$
$$= BA' + (X'B)'A$$
$$= A'B + (X+B')A$$
$$= A'B + XA + AB'$$

$$B(t+1) = J_B*B' + K'_B*B$$
$$= X'B' + (XA' + X'A)'B$$
$$= X'B' + X'A'B + XAB$$

: 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

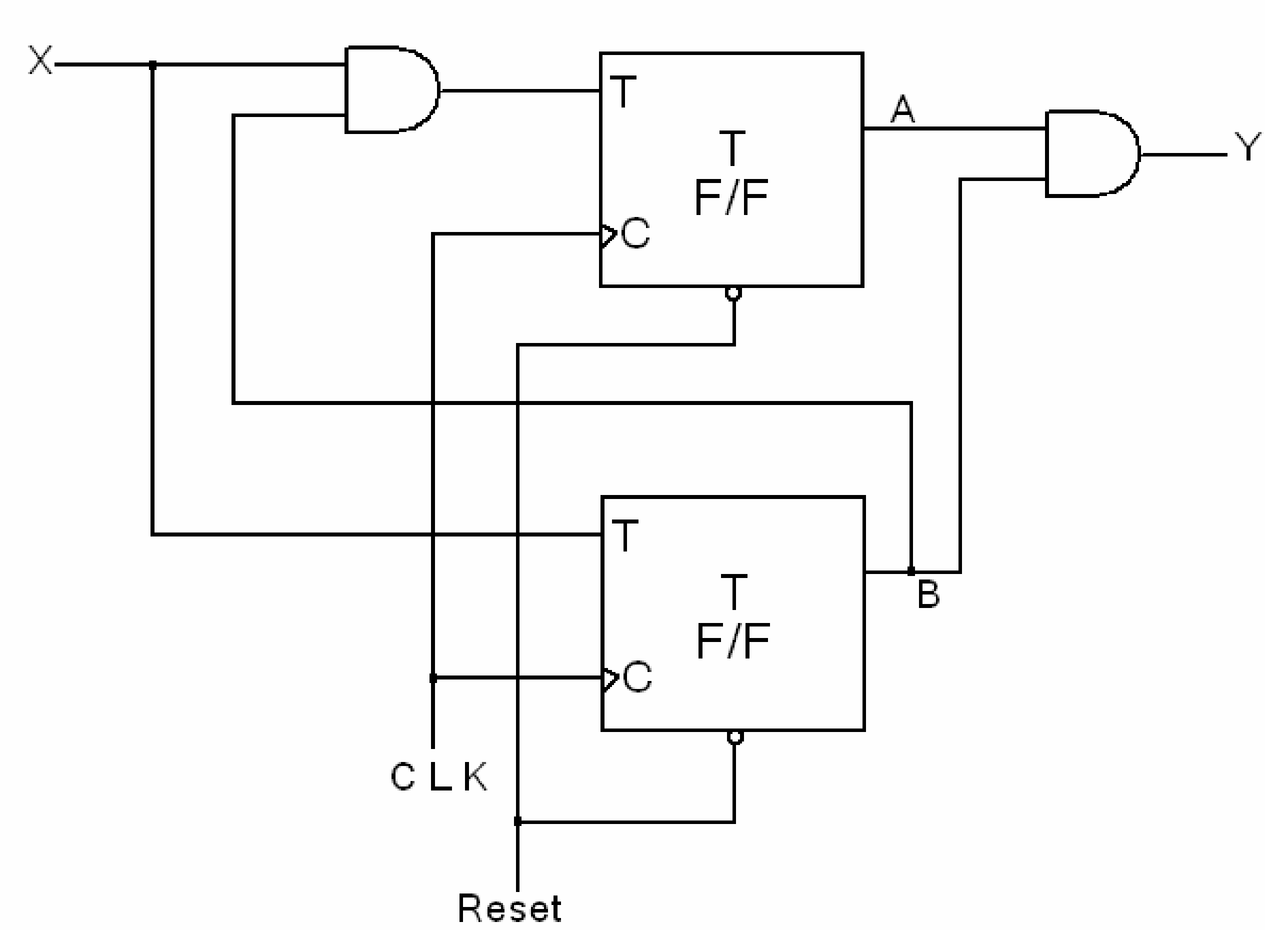


الشرح explain :

لاحظو أنه يجود قيمة واحدة على الأسهم وهي قيمة (X) وذلك لأنه لا يوجد قيمة لـ (Y)في هذا المثال .

: Example مثال

: Analysis with T Flip flop



: Selution الحل

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

$T_A = XB$ $\qquad$ $T_B = X$

$A(t+1) = T_A \oplus A'$
$= XB \oplus A$

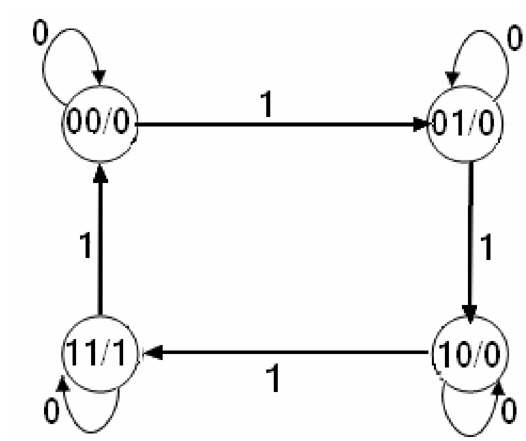
$B(t+1) = T_B \oplus B$
$= X \oplus B$

$Y = AB$

: 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 :

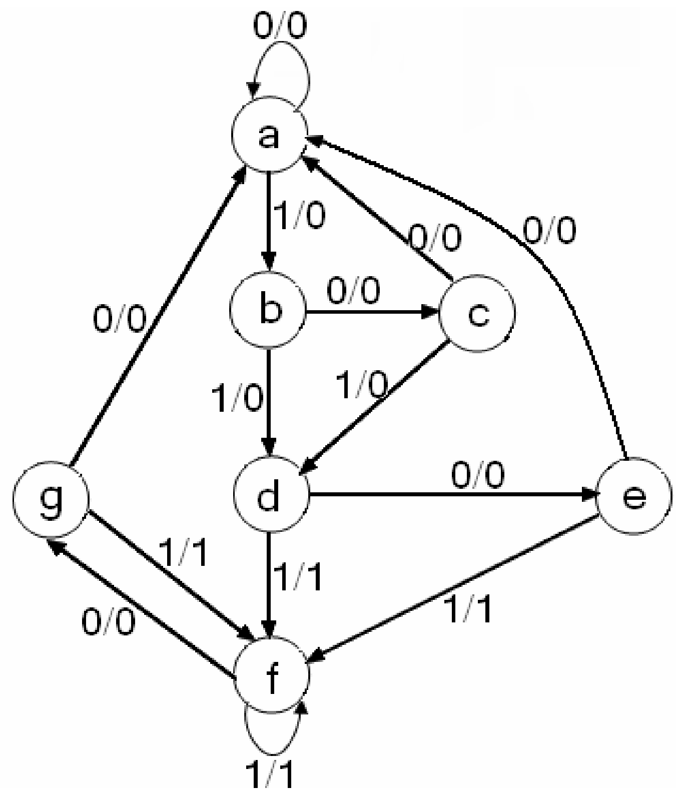
لاحظو على رسمة (State diagram) أن قيمة (Y) لم تكتب على السهم كما تعلمنا سابقاً وإنما كتبت داخل الدوائر مع قيم (N.S) أي مع قيم (A,B) والسبب أن قيمة (Y) تعتمد في تكوينها على قيمة (N.S) دون قيمة (X) مباشرة أما في الأمثلة السابقة فإن (X) كان يدخل في تكوين قيمة (Y) لذلك كنا نضع قيمة (Y)على الأسهم مع قيمة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

٦-٥ State Reduction and Assignment :

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

مثال Example :



الحل Selution :

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 لاعليك إلا فقط تتبع الأسهم من أين إنطلقت وإلى أين إتجهت .

: State Table

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 table) لرسمة (State diagram) 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 لو أخذنا الصف الأول من الجدول لنبين كيف قمنا بتعبئه :
نلاحظ على الجدول أن قيمة (P.S = a) بعد ذلك نرجع لرسمة (State diagram) ونتجه للدائرة التي تحتوي على القيمة (a) ونتتبع 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 نكون 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 نكون قد إنتهينا من تعبئة حقول الصف الأول من الجدول .
ونعمل مع باقي الصفوف مثل ماعملنا مع الصف الأول .
وبالتالي نكون قد إنتهينا من الخطوة الأولى وهي تعبئة الجدول .

ثم ننتقل للخطوة الثانية وهي تبسيط الجدول
ننظر في قيم الجدول ونبحث إذا كان هنالك يوجد صفوف متشابهه لكي نختصرها معاً.
ونلاحظ أن الصفين (e) و (g) متشابهان أي لهما نفس القيم ولذلك سوف نقوم بإلغاء أحدهما وليكن 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
d	e	f	0	1
e	a	f	0	1
f	e	f	0	1

بعد إلغائنا للصف (g) وإستبدال كل (g) بالقيمة (e) أصبح لدينا صفين آخرين متشابهين وهما (d) و (f)
نقوم بإلغاء 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	d	0	1
e	a	d	0	1

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

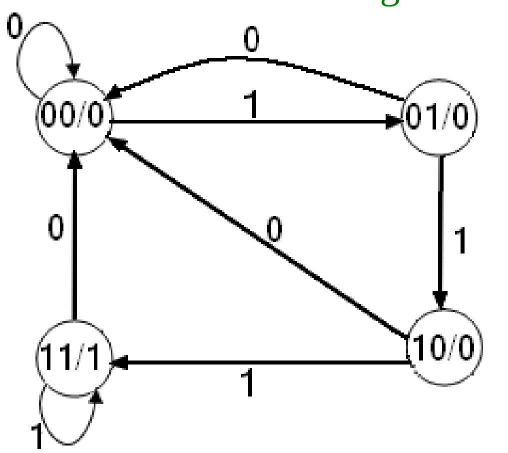
٧-٥ إجراء التصميم Design Procedure :

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 (State diagram) ومنها سوف تقوم بإجاد جدول (State table) بعد ذلك سوف تقوم بتبسيط مخرجات الجدول وإجاد قيم (N.S) و(O/P) وأخيراً نقوم برسم الدوال الناتجة .

مثال Example :

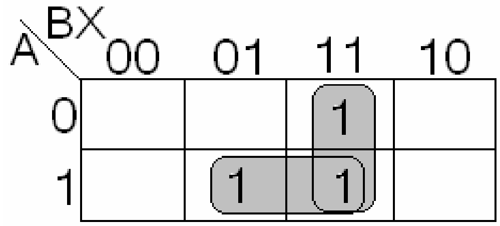
Design circuit thet delects three or more consecutive 1'S in a string of bits :

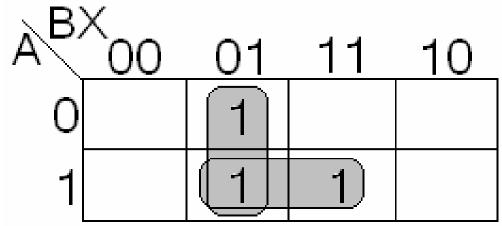


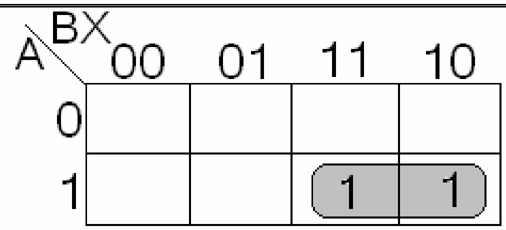
الحل Selution :

حل المثال بـ D Flip flop :

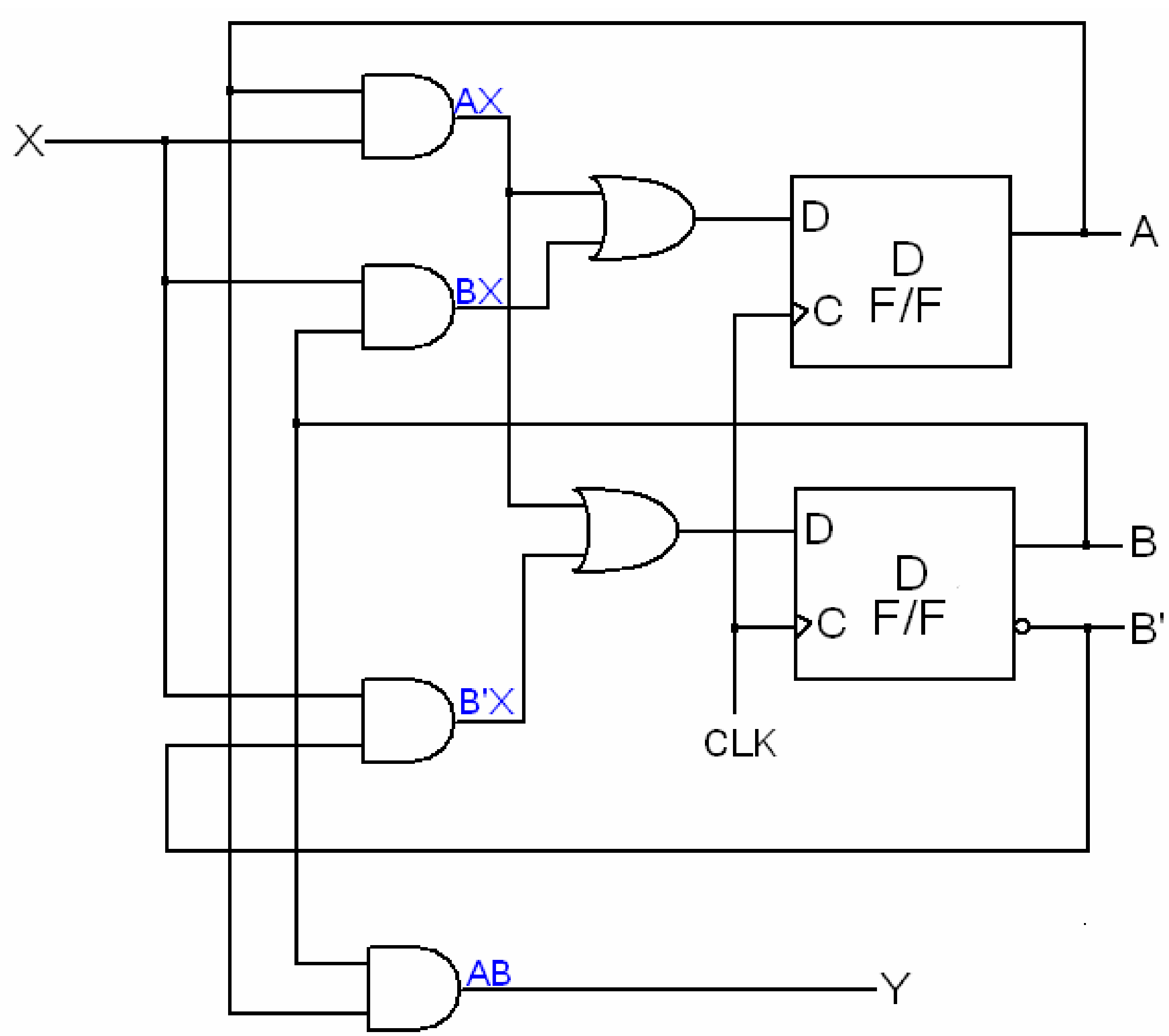
P.S		I/P	N.S		O/P		
A	B	X	A	B	Y	DA	DB
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	1
0	1	0	0	0	0	0	0
0	1	1	1	0	0	1	0
1	0	0	0	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1

<table>
<tr><th>D_A</th><th>D_B</th></tr>
<tr><td>

$D_A = AX + BX$</td><td>

$D_B = AX + B'X$</td></tr>
<tr><th colspan="2">Y</th></tr>
<tr><td colspan="2">

$Y = AB$</td></tr>
</table>



حل المثال بـــ J K Flip flop :

P.S		I/P	N.S		O/P				
A	B	X	A	B	Y	J_A	K_B	J_A	K_B
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

J_A

A \ BX	00	01	11	10
0			1	
1	x	x	x	x

$J_A = BX$

K_A

A \ BX	00	01	11	10
0	x	x	x	x
1	1			1

$K_A = X'$

J_B

A \ BX	00	01	11	10
0		1	x	x
1		1	x	x

$J_B = X$

K_B

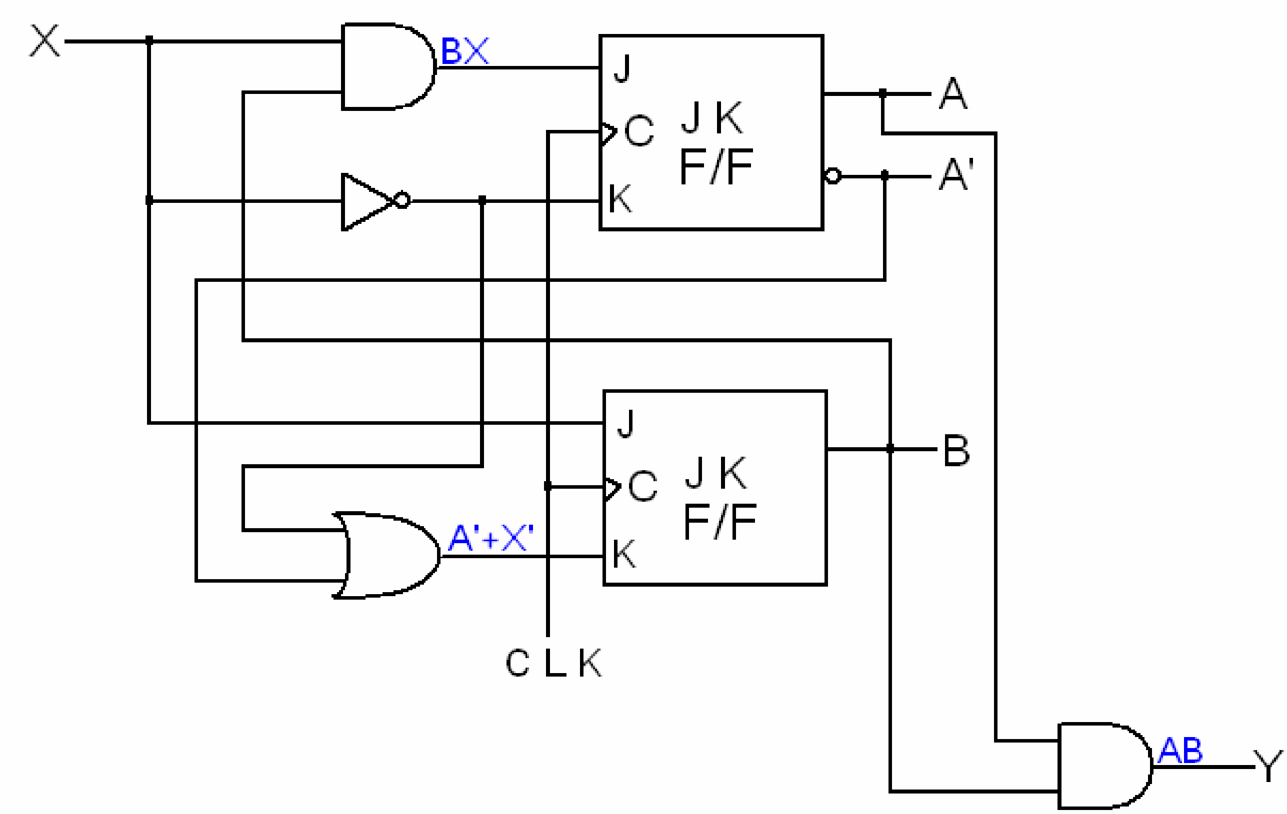
A \ BX	00	01	11	10
0	x	x	1	1
1	x	x		1

$K_B = A'+X'$

Y

A \ BX	00	01	11	10
0				
1			1	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

A \ BX	00	01	11	10
0			1	
1	1			1

TA = A'BX + AX'

TB

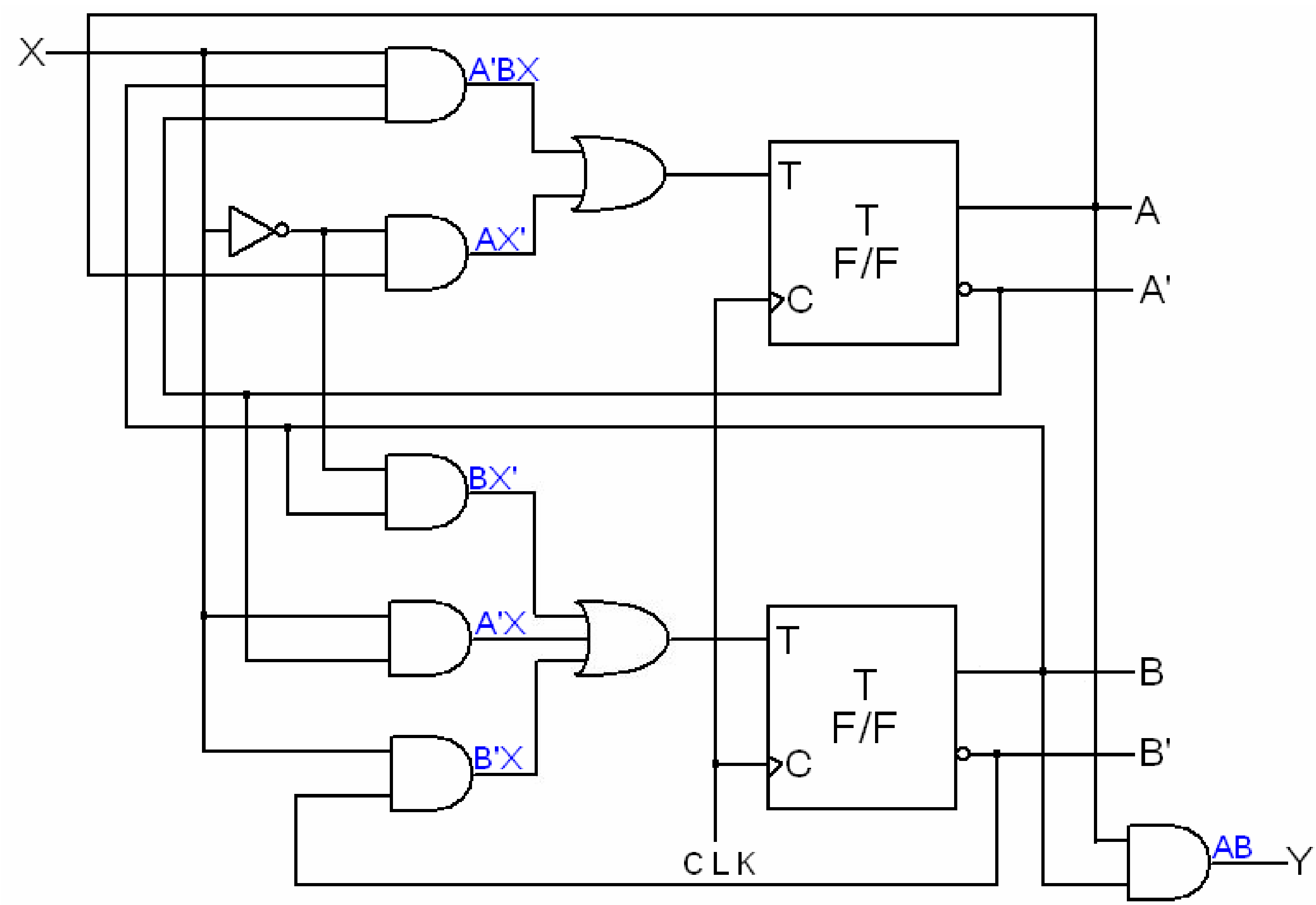
A \ BX	00	01	11	10
0		1	1	1
1		1		1

TA = B'X + A'X + Bx'

Y

A \ BX	00	01	11	10
0				
1			1	1

Y= AB



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)
كما سوف نتعرف على العداد (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 جهه كانت اليمين أو اليسار سوف تطرد آخر قيمة في (Register) خارج (Register) وتحل محلها القيمة التي قبلها وتحل محل القيمة القبل 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

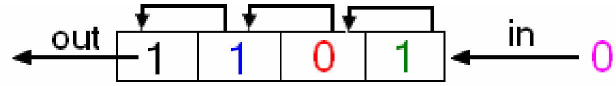
1	1	0	1

الحل Selution :

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

تتبع قيم (Register) قبل وبعد عملية (Shift) وتتبع القيم وقد قمت بتلوين القيم لتلاحظ إنتقال 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

مثال Example :

R Shift Right :

R

0	1	0	1

الحل Selution :

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

R

0	0	1	0

٢- Rotate Register :

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)

مثال Example :

R Rotate left :

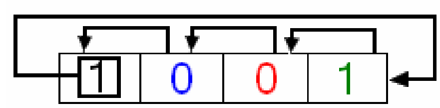
R

1	0	0	1

الحل Selution :

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

مثال Example :

Rotate Right R :

R

0	1	0	1

الحل Selution :

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

R

1	0	1	0

مثال Example :

Rotate Right R 3 himes :

R

1	1	0	0	0	1	0	1

الحل Selution :

المطلوب عمل (Rotate) إلى جهة اليمين 3 مرات لقيم (Register   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 Selution :

المطلوب عمل (Shift) إلى جهة اليسار 4 مرات لقيم (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 .

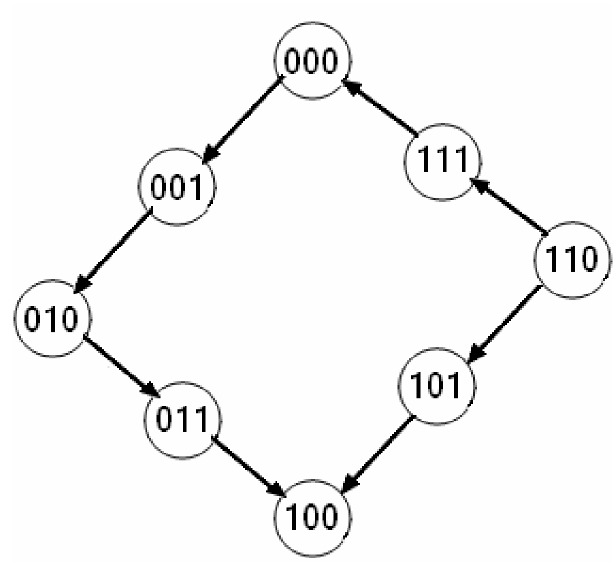
وفي المرة الثانية نأخذ الرقم قبل الأخير من (Serial) وهو الرقم 0 ونعمل مع مثل ماعملنا في المرة الأولى وهكذا إلى أن ننتهي من عمل (Shift) 4 مرات وبالتالي نكون قد إنتهينا من حل هذا المثال .

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

٣-٦ العداد Counter :

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 الموضح برسمة (State diagram) السابقة
كل صف مكون 3 أعداد ، وسوف يكون لدينا 8 صفوف .
وذلك حسب العملية الحسابية التالية :
$2^3 = 8$ حيث أن :
الأس وهو العدد 3 يمثل العدد الذي يستقبله العداد وهو (3-bit) الذي يمثل العدد المكون منه كل صف.
وناتج العملية الحسابية السابقة وهو العدد 8 يمثل عدد الصفوف .

لو أن العداد يستقبل (4-bit)
سوف تنتج لدينا العملية الحسابية التالية :
$2^4 = 16$ ومنها نوجد
أن كل صف مكون من 4 قيم ، وأن عدد الصفوف = 16 صف
نرجع لمثالنا وأريد أن أبين أن الطريقة والخطوات المتبعة في تصميم (Counter) هي نفس الطريقة ونفس الخطوات المتبعة لحل مسائل (Design Procedure) والتي سبقت أن مرت بنا في الباب الخامس .

حيث أن خطوات الحل سوف تكون كالتالي :
أعطانا في السؤال رسمة (State diagram) لـ (Counter) ومنها سوف تقوم بإجاد جدول (State table) ونقوم بتعبئة خلايا الجدول كما تعلمنا حيث :
نضع في الأعمدة (P.S) القيم التي سوف تنطلق منها السهم .
ونضع في الأعمدة (N.S) القيم التي سوف تصل إليها الأسهم .

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 لستهل عليكم 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 (N.S)
وأريد أن أبين كيف قمنا بتعبئة العمود (TA_2) :
لإجاد العمود (TA_2) يجب علينا أولا إيجاد العمود (A_2) الذي هو جزء من أعمدة (P.S)
وكذلك إيجاد العمود (A_2) الذي هو جزء من أعمدة (N.S)
وقد قمنا بإيجادها .

بعد ذلك نقوم بالتعويض في جدول (Excitation table)
وبالتالي نكون قد إنتهينا من إيجاد جدول (State table) للعداد الموضح برسمة (State diagram) السابقة .

T_{A2}

$A_2 \backslash A_1A_0$	00	01	11	10
0			1	
1			1	

$T_{A2} = A_1A_0$

T_{A1}

$A_2 \backslash A_1A_0$	00	01	11	10
0		1	1	
1		1	1	

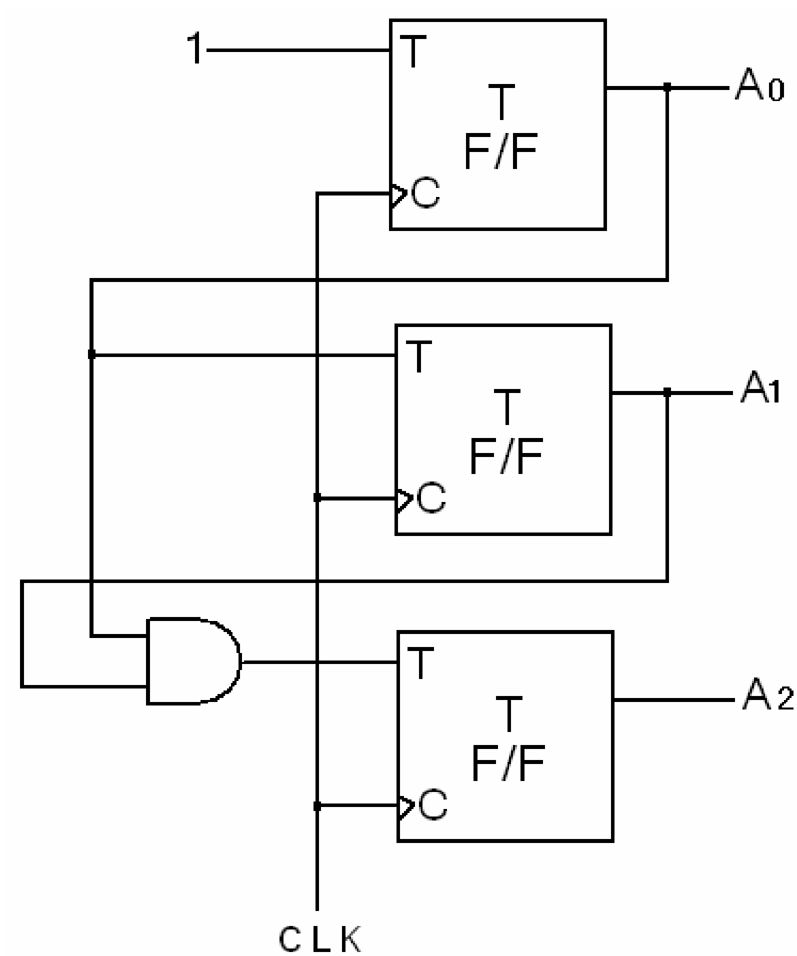
$T_{A1} = A_0$

T_{A0}

$A_2 \backslash A_1A_0$	00	01	11	10
0	1	1	1	1
1	1	1	1	1

$T_{A0} = 1$

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 نكون قد أوجدنا حل المثال السابق .

: Example مثال

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

: Selution الحل

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 مجددا إلى أول قيمة وهي العدد 0

أي أن تسلسل القيم السابقة مثل الحلقة المتصلة وسوف يصبح التسلسل بالشكل التالي :

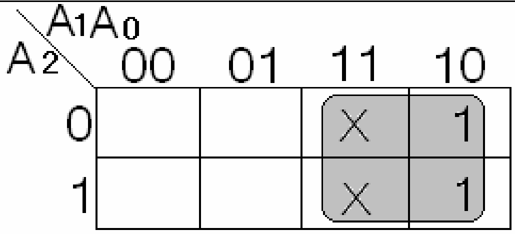
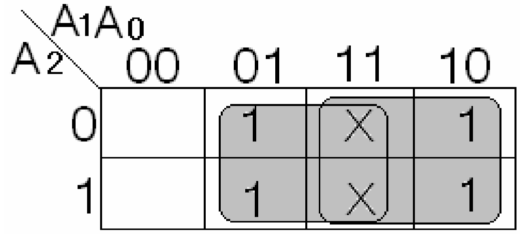
$0 \rightarrow 1 \rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 0$

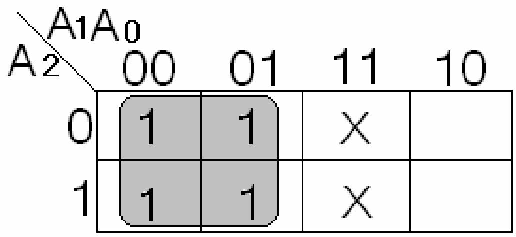
توجد بعض القيم الموجودة في (P.S) ولكن لا يوجد لها قيم تتجه إليها في (N.S) وهي القيمتين 7 و 3

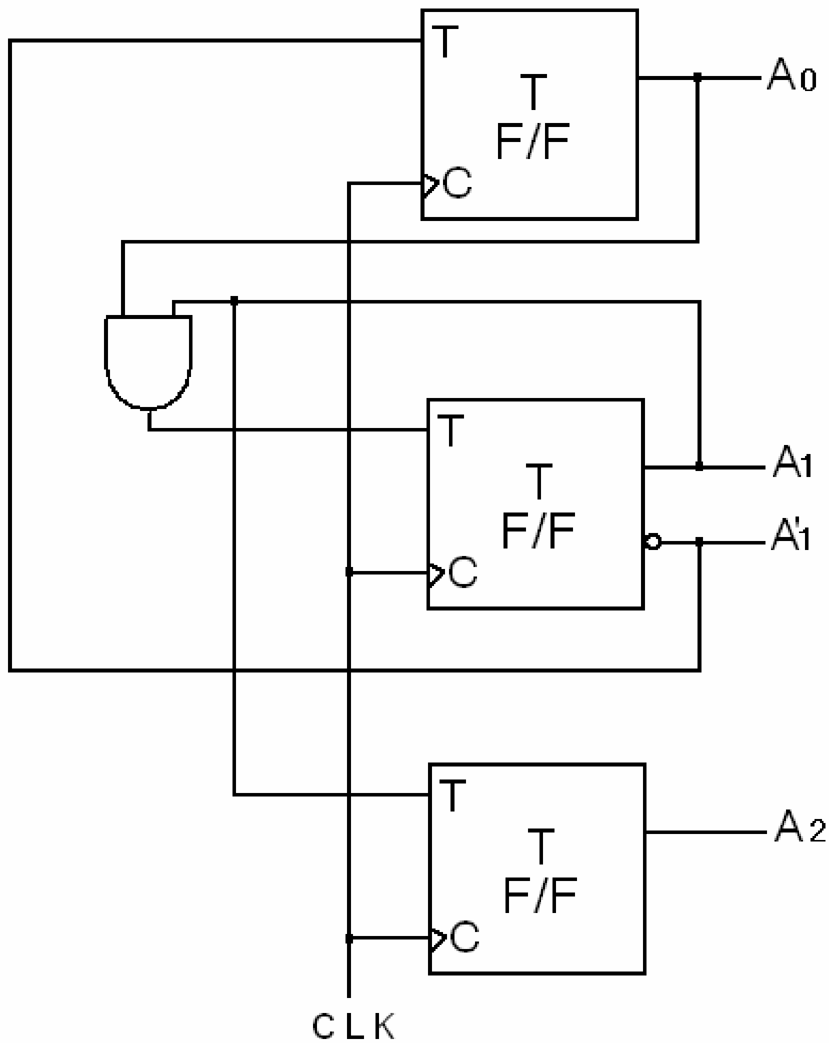
أي قيمة في (P.S) لا يوجد لها قيمة تتجه إليها في (N.S) يكون إتجاهها مباشرة إلى (Don't care)

بعد ذلك نقوم بإكمال الحل كما تعلمنا سابقاً.

P.S			N.S					
A_2	A_1	A_0	A_2	A_1	A_0	TA_2	TA_1	TA_0
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
	
TA0 = 1	TA1 = A0

TA0

TA2 = A1A0



الخاتمة

أسأل الله العلي القدير أن أكون وفقت في عمل هذا الكتاب وأعطى 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 ومراسلتي عبر البريد الإلكتروني المذكور في الكتاب

وكذلك إذا كان لديكم أي ملاحظات وإقتراحات حول العمل أرجو أن لا تبخلوا علي بها أيضاً

وذلك لكي تكون النسخة القادمة بمشئية الله تعالى من هذا الكتاب أكثر شمولا وأعم فائدة

وختاماً أسأل 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 الحاسب الآلي

Ahmad_911@hotmail.com