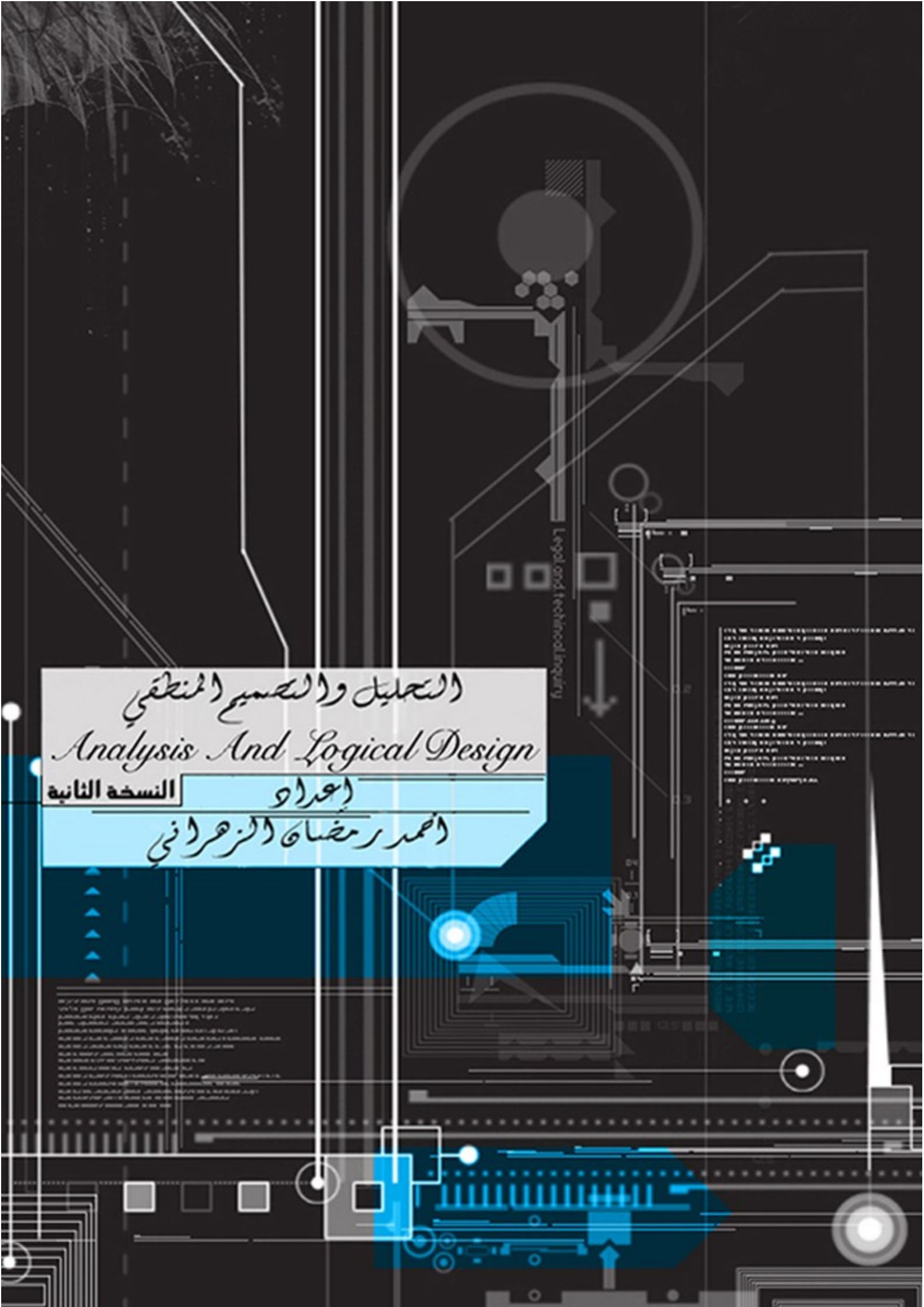
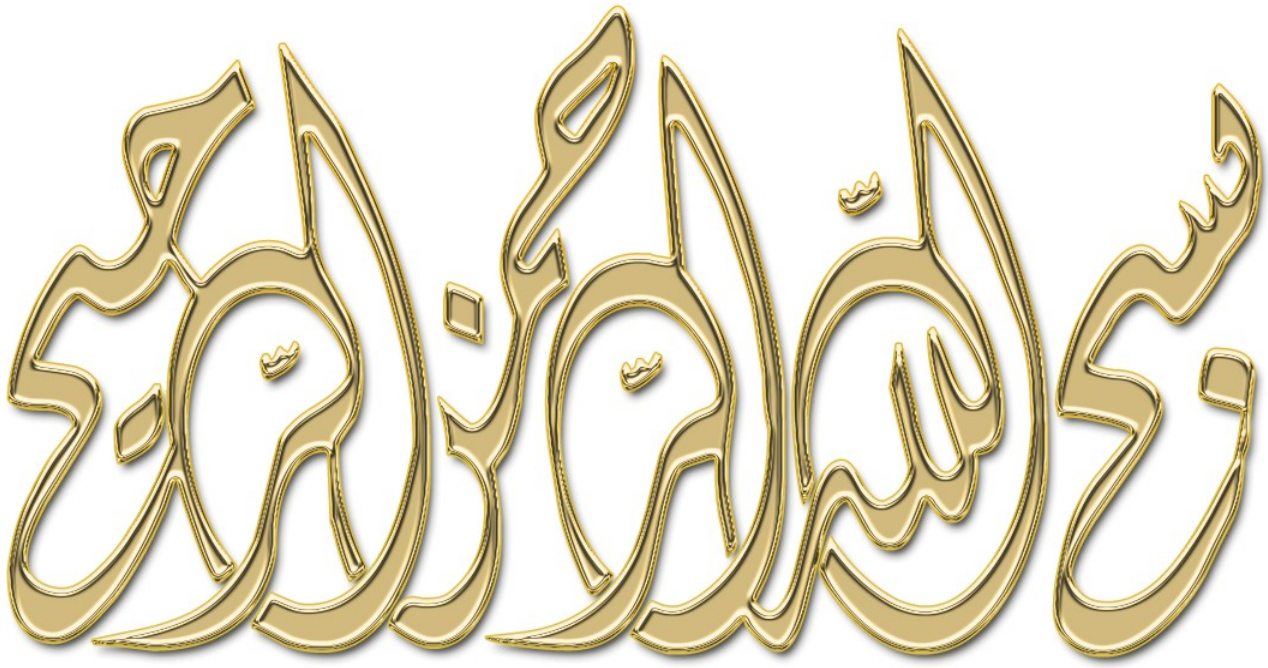

التحليل والتصميم المنطقي
Analysis And Logical Design
النسخة الثانية
إعداد
أحمد رمضان الزهراني



المقدمة

بسم الله الرحمن الرحيم

الحمد لله رب العالمين والصلاة والسلام على أشرف الأنبياء وسيد المرسلين

سيدنا ونبينا محمد عليه أفضل الصلاة وأتم التسليم وبعد :

بعد نجاح النسخة الأولى من كتابي في مادة التحليل والتصميم المنطقي

وذلك بتوفيق من الله عز وجل أولاً وأخيراً ثم بملاحظاتكم واهتمامكم وتشجيعكم

كان ولابد من مواصلة هذا النجاح والعمل على تطوير هذا الكتاب ليكون في أجمل وأتم صورة

وقد منّ الله علي بأن وفقني لعمل النسخة الثانية منه

تحتوي النسخة الجديدة على تعديل الأخطاء المنطقية والإملائية

التي كانت في النسخة الأولى والتي أعتذر منكم على حدوثها وأتمنى أنه تمت معالجتها جميعها

كذلك قمت بإعادة صياغة شرح كثير من النقاط الغير واضحة في النسخة السابقة

وأتمنى أن تكون أوضح في هذه النسخة بإذن الله تعالى

وأخيراً قمت بإضافة حلول أهم تمارين كتاب Digital Design للمؤلف M. Morris Mano

وهو الكتاب المقرر على دراسة مادة التحليل والتصميم المنطقي

والمرجع الذي اعتمدت عليه بعد الله عز وجل في صياغة هذا الكتاب

وكذلك لا أنسى جهود معلمي الفاضل المهندس / ماهر الشقنقيري والذي كان خير معلم لها

أشكر كل من قدم لي فكرة .. ملاحظة .. تشجيعاً .. اهتماماً

مما زادني إصراراً على عمل النسخة الثانية من هذا الكتاب

كذلك أشكر الأخت / تحكمني قناعاتي على الغلاف الرائع للكتاب

ختاماً أتمنى أن يكون هذا الكتاب عوناً لكم في فهم مادة التحليل والتصميم المنطقي

وأن ينال على رضاكم واستحسانكم

أسأل الله العلي القدير الأجر والمثوبة على هذا العمل وأن يكون خالصاً لوجهه الكريم

إنه ولي ذلك والقادر عليه

وصلى الله على سيدنا محمد وعلى آله وصحبه وسلم

أحمد رمضان الزهراني
طالب في جامعة أم القرى
Ahmad_911@hotmail.com

الإهداء

أهدي هذا الكتاب إلى من علمتني وربتني

إلى الشمعة التي ذابت لتنير دربي .. إلى البلسم الشافي لكل جروحي

إلى مصدر العطاء .. إلى بحر الحنان .. إلى صدق الإحساس .. إلى مصنع الأفراح

إلى أمي الحبيبة

أدام الله عليها لباس الصحة والعافية وأطال الله في عمرها

كما أتشرف بإهداء هذا الكتاب لموقع/ الحاسب الآلي ونظم المعلومات www.uqucs.com

والذي سوف يخصص كتابي وسيحظى كتابي بشرف رعاية الموقع له

كما أهدي هذا الكتاب لمعلمي الفاضل لهذه المادة المهندس/ ماهر السقنقيري

ولكل من يقرأ هذا الكتاب وسوف يستفيد منه بإذن الله تعالى

محبكم/ أحمد رمضان الزهراني

Ahmad_911@hotmail.com

الفهرس

الموضوع	الصفحة
الفصل الأول Binary Systems	9
1-1 مقدمة	10
1-2 أنظمة الأعداد Digital Systems.....	10
1-3 التحويل بين أنظمة الأعداد Number Base Conversions.....	11
1- من النظام الثنائي إلى النظام العشري From Binary To Decimal.....	11
2- من النظام الثماني إلى النظام العشري Form Octet To Decimal.....	12
3- من النظام الست عشري إلى النظام العشري Form Hexadecimal To Decimal.....	13
4- من النظام العشري إلى النظام الثنائي Form Decimal To Binary.....	14
5- من النظام العشري إلى النظام الثماني Form Decimal To Octet.....	16
6- من النظام العشري إلى النظام الست عشري Form Decimal To Hexadecimal.....	17
7- من النظام الثنائي إلى النظام الثماني Form Binary To Octet.....	18
8- من النظام الثماني إلى النظام الثنائي Form Octet To Binary.....	20
9- من النظام الثنائي إلى النظام الست عشري Form Binary To Hexadecimal.....	21
10- من النظام الست عشري إلى النظام الثنائي Form Hexadecimal To Binary.....	22
1-4 العمليات على الأعداد الثنائية Operations On Binary Numbers.....	23
1- المتممة الأولى One's Complement.....	23
2- المتممة الثانية Two's Complement.....	24
3- الجمع والطرح Adding & Subtraction.....	26
1-5 حل أهم تمارين الفصل.....	28
الفصل الثاني Boolean Algebra And Logic Gates	32
2-1 مقدمة	33
2-2 المنطق الثنائي Binary Logic.....	33
2-3 القواعد Grammars.....	33
2-4 Logic Gates.....	34
2-5 متممة الدالة Complement Of a Function.....	36
2-6 Canonical And Standard Forms.....	37
2-7 Digital Logic Gates.....	41
2-8 حل أهم تمارين الفصل.....	43

الموضوع	الصفحة
الفصل الثالث Gate – Level Minimization	46
3-1 مقدمة.....	47
3-2 طريقة الخريطة Map Method.....	47
3-3 خريطة 3 متغيرات 3 Variable Map.....	50
3-4 خريطة 4 متغيرات 4 Variable Map.....	53
3-5 Don't Care Conditions.....	55
3-6 حل أهم تمارين الفصل.....	56
الفصل الرابع Combinational Logic	58
4-1 مقدمة.....	59
4-2 إجراء التحليل Analysis Procedure.....	59
4-3 إجراء التصميم Design Procedure.....	60
4-4 Half Adder & Full Adder.....	67
1- Half Adder.....	67
2- Full Adder.....	68
4-5 Decoders.....	70
1- 2*4 Decoder.....	70
2- 3*8 Decoder.....	71
3- 4*16 Decoder.....	71
4-6 Decoder With Enabel.....	72
4-7 Multiplexers.....	75
1- 2*1 Multiplexer.....	75
2- 4*1 Multiplexer.....	76
3- 8*1 Multiplexer.....	76
4-8 حل أهم تمارين الفصل.....	81

الموضوع	الصفحة
الفصل الخامس Synchronous Sequential Logic	95
5-1 مقدمة.....	96
5-2 أنواع الـ Types Of Flip Flops.....	96
D Flip Flop -1.....	96
J K Flip Flop -2.....	97
T Flip Flop -3.....	98
5-3 Analysis Of Clocked Sequential Circuits.....	101
State Table -.....	102
State Diagram -.....	104
5-4 State Reduction And Assignment.....	110
5-5 إجراء التصميم Design Procedure.....	115
5-6 حل أهم تمارين الفصل.....	120
الفصل السادس Registers and Counters	134
6-1 مقدمة.....	135
6-2 المسجل Register.....	135
Shift Register -1.....	135
Rotate Register -2.....	136
6-3 العداد Counter.....	139
6-4 حل أهم تمارين الفصل.....	146
الخاتمة	153

الفصل الأول

Binary Systems

1-1 مقدمة :

في هذا الفصل سوف نتحدث عن أنظمة الأعداد (Digital Systems) وهي :

1- النظام العشري (Decimal System)
2- النظام الثنائي (Binary System)
3- النظام الثماني (Octet System)
4- النظام 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:

1- المتممة الأولى (One's Complement)
2- المتممة الثانية (Two's Complement)
3- الجمع والطرح (Adding & Subtraction)

1-2 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

1-3 التحويل بين أنظمة الأعداد Number Base Conversions :

1- من النظام الثنائي إلى النظام العشري From Binary To Decimal :

مثال :

حول العدد الثنائي التالي إلى النظام العشري :

$(1011)_2$

الحل :

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

الشرح :

في هذا المثال قمنا بتحويل العدد $(1011)_2$ من النظام الثنائي إلى النظام العشري وطريقة التحويل كالتالي :

نبدأ بأخذ الأعداد من اليمين إلى اليسار

أول عدد هو العدد (1) ونقوم بضربه في 2^0

ثم نأخذ العدد الثاني وهو العدد (1) ونضربه في 2^1

ثم نأخذ العدد الثالث وهو العدد (0) ونضربه في 2^2

وأخيراً نأخذ العدد الرابع والأخير وهو العدد (1) ونضربه في 2^3

بعد ذلك نقوم بجمع حاصل ضرب الأعداد السابقة

حاصل عملية الجمع يمثل العدد في النظام العشري = 11

مثال :

حول العدد الثنائي التالي إلى النظام العشري :

$(110.1)_2$

الحل :

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

الشرح :

في هذا المثال العدد الثنائي مكوّن من جزئين الجزء الأول صحيح والجزء الثاني كسري

وعند التحويل نعامل كل جزء على حدا

العدد الصحيح نعمل معه كما تعلمنا في المثال السابق

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

نبدأ بأخذ الأعداد من اليسار إلى اليمين

وفي مثلنا لدينا فقط عدد واحد كسري وهو العدد (1) ونقوم بضربه في 2^{-1}

ثم نجمعه 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:

حول العدد الثنائي التالي إلى النظام العشري :

$(1100.101)_2$

الحل:

$(1100.101)_2$

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

2- من النظام الثماني إلى النظام العشري From Octet To Decimal :

مثل طريقة تحويل العدد من النظام الثنائي إلى النظام العشري

ولكن الاختلاف فقط أننا سوف نضرب العدد الثماني في العدد (8) الذي يمثل أساس النظام المحول منه

مثال:

حول العدد الثماني التالي إلى النظام العشري :

$(752)_8$

الحل:

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

مثال:

حول العدد الثماني التالي إلى النظام العشري :

$(35.6)_8$

الحل:

$$(35.6)_8 = 5*8^0 + 3*8^1 + 6*8^{-1}$$
$$= 5 + 24 + 0.75$$
$$= (29.75)_{10}$$

3- من النظام الست عشري إلى النظام العشري From Hexadecimal To Decimal :

لا تختلف طريقة التحويل من النظام الست عشري إلى النظام العشري عن التحويلات السابقة إلا فقط في الأساس الذي سوف نضرب فيه العدد الست عشري المراد تحويله وهو العدد (16)

مثال :

حول العدد الست عشري التالي إلى النظام العشري :

$(ABC)_{16}$

الحل :

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

مثال :

حول العدد الست عشري التالي إلى النظام العشري :

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

الحل :

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

الخلاصة :

عند التحويل من أي نظام (الثنائي أو الثماني أو الست عشري) إلى النظام (العشري) فإننا نضرب العدد المراد تحويله في أساس نظامه المحول منه

4- من النظام العشري إلى النظام الثنائي From Decimal To Binary :

مثال :

حول العدد العشري التالي إلى النظام الثنائي :

$(59)_{10}$

الحل :

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

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

الشرح :

في هذا المثال قمنا بتحويل العدد 59 من النظام العشري إلى الثنائي والطريقة هي :
قسمة العدد العشري المراد تحويله على أساس النظام المحول إليه

حيث قمنا بقسمة العدد (59) على أساس النظام المحول إليه وهو العدد 2 نتج عن عملية القسمة العدد (59.5)

ولكن نحن نريد عدد صحيح فقط بدون كسور
ثم نقوم بضرب العدد الكسري الناتج عن عملية القسمة (0.5) في أساس النظام المحول إليه وهو العدد 2
ينتج عن عملية الضرب العدد (1) ويعتبر أول باقي عملية القسمة ويكتب في الطرف الثاني على يمين الأعداد

بعد ذلك نقوم بقسمة ناتج عملية القسمة الأولى وهو العدد 29 على 2
وهكذا نعمل مع باقي نواتج عمليات القسمة إلى أن نصل إلى أن يكون ناتج القسمة = 0 والباقي = 1
بواقي عمليات القسمة وهو العدد (111011) يمثل العدد 59 في النظام العشري

توجد بعض الملاحظات التي لابد أن نراعيها أثناء عملية التحويل وهي :
1- لو كان ناتج عملية القسمة عدد صحيح بدون كسور كما حدث في مثالنا السابق حيث كان
14 / 2 = 7 ولا يوجد باقي , عندها إذن يكون الباقي = 0
2- عند كتابة ناتج عملية التحويل يكتب العدد من الأسفل إلى الأعلى

مثال :

حول العدد العشري التالي إلى النظام الثنائي :

$(0.78125)_{10}$

الحل :

$0.78125 * 2 = 1.5625 \rightarrow 1$

$0.5625 * 2 = 1.125 \rightarrow 1$

$0.125 * 2 = 0.25 \rightarrow 0$

$0.25 * 2 = 0.5 \rightarrow 0$

$0.5 * 2 = 1 \rightarrow 1$

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

الشرح :

في هذا المثال قمنا بتحويل العدد (0.78125) من النظام العشري إلى النظام الثنائي طريقة التحويل هي كالآتي :

كما تلاحظ عزيزي القارئ أن العدد العشري عدد كسري وليس صحيح

عند تحويل العدد الكسري نقوم بضربه في أساس النظام المحول إليه وهو العدد 2

العدد العشري المراد تحويله هو العدد (0.78125) نقوم بضربه في العدد 2

ينتج عن عملية الضرب العدد (1.5625)

نأخذ الجزء الصحيح وهو العدد (1) ويعتبر أول عدد ناتج عن عملية التحويل

ويتبقى الجزء الكسري وهو العدد (0.5623) ونكرر معه الخطوات السابقة

إلى أن نصل أن يكون ناتج عملية الضرب عدد صحيح فقط بدون كسور, عندها تكون قد انتهت عملية التحويل

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

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

عكس طريقة كتابة تحويل العدد الصحيح

2- 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 :

حول العدد العشري التالي إلى النظام الثنائي :

$(35.375)_{10}$

الحل :

هذا العدد مكوّن من جزئين جزء صحيح والآخر كسري وعند تحويله إلى النظام الثنائي

نعامل كل جزء على حدا

أي نأخذ الجزء الصحيح ونحوله ثم نأخذ الجزء الكسري ونحوله ثم نكتب العدد كاملاً

$0.375 * 2 = 0.75 \rightarrow 0$

$0.75 * 2 = 1.5 \rightarrow 1$

$0.5 * 2 = 1 \rightarrow 1$

(0.011)

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

(100011)

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

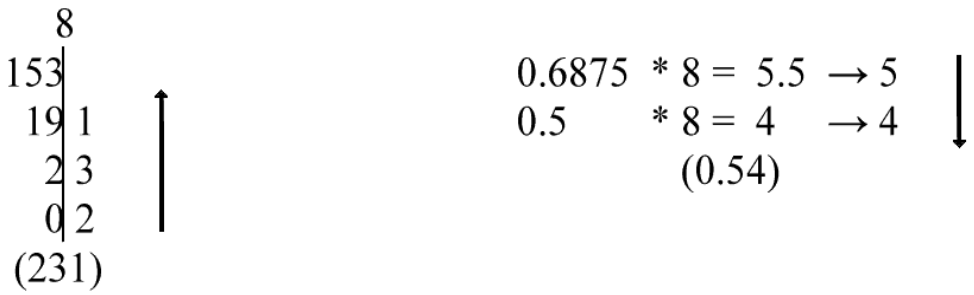
5- من النظام العشري إلى النظام الثماني From Decimal To Octet :

مثال :

حول العدد العشري التالي إلى النظام الثماني :

$(153.6875)_{10}$

الحل :



$(153.6875)_{10} = (231.54)_{8}$

الشرح :

لا تختلف طريقة التحويل العدد من النظام العشري إلى النظام الثماني عن طريقة تحوليه إلى النظام الثنائي سواءً كان العدد صحيح أم كسري ولكن الاختلاف فقط

في الأساس الذي نقسم أو نضرب العدد العشري فيه وهو العدد 8

توجد نقطة مهمة سبق وأن تحدثنا عنها في طريقة تحويل العدد العشري الصحيح إلى النظام الثنائي وهي إيجاد الباقي

عند قسمة العدد (153) على العدد 8 ينتج عن عملية القسمة العدد (19.125)

نأخذ الجزء الكسري وهو العدد (0.125) ونضربه في العدد 8 ينتج عن عملية الضرب العدد (1) ويعتبر العدد (1) هو باقي عملية القسمة

ثم نكمل خطوات التحويل كما تعلمنا في الأمثلة السابقة

وتنتهي عملية التحويل عندما يكون ناتج القسمة = 0 والباقي = عدد صحيح أصغر من العدد 8

بعد ذلك ننتقل للجزء الكسري

وطريقة تحويله إلى النظام الثماني هي نفس طريقة تحويله إلى النظام الثنائي

مع اختلاف الأساس الذي نضرب فيه العدد العشري الكسري وهو العدد 8

وتنتهي عملية التحويل عندما يصبح الناتج = عدد صحيح

6- من النظام العشري إلى النظام الست عشري From Decimal To Hexadecimal :

مثل طريقة التحويل إلى الأنظمة السابقة والاختلاف كما وسبق أن ذكرنا
في الأساس الذي سوف نضرب أو نقسم عليه العدد العشري وهو العدد 16
توجد ملاحظة بسيطة وهي كما سوف تلاحظها عزيزي القارئ في المثال التالي
أن باقي عملية قسمة العدد 125 على 16 = 13
(الباقي = 13 , 7 = 125 /16)
العدد 13 لا يكتب بصورته المعروفة في النظام العشري
وإنما يكتب بصورته في النظام الست عشري أن يكتب (D)

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

16
125
7 13
0 7
(7D)

0.34375 * 16 = 5.5 → 5
0.5 * 16 = 8 → 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 المراد تحويله عدد صحيح فإننا نقسمه على أساس النظام المحول إليه
أما إذا كان العدد العشري المراد تحويله عدد كسري فإننا نضربه في أساس النظام المحول إليه

7- من النظام الثنائي إلى النظام الثماني From 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 :

حول العدد الثنائي التالي إلى النظام الثماني :

$(10011101110)_2$

الحل :

010	011	101	110
2	3	5	6

$(10011101110)_2 = (2356)_8$

الشرح :

عند التحويل من النظام الثنائي إلى النظام الثماني نقوم بأخذ كل 3 أعداد ثنائية من جهة اليمين
لماذا نبدأ بأخذ الأعداد من جهة اليمين ولم نبدأ بأخذها من جهة اليسار ؟
وذلك لأنه في بعض الأحيان يتبقى عدد ثنائي بمفرده دون الثاني والثالث
أو عددين ثنائيين بمفردهما دون الثالث
عندها نقوم نحن بتكملة الأعداد الناقصة بإضافة أصفار لها وذلك لكي تصبح مكوّنة من 3 أعداد
كما فعلنا في المثال حيث أضفنا العدد (0) على آخر عددين وهما (01) وأصبح العدد = (010)
وكما نعلم أن الصفر في خانة اليسار ليس له قيمة كما في المثال التالي :

$(001) = (1)$

أما لو كنا نأخذ الأعداد من جهة اليمين فعندما نريد تكملة الأعداد الناقصة
سوف نضيف الصفر من جهة اليمين وبذلك يصبح للصفر قيمة وبالتالي يختلف العدد تماماً
كما في المثال التالي :

$(100) \neq (1)$

وبالتالي نكون قد قسمنا العدد الثنائي إلى عدة أقسام كل قسم مكوّن من 3 أعداد
ثم نضع قيمة العدد الثماني المقابل لكل 3 أعداد ثنائية وذلك من خلال الجدول

مثال :

حول العدد الثنائي التالي إلى النظام الثماني :

$(.0101111)_2$

الحل :

010	111	100
2	7	4

$(.0101111)_2 = (.274)$

الشرح :

في هذا المثال العدد الثنائي المراد تحويله عدد كسري
والفرق بين تحويل العدد الثنائي الكسري عن العدد الثنائي الصحيح
أننا في الصحيح نأخذ كل 3 أعداد ثنائية من جهة اليمين ونضيف الأصفار على العدد من جهة اليسار

أما في العدد الكسري فإننا نعمل العكس تماماً
نأخذ كل 3 أعداد ثنائية من جهة اليسار (أول عدد بعد الفاصلة)
ونضيف الأصفار لتكملة العدد من جهة اليمين .. لماذا ؟ أدع الإجابة لك عزيزي القارئ

ثم نكمل باقي خطوات الحل كما تعلمنا في المثال السابق

مثال :

حول العدد الثنائي التالي إلى النظام الثماني :

$(11001.01)_2$

الحل :

011	001	.010
3	1	2

$(11001.01)_2 = (31.2)_8$

الشرح :

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

8- من النظام الثماني إلى النظام الثنائي From 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:

حول العدد الثماني التالي إلى النظام الثنائي :

$(62.7)_8$

الحل:

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

مثال:

حول العدد الثماني التالي إلى النظام الثنائي :

$(35.41)_8$

الحل:

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

9- من النظام الثنائي إلى النظام الست عشري From Binary To Hexadecimal :

هي نفس طريقة التحويل من النظام الثنائي إلى النظام الثماني
ولكن الاختلاف فقط أن كل عدد ست عشري يكافئ 4 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 :
حول العدد الثنائي التالي إلى النظام الست عشري :
$(0010\ 1110.\ 1010)_2$
الحل :
$(0010\ 1110.1010)_2 = (2E.A)_{16}$

مثال :
حول العدد الثنائي التالي إلى النظام الست عشري :
$(1111\ \ 1100.\ 0101\ 1011)_2$
الحل :
$(1111\ \ 1100.\ 0101\ 1011)_2 = (FC.5B)_{16}$

10- من النظام الست عشري إلى النظام الثنائي From Hexadecimal To Binary :

مثال :

حول العدد الست عشري التالي إلى النظام الثنائي :

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

الحل :

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

مثال :

حول العدد الست عشري التالي إلى النظام الثنائي :

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

الحل :

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

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

1- المتممة الأولى One's Complement :

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

مثال :
أوجد المتممة الأولى للعدد الثنائي التالي :
$(1100101001)_2$
الحل :
المتممة الأولى = 0011010110

مثال :
أوجد المتممة الأولى للعدد الثنائي التالي :
$(10000000000)_2$
الحل :
المتممة الأولى = 01111111111

2- 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 موجبة

يتم إيجاد المتممة الثانية بإحدى الطريقتين :

الطريقة الأولى :

وذلك بأن توجد المتممة الأولى للعدد الثنائي ثم نجمع القيمة (1) على المتممة الأولى

مثال :

أوجد المتممة الثانية للعدد الثنائي التالي :

$(1100101001)_2$

الحل :

أولا : نوجد المتممة الأولى = 0011010110
ثانيا : نقوم بجمع القيمة 1 على المتممة الأولى

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

المتممة الثانية = 0011010111

مثال :

أوجد المتممة الثانية للعدد الثنائي التالي :

$(1111000000)_2$

الحل :

المتممة الأولى = 0000111111

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

المتممة الثانية = 0001000000

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

سوف نقوم بإيجاد المتممة الثانية للأمثلة السابقة بهذه الطريقة

مثال:

أوجد المتممة الثانية للعدد الثنائي التالي :

$(1100101001)_2$

الحل:

1100101001
0011010111

المتممة الثانية = 0011010111

الشرح:

في هذا المثال 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:

أوجد المتممة الثانية للعدد الثنائي التالي :

$(1111000000)_2$

الحل:

1111000000
0001000000

المتممة الثانية = 0001000000

الشرح:

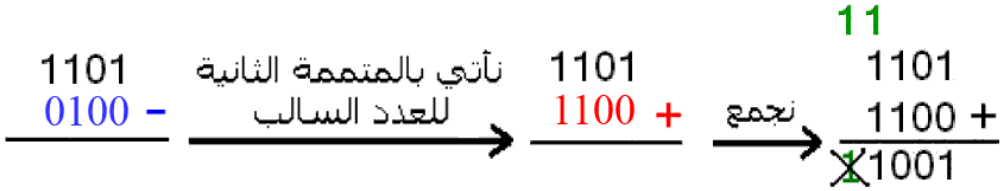
قمنا بكتابة أول ستة أرقام في الناتج كما هي إلى أن وصلنا للرقم السابع وهو أول رقم 1 في العدد
قمنا بكتابته في الناتج كما هو وغيرنا باقي العدد من بعده من 0 إلى 1 ومن 1 إلى 0

3- الجمع والطرح Adding & Subtraction :

في هذه الجزئية لن نتحدث عن عملية الجمع وإنما سوف نتحدث فقط عن عملية الطرح
وذلك لأن عملية الطرح في الأساس ماهي إلا عملية جمع عدد موجب مع عدد سالب

في النظام الثنائي لا يمكن إجراء عملية الطرح مباشرةً كما نفعل في النظام العشري
بل نقوم بتحويل عملية الطرح إلى عملية جمع وذلك باستخدام المتممة الثانية

مثال :
أجري عملية الطرح التالية :
1101 - 0100
الحل :



1101 - 1100 = +1001

الشرح :

في هذا المثال قمنا بإجراء عملية الطرح على عددين ثنائيين

العدد الأول وهو العدد (1101) عدد موجب لذلك نبقيه كما هو
أما العدد الثاني وهو العدد (0100) عدد سالب
إذن نوجد المتممة الثانية له لكي نحوله إلى عدد موجب
المتممة الثانية للعدد (0100) = (1100)
بعد ذلك نقوم بعملية الجمع العدد الأول (1101) والعدد الثاني بعد إيجاد المتممة الثانية له (1100)
1101 + 1100 = 11001
كما تلاحظ عزيزي القارئ أن ناتج عملية الجمع وهو العدد (11001) يوجد به (Overflow)
وذلك لأن العدد الأول يمثل في 4 بايت
والعدد الثاني كذلك يمثل في 4 بايت
أما الناتج فإنه يمثل في 5 بايت

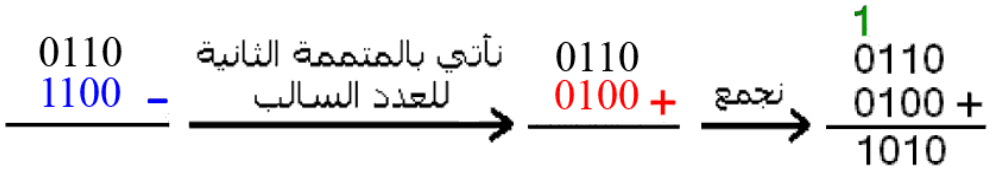
نحن نريد أيضاً الناتج يمثل في 4 بايت
لذلك نقوم بحذف آخر عدد من الناتج 1001 ونضع أمام الناتج إشارة +

مثال:

أجري عملية الطرح التالية :

0110 - 1100

الحل:



0110 - 1100 = -0110

الشرح:

في هذا المثال الناتج لا يوجد به Overflow لذلك نقوم بإيجاد المتممة الثانية للناتج ونضع أمامه 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 –

1-5 حل أهم تمارين الفصل :

: Express the following numbers in Decimal /Q7
$(10110.0101)_2$, $(16.5)_{16}$, $(26.24)_8$

المطلوب :
تحويل الأعداد إلى النظام العشري

الحل :

$$(10110.0101)_2 = 0*2^0 + 1*2^1 + 1*2^2 + 0*2^3 + 1*2^4 + 0*2^{-1} + 1*2^{-2} + 0*2^{-3} + 1*2^{-4}$$
$$= 0 + 2 + 4 + 0 + 16 + 0 + 0.25 + 0 + 0.0625$$
$$= 22.3125$$

$$(16.5)_{16} = 6*16^0 + 1*16^1 + 5*16^{-1}$$
$$= 6 + 16 + 03125$$
$$= 22.3125$$

$$(26.24)_8 = 6*8^0 + 2*8^1 + 2*8^{-1} + 4*8^{-2}$$
$$= 6 + 16 + 0.25 + 0.0625$$
$$= 22.3125$$

: Convert the Hexadecimal number $(68BE)_{16}$ to Binary and then / Q9
From Binary convert it to Octal

المطلوب :
تحويل العدد من النظام الست عشري إلى النظام الثنائي وبعد ذلك من النظام الثنائي إلى النظام الثماني

الحل :

$$(68BE)_{16} = (0110\ \ 1000\ \ 1011\ \ 1110)_2$$

$$(000\ \ 110\ \ 100\ \ 010\ \ 111\ \ 110)_2 = (64276)_8$$

Obtain the 1's and 2's Complements of the following Binary number / Q16:

(a) 11101010 , (b) 01111110 , (c) 00000001

(d) 10000000 , (e) 00000000

المطلوب :

إيجاد المتممة الأولى والمتممة الثانية للأعداد

الحل :

(a) 11101010

المتممة الأولى = 00010101

المتممة الثانية =

الطريقة الأولى	الطريقة الثانية
1	
00010101	11101010
1 +	
00010110	00010110

(b) 01111110

المتممة الأولى = 10000001

المتممة الثانية =

الطريقة الأولى	الطريقة الثانية
1	
10000001	01111110
1 +	
10000010	10000010

(c) 00000001

المتممة الأولى = 11111110

المتممة الثانية =

الطريقة الأولى

```
 11111110
        1 +
----------
 11111111
```

الطريقة الثانية

```
 00000001
----------
 11111111
```

(d) 10000000

المتممة الأولى = 01111111

المتممة الثانية =

الطريقة الأولى

```
  1111111
 01111111
        1 +
----------
 10000000
```

الطريقة الثانية

```
 10000000
----------
 10000000
```

(e) 00000000

المتممة الأولى = 11111111

المتممة الثانية =

الطريقة الأولى

```
 1 1111111
 11111111
        1 +
----------
X 00000000
```

الطريقة الثانية

```
 00000000
----------
 00000000
```

لوجود Overflow في ناتج الجمع نحذف آخر عدد في ناتج وذلك حسب قواعد جمع الأعداد الثنائية

Perform subtraction on the following unsigned Binary numbers / Q18
Using the 2'S-Complement of the subtrahend
Where the result should be negative 2'S-Complement it and affix a minus sign

(a) 11011 – 11001 , (b) 110100 – 10101
(c) 1011 – 110000 , (d) 101010 – 101011

المطلوب :

إجراء عمليات الطرح على الأعداد وذلك حسب قواعد طرح الأعداد الثنائية وهي :
إذا وجد Overflow نحذف آخر عدد من الناتج ونضع أمامه إشارة +
وإذا لم يوجد Overflow نوجد المتممة الثانية للناتج ونضع أمامه إشارة –

الحل :

(a) 11011 – 11001

11011 − 11001 → نأتي بالمتممة الثانية للعدد السالب → 11011 + 00111 → نجمع →

1 1111
11011
00111+
X 00010

11011 – 11001 = +00111

(b) 110100 – 10101

110100 − 010101 → نأتي بالمتممة الثانية للعدد السالب → 110100 + 101011 → نجمع →

1 110100
101011+
X 011111

110100 – 10101 = +01111

(c) 1011 – 110000

001011 − 110000 → نأتي بالمتممة الثانية للعدد السالب → 001011 + 010000 → نجمع → 001011 + 010000 = 011011 → نأتي بالمتممة الثانية للناتج → 100101

1011 – 110000 = – 100101

(d) 101010 – 101011

101010 − 101011 → نأتي بالمتممة الثانية للعدد السالب → 101010 + 010101 → نجمع → 101010 + 010101 = 111111 → نأتي بالمتممة الثانية للناتج → 000001

101010 – 101011 = – 000001

الفصل الثاني

Boolean Algebra And Logic Gates

2-1 مقدمة :

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 وأوضح 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 وعملها
وأيضاً سوف نتحدث عن Minterms و Maxterms وكيفية إيجادها

2-2 المنطق الثنائي Binary Logic :

الجدول التالي يوضح أهم الروابط المستخدمة لإنشاء الدوال (Functions) ويوضح أيضاً (Truth Table) لها :

		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

2-3 القواعد Grammars :

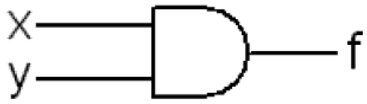
AND	
1	x.1 = x
2	x.x' = 0
3	x.x = x
4	x.0 = 0
5	(x')' = x
6	x.y = y.x
7	x.(y.z) = (x.y).z
8	x+y.z = (x+y).(x+z)
9	(x.y)' = x'+y'
10	x.(x+y) = x

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

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

: Logic Gates 4-5

الجدول التالي يوضح أشكال الروابط بالرسم وذلك لكي نتمكن من تمثيل الدوال بالرسم :

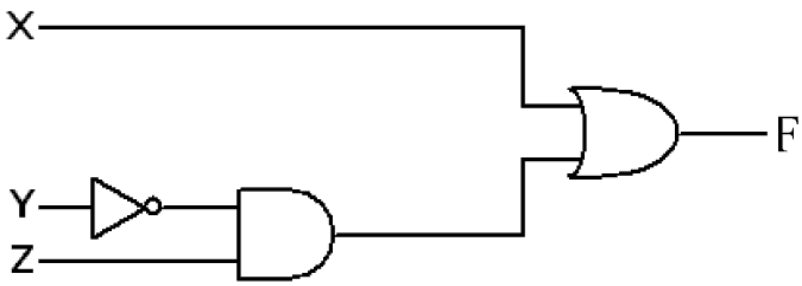
Name	Graphic Symbol	Algebraic Function
AND		F = xy
OR		F = x+y
Inverter		F = x'

مثال :

أرسم الدالة التالية :

$$F = x + y'z$$

الحل :

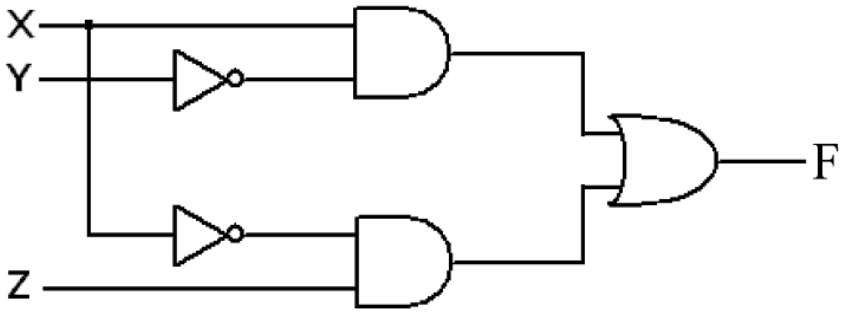


مثال:

أرسم الدالة التالية :

$F = xy' + x'z$

الحل:



مثال:

بسط الدوال المنطقية التالية Simplify the following Boolean functions :

1- $x(x' + y)$

2- $x + x'y$

3- $(x + y).(x + y')$

4- $xy + x'z + yz$

الحل:

سوف نقوم بتبسيط هذه الدوال بطريقة الجبر البوليني (Boolean Algebra) معتمدين على قواعد الروابط AND و OR

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- $xy + x'z + yz = xy + x'z + yz.(x+x')$ $= xy + x'z + xyz + x'yz$ $= xy(1+z) + x'z(1+y)$ $=xy +x'z$

2-5 متممة الدالة Complement Of a Function :

لإيجاد متممة الدالة يجب علينا إستخدام قواعد (De Morgan) وهي :

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

تتم هذه القاعدة بنفي الدالة كاملة وعند نفيها يحدث ما يلي :
1- تتحول الروابط بين عناصر الدالة من AND إلى OR والعكس
2- كل عنصر غير منفي يصبح منفي والعكس

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

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

الحل:

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

الشرح:
كما تلاحظ عزيزي القارئ أن المثال مكوّن من حدين

خطوات الحل كالتالي :
1- تغيير الرابط بين الحدين من OR إلى AND
2- نفي كل حد على حدا
3- تغيير الروابط بين عناصر كل حد من AND إلى OR
4- نفي كل عنصر مثبت وإثبات كل عنصر منفي

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

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

الحل:

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

الجدول التالي يوضح كيفية كتابة (Truth Table) لـ (X,Y,Z) بواسطة Minterms و Maxterms :

			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

الجدول التالي يقارن بين Minterms و Maxterms :

	Minterms	Maxterms
1	الرابط (AND) هوالذي يربط بين عناصر الدالة	الرابط (OR) هوالذي يربط بين عناصر الدالة
2	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 قمنا بقلب كل 0 إلى 1 والعكس
3	يمثل قيمة الدالة عندما تكون قيمة الدالة = 1	يمثل قيمة الدالة عندما تكون قيمة الدالة = 0

مثال :

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 :

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 :

$$F1 = (x+y+z)(x+y'+z)(x+y'+z')(x'+y+z')(x'+y+'z) = M0.M2.M3.M5.M6 \rightarrow \prod(0,2,3,5,6)$$

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

مثال:

: Express the Boolean function F = A + B'C in a Sum Of Minterms

الحل:

هنالك طريقتان للحل وأنا أرى أن الطريقة الثانية أسهل

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 :

في هذا المثال قمنا بإيجاد Sum Of Minterms للدالة F = A + B'C

لاحظ عزيزي القارئ أن الدالة تحتوي على 3 متغيرات بغض النظر عن أن المتغير مثبت أم منفي وهي: A,B,C

وكذلك لاحظ أن الدالة مكوّنة من حدين الحد الأول (A) والحد الثاني (B'C)

عندما نريد أن نوجد Sum Of Minterms للدالة نعامل كل حد على حدا

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 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 عن عملية الضرب حدين وهما : (AB') , (AB) نقوم بضربها في (C+C')

وينتج عن عملية الضرب 4 حدود وهي : (AB'C') , (AB'C) , (ABC') , (ABC)

وبالتالي نكون قد أكملنا للحد الأول الحدود الناقصة

ثم ننتقل للحد الثاني وكما تلاحظ ينقص هذا الحد متغير واحد وهو المتغير (A)

ونقوم بعمل نفس الخطوات التي عملناها مع الحد الأول

وبالتالي يصبح لدينا 6 حدود وأخيراً نقوم باختصار 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 :

هي ليست طريقة مختلفة عن الطريقة الأولى وإنما هي فقط اختصار لخطوات الحل

الحد الأول (A) ينقصه المتغيرين C و B ونقوم بضربه في هذه الحدود مباشرة بكل احتمالاتهما وأقصد أن يكونا :

مرة (C و B) مثبتين , ومرة (B) مثبت و(C) منفي , ومرة (B) منفي و(C) مثبت , وأخيراً (C و B) منفيين

الحد الثاني (B'C) ينقصه المتغير A ونعمل معه نفس الخطوات التي قمنا بعملها مع الحد الأول

بعد ذلك نكتب كامل الحدود الناتجة ونختصر الحدود المتكررة كما فعلنا مع الحدود الملونة بالأحمر

مثال :

: Express the Boolean function F = xy + x'z in a Product Of Maxterms from

الحل :

الطريقة الأولى :

$$
\begin{aligned}
F &= xy + x'z \\
&= (x+x'z)(y+x'z) \\
&= (x+x')(x+z)(y+x')(y+z) \\
&= 1(x+z)(y+x')(y+z) \\
&= (x+z)(y+x')(y+z) \\
&= (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z')(x+y+z)(x'+y+z) \\
&= (x+y+z)(x+y'+z)(x'+y+z)(x'+y+z') \\
&= M0 \cdot M2 \cdot M4 \cdot M5 = \prod(0,2,4,5) = \Sigma(1,3,6,7)
\end{aligned}
$$

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

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 حدا

نأخذ الحد الأول ونقوم بتوزيعه, ثم نأخذ الحد الثاني ونقوم بتوزيعه أيضاً

تنتج حدود أخرى عن عملية التوزيع ونستمر في توزيع الحدود إلى أن نصل أن يكون كل حد مكوّن من 3 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 :

كما سبق أن ذكرنا أن Minterms يمثل قيمة الدالة عندما تكون قيمة الدالة = 1

و Maxterms يمثل قيمة الدالة عندما تكون قيمة الدالة = 0 طبعاً لنفس الدالة

إيجاد Sum Of Minterms أسهل من إيجاد Product Of Maxterms

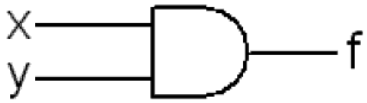
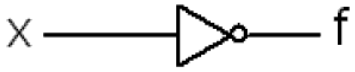
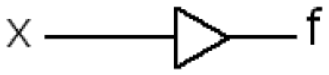
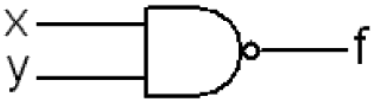
أقصد بذلك أنه لو طلب منك إيجاد Product Of Maxterms قم بإيجاد Sum Of Minterms

وعند إيجادك لـ Sum Of Minterms تكون قد أوجدت الحدود التي تكون عندها الدالة =1

حيث نتجت هذه الحدود التي تعطي مجموعة الحل هذه Σ(1,3,6,7)

أذن متممة الحل الذي أوجدناه وهي مجموعة الحل ∏(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	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 1 0 1 1 1 0 1 1 1 0
NOR		F = (x+y)'	X Y F 0 0 1 0 1 0 1 0 0 1 1 0

<table>
<tr><th>Name</th><th>Graphic Symbol</th><th>Algebraic Function</th><th>Truth Table</th></tr>
<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>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td>
</tr>
<tr>
<td>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>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table>
</td>
</tr>
</table>

2-8 حل أهم تمارين الفصل :

: Simplify the following Boolean expressions to a minimum number of literals / Q2

(a) xy + xy' , (b) (x+y)(x+y')

(c) xyz + x'y + xyz' , (d) (A+B)'(A' + B')

المطلوب :

تبسيط التعبيرات البولينية

الحل :

(a) xy + xy'	(b) (x+y)(x+y')
= x(y+y') = x.1 = x	= x(1+y)(1+y') = x.1.1 = x
(c) xyz + x'y + xyz'	(d) (A+B)'(A' + B')
= y(xz +x' + xz') = y(x(z+z')x') = y(x(1)x') = y.1 = y	= (A'B').(A'+B') = A'A'B'+A'B'B' = A'B' + A'B' = A'B'(0+0) = A'B'(0) = 0

: Find the Complement to the following expressions / Q6

(a) xy' + x'y , (b) (AB + C)D' + E , (c) (x + y' + z)(x' + z')(x' + y)

المطلوب :

إيجاد متممة التعبيرات (تطبيق قاعدة De Morgan عليها)

الحل :

(a) xy'+x'y
= (xy'+x'y)' = (xy')'.(x'y)' = (x'+y).(x+y')

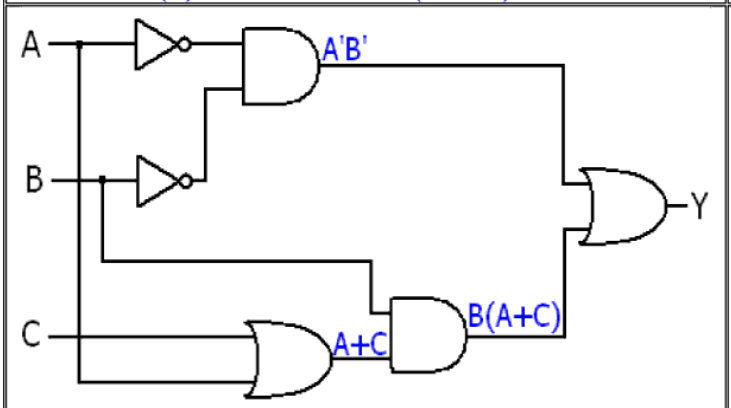
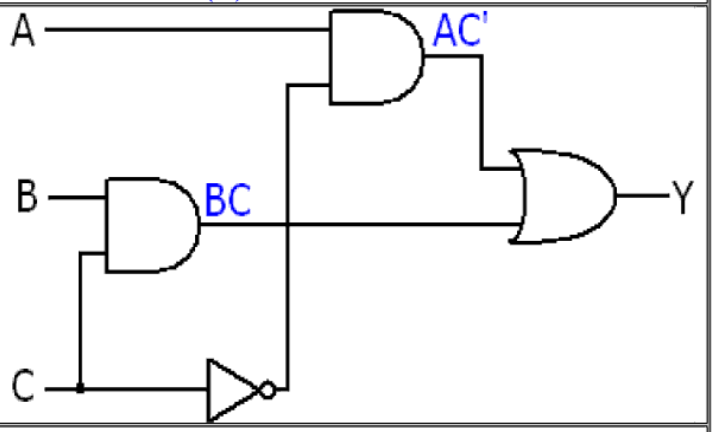
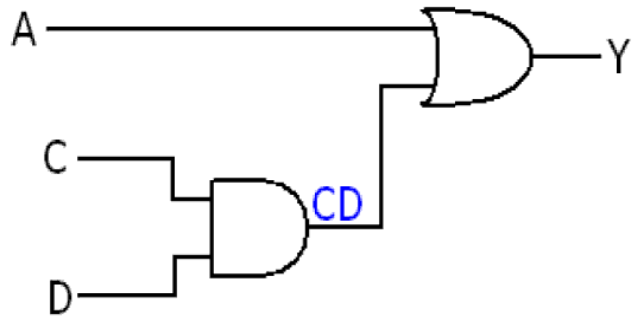
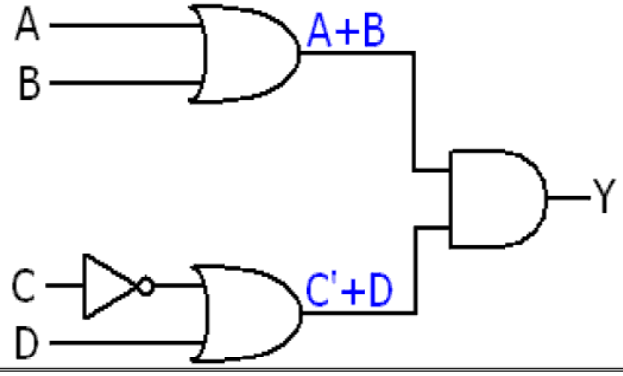
(b) (AB+C)D'+E	(c) (x+y'+z)(x'+z')(x+y)
= ABD' + CD' + E = (ABD' + CD' + E)' = (ABD')'.(CD')'.E' = (A'+B'+D).(C'+D).E'	= ((x+y'+z)(x'+z')(x'+y))' = (x+y'+z)'+(x' + z')'+(x'+y)' = (x'yz')+(xz)+(x'y')

: Draw the logic diagrams for the following Boolean expressions / Q10

(a) Y = A'B' + B(A + C) , (b) Y = BC + AC'
(c) Y = A + CD , (d) Y = (A + B)(C' + D)

المطلوب :
رسم التعبيرات البولينية

الحل :

<table>
<tr><td>(a) Y = A'B'+B(A+C)</td><td>(b) Y = BC +AC'</td></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>(c) Y = A+CD</td><td>(d) Y = (A+B)(C'+D)</td></tr>
<tr><td> 
</td><td> 
</td></tr>
</table>

:Express the following function in Sum Of Minterms and Product Of Maxterms /Q16

$$F(A,B,C,D) = B'D + A'D + BD$$

المطلوب :

التعبير بـ Sum Of Minterms و Product Of Maxterms للدالة

الحل :

$$\begin{aligned} F = \quad & AB'CD + AB'C'D + A'BCD + A'BC'D \\ & + A'BCD + A'BC'D + A'B'CD + A'B'C'D \\ & + ABCD + ABC'D + A'BCD + A'BC'D \end{aligned}$$

$$F = AB'CD + AB'C'D + A'BCD + A'BC'D + A'B'CD + A'B'C'D + ABCD + ABC'D$$

$$F = \sum(1,3,5,7,9,11,13,15) = \Pi(0,2,4,6,8,10,12,14)$$

: Convert the following to the other canonical from /Q18

(a) $F(x,y,z) = \sum(1,3,7)$, (b) $F(A,B,C,D) = \Pi(0,1,2,3,4,12)$

الحل :

(a) $F(x,y,z) = \sum(1,3,7)$	(b) $F(A,B,C,D) = \Pi(0,1,2,3,4,12)$
$F(x,y,z) = \Pi(0,2,4,5,6)$	$F(A,B,C,D) = \sum(2,5,6,7,8,9,10,11,13,14,15)$

الفصل الثالث

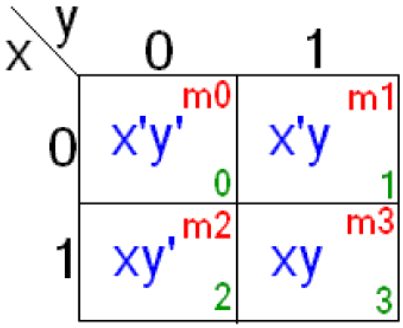
Gate - Level Minimization

3-1 مقدمة:

في هذا الفصل سوف نتحدث عن كيفية تبسيط الدوال (Functions) بواسطة (Karnaugh Map)
وسوف نتعرف عن أشكال Karnaugh Map وكيفية استخدامها

3-2 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



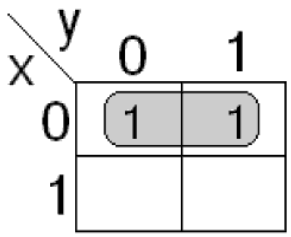
الشكل السابق يوضح الشكل العام لـ Karnaugh Map لمتغيرين
X لها قيمتين(0,1) وكذلك Y لها نفس القيم
تقاطع قيم X مع قيم Y تكون قيم هذه الخريطة
X يمثل الجانب العمودي بينما Y يمثل الجانب الأفقي

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



$F(x,y) = x'$

الشرح :

في هذا المثال قمنا بتبسيط الدالة $x'y + x'y'$ وكما تلاحظ عزيزي القارئ الدالة مكوّنة من حدين

الحد الأول $(x'y)$ مكوّن من تقاطع العمود X عند القيمة (0) أفقياً مع الصف Y عند القيمة (1)

نضع في المربع الناتج عن عملية التقاطع وهو المربع رقم 1 القيمة (1)

الحد الثاني $(x'y')$ مكوّن من تقاطع العمود X عند القيمة (0) أفقياً مع الصف Y عند القيمة (0)

نضع في المربع الناتج عن عملية التقاطع وهو المربع رقم 0 القيمة (1)

وبالتالي نكون قد انتهينا من الخطوة الأولى وهي تعبئة المربعات التي تمثل الدالة بالقيمة (1)

والآن ننتقل للخطوة الثانية وهي اختيار المربعات ومن ثم تبسيط الدالة :

لدينا المربعين 0 و 1 يحتويان على القيمة (1) وبما أنهما بجانب بعضهما نختارهما معاً لكي نبسط الدالة

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

أصبح الآن لدينا مستطيل مكوّن من المربعين 0 و 1

ولكي نتمكن من تبسيط الدالة ننظر أولاً ماذا يمثل هذا المستطيل بالنسبة للجانب العمودي(X)

وكما نعلم أن لـ (X) قيمين : 0 وتعني X' , 1 وتعني X

وعند التبسيط لابد أن تكون قيم (X) متشابهة وأقصد إما أن تكون كلها 0 أو كلها 1

أما إذا كانت مختلفة فإننا نشطب (X) ولا يكون له وجود في الحل تماماً مثل ماحدث مع (Y)

وذلك لأن المستطيل المكوِّن للدالة واقع تحت قيمتين مختلفتين للجانب الأفقي (Y)

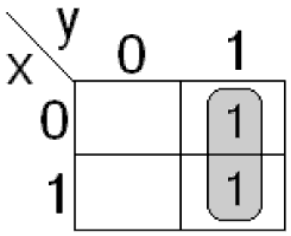
ناتج عملية التبسيط = X' وذلك لأن المستطيل المكوِّن للدالة واقع تحت قيمة متشابهة للجانب العمودي (X) وهي القيمة 0

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



$F(x,y) = y$

الشرح :

في هذا المثال قمنا أولاً بتعبئة المربعات التي تمثل الدالة بالقيمة (1)

ثم انتقلنا لعملية التبسيط وكما تلاحظ عزيزي القارئ أنه حدث العكس تماماً عن المثال السابق

حيث أن المتسطيل المكوِّن للدالة واقع تحت قيمة متشابهة للجانب الأفقي (Y) وهي القيمة 1

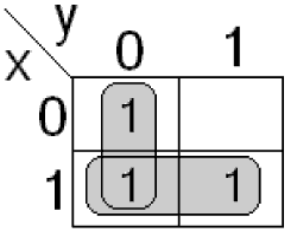
لذلك كان ناتج عملية التبسيط = Y

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



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

الشرح:

في هذا المثال قمنا أولاً بتعبئة المربعات التي تمثل الدالة بالقيمة (1)

ثم انتقلنا لتبسيط الدالة وكما تلاحظ أنه لدينا 3 مربعات تحتوي على القيمة (1) وهي 0 و 2 و 3

وعند عملية تكوين أول مستطيل ننظر أولاً إلى الـ 1 البعيد

وكما تلاحظ أن الـ 1 الواقع في المربع رقم 3 تجد نفسك مجبراً على اختياره مع الـ 1 الواقع في المربع رقم 2

وكذلك الحال ينطبق مع الـ 1 الواقع في المربع رقم 0

أما لو نظرت إلى الـ 1 الواقع في المربع رقم 2 تستطيع اختياره إما مع المربع رقم 0 أو المربع رقم 3

في خريطة متغييرين موضوع اختيار المربعات سهل أما في خريطة 3 متغيرات و4 متغيرات

سوف يصبح الموضوع أكثر تعقيداً لذلك وجب التنويه عن هذه النقطة لأهميتها

هنالك نقطة أخرى مهمة وهي أنك تستطيع استخدام المربع أكثر من مرة لتكوين مستطيل الدالة إذا دعت الحاجة

لاستخدامه أكثر من مرة كما فعلنا مع المربع رقم 2

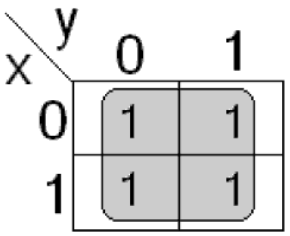
بالتالي نكون قد كوّنا المستطيلات التي تمثل الدالة ونكمل باقي الحل كما تعلمنا من الأمثلة السابقة

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



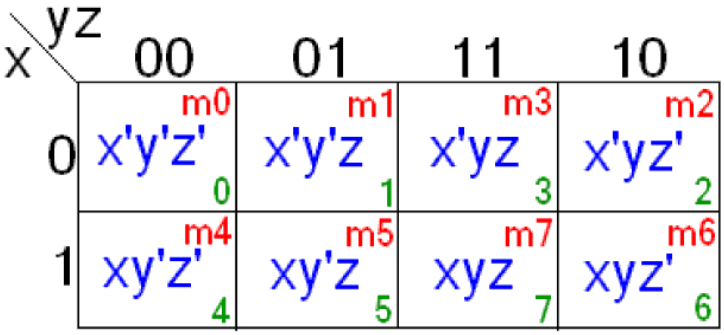
$F(x,y) = 1$

الشرح :

في هذا المثال عند تبسيطنا لهذه الدالة تمكنا من اختيار 4 مربعات معاً

وإذا أخذنا جميع المربعات المكوّنة للخريطة يكون ناتج عملية التبسيط = 1

3-3 خريطة 3 متغيرات 3 Variables Map :



الشكل السابق يوضح الشكل العام لـ Karnaugh Map لـ 3 متغيرات

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

توجد ملاحظات مهمة وهي :

1- 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 أن الترتيب غير تسلسلي

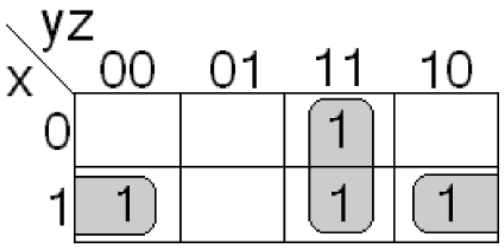
2- شكل الخريطة ليس مستطيلاً وإنما يشبه الأسطوانة وأقصد أن المربع رقم 0 ملاصق للمربع رقم 2 وكذلك المربع رقم 4 مع المربع رقم 6 وهذا يعني أنه لو كان لدينا المربعان 0 و 2 يحتويان على القيمة (1) نستطيع أخذهما معاً لتكوين مستطيل وذلك لأنهما بجانب بعضهما وكذلك الحال بالنسبة للمربعين 4 و 6

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



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

الشرح :

في هذا المثال تغيرت صيغة السؤال حيث أنه أعطانا المربعات التي تحتوي على القيمة (1) بينما في السابق كان يعطينا حدود الدالة , وبلا شك فإن الصيغة الجديدة أسهل من السابقة

قمنا أولاً بتعبئة المربعات بالقيمة (1) ومن ثم تكوين المستطيلات
وتبقى كتابة الحدود الناتجة عن عملية التبسيط

نأخذ المستطيل الأول المكوّن من المربعين 4,6 وننظر أولاً ماذا يمثل بالنسبة للجانب العمودي (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 (Y,Z) , ونعامل كلاً منهما على حدا

ننظر أولاً ماذا يمثل بالنسبة لـ (Y)
ونلاحظ أن قيمة (Y) اختلفت مرة 0 ومرة 1 لذلك نشطب (Y) ولا نكتبه في الناتج لأن قيمته مختلفة
ثم ننظر للمتغير(Z) ونلاحظ أن قيمة (Z) متشابهة حيث أنها = 0 لذلك نكتب في الناتج Z'

وبذلك يكون قد إنتهينا من أول مستطيل وإستنتجنا أول حد وهو XZ'

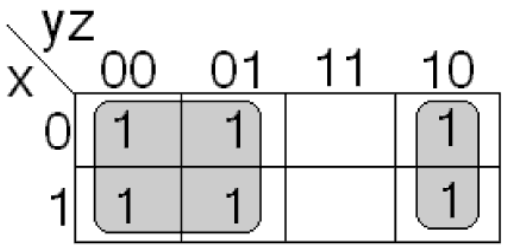
ثم ناخذ المستطيل الثاني المكوّن من المربعين 3,7 ونعمل معه مثل ما عملنا مع المستطيل السابق تماماً
وسوف ينتج لنا الحد الثاني وهو YZ
ثم بعد ذلك نكتب ناتج عملية التبسيط ونربط بين الحدين بعلامة +
ناتج عملية التبسيط = XZ' + YZ

مثال :

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل :



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

مثال :

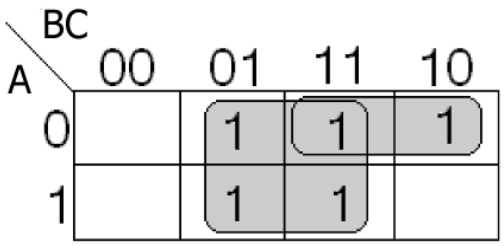
Given the Boolean function :

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

1- Express it in Sum Of Minterms

2- Find the minimal Sum Of Products expression

الحل :



1- $F(A,B,C) = \Sigma(1,2,3,5,7)$

2- $F(A,B,C) = C + A'B$

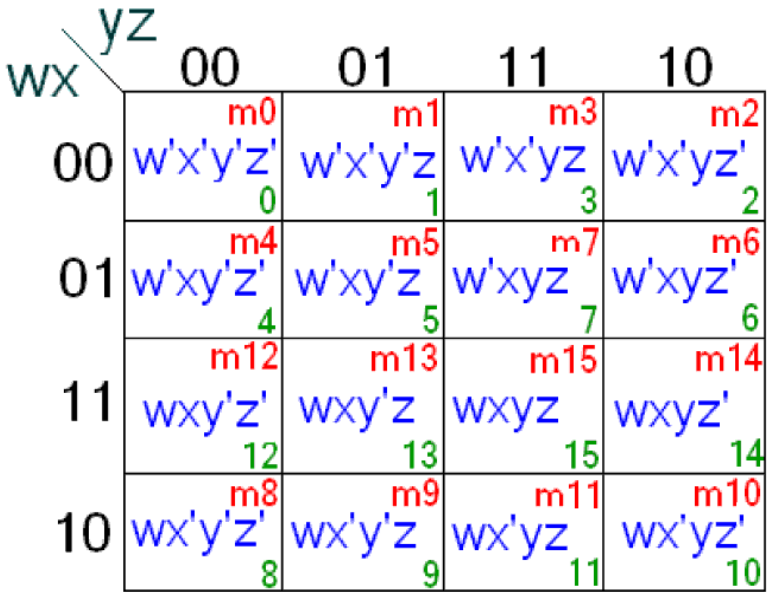
الشرح :

في هذا المثال قمنا أولاً بتعبئة المربعات التي تمثل الدالة بالقيمة (1)

ثم أوجدنا Sum Of Products ويمثل المربعات التي تحتوي على القيمة (1)

وأخيراً قمنا بتكوين المربعات والمستطيلات التي تمثل الدالة ومن ثم تبسيطها

3-4 خريطة 4 متغيرات 4 Variables Map :



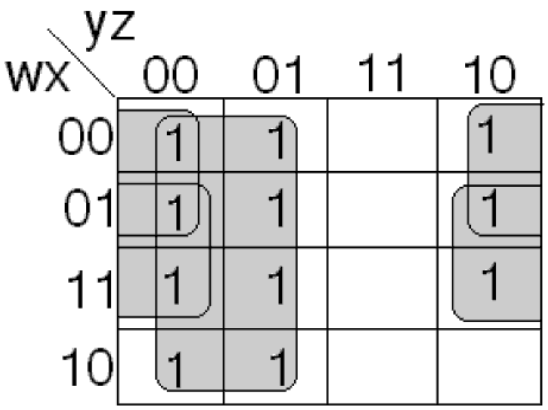
الشكل السابق يوضح الشكل العام لـ Karnaugh Map لـ 4 متغيرات
الجديد في الأمر زيادة متغير جديد وهو (W) ليمثل مع (X) الجانب العمودي
ترتيب المربعات غير متسلسل
شكل الخريطة ليس مربع وإنما يشبه المكعب
وأقصد أن المربعات متصلة مع بعضها من الجانبين ومن الأعلى والأسفل
ولو أخذنا على سبيل المثال المربع رقم 0 نجد أنه مجاور للمربعات 2 و 8 و 10

مثال:

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:



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

الشرح:

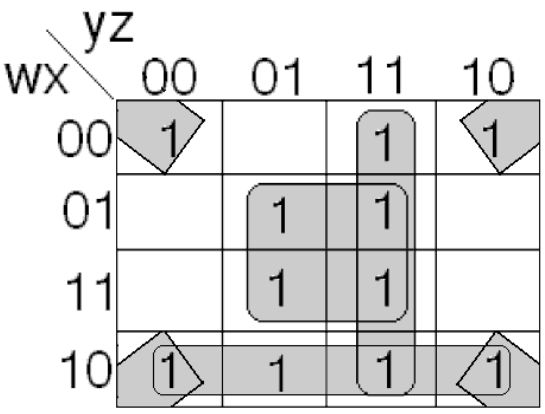
لاتختلف طريقة تبسيط 4 متغيرات عن تبسيط 3 متغيرات و متغيرين
لاحظ فقط في هذا المثال أننا تمكنا من اختيار 8 مربعات معاً

مثال:

بسط الدالة المنطقية التالية Simplify the following Boolean function :

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

الحل:



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

لاحظ أنه يمكننا اختيار المربعات الموجودة في الأركان وذلك لأنها مربعات متجاورة

3-5 Don't Care Conditions :

مثال :

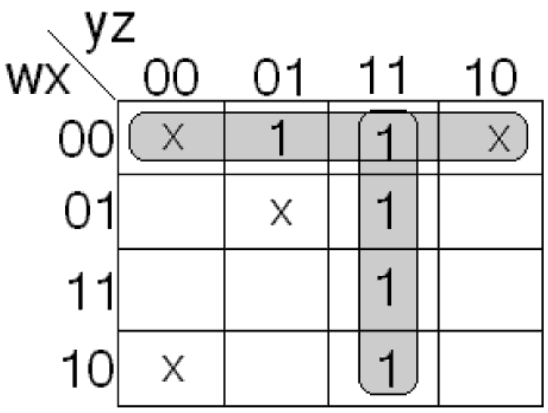
بسط الدالة المنطقية التالية Simplify the following Boolean function :

$F(w,x,y,z) = \Sigma(1,3,7,11,15)$

Which has the Don't Care conditions

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

الحل :



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

الشرح :

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(x)بالكامل, ولكن إذا احتجنا لاستخدامها نستخدمها عكس المربعات التي تحتوي على قيمة (1) فإنه يتوجب عليك تغطيتها بالكامل وإلا فإن حلك خاطئ

قمنا أولاً بتعبئة المربعات بقيم (1) ومن ثم تعبئتها بقيم (x)

ثم قمنا بتكوين المستطيلات التي تعطينا قيمة الدالة , نتج لدينا مستطيلين

المستطيل الأول المستطيل العمودي المكوّن من المربعات 3,7,15,11

وهذا المستطيل طريقة تبسيطه مثل الأمثلة السابقة ولن نتطرق إليه

المستطيل الآخر المستطيل الأفقي المكوّن 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 كان المستطيل أكبر كان الحل أكثر اختصاراً

عند وجود Don't Care يتوجب عليك استخدامها إذا احتجت إليها ولا يجب عليك تغطيتها بالكامل

وكما تلاحظ عزيزي القارئ تجاهلنا Don't Care الموجودة في المربعين 5,8 وذلك لعدم حاجتنا إليها

بعد ذلك نكمل باقي خطوات الحل كما تعلمنا سابقاً

3-6 حل أهم تمارين الفصل :

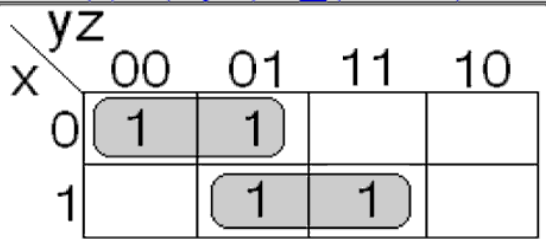
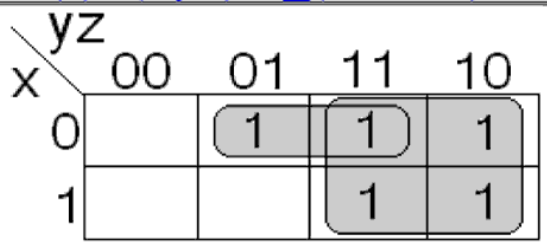
: Simplify the following Boolean function, using 3-variable maps /Q2

(a) F(x,y,z) =∑(0,1,5,7) , (b) F(x,y,z) = ∑(1,2,3,6,7)

المطلوب :

تبسيط الدوال المنطقية باستخدام خريطة 3 متغيرات

الحل :

(a) F(x,y,z) =∑(0,1,5,7)	(b) F(x,y,z) = ∑(1,2,3,6,7)
 F = x'y' + xz	 F = x'z + y

: Simplify the following Boolean function, using x maps /Q4

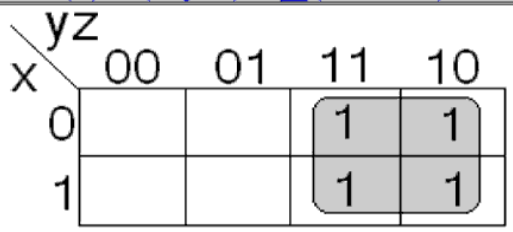
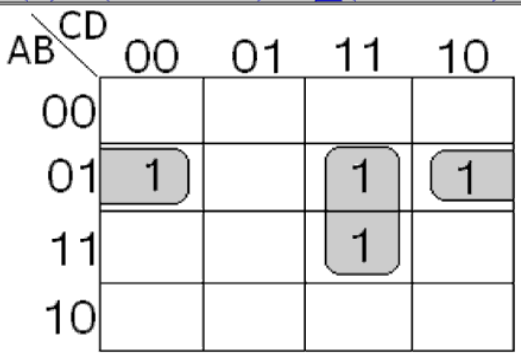
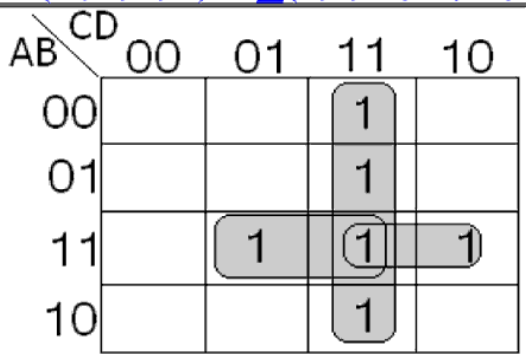
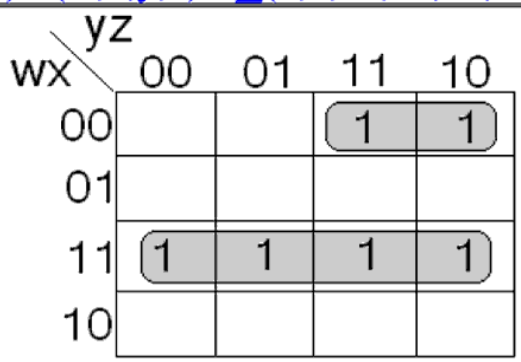
(a) F(x,y,z) =∑(2,3,6,7) , (b) F(A,B,C,D) = ∑(4,6,7,15)

(c) F(A,B,C,D) = ∑(3,7,11,13,14,15) , (d) F(w,x,y,z) =∑(2,3,12,13,14,15)

المطلوب :

تبسيط الدوال المنطقية باستخدام الخريطة المناسبة

الحل :

(a) F(x,y,z) =∑(2,3,6,7)	(b) F(A,B,C,D) = ∑(4,6,7,15)
 F = y	 F = A'BD' + BCD
(c) F(A,B,C,D) = ∑(3,7,11,13,14,15)	(d) F(w,x,y,z) =∑(2,3,12,13,14,15)
 F = ABD + ABC + CD	 F = w'x'y + wx

: Simplify the following Boolean expressions, using 4-variable maps /Q6

(a) A'B'C'D' + AC'D' + B'CD' + A'BCD + BC'D

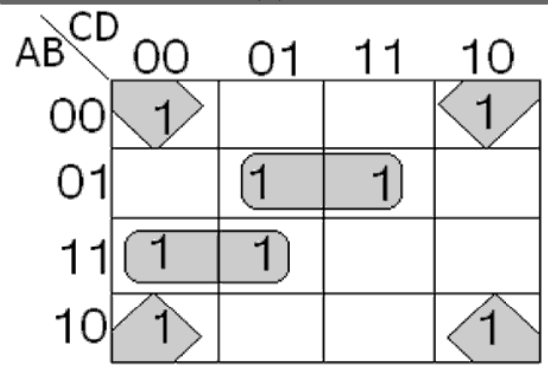
(b) x'z + w'xy' + w(x'y + xy')

المطلوب :

تبسيط التعبيرات المنطقية باستخدام خريطة 4 متغيرات

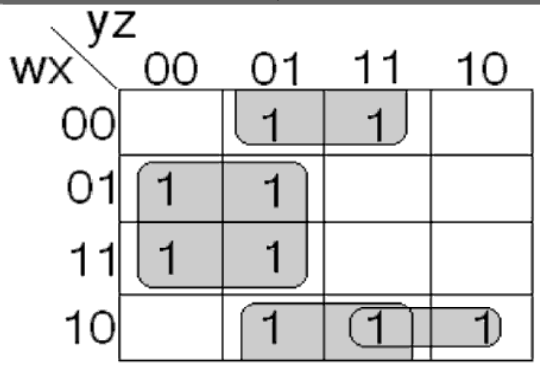
الحل :

(a)



F = B'D' + A'BD + ABC'

(b)



F = x'z + xy' + wx'y

: Simplify the following Boolean function F, together with Don't Care conditions d /Q15

And then express the simplified function in Sum Of Minterms

(a) F(x,y,z) =∑(0,1,2,4,5) d(x,y,z) =∑(3,6,7)

(b) F(A,B,C,D) =∑(1,3,5,7,9,15) d(A,B,C,D) =∑(4,6,12,13)

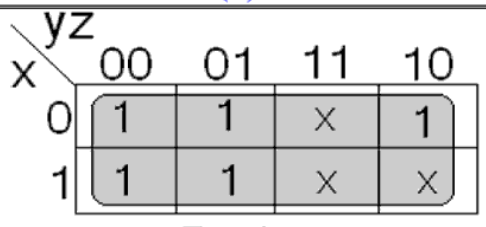
(c) F(A,B,C,D) =∑(0,6,8,13,14) d(A,B,C,D) =∑(2,4,10)

المطلوب :

تبسيط الدوال المنطقية باستخدام الخريطة المناسبة مع مراعاة وجود Don't Care

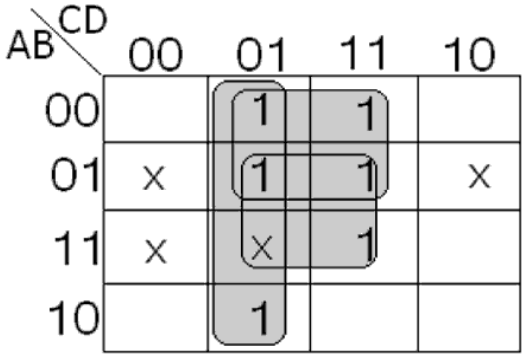
الحل :

(a)



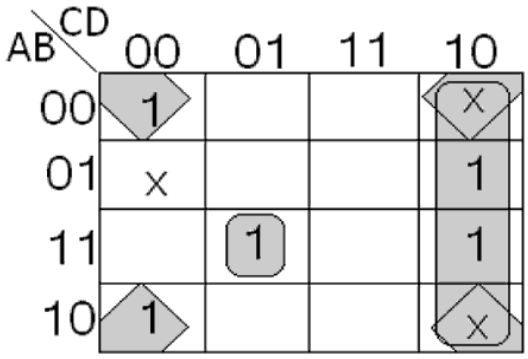
F = 1

(b)



F = C'D + A'D + BD

(c)



F = B'D' + CD' + ABC'D

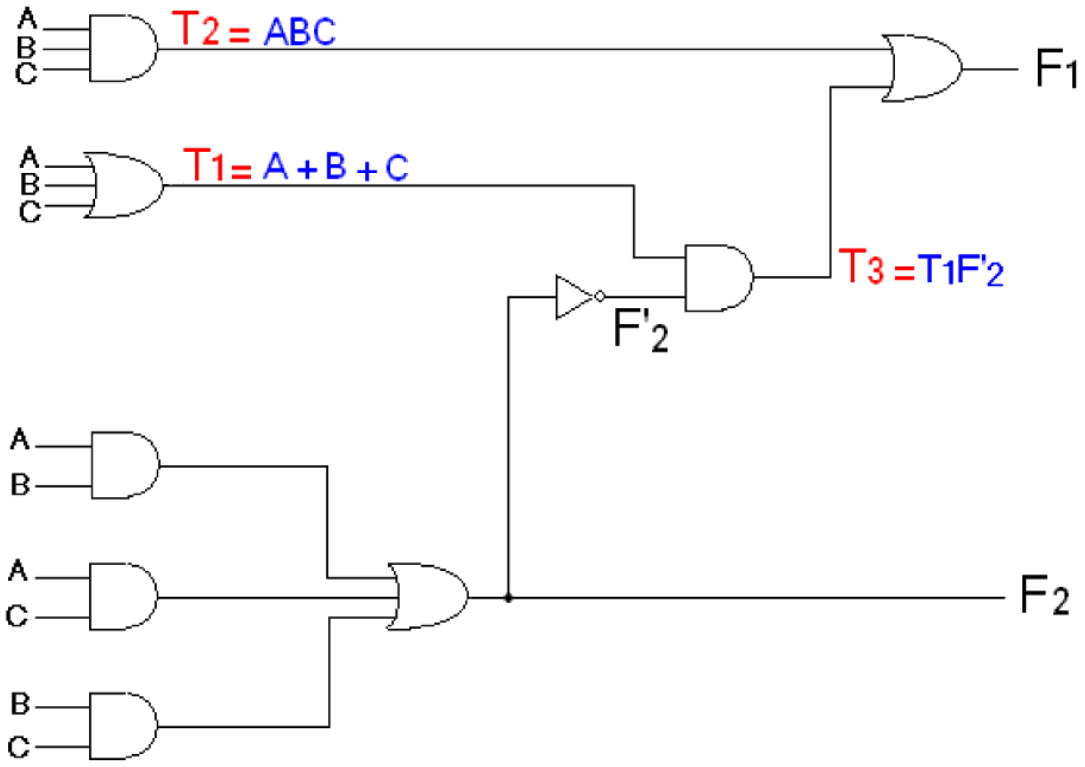
الفصل الرابع

Combinational Logic

4-1 مقدمة :

في هذا الفصل سوف نتحدث عن تحليل الدوال (Analysis) وإيجاد مخرجاتها
وكذلك سوف نتحدث عن تصميم الدوال وحل مسائل التصميم (Design)
كما سوف نتعرف على بعض الدوائر وهي :
Multiplexer ، Decoder ، Full Adder ، Half Adder
وسوف نناقشها من حيث :
تعريفها وأشكالها وأنواعها أقسامها وعملها و حل المسائل بواسطتها

4-2 إجراء التحليل Analysis Procedure :



الحل :

المطلوب إيجاد مدخلات وحدود الدالتين F1 و F2
وكما تلاحظ عزيزي القارئ أن عدد المدخلات كثير وربما أنك تنسى كتابة حد أو متغير أثناء قيامك بكتابة حدود كل دالة ولتفادي هذه المشكلة نقوم بتجزئة الدائرة إلى عدة أجزاء ومن ثم نحدد نوجد مخرجات كل جزء على حدا
بعد ذلك نقوم بكتابة مخرجات الدالتين المطلوبة كما في التالي :

$$T1 = A + B + C$$
$$T2 = ABC$$
$$F2 = AB + AC + BC$$
$$T3 = T1F'2$$
$$= (A+B+C)(AB+AC+BC)'$$
$$= (A+B+C)(A'+B')(A'+C')(B'+C')$$
$$F1 = T2 + T3$$

4-3 إجراء التصميم Design Procedure :

خطوات إجراء التصميم (Design Procedure) :

1- تحديد مدخلات ومخرجات الدائرة
2- إنشاء Truth Table لمدخلات ومخرجات الدائرة
3- تبسيط مخرجات الدائرة (Simplify)
4- رسم Diagram لمخرجات الدائرة بعد تبسيطها

مثال :

Design a combinational circuit that converts the Binary Coded Decimal (BCD)
The excess-3 code for the Decimal digit

الحل :

1و2- تحديد المدخلات والمخرجات وإنشاء Truth Table :

المطلوب تصميم دائرة تحول التشفير الثنائي إلى العشري (BCD) وذلك بزيادة 3 على العدد العشري
نقوم أولاً بتعبئة أعمدة المدخلات (Inputs) وهي الأعمدة (A,B,C,D) بالطريقة المعتادة
ننتقل الآن لتعبئة أعمدة المخرجات (Outputs) وهي الأعمدة (W,X,Y,Z)
ولتعبئتها نقوم بجمع القيمة (3) على كل صف من المدخلات
وينتج لنا في المقابل أعمدة المخرجات
على سبيل المثال الصف الأول :

$$(0000)_2 = (0)_{10} \longrightarrow (0011)_2 = (3)_{10}$$

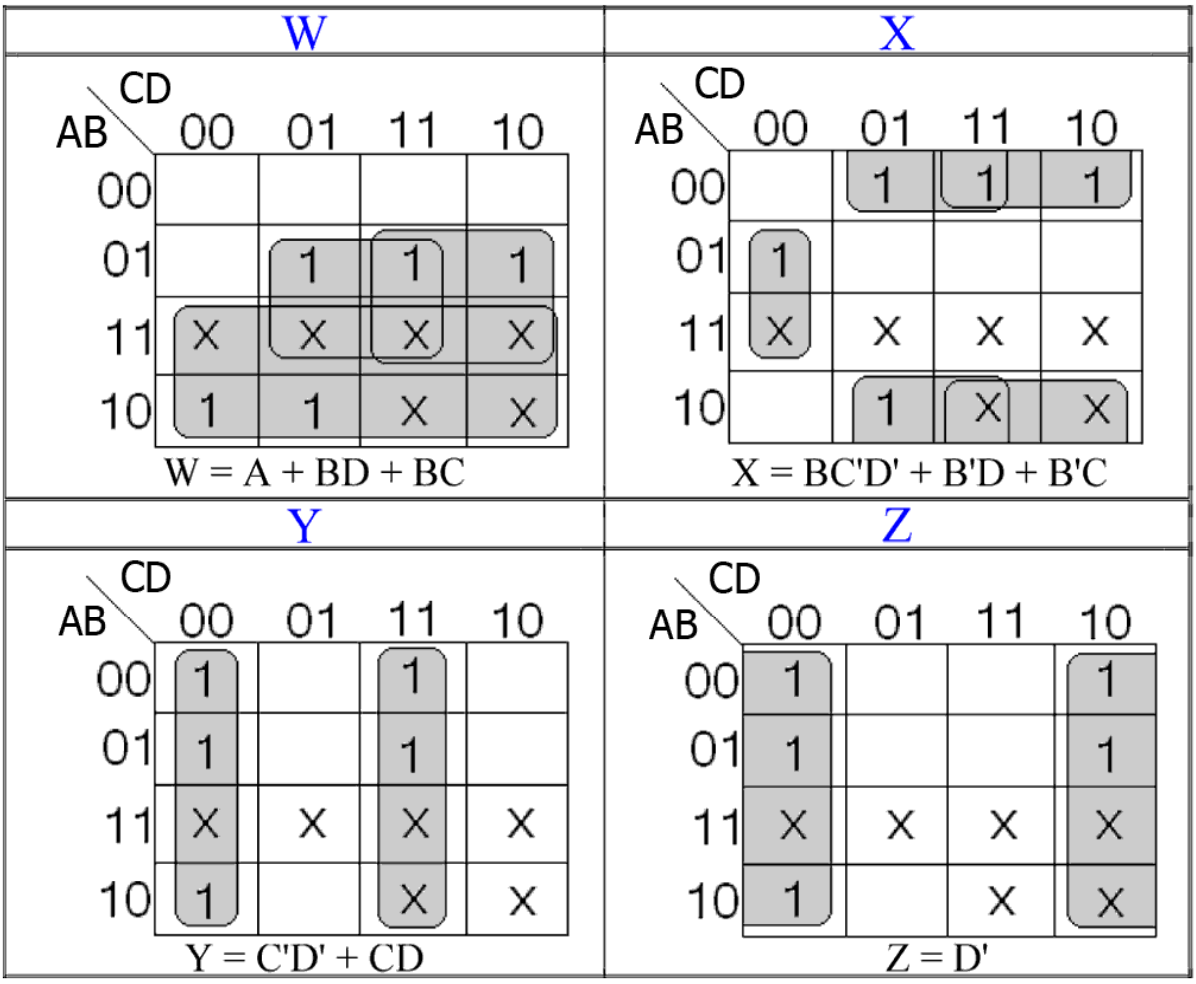
وكذلك الصف الخامس :

$$(0100)_2 = (4)_{10} \longrightarrow (0111)_2 = (7)_{10}$$

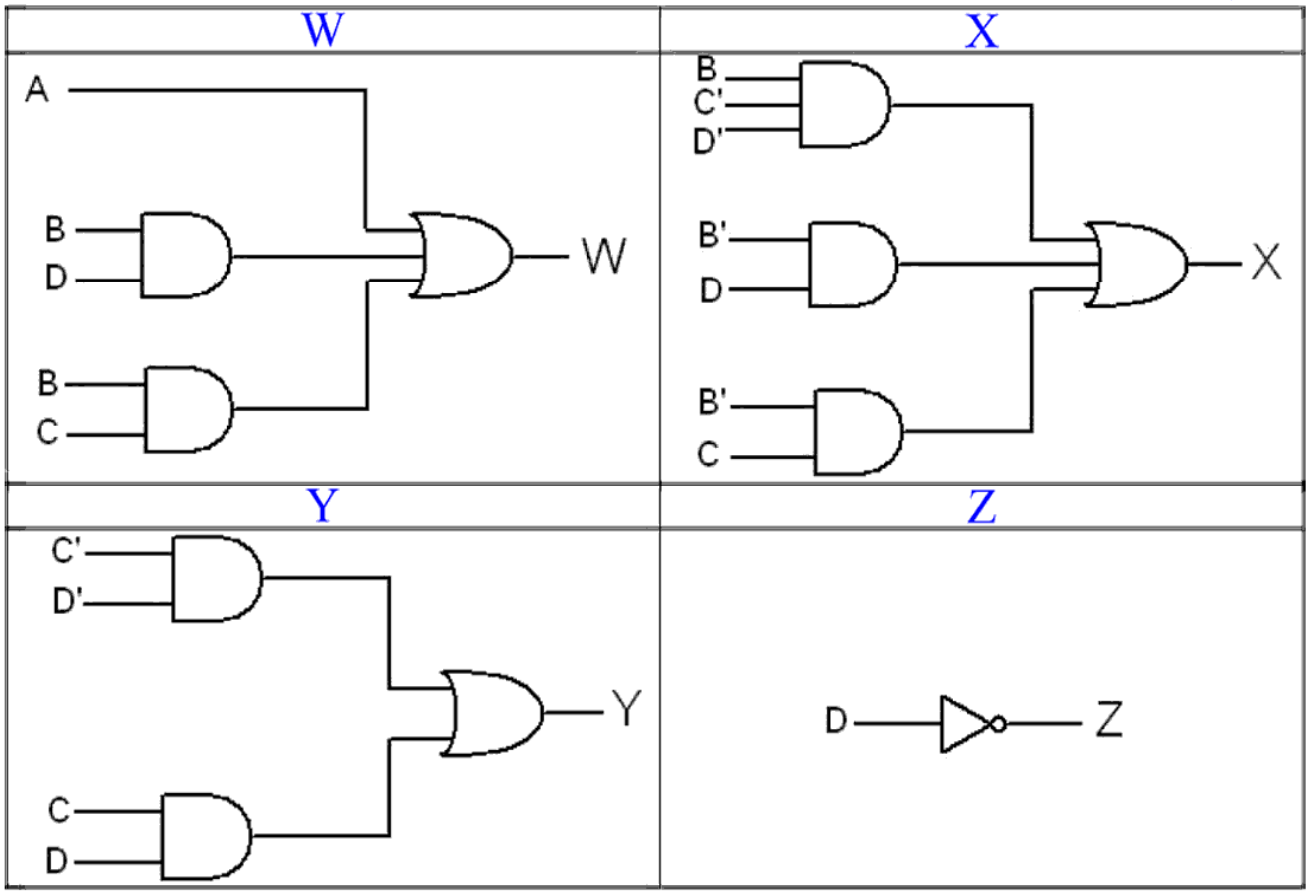
عزيزي القارئ لاحظ أن المدخلات تمثل أعداد عشرية
فعندما نكون قيمة المدخلات أكبر من العدد $(9)_{10} = (1001)_2$ والذي يمثل آخر عدد في النظام العشري
يكون انتقالها في المخرجات إلى Don't Care كما فعلنا من الصف الحادي عشر إلى الصف الخامس عشر

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

3- تبسيط مخرجات الدائرة (Simplify) :



4- رسم Diagram لمخرجات الدائرة بعد تبسيطها :



مثال :

: Design a 2 - bit binary Multiplier

الحل :

1و2- تحديد المدخلات والمخرجات وإنشاء Truth Table :

المطلوب تصميم دالة Multiplier , عملها تقوم بضرب قيم المدخلات وتضع الناتج في المخرجات
حيث تعتبر العمودين (A1 , A0) عدد واحد وتقوم بضربه بالعدد الآخر المكوّن من العمودين (B1 , B0)
على سبيل المثال الصف الثالث :

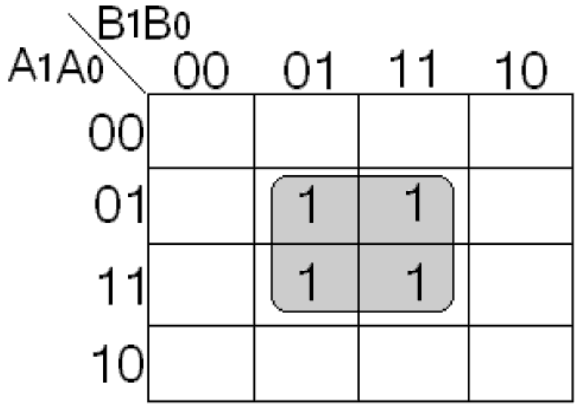
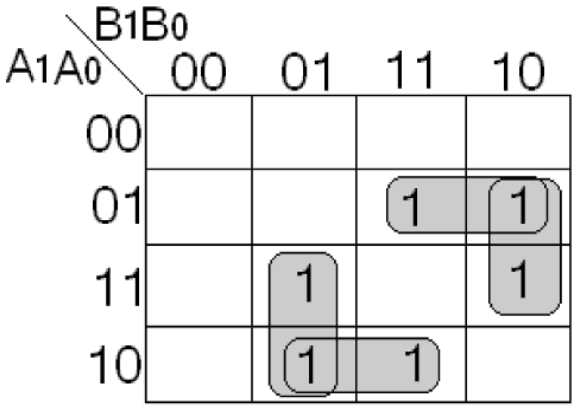
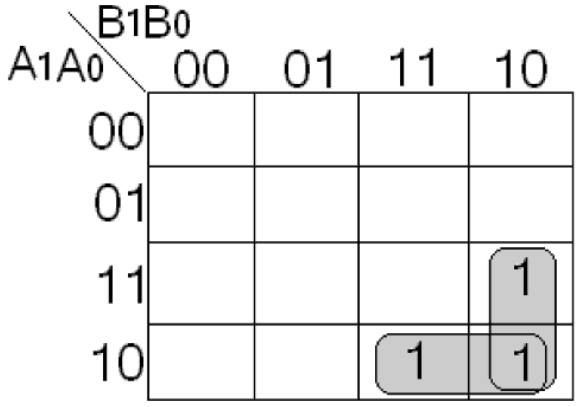
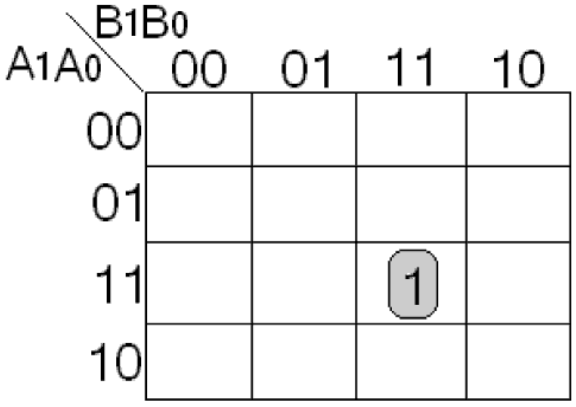
$$(00)_2 = (0)_{10} * (10)_2 = (2)_{10} \longrightarrow (0000)_2 = (0)_{10}$$

وكذلك الصف الثامن :

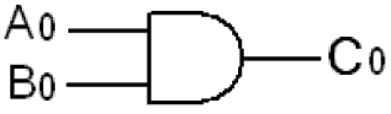
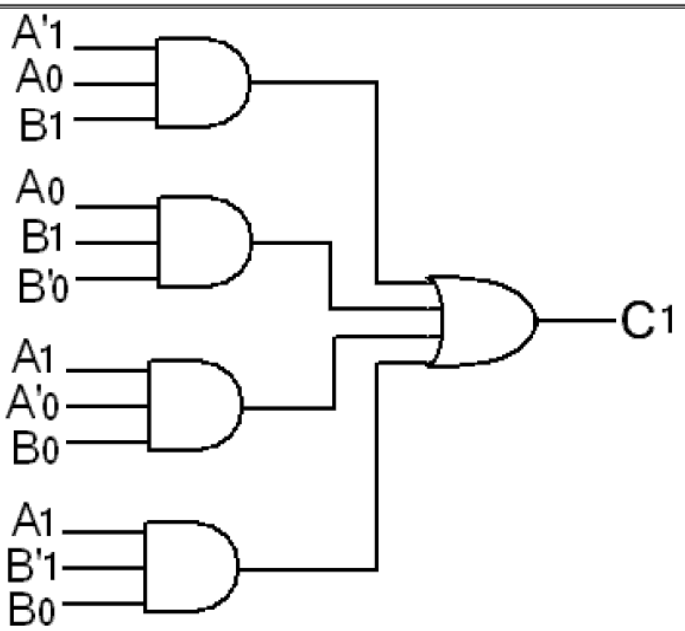
$$(01)_2 = (1)_{10} * (11)_2 = (3)_{10} \longrightarrow (0011)_2 = (3)_{10}$$

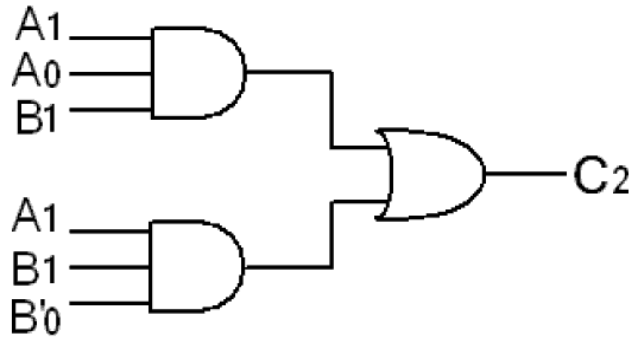
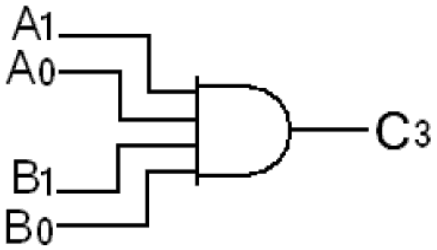
InPuts				OutPuts			
A_1	A_0	B_1	B_0	C_3	C_2	C_1	C_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	0	0	0	0
0	1	0	0	0	0	0	0
0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	1
1	0	0	0	0	0	0	0
1	0	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	0	1	1	0	1	1	0
1	1	0	0	0	0	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	1	1	0
1	1	1	1	1	0	0	1

: Simplify -3

C_0	C_1
	
$C_0 = A_0B_0$	$C_1 = A'_1A_0B_1 + A_0B_1B'_0 + A_1A'_0B_0 + A_1B'_1B_0$
C_2	C_3
	
$C_2 = A_1A_0B_1 + A_1B_1B'_0$	$C_3 = A_1A_0B_1B_0$

: Diagram -4

C_0	C_1
	

C_2	C_3
	

مثال :

: Design a 2 - bit Magnitude Comparator

الحل :

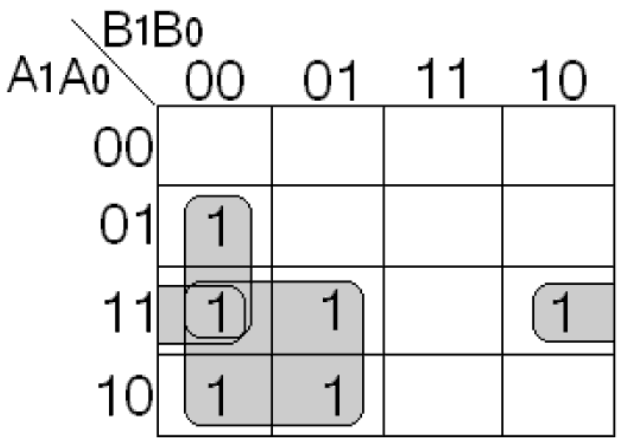
1و2- تحديد المدخلات والمخرجات وإنشاء Truth Table :

المطلوب تصميم دائرة Magnitude Comparator , وعملها تقارن بين قيمة المدخلات
حيث تعتبر العمودين (A1,A0) عدد واحد وتقارن قيمته مع العدد الآخر المكوّن من العمودين (B1,B0)
هل هي أكبر أم أصغر أم مساوية وتضع النتيجة في المخرجات

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

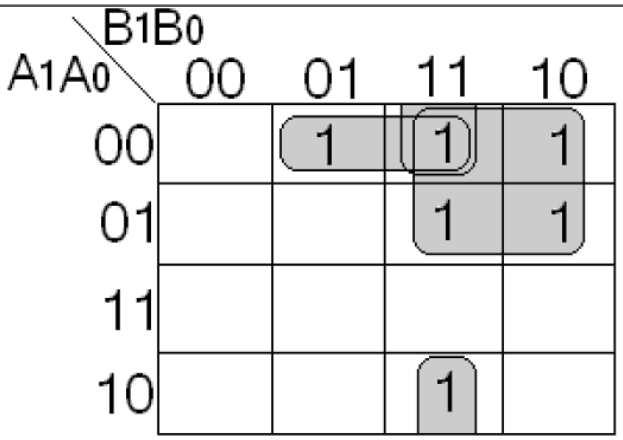
: Simplify -3

X



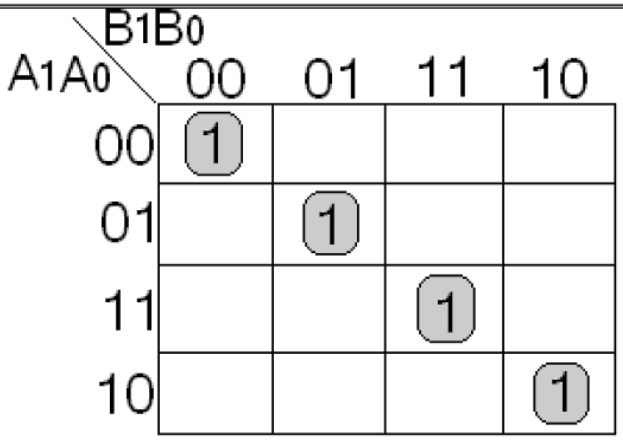
$X = A_1B'_1 + A_1A_0B'_0 + A_0B'_1B_0$

Y



$Y = A'_1B_1 + A'_0B_1B_0 + A'_1A'_0B_0$

Z

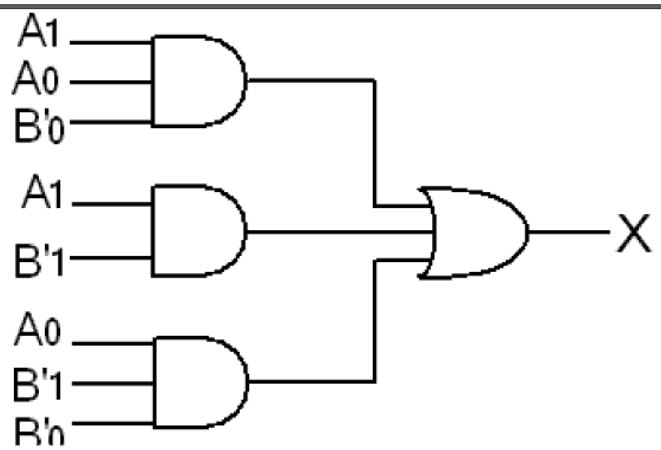


$Z = A'_1A'_0B'_1B'_0 + A'_1A_0B'_1B_0 + A_1A_0B_1B_0 + A_1A'_0B_1B'_0$ (A حل)

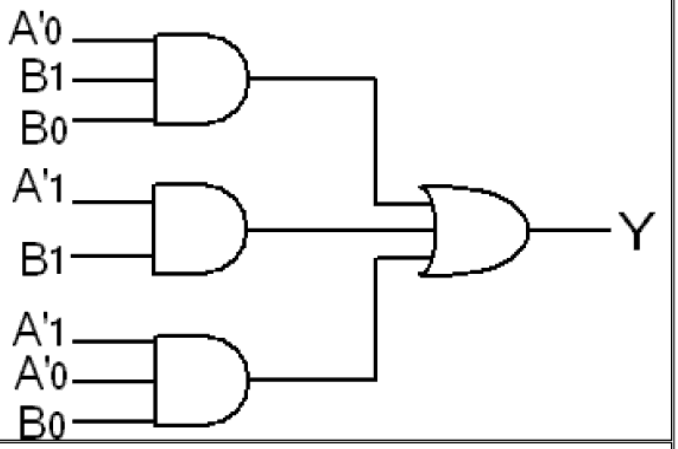
$= A_1 \oplus A_0 \oplus B_1 \oplus B_0$ (B حل)

: Diagram -4

X

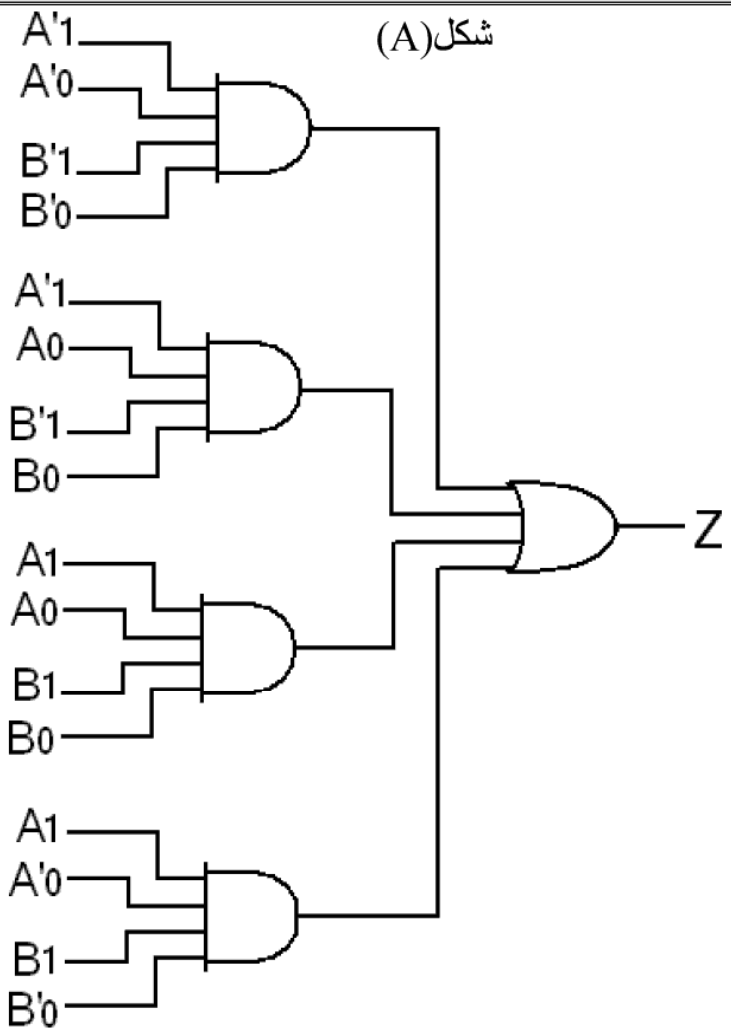


Y

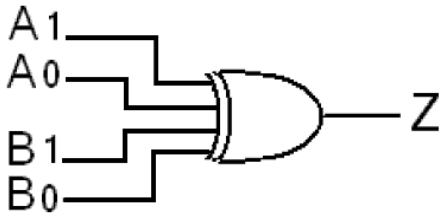


Z

شكل(A)



شكل(B)



: Half Adder & Full Adder 4-4

تستخدم هذه الدالتين للقيام بعمليتي الجمع والطرح

: Half Adder -1

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 العمود من الجدول نضع قيمة جمع كل صف من المدخلات أي X+Y

(C) إختصار لكلمة (Carry) ، وإذا وجد Carry نضع في هذه الخانة 1 وإذا لم يوجد نضع 0 ويوجد Carry إذا كان حاصل جمع X+Y أكبر من 1 كما في آخر صف في الجدول

Algebraic Function

(A)

$S = xy' + x'y$

$C = xy$

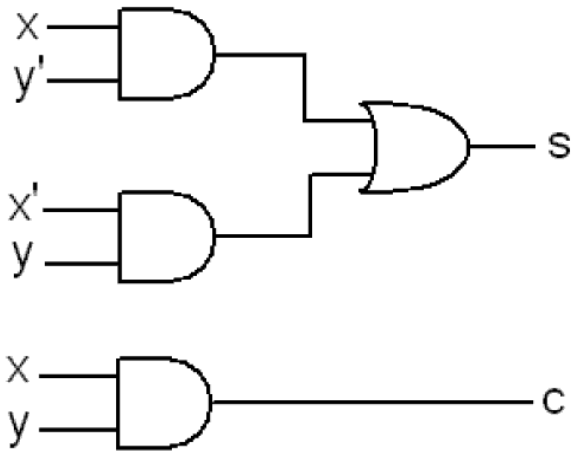
(B)

$S = x \oplus y$

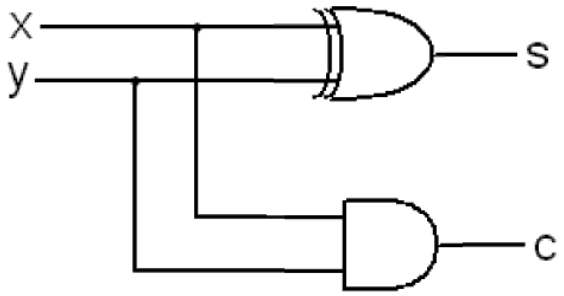
$C = xy$

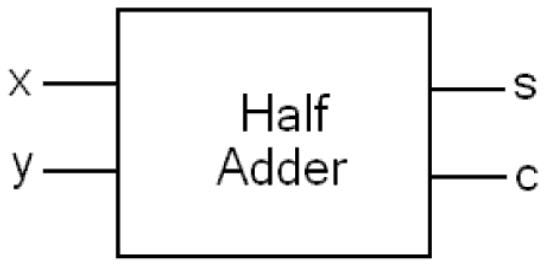
Graphic Symbol

شكل(A)



شكل(B)



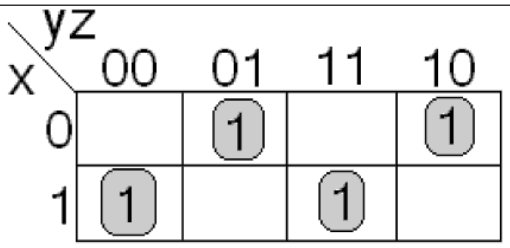


Truth Table

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

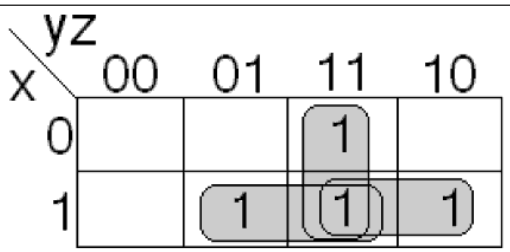
(Ci) إختصار لكلمة (Carry in) ، (Co) إختصار لكلمة (Carry out)

Algebraic Function



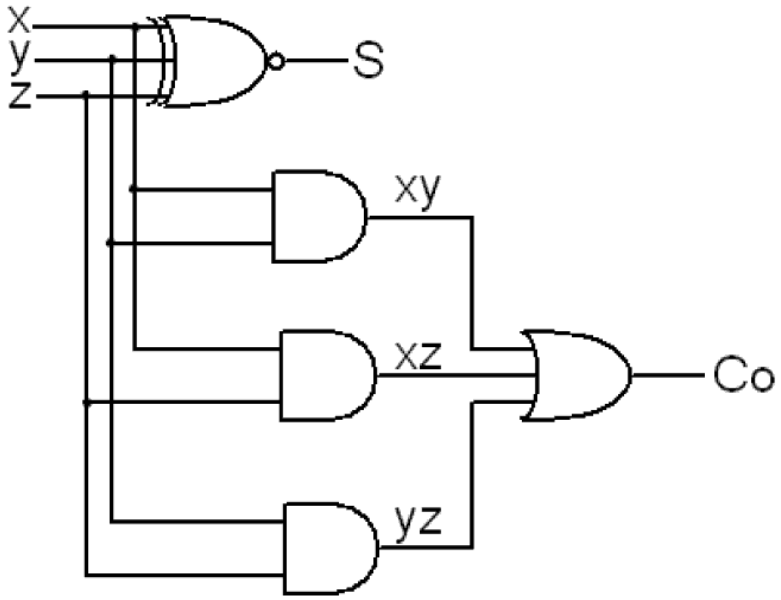
$S = x'y'z + x'yz' + xy'z' + xyz$

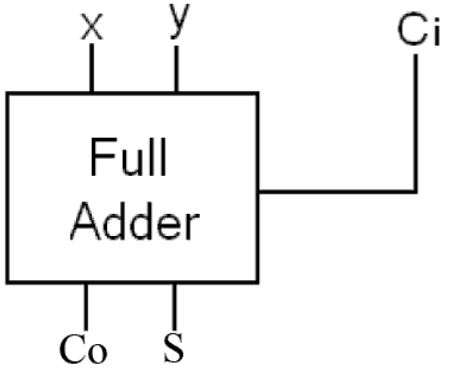
$= x \oplus y \oplus z$



$C = xy + xz + yz$

Graphic Symbol



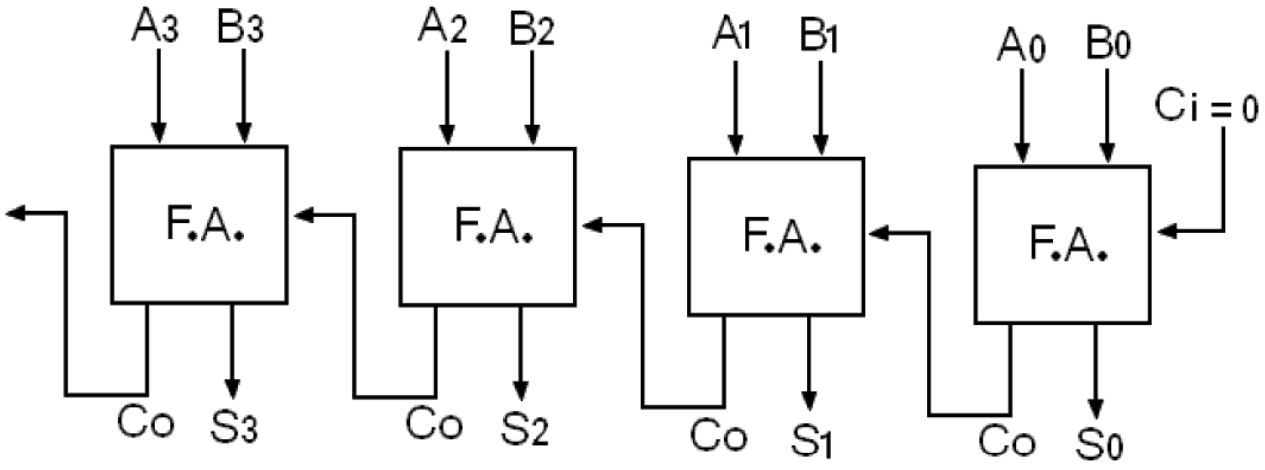


مثال :

: Design a 4 - bit Full Adder

الحل :

المطلوب تصميم دائرة Full Adder تستقبل 4-bit لتقوم بعملية الجمع



شكل مبسط يوضح ما الذي يحدث :

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

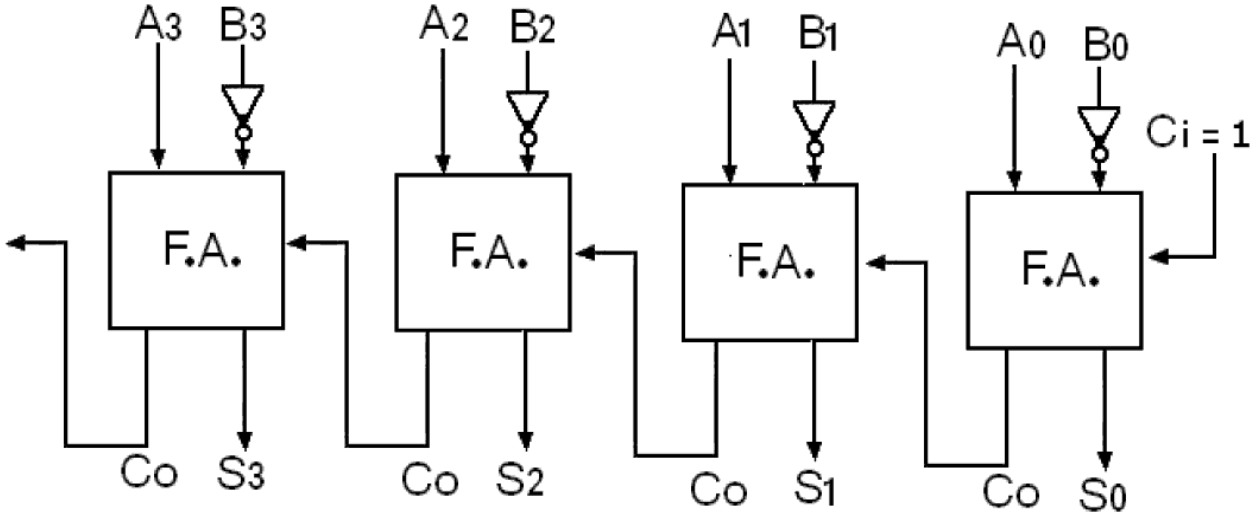
مثال :

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

الحل :

المطلوب تصميم دائرة Full Adder تستقبل 4-bit لتقوم بعملية الطرح

وللقيام بعملية الطرح نحتاج إلى Two's Complement



العملية التالية توضح ما الذي يحدث :

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

: Decoder 4-5

A Decoder has n inputs and 2^n outputs

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

على سبيل المثال

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

كذلك لو كان عدد المدخلات = 3 فإن عدد المخرجات = 2^3 = 8

بالتالي فإن نوع وشكل Decoder يعتمد على عدد مدخلاته

: Decoder 2 * 4 (Has 2 inputs and 4 outputs) -1

Truth Table

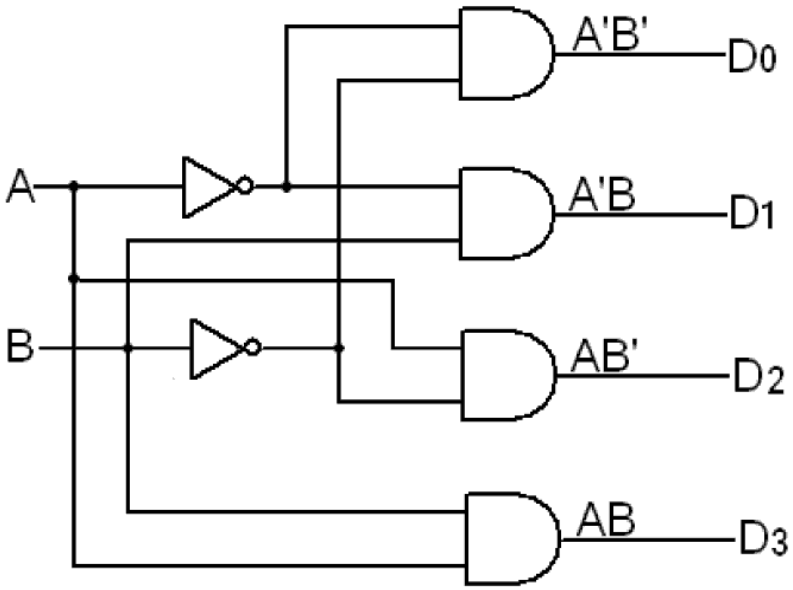
InPuts		OutPuts			
A	B	D_3	D_2	D_1	D_0
0	0	0	0	0	1
0	1	0	0	1	0
1	0	0	1	0	0
1	1	1	0	0	0

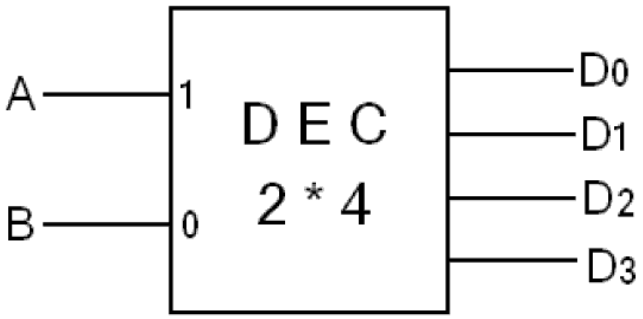
كل صف من المدخلات(A,B) يكوّن رقم

إذا كان الرقم المكوّن = 0 نضع في الخانة D_0 القيمة (1) وباقي الخانات القيمة (0)

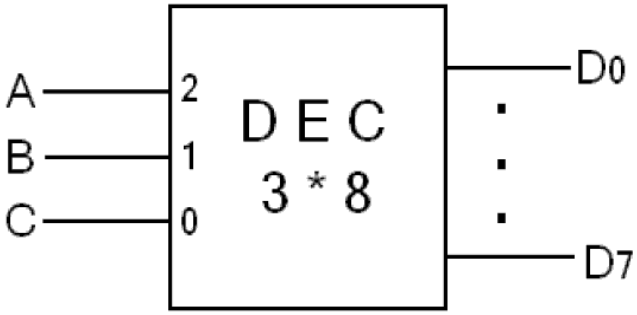
وإذا كان الرقم المكوّن=1 نضع في الخانة D_1 القيمة (1) , وهكذا

Graphic Symbol

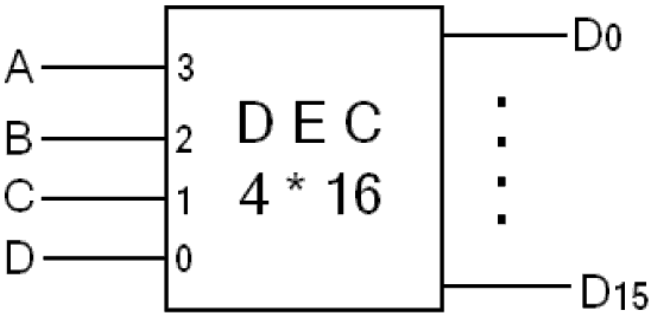




: Decoder 3 * 8 -2



: Decoder 4 * 16 -3



: Decoder With Enabel 4-6

يعتبر Enable مفتاح لدائرة Decoder بمعنى :
إذا كانت قيمة Enabel = 0 فإن Decoder لن يعمل , أما إذا كانت قيمة Enabel = 1 فإن Decoder يعمل

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 تلاحظ عزيزي القارئ أن العمود (E) هو Enabel

عندما كانت قيمة Enabel = 0 وذلك في أول 4 صفوف
كان Decoder لا يعمل وبالتالي كان يخرج القيمة (0) فقط
أما عندما أصبحت قيمة Enabel = 1 وذلك في آخر 4 صفوف
أصبح Decoder يعمل وظهرت نتيجة عمله 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 فقط اختصار للجدول الذي قبله

وكما تلاحظ على أول 4 صفوف من جدول الأول أنها تخرج قيمة واحدة وهي القيمة (0)
إذن وجود هذه الصفوف زائد
لذلك قمنا باختصار هذه الصفوف في صف واحد فقط وهو الصف الأول من الجدول الجديد
تلاحظ في هذا الصف أن : A=X , B=X
القيمة X في هذا الجدول ليست Don't Care
لأن كما تعلمنا أن Don't Care توجد في المخرجات فقط ولايمكن أن توجد في المدخلات
أذن X ليست Don't Care وإنما فقط رمز ويعني : X = {0,1}

بمعنى آخر أنه مهما كانت قيم العمودين A و B فإنه سوف تخرج قيمة واحدة وهي القيمة (0)

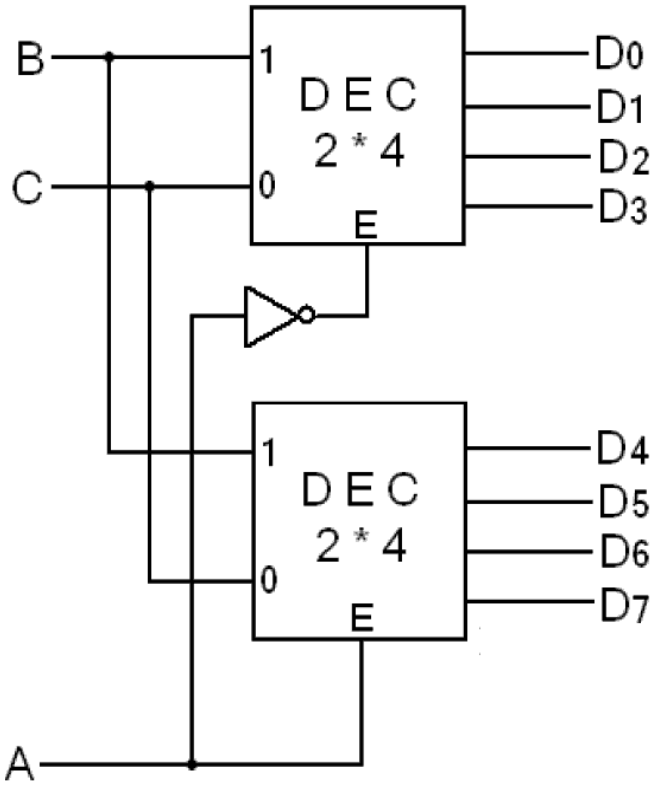
أذن الجدول الجديد فقط يختصر أول 4 صفوف من الجدول القديم في صف واحد
ويمكنك تجاهله تماماً وكتابة الجدول كاملاً كما في الجدول القديم أن أردت

مثال :

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

الحل :

المطلوب تصميم Decoder 3*8 باستخدام Decoder 2*4 مع Enable



الشرح :

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

الجدول السابق ليس من الحل وإنما يوضح الحل

تابع الشرح

كما تلاحظ عزيزي القارئ أن العمود (A) هو Enabel
عندما كانت قيمة A=0 :
كانت القيمة الداخلة إلى Decoder العلوي = 1 وذلك لوجود Inverter
وبالتالي فإنه كان يعمل وأخرج لنا أول 4 مخرجات وهي : D0,D1,D2,D3

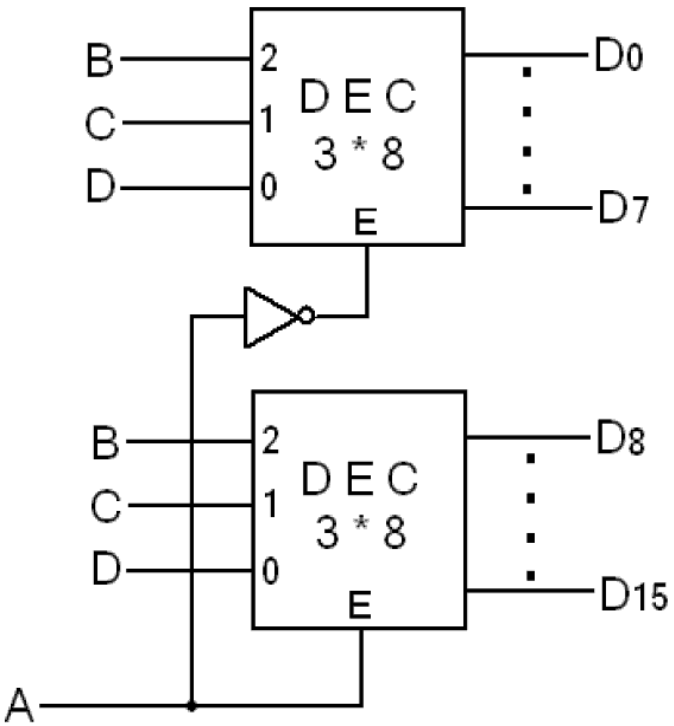
أما Decoder السفلي لم يكن يعمل وبالتالي أخرج لنا القيمة (0)

أما عندما أصبحت قيمة A=1 :
أصبحت القيمة الداخلة إلى Decoder العلوي = 0 وذلك لوجود Inverter
وبالتالي فإنه لم يكن يعمل وأخرج لنا القيمة (0)

أما Decoder السفلي فإنه كان يعمل وأخرج لنا آخر 4 مخرجات وهي : D4,D5,D6,D7

وبالتالي نكون قد حصلنا على Decoder 3*8 يخرج : D0 ⟶ D7
باستخدام Decoder 2*4 مع Enabel كما هو مطلوب

مثال :
: Desing a Decoder 4*16 using a Decoder 3*8 with Enable and additional gate
الحل :
المطلوب تصميم Decoder 4*16 باستخدام Decoder 3*8 مع Enable
طريقة الحل هي نفس الطريقة المتبعة لحل المثال السابق مع ملاحظة اختلاف المطلوب تصميمه



: Multiplexer 4-7

Multiplexer has 2^n inputs , 1outputs and n Selections

هذه الدالة مدخلاتها = 2^n ومخرجاتها = 1 وعدد n = Selections

على سبيل المثال :

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

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

: Multiplexer 2*1 -1

A Multiplexer has 2 inputs , 1 outputs and 1 selections

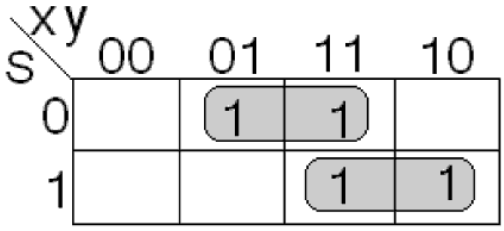
Truth Table

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

إذا كان S=0 فإن المخرجات هي قيمة Y , (Q = Y)

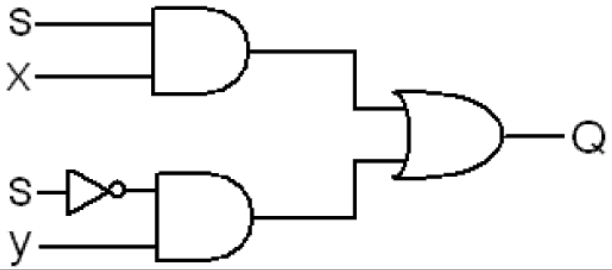
أما إذا كان S=1 فإن المخرجات هي قيمة X , (Q = X)

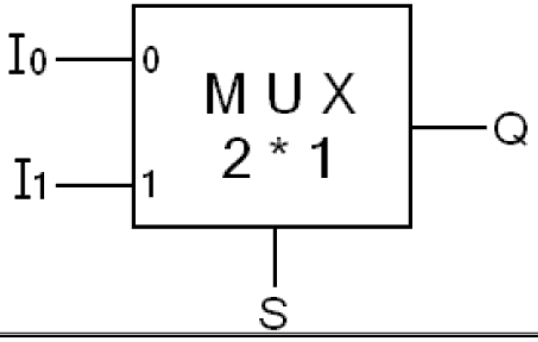
Algebraic Function



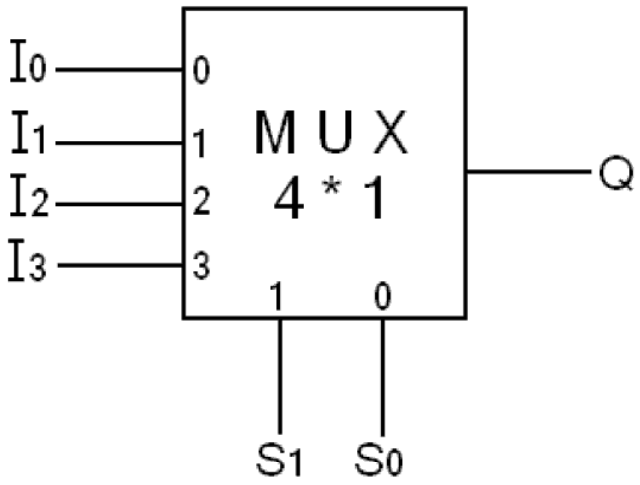
Q = s'y + sx

Graphic Symbol

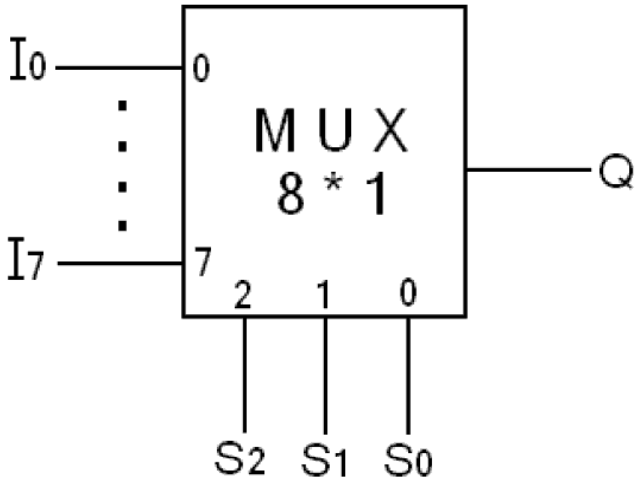




: Multiplexer 4*1 -2



: Multiplexer 8*1 -3

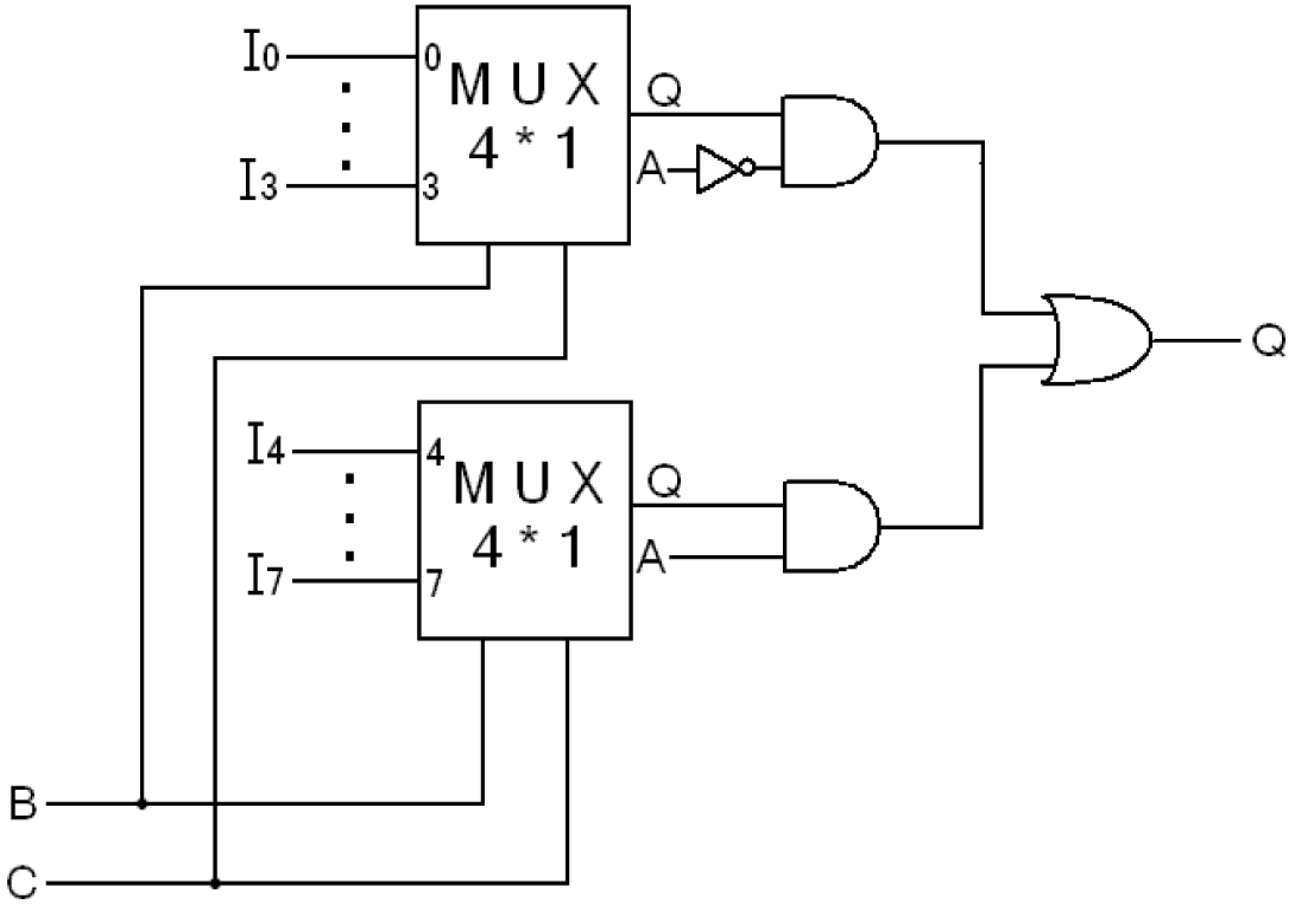


مثال :

: Construct an Multiplexer 8*1 with 2 Multiplexer 4*1 and additional gate

الحل :

المطلوب بناء (Multiplexer 8*1) باستخدام (Multiplexer 4*1) 2 و دوال إضافية

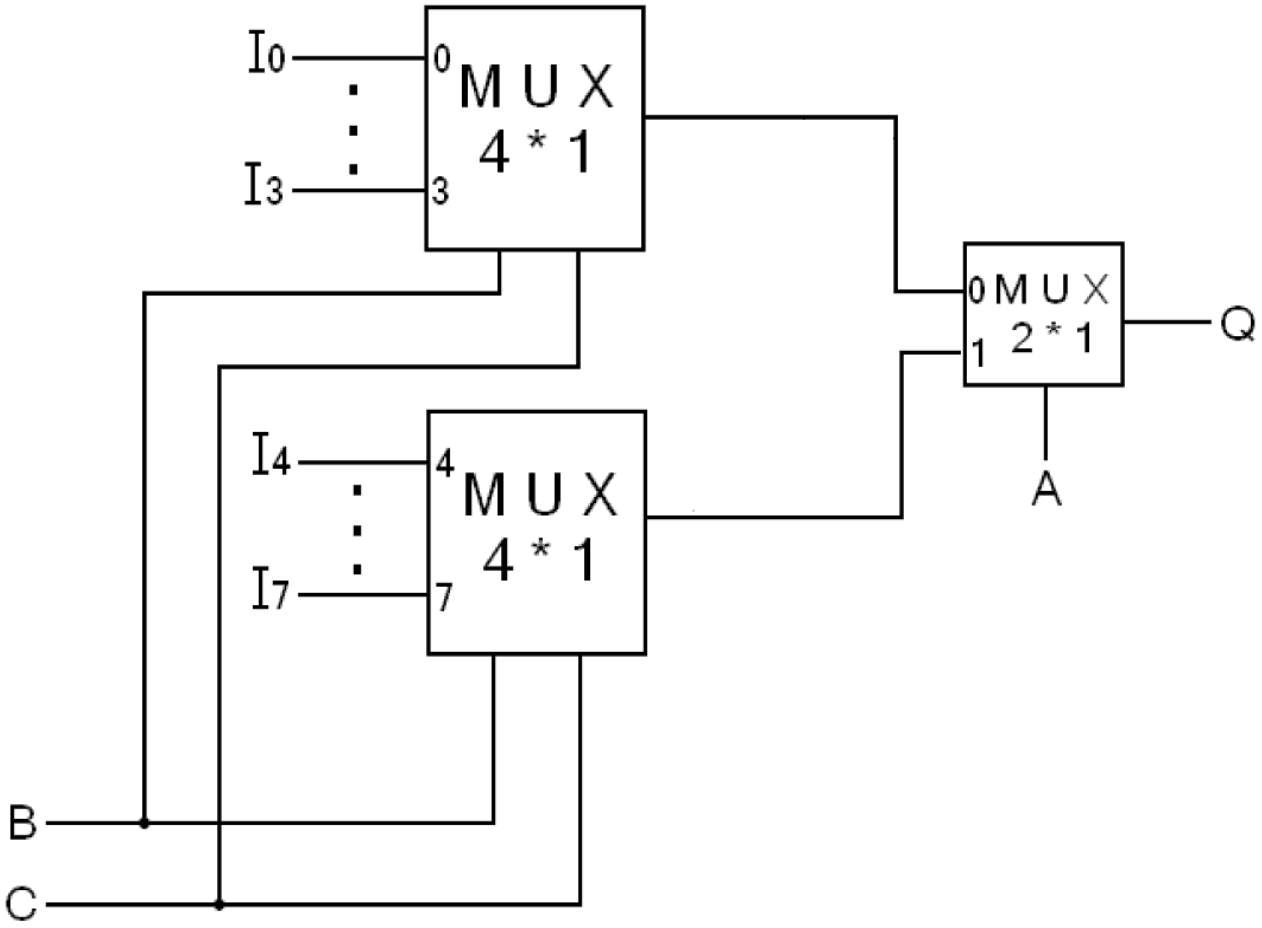


مثال :

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

الحل :

المطلوب بناء (Multiplexer 8*1) باستخدام (Multiplexer 4*1) 2 و (Multiplexer 2*1)

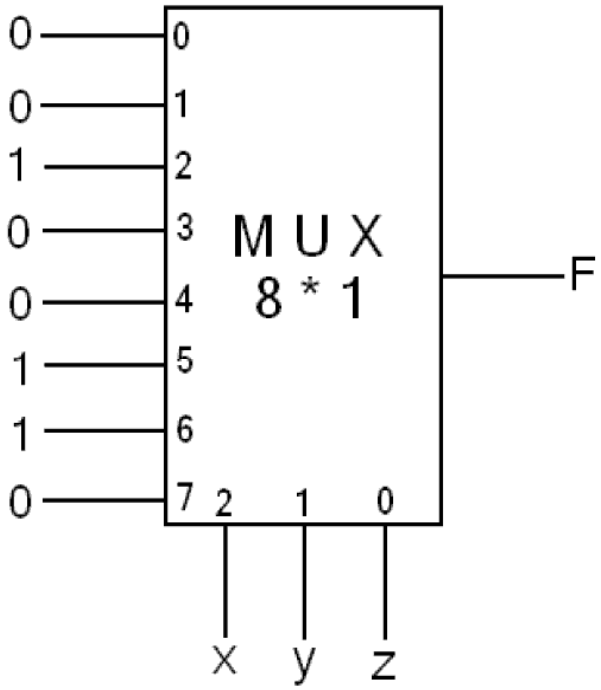


مثال :

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

الحل:

المطلوب تنفيذ الدالة المنطقية التالية باستخدام 8*1 Multiplexer



الشرح :

عند تنفيذ الدوال باستخدام دائرة Multiplexer نقوم بالتالي :

1- نصمم (نرسم) Multiplexer المطلوب استخدامه , وفي مثلنا Multiplexer المطلوب استخدامه (8*1)

2- بعد تصميم Multiplexer ندخل فيه مدخلات الدالة المطلوب تنفيذها

وهي مثالنا مدخلات الدالة المطلوب تنفيذها هي : ∑(2,5,6)

حيث نقوم بإدخال القيمة (1) من كل مدخل للدائرة يعطي قيمة الدالة المطلوب تنفيذها

أي من المدخل 2 للدائرة ندخل القيمة (1)

ثم المدخل 5 ندخل منه أيضاً القيمة (1) , وأخيراً من المدخل 6 ندخل القيمة (1)

باقي مداخل الدائرة والتي لم نستخدمها نقوم بإدخال القيمة (0) منها

وبالتالي نكون قد نفذنا الدالة المطلوبة

مثال:

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

الحل:

المطلوب تنفيذ الدالة المنطقية التالية باستخدام Multiplexer 4*1

هذه الدالة من المفترض أن ننفذها باستخدام Multiplexer 8*1
وذلك لأنه قيم الدالة المطلوب تنفيذها تحتوي على قيم أكبر من Multiplexer المطلوب استخدامه وهو (4*1)
أي أنه لايوجد لدينا مداخل ندخل منها القيمتين 5 و 6
لحل هذه المشكلة علينا التفكير في طريقة أخرى لإدخال القيم للدائرة

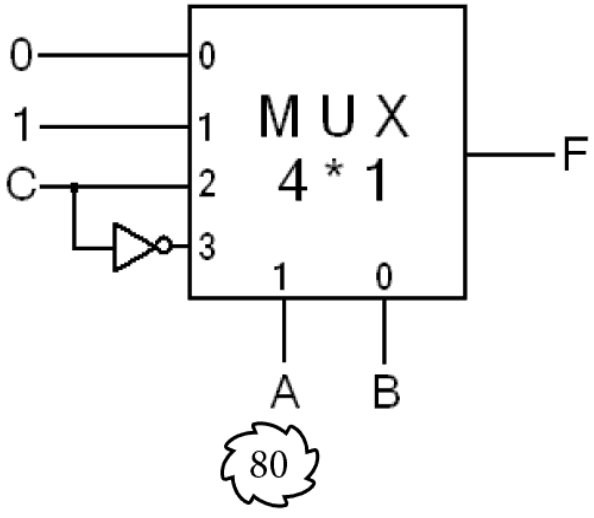
نقوم أولاً بعمل Truth Table للدالة وذلك لكي ربما نجد طريقة نعوض بها عن مدخلات الدائرة

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

من خلال Truth Table نلاحظ على قيم العمود F
أننا نستطيع أن نعوض عن كل صفين متتاليين من هذا العمود بقيمة واحدة
وبالتالي نختزل عدد مدخلات الدالة إلى النصف ونتمكن من تطبيقها باستخدام Multiplexer 4*1

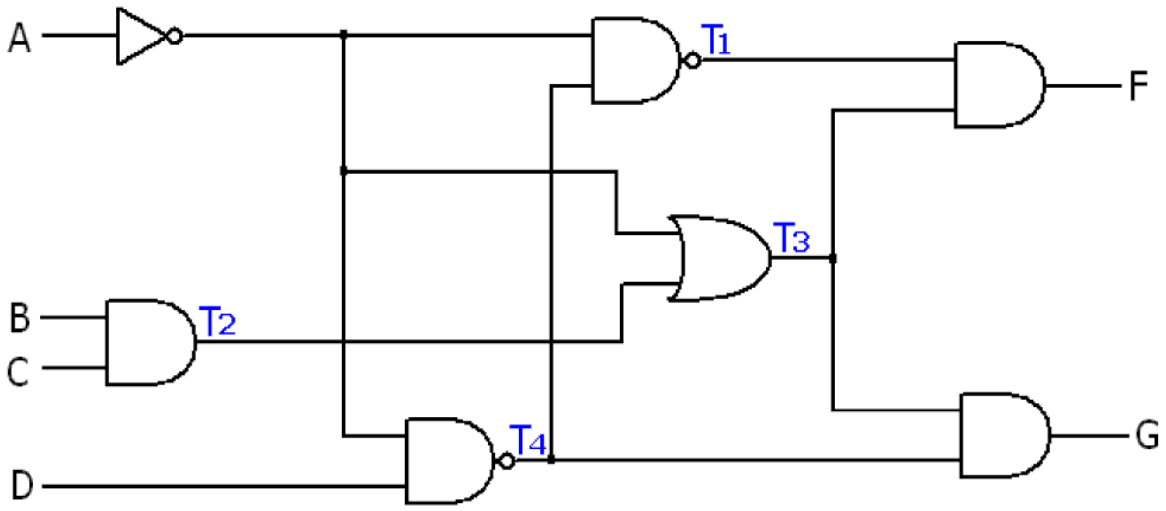
حيث يكون التعويض بالشكل التالي :
الصفين 1 و 2 نعوض عنها بالقيمة (0)
الصفين 3 و 4 نعوض عنها بالقيمة (1)
الصفين 5 و 6 نعوض عنها بالقيمة (C)
الصفين 7 و 8 نعوض عنها بالقيمة (C')

أصبح الآن لدينا 4 مدخلات وهي : $C', C, 0, 1$
بالتالي نستطيع الآن تنفيذ الدالة باستخدام Multiplexer 4*1 وتصبح بالشكل التالي :



4-8 حل أهم تمارين الفصل :

: Obtain the Simplified Boolean expressions for output F and G /Q2
In terms of the input variables in the Circuit

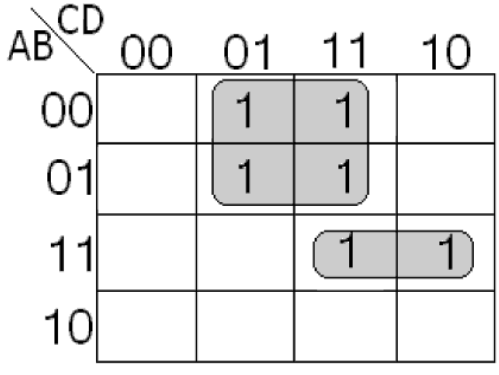
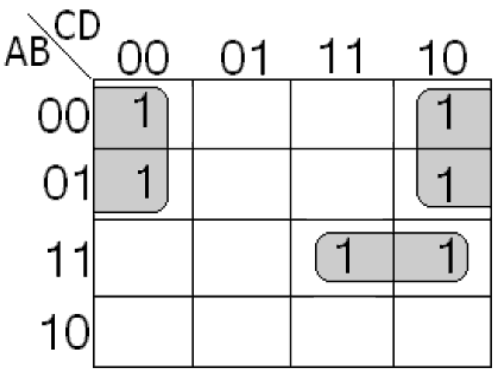


المطلوب :
إيجاد مدخلات ومخرجات الدائرة وتبسيط مخرجاتها

الحل :
1) تحديد المدخلات والمخرجات :

$T_1 = (A'T4)' = A + T_4' = A + (A + D')' = A + A'D = A + D$
$T_2 = BC$
$T_3 = T_2 + A' = A' + BC$
$T_4 = (A'D)' = A + D'$
$F = T_1T_3 = (A + D)(A' + BC) = ABC + A'D$
$G = T_3T_4 = (A' + BC)(A+D') = ABC + A'D'$

: Simplify (2

F	G
	
F = ABC + A'D	G = ABC + A'D'

: Design a combinational Circuit with 3 inputs and 1 output /Q4
The output is 1 when the Binary value of the inputs is less than 3
The output is 0 otherwise

المطلوب :

تصميم دائرة عدد مدخلاتها = 3 و عدد مخرجاتها = 1

إذا كانت قيمة المدخلات أقل من 3 نضع في المخرجات 1

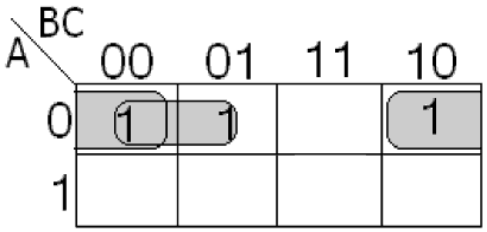
وإذا كانت قيمة المدخلات خلاف ذلك (أكبر أو تساوي 3) تكون المخرجات بـ 0

الحل :

: Truth Table (1

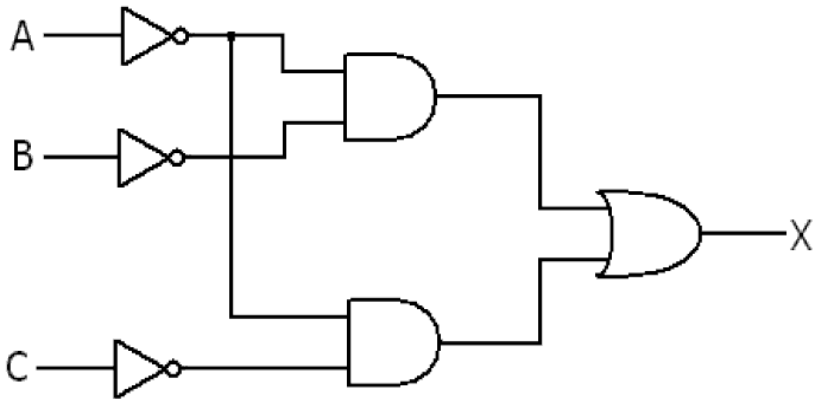
Inputs			Outputs
A	B	C	X
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0

: Simplify (2



X= A'B' + A'C'

: Diagram (3



: Design a combinational Circuit with 3 inputs X,Y,Z and 3 output A,B,C /Q5
When the Binary input is 0,1,2,3 the Binary output is 1 greater than the input
When the Binary input is 4,5,6,7 the Binary is 1 less than the input

المطلوب :

تصميم دائرة لها 3 مدخلات (X,Y,Z) وتنتج 3 مخرجات (A,B,C)

إذا كانت قيمة المدخلات من 0 إلى 3 فإن ناتج المخرجات = العدد المدخل +1

وإذا كانت قيمة المدخلات 4 إلى 7 فإن ناتج المخرجات = العدد المدخل -1

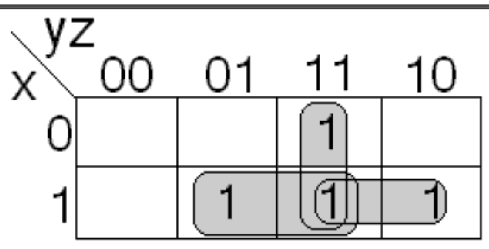
الحل :

: Truth Table (1

Inputs			Outputs		
X	Y	Z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

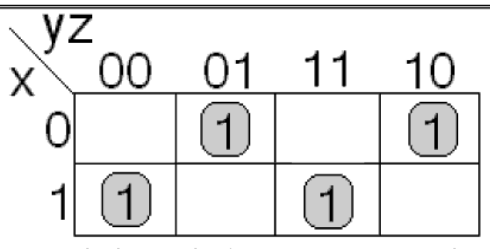
: Simplify (2

A



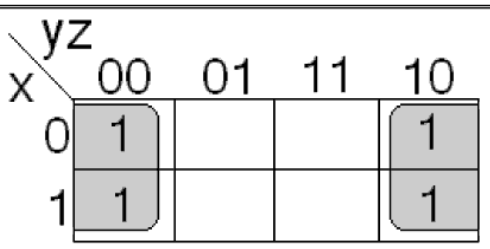
A = XZ + XY + YZ

B



B = XY'Z' + X'Y'Z + XYZ + X'YZ'

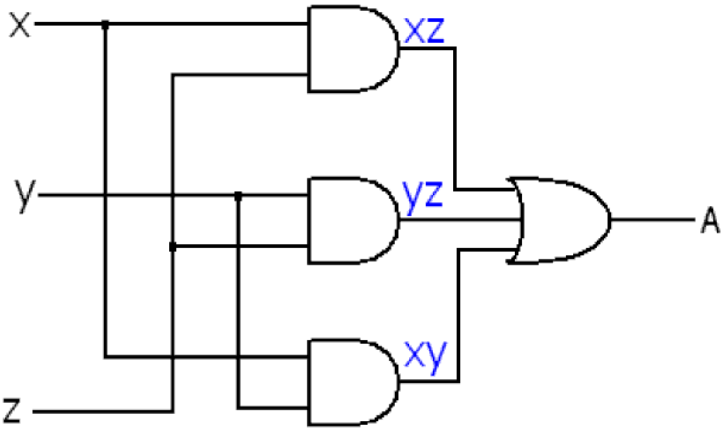
C



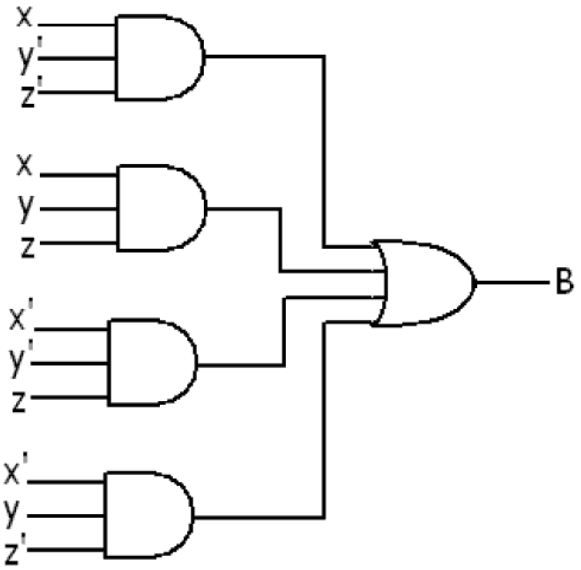
C = Z'

: Diagram (3

A



B



C

/Q6 A majority Circuit is a combinational Circuit whose output is equal to 1 if the input Variables have more 1's then 0's The output is 0 otherwise Design a 3-input majority Circuit :

المطلوب :

تصميم دائرة عدد مدخلاتها = 3 و عدد مخرجاتها = 1

إذا كان عدد الـ 1's في صف المدخلات أكثر من عدد الـ 0's نضع في المخرجات 1

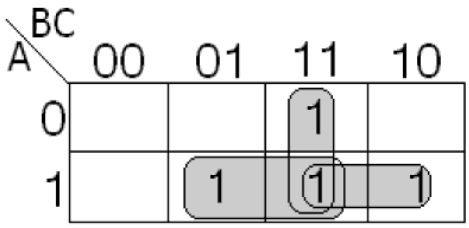
وإذا كان خلاف ذلك (عدد الـ 1's أقل من أو يساوي عدد الـ 0's) نضع في المخرجات 0

الحل :

1) Truth Table :

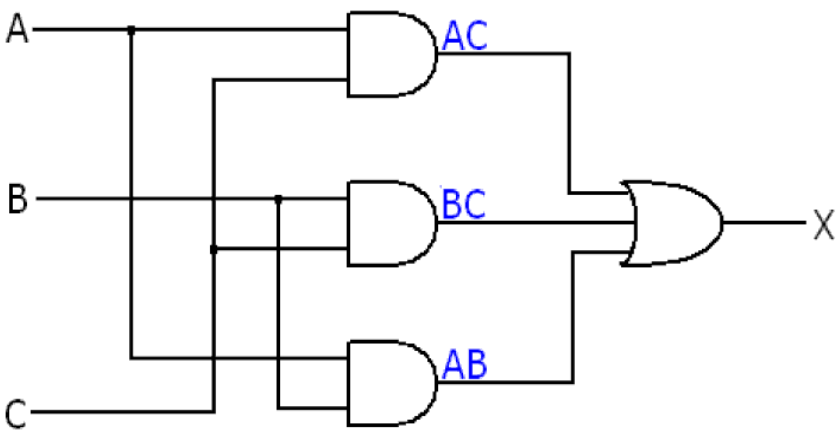
Inputs			Outputs
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

2) Simplify :



X= AB + AC + BC

3) Diagram :



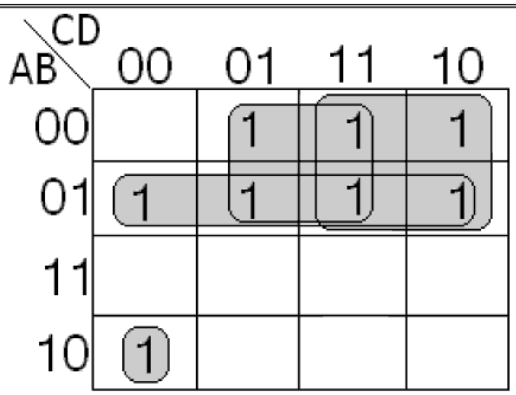
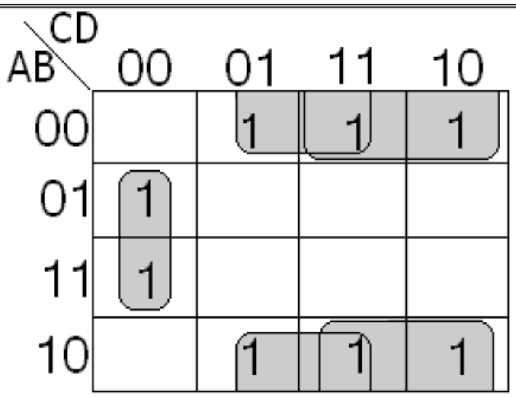
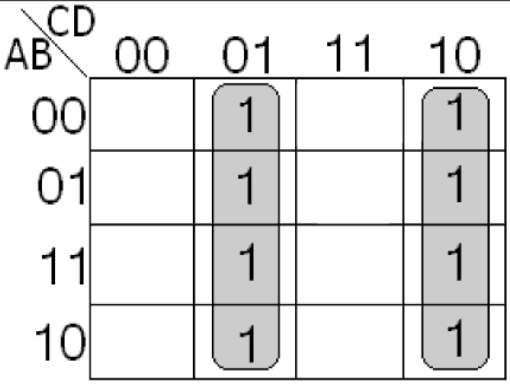
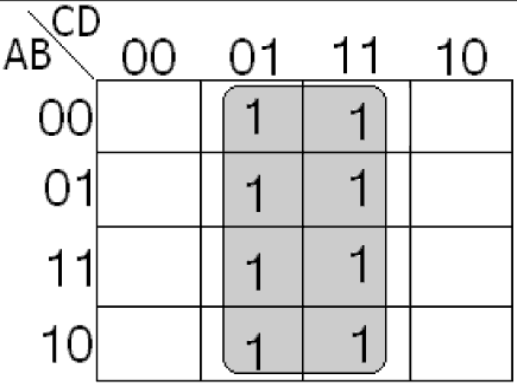
/Q10 Design a 4-bit combinational Circuit 2's complementer :
(The output generates the 2's Complement of the input Binary number)
Show that the Circuit can be constructed using exclusive-OR gates
Can you predict what the output functions are a5-bit 2's complementer ?

المطلوب :
تصميم دائرة مدخلاتها 4-bit ومخرجاتها 4-bit
ناتج المخرجات = المتممة الثانية (2's Complement) للعدد المدخل

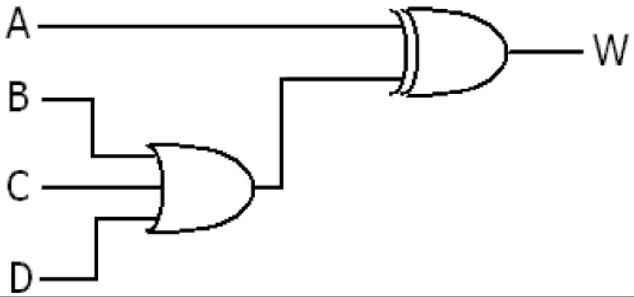
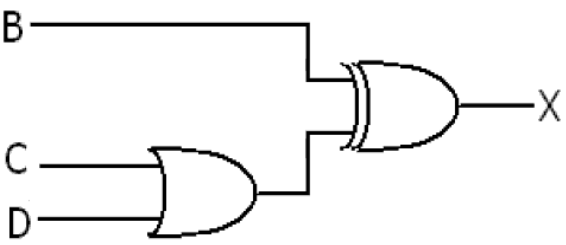
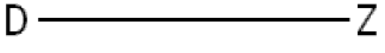
الحل :
1) Truth Table :

Inputs				Outputs			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1
0	0	1	0	1	1	1	0
0	0	1	1	1	1	0	1
0	1	0	0	1	1	0	0
0	1	0	1	1	0	1	1
0	1	1	0	1	0	1	0
0	1	1	1	1	0	0	1
1	0	0	0	1	0	0	0
1	0	0	1	0	1	1	1
1	0	1	0	0	1	1	0
1	0	1	1	0	1	0	1
1	1	0	0	0	1	0	0
1	1	0	1	0	0	1	1
1	1	1	0	0	0	1	0
1	1	1	1	0	0	0	1

: Simplify (2

W	X
	
W = AB'C'D' + A'B + A'C + A'D = A'(B+C+D) + A(B+C+D)' = A ⊕ (B+C+D)	X = B'D + B'C + BC'D' = B'(C+D) + (C+D)' = B ⊕ (C+D)
Y	**Z**
	
Y = CD' + C'D = C ⊕ D	Z = D

: Diagram (3

W	X
	
Y	**Z**
	

Q11/ Design a 4-bit combinational Circuit incrementer (A Circuit that adds 1 : To a 4-bit Binary number) the Circuit can be designed using 4 Half Adders

المطلوب :

تصميم دائرة مدخلاتها 4-bit ومخرجاتها 4-bit

المخرجات = العدد المدخل +1

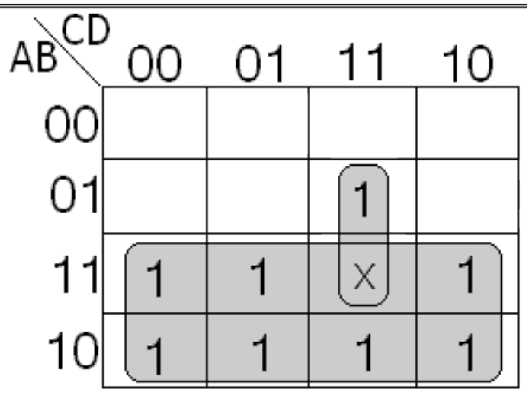
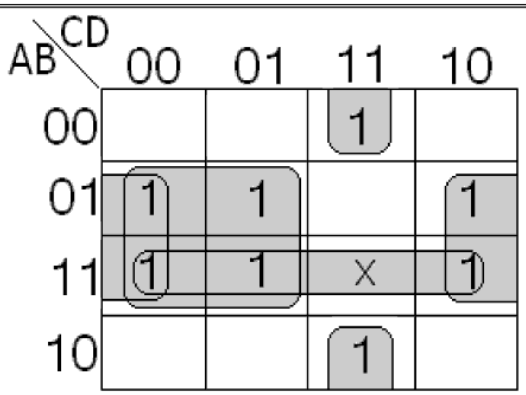
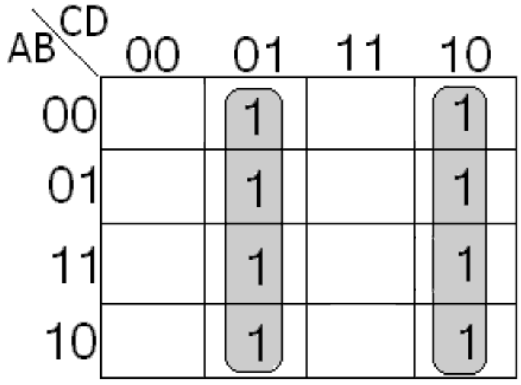
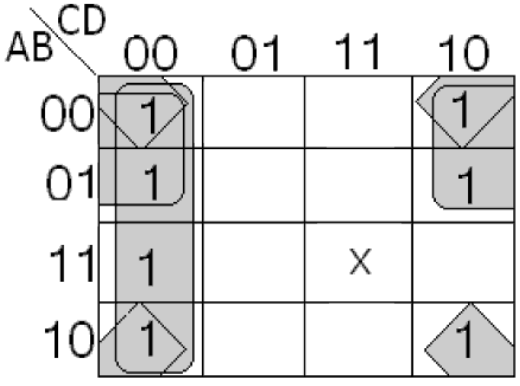
ومن ثم رسم الدالة الناتجة باستخدام 4 (Half Adders)

الحل :

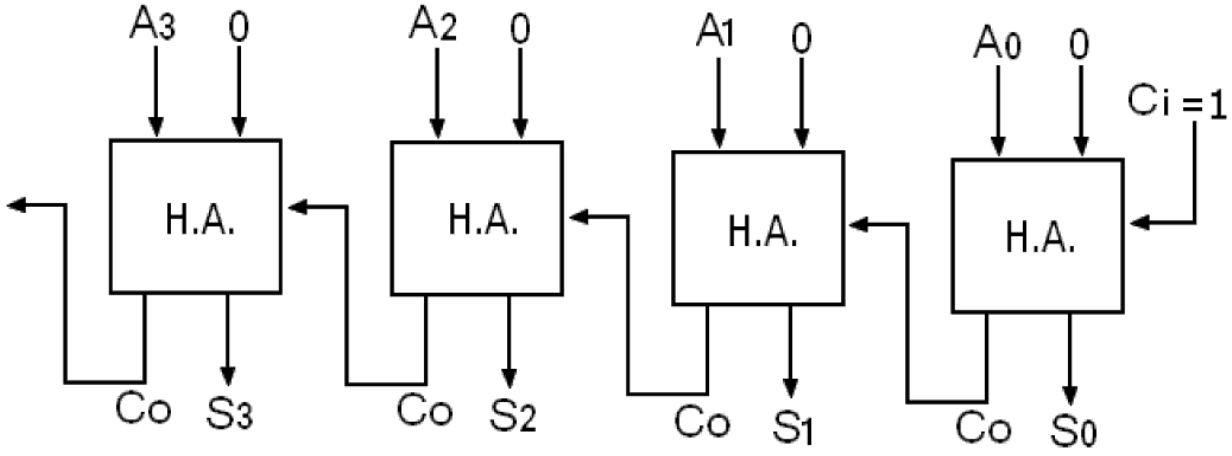
1) Truth Table :

Inputs				Outputs			
A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	0	1	1	0	1
1	1	0	1	1	1	1	0
1	1	1	0	1	1	1	1
1	1	1	1	x	x	x	x

: Simplify (2

W	X
	
W = A + BCD	X = AB + BC' + BD' + B'CD
Y	**Z**
	
Y = C'D + CD'	Z = C'D' + B'D' + A'D'

: Diagram (3



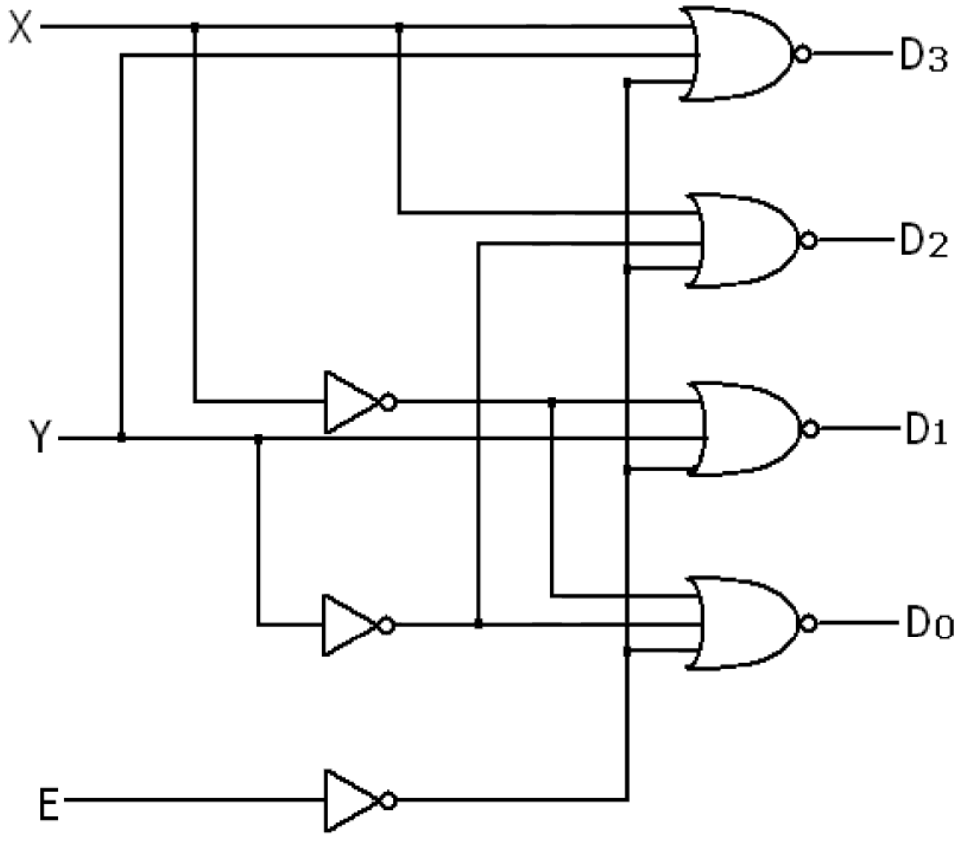
: Draw the Logic Diagram of a 2-to-4-line Decoder using NOR gates only /Q23
Include an Enable input

المطلوب :

رسم (2-to-4-line decoder) باستخدام NOR gates و Enable

الحل :

X	Y	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



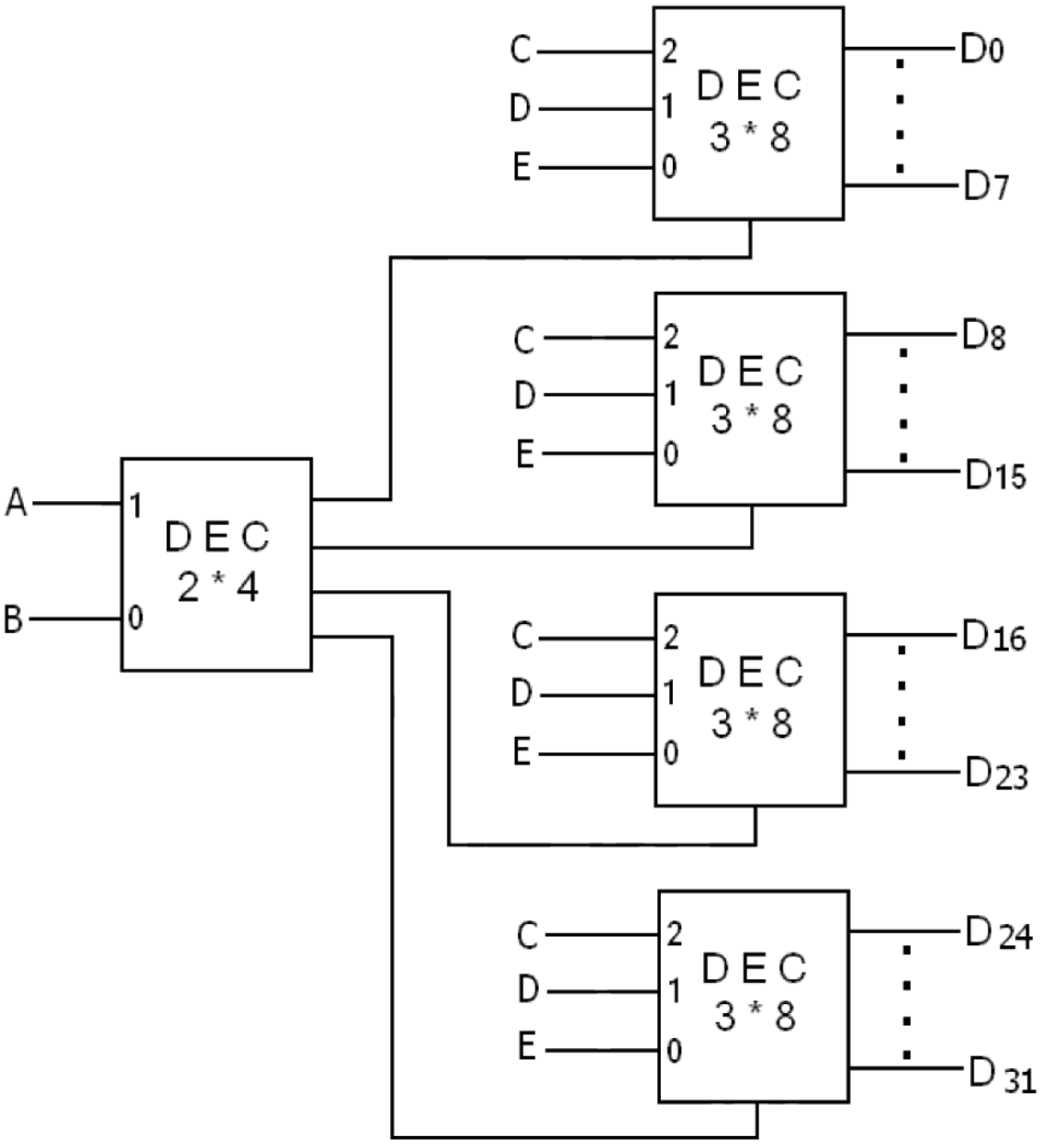
/Q25 Construct a (5-to-32-line Decoder) with 4 (3-to-8-line decoders with Enable) :
And a (2-to-4-line Decoder) Use block diagrams for the components

المطلوب :

بناء (5-to-32-line Decoder)

باستخدام (3-to-8-line Decoders with Enable) 4 و (2-to-4-line decoder)

الحل :



: A combinational Circuit is specified by the following 3 Boolean functions /Q28

$$F_1 = x'y'z' + xz$$
$$F_2 = xy'z' + x'y$$
$$F_3 = x'y'z + xy$$

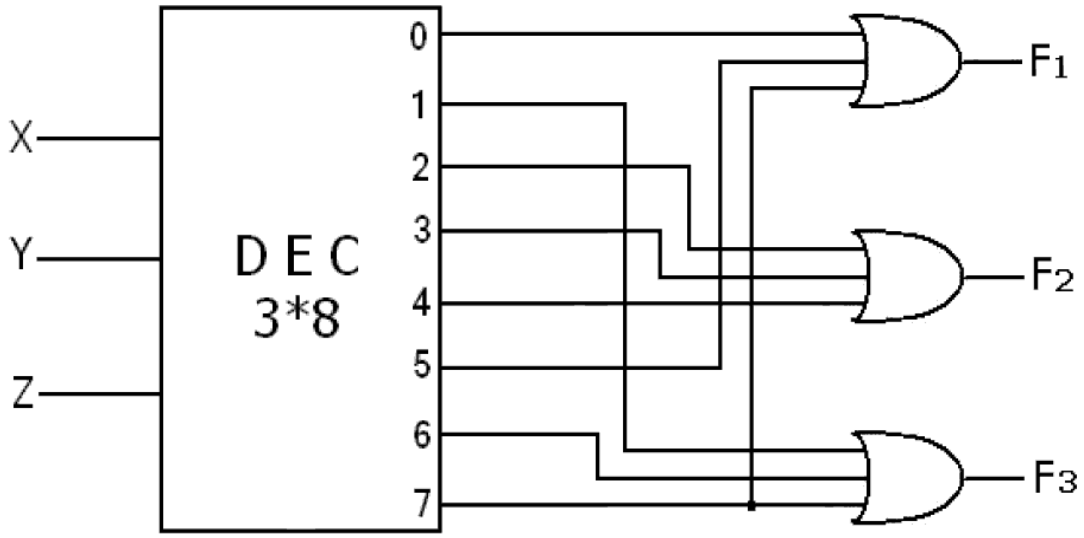
Design the Circuit with Decoder and external gates

المطلوب :

تصميم الدوال المنطقية باستخدام Decoder ودوائر أخرى

الحل :

$$F1 = x'y'z' + xz \Rightarrow x'y'z' + xyz + xy'z = \sum(0,7,5)$$
$$F2 = xy'z' + x'y \Rightarrow xy'z' + x'yz + x'yz' = \sum(4,3,2)$$
$$F3 = x'y'z + xy \Rightarrow x'y'z + xyz + xyz' = \sum(1,7,6)$$

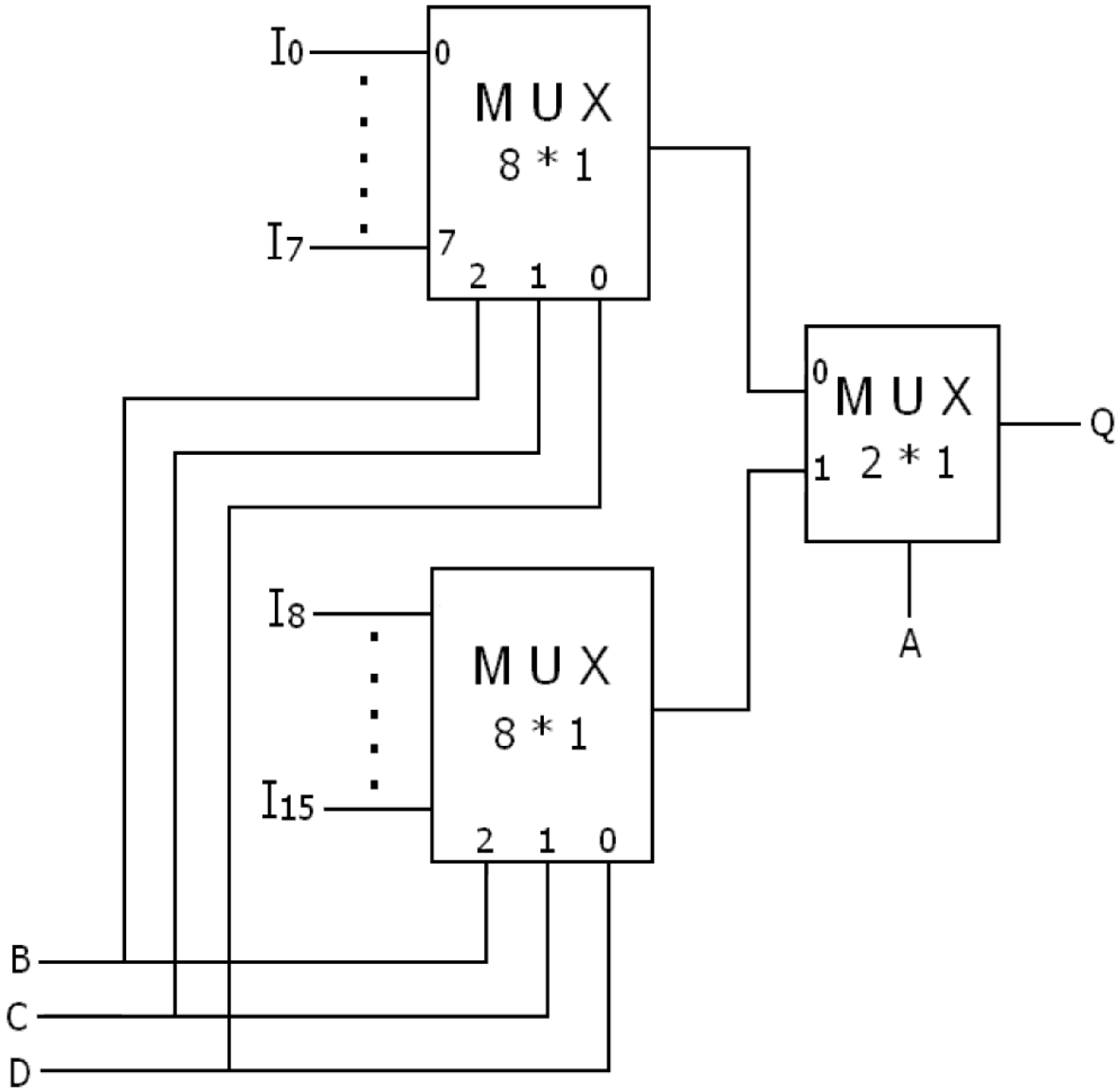


: Construct a (Multiplexer 16*1) with 2 (Multiplexer 8*1) and 1 (Multiplexer 2*1) /Q31
Use block diagrams

المطلوب :

بناء (Multiplexer 16*1) باستخدام (Multiplexer 8 * 1) 2 و (Multiplexer2*1) 1

الحل :



: Implement the following Boolean function with a Multiplexer /Q32

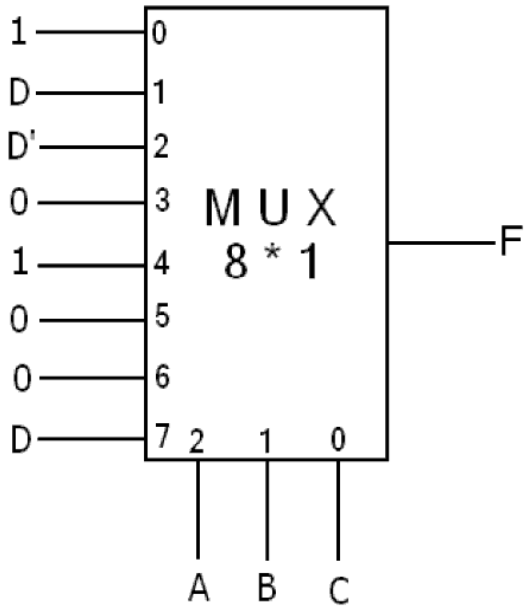
$F(A,B,C,D) = \sum(0,1,3,4,8,9,15)$

المطلوب :

تنفيذ الدالة المنطقية باستخدام Multiplexer

الحل :

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



تستطيع حل هذا السؤال باستخدام Multiplexer 16*1 وذلك لأنه لم يشترط في السؤال استخدام Multiplexer 8*1
ولكن قمنا بحله بـ Multiplexer 8*1 من باب الاختصار فقط وذلك لأننا نستطيع أن نعوض عن كل قيمتين مدخلة بقيمة واحدة
ولك الحرية المطلقة في حل السؤال إما باستخدام Multiplexer 16*1 أو Multiplexer 8*1

الفصل الخامس

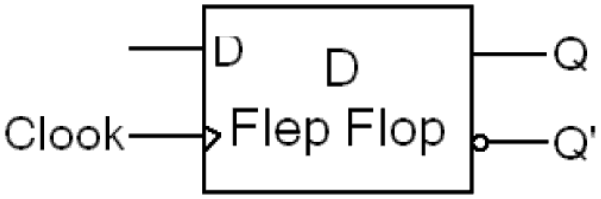
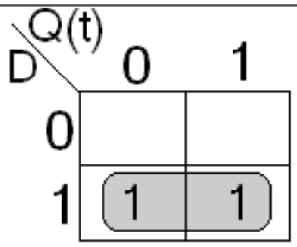
Synchronous Sequential Logic

5-1 المقدمة :

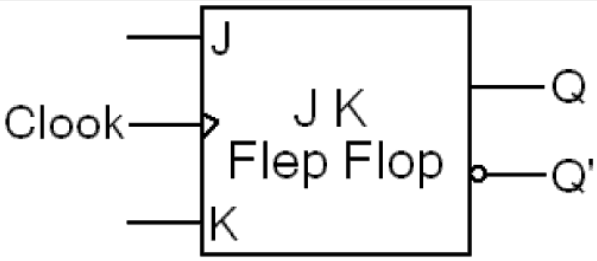
في هذا الفصل سوف نتحدث عن دائرة (Flip Flop)
وسوف نناقشها من حيث تعريفها وأشكالها وأنواعها وأقسامها وعملها وحل المسائل بواسطتها
كما سوف نتحدث عن Analysis Of Clocked Sequential Circuits
كذلك سوف نتحدث عن State Reduction And Assignment
وأخيراً سوف نتحدث عن إجراء التصميم (Design Procedure)

5-2 أنواع الـ Types Of Flip Flop :

1- D Flip Flop :

<table>
<tr><td></td><td> 
</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>
هذا الجدول يوضح كيفية تكوين العمود Q(t+1)
كما تلاحظ عزيزي القارئ أن Q(t+1) = 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 جدول CharacteristicTable
</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>1</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>
هذا الجدول يوضح باختصار مايلي :
عندما ننتقل من Q(t) إلى Q(t+1) كم تصبح قيمة D
</td></tr>
</table>



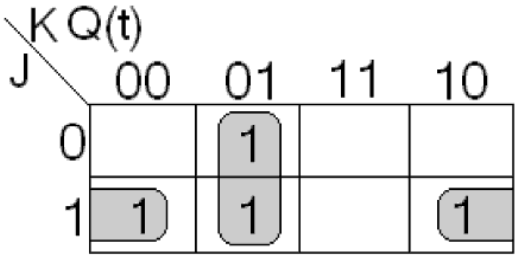
Characteristic Table

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

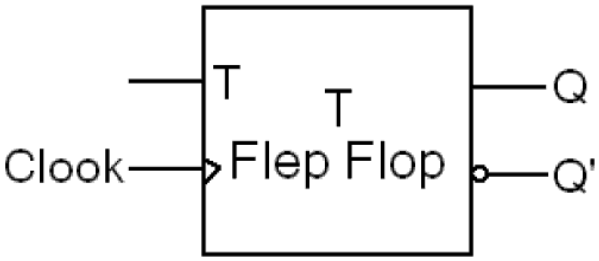
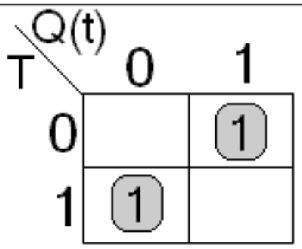
Characteristic Equation



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

Excitation Table

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

<table>
<tr><td></td><td> 
</td></tr>
<tr><td>Characteristic Table</td><td>
<table>
<tr><th>T</th><th>Q(t+1)</th></tr>
<tr><td>0</td><td>Q(t)</td></tr>
<tr><td>1</td><td>Q'(t)</td></tr>
</table>
</td></tr>
<tr><td>Truth Table</td><td>
<table>
<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>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
<tr><td>Characteristic Equation</td><td>


$Q(t+1) = T'Q(t) + TQ'(t) = T \oplus Q(t)$
</td></tr>
<tr><td>Excitation Table</td><td>
<table>
<tr><th>Q(t)</th><th>Q(t+1)</th><th>T</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
</table>
</td></tr>
</table>

خطوات تصميم Flip Flop باستخدام Flip Flop آخر :

1- Truth Table 2- Simplify 3- Diagram

مثال :

Desing a J K Flip Flop using D Flip Flop :

الحل :

المطلوب تصميم J K Flip Flop باستخدام D Flip Flop

1- Truth Table :

عند إنشاء Truth Table

نبدأ من Flip Flop المراد تصميمه وننتهي بـ Flip Flop المستخدم في التصميم

الشكل العام لتكوين أعمدة Truth Table :

Presnt State		Next State	الـ Flip Flop المستخدم
الـ Flip Flop المراد تصميمه	Q(t)	Q(t+1)	

أما بالنسبة لتعبئة الجدول نتبع الآتي :

أعمدة Presnt State نقوم بتعبئتها بالطريقة المعتادة

أما عمود Next State وهو العمود Q(t+1) نعوض في Characteristic Table الخاص بدائرة JK Flip Flop

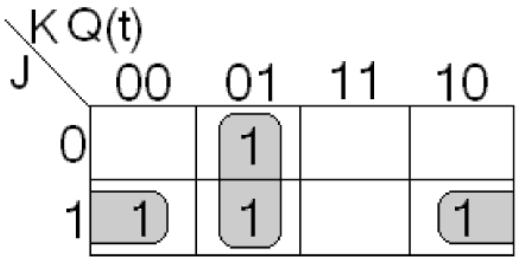
وأخيراً العمود D من خلال التعويض في Excitation Table الخاص بدائرة D Flip Flop

ويصبح Truth Table بالشكل التالي :

P.S			N.S	
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

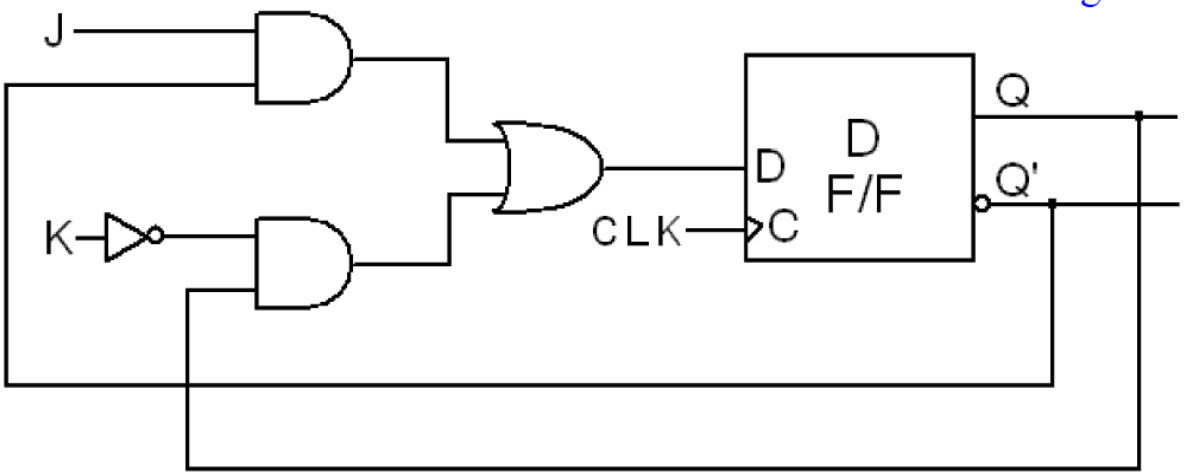
2- Simplify :

حيث نقوم بتبسيط قيمة عمود الـ Flip Flop المستخدم في التصميم وفي مثالنا هو D Flip Flop



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

: Diagram -3



مثال :

: Desing a T Flip flop using J K Flip flop

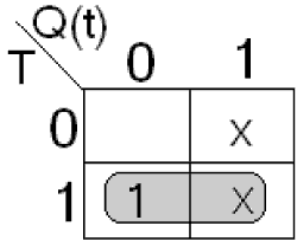
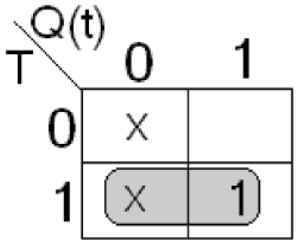
الحل :

المطلوب تصميم T Flip Flop باستخدام JK Flip Flop

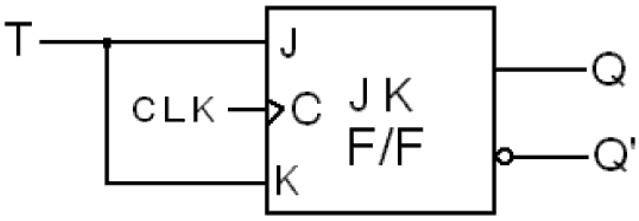
: Truth Table -1

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

: Simplify -2

J	K
	
K = T	J = T

: Diagram -3



5-3 Analysis Of Clocked Sequential Circuits :

خطوات إيجاد Analysis of clocked sequential circuits :

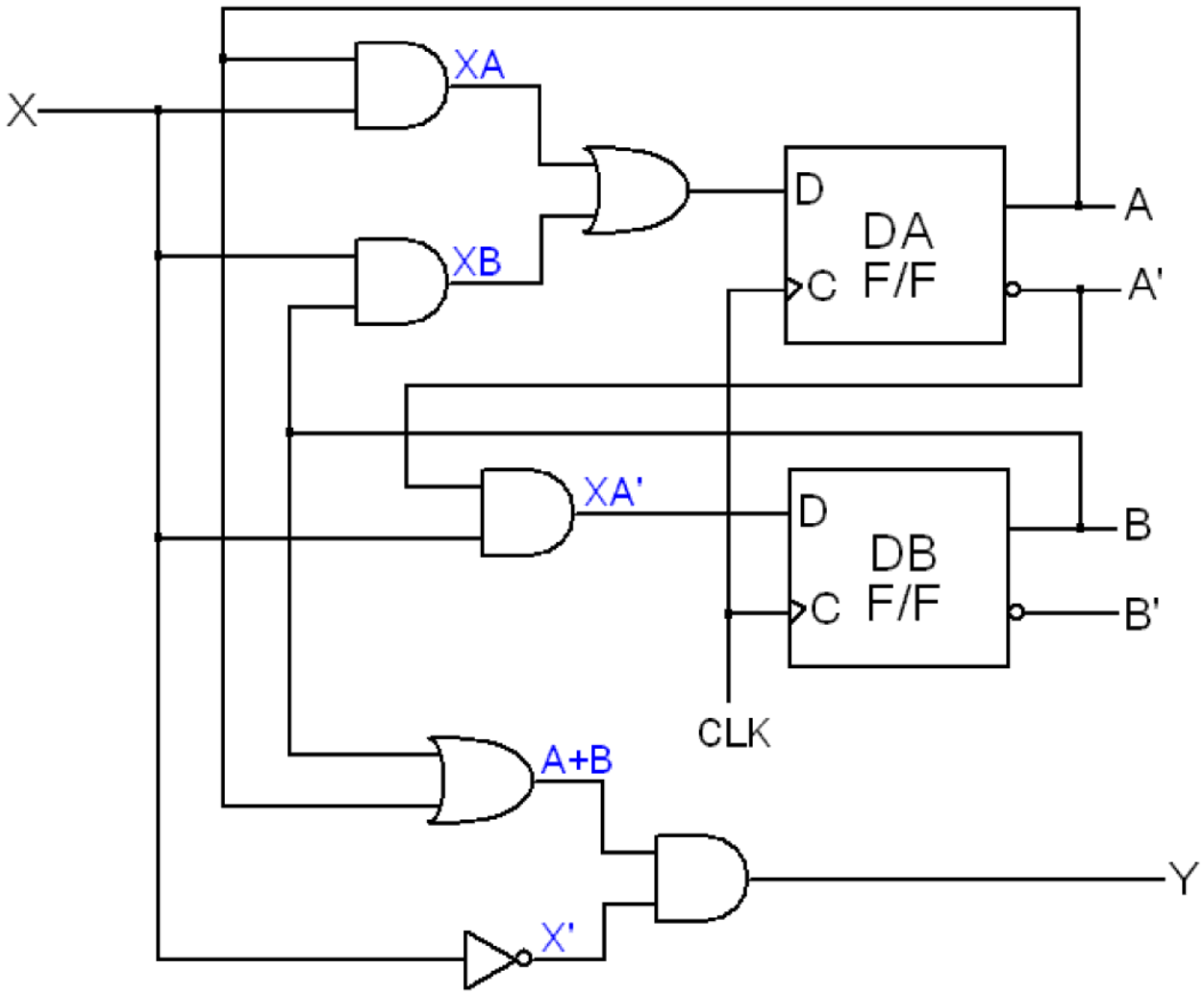
1- تحليل الدائرة وتحديد مدخلاتها ومخرجاتها [تحديد قيم Next State (N.S) و Output (O/P)]

2- State Table

3- State Diagram

مثال:

Analysis of clocked sequential circuit :



الحل:

1- تحليل الدائرة وتحديد مدخلاتها ومخرجاتها :

عند تحليل الدائرة نعامل كل جزء منها على حدا

نبدأ أولاً بـ A Flip Flop ومنه نحصل على القيمة :

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

ثم نأخذ B Flip Flop ومنه نحصل على القيمة :

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

وأخيراً نأخذ Y ومنه نحصل على القيمة :

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

وبذلك نكون قد حللنا الدائرة كاملةً وحصلنا على المطلوب إيجاده

: State Table -2

وفيها نقوم أولاً بتقسيم خلايا State Table ومن ثم تعبئتها

الشكل العام لتكوين أعمدة State Table :

P.S	I/P	N.S	O/P

والآن ننتقل للخطوة التالية وهي تعبئة الحقول

نبدأ أولاً بعمودي (P.S) والعمود (I/P) ونقوم بتعبئة حقولها بالطريقة المعتادة

أما بالنسبة لباقي الأعمدة [عمودي (N.S) والعمود (O/P)]

فيتم تعبئتها بناءً على التعويض في قيم الدائرة والتي قمنا بإيجادها في الخطوة الأولى

من التعويض في قيمة A(t+1) نوجد العمود A

ومن التعويض في قيمة B(t+1) نوجد العمود B

ومن التعويض في قيمة Y نوجد العمود Y

على سبيل المثال لو أخذنا الصف الرابع لنبين كيف قمنا بتعبئة حقوله

قيم أعمدة (P.S) و (I/P) في هذا الصف كالتالي : (X = 1) ، (B = 1) ، (A = 0)

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

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

ثم نعوض في B(t+1) = XA' ومنه نوجد قيمة الخانة B

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

وأخيراً نعوض في Y = X'A + X'B ومنه نوجد قيمة الخانة Y

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

تابع الحل

ثم نقوم بتعبئة حقول الجدول بناءً على القيم التي توصلنا إليها
كما فعلنا في الجدول التالي :

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

كما تلاحظ عزيزي القارئ أن الجدول إلى هذه اللحظة
يحتوي على قيم عمودي (P.S) والعمود (I/P)
والصف الرابع من عمودي (N.S) وعمود (O/P)

بعد ذلك نقوم **بإكمال** تعبئة باقي الحقول كما فعلنا مع الصف الرابع
ويصبح الجدول بالشكل التالي :

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

وبالتالي نكون قد انتهينا من انشاء State Table

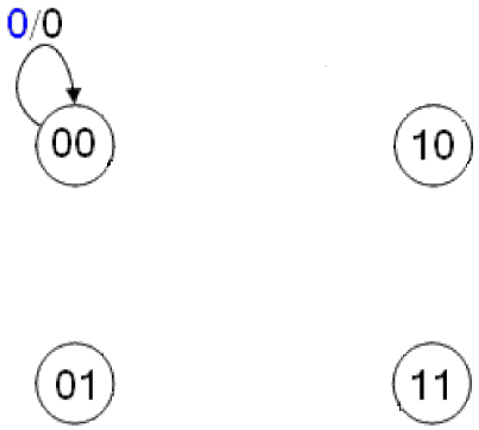
3- State Diagram :

State Diagram عبارة عن وصف لـ State Table
وعند وجود أحدهما نستطيع إيجاد الآخر منه

رسمة State Diagram باختصار عبارة عن دوائر وأسهم :
نضع داخل الدوائر قيم (P.S) و(N.S)
أما على الأسهم فنضع قيم (I/P) و (O/P) بحيث أن :
القيمة الواقعة على يسار الخط الفاصل تمثل قيمة (I/P)
أما القيمة الواقعة على يمين الخط الفاصل تمثل قيمة (O/P)

عندما نريد رسم State Diagram لـ State Table
نطلق سهم من دائرة (P.S) إلى دائرة (N.S) , ونضع على السهم قيم (I/P) و (O/P)

على سبيل المثال لو أخذنا الصف الأول من State Table لنبين كيف قمنا برسم State Diagram له
قيم (P.S) و (I/P) و (N.S) و (O/P) في هذا الصف كالتالي :
P.S = 00 , I/P = 0 , N.S = 00 , O/P = 0
أذن سوف نطلق سهم من الدائرة (00) إلى الدائرة (00) ونضع على السهم القيمة 0/0
وبالتالي سوف تصبح رسمة State Diagram لهذا الصف كالتالي :

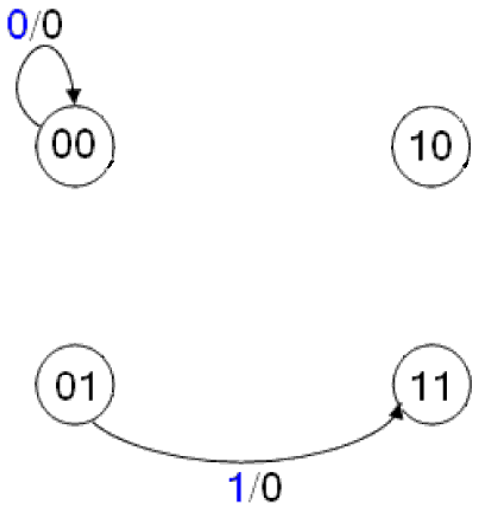


تابع الحل

كذلك لو أخذنا الصف الرابع من State Table لنبين كيف قمنا برسم State Diagram له
قيم (P.S) و (I/P) و (N.S) و (O/P) في هذا الصف كالتالي :

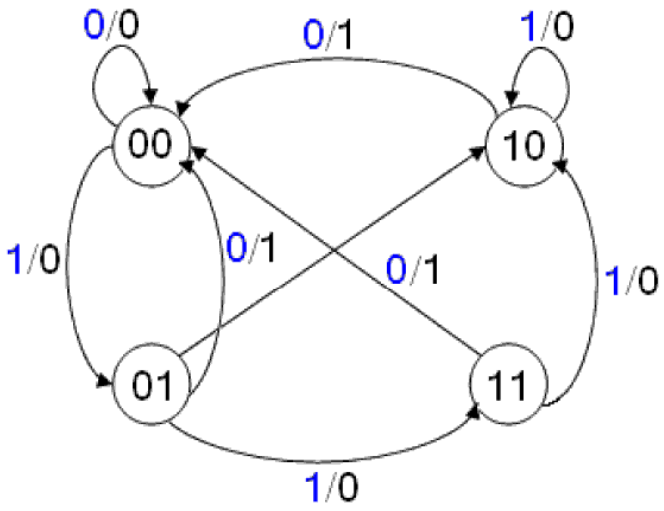
P.S = 01 , I/P = 1 , N.S = 11 , O/P = 0

بالتالي سوف تكون رسمة State Diagram لهذا الصف كالتالي :



بعد ذلك نقوم بإكمال رسمة State Diagram لباقي صفوف State Table
كما فعلنا مع الصفين الأول والرابع

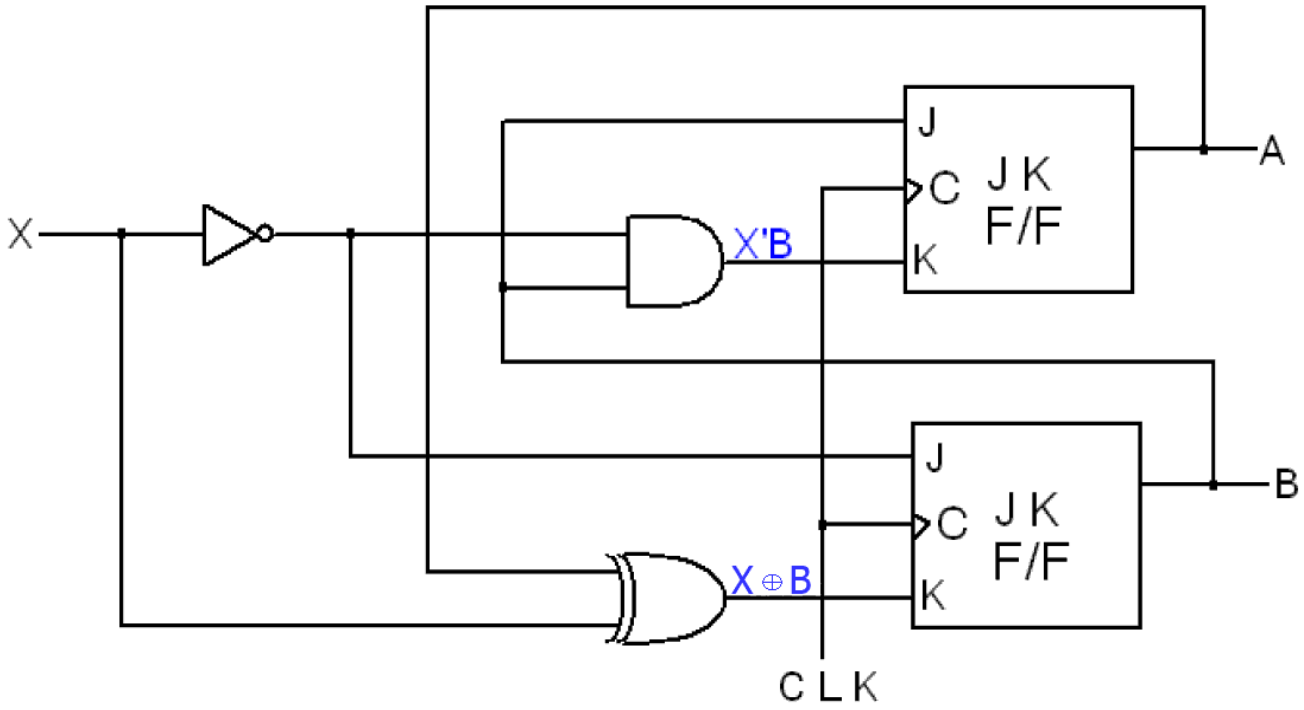
وتصبح الرسمة بالشكل التالي :



وبالتالي نكون قد انتهينا من رسم State Diagram

مثال:

: Analysis of clocked sequential circuit



الحل :

1- تحليل الدائرة وتحديد مدخلاتها ومخرجاتها :

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

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

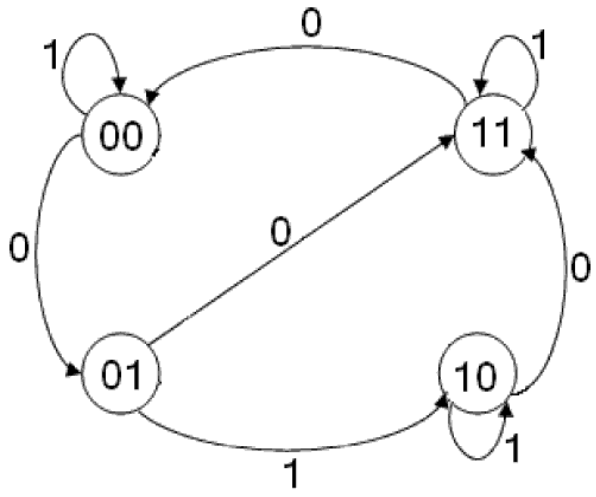
$$A(t+1) = J_A.A' + K'_A.A$$
$$= BA' + (X'B)'A$$
$$= A'B + (X+B')A$$
$$= A'B + XA + AB'$$

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

: State Table -2

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

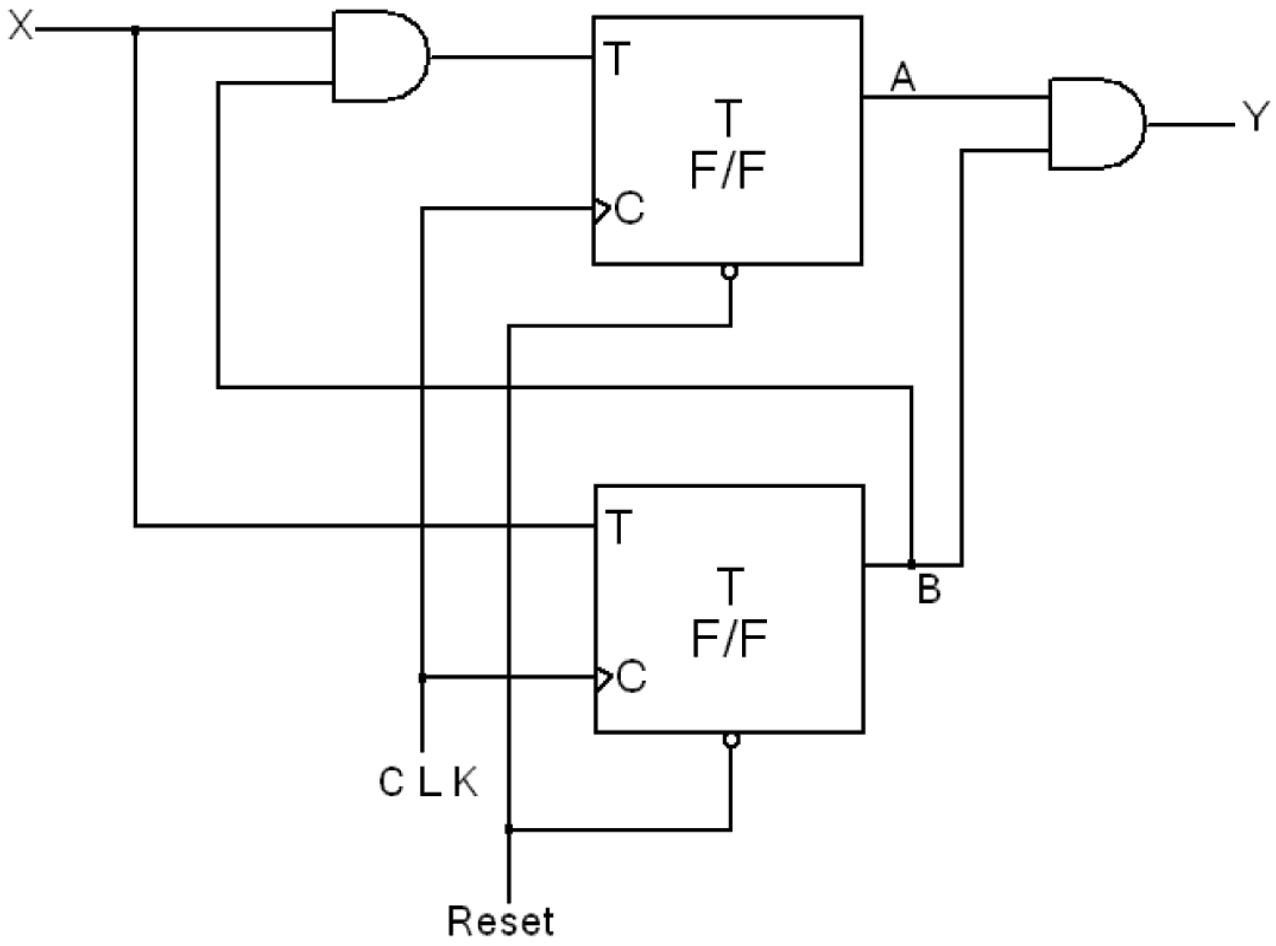
: State Diagram -3



لاحظ عزيزي القارئ على رسمة State Diagram لهذا المثال
أنه توجد فقط قيمة واحدة على الأسهم وهي قيمة (I/P)
وذلك لأنه لا يوجد (O/P)

مثال:

: Analysis of clocked sequential circuit



الحل:

1- تحليل الدائرة وتحديد مدخلاتها ومخرجاتها:

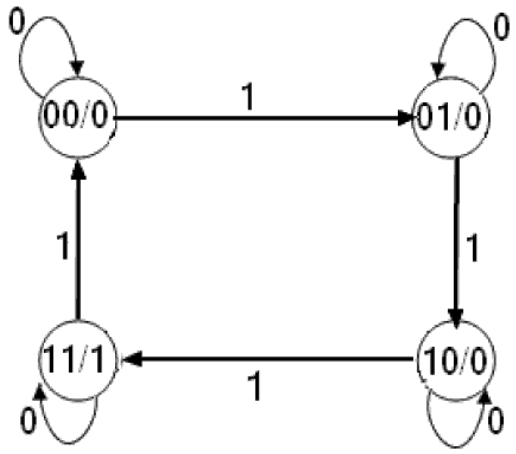
$T_A = XB$ $T_B = X$

$A(t+1) = T_A \oplus A = XB \oplus A$ $B(t+1) = T_B \oplus B = X \oplus B$ $Y = AB$

: State Table -2

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 -3



لاحظ عزيزي القارئ على رسمة State Diagram أنه تم وضع قيم (O/P) داخل الدوائر
وذلك لأن (O/P) لايعتمد على قيمة (I/P)
إرجع لرسمة الدائرة التي قمنا بتحليلها وسوف تلاحظ أنه لايوجد خط من X داخل إلى Y
بينما تلاحظ أنه الخطوط الداخلة إلى Y خارجة من AB (Y = AB)
لذلك نقوم بوضع قيم (O/P) داخل الدوائر

ملاحظة :

مرت بنا 3 أنواع من Analysis of clocked sequential circuits :
الأول / وفيه (O/P) يعتمد في تكوينه على (I/P)
الثاني / لا يوجد به (O/P)
الثالث / (O/P) لا يعتمد في تكوينه على (I/P)

ونلاحظ أنه اختلف تكوين State Table ورسم State Diagram لكل نوع عن الآخر

إرجع لـ الأمثلة الـ 3 السابقة لتلاحظ الاختلافات

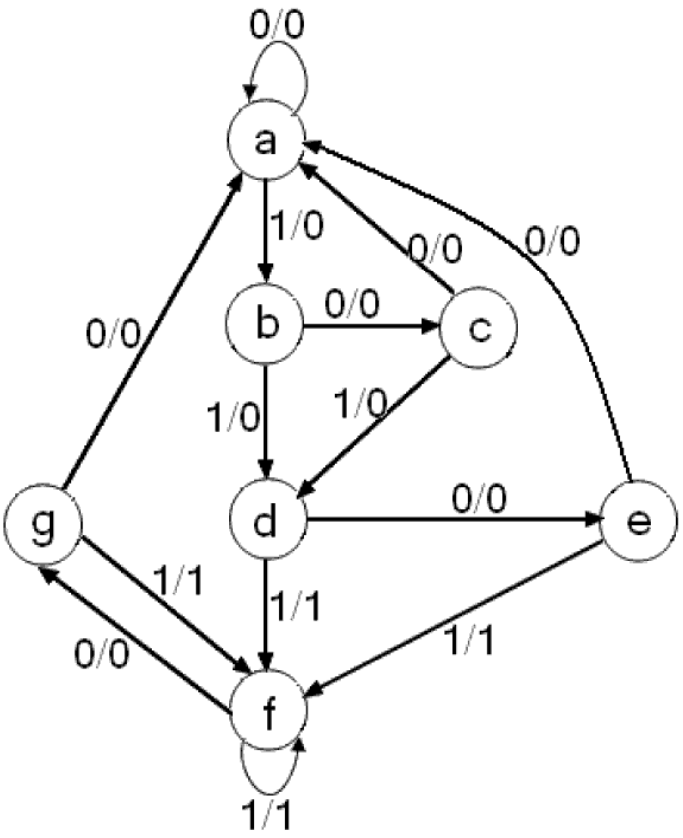
5-4 State Reduction And Assignment :

خطوات إيجاد state reduction and assignment :

1- تحليل رسمة State Diagram و إنشاء State Table

2- تبسيط State Table إلى أبسط صورة ممكنة

مثال :



الحل:

1- تحليل رسمة State Diagram وإنشاء State Table :

داخل الدوائر توجد قيم (P.S) و (N.S)
الجديد في الأمر أنه عبر عنها هذه المرة بحروف ولم يعبر عنها بأرقام
ليعبر بما شاء لا يهمنا ذلك الذي يهمنا فقط أنها تمثل قيمة (P.S) و (N.S)

أما القيم التي على الأسهم فهي :
القيمة التي على يمين الخط الفاصل تمثل قيمة (O/P)
أما القيمة التي على يمين الخط الفاصل فهي ليست (I/P)
وسوف نقوم بتوضيحها عند إنشاء State Table

الشكل العام لتكوين أعمدة State Table :

P.S	N.S		O/P	
	X = 0	X = 1	X = 0	X = 1

تعرفنا على الشكل العام لـ State Table والآن ننشئ State Table لرسمة State Diagram السابقة
نقوم الآن بتحليل رسمة State Diagram ونلاحظ أنها تتكون من 7 دوائر
نقوم وضع قيم هذه الدوائر في العمود (P.S) من الجدول ويصبح الجدول بالشكل التالي :

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

وبالتالي نكون قد انتهينا من تعبئة أول عمود من الجدول

نقوم الآن بتعبئة باقي الأعمدة
ولتعبئتها نتتبع الأسهم من أين انطلقت وإلى أين وصلت
[من أي دائرة (P.S) انطلق وإلى أي دائرة (N.S) وصل]

على سبيل المثال نأخذ الجزء التالي من الرسمة لنقوم بتحليله وتعبئة حقول الجدول بناءً على تحليله :

تحليل الشكل السابق يكون كالتالي :

انطلق سهم من الدائرة (a) إلى الدائرة (a) وذلك عندما كانت X = 0 و المخرجات = 0

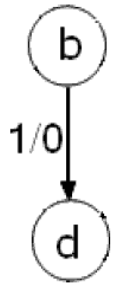
[P.S = a ⟶ N.S = a when X = 0 , O/P = 0]

بعد ذلك نقوم بتعبئة حقول الجدول التي أشار إليها تحليل رسمة State Diagram كما في الصفحة التالية

تابع الحل

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

كذلكً على سبيل المثال نأخذ الجزء التالي من الرسمة لنقوم بتحليله وتعبئة حقول الجدول بناءً على تحليله :



تحليل الشكل السابق يكون كالتالي :

انطلق سهم من الدائرة (b) إلى الدائرة (d) وذلك عندما كانت X = 1 و المخرجات = 0

[P.S = b ⟶ N.S = d when X = 1 , O/P = 0]

ويصبح State Table بالشكل التالي :

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

State Table إلى هذه اللحظة مكوّن من تحليل سهمين من رسمة State Diagram

تابع الحل

بعد ذلك نكمل تتبع باقي الأسهم بنفس الطريقة التي تتبعنا بها السهمين السابقين
إلى أن ننتهي من تتبع جميع الأسهم وتعبئة الجدول
عندها نكون قد انتهينا من الخطوة الأولى وهي تحليل رسمة State Diagram وإنشاء State Table
ويصبح شكل State Table كالتالي :

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

الآن ننتقل للخطوة الثانية وهي تبسيط State Table إلى أبسط صورة ممكنة

وفي هذه الخطوة نقوم بالبحث عن صفين لهما نفس القيم

نلاحظ أن الصفين (e) و (g) لهما نفس القيم كما تلاحظ في الجدول التالي :

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

نقوم الآن بإلغاء أحد هاذين الصفين وليكن (g)
وعند إلغاء الصف يحدث ما يلي :
1- نحذف الصف الملغى من الحل [الصف (g)]
2- نستبدل كل (g) بـ (e)
كما في الصفحة التالية

تابع الحل

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

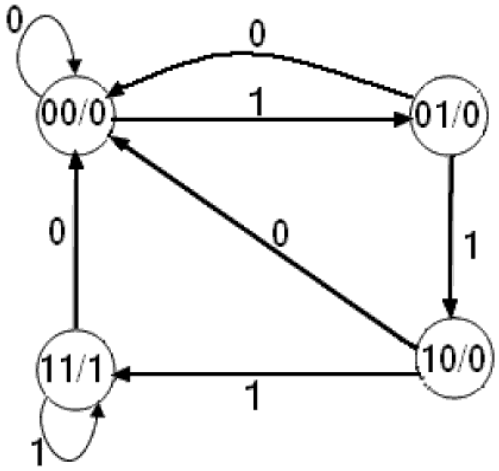
بعد إلغائنا الصف (f) وإستبدال كل (f) بالقيمة (d) نلاحظ أن هذه الصورة للجدول هي أبسط صورة ممكنة وبالتالي نكون قد تمكنا من الوصول للحل النهائي للمثال

5-5 إجراء التصميم Design Procedure :

خطوات إجراء التصميم(Design Procedure):

State Table -1
Simplify -2
Diagram -3

مثال:



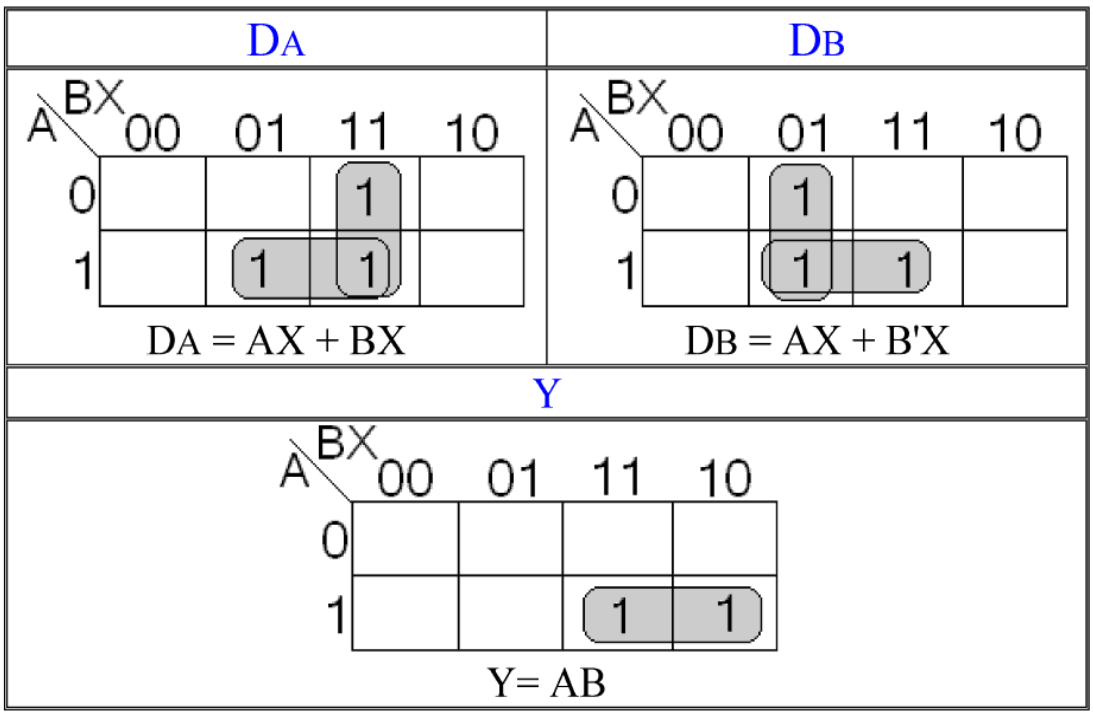
الحل :

حل المثال بـ D Flip flop :

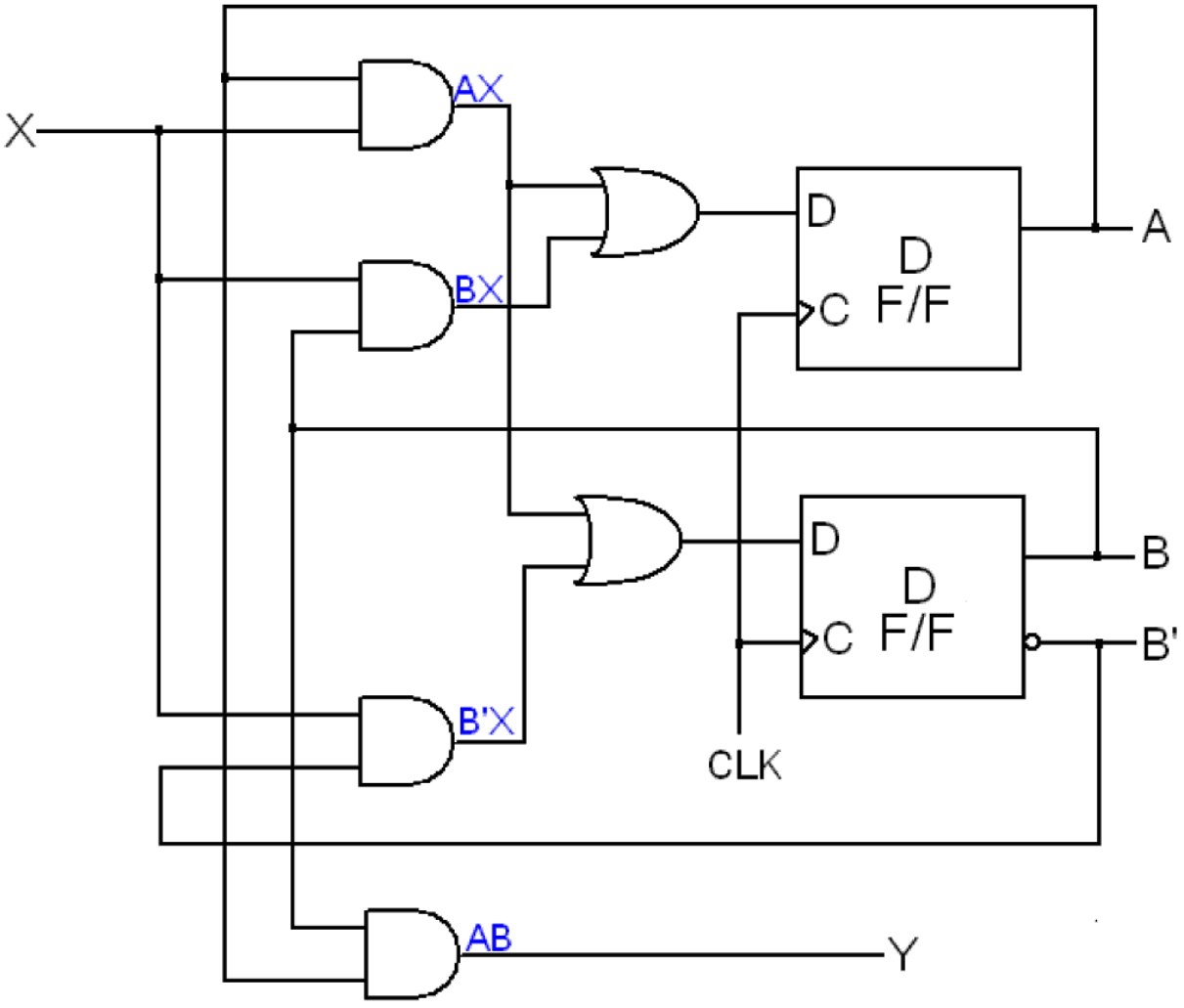
State Table -1 :

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

: Simplify -2



: Diagram -3



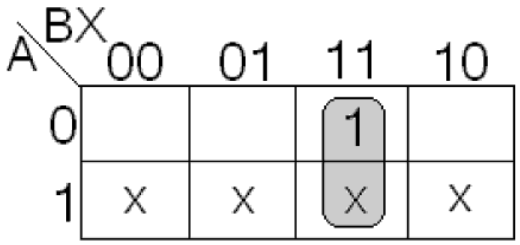
حل المثال بـ J K Flip flop :

: State Table -1

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	1	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	1	0	X	0	1	X
1	1	0	0	0	1	X	1	X	1
1	1	1	1	1	1	X	0	X	0

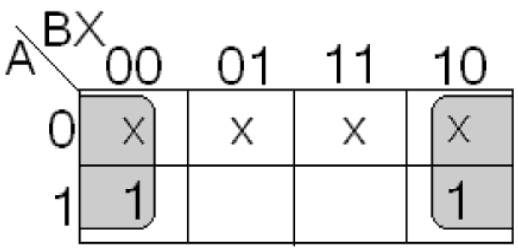
: Simplify -2

JA



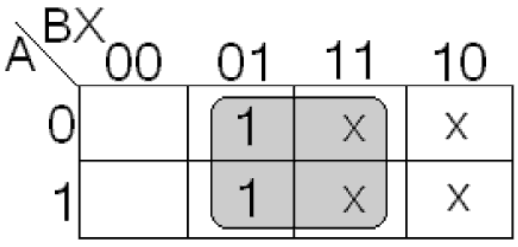
JA = BX

KA



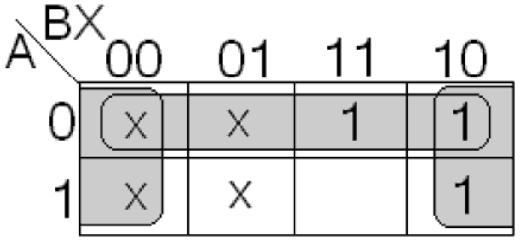
KA = X'

JB



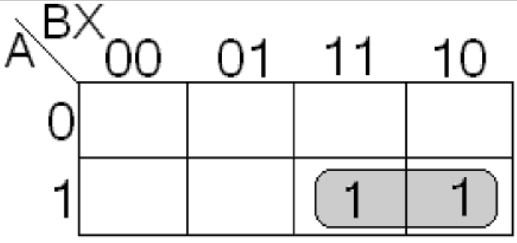
JB = X

KB



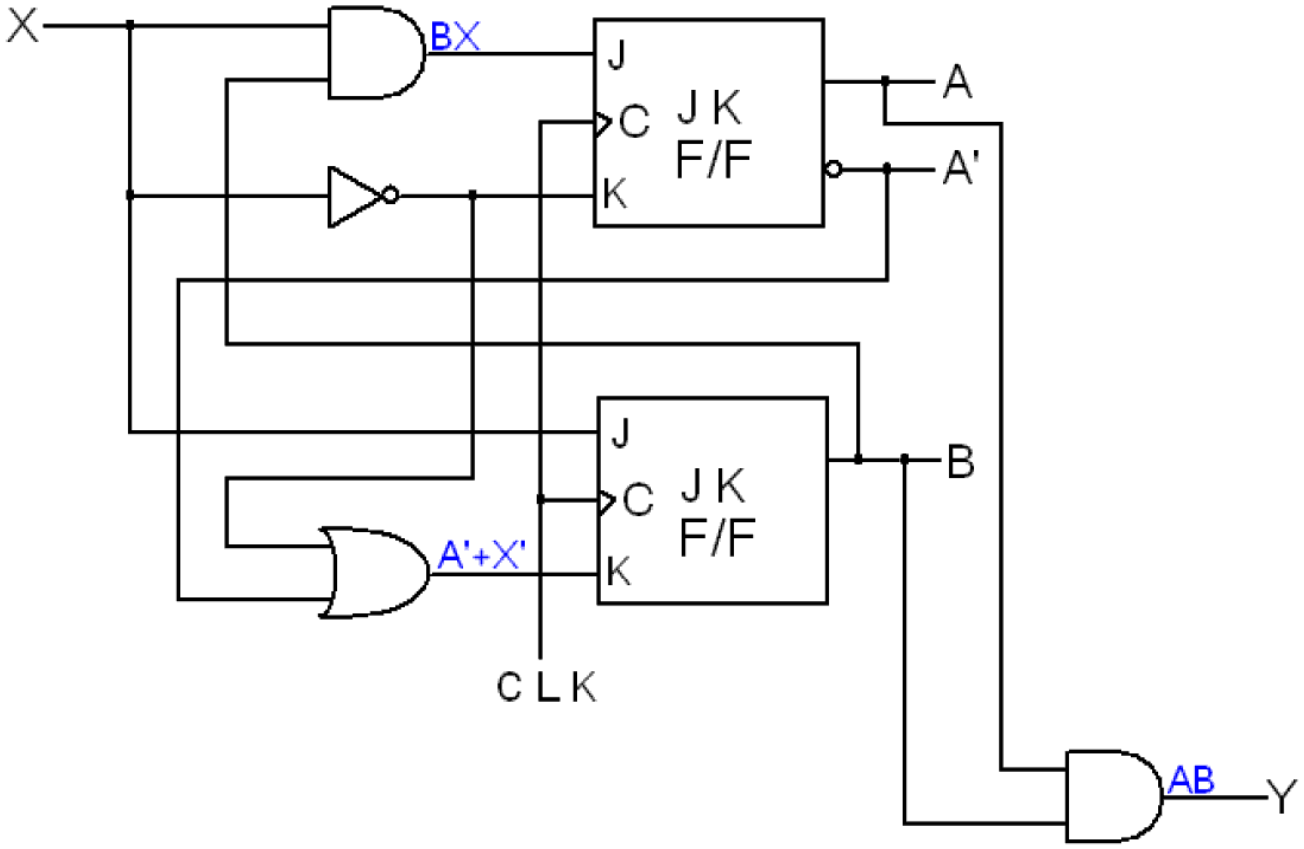
KB = A'+X'

Y



Y= AB

: Diagram -3

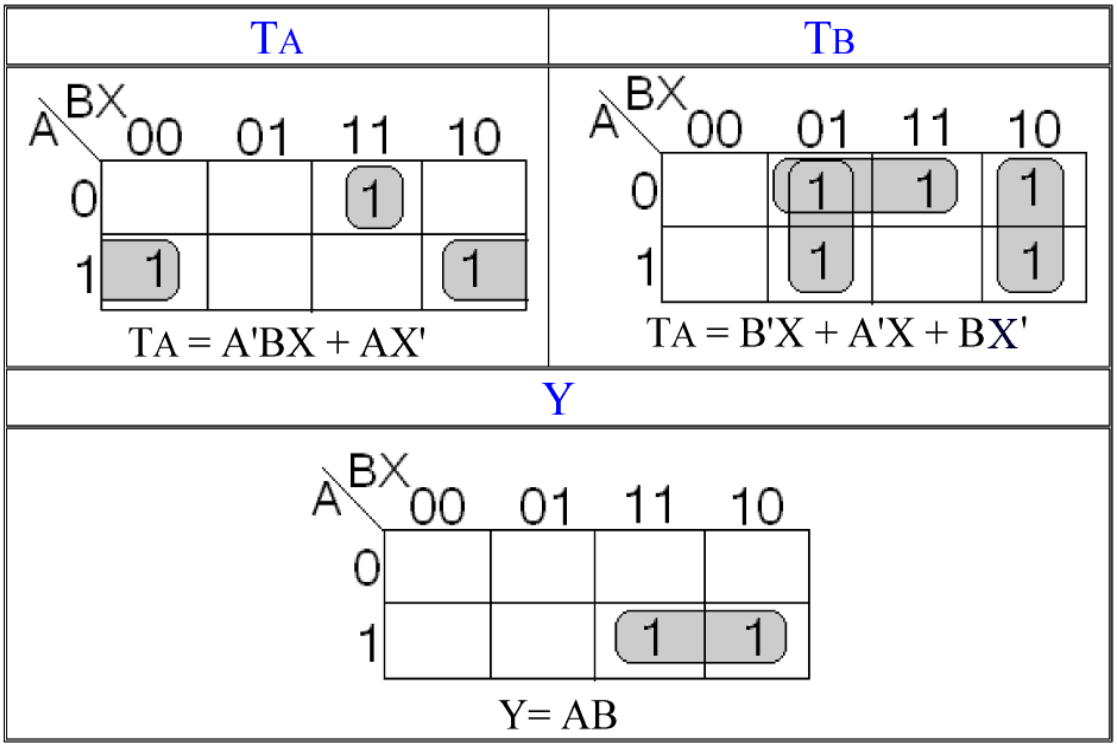


حل المثال بـ T Flip flop :

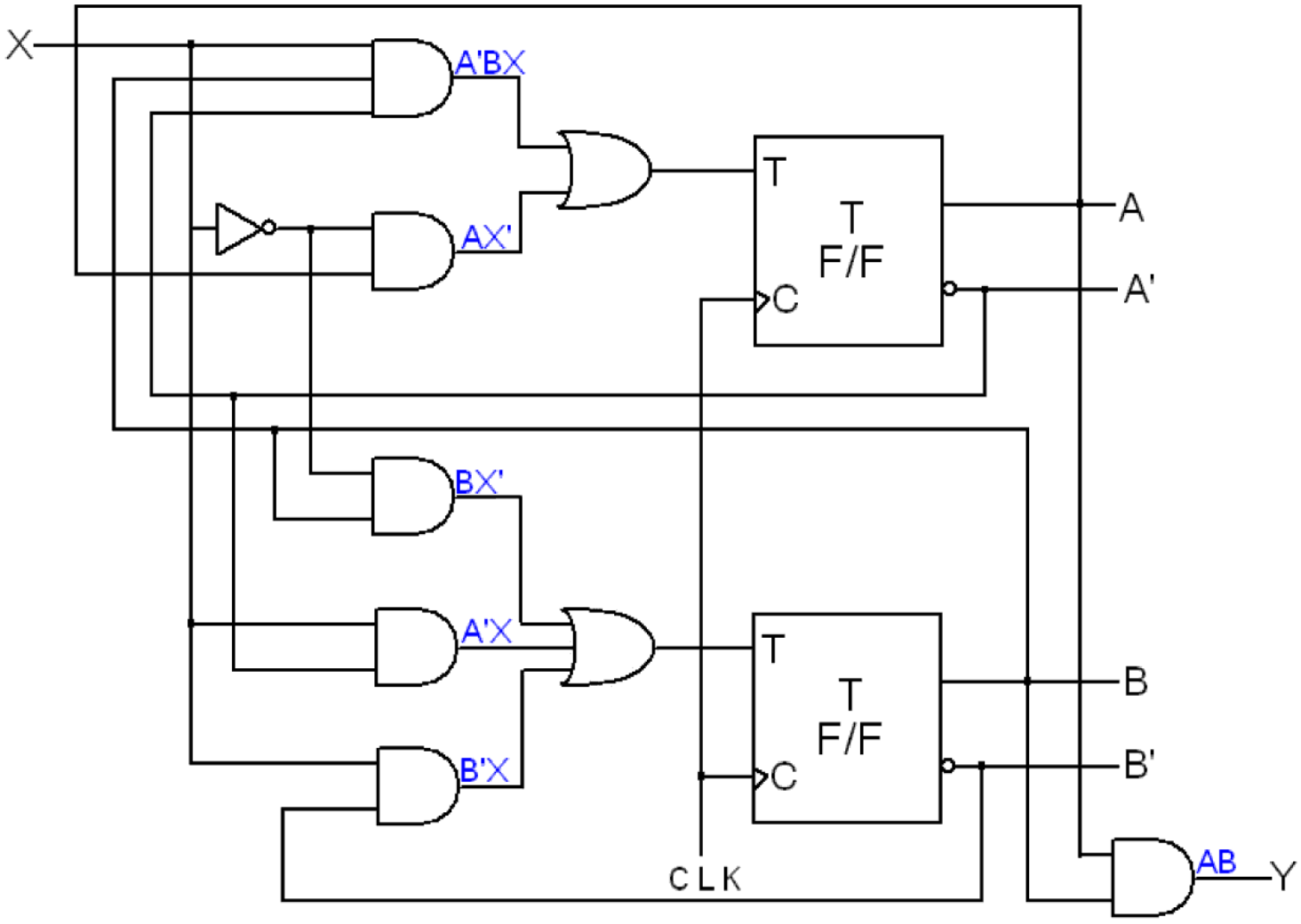
: State Table -1

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

: Simplify -2



: Diagram -3



5-6 حل أهم تمارين الفصل :

:Construct a JK Flip Flop using a D Flip Flop, a 2 to 1 line Multiplexer and an inverter /Q2

المطلوب :

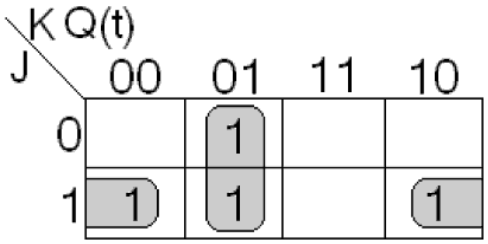
بناء JK Flip Flop باستخدام D Flip Flop و (2 to 1 line Multiplexer) و inverter

الحل :

1) Truth Table :

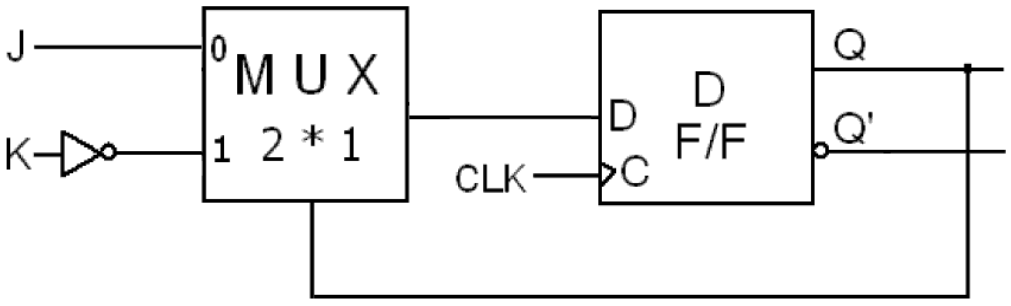
J	K	Q(t)	Q(t+1)	D
0	0	0	0	0
0	0	1	1	1
0	1	0	0	0
0	1	1	0	0
1	0	0	1	1
1	0	1	1	1
1	1	0	1	1
1	1	1	0	0

2) Simplify :

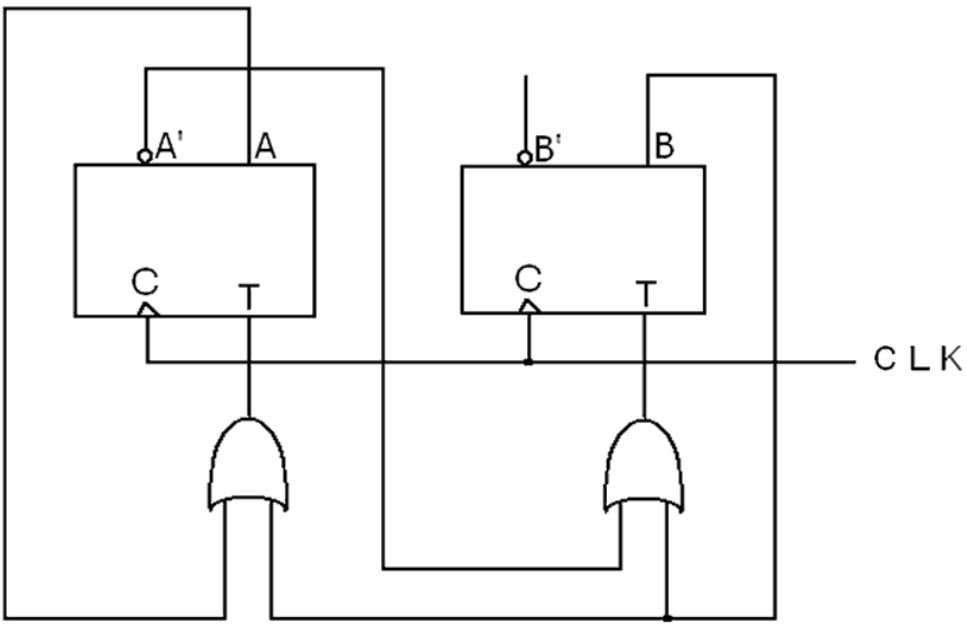


D = JQ' + K'Q

3) Diagram :



: Derive the Stats Table and the Stats Diagram of the Sequential Circuit /Q8



Explain the function that the Circuit performs

الحل :

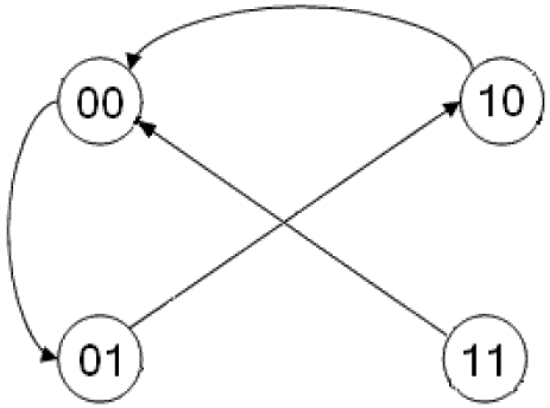
: Stats Table (1

$$A(t+1) = T_A \oplus A$$
$$= A+B \oplus A$$

$$B(t+1) = T_B \oplus B$$
$$= A'+B \oplus B$$

P.S		N.S	
A	B	A	B
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	0

: Stats Diagram (2



/Q9 : A Sequential Circuit has 2 JK Flip Flops A and B and 1 input X
The Circuit is described by the following Flip Flop input equations :

$J_A = X \qquad K_A = B'$
$J_B = X \qquad K_B = A$

المطلوب :
إيجاد Stats Table ورسم Stats Siagram لدائرة JK flip flop مكوّنة من A,B ومدخلاتها X
حسب القيم المعطاه لنا في السؤال

الحل :

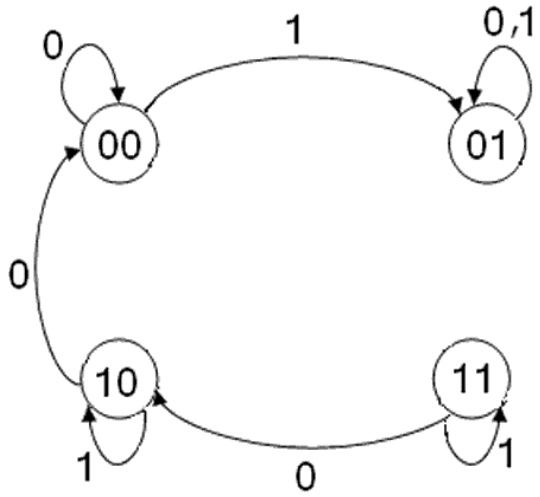
1) Stats Table :

$A(t+1) = J_A A' + K_A'A$
$= XA' + (B')'A$
$= XA + AB$

$B(t+1) = JBB + KA'B$
$= XB' + A'B$

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

2) Stats Diagram :



: Reduce the number of states in the following State Table and tabulate the reduced State Table /Q12

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

المطلوب :

تبسيط جدول State Table إلى أبسط صورة ممكنة

الحل :

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

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

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

/Q16 Design a Sequential Circuit with 2 D Flip Flops A and B and 1 input X :
When X = 0, the state of the Circuit remains the same When x=1, the Circuit goes
Through the state transitions form 00 to 01 to 11 to 10 back to 00 and repeats

المطلوب :

تصميم D Flip Flops مكوّن من A,B ومدخلاته X

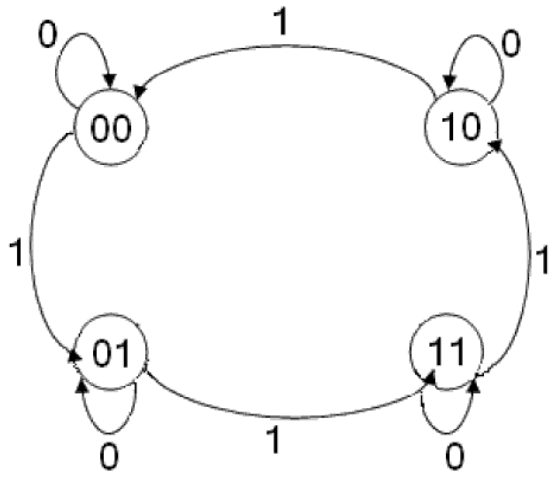
إذا كان X=0 فإن قيمة N.S = قيمة P.S بدون أي تغيير

وإذا كانت X=1 فإن قيمة N.S تسير حسب التسلسل المعطى :

Form 00 to 01 to 11 to 10 back to 00

الحل :

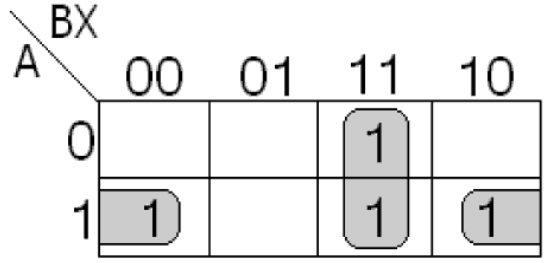
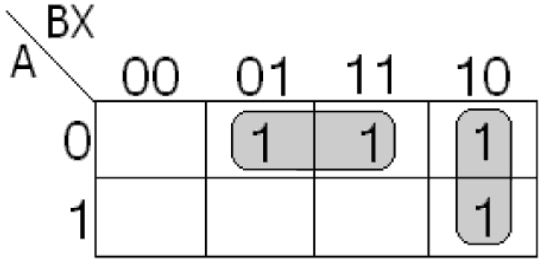
1) Stats Diagram :



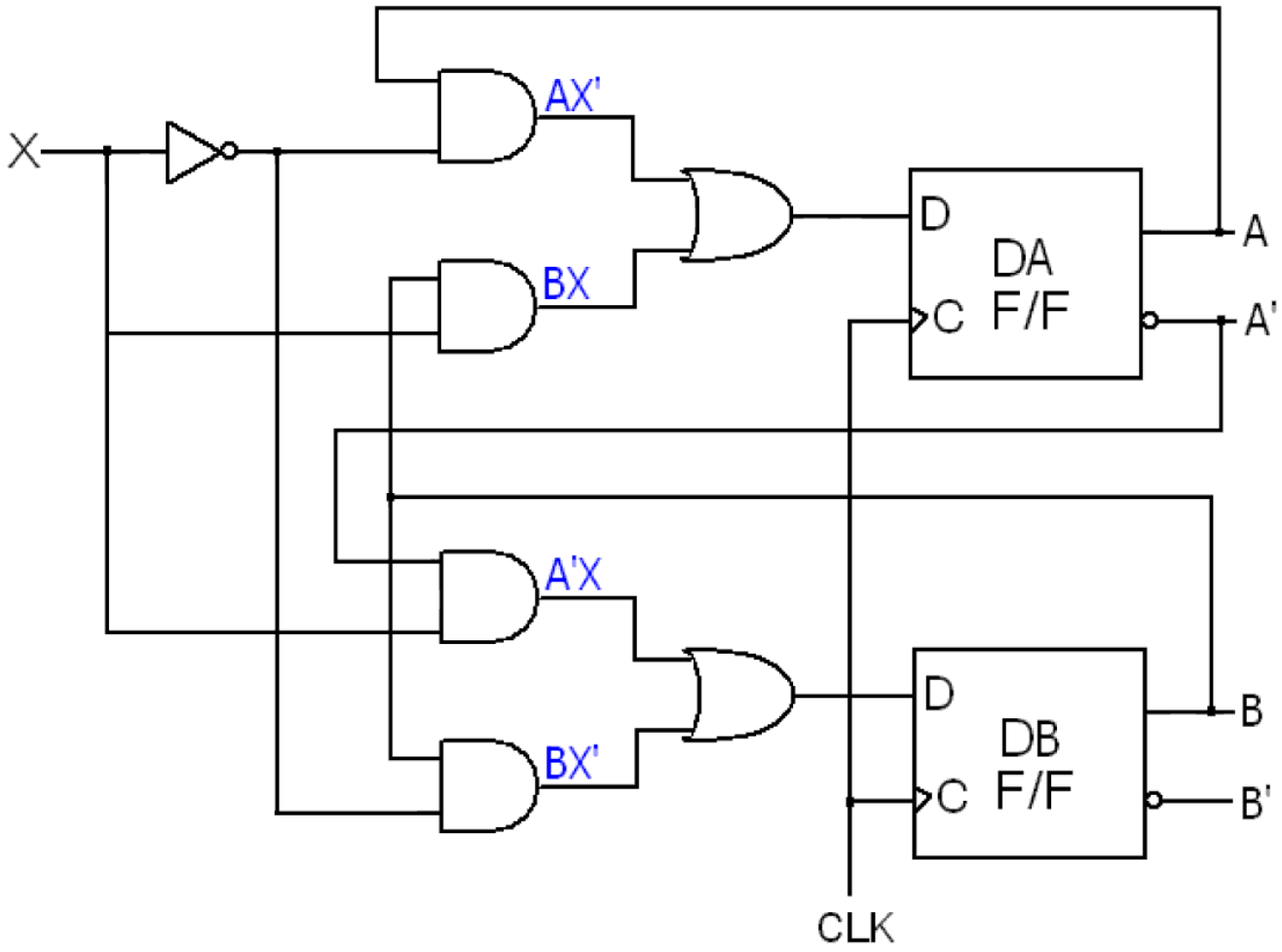
2) Stats Table :

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

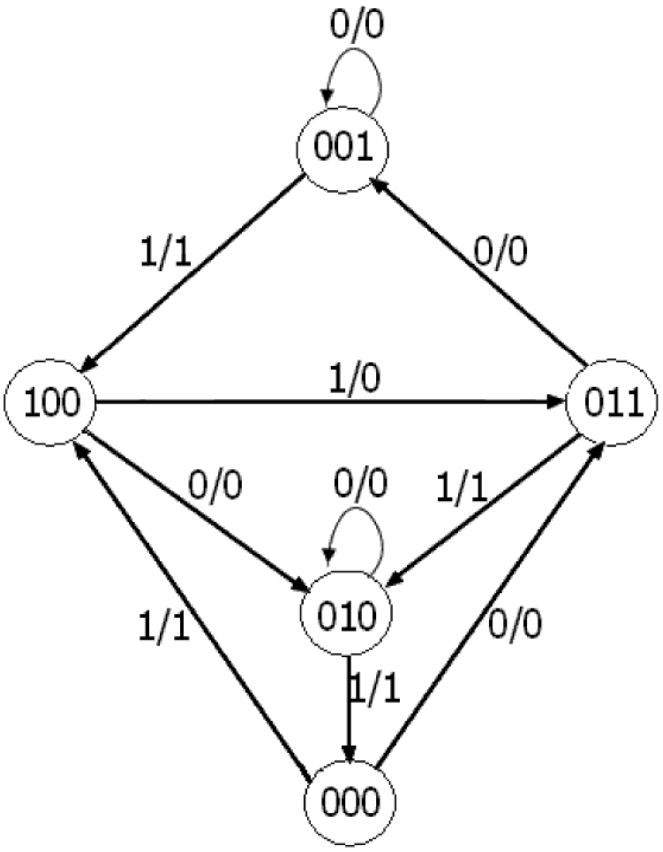
: Simplify (3

D_A	D_B
	
$D_A = AX' + BX$	$D_B = A'X + BX'$

: Diagram (4



: A Sequential Circuit has 3 Flip Flops A,B,C , 1 input X , and 1 output Y /Q19
The Circuit is to be designed by treating the unused states as Don't Care conditions
Analyze the Circuit obtained form the Design to determine
The effect of the unused states
(a) Use D Flip Flops in the Design , (b) Use JK Flip Flops in the Design



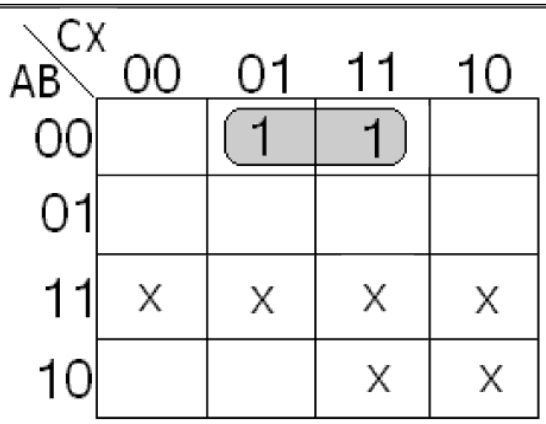
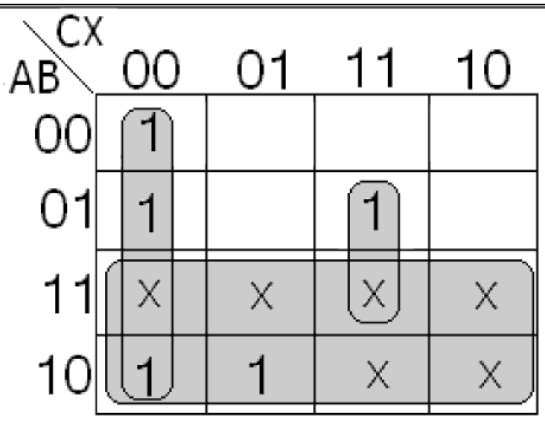
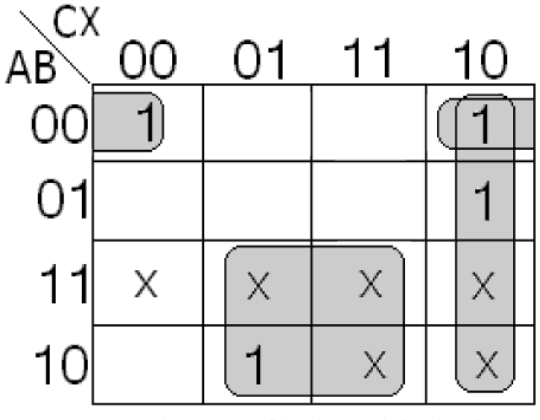
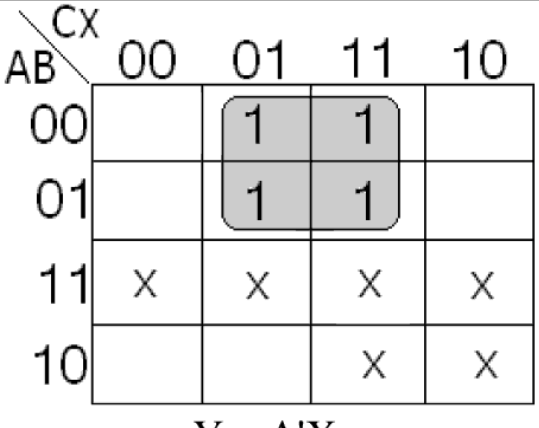
المطلوب :
تصميم Flip Flops مكوّن من A,B,C ومدخلاته X
وذلك من خلال تحليل رسمة Stats Diagram
في الفقرة (a) باستخدام D Flip Flops , وفي الفقرة (b) باستخدام JK Flip Flops

(a) Use D Flip Flops in the Design

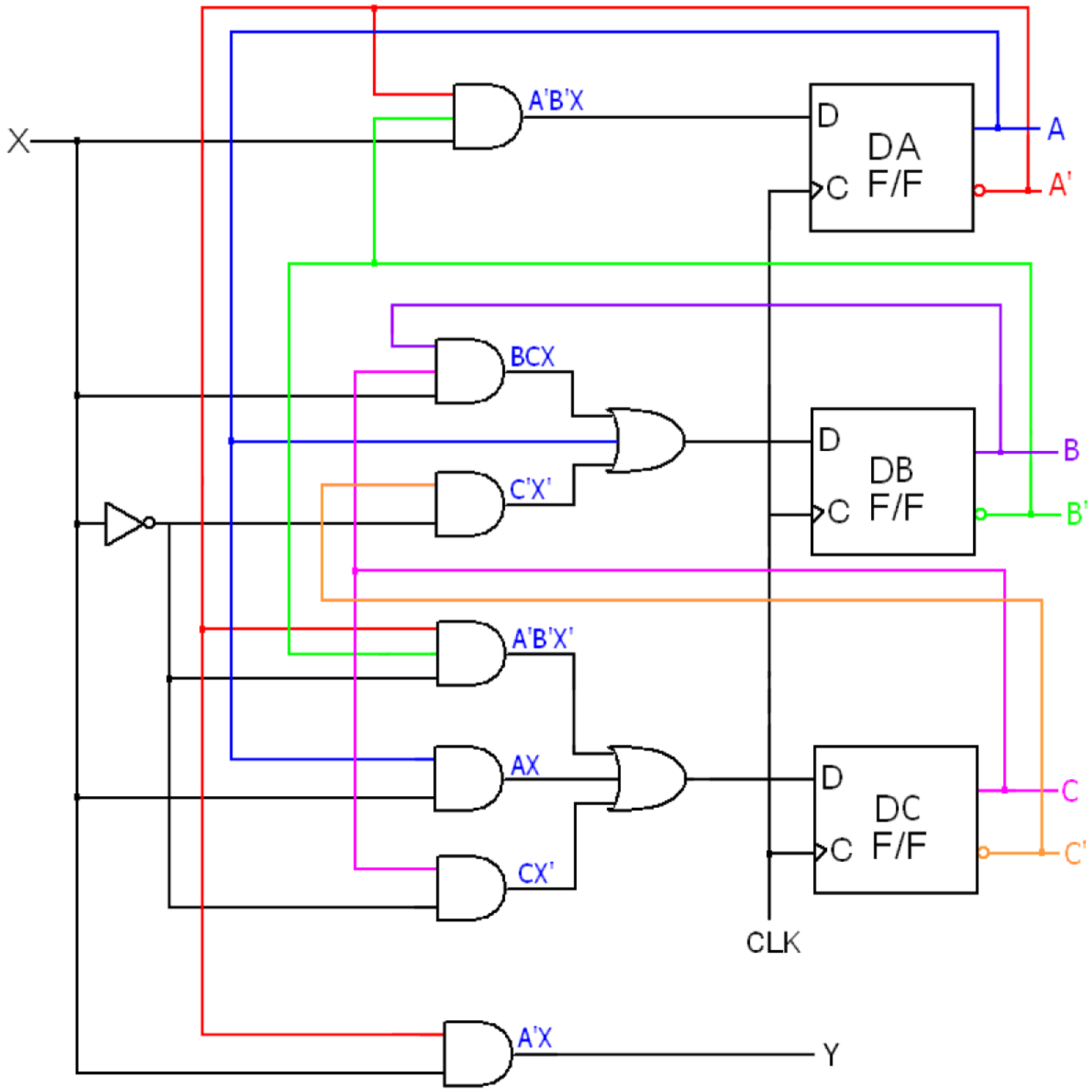
الحل :

1) Stats Table :

P.S			I/P	N.S			O/P			
A	B	C	X	A	B	C	Y	D_A	D_B	D_C
0	0	0	0	0	1	1	0	0	1	1
0	0	0	1	1	0	0	1	1	0	0
0	0	1	0	0	0	1	0	0	0	1
0	0	1	1	1	0	0	1	1	0	0
0	1	0	0	0	1	0	0	0	1	0
0	1	0	1	0	0	0	1	0	0	0
0	1	1	0	0	0	1	0	0	0	1
0	1	1	1	0	1	0	1	0	1	0
1	0	0	0	0	1	0	0	0	1	0
1	0	0	1	0	1	1	0	0	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

D_A	D_B
 $D_A = A'B'X$	 $D_B = C'X' + A + BCX$
D_C	Y
 $D_C = AX + CX' + A'B'X'$	 $Y = A'X$

: Diagram (3



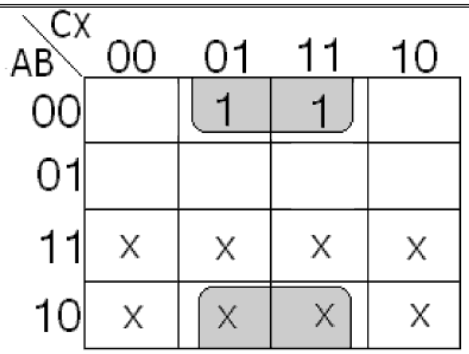
الحل :

: Stats Table (1

P.S			I/P	N.S			O/P						
A	B	C	X	A	B	C	Y	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	1	0	0	x	1	x	1	x
0	0	0	1	1	0	0	1	1	x	0	x	0	x
0	0	1	0	0	0	1	0	0	x	0	x	x	0
0	0	1	1	1	0	0	1	1	x	0	x	x	1
0	1	0	0	0	1	0	0	0	x	x	0	0	x
0	1	0	1	0	0	0	1	0	x	x	1	0	x
0	1	1	0	0	0	1	0	0	x	x	1	x	0
0	1	1	1	0	1	0	1	0	x	x	0	x	1
1	0	0	0	0	1	0	0	x	1	1	x	0	x
1	0	0	1	0	1	1	0	x	1	1	x	1	x
1	0	1	0	x	x	x	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x	x	x	x

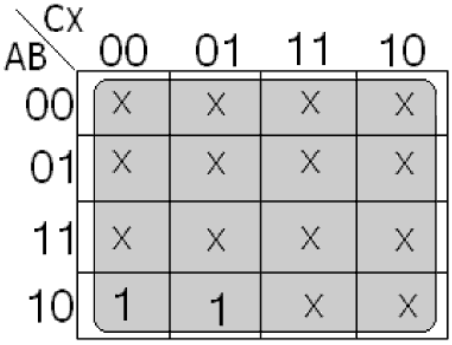
: Simplify (2

J_A



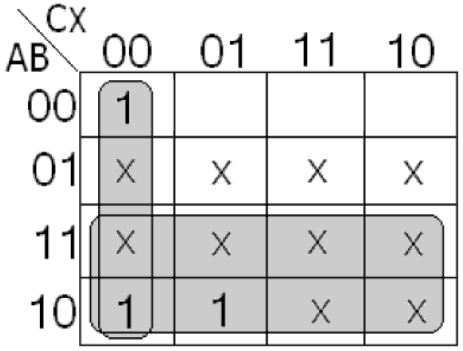
$J_A = B'X$

K_A



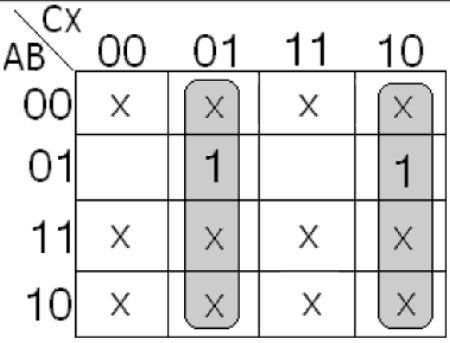
$K_A = 1$

J_B



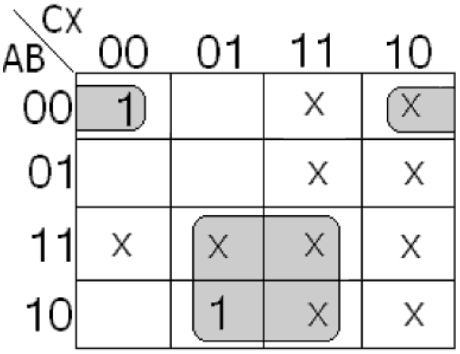
$J_B = A + C'X'$

K_B



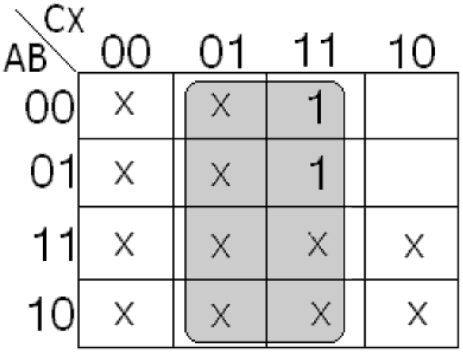
$K_B = C'X + CX' = C \oplus X$

J_C



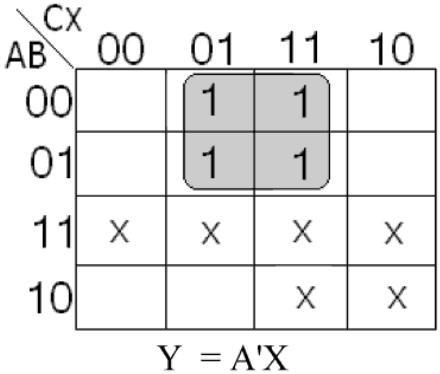
$J_C = AX + A'B'X'$

K_C



$K_C = X$

Y



$Y = A'X$

: Diagram (3

أدعها لك عزيزي القارئ

: Design the Sequential Circuit specified the State Diagram using T Flip Flops /Q20

المطلوب :

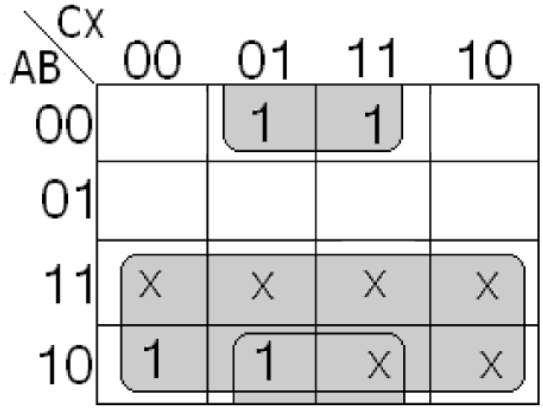
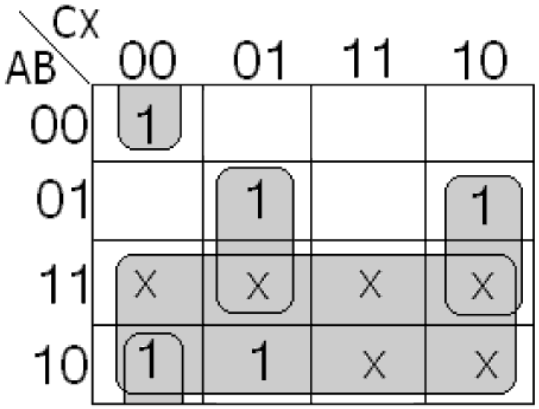
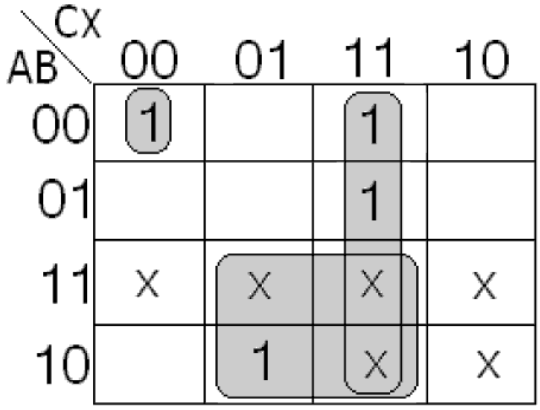
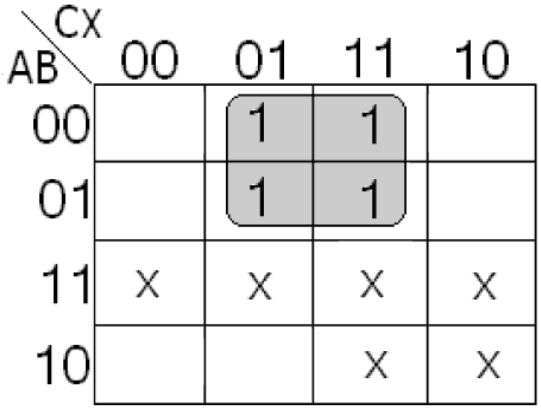
تصميم T Flip Flops من خلال تحليل رسمة Stats Diagram والتي مرت بنا في سؤال 19

الحل :

: Stats Table (1

P.S			I/P	N.S			O/P			
A	B	C	X	A	B	C	Y	T_A	T_B	T_C
0	0	0	0	0	1	1	0	0	1	1
0	0	0	1	1	0	0	1	1	0	0
0	0	1	0	0	0	1	0	0	0	0
0	0	1	1	1	0	0	1	1	0	1
0	1	0	0	0	1	0	0	0	0	0
0	1	0	1	0	0	0	1	0	1	0
0	1	1	0	0	0	1	0	0	1	0
0	1	1	1	0	1	0	1	0	0	1
1	0	0	0	0	1	0	0	1	1	0
1	0	0	1	0	1	1	0	1	1	1
1	0	1	0	x	x	x	x	x	x	x
1	0	1	1	x	x	x	x	x	x	x
1	1	0	0	x	x	x	x	x	x	x
1	1	0	1	x	x	x	x	x	x	x
1	1	1	0	x	x	x	x	x	x	x
1	1	1	1	x	x	x	x	x	x	x

: Simplify (2

T_A	T_B
	
$T_A = A + B'X$	$T_B = A + B'C'X' + BC'X + BCX'$
T_C	Y
	
$T_C = A'B'C'X' + CX + AX$	$Y = A'X$

: Diagram (3

أدعها لك عزيزي القارئ

الفصل السادس

Registers And Counters

6-1 مقدمة:

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

6-2 المسجل Register :

الشكل العام لـ Register :

A7	A6	A5	A4	A3	A2	A1	A0

1- Shift Register :

عملية Shift عبارة عن عملية إزاحة لقيم Register إما إلى جهة اليمين أو إلى جهة اليسار
حيث نقوم بإخراج القيمة التي تقع في الخانة الأخيرة في Register خارج Register
وتحل محل القيمة الأخيرة القيمة التي قبلها (قبل الأخيرة) , وتحل محل القيمة التي قبل الأخيرة
القيمة التي قبلها .. وهكذا إلى أن نصل إلى أول خانة من Register وقد أصبحت خالية فنضع فيها القيمة (0)

إذا كانت عملية Shift إلى جهة اليسار :
فإن الخانة الأولى A0 والخانة الأخيرة هي A7
أما إذا كانت عملية Shift إلى جهة اليمين :
فإن الخانة الأولى A7 والخانة الأخيرة هي A0

مثال :

Shift left R :

R

1	1	0	1

الحل :

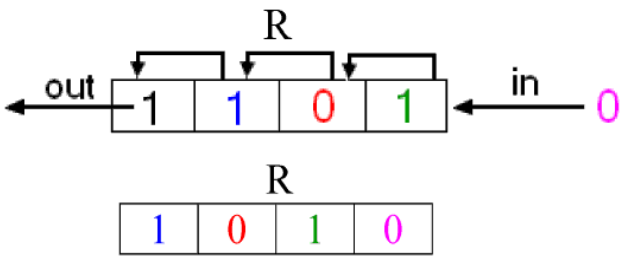
المطلوب عمل Shift إلى جهة اليسار لقيم Register R

R

1	0	1	0

الشرح :

كما تلاحظ عزيزي القارئ أن القيم ملونة وذلك ليسهل عليك ملاحظة ماحدث من تغيير حيث حدث ما يلي :
أولاً خرجت القيمة الواقعة في الخانة A3 خارج Register وهي القيمة (1)
ثم انتقلت القيمة الواقعة في الخانة A2 إلى الخانة A3 وهي القيمة (1)
ثم انتقلت القيمة الواقعة في الخانة A1 إلى الخانة A2 وهي القيمة (0)
ثم انتقلت القيمة الواقعة في الخانة A0 إلى الخانة A1 وهي القيمة (1)
أصبحت الآن الخانة A0 خالية فنضع فيها القيمة (0) , كما هو موضح في الشكل التالي :



مثال :

: Shift Right R

R

0	1	0	1

الحل :

المطلوب عمل Shift إلى جهة اليمين لقيم R Register

R

0	0	1	0

2- Rotate Register :

عملية Rotate عملها مثل عمل Shift

ولكن الاختلاف أن Shift تخرج القيمة التي في الخانة الأخيرة من Register خارج Register وتضع في الخانة الأولى القيمة (0)

أما Rotate فإنها لا تخرج القيمة التي في الخانة الأخيرة من Register خارج Register وإنما تضعها في الخانة الأولى والتي أصبحت خالية نتيجة عملية انتقال كل قيمة للخانة التي بعدها

مثال :

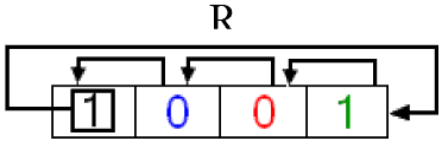
: Rotate left R

R

1	0	0	1

الحل :

المطلوب عمل Rotate إلى جهة اليسار لقيم R Register



R

0	0	1	1

مثال:

: Rotate Right R

R

0	1	0	1

الحل:

المطلوب عمل Rotate إلى جهة اليمين لقيم R Register

R

1	0	1	0

مثال:

: Rotate Right R 3 himes

R

1	1	0	0	0	1	0	1

الحل:

المطلوب عمل Rotate إلى جهة اليمين 3 مرات لقيم R Register

R

1	1	1	0	0	0	1	0	المرة الأولى
0	1	1	1	0	0	0	1	المرة الثانية
1	0	1	1	1	0	0	0	المرة الثالثة

مثال :

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 :

المطلوب عمل Shift إلى جهة اليسار 4 مرات لقيم A Register
ولكن في هذا المثال لن نضيف القيمة (0) في الخانة الأخيرة التي تكون خالية من القيم
إذن ماذا سوف نضع في الخانة الخالية ؟

نلاحظ في السؤال شيء جديد وهو serial input , ومن قيمه سوف نقوم بتعبئة الخانات الخالية

في المرة الأولى :
نقوم أولاً بعمل Shift إلى جهة اليسار وسوف تصبح قيم A Register بالشكل التالي :

1	0	1	0	1	0	0	

ثم نضع في الخانة الخالية (A0) قيمة من serial input
ودائماً نبدأ باختيار قيم serial input من اليسار إلى اليمين
أي أننا سوف نضع آخر قيمة من قيم serial input في الخانة A0 وهي القيمة (1)
ويصبح A Register بالشكل التالي :

1	0	1	0	1	0	0	1

في المرة الثانية :
نقوم أيضاً بعمل Shift إلى جهة اليسار وسوف تصبح قيم A Register بالشكل التالي :

0	1	0	1	0	0	1	

ونضع القيمة قبل الأخيرة من قيم serial input في الخانة الخالية وهي القيمة (0)
ويصبح A Register بالشكل التالي :

0	1	0	1	0	0	1	0

وفي المرة الثالثة والرابعة نقوم بنفس الخطوات
ويصبح الشكل النهائي A Register بالشكل التالي :

1	1	0	0	1	0	1	1

وبالتالي نكون قد قمنا بعملية Shift لقيم A Register وهو المطلوب

وإليكم الآن شكل يوضح خطوات الحل كاملةً :

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

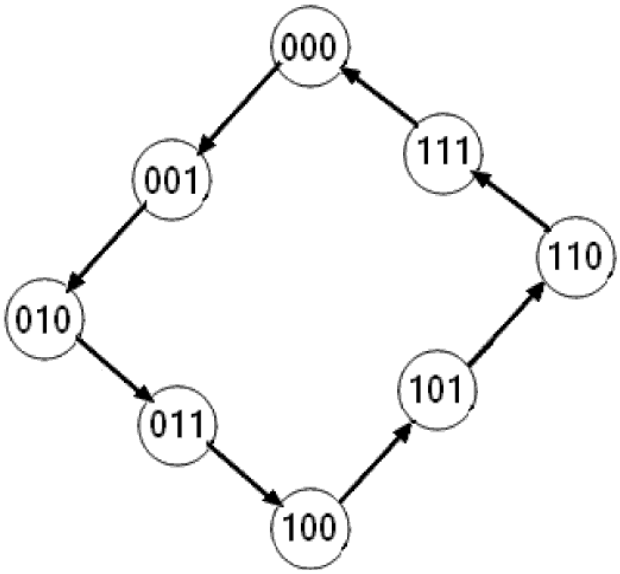
6-3 العداد Counter :

خطوات تصميم العداد (Counter) :

State Table -1 Simplify -2 Diagram -3

مثال:

: Design a 3-bit Counter using T Flip flop



الحل:

المطلوب تصميم عداد يستقبل (3-bit) باستخدام T Flip Flop

State Table -1 :

وفيها نقوم أولاً بتقسيم خلايا State Table

الشكل العام لتكوين أعمدة State Table لإنشاء عداد :

P.S			N.S			الـ Flip Flop المستخدم
A2	A1	A0	A2	A1	A0	

الآن ننشئ State Table لرسمة State Diagram للعداد السابق

نقوم أولاً بتحليل رسمة State Diagram :

نلاحظ أنها تتكون من 8 دوائر

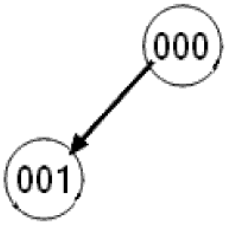
نقوم بوضع قيم هذه الدوائر في أعمدة (P.S) , ويصبح شكل State Table بالشكل التالي :

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

تابع الحل

نقوم الآن بتعبئة أعمدة (N.S)
ولتعبئتها نتتبع الأسهم من أين انطلقت وإلى أين وصلت
[من أي دائرة (P.S) انطلق وإلى أي دائرة (N.S) وصل]

على سبيل المثال نأخذ الجزء التالي من الرسمة ونقوم بتحليله لتعبئة حقول أعمدة (N.S) بناءً على التحليل :



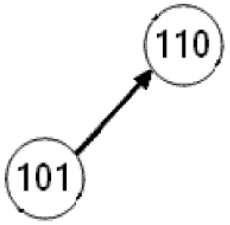
انطلق السهم من الدائرة (000) إلى الدائرة (001)
[P.S = 000 ⟶ N.S = 001]

نقوم الآن بتعبئة حقول State Table بناء على القيمة التي توصلنا إليها من تحليل رسمة State Diagram
ويصبح الجدول بالشكل التالي :

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

تابع الحل

كذلك على سبيل المثال نأخذ الجزء التالي من الرسمة ونقوم بتحليله لتعبئة حقول أعمدة (N.S) بناءً على التحليل :



انطلق السهم من الدائرة (101) إلى الدائرة (110)

[P.S = 101 ⟶ N.S = 110]

نقوم الآن بتعبئة حقول State Table بناء على القيمة التي توصلنا إليها من تحليل رسمة State Diagram ويصبح الجدول بالشكل التالي :

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

بعد ذلك نكمل تتبع باقي الأسهم بنفس الطريقة التي تتبعنا بها السهمين السابقين
إلى أن ننتهي من تتبع جميع الأسهم وتعبئة أعمدة (N.S)
ويصبح الجدول بالشكل التالي :

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

تابع الحل ⟵

وبالتالي نكون قد انتهينا من تعبئة العمود (P.S) وأعمدة (N.S)
وبقي علينا أعمدة (T Flip Flop)
ولتعبئة أعمدة (T Flip Flop) نعوض في Excitation Table الخاص بدائرة T Flip Flop
وهذا هو Excitation Table لتسهل علينا عملية التعويض :

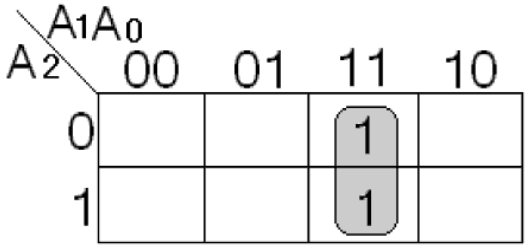
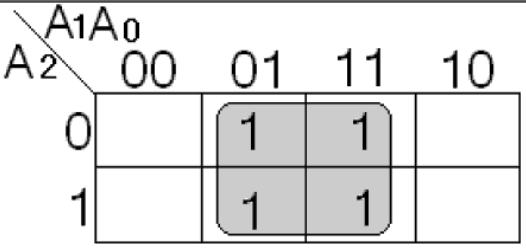
Q(t)	Q(t+1)	T
0	0	0
0	1	1
1	0	1
1	1	0

التعويض بالعمود A0 التابع لأعمدة (P.S) و العمود A0 التابع لأعمدة (N.S) يعطينا العمود TA0
والتعويض بالعمود A1 التابع لأعمدة (P.S) و العمود A1 التابع لأعمدة (N.S) يعطينا العمود TA1
والتعويض بالعمود A2 التابع لأعمدة (P.S) و العمود A2 التابع لأعمدة (N.S) يعطينا العمود TA2

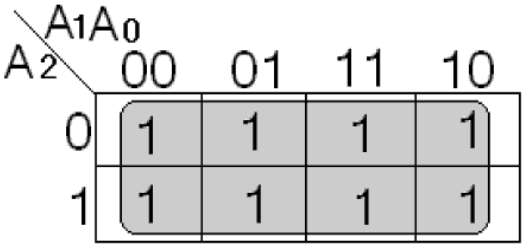
ويصبح الشكل النهائي لـ State Table بالشكل التالي :

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

: Simplify -2

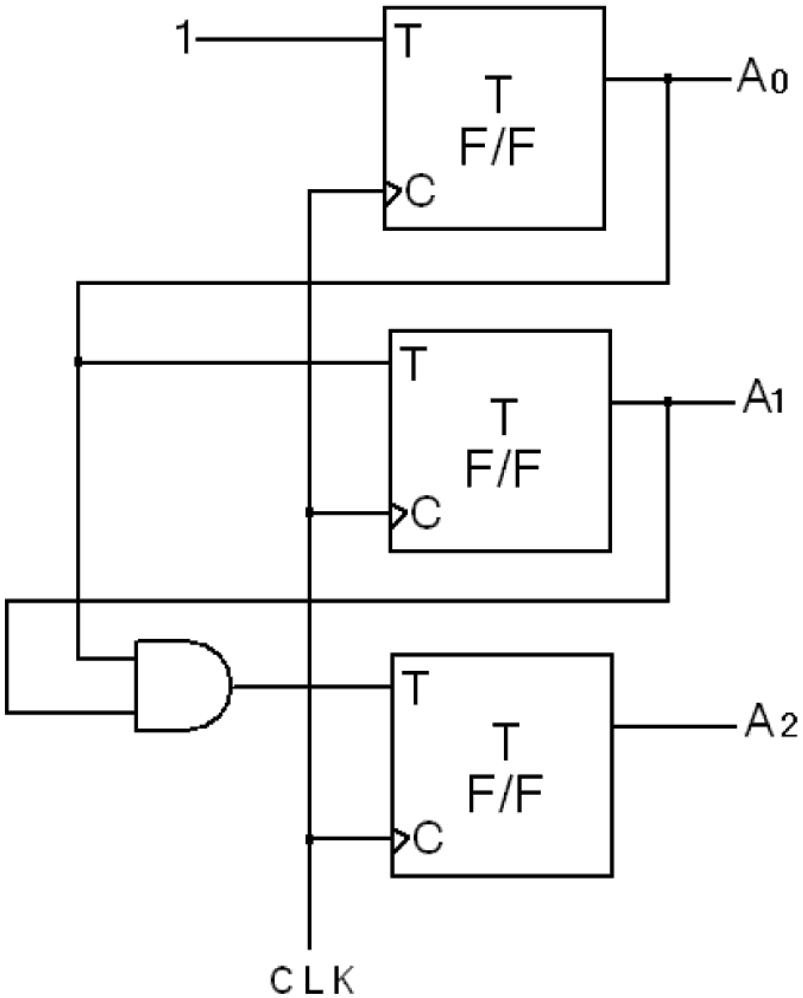
T_{A2}	T_{A1}
	
$T_{A2} = A_1A_0$	$T_{A1} = A_0$

T_{A0}



$T_{A0} = 1$

: Diagram -3



مثال :

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

الحل :

المطلوب تصميم عداد باستخدام T Flip Flop

حسب تسلسل القيم المعطى وهو :

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

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 في حالة (P.S) التي لا يوجد لها قيم تنتقل إليها في حالة (N.S) وهي القيمتين :

3 = (011) و 7 = (111)

فإن انتقالها سوف يكون إلى Don't Care

أي أن انتقال القيمتين (011) و (111) سوف يصبح بالشكل التالي :

011 ⟶ XXX

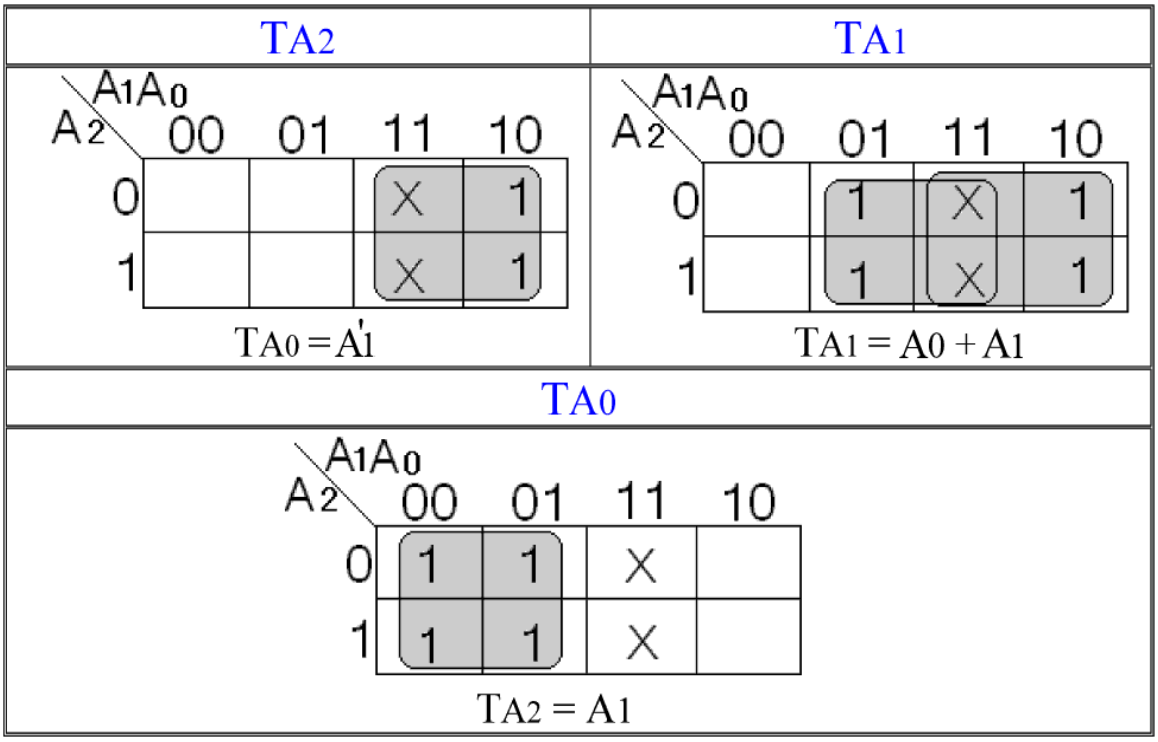
111 ⟶ XXX

أصبحت الآن الصورة واضحة لدينا ونقوم الآن بتعبئة State Table وسوف يصبح بالشكل التالي :

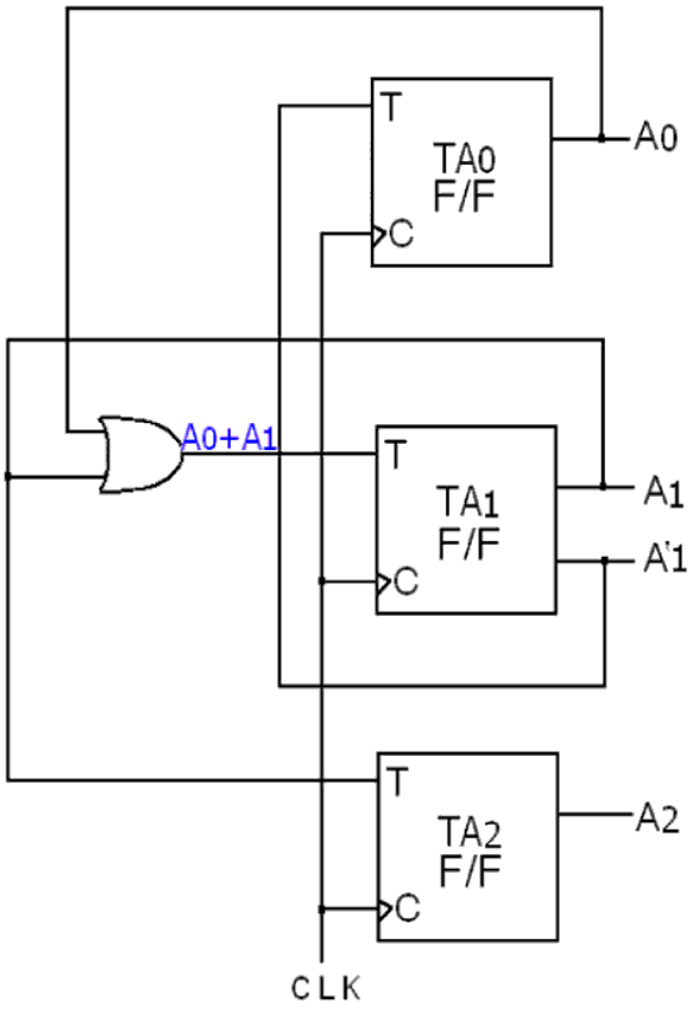
: State Table -1

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

: Simplify -2



: Diagram -3



6-4 حل أهم تمارين الفصل :

/Q4 The content of a 4-bit Register is initially 1101 The Register is Shifted 6 times :
To the Right with the Serial being 101101 What is the content of the Register
After each Shift?

المطلوب :

عمل Shift إلى جهة اليمين 6 مرات لقيم Register

الحل :

The content of a 4-bit Register is initially 1101

Serial input = 101101

1101

1110	-1	0111	-2
1011	-3	1101	-4
0110	-5	1011	-6

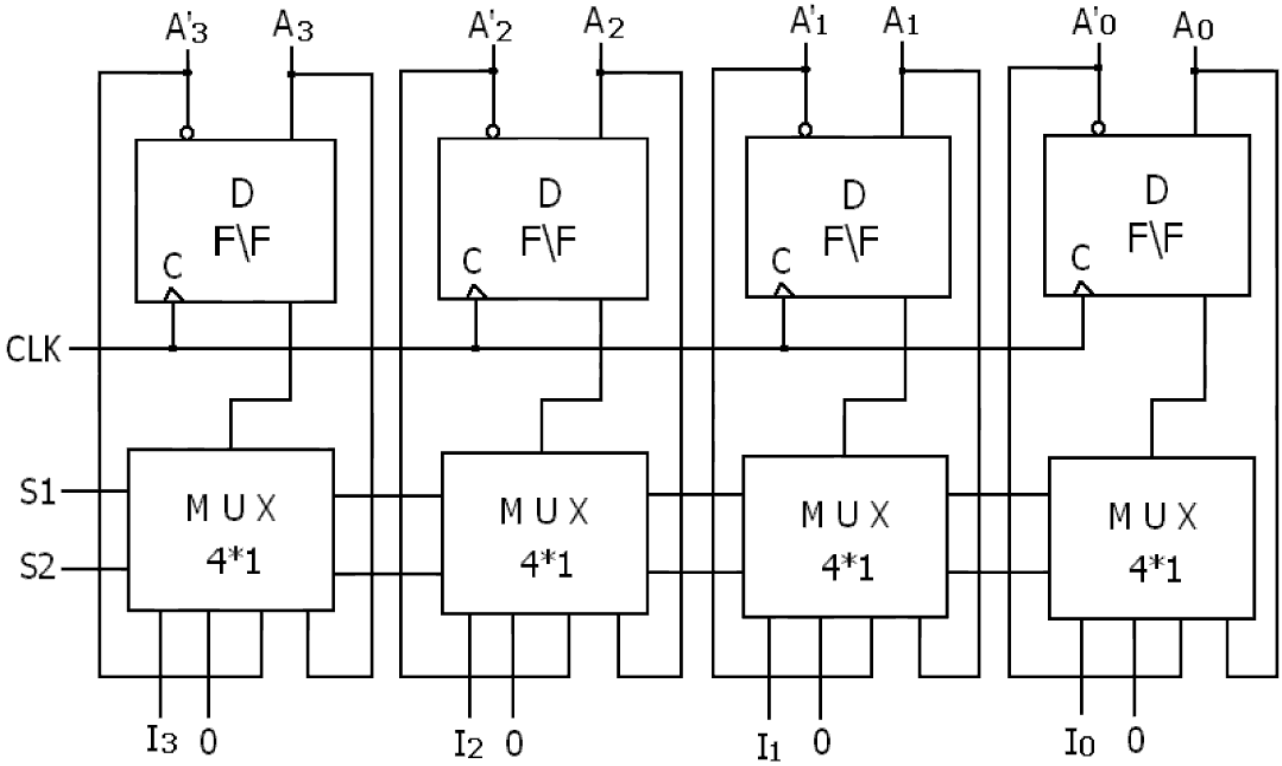
/Q7 Draw the Logic Diagram of a 4-bit Register with 4 D Flip Flops :
And 4 Multiplexers with mode Selection inputs s_1 and s_0
The Register operates according to the following function table

S_1	S_0	Register Operation
0	0	No change
0	1	Complement the 4 outputs
1	0	Clear Register to 0 (synchronous with the clock)
1	1	Load parallel data

المطلوب :

رسم Logic Diagram يستقبل 4-bit باستخدام 4 (D flip flops) و 4 (Multiplexers)

الحل :



: Design a Counter with T Flip Flops that goes through the following Binary repeated /Q24
Sequence : 0,1,3,7,6,4 Show that when Binary states 010 and 101 are considered as
Don't Care conditions the counter may not operate properly Find a way to correct the Design

المطلوب :

تصميم عداد باستخدام T Flip Flops حسب التسلسل المعطى
وإذا وجدت أعداد خارج التسلسل وهي الأعداد (2) = (010) و (5) = (101)
يكون انتقالها إلى Don't Care

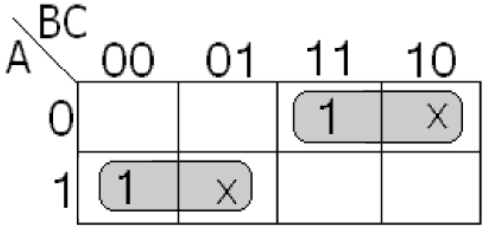
الحل :

: Stats Table (1

P.S			N.S					
A	B	C	A	B	C	T_A	T_B	T_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	1	0	1	0
0	1	0	x	x	x	x	x	x
0	1	1	1	1	1	1	0	0
1	0	0	0	0	0	1	0	0
1	0	1	x	x	x	x	x	x
1	1	0	1	0	0	0	1	0
1	1	1	1	1	0	0	0	1

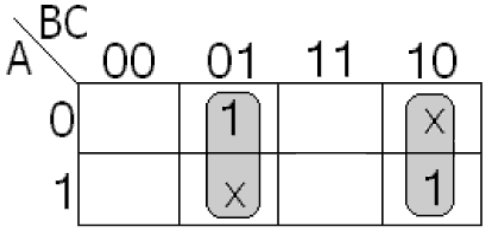
: Simplify (2

T_A



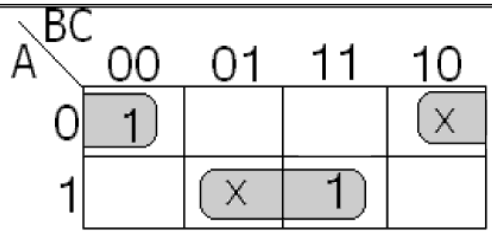
$T_A = AB' + A'B = A \oplus B$

T_B



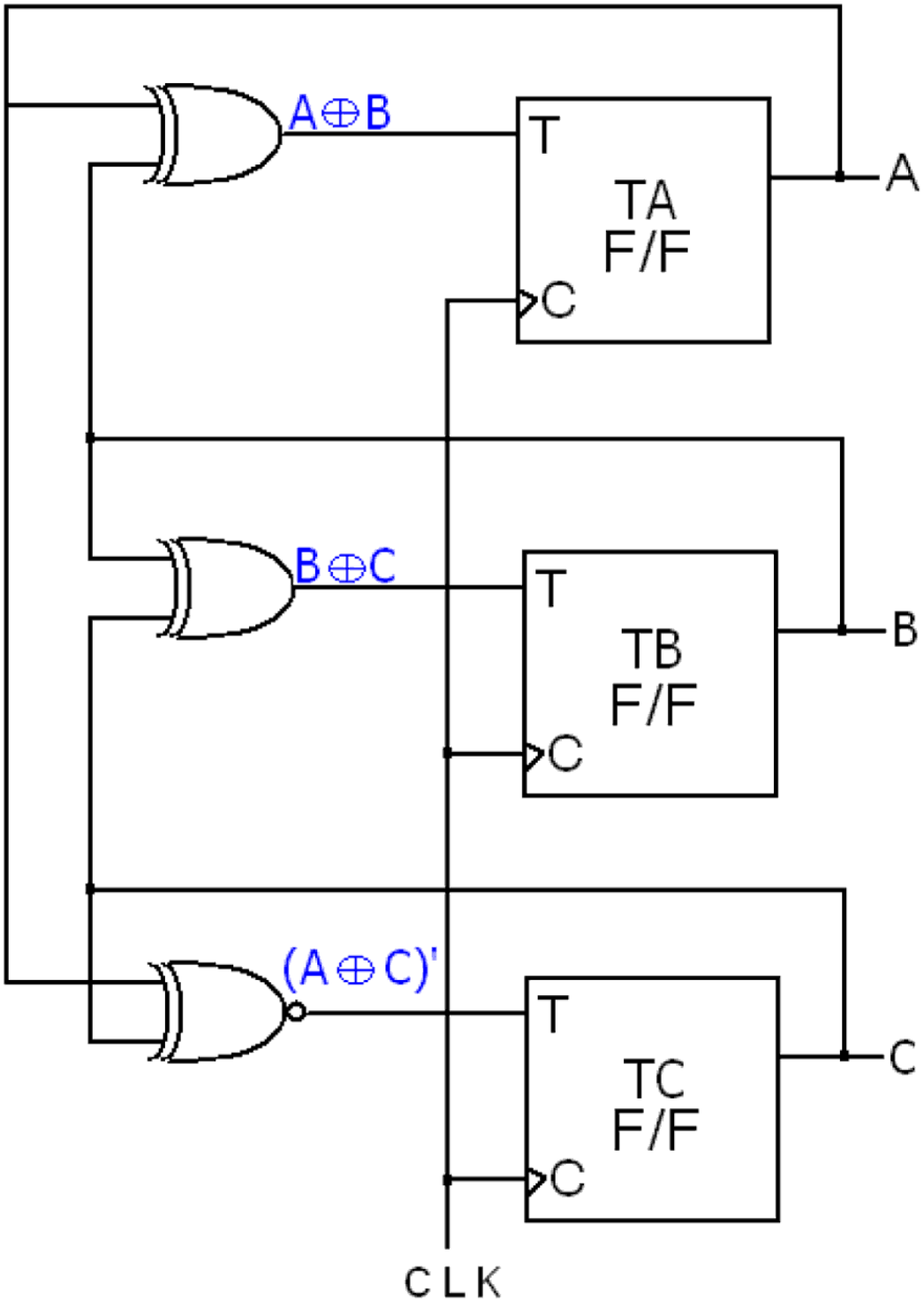
$T_B = BC' + B'C = B \oplus C$

T_C



$T_C = AC + A'C' = (A \oplus C)'$

: Diagram (3



: Design a Counter with the following repeated Binary Sequence:0,1,2,3,4,5,6 /Q27
Use JK Flip Flops

المطلوب :

تصميم عداد حسب التسلسل المعطى باستخدام JK Flip Flops

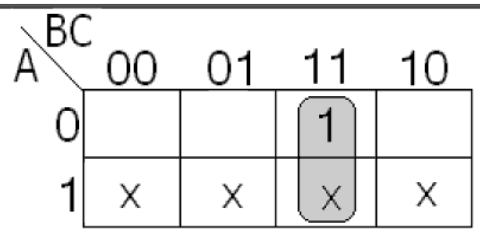
الحل :

: Stats Table (1

P.S			N.S								
A	B	C	A	B	C	J_A	K_A	J_B	K_B	J_C	K_C
0	0	0	0	0	1	0	x	0	x	1	x
0	0	1	0	1	0	0	x	1	x	x	1
0	1	0	0	1	1	0	x	x	0	1	x
0	1	1	1	0	0	1	x	x	1	x	1
1	0	0	1	0	1	x	0	0	x	1	x
1	0	1	1	1	0	x	0	1	x	x	1
1	1	0	0	0	0	x	1	x	1	0	x
1	1	1	x	x	x	x	x	x	x	x	x

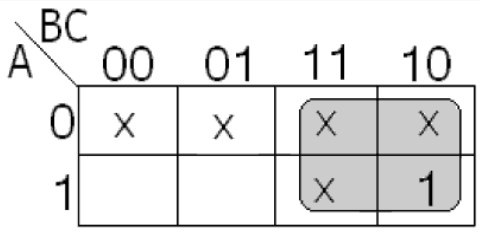
: Simplify (2

J_A



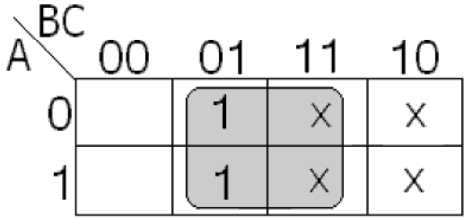
$J_A = BC$

K_A



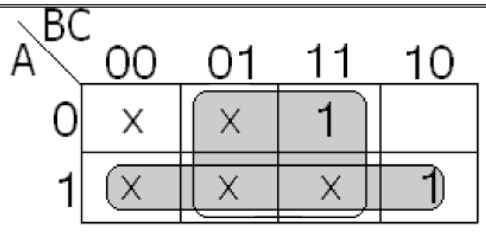
$K_A = B$

J_B



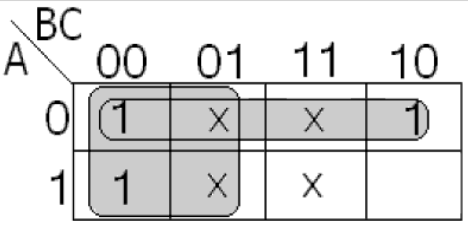
$J_B = C$

K_B



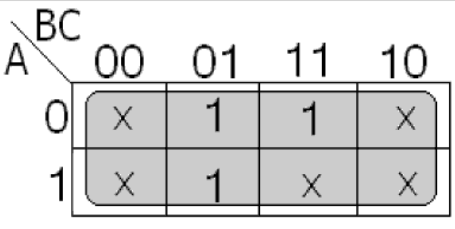
$K_B = A + C$

J_C



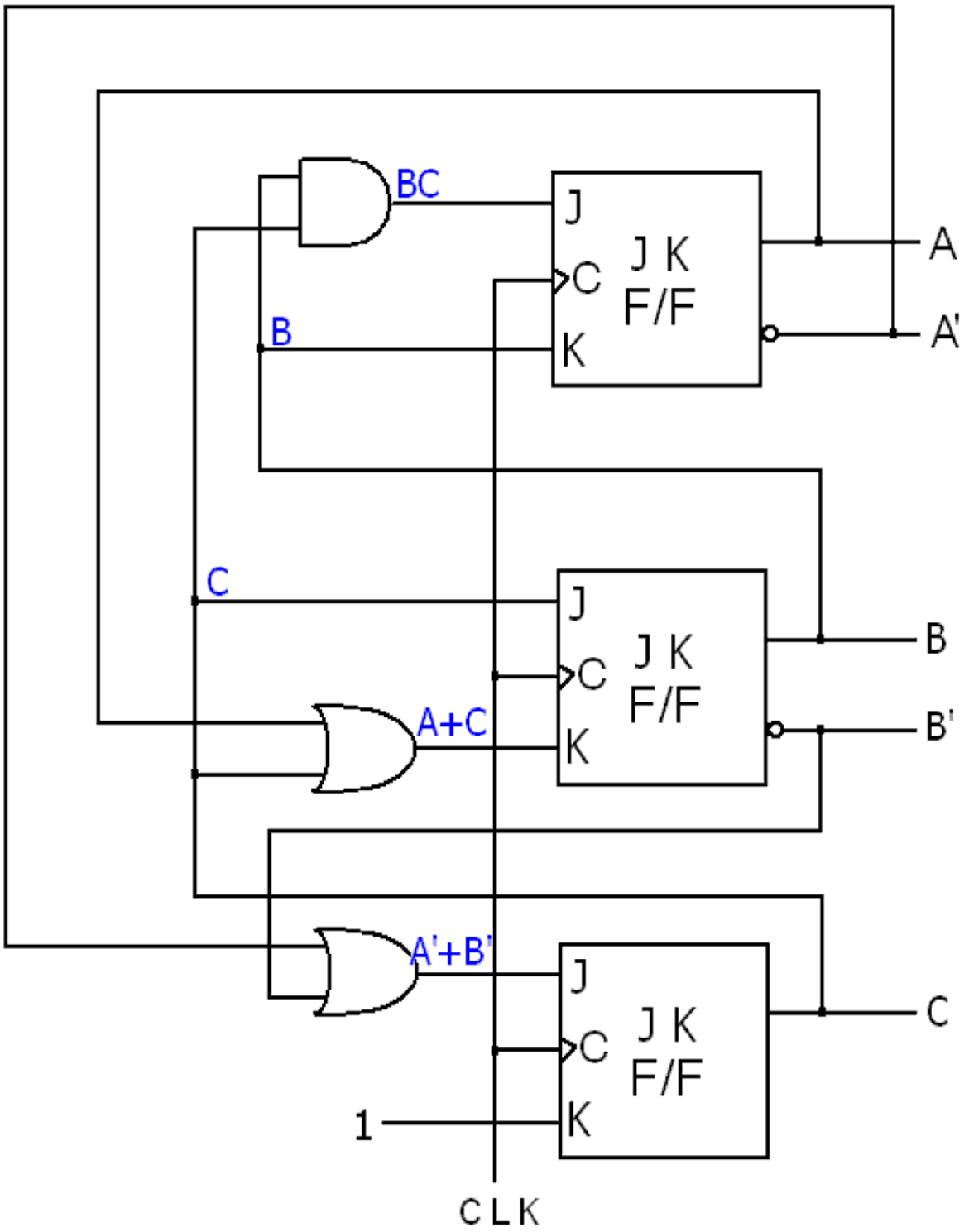
$J_C = A' + B'$

K_C



$K_C = 1$

: Diagram (3



: Design a Counter with the following repeated Binary Sequence:0,1,2,4,6 /Q28
Use D Flip Flops

المطلوب :

تصميم عداد حسب التسلسل المعطى باستخدام D Flip Flops

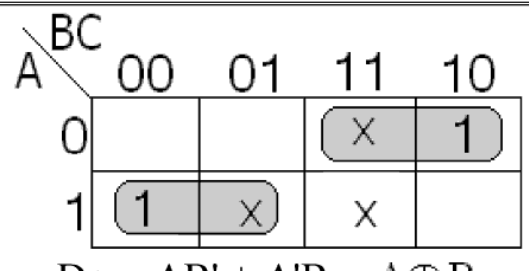
الحل :

: Stats Table (1

P.S			N.S					
A	B	C	A	B	C	D_A	D_B	D_C
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	1	0	0	1	0	0
0	1	1	x	x	x	x	x	x
1	0	0	1	1	0	1	1	0
1	0	1	x	x	x	x	x	x
1	1	0	0	0	0	0	0	0
1	1	1	x	x	x	x	x	x

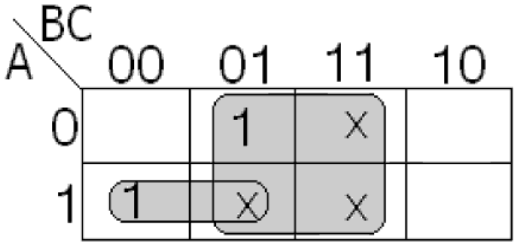
: Simplify (2

D_A



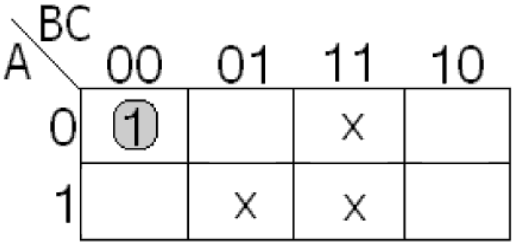
$D_A = AB' + A'B = A \oplus B$

D_B



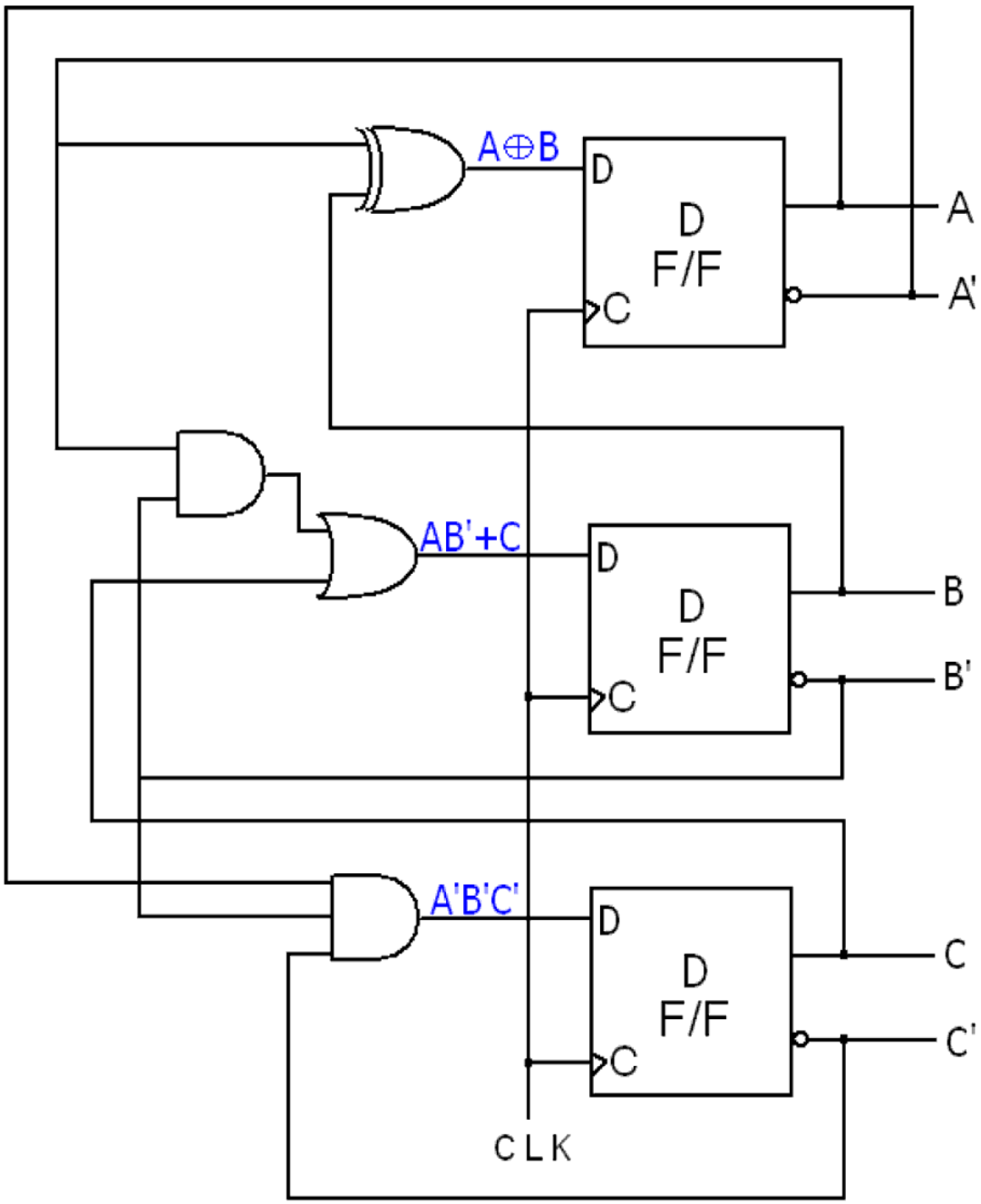
$D_B = AB' + C$

D_C



$D_C = A'B'C'$

: Diagram (3



الخاتمة

أسأل الله العلي القدير أن أكون وفقت في طرح ما يفيدكم

ويعينكم على فهم مادة التحليل والتصميم المنطقي

عند وجود ملاحظات واستفسارات واقتراحات حول هذا العمل

أرجوا أن لا تترددوا في إخباري عنها عبر البريد الإلكتروني المذكور

ختاماً أسأل الله العلي القدير التوفيق لي ولكم

وآخر دعوانا أن الحمد لله رب العالمين وصلى الله على سيدنا محمد وعلى آله وصحبه وسلم

والسلام عليكم ورحمة الله وبركاته

أحمد رمضان الزهراني
طالب في جامعة أم القرى
Ahmad_911@hotmail.com