الفصل الرابع

الجبر البولياني والتوابع المنطقية

Boolean Algebra and Logic Functions

يُعدّ جبر بول Boolean Algebra والمنطق الثنائي الذي سبق وأشرنا إليه بالفصل الأول متكافئان، إلا انه سيتم هنا الاشارة إلى أهم القوانين الجبرية وتطبيقاتها على التعبير المنطقي تحت مفهوم جبر بول وإلى البوابات المنطقية وتوابعها المختلفة تحت مفهوم المنطق الثنائي.

٤-١ جبر بول

ظهر العلم المسمى بجبر بول، (الجبر البولياني Boolean Algebra) لأول مرة عام 1854م، إلا انه لم يطبق او يستخدم قبل العام 1934 وقد سمي بهذا الاسم نسبة إلى العالم الانكليزي جورج بول. يعرّف الجبر البولياني على انه المجموعة B التي تحتوي المتحولات المنطقية والثوابت المنطقية (0 و1) والعمليات المنطقية الأساسية NOT و AND و OR . يغطي جبر بول برغم المحدودية التي يبدو بها بالمقارنة مع الجبر العادي (بسبب محدودية العمليات المعرّفة فيه والقيم المسموح التعامل بها) بعض جوانب الضعف للجبر العادي المتعلقة بجوانب الفكر. فمثلاً، التفاح يمثل نوع من انواع الفواكه، وكذلك البرتقال. وانطلاقاً من ان التعامل بالمنطق يكون مع متحولات منطقية، كل منها يأخذ قيمتين فقط، صح أو خطأ، يمكن ان يرمز للتفاح Apple بالمتحول المنطقي A وللبر تقابل Orange بالمتحول المنطقي O وللفواكه Fruit بالمتحول المنطقي F. يأخذ كل من هذه المتحولات المنطقية الثلاثة: A, O, F قيمتين (مثلاً، المتحول A يمكن ان يكون A = 1 وهذا يعني تفاح ويمكن ان يكون A = 0 وهذا يعني غير تفاح ويكتب ذلك بالشكل 'A). يمكن بالجبر العادي التعبير عن نفس المثال بالشكل A = F وكذلك O = F وهذا يعني ان A = O .

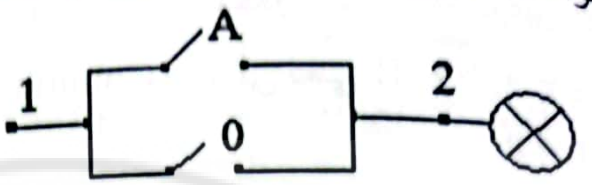
هذا الاستنتاج صحيح في الجبر العادي، إلا انه غير صحيح في هذا المثال لأن الاستنتاج يعني ان التفاح هو نفسه البرتقال. لا يستطيع الجبر العادي رغم فوائده في العديد من العمليات الذهنية التعامل مع جميع جوانب الفكر. والمشكلة الأساسية في ذلك هي مشكلة اللغة. المشكلة بالمناقشة أعلاه هي في إشارة التساوي (=) الرياضية التي استخدمت لتمثيل: " ان التفاح فواكه " . فبالرغم من ان التفاح حقيقة هو فاكهة، إلا انه من الخطأ ان يقال بان التفاح يساوي فواكه وذلك لأن التفاح هو جزء من فئة عامة هي فئة جميع انواع الفواكه المختلفة. تبرز أهمية النظام الرياضي الذي وضعه بول والذي يمثل جبر الفئات بتغطية بعض المسائل التي يعجز فيها الجبر العادي عن التوصيف التام للمسألة ويستخدم في توصيف عمل الدارات المنطقية ويعدّ بقوانينه الأساسية أداة لتبسيط التوابع المنطقية.

٤-١-١ قوانين بول

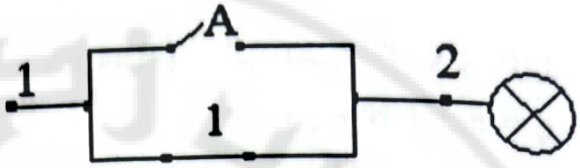
يوجد لدى جبر بول مجموعة من القوانين الناظمة التي تستخدم في أثناء كتابة أي عبارة منطقية واختصار العبارات المركبة منها. تأتي أهمية هذه القوانين أيضاً من ان صحة كثير من العبارات المنطقية ستكون بديهية، أما بعض العبارات الأخرى قد لا تكون بنفس الوضوح وبالتالي قوانين بول تساعد في اختبار صحتها. يقال عند تمثيل العبارات المنطقية، بان العبارة صحيحة أو محققة إذا كانت قيمتها واحداً أما إذا كانت عكس ذلك فيكون لها القيمة صفر. سيتم التعلمل مع الصفر والواحد المستخدمان بجبر بول باعتبارهما ثوابت منطقية (True or False) وليس باعتبارهما أرقاما ثنائية رغم انهما ينتميان للنظام الثنائي. يمكن من أجل عبارة منطقية بمتحول منطقي واحد A صياغة ثمانية قوانين منطقية مختلفة، وفق ما هو مبين في الشكل ٤-١ وتمثيل هذه العمليات على شكل دارات كهربائية باستخدام المفاتيح الذي يمر كل منها بحالتين، مثله مثل المتحول المنطقي. فإذا انطلقنا من ان العملية المنطقية AND التي تعني "و" تقابل في الدارات الكهربائية الوصل التسلسلي وان العملية المنطقية OR

التي تعني "أو" تقابل الوصل التفرعي. وان المفتاح الذي يكـون موصـولاً سـيكون متممه مفتوحاً، يمكن بوساطة الدارات الكهربائية التعبير عن أي عبارة منطقية.

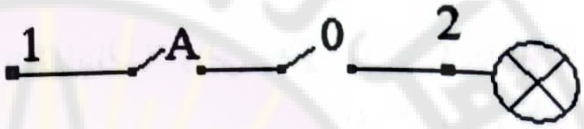
1) A + 0 = A



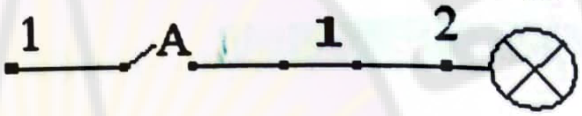
2) A + 1 = 1



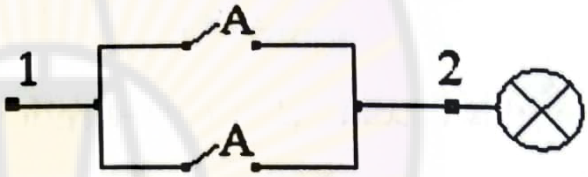
3) A . 0 = 0



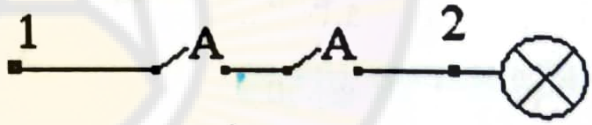
4) A . 1 = A



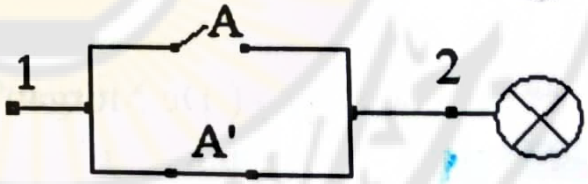
5) A + A = A



6) A . A = A



7) A + A' = 1



8) A . A' = 0



شكل ٤-١: الرمز الكهربائي لكل من قوانين بول على تابع منطقي بمتحول واحد

يمثل كل متحول منطقي مفتاح كهربائي يعبّر عنه بحالة مفتوح ويقابل القيمة " خطأ " ولا يمرر التيار وإما بحالة موصول ويقابل القيمة " صح " ويمرر التيار. تعطي لمبـة الإشارة الموجودة على خرج الدارة الكهربائية النتيجة النهائية.

يوجد بعض القوانين الأخرى التي تنطبق على توابع منطقية بأكثر من متحول. فمن أجل ثلاثة متحولات منطقية a, b, c يوجد ست مجموعات من القوانين الأخرى:

١ – قوانين التبديل (Commutative Laws) : والي تعني $\forall a, b \in B$ فان

$$a + b = b + a$$
$$a \cdot b = b \cdot a$$

٢ – قوانين التجميع (Associative Laws): والي تعني $\forall a, b, c \in B$ فان

$$(a + b) + c = a + (b + c)$$
$$(a \cdot b) \cdot c = a \cdot (b \cdot c)$$

٣ – قوانين التوزيع (Distributive Laws): والي تعني $\forall a, b, c \in B$ فان

$$(a + b) \cdot c = a \cdot c + b \cdot c$$
$$a \cdot c + b = (a + b) \cdot (c + b)$$

٤– قوانين الدمج (التبسيط):

$$a.b + a.b' = a$$
$$(a + b).(a + b') = a$$

٥ – قوانين الامتصاص (Absorption Laws) : ولها شكلان، هما:

الأول– مباشر ويقتضي ان:

$$a + a.b = a \quad \text{و} \quad a \cdot (a + b) = a$$

الثاني– عكسي ويقتضي ان:

$$a' \cdot b + a = ab' + b = a + b$$
$$a'b' + b = a' + b$$

٦ – قوانين ديمورغان (De Morgan's Laws) وهي ثلاثة:

$$(a + b)' = a' \cdot b' ;$$
$$(a \cdot b)' = a' + b' ; \quad a'' = a$$

٤–١–٢ تطبيقات جبر بول

يمكن تعميم بعض القوانين المبينة أعلاه مثل قوانين ديمورغان الأول والثاني على أكثر من متحولين، حيث:

$$(a + b + c + \ldots)' = a' \cdot b' \cdot c' \cdot \ldots$$
$$(a \cdot b \cdot c \cdot \ldots)' = a' + b' + c' + \ldots$$

ويمكن استخدام هذه القوانيين بالعديد من التطبيقات مثل اختصار العبارات المنطقية وإثبات صحة بعض القوانيين أو إيجاد متمم عبارة منطقية وغيرها من التطبيقات.

٤-١-٢-١ اختصار العبارات المنطقية

تسمى الطريقة التي تستخدم بها قوانين بول في تبسيط عبارة منطقية مركبة إلى عبارة أبسط بالطريقة الجبرية.

مثال: يمكن اختصار تابع منطقي بثلاثة متحولات من الشكل:

$$F(a, b, c) = a'b'c' + a'b'c + a'bc + ab'c' + ab'c + abc$$

وفق الطريقة الجبرية باستخدام قوانين بول إلى الشكل:

$$F(a, b, c) = a'b'c' + a'b'c + a'bc + ab'c' + ab'c + abc =$$
$$a'b'(c' + c) + bc(a' + a) + ab'(c' + c) =$$
$$a'b' + bc + ab' = b'(a' + a) + bc = b' + bc$$

٤-١-٢-٢ إثبات صحة العبارات المنطقية

يمكن إثبات صحة أي عبارة منطقية (بما في ذلك بعض قوانين بول) باستخدام الطريقة الجبرية أو بوساطة طريقة المفاتيح الكهربائية.

مثال: حول استخدام كلاً من الطريقة الجبرية وطريقة المفاتيح الكهربائية في البرهان على صحة أحد قوانيين بول، وليكن قانون التوزيع:

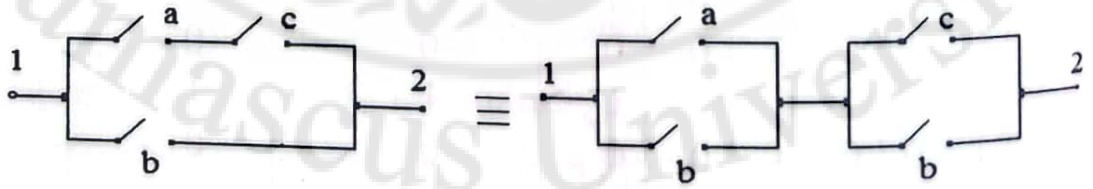
$$a \cdot c + b = (a + b)(c + b)$$

أولاً- باستخدام الطريقة الجبرية:

يلاحظ ان: $(a + b) \cdot (c + b) = a.c + a.b + b.c + b.b =$

$a.c + b.(a + c + 1) = a.c + b$ وهو المطلوب

ثانياً- باستخدام طريقة المفاتيح: يمكن تمثيل قانون التوزيع بمخطط، وفق ما هو مبين في الشكل ٤-٢.



$a \cdot c + b$ = $(a + b) \cdot (c + b)$

شكل٤-٢: طريقة المفاتيح الكهربائية بإثبات صحة قانون التوزيع

يتضح من الشكل أعلاه، ان التيار سيمر بين النقطتين 1 و 2 في الجزء الأيسر من الدارة إذا كان المفتاح b بمفرده مغلقاً أو إذا كان المفتاحان a و c مغلقين معاً (أو إذا كانت جميع المفاتيح مغلقة). يمر التيار بالجزء الأيمن من الدارة بين النقطتين 1 و 2 بنفس الحالات المبينة بالجزء الأيسر، مما يدل على ان الشكلين ينجزان نفس التابع. وبالتالي الجزء الأيسر للعبارة يساوي الجزء الأيمن، وهذا ما يثبت صحة القانون. توجد طرائق أخرى لإثبات صحة العبارات المنطقية، منها:

أولاً: طريقة استخدام جدول الحقيقة

يتم وفق هذه الطريقة وضع جدول الحقيقة للمتحولات أو الحدود لكل من طرفي العبارة وبعد ذلك تتم المقارنة بين طرفي المعادلة من خلال المقارنة بجدول الحقيقة لجميع الحالات الممكنة للمتغيرات التي تعتمد عليها المعادلة.

مثال: اثبت باستخدام طريقة جدول الحقيقة صحة عبارة منطقية من الشكل:

$$ab + bc + a'c = ab + a'c$$

الجواب: بأخذ الحالات الممكنة لمختلف المتحولات والحدود في العلاقة المعطاة بطرفيها الأيمن والأيسر ووضعها بجدول للحقيقة كالجدول المبين أدناه، حيث يلاحظ من آخر عامودين ان طرفي العبارة المعطاة متساويان فعلاً من أجل جميع القيم الممكنة للمتحولات a, b, c وهذا يدل على ان وجود الحد bc بالطرف الأيمن من عدمه لا يؤثر على قيمة العبارة.

Index	a	b	c	ab	bc	a'c	ab + bc + a'c	ab + a'c
0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	1	1	1
2	0	1	0	0	0	0	0	0
3	0	1	1	0	1	1	1	1
4	1	0	0	0	0	0	0	0
5	1	0	1	0	0	0	0	0
6	1	1	0	1	0	0	1	1
7	1	1	1	1	1	0	1	1

ثانياً: طريقة تحويل إحدى طرفي العبارة المنطقية إلى الشكل المماثل للطرف الآخر.

تتم بهذه الحالة معالجة أحد الطرفين للعبارة بتطبيق القوانين المختلفة لبول وتحويلها إلى نفس صورة الطرف الآخر لتتم عملية المقارنة.

مثال: اثبت صحة العبارة المنطقية التالية:

$$a + b'c' + bd + c'd = (a + b' + d).(a + b + c')$$

لإثبات صحة مثل هذه العبارة، يتم تحويل صورة الطرف الأيمن (جداء مجاميع) إلى نفس صورة الطرف الأيسر. بفك الأقواس ستصبح العبارة من الشكل:

$$(a + b' + d).(a + b + c') = (aa + ab + ac' + ab' + bb' + b'c' + ad + bd + c'd).(a + b + c) =$$

$$(a + ac' + ab' + b'c' + ad + bd + c'd) =$$

$$(a + ac' + ab' + b'c' + bd + c'd)$$

$$= a + b'c' + bd + c'd$$ وهذا يؤكد التساوي

ثالثاً: طريقة إجراء عملية ما على طرفي المعادلة شرط ان يكون مسموحاً عكسها.

يوجد عمليات غير مسموح بها في المنطق، مثل ضرب طرفي العبارة بعبارة أخرى، لأن عملية القسمة المعاكسة للضرب غير مسموح بها في الجبر البولياني. كذلك لا يجوز إضافة عبارة ما إلى كلا الطرفين، لأن عملية الطرح غير مسموح بها أيضاً. فمثلاً، إذا كان $y + x = y + z$ ، فهذا لا يعني ان $x = z$ ، لأنه لا يمكن اعتبار $y - y + x = y - y + z$ لعدم وجود مثل عملية الطرح في المنطق. ويمكن التأكد من ذلك، إذا تم فرض ان $x = 1$ و $z = 0$. ينتج بالتعويض بالمعادلة السابقة بان: $y + 1 = y + 0$ وهذا غير صحيح رغم انه لو فرضت ان قيمة $y = 1$ ، فان المعادلة تصبح محققة، لأن: $1 + 1 = 1 + 0$.

من العمليات الممكن إجرائها على طرفي المعادلة هي، أخذ المتمم لكلا الطرفين

رابعاً: طريقة استخدام الازدواجية Duality

انطلاقاً من قوانين دي مورغان : $(a . b)' + a' + b'$ و $(a + b)' = a' . b'$

يلاحظ ان الاختلاف بين الشكلين هو ان عمليات (+) في الطرف الأيسر

يقابلها عمليات (.) في الطرف الأيمن وبالعكس، وان المتحول بشكله المباشر في طرف للعلاقة يصبح بالشكل المتمم في الطرف الأخر، وهذا يعني رقمياً استبدال "0" بـ "1" . إذاً عملية الحصول على الازدواجية لعبارة منطقية تتم على مرحلتين: الأولى ـ إيجاد متمم العبارة المعطاة.

الثانية ـ استبدال المتحولات بمتمماتها انطلاقاً من ان a" = a

مثال: اوجد الشكل الازدواجي Dual Form للعبارة التالية:

$(ab' + cd')(mn' + vw) + x'(1 + y)$ ، حيث كافة المتحولات منطقية.

الجواب: يتم بالخطوة الأولى إيجاد متمم العبارة المعطاة وفق الشكل التالي:

$$[(ab' + cd')(mn' + vw) + x'(1 + y)\ ]^D =$$
$$(\ (a' + b).(c' + d) + (m' + n).(v' + w').x\ + 0.y')$$

وبالخطوة الثانية، يتم استبدال المتحولات بمتمماتها

$$(a + b').(c + d') + (m + n').(v + w).x' + 0.y$$

وهكذا يمكن الاستنتاج بان لكل من القوانين البوليانية شكلين، وانه إذا ثبتت صحة قانون ما فان الشكل الازدواجي لهذا القانون يكون صحيحاً أيضاً.

خامساً: طريقة استخدام قانون الإفاضة

من قوانين جبر بول المستخدمة في تبسيط العبارات المنطقية، يوجد قانون الحد الفائض. فمثلاً، العبارة المنطقية $ab + bc + a'c$ مساوية (كما سبق وأشرنا لذلك عند استخدام طريقة جدول الحقيقة) للعبارة $ab + a'c$. ولإثبات صحة ذلك، يمكن الانطلاق من الطرف $ab + a'c$ ، فنجد ان:

$$ab + a'c = ab(c + c') + a'c(b + b') = abc + abc' + a'bc + a'b'c =$$
$$bc(a + a') + abc' + a'b'c = bc + abc' + a'b'c = b(c + ac') + a'b'c$$
$$= b(a + c) + a'b'c = ab + bc + a'b'c = ab + c(b + a'b')$$
$$= ab + \quad c(a' + b) = ab + a'c + bc$$

ولإيجاد الحدود الفائضة في عبارة منطقية يتم بداية البحث عن المتحول الموجود في حدين مختلفين من العبارة بشرط ان يكون المتحول في أحد هذين الحدين بصورته العادية وفي الحد الثاني بصورة المتمم، مثل الحد bc في المثال أعلاه. إذ ان

المتحول الأول b من هذا الحد موجود في الحد الأول للعبارة ab مع المتحول a بشكله المباشر والمتحول الثاني c من نفس الحد موجود في الحد الثاني للعبارة a'c مع المتحول a'. عند تحقيق مثل هذا الشرط يكون الحد الفائض لهذين الحدين مكوناً من المتحولات المتبقية في كلا الحدين، أي bc. بالطبع الحد الفائض يمكن حذفه من العبارة المنطقية، ويمكن إضافة لعبارة ما ان لم يكن موجوداً بقصد الاختصار أو التوضيح. يلاحظ في العبارة : $(a + b).(a' + c).(b + c)$ بصورة جداء مجاميع وجود حد فائض $(b + c)$، حيث $(a + b).(a' + c) = (a + b).(a' + c).(b + c)$ وهو الشكل الثاني للقانون. بمعنى آخر في الشكل الأول لقانون الإفاضة يتم الحصول على الحد الفائض لحدين ما في العبارة التالية بربط المتحولات المتبقية منها بعملية "و" ، أي ضرب، بينما في الشكل الثاني للقانون يتم ربط المتحولات المتبقية مع بعضها بعملية " أو" أو جمع للحصول على الحد الفائض.

٤-٢ المنطق الرقمي

يتعامل المنطق الرقمي (الثنائي) مع متحولات يأخذ كل منها واحدة من قيمتين متقطعتين فقط هما: 0 و 1 ومع ثلاث عمليات منطقية أساسية هي:

- الجمع المنطقي ويرمز له بالرمز U أو (+) ويعني " أو" OR.
- الضرب المنطقي ويرمز له بالرمز ∩ أو (.) ويعني " و" AND.
- النفي المنطقي ويرمز له بخط (¯) أو بفاصلة فوق المتحول (حيث نفي المتحول x هو x') وهو يمثل المتمم للواحد لقيمة المتحول ويعني النفي " لا" NOT.

تشبه العمليات الأساسية بالمنطق الرقمي بعض العمليات بالحساب الثنائي، حيث العمليتان AND و OR تشبه عمليتي الضرب والجمع، ورغم ذلك يجب التفريق بينهما، لأن المتحول الحسابي يمكن ان يدل على عدد مؤلف من عدة أرقام في حين ان المتحول المنطقي لا يدل إلا على واحدة من قيمتين هما الواحد التي تعني صح والصفر التي تعني خطأ. لذلك 1 + 1 مثلاً، تساوي وفق الحساب الثنائي (2) وتكتب

بالشكل الثنائي $_2(10)$، في حين يكتب الناتج وفق المنطق بالشكل (1) ويقرأ " صـح".
يمكن من أجل متحولين منطقيين A و B تشكيل $2^2 = 4$ من التراكيب المختلفة.
يظهر الجدول ٤-١ هذه التراكيب والعمليات المنطقية الثلاثة الأساسية على متحولين.

المتحول A	المتحول B	الجمع A + B OR	الضرب A.B AND	النفي لـ A أي A' NOT
0	0	0	0	1
1	0	1	0	0
0	1	1	0	1
1	1	1	1	0

جدول ٤-١: التوابع المنطقية الأساسية

تنفذ العمليات المنطقية عند ورودها بعبارة منطقية وفق أولويات Priority محددة هي NOT أولا و AND ثانياً و OR ثالثاً. وفي حال ورود أكثر من عملية بنفس الأولوية وراء بعضها البعض تكون الأفضلية المباشرة بتنفيذ العمليات ابتداء من اليسار باتجاه اليمين. وفي حال وجود أقواس فان الأفضلية أو الأسبقية Precedence تكون بتنفيذ ما هو ضمن الأقواس أولا بالترتيب المشار إليه أعلاه.

٤-٢-١ البوابات المنطقية الأساسية

تمثل البوابات المنطقية Logic gates العناصر المنطقية الأساسية في الدارات المنطقية. يعرف العنصر المنطقي، بانه أبسط كينونة فيزيائية تقوم بتنفيذ عملية منطقية من العمليات الثلاث الأساسية. تمثل كل بوابة منطقية دارات الكترونية تؤدي عملية محددة. تسمى البوابات المنطقية التي تقابل التوابع المنطقية الثلاثة NOT و AND و OR بالبوابات الأساسية (أو البوابات البسيطة). تتميز كل بوابة منطقية عن غيرها بأربعة جوانب تعتمد عند دراستها هي: الاسم والرمز المنطقي وجدول الحقيقة والتابع المنطقي.

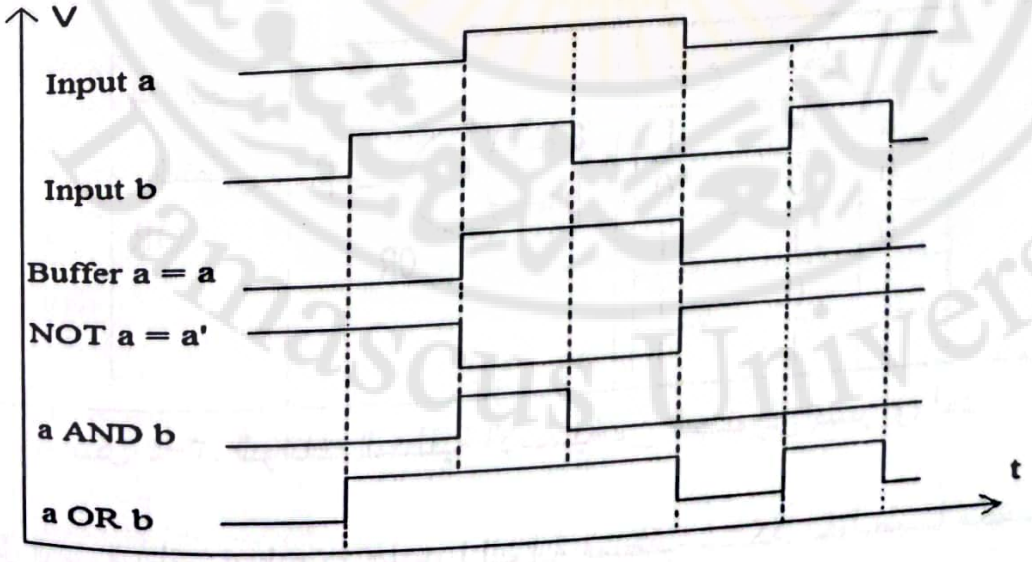
مجمل هذه الجوانب الأربعة لأول ثلاث بوابات منطقية أساسية من المستوى واحد وبوابة رابعة Buffer من المستوى صفر مبينة في الشكل ٤-٣.

المستوى	الاسم	الرمز	جدول الحقيقة	التابع
صفر	١- دارئ Buffer	a — F, BUFFER	دخل: a / خرج: F 0 / 0 1 / 1	F = a
الأول	٢-العاكس NOT	a — F, NOT	دخل: a / خرج: F 0 / 1 1 / 0	F = a'
	3-الضرب المنطقي AND	a, b — F, AND	دخل: a, b / خرج: F 0 0 / 0 0 1 / 0 1 0 / 0 1 1 / 1	F = a . b
	4- الجمع المنطقي OR	a, b — F, OR	دخل: a, b / خرج: F 0 0 / 0 0 1 / 1 1 0 / 1 1 1 / 1	F = a + b

شكل ٤-٣: البوابات المنطقية الأساسية من المستويين صفر وواحد

يمثل الرمز المنطقي Logic symbol للبوابة المنطقية شكلاً تخطيطياً مختصراً يستخدم عند رسم المخططات المنطقية. يقصد هنا بالمستوى الذي تنتمي له البوابة، عدد العمليات المنطقية التي تقوم البوابة بتنفيذها. فمثلاً، بوابة دارئ Buffer، تمثل

عملية النقل Transfer وهي تنتمي للمستوى صفر، لأنها لا تنفذ أي عملية منطقية، بل يمرر الدخل كما هو إلى الخرج من دون معالجة، بحيث الخرج يكون باستطاعة أعلى. ويرمز للبوابة بمثلث له دخل وحيد وخرج وحيد. وبأسلوب مماثل، يرمز لبوابة النفي بمثلث له مدخل وحيد ومخرج وحيد وعلى رأس المثلث دائرة (فقاعة Bubble) تشير أينما وجدت إلى عملية النفي. تنتمي هذه البوابة للمستوى الأول، لأنها تقوم بعملية منطقية واحدة، هي عكس القيمة الموجودة على الدخل. ويقصد بجدول الحقيقة Truth table، الجدول الذي يبيّن كافة التراكيب الممكنة لمتحولات الدخل والقيمة المقابلة لكل منها على الخرج. يمكن من جدول الحقيقة صياغة شكل التابع المنطقي Logic function الذي يمثل التوصيف المنطقي للبوابة. يلاحظ من الرموز المنطقية للبوابات في الشكل ٤-٣، ان بعضها مثل الدارئ والعاكس يمتاز بوجود مدخل واحد ومخرج واحد، وان للبعض الآخر مثل AND و OR يمكن ان يكون لكل منهما مدخلان أو أكثر لتنتج إشارة خرج واحدة. الشكل الممكن لقيم الدخل والخرج على شكل إشارات ثنائية المستوى لكل من الدارئ والعاكس وبوابات AND وOR (بمدخلين) مبين في الشكل ٤-٤.



شكل ٤-٤: شكل إشارات الدخل والخرج لبعض البوابات المنطقية الأساسية

٤-٢-٢ البوابات المنطقية المركبة

يمكن من دارة منطقية بمدخل واحد (ممثلاً بالمتحول الثنائي a) تشكيل تابعين على الخرج كل منهما بقيمة مغايرة عن الأخر، هما التابعين: $F_0 = a'$ و $F_1 = a$ (إذاً من أجل $n = 1$ متحول، يمكن تشكيل $2^n = 2^1 = 2$ تابع منطقي). ومن أجل متحولين مثل a و b يمكن تشكيل 2^n تابع للمتحول الأول و 2^n تابع للمتحول الثاني، أي $2^n \cdot 2^n = 2^{2n}$ تابع بالنسبة للمتحولين معاً. إذاً من أجل $n = 2$ متحول منطقي على الدخل يمكن تشكيل $2^4 = 16$ تابع منطقي مختلف على الخرج. مجمل التوابع المنطقية بالمتحولين a, b الممكن تشكيلها مبينة في الجدول ٤-٢.

Inputs	a	0	0	1	1		
	b	0	1	0	1	قيمة التابع	اسم التابع
Functions	F_0	0	0	0	0	$F_0 = 0$	Null
	F_1	0	0	0	1	$F_1 = ab$	AND
	F_2	0	0	1	0	$F_2 = ab'$	Inhibition
	F_3	0	0	1	1	$F_3 = a$	Transfer (a)
	F_4	0	1	0	0	$F_4 = a'b$	Inhibition
	F_5	0	1	0	1	$F_5 = b$	Transfer (b)
	F_6	0	1	1	0	$F_6 = ab' + a'b$	XOR
	F_7	0	1	1	1	$F_7 = a + b$	OR
	F_8	1	0	0	0	$F_8 = (a + b)'$	NOR
	F_9	1	0	0	1	$F_9 = ab + a'b'$	XNOR
	F_{10}	1	0	1	0	$F_{10} = b'$	Complement
	F_{11}	1	0	1	1	$F_{11} = a + b'$	Implication
	F_{12}	1	1	0	0	$F_{12} = a'$	Complement
	F_{13}	1	1	0	1	$F_{13} = a' + b$	Implication
	F_{14}	1	1	1	0	$F_{14} = (ab)'$	NAND
	F_{15}	1	1	1	1	$F_{15} = 1$	Identity

جدول ٤-٢: جدول الحقيقة للتوابع الستة عشر التي يمكن تعريفها باستخدام متحولين

تنتج التوابع F_0 و F_{15} المبينة في الجدول أعلاه، الثوابت 0 و 1 وتقابل التسميات:

- Null: for binary constant 0 و

- Identity: for binary constant 1 ولا يقابل أي منهما رمزاً لبوابة منطقية.

تمثل البوابات الأساسية المبينة في الجدول ٤-٣ ستة من التوابع المبينة في الجدول ٤-٢، هي التوابع : $F_1, F_3, F_5, F_7, F_{10}, F_{12}$. اثنان منها (F_1, F_7) يقابلان البوابتين AND و OR والأربعة المتبقية تقابل بوابتي Buffer و NOT (حيث ان بعض البوابات تستخدم لتنجيز تابعين مختلفين، كل منهما يؤدي وظيفة مماثلة لوظيفة الآخر، مثل عملية النقل Transfer التي تقابل تابعين منطقيين، احدهما للمتحول a والثاني للمتحول b ونفس الشيء بالنسبة للمتمم).

يقابل عملية النقل التابعين: $F_3 = a$ و $F_5 = b$.

ويقابل عملية النفي التابعين: $F_{10} = b'$ و $F_{12} = a'$

بقية التوابع المنطقية في الجدول أعلاه تقابل بوابات منطقية مركبة، بمعنى ان البوابة الواحدة تنجز أكثر من عملية منطقية. تصنف البوابات المركبة بمجموعتين، هما:

الأولى: البوابات التي تقابل توابع غير قياسية وعددها أربعة، هي:

- اثنان منها للمنع Inhibition، هما: $F_2 = ab'$ و $F_4 = a'b$ ويقصد بالمنع وجود أحد المداخل وعدم وجود المدخل الآخر.

- واثنان للتضمين Implication، هما: $F_{11} = a + b'$ و $F_{13} = a' + b$

ويقصد بالتضمين إذا وجد دخل، "إذاً" يوجد الآخر.

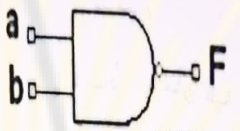
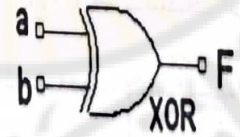
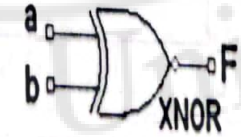
لا يستخدم لهذه التوابع المنطقية بوابات منطقية برموز محدّدة.

الثانية: البوابات التي تقابل توابع قياسية، وتمثل بقية التوابع المبينة في الجدول أعلاه وعددها أربعة، هي: F_6, F_8, F_9, F_{14}. تُمثل هذه التوابع بأربع بوابات قياسية تسمى : NAND و NOR و XOR و XNOR

يمكن بحسب عدد العمليات المنطقية التي تنفذها كل بوابة على حدة، تصنيف البوابات القياسية إلى ثلاثة مستويات هي:

- بوابات المستوى الأول وتضم البوابات OR، AND، NOT .
- بوابات المستوى الثاني وتضم البوابتين NAND و NOR.
- بوابات المستوى الثالث وتضم البوابتين XOR و XNOR.

يظهر الشكل ٤-٥ رموز بوابات المستويين الثاني والثالث (البوابات المنطقية المركبة القياسية) مع جدول الحقيقة والتابع المنطقي المقابل لكل منها على حدة.

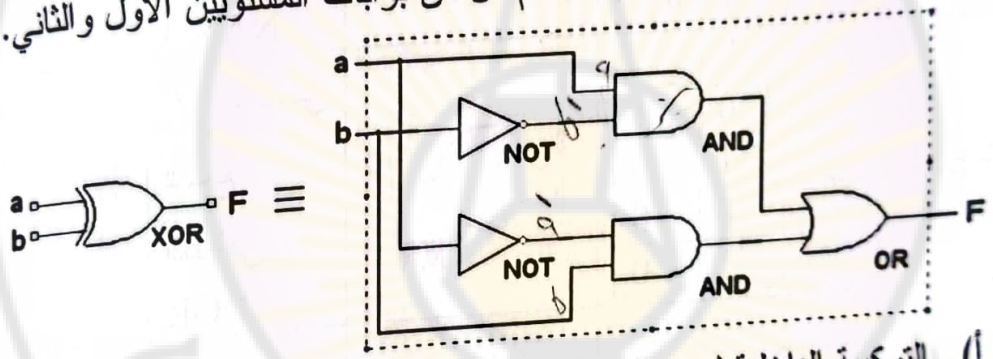
المستوى	الاسم	الرمز	جدول الحقيقة	التابع
الثاني	5- نفي الضرب المنطقي NAND		A B F 0 0 1 0 1 1 1 0 1 1 1 0	F = (a . b)'
	6- نفي الجمع المنطقي NOR		a b F 0 0 1 0 1 0 1 0 0 1 1 0	F = (a+ b)'
الثالث	7- عدم التكافؤ XOR		a b F 0 0 0 0 1 1 1 0 1 1 1 0	F = ab'+a'b
	8- التكافؤ XNOR		a b F 0 0 1 0 1 0 1 0 0 1 1 1	F = ab+a'b'

شكل ٤-٥: البوابات المنطقية المركبة (من المستويين الثاني والثالث)

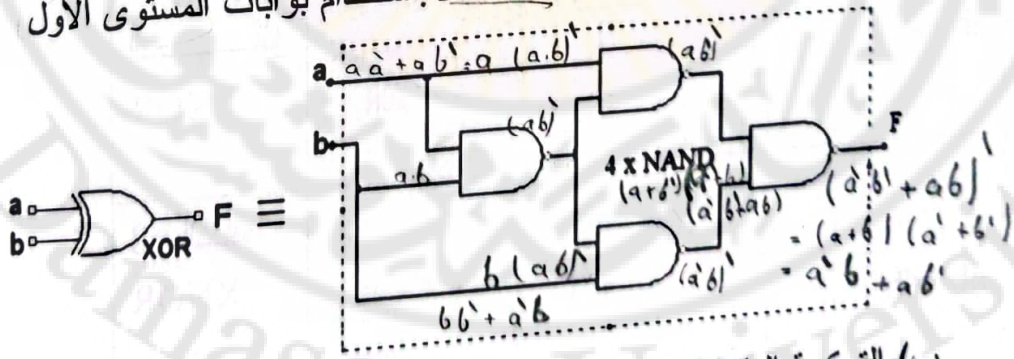
- تُعدّ بوابات المستوى الثاني من أكثر انواع البوابات انتشاراً، حيث كل منهما تقوم بعمليتين بان واحد. فالبوابة NAND (وتسمى Shafferova function) تعني

NOT AND وتقوم في المرحلة الأولى بالضرب وبنفي الناتج في المرحلة الثانية. أما البوابة NOR (وتسمى Pierceova function) تعني NOT OR وتقوم في المرحلة الأولى بالجمع المنطقي وبالنفي للناتج في المرحلة الثانية. أما بوابات المستوى الثالث فتقوم كل منهما بالعمليات المنطقية الثلاث بآن واحد. وبالتالي كل منهما تمثل دارة مركبة من بوابات المستويات الأدنى، حيث:

- البوابة XOR، تعني Exclusive-OR وهي تشبه التابع OR، إلا انها تستثني التركيبة التي يكون فيها كل من المدخلين مساوياً للواحد، بمعنى ان XOR تكون بحالة 1 إذا كان أحد المداخل (a مثلاً) فقط يساوي الواحد. يظهر الشكل ٤-٦ التركيبة الداخلية لبوابة XOR باستخدام كل من بوابات المستويين الأول والثاني.



أ) التركيبة الداخلية لرمز البوابة XOR باستخدام بوابات المستوى الأول



ب) التركيبة الداخلية لرمز البوابة XOR باستخدام بوابات المستوى الثاني

شكل ٤-٦: التركيبة الداخلية للبوابة XOR

تمثل عملية XOR الجمع بالمود 2 ويرمز لها بالرمز ⊕، حيث:

$a \oplus b = ab' + a'b$ بالتالي:

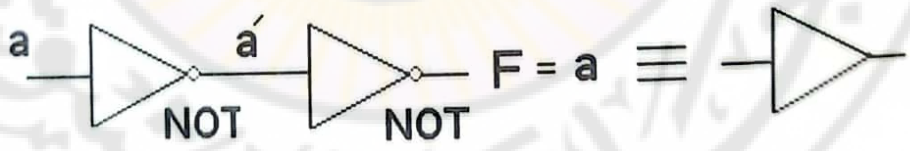
$a \oplus 0 = a$ ، $a \oplus 1 = a'$

$a \oplus a = 0$ ، $a \oplus a' = 1$

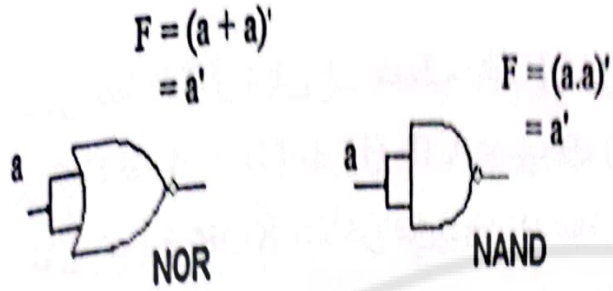
وتجدر هنا الإشارة إلى ان العملية XOR هي تبادلية وتجميعية، بمعنى ان: $(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$، أي يمكن مبادلة أي من مداخل البوابة XOR بالآخر من دون ان يؤثر ذلك على نتيجة العملية المنطقية.

- البوابة XNOR، تعني التكافؤ Equivalence وهي تمثل التابع NOT XOR، أي متمم XOR وتكتب وفق الشكل التالي: $(a \oplus b)' = ab + a'b'$ بمعنى ان خرج هذه العملية المنطقية يكون بحالة 1 عندما يكون المدخلان متماثلين سواء أكانت قيمتهما صفراً أم واحداً.

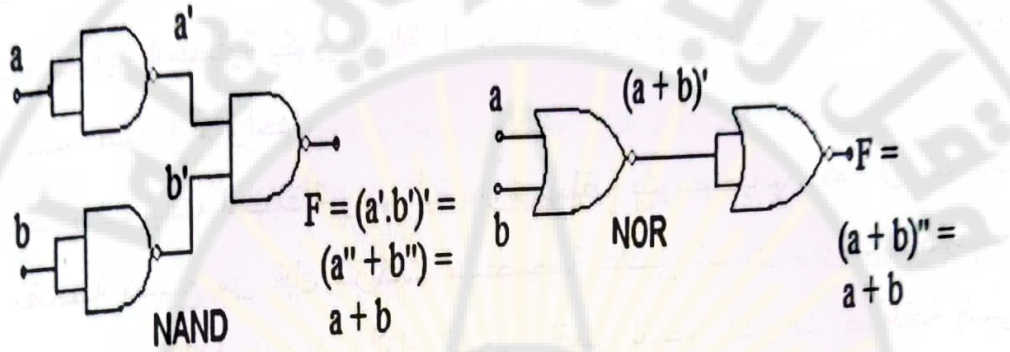
يمكن باستخدام البوابات المنطقية رسم أية دارة منطقية يتم تصميمها، كما ان البوابات المنطقية القياسية تصنع من قبل شركات متخصصة على شكل دارات متكاملة. يتم اللجوء بكثير من التطبيقات في أثناء التنجّيز الفعلي لبعض الدارات المنطقية إلى تشكيل بوابات منطقية غير متوافرة من بوابات أخرى متوافرة، مثل تشكيل بوابات المستوى الأول من بوابات المستوى الثاني التي تعّد أكثر انتشاراً بحكم إنها تمثل البوابات القياسية بالنسبة للدارات المتكاملة. بالطبع يمكن تشكيل بوابة الدارئ من بوابة العاكس بوصل بوابتين معاً وفق الشكل المبيّن أدناه.



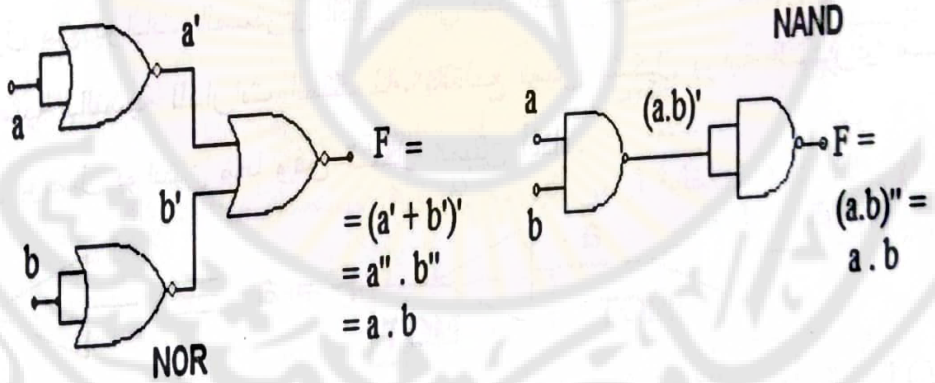
ويمكن بالإنطلاق من قوانين بول: $a + a = a \cdot a = a$ تشكيل بوابات NOT وAND وOR من بوابات NOR أو من بوابات NAND، حيث باعتماد بوابات بمدخلين أو أكثر وقصر (وصل) هذين المدخلين معاً، يمكن الحصول على بوابة بمدخل واحد تكافئ بالوظيفة البوابة NOT، كما هو مبين في الشكل ٤-٧آ. وبأسلوب مماثل يمكن تشكيل البوابة OR من NAND و NOR، كما هو مبين في الشكل ٤-٧ب، وتشكيل البوابة AND من NAND و NOR، كما هو مبين في الشكل ٤-٧ج.



أ) NOT From NAND & NOR



ب) OR From NAND & NOR



ج) AND From NAND & NOR

شكل ٤-٧: تشكيل البوابات NOT و AND و OR من البوابتين NAND و NOR.

٤-٣ التابع المنطقي

يُعرف التابع المنطقي أو التابع البولياني بانه سلسلة من العمليات المنطقية على مجموعة من المتحولات المنطقية ويعبّر عنه بشكل رياضي على انه تمثيل لـ n حالة لمتحولات منطقية على الدخل غير مرتبطة $x_1, x_2, \ldots, x_n$ على m حالة

لمتحولات منطقية مرتبطة y_m ،y_2،...، y_1 على الخرج، بمعنى ان: $y_j = f(x_i)$ من أجل $i = 1,2, \ldots, n$ و $j = 1,2, \ldots, m$. يُمثل التابع المنطقي بأشكال متعددة، نعرض فيما يلي طريقة تمثيل التابع بوساطة جدول الحقيقة، وفق ما هو مبين في الجدول ٤-٣ لتابع منطقي بثلاثة متحولات دخل.

المداخل Inputs			المخارج Outputs	الدليل Index
a	b	c	y	S
0	0	0	1	0
0	0	1	0	1
0	1	0	1	2
0	1	1	0	3
1	0	0	1	4
1	0	1	0	5
1	1	0	1	6
1	1	1	0	7

جدول ٤-٣: جدول الحقيقة لتابع منطقي بثلاث متحولات

فمن أجل تابع بـ n متحول، سيكون جدول الحقيقة المقابل له مكوناً من 2^n سطراً. بنى جدول الحقيقة، بحيث المتحول الأدنى c (LSB) يغير قيمته كل مرة من 0 إلى 1 بالتتابع، والمتحول الثاني b يتغير بعد انتهاء تغيرات المتحول الأول (أي كل مرتين) والمتحول الثالث a كل أربع مرات وهكذا (لتشكيل كافة التراكيب الممكنة في الجدول بشكل يتطابق مع الترتيب التصاعدي للترميز الثنائي للأرقام العشرية). تقابل كل تركيبة للمتحولات على الدخل سطراً واحداً من الجدول، ويدعى الرقم العشري المقابل بدليل الحالة Index state ويأخذ قيم من 0 وحتى $S = 2^n - 1$. تم في هذا المثال إظهار كافة التراكيب الممكنة على المداخل a, b, c وعلى الخرج تم في هذه المرحلة وضع قيم لـ y، بحيث وجود قيمة 1 يعني ان التابع محقق.

٤-٣-١ أشكال التابع المنطقي

يعبّر عن كل تابع منطقي بقيم تقابل حالة الإمكانية التي يعبّر عنها بالقيمة " 1" أو عدم الإمكانية بتحقيق التابع والتي يعبّر عنها بالقيمة " 0" . يمكن بالنسبة لأي تابع منطقي إظهار حالة الإمكانية بالحدوث من حالة عدم الإمكانية بأسلوبين مختلفين: مباشر وغير مباشر. فمثلاً، بالنسبة لانسان ما ان كان طويل القامة يقال عنه، انه طويل، وهو تعبير مباشر أو انه غير قصير (أي انه طويل) وهو تعبير غير مباشر.

يوصف الجدول ٤-٣ مثلاً، تابعاً منطقياً محققاً من أجل دلائل الحالة ذات الأرقام $F(a, b, c) = (0, 2, 4, 6)$، أي انه التابع الذي يظهر فيه على الخرج فقط الأعداد الزوجية من جملة الأعداد 0 وحتى 7. الصياغة الجبرية للتابع المعطى بجدول الحقيقة أعلاه يمكن ان تكون بشكلين أساسين:

الأول: حالة الإمكانية $y = F_1(a, b, c) = a'b'c' + a'bc' + ab'c' + abc'$ وقد تمت كتابة هذه الصيغة انطلاقاً من الأسس التالية:

- تكتب جميع الحدود التي تكون فيها قيمة التابع واحداً منطقياً فقط.
- يمثل كل متحول من متحولات الدخل تقابل قيمته الواحد في تركيبة ما بشكل مباشر كما هو وكل متحول تقابل قيمته الصفر يمثل بشكل منفي.
- تمثل كل متحولات الدخل عند كتابة أي تركيبة (على شكل جداء للمتحولات) سواء أكانت في الشكل المباشر أم المنفي.

يسمي هذا الشكل من التعبير عن التابع " الشكل الجامع "Sum of Product

الثاني: حالة عدم الإمكانية، حيث يمكن التعبير عن نفس التابع المنطقي المعطى في الجدول أعلاه بشكل آخر يدعى " الشكل الضارب "Product of Sum ويعني جداء مجاميع المتحولات التي تشكل التابع بشكلها المباشر أو المتمم من الشكل:

$$y = F_2(a,b,c) = (a+b+c')(a+b'+c')(a'+b+c')(a'+b'+c')$$

وقد تمت كتابة هذه الصيغة انطلاقاً من الأسس التالية:

- تكتب جميع الحدود التي تكون فيها قيمة التابع صفراً منطقياً فقط.
- يكتب كل متحول تقابل قيمته الصفر بشكل مباشر كما هو، وكل متحول تقابل قيمته الواحد يكتب بشكل منفي.
- تمثل كل متحولات الدخل عند كتابة أي تركيبة (على شكل مجموع)، بحيث كل مجموع يحوي كافة متحولات الدخل.

٤-٣-٢ كتابة التابع المنطقي باستخدام الحدود الصغرى والعظمى

يدعى كل جداء (حدّ) من جداءات التابع المنطقي المكتوب وفق الشكل الجامع P – Term من Product term وإذا كان الجداء يحوي كافة المتحولات غير المرتبطة (أي متحولات الدخل) يدعى بالحد الاصغري ويرمز له بالرمز m بدلاً من P للدلالة على انه حد أصغري minterm. يمكن كتابة التابع $F_1(a, b, c)$ الممثل في الجدول ٤-٣ وفق الشكل الجامع بشكل آخر أبسط عما هو ورد أعلاه، وفق الشكل: $F(a,b,c) = \sum(0, 2, 4, 6)$. وتسمى الصيغة الجديدة بالصيغة المبسطة للتابع، وفيها تكتب دلائل الحالة التي يكون فيها التابع محققاً وتوضع بين قوسين تحت رمز الجمع $\sum$ بالدليل m للدلالة على الحدود الأصغرية.

مثال: التابع المنطقي $F_1(a, b, c) = ab + ab'c + b$ يحتوي على ثلاثة حدود، منها حد واحد فقط هو m-Term هو الحد $m_5 = a.b'.c$ ، لأنه كما هو واضح من الصيغة المعطى بها يتكون من ثلاثة متحولات a, b, c.

بأسلوب مماثل يمكن ان يوصف الشكل الثاني للتابع المنطقي أي الشكل الضارب، بحيث من أجل تابع منطقي من الشكل $F_2(a,b,c) = b.(a + b).(a + b + c')$، يلاحظ ان التابع يحتوي على ثلاثة مجاميع كتبت مع بعضها على شكل جداء لهذه المجاميع. كل مجموع من هذه المجامع يدعى S-Term من Sum term، وإذا كان حدّ ما (المجموع) يحوي كافة المتحولات غير المرتبطة، يدعى الحد بالأعظمي ويرمز له بالرمز M بدلاً من S للدلالة على انه حد أعظمي Maxterm.

ففي المثال الوارد أعلاه يوجد حد أعظمي واحد، هو $(M_1 = a + b + c')$، لأنه يحوي كل متحولات التابع. ويمكن كتابة نفس التابع المعطى في جدول الحقيقة السابق ٤-٣ بالشكل الضارب أيضاً والتعبير عنه بالشكل المبسط، أي بكتابة دلائل الحالة التي يأخذ فيها التابع قيمة 0 ووضعها بين قوسين تحت رمز الجداء $\prod$ بالشكل $F(a, b, c) = \prod_M(1,3,5,7)$ للدلالة على الحدود الأعظمية. تمثل دلائل الحالة للشكل الضارب بالمثال أعلاه نفس دلائل الحالة المتبقية من الشكل الجامع لنفس التابع، لأن الشكلين الجامع والضارب لتابع ما متممان لبعضهما البعض، وبالتالي لا يمكن لدليل حالة ما ان يكون مشتركاً بين الشكلين بان واحد. وهكذا من أجل تابع منطقي بثلاثة متحولات يكون العدد الإجمالي للحدود الصغرى $2^3 = 8$ (وكذلك عدد الحدود العظمى)، وفق ما هو مبين في الجدول ٤-٤.

a b c	minterms	Maxterms
0 0 0	$m_0 = a'b'c'$	$M_0 = a + b + c$
0 0 1	$m_1 = a' b'c$	$M_1 = a + b + c'$
0 1 0	$m_2 = a'bc'$	$M_2 = a + b' + c$
0 1 1	$m_3 = a'b'c'$	$M_3 = a + b' + c'$
1 0 0	$m_4 = a'b\ c$	$M_4 = a' + b + c$
1 0 1	$m_5 = a\ b'c$	$M_5 = a' + b + c'$
1 1 0	$ın_6 = a\ bc'$	$M_6 = a' + b' + c$
1 1 1	$m_7 = a\ b\ c$	$M_7 = a' + b' + c'$

جدول ٤-٤: الحدود الصغرى والعظمى الممكنة لتابع منطقي بثلاثة متحولات

عندما يكتب التابع المنطقي على شكل مجاميع لـ m فقط أو مضاريب لـ M فقط يقال عن التابع انه بالصورة القياسية Standard form وهذا ما يعّبر عنه رياضياً

بالنسبة لتابع ما مثل التابع الذي يحدّد الأعداد الزوجية ضمن المجال 7 ÷ 0 الوارد في المثال أعلاه بالشكل التالي:

ان كان التابع معطى وفق الشكل الجامع يكتب: $F_1 = m_0 + m_2 + m_4 + m_6$

ولن كان التابع معطى وفق الشكل الضارب يكتب: $F_2 = M_1 . M_3. M_5. M_7$

يكفي للتعامل مع التابع المنطقي استخدام أحد الشكلين القياسيين أعلاه واللذين يسميان أيضا بالصيغ القانونية للتابع المنطقي Canonical Form، لأنه عند معرفة إحدى أشكال التابع سواء أكان بالصيغة العادية أم القياسية يمكن بسهولة استنتاج الشكل الآخر. ولذلك سيتم من الان وصاعداً التعامل مع شكل واحد فقط للتابع المنطقي هو الشكل الجامع (الأكثر استخداما) ما لم يذكر عكس ذلك.

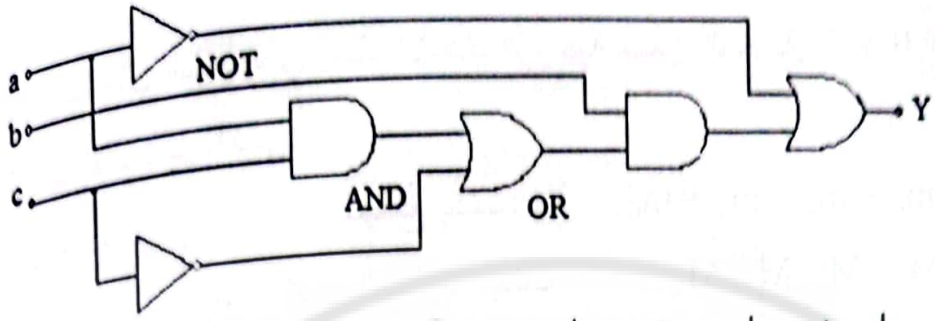
٤-٣-٣ التحويل إلى الشكل القياسي للتابع المنطقي

غالبا ما يكون التابع المنطقي بمرحلة التنجيز على شكل دارة منطقية بالصيغة المختصرة التي لا تقابل الشكل القياسي. يوجد بعض التطبيقات التي يلزم فيها التعامل مع الشكل القياسي للتابع. سيتم فيما يلي عرض مثال حول تحويل الشكل غير القياسي للتابع المنطقي إلى الشكل القياسي المقابل.

مثال: ارسم الدارة المنطقية التي تحقق تابع منطقي معطى بالشكل المبسط التالي:

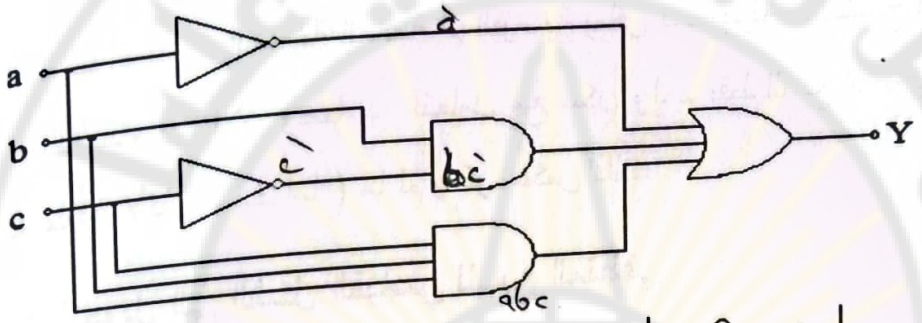
$$y = a' + b\,(c' + ac)$$

الجواب: يتضح من الشكل المعطى به التابع بانه غير قياسي لأنه ليس بصيغة الشكل الجامع ولا بصيغة الشكل الضارب. يخصص عادة لكل حد بالتابع بوابة ولكل متحول بالحد مدخل من مداخل البوابة. يتضح من شكل التابع انه مكون من ثلاثة متحولات منطقية a, b, c. باستخدام البوابات المنطقية المناسبة يمكن رسم الدارة المنطقية التي تحققه وفق ماهو مبيّن في الشكل ٤-٨آ. تحتاج الدارة لبوابتي NOT (لنفي المتحولين a, c) وبوابتي AND كل منهما بمدخلين وبوابتي OR كل منهما بمدخلين أيضا موزعة على أربعة مستويات.



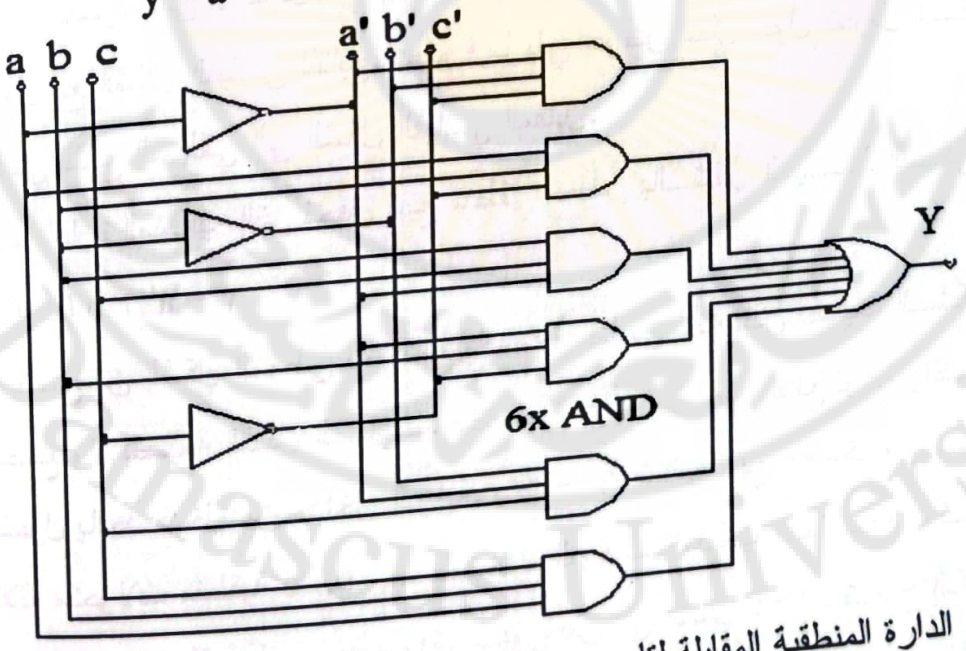
المستوى Level: | 1 | 2 | 3 | 4 |

آ) الدارة المنطقية المقابلة لتابع من الشكل $y = a' + b\,(c' + ac)$



Level: | 1 | 2 |

ب) الدارة المنطقية المقابلة لتابع من الشكل $y = a' + b\,c' + abc$



ج) الدارة المنطقية المقابلة لتابع من الشكل

$y = a'bc + a'bc' + a'b'c + a'b'c' + abc' + abc$

شكل ٤-٨: أشكال مختلفة لدارات منطقية تقابل نفس التابع المنطقي

يرسم بالمستوى (المرحلة) الأول وفق ترتيب الأسبقيات في تنفيذ العمليات المنطقية بوابة AND التي تقابل الحد ac ، وبالمستوى الثاني رسمت بوابة OR لجمع خرج المستوى السابق (الأول) مع المتحول 'c. تم بالمستوى الثالث رسم بوابة AND أخرى تقابل ضرب ناتج المستوى السابق بالمتحول b وأخيرا بالمستوى الرابع تم رسم بوابة OR لجمع خرج ما سبق مع المتحول 'a وهو الخرج النهائي للدارة الذي يعطي القيمة النهائية للتابع. لإيجاد الصورة القياسية للتابع المعطى أعلاه، يعدّل بالخطوة الأولى، الشكل العام للتابع بفك الأقواس، ليتم الحصول على الشكل الجديد التالي: $y = a' + bc' + abc$ أصبح التابع في الشكل الجامع ولكن ليس بالصيغة القياسية، لأنه ينقص بعض حدوده متحول أو أكثر من متحولات التابع الثلاثة. يضم التابع المعطى حداً أصغرياً واحداً m-Term وحدين نوع P-term. ويصبح المخطط العام للدارة المقابلة لهذا الشكل الجديد للتابع كما في الشكل ٤-٨ب وهو بالمقارنة مع الشكل ٤-٨آ مكون من مستويين فقط (تم تنقيص عدد المستويات والبوابات)، علماً ان قيمة التابع لم تتغير. يُعّد الشكل ٤-٨ب أفضل بكثير من الشكل ٤-٨آ، لاعتبار آخر يتمثل في ان انجاز تابع بمستويين يرافقه، كما سنجد لاحقا قيمة تأخير أقل لانتشار الإشارات عبر البوابات المنطقية من الدخل إلى الخرج النهائي. ولإيجاد الشكل القياسي للتابع يتم اللجوء إلى قوانين جبر بول لتحويل كل P-term إلى حد اصغري m-term باستكمال أي متحول ناقص. وبحسب الشكل المعطى به التابع، تستخدم قوانين محددة من جبر بول وفق التالي:

أولاً: إذا كان التابع معطى بالشكل الجامع، تستخدم القوا نين التالية:

$$A \cdot 1 = A \text{ ،} \qquad A + A = A \text{،} \qquad A + A' = 1$$

وهكذا يصبح الشكل $y = a' + bc' + abc$ للتابع المعطى على النحو التالي:

$$y = a' + bc' + abc = a'(b + b')(c + c') + (a + a')bc' + abc$$

$$y = a'bc + a'bc' + a'b'c + a'b'c' + abc' + abc \sim$$

$$011 + 010 + 001 + 000 + 110 + 111 \sim$$

$$m_3 + m_2 + m_1 + m_0 + m_6 + m_7 = \sum_m(0, 1, 2, 3, 6, 7)$$

يتم بالمرحلة الأخيرة من إيجاد الشكل القياسي للتابع، ترميز كل متحول ظهر بشكله المباشر بالقيمة 1 وكل متحول ظهر بشكله المنفي بالقيمة 0 . الدارة المنطقية التي تقابل هذا الشكل الأخير للتابع، مبينة في الشكل ٤-٨ج.

ثانياً: إذا كان التابع معطى في الشكل الضارب، تستخدم القوانين التالية:

$$A \cdot A = A, \qquad A \cdot A' = 0, \qquad A + 0 = A$$

مثال: اوجد الشكل القياسي للتابع المنطقي: $F(a,b,c) = (a + b) .(b + c)$

الجواب: يتضح ان التابع مكون من ثلاثة متحولات وفيه حدين من نوع S-Term.

لتحويل التابع إلى الشكل القياسي، يمكن استكمال المتحولات الناقصة في كل حد على حدة وتحويل كل S-Term إلى M-term.

$$F = (a + b) .(b + c) = (a + b + c.c').(a.a' + b + c) =$$
$$(a + b + c).(a + b + c') (a + b + c).(a' + b + c) =$$
$$M_0 . M_1 .M_4 = \prod(0,1,4)$$

يمكن إيجاد الشكل القياسي للتابع المنطقي سواء كان معطى في الشكل الضارب أم في الشكل الجامع بشكل أسرع وأسهل باستخدام جدول الحقيقة، كما في المثال التالي:

مثال: اوجد باستخدام جدول الحقيقة الشكل القياسي للتابع $y = a' + bc' + abc$ ؟

الجواب: بما ان التابع المعطى هو بثلاث متحولات ومعرّف من أجل ثلاثة حدود فقط a' أو bc' أو abc، يكون جدول الحقيقة المقابل له من الشكل المبين جانباً.

a	b	c	Y
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	
1	0	1	
1	1	0	1
1	1	1	1

لإيجاد أين يكون المتحول a في الشكل المنفي مثلاً، أي بالقيمة صفر؟ ينظر إلى الجدول، فيلاحظ ان ذلك يظهر بالأسطر الأربع الأولى. توضع قيمة 1 لـ y في هذه الأسطر للتعبير عن ان التابع معرّف من أجل هذه التراكيب. بنفس الطريقة يتابع من أجل كل من الحدين bc' و a'bc' فيلاحظ ان السطر الثالث من الأعلى المقابل للتركيبة a'bc' = 010 مثلاً محقق من أجل

الحدين a' و bc' بأن واحد (يكتب واحد لمرة واحدة). بقية قيم y غير المحدّدة ستكون صفر. والشكل القياسي للتابع يكون مجموع الحدود التي في كل منها y = 1 أي:

$$y = \sum_m(0, 1, 2, 3, 6, 7) = m_0 + m_1 + m_2 + m_3 + m_6 + m_7 =$$
$$000 + 001 + 010 + 011 + 110 + 111$$
$$= a'b'c' + a'b'c + a'bc' + a'bc + abc' + abc$$

٤-٣-٤ التحويل بين الأشكال القانونية للتوابع المنطقية

نستعرض فيما يلي من خلال مثال محدّد آلية تحويل الشكل الجامع لتابع منطقي إلى الشكل الضارب المتمم له.

مثال: ليكن معطى تابع منطقي بالشكل الجامع $F = \sum_m (1, 4, 5, 6)$، يطلب إيجاد الشكل الضارب المقابل لهذا التابع؟

الجواب: بما ان التابع معطى في الشكل الجامع يعني ان الأرقام الواردة بين قوسين تمثل حالات التابع التي تكون فيها قيمته تساوي الواحد.، بالتالي التابع المعطى سيضم الحدود التالية: $F(a,b,c) = 1 + 4 + 5 + 6 =$

$$001 + 100 + 101 + 110 = a'b'c + ab'c' + ab'c + abc'$$

لتحويل هذه الصورة إلى الشكل الضارب تتبع الخطوات التالية:

١- تؤخذ دلائل الحالة المتبقية للتابع المنطقي المعطى ويكتب كل منها على شكل جداء عادي وكأن التابع يكتب من جديد بهذه الدلائل الجديدة بصورة الشكل الجامع، مع فارق واحد هو بدلاً من F يكتب ' F كالتالي:

$$F'(a,b,c) = \sum_m(0,2,3,7) = a'b'c' + a'bc' + a'bc + abc$$

٢- يؤخذ نفي التابع الجديد مرة أخرى أي نفي ' F للحصول على F = " F وفق الشكل:

$$F''(a,b,c) = F = (a'b'c' + a'bc' + a'bc + abc)'$$

وبإصلاح العبارة الناتجة يتم الحصول على الشكل التالي:

$$F = (a + b + c).(a + b' + c).(a + b' + c').(a' + b' + c')$$ وهو المطلوب.

٤-٤ أسئلة وتمارين

١- اثبت ان بوابة NOR وفق المنطق الموجب تكافئ بوابة NAND وفق المنطق السالب وبالعكس؟

٢- بأخذ رماز ASCII لكل من الحرفين a و A مثلاً، اوجد الرمز المقابل للرماز الناتج من العمليات المنطقية التالية؟

a) a AND A b) NOT A c) a OR A

٣- ارسم المخطط المنطقي (التركيبة الداخلية) لبوابة التكافؤ (XNOR) باستخدام بوابات NOR حصراً؟

٤- باستخدام طريقتي جدول الحقيقة والمفاتيح الكهربائية، اثبت صحة قانون الامتصاص العكسي $a'b' + b = a' + b$ ؟

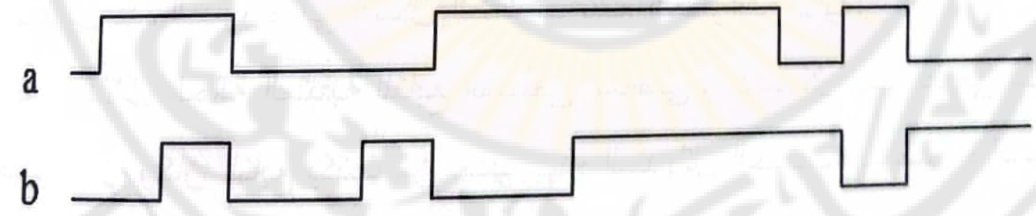
٥- باستخدام الطريقة الجبرية اختصر العبارة المنطقية التالية:

$a'b'c' + a'b'c + ab'c' + ab'c + abc' + abc$ إلى أبسط شكل ممكن؟

٦- ارسم المخطط المنطقي المقابل لكل من العبارات (التوابع) المنطقية التالية:

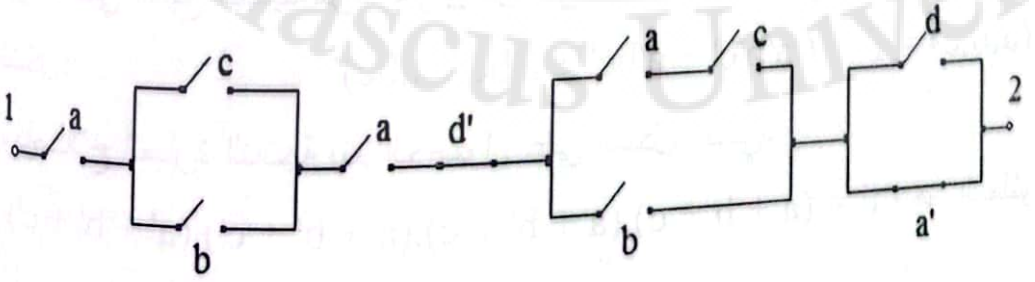
a) $F = c' + ab + a'b'c$ b) $F = ab' + c(a + b)$

٧- إذا كان كل من المتحولين المنطقيين a, b معطى وفق المخطط التالي:



a) $a \oplus b$ b) a NAND b اوجد المخطط المقابل لكل من

٨- اكتب العبارة المنطقية المقابلة للمخطط الكهربائي التالي:



من هذه الدراسة

الفصل التاسع

الدارات المنطقية التركيبية

Combinational logic circuits

تصنف الدارات المنطقية بحسب آلية عملها بواحدة من المجموعتين التاليتين:

- مجموعة الدارات المنطقية التركيبية Combinational Logic Circuits
- مجموعة الدارات المنطقية التتابعية Sequential Logic Circuits

تعرّف كل دارة منطقية تركيبية (تجميعية)، بانها جملة مكونة من مجموعة عناصر منطقية موصولة مع بعضها البعض لتحقيق تابع منطقي. لا تحتوي الدارات المنطقية التركيبية على أي عنصر تخزين (أو ذاكرة Memory)، وبالتالي تتحدّد قيم مخارج الدارات التركيبية في أي لحظة بالقيم الموجودة على مداخلها بتلك اللحظة. يمكن بناء على التعريف المبين أعلاه للدارات المنطقية التركيبية، الحكم على ان كل بوابة منطقية من البوابات التي سبق وأشرنا لها، هي دارة منطقية من النوع التركيبي. أما الدارات المنطقية التتابعية فتحتوي على عناصر ذاكرة ويقاد عملها بالساعة Clock، بالتالي نميز بالنسبة للدارات التتابعية بين كل من الدارات المتزامنة Synchronous circuits والدارات غير المتزامنة Asychronous circuits.

تختلف آلية عمل الدارات المنطقية التتابعية عن الدارات التركيبية، لأن قيمة خرج الدارة لاتعتمد على قيم المداخل فقط وانما تأخذ بعين الاعتبار الحالة السابقة للدارة.

٩- ١ تصميم الدارات المنطقية التركيبية

بعد ان تم التعريف بالمفهوم العام للدارات المنطقية من النوع التركيبي، نشير فيما يلي إلى آلية تصميمها، على ان تتم الاشارة إلى آلية تصميم الدارات المنطقية من النوع التتابعي بعد التعريف بالمفهوم العام لهذه الدارات في فصول لاحقة. يمكن تصميم أي دارة منطقية من النوع التركيبي بطريقتين أساسيتين، هما:

الأولى: باتباع خطوات محدّدة وفق الشكل التقليدي.

الثانية: بمعونة الحاسب باستخدام احدى لغات توصيف العتاديات.

بالطبع كلا الطريقتين تؤدي إلى نفس النتيجة، إلا ان الطريقة الثانية أحدث وتساعد في اختبار التصميم قبل تنجيزه. تمر عملية تصميم الدارة منطقية من النوع التركيبي بسلسلة من الخطوات التي يجب إتباعها بتسلسل محّدد ومن هذه الخطوات:

- توصيف عمل الدارة المطلوب تصميمها بتحديد متحولات الدخل ومتحولات الخرج وبناء جدول الحقيقة المقابل الذي يحدّد علاقة الخرج بالدخل.
- صياغة التابع المنطقي الذي يقابل المسألة من جدول الحقيقة الذي يتم التوصل اليه، بصيغة الإمكانية (الشكل الجامع) أو عدم الإمكانية (الشكل الضارب).
- اختصار التابع المنطقي الناتج بواحدة من طرق الاختصار المشار إليها بالفصل الخامس إلى أبسط شكل ممكن.
- رسم الدارة المنطقية وفق إحدى الأشكال الممكنة للشبكات المنطقية المقابلة بأقل عدد ممكن من المستويات.

وفيما يلي نستعرض مثالاً يبيّن مراحل التصميم أعلاه، ويصلح للتعميم عند تصميم أي دارة أخرى مماثلة.

مثال: صمم (ارسم) الدارة المنطقية التي تشير للأعداد الأولية فقط من جملة الأعداد العشرية 0 وحتى 15؟

الجواب: يتم في الخطوة الأولى توصف المسألة المطلوب معالجتها بتحديد متحولات الدخل والخرج وبناء جدول الحقيقة المقابل الذي يحدد علاقة الخرج بالدخل. يبنى جدول الحقيقة بمثل هذا المثال، انطلاقاً من ان أعلى عدد عشري معطى هو 15 وانه يقابل في النظام الثنائي التركيبة $(1111)_2$، بالتالي يلزم لتمثيل الدخل أربع متحولات. بفرض ان هذه المتحولات المنطقية الأربعة على الدخل : a, b, c, d وان رمز الخرج هو y، فيكون جدول الحقيقة المقابل، مكوناً من $2^4 = 16$ سطراً. تسند قيمة واحد لـ y مقابل كل تركيبة تعادل قيمتها عدداً أولياً من الأعداد

المحددة ضمن المجال 0 وحتى 15 وفق ماهو مبين في الجدول أدناه.

index	Inputs	output
S	a b c d	y
0	0 0 0 0	0
1	0 0 0 1	1
2	0 0 1 0	0
3	0 0 1 1	1
4	0 1 0 0	0
5	0 1 0 1	1
6	0 1 1 0	0
7	0 1 1 1	1
8	1 0 0 0	0
9	1 0 0 1	0
10	1 0 1 0	0
11	1 0 1 1	1
12	1 1 0 0	0
13	1 1 0 1	1
14	1 1 1 0	0
15	1 1 1 1	0

٢- يكتب من جدول الحقيقة الناتج، التابع المنطقي المقابل، بصيغة الشكل الجامع المبسطة بالشكل التالي: $y = \sum_m(1, 3, 5, 7, 11,13)$

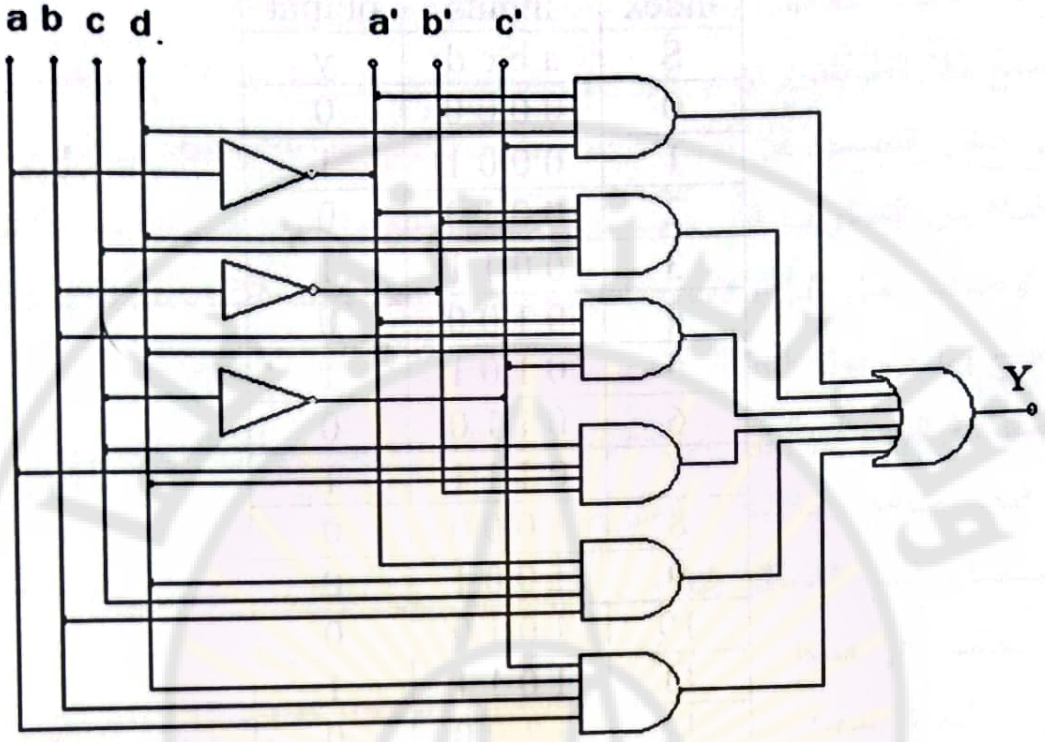
٣- تختصر الصيغة التي تم التوصل إليها للتابع إلى أبسط شكل ممكن. وبما ان التابع الناتج غير قابل للاختصار، يكتب الشكل الجبري للتابع وفق الشكل التالي:

$$y = \sum_m(1, 3, 5, 7, 11,13)$$
$$= a'b'c'd + a'b'cd + a'bc'd + a'bcd + ab'cd + abc'd$$

تمهيداً لرسمه على شكل دارة منطقية.

٤- باستخدام العناصر المنطقية المناسبة، يرسم المخطط المنطقي للدارة المطلوبة وفق إحدى الأشكال الممكنة للشبكة المنطقية المقابلة.

باستخدام التركيبة القياسية AND – OR يكون المخطط المقابل وفق ما هو مبين في الشكل ٩-١.



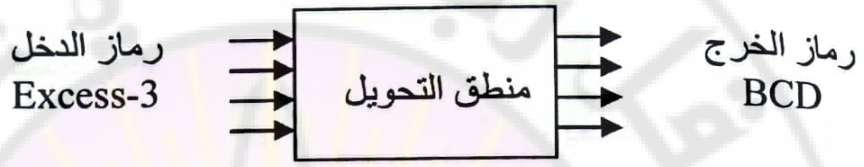
شكل ٩-١: الدارة المنطقية للتابع الذي يحدّد الأعداد الأولية من 0 وحتى 15.

٩-٢ تطبيقات متفرقة باستخدام الدارات المنطقية التركيبية

يصادف في التطبيقات العملية كثيراً من العناصر المنطقية الذي تتواجد عادة بشكل قياسي والذي كل منها يقوم بتنفيذ وظيفة محدّدة، ويصنع على شكل دارات متكاملة للاستخدامات العامة. تشكّل هذه العناصر المكونات الرقمية الأساسية Digital Components لأي نظام رقمي. سنحاول بالفقرات التالية من هذا الفصل التعرّف على بعض هذه العناصر (المكونات الرقمية) واسعة الانتشار والتي كل منها يمثل دارة منطقية من النوع التركيبي، مثل المرمز للترميز والمقارن للمقارنة وغيرها. وبهذه الفقرة سنحاول من خلال الأمثلة، إظهار بعض التطبيقات العامة للدارات المنطقية التركيبية غير القياسية، لإظهار مراحل التصميم لكل منها.

مثال: حول التحويل بين الأرمزة، وليكن إظهار آلية تحويل أي كلمة غير موزّنة بالرماز Excess-3 مثلاً إلى مايقابلها بالرماز الموزّن BCD؟

الجواب: لإظهار تركيبة الصندوق الأسود في الشكل أدناه بالاسم " منطق التحويل" ننطلق في مثل هذه الحالة من جدول الحقيقة المقابل لكل رماز على حدة. بما ان طول كل من الرمازين هو 4bits، يمكن ترميز متحولات رماز الدخل بالرموز x_0, x_1, x_2, x_3 ومتحولات رماز الخرج بالرموز Y_0, Y_1, Y_2, Y_3.



Inputs $x_3x_2x_1x_0$	Outputs $Y_3Y_2Y_1Y_0$
0 0 1 1	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 0 1 1
0 1 1 1	0 1 0 0
1 0 0 0	0 1 0 1
1 0 0 1	0 1 1 0
1 0 1 0	0 1 1 1
1 0 1 1	1 0 0 0
1 1 0 0	1 0 0 1

وبكتابة التوابع المنطقية المقابلة، بحيث كل رمز من رموز الخرج يمثل تابع لأربع متحولات للدخل، تكون التوابع المنطقية المقابلة هي على الشكل التالي:

$$Y_0 = x_3'x_2x_1'x_0' + x_3'x_2x_1x_0' + x_3x_2'x_1'x_0' + x_3x_2'x_1x_0' + x_3x_2x_1'x_0'$$
$$Y_1 = x_3'x_2x_1'x_0 + x_3'x_2x_1x_0' + x_3x_2'x_1'x_0 + x_3x_2'x_1x_0'$$
$$Y_2 = x_3'x_2x_1x_0 + x_3x_2'x_1'x_0' + x_3x_2'x_1'x_0 + x_3x_2'x_1x_0'$$
$$Y_3 = x_3x_2'x_1x_0 + x_3x_2x_1'x_0'$$

وبرسم المخططات المنطقية المقابلة لهذه التوابع بعد اختصارها يتم الحصول على المنطق اللازم لعملية التحويل أي رقم عشري من 0 وحتى 9. إضافة للمثال العام المبيّن أعلاه بدون مخطط منطقي، نورد فيما يلي تطبيقين آخرين مع المخطط المنطقي لكل منهما وهما:

أولا- مولد وفاحص النّدية

بالنظر إلى ان مفهوم مولد أو فاحص الندّية يعتمد على مفهوم التابع الفردي، نشير بداية إلى مفهوم كل من التابع الفردي والزوجي Odd & Even Function. سبق وأشرنا إلى ان عملية XOR هي تبادلية وتجميعية وهذا يعني انه يمكن تقييّم XOR بثلاثة متحولات وفق أي ترتيب، لأنه يمكن مبادلة مدخلي XOR من دون ان تتأثر نتيجة العملية. فمن أجل XOR بثلاثة مداخل a, b, c يكون الخرج كالتالي:

$$a \oplus b \oplus c = (a \oplus b)'.c + (a \oplus b).c' = (ab' + a'b)' . c + (ab + a'b') . c$$
$$= ab'c' + a'bc' + abc + a'b'c \sim 100 + 010 + 111 + 001 \sim$$
$$4, 2, 7, 1 = \sum_m (1, 2, 4, 7)$$

سيكون هذا التابع بثلاثة متحولات، كما هو واضح من الشكل أعلاه بالقيمة 1 إذا كانت قيمة أحد المتحولات فقط بحالة الواحد، أو إذا كانت قيمة جميع المتحولات الثلاثة بالقيمة 1. بمعنى انه في الحالة التي يكون فيها عدد المتحولات ثلاثة أو أكثر، يجب ان يكون عدد المتحولات التي تساوي 1 فردياً. وهذا يقود إلى التأكيد بان العملية XOR المتعددة المتحولات، تمثل تابعاً فردياً. يتضح أيضا من ان عدد الواحدات في كل دليل حالة معّرف به التابع XOR هو دوما فردي وعدد الواحدات بالحدود الصغرى المقابلة لدلائل الحالة غير المتضمنة في التابع، أي (0, 3, 5, 6) هو دوما زوجي. أي ان متمم التابع الفردي هو تابع زوجي. فمثلاً، تابع XOR

بأربع متحولات سيضم 16 حداً أصـــغرياً، نـــصفها (1, 2, 4, 7, 8, 11, 13, 14) يمتلك عدداً فردياً من الواحدات والنصف المتبقي سيمتلك عدداً زوجياً من الواحدات.

تستخدم بوابات XOR في المنظومات التي تتطلب كشف الخطأ. وتسمى الدارة التي تولد بتّ النّدية في المرسل بمولد النّدية Parity Generator (PG) وتسمى الدارة التي تفحص النّدية بالجهـــة المقابلـــة بفـــاحص النّديـــة Parity Checker (PC).

لتصميم مثل هذه الدارات، يمكن الانطلاق من مثال بـــسيط، يتمثـــل برســـم الـــدارة المناسبة لتوليد النّدية. وهنا يجب ان يتم التمييز بين النّدية الزوجية والنّديـــة الفرديـــة.

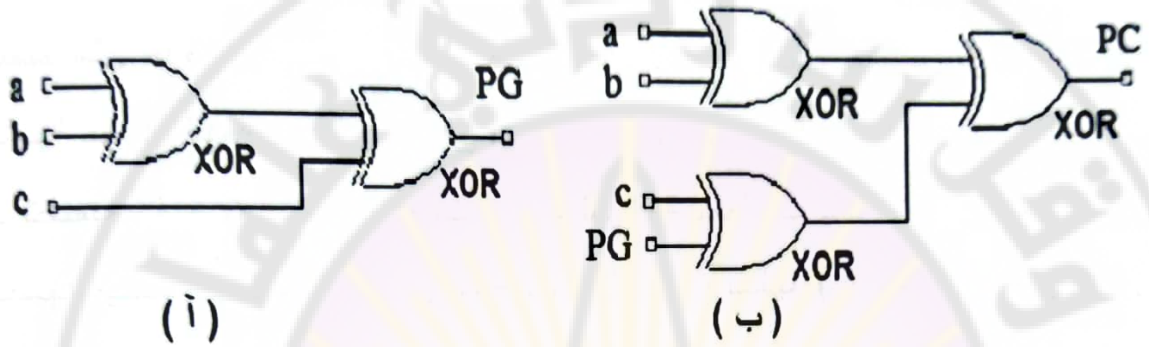
عند تصميم دارة لتوليد ندّية زوجية مثلاً، لمراقبة معطيات بطول 3bits فقط وأخرى لفحصها، يجب الانطلاق من ان المعطيات المراد إرسالها (أو تخزينها في الذاكرة) هي من الشكل abc، بالتالي المتحولات a, b, c تصبح بمثابة مداخل للدارة المطلوبة وبتّ النّدية PG هو خرجها. وباعتماد النّدية الزوجية، يصبح جدول الحقيقـــة لمولـــد النّدية المطلوب كالتالي:

بتّات الرسالة			بتّ النّدية
a	b	c	PG
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

وبما ان المعطيات على الدخل يمكن ان تكون بأي تركيبـــة مـــن 000 وحتـــى 111 يفترض وضع الخرج PG بالقيمة 0 أو 1، بحيث يكون العدد الكلي للوا حدات بكـــل تركيبة بما في ذلك قيمة PG زوجياً. وبكتابة التابع المنطقي PG(a,b,c) من جـــدول الحقيقة أعلاه يمكن الحصول على الصيغة التالية:

$$PG(a,b,c) = a'b'c + a'bc' + ab'c' + abc$$
$$= a'(b'c + bc') + a(b'c' + ab)c = a \oplus b \oplus c$$

يسمى هذا التابع الناتج أيضا بتابع النّدية الفردية، لأن قيمة التابع PG هي دوما 1 عند كل حد تتضمن قيمته الثنائية عدداً فردياً من الواحدات. ويكون المخطط المنطقي المقابل لمثل هذا التابع بوصفه مولد ندية زوجية، وفق ما هو مبين في الشكل ٩-٢آ.



شكل ٩-٢: المخطط المنطقي لمولد وفاحص النّدية

يرسل بتّ الندية المولد بالمنبع Source (سواء أكان زوجياً أم فردياً) مع الرسالة الأساسية إلى الهدف Destination. الرسالة التي يتم استلامها بطول 4 bits تدخل إلى دارة فاحص النّدية للكشف عن وقوع أي خطأ. وبالنظر إلى ان المعطيات مرسلة بنّدية زوجية، سيكون خرج فاحص النّدية PC بحالة الواحد في حال وقوع عدد فردي من الأخطاء، لأن عدد الواحدات بهذه الحالة سيكون فردياً (والخرج سيتممه إلى زوجي). وهكذا جدول الحقيقة لفاحص النّدية سيكون مكوناً من أربعة مداخل وخرج وحيد، وفق ما هو مبين في الجدول ٩-١، حيث من أجل ثماني كلمات مرسلة سيتم فحص ثمانية كلمات أخرى جديدة (ضعف العدد – الكلمات التي يكون فيها عدد الواحدات فردياً). يمثل الجدول الجديد التابع المنطقي: $PC = a \oplus b \oplus c \oplus PG$، وهو يقابل المخطط المنطقي المبين في الشكل ٩-٢ب. يمكن جعل الدارة في الشكل ٩-٢ب لتوليد النّدية، إذا وصل المدخل PG إلى الصفر (وتصبح نفس الدارة صالحة لتوليد النّدية وفحصها).

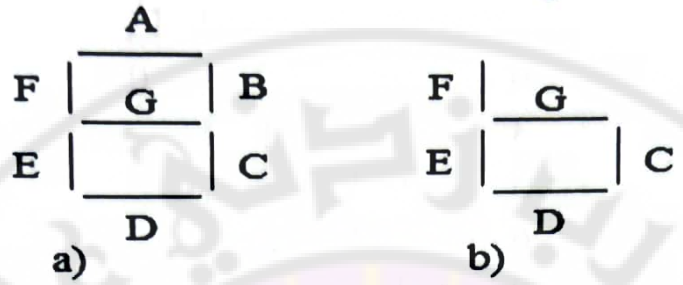
دليل الحالة	البتّات المستقبلة	فاحص النّدية
S	a b c PG	PC
0	0 0 0 0	0
3	0 0 1 1	
5	0 1 0 1	
6	0 1 1 0	
9	1 0 0 1	
10	1 0 1 0	
12	1 1 0 0	
15	1 1 1 1	
التراكيب الإضافية الجديدة بعدد فردي للوا حدات		
1	0 0 0 1	1
2	0 0 1 0	
4	0 1 0 0	
7	0 1 1 1	
8	1 0 0 0	
11	1 0 1 1	
13	1 1 0 1	
14	1 1 1 0	

جدول ٩-١: جدول الحقيقة لفاحص النّدية (الزوجية) لرسالة مكونة من 4 bits

Damascus University

ثانيا- إظهار الأرقام العشرية بوساطة السبع مقاطع المضيئة

لتصميم الدارة التي تعمل على إظهار الأرقام العشرية بوساطة العناصر ذات السبع قطع المضيئة Seven segment ذات الشكل المبين أدناه،



نعلم انه حتى يضيء الرقم 6 مثلاً، يجب ان تضيء المقاطع المبينة في الشكل (b، أي كافة المقاطع عدا المقطعين B و A (A = B = 0). لذلك أول خطوة بتصميم الدارة المطلوبة، هي رسم جدول الحقيقة الذي يوصف ماهو مطلوب، بحيث إذا وضع أحد المقاطع بالقيمة 0، يعني انه في حالة OFF مطفي. وبما انه يلزم لتمثيل الأرقام العشرية بشكل ثنائي أربعة متحولات منطقية، وان كل رقم مكون من سبعة مقاطع، فان كل مقطع على الخرج من هذه المقاطع السبعة سيمثل تابع منطقي بأربع متحولات على الدخل. بفرض ان متحولات الدخل هي: X_3, X_2, X_1, X_0. وان توابع الخرج هي: A, B, C, D, E, F, G. يكون جدول الحقيقة للتطبيق المعطى وفق ماهو مبين في الجدول ٩-٢ وفيه تم أيضاً إضافة عمود للتعبير عن كل رقم بما يقابله في النظام الستة عشري، بحيث ان الرقم 5 مثلاً يقابل القيمة 6DH. لإضاءة رقم ما، يوضع بالجدول أدناه مقابل التركيبة الثنائية التي تمثله على الدخل القيمة 1 مقابل كل مقطع من المقاطع السبعة التي يفترض ان تضاء بالرقم. فمثلاً، الرقم العشري 6 يمثل على الدخل بشكل ثنائي على أربع خانات وفق الشكل (0110). لإظهار مثل هذا الرقم على الخرج (على المقاطع السبعة)، يجب وضع التركيبة $(0111\ 1100)_2 = 7CH$ على المقاطع والتي تعني إضاءة كافة المقاطع باستثناء المقطعين A و B. يمثل بمخطط كارنوف من الجدول أعلاه، التابع المنطقي المقابل لكل مقطع من المقاطع السبعة على حده بقصد الاختصار.

No	$X_3X_2X_1X_0$	Hex	$F(X_1, X_2, X_3, X_4)$ = G, F, E, D, C, B, A
0	0 0 0 0	3F	0 0 1 1 1 1 1 1
1	0 0 0 1	06	0 0 0 0 0 1 1 0
2	0 0 1 0	5B	0 1 0 1 1 0 1 1
3	0 0 1 1	4F	0 1 0 0 1 1 1 1
4	0 1 0 0	66	0 1 1 0 0 1 1 0
5	0 1 0 1	6D	0 1 1 0 1 1 0 1
6	0 1 1 0	7C	0 1 1 1 1 1 0 0
7	0 1 1 1	07	0 0 0 0 0 1 1 1
8	1 0 0 0	7F	0 1 1 1 1 1 1 1
9	1 0 0 1	67	0 1 1 0 0 1 1 1

جدول٩-٢: جدول الحقيقة لإظهار الأرقام العشرية على سبع مقاطع مضيئة

فمثلاً، يمكن تمثيل التابع المنطقي المقابل للمقطع A، أي التـــابع $A(X_0,X_1,X_2,X_3)$ بالمخطط التالي:

$A(X_0,X_1,X_2,X_3)$ =

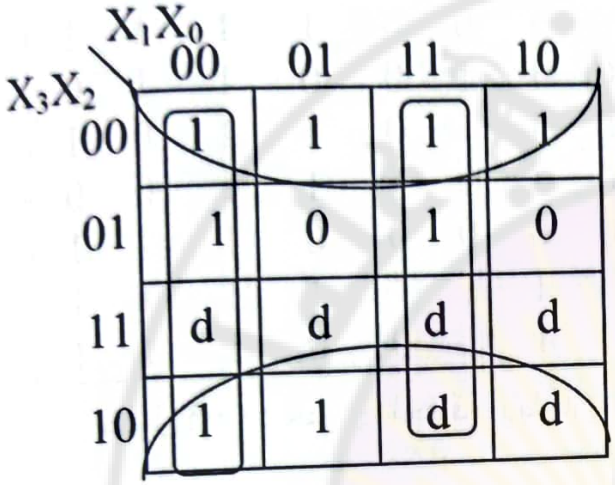
X_3X_2 \ X_1X_0	00	01	11	10
00	1	1	1	0
01	0	1	1	0
11	d	d	d	d
10	1	1	d	d

بالطبع بأربع متحولات على الدخل يمكن تمثيل 16 تركيبة مختلفة، يلزم منهـــا فقـــط عشرة تراكيب، لذلك يلاحظ بمخططات كارنوف ان الأعداد العشرية من 10 وحتـــى 15 تقابل حالات عدم اكتراث d. وباختصار التـــابع $A(X_0,X_1,X_2,X_3)$ نجـــد ان:

$$A(X_0,X_1,X_2,X_3) = A = X'_2X'_1 + X_0$$

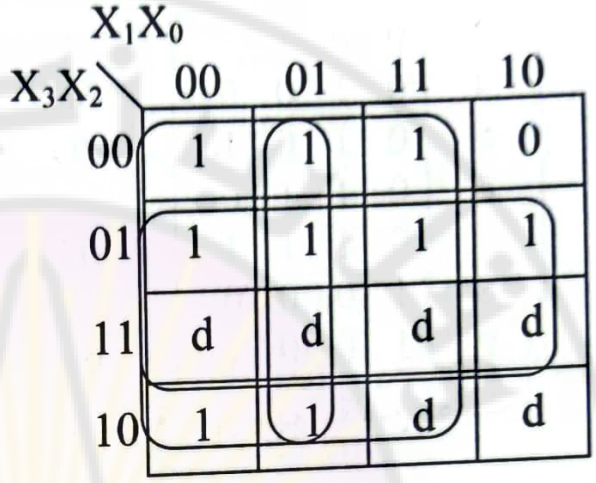
وبنفس الأسلوب يتم تمثيل التوابع الستة المتبقية بمخططات كارنوف، وفق ماهو مبين أدناه ويكتب الشكل المختصر لكل تابع على حدة، للحصول على التوابع:

$$B(X_0, X_1, X_2, X_3), \ldots, G(X_0, X_1, X_2, X_3)$$



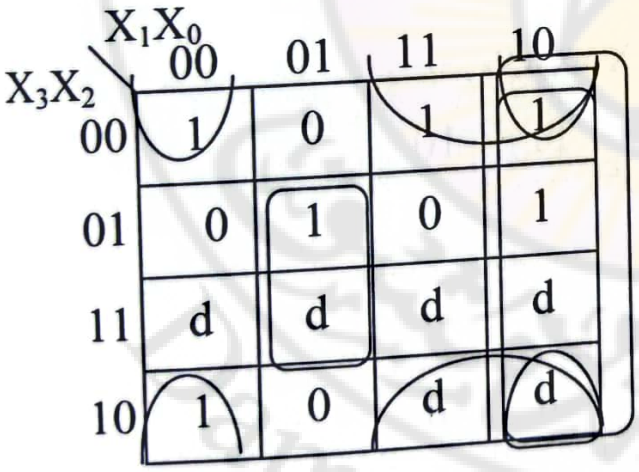
X_3X_2 \ X_1X_0	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	d	d	d	d
10	1	1	d	d

$$B = X'_2 + X_0X_1 + X'_0X'_1;$$



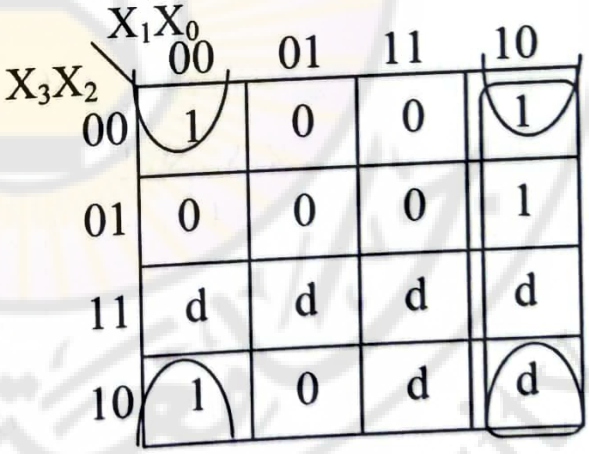
X_3X_2 \ X_1X_0	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	d	d	d	d
10	1	1	d	d

$$C = X_0 + X'_1 + X_2;$$



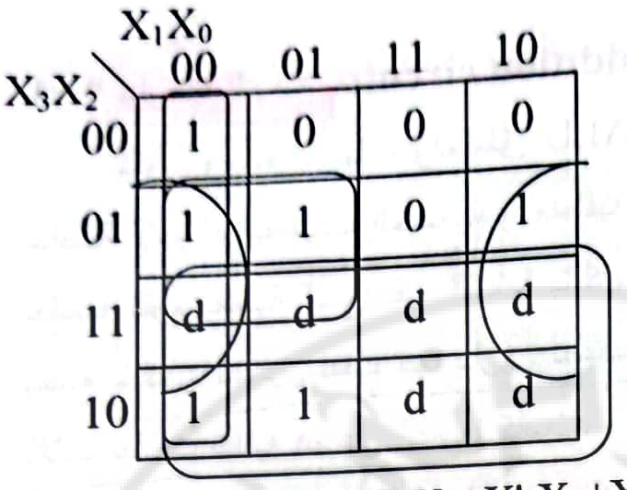
X_3X_2 \ X_1X_0	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	d	d	d	d
10	1	0	d	d

$$D = X'_0X'_2 + X'_0X_1 + X_1X'_2 + X_0X'_1X_2;$$

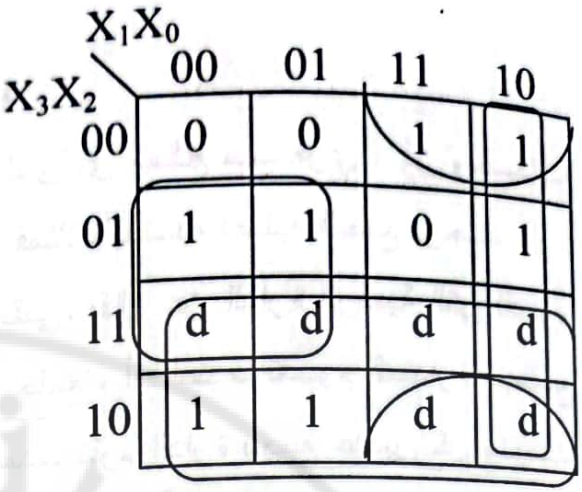


X_3X_2 \ X_1X_0	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	d	d	d	d
10	1	0	d	d

$$E = X'_0X_1 + X'_0X'_2;$$

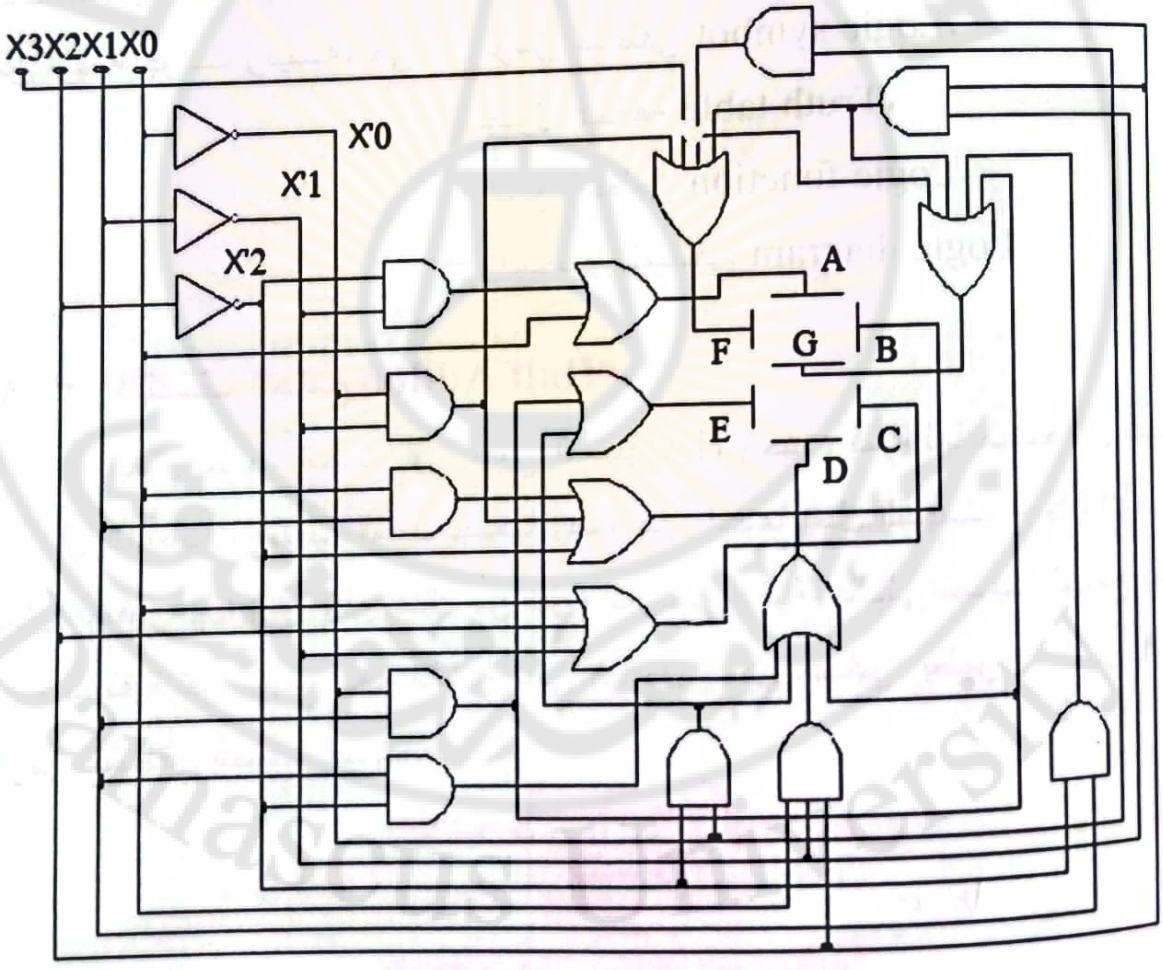


$$F=X'_0X'_1+X'_0X_2+X'_1X_2+X_3;$$



$$G = X'_0X_1 + X_1X'_2 + X'_1X_2 + X_3$$

وبرسم هذه المعادلات تكون الدارة المطلوبة، وفق ما هو مبين في الشكل ٩-٣.



الشكل٩-٣ : دارة إظهار الأرقام العشرية على عنصر إظهار من سبع مقاطع

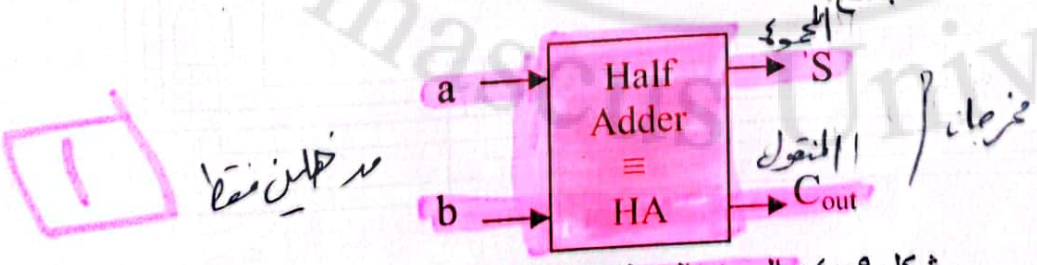
٩-٣ دارات الجمع Addition circuits

تتكون وحدة الحساب والمنطق ALU لدى أي معالج صغري من وحدات خاصة لإجراء العمليات الحسابية والمنطقية. فمثلاً، بالنسبة لعملية الجمع يوجد دارة خاصة تقوم بعملية جمع القيم الثنائية بشكل تقني. يقال عن الدارة الرقمية التي تقوم بجمع خانتين ثنائتين فقط، انها دارة " نصف جامع"، أما عندما تقوم الدارة بجمع ثلاث خانات ثنائية فتدعى "جامعاً كاملاً"، وعندما تقوم الدارة بجمع عددين كل منهما مكون من عدة خانات معاً، فتدعى "جامعاً متوازياً". وبأسلوب مماثل يصادف بالنسبة لعملية الطرح، دارات تسمى: " نصف طارح" و "طارح تام" و " طارح متوازي". سيتم فيما يلي التعريف بكل من دارات الجمع وبكل من المكونات الرقمية الأساسية، من أربعة جوانب رئيسية، هي: - الرمز المنطقي Logic symbol،

- جدول الحقيقة Truth table،
- التابع المنطقي Logic function،
- والمخطط المنطقي Logic diagram

٩-٣-١ نصف الجامع Half Adder

يمثل نصف الجامع أو الجامع البسيط HA وحدة رقمية لها مدخلان فقط لإدخال المعطيات الثنائية المراد جمعها ومخرجان احدها يمثل المجموع S والآخر يمثل المنقول C_o (ان كان هناك من نقل) للمرحلة التالية، بالتالي نصف الجامع يمثل دارة منطقية تقوم بجمع القيم الثنائية على المدخل بشكل حسابي. يظهر الشكل ٩-٤ الرمز المنطقي لنصف الجامع.



شكل ٩-٤: الرمز المنطقي لنصف الجامع

ترمز المتحولات الثنائية a, b للقيم المراد جمعها ويرمز للخرج بالمتحولين:

- S (Summation) لتمثيل المجموع.
- C_{out} (Carry Out اختصاراً C_o) لتمثيل المنقول.

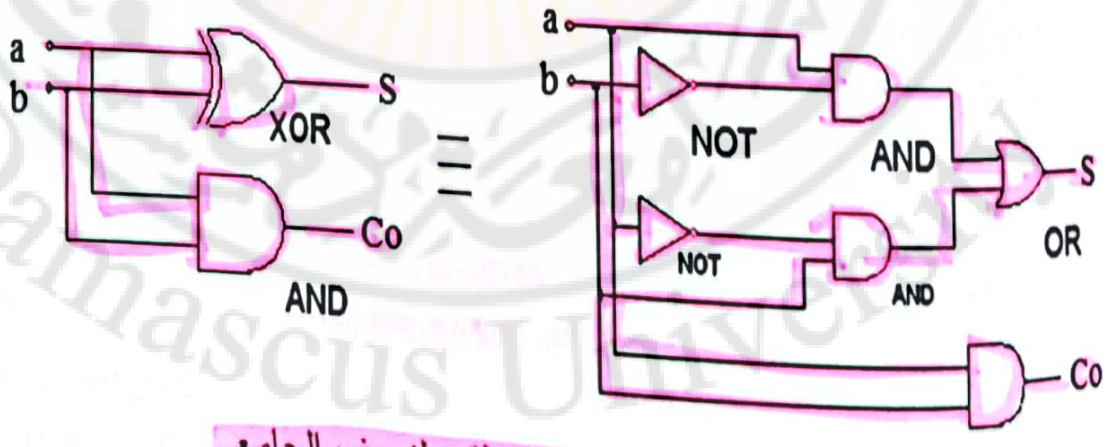
لمعرفة التركيبة الداخلية للصندوق الأسود الذي يمثل دارة نصف الجامع، تعتمد نفس الأسس المتبعة بتصميم أي دارة منطقية، ابتدأ من تحديد جدول الحقيقة الذي يظهر علاقة المخارج بالمداخل. يظهر الجدول أدناه جدول الحقيقة لنصف الجامع.

Inputs		Outputs	
a	b	S	C_o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

يتضح من جدول الحقيقة ان التوابع المنطقية المقابلة لنصف الجامع هي:

$$S = ab' + a'b = a \oplus b$$
$$C_o = ab$$

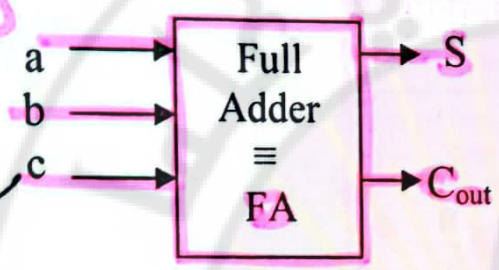
المخطط المنطقي الإجمالي الذي يقابل المعادلتين أعلاه وبالتالي بنية الصندوق الأسود مبين في الشكل ٩-5.



شكل٩-٥: المخطط المنطقي لنصف الجامع

٩-٣-٢ الجامع التام Full Adder

يمثل الجامع التام أو الكامل FA وحدة رقمية بثلاثة مداخل لإدخال المعطيات، أثنين منهما a, b لإدخال البيانات الخارجية والثالث يمثل المنقول من المرحلة السابقة Carry In (C_I اختصاراً c) الذي ينتج من عملية جمع سابقة. يرمز لمخارج الجامع التام بنفس رموز مخارج نصف الجامع والتي لها نفس المفهوم ويصبح الرمز المنطقي للجامع التام، وفق ما هو مبين في الشكل ٩-٦.



الشكل ٩-٦: الرمز المنطقي للجامع التام

Inputs			Outputs	
a	b	c	S	C_o
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

جدول الحقيقة للجامع التام

يمكن من جدول الحقيقة للجامع التام، كتابة التوابع المنطقية S وC_o وفق التالي:

$$S = a'b'c + a'bc' + ab'c' + abc$$

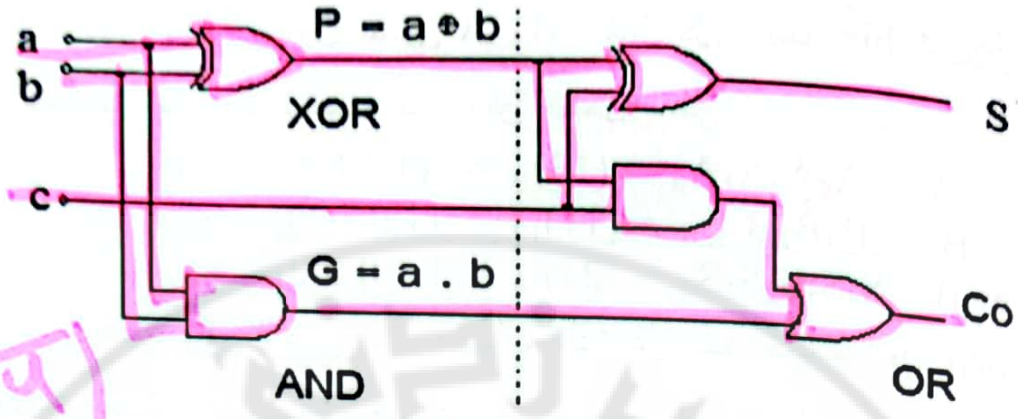
$$C_o = a'bc + ab'c + abc' + abc$$

وبالنظر لعدم إمكانية اختصار S وامكانية اختصار C_o بمتحول واحد من حدين. يعدل الشكل العام لكل من التابعين أعلاه، إلى الصيغة التالية:

$$S = c'(a'b + ab') + c(a'b' + ab) = a \oplus b \oplus c$$

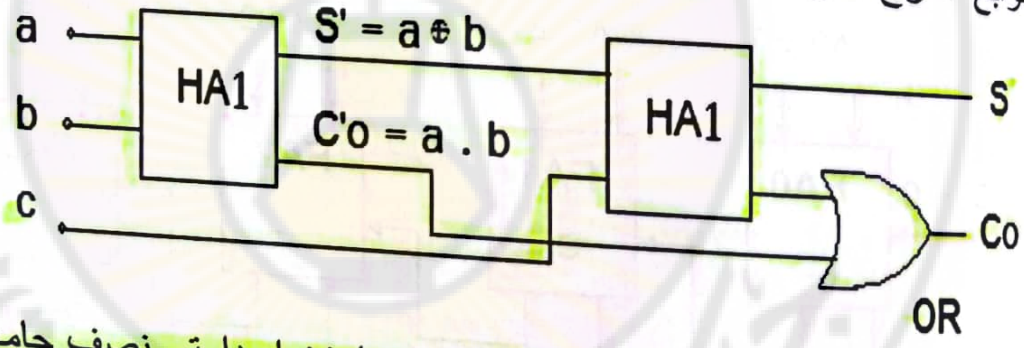
$$C_o = (a'b + ab')c + ab(c' + c) = (a \oplus b).c + ab$$

بحيث يصبح من السهل تذكرهما ويكون المخطط المنطقي المقابل، وفق ما هو مبين في الشكل ٩-7.



شكل ٩-7: المخطط المنطقي للجامع الكامل

ويمكن من نصفي جامع وباستخدام بوابة OR واحدة تشكيل جامعاً كاملاً وذلك بربط نصفي جامع مع بعضهما، وفق ما هو مبين في الشكل ٩-٨، بحيث يحافظ على قيم توابع الخرج S و Co من دون تغيير.



شكل ٩-٨: تشكيل دارة جامع كامل باستخدام دارتي نصف جامع

٩-٣-٣ الجامع الثنائي المتوازي Parallel binary adder

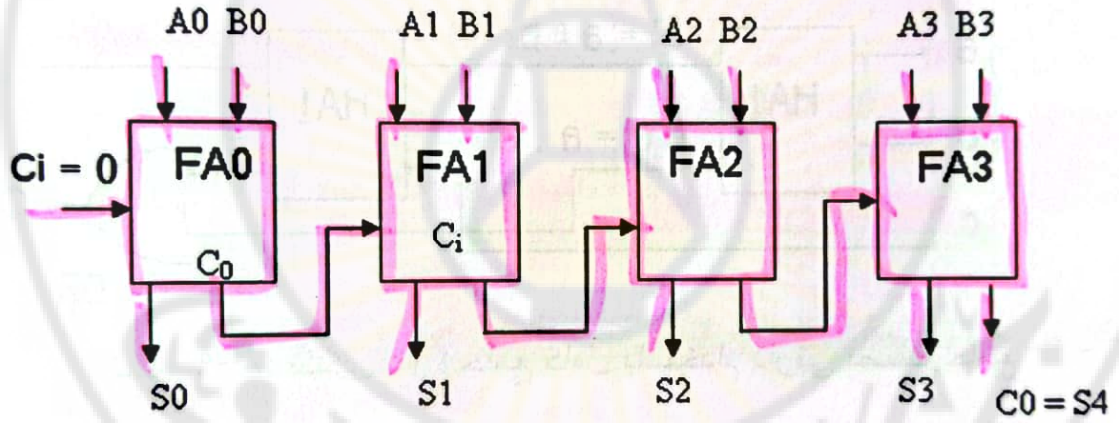
يلزم لجمع الأعداد الممثلة ثنائياً بعدد من البتّات، مجموعة من الجوامع بعدد الخانات الثنائية في الرقم الأكبر مشكلة معاً ما يسمى الجامع المتوازي. تقوم هذه الأداة بعملية الجمع بنفس الوقت لجميع البتّات المكونة للعددين مع الأخذ بعين الاعتبار إمكانية وجود منقول سابق وحصول منقول لاحق عند جمع كل بتّين فيما بينهما، كما هو موضح في المثال التالي:

مثال: ارسم الدارة المنطقية المناسبة لجمع العددين العشريين: 11 + 13

الجواب: بالإطار العام من أجل عددين A و B طول كل منهما 4 bits مثلاً، سيكون حاصل جمعهما معاً، عدداً بخمسة بتّات وفق الشكل التالي:

$$A = A_3A_2A_1A_0 = (13)_{10} = (1\ 1\ 0\ 1)_2$$
$$\underline{B = B_3B_2B_1B_0 = (11)_{10} = (1\ 0\ 1\ 1)_2}$$
$$A+B = S_4S_3S_2S_1S_0 = (24)_{10} = (1\ 1\ 0\ 0\ 0)_2$$

يكون عدد خانات الناتج النهائي أكبر بخانة واحدة عن عدد خانات العدد الأطول من بين العددين المراد جمعهما وتستخدم هذه الخانة لتمثل المنقول الأخير. بالتالي يلزم نصف جامع للخانتين الأوليتين وثلاثة جوامع كاملة لبقية الخانات، وينتج جامع على التوازي بطول4 bits، وفق ما هو مبين في الشكل ٩-٩. ولتوحيد العناصر المستخدمة في بنية الجامع المتوازي يمكن عوضاً عن نصف الجامع الأول استخدام جامع كامل بعد جعل منقول الدخل العائد له صفر $C_i = 0$.



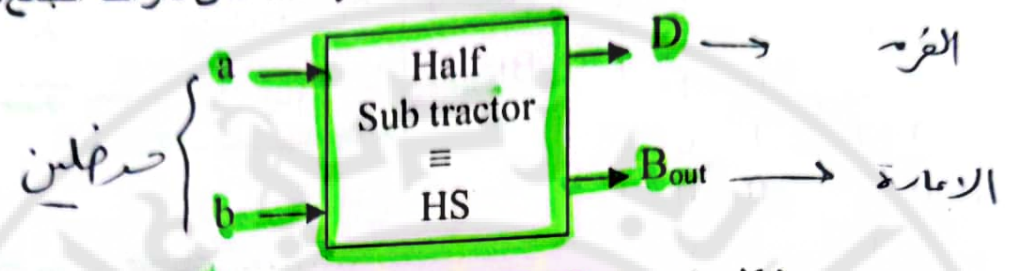
شكل ٩-٩: دارة جامع ثنائي متوازٍ بطول 4 bits

٩-٤ دارات الطرح Subtraction circuits

بنفس الآلية التي تم التعرّف بها على دارات الجمع بانواعها الثلاثة، يمكن التعرّف فيما يلي على دارات الطرح بانواعها الثلاثة أيضاً (نصف طارح و طارح تام وطارح على التوازي) مع فارق واحد بانه في عملية الطرح يستخدم عوضاً عن مفهوم المنقول Carry بالجمع، مفهوم اقتراض Borrow.

أولا: نصف الطارح Half Subtractor

١- الرمز المنطقي: يظهر الشكل ٩-١٠، الرمز المنطقي لنصف الطارح HS، حيث الرمز D (Difference) يعني الفرق، أما B_{out} (Borrow out) فتعني الإعارة وهي تأخذ مفهوماً مماثلاً للمنقول Carry لدى دارات الجمع.



شكل ٩-١٠: الرمز المنطقي لنصف الطارح

٢- جدول الحقيقة: يكون جدول الحقيقة لنصف الطارح وفق ما هو مبين في الجدول أدناه، بحيث التركيبية الثانية: 1 – 0 تقابل الناتج D = 1 والإعارة B_o 1 =، لأن العملية 1 – 0 غير ممكنة من دون استعارة واحد من الخانة الأعلى. وتصبح العملية بعد اقتراض واحد، أي $B_o = 1$ وكأنها D = 10 – 1 = 1.

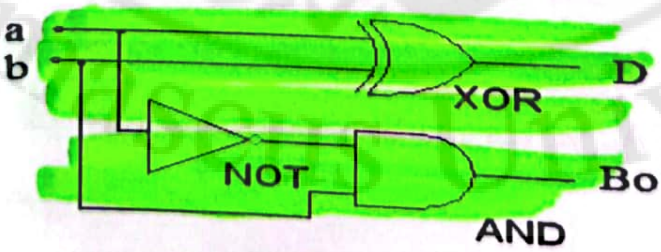
٣- التوابع المنطقية: يمكن من جدول الحقيقة لنصف الطارح الحصول على المعادلتين التاليتين:

$D = a'b + ab' = a \oplus b \quad B_o = a'b$

Inputs		Outputs	
a	b	D	B_o
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

٤- المخطط المنطقي: يظهر الشكل ٩-١١

المخطط المنطقي لنصف الطارح HS

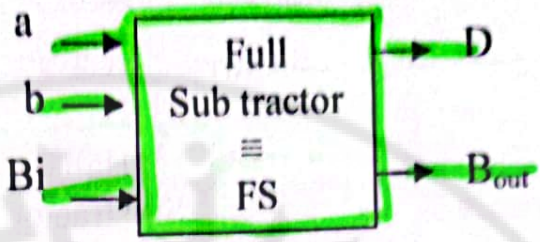


شكل ٩-١١: المخطط المنطقي لنصف الطارح HS

ثانيا: الطارح التام Full Subtractor

Inputs			Outputs	
a	b	B	D	B_o
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

١- الرمز المنطقي:



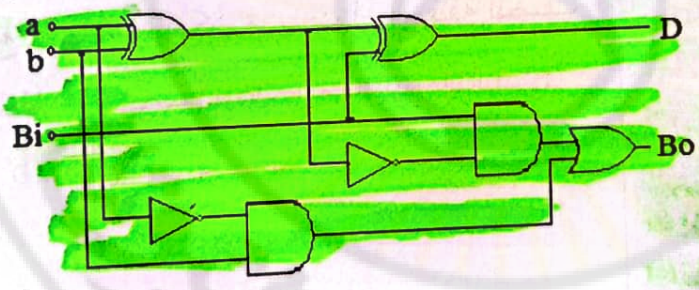
٢- جدول الحقيقة: ..... ←

٣- التوابع المنطقية:

$$D = a'b'B + a'bB' + ab'B' + abB = a \oplus b \oplus B$$

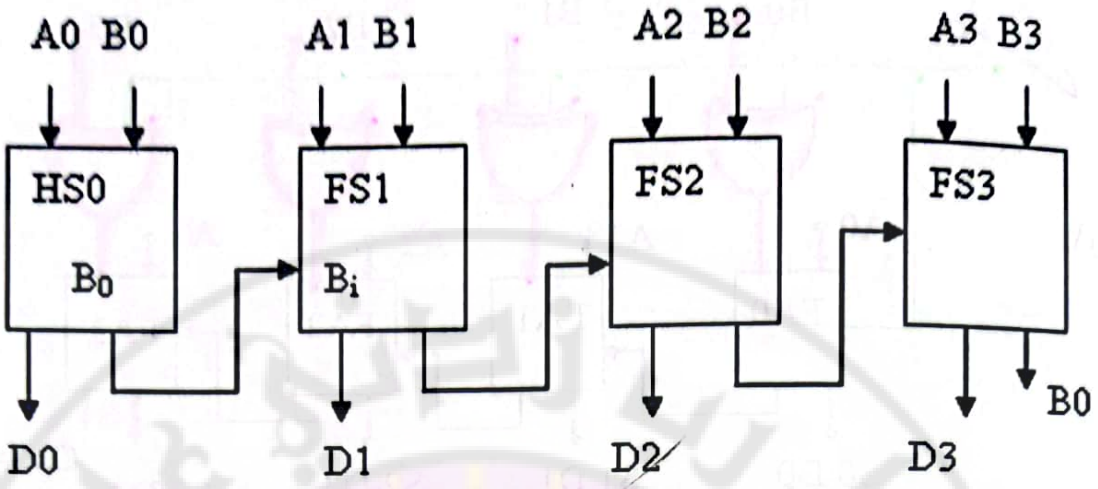
$$Bo = a'b'B + a'bB' + a'bB + abB = (a \oplus b)'.B + a'b$$

٤- المخطط المنطقي:



ثالثا: الطارح الثنائي المتوازي Parallel binary subtractor

بشكل مماثل لدارة الجامع الثنائي المتوازي، يبدو الشكل العام لطارح متوازٍ بطول 4 bits مثلاً، كما هو مبين في الشكل ٩-١٢. وهو يماثل المخطط المنطقي للجامع على التوازي باستثناء استخدام FS عوضاً عن FA وعوضاً عن مفهوم المنقول يستخدم مفهوم الإعارة.



٩-١٢ المخطط الصندوقي للطارح المتوازي بطول 4 bits

٩-٥ الجامع / الطارح الثنائي المتوازي

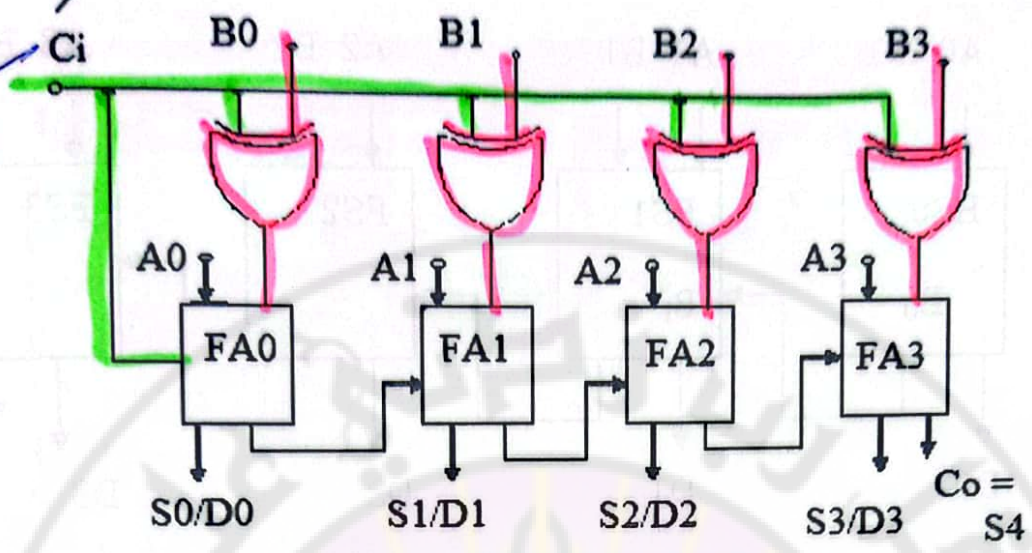
.Parallel binary adder / subtractor

بالنظر إلى ان عمليتي جمع أو طرح عددين ثنائيين، تعّد من أكثر العمليات الحسابية أهمية وهي متلازمة في كثير من العمليات التي تنفذ بالحواسيب الرقمية خلال معالجة المعطيات، فانه عوضاً عن تصميم دارات خاصة بالجمع وأخرى خاصة بالطرح على التوازي، غالبا ما يتم تصميم داراتٍ للجمع والطرح على التوازي بدارة واحدة وبالطول المناسب، تسمى بدارات الجامع/الطارح على التوازي. انطلاقاً من ان عملية طرح أي عددين تعني جمع المطروح منه مع المتمم للأثنين للمطروح، يمكن ان يكتب من أجل العددين A و B إذا كان كل منهما مكوناً من 4 bits مثلاً، بالنسبة لعملية الطرح:

$$A - B = A_3\ A_2\ A_1\ A_0 + \underline{B'_3 B'_2 B'_1 B'_0}$$
$$V_4 V_3 V_2 V_1 V_0 + 1 = S_4 S_3 S_2 S_1 S_0$$

حيث V_4 يلغى (لأن الجمع يتم مع المتمم للواحد).

وبالنظر إلى ان: $B \oplus 0 = B$ و $B \oplus 1 = B'$، يمكن باستخدام بوابات XOR تحقيق دارة جمع / طرح بطول 4 bits، وفق ما هو مبين في الشكل ٩-١٣.



شكل ٩-١٣: دارة جامع طارح على التوازي بطول 4 bits

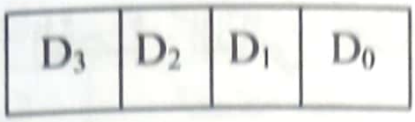
تم في دارة الجامع / الطارح وصل مكونات العدد الأول A مباشرة إلى دارة الجامع التام، أما المدخل الثاني للجامع، أي B يتم وصله مع إشارة التحكم C_i عن طريق بوابة XOR لتحقيق الوظيفة المزدوجة للدارة. إشارة التحكم الخارجية C_i في الدارة هي التي تتحكم بتحديد وظيفة الدارة. فإذا كانت $C_i = 0$ مثلاً، تكون الدارة للجمع، لأن بوابة XOR تكون وكأنها غير موجودة وان قيمة المنقول الداخل في دارة الجامع التام الأول صفر أيضاً. أما إذا كانت $C_i = 1$، تتحول وظيفة الدارة إلى الطرح، لأن بوابة XOR تؤمن المتمم للواحد للعدد B ليتم جمعه مع العدد A. ولتحويل المتمم للواحد إلى متمم للاثنين، فإن الواحد اللازم لذلك موجود على مدخل أول جامع تام بالدارة ويجمع مع أول خانة من النتيجة. يمكن تشكيل مثل هذه الدارة بالطول المطلوب، علماً ان عمليتي جمع عددين كل منهما بطول 4 bits وطرحهما تتطلب وفق الطريقة التقليدية، انشاء جدول حقيقة بـ $2^9 = 512$ Entries. وهنا تجدر الإشارة إلى انه خلال التعامل مع جامع / طارح على التوازي يمكن مصادفة موضوعين متباينين، هما: - الأول يتعلق بانتشار المنقول

- والثاني يتعلق بالطفح أو الفيض

وفيما يلي توضيح لكل من هذين المفهومين، وفق ماهو مبين أدناه.

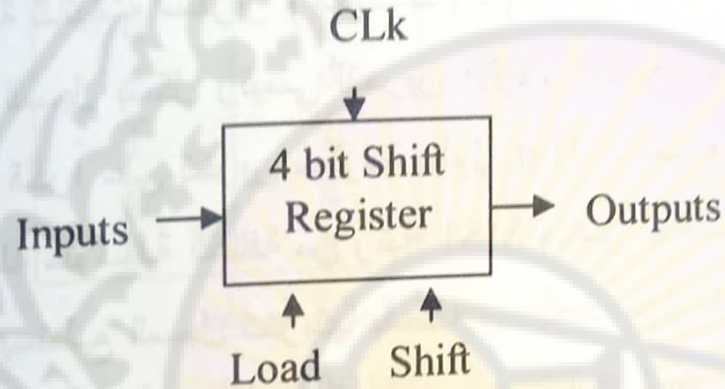
الفصل الثاني عشر
السجلات Registers

تُعدّ السجلات واحدة من أهم الدارات المنطقية التتابعية، نظراً لاستخداماتها العديدة في مختلف الأنظمة الرقمية. والسجل Register هو عبارة عن دارة تتابعيه مكونة من مجموعة ثنائيات استقرار (قلابات) بالإضافة إلى بعض البوابات المنطقية. يستخدم السجل بشكل رئيسي لتخزين الأرقام الثنائية، بحيث كل خلية بالسجل تمثل قلاباً قادراً على تخزين معلومة بطول 1bit. تصنع السجلات على شكل دارات متكاملة متوسطة الكثافة من رتبة MS1 وتتميز بمعامل أساسي هو طول الكلمة الذي يستطيع السجل تخزينها بلحظة ما. يحدّد طول السجل بعدد البتّات التي يتشكل منها. فمثلاً، لتخزين كلمة مكونة من أربعة أرقام ثنائية مثل الكلمة $(1110)_2$، يلزم سجل مكون من أربع قلابات موصولة مع بعضها بطريقة ما تجعل من هذه البتّات تشكل مجموعة متكاملة عند التعامل معها. وبالنظر إلى ان السجلات تستخدم لتخزين البيانات بشكل مؤقت، فهي تشكل جزءاً مهماً في بنية وحدة المعالجة المركزية. من أهم الأسباب التي أدت إلى انتشار استعمال السجلات، هو قدرتها على تخزين البيانات وإجراء العديد من العمليات على المعطيات الثنائية المخزنة بها، حيث يمكن باستخدامها إجراء عمليات، مثل الإزاحة والتدوير والتي تعني بدورها بعض العمليات الحسابية من ضرب وقسمة وغيرها. تمتاز السجلات باعتبارها وسائط تخزين بشكل أساسي بسرعتها في الاستجابة الزمنية بالمقارنة مع غيرها من وسائط التخزين الأخرى. سنتعرف فيما يلي على التركيبة العامة للسجلات من خلال أمثلة على سجلات بطول 4 bits فقط، بحيث بتّات السجل الواحد تكون مرتبة بجنب بعضها البعض، وفق ما هو مبين في الشكل ١٢-١.



شكل ١٢-١: سجل بطول 4 bits

يمكن تعميم كل ماينطبق على سجلات بطول 4bits على سجلات بأطوال كلمة مختلفة. يظهر الرمز المنطقي للسجل في الشكل ١٢-٢ طول كلمة السجل وبعض الإشارات القيادية للتحكم.



شكل ١٢-٢: الرمز العام للسجل

يلاحظ من الرمز المنطقي للسجل، بانه خلال التعامل مع السجل قد يصادف بالإضافة للاسم (والذي غالبا ما يدل على الوظيفة) طول الكلمة، خطوط للتغذية ومجموعة من الإشارات الأخرى مثل:

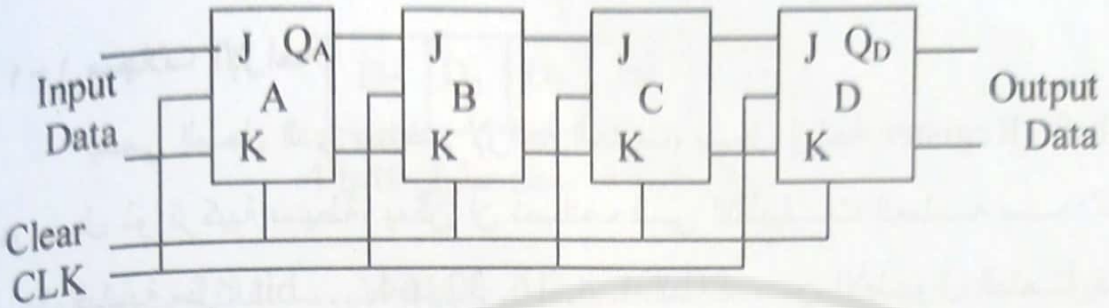
- المداخل Inputs: وقد تكون مدخلاً واحداً يستخدم لإدخال المعطيات للسجل أو عدة مداخل مساوية لطول السجل.
- المخارج Outputs: وتستخدم لإظهار محتويات السجل وعددها قد يكون مخرج واحد أو عدة مخارج بعدد خلايا السجل.
- نبضات الساعة CLK، حيث لكل قلاب يدخل في تركيبة السجل يوجد مدخل خاص لنبضات الساعة التي تقود عمل السجل.
- إشارات تحكم أخرى مثل ... Load, Shift, Reset, وقد تصادف كلها بالسجل الواحد أو البعض منها بحسب تركيبة ونوع أو وظيفة السجل.

١٢-١ سجلات الإزاحة

يسمى السجل الذي يستخدم لإزاحة البيانات، بسجل إزاحة Shift Register وهو سجل ذو تركيبة بسيطة. يمكن ان نصادف في التطبيقات العملية سجلات بأطوال قياسية مثل: bit ... ,64 ,32 ,16 ,8 ,4 انطلاقاً من الأطوال المتعارف عليها لكلمة الحاسوب. بالتالي الاختلاف بين سجل إزاحة وآخر يكمّن بطول كلمة السجل وبطريقة التوصيل للمكونات الداخلية للسجل. بحسب آلية التوصيل الداخلية للسجل نميز بين ثلاثة انواع من سجلات الإزاحة، هي:

١٢-١-١ سجلات الإزاحة نحو اليمين

يرتبط مفهوم إزاحة البيانات أو نقلها بوساطة سجل بطريقة إدخال البيانات وإخراجها. بالتالي لتوضيح التركيبة العامة لسجلات الإزاحة نحو اليمين (وغيرها من السجلات الأخرى) ننطلق من الترميز المعتمد للبيانات بالسجلات. ان الترميز المعياري للبيانات ضمن سجل هو ان البتّ بأقصى اليمين دوما يمثل البتّ ذو الوزن الأدنى LSB، والبتّ بأقصى اليسار يمثل الـ MSB. فمثلاً، عند تمثيل المعطيات $(1110)_2$ بسجل طوله 4 bits مثل السجل المبين في الشكل ١٢-١، يجب ان نحرص على ترميز $D_0 = 0$ بانها الـ LSB وبالتالي يجب ان تكون بأقصى اليمين و $D_3 = 1$ هي MSB بأقصى اليسار. تعني سجلات الإزاحة نحو اليمين، سجلات بإمكانية إدخال معطيات خارجية بالخانة المقابلة لـ MSB وإزاحة ماهو موجود بالخانة MSB نحو اليمين وإخراج القيمة الموجودة بالخانة التي تقابل الـ LSB خارج السجل. التركيبة العامة لمثل هذا السجل بطول 4bits، مبينة في الشكل١٢-٣. يتضح من الشكل أدناه، ان كل خانة من السجل خصصت بقلاب وقد تم ترميز القلابات الأربعة بالأحرف A, B, C, D، بحيث ان القلاب A يقابل الخانة MSB والقلاب D يقابل الخانة LSB.



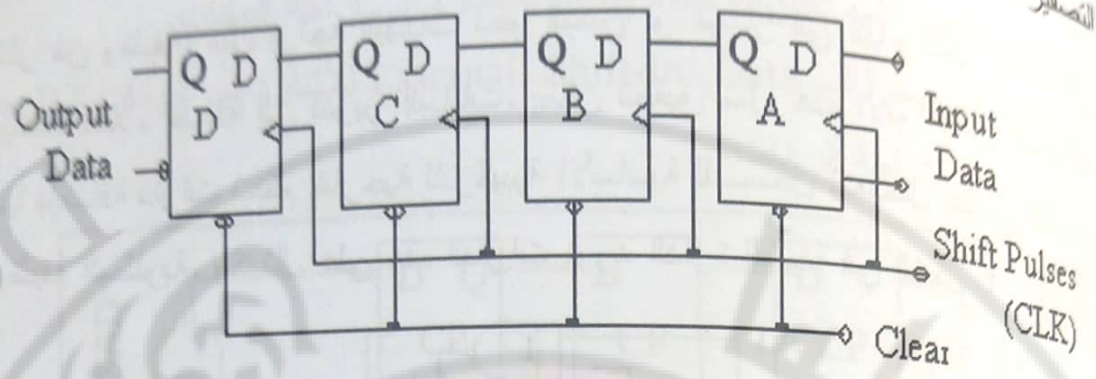
شكل ١٢-٣: سجل إزاحة نحو اليمين بطول 4 bits.

تم في الشكل أعلاه استخدام القلاب J-K، علماً انه كان بالإمكان استخدام أي نوع آخر من القلابات. تُعدّ دارة السجل أعلاه بسيطة، لأنها مكونة من القلابات فقط وتسمح بإزاحة البيانات نحو اليمين عند كل إدخال للمعطيات من القلاب A (مترافقاً مع ورود النبضة الساعية) من دون الحاجة لوجود أي بوابة منطقية إضافية. تقدح الساعة المشتركة CLK جميع القلابات بآن واحد (في الشكل أعلاه لم تحدد الحافة التي يتم عندها القدح مع كل نبضة، لأنها يمكن ان تكون الموجبة ويمكن ان تكون السالبة بحسب نوع القلاب). توصل جميع مداخل التصفير للقلابات الأربعة بآن واحد إلى مدخل واحد غير متزامن Clear الذي يمكن ان يكون فعّالاً بالصفر أو بالواحد. بمجرد تفعيل إشارة Clear، تصفر كافة القلابات دفعة واحدة وتستخدم عادة هذه الإشارة قبل بدء التعامل مع السجل. تزاح المعطيات المخزنة بالسجل باتجاه اليمين مع كل ورود للنبضة الساعية بتطبيق قيمة الواحد أو الصفر على المدخل الرئيسي للقلاب ويمكن أخذ عينة من المخارج لحظة بلحظة.

١٢-١-٢ سجلات الإزاحة نحو اليسار

يعمل هذا النوع من السجلات Shift- Left Registers بعكس النوع السابق، إذ يسمح بإدخال بيانات الإزاحة من البتّ الذي يقابل خانة الـ LSB ويقوم بإزاحة البيانات المخزنة بالسجل باتجاه البتّ MSB (اليسار). وبالتالي التركيبة العامة لمثل هذا النوع من السجلات بطول 4 bits مبينة في الشكل ١٢-٤. وهنا تم من باب التنويع، استخدام قلابات من نوع D ، كما انه حدّد بان القلابات المستخدمة

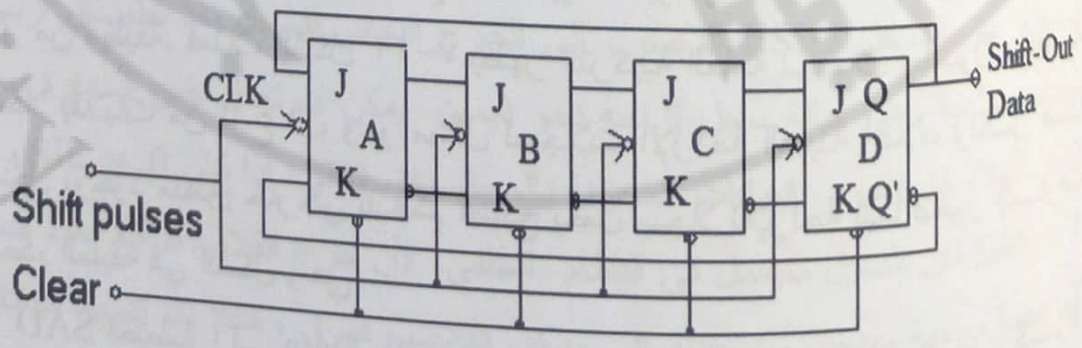
بالسجل تستجيب للوجه الصاعد للنبضة. وان إشارة المسح الموصولة مع إشارة التصفير فعّالة بالصفر.



شكل ١٢-٤: سجل إزاحة نحو اليسار بطول 4 bits

١٢-١-٣ سجلات الإزاحة الحلقية

يلزم بسجلات الإزاحة نحو اليمين أو اليسار إدخال معطيات خارجية من مدخل النقل التسلسلي للسجل وإخراج بيانات من مخرج النقل التسلسلي حتى تتم عملية الإزاحة. أما بالنسبة للسجلات الحلقية Shift-Around Registers فتدور البيانات التي تكون مخزنة مسبقاً بهذا النوع من السجلات من طرف إلى آخر من دون أي فقد بالبيانات، ويمكن مع كل نبضة ساعية إجراء عملية الدوران وأخذ عينة عن الخرج. وهنا نميز أيضا بين نوعين من سجلات الدوران، إما سجلات دوران باتجاه اليمين وإما دوران باتجاه اليسار. الشكل العام لسجل حلقي نحو اليمين بطول 4bits يبدو كما هو مبين في الشكل ١٢-٥.



شكل ١٢-٥: سجل دوران (حلقي) نحو اليمين بطول 4 bits

١٢-٢ الانواع الأساسية للسجلات

تصنف السجلات بحسب طرق إدخال البيانات وإخراجها إلى أربعة انواع أساسية

١- سجل بدخل تسلسلي - خرج تسلسلي Serial In- Serial Out اختصاراً SISO

٢- سجل بدخل تسلسلي-خرج تفرعي Serial In–Parallel Out اختصاراً SIPO

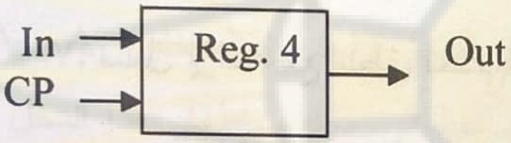
٣- سجل بدخل تفرعي-خرج تفرعي Parallel In – Parallel Out اختصاراً PIPO.

٤- سجل بدخل تفرعي-خرج تسلسلي Parallel In – Serial Out اختصاراً PISO.

وفيما يلي نعرض باختصار التركيبة العامة لكل من هذه الانواع الأربعة .

١٢-٢-١ سجل بدخل تسلسلي - خرج تسلسلي

الرمز الصندوقي لسجل إزاحة Shift register من هذا النوع بطول 4bits مثلاً، يبدو كما في الشكل أدناه.



ويقابل هذا النوع من السجلات تقنية FI FO (First-In First-Out)، أي أول من يدخل هو أول من يخرج. وفيما يلي نظهر عملية إدخال البيانات وإخراجها في سجلات SISO من خلال المثال التالي:

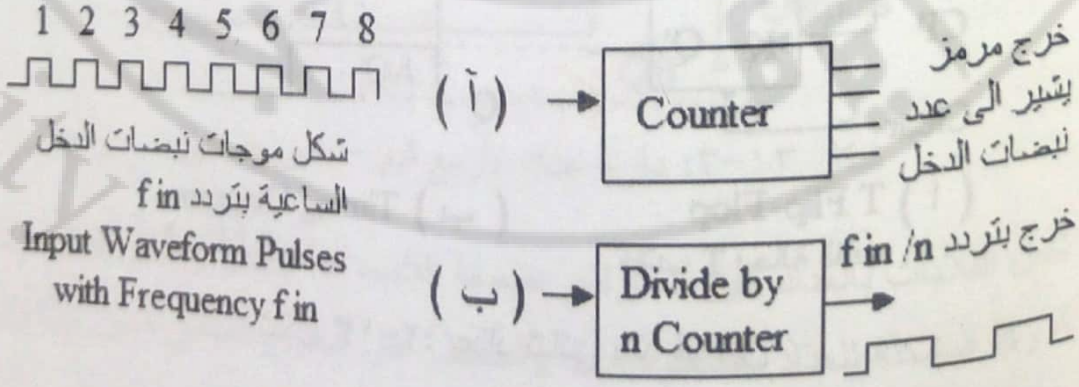
مثال: من أجل سجل نوع SISO بطول 4bit وخلاياه مكونة من القلابات RS يكون المخطط المنطقي، وفق ما هو مبين في الشكل ١٢-٨. يتضح من الشكل ان القلابات المشكلة للسجل موصولة على التسلسل، بحيث خرج كل قلاب يكون موصولاً مع مدخل القلاب الذي يليه. وان إشارة التحكم Clear للتصفير هي فعّالة بالصفر وموصولة مع Reset كافة القلابات لتتم عملية التصفير لكامل السجل دفعة واحدة. توصل إشارة الساعة CLK إلى كافة القلابات بآن واحد. وتدخل البيانات بالتوصيلية

الفصل الثالث عشر
العدّادات Counters

بالنظر إلى انه أثناء عملية العّد من قيمة إلى أخرى يجب بكل لحظة تذكّر القيمة السابقة التي تم التوصل إليها خلال عملية العّد لمعرفة القيمة اللاحقة، يلزم العدّاد الذي سيقوم بعملية العّد ذاكرة مؤقتة لحفظ القيمة السابقة. تستخدم القلابات بوصفها ذاكرة مؤقته في تركيبة العدّاد لحفظ القيمة التي يتوصل لها العدّاد في كل لحظة. إذاً العدّاد Counter بالإطار العام هو مجموعة من القلابات، مربوطة مع بعضها البعض بطريقة ما تسمح بمعرفة الحالة التي يتم التوصل إليها خلال عملية العّد وتسمح تحت تأثير نبضات الساعة بالانتقال من حالة إلى أخرى. يمكن ان يكون الدخل الرئيسي للعدّاد قطار من النبضات الساعية المتولدة من أي مصدر خارجي للنبضات يصدر نبضات يفصل بينها فواصل زمنية ثابتة (أو عشوائية). يمكن استخدام العدادات للقيام بواحدة من الوظيفتين التاليتين:

- إعطاء عدد النبضات المطبقة على الدخل.
- القيام بعملية القسمة، حيث يستطيع العدّاد إعطاء نبضات في الخرج بتردد أصغر من تردد اشارة الدخل، بالتالي العدّاد يعمل مقسماً للتردد Frequency divider.

يمكن تمثيل أشكال استخدام العدادات بالرموز التوضيحية المبينة في الشكل ١٣-١.

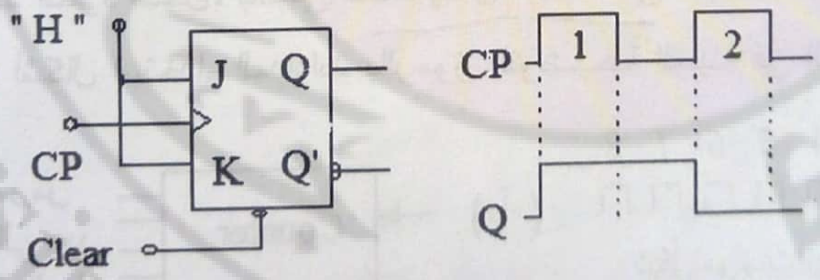


شكل ١٣-١: الرموز العامة للعدّاد

يكون دخل العّداد دوما نبضات ساعية، في الوقت الذي يكون فيه الخرج إما قيمة مرمزة كما في الشكل ١٣-١آ. وإما نبضة بتردد معين كما في الشكل ١٣-١ب.
نميز بحسب طريقة وصل الساعة إلى القلابات المكونة للعدّاد بين نوعين رئيسين من العدّادات هما:

- العدّادات غير المتزامنة Asynchronous Counter أو غير المتوافقة والاسم الشائع لها هو العدّادات التموجية (Ripple Counters عدادات ريبل).
- العدّادات المتزامنة Synchronous Counter أو العدّادات المتوافقة.

تستخدم العدّادات بنوعيها في النظم الرقمية لتعدّاد عدد حدوث حدث ما، ولتوليد إشارات توقيت Timing Signals للتحكم بتتابع عمليات محددة في هذه الأنظمة. يمكن ان يكون تتابع حالات العدّاد عداً ثنائياً، فينتج في هذه الحالة عدّاد ثنائي ويمكن ان يكون أي تتابع عددي آخر غير الثنائي. ولفهم آلية العّد باستخدام العدّادات الثنائية مثلاً، نفترض انه لدينا قلاب من نوع T أو قلاب J-K (ربطت مداخله مع بعضها) ووصل مدخل القلاب على فولط عالٍ ليعمل بصيغة القلب، وفق ما هو مبين في الشكل ١٣-٢آ، أي سيغير القلاب حالته (إلى الحالة المعاكسة) مع كل ورود جديد للنبضة الساعية. بتتبع عمل هذه الدارة من خلال المخطط الزمني المبين في الشكل ١٣-٢ب، يلاحظ انه بعد تفعيل المدخل Clear سيصبح محتوى القلاب Q = 0.



(آ) T Flip-Flop
قلاب T بحالة القلب

(ب) Timing Diagram
المخطط الزمني

شكل ١٣-٢: عدّاد ثنائي يعّد قيمتين

الفصل الرابع عشر

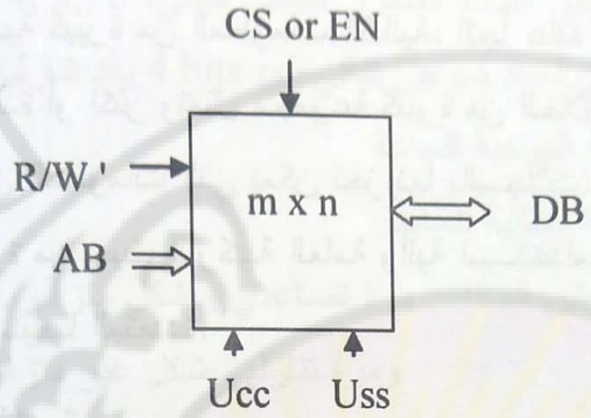
الذواكر Memories

لتخزين كمية كبيرة من المعلومات الثنائية، نلجأ عادة لاستخدام وحدة الذاكرة التي تضم دارة واحدة أو أكثر وتمثل مجموعة كبيرة من الخلايا القادرة على تخزين حجم أكبر بكثير من المعلومات التي يمكن تخزينها بالسجلات. سنتناول بهذا الفصل المفهوم العام للذاكرة من حيث التركيبة العامة وآلية استخدامها من دون الدخول بتفاصيل تقانات تصنيعها المتعددة.

١٤-١ مفهوم الذاكرة

سنكتفي هنا من مجمل انواع الذواكر المعروضة، الإشارة إلى الذواكر المصنعة من انصاف النواقل فقط، بعدما حل هذا النوع من الذواكر محل الحلقات المغناطيسية (التي كانت توجد في الأجيال الأولى للحاسبات). ينظر للذاكرة Memory على انها دارة منطقية تستعمل لتخزين المعلومات وترتبط بشكل وثيق مع دارة منطقية أخرى أكثر تعقيداً هي دارة المعالج الصغري. تشكل الذاكرة مع المعالج العناصر الأساسية في تركيبة أي حاسوب، حيث لا يمكن لمعالج صغري مستخدم بتطبيق ما ان يكمل أعماله بمعزل عن وجود ذاكرة (كجزء منه أو مستقلة عنه) لتخزين المعلومات المستقبلة من تجهيزات الدخل لمعالجتها ونقل النتائج إلى تجهيزات الخرج. يمكن تصور الذاكرة بوصفها مجموعة من السجلات المتماثلة في الطول. بالتالي فهي تمثل مصفوفة خلايا تخزين مرتبة وفق شكل محدد وان كل خلية ممثلة بإحداثيتين على محوري المصفوفة، بحيث يسهل الوصول عبرهما بأي لحظة إلى أي خلية في الذاكرة. تمثل الخلية Cell الواحدة في الذاكرة أصغر حيز لتخزين البيانات، حيث كل خلية قادرة على تخزين معلومة بطول bit 1 (0 أو 1). تسمى مجموعة الخلايا التي تشكل صفاً واحداً في مصفوفة الذاكرة بالموقع Location أو

بطول كلمة الذاكرة Memory Word. يوجد لكل ذاكرة مصنعة على شكل دارة متكاملة مجموعة من الإشارات القياسية مثل الإشارات المبينة بالرمز الصندوقي للذاكرة في الشكل ١٤-١.



شكل ١٤-١: الرمز الصندوقي للذاكرة مع أهم الإشارات القياسية

تشكل الرموز m, n في الشكل أعلاه مصفوفة الذاكرة Memory Matrix، بحيث يشير الرمز m لعدد صفوف المصفوفة (المواضع) و n لعدد الأعمدة (عدد الخلايا في الصف الواحد) و الجداء mxn إلى الحجم الكلي للذاكرة بالبتّ. يتم التعامل مع الذاكرة من خلال عمليتين أساسيتين هما:

الأولى – كتابة المعلومات ضمن الذاكرة Write Operation (ونرمز لها W).

الثانية – قراءة المعلومات من الذاكرة Read Operation (ونرمز لها R).

توجد إشارة تحكم منفصلة لكل من هاتين العمليتين، حيث كل إشارة منهما قد تكون فعّالة بالصفر أو بالواحد (بحسب نوع الدارة). ويمكن دمج الإشارتين بخط واحد بالاسم R/W'، بحيث تكون إحداها فعّالة بعكس الأخرى (لأنه لا يمكن بنفس اللحظة إجراء عمليتي القراءة والكتابة في الذاكرة). تكون الذاكرة موجودة مع غيرها من الدارات المتكاملة ضمن منظومة ما. لتميز قطعة الذاكرة المراد التعامل معها بلحظة ما عن غيرها من الدارات الأخرى. تستخدم إشارة تحكم بالاسم CS (اختصاراً من Chip Select وفي بعض الدارات يستخدم الرمز EN من Enable أي السماحية)

لانتقاء الدارة. قد يكون للقطعة الواحدة أكثر من إشارة CS ($CS_1, CS_2, \ldots$ بعضها فعّال بالصفر والبعض الآخر فعّال بالواحد). بالتالي عند الرغبة بالتعامل مع دارة ما، يجب تأمين سوية الجهد المناسبة لكل إشارة انتقاء CS على حدة. يشير الرمز AB في الشكل أعلاه إلى ممرات العناوين Address Bus والرمز DB إلى ممرات المعطيات Data Bus. وقد أشرنا في المخطط لكل من هذين الرمزين بسهم مزدوج للإشارة إلى ان هذه الممرات قد تكون أكثر من خط (يرسم عادة بالتصاميم الإلكترونية خط مائل على الممر ويكتب عليه رقم يدل على عددها). يكون مصدر ممرات العناوين AB دوما المعالج، أما ممرات المعطيات فقد تكون بالاتجاهين أي للكتابة وتكون داخلة للذاكرة أو للقراءة وتكون خارجة من الذاكرة. تحمل الممرات AB قيماً ثنائية تمثل العناوين و DB قيماً ثنائية تمثل المعطيات. يعطي عدد خطوط AB المجال العنوني الممكن للذاكرة، بمعنى انه بخط عنونة واحد A_0 يمكن عنونة $2^1 = 2$ موضع في الذاكرة، هما 0 لموضع و 1 لموضع آخر. ويمكن بخطين A_1A_0 عنونة $2^2 = 4$ مواضع تكون بالعناوين 00, 01, 10, 11 ويمكن بعشرة خطوط مثلاً $A_9A_8 \ldots A_0$ عنونة 1024 موضع أو ما يعادل 1 KB ويمكن بـ 16 خط، عنونة 2^{16} موضع، أي من العنوان:

$$0000_H = 0000\ 0000\ 0000\ 0000_B$$
$$0001_H = 0000\ 0000\ 0000\ 0001_B$$
.....

وحتى $FFFF_H = 1111\ 1111\ 1111\ 1111_B$ وهذا ما يعادل 64 KB .

يعطي عدد خطوط DB طول كلمة الذاكرة وقد يكون هذا الطول 1 bit وبهذه الحالة تكون مصفوفة الذاكرة مكونة من عمود واحد، وقد يكون طول الكلمة 4 bit. الطول القياسي لكلمة الذاكرة هو 1 byte = 8 bits انطلاقاً من ان المعطيات تكون ممثلة في الذاكرة عادة وفق كود ASCII الذي يخصص بايت واحد لكل محرف. آلية التعامل مع الذاكرة بغض النظر عن نوعها تتم كالتالي:

يرسل المعالج عندما يقرر التعامل مع الذاكرة، عنوان الموضع الذي يرغب التعامل معه في الذاكرة عبر AB وبعد تفعيل الإشارة CS (يمكن جعل الإشارة CS مؤهلة بشكل مستمر بوصلها مع التغذية بسوية الجهد المناسبة، ولكن لا يفضل ذلك) وانتقاء الذاكرة عن غيرها من الدارات الأخرى، يرسل المعالج إشارة التحكم بحسب العملية المطلوبة على المأخذ 'R/W وفور جهوزية الذاكرة للاستجابة يتم إرسال المعطيات أو استقبالها بحسب العملية المقررة عبر ممر المعطيات DB.

١٤-٢ انواع الذواكر

تصنف الذواكر المصنعة من انصاف النواقل وفق معايير عدة، نذكر منها المعايير الأكثر انتشاراً مثل معيار: - العبور (الوصول) إلى المعطيات داخل الذاكرة

- مبدأ تحقيق خلايا الذاكرة الأولية
- إمكانية تغيير محتويات الذاكرة
- الوثوقية والكلفة وغيرها من المعايير الأخرى

أصبحت الذواكر المستخدمة في كل من المجموعات المبينة أعلاه متقاربة ومتداخلة، بحيث يصعب تحديد حدود معينة بين عناصر مجموعة وأخرى. فمثلاً، وفق معيار إمكانية تغيير المعطيات، نميز بين الانواع الأساسية التالية:

- ذواكر القراءة والكتابة RWM من Read Write Memories التي يمكن استخدامها لعملية قراءة المعطيات وكتابتها.
- ذواكر القراءة فقط ROM من Read Only Memories وهي ذواكر مخصصة للقراءة فقط ولا يمكن الكتابة فيها.
- ذواكر من نوع خاص تسمى RMM من Read Mostly Memories تخصص غالبا للقراءة إلا انه يمكن تغيير محتواها قبل وضعها بالخدمة.

وبحسب معيار المبدأ المستخدم في تنجيز خلايا الذاكرة الأولية نميز بين نوعين أساسيين من الذواكر هما:

- ذواكر بخلايا ساكنة Static وفيها تكون كل خلية مكونة من ثنائي استقرار.
- ذواكر بخلايا متحركة Dynamic وفيها تكون كل خلية في الذاكرة مبنية على استخدام السعات البارازيتية.

سنتعامل مع الذواكر على مختلف انواعها باعتبارها دارات دعم (للمعالجات الصغرية) على انها:

١- ذواكر غير دائمة Volatile Memories تفقد محتوياتها بمجرد انقطاع التغذية الكهربائية وهي تقابل النوع RAM وتقسم بدورها إلى:
- ذواكر استاتيكية SRAM ، تخزن فيها المعلومات داخل تشكيل من الترانزستورات التي تمر بحالتين مستقرتين (0 أو 1) وتبقى بالوضع الذي تستقر به طالما هناك وجود للتيار الكهربائي.
- ذواكر ديناميكية DRAM ، تخزن فيها المعلومات وتحفظ على هيئة شحنة كهربائية بمكثف. لكن بالنظر للتسريب الذي يمكن ان يحدث في المكثفات يلزم تعريض هذا النوع من الذواكر إلى دورات انعاش Refresh لتعويض جزء الشحنة المتسرب من المكثف على فترات زمنية تتفاوت بحسب نوع الذاكرة ، علماً ان هذه الفترات لا تتجاوز عند أي منها بضعة ميلي ثانية.

٢- ذواكر دائمة Nonvolatile Memories وهي ذواكر تحتفظ بمحتوياتها حتى ولو انقطعت التغذية الكهربائية عنها وهي تقابل الذواكر ROM. يوجد عدة انواع من الذواكر القابلة للقراءة ROM. تعدد انواع ROM يأتي من تعدد طرق برمجتها.

وبالرغم من استخدام كلمة برمجة هنا، فان جميع إجرائيات برمجة ذواكر القراءة فقط هي إجرائيات عتا دية وغير قابلة للعكس، تتم بوساطة مبرمج خاص للذاكرة. يوجد أربع طرق مختلفة للبرمجة هي:
- برمجة القناع Mask Programming وتسمى ذواكر ROM ذي القناع. تتم برمجة هذا النوع من الذواكر أثناء عملية تصنيع الذاكرة.
- البرمجة لمرة واحدة فقط وتسمى بذواكر PROM من Programmable ROM

يستخدم ها النوع من الذواكر التي تبرمج من قبل المستخدم في التطبيقات التي يرغب فيها المستخدم برمجة بعض الأوامر الخاصة بالتطبيق الخاص به لمرة واحدة من دون أي تعديل على محتواها بعد برمجتها. تصنّع ذواكر PROM خالية من المعلومات (أي ان محتوياتها تكون بالكامل واحدات). تحتوي على وصلات يمكن صهرها لتحويلها من واحدات إلى أصفار عند البرمجة، بالتالي عملية البرمجة تعني تحويل بعض الواحدات فيها إلى أصفار وترك بعض الواحدات كما هي، بحيث بعد الانتهاء من عملية البرمجة تصبح المعلومات ثابتة لا تتأثر بوجود التغذية الكهربائية من عدمه.

- البرمجة لعدة مرات من قبل المستخدم وهي تقابل ذواكر قابلة للبرمجة والمحو. ونميز بين نوعين من هذه الذواكر هما:

*- ذواكر EPROM من Erasable Programmatically ROM ويمكن إعادة هذا النوع من الذواكر إلى الحالة الابتدائية حتى ولو كانت مبرمجة، بتعريضها للأشعة فوق البنفسجية Ultraviolet Light.

*- ذواكر EEPROM من Electrically-Erasable PROM ويمكن بهذا النوع من الذواكر مسح الوصلات المبرمجة بوساطة إشارة كهربائية، حتى من دون الحاجة لإخراجها من القاعدة التي تكون متوضعة بها.

تصنع الذواكر المشكلة من عناصر نصف ناقلة مثل غيرها من الدارات المنطقية بتقانات متعددة، لكن معظم الذواكر التي يمكن ان نصادفها قد تكون مصنعة بواحدة من تقانات أحادي القطبية لصعوبة تأمين درجة تكامل عالية بتقانات ثنائي القطبية.

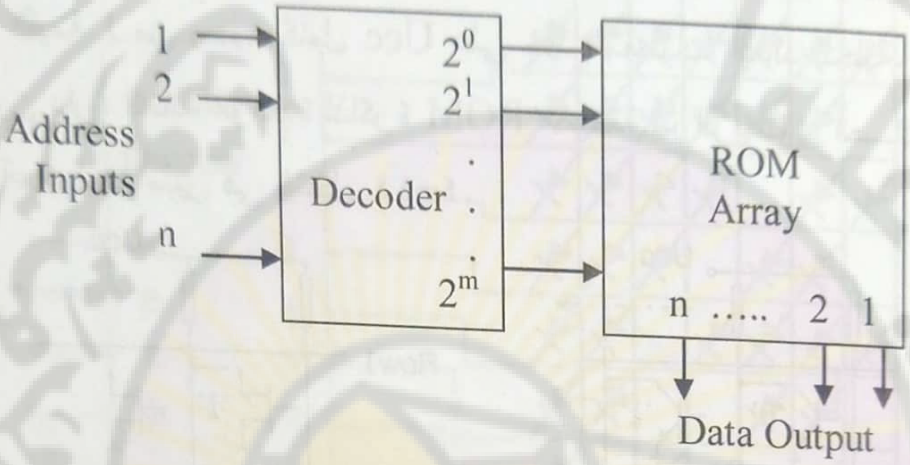
١٤-٣ تركيبة ذواكر القراءة ROM واستخداماتها

يصلح الرمز الصندوقي للذاكرة في الشكل ١٤-١ لذواكر من نوع RAM فقط لسببين هما: - وجود إشارة للكتابة في الذاكرة

- وجود اتجاهين لممر المعطيات

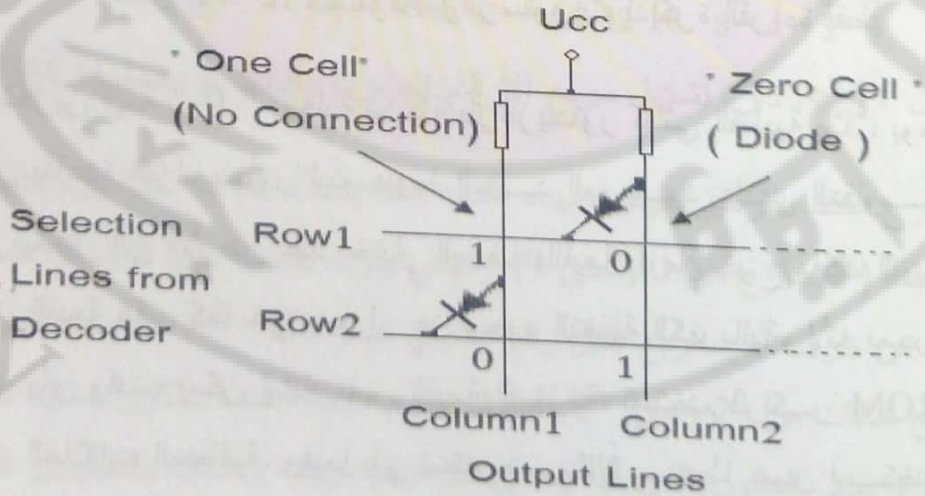
كلا الأمرين غير ممكن بالنسبة لذواكر ROM، لأنها تعلون وفق منطق اضافي للذاكرة وتسمح بمرور المعطيات باتجاه واحد فقط. يمكن تمثيل ذاكرة ROM بمخطط صندوقي، كما في الشكل ١٤-٢. يتضح من المخطط، ان الأجزاء الرئيسية لذاكرة ROM هي: - دارة فك الترميز التي تستخدم لتحديد مواقع الذاكرة المختلفة.

- دارة المصفوفة التي تمثل ساحة التخزين.



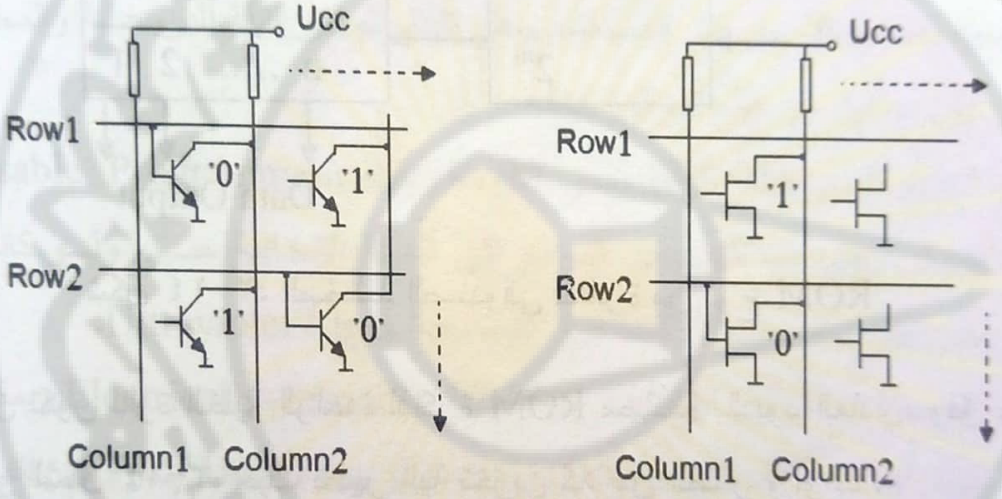
شكل ١٤-٢: المخطط الصندوقي لذاكرة من نوع ROM

يمكن ان تكون دارة الخلية الواحدة لذاكرة ROM مصنّعة بالديود العادي، وفق ماهو مبين في الشكل ١٤-٣، حيث تظهر آلية تخزين كلاً من الصفر والواحد.



شكل ١٤-٣: التركيبة العامة لخلية ذاكرة من نوع ROM بالديود

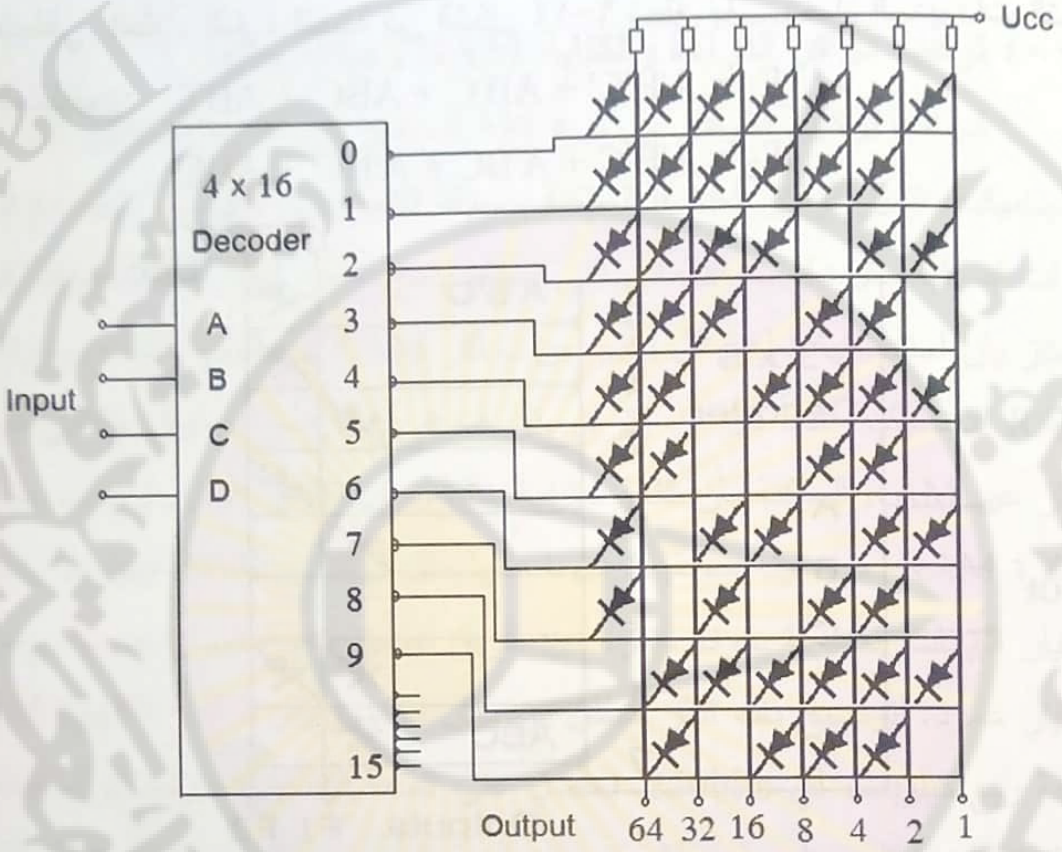
فمثلاً، إذا كان خرج مفكك الترميز فعّال بالـ 0، يمكن تخزين الصفر بالخلية بوصل خط العنوان (الصف) مع العمود (الخرج) عن طريق ديود بالانحياز الأمامي. يهبط بهذه الحالة كامل Ucc الذي تمثل واحد منطقي والموجود على المقاومة R عبر مفكك الترميز (عبوراً بالديود المنحاز أمامياً) إلى الأرض ويصبح على خط المعطيات العائد للخلية القيمة 0. بالطبع في حال عدم وجود الديود سيبقى عند انتقاء الخلية القيمة 1 من وجود كامل Ucc على خط المعطيات العائد للخلية. يمكن أيضاً ان تكون دارة الخلية الواحدة لذاكرة ROM مصنعة بترانزيستور أحادي أو ثنائي القطبية، كما هو مبين في الشكل ١٤-٤.



شكل ١٤-٤: مصفوفة ترانزستورات لذاكرة القراءة فقط

يمكن تخزين قيمة " 0 " بخلية مشكلة بترانزيستور ثنائي القطبية مثلاً، بوصل خط العنوان بقاعدة الترانزيستور ليتم هبوط الجهد الموجود على المقاومة بمجمع الترانزيستور إلى الأرض عند اختيار الصف بالقيمة 1 من خرج مفكك الترميز. بعد تخزين القيمة تبقى كما هي بمعزل عن وجود التغذية الكهربائية، لأنه بمجرد وجود التغذية بأي وقت، يمكن إيجاد نفس القيمة المخزنة. تستخدم ذواكر ROM عموماً لتتجيّز العلاقات المنطقية. وفيما يلي نستعرض مثالاً بسيطاً عن استخدام ذاكرة ROM المشكلة باستخدام مصفوفة من الديودات لتحقيق التابع المنطقي الذي يعطي على الخرج مربع أي رقم عشري يوضع على الدخل. تعالج هذه المسألة وفق التالي:

بداية لتمثيل أي رقم عشري من 0 وحتى 9 بشكل ثنائي، يلزم أربعة متحولات ثنائية، لتكن هذه المتحولات A, B, C, D ، بحيث A يمثل الـ LSB مثلاً و D يمثل الـ MSB. هذه المتحولات الأربعة تطبق على المداخل لدارة مفكك ترميز. ويمكن تنفيذ المطلوب بالدارة المبينة في الشكل ١٤-٥،

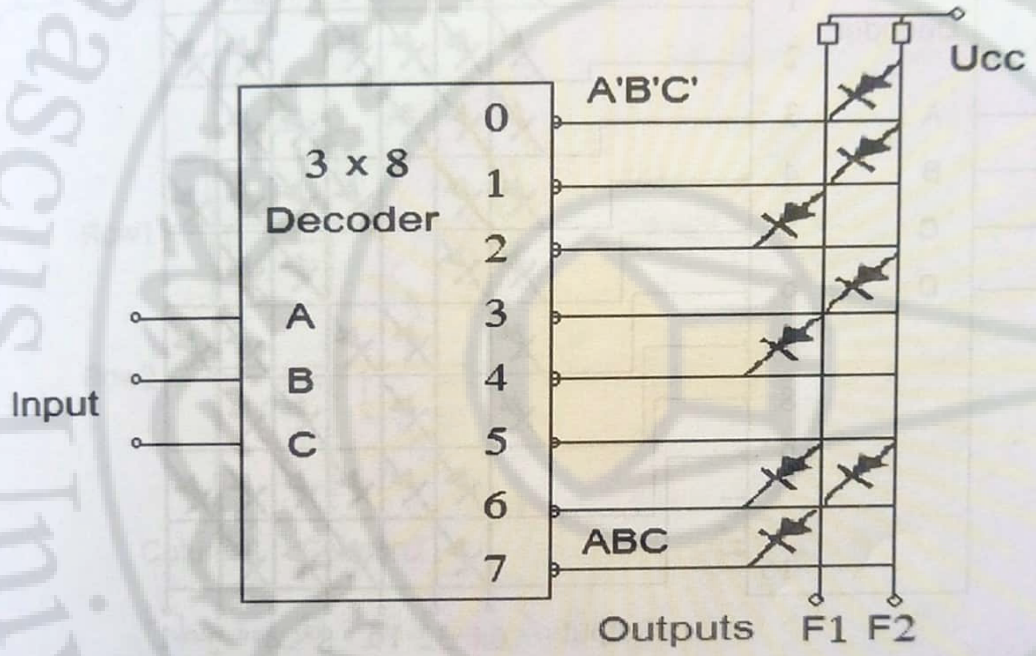


شكل ١٤-٥: دارة تشكيل مربع الأرقام العشرية باستخدام مصفوفة ROM

عدد مخارج مفكك الترميز سيكون بهذه الحالة $2^4 = 16$ Outputs. يلزم منها عشرة مخارج فقط (المخارج من 0 وحتى 9) والباقي يترك بالنسبة لهذا المثال من دون استخدام. من طرف آخر مربع أكبر رقم عشري هو $9^2 = 81$. لتمثيل الرقم 81 يكفي سبع خانات من 2^0 وحتى 2^6 . بالتالي يلزم سبعة أعمدة ستكون بمثابة الخرج من الذاكرة. يبنى الجدول الذي فيه يعطى لكل رقم مربعه وذلك بالتمثيل الثنائي لمربع كل رقم على حدة، بحيث مكان وجود الصفر يترك الديود ومكان وجود الواحد

يلغى. ذواكر PROM القابلة للبرمجة لمرة واحدة من قبل المستخدم مماثلة بتركيبتها لذواكر ROM، فهي على هيئة مصفوفة من الديودات (الترانزستورات مع خلاف ان باعث كل ترانزيستور موصول عبر وصلة Fuses (قابلة للصهر) إلى العمود، أي الخرج. بالتالي يمكن بأسلوب مماثل باستخدام ذواكر PROM تحقيق أي تابع منطقي. فمثلاً، الدارة المبينة في الشكل ١٤-٦ تحقق باستخدام الديودات التابعين المنطقيين: $F_1 = A'B'C' + AB'C + ABC' + AB'C'$

$F_2 = A'B'C + A'BC + A'BC' + ABC$.



شكل ١٤-٦: مثال حول بناء مصفوفة لتحقيق توابع منطقية باستخدام PROM

لأنه من أجل التابع F_1 مثلاً (إذا افترضنا ان A تمثل الـ LSB) كافة التراكيب CBA = 000, 101, 011, 001 التي تقابل الأرقام العشرية 0, 1, 3, 5 يكون فيها التابع F_1 محقق لعدم وجود ديودات (أي قيمة الواحد). تتم برمجة ذواكر PROM عن طريق صهر الوصلات (الموصولة على التوالي مع الديود أو الترانزيستور) التي لا يراد وجودها بالمصفوفة بتمرير نبضة تيار كهربائي من خلالها. بمجرد برمجة الدارة لا يمكن تغيير محتواها أو إعادة مسحها وتصبح وكأنها ذاكرة ROM.

أما ذواكر EPROM فتختلف عن ذواكر PROM العادية، لأنه يمكن مسح محتوياتها بعد برمجتها. نكتفي بهذا الحد، من دون الدخول بتفاصيل عملية البرمجة، التي تُعدّ خارج مجال الكتاب. يمكن إيجاد مزيد من المعلومات حول الموضوع في المرجع [6]

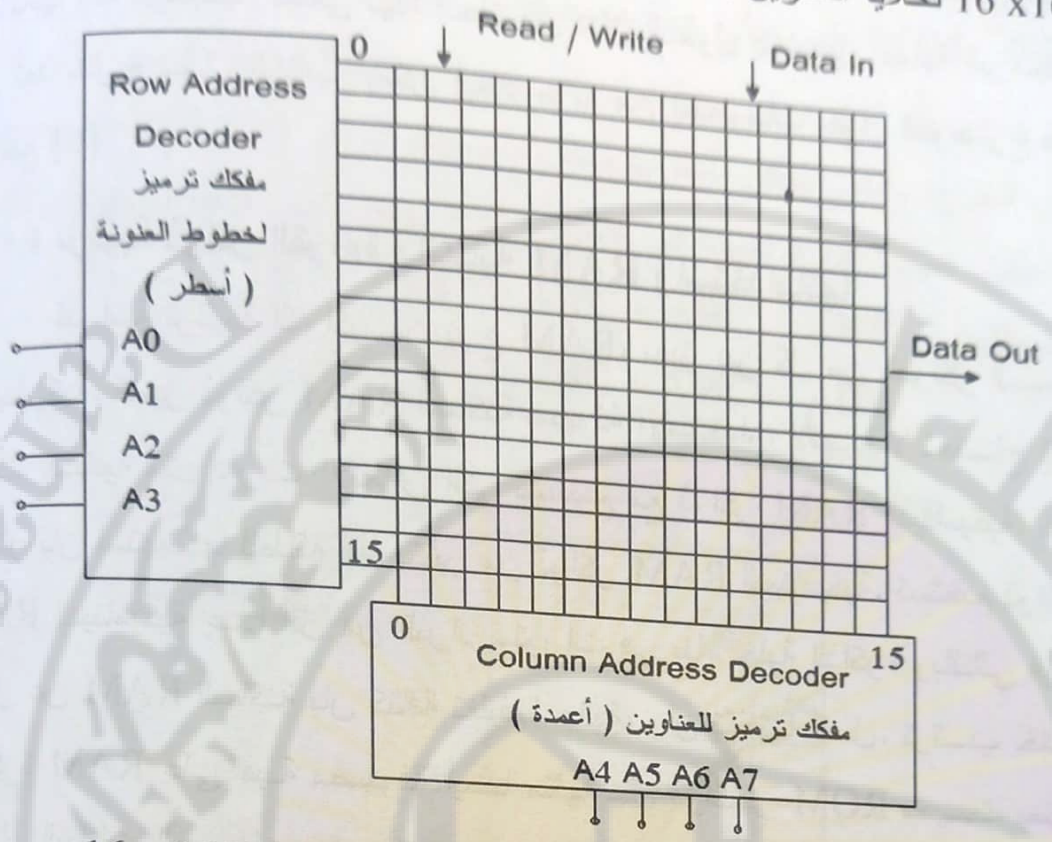
١٤-٤ تركيبة ذواكر القراءة والكتابة RAM واستخداماتها

لدراسة تركيبة الذواكر من نوع RAM، نميز بين كل من الذواكر الساكنة والديناميكية. تتميز ذواكر RAM الساكنة بسهولة الاستخدام، لأنها لا تحتاج إلى دارات إضافية مثل دارات الانعاش التي تستخدم مع ذواكر RAM الديناميكية. كما وتمتاز بان استهلاكها للطاقة أقل بكثير من ذواكر RAM الديناميكية. تستخدم ذواكر RAM الديناميكية عدداً أقل من الترانزستورات في بناء خلية الذاكرة وبالتالي فهي تمتاز عن RAM الساكنة بان كثافة تغليفها عالية، وان تكلفتها أقل. ترتب خلايا الذاكرة RAM على هيئة مصفوفة أيضا مثلها مثل ذواكر ROM بحيث يمكن اختيار البيانات المخزنة بموقع ذاكرة عن طريق تطابق خط اختيار صف وخط اختيار عمود. لتوضيح ذلك أكثر ننطلق من مثال محدد وليكن تركيبة ذاكرة RAM بسعة ربع كيلوبت، أي 256bits. نحصل على السعة المذكورة بترتيب مصفوفة الذاكرة على شكل جداء مابين كل من عدد الصفوف وعدد الأعمدة وفق الشكل:

256 x1 bits أو بالشكل 128 x 2 أو 64 x 4 أو 32 x 8 وأخيرا 16 x16 bits

يعبّر الرقم الأول في الجداء عن عدد المواضع في الذاكرة والعدد الثاني يمثل طول الموضع الواحد. يتم التعبير عن ذاكرة RAM بشكل جداء باستثناء حالة واحدة ان كان طول كلمة الذاكرة قياسياً أي 8 bits لا داعي لكتابة الجداء. فمثلاً، بالنسبة لذاكرة RAM بسعة 32 byte يفهم بان السعة الإجمالية للذاكرة 256 bits = 32x8 يتحدّد عدد خطوط العنونة اللازم للذاكرة بعدد مواضع الذاكرة، كما انه باختلاف ترتيب مصفوفة الذاكرة، تختلف التركيبة العامة للذاكرة. فمثلاً، إذا تم اعتماد الترتيب

16 x16 لخلايا التخزين بالمصفوفة، كما هو مبين في الشكل ١٤-٧.



شكل ١٤-٧: المخطط الصندوقي لذاكرة RAM 16 x 16

يعني ذلك وجوب توافر أربعة خطوط عنونة A_0, A_1, A_2, A_3 لتحديد 16 صف بالمصفوفة عن طريق مفكك ترميز 4x16 وأربعة خطوط عنونة A_4, A_5, A_6, A_7 لتحديد عدد أعمدة المصفوفة وهو 16 عن طريق مفكك ترميز 4 x 16 أيضاً، ليصبح اجمالي عدد خطوط العنونة المطلوب ثمانية والتي يمكن بوسطتها عنونة 2^8 أي مايعادل 256 bits وهو نفس ناتج 16 x 16.

١٤-٤-١ تركيبة خلية الذاكرة RAM

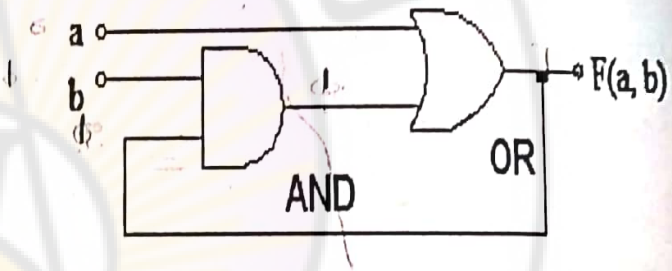
بعد إظهار التركيبة العامة لذاكرة من نوع RAM في الشكل ١٤-٧ (الذي يصلح للتعميم على ذواكر أخرى)، نبّين فيما يلي بغض النظر عن نوع الذاكرة ان كانت استاتيكية أو ديناميكية الفوارق البنيوية بين النوعين الأساسيين للذاكرة RAM

يوجد دارات منطقية لا يتحدّد خرجها بالقيمة الموجودة على الدخل فقط، كما هو الحال بالنسبة للدارات التركيبية، وإنما تأخذ بعين الاعتبار (الحالة) القيمة السابقة للدارة. تتضمن مثل هذه الدارات عناصر تخزين، وتسمى بالدارات المنطقية التتابعية.

9-1 المفهوم العام للدارات التتابعية

للتعريف بالدارات التتابعية، يمكن الانطلاق من مثال بسيط، كالدارة المبينة بالشكل أدناه التي تتكون من عنصرين منطقيين (كل منهما من النوع التركيبي) مربوطين مع بعضهما بوجود رابط عكسي Feed back من الخرج إلى الدخل، ولنحاول إيجاد جدول الحقيقة للتابع المنطقي F(a, b) الذي يمثل خرج الدارة.

a	b	F = 0	F= 1
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	1



يتضح من المخطط المنطقي للدارة، أنه يمكن تحديّد قيمة الخرج F مجرد معرفة التركيبة المطبقة على الدخل باستثناء التركيبة ab = 01، حيث لا يمكن معرفة القيمة الجديدة لـ F من دون معرفة القيمة السابقة لـ F. وبالتالي الخرج F يأخذ من أجل التركيبة ab = 01 على الدخل قيمتين مختلفتين هما:

الأولى: إذا كانت القيمة السابقة لـ F صفراً تصبح القيمة الجديدة لـ F صفراً أيضاً.

الثانية: إذا كانت القيمة السابقة لـ F واحداً تصبح القيمة الجديدة لـ F واحداً أيضاً.

بالتالي جدول الحقيقة لم يعد يصلح لتوصيف مثل هذه الدارة ولابد من استبداله بجدول آخر، لأن خرج الدارة لا يمكن تحديده بالقيمة الموجودة على الدخل. ولتحديد خرج الدارة لابد من معرفة القيمة بالرابطة العكسية. إذا كان الخرج يمثل بعمود يقابل القيمة " 0 " لقيمة الرابطة العكسية، يلزم عمود آخر في الجدول لتحديد قيمة الخرج عندما تكون قيمة الرابطة العكسية " 1 ". لتحديد الشكل العام للجدول الجديد الذي يلحظ هذه الحالة السابقة، باعتماد مصطلح

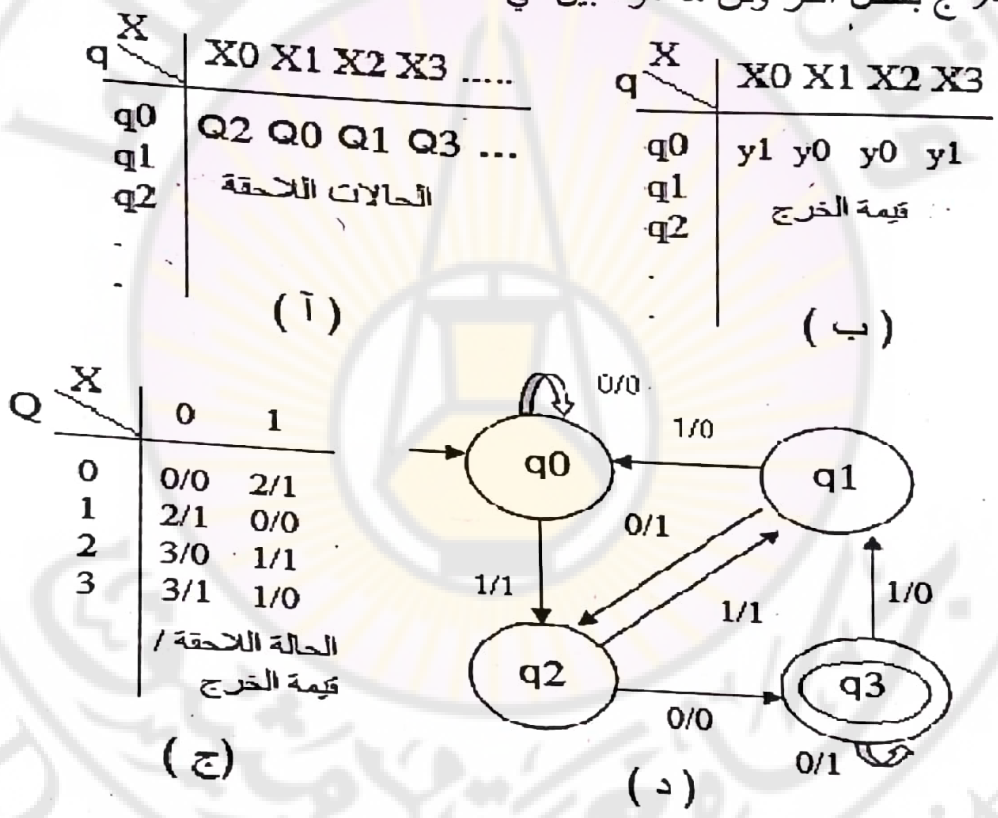
Q_p : القيمة السابقة

أن جميع قيم الدارة من البداية حتى اللحظة الحالية تمثل الحالة State السابقة للدارة (بما في ذلك القيمة الحالية) والقيم المتبقية من لحظة وجودنا وإلى الأمام بالقيمة اللاحقة للدارة، يمكننا تحليل عمل الدارة المنطقية التتابعية وفق التالي: بداية نرمز للقيمة السابقة للدارة بالرمز Q_p (P من Present أو Previous) أو Q_t. وتسمى القيم المتبقية من محور الزمن من لحظة وجودنا وإلى الأمام بالقيمة اللاحقة للدارة ويرمز لها بالرمز Q_n (n من Next) أو Q_{t+1}. ويستخدم أحياناً، بقصد التبسيط الحرف q الصغير للرمز إلى الحالة الحالية أو الحالة السابقة للدارة (عوضاً عن Q_p أو Q_t) وحرف Q الكبير للرمز إلى الحالة التالية أو الحالة اللاحقة للدارة (عوضاً عن Q_n أو Q_{t+1}). وهكذا من الآن وصاعداً، أينما وجد أي من الرموز التالية:Q_p, Q_t, q يعني ذلك رمزاً للحالة السابقة للدارة، بينما أي رمز من الرموز Q_n, Q_{t+1}, Q يعني رمزاً للحالة اللاحقة للدارة. تمثل الحالة السابقة للدارة Q_p مدخلاً غير ظاهر للدارة التتابعية (مدخل إضافي) ويعّبر عن حالة داخلية للدارة Internal state مخزنة بالتغذية الراجعة. بالتالي الدارة أعلاه تتصرف كأن لديها ذاكرة للحالة السابقة للدارة. وهكذا، إذا رمز للتراكيب: 11 10 01 00 الممكنة لمتحولات الدخل a و b بالرموز: X_0, X_1, X_2, X_3 على التتالي، وللقيمتين 0 و 1 للمتحول الداخلي بـ Q_p، يصبح الشكل الجديد لجدول الحقيقة، وفق ما هو مبين أدناه.

External Inputs / Internal States Q	X0 00	X1 01	X2 10	X3 11
Q0	0	1	0	1
Q1	0	1	1	1

يسمى الشكل الجديد لجدول الحقيقة بجدول الحالات الداخلية أو جدول العبور وهو الجدول الذي يعّبر عن تصرفات الدارة، حيث Q_0 تقابل القيمة 0 و Q_1 تقابل القيمة 1. يأخذ جدول العبور أو جدول الانتقال للدارة التتابعية أشكالاً عدة. فإذا رمز للتراكيب الممكنة لمتحولات الدخل لدارة تتابعية (أبجدية الدخل) بالرموز $x_1, x_2, \ldots, x_n$ ولقيم الخرج المقابلة (أبجدية الخرج) بـ $y_1, y_2, \ldots, y_m$ وللحالات الداخلية (السابقة) للدارة بـ $q_1, q_2, \ldots, q_p$، يمكن

توصيف الدارة التتابعية بأربعة أشكال، وفق ما هو مبين في الشكل 1-9. بحيث يمثل الشكل1-9 آ جدول عبور والناتج هو قيمة الحالة اللاحقة (Q_n) للدارة، أما الشكل 1-9 ب، فيمثل جدول عبور والناتج فيه هو قيمة الخرج y للدارة. يمكن أن يعّبر عن الحالة اللاحقة للدارة والخرج في جدول واحد، كما هو مبين في الشكل 1-9 ج لدارة بسيطة تمر بأربع حالات ولها مدخل خارجي واحد X وخرج واحد Y، حيث دوما القيمة التي تكتب فوق الإشارة (/) تدل على رقم الحالة والقيمة التي تظهر تحتها تدل على قيمة الخرج. يُعبّر عن الشكل 1-9 ج بشكل آخر وفق ما هو مبين في الشكل 1-9 د ويسمى بمخطط الحالة.



الشكل1-9: الأشكال الممكنة للتعبير عن الدارة التتابعية

يرمز بمخطط الحالة لكل حالة داخلية للدارة بدائرة (عقدة) وبالتالي الأوجه الموجودة بين العقد تمثل قيم النقل، حيث كل من الدخل والخرج يأخذ قيمة 0 أو 1. يقرأ مخطط الحالة بالنسبة للعقدة الأولى مثلاً على النحو التالي: إذا كانت الدارة بالحالة q_0، وتمّت من الدخل قراءة القيمة 0 تبقى الدارة بالحالة q_0 ويكون الخرج 0 أيضاً، أما إذا كانت الدارة بالحالة q_0 وتم من الدخل قراءة القيمة 1 تنتقل الدارة إلى الحالة q_2 ويكون الخرج 1. وهكذا تتم قراءة

كل حالة من الحالات الثلاث المتبقة. ويشار عادة لحالة البداية (حالة الإقلاع) Initial بسهم، أما الحالة النهائية أو حالة القبول Final or Accept state فيشار إليها بدائرتين ولا يمكن أن يكون في المخطط الواحد أكثر من حالة إقلاع واحدة، أما حالات القبول فيمكن أن تكون واحدة أو بعدد حالات المخطط المقابل لدارة ما. ويمكن فهم الدارة التتابعية على أنها دارة تركيبية بممرات للتغذية الخلفية، كل منها يمثل عنصر تخزين. وبحسب التركيبة العامة لممر التغذية الخلفية أو نظام فعّالية الدارة التتابعية، تصنف هذه الدارات إلى نوعين أساسين هما دارات تتابعية متزامنة بوجود Clock ودارات تتابعية غير متزامنة. أهم ما يميز الدارات التتابعية على مختلف أنواعها عنصر التخزين (الذاكرة)، ومن أكثر عناصر الذاكرة استعمالاً في الدارات التتابعية الإلكترونية هي القلابات Flip-Flops. يمثل القلاب تجهيزه Device قادرة على تخزين المعلومات الثنائية، وهو يمثل شكلاً من أشكال الدارات ثنائية الاستقرار التي تستقر في واحدة من حالتين 0 أو 1 وله مخرجان أحدهما متمم للآخر، ويتمتع بمداخل للتحكم. وفيما يلي نبيّن تركيبة الدارة ثنائية الاستقرار.

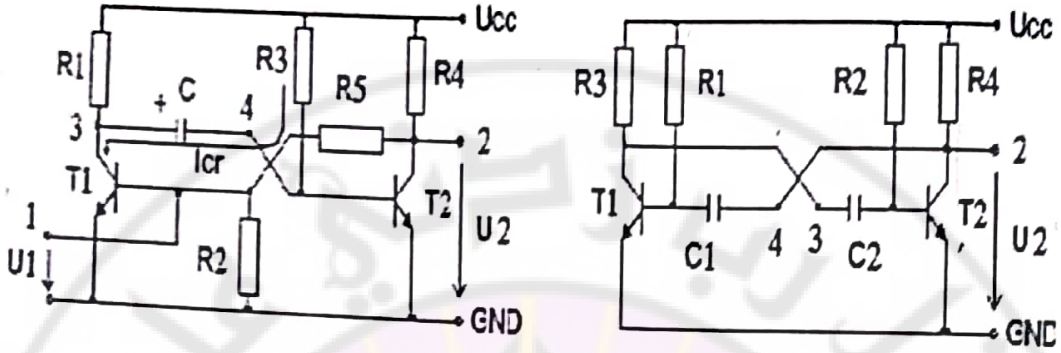
9-2 الدارات ثنائية وأحادية وعديمة الاستقرار

بالإضافة إلى الدارات أحادية وعديمة الاستقرار التي تستخدم بتوليد إشارات بتواترات مختلفة، يمكن أن نصادف دارات تستقر بواحدة من حالتين، تسمى بالدارات ثنائية الاستقرار. وفيما يلي نعرض التركيبة الإلكترونية لكل من هذه الأنواع الثلاثة للدارات، ونركز على الدارات ثنائية الاستقرار بحكم أن القلابات تمثل هذا النوع من الدارات التتابعية.

أولا- الدارات عديمة الاستقرار

تسمى الدارات عديمة الاستقرار أيضاً بالمهتزات Multivibrator، لأنها لا تتمتع بأية وضعية ساكنة بل هي مهتزة باستمرار Astable. الشكل العام لهذا النوع من الدارات، مبين في الشكل 2-9 آ. بالنظر إلى دارة عديم الاستقرار، يلاحظ أنه إذا كان الترانزيستور T_2 مفتوحاً في الحالة الابتدائية وT_1 مغلقاً، يشحن بهذه الحالة المكثف C_2 بكامل جهد المصدر Ucc. ويفتح في اللحظة التالية T_1، وهذا يعني وصل اللوح 3 للمكثف C_2 بالأرض ويصبح بهذا الشكل على اللوح الثاني لهذا المكثف جهد سالب نسبياً، ما يؤدي إلى إغلاق

الترانزيستور T_2 طول المدة اللازمة لتفريغ C_2 عبر R_2، بنفس الفترة يشحن المكثف C_1 ويصبح على لوحه 4 جهد موجب.



(ب) أحادي الاستقرار

(أ) عديم الاستقرار (مهتز)

الشكل2-9: دارة أحادي وعديم الاستقرار

تكتمل كل من العمليتين فتح T_2 وإغلاق T_1 ومن ثم إغلاق T_2 وفتح T_1 وتتكرر باستمرار بفارق زمني محدّد، بحيث المدة بالنسبة للترانزيستور الأول تقارب $t_1 \approx 0.7\ C_1\ R_1$ وبالنسبة للترانزيستور الثاني تقارب $t_2 \approx 0.7C_2R_2$. ولا تستقر الدارة بأي حالة.

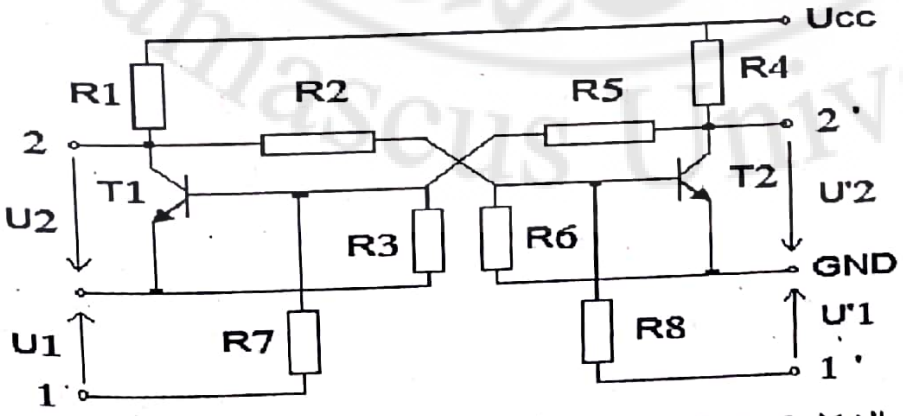
ثانيا– الدارات أحادية الاستقرار

بأسلوب مماثل لما سبق، نجد أن توصيلة دارة أحادية الاستقرار Monostable، مبينة في الشكل 2-9 ب. يبقى الترانزيستور T_2 في حالة الفتح بفعل التيار الدائم من مصدر التغذية عبر R_3، بينما جهد الإشباع للترانزيستور T_2 يحافظ عبر المقاومات R_2 وR_5 على T_1 في حالة القطع. يشحن المكثف C عملياً بكامل جهد المصدر Ucc بقطبية موجبة على مجمع T_1 (عندما يكون T_1 مغلقاً يمر التيار عبر R_1 باتجاه قاعدة T_2)، بالتالي نبضة ضيقة موجبة U_1 المطبقة على قاعدة T_1 تؤدي إلى فتحه. يصبح في هذه اللحظة مجمع T_1 عملياً بجهد صفري، لأن الجهد يهبط عبر T_1 على الأرض، ولذلك المأخذ 3 للمكثف C يصبح وكأنه مؤرض، ويصبح القطب الثاني للمكثف C (أي النقطة 4) بجهد سالب نسبياً (مقارنة مع 3) ويغلق T_2. والجهد الموجب لمجمع T_2 سيحافظ على بقاء T_1 في حالة الفتح. هذه الحالة تستغرق مدة تقارب $t_1 = 0.7\ R_3C$. يشحن خلال هذه المدة المكثف C بالتيار I_{CR} عبر R_3 بقطبية معاكسة وبعد ذلك بجهد موجب صغير على اللوح الأيمن

للمكثف C يفتح T_2 ويغلق T_1 واللوح 3 للمكثف C يشحن بسرعة إلى قيمة الجهد الأساسية الساكنة Ucc وينتج دارة أحادية الاستقرار، مستقرة في وضع (خرج T_2) وغير مستقرة في الوضع الآخر (خرج T_1).

ثالثا- دارة ثنائي الاستقرار

تضم دارات ثنائي الاستقرار Bistable Circuit ترانزستورين يعملان بنظام الإشباع ويشكلان من أجل تنظيم محدّد للروابط المتبادلة ما يسمى بدارة الوصل ثنائية الاستقرار، كما هو مبين في الشكل 3-9، حيث بمجرد وصل التغذية الكهربائية يمكن أن يفتح أي من الترانزيستور ين T_1 أو T_2 (ولكن لأمور تصنيعية سيكون أحدهما أسرع من الآخر ولو بلحظة). بفرض أن T_1 هو الأسرع، معنى ذلك هو الذي سيفتح أولاً، سيهبط الجهد على مجمعه إلى الأرض، وبهذا الشكل عبر R_2 توصل قاعدة T_2 إلى الأرض ويغلق T_2. يمكن أن يبقى كلا الترانزستورين T_1 مفتوحاً وT_2 مغلقاً إلى فترة غير محدودة (أي أن الدارة تستقر بواحدة من حالتين)، حتى يتغير الجهد u1 (أو u'1) إلى الحالة المعاكسة التي فيها يصبح T_2 مفتوحاً وT_1 مغلقاً. تنتقل الدارة من حالة إلى أخرى بتطبيق جهد سالب على الدخل U_1 (أو جهد موجب على U'1). يظهر هذا التغيير على المخارج 2 و2' للدارة ثنائية الاستقرار. تشكل هذه التوصيلة الأساس في تشكيل القلابات التي تكون دوماً مستقرة في واحدة من حالتين: حالة "0" التي تقابل خرج T_1 مثلاً والحالة "1" التي تقابل خرج T_2، حيث إن إحداها متممة للأخرى.



الشكل3-9: التوصيلة الأساسية لدارة ثنائي استقرار

3-9 القلابات Flip-Flops

يمثل القلاب دارة تتابعية ثنائية الاستقرار، ويأخذ بالمراجع العلمية تسميات متعددة، نذكر منها: الرج راج Trigger أو المرجاح أو النطاطة. ويسمى أيضاً بالماسك Latch وتأتي هذه التسمية من قدرة القلاب على إمساك معطيات في داخله. ولتحديد مفهوم القلاب بشكل أكثر دقة، نعود لتصنيف النموذج العام للدارات التتابعية لنموذجين يختلفان عن بعضيهما البعض بطريقة توليد تابع الخرج والنموذجان هما:

- نموذج ميلي Mealy Model والذي يكون فيه الخرج تابعاً لكل من الحالة السابقة والدخل ويوصف الدارة المنطقية التتابعية بالمعادلتين، تابع النقل: $Q_{s,t+1} = \varphi (X_{i,t} , q_{k,t})$ وتابع الخرج: $Y_{j,t} = \Psi(X_{i,t} , q_{k,t})$

- نموذج مور Moore Model وفيه الخرج تابع للحالة الراهنة فقط (السابقة)، أي يتميز بالمعادلتين، تابع النقل $Q_{s,t+1} = \varphi (X_{i,t} , q_{k,t})$ وتابع الخرج: $Y_{j,t} = \Psi(q_{k,t})$ ويستخدم عند التعامل مع هذين النموذجين تعبير آلات الحالة المحدودة Finite State Machines أختصاراً FSM . يتم تزامن المخارج في الدارات التتابعية من نوع مور بنبضة الساعة، أما المخارج في الدارات التتابعية من نوع ميلي فيمكن أن تتغير إذا تغيرت المداخل خلال دورة الساعة، وبالتالي يمكن أن تأخذ المخارج قيم خاطئة بسبب التأخير الزمني بين تغير الدخل والخرج. تمثل القلابات التي قد نصادفها على مختلف أنواعها دارات تتابعية نموذج مور وكل منها يمثل ذاكرة بطول 1bit. يوجد أربعة أنواع للقلابات هي: R-S و J-K وT وD بعضها له مدخل واحد عدا مدخل الـ Clock (مثل T, D) والبعض الآخر له مدخلين (مثل RS, JK) لكن لكل من القلابات الاربع مخرجان أحدهما متمم للآخر ويعطي الحالة اللاحقة للقلاب. تُعدّ القلابات بشكل عام عناصر أساسية في تصميم الأجهزة الإلكترونية الرقمية من عدادات، وسجلات، وغيرها من أجهزة التحكم.، بالتالي يوجد العديد من التطبيقات التي تستخدم القلابات، خاصة التطبيقات التي يحتاج الأمر فيها إلى التبديل بين وضع وآخر أو يلزمها ذاكرة قصيرة، تستخدم القلابات والمواسك بوصفها دارة ثنائية الاستقرار bistable multivibrator كعناصر تخزين، حيث إن القلاب يخزن معطيات بطول 1 bit (أي إحدى حالاته تمثل 1 والحالة الثانية تمثل 0).

وفيما يلي نتعرف على مفهوم كل من القلابات الأساسية من عشرة أوجه هي:

1- الرمز المنطقي Logic symbol

2- المخطط المنطقي Logic diagram

3- جدول الحالة Function (Status) table

4- التابع المنطقي Logic function

5- جدول التحريض Excitation table

6- القلاب المتزامن Synchronous F.F

7- القلاب القائد- مقاد Master- Slave F.F

8- القلاب بمداخل مباشرة F.F with direct inputs

9- المخطط الزمني Timing diagram

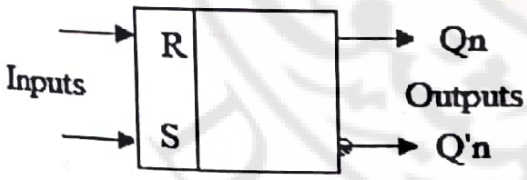
10- الاستخدام Application

9-3-1 القلاب R-S

نتناول فيما يلي القلاب RS بوصفة دارة منطقية من النوع التتابعي من الأوجه العشرة المذكورة أعلاه، نظراً لإمكانية تعميم كثير من النقاط لهذا القلاب على القلابات الأخرى.

أولاً- الرمز المنطقي Logic symbol

يرمز للقلاب R-S بالرمز المبين جانباً



بحيث يظهر بالرمز المنطقي للقلاب المدخلان:

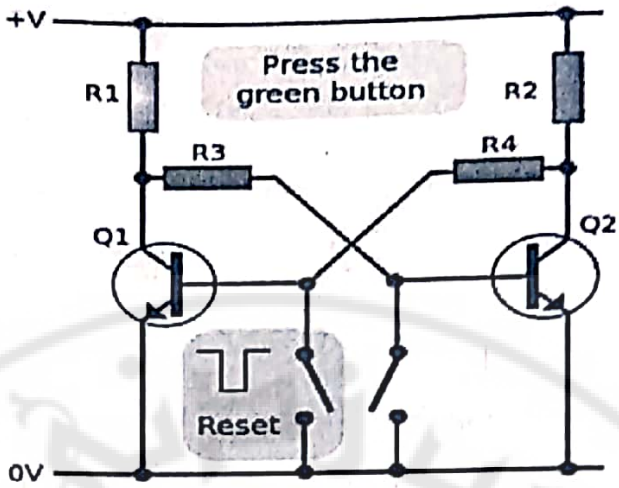
R – الذي يمثل مدخلاً للتصفير Reset.

S- الذي يمثل مدخلاً للوضع (أو التعبئة) Set.

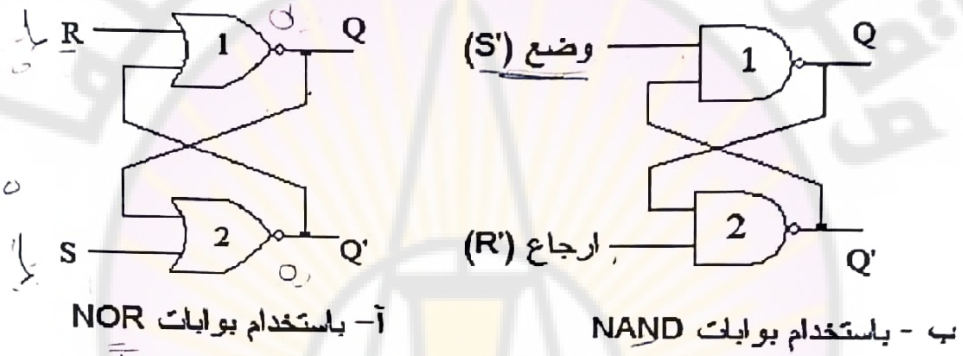
والمخرج Q_n الذي يمثل الحالة اللاحقة للقلاب ومتممه Q'_n .

ثانياً- المخطط المنطقي Logic diagram

يمكن تحويل المخطط الالكتروني بالشكل المبين أدناه والذي يمثل قلاب (من أجل قيم تقريبية للمكونات $R1, R2 = 1\ k\Omega$ $R3, R4 = 10\ k\Omega$) SR latch إلى بوابتين من نوع NOR أو بوابتي NAND موصلتين مع بعضهما بشكل متصالب، بحيث يصبح المخطط



المنطقي للقلاب RS وفق ما هو مبين في الشكل 4-9.



الشكل4-9: المخطط المنطقي للقلاب RS

يضم المخطط المنطقي جزءاً تركيبياً وخطاً راجعاً من الخرج إلى الدخل ليمثل جزء الذاكرة .

ثالثاً- جدول الحالة Status table

يختلف جدول الحالة أو الجدول الوظيفي للقلاب باختلاف العنصر الأساسي المكون له، خاصة وأن كلاً من بوابتي NAND و NOR هي متممة للأخرى. يُظهر المخطط المنطقي في الشكل 4-9 آ مثلاً أن القلاب RS يمثل ثنائي استقرار يكون فيه المخرج Q دائماً من البوابة التي أحد مداخلها R (NOR1). لتحليل عمل القلاب المشكل ببوابات NOR، لابد من الانطلاق من جدول الحقيقة للبوابة NOR والأخذ بعين الاعتبار الحالة السابقة للدارة. يوجد بالنسبة للقلاب RS، بالإضافة إلى المدخلين R و S مدخل ثالث داخلي غير ظاهر في المخطط المنطقي يمثل المتحول Q_p (الحالة السابقة للقلاب). وبالتالي الجدول الوظيفي (أو جدول العبور) للقلاب، سيضم ثلاثة متحولات منطقية

باعتبارها دخلاً للدارة ومخرجين أحدهما متمم للآخر(نكتفي باعتماد المخرج Q_n فقط). يصبح الشكل العام للجدول الوظيفي للقلاب RS المشكل من بوابات NOR، وفق ما هو مبين في الجدول 1-9. بالطبع يمكن بأسلوب مماثل تشكيل الجدول الوظيفي للقلاب RS من البوابات NAND. يتضح من المخطط المنطقي للقلاب RS في الشكل 4-9آ أنه من أجل التركيبة: $R = S = 0$ على الدخل، توجد حالتان:

الأولى: من أجل القيمة $Q_p = 0$ ، وهذا يعني أن Q'_p سيكون واحداً، وبالتالي خرج البوابة NOR رقم 1 سيكون صفراً، وهو الخرج النهائي للقلاب من أجل هذه التركيبة.

الثانية: من أجل القيمة $Q_p = 1$ وبهذه الحالة نجد بأسلوب مماثل إن خرج البوابة NOR1 هو واحد، وهو الخرج النهائي للقلاب من أجل هذه التركيبة.

الجدول1-9: جدول الوظائف للقلاب RS

Inputs			Output	Function
R	S	Q_p	Q_n	الوظيفة
0	0	0	0	Memory
0	0	1	1	حالة الذاكرة
0	1	0	1	Set
0	1	1	1	حالة التعبئة
1	0	0	0	Reset
1	0	1	0	حالة التصفير
1	1	0	d	Don't Care
1	1	1	d	حالة ممنوعة

يمكن متابعة خرج القلاب من أجل بقية التراكيب الممكنة على الدخل وفق ما هو مبيّن بالجدول الوظيفي. وبالتمعّن في الجدول الوظيفي الناتج للقلاب RS وتحديداً التركيبة الأولى الممكنة للمداخل الخارجية، أي: $R = S = 0$، يتضح أن القلاب يتصرف وكأنه ذاكرة، لأن الحالة السابقة للدارة تبقى مطابقة للحالة اللاحقة من دون أي تغيير. أما عندما تكون $S = 1$ و $R = 0$ فإن القلاب يكون بحالة Set، لأن الخرج Q_n سيكون واحداً بغض النظر عن

قيمة الحالة السابقة للقلاب. وبالعكس عندما تكون S = 0 و R = 1 فإن القلاب يكون بحالة Reset، لأن الخرج Q_n سيكون صفراً. التركيبة S = R = 1 غير مسموح بها، لأنها تعني للقلاب أن يكون بنفس اللحظة بكل من حالتي التعبئة والتصفير وهذا ليس له معنى (علاوة على أن قيمة كل من Q_n و Q'_n ستكون 0 وهذا غير ممكن)، ويعبّر عن ذلك في الجدول الوظيفي بحالة d. وهنا نشير إلى أنه إذا كان القلاب RS مبنياً بالبوابات NAND فإن الجدول الوظيفي يتغير، بحيث التركيبة S = R = 1 تمثل حالة الذاكرة والتركيبة S = R = 0 تصبح من دون معنى، ويجب محاولة تفادي وضع هذه التركيبة على دخل القلاب، لأن الاهتزازات الداخلية التي تنتج من تطبيق مثل هذه التركيبة يمكن أن تدمر الدارة القلابة. ولإبقاء كل من R و S بمعناه، يبدل في المخطط المنطقي للقلاب RS المشكل بالبوابات NAND أيضاً موضع المداخل، بحيث يصبح S أحد مداخل بوابات NAND التي خرجها Q_n و R احد مداخل البوابة الثانية، كما هو مبين في الشكل 4-9 ب، وتبدل قيم المداخل، لأن بوابات NOR تتطلب لتغير حالتها تطبيق إشارة "1"، بينما NAND تتطلب تطبيق "0"، بمعنى أنه بدلاً من S مثلاً، يوضع 'S ، ويشار إلى القلاب بالاسم 'R'-S. وتبقى حالتي الوضع والإرجاع على حالها، كما في القلاب المشكل من البوابات NOR، بمعنى إذا كانت S = 1 ، أي 'S = 0 سيكون القلاب بحالة الوضع.

رابعاً- التابع المنطقي Logic function

باعتماد القلاب RS المبني من بوابات NOR يمكن تمثيل الجدول الوظيفي للقلاب بمخطط كارنوف، لكتابة التابع المنطقي وفق ما هو مبين في الشكل أدناه.

Q_p \ RS	00	01	11	10
0	0	1	d	0
1	1	1	d	0

يتضح من مخطط كارنوف بوجود حلقتين، أن المعادلة المميزة للقلاب RS هي:

$$Q_n = S + R'Q_p$$

خامساً- جدول التحريض Excitation table

يقصد بجدول التحريض (أو جدول التهيج) بالنسبة لأي قلاب تحديّد القيم المناسبة على دخل القلاب لتأمين عملية الانتقال من وضعية ما إلى أخرى. بالتالي هذا الجدول هو الأساس الذي يستخدم عند تصميم الدارات التتابعية. وبالنظر إلى أن هذا الجدول بالنسبة لكل قلاب يتناول كافة التراكيب الممكنة مابين الحالة السابقة Q_p والحالة اللاحقة Q_n للقلاب، فإن كل جدول من هذا النوع لكل قلاب سيتكون من أربعة أسطر، يحدّد فيه كل سطر وضعية محدّدة مابين الحالتين السابقة واللاحقة. وبالنسبة للقلاب RS تحديداً فإن جدول التحريض يبدو كما هو مبين في الجدول 2-9.

الجدول2-9: جدول التحريض للقلاب RS

Q_p	Q_n	R	S
0	0	d	0
0	1	0	1
1	0	1	0
1	1	0	d

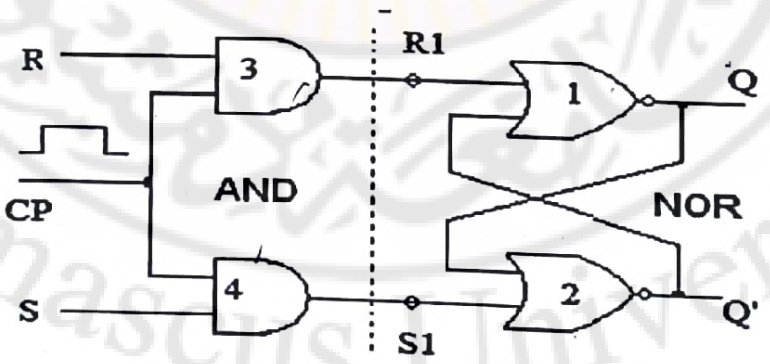
$S = Q_n$

$R = Q_n'$

يقرأ الجدول على النحو التالي: السطر الأول- إذا كانت الحالة السابقة للقلاب $Q_p = 0$ وأردنا أن يبقى القلاب بالحالة صفر، أي $Q_n = 0$، يطلب تحديد القيم التي يجب وضعها على كل من R و S؟ للإجابة: نعلم إذا كانت قيم المداخل $R = S = 0$ سيكون الخرج المقابل هو 0، لأن هذه التركيبة على المداخل تعني أن القلاب هو بحالة ذاكرة أي أنه كان بالصفر وسيبقى بالصفر. يوجد إمكانية أخرى للوصول إلى نفس النتيجة وهي إذا كانت $R = 1$ و $S = 0$، لأن هذه التركيبة على الدخل تعني حالة Reset أي جعل الخرج صفراً بغض النظر عن الحالة السابقة. هذا يعني أنه لانتقال القلاب من الوضعية 0 إلى الوضعية 0 يوجد تركيبتان ممكنتان على الدخل في كل منهما $S = 0$، أما R فيمكن أن تأخذ أياً من القيمتين 0 أو 1 أي d. وبالنسبة للسطر الثاني يوجد إمكانية وحيدة لانتقال القلاب من الحالة 0 إلى الحالة 1 هي التركيبة $S = 1$ و $R = 0$ على الدخل التي تعني حالة الوضع Set. وهكذا بنفس الأسلوب يمكن تعبئة الأسطر المتبقية من الجدول.

سادساً- القلاب المتزامن Synchronous R-S F.F

خلال التوصيف السابق للقلاب RS، لم يستخدم مفهوم الساعة CLK لقيادة عمل القلاب، الذي كان يقاد بترتيب الأحداث Events، بمعنى أن الحالة اللاحقة في لحظة ما كانت تحدّد بقيمة الحالة السابقة Q_p وقيم المداخل الخارجية R و S من دون وجود نبضات للساعة Clock Pulse (CP). يسمى مثل هذا القلاب المقاد بالأحداث لا بالزمن، قلاباً غير متزامن Asynchronous R-S Flip Flop. ويسمى القلاب الذي يقاد بالنبضة الساعية بالقلاب المتزامن Synchronous R-S Flip Flop. يمكن اختصار الفرق بين كل من القلاب المتزامن والقلاب غير المتزامن بالتالي: يعتمد سلوك القلاب غير المتزامن على الترتيب الذي تتغير فيه إشارات الدخل، أما القلاب المتزامن فيمثل نظاماً يمكن تعين سلوكه من معرفة إشاراته في لحظات متقطعة زمنياً. يتحقق التزامن بوساطة تجهيزه إضافية، تدعى بمولد نبضات الساعة Clock generator الذي ينتج قطاراً من نبضات ساعة دورية. تتغير متحولات الحالة في القلابات المتزامنة فقط عندما تقدح القلابات بنبضة ساعية. يتشكل القلاب المتزامن من القلاب غير المتزامن بعد إضافة بوابتي AND كل منهما بمدخلين تربط على مداخل دارة القلاب الأساسية، كما هو مبين في الشكل 5-9 (البوابات 3 و 4) للتحكم بالقلاب وحمايته من التغيرات الطارئة.



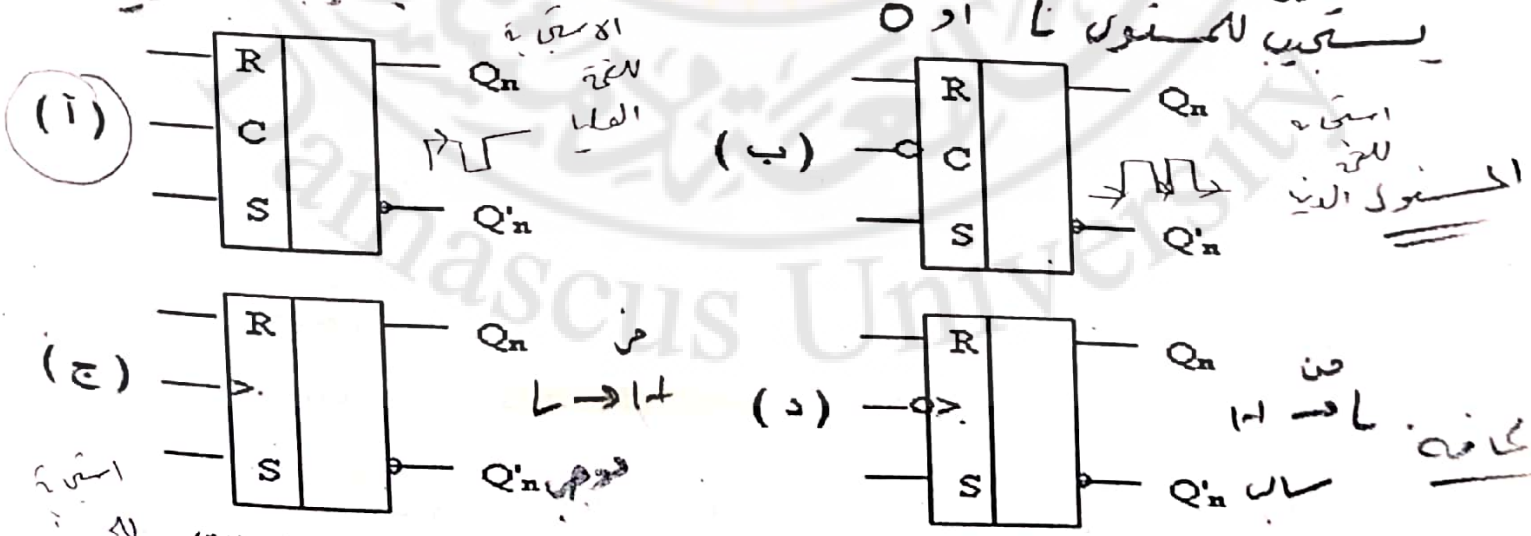
الشكل5-9: المخطط المنطقي لقلاب RS المتزامن

يطبق المدخل R على أحد مداخل البوابتين AND (تحديداً AND3) و S على أحد مداخل البوابة الثانية AND4 ويوصل المدخل المتبقي من كلتا البوابتين AND بخرج مولد نبضات ساعية خارجي للتحكم بفتح البوابتين عند الطلب (وكأنه مفتاح تحكم). طالما إن

نبضة الساعة غير موجودة CP = 0، يكون خرج بوابتي AND مساوياً للصفر بغض النظر عن القيم الموجودة على كل من المداخل R و S، وبالتالي القلاب سيكون في وضعية ذاكرة للمرحلة السابقة لوجود صفر على كل من المدخلين R_1 و S_1 ويبقى بهذه الوضعية إلى حين ورود نبضة ساعية على المدخل CP. بمجرد أن يصبح CP = 1 تحرر كل من البوابتين AND البيانات الموجودة على مداخل القلاب الخارجية R و S كما هي،

سابعاً– المخطط الزمني Timing diagram

بعد أن تعرفنا على الشكل العام لكل من القلاب المتزامن والقلاب غير المتزامن، نشير إلى أن المفهوم العام للمعادلة المميزة للقلاب RS لا يتغير في كلا النوعين من القلابات على الرغم من أن القلاب المتزامن يمتاز بوجود مدخل إضافي خاص لتطبيق النبضات الساعية، كما هو مبين في الشكل 6-9. ولا يقدم المدخل الجديد بالنسبة للقلاب المتزامن أي معلومات من حيث تحديد تغير الحالة، إنما يزامن تغيير الحالة، وبالتالي هو يمثل مفتاحاً بالاسم " ساعة CP ". بمعنى آخر، إن أي تغير غير مرغوب على مداخل القلاب غير المتزامن يظهر على الخرج على عكس القلاب المتزامن الذي لا يتغير خرجه إلا عندما يقدح باستخدام مفتاح خارجي يمثل مدخلاً للتحكم (وهكذا تتم حماية خرج الدارة من أثر أي تغير غير مرغوب به على الدخل).



الشكل6-9: الرموز المعتمدة للاستجابة للساعة (المستوى والحافة)

تتبدل حالة القلاب المتزامن عند حدوث تغير في مدخل التحكم ويدعى هذا التغير اللحظي بالقدح. ويسبب القدح انتقال للقلاب من حالة إلى أخرى. وبشكل عام تقدح القلابات:

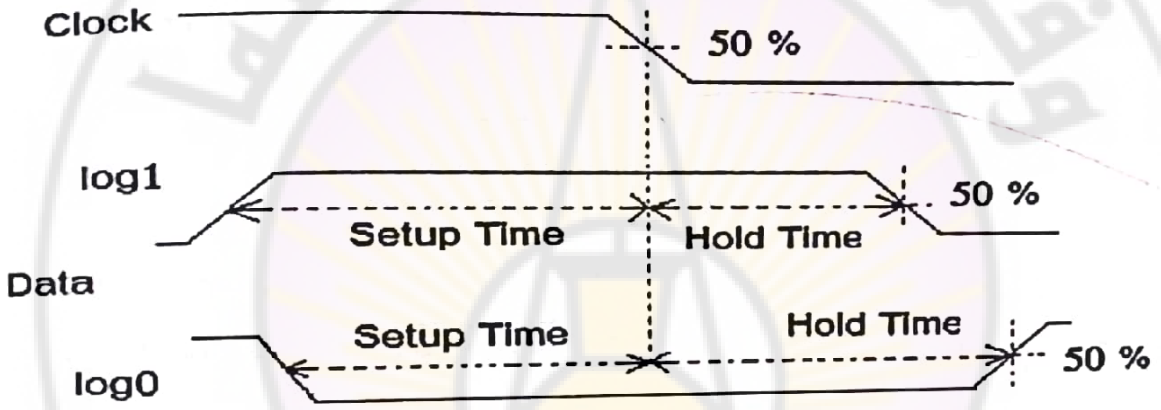
- إما على المستوى Level (المنخفض 0 أو العالي 1)
- أو على الجبهة Edge (الصاعدة أو الهابطة). القلاب R-S الذي يقدح على المستوى مثلاً، يقدح كل مرة تنتقل فيه النبضة إلى المستوى المحدّد على مدخل التحكم. وتنتقل التغيرات التي تحصل على مداخل القلاب إلى مخرجه طالما مستوى النبضة على مدخل التحكم محافظاً على مستواه. يرمز للقلاب الذي يستجيب للمستوى " High ~ 1" ، بالرمز المبين في الشكل 6-9 آ. وللقلاب الذي يستجيب للمستوى " Low ~ 0" بالرمز المبين بالشكل 6-9 ب إذا كان خرج القلاب الذي يستجيب للمستوى موصولاً إلى مداخل قلابات أخرى من خلال الجزء التركيبي للدارة التتابعية، يظهر بعض المشاكل، لأن أي تغير لقيم مداخل القلابات في الوقت الذي لا تزال فيه النبضة الساعية موجودة قد يحدث حالة جديدة على الخرج بسبب استجابة القلاب لها. والنتيجة هي حالة لا يمكن التنبؤ بها نظراً لأن حالة القلاب قد تستمر بالتغير مادامت نبضة الساعة في المستوى الفعّال. وبسبب ذلك لا يمكن تطبيق مخرج القلاب بصورة مباشرة أو عبر الجزء التركيبي على مدخل قلاب آخر (أو مدخل القلاب نفسه) عندما تقدح جميع القلابات المكونة للدارة الواحدة باستخدام مصدر ساعة مشترك. أما بالنسبة للقلابات التي تقدح على الجبهة، أي أثناء انتقال الإشارة من مستوى إلى آخر (حافة الإشارة) فنميز بين شكلين للقلاب:

الأول: يرمز له بالرمز المبين في الشكل 6-9 ج، ويمثل انتقال إشارة التحكم من المستوى L إلى المستوى H، ويعرف هذا الانتقال بالانتقال الموجب Positive transition أو الانتقال بالحافة الموجبة Positive edge.

الثاني: يرمز له بالرمز المبين في الشكل 6-9 د ويمثل انتقال إشارة التحكم من المستوى H إلى L ويعرف بالانتقال السالب Negative transition أو الانتقال بالحافة السالبة Negative edge.

يجب عند التعامل مع القلابات التي تستجيب لوجه النبضة أن تؤخذ بعين الاعتبار أزمنة التهيئة Setup time أو الوقوف Hold time المطلوبة. يمكن من أجل قلاب يستجيب

للوجه الهابط للنبضة الساعية مثلاً، بوجود قيمة 0 أو 1 على مدخل المعطيات للقلاب
إظهار المقصود بكل من زمني التهيئة والوقوف من المخطط الزمني المبين في الشكل 7-9.
تقاس الأزمنة في الدارات التي تستجيب لوجه النبضة عادة عند قيمة الـ 50% من الإشارة
أي بالنقطة التي تتوسط أطراف النبضة وبالتالي يمكننا كما هو واضح من المخطط
تعريف زمن التهيئة، بأنه الفترة الزمنية من بداية تطبيق نبضة المعلومات إلى لحظة تبدل
مستوى النبضة الساعية من المستوى العالي إلى المستوى المنخفض. أما زمن الإمساك،
فيمثل الفترة الزمنية بين لحظة تبديل مستوى النبضة الساعية وبين نهاية نبضة المعلومات
(الزمن المتبقي من نبضة المعلومات).



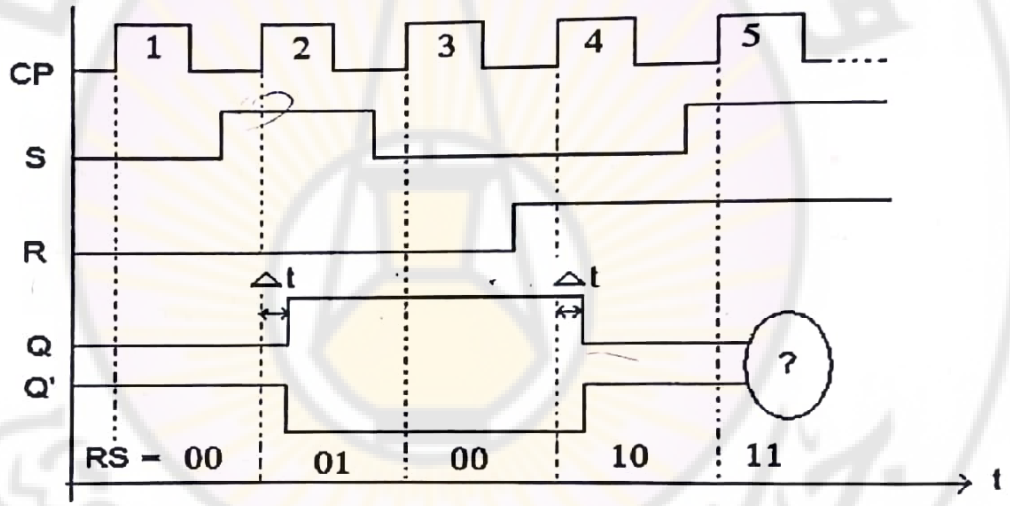
الشكل7-9: أزمنة التهيئة والوقوف

يرد بالنشرات الفنية للشركات الصانعة للقلابات بخصوص الأزمنة عادة ثلاث بارامترات:

- الزمن الاصغري الذي يدعى بزمن التهيئة، ويجب أن يحافظ فيه الدخل على قيمة ثابتة قبل حدوث الانتقال بالنبضة.
- الزمن الاصغري الذي يدعى بزمن الإمساك الذي يجب فيه أن تستمر القيمة على الدخل من دون تغير بعد تطبيق الانتقال الموجب للساعة.
- زمن تأخير الانتشار Propagation delay time للقلاب والذي يعّرف بالمدة التي تفصل بين حافة القدح ولحظة استقرار الخرج عند الحالة الجديدة.

يمكن التعّبير عن القلاب بمخطط زمني يلحظ تغيرات الحالات الداخلية، ويظهر تأخير الإشارة. فمن أجل قلاب RS متزامن، يستجيب للوجه الصاعد للنبضة مثلاً، يمكن بوجود نبضات ساعية إظهار المخطط الزمني للقلاب، وفق ما هو مبين في الشكل 8-9.

يلاحظ من المخطط الزمني للقلاب بأنه من أجل قيم عشوائية على الدخل، يستجيب القلاب للقيم على الموجودة على المداخل الخارجية، بعد الأخذ بعين الاعتبار بكل نبضة ساعية الحالة السابقة للدارة. وتتم الاستجابة فقط بوجود النبضات الساعية، ويستمر القلاب بأي حالة يدخل بها إلى حين ظهور النبضة التالية. يظهر من الشكل أدناه أيضاً أنه بعد ظهور النبضة الخامسة بفترة زمنية مساوية لزمن تأخير القلاب RS (أي Δt) ، وكيف أن الخرج Q_n يدخل في حالة عدم تحديد، مما يبّين أن القلاب دخل بوضعية غير صحيحة بسبب وجود قيمة الواحد على كل من المدخلين R و S، وبالتالي بعد اختفاء النبضة الخامسة يصعب تحديد القيمة التي سيستقر بها القلاب.



الشكل 8-9: المخطط الزمني للقلاب RS

ثامناً- قلاب القائد – مقاد Master - Slave F.F

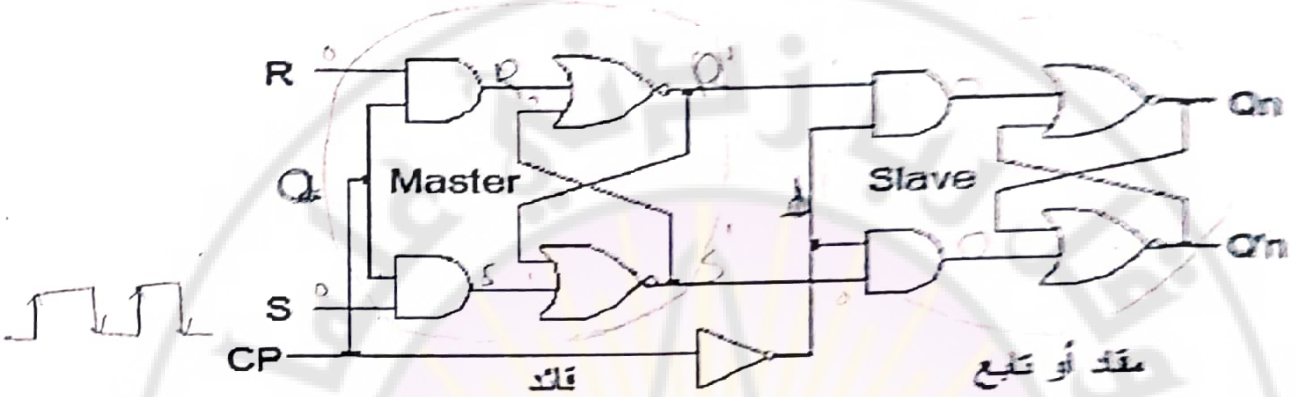
تستخدم الدارات التتابعية قلابات لكل منها زمن تأخير يختلف من قلاب إلى آخر. وللساعة القادمة من مولد نبضات خارجي مواصفات تختلف من مولد إلى آخر بحسب عرض النبضة Pulse Width والفترة الزمنية ما بين ظهور نبضة وأخرى. وبالتالي يوجد للتوقيت في الدارات التتابعية مشاكل متعددة. من أولى محظورات التوقيت Timing problems أن نبضة القدح يجب أن يكون لها عرض أصغري معين، وأن الفترة بين التغييرات المتتالية لمتحول واحد يجب آلاّ تقل عن زمن التأخير الأساسي للعناصر المكونة للدارة. فإذا كان زمن التأخير لبوابة منطقية هو 20ns مثلاً، فإنه بالتأكيد لا يمكن أن تستجيب بشكل صحيح

إلى تغيرات إشارة خلال 10ns فقط، ويجب على المصمم أن يختار نوعية من الدارات المنطقية ذات سرعة كافية، بحيث تتجاوب مع سرعة الإشارات المتوقع تطبيقها على الدارة. حتى يعمل القلاب بشكل صحيح ومضبوط، يجب آلاّ تتغير قيم مداخله قبل اختفاء النبضة الساعية، لأن وجود قيم جديدة على مداخل R و S مثلاً قبل اختفاء النبضة الساعية قد يؤدي إلى عدم استقرار القلاب في حالة محددة. ويكون عرض نبضة القدح الاصغري عادة من المواصفات الأساسية التي تعطى من الصانع ويجب الانتباه له. من مشاكل التوقيت الأخرى التي يمكن مصادفتها أيضاً أنه إذا أتت نبضة ثانية مباشرة بعد النبضة الأولى بوجود قيم على مداخل الدارة، فإن الجبهة الأمامية للنبضة قد تحدث تغييراً تالياً غير مرغوب به بنفس المتحول. لذا لا يسمح بأن يحصل ذلك، إذ لابد من وجود حد أصغري للفترة بين نبضات القدح، وغالباً ما يعطى على شكل تردد أعظمي مثل 10MHz، وهذا يقابل 100 ns كفترة أصغريه بين نبضات الإطلاق، وبشكل عام يكون الزمن الاصغري الفاصل بين النبضات أكبر بكثير من العرض الاصغري للنبضة. فمثلاً، في قلاب يعمل على 10MHz يكون عرض النبضة الأصغري 20 ns، وهذا يساوي 1/5 الفترة الفاصلة بين النبضات. ونظراً لصعوبة تحديد عرض أصغري وأعظمي للنبضة الساعة في معظم الدارات المتكاملة، بسبب التفاوت في أزمنة التأخير للبوابات المكونة لها، يمكن تجاوز مثل هذه الصعوبات والتخلص من تأثير مشاكل التزامن بما في ذلك زمن مكوث النبضة في دارة القلاب باستخدام أشكال أخرى لدارات القلابات العادية من أبرزها: – دارة القائد المقاد Master – Slave

– القلاب الطرفي Edge Triggered F.F

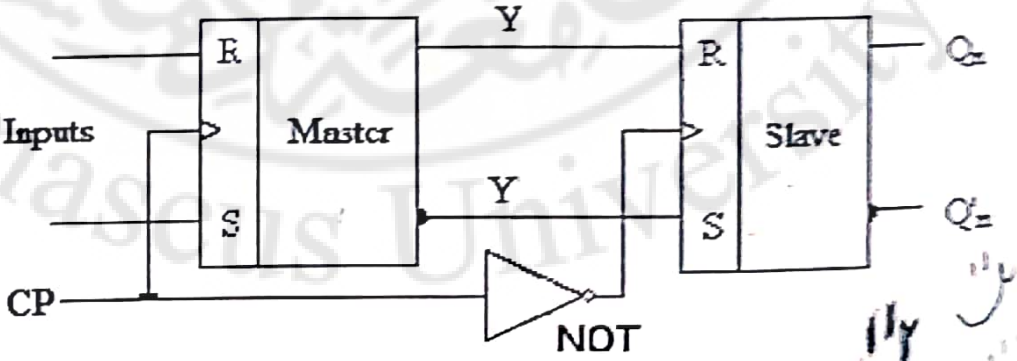
نبّين فيما يلي المفهوم العام لدارة القلاب من نوع قائد– مقاد، على أن نعود لإظهار المفهوم العام للقلاب الذي يقدح على الجبهة (الطرفي) لاحقاً. ويمكن تجاوز مشاكل التزامن المتعلقة بعرض النبضة Pulse width وزمن تأخير الإشارة Delay time باستخدام قلابات أكثر تعقيداً، هي قلابات القائد – مقاد، وهي قلابات تستخدم عادة مبدأ القيادة بطورين أي مبدأ القائد أولاً ومن ثم المقاد ثانياً. نميز بين نوعين من هذه الدارات التتابعية: دارات قائد – مقاد متزامنة وقائد – مقاد غير متزامنة. يتم بناء دارة قلاب قائد– مقاد متزامن من قلابين موصولين بنفس الساعة، بحيث نبضة الساعة في القلاب الأول (القائد) تكون

عكس نبضة الساعة في القلاب الثاني (المقاد) وتعمل الدارة الناتجة على نقل البيانات على مرحلتين متتاليتين لعمليتها من أي تغير غير مرغوب به. الشكل العام لدارة قائد– مقاد، مبينة في الشكل 9-9. يجب لضبط عملية التزامن بشكل صحيح ألاّ تظهر قيم جديدة على RS أثناء وجود نبضة الساعة، وإنما يجب تغيير قيم المداخل قبل ظهور المستوى العالي للنبضة



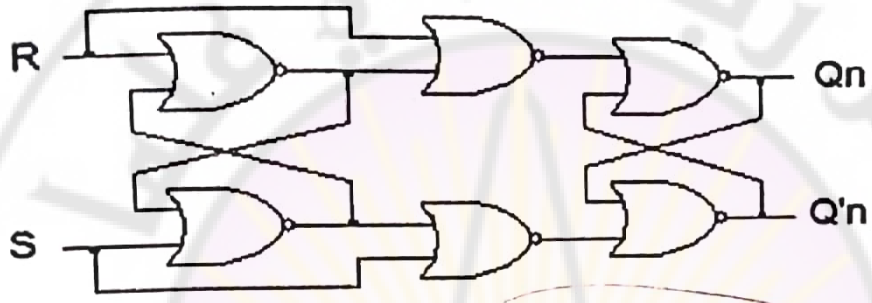
الشكل9-9: المخطط المنطقي لقلاب RS نوع قائد – مقاد

يمكن التعبير عن دارة قلاب من نوع قائد/ مقاد صندوقياً، وفق ما هو مبين بالشكل 10-9، حيث إن قيم المخارج للقلاب القائد لا تنتقل إلى قلاب المقاد إلا بعد اختفاء النبضة، بمعنى أن زمن مكوث النبضة لا يؤثر على عمل الدارة. يتم بالمرحلة الأولى نقل البيانات من مداخل القائد RS Master إلى المخارج Y,Y' عندما تكون نبضة الساعة موجودة، وفي نفس هذه المرحلة تكون نبضة الساعة على مدخل المقاد تساوي صفراً، وهذا يعني أن دارة قلاب المقاد لا تغير حالتها وتمنع مرور البيانات الموجودة على مخارج القائد.



الشكل10-9: المخطط الصندوقي للقلاب المتزامن RS

وفي المرحلة الثانية عندما تختفي نبضة الساعة، يسمح المقاد بمرور البيانات. وهكذا تنتقل البيانات الموجودة على مخارج القائد إلى مخارج المقاد في الوقت الذي يكون فيه القائد بحالة راحة. ويمكن أيضاً بناء دارة لقلاب من نوع قائد – مقاد، يقاد بالأحداث لابالنبضة الساعية، أي غير متزامن، وفق ما هو مبين في الشكل 11-9، حيث يتضح من دارة القائد – مقاد غير المتزامن بأن الجزء الثاني من المخطط المنطقي، أي زوج البوابات NOR المتصالبة للمقاد لا يمكن أن يظهر على مداخلها سوى حالتين هما:



الشكل11-9: المخطط المنطقي للقلاب RS غير المتزامن

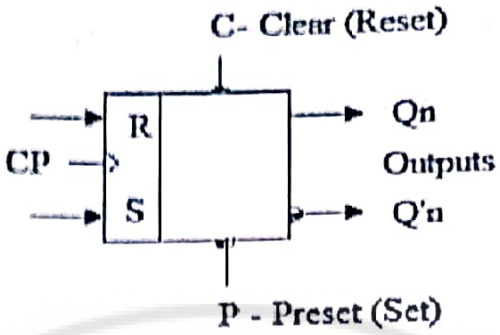
$S = 0, R = 1$ و $S = 1, R = 0$. كما أن الحالتين المتبقيتين: $R = S =$ و $R = S = 1$ غير ممكنتين لأن هذه المداخل تمثل مخارج زوج البوابات المتصالب للقائد. وبما أنه لا يمكن أن يكون $Q_n = Q'_n$ يختصر جدول العبور من أجل جميع قيم الوضع والإرجاع إلى قيمتين تقابلان الحالات التي يكون فيها:

$R = S = 0$ و $Q_p = 0$ من أجل التصفير (الإرجاع) .

$R = S = 0$ و $Q_p = 1$ من أجل التعيير (التهيئة أو الوضع).

تاسعاً– القلاب بالمداخل المباشرة F.F with direct inputs

عند تصنيع القلابات على شكل دارات متكاملة، غالباً ما يجري تعليب القلاب (أو عدد من القلابات) في شريحة واحدة بعد إضافة مداخل خاصة للتحكم بحالات القلاب، منها: مدخل خاص بوظيفة وضع القلاب في حالة الواحد Set بغض النظر عن قيم مداخل القلاب الأساسية، يطلق عليه اسم Preset input. ومدخل خاص آخر بوظيفة وضع القلاب في حالة الصفر بغض النظر عن قيم مداخل القلاب، يسمى Clear input. يظهر الشكل 12-9 الرمز العام لقلاب RS بوجود هذين المدخلين.



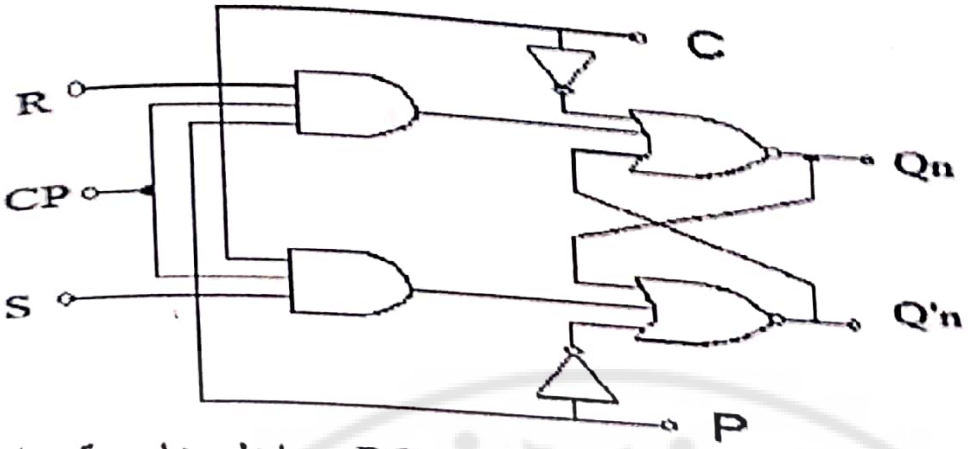
الشكل12-9: قلاب RS بمداخل خارجية مباشرة

يحتوي القلاب الواحد على أحد المدخلين أو كليهما معاً. يستخدم المدخلان الإضافيان بالقلاب لتثبيت المخارج على حالة منطقية معينة بغض النظر عن قيم R و S . يكون المدخلان الإضافيان للقلاب RS فعّالين بالمستوى " 0" أو بالمستوى "1" بحسب نوع الدارة. بالطبع لا يجوز أن يفعّل المدخلين (بالصفر) معاً. وبالتالي جدول الوظائف لهذا النوع من القلابات يبدو كما في الجدول 3-9.

الجدول 3-9: جدول الوظائف للقلاب RS بمداخل خارجية مباشرة

Direct Inputs		Normal Inputs		Output
C	P	R	S	Q_n
0	0	d	d	غير مسموح
0	1	d	d	0
1	0	d	d	1
1	1	0	0	Q_p
		0	1	1
		1	0	0
		1	1	غير مسموح

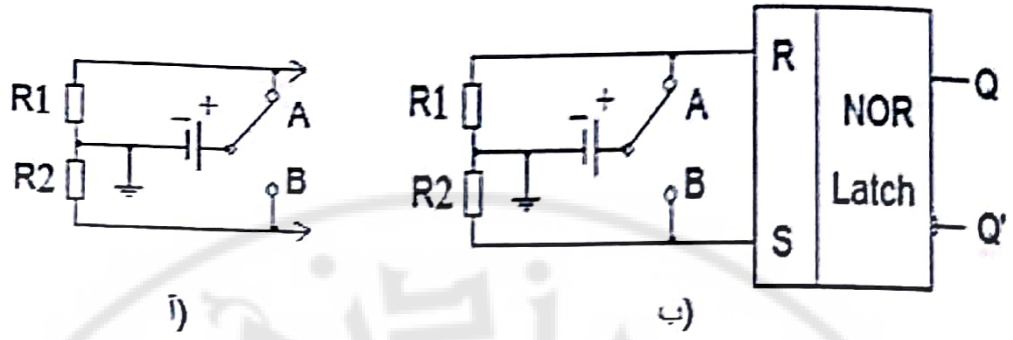
المخطط المنطقي الذي يمكن أن يحقق جدول الوظائف أعلاه، يمكن أن يكون كما هو مبين في الشكل 13-9. يتضح من المخطط المنطقي، إذا كان P = C = 1 (غير فعّالين) سيكون خرج كل من العاكسين صفراً، وبالتالي خرج كل من بوابتي NOR المشكلتين للقلاب يصبح مرتبطاً بالقيم الموجودة على كل من المدخلين R و S.



الشكل13-9: المخطط المنطقي لقلاب RS بمداخل خارجية مباشرة

عاشراً- تطبيق باستخدام القلاب RS

تمثل القلابات بما فيها القلاب RS كما سنجد لاحقاً العناصر الرئيسية الداخلة في بنية الدارات التتابعية، وهذا يجعلها تنفرد بالعديد من التطبيقات الخاصة. نعرض فيما يلي من التطبيقات العديدة للقلاب RS تطبيقاً واحداً، وهو استخدام القلاب RS لحذف ارتداد مفتاح ميكانيكي Switch bounce. يمكن تشكيل مفتاح بسيط لتوليد معلومات دخل ثنائية لنظام رقمي يدوياً (كما هو الحال في المخبر) باستخدام مفتاح ميكانيكي يعطي بإحدى وضعياته جهداً يكافئ " 1" ويعطي بالوضع الآخر جهداً يكافئ " 0". يستخدم عادة مفتاح من النوع الذي يعمل بالضغط Pushbutton، بالتالي يمر بوضعيتين A و B ، وفق ما هو مبين بالدارة الكهربائية بالشكل 14-9 آ. عندما ينتقل ذراع المفتاح من وضع لآخر يهتز تماس المفتاح عدة مرات قبل أن يستقر في الوضع النهائي المطلوب. يستغرق عادة اهتزاز التماس الذي لا يلاحظه المستخدم العادي أجزاء من الثانية قبل اختفائه (وهذا ليس بالزمن القليل بالنسبة للأنظمة الرقمية)، وبالتالي هذا الاهتزاز يؤدي إلى تذبذب الإشارة بين الفتح والغلق نتيجة اهتزاز نقاط تماس المفتاح بشكل لا يسمح به عند تشغيل أو إيقاف النظام. يستخدم للتخلص من هذه السلبية، دارة إلغاء الاهتزاز لتقوم بإلغاء أثر سلسلة النبضات الناتجة عن اهتزاز التماس، وتعطي انتقالاً واحداً للإشارة. يمكن تشكيل مثل هذه الدارة باستخدام القلاب RS ومفتاح وحيد القطب بوضعيتين، وفق ما هو مبين في الشكل 14-9 ب، لتوليد معطيات ثنائية والانتقال من 0 إلى 1 مرة واحدة عند ضغط المفتاح.



الشكل14-9: دارة لإلغاء أثر اهتزاز مفتاح ميكانيكي باستخدام القلاب RS

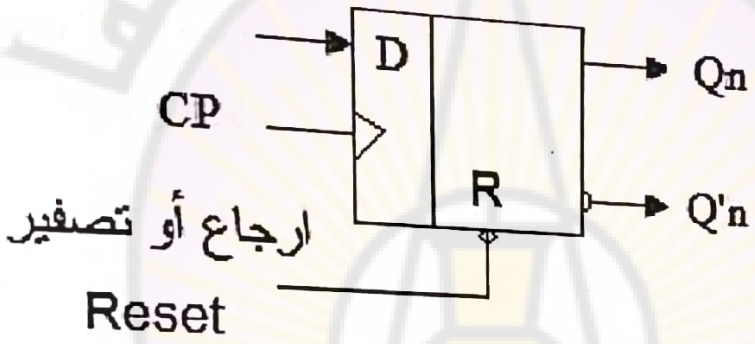
يتضح من الشكل أعلاه أنه إذا كان المفتاح بالوضع A الموصول على المدخل R، سيكون $R = 1$ و $S = 0$ ويكون الخرج $Q_n = 0$. وعند الضغط على المفتاح ونقله من الوضع A إلى الوضع B، ينتج أول تماس مع B ، ويبدأ المفتاح بالاهتزاز عدة مرات. ولكي يكون العمل سليماً، يجب أن نضمن عدم بلوغ نقطة التماس A عند الارتداد. إذاً يصبح $R = 0$ و $S = 1$ ، وينتقل الخرج إلى القيمة واحد ($Q_n = 1$). الاهتزاز المتكرر في النقطة B وتوليد عدة نبضات لم يعّد يؤثر على خرج القلاب، لأن R تبقى بالقيمة الصفر و S يمكن أن تتغير مابين 0 و 1. وإذا كانت $S = 1$ يكون الخرج بالقيمة 1، وهذا هو المطلوب. عند انتقال S للقيمة 0 يصبح $R = S = 0$ أي حالة عدم التغير(الذاكرة)، بالتالي يبقى القلاب بنفس الحالة السابقة أي $Qn = 1$، وبالتالي الخرج يظهر لمرة واحدة، ويبقى ثابتاً من دون تغير على الرغم من اهتزاز المفتاح. بعد هذا العرض الشامل حول القلاب RS، نشير هنا إلى أنه من النادر أن نصادف القلاب RS كدارة متكاملة جاهزة للتوصيل والاستخدام المباشر على عكس غيره من القلابات الأخرى وبخاصة القلابين D و JK التي تتواجد بكثرة كدارات متكاملة، وذلك لعدم الحاجة لوجود RS كدارة متكاملة لعدة اعتبارات من أهمها: إذا كان لدينا قلاب نوع JK يمكننا كما سنجد لاحقاً تشكيل أي قلاب بما في ذلك القلاب RS ومن جهة أخرى يمكن تشكيل القلاب RS من دارات تكاملية بسيطة تحتوي على بوابات NAND كل منها بمدخلين مثل الدارة 7400 أو دارة تحتوي على بوابات NOR كل منها بمدخلين مثل الدارة 7402، حيث يمكن تشكيل المخطط المنطقي الذي يقابل قلاب واحد أو أكثر من قلابات RS بالبوابات NAND أو بالبوابات NOR من هذه الدارات.

9-3-2 القلاب D

يقصد بالقلاب D من Delay التأخير، وهو قلاب بمدخل وحيد للمعطيات D وبمخرجين متممين لبعضهما(Q و 'Q). ويستخدم القلاب D مثل القلاب RS للتخزين المؤقت للمعلومات الثنائية. ويتم فيه تمرير أي إشارة على الدخل كما هي على الخرج بتأخير زمني.

1- الرمز المنطقي

يمكن عرض الرمز المنطقي للقلاب D بوجود مدخل خاص للساعة وآخر خاص بالمحو غير المتزامن R (فعّال بالصفر مثلاً)، وفق ما هو مبين في الشكل أدناه.

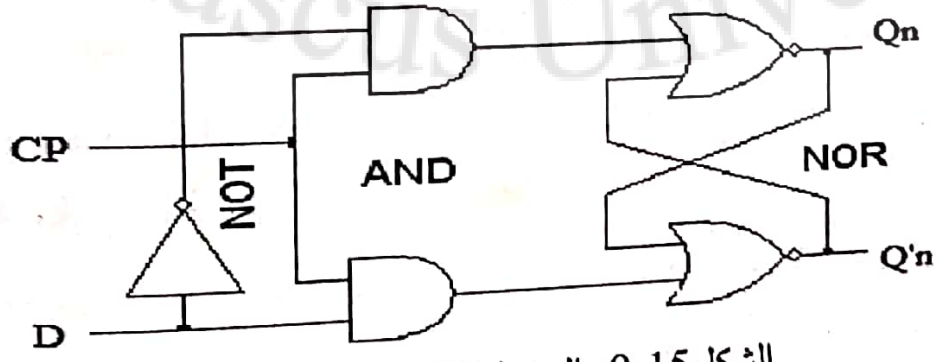


D	Q_P	Q_N
0	0	0
0	1	0
1	0	1
1	1	1

يظهر بالرمز المنطقي، بالإضافة إلى المخرجين Q'_n , Q_n، مدخل ظاهر واحد للمعطيات بالاسم D. ويمكن تصور مدخل آخر غير ظاهر Q_p، يمثل الحالة السابقة للقلاب.

2- المخطط المنطقي

تتكون دارة القلاب أو الماسك D من بوابتين NOR (أو NAND) وبوابتي AND بشكل مماثل لتركيبة القلاب RS المتزامن مع فارق أن المدخل D يوصل مكان المدخل S ونفيه يوصل بالمدخل R، وفق ما هو مبين في الشكل 15-9.



الشكل15-9: المخطط المنطقي للقلاب D

3- الجدول الوظيفي

يمتاز القلاب D انطلاقا من تركيبته، بأن ما هو موجود على المدخل D يظهر على المخرج Q_n بعد ظهور النبضة الساعية بتأخير معين وبغض النظر عن الحالة السابقة للقلاب. بالتالي جدول الوظائف للقلاب D مبين في الجدول 4-9.

الجدول4-9: جدول الوظائف للقلاب D

Inputs		Outputs
D	Q_p	Q_n
0	0	0
0	1	0
1	0	1
1	1	1

4- التابع المنطقي

يتضح من جدول الوظائف للقلاب D أن التابع المنطقي الذي يوصف عمل القلاب D هو $Q_n = D$. أي عندما يكون مدخل التحكم CP مؤهلاً ينقل ما هو على D إلى Q_n.

4- جدول التهيج (التحريض)

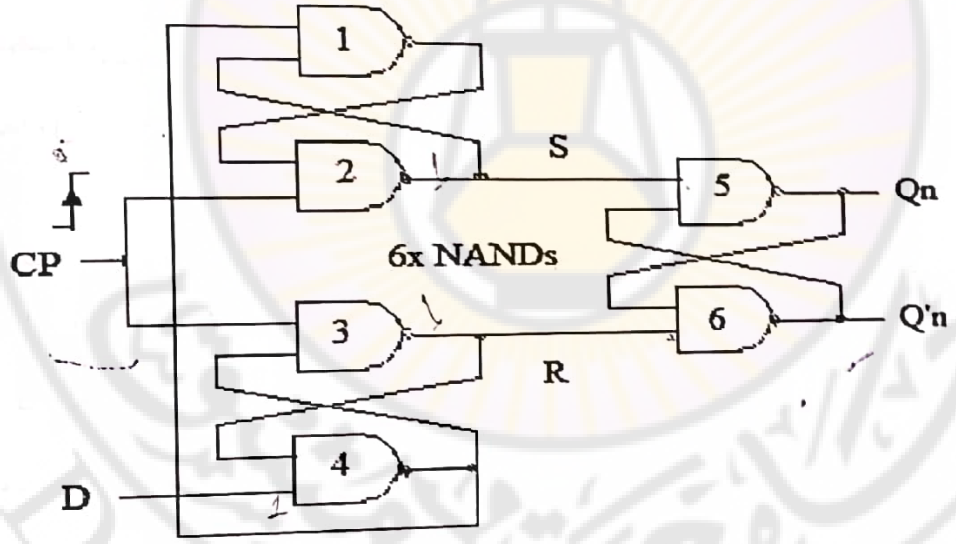
يحدّد جدول التهيج بالنسبة للقلاب D القيمة التي يفترض وضعها على المدخل D كي ينتقل القلاب من حالة ما إلى حالة أخرى، وبالتالي جدول التحريض بالنسبة للقلاب D، يبدو كما هو مبين في الجدول 5-9.

الجدول5-9: جدول التحريض للقلاب D

Q_p	Q_n	D
0	0	0
0	1	1
1	0	0
1	1	1

نكتفي بتناول هذه الجوانب الأساسية بالنسبة للقلاب D، حيث بقية الجوانب الأخرى المتعلقة به مماثلة لما تم تناوله بالنسبة للقلاب RS.

وللتخلص من مشاكل التزامن في الدارات التتابعية، أصبح بامكاننا التعريف بالطريقة الثانية التي سبق وأشرنا اليها، المتمثلة باستخدام قلابات مقدوحة بالحافة. ويمكن باستخدام القلاب D (Edge-Triggered D Flip-Flop) إظهار آلية ضبط عملية التزامن، بحيث يمكن التحكم في أن عملية تغيير حالة الخرج تحدث فقط، إما عند الحافة الصاعدة، أو عند الحافة الهابطة لنبضة الساعة. فإذا صممت الدارة بطريقة يحدث فيها التغير على الخرج مع الوجه الصاعد، تسمى الدارة بـ Positive) – Edge triggered Flip – Flop. وبالحالة المعاكسة تسمى الدارة بـ Negative – Edge triggered Flip-Flop. وكمثال عن قلاب D يقدح بالحافة، نورد البنية المبينة في الشكل 16-9، التي تُعدّ أكثر فعّالية للقلاب D، لأنها تقدح بالحافة الموجبة ثلاثة مواسك RS بآن واحد، اثنين منهما يستجيبان للمداخل D و CP والثالث يمثل مخرجاً للقلابين الأوليين.



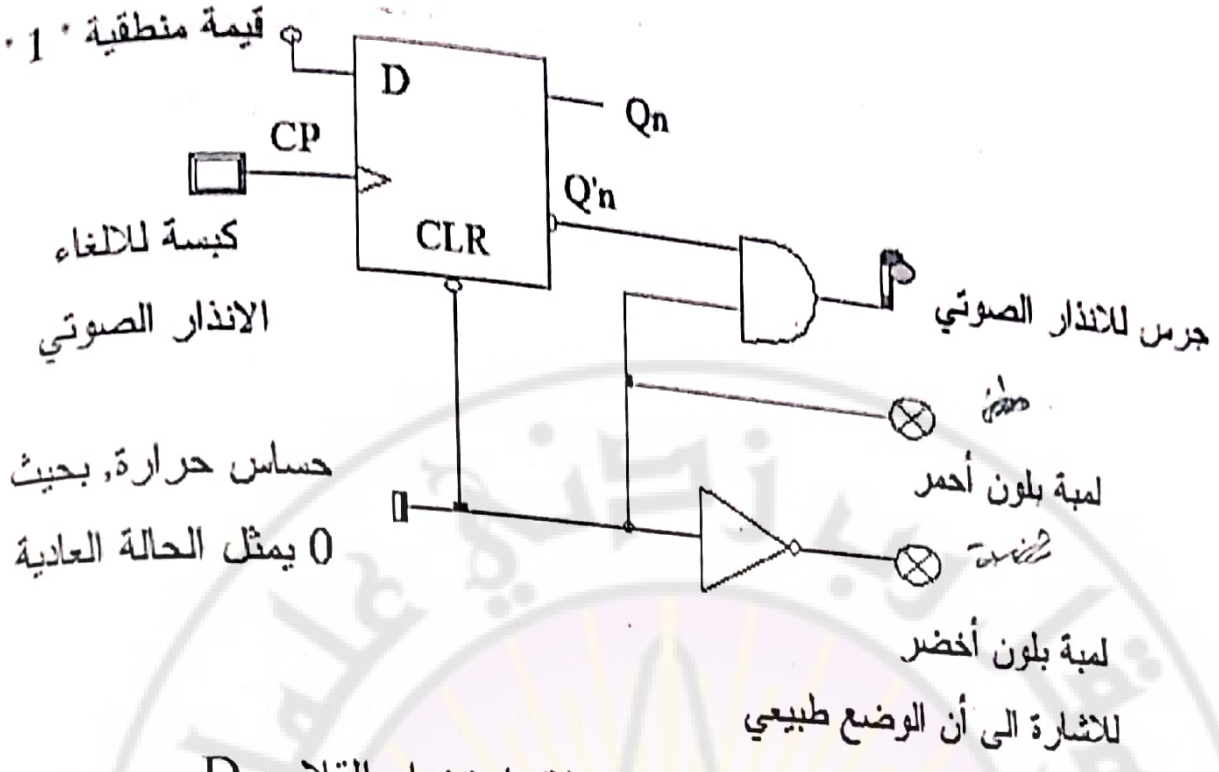
الشكل16-9: قلاب D يقدح بالحافة

تم في المخطط المنطقي للقلاب D الذي يقدح بوجه النبضة استخدام بوابات NAND، وبالتالي التركيبة التي تقابل $R = S = 0$ غير مسموح بها. ولنظهر الآن آلية استجابة الدارة لحافة النبضة الصاعدة، وليس لأي شيء آخر. بفرض أن $D = 1$ وأن $CP = 0$، هذا يعني أن خرج كل من بوابتي NAND ذات الأرقام 2, 3 سيكون 1، أي أن القلاب RS على الخرج بالبوابتين 5, 6 سيكون بحالة ذاكرة ولن يتغير الخرج (مهما تتغير قيم المدخل الخارجي D). بفرض أن إشارة الساعة CLK انتقلت من المستوى 0 إلى المستوى 1

يصبح مدخلا البوابة 2 بحالة الواحد (لأن الحالة السابقة للدارة R = 1 و D = 1، أي أن خرج البوابة 4 هو صفر وهو مدخل للبوابة رقم 1، وبالتالي خرج البوابة 1 هو أيضاً 1) وهذا يعني أن المدخل S للبوابة 5 أصبح بحالة الصفر وأن الخرج $Q_n = 1$، أي مع ورود الوجه الصاعد للنبضة ظهر ما هو موجود على المدخل D على الخرج Q_n. والآن بفرض أن قيمة D تغيرت من 1 إلى 0 مع بقاء المستوى العالي للإشارة CLK، يلاحظ التالي: ظهور الصفر على المدخل D يؤدي إلى جعل خرج البوابة 4 بحالة الواحد. ولكن هذا لا يعطي أي تغيير على خرج البوابة رقم 1 بسبب الصفر القادم من S (مخرج البوابة رقم 2)، وكذلك لا يحدث أي تغيير على حالة خرج البوابة 3 لنفس السبب، أي أن وضع صفر على المدخل D لم يحدث أي تغيير للقيم الموجودة على المداخل R و S ، وبالتالي قيمة Q_n بقيت كما هي من دون تغيير. وهكذا نكمل من أجل التراكيب المتبقية. ويلاحظ أن القيمة الموجودة على D لا تظهر على الخرج إلا بوجود الوجه الصاعد للنبضة على مدخل التحكم CP، ولا يؤثر الانتقال السالب للنبضة (من 1 إلى 0) على الخرج Q_n، كما لا يؤثر استقرار CLK بالمستوى 1 أو 0. وأخيرا نشير إلى أنه باستخدام منطق خارجي إضافي مع القلاب D يمكن بناء أنواع أخرى من القلابات كما سنجد لاحقاً. نورد فيما يلي أحد الأمثلة الممكنة حول استخدام القلاب D،بالتطبيقات العملية دون الدخول حالياً بتفاصيل التصميم.

مثال: صمم دارة تحكم بسيطة تعمل على مراقبة درجة الحرارة (لأي شيء تؤثر زيادة الحرارة عن حد ما على عمله)، بحيث إن درجة الحرارة المراقبة عن طريق حساس تعمل على إضاءة مؤشر بلون أخضر إذا كانت درجة الحرارة طبيعية. أما عند ارتفاع درجة الحرارة عن الحد المسموح به فتعمل الدارة على إطفاء الأخضر وإضاءة مؤشر بلون أحمر، وبنفس الوقت تعطي إنذارا صوتياً.

الجواب: بما أن هذا التطبيق يتطلب تبديلاً في الوضعية وذاكرة للوضع الذي تكون فيه الدارة، فإن من إحدى إمكانيات تصميم مثل هذه الدارة هو استخدام قلاب مثل D مع بعض البوابات المنطقية المتممة، وفق ما هو مبين في الشكل 17-9.

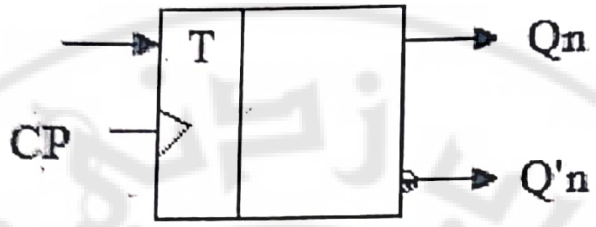


الشكل17-9: دارة مراقبة بسيطة باستخدام القلاب D

يتضح من الشكل 17-9 الذي يضم فقط قلاباً واحداً من نوع D مع بوابتين أنه طالما قيمة الإشارة القادمة من حساس الحرارة هي صفر (الوضع سليم) ستبقى اللمبة الخضراء عبر العاكس مضاءة والحمراء غير مضاءة، وكذلك الإنذار الصوتي سيكون عبر AND غير موجود أيضاً، لأنه بسبب ربط الإشارة القادمة من الحساس مع إشارة Clear للقلاب، تصبح $Q_n = 0$. ومجرد قدوم قيمة "1" من الحساس سيضاء الأحمر ويصبح الأخضر غير مضاء، وبنفس الوقت سيعمل الإنذار الصوتي، لأن قيمة Q'_n من المرحلة السابقة هي 1. ويستمر الوضع بهذا الشكل إلى حين الانتهاء من إزالة سبب ارتفاع الحرارة. وعند قدوم إشارة صفر من الحساس بعد الإصلاح، يضاء اللون الأخضر من جديد، ويصبح الأحمر غير مضاء، لكن الإنذار الصوتي يبقى موجوداً، حيث يمكن إلغاؤه بالضغط على المفتاح الموصول بـ Clock القلاب، الذي يجعل القلاب بحالة 1، وهذا بدوره يؤدي إلى جعل قيمة $Q'_n = 0$، وتوقيف الإنذار الصوتي. هذا التطبيق البسيط يمثل عينة من أمثلة عديدة تتطلب تخزين معلومة والانتقال من حالة لأخرى، يمكن تطبيقها باستخدام قلاب مثل D.

9-3-3 القلاب T

الرمز المنطقي للقلاب T(Toggle القلب) بوجود مدخل التحكم CP للنبضات الساعية مبين في الشكل 18-9.



الشكل18-9: الرمز المنطقي للقلاب T

يمكن توصيف عمل القلاب T وفق الجدول الوظيفي المبين بالجدول 6-9، الذي يظهر أن القلاب لا يغير حالته طالما المدخل T = 0 (حالة ذاكرة). عندما يكون T = 1 فان القلاب مع ورود النبضة الساعية يقلب إلى الحالة المعاكسة للحالة التي يكون فيها.

الجدول6-9: الجدول الوظيفي للقلاب T

Inputs		Outputs
T	Q_p	Q_n
0	0	0
0	1	1
1	0	1
1	1	0

ومن جدول الوظائف يمكن إيجاد التابع المنطقي المميز لعمل القلاب T بالعلاقة:

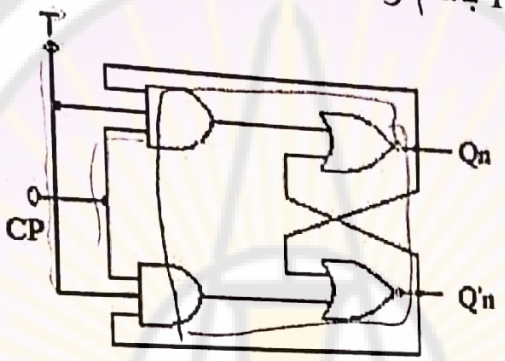
$$Q_n = T.Q'_p + T'.Q_p = T \oplus Q_p$$

أما جدول التحريض الذي يحدد القيمة التي يفترض تطبيقها على المدخل T للانتقال من حالة إلى حالة أخرى مبين في الجدول 7-9، حيث يظهر من الجدول أن القلاب عندما يغير حالته (الانتقال من 0 إلى 1 وبالعكس) يجب أن يكون T = 1. أما عندما يحافظ القلاب على الوضعية الموجود فيها سواء أكانت 0 أم 1، فيجب أن تكون T = 0.

الجدول7-9: جدول التحريض للقلاب T

Q_p	Q_n	T
0	0	0
0	1	1
1	0	1
1	1	0

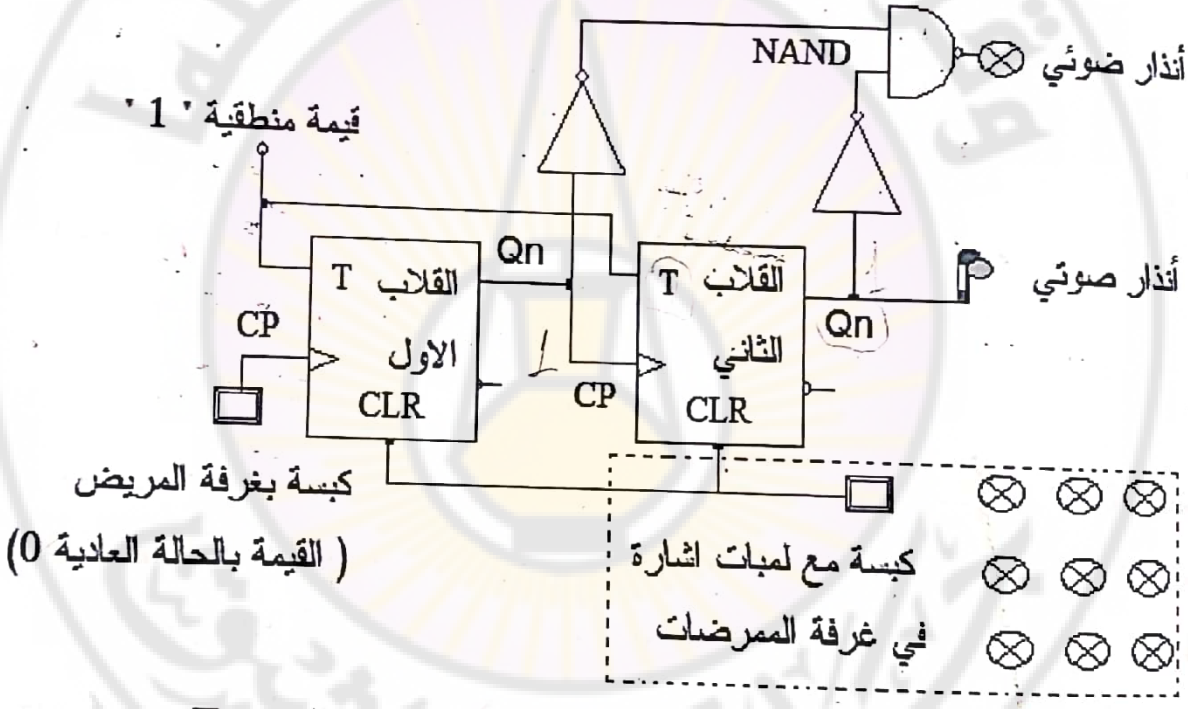
أما بخصوص المخطط المنطقي للقلاب T نعرضه بدون شرح وفق ما هو مبين بالمخطط أدناه، إلى حين التعرف على القلاب JK، لأن القلاب T هو حالة خاصة من القلاب JK تتمثل بقصر المدخلين J و K باسم واحد هو T.



وكتطبيق مباشر على استخدام القلاب T نورد فيما يلي تطبيقاً مماثلاً للتطبيق السابق:

مثال- حول تصميم نظام إنذار بسيط ضوئي وصوتي لمشفى بالمواصفات التالية: ليكن بكل غرفة بالمستشفى وبالقرب من كل سرير كبسة خاصة بالمريض ليتمكن بوساطتها من طلب الخدمات. وليكن بغرفة الممرضات لوحة إظهار تضم لمبات إشارة لأرقام الأسرة بكامل المشفى وكبسة جرس مشتركة لجميع الغرف. يطلب من النظام المراد تصميمه أن يستطيع المريض من خلال الضغط لمرة واحدة على الكبسة الموجود بالقرب من سريره طلب خدمة ما من الممرضات وذلك بإعلامهن من خلال إضاءة اللمبة التي تحدد رقم غرفته وسريره . وفي حال عدم تلبية الطلب (لعدم انتباه الممرضة للإنذار الضوئي) أن يستطيع المريض بالضغطة الثانية على نفس الكبسة أن يفعّل بالإضافة للإنذار الضوئي أيضاً الإنذار الصوتي. وعند انطلاق أي ممرضة لتأدية الخدمة، يمكنها من خلال كبسة خاصة بغرفتها أن تلغي الإنذارين من المريض الذي انطلقت لخدمته. الجواب: إذا تجاهلنا وحدة التغذية وبعض دارات التحويل البسيطة فإن أحد الحلول الممكنة يكون باستخدام قلابات مثل T مع بعض البوابات المنطقية موصولة مع بعضها بمخطط عام، وفق ما هو مبين في

الشكل 19-9، حيث يتضح من الشكل أن الدارة تعمل وفق التالي: بالحالة العادية، أي بعد تصفير الدارة بتفعيل إشارة المسح Clear بوضعها بالقيمة صفر، سيصبح خرج كُل من القلابين بالقيمة صفر. واللمبات في غرفة الممرضات تكون مضاءة باللون الأخضر. إذا احتاج المريض شيئاً وضغط لمرة واحدة على الكبسة بالقرب من سريره، يفترض أن تضاء اللمبة المقابلة لسريره باللون الأحمر في غرفة الممرضات، لأن الكبسة الأولى بوجود قيمة 1 على المدخل T للقلاب الأول، ستجعله يبدّل حالته، ويصبح Q_n مساوياً للواحد. وهذا الواحد عبر العاكس سيصبح صفراً، وبالتالي خرج NAND سيصبح 1 ويعمل الإنذار الضوئي.

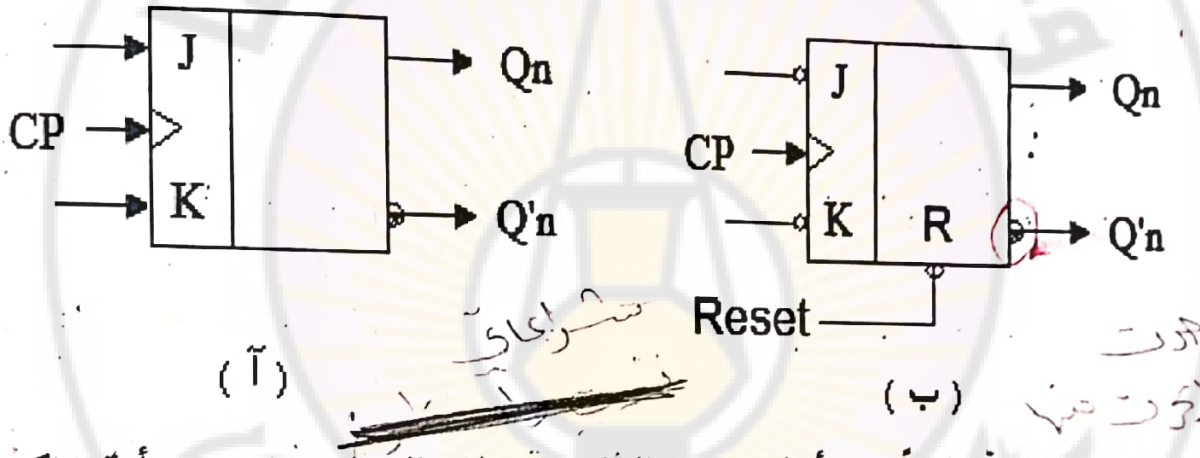


الشكل19-9: نظام تحكم ضوئي وصوتي باستخدام القلاب T

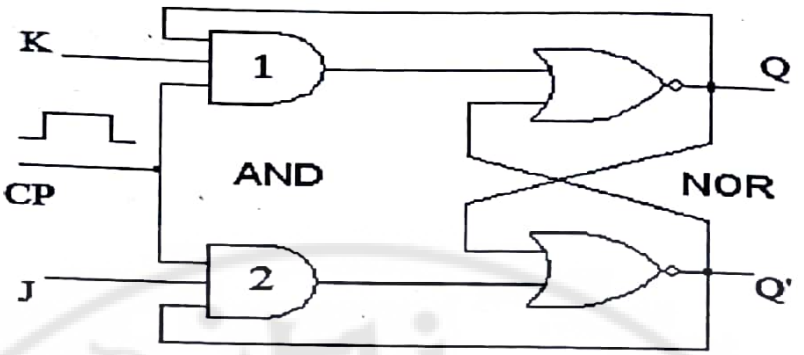
لا يعمل في هذه الحالة الإنذار الصوتي لوجود 0 على خرج القلاب الثاني. وعند الضغطة الثانية على نفس الكبسة بجانب المريض، سيبّدل كل من القلابين حالته، بحيث يصبح القلاب الأول في الحالة 0، والقلاب الثاني في الحالة 1. وهذا يؤدي بدوره إلى تشغيل الإنذارين معاً. ويستمر الوضع إلى حين الضغط على كبسة الجرس (الموصولة إلى إشارة تصفير القلابات) بغرفة الممرضات التي تقوم بتصفير كل من القلابين وتوقيف الإنذارين.

9-3-4 القلاب J-K

يعدّ القلاب J-K من أكثر القلابات انتشاراً، لأنه بوجود مثل هذا القلاب يمكن الاستغناء عن الأنواع الأخرى للقلابات. يمكن تشكيل القلاب J-K باستخدام البوابات NAND أو باستخدام البوابات NOR مثله مثل القلاب R-S. فمن أجل قلاب مشكل بالبوابات NOR يمكننا تصور أن المدخل J بالقلاب J – K مماثل للمدخل S بالقلاب RS بوظيفة الوضع Set، والمدخل K مماثل للمدخل R بوظيفة Reset، يمكن تصور الرمز المنطقي للقلاب J-K وفق ما هو مبين بالشكل (آ) أدناه. ويظهر الشكل (ب)، تشكيل القلاب J-K بالبوابات NAND بوجود مدخل مباشر للمحو غير المتزامن R.



يُعدّ القلاب J-K معرّفاً من أجل جميع التراكيب على الدخل ولا توجد أية تركيبة محظورة، كما هو الحال بالنسبة للقلاب RS. وصمم القلاب J-K، بحيث التركيبة $J = K = 1$ تصبح مسموحة. وتكون الحالة اللاحقة للقلاب من أجل هذه التركيبة على الدخل بعكس الحالة السابقة. ولكي تكون التركيبة $J = K = 1$ مقبولة، وتعطي خرجاً $Q_n = Q'_p$، أخذت تركيبة القلاب RS المتزامن وعدّلت، بحيث أضيف في مخطط القلاب RS إشارات راجعة من المخرجين Q و 'Q إلى مداخل بوابات AND، وفق ما هو مبين في الشكل 20-9. وبخصوص الجدول الوظيفي للقلاب J – K، سيكون من أجل المدخلين الظاهرين K و J والمدخل الثالث غير الظاهر Q_p مكون من ثمانية تراكيب مختلفة على الدخل. فإذا عدّنا $K = R$ و $J = S$ تكون التراكيب الستة الأولى هي نفسها التراكيب الستة الأولى بالنسبة للقلاب RS. والاختلاف هو فقط في آخر تركيبتين عندما تكون $J = K = 1$ و $Q_p = 0$ أو $Q_p = 1$.



الشكل20-9: المخطط المنطقي للقلاب J-K

يقلب القلاب إلى الحالة المعاكسة للحالة التي يكون فيها، كما هو مبين في الجدول 8-9.

الجدول8-9: الجدول الوظيفي للقلاب JK

Inputs			Output	Function
K	J	Q_p	Q_n	الوظيفة
0	0	0	0	Memory
0	0	1	1	حالة الذاكرة
0	1	0	1	Set
0	1	1	1	حالة التعبئة
1	0	0	0	Reset
1	0	1	0	حالة التصفير
1	1	0	1	Toggle
1	1	1	0	قلب الحالة

يمكن من الجدول الوظيفي للقلاب J – K تمثيل الوحدات بمخطط كارنوف واختصار التابع المنطقي إلى الشكل: $Q_n = J.Q'_p + K'.Q_p$. ويمكن بناء جدول التجريض بالنسبة للقلاب J-K، وفق ما هو مبين في الجدول 9-9، حيث كل انتقال من حالة إلى آخرى يقابله على الدخل إمكانيتان وهذا ما تم التعبير عنه باستخدام الرمز d (حالة عدم تعين). إذ إن التركيبة $J = 0$ و $K = d$ مثلاً، من أجل الانتقال من الحالة 0 إلى الحالة 0 يفهم بأنه يتم بحالتين هما: الأولى– حالة الذاكرة وهي تقابل التركيبة $J = K = 0$ على الدخل.

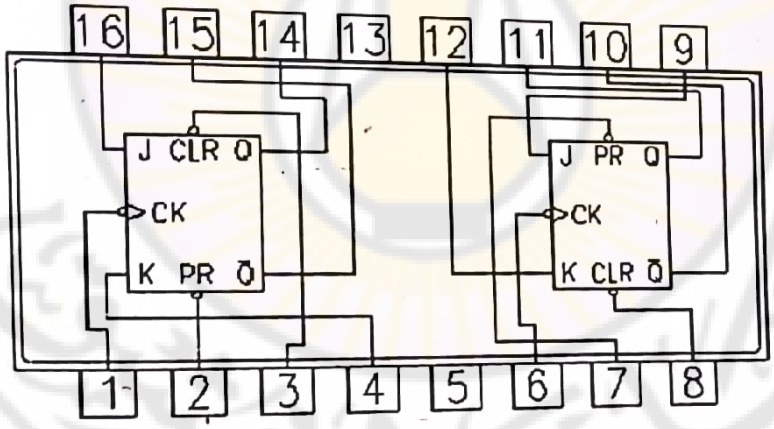
الثانية– حالة التصفير وتقابل التركيبة $J = 0$ و $K = 1$ على الدخل.

أي بالحالتين J = 0، أما K فيمكن أن تأخذ واحدة من القيمتين 0, 1 وبالتالي يكتب أن K = d، و J = 0. وهكذا الحال بالنسبة لبقية الانتقالات.

الجدول9-9: جدول التحريض للقلاب JK

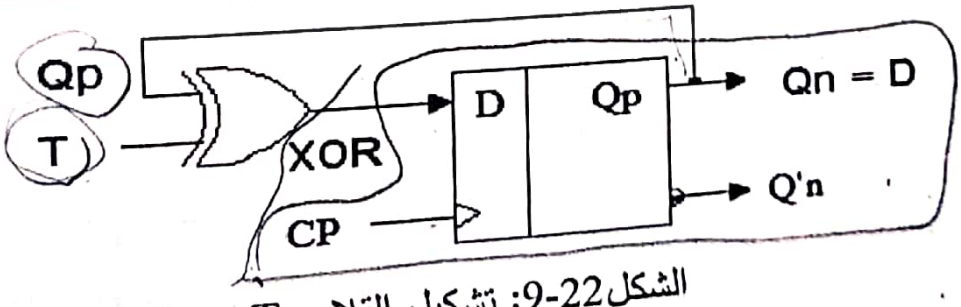
Q_p	Q_n	J	K
0	0	0	d
0	1	1	d
1	0	d	1
1	1	d	0

وبخصوص التطبيقات باستخدام القلاب J-K فهي عديدة، خاصة وأنه من القلاب J-K يمكن تأمين وظيفة أي من القلابات الأخرى، ويظهر الشكل 21-9 مثالاً عن دارة متكاملة تمثل قلابين من النوع J-K بدارة واحدة تحت الرقم 7476. وبالتالي يمكن استخدام هذه الدارة (القلاب) في كل من الأمثلة الثلاثة الواردة أعلاه بالوصل المناسب للمداخل J و K .



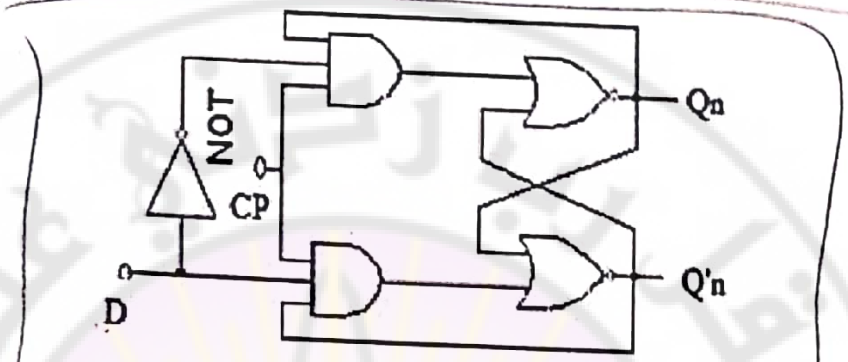
الشكل21-9: شكل توزع مآخذ القلابات J-K في الدارة 7476

ونختم الحديث عن القلابات بإظهار إمكانية تشكيل القلابات من بعضها، حيث يمكن بناء القلاب T من القلاب D باستخدام بوابة XOR وفق ما هو مبين في الشكل 22-9.



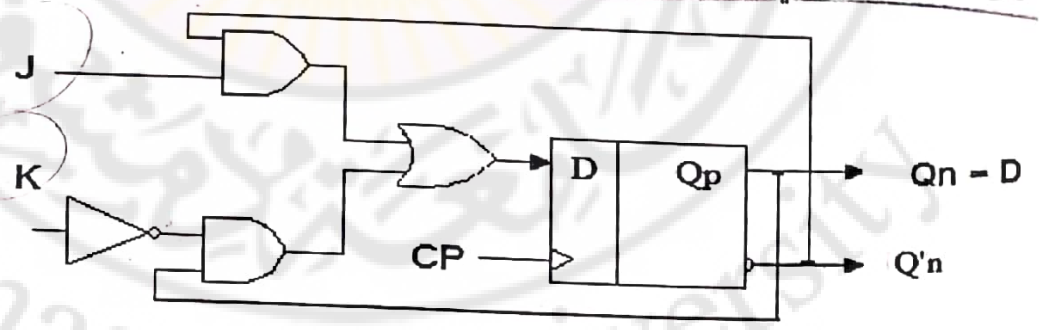
الشكل22-9: تشكيل القلاب T من القلاب D

كما يمكن من القلاب J – K تشكيل الأنواع الأخرى من القلابات، وهذا هو سبب تواجد JK أكثر من غيره على شكل دارات متكاملة. فمثلاً، يمكن تشكيل القلاب D من القلاب JK (مثل تشكيل القلاب D من القلاب RS) وذلك باعتبار المدخل J هو D وقصره عبر عاكس مع المدخل K، وفق ما هو مبين في الشكل 23-9.



الشكل23-9: تشكيل القلاب D من القلاب J-K

يمكن تشكيل القلاب T من J – K بقصر المدخلين معاً وتسمية المدخل الوحيد الناتج بالاسم T، كما سبق وأشرنا لذلك بالفقرة السابقة. كما ويمثل القلاب RS حالة خاصة من القلاب J-K، أو هو نفسه بفارق واحد هو أن التركيبة J = K = 1 لا تستخدم إذا اعتبرنا أن القلاب J–K هو القلاب RS. وبالعكس يمكن تشكيل القلاب J-K من القلاب D مثلاً، وفق ما هو مبين في الشكل 24-9.



الشكل24-9: تشكيل القلاب J-K من القلاب D

وأخيراً نشير إلى أن القلابات تصنع على شكل دارات متكاملة للاستخدامات العامة، وذلك بالتقانتين TTL وMOS. ومن الدارات المتكاملة الخاصة بالقلابات الأربعة والمصنعة بالتقانة TTL نورد في الجدول 10-9 أرقام بعض الدارات للاستخدامات العامة.

9-4 أسئلة وتمارين

اختر الجواب الصحيح لكل من التساؤلات التالية:

1- أي من العناصر المنطقية التالية يمثل دارة تتابعية؟

a. decoder
b. multiplexer
c. R-S Flip-Flop
d. comparator

2- القلاب يمثل دارة

a. Bistable
b. Monostable
c. Astable
d. All of the above

3- كل قلاب من القلابات D, T, RS, JK يمثل

a. the main elements to built sequential circuits
b. sequential circuits
c. memory with one bit
d. all the above

4- يمكننا بناء أو تركيب القلاب باستخدام

a- NAND gates only
b- NOR gates only
c- NAND and NOR gates
d- None of the above

5- يمكننا تشكيل القلاب D من القلاب JK بوسطه

a- by connect input J with input K directly
b- by connect input J with not input K
c- by connect not input J with input K
d- all the above

6- يمكننا تشكيل القلاب T من القلاب JK بوسطه

a- by connect input J with input K directly
b- by connect input J with not input K
c- by connect not input J with input K
d- all the above

7- يكون القلاب RS المشكل بالبوابات NOR بحالة عدم التعريف عندما:

a. R = 0, S = 0;
b. R = 0, S = 1;
c. R = 1, S = 0;
d. R = 1, S = 1

8- يكون القلاب RS المشكل بالبوابات NOR بحالة ذاكرة، عندما:

a. R = 0, S = 0;
b. R = 0, S = 1;
c. R = 1, S = 0;
d. R = 1, S = 1

9- يكون القلاب RS المشكل بالبوابات NAND بحالة عدم التعريف عندما:

a. $R = 0, S = 0$;
b. $R = 0, S = 1$;
c. $R = 1, S = 0$;
d. $R = 1, S = 1$

10- يكون القلاب RS المشكل بالبوابات NOR بحالة التعبئة Set عندما:

a. $R = 0, S = 0$
b. $R = 0, S = 1$
c. $R = 1, S = 0$
d. $R = 1, S = 1$

11- القيم على مداخل القلاب RS المشكل بالبوابات NOR التي تسبب الانتقال من الحالة السابقة $Q_p = 1$ إلى الحالة اللاحقة $Q_n = 0$، هي:

a. $R = 0, S = 1$;
b. $R = 0, S = d$;
c. $R = 1, S = 0$;
d. $R = d, S = 0$

12- القيم على مداخل القلاب RS المشكل بالبوابات NOR التي تسبب الانتقال من الحالة السابقة $Q_p = 1$ إلى الحالة اللاحقة $Q_n = 1$، هي:

a. $R = 0, S = 1$
b. $R = 0, S = d$
c. $R = 1, S = 0$
d. $R = d, S = 0$

13- يكون خرج القلاب RS بالمداخل المباشرة P-Preset و C- Clear بالقيمة واحد، عندما:

a. $R = S = d, C = P = 0$
b. $R = S = d, C = 0, P = 1$
c. $R = S = d, C = 1, P = 0$
d. $R = S = d, C = 1, P = 1$

14- خرج القلاب T من أجل الحالة الحالية Q_p يعطى بالعلاقة

a. $Q_n = T$
b. $Q_n = T.Q_p$
c. $Q_n = T \oplus Q_p$
d. None of the above.

15- إن القيمة التي يجب وضعها على مدخل القلاب D لنقله من الحالة $Q_p = 0$ إلى الحالة $Q_n = 1$ هي:

a- $D = 0$ b- $D = 1$ c- $Q_p = 1$ d- None of all

الفصل الأول

انظمة العّد Number Systems

نظرا للأنتشار الواسع للحواسيب الإلكترونية، وتنوع مجالات استخداماتها، قد يكون من المفيد بمقرر الدارات المنطقية، عرض المفاهيم الأساسية والمعايير اللازمة لتصميم مختلف العناصر المكونة لمختلف دارات الحاسوب وتوصيف آلية عمل كل منها على حدة وعرض أسس ربطها مع بعض البعض. بالتالي ستكون بداية هذا الفصل، مقدمة عامة للتعريف بالحاسوب الرقمي ومكوناته الأساسية، يليها التعرف إلى انظمة العّد الأساسية Number Systems وصولاً إلى آلية ترميز المعطيات وتمثيلها في الدارات الرقمية واستخدام قواعد التصميم المنطقي في تصميمها.

١-١ الحاسوب الرقمي

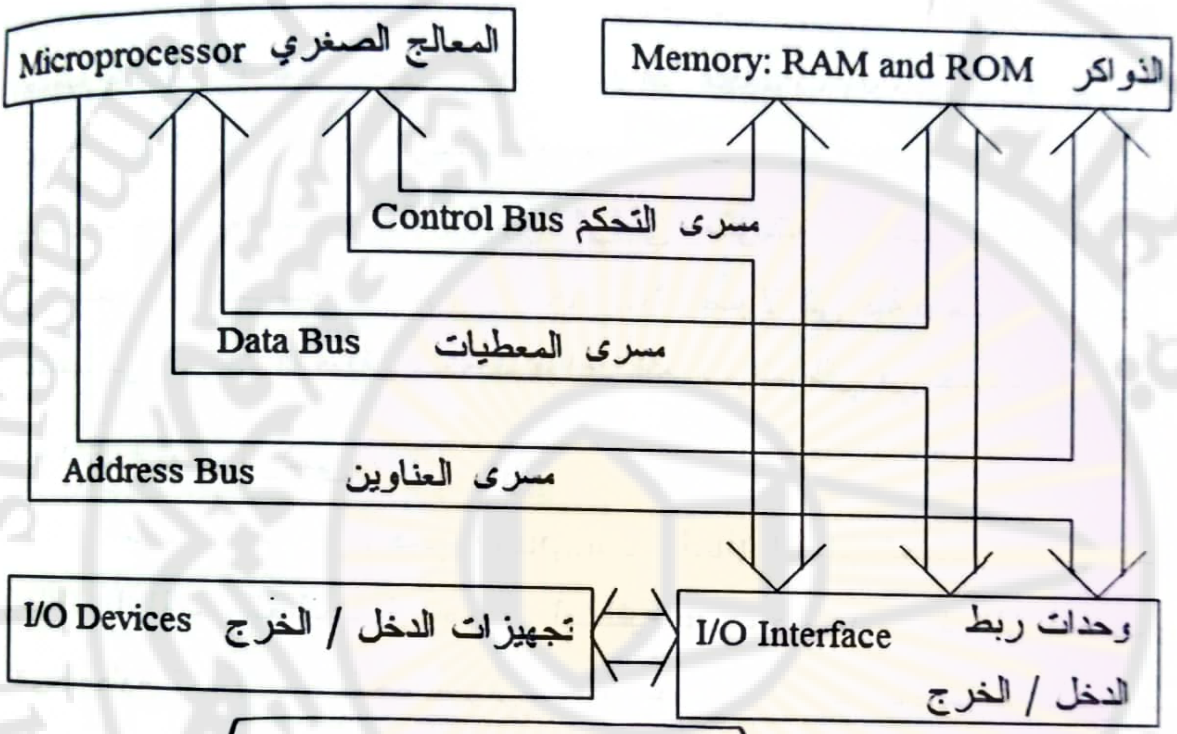
يعّرف الحاسوب الرقمي Digital computer بانه الجهاز الذي يقوم بإجراء سلسلة من العمليات الرياضية والمنطقية على مجموعة من البيانات للحصول على المعلومات اللازمة لاتخاذ القرارات المناسبة. يمكن النظر للحاسوب على انه جهاز إلكتروني متعدد الوظائف، يقوم بأي عمل يطلب منه وفق قواعد يحدّدها له المستخدم بوساطة برنامج Program يعبّر بدوره عن الخوارزمية التي توصف المسألة المطلوب معالجتها. ان سر قوة الحواسيب الرقمية وانتشارها هو امتيازها بالعديد من الميزات، مثل قدرتها على ديمومة العمل والسرعة والدقة الفائقة بالمعالجة والكمية الكبيرة من المعلومات التي يمكن ان تخزنها والاسترجاع السريع للمعلومات المخزنة، علاوة على قابليتها للتعديل والتطوير بإمكانية ربطها بالعديد من الأجهزة المحيطية الأخرى. يتكون أي حاسب من جزأين أساسين متممين لبعضهما البعض، بحيث ان كل جزء منهما يفقد معناه بغياب الآخر، هما:

يتكون أي حاسب من جزأين أساسيين

أ- الجزء التقني (أو العتاد) Hardware الذي يمثل كل ما يمكن تصنيعه ماديا من عناصر وبطاقات الإلكترونية لإدخال ومعالجة وإخراج المعلومات.

ب- الجزء البرمجي Software الذي يشمل كل ما هو غير ملموس وضروري لعمل الحاسوب من بيانات وتعليمات يحتاجها لأداء مهامه.

يمكن تصور التركيبة التقنية للحاسوب، وفق ماهو مبين في الشكل ١-١.



شكل ١-١: التركيبة العامة للحاسوب الرقمي

١-١-١ المكونات الأساسية للحاسوب الرقمي

يظهر من الشكل ١-١، ان الحاسوب الرقمي يتكون من أربع وحدات رئيسية:

١- وحدة المعالجة المركزية Central Processing Unit (CPU) وقد ظهر منها في الشكل أعلاه فقط المعالج الصغري Microprocessor، لأن هذا العنصر يمثل الأساس بالنسبة لهذه الكتلة وبالتالي بالنسبة لأي حاسب. تتكون وحدة المعالجة المركزية من:

- وحدة الحساب والمنطق Arithmetic and Logic Unit (ALU) التي تقوم بإجراء انماط مختلفة من العمليات الحسابية والمنطقية.

- وحدة التوقيت والتحكم Timing and Control Unit التي تقوم بوظائف التوقيت والتحكم بسير مختلف المسارات Processes.

- مجموعة من السجلات Registers التي تقسم بدورها إلى سجلات خاصة Special registers بعمل المعالج (ليس للمستخدم إمكانية الولوج لها وتستخدم لمهام محددة) وسجلات عامة General registers يمكن للمستخدم الولوج لها واستخدامها للتخزين المؤقت للمعطيات ونقل محتوياتها.

٢- وحدة الذاكرة Memory Unit لتخزين المعطيات والبرامج، وتمثل أيضا ساحة العمل للمعالج أثناء قيامه بتنفيذ برنامج ما. تشكل هذه الوحدة عنصراً أساسياً من تركيبة الحاسوب وأي نظام تحكمي آخر، ولذلك سيتم في هذا الكتاب تخصيص فصل كامل لها (الفصل الرابع عشر) وهي تتكون بدورها من جزأين:

الأول- لتخزين البرامج الأساسية المرتبطة بالعتاد التي تضم تعليمات تشغيل الحاسوب وتسمى بالبرامج المستقرة Firmware. وبالتالي تكون هذه الذواكر من النوع الدائم (غير المتطاير) Non-Volatile وتسمى بالذاكرة الثابتة وهي مخصصة للقراءة فقط Read Only Memory(ROM)

الثاني- لتخزين البرامج والمعطيات العامة وهي من النوع المؤقت (المتطاير أو غير الدائم Volatile) تفقد كل محتوياتها بمجرد انقطاع التيار الكهربائي عنها، وهي متاحة للمستخدم للقراءة منها والكتابة فيها والولوج إليها بشكل عشوائي وتسمى Random Access Memory (RAM).

٣- وحدة الإدخال والإخراج Input / Output Interface Unit تسمح هذه الوحدة بربط الحاسوب مع الأجهزة المحيطية Peripherals، وهي المسؤولة عن تأمين الموائمة اللازمة عند القيام بعملية نقل المعلومات بين أجهزة الإدخال والإخراج الخارجية ووحدة الذاكرة أو المعالج الصغري.

تجهز هذه الوحدة عادة بعدد من المداخل والمخارج يدعى كل منها بوابة Port وهنا نميز بين نوعين من البوابات: - بوابات للإدخال Input port

- وبوابات للإخراج Output port

يمكن من خلال البوابة الواحدة نقل مجموعة من البيانات الثنائية من خلال خط واحد أو من خلال عدة خطوط بآن واحد، تربط بالجهاز المحيطي (بحسب نوعه) فإذا تم نقل المعطيات عبر عدة خطوط من البوابة الواحدة بآن واحد، تسمى البوابة تفرعيه Parallel port وإذا قامت البوابة بنقل البيانات من أو إلى الجهاز المحيطي على خط اتصال واحد بشكل تتابعي (بتّ بعد بتّ)، تسمى بوابة تسلسلية Serial port.

٤- وحدة النواقل أو الممرات Bus Unit ونميز بين نوعين من النواقل:

الأول: النواقل الداخلية Internal bus وتكون ضمن الدارة المصنعة على شكل دارة متكاملة (مثل المعالج الصغري) وتستخدم لربط مختلف المكونات الأساسية الداخلية للدارة مع بعضها البعض.

الثاني: النواقل الخارجية External bus وتتكون من جملة خطوط اللازمة لتأمين الاتصال بين الوحدات المختلفة للحاسوب الرقمي وتقسم إلى ثلاثة انواع:

- نواقل العناوين Address bus: وهي نواقل وحيدة الاتجاه وتمثل جملة الخطوط المتوازية المخصصة لنقل العناوين من المعالج إلى مختلف المكونات الأخرى للحاسوب من ذواكر وبوابات.
- نواقل البيانات Data bus: وتتكون من مجموعة من الخطوط المتوازية لنقل المعطيات بين مختلف مكونات الحاسوب.
- نواقل التحكم Control bus: وهي نواقل مخصصة لنقل الإشارات التحكمية ولا يوجد لها نمط قياسي، إذ يختلف عدد خطوطها من معالج لآخر وان البعض منها قد يكون دخلاً والآخر قد يكون خرجاً.

تصنف النواقل من حيث الاتجاهية إلى نوعين رئيسين:

الأول- نواقل وحيدة الاتجاه تقوم بنقل الإشارات باتجاه واحد فقط وبنفس الاتجاه لجميع الخطوط.

الثاني- نواقل ثنائية الاتجاه Bidirectional تقوم بنقل الإشارات باتجاه واحد فقط في وقت محدد، ثم تقوم بنقل الإشارات بالاتجاه المعاكس في وقت آخر.

أما بخصوص الجزء البرمجي الذي لا يدخل في صلب هذا المقرر، يكفي هنا الإشارة إلى انه يمكن أيضا تقسيمه إلى ثلاثة مجموعات رئيسية:

الأولى: تضم مجموعة برامج الذاكرة الثابتة اللازمة لتشغيل الحاسوب وتجهيز منافذه وضبطها ومنها نذكر:

١- برامج التحفيز الذاتي Power On Self Test (POST): يتم تشغيل هذه البرامج آليا لحظة وصل التيار الكهربائي للحاسوب وينتهي عملها في حال عدم وجود خلل ويبدأ تنفيذ البرامج التالية.

٢- برامج الدخل والخرج الأساسية Basic Input Output Systems (BIOS): تنقل نسخة من هذه البرامج فور استدعائها إلى الذاكرة الرئيسية وتستقر فيها طول فترة عمل الحاسوب لتنظيم عملية تبادل المعطيات عبر النواقل على اللوحة الأم Motherboard بين مختلف منافذ الحاسوب.

٣- برامج الإعدادات Setup: لا تنفذ هذه البرامج آليا مثل البرامج السابقة لحظة تشغيل الحاسوب وانما يتم استدعاؤها من قبل المستخدم لتغيير إعدادات النظام وضبطه.

الثانية: وتضم البرمجيات الأساسية لإدارة عمل الحاسوب، أي برامج انظمة التشغيل Operating Systems (OS) وهي تمثل الوسيط بين المستخدم والبرامج التطبيقية من طرف ومكونات الحاسوب من طرف آخر، علماً ان هذه الأنظمة على اختلاف انواعها تصنف إلى:

- انظمة وحيدة المستخدم Single user OS التي لا تسمح بان واحد لأكثر من مستخدم التعامل مع الحاسوب، مثل نظام التشغيل DOS.
- انظمة تشغيل متعددة المستخدمين Multi-user OS التي تسمح بان واحد لأكثر من مستخدم التعامل مع الحاسوب، مثل نظام التشغيل UNIX

الثالثة: وتضم البرمجيات الأساسية لاستثمار الحاسوب، أي البرامج التطبيقية Application programs وهي البرامج التي تعطي الحاسوب صفة تعدد المهام، لأنه بحسب البرنامج التطبيقي الذي يحمّل في الذاكرة للتنفيذ، تتحدّد وظيفة الحاسوب. وتصنف هذه البرمجيات إلى نوعين أساسين، هما:

- حزم البرامج الجاهزة Software packages أو العامة، وهي برامج معّدة بشكل قياسي مثل: برامج معالجة النصوص والرسم وغيرها.
- البرامج التطبيقية الخاصة Special Application Programs وهي برامج يقوم بإعدادها المستخدم (المبرمج) بإحدى لغات البرمجة المعروفة.

يستخدم في الحاسوب الرقمي لتمثيل المعلومات، وحدة تدعى بتّ Bit، حيث البتّ الواحد يمثل أصغر وحدة لقياس المعلومات.

وان كل 4 bits تدعى Nibble وهي وحدة نادرة الاستخدام لقياس المعلومات. كل 8bits تستخدم لتمثيل محرف Character (أبجدي أو رقم أو رمز) وتشكل ما يعرف باسم بايت Byte.

من مضاعفات البايت، الكيلو بايت Kilobyte اختصاراً KB حيث، كل 1KB يعادل $1\ KB = 2^{10}\ bytes = 1024\ bytes$.

ومن المضاعفات الأخرى للبايت الميغابايت Megabyte اختصاراً MB، حيث كل 1MB يعادل $1MB = 2^{20} = 1024\ KB = 1024 \times 1024\ bytes$.

والجيكابايت Gigabyte اختصاراً GB، تعادل $1GB = 2^{30} = 1024\ MB$.

والتيرابايت Terabyte اختصاراً TB، تعادل $1TB = 2^{40} = 1024\ GB$.

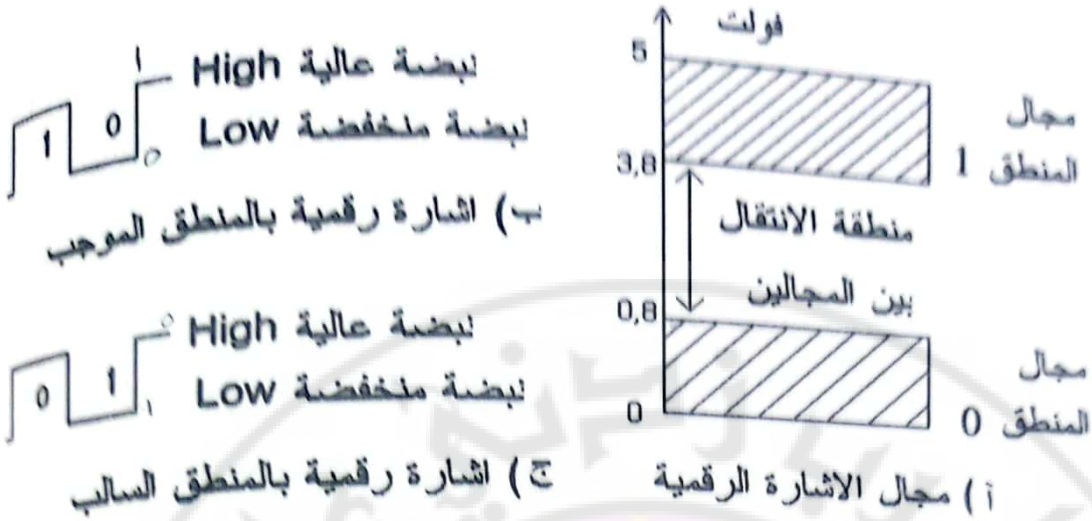
١-١-٢ أسس المنطق الرقمي

تمتاز النظم الرقمية بقدرتها على معالجة القيم المتقطعة Discrete ممثلة بالصيغة الثنائية Binary form، حيث تمثل القيم المتقطعة للمعلومات في النظم الرقمية بوساطة كميات فيزيائية، تدعى بالإشارات Signals. تُعد الإشارات الكهربائية مثل، الجهد Voltage والتيار Current أكثر الإشارات شيوعا. ولا يزال الترانزيستور هو العنصر الأساسي لانجاز هذه الإشارات في الدارات الإلكترونية. تستجيب الدارات الإلكترونية (المنطقية) التي تعمل بالجهد، لمستويين منفصلين من الإشارات لتمثيل القيم الممكنة لمتحول ثنائي. يأخذ كل متحول ثنائي قيمتين (1 , 0) تقابل بدورها حالتي الوصل والفصل (On و Off) للدارة المنطقية.

يُعبّر عن مجالات القيم التي تأخذها المتحولات الثنائية بوساطة نبضات كهربائية Pulses عالية المستوى H(High) ومنخفضة المستوى L(Low)، حيث يوجد لكل مستوى جهد مجال مقبول للتغيير ضمنه، كما هو مبين في الشكل ١-٢آ. يرمز لجهد التغذية Ucc بالحرف H، ويتم التعامل معه على انه يمثل واحداً منطقياً. ويرمز للجهد Uss بالحرف L ويتم التعامل معه على انه يمثل صفراً منطقياً. وبما ان قيمة الإشارة الرقمية على دخل أو خرج أي دارة رقمية تخصص بواحدة من قيمتين منطقيتين، يوجد طريقتان مختلفتان لتخصيص مستوى الإشارة (النبضة) في الدارات الإلكترونية الرقمية.

الأولى: وتسمى بالمنطق الموجب Positive logic وفيه يتم اختيار المستوى العالي H لتمثيل المنطق 1 والمستوى المنخفض L لتمثيل المنطق 0، كما هو مبين في الشكل ١-٢ب.

الثانية: وتسمى بالمنطق السالب Negative logic وهي عكس السابقة، يتم فيها اختيار المستوى المنخفض L لتمثيل المنطق 1 والمستوى المرتفع H لتمثيل المنطق 0، كما هو مبين في الشكل ١-٢ج.



الشكل ١-٢: انواع الإشارات الرقمية ومجالها

بالرغم من ان هذا التصنيف إلى " منطق موجب ومنطق سالب" معمول به، قد يربك أحياناً، لأنه يمكن للإشارتين ان تكونا موجبتين أو سالبتين، بمعنى ان قيمة الإشارة ليست هي التي تحدد نوع المنطق (موجب أو سالب)، ولكن عملية إسناد القيم المنطقية إلى المطالات النسبية لمستويي الإشارتين هي التي تحدده. المنطق الذي سيتم اعتماده في التوصيف من الان وصاعدا هو المنطق الموجب.

١-٢ الأنظمة الرقمية (أنظمة العّد)

يصادف في التطبيقات العملية انواع مختلفة من الإشارات. فمنها ماهو مستمر مثل الإشارات الصوتية أو الجهد المتغير باستمرار، وتسمى بالإشارات التمثيلية Analog. ومنها ماهو متقطع Discrete مثل النبضات الواردة من كاشف أو مفتاح كهربائي، وتسمى بالإشارات الرقمية Digital. يمكن تحويل الإشارات التمثيلية إلى إشارات رقمية باستخدام محولات خاصة، تسمى A/D Converter، وبالعكس يمكن تحويل الإشارات الرقمية إلى إشارات تمثيلية باستخدام محولات، تسمى D/A Converter. سيتم في هذا الكتاب التركيز على معالجة الإشارات الرقمية فقط، التي يعّبر عنها بنظام عددي محدّد، يسمى النظام الثنائي. يستخدم النظام الثنائي Binary system لتمثيل مختلف أشكال الإشارات الرقمية بالقيمتين 0 أو 1.

يوجد بالإضافة للنظام الثنائي انظمة عدّ أخرى تستخدم لتمثيل المعطيات على مختلف انواعها، مثل النظام الثماني والنظام الست عشري. يتميز كل نظام عدّ بعاملين رئيسين هما: - أساس النظام base (اختصاراً b)

- مكونات النظام Components

لا يُعّد أساس نظام العّد b من المكونات الأساسية للنظام، أما مكونات أي نظام عدّ فتبدأ من الصفر وتنتهي بالقيمة b – 1. يأتي اسم نظام العدّ من قيمة الأساس. فمثلاً، بالنسبة للرقم 5 يمكن تشكيل نظام عدّ وتسميته بالنظام الخماسي (أي نظام عددي بالأساس 5 والمكونات 0, 1, 2, 3, 4). تسمى مكونات النظام عادة بالأرقام Digits أما التراكيب المختلفة للأرقام ضمن النظام الواحد، تسمى بالأعداد Numbers. يمثل النظام العشري Decimal system، نظام العّد الذي يتعامل به البشر بشكل مباشر. يتم التعامل بالحياة اليومية مع أرقام وأعداد بوصفها أداة في تكوين معلومة وإيصالها. فمثلاً، عندما يُنفق مبلغاً ما لشراء حاجة معينة، يتم التعامل مع معلومة يعّبر عنها بأعداد عشرية. يسمى النظام الذي أساسه 10 ويضم الأرقام من 0, ... ,9 بالنظام العشري وفيه إذا زادت قيمة ما عن قيمة الأساس أو كانت مساوية له يُعّبر عنها بأرقام متوضعة في خانات متعددة، بحيث ان مجموع أوزنها يقابل القيمة نفسها. تعتمد انظمة العّد الثنائية والثمانية والستة عشرية نفس الأسس المتبعة في النظام العشري وهي مماثلة له تماماً مع اختلاف يتعلق بطبيعة كل نظام على حدة. بالتالي عرض المفاهيم المرتبطة في النظام العشري يساعد بفهم بقية انظمة العّد الأخرى. تتكون الأعداد في النظام العشري مهما كانت كبيرة أم صغيرة أو صحيحة أم كسرية أو موجبة أم سالبة من تراكيب مختلفة للأرقام 0, 1, 2, ... , 9. هذا يعني ان العدد العشري الذي سيتم التعامل معه سيكون ناتجاً عن ترتيب معين للبعض من هذه الأرقام العشرة وفق أسس محددة. فمثلاً، بالنسبة للعدد 789 يوجد بالخانة الأولى (آحاد) الرقم 9 وله وزن الواحد من $10^0 = 1$ والرقم الثاني 8 في الخانة الثانية (عشرات) له الوزن 10 من $10^1 = 10$ وهكذا على التوالي.

بالتالي هذه المجموعة من الأرقام لتعبّر عن العدد 789 تمثل بالشكل:

$$789 = 9 \times 10^0 + 8 \times 10^1 + 7 \times 10^2$$

يتعلق تزايد الأعداد أو تناقصها في النظام العددي بمفهومين رئيسيين، هما: القيمة المطلقة Absolute value والقيمة الموضعية Positional value، حيث القيمة المطلقة تعني الأرقام من 0, 1, ..., 9 التي يمكن ان توضع في كل خانة من خانات العدد. أما عند ما يتم تجاوز القيمة تسعة فانه يتم الانتقال إلى استخدام القيمة الموضعية التي تدل على مكان وقوع الرقم بالنسبة إلى بقية الخانات. يتم الحصول على القيمة الموضعية بضرب الأساس عشرة بنفسه عدد من المرات يساوي عدد الخانات المطلوبة لتمثيل قيمة العدد. وبشكل عام من أجل عدد عشري N يمكن التأكيد على وجود سلسلة رقمية وحيدة فقط من الشكل: $a_n\ a_{n-1}\ \ldots\ a_1\ a_0$، بحيث يكون فيها العدد N معادلاً للقيمة $N = a_0b^0 + a_1b^1 + \ldots + a_nb^n$، علماً ان $a_n\ \ldots\ a_0$ تعني المعاملات Coefficient وتمثل القيمة المطلقة. أما b فتعني الأساس. وبالنسبة للأعداد بالفاصلة العشرية، يجب فصل العدد إلى جزء صحيح Integer وجزء كسري Fraction part، لأن آلية التعامل مع كل منهما تختلف عن الآخر، حيث يعبّر عن كل قيمة موضعية تقع على يمين الفاصلة بالقوة السالبة التي يرفع لها الأساس. فمثلاً، العدد 23,741 يمكن التعبير عنه بالشكل:

$$N = 23{,}741 = (3 \times 10^0 + 2 \times 10^1).(7 \times 10^{-1} + 4 \times 10^{-2} + 1 \times 10^{-3})$$

ويعمم هذا الشكل باستخدام طريقة كثير الحدود Polynomial إلى النحو التالي:

$$N = (a_n\ b^n a_{n-1}\ b^{n-1} \ldots\ a_1\ b^1 a_0 b^0\ .\ a_{11}\ b^{-1} a_{22}\ b^{-2} \ldots\ a_{mm}\ ^{-m})$$

١-٢-١ النظام الثنائي

بما ان معظم التجهيزات الإلكترونية بما فيها الحواسيب تعتمد على وجود المكونات الإلكترونية الأساسية مثل: الترانزيستور والحلقات المغناطيسية أو الدارات المتكاملة التي كل منها يمر في حالتين فقط (فالترانزيستور يكون في حالة نقل التيار الكهربائي (حالة الفتح) أو عدم نقله (حالة الإغلاق)، وكذلك الحال بالنسبة للحلقات

المغناطيسية تكون ممغنطة في اتجاه ما (عكس عقارب الساعة) أو بعكس ذلك). يسمى النظام العددي الذي يوصف التعامل مع مثل هذه الدارات الرقمية، بالنظام الثنائي الذي يعبّر عن حالة الفتح ON بالرقم 1 وعن حالة الإغلاق OFF بالرقم 0. يُعدّ العدد 2 أساس النظام الثنائي والأرقام الصحيحة الأصغر من 2 فقط، أي الرقمين 0, 1 هما مكوناته. بقية المفاهيم المتعلقة بتمثيل أي عدد في هذا النظام مماثلة، كما سنجد لاحقاً، لما ورد بالنسبة للنظام العشري.

١-٢-٢ النظامان الثماني والستة عشري :

يستخدم كل من النظام الثماني Octal System والنظام الستة عشري Hexadecimal System لتسهيل التعامل مع المعطيات التي يأخذ تمثيلها في النظام الثنائي عدداً كبيراً من الخانات. يُختزل عدد الخانات لأي عدد ثنائي إلى الثلث عند التعبير عنه في النظام الثماني وإلى الربع عند التعبير عنه في النظام الستة عشري. يمثل النظام الثماني، نظام عددي أساسه 8 ومكوناته الأرقام: 0, 1, ..., 7.
أما النظام الستة عشري فيمثل نظام عددي أساسه 16 ومكوناته الأرقام:
0, 1, ...,9, 10, 11, 12, 13, 14, 15
وتجنباً لأي غموض بالنسبة للنظام الستة عشري بين الأرقام والأعداد، تستخدم بالنسبة للأرقام التي هي أكبر من 9، أي: 10, 11, 12, 13, 14, 15 رموز أبجدية، مثل A, B, C, D, E, F على التوالي، بحيث ان الرمز D مثلاً، يعادل الرقم 13 ويعبّر عن ان القيمة 13 هي وحدة غير قابلة للتجزئة، مثل أي رقم من 0,.. , 9.

١-٣ التحويل بين انظمة العّد

لتحديد النظام العددي الذي يوجد به عدد ما، يصطلح على وضع قيمته بين قوسين مع دليل Index إلى يمين القوس، تدل قيمته على اسم النظام. فمثلاً، العدد (1011) يمكن عدّه ثنائياً (اثنانياً) ويعبّر عنه بالشكل $(1011)_2$ أو ثمانياً ويعبّر عنه بالشكل $(1011)_8$ أو عشرياً ويعبّر عنه بالشكل $(1011)_{10}$.

يستخدم في بعض التطبيقات مثل البرمجة بلغة التجميع Assembly Language ترميز آخر للأعداد بالنظم العددية مستمد من اسم النظام.

فمثلاً، الرمز D من Decimal، يشير للأعداد العشرية.

والرمز B من Binary، يشير للأعداد الثنائية.

والرمز O من Octal، يشير للأعداد في النظام الثماني.

وأخيراً الرمز H من Hexadecimal، يشير إلى الأعداد في النظام الستة عشري وبالتالي العدد الوارد بالمثال أعلاه يمكن التعبير عنه بالأشكال التالية: 1011B بدلاً من $(1011)_2$ للإشارة إلى انه عدد ثنائي أو 1011O أو 1011D أو 1011H للدلالة على النظام المقابل.

عند التعامل مع أي عدد طوله n خانة بأي نظام عددي يصطلح أيضاً على ان الخانة الأدنى مرتبة أو وزناً في العدد تمثل دوما بأقصى اليمين ويعبّر عنها بالدليل 0 وأعلى خانة تمثل بأقصى اليسار ويعبّر عنها بالدليل n-1. فمثلاً، بالنسبة للبايت يكتب: رقم البتّ في البايت : 7 6 5 4 3 2 1 0

بتّات البايت : 1 1 0 1 0 1 0 0 LSB MSB

يطلق على الخانة ذات المرتبة 0 من أقصى اليمين بالعدد الثنائي $(1101\ 0100)_2$ مصطلح LSB أو LSD (Least Significant Bit or Digit) أي البتّ ذي الوزن الأدنى أو البتّ الأقل دلالة. وعلى الخانة ذات القيمة 1 من أقصى اليسار MSB أو MSD (Most Significant Bit or Digit) البتّ ذو الوزن الأعلى.

١-٣-١ التحويل من النظام العشري إلى النظام الثنائي وبالعكس

أولا- التحويل من النظام العشري إلى النظام الثنائي

تتم عملية تحويل الأعداد من النظام العشري إلى النظام الثنائي بتقسيم العدد العشري المعطى على 2 (أساس النظام) والاحتفاظ بباقي القسمة الذي سيكون مساوياً للواحد إذا كان العدد المقسوم فردياً، أو صفراً إذا كان زوجياً، وتعاد عملية

القسمة للناتج مرة أخرى على الأساس 2 وتستمر عملية القسمة حتى يتم الوصول إلى الحالة التي لا يمكن عندها التقسيم على الأساس 2. تكتب بعد ذلك البواقي على التوالي من اليمين إلى اليسار، ابتداء من أول باق.

مثال: حول العدد الفردي (23) من النظام العشري إلى ما يقابله في النظام الثنائي ؟

الجواب: للتحويل تطبق القاعدة أعلاه، فيكون:

23/2 = 11 Reminder = 1 (First reminder)
11/2 = 5 Reminder = 1
5/2 = 2 Reminder = 1
2/2 = 1 Reminder = 0
Last Result = 1 < 2 (base) ~ reminder = 1

يكون ترتيب البواقي من اليمين إلى اليسار ابتداءً من أول باق وفق الشكل التالي: (10111) وتكتب نتيجة التحويل $(10111)_2 = (23)_{10}$.

ويمكن تحويل أي عدد عشري إلى ما يقابله في النظام الثنائي بشكل آخر وفق الطريقة المباشرة، انطلاقاً من التالي: يمكن بخانة واحدة في النظام الثنائي تمثيل قيمتين فقط، هما 0 أو1. ويمكن بخانتين تمثيل أربعة قيم ثنائية هي 00, 01, 10, 11 وهذا ما يقابل أعداد عشرية من 0 وحتى 3، وبثلاث خانات يمكن تمثيل 8 قيم، وهكذا من أجل n خانة يمكن تمثيل أعداد عشرية من 0 وحتى $2^n - 1$. فإذا تم التعبير عن الخانات بالأوزان المقابلة لها وفق الشكل: 2^0 2^1 2^2 2^3 2^4 .... يمكن تمثيل أي عدد عشري بالطريقة المباشرة بوضع قيمة واحد في الموضع الذي يلزم لتشكيل العدد المعطى وصفر في الموضع الذي لا يلزم.

مثال: العدد العشري $(41)_{10}$ يعبّر عنه بالشكل التالي:

الأوزان	:	2^5	2^4	2^3	2^2	2^1	2^0
القيمة الموضعية:		32	16	8	4	2	1
مكونات العدد	:	32		8			1
القيمة المطلقة	:	1	0	1	0	0	1

القيمة الأخيرة (المطلقة) هي التي تقابل الترميز الثنائي للعدد العشري $(41)_{10}$ ويكتب $(1\ 0\ 1\ 0\ 0\ 1)_2 = (41)_{10}$، لأن العدد المعطى يقابل من قيم 2^n مجموع القيم 1, 8, 32 حصراً.

ولتحويل العدد العشري الكسري إلى ما يقابله في النظام الثنائي، فان الأمر لا يختلف بالنسبة للجزء الصحيح عما سبق، أما بالنسبة للجزء الكسري، فعوضاً عن القسمة على الأساس 2 يتم ضرب هذا الجزء بالقيمة 2 وذلك عدد من المرات حتى يصبح الناتج صفراً. يسجل بعد ذلك النقل الناتج ابتداء من الفاصلة لكن من اليسار إلى اليمين مثال : حول العدد $(13,40625)_{10}$ إلى ما يقابله في النظام الثنائي؟

الجواب: يحول كل من الجزء الصحيح والجزء الكسري على حدة.

يحول الجزء العشري الكسري للشكل الثنائي وفق التالي:

(Carry)	(Number)
	0, 4 0 6 2 5
	x 2
0	8 1 2 5 0
	x 2
1	6 2 5 0 0
	x 2
1	2 5 0 0 0
	x 2
0	5 0 0 0 0
	x 2
1	0 0 0 0 0

ويكون الناتج $(.01101)_2 = (0,40625)_{10}$.

يتم تحويل الجزء الصحيح بواحدة من الطريقتين إلى الشكل: $(1101)_2 = (13)_{10}$

ويصبح الناتج النهائي: $(1101.01101)_2 = (13,40625)_{10}$.

يُعدّ العدد المعطى بالمثال أعلاه بجزئه الكسري نظامياً، لأنه بنتيجة الضرب عدة مرات بالعدد (2) يتم الحصول على الصفر، ولكن هناك أعداد غير نظامية، بمعنى انه مهما يضرب الجزء الكسري بالعدد (2) لا يتم الحصول على الصفر. بمثل هذه

الحالة يتم الضرب عدداً من المرات حتى يلاحظ ان العدد بدأ يتكرر. يؤخذ واحد من الأجزاء المكررة أو يؤخذ الناتج بتقريب معين لعدد من الخانات. فمثلاً، لتحويل العدد العشري $(22,27)_{10}$ إلى ما يقابله في النظام الثنائي، يلاحظ ان الجزء الكسري يقابل القيمة غير المنتهية $(.0100010100011110101111)_2$، أي ان السلسلة بعد أول رقمين 01، أي 0001010001111010111 ستتكرر، وبالتالي يكفي بالجواب كتابة هذه السلسلة لمرة واحدة أو ان يقرب الناتج إلى خانتين ويكتب الجواب النهائي بالشكل: $(10110.01)_2 = (22,27)_{10}$.

ثانيا- التحويل من النظام الثنائي إلى النظام العشري

لتحويل أي عدد ينتج من الدارات الرقمية التي تتعامل في النظام الثنائي (بعد الانتهاء من المعالجة) إلى الشكل الذي يتعامل به الانسان، أي النظام العشري، يتم انطلاق من الشكل العام لكثير الحدود الذي سبق وأشرنا إليه عن نشر قيمة عدد ما N من الشكل: $N = a_0b^0 + a_1b^1 + \ldots + a_{n-1}b^{n-1}$.

يمكن تحويل أي عدد ثنائي إلى ما يقابله في النظام العشري بجمع أوزان خانات العدد الثنائي المحتوية على واحد فقط وإهمال قيم الخانات المحتوية على أصفار. يعّبر عن ذلك وفق ما هو مبين بالمثال التالي:

مثال: حول العدد الثنائي $(101110)_2$ إلى ما يقابله في النظام العشري $_{10}$(؟)

الجواب: العدد المعطى مكون من ست خانات، توزع على الأوزان المقابلة لكل منها وفق التالي:

0	1	1	1	0	1	الترميز الثنائي المعطى (القيمة المطلقة)
2^0	2^1	2^2	2^3	2^4	2^5	الأوزان
1	2	4	8	16	32	القيمة الموضعية

وبالتالي سيمثل الناتج مجموع القيم العشرية التي تقابل الواحد، أي $2 + 4 + 8 + 32$ وهو المقابل العشري للعدد الثنائي المعطى ويكتب:

$$(101110)_2 = 0x2^0+1x2^1+1x2^2+1x2^3+0x2^4+1x2^5$$
$$0 + 2 + 4 + 8 + 0 + 32 = (46)_{10}$$

وفي حال احتواء العدد الثنائي على جزء كسري، يحوّل كلاً من الجزء الصحيح والجزء الكسري إلى ما يقابلها في النظام العشري على حدة.

مثال : اوجد القيمة العشرية للعدد الثنائي $(11001,1011)_2$ ؟

الجواب:بحساب قيمة كل من الجزء الصحيح والكسري للعدد المعطى على حدة، يكون

$$N = (11001, 1011)_2 = (1.2^0 + 0.2^1 + 0.2^2 + 1.2^3 + 1.2^4).$$
$$* (1.2^{-1} + 0.2^{-2} + 1.2^{-3} + 1.2^{-4}) = (1+8+16).\ (1/2 +1/8 +1/16)$$
$$= (25.687)_{10}$$

١-٣-٢ التحويل من النظام العشري إلى النظام الثماني وبالعكس

أولا- التحويل من النظام العشري إلى النظام الثماني

يتم هنا بدلاً من التقسيم على 2 بالنسبة للنظام الثنائي التقسيم على 8، حيث أي رقم في النظام الثماني سيكون أصغر من الرقم 8 (الأساس). وتبقى عملية كتابة البواقي والترتيب كما هي في النظام الثنائي.

مثال : حول العدد العشري (397) إلى ما يقابله بالثماني، أي $(397)_{10} = (\ ؟\ )_8$.

الجواب: 397/8 = 49 Reminder = 5

49/8 = 6 Reminder = 1

وبما ان الناتج الأخير 6 أصغر من الأساس 8، فانه يُعدّ باقياً أيضاً. وتكون البواقي هي الرقم 5 بوصفه أول باقٍ ويكتب بأقصى يمين الناتج، يليه الرقم 1 وأخيرا الرقم 6، ويصبح الجواب النهائي من الشكل: $(397)_{10} = (615)_8$.

أما بخصوص الأعداد الكسرية فتحول بنفس الآلية التي تم فيها تحويل عدد عشري إلى ثنائي بعد مراعاة قيمة الأساس ومبدأ التحويل.

مثال: اوجد ما يقابل العدد العشري $(75,513)_{10}$ في النظام الثماني؟

الجواب: يحول كل من الجزء الصحيح (75) والجزء الكسري (513) على حدة

- الجزء الكسري: يتم الضرب بقيمة الأساس 8 ويسجل المنقول وتستمر

عملية الضرب حتى يتم الحصول على الناتج صفر.

$0.513 \times 8 = 4.104$
$0.104 \times 8 = 0.832$
$0.832 \times 8 = 6.656$
$0.656 \times 8 = 5.248$
$0.248 \times 8 = 1.984$
$0.984 \times 8 = 7.872$

وبما ان العدد المعطى لا يصل بنتيجة الضرب المتكرر بالأساس 8 إلى الصفر، يكفي بهذه الحالة التقريب إلى عدد ما من الخانات. فإذا تم الاكتفاء بأول ست خانات مثلاً، يصبح الجواب: $(0,513)_{10} = (0,406517)_8$

- الجزء الصحيح يعادل: $75 / 8 = 9$ Reminder $= 3$

$9 / 8 = 1$ Reminder $= 1$

Last Result = 1 is reminder

بالتالي الناتج $(75)_{10} = (113)_8$

ويصبح الناتج النهائي: $(75,513)_{10} = (113,406517)_8$ وهو المطلوب.

ثانيا- التحويل من النظام الثماني إلى النظام العشري

تُعدّ عملية التحويل Octal to Decimal Conversion معاكسة للعملية السابقة وهي مماثلة لعملية التحويل من النظام الثنائي إلى العشري.

مثال : حول العدد (761) في النظام الثماني إلى ما يقابله في النظام العشري، أي اوجد $(761)_8 = (\ ?\ )_{10}$

الجواب: $(761)_8 = 1.8^0 + 6.8^1 + 7.8^2 = (497)_{10}$

١-٣-٣ التحويل من النظام العشري إلى النظام الستة عشري وبالعكس

أولا- التحويل من النظام العشري إلى النظام الستة عشري

مثال : حول العدد $(3915)_{10}$ إلى ما يقابله بالستة عشري $(\ ؟\ )_{16} = (3915)_{10}$

الجواب: عند إجراء عملية التقسيم على الأساس 16 في النظام الستة عشري قد يكون

الباقي رقماً من 0 وحتى 15. يكتب الباقي كما هو باستثناء الأرقام من 10 وحتى 15 يكتب عوضاً عنها الرمز المقابل لكل منها من A وحتى F.

3915/16 = 244 Reminder = 11 ~ B
244/16 = 15 Reminder = 4
Last Result = 15 ~ F (< 16 (base))

وبالتالي الجواب يكون: $(3915)_{10} = (F4\ B)_{16}$.

ثانيا- التحويل من النظام الستة عشري إلى النظام العشري

بأسلوب مماثل لما سبق يمكن توضيح عملية التحويل من النظام الستة عشري إلى العشري Hexadecimal to Decimal Conversion بالرجوع إلى كثير الحدود: $N = a_0b^0 + a_1b^1 + \ldots + a_{n-1}b^{n-1}$

مثال: حول العدد الست عشري $(1A9)_{16}$ إلى ما يقابله في النظام العشري؟

الجواب:

$$(1A9)_{16} = 9.16^0 + A.16^1 + 1.16^2 =$$
$$9.16^0 + 10.16^1 + 1.256 = (425)_{10}$$

أما بخصوص تحويل الأعداد بفاصلة من النظام العشري إلى النظام الستة عشري وبالعكس، تستخدم نفس الطريقة المستخدمة بكل من النظامين الثنائي والثماني.

مثال: اوجد القيمة العشرية المقابلة للعدد الستة عشري $(2A9,8C)_{16}$ ؟

الجواب: بتمثيل كل من الجزء الصحيح والجزء الكسري على حدة يكون:

$$(2A9,8C)_{16} = (2x16^2 + Ax16^1 + 9\ x16^0)\ .\ (8x16^{-1} + C\ x16^{-2})_{10}$$
$$= (512+160+9)\ .\ (8/6+12/256)_{10}$$
$$= (681.140 / 256)_{10}$$

١-٣-٤ التحويل بين الأنظمة الثنائية والثمانية والستة عشرية

بما انه يمكن كتابة أساس النظام الثنائي بالشكل 2^1 والثماني بالشكل 2^3 والستة عشري بالشكل 2^4، هذا يعني، إذا كانت الأعداد ممثلة ثنائياً، فان كل خانة في النظام الثماني تقابل ثلاث خانات في النظام الثنائي وكل خانة في النظام الستة عشري

تقابل أربع خانات في النظام الثنائي. يبيّن الجـدول ١-١ تحـويلات لأول 20 قيمـة عشرية بين مختلف هذه الأنظمة.

النظام العشري b = 10	الثنائي b = 2	الثماني b = 8	الست عشري b =16
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	1 0000	20	10
17	1 0001	21	11
18	1 0010	22	12
19	1 0011	23	13

جدول ١-١: بعض التحويلات بين انظمة العّد الأربعة الأساسية

يمكن باستخدام القاعدة المشار إليها أعلاه، إجراء عملية التحويل من النظـام الثمـاني إلى الثنائي وبالعكس وفق ماهو مبيّن بالأمثلة أدناه.

مثال : حول العدد $(437)_8$ من النظام الثماني إلى ما يقابله في النظام الثنائي $(\ ؟\ )_2$

الجواب: يمكن تمثيل كل رقم من أرقام العدد $(437)_8$ على ثلاثة خانات في النظام الثنائي كالتالي: $(7)_8 = (111)_2$، $(3)_8 = (011)_2$، $(4)_8 = (100)_2$

تكتب هذه الأرقام جنب بعضها وفق ورودها بالعدد المعطى فيكون الناتج: $(100\ 011\ 111)_2 = (437)_8$ وهو المطلوب. ويمكن إجراء عملية التحويل مباشرة من النظام الثنائي إلى الثماني وفق التالي:

مثال: يمكن من أجل عدد ثنائي من الشكل $(10\ 110\ 101)_2$، تجميع كل ثلاثة بتّات مع بعضها ابتداء من اليمين وكتابة المقابل الثماني لها، لأنه بثلاث خانات يمكن على الأكثر تمثيل التراكيب من 000 وحتى 111 التي تعادل الأرقام من 0 وحتى الرقم 7 وهي مكونات النظام الثماني. بالتالي يتم بالمثال المعطى الحصول على $(\ 2\ 6\ 5\ )_8 = (010)\ (110)\ (101)$ وهو المطلوب.

ويمكن بنفس الطريقة تحويل أي عدد في النظام الستة عشري إلى ما يقابله في النظام الثنائي وبالعكس وذلك بعد تخصيص كل رقم ستة عشري بأربع خانات ثنائية.

مثال : حول العدد الستة عشري $(B5)_{16}$ إلى ما يقابله في الثنائي $(\ ؟\ )_2 = (B5)_{16}$

الجواب: بما ان: $(5)_{16} = (0101)_2$ و $(B)_{16} = (1011)_2$، يصبح الجواب $(1011\ 0101)_2 = (B5)_{16}$ والعكس صحيح.

يلاحظ مما سبق، أهمية استخدام كلاً من النظامين الثماني والستة عشري باختزال عدد خانات أي عدد ثنائي إلى الثلث أو إلى الربع.

وأخيرا بخصوص التحويل من النظام الثماني إلى النظام الستة عشري وبالعكس. العملية غير هامة، لأن التحويل بين هذين النظامين غير مستخدم تقريباً. ورغم ذلك يمكن تحويل العدد المعطى بأي من النظامين إلى النظام العشري ومنه إلى النظام المطلوب. أو بتحويل العدد المعطى إلى النظام الثنائي مباشرة ومنه للنظام المطلوب.

مثال: حول قيمة العدد الستة عشري $(1A4)_{16}$ إلى ما يقابله بكل من الأنظمة التالية: $(?)_2$ ، $(\ ?\ )_8$ ، $(?)_{10}$

الجواب:

$(1A4)_{16} = (\ 1\ \ 1010\ \ 0100\ )_2$

$(1A4)_{16} = (\ 1\ 1010\ 0100\ )_2 =$

$(\ 110)(100)(100) = (\ 644)_8 = (420)_{10}$

يطبق المبدأ المبين أعلاه حول تخصيص كل رقم في النظام الثماني بثلاث خانات في النظام الثنائي على الأعداد الكسرية أيضاً.

مثال: اوجد المكافئ الثماني للعدد الثنائي: $(10101001111.11001101)_2$

الجواب: يجمّع كل من جزئي العدد إلى مجموعات بكل منها ثلاثة أرقام، بحيث التجميع يكون بالنسبة للجزء الصحيح من اليمين باتجاه اليسار وبالنسبة للجزء الكسري من اليسار باتجاه اليمين وفق التالي:

$(10\ \ 101\ \ 001\ \ 111\ .110\ \ 011\ \ 01)_2$

$2\ \ 5\ \ 1\ \ 7\ ,\ \ 6\ \ 3\ \ 1\ \ = (2517, 631)_8$

ماينطبق على النظامين الثنائي والثماني بخصوص تحويل الجزء الكسري، ينطبق أيضا على النظامين الثنائي والستة عشري.

مثال: اوجد المكافئ الثنائي للعدد الستة عشري $(3BF9,0D7)_{16}$ ؟

الجواب: $(3BF9, 0D7)_{16} = (0011\ 1011\ 1111\ 1001.\ 0000\ 1101\ 0111)_2$

مثال: اوجد المكافئ الثماني والستة عشري للعدد الثنائي: $(1100101.\ 10011)_2$؟

الجواب: $(1100101.10011)_2 = (001\ 100\ 101.100\ 011) = (145,43)_8$

$(1100101.10011)_2 = (0110\ 0101.1001\ 0001) = (65,91)_{16}$

١-٤ أسئلة وتمارين

١- اكتب قائمة بالأعداد الثمانية والتساعية والست عشرية المقابلة للأعداد العشرية، بدءاً من 10 ولغاية 35.

٢- حول العدد الستة عشري 1A05C إلى ما يقابله بكل من النظام الثنائي والثماني والعشري؟

٣- حول العدد الثنائي 1011 101 إلى ما يقابله بكل من النظام الثماني والعشري والستة عشري؟

٤- ماهو أكبر عدد ثنائي يمكن تمثيله بوساطة 10 بتّ (خانات ثنائية)؟ وما هو المكافئ الثماني والعشري والست عشري؟

٥- مثّل كل من الأعداد التالية: $(2B.24)_{16}$، $(10010.0101)_{10}$ ، $(26.5)_8$ في النظام الثنائي ؟

٦- اوجد القيمة المكافئة للعدد الثنائي $(11101.0011)_2$ بكل من الأنظمة: الثمانية والعشرية والستة عشرية ؟

٧- ما هو المكافئ الثنائي والعشري والستة عشري لأكبر عدد ثماني ممثل بواسطة أربعة أرقام ثمانية؟

٨- حول الأعداد التالية بالأساس المبين بجانب كل منها $(426)_6$، $(197)_{12}$ إلى ما يكافئها في النظام العشري.

٩- حول العدد العشري 4287 إلى ما يقابله في النظام الثلاثة عشري مستخدماً ما تراه مناسباً من رموز؟

١٠- عرف النظام السداسي وبيّن فيه المقابل للعدد العشري $(23)_{10}$؟

١١- كم هو عدد البتّ في منظومة تحتوي على 16 كيلو بايت ؟ أو على 4 ميغابايت ؟

١٢- كم هو عدد الكيلو بايت في منظومة تحتوي:

آ) عشرة خطوط عنونة؟ ب) ستة عشر خطاً للعنونة ؟