بسم الله الرحمن الرحيم

جامعة السّودان المفتوحة

برنامج الحاسوب

أساسيات التصميم المنطقي

رمز المقرر ورقمه: حسب 2035

تأليف: د. يحيى عبد الله محمد

أ. أمير عبد الفتاح أحمد

تحكيم علمي : د. أحمد صلاح الدين عبد الله

تصميم تعليمي: د. عبد الباسط محمد شريف موسى

التدقيق اللغوي: أ. الهدي عبد الله محمد

التصميم الفني: منى عثمان أحمد النّقة

منشورات جامعة السودان المفتوحة، الطبعة الأولى 2007

مقدمة للمقرر

الحمد لله رب العالمين، والصلاة والسلام على رسوله الكريم، محمد صلى الله عليه وسلم وآله وصحبه، وبعد

يعتبر مقرر "أساسيات التصميم المنطقي" المدخل الأساس لفهم التركيب الداخلي للحاسوب، حيث يبدأ المقرر في وصف الدوائر البسيطة التي تعتبر الأساس لبناء الوحدات الرئيسة التي تكوّن الحاسوب، ثم يستمر في تناول الدوائر الأكثر تعقيداً.

يفترض أن يكون للدارس معرفة بالنظم الرقمية وبالذات النظام الثنائي، حيث يتناول هذا المقرر كيفية التعامل مع البيانات المختلفة في النظام الثنائي الذي يعتبر الأساس في بناء كل الدوائر المنطقية.

ولتوضيح أهمية هذا المقرر يجب على أي متخصص في الحاسوب فهم التركيب الداخلي للحاسوب لكي يحيط بكيفية تنفيذ برامجه، ومما لاشك فيه أن المبرمج الذي يفهم كيفية عمل الجهاز وكيفية تنفيذ برامجه فيه يستطيع أن ينتج برامج تستفيد من الإمكانات المتاحه في جهاز الحاسوب إستفادة قصوى مما يؤدي إلى إنتاج برامج تعمل بكفاءة عالية من ناحية السرعة واستغلال أقل الإمكانات لتنفيذ البرنامج، كما أنه يعلم بالتفصيل أي أخطاء متوقعه قد تحدث في برامجه وبالتالي يستطيع تجنبها.

يتضمن هذا المقرر ست وحدات، تناولت الوحدة الأولى تمثيل البيانات في الأنظمة الرقمية، حيث سندرس في هذه الوحدة الطريقة التي يتم بها تمثيل مختلف أنواع البيانات داخل جهاز الحاسوب، أو داخل الأنظمة الرقمية بصفة عامة، وهي تهدف إلى تزويد الدارس بمعرفة كيفية تعامل الحاسوب مع الأرقام والرموز.

أما الوحدة الثانية أساسيات الجبر البولياني فقد تناولت بعض المفاهيم الأساسية التي نحتاج إليها في دراستنا للأجزاء التالية من المقرر.

تناولت الوحدة الثالثة خطوات تصميم الدوائر المنطقية ابتداء من تحديد المواصفات، فكتابة التعبيرات المنطقية، فتبسيط تلك التعبيرات، ثم بناء الدائرة.

عرضت الوحدة الرابعة بعض الدوائر المنطقية الترابطية التي تقوم بأداء وظائف مفيدة، والتي يشيع استخدامها في الأنظمة الرقمية مثل الجامع بأنواعه وفاك الشفرة والمشفر والدامج والمفرق.

أما الوحدة الخامسة فقد وضحت المقصود بالدوائر المنطقية التتابعية، والفرق بينها وبين الدوائر المنطقية الترابطية، كما عرفت بعض أنواع الدوائر المنطقية التتابعية الشائعة الاستخدام في الأنظمة الرقمية.

أما الوحدة السادسة فقد تناولت تقنيات التخزين المختلفة في الأنظمة الرقمية ومميزات واستخدامات كل تقنية منها.

نأمل أن تقدم هذه الوحدات صورة واضحة عن أساسيات التصميم المنطقي، وأن تتيح لك عزيزي الدارس الفرصة لاكتساب المفاهيم والمهارات، مما يمكن لك مستقبلا فهم بقية المقررات التي تعتمد على هذا المقرر.

الأهداف العامة للمقرر

بعد فراغك من دراسة هذا المقرر يؤمل أن تكون لك القدرة على :

- **التعرف علي** كيفية تعامل الحاسوب مع الأرقام والرموز.
- **التعامل مع** المكونات الدقيقة للحاسوب.
- **تصميم وتبسيط** دوائر منطقية مفيدة.
- **التعرف على** أساسيات طريقة عمل وحدة الحساب والمنطق بالحاسوب.
- **بناء** دوائر منطقية باستخدام البوابات الأساسية الثلاث، أو باستخدام نوع واحد من البوابات.
- **المقارنة بين** الدوائر المنطقية الترابطية والدوائر المنطقية التتابعية.
- **إجادة التعامل مع** وحدات التخزين الأساسية الممثلة في المسجلات والعدادات.
- **التعرف على** الذاكرة المستخدمة في الحاسوب وتحديد خصائص كل أنواعها.

وختاما نرجو أن نكون قد وفقنا في تقديم إضافة جديدة، لمكتبة جامعة السودان المفتوحة من خلال هذا الكتاب ونسأل الله تعالى أن يكون هذا العمل خالصاً لوجهه، وأن يكون في ميزان حسناتنا، و أن ينتفع به الجميع، و يغفر لنا ما سهونا عنه و عجزنا عن إدراكه . ونحن نرحب بكل نقد بناء يسهم في تطوير هذا العمل إلى الصورة الأفضل، والله نسأل الأجر و الثواب. و أخر دعوانا أن الحمد لله رب العالمين إنه نعم المولى ونعم النصير .

محتويات المقرر


الوحدة الأولى
تمثيل البيانات في
الأنظمة الرقمية

محتويات الوحدة

مقدمة

تمهيد

مرحباً بك عزيزي الدارس، في الوحدة الأولى من مقرر "أساسيات التصميم المنطقي". توضح هذه الوحدة الطريقة التي يتم بها تمثيل مختلف أنواع البيانات داخل جهاز الحاسوب، أو داخل الأنظمة الرقمية (Digital Systems) بصفة عامة. حيث يتم تعريف الأنواع الأساسية من البيانات و تشمل الأعداد الصحيحة و الأعداد الحقيقية و الرموز)، وتوضيح طريقة تمثيل كل نوع منها، ومدى القيم التي يقبلها كل نوع، والاستخدامات المناسبة لكل نوع. كما تتناول الوحدة بعض أنواع الشفرات القياسية المستخدمة في الحاسوب لتمثيل البيانات.

أهداف الوحدة

✓

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- **تعرف** الأنواع الأساسية من البيانات و طريقة تمثيلها.
- **تستنتج** مدى القيم الذي يمكن استخدامه مع كل نوع.
- **تشرح** أنواع الشفرات القياسية المستخدمة في الحاسوب.

1. الأعداد الصحيحة (Integers)

عزيزي الدارس، حتى يتمكن أي نظام رقمي مثل الحاسوب من التعامل مع أي نوع من أنواع البيانات فإن تلك البيانات يجب أن تكون ممثلة فى الصورة الثنائية (Binary)، أي فى صورة مجموعة من الـ 0‘s والـ 1‘s. حيث يتم تمثيل القيمة المنطقية 0 بمستوى جهد معين داخل الدوائر الإلكترونية للنظام الرقمي، و يتم تمثيل القيمة المنطقية 1 بمستوى جهد آخر. مثلاً الجهد 5 Volt+ يمثل القيمة المنطقية 1 والجهد 0 Voltيمثل القيمة المنطقية 0.

أنواع الأعداد الصحيحة

تنقسم الأعداد الصحيحة إلى عدة أنواع حسب المساحة المستخدمة في تخزين العدد:

(1) عدد صحيح قصير (Short Integer) و طوله 1 Byte = 8 bits

(2) عدد صحيح (Integer) و طوله 2 Byte = 16 bits

(3) عدد صحيح طويل (Long Integer) و طوله 4 Byte = 32 bits

من ناحية أخرى تنقسم الأعداد الصحيحة حسب طبيعة الأعداد التي يتم تخزينها فيها إلى نوعين وهما:

(1) الأعداد الصحيحة بدون إشارة (Unsigned Numbers)، وفيها يتم تخزين الأعداد الموجبة فقط.

(2) الأعداد الصحيحة بإشارة (Signed Integers)، وفيها يتم تخزين الأعداد الموجبة والسالبة.

1.1 الأعداد الصحيحة بدون إشارة Unsigned Numbers

لتمثيل العدد الصحيح 135 مثلاً، يجب تحويله أولاً من الصورة العشرية (Decimal) إلى الصورة الثنائية (Binary). و يتم ذلك، كما تعلم عزيزي الدارس، بالقسمة المتكررة على 2 والاحتفاظ بباقي القسمة (كما تعلمت في مقرر "مبادئ علوم الحاسوب" الذي قمت بدراسته). فالعدد الصحيح العشري 135 يكافئ العدد الثنائي 10000111، ويكتب ذلك رياضياً في الصورة:

$$(135)_{10} = (10000111)_2$$

و للتحقق من ذلك يمكن أن نقوم بالعملية العكسية، أي تحويل العدد الثنائي 10000111 إلى الصورة العشرية.

تدريب(1)

حول الأعداد العشرية التالية إلى الصورة الثنائية

(1) 55 (2) 200 (3) 300 (4) 600

تدريب(2)

حول الأعداد الثنائية التالية إلى الصورة العشرية

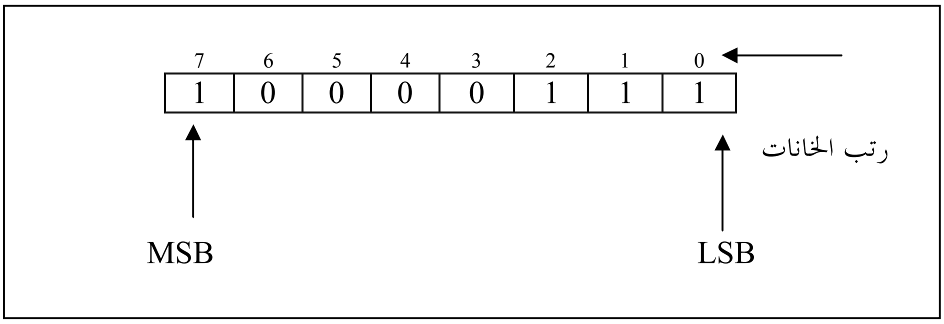
(1) 10111011 (2) 10000000 (3) 11100111

(4) 11111111

تسمى الخانة الواقعة في أقصى اليمين في العدد الثنائي بالخانة الدنيا (Least Significant Bit)، أو LSB اختصاراً، و ذلك لأنها الخانة الأقل وزناً. في حين

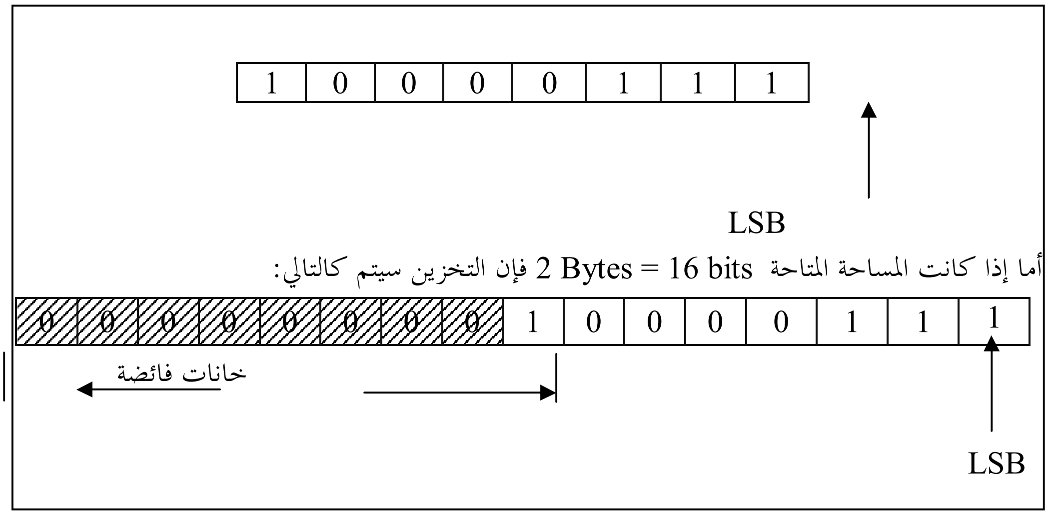
تسمى الخانة الواقعة في أقصى اليسار بالخانـــة العليـــا (Most Significant Bit)، أو MSB اختصاراً، و ذلك لأنها الخانة الأعلى وزناً.

تذكر أن وزن الخانة هو عبارة عن الأساس 2 مرفوع لأس يساوى رتبة الخانة، وأننـــا نحصل على رتب الخانات بترقيم الخانات ابتداء من الخانة التي تقع في أقصى اليمـــين، مبتدئين الترقيم بالقيمة صفر.



بعد تحويل العدد إلى الصورة الثنائية ننظر إلى المساحة المتاحة لتخزين العدد ، و نقـــوم بوضع الخانات بالترتيب فيها مبتدئين بالخانة الدنيا (LSB)، مع ملء أي خانات فائضـــة إلى اليسار بأصفار (0's).

مثلاً إذا كانت المساحة المتاحة 1 Byte = 8 bits فإن التخزين سيتم كالتالي:



أي انه إذا كان طول العدد الثنائي أقل من المساحة المتاحة يتم محاذاته إلى اليمين ثم تملأ الخانات الزائدة إلى اليسار بأصفار (0's). تسمى هذه العملية **بالمحاذاة إلى اليمين مع الملء بأصفار (Right Justify- Zero Fill)**.

يمكن حساب مدى القيم التي يمكن تخزينها في صورة عدد صحيح قصير (Short Integer) كالتالي:

المساحة المتاحة هي 1 Byte = 8 bits أي 8 خانات ثنائية

نحصل على أصغر قيمة بملء جميع الخانات بـــ 0's

0	0	0	0	0	0	0	0

= **0**

و نحصل على أكبر قيمة بملء جميع الخانات بـــ 1's

1	1	1	1	1	1	1	1

= **255**

و عليه فإن مدى القيم التي يمكن تمثيلها فى صورة عدد صحيح قصـــير (Short) هـــو $0 \sim 255$ أو $0 \sim (2^8 -1)$

و بالمثل يمكن إثبات أن مدى القيم التي يمكن تمثيلها في صورة عدد صحيح (Integer) هو $0 \sim (2^{16} -1)$

وعموماً إذا كان عدد الخانات المتاحة هو N فإن المدى هو $0 \sim (2^N -1)$

الجدول التالي يوضح أنواع الأعداد الصحيحة وطول كل منها ومدى القيم الـــذي يمكـــن تخزينه في كل نوع

نوع العدد الصحيح	طوله	مدى القيم
Short Integer	1 Byte = 8 bits	$0 \sim (2^8 - 1)$ $0 \sim 255$
Integer	2 Bytes = 16 bits	$0 \sim (2^{16} - 1)$ $0 \sim 65,535$
Long Integer	4 Bytes = 32 bits	$0 \sim (2^{32} - 1)$ $0 \sim 4,294,967,295$
————	N	$0 \sim (2^N - 1)$

تسمى الأعداد الصحيحة التي تعاملنا معها في ما سبق بالأعداد الصحيحة بدون إشارة (Unsigned Integers)

تدريب 3

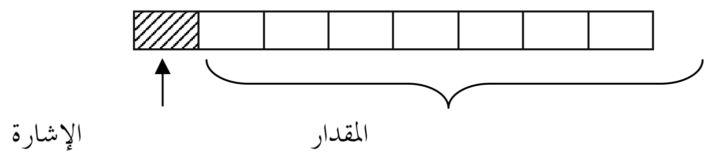
طالما أن مدى القيم التي يمكن تخزينها في الأعداد الصحيحة يزداد كلما ازداد طول العدد فلماذا تم استخدام أطوال مختلفة للأعداد (حيث استخدمت الأطوال 8 و 16 و 32 خانة)؟

2.1 الأعداد الصحيحة ذات الإشارة (Signed Number)

تناولنا في الجزء السابق طريقة تمثيل الأعداد الصحيحة بدون إشارة (Unsigned Number) والتي يتم تخزين قيم موجبة فقط بها. و بالتالي فإن أصغر قيمة يمكن تخزينها فيها هي الصفر.

والسؤال الآن هو كيف يتم تمثيل الأعداد السالبة في الحاسوب؟

لتمثيل الأعداد السالبة يتم حجز خانة bit لتمثيل إشارة العدد (Sign)، وعادة ما تكون هذه الخانة هي الخانة العليا MSB، و يتم تخزين مقدار العدد (Magnitude) في بقية الخانات.



و عادة ما تستخدم القيمة 0 في الخانة العليا MSB لتمثيل الإشارة الموجبة، في حين تستخدم القيمة 1 لتمثيل الإشارة السالبة. فلمعرفة إشارة العدد ننظر إلى الخانة العليا MSB فإذا كان:

MSB = 0 فالعدد موجب

MSB = 1 فالعدد سالب

مثلاً إذا أردنا تمثيل القيمة 27+ في صورة عدد صحيح بإشارة في مساحة تبلغ 1 Byte = 8 bits فإننا نتجاهل إشارة القيمة مؤقتاً و نقوم بتحويل المقدار من الصورة العشرية إلى الصورة الثنائية.

$$27 = (11011)_2$$

المساحة المتاحة تبلغ 8 خانات، نستبعد منها الخانة العليا MSB لتمثيل الإشارة، فيتبقى 7 خانات لتمثيل المقدار. يتم تخزين مقدار العدد الصحيح ذي الإشارة في المساحة المتاحة له بنفس طريقة تخزين الأعداد الصحيحة بدون إشارة (Unsigned Number). و أخيراً نضع 0 في خانة الإشارة لأن القيمة موجبة.

0	0	0	1	1	0	1	1	=	**+27**

الإشارة المقدار

و تمثيل القيمة 27- يتم بنفس الطريقة و لكن مع وضع 1 في خانة الإشارة لأن القيمة سالبة.

1	0	0	1	1	0	1	1	=	**-27**

الإشارة المقدار

يسمى هذا الأسلوب في تمثيل الأعداد الصحيحة ذات الإشارة بطريقة **المقدار–الإشارة (Sign-Magnitude)**، حيث تم الفصل بصورة كاملة ما بين إشارة القيمة و مقدارها.

هذا الأسلوب في تمثيل الأعداد الصحيحة ذات الإشارة به مشكلة خطيرة تتمثل في أن القيمة صفر لها شكلين

0	0	0	0	0	0	0	0	=	**+0**	صفر موجب
1	0	0	0	0	0	0	0	=	**-0**	صفر سالب

و وجود شكلين للصفر يعتبر مشكلة لأن عملية فحص قيمة معينة لمعرفة ما إذا كانت مساوية للصفر أم لا هي من أكثر العمليات التي يتم إجراؤها داخل الأنظمة الرقمية، و

وجود شكلين للصفر يعني أن هذه العملية يجب إجراؤها مرتين، مما يقلل كثيراً من كفاءة النظام الرقمي.

حلاً لهذه المشكلة يستخدم أسلوب مكمل الاثنينات **(2's Complement)** لتمثيل الأعداد الصحيحة ذات الإشارة.

مثلاً إذا أردنا تمثيل القيمة 27+ في صورة عدد صحيح بإشارة في مساحة تبلغ 1 Byte = 8 bits فإننا نتجاهل إشارة القيمة مؤقتاً و نقوم بتحويل المقدار من الصورة العشرية إلى الصورة الثنائية.

$$27 = (11011)_2$$

المساحة المتاحة تبلغ 8 خانات، لذلك نقوم بإكمال طول العدد الثنائي إلى 8 خانات و ذلك بإضافة أصفار (0's) إلى يسار العدد.

$$(11011)_2 = (00011011)_2$$

و أخيراً نقوم بوضع العدد الثنائي في المساحة المتاحة له

0	0	0	1	1	0	1	1	= +27

أما لتمثيل القيمة 27- فإننا نبدأ بنفس خطوات تمثيل القيمة 27+، حيث نتجاهل إشارة القيمة مؤقتاً و نقوم بتحويل المقدار من الصورة العشرية إلى الصورة الثنائية، ثم نقوم بإكمال طول العدد الثنائي إلى 8 خانات و ذلك بإضافة أصفار (0's) إلى يسار العدد.

$$27 = (11011)_2 = (00011011)_2$$

و بما أن القيمة المطلوب تمثيلها سالبة فإننا نحتاج إلى إيجـــاد مكمـــل الاثنينـــات (2's Complement) للعدد الثنائي الناتج، حيث إن المكمل الثاني لعدد ثنائي هنا يمثـــل القيمـــة السالبة للعدد.

إيجاد مكمل الاثنينات لعدد ثنائي يتم في خطوتين. الخطوة الأولى هي إيجـــاد المكمـــل الأول (1's Complement) و ذلك بعكس جميع خانات العدد الثنـــائي، أي تحويـــل أي 0 إلى 1 و تحويل أي 1 إلى 0. الخطوة الثانية هي إضافة 1 للمكمل الأول لنحصل علـــى المكمل الثاني.

العدد		00011011
المكمل الأول		11100100
	+	1
المكمل الثاني		11100101

أخيراً نقوم بوضع العدد الثنائي الناتج في المساحة المتاحة له.

1	1	1	0	0	1	0	1	=	-27

لاحظ الآتي :

الخانة العليا MSB هنا ما زالت تمثل إشارة العدد، فـــ MSB=0 للقيمة الموجبـــة 27+ و MSB=1 للقيمة السالبة 27-.

مكمل الاثنينات (2's Complement) لعدد ثنائي يمثل سالب ذلك العدد.

لا يوجد فصل ما بين مقدار العدد (Magnitude) و إشـــارته (Sign)، حيـــث إن جميـــع الخانات بما في ذلك خانة الإشارة تدخل في حساب مقدار العدد.

تدريب 4:

وضح طريقة تمثيل كل من القيم التالية في صورة عدد صحيح قصير بإشارة (Signed Short Integer)

(1) 15+ و 15- (2) 65+ و 65- (3) 1+ و 1-

(4) 127+ و 127-

إيجاد مقدار العدد السالب المطلوب مثلاً إيجاد القيمة العشرية للعدد الثنائي 11100101 إذا كان يُمَثِل عدداً صحيحاً قصيراً بإشارة.

نبدأ بتحديد إشارة العدد و ذلك بالنظر للخانة العليا MSB. في هذه الحالة نجد أن الخانة العليا MSB=1 مما يعني أن العدد سالب. لإيجاد مقدار عدد سالب نقوم بإيجاد مكمل الاثنينات له، لأن سالب العدد السالب عبارة عن عدد موجب كما نعلم.

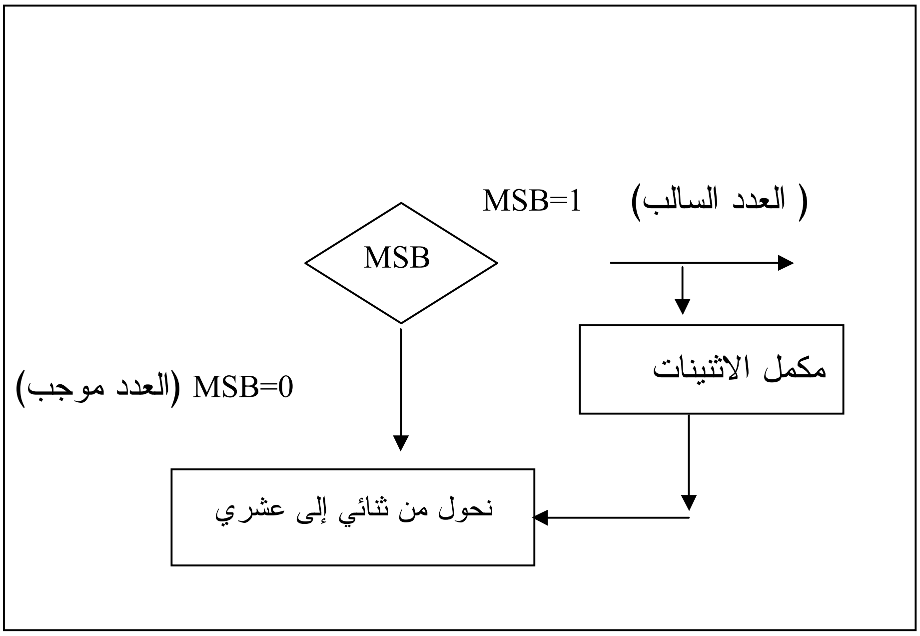
العدد	11100101
المكمل الأول	00011010
+	1
مكمل الاثنينات	00011011

أخيراً نقوم بتحويل المقدار من الصورة الثنائية للصورة العشرية

$$(00011011)_2 = (11011)_2 = 27$$

إذن العدد هو 27−

وعموماً لإيجاد قيمة عدد صحيح بإشارة يمكن استخدام المخطط التالي



مثال:

وضح طريقة تمثيل القيمة 13- في صورة:

(أ) عدد صحيح قصير بإشارة (Signed Short Integer)

(ب) عدد صحيح بإشارة (Signed Integer)

نقوم أولاً بتحويل المقدار إلى الصورة الثنائية $13 = (1101)_2$

(أ) عدد صحيح قصير بإشارة:

نكمل طول العدد إلى 8 خانات ثم نقوم بإيجاد مكمل الاثنينات له.

العدد		00001101
المكمل الأول		11110010
	+	1
مكمل الاثنينات		11110011

أي أن $\quad -13 = (11110011)_2$

(ب) عدد صحيح بإشارة:

نكمل طول العدد إلى 16 خانة ثم نقوم بإيجاد مكمل الاثنينات له

العدد	0000000000001101
المكمل الأول	1111111111110010
+	1
المكمل الثاني	1111111111110011

أي أن $\quad -13 = (1111111111110011)_2$

في المثال السابق قمنا في (أ) بتمثيل العدد الصحيح ذي الإشارة 13- في 8 خانات ثـــم قمنا في (ب) بزيادة طول العدد إلى 16 خانة

1	1	1	1	0	0	1	1

								1	1	1	1	0	0	1	1

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

لاحظ أننا قد قمنا بملء الخانات الفائضة إلى اليسار بـــ 1's

و بالمقارنة إذا أردنا تمثيل القيمة الموجبة 13+ في 8 خانات ثم في 16 خانة

0	0	0	0	1	1	0	1

								0	0	0	0	1	1	0	1

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

ملاحظة

لاحظ هنا أننا قد قمنا بملء الخانات الفائضة إلى اليسار بـــ 0's

يمكن بصورة عامة القول أنه عند زيادة طول العدد الصحيح ذي الإشـــارة فإننـــا نقـــوم بملء الخانات الفائضة إلى اليسار بإشارة العدد. و تسمى هذه العملية بـــتمديد الإشـــارة (Sign Extension).

مثال:

أوجد القيمة العشرية للعدد الثنائي 11110101 وذلك إذا كان يمثل:

(أ) عدداً صحيحاً قصيراً بدون إشارة (Unsigned Short Integer).

(ب) عدداً صحيحاً قصيراً بإشارة (Signed Short Integer).

(أ) العدد بدون إشارة ،(Unsigned) و بالتالي فإن كل الخانات تمثل مقدار العدد، و مـــا علينا إلا التحويل من الصورة الثنائية للصورة العشرية

$$(11110101)_2 = 2^7 + 2^6 + 2^5 + 2^4 + 2^2 + 2^0 = 128 + 64 + 32 + 16 + 4 + 1 = 245$$

(ب) العدد بإشارة (Signed) و عليه ننظر للخانة العليا MSB لتحديد إشـــارته. MSB=1 مما يعني أن العدد سالب. لحساب المقدار نقوم بإيجاد مكمل الاثنينات.

العدد		11110101
المكمل الأول		00001010
	+	1
المكمل الثاني		00001011

ثم نحول المقدار للصورة العشرية $(00001011)_2 = (1011)_2 = 2^3 + 2^1 + 2^0 = 11$

أي أن القيمة هي 11-

تدريب 5

أوجد القيمة العشرية لكل من الأعداد الثنائية التالية إذا كان كل منها يمثل عدداً قصيراً بإشارة (Signed Short)

(1) 10111111　　(2) 10000000　　(3) 01111111

(4) 11111111

مدى القيم التي يمكن تخزينها في مساحة معينة في صورة عدد صحيح بإشارة

لتوضيح الأمر نبدأ بالمثال التالي.

مثال: حدد جميع الأعداد الصحيحة ذات الإشارة (Signed Integers) التى يمكن تمثيلها في مساحة قدرها 4 خانات.

قيم موجبة **(MSB=0)**	القيمة العشرية **(Decimal)**	قيم سالبة **(MSB=1)**	القيمة العشرية **(Decimal)**
0000	+0	1000	-8
0001	+1	1001	-7
0010	+2	1010	-6
0011	+3	1011	-5
0100	+4	1100	-4
0101	+5	1101	-3
0110	+6	1110	-2
0111	+7	1111	-1

و عليه فإن مدى القيم التي يمكن تمثيلها في صورة عـدد صـحيح بإشـارة (Signed Numbers) طوله 4 خانات هو

-8 ~ +7

−23 ~ +23 -1

−24-1 ~ +24-1 -1

و بصورة عامة فإن مدى الأعداد الصحيحة ذات الإشارة (Signed Numbers) التـــى يمكن تمثيلها في مساحة تبلغ N خانة هو

$-2N-1 \sim +2N-1 - 1$

و كملخص لما سبق فإن الأعداد الصحيحة (Numbers) تتقسم من حيث الإشارة إلـــى نوعين: بإشارة (Signed) و بدون إشارة (Unsigned)
كما تتقسم الأعداد الصحيحة (سواء أكانت بإشارة أو بدون إشارة)، من حيث الطول، إلى ثلاثة أنواع: Short و Integer و Long

Integers

<u>Signed</u>	**<u>Unsigned</u>**
Short	Short
Numbers	Numbers
Long	Long

ملاحظة

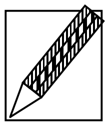
عادة لا تذكر كلمة Signed صراحة في لغات البرمجة و إنما تفهم ضمناً، فمثلاً Integer تعني Signed Numbers وShort Numbers تعني Signed Short Integer. أما كلمة Unsigned فيجب أن تذكر صراحة.

الجدول التالي يوضح أنواع الأعداد الصحيحة و مدى القيم الذي يقبلها كل نوع

نوع العدد الصحيح	طوله	مدى القيم	
		Unsigned	**Signed**
Short Integer	1 Byte = 8 bits	$0 \sim (2^8 - 1)$ $0 \sim 255$	$-2^7 \sim +(2^7 - 1)$ $-128 \sim +127$
Integer	2 Bytes =16 bits	$0 \sim (2^{16} - 1)$ $0 \sim 65{,}535$	$-2^{15} \sim +(2^{15} - 1)$ $-32{,}768 \sim +32{,}767$
Long Integer	4 Bytes = 32 bits	$0 \sim (2^{32} - 1)$ $0 \sim 4{,}294{,}967{,}295$	$-2^{31} \sim +(2^{31} - 1)$ $-2{,}147{,}483{,}648 \sim +2{,}147{,}483{,}647$
————	N bits	$0 \sim (2^N - 1)$	$-2^{N-1} \sim +(2^{N-1} - 1)$

مما سبق يتضح لنا أن الأعداد الصحيحة يتم تمثيلها دون أي خطأ، أي بالدقة الكاملة، طالما أن عدد الخانات المتاحة يكفي لتمثيل القيمة. المشكلة الوحيدة التي يمكن أن تظهر في تمثيل الأعداد الصحيحة هي أن تكون القيمة المطلوب تخزينها خارج المدى المحدد للمساحة المتاحة، عند ذلك يحدث ما يسمى **Mathematical Over Flow**

تدريب 6



وضح ما يحدث إذا أردنا أن نقوم بتخزين القيمة العشرية 200 في صورة

(أ) عدد صحيح قصير بدون إشارة (Unsigned Short Integer).

(ب) عدد صحيح قصير بإشارة (Signed Short Integer).

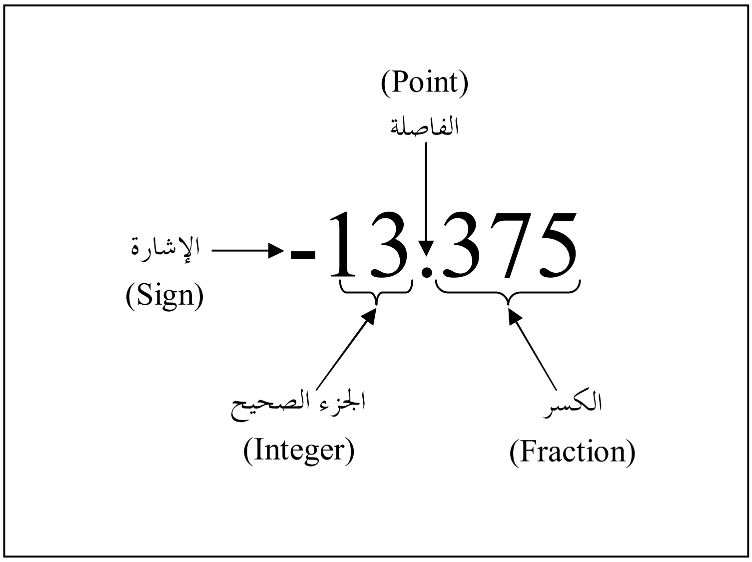
أسئلة تقويم ذاتي

1- قارن بين عملية تمديد العدد، أي زيادة طوله، في كل من الأعداد الصحيحة بدون إشارة (Unsigned Integers) والأعداد الصحيحة ذات الإشارة (Signed Integers).

2- ما المقصود بهذا المصطلح **Mathematical Over Flow**؟

2. الأعداد الحقيقية (Real Numbers)

العدد الحقيقي (Real Number) هو العدد الذي قد يكون محتوياً على كسر (Fraction)، مثل 13.375- أو 0.1 أو 5.3333 يتكون العدد الحقيقي من جزئين: عدد صحيح (Integer) و كسر (Fraction)، تفصل بينهما الفاصلة (Point)، و التي يطلق عليها في النظام العشري الفاصلة العشرية (Decimal Point). وللعدد الحقيقي إشارة (Sign). الشكل التالي يوضح أجزاء العدد الحقيقي.



التحويل من الصورة العشرية للصورة الثنائية

لتمثيل العدد الحقيقي في الحاسوب، أو الأنظمة الرقمية عموماً، يجب أن يتم تحويله أولاً من الصورة العشرية (Decimal) إلى الصورة الثنائية (Binary). و هنا يتم تحويل كـل من جزئي العدد الحقيقي على حدة. حيث نبدأ بتحويل الجزء الصـحيح إلـى الصـورة الثنائية، و ذلك بنفس طريقة تحويل الأعداد الصحيحة، أي بالقسمة المتكررة علـى 2 و الاحتفاظ ببواقي القسمة. فإذا قمنا بتحويل الجزء الصحيح من العدد الحقيقـي 13.375- إلى الصورة الثنائية نحصل على

$$13 = (1101)_2$$

لاحظ هنا أننا قد تجاهلنا إشارة العدد الحقيقي مؤقتاً، و سنقوم لاحقـاً بتوضـيح طريقـة التعامل مع الإشارة.

بعد ذلك نقوم بتحويل الكسر من الصورة العشرية إلى الصورة الثنائيـة، و يـتم ذلـك بالضرب المتكرر في 2 و الاحتفاظ بالجزء الصحيح من النتيجة. فـإذا أردنـا تحويـل الكسر في العدد الحقيقي 13.375- إلى الصورة الثنائية يتم ذلك كالتالي

$$0.375 \times 2 = 0.75 \rightarrow 0$$

عندما قمنا بضرب الكسر 0.375 في 2 حصلنا على 0.75 ، نقـوم بالاحتفـاظ بـالجزء الصحيح من النتيجة و هو في هذه الحالة 0، ثم نقوم بضرب الكسر المتبقي في 2

$$0.75 \times 2 = 1.5 \rightarrow 1$$

عندما ضربنا الكسر المتبقي 0.75 في 2 حصلنا على 1.5 ، نحتفظ بالجزء الصحيح مـن النتيجة 1 ثم نضرب الكسر المتبقي في 2. لاحظ أن الكسر المتبقي هنا هـو 0.5 بعـد احتفاظنا بالـ 1.

$$0.5 \times 2 = 1.0 \rightarrow 1$$

بضرب الكسر المتبقي 0.5 في 2 حصلنا على 1.0 ، نحتفظ بالجزء الصحيح من النتيجة 1 ، و نجد أن الكسر المتبقي قد أصبح 0.0 فنتوقف عن الضرب في 2 و بذلك تكون عملية تحويل الكسر إلى الصورة الثنائية قد انتهت. لمزيد من الوضوح نقوم بوضع خطوات التحويل معاً

$$
\begin{array}{llllll}
0.375 & \times & 2 & = & 0.75 & \rightarrow 0 \\
0.75 & \times & 2 & = & 1.5 & \rightarrow 1 \quad \downarrow \\
0.5 & \times & 2 & = & 1.0 & \rightarrow 1 \\
0.0 & & & & &
\end{array}
$$

لتكوين الكسر في الصورة الثنائية نأخذ الأجزاء الصحيحة التي احتفظنا بها و نضعها بالترتيب (من أعلى إلى أسفل) على يمين الفاصلة (Point)، و نضع 0 يسار الفاصلة

$$0.011$$

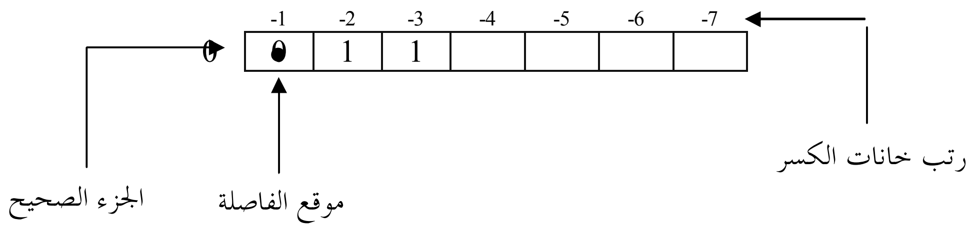
و عليه فإن الكسر العشري 0.375 يساوي 0.011 في الصورة الثنائية، و نكتب ذلك كالتالي

$$0.375 = (0.011)_2$$

لاحظ أن الفاصلة (Point) في الكسر الثنائي يطلق عليها الفاصلة الثنائية (Binary Point)، في حين يطلق عليها الفاصلة العشرية (Decimal Point) في الكسر العشري. و تسهيلاً للأمور سنستخدم تسمية الفاصلة (Point) فقط في النظامين الثنائي و العشري.

و للتحقق من صحة النتيجة التي حصلنا عليها عند تحويل الكسر من الصورة العشرية للصورة الثنائية يمكن أن نقوم بالعملية العكسية، أي تحويل الكسر من الصورة الثنائية و إعادته مرة أخرى إلى الصورة العشرية. و يتم ذلك بطريقة مشابهة لتحويل العدد الصحيح من الصورة الثنائية للصورة العشرية، أي بجمع أوزان الخانات الحاوية على

القيمة 1 في الكسر الثنائي. و وزن الخانة هنا هو أيضاً عبارة عن الأساس 2 مرفــوع لأس يساوي رتبة الخانة. و نحصل على رتب الخانات هنا بترقيم الخانات بقــيم ســالبة مبتدئين بالقيمة 1- لأول خانة يمين الفاصلة ثم 2- للخانة التي تليها ... و هكذا.



الخانات الحاوية على القيمة 1 هنا هي الخانة ذات الرتبة 2- و الخانة ذات الرتبة 3-. و أوزان هاتين الخانتين هي 2^{-2} و 2^{-3} على التوالي. و بناء عليه فإن

$$(0.011)_2 = 2^{-2} + 2^{-3} = \frac{1}{2^2} + \frac{1}{2^3} = \frac{1}{4} + \frac{1}{8} = 0.25 + 0.125 = 0.375$$

الآن و بعد أن قمنا بتحويل جزئي العدد الحقيقي، الجزء الصحيح و الكسر، كل على حدة، من الصورة العشرية إلى الصورة الثنائية نقوم بوضع الجزئين معاً لتكوين العدد الحقيقي في الصورة الثنائية

$$\begin{aligned} 13 &= (1101)_2 \\ 0.375 &= (0.011)_2 \\ 13.375 &= (1101.011)_2 \end{aligned}$$

من الممكن أن نقوم بتحويل العدد الحقيقي كاملاً (بجزئيه) من الصورة الثنائية إلى الصورة العشرية كالتالي

3	2	1	0	-1	-2	-3
1	1	0	1	0	1	1

موقع الفاصلة

بعد ترقيم الخانات و تحديد رتبها ما علينا إلا جمع أوزان الخانات الحاوية على 1، و وزن الخانة، كما ذكرنا من قبل، هو عبارة عن الأساس 2 مرفوع لأس يساوي رتبة الخانة. و عليه

$$(1101.011)_2 = 2^3 + 2^2 + 2^0 + 2^{-2} + 2^{-3} = 8 + 4 + 1 + 0.25 + 0.125 = 13.375$$

كنوع من التسهيل فإن الجدول التالي يحتوي على أوزان الخانات الثمانية الأولى ذات الرتب الموجبة و الخانات الثمانية الأولى ذات الرتب السالبة

رتبة الخانة	وزنـــهـــــــــــا		رتبة الخانة	وزنـــهـــــــــــا	
0	2^0	1			
1	2^1	2	-1	2^{-1}	0.5
2	2^2	4	-2	2^{-2}	0.25
3	2^3	8	-3	2^{-3}	0.125
4	2^4	16	-4	2^{-4}	0.0625
5	2^5	32	-5	2^{-5}	0.03125
6	2^6	64	-6	2^{-6}	0.015625
7	2^7	128	-7	2^{-7}	0.0078125
8	2^8	256	-8	2^{-8}	0.00390625

مثال:

حول العدد العشري 9.625 إلى الصورة الثنائية.

نبدأ بتحويل الجزء الصحيح إلى الصورة الثنائية $9 = (1001)_2$

ثم نقوم بتحويل الكسر إلى الصورة الثنائية

$$\begin{array}{llllllll} 0.625 & \times & 2 & = & 1.25 & \rightarrow & 1 & \\ 0.25 & \times & 2 & = & 0.5 & \rightarrow & 0 & \downarrow \\ 0.5 & \times & 2 & = & 1.0 & \rightarrow & 1 & \\ 0.0 & & & & & & & \end{array}$$

أي أن $0.625 = (0.101)_2$

$$9.625 = (1001.101)_2$$

وعليه إذا أردنا تحويل العدد الثنائي الناتج مرة أخرى إلى الصورة العشرية نقوم بترقيم الخانات ثم بجمع أوزان الخانات الحاوية على 1

$$\begin{array}{ccccccc} 3 & 2 & 1 & 0 & -1_{\bullet} & -2 & -3 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 \end{array}$$

و عليه $(1001.101)_2 = 2^3 + 2^0 + 2^{-1} + 2^{-3} = 8 + 1 + 0.5 + 0.125 = 9.625$

مثال:

حول العدد الحقيقي 0.34375 من الصورة العشرية للصورة الثنائية.

العدد الحقيقي هنا مكون من كسر فقط و الجزء الصحيح فيه يساوي صفراً، لذلك نقوم فقط بتحويل الكسر من الصورة العشرية للصورة الثنائية

0.34375	$\times$	2	=	0.6875	$\rightarrow$	0
0.6875	$\times$	2	=	1.375	$\rightarrow$	1
0.375	$\times$	2	=	0.75	$\rightarrow$	0
0.75	$\times$	2	=	1.5	$\rightarrow$	1
0.5	$\times$	2	=	1.0	$\rightarrow$	1
0.0						

أي أن

$$0.34375 = (0.01011)_2$$

و يمكنك التحقق من صحة النتيجة بتحويل العدد الثنائي مرة أخرى إلى الصورة العشرية.

مثال:

حول العدد الحقيقي 0.1 من الصورة العشرية إلى الصورة الثنائية.

العدد هنا عبارة عن كسر فقط نقوم بتحويله إلى الصورة الثنائية.

$$
\begin{array}{cccccccl}
0.1 & \times & 2 & = & 0.2 & \rightarrow & 0 & \\
0.2 & \times & 2 & = & 0.4 & \rightarrow & 0 & \\
0.4 & \times & 2 & = & 0.8 & \rightarrow & 0 & \\
0.8 & \times & 2 & = & 1.6 & \rightarrow & 1 & \\
0.6 & \times & 2 & = & 1.2 & \rightarrow & 1 & \\
0.2 & \times & 2 & = & 0.4 & \rightarrow & 0 & \\
0.4 & \times & 2 & = & 0.8 & \rightarrow & 0 & \\
0.8 & \times & 2 & = & 1.6 & \rightarrow & 1 & \downarrow \\
0.6 & \times & 2 & = & 1.2 & \rightarrow & 1 & \\
0.2 & \times & 2 & = & 0.4 & \rightarrow & 0 & \\
0.4 & \times & 2 & = & 0.8 & \rightarrow & 0 & \\
0.8 & \times & 2 & = & 1.6 & \rightarrow & 1 & \\
0.6 & \times & 2 & = & 1.2 & \rightarrow & 1 & \\
0.2 & \times & 2 & = & 0.4 & \rightarrow & 0 & \\
0.4 & \times & 2 & = & 0.8 & \rightarrow & 0 & \\
\vdots & & & & \vdots & & \vdots &
\end{array}
$$

تذكر أننا نتوقف عن الضرب في 2 عندما يصبح الكسر المتبقي مساوياً صفراً، و هو ما لا يحدث في هذا المثال. السبب في ذلك هو أن الكسر في الصورة الثنائية عبارة عن كسر غير منتهٍ. و الكسور غير المنتهية أمر شائع في النظام العشري مثلاً

$$\frac{1}{3} = 0.333333333333333\cdots$$

$$\frac{1}{6} = 0.166666666666666\cdots$$

$$\frac{1}{12} = 0.08333333333333\ldots$$

$$\frac{1}{\sqrt{2}} = 0.7071067811865\cdots$$

في مثل هذه الحالات التي يكون فيها الكسر في الصورة الثنائية غير منتهٍ نتوقف عن الضرب في 2 عندما نحصل على عدد كافٍ من الخانات. و العدد الكافي من الخانات

هنا غالباً ما تحدده المساحة المتاحة لتخزين الكسر، و هذه المساحة دائماً محدودة. و في المثال الذي نحن بصدده مثلاً توقفنا بعد الحصول على 15 خانة من الكسر الثنائي. أي أن

$$0.1 = (0.000110011001100\cdots)_2$$

و بالطبع فإن 15 خانة فقط في الكسر الثنائي لا تمثل الكسر العشري 0.1 بدقة كاملة و إنما تمثله فقط بصورة تقريبية، و كلما زاد عدد خانات الكسر الثنائي كان أكثر دقة في تمثيل الكسر العشري، و لكن يتعذر تمثيل الكسر العشري بدقة كاملة لأن ذلك يتطلب عدداً لا نهائياً من الخانات في الكسر الثنائي.

من المثال السابق نستنتج أن تحويل بعض الأعداد الحقيقية إلى الصورة الثنائية قد ينتج عنه كسر غير منتهٍ ، و مثل هذه الكسور غير المنتهية يتعذر تخزينها كاملة لأن عدد الخانات المتاحة للتخزين دائماً محدودة.

تدريب 7

حوّل القيم التالية من الصورة العشرية إلى الصورة الثنائية، ثم تحقق من صحة النتيجة بإعادة العدد الثنائي الناتج مرة أخرى إلى الصورة العشرية.

(1) 53.8125 (2) 0.1015625 (3) 0.7

بهذا نكون قد أنجزنا الخطوة الأولى في تمثيل الأعداد الحقيقية (Real Numbers)، و هي تحويل العدد الحقيقي من الصورة العشرية إلى الصورة الثنائية، أي إلى مجموعة من

الـــ0's و الـــ1's، لأن هذه هي الصورة الوحيدة التي يمكن أن تتعامـــل معهـــا الـــدوائر الإلكترونية للحاسوب و بقية الأنظمة الرقمية (Digital Systems).

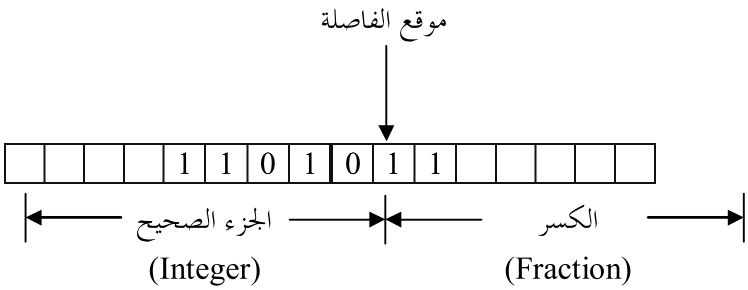
الخطوة التالية هي وضع العدد الحقيقي في صورته الثنائيـــة فـــي المســـاحة التخزينيـــة المتاحة له. و هنا تواجهنا أول مشكلة، و هي أن العدد الحقيقي مكون من جزئين: جزء صحيح (Integer) و كسر (Fraction)، تفصل بينهما الفاصلة (Point). مـــثلاً إذا كـــان العدد الحقيقي المطلوب تمثيله هو 13.375-، فقد نجد أن

$$13.375 = (1101.011)_2$$

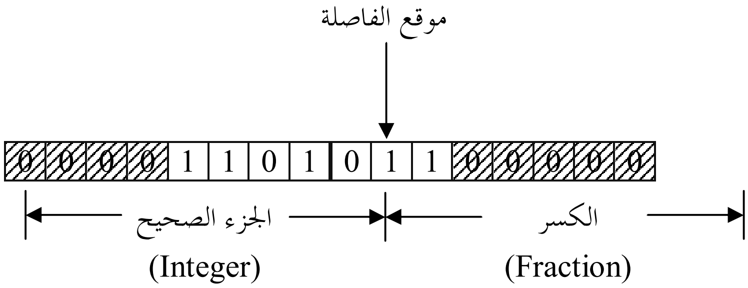
فإذا كانت المساحة التخزينية المتاحة عبارة عن 16 خانة ثنائية (16 bit)، فأين نضع الجزء الصحيح؟ و أين نضع الكسر؟ و كيف نقوم بتمثيل الفاصلة؟

أسلوب الفاصلة الثابتة (Fixed Point)

أحد الحلول البديهية التي قد تتبادر إلى الذهن هو أن نقوم بتقسيم المساحة المتاحة مابين الجزء الصحيح و الكسر. فنقول مثلاً إن الخانات الثماني الواقعة على يمين الكسر و الخانات الثمانية الواقعة على اليسار للجزء الصحيح.

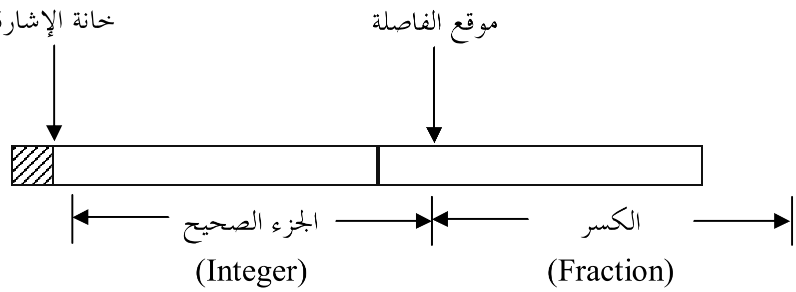


لاحظ أننا قد قمنا بمحاذاة الجزء الصحيح إلى يمين الجزء المخصص له، في حين قمنا بمحاذاة الكسر إلى يسار الجزء المخصص له، ثم نقوم بملء الخانات الفائضة على كل من يمين الكسر و يسار الجزء الصحيح بأصفار



لاحظ أنه في هذا الأسلوب لتمثيل الأعداد الحقيقية نفترض وجود الفاصلة (Point) عند الحد الفاصل ما بين الجزء من المساحة المخصصة للعدد الصحيح و الجزء المخصص للكسر. و موقع الفاصلة هنا ثابت (Fixed)، لذلك يسمى هذا الأسلوب في تمثيل الأعداد الحقيقية بأسلوب الفاصلة الثابتة (Fixed Point).

أما عن إشارة العدد الحقيقي فيمكن تمثيلها هنا بأن يتم تخصيص خانة لها، و لتكن الخانة العليا (MSB)، و نضع في هذه الخانة القيمة 0 إذا كان العدد الحقيقي موجباً و القيمة 1 إذا كان سالباً.



يعيب هذا الأسلوب في تمثيل الأعداد الحقيقية عدم الإستغلال الأمثل للمساحة التخزينية المتاحة. فكثيراً ما يكون الجزء الصحيح من العدد الحقيقي مساوياً صفراً، أي أن العدد الحقيقي عبارة عن كسر فقط، و بالتالي يكون الجزء من المساحة التخزينية المخصص للعدد الصحيح غير مستغل. و يا حبذا لو كان في الإمكان أن نضيف هذه المساحة غير

المستغلة للجزء المخصص للكسر، حيث أنه كلما زاد عدد خانات الكسر كان تمثيل العدد الحقيقي أكثر دقة (بسبب مشكلة الكسور غير المنتهية). و لكن مع الأسف هذا غير ممكن في أسلوب الفاصلة الثابتة (Fixed Point).

لاستغلال المساحة التخزينية المتاحة للعدد الحقيقي بصورة أكثر كفاءة يستخدم أسلوب آخر في تمثيل الأعداد الحقيقية يسمى بأسلوب الفاصلة المتحركة (Floating Point).

أسلوب الفاصلة المتحركة (Floating Point)

يقوم هذا الأسلوب في تمثيل الأعداد الحقيقية على التخلص من الجزء الصحيح من العدد الحقيقي بحيث يصبح العدد بأكمله عبارة عن كسر. و يتم ذلك بتحريك أو إزاحة الفاصلة. و عملية تحريك الفاصلة من العمليات المألوفة بالنسبة لنا في الصورة العشرية (Decimal). مثلاً يمكن أن نقوم بتحريك الفاصلة في العدد الحقيقي العشري 123.456 يميناً أو يساراً كالتالي:

تحريك الفاصلة يساراً

$$\begin{aligned} 123.456 &= 12.3456\times10^{1} \\ &= 1.23456\times10^{2} \\ &= 0.123456\times10^{3} \\ &= 0.0123456\times10^{4} \end{aligned}$$

تحريك الفاصلة يميناً

$$\begin{aligned} 123.456 &= 1234.56\times10^{-1} \\ &= 12345.6\times10^{-2} \\ &= 123456.0\times10^{-3} \\ &= 1234560.0\times10^{-4} \end{aligned}$$

لاحظ أنه عندما يتم إزاحة الفاصلة من موقعها الأصلي يميناً أو يساراً يتم الحفاظ على قيمة العدد بالضرب في الأساس 10 مرفوعاً لأس يساوي عدد خانات الإزاحة، و يكون هذا الأس موجباً عندما تكون الإزاحة إلى اليسار و سالباً عندما تكون الإزاحة إلى

اليمين. أي أن مقدار الأس يمثل عدد خانات الإزاحة للفاصلة، و إشارته تمثل اتجاه تلك الإزاحة.

و بنفس الطريقة يمكن إزاحة الفاصلة في الأعداد الحقيقية في الصورة الثنائيـــة، و لكـــن الحفاظ على القيمة هنا يتم بالضرب في الأساس 2 مرفوعاً لأس يساوي مقـــداره عـــدد خانات الإزاحة و إشارته اتجاه الإزاحة. مثلاً يمكن تحريك الفاصلة في العدد الحقيقـــي الثنائي 1101.011 يميناً أو يساراً كالتالي:

تحريك الفاصلة يساراً

$$\begin{aligned} 1101.011 &= 110.1011 \times 2^1 \\ &= 11.01011 \times 2^2 \\ &= 1.101011 \times 2^3 \\ &= 0.1101011 \times 2^4 \end{aligned}$$

تحريك الفاصلة يميناً

$$\begin{aligned} 1101.011 &= 11010.11 \times 2^{-1} \\ &= 110101.1 \times 2^{-2} \\ &= 1101011.0 \times 2^{-3} \\ &= 11010110.0 \times 2^{-4} \end{aligned}$$

لتمثيل العدد الحقيقي نستخدم عملية تحريك الفاصلة للتخلص من الجزء الصـــحيح مـــن العدد الحقيقي و تحويل العدد بأكمله إلى كسر. و تسمى هذه العمليـــة بعمليـــة التطبيـــع **(Normalization)**. فإذا أردنا مثلاً إجراء عملية تطبيع للعدد الحقيقي الثنائي 1101.011 يتم ذلك كالتالي:

$$1101.011 = 0.1101011 \times 2^4$$

لاحظ أننا قد قمنا هنا بإزاحة الفاصلة 4 خانات إلى اليسار حتى أصبح الجزء الصحيح من العدد الحقيقي مساوياً 0. و بالطبع كان يمكن أن نقوم بإزاحة الفاصلة إلى اليسار أكثر من ذلك، 5 أو 6 خانات، و سيظل الجزء الصحيح من العدد الحقيقي مساوياً 0 أيضاً. و لكن هناك شرط آخر مهم في عملية التطبيع يضمن لنا عدم إزاحة الفاصلة أكثر من اللازم، و هذا الشرط هو أن تكون أول خانة ثنائية على يمين الفاصلة في الكسر حاوية على القيمة 1. أي أنه في عملية التطبيع نقوم بإزاحة الفاصلة عدداً من الخانات يميناً أو يساراً بحيث:

1- يصبح الجزء الصحيح مساوياً 0.

2- تكون أول خانة ثنائية على يمين الفاصلة حاوية على 1.

بعد إجراء عملية التطبيع (Normalization) يصبح العدد الحقيقي مكوناً من كسر **(Fraction)** و أس **(Exponent)**.

$$\mathbf{\underbrace{0.1101011}_{\text{Fraction}} \times 2^{4}}$$

كسر

(Fraction)

أس

(Exponent)

الكسر (Fraction) يشتمل على كل الخانات الثنائية (bits) الممثلة للعدد الحقيقي، والأس (Exponent) يحتفظ بموقع الفاصلة (Point).

مثال:

حول العدد العشري 3.625 إلى الصورة الثنائية ثم قم بتطبيعه.

نبدأ بتحويل الجزء الصحيح إلى الصورة الثنائية $3 = (11)_2$

ثم نقوم بتحويل الكسر إلى الصورة الثنائية

$$\begin{array}{lcccccc} 0.625 & \times & 2 & = & 1.25 & \rightarrow & 1 \\ 0.25 & \times & 2 & = & 0.5 & \rightarrow & 0 \quad \downarrow \\ 0.5 & \times & 2 & = & 1.0 & \rightarrow & 1 \\ 0.0 & & & & & & \end{array}$$

أي أن $0.625 = (0.101)_2$

و عليه $3.625 = (11.101)_2$

لتطبيع العدد 11.101 نقوم بإزاحة الفاصلة خانتين إلى اليسار

$$11.101 = 0.11101 \times 2^2$$

مثال:

حول العدد العشري 0.203125 إلى الصورة الثنائية ثم قم بتطبيعه.

نقوم بتحويل الكسر إلى الصورة الثنائية

$$\begin{array}{lcccccc} 0.203125 & \times & 2 & = & 0.40625 & \rightarrow & 0 \\ 0.40625 & \times & 2 & = & 0.8125 & \rightarrow & 0 \\ 0.8125 & \times & 2 & = & 1.625 & \rightarrow & 1 \quad \downarrow \\ 0.625 & \times & 2 & = & 1.25 & \rightarrow & 1 \\ 0.25 & \times & 2 & = & 0.5 & \rightarrow & 0 \\ 0.5 & \times & 2 & = & 1.0 & \rightarrow & 1 \\ 0.0 & & & & & & \end{array}$$

أي أن $0.203125 = (0.001101)_2$

لإجراء عملية التطبيع نحتاج لإزاحة الفاصلة خانتين إلى اليمين

$$0.001101 = 0.1101 \times 2^{-2}$$

تدريب 8

حول القيم التالية من الصورة العشرية إلى الصورة الثنائية، ثم قم بتطبيعها.

(1) 53.8125 (2) 0.1015625 (3) 0.7

تخزين الأعداد الحقيقية

بعد الانتهاء من تحويل العدد الحقيقي من الصورة العشرية إلى الصورة الثنائية ثم تطبيعه، مطلوب الآن وضع العدد في المساحة التخزينية المتاحة له. و المعلومات المطلوب وضعها في المساحة المتاحة هي:

1- الكسر (Fraction)
2- الأس (Exponent)
3- الإشارة (Sign)

المقصود بالإشارة (Sign) هنا هو إشارة العدد الحقيقي.

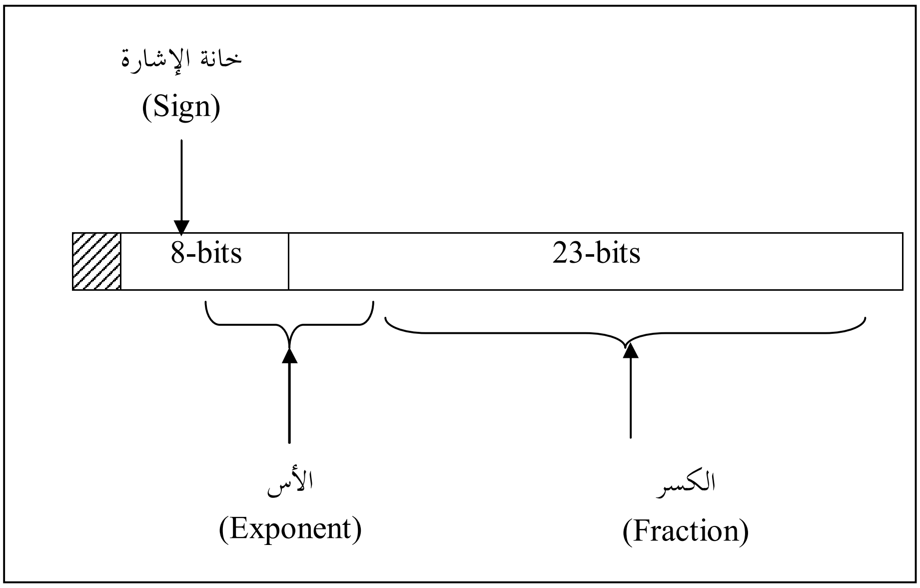
لاحظ أن الكسر (Fraction) في هذه المرحلة جاهز في الصورة الثنائية، و لكن الأس عبارة عن عدد صحيح بإشارة في الصورة العشرية و يجب تحويله إلى الصورة الثنائية. في بدايات تعامل الحواسيب مع الأعداد الحقيقية (Real Numbers) – و كانت قبل ذلك تتعامل مع الأعداد الصحيحة (Integers) فقط – حدث اختلاف ما بين الشركات الرئيسية المصنعة للحواسيب في ذلك الوقت في طريقة وضع المعلومات الممثلة للعدد الحقيقي (الكسر و الأس و الإشارة) داخل المساحة المخصصة لتخزين العدد الحقيقي، الأمر الذي أدى إلى حدوث مشاكل عند نقل البيانات ما بين تلك الحواسيب. استمر ذلك حتى قامت جمعية معهد مهندسي الكهرباء والالكترونيات. IEEE (**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers) بتدارك الأمر بوضع مواصفات قياسية (Standards) لتخزين الأعداد الحقيقية. تم في هذه المواصفات تعريف نوعين من الأعداد الحقيقية:

1- العدد الحقيقي ذو الدقة العادية (IEEE Single Precision Float)

2- العدد الحقيقي ذو الدقة المضاعفة (IEEE Double Precision Float)

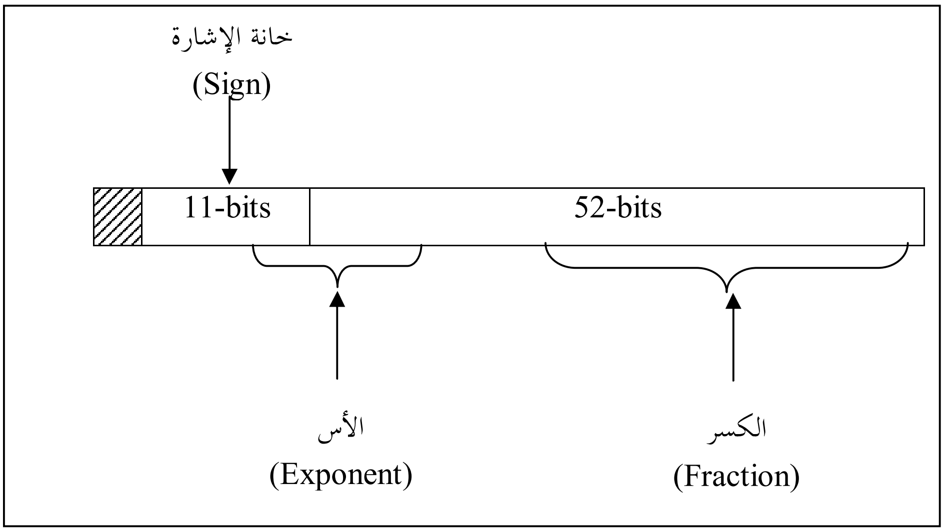
العدد الحقيقي ذو الدقة العادية (Single Precision)

طوله هو 4 Bytes = 32 bits ، مقسمة على النحو التالي:



العدد الحقيقي ذو الدقة المضاعفة (Double Precision)

طوله هو 8 Bytes = 64 bits ، مقسمة على النحو التالي:



عند وضع الكسر في الجزء المخصص له تتم محاذاته إلى اليسار، مع ملء الخانات الفائضة على يمينه بأصفار. أما الأس فيتم عند وضعه في الجزء المخصص له التعامل معه كعدد صحيح بإشارة (Signed Number)، أي تتم محاذاته إلى اليمين مع ملء الخانات الفائضة على يساره بإشارته (راجع تمثيل الأعداد الصحيحة ذات الإشارة). أما خانة الإشارة فنضع فيها القيمة 0 إذا كان العدد الحقيقي موجباً و القيمة 1 إذا كان العدد سالباً.

إذا أردنا تمثيل العدد الحقيقي 0.1101011×2^4 في صورة عدد حقيقي ذو دقة عادية (Single Precision) فإن ذلك يتم كالتالي:

نقوم أولاً بتحويل الأس من الصورة العشرية للصورة الثنائية فنحصل على $4 = (100)_2$

نضع كل من الكسر و الأس في الجزء المخصص له

نضع الإشارة

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

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

الخانات المظللة هنا هي الخانات الفائضة. فالخانات الفائضة على يمين الكسر يتم ملؤها بأصفار دائماً، أما الخانات الفائضة على يسار الأس فقد يتم ملؤها بأصفار (كما هو الحال هنا) إذا كان الأس موجباً، و قد يتم ملؤها بـــ1's إذا كان الأس سالباً (راجع تمثيل الأعداد الصحيحة ذات الإشارة).

إذا أردنا تمثيل نفس القيمة في صورة عدد حقيقي بدقة مضاعفة (Double Precision) يكون ذلك كالتالي:

مثال:

وضح طريقة تمثيل القيمة 25.687- في صورة عدد حقيقي:

(أ) ذو دقة عادية

(ب) ذو دقة مضاعفة.

الخطوة الأولى هي التحويل من الصورة العشرية للصورة الثنائية

$$25.6875 = (11001.1011)_2$$

الخطوة الثانية هي التطبيع

$$11001.1011 = 0.110011011 \times 2^5$$

نقوم بتحويل الأس من الصورة العشرية إلى الصورة الثنائية

$$5 = (101)_2$$

الخطوة الثالثة و الأخيرة هي التخزين:

الدقة العادية:

1	00000101	11001101100000000000000

الدقة المضاعفة:

1	00000000101	1100110110000000000000000000000000000000000000

0	00000000100	11010110000000000000000000000000000000000000000 0000

لاحظ أن العدد الحقيقي في هذا المثال سالب لذلك قمنا بوضع القيمة 1 في خانة الإشارة.

مثال:

وضح طريقة تمثيل القيمة 0.1 في صورة عدد حقيقي:

(أ) ذو دقة عادية

(ب) ذو دقة مضاعفة.

الخطوة الأولى هي التحويل من الصورة العشرية للصورة الثنائية
و قد قمنا بتحويل القيمة 0.1 من الصورة العشرية إلى الصورة الثنائية في مثال سابق و كانت النتيجة

$$0.1 = (0.000110011001100\cdots)_2$$

و لاحظنا حينذاك أن الكسر الثنائي الناتج غير منتهي

الخطوة الثانية هي التطبيع

$$0.000110011001100\cdots = 0.110011001100\cdots \times 2^{-3}$$

نقوم بتحويل الأس من الصورة العشرية إلى الصورة الثنائية $3 = (11)_2$
و لأن الكسر سالب يجب إيجاد مكمل الاثنينات (2's Complement) له

العدد (في 8 خانات)	00000011
المكمل الأول	11111100
+	1
المكمل الثاني	11111101

أي أن $-3 = (11111101)_2$
الخطوة الثالثة و الأخيرة هي التخزين:
لاحظ أننا نحتاج لملء الجزء المخصص للكسر إلى 23 خانة من خانات الكسر غير المنتهي في حالة الدقة العادية، و إلى 52 خانة في حالة الدقة المضاعفة، هذا بخلاف الأصفار الثلاثة الأولى يمين الفاصلة و التي يتم التخلص منها في عملية التطبيع. ففي عملية التحويل يجب أن نستمر في الضرب في 2 حتى نحصل على كل الخانات المطلوبة.

لكن إذا عدت إلى المثال الذي قمنا فيه بتحويل القيمة 0.1 من الصورة العشرية إلى الصورة الثنائية تجد أننا قد توقفنا عن الضرب في 2 بعد الحصول على 15 خانة فقط من خانات الكسر غير المنتهي. وسبب توقفنا هو أننا نلاحظ بعد ذلك أن الكسر عبارة عن كسر دوري (periodic) يتكرر فيه النمط 1100، بعد الأصفار الثلاثة الأولى يمين الفاصلة، باستمرار إلى ما لا نهاية. أي أن

$$0.1 = (0.0001100110011001100110011001100110011001100110011001100\cdots)_2$$

الدقة العادية:

0	11111101	11001100110011001100110

لاحظ أنه قد تم تخزين أول 23 خانة فقط من الكسر غير المنتهي و اقتطاع الباقي.

الدقة المضاعفة:

0	11111111101	1100110011001100110011001100110011001100110011001100

لاحظ أنه قد تم تخزين أول 52 خانة فقط من الكسر غير المنتهي و اقتطاع الباقي.

مثل هذا الاقتطاع للكسر الذي تم في المثال السابق يؤدي إلى حدوث خطأ (Error) في تمثيل العدد الحقيقي. فالقيمة المخزنة الآن ليست هي القيمة 0.1 تماما و إنما تقريب لها. و بالطبع فإن الخطأ في حالة الدقة المضاعفة أقل منه في حالة الدقة العادية، حيث تمكنا من تخزين عدد أكبر من خانات الكسر في حالة الدقة المضاعفة. و مثل هذه الأخطاء غالباً ما تحدث عند تمثيل الأعداد الحقيقية و يطلق عليها أخطاء التقريب (Round off Errors)، و هذه الأخطاء غير موجودة في تمثيل الأعداد الصحيحة.

تدريب 9

وضح طريقة تمثيل كل من القيم التالية في صورة عدد حقيقي ذو دقة عادية (Single Precision).

(1) 53.8125 (2) -0.1015625 (3) 0.7

حساب مقدار الخطأ في تمثيل العدد الحقيقي

ذكرنا أنه غالباً ما يكون هناك خطأ (Error) في تمثيل الأعداد الحقيقية، و أن الخطأ ناتج عن اقتطاع جزء من الكسر نظراً لمحدودية المساحة التخزينية المتاحة له.

سنقوم الآن بحساب مقدار الخطأ في تمثيل العدد العشري الحقيقي 0.1 في المثال السابق بالدقة العادية.

الأسلوب الذي سنتبعه في حساب مقدار الخطأ في التمثيل هو أن ننظر للقيمة المخزنة فعلاً و نقوم بتحويلها من الصورة الثنائية إلى الصورة العشرية، و نقارن النتيجة مع القيمة التي كان من المفترض تمثيلها. الفرق بين القيمتين يمثل مقدار الخطأ.

القيمة المخزنة هي

$$0.110011001100110011001100110 \times 2^{-3}$$

نقوم بإعادة الفاصلة إلى موقعها الأصلي

$$0.000110011001100110011001100110$$

نوجد رتب جميع الخانات الحاوية على 1 و هي 4-، 5-، 8-، 9-، 12-، 13-، 16-، 17-، 20-، 21-، 24-، 25-

وعليه تكون القيمة المخزنة هي

$$\begin{aligned} x &= 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-20} + 2^{-21} + 2^{-24} + 2^{-25} \\ &= 0.09999999404 \end{aligned}$$

أي ان مقدار الخطأ في التمثيل هو

$$0.1 - x = 0.1 - 0.09999999404 = 5.96 \times 10^{-9}$$

تدريب 10

احسب مقدار الخطأ في تمثيل القيمة 0.7 في صورة عدد حقيقي ذو دقة عادية (Single Precision)

مثال:

أوجد القيمة العشرية للعدد الثنائي

11111111110100000000000000000000

إذا علمت أنه يمثل عدداً حقيقياً ذو دقة عادية (IEEE Single Precision Float).

نقوم بتقسيم العدد إلى إشارة و أس و كسر

1	11111111	11010000000000000000000

العدد الحقيقي سالب لأن خانة الإشارة تحتوي على القيمة 1.

الأس عبارة عن عدد صحيح سالب لأن الخانة العليا (MSB) فيه مساوية 1. فبإيجاد مكمل الاثنينات له و تحويله إلى الصورة العشرية نجد أن الأس يساوي 1-.

وعليه فإن القيمة المخزنة هي 0.1101×2^{-1}

بإعادة الفاصلة إلى موقعها الأصلي يصبح العدد 0.01101

بالتحويل من الصورة الثنائية إلى الصورة العشرية نجد أن مقدار القيمة المخزنة هو

$$x = 2^{-2} + 2^{-3} + 2^{-5} = 0.40625$$

و بأخذ الإشارة السالبة في الاعتبار تكون القيمة المخزنة هي 0.40625-

مدى القيم التي يمكن تمثيلها فى صورة أعداد حقيقية

- الأعداد الحقيقية ذات الدقة العادية (Single Precision)

$$\pm 10^{-38} \quad - \quad \pm 10^{+38}$$

- الأعداد الحقيقية ذات الدقة المضاعفة (Double Precision)

$$\pm 10^{-308} \quad - \quad \pm 10^{+308}$$

أسئلة تقويم ذاتي

؟

1- في بدايات تعامل الحواسيب مع الأعداد الحقيقية (Real Numbers) حدث اختلاف ما بين الشركات الرئيسية المصنعة للحواسيب في ذلك الوقت في طريقة وضع المعلومات الممثلة للعدد الحقيقي (الكسر و الأس و الإشارة) داخل المساحة المخصصة لتخزين العدد الحقيقي، الأمر الذي أدى إلى حدوث مشاكل عند نقل البيانات ما بين تلك الحواسيب، وضح كيف تم تدارك هذا الاختلاف؟

2- ما هو طول العدد الحقيقي ذو الدقة العادية (Single Precision)؟

3- ما هو طول العدد الحقيقي ذو الدقة المضاعفة (Double Precision)؟

4- ما هي المعلومات المطلوب وضعها في المساحة المتاحة لتخزين الأعداد الحقيقية ؟

5- ما هي أنواع الأعداد الحقيقية حسب المواصفات القياسية ؟

6- ما المقصود بهذا المصطلح (Round off Errors)؟

3. تمثيل الرموز (Characters)

المقصود بالرموز (characters) هنا هو

الحروف الكبيرة (Capital Letters)   A, B, C, D, …, Z   (و عددها 26)

- الحروف الصغيرة (Small Letters)   **a, b, c, d, …, z**   (و عددها 26)
- الأرقام (Digits)   **0, 1, 2, 3, …, 9**   (و عددها 10)
- علامات الترقيم (Punctuation Marks)

~ { | } ’ _ ^ [\ ] @ ? < = > ; : / . - , + * ( ) ‘ & % $ # “ !   (و عددها 32)

- الرموز البيضاء (White Characters) مثل:

Space، **Horizontal Tab**، **Carriage Return**، **New Line**، ... (و عددها حوالي 6)

- رموز تحكم (Control Characters) مثل:

Del، **ESC**، **Bell**، **Back Space**، **Form Feed**، ... (و عددها حوالي 10)

أي أن العدد الكلي للرموز هو 26+26+10+32+6+10 أي حوالي **110** رمزاً

و يتم تمثيل هذه الرموز باستخدام شفرة ثنائية (Binary Code) بحيث يكون لكل رمز منها شفرة فريدة تميزه.

و اقل عدد من الخانات يلزم لتمثيل جميع الرموز هو 7 خانات (7 bits)، حيث أن عدد الشفرات الثنائية المتاحة في هذه الحالة هو $2^7 = 128$ و هذا العدد يكفي لتمثيل جميع الرموز.

أسئلة تقويم ذاتي

1- ما المقصود بالرموز وما هو العدد الكلي لها؟

2- ما هي الشفرة المستخدمة لتمثيل هذه الرموز ؟

4. شفرات تمثيل البيانات

توجد طرق عديدة يمكن بها أن يتم تخصيص الشفرات الثنائية المتاحة للرموز المختلفة، مما قد يؤدي إلى اختلافات كبيرة في تمثيل البيانات. و منعاً للاختلاف تم الاتفاق عالمياً على طرق محددة لتمثيل البيانات، و تم توثيق هذه الطرق في المؤسسات المعنية، و يتم مراجعتها و تطويرها و نشرها بانتظام لكي يلتزم الجميع بها. الأمر الذي جعل تبادل البيانات على نطاق واسع، خاصة في عصر الإنترنت، أمراً ممكناً.

سنتعرض في الجزء التالي لعدد من الشفرات القياسية (Standard Codes) المستخدمة حالياً في تمثيل البيانات.

شفرة ASCII

كلمة ASCII اختصار للعبارة (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). ونعني الشفرة الأمريكية القياسية لتبادل المعلومات و شفرة ASCII عبارة عن شفرة ثنائية مكونة من سبعة خانات تستخدم في تمثيل الرموز. و تعتبر الشفرة الأكثر استخداماً لهذا الغرض و الأوسع انتشاراً حالياً. تم ابتكار شفرة ASCII في الأساس لتمثيل الرموز في آلات تسمى التيلي تايب (Teletype Machines) وهي عبارة عن وسيلة اتصال استخدمت في السابق لنقل البيانات، و تتكون مما يشبه الآلتين الكاتبتين (Typewriters)، إحداهما مرسلة و الأخرى مستقبلة. عند طباعة أي نص على لوحة مفاتيح الآلة المرسلة يظهر ذلك النص مطبوعاً على الورق في الآلة المستقبلة. و يعتبر جهاز التلكس (Telex) مثالاً لهذا النوع من الآلات.

و يوضح الجدول التالي الشفرات الثنائية المستخدمة لتمثيل الرموز في شفرة ASCII:

Code		Char
Binary	**Hex**	
0000000	00	**nul**
0000001	01	**soh**
0000010	02	**stx**
0000011	03	**etx**
0000100	04	**eot**
0000101	05	**enq**
0000110	06	**ack**
0000111	07	**bel**
0001000	08	**bs**
0001001	09	**ht**
0001010	0A	**nl**
0001011	0B	**vt**
0001100	0C	**ff**
0001101	0D	**cr**
0001110	0E	**so**
0001111	0F	**si**
0010000	10	**dle**
0010001	11	**dc1**
0010010	12	**dc2**
0010011	13	**dc3**
0010100	14	**dc4**
0010101	15	**nak**
0010110	16	**syn**
0010111	17	**etb**
0011000	18	**can**
0011001	19	**em**
0011010	1A	**sub**
0011011	1B	**esc**
0011100	1C	**fs**
0011101	1D	**gs**
0011110	1E	**rs**
0011111	1F	**us**

Code		Char
Binary	**Hex**	
0100000	20	**sp**
0100001	21	**!**
0100010	22	**“**
0100011	23	**#**
0100100	24	**$**
0100101	25	**%**
0100110	26	**&**
0100111	27	**‘**
0101000	28	**(**
0101001	29	**)**
0101010	2A	*
0101011	2B	+
0101100	2C	**,**
0101101	2D	**-**
0101110	2E	**.**
0101111	2F	/
0110000	30	**0**
0110001	31	**1**
0110010	32	**2**
0110011	33	**3**
0110100	34	**4**
0110101	35	**5**
0110110	36	**6**
0110111	37	**7**
0111000	38	**8**
0111001	39	**9**
0111010	3A	**:**
0111011	3B	**;**
0111100	3C	<
0111101	3D	=
0111110	3E	>
0111111	3F	**?**

Code		Char
Binary	**Hex**	
1000000	40	@
1000001	41	**A**
1000010	42	**B**
1000011	43	**C**
1000100	44	**D**
1000101	45	**E**
1000110	46	**F**
1000111	47	**G**
1001000	48	**H**
1001001	49	**I**
1001010	4A	**J**
1001011	4B	**K**
1001100	4C	**L**
1001101	4D	**M**
1001110	4E	**N**
1001111	4F	**O**
1010000	50	**P**
1010001	51	**Q**
1010010	52	**R**
1010011	53	**S**
1010100	54	**T**
1010101	55	**U**
1010110	56	**V**
1010111	57	**W**
1011000	58	**X**
1011001	59	**Y**
1011010	5A	**Z**
1011011	4B	**[**
1011100	5C	\
1011101	5D	**]**
1011110	5E	^
1011111	5F	_

Code		Char
Binary	**Hex**	
1100000	60	**’**
1100001	61	**A**
1100010	62	**B**
1100011	63	**C**
1100100	64	**D**
1100101	65	**E**
1100110	66	**F**
1100111	67	**G**
1101000	68	**H**
1101001	69	**I**
1101010	6A	**J**
1101011	6B	**K**
1101100	6C	**L**
1101101	6D	**M**
1101110	6E	**N**
1101111	6F	**O**
1110000	70	**P**
1110001	71	**Q**
1110010	72	**R**
1110011	73	**S**
1110100	74	**T**
1110101	75	**U**
1110110	76	**V**
1110111	77	**W**
1111000	78	**X**
1111001	79	**Y**
1111010	7A	**Z**
1111011	7B	{
1111100	7C		
1111101	7D	}
1111110	7E	~
1111111	7F	**Del**

إذا قمنا بالتدقيق في الجدول السابق نتوصل إلى بعض الملاحظات الدالة على أن تخصيص الشفرات الثنائية للرموز المختلفة في شفرة ASCII لم يتم بصورة اعتباطية و إنما تم بطريقة حكيمة. لاحظ مثلاً العلاقة ما بين الشفرات الممثلة للأرقام (Digits) 9-0 و قيم تلك الارقام، تجد أن هناك فرقاً ثابتاً مقداره $(30)_{16}$ ما بين شفرة الرقم و قيمته، مما يسهل من عملية تحويل رموز الأرقام إلى القيم المقابلة لها، و هي عملية نحتاج لها كثيراً في الحاسوب و الأنظمة الرقمية الأخرى. لاحظ أيضاً وجود علاقة رياضية ثابتة ما بين شفرة ASCII للحرف الكبير (Capital Letter) ونظيره الصغير (Small Letter)، تجد أن الفرق بين شفرتيهما هو $(20)_{16}$، مما يجعل من عملية تحويل الأحرف الكبيرة إلى أحرف صغيرة أو العكس في نص معين عملية سهلة.

عندما استخدمت شفرة ASCII في تمثيل الرموز في الحاسوب، ظهرت مشكلة الخانة الثامنة (8th bit)، حيث أن التخزين فى الحواسيب مبني على نظام الـــByte المكون من 8 bits، في حين أن شفرة ASCII عبارة عن شفرة مكونة من سبعة خانات (7-bit Code)، لذلك كان لابد من إيجاد استخدام للخانة الثامنة. و هنالك طريقتان لاستغلال هذه الخانة:

1/ يمكن استخدام الخانة الثامنة لمضاعفة عدد الرموز التي يمكن تمثيلها بحيث يصبح 256 رمزاً بدلاً عن 128 رمزاً. هذه الـــ 256 رمزاً تكون الـــ 128 رمزاً الأولى منها هي رموز شفرة ASCII القياسية، أما الـــ 128 رمزاً الإضافية فيمكن استخدامها في تمثيل أحرف اللغات الأخرى، مثل اللغة العربية، أو في تمثيل بعض الرموز الخاصة المستخدمة مثلاً فى الرسومات أو في بناء الجداول أو في كتابة المعادلات الرياضية و غير ذلك.

2/ يمكن إستخدام الخانة الثامنة في عملية تسمى **فحص المثيل (Parity Check)**، وهي عملية تستخدم لاكتشاف حدوث خطأ (Error) في نقل البيانات. حيث أنه عند نقل البيانات لمسافات طويلة عبر وسائل الاتصال المختلفة قد تتعرض تلك البيانات لحدوث أخطاء. فلاكتشاف حدوث مثل هذه الأخطاء يتفق كل من الطرف المرسل للبيانات و

الطرف المستقبل لها على أن يكون العدد الكلى للـــ 1‘s في أي رمز مرسل فردياً مثلاً، و هو مايسمى **المثيل الفردي (Odd Parity)**. و بناء على ذلك يقوم الطرف المرسل قبل إرسال أي رمز بحساب عدد الـــ 1‘s الموجودة فيه، فإذا وجد أن عددها فردي يقوم بوضع 0 في الخانة الثامنة، وذلك للحفاظ على العدد الكلي للـــ 1‘s في الرمز فردياً. أما إذا وجد أن عدد الـــ 1‘s في الرمز المرسل زوجياً فإنه يقوم بوضع 1 في الخانة الثامنة، بحيث يصبح العدد الكلي للـــ 1‘s في الرمز فردياً. أي أن مهمة الطرف المرسل هي التأكد من أن عدد الـــ 1’s فردي في كل رمز يقوم بإرساله، و ذلك بوضع القيمة المناسبة في الخانة الثامنة، و التي يطلق عليها **خانة المثيل (Parity bit)**. أما بالنسبة للطرف المستقبل فإنه يقوم بحساب عدد الـــ 1‘s فى أي رمز يصل إليه، فإذا وجد أن عددها فردي كان معنى ذلك عدم حدوث خطأ أثناء عملية النقل، أما إذا وجد أن عددها زوجي فمعنى ذلك حدوث خطأ. و الطريقة الوحيدة الممكنة لتصحيح الخطأ الذي حدث هنا هي أن يطلب الطرف المستقبل من الطرف المرسل إعادة إرسال الرمز الذي وصله خاطئاً، وهذا يتطلب بالطبع وجود إمكانية الاتصال في الاتجاهين، و هو أمر غير متاح في كثير من الأحيان. لاحظ أن هذا الأسلوب فى اكتشاف حدوث الأخطاء يعجز عن اكتشاف حدوث خطأ فى خانتين فى وقت واحد. و لا توجد مشكلة هنا حيث أنه في أي نظام رقمي مصمم بصورة جيدة يكون احتمال حدوث خطأ في خانتين في وقت واحد أمراً نادر الحدوث بحيث يمكن تجاهله.

يمكن أيضاً أن يتفق الطرفان المرسل والمستقبل على أن يكون العدد الكلى للـــ 1‘s في أي رمز مرسل زوجياً، و يسمى هذا **بالمثيل الزوجي (Even Parity)**.

شفرة BCD (Binary Coded Decimal)

استخدمت هذه الشفرة في الماضي لتمثيل الأعداد الصحيحة (Integers) في الحواسيب الكبيرة (Main Frames) القديمة، خاصة تلك التي قامت بإنتاجها شركة IBM. في هذه الشفرة يتم تمثيل كل رقم من الأرقام من 0-9 باستخدام شفرة ثنائية مكونة من أربعة خانات (4-bits Binary Code)، و ذلك كالتالي:

الرقم (Digit)	الشفرة (Code)
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

لاحظ أن الخانات الأربعة المستخدمة في التمثيل هنا تعطينا 16 شفرة (Code) مختلفة، استخدمنا منها فقط العشرة الأولى و تبقت 6 شفرات غير مستخدمة هي: 1010، 1011، 1100، 1101، 1110، 1111.

لتمثيل أي عدد صحيح باستخدام شفرة BCD نأخذ أرقام العدد فى الصورة العشرية و نستبدل كل رقم بشفرة BCD الخاصة به. مثلاً:

831

1000 0011 0001

بتجميع شفرات BCD للأرقام فنحصل على $831 = (100000110001)_{BCD}$

لاحظ أن الأعداد الصحيحة الممثلة فى صورة BCD تشغل مساحة تخزينية أكبر من تلك التي تشغلها الأعداد الصحيحة الممثلة بالصورة التقليدية التى سبق لنا دراستها. كما أن إجراء العمليات الحسابية على الأعداد الممثلة فى صورة BCD به الكثير من المشاكل والصعوبات والتعقيدات.

تدريب 11

ما هو أكبر عدد يمكن تخزينه في 1 Byte باستخدام شفرة BCD؟

شفرة EBCDIC (Extended Binary Coded Decimal Information Code)

هذه الشفرة هي عبارة عن تطوير لشفرة BCD بحيث تتمكن من تمثيل الرموز. و هي تشبه إلى حد كبير شفرة ASCII، إلا أن شفرة EBCDIC مكونة من 8 خانات (8 bits).
استخدمت شفرة EBCDIC لتمثيل الرموز فى الحواسيب الكبيرة (Main Frames) التي تنتجها شركة IBM. و ما زالت إمكانية التعامل مع البيانات الممثلة باستخدام شفرة EBCDIC موجودة حتى الآن في الحواسيب التي تقوم بإنتاجها شركة IBM وذلك لتمكين مستخدمي هذه الأجهزة من الرجوع لبياناتهم القديمة.

الشفرة الرمادية (Gray Code)

يطلق على هذه الشفرة أيضاً تسمية **الشفرة المعكوسة (Reflected Code)**، و ذلك بسبب الأسلوب المستخدم في توليدها. تمتاز هذه الشفرة بأن كل رمزين متتاليين فيها يختلفان عن بعضهما البعض في bit واحد فقط. و يمكن أن نقوم بتوليد الشفرة الرمادية كالتالي:

شفرة رمادية من خانة واحدة:

الشفرة الرمادية المكونة من خانة واحدة هي نفس الشفرة الثنائية المكونة من خانة واحدة، أي

0
1

شفرة رمادية من خانتين:

لتوليد شفرة رمادية مكونة من خانتين نبدأ بالشفرة الرمادية المكونة من خانة واحدة، أي

0
1

نقوم بإجراء عملية عكس (Reflection) لهذه الشفرة (على سطح عاكس وهمي) فنحصل على

0
1
1
0

السطح العاكس (الوهمي)

أخيراً نقوم بملء الخانة الثانية الواقعة إلى اليسار بـــ 0's أعلى السطح العاكس، و بـــ 1's أسفله، فنحصل على

0 0
0 1
1 1
1 0

و بذلك تكون الشفرة الرمادية المكونة من خانتين هي

00
01
11
10

قارن مع الشفرة الثنائية المكونة من خانتين و هي

00
01
10
11

لاحظ الميزة الأساسية للشفرة الرمادية، و هي أنه عند الانتقال من قيمة إلى القيمة التي تليها فإن خانة واحدة فقط تتغير من 0 إلى 1 أو من 1 إلى 0. أما في الشفرة الثنائية فإنه عند الانتقال من 01 إلى 10 فإن كلا الخانتين تتغيران. و المشكلة هنا أنه من الصعب جداً في الأنظمة الرقمية أن نضمن حدوث التغير في الخانتين في نفس اللحظة، و غالباً ما تتغير إحدى الخانتين قبل الأخرى، مما قد ينتج عنه ظهور قيم وسطية غير صحيحة. مثلاً عند الانتقال من 01 إلى 10 إذا تغيرت الخانة اليمنى قبل الخانة اليسرى تصبح الخانة اليمنى 0 في حين أن الخانة اليسرى لم تزل 0 و لم تتغير بعد إلى 1 فتظهر القيمة الوسطية الخاطئة 00 لفترة زمنية قصيرة جداً، ثم تتغير الخانة اليسرى من 0 إلى 1 فتظهر القيمة الصحيحة 10.

0 1

↓

0 0

↓

1 0

قيمة وسطية غير صحيحة

و مثل هذه القيم الوسطية غير الصحيحة (Incorrect Intermediate Values) قـــد تـــؤدي إلى حدوث مشاكل في الأنظمة الرقمية على الرغم من فترة ظهورها القصيرة جـــداً. و في الشفرة الرمادية لا يمكن ظهور مثل هذه القيم.

شفرة رمادية من ثلاثة خانات:

لتوليد شفرة رمادية مكونة من ثلاثة خانات نبدأ بالشفرة الرمادية المكونة من خانتين التي قمنا بتوليدها، أي

0	0
0	1
1	1
1	0

نقوم بإجراء عملية عكس (Reflection) لهذه الشفرة (على سطح عاكس وهمي) فنحصل على

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

أخيراً نقوم بملء الخانة الثالثة الواقعة إلى اليسار بـــ 0's أعلى السطح العاكس، و بـــ 1's أسفله، فنحصل على

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

و بذلك تكون الشفرة الرمادية المكونة من ثلاثة خانات هي

000
001
011
010
110
111
101
100

شفرة رمادية من أربعة خانات:

لتوليد شفرة رمادية مكونة من أربعة خانات نبدأ بالشفرة الرمادية المكونة من ثلاثة خانات التي قمنا بتوليدها، أي

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

نقوم بإجراء عملية عكس (Reflection) لهذه الشفرة (على سطح عاكس وهمي) فنحصل على

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

أخيراً نقوم بملء الخانة الرابعة الواقعة إلى اليسار بـــ 0's أعلى السطح العاكس، و بـــ 1's أسفله، فنحصل على

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

تستخدم الشفرة الرمادية فى التطبيقات الصناعية التي تستخدم فيها الأنظمة الرقمية في التحكم فى الآلات. وسبب استخدام الشفرة الرمادية هنا هو عدم إمكانية ظهور قيم وسطية خاطئة.

أسئلة تقويم ذاتي

1- ما هي الملاحظات الدالة على أن تخصيص الشفرات الثنائية للرموز المختلفة في شفرة ASCII لم يتم بصورة اعتباطية و إنما تم بطريقة حكيمة؟

2- ما هي المشكلة التي ظهرت عندما استخدمت شفرة ASCII في تمثيل الرموز في الحاسوب؟

3- كيف يتم تمثيل العدد الصحيح باستخدام شفرة BCD؟

الخلاصة

عزيزي الدارس، ها نحن قد وصلنا إلي نهاية الوحدة الأولى، و التي تناولنا فيها طريقة تمثيل أنواع البيانات المختلفة في الحاسوب و بقية الأنظمة الرقمية. جيث تعرفنا أولاً على طريقة تمثيل الأعداد الصحيحة بأنواعها المختلفة، قصيرة (Short) أو عادية (Integer) أو طويلة (Long)، بإشارة (Signed) أو بدون إشارة (Unsigned). كما حددنا مدى القيم التي يقبلها كل نوع. و عرفنا أن الأعداد الصحيحة يتم تمثيلها بدقة كاملة دون أي خطأ طالما أن عدد الخانات المتاحة يكفي لتمثيل القيمة. ثم انتقلنا إلى الأعداد الحقيقية و درسنا طريقة تمثيلها في الحاسوب أو الأنظمة الرقمية وذلك بتحويله أولاً من الصورة العشرية (Decimal) إلى الصور الثنائية (Binary)، و عرفنا أنه من غير الممكن تمثيلها بدقة كاملة نظراً لمحدودية المساحة المتاحة لتخزينها. والمعلومات المطلوب وضعها في المساحة المتاحة هي :

1- الكسر (Fraction). 2- الأس (Exponent). 3- الإشارة (Sign).

قامت جمعية IEEE بوضع مواصفات قياسية لتخزين الأعداد الحقيقية وتم في هذه المواصفات تعريف نوعين من الأعداد الحقيقية:

1- العدد الحقيقي ذو الدقة العدية (Single Precision).

2- العدد الحقيقي ذو الدقة المضاعفة (Double Precision).

و خلصنا إلى أنه كلما زاد حجم المساحة المتاحة لتخزين العدد، سواء كان عدد صحيح أو عدد حقيقي، فإننا نحصل على إمكانية التعامل مع قيم أكبر، كما نحصل علي دقة أفضل في حالة الأعداد الحقيقية. وعند حساب مقدار الخطأ في تمثيل العدد الحقيقي ننظر للقيمة المخزنة فعلاً ونقوم بتحويلها من الصورة الثنائية الى الصورة العشرية ، ونقارن النتيجة مع القيمة التي كان من المفترض تمثيلها ، الفرق بين القمتين يمثل مقدار الخطأ.

بعد ذلك تعرضنا لبعض أنواع الشفرات المستخدمة لتمثيل البيانات في الحاسوب، حيث وضحنا الخصائص العامة لكل شفرة.

لمحة مسبقة عن الوحدة التالية

في الوحدة الثانية سنتعرف على أساسيات الجبر البولي (Boolean Algebra)، و هو جبر المتغيرات المنطقية. و المتغيرات المنطقية هو نوع المتغيرات الذي يتم التعامل معه في الدوائر المنطقية (Logic Circuits)، التي تسمى أيضاً بالدوائر الرقمية (Digital Circuits). حيث نتعرف على المتغير المنطقي (Logical Variable)، و العمليات المنطقية الأساسية الثلاث NOT و AND و OR، و البوابات المنطقية (Logic Gates) التي تقوم بإجراء هذه العمليات، و هذه البوابات هي الوحدات الأساسية المستخدمة في بناء أي نظام رقمي. كما نتعرف على التعبير المنطقي (Logical Expression) و جدول الصواب (Truth Table). وفي نهاية الوحدة نتعرف على بعض نظريات الجبر البولي و استخدامها في تبسيط التعبيرات المنطقية. و تبسيط التعبيرات المنطقية هي خطوة مهمة جداً في تصميم الدوائر المنطقية، و هو موضوع الوحدة الثالثة.

إجابات التدريبات

تدريب 1: (1) $55 = (110111)_2$

(2) $200 = (11001000)_2$

(3) $300 = (100101100)_2$

(4) $600 = (1001011000)_2$

تدريب 2: (1) $(10111011)_2 = 187$

(2) $(10000000)_2 = 128$

(3) $(11100111)_2 = 231$

(4) $(11111111)_2 = 255$

تدريب 3: لا نستخدم مدى من القيم، و بالتالي طولاً، أكثر مما نحتاج إليه توفيراً للمساحة التخزينية، حيث أنها محدودة دوماً في الأنظمة الرقمية.

تدريب 4: (1) $+15 = (00001111)_2$

$-15 = (11110001)_2$

(2) $+65 = (01000001)_2$

$-65 = (10111111)_2$

(3) $+1 = (00000001)_2$

$-1 = (11111111)_2$

(4) $+127 = (01111111)_2$

$-127 = (10000001)_2$

تدريب 5:
(1) $(10111111)_2 = -65$
(2) $(10000000)_2 = -128$
(3) $(01111111)_2 = +127$
(4) $(11111111)_2 = -1$

تدريب 6: (أ) لا تحدث أي مشكلة.

(ب) يحدث Overflow لأن القيمة المراد تخزينها خارج المدى المحدد للعدد القصير الصحيح ذو الإشارة. فيتم تخزين القيمة 56- و ليس القيمة 200.

تدريب 7:
(1) $53.8125 = (110101.1101)_2$
(2) $0.1015625 = (0.0001101)_2$
(3) $0.7 = (0.101100110011001100\cdots)_2$

تدريب 8: :
(1) $53.8125 = (110101.1101)_2 = 0.1101011101 \times 2^6$
(2) $0.1015625 = (0.0001101)_2 = 0.1101 \times 2^{-3}$
(3)
$$0.7 = (0.101100110011001100\cdots)_2 = 0.101100110011001100\cdots \times 2^0$$

تدريب 9:

(1)

0	00000110	11010111010000000000000

(2)

1	11111101	11010000000000000000000

(3)

0	00000000	10110011001100110011001

تدريب 10: 7.15×10^{-8}

تدريب 11: 99

مسرد المصطلحات

- **(least Significant Bit) LSB**

الخانة الواقعة في أقصى اليمين في العدد الثنائي ويسمى بالخانة الدنيا وذلك لأنها الخانة الأقل وزناً .

- **(Most Significant Bit) MSB**

الخانة الواقعة في أقصى اليسار في العدد الثنائي ، ويسمى بالخانة العليا وذلك لأنها الخانة الأعلى وزناً .

- **: Right Justify _ Zero Fill**

هي عملية ملء الخانات الزائدة إلى اليسار بأصفار في المساحة المتاحة لتخزين العدد الثنائي وذلك عندما يكون طول العدد الثنائي أقل من المساحة المتاحة .

- **المكمل الأول (1,s Complement)**

هو الخطوة الأولي لإيجاد المكمل الثاني ، وفيها يتم عكس جميع خانات العدد الثنائي وذلك بتحويل أي 0 إلى 1 وتحويل أي 1 إلى 0 .

- **المكمل الثنائي (2,s Complement)**

هو إضافة واحد (1) للمكمل الأول . والمكمل الثاني لعدد ثنائي يمثل سالب ذلك العدد .

- **تمديد الإشارة (Sign Extension)**

هو زيادة طول العدد الصحيح ذو الإشارة بملء الخانات الفائضة إلى اليسار بإشارة العدد .

- **Mathematical Over Flow**

هو مصطلح يشير إلى المشكلة التي تظهر في تمثيل الأعداد الصحيحة عندما تكون القيمة المطلوب تخزينها خارج المدى المحدد للمساحة المتاحة .

•أسلوب الفاصلة الثابتة (Fixed Point)

هو أسلوب لتمثيل الأعداد الحقيقية يفترض وجود الفاصلة (Point) عند الحد الفاصل ما بين الجزء من المساحة المخصص للعدد الصحيح والجزء المخصص للكسر وموقع الفاصلة هنا ثابت (Fixed) .

• أسلوب الفاصلة المتحركة (Floating Point)

يقوم هذا الأسلوب في تمثيل الأعداد الحقيقية على التخلص من الجزء الصحيح من العدد الحقيقي بحيث يصبح العدد بأكمله عبارة عن كسر ويتم ذلك بتحريك أو إزاحة الفاصلة . وذلك لاستغلال المساحة التخزينية المتاحة للعدد الحقيقي بصورة أكثر كفاءة .

• التطبيع (Normalization)

هو عملية تحريك الفاصلة للتخلص من الجزء الصحيح من العدد الحقيقي في صورته" الثنائية " وتحويل العدد بأكمله إلى كسر.

•أخطاء التقريب (Round off Errors)

هو الخطأ الذي يحدث نتيجة الاقتطاع للكسر غير المنتهي عند القيام بعملية التخزين وذلك نظراً لمحدودية المساحة التخزينية المتاحة له مما يؤدي إلى حدوث خطأ (Error) في تمثيل العدد الحقيقي حيث تصبح القيمة المخزنة للكسر تقريبية وهذه الأخطاء غير موجودة في تمثيل الأعداد الصحيحة .

• شفرة ASCII

كلمة ASCII اختصار للعبارة (American Standard Code for Information Interchange). و شفرة ASCII عبارة عن شفرة ثنائية مكونة من سبعة خانات تستخدم في تمثيل الرموز. وتعتبر الشفرة الأكثر استخداماً لهذا الغرض و الأوسع انتشاراً حالياً.

• فحص المثيل (Parity Check)

وهي عملية تستخدم لاكتشاف حدوث خطأ (Error) في نقل البيانات.

- **شفرة BCD (Binary Coded Decimal)**

استخدمت هذه الشفرة في الماضي لتمثيل الأعداد الصحيحة (Integers) في الحواسيب الكبيرة (Main Frames) القديمة، خاصة تلك التي قامت بإنتاجها شركة IBM. في هذه الشفرة يتم تمثيل كل رقم من الأرقام من 0-9 باستخدام شفرة ثنائية مكونة من أربعة خانات (4-bits Binary Code

- **شفرة EBCDIC (Extended Binary Coded Decimal Information Code)**

هذه الشفرة هي عبارة عن تطوير لشفرة BCD بحيث تتمكن من تمثيل الرموز. و هي تشبه إلى حد كبير شفرة ASCII، إلا أن شفرة EBCDIC مكونة من 8 خانات (8 bits). استخدمت شفرة EBCDIC لتمثيل الرموز فى الحواسيب الكبيرة (Main Frames) التي تنتجها شركة IBM. و ما زالت إمكانية التعامل مع البيانات الممثلة باستخدام شفرة EBCDIC موجودة حتى الآن في الحواسيب التي تقوم بإنتاجها شركة IBM وذلك لتمكين مستخدمي هذه الأجهزة من الرجوع لبياناتهم القديمة .

- **الشفرة الرمادية (Gray Code)**

يطلق على هذه الشفرة أيضاً تسمية الشفرة المعكوسة (Reflected Code)، و ذلك بسبب الأسلوب المستخدم في توليدها. تمتاز هذه الشفرة بأن كل رمزين متتاليين فيها يختلفان عن بعضهما البعض في bit واحد فقط.

المصادر و المراجع

1. M. Morris Mano, ***Digital Design***, Prentice-Hall, 2002
2. R. F. Tinder, ***Engineering Digital Design***, Academic Press, 2000
3. JohnWakerly, ***Digital Design: Principles and Practices***, Prentice-Hall, 2000
4. Daniel D. Gajski, ***Principles of Digital Design,*** Prentice Hall, 1997
5. R. H. Katz, Benjamin/Cummings, ***Contemporary Logic Design***, 1994
6. J. P. Hayes, ***Introduction to Digital Logic Design***, Addison-Wesley, 1993


الوحدة الثانية
أساسيات الجبر البولي

محتويات الوحدة

المقدمة

تمهيد

مرحباً بك **عزيزي الدارس** في الوحدة الثانية من مقرر "أساسيات التصـــميم المنطقـــي". تتناول هذه الوحدة أساسيات الجبر البولي (Boolean Algebra)، و هو جبـــر المتغيـــرات المنطقية. و المتغيرات المنطقية هو نوع المتغيرات الذي يتم التعامل معه فـــي الـــدوائر المنطقية (Logic Circuits).في القسم الثاني من الوحدة نتناول العمليات المنطقيـــة وهـــي العمليات التي يمكن إجراؤها على المتغيرات المنطقية ، والعمليات هنا نوعان:

أساسية وهي :NOT و AND وOR

وعمليات غير أساسية مثل : NAND و NOR وXOR

وفي القسم الثالث سنتعامل مع بوابات منطقية بعدد من أطراف الدخل (Fan- In)أكبر أو أقل مما نحتاج اليه.القسم الرابع نعالج فيه مجموعة ن المتغيرات المنطقية المرتبطة مع بعضها البعض بعمليات منطقية ، في القسم الخامس سنقوم بتمثيل التعبير المنطقي بدائرة منطقية، أما القسم السادس سنوضح فيه متغيرات الدخل للدائرة المنطقية ومسمياتها والخرج ومسمياتها ، القسم السابع يوضح جدول الصواب الذي نعالج فيه احتمالات الدخل للدائرة المنطقية وقيم الخرج المقابل لكل منها، تناول القسم الثامن نظريات الجبر البولي وهو جبر المتغيرات المنطقية ، أخيراً قمنا باستخدام نظريات الجبر البولي في تبسيط التعبيرات المنطقية .

أهداف الوحدة

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على:

- **تكتب** التعبيرات المنطقية و التعامل معها.
- **تنشئ** جداول الصواب و استخدامها.
- **تشرح** نظريات الجبر البولي.
- **تبسط** التعبيرات المنطقية باستخدام النظريات.

1. المتغير المنطقي (Logical Variable)

المتغير المنطقي هو أي متغير يمكن أن يأخذ قيمة واحدة فقط من قيمتين. مثلاً:

صواب	أو	**خطأ**
True	أو	**False**
ON	أو	**OFF**
+5 Volts	أو	**0 Volts**
High	أو	**Low**
أبيض	أو	**أسود**
Male	أو	**Female**

يرمز لإحدى القيمتين بالرمز **1** و للقيمة الأخرى بالرمز **0** ... فأي متغير منطقي لا يمكن أن يأخذ إلا إحدى هاتين القيمتين، و لا يوجد أي احتمال ثالث. فإذا كان x متغير منطقي فإنه إما أن يكون $x = 1$ أو $x = 0$.

2. العمليات المنطقية (Logical Operations)

العمليات المنطقية هي **العمليات التي يمكن إجراؤها على المتغيرات المنطقية**. بعض هذه العمليات هي عمليات أساسية، و هي عمليات NOT و AND و OR، و بعضها عمليات غير أساسية، مثل عمليات NAND و NOR و XOR، و هذه العمليات يمكن التعبير عنها باستخدام العمليات الأساسية.

1.2 عملية NOT

يطلق عليها أيضاً عملية العكس المنطقي (Logical Inversion)، وفيها يكون الخرج عبارة عن معكوس الدخل، فإذا كان الدخل مساوياً 1 فإن الخرج يكون مساوياً 0، و إذا

كان الدخل مساوياً 0 فإن الخرج يكون مساوياً 1. يرمز للعملية بوضح خط فوق المتغير، مما يعني أنه معكوس.

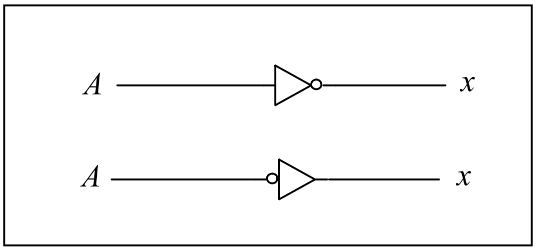
$$x = NOT\ A$$
$$x = \overline{A}$$

الجدول التالي يسمى جدول الصواب (Truth Table)، وهو جدول الصواب لعملية NOT، و جدول الصواب يوضح جميع احتمالات الدخل و الخرج المقابل لكل منها.

A	x
0	1
1	0

لاحظ أن الدخل هنا هو A و الخرج هو x. و الدخل في هذه الحالة عبارة عن متغير واحد يمكن أن يأخذ واحدة من قيمتين، إما 0 أو 1، أي أن هناك احتمالين فقط للدخل.

البوابة المنطقية (Logic Gate) التي تقوم بإجراء هذه العملية هي بوابة NOT (NOT Gate)، التي يطلق عليها أيضاً **العاكس المنطقي (Logic Inverter)**. و يمكن استخدام أي من الشكلين التاليين في تمثيل بوابة NOT:



2.2 عملية التكافؤ (Equivalence)

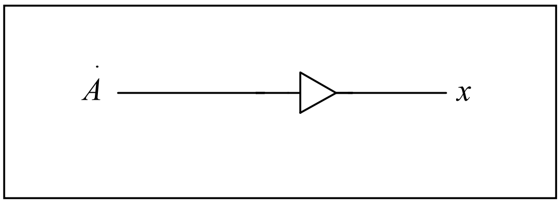
في هذه العملية يكون الخرج مساوياً للدخل، و يرمز لها بعلامة التساوي

$$x = A$$

وجدول الصواب للعملية هو

A	x
0	0
1	1

البوابة التي تقوم بإجراء هذه العملية تسمى العازل (Buffer)، و يتم تمثيلها بالشكل التالي:



3.2 عملية AND

في هذه العملية يكون الخرج مساوياً 1 فقط إذا كانت جميع متغيرات الدخل مساوية 1، و يكون الخرج يكون مساوياً 0 إذا كان أي متغير من متغيرات الدخل مساوياً 0. و يرمز لهذه العملية بأي من الطرق التالية

$$x = A \ AND \ B$$
$$x = A \cdot B$$
$$x = AB$$

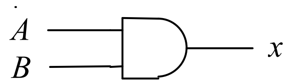
(الطريقة الأخيرة هي الأكثر استخداماً)

فيما يلي جدول الصواب لبوابة AND بمدخلين

A	B	x
0	0	0
0	1	0
1	0	0
1	1	1

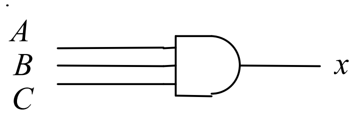
لاحظ أنه نظراً لوجود متغيرين للدخل هنا هما A و B، فأنه توجد أربعة احتمالات للدخل. و القاعدة العامة في جداول الصواب هي أنه إذا كان عدد متغيرات الدخل هو N فإن عدد احتمالات الدخل، أي عدد أسطر جدول الصواب، هو 2^N.

البوابة المنطقية التي تقوم بإجراء هذه العملية هي بوابة AND، و يرمز لها بالشكل التالي



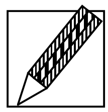
بوابة AND بمدخلين.
(2-Input AND Gate)

قد يكون لبوابة AND أكثر من مدخلين. مثلاً



بوابة AND بثلاثة مداخل
(3-Input AND Gate)

تدريب 1



قم بإنشاء جدول الصواب (Truth Table) لبوابة AND بثلاثة مداخل.

4.2 عملية OR

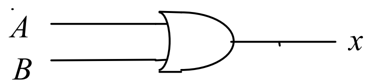
في هذه العملية يكون الخرج مساوياً 1 إذا كان أي من متغيرات الدخل مساوياً 1، و يكون الخرج يكون مساوياً 0 إذا كانت جميع متغيرات الدخل مساوية 0. و يرمز لهذه العملية بأي من الطريقتين التاليتين

$$x = A \; OR \; B$$
$$x = A + B$$

فيما يلي جدول الصواب لبوابة OR بمدخلين

A	B	x
0	0	0
0	1	1
1	0	1
1	1	1

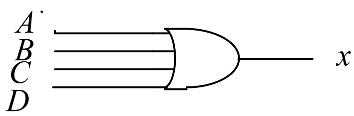
البوابة المنطقية التي تقوم بإجراء هذه العملية هي بوابة OR، و يرمز لها بالشكل التالي



بوابة OR بمدخلين.

(2-Input OR Gate)

قد يكون لبوابة OR أكثر من مدخلين. مثلاً



بوابة OR بأربعة مداخل

(4-Input OR Gate)

تدريب 2

قم بإنشاء جدول الصواب (Truth Table) لبوابة OR بأربعة مداخل.

5.2 عملية NAND

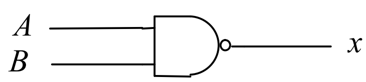
عملية NAND هي عبارة عن عملية AND متبوعة بعملية NOT، أي أنها عملية NOT AND، و يرمز لها بأي من الطرق التالية

$$x = A\ NAND\ B$$
$$x = \overline{A\ AND\ B}$$
$$x = \overline{A \cdot B}$$
$$x = \overline{AB}$$
$$x = A \uparrow B$$

الجدول التالي هو جدول الصواب لعملية NAND، وهو عكس عملية AND كما هو متوقع

A	B	x
0	0	1
0	1	1
1	0	1
1	1	0

البوابة المنطقية التي تقوم بإجراء هذه العملية هى بوابة NAND، و يرمز لها بالشكل التالي



بوابة NAND بمدخلين
(2-Input NAND Gate)

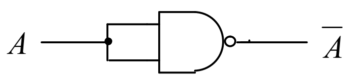
كفاية عملية NAND (Sufficiency of NAND)

المقصود بكفاية عملية NAND هو أن العمليات المنطقية الأساسية الثلاث (NOT، AND، OR) يمكن إجراؤها جميعاً باستخدام بوابات NAND، و بالتالي يمكن بناء أي دائرة منطقية بالكامل باستخدام بوابات NAND فقط.

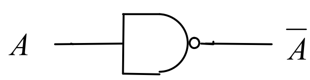
في الجزء التالي سنوضح طريقة إجراء العمليات المنطقية الأساسية الثلاث باستخدام بوابات NAND.

عملية NOT:

يمكن أن نقوم باستخدام بوابة NAND كعاكس منطقي بربط جميع أطراف الدخل لها في طرف واحد

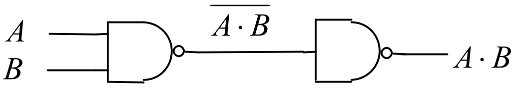


و يمكن أن نرمز لبوابة NAND المستخدمة كعاكس منطقي ببوابة NAND بطرف دخل واحد، أي



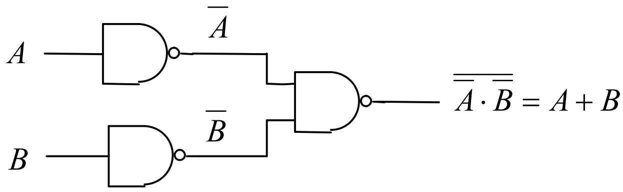
عملية AND:

يمكن إجراء عملية AND عن طريق إجراء عملية NAND متبوعة بعملية عكس منطقي



عملية OR:

يمكن إجراء عملية OR عن طريق إجراء عملية NAND مسبوقة بعملية عكس منطقي لكل طرف من أطراف الدخل



جدول الصواب التالي يثبت أن $\overline{\overline{A} \cdot \overline{B}} = A + B$

A	B	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$	$A + B$
0	0	1	1	1	0	0
0	1	1	0	0	1	1
1	0	0	1	0	1	1
1	1	0	0	0	1	1

6.2 عملية NOR

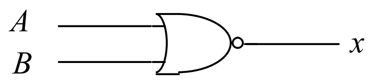
عملية NOR هي عبارة عن عملية OR متبوعة بعملية NOT، أي أنها عملية NOT OR، و يرمز لها بأي من الطرق التالية

$$x = A\ NOR\ B$$
$$x = \overline{A\ OR\ B}$$
$$x = \overline{A + B}$$
$$x = A \downarrow B$$

الجدول التالي هو جدول الصواب لعملية NOR، و هو عكس عملية OR كما هو متوقع

A	B	x
0	0	1
0	1	0
1	0	0
1	1	0

البوابة المنطقية التي تقوم بإجراء هذه العملية هى بوابة NOR، و يرمز لها بالشكل التالي



بوابة NOR بمدخلين.
(2-Input NOR Gate)

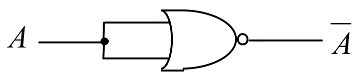
كفاية عملية NOR (Sufficiency of NOR)

المقصود بكفاية عملية NOR هو أن العمليات المنطقية الأساسية الثلاث (NOT، AND، OR) يمكن إجراؤها جميعاً باستخدام بوابات NOR، و بالتالي يمكن بناء أي دائرة منطقية بالكامل باستخدام بوابات NOR فقط.

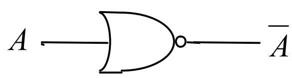
في الجزء التالي سنوضح طريقة إجراء العمليات المنطقية الأساسية الثلاث باستخدام بوابات NOR.

<u>عملية NOT:</u>

يمكن أن نقوم باستخدام بوابة NOR كعاكس منطقي بربط جميع أطراف الدخل لها في طرف واحد

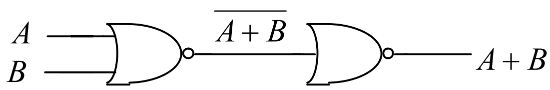


و يمكن أن نرمز لبوابة NOR المستخدمة كعاكس منطقي ببوابة NOR بطرف دخل واحد، أي



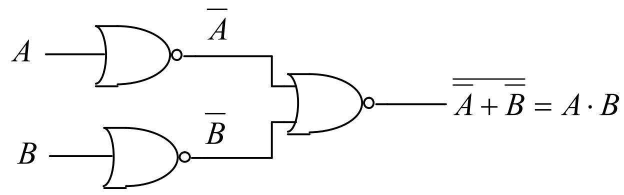
<u>عملية OR:</u>

يمكن إجراء عملية OR عن طريق إجراء عملية NOR متبوعة بعملية عكس منطقي



<u>عملية AND:</u>

يمكن إجراء عملية AND عن طريق إجراء عملية NOR مسبوقة بعملية عكس منطقي لكل طرف من أطراف الدخل



جدول الصواب التالي يثبت أن $\overline{\overline{A}+\overline{B}} = A \cdot B$

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$	$\overline{\overline{A}+\overline{B}}$	$A \cdot B$
0	0	1	1	1	0	0
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	0	1	1

و تتوفر بوابات NAND و بوابات NOR بأكثر من مدخلين، مثلها في ذلك مثل بوابات AND و بوابات OR.

أسئلة تقويم ذاتي

؟ ما المقصود بكفاية عملية NAND ، وبكفاية عملية NOR ؟

7.2 عملية XOR

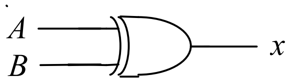
XOR هو اختصار لعبارة Exclusive OR، و تسمى عملية الاختلاف، حيث أن الخرج يساوي 1 إذا كان الدخلان مختلفين، و يساوي 0 إذا كانا متشابهين. و يرمز لها بإحدى الطريقتين التاليتين

$$x = A \; XOR \; B$$
$$x = A \oplus B$$

الجدول التالي هو جدول الصواب لعملية XOR

A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

البوابة المنطقية التي تقوم بإجراء هذه العملية هى بوابة XOR، و يرمز لها بالشكل التالي



بوابة XOR بمدخلين

(2-Input XOR Gate)

و يمكن التعبير عن عملية XOR باستخدام العمليات الأساسية الثلاث كالتالي

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

و جدول الصواب التالي يثبت أن $A \oplus B = \overline{A}B + A\overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$\overline{A}B$	$A\overline{B}$	$\overline{A}B + A\overline{B}$	$A \oplus B$
0	0	1	1	0	0	0	0
0	1	1	0	1	0	1	1
1	0	0	1	0	1	1	1
1	1	0	0	0	0	0	0

8.2 عملية XNOR

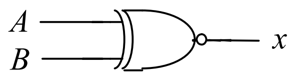
هي معكوس عملية XOR، و تسمى عملية التساوي، حيث أن الخرج يساوي 1 إذا كان الدخلان متساويين، و يساوي 0 إذا كانا مختلفين. و يرمز لها بإحدى الطريقتين التاليتين

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

الجدول التالي هو جدول الصواب لعملية XNOR

A	B	x
0	0	1
0	1	0
1	0	0
1	1	1

البوابة المنطقية التي تقوم بإجراء هذه العملية هى بوابة XNOR، و يرمز لها بالشكل التالي



بوابة XNOR بمدخلين.
(2-Input XNOR Gate)

و يمكن التعبير عن عملية XNOR باستخدام العمليات الأساسية الثلاث كالتالي

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

تدريب 3

قم بإنشاء جدول الصواب الذي يثبت أن $\overline{A \oplus B} = AB + \overline{A}\,\overline{B}$.

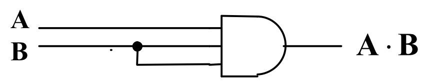
و بخلاف بوابات AND و OR و NAND و NOR، لا تتوفر بوابات XOR أو بوابات XNOR بأكثر من مدخلين.

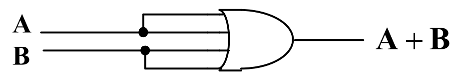
3. تغيير عدد أطراف الدخل (Fan-In) للبوابة المنطقية

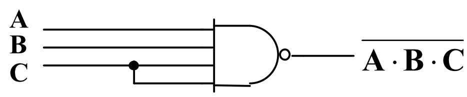
في كثير من الأحيان قد تتوفر لنا بوابات منطقية بعدد من أطراف الدخل (Fan-In) أكبر أو أقل مما نحتاج إليه. سنوضح في هذا الجزء الأساليب المختلفة التي يمكننا إتباعها لتغيير عدد أطراف الدخل للبوابة المنطقية بالزيادة أو بالنقصان.

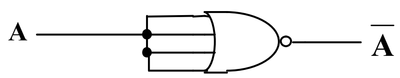
<u>تقليل عدد أطراف الدخل:</u>

يتم ذلك بربط طرف الدخل الزائد بأحد أطراف الدخل المستخدمة، مثلاً



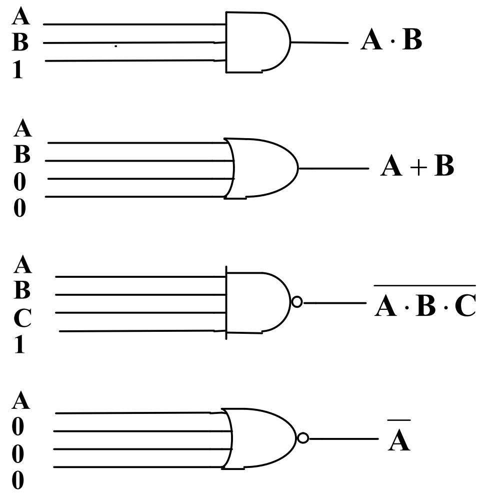






في الحالة الأولى استخدمنا بوابة AND بثلاثة مداخل كبوابة AND بمدخلين، و ذلك بالتخلص من طرف الدخل الثالث غير المرغوب فيه بربطه بأحد طرفي الدخل المستخدمين. و في الحالة الثانية استخدمنا بوابة OR بأربعة مداخل كبوابة OR بمدخلين.

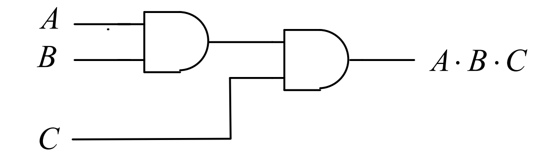
كما يمكن أن يتم التخلص من طرف الدخل الزائد بوضع القيمة المنطقية 1 في طرف الدخل الزائد في بوابات AND و NAND، و وضع القيمة المنطقية 0 في طرف الدخل الزائد في بوابات OR و NOR، مثلاً

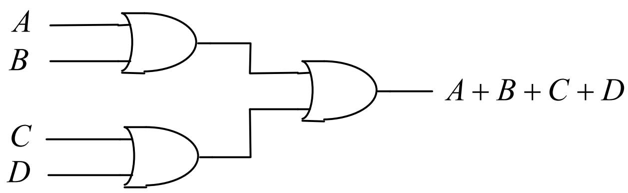


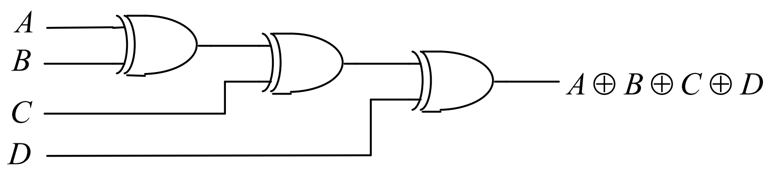
هنا أيضاً في الحالة الأولى استخدمنا بوابة AND بثلاثة مداخل كبوابة AND بمدخلين، و ذلك بالتخلص من طرف الدخل الثالث غير المرغوب فيه بوضع القيمة المنطقية 1 فيه. و في الحالة الثانية استخدمنا بوابة OR بأربعة مداخل كبوابة OR بمدخلين، و ذلك بوضع القيمة المنطقية 0 في طرفي الدخل الزائدين.

زيادة عدد أطراف الدخل:

يتم ذلك باستخدام أكثر من بوابة واحدة و استخدام خرج البوابة الأولى كدخل للبوابة الثانية، مثلاُ







في الحالة الأولى استخدمنا بوابتي AND، كل منهما بمدخلين، كوابة AND بثلاثة مداخل. و في الحالة الثانية استخدمنا ثلاثة بوابات OR، كل بوابة منها بمدخلين، كبوابة OR بأربعة مداخل. و في الحالة الثالثة استخدمنا ثلاثة بوابات XOR، كل بوابة منها بمدخلين، كبوابة XOR بأربعة مداخل.

4. التعبير المنطقي (Logical Expression)

التعبير المنطقي هو عبارة عن مجموعة من المتغيرات المنطقية المرتبطة مع بعضها البعض بعمليات منطقية. مثل

$$x = A + \overline{B} \cdot \overline{C}$$

يتكون التعبير المنطقي هنا من أربعة متغيرات هي A و B و C و x، تربط بينها عمليات NOT و AND و OR و عملية التكافؤ (=).

أسبقية إجراء العمليات (Operation Precedence):

يتم إجراء العمليات المنطقية الأساسية الثلاث بالترتيب التالي:

1- عملية العكس المنطقي NOT

2- عملية AND

3- عملية OR

ففي التعبير أعلاه، مثلاً، يتم أولاً إجراء عملية العكس المنطقي للمتغيرين B و C اولاً، ثم عملية AND بين $\overline{B}$ و $\overline{C}$، و أخيراً عملية OR.

في حالة ظهور عدة عمليات متساوية من حيث الأسبقية في التعبير المنطقي يتم إجراؤها بالترتيب من اليسار لليمين.

يمكن استخدام الأقواس للتحكم في ترتيب إجراء العمليات، حيث أن الأقواس لها الأسبقية العليا، أي أن ما بين الأقواس يتم حسابه دائماً أولاً. مثلاً إذا قمنا في التعبير السابق بإضافة قوسين كالتالي

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

فإنه يتم إجراء عملية OR الموجودة بين القوسين قبل عملية AND، و ذلك على الرغم من أن عملية AND لها أسبقية أعلى من عملية OR. و السبب في ذلك هو وجود عملية OR ما بين القوسين. حيث يتم أولاً حساب ما بين القوسين، فيتم إجراء عملية العكس المنطقي للمتغير B، ثم عملية OR بين A و $\overline{B}$، و بعد الانتهاء من الأقواس يتم إجراء العمليات خارجها، فيتم إجراء عملية العكس المنطقي للمتغير C، ثم عملية AND لما بين القوسين و $\overline{C}$.

أسئلة تقويم ذاتي

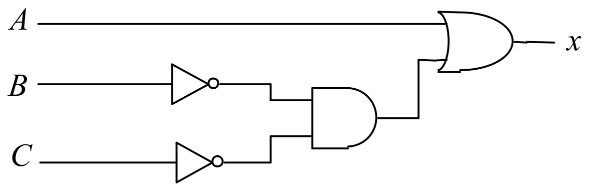
؟ ما هو التعبير المنطقي؟

5. الدائرة المنطقية (Logic Circuit)

يمكن تمثيل أي تعبير منطقي بدائرة منطقية، حيث ننظر للعمليات المنطقية الموجودة بالتعبير و نقوم بربط البوابات المنطقية التي تقوم بإجراء تلك العمليات بالأسلوب المناسب. مثلاً، التعبير المنطقي

$$x = A + \overline{B} \cdot \overline{C}$$

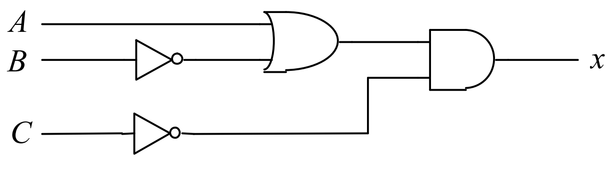
يمكن تمثيله بالدائرة المنطقية



و التعبير المنطقي

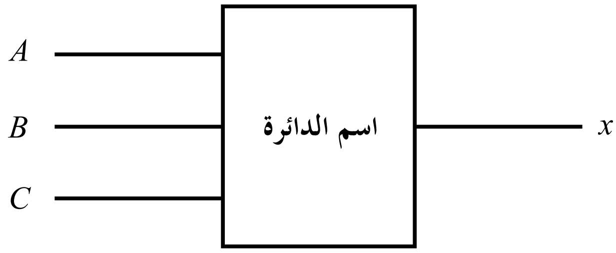
$$x = (A + \overline{B}) \cdot \overline{C}$$

يمكن تمثيله بالدائرة المنطقية



6. المخطط المنطقي (Logic Diagram)

هو عبارة عن مخطط مبسط يوضح متغيرات الدخل للدائرة المنطقية و مسمياتها و متغيرات الخرج ومسمياتها، بالإضافة إلى اسم الدائرة الدال على وظيفتها.
مثلاً، كلا الدائرتين المنطقيتين أعلاه يمكن تمثيلهما بالمخطط المنطقي التالي:



و نقوم باستخدام المخططات المنطقية كبديل للدائرة المنطقية المفصلة كنوع من التبسيط، و ذلك عندما لا نكون بحاجة للتفاصيل الداخلية للدائرة المنطقية. كما في الدوائر المعقدة المكونة من عدد من الدوائر الصغيرة المربوطة مع بعضها البعض، حيث نقوم بتمثيل تلك الدوائر الصغيرة بمخططاتها المنطقية.

7. جدول الصواب (Truth Table)

عبارة عن جدول يوضح جميع احتمالات الدخل للدائرة المنطقية و قيم الخرج المقابل لكل منها. مثلاً، لإنشاء جدول صواب للتعبير المنطقي

$$x = A + \overline{B} \cdot \overline{C}$$

نبدأ بتحديد عدد الصفوف و عدد الأعمدة في الجدول. متغيرات الدخل هي A و B و C، و عددها 3، أي عن عدد احتمالات الدخل هو $2^3 = 8$، و هو عدد أسطر (صفوف) جدول الصواب. أما عن الأعمدة فنحتاج عموداً لكل متغير من متغيرات الدخل و عموداً لكل متغير من متغيرات الخرج. متغيرات الدخل عددها 3، كما ذكرنا من قبل،

و هناك متغير خرج واحد هو x، أي نحتاج إلى أربعة أعمدة لمتغيرات الدخل و متغيرات الخرج. كما نحتاج إلى أعمدة إضافية لإجراء العمليات المنطقية، حيث نحتاج عموداً لإيجاد $\overline{B}$، و عموداً آخر لإيجاد $\overline{C}$، كما نحتاج عموداً لإيجاد $\overline{B}\cdot\overline{C}$، و أخيراً نحتاج عموداً لإيجاد $A+\overline{B}\cdot\overline{C}$، و هو في هذه الحالة نفس عمود الخرج x.

A	B	C	$\overline{B}$	$\overline{C}$	$\overline{B}\cdot\overline{C}$	$x=A+\overline{B}\cdot\overline{C}$
0	0	0	1	1	1	1
0	0	1	1	0	0	0
0	1	0	0	1	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	1	0	0	1
1	1	0	0	1	0	1
1	1	1	0	0	0	1

و بالمثل جدول الصواب للتعبير المنطقي

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

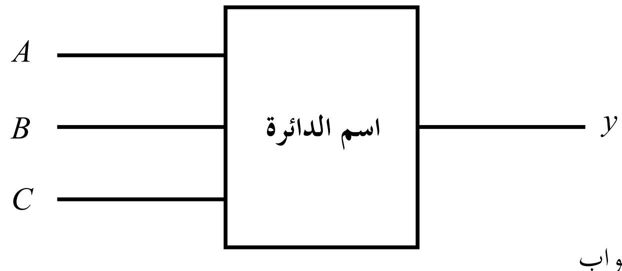
هو

A	B	C	$\overline{B}$	$\overline{C}$	$A+\overline{B}$	$x=(A+\overline{B})\cdot\overline{C}$
0	0	0	1	1	1	1
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	1	0	0	0	0
1	0	0	1	1	1	1
1	0	1	1	0	1	0
1	1	0	0	1	1	1
1	1	1	0	0	1	0

مثال:ارسم المخطط المنطقي، و أكمل جدول الصواب، ثم ارسم الدائرة المنطقية للتعبير المنطقي

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

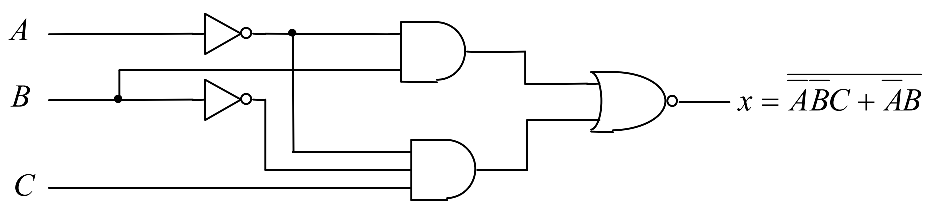
المخطط المنطقي



جدول الصواب

A	B	C	$\overline{A}$	$\overline{B}$	$\overline{A}\,\overline{B}C$	$\overline{A}B$	$\overline{A}\,\overline{B}C+\overline{A}B$	$x=\overline{\overline{A}\,\overline{B}C+\overline{A}B}$
0	0	0	1	1	0	0	0	1
0	0	1	1	1	1	0	1	0
0	1	0	1	0	0	1	1	0
0	1	1	1	0	0	1	1	0
1	0	0	0	1	0	0	0	1
1	0	1	0	1	0	0	0	1
1	1	0	0	0	0	0	0	1
1	1	1	0	0	0	0	0	1

الدائرة المنطقية



تدريب 4:

ارسم المخطط المنطقي، و أكمل جدول الصواب، ثم ارسم الدائرة المنطقية لكل تعبير من التعبيرات المنطقية التالية:

1- $x=\overline{A(\overline{B}+C)}$

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

3- $z=\overline{A\overline{B}+C\overline{D}}$

8. نظريات الجبر البولي (Boolean Algebra Theorems)

الجبر البولي هو جبر المتغيرات المنطقية، و الهدف الأساسي من دراستنا لنظريات الجبر البولي هو استخدام تلك النظريات في تبسيط التعبيرات المنطقية.

لكل نظرية (Theorem) من نظريات الجبر البولي نظرية مقابلة أو مناظرة لها (Dual Theorem). و للحصول على النظرية المقابلة لأي نظرية نقوم بإجراء التبديلات التالية في النظرية الأصلية:

- استبدال أي 0 بـــ 1
- استبدال أي 1 بـــ 0
- استبدال أي عملية AND بعملية OR
- استبدال أي عملية OR بعملية AND

وعموماً يمكن إثبات صحة أي نظرية باستخدام جداول الصواب. الجدول التالي يوضح النظريات الأساسية المستخدمة في الجبر البولي

اسم النظرية	النظرية	النظرية المقابلة
عكس العكس	$\overline{\overline{A}} = A$	$\overline{\overline{A}} = A$
العمليات مع 1 و 0	$A + 1 = 1$ $A + 0 = A$	$A \cdot 0 = 0$ $A \cdot 1 = A$
المتغير مع نفسه	$A + A = A$	$A \cdot A = A$
المتغير مع عكسه	$A + \overline{A} = 1$	$A \cdot \overline{A} = 0$
النظرية الإبدالية	$A + B = B + A$	$A \cdot B = B \cdot A$
النظرية التجميعية	$(A + B) + C = A + (B + C)$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$
النظرية التوزيعية	$A \cdot (B + C) = A \cdot B + A \cdot C$	$A + B \cdot C = (A + B) \cdot (A + C)$
الامتصاص أو الابتلاع	$A + A \cdot B = A$ $A + \overline{A} \cdot B = A + B$	$A \cdot (A + B) = A$ $A \cdot (\overline{A} + B) = A \cdot B$
دي مورغـــان (De Morgan)	$\overline{A + B} = \overline{A} \cdot \overline{B}$	$\overline{A \cdot B} = \overline{A} + \overline{B}$

أسئلة تقويم ذاتي

> ❓ ما هو الهدف الأساسي من دراسة نظريات الجبر

9. استخدام نظريات الجبر البولي فى تبسيط التعبيرات المنطقية

الهدف من تبسيط التعبير المنطقي هو تبسيط الدائرة المنطقية، أي تقليل عدد البوابات المنطقية الداخلة فى بنائها، و ذلك لتقليل تكلفتها. كما يعتبر تقليل تفرع الدخل للبوابات المنطقية المستخدمة فى بناء الدائرة نوعاً من التبسيط أيضاً.

سنعرض في هذا الجزء عدداً من الأمثلة لتوضيح طريقة تبسيط التعبيرات المنطقية باستخدام النظريات. و نذكر القارئ بضرورة التدرب على عملية التبسيط بفهم الأمثلة جيداً و إعادة حلها و حل التدريبات.

<u>مثال:</u>

استخدم نظريات الجبر البولي فى تبسيط التعبير المنطقي

$$y = \overline{\overline{A}\,\overline{B}C + \overline{A}B}$$

ثم ارسم الدائرة المنطقية قبل التبسيط و بعده.

<u>الحل:</u>

	$y = \overline{\overline{A}\,\overline{B}C + \overline{A}B}$
دي مورغان	$y = (\overline{\overline{A}} + \overline{\overline{B}} + \overline{C}) \cdot (\overline{\overline{A}} + \overline{B})$
عكس العكس	$y = (A + B + \overline{C}) \cdot (A + \overline{B})$
التوزيعية	$y = A + (B + \overline{C}) \cdot \overline{B}$
الابتلاع	$y = A + \overline{C}\,\overline{B}$

حل آخر:

$$y = \overline{\overline{A}\,\overline{B}C + \overline{A}B}$$

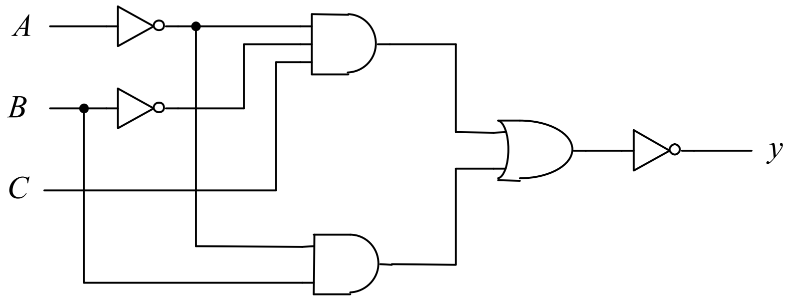
التوزيعية $y = \overline{\overline{A} \cdot (\overline{B}C + B)}$

الابتلاع $y = \overline{\overline{A} \cdot (C + B)}$

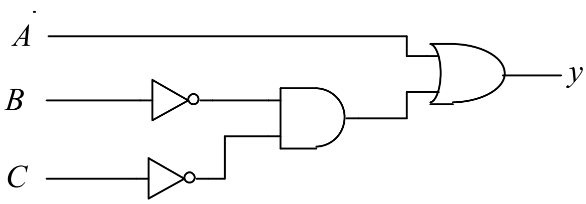
دي مورغان $y = \overline{\overline{A}} + \overline{C}\,\overline{B}$

عكس العكس $y = A + \overline{C}\,\overline{B}$

الدائرة قبل التبسيط:



الدائرة بعد التبسيط:



لاحظ أن الدائرة قبل التبسيط مكونة من 6 بوابات، و بعد التبسيط أصبحت مكونة من 4 بوابات فقط.

مثال:

استخدم نظريات الجبر البولي فى تبسيط التعبير المنطقي

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

ثم ارسم الدائرة المنطقية قبل التبسيط و بعده.

<u>الحل:</u>

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

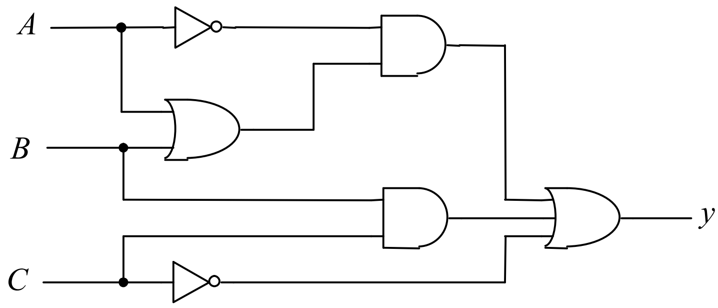
$$y = \overline{A}B+\overline{C}+CB \qquad \text{الابتلاع}$$

$$y = \overline{A}B+\overline{C}+B \qquad \text{الابتلاع}$$

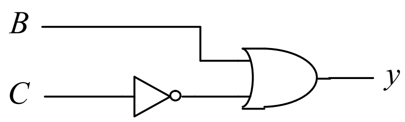
$$y = \overline{A}B+B+\overline{C} \qquad \text{الإبدالية}$$

$$y = B+\overline{C} \qquad \text{الابتلاع}$$

الدائرة قبل التبسيط



الدائرة بعد التبسيط



<u>مثال:</u>

استخدم نظريات الجبر البولي في تبسيط التعبير المنطقي

$$y = \overline{A}\overline{B}\overline{C}+\overline{A}\overline{B}C+\overline{A}BC+ABC$$

<u>الحل:</u>

لهذا المثال أهمية خاصة، و ذلك نظراً إلى أن التعبير المنطقي يظهر في صورة مميزة تسمى **صورة مجموع الحدود الصغرى (Sum of minterms)**. و في هذه الصورة يتكون التعبير المنطقي من مجموعة من الحدود المرتبطة مع بعضها البعض بعمليات OR. و يسمى كل حد منها بالحد الأصغر (minterm). و الحد الأصغر تظهر فيه

جميع متغيرات الدخل مرتبطة مع بعضها البعض بعمليات AND، و يكون بعض هذه المتغيرات معكوساً و بعضها الآخر غير معكوس.

لتبسيط هذا النوع من التعبيرات نبحث عن التشابهات ما بين الحدود. و الحدان المتشابهان هما حدين يتفقان فى كل شيء عدا متغير واحد يظهر فى أحدهما معكوساً و في الآخر بدون عكس. مثلاً، في التعبير أعلاه الحد الأول $\overline{A}\overline{B}\overline{C}$ يشبه الحد الثاني $\overline{A}\overline{B}C$، حيث يتفق الحدان في كل شيء عدا المتغير C الذي يظهر في الحد الأول معكوساً و في الحد الثاني بدون عكس. و بنفس الطريقة يتشابه الحدان الثالث $\overline{A}BC$ و الرابع ABC، حيث يتفقان في كل شيء عدا المتغير A الذي يظهر في الحد الثالث معكوساً و في الحد الرابع بدون عكس.

$$y = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + ABC$$

ملاحظة

الاختلاف ما بين الحدين المتشابهين يجب أن يكون في متغير واحد فقط و لا يجوز أن يكون في أكثر من متغير.

بعد إيجاد التشابهات ما بين الحدود نقوم بجمع كل حدين متشابهين في حد واحد هو عبارة عن العامل المشترك ما بين الحدين، أما المتغير المختلف فيتم اختصاره.

$$y = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}BC + ABC$$

بإخراج العامل المشترك في كل حدين متشابهين $\quad y = \overline{A}\overline{B}(\overline{C} + C) + BC(\overline{A} + A)$

بجمع المتغير مع عكسه $\quad y = \overline{A}\overline{B}(1) + BC(1)$

بالعمليات مع 1 $\quad y = \overline{A}\overline{B} + BC$

لاحظ في المثال السابق وجود تشابه إضافي بين الحدود، حيث أن الحد الثاني $\overline{A}\overline{B}C$ يشبه الحد الثالث $\overline{A}BC$، و لكن لم نكن في حاجة لاستخدام هذا التشابه في عملية التبسيط.

مثال:

استخدم نظريات الجبر البولي في تبسيط التعبير المنطقي

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

الحل:

التعبير هنا في صورة مجموع الحدود الصغرى، لذلك نبحث عن التشابهات ما بين الحدود.

الحد الأول يشبه الحد الثاني، و الحد الرابع يشبه الحد الخامس، و الحد الثالث يشبه الحد الأول.

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

نلاحظ هنا وجود مشكلة تتمثل في أن الحد الأول يتشابه في نفس الوقت مع كل من الحدين الثاني و الثالث. في مثل هذه الحالات نقوم بتكرار الحد الأول (مستخدمين نظرية المتغير مع نفسه) بحيث يتم جمعه مع كلا الحدين الثاني و الثالث.

$y = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C$	
$y = \overline{A}\overline{B}\overline{C} + \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C} + A\overline{B}C$	بتكرار الحد الأول
$y = \overline{A}\overline{B} + \overline{A}\overline{C} + A\overline{B}$	بجمع كل حدين متشابهين
$y = \overline{A}\overline{B} + A\overline{B} + \overline{A}\overline{C}$	بالنظرية الإبدالية
$y = \overline{B} + \overline{A}\overline{C}$	بجمع الحدين المتشابهين

مثال:

استخدم نظريات الجبر البولي في تبسيط التعبير المنطقي

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

الحل:

نلاحظ أن ما أسفل خط العكس المنطقي الخارجي هو عبارة عن تعبير في صورة مجموع الحدود الصغرى، لذلك نبحث عن التشابهات ما بين الحدود.

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

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

بجمع كل حدين متشابهين

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

بنظرية دي مورغان

$$y = (\overline{B\overline{C}}) \cdot (\overline{A\overline{B}})$$

بنظرية دي مورغان

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

مثال:

استخدم نظريات الجبر البولي في تبسيط التعبير المنطقي

$$y = \overline{\overline{A}BC + AB\overline{C} + ABC}$$

الحل:

نلاحظ أن ما أسفل خط العكس المنطقي الخارجي هو عبارة عن تعبير في صورة مجموع الحدود الصغرى، لذلك نبحث عن التشابهات ما بين الحدود.

$$y = \overline{\overline{A}BC + AB\overline{C} + ABC}$$

$$y = \overline{\overline{A}BC + AB\overline{C} + ABC}$$

بتكرار الحد الثالث
$$y = \overline{\overline{A}BC + ABC + AB\overline{C} + ABC}$$

بجمع كل حدين متشابهين
$$y = \overline{BC + AB}$$

بأخذ العامل المشترك
$$y = \overline{B(C + A)}$$

بنظرية دي مورغان
$$y = \overline{B} + \overline{(C + A)}$$

بنظرية دي مورغان
$$y = \overline{B} + \overline{C}\,\overline{A}$$

تدريب 5

استخدم نظريات الجبر البولي في تبسيط كل من التعبيرات المنطقية التالية

1- $A = x + xyz + \overline{x}yz + xw + x\overline{w} + \overline{x}y$

2- $B = (x + \overline{y} + xy)(x + \overline{y})\overline{x}y$

3- $C = (x + \overline{y} + x\overline{y})(xy + \overline{x}z + yz)$

الخلاصة

عزيزي الدارس، تعلمنا في هذه الوحدة بعض المهارات الأساسية التي سنحتاج إليها في دراستنا لبقية المقرر. حيث تعرفنا على العمليات المنطقية المختلفة وهي العمليات التي يمكن إجراؤها على المتغيرات المنطقية ، والعمليات هنا نوعان: أساسية وهي :NOT و AND وOR .

وعمليات غير أساسية مثل : NAND و NOR وXOR

وفي قسم آخر من الوحدة تناولنا الأساليب المختلفة التي يمكن إتباعها لتغير عدد أطراف الدخل للبوابة المنطقية بالزيادة أو النقصان وهذا ما يعرف بتغير عدد أطراف الدخل (Fan - In) للبوابة المنطقية.

وفي قسم آخر عرفنا أن التغير المنطقي وهو مجموعة من المتغيرات المنطقية المرتبطة مع بعضها البعض بعمليات منطقية ، وهنا يتم إجراء العمليات المنطقية الأساسية الثلاث بالترتيب التالي:

1- عملية العكس المنطقي NOT.
2- عملية AND.
3- عملية OR.

وفي قسم الدائرة المنطقية عرفنا انه يمكن تمثيل أي تعبير منطقي بدائرة منطقية ، حيث ننظر للعمليات المنطقية الموجودة بالتعبير ونقوم بربط البوابات المنطقية التي تقوم بإدارة تلك العمليات بالأسلوب المناسب.

أما المخطط المنطقي فهو عبارة عن مخطط مبسط يوضح متغيرات الدخل للدائرة المنطقية ومسمياتها ومتغيرات الخرج ومسمياتها بالإضافة إلى اسم الدائرة الدال على وظيفتها.

ثم انتقلت بك الوحدة الى جدول الصواب والذي نوضح فيه جميع احتمالات الدخل للدائرة المنطقية وقيم الخرج المقابل لكل منها.

وفي القسم الثامن من الوحدة نظريات الجبر البولي عرفنا كيف نحصل علـــى النظريـــة المقابلة لأي نظرية وذلك بإجراء التبديلات التالية في النظرية الأصلية:

- استبدال أي 0 بـــ 1.
- استبدال أي 1 بـــ 0.
- استبدال أي عملية AND بعملية OR.
- استبدال أي عملية OR بعملية AND.

وفي القسم الأخير من الوحدة استخدام نظريات الجبر البـــولي فـــي تبســـيط التعبيـــرات المنطقية عرفنا أن الهدف من تبسيط التعبير المنطقي هو تبســـيط الـــدائرة المنطقيـــة أي تقليل عدد البوابات المنطقية الداخلة في بنائها وذلك لتقليل تكلفتها.

لمحة مسبقة عن الوحدة التالية

في الوحدة التالية سنتناول الخطوات المتبعة في تصميم الدوائر المنطقيـــة، ابتـــداءاً مـــن تحديد مواصفات الدائرة، ثم كتابة التعبيرات المنطقية للدائرة فـــي الصـــورة المناســـبة، فتبسيط تلك التعبيرات المنطقية، وختاماً بناء الدائرة المنطقية، إمـــا باســـتخدام البوابـــات الأساسية الثلاث NOT و AND و OR، أو باستخدام نوع واحد من البوابـــات (NAND أو NOR).

إجابات التدريبات

تدريب 1

A	B	C	$x = A \cdot B \cdot C$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

تدريب 2

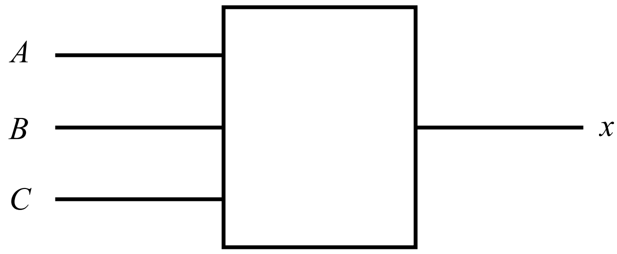
A	B	C	D	$x = A + B + C + D$
0	0	0	0	0
0	0	0	1	1
0	0	1	0	1
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

تدريب 3

A	B	$\overline{A}$	$\overline{B}$	AB	$\overline{A}\,\overline{B}$	$AB+\overline{A}\,\overline{B}$	$\overline{A\oplus B}$
0	0	1	1	0	1	1	1
0	1	1	0	0	0	0	0
1	0	0	1	0	0	0	0
1	1	0	0	1	0	1	1

تدريب 4

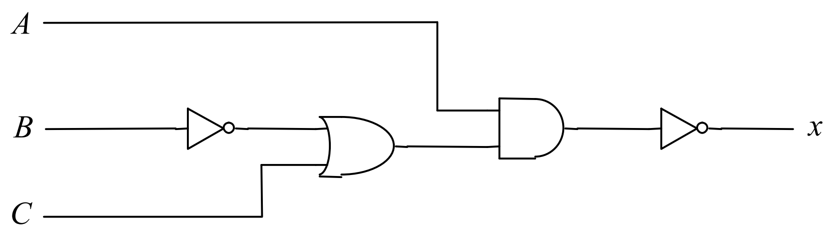
1- المخطط المنطقي



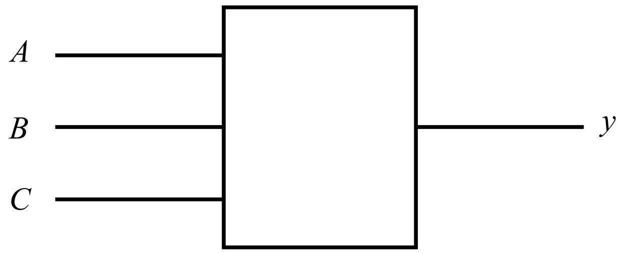
جدول الصواب

A	B	C	$\overline{B}$	$\overline{B}+C$	$A(\overline{B}+C)$	$x=\overline{A(\overline{B}+C)}$
0	0	0	1	1	0	1
0	0	1	1	1	0	1
0	1	0	0	0	0	1
0	1	1	0	1	0	1
1	0	0	1	1	1	0
1	0	1	1	1	1	0
1	1	0	0	0	0	1
1	1	1	0	1	1	0

1- الدائرة المنطقية



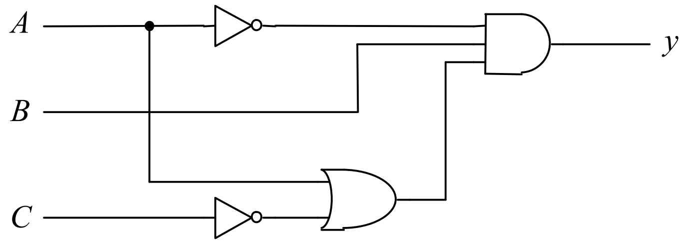
2- المخطط المنطقي



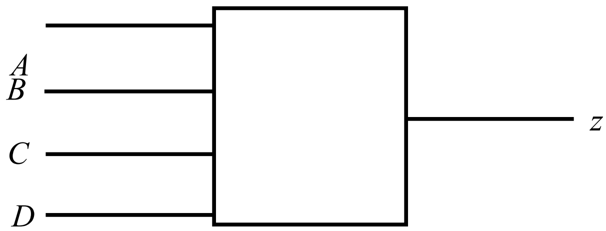
جدول الصواب

A	B	C	$\overline{A}$	$\overline{C}$	$\overline{A}B$	$A+\overline{C}$	$y=\overline{A}B(A+\overline{C})$
0	0	0	1	1	0	1	0
0	0	1	1	0	0	0	0
0	1	0	1	1	1	1	1
0	1	1	1	0	1	0	0
1	0	0	0	1	0	1	0
1	0	1	0	0	0	1	0
1	1	0	0	1	0	1	0
1	1	1	0	0	0	1	0

الدائرة المنطقية



3- المخطط المنطقي



جدول الصواب

A	B	C	D	$\overline{B}$	$\overline{D}$	$A\overline{B}$	$C\overline{D}$	$A\overline{B}+C\overline{D}$	$y=\overline{A\overline{B}+C\overline{D}}$
0	0	0	0	1	1	0	0	0	1
0	0	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0
0	0	1	1	1	0	0	0	0	1
0	1	0	0	0	1	0	0	0	1
0	1	0	1	0	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0
0	1	1	1	0	0	0	0	0	1
1	0	0	0	1	1	1	0	1	0
1	0	0	1	1	0	1	0	1	0
1	0	1	0	1	1	1	1	1	0
1	0	1	1	1	0	1	0	1	0
1	1	0	0	0	1	0	0	0	1
1	1	0	1	0	0	0	0	0	1
1	1	1	0	0	1	0	1	1	0
1	1	1	1	0	0	0	0	0	1

تدريب 5

1- $A = x + y$

2- $B = 0$

3- $C = xy + \bar{x}\bar{y}z$

مسرد المصطلحات

• **المتغير المنطقي (Logical Variable)**

المتغير المنطقي هو أي متغير يمكن أن يأخذ قيمة واحدة فقط من قيمتين.

• **العمليات المنطقية (Logical Operations)**

العمليات المنطقية هي **العمليات التي يمكن إجراؤها على المتغيرات المنطقية.**

• **عملية NOT**

يطلق عليها أيضاً عملية العكس المنطقي (Logical Inversion)، وفيها يكون الخرج عبارة عن معكوس الدخل، فإذا كان الدخل مساوياً 1 فإن الخرج يكون مساوياً 0، و إذا كان الدخل مساوياً 0 فإن الخرج يكون مساوياً 1.

• **عملية AND**

في هذه العملية يكون الخرج مساوياً 1 فقط إذا كانت جميع متغيرات الدخل مساوية 1، و يكون الخرج يكون مساوياً 0 إذا كان أي متغير من متغيرات الدخل مساوياً 0.

• **عملية OR**

في هذه العملية يكون الخرج مساوياً 1 إذا كان أي من متغيرات الدخل مساوياً 1، و يكون الخرج يكون مساوياً 0 إذا كانت جميع متغيرات الدخل مساوية 0.

• **عملية NAND**

عملية NAND هي عبارة عن عملية AND متبوعة بعملية NOT، أي أنها عملية NOT AND .

• **عملية NOR**

عملية NOR هي عبارة عن عملية OR متبوعة بعملية NOT، أي أنها عملية NOT OR.

• **عملية XOR**

XOR هو اختصار لعبارة Exclusive OR، و تسمى عملية الاختلاف، حيث أن الخرج يساوي 1 إذا كان الدخلان مختلفين، و يساوي 0 إذا كانا متشابهين.

- **عملية XNOR**

هي معكوس عملية XOR، و تسمى عملية التساوي، حيث أن الخرج يساوي 1 إذا كان الدخلان متساويين، و يساوي 0 إذا كانا مختلفين.

- **التعبير المنطقي (Logical Expression)**

التعبير المنطقي هو عبارة عن مجموعة من المتغيرات المنطقية المرتبطة مع بعضها البعض بعمليات منطقية.

- **المخطط المنطقي (Logic Diagram)**

هو عبارة عن مخطط مبسط يوضح متغيرات الدخل للدائرة المنطقية و مسمياتها و متغيرات الخرج ومسمياتها، بالإضافة إلى اسم الدائرة الدال على وظيفتها.

- **جدول الصواب (Truth Table)**

عبارة عن جدول يوضح جميع احتمالات الدخل للدائرة المنطقية و قيم الخرج المقابل لكل منها.

المصادر و المراجع

Fredrick J. Hill & Gerald R. Peterson, “Introduction to Switching Theory & Logical Design”, Third Edition, John Wiley & Sons, 1981


الوحدة الثالثة
تصميم الدوائر المنطقية

محتويات الوحدة

المقدمة

تمهيد

مرحباً بك عزيزي الدارس، في الوحدة الثالثة من مقرر "أساسيات التصميم المنطقي". تتناول هذه الوحدة تصميم الدوائر المنطقية (Logic Circuit Design). حيث يتم شرح خطوات التصميم بالتفصيل ابتداءاً من تحديد مواصفات الدائرة، ثم كتابة التعبيرات المنطقية في واحدة من أربع صور مختلفة، فتبسيط تلك التعبيرات إما باستخدام نظريات الجبر البولي أو باستخدام مخططات كارنوف، و اخيراً بناء الدائرة المنطقية التي تم تصميمها، إما باستخدام البوابات الأساسية NOT و AND و OR أو باستخدام نوع واحد من البوابات NAND أو NOR. و نختم الوحدة بعدد من الأمثلة التي نطبق فيها عملية التصميم التي درسناها. و تصميم الدوائر المنطقية تعتبر المهارة الأساسية التي يجب أن يتعلمها دارس هذا المقرر.

أهداف الوحدة

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- **تحدد** مواصفات الدائرة المنطقية.
- **تكتب** التعبيرات المنطقية بالصور المختلفة و باختيار الصورة المناسبة.
- **تبسط** التعبيرات المنطقية و باختيار أسلوب التبسيط المناسب.
- **تبني** الدائرة المنطقية باستخدام البوابات الأساسية الثلاث أو باستخدام نوع واحد من البوابات.

1. تحديد مواصفات الدائرة المنطقية

الخطوة الأولى في تصميم أي دائرة منطقية هي تحديد مواصفات تلك الدائرة بدقـــة. و يتم ذلك بإعطاء:

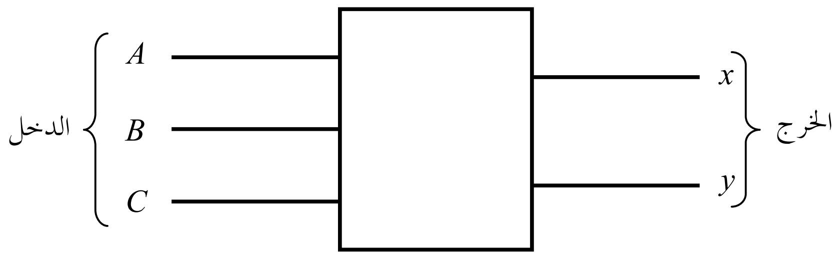
1- مخطط منطقي (Logic Diagram).

2- جدول صواب (Truth Table).

المخطط المنطقي يوضح متغيرات الدخل و متغيرات الخرج للدائرة، و جدول الصـــواب يوضح العلاقة التفصيلية ما بين الدخل و الخرج، أي قيم متغيرات الخرج لكل احتمـــال من احتمالات الدخل، و هذا يتحدد بالطبع من الوظيفة المطلوب أن تؤديهـــا الـــدائرة. و يتم استخراج هذه المعلومات من نص المسألة.

مثال:

صمم الدائرة المنطقية التي يوضحها المخطط المنطقي و جدول الصواب لها أدناه.



المخطط المنطقي

(Logic Diagram)

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

جدول الصواب
(Truth Table)

المخطط المنطقي هنا يوضح أن الدائرة المطلوب تصميمها لها ثلاثة متغيرات دخل هي A و B و C، و متغير خرج هما x و y. و جدول الصواب يحدد قيم متغيري الخرج x و y المطلوبة لكل احتمال من احتمالات الدخل.

2. كتابة التعبيرات المنطقية

في هذه الخطوة يتم كتابة تعبير منطقي لكل متغير من متغيرات الخرج، بحيث يعطي التعبير نفس قيم الخرج المطلوبة و الموضحة في جدول الصواب. و يتم كتابة هذه التعبيرات المنطقية من جدول الصواب.

توجد أربعة صور مختلفة للتعبير المنطقي هي:

1. صورة مجموع الحدود الصغرى (Sum of minterms).
2. صوره مضروب الحدود الكبرى (Product of Maxterms).
3. صورة AND-OR-Invert.
4. صورة OR-AND-Invert.

1.2 صورة مجموع الحدود الصغرى (Sum of minterms)

الحد الأصغر (minterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات AND، و قد يظهر متغير معين في الحد الأصغر معكوساً أو بدون عكس. يوجد عدد من الحدود الصغرى يساوي عدد احتمالات الدخل، أي عدد أسطر جدول الصواب، بحيث يكون هناك حد أصغر مقابل كل سطر من أسطر جدول الصواب. لإيجاد الحد الأصغر المقابل لسطر معين من أسطر جدول الصواب ننظر إلى قيم متغيرات الدخل في ذلك السطر، فإذا كانت قيمة المتغير هي 0 يظهر ذلك المتغير في الحد الأصغر معكوساً، أما إذا كانت قيمة المتغير هي 1 يظهر ذلك المتغير في الحد الأصغر بدون عكس. كما هو موضح في الجدول التالي، الذي يوضح جميع الحدود الصغرى لثلاثة متغيرات دخل A و B و C.

A	B	C	Minterm
0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$
0	0	1	$\overline{A}\,\overline{B}C$
0	1	0	$\overline{A}B\overline{C}$
0	1	1	$\overline{A}BC$
1	0	0	$A\overline{B}\,\overline{C}$
1	0	1	$A\overline{B}C$
1	1	0	$AB\overline{C}$
1	1	1	ABC

لاحظ أن أي حد أصغر يساوي 1 فقط لاحتمال الدخل المقابل له في جدول الصواب، و يساوي 0 خلاف ذلك، أي يساوي 0 لجميع احتمالات الدخل الأخرى في جدول الصواب.

لكتابة التعبير المنطقي لمتغير معين من متغيرات الخرج في صورة مجموع الحدود الصغرى ننظر إلى قيم ذلك المتغير في جدول الصواب و نبحث عن الـــ 1's، ثم نقوم بأخذ الحدود الصغرى المقابلة لهذه الـــ 1's و نقوم بجمعها، أي نربط بينها بعمليات OR.

<u>**مثال:**</u>

اكتب التعبيرين المنطقيين لمتغيري الخرج x و y في جدول الصواب التالي في صورة مجموع الحدود الصغرى.

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

<u>**الحل:**</u>

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

$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

و يمكن بسهولة التحقق من أن التعبير المنطقي المكتوب في صورة مجموع الحدود الصغرى يعطي فعلاً قيم الخرج المطلوبة الموضحة في جدول الصواب عند تعويض احتمالات الدخل فيه. فحيث إننا قد قمنا باختيار الحدود الصغرى المقابلة للـــ 1's فقط في جدول الصواب و ربطناها بعمليات OR، و حيث إن الحد الأصغر يساوي 1 فقط لاحتمال الدخل المقابل له في جدول الصواب و يساوي 0 خلاف ذلك، فإنه عند تعويض أي احتمال من احتمالات الدخل المقابلة للـــ 1's في التعبير المنطقي فإن الحد الأصغر المقابل لذلك الاحتمال سيساوي 1، و بالتالي فإن التعبير بأكمله سيساوي 1 أيضاً. أي أن كل حد من الحدود الصغرى المضمنة في التعبير المنطقي يضمن لنا ظهور الـــ 1 المقابل له في جدول الصواب. أما عند تعويض أحد احتمالات الدخل المقابلة للـــ 0's في التعبير المنطقي فإن أياً من الحدود الصغرى المضمنة في التعبير المنطقي لن يكون مساوياً 1 بل ستكون جميعاً مساوية للـــ 0 و بالتالي فإن التعبير المنطقي بأكمله سيساوي 0.

لتسهيل كتابة التعبيرات المنطقية في صورة مجموع الحدود الصغرى يتم ترقيم أسطر جدول الصواب (ابتداء بالقيمة 0)، و استخدام الرمز m_k للحد الأصغر المقابل للسطر k من جدول الصواب. كما هو موضح بالجدول التالي:

#	A	B	C	Minterm	
0	0	0	0	$\overline{A}\,\overline{B}\,\overline{C}$	m_0
1	0	0	1	$\overline{A}\,\overline{B}C$	m_1
2	0	1	0	$\overline{A}B\overline{C}$	m_2
3	0	1	1	$\overline{A}BC$	m_3
4	1	0	0	$A\overline{B}\,\overline{C}$	m_4
5	1	0	1	$A\overline{B}C$	m_5
6	1	1	0	$AB\overline{C}$	m_6
7	1	1	1	ABC	m_7

و عليه يمكن كتابة التعبيرين المنطقيين لمتغيري الخرج x و y في المثال السابق بصورة مختصرة، باستخدام رموز الحدود الصغرى، كالتالي:

$$x = m_0 + m_1 + m_3 + m_7$$
$$y = m_0 + m_1 + m_2 + m_4 + m_5$$

كما يمكن تسهيل كتابة التعبير أكثر من ذلك باستخدام رمز المجموع $\sum$، و ذلك كالتالي

$$x = \sum m\ (0,1,3,7)$$
$$y = \sum m\ (0,1,2,4,5)$$

أي أن التعبير المنطقي في صورة مجموع الحدود الصغرى يمكن أن يكتب كاملاً، أو مختصراً باستخدام رموز الحدود الصغرى، أو مختصراً باستخدام رمز المجموع $\sum$

و أرقام الحدود الصغرى (أي أرقام أسطر جدول الصواب التي تحتوي على القيمة 1 للمتغير الذي يتم كتابة التعبير له). و بناء عليه فإنه في المثال السابق

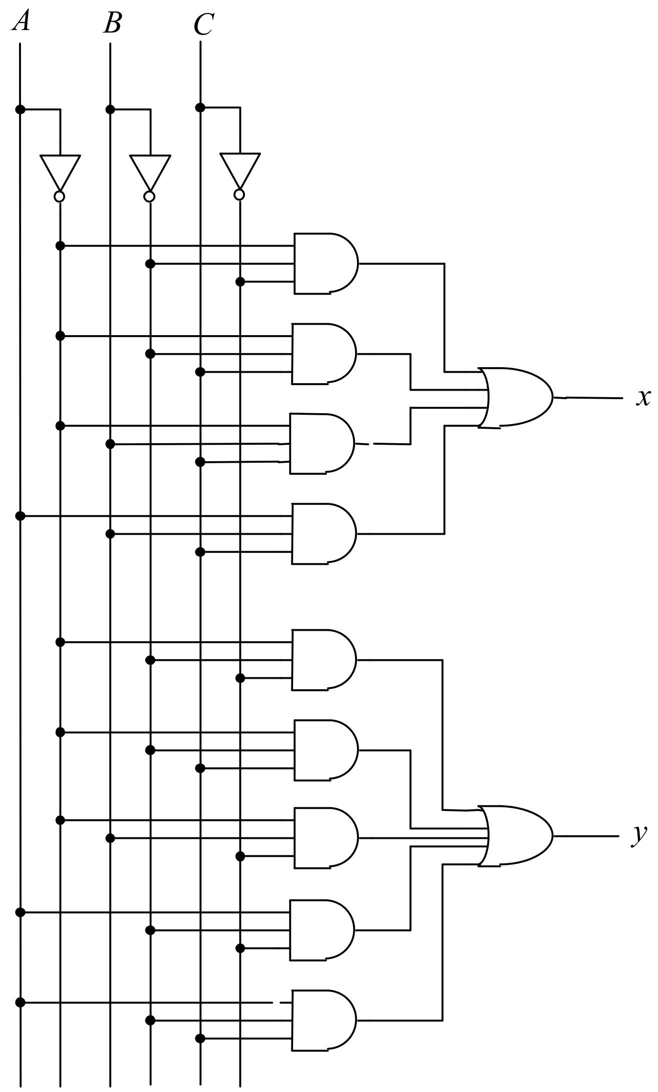
$$
\begin{aligned}
x &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + ABC \\
&= m_0 + m_1 + m_3 + m_7 \\
&= \sum m\ (0,1,3,7)
\end{aligned}
$$

$$
\begin{aligned}
y &= \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C \\
&= m_0 + m_1 + m_2 + m_4 + m_5 \\
&= \sum m\ (0,1,2,4,5)
\end{aligned}
$$

لاحظ هنا أهمية مراعاة استخدام الحرف m الصغير (small letter) للرمز للحدود الصغرى (minterms)، حتى لا يحدث خلط بينها و بين الحدود الكبرى (Maxterms)، حيث إننا سنستخدم الحرف M الكبير (capital letter) للرمز للحدود الكبرى.

2.2 الدائرة المنطقية للتعبير في صورة مجموع الحدود الصغرى

إذا قمنا برسم الدائرة المنطقية للتعبيرين أعلاه المكتوبين في صورة مجموع الحدود الصغرى نحصل على هذا الشكل:



لاحظ أن الدائرة تتكون من مجموعة من بوابات AND، مهمة كل منها توليد أحد الحدود الصغرى، تليها مجموعة من بوابات OR مهمة كل منها جمع الحدود الصغرى لتوليد أحد متغيرات الخرج. و يطلق على هذا النوع من الدوائر تسمية AND-OR Structure.

صورة مضروب الحدود الكبرى (Product of Maxterms)

الحد الأكبر (Maxterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات OR، و قد يظهر متغير معين في الحد الأكبر معكوساً أو بدون عكس. يوجد عدد من الحدود الكبرى يساوي عدد احتمالات الدخل، أي عدد أسطر جدول الصواب، بحيث يكون هناك حد أكبر مقابل كل سطر من أسطر جدول الصواب.

لإيجاد الحد الأكبر المقابل لسطر معين من أسطر جدول الصواب ننظر إلى قيم متغيرات الدخل في ذلك السطر، فإذا كانت قيمة المتغير هي 0 يظهر ذلك المتغير في الحد الأكبر بدون عكس، أما إذا كانت قيمة المتغير هي 1 يظهر ذلك المتغير في الحد الأكبر معكوساً. كما هو موضح في الجدول التالي، الذي يوضح جميع الحدود الكبرى لثلاثة متغيرات دخل A و B و C.

A	B	C	Maxterm
0	0	0	$A+B+C$
0	0	1	$A+B+\overline{C}$
0	1	0	$A+\overline{B}+C$
0	1	1	$A+\overline{B}+\overline{C}$
1	0	0	$\overline{A}+B+C$
1	0	1	$\overline{A}+B+\overline{C}$
1	1	0	$\overline{A}+\overline{B}+C$
1	1	1	$\overline{A}+\overline{B}+\overline{C}$

لاحظ أن أي حد أكبر يساوي 0 فقط لاحتمال الدخل المقابل له في جدول الصواب، و يساوي 1 خلاف ذلك، أي يساوي 1 لجميع احتمالات الدخل الأخرى في جدول الصواب.

لكتابة التعبير المنطقي لمتغير معين من متغيرات الخرج في صورة مضروب الحدود الكبرى ننظر إلى قيم ذلك المتغير في جدول الصواب و نبحث عن الـــ 0's، ثم نقوم بأخذ الحدود الكبرى المقابلة لهذه الـــ 0's و نقوم بضربها، أي نربط بينها بعمليات AND.

<ins>مثال:</ins>

اكتب التعبيرين المنطقيين لمتغيري الخرج x و y في جدول الصواب التالي في صورة مضروب الحدود الكبرى.

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

<ins>الحل:</ins>

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

و يمكن بسهولة التحقق من أن التعبير المنطقي المكتوب في صورة مضروب الحدود الكبرى يعطي فعلاً قيم الخرج المطلوبة الموضحة في جدول الصواب عند تعويض احتمالات الدخل فيه. فحيث أننا قد قمنا باختيار الحدود الكبرى المقابلة للـــ 0's فقط في جدول الصواب و ربطناها بعمليات AND، و حيث أن الحد الأكبر يساوي 0 فقط لاحتمال الدخل المقابل له في جدول الصواب و يساوي 1 خلاف ذلك، فإنه عند تعويض أي احتمال من احتمالات الدخل المقابلة للـــ 0's في التعبير المنطقي فإن الحد الأكبر المقابل لذلك الاحتمال سيساوي 0، و بالتالي فإن التعبير بأكمله سيساوي 0 أيضاً. أي أن كل حد من الحدود الكبرى المضمنة في التعبير المنطقي يضمن لنا ظهور الـــ 0 المقابل له في جدول الصواب. أما عند تعويض أحد احتمالات الدخل المقابلة للـــ 1's في التعبير المنطقي فإن أياً من الحدود الكبرى المضمنة في التعبير المنطقي لن يكون مساوياً 0 بل ستكون جميعاً مساوية للـــ 1 و بالتالي فإن التعبير المنطقي بأكمله سيساوي 1.

لتسهيل كتابة التعبيرات المنطقية في صورة مضروب الحدود الكبرى يتم ترقيم أسطر جدول الصواب (ابتداءاً بالقيمة 0)، و استخدام الرمز M_k للحد الأكبر المقابل للسطر k من جدول الصواب. كما هو موضح بالجدول التالي

#	A	B	C	Maxterm	
0	0	0	0	$A+B+C$	M_0
1	0	0	1	$A+B+\overline{C}$	M_1
2	0	1	0	$A+\overline{B}+C$	M_2
3	0	1	1	$A+\overline{B}+\overline{C}$	M_3
4	1	0	0	$\overline{A}+B+C$	M_4
5	1	0	1	$\overline{A}+B+\overline{C}$	M_5
6	1	1	0	$\overline{A}+\overline{B}+C$	M_6
7	1	1	1	$\overline{A}+\overline{B}+\overline{C}$	M_7

و عليه يمكن كتابة التعبيرين المنطقيين لمتغيري الخرج x و y في المثال السابق بصورة مختصرة، باستخدام رموز الحدود الكبرى، كالتالي

$$x = M_2 \cdot M_4 \cdot M_5 \cdot M_6$$
$$y = M_3 \cdot M_6 \cdot M_7$$

كما يمكن تسهيل كتابة التعبير أكثر من ذلك باستخدام رمز المضروب $\prod$، و ذلك كالتالي

$$x = \prod M(2,4,5,6)$$
$$y = \prod M(3,6,7)$$

أي أن التعبير المنطقي في صورة مضروب الحدود الكبرى يمكن أن يكتب كاملاً، أو مختصراً باستخدام رموز الحدود الكبرى، أو مختصراً باستخدام رمز المضروب $\prod$

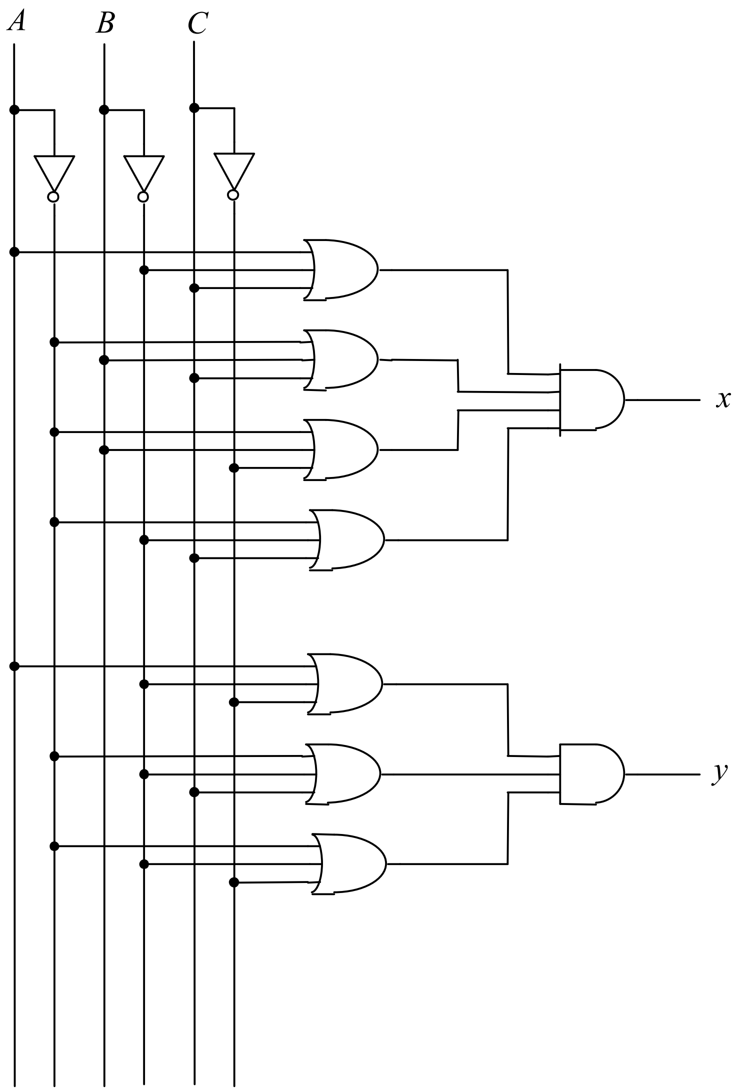
و أرقام الحدود الكبرى (أي أرقام أسطر جدول الصواب الحاوية على القيمة 0 للمتغير الذي يتم كتابة التعبير له). و بناء عليه فإنه في المثال السابق

$$\begin{aligned} x &= (A+\overline{B}+C)(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+C) \\ &= M_2 \cdot M_4 \cdot M_5 \cdot M \\ &= \prod M(2,4,5,6) \end{aligned}$$

$$\begin{aligned} y &= (A+\overline{B}+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C}) \\ &= M_3 \cdot M_6 \cdot M_7 \\ &= \prod M(3,6,7) \end{aligned}$$

الدائرة المنطقية للتعبير في صورة مضروب الحدود الكبرى

إذا قمنا برسم الدائرة المنطقية للتعبيرين أعلاه المكتوبين في صورة مضروب الحدود الكبرى نحصل على الشكل التالي:



لاحظ أن الدائرة تتكون من مجموعة من بوابات OR، مهمة كل منها توليد أحد الحدود الكبرى، تليها مجموعة من بوابات AND مهمة كل منها ضرب الحدود الكبرى لتوليد أحد متغيرات الخرج. و يطلق على هذا الشكل من الدوائر تسمية OR-AND Structure.

3.2 صورة AND-OR-Invert

هذه الصورة تشبه إلى حد كبير صورة مجموع الحدود الصــغرى (Sum of minterms) إلا أن التعبير المنطقي بأكمله يكون معكوساً. فلكتابة التعبير المنطقي في هذه الصــورة لمتغير معين من متغيرات الخرج ننظر إلى قيم ذلك المتغير في جدول الصواب و نبحث عن الــ 0's، ثم نقوم بأخذ الحدود الصغرى المقابلة لهذه الــ 0's و نقــوم بجمعهــا، أي نربط بينها بعمليات OR، و أخيراً نقوم بعكس التعبير المنطقي الناتج بأكملــه. أي أننــا نقوم بكتابة التعبير المنطقي في صورة مجموع الحدود الصغرى لمعكوس متغير الخرج، ثم نقوم بعكس التعبير المنطقي الناتج بأكمله.

<u>مثال:</u>

اكتب التعبيرين المنطقيين لمتغيري الخرج x و y في جدول الصواب التالي في صورة AND-OR-Invert.

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

الحل:

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

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

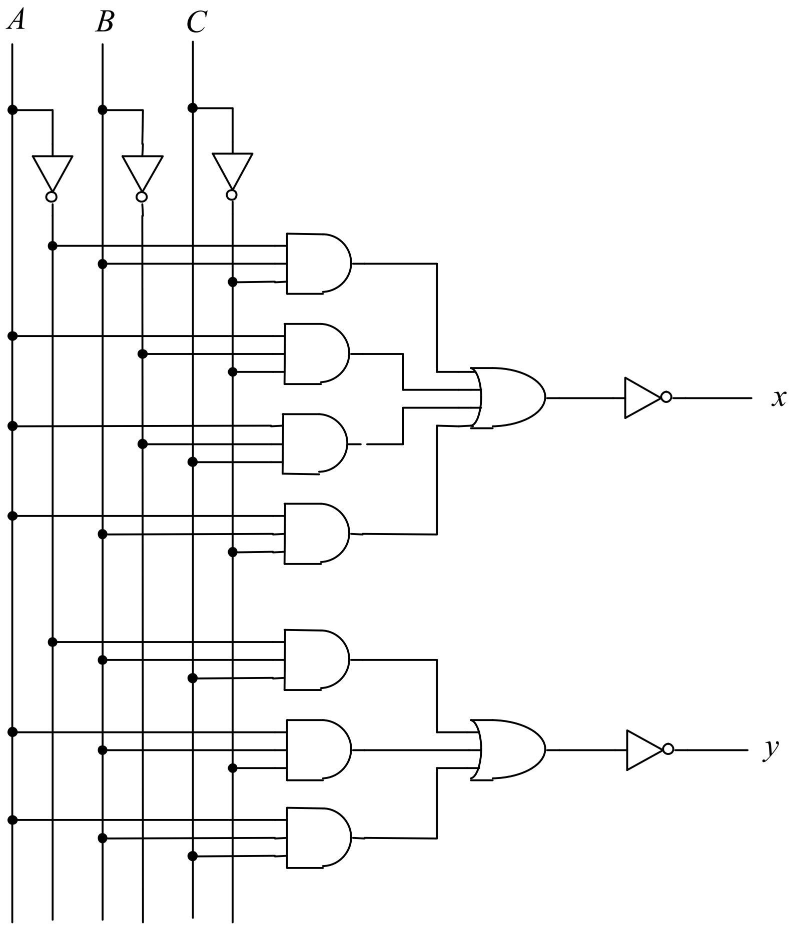
$$\overline{y} = \overline{A}BC + AB\overline{C} + ABC$$

$$y = \overline{\overline{A}BC + AB\overline{C} + ABC}$$

لاحظ أنه إذا قمنا بتطبيق نظرية دي مورغان (DeMorgan's Theorem) على التعبيرات المنطقية المكتوبة في صورة AND-OR-Invert فإننا نحصل على نفس التعبيرات المنطقية التي قمنا بكتابتها من قبل في صورة مضروب الحدود الكبرى (Product of Maxterms).

الدائرة المنطقية للتعبير في صورة AND-OR-Invert

إذا قمنا برسم الدائرة المنطقية للتعبيرين أعلاه المكتوبين في صورة AND-OR-Invert نحصل على



لاحظ أن الدائرة تتكون من مجموعة من بوابات AND، مهمة كل منها توليد أحد الحدود الصغرى، تليها مجموعة من بوابات OR مهمة كل منها جمع الحدود الصغرى لتوليد معكوس أحد متغيرات الخرج، ثم مجموعة من العواكس المنطقية مهمة كل منها عكس خرج إحدى بوابات OR لتوليد أحد متغيرات الخرج. و يطلق على هذا الشكل من الدوائر تسمية AND-OR-Invert Structure.

4.2 صورة OR-AND-Invert

هذه الصورة تشبه إلى حد كبير صورة مضروب الحدود الكبرى (Product of Maxterms) إلا أن التعبير المنطقي بأكمله يكون معكوساً. فلكتابة التعبير المنطقي في هذه الصورة لمتغير معين من متغيرات الخرج ننظر إلى قيم ذلك المتغير في جدول الصواب و نبحث عن الـ 1's، ثم نقوم بأخذ الحدود الكبرى المقابلة لهذه الـ 1's و نقوم بضربها، أي نربط بينها بعمليات AND، و أخيراً نقوم بعكس التعبير المنطقي الناتج بأكمله. أي أننا نقوم بكتابة التعبير المنطقي في صورة مضروب الحدود الكبرى لمعكوس متغير الخرج، ثم نقوم بعكس التعبير المنطقي الناتج بأكمله.

مثال:

اكتب التعبيرين المنطقيين لمتغيري الخرج x و y في جدول الصواب التالي في صورة OR-AND-Invert.

A	B	C	x	y
0	0	0	1	1
0	0	1	1	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	0

الحل:

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

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

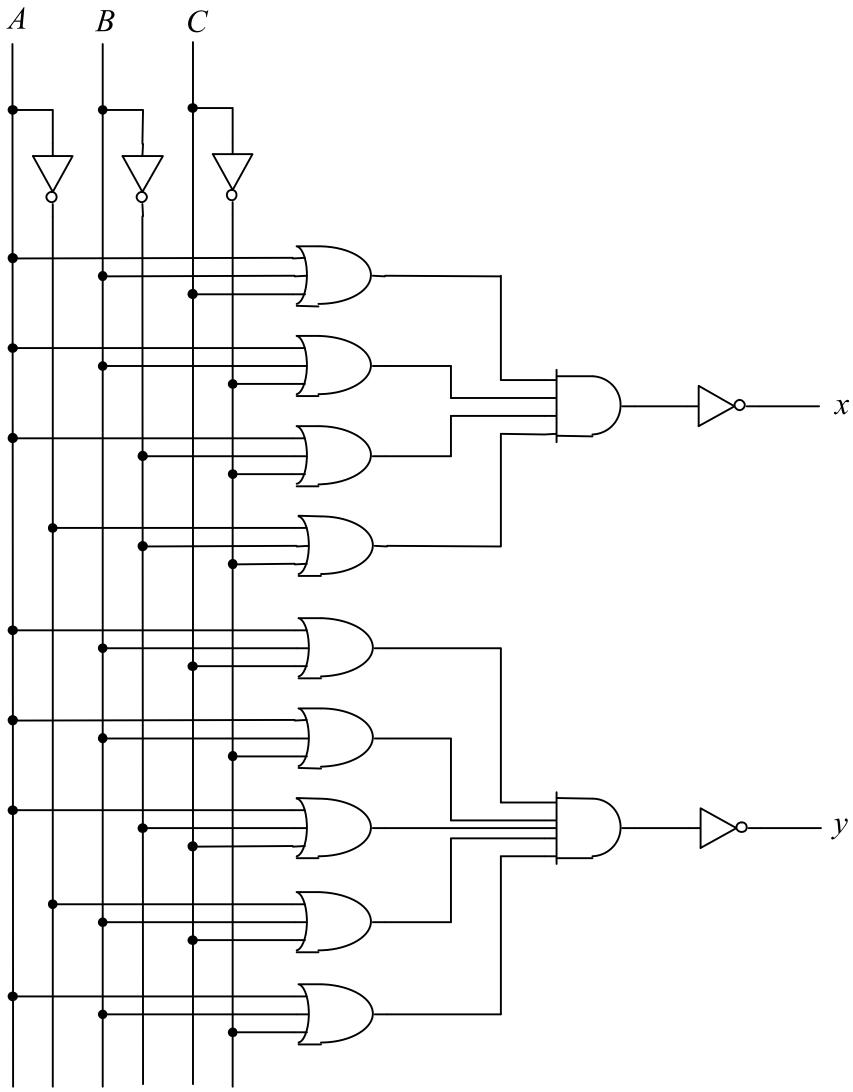
$$\overline{y} = (A+B+C)\cdot(A+B+\overline{C})\cdot(A+\overline{B}+C)\cdot(\overline{A}+B+C)\cdot(\overline{A}+B+\overline{C})$$

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

لاحظ أنه إذا قمنا بتطبيق نظرية دي مورغان (DeMorgan's Theorem) على التعبيرات المنطقية المكتوبة في صورة OR-AND-Invert فإننا نحصل على نفس التعبيرات المنطقية التي قمنا بكتابتها من قبل في صورة مجموع الحدود الصغرى (Sum of minterms).

الدائرة المنطقية للتعبير في صورة OR-AND-Invert

إذا قمنا برسم الدائرة المنطقية للتعبيرين أعلاه المكتوبين في صورة OR-AND-Invert نحصل على



لاحظ أن الدائرة تتكون من مجموعة من بوابات OR، مهمة كل منها توليد أحد الحدود الكبرى، تليها مجموعة من بوابات AND مهمة كل منها جمع الحدود الكبرى لتوليد معكوس أحد متغيرات الخرج، ثم مجموعة من العواكس المنطقية مهمة كل منها عكس خرج إحدى بوابات AND لتوليد أحد متغيرات الخرج. و يطلق على هذا الشكل من الدوائر تسمية OR-AND-Invert Structure.

اختيار الصورة المناسبة للتعبيرات المنطقية

نختار الصورة المناسبة للتعبيرات المنطقية من الصور الأربعة التي درسناها بناء على شكل الدائرة المطلوب. فإذا كنا نريد دائرة في شكل AND-OR Structure نختار صورة مجموع الحدود الصغرى (Sum of minterms)، أما إذا أردنا دائرة في شكل OR-AND Structure فإننا نختار صورة مضروب الحدود الكبرى (Product of Maxterms). و نفس الشيء ينطبق على صورتي AND-OR-Invert و OR-AND-Invert.

تدريب 1

من جدول الصواب التالي اكتب التعبيرين المنطقيين لمتغيري الخرج x و y في صورة:

1- مجموع الحدود الصغرى.

2- مضروب الحدود الكبرى.

3- AND-OR-Invert.

4- OR-AND-Invert.

A	B	x	y
0	0	1	0
0	1	0	1
1	0	1	1
1	1	1	

تدريب 2

من جدول الصواب التالي اكتب التعبيرات المنطقية لمتغيرات الخرج x و y و z في صورة:

1 – مجموع الحدود الصغرى.

2– مضروب الحدود الكبرى.

3– AND-OR-Invert.

4– OR-AND-Invert.

A	B	C	x	y	z
0	0	0	1	1	1
0	0	1	1	0	0
0	1	0	1	1	1
0	1	1	0	1	1
1	0	0	1	1	0
1	0	1	1	0	0
1	1	0	0	1	1
1	1	1	0	1	0

3. تبسيط التعبيرات المنطقية

الهدف من عملية تبسيط التعبيرات المنطقية هو وضع تلك التعبيرات في أبسط صورة ممكنة، و ذلك لتبسيط الدائرة المنطقية، أي تقليل عدد البوابات المنطقية المستخدمة في بنائها، و بالتالي تقليل تكلفتها. و يتم تبسيط التعبيرات المنطقية بإحدى طريقتين:

1. باستخدام نظريات الجبر البولي (Boolean Algebra Theorems)
2. باستخدام مخططات كارنوف (Karnaugh Maps)

1.3 التبسيط باستخدام نظريات الجبر البولي

يتم التبسيط هنا بالبحث عن التشابهات ما بين الحدود. و الحدان المتشابهان هما حدان يتشابهان في كل شيء عدا متغير واحد يظهر في أحدهما معكوساً و في الآخر بدون عكس. و يتم جمع كل حدين متشابهين في حد واحد هو عبارة عن العامل المشترك ما بين الحدين، أما المتغير المختلف فيتم اختصاره. و في حالة وجود حد معين يتشابه مع أكثر من حد آخر فإنه يمكن تكرار ذلك الحد حسب الحاجة. و يمكن استخدام هذا الأسلوب في التبسيط للتعبيرات المنطقية في أي صورة من الصور الأربعة التي درسناها.

<u>مثال:</u> صورة مجموع الحدود الصغرى

استخدم نظريات الجبر البولي في تبسيط التعبيرين المنطقيين التاليين المكتوبين في صورة مجموع الحدود الصغرى

$$x = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}BC + ABC$$
$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

<u>الحل:</u>

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

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

$$x = \overline{A}\,\overline{B} + BC$$

$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$

$$y = \overline{A}\,\overline{B} + \overline{A}\,\overline{C} + A\overline{B}$$

$$y = \overline{B} + \overline{A}\,\overline{C}$$

ملاحظة

يمكنك عزيزي الدارس الرجوع إلى الوحدة الثانية "ساسيات الجبر البولي" لتجد المثال السابق محلولاً بتفصيل أكبر.

مثال: صورة مضروب الحدود الكبرى

استخدم نظريات الجبر البولي في تبسيط التعبيرين المنطقيين التاليين المكتوبين في صورة مضروب الحدود الكبرى

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

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

الحل:

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

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

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

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

ملاحظة

لاحظ أنه في التعبير المنطقي المختصر للمتغير y يمكن التبسيط أكثر من ذلك بإخراج $\overline{B}$ كعامل مشترك كالتالي:

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

و لكن يجب عدم القيام بذلك لأنه يؤدي إلى تغيير صورة التعبير المنطقي، حيث إن التعبير قبل القيام بهذه الخطوة كان في صورة مضروب الحدود الكبرى و أصبح بعدها في صورة مجموع الحدود الصغرى.

<u>مثال:</u> صورة AND-OR-Invert

استخدم نظريات الجبر البولي في تبسيط التعبيرين المنطقيين التاليين المكتوبين في صورة AND-OR-Invert

$$x = \overline{\bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}}$$
$$y = \overline{\bar{A}BC + AB\bar{C} + ABC}$$

الحل:

$$x = \overline{\bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}}$$

$$x = \overline{\bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C}}$$
$$x = \overline{B\bar{C} + A\bar{B}}$$

$$y = \overline{\bar{A}BC + AB\bar{C} + ABC}$$

$$y = \overline{\bar{A}BC + AB\bar{C} + ABC}$$
$$y = \overline{BC + AB}$$

ملاحظة

لاحظ أنه هنا أيضاً يمكننا تبسيط التعبير المختصر للمتغير y أكثر من ذلك بإخراج المتغير B كعامل مشترك، كما يمكننا تطبيق نظرية دي مورغان (DeMorgan's Theorem) على التعبير المختصر لكل من المتغيرين x و y. و لكن يجب تجنب القيام بأي من ذلك لأنه يؤدي إلى تغيير صورة التعبير المنطقي.

<u>مثال:</u> صورة OR-AND-Invert

استخدم نظريات الجبر البولي في تبسيط التعبيرين المنطقيين التاليين المكتوبين في صورة OR-AND-Invert

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

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

<u>**الحل:**</u>

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

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

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

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

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

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

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

ملاحظة

نؤكد هنا مرة أخرى على ضـــرورة تجنـــب تطبيـــق نظريـــة دي مورغان (DeMorgan's Theorem) على التعبيـــرين المختصـــرين للمتغيرين x و y لأن ذلك يؤدي إلـــى تغييـــر صـــورة التعبيـــر المنطقي.

لاحظ أن هذا الأسلوب في تبسيط التعبيرات المنطقية، و القائم على إيجاد التشابهات ما بين الحدود، يصبح أكثر صعوبة بزيادة عدد الحدود أو زيادة عدد المتغيرات في التعبير المنطقي، حيث يصبح من الصعب اكتشاف جميع التشابهات ما بين الحدود. لذلك تم تطوير أسلوب التبسيط هذا بحيث يصلح لمثل هذه الحالات الصعبة، و تم تسمية الأسلوب المطور بطريقة الجدولة (Tabular Method)، مثل طريقة كوين–مكلسكي (Quine-McCluskey Method). و في طريقة الجدولة هذه يتم البحث عن التشابهات ما بين الحدود، و جمع الحدود المتشابهة بطريقة منظمة و بخطوات محددة، يمكن بسهولة كتابتها في شكل خوارزمية (Algorithm) و برمجتها في الحاسوب. و لن نقوم بتغطية طرق الجدولة في هذا المقرر.

تدريب 3

استخدم نظريات الجبر البولي في تبسيط التعبيرات المنطقية التي قمت بكتابتها في تدريب 1.

تدريب 4

استخدم نظريات الجبر البولي في تبسيط التعبيرات المنطقية التي قمت بكتابتها في تدريب 2.

2.3 التبسيط باستخدام مخططات كـــارنوف (Karnaugh Maps)

مخطط كارنوف (Karnaugh Map)، أو K-Map اختصاراً، هو عبارة عن طريقة أخرى للتعبير عن المعلومات الموجودة في جدول الصــواب (Truth Table)، و الهــدف مــن استخدام المخطط هو تسهيل عملية اكتشاف التشابهات ما بين الحدود و جمــع الحــدود المتشابهة.

مخطط كارنوف لمتغيرين:

سنقوم بتحويل جدول الصواب التالي، و الذي يحتوي على متغيري دخل هما A و B، إلى مخطط كارنوف في متغيرين.

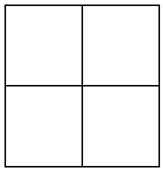
#	A	B	x
0	0	0	1
1	0	1	0
2	1	0	0
3	1	1	1

جدول الصواب

B \ A	**0**	**1**	
0	1 (0)	0 (2)	x
	0 (1)	1 (3)	

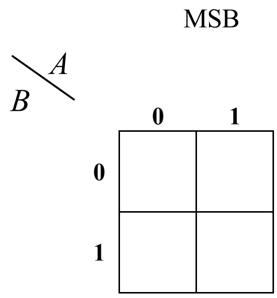
مخطط كارنوف

يتكون مخطط كارنوف من عدد من الخلايا (Cells) مرتبة في شكل صفوف و أعمدة، و عدد هذه الخلايا يساوي عدد أسطر جدول الصواب، حيث إن كل خلية منها تقابل سطراً من أسطر جدول الصواب. فمخطط كارنوف لمتغيرين، مثلاً، يتكون من 4 خلايا مرتبة في شكل صفين و عمودين.

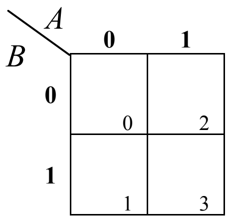


بما أن مخطط كارنوف ما هو إلا طريقة أخرى للتعبير عن المعلومات الموجودة في جدول الصواب، لذلك يجب أن تظهر فيه كل تلك المعلومات، و المتمثلة في متغيرات الدخل و قيمها، و متغير الخرج و قيمه، إضافة إلى أرقام الأسطر التي نستخدمها في ترقيم الحدود الصغرى (minterms) أو الحدود الكبرى (Maxterms).

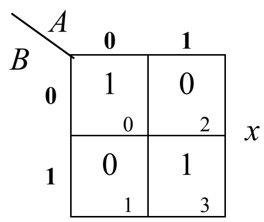
نبدأ بمتغيرات الدخل فنضع قيمها على الصفوف و الأعمدة مبتدئين **بالمتغير الواقع في الخانة العليا (MSB)**، أي المتغير A، فنضع القيم المحتملة له على الأعمدة. و المتغير A كما نعلم يمكن أن يأخذ واحدة من قيمتين، إما 0 و إما 1، فنضع القيمة 0 على العمود الأول (من اليسار) و القيمة 1 على العمود الثاني. أي أن كل خلية من خلايا العمود الأول تكون قيمة المتغير A فيها مساوية 0، و كل خلية من خلايا العمود الثاني تكون قيمة المتغير A فيها مساوية 1. أما المتغير الثاني B فنضع القيم المحتملة له على الصفوف، حيث نضع القيمة 0 على الصف الأول (من أعلى) و القيمة 1 على الصف الثاني. أي أن كل خلية من خلايا الصف الأول تكون قيمة المتغير B فيها مساوية 0، و كل خلية من خلايا الصف الثاني تكون قيمة المتغير B فيها مساوية 1.



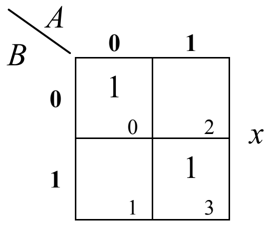
بعد ذلك نقوم بترقيم الخلايا، حيث نقوم بكتابة رقم الخلية بخط صغير في الزاوية السفلية اليمنى من الخلية. و نستخدم الخط الصغير في ترقيم الخلايا حتى لا يحدث خلط بـين رقم الخلية و محتوياتها، حيث تُكتب محتويات الخلايا بالخط العادي. نبدأ ترقيم الخلايـا بالعمود الأول من أعلى إلى أسفل، حيث تأخذ الخلية الواقعة في الصف الأول و العمـود الأول الرقم 0، و تأخذ الخلية الواقعة أسفلها الرقم 1. ثم نواصل التـرقيم فـي العمـود الثاني من أعلى إلى أسفل أيضاً، حيث تأخذ الخلية الواقعة في الصـف الأول و العمـود الثاني الرقم 2، و تأخذ الخلية الواقعة أسفلها الرقم 3. لاحظ أن كل خلية تقابل السـطر الذي يحمل رقمها في جدول الصواب، فالخلية رقم 0 تقابل السطر رقم 0 لأن $A = 0$ و $B = 0$ في كليهما، و الخلية رقم 1 تقابل السطر رقم 1 لأن $A = 0$ و $B = 1$ في كليهما، ... و هكذا. لاحظ أيضاً أن الأسلوب المتبع في ترقيم الخلايا يعتمد على طريقة وضـع متغيرات الدخل على الصفوف و الأعمدة، حيث يجب أن نبدأ دائماً بالمتغير الواقع فـي الخانة العليا (MSB) و نضعه على الأعمدة.



و أخيراً نقوم بوضع متغير الخرج x و قيمه، حيث يوضع اسم المتغير على يمين المخطط، و توضع قيمه في داخل الخلايا.



سنبدأ باستخدام مخططات كارنوف في تبسيط التعبيرات المنطقية المكتوبة في صورة مجموع الحدود الصغرى، و في مثل هذه الحالات عادة ما تظهر الـ 1's فقط في مخطط كارنوف و تترك الخلايا التي تحتوي على 0's فارغة، لأننا عند كتابة التعبير المنطقي نأخذ الحدود الصغرى المقابلة للـ 1's فقط في جدول الصواب.



و كنوع من التسهيل سنقوم بكتابة التعبيرات المنطقية في صورة تتضمن كل المعلومات الموجودة في جدول الصواب، بحيث يمكن رسم مخطط كارنوف من التعبير المنطقي مباشرة دون الحاجة للرجوع إلى جدول الصواب. و يتم ذلك كالتالي

$$x = f(A,B) = \sum m\ (0,3)$$

و الجزء $x = f(A,B)$ يعني أن متغير الخرج x هو عبارة عن دالة (function) في متغيري الدخل A و B، و توضع متغيرات الدخل ما بين قوسي الدالة بنفس ترتيب ظهورها في جدول الصواب، بحيث يكون المتغير الأول (من اليسار) هو المتغير الواقع في الخانة العليا (MSB). أما الجزء $\sum m\ (0,3)$ فيدل على أن الخانات الحاوية على القيمة 1 في مخطط كارنوف هي الخانات 0 و 3.

<u>مخطط كارنوف لثلاثة متغيرات:</u>

سنقوم بتحويل جدول الصواب التالي إلى مخطط كارنوف في ثلاثة متغيرات.

#	A	B	C	x	y
0	0	0	0	1	1
1	0	0	1	1	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	0	1
6	1	1	0	0	0
7	1	1	1	1	0

التعبيرات المنطقية في صورة مجموع الحدود الصغرى

$$x = f(A,B,C) = \sum m\ (0,1,3,7)$$

$$y = f(A,B,C) = \sum m\ (0,1,2,4,5)$$

مخطط كارنوف لثلاثة متغيرات يتكون من 8 خلايا مرتبة في شكل صفين و أربعة أعمدة

نضع متغيرات الدخل على الصفوف و الأعمدة مبتدئين بالمتغير الواقع في الخانة العليا (MSB)، حيث نضع المتغيرين A و B معاً على الأعمدة، و نضع المتغير C على الصفوف. لاحظ أنه توجد أربعة احتمالات لقيم المتغيرين A و B معاً، يوضع كل احتمال منها على أحد الأعمدة، **<u>مع مراعاة عكس ترتيب العمودين الأخيرين</u>** (و سنقوم بتوضيح سبب ذلك لاحقاً). و يوجد احتمالان فقط لقيم المتغير C يوضعان على الصفوف.

B \ A	00	01	11	10
0				
1				

الخطوة التالية هي ترقيم الخلايا من أعلى إلى أسفل و من اليسار إلى اليمين، مع مراعاة عكس ترتيب العمودين الأخيرين

B \ A	00	01	11	10
0	0	2	6	4
1	1	3	7	5

الخطوة الأخيرة هي وضع اسم متغير الخرج على يمين المخطط و وضع الـــ 1's في الخلايا المناسبة

C \ AB	00	01	11	10	
0	1 0	2	6	4	x
1	1 1	1 3	1 7	5	

C \ AB	00	01	11	10	
0	1 0	1 2	6	1 4	y
1	1 1	3	7	1 5	

لاحظ أنه في حالة وجود أكثر من متغير خرج واحد فإننا نحتاج لمخطط كارنوف منفصل لكل متغير من متغيرات الخرج.

مخطط كارنوف لأربعة متغيرات:

سنقوم بتحويل جدول الصواب التالي إلى مخطط كارنوف في أربعة متغيرات.

#	A	B	C	D	x
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	0
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	1
9	1	0	0	1	0
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

التعبيرات المنطقية في صورة مجموع الحدود الصغرى

$$x = f(A,B,C,D) = \sum m\ (2,3,8,10,12)$$

مخطط كارنوف لأربعة متغيرات يتكون من 16 خلية مرتبة في شكل أربعة صفوف و أربعة أعمدة

نضع متغيرات الدخل على الصفوف و الأعمدة مبتدئين بالمتغير الواقع في الخانة العليا (MSB)، حيث نضع المتغيرين A و B معاً على الأعمدة، و نضع المتغيرين C و D معاً على الصفوف. **و نراعي عكس ترتيب العمودين الأخيرين و الصفين الأخيرين**

B \ A	**00**	**01**	**11**	**10**
00				
01				
11				
10				

بعد ذلك نقوم بترقيم الخلايا من أعلى إلى أسفل و من اليسار إلى اليمين، مع مراعاة عكس ترتيب العمودين الأخيرين و الصفين الأخيرين

B \ A	**00**	**01**	**11**	**10**
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

أخيراً نقوم بوضع اسم متغير الخرج على يمين المخطط و وضع الـــ 1's في الخلايا المناسبة

$B \backslash A$				
00	$_{0}$	$_{4}$	1 $_{12}$	1 $_{8}$
01	$_{1}$	$_{5}$	$_{13}$	$_{9}$
11	1 $_{3}$	$_{7}$	$_{15}$	$_{11}$
10	1 $_{2}$	$_{6}$	$_{14}$	1 $_{10}$

استخدام مخططات كارنوف في تبسيط التعبيرات المنطقية

في مخطط كارنوف يتحول التشابه ما بين الحدود إلى تجاور ما بين الخلايا، فالحدود الواقعة في خلايا متجاورة على المخطط هي حدود متشابهة يمكن جمعها بغرض الاختصار. مثلاً

$B \backslash A$	**0**	**1**	
0	1 $_{0}$	1 $_{2}$	x
1	$_{1}$	$_{3}$	

في مخطط كارنوف لمتغيرين الموضح أعلاه الحدان الواقعان في الخليتين المتجاورتين 0 و 2 هما حدان متشابهان يمكن جمعهما بغرض الاختصار.

$C \backslash AB$	**00**	**01**	**11**	**10**	
0	1 $_{0}$	$_{2}$	$_{6}$	$_{4}$	x
1	1 $_{1}$	1 $_{3}$	1 $_{7}$	$_{5}$	

في مخطط كارنوف لثلاثة متغيرات الموضح أعلاه الحدان الواقعان في الخليتين المتجاورتين 0 و 1 هما حدان متشابهان، و الحدان الواقعان في الخليتين المتجاورتين 3 و 7 هما حدان متشابهان أيضاً، كما أن الحدان الواقعان في الخليتين المتجاورتين 1 و 3 هما حدين متشابهين أيضاً. و كما نعلم فإن أي حدين متشابهين يمكن جمعهما بغرض الاختصار.

$CD \backslash AB$	**00**	**01**	**11**	**10**
00	0	4	1 12	1 8
01	1	5	13	9
11	1 3	1 7	15	11
10	1 2	6	14	10

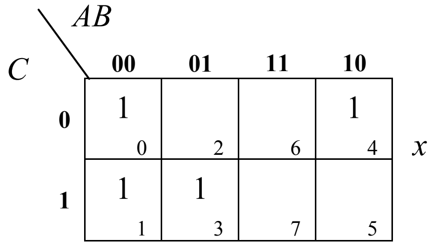
x

في مخطط كارنوف لأربعة متغيرات الموضح أعلاه الحدان 8 و 12 متشابهين، و الحد 3 يتشابه مع كل من الحد 2 و الحد 7.

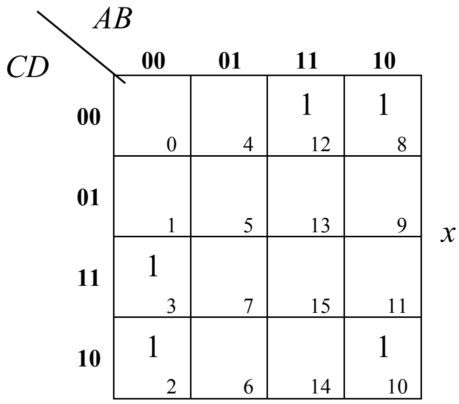
ملاحظة

لاحظ أن سبب قيامنا بعكس ترتيب العمودين الأخيرين في مخطط كارنو لثلاثة متغيرات، و عكس ترتيب العمودين الأخيرين و الصفين الأخيرين في مخطط كارنوف لأربعة متغيرات هو سعينا لجعل الحدود الواقعة في خلايا متجاورة حدوداً متشابهة. تذكر أن الحدين المتشابهين هما حدان يتشابهان في كل شيء عدا متغير واحد يظهر في أحدهما معكوساً و في الآخر بدون عكس.

هنالك تجاورات غير ظاهرة بصورة مباشرة، حيث إنه في حالة وجود أربعة صفوف أو أربعة أعمدة فإن كل خلية في العمود الأول تجاور الخلية المناظرة لها في الموقـــع فـــي العمود الأخير، و كذلك فإن كل خلية في الصف الأول تجاور الخلية المناظرة لهـــا فـــي الموقع في الصف الأخير. مثلاً



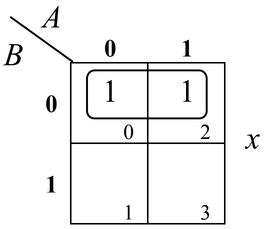
في مخطط كارنوف لثلاثة متغيرات الموضح أعلاه الحد 1 يشبه الحد 3، و الحد 0 يشبه كل من الحد 1 و الحد 4.



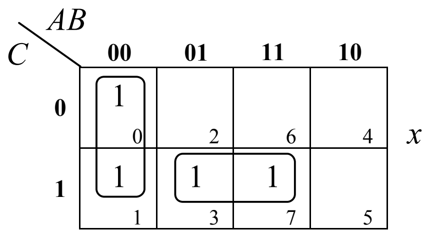
في مخطط كارنوف لأربعة متغيرات الموضح أعلاه الحد 8 يشبه الحـــد 12، و الحـــد 2 يشبه الحد 3، و الحد 10 يشبه كل من الحد 8 و الحد 2.

تكوين المجموعات في مخططات كارنوف

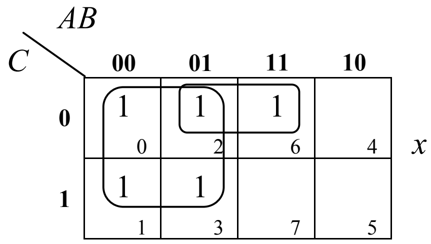
لتبسيط التعبير المنطقي باستخدام مخططات كارنوف نقوم بتكوين مجموعات من الحدود المتشابهة، و هذه المجموعات قد تكون ثنائية أو رباعية أو ثمانية. المجموعة الثنائية تتكون من حدين متشابهين و تختصر متغيراً واحداً، و المجموعة الرباعية تتكون من أربعة حدود متشابهة و تختصر متغيرين، أما المجموعة الثمانية فتتكون من ثمانية حدود متشابهة و تختصر ثلاثة متغيرات. أي أنه كلما كانت المجموعة أكبر كان الاختصار أكثر. فعلينا أن نحاول إدخال أي حد في مجموعة، و كلما كانت تلك المجموعة أكبر كان ذلك أفضل. مثلاً



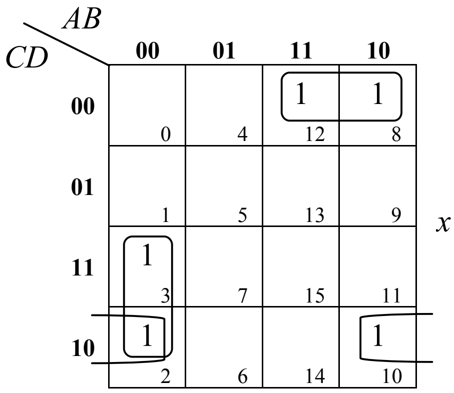
في مخطط كارنوف لمتغيرين الموضح أعلاه قمنا بتكوين مجموعة ثنائية من الحدين 0 و 2.



في مخطط كارنوف لثلاثة متغيرات الموضح أعلاه قمنا بتكوين مجموعتين ثنائيتين. مجموعة ثنائية من الحدين 0 و 1، و مجموعة ثنائية أخرى من الحدين 3 و 7. لاحظ أنه على الرغم من أن الحدين 1 و 3 يتشابهان إلا أننا لا نحتاج لتكوين مجموعة ثنائية منهما لأن كلاً منهما قد دخل في مجموعة أخرى.



في مخطط كارنوف لثلاثة متغيرات الموضح أعلاه قمنا بتكوين مجموعة رباعية و مجموعة ثنائية. لاحظ أن الحد 2 قد دخل في تكوين كل من المجموعة الثنائية و المجموعة الرباعية، و هذا يناظر تكرار الحد الذي يتشابه مع أكثر من حد آخر.



في مخطط كارنوف لأربعة متغيرات الموضح أعلاه قمنا بتكوين ثلاثة مجموعات ثنائية. لاحظ المجموعة المكونة من الحدين 2 و 10. لاحظ أيضاً أنه كان في إمكاننا جمع الحد 10 في مجموعة ثنائية مع الحد 8 بدلاً عن الحد 2.

كتابة التعبير المنطقي المختصر من مخطط كارنوف

بعد تكوين المجموعات بالطريقة المثلى يتم كتابة التعبير المنطقي المختصر مباشرة من مخطط كارنوف، و ذلك بكتابة حد مختصر لكل مجموعة. لكتابة الحد المختصر لمجموعة معينة ننظر لقيم المتغيرات داخل المجموعة، فأي متغير تتغير قيمته داخل المجموعة من 0 إلى 1 أو من 1 إلى 0 يتم اختصاره، أما المتغير الذي تكون قيمته ثابتة

داخل المجموعة فنأخذه معكوساً إذا كانت قيمته ثابتة في 0، و بدون عكس إذا كانت قيمته ثابتة في 1، ثم نقوم بربط هذه المتغيرات مع بعضها البعض بعمليات AND. وبعد كتابة الحد المختصر المقابل لكل مجموعة نقوم بربط هذه الحدود مع بعضها البعض
بعمليات OR (أي نجمعها).

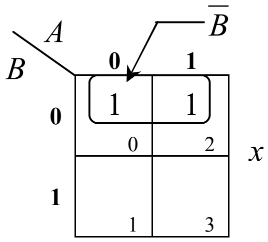
مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B) = \sum m\ (0,2)$$

<u>الحل:</u>

نقوم برسم مخطط كارنوف و تكوين المجموعات



لدينا مجموعة واحدة لذلك يوجد حد واحد فقط. لكتابة الحد ننظر لقيم المتغيرات داخل المجموعة فنجد أن المتغير A قد تغيرت قيمته من 1 إلى 0 داخل المجموعة فيختصر، أما المتغير B فقيمته ثابتة في 0 داخل المجموعة لذلك نأخذه معكوساً. و بالتالي فإن التعبير المختصر هو

$$x = \overline{B}$$

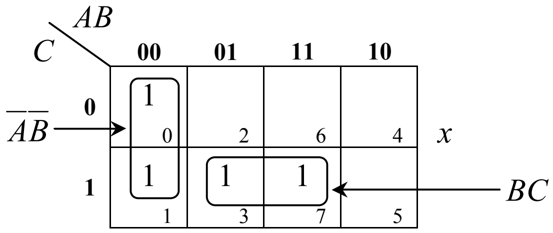
<u>**مثال:**</u>

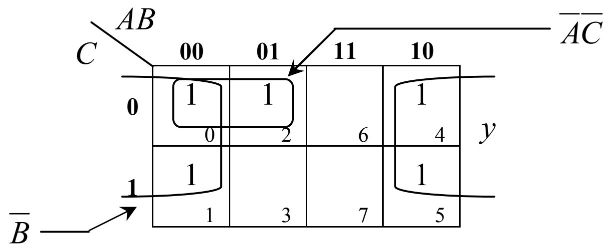
استخدم مخططات كارنوف في تبسيط التعبيرين المنطقيين

$$x = f(A,B,C) = \sum m\ (0,1,3,7)$$
$$y = f(A,B,C) = \sum m\ (0,1,2,4,5)$$

<u>**الحل:**</u>

نقوم برسم مخطط كارنوف لكل متغير من متغيري الخرج ثم نقوم بتكوين المجموعات





في مخطط كارنوف للمتغير x لدينا مجموعتان ثنائيتان. المجموعة الرأسية فيها كلا المتغيرين A و B ثابتين في 0 داخل المجموعة لذلك نأخذهما معكوسين، أما المتغير C فقد تغيرت قيمته داخل المجموعة فيختصر، و عليه يكون الحد المختصر المقابل لهذه المجموعة هو $\overline{A}\,\overline{B}$. أما بالنسبة للمجموعة الأفقية فقد تغيرت قيمة المتغير A داخلها لذلك يتم إختصاره، أما المتغيران B و C فكلاهما ثابت في 1 داخل المجموعة لذلك نأخذهما بدون عكس، و عليه يكون الحد المختصر المقابل لهذه المجموعة هو BC.
أخيراً نقوم بربط الحدود المختصرة المقابلة للمجموعات بعمليات OR فنحصل على

$$x = \overline{A}\,\overline{B} + BC$$

أما بالنسبة لمخطط كارنوف للمتغير y فلدينا مجموعة رباعية و مجموعة ثنائية. في المجموعة الرباعية كلا المتغيرين A و C تغيرت قيمتهما داخل المجموعة لذلك يتم إختصارهما، أما المتغير B فقيمته ثابتة داخل المجموعة في 0 لذلك نأخذه معكوساً، أي أن الحد المختصر المقابل لهذه المجموعة هو $\overline{B}$. أما في المجموعة الثنائية فالمتغير B تغيرت قيمته داخل المجموعة فيتم إختصاره، أما المتغيرين A و C فكلاهما ثابت داخل المجموعة في 0 لذلك نأخذهما معكوسين، أي أن الحد المختصر المقابل لهذه المجموعة هو $\overline{A}\overline{C}$. أخيراً نقوم بربط الحدود المختصرة المقابلة للمجموعات بعمليات OR فنحصل على

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

<u>مثال:</u>

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (1,2,3,5,7,8,10,12)$$

<u>الحل:</u>

x

$CD \backslash AB$	00	01	11	10
00	0	4	1 (12)	1 (8)
01	1 (1)	1 (5)	13	9
11	1 (3)	1 (7)	15	11
10	1 (2)	6	14	1 (10)

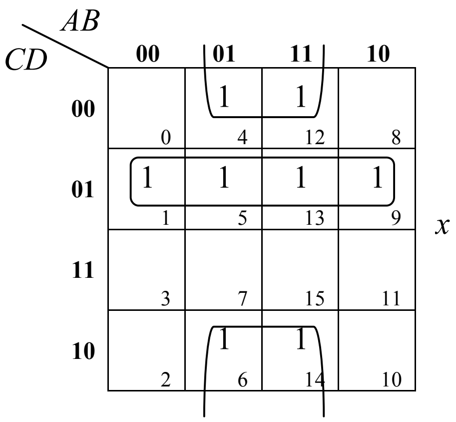
$$x = \overline{A}D + A\overline{C}\overline{D} + \overline{B}C\overline{D}$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (1,4,5,6,9,12,13,14)$$

الحل:



$$x = \overline{C}D + B\overline{D}$$

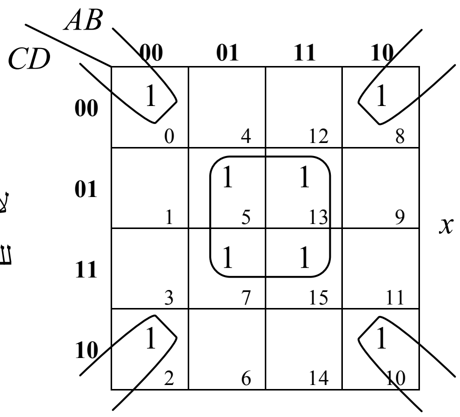
مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (0,2,5,7,8,10,13,15)$$

الحل:

لاحظ أن الزوايا الأربع
للمخطط تشكل مجموعة رباعية



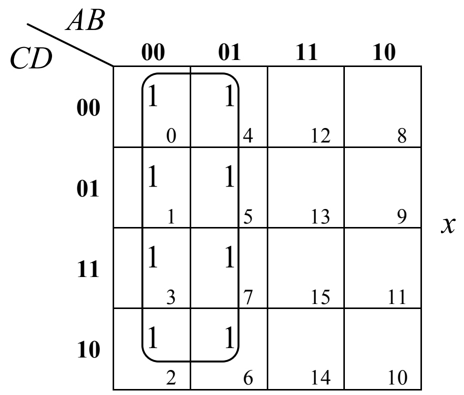
$$x = \overline{B}\,\overline{D} + BD$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (0,1,2,3,4,5,6,7)$$

الحل:



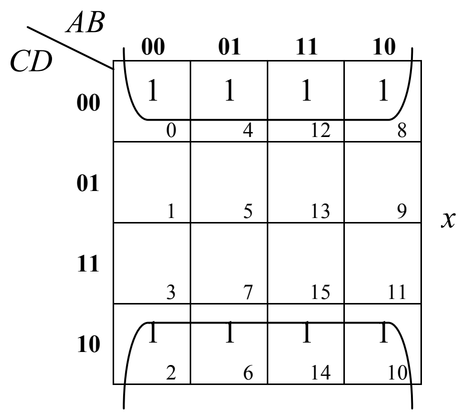
$$x = \overline{A}$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (0,2,4,6,8,10,12,14)$$

الحل:



$$x = \overline{D}$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (0,2,3,4,5,7,8,9,13,15)$$

الحل:

$CD \backslash AB$	00	01	11	10
00	1_{0}	1_{4}	$_{12}$	1_{8}
01	$_{1}$	1_{5}	1_{13}	1_{9}
11	1_{3}	1_{7}	1_{15}	$_{11}$
10	1_{2}	$_{6}$	$_{14}$	$_{10}$

x

$$x = BD + \overline{A}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (1,5,6,7,11,12,13,15)$$

الحل:

$CD \backslash AB$	00	01	11	10
00	$_{0}$	$_{4}$	1_{12}	$_{8}$
01	1_{1}	1_{5}	1_{13}	$_{9}$
11	$_{3}$	1_{7}	1_{15}	1_{11}
10	$_{2}$	1_{6}	$_{14}$	$_{10}$

x

$$x = AB\overline{C} + \overline{A}\,\overline{C}D + \overline{A}BC + ACD$$

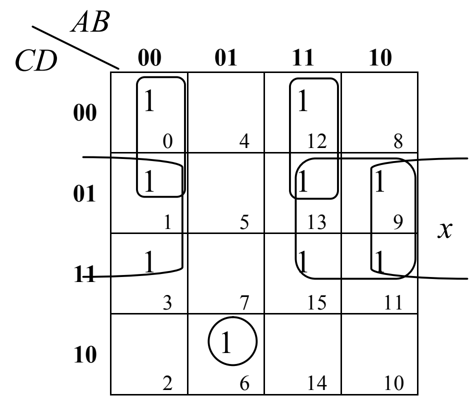
لاحظ أنه من السهل في المثال السابق الوقوع في خطأ تكـــوين المجموعـــة الرباعيـــة، الموضحة بالخط المتقطع، و التي لا داعي لها نظراً إلى أن جميع الحدود المكونة لها قد دخلت في تكوين المجموعات الثنائية. لتجنب الوقوع في مثل هذا الخطأ يجب ألا نبـــدأ بتكوين المجموعات الكبيرة، بل نبدأ دائماً بتكوين المجموعـــات الصـــغيرة ثـــم ننتقـــل للمجموعات الأكبر. أي نبدأ بتكوين المجموعات الأحادية، فأي حد لا يمكن جمعه مـــع أي حد آخر نكون منه مجموعة أحادية. ثم ننتقل لتكوين المجموعات الثنائية، فأي حـــد لا يمكن ضمه إلا لمجموعة ثنائية نستخدمه في تكـــوين مجموعـــة ثنائيـــة. و بـــنفس الأسلوب ننتقل لتكوين المجموعات الرباعية ثم تكوين المجموعات الثمانية.

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\ (0,1,3,6,9,11,12,13,15)$$

<u>الحل</u>:



$$x = \overline{A}BC\overline{D} + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + AD + \overline{B}D$$

مخططات كارنوف لخمسة و ستة متغيرات

نظراً لكبر عدد الخلايا في مخطط كارنوف لخمسة و ستة متغيرات، و لتسهيل التعامل معه، يتم تقسيم المخطط إلى عدد من مخططات كارنوف لأربعة متغيرات.

خمسة متغيرات:

يتكون المخطط من 32 خلية يتم تقسيمها إلى مخططي كارنوف لأربعة متغيرات، يتكون كل مخطط منهما من 16 خلية. المخطط الأول يمثل النصف الأعلى من جدول الصواب (الحدود من 0 إلى 15)، و المخطط الثاني يمثل النصف الأسفل من جدول الصواب (الحدود من 16 إلى 31)

$A = 0$

$DE \backslash BC$	**00**	**01**	**11**	**10**
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

$A = 1$

$DE \backslash BC$	**00**	**01**	**11**	**10**
00	16	20	28	24
01	17	21	29	25
11	19	23	31	27
10	18	22	30	26

x

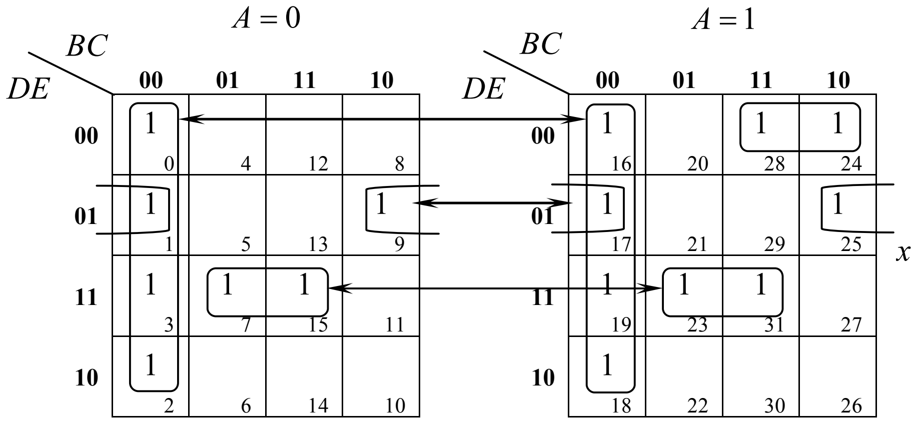
إضافة للتجاورات ما بين الخلايا المألوفة لدينا في مخطط كارنوف لأربعة متغيرات تظهر تجاورات جديدة في مخطط كارنوف لخمسة متغيرات، حيث أن كل خلية في المخطط الأول ($A = 0$) تجاور الخلية المماثلة لها في الموقع في المخطط الثاني ($A = 1$)، مثلاً الخلية 0 تجاور الخلية 16، و الخلية 5 تجاور الخلية 21، و الخلية 10 تجاور الخلية 26. و يمكن تصور هذه التجاورات الجديدة بوضع المخططين فوق بعضهما البعض، فكل خلية في المخطط الأول تجاور التي تعلوها في المخطط الثاني.

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \sum m\,(0,1,2,3,7,9,15,16,17,18,19,23,24,25,28,31)$$

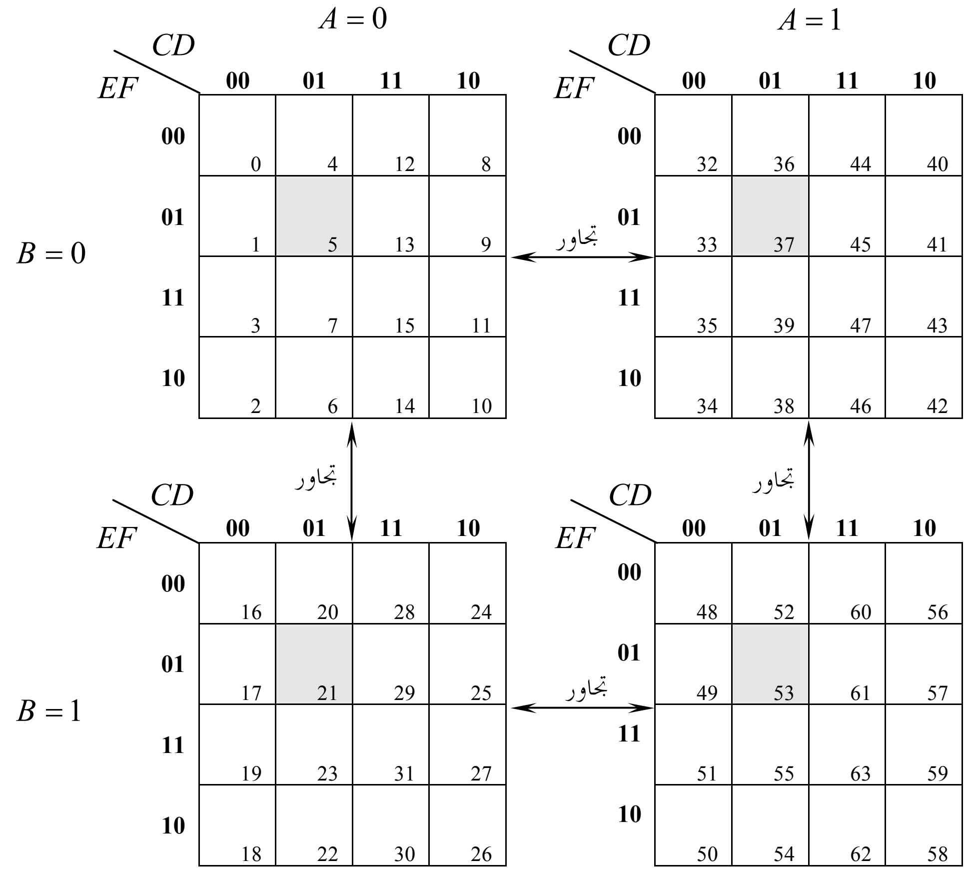
<u>الحل:</u>



$$x = \overline{B}\,\overline{C} + CDE + \overline{C}\,\overline{D}E + AB\overline{D}\,\overline{E}$$

ستة متغيرات:

يتكون المخطط من 64 خلية يتم تقسيمها إلى أربعة مخططات كارنوف لأربعة متغيرات، يمثل كل مخطط من الأربعة أحد أرباع جدول الصواب



لاحظ أن كل خلية تجاور الخلية المماثلة لها في الموقع في المخطط المجاور أفقياً و رأسياً. مثلاً الخلية 5 تجاور كلاً من الخلية 21 و الخلية 37، و الخلية 21 تجاور كلاً من الخلية 5 و الخلية 53، و الخلية 37 تجاور الخلية 5 و الخلية 53، و الخلايا الأربع تكون مع بعضها البعض مجموعة رباعية. و من الواضح صعوبة تكوين المجموعات في مخطط كارنوف لستة متغيرات.

سبعة متغيرات فما فوق

عادة لا يتم استخدام مخططات كارنوف في حالة سبعة متغيرات فما فوق، و إنما يتم في مثل هذه الحالات استخدام إحدىطرق الجدولة (Tabular Methods) التي أشرنا إليها من قبل، و الاستعانة بالحاسوب. و من الناحية العملية عادة ما يتم تجنب تصـــميم دوائـــر منطقية بعدد كبير من متغيرات الدخل، حيث يتم تقسيم مثل هذه الدوائر إلى عـــدد مـــن الوحدات الصغيرة، كل وحدة منها بعدد محدود من متغيرات الدخل، ثم يتم بعد ذلك ربط هذه الوحدات الصغيرة معاً لأداء وظيفة الدائرة الكبيرة.

مخططات كارنوف للتعبيرات المنطقية في صورة مضروب الحدود الكبرى

هنا يتم وضع 0's في الخلايا المقابلة للحدود الكبرى و تُترك بقية الخلايا فارغة، ثم يـــتم تكوين مجموعات من الـــ 0's بنفس الأسلوب المتبع من قبل في تكوين مجموعات مـــن الـــ 1's، و أخيراً يتم كتابة التعبير المختصر في صورة مضـــروب الحـــدود الكبـــرى. فلكتابة الحد المختصر المقابل لمجموعة معينة ننظر إلى قيم المتغيرات داخل المجموعة، فأي متغير تتغير قيمته داخل المجموعة يتم اختصـــاره، أمـــا المتغيـــر الثابـــت داخـــل المجموعة فيؤخذ معكوساً إذا كان ثابتاً في 1، و يؤخذ بدون عكس إذا كان ثابتاً فـــي 0، ثم يتم ربط المتغيرات معاً في الحد بعمليات OR. بعد كتابة الحد المختصر المقابل لكل مجموعة يتم ربط تلك الحدود معاً بعمليات AND.

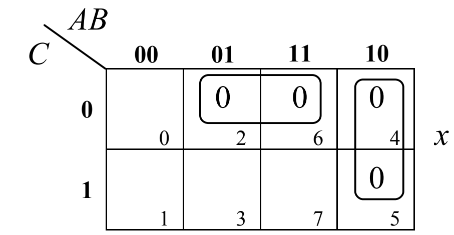
مثال:

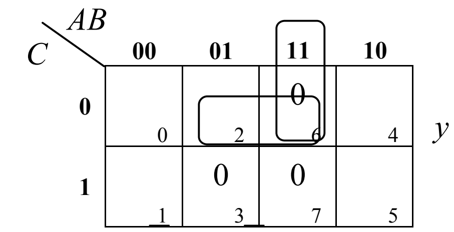
استخدم مخططات كارنوف في تبسيط التعبيرين المنطقيين

$$x = f(A,B,C) = \prod M(2,4,5,6)$$

$$y = f(A,B,C) = \prod M(3,6,7)$$

الحل:





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

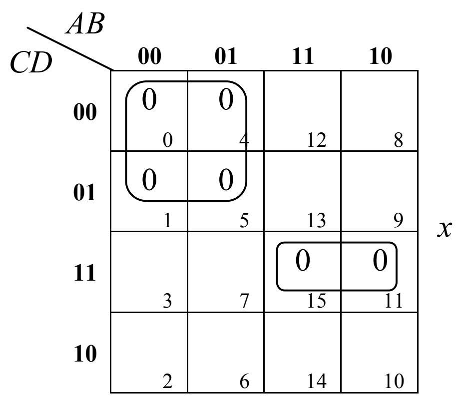
$$y = (\overline{A} + \overline{B})(\overline{B} + \overline{C})$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \prod M(0,1,4,5,11,15)$$

الحل:



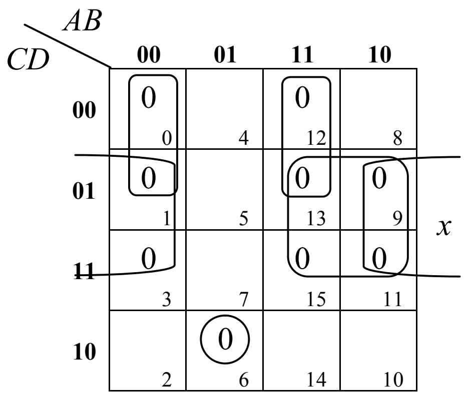
$$x = (A + C)(\overline{A} + \overline{C} + \overline{D})$$

مثال:

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$x = f(A,B,C,D) = \prod M(0,1,3,6,9,11,12,13,15)$$

الحل:



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

مخططات كارنوف للتعبيرات المنطقية في صورتي

AND-OR-Invert و OR-AND-Invert

بالنسبة لهاتين الصورتين يتم رسم مخطط كارنوف و كتابة التعبير المختصر لمعكوس متغير الخرج. حيث أن التعبير المنطقي لمعكوس متغير الخرج في صورة AND-OR-Invert هو عبارة عن تعبير في صورة مجموع الحدود الصغرى، و التعبير المنطقي لمعكوس متغير الخرج في صورة OR-AND-Invert هو عبارة عن تعبير في صورة مضروب الحدود الكبرى. و بعد كتابة التعبير المختصر يتم عكسه بالكامل.

مثال:

من جدول الصواب التالي استخدم مخططات كارنوف لكتابة التعبير المختصر لكل من متغيري الخرج x و y في صورة:

#	A	B	C	x	y
0	0	0	0	1	1
1	0	0	1	1	1
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	0	1
6	1	1	0	0	0
7	1	1	1	1	0

(أ) AND-OR-Invert

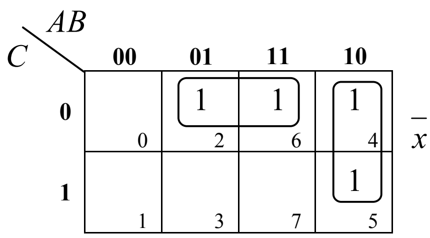
(ب) OR-AND-Invert

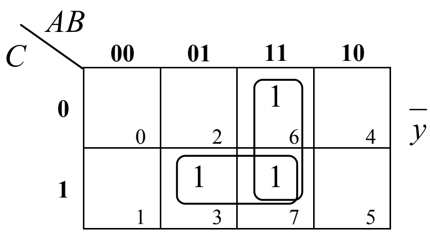
<u>الحل:</u>

(أ) صورة AND-OR-Invert

$$\overline{x} = f(A,B,C) = \sum m\ (2,4,5,6)$$

$$\overline{y} = f(A,B,C) = \sum m\ (3,6,7)$$





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

$$\overline{y} = AB + BC$$
$$y = \overline{AB + BC}$$

ب) صورة OR-AND-Invert

$$\overline{x} = f(A,B,C) = \prod M(0,1,3,7)$$
$$\overline{y} = f(A,B,C) = \prod M(0,1,2,4,5)$$

$\overline{x}$

C \ AB	00	01	11	10
0	0 0	2	6	4
1	0 1	0 3	0 7	5

$\overline{y}$

C \ AB	00	01	11	10
0	0 0	0 2	6	0 4
1	0 1	3	7	0 5

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

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

الدوال غير المحددة بالكامل (Incompletely Specified Functions)

في بعض الدوائر المنطقية تكون قيم الخرج المقابلة لبعض احتمالات الدخل غير محددة، أي غير معلوم ما إذا كانت مساوية 1 أو 0، و تسمى بـ القيم غير المحددة (Don't Cares)، ويرمز لها في جدول الصواب و في مخططات كارنوف بالرمز ×. و السبب في عدم تحديد قيم الخرج تلك يرجع لأحد سببين:

1) إن قيمها لا تؤثر في وظيفة الدائرة المنطقية، أي أن الدائرة تؤدي الوظيفة المطلوبة منها سواء أكانت أي من تلك القيم مساوية 1 أو مساوية 0.

2) إن احتمالات الدخل المقابلة لها في جدول الصواب غير واردة، أي لا يمكن ظهور أي من هذه الاحتمالات في دخل الدائرة المنطقية.

و في ما يلي نوضح طريقة ظهور القيم غير المحددة في جداول الصواب، و في مخططات كارنوف، و في التعبيرات المنطقية المكتوبة في صورة مجموع الحدود الصغرى و في صورة مضروب الحدود الكبرى

#	A	B	C	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	×
6	1	1	0	0
7	1	1	1	×

$$y = f(A,B,C) = \sum m\ (1,2,4) + \sum d\ (5,7)$$

$$y = f(A,B,C) = \prod M(0,3,6) \cdot \prod d\ (5,7)$$

$C \backslash AB$	00	01	11	10	
0	(0)	1 (2)	(6)	1 (4)	y
1	1 (1)	(3)	× (7)	× (5)	

$C \backslash AB$	00	01	11	10	
0	0 (0)	(2)	0 (6)	(4)	y
1	(1)	0 (3)	× (7)	× (5)	

عند تبسيط التعبيرات المنطقية باستخدام مخططات كارنوف يتم إدخال القيم غير المحددة (×'s) في المجموعات بهدف تكوين مجموعات أكبر، و أي قيمة غير محددة لا تخدم هذا الغرض يتم تجاهلها. مع ملاحظة تجنب الوقوع في خطأ تكــوين مجموعــات مكونــة بالكامل من القيم غير المحددة (×'s).

مثال:

من جدول الصواب التالي قم بكتابة التعبير المنطقي لمتغير الخرج y في صورة:

(أ) مجموع الحدود الصغرى.

(ب) مضروب الحدود الكبرى.

ثم قم بتبسيط كل من التعبيرين الناتجين باستخدام مخططات كارنوف.

#	B_3	B_2	B_1	B_0	y
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	0
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	×
11	1	0	1	1	×
12	1	1	0	0	×
13	1	1	0	1	×
14	1	1	1	0	×
15	1	1	1	1	×

الحل:

(أ) صورة مجموع الحدود الصغرى

$$y = f(B_3, B_2, B_1, B_0) = \sum m\ (1,2,5,6,9) + \sum d\ (10,11,12,13,14,15)$$

B_1B_0 \ B_3B_2	00	01	11	10
00	₍₀₎	₍₄₎	× ₍₁₂₎	₍₈₎
01	1 ₍₁₎	1 ₍₅₎	× ₍₁₃₎	1 ₍₉₎
11	₍₃₎	₍₇₎	× ₍₁₅₎	× ₍₁₁₎
10	1 ₍₂₎	1 ₍₆₎	× ₍₁₄₎	× ₍₁₀₎

y

$$y = \overline{B_1}B_0 + B_1\overline{B_0}$$

(ب) صورة مضروب الحدود الكبرى

$$y = f(B_3, B_2, B_1, B_0) = \prod M(0,3,4,7,8) \cdot \prod d\ (10,11,12,13,14,15)$$

B_1B_0 \ B_3B_2	00	01	11	10
00	0 (0)	0 (4)	× (12)	0 (8)
01	(1)	(5)	× (13)	(9)
11	0 (3)	0 (7)	× (15)	× (11)
10	(2)	(6)	× (14)	× (10)

y

$$y = (B_1 + B_0)(\overline{B_1} + \overline{B_0})$$

تدريب 5

استخدم مخططات كارنوف في تبسيط التعبيرات المنطقية التي قمت بكتابتها في تدريب 1.

تدريب 6

استخدم مخططات كارنوف في تبسيط التعبيرات المنطقية التي قمت بكتابتها في تدريب 2.

تدريب 7

استخدم مخططات كارنوف في تبسيط كل من التعبيرات المنطقية التالية:

$$x = f(A_3, A_2, A_1, A_0) = \sum m\ (0,1,2,3,4,5,8,10,12)$$

$$y = f(L_3, L_2, L_1, L_0) = \sum m\ (1,3,4,5,6,7,8,9,11,13,14,15)$$

$$z = f(D, C, B, A) = \sum m\ (0,1,7,8,9,10,15)$$

تدريب 8

استخدم مخططات كارنوف في تبسيط كل من التعبيرات المنطقية التالية:

$$x = f(A_3, A_2, A_1, A_0) = \prod M(6,7,9,11,13,14,15)$$

$$y = f(L_3, L_2, L_1, L_0) = \prod M(0,2,10,12)$$

$$z = f(D, C, B, A) = \prod M(2,3,4,5,6,11,12,13,14)$$

تدريب 9

استخدم مخططات كارنوف في تبسيط التعبير المنطقي

$$y = f(B_4, B_3, B_2, B_1, B_0) = \prod M(3,4,5,7,12,13,18,19,20,21,22,23,26$$

تدريب 10

استخدم مخططات كارنو في تبسيط كل من التعبيرين المنطقيين التاليين:

$$x = f(A_3, A_2, A_1, A_0) = \sum m\ (0,7,13,15) + \sum d\ (2,6,8,9,10,11,14)$$

$$y = f(B_3, B_2, B_1, B_0) = \prod M(1,3,4,5,12) \cdot \prod d\ (2,6,8,9,10,11,14)$$

أسئلة تقويم ذاتي

1- ما الهدف من تبسيط التعبيرات المنطقية؟ وكيف تتم عملية التبسيط ؟

2- ما الهدف من استخدام مخطط كارنوف؟

3- عدد أسباب عدم تحديد قيم الخرج في بعض الدوائر المنطقية.

4. بناء الدوائر المنطقية

بعد تبسيط التعبيرات المنطقية تأتي الخطوة الأخيرة في عملية تصميم الدوائر المنطقيـــة ألا و هي بناء الدائرة المنطقية. و البناء هنا إما أن يتم باســـتخدام البوابـــات الأساســـية الثلاث (AND و OR و NOT)، و إما أن يتم باستخدام نوع واحد فقط من البوابات (إما NAND أو NOR).

1.4 البناء باستخدام البوابات الأساسية الثلاث

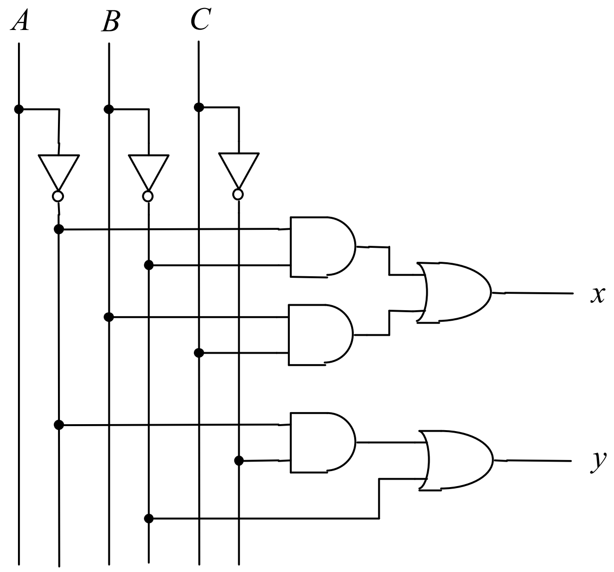
كما ذكرنا من قبل فإن شكل الدائرة المنطقية يعتمد علـــى الصـــورة التـــي كُتبـــت بهـــا التعبيرات المنطقية. فالتعبيرات المكتوبة في صورة مجموع الحدود الصغرى ينتج عنها دائرة في شكل AND-OR Structure، أما التعبيرات المكتوبة فـــي صـــورة مضـــروب الحدود الكبرى فينتج عنها دائرة في شكل OR-AND Structure.

مثال:

استخدم البوابات الأساسية الثلاث (AND و OR و NOT) في بنـــاء الـــدائرة المنطقيـــة الممثلة بالتعبيرين المختصرين التاليين المكتوبين في صورة مجموع الحدود الصغرى

$$x = \overline{A}\overline{B} + BC$$
$$y = \overline{B} + \overline{A}\overline{C}$$

الحل:

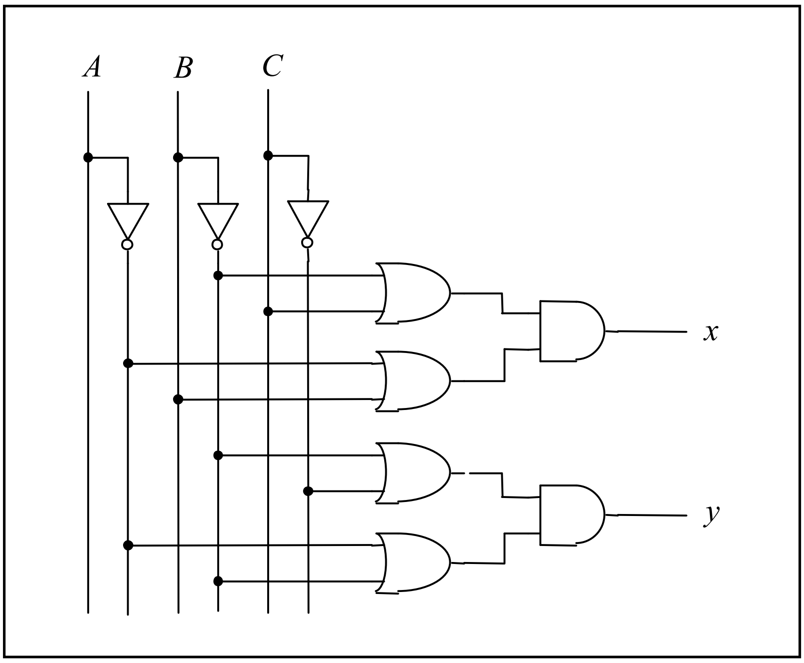


مثال:

استخدم البوابات الأساسية الثلاث (AND و OR و NOT) في بنـــاء الـــدائرة المنطقيـــة الممثلة بالتعبيرين المختصرين التاليين المكتوبين في صورة مضروب الحدود الكبرى

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

الحل:



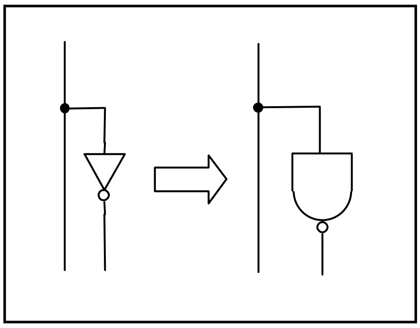
2.4 البناء باستخدام نوع واحد فقط من البوابات

عندما يتم تصنيع الدائرة المنطقية في شكل دائرة متكاملـــة (Integrated Circuit) أو IC فإنه عادة ما يتم بناء الدائرة المنطقية بالكامل باستخدام نوع واحد فقط من البوابـــات. و البوابات المستخدمة هنا إما أن تكون بوابات NAND أو بوابات NOR.

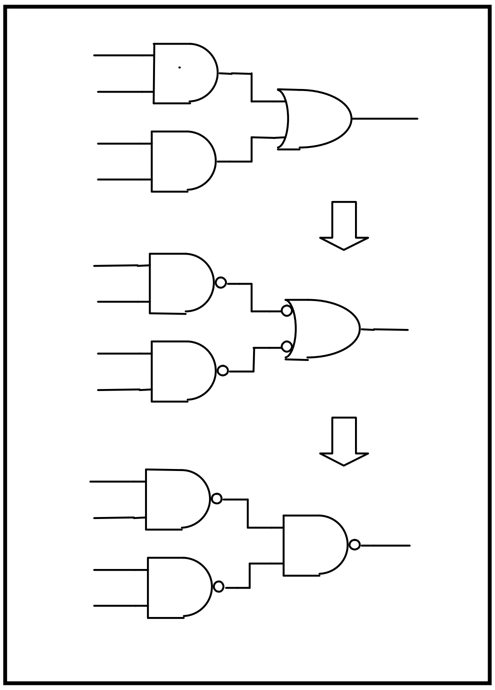
البناء باستخدام بوابات NAND

يجب أن تكون التعبيرات المنطقية مكتوبة في صورة مجموع الحدود الصغرى للحصول على AND-OR Structure، ثم يتم بعد ذلك تحويل الـــ AND-OR Structure إلى دائرة مبنية بالكامل من بوابات NAND كالتالي:

يتم استبدال كل عاكس منطقي ببوابة NAND ذات طرف دخل واحد



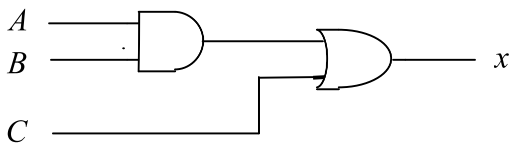
يتم تحويل بوابات AND مع بوابات OR إلى بوابات NAND كالتالي:



حيث نقوم بإجراء عملية عكس منطقي مرتين متتاليتين، مرة في خـرج بوابـة AND و مرة أخرى في دخل بوابة OR، و العكس المنطقي مرتين متتاليتين، كما نعلم، لا يغيـر من قيمة المتغير لأن $\overline{\overline{A}} = A$. ثم نستبدل بوابة OR التي تم عكس جميع أطراف الدخل لها ببوابة NAND، لأن عملية OR مسبوقة بعكس الدخل تكافئ عمليـة NAND، أي أن $\overline{A} + \overline{B} = \overline{A \cdot B}$. أي أنه لتحويل الـ AND-OR Structure إلى دائرة مبنية بالكامل من بوابات NAND فإننا نقوم باستبدال جميع العواكس المنطقية ببوابات NAND، و استبدال

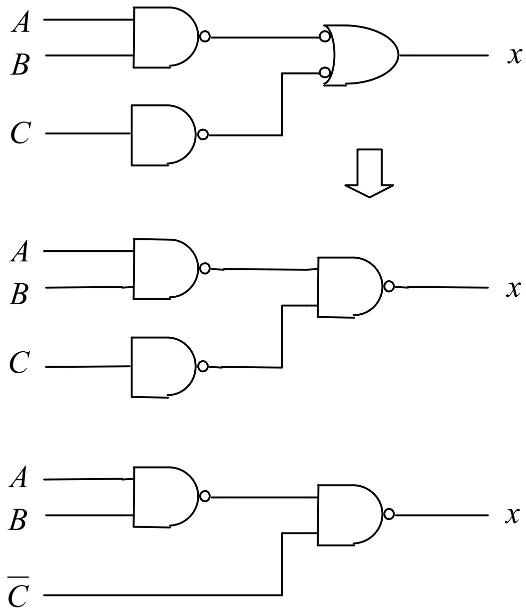
جميع بوابات AND ببوابات NAND، و استبدال جميع بوابـــات OR ببوابـــات NAND، أي نقوم ببساطة باستبدال أي بوابة بالـــ AND-OR Structure ببوابة NAND لها نفـــس عدد أطراف الدخل.

و هناك مشكلة صغيرة قد تواجهنا عند تحويل الـــ AND-OR Structure إلى دائرة مبنية بالكامل من بوابات NAND، و تتضح هذه المشكلة في الدائرة التالية:



نلاحظ هنا أن المتغير C يدخل مباشرة إلى بوابة OR دون أن يكون خارجاً من بوابـــة AND. و لتحويل الدائرة إلى دائرة مبنية بالكامل من بوابات NAND نحتاج، كما نعلم، لإجراء عملية عكس منطقي في كل طرف من أطراف الدخل لبوابـــة OR حتـــى نقـــوم بتحويلها إلى بوابة NAND، و نقوم بإلغاء تأثير هذا العكس المنطقـــي بـــإجراء عكـــس منطقي آخر في خرج بوابة AND. المشكلة هنا هي عدم وجود بوابـــة AND لإجـــراء عملية عكس منطقي في خرجها لمعادلة العكس المنطقي الذي تم في دخـــل بوابـــة OR.

يتم حل هذه المشكلة كالتالي:



قمنا هنا بإضافة بوابة NAND ذات طرف دخل واحد لتقوم بإجراء عملية عكس منطقي تعادل عملية العكس المنطقي التي تمت في دخل بوابة OR. و يمكــن الاســتغناء عــن بوابة NAND ذات طرف الدخل الواحد هذه إذا توفر معكوس المتغيــر الــذي يــدخل مباشرة إلى بوابة OR، حيث نقوم بإدخال هذا المعكوس مباشرة إلى بوابة NAND التـــي حلت محل بوابة OR.

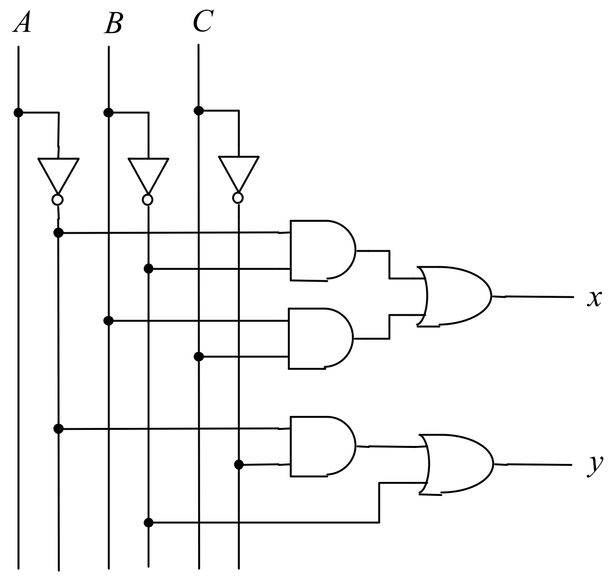
مثال:

قم ببناء الدائرة المنطقية الممثلة بالتعبيرين المنطقيين المختصـــرين التـــاليين باســـتخدام بوابات NAND فقط

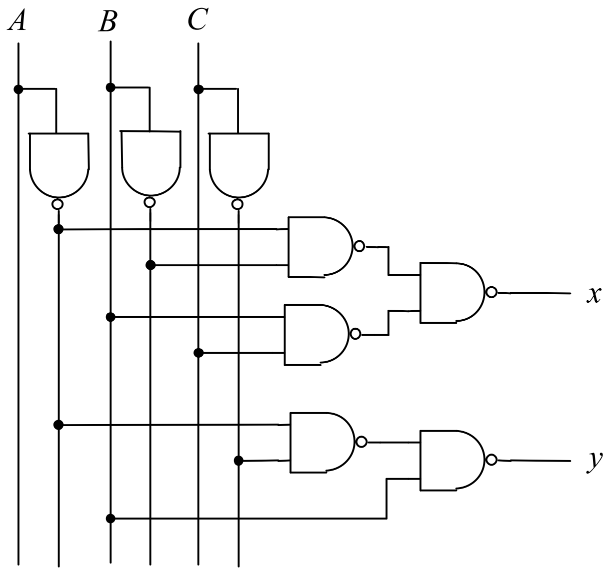
$$x = \overline{A}\overline{B} + BC$$
$$y = \overline{B} + \overline{A}\overline{C}$$

الحل:

نقوم أولاً ببناء الدائرة باستخدام البوابات الأساسية الثلاث (AND و OR و NOT)



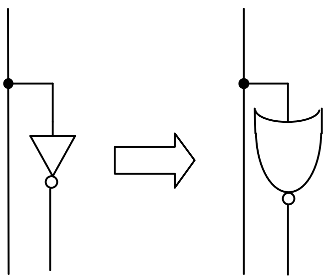
ثم نقوم باستبدال جميع البوابات بالدائرة ببوابات NAND بنفس عدد أطراف الدخل، مع مراعاة عكس المتغير الذي يدخل مباشرة إلى بوابة OR



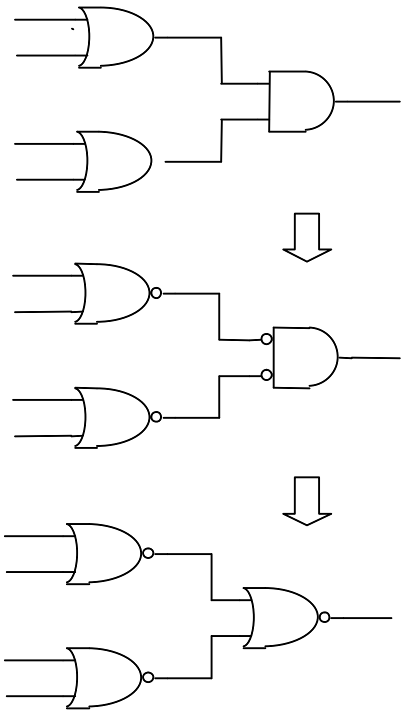
البناء باستخدام بوابات NOR

يجب أن تكون التعبيرات المنطقية مكتوبة في صورة مضروب الحدود الكبرى للحصول على OR-AND Structure، ثم يتم بعد ذلك تحويل الـــ OR-AND Structure إلى دائرة مبنية بالكامل من بوابات NOR كالتالي:

يتم استبدال كل عاكس منطقي ببوابة NOR ذات طرف دخل واحد



- يتم تحويل بوابات OR مع بوابات AND إلى بوابات NOR كالتالي:



حيث نقوم بإجراء عملية عكس منطقي مرتين متتاليتين، مرة في خرج بوابة OR و مرة أخرى في دخل بوابة AND، ثم نستبدل بوابة AND التي تم عكس جميع أطراف الدخل لها ببوابة NOR، لأن عملية AND مسبوقة بعكس الدخل تكــافئ عمليــة NOR، أي أن $\overline{A}\cdot\overline{B}=\overline{A+B}$. أي أنه لتحويل الــ Structure OR-AND إلى دائرة مبنية بالكامل من بوابات NOR فإننا نقوم ببساطة باستبدال أي بوابة ببوابة NOR لها نفس عــدد أطــراف الدخل. وفي حالة دخول متغير مباشرة إلى بوابة AND نقوم بإدخــال معكــوس ذلــك المتغير إلى بوابة NOR التي حلت محل بوابة AND.

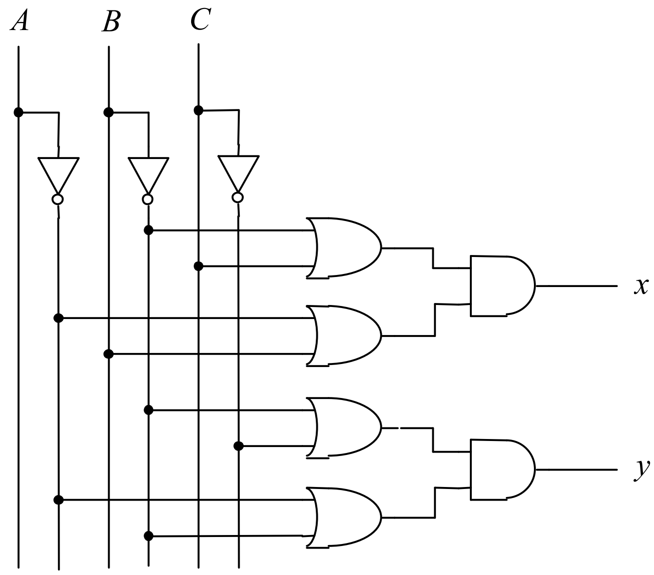
مثال:

قم ببناء الدائرة المنطقية الممثلة بالتعبيرين المنطقيين المختصــرين التــاليين باســتخدام بوابات NOR فقط

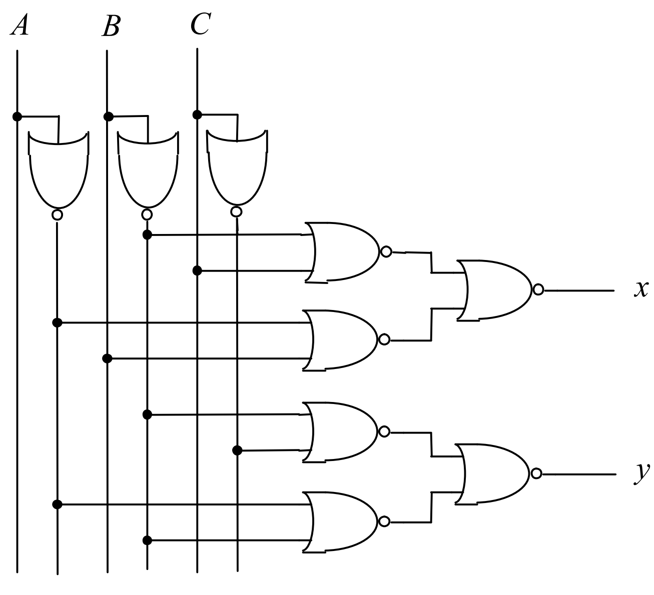
$$x = (\overline{B} + C)(\overline{A} + B)$$
$$y = (\overline{B} + \overline{C})(\overline{A} + \overline{B})$$

الحل:

نقوم أولاً ببناء الدائرة باستخدام البوابات الأساسية الثلاث (AND و OR و NOT)



ثم نقوم باستبدال جميع البوابات بالدائرة ببوابات NOR



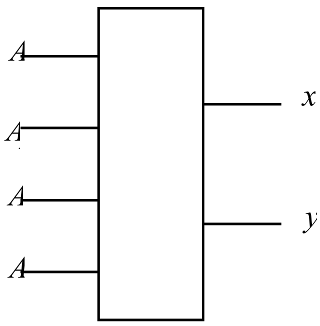
تدريب 11

صمم الدائرة المنطقية التي يوضحها المخطط المنطقي و جدول الصواب لها أدناه، ثم قم ببناء الدائرة باستخدام:

(أ) بوابات NAND فقط

(ب) بوابات NOR فقط

#	A_3	A_2	A_1	A_0	x	y
0	0	0	0	0	0	1
1	0	0	0	1	0	1
2	0	0	1	0	×	×
3	0	0	1	1	0	1
4	0	1	0	0	0	1
5	0	1	0	1	0	0
6	0	1	1	0	×	×
7	0	1	1	1	1	0
8	1	0	0	0	×	×
9	1	0	0	1	×	×
10	1	0	1	0	×	×
11	1	0	1	1	×	×
12	1	1	0	0	0	1
13	1	1	0	1	1	0
14	1	1	1	0	×	×
15	1	1	1	1	1	0



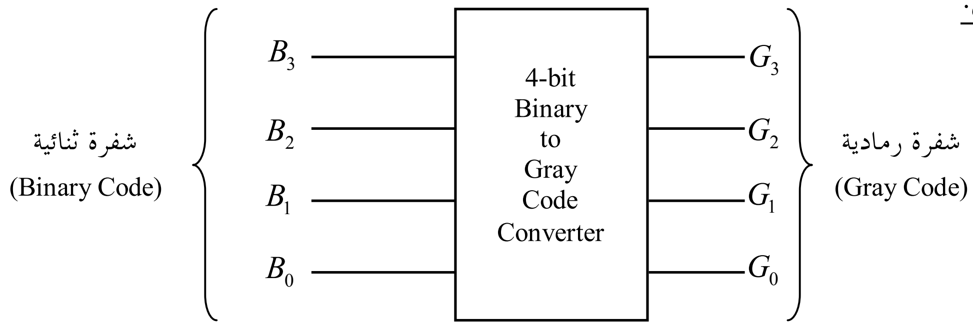
5. أمثلة على تصميم الدوائر المنطقية

سنقوم في هذا الجزء بالتدرب على عملية تصميم الدوائر المنطقية و ذلك بتصميم دوائر منطقية تؤدي وظائف مفيدة.

<u>مثال (1):</u> **4-bit Binary-to-Gray Code Converter**

صمم دائرة منطقية تقوم بتحويل شفرة ثنائية مكونة من 4 خانات إلى الشفرة الرمادية، ثم قم ببناء الدائرة باستخدام بوابات NAND فقط.

الحل:



المخطط المنطقي

#	B_3	B_2	B_1	B_0	G_3	G_2	G_1	G_0
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	1
3	0	0	1	1	0	0	1	0
4	0	1	0	0	0	1	1	0
5	0	1	0	1	0	1	1	1
6	0	1	1	0	0	1	0	1
7	0	1	1	1	0	1	0	0
8	1	0	0	0	1	1	0	0
9	1	0	0	1	1	1	0	1
10	1	0	1	0	1	1	1	1
11	1	0	1	1	1	1	1	0
12	1	1	0	0	1	0	1	0
13	1	1	0	1	1	0	1	1
14	1	1	1	0	1	0	0	1
15	1	1	1	1	1	0	0	0

جدول الصواب

التعبيرات المنطقية:

يجب أن تكون التعبيرات المنطقية في صورة مجموع الحدود الصــغرى لأن المطلــوب بناء الدائرة باستخدام بوابات NAND فقط

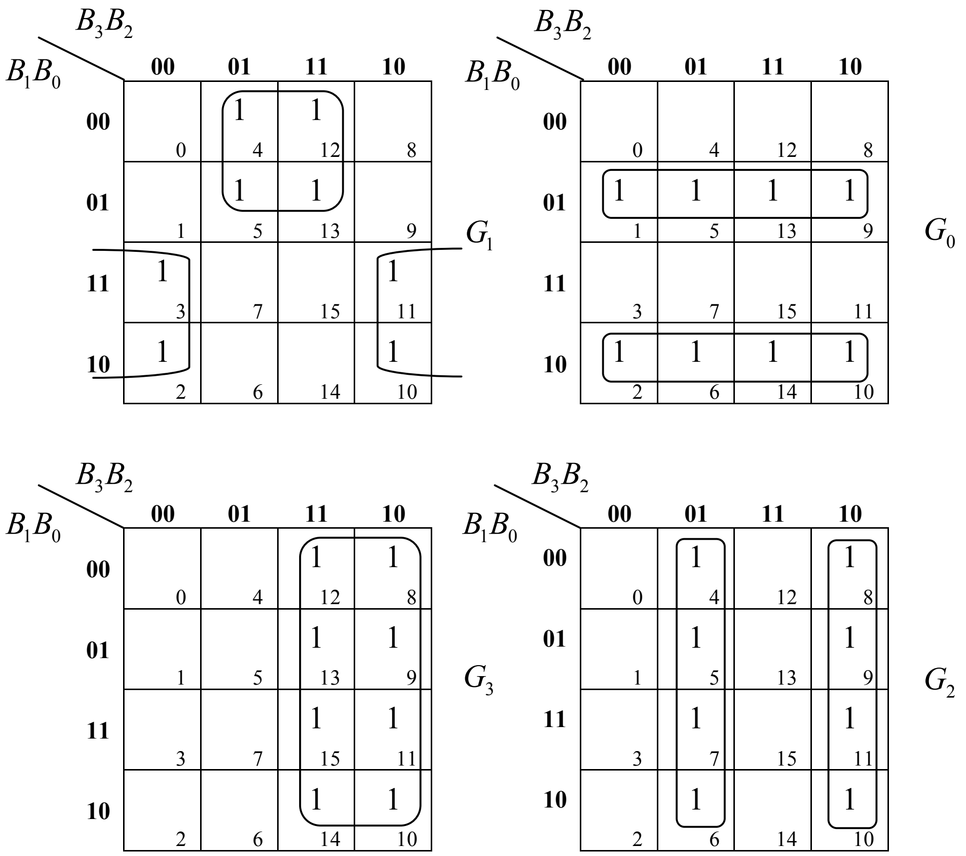
$$G_0 = f(B_3, B_2, B_1, B_0) = \sum m\ (1,2,5,6,9,10,13,14)$$

$$G_1 = f(B_3, B_2, B_1, B_0) = \sum m\ (2,3,4,5,10,11,12,13)$$

$$G_2 = f(B_3, B_2, B_1, B_0) = \sum m\ (4,5,6,7,8,9,10,11)$$

$$G_3 = f(B_3, B_2, B_1, B_0) = \sum m\ (8,9,10,11,12,13,14,15)$$

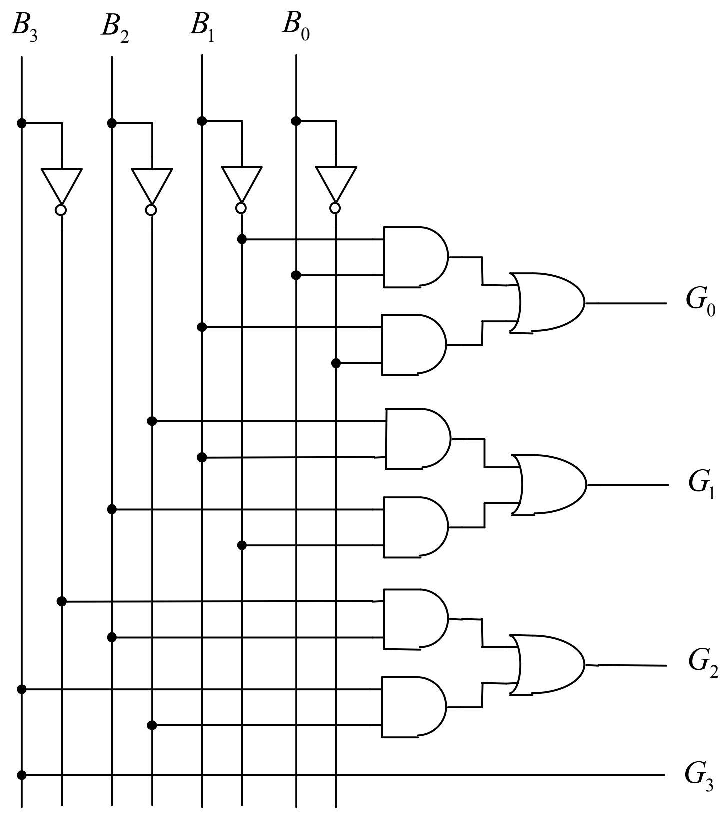
مخططات كارنوف:

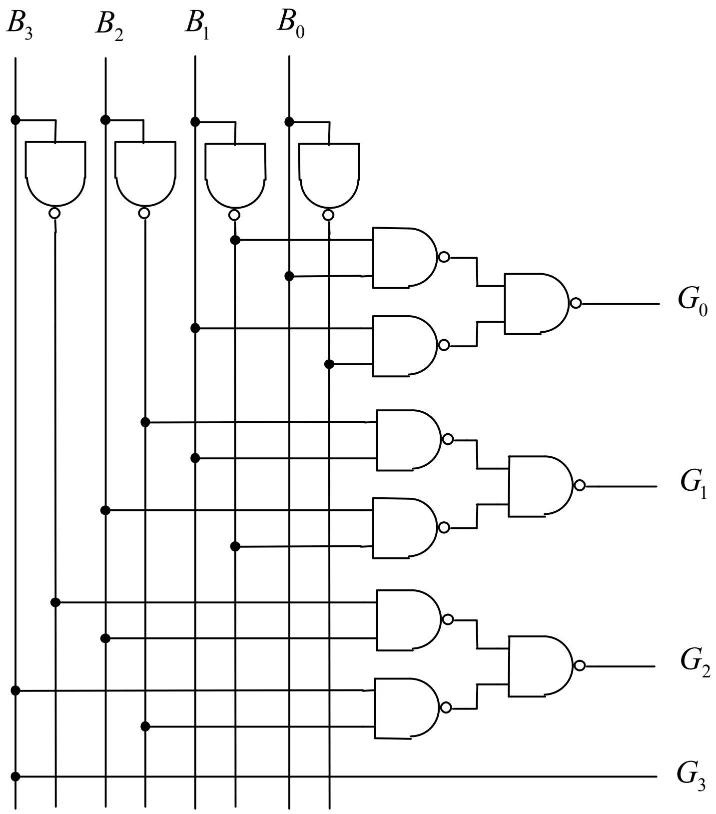


التعبيرات المنطقية المختصرة:

$$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$$
$$G_1 = \overline{B_2}B_1 + B_2\overline{B_1}$$
$$G_2 = \overline{B_3}B_2 + B_3\overline{B_2}$$
$$G_3 = B_3$$

الدائرة المنطقية:

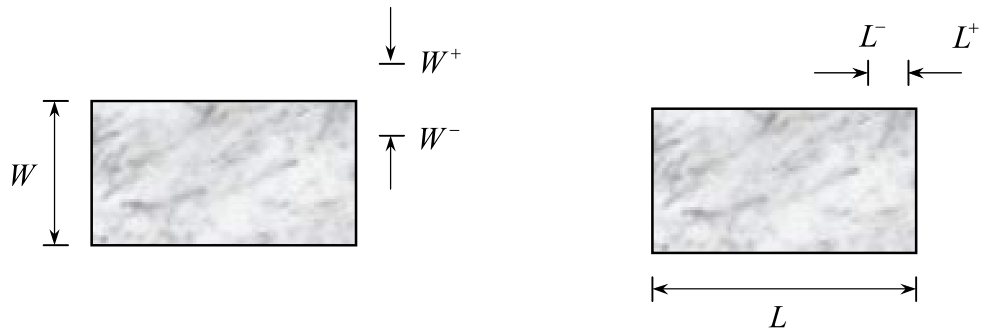



B_3
B_2
B_1
B_0
G_0
G_1
G_2
G_3

مثال (2)

دائرة فحص الأبعاد (Dimensions Checker)

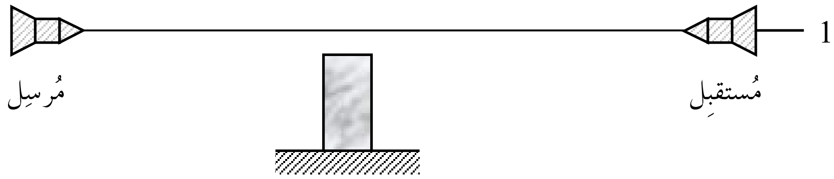
في عملية صناعية معينة يتم تقطيع المادة الخام إلى قطع مستطيلة الشكل طولها L و عرضها W. و حيث أنه من غير الممكن عملياً أن يتم قياس أبعاد القطعة بدقة كاملة فإنه من المقبول أن يكون هناك خطأ في القياس في حدود معينة. فيكون الطول L مقبولاً إذا لم يقل عن الحد الأدنى L^- و لم يتجاوز الحد الأقصى L^+، و يكون العرض W مقبولاً إذا لم يقل عن الحد الأدنى W^- و لم يتجاوز الحد الأقصى W^+، كما هو موضح بالشكل التالي

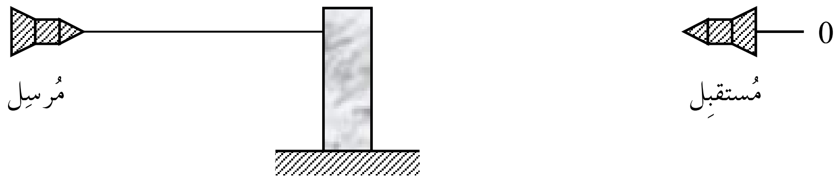


إذا كان الخطأ في القياس بالنقصان، إي إذا كان الطول L أقل من الحد الأدنى L^-، أو إذا كان العرض W أقل من الحد الأدنى W^-، فإن مثل هذا الخطأ غير قابل للإصلاح و تعتبرالقطعة تالفة. أما إذا كان الخطأ في القياس بالزيادة، أي إذا تجاوز الطول L الحد الأقصى L^+، أو إذا تجاوز العرض W الحد الأقصى W^+، فإن مثل هذا الخطأ يمكن إصلاحه بإعادة عملية القطع و إزالة الجزء الزائد.

يمكن التأكد من أن أبعاد القطعة ضمن الحدود المطلوبة باستخدام مجسات (Sensors). و أبسط نوع من المجسات يمكن أن يستخدم هنا هو مجس ليزر (Laser Sensor) يتكون من جزئين، جزء مُرسِل يقوم بتوليد شعاع من الليزر، و جزء آخر مُستقبِل يقوم باستقبال الشعاع و توليد إشارة كهربائية تدل على سقوط شعاع الليزر عليه. و عند قيام أي جسم باعتراض طريق شعاع الليزر فإنه يقوم بحجبه عن المُستقبِل و تغيب بالتالي

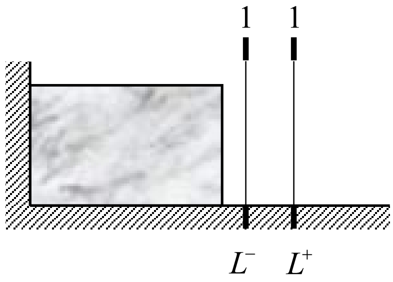
الإشارة الكهربائية التي يقوم بتوليدها المُستقبِل و الدالة على سقوط شعاع الليزر عليـــه، كما هو موضح بالشكل التالي:



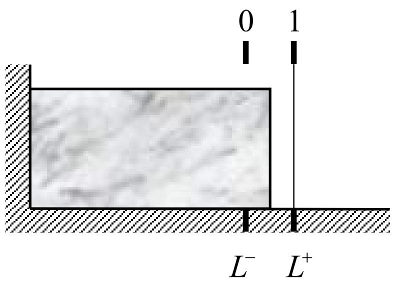


و قد اعتبرنا أن وجود الإشارة الكهربائية التي يولدها المُستقبِل يمثل القيمة المنطقيـــة 1، و غياب تلك الإشارة يمثل القيمة المنطقية 0. أي أن المجس يعطي القيمـــة المنطقيـــة 1 عندما لا يكون شعاع الليزر محجوباً، و يعطي القيمة المنطقية 0 عندما يكـــون الشـــعاع محجوباً.

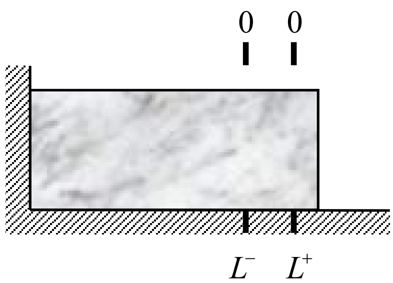
و لمعرفة ما إذا كان الطول L ضمن الحدود المطلوبة نحتاج لاستخدام مجسين، مجـــس يوضع عند الطول الأدنى L^-، و مجس آخر يوضـــع عنـــد الطـــول الأقصـــى L^+. و بملاحظة الإشارات التي يصدرها المجسين يمكن الحصول على المعلومة المطلوبة، كما هو موضح بالشكل التالي:



الطول ناقص



الطول مقبول



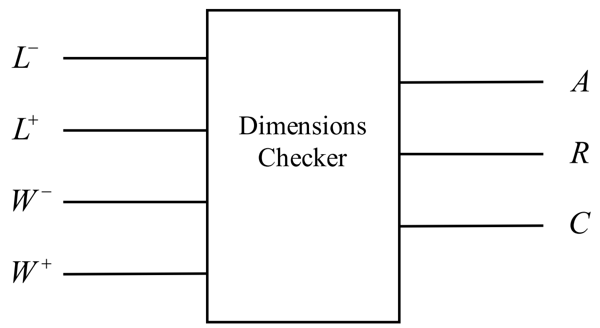
الطول زائد

و بنفس الطريقة نحتاج لاستخدام مجسين للعرض W، يوضع أحدهما عند العرض الأدنى W^- و الآخر عند العرض الأقصى W^+.

المطلوب الآن هو تصميم دائرة فحص الأبعاد (Dimensions Checker)، و هـــي دائــرة منطقية تستقبل الإشارات من المجسات الأربعة (L^-، L^+، W^-، W^+) و تقــوم بتوليــد خرج يوضح ما إذا كانت القطعة التي يتم فحصــها مقبولــة (Accepted) أو مرفوضـــة (Rejected) أو قابلة للإصلاح بإعادة القطع (reCut). أي أن للدائرة ثلاثة أطراف خرج هي A و R و C، فإذا كانت القطعة مقبولة فإن طرف الخرج A يكون مساوياً 1 في حين يكون الطرفان R و C مساويين 0، و إذا كانت القطعة مرفوضـــة فـــإن طـــرف الخرج R يكون مساوياً 1 في حين يكون الطرفان A و C مســـاويين 0، و إذا كانـــت القطعة قابلة للإصلاح بإعادة القطع فإن طرف الخرج C يكون مساوياً 1 في حين يكون الطرفين A و R مساويين 0.

الحل:

المخطط المنطقي و جدول الصواب:



#	W^+	W^-	L^+	L^-	A	R	C
0	0	0	0	0	0	0	1
1	0	0	0	1	×	×	×
2	0	0	1	0	0	0	1
3	0	0	1	1	0	1	0
4	0	1	0	0	×	×	×
5	0	1	0	1	×	×	×
6	0	1	1	0	×	×	×
7	0	1	1	1	×	×	×
8	1	0	0	0	0	0	1
9	1	0	0	1	×	×	×
10	1	0	1	0	1	0	0
11	1	0	1	1	0	1	0
12	1	1	0	0	0	1	0
13	1	1	0	1	×	×	×
14	1	1	1	0	0	1	0
15	1	1	1	1	0	1	0

التعبيرات المنطقية:

نختار صورة التعبيرات المنطقية حسب شكل الدائرة المطلوب، و بما أنه لم يتم تحديد شكل معين للدائرة هنا فلنا مطلق الحرية في اختيار صورة التعبيرات المنطقية، فنختار صورة مجموع الحدود الصغرى

$$A = \sum m\ (10) + \sum d\ (1,4,5,6,7,9,13)$$
$$R = \sum m\ (3,11,12,14,15) + \sum d\ (1,4,5,6,7,9,13)$$
$$C = \sum m\ (0,2,8) + \sum d\ (1,4,5,6,7,9,13)$$

مخططات كارنوف:

$L^{+}L^{-}$ \ $W^{+}W^{-}$	00	01	11	10
00	$_{0}$	$\times_{4}$	$_{12}$	$_{8}$
01	$\times_{1}$	$\times_{5}$	$\times_{13}$	$\times_{9}$
11	$_{3}$	$\times_{7}$	$_{15}$	$_{11}$
10	$_{2}$	$\times_{6}$	$_{14}$	1_{10}

A

$L^{+}L^{-}$ \ $W^{+}W^{-}$	00	01	11	10
00	$_{0}$	$\times_{4}$	1_{12}	$_{8}$
01	$\times_{1}$	$\times_{5}$	$\times_{13}$	$\times_{9}$
11	1_{3}	$\times_{7}$	1_{15}	1_{11}
10	$_{2}$	$\times_{6}$	1_{14}	$_{10}$

R

W^{+} \ $W^{+}W^{-}$	00	01	11	10
00	1_{0}	$\times_{4}$	$_{12}$	1_{8}
01	$\times_{1}$	$\times_{5}$	$\times_{13}$	$\times_{9}$
11	$_{3}$	$\times_{7}$	$_{15}$	$_{11}$
10	1_{2}	$\times_{6}$	$_{14}$	$_{10}$

C

التعبيرات المنطقية المختصرة:

$$A = W^{+}\overline{W^{-}}L^{+}\overline{L^{-}}$$
$$R = W^{-} + L^{-}$$
$$C = \overline{L^{-}}\ \overline{W^{+}} + \overline{W^{-}}\ \overline{L^{+}}$$

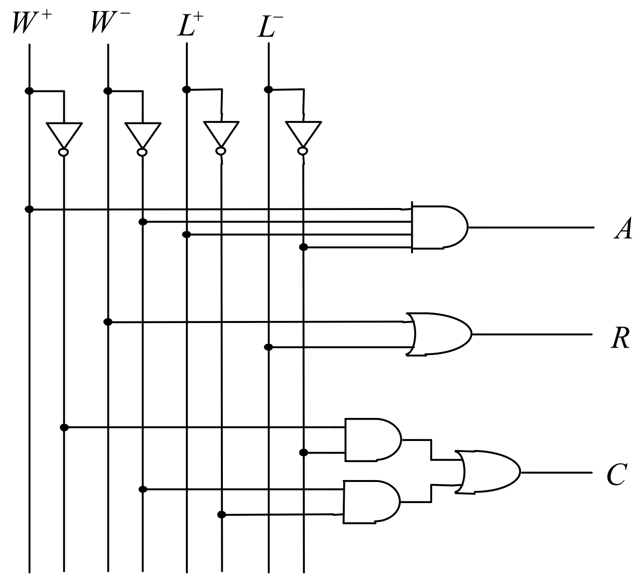
<u>لاحظ</u> من التعبيرات المختصرة أن:

قبول القطعة يتم في حالة واحدة فقط و هي أن يكون كلا مجسي الحد الأدنى غير معطيين لإشارة و كلا مجسي الحد الأقصى معطيين لإشارة، مما يعني أن الطول L و العرض W كلاهما قد تجاوز الحد الأدنى و كلاهما لم يتجاوز الحد الأقصى، أي أن كليهما ضمن الحدود المطلوبة.

رفض القطعة يتم في إحدى حالتين و هما إما أن يكون مجس الحد الأدنى للعرض معطياً إشارة مما يعني أن العرض أقل من الحد الأدنى، أو أن يكون مجس الحد الأدنى للطول معطياً إشارة مما يعنى أن الطول أقل من الحد الأدنى.

إعادة القطع تتم في إحدى حالتين أولاهما هي أن يكون مجس الحد الأدنى للطول غير معطيٍ لإشارة و مجس الحد الأعلى للعرض غير معطيٍ لإشارة أيضاً، مما يعني أن الطول قد تجاوز الحد الأدنى، فهو إما مقبول أو زائد، و العرض قد تجاوز الحد الأعلى فهو زائد، و الحالة الثانية هي أن يكون مجس الحد الأدنى للعرض غير معطياً لإشارة و مجس الحد الأعلى للطول غير معطيٍ لإشارة أيضاً، مما يعني أن العرض قد تجاوز الحد الأدنى، فهو إما مقبول أو زائد، و الطول قد تجاوز الحد الأعلى فهو زائد.

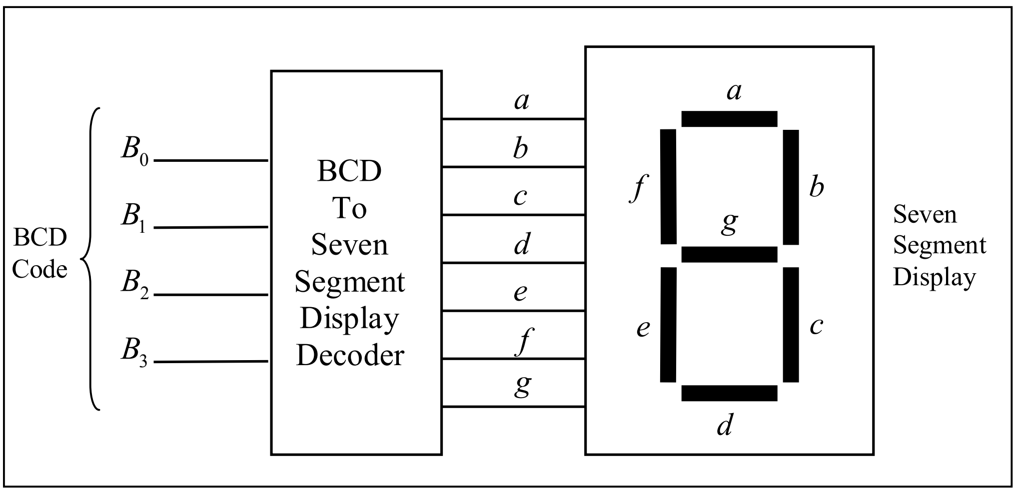
الدائرة المنطقية:



مثال (3):

BCD to Seven Segment Display Decoder

صمم دائرة BCD to Seven Segment Display Decoder للمخطط المنطقي لها أدناه.



دخل الدائرة عبارة عن رقم من الأرقام 0–9 ممثل في صورة شفرة BCD، و خرجها عبارة عن الإشارات التي تتحكم في إضاءة القطع (Segments) السبعة لعرض الرقم

المدخل على الـــ Seven Segment Display. أي قطعة من القطع السبعة عبـــارة عـــن ديود باعث للضوء (LED) يضيء عند وضع القيمة 1 في طرف الدخل الخاص به و لا يضيء عند وضع القيمة 0 في ذلك الطرف. فمثلاً لعرض الرقم 3 يجب أن نضـــع 1 في القطع a b c d و g و نضع 0 في القطع e و f، و لعرض الرقم 0 نضـــع 1 في جميع القطع عدا القطعة g، و لعرض الرقم 8 نضع 1 في جميع القطع الســـبعة، ... و هكذا.

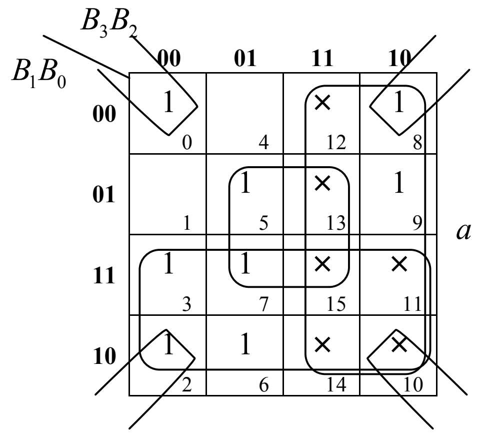
#	B_3	B_2	B_1	B_0	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	×	×	×	×	×	×	×
11	1	0	1	1	×	×	×	×	×	×	×
12	1	1	0	0	×	×	×	×	×	×	×
13	1	1	0	1	×	×	×	×	×	×	×
14	1	1	1	0	×	×	×	×	×	×	×
15	1	1	1	1	×	×	×	×	×	×	×

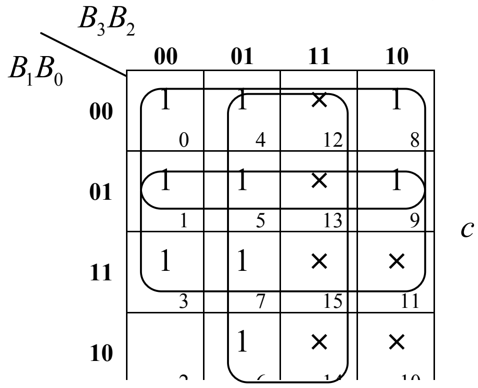
التعبيرات المنطقية:

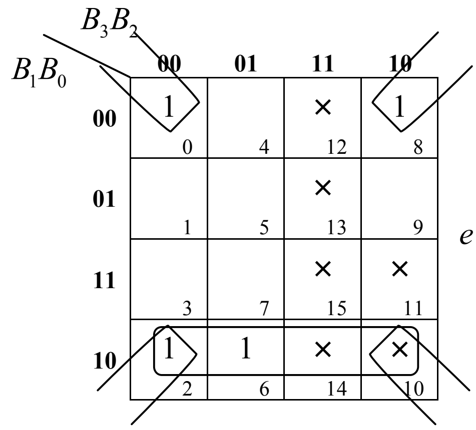
$$a = \sum m\,(0,2,3,5,6,7,8,9) + \sum d\,(10,11,12,13,14,15)$$
$$b = \sum m\,(0,1,2,3,4,7,8,9) + \sum d\,(10,11,12,13,14,15)$$
$$c = \sum m\,(0,1,3,4,5,6,7,8,9) + \sum d\,(10,11,12,13,14,15)$$
$$d = \sum m\,(0,2,3,5,6,8,9) + \sum d\,(10,11,12,13,14,15)$$
$$e = \sum m\,(0,2,6,8) + \sum d\,(10,11,12,13,14,15)$$
$$f = \sum m\,(0,4,5,6,8,9) + \sum d\,(10,11,12,13,14,15)$$
$$g = \sum m\,(2,3,4,5,6,8,9) + \sum d\,(10,11,12,13,14,15)$$

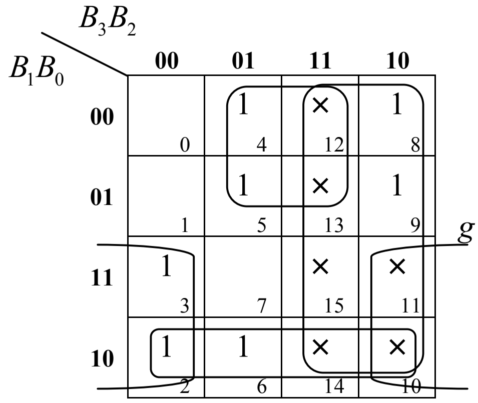
التعبيرات المنطقية المختصرة:

سنقوم هنا باختصار التعبيرات المنطقية لمتغيرات الخرج a c e و g، و نترك بقيــة المتغيرات للدارس كتدريب.









$$a = B_3 + B_1 + B_2B_0 + \overline{B_2}\,\overline{B_0}$$
$$c = \overline{B_1} + B_0 + B_2$$
$$e = \overline{B_2}\,\overline{B_0} + B_1\overline{B_0}$$
$$g = B_3 + B_2\overline{B_1} + B_1\overline{B_0} + \overline{B_2}B_1$$

الدائرة المنطقية:

نترك رسم الدائرة المنطقية للدارس كتدريب، بعد كتابة التعبيرات المختصرة لبقية متغيرات الخرج.

الخلاصة

تعلمنا في هذه الوحدة الخطوات المتبعة في تصميم أي دائرة منطقية، ابتداء مـــن تحديـــد المواصفات ويتم ذلك بإعطاء المخطط المنطقي الذي يوضح متغيرات الـــدخل والخـــرج للدائرة ،وجدول الصواب الذي يوضح العلاقة التفصيلية مابين الـــدخل والخـــرج. فكتابة التعبيرات المنطقية بحيث يعطي التعبير نفس قيم الخرج المطلوبة والموضحة فـــي جدول الصواب، وتوجد أربعة صور للتعبير المنطقي هي:

صورة مجموع الحدود الصغرى.

صورة مضروب الحدود الكبرى.

صورة AND-OR-Invert.

صورة OR-AND-Invert.

في محور آخر انتقلنا إلى تبسيط تلك التعبيرات باستخدام نظريات الجبر البـــولي حيـــث يتم التبسيط بالبحث عن التشابهات ما بين الحدود ، التبسيط باستخدام مخططات كارنوف والهدف من استخدامه هو تسهيل عملية اكتشاف التشابهات مابين الحدود وجمع الحـــدود المتشابهة ، في بعض الدوائر المنطقية تكون قيم الخرج المقابلة لبعض احتمالات الـــدخل غير محددة، أي غير معلوم ما إذا كانت مساوية 1 أو 0، و تسمى بـــ القيم غير المحددة (Don't Cares)، ويرمز لها في جدول الصواب و في مخططات كارنوف بالرمز ×. ثم انتقلنا الى بناء الدائرة إما باستخدام البوابات الأساســـية الـــثلاث (AND و OR و NOT) حيث يعتمد شكل الدائرة المنطقية على الصورة التي كتبت بها التعبيرات المنطقية، أمـــا باستخدام نوع واحد فقط من البوابات (إما NAND أوNOR) وذلك عندما يـــتم تصـــنيع الدائرة المنطقية في شكل دائرة متكاملة (Integrated Circuit) أو IC فإنه عادة مـــا يـــتم بناء الدائرة المنطقية بالكامل باستخدام نوع واحد فقط من البوابات .

كما قمنا في نهاية الوحدة بتصميم بعض الدوائر المنطقية التي تقوم بأداء وظائف مفيـــدة.

و تصميم الدوائر المنطقية هي المهارة الأساسية التي يتوقع من دارس هذا المقـــرر تعلمها.

لمحة مسبقة عن الوحدة التالية

عملياً ليس من الضروري أن نستخدم المهارات التي تعلمناها في هذه الوحدة لتصميم أي دائرة منطقية نحتاج إليها في نظام رقمي (Digital System) معين، حيث إن هناك عـــدداً كبيراً من الدوائر المنطقية الجاهزة التي تؤدي وظائف مفيدة، و التي يمكـــن شـــراؤها و استخدامها مباشرة في بناء الأنظمة الرقمية. و سنتعرف في الوحدة التالية على عدد من تلك الدوائر الجاهزة و ندرس خصائص كل منها و استخداماتها، كمـــا ســـنتعرف علـــى طرق بديلة لتصميم الدوائر المنطقية باستخدام تلك الدوائر الجاهزة.

إجابات التدريبات

تدريب 1:

1-

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

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

2-

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

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

3-

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

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

4-

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

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

تدريب 2

1-

$$x = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C} + A\overline{B}C$$
$$y = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$
$$z = \overline{A}\,\overline{B}\,\overline{C} + \overline{A}B\overline{C} + \overline{A}BC + AB\overline{C}$$

2-

$$x = (A+\overline{B}+\overline{C})(\overline{A}+\overline{B}+C)(\overline{A}+\overline{B}+\overline{C})$$
$$y = (A+B+\overline{C})(\overline{A}+B+\overline{C})$$
$$z = (A+B+\overline{C})(\overline{A}+B+C)(\overline{A}+B+\overline{C})(\overline{A}+\overline{B}+\overline{C})$$

3-

$$x = \overline{\overline{A}BC + AB\overline{C} + ABC}$$
$$y = \overline{\overline{A}\,\overline{B}C + A\overline{B}C}$$
$$z = \overline{\overline{A}\,\overline{B}C + A\overline{B}\,\overline{C} + A\overline{B}C + ABC}$$

4-

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

تدريب 3، 5

1-

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

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

2-

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

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

3-

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

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

4-

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

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

تدريب 4، 6

1-
$$x = \overline{B} + \overline{A}\,\overline{C}$$
$$y = \overline{C} + B$$
$$z = B\overline{C} + \overline{A}\,\overline{C} + \overline{A}B$$

2-
$$x = (\overline{A} + \overline{B})(\overline{B} + \overline{C})$$
$$y = B + \overline{C}$$
$$z = (\overline{A} + B)(\overline{A} + \overline{C})(B + \overline{C})$$

3-
$$x = \overline{AB + BC}$$
$$y = \overline{\overline{B}C}$$
$$z = \overline{A\overline{B} + AC + \overline{B}C}$$

4-
$$x = \overline{B(A + C)}$$
$$y = \overline{C\overline{B}}$$
$$z = \overline{(\overline{B} + C)(A + C)(A + \overline{B})}$$

تدريب 7

$$x = \overline{A_1}\,\overline{A_0} + \overline{A_3}\,\overline{A_2} + \overline{A_3}\,\overline{A_1} + \overline{A_2}\,\overline{A_0}$$
$$y = L_0 + \overline{L_3}L_2 + L_2L_1 + L_3\overline{L_2}L_1$$
$$z = \overline{C}\,\overline{B} + CBA + D\overline{C}\,\overline{A}$$

تدريب 8

$$x = (\overline{A_3} + \overline{A_0})(\overline{A_2} + \overline{A_1})$$
$$y = (\overline{L_3} + \overline{L_2} + L_1 + L_0)(L_3 + L_2 + L_0)(L_2 + \overline{L_1} + L_0)$$
$$z = (\overline{C} + B)(\overline{C} + \overline{A})(D + C + \overline{B})(C + \overline{B} + \overline{A})$$

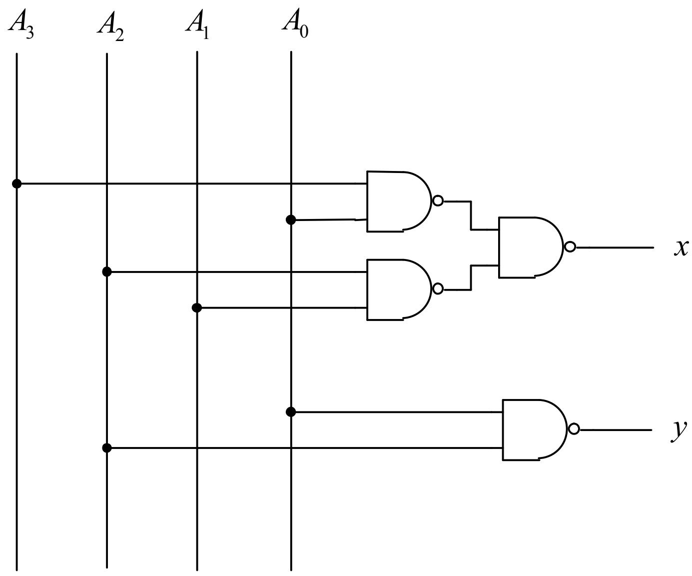
تدريب 9

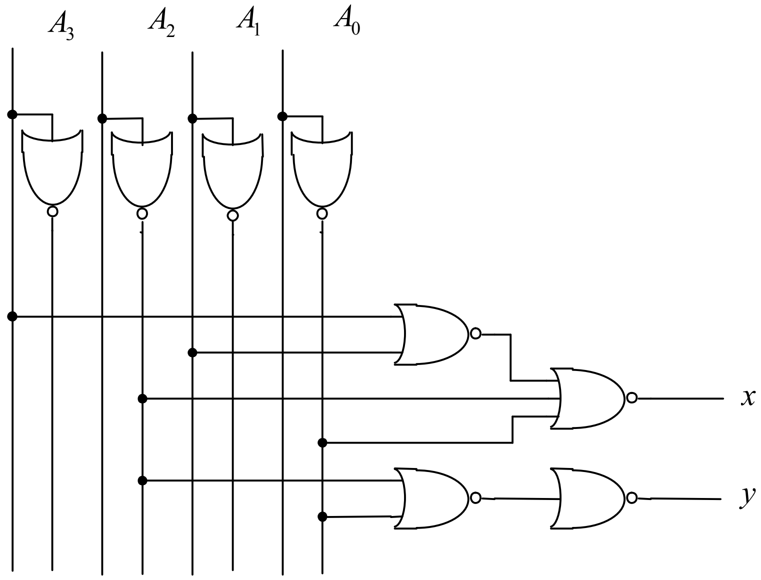
$$y = (\overline{B_2} + B_1)(B_3 + \overline{B_1} + \overline{B_0})(\overline{B_4} + \overline{B_1} + B_0)$$

تدريب 10

$$x = A_3 A_0 + A_2 A_1 + \overline{A_2}\,\overline{A_0}$$
$$y = (\overline{B_2} + B_0)(B_2 + \overline{B_0})(B_3 + B_1 + \overline{B_0})$$

تدريب 11




A_3
A_2
A_1
A_0
x
y

مسرد المصطلحات

- **صورة مجموع الحدود الصغرى (Sum of minterms)**

الحد الأصغر (minterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات AND، و قد يظهر متغير معين في الحد الأصغر معكوساً أو بدون عكس.

- **صورة مضروب الحدود الكبرى (Product of Maxterms)**

الحد الأكبر (Maxterm) هو عبارة عن حد تظهر فيه جميع متغيرات الدخل مربوطة مع بعضها البعض بعمليات OR، و قد يظهر متغير معين في الحد الأكبر معكوساً أو بدون عكس.

- **صورة AND-OR-Invert**

هذه الصورة تشبه إلى حد كبير صورة مجموع الحدود الصغرى (Sum of minterms) إلا أن التعبير المنطقي بأكمله يكون معكوساً.

- **صورة OR-AND-Invert**

هذه الصورة تشبه إلى حد كبير صورة مضروب الحدود الكبرى (Product of Maxterms) إلا أن التعبير المنطقي بأكمله يكون معكوساً.

- **التبسيط باستخدام مخططات كارنوف (Karnaugh Maps)**

مخطط كارنوف (Karnaugh Map)، أو K-Map اختصاراً، هو عبارة عن طريقة أخرى للتعبير عن المعلومات الموجودة في جدول الصواب (Truth Table)، و الهدف من استخدام المخطط هو تسهيل عملية اكتشاف التشابهات ما بين الحدود و جمع الحدود المتشابهة.


الوحدة الرابعة
الدوائر المنطقية
المترابطة

محتويات الوحدة

مقدمة

تمهيد

مرحباً بك **عزيزي الدارس،** في الوحدة الرابعة من مقرر "أساسيات التصميم المنطقي". نقوم في هذه الوحدة بعرض بعض الدوائر المنطقية الترابطية (Combinational Logic Circuits) التي تقوم بأدا وظائف مفيدة، و التي يتوفر أغلبها تجارياً بصورة جاهزة في شكل دوائر متكاملة (Integrated Circuits أو IC's)، بحيث يمكن شراؤها و استخدامها مباشرة في بناء الأنظمة الرقمية. أي أنه من الناحية العملية ليس من الضروري أن نستخدم مهارات تصميم الدوائر المنطقية التي تعلمناها في الوحدة السابقة لتصميم جميع الدوائر المنطقية التي نحتاج إليها في نظام رقمي معين، بل يمكن استخدام الدوائر الجاهزة في بناء جزء كبير من النظام الرقمي، و تصميم عدد قليل من الدوائر التي قد لا تكون متوفرة تجارياً. و سنتعرف في هذه الوحدة على عدد من تلك الدوائر الجاهزة و ندرس خصائص كل دائرة منها و استخداماتها و طرق ربطها مع بعضها البعض، كما سنتعرف على طرق بديلة لتصميم الدوائر المنطقية باستخدام تلك الدوائر الجاهزة.

أهداف الوحدة

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- **توضح** المقصود بالدوائر المنطقية الترابطية.
- **تسخدم** بوابة XOR في تصميم دوائر منطقية مفيدة.
- **تصمم** دوائر منطقية تقوم بإجراء عمليتي الجمع و الطرح.
- **تصمم** وحدة حساب (Arithmetic Unit) و تستخدمها في الأنظمة الرقمية.
- **تعرف** وظيفة كل من فاك الشفرة و المشفر و الدامج و المفرق و استخدامها في الأنظمة الرقمية.
- **تربط** الدوائر المنطقية الترابطية مع بعضها البعض لبناء دوائر أكبر.
- **تصمم** الدوائر المنطقية بالطرق البديلة التي تستخدم فيها الدوائر الجاهزة.

1. الدوائر المنطقية الترابطية (Combinational Logic Circuits)

جميع الدوائر المنطقية التي تعاملنا معها حتى الآن هي دوائر منطقية ترابطية (Combinational)، و سميت بالترابطية لأن وظيفة الدائرة تقتصر على ربط متغيرات الدخل بعمليات منطقية لتوليد متغيرات الخرج. و من الواضح أن الخرج في الدوائر الترابطية يعتمد فقط على القيم الحالية للدخل، فمتى ما تغيرت قيم الدخل تغيرت معها قيم الخرج. و النوع الآخر من الدوائر المنطقية هو الدوائر التتابعية (Sequential) التي سنقوم بدراستها في الوحدة الخامسة من هذا المقرر. أما في هذه الوحدة فسنقوم بدراسة بعض الدوائر الترابطية شائعة الاستخدام في الأنظمة الرقمية نظراً لقيامها بأداء وظائف مفيدة يتكرر ظهورها في تلك الأنظمة، و التي عادة ما تكون متوفرة تجارياً في صورة جاهزة في شكل دوائر متكاملة (IC's) بحيث يمكن شراؤها و استخدامها مباشرة في بناء الأنظمة الرقمية.

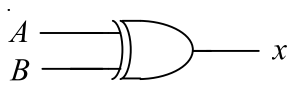
2. بعض استخدامات بوابة XOR

لبوابة XOR بعض الاستخدامات المفيدة في الدوائر المنطقية، و سنتعرض في هذا الجزء من المقررلثلاثة من أبرز تلك الاستخدامات.

في البداية نذكرك بأن عملية XOR تسمى عملية الاختلاف، حيث إن الخرج يساوي 1 إذا كان الدخلان مختلفين، و يساوي 0 إذا كانا متشابهين. ويمثل لها بالطريقة التالية:

$$x = A \oplus B$$

و في ما يلي شكل بوابة XOR و جدول الصواب لها:



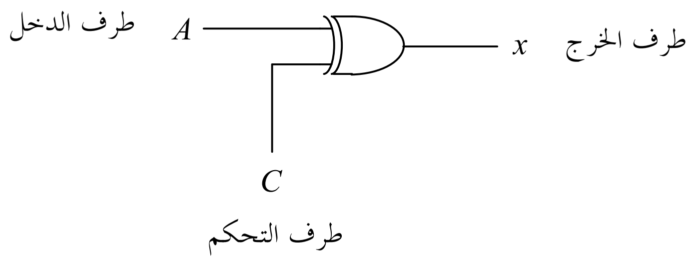
A	B	x
0	0	0
0	1	1
1	0	1
1	1	0

و كما نعلم فإنه يمكن التعبير عن عملية XOR باستخدام العمليات الأساسية الثلاث كالتالي:

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

1.2 العاكس المنطقي المحكوم (Controlled Logic Inverter)

يمكن استخدام بوابة XOR كعاكس منطقي محكوم (Controlled)، أي عاكس منطقي يمكننا أن نتحكم به بحيث يقوم بإجراء عملية العكس المنطقي للمتغير الداخل إليه أو لا يقوم بإجرائها، و ذلك حسب القيمة المنطقية التي نقوم بوضعها في طرف التحكم الخاص به. و يتم ذلك كالتالي:



حيث قمنا باستخدام طرف الدخل الثاني لبوابة XOR كطرف تحكم (Control Line).
و يمكن فهم طريقة عمل العاكس المنطقي المحكوم من جدول الصواب التالي

C	A	x
0	0	0
0	1	1
1	0	1
1	1	0

نلاحظ أنه في النصف الأعلى من جدول الصواب، عندما يكون طرف التحكم C مساوياً 0، يكون الخرج x مساوياً للدخل A، أي أن القيمة الموضوعة في طرف الدخل تمر إلى الخرج كما هي. و في النصف الأسفل من جدول الصواب، عندما يكون طرف التحكم C مساوياً 1، يكون الخرج x مساوياً لمعكوس الدخل A، أي أن القيمة الموضوعة في طرف الدخل يتم عكسها.

و باستخدام الجبر البولي

$$x = C \oplus A$$
$$x = \overline{C}\,A + C\,\overline{A}$$

فعندما يكون $C = 0$ فإن

$$x = 1 \cdot A + 0 \cdot \overline{A} = A$$

و عندما يكون $C = 1$ فإن

$$x = 0 \cdot A + 1 \cdot \overline{A} = \overline{A}$$

أي أن العاكس المنطقي المحكوم لا يقوم بإجراء عملية العكس و يمرر القيمة الموضوعة في طرف الدخل كما هي إلى الخرج عندما نضع القيمة 0 في طرف التحكم الخاص به، و يقوم بعكس القيمة الموضوعة في طرف الدخل عندما نضع القيمة 1 في طرف التحكم الخاص به. و يمكن تلخيص ذلك في جدول الصواب التالي:

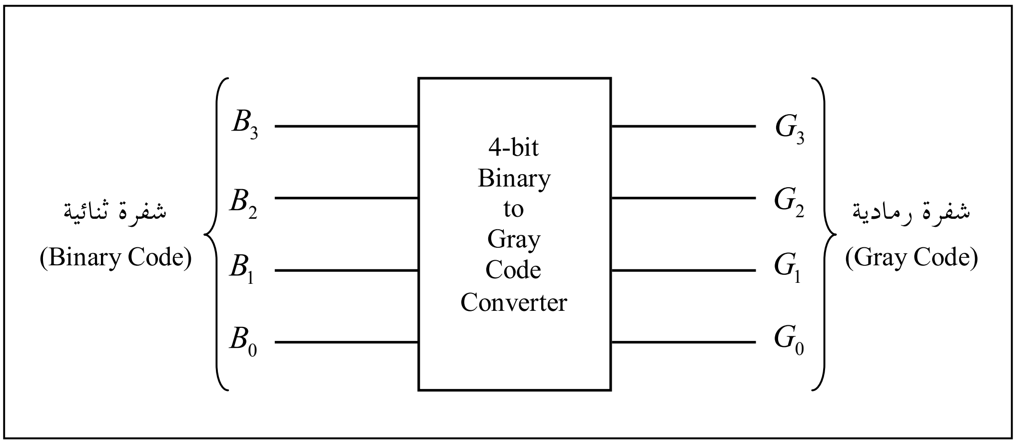
C	x
0	A
1	$\overline{A}$

أي أن

$$x = \begin{cases} A & , C = 0 \\ \overline{A} & , C = 1 \end{cases}$$

2.2 التحويل من الشفرة الثنائية إلى الشفرة الرمادية
(Binary-to-Gray Code Conversion)

سبق و أن قمنا في الوحدة السابقة بتصميم دائرة تقوم بتحويل شفرة ثنائية مكونة من أربعة خانات إلى الشفرة الرمادية (4-bit Binary-to-Gray Code Converter)، والتي يوضحها المخطط المنطقي لها أدناه:



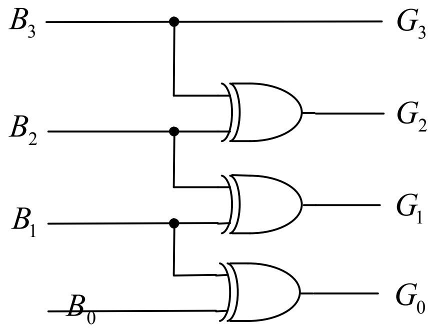
و حصلنا على التعبيرات المنطقية المختصرة التالية:

$$G_3 = B_3$$
$$G_2 = \overline{B_3}B_2 + B_3\overline{B_2}$$
$$G_1 = \overline{B_2}B_1 + B_2\overline{B_1}$$
$$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$$

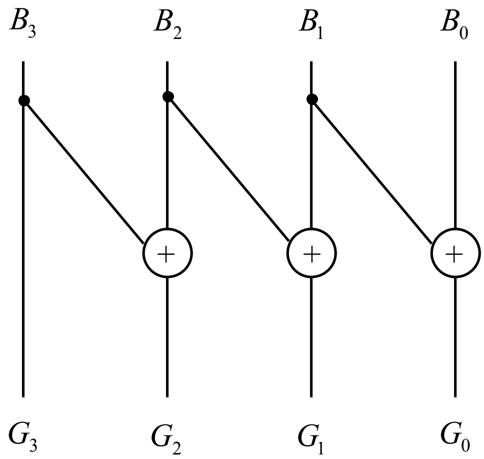
من الواضح أنه يمكن كتابة التعبيرات المنطقية أعلاه بدلالة عملية XOR كالتالي:

$$G_3 = B_3$$
$$G_2 = B_3 \oplus B_2$$
$$G_1 = B_2 \oplus B_1$$
$$G_0 = B_1 \oplus B_0$$

و عليه يمكن بناء الدائرة بالكامل باستخدام بوابات XOR كالتالي:



الدائرة أعلاه توحي لنا بطريقة سهلة و مباشرة للتحويل من الشفرة الثنائية إلى الشفرة الرمادية، كم هو مبين أدناه



حيث نبدأ بالخانة العليا في الشفرة الرمادية G_3 و هي نفسها الخانة العليا في الشفرة الثنائية B_3، ثم ننتقل للخانة التالية في الشفرة الرمادية G_2 و نحصل عليها عن طريق جمع الخانة المقابلة لها في الشفرة الثنائية B_2 جمعاً ثنائياً مع الخانة الأعلى منها B_3، و الخانة التالية في الشفرة الرمادية G_1 نحصل عليها عن طريق جمع الخانة المقابلة لها في الشفرة الثنائية B_1 جمعاً ثنائياً مع الخانة الأعلى منها B_2، و الخانة G_0 نحصل عليها بجمع B_0 مع B_1. أي أن كل خانة من خانات الشفرة الرمادية نحصل عليها عن

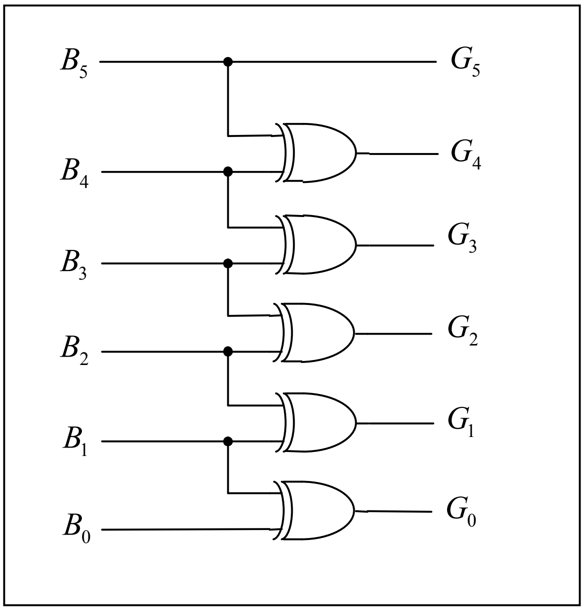
طريق جمع الخانة الثنائية المقابلة لها جمعاً ثنائياً مع الخانة الثنائية الأعلى منها، ما عــدا الخانة العليا في الشفرة الرمادية و التي نأخذها مباشرة كالخانة العليا في الشفرة الثنائية.

ملاحظة

لاحظ أن عملية الجمع الثنائي التي استخدمناها أعلاه في التحويل من الشفرة الثنائية إلى الشفرة الرمادية هي نفسها عملية XOR.

.

و بناء على ما سبق يمكننا تعديل الدائرة المبنية من بوابات XOR أعلاه، و التي تقوم بتحويل شفرة ثنائية مكونة من أربعة خانات إلى الشفرة الرمادية، بحيث تقوم بتحويل شفرة ثنائية مكونة من أي عدد من الخانات إلى الشفرة الرمادية، مثلاً:



الدائرة أعلاه تقوم بتحويل شفرة ثنائية مكونة من ستة خانات إلى الشفرة الرمادية.

3.2 فحص المثيلين (Parity Checking)

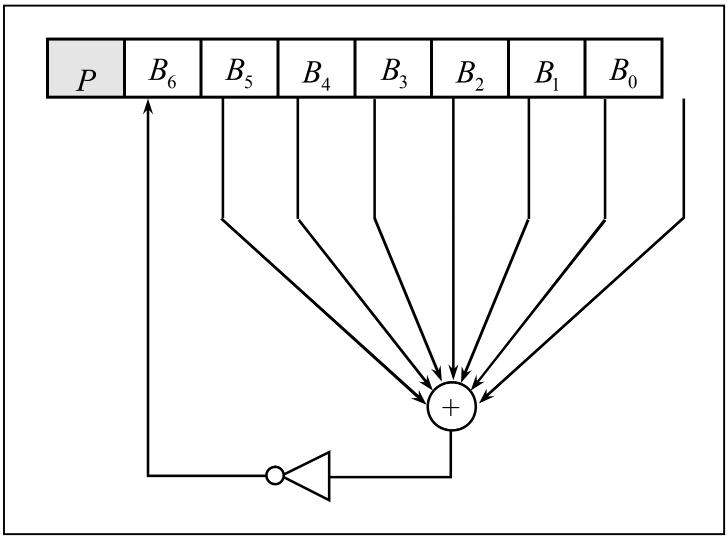
عند نقل البيانات رقمياً عبر وسائل الاتصال المختلفة في شــكل سلســلة مــن الرمــوز (Characters) الممثلة في الصورة الثنائية (Binary) قد تتعرض تلك البيانــات لحــدوث أخطاء (Errors). و يتمثل الخطأ هنا في تغير قيمة رقم ثنائي (bit) أو أكثر فــي أحــد الرموز المُرسلة من 0 إلى 1 أو من 1 إلى 0. و فحص المثيل (Parity Checking) هي عملية تستخدم لاكتشاف حدوث خطأ في البيانات المنقولة، حيث تضاف لكل رمز خانــة تسمى خانة المثيل (Parity bit)، و يتفق كل من الطرف المُرسِــل للبيانــات و الطــرف المُستقبِل لها على أن يكون العدد الكلي للـــ 1‘s في أي رمز مُرسَل فردياً مثلاً، و هــذا ما يسمى فحص المثيل الفردي(Odd Parity Checking). و بناءً على ذلك يقوم الطــرف المُرسِل قبل إرسال أي رمز بحساب عدد الـــ 1‘s الموجودة فيه، فإذا وجــد أن عــددها فردي يقوم بوضع 0 في خانة التحقق، وذلك للحفاظ على العدد الكلي للـــ 1‘s في الرمز فردياً. أما إذا وجد أن عدد الـــ 1‘s في الرمز المُرسَل زوجي فإنه يقوم بوضع 1 فـــي خانة التحقق، بحيث يصبح العدد الكلي للـــ 1‘s في الرمز فردياً. أي أن مهمة الطــرف المُرسِل هي التأكد من أن عدد الـــ 1’s فردي في كل رمز يقوم بإرساله، و ذلك بوضـــع القيمة المناسبة في خانة التحقق. أما بالنسبة للطرف المُستقبِل فإنه يقوم بحســاب عــدد الـــ 1‘s فى أي رمز يصل إليه، فإذا وجد أن عددها فردي كان معنى ذلك عدم حــدوث خطأ أثناء عملية النقل، أما إذا وجد أن عددها زوجي فمعنى ذلــك حــدوث خطــأ. و الطريقة الوحيدة الممكنة لتصحيح الخطأ الذي حدث هنا هي أن يطلب الطرف المُســتقبِل من الطرف المُرسِل إعادة إرسال الرمز الذي وصله خاطئاً، وهذا يتطلب بالطبع وجــود إمكانية الاتصال في الاتجاهين، و هو أمر غير متاح في كثير من الأحيان. لاحــظ أن هذا الأسلوب في اكتشاف حدوث الأخطاء يعجز عن اكتشاف حدوث خطأ فى خانتين فى وقت واحد، و لا توجد مشكلة هنا حيث أنه في أي نظام رقمي مصمم بصــورة جيــدة يكون احتمال حدوث خطأ في خانتين في وقت واحد أمراً نادر الحدوث بحيــث يمكــن تجاهله. يمكن أيضاً أن يتفق الطرفان المُرسِل والمُستقبِل على أن يكون العدد الكلي للـــ

1‘s في أي رمز مُرسَل زوجياً، و يسمى هذا بالتحقق الزوجي (Even Parity Checking).

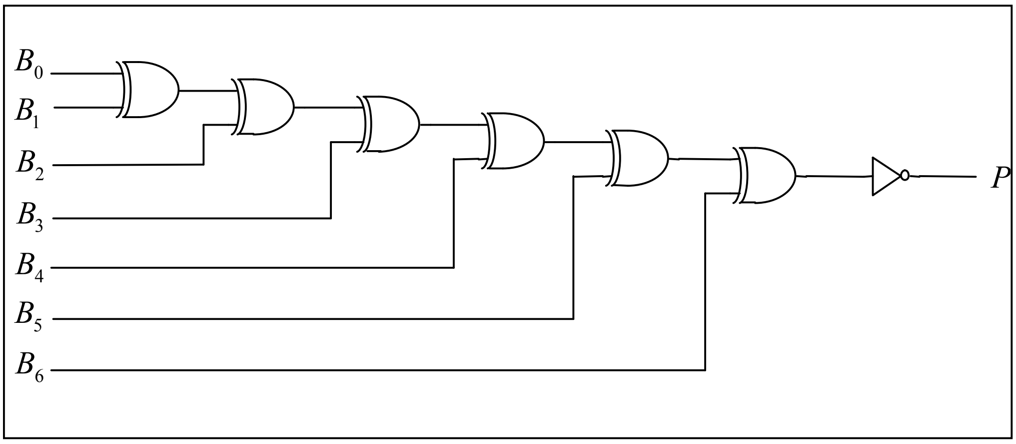
سنقوم الآن بتصميم الدوائر المنطقية المستخدمة في حالة فحص المثيل الزوجي (Odd Parity Checking). حيث سنصمم الدائرة المنطقية التي يستخدمها الطرف المُرسِل في توليد القيمة التي توضع في خانة التحقق، و سنطلق عليها اسم مولد خانة التحقق الفردي (Odd Parity bit Generator)، و الدائرة التي يستخدمها الطرف المُستقبِل لتحديد ما إذا كان هناك خطأ في الرمز الواصل إليه أم لا، و سنطلق عليها اسم فاحص المثيل الفردي (Odd Parity Checker). و سنفترض أن الرموز ممثلة باستخدام شفرة ASCII، أي أن الرمز مكون من سبعة خانات و خانة التحقق هي الخانة الثامنة.

دائرة توليد خانة المثيل الفردي (Odd Parity bit Generator)

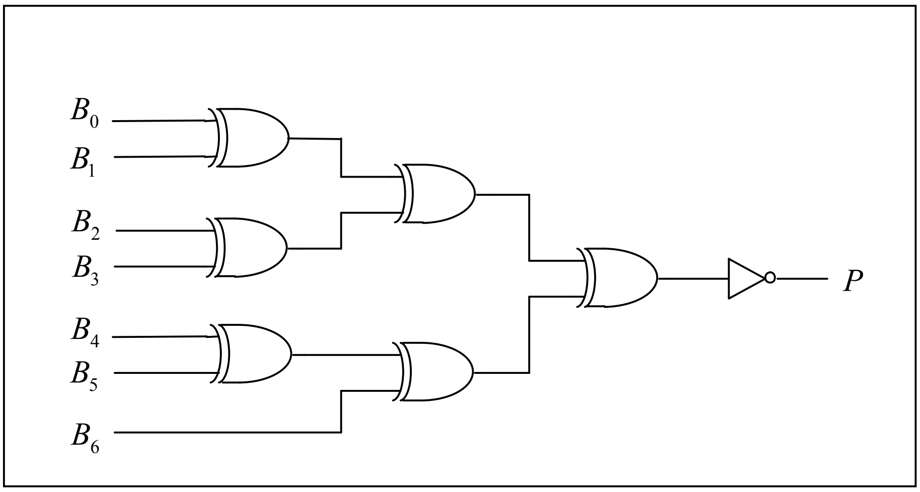
مهمة هذه الدائرة حساب عدد الـ 1’s في الخانات السبعة للرمز المُرسَل لتحديد ما إذا كان عددها فردياً أم لا، و تحديد القيمة المناسبة التي يجب وضعها في خانة التحقق بناءً على ذلك. يتم ذلك بجمع الخانات السبعة جمعاً ثنائياً، أي إجراء عملية XOR في ما بينها، فإذا كان المجموع مساوياً 0 فمعنى ذلك أن عدد الـ 1’s في الخانات السبعة زوجي و نحتاج لوضع 1 في خانة التحقق، و إذا كان المجموع مساوياً 1 فمعنى ذلك أن عدد الـ 1’s في الخانات السبعة فردي و نحتاج لوضع 0 في خانة التحقق. أي أن القيمة التي توضع في خانة التحقق هي معكوس حاصل الجمع، كما هو موضح بالمخطط المنطقي التالي:



نظراً لعدم توفر بوابة XOR بسبعة مداخل نستخدم في بناء الدائرة عدداً من بوابات XOR ذات المدخلين، كما هو موضح أدناه

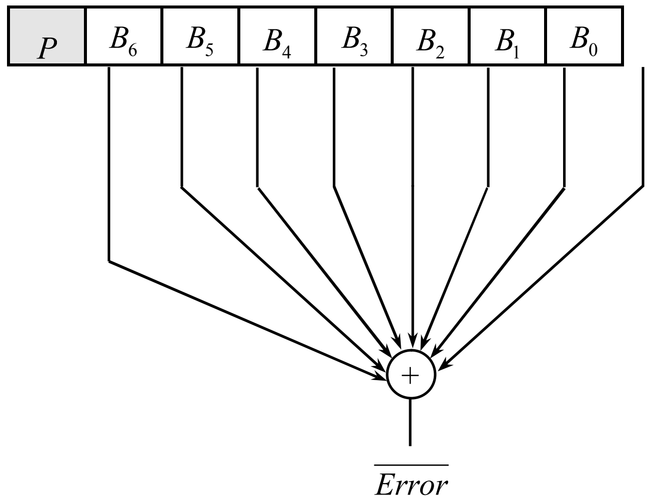


أو



فاحص المثيل الفردي (Odd Parity Checker)

مهمة هذه الدائرة حساب عدد الــ 1's في خانات الرمز الذي تم استقباله، بما فيها خانة التحقق، لتحديد ما إذا كان عددها فردياً أو زوجياً، و تحديد ما إذا كان هنالك خطأ في الرمز أم لا بناء على ذلك. يتم ذلك بجمع الخانات الثمانية جمعاً ثنائياً، أي إجراء عملية XOR في ما بينها، فإذا كان المجموع مساوياً 0 فمعنى ذلك أن عدد الــ 1's زوجي و هنالك خطأ في الرمز، أما إذا كان المجموع مساوياً 1 فمعنى ذلك أن عدد الــ 1's فردي و لا يوجد خطأ في الرمز. و يمكن توضيح ذلك بالمخطط المنطقي التالي:



لاحظ أننا استخدمنا الرمز $\overline{Error}$ لخرج الدائرة، فإذا كان $\overline{Error}=0$ فإن $Error=1$ و هنالك خطأ في الرمز، أما إذا كان $\overline{Error}=1$ فإن $Error=0$ و لا يوجد خطأ.
و سنترك لك، عزيزي الدارس، بناء الدائرة باستخدام بوابات XOR كتدريب.

تدريب 1

قم بتصميم كل من دائرة توليد خانة المثيل الزوجي (Even Parity bit Generator) وفاحص المثيل الزوجي (Even Parity Checker)، ثم قم ببنائهما باستخدام بوابات XOR.

أسئلة تقويم ذاتي

1- كيف نعبر عن عملية XOR باستخدام العمليات الأساسية؟

2- مستخدماً جدول الصواب بين عمل العاكس المنطقي المحكوم.

3- اكتب التعبيرات المنطقية المختصرة التالية بدلالة عملية XOR

$$G_3 = B_3$$

$$G_2 = \overline{B_3}B_2 + B_3\overline{B_2}$$

$$G_1 = \overline{B_2}B_1 + B_2\overline{B_1}$$

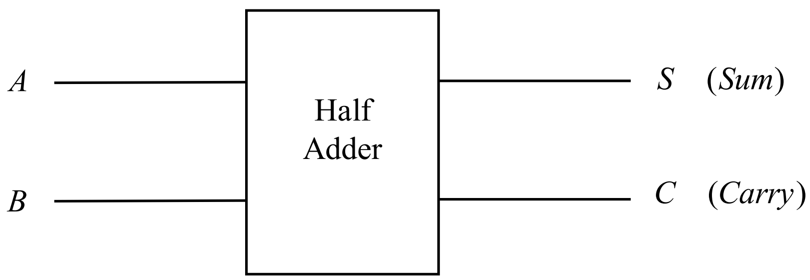
$$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$$

3 . دوائر الجمع (Adders)

دوائر الجمع أو الجوامع (Adders) هي دوائر منطقية تقوم بإجراء عملية جمع الأعـــداد الممثلة في الصورة الثنائية.

1.3 نصف الجامع (Half Adder)

نصف الجامع هو أبسط أنواع الجوامع، و هو عبارة عن دائرة منطقيـــة تقـــوم بجمـــع خانتين ثنائيتين إلى بعضهما البعض و إيجاد حاصل الجمع (Sum) و الحمـــل (Carry)، كما هو موضح بالمخطط المنطقي و جدول الصواب التاليين:



A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

و يمكن تصميم دائرة نصف الجامع بسهولة باستخدام طريقة التصميم التي درسناها في الوحدة السابقة.

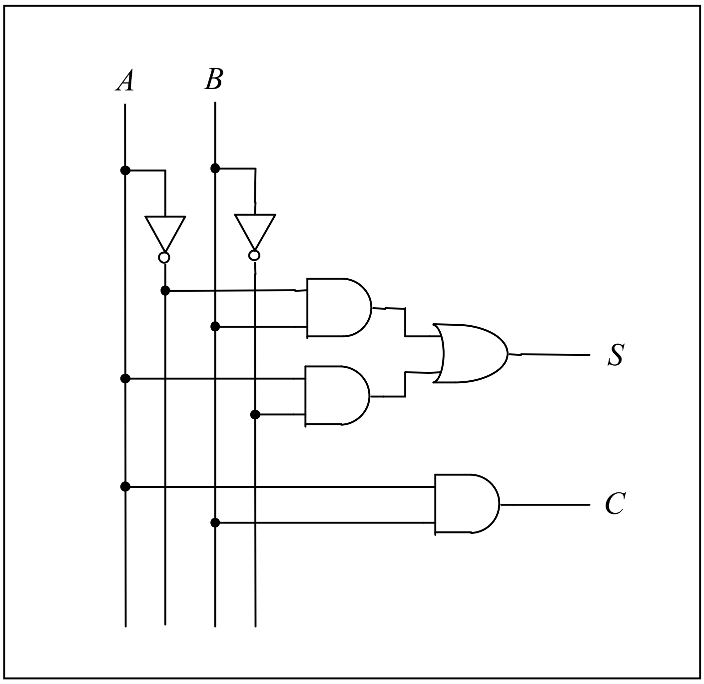
التعبيرات المنطقية:

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

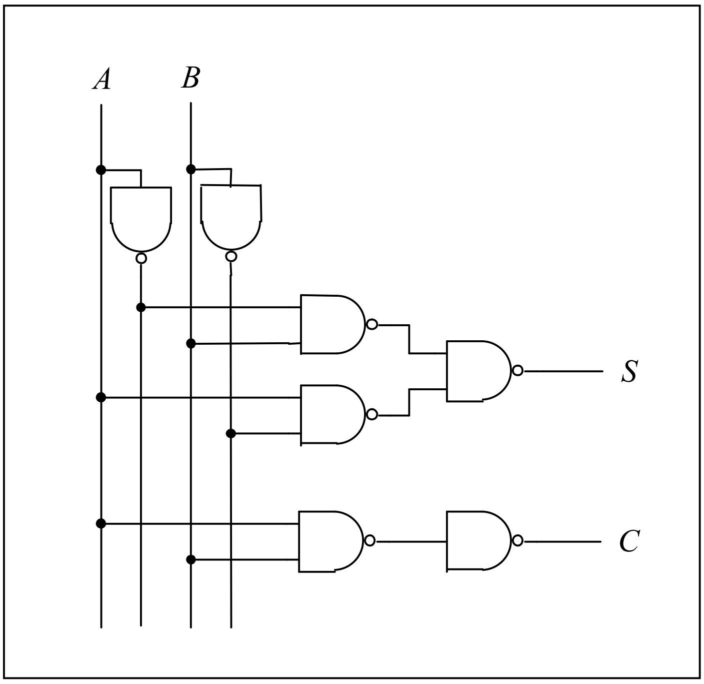
$$C = AB$$

و التعبيرات هنا في أبسط صورة و لا تحتاج إلى تبسيط

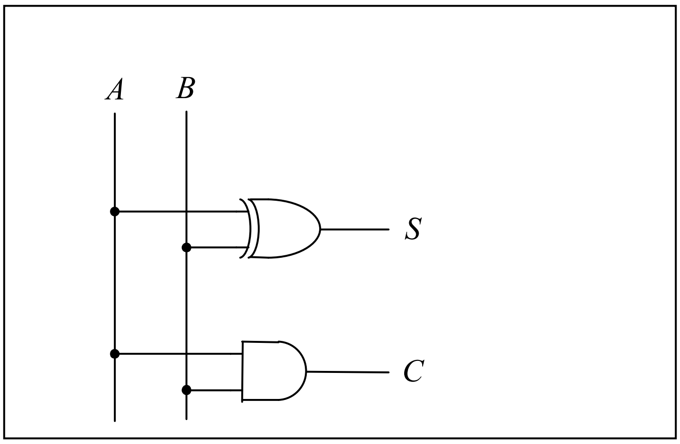
الدائرة المنطقية:



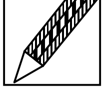
أو



أو



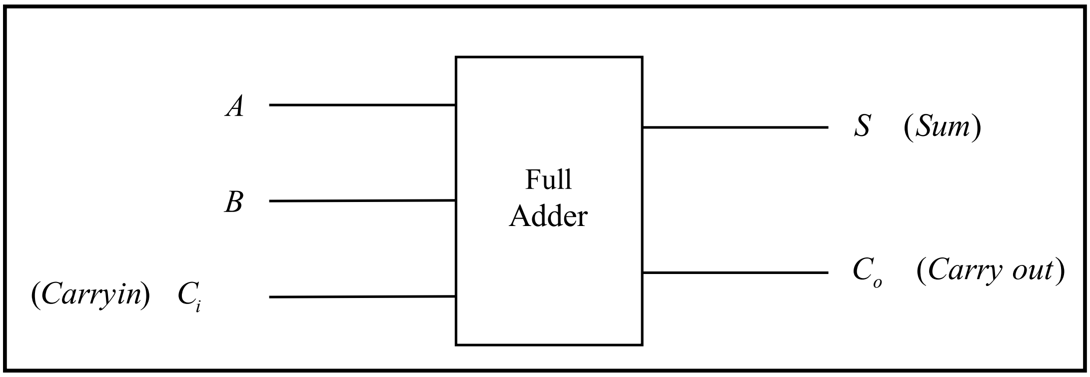
تدريب 2



استخدم بوابات NOR في بناء دائرة نصف الجامع (Half Adder).

2.3 الجامع الكامل (Full Adder)

تتشابه دائرة الجامع الكامل مع دائرة نصف الجامع في أنها تقوم بإجراء عملية الجمـــع و إيجاد كل من المجموع (Sum) و الحمل الخارج (Carry out)، إلا أن لها دخلاً ثالثاً هـــو عبارة عن حمل داخل (Carry in)، كما هو موضح بالمخطط المنطقي و جدول الصواب التاليين:



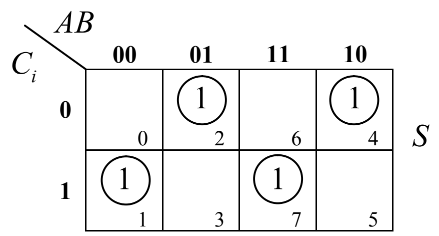
#	A	B	C_i	S	C_o
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

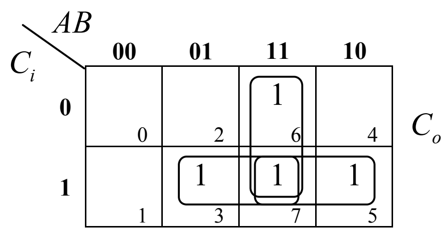
التعبيرات المنطقية:

$$S = \sum m\ (1,2,4,7)$$

$$C_o = \sum m\ (3,5,6,7)$$

التعبيرات المنطقية المختصرة:





$$S = \overline{A}\,\overline{B}C_i + \overline{A}B\overline{C_i} + ABC_i + A\overline{B}\,\overline{C_i}$$
$$C_o = AB + BC_i + AC_i$$

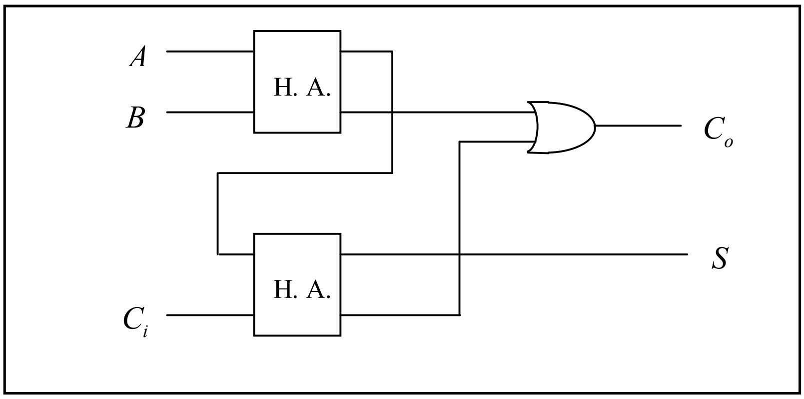
تدريب 3

قم ببناء دائرة الجامع الكامل باستخدام:

(أ) البوابات الأساسية الثلاث (AND، OR، NOT).

(ب) بوابات NAND.

بناء الجامع الكامل باستخدام دائرتي نصف جامع



لاحظ أننا قد استخدمنا هنا نصف الجامع الأول لجمع الخـــانتين A و B، ثـــم أدخلنـــا حاصل الجمع الناتج إلى نصف الجامع الثاني مع الخانـــة الثالثـــة C_i، فحصـــلنا علـــى مجموع الخانات الثلاثة. أما الحمل الخارج C_o فإنه إما أن ينتج عن عملية الجمع الأولى أو عن عملية الجمع الثانية، لذلك ربطنا الحمل الخارج من دائرتي نصف الجامع بعملية OR.

3.3 الجامع المتعدد الخانات (Multi-bits Adder)

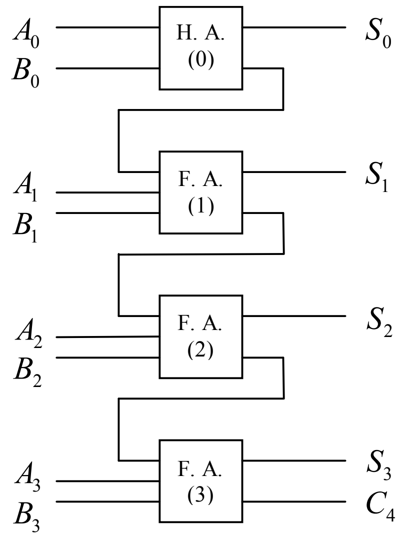
المطلوب الآن تصميم دائرة منطقية تقوم بجمع عددين ثنائيين يتكون كل منهما من أربعة خانات (4-bit Adder).

إذا أردنا إجراء عملية الجمع هذه يدوياً فإننا نبدأ بوضع العددين الثنائيين فوق بعضـــهما البعض، ثم نقوم بجمع كل خانة من العدد الأول مع الخانة المقابلة لها من العدد الثـــاني، مبتدئين بالخانة الدنيا (LSB)، مع ترحيل الحمل الخارج الناتج من خانة معينة إلى الخانة التي تليها كحمل داخل، كما هو مبين أدناه:

$$\begin{array}{ccccc} & C_3 & C_2 & C_1 & \\ & A_3 & A_2 & A_1 & A_0 \\ & B_3 & B_2 & B_1 & B_0 \\ \hline C_4 & S_3 & S_2 & S_1 & S_0 \end{array}$$

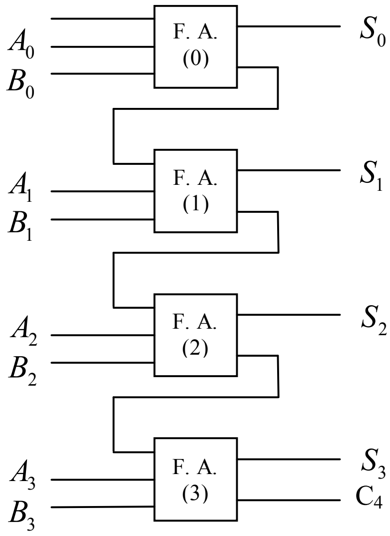
حيث جمعنا A_0 مع B_0 فحصلنا على المجموع S_0 و الحمل الخارج C_1 الـــذي قمنـــا بترحيله إلى الخانة التالية، ثم جمعنا C_1 مع A_1 و B_1 فحصلنا علـــى المجمـــوع S_1 و الحمل الخارج C_2 الذي قمنا بترحيله إلى الخانة التالية، ... و هكذا، حتى الخانة الأخيرة حيث جمعنا C_3 مع A_3 و B_3 فحصلنا على المجموع S_3 و الحمل الخارج C_4.

يمكننا استخدام نصف جامع لإجراء عملية الجمع في الخانة الدنيا (LSB)، و جامع كامل لإجراء عملية الجمع في كل خانة من الخانات التالية، مع مراعاة ترحيل الحمل من خانة إلى الخانة التي تليها، كما هو مبين أدناه:

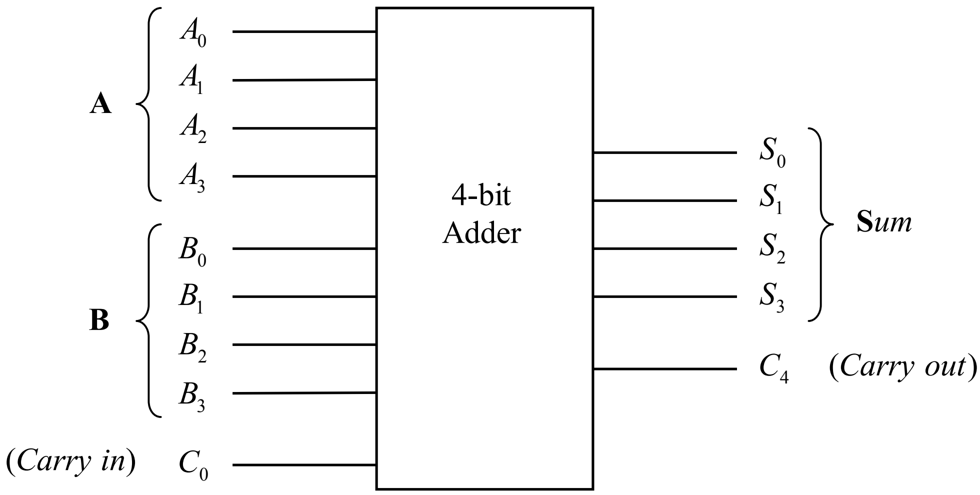


عادة ما يستخدم جامع كامل بدلاً عن نصف الجامع في الخانة الدنيا (LSB)، نجـــد أنـــه يسمح بوجود حمل داخل (Carry in) للجامع المتعدد الخانات. و يستخدم هـــذا الحمـــل الداخل فى عمليات ربط الدوائر مع بعضها البعض و في إجراء عمليـــة الطـــرح، كمـــا سيتضح لاحقاً. و بذلك يكون الشكل النهائي لدائرة الجامع ذي الخانات الأربعة هو:

$(Carry\ in)\quad C_0$



الشكل التالي يمثل المخطط المنطقي للجامع ذي الخانات الأربعة



المخطط المنطقي للجامع ذو الأربعة خانات

لاحظ أن دائرة الجامع ذي الخانات الأربعة (4-bit Adder) لها تسعة أطراف دخل، مما يجعل من تصميم هذه الدائرة باستخدام أسلوب التصميم الذي درسناه في الوحدة السابقة أمراً غاية في الصعوبة (لاحظ أن عدد أسطر جدول الصواب وحده هو $2^9 = 512$). تم حل هذه الإشكالية هنا بتقسيم الدائرة الكبيرة إلى عدد من الوحدات الصغيرة (جوامع كاملة) كل وحدة منها بعدد محدود من أطراف الدخل، و بعد تصميم الوحدة الصغيرة تم ربط الوحدات مع بعضها البعض بحيث تؤدي وظيفة الدائرة الكبيرة. و هذا الأسلوب في التصميم شائع الاستخدام في الأنظمة الرقمية حيث يتم تقسيم أي نظام رقمي معقد إلى عدد من الوحدات الأصغر، ثم تقسيم كل وحدة من هذه الوحدات بدورها إلى عدد من الوحدات الأصغر، ... و هكذا.

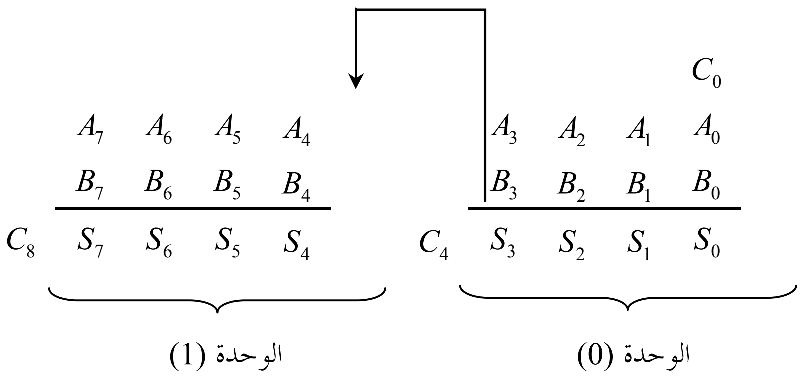
لاحظ أيضاً أنه يمكن بسهولة زيادة عدد خانات الجامع متعدد الخانات بزيادة عدد الجوامع الكاملة، بحيث نستطيع تصميم جامع بأي عدد من الخانات.

تدريب 4

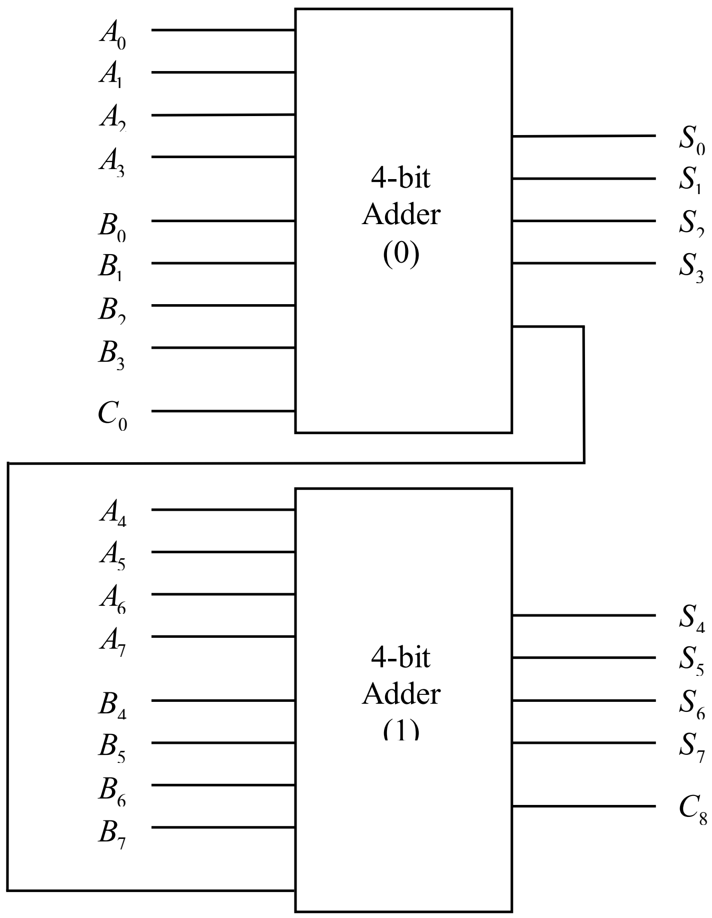
صمم جامعاً ذا ثمانية خانات (8-bit Adder).

ربط الجوامع

يمكن ربط وحدات جامع صغيرة لبناء جامع أكبر. مثلاً إذا قمنا بربط وحدتي جامع ذي خانات أربعة نحصل على جامع ذي خانات ثمانية، كما هو موضح أدناه



أي أننا يجب أن نقوم بترحيل الحمل الخارج (Carry out) من الوحدة الأولى و إدخاله كحمل داخل (Carry in) إلى الوحدة الثانية.



المخطط المنطقي للجامع ذو ثمانية خانات

تدريب 5

وضح طريقة ربط 4 وحدات جامع ذي خانات أربعة (4-bit Adders) لبناء جامع ذو 16 خانة (16-bit Adder).

4.3 عملية الطرح (Subtraction)

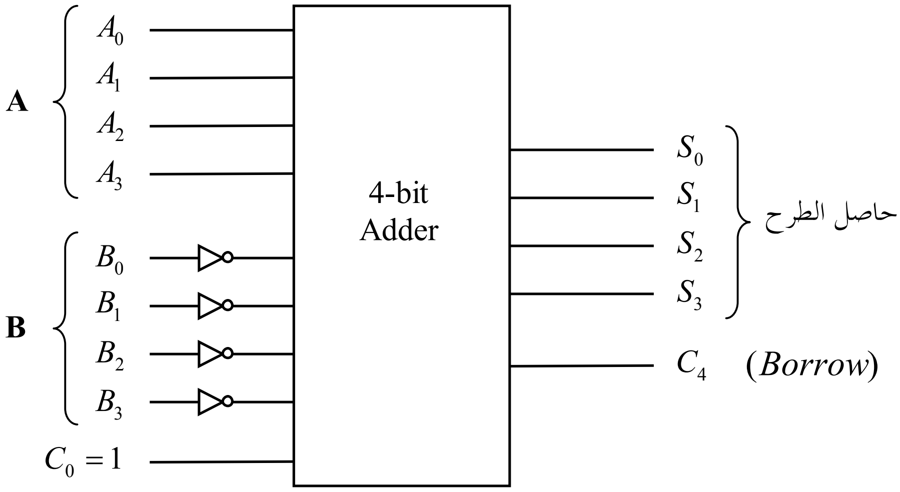
يتم تحويل عملية الطرح إلى عملية جمع مع سالب العدد المطروح كالتالي:

$$A - B = A + (-B)$$

و سالب العدد B هو المكمل الثاني (2's Complement) له، و نحصل عليه بعكس جميع خانات العدد B ثم إضافة 1 إلى الخانة الدنيا (LSB). فإذا اعتبرنا أن كلاً من A و B عبارة عن عدد ثنائي ذي خانات أربعة فإن عملية الطرح تتم كالتالي:

$$\begin{array}{ccccc}
 & & & & 1 \\
 & A_3 & A_2 & A_1 & A_0 \\
 & \overline{B_3} & \overline{B_2} & \overline{B_1} & \overline{B_0} \\
\hline
C_4 & S_3 & S_2 & S_1 & S_0
\end{array}$$

و يتم إجراء العملية باستخدام الجامع ذي خانات أربعة كما في الشكل التالي:

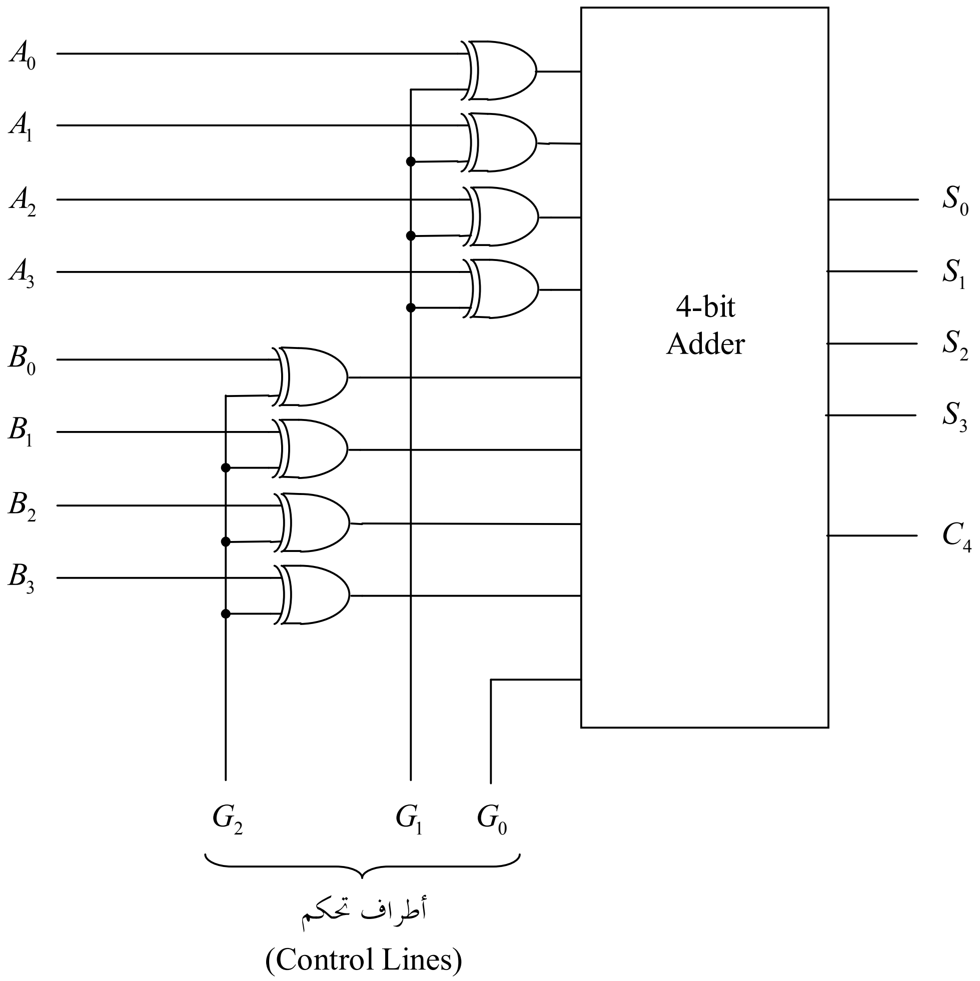


ملاحظة

لاحظ أنه إذا كان الخرج C_4 مساوياً 1 فإن ذلك يدل على حدوث استلاف (Borrow) من الخانة التي تلي الخانة العليا (MSB)، و يحدث هذا الاستلاف إذا كان العدد المطروح B أكبر من العدد المطروح منه A. أي أنه إذا كان C_4 مساوياً 1 فإن ذلك يدل على أن حاصل الطرح عبارة عن عدد سالب، أي أن C_4 يمثل إشارة حاصل الطرح.

و لكن الدائرة بشكلها هذا لا تستطيع إجراء أي عملية بخلاف العملية $A-B$، حيث لا يمكن مثلاً إجراء عمليات مثل $B-A$ أو $A+B$.

لزيادة مرونة الدائرة يمكن أن نستخدم عواكس منطقية محكومة (Controlled Logic Inverters) بدلاً عن العواكس المنطقية العادية، بحيث نستطيع أن نقوم بإجراء عملية العكس المنطقي أو عدم إجرائها، حسب الحاجة. و ذلك كالتالي



تذكر أن العاكس المنطقي المحكوم يقوم بإجراء عملية العكس إذا وضعنا القيمة 1 في طرف التحكم الخاص به، و يمرر القيمة دون عكس إذا وضعنا القيمة 0 في طرف التحكم الخاص به.

و عليه يمكن إجراء عدد من العمليات الحسابية المختلفة بوضع القيم المناسبة في أطراف التحكم G_0 G_1 G_2. مثلاً:

- لإجراء العملية $A-B$ نحتاج لعكس خانات العدد B لإيجاد المكمل الأول لـــه، لذلك نضع 1 في طرف التحكم G_2، كما نحتاج لوضع 1 في طرف التحكم G_0، الذي يمثل الحمل الداخل (Carry in) للجامع ذي الخانات الأربعة ، و ذلك لإيجاد المكمل الثاني للـــ B. أما العدد A فلا نحتاج لعكس خاناته بل نريدها أن تمـــر كما هي لذلك نضع 0 في طرف التحكم G_1.
- لإجراء العملية $A+B$ لا نحتاج لعكس خانات أي من العددين A و B لذلك نضع 0 في طرفي التحكم G_1 و G_2. كما لا نحتاج لوضع 1 في الطرف G_0 لذلك نضع فيه 0 أيضاً.
- لإجراء العملية $B-A$ نحتاج لعكس خانات العدد A بوضع 1 في طرف التحكم G_1، كما نحتاج لوضع 1 في طرف التحكم G_0 لإيجاد المكمل الثاني للـــ A.
أما العدد B فنريد أن تمر خاناته كما هي لذلك نضع 0 في طرف التحكم G_2.

و يمكن تلخيص ذلك في الجدول التالي:

العملية (Operation)	إشارات التحكم		
	G_2	G_1	G_0
$A-B$	1	0	1
$A+B$	0	0	0
$B-A$	0	1	1

ملاحظة

لاحظ أن عدد أطراف التحكم هو 3 و معنى ذلك أنه يوجد $2^3=8$ احتمالات مختلفة للقيم التي يمكن وضعها على هذه الأطـــراف. و قد أخذنا في الاعتبار ثلاثة فقط من هذه الاحتمالات فـــي الجـــدول أعلاه. المطلوب الآن إنشاء جدول صواب يحتوي علـــى جميـــع احتمالات الدخل لأطراف التحكم و العملية التي تقابل كل احتمـــال منها.

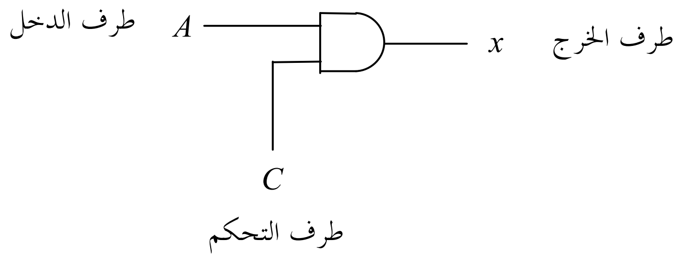
G_2	G_1	G_0	Operation
0	0	0	$A+B$
0	0	1	$A+B+1$
0	1	0	$B-A-1$
0	1	1	$B-A$
1	0	0	$A-B-1$
1	0	1	$A-B$
1	1	0	$-A-B-2$
1	1	1	$-A-B-1$

يمكننا الآن القول أن الدائرة السابقة عبارة عن دائرة جامع/طارح ذي خانات أربعة (4-bit Adder/Subtracter).

5.3 وحدة الحساب (Arithmetic Unit)

يمكننا إجراء المزيد من العمليات الحسابية المفيدة باستخدام دائرة الجامع/الطارح ذي الخانات الأربعة إذا أضفنا إليها إمكانية تصفير أحد العددين A أو B، أي التعويض عنه بصفر. مثلاً في العملية $A+B+1$ إذا عوضنا عن العدد B بصفر نحصل على العملية $A+1$، أي عملية Increment A، التي يرمز لها في بعض لغات البرمجة بالرمز $A++$. و في العملية $B-A-1$ إذا عوضنا عن العدد A بصفر نحصل على العملية $B-1$، أي عملية Decrement B، التي يرمز لها في بعض لغات البرمجة بالرمز $B--$. و في العملية $B-A$ إذا عوضنا عن العدد B بصفر نحصل على العملية $-A$، أي عملية Negate A.

يمكن إضافة إمكانية تصفير أحد العددين A أو B إلى الدائرة باستخدام مجموعة من بوابات AND كبوابات تحكم، كالتالي:

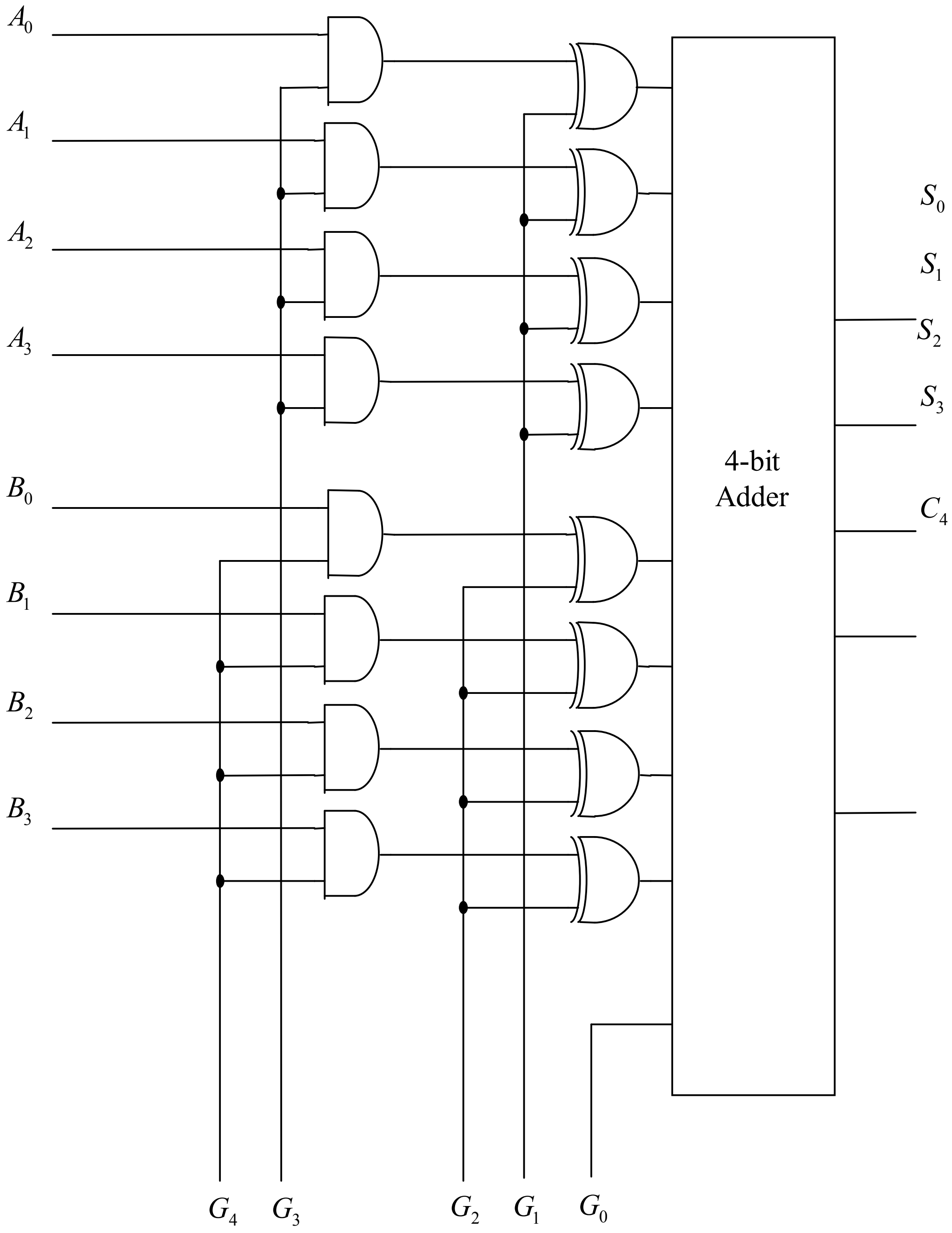


	C	A	x	
$C = 0$	0	0	0	تصفير
	0	1	0	
$C = 1$	1	0	0	تمرير
	1	1	1	

C	x
0	0
1	A

أي أنه عند وضع القيمة 1 في طرف التحكم لبوابة AND فإنها تمرر القيمة الموضـــوعة في طرف الدخل لها كما هي، و عند وضع القيمة 0 في طرف التحكم لبوابة AND فإنها تقوم بتصفير خرجها.

يتم إضافة بوابات AND لدائرة الجامع/الطارح ذي الخانات الأربعة كالتالي:



الدائرة المعدلة لاجراء عمليات

و الجدول التالي يوضح بعض العمليات التي يمكن إجراؤها باستخدام الدائرة المعدلة و إشارات التحكم اللازمة للقيام بكل عملية

العملية (Operation)	إشارات التحكم				
	تصفير		عكس		
	G_4	G_3	G_2	G_1	G_0
$A++$	0	1	0	0	1
$B--$	1	0	0	0	1
$-A$	0	1	0	1	1
$-B$	1	0	1	0	1
$\overline{A}=-A-1$	0	1	0	1	0
$\overline{B}=-B-1$	1	0	1	1	1

اجراء العمليات باستخدام الدائرة المعدلة

الدائرة المعدلة تمثل وحدة حساب (Arithmetic Unit) ذات خانات أربعة، و إذا أضيف لها الجزء الخاص بإجراء العمليات المنطقية تصبح وحدة حساب و منطق (Arithmetic Logic Unit) أو ALU.

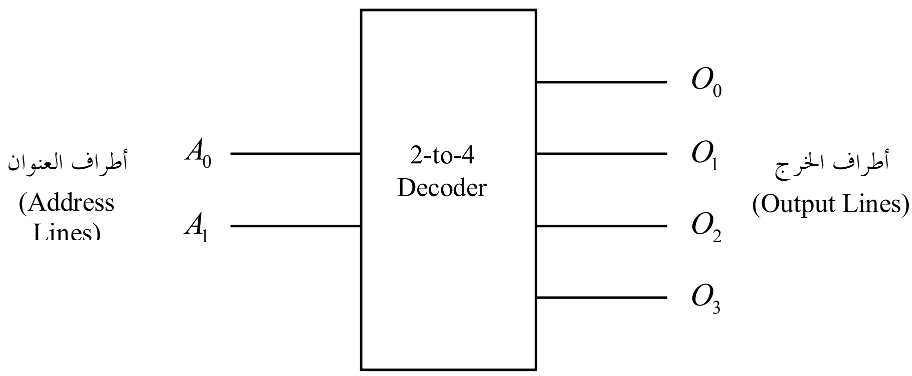
تدريب 6

وضح في شكل جدول جميع العمليات الحسابية التي يمكن إجراؤها باستخدام وحدة الحساب أعلاه، و إشارات التحكم اللازمة للقيام بكل عملية.

4 . جهاز فك الشفرة (Decoder)

جهاز فك الشفرة عبارة عن دائرة منطقية لها عـدة أطـراف خـرج (Output Lines). واحد فقط من أطراف الخرج هذه يكون نشطاً (Active) أما بقية أطراف الخرج تكـون غير نشطة. طرف الخرج النشط تظهر فيه القيمة المنطقية 1، أما بقية أطراف الخـرج (غير النشطة) فتظهر في كل منها القيمة المنطقية 0. يتم اختيار طرف الخرج النشـط بواسطة أطراف الدخل للدائرة و التي تسمى أطراف العنـوان (Address Lines)، فلكـل طرف من أطراف الخرج عنوان (Address) فريد يميزه، و هذا العنـوان عبـارة عـن شفرة ثنائية (Binary Code) عندما توضع على أطراف العنوان ينشط طـرف الخـرج المقابل لذلك العنوان.

و في ما يلي المخطط المنطقي و جدول الصواب لجهاز فك الشفرة من نوع 2 إلى 4 (2-to-4 Decoder)



المخطط المنطقي لجهاز فك الشفرة من نوع 2 الى 4

#	A_1	A_0	O_3	O_2	O_1	O_0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

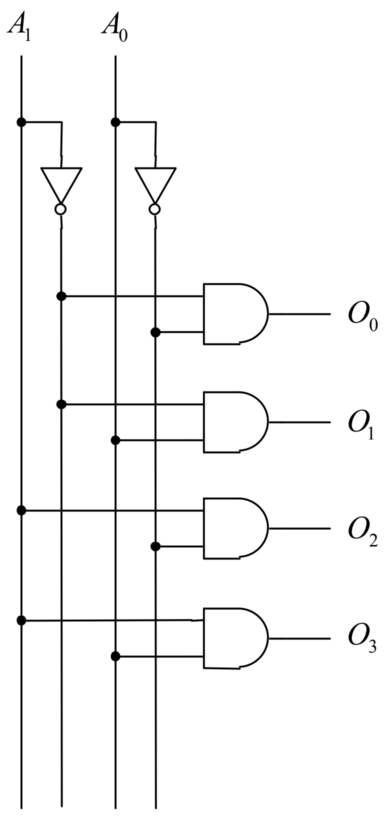
إذا أردنا تصميم دائرة جهاز فك الشفرة من نوع 2 إلى 4 التي يوضحها المخطط المنطقي و جدول الصواب لها أعلاه فما علينا إلا إتباع خطوات التصميم التي تدربنا عليها في الوحدة السابقة.

التعبيرات المنطقية:

$$O_0 = \overline{A_1}\,\overline{A_0} = m_0$$
$$O_1 = \overline{A_1}A_0 = m_1$$
$$O_2 = A_1\overline{A_0} = m_2$$
$$O_3 = A_1A_0 = m_3$$

نلاحظ أن جهاز فك الشفرة يقوم بتوليد الحدود الصغرى (minterms) لمتغيرات الدخل في أطراف الخرج له.

لدائرة المنطقية:



نلاحظ أن الدائرة المنطقية لجهاز فك الشفرة تتكون أساساً من مجموعة من بوابات AND بعدد أطراف الخرج.

__**لاحظ**__ وجود علاقة ما بين عدد أطراف الخرج (Output Lines) و عدد أطراف العنوان (Address Lines) لفاك الشفرة، فإذا كان عدد أطراف العنوان هو N فمعنى ذلك أن عدد العناوين الممكنة هو 2^N، و بالتالي فإن عدد أطراف الخرج يجب أن يكون أقل من أو مساوياً 2^N، بحيث يكون لكل طرف من أطراف الخرج عنوان فريد يميزه.

__لاحظ__ أيضاً الأسلوب المتبع في تسمية فك الشفرة، حيث نذكر في الاسم عدد أطراف الدخل ثم عدد أطراف الخرج و بينهما كلمة إلى (to).

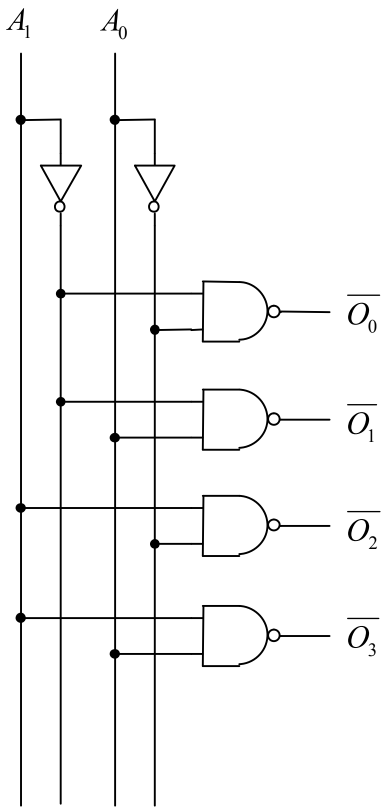
تدريب 7

وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبيرات المنطقية و ارسم الدائرة المنطقية لـــ:

(أ)جهاز فك شفرة من نوع 1 إلى 2 (1-to-2 Decoder).

(ب)جهاز فك شفرة من نوع 3 إلى 8 (3-to-8 Decoder).

أحياناً يتم استبدال بوابات AND في دائرة جهاز فك الشفرة ببوابات NAND، كما هو موضح في الشكل التالي لجهاز فك شفرة من نوع 2 إلى 4 (2-to-4 Decoder)



في هذه الحالة يكون الخرج معكوساً، و بالتالي فإن طرف الخرج النشط تظهر فيه القيمة المنطقية 0، و أطراف الخرج الأخرى (غير النشطة) تظهر في كل منها القيمة المنطقية 1. و نقول في مثل هذه الحالة إن فاك الشفرة ذو خرج نشط مــنخفض (Active Low Outputs). و كثيراً ما يستخدم مصطلحا منخفض (Low) و مرتفع (High) للإشارة إلى حالة أطراف الخرج (أو أطراف الدخل) في الدوائر المنطقية، لأنه عادة ما يــتم تمثيــل القيمة المنطقية 0 في تلك الدوائر بجهد كهربائي منخفض (مثلاً V 0)، و القيمة المنطقية 1 بجهد كهربائي مرتفع (مثلاً V 5+).

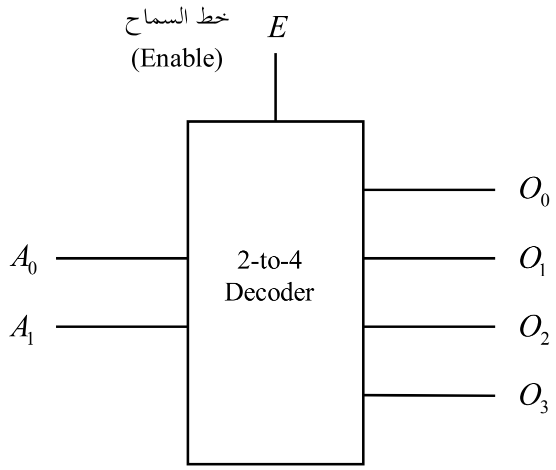
تدريب 8

وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبيرات المنطقية لجهاز فك الشفرة من نوع 2 إلى 4 بخرج نشط منخفض (2-to-4 Decoder with Active Low Outputs) الذي توضحه الدائرة المنطقية له

خط السماح (Enable)

عادة ما يكون جهاز فاك الشفرة مزوداً بخط سماح (Enable). و خط السماح، في الدوائر المنطقية بصورة عامة، هو عبارة عن طرف تحكم يمكن بواسطته أن نبطل عمل الدائرة، أو نسمح لها بالعمل كالمعتاد.

و في ما يلي المخطط المنطقي لجهاز فك الشفرة من نوع 2 إلى 4 مزود بخط سماح (2-to-4 Decoder with Enable)



عند وضع القيمة المنطقية 0 في خط السماح E يبطل عمل جهاز فك الشفرة فلا يستجيب للقيم الموضوعة في أطراف العنوان و تكون جميع أطراف الخرج له غير نشطة، أما عند وضع القيمة المنطقية 1 في خط السماح E فإن جهاز فك الشفرة يعمل كالمعتاد. و يمكن توضيح ذلك بجدول الصواب التالي:

E	A_1	A_0	O_3	O_2	O_1	O_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

و يمكن كتابة جدول الصواب بصورة مختصرة كما في الجدول التالي:

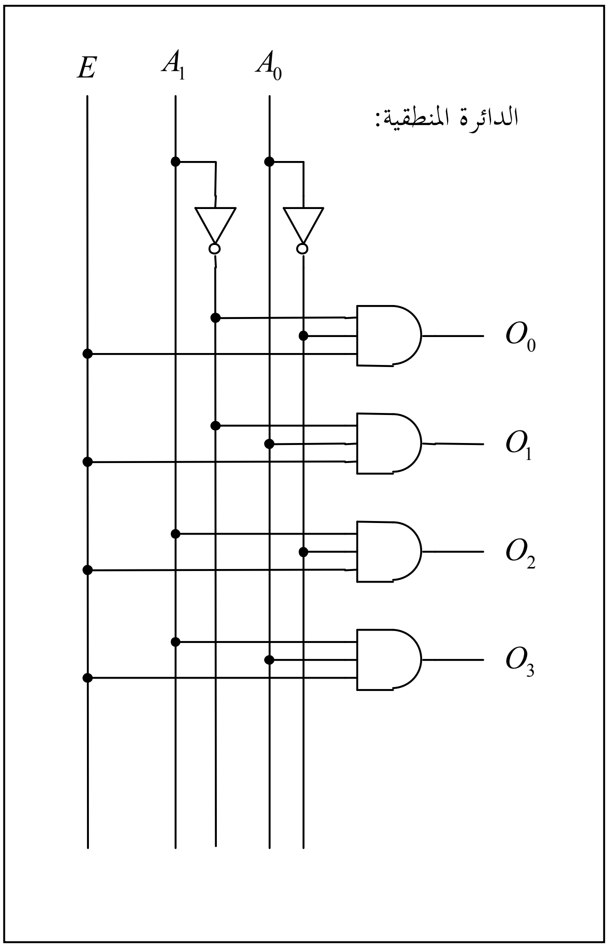
E	A_1	A_0	O_3	O_2	O_1	O_0
0	×	×	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

السطر الأول من جدول الصواب هنا يعني أنه طالما كان خط السماح $E = 0$ فإنه بغض النظر عن قيم طرفي العنوان A_0 تكون جميع أطراف الخرج لجهاز فك الشفرة غير نشطة. فاستخدامنا لرمز القيم غير المحددة × هنا مكننا من دمج أربعة أسطر من جدول الصواب في سطر واحد نظراً لتشابه قيم الخرج في هذه الأسطر الأربعة.

و يمكن بسهولة تصميم دائرة جهاز فك الشفرة من نوع 2 إلى 4 بخط سماح (2-to-4 Decoder with Enable) كالتالي

التعبيرات المنطقيةهي:

$$O_0 = E\overline{A_1}\,\overline{A_0}$$
$$O_1 = E\overline{A_1}A_0$$
$$O_2 = EA_1\overline{A_0}$$
$$O_3 = EA_1A_0$$



خط السماح (Enable) يمكن أن يكون نشطاً منخفضاً (Active Low) أيضاً، و يرمز له في هذه الحالة بالرمز $\overline{E}$، و يسمح للدائرة بالعمل عندما توضع فيه القيمة المنطقية 0، و يبطل عملها عندما توضع فيه القيمة المنطقية 1.

تدريب 9

وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبيرات المنطقية و ارسم الدائرة المنطقية لجهاز فك الشفرة من نوع:

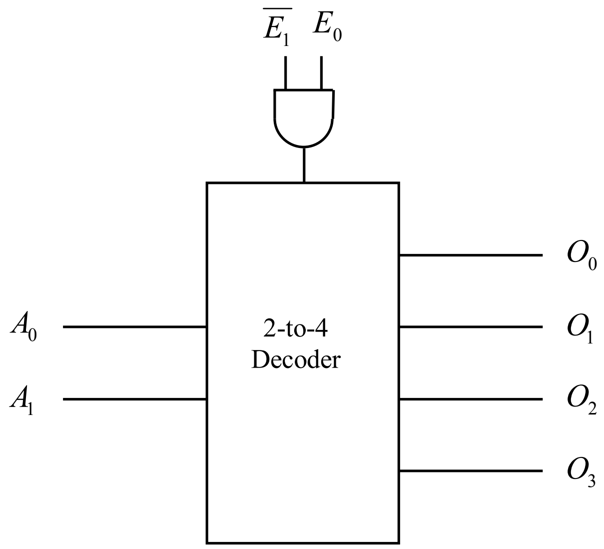
(أ) 3 إلى 8 بخط سماح (3-to-8 Decoder with Enable).

(ب) 3 إلى 8 بخط سماح و خرج نشط منخفض (3-to-8 Decoder with Enable and Active Low Outputs).

(ج) 3 إلى 8 بخط سماح نشط منخفض (3-to-8 Decoder with Active Low Enable).

خطوط التمكين المتعددة (Multiple Enables)

في بعض الأحيان قد يكون لدائرة ما أكثر من خط سماح واحد، ترتبط مع بعضها البعض بعمليات منطقية، و تعمل معاً على إبطال عمل الدائرة أو السماح لها بالعمل. مثلاً



خطا السماح E_0 و E_1 هنا مرتبطان بعملية AND، و شرط عمل الدائرة هنا هو أن يكون $\overline{E_1}E_0 = 1$، أي أن يكون $E_0 = 1$ و $E_1 = 0$.

تدريب 10

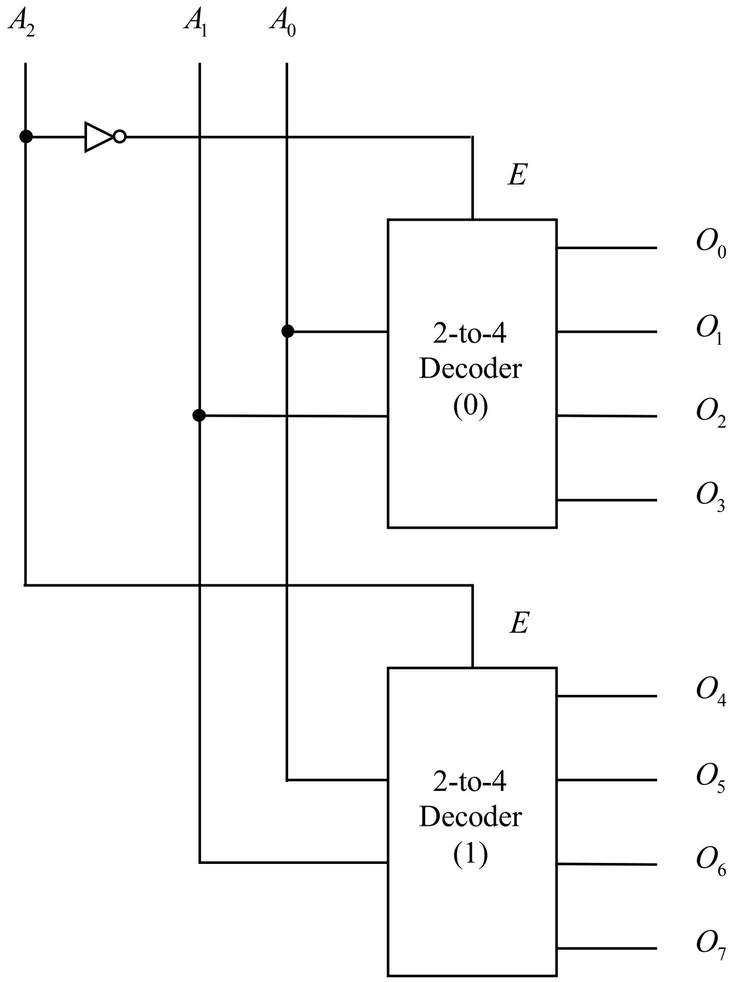
وضح جدول الصواب، ثم اكتب التعبيرات المنطقية و ارسم الدائرة المنطقية لجهاز فك الشفرة من نوع 2 إلى 4 بخطي سماح المخطط المنطقي له.

الاستخدام الأساسيلجهاز فك الشفرة

لجهاز فك الشفرة استخدامات عديدة، إلا أن أهم تلك الاستخدامات هو استخدامه في دوائر الذاكرة (Memory)، بأنواعها المختلفة، للوصول إلى موقع معين من مواقع الذاكرة عن طريق عنوانه. فلكل موقع من مواقع الذاكرة عنوان (Address) خاص به، و للوصول إلى ذلك الموقع يتم وضع عنوانه على أطراف العنوان لجهاز فك الشفرة، فينشط طرف الخرج في الجهاز المتصل بذلك الموقع و يقوم بفتح الموقع لعمليات القراءة (Read) أو الكتابة (Write). أي أن مهمة جهاز الشفرة هي الربط ما بين مواقع الذاكرة و عناوينها.

ربط دوائر جهاز فك

يمكن أن يتم ربط عدد من الوحدات الصغيرة من دوائر جهاز فك الشفرة لبناء وحدة كبيرة. مثلاً، يمكن ربط وحدتي فك شفرة من نوع 2 إلى 4 لبناء جهاز فك الشفرة من نوع 3 إلى 8، كما هو موضح أدناه



مفك الشفرة

نلاحظ أن وحدات فاك الشفرة المطلوب ربطها يجب أن تكون مزودة بخط سماح (Enable).

من جدول الصوابلجهاز فك الشفرة من نوع 3 إلى 8 الموضح أدناه. يمكن توضيح طريقة الربط:

A_2	A_1	A_0	O	
0	0	0	0	Decoder (0)
0	0	1	1	
0	1	0	2	
0	1	1	3	
1	0	0	4	Decoder (1)
1	0	1	5	
1	1	0	6	
1	1	1	7	

قمنا هنا بتقسيم جدول الصواب إلى نصفين، النصف الأعلى يقابل الوحدة الأولى (0)، والنصف الأسفل يقابل الوحدة الثانية (1). و من الجدول يمكن أن نلاحظ الآتي:

1/ أطراف الخرج للوحدة الكبيرة، و عددها هنا هو 8، موزعة بالتساوي ما بين الوحدات الصغيرة. مع ضرورة ترقيم الوحدات و مراعاة الترتيب.

2/ أطراف العنوان الدنيا، و هي أطراف العنوان التي تظهر في كل وحدة من الوحـــدات الصغيرة المطلوب ربطها، و هي هنا عبارة عن الطرفين A_0 و A_1، تكون مشتركة. و السبب في ذلك أن قيم هذه الأطراف تكون متشابهة في نصفي جدول الصواب الأعلى و الأسفل.

3/ طرف العنوان الأعلى A_2 يستخدم في اختيار الوحدة النشطة (Active Unit) من بين الوحدات المربوطة مع بعضها البعض، و ذلك عن طريق خطوط السماح (Enable) لتلك الوحدات.

لاحظ أنه عند إدخال أي عنوان من عناوين النصف الأعلى من جدول الصواب، و فيهــا جميعاً طرف العنوان الأعلى $A_2 = 0$، تنشط الوحدة الأولى (0) لأن القيمة التي تظهر في خط السماح لها هي $\overline{A_2} = 1$، في حين تكون الوحدة الثانية (1) غير نشـــطة لأن القيمـــة التي تظهر في خط السماح لها هي $A_2 = 0$. و بما أن الوحدة الأولى (0) نشطة فإنهـــا

تستجيب للقيم الموضوعة في أطراف العنوان الخاصة بها، و هي أطراف العنوان الدنيا A_0 و A_1، و ينشط أحد أطراف الخرج لها (و هي أطراف الخرج الأربعة الأولى) بناء على ذلك. أما الوحدة الثانية (1)، غير النشطة، فلا تستجيب لأطراف العنوان و تكون جميع أطراف الخرج لها (و هي أطراف الخرج الأربعة الأخيرة) غير نشطة. و عند إدخال أي عنوان من عناوين النصف الأسفل من جدول الصواب، و فيها جميعاً طرف العنوان الأعلى $A_2 = 1$، يحدث العكس، حيث تنشط الوحدة الثانية (1) و تستجيب لأطراف العنوان الدنيا A_0 و A_1، و ينشط أحد أطراف الخرج لها بناء على ذلك، في حين تكون الوحدة الأولى (0) غير نشطة و لا تستجيب لأطراف العنوان و تكون جميع أطراف الخرج لها غير نشطة.

<u>مثال:</u>

وضح طريقة ربط وحدات فك شفرة من نوع 2 إلى 4 لبناء وحدة فك من نوع 4 إلى 16.

<u>الحل:</u>

- الخطوة الأولى هنا هي تحديد عدد الوحدات الصغيرة التي نحتاج إليها في البناء، و يتم ذلك بملاحظة عدد أطراف الخرج للوحدة الصغيرة و عدد أطراف الخرج للوحدة الكبيرة المطلوب بناؤها. عدد أطراف الخرج للوحدة الصغيرة هنا هو 4، و عدد أطراف الخرج للوحدة الكبيرة هو 16، أي أننا نحتاج إلى أربعة من الوحدات الصغيرة، أي أربعة وحدات فاك شفرة من نوع 2 إلى 4.

- الخطوة الثانية هي جدول الصواب للوحدة الكبيرة، أي جدول الصواب لفك شفرة من نوع 4 إلى 16

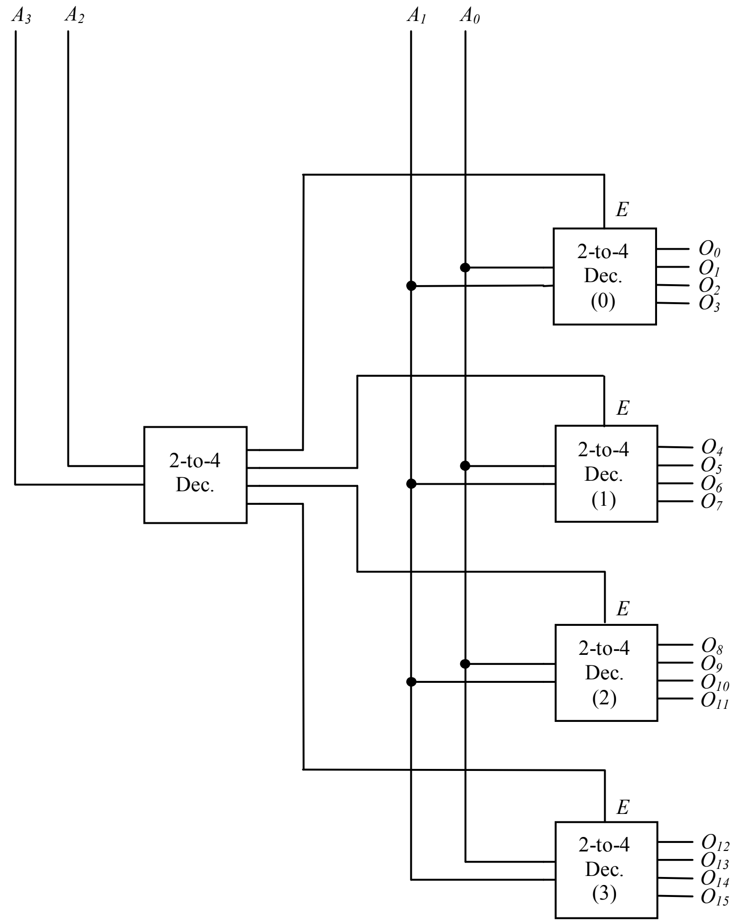
A_3	A_2	A_1	A_0	O	
0	0	0	0	0	Decoder (0)
0	0	0	1	1	
0	0	1	0	2	
0	0	1	1	3	
0	1	0	0	4	Decoder (1)
0	1	0	1	5	
0	1	1	0	6	
0	1	1	1	7	
1	0	0	0	8	Decoder (2)
1	0	0	1	9	
1	0	1	0	10	
1	0	1	1	11	
1	1	0	0	12	Decoder (3)
1	1	0	1	13	
1	1	1	0	14	
1	1	1	1	15	

- الخطوة الثالثة هي عملية الربط:

1/ أطراف الخرج للوحدة الكبيرة موزعة بالتساوي ما بين الوحدات الصغيرة.

2/ أطراف العنوان الدنيا A_0 و A_1 مشتركة.

3/ أطراف العنوان العليا A_2 و A_3 تستخدم في اختيار الوحدة النشطة. و نستعين في اختيار الوحدة النشطة هنا بفاك شفرة من نوع 2 إلى 4.



بناء مفكك شفرة من نوع 4 الى 16

نشاط

تحقق من صحة عمل الدائرة أعلاه، و ذلك بوضع عدد من العناوين المختلفة على أطراف العنوان للدائرة و إيجاد الطرف الذي ينشط، و ذلك للتأكد من أنه فعلاً الطرف صاحب العنوان الموضوع.

ملاحظة

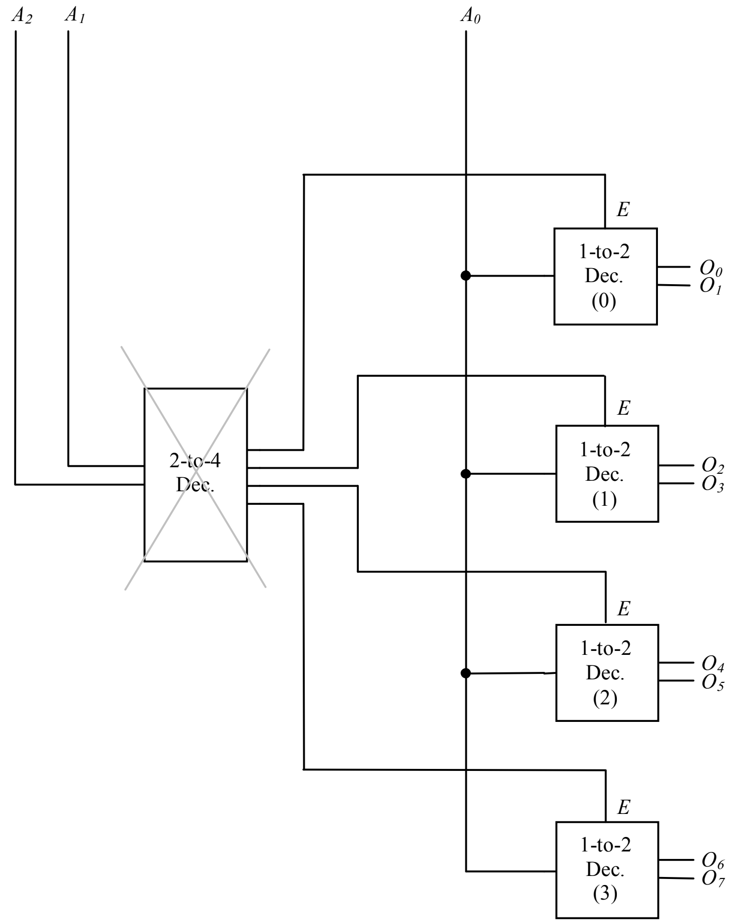
عندما قمنا بربط دائرتي فك شفرة من نوع 2 إلى 4 لبناء وحدة فـك شفرة من نوع 3 إلى 8 استخدمنا في اختيار الوحدة النشـطة عاكسـاً منطقياً، و بالطبع فإن في إمكاننا أن نستخدم في اختيار الوحدة النشطة فاك شفرة من نوع 1 إلى 2. و في واقع الأمر فإن وحدة فك الشفرة من نوع 1 إلى 2 (بدون خـط سـماح (Enable)) تتكـون دائرتـه المنطقية من عاكس منطقي واحد فقط. (ارجع إلى تدريب 7 (أ))

<u>مثال:</u>

وضح طريقة بناء وحدة فك شفرة من نوع 3 إلى 8 باستخدام وحدات فك شفرة من نوع 1 إلى 2.

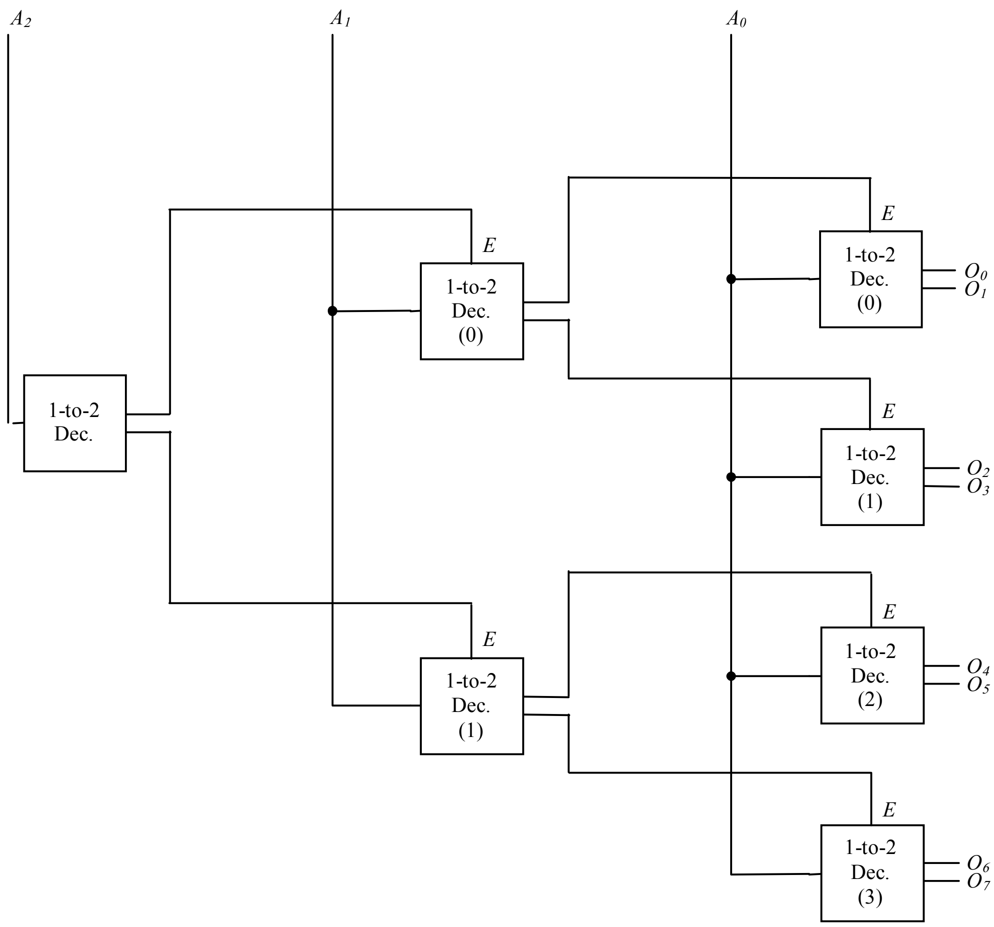
<u>الحل:</u>

الحل التالي (الذي يشبه إلى حد كبير حل المثال السابق) هو أول ما يتبادر إلى الذهن



و لكن هذا الحل ليس هو بالحل الصحيح، لأنه إذا قرأنا نص المسألة بدقة نجد أن المطلوب استخدام دوائر فك شفرة من نوع 1 إلى 2 فقط في البناء، و عليه فمن غير المسموح لنا استخدام فك الشفرة من نوع 2 إلى 4 الذي نحتاج إليه في عملية الربط. وحل هذه الإشكالية بسيط، حيث نقوم ببناء جهاز فك الشفرة من نوع 2 إلى 4 نفسه باستخدام دوائر وحدة فك شفرة من نوع 1 إلى 2.

و عليه فإن الحل الصحيح للمسألة هو



تدريب 11

وضح طريقة بناء وحدة فك شفرة من نوع 4 إلى 16 (4-to-16 Decoder) و ذلك باستخدام دوائر وحدة فك شفرة من نوع:

(أ) 3 إلى 8 (3-to-8 Decoders).

(ب) 2 إلى 4 (2-to-4 Decoders).

(ج) 1 إلى 2 (1-to-2 Decoders).

أسئلة تقويم ذاتي

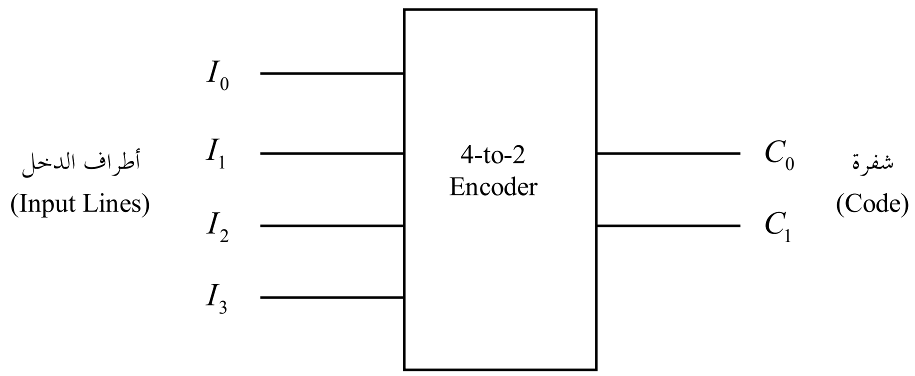
1- متى يكون فاك الشفرة ذي خرج نشط منخفض؟

2- ما أهم استخدامات فاك الشفرة؟

5 . المشفر (Encoder)

كما هو واضح من التسمية فإن المشفر (Encoder) يؤدي عكس الوظيفة التـــي يؤديهـــا وحدة فك الشفرة (Decoder). حيث أن المشفر عبارة عن دائرة منطقية لها عدة أطراف دخل (Input Lines)، و يكون واحد فقط من أطراف الدخل هـــذه نشـــطاً (Active)، أي مساوياً 1، أما بقية أطراف الدخل تكون غير نشطة، أي مساوية 0. خرج الدائرة عبارة عن شفرة (Code) تمثل طرف الدخل النشط.

و في ما يلي المخطط المنطقي و جدول الصواب لمشفر من نوع 4 إلى 2 (4-to-2 Encoder)



I_3	I_2	I_1	I_0	C_1	C_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

لاحظ أن جدول الصواب الموضح أعلاه هو جدول صواب مختصر، تظهر فيه احتمالات الدخل الواردة فقط، و عددها أربعة، حيث إن طريقة عمل المشفر تشترط أن يكون طرف واحد فقط من أطراف الدخل نشطاً. أما جدول الصواب الكامل فيحتوي على 16 احتمال دخل، الخرج المقابل للـــ 12 احتمال دخل غير الواردة منها عبارة عن قيم غير محددة (Don't Cares).

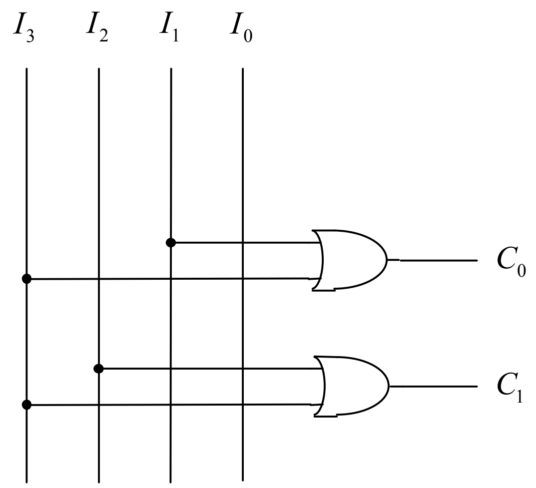
التعبيرات المنطقية:

سنقوم هنا بكتابة التعبيرات المنطقية المختصرة مباشرة من جدول الصـــواب المختصـــر بأسلوب غير تقليدي. فلكتابة التعبير المختصر لمتغير معين من متغيرات الخرج نبحث أسفل ذلك المتغير في جدول الصواب عن الـــ 1's، و لكل 1 نجده أسفل متغير الخـــرج نقوم بأخذ متغير الدخل الذي يساوي 1 في نفس السطر من جدول الصواب، ثـــم نـــربط متغيرات الدخل هذه مع بعضها البعض بعمليات OR.

$$C_0 = I_1 + I_3$$
$$C_1 = I_2 + I_3$$

أي أن متغير الخرج C_0 يساوي 1 إذا كان طرف الدخل I_1 مساوياً 1 أو إذا كان طرف الدخل I_3 مساوياً 1. و متغير الخرج C_1 يساوي 1 إذا كان طرف الدخل I_2 مساوياً 1 أو إذا كان طرف الدخل I_3 مساوياً 1.

الدائرة المنطقية:



المشفر من 4 الى 2

لاحظ أن الدائرة المنطقية للمشفر تتكون أساساً من مجموعة من بوابات OR بعدد أطراف الخرج.

لاحظ أنه يجب أن يكون لكل طرف من أطراف الدخل (Input Lines) في المشفر شفرة فريدة تميزه. فإذا كان عدد أطراف الخرج هو N فإن عدد الشفرات المتاحة 2^N، و بالتالي فإن عدد أطراف الدخل يجب أن يكون أقل من أو مساوياً 2^N.

نشاط

قمنا بكتابة التعبيرات المنطقية المختصرة لدائرة المشفر من نوع 4 إلى 2 الموضحة أعلاه مباشرة من جدول الصواب المختصر، و ذلك باستخدام أسلوب غير تقليدي. المطلوب الآن اتباع الأسلوب التقليدي للوصول إلى **نفس التعبيرات المختصرة**، و ذلك كالتالي:

- إنشاء جدول الصواب الكامل.
- كتابة التعبيرات المنطقية في صورة: – مجموع الحدود الصغرى (Sum of minterms).

– مضروب الحدود الكبرى (Product of Maxterms).

- تبسيط التعبيرات المنطقية في كلا الصورتين باستخدام مخططات كارنو.

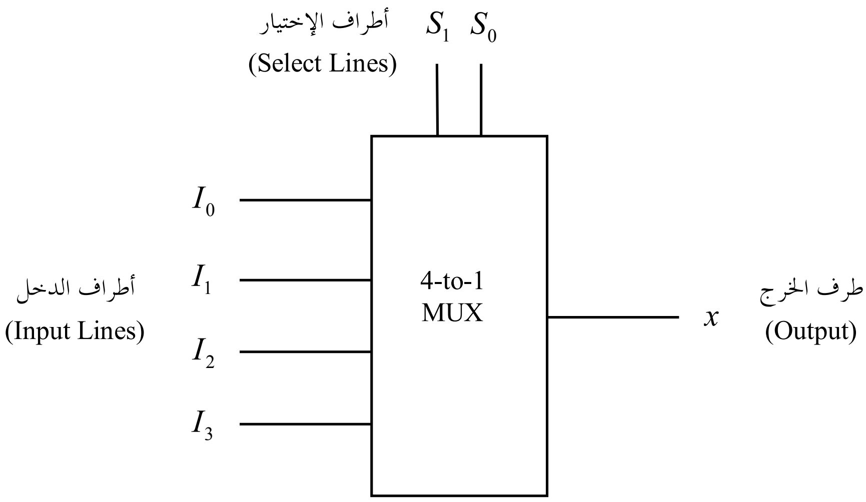
تدريب 12

وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبيرات المنطقية و ارسم الدائرة المنطقية لمشفر من نوع 8 إلى 3 (8-to-3 Encoder).

6 . الدامج (Multiplexer)

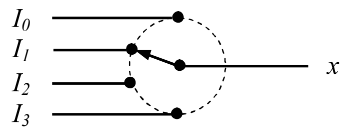
الدامج عبارة عن دائرة منطقية لها عدة أطراف دخل، و طرف خرج واحد. يتم توصيل واحد من أطراف الدخل مع طرف الخرج، و يتم اختيار طرف الدخل الذي يتم توصيله بالخرج بواسطة أطراف الإختيار (Select Lines).

و في ما يلي المخطط المنطقي و جدول الصواب لدامج من نوع 4 إلى 1 (4-to-1 Multiplexer)



S_1	S_0	x
0	0	I_0
0	1	I_1
1	0	I_2
1	1	I_3

هذا و يمكن تشبيه عمل الدامج بعمل المفتاح الدائري (Rotary Switch) الموضح أدناه

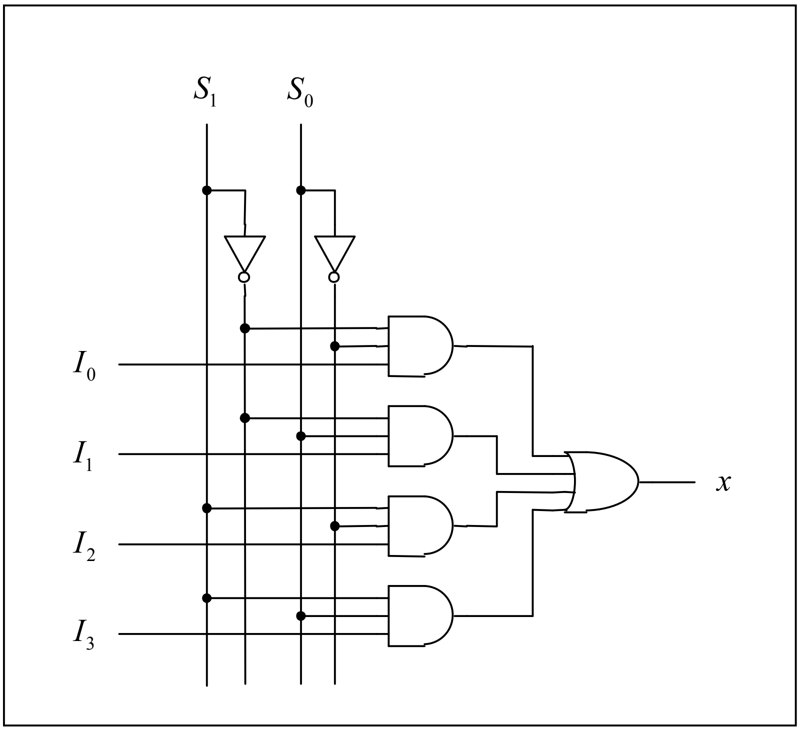


التعبير المنطقي:

$$x = \overline{S_1}\,\overline{S_0}I_0 + \overline{S_1}S_0I_1 + S_1\overline{S_0}I_2 + S_1S_0I_3$$
$$x = m_0I_0 + m_1I_1 + m_2I_2 + m_3I_3$$

حيث m_0 ، m_1 ، m_2 ، m_3 هي الحدود الصغرى (minterms) لمتغيري الاختيار S_0 و S_1.

الدائرة المنطقية:



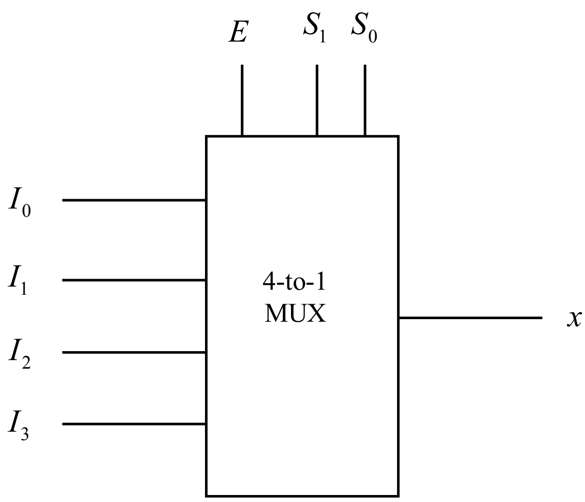
تدريب 13

وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبير المنطقي و ارسم الدائرة المنطقية لدامج من نوع 8 إلى 1 (8-to-1)

خط السماح (Enable)

في بعض الأحيان قد يكون الدامج مزوداً بخط سماح (Enable). و وظيفة خط السماح، كما نعلم، هي إبطال عمل الدائرة أو السماح لها بأن تؤدي وظيفتها كالمعتاد.

و في ما يلي طريقة ظهور خط السماح في دامج من نوع 4 إلى 1

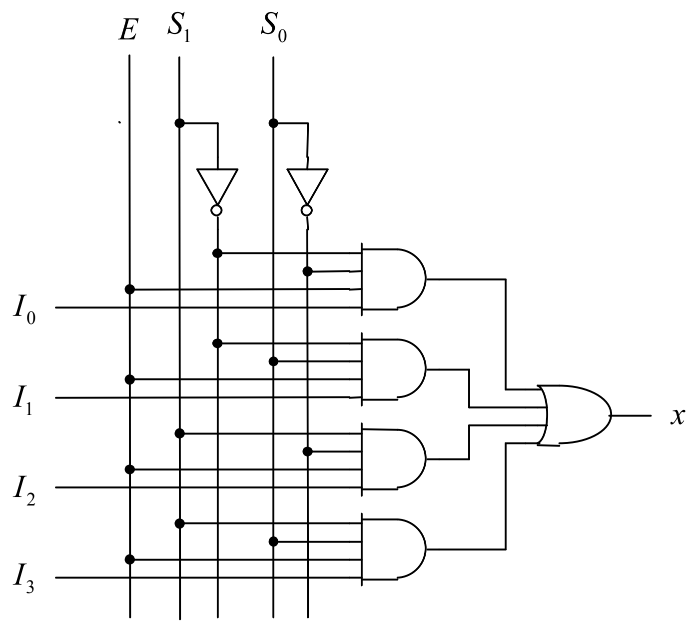


E	S_1	S_0	x
0	×	×	0
1	0	0	I_0
1	0	1	I_1
1	1	0	I_2
1	1	1	I_3

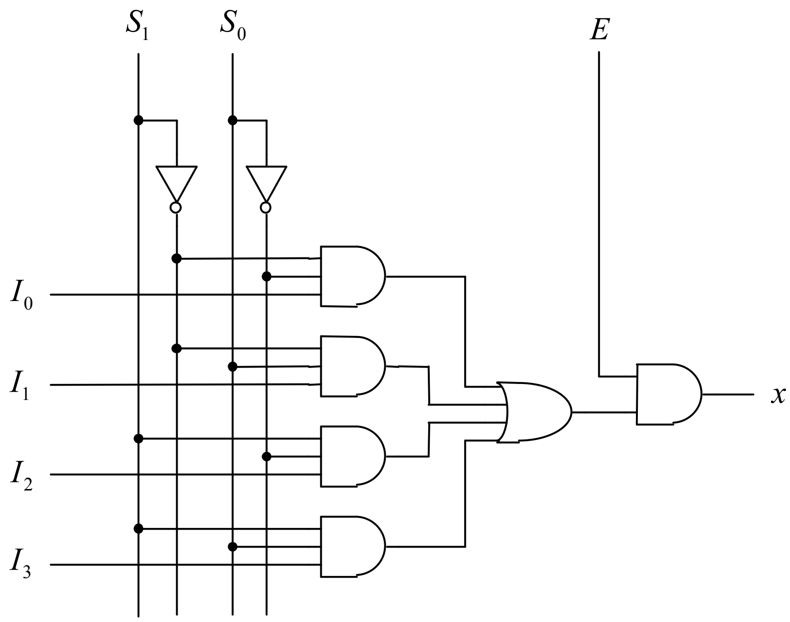
التعبير المنطقي:

$$x = E\overline{S_1}\,\overline{S_0}I_0 + E\overline{S_1}S_0I_1 + ES_1\overline{S_0}I_2 + ES_1S_0I_3$$
$$x = E(\overline{S_1}\,\overline{S_0}I_0 + \overline{S_1}S_0I_1 + S_1\overline{S_0}I_2 + S_1S_0I_3)$$

الدائرة المنطقية:



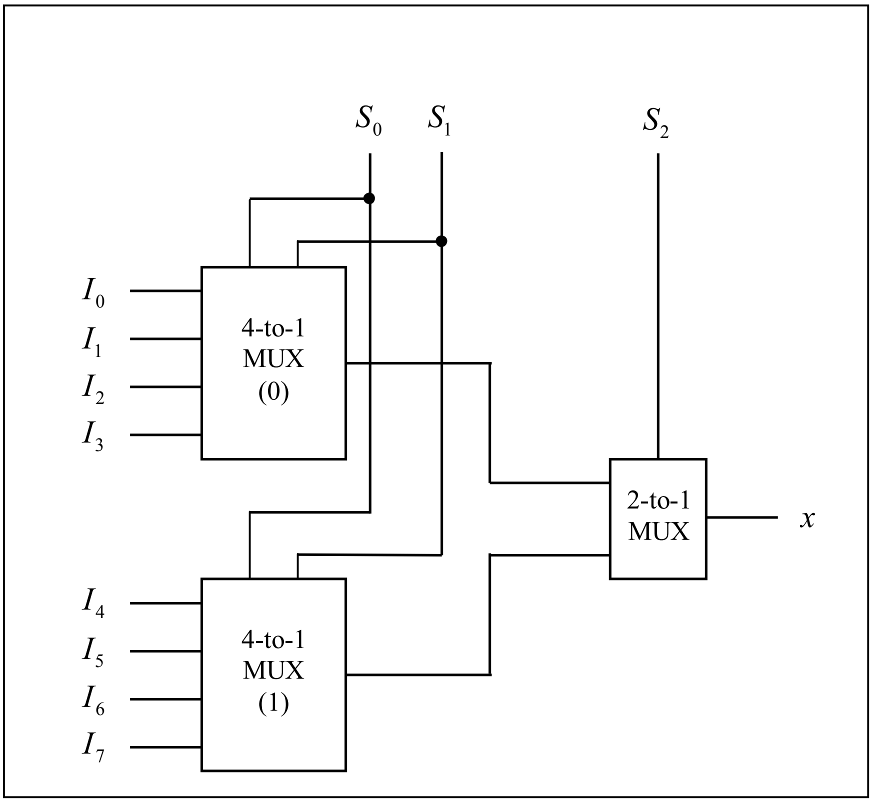
أو



من الواضح أن الدائرة الثانية أفضل من الأولى لأنها أبسط.

ربط الدوامج:

يمكن ربط عدد من وحدات الدامج الصغيرة لبناء وحدة دامج أكبر. مثلاً، يمكن ربط وحدتي دامج من نوع 4 إلى 1 (4-to-1 MUX's) لبناء دامج من نوع 8 إلى 1 (8-to-1 MUX)، كما هو موضح أدناه



دامج من نوع 8 إلى 1 باستخدام وحدتي دمج من نوع 4 الى 1

من الشكل أعلاه يمكن ملاحظة أن خطوات الربط هي:

1/ أطراف الدخل للوحدة الكبيرة موزعة بالتساوي ما بين الوحدات الصغيرة.

2/ أطراف الاختيار الدنيا، و هي أطراف الاختيار التي تظهر في كل وحدة من الوحدات الصغيرة المطلوب ربطها، تكون مشتركة.

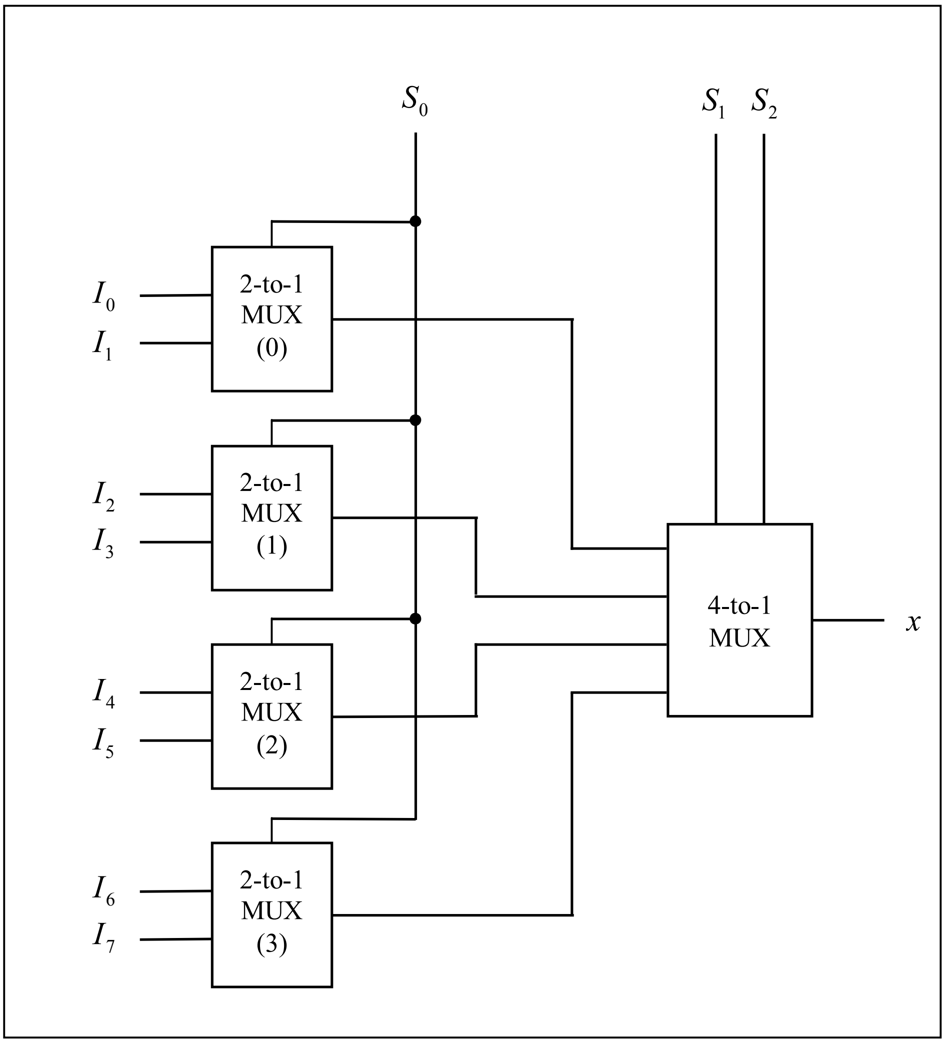
3/ طرف الاختيار الأعلى يستخدم في اختيار الوحدة الصغيرة التي يتم تمرير خرجها إلى خرج الوحدة الكبيرة من بين الوحدات المربوطة مع بعضها البعض، و ذلك باستخدام دامج عدد أطراف الدخل له يساوي عدد الوحدات المربوطة.

مثال:

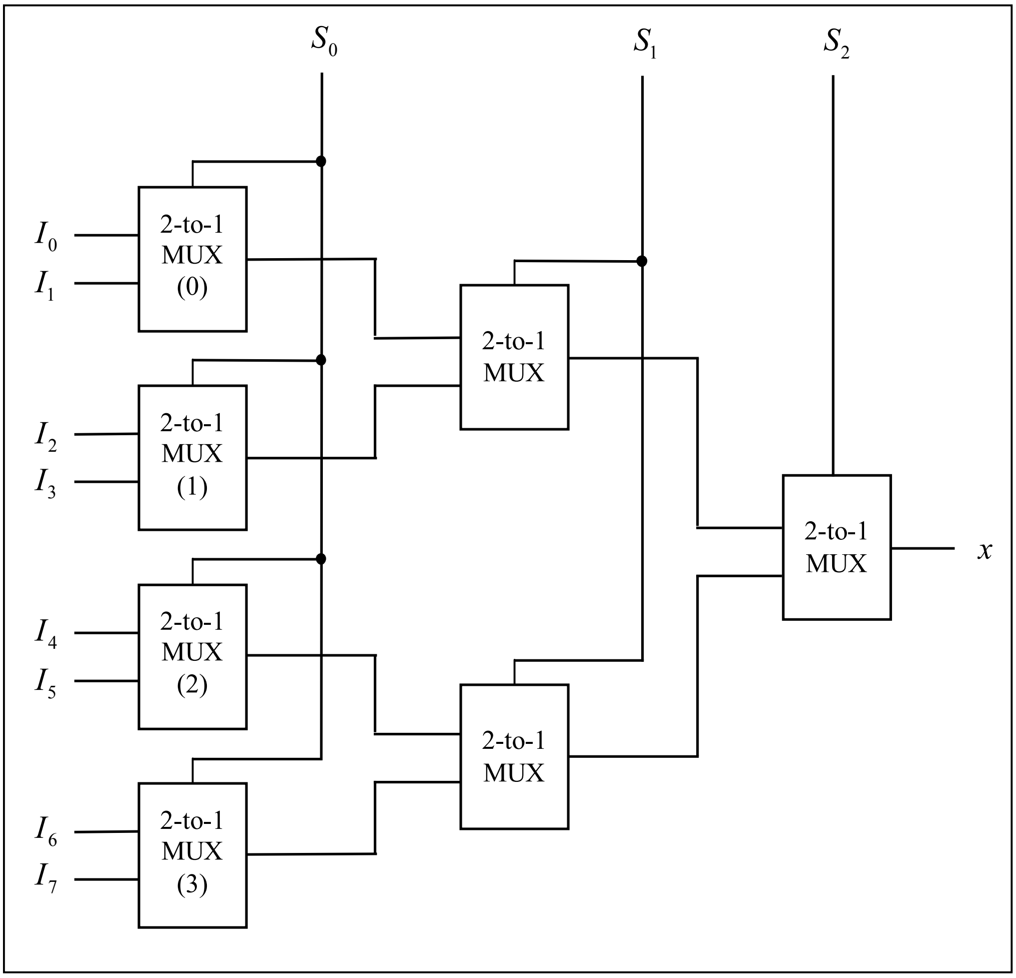
وضح طريقة ربط وحدات دامج من نوع 2 إلى 1 لبناء دامج من نوع 8 إلى 1.

الحل:

بمقارنة عدد أطراف الدخل للوحدة الصغيرة المستخدمة في البناء و عدد أطراف الدخل للوحدة الكبيرة المطلوب بناؤها يتضح لنا أن عدد الوحدات الصغيرة التي نحتاج إليها هو 4. أما الربط فيتم كالتالي:

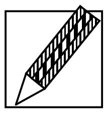


في المثال أعلاه إذا كان مطلوباً بناء الدامج من نوع 8 إلى 1 باستخدام وحدات دامج من نوع 2 إلى 1 فقط فما علينا إلا أن نقوم ببناء الدامج من نوع 4 إلى 1 المستخدم في الربط باستخدام وحدات دامج من نوع 2 إلى 1، كما هو موضح أدناه.



بناء دامج من النوع 8 الى 1

تدريب 14



وضح طريقة بناء دامج من نوع 16 إلى 1 (16-to-1 Multiplexer) و ذلك باستخدام وحدات دامج من نوع:

(أ) 4 إلى 1 (4-to-1 Multiplexers).

(ب) 2 إلى 1 (2-to-1 Multiplexers).

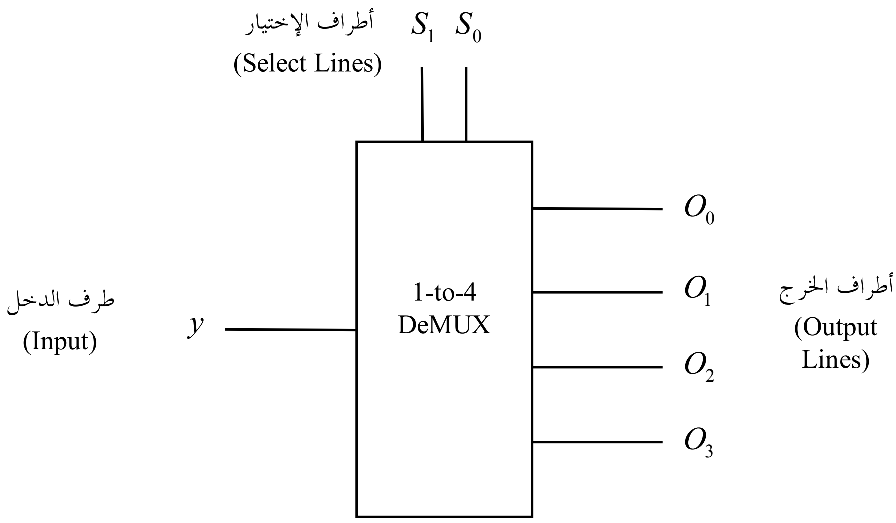
أسئلة تقويم ذاتي

1- ما خطوات ربط وحدات الدمج؟

7 . المفرق (Demultiplexer)

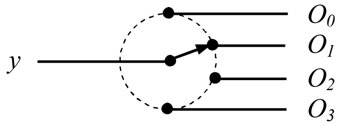
واضح من التسمية أن المفرق (Demultiplexer) يؤدي عكس الوظيفة التي يؤديها الدامج (Multiplexer)، فالمفرق عبارة عن دائرة منطقية لها عدة أطراف خرج، و طرف دخل واحد. يتم توصيل طرف الدخل مع أحد أطراف الخرج، و يتم اختيار طرف الخرج الذي يتم توصيله بالدخل بواسطة أطراف الاختيار (Select Lines).

و في ما يلي المخطط المنطقي و جدول الصواب لمفرق من نوع 1 إلى 4 (1-to-4 Demultiplexer)



S_1	S_0	O_3	O_2	O_1	O_0
0	0	0	0	0	y
0	1	0	0	y	0
1	0	0	y	0	0
1	1	y	0	0	0

هذا و يمكن تشبيه عمل المفرق بعمل المفتاح الدائري (Rotary Switch) الموضح أدناه

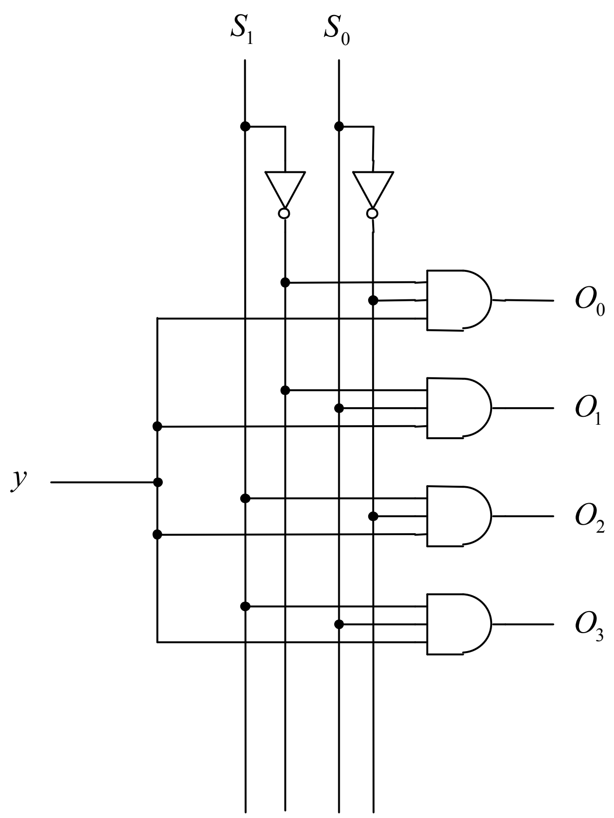


التعبيرات المنطقية:

$$O_0 = \overline{S_1}\,\overline{S_0}y = m_0 y$$
$$O_1 = \overline{S_1}S_0 y = m_1 y$$
$$O_2 = S_1\overline{S_0}y = m_2 y$$
$$O_3 = S_1 S_0 y = m_3 y$$

حيث m_0، m_1، m_2، m_3 هي الحدود الصغرى (minterms) لمتغيري الاختيار S_0 و S_1.

الدائرة المنطقية:



<u>لاحظ</u> أن هذه الدائرة المنطقية تتطابق تماماً مع الدائرة المنطقية لوحدة فك شفرة من نوع 2 إلى 4 مزود بخط سماح، مما يعني أن المفرق من نوع 1 إلى 4 يمكن استخدامه كفاك شفرة من نوع 2 إلى 4 مزود بخط سماح، و ذلك باستبدال طرفي الاختيار S_0 و S_1 بطرفي العنوان A_0 و A_1، و استبدال طرف الدخل y بخط السماح E.

تدريب 15

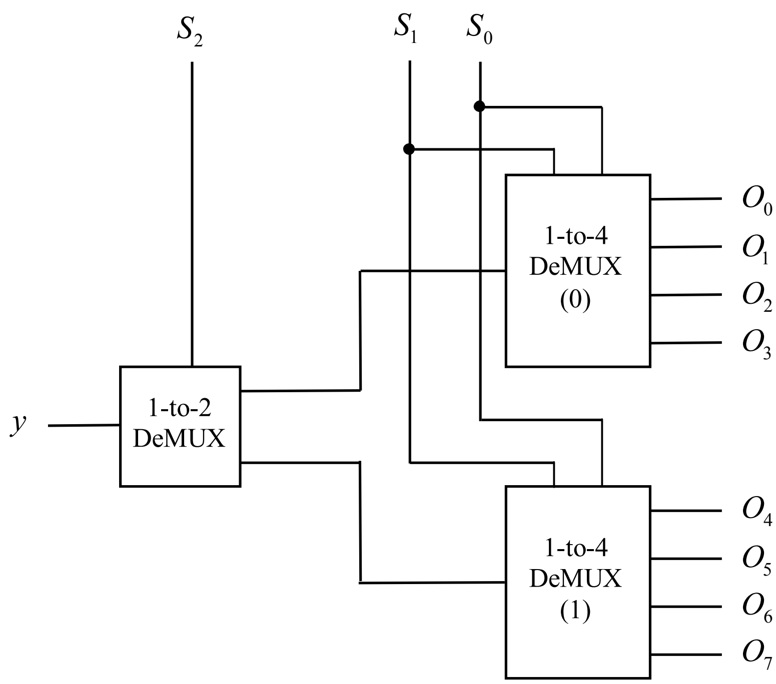
وضح المخطط المنطقي و جدول الصواب، ثم اكتب التعبيرات المنطقية و ارسم الدائرة المنطقية لمفرق من نوع:

(أ) 1 إلى 2 (1-to-2 Demultiplexer).

(ب) 1 إلى 8 (1-to-8 Demultiplexer).

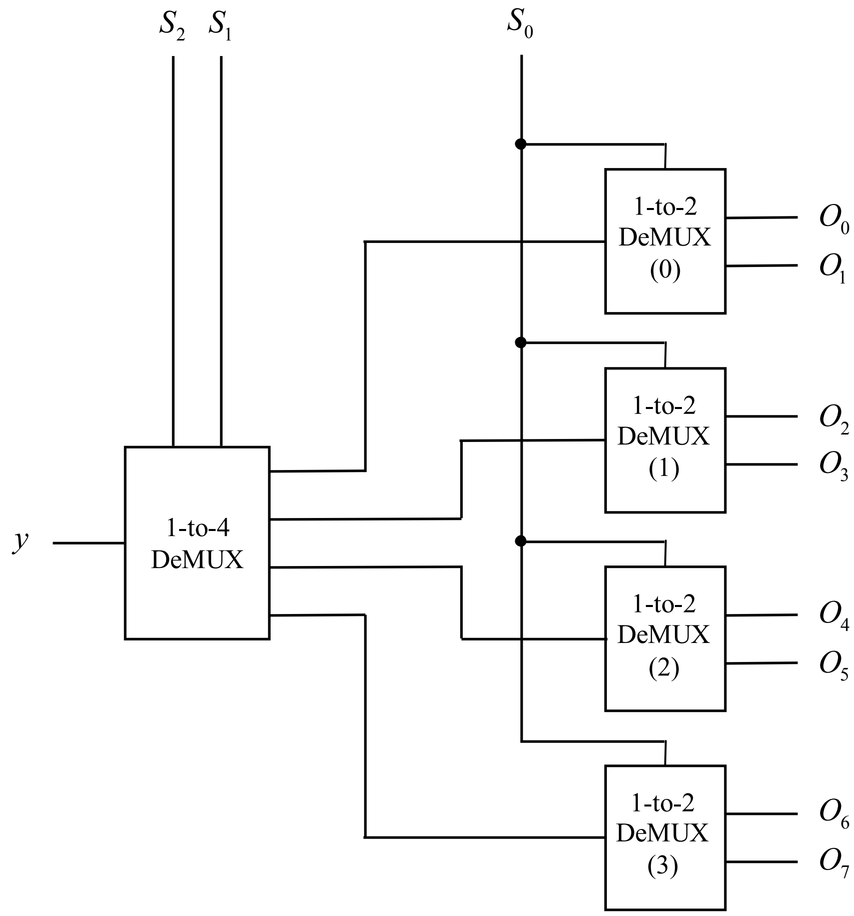
ربط المفرقات

يتم ربط المفرقات بنفس الأسلوب الذي أستخدم في ربط الدوامج. على سبيل المثال يمكن ربط وحدتي مفرق من نوع 1 إلى 4 لبناء مفرق من نوع 1 إلى 8 كما هو موضح بالشكل التالي:



مفرق من النوع 1 الى 8

كما يمكن ربط 4 وحدات مفرق من نوع 1 إلى 2 لبناء مفرق من نوع 1إلى 8 كما هو موضح أدناه



تدريب 16

وضح طريقة بناء مفرق من نوع 1 إلى 8 باستخدام وحدات مفرق من نوع 1 إلى 2.

استخدام عملي للدامج و المفرق

يوجد أسلوبان لنقل البيانات (Data Transmission) في الأنظمة الرقمية:

1/ النقل على التوازي (Parallel Transmission).

2/ النقل على التوالي (Serial Transmission).

النقل على التوازي (Parallel Transmission):

في هذا الأسلوب يتم نقل مجموعة من الـــ bits دفعة واحدة على التوازي. و يتطلب هذا وجود موصل (سلك) لكل bit من الـــ bits المنقولة، إضافة إلى موصل أرضي (Ground) يستخدم كمرجع لقياس الجهد الكهربائي على بقية الموصلات، كما هو موضح أدناه

B_0 ————————————

B_1 ————————————

B_2 ————————————

B_3 ————————————

GND ————————————

يمتاز هذا الأسلوب في نقل البيانات بالسرعة، و لكن يعيبه ارتفاع تكلفة الكيبل المستخدم و عدم إمكانية نقل البيانات لمسافات طويلة.

يستخدم هذا الأسلوب في نقل البيانات داخل جهاز الحاسوب بين أجزائه المختلفة مثل المعالج (Processor) و الذاكرة (Memory) عبر الناقل (Bus)، كما يستخدم في نقل البيانات من جهاز الحاسوب إلى الطابعة (Printer).

النقل على التوالي (Serial Transmission):

في هذا الأسلوب يتم استخدام موصلين فقط لنقل مجموعة من الـــ bits واحداً تلو الآخر، كما هو موضح أدناه

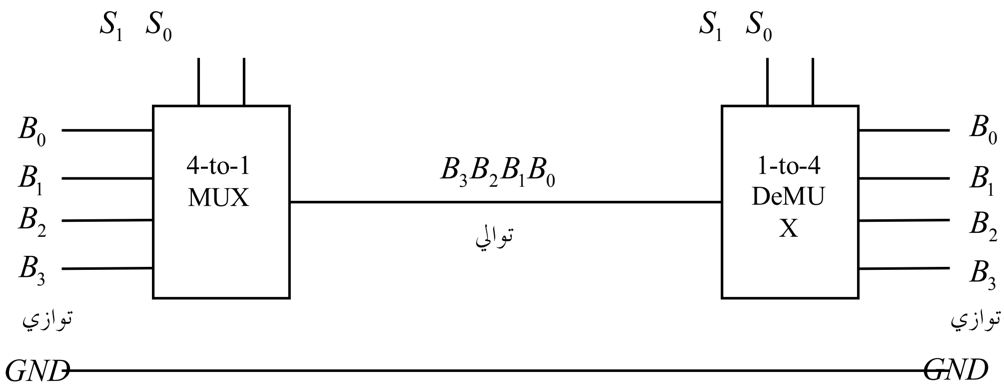
$B_3B_2B_1B_0$ ————————————

GND ————————————

هذا الأسلوب في نقل البانات يعيبه البطء، إلا أنه يمتاز بانخفاض تكلفة الكيبل المستخدم و إمكانية نقل البيانات لمسافات طويلة.

يستخدم هذا الأسلوب في نقل البيانات في شبكات الحاسوب (Networks)، كما يستخدم عندما تكون السرعة في نقل البيانات غير مطلوبة مثل نقل البيانات من لوحة المفاتيح (Keyboard) و الفأرة (Mouse) إلى جهاز الحاسوب.

في بعض الأحيان قد يكون مطلوباً تحويل البيانات المنقولة داخل النظام الرقمي من توازي إلى توالي أو العكس، كما يحدث داخل كرت الشبكة (Network Interface Card)، حيث يستقبل الكرت البيانات من جهاز الحاسوب على التوازي و يرسلها عبر كيبل الشبكة على التوالي. فلابد هنا من إجراء عملية تحويل للبيانات المنقولة من توازي إلى توالي. كما يجب إجراء العملية العكسية، أي التحويل من توالي إلى توازي، عند استقبال البيانات من سلك الشبكة ونقلها إلى جهاز لحاسوب. هنا يأتي دور كل من الدامج و المفرق، حيث يمكن استخدام الدامج في التحويل من توازي إلى توالي، و استخدام المفرق في التحويل من توالي إلى توازي، كما هو موضح بالشكل أدناه



حيث يقوم كل من المرسل والمستقبل بوضع القيمة 00 في طرفي الإختيار S_0 و S_1 مما يعني إرسال القيمة الأولى B_0 ، بعدها يتم وضع القيمة 01 في طرفي الإختيار ممايعني إرسال القيمة B_1 ...... وهكذا

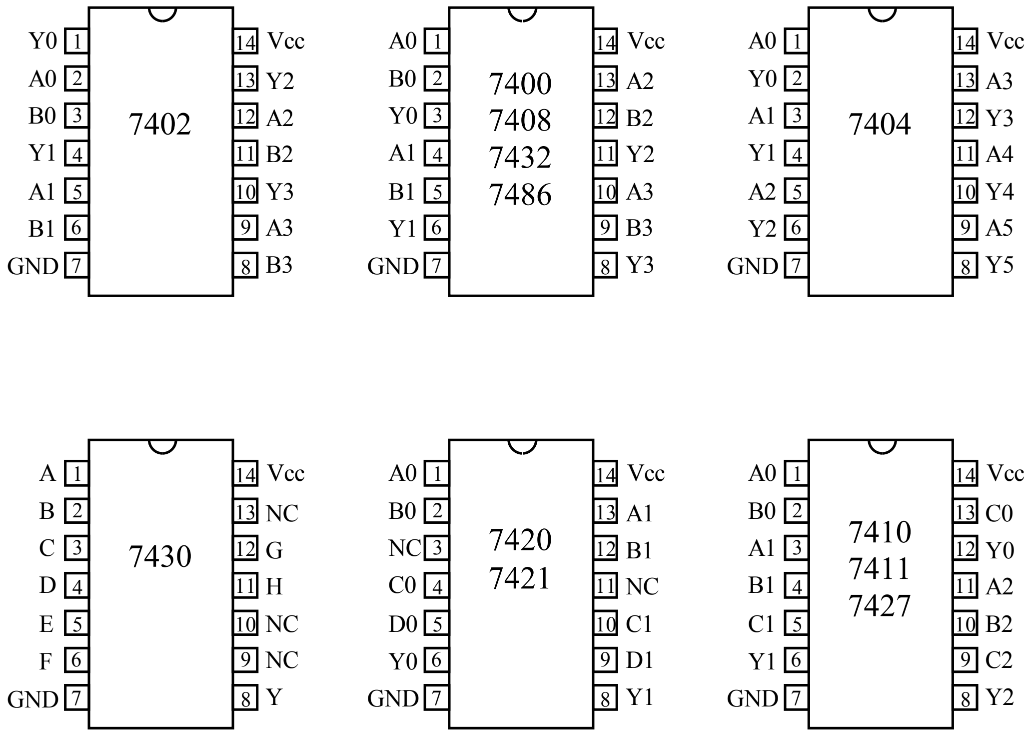
أسئلة تقويم ذاتي

❓ قارن بين أسلوبي النقل على التوازي والنقل على التوالي.

8 . طرق بديلة لتصميم الدوائر المنطقية

السؤال الذي قد يتبادر إلى الذهن هنا هو: لماذا نحتاج إلى طرق تصميم بديلة؟

الطريقة التي درسناها لتصميم الدوائر المنطقية تهدف بالأساس إلى تقليل عدد البوابات المنطقية المستخدمة في بناء الدائرة لتقليل تكلفة الدائرة. و يصلح هذا الأسلوب إذا كان بناء الدائرة سيتم في مصنع متخصص في تصنيع الدوائر المنطقية، حيث يتم في هذه الحالة تصنيع البوابات اللازمة لبناء الدائرة من مكوناتها الأساسية (بلورات شبه موصلة)، و ذلك على سطح شريحة صغيرة من السيليكون، ثم يتم ربط تلك البوابات مع بعضها البعض لبناء الدائرة، و أخيراً يتم تغليف شريحة السيليكون في شكل دائرة متكاملة (Integrated Circuit). و يتم بناء الدائرة المنطقية بهذه الطريقة عندما يكون مطلوباً تصنيع عدد كبير من الوحدات من تلك الدائرة، أما إذا كان المطلوب هو عدد محدود من الوحدات من الدائرة فإن تصنيع تلك الدائرة في مصنع متخصص يكون غير عملي، حيث إن تكلفة إنتاج الوحدة الواحدة هنا ستكون مرتفعة للغاية. لذلك يتم بناء الدائرة المنطقية في مثل هذه الحالات باستخدام البوابات المنطقية الجاهزة المتوفرة تجارياً في شكل دوائر متكاملة (IC's). و الشكل التالي يوضح بعضاً من تلك الدوائر المتكاملة المتوفرة تجارياً



بعض الدوائر المتكاملة المتوفرة تجارياً

الدائرة المتكاملة	محتوياتها
7404	Hex (6) Inverter
7400	Quad (4) 2-Input NAND Gate
7408	Quad (4) 2-Input AND Gate
7432	Quad (4) 2-Input OR Gate
7486	Quad (4) 2-Input XOR Gate
7402	Quad (4) 2-Input NOR Gate
7410	Triple (3) 3-Input NAND Gate
7411	Triple (3) 3-Input AND Gate
7427	Triple (3) 3-Input NOR Gate
7420	Dual (2) 4-Input NAND Gate
7421	Dual (2) 4-Input AND Gate
7430	8-Input NAND Gate

Vcc و GND هما طرفا تزويد الدائرة المتكاملة بالقدرة الكهربائية (Power).

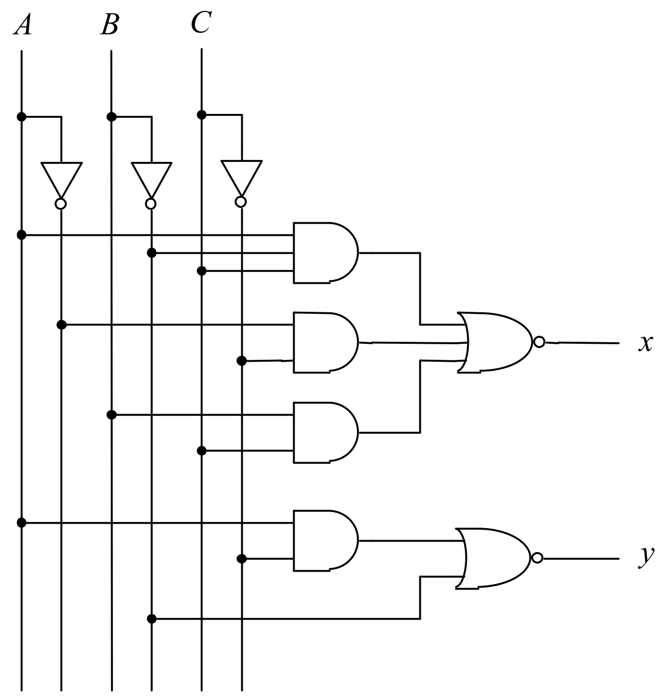
NC تعني (No Connection) أي أن الطرف غير مستخدم.

A0، B0، C0، .... هي أطراف الدخل للبوابة رقم 0

Y0 هو طرف الخرج للبوابة رقم 0

من المذكور أعلاه نلاحظ أن البوابات المنطقية الجاهزة لا تُباع منفردة و إنما تأتي مجموعة منها من نوع واحد في شكل دائرة متكاملة (.I.C). فمثلاً العواكس المنطقية تأتي ستة منها في الدائرة المتكاملة 7404، و بوابات AND بمدخلين تأتي أربعة منها في الدائرة المتكاملة 7408، و بوابات NOR بثلاثة مداخل تأتي ثلاثة منها في الدائرة المتكاملة 7427، و بوابات NAND بأربعة مداخل تأتي اثنتان منها في الدائرة المتكاملة 7420، ... و هكذا.

فإذا أردنا بناء الدائرة المنطقية التالية، مثلاً، باستخدام البوابات الجاهزة



فإننا نحتاج إلى:

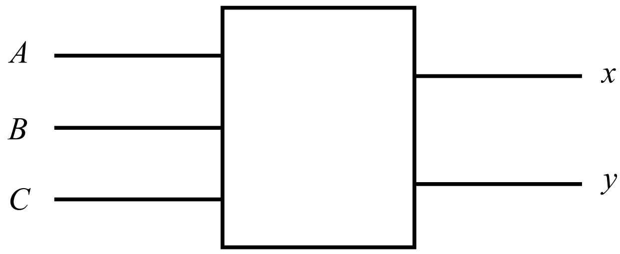
- 3 عواكس منطقية ($\frac{3}{6}\times 7404$).
- 3 بوابات AND بمدخلين ($\frac{3}{4}\times 7408$).
- بوابة AND بثلاثة مداخل ($\frac{1}{3}\times 7411$).
- بوابة NOR بمدخلين ($\frac{1}{4}\times 7402$).
- بوابة NOR بثلاثة مداخل ($\frac{1}{3}\times 7427$).

أي أننا نحتاج إلى 5 دوائر متكاملة، وهذا العدد من الدوائر المتكاملة يشغل مساحة كبيرة من سطح لوحة التوصيل. كما أن عدداً مقدراً من البوابات بتلك الدوائر المتكاملة غير مستخدم، مع العلم بأن تلك البوابات غير المستخدمة تستهلك القدرة الكهربائية، مما يعني هدراً للأموال و للطاقة الكهربائية و انبعاث كمية من الحرارة قد لا يكون من السهل التخلص منها.

إذن المطلوب الآن هو طرق تصميم بديلة للدوائر المنطقية تهدف بالأساس إلى تقليل عدد الدوائر المتكاملة (IC's) المستخدمة في بناء الدائرة المنطقية، بدلاً عن تقليل عدد البوابات المنطقية المستخدمة في بناء الدائرة المنطقية. في طرق التصميم البديلة هذه نستعين بالدوائر المنطقية الترابطية الجاهزة مثل وحدة فك الشفرة (Decoder) و الدامج (Multiplexer).

التصميم باستخدام وحدة فك شفرة و مشفر (Decoder & Encoder)

المطلوب تصميم الدائرة المنطقية التي يوضحها المخطط المنطقي و جدول الصواب لها أدناه باستخدام فاك شفرة و مشفر



#	A	B	C	x	y
0	0	0	0	0	1
1	0	0	1	1	0
2	0	1	0	0	1
3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	0	1
6	1	1	0	0	0
7	1	1	1	1	0

خطوات التصميم:

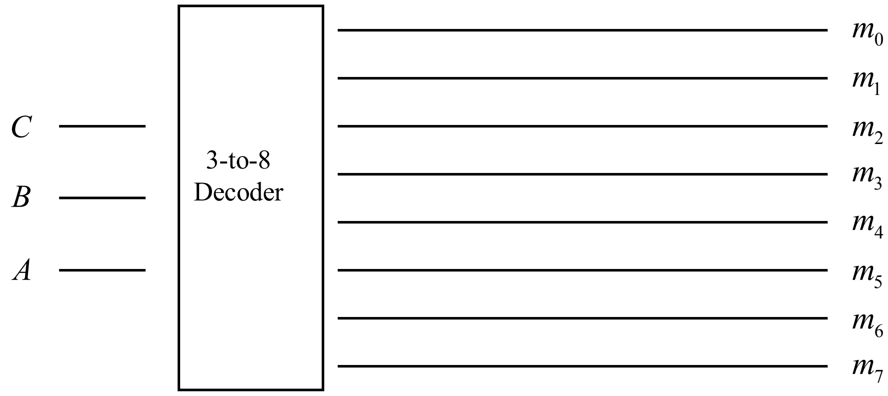
1/ نقوم بكتابة التعبيرات المنطقية في صورة مجموع الحدود الصغرى (Sum of minterms)

$$x = f(A,B,C) = \sum m\ (1,3,7)$$
$$y = f(A,B,C) = \sum m\ (0,2,4,5)$$

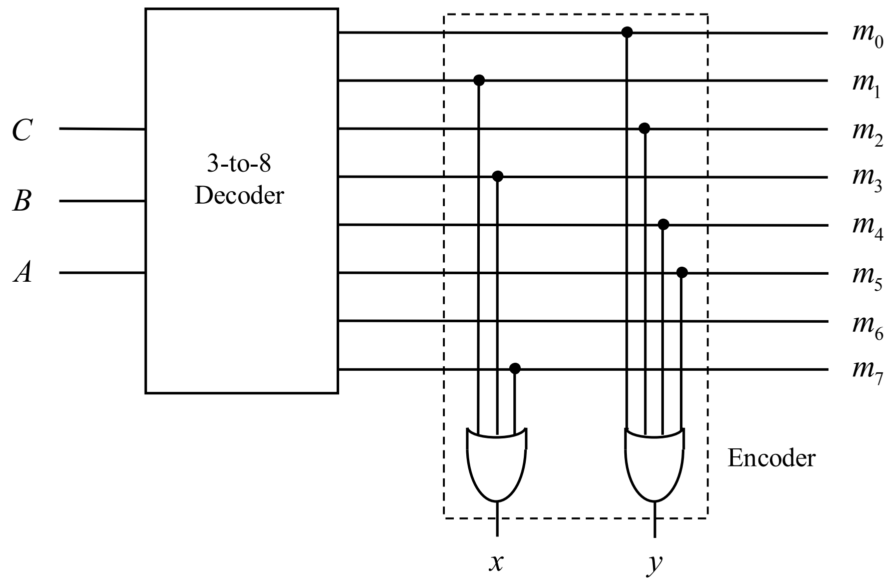
2/ نحتاج إلى فاك شفرة عدد أطـراف العنـوان (Address Lines) لـه يسـاوي عـدد متغيرات الدخل للدائرة المنطقية المطلوب تصميمها، أي وحدة فك شفرة من نوع 3 إلى 8.

نقوم بإدخال متغيرات الدخل للدائرة المنطقية المطلوب تصميمها إلى أطـراف العنـوان لفك الشفرة، مع مراعاة الترتيب (أي أن الـ MSB في متغيرات الدخل يجب إدخاله إلى

الـــ MSB في أطراف العنوان)، فتقوم وحدة فك الشــفرة بتوليــد الحــدود الصــغرى (minterms) لمتغيرات الدخل في أطراف الخرج له، كما هو موضح أدناه



3/ نستخدم بوابات OR في جمع الحدود الصغرى المناسبة لتوليد متغيرات الخرج



لاحظ أن بوابات OR هنا تمثل مشفراً من نوع 8 إلى 2.

احتجنا لبناء الدائرة المنطقية هنا إلى ثلاثة دوائر متكاملة:

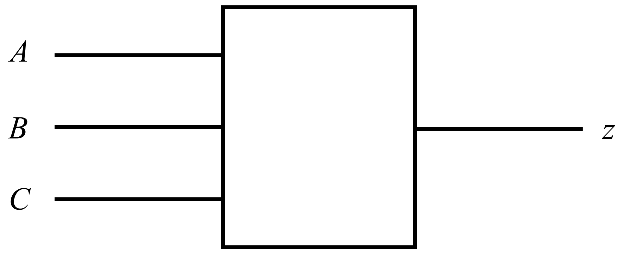
- مشفر من نوع 3 إلى 8

- بوابات OR بثلاثة مداخل
- بوابات OR بأربعة مداخل

و في واقع الأمر فإنه توجد دائرة متكاملة (.I.C) تحتوي على فاك الشفرة و المشفر معاً و هي ذاكرة ROM كما سنرى فيما بعد.

التصميم باستخدام الدامج (Multiplexer)

المطلوب تصميم الدائرة المنطقية ذات المخطط المنطقي و جدول الصواب أدناه باستخدام دامج



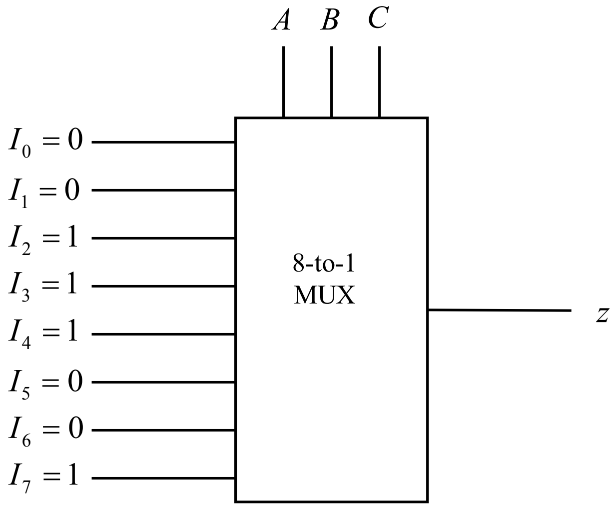
A	B	C	z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

خطوات التصميم:

1/ نحتاج إلى دامج عدد أطراف الاختيار (Select Lines) له يساوي عدد متغيرات الدخل للدائرة المنطقية المطلوب تصميمها، أي دامج من نوع 8 إلى 1 (لاحظ أن عدد أطراف الدخل للدامج هنا يكون مساوياً لعدد أسطر جدول الصواب).

2/ نقوم بإدخال متغيرات الدخل للدائرة المنطقية المطلوب تصميمها إلى أطراف الاختيار للدامج، مع مراعاة الترتيب.

3/ نأخذ قيم الخرج من جدول الصواب و نضعها بالترتيب على أطراف الدخل للدامج.
4/ نأخذ خرج الدائرة المنطقية من طرف الخرج للدامج.
كما هو موضح بالشكل التالي



<u>لاحظ</u> أننا قد احتجنا هنا لبناء الدائرة المنطقية إلى دائرة متكاملة واحدة فقط.

<u>لاحظ</u> أيضاً أن هذا الأسلوب في تصميم الدوائر المنطقية يصلح للدوائر ذات طرف الخرج الواحد، فإذا كان للدائرة أكثر من طرف خرج واحد نحتاج لدامج لكل طرف من أطراف الخرج.

التصميم باستخدام دامج عدد أطراف الإختيار له أقل من عدد متغيرات الدخل بواحد

المطلوب الآن تصميم نفس الدائرة المنطقية أعلاه و لكن باستخدام دامج من نوع 4 إلى 1.

خطوات التصميم:

1/ نحتاج إلى دامج عدد أطراف الإختيار (Select Lines) له أقل من عدد متغيرات الدخل للدائرة المنطقية المطلوب تصميمها بواحد، أي دامج من نوع 4 إلى 1 (لاحظ أن عدد أطراف الدخل للدامج هنا يكون مساوياً لنصف عدد أسطر جدول الصواب).

2/ نقوم بفصل متغير الدخل الأعلى A عن بقية متغيرات الدخل في جدول الصواب، مما يؤدي إلى تقسيم جدول الصواب إلى نصفين: النصف الأعلى و فيه $A = 0$، و النصف الأسفل و فيه $A = 1$

A	B	C	z
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

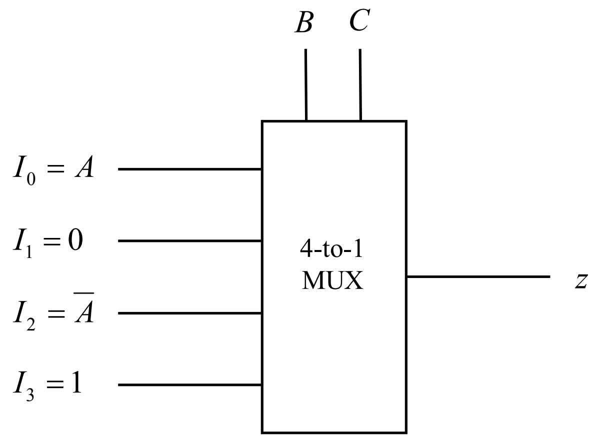
3/ نقوم بمقارنة قيمة متغير الخرج z في كل سطر من أسطر النصف الأعلى من جدول الصواب مع السطر المقابل له من أسطر النصف الأسفل من جدول الصواب، و ذلك لإيجاد علاقة مابين متغير الخرج z و متغير الدخل الأعلى A. فبمقارنة السطر الأول من النصف الأعلى مع السطر الأول من النصف الأسفل نجد أن $z = A$، و بمقارنة السطر الثاني من النصف الأعلى مع السطر الثاني من النصف الأسفل نجد أن $z = 0$، و بمقارنة السطر الثالث من النصف الأعلى مع السطر الثالث من النصف الأسفل نجد أن $z = \overline{A}$، و بمقارنة السطر الرابع من النصف الأعلى مع السطر الرابع من النصف الأسفل نجد أن $z = 1$.

4/ نقوم بإدخال متغيرات الدخل الدنيا للدائرة المنطقية المطلوب تصميمها إلى أطراف الإختيار للدامج، مع مراعاة الترتيب.

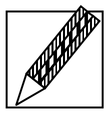
5/ نقوم بإدخال العلاقات التي توصلنا إليها في الخطوة 3 بالترتيب إلى أطراف الدخل للدامج.

6/ نأخذ خرج الدائرة المنطقية من طرف الخرج للدامج.

كما هو موضح بالشكل التالي:



تدريب 17



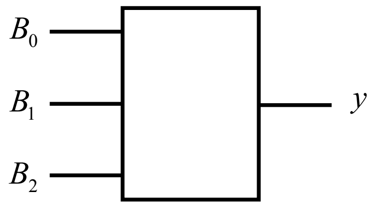
صمم الدائرة المنطقية التي يوضحها المخطط المنطقي وجدول الصواب لها أدناه، و ذلك باستخدام:

(أ)وحدة فك الشفرة و مشفر (Decoder & Encoder).

(ب) دامج من نوع 8 إلى 1 (8-to-1 MUX).

(ج) دامج من نوع 4 إلى 1 (4-to-1 MUX).

B_2	B_1	B_0	y
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



الخلاصة

تعرفنا في هذه الوحدة على بعض الدوائر المنطقية الترابطية التي تقوم بـــأداء وظـــائف مفيدة، و التي يشيع استخدامها في الأنظمة الرقمية، كما تعرفنا على أبـــرز اســـتخدامات بوابة XOR في الدوائر المنطقية باستخدام العمليات الأساسية الثلاث كالتالي :

• العاكس المنطقي المحكوم حيث يقوم بعملية العكس المنطقي للمتغير الداخل إليه ويقوم بإجرائها حسب القيمة المنطقية التي نقوم بوضعها في طرف التحكم الخاص به.

• التحويل من الشفرة الثنائية إلى الشفرة الرمادية هنا يمكن كتابة التعبيـــرات المنطقيـــة بدلالة عملية XOR .

• دوائر التحقق وهي عملية تستخدم لاكتشاف حدوث خطأ في البيانات المنقولة ،وتتقســـم الى دوائر توليد خانة التحقق الفردي ، والتحقق الزوجي.

وفي قسم آخر من الوحدة تناولنا دوائر الجمع وهي دوائر منطقية تقوم بـــإجراء عمليـــة جمع الأعداد المختلفة في الصورة الثنائية نأخذ منها:

- نصف الجامع، تقوم بجمع خانتين ثنائيتين إلى بعضهما البعض وإيجاد حاصـــل الجمع.
- الجامع الكامل، تقوم بإجراء عملية الجمع وإيجاد كل مـــن المجمـــوع والحمـــل الخارج. ولها دخل ثالث هو الحمل الداخل.
- الجامع المتعدد الخانات، يقوم بإجراء عملية جمع عددين ثنائيين يتكون كل منهما من أربعة خانات.

وتعرفنا في هذا القسم على ربط الجوامع وذلك بربط وحدات جامع صغيرة لبناء جـــامع أكبر ، ثم انتقلنا بعد ذلك الى علمية الطرح وذلك بتحويل عملية الطرح إلى عملية جمـــع مع سالب العدد المطروح ، وفي وحدة الحساب عرفنا انه يمكننا إجـــراء المزيـــد مـــن العمليات الحسابية باستخدام دائرة الجامع / الطارح ذي الأربعة خانات إذا أضفنا اليهـــا إمكانية تصفير أحد العددين Aأو B .

وفي القسم الرابع من الوحدة تحدثنا عن وحدة فك الشفرة والذي هو عبارة عـن دائـرة منطقية لها عدة أطراف خرج. واحد فقط من أطراف الخرج هذه يكون نشطاً. وعرفنـا في هذا القسم أيضاً خط السماح والذي يكون عادة موجوداً في وحدة فك الشـفرة، وهـو عبارة عن طرف تحكم يمكن بواسطته أن تبطل عمل الـدائرة أو تسـمح لهـا بالعمـل كالمعتاد . انتقلنا بعد ذلك الى الاستخدام الأساسي لفاك الشفرة وهو استخدامه في دوائـر الذاكرة.

في القسم الخامس عرفنا المشفر و هو الذي يؤدي عكس الوظيفة التي يؤديها فك الشفرة وهو دائرة منطقية لها عدة أطراف دخل. ويكون واحد فقط من أطراف الدخل هذه نشطاً أي مساوياً 1 . خرج الدائرة عبارة عن شفرة (Code) تمثل طرف الدخل النشط.

القسم السادس تناول الدامج والذي هو عبارة عن دائرة منطقية لها عدة أطـراف دخـل وطرف خرج واحد ، وفي بعض الأحيان يكون الدامج مزوداً بخط السـماح ووظيفتـه إبطال عمل الدائرة أو السماح لها بأن تؤدي وظيفتها كالمعتاد . انتقلنا بعد ذلك إلى ربـط الدوامج حيث يمكن ربط عدد من وحدات الدامج الصغيرة لبناء وحدة دامج أكبر.

في القسم السابع تناولنا المفرق الذي يؤدي عكس الوظيفة التي يؤديهـا الـدامج وهـو عبارة عن دائرة منطقية لها عدة أطراف خرج وطرف دخل واحد ، يتم توصيل طـرف الدخل مع أحد أطراف الخرج.

في القسم الأخير من الوحدة تناولنا طرق بديلة لتصميم الدوائر المنطقية وهـي تهـدف بالأساس إلى تقليل عدد الدوائر المتكاملة (IC's) المستخدمة في بناء الدائرة المنطقية بدلاً عن تقليل عدد البوابات المنطقية المستخدمة في بناء الدائرة المنطقية وهنا نستعين بدوائر منطقية جاهزة مثل فاك الشفرة والدامج.

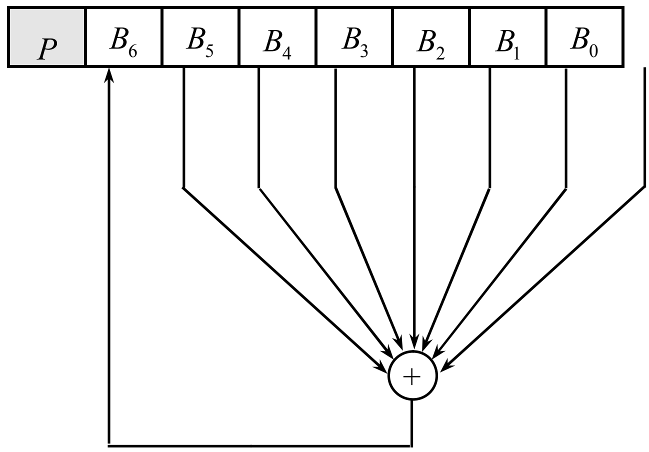
لمحة مسبقة عن الوحدة التالية

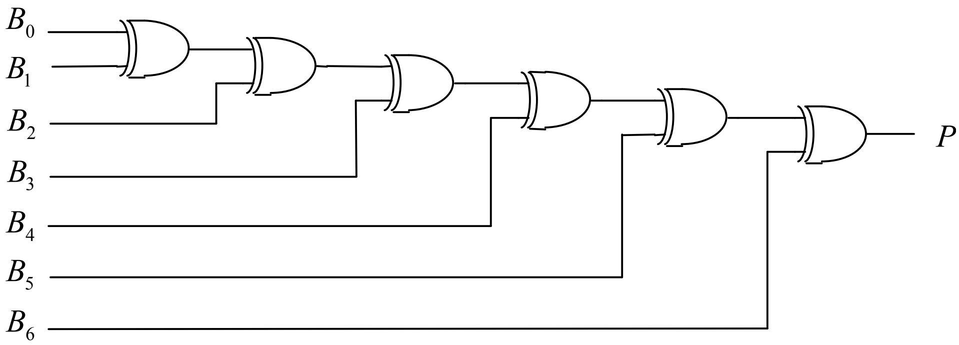
نعلم الآن أن جميع الدوائر المنطقية التي تعاملنا معها حتى الآن هـي دوائـر منطقيـة ترابطية (Combinational)، و أنها سميت بالترابطية لأن وظيفة الدائرة تقتصـر علـى ربط متغيرات الدخل بعمليات منطقية لتوليد متغيرات الخرج. كما نعلم أن الخرج فـي الدوائر الترابطية يعتمد فقط على القيم الحالية للدخل، فمتى ما تغيرت قيم الدخل تغيـرت معها قيم الخرج. في الوحدة التالية من المقرر سنتعرف على النوع الآخر من الـدوائر المنطقية و هو الدوائر المنطقية التتابعية (Sequential)، مثل المـراجيح (Flip Flops) و المسجلات (Registers) و العدّادات (Counters). حيث سنعرف أن الخـرج فـي هـذا النوع من الدوائر المنطقية لا يعتمد فقط على القيم الحالية للدخل، و إنما يعتمـد أيضـاً على القيم السابقة للخرج. أي أن هذا النوع من الدوائر له ذاكـرة (Memory) تسـتطيع اختزان ماضي الدائرة بحيث يؤثر على خرجها الحالي.

إجابات التدريبات

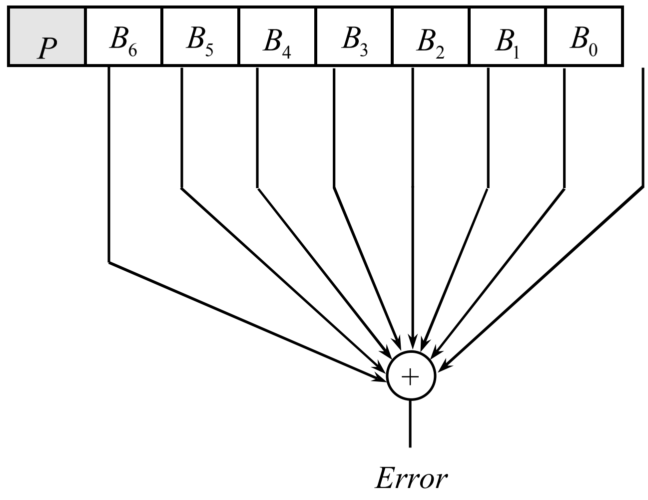
تدريب 1:

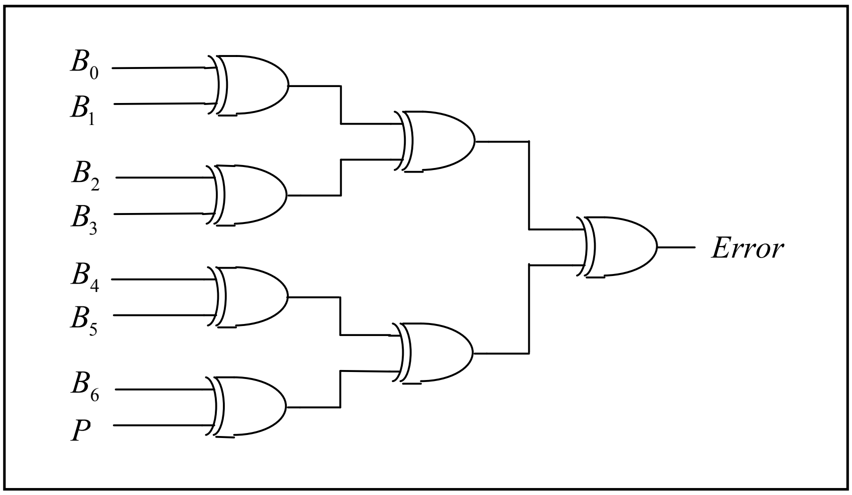
دائرة توليد خانة التحقق الزوجي (Even Parity bit Generator):





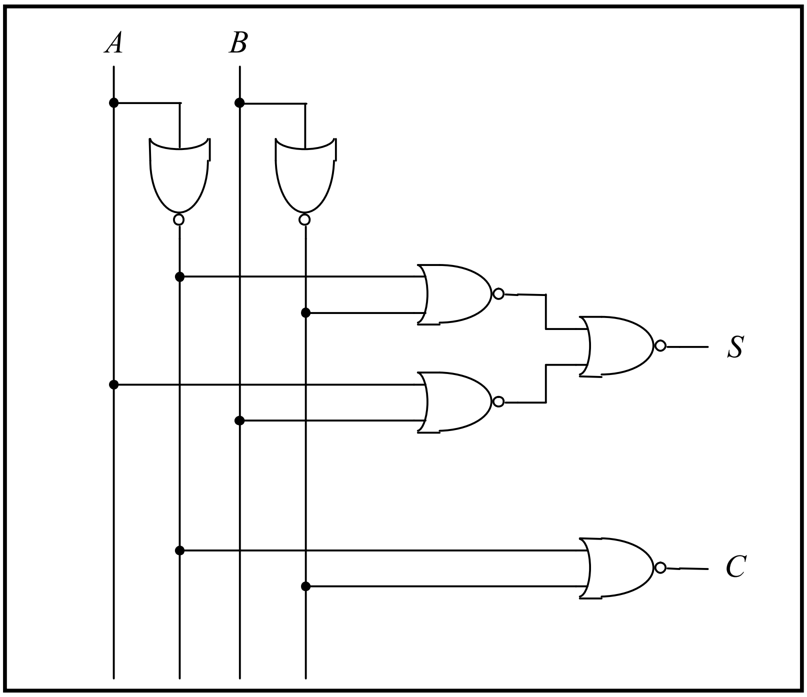
دائرة التحقق الزوجي (Even Parity Checker):



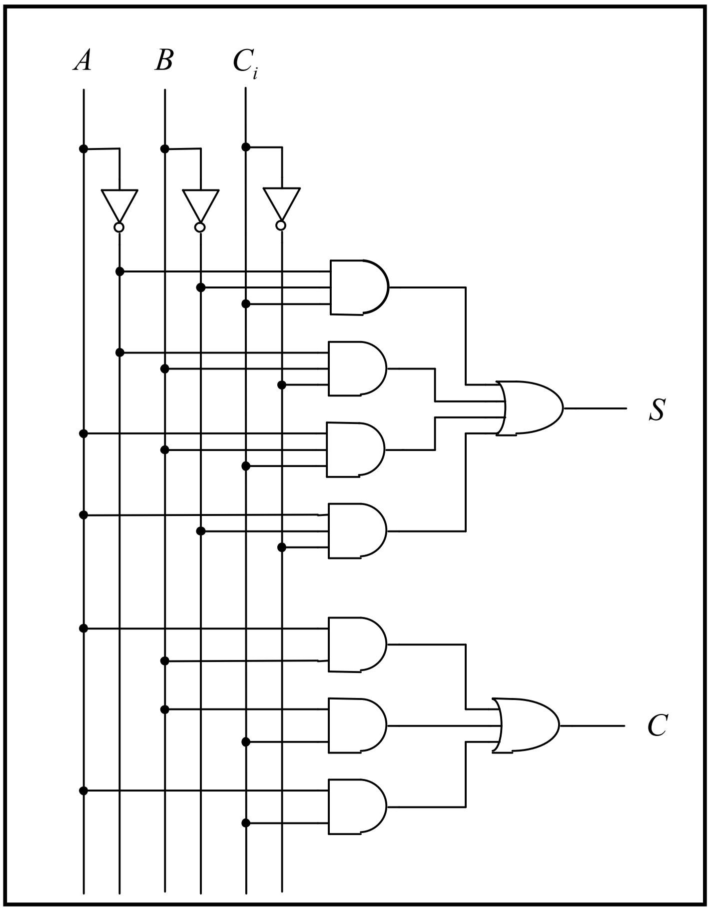


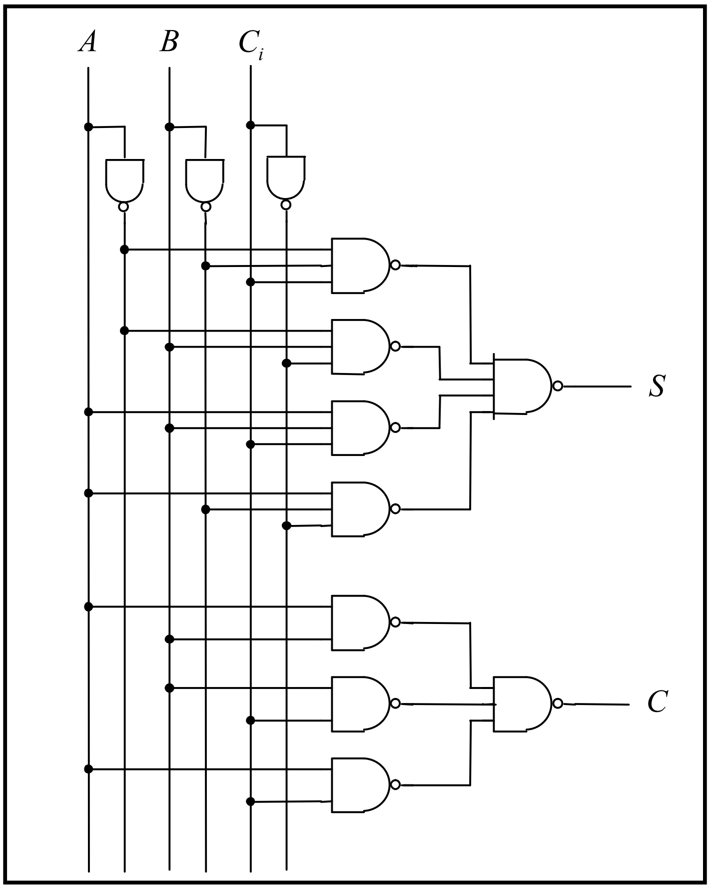
إذا كان $Error = 1$ فإن هنالك خطأ في الرمز، أما إذا كان $Error = 0$ فلا يوجد خطأ.

تدريب 2:

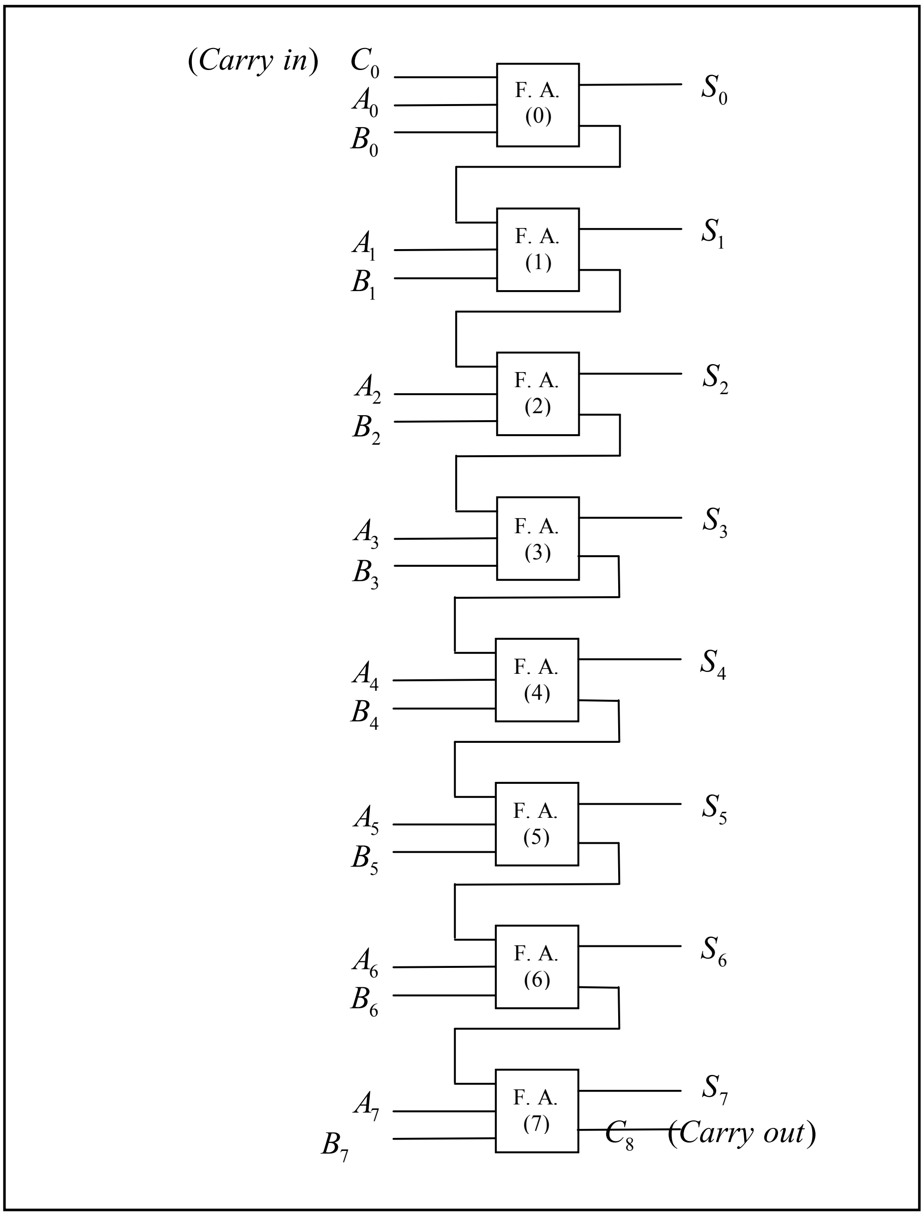


تدريب **3**:



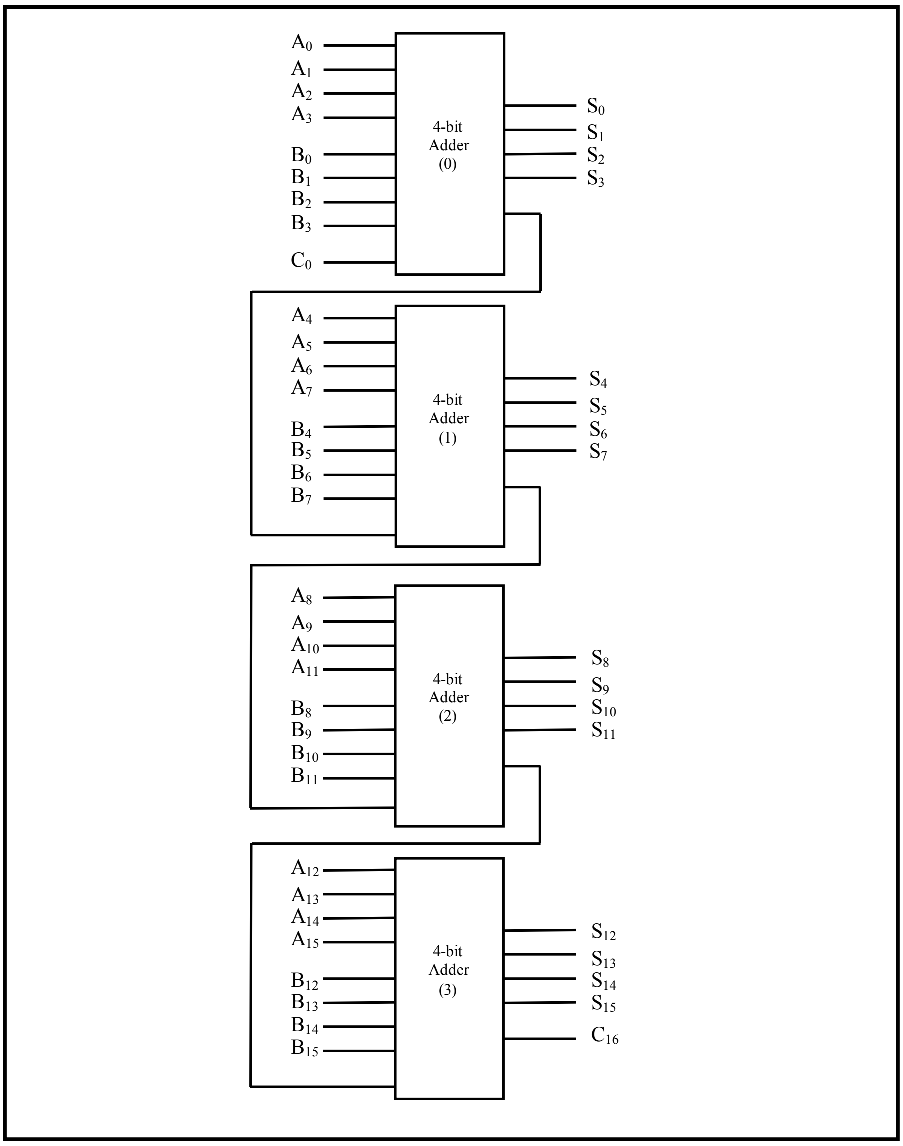

A
B
C_i
S
C

تدريب 4:



(Carry in) C_0
A_0
B_0
F. A. (0)
S_0
A_1
B_1
F. A. (1)
S_1
A_2
B_2
F. A. (2)
S_2
A_3
B_3
F. A. (3)
S_3
A_4
B_4
F. A. (4)
S_4
A_5
B_5
F. A. (5)
S_5
A_6
B_6
F. A. (6)
S_6
A_7
B_7
F. A. (7)
S_7
C_8 (Carry out)

تدريب 5:



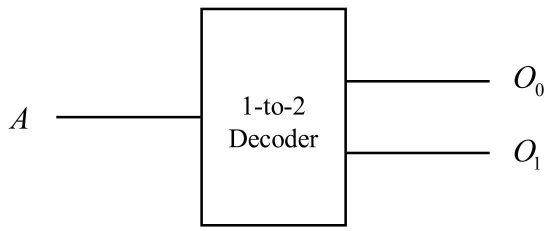
A0
A1
A2
A3
B0
B1
B2
B3
C0
4-bit Adder (0)
S0
S1
S2
S3
A4
A5
A6
A7
B4
B5
B6
B7
4-bit Adder (1)
S4
S5
S6
S7
A8
A9
A10
A11
B8
B9
B10
B11
4-bit Adder (2)
S8
S9
S10
S11
A12
A13
A14
A15
B12
B13
B14
B15
4-bit Adder (3)
S12
S13
S14
S15
C16

تدريب 6:

G_4	G_3	G_2	G_1	G_0	Operation
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	-1
0	0	0	1	1	0
0	0	1	0	0	-1
0	0	1	0	1	0
0	0	1	1	0	-2
0	0	1	1	1	-1
0	1	0	0	0	A
0	1	0	0	1	$A+1=A++$
0	1	0	1	0	$-A-1=\overline{A}$
0	1	0	1	1	$-A$
0	1	1	0	0	$A-1=A--$
0	1	1	0	1	A
0	1	1	1	0	$-A-2$
0	1	1	1	1	$-A-1=\overline{A}$
1	0	0	0	0	B
1	0	0	0	1	$B+1=B++$
1	0	0	1	0	$B-1=B--$
1	0	0	1	1	B
1	0	1	0	0	$-B-1=\overline{B}$
1	0	1	0	1	$-B$
1	0	1	1	0	$-B-2$
1	0	1	1	1	$-B-1=\overline{B}$
1	1	0	0	0	$A+B$
1	1	0	0	1	$A+B+1$
1	1	0	1	0	$B-A-1$
1	1	0	1	1	$B-A$
1	1	1	0	0	$A-B-1$
1	1	1	0	1	$A-B$
1	1	1	1	0	$-A-B-2$
1	1	1	1	1	$-A-B-1$

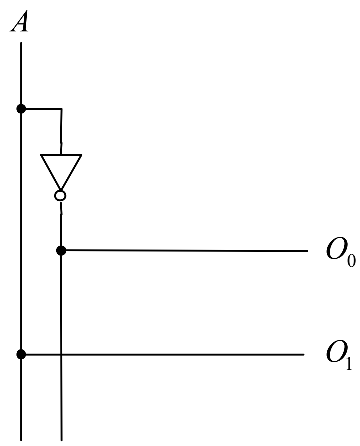
تدريب 7:

(أ) وحدة فك شفرة من نوع 1 إلى 2 (1-to-2 Decoder)

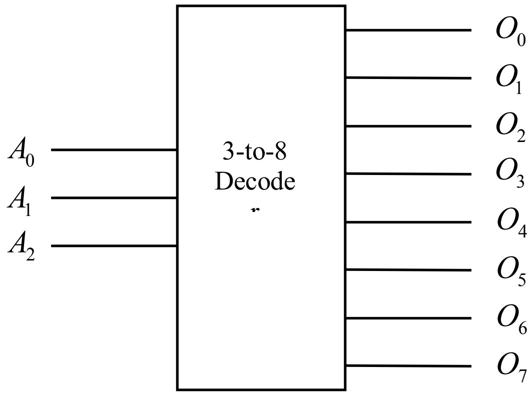


A	O_1	O_0
0	0	1
1	1	0

$$O_0 = \overline{A}$$
$$O_1 = A$$

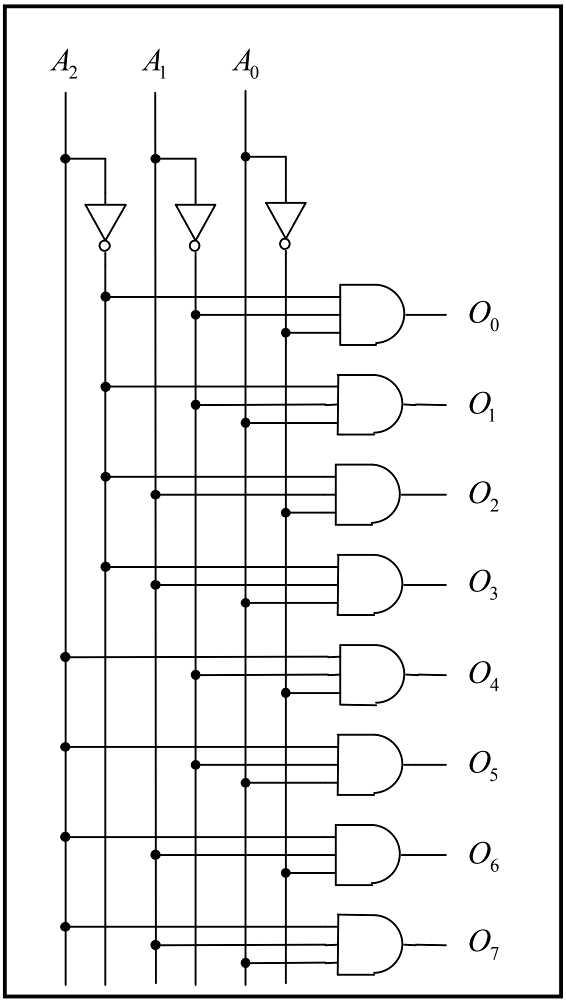


(ب) وحدة فك شفرة من نوع 3 إلى 8 (3-to-8 decoder)

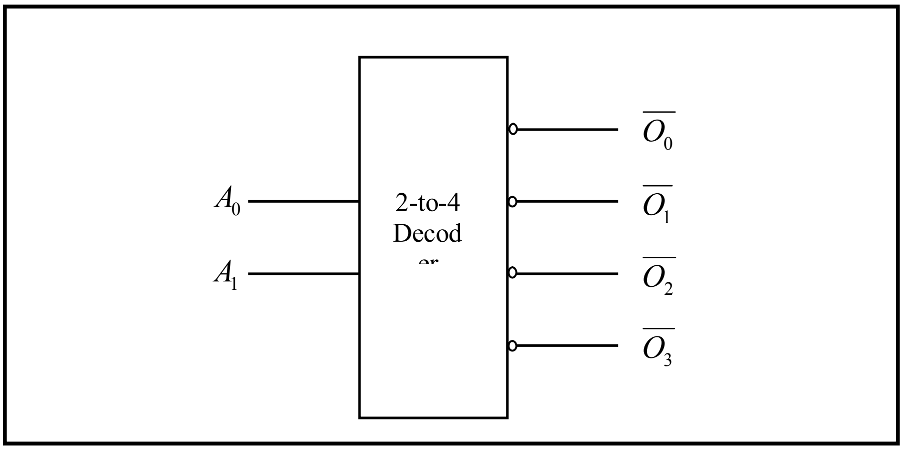


A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

$$O_0 = \overline{A_2}\,\overline{A_1}\,\overline{A_0}$$
$$O_1 = \overline{A_2}\,\overline{A_1}A_0$$
$$O_2 = \overline{A_2}A_1\overline{A_0}$$
$$O_3 = \overline{A_2}A_1A_0$$
$$O_4 = A_2\overline{A_1}\,\overline{A_0}$$
$$O_5 = A_2\overline{A_1}A_0$$
$$O_6 = A_2A_1\overline{A_0}$$
$$O_7 = A_2A_1A_0$$


A_2
A_1
A_0
O_0
O_1
O_2
O_3
O_4
O_5
O_6
O_7

تدريب 8:


A_0
A_1
2-to-4
Decod
$\overline{O_0}$
$\overline{O_1}$
$\overline{O_2}$
$\overline{O_3}$

A_1	A_0	$\overline{O_3}$	$\overline{O_2}$	$\overline{O_1}$	$\overline{O_0}$
0	0	1	1	1	0
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	1	1	1

$$\overline{O_0} = \overline{\overline{A_1}\,\overline{A_0}}$$

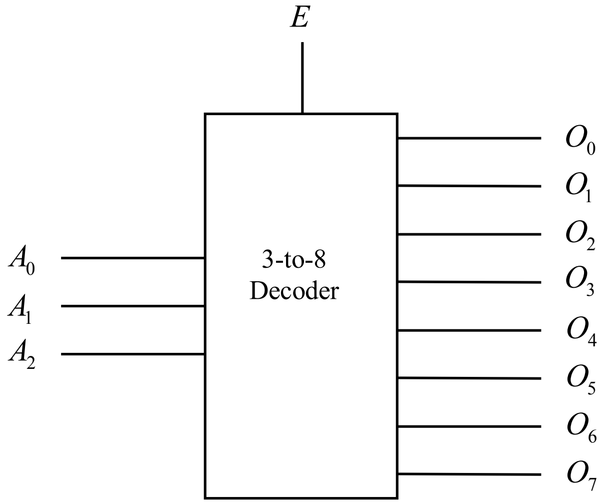
$$\overline{O_1} = \overline{\overline{A_1}A_0}$$

$$\overline{O_2} = \overline{A_1\overline{A_0}}$$

$$\overline{O_3} = \overline{A_1A_0}$$

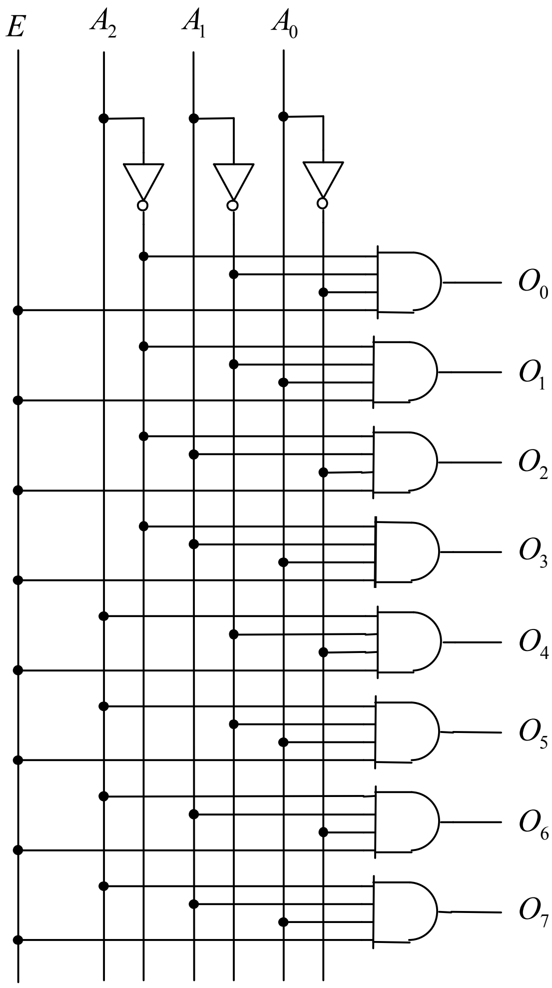
تدريب 9:

(أ)وحدة شفرة من نوع 3 إلى 8 بخط سماح (3-to-8 Decoder with Enable)

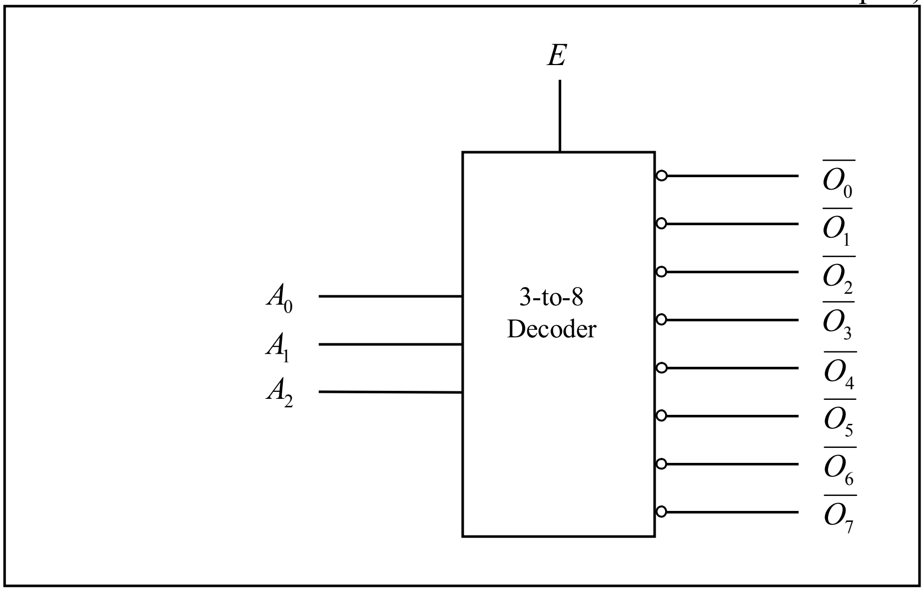


E	A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2		O_0
0	×	×	×	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

$$O_0 = E\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}$$
$$O_1 = E\,\overline{A_2}\,\overline{A_1}\,A_0$$
$$O_2 = E\,\overline{A_2}\,A_1\,\overline{A_0}$$
$$O_3 = E\,\overline{A_2}\,A_1\,A_0$$
$$O_4 = E\,A_2\,\overline{A_1}\,\overline{A_0}$$
$$O_5 = E\,A_2\,\overline{A_1}\,A_0$$
$$O_6 = E\,A_2\,A_1\,\overline{A_0}$$
$$O_7 = E\,A_2\,A_1\,A_0$$

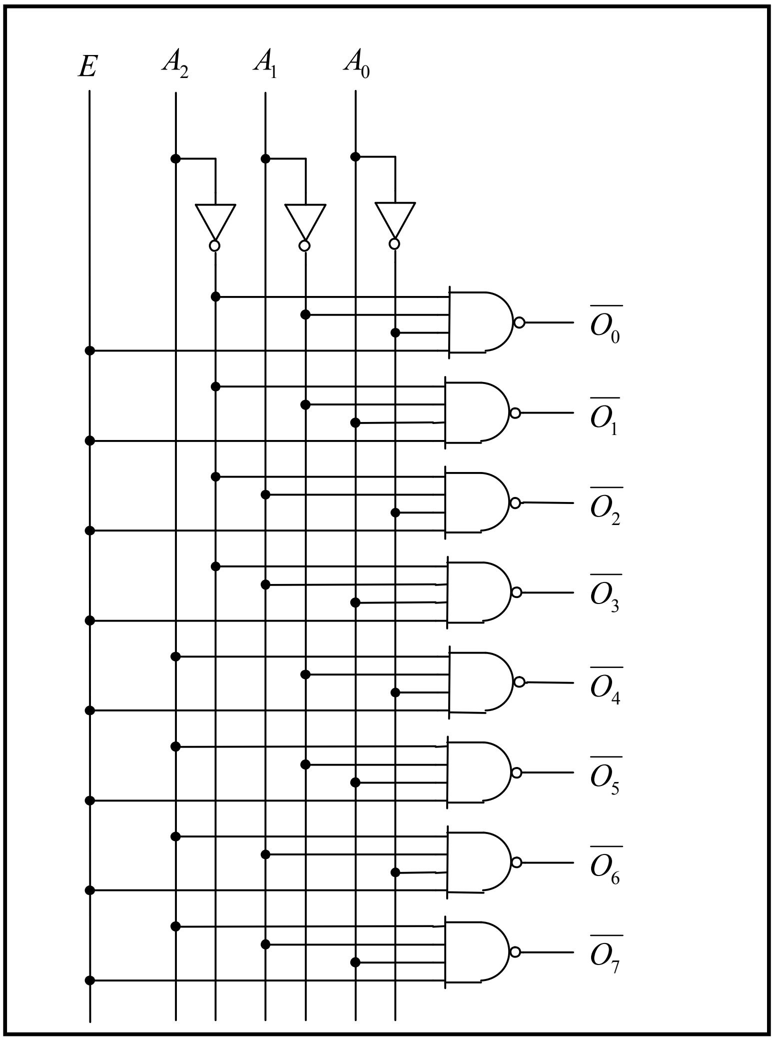


(ب) وحدة شفرة من نوع 3 إلى 8 بخط سماح و خرج نشط منخفض (3-to-8 Decoder
with Enable and Active Low Outputs)

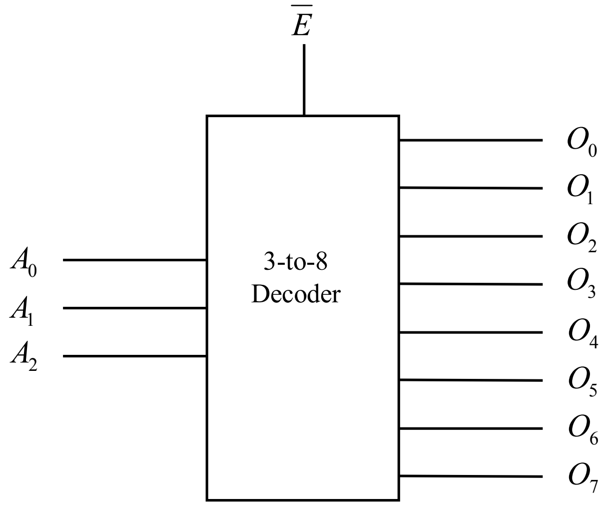


E	A_2	A_1	A_0	$\overline{O_7}$	$\overline{O_6}$	$\overline{O_5}$	$\overline{O_4}$	$\overline{O_3}$	$\overline{O_2}$	$\overline{O_1}$	$\overline{O_0}$
0	×	×	×	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	0
1	0	0	1	1	1	1	1	1	1	0	1
1	0	1	0	1	1	1	1	1	0	1	1
1	0	1	1	1	1	1	1	耰	1	1	1
1	1	0	0	1	1	1	0	1	1	1	1
1	1	0	1	1	1	0	1	1	1	1	1
1	1	1	0	1	0	1	1	1	1	1	1
1	1	1	1	0	1	1	1	1	1	1	1

$$\overline{O_0} = \overline{E\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}}$$
$$\overline{O_1} = \overline{E\,\overline{A_2}\,\overline{A_1}\,A_0}$$
$$\overline{O_2} = \overline{E\,\overline{A_2}\,A_1\,\overline{A_0}}$$
$$\overline{O_3} = \overline{E\,\overline{A_2}\,A_1\,A_0}$$
$$\overline{O_4} = \overline{E\,A_2\,\overline{A_1}\,\overline{A_0}}$$
$$\overline{O_5} = \overline{E\,A_2\,\overline{A_1}\,A_0}$$
$$\overline{O_6} = \overline{E\,A_2\,A_1\,\overline{A_0}}$$
$$\overline{O_7} = \overline{E\,A_2\,A_1\,A_0}$$

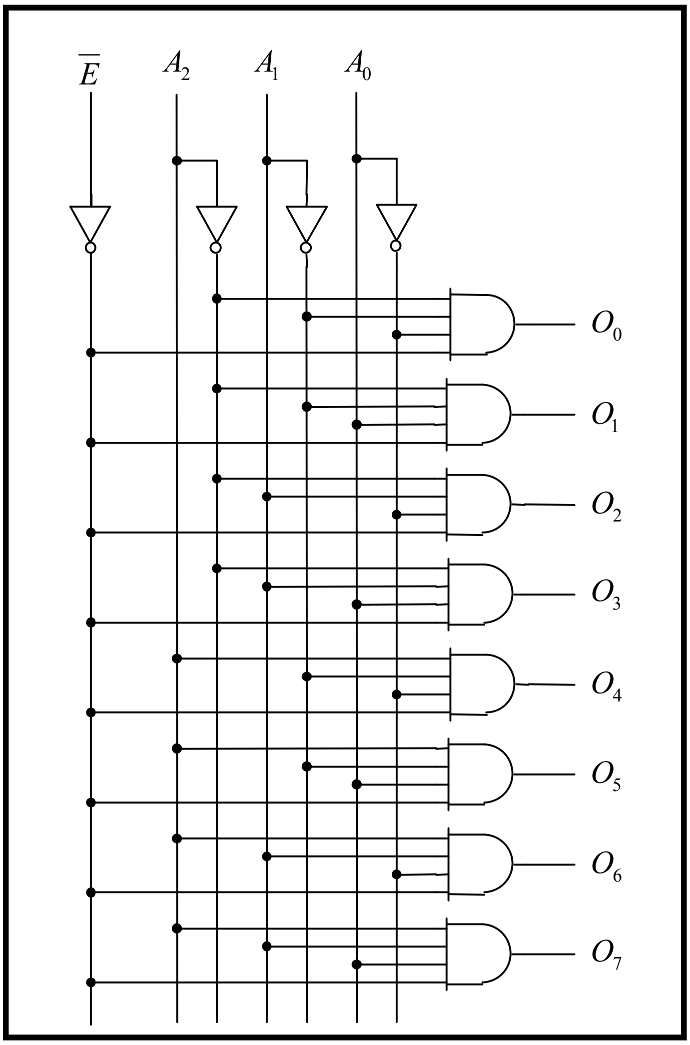


(ج)وحدة فك شفرة من نوع 3 إلى 8 بخط سماح نشط منخفض (3-to-8 Decoder with Active Low Enable)



E	A_2	A_1	A_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	0	0	1	0
0	0	1	0	0	0	0	0	0	1	0	0
0	0	1	1	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	1	0	0	0	0
0	1	0	1	0	0	1	0	0	0	0	0
0	1	1	0	0	1	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0
1	×	×	×	0	0	0	0	0	0	0	0

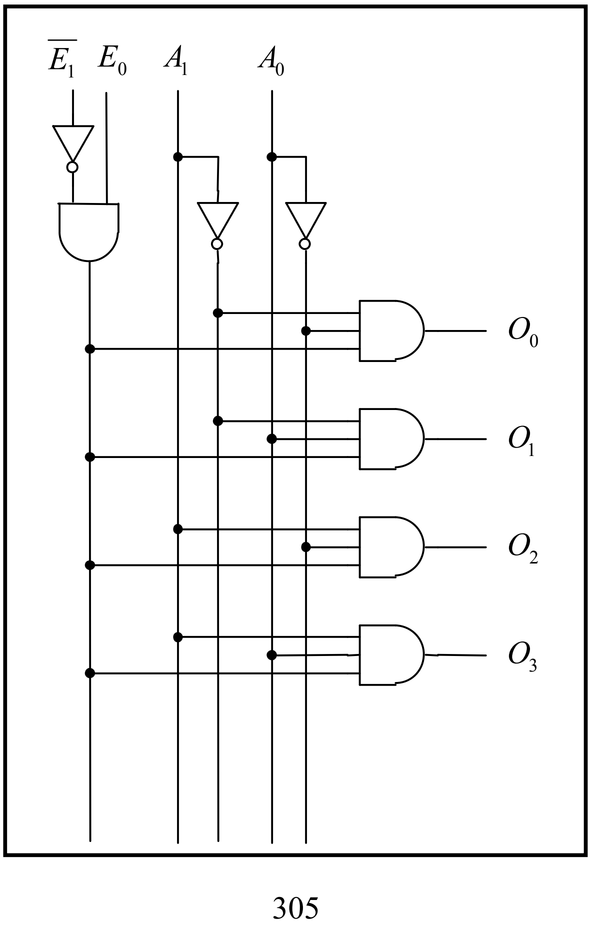
$$O_0 = \overline{E}\,\overline{A_2}\,\overline{A_1}\,\overline{A_0}$$
$$O_1 = \overline{E}\,\overline{A_2}\,\overline{A_1}A_0$$
$$O_2 = \overline{E}\,\overline{A_2}A_1\overline{A_0}$$
$$O_3 = \overline{E}\,\overline{A_2}A_1A_0$$
$$O_4 = \overline{E}\,A_2\overline{A_1}\,\overline{A_0}$$
$$O_5 = \overline{E}\,A_2\overline{A_1}A_0$$
$$O_6 = \overline{E}\,A_2A_1\overline{A_0}$$
$$O_7 = \overline{E}\,A_2A_1A_0$$



تدريب 10:

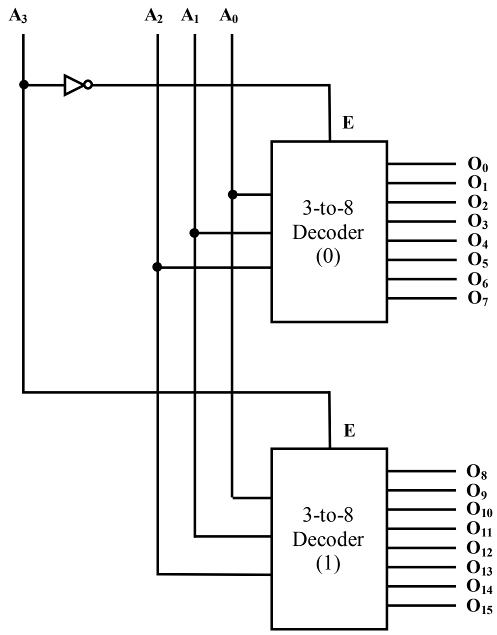
E_1	E_0	A_1	A_0	O_3	O_2	O_1	O_0
0	0	×	×	0	0	0	0
0	1	0	0	0	0	0	1
0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0
0	1	1	1	1	0	0	0
1	0	×	×	0	0	0	0
1	1	×	×	0	0	0	0

$$O_0 = \overline{E_1} E_0 \overline{A_1}\,\overline{A_0}$$
$$O_1 = \overline{E_1} E_0 \overline{A_1} A_0$$
$$O_2 = \overline{E_1} E_0 A_1 \overline{A_0}$$
$$O_3 = \overline{E_1} E_0 A_1 A_0$$

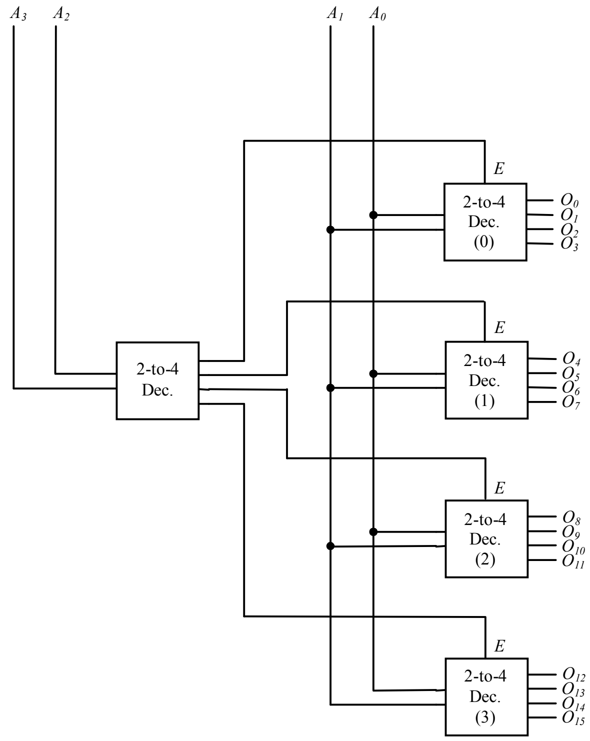


تدريب 11:

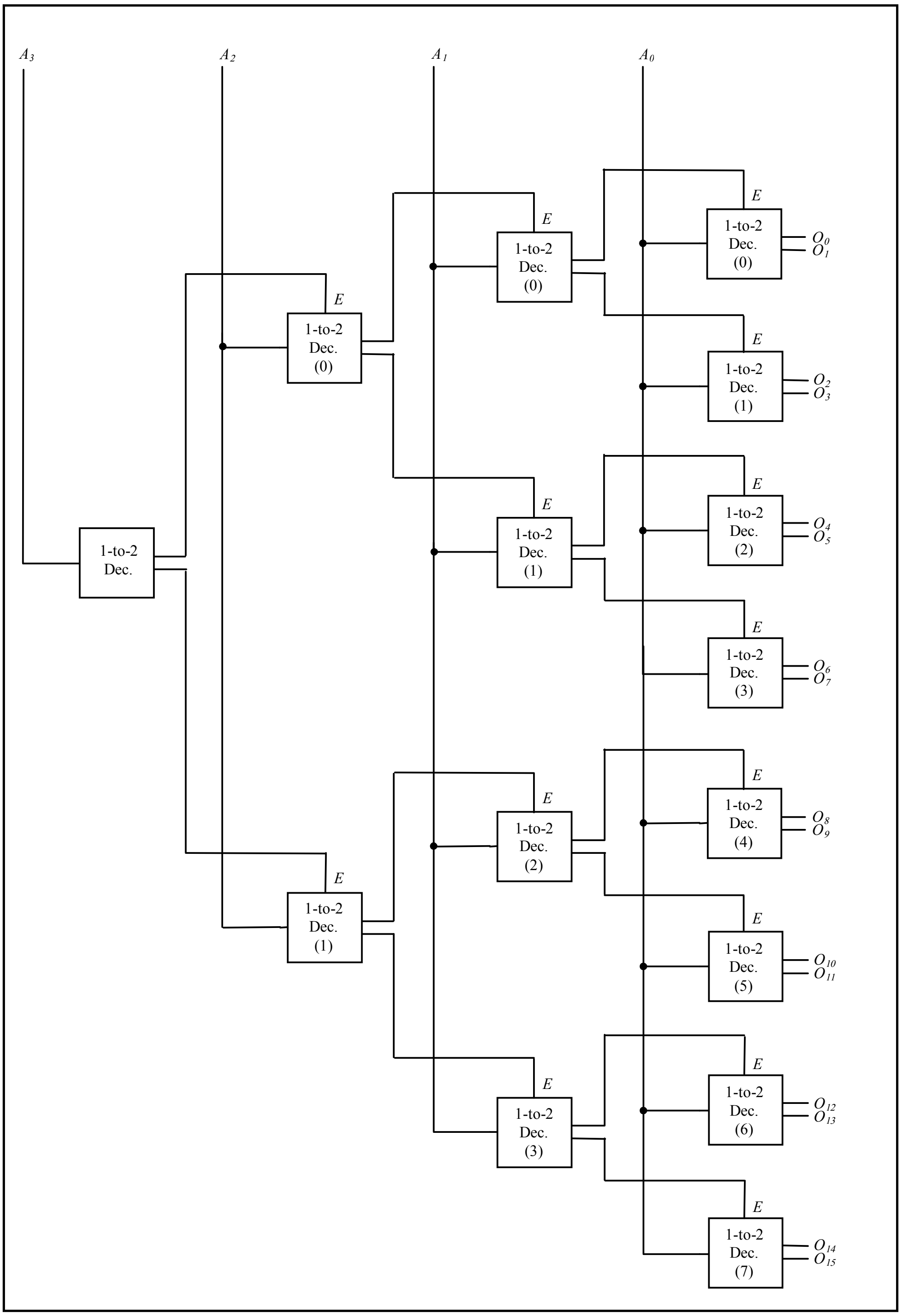
(أ) بناء وحدة فك شفرة من نوع 4 إلى 16 باستخدام دوائر وحدة فك شفرة من نوع 3 إلى 8



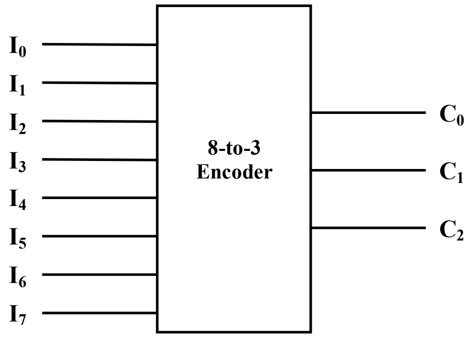
(ب) بناء وحدة فك شفرة من نوع 4 إلى 16 باستخدام دوائر وحدة فك شفرة من نوع 2 إلى 4



(ج) بناء فاك شفرة من نوع 4 إلى 16 باستخدام دوائر فاك شفرة من نوع 1 إلى 2

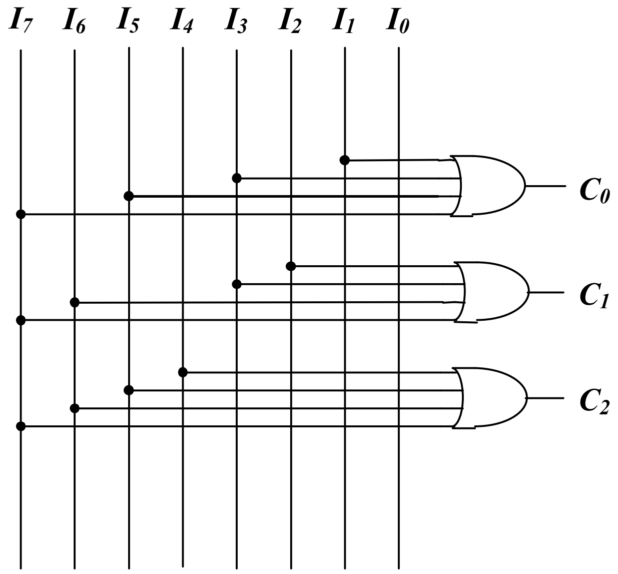


تدريب 12



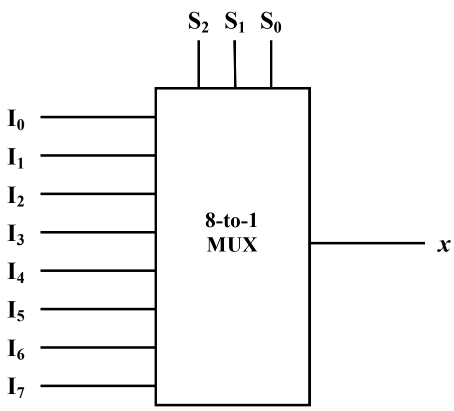
I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	C_2	C_1	C_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

$$C_0 = I_1 + I_3 + I_5 + I_7$$
$$C_1 = I_2 + I_3 + I_6 + I_7$$
$$C_2 = I_4 + I_5 + I_6 + I_7$$

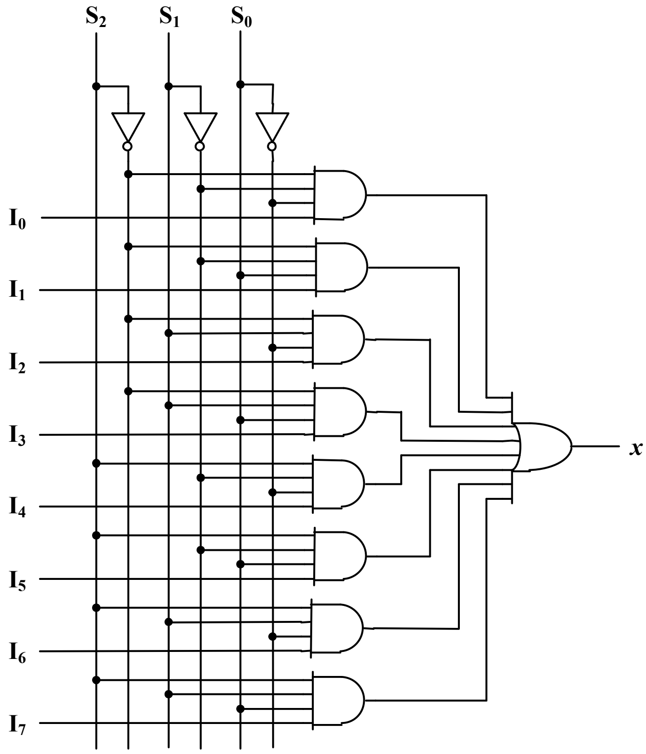


تدريب 13:

S_2	S_1	S_0	x
0	0	0	I_0
0	0	1	I_1
0	1	0	I_2
0	1	1	I_3
1	0	0	I_4
1	0	1	I_5
1	1	0	I_6
1	1	1	I_7



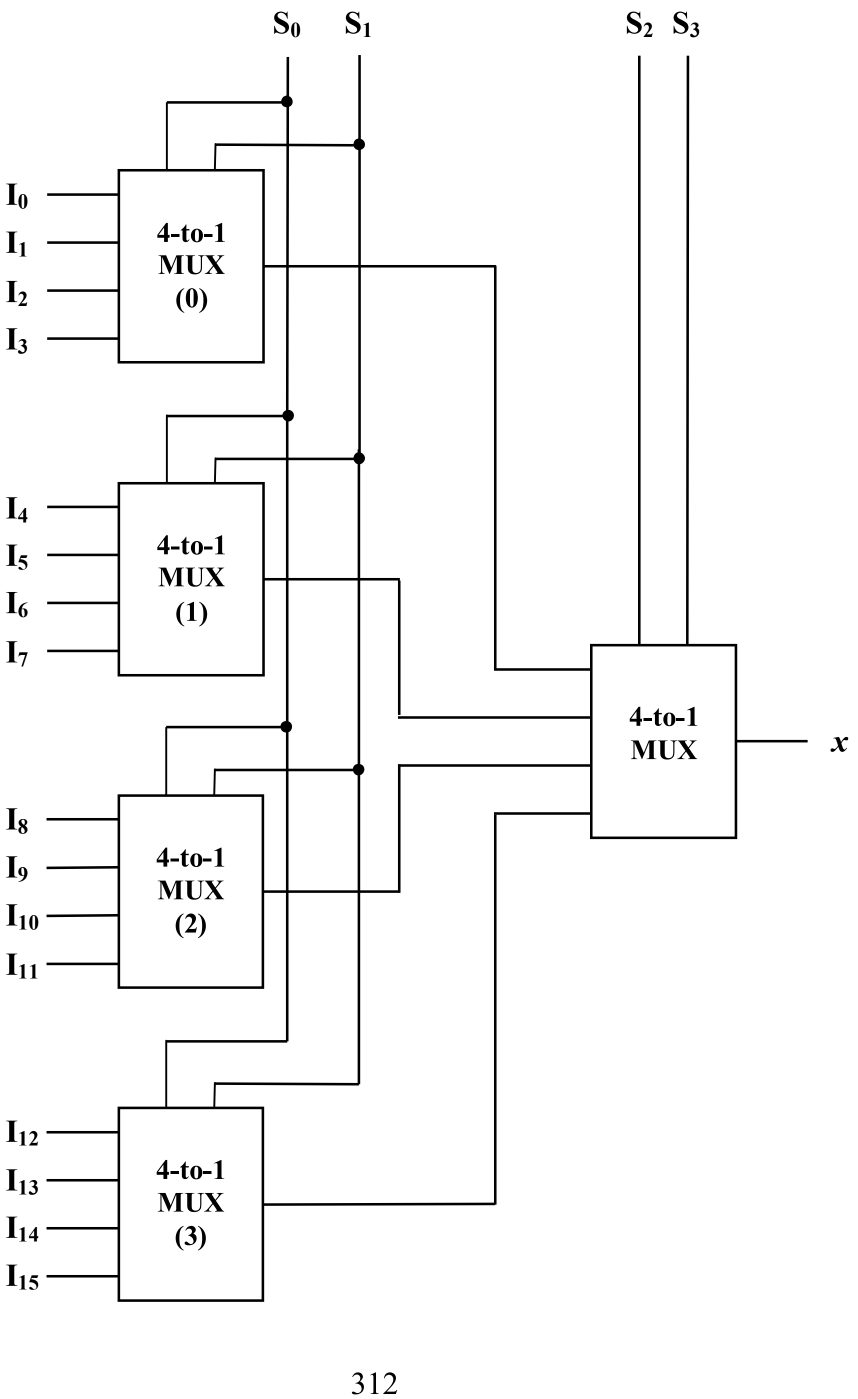
$$x = \overline{S_2}\,\overline{S_1}\,\overline{S_0}I_0 + \overline{S_2}\,\overline{S_1}S_0I_1 + \overline{S_2}S_1\overline{S_0}I_2 + \overline{S_2}S_1S_0I_3 + S_2\overline{S_1}\,\overline{S_0}I_4 + S_2\overline{S_1}S_0I_5 + S_2S_1\overline{S_0}I_6 + S_2S_1S_0I_7$$



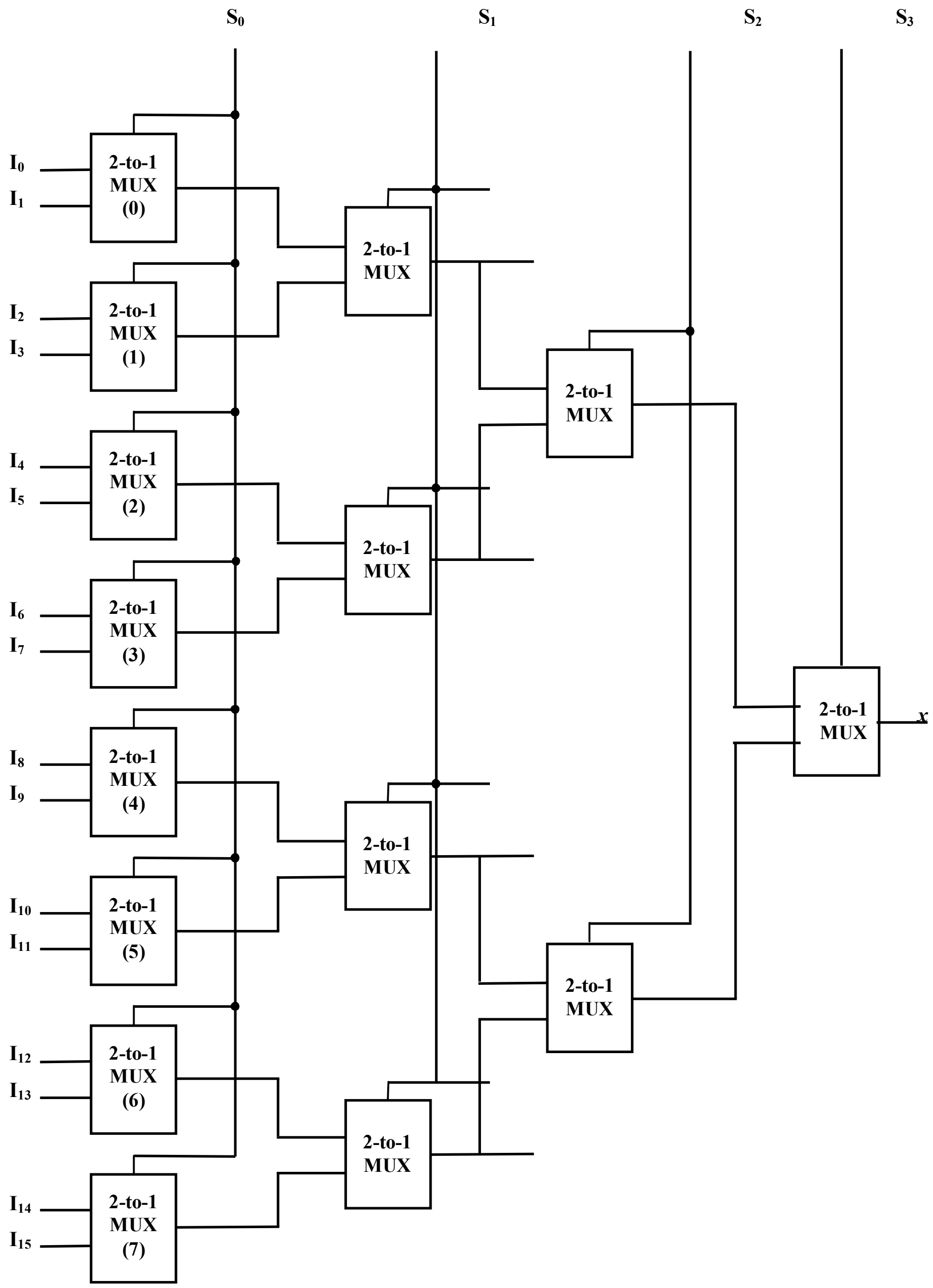
S2
S1
S0
I0
I1
I2
I3
I4
I5
I6
I7
x

تدريب **14**:

بناء برامج من نوع 1 إلى 4 باستخدام وحدات دامج من نوع 1 إلى 16تقديم.



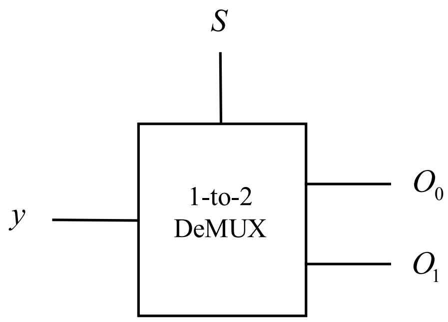
(ب) بناء دامج من نوع 16 إلى 1 باستخدام وحدات دامج من نوع 2 إلى 1



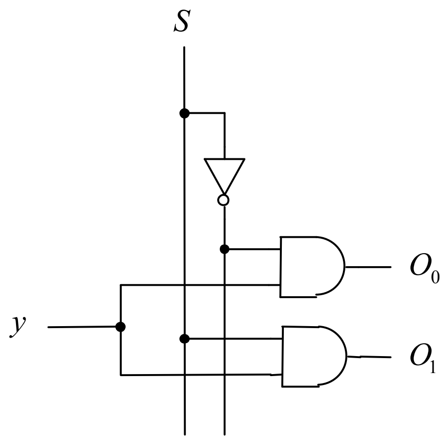
تدريب 15:

(أ) مفرق من نوع 1 إلى 2 (1-to-2 Demultiplexer)

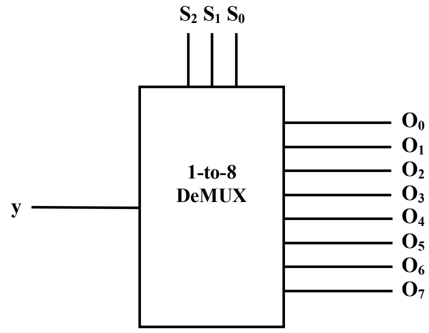
S	O_1	O_0
0	0	y
1	y	0



$$O_0 = \overline{S}y$$
$$O_1 = Sy$$

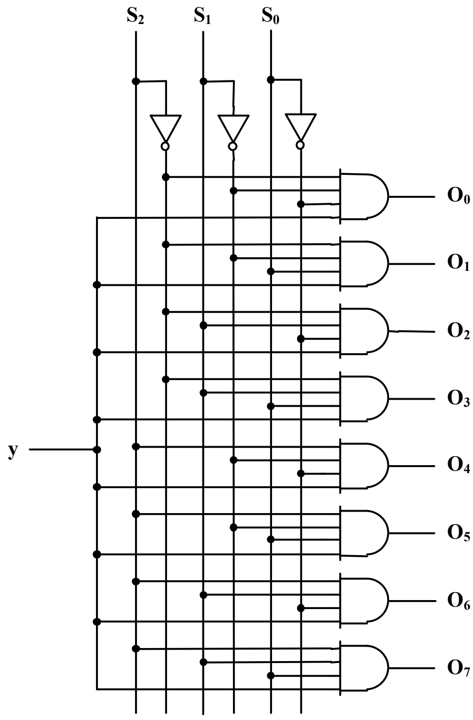


(ب) مفرق من نوع 1 إلى 8 (1-to-8 Demultiplexer)

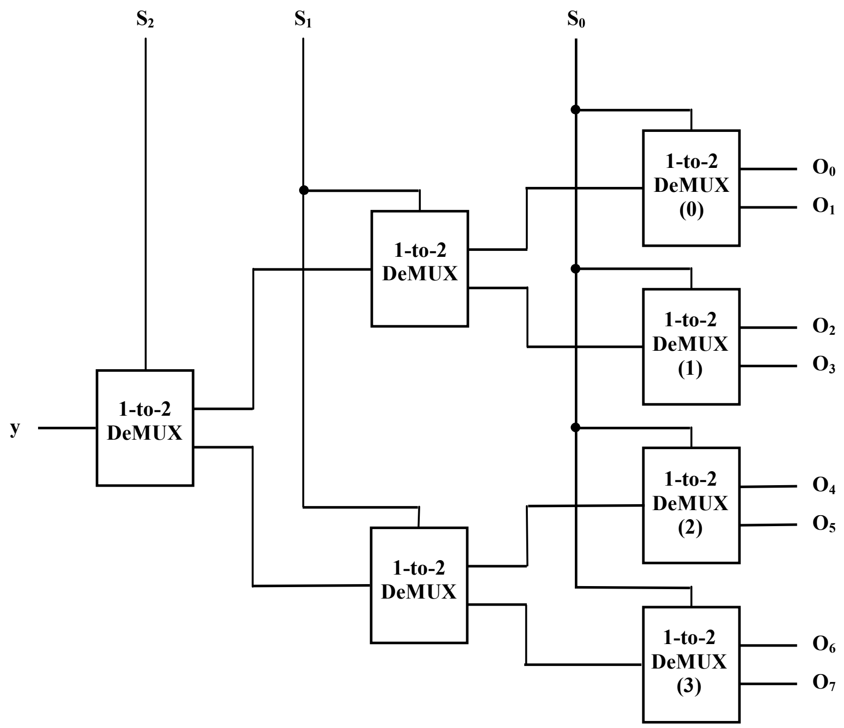


S_2	S_1	S_0	O_7	O_6	O_5	O_4	O_3	O_2	O_1	O_0
0	0	0	0	0	0	0	0	0	0	y
0	0	1	0	0	0	0	0	0	y	0
0	1	0	0	0	0	0	0	y	0	0
0	1	1	0	0	0	0	y	0	0	0
1	0	0	0	0	0	y	0	0	0	0
1	0	1	0	0	y	0	0	0	0	0
1	1	0	0	y	0	0	0	0	0	0
1	1	1	y	0	0	0	0	0	0	0

$$O_0 = \overline{S_2}\,\overline{S_1}\,\overline{S_0}\,y$$
$$O_1 = \overline{S_2}\,\overline{S_1}\,S_0\,y$$
$$O_2 = \overline{S_2}\,S_1\,\overline{S_0}\,y$$
$$O_3 = \overline{S_2}\,S_1 S_0\,y$$
$$O_4 = S_2\,\overline{S_1}\,\overline{S_0}\,y$$
$$O_5 = S_2\,\overline{S_1}\,S_0\,y$$
$$O_6 = S_2 S_1\,\overline{S_0}\,y$$
$$O_7 = S_2 S_1 S_0\,y$$

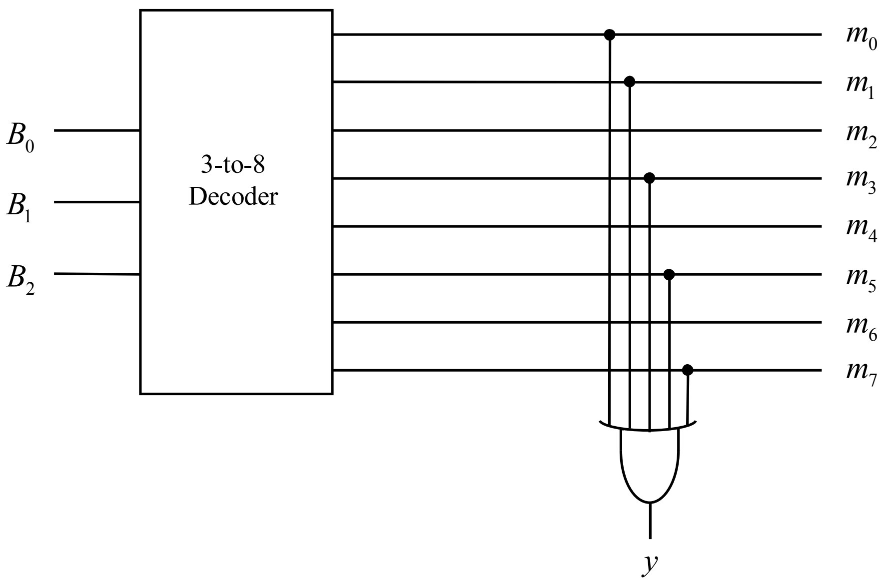

S2
S1
S0
y
O0
O1
O2
O3
O4
O5
O6
O7

تدريب 16:

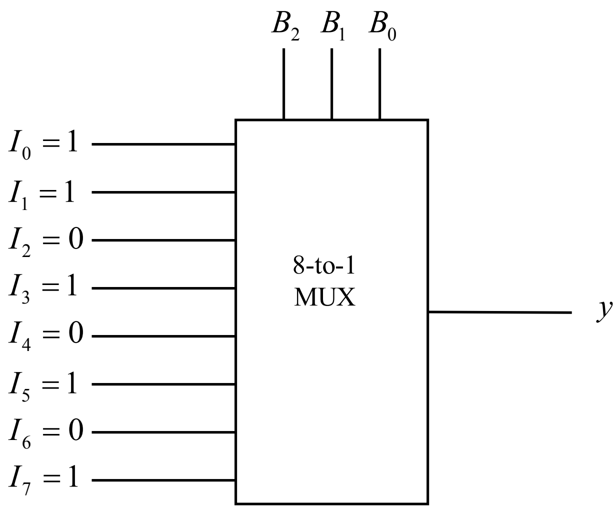

S2
S1
S0
y
1-to-2
DeMUX
1-to-2
DeMUX
1-to-2
DeMUX
1-to-2
DeMUX
(0)
1-to-2
DeMUX
(1)
1-to-2
DeMUX
(2)
1-to-2
DeMUX
(3)
O0
O1
O2
O3
O4
O5
O6
O7

تدريب 17:

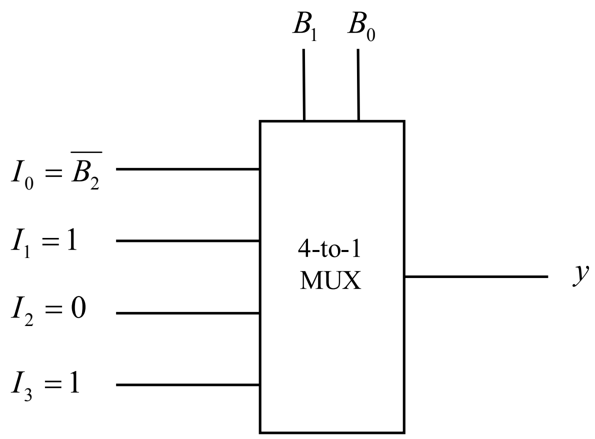
(أ) باستخدام فاك شفرة و مشفر (Decoder & Encoder)



(ب) باستخدام دامج من نوع 8 إلى 1 (8-to-1 MUX)



(ج) باستخدام دامج من نوع 4 إلى 1 (4-to-1 MUX)



مسرد المصطلحات

دوائر التحقق (Parity Checking)

عملية التحقق (Parity Checking) هي عملية تستخدم لاكتشاف حدوث خطأ في البيانـــات المنقولة.

دائرة التحقق الفردي (Odd Parity Checker)

مهمة هذه الدائرة حساب عدد الـــ 1's في خانات الرمز الذي تم استقباله، بما فيها خانـــة التحقق، لتحديد ما إذا كان عددها فردياً أو زوجياً، و تحديد ما إذا كان هنالك خطأ فـــي الرمز أم لا بناء على ذلك.

دوائر الجمع (Adders)

دوائر الجمع أو الجوامع (Adders) هي دوائر منطقية تقوم بإجراء عملية جمع الأعـــداد الممثلة في الصورة الثنائية.

نصف الجامع (Half Adder)

نصف الجامع هو أبسط أنواع الجوامع، و هو عبارة عن دائرة منطقيـــة تقـــوم بجمـــع خانتين ثنائيتين إلى بعضهما البعض و إيجاد حاصل الجمع (Sum) و الحمل.

الجامع الكامل (Full Adder)

تتشابه دائرة الجامع الكامل مع دائرة نصف الجامع في أنها تقوم بإجراء عملية الجمـــع و إيجاد كل من المجموع (Sum) و الحمل الخارج (Carry out)، إلا أن لها دخلاً ثالثاً هـــو عبارة عن حمل داخل (Carry in).

جهاز فك الشفرة (Decoder)

جهاز فك الشفرة عبارة عن دائرة منطقية لها عـــدة أطـــراف خـــرج (Output Lines). واحد فقط من أطراف الخرج هذه يكون نشطاً (Active) أما بقية أطراف الخرج تكـــون غير نشطة.

خط السماح (Enable)

و خط السماح، في الدوائر المنطقية بصورة عامة، هو عبارة عن طرف تحكم يمكن بواسطته أن نبطل عمل الدائرة، أو نسمح لها بالعمل كالمعتاد.

المشفر (Encoder)

عبارة عن دائرة منطقية لها عدة أطراف دخل (Input Lines)، و يكون واحد فقط من أطراف الدخل هذه نشطاً (Active)، أي مساوياً 1، أما بقية أطراف الدخل تكون غير نشطة، أي مساوية 0. خرج الدائرة عبارة عن شفرة (Code) تمثل طرف الدخل النشط.

الدامج (Multiplexer)

الدامج عبارة عن دائرة منطقية لها عدة أطراف دخل، و طرف خرج واحد. يتم توصيل واحد من أطراف الدخل مع طرف الخرج، و يتم اختيار طرف الدخل الذي يتم توصيله بالخرج بواسطة أطراف الاختيار (Select Lines).

المفرق (Demultiplexer)

فالمفرق عبارة عن دائرة منطقية لها عدة أطراف خرج، و طرف دخل واحد. يتم توصيل طرف الدخل مع أحد أطراف الخرج، و يتم اختيار طرف الخرج الذي يتم توصيله بالدخل بواسطة أطراف الاختيار (Select Lines).


الوحدة الخامسة
الدوائر المنطقية التتابعية

محتويات الوحدة

المقدمة

تمهيد

مرحباً بك **عزيزي الدارس**، في الوحدة الخامسة من مقرر "أساسيات التصميم المنطقي". في ما سبق من المقرر تعاملنا مع دوائر منطقية ترابطية (Combinational Logic Circuits)، و فيها يعتمد خرج الدائرة فقط على القيم الحالية للدخل، أما في هذه الوحدة فسنتعرف على النوع الآخر من الدوائر المنطقية و هو الدوائر المنطقية التتابعية (Sequential Logic Circuits)، مثل المراجيح (Flip Flops) و المسجلات (Registers) و العدّادات (Counters)، و فيها لا يعتمد الخرج فقط على القيم الحالية للدخل، و إنما يعتمد أيضاً على القيم السابقة للخرج. أي أن هذا النوع من الدوائر له ذاكرة (Memory) تستطيع اختزان القيم السابقة لخرج الدائرة بحيث تستطيع التأثير على خرجها الحالي. والسبب في ظهور هذه القدرة التخزينية هو وجود تغذية مرتدة (Feedback) من خرج الدائرة إلى دخلها. حيث سنقوم في هذه الوحدة بعرض مبسط لأهم أنواع الدوائر المنطقية التتابعية و أكثرها شيوعاً في الاستخدام، و لن نتعرض لتصميم الدوائر المنطقية التتابعية بالتفصيل، كما فعلنا بالنسبة للدوائر المنطقية الترابطية، بل سنترك هذه الدراسة التفصيلية لمقرر آخر متقدم في التصميم المنطقي. نبدأ هنا بدراسة الوحدة الأساسية في بناء الدوائر المنطقية التتابعية و هي المراجيح (Flip Flops)، حيث نقوم بتوضيح بنائها و طريقة عملها و أنواعها المختلفة و استخدامات كل نوع. ثم ننتقل للمسجلات (Registers) حيث نقوم بتوضيح بنائها و كيفية الكتابة فيها و القراءة منها و كيفية نقل البيانات بينها، كما نتعرف على مسجلات الإزاحة (Shift Registers) بأنواعها المختلفة. و في نهاية الوحدة نتعرف على العدّادات (Counters)، حيث نتعرف على بنائها و أنواعها المختلفة و استخداماتها.

أهداف الوحدة

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

- **تفرق** ما بين الدوائر المنطقية الترابطية و الدوائر المنطقية التتابعية.
- **تصمم** المراجيح بأنواعها المختلفة و توضيح طريقة عملها.
- **تستخدم** المراجيح في تصميم الأنظمة الرقمية.
- **تستخدم** مخططات التزامن في تحليل الدوائر المنطقية التتابعية.
- **تصمم** المسجلات بأنواعها و تستخدمها في الأنظمة الرقمية.
- **توضح** طريقة نقل البيانات بين المسجلات.
- **تصمم** مسجلات الإزاحة.
- **تصمم** العدّادات بأنواعها المختلفة و تشرح طريقة عملها و تستخدمها في الأنظمة الرقمية.

1. الدوائر المنطقية التتابعية (Sequential Logic Circuits)

تنقسم الدوائر المنطقية إلى نوعين؛ دوائر منطقية ترابطية (Combinational Logic Circuits) و دوائر منطقية تتابعية (Sequential Logic Circuits). سميت الدوائر المنطقية الترابطية بهذا الاسم نظراً إلى أن وظيفة الدائرة هي ربط متغيرات الدخل بعمليات منطقية لتوليد متغيرات الخرج، و بالتالي فإن خرج هذا النوع من الدوائر يعتمد فقط على القيم الحالية للدخل، فمتى ما تغير الدخل تبع ذلك تغير الخرج، و إذا لم يتغير الدخل يظل الخرج كما هو. و جميع الدوائر المنطقية التي تعاملنا معها في هذا المقرر حتى الآن، مثل الجوامع (Adders) وجهاز فك الشفرة (Decoder) و المشفر (Encoder) و الدامج (Multiplexer) و المفرق (Demultiplexer)، هي دوائر منطقية ترابطية. أما الدوائر المنطقية التتابعية فلا يعتمد خرجها على القيم الحالية للدخل فقط و إنما يعتمد بالإضافة إلى ذلك على القيم السابقة للخرج، حيث إن هذا النوع من الدوائر له ذاكرة (Memory) تستطيع اختزان ماضي الدائرة بحيث يؤثر هذا الماضي على الخرج الحالي. و السبب في ظهور القدرة التخزينية في الدوائر المنطقية التتابعية هو وجود تغذية مرتدة (Feedback)، حيث أن خرج الدائرة يتم أخذه عبر هذه التغذية المرتدة و إدخاله إلى الدائرة مرة أخرى مع متغيرات الدخل. و نظراً لوجود ماضٍ و حاضر في الدوائر المنطقية التتابعية نستطيع القول إن الزمن (Time) يدخل فيها كمتغير. و دخول الزمن كمتغير يتطلب وجود إشارة التزامن (Clock Signal) في الدوائر المنطقية التتابعية للقيام بدور تنسيقي و تنظيمي هام في النظام الرقمي.

؟

1- أكمل الجدول التالي والذي يلخص الفروقات ما بين الدوائر المنطقية الترابطية و الدوائر المنطقية التتابعية .

	الدوائر المنطقية الترابطية	**الدوائر المنطقية التتابعية**
1. الخرج	يعتمد على القيم الحالية للدخل فقط	....
2. الذاكرة (Memory)	....	لهـــا ذاكـــرة
3. التغذيـــــة المرتـــــدة (Feedback)	....	....
4. الزمن	....	....
5. إشارة التزامن (Clock)	....	....
6. أمثلة	الجوامع، ....،......، الدامج،	........، ....، العدّادات

2- علل لظهور القدرة التخزينية في الدوائر المنطقية التتابعية ؟

2. المراجيح (Flip Flops)

المرجاح (Flip Flop) عبارة عن دائرة منطقية تتابعية لها القدرة على تخـــزين خانـــة ثنائية واحدة (1-bit) فقط من البيانات. و يطلق عليه باللغة العربية أيضاً تسمية القلاب أو النطاط، و لكن سنستخدم هنا تسمية المرجاح نظـــراً لفصـــاحتها و لأدائهـــا للمعنـــى المطلوب بدقة أكبر. حيث إن للمرجاح حالتين (two states) يتأرجح بينهما، أي ينتقل من إحداهما إلى الأخرى تحت تأثير متغيرات الدخل. تسمى الحالة الأولى للمرجـــاح و التي يكون محتفظاً فيها بالقيمة المنطقية 1 بحالة SET، في حين تسمى الحالة الأخـــرى

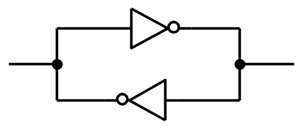
و التي يكون محتفظاً فيها بالقيمة المنطقية 0 بحالـــة RESET أو CLEAR. هـــذا و يعتبر المرجاح وحدة البناء الأساسية لجميع الدوائر المنطقية التتابعية.

1.2 بناء المراجيح

من الممكن أن يتم بناء المراجيح باستخدام العواكس المنطقية أو باستخدام بوابات NOR أو باستخدام بوابات NAND.

مرجاح من العواكس المنطقية:

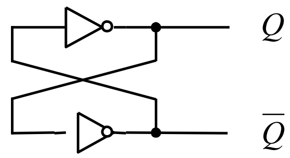
يتكون أبسط أنواع المراجيح من عاكسين منطقيين يقوم خرج كل منهما بتغذية دخل الآخر، كما هو موضح بالشكل التالي:



يسمى الطرف Q بالخرج غير المعكوس للمرجاح، في حين يسمى الطرف $\overline{Q}$ بالخرج المعكوس.

لتخزين قيمة معينة في المرجاح نقوم بتسليط الجهد الكهربائي الممثل لتلك القيمـــة مـــن مصدر خارجي على الطرف Q لفترة زمنية قصيرة جداً (الفترة الزمنية اللازمة لظهور خرج العاكس المنطقي الثاني)، ثم نقوم بإزالة مصدر الدخل الخارجي، فيظل المرجـــاح محتفظاً بتلك القيمة المخزنة به ما دامت تغذية بواباتـــه المنطقيـــة بالقـــدرة الكهربائيـــة مستمرة، و يفقد القيمة المخزنة به عند انقطاع تلك التغذية.

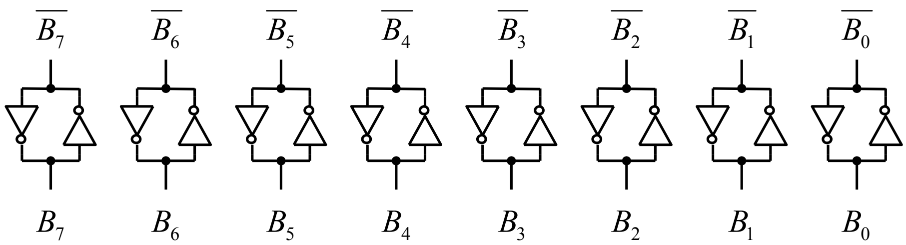
يمكن رسم دائرة المرجاح البسيط المكون من عاكسين منطقيين بالصورة التالية:



نلاحظ هنا وجود التغذية المرتدة (Feedback) من طرفي الخرج للعاكسين المنطقيين إلى طرفي الدخل لهما.

يطلق على هذا المرجاح تسمية Static Latch. و مصطلح Static في الدوائر المنطقيــة يشير إلى غياب إشارة التزامن (Clock)، و المصطلح العكســي Dynamic يشــير إلــى وجود تلك الإشارة. و غياب إشارة التزامن هنا يعني عدم إمكانية تغير حالــة الــدائرة بمرور الزمن فقط، أي أن القيمة المخزنة في المرجاح ستظل كما هي حتى يتم استبدالها بقيمة أخرى. يستخدم هذا المرجاح كوحدة بناء أساسية في نوع مــن أنــواع الــذاكرة (Memory) يسمى Static RAM أو SRAM، كما سيتم توضيحه بالتفصيل فــي الوحــدة التالية من المقرر.

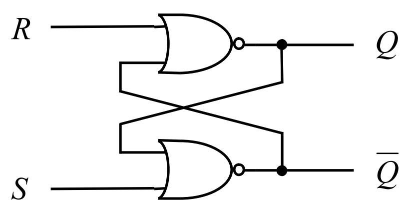
ذكرنا أن المرجاح له القدرة على تخزين خانة ثنائية واحدة (1-bit) فقط من البيانات، فلتخزين معلومة مكونة من مجموعة من الخانات الثنائية نحتاج لعدد من المراجيح بعدد الخانات الثنائية (bits) المطلوب تخزينها، كما هو موضح بالشكل التالي



و تسمى مجموعة المراجيح المستخدمة في تخزين معلومة مكونة من عدد من الخانـــات الثنائية بالمسجل (Register).

مرجاح من بوابات NOR:

بما أن بوابة NOR يمكن أن تعمل عمل العاكس المنطقي، لذلك يمكن استخدامها في بناء المراجيح كما هو موضح أدناه

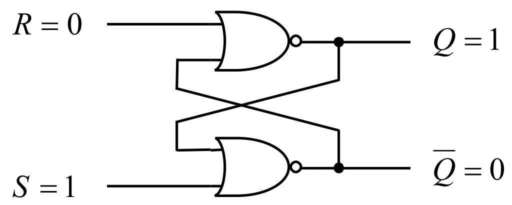


لاحظ أن وجود أكثر من طرف دخل لبوابة NOR سمح لنا بإضـــافة أطــراف أخــرى للمرجاح هي أطراف التحكم S و R، و التي يمكن عن طريقهـا الــتحكم فـــي حالـــة المرجاح. فــ S هو إختصار لكلمة SET و هي حالة المرجاح التي تكون فيها القيمـــة المنطقية 1 مخزنة فيه، و R هو اختصار لكلمة RESET و هي حالة المرجـــاح التـــي تكون فيها القيمة المنطقية 0 مخزنة فيه. أي أن المرجاح يكون في حالة SET إذا كانت القيمة المخزنة فيه هي 1، و يكون في حالة RESET إذا كانت القيمة المخزنة فيه هـــي 0، علماً بأن القيمة المخزنة في المرجاح هي القيمة التي تظهر في طرف الخرج غيـــر المعكوس Q.

يطلق على هذا المرجاح تسمية مرجاح SET/RESET أو مرجاح SR (SR Flip Flop) إختصاراً.

- إجراء عملية SET للمرجاح:

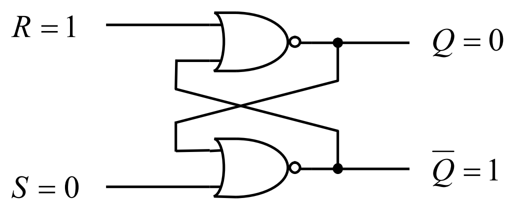
لإجراء عملية SET للمرجاح نضع القيمة المنطقية 1 في الطرف المقابل للعملية المطلوب إجراؤها، أي الطرف S، و نضع القيمة المنطقية 0 في الطرف الآخر، أي الطرف R، كما هو موضح أدناه



لاحظ أن دخول الـــ 1 إلى بوابة NOR الموجودة بالأسفل يحدد خرجها بـــ $\overline{Q}=0$.
و بمعلومية قيمة $\overline{Q}$ و قيمة R يمكن تحديد خرج بوابة NOR الموجودة بالأعلى بـــ $Q=1$.

- إجراء عملية RESET للمرجاح:

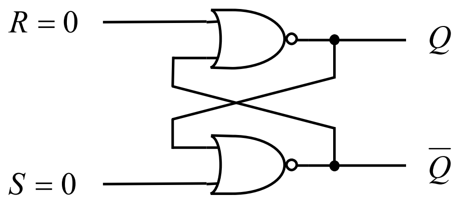
لإجراء عملية RESET للمرجاح نضع القيمة المنطقية 1 في الطرف المقابل للعمليـــة المطلوب إجراؤها، أي الطرف R، و نضع القيمة المنطقية 0 في الطرف الآخـــر، أي الطرف S، كما هو موضح أدناه



لاحظ أن دخول الـــ 1 إلى بوابة NOR الموجودة بالأعلى يحدد خرجها بـــ $Q=0$.
و بمعلومية قيمة Q و قيمة S يمكن تحديد خرج بوابة NOR الموجودة بالأسفل بـــ $\overline{Q}=1$.

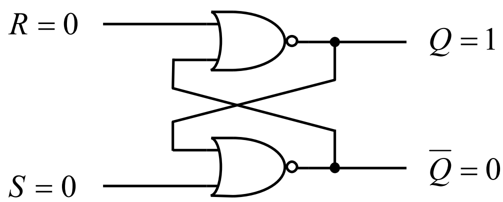
يوجد احتمالا دخل آخران للطرفين S و R، الاحتمال الأول هـــو $S=0$ و $R=0$، و الاحتمال الثاني هو $S=1$ و $R=1$. المطلوب الآن إيجاد حالة المرجاح لكل احتمـــال دخل منهما

• احتمال الدخل $S = 0$ و $R = 0$:



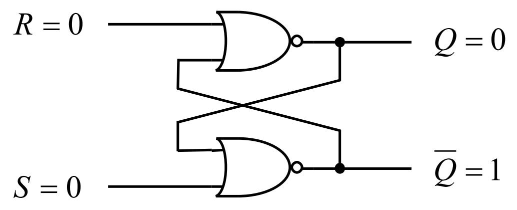
لا نستطيع هنا تحديد حالة المرجاح الجديدة دون معرفة حالته السابقة، لأنه إذا كانت القيمة الموجودة على أحد طرفي الدخل لبوابة NOR هي 0 فلا يمكن تحديد خرجها دون معرفة القيمة الموجودة على طرف الدخل الآخر.

أولاً: إذا كان المرجاح في حالة SET، أي أن $Q = 1$ و $\overline{Q} = 0$



نجد أن الحالة الجديدة للمرجاح هي أيضاً حالة SET، أي أن المرجاح احتفظ بحالته السابقة.

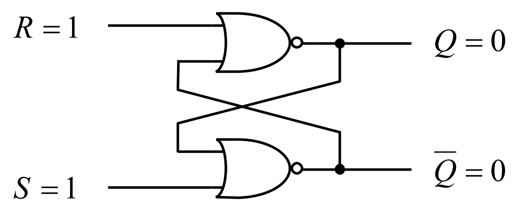
ثانياً: إذا كان المرجاح في حالة RESET، أي أن $Q = 0$ و $\overline{Q} = 1$



نجد أن الحالة الجديدة للمرجاح هي أيضاً حالة RESET، أي أن المرجاح احتفظ بحالته السابقة.

و عليه نستنتج أنه في حالة الدخل $S = 0$ و $R = 0$ يحتفظ المرجاح بحالته السابقة.

- احتمال الدخل $S = 1$ و $R = 1$:



يؤدي احتمال الدخل هذا إلى جعل كلا طرفي الخرج Q و $\overline{Q}$ مساويين 0، و هو أمر غير مسموح به. كما أنه عند عودة القيمة الموضوعة على طرفي الدخل S و R من 1 إلى 0 في وقت واحد فإن حالة المرجاح تكون غير محددة، أي لا يمكن التكهن بها، لأنها تعتمد على أي الطرفين S و R تغير قبل الآخر. لذلك فإن احتمال الدخل $S = 1$ و $R = 1$ غير مستخدم أو غير مسموح به (Invalid).

هذا و يمكن تلخيص النتائج السابقة في جدول الصواب (Truth Table) التالي:

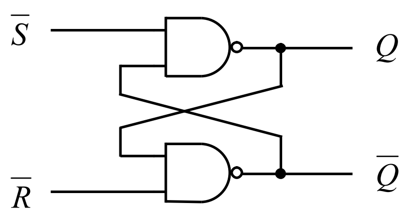
S	R	Q_{n+1}	
0	0	Q_n	Keep
0	1	0	RESET
1	0	1	SET
1	1	Invalid	

حيث Q_{n+1} هي الحالة الجديدة للمرجاح، و Q_n هي الحالة السابقة للمرجاح.
كما يمكن أن نقوم في جدول الصواب بإدراج الخرج المعكوس $\overline{Q}$ إضافة إلى الخرج غير المعكوس Q، و ذلك لتوضيح ما يحدث في حالة الدخل $S=1$ و $R=1$، كما هو موضح أدناه

S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	Q_n	$\overline{Q}_n$	Keep
0	1	0	1	RESET
1	0	1	0	SET
1	1	0	0	Invalid

مرجاح من بوابات NAND:

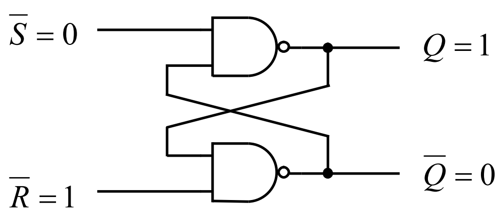
بما أن بوابة NAND، مثلها في ذلك مثل بوابة NOR، يمكن أن تعمل عمل العاكس المنطقي، لذلك يمكن استخدامها في بناء المراجيح كما هو موضح أدناه



المرجاح هنا أيضاً عبارة عن مرجاح SET/RESET أو مرجاح SR (SR Flip Flop)، إلا أن دخله نشط منخفض (Active Low)، أي أن العملية المطلوبة يتم إجراؤها بوضع 0 في الطرف المقابل لها.

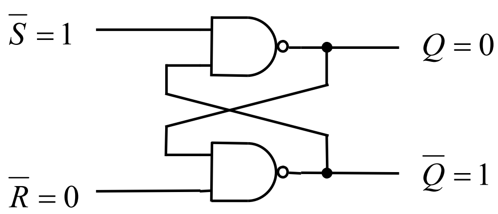
- إجراء عملية SET للمرجاح:

يتم ذلك بجعل $\overline{S}=0$ و $\overline{R}=1$، كما هو موضح أدناه



- إجراء عملية RESET للمرجاح:

يتم ذلك بجعل $\overline{R}=0$ و $\overline{S}=1$، كما هو موضح أدناه



هذا و يمكن بسهولة إثبات أن الدخل $\overline{S}=1$ و $\overline{R}=1$ يؤدي لاحتفاظ المرجاح بحالته السابقة، و الدخل $\overline{S}=0$ و $\overline{R}=0$ يؤدي إلى جعل كلا الخرجين Q و $\overline{Q}$ مساويين للقيمة 1.

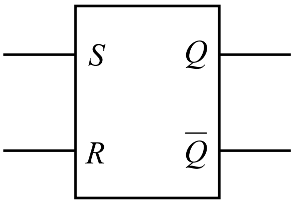
و يمكن تلخيص هذه النتائج في جدول الصواب التالي

$\overline{S}$	$\overline{R}$	Q_{n+1}	
0	0	Invalid	
0	1	1	SET
1	0	0	RESET
1	1	Q_n	Keep

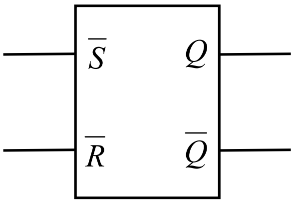
أو

$\overline{S}$	$\overline{R}$	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	1	1	Invalid
0	1	1	0	SET
1	0	0	1	RESET
1	1	Q_n	$\overline{Q}_n$	Keep

و الشكل التالي يوضح المخطط المنطقي (Logic Diagram) لمرجاح SR



دخل نشط مرتفع
(Active High Inputs)



دخل نشط منخفض
(Active Low Inputs)

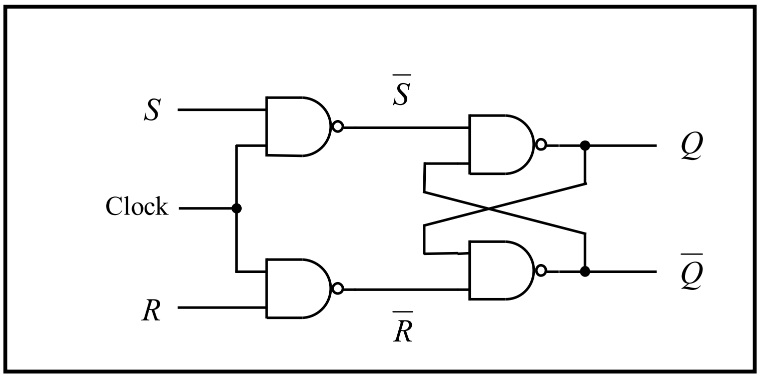
أسئلة تقويم ذاتي

1- عرف المرجاح (Flip Flop) .

2- إلى ماذا يشير كل مـــن مصـــطلحي (Static) و (Dynamic) في الدوائر المنطقية .

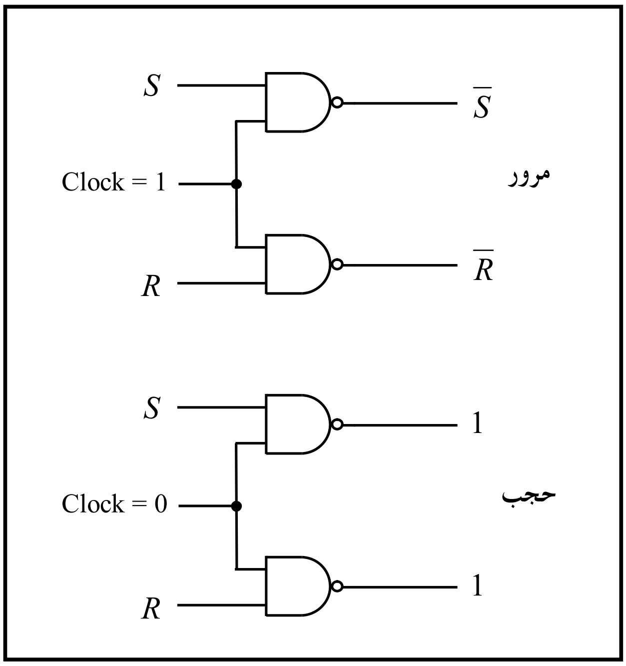
2.2 المراجيح المتزامنة (Clocked or Gated Flip Flops)

المرجاح المتزامن (Clocked Flip Flop) تدخل عليه إشارة تسمى إشارة التزامن (Clock Signal) أو Clock اختصاراً. و تدخل إشارة التزامن على مرجاح SR بالطريقة الموضحة بالشكل التالي:



و يطلق على المرجاح هنا تسمية مرجاح SR المتزامن (Clocked SR Flip Flop).

و إشارة التزامن (Clock) تشبه في عملها إلى حد كبير إشارة السماح (Enable)، فإذا كانت إشارة التزامن مرتفعة (High)، أي مساوية 1، تمر الإشارتان S و R إلى المرجاح و يستجيب لهما بالصورة المعتادة، أما إذا كانت إشارة التزامن منخفضة (Low)، أي مساوية 0، فيتم حجب الإشارتين S و R عن المرجاح و يظل المرجاح محتفظاً بحالته السابقة. كما هو موضح أدناه



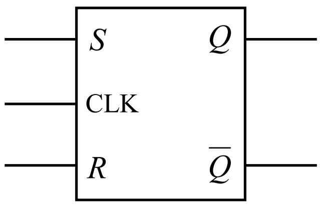
و في ما يلي جدول الصواب لمرجاح SR المتزامن

C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	×	×	Q_n	$\overline{Q}_n$	Keep
1	0	0	Q_n	$\overline{Q}_n$	Keep
1	0	1	0	1	RESET
1	1	0	1	0	SET
1	1	1	1	1	Invalid

حيث المتغير C يمثل قيمة إشارة التزامن (Clock).

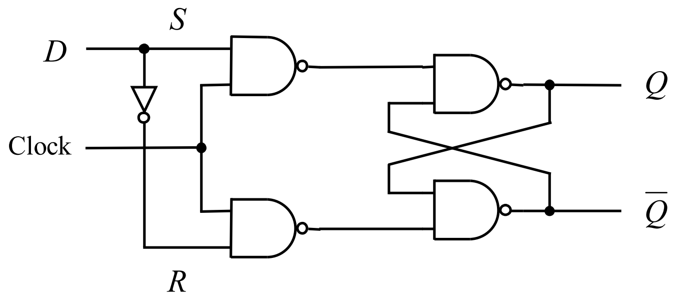
<u>لاحظ</u> أن دخل مرجاح SR المتزامن نشط مرتفع (Active High).

الشكل التالي يمثل المخطط المنطقي لمرجاح SR المتزامن



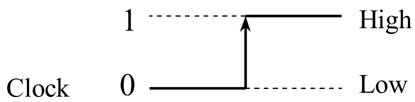
مرجاح D (D Flip Flop)

و D هنا اختصار لكلمة Data، أي أن الأسم الكامل للمرجاح هو Data Flip Flop. و مرجاح D عبارة عن مرجاح SR متزامن تم ربط طرفي الدخل S و R له في طرف دخل واحد هو D باستخدام عاكس منطقي، كما هو موضح بالشكل التالي

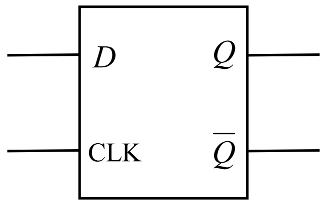


فإذا وضعنا القيمة المنطقية 0 في الطرف D يكون $S = 0$ و $R = 1$ فتحدث عملية RESET للمرجاح، أي يتم إختزان القيمة 0 فيه. و إذا وضعنا القيمة المنطقية 1 في الطرف D يكون $S = 1$ و $R = 0$ فتحدث عملية SET للمرجاح، أي يتم اختزان القيمة 1 فيه. أي أن القيمة التي يتم وضعها على الطرف D يتم إختزانها داخل المرجاح.

لاحظ ارتباط إنتقال القيمة الموضوعة على الطرف D و اختزانها داخل المرجاح بإشارة التزامن (Clock)، حيث تنتقل القيمة إلى داخل المرجاح و تختزن في اللحظة التي تتغير فيها إشارة التزامن من Low إلى High، كما هو موضح أدناه



و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح D



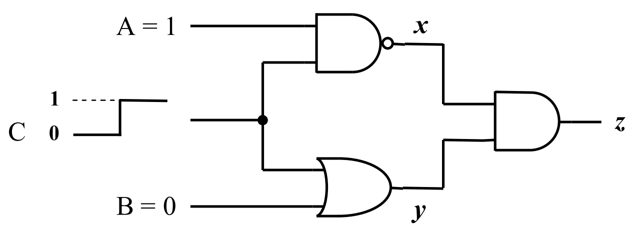
C	D	Q_{n+1}
0	×	Q_n
1	0	0
1	1	1

هذا و يطلق على مرجاح D أيضاً تسمية Dynamic Latch، و يستخدم أساساً في بناء المسجلات (Registers).

أهمية التزامن (Timing)

لاحظنا ظهور إشارة التزامن (Clock) في الدوائر المنطقية التتابعية (Sequential Logic Circuits)، و لم نلاحظها من قبل في الدوائر المنطقية الترابطية (Combinational Logic Circuits)، فما أهمية التزامن بالنسبة للدوائر المنطقية التتابعية؟

سنقوم بتوضيح أهمية التزامن باستخدام الدائرة البسيطة التالية



للدائرة ثلاثة متغيرات دخل هي A و B و C. المتغير A ثابت دوماً في القيمة 1، و المتغير B ثابت دوماً في القيمة 0، أما المتغير C فتتغير قيمته في لحظة معينة من 0 إلى 1. و المطلوب إيجاد خرج الدائرة z.

- كدائرة منطقية ترابطية:

إذا تعاملنا مع الدائرة كدائرة منطقية ترابطية و لم نأخذ عامل الزمن في الاعتبار نجد أن

$$x = \overline{AC}$$
$$y = B + C$$
$$z = xy = \overline{AC}\,(B + C)$$

و بوضع A=1 و B=0 يكون

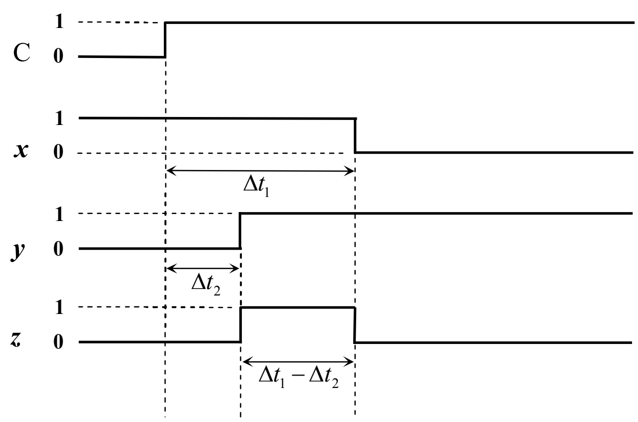
$$z = \overline{1 \cdot C}\,(0 + C) = \overline{C}\,C = 0$$

أي أن الخرج z يساوي 0 على الدوام بغض النظر عن التغير الحادث في قيمة المتغير C.

- كدائرة منطقية تتابعية:

إذا تعاملنا مع الدائرة كدائرة منطقية تتابعية فلابد من أخذ عامل الزمن في الاعتبار، فلكل بوابة منطقية زمن تأخر إنتقال (Propagation Delay) هو عبارة عن الفترة الزمنية التي تمضي ما بين تسليط الدخل على البوابة و ظهور الاستجابة في خرجها. و زمن تأخر الإنتقال للبوابات المنطقية صغير جداً و يقاس بالنانوثانية (ns). و يختلف زمن تأخر الإنتقال من بوابة إلى أخرى، كما يختلف لنفس البوابة باختلاف الظروف المحيطة بها مثل درجة الحرارة. أي أنه من الصعب تحديد زمن تأخر الانتقال لبوابة معينة بدقة.

في الدائرة أعلاه نفترض أن زمن تأخر الإنتقال لبوابة NAND هو Δt_1، و زمن تأخر الانتقال لبوابة OR هو Δt_2، كما نفترض أن $\Delta t_1 > \Delta t_2$. و عليه فإن التغير في خرج بوابة NAND، أي التغير في قيمة المتغير x، يحدث بعد التغير في دخلها، أي التغير في قيمة المتغير C، بزمن مقداره Δt_1. و التغير في خرج بوابة OR، أي التغير في قيمة المتغير y، يحدث بعد التغير في دخلها، أي التغير في قيمة المتغير C، بزمن مقداره Δt_2. كما هو موضح بالشكل التالي



<u>نلاحظ</u> هنا أن الخرج z قد أصبح مساوياً 1 لفترة زمنية قصيرة جداً تساوي $\Delta t_1 - \Delta t_2$، و هو أمر غير متوقع. و يسمى مثل هذا الخرج غير المتوقع، و الناتج عن اختلاف زمن تأخر الانتقال للبوابات المنطقية، بالـــ (التخمين) وهي عبارة عن قيم غير معروفة و من الواضح أنه من الصعب جداً التنبوء بمكان أو زمان ظهور هذه الـــ القيم في الدوائر المنطقية، حيث أن ذلك يتطلب تحليلاً غاية في الدقة للدائرة المنطقية، يؤخذ فيه في الإعتبار زمن تأخر الانتقال لكل بوابة منطقية. و إن كان هذا ممكناً للدائرة البسيطة أعلاه فإنه يكاد يكون مستحيلاً بالنسبة للدوائر المعقدة.

السؤال الآن هو ما تأثير هذه القيم على الدائرة المنطقية؟

- بالنسبة للدوائر المنطقية الترابطية:

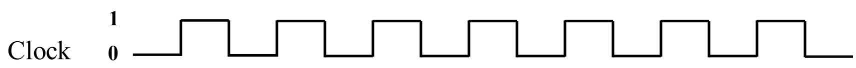
نظراً لظهور هذه القيم لفترة زمنية غاية في القصر و تلاشيها بعد ذلك فإنها تكاد أن تمر دون أن تلاحظ، و لا يكون لها بالتالي أي تأثير على الدائرة المنطقية.

- بالنسبة للدوائر المنطقية التتابعية:

  يوجد هنا احتمال أن يقوم أحد المراجيح بالتقاط هذه القيم أثناء فترة ظهورها القصيرة و تخزينها. عند ذلك لا يعود تأثير هذه ا لقيم على الدائرة المنطقية تأثيراً وقتياً و إنما يصبح تأثيراً دائماً.

إذن كيف نتلافى تأثير هذه القيم على الدوائر المنطقية التتابعية؟

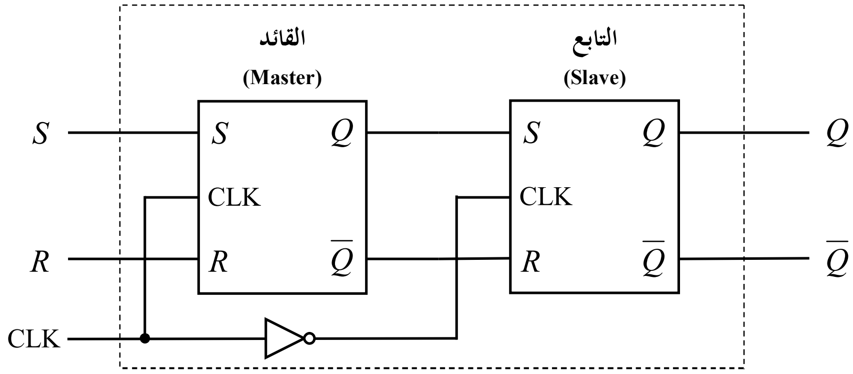
لتلافي تأثير هذه القيم على الدائرة المنطقية التتابعية يكفي الانتظار لفترة زمنية كافية لتلاشي هذه القيم قبل قراءة خرج الدائرة، بحيث نضمن أن ذلك الخرج خالٍ من القيم غير المعروفة وهنا يأتي دور إشارة التزامن (Clock) التي تقوم بتنظيم فترات الانتظار هذه. فإشارة التزامن عبارة عن إشارة تتغير قيمتها بانتظام ما بين 0 و 1، كما هو موضح بالشكل التالي:



فالفترات التي تكون فيها إشارة التزامن منخفضة (Low) هي عبارة عن فترات انتظار لضمان تلاشي هذه القيم غير المتوقعة.

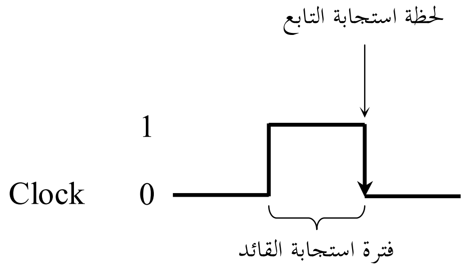
3.2 مرجاح القائد–التابع (Master-Slave Flip Flop)

يتكون مرجاح القائد–التابع من مرجاحي SR متزامنين متصلين ببعضهما البعض بحيث يغذي خرج أولهما دخل الثاني، كما هو موضح بالشكل التالي:



يسمى المرجاح الأول بالقائد (Master) و يسمى المرجاح الثاني بالتابع (Slave).

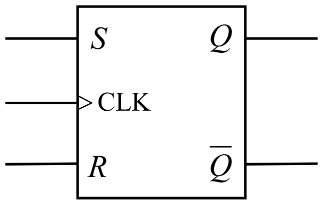
<u>لاحظ</u> أن إشارة التزامن (Clock) تدخل مباشرة إلى مرحلة القائد في حين تدخل معكوسة إلى مرحلة التابع، و معنى هذا أن إستجابة المرجاحين لا تتم في وقت واحد. فعندما تكون إشارة التزامن مرتفعة (High) يستجيب مرجاح القائد للدخل S و R، في حين يكون مرجاح التابع في ذلك الوقت مغلقاً و محتفظاً بحالته السابقة، و في اللحظة التي تهبط فيها إشارة التزامن من High إلى Low ينغلق مرجاح القائد و تنتقل حالته إلى مرجاح التابع و تظهر في الخرج. كما هو موضح بالشكل التالي:



أي أن مرجاح القائد–التابع يستجيب لأي تغير يحدث في طرفي الدخل S و R طالما كانت إشارة التزامن (Clock) مرتفعة (High)، و تظهر الاستجابة في خرجه لحظة هبوط إشارة التزامن من High إلى Low. و الاستجابة التي تظهر في الخرج هنا هي

آخر حالة للمرجاح مباشرة قبل هبوط إشارة التزامن. و يظل خرج المرجاح ثابتاً بعد هبوط إشارة التزامن و ذلك حتى الهبوط الذي يليه.

و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح القائد–التابع

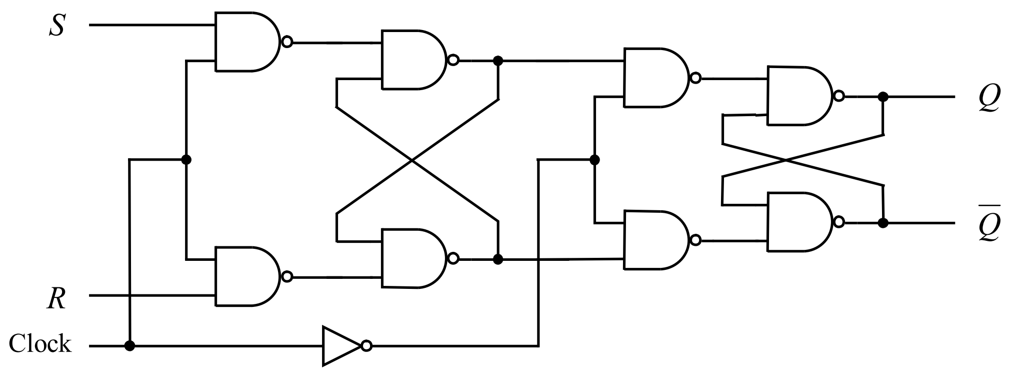


C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	×	×	Q_n	$\overline{Q}_n$	Keep
1	0	0	Q_n	$\overline{Q}_n$	Keep
1	0	1	0	1	RESET
1	1	0	1	0	SET
1	1	1	1	1	Invalid

<u>لاحظ</u> في المخطط المنطقي المثلث الصغير الموضوع عند طرف الدخل لإشارة التزامن (Clock) و الذي يدل على أن دخل المرجاح ينشط مع الحافة الصاعدة لنبضة التزامن، أي لحظة انتقال إشارة التزامن من Low إلى High.

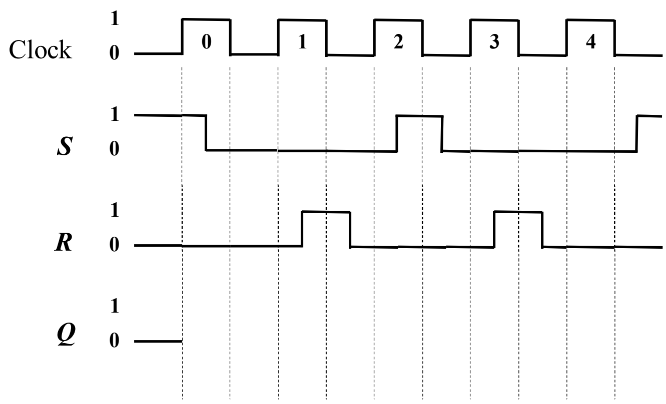
<u>لاحظ</u> أيضاً أن جدول الصواب لمرجاح القائد–التابع هو نفسه جدول الصواب لمرجاح SR المتزامن، أي أن كلا المرجاحين يستجيبان للدخل S و R بنفس الطريقة، و لكن الفرق بينهما يكون في لحظة ظهور الاستجابة في الخرج. ففي مرجاح SR المتزامن تظهر الاستجابة في الخرج فور حدوث التغير في الدخل، ما دامت إشارة التزامن (Clock) مرتفعة، أما في مرجاح القائد–التابع فلا تظهر الاستجابة في الخرج إلا لحظة هبوط نبضة التزامن من High إلى Low.

الشكل التالي يوضح الدائرة المنطقية لمرجاح القائد–التابع



مخططات التزامن (Timing Diagrams)

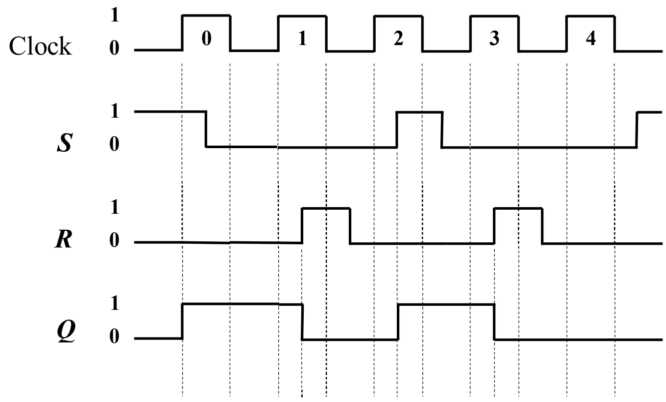
نظراً إلى أن الزمن (Time) يدخل كمتغير في الدوائر المنطقية التتابعية فلابد من وسيلة لمتابعة التغير الذي يحدث في حالة الدائرة مع الزمن. هذه الوسيلة هي مخطط التزامن (Timing Diagram). فمخطط التزامن يوضح التغير الذي يحدث في متغيرات الدخل و الخرج للدائرة المنطقية مع الزمن. على سبيل المثال يوضح الشكل التالي مخطط تزامن معطى فيه الإشارات الداخلة إلى مرجاح SR متزامن، و هي إشارة التزامن (Clock) و متغيرا الدخل S و R، و معطى فيه أيضاً الحالة الابتدائية للمرجاح و هي حالة RESET، و مطلوب إيجاد خرج المرجاح Q



(**<u>ملاحظة</u>:** لتسهيل متابعة الشرح قمنا بترقيم نبضات التزامن)

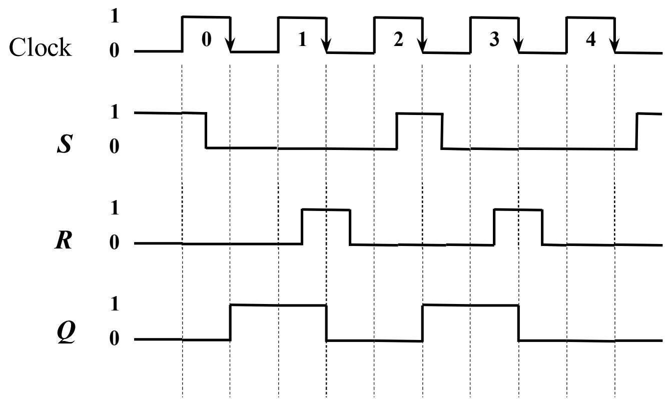
نعلم أن مرجاح SR المتزامن يستجيب للدخل S و R فقط عندما تكون إشارة التزامن مرتفعة (High)، و يظل محتفظاً بآخر حالة وصل إليها عندما تكون إشارة التزامن منخفضة (Low). لذلك ننظر إلى قيم الدخل S و R و التغير الذي يحدث فيها في كل نبضة من نبضات التزامن، منذ بداية النبضة و حتى نهايتها، و نوضح الاستجابة لهذه التغيرات على الخرج Q. فمع بداية النبضة رقم 0 نجد أن $S=1$ و $R=0$ مما يؤدي إلى حدوث عملية SET للمرجاح، و عند منتصف النبضة رقم 0 تقريباً تتحول قيمة S من 1 إلى 0، أي يصبح دخل المرجاح هو $S=0$ و $R=0$ و يؤدي هذا لاحتفاظ المرجاح بحالته أي حالة SET، و يستمر ذلك حتى نهاية النبضة رقم 0. و ما بين النبضة رقم 0 و النبضة رقم 1 لا يحدث أي تغير في حالة المرجاح نظراً إلى أن إشارة التزامن منخفضة (Low). مع بداية النبضة رقم 1 نجد أن $S=0$ و $R=0$ مما يؤدي إلى احتفاظ المرجاح بحالته، أي حالة SET، و ذلك حتى منتصف النبضة رقم 1 تقريباً، حيث تتحول قيمة R من 0 إلى 1، أي يصبح دخل المرجاح هو $S=0$ و $R=1$ و يؤدي هذا لحدوث عملية RESET للمرجاح، و يستمر ذلك حتى نهاية النبضة رقم 1. ما بين النبضة رقم 1 و النبضة رقم 2 لا يحدث أي تغير في حالة المرجاح. مع بداية النبضة رقم 2 نجد أن $S=0$ و $R=0$ مما يؤدي إلى احتفاظ المرجاح بحالته، أي حالة RESET، و ذلك حتى منتصف النبضة رقم 2 تقريباً، حيث تتحول قيمة S من 0 إلى 1، أي يصبح دخل المرجاح هو $S=1$ و $R=0$ و يؤدي هذا لحدوث عملية SET للمرجاح، و يستمر ذلك حتى نهاية النبضة رقم 2. ما بين النبضة رقم 2 و النبضة رقم 3 لا يحدث أي تغير في حالة المرجاح. مع بداية النبضة رقم 3 نجد أن $S=0$ و $R=0$ مما يؤدي إلى احتفاظ المرجاح بحالته، أي حالة SET، و ذلك حتى منتصف النبضة رقم 3 تقريباً، حيث تتحول قيمة R من 0 إلى 1، أي يصبح دخل المرجاح هو $S=0$ و $R=1$ و يؤدي هذا لحدوث عملية RESET للمرجاح، و يستمر ذلك حتى

نهاية النبضة رقم 3. ما بين النبضة رقم 3 و النبضة رقم 4 لا يحدث أي تغير في حالة المرجاح. في بداية النبضة رقم 4 نجد أن $S = 0$ و $R = 0$ مما يؤدي لاحتفاظ المرجاح بحالته، أي حالة RESET، و يستمر ذلك حتى نهاية النبضة. و بعد نهاية النبضة رقم 4 لا يحدث أي تغير في حالة المرجاح. كما هو موضح بالشكل التالي



ماذا لو كان مطلوباً إكمال نفس مخطط التزامن و لكن لمرجاح من نوع القائد–التابع (Master-Slave Flip Flop)؟

نعلم أن مرجاح القائد–التابع يستجيب للدخل S و R بنفس الطريقة التي يستجيب بها مرجاح SR المتزامن، إلا أن استجابة مرجاح القائد التابع لا تظهر في خرجه إلا لحظة هبوط نبضة التزامن من High إلى Low. و عليه يكون شكل مخطط التزامن لمرجاح القائد–التابع هو

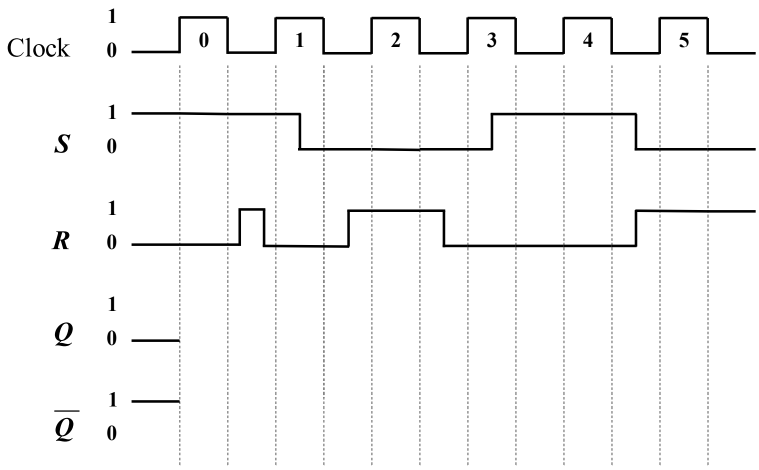


لاحظ أن الاستجابة لعملية SET التي حدثت في بداية النبضة رقم 0 لم تظهر في خـرج المرجاح إلا في نهاية النبضة، لحظة الهبوط من High إلـى Low. و لـم يحـدث أي تغيير في خرج المرجاح بين ذلك الهبوط و الهبوط الذي يليه في نهاية النبضة رقـم 1. ففي منتصف النبضة رقم 1 حدثت عملية RESET و لكن لم تظهر الاستجابة لهـا فـي الخرج إلا لحظة الهبوط في نهاية النبضة رقم 1. و لم يحدث أي تغييـر فـي خـرج المرجاح بين ذلك الهبوط و الهبوط التالي في نهاية النبضـة رقـم 2. ففـي منتصـف النبضة رقم 2 حدثت عملية SET و لكن لم تظهر في الخرج إلا لحظة الهبوط في نهاية النبضة رقم 2. و لم يحدث أي تغيير في خرج المرجاح بين ذلـك الهبـوط و الهبـوط التالي في نهاية النبضة رقم 3. ففي منتصف النبضة رقم 3 حـدثت عمليـة RESET و لكن لم تظهر في الخرج إلا لحظة الهبوط في نهاية النبضة رقـم 3. و لـم يحـدث أي تغيير في خرج المرجاح بين ذلك الهبوط و الهبوط التالي في نهاية النبضة رقـم 4. و في لحظة الهبوط في نهاية النبضة رقم 4 لم يحدث تغييـر فـي خـرج المرجـاح لأن المرجاح ظل طوال تلك النبضة محتفظاً بآخر حالة وصل إليها.

مثال:

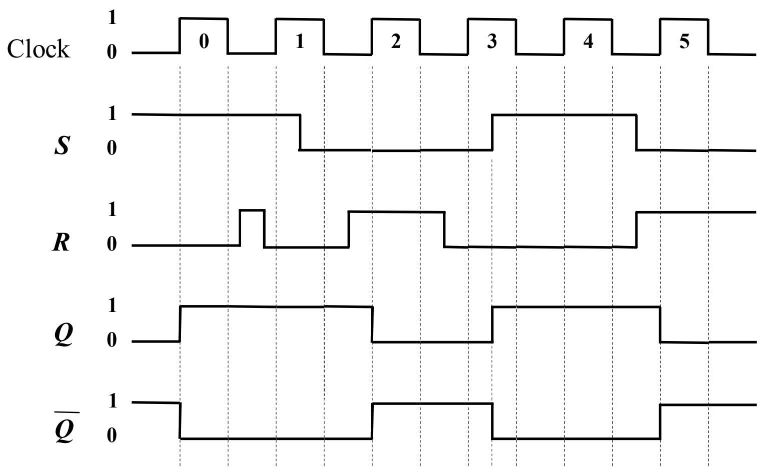
أكمل مخطط التزامن التالي و ذلك:

(أ) لمرجاح SR متزامن. (ب) لمرجاح القائد–التابع.

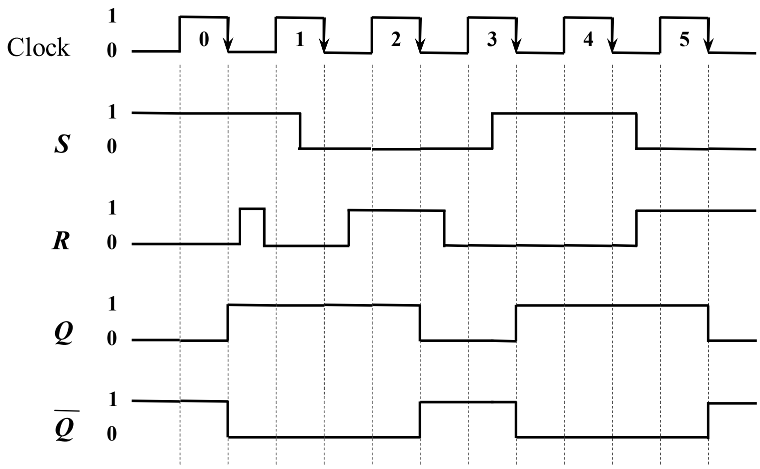


الحل:

(أ) مرجاح SR متزامن

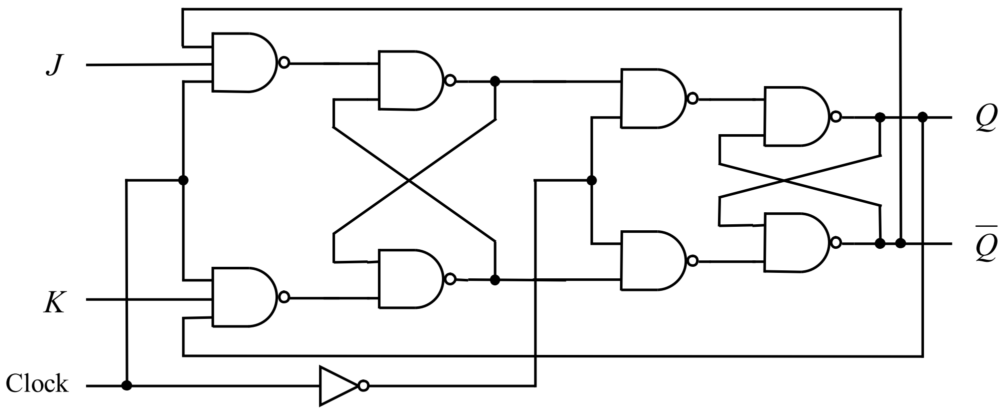


(ب) مرجاح القائد–التابع



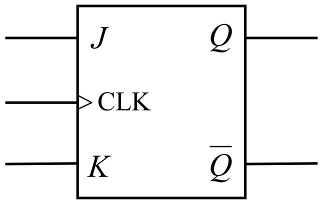
مرجاح JK (JK Flip Flop)

مرجاح JK هو عبارة عن مرجاح من نوع القائد–التابع مزود بتغذية مرتدة (Feedback) إضافية، كما هو موضح بالشكل التالي:



مرجاح الـــJK

و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح JK

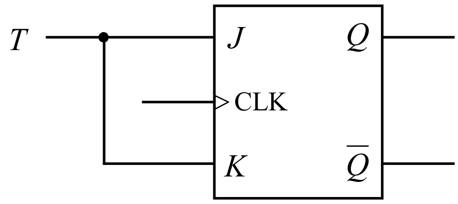


C	J	K	Q_{n+1}	
0	×	×	Q_n	Keep
1	0	0	Q_n	Keep
1	0	1	0	RESET
1	1	0	1	SET
1	1	1	$\overline{Q}_n$	Toggle

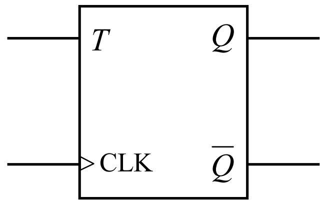
نلاحظ هنا أن جدول الصواب لمرجاح JK يشبه إلى حد كبير جدول الصواب لمرجاح القائد–التابع، حيث يحل الطرف J محل الطرف S في إجراء عملية SET للمرجاح و يحل الطرف K محل الطرف R في إجراء عملية RESET للمرجاح. و لكن يتميز مرجاح JK عن مرجاح القائد–التابع في عدم وجود دخل غير مسموح به أو غير مستخدم، حيث إن الدخل $J=1$ و $K=1$ يؤدي إلى عكس حالة المرجاح، و هي العملية التي تسمى Toggle.

مرجاح T (T Flip Flop)

و T هنا هي اختصار لكلمة Toggle، بمعنى عكس الحالة، كما سبق و أن أوضحنا. و مرجاح T هو عبارة عن مرجاح JK تم ربط طرفي الدخل له في طرف واحد هو الطرف T، كما هو موضح بالشكل التالي:



و في ما يلي المخطط المنطقي و جدول الصواب لمرجاح T



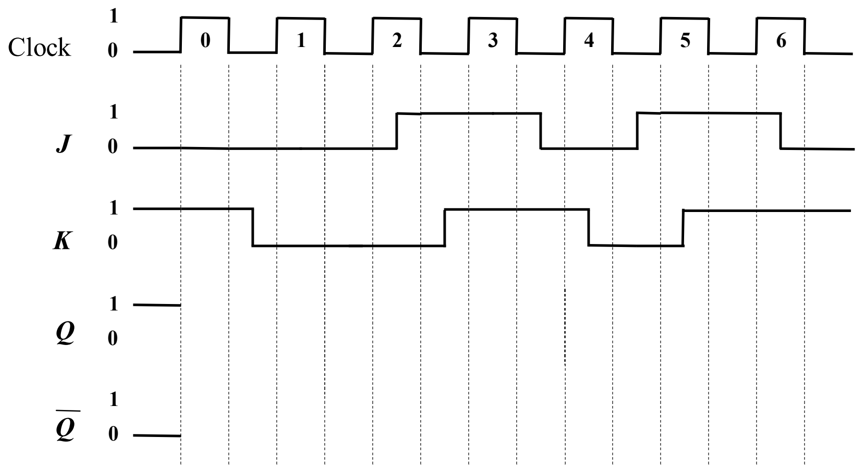
C	T	Q_{n+1}	
0	×	Q_n	Keep
1	0	Q_n	Keep
1	1	$\overline{Q}_n$	Toggle

لاحظ عدم إمكانية إجراء عملية SET أو عملية RESET لمرجاح T، بـل يمكـن فقـط الاحتفاظ بحالته السابقة أو عكس تلك الحالة.

تستخدم مراجيح JK و مراجيح T في بناء العدّادات (Counters).

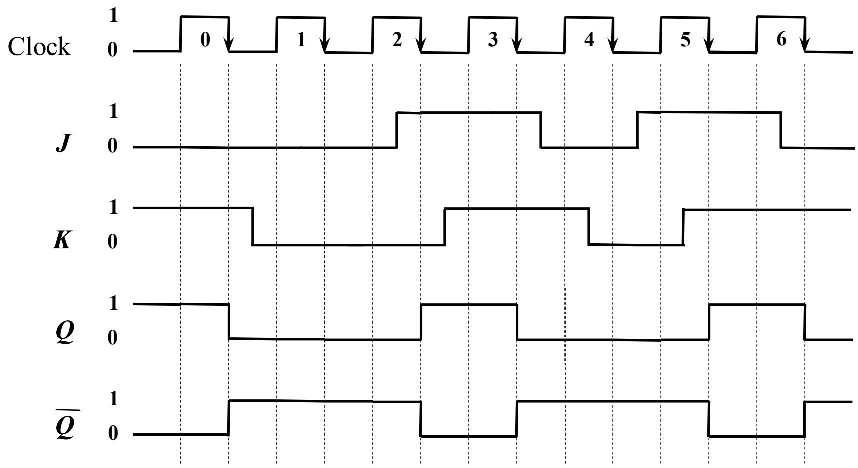
مثال:

أكمل مخطط التزامن التالي لمرجاح JK



Clock
J
K
Q
$\overline{Q}$
1
0
0
1
2
3
4
5
6

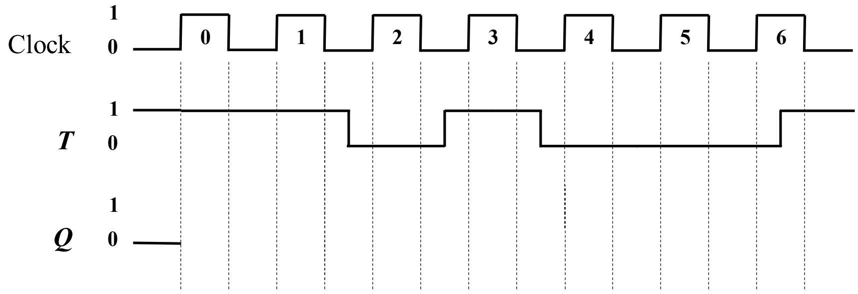
الحل:



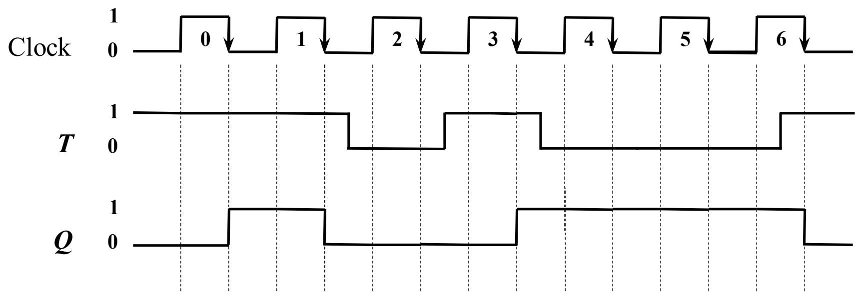
Clock
J
K
Q
$\overline{Q}$
1
0
0
1
2
3
4
5
6

مثال:

أكمل مخطط التزامن التالي لمرجاح T

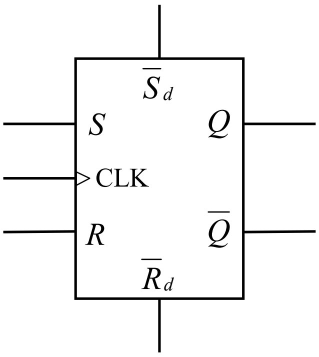


الحل:



4.2 أطراف الدخل المباشر (Direct Inputs)

في بعض الأحيان قد يكون مطلوباً تغيير حالة المرجاح بصورة استثنائية، بغض النظر عن حالة إشارة التزامن. فعلى سبيل المثال قد يكون مطلوباً وضع حالة ابتدائية في المرجاح في غياب إشارة التزامن، بحيث يبدأ المرجاح العمل من تلك الحالة عندما تبدأ نبضات التزامن، أو قد يكون مطلوباً تغيير التسلسل الطبيعي الذي تمر به حالات المرجاح و وضع حالة معينة فيه بصورة استثنائية. تستخدم لهذا الغرض أطراف الدخل المباشر (Direct Inputs)، التي تسمى أيضاً بأطراف الدخل غير المتزامن (Asynchronous Inputs)، و التي يرمز لها بـ $\overline{S}_d$ و $\overline{R}_d$. و الشكل التالي يمثل مرجاحاً من نوع القائد–التابع مزود بأطراف دخل مباشر



حيث يستخدم الطرف $\overline{S}_d$ في إجراء عملية SET للمرجاح بصورة مباشرة، و يستخدم الطرف $\overline{R}_d$ في إجراء عملية RESET للمرجاح بصورة مباشرة. أحياناً يرمز لأطراف الدخل المباشر بـ $\overline{\text{PRESET}}$ و $\overline{\text{CLEAR}}$.

لاحظ أن أطراف الدخل المباشر نشطة منخفضة (Active Low)، بمعنى أن العملية المطلوبة يتم إجراؤها بوضع 0 في الطرف المقابل لها. فلإجراء عملية SET بصورة مباشرة نضع القيمة 0 على الطرف $\overline{S}_d$ و القيمة 1 على الطرف $\overline{R}_d$، و لإجراء عملية RESET بصورة مباشرة نضع القيمة 0 على الطرف $\overline{R}_d$ و القيمة 1 على الطرف $\overline{S}_d$، و العملية المطلوبة هنا تتم بغض النظر عن حالة إشارة التزامن. أما إذا تم وضع القيمة 1 على كلا الطرفين $\overline{S}_d$ و $\overline{R}_d$ فإن المرجاح يستجيب لأطراف الدخل المتزامن (S و R و CLK) بالصورة المعتادة. أما الدخل $\overline{S}_d = 0$ و $\overline{R}_d = 0$ فهو دخل غير مسموح به لأنه يجعل كلا طرفي الخرج Q و $\overline{Q}$ مساويين 1. و يمكن تلخيص ذلك في جدول الصواب التالي:

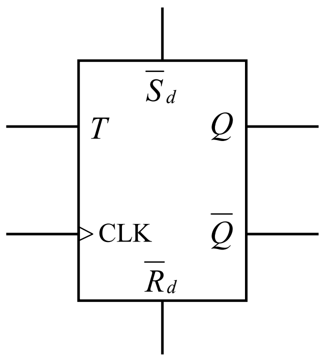
$\overline{S}_d$	$\overline{R}_d$	Q_{n+1}	
0	0	Not Used	
0	1	1	SET Direct
1	0	0	RESET Direct
1	1	**استجابة للدخل المتزامن**	

لاحظ أن أي طرف من أطراف الدخل المباشر مطلوب وضع القيمة المنطقية 1 فيه يمكن تركه دون توصيل، أي تركه مفتوحاً (Open). فمثلاً لإجراء عملية SET نضع 0 على الطرف $\overline{S}_d$ و نترك الطرف $\overline{R}_d$ بدون توصيل، و لجعل المرجاح يستجيب للدخل المتزامن نترك كلا الطرفين $\overline{S}_d$ و $\overline{R}_d$ بدون توصيل. و السبب في ذلك هو أنه في الدوائر المنطقية المنتمية لعائلة TTL (Transistor Transistor Logic)، و هي من أكثر عائلات المنطق شيوعاً في الاستخدام، ترك الطرف دون توصيل يكافيء وضع القيمة المنطقية 1 فيه.

و في ما يلي جدول الصواب لمرجاح SR متزامن (أو مرجاح القائد–التابع) مزود بأطراف دخل مباشر

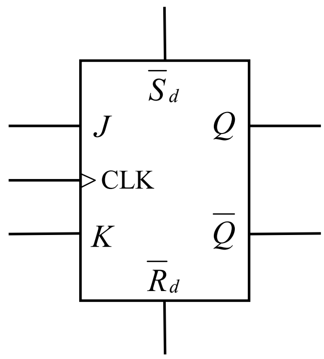
$\overline{S}_d$	$\overline{R}_d$	C	S	R	Q_{n+1}	$\overline{Q}_{n+1}$	
0	0	×	×	×	1	1	Invalid
0	1	×	×	×	1	0	SET Direct
1	0	×	×	×	0	1	RESET Direct
1	1	0	×	×	Q_n	$\overline{Q}_n$	Keep
1	1	1	0	0	Q_n	$\overline{Q}_n$	Keep
1	1	1	0	1	0	1	RESET
1	1	1	1	0	1	0	SET
1	1	1	1	1	1	1	Invalid

أي نوع من أنواع المراجيح المتزامنة التي درسناها يمكن أن يكون مزوداً بأطراف دخل مباشر (Direct Inputs)، و يرمز لأطراف الدخل المباشر في هذه الحالات دائماً بـ $\overline{S}_d$ و $\overline{R}_d$ بغض النظر عن نوع المرجاح، كما هو موضح أدناه



مرجاح T

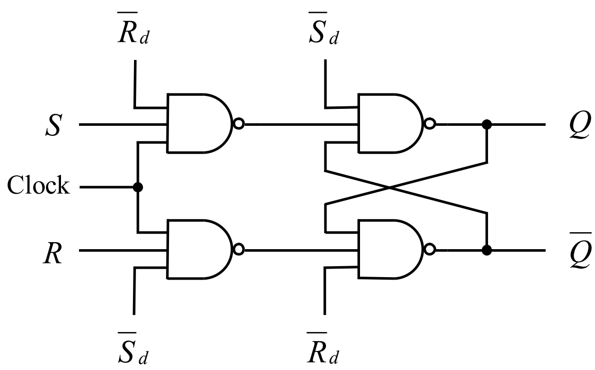
مزود بأطراف دخل مباشر



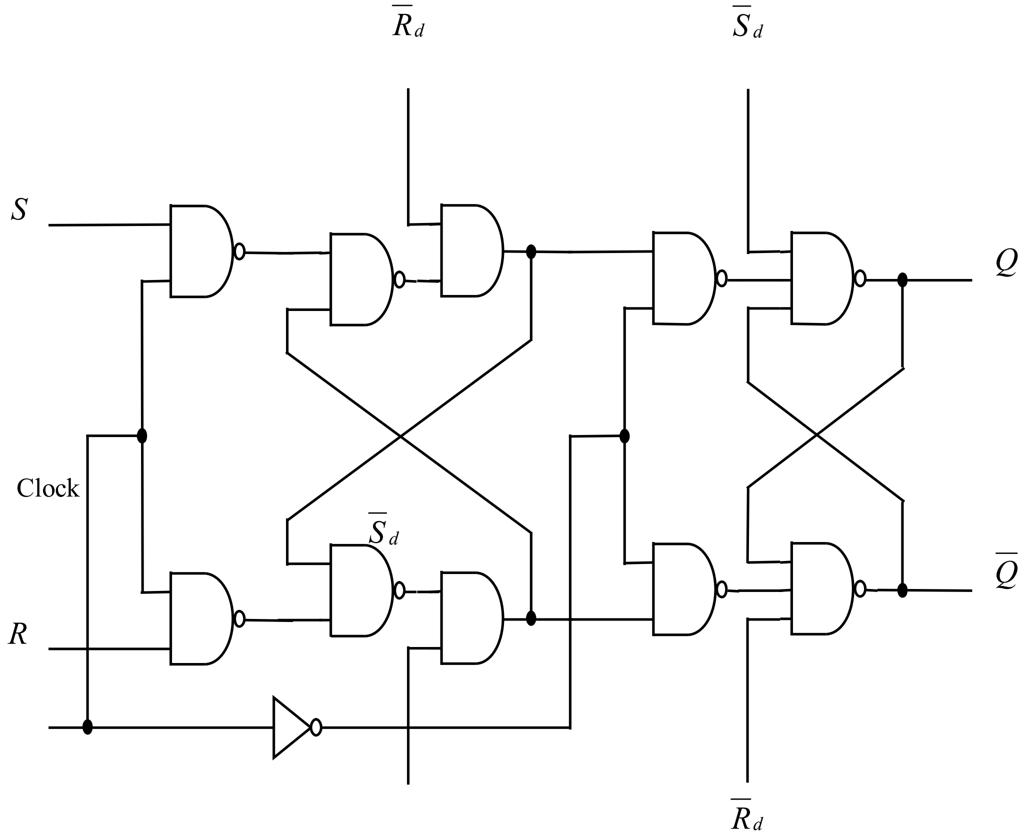
مرجاح JK

مزود بأطراف دخل مباشر

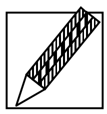
الشكل التالي يوضح كيفية ظهور أطراف الدخل المباشر في الدائرة المنطقية لمرجاح SR المتزامن



الشكل التالي يوضح كيفية ظهور أطراف الدخل المباشر في الدائرة المنطقية لمرجاح القائد–التابع



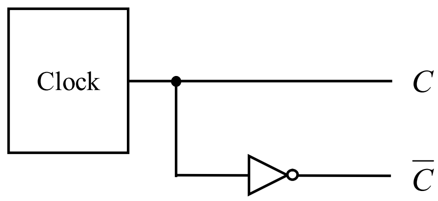
تدريب 1



ارسم الدائرة المنطقية لمرجاح JK مزوداً بأطراف دخل مباشر.

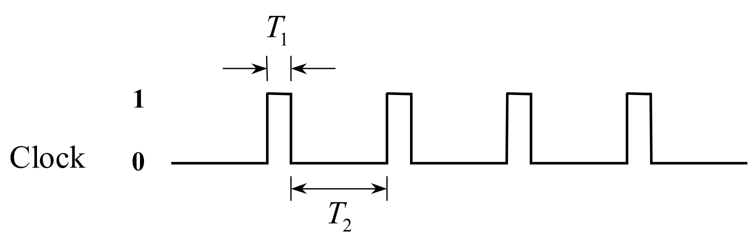
5.2 التزامن ثنائي الطور (Two-Phase Clocking)

في معظم الأنظمة الرقمية تستخدم مراجيح من نوع القائد–التابع (Master-Slave Flip Flops). و في هذا النوع من المراجيح، كما نعلم، تدخل إشارة التزامن (Clock) مباشرة إلى مرحلة القائد (Master)، و تدخل معكوسة إلى مرحلة التابع (Slave)، و يتم عكس إشارة التزامن باستخدام عاكس منطقي موجود داخل كل مرجاح. و يمكن الاستغناء عن كل هذه العواكس المنطقية إذا قمنا بعكس إشارة التزامن عند مصدرها باستخدام عاكس منطقي واحد، كما هو موضح أدناه



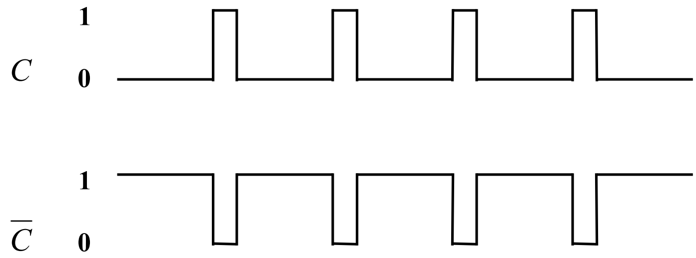
فنحصل على إشارتين؛ إشارة التزامن C و معكوسها $\overline{C}$. يتم بعد ذلك توزيع الإشارتين على أجزاء النظام الرقمي بحيث تدخل الإشارة C إلى مرحلة القائد (Master) و الإشارة $\overline{C}$ إلى مرحلة التابع (Slave) من كل مرجاح.

و لكن تظهر هنا مشكلة ناتجة عن شكل إشارة التزامن (Clock)، و الموضح أدناه

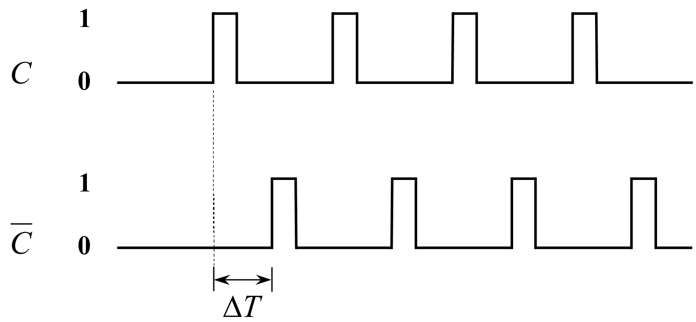


نلاحظ أن إشارة التزامن (Clock) عبارة عن سلسلة من النبضات الضيقة تفصل بينها فترات إنتظار واسعة نسبياً، أي أن الفترة الزمنية T_1 التي تكون فيها إشارة التزامن مرتفعة (High) أقصر بكثير من الفترة الزمنية T_2 التي تكون فيها الإشارة منخفضة

(Low). فإذا ما قمنا بعكس هذه الإشارة باستخدام عاكس منطقي نحصل على النتيجـــة الموضحة بالشكل التالي



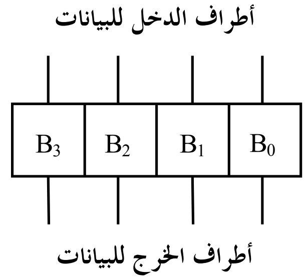
نلاحظ أن الإشارة $\overline{C}$ هنا مكونة من نبضات واسعة و فترات انتظار ضيقة، و مثل هذه الإشارة قد لا تصلح كإشارة تزامن نظراً إلى أن فترات الإنتظار الضيقة قد لا تكون كافية لتلاشي القيم غير المتوقعة. لذلك لا يتم عادة الحصول على الإشارة $\overline{C}$ بعكس الإشارة C باستخدام عاكس منطقي و إنما يتم الحصول عليها بعمل إزاحة زمنية (Time Shift) للإشارة C، كما هو موضح أدناه



تسمى الإشارة C بالطور الأول (Phase 1) من إشارة التزامن، و يرمز لها بالرمز ϕ_1، و تسمى الإشارة $\overline{C}$ بالطور الثاني (Phase 2) من إشارة التزامن، و يرمز لهـــا بـــالرمز ϕ_2. و يطلق على هذا الأسلوب في التزامن تسمية التزامن ثنائي الطور (Two-Phase Clocking).

3. المسجلات (Registers)

المسجل (Register) هو عبارة عن موقع تخزيني له القدرة على اختزان معلومة مكونة من عدة خانات. و الشكل التالي يوضح المخطط المنطقي لمسجل مكون من أربعة خانات (4-bit Register)

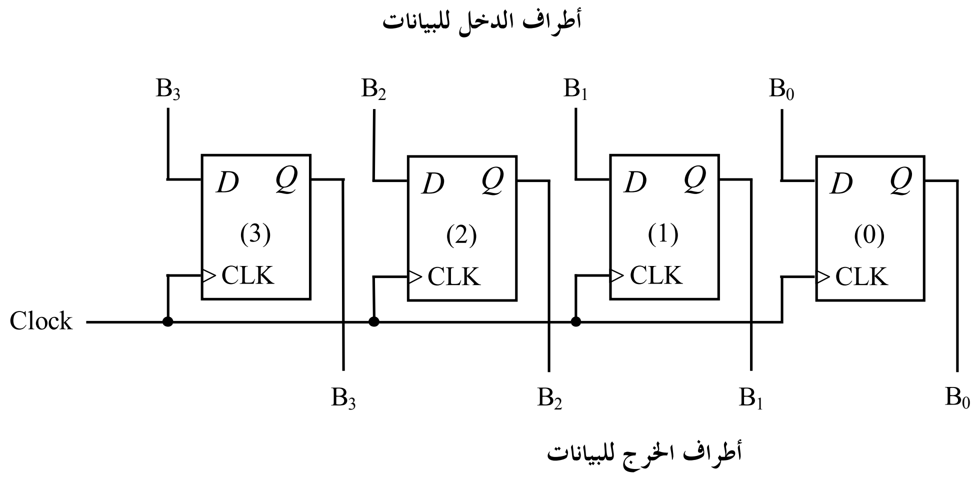


و العمليات التي يمكن إجراؤها على المسجلات هي:

1. الكتابة (Write)، أي تخزين معلومة في المسجل.
2. القراءة (Read)، أي إسترجاع معلومة مخزنة في المسجل.
3. نقل البيانات ما بين المسجلات (Register-to-Register Transfer).

1.3 بناء المسجلات

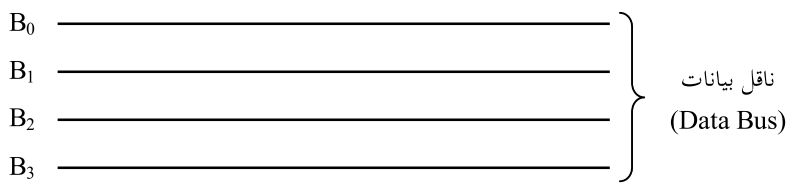
يتم بناء المسجلات باستخدام مراجيح D (D Flip Flops)، و نحتاج عدداً من المراجيح بعدد الخانات الثنائية (bits) المطلوب تخزينها. الشكل التالي يوضح الدائرة المنطقية لمسجل مكون من أربعة خانات (4-bit Register)



2.3 الكتابة في المسجلات و القراءة منها

(Write and Read Operations)

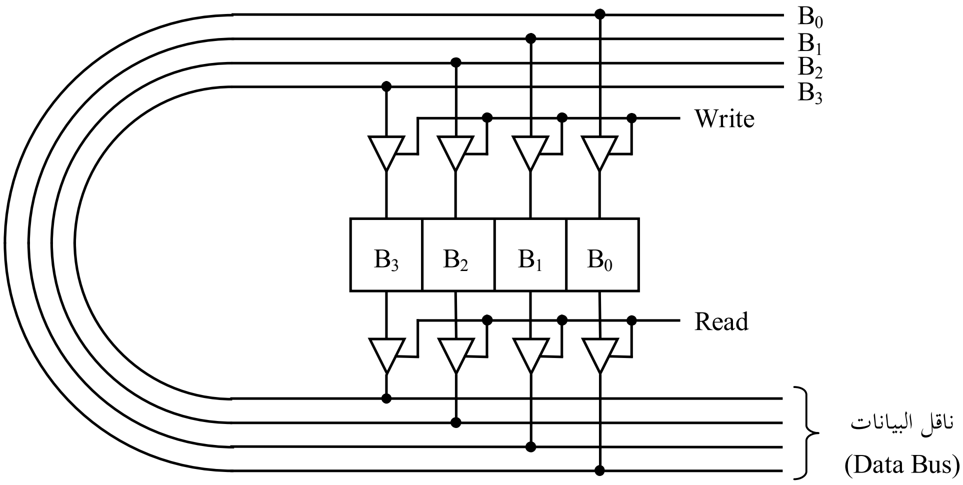
عند إجراء عملية كتابة (Write) في المسجل فإن المعلومة المطلوب تخزينها عـــادة مـــا تصل إلى المسجل من خلال ناقل بيانات (Data Bus)، و عنـــد إجـــراء عمليـــة قـــراءة (Read) من المسجل فإن المعلومة التي تم إسترجاعها عادة ما تنقل من المســـجل إلـــى الجهة المقصودة عبر ناقل البيانات (Data Bus) أيضاً. و ناقل البيانات هذا هو عبـــارة عن مجموعة من الموصلات المتوازية كل منها يحمل bit واحد فقط مـــن البيانـــات، و الشكل التالي يوضح ناقل بيانات ذا أربعة خانات (4-bit Data Bus)



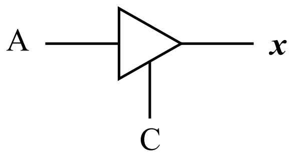
لاحظ أنه من الناحية الكهربائية لابد من وجود موصل خامس في ناقل البيانات ذو الخانات الأربعة الموضح أعلاه، و هذا الموصل الخامس هو الموصل الأرضي

(Ground أو GND) الذي يعتبر مرجع قياس الجهود بالنسبة لبقية الموصلات. و لكن لا يتم عادة توضيح هذا الموصل الأرضي و إنما يُفهم وجوده ضمناً، و ذلك كنوع من التبسيط.

هذا و يتم ربط كل من أطراف الدخل للبيانات و أطراف الخرج للبيانات للمسجل بناقل البيانات باستخدام عوازل ثلاثية الحالة (Tristate Buffers)، كما هو موضح أدناه



و العازل ثلاثي الحالة (Tristate Buffer) هو عبارة عن بوابة منطقية لها طرف دخل A و طرف خرج x و طرف تحكم C، كما هو موضح أدناه



عند وضع القيمة 1 في طرف التحكم C يمر الدخل كما هو إلى الخرج، أي يكـــون $x =$ A، أما عند وضع القيمة 0 في طرف التحكم C تدخل البوابة في الحالة الثالثة أي حالـــة

المعاوقة العالية (High Impedance) ، و فيها يتم عزل خرج البوابة عن دخلها بمعاوقة عالية.

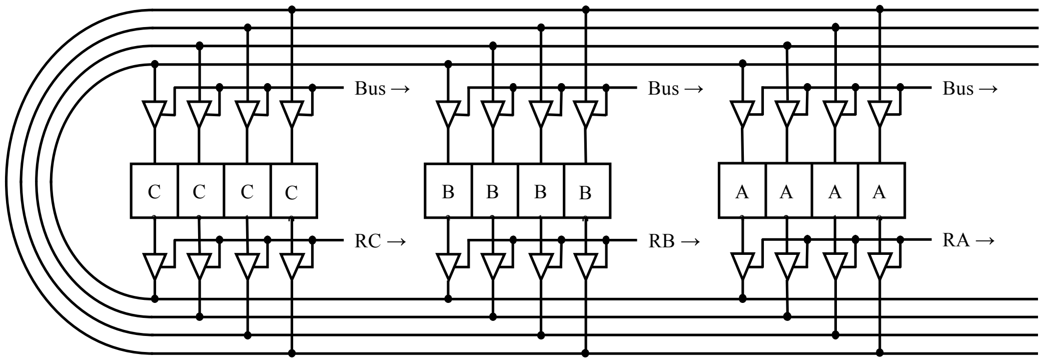
- لإجراء عملية كتابة (Write) للبيانات الظاهرة على الناقل (Bus) في المسجل نقوم بجعل الإشارة Write مساوية 1، فيتم توصيل أطراف الدخل للمسجل مع الناقل، و تنتقل البيانات الموجودة على الناقل إلى داخل المسجل و يتم إختزانها. بعد ذلك يجب إعادة الإشارة Write إلى 0 مرة أخرى لفصل أطراف الدخل للمسجل عن الناقل، و ذلك لإخلاء الناقل بحيث يكون متاحاً للاستخدام في عمليات نقل بيانات أخرى.

- لإجراء عملية قراءة (Read) للبيانات المخزنة في المسجل نقوم بجعل الإشارة Read مساوية 1، فيتم توصيل أطراف الخرج للمسجل مع الناقل، و تظهر البيانات المخزنة في المسجل على الناقل و تكون متاحة لقراءتها من الناقل بواسطة أي جهة طالبة لها. بعد ذلك يجب إعادة الإشارة Read إلى 0 مرة أخرى لفصل أطراف الخرج للمسجل عن الناقل، و ذلك لإخلاء الناقل بحيث يكون متاحاً للاستخدام في عمليات نقل بيانات أخرى.

3.3 نقل البيانات بين المسجلات

(Register-to-Register Transfer)

لنقل البيانات بين مجموعة من المسجلات يتم ربط تلك المسجلات بناقل مشترك (Common Bus)، كما هو موضح أدناه



لنقل البيانات من مسجل إلى آخر يتم إستخدام الناقل (Bus) كوسيط، حيث يتم قراءة محتويات المسجل الأول لتظهر تلك المحتويات على الناقل، بعد ذلك يتم قراءتها من الناقل بواسطة المسجل الثاني. مثلاً

- لإجراء عملية نقل البيانات $RB \rightarrow RA$ (أي نسخ محتويات المسجل RA للمسجل RB).

1. نجعل الإشارة $RA \rightarrow Bus$ مساوية 1 فتظهر محتويات المسجل RA على الناقل.
2. نجعل الإشارة $Bus \rightarrow RB$ مساوية 1 فتنتقل البيانات الظاهرة على الناقل إلى المسجل RB.
3. نعيد الإشارتين $RA \rightarrow Bus$ و $Bus \rightarrow RB$ إلى 0 مرة أخرى لإخلاء الناقل.

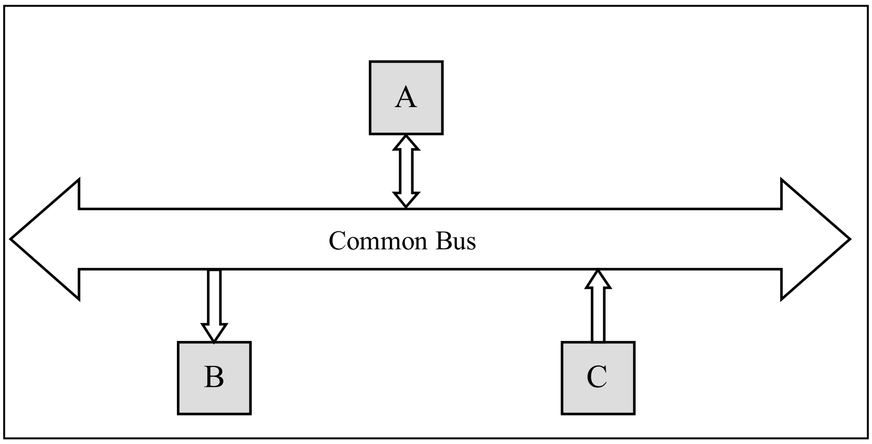
- لإجراء عملية نقل البيانات $RB \rightarrow \begin{cases} RA \\ RC \end{cases}$ (أي نسخ محتويات المسجل RB للمسجلين RA و RC).

1/ نجعل الإشارة $RB \rightarrow Bus$ مساوية 1 فتظهر محتويات المسجل RB على الناقل.

2/ نجعل الإشارتين $Bus \rightarrow RA$ و $Bus \rightarrow RC$ مساويتين 1 فتنتقل البيانات الظاهرة على الناقل إلى كلا المسجلين RA و RC.

3/ نعيد الإشارات $RB \rightarrow Bus$ و $Bus \rightarrow RA$ و $Bus \rightarrow RC$ إلى 0 مرة لإخلاء الناقل.

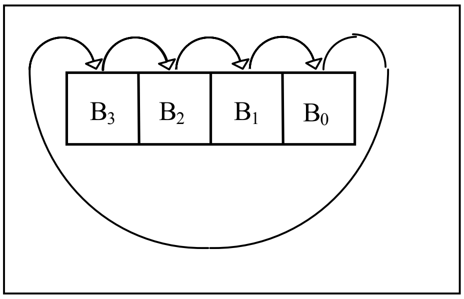
هذا و من الشائع في الأنظمة الرقمية إستخدام ناقل مشترك (Common Bus) لنقل البيانات بين الأجزاء المختلفة للنظام الرقمي، كما هو موضح أدناه



هذا و يعتبر عرض الناقل، أي عدد الـــ bits التي يحملها، من العوامل الهامة جداً في تحديد سرعة عمل النظام الرقمي، فكلما زاد عرض الناقل أمكن نقل كمية أكبر من البيانات عبره في عملية النقل الواحدة.

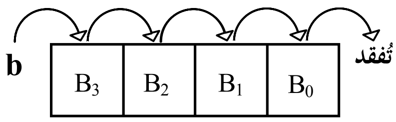
4.3 مسجلات الإزاحة (Shift Registers)

مسجل الإزاحة (Shift Register) هو عبارة عن مسجل يستطيع، إضافة إلـــى العمليـــات السابقة، عمل إزاحة للبيانات الموجودة بداخله بمقدار خانة واحـــدة أو أكثـــر يمينـــاً أو يساراً. و هناك عدة أنواع من الإزاحة



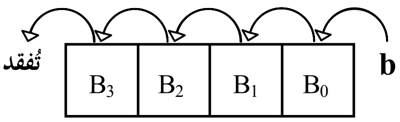
- **الإزاحة إلى اليمين (Shift Right)**

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليمين حيث تُفقد الخانة الـــدنيا B_0 و تحـــل الخانة B_1 محلها، و تحل الخانة B_2 محل الخانة B_1، و تحل الخانة B_3 محل الخانـــة B_2، و يتم إدخال bit من الخارج إلى الخانة العليا (MSB) ليحل محـــل الخانـــة B_3. كما هو موضح أدناه



- **الإزاحة إلى اليسار (Shift Left):**

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليسار حيث تُفقد الخانة العليـــا B_3 و تحـــل الخانة B_2 محلها، و تحل الخانة B_1 محل الخانة B_2، و تحل الخانة B_0 محل الخانـــة B_1، و يتم إدخال bit من الخارج إلى الخانة الدنيا (LSB) ليحل محـــل الخانـــة B_0. كما هو موضح أدناه

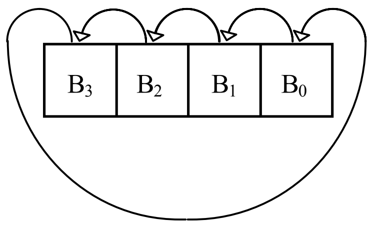


- **الإزاحة الدورانية إلى اليمين (Rotate Right)**

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليمين و لكن لا يحدث أي فقد أو إدخال من الخارج، حيث إن الخانة الدنيا B_0 تحل محل الخانة العليا B_3. كما هو موضح أدناه

- **الإزاحة الدورانية إلى اليسار (Rotate Left):**

هنا تتم الإزاحة بمقدار خانة واحدة إلى اليسار بدون أي فقد أو إدخال من الخـــارج، حيث أن الخانة العليا B_3 تحل محل الخانة الدنيا B_0. كما هو موضح أدناه



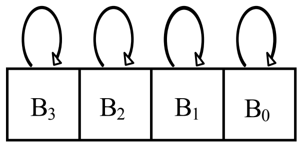
لاحظ أن كل أنواع الإزاحة الموضحة أعلاه تحدث بصورة متزامنة، أي مرتبطة بإشارة التزامن (Clock)، فمع كل نبضة من نبضات التزامن تحدث إزاحة بمقدار خانة واحـــدة في الإتجاه المحدد، و تستمر الإزاحة ما دامت إشارة التزامن مستمرة.

و هناك عمليات أخرى، بخلاف عملية الإزاحة بأنواعها، يمكن إجراؤها على مســجلات الإزاحة مثل:

- **التوقف (Hold):**

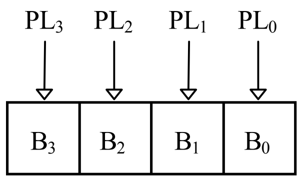
المقصود هنا هو إيقاف عملية الإزاحة الجارية بصورة مؤقتة، و يمكن أن يتم ذلك بالطبع بإيقاف إشارة التزامن (Clock)، حيث أن الإزاحة مرتبطة بإشارة التزامن

كما ذكرنا من قبل، و لكن الأسلوب الأفضل هنا هو أن يتم ذلك بأن تحل كل خانة من خانات المسجل محل نفسها، كما هو موضح أدناه



- **التعبئة على التوازي (Parallel Load):**

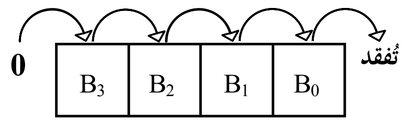
المقصود هنا هو تعبئة المسجل بالبيانات من الخارج إستعداداً للبدء بعملية الإزاحة، كما هو موضح أدناه

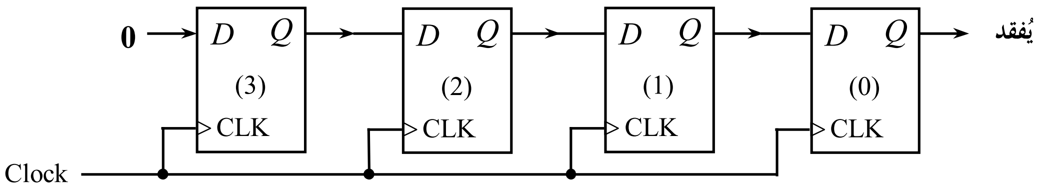


بناء مسجلات الإزاحة

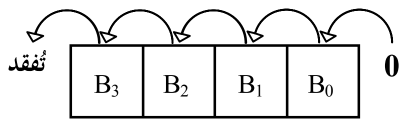
كما ذكرنا من قبل فإن بناء المسجلات يتم باستخدام مراجيح D (D Flip Flops)، و نحتاج عدداً من المراجيح بعدد الخانات الثنائية (bits) المكونة للمسجل. و في مسجلات الإزاحة يتم ربط المراجيح مع بعضها البعض على التوالي بحيث يكون خرج كل مرجاح دخلاً للمرجاح المجاور له و ذلك حسب إتجاه الإزاحة المطلوب.

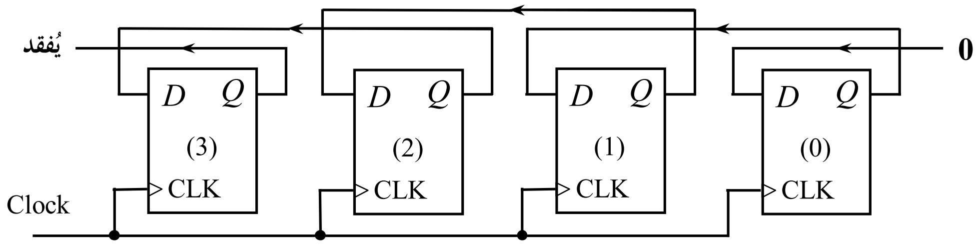
الشكل التالي يوضح المخطط المنطقي و الدائرة المنطقية لمسجل إزاحة إلى اليمين، مع الملء بأصفار، مكون من أربعة خانات (4-bit Shift Right Zero-Fill Register)





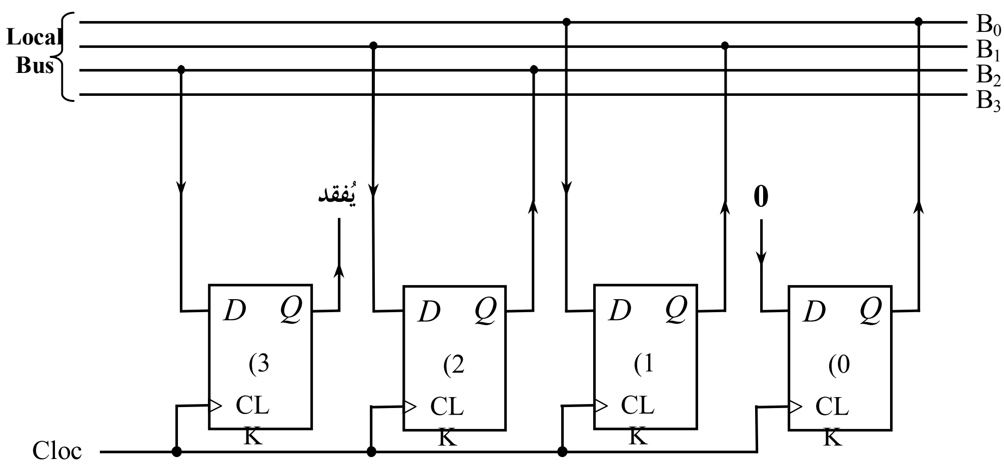
الشكل التالي يوضح المخطط المنطقي و الدائرة المنطقية لمسجل إزاحة إلى اليسار، مع الملء بأصفار، مكون من أربعة خانات (4-bit Shift-Left Zero-Fill Register)



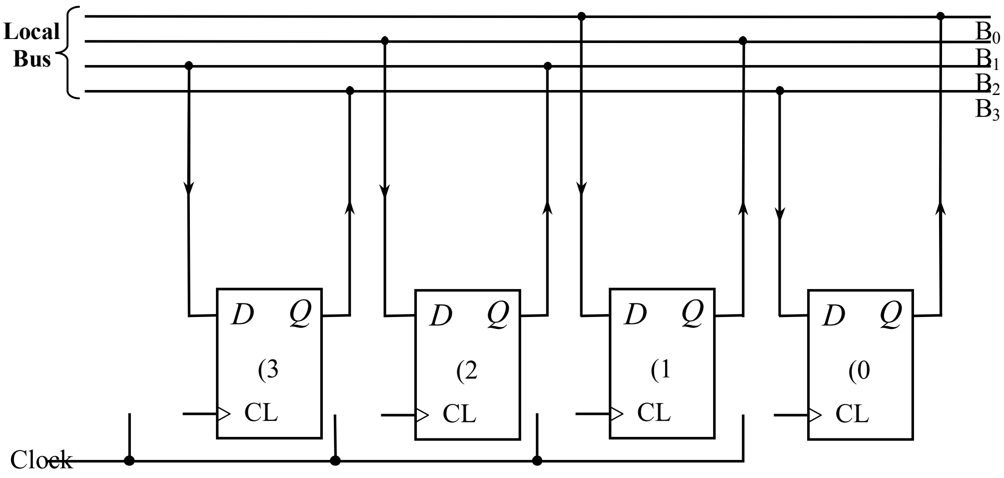


نلاحظ أن الأسلوب المستخدم في رسم الدائرة أعلاه يجعل من الصعب متابعتها و فهم عملها، خصوصاً إذا كانت الدائرة أكبر و أكثر تعقيداً. لذلك سنستخدم في رسم دوائر مسجلات الإزاحة أسلوباً أفضل يجعل من السهل متابعتها و فهم عملها، بل و يسهل بناءها في المعمل أيضاً، و نستخدم في هذا الأسلوب ناقلاً محلياً (Local Bus) في ربط

المراجيح المكونة للمسجل مع بعضها البعض. و نؤكد هنا على أن الناقل المستخدم هنا هو ناقل محلي (Local)، بمعنى أنه مضمن داخل دائرة مسجل الإزاحة، و يختلف عن الناقل المشترك (Common Bus) المستخدم في نقل البيانات بين أجزاء النظام الرقمي الذي أشرنا إليه من قبل. و الشكل التالي يوضح الدائرة المنطقية لمسجل الإزاحة إلى اليسار، مع الملء بأصفار، المكون من أربعة خانات (4-bit Shift-Left Zero-Fill Register) بعد إعادة رسمها بالأسلوب الجديد



و بتعديل طفيف على الدائرة الموضحة أعلاه يمكن تحويل المسجل إلى مسجل إزاحة دورانية إلى اليسار مكون من أربعة خانات (4-bit Rotate Left Register)، كما هو موضح بالشكل التالي

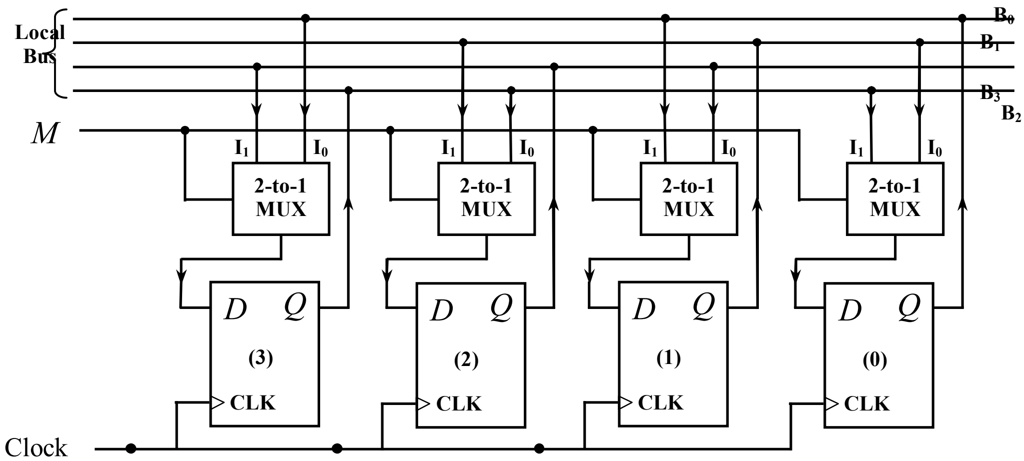


الإزاحة في الاتجاهين

المطلوب الآن تصميم مسجل إزاحة مكون من أربعة خانات و يستجيب لإشارة تحكم M، بحيث يقوم بالإزاحة الدورانية إلى اليمين (Rotate Right) عندما تكون $M = 0$، و بالإزاحة الدورانية إلى اليسار (Rotate Left) عندما تكون $M = 1$.

نعلم أن الإزاحة إلى اليمين تتطلب ربط المراجيح مع بعضها البعض بطريقة معينة، و الإزاحة إلى اليسار تتطلب ربطها بطريقة أخرى مختلفة. فكيف يمكن ربط المراجيح بكلا الطريقتين في وقت واحد ثم اختيار إحداهما بناء على قيمة إشارة التحكم M؟

يتم ذلك باستخدام دوائر دامج من نوع 1 إلى 2 (2-to-1 MUX's) كما هو موضح بالشكل التالي:



استخدمنا هنا دوائر دامج من نوع 1 إلى 2 بعدد المراجيح المكونة للمسجل، حيث يحدد كل دامج دخل المرجاح المقابل له بناء على قيمة إشارة التحكم M التي تدخل إلى طرف الاختيار (Select Line) لكل دامج. فعندما تكون $M = 0$ يتم توصيل الطرف I_0 لكل دامج مع طرف الدخل للمرجاح المقابل، و يؤدي هذا لربط المراجيح بحيث تكون الإزاحة لليمين، و عندما تكون $M = 1$ يتم توصيل الطرف I_1 لكل دامج مع طرف الدخل للمرجاح المقابل، و يؤدي هذا لربط المراجيح بحيث تكون الإزاحة لليسار.

تدريب 2

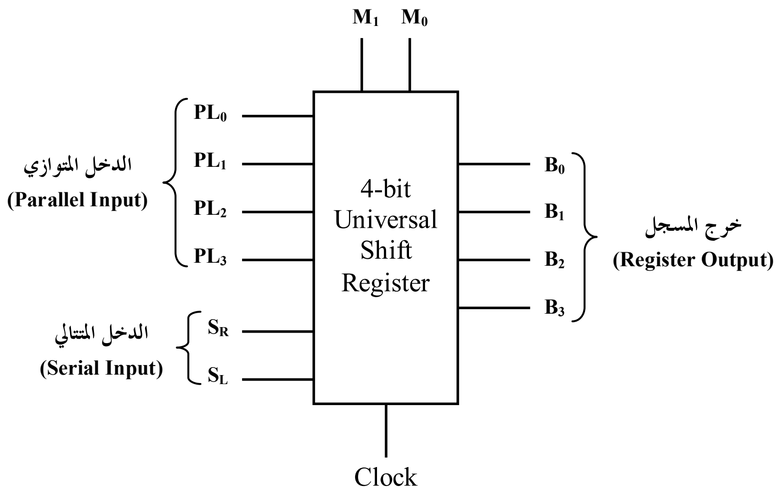
صمم مسجل إزاحة من أربعة خانات (4 bits) يستجيب لإشارة تحكم M، فيقبل دخلاً متوازياً عندما تكون $M = 0$، و يقوم بالإزاحة إلى اليمين مع الملء بأصفار عندما تكون $M = 1$.

مسجل الإزاحة العام (Universal Shift Register)

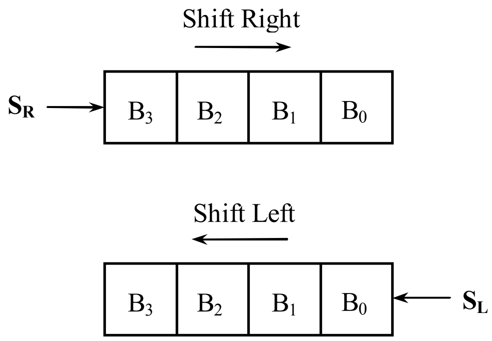
مسجل الإزاحة العام هو عبارة عن مسجل إزاحة يمكن أن يقبل دخلاً متوازياً (Parallel Input) أو دخلاً متتالياً (Serial Input)، و يقوم بالإزاحة يميناً أو يساراً أو يتوقف عن الإزاحة بناء على قيم إشارتي تحكم هما M_0 و M_1، كما هو موضح بالجدول التالي:

M_1	M_0	Operation
0	0	Hold
0	1	Shift Right
1	0	Shift Left
1	1	Parallel Load

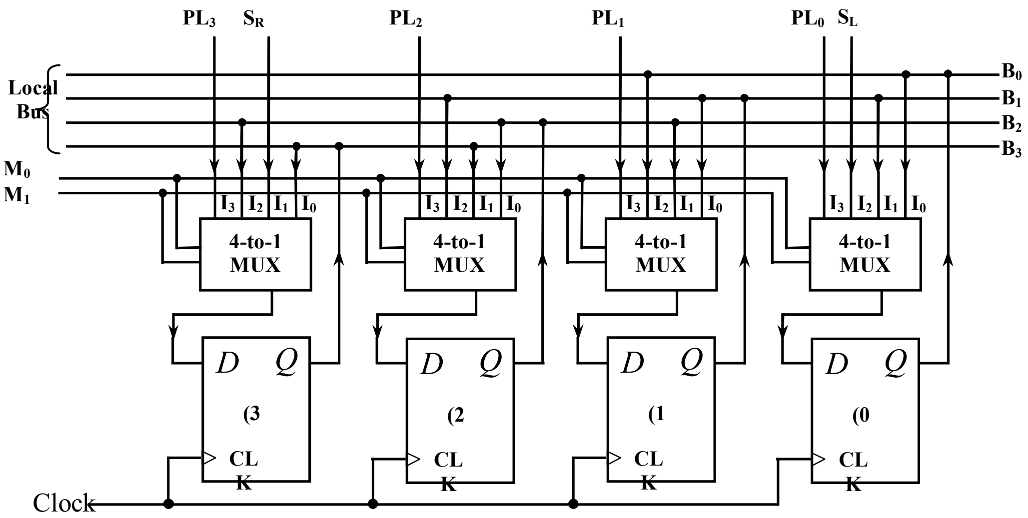
و الشكل التالي يوضح المخطط المنطقي لمسجل إزاحة عام ذو أربعة خانات (4-bit Universal Shift Register)



الدخل المتوازي (Parallel Input) يتم عبر أطراف الدخل المتوازي PL_0، PL_1، PL_2، و PL_3. أما الدخل المتتالي (Serial Input) فيتم أثناء عملية الإزاحة إلى اليمين عبر طرف الدخل S_R إلى الخانة العليا (MSB) من المسجل، و أثناء عملية الإزاحة إلى اليسار عبر طرف الدخل S_L إلى الخانة الدنيا (LSB) من المسجل، كما هو موضح بالشكل التالي



نظراً إلى أن لدينا في هذا المسجل أربع طرق مختلفة لربط المراجيح، يجب أن تكون جميعاً موجودة و أن يتم إختيار واحدة منها بناء على قيم إشارتي التحكم M_0 و M_1، فإنه يتم استخدام دوائر دامج من نوع 4 إلى 1 (4-to-1 MUX's) كما هو موضح بالشكل التالي الذي يمثل الدائرة المنطقية لمسجل إزاحة عام ذو أربعة خانات



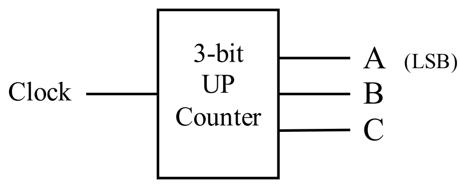
هذا و يمكن استخدام مسجل الإزاحة العام في تحويل البيانات من تـــوازي إلـــى تتـــالي (Parallel to Serial) أو من تتالي إلى توازي (Serial to Parallel). فعند التحويـــل مـــن توازي إلى تتالي يتم إدخال الـــ bits للمسجل على التوازي بإجراء عملية تعبئـــة علـــى

التوازي (Parallel Load)، ثم يتم إجراء إزاحة إلى اليمين و الحصول على الـ bits على التتالي واحداً تلو الآخر بدءاً بالخانة الدنيا (LSB) عبر طرف الخرج B_0، أو إجراء إزاحة إلى اليسار و الحصول على الـ bits على التتالي واحداً تلو الآخر بدءاً بالخانة العليا (MSB) عبر طرف الخرج B_3. و عند التحويل من تتالي إلى توازي يتم إدخال الـ bits إلى المسجل واحداً تلو الآخر بدءاً بالخانة الدنيا (LSB) عبر الطرف S_R أثناء عملية الإزاحة إلى اليمين، أو بدءاً بالخانة العليا (MSB) عبر الطرف S_L أثناء عملية الإزاحة إلى اليسار، و بعد إكتمال دخول الـ bits يتم قراءتها على التوازي عبر أطراف الخرج للمسجل.

4. العدّادات (Counters)

العدّاد (Counter) هو عبارة عن دائرة منطقية تتابعية لها القدرة على العد ثنائياً بترتيب معين. و ترتيب العد قد يكون ترتيباً تصاعدياً (Up Counting)، أو قد يكون ترتيباً تنازلياً (Down Counting)، أو قد يكون بأي ترتيب آخر. كل قيمة يصل إليها العدّاد أثناء عملية العد تسمى حالة (State)، و ينتقل العدّاد من حالة إلى أخرى من حالاته مع نبضات التزامن (Clock) و بترتيب معين. أي أن كل نبضة من نبضات التزامن تنقل العدّاد من الحالة التي هو فيها إلى الحالة التي تليها في ترتيب العد. و يمكن أن يبدأ العدّاد العد من أي حالة من حالاته، و يطلق على الحالة التي يبدأ العد منها تسمية الحالة الإبتدائية (Initial State).

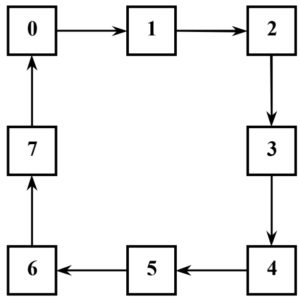
الشكل التالي يوضح المخطط المنطقي لعدّاد تصاعدي ذا خانات ثلاثة (3-bit Up Counter)



و الجدول التالي يوضح تسلسل العد (Counting Sequence) للعدّاد

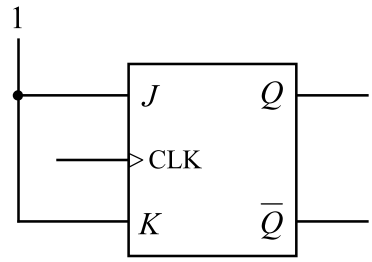
C	**B**	**A**	State
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	0
0	0	1	1
0	1	0	2
⋮	⋮	⋮	⋮

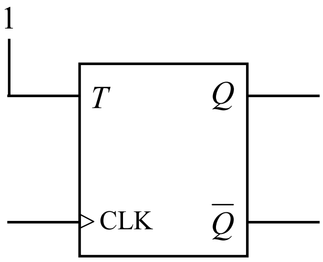
<u>لاحظ</u> أن للعدّاد ثمانية حالات هي 0، 1، 2، 3، 4، 5، 6، 7 يمر بهـا العـدّاد بترتيـب تصاعدي إبتداءاً من الحالة 0 التي تعتبر الحالة الإبتدائية (Initial State) للعدّاد. لاحـظ أيضاً أن الخانة الدنيا (LSB) للعدّاد A تعكس حالتها مع كل نبضة من نبضات التـزامن، في حين أن الخانة الثانية B تعكس حالتها كل نبضتين، و الخانة الثالثة C تعكس حالتهـا كل أربعة نبضات. لاحظ أيضاً أنه بعد وصول العدّاد إلى آخر حالة من حالاتـه فـي تسلسل العد يعود مرة أخرى إلى الحالة الإبتدائية و يكرر العملية طالما كانـت نبضـات التزامن (Clock) مستمرة. هذا و يمكن توضيح حالات العدّاد و ترتيـب المـرور بهـا باستخدام مخطط يسمى بمخطط الحالات (State Diagram)، كما هو موضح أدناه



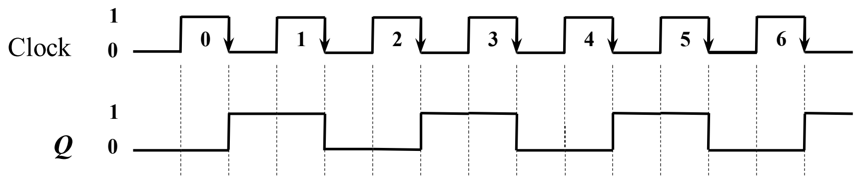
1.4 بناء العدّادات

يستخدم في بناء العدّادات مراجيح JK أو مراجيح T في وضع التردد بين حالتين (Toggle Mode)، كما هو موضح أدناه





و في هذا الوضع يقوم المرجاح بعكس حالته مع كل نبضة من نبضات التزامن، كما هو موضح أدناه



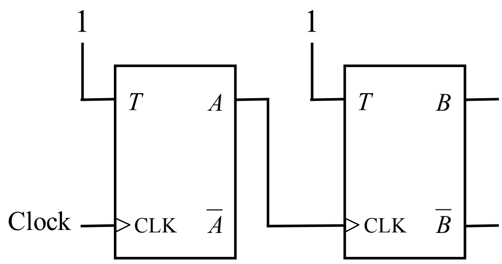
لاحظ أن الإشارة Q الخارجة من المرجاح يمكن إعتبارها أيضاً إشارة تزامن، و لكن إشارة التزامن الخارجة من المرجاح ترددها هو نصف تردد إشارة التزامن الداخلة إليه. أي أن المرجاح قد قام بقسمة تردد إشارة التزامن على 2، أو

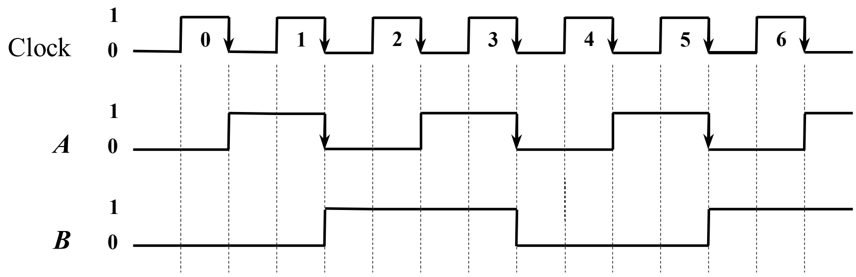
$$f_Q = \frac{1}{2} f_C$$

حيث f_C هو تردد إشارة التزامن (Clock)، و f_Q هو تردد الإشارة Q الخارجة من المرجاح.

2.4 العد تصاعدياً (Up Counting)

إذا قمنا بإدخال الإشارة الخارجة من المرجاح الأول كإشارة تزامن إلى مرجاح ثاني من نفس النوع فإن المرجاح الثاني سيقوم بقسمة تردد تلك الإشارة على 2 أيضاً، كما هو موضح أدناه

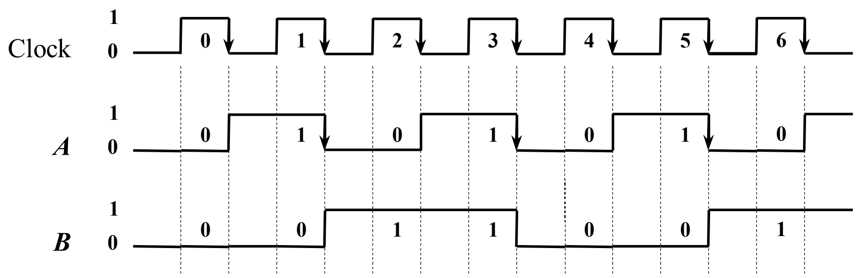




$$f_B = \frac{1}{2} f_A = \frac{1}{4} f_{Clock}$$

<u>لاحظ</u> أن المرجاح الأول A يعكس حالته مع كل نبضة من نبضات التزامن، و المرجاح الثاني B يعكس حالته كل نبضتين. و بناء عليه تصلح A أن تكون الخانة الدنيا (LSB)، و تصلح B أن تكون الخانة الثانية، في عدّاد تصاعدي ذي خانتين (2-bit Up Counter)، أي أن الدائرة المنطقية أعلاه تمثل دائرة عدّاد تصاعدي ذي خانتين.

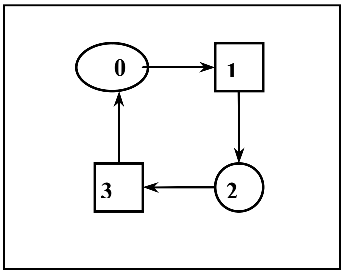
هذا و يمكن الحصول على تسلسل العد للعدّاد من مخطط التزامن (Timing Diagram) له كالتالي



و عليه يكون تسلسل العد (Counting Sequence) للعدّاد هو

B	A	State
0	0	0
0	1	1
1	0	2
1	1	3
0	0	0
0	1	1
1	0	2
⋮	⋮	⋮

و مخطط الحالات (State Diagram) للعدّاد هو

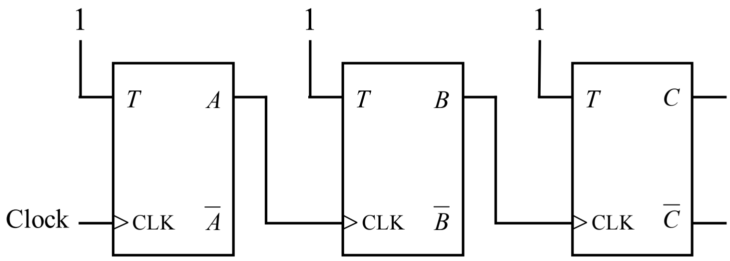


<u>**مثال:**</u>

صمم عدّاداً تصاعدياً ذا خانات ثلاثة (3-bit Up Counter) و أرسم مخطط التزامن له، ثم وضح تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العدّاد العد من الحالة 3.

<u>**الحل:**</u>

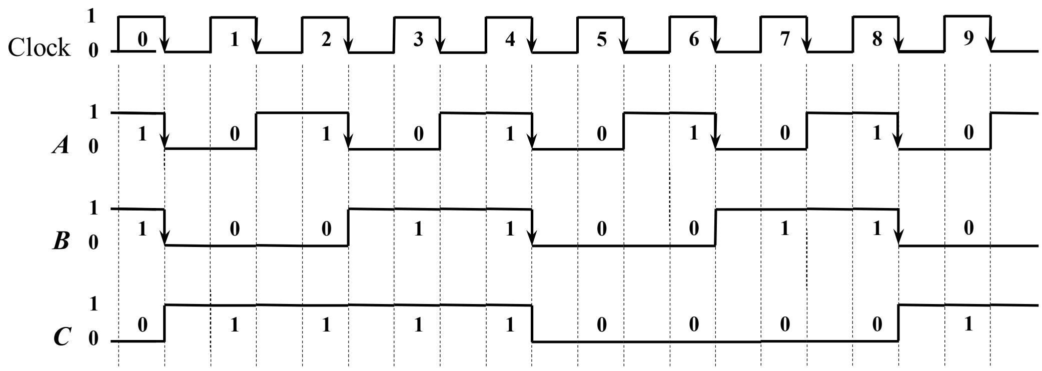
نحتاج عدداً من مراجيح T بعدد خانات العدّاد، أي ثلاثة مراجيح، و نقوم بإدخال الخرج غير المعكوس لكل مرجاح كإشارة تزامن للمرجاح الذي يليه، كما هو موضح أدناه



لرسم مخطط التزامن نحتاج لمعرفة الحالة الإبتدائية لكل مرجاح من المراجيح الثلاثة، و يمكن معرفتها من الحالة الإبتدائية للعدّاد كالتالي

$$\begin{array}{ccccccc} & & & C & B & A & \\ 3 & = & (& 0 & 1 & 1 &)_2 \end{array}$$

أي أن الحالة الإبتدائية لكل من A و B هي 1، و الحالة الإبتدائية لـــ C هي 0. و عليه يمكن رسم مخطط التزامن للعدّاد كالتالي

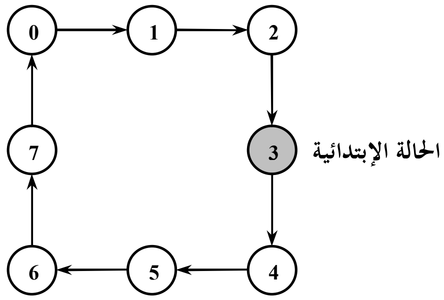


و من مخطط التزامن يمكن إيجاد تسلسل العد (Counting Sequence)

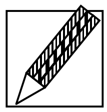
C	B	A	State
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
⋮	⋮	⋮	⋮

لاحظ أنه رغم أن العدّاد قد بدأ العد من الحالة 3 إلا أن ترتيب العد تصاعدي كما هو مطلوب.

و مخطط الحالات (State Diagram) هو



تدريب 3:

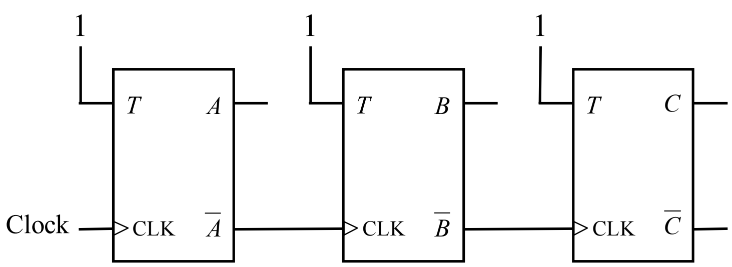


صمم عدّاداً تصاعدياً ذا خانات أربعة (4-bit Up Counter) و ارسم مخطط التزامن له، ثم وضح تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العدّاد العد من الحالة 10.

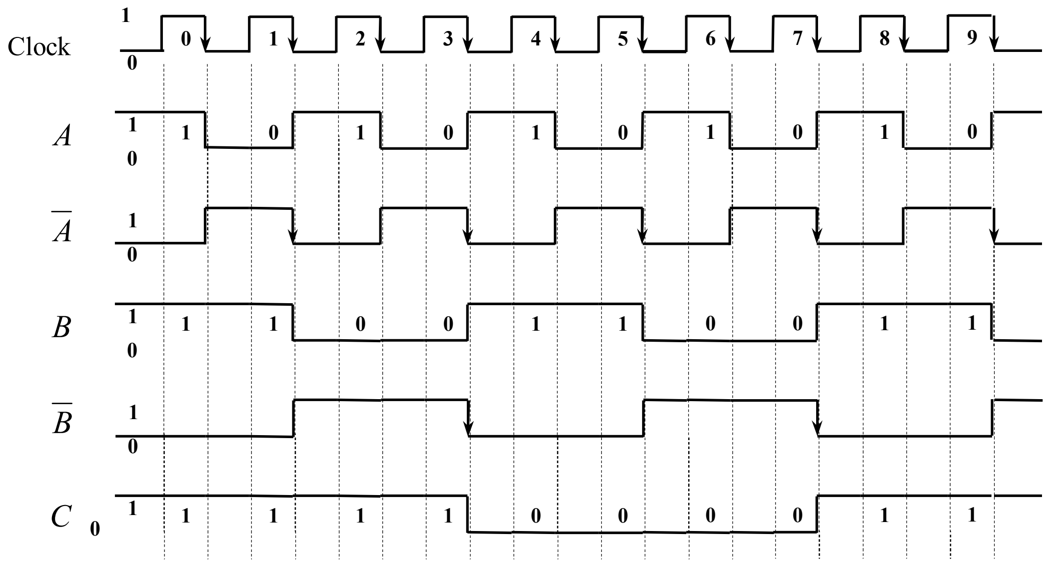
3.4 العد تنازلياً (Down Counting)

إذا استخدمنا الخرج المعكوس لكل مرجاح كإشارة تزامن للمرجاح الذي يليه فإن العـــدّاد الناتج يقوم بالعد تنازلياً.

الشكل التالي يوضح الدائرة المنطقية لعدّاد تنازلي مكون من ثلاثة خانات (3-bit Down Counter)



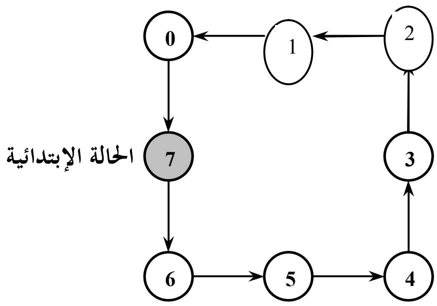
فإذا قمنا برسم مخطط التزامن للعدّاد مفترضين أنه بدأ العد من الحالة $(111)_2 = 7$ نحصل على



و من مخطط التزامن نجد أن تسلسل العد للعدّاد هو

C	B	A	State
1	1	1	7
1	1	0	6
1	0	1	5
1	0	0	4
0	1	1	3
0	1	0	2
0	0	1	1
0	0	0	0
1	1	1	7
1	1	0	6
⋮	⋮	⋮	⋮

أي أن العدّاد يقوم فعلاً بالعد تنازلياً، كما هو متوقع، ومخطط الحالات (State Diagram) له هو

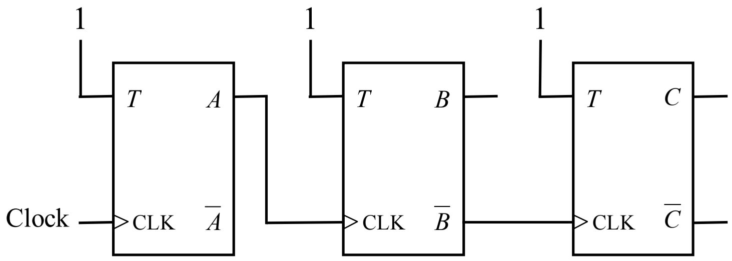


تدريب 4

صمم عدّاداً تنازلياً ذا خانات اربعة (4-bit Down Counter) و أرسم مخطط التزامن له، ثم وضح تسلسل العد و مخطط الحالات، و ذلك إذا بدأ العدّاد العد من الحالة 12.

مثال:

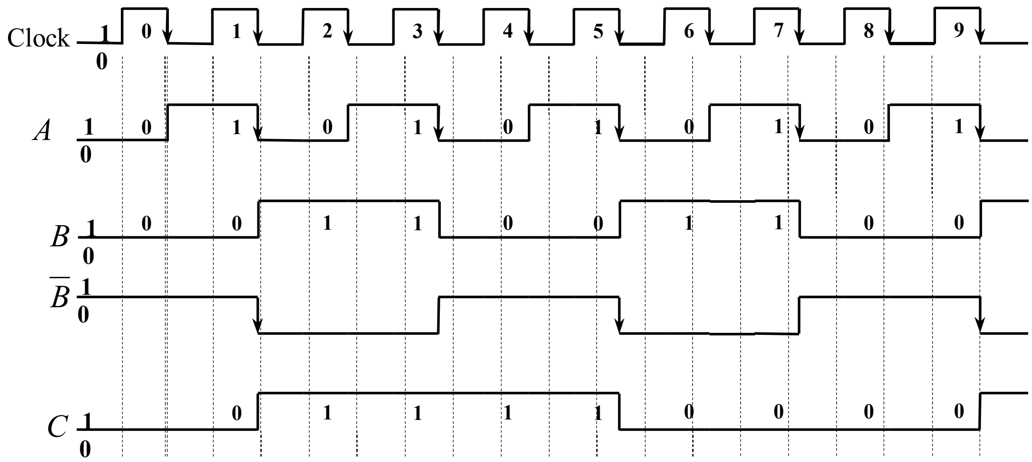
ارسم مخطط التزامن للعدّاد الذي توضحه دائرته المنطقية أدناه إذا بدأ العد من الحالة 0، ثم وضح تسلسل العد و أرسم مخطط الحالات له.



الحل:

من الواضح أن العدّاد هنا ليس عدّاداً تصاعدياً و لا هو عدّاد تنازلي.

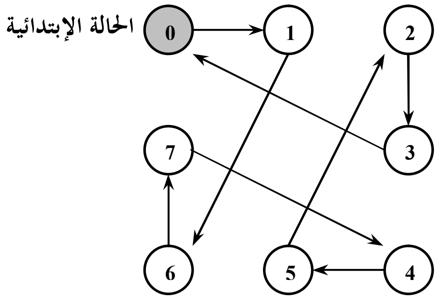
الحالة الابتدائية هنا هي $0 = (000)_2$



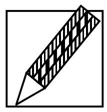
و من مخطط التزامن نجد أن تسلسل العد للعدّاد هو

C	B	A	State
0	0	0	0
0	0	1	1
1	1	0	6
1	1	1	7
1	0	0	4
1	0	1	5
0	1	0	2
0	1	1	3
0	0	0	0
0	0	1	1
⋮	⋮	⋮	⋮

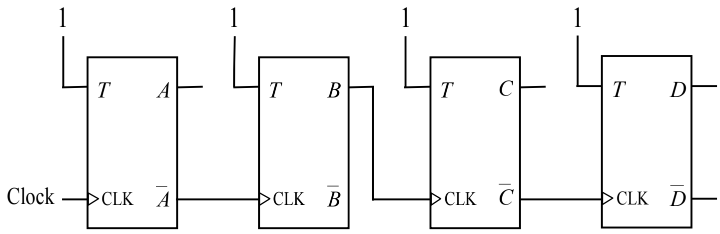
ومخطط الحالات (State Diagram) للعدّاد هو



تدريب 5



أرسم مخطط التزامن للعدّاد الموضحة دائرته المنطقية أدناه إذا بدأ العد من الحالة 5، ثم وضح تسلسل العد و أرسم مخطط الحالات له.



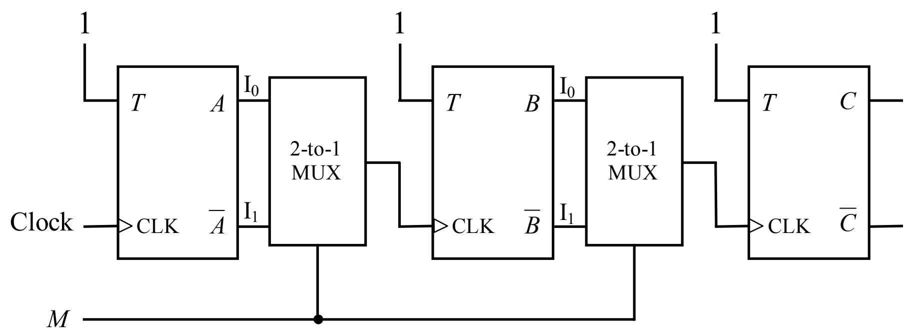
نلاحظ في جميع العدّادات التي قمنا بدراستها حتى الآن أن إشارة التزامن الداخلة للمراجيح المختلفة المكونة للعدّاد ليست واحدة، لذلك يسمى هذا النوع من العدّادات بالعدّادات غير المتزامنة (Asynchronous Counters)، و تسمى أيضاً Ripple Counters. هذا و يوجد نوع آخر من العدّادات هي العدّادات المتزامنة (Synchronous Counters)، و فيها تكون إشارة التزامن مشتركة ما بين جميع المراجيح المكونة للعدّاد، و دراسة هذا النوع من العدّادات خارج نطاق هذا المقرر.

4.4 العد في الاتجاهين (Up/Down Counting)

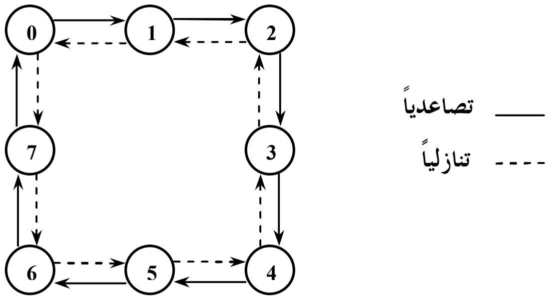
المطلوب الآن هو تصميم عدّاد يستطيع العد في الاتجاهين، أي العد تصـــاعدياً أو العــد تنازلياً، بناء على قيمة إشارة تحكم M، فيقوم بالعد تصاعدياً عنــدما تكــون $M = 0$ و تنازلياً عندما تكون $M = 1$.

نعلم أن العد تصاعدياً يتطلب ربط المراجيح بطريقة معينة، و العد تنازلياً يتطلب ربطها بطريقة أخرى مختلفة. فكيف يمكن ربط المراجيح بكلا الطريقتين في وقت واحــد ثــم اختيار إحداهما بناء على قيمة إشارة التحكم M؟

يتم ذلك باستخدام دوائر دامج من نوع 1 إلى 2 (2-to-1 MUX's)، مثلما فعلنا من قبــل بالنسبة لمسجلات الإزاحة. حيث يدخل كل من الخــرج المعكــوس و الخــرج غيــر المعكوس للمرجاح لطرفي الدخل للدامج الذي يحدد أيهما يمر كإشارة تزامن للمرجــاح التالي بناء على قيمة الإشارة M، فعندما تكون $M = 0$ يمر الخرج غير المعكــوس و بالتالي يكون العد تصاعدياً، أما عندما تكون $M = 1$ يمر الخرج المعكوس فيكون العــد تنازلياً. و الشكل التالي يوضح الدائرة المنطقية لعدّاد تصاعدي/تنازلي ذي خانات ثلاثة (3-bit Up/Down Counter)



و يمكن رسم مخطط الحالات (State Diagram) لهذا العدّاد بالصورة التالية



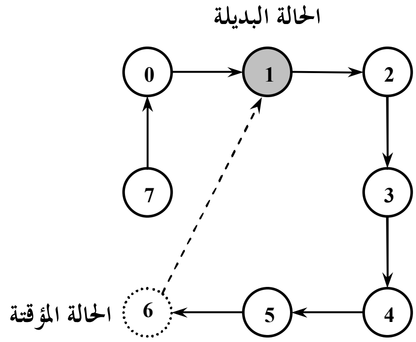
تدريب 6

صمم عدّاداً ذو أربعة خانات (4-bit Counter) يستجيب لإشارتي تحكم M و E. الإشارة M تحدد ترتيب العد للعدّاد فيقوم بالعد تصاعدياً عندما تكون مساوية 0، و تنازلياً عندما تكون مساوية 1، و الإشارة E عبارة عن إشارة سماح (Enable) تسمح للعدّاد بالعمل عندما تكون مساوية 1 و توقف العدّاد عن العد عندما تكون مساوية 0.

5.4 العد ضمن نطاق معين

جميع العدّادات التي قمنا بتصميمها حتى الآن تمر أثناء عملية العد بجميع حالاتها، فمثلاً العدّاد ذو الثلاثة خانات يعد تصاعدياً من 0 إلى 7 أو تنازلياً من 7 إلى 0 و يمر دائماً بكل حالة من حالاته الثمانية. المطلوب الآن هو جعل العدّاد يقوم بالعد ضمن نطاق معين لا يتضمن جميع حالاته، مثلاً جعل العدّاد ذو الثلاثة خانات يقوم بالعد تصاعدياً من 1 إلى 5 فقط. يتم ذلك بالتدخل في عمل العدّاد و تغيير التسلسل الطبيعي لحالاته باستبدال حالة معينة من حالاته بحالة أخرى بصورة غير متزامنة عن طريق أطراف الدخل المباشر (Direct Inputs). فإذا بدأ العدّاد العد من الحالة 1 فإنه سيسير بالترتيب المطلوب حتى يصل إلى الحالة 5، و هي آخر حالة في تسلسل العد المطلوب، و بعد

ذلك سينتقل إلى الحالة 6. يجب عندها إستبدال الحالة 6 بصورة فورية بالحالــة 1 عــن طريق أطراف الدخل المباشر للمراجيح. تسمى الحالة 6، و هي الحالة التي تلي مباشرة آخر حالة في نطاق العد المطلوب، بالحالة المؤقتة، في حين تسمى الحالة 1، و هي أول حالة في نطاق العد المطلوب، بالحالة البديلة. و يمكن توضيح ذلك بمخطــط الحــالات التالي

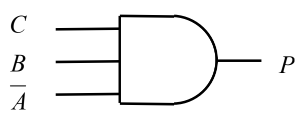


<u>لاحظ</u> أن الانتقال من الحالة المؤقتة إلى الحالة البديلة هو إنتقال غير متزامن لأنه يتم بصورة سريعة عبر أطراف الدخل المباشر، بحيث لا تظهر الحالة المؤقتة و إنما تظهر بدلاً عنها الحالة البديلة.

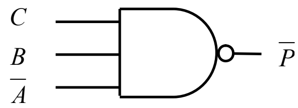
- <u>اكتشاف وصول العدّاد للحالة المؤقتة:</u>

يتم ذلك باستخدام بوابة AND أو بوابة NAND كما هو موضح أدناه

الحالة المؤقتة هي $\quad 6 = \begin{pmatrix} C & B & A \\ 1 & 1 & 0 \end{pmatrix}_2$



عند الوصول للحالة المؤقتة يصبح خرج بوابة AND P مساوياً 1.



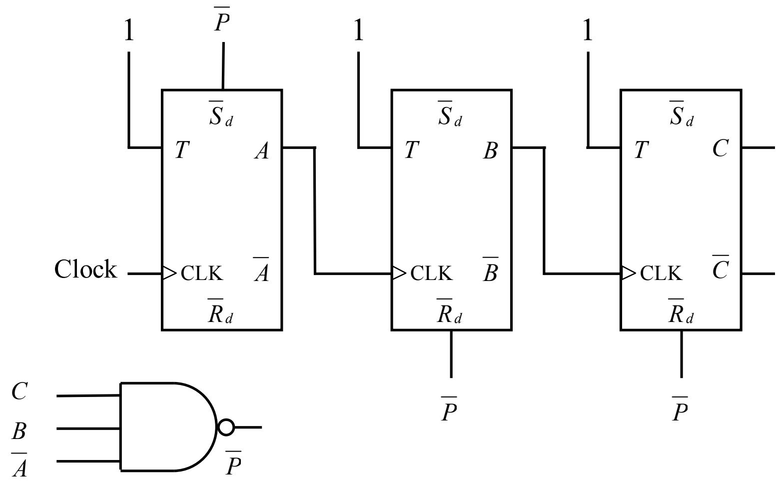
عند الوصول للحالة المؤقتة يصبح خرج بوابة NAND $\overline{P}$ مساوياً 0.

- <u>استبدال الحالة المؤقتة بالحالة البديلة:</u>

يتم ذلك، كما ذكرنا، عن طريق أطراف الدخل المباشر (Direct Inputs) للمراجيح، حيث تستخدم الإشارة P الخارجة من بوابة AND، أو الإشارة $\overline{P}$ الخارجة من بوابة NAND، في إجراء عمليات SET أو RESET للمراجيح بصورة مباشرة بحيث تُستبدل الحالة المؤقتة بالحالة البديلة، كما هو موضح أدناه

$$\begin{array}{ccccccc} & & & C & B & A & \\ 6 & = & (& 1 & 1 & 0 &)_2 \\ & & & \downarrow & \downarrow & \downarrow & \\ 1 & = & (& 0 & 0 & 1 &)_2 \end{array}$$

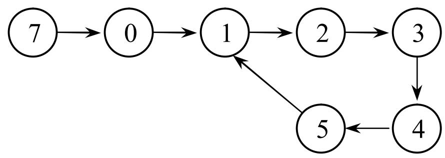
أي أننا نحتاج لإجراء عملية SET للمرجاح A و عملية RESET لكل من المرجاحين B و C. نقوم بذلك عن طريق إدخال الإشارة $\overline{P}$ إلى الطرف $\overline{S}_d$ في المرجاح A، و إلى الطرف $\overline{R}_d$ في المرجاحين B و C، كما هو موضح بالشكل التالي:



<u>لاحظ</u> أننا قد تركنا الطرف $\overline{R}_d$ في المرجاح A و الطرف $\overline{S}_d$ في المرجاحين B و C دون توصيل، و هذا يعادل إدخال القيمة 1 إلى تلك الأطراف (راجع طريقة استخدام أطراف الدخل المباشر في المراجيح).

<u>لاحظ</u> أنه طالما كانت الإشارة $\overline{P}$ مساوية 1 لا يكون لها أي تأثير على المراجيح، و يقوم العدّاد بالعد تصاعدياً بالصورة المعتادة، و لكن فور وصول العدّاد للحالة المؤقتة 6 تتغير قيمة الإشارة $\overline{P}$ من 1 إلى 0، فتقوم بإجراء عملية SET للمرجاح A و عملية RESET للمرجاحين B و C، مما يؤدي لاستبدال الحالة المؤقتة 6 بالحالة البديلة 1، عند ذلك تعود الإشارة $\overline{P}$ إلى 1 مرة أخرى و لا يعود لها أي تأثير على المراجيح و يعود العدّاد للعد تصاعدياً.

<u>لاحظ</u> إنه إذا بدأ العدّاد العد من إحدى الحالات الواقعة خارج نطاق العد المطلوب فإنه يقوم بالعد تصاعدياً حتى يدخل ضمن النطاق المطلوب و بعد ذلك لا يخرج منه. مثلاً إذا بدأ العدّاد العد من الحالة 7 فإنه يعد كالتالي

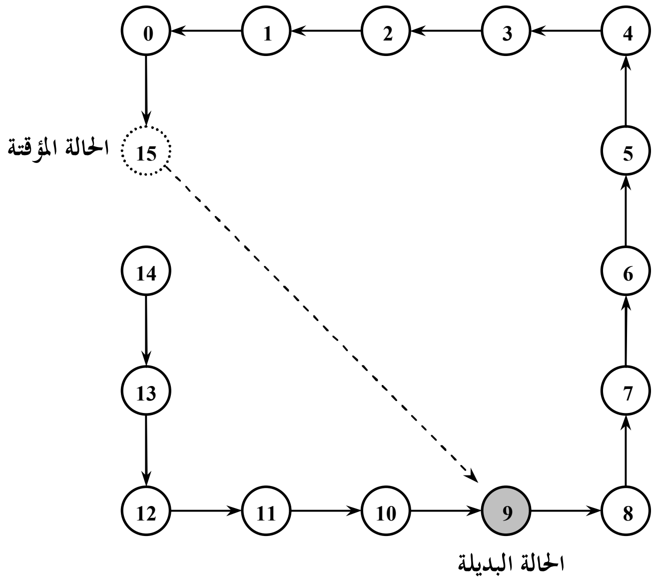


مثال:

صمم عدّاداً يقوم بالعد تنازلياً من 9 إلى 0.

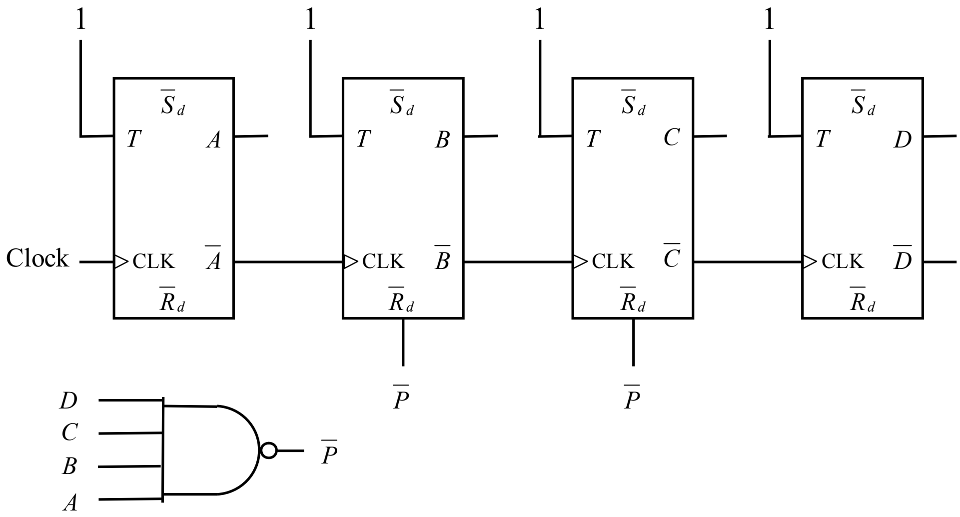
الحل:

سنقوم أولاً ببناء عدّاد ذي خانات أربعة يعد تنازلياً من 15 إلى 0، ثم نقوم بإجراء تعديل على دائرته بحيث يكون العد من 9 إلى 0. لإجراء التعديل اللازم على الدائرة يجب أن نقوم برسم مخطط الحالات للعدّاد المطلوب تصميمه لمعرفة الحالة المؤقتة و الحالة البديلة.



$$\begin{array}{ccccccccc} & & & D & C & B & A & & \\ 15 & = & (& 1 & 1 & 1 & 1 &)_2 \\ & & & \downarrow & \downarrow & \downarrow & \downarrow & & \\ 9 & = & (& 1 & 0 & 0 & 1 &)_2 \end{array}$$

لاستبدال الحالة المؤقتة بالحالة البديلة نحتاج لإجراء عملية RESET للمرجاحين B و C.

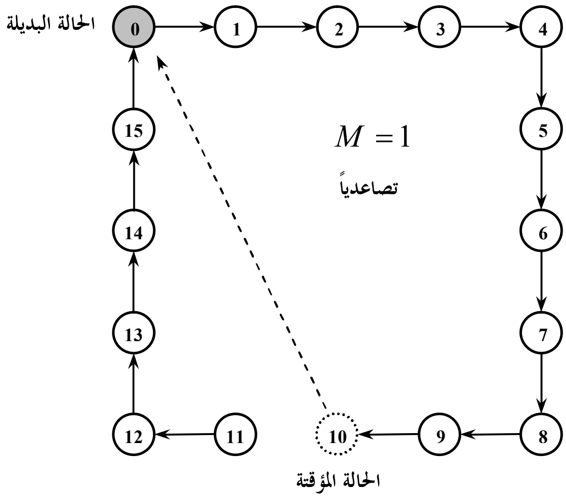


مثال:

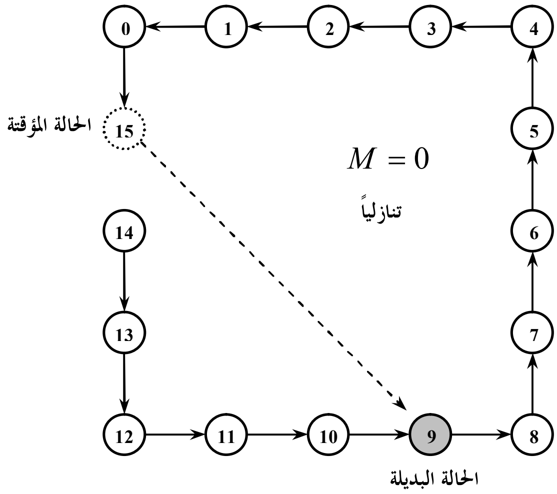
صمم عدّاداً يستجيب لإشارة تحكم M فيقوم بالعد تصاعدياً من 0 إلى 9 عندما تكون $M=1$ و تنازلياً من 9 إلى 0 عندما تكون $M=0$.

الحل:

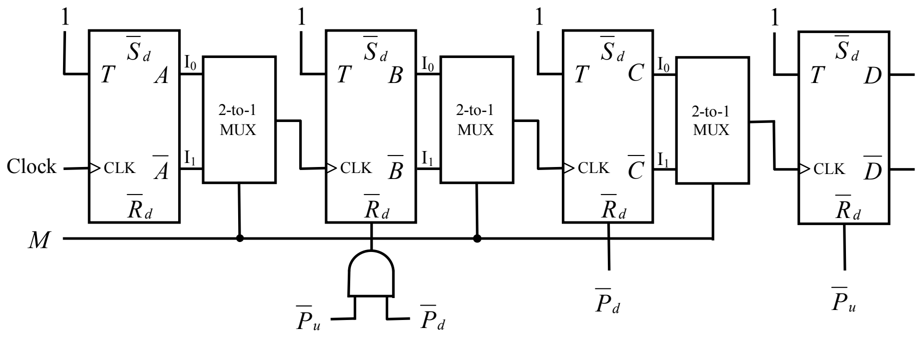
نقوم أولاً ببناء عدّاد تصاعدي/تنازلي ذو أربعة خانات، ثم نقوم بإجراء تعديل على دائرته بحيث يكون العد التصاعدي من 0 إلى 9 و العد التنازلي من 9 إلى 0. لإجراء التعديل اللازم على الدائرة يجب أن نقوم برسم مخطط الحالات للعدّاد المطلوب تصميمه لمعرفة الحالة المؤقتة و الحالة البديلة عند العد تصاعدياً و عند العد تنازلياً.



$$\begin{array}{ccccccccc} & & & D & C & B & A & & \\ 10 & = & (& 1 & 0 & 1 & 0 &)_2 \\ & & & \downarrow & \downarrow & \downarrow & \downarrow & & \\ 0 & = & (& 0 & 0 & 0 & 0 &)_2 \end{array}$$



$$
\begin{array}{cccccccc}
 & & & D & C & B & A & \\
15 & = & (& 1 & 1 & 1 & 1 &)_2 \\
 & & & \downarrow & \downarrow & \downarrow & \downarrow & \\
9 & = & (& 1 & 0 & 0 & 1 &)_2
\end{array}
$$

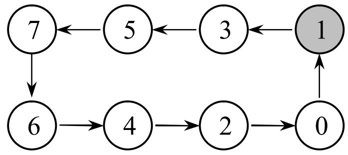


<u>لاحظ</u> أننا استخدمنا هنا بوابة NAND لاكتشاف الحالة المؤقتة في حالة العد التصـــاعدي، و خرجها هو $\overline{P}_u$، و بوابة NAND أخرى لاكتشاف الحالة المؤقتـــة فـــي حالـــة العـــد التنازلي، و خرجها هو $\overline{P}_d$. كما قمنا بإدخال إشارة التحكم M إلـــى بـــوابتي NAND (معكوسة لبوابة NAND الخاصة بالعد التنازلي و بدون عكس لبوابة NAND الخاصـــة بالعد التصاعدي) لضمان أن تنشط كل بوابة فقط في حالة ترتيب العد الخاص بها.

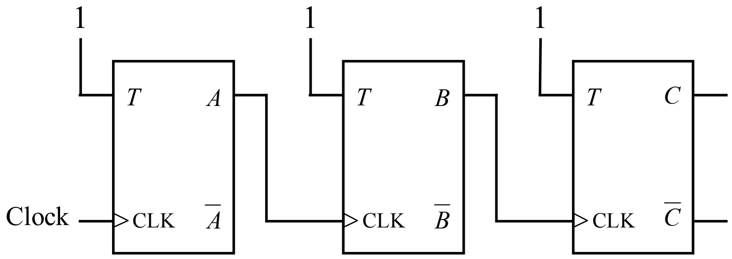
<u>لاحظ</u> أيضاً أننا قد استخدمنا بوابة AND في إدخال كلا الإشارتين $\overline{P}_u$ و $\overline{P}_d$ إلى الطرف $\overline{R}_d$ في المرجاح B.

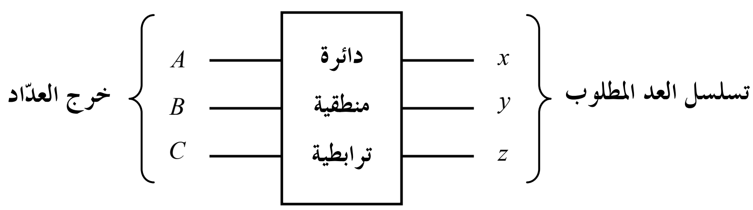
6.4 العد بأي ترتيب

المطلوب الآن تصميم عدّاد يقوم بالعد بتسلسل معين، و لكن بترتيب عد ليس تصاعدياً و لا تنازلياً. مثلاً مطلوب تصميم عدّاد ذي خانات ثلاثة يعد بالترتيب التالي



أبسط حل هنا هو أن نقوم بتصميم عدّاد تصاعدي ذي خانات ثلاثة (يقوم بالعد تصاعدياً من 0 إلى 7)، ثم نقوم بإدخال خرج هذا العدّاد إلى دائرة منطقية ترابطية تقوم بتحويل تسلسل العد إلى التسلسل المطلوب

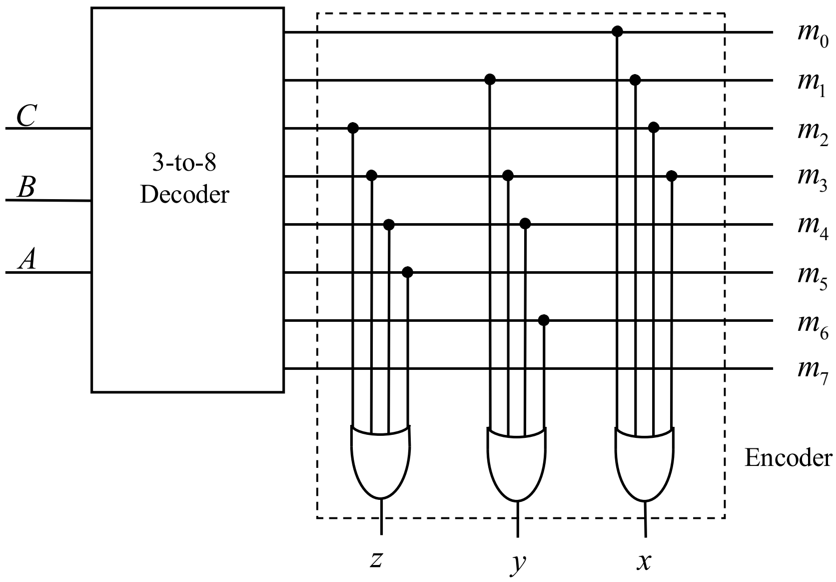




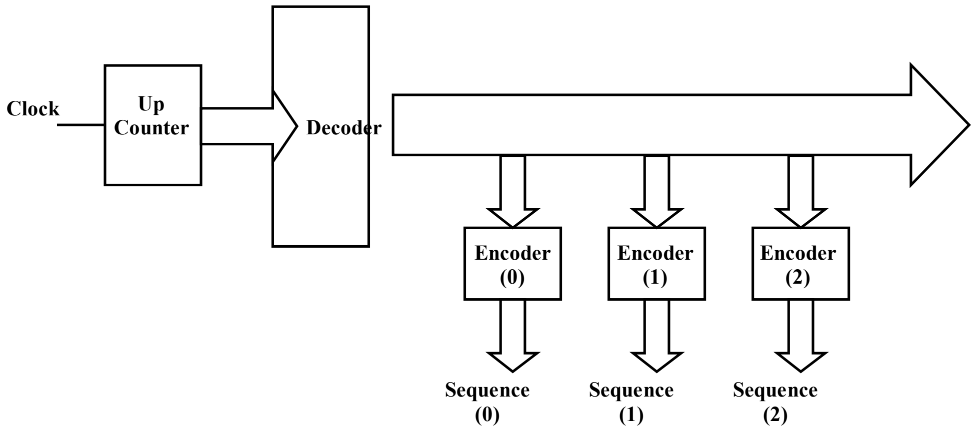
#	C	B	A	z	y	x	Dec.
0	0	0	0	0	0	1	1
1	0	0	1	0	1	1	3
2	0	1	0	1	0	1	5
3	0	1	1	1	1	1	7
4	1	0	0	1	1	0	6
5	1	0	1	1	0	0	4
6	1	1	0	0	1	0	2
7	1	1	1	0	0	0	0

عادة ما يتم تصميم الدائرة المنطقية الترابطية هنا باستخدام فاك شفرة و مشفر (Decoder & Encoder)

$$z = \sum m\ (2,3,4,5)$$
$$y = \sum m\ (1,3,4,6)$$
$$x = \sum m\ (0,1,2,3)$$



كما يمكن الحصول على مجموعة من تسلسلات العد المختلفة في وقت واحد، و ذلك باستخدام عدّاد تصاعدي واحد و فاك شفرة (Decoder) واحد و مجموعة من المشفرات (Encoders)، كل مشفر منها لتوليد تسلسل عد معين، كما هو موضح بالشكل التالي:



الخلاصة

قمنا في هذه الوحدة بتوضيح المقصود بالدوائر المنطقية التتابعية، و الفرق مـــا بينهـــا و بين الدوائر المنطقية الترابطية. كما تعرفنا على بعض أنواع الدوائر المنطقية التتابعية شائعة الاستخدام في الأنظمة الرقمية مثل المراجيح (Flip Flops) والتي هي عبارة عن دائرة منطقية تتابعية لها القدرة على تخزين خانة ثنائية واحدة (1-bit) فقط من البيانات ، ويتم بناء المراجيح باستخدام العواكس المنطقية أو باستخدام بوابات NOR أو باســتخدام (Clock Signal) ، وتحدث القسم أيضاً عن مرجاح القائد التـــابع الـــذي يتكـــون مـــن مرجاحي SR متزامنين متصلين ببعضهما البعض بحيث يغذي خرج أولهما دخل الثـــاني ، وفي هذا القسم أيضاً تعرضنا لمخططات التزامن وهو الوسيلة التي تتابع بهـــا التغيـــر الذي يحدث في الدائرة مع الزمن فهذا المخطط يوضح لنا التغيـــر الـــذي يحـــدث فـــي متغيرات الدخل والخرج للدائرة المنطقية مع الزمن.

انتقلنا بعد ذلك الى أطراف الدخل المباشر حيث يكون مطلوبـــاً تغيـــر حالـــة المرجـــاح بصورة استثنائية بغض النظر عن حالة إشارة التزامن. في القسم الثالـــث مـــن الوحـــدة تابعنا المسجلات والعمليات التي يمكن إجراؤها على المسجلات وهي :

– الكتابة أي تخزين معلومة في المسجل – القراءة أي استرجاع معلومة في المســـجل – نقل البيانات ما بين المسجلات وعرفنا كيف يتم بناء المسجلات باستخدام مراجيح D (D Flip Flops) ، وفي محور أخر من هذا القسم عرفنا مسجلات الإزاحة والذي هو عبـــارة عن مسجل يستطيع عمل إزاحة للبيانات الموجودة بداخلة بمقدار خانة واحـــدة أو أكثـــر يميناً أو يساراً، في القسم الأخير من الوحدة تعرفنا على العداد وترتيب العد الـــذي قـــد يكون ترتيباً تصاعدياً أو تنازلياً ، وفي بناء العدادات استخدمنا مراجيح JK ، ومـــراجيح T في وضع عكس الحالة (Toggle Mode)

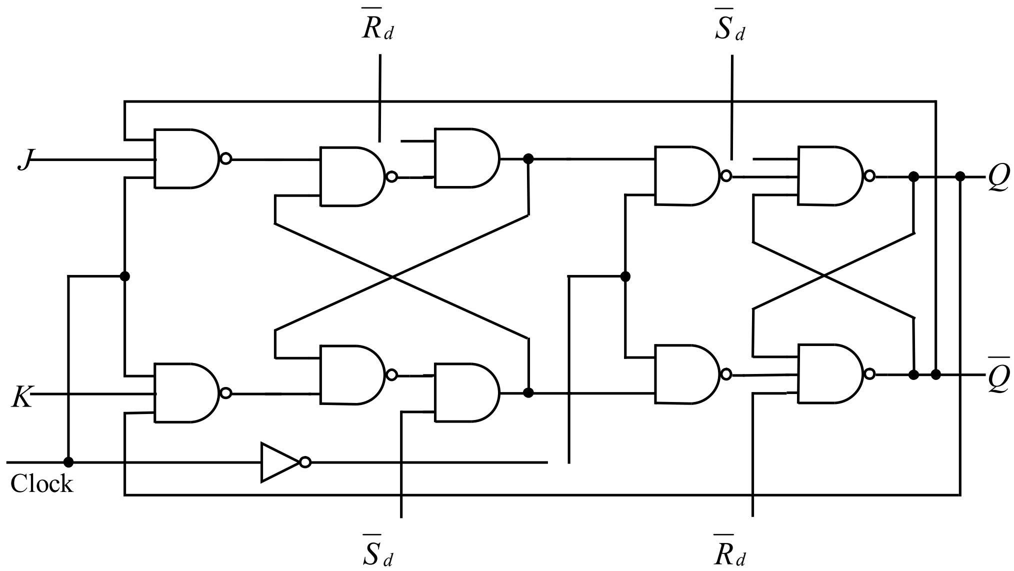
و أخيراً قمنا بعرض مبسط لبعض أنواع العدّادات و وضحنا كيفية بنائها و اســـتخدامها لتوليد تسلسل (Sequence) معين.

لمحة مسبقة عن الوحدة التالية

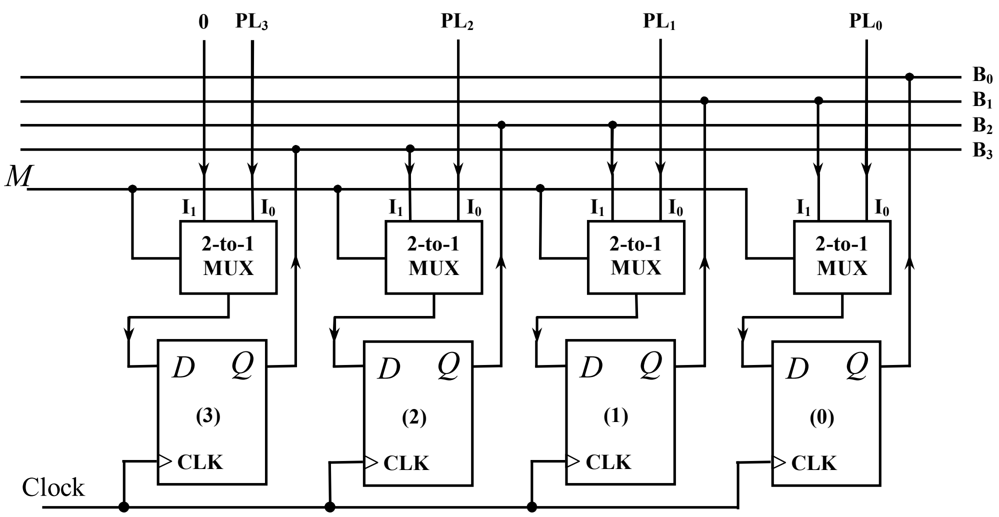
نقوم في الوحدة التالية، و هي الوحدة السادسة و الأخيرة في هذا المقرر، بدراسة أهم تقنيات التخزين (Storage) المستخدمة في الأنظمة الرقمية. فنبدأ بتوضيح التنظيم المنطقي للذاكرة (Memory)، ثم ننتقل لدراسة شرائح الذاكرة (Memory Chips) و أطراف التوصيل لها و طرق ربطها مع بعضها البعض. بعد ذلك ندخل إلى شريحة الذاكرة نفسها و نقوم بتوضيح البناء الداخلي لها، و نبدأ في ذلك بذاكرة الـ RAM حيث نوضح بناءها الداخلي و خصائصها و أنواعها المختلفة و استخدامات كل نوع منها، ثم ننتقل إلى ذاكرة الـ ROM و نقوم أيضاً بتوضيح البناء الداخلي لها و خصائصها و أنواعها. و في نهاية الوحدة نقوم بعرض أهم تقنيات التخزين الثانوي (Secondary Storage) مثل الأشرطة الممغنطة و الأقراص الممغنطة و الأقراص الضوئية.

إجابات التدريبات

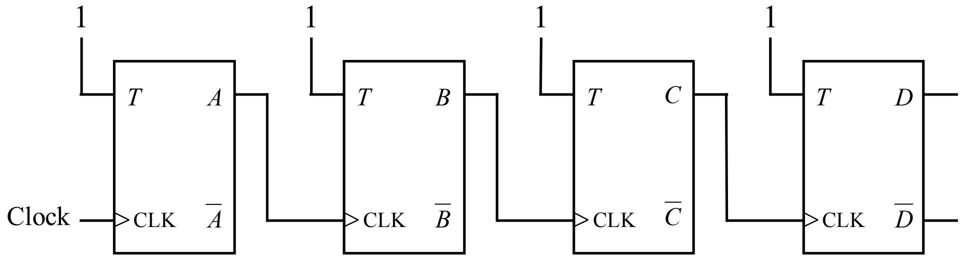
تدريب 1:



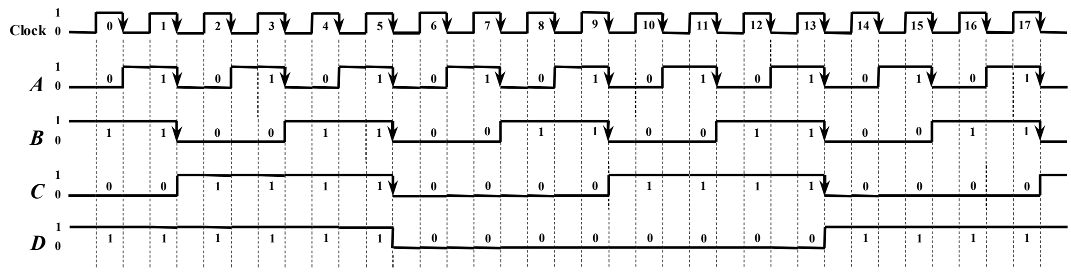
تدريب 2:



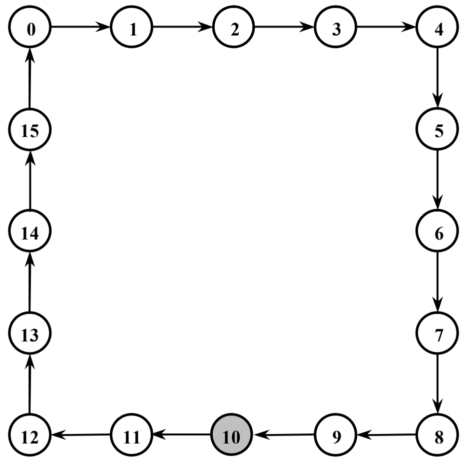
تدريب 3:



$$\begin{array}{ccccccc} & & & D & C & B & A \\ 10 & = & (& 1 & 0 & 1 & 0 &)_2 \end{array}$$

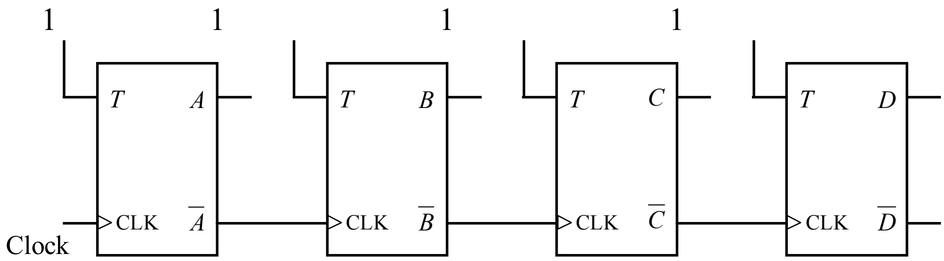


D	**C**	**B**	**A**	State
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
⋮	⋮	⋮	⋮	⋮

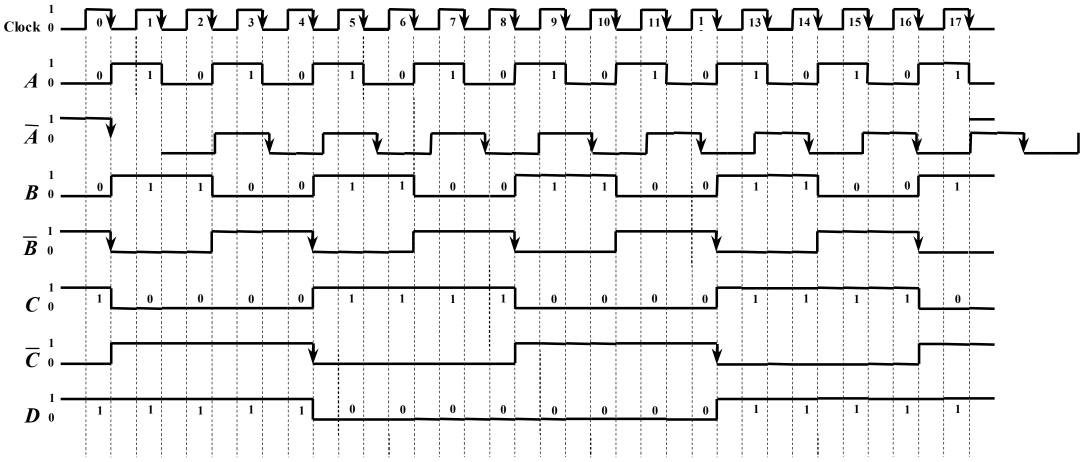


الحالة الإبتدائية

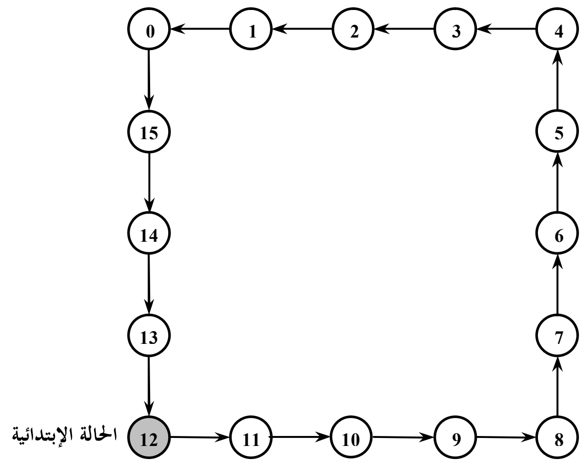
تدريب 4:



$$\begin{array}{ccccccc} & & & D & C & B & A \\ 12 & = & (& 1 & 1 & 0 & 0 &)_2 \end{array}$$

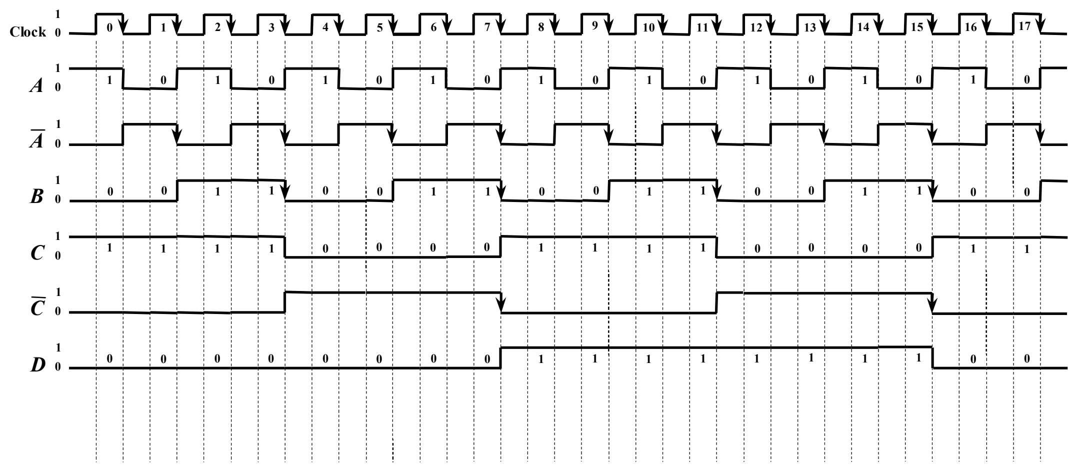


D	C	B	A	State
1	1	0	0	12
1	0	1	1	11
1	0	1	0	10
1	0	0	1	9
1	0	0	0	8
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0
1	1	1	1	15
1	1	1	0	14
1	1	0	1	13
1	1	0	0	12
1	0	1	1	11
⋮	⋮	⋮	⋮	⋮

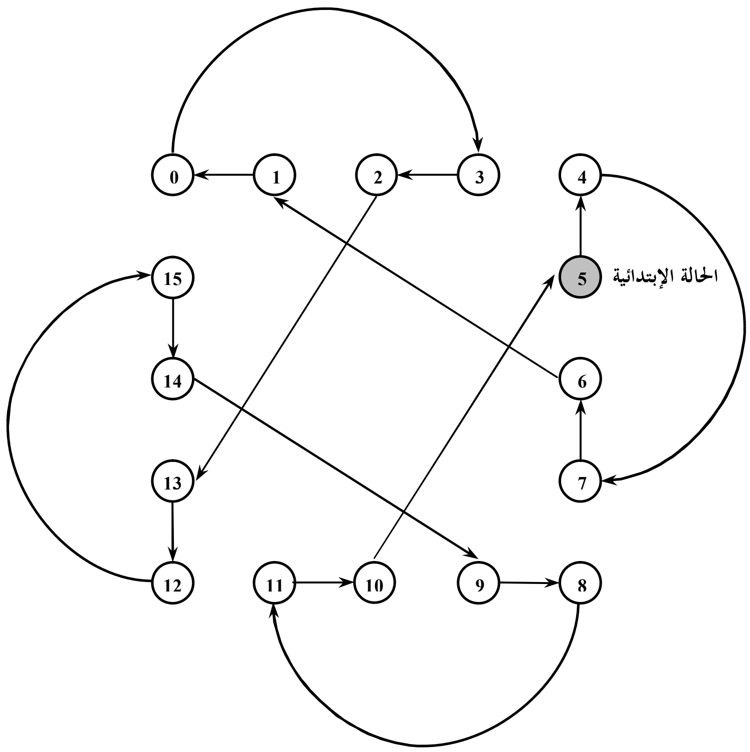


تدريب 5:

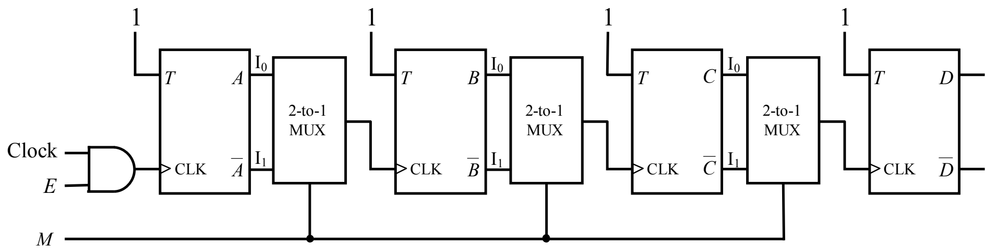
$$\begin{array}{ccccccc} & & & D & C & B & A \\ 5 & = & (& 0 & 1 & 0 & 1 &)_2 \end{array}$$



D	**C**	**B**	**A**	State
0	1	0	1	5
0	1	0	0	4
0	1	1	1	7
0	1	1	0	6
0	0	0	1	1
0	0	0	0	0
0	0	1	1	3
0	0	1	0	2
1	1	0	1	13
1	1	0	0	12
1	1	1	1	15
1	1	1	0	14
1	0	0	1	9
1	0	0	0	8
1	0	1	1	11
1	0	1	0	10
0	1	0	1	5
0	1	0	0	4
⋮	⋮	⋮	⋮	⋮


0
1
2
3
4
5
الحالة الإبتدائية
6
7
8
9
10
11
12
13
14
15

تدريب 6:


1
1
1
1
T
A
$\bar{A}$
CLK
I_0
I_1
2-to-1
MUX
T
B
$\bar{B}$
CLK
I_0
I_1
2-to-1
MUX
T
C
$\bar{C}$
CLK
I_0
I_1
2-to-1
MUX
T
D
$\bar{D}$
CLK
Clock
E
M

مسرد المصطلحات

- **المرجاح (Flip Flop)**

عبارة عن دائرة منطقية تتابعية لها القدرة على تخزين خانة ثنائية واحدة (1-bit) فقط من البيانات.

- **مرجاح D (D Flip Flop)**

و D هنا اختصار لكلمة Data، أي أن الاسم الكامل للمرجاح هو Data Flip Flop. و مرجاح D عبارة عن مرجاح SR متزامن تم ربط طرفي الدخل S و R له في طرف دخل واحد هو D باستخدام عاكس منطقي .

- Propagation Delay

هو عبارة عن الفترة الزمنية التي تمضي ما بين تسليط الدخل على البوابة و ظهور الاستجابة في خرجها.

- Hazard (التخمين) عبارة عن قيم غير معروفة تحدث في الفترة الزمنية التي تمضي مابين تسليط الدخل على البوابة وظهور الاستجابة في خرجها . ويحدث هذا في الدوائر المنطقية التتابعيه .

- **مرجاح القائد–التابع (Master-Slave Flip Flop)**

يتكون مرجاح القائد–التابع من مرجاحي SR متزامنين متصلين ببعضهما البعض بحيث يغذي خرج أولهما دخل الثاني.

- **مخطط التزامن (Timing Diagram)**

مخطط التزامن يوضح التغير الذي يحدث في متغيرات الدخل و الخرج للدائرة المنطقية مع الزمن.

- **مرجاح JK (JK Flip Flop)**

مرجاح JK هو عبارة عن مرجاح من نوع القائد–التابع مزود بتغذية مرتدة (Feedback) إضافية .

- **مرجاح T (T Flip Flop)**

و T هي إختصار لكلمة Toggle، بمعنى عكس الحالة ، و مرجاح T هو عبارة عن مرجاح JK تم ربط طرفي الدخل له في طرف واحد هو الطرف T .

- **المسجل (Register)**

المسجل (Register) هو عبارة عن موقع تخزيني له القدرة على اختزان معلومة مكونة من عدة خانات.

- **مسجل الإزاحة (Shift Register)**

مسجل الإزاحة (Shift Register) هو عبارة عن مسجل يستطيع، إضافة إلى العمليات السابقة، عمل إزاحة للبيانات الموجودة بداخله بمقدار خانة واحدة أو أكثر يميناً أو يساراً ، و هناك عدة أنواع من الإزاحة .

- **مسجل الإزاحة العام (Universal Shift Register)**

مسجل الإزاحة العام هو عبارة عن مسجل إزاحة يمكن أن يقبل دخلاً متوازياً (Parallel Input) أو دخلاً متتالياً (Serial Input)، و يقوم بالإزاحة يميناً أو يساراً أو يتوقف عن الإزاحة بناء على قيم إشارتي تحكم هما M_0 و M_1 .

- **العدّادات (Counters)**

العدّاد (Counter) هو عبارة عن دائرة منطقية تتابعية لها القدرة على العد ثنائياً بترتيب معين. و ترتيب العد قد يكون ترتيباً تصاعدياً (Up Counting)، أو قد يكون ترتيباً تنازلياً (Down Counting)، أو قد يكون بأي ترتيب آخر.


الوحدة السادسة
الذاكرة

محتويات الوحدة

مقدمة

تمهيد

مرحباً بك عزيزي الدارس في الوحدة السادسة و الأخيرة من مقرر "أساسيات التصـــميم المنطقي". نقوم في هذه الوحدة بدراسة أهم تقنيات التخزين (Storage) المستخدمة فـــي الأنظمة الرقمية، حيث تعتبر القدرة علـــى تخـــزين كميـــات كبيـــرة مـــن البيانـــات أو المعلومات، لفترة زمنية قد تطول أو تقصر، من أهم ما يميز الأنظمة الرقمية (Digital System) على نظيرتها التماثلية (Analog Systems). نبدأ الوحدة بتوضيح التنظـــيم المنطقي للذاكرة، أي الذاكرة كما يراها المبرمج (Programmer)، ثم ننتقـــل لدراســـة شرائح الذاكرة و أطراف التوصيل لها و طرق ربطها مع بعضها البعض، و ذلك قبـــل الدخول إلى شريحة الذاكرة نفسها و توضيح البناء الداخلي لها. ونبدأ في ذلـــك بـــذاكرة الـــ RAM حيث نوضح بناءها الداخلي و خصائصها و أنواعها المختلفة و استخدامات كل نوع منها. ثم ننتقل إلى ذاكرة الـــ ROM و نقوم أيضاً بتوضيح البناء الداخلي لها، كما نقوم بتوضيح خصائصها و أنواعها. ثم نختتم الوحدة بعرض أهم تقنيات التخـــزين الثانوي (Secondary Storage) مثل الأشرطة الممغنطـــة و الأقـــراص الممغنطـــة و الأقراص الضوئية.

أهداف الوحدة

عزيزي الدارس، بعد دراسة هذه الوحدة ينبغي أن تكون قادراً على أن:

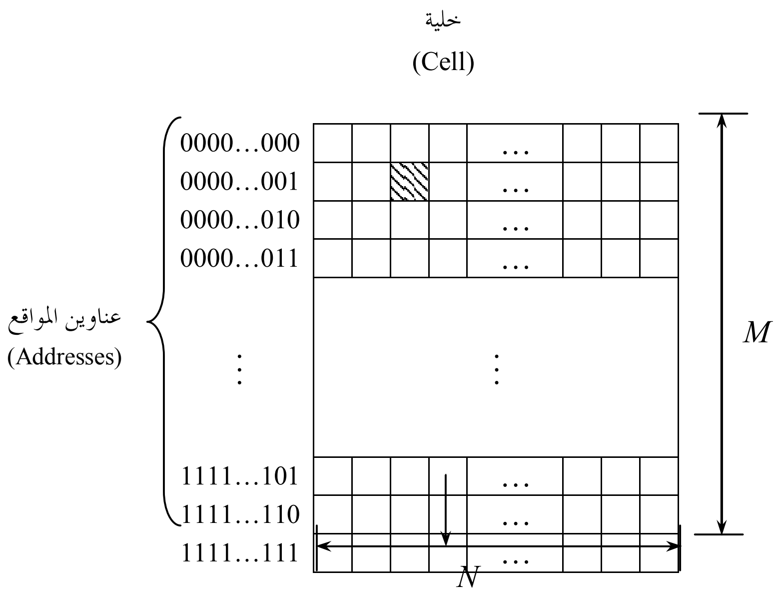
- **توضح** التنظيم المنطقي للذاكرة.
- **تستخدم** شرائح الذاكرة في الأنظمة الرقمية و تربطها مع بعضها البعض.
- **تصف** البناء الداخلي لذاكرة الــ RAM.
- **تحدد الفرق بين** الــ SRAM و الــ DRAM و خصائص و استخدامات كل منهما.
- **تشرح** البناء الداخلي لذاكرة الــ ROM.
- **تقارن بين** الأنواع المختلفة للــ ROM و خصائص و استخدامات كل نوع.
- **تعرف** أهم تقنيات التخزين الثانوي و إمكانيات و خصائص كل تقنية منها.

1. التنظيم المنطقي للذاكرة

(Logical Memory Organization)

المقصود بالتنظيم المنطقي للذاكرة هنا هو صورة الذاكرة كما يراهـا مبـرمج النظـام الرقمي (Programmer)، أي مجموعة من المواقع التخزينية المتتالية المتسـاوية فـي الطول، كل موقع منها يتكون من عدد من الخلايا التخزينية (Cells)، كل خليـة منهـا تستطيع تخزين bit واحد فقط من البيانات، و لكل موقع مـن هـذه المواقـع عنـوان (Address) فريد يميزه عن سواه من المواقع. يمكن الوصول لأي موقع مـن مواقـع الذاكرة عن طريق عنوانه، و ذلك إما لإجراء عملية قراءة (Read) منه، أي إسـترجاع للبيانات المخزنة فيه، أو عملية كتابة (Write) عليه، أي تخزين لبيانات فيه. و عمليـة القراءة غير مدمرة (Nondestructive) لمحتويات الموقع، أي أن الموقع يظل محتفظاً بالبيانات المخزنة فيه كما هي بعد عملية القراءة. أمـا عمليـة الكتابـة فهـي مـدمرة (Destructive) للمحتويات السابقة للموقع، حيث تحل البيانات الجديدة التي تمت كتابتها على الموقع محل البيانات السابقة التي كانت مخزنة فيه.

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



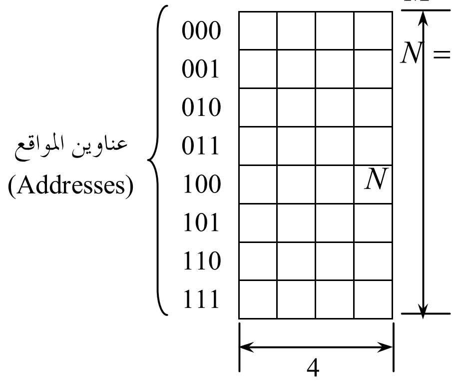
مثال:

وضح التنظيم المنطقي لذاكرة من نوع 4×8.

الحل:

عدد مواقع الذاكرة هو $M=8$

طول الموقع الواحد هو $N=4$



العلاقة بين عدد مواقع الذاكرة و عدد خانات العنوان

توجد علاقة بين عدد مواقع الذاكرة، أي طولها M، و عدد خانات العنوان A. فإذا كان عدد خانات العنوان هو A فإن عدد العناوين الممكنة، و بالتالي عدد مواقع الذاكرة هو 2^A. أي أن:

$$\boxed{M=2^A}$$

أو (بأخذ ln الطرفين)

$$\boxed{A=\frac{\ln(M)}{\ln(2)}}$$

مثال:

أوجد عدد خانات العنوان لذاكرة من نوع:

(أ) $1\text{K}\times 8$

(ب) $4\text{M}\times 16$

الحل:

أ) عدد مواقع الذاكرة هنا هو $M=1\text{K}=1024$، أي أن

$$A = \frac{\ln(1024)}{\ln(2)} = 10$$

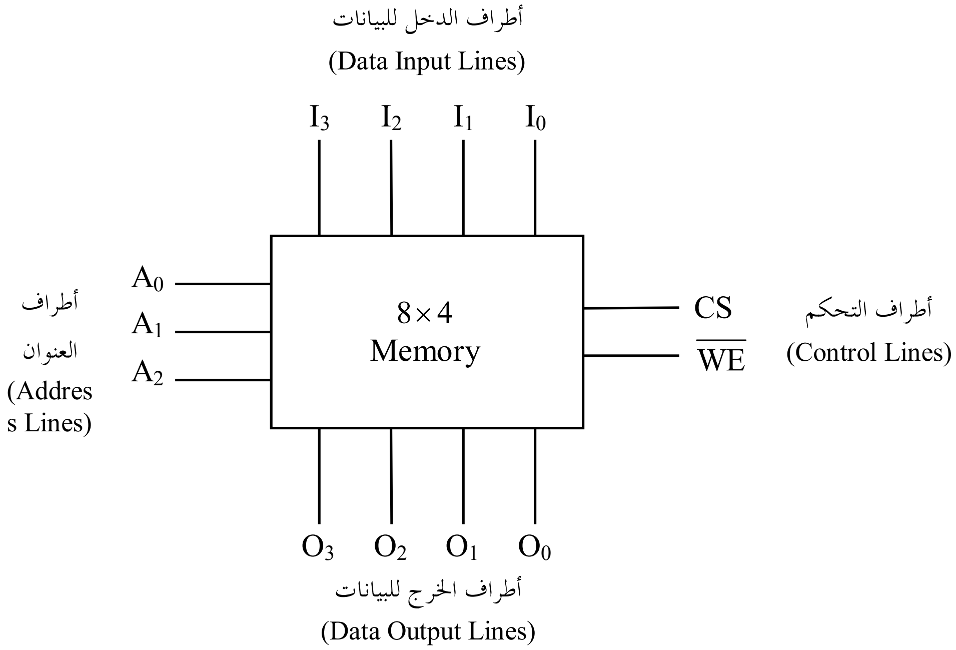
(ب) عدد مواقع الذاكرة هنا هو $M = 4\text{M} = 4(1024 \times 1024)$، أي أن

$$A = \frac{\ln(4(1024 \times 1024))}{\ln(2)} = 22$$

عادة ما يكون طول الموقع N في الحواسيب الشخصية (PC's) مساوياً 8-bits، و يطلق على الموقع في هذه الحالة تسمية Byte. أما في أنواع الحواسيب الأخرى فقد يكون طول الموقع مساوياً 16 أو 32 أو حتى 64-bits. و بشكل عام يطلق على موقع الذاكرة، بغض النظر عن طوله، تسمية Word. فالـ Word هي عبارة عن موقع الذاكرة الذي تتم قراءته منها أوكتابته فيها كاملاً و لا يمكن تجزئته.

2. شريحة الذاكرة (Memory Chip)

الشكل التالي يوضح المخطط المنطقي لشريحة ذاكرة من نوع 8×4



أطراف التوصيل لشريحة الذاكرة

1. أطراف الدخل للبيانات (Data Input Lines)، و عددها يساوي طـول الموقـع أو عرض الذاكرة N.
2. أطراف الخرج للبيانات (Data Output Lines)، و عددها يساوي طول الموقع أو عرض الذاكرة N.
3. أطراف العنوان (Address Lines)، و عددها يعتمد على عدد المواقـع أو طـول الذاكرة M، و ذلك حسب العلاقة $A=\dfrac{\ln(M)}{\ln(2)}$.
4. أطراف التحكم (Control Lines)، و عددها 2، و هي:

- $\overline{WE}$ أي طرف تمكين الكتابة (Write Enable)، و مهمته تحديد نوع العمليــة المطلوب إجراؤها على الذاكرة، و ذلك على النحو التالي:

  - $\overline{WE} = 0$ ← عملية كتابة (Write)
  - $\overline{WE} = 1$ ← عملية قراءة (Read)

  و يرمز لهذا الطرف أيضاً بالرمز $R/\overline{W}$ أو بالرمز $\overline{W}$

- CS أي طرف إختيار الشريحة (Chip Select)، و هو عبارة عن خط ســماح (Enable) يسمح لشريحة الذاكرة بالعمل كالمعتاد عند وضع القيمــة 1 فيــه، و يبطل عملها و يعزل أطراف الدخل و الخرج للبيانات عــن العــالم الخــارجي بمعاوقة عالية (High Impedance) عند وضع القيمة 0 فيه.

  و يرمز لهذا الطرف أيضاً بالرمز ME (Memory Enable) أو CE (Chip Enable)

5. أطراف التغذية بالقدرة (Power Supply)، و عددها 2 همــا $+V_{cc}$و GND. و لا يتم توضيحهما عادة على المخطط المنطقي لشريحة الذاكرة.

<u>مثال:</u>

احسب عدد أطراف التوصيل لشريحة ذاكرة من نوع:

(أ) 32×4

(ب) $1K \times 8$

الحل:

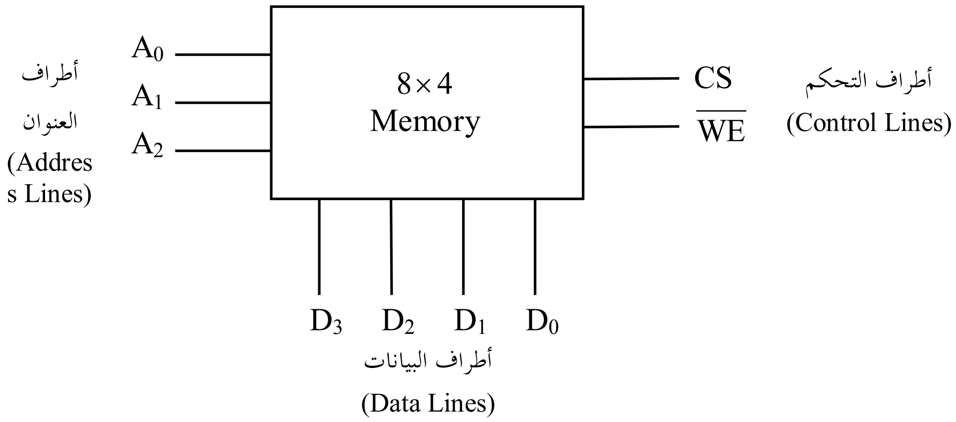
(أ)

عدد أطراف الدخل	4
عدد أطراف الخرج	4
عدد أطراف العنوان	5
عدد أطراف التحكم	2
عدد أطراف القدرة	2
المجموع	17

(ب)

عدد أطراف الدخل	8
عدد أطراف الخرج	8
عدد أطراف العنوان	10
عدد أطراف التحكم	2
عدد أطراف القدرة	2
المجموع	30

في بعض الأحيان تكون أطراف الدخل و الخرج للبيانات في شريحة الذاكرة مشتركة، و يطلق عليها في هذه الحالة أطراف البيانات (Data Lines)، كما هو موضح أدناه



تدريب 1

احسب عدد أطراف التوصيل لشريحة ذاكرة من نوع:

(أ) 8K×16 (ب) 1M×8

وذلك إذا كانت أطراف الدخل و الخرج للبيانات:

1. منفصلة 2. مشتركة

عملية الكتابة (Write)

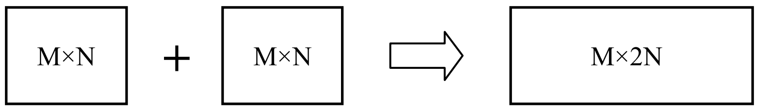
لتخزين بيانات معينة في موقع محدد من مواقع الذاكرة يجب أولاً وضع عنوان ذلك الموقع على أطراف العنوان لشريحة الذاكرة، و وضع البيانات المطلوب تخزينها على أطراف الدخل للبيانات، ثم اختيار عملية الكتابة بوضع القيمة 0 على طرف تمكين الكتابة $\overline{WE}$، و أخيراً تغيير القيمة الموضوعة على طرف إختيار الشريحة CS من 0 إلى 1، فتنتقل القيم الموضوعة على أطراف الدخل للبيانات إلى داخل الشريحة و يتم تخزينها في العنوان المحدد.

عملية القراءة (Read)

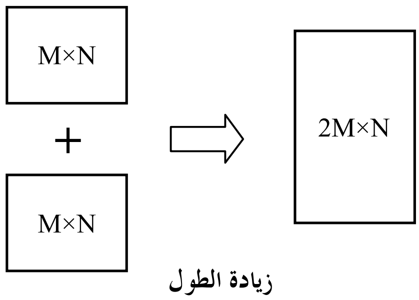
لقراءة البيانات المخزنة في موقع معين من مواقع الذاكرة يجب أولاً وضع عنوان ذلك الموقع على أطراف العنوان لشريحة الذاكرة، و إختيار عملية القراءة بوضع القيمة 1 على طرف تمكين الكتابة $\overline{WE}$، ثم تغيير القيمة الموضوعة على طرف إختيار الشريحة CS من 0 إلى 1، فتظهر القيم المخزنة في العنوان المحدد على أطراف الخرج للبيانات لشريحة الذاكرة.

3. ربط شرائح الذاكرة

الهدف من ربط شرائح الذاكرة هو إما زيادة عرض الذاكرة، أي زيادة طول الموقـــع أو الـــ Word، أو زيادة طول الذاكرة، أي زيادة عدد المواقع. كما هو موضح أدناه



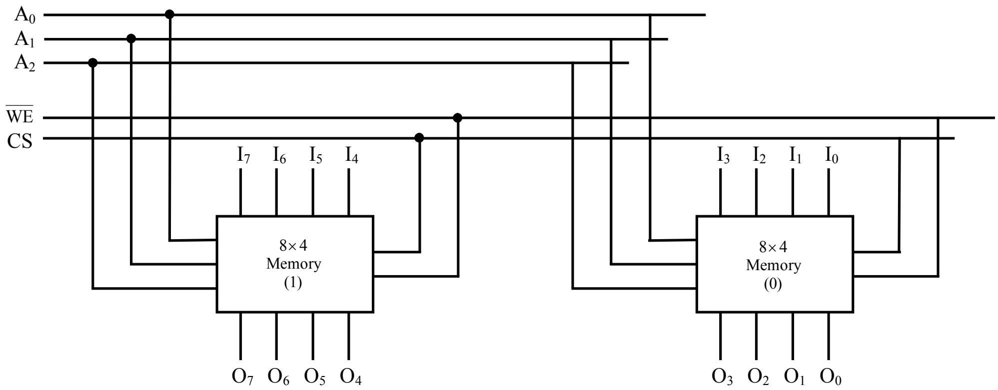
زيادة العرض



زيادة الطول

زيادة العرض

الشكل التالي يوضح طريقة ربط شريحتي ذاكرة من نوع 4×8 لبناء شريحة ذاكرة من نوع 8×8



خطوات الربط

1. أطراف الدخل و الخرج للبيانات يتم توزيعها بالتساوي ما بين الوحدات المربوطة.
2. أطراف العنوان مشتركة.
3. أطراف التحكم مشتركة.

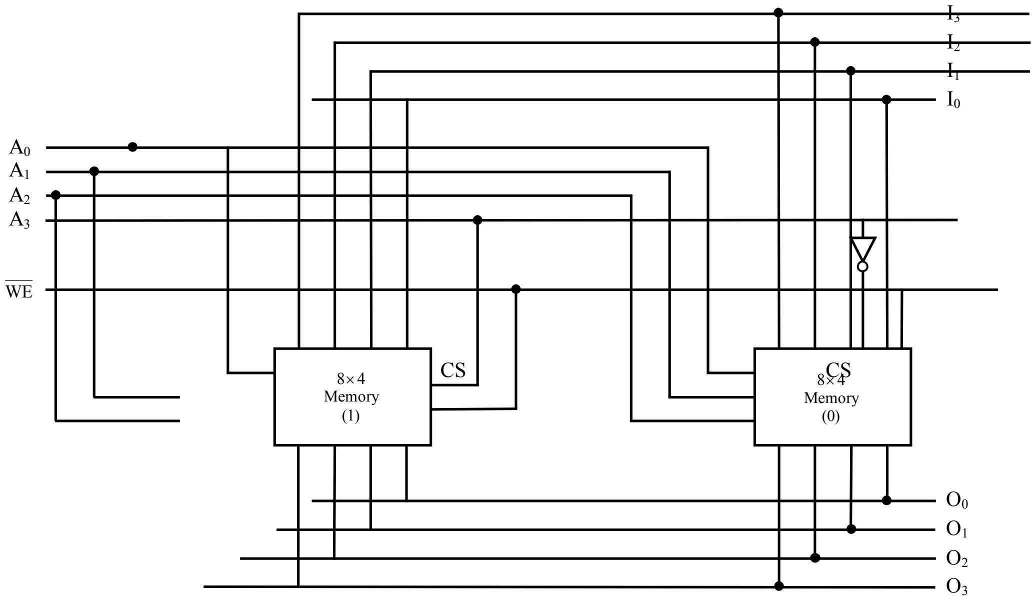
لاحظ هنا أن الخانات الأربعة الأولى من كل Word (أي نصف الـ Word) تكون مخزنة في عنوان معين في الشريحة رقم (0) و الخانات الأربعة الأخيرة من الـ Word (النصف الآخر) تكون مخزنة في نفس العنوان في الشريحة رقم (1). و في حالة ربط أربعة شرائح فإن الربع الأول من الـ Word يكون مخزناً في الشريحة رقم (0)، و ربعها الثاني في الشريحة رقم (1)، و ربعها الثالث في الشريحة رقم (2)، و ربعها الأخير في الشريحة رقم (3).

تدريب 2

وضح طريقة ربط 4 شرائح ذاكرة من نوع 4×8، بأطراف دخل و خرج مشتركة للبيانات، للحصول على شريحة ذاكرة من نوع 16×8.

زيادة الطول

الشكل التالي يوضح طريقة ربط شريحتي ذاكرة من نوع 4×8 لبناء شريحة ذاكرة من نوع 4×16



خطوات الربط:

1/ أطراف الدخل للبيانات مشتركة و أطراف الخرج للبيانات مشتركة.

2/ أطراف العنوان الدنيا مشتركة.

3/ طرف العنوان الأعلى يستخدم في اختيار الشريحة (Chip Select).

4/ طرف تمكين الكتابة $\overline{WE}$ مشترك.

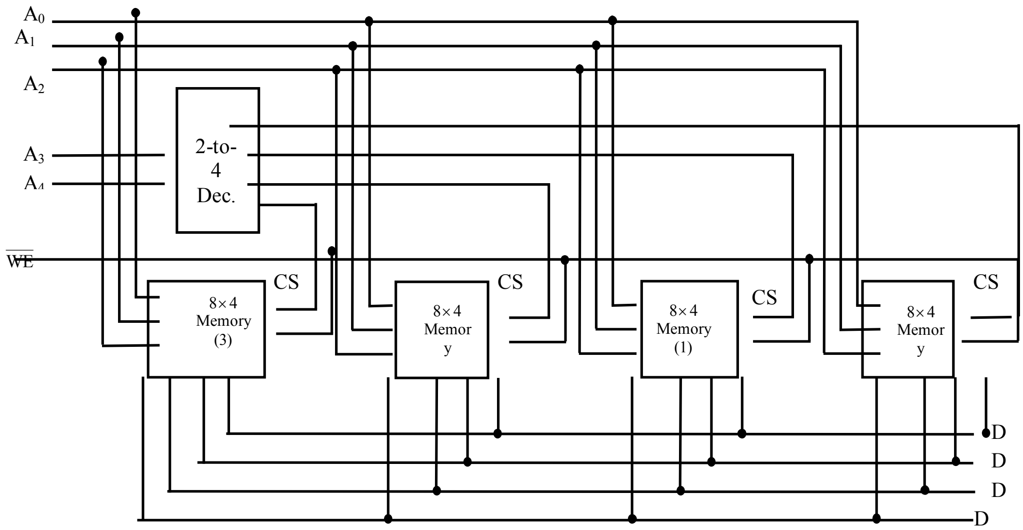
لاحظ هنا أن الكلمة الثمانية الأولى تكون مخزنة في الشريحة رقم (0)، و الكلمة الثمانية الأخيرة تكون مخزنة في الشريحة رقم (1). لاحظ أيضاً أن عنوان أي من الكلمات الثمانية الأولى يكون فيه طرف العنوان الأعلى A_3 مساوياً 0، و عنوان أي من الكلمات الثمانية الأخيرة يكون فيه طرف العنوان الأعلى A_3 مساوياً 1. فوضع القيمة 0 على

طرف العنوان الأعلى A_3 يؤدي لتنشيط شريحة الذاكرة رقم (0)، و بالتـــالي فـــإن أي عملية قراءة أو كتابة ستتم على العنوان من تلك الشريحة الذي تحدده أطراف العنـــوان الثلاثة الدنيا. و وضع القيمة 1 على طرف العنوان الأعلى A_3 يؤدي لتنشيط شـــريحة الذاكرة رقم (1)، و بالتالي فإن أي عملية قراءة أو كتابة ستتم على تلك الشريحة.

عند ربط أربعة شرائح ذاكرة فإننا نحتاج لاستخدام فاك شفرة من نوع 2 إلى 4 (2-to- 4 Decoder) للقيام بعملية إختيار الشريحة (Chip Select)، و دخل فاك الشفرة هنا هو عبارة عن طرفي العنوان الأعليين.

الشكل التالي يوضح طريقة ربط أربعة شرائح ذاكرة من نوع 4×8 للحصول على شريحة ذاكرة من نوع 4×32

(كنوع من التبسيط استخدمنا هنا شرائح ذاكرة بأطراف دخل و خرج مشتركة للبيانات)



لاحظ أنه في حالة ربط 8 شرائح ذاكرة فإننا نحتاج لاستخدام فاك شفرة من نوع 3 إلى 8 (3-to-8 Decoder) في عملية إختيار الشريحة (Chip Select)، و دخل فاك الشفرة هنا هو أطراف العنوان الثلاثة العليا.

تدريب 3

وضح طريقة ربط 4 شرائح ذاكرة من نـــوع 3×4، بـــأطراف دخـــل و خرج مشتركة للبيانات، للحصول على شريحة ذاكرة من نوع 3×16.

تدريب 4

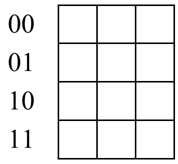
وضح طريقة ربط شرائح ذاكرة من نوع 4×8، بأطراف دخل و خـــرج مشتركة للبيانات، للحصول على شريحة ذاكرة من نوع 8×16.

4. ذاكرة الـــ RAM

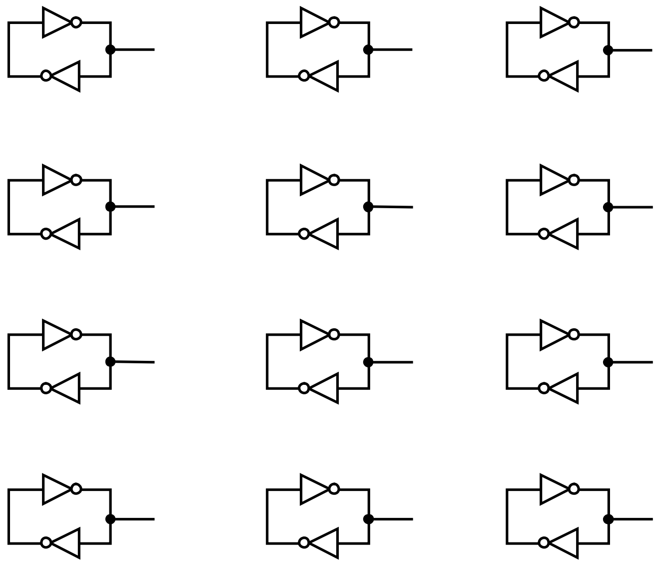
مصطلح RAM هو اختصار لعبارة Random Access Memory أي ذاكرة الوصول العشوائي. و سبب هذه التسمية هو أنه من الممكن في هذا النوع من الذاكرة الدخول إلى أي موقع من مواقعها بصورة عشوائية عن طريق عنوان ذلك الموقع دون الحاجة لإتباع ترتيب معين في الوصول، و ذلك بخلاف ما يسمى بذاكرة الوصول التتابعي أو Sequential Access Memory التي يجب الوصول على مواقعها من البداية حسب ترتيب التخزين، فلقراءة أي موقع من مواقع هذا النوع من الذاكرة يجب قراءة الذاكرة من بدايتها و حتى ذلك الموقع. و ذاكرة الـــ RAM تستخدم عادة كذاكرة أساسية (Main Memory) في الحاسوب و بقية الأنظمة الرقمية، حيث يتم تخزين البرامج و البيانات بها بصورة مؤقتة أثناء المعالجة (Processing)، و ذلك لأنها ذاكرة متطايرة (Volatile)، بمعنى أن احتفاظها بمحتوياتها مرتبط بتغذيتها بالقدرة، حيث تفقد تلك المحتويات بمجرد فصل مصدر التغذية بالقدرة عنها، فهي لا تصلح للتخزين الدائم. و يمكن إجراء كلا من عمليتي القراءة (Read) و الكتابة (Write) على ذاكرة الـــ RAM بنفس الدرجة من السهولة، و بالتالي يمكن مسح أو تغيير محتوياتها في أي وقت.

1.4 البناء الداخلي لذاكرة الــ RAM

سنقوم بتوضيح البناء الداخلي (Internal Structure) لذاكرة RAM مــن نــوع 4×3 الموضح المخطط المنطقي لها أدناه



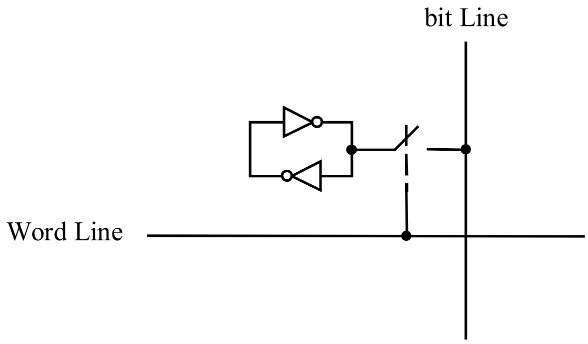
تتكون الذاكرة من 12 خلية تخزينية (Cell) مرتبة في شكل 4 صفوف و 3 أعمدة، كما هو موضح أدناه، و كل خلية هنا عبــارة عــن مرجــاح (Flip Flop). و المرجــاح المستخدم هنا هو أبسط أنواع المراجيح و الذي يطلق عليــه تســمية Static Latch، و يتكون من عاكسين منطقيين. و كل صف من الخلايا يمثل كلمة من الذاكرة.



الوصول للخلية

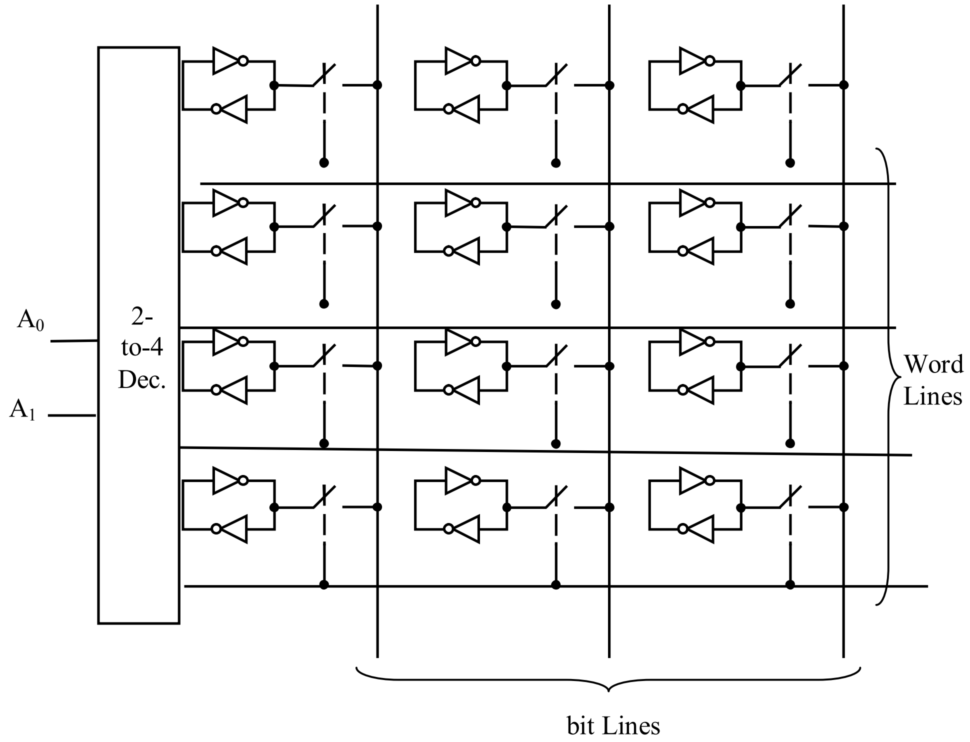
السؤال الآن هو كيف نصل لخلية معينة من خلايا الذاكرة لكتابة bit من البيانات فيها أو لقراءة الخانة المخزن بها؟

يتم ذلك باستخدام موصلين، أحدهما عمودي و يطلق عليه تسمية (خط الخانة)، و الآخر أفقي و يطلق عليه تسمية الـــ (خط الكلمة). تتصل الخلية بالـــ bit Line عن طريـــق مفتاح ترانزستوري (Transistor Switch)، و يتم التحكم في هذا المفتاح عن طريـــق الـــ Word Line. فعند وضع القيمة المنطقية 1 (الممثلة بجهد كهربائي عالي) علـــى الـــ Word Line يغلق المفتاح فيتم توصيل الخلية مـــع الـــــ bit Line ممـــا يســـمح بالوصول إليها لإجراء عملية كتابة فيها أو عملية قراءة منهـــا، و عنـــد وضـــع القيمـــة المنطقية 0 (الممثلة بجهد كهربائي منخفض) على الـــ Word Line يفتح المفتاح فيـــتم عزل الخلية عن الـــ bit Line. كما هو موضح أدناه



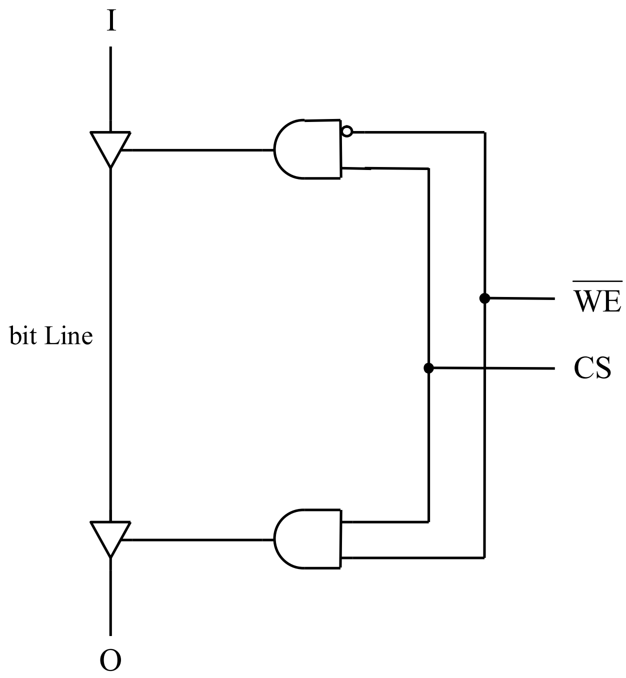
الوصول للـــ Word

للوصول إلى موقع معين من مواقع الذاكرة يستخدم جهاز فك شفرة تتصل أطراف الخرج له بالـــ Word Lines للذاكرة. فعندما يتم وضع عنوان الموقع المطلوب الوصول إليه على أطراف العنوان لفاك الشفرة يقوم فاك الشفرة بتنشيط الـــ Word Line المقابل لذلك الموقع، فتغلق مفاتيح جميع الخلايا المتصلة بذلك الـــ Word Line و يتم توصيل كل خلية مع الـــ bit Line المقابل لها، و بالتالي يتم الوصول للموقع و إجراء عمليات الكتابة عليه أو القراءة منه من خلال الـــ bit Lines. أما بقية مواقع الذاكرة فتكون خلاياها معزولة عن الـــ bit Lines لأن الـــ Word Lines المقابلة لها غير نشطة.



التحكم في الذاكرة

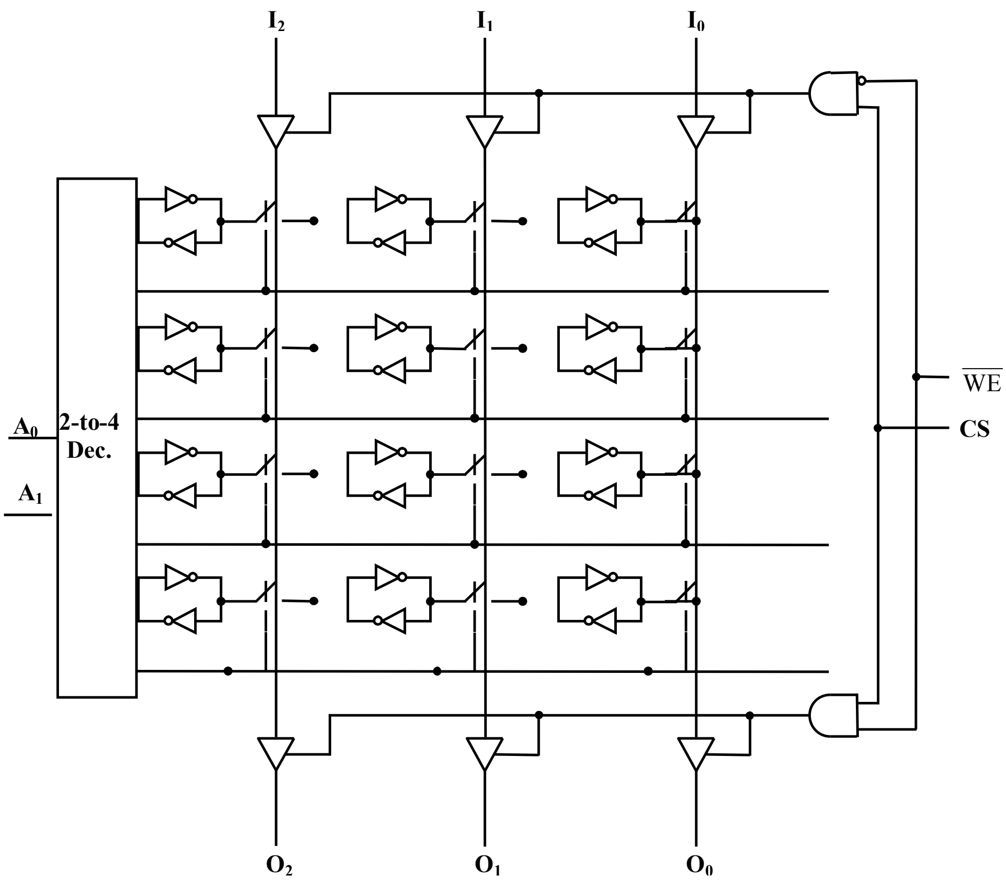
السؤال الآن هو كيف يتم التحكم في الذاكرة و إختيار العملية المطلوب إجراؤها عليها؟
يتم ذلك بوضع عازل ثلاثي الحالة (Tristate Buffer) في بداية و نهاية كل bit Line، و تغذية هذه العوازل ثلاثية الحالة بخرج دائرة تحكم تتكون من بوابتي AND، كما هو موضح بالشكل التالي



- عند وضع القيمة 0 في طرف إختيار الشريحة CS يكون خـرج كـلا بـوابتي AND مساوياً 0 و بالتالي يدخل كلا العازلين في حالة المعاوقة العالية (High Impedance) و يقومان بعزل الـ bit Line عن العالم الخارجي.
- عند وضع القيمة 1 في طرف إختيار الشريحة CS فإن خـرج بـوابتي AND يعتمد على القيمة الموضوعة في طرف تمكين الكتابة $\overline{WE}$.
  - عند وضع القيمة 0 على الطرف $\overline{WE}$ يكون خرج بوابة AND العليـا مساوياً 1، و بالتالي يسمح العازل ثلاثي الحالة بمرور القيمة الموضوعة في طرف الدخل I إلى الـ bit Line و حدوث عملية كتابـة (Write)، أما بوابة AND السفلى فيكون خرجها مساوياً 0 فيدخل العازل الموجود في طرف الخرج O في حالة المعاوقة العالية و يقوم بعزل طرف الخرج عن الـ bit Line.

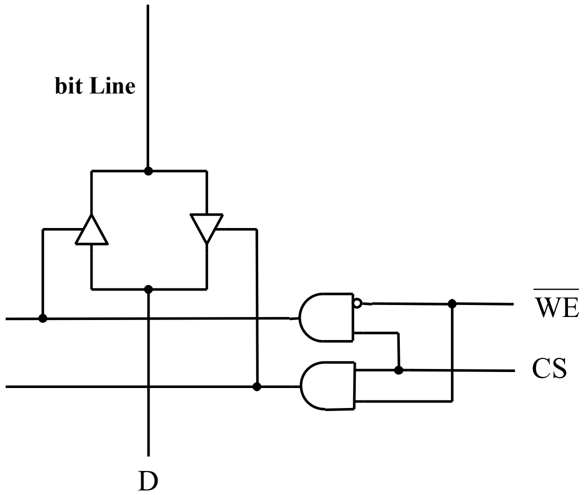
▪ عند وضع القيمة 1 على الطرف $\overline{WE}$ يكون خرج بوابة AND السفلى مساوياً 1، و بالتالي يسمح العازل ثلاثي الحالة بمرور القيمة الظاهرة على الـــ bit Line إلى طرف الخرج O و حدوث عملية قراءة (Read)، أما بوابة AND العليا فيكون خرجها مساوياً 0 فيدخل العازل الموجود في طرف الدخل I في حالة المعاوقة العالية و يقوم بعزل طرف الدخل عن الـــ bit Line.

و عليه يكون الشكل النهائي للبناء الداخلي لذاكرة الـــ RAM من نوع 4×3 كالتالي



أطراف الدخل و الخرج المشتركة للبيانات

عندما تكون أطراف الدخل و الخرج للبيانات مشتركة يتم وضع كلا العازلين في طرف واحد من الـــ bit Line، كما هو موضح بالشكل التالي:



زيادة سعة الذاكرة

لزيادة طول الذاكرة يتم إضافة Word Lines، و لزيادة عرض الذاكرة يتم إضافة bit Lines.

تدريب 5

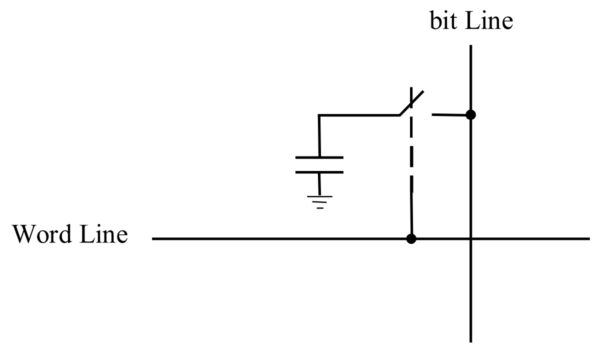
وضح البناء الداخلي لذاكرة RAM من نوع 4×8 بأطراف دخل و خرج مشتركة للبيانات.

2.4 ذاكرة الـــ SRAM

ذاكرة الـــ RAM التي قمنا بتوضيح بنائها الداخلي في ما سبق هي عبارة عن نوع من أنواع الـــ RAM يطلق عليه تسمية Static RAM، أو SRAM اختصاراً. و سميت بهذا الأسم لأنه لا يحدث تغير في محتوياتها بمرور الزمن، حيث تظل محتفظة بمحتوياتها كما هي ما دامت تغذيتها بالقدرة مستمرة. و ذلك بخلاف نوع آخر من ذاكرة الـــ RAM يطلق عليه تسمية Dynamic RAM أو DRAM، يفقد محتوياته بالتدريج مع مرور الزمن و يحتاج لإعادة كتابة المحتويات بصورة دورية. و الخلية التخزينية في الـــ SRAM، كما نعلم، هي عبارة عن مرجاح (Flip Flop). و تتميز ذاكرة الـــ SRAM بالسرعة العالية، و لكن يعيبها ارتفاع التكلفة، مقارنة بالـــ DRAM. لذلك يتم استخدام ذاكرة الـــ SRAM، بسعات صغيرة نسبياً، حيثما تكون السرعة العالية مطلوبة في الأنظمة الرقمية، مثل ذاكرة الكاش (Cache Memory) و ذاكرة الفيديو (Video Graphics).

3.4 ذاكرة الـــ DRAM

الخلية التخزينية في ذاكرة الـــ DRAM عبارة عن مكثف (Capacitor). كما هو موضح بالشكل التالي:



فالمكثف المشحون يخزن القيمة المنطقية 1 و المكثف غير المشحون يختزن القيمة المنطقية 0.

و المكثف المستخدم كخلية تخزينية في الـ DRAM يشغل مساحة من سطح شبه الموصل أقل بكثير من تلك التي يشغلها المرجاح (Flip Flop) المستخدم كخلية تخزينية في الـ SRAM، الأمر الذي يسمح بكثافة تخزينية أعلى في ذاكرة الـ DRAM. فذاكرة الـ DRAM تمتاز بإمكانية تصنيعها بسعات عالية و بتكلفة منخفضة، كما تمتاز بأن استهلاكها للقدرة الكهربائية أقل من الـ SRAM. لذلك يستخدم الـ DRAM حالياً بكثرة في أجهزة الحاسوب الشخصية (PC's) كذاكرة رئيسية (Main Memory). و يعيب ذاكرة الـ DRAM أنها تفقد محتوياتها بمرور الزمن، نظراً إلى أن المكثفات المستخدمة كخلايا تخزينية فيها هي مكثفات ذات تسريب (Leaky Capacitors) تفقد شحنتها بالتدريج. لذلك تحتاج ذاكرة الـ DRAM إلى إعادة كتابة محتوياتها، أي إلى إعادة شحن المكثفات، بصورة دورية، و تسمى هذه العملية بعملية الإنعاش (Refreshing). و تحتاج عملية الإنعاش إلى دوائر خاصة في النظام الرقمي أو داخل شريحة الذاكرة نفسها. كما يعيب ذاكرة الـ DRAM بطؤها مقارنة بذاكرة الـ SRAM، و إن عملية القراءة (Read) منها مدمرة لمحتوياتها، حيث أن عملية القراءة تعمل على تفريغ المكثف من شحنته، الأمر الذي يتطلب إتباع كل عملية قراءة من الـ DRAM بعملية إعادة كتابة لمحتويات الموقع الذي تمت قراءته. و حالياً توجد أنواع حديثة و متطورة من ذاكرة الـ DRAM مثل الـ SDRAM (Synchronous DRAM) أو الـ DRAM المتزامن، و فيه تمت زيادة سرعة الذاكرة عن طريق قراءة أو كتابة مجموعة من العناوين المتتالية مرة واحدة بصورة متزامنة أي متفقة مع إشارة التزامن (Clock) الرئيسية في النظام الرقمي، حيث يتم الوصول لأول عنوان في المجموعة بالطريقة المعتادة، و التي فيها شيء من البطء، أما بقية عناوين المجموعة فيتم الوصول إليها بطريقة أسرع بالتزامن مع إشارة التزامن الرئيسية في النظام الرقمي. كما تم في السنوات الأخيرة تطوير الـ SDRAM نفسه إلى ما يطلق عليه

DDR SDRAM أو (Double Data Rate SDRAM)، و فيـــه تمـــت مضـــاعفة سرعة الذاكرة عن طريق إجراء عمليتي قراءة أو كتابة في نبضة التزامن الواحدة، حيث تتم عملية منها في الحافة الصاعدة لنبضة التزامن، و العملية الأخرى في الحافة الهابطة لنبضة التزامن.

تدريب 6

وضح البناء الداخلي لذاكرة DRAM من نوع 4×8 بـــأطراف دخـــل و خرج مشتركة للبيانات.

أسئلة تقويم الذاتي

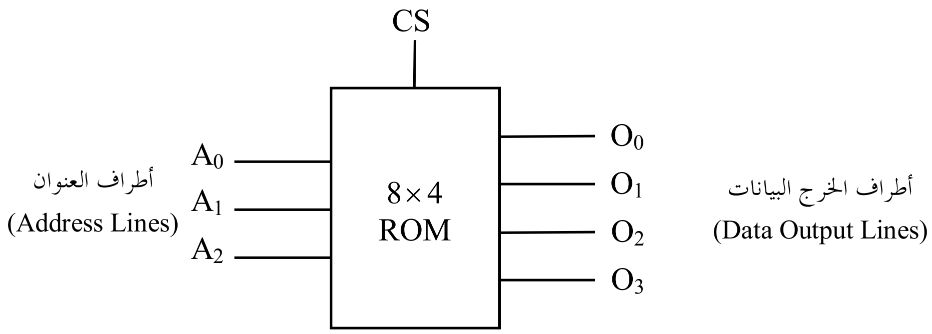
1- احد صفات أو خصائص ذاكرة الـــ RAM هو انها ذاكرة متطايرة (Volatile) اشرح هذه العبارة .

2- ما المقصود بعملية الانعاش (Refreshing) التي تجري في ذاكرة الـــ DRAM ؟

3- قارن بين كل من ذاكرة الـــ SRAM وذاكرة الـــ DRAM .

5. ذاكرة الـــ ROM

ROM هو إختصار لعبارة Read Only Memory أي ذاكرة القراءة فقط. فذاكرة الـــ ROM، كما تدل التسمية، يمكن قراءة محتوياتها فقط و لا يمكن الكتابة فيها، حيث أن محتوياتها ثابتة و غير قابلة للمحو أو التعديل. و إحتفاظ ذاكرة الـــ ROM بمحتوياتها غير مرتبط بتغذيتها بالقدرة، كما هو الحال في ذاكرة الـــ RAM، حيث تظل ذاكرة الـــ ROM محتفظة بمحتوياتها حتى بعد فصل التغذية بالقدرة عنها.
الشكل التالي يوضح المخطط المنطقي لشريحة ذاكرة ROM من نوع 8×4



لاحظ وجود طرف تحكم واحد فقط هو طرف إختيار الشريحة (Chip Select)

1.5 البناء الداخلي لذاكرة الـــ ROM

نظراً لثبات محتويات ذاكرة الـــ ROM فإنه من الممكن التعبير عن العلاقة ما بين عناوين مواقع الذاكرة و محتويات تلك المواقع باستخدام جدول صواب (Truth Table)، كما هو موضح أدناه

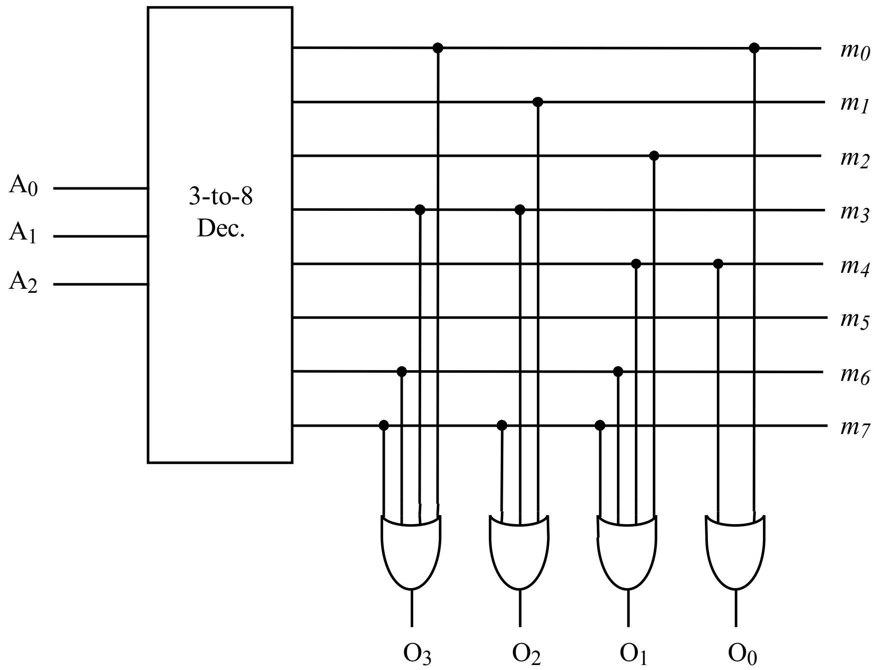
	Address			Contents			
#	A_2	A_1	A_0	O_3	O_2	O_1	O_0
0	0	0	0	1	0	0	1
1	0	0	1	0	1	0	0
2	0	1	0	0	0	1	0
3	0	1	1	1	1	0	0
4	1	0	0	0	0	1	1
5	1	0	1	0	0	0	0
6	1	1	0	1	0	1	0
7	1	1	1	1	1	1	0

ملاحظة

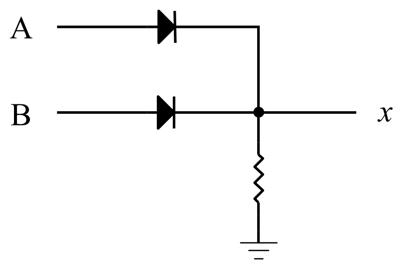
(محتويات الذاكرة المستخدمة هنا مجرد مثال و ليس لها أي مدلول معين)

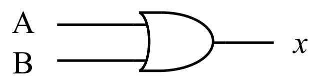
و بالتالي يمكن أن ننظر لذاكرة الـــ ROM كدائرة منطقية ترابطية (Combinational Logic Circuit) دخلها هو عناوين المواقع و خرجها عبارة عن محتويات تلك المواقع. و يمكن أن نقوم بتصميم الدائرة بسهولة باستخدام فك الشفرة و مشفر (Decoder & Encoder)، كما هو موضح أدناه

$$O_0 = \sum m\ (0,4)$$
$$O_1 = \sum m\ (2,4,6,7)$$
$$O_2 = \sum m\ (1,3,7)$$
$$O_3 = \sum m\ (0,3,6,7)$$



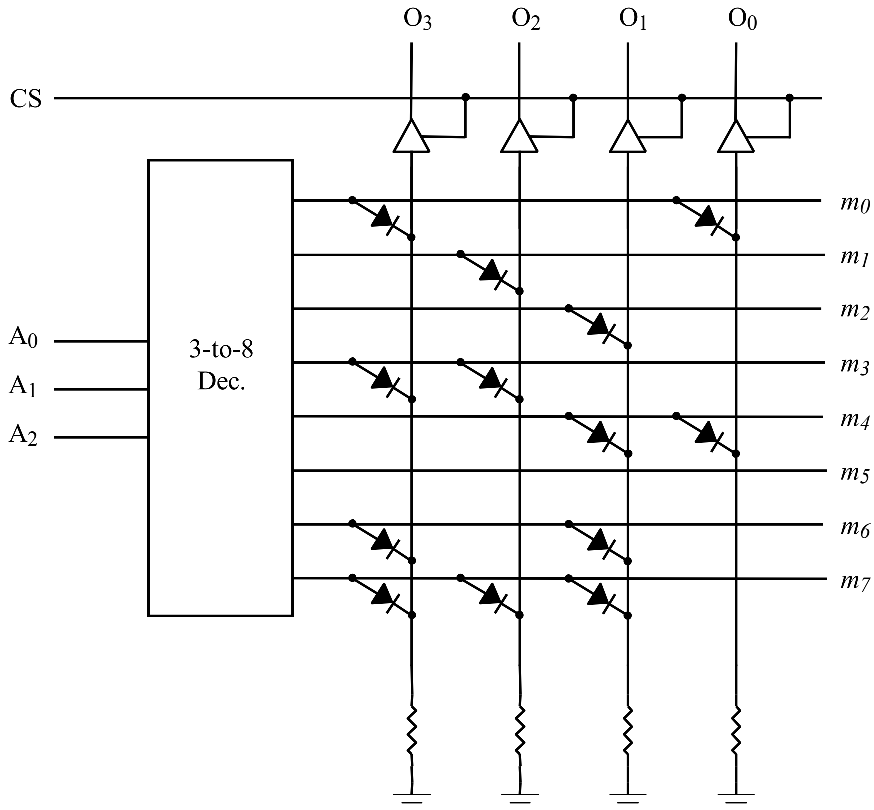
و بوابات OR المستخدمة في المشفر (Encoder) هنا هي من أبسط الأنواع، و تتكون من مجموعة من (الديودات) الثنائيات (Diodes)، كما هو موضح أدناه





لاحظ أن خرج الدائرة x يساوي صفراً في حالة واحدة فقط وهي عدم مرور تيار في المقاومة (أي عندما يكون كل من الديودين في حالة فصل) ويكون هذا عندما يكون طرفي الدخل A و B موصلان بصفر. وفي حال وضع 1 في أي من طرفا الدخل فإن الديود المناظر يكون في حالة التوصيل فيكون الخرج x يساوي 1.

و بالتالي يكون البناء الداخلي لذاكرة الـــ ROM من نوع 4×8 التي يوضحها المخطط المنطقي و جدول الصواب لها أعلاه هو



تذكر أن الموصلات الرأسية تسمى الـــ bit Lines، و الموصلات الأفقية تسمى الـــ Word Lines.

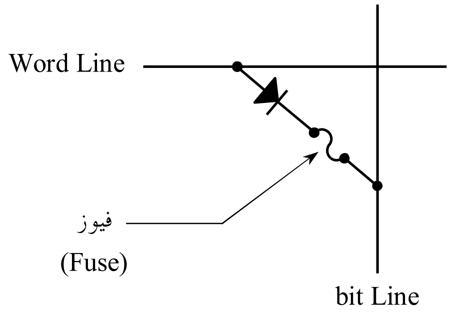
لاحظ أنه حيثما يوجد 1 في الذاكرة يتم وضع ديود ليربط ما بين الـ bit Line و الـ Word Line، و حيثما يوجد 0 لا يتم وضع ديود. فعندما ينشط الـ Word Line المقابل للموقع المطلوب قراءته، و يصبح جهده الكهربائي مساوياً للجهد المرتفع (High) الذي يمثل القيمة المنطقية 1، تنحاز جميع الديودات المتصلة به إنحيازاً أمامياً و تصبح عبارة عن دوائر مقصورة (Short Circuit)، فتقوم بنقل الجهد المرتفع إلى الـ bit Lines التي تتصل بها. بمعنى أن أي bit Line يربطه ديود بالـ Word Line النشط يظهر عليه جهد مرتفع يمثل القيمة المنطقية 1، و أي bit Line لا يربطه ديود بالـ Word Line النشط يظهر عليه جهد الأرض (Ground) المنخفض الذي يمثل القيمة المنطقية 0.

يتم تحديد مواضع الديودات، أي محتويات ذاكرة الـ ROM، أثناء عملية التصنيع، و بعد ذلك لا يمكن تغيير تلك المحتويات حيث إن ذلك يتطلب إزالة أو إضافة ديودات، و هذا بالطبع غير ممكن عملياً.

لاحظ أنه لا يوجد تخزين فعلي في ذاكرة الـ ROM، مثلما هو الحال في ذاكرة الـ RAM، و إنما يتم توليد محتويات أي موقع من مواقع الذاكرة من عنوانه عن طريق إجراء عمليات منطقية.

2.5 ذاكرة الـ PROM

المقصود هنا هو الـ Programmable ROM أي ذاكرة الـ ROM القابلة للبرمجة. و هذا النوع من الذاكرة يشبه في تركيبه الداخلي ذاكرة الـ ROM إلى حد كبير، إلا أنه يأتي من المصنع في صورة خام فيها كل خلية من خلاياه يوجد بها ديود يربط ما بين الـ bit Line و الـ Word Line، أي أن جميع خلاياه تحتوي على القيمة المنطقية 1. و يقوم مستخدم الـ PROM ببرمجته، أي تحديد محتوياته، و ذلك باستخدام جهاز برمجة خاص. و في عملية البرمجة يجب إزالة الديود من كل خلية نريد تخزين القيمة المنطقية 0 فيها، حيث يكون أي ديود متصلاً على التوالي مع فيوز (Fuse)، كما هو موضح بالشكل التالي:



و يتم إزالة الديود غير المرغوب فيه بفصله عن طريق صهر الفيوز المتصل معه على التوالي بإمرار نبضة تيار عالٍ فيه. و عملية برمجة الـــ PROM، أي الكتابة فيه، عملية بطيئة جداً إذا ما قورنت بعملية القراءة منه. و بعد برمجة الـــ PROM يصبح عبارة عن ROM و لا يمكن بعد ذلك محو أو تغيير محتوياته (يمكن فقط إضافة المزيد من الـــ 0's).

3.5 ذاكرة الـــ EPROM

المقصود هنا هو الـــ Erasable PROM أي ذاكرة الـــ PROM القابلة للمسح و إعادة البرمجة. و يتم مسح ذاكرة الـــ EPROM عن طريق تعريضها للأشعة فوق البنفسجية (Ultra Violet Light) عبر نافذة زجاجية بشريحة الذاكرة لفترة زمنية تتراوح ما بين 15 و 20 دقيقة، و عملية المسح هنا تشمل الذاكرة بأكملها و من غير الممكن اختيار جزء معين منها ليتم مسحه دون سواه، مع ملاحظة أنه بعد الانتهاء من عملية المسح يجب تغطية النافذة الزجاجية لتجنب تعرض الذاكرة للمسح بصورة عرضية، حيث أن ضوء الشمس العادي يحتوي على نسبة من الأشعة فوق البنفسجية. و بعد مسح ذاكرة الـــ EPROM يمكن برمجتها مرة أخرى. و في الإمكان تكرار عملية المسح و إعادة البرمجة أي عدد من المرات. و برمجة الـــ EPROM تتم بطريقة مشابهة لبرمجة الـــ PROM، حيث يجب أن تتم خارج النظام الرقمي الذي ستستخدم فيه الذاكرة، و يستخدم في إجرائها جهاز برمجة خاص، و هي عملية بطيئة نسبياً.

4.5 ذاكرة الـــ EEPROM

المقصود هنا هو الـــ Electrically Erasable PROM أي ذاكرة الـــ PROM القابلة للمسح كهربائياً، حيث إن هذا النوع من ذاكرة الـــ PROM يمكن مسحه عن طريق تسليط نبضة جهد عالٍ على الخلية التخزينية. و على الرغم من أن هذا النوع من ذاكرة الـــ PROM أكثر تكلفة و أقل كثافة تخزينية من الـــ EPROM إلا أنه يمتاز بإمكانية اختيار الموقع من الذاكرة الذي يتم مسحه، و بإمكانية إجراء عملية المسح و إعادة البرمجة على الذاكرة و هي في مكانها داخل النظام الرقمي دون الحاجة إلى إزالتها منه كما هو الحال في الـــ EPROM، حيث أن دوائر المسح و إعادة البرمجة مضمنة داخل شريحة الـــ EEPROM نفسها. و ذاكرة الـــ EEPROM بخصائصها هذه أصبحت شبيهة إلى حد كبير بذاكرة الـــ RAM من حيث إمكانية الكتابة في أي موقع من مواقعها في أي وقت، إلا أنها تختلف عن الـــ RAM في أنها ذاكرة غير متطايرة (Nonvolatile)، حيث أن إحتفاظها بمحتوياتها غير مرتبط باستمرار تغذيتها بالقدرة، كما أن عملية مسح و إعادة برمجة موقع معين من مواقع الـــ EEPROM أبطأ بكثير من عملية الكتابة في الـــ RAM. و معلومة أخيرة عن ذاكرة الـــ EEPROM يجب الإلمام بها و هو أنها تتعرض للتلف من تكرار عملية المسح، حيث تتلف بعد عدد معين من مرات المسح.

5.5 ذاكرة الـــ Flash

تجمع ذاكرة الـــ Flash بين الخصائص المرغوبة لكل من ذاكرة الـــ EPROM و ذاكرة الـــ EEPROM، حيث تتميز بالكثافة التخزينية العالية و قلة التكلفة التي تميز الـــ EPROM، مع إمكانية المسح و إعادة البرمجة داخل النظام الرقمي التي تميز الـــ EEPROM. و لكن عملية مسح ذاكرة الـــ Flash تختلف قليلاً عن عملية مسح ذاكرة الـــ EEPROM، ففي حين يمكن مسح أي موقع من مواقع الـــ EEPROM منفرداً يتم مسح المواقع في ذاكرة الـــ Flash في شكل مجموعة (Block) من المواقع

المتتالية. و بما أن ذاكرة الـ Flash هي في الأساس ذاكرة EEPROM فإنها تتعرض للتلف من تكرار عملية المسح، حيث تتلف بعد حوالي 10,000 عملية مسح. و تتوفر ذاكرة الـ Flash حالياً بأشكال عديدة و أحجام صغيرة و أسعار رخيصة، و بسعات عالية تصل إلى 1 GB، و تستخدم للتخزين الثانوي (Secondary Storage) في الأجهزة الرقمية الصغيرة الحجم مثل الكاميرات الرقمية (Digital Cameras) و مشغلات الموسيقى (MP3 Players) و الهواتف المحمولة (Mobile Phones)، بل و تكاد الـ USB Flash Drives، التي يطلق عليها أيضاً تسمية الـ Thumb Drives أو الـ Pen Drives، أن تحل محل وسائط التخزين الثانوي المحمولة (Removable Secondary Storage) الأخرى مثل الأقراص المرنة (Floppy Disks) و الأقراص الضوئية (Optical Disks) كالـ CD-ROM و الـ DVD-ROM نظراً لصغر حجمها و سهولة حملها و استخدامها.

أسئلة تقويم الذاتي

1- وضح كيف يتم مسح ذاكرة الـ EPROM .

2- قارن بين ذاكرة الـ EPROM وذاكرة الـ EEPROM .

3- حدد أوجه الاختلاف بين ذاكرة الـ EEPROM وذاكرة الـ RAM .

4- عدد مميزات ذاكرة الـ Flash .

5- ذاكرة الـ Flash هي في الأساس ذاكرة .... ، وهي تتعرض لـ......... من تكرار عملية .... حيث تتلف بعد حوالي .... عملية .... أكمل

6. التخزين الثانوي (Secondary Storage)

بما أن الذاكرة (Memory) في الأنظمة الرقمية تصلح للتخزين المؤقت لكميات قليلة فقط من البرامج و البيانات بغرض المعالجة (Processing)، تظهر الحاجة لتخزين كميات كبيرة من البيانات و البرامج تخزيناً دائماً و بتكلفة قليلة. يأتي هنا دور التخزين الثانوي (Secondary Storage)، حيث تسمح وسائط التخزين الثانوي، مثل الأشرطة الممغنطة (Magnetic Tapes) و الأقراص الممغنطة (Magnetic Disks) و الأقراص الضوئية (Optical Disks)، بتخزين كميات ضخمة من البيانات تخزيناً دائماً و بتكلفة قليلة. إلا أن سرعة الكتابة في أجهزة التخزين الثانوي و القراءة منها أبطأ بكثير من سرعة التعامل مع الذاكرة، لذلك يجب نقل البرامج و البيانات، أو أجزاء منها، بصورة مؤقتة إلى الذاكرة الرئيسية (Main Memory) لمعالجتها، و ذلك لضمان سرعة المعالجة.

1.6 الأشرطة الممغنطة (Magnetic Tapes)

تعتبر الأشرطة الممغنطة من أقدم وسائط التخزين التي استخدمت في أجهزة الحاسوب و نظم المعلومات. و الشريط الممغنط عبارة عن شريط بلاستيكي مغطى بطبقة رقيقة من مادة مغناطيسية، يتم تخزين البيانات فيه عن طريق مغنطة مناطق معينة من المادة التي تغطيه بمجال مغناطيسي في إتجاه معين لتمثيل القيمة المنطقية 1، و بمجال مغناطيسي في الإتجاه المعاكس لتمثيل القيمة المنطقية 0. و تتم الكتابة على الشريط بواسطة رأس الكتابة (Write Head) الذي يستقبل الـــ bits المطلوب كتابتها و يقوم بتوليد المجال المغناطيسي الذي يقوم بمغنطة الشريط بالصورة المناسبة لتمثيل تلك الـــ bits أثناء مرور الشريط أمامه بسرعة ثابتة. أما عملية القراءة فتتم بواسطة رأس القراءة (Read Head) الذي يقوم أثناء مرو الشريط أمامه بتحويل حالة المغنطة للشريط إلى نبضات كهربائية تمثل الـــ bits المخزنة. و عملية الكتابة على الشريط الممغنط أو القراءة منه تتم بصورة تتابعية (Sequential)، فقراءة معلومة معينة في وسط الشريط أو نهايته

يتطلب قراءة الشريط منذ بدايته و حتى موقع المعلومة المطلوبة، لذلك تستخدم الأشرطة الممغنطة حالياً فقط للتخزين الاحتياطي (Backup) و للتخزين طويل الأجل للبيانات القديمة.

2.6 الأقراص الممغنطة (Magnetic Disks)

تشبه الأقراص الممغنطة في مبدأ عملها الأشرطة الممغنطة، فالقرص الممغنط عبارة عن قرص بلاستيكي أو معدني مغطى على وجهيه بطبقة رقيقة من مادة مغناطيسية، و يتم تخزين البيانات فيه عن طريق مغنطة مناطق معينة من المادة التي تغطيه بمجال مغناطيسي في اتجاه معين لتمثيل القيمة المنطقية 1، و بمجال مغناطيسي في الاتجاه المعاكس لتمثيل القيمة المنطقية 0. إلا أن الإختلاف الرئيسي بين الأقراص الممغنطة و الأشرطة الممغنطة هو أن الدخول للأقراص بغرض القراءة أو الكتابة يكون مباشراً (Direct Access) و ليس تتابعياً (Sequential) مثل الأشرطة، مما يسمح بالوصول لأي معلومة بالقرص بصورة سريعة، حيث إن رأس الكتابة و القراءة يمكنه الوصول لأي نقطة على سطح القرص بسهولة أثناء دورانه. و الأقراص الممغنطة تأتي في أشكال مختلفة و تتفاوت سعاتها تفاوتاً كبيراً، ابتداء من الأقراص المرنة (Floppy Disks) التي تقل سعتها عن 2 MB، مروراً بالأقراص ذات السعة العالية (High Capacity Disks) مثل تلك المستخدمة في الـــ Zip® drive و الـــ SuperDisk™ drive و الـــ HiFD™ drive، و التي تتجاوز سعتها 100 MB و تصل إلى 250 MB، و حتى الأقراص الصلبة (Hard Disks) شائعة الاستخدام و التي وصلت سعاتها حالياً إلى 400 GB وهي في زيادة مضطردة.

3.6 الأقراص الضوئية (Optical Disks)

القرص الضوئي عبارة عن قرص مصنوع من مادة بلاستيكية شفافة و مغطى في أعلاه بطبقة معدنية رقيقة عاكسة للضوء. يتم كتابة البيانات الثنائية على القرص باستخدام شعاع دقيق جداً من الليزر عالي الطاقة يتم تسليطه على السطح المعدني العاكس للقرص من أسفل فيقوم بعمل حُفر دقيقة جداً على الطبقة المعدنية العاكسة بتأثير الحرارة العالية، و تلك الحفر الدقيقة تمثل الـــ bits، حيث أن وجود حفرة يمثل القيمة المنطقية 0 و عدم وجودها يمثل القيمة المنطقية 1. و استخدام ضوء الليزر هنا يسمح بكثافة تخزينية عالية جداً على سطح القرص، حيث أنه من الممكن تركيز ضوء الليزر في شعاع متناهي في الدقة. يتم قراء البيانات المخزنة على القرص باستخدام شعاع ليزر دقيق منخفض الطاقة يتم تسليطه من أسفل على السطح المعدني العاكس للقرص فحيثما توجد حفرة يتم إمتصاص شعاع الليزر و حيثما لا توجد حفرة يتم عكسه، و تقوم عدسة بالتقاط ضوء الليزر المنعكس من على الطبقة المعدنية و تسليطه على ديود ضوئي (Photodiode) يقوم بتحويله إلى نبضات كهربائية تمثل الـــ bits المخزنة. يتم تخزين الـــ bits على القرص الضوئي في شكل مسار لولبي (spiral) متصل و يقوم شعاع الليزر المستخدم في القراءة أو الكتابة بمتابعة ذلك المسار اللولبي بدقة عالية أثناء دوران القرص. و تتوفر الأقراص الضوئية حالياً بأشكال عديدة منها الأقراص المدمجة (CD's) بنوعياتها المختلفة كالـــ CD-ROM الذي يمكن قراءته فقط، و الـــ CD-R الذي يمكن الكتابة فيه و لكن لا يمكن مسح أو تعديل البيانات المكتوبة، و الـــ CD-RW الذي يمكن مسحه و إعادة الكتابة فيه، إضافة إلى أقراص الـــ DVD التي تمتاز بسعة تخزينية أكبر من أقراص الـــ CD نظراً لاستخدامها لشعاع ليزر بطول موجي أقصر مما يسمح بتركيزه في شعاع أدق، و بالتالي كثافة تخزين أعلى. و تتوفر أقراص الـــ DVD أيضاً بنوعيات مختلفة مثل الـــ DVD-ROM و الـــ DVD-R و الـــ DVD+RW. هذا و يتوقع الخبراء أن تحل أقراص الـــ DVD، مع إنخفاض تكلفتها، كلياً محل أقراص الـــ CD قريباً.

أسئلة تقويم الذاتي

1- حدد ما هو الاختلاف الرئيس بين الأشرطة الممغنطة **(Magnetic Tapes)** ، والأقراص الممغنطة **(Magnetic Disks)**.

2- علل لماذا امتازت اقراص الـــ DVD بسعة تخزينية أكبر من أقراص الـــ CD .

3- ما هو القرص الضوئي وكيف تتم كتابة البيانات الثنائية عليه .

الخلاصة

تعرفنا في هذه الوحدة على تقنيات التخزين المختلفة المستخدمة في الأنظمة الرقمية و مميزات و استخدامات كل تقنية منها. حيث بدأنا بالذاكرة (Memory) فقمنا بتوضيح التنظيم المنطقي لها، ثم تعرفنا على شريحة الذاكرة و أطراف التوصيل لها و طريقة ربط شرائح الذاكرة مع بعضها البعض بهدف زيادة عرض الذاكرة أو طولها. ثم إنتقلنا بعد ذلك لتوضيح البناء الداخلي لذاكرة الـــ RAM و الفرق بين الـــ SRAM و الـــ DRAM و خصائص و استخدامات كل منهما. بعد ذلك قمنا بتوضيح البناء الداخلي لذاكرة الـــ ROM، و الفرق بين أنواع الـــ ROM المختلفة مثل الـــ PROM و الـــ EPROM و الـــ EEPROM و الـــ Flash، و خصائص و استخدامات كل نوع. و في نهاية الوحدة تعرفنا على مختلف تقنيات التخزين الثانوي (Secondary Storage) و مبدأ عمل و خصائص و استخدامات كل تقنية منها.

إجابات التدريبات

تدريب 1:

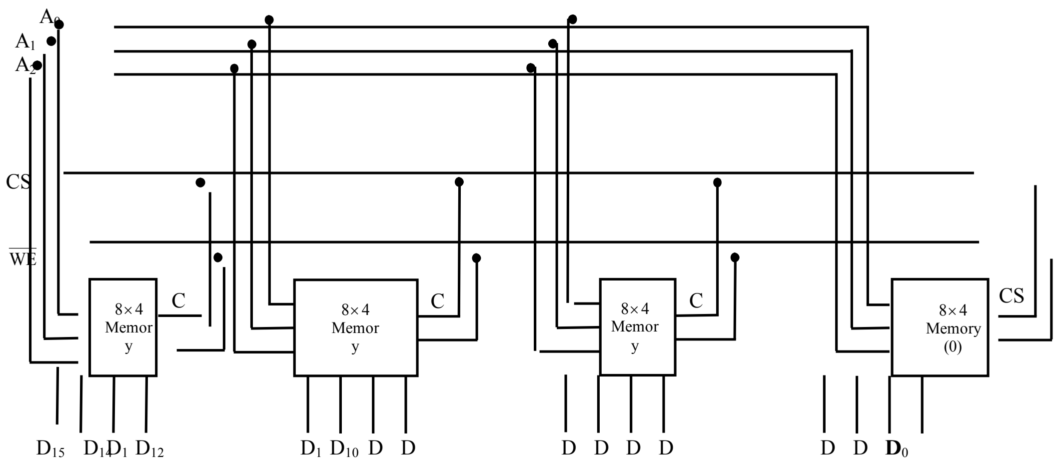
1. أطراف الدخل و الخرج للبيانات منفصلة

(أ) 49 (ب) 40

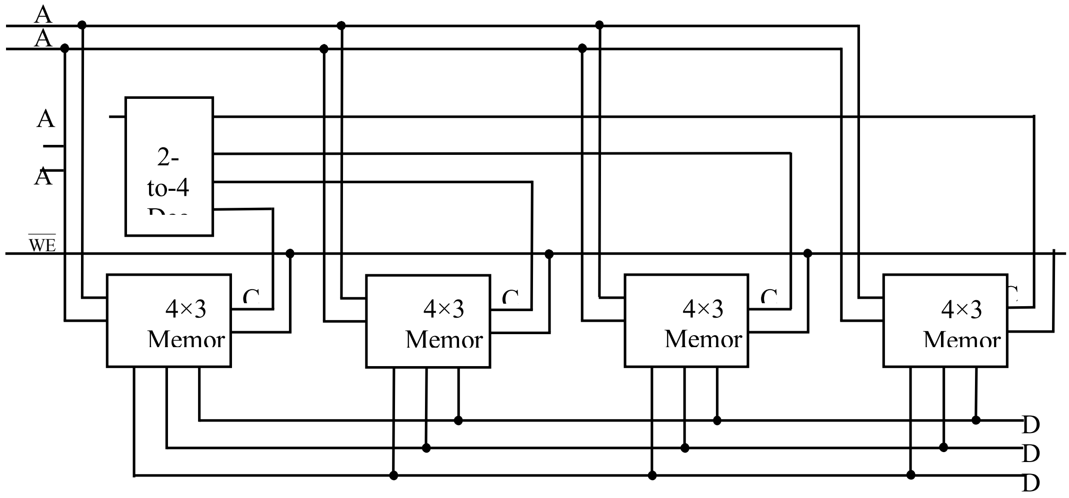
2. أطراف الدخل و الخرج للبيانات مشتركة

(أ) 33 (ب) 32

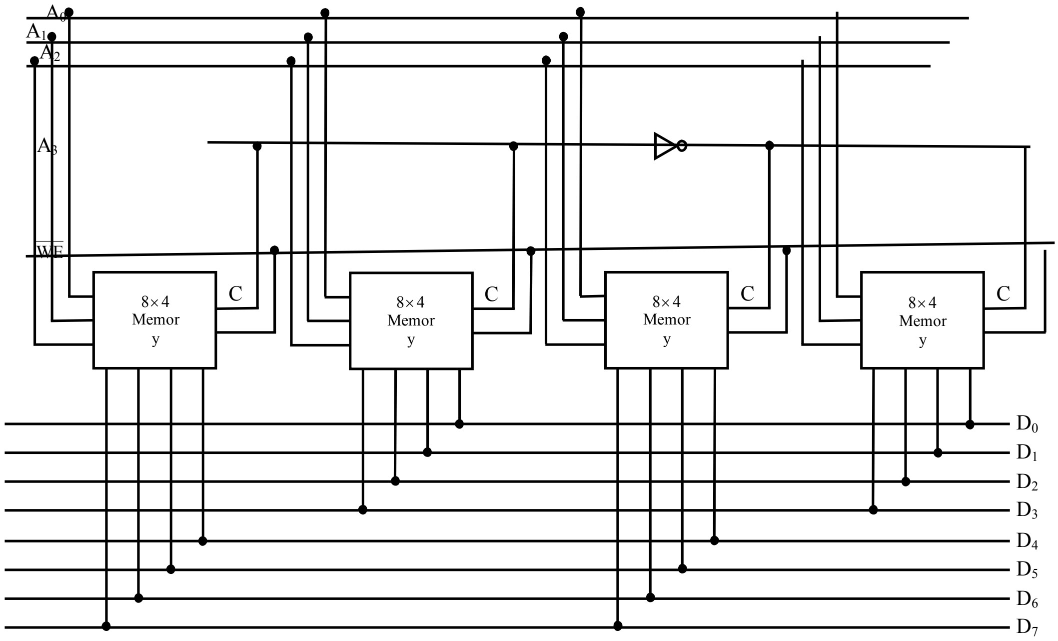
تدريب 2:



تدريب 3:

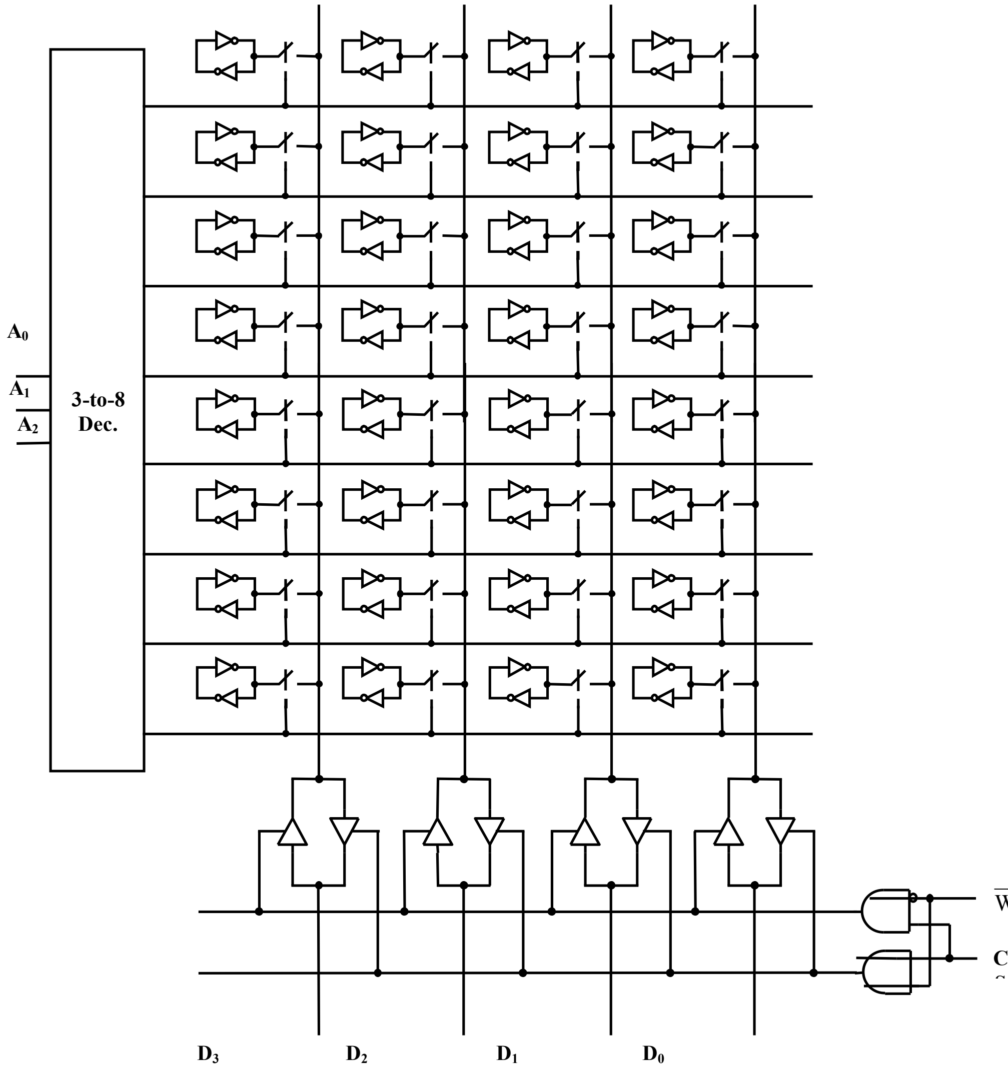


تدريب 4:

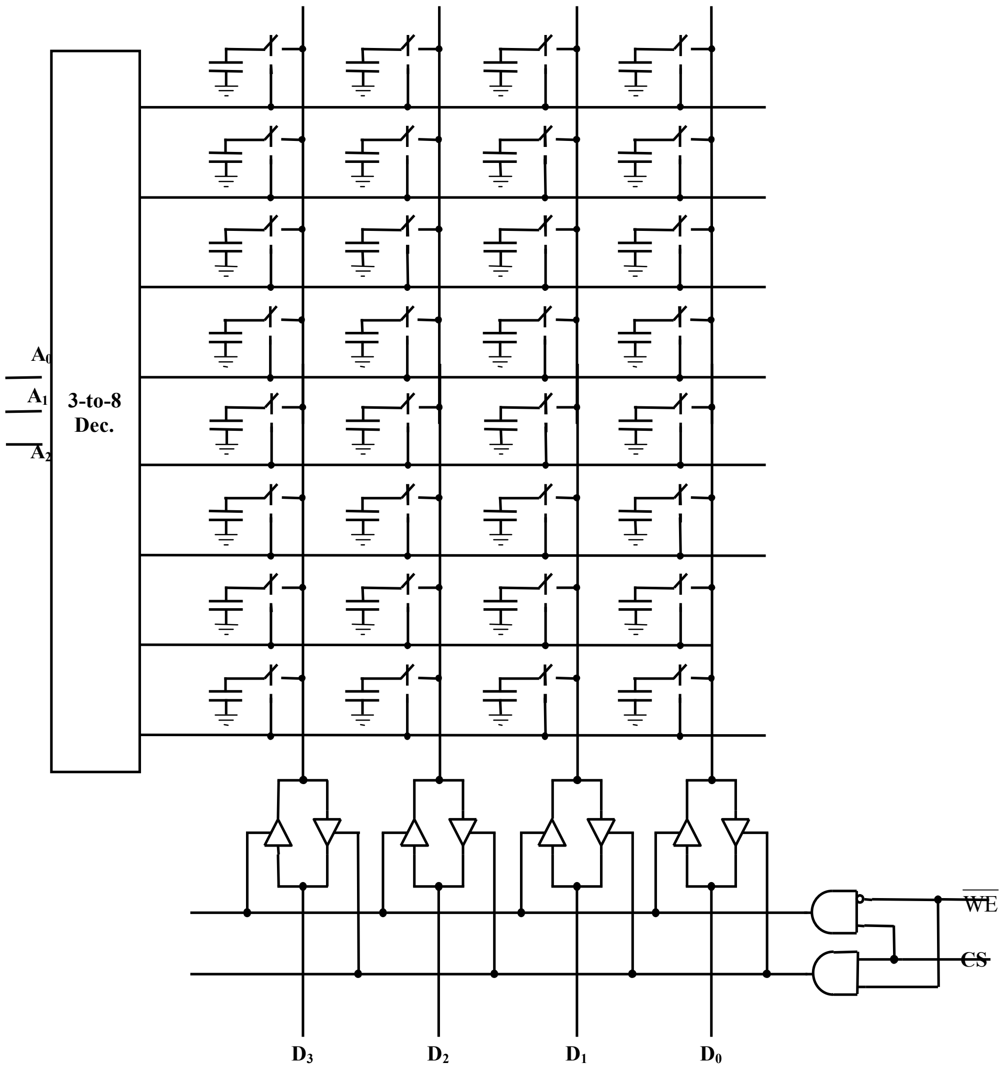


A0
A1
A2
A3
WE
8×4
Memor
y
C
8×4
Memor
y
C
8×4
Memor
y
C
8×4
Memor
y
C
D0
D1
D2
D3
D4
D5
D6
D7

تدريب 5:



تدريب 6:



مسرد المصطلحات

- **التنظيم المنطقي للذاكرة (Logical Memory Organization)**

المقصود هو صورة الذاكرة كما يراها مبرمج النظام الرقمي (Programmer).

- Length

هو طول الذاكرة ويرمز له بـــ M وهو عبارة عن عدد صحيح يمثل عدد مواقع الذاكرة.

- Width

هو عرض الذاكرة ويرمز له بـــ N وهو عبارة عن عدد صحيح يمثل طول الموقع الواحد أي عدد خاناته الثنائية (bits) أو خلاياه التخزينية .

- Word

هو عبارة عن موقع الذاكرة الذي تتم قراءته منها أو كتابته فيها كاملاً ولا يمكن تجزئته ويطلق هذا المصطلح بشكل عام على موقع الذاكرة بغض النظر عن طوله .

- Byte

هو موقع الذاكرة في الحواسيب الشخصية (PC`s) ، والتي عادة ما يكون طول الموقع فيها مساوياً bits – 8 .

- $\overline{WE}$

أي طرف تمكين الكتابة (Write Enable)، و مهمته تحديد نوع العملية المطلوب إجراؤها على الذاكرة فإما عملية كتابة (Write) وفيها يكون $\overline{WE} = 0$ ، أو عملية قراءة (Read) وفيها يكون $\overline{WE} = 1$ ، و يرمز لهذا الطرف أيضاً بالرمز $R/\overline{W}$ أو بالرمز $\overline{W}$.

- CS

هو طرف إختيار الشريحة (Chip Select)، و هو عبارة عن خط سماح (Enable) يسمح لشريحة الذاكرة بالعمل كالمعتاد عند وضع القيمة 1 فيه، و يبطل عملها و يعزل

أطراف الدخل و الخرج للبيانات عن العالم الخارجي بمعاوقة عالية (High Impedance) عند وضع القيمة 0 فيه.

• أطراف البيانات Data Lines

مصطلح يطلق على أطراف الدخل والخرج للبيانات في شريحة الذاكرة عندما تكون هذه الأطراف مشتركة .

• **ذاكرة الـــ RAM**

مصطلح RAM هو اختصار لعبارة Random Access Memory أي ذاكرة الدخول العشوائي. و سبب هذه التسمية هو أنه من الممكن في هذا النوع من الذاكرة الدخول إلى أي موقع من مواقعها بصورة عشوائية عن طريق عنوان ذلك الموقع دون الحاجة لإتباع ترتيب معين في الدخول.

• Sequential Access Memory

وهي ذاكرة الدخول ألتتابعي و الدخول على مواقع هذه الذاكرة يجب أن يكون من البداية وحسب ترتيب التخزين ، فلقراءة أي موقع من مواقع هذا النوع من الذاكرة يجب قراءة الذاكرة من بدايتها و حتى ذلك الموقع.

• **ذاكرة الـــ SRAM**

هي عبارة عن نوع من أنواع الـــ RAM يطلق عليه تسمية Static RAM، أو SRAM إختصاراً. و سميت بهذا الأسم لأنه لا يحدث تغير في محتوياتها بمرور الزمن، حيث تظل محتفظة بمحتوياتها كما هي ما دامت تغذيتها بالقدرة مستمرة.

• DRAM

هي عبارة عن نوع من أنواع الـــ RAM يطلق عليه تسمية Dynamic RAM يفقد محتوياته بالتدريج مع مرور الزمن و يحتاج لإعادة كتابة المحتويات بصورة دورية.

• **ذاكرة الـــ ROM**

وهي إختصار لعبارة Read Only Memory أي ذاكـــرة القـــراءة فقط. فذاكـــرة الـــ ROM، كما تدل التسمية، يمكن قراءة محتوياتها فقط و لا يمكن الكتابة فيها، حيث أن محتوياتها ثابتة و غير قابلة للمحو أو التعديل. و احتفاظ ذاكرة الـــ ROM بمحتوياتها غير مرتبط بتغذيتها بالقدرة .

• **ذاكرة الـــ PROM**

وهي الـــ Programmable ROM أي ذاكرة الـــ ROM القابلة للبرمجة. و هذا النوع من الذاكرة يشبه في تركيبه الداخلي ذاكرة الـــ ROM إلى حد كبير.

ذاكرة الـــ EPROM

وهي الـــ Erasable PROM أي ذاكرة الـــ PROM القابلة للمسح و إعادة البرمجة و يتم مسح ذاكرة الـــ EPROM عن طريق تعريضها للأشعة فوق البنفسجية (Ultra Violet Light) عبر نافذة زجاجية بشريحة الذاكرة .

ذاكرة الـــ EEPROM

المقصود هنا هو الـــ Electrically Erasable PROM أي ذاكرة الـــ PROM القابلة للمسح كهربائياً.

ذاكرة الـــ Flash

تجمع ذاكرة الـــ Flash بين الخصائص المرغوبة لكل من ذاكرة الـــ EPROM و ذاكرة الـــ EEPROM .

• **الأشرطة الممغنطة (Magnetic Tapes)**

الشريط الممغنط عبارة عن شريط بلاستيكي مغطى بطبقة رقيقة من مادة مغناطيسية، يتم تخزين البيانات فيه عن طريق مغنطة مناطق معينة من المادة التي تغطيه بمجال مغناطيسي في اتجاه معين لتمثيل القيمة المنطقية 1، و بمجال مغناطيسي في الاتجاه المعاكس لتمثيل القيمة المنطقية 0 ، و عملية الكتابة على الشريط الممغنط أو القراءة منه تتم بصورة تتابعيه (Sequential) .

• **الأقراص الممغنطة (Magnetic Disks)**

تشبه الأقراص الممغنطة في مبدأ عملها الأشرطة الممغنطة، فالقرص الممغنط عبارة عن قرص بلاستيكي أو معدني مغطى على وجهيه بطبقة رقيقة من مادة مغناطيسية، و يتم تخزين البيانات فيه عن طريق مغنطة مناطق معينة من المادة التي تغطيه بمجال مغناطيسي في إتجاه معين لتمثيل القيمة المنطقية 1، و بمجال مغناطيسي في الاتجاه المعاكس لتمثيل القيمة المنطقية 0. إلا أن الاختلاف الرئيسي بين الأقراص الممغنطة و الأشرطة الممغنطة هو أن الدخول للأقراص بغرض القراءة أو الكتابة يكون مباشراً (Direct Access) ، و الأقراص الممغنطة تأتي في أشكال مختلفة و تتفاوت سعاتها تفاوتاً كبيراً، ابتداء من الأقراص المرنة (Floppy Disks) التي تقل سعتها عن 2 MB، مروراً بالأقراص ذات السعة العالية (High Capacity Disks) مثل تلك المستخدمة في الــ Zip® drive و الــ Super Disk™ drive و الــ HiFD™ drive، و التي تتجاوز سعتها 100 MB و تصل إلى 250 MB، و حتى الأقراص الصلبة (Hard Disks) الشائعة الاستخدام و التي وصلت سعاتها حالياً إلى 400 GB و في زيادة مضطردة.

• **الأقراص الضوئية (Optical Disks)**

القرص الضوئي عبارة عن قرص مصنوع من مادة بلاستيكية شفافة و مغطى في أعلاه بطبقة معدنية رقيقة عاكسة للضوء. يتم كتابة البيانات الثنائية على القرص باستخدام شعاع دقيق جداً من الليزر عالي الطاقة يتم تسليطه على السطح المعدني العاكس للقرص من أسفل فيقوم بعمل حُفر دقيقة جداً على الطبقة المعدنية العاكسة بتأثير الحرارة العالية، و تلك الحفر الدقيقة تمثل الــ bits، حيث أن وجود حفرة يمثل القيمة المنطقية 0 و عدم وجودها يمثل القيمة المنطقية 1 ، و تتوفر الأقراص الضوئية حالياً بأشكال عديدة منها الأقراص المدمجة (CD's) بنوعياتها المختلفة كالــ CD-ROM الذي يمكن قراءته فقط، و الــ CD-R الذي يمكن الكتابة فيه و لكن لا يمكن مسح أو تعديل البيانات المكتوبة، و الــ CD-RW الذي يمكن مسحه و إعادة الكتابة فيه، إضافة إلى أقراص

الـــ DVD التي تمتاز بسعة تخزينية أكبر من أقراص الـــ CD نظراً لاستخدامها لشعاع ليزر بطول موجي أقصر مما يسمح بتركيزه في شعاع أدق، و بالتالي كثافة تخزين أعلى. و تتوفر أقراص الـــ DVD أيضاً بنوعيات مختلفة مثل الـــ DVD-ROM و الـــ DVD-R و الـــ DVD+RW. هذا و يتوقع الخبراء أن تحل أقراص الـــ DVD، مع إنخفاض تكلفتها، كلياً محل أقراص الـــ CD قريباً.