



الدارات المنطقية



منشورات جامعة دمشق

كلية الهندسة

الميكانيكية والكهربائية

الدارات المنطقية

الدكتور أحمد خضور

أستاذ مساعد في قسم الهندسة الطبية

1438 - 1439 هـ
2016 - 2017 م

جامعة دمشق



الصفحة	الفهرس
13	مقدمة
15	الفصل الأول: أنظمة العدّ Number Systems
15	1-1 الحاسوب الرقمي
18	2-1 أسس المنطق الرقمي
20	3-1 النظام الثنائي
21	4-1 النظامان الثماني والستة عشري
21	5-1 التحويل بين أنظمة العدّ
30	6-1 أسئلة وتمارين
33	الفصل الثاني: تمثيل المعطيات Data Representation
33	1-2 أنواع المعطيات وتمثيلها
36	1-1-2 طريقة تمثيل الأعداد السالبة باستخدام المقدار والإشارة
38	2-1-2 طريقة تمثيل الأعداد السالبة باستخدام المتممات
41	2-2 تمثيل الأعداد بالفاصلة العشرية
42	1-2-2 تمثيل الأعداد بالفاصلة الثابتة
43	2-2-2 تمثيل الأعداد بالفاصلة العائمة
48	3-2 العمليات الحسابية على أنظمة العدّ
48	1-3-2 العمليات الحسابية في النظام الثنائي
51	2-3-2 العمليات الحسابية في النظام الثماني
53	3-3-2 العمليات الحسابية في النظام الستة عشري
55	4-2 أسئلة وتمارين

57	الفصل الثالث: الأرمزة Codes
57	1-3 الأرمزة العددية
58	1-1-3 الأرمزة الثنائية الموزنة
58	1-1-1-3 الرماز BCD
61	2-1-1-3 الرماز 8 4-2-1
63	2-1-3 الأرمزة الثنائية غير الموزنة
63	1-2-1-3 الرماز Excess - 3
65	2-2-1-3 رماز كراي Gray Code
66	2-3 الأرمزة الأبجدية الرقمية
66	1-2-3 رماز ASCII
69	3-3 أرمزة ضغط البيانات وتشفيرها
70	4-3 أرمزة اكتشاف الأخطاء وتصحيحها
71	1-4-3 أرمزة كشف الخطأ
71	1-1-4-3 الندية Parity
72	2-1-4-3 فاحص الاحتياط التكراري CRC
75	2-4-3 أرمزة تصحيح الخطأ
77	1-2-4-3 رماز هامينغ Hamming code
83	5-3 أسئلة وتمارين
85	الفصل الرابع: الجبر البولياني والتوابع المنطقية
85	1-4 جبر بول
85	1-1-4 قوانين بول
88	2-1-4 اختصار العبارات المنطقية
88	3-1-4 إثبات صحة العبارات المنطقية

90	2-4 التابع المنطقي
93	1-2-4 أشكال التابع المنطقي
96	2-2-4 التحويل بين الأشكال القانونية للتوابع المنطقية
98	3-4 البوابات المنطقية الأساسية
105	4-4 الشبكات المنطقية
107	1-4-4 الشبكات المنطقية المقابلة للأشكال القياسية للتوابع المنطقية
109	2-4-4 شبكات NAND و NOR
112	3-4-4 التأخير الزمني في الشبكات المنطقية
115	5-4 أسئلة وتمارين
117	الفصل الخامس: تمثيل التوابع المنطقية واختصارها
117	1-5 تمثيل المتحولات المنطقية باستخدام المخططات
121	1-1-5 مخطط كارنوف والأشكال القياسية للتابع المنطقي
123	2-1-5 حالات عدم الاكتراث (التعين)
125	2-5 اختصار التوابع المنطقية
126	1-2-5 الطريقة الجبرية
127	2-2-5 طريقة المخططات
134	1-2-2-5 استخدام حالات عدم الاكتراث باختصار التابع المنطقي
137	3-2-5 الطريقة الجدولية
142	3-5 أسئلة وتمارين
143	الفصل السادس: التنفيذ الفيزيائي للعناصر المنطقية
143	1-6 الدارات التلامسية Contact's Logic Circuits
144	2-6 شروط ربط العناصر المنطقية

146	3-6 خصائص العناصر المنطقية
150	4-6 تقانات تشكيل العناصر المنطقية
161	5-6 التقانة (العائلة) TTL
163	1-5-6 العناصر المنطقية الأساسية بالمنطق TTL
165	2-5-6 تراكيب المنطق TTL الخاصة
165	1-2-5-6 عناصر TTL بمجمع مفتوح
169	2-2-5-6 عناصر TTL التوسعية
170	3-2-5-6 عناصر TTL بثلاث حالات
172	3-5-6 مبادئ التعامل مع عناصر العائلة TTL
173	6-6 تقانات أحادي القطبية بتشكيل العناصر المنطقية
175	1-6-6 الأحمال بتقانات أحادي القطبية
176	2-6-6 التقانات PMOS, NMOS, CMOS
179	1-2-6-6 بوابة NOT بتقانات MOS
180	2-2-6-6 البوابتين الأساسيتين NAND و NOR بتقانات MOS
181	7-6 الموازنة بين عناصر ثنائية القطبية و أحادية القطبية
183	8-6 أسئلة وتمارين
185	الفصل السابع: الدارات المتكاملة
185	1-7 تصنيف الدارات المتكاملة
189	2-7 خصائص الدارات المتكاملة
197	3-7 تغليف الدارات المتكاملة
200	4-7 الدارات المتكاملة التمثيلية
202	5-7 استخدام الدارات المتكاملة
204	6-7 أسئلة وتمارين

205	الفصل الثامن: الدارات المنطقية التركيبية
205	1-8 تصميم الدارات المنطقية التركيبية
208	2-8 تطبيقات متفرقة باستخدام الدارات المنطقية التركيبية
216	3-8 Addition circuits دارات الجمع
216	1-3-8 Half Adder نصف الجامع
218	2-3-8 Full Adder الجامع التام
220	3-3-8 Parallel binary adder الجامع الثنائي المتوازي
221	4-8 Subtraction circuits دارات الطرح
223	5-8 الجامع / الطارح الثنائي المتوازي
230	6-8 تطبيقات على الجمع / الطرح الثنائي
236	7-8 Decimal adder/ subtractor الجامع / الطارح العشري
241	8-8 Decoder مفكك الترميز
244	9-8 Encoder المرمز
246	10-8 تطبيقات على استخدام الرموزات ومفككات الترميز
251	11-8 Multiplexers / Demultiplexer المضمّمات والمفركات
255	12-8 Comparator المقارن
259	13-8 Converters محولات الأرقام
263	14-8 أسئلة وتمارين
265	الفصل التاسع: الدارات المنطقية المتتابعة Sequential Circuits
265	1-9 المفهوم العام للدارات المتتابعة
268	2-9 الدارات ثنائية وأحادية وعديمة الاستقرار
271	3-9 Flip-Flops القلابات
272	1-3-9 R-S القلاب

288	D 2-3-9 القلاب
293	T 3-3-9 القلاب
296	J-K 4-3-9 القلاب
301	4-9 أسئلة وتمارين
303	الفصل العاشر: السجلات Registers
304	1-10 سجلات الإزاحة
305	1-1-10 سجلات الإزاحة نحو اليمين
306	2-1-10 سجلات الإزاحة نحو اليسار
306	3-1-10 سجلات الإزاحة الحلقية
308	2-10 الأنواع الأساسية للسجلات
308	1-2-10 سجل يدخل تسلسلي - خرج تسلسلي
310	2-2-10 سجل يدخل تسلسلي - خرج تفرعي
313	3-2-10 سجل يدخل تفرعي - خرج تفرعي
312	4-2-10 سجل يدخل تفرعي - خرج تسلسلي
314	3-10 سجلات الإزاحة ذات الاتجاهين
315	4-10 السجلات بالتحميل المتوازي
317	5-10 تطبيقات على استخدام السجلات
324	6-10 أسئلة وتمارين
325	الفصل الحادي عشر: العدادات Counters
330	1-11 العدادات غير المتزامنة Asynchronous Counters
331	1-1-11 العدادات غير المتزامنة التصاعدية
332	2-1-11 العدادات غير المتزامنة التنازلية

334	3-1-11 العدادات غير المتزامنة التصاعدية - التنازلية
335	4-1-11 العدادات مقاس N
336	5-1-11 مراحل تصميم العدادات غير المتزامنة
340	6-1-11 تطبيقات على استخدام العدادات غير المتزامنة
343	2-11 العدادات المتزامنة
345	1-2-11 مراحل تصميم العدادات المتزامنة
348	2-2-11 العدادات المتزامنة التصاعدية - التنازلية
350	3-2-11 العدادات ذات الدخل المتوازي
353	3-11 قطر العدادات Counter Cascading
354	4-11 العدادات الخاصة
358	5-11 مولدات النبضات الساعية
361	6-11 أسئلة وتمارين
363	الفصل الثاني عشر: تحليل الدارات المتتابعة وتصميمها
364	1-12 الدارات المتتابعة المتزامنة
367	1-1-12 تحليل الدارات المتتابعة المتزامنة
368	1-1-1-12 بناء جدول الحالة State table
369	2-1-1-12 تحديد معادلات الحالة State equations
369	3-1-1-12 رسم مخطط الحالة State diagram
370	4-1-1-12 بناء جداول التهييج للقلابات Excitation table
372	5-1-1-12 اختصار الحالات وإسنادها
378	2-1-12 تصميم الدارات المتتابعة المتزامنة
383	2-12 الدارات المتتابعة غير المتزامنة
385	1-2-12 تحليل الدارات المتتابعة غير المتزامنة بوجود التغذية الخلفية

402	2-2-12 الدارات المتتابعة غير المتزامنة بوجود القلايات
405	3-2-12 Hazards المخاطر
408	4-2-12 تصميم الدارات المتتابعة غير المتزامنة
412	3-12 أسئلة وتمارين
	الفصل الثالث عشر: استخدام لغة Verilog HDL في تصميم الدارات المنطقية
413	1-13 لغة Verilog HDL
418	1-1-13 مؤثرات لغة Verilog
420	2-1-13 تقنيات النمذجة
425	3-1-13 تمثيل التعبيرات البوليانية
425	2-13 وظائف Verilog للإظهار
426	3-13 الحلقات Loops
428	4-13 منصة الاختبار Test Bench
432	5-13 لغة وصف العتاديات للدارات التركيبية
439	6-13 لغة وصف العتاديات للدارات المتتابعة
447	الفصل الرابع عشر: المنطق القابل للبرمجة
447	1-14 التجهيزات التركيبية المنطقية القابلة للبرمجة
448	1-1-14 PLA التشكيل
452	2-1-14 PAL التشكيل
454	2-14 التجهيزات المتتابعة المنطقية القابلة للبرمجة
457	الفصل الخامس عشر: القسم العملي
471	المراجع
473	الملاحق

مقدمة

يتناول كتاب الدارات المنطقية بعض المعايير النظرية والأسس اللازمة لتصميم الدارات الرقمية. ويعتمد مبادئ المنطق بالشكل الذي يساعد في استخدام قواعد التصميم المنطقي لبناء الدارات الأساسية المكونة للأنظمة الرقمية.

قسم هذا الكتاب في اصداره الثاني إلى خمسة عشر فصلاً وثلاثة ملاحق. تشكل الفصول الخمسة الأولى مدخلاً للدارات المنطقية، حيث إن الفصل الأول يعرض بعض المفاهيم الأساسية المتعلقة بأنظمة العد. ويتناول الفصل الثاني آلية تمثيل المعطيات الثنائية بنوعيتها المؤشرة وغير المؤشرة، وكيفية إجراء العمليات الحسابية الأساسية في كل من هذه الأنظمة. يشير الفصل الثالث إلى ترميز المعطيات بأرمزة مختلفة، حيث تم استعراض عينة عن أهم الأرمزة Codes المعمول بها في الدارات الرقمية. ويتطرق الفصل الرابع لمفهوم الجبر البوليني، ومفهوم التتابع والبوابات المنطقية. أما الفصل الخامس، فيركز على أشكال تمثيل التتابع المنطقية وطرق اختصارها. يشكل الفصلان السادس والسابع وصفاً لتكنولوجيا تصنيع الدارات المنطقية، حيث يتناول الفصل السادس شكل التحقيق (الانجاز) الفيزيائي للعناصر المنطقية من خلال استعراض بعض التقانات الإلكترونية للتصنيع. ويدور الفصل السابع حول الدارات المتكاملة بوصفها منتجاً جاهزاً، ويشير بشكل أساسي إلى ورقة البيانات التي تظهر المواصفات الإجمالية للدارة المتكاملة واستخدامها في التطبيقات العملية المختلفة. يتضمن الفصل الثامن تعريفاً بالمنطق التركيبي وفيه يتم توصيف أهم الدارات التركيبية لإجراء العمليات الحسابية، وكذلك التعريف ببعض الدارات التي يكاد لا يخلو منها أي مخطط منطقي مثل المرمز ومفكك الترميز والمجمع والمقارن وغيرها من الدارات الأساسية. ويقتصر الفصل التاسع على التعريف العام بالمنطق التتابعي ويغطي توصيف الأنواع الأربعة الأساسية للقلابات من مختلف الجوانب. أما الفصل العاشر فيتناول موضوع السجلات Registers. ويتناول الفصل الحادي عشر موضوع العدادات Counters بنوعيتها المتزامن وغير المتزامن، إضافة إلى بعض الأنواع الخاصة من العدادات ومولدات الإشارة. يتطرق الفصل الثاني عشر لتحليل الدارات التتابعية بنوعيتها المتزامن وغير المتزامن وآلية تصميمها. أما الفصل الثالث عشر فقد خصص للتعريف بلغة Verilog HDL لتوصيف

الدارات المنطقية وتصميمها بمعونة الحاسوب، وهي واحدة من أهم لغات وصف العتاديات HDL المستخدمة بتصميم الدارات المنطقية. يظهر الفصل الرابع عشر كيفية تصميم النظم الرقمية باستخدام المنطق القابل للبرمجة. وقد خصص الفصل الخامس عشر للقسم العملي للكتاب (عرض لبعض تجارب المخبر). يضم الكتاب العديد من الجداول والأشكال التي تساعد المتتبع على فهم المواضيع التي يتناولها الكتاب بسهولة، وتحليل بعض المخططات. كما يضم العديد من الأمثلة المحولة، علاوة على وجود بعض الاسئلة متعددة الاجابات والتمارين غير المحولة في نهاية بعض الفصول.

أهم ما يميز الأصدار الثاني لهذا الكتاب عن الإصدار الأول، يتمثل في إلغاء فصل كامل حول التعريف بالأنواع الأساسية للذاكرة وكيفية تصميم كتلة من الذاكرة، والاستعاضة عنه بفصل جديد يمثل الجانب العملي لمقرر الدارات المنطقية ويضم اقتراحات لبعض التجارب المخبرية وتعريف أولي لكل من هذه التجارب. وقد تم أيضاً تعديل فقرة أسئلة وتمارين من معظم فصول الكتاب بإضافة بعض الاسئلة متعددة الاجابات. كما أنه تم إلغاء الملحق I من الإصدار الأول والذي كان يضم جدولين ببعض العناصر والدارات المتكاملة والاستعاضة عنه بملحقين يضم الأول قائمة بالجداول والثاني قائمة بالأشكال الواردة في الكتاب. علاوة على حذف بعض الفقرات وإضافة فقرات جديدة وتعديل الكثير من فقرات الإصدار الأول بما ينسجم مع آلية تدريس المقرر بالوقت المخصص له، وتخفيض عدد صفحات الإصدار الثاني بما يزيد قليلاً عن تسعين صفحة عن الأصدار الأول.

الفصل الأول: أنظمة العدّ Number Systems

بالنظر إلى أن أنظمة العدّ تستخدم في تمثيل البيانات بالحاسوب وفي قواعد التصميم المنطقي، نذكر بداية بالمكونات الأساسية للحاسوب الرقمي قبل المباشرة بالتعرّف على أنظمة العدّ الأساسية

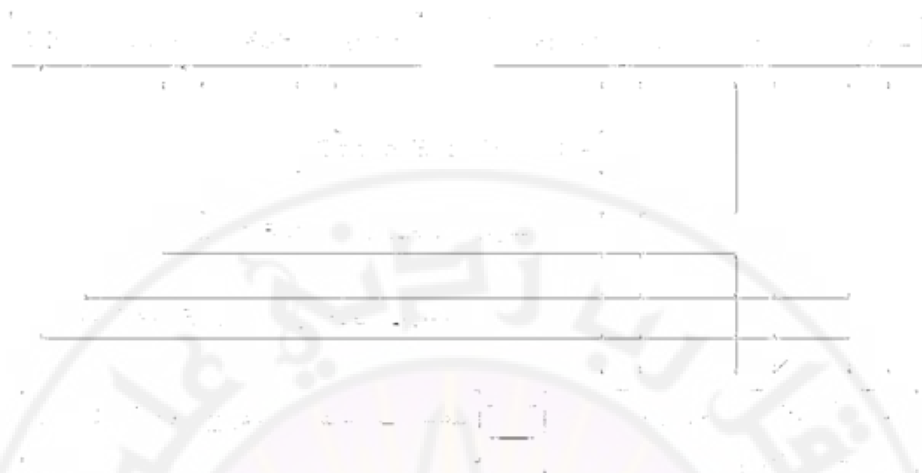
1-1 الحاسوب الرقمي

يُعرف الحاسوب الرقمي Digital computer بأنه الجهاز الذي يقوم بإجراء سلسلة من العمليات الرياضية والمنطقية على مجموعة من البيانات للحصول على المعلومات اللازمة لاتخاذ القرارات المناسبة. ويمكن النظر للحاسوب على أنه جهاز إلكتروني متعدد الوظائف، يقوم بأي عمل يطلب منه وفق قواعد يحددها له المستخدم بواسطة برنامج Program الذي يعبّر بدوره عن الخوارزمية التي توصف المسألة المطلوب معالجتها. إن 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 أن تخزنها والاسترجاع السريع للمعلومات المخزنة، علاوة على قابليتها للتعديل والتطوير بإمكانية ربطها بالعديد من الأجهزة المحيطية الأخرى. يتكون أي حاسوب من جزأين أساسيين متممين لبعضهما هما:

- الجزء التقني (أو العتاد) Hardware الذي يمثل كلّ ما يمكن تصنيعه مادياً من عناصر وبطاقات إلكترونية لإدخال ومعالجة وإخراج المعلومات.

- الجزء البرمجي Software الذي يشمل كلّ ما هو غير ملموس وضروري لعمل الحاسوب من تعليمات وبيانات يحتاجها لأداء مهامه.

يمكن تصور التركيبة التقنية للحاسوب وفق نموذج Havard بأن كلّ حاسوب يتكون بالإضافة إلى وحدة المعالجة المركزية من ذاكرة يخصص قسم منها للبرامج وقسم منفصل للمعطيات. أما النموذج الأعم للحواسيب فيتمثل بنموذج Von Neumann والذي يعني أن كلّ حاسوب يتكون بالإضافة إلى وحدة المعالجة المركزية من ذاكرة (للبرامج والمعطيات) وبوابات للدخل والخرج ومجموعة من السواقات وفق ما هو مبين بالشكل 1-1.



الشكل 1-1: التركيبة العامة للحاسوب الرقمي

يظهر الشكل 1-1 أن كل حاسوب رقمي يتكون بشكل عام من أربع وحدات رئيسية هي:

- 1- وحدة المعالجة المركزية (CPU) Central Processing Unit ممثلة بالمعالج الصغري Microprocessor، الذي يمثل الأساس بالنسبة لهذه الكتلة، وبالتالي بالنسبة لكامل الحاسوب. يتكون كل معالج بدوره من عدة وحدات لاحاجة لذكرها الان.

- 2- وحدة الذاكرة Memory Unit لتخزين المعطيات والبرامج، وتمثل أيضاً ساحة العمل للمعالج أثناء قيامه بتنفيذ برنامج ما، وهي تتكون بدورها من جزأين:

الأول- لتخزين البرامج الأساسية المرتبطة بالعتاد التي تضم تعليمات تشغيل الحاسوب وهي من النوع الدائم (غير المتطاير Non-Volatile) مخصصة للقراءة فقط ROM (Read Only Memory)

الثاني- لتخزين البرامج والمعطيات العامة، وهي من النوع المؤقت (المتطاير Volatile)
تفقد كل محتوياتها بمجرد انقطاع التيار الكهربائي، وهي متاحة للمستخدم للولوج إليها
بشكل عشوائي (قراءة وكتابة) وتسمى RAM (Random Access Memory)

- ### 3- 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. وتجهز هذه الوحدة عادة بعدد من المداخل

والمخارج تسمى بوابات ونميز بين نوعين من البوابات: - بوابات للإدخال Input port
- بوابات للإخراج Output port

يمكن من خلال البوابة الواحدة نقل مجموعة من البيانات الثنائية من خلال خط واحد أو من خلال عدة خطوط بأن واحد، تربط بالجهاز المحيطي (بحسب نوعه)، فإذا تم نقل المعطيات عبر عدة خطوط من البوابة الواحدة بأن واحد، تسمى البوابة Parallel port أي تفرعيه. وإذا قامت البوابة بنقل البيانات من أو إلى الجهاز المحيطي على خط اتصال واحد بشكل متتابعي، تسمى بوابة تسلسلية Serial port. بالطبع النقل التسلسلي للبيانات يكون بطيئاً ويستخدم للنقل إلى مسافات بعيدة ، أما النقل التفرعي فهو سريع إلا أنه مكلف ويستخدم للنقل إلى مسافات قصيرة.

4- وحدة النواقل أو الممرات Bus Unit التي يتم عبرها نقل مختلف أشكال البيانات من معطيات وعناوين وإشارات تحكم مختلفة بين مختلف مكونات الحاسوب الأساسية. تستخدم في الحاسوب الرقمي لتمثيل المعلومات وحدة تدعى بت Bit، حيث البت الواحد يمثل أصغر وحدة لقياس المعلومات، وكل 4 bits تدعى Nibble.

يستخدم كل 8 bits 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 $1024 \text{ bytes} = 2^{10} \text{ bytes}$.

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

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

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

اضافة للوحدات المبينة أعلاه، يوجد بعض الوحدات الأخرى الأقل استخداماً لاداعي لذكرها.

1-2 أسس المنطق الرقمي

تمتاز النظم الرقمية بقدرتها على معالجة القيم المنقطعة Discrete ممثلة بالصيغة الثنائية Binary form، حيث تمثل القيم المنقطعة للمعلومات في النظم الرقمية بوساطة كميات فيزيائية، تدعى بالإشارات Signals. وتعدّ الإشارات الكهربائية مثل الجهد Voltage والتيار Current أكثر الإشارات شيوعاً. ولا يزال الترانزيستور هو العنصر الأساسي للتعامل مع هذه الإشارات في الدارات الإلكترونية. وتستجيب الدارات الإلكترونية (المنطقية) التي تعمل بالجهد لمستويين منفصلين من الإشارات لتمثيل القيم الممكنة لمتحول ثنائي، حيث يأخذ كلّ متحول ثنائي قيمتين (0 , 1) تقابل بدورها حالتَي الوصل والفصل (On و Off) للدارة المنطقية. ويُعبّر عن مجالات القيم التي تأخذها المتحولات الثنائية بوساطة نبضات كهربائية Pulses بالمستوى العالي (H High) وبالمستوى المنخفض (L Low)، حيث يوجد لكلّ مستوى جهد مجال مقبول للتغيير ضمنه، كما هو مبين في الشكل 1-2 أ.



الشكل 1-2: أنواع الإشارات الرقمية ومجالها

يرمز لجهد التغذية V_{cc} بالحرف H من High، ويتم التعامل معه على أنه يمثل واحداً منطقياً. ويرمز للأرضي GND بالجهد U_{ss} أو الحرف L من Low ويتم التعامل معه على أنه يمثل صفراً منطقياً. وبما أن قيمة الإشارة الرقمية على دخل أو خرج أي دارة رقمية تخصص بوحدة من قيمتين منطقيتين، توجد طريقتان مختلفتان لتخصيص مستوى الإشارة في الدارات الإلكترونية الرقمية.

الأولى: تسمى بالمنطق الموجب Positive logic، وفيه يتم اختيار المستوى العالي H لتمثيل المنطق 1 والمستوى المنخفض L لتمثيل المنطق 0، كما في الشكل 1-2 ب. الثانية: تسمى بالمنطق السالب Negative logic وهي عكس السابقة، ويتم فيها اختيار المستوى المنخفض L لتمثيل المنطق 1 والمستوى المرتفع H لتمثيل المنطق 0، كما هو مبين في الشكل 1-2 ج. بالرغم من أن هذا التصنيف إلى "منطق موجب ومنطق سالب" معمول به، إلا أنه قد يربك أحياناً، لأنه يمكن للإشارتين أن تكونا موجبتين أو سالبتين، بمعنى أن قيمة الإشارة ليست هي التي تحدد نوع المنطق (موجب أو سالب)، ولكن عملية إسناد القيم المنطقية إلى المطالات النسبية لمستويي الإشارتين هي التي تحدده. والمنطق الذي سيتم اعتماده في التوصيف من الآن وصاعداً هو المنطق الموجب مالم يتم الإشارة لذلك. يصادف في التطبيقات العملية أنواع مختلفة من الإشارات. فمنها ما هو مستمر مثل الإشارات الصوتية أو الجهد المتغير باستمرار، وتسمى بالإشارات التتميلية Analog. ومنها ما هو متقطع Discrete مثل النبضات الواردة من كاشف أو مفتاح كهربائي، وتسمى بالإشارات الرقمية Digital. يعبر عن الإشارات الرقمية بنظام عددي محدد، يسمى النظام الثنائي Binary system. ويستخدم هذا النظام لتمثيل مختلف أشكال الإشارات الرقمية بالقيمتين 0 أو 1. يوجد بالإضافة للنظام العشري المخصص للتعامل العادي بين البشر والنظام الثنائي المخصص لتمثيل المعالجة ضمن الحاسوب، أنظمة عدّ أخرى تستخدم لتمثيل المعطيات على مختلف أنواعها، مثل النظام الثماني والنظام الست عشري. يتميز كلّ نظام عدّ بعاملين رئيسيين هما: - أساس النظام base (اختصاراً ب)

- مكونات النظام Components

لا يُعدّ أساس نظام العدّ b من المكونات الأساسية للنظام، وبالتالي مكونات أي نظام عدّ تبدأ من الصفر، وتنتهي بالقيمة 1 - b. 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). تعتمد أنظمة العدّ الثنائية والثمانية والستة عشرية نفس الأسس المتبعة في النظام العشري. وبالتالي عرض المفاهيم المرتبطة في النظام العشري يساعد بفهم بقية أنظمة العدّ الأخرى. تتكون الأعداد في النظام العشري مهما كانت كبيرة أم

صغيرة أو صحيحة أم كسرية أو موجبة أم سالبة من تراكيب مختلفة للأرقام العشرية 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, لأن أساسه 10 وهذا يعني أن العدد العشري الذي سيتم التعامل معه سيكون ناتجاً عن ترتيب معين لبعض 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. بالتالي هذه المجموعة من الأرقام لتعبر 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 أن توضع في كل خانة من خانات العدد. أما عندما يتم تجاوز القيمة تسعة فإنه يتم الانتقال إلى استخدام القيمة الموضعية التي تدل على مكان وقوع الرقم بالنسبة إلى بقية الخانات. ويتم الحصول على القيمة الموضعية بضرب الأساس بنفسه عدد من المرات يساوي عدد الخانات المطلوبة لتمثيل قيمة العدد. بالنسبة للأعداد بالفاصلة العشرية، يجب فصل العدد إلى جزء صحيح Integer وجزء كسري Fraction part، لأن آلية التعامل مع كل منهما تختلف عن الآخر، حيث يعبر عن كل قيمة موضعية تقع على يمين الفاصلة بالقوة السالبة التي يرفع لها الأساس. فمثلاً العدد 23.741 يمكن التعبير عنه بالشكل:

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

1-3 النظام الثنائي

بما أن معظم التجهيزات الإلكترونية بما فيها الحواسيب تعتمد على وجود المكونات الإلكترونية الأساسية مثل: الترانزستور والحلقات المغناطيسية أو الدارات المتكاملة التي كل منها يمر في حالتين فقط (فالترانزستور يكون في حالة نقل التيار الكهربائي (حالة الفتح) أو عدم نقله (حالة الإغلاق))، ويسمى النظام العددي الذي يوصف التعامل مع مثل هذه الدارات الرقمية، بالنظام الثنائي. يُعدّ العدد 2 أساس النظام الثنائي والأرقام الصحيحة الأصغر من 2 فقط، أي الرقمين 0, 1 هما مكوناته. وبقيّة المفاهيم المتعلقة بتمثيل أي عدد في هذا النظام مماثلة لما ورد بالنسبة للنظام العشري.

1-4 النظام الثماني والستة عشري

يستخدم كل من النظام الثماني Octal 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 نظاماً عددياً أساسه 8 ومكوناته الأرقام: 0, 1, ..., 7. أما النظام الستة عشري فيمثل نظاماً عددياً أساسه 16 ومكوناته الأرقام: 0, 1, ..., 9, 10, ..., 15. وتجنباً لأي غموض بالنسبة للنظام الستة عشري بين الأرقام والأعداد، تستخدم بالنسبة للأرقام المكونة من خانتين (التي هي أكبر من 9)، أي: 10, 11, 12, 13, 14, 15 رموزاً أبجدية، مثل A, B, C, D, E, F على التوالي، حيث إن الرمز D مثلاً، يعادل الرقم 13 وينظر للكمية 13 على أنها وحدة غير قابلة للتجزئة، مثل أي رقم من 0, ..., 9 ولا ينظر لها على أنها واحد بالخانة الأولى وثلاثة بالخانة الثانية بل واحد وثلاث بنفس الخانة.

1-5 التحويل بين أنظمة العد

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 (اثنان)، ويعبر عنه بالشكل $(1011)_2$ أو ثمانياً ويعبر عنه بالشكل $(1011)_8$ أو عشرياً ويعبر عنه بالشكل $(1011)_{10}$. ويستخدم في بعض التطبيقات مثل البرمجة بلغة 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 بالمثل أعلاه يمكن التعبير عنه بالأشكال التالية: 1011B بدلاً من $(1011)_2$ للإشارة إلى أنه عدد ثنائي أو 1011O أو 1011D أو 1011H للدلالة على النظام

المقابل. وعند التعامل مع أي عدد طوله n خانة بأي نظام عددي يصطلح أيضاً على أن الخانة الأدنى وزناً في العدد تمثل دوماً بأقصى اليمين ويعبر عنها بالدليل 0 وأعلى خانة تمثل بأقصى اليسار، ويعبر عنها بالدليل $n-1$. فمثلاً بالنسبة للبايت المكون من 8 bits مثل البايت (1 1 0 1 0 1 0 0) يكتب كالتالي: رقم البت في البايت 7 6 5 4 3 2 1 0 قيمة البت في البايت 1 1 0 1 0 1 0 0 يطلق على الخانة ذات المرتبة 0 من أقصى اليمين بالعدد الثنائي $(1101\ 0100)_2$ مصطلح LSB، أو (Least Significant Bit or Digit) LSD، أي البت ذي الوزن الأدنى أو البت الأقل دلالة. وعلى الخانة ذات القيمة 1 من أقصى اليسار MSB أو MSD (Most Significant Bit or Digit) البت ذو الوزن الأعلى. وفيمل يلي ندرس مختلف التحويلات بين مختلف الأنظمة العددية وفق التالي:

أولاً- 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 للنواتج مرة أخرى على الأساس 2، وتستمر عملية القسمة حتى يتم الوصول إلى الحالة التي لا يمكن عندها التقسيم على الأساس 2. وتكتب بعد ذلك البواقي على التوالي من اليمين إلى اليسار، ابتداءً من أول باقٍ.

مثال: حول العدد الفردي (23) من النظام العشري إلى ما يقابله في النظام الثنائي.
الجواب: للتحويل تطبيق القاعدة أعلاه، فيكون:

$$23/2 = 11 \text{ Remainder} = 1 \text{ (First remainder)}$$

$$11/2 = 5 \text{ Remainder} = 1$$

$$5/2 = 2 \text{ Remainder} = 1$$

$$2/2 = 1 \text{ Remainder} = 0$$

$$\text{Last Result} = 1 < 2 \text{ (base)} \sim \text{remainder} = 1$$

يكون ترتيب البواقي من اليمين إلى اليسار ابتداءً من أول باقٍ وفق الشكل التالي:
(10111)₂، وتكتب نتيجة التحويل كالتالي: $(23)_{10} = (10111)_2$.

يمكن 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 أو 1. ويمكن بخانتين تمثيل أربعة قيم ثنائية هي 00, 01, 10, 11، وهذا ما يقابل أعداداً عشرية من 0 حتى 3، وبثلاث خانات يمكن تمثيل 8 قيم، وهكذا من أجل n خانة يمكن تمثيل أعداد عشرية من 0 حتى $2^n - 1$. فإذا تم التعبير عن الخانات بالأوزان المقابلة لها وفق الشكل $2^0, 2^1, 2^2, 2^3, \dots$ ، يمكن تمثيل أي عدد عشري بالطريقة المباشرة بوضع قيمة واحد في الموضع الذي يلزم لتشكيل العدد المعطى وصفر في الموضع الذي لا يلزم. مثال: يمكن التعبير عن العدد العشري $(41)_{10}$ بالشكل التالي:

- الأوزان : $2^0, 2^1, 2^2, 2^3, 2^4, 2^5$

- القيمة الموضعية : 1 2 4 8 16 32

- مكونات العدد : 1 8 32

- القيمة المطلقة : 1 0 0 1 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 مجموع القيم 1, 8, 32 حصراً. مثال: أوجد المقابل الثنائي للعدد العشري $(154)_{10}$. للجواب بالطريقة المباشرة ننطلق من أن أقرب قيمة من قيم 2^n للعدد المعطى $(154)_{10}$ هو القيمة $128 = 2^7$ فنضع 1 تحت القيمة 2^7 وكأننا أخذنا من 154 القيمة 128 فيصبح الباقي من العدد المعطى 26 نوزعه على الأوزان الأدنى وفق الشكل التالي: $(1 0 0 1 10 1 0)_2 = (154)_{10}$ وهو المطلوب. ولتحويل العدد العشري الكسري إلى ما يقابله في النظام الثنائي، فإن 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. ويسجل بعد ذلك النقل الناتج ابتداء من الفاصلة باتجاه اليمين. بالطبع لا يدخل النقل Carry في عملية الضرب التالية. وفي حال عدم وجود نقل يعتمد الصفر نقل.

مثال : حول العدد $(13.40625)_{10}$ إلى ما يقابله في النظام الثنائي؟

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

الكسري للشكل الثنائي وفق التالي:

(Carry) (Number)

$$\begin{array}{r}
 0.40625 \\
 \times 2 \\
 \hline
 0.81250 \\
 \times 2 \\
 \hline
 1.62500 \\
 \times 2 \\
 \hline
 1.25000 \\
 \times 2 \\
 \hline
 0.50000 \\
 \times 2 \\
 \hline
 1.00000
 \end{array}$$

يكون الناتج المقابل للجزء الكسري: $(.01101)_2 = (0.40625)_{10}$. ويصبح الناتج النهائي لعملية التحويل: $(1101.01101)_2 = (13.40625)_{10}$. 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 أنه مهما يضرب الجزء الكسري بالعدد (2) لا يتم الحصول على الصفر. يمثل 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 $(22,15)_{10}$ إلى ما يقابله في النظام الثنائي، يلاحظ إن الجزء الكسري يقابل القيمة غير المنتهية $(.001\ 0011\ 0011\ 0011\dots)_2$ ، أي أن النتيجة بعد أول ثلاثة أرقام 001، ستتكرر القيمة 0011، وبالتالي يكفي بالجواب كتابة هذه السلسلة لمرة واحدة وفق الشكل: $(10110.001\ 0011)_2 = (22,15)_{10}$ أو أن يقرب الناتج إلى أربع خانات مثلاً ويكتب الجواب النهائي بالشكل: $(10110.0010)_2 = (22,15)_{10}$.

ثانياً- التحويل من النظام الثنائي إلى النظام العشري

لتحويل أي عدد ينتج من الدارات الرقمية التي تتعامل في النظام الثنائي إلى النظام العشري، يتم الانطلاق من الشكل العام لكثير الحدود الذي يشير إلى نشر قيمة عدد ما N وفق الشكل: $N = a_0b^0 + a_1b^1 + \dots + a_{n-1}b^{n-1}$ ، بحيث يمكن تحويل أي عدد ثنائي إلى ما يقابله في النظام العشري بجمع أوزان خانات العدد الثنائي المحتوية على واحد فقط.

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

الجواب: العدد مكون من ست خانات، توزع على الأوزان المقابلة لكل منها كالتالي:

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

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

$$(101110)_2 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2 + 1 \times 2^3 + 0 \times 2^4 + 1 \times 2^5$$

$$= 0 + 2 + 4 + 8 + 0 + 32 = (46)_{10}$$

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

مثال : أوجد القيمة العشرية للعدد الثنائي $(11001.1011)_2$ ؟

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

$$N = (11001.1011)_2 = (1.2^0 + 0.2^1 + 0.2^2 + 1.2^3 + 1.2^4). (1.2^{-1} + 0.2^{-2} + 1.2^{-3} + 1.2^{-4}) = (1+8+16). (1/2 + 1/8 + 1/16) = (25.687)_{10}$$

ثالثاً- التحويل من النظام العشري إلى النظام الثماني

عند تحويل عدد ما من النظام العشري إلى ما يقابله بالنظام الثماني يتم التقسيم على قيمة الأساس 8 (بدلاً من التقسيم على 2 بالنسبة للنظام الثنائي)، 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 $(?)_8 = (397)_{10}$.

الجواب: $397/8 = 49 \text{ Remainder} = 5$

$49/8 = 6 \text{ Remainder} = 1$

بما أن الناتج الأخير 6 أصغر من الأساس 8، فإنه يُعد باقياً أيضاً. وتكون البواقي هي الرقم 5 بوصفه أول باقٍ ويكتب بأقصى يمين الناتج، يليه الرقم 1 وأخيراً الرقم 6، ويصبح الجواب

النهائي من الشكل: $(615)_8 = (397)_{10}$. أما بخصوص الأعداد الكسرية فتحول بنفس الآلية التي تم فيها تحويل عدد من نظام عشري إلى ثنائي بعد مراعاة قيمة الأساس. مثال: أوجد 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 ضرب بقيمة الأساس 8 ويسجل المنقول وتستمر عملية الضرب حتى يتم الحصول على الناتج صفر وفق ما هو مبين أدناه:

$$0.513 \times 8 = 4.104$$

$$0.104 \times 8 = 0.832$$

$$0.832 \times 8 = 6.656$$

$$0.656 \times 8 = 5.248$$

$$0.248 \times 8 = 1.984$$

$$0.984 \times 8 = 7.872$$

بما أن العدد المعطى لا يصل بنتيجة الضرب المتكرر بالأساس 8 إلى الصفر، يكفي بهذه الحالة التقريب إلى عدد ما من الخانات. فإذا تم الاكتفاء بأول خمس خانات مثلاً، يصبح

$$(0,513)_{10} = (0,40651)_8$$

- الجزء الصحيح يعادل: $75 / 8 = 9 \text{ Remainder} = 3$

$$9 / 8 = 1 \text{ Remainder} = 1$$

Last Result = 1 is reminder

بالتالي $(75)_{10} = (113)_8$ ، ويصبح الناتج النهائي: $(75,513)_{10} = (113,40651)_8$.

رابعاً- التحويل من النظام الثماني إلى النظام العشري

تعدّ عملية التحويل من النظام الثماني إلى العشري (Octal to Decimal Conversion) معاكسة لعملية تحويل العشري إلى الثماني وهي ممانلة لعملية التحويل من الثنائي للعشري.

مثال : حول العدد (761) في النظام الثماني إلى ما يقابله في النظام العشري.

$$(761)_8 = 1 \cdot 8^0 + 6 \cdot 8^1 + 7 \cdot 8^2 = (497)_{10}$$

فاصلة فالمبدأ نفسه، إذ نرفع الأساس 8 قبل الفاصلة بأس موجب وبعد الفاصلة بأس

سالب بحسب موضعه.

خامساً- التحويل من النظام العشري إلى النظام الستة عشري

بأسلوب مماثل للتحويل من النظام العشري إلى كل من النظامين الثنائي والثماني، يتم التحويل من النظام العشري إلى النظام الستة عشري بعد مراعاة قيمة الأساس والمكونات.

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 حتى 15. مع مراعاة أن البواقي من 10 حتى 15 يكتب عوضاً عنها الرمز المقابل لكل منها من A حتى F.

$$3915/16 = 244 \text{ Remainder} = 11 \sim B$$

$$244/16 = 15 \text{ Remainder} = 4$$

$$\text{Last Result} = 15 \sim F (< 16 \text{ (base)})$$

يكون الجواب: $(3915)_{10} = (F4B)_{16}$.

سادساً- 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:

$$N = a_0b^0 + a_1b^1 + \dots + a_{n-1}b^{n-1} \text{ بعد جعل الأساس } b = 16$$

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

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

$$= 9.16^0 + 10.16^1 + 1.256 = (425)_{10}$$

سابعاً- التحويل بين الأنظمة الثنائية والثمانية والستة عشرية

بما أنه يمكن كتابة أساس النظام الثنائي بالشكل 2^1 والثماني بالشكل 2^3 والستة عشري بالشكل 2^4 ، هذا يعني أنه إذا كانت الأعداد ممثلة ثنائياً، فإن كل خانة (رقم) في النظام الثماني تقابل ثلاث خانات في النظام الثنائي وكل خانة في النظام الستة عشري تقابل أربع خانات في النظام الثنائي. يمكن باستخدام ذلك إجراء عملية التحويل من النظام الثماني إلى الثنائي وبالعكس وفق ما هو مبين 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: $(437)_8 = (100\ 011\ 111)_2$. يمكن إجراء عملية التحويل مباشرة من النظام الثنائي إلى الثماني وفق المثال التالي: مثال: حول العدد الثنائي $(10\ 110\ 101)_2$ إلى ما يقابله بالنظام الثماني. ويكون الجواب بالطبع من أجل كل عدد ثنائي، يمكن تجميع كل ثلاثة بتات مع بعضها ابتداء من اليمين وكتابة المقابل الثماني لها، لأنه بثلاث خانات يمكن على الأكثر تمثيل التراكيب من 000 حتى 111 التي تعادل الأرقام من 0 حتى الرقم 7 وهي مكونات النظام الثماني، أي إن: $(5\ 6\ 2)_8 = (101\ 110\ 010)$. ويمكن بنفس الطريقة تحويل أي عدد في النظام الستة عشري إلى ما يقابله في النظام الثنائي وبالعكس بتخصيص كل رقم ستة عشري بأربع خانات ثنائية.

مثال: حول العدد الستة عشري $(B5)_{16}$ إلى ما يقابله في الثنائي $(?)_2 = (B5)_{16}$ الجواب: بما أن: $(5)_{16} = (0101)_2$ و $(B)_{16} = (1011)_2$ ، يصبح الجواب مباشرة من الشكل التالي: $(B5)_{16} = (1011\ 0101)_2$ والعكس صحيح.

يلاحظ مما سبق أهمية استخدام كل من النظامين الثماني والستة عشري باختزال عدد خانات أي عدد ثنائي إلى الثلث أو إلى الربع. أما بخصوص التحويل من النظام الثماني إلى النظام الستة عشري وبالعكس. العملية غير هامة، لأن التحويل بين هذين النظامين غير مستخدم تقريباً. وعلى الرغم من 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: $(1A4)_{16} = (1\ 1010\ 0100)_2$

$$(1A4)_{16} = 4 \times 16^0 + A \times 16^1 + 1 \times 16^2 = (420)_{10}$$

$$(1A4)_{16} = (1\ 1010\ 0100)_2 = (110)(100)(100) = (644)_8$$

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

مثال: أوجد المكافئ الثماني لعدد ثنائي من الشكل: $(10101001111.110011101)_2$.
 الجواب: يجمع كل من جزئي العدد (الصحيح والكسري) 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:

$$(010 \ 101 \ 001 \ 111 \ 110 \ 011 \ 101)_2$$

$$2 \ 5 \ 1 \ 7 \ 6 \ 3 \ 5$$

أي أن العدد الثنائي المعطى يعادل $(2517,635)_8$. وما ينطبق على النظامين الثنائي والثماني بخصوص تحويل الجزء الكسري، ينطبق على النظامين الثنائي والستة عشري.
 مثال: أوجد المكافئ الثنائي للعدد الستة عشري $(3BF9.0D7)_{16}$.

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

مثال: أوجد المكافئ الثماني والستة عشري للعدد الثنائي: $(1100101.10011)_2$ ، الجواب:

$$(1100101.10011)_2 = (001 \ 100 \ 101.100 \ 011) = (145.43)_8$$

$$(1100101.10011)_2 = (0110 \ 0101.1001 \ 0001) = (65.91)_{16}$$

مثال: أوجد القيمة العشرية المقابلة للعدد الستة عشري $(2A9.8C)_{16}$.

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

$$(2A9.8C)_{16} = (2 \times 16^2 + A \times 16^1 + 9 \times 16^0) + (8 \times 16^{-1} + C \times 16^{-2})_{10}$$

$$= (512 + 160 + 9) + (8/6 + 12/256)_{10} = (681.140 / 256)_{10}$$

مثال: أوجد المقابل الثنائي والثماني والعشري والستة عشري للعدد العشري 1337.

الجواب: لإيجاد المقابل الثنائي للعدد المعطى ننطلق من نشر كثير حدود بعدد من الحدود يقارب مجموعها قيمة العدد، بمعنى أن أقرب 2^n للعدد 1337 يكون من أجل $n = 10$. إذاً

$$2^{10} \ 2^9 \ 2^8 \ 2^7 \ 2^6 \ 2^5 \ 2^4 \ 2^3 \ 2^2 \ 2^1 \ 2^0$$

$$1024 \ 512 \ 256 \ 128 \ 64 \ 32 \ 16 \ 8 \ 4 \ 2 \ 1$$

$$1 \ 0 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \rightarrow (10100111001)_2$$

أما المقابل الثماني، فيمكن الحصول عليه من تجميع كل ثلاث خانات بالثنائي بخانة واحدة بالثماني ابتداء من أقصى اليسار، فيكون الناتج: $(2471)_8 \rightarrow (10 \ 100 \ 111 \ 001)_2$.

والعشري: $1337 = 7 \cdot 10^0 + 3 \cdot 10^1 + 3 \cdot 10^2 + 1 \cdot 10^3 = 7 + 30 + 300 + 1000$

والستة عشري: $(101 \ 0011 \ 1001)_2 \rightarrow (539)_{16}$

1-6 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من تساؤلات التالية:

1- النيبيل Nibble يمثل ؟

- a- a string of 4 bits
- b- a string of 8 bits
- c- a string of 16 bits
- d- a string of 64 bits

2- الرمز K يمثل اختصار يستخدم مع عدد الوحدات Units وبالتالي 2K تعني

- a- 2000 units
- b- 2048 units
- c- 2024 units
- d- none of these

3- كل 1 MB يعادل

- a- 1000 Bytes
- b- 1000KB
- c- 1024KB
- d- none of all

4- أي من الأنظمة العددية التالية يستخدم لتمثيل البيانات في الحاسوب الرقمي

- a- Decimal
- b- Binary
- c- Octal
- d- Hexadecimal

5- العدد العشري 17 يقابل بالنظام الثنائي

- a- 10010
- b- 11000
- c- 10001
- d- 01001

6- العدد الثنائي 11011101 يقابل بالنظام العشري

- a- 121
- b- 221
- c- 441
- d- 256

7- القيمة الثنائية $(100101)_2$ تقابل

- a- $(37)_{10}$
- b- $(69)_{10}$
- c- $(41)_{10}$
- d- $(5)_{10}$

8- القيمة الستة عشرية $(64)_{16}$ تقابل

- a- $(1100100)_2$
- b- $(144)_8$
- c- $(100)_{10}$
- d- all the above

9- القيمة العشرية $(59)_{10}$ تقابل بالنظام الثنائي

- a- 1010101
- b- 1101101
- c- 1011001
- d- 1001101

10- القيمة الثنائية $(101101)_2$ تقابل بالنظام الثماني

- a- 65
- b- 55
- c- 51
- d- 45

- 11- القيمة الستة عشرية $(A09)_{16}$ تقابل
- a- $(101000001001)_2$ b- $(5011)_8$
c- $(2569)_{10}$ d- all the above

- 12- القيمة الثنائية $(1010101)_2$ تقابل
- a- $(321)_8$ b- $(57)_{10}$ c- $(A5)_{16}$ d- none of all

- 13- المكافئ العشري للعدد الستة عشري 1A53 هو
- a- 6793 b- 6739 c- 6973 d- 6379

- 14- $(734)_8 = (?)_{16}$
- a- C1D b- DC1 c- 1CD d- 1DC

- 15- القيمة 1011H تكافئ
- a- $(11)_D$ b- $(10021)_O$
c- $(10021)_D$ d- none of all

- 16- العنوان (أو القيمة) FF0F تكافئ
- a- $(1111\ 1111\ 0000\ 1111)_2$
b- $(177617)_O$
c- $(1111\ 1111\ 0000\ 1111)_B$
d- all the above

- 17- الأساس الصحيح للأرقام بالمعادلة $365 + 413 = 1000$ هو
- a- 8 b- 10 c- 16 d- none of all

- 18- الأساس 6 لنظام عددي جديد تحقق وفقه المعادلة التالية:
- a- $705 + 222 = 1331$ b- $513 + 314 = 1231$
c- $513 + 314 = 827$ d- none of all

ثانياً: أسئلة متفرقة

- 1- اكتب قائمة بالأعداد الثمانية والست عشرية المقابلة للأعداد العشرية، بدءاً من 10 ولغاية 35.
- 2- حول العدد الستة عشري 1A05C إلى ما يقابله بكل من النظام الثنائي والثماني.
- 3- حول العدد الثنائي 101 1011 إلى ما يقابله بكل من النظام الثماني والعشري والستة عشري.
- 4- ما هو أكبر عدد ثنائي يمكن تمثيله بواسطة 10 بت (خانات ثنائية)؟ وما هو المكافئ الثماني والعشري والست عشري المقابل له.
- 5- مثل كل من الأعداد: $(2B.24)_{16}$ ، $(10010.0101)_{10}$ ، $(26.5)_8$ بالنظام الثنائي.
- 6- أوجد القيمة المكافئة للعدد الثنائي $(11101. 0011)_2$ في كل من الأنظمة: الثمانية والعشرية والستة عشرية .
- 7- ما هو المكافئ الثنائي والعشري والستة عشري لأكبر عدد ثماني ممثل بواسطة أربعة أرقام ثمانية؟
- 8- حول الأعداد التالية بالأساس المبين بجانب كل منها $(426)_6$ ، $(197)_{12}$ إلى ما يكافئها في النظام العشري.
- 9- عرف نظام خماسي (الأساس والمكونات) وبيّن فيه المقابل للعدد العشري $(23)_{10}$.
- 10- حدّد الأساس المناسب للأرقام بالعلاقة $802 = 416 + 375$ لتكون العملية صحيحة.

الفصل الثاني: تمثيل المعطيات Data Representation

التعرف على الطريقة التي تمثل بها البيانات على مختلف أنواعها (من أحرف ورموز وأرقام) وآلية معالجتها، يُعدّ مدخلاً وشرطاً أساسياً لفهم عمل الدارات الرقمية. وسيتم بهذا الفصل التعريف بآلية تمثيل كل من الأعداد المؤشرة والأعداد غير المؤشرة بكل من الأنظمة الأربعة التي أشير لها في الفصل الأول، وعرض للعمليات الحسابية الأربعة الأساسية بكل نظام على حدة.

1-2 أنواع المعطيات وتمثيلها

تصنف المعطيات بشكل عام إلى معطيات عديمة الإشارة Unsigned مثل المعطيات التي تمثل الأحرف والرموز الخاصة والمعطيات التي تقابل الأرمزة Codes وغيرها، وإلى معطيات بإشارة Signed (مقيدة بوجود الإشارة) مثل المعطيات التي تمثل الأرقام التي يتم التعامل معها أثناء إجراء العمليات الحسابية. إشارة المعطيات المؤشرة قد تكون موجبة أو سالبة وفي كلتا الحالتين يجب تمثيلها مع العدد المعطى. وتُمثل المعطيات على مختلف أنواعها في الدارات الرقمية (الحواسيب) باستخدام مكونات النظام الثنائي (القيمتين 0 و 1).
نميز بتمثيل المعطيات مابين تمثيل المعطيات المؤشرة والمعطيات غير المؤشرة، حيث إن المعطيات المؤشرة مرفقة بإشارة ويجب تمثيلها. بالتالي لتمثيل المعطيات المقيدة بإشارة، نشير بداية إلى نقطتين أساسيتين تتعلّقان بتمثيل إشارة العدد، هما:

الأولى- شكل الإشارة: تُمثل الإشارة الموجبة بنظام عدّ أساسه b بالقيمة 0 والإشارة السالبة بالقيمة $b - 1$ ، بالتالي إذا استثنينا النظام العشري الذي يستخدم الرمز " + " للدلالة إلى القيمة الموجبة والرمز " - " للدلالة إلى القيمة السالبة. يشار في النظام الثنائي إلى الإشارة الموجبة بالرقم 0 وإلى الإشارة السالبة بالـ 1. ونفس الشيء بالنسبة للنظامين الثماني والستة عشري، إذ إن الإشارة الموجبة تمثل بالـ 0 والإشارة السالبة في الثماني تمثل بالـ 7 وفي النظام الستة عشري بالـ $F = 15$.

الثانية- مكان توضع الإشارة: تخزن المعطيات في الحاسوب مثلاً، بوحدات خاصة تدعى سجلات. يمكن تصور السجل بشكل مبدئي (والذي سيتم التعرف عليه لاحقاً) على أنه

مجموعة من الحجيرات المتجانبة بطول 1bit لكل حجره. تسمى الحجرة الواحدة خلية. يتميز سجل عن آخر وفقاً لوظيفته وعدد بتّاته، فيقال سجل تخزين طوله أربع بتّ أو ثماني بتّ فمثلاً، يمكن تمثيل (تخزين) الرقم الثنائي $(111011)_2$ بسجل طوله ست خانات بعدد بتّات العدد، وفق ما هو مبين بالشكل أدناه.

1	1	1	0	1	1
MSB					LSB

يحدّد السجل عادة بالمصطلحين LSB و MSB لتمييز بداية السجل عن نهايته. وتشغل إشارة العدد عند تمثيله بسجل بتّ واحد يكون دوماً أقصى بتّ من اليسار (خانة MSB). لتمثيل أعداد مكونة من n خانة يلزم سجل بطول $n + 1$ خانة، بحيث الخانة $n + 1$ تسمى بتّ الإشارة Sign bit. وعند تمثيل الإشارة الموجبة لعدد ما، سيكون بأقصى خانة من اليسار في السجل الذي يضم العدد المراد تمثيله القيمة 0، وعند تمثيل الإشارة السالبة سيكون بأقصى خانة من اليسار في السجل، القيمة 1. وانطلاقاً من أنّ جميع العمليات الحسابية التي تتم داخل الحاسوب تكون على أعداد ممثلة في النظام الثنائي، سيتم فيما يلي الإشارة إلى تمثيل إشارتي الموجب والسالب في النظام الثنائي فقط (علماً أنّ آلية تمثيل الأعداد المؤشرة ببقية الأنظمة تتم من حيث المبدأ في حال حدوثها بأسلوب مماثل لما سيتم في النظام الثنائي). لمعرفة إشارة أي عدد ممثل بسجل، ينظر دوماً إلى القيمة الموجودة ببتّ الإشارة. أما لمعرفة قيمة العدد نفسه إن كان ممثلاً على مستوى البايت مثلاً، فيتم التعامل مع قيمة البتّ MSB على أنه بالوزن (-2^7) .

مثال: القيمة العشرية لبايت مؤشر من الشكل $(1011\ 0000)_2$ ، تحسب كالتالي:

بما أنّ البتّ الأخير $MSB = 1$ إذا العدد سالب ويكتب على النحو التالي:

$$0.2^0 + 0.2^1 + 0.2^2 + 0.2^3 + 1.2^4 + 1.2^5 + 0.2^6 + 1.(-2^7) = \\ 0 + 0 + 0 + 0 + 16 + 32 + 0 + (-128) = (-80)_{10}$$

أما العدد المؤشر بطول 1 byte من الشكل $(0011\ 0000)$ ، أي نفس العدد السابق باستثناء الواحد في أقصى بتّ من اليسار (يصبح موجباً، لوجود الصفر في خانة الـ MSB) تحسب قيمته وفق التالي:

$$0.2^0 + 0.2^1 + 0.2^2 + 0.2^3 + 1.2^4 + 1.2^5 + 0.2^6 + 0.2^7 =$$

$$0 + 0 + 0 + 0 + 16 + 32 + 0 + 0 = (+48)_{10}$$

وهكذا، فإن مجال الأعداد المؤشرة على مستوى البايت سيكون من $(-2^7 = -128)$ حتى

$(+2^7 = +127)$ ممثلة وفق التالي:

$$-128 \sim 1000\ 0000$$

$$-127 \sim 1000\ 0001$$

.....

$$-1 \sim 1111\ 1111$$

$$0 \sim 0000\ 0000$$

$$+1 \sim 0000\ 0001$$

.....

$$+126 \sim 0111\ 1110$$

$$+127 \sim 0111\ 1111$$

أما بالنسبة للأعداد غير المؤشرة، فتمثل على مستوى كامل البايت ببتاته الثمانية لعدم وجد

بت للإشارة (قيمة الواحد من أقصى اليسار تمثل بالنسبة للعدد غير المؤشر بت مفيد)،

وبالتالي مجال الأعداد غير المؤشرة على مستوى البايت هو من 0 حتى 2^8 . فمثلاً، بالنسبة

للعدد (1011 0000) ان كان غير مؤشر تحسب قيمته كالتالي:

$$0.2^0 + 0.2^1 + 0.2^2 + 0.2^3 + 1.2^4 + 1.2^5 + 0.2^6 + 1.2^7 =$$

$$0 + 0 + 0 + 0 + 16 + 32 + 0 + 128 = (176)_{10}$$

حيث إن الوزن 2^7 - المقابل لببت الإشارة لا يستخدم مع القيم غير المؤشرة، ويحسب (2^7) .

وبأسلوب مماثل، لمعرفة قيمة عدد على مستوى كلمة بطول 2 bytes نميز بين الأعداد

المؤشرة والأعداد غير المؤشرة، حيث بالنسبة للأعداد غير المؤشرة بطول 16 bits يكون

مجال القيم من 0 حتى 2^{15} ، أي من التركيبية $0000H = 0000\ 0000\ 0000\ 0000$ التي

تعادل القيمة 0

حتى التركيبية التي تعادل القيمة $(65535)_{10}$. $FFFFH = 1111\ 1111\ 1111\ 1111$

وبالنسبة لقيمة عدد مؤشر على مستوى كلمة بطول 2 bytes، فيتم النظر إلى قيمة البت

MSB، ويتم التعامل معه على أنه بالوزن (-2^{15}) . بالتالي مجال الأعداد المؤشرة على

مستوى الكلمة من أدنى قيمة حتى أعلى قيمة هو من -2^{15} (التي تعادل -32768)

حتى 2^{15} (التي تعادل +32767)، وفق ما هو مبين أدناه:

- 32 768 ~ 1000 0000 0000 0000

- 32 767 ~ 1000 0000 0000 0001

.....
-2 ~ 1111 1111 1111 1110

-1 ~ 1111 1111 1111 1111

0 ~ 0000 0000 0000 0000

+1 ~ 0000 0000 0000 0001

+2 ~ 0000 0000 0000 0010

.....
+32 766 ~ 0111 1111 1111 1110

+32 767 ~ 0111 1111 1111 1111

سيتم التعامل مع الأعداد الثنائية على إنها تتألف من سلسلة String من البتات، مع العلم أن المستخدم هو الذي يحدد نوعية العدد، بمعنى إذا كان العدد مؤشراً فإن البت في أقصى اليسار يمثل الإشارة وبقية البتات تمثل قيمة العدد، أما إذا تم اعتبار العدد غير مؤشر فإن كل البتات المستخدمة تكون لتمثيل قيمة العدد. فمثلاً، يمكن النظر إلى سلسلة البتات $(11001)_2$ كما سنجد لاحقاً، على أنها تمثل القيمة 25 (عدد غير مؤشر) أو القيمة 9- (عدد ثنائي مؤشر، إذا كان ممثلاً بطريقة المقدار والإشارة) أو القيمة 7- (عدد ثنائي مؤشر إذا كان ممثلاً بطريقة المتممات). وهذا يعني أنه يوجد طريقتين لتمثيل العدد السالب، هما طريقة المقدار والإشارة وطريقة المتممات.

2-1-1 طريقة تمثيل الأعداد السالبة باستخدام المقدار والإشارة

استخدمت هذه الطريقة في التجهيزات (الحواسيب) القديمة، ومن أهم سلبياتها أن النتائج ستكون مغلوبة إن لم تؤخذ قيمة الإشارة بعين الاعتبار. تستخدم وفق طريقة المقدار والإشارة أقصى خانة من اليسار في سجل تمثيل العدد لتمثيل قيمة الإشارة، وتستخدم بقية الخانات بالسجل لتمثيل القيمة المطلقة للعدد. فمن أجل سجل بست خانات مثلاً، تخصص خمس خانات لتمثيل العدد والخانة السادسة لتمثيل قيمة الإشارة. يظهر الجدول 1-2 طريقة تمثيل الإشارة (من حيث قيمة توضع الإشارة ومكانها بنوعيتها) للأعداد الممثلة بطريقة المقدار والإشارة بكل من أنظمة العد الأربعة الأساسية.

الجدول 1-2: ترميز إشارتي + و - في كل من أنظمة العدّ الأربعة الأساسية

النظام الثنائي	(+ 6)	(0 00110) ₂
	(- 6)	(1 00110) ₂
النظام الثماني	(+45)	(0 00045) ₈
	(- 45)	(7 00045) ₈
النظام العشري	(+85)	(0 00085) ₁₀
	(- 85)	(9 00085) ₁₀
النظام الستة عشري	(+3E)	(0 0003E) ₁₆
	(- 3E)	(F 0003E) ₁₆

لتوضيح هذه الطريقة، دعنا نستخدم سجلات بطول قياسي، كأن نعتمد سجلاً بطول 8bits يحاكي طول كلمة معالج صغري مثل المعالج 8080 وفق الترميز التالي:

7	6	5	4	3	2	1	0
S							
Sign	Information bits						

يتم عند استخدام هذه الطريقة في تمثيل الأعداد معالجة كل من القسم المطلق للعدد والإشارة S على حدة. وبالتالي قد ينتج عن عملية حسابية ظهور رقم 1 في خانة الإشارة، ما يدل على أنّ النتيجة سالبة حتى لو كانت النتيجة الفعلية موجبة. ولتبيان ذلك ننطلق من أن عملية جمع عددين ثنائيين كما سنجد لاحقاً تخضع إلى القاعدة: $1 + 1 = 10$ (أي الجواب يكون 2، ولكن 2 ليست من مكونات النظام الثنائي، ويكتب الناتج بالشكل 10 بمعنى أن الناتج 0 والنقل 1). تتم عملية الجمع الفعلية وفق النظام العشري، ولكن النتيجة تكتب بالثنائي. مثال: لجمع عددين موجبين طول كل منهما 8 bits (بما في ذلك بت الإشارة) من الشكل 0110 1100 للعدد الأول و 0100 1000 للعدد الثاني، حيث $S = 0$ بكل من العددين مما يدل على أن العددين موجبين، بالتالي يفترض أن يكون الناتج عدداً موجباً. نجد بجمع العددين وفق طريقة المقدار والإشارة بأن الناتج سيكون:

$$\begin{array}{r}
 01101100 \\
 + 01001000 \\
 \hline
 10110100
 \end{array}$$

وبما إن البت الثامن للناتج يحتوي على القيمة 1، فهذا يعني إن قيمة الناتج سالبة، أي إن العملية خاطئة. ظهرت هذه المشكلة بسبب استخدام سجلات بطول كلمة لا يكفي لاستيعاب

الناتج، أي يحدث فيض في طول النتيجة (Over flow)، بالتالي يكون الحل في هذا المثال باستخدام سجلات بطول كلمة أكبر. فمثلاً، باستخدام سجلات بطول 9 bits لجمع العددين، تصبح عملية الجمع كالتالي:

$$\begin{array}{r}
 \text{Sign} \quad \text{Information bits} \\
 0 \quad 01101100 \\
 + 0 \quad 01001000 \\
 \hline
 = 0 \quad 10110100
 \end{array}$$

وبما أن البت التاسع هو بت الإشارة، ويحتوي القيمة صفر، تصبح النتيجة موجبة وصحيحة بعدما تم التخلص من الفيض بحجز مكان إضافي للنقل.

2-1-2 طريقة تمثيل الأعداد السالبة باستخدام المتممات

تستخدم الطريقة السابقة "طريقة المقدار والإشارة" بتمثيل الأعداد المقيدة بإشارة في الرياضيات العادية، حيث يتم التعامل مع أي عدد على أنه مقدار ورمز (+ ، -). أما لتنفيذ العمليات الحسابية في الدارات الرقمية، فيفضل استخدام نظام آخر في تمثيل الأعداد المؤشرة، هو نظام المتممات Complements. تستخدم طريقة المتمم (المكمل) في النظام الثنائي لإجراء عملية الطرح كما هو الحال في النظام العشري. فمثلاً، ناتج عملية الطرح بالنسبة للعددين العشرينين: 132 - 425 وفق الطريقة المباشرة هو 293، ويمكن الحصول على نفس النتيجة باستخدام طريقة المتمم بشكل آخر وفق التالي: باعتماد المتمم العشري للمطروح (أي للعدد 132) وهو العدد 868، يمكن تحويل عملية الطرح إلى عملية جمع بإضافة المتمم العشري للمطروح إلى المطروح منه وفق الشكل $425 + 868 = 1293$ ، وبحذف الفائض (أي الواحد من أقصى اليسار، لأنه لا يعقل أن يكون ناتج طرح عددين هو عدد أكبر من الكبير من بينهما). ويتم الحصول على نفس النتيجة السابقة وهي 293. يمكن إجراء عملية الطرح السابقة باستخدام المتمم التساعي (أي المتمم بالنسبة للأساس ناقص واحد). وفي حال استخدام هذه الطريقة، يجب أن يضاف للناتج النهائي قيمة 1 حتى تصبح النتيجة صحيحة. ويقال إن العملية قد تمت باستخدام متمم الأساس المخفض Diminished Radix Complement، أي المتمم لـ (b - 1). المبدأ المشار إليه أعلاه

في عملية الطرح في النظام العشري هو نفس المبدأ المستخدم في الدارات الرقمية لطرح الأعداد في النظام الثنائي، حيث إن إشارة الناقص في النظام الثنائي، تعني المتمم للاثنتين. وبما أن $b = 2$ في النظام الثنائي، فإن التعامل سيكون مع مفهوم المتمم للأساس، الذي يسمى بالمتمم الثنائي (Second Complement أو Two's Complement) المتمم للاثنتين) أو مع المتمم لواحد، أي $b - 1 = 2 - 1 = 1$ (First Complement) الذي يقابل المتمم التساعي في النظام العشري. يجب عند استخدام المتمم للواحد في عملية الطرح في النظام الثنائي أن يضاف للنتائج الأولي واحداً للحصول على الناتج النهائي. أو أن يضاف واحد إلى المتمم للواحد وبعد ذلك يضاف الناتج إلى المطروح، كما هو مبين بالمثال التالي: مثال: أوجد ناتج طرح العددين العشريين: $19 - 29$ في النظام الثنائي باستخدام المتممات. الجواب: بما أن العدد الأول (المطروح منه) $(29)_{10} = (11101)_2$ ، والعدد الثاني $(19)_{10} = (10011)_2$ ، يفترض أن يكون ناتج عملية الطرح $(10)_{10}$ ، أو ما يقابل ثنائياً القيمة $(1010)_2$. يمكن باستخدام طريقة المتممات الحصول على نفس الناتج، وفق ما سنبينه أدناه بالنسبة لكل نوع من المتممين.

أولاً- باستخدام طريقة المتمم للواحد First Complement:

أول ما يجب فعله في أثناء عملية الطرح باستخدام طريقة المتمم للواحد، هو الحصول على المتمم للواحد للعدد المطروح. ويتم الحصول على المتمم للواحد لأي عدد في النظام الثنائي باستبدال الواحد بصفر والنصف بواحد. وبالتالي المتمم للواحد للمطروح (10011) بالمثال أعلاه هو (01100) . وبإضافة قيمة هذا المتمم إلى قيمة المطروح منه، أي: $(10011) + (01100) = (101001)$ ، يتم بعد حذف النقل الأخير أن نجد الحصول على ناتج أولي من الشكل (01001) . يتم الحصول على الناتج النهائي بإضافة واحد إلى الناتج الأولي $(1010) + 1 = 01001$ وهو المطلوب.

ثانياً - باستخدام طريق المتمم للاثنتين Two's Complement

ينتج المتمم للاثنتين لأي عدد في النظام الثنائي من إضافة واحد للمتمم للواحد لنفس العدد، أو يمكن الحصول عليه بالطريقة المباشرة وفق التالي: بتتبع العدد المراد إيجاد متممه للاثنتين من اليمين باتجاه اليسار، تكتب مكونات العدد إلى حين الوصول إلى أول واحد (وقد يكون

هذا الواحد بأول خانة)، يكتب وبعد ذلك تبدل قيم بقية الخانات في العدد، بحيث 1 يصبح 0 وبالعكس. يوجد بالمثال أعلاه بالخانة الأولى من اليمين بالمطروح المعطى (10011) قيمة واحد، يكتب كما هو، وبعد ذلك يتم عكس قيمة بقية الخانات ليتم الحصول على القيمة (01101) باعتبارها متمماً للاثنتين وبإضافة هذه القيمة لقيمة المطروح منه بعد حذف الواحد الذي يزيد عن خانات العدد $(1\ 01010) = (01101) + (11101)$ ، يتم الحصول على الجواب النهائي $(1010)_2$. بالطبع العدد الموجب لا يتغير شكل تمثيله أي كانت الطريقة المستخدمة في التمثيل، حيث القيمة 0 تكون دوماً في أقصى اليسار للتعبير عن إشارة الزائد للعدد أما بالنسبة للعدد السالب، كيفما كانت طريقة تمثيله، فيكون في أقصى خانة من اليسار قيمة 1 للتعبير عن إشارة الناقص، لكن شكل العدد يختلف بحسب الطريقة التي يمثل بها (باستخدام المقدار والإشارة أو باستخدام المتممات). فمثلاً، لتمثيل عدد عشري، مثل (25-) بثمانى خانات باستخدام طريقة المتممات، يتم الانطلاق من القيمة المطلقة للعدد، أي (25). تكتب التركيبية الثنائية المقابلة في النظام الثنائي بخمس خانات وفق الشكل (11001). وبالنظر إلى أن لكل بت من البتات الثمانية المقترح استخدامها (في السجل) قيمة محددة، تتم إضافة أصفار على يسار العدد كي لا تتغير قيمته ويصبح من الشكل 11001 000. وللتعبير عن أن العدد سالب، يؤخذ المتمم الثنائي للشكل الجديد للحصول على القيمة 11100111. وعند استخدام طريقة المتممات بتنفيذ العمليات الحسابية، فإن خانة الإشارة تشارك في العملية.

مثال: أوجد المتمم للاثنتين للعدد الثنائي $(10011000)_2$.

الجواب: باستخدام طريقة المتمم الأحادي: $01101000 + 1 = 01100111$

أما وفق الطريقة المباشرة، فيكون المتمم للاثنتين هو (01101000) ، حيث إن أول 1 نصادفه هو في الخانة الرابعة من اليمين، نكتبه وبعد ذلك نباشر بقلب القيم (من 1 إلى 0 وبالعكس). والجواب هو نفسه بالحالتين.

مثال: أوجد ناتج العملية الحسابية على أعداد مؤشرة من الشكل 37-14؟

الجواب: لتنفيذ العملية المطلوبة يلزم بالحد الأدنى سجلات بطول 7 Bits لكل عدد حتى تكون النتيجة صحيحة، وذلك لأن 37 تحتاج بدون إشارة إلى 6 Bits وبالتالي تحتاج لبيت

سابع للإشارة وتصبح من الشكل 1011011 بصيغة المتمم الثنائي. أما العدد 14 فيحتاج مع الإشارة إلى خمس بتات، وبما أنه موجب ليس له متمم ويجب أن يساوي بالطول العدد الأول وهكذا يصبح من الشكل 0001110 وجمع العددين يصبح الناتج من الشكل 1101001 أو ما يعادل القيمة 23- ممثلة بصيغة المتمم الثنائي لأنها سالبة ولمعرفة القيمة المطلقة نأخذ المتمم الثنائي مرة أخرى فينتج القيمة 23 مسبقة بإشارة الناقص. مثال: باستخدام طريقة المتممات، أوجد ناتج طرح العددين (35 - 30) بعد تمثيل كل منهما بثمان خانات (8 bits).

الجواب: يمثل العدد الأول 30 في سجل بطول 8 bits بالقيمة المطلقة (أي من دون إشارة) وفق الشكل $(0001\ 1110)_2$ ، وباستخدام المتمم للاثنتين يصبح الناتج من الشكل $(1110\ 0010)_2$ ، وهو ما يقابل العدد (-30). ونفس الشيء بالنسبة للعدد (-35) سيصبح من الشكل $(1101\ 1101)_2$ ، وعند جمع القيمتين معاً $(-35) + (-30)$ يكون الجواب: $(1011\ 1111)_2$ ، وهو ما يعادل القيمة العشرية (65-) لكن بصيغة المتمم الثنائي لأنه سالب. للتأكد من صحة الناتج، يؤخذ المتمم للاثنتين للقيمة $(1011\ 1111)_2$ مرة ثانية، فينتج $(0100\ 0001)_2$ أي 65 كقيمة مطلقة مسبقة بإشارة الناقص. بالطبع ما ينطبق على الأعداد الصحيحة المؤشرة ينطبق أيضاً على الأعداد الكسرية المؤشرة من حيث إيجاد المتمم.

2-2 تمثيل الأعداد بالفاصلة العشرية

تخضع الطريقة التي يتم استخدامها في تمثيل الأعداد بالدارات الرقمية إلى عدة عوامل، يأتي في مقدمتها الدقة المطلوبة بالتمثيل ومجال القيم المطلوب تمثيلها ونوع العدد وكلفة العتاد اللازم لتنفيذ التمثيل. وبأخذ هذه العوامل بعين الاعتبار، يجب بداية التمييز بين كل من العدد الصحيح والعدد الكسري. يقصد بالعدد الكسري كل عدد يحتوي على الفاصلة. والفاصلة بالتعريف هي رمز خاص يستعمل لفصل القسم الصحيح عن الكسري. ويمكن التمييز بحسب

طريقة تمثيل الفاصلة بين نوعين، هما: - الفاصلة الثابتة Fixed Point

- الفاصلة العائمة Floating Point

تستخدم الفاصلة الثابتة لتمثيل الأعداد بمجال محدود، وبالتالي فهي بسيطة والعتاد اللازم

لتنفيذها غير معقد. أما الفاصلة العائمة (المتحركة)، فتستخدم لتمثيل الأعداد بمجال أكبر، ويكون العتاد اللازم لتنفيذها أكثر تعقيداً. وبالتالي لإجراء العمليات الحسابية على الأعداد بالفاصلة العائمة يجب اعتماد دارات خاصة. وفي حال عدم توافر هذه الدارات يمكن تنفيذ العمليات في برامج خاصة مكتوبة بلغة الاسبلي.

2-2-1 تمثيل الأعداد بالفاصلة الثابتة

يتم تثبيت الفاصلة الثابتة في العدد بعد التأكد من أن مكانها سيبقى ثابتاً ، ولن يتغير في حدود المعالجة التي ستتم على العدد. ويتم استعمال هذا التمثيل للأعداد عندما تكون الأعداد التي ستعالج متساوية من حيث الحجم وتقع جميعها ضمن مجال محدد، ولا يوجد فوارق كبيرة بينها (مثل القروش في الليرات السورية، التي تأخذ خاننتين). والاصطلاح الأكثر استعمالاً لمكان الفاصلة هو أن يتم تثبيتها في الموقع الأخير من العدد لجهة اليسار أو اليمين (وليس تركها في المكان الذي ترد فيه ضمن العدد). عندما يتم تثبيت الفاصلة في أقصى يمين الأرقام التي تولف العدد يتم الحصول على أعداد صحيحة، كما في الشكل 11001.0 وعندما يتم تثبيت الفاصلة في أقصى اليسار، يتم الحصول على الأعداد الكسرية، كما في الشكل 0.11001. في كلتا الحالتين، تنتج مشاكل عن تمثيل الفاصلة، من أهمها زيادة حجم النتائج بعد تنفيذ العمليات الحسابية، أي قد يحدث فيض في النتائج. قد يؤدي هذا الفيض إلى إلغاء بت الإشارة

$$\begin{array}{r} 0.10110 \\ + 0.01101 \\ \hline 1.00011 \end{array}$$

تُظهر عملية الجمع لعددتين موجبين بهذا المثال نتيجة سالبة لوجود واحد في موقع الإشارة وهذا خطأ حتماً. وفي حال اعتماد الفاصلة في أقصى يمين العدد (حالة الأعداد الصحيحة) قد لا تستطيع السجلات استيعاب النتيجة، وهذا يؤدي بدوره إلى وقوع خطأ في النتيجة. أما في الحالة التي يتم فيها تثبيت الفاصلة في أقصى اليسار (حالة الأعداد الكسرية) فإن الخطأ الذي قد ينتج، سيكون أصغر بكثير من الخطأ الناتج عن الحالة الأولى. وبالتالي يفضل دوماً اعتماد مبدأ تثبيت الفاصلة في أقصى اليسار.

2-2-2 تمثيل الأعداد بالفاصلة العائمة

مجال الأعداد التي يمكن تمثيلها باستخدام الفاصلة الثابتة محدّد ويعتمد على عدد الوحدات الثنائية المستخدمة لذلك. وبالتالي للحصول على مجال أوسع في التمثيل لا بد من استخدام طريقة الفاصلة العائمة التي تتناسب الحالات التي تكون فيها أطول الأعداد سريعة التغيير. بنتيجة إجراء العمليات الحسابية عليها وبشكل يصعب معه مراقبتها ومعرفة حدود التغيير فيها. كما أنّ حجم الأعداد كبير ويفوق إمكانية تمثيلها في كلمة الآلة. فمثلاً، العدد العشري

الصحيح $N = 237000$ يمكن تمثيله بواحد من الأشكال التالية:

$$N = 237.10^{-6}$$

$$N = 0.00237.10^{-8}$$

$$N = 237000000.10^{-2}$$

يمكن تحريك الفاصلة العائمة إلى اليسار وإلى اليمين. وبالتالي طريقة تمثيل الأعداد بالفاصلة العائمة مناسبة لعدم ضياع في، وتستخدم لتفادي المشاكل الناتجة عن حجم الأعداد وطولها. ومن أشكال التمثيل أعلاه، يُعدّ الشكل الأول $N = 237.10^{-6}$ أفضل شكل لتمثيل العدد 237000 ويسمى بالكسر المعياري Normalized Mantissa، لأن الجزء الصحيح معدوم، ولا يوجد أصفار على يمين الفاصلة. تعرف الأعداد الكسرية بالفاصلة العائمة Floating-Point Numbers بالأعداد الحقيقية Real Number والشكل العام لتمثيل الأعداد باستخدام الفاصلة العائمة هو $N = M.B \pm E$ ، حيث:

N - يمثل العدد المفروض تمثيله.

M - الجزء الكسري ويدعى Mantissa

B - أساس النظام وهو عشرة في النظام العشري و 2 بالثنائي.

E - أس عشري (Power) وهو عدد صحيح سلبى أو إيجابى.

فمثلاً من أجل العدد العشري 351407900 تكون المن تيسا (التي قيمتها دوماً تتراوح ما بين 0 و 1) هي 0.3514079 والأس 9 ويكتب العدد بالشكل: 0.3514079×10^9 ، أما بالنسبة للأعداد الثنائية بالفاصلة العائمة، فيصنف شكلها (صغيتها) بحسب المعيار ANSI/IEEE إلى ثلاثة أشكال هي:

- أعداد بدقة بسيطة Single-precision

- أعداد بدقة مضاعفة Double-precision

- أعداد بدقة موسعة Extended-precision

وهي جميعها متماثلة من حيث الشكل وتختلف عن بعضها بعدد بتات التمثيل. حيث إن الأعداد الثنائية بالفاصلة العائمة والدقة البسيطة تمثل بـ 32 bits والأعداد بالدقة المضاعفة تمثل بـ 64 bits. أما الموسعة فتستخدم 80 bits. نكتفي هنا بعرض الشكل العام للأعداد الثنائية بالفاصلة العائمة بالدقة البسيطة، حيث يخصص بيت واحد في أقصى اليسار للإشارة S يليه 8 bits لتمثيل قيمة الأس E و 23 bit المتبقية لتمثيل قيمة المن تيسا (M or F) وفق ما هو مبين بالشكل أدناه.

1 bit لإشارة الأس Sign of power (S)	8 bit لقيمة الأس Value of power Exponent (E)	23 bit لقيمة المن تيسا Value of mantissa (M) (or Fraction (F))
--	--	--

تمثل النقطة (الفاصلة) الثنائية بالجزء الكسري للعدد (المن تيسا)، وتفهم على أنها واقعة على يسار الـ 23 bit بمعنى أن للمن تيسا 24 bits وذلك لأنه في أي عدد ثنائي يوجد بأقصى اليسار قيمة واحد. ويفهم أن هذا الواحد موجود بالرغم من أنه لا يشغل موضع بالتوضع الحالي. أما الـ 8 bit المخصصة للأس، فتشمل إشارة الأس أيضاً، وبالتالي مجال الأس يمكن أن يتراوح من -127 حتى +128. ولنوضح ذلك من خلال مثال محدد. لتمثيل عدد ثنائي من الشكل 1 0110 1001 0001 بصيغة الفاصلة العائمة، نقوم بداية بتمثيل القيمة كواحد زائد العدد الثنائي الكسري F بإزاحة النقطة (الفاصلة) الثنائية 12 موضع لليسار وضرباً بالقوة المناسبة للأساس 2. وتصبح القيمة من الشكل: $1.011010010001 \times 2^{12}$ ، وبفرض أن القيمة موجبة، تكون الإشارة $S = 0$ والأس 12 يمثل أس منحاز بجمعه للقيمة 127 ($12 + 127 = 139$). الأس الجديد (المنحاز) E يمكن أن يعبر عنه بالصيغة $(10001011)_2 = (139)_{10}$. المن تيسا تمثل الجزء الكسري F للعدد 011010010001. لأنه يوجد هناك دائماً 1 على

يسار الفاصلة الثنائية، ويصبح العدد بالفاصلة العائمة بشكله النهائي وفق التالي:

S	E	F
0	10001011	011010010001000000000000

مثال آخر: حول العدد العشري 3.248×10^4 إلى عدد ثنائي بفاصلة عائمة بدقة بسيطة.

الجواب: نقوم بداية بتحويل العدد العشري إلى الشكل الثنائي وفق التالي:

$$3.248 \times 10^4 = 32480 = (11111011100000)_2 = (1.1111011100000)_2 \times 2^{14}$$

البت MSB في أقصى اليسار هو دوماً واحد (أي الرقم قبل الفاصلة) لا يحجز له موضع

ولذلك الماتيسا هي عدد ثنائي كسري بطول 23 bit من الشكل

111110111000000000000000 والأس المنحاز يكون:

$$14 + 127 = 141 = (10001101)_2$$

يصبح الشكل النهائي للعدد المعطى كالتالي:

0	10001101	111110111000000000000000
---	----------	--------------------------

لننظر الآن إلى كيفية تقييم العدد الثنائي الممثل بصيغة الفاصلة العائمة. الاتجاه العام

لتحديد قيمة العدد الثنائي يعبر عنه بالعلاقة: $\text{Number} = (-1)^S (1 + F) (2^{E - 127})$

مثال: من أجل عدد ثنائي معطى بصيغة الفاصلة العائمة مثل:

S	E	F
1	10010001	100011100010000000000000

بيت الإشارة $S = 1$ والأس المنحاز $E = 10010001 = 145$ وب تطبيق الصيغة أعلاه،

نحصل على العدد: $\text{Number} = (-1)^1 (1 + .10001110001) (2^{145 - 127})$

$$= (-1)(1.10001110001) (2^{18}) = -110\ 001110001000\ 0000$$

الذي يحول بدوره إلى المكافئ العشري.

وأخيراً نشير إلى أنه في بعض المراجع يعبر عن الأعداد بدقة بسيطة، بصيغة العدد الحقيقي

القصير (أو الصغير) Short-real Format وعن الأعداد بدقة مضاعفة، بصيغة العدد

الحقيقي الطويل Double-precision ، وعن الأعداد بدقة موسعة بصيغة الأعداد الحقيقية

المؤقتة Temporary-real. تختلف الصيغ المتبقية لتمثيل الأعداد بالفاصلة العائمة عن

صيغة الأعداد بالدقة البسيطة فقط بعدد البتات المخصص لكل حقل من الحقول الثلاثة

الأساسية لتمثيل العدد، حيث بالنسبة للأعداد بالدقة المضاعفة التي تستخدم 64 bits لتمثيل العدد يخصص بيت لتمثيل الإشارة و 11 bit لتمثيل قيمة الأس وال 52 bit المتبقية لتمثيل مقدار العدد. أما بخصوص الأعداد بدقة موسعة والتي تستخدم 80 bits يخصص بيت للإشارة و 15 bit لتمثيل قيمة الأس المزاح وال 64 bit المتبقية لتمثيل مقدار الأساس. وكملاحظ بخصوص إيجاد متممات الأعداد بفاصلة، فإن عملية إيجاد المتمم للواحد أو للاثنتين، تتم بغض النظر عن وجود الفاصلة، بمعنى أنه يتم إيجاد المتمم للعدد كما هو من دون فاصلة، وبعد إيجاده تعاد الفاصلة إلى المكان النسبي في العدد المراد إيجاد متممه.

ملاحظة: حول تمثيل الفاصلة العائمة للمطالعة. للتحويل من العشري إلى الفاصلة العائمة، يوجد اجرائية خاصة يجب اتباعها لتحويل أي رقم عشري إلى الفاصلة العائمة تتمثل بالتالي:

- A- Convert the absolute value of the number to binary, This can be done by converting the integral and fractional parts separately.
- B- Append $\times 20$ to the end of the binary number (which does not change its value).
- C- Normalize the number. Move the binary point so that it is one bit from the left. Adjust the exponent of two so that the value does not change.
- D- Place the mantissa into the mantissa field of the number. Omit the leading one, and fill with zeros on the right.
- E- Add the bias to the exponent of two, and place it in the exponent field. The bias is $2^{k-1} - 1$, where k is the number of bits in the exponent field. For the eight-bit format, $k = 3$, so the bias is $2^{3-1}-1 = 3$. For IEEE 32-bit, $k = 8$, so the bias is $2^{8-1}-1 = 127$
- F- Set the sign bit, 1 for negative, 0 for positive, according to the sign of the original number.

مثال: حول العدد العشري 2,625 إلى صيغة الفاصلة العائمة بـ 8 bits، فيكون الجواب:
الخطوة a: وتعني تحويل الجزء الصحيح $2_{10} = 10_2$ والكسري $0.625_{10} = 0.101_2$ أي $2.625_{10} = 10.101_2$

الخطوة b: Add an exponent part: $10.101_2 = 10.101_2 \times 2^0$

الخطوة c: Normalize: $10.101_2 \times 2^0 = 1.0101_2 \times 2^1$

الخطوة d: Mantissa: 0101

الخطوة e: Exponent: $1 + 3 = 4 = 100_2$

الخطوة f: Sign bit is 0.

The result is 0 100 0101. Represented as hex, that is 45_{16} .

مثال آخر: حول العدد العشري -4,75 إلى صيغة الفاصلة العائمة بـ 8 bits، فيكون الجواب بدون مزيد من التفاصيل وفق التالي:

الخطوة a: $4.75_{10} = 100.11_2$

الخطوة b: Normalize: $100.11_2 = 1.0011_2 \times 2^2$

الخطوة c: Mantissa is 0011, exponent is $2+3 = 5 = 101_2$, sign bit is 1.
So -4.75 is $11010011 = D3_{16}$

مثال: Convert -1313.3125 to IEEE 32-bit floating point format

a. The integral part is $1313_{10} = 10100100001_2$. The fractional:

$0.3125 \times 2 = 0.625$ 0 Generate 0 and continue.

$0.625 \times 2 = 1.25$ 1 Generate 1 and continue with the rest.

$0.25 \times 2 = 0.5$ 0 Generate 0 and continue.

$0.5 \times 2 = 1.0$ 1 Generate 1 and nothing remains.

b. So $1313.3125_{10} = 10100100001.0101_2$.

c. Normalize: $10100100001.0101_2 = 1.01001000010101_2 \times 2^{10}$.

d. Mantissa is 01001000010101000000000, exponent is $10 + 127 = 137 = 10001001_2$, sign bit is 1.

So -1313.3125 is $11000100101001000010101000000000 = C4A42A00_{16}$

3-2 العمليات الحسابية على أنظمة العد

تخضع العمليات الحسابية الأساسية (الجمع والطرح والضرب والقسمة) في كل من النظام الثنائي والثماني والستة عشر إلى نفس القواعد التي تخضع لها نفس العمليات الحسابية في النظام العشري. وسيتم فيما يلي توصيف كل من هذه العمليات بكل نظام على حدة.

1-3-2 العمليات الحسابية في النظام الثنائي

سيتم هنا إظهار كل من العمليات الأربعة الأساسية على أعداد ثنائية حسابياً، وسيتم لاحقاً إظهار هذه العمليات باستخدام الدارات المناسبة تقنياً.

أولاً- الجمع في النظام الثنائي

سبق وأشرنا إلى عملية الجمع الحسابي للأعداد الثنائية Binary addition، وفيما يلي سيتم عرض بعض الأمثلة.

مثال : أوجد ناتج جمع العددين الثنائيين: $(11001)_2$ و $(10111)_2$.

الجواب: $(110000)_2 = (11001)_2 + (10111)_2$.

يوجد أكثر من طريقة للتأكد من صحة النتيجة، منها تحويل الأعداد المعطاة إلى النظام العشري. ففي المثال أعلاه، يلاحظ أن: $(11001)_2 = (25)_{10}$ ، $(10111)_2 = (23)_{10}$ ، $(23)_{10} + (25)_{10} = (48)_{10} = (110000)_2$.

مثال: أوجد ناتج جمع العددين العشرين $8 + 8 = 16$ بالصيغة الثنائية

الجواب: $10000 = 1000 + 1000$ وهو المطلوب

مثال: أوجد ناتج جمع العددين: $(101.0111)_2$ و $(10111.101)_2$.

الجواب: بداية إن لم تكن الأعداد المعطاة متساوية بالطول، يضاف للعدد الأقصر من اليسار أصفار حتى يصبح كلا العددين بنفس الطول قبل المباشرة بعملية الجمع. وبالنسبة

للأعداد الكسرية تضاف الأصفار على الجزء الكسري من اليمين،

$(10111.101)_2 + (101.0111)_2 = (11101.0001)_2$

مثال: أوجد ناتج جمع العددين: $(1110111)_2$ و $(10111.101)_2$.

الجواب: $(10001110.101)_2$.

ثانياً- الطرح في النظام الثنائي

تخضع عملية الطرح الثنائي Binary subtraction لقواعد الطرح محددة هي: $0 - 0 = 0$ ، $0 - 1 = 1$ ، $1 - 0 = 1$ ، $1 - 1 = 0$ وبالنسبة للتركيبية الثانية، أي $(0 - 1)$ فإن عملية الطرح غير ممكنة، بالتالي لإجرائها يفترض اقتراض واحد، وتصبح العملية $10 - 1$ (أي $1 - 2$ عشرياً)، ويفهم أن الناتج هو 1 والاستعارة (Borrow) 1.

مثال : أوجد ناتج طرح العدد الثنائي $(1000110)_2$ من العدد $(1110011)_2$.

$$(1110011)_2 - (1000110)_2 = (0101101)_2 \text{ يكون}$$

يلاحظ بهذا المثال، أن عملية الاستعارة تتم تماماً كما هو الحال في النظام العشري، وأن عملية الطرح تبدأ من خانة الآحاد. وللتأكد من صحة النتيجة، تعاد الأعداد المعطاة إلى ما يقابلها في النظام العشري، أي: $(101101)_2 = (45)_{10} = (70)_{10} - (115)_{10}$ تتم عملية الطرح في الحاسوب بطريقة مختلفة، حيث تستخدم طريقة المتممات التي تسمح بتحويل عملية الطرح إلى جمع، ما يسهل العمليات الحسابية في الدارات الكهربية ويجعلها قابلة للتنفيذ. ولطرح عدد ما ممثل بعدد خانات أقل من خانات العدد المطروح منه يتم في أثناء إيجاد قيمة المتمم للمطروح، تكملة خاناته من اليسار بأصفار إلى أن يصبح طوله مساوياً لطول المطروح منه قبل تشكيل المتمم.

مثال: أوجد ناتج طرح العددين التاليين: $(1011)_2 - (1100\ 0010)_2$.

الجواب: يضاف للمطروح (1011) أصفار من اليسار إلى أن يصبح طوله ثماني خانات بطول المطروح منه، وبعد ذلك يصبح المتمم للاثني للمطروح من الشكل $(1111\ 0101)_2$.

وبالجمع مع المطروح منه يكون $(1011\ 0111)_2 = (1111\ 0101)_2 + (1100\ 0010)_2$.

مثال: أوجد حاصل طرح العددين الثنائيين $(1011.011)_2 - (11001.1101)_2$.

الجواب: في حال وجود الفاصلة تتم المحافظة على موضعها وزيادة أصفار في حال عدم تساوي أطوال الأعداد وفق ما هو مبين بهذا المثال.

$$(110011101)_2 + (010110110)_2 = (011100111)_2 \sim (1110.0111)_2$$

ثالثاً- الضرب في النظام الثنائي

يمكن عند الضرورة تنفيذ مختلف العمليات الحسابية باستخدام الدارات الرقمية للوصول إلى سرعة عالية بالتنفيذ على حساب درجة تعقيد الدارات وكلفتها، لكن عمليتي القسمة والضرب تحديداً في حال عدم الحاجة للسرعات العالية بالتنفيذ، يفضل إنجازهما برمجياً، نظراً لدرجة تعقيد الدارات المطلوبة لتنفيذ كل منهما. تخضع بالاطار العام عملية الضرب في النظام الثنائي Binary multiplication لنفس الأسس المتبعة في النظام العشري، أي:

$$1 \cdot 1 = 1, 1 \cdot 0 = 0, 0 \cdot 1 = 0, 0 \cdot 0 = 0$$

وهنا تجدر الإشارة إلى أنه بالنسبة لعملية الضرب قد يضاعف طول الناتج.

مثال: أوجد ناتج ضرب العددين التاليين: $(1011)_2 \cdot (1110)_2$ ؟

الجواب: القيمة $(1110)_2$ تعادل عشرياً العدد $(14)_{10}$ والقيمة $(1011)_2$ تعادل عشرياً العدد $(11)_{10}$. وتكون عملية الضرب الثنائية وفق شكلها التقليدي كالتالي:

$$\begin{array}{r} 1110 \\ \times 1011 \\ \hline 1110 \\ 11100 \\ 0000 \\ 1110 \\ \hline 10011010 \end{array}$$

أي ما يعادل $(154)_{10}$ ، وبالتالي نتيجة عملية الضرب لعددين طول كل منهما 4 bits هي بطول 8 bits. وبأسلوب مماثل يتم ضرب الأعداد الثنائية بوجود الفاصلة.

رابعاً- القسمة في النظام الثنائي

تمثل عملية قسمة عدد ثنائي على آخر Binary division سلسلة من عمليات الجمع المتكررة مابين المقسوم والمتمم للثلاثين للمقسوم عليه (بعد حذف أي نقل يحدث أثناء عملية الجمع) إلى حين وصول ناتج الجمع المتكرر إلى قيمة الصفر. وبعد ذلك تكون نتيجة عملية القسمة مساوية لعدد مرات الجمع.

مثال: أوجد ناتج قسمة العدد الثنائي $(1111000)_2$ على العدد $(11110)_2$.
 الجواب: بداية نساوي أطوال العددين (المقسوم والمقسوم عليه)، أي أن العدد $(11110)_2$ يصبح من الشكل $(0011110)_2$ بطول المقسوم. بعد ذلك نوجد المتمم الثنائي للمقسوم عليه فيكون من الشكل $(1100010)_2$ ومن ثم نقوم بجمع هذا الناتج مع المقسوم عدد من المرات إلى أن يصبح الناتج صفراً وذلك بعد حذف النقل في كل عملية جمع وفق التالي:

$$\begin{array}{r}
 (1111000)_2 \\
 + \quad 1100010 \\
 \hline
 1 \quad 1011010 \\
 + \quad 1100010 \\
 \hline
 1 \quad 0111100 \\
 + \quad 1100010 \\
 \hline
 1 \quad 0011110 \\
 + \quad 1100010 \\
 \hline
 1 \quad 0000000
 \end{array}$$

يلاحظ أن عدد مرات الجمع المنفذة حتى تم التوصل إلى النتيجة صفر هو أربعة، لذلك النتيجة هي: $(100)_2 = (4)_{10}$.

2-3-2 العمليات الحسابية في النظام الثماني

بالنظر إلى أن العمليات الحسابية بالدارات الرقمية تتم على قيم بالنظام الثنائي وأن النظام الثماني لا يستخدم بشكل مباشر في عمليات المعالجة ضمن الدارات الرقمية، ستتم الإشارة فيما يلي إلى عمليتي الجمع والطرح فقط، خاصة أن هاتين العمليتين تمثلان الأساس بالنسبة لبقية العمليات الأخرى وهما مماثلتان للعمليات في النظام العشري بعد الأخذ بعين الاعتبار قيمة الأساس ومكونات النظام.

أولاً- الجمع في النظام الثماني

تتم عملية الجمع في النظام الثماني عشرياً وتمثل ثمانية. وبالتالي أثناء جمع عددين بالنظام الثماني، يتم الانتقال إلى الخانة التالية عندما يتجاوز المجموع الرقم (7) بمنقول يساوي 1، بحيث إن وزن هذا المنقول هو (8). نميز بعملية الجمع بالثماني بين حالتين:

الأولى - إذا كان مجموع أي رقمين متقابلين من عددين أقل من الرقم (8)، يكتب ناتج الجمع كما هو من دون تغيير .

مثال: أوجد ناتج جمع العددين الثمانيين التاليين: $(21)_8 + (43)_8$.

الجواب: الناتج هو $(64)_8$ ، لأن $3 + 1$ بالخانة الأولى من كلا العددين هو (4) نكتب كما هي، لأنها أصغر من أساس النظام (8). والأمر نفسه يتكرر بالنسبة للعددين (2 و 4) بالخانة الثانية، حيث إن مجموعهما 6 أصغر من الأساس ويكتب كما هو .

الثانية - إذا كان مجموع أي رقمين متقابلين من عددين أكبر أو يساوي أساس النظام، يحول المجموع إلى النظام الثماني ويتم ذلك وفق الطريقة المباشرة بكتابة الفرق بالخانة المقابلة لكل من الرقمين مابين المجموع والأساس، وينقل 1 للخانة التالية.

مثال: أوجد ناتج جمع العددين الثمانيين التاليين: $(25)_8 + (47)_8$.

الجواب: الناتج في هذا المثال هو $(74)_8$. لأن (5 + 7) بالخانة الأولى من كلا العددين هو (12) أكبر من قيمة أساس النظام الثماني، وبالتالي هذه القيمة لا تنتمي للنظام الثماني لأنها من خارج مكوناته، ولابد من تحويلها إلى ما يقابلها في النظام الثماني. ولإجراء عملية التحويل بالطريقة المباشرة، يطرح من هذا الناتج قيمة الأساس (8) فيبقى (4) تثبت بالخانة (الأولى) المقابلة وينقل واحد للخانة التالية. والأمر نفسه يتكرر عند جمع 2 و 4 بالخانة الثانية حيث مجموعهما الرقم 6 مضافاً له 1 من الخانة السابقة ليصبح الناتج 7 يكتب كما هو، لأنه أصغر من 8.

ثانياً - الطرح في النظام الثماني

يفضل خلال عملية الطرح أن تستخدم طريقة المتممات، كما تم ذلك في النظامين العشري والثنائي، أي باستخدام المتمم للأساس (8) أو المتمم للأساس ناقص واحد (المتمم السباعي)

مثال: أوجد ناتج عملية الطرح للعددين الثمانيين $(45)_8 - (76)_8$.

الجواب: نبين فيما يلي عملية الطرح بطريقتين

الأولى: باستخدام المتمم بالنسبة للأساس ناقص واحد، أي باستخدام المتمم السباعي، على اعتبار أن القيمة (32) تمثل المتمم السباعي للمطروح (45). يكون الناتج الأولي

$30 = 76 + 32$ بعد إهمال النقل الأخير، وبما أنه تم اعتماد المتمم للأساس -1 يضاف واحد للنتائج ليصبح الناتج النهائي: $(31)_8 = 30 + 1$.

الثانية: باستخدام طريقة المتمم للأساس، يتم الحصول على نفس النتيجة أيضاً، لأن المتمم الثماني للمطروح (45) هو (33)، وبالتالي $31 = 76 + 33$ ، ويكون الجواب بعد حذف النقل (أقصى واحد من اليسار) هو $(31)_8$.

وبالنسبة للأعداد التي تحتوي الفاصلة، فإن الأمر مماثل لما سبق بالنسبة للأنظمة الأخرى. إذ تتم عملية الطرح وكأن الفاصلة غير موجودة، وبعد الانتهاء من عملية الطرح تعاد الفاصلة إلى موضعها النسبي. فمثلاً $(12.15)_8 = (51.23)_8 - (63.4)_8$.

2-3-3 العمليات الحسابية في النظام الستة عشري

النظام الستة عشري مماثل أيضاً للنظام الثماني من حيث عدم الاستخدام في عمليات المعالجة الحسابية ضمن الدارات وبالتالي سيتم الاكتفاء بعرض عمليتي الجمع والطرح فقط.

أولاً- الجمع في النظام الستة عشري

تتم عملية الجمع بشكل عشري، ويمثل الناتج بشكل ستة عشري. وهنا نميز بين حالتين: آ- إذا كان مجموع أي رقمين متقابلين من عددين أقل من الرقم 16 (أساس النظام) يكتب ناتج الجمع كما هو من دون تغيير.

مثال: أوجد ناتج جمع الأعداد الستة عشرية $(3C03)_{16} + (A39B)_{16}$ ؟

الجواب: عندما يتم جمع الرقمين 3 و B بالخانة الأولى من كلا العددين، تعوض B بقيمتها وهي $(11)_{10}$ ويصبح ناتج جمع $B + 3$ القيمة $(14)_{10}$ ، ولكن هذه النتيجة تعادل الرمز E، لذا يكتب بالخانة الأولى من الناتج الرمز E، وبنفس الأسلوب تتم التكملة ببقية الخانات، فيكون الناتج النهائي هو $(DF9E)_{16}$.

ب- إذا كان مجموع الرقمين المتقابلين من عددين أكبر أو يساوي العدد 16 يسجل الفرق ما بين المجموع والقيمة 16 بالخانة المقابلة، وينقل واحد للخانة التالية.

مثال: أوجد ناتج جمع الأعداد الستة عشرية $(5893)_{16} + (F70D)_{16}$.

الجواب: بما إن ناتج جمع الرقمين 3 و D بالخانة الأولى من كلا العددين هو 16، وأن 16 ليست من مكونات النظام الستة عشري، يكتب بالخانة الأولى من الناتج حاصل فرق $16 - 16 = 0$ ، وينقل واحد للخانة التالية. والناتج النهائي هو $(14FA0)_{16}$.

ثانياً- الطرح في النظام الستة عشري

تحول عملية الطرح إلى عملية جمع باستخدام طريقة المتممات.

مثال: أوجد ناتج طرح العددين: $(A8F)_{16} - (F9A)_{16}$.

الجواب: باستخدام الطريقة المباشرة، يلاحظ من الخانة الأولى أن $A - F$ غير ممكن من دون استعارة. عند استعارة واحد من الخانة الأعلى يصبح الوضع وكأن $1A - F$ (حيث $(1A)_{16} = (26)_{10}$) أي $26 - 15 = 11$ ، ولكن القيمة (11) هي عدد عشري يعادل في النظام الستة عشري الرمز B وهي قيمة الخانة الأولى من الناتج. وبنفس الأسلوب، يلاحظ أن الوضع بالخانة الثانية أصبح على الشكل: $8 - 8 = 0$ وأخيراً بالخانة الثالثة $F - A = 5$ ويصبح الجواب النهائي $(50B)_{16}$. يمكن الحصول على نفس النتيجة باستخدام طريقة المتممات وفق التالي:

أولاً - باستخدام المتمم للأساس ناقص واحد، أي للقيمة $F = 15$. بأخذ متمم المطروح $(A8F)_{16}$ إلى القيمة F يكون الناتج 570، وبجمعه مع المطروح منه، يكون الناتج الأولي $F9A + 570 = 50A$ ، وبعد إضافة واحد لهذا الناتج، يصبح الجواب النهائي $(50B)_{16}$. ثانياً- باستخدام المتمم للأساس، أي للقيمة 16 نحصل على نفس النتيجة.

مثال: أوجد القيمة الثنائية للمتحويل Y المعطى بالعلاقة: $Y = A / B - C.D$ إذا علمت أن المتحولات معطاة بالشكل: $A = (10011001)_2$, $B = (51)_{10}$, $C = (3)_8$, $D = (1B)_{16}$ على أن تمثل جميع المتحولات المستخدمة بـ 8 bits، وأن تجري كل العمليات الحسابية المبينة بالعلاقة في النظام الثنائي.

الجواب: نقوم في الخطوة الأولى بتحويل كافة المعطيات إلى ما يقابل كل منها بالنظام الثنائي وبالتعويض يكون:

$$Y = (10011001) / (110011) - (11)^* (11011)$$

$$Y = 00000011 - 01010001 = (10110010)_2 = (-78)_{10}$$

2-4 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

- 1- المتمم الأحادي للعدد الثنائي $(1001)_2$ هو
a- 1001 b- 1110 c- 0110 d- 0111
- 2- المتمم الثنائي للعدد الثنائي $(1000\ 0000)_2$ هو
a- (0111 1111) b- (1000 0001)
c- (1000 0000) d- None of all
- 3- يقابل العدد الثنائي 11010011101101 بالنظام الستة عشري الشكل
a- (D3B1)₁₆ b- (34ED)₁₆
c- (3B945)₁₆ d- None of all
- 4- العدد الثنائي المؤشر $(1000\ 1111)_2$ ممثلاً بـ 8 bits يساوي:
a- $(-113)_{10}$ b- $(+15)_{10}$ c- $(+143)_{10}$ d- $(-15)_{10}$
- 5- العدد الثنائي غير المؤشر $(1000\ 1000)_2$ ممثلاً بـ 8 bits يساوي:
a- $(210)_8$ b- $(136)_{10}$ c- $(88)_{16}$ d- None of all
- 6- العدد العشري $(-45)_{10}$ ممثلاً بـ 8 bits يعني:
a- $(1010\ 1101)_2$ b- $(1101\ 0011)_2$
c- $(0001\ 0011)_2$ d- none of all
- 7- الأعداد الثنائية بالفاصلة العائمة والدقة البسيطة تمثل بـ
a- 32 bits b- 64 bits c- 80 bits d- None of all
- 8- مجال الأعداد المؤشرة على مستوى البايت يتراوح ضمن المجال:
a. 1000 0000 to 0111 1111 b. 0000 0000 to 0111 1111
c. 1000 0000 to 0000 0000 d. None of all
- 9- المجموع الثنائي للعددين الثنائيين $(11111)_2$ and $(00001)_2$ يكون
a- 100100 b- 100000 c- 100001 d- 100010
- 10- ناتج طرح العددين الثنائيين $(10111010)_2$ - $(11101100)_2$ يكون
a- 10010001 b- 00110010 c- 10100110 d- none of all
- 11- ناتج جمع العددين $(421)_8 + (460)_8$ هو
a. $(101)_8$ b. $(881)_8$ c. $(1001)_8$ d. None of all

12- ناتج طرح العددين $(460)_8 - (421)_8$ هو

- a. $(38)_8$ b. $(37)_8$ c. $(105)_8$ d. None of all

13- ناتج جمع العددين $(ABCD)_{16} + (5433)_{16}$ هو

- a. $(1\ 0000)_{16}$ b. $(F00F)_{16}$
c. $(110F0)_{16}$ d. None of all

14- ناتج طرح العددين $(290D)_{16} - (1FA8)_{16}$ هو

- a. $(965)_{16}$ b. $(305)_{16}$ c. $(48B5)_{16}$ d. None of all

ثانياً: أسئلة متفرقة

1- مثل كلاً من العددين المؤشرين $(+ 41)$ و (-31) باستخدام المتمم للثلاثين وحدد طول السجل المناسب لتخزين كل منهما.

2- أوجد المتمم الأحادي والمتمم الثنائي لكل من القيمتين:

- a) $(1100010100)_2$ b) $(1001011.001101)_2$

3- أوجد كلاً من المتمم السباعي والعشري للعدد 6571.

4- أوجد ناتج جمع وضرب العددين الثنائيين 11010 و 110 من دون تحويلهما إلى النظام العشري.

5- حدد على مستوى البايت كل من المتمم للواحد وللثلاثين لكل من الأعداد المبينة أدناه، وبين القيمة المقابلة في النظام العشري مع الإشارة المناسبة.

- a) 1000 0000 b) 0000 0001 c) 0000 0000 d) 1111 1111

6- بين ناتج قسمة العددين الثنائيين $101011 / 11001$ ومراحل عملية القسمة.

7- أوجد ناتج طرح الأعداد الستة عشرية: $4F2A6D - D593$ باستخدام كل من المتمم للأساس والمتمم المخفض للأساس.

8- مثل العدد الثنائي الكسري 1100.111001 باستخدام الفاصلة العائمة؟

9- أوجد القيمة الثنائية للمتحول Y المعطى بالعلاقة: $Y = D / (B - C) - A$ إذا

علمت أن المتحولات A, B, C, D معطاة بالشكل التالي: $A = (100\ 0110)_2$ ،

$B = (1027)_{10}$ ، $C = (524)_8$ ، $D = (2C)_{16}$ على أن تمثل جميع المتحولات

المستخدمة في سجلات 8 bits، وأن تجري كل العمليات بالعلاقة في النظام الثنائي.

الفصل الثالث: الأرمزة Codes

يجب عند التعامل مع المعطيات على مختلف أنواعها في الدارات الرقمية (بما في ذلك الحواسيب الالكترونية) أن تكون هذه المعطيات مرمزة ثنائياً، انطلاقاً من أن الدارات الرقمية تتعامل مع سويتين فقط للإشارة، هما 0V الذي يقابل الصفر و 5V+ الذي يقابل الواحد. ويجب أن يكون هذا الترميز وفق أشكال محددة تدعى أرمزة Codes. يمثل كل رماز (شفرة) شكلاً اصطلاحياً لتمثيل المعلومات، ويمتاز بطول محدد. ويمكن برماز ثنائي مثلاً، طوله n bits تمثيل حتى 2^n تركيبة مختلفة من الأصفار والوحدات بالقيم من 0 حتى $2^n - 1$. يؤخذ منها بحسب نوع الرماز عدد محدد من التراكيب بحيث كل تركيبة تشكل عنصر واحد من المجموعة التي تم ترميزها. وبالتالي لترميز مجموعة من المعلومات مؤلفة من 8 عناصر مثلاً، يلزم ثلاثة بتات، بحيث يخصص لكل عنصر من هذه العناصر الثمانية تركيبة واحدة من التراكيب 000, 001, ..., 111. يختلف كل رماز عن الآخر بحسب مجال التطبيق. يمكن بشكل عام تصنف الأرمزة إلى المجموعات التالية:

- 1- أرمزة عددية، تستخدم لتمثيل الأرقام Numerical Codes
- 2- أرمزة أبجدية رقمية Alpha numeric Codes، تستخدم لتمثيل البيانات الرقمية والحرفية والرموز الخاصة.
- 3- أرمزة لضغط البيانات وتشفيرها، تستخدم لضغط البيانات (تقليص عدد البتات المحجوزة لتمثيلها) Compression and Encryption & Decryption Codes
- 4 - أرمزة لاكتشاف الأخطاء وتصحيحها Error-Detection & Correction Codes وتستخدم لكشف الخطأ عند حدوثه وتصحيحه تلقائياً.

1-3 الأرمزة العددية

نشير فيما يلي إلى الأرمزة التي تمثل كل رقم عشري باستخدام مكونات النظام الثنائي ونعنيها بالأرمزة الثنائية Binary codes. وبحسب تركيبة الرماز ومجال استخدامه يمكن أن نميز بين نوعين أساسيين من الأرمزة الثنائية، هما: أ - الأرمزة الثنائية الموزنة
ب - الأرمزة الثنائية غير الموزنة

3-1-1 الأرمزة الثنائية الموزنة

يقصد بالأرمزة الثنائية الموزنة Weighted codes، الأرمزة التي يمكن فيها تمثيل كل رقم عشري وفق أوزان محددة. ويوجد العديد من الأرمزة الموزنة بأوزان مختلفة مثل الأوزان: ... 6311، 84-2-1، 2421، (BCD) 8421 ندرس منها:

3-1-1-1 الرماز BCD

يمثل الرماز BCD (Binary Coded Decimal) أول محاولة لتمثيل المعلومات في الحاسوب، وهو يركز على تمثيل كل رقم عشري Decimal digit بما يوازيه في النظام الثنائي. وانطلاقاً من أن الحاسوب لا يقبل إلا القيم الثنائية، تم اعتماد الرماز BCD لتمثيل الأرقام العشرية برماز ثنائي يسمح القيام بالعمليات الحسابية على الأعداد العشرية. يُعدّ الرماز BCD موزناً، لأن أي قيمة بهذا الرماز منظمة وفق الأوزان 2^0 2^1 2^2 2^3 التي تقابل القيم 1 2 4 8 (ويسمى أيضاً الرماز BCD بالـ 8421 code). وبالنظر إلى أن أكبر رقم عشري هو (9)، ويمثل ثنائياً بالشكل (1001)، يلزم أربع خانات لتمثيل كل من هذه الأرقام، بحيث يكون لكل رقم عشري من 0 حتى 9 قيمة محددة بطول 4 bits. وبما أنه يمكن بأربع خانات تمثيل حتى $2^4 = 16$ قيمة مختلفة، فإن الرماز BCD يستخدم من هذه التراكيب الست عشر فقط عشرة تراكيب هي: 0000 حتى 1001، أما المجموعات الستة المتبقية، أي التراكيب من (1010) حتى (1111) التي تقابل الأعداد العشرية من 10 حتى 15، لا معنى لها بالرماز BCD، ولا يجب استعمالها. وبسبب هذه التراكيب الستة التي لا تدخل بتركيبية الرماز BCD، ينتج مشاكل عند استعمال هذا الرماز بإجراء العمليات الحسابية على البيانات الممثلة بوساطته، لذلك تعمل الدارات الرقمية الخاصة بهذا الرماز على منع استعمال المجموعات الستة الفائضة بإضافة عدد هذه المجموعات، أي المقابل الثنائي للرقم (6) إلى نتيجة أي عملية إن لم تكن بالرماز BCD لتحويلها إلى ما يقابلها بالرماز BCD. مثال: لإجراء عملية جمع للعديدين: (1000) و (0101) بالرماز BCD والذين يقابلان الرقمين العشريين (8) و (5)، يلاحظ عند الجمع الثنائي للعديدين أن الناتج سيكون كالتالي: $1000 + 0101 = 1101$ وهذا يقابل العدد العشري (13)، لكن الشكل الثنائي

للناتج لا يُعدّ من الرماز BCD (لأنه من خارج مجموعة الأعداد من 0000 حتى 1001) بالتالي النتيجة خاطئة. لإصلاح الناتج يجب إضافة الرقم 6 أي $(0110)_2$ إلى

$$\begin{array}{r} 1011 \\ + 0110 \\ \hline 1\ 0011 \end{array}$$

القيمة الناتجة أعلاه مكونة من خمس خانات، تقسم إلى مجموعتين بكل منهما أربع خانات ابتداء من اليمين وفق الشكل 0011 : 0001 . وبتحويل التركيبة الثنائية لكل أربع بتات إلى ما يقابلها عشرياً، يمكن الحصول على الجواب الصحيح بالرماز BCD وهو 3 1، أي $(13)_{10}$ وهو الجواب الصحيح، بمعنى أن لكل رقم من الرماز BCD يخصص أربع خانات. ويلاحظ من المثال أعلاه وجود فرق ما بين تحويل الأعداد العشرية إلى أعداد ثنائية وما بين الترميز الثنائي للأرقام العشرية. فمثلاً، بالنسبة للعدد (99) يمكن النظر إليه على أنه عدد عشري، وبالتالي عند تحويله إلى النظام الثنائي يتم الحصول على سلسلة متلاحقة من الأرقام الثنائية وفق الشكل $(110011)_2 = (99)_{10}$ ، لكن عند تمثيل نفس العدد بالرماز BCD يلاحظ أنه يأخذ الشكل: $(1001\ 1001)_2 = (99)_{BCD}$ ، حيث أسند لكل خانة بالرماز BCD أربع خانات ثنائية، أي أن الأعداد في الرماز BCD هي أعداد عشرية وليست أعداداً ثنائية ولو كانت تستخدم البتات الثنائية في تمثيلها. الفرق الجوهرى بين العدد العشري والعدد بالرماز BCD هو أن الأعداد العشرية تكتب بوساطة الأرقام 0, 1, ..., 9 في حين الرماز BCD يستخدم الترميز الثنائي لهذه الأرقام العشرة، أي: 0000, ..., 1001. والقيمة العشرية هي ذاتها تماماً في الحالتين، ويمكن كتابة نفس القيمة بثلاثة أنظمة وفق التالي: $(1001\ 1001)_{BCD} \sim (110011)_2 \sim (99)_{10}$. يخصص بالرماز BCD لكل رقم لعدد مكون من أكثر من رقم أربع خانات. يظهر الجدول 1-3 المقابل الثنائي بالرماز BCD لأول عشرين قيمة عشرية. أما بخصوص ما يعرف باسم الحساب العشري، تتم العمليات الحسابية بالرماز BCD، وفق ما هو مبين بالأمثلة أدناه، حيث إن تعديل أي قيمة لل BCD يتم على مستوى كل خانة على حده.

الجدول 1-3: أول عشرين قيمة عشرية بالرماز BCD

Dec	2^3 2^2 2^1 2^0 BCD Code	Dec	2^3 2^2 2^1 2^0 2^3 2^2 2^1 2^0 BCD Code
0	0 0 0 0	10	0 0 0 1 0 0 0 0
1	0 0 0 1	11	0 0 0 1 0 0 0 1
2	0 0 1 0	12	0 0 0 1 0 0 1 0
3	0 0 1 1	13	0 0 0 1 0 0 1 1
4	0 1 0 0	14	0 0 0 1 0 1 0 0
5	0 1 0 1	15	0 0 0 1 0 1 0 1
6	0 1 1 0	16	0 0 0 1 0 1 1 0
7	0 1 1 1	17	0 0 0 1 0 1 1 1
8	1 0 0 0	18	0 0 0 1 1 0 0 0
9	1 0 0 1	19	0 0 0 1 1 0 0 1

مثال : اجمع وفق الرماز BCD العددين التاليين: $836 + 645 = 1481$

الجواب: $836 \sim 1000\ 0011\ 0110$

$+ 645 \sim 0110\ 0100\ 0101$

النتائج الأولى لعملية الجمع الثنائي للعددين

$1110\ 0111\ 1011$

ليس بالرماز BCD. لجعل الجواب بالرماز BCD، يفترض إن كان الناتج في أي وحدة بطول 4bits (Nibble) أكبر من الرقم (9) أن يضاف له الرقم (6). يقسم الناتج (1110 0111 1011) إلى ثلاث وحدات ابتداء من اليمين بطول أربع خانات لكل منها. فيلاحظ أن الوحدتين الأولى والثالثة تحتاج إلى تصحيح، حيث يمكن تصحيح كل منهما

على حدة أو الاثنتين معاً كالتالي: $1110\ 0111\ 1011$

$+ \quad 110 \quad \quad 110$

$1\ 0100\ 1000\ 0001$

وبأخذ كل 4 bits معاً من اليمين إلى اليسار يكون المقابل العشري للنتائج هو العدد 1481. تحتاج بعض الأعداد بالرماز BCD إلى عدد أكبر من البتات مما تحتاجه مكافئاتها في النظام الثنائي. فعوضاً عن ثلاث وحدات بطول 12 bits لكل عدد بالمثال أعلاه، يحتوي الناتج أربع وحدات بطول 16 bits ويقابل $(1\ 4\ 8\ 1)_{BCD}$.

مثال آخر: أوجد وفق الرمز BCD ناتج طرح العددين التاليين: (142 - 264).
 الجواب: يمكن استخدام أية طريقة من الطرائق الثلاثة المعروفة لتمثيل الأعداد السالبة.
 ونظراً لندرة استخدام طريق المقدار والإشارة في الحاسوب لتمثيل الأعداد السالبة، نظهر فيما يلي استخدام طريقة المتممات. تمثل الإشارة في الرمز BCD مثلها مثل أي رقم عشري، وبالتالي يلزم لتمثيلها 4bits، بحيث يخصص لإشارة (+) التركيبية الثنائية 0000 المقابلة للقيمة 0، ولإشارة (-) التركيبية الثنائية $(1001)_2$ المقابلة للقيمة $b-1$ بالنسبة للنظام العشري. وبما أن المتمم التساعي للمطروح 142 هو 857 تحول عملية الطرح المطلوبة إلى جمع وفق الشكل التالي:

$$\begin{array}{r}
 0264 \sim 0000\ 0010\ 0110\ 0100 \\
 + 9857 \sim 1001\ 1000\ 0101\ 0111 \\
 10121 \sim 1001\ 1010\ 1011\ 1011 \\
 + \quad \quad 0110\ 0110\ 0110 \\
 \hline
 1010\ 0001\ 0010\ 0001 \\
 + \quad 0110 \\
 \hline
 1\ 0000\ 0001\ 0010\ 0001
 \end{array}$$

وبإهمال المنقول الأخير (لأن العملية هي طرح)، وبإضافة واحد للمتمم التساعي حتى يصبح متمماً للأساس (10) وفق الشكل:

$$\begin{array}{r}
 0000\ 0001\ 0010\ 0001 \\
 + \quad \quad \quad 0001 \\
 \hline
 0000\ 0001\ 0010\ 0010
 \end{array}$$

يصبح الناتج النهائي (+122).

3-1-1-2 الرمز 8 4-2-1

بما أن المهم في كل رمز هو تمثيل الأرقام العشرية من 0, 1, ..., 9 فإنه في كل رمز يلزم 4bits على الأقل لتمثيل هذه الأرقام العشرة، ويقال أن طول الرمز 4 bits يؤخذ من تراكيبيها الستة عشر الممكنة أول عشرة تراكيب فقط والتراكيب الباقية غير المستخدمة، يجب تجنبها. وعند تمثيل رقم ما برمز موّزون نضع 1 تحت الوزن اللازم لتشكيل قيمة الرقم و0 تحت الوزن الذي لا يلزم. تمتاز بعض الأرمزة الموزّنة مثل الرمز 8 4-2-1 بالإضافة إلى الأوزان الخاصة بها عند تمثيل الأرقام العشرية بسمات أخرى كتخصيص أوزان موجبة وسالبة للعدد المراد ترميزه. فمثلاً، التركيبية الثنائية (0101) بالرمز 8 4-2-1 تكتب بالشكل

التالي: $1x(-1) + 0x(-2) + 1x(4) + 0x(8) = -1 + 0 + 4 + 0 = 3$ وهي تقابل القيمة $(-1 + 0 + 4 + 0) = 3$ وبالتالي يكتب العدد بالشكل: $(0101)_2 = (3)_{84-2-1}$. كما أن $(1010)_{84-2-1} = (6)_{10}$ أي أن العدد العشري (63) مثلاً المكون من خانتين، يمكن كتابته بهذا الرمز وفق الشكل $(1010 \ 0101)_{84-2-1} = (63)_{10}$. ويبين الجدول 3-2 الرمزين 2 4 2 1 و 8 4-2-1 الموزنيين اللذين يمكن مصادفتها بتمثيل المعطيات.

الجدول 3-2: الرمزان الموزنان: 2 4 2 1 و 8 4-2-1

Dec. Digit	Code 2 4 2 1	Code 8 4-2-1
0	0 0 0 0	0 0 0 0
1	0 0 0 1	0 1 1 1
2	0 0 1 0	0 1 1 0
3	0 0 1 1	0 1 0 1
4	0 1 0 0	0 1 0 0
5	1 0 1 1	1 0 1 1
6	1 1 0 0	1 0 1 0
7	1 1 0 1	1 0 0 1
8	1 1 1 0	1 0 0 0
9	1 1 1 1	1 1 1 1

الرمز 2 4 2 1: يحد هذا الرمز واحداً من الأرمزة بطول 4 bits ومن الأرمزة المتممة ذاتياً Self Complementing أي الأرمزة التي يمكن منها إيجاد أحد متممات العدد المعطى. يسمح الرمز 2 4 2 1 بوجود أكثر من إمكانية لتمثيل نفس الرقم. فمثلاً، يمكن التعبير عن الرقم 6 بالرمز 2 4 2 1 بالشكلين: 2 4 2 1 أو 0 1 1 0

لأن مجموع أوزان البتات المقابلة يساوي في كلا التركيبتين الرقم (6). والجواب الصحيح هو 1100 وليس 0110. يمكن انتقاء المقابل الصحيح لأي رقم بالرمز 2 4 2 1 بأخذ التركيبة التي يقابل متممها الأحادي المتم التساعي للرقم المعطى. بمعنى أنه يمكن إيجاد المتمم التساعي للعدد العشري المعطى مباشرة بالتركيبة المعتمدة بتغير الوحدات إلى أصفار، والأصفار إلى واحدات. فمن أجل الرقم (6) معلوم أن المتمم التساعي له هو الرقم (3) وأن

الترميز الثنائي للرقم (3) هو $2(0011)$ وهو ما يقابل المتمم للواحد للتركيبة التي تم اختيارها بالنسبة للرقم (6) في هذه الرماز، أي (1100). التركيبة الثانية (0110) الواردة لنفس الرقم (6) في هذا الرماز لا تؤخذ، لأنها لا تقابل أي شيء ذي معنى. ولنفس الأسباب بالنسبة للرقم (5) يتم بالرماز 2 4 2 1 اختيار التركيبة (1011) وليس التركيبة (0101) لأن المتمم للواحد للتركيبة (1011) هو (0100) الذي يقابل عشريا القيمة (4) باعتبارها متمماً تساعياً للعدد العشري (5). إضافة لما تم عرضه من أرمزة موزونة، يوجد أيضاً بعض الأرمزة الموزونة الأخرى وكل منها بتركيبة خاصة، مثل الرماز Biquinary بطول 7 bit بالأوزان (5 0 4 3 2 1 0)، حيث يسمح في كل كلمة من هذا الرماز بوجود واحدین على الأكثر تحت الأوزان المناسبة، كما هو مبين لبعض الأرقام العشرية أدناه، بحيث الصفر اليساري في الأوزان يشغل بواحد لكل الأرقام من 0 حتى 5. فمثلاً:

$$(0)_{10} = (0100001)_{5043210}$$

$$(3)_{10} = (0101000)_{5043210}$$

$$(7)_{10} = (1000100)_{5043210}$$

$$(9)_{10} = (1010000)_{5043210}$$

3-1-2 الأرمزة الثنائية غير الموزونة

لا توجد في مجموعة الأرمزة غير الموزونة أوزان محددة لتمثيل الأرقام العشرية، كما هو الحال بالنسبة للأرمزة الموزونة، لكن يوجد لكل منها مبدأ محدد. ويختلف المبدأ المستخدم بكل رماز عن الآخر. يمكن تشكيل العديد من الأرمزة غير الموزونة، ولكن هذا قد لا يكون له أي تطبيق. وكأمثلة على هذا النوع من الأرمزة التي تصادف لها بعض التطبيقات، نعرض فيما يلي بشيء من التفصيل الرمازين Gray & Excess - 3 Codes.

3-1-2-1 الرماز Excess - 3

هو من الأرمزة البسيطة التي كانت تستخدم بالحواسيب القديمة، ويمثل بطول 4 bits. ويتلخص مبدأ هذا الرماز غير الموزون بأنه يضاف لكل عدد عشري القيمة 3 وبعد ذلك يؤخذ المقابل الثنائي للقيمة الجديدة وفق ما هو مبين بالجدول 3-3، حيث لتمثيل عدد ما يخصص 4 bit لكل رقم على حدة، والنتيجة سيكون قيمة لاتخضع لأي وزن محدد.

الجدول 3-3: الرمز Excess-3

Dec. Digit	الرمز Excess-3
0	0 0 1 1
1	0 1 0 0
2	0 1 0 1
3	0 1 1 0
4	0 1 1 1
5	1 0 0 0
6	1 0 0 1
7	1 0 1 0
8	1 0 1 1
9	1 1 0 0

يمكن بهذا الرمز التكملة بترميز بقية الأعداد التي تمثل بـ 4 bit بحيث أي رقم يزيد عن الأربع خانات يحذف وبالتالي: $(10)_{10} = (1101)_{\text{Excess-3}}$ أما $(15)_{10} = (0010)_{\text{Excess-3}}$ لأن $15 + 3 = 18$ و $(18)_{10} = (10010)_2$ يؤخذ منها فقط أول أربع خانات بعد حذف البت الخامس ليعود الناتج وكأنه يقابل العدد 2 الذي لم ترد التوافقية الثنائية المقابلة له في جدول الترميز أعلاه. لترميز عدد ما مكون من خانتين أو أكثر في الرمز Excess-3، يخصص بالمرحلة الأولى أربع خانات لكل رقم بالعدد المعطى، ثم يتم بالمرحلة الثانية حساب التركيبة المقابلة. فمثلاً، لترميز العدد العشري (64) بالرمز Excess-3، يحول العدد المعطى بالمرحلة الأولى إلى الشكل (97) بعد إضافة القيمة 3 إلى كل خانة من خانات العدد. ويكتب بالمرحلة الثانية المقابل الثنائي لـ (97) بترميز كل رقم ثنائياً، ليصبح الناتج من الشكل: 1001 0111.

يمتاز الرمز Excess-3 أيضاً مثل الرمز الموزن 2 4 2 1 بخاصية الأرمزة المتممة ذاتياً. فمثلاً، مكافئ العدد العشري (64) في الرمز Excess-3 هو 1001 0111. بأخذ المتمم للواحد لهذا العدد يكون الناتج 0110 1000 وهو يعادل العدد 68، ولكن العدد 68 بالرمز Excess-3 يقابل العدد العشري 35 الذي يمثل بدوره المتمم التساعي للعدد (64).

3-1-2 رماز كراي Gray Code

بما أن رماز كراي هو من الأرمزة غير الموزونة، لا يمكن ضبط مكونات الأعداد العشرية فيه وفق وزن محدد. يعتمد هذا الرماز مبدأ التجاور. يظهر الجدول 3-4 تمثيل 16 قيمة برماز كراي من 0 حتى 15 (على مدى 4 bits).

الجدول 3-4: رماز كراي

Dec. No.	Gray Code			
	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0
8	1	1	0	0
9	1	1	0	1
10	1	1	1	1
11	1	1	1	0
12	1	0	1	0
13	1	0	1	1
14	1	0	0	1
15	1	0	0	0

يقصد بالتجاور بين كلمتين اختلاف الكلمتين عن بعضهما ببت واحد. وبناء على ذلك من أجل كلمات بطول 8 bit يقال عن الكلمتين 0011 0001 و 0010 0001 إنهما متجاورتان بغض النظر عن المقابل العشري لهما لاختلافهما عن بعضهما بالبت الخامس. بالتالي الكلمة 0000 0000 التي تقابل الصفر يمكن أن تكون مجاورة لثمانية قيم مختلفة من 0000 0001 وحتى 1000 0000. يتضح من الجدول أعلاه أن الانتقال من كلمة برماز كراي لأخرى مجاورة يتم بتغيير خانة واحدة فقط من خانات الكلمة السابقة ابتداء من الخانة

الأدنى. فمثلاً، عند الانتقال من الكلمة (1010) التي تقابل في الجدول، العدد العشري 12 إلى الكلمة التالية (1011) يلاحظ أن التغيير قد حدث ببت واحد هو G_0 ، أما بقية البتات الأخرى فقد بقيت كما هي من دون تغيير. إذاً الكلمتان (1010) و (1011) متجاورتان. يدعى رماز كراي أيضاً بالرماز الدوراني Cyclic Code ويستخدم في توصيف مخطط كارنوف كما سنجد لاحقاً وغيرها من التطبيقات الأخرى.

3-2 الأرمزة الأبجدية الرقمية

تستخدم أرمزة هذه المجموعة لتمثيل مختلف أشكال البيانات على مختلف أنواعها بشكل مباشر أثناء تخزينها ضمن وسائط تخزين محدّدة أو نقلها من مكان إلى آخر عبر وسائط نقل المعطيات بشكل غير مباشر، وهي عديدة نذكر منها:

1- الرماز التلفوني Telegraphic Code، ويمثل رماز من خمسة عناصر، أربعة منها لتمثيل المعلومات المفيدة، والبت الخامس يستعمل للندية Parity bit.

2- الرماز المتبادل العشري المرمز ثنائياً الموسع EBCDIC المطور من IBM Extended Binary Coded Decimal Interchange Code، ويتألف هذا الرماز من ثمانية عناصر (بتات) لتمثيل المعلومات.

3- رماز ASMO لترميز بيانات غير لائتينية.

4- رماز ASCII الذي اعتمد من قبل اللجنة العالمية للتلفون والتلغراف ISO. وبالنظر إلى أن الرماز ASCII هو من أكثر أرمزة هذه المجموعة استخداماً، سنكتفي هنا بتوصيف هذا الرماز فقط.

3-2-1 رماز ASCII

مع بدايات ظهور الحاسبات كانت كل شركة منتجة للحواسيب تستخدم ترميز خاص بها لتمثيل البيانات من أرقام وأحرف ورموز مما جعل من الصعوبة بمكان نقل برنامج ينفذ على حواسيب شركة ما لتنفيذه على حواسيب شركة أخرى على اعتبار أن حرف A مثلاً من الشركة الأولى قد يقابل الحرف R من شركة أخرى في الوقت الذي يقابل به الحرف R رمزاً آخر مغاير. بالتالي تم اعتماد معيار موحد لتمثيل البيانات لدى كل الشركات هو الترميز

المعياري الأمريكي (والعالمي) لتبادل المعلومات بين الحواسيب اختصاراً الرمز ASCII (American Standard Code for Information Interchange)، وأصبح الرمز القياسي Standard لتمثيل المعطيات في مختلف الحواسيب. يتشكل رمز ASCII من 7 bits للمعلومات المفيدة وبت إضافي يستخدم للتحكم أو النذية Parity bit. يبين الجدول 3-5 رموز ASCII الكامل من دون بت النذية لمختلف الرموز المعمول بها في الحواسيب. ويمكن وفق هذا الرمز تمثيل $2^7 = 128$ رمز مختلف من الأحرف والأرقام والرموز الخاصة. ويمكن باستخدام البت الخاص بالنذية تمثيل رموز إضافية يصل عددها إلى $2^8 = 256$ رمز مختلف (لتمثيل رموز أي أبجدية أخرى غير الانكليزية). نظم جدول ASCII بحيث يحتوي على $2^4 = 16$ سطر من 0000 حتى 1111 وثمانية أعمدة ($2^3 = 8$) بالترميز من 000 حتى 111 (على اعتبار أن البت الثامن مخصص للنذية $b7 = 0$ غير مستخدم). يمثل ASCII مجموعة من الخلايا ($8 \times 16 = 128$ خلية) تضم كافة حروف اللغة الانكليزية بشكلها الكبير والصغير وكذلك الأرقام العربية من 0 حتى 9 والرموز الخاصة وأربعة وثلاثين أمراً تحكيمياً، أشير إليها في الجدول بأسماء مختصرة بالعمودين الأول والثاني. ويكتب الرمز المقابل لكل عنصر مستقل من المعلومات في الجدول بالشكل الثنائي بكتابة رقم العمود Column الذي ينتمي له الرمز (مكوناً من 3bits) ابتداء من اليسار باتجاه اليمين، يعقبه رقم السطر Row الموجود به الرمز (مكوناً من 4 bits) ليصبح الطول الإجمالي للرمز الواحد 7 bits. فمثلاً، بالنسبة للأحرف: عند النقر بلوحة المفاتيح على حرف مثل f الصغير (الموجود بالسطر السادس والعمود الرابع) يتم إدخال التركيبة من الخلية المقابلة بالرمز 46H أو $(100\ 0110)_2$ ، وعند النقر على الحرف f مع Shift أي حرف F كبير، يتم إدخال التركيبة التي تقابل السطر السادس والعمود السادس، أي $(1100110)_2 = 66H$ ، حيث تم الحصول على f الصغيرة من F الكبيرة بإضافة القيمة 20H (أي $46H + 20H = 66H$) وبالتالي الفرق بين الرمز المقابل لحرف بين شكله الكبير وشكله الصغير هو 20H .

الجدول 5-3: رماز ASCII

		b ₇ b ₆ b ₅ b ₄ , b ₇ = 0						
b ₃ b ₂ b ₁ b ₀		001	010	011	100	101	110	111
0- 0000	NUL	DLE	SP	0	@	P	'	p
1- 0001	SOH	DC1	!	1	A	Q	a	q
2- 0010	STX	DC2	"	2	B	R	b	r
3- 0011	ETX	DC3	#	3	C	S	c	s
4- 0100	EOT	DC4	\$	4	D	T	d	t
5- 0101	ENQ	NAK	%	5	E	U	e	u
6- 0110	ACK	SYN	&	6	F	V	f	v
7- 0111	BEL	ETB	,	7	G	W	g	w
8- 1000	BS	CAN	(	8	H	X	h	x
9- 1001	HT	EM	)	9	I	Y	i	y
A-1010	LF	SUB	*	:	J	Z	j	z
B-1011	VT	ESC	+	:	K	[	k	{
C-1100	FF	FS	,	<	L	\	l	
D-1101	CR	GS	_	>	M	]	m	}
E-1110	SO	RS	.	=	N	^	n	~
F-1111	SI	US	/	?	O	-	o	DEL
NUL ~ NULI			SO ~ ShiftOut			ETB ~ End-of-		
SOH ~ Start Of Heading			SI ~ Shift In			Transm. Block		
STX ~ Start of Text			SP ~ SSpace			CAN ~ CANcel		
ETX ~ End of Text			DLE ~ Data-			EM ~ End of		
EOT ~ End Of Transmission			Link Esc.			Medium		
ENQ ~ ENquiry			DC1, 2, 3, 4 ~			SUB ~		
ACK ~ ACKnowledge			Device			SUBstitute		
HT ~ Horizontal Tab			Contrl1, 2,3, 4			ESC ~ ESCape		
LF ~ Line Feed			NAK ~			FS, GS, RS, US		
VT ~ Vertical Tab			Negative ACK			~ File, Group,		
FF ~ Form Feed			SYN ~			Record, Unit		
CR ~ Carriage Return			SYNchr. Idle			Separator		
DEL ~ DElete								

لتمثيل أي عدد عشري بالرماز ASCII، يجب إضافة 30H إلى قيمته الثنائية لوجود جميع الأرقام العربية بالعمود الثالث. بمعنى أن الرقم العشري (6) مثلاً يقابل بالرماز ASCII العمود الثالث والسطر السادس، أي التركيبة: $36H = 30H + 6_{10}$.

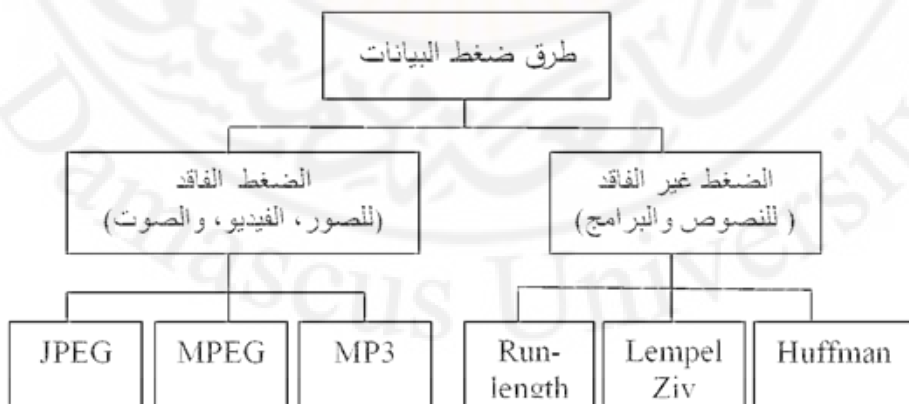
3-3 أرمزة ضغط البيانات وتشفيرها

يقصد بضغط البيانات Data Compression تقليص عدد البتات المحجوزة لتمثيل كل نقطة bits per pixel من صورة مثلاً لتخزينها أو إرسالها عبر الشبكة. وضغط الملفات هو فرع من فروع هندسة المعلومات أو هندسة الحاسبات وكذلك يستخدم كثيراً في هندسة الاتصالات، وهو يعتمد على قضيتين أساسيتين: 1- إزالة المعلومات الإضافية 2- إزالة المعلومات غير المهمة

ويقسم ضغط المعطيات إلى: 1- الضغط الفاقد Lossy Compression

2- الضغط المحافظ Lossless Compression

إن القيود الطبيعية التي تشوب عملية الإرسال أو الاتصال من بطئ وقابلية للضياع يستدعي الحاجة إلى تغيير الترميز، وذلك بترميز آخر أقل حجماً ويحمل نفس المعلومات، مما يسرع من عملية الإرسال، ويقلل من احتمال الضياع وورود الأخطاء. ويلاحظ إذا أردنا أن نضغط معلومة وجود عدد كبير من تقنيات الضغط. ويمكن تصنيف أهم تقنيات ضغط البيانات والتي تمثل شكل من أشكال الترميز) بالمخطط المبين بالشكل 1-3.



الشكل 1-3: أهم تقنيات ضغط البيانات

وبخصوص هذه المجموعة من الشيفرات نكتفي بالتعريف بواحدة منها فقط ولتكن تقنية Run-length encoding التي تعتبر من أبسط طرق ضغط المعلومات وتستخدم بضغط النصوص والبرامج (ضغط غير فاقد)، دون الدخول بتفاصيل أي من الشيفرات المتبقية باعتبار موضوعها يعد خارج سياق هذا الكتاب.

تتمثل تقنية Run-length encoding بالتالي: يؤخذ من رسالة ما ممثلة بشكل ثنائي الرقم ذو التكرار الأكثر من بين 0 و 1 (وليكن صفر مثلاً) ثم يتم استبدال عدد التكرارات للصفر كرقم عشري بشكل ثنائي ولا يشار إلى قيم الواحد على اعتبار أن الواحد سيأتي الصفر وهذا مانوضحه من خلال المثال التالي: بفرض لدينا رسالة مشفرة بالشكل الثنائي: Original data: (000000000000000010000110000000000000) الأصلية وفق التالي:

(000000000000000010000110000000000000)
 ↓ ↓ ↓ ↓
 14 4 0 12

ويصبح شكل المعطيات المضغوطة Compressed data وفق التالي:

1110 0100 0000 1100

3-4 أرمزة اكتشاف الأخطاء وتصحيحها

نظراً لإمكانية حدوث بعض الأخطاء أثناء نقل البيانات الثنائية من مكان إلى آخر أو في أثناء استرجاع بعض البيانات المخزنة بالذاكرة، يلزم إضافة بعض التجهيزات الإضافية للكشف عن بعض الأخطاء وتصحيح بعضها مما هو مكتشف. من أهم طرق تحسين وثوقية Reliability وحدة الذاكرة، هو استخدام أرمزة خاصة لكشف الخطأ في حال حدوثه وتصحيحه تلقائياً. ويمكن أيضاً اكتشاف بعض الأخطاء وتصحيحها باستخدام برامج خاصة، لكن هنا سنتم الإشارة فقط إلى بعض الإجراءات التقنية التي تستخدم أرمزة خاصة بالكشف عن الأخطاء وإصلاحها. ويضيف هذا النوع من الأرمزة بعض البتات الإضافية للمعطيات الأساسية بقصد تحديد مكان وقوع الخطأ، وبالتالي إصلاحه.

3-4-1 أرمزة كشف الخطأ

تمتاز أرمزة كشف الخطأ بتراكيب خاصة ليس المهم فيها أوزان أو طول الرماز وإنما حجم (عدد) الأخطاء التي يمكن كشفها. وهنا يجب أن نميز بين مفهومي:

- العطل Failure: ويعرف بأنه الظاهرة المعبرة عن إنهاء القدرة في تحقيق الوظيفة المطلوبة بحسب الشروط التقنية الموضوعة.

- الخطأ Error: ويقصد به عدم التوافق بين القيمة الظاهرة والقيمة الفعلية، وبالتالي يمكن الاستنتاج بأن الخطأ هو ناتج عطل ما، ولكن ليس بالضرورة أن يظهر كل عطل على شكل خطأ. وسنحاول فيما يلي التعرف على بعض أرمزة كشف الخطأ.

3-4-1-1 الندية Parity

لمعرفة الندية لكلمة ما يجب معرفة عدد الواحدات في الكلمة وبالتالي يوجد نوعين من الندية هما الندية الزوجية Even Parity إذا كان عدد الواحدات في الكلمة زوجي وندية فردية Odd Parity إذا كان عدد الواحدات في الكلمة فردي، حيث إن عملية إضافة بت الندية Parity bit تُعدّ من أكثر الطرائق شيوعاً لكشف الخطأ. فعند تمثيل أي محرف وفق رماز ASCII المعياري يلزم 7 bits، ويخصص البت الثامن للندية. تُعدّ طريقة كشف الخطأ باستخدام بت الندية بسيطة وتتمثل بإضافة بت إضافي لكل كلمة يراد مراقبتها. وقيمة هذا البت ستكون صفراً أو واحداً بحسب نوع الندية المراد استخدامها زوجية أو فردية مشكلة بذلك ما يعرف برماز كاشف الخطأ. في حال اعتماد الندية الفردية مثلاً، ينظر إلى عدد الواحدات في الكلمة، فإذا كان فردياً نضع بيت الندية صفر لحفاظ على العدد الفردي للواحدات في الكلمة، وإذا كان عدد الواحدات في الكلمة زوجي نضع بيت الندية بالقيمة واحد حتى يصبح العدد الاجمالي للواحدات في الكلمة فردياً. يضاف بت الندية، ويضمن بوصفه جزءاً من المعلومة المراد تخزينها في الذاكرة فمثلاً، عند إرسال رسالة من الشكل "Good bye" من حاسوب إلى آخر، قد يحدث في أثناء عملية النقل أخطاء تتمثل بتغير القيمة الثنائية المقابلة لبعض الأحرف المرسلة. ومن إحدى الطرائق لكشف مثل هذه الأخطاء إضافة بت ثامن للتركيبة السباعية المقابلة لكل حرف. إذا تم اعتماد الندية الفردية (علماً أنه لا يوجد فرق

في استخدام الندية الزوجية أو الفردية بالرغم من أن استخدام الندية الزوجية هو الغالب)،

وترمز الرسالة "Good bye" التي سترسل وفق الشكل التالي: Parity Message :

G ~ 1 100 0111

o ~ 1 110 1111

o ~ 1 110 1111

d ~ 0 110 0100

b ~ 0 110 0010

y ~ 1 110 1001

e ~ 0 110 0101

الرسالة بالصيغة الثنائية التي أصبح فيها كل حرف ممثلاً بـ 8bits وعدد الوحدات بالتركيبة المقابلة لكل حرف تضم عدداً فردياً من الوحدات، ترسل إلى وجهتها (أو تخزن في الذاكرة) متضمنة قيمة بت المراقبة. تفحص الندية في الطرف المستقبل للرسالة لكل حرف على حدة، فإذا لم تتطابق نتائج الفحص مع الترميز المستخدم بالإرسال تتم الإشارة إلى حدوث خطأ. بمعنى إذا لم تكن الندية فردية لحرف ما، يعني انه حدث خطأ في أثناء عملية النقل، أي إن بتاً واحداً على الأقل قد تغيرت قيمته أو ثلاث بتات أو أي عدد فردي من البتات. وعند كشف خطأ ما يطلب إعادة إرسال الرسالة من جديد، انطلاقاً من أن الخطأ قد يكون عشوائياً ولن يتكرر ظهوره مرة ثانية وفي حال استمرار وجود الخطأ يلجأ إلى الطرق التقنية للكشف عنه. أهم ساليب طريقة استخدام بت ندية واحد، هي أنها لا تكشف الخطأ في حال تغير عدد زوجي من البتات، لأن تغير عدد زوجي من البتات لا ينعكس على بت المراقبة، وستبدو الأمور طبيعية على الرغم من وجود أخطاء. وتعد هذه الطريقة البسيطة غير مكلفة، إلا أنها غير مجدية إذا كان المطلوب درجة عالية من الأمان في نقل المعلومات.

3-4-1-2 فاحص الاحتياط التكراري CRC

يستخدم الرمز CRC (Cyclical Redundancy Check) بشكل واسع في الكشف عن خطأ نقل بـ 1 أو 2 bit عندما تنقل المعطيات الرقمية عبر خطوط الاتصال بين حاسوبين في شبكة أو بين جهاز تخزين رقمي (مثل CD , DVD, or Hard Driver) والحاسوب الشخصي. ويضاف بالرمز CRC عدد من بتات المراقبة إلى نهاية بيتات المعطيات المراد إرسالها، وتفحص المعطيات من قبل المستقبل لكشف الأخطاء باستخدام CRC. بالرغم من

أن CRC يستطيع كشف الأخطاء المتعددة، لكن ليس كل خطأ يمكن حدوثه، يمكن كشفه بواسطة CRC، ومع ذلك يبقى هذا الرمز أفضل من غيره بكشف الأخطاء. غالباً ما يوصف CRC رياضياً كحاصل تقسيم كثيري حدود لتوليد باق، ويستخدم في ذلك التابع المنطقي XOR. ويمكننا باختصار تلخيص مسار تشكيل CRC بالنقاط التالية:

- 1- يتم انتقاء مولد رموز (شفرة) ثابت، الذي يمكن أن يحتوي على عدد أقل من البتات بالمقارنة مع بيتات البيانات ويجب أن يكون هذا الرمز نفسه للمرسل والمستقبل.
- 2- إلحاق عدد من الأصفار يساوي عدد بيتات مولد الرمز إلى نهاية بتات البيانات.
- 3- تقسيم بتات البيانات بما فيها البتات الملحقة على بتات مولد الرمز باستخدام XOR
- 4- إذا كان باقي القسمة صفراً، ترسل البيانات والبتات الملحقة كما هي. وإذا كان مغايراً للصفر يتم جعل البتات الملحقة مساوية لبتات الباقي بترتيب للحصول على الباقي الصفرى
- 5- على طرف المستقبل، يقسم المستقبل بتات البيانات الملحقة للرمز القادم على نفس مولد الرمز الذي استخدم من قبل المرسل.

6- إذا كان الباقي صفراً، لا يوجد هناك خطأ مكتشف. وإذا كان الباقي مغايراً للصفر، يعني ذلك أنه تم اكتشاف خطأ بالإرسال، وبهذه الحالة يطلب المستقبل إعادة الإرسال. نوضح فيما يلي هذه النقاط من خلال الأمثلة من دون الدخول بمزيد من التفاصيل. مثال: حدد CRC المرسل من أجل البايت D للبيانات ومولد الرمز G، وتحقق من أن الباقي هو صفر، علماً أن: $D = 1101\ 0011$ و $G = 1010$.

الجواب: بما أن مولد الرمز يملك بيانات بطول 4 bits، نضيف أربعة أصفار لبيانات البايت، فنحصل على $D' = 11010011\ 0000$ وفق التالي: وبعد ذلك، نقسم البيانات الجديدة (مع البيانات الملحقة) على بيانات مولد الرمز G باستخدام عملية modulo-2 أي $D'/G = 11010011\ 0000 / 1010$ وفق ما هو مبين بالجدول 3-6. حيث تتم عملية XOR على أول 4 bit من البيانات مع 4 bit لمولد الرمز، وإذا كان الناتج سيحتوي على صفر بأقصى اليسار (لتماثل البتات) تتم الإزاحة نحو اليمين في البيانات الملحقة بخانة واحدة (وقد تتم بأكثر من خانة)، ليتم بالمرحلة التالية تطبيق XOR على بيتات المولد

الأربعة مع أربعة بيتات من البيانات، وهكذا حتى الوصول إلى نهاية البيتات المستخدمة على أن يلغى أي بيت من اليسار تكون قيمته صفراً.

الجدول 3-6: آلية تقسيم مولد رماز 4 bit للحصول على D' وفق فاحص الاحتياط CRC

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

الباقي النهائي يساوي 0100 . وبما أنه مغاير للصفر، يتم إلحاق هذا الباقي بالبيانات الأساسية لتصبح البيانات الجديدة من الشكل: 11010011 0100 . وتتم من جديد عملية القسمة على 1010 وبفس الآلية حتى ينتج أن Remainder = 0 وفق الجدول أدناه

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

مثال آخر: بفرض أن خطأ حدث خلال إرسال البيانات المولدة بالمثال السابق، وتحديدًا في البيت الثاني من اليسار من بايت البيانات الملحقة، حيث البيانات المستقبلية هي من

الشكل $D' = 100100110100$ ، يطلب تطبيق المسار CRC على البيانات المستقبلية لكشف الخطأ باستخدام نفس مولد الرماز 1010.

الجواب: بشكل مماثل للمثال السابق نجري عملية القسمة والإزاحة وفق الجدول أدناه

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

يتضح أن الباقي هو 0100 مغاير للصفر ، وبالتالي سيشار إلى وجود خطأ، وهكذا الحال بالنسبة لأي مثال مماثل، حيث ان استخدام هذا الرماز بالمقارنة مع رماز ال Parity هو الانسب في حال ان المعلومات المراد فحصها أكثر أهمية.

3-4-2 أرمزة تصحيح الخطأ

يولد الرماز المصحح للخطأ Error-Correction Code مجموعة بتات ندية تخزن في الذاكرة مع المعطيات الفعلية. يكون كل بت ندية بت اختبار Check bit لمجموعة محددة من بتات كلمة المعطيات. وعند إعادة قراءة الكلمة من الذاكرة، نقرأ أيضاً بتات الندية، ونقارن مع مجموعة بتات اختبار تولد من المعطيات المقروءة. فإذا كانت بتات الاختبار صحيحة، فهذا دليل على عدم حدوث خطأ. وإذا اختلفت بتات الاختبار عن بتات الندية المخزنة، فإنها تولد نمطاً متميزاً Unique pattern يسمى العرض Syndrome يمكن استخدامه لمعرفة البت الذي وقع فيه الخطأ. ومن أكثر أرمزة هذه المجموعة استخداماً رماز هامينغ Hamming code الذي يعتمد بكشف الأخطاء أو إصلاحها على بعد هامينغ. نبيّن فيما يلي مفهوم بعد هامينغ المرتبط بالتمثيل الهندسي للمعطيات الثنائية.

التمثيل الهندسي للمتحويلات الثنائية: يمكن تمثيل المتحويلات المنطقية بشكل هندسي، بحيث كل متحول A يمثل بنقطتين على نهايتي مستقيم (بطول وحدة الطول، ولتكن 1 cm)، كما هو مبين في الشكل 2-3، ويمكن تمثيل متحولين منطقيين مثل A, B بمربع كل رأس من رؤوسه الأربعة يقابل واحدة من التراكيب الأربعة الممكنة: 00, 01, 10, 11 للمتحولين AB، كما هو موضح في الشكل ب 2-3. وهكذا لتمثيل ثلاثة متحويلات منطقية A, B, C يلزم شكل فراغي ب $2^3 = 8$ رأس وهو ما يقابل مكعباً، كما هو مبين في الشكل ج 3-4.



الشكل 2-3: التمثيل الهندسي للمتحويلات المنطقية

يلاحظ من الشكل ج 2-3 أن كل رأسين متجاورين بالمكعب يقابلهما رمزان يختلفان ببِت واحد، مثل الرأسين 100 و 101. يوجد لهذه الطريقة في تمثيل المتحويلات المنطقية تطبيقات عديدة، لكن من أهم سيئاتها هو صعوبة تخيل التمثيل الهندسي لتابع بأربعة متحويلات وأكثر، حيث لتمثيل تابع ب n متحول يلزم فراغ منتظم ب n بعد. والشكل الناتج عن تمثيل المتحويلات المنطقية بهذه الطريقة، يسمى بمكعب بول ب 2^n رأس. تشكل الرؤوس الأولية لمكعبات بول ما يعرف بالمكعب 0، لأن كلاً منها يمثل نقطة (بعدها صفر). فمثلاً، بالنسبة لمكعب بول ب $2^3 = 8$ رأس يقال إن لهذا المكعب ثلاثة أبعاد وثمانية رؤوس أولية تقابل الحدود الصغرى. وتشكل كل من الرؤوس 000, 001, ..., 111 ما يعرف بالمكعب 0. يمكن من كل مكعب 0 أن يشكل مكعباً 1، وهو يمثل خطأ. وبالطبع فإن طرفي الخط الواحد متجاوران ويمثلان رمزين يختلفان عن بعضهما ببِت واحد. يرمز لهذا البت المغاير بالرمز x. فمثلاً: الرمز 010 يمثل مكعباً من الدرجة 0 والرمز 000 يمثل مكعباً من الدرجة 0.

يمكن من هذين المكعبين من الدرجة 0 تشكيل مكعب من الدرجة 1 يرمز له بالمكعب 0×0 ، حيث يوضع الرمز x بموضع البت الذي يقابل الاختلاف في المكعبين 0. يمكن بطريقة مماثلة لما سبق من مجموعة مكعبات 1 تشكيل مكعبات 2 التي تمثل سطح. ويمكن من المكعبين 1 بالتركيبة 0×0 و 0×1 مثلاً، تشكيل المكعب $0 \times x$ وهو مكعب 2، لأنه يحتوي على $2x$ في المواضع التي يوجد فيها اختلاف بقيم البتات بين الكلمتين. يمثل عدد الاختلافات بين رمزين ما يعرف بالبعد بين الرمزتين. فمثلاً، البعد بين 000 و 100 هو واحد فقط، لأن عدد التغيرات في البتات هو بموقع واحد. وبمعنى آخر، يقال ان البعد واحد، لأنه يمكن الوصول من 000 إلى 100 بخطوة واحدة. وهكذا البعد بين 000 و 111 هو 3، لأنه كي يتم الوصول من 000 إلى 111 على رؤوس المكعب، يجب أن تسلك الطريق من 000 إلى 001 إلى 011 إلى 111 أو من 000 إلى 010 إلى 011 إلى 111 وفي كلتا الحالتين يلزم ثلاثة خطوات. وبالتالي البعد هو 3. يسمى البعد d بين كلمتين من رماز ثنائي، بعد هامينغ، ويعرف بأنه عدد البتات التي يحدث فيها تغيير ويمكن تحديد هذا البعد باستخدام XOR.

3-4-2-1 رماز هامينغ Hamming code

يعتمد الرماز الذي وضعه هامينغ لتصحيح الأخطاء في ذواكر النفاذ العشوائي RAM على إضافة عدد من بتات المراقبة للكلمة المراد مراقبتها بقصد الكشف عن الخطأ وإصلاحه في حال حدوثه. وانطلاقاً من عدد بتات المعلومات المراد مراقبتها، استطاع هامينغ بطريقة رياضية تحديد عدد بتات المراقبة التي يمكن إضافتها بالعلاقة:

$$2^i \geq m + i + 1 \text{ حيث: } m - \text{تمثل المعلومات المفيدة}$$

$$I - \text{عدد بتات المراقبة } P_i \text{ أو بتات الندية المطلوب إضافتها}$$

وهكذا من أجل معطيات مفيدة بطول m bits أضاف هامينغ i bits للمراقبة ليصبح الطول الجديد للمعطيات بطول $m + i$ bits. تخصص المواقع ذات الأرقام التي تمثل قوى الرقم (2) لبتات الندية وباقي المواقع لبتات المعطيات الفعلية، حيث يمكن استخدام رماز هامينغ مع كلمات بأطوال مختلفة. بفرض أن عدد البتات التي تحوي على المعلومات في الكلمة

$m = 7$ فإنه وفق المتراجحة أعلاه، يلزم إضافة أربعة بتات مراقبة P_1, P_2, P_3, P_4 وبالتالي يتضح من المتراجحة أعلاه أيضاً أن عملية الاحتياط Redundancy تكون مجدبة كلما كان طول الكلمة أكبر، لأنه من أجل كلمة بطول 1 bit يلزم 2 bits للمراقبة، بينما كلمة بطول 11 bits يكفيها 4 bits للمراقبة. ويجب أن يكون للرماز القادر على إصلاح خطأ واحد بعد هامينغ أدنى $d_{\min}$ بطول 3. ومن أجل n خطأ قابل للإصلاح يجب أن يكون بعد هامينغ الأصغري $d_{\min} = 2n + 1$. ويمكن اظهار كيفية حساب بعد هامينغ من خلال مثال محدد، وليكن إيجاد بعد هامينغ بين الكلمتين 10011 و 00101. الجواب: $10011 \oplus 00101 = 10110$. يتضح من ناتج العملية وجود ثلاث واحدات بالناتج وبالتالي $d_H = 3$. وهذا هو الأسلوب المتبع في حساب البعد d بين كلمتين. يعرف رماز هامينغ بشكله العادي على أنه رماز يمتاز ببعد أصغري $d_{\min} = 3$. كلما زاد عدد بتات المراقبة، سيزداد حجم المعلومات غير المفيدة وهذا يتطلب إضافة تجهيزات. هذه هي من أهم سلبيات هذا الرماز. يظهر الجدول 3-7 عدد الأخطاء التي يمكن كشفها أو إصلاحها بواسطة رماز هامينغ بحسب بعد هامينغ.

الجدول 3-7: عدد الأخطاء التي يمكن كشفها وإصلاحها باستخدام رماز هامينغ

عدد الأخطاء التي يمكن كشفها	عدد الأخطاء التي يمكن إصلاحها	بعد هامينغ الأدنى $d_{\min}$
1	1	3
2	2	4
3	3	5
4	4	6

يتضح من الجدول أن رماز هامينغ العادي بالبعد $d_{\min} = 3$ قادر على كشف خطأ واحد وإصلاح خطأ واحد أو كشف خطأين من دون إصلاح أي منهما. ويمكن إظهار آلية كشف

الأخطاء باستخدام رماز هامينغ وإصلاحها من خلال مثال محدد، وفق ما هو مبين أدناه.
مثال: بفرض إن طول الكلمة المراد مراقبتها باستخدام رماز هامينغ 4bits، أي $m = 4$ ،
فهذا يعني انه يلزم وفق هامينغ ثلاث بتات مراقبة P_1, P_2, P_3 .

والسؤال: ما قيم هذه البتات وأين أماكن توزيعها؟

الجواب: حدد هامينغ مواقع هذه البتات ضمن تركيبة الرماز الأساسية بالمواقع التي تقابل القيمة 2^i من أجل $i = 0, 1, \dots$ أي المواقع التي تقابل الأوزان $2^0 = 1, 2^1 = 2, \dots$ وبالتالي عند معالجة كلمة بنظام يستخدم رماز هامينغ يقوم النظام فوراً بشكل آلي بإضافة بتات المراقبة إلى الكلمة الأساسية ليصبح الرماز بتركيبته الجديدة وفق الجدول 3-8، انطلاقاً من أن الترتيب العام للكلمة المعطاة إن كانت بطول 4 bits هو من الشكل: $B_4B_3B_2B_1$ (بحيث LSB تقابل B_1). ولتوضيح عملية حساب قيم بتات المراقبة، ننطلق من كلمة محددة ولتكن من الشكل (1011) التي ترتب على النحو $B_4B_3B_2B_1 = 1011$ ويكون حساب قيم بتات المراقبة P_1, P_2, P_3 اللازمة وفق هامينغ على النحو التالي:

الجدول 3-8: توزيع بتات المراقبة وفق كود هامينغ على كلمة بطول 4 bits

No. of bits	P_1	P_2	B_4	P_3	B_3	B_2	B_1
Bits position	1	2	3	4	5	6	7
	001	010	011	100	101	110	111

- البت p_1 الذي يقع بالموقع 001 يراقب كافة البتات التي فيها مقابل الواحد الموجود بالموقع الأول من P_1 قيمة الواحد. أي أنه يراقب كافة البتات الموجودة في المواقع التالية: 011, 101, 111 والتي تقابل المواضع 3, 5, 7 وهي البتات B_4, B_3, B_1 التي في كل منها قيمة واحد بالخانة الأولى. وانطلاقاً من الكلمة المعطاة، تحسب قيمة هذه البتات الثلاثة كالتالي: $B_4 B_3 B_1 = 101$. فإذا تم اعتماد بت الندية الزوجية، يجب أن تكون قيمة p_1 مساوية للصفر حتى يبقى عدد الوحدات بالكلمة زوجي. وهذا ما يعبر عنه رياضياً بالشكل التالي: $p_1 = B_4 \oplus B_3 \oplus B_1 = 0$

- البت P_2 بالموضع 010 يراقب كل البتات الموجودة بالأمكنة 3, 6, 7، وهي تقابل B_4, B_2, B_1 . وبما أن $B_4B_2B_1 = 111$ ، يجب أن تكون قيمة P_2 بعد مراعاة الندية

$$P_2 = B_4 \oplus B_2 \oplus B_1 = 1 \quad \text{أي أن:}$$

- البت P_3 بالموضع 100 يراقب كل البتات الموجودة بالأمكنة 5, 6, 7، وهي تقابل البتات B_3, B_2, B_1 وبما أن $B_3B_2B_1 = 011$ ، يجب أن تكون $P_3 = 0$ حتى تحقق الندية

الزوجية، أي أن: $P_3 = B_3 \oplus B_2 \oplus B_1 = 0$. ونلخص ما سبق بالجدول التالي:

Bits	P_1	P_2	B_4	P_3	B_3	B_2	B_1
Position of Bits	1	2	3	4	5	6	7
information	001	010	011	100	101	110	111
Parity bits	-	-	1	-	0	1	1
Final Code	0	1	-	0	-	-	-
	0	1	1	0	0	1	1

أي من أجل كلمة بطول 4bits يتم الحصول على رماز هامينغ مقابل بطول 7bits. تختبر البتات السبع المخزنة في الذاكرة عند القراءة للكشف عن حدوث خطأ. تحسب قيم بتات الاختبار بوجود بت الندية كالتالي:

$$C_1 = \text{XOR of bits (1, 3, 5, 7)}$$

$$C_2 = \text{XOR of bits (2, 3, 6, 7)}$$

$$C_3 = \text{XOR of bits (4, 5, 6, 7)}$$

انطلاقاً من أن عدد بتات الاختبار i bit (قيمة العرض Syndrome Value C) بعدد بتات المراقبة، أي 3 bits بهذا المثال مرتبة وفق الشكل $C = C_3C_2C_1$ ومجالها 2^3 ، تكون قيمها من 0 حتى $2^3 - 1$ بحيث تدل إحدى هذه القيم (التي تساوي $C = 000$) على الحالة التي لا يكشف فيها خطأ، وبقيّة القيم $2^3 - 1$ (عندما $C \neq 0$) تدل على موقع بت الخطأ من بين مجموع البتات $m + i$. بمعنى آخر، ترتب بتات الاختبار الثلاثة المضافة بالشكل $C_3C_2C_1$ لتأخذ سبع قيم من 001 حتى 111، بحيث كل تركيبة تقابل رقم أحد بتات الرماز الإجمالية، ولما كانت البتات قد خزنت مع ندية زوجية، فإن النتيجة $C = C_3C_2C_1 = 000$ تدل على عدم حدوث خطأ، وأن النتيجة $C_3C_2C_1 = 101$ مثلاً تدل على حدوث خطأ

بالبت الخامس. إذا حصل التعرف على البت الخطأ يمكن تصحيح الخطأ بإتمام قيمة البت. فمن أجل نفس المثال المبين أعلاه حول تخزين الكلمة (1011) بذاكرة تستخدم رماز هامينغ باعتماد الندية الزوجية حصلنا على الشكل (0110011). ويفرض عند قراءة نفس الكلمة المخزنة تمت قراءة كلمة خاطئة من الشكل 0110001 (التي تقابل كلمة أخرى)، نظراً لتغير قيمة أحد البتات من الواحد إلى الصفر في أثناء عملية القراءة (البت السادس من اليسار تغيرت قيمته) ولنظهر آلية كشف الخطأ وإصلاحه. يجب أن يتحقق هذا الرماز من نفسه أوتوماتيكياً، بحيث عند عملية القراءة يتم حساب قيم بتات الاختبار كالتالي:

1- البت الأول (C_1): انطلاقاً من الشكل العام للكلمة المستلمة عند القراءة

$$P_1 P_2 B_4 P_3 B_3 B_2 B_1 = 0 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1$$

المواقع 1, 3, 5, 7 وتقابل البتات: P_1, B_4, B_3, B_1 . ولكن المجموعة P_1, B_4, B_3, B_1

بالكلمة المستلمة تقابل التركيبية $p_1 \ 1 \ 0 \ 1$. ولكي يكون عدد الوحدات في هذه المجموعة

زوجياً (بسبب اعتماد الندية الزوجية)، يجب أن تكون قيمة $P_1 = 0$. وبالنظر إلى الكلمة

المستلمة، يلاحظ أن P_1 تساوي 0 فعلاً. إذا نتيجة الاختبار صحيحة ويمكن حساب قيمة

$$C_1 \text{ من العلاقة: } C_1 = P_1 \oplus B_4 \oplus B_3 \oplus B_1 = 0$$

2- البت الثاني (C_2): بنفس الآلية السابقة، إن هذا البت يراقب البتات الموجودة بالمواقع

2, 3, 6, 7 وهي تقابل البتات P_2, B_4, B_2, B_1 التي تقابل بدورها من الكلمة المستلمة

القيم $p_2 \ 1 \ 0 \ 1$. ولكي يكون عدد الوحدات في هذه المجموعة زوجياً يجب أن تكون قيمة

$P_2 = 0$. وبالنظر إلى الكلمة المستلمة يلاحظ أن P_2 تساوي الواحد، وهذا يعني أن

$$\text{الاختبار خاطئ وتحسب قيمة } C_2 \text{ من العلاقة } C_2 = P_2 \oplus B_4 \oplus B_2 \oplus B_1 = 1$$

3- البت الثالث (C_3): يراقب البتات الموجودة في المواقع 4, 5, 6, 7 وهي تقابل

P_3, B_3, B_2, B_1 التي بدورها تقابل من الكلمة المستلمة القيم $0 \ 0 \ 0 \ 1$ ، ولكي يكون

عدد الوحدات في هذه المجموعة زوجياً، يجب أن تكون قيمة $P_3 = 1$ ، لأن العدد

الإجمالي للوحدات بالتركيبية أعلاه هو فردي. وبالنظر إلى الكلمة المستلمة يلاحظ أن P_3

تساوي الصفر وهذا يعني أن الاختبار خاطئ، وتحسب قيمة C_3 من العلاقة:

$$C_3 = P_3 \oplus B_3 \oplus B_2 \oplus B_1 = 1$$

أعلاه بالترتيب $C_3C_2C_1$ يلاحظ أنها تقابل التركيبة $(6)_{10} = 110$ التي تعادل رقم البت الذي حدث فيه الخطأ، مما يعني أن قيمة البت السادس مغلوطة. يتم تغير قيمة هذا البت من الشكل الذي تم فيه الاستلام (أي 0) إلى القيمة المعاكسة (المتمم)، ويصحح الرماز المستلم إلى الشكل 0110011 وهو الشكل الفعلي للكلمة المرسله. وهناك رماز هامينغ الموسع الذي يمتاز ببعد هامينغ $d_{\min} = 4$. يمكن بمثل هذا الرماز الجديد تصحيح خطأ وحيد والكشف عن خطأين. فمثلاً، من أجل كلمة معطيات بطول 8 bits من الشكل: (1100 0100) يلزم وفق رماز هامينغ العادي أربعة بتات ندية، ترتب وفق الشكل التالي: $P_1 P_2 1 P_3 1 0 0 P_4 0 1 0 0$ وباعتماد الندية الزوجية يمكن حساب قيم p_1, p_2, p_3, p_4 والحصول على الشكل الجديد للكلمة المعطاة وفق رماز هامينغ من الشكل 001110010100. لا يستطيع هذا الرماز كشف الأخطاء المتعددة. لكن بإضافة بت آخر إلى الكلمة، أي استخدام رماز هامينغ الموسع ببعد $d_{\min} = 4$ يمكن تصحيح خطأ وحيد والكشف عن خطأين.

3-5 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

- 1- يمثل كل رقم عشري بالرماز BCD بـ
a- unit b- bit c- byte d- nibble
- 2- العدد BCD المقابل للعدد العشري 473 هو
a- (111011010) b- (10001110011)
c- (110001110011) d- None of all
- 3- ناتج طرح العددين 10-15 بالـ BCD code هو
a- 0101 b- 1010 c- 1011 d- none of all
- 4- تقابل القيمة الثنائية $(11100\ 0111\ 0000)_2$ بالرماز BCD العدد
a- 3670 b- 11270 c- 7360 d- none of all
- 5- الترميز المقابل للرقم 8 وفق ترميز Excess-3 هو
a- 1100 b- 1011 c- 0101 d- 0110
- 6- أي من الكلمات التالية متجاورة
a- (1110 & 1010) b- (1000 & 1001)
c- (1011 & 1111) d- all the above
- 7- قيمة الفرق بين الترميز بالرماز ASCII المقابل لكل من الحرفين "a" و "A"
بالنظام الستة عشري هو
a- 10 b- 16 c- 20 d- None of all
- 8- يمثل العدد العشري (12) بالرماز الموزون 1-2-4-8 بالشكل:
a- (1110100) b- (1000100100)
c- (1 10) d- (10010)
- 9- يمثل الرقم 4 بالرماز الموزون 1 2 4 2 القيمة:
a- 0100 b- 1010 c- 1110 d- None of all
- 10- أي من الشفرات التالية موزون
a- Eccess-3 b- Hamming code
c- Gray code d- none of all

11- أي من التقنيات التالية تصنف ضمن فئة طرق الضغط دون فقدان للمعلومات المضغوطة التي تؤمن إعادة المعلومات إلى صورتها الأصلية بالضبط

- a- Lempel Ziv b- MP3 c- JPEG d- None of all

12- كود هامينغ بالبعد الاصغري $d_{min} = 4$ قادر على

- a- تصحيح خطأين وكشف خطأين بدون اصلاح
b- تصحيح خطأين وكشف خطأ واحد
c- تصحيح خطأ واحد وكشف خطأين دون اصلاح أي منهما
d- تصحيح خطأ واحد وكشف خطأ واحد بدون اصلاح

ثانياً: أسئلة متفرقة

- 1- مثل العدد العشري 803 بكل من الرامزين 84-2-1 a) Excess-3
2- مثل الأرقام العشرية من 0 حتى 9 برماز موزّن وفق الأوزان 6 3 1 1 .
3- حول كلاً من الأعداد العشرية التالية إلى ما يقابلها بالرماز BCD
a) 57 b) 6249
4- أوجد ناتج مجموع العددين العشريين $708 + 695$ باستخدام الرماز BCD، مبيناً مراحل الوصول للناتج الصحيح.
5- أوجد شكل رماز هامينغ لمراقبة كلمة من الشكل: 11000011101 قادر على كشف خطأ واحد وإصلاح خطأ واحد عند تخزين هذه التركيبة في ذاكرة RAM.

الفصل الرابع: الجبر البوليني والتوابع المنطقية

Boolean Algebra and Logic Functions

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

1-4 جبر بول

ظهر جبر بول (الجبر البوليني Boolean Algebra) لأول مرة عام 1854م، إلا أنه لم يطبق أو يستخدم قبل العام 1934 وقد سمي بهذا الاسم نسبة إلى العالم جورج بول. يعرف الجبر البوليني على أنه المجموعة B التي تحتوي المتحولات المنطقية والثوابت المنطقية (0 و 1) والعمليات المنطقية الأساسية NOT و AND و OR . يغطي جبر بول على الرغم من المحدودية التي يبدو بها بالمقارنة مع الجبر العادي (بسبب محدودية العمليات المعرفة فيه والقيم المسموح التعامل بها) بعض جوانب الضعف للجبر العادي المتعلقة بجوانب الفكر . لا يستطيع الجبر العادي رغم فوائده في العديد من العمليات الذهنية التعامل مع جميع جوانب الفكر . والمشكلة الأساسية في ذلك هي مشكلة اللغة. تبرز أهمية النظام 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 أداة لتبسيط التوابع المنطقية.

1-1-4 قوانين بول

يوجد لدى جبر بول مجموعة من القوانين النازمة التي تستخدم في أثناء كتابة أي عبارة منطقية واختصار العبارات المركبة منها. تأتي أهمية هذه القوانين أيضاً من أن 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 إن العبارة صحيحة أو محققة إذا كانت قيمتها واحداً. أما إذا كانت عكس ذلك فيكون لها القيمة صفر. وسيتم التعامل مع الصفر والواحد المستخدمان بجبر بول باعتبارهما ثوابت منطقية صح أو

خطأ (True or False) وليس باعتبارهما أرقاماً ثنائية رغم إنهما ينتميان للنظام الثنائي. يمكن من أجل عبارة منطقية بمتحول منطقي واحد A صياغة ثمانية قوانين منطقية مختلفة، وفق ما هو مبين في الشكل 1-4 وتمثيل هذه العمليات على شكل دارات كهربائية باستخدام المفاتيح التي يمر كل منها بحالتين، مثله مثل المتحول المنطقي. فإذا انطلقنا من أن العملية المنطقية AND التي تعني "و" تقابل في الدارات الكهربائية الوصل التسلسلي وأن العملية المنطقية OR التي تعني "أو" تقابل الوصل التفرعي، وأن المفتاح الذي يكون موصولاً سيكون متممه مفتوحاً، يمكن بوساطة الدارات الكهربائية التعبير عن العبارات المنطقية، حيث أن كل متحول منطقي يمثل مفتاحاً كهربائياً يعبر عنه بحالة مفتوح، ويقابل القيمة "خطأ"، ولا يمرر التيار. وتعطي لمبة الإشارة الموجودة على خرج الدارة الكهربائية النتيجة النهائية (غير مضاءة تقابل 0 ومضاءة تقابل 1).

الشكل 1-4: الرمز الكهربائي لكل من قوانين بول لعبارة منطقية بمتحول واحد

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

1 - قوانين التبديل (Commutative Laws) : التي تعني $\forall a, b \in B$ فإن

$$a + b = b + a$$

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

2 - قوانين التجميع (Associative Laws) : التي تعني $\forall a, b, c \in B$ فإن

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

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

3 - قوانين التوزيع (Distributive Laws) : التي تعني $\forall a, b, c \in B$ فإن

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

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

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

4- قوانين الدمج (التبسيط):

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

5- قوانين الامتصاص (Absorption Laws) : ولها شكلان، هما

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

الأول - مباشر ويقتضي:

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

الثاني - عكسي ويقتضي:

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

6 - قوانين ديمورغان (De Morgan's Laws) وهي ثلاثة: $(a + b)' = a' \cdot b'$;

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

$$a'' = a$$

يمكن تعميم بعض القوانين المبينة أعلاه مثل قوانين ديمورغان الأول والثاني على أكثر من

$$(a + b + c + \dots)' = a' \cdot b' \cdot c' \cdot \dots \quad \text{حيث:}$$

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

يمكن استخدام هذه القوانين بالعديد من التطبيقات مثل اختصار العبارات المنطقية وإثبات

صحة بعض القوانين أو إيجاد متمم عبارة منطقية وغيرها.

2-1-4 اختصار العبارات المنطقية

تسمى الطريقة التي تستخدم بها قوانين بول في تبسيط عبارة منطقية مركبة إلى عبارة أبسط بالطريقة الجبرية. فمثلاً يمكن اختصار تابع منطقي بثلاثة متحولات من الشكل:

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

إلى الشكل التالي:

$$\begin{aligned} F(a, b, c) &= a'b'c' + a'b'c + a'bc + ab'c' + ab'c + abc = \\ &= a'b'(c' + c) + bc(a' + a) + ab'(c' + c) = \\ &= a'b' + bc + ab' = b'(a' + a) + bc = b' + bc \end{aligned}$$

3-1-4 إثبات صحة العبارات المنطقية

يمكن إثبات صحة أي عبارة منطقية (بما في ذلك بعض قوانين بول) بتحويل أحد طرفي العبارة المنطقية إلى الشكل المماثل للطرف الآخر أو باستخدام إحدى الطرق التالية:

أولاً- الطريقة الجبرية (أو طريقة المفاتيح الكهربية): ونوضحها من خلال الأمثلة.

مثال: برهن صحة قانون التوزيع: $(a + b)(c + b) = a.c + b$

$$\begin{aligned} (a + b)(c + b) &= a.c + a.b + b.c + b.b = \\ &= a.c + b.(a + c + 1) = a.c + b \end{aligned}$$

- باستخدام المفاتيح الكهربية: يمكن تمثيل قانون التوزيع بمخطط، وفق الشكل 2-4.

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

الشكل 2-4: طريقة المفاتيح الكهربية بإثبات صحة قانون التوزيع

يتضح من الشكل أعلاه، أن 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، ما يدل على أن الشكليين ينجزان نفس التابع.

ثانياً- طريقة استخدام جدول الحقيقة: بداية يعرف جدول الحقيقة Truth table بأنه الجدول الذي يضم جميع التوافقيات الممكنة لمداخل دارة منطقية، وما يقابل كل منها على خرج الدارة، بمعنى أنه الجدول الذي يعكس علاقة خرج دارة ما بمدخلاتها. وبالتالي يتم وفق جدول الحقيقة وضع المتحولات أو الحدود لكل من طرفي العبارة (أي جميع التراكيب الممكنة)، وبعد ذلك تتم المقارنة بين طرفي المعادلة من خلال المقارنة بجدول الحقيقة لجميع الحالات الممكنة للمتغيرات التي تعتمد عليها المعادلة.

مثال: أثبت باستخدام طريقة جدول الحقيقة صحة قانون الامتصاص $a'b' + b = a' + b$
 الجواب: بأخذ الحالات الممكنة لمختلف المتحولات والحدود في العلاقة المعطاة بطرفيها الأيمن والأيسر ووضعها بجدول للحقيقة كما هو مبين في الجدول 4-1.

الجدول 4-1: جدول الحقيقة بتراكيبه المختلفة لعبارة منطقية

Index	a	b	a'	b'	a'b'	a'b' + b	a' + b
0	0	0	1	1	1	1	1
1	0	1	1	0	0	1	1
2	1	0	0	1	0	0	0
3	1	1	0	0	0	1	1

يلاحظ من آخر عامودين أن طرفي العبارة المعطاة متساويان فعلاً من أجل جميع القيم الممكنة للمتحويلات a, b إذا العبارة محققة.

ثالثاً- طريقة إجراء عملية ما على طرفي المعادلة شرط ان يكون مسموحاً عكسها. فمثلاً من العمليات الممكن إجراؤها على طرفي المعادلة هي أخذ المتمم لكلا الطرفين، في حين أنه يوجد عمليات غير مسموح بها في المنطق، مثل ضرب طرفي العبارة بعبارة أخرى، لأن عملية القسمة المعاكسة للضرب غير مسموح بها في الجبر البوليني. 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 على الرغم من أنه لو فرضت أن قيمة $y = 1$ ، فإن المعادلة تصبح محققة، لأن: $1 + 1 = 1 + 0$.

رابعاً- طريقة استخدام قانون الإضافة

من قوانين جبر بول المستخدمة في تبسيط العبارات المنطقية، يوجد قانون الحد الفائض. فمثلاً، العبارة المنطقية $ab + bc + a'b'c$ مساوية للعبارة $ab + a'b'c$. أي وكأن الحد bc في الطرف الثاني للعبارة لا دور له، أي حد فائض. ولإثبات صحة ذلك، يمكن الانطلاق من الطرف $ab + a'b'c$ ، فنجد إن:

$$\begin{aligned} ab + a'b'c &= ab(c + c') + a'b'c(b + b') = abc + abc' + a'bc + a'b'c = \\ &= bc(a + a') + abc' + a'b'c = bc + abc' + a'b'c = \\ &= b(c + ac') + a'b'c = b(a + c) + a'b'c = ab + bc + a'b'c = \\ &= ab + c(b + a'b') = ab + c(a' + b) \\ &= ab + a'b'c + bc \end{aligned}$$

لإيجاد الحدود الفائضة في عبارة منطقية يتم بداية البحث عن المتحول الموجود في حدين مختلفين من العبارة بشرط أن يكون المتحول في أحد هذين الحدين بصورته العادية وفي الحد الثاني بصورة المتمم، مثل الحد bc في المثال أعلاه. إذ إن 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 $a'b'c$ مع المتحول a' . وعند 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 إن لم يكن موجوداً بقصد الاختصار. ويلاحظ في العبارة $(a + b)(a' + c)(b + c)$ وجود حد فائض $(b + c)$ ، وهو الشكل الثاني للقانون بصورة جداء مجاميع.

4-2 التابع المنطقي

يُعرف التابع المنطقي بأنه سلسلة من العمليات المنطقية على مجموعة من المتحولات المنطقية ويعبر عنه بشكل رياضي على أنه تمثيل لـ n حالة لمتحولات منطقية على الدخول غير مرتبطة $x_1, x_2, \dots, x_n$ على m حالة لمتحولات مرتبطة $y_1, y_2, \dots, y_m$ على الخرج، بمعنى أن: $y_j = f(x_i)$ من أجل $i = 1, 2, \dots, n$ و $j = 1, 2, \dots, m$.

من أكثر الطرق المستخدمة لتمثيل التابع المنطقي طريقة استخدام جدول الحقيقة. فمن أجل تابع ب n متحول، سيكون جدول الحقيقة المقابل له مكوناً من 2^n سطراً. يتعامل التابع المنطقي 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 $\cap$ أو $(.)$ ، ويعني "و" AND.
- النفي المنطقي ويرمز له بخط فوق المتحول $(-)$ أو بفاصلة فوق المتحول $(\neg)$ حيث نفي x هو x' ، ويعني "لا" NOT.

تشبه العمليات الأساسية بالمنطق الرقمي بعض العمليات بالحساب الثنائي، حيث إن العمليتين AND و OR تشبهان عمليتي الضرب والجمع، وعلى الرغم من ذلك يجب التفريق بينهما، لأن المتحول الحسابي يمكن أن يدل على عدد مؤلف من عدة أرقام في حين أن 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 10_2 ، في حين يكتب الناتج وفق المنطق بالشكل (1)، ويقراً "صح". يمكن من أجل متحولين منطقيين A و B تشكيل $2^2 = 4$ من التراكيب المختلفة. يظهر الجدول 4-2 هذه التراكيب والعمليات المنطقية الثلاثة الأساسية على متحولين منطقيين A, B.

الجدول 4-2: التوابع المنطقية الأساسية

المتحول A	المتحول B	OR الجمع A + B	AND الضرب A.B	NOT النفي A' أي $\neg A$
0	0	0	0	1
1	0	1	0	0
0	1	1	0	1
1	1	1	1	0

يتم ترتيب تنفيذ العمليات المنطقية عند ورودها بعبارة منطقية وفق أولويات Priority محددة هي NOT أولاً و AND ثانياً و OR ثالثاً. وفي حال ورود أكثر من عملية منطقية في

العبرة الواحدة بنفس الأولوية وراء بعضها تكون الأفضلية المباشرة بتنفيذ العمليات ابتداء من اليسار باتجاه اليمين. وفي حال وجود أقواس فإن الأفضلية أو الأسبقية Precedence ، تكون بتنفيذ ما هو ضمن الأقواس أولاً بالترتيب المشار إليه أعلاه. يوصف التابع المنطقي بجدول الحقيقة الذي يعطي العلاقة ما بين متحولات الدخل والخرج. ويعكس ما هو المطلوب من المسألة المراد إيجاد دارة لتجيزها. فمثلاً إذا طلب باستخدام تابع منطقي تحديد الأعداد الزوجية فقط من مجموعة الأعداد من 0 حتى 7 يكون جدول الحقيقة المقابل لمثل هذا التابع مبين بالجدول 3-4 وهو تابع منطقي بثلاثة متحولات دخل ومتحول خرج واحد. يلزم مثل هذا التابع ثلاث متحولات دخل لأن أعلى قيمة على الدخل المتمثلة بالرقم 7 يلزم لتمثيلها ثنائياً ثلاث خانات لأن $(111)_2 = (7)_{10}$ أي يلزم لكل خانة متحول، ولتكن هذه المتحولات المنطقية هي a, b, c. كما يلزم متحول واحد على الخرج، وليكن y، بحيث إذا كان $y = 0$ (أي خطأ False) التابع غير محقق والرقم المقابل ليس زوجياً، أما إذا كان $y = 1$ (أي صح True) يكون التابع محقق والرقم المقابل زوجياً.

الجدول 3-4: جدول الحقيقة لتابع منطقي بثلاث متحولات

المدخل 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

أثناء كتابة التابع المنطقي يجب أن نميز بين المتحول الأدنى أهمية LSB أي المتحول المقابل للخانة الأولى من اليمين وليكن c والمتحول المقابل للخانة ذات الأهمية الأكبر

MSB بأقصى اليسار وليكن المتحول a . يظهر من الجدول أعلاه أن المتحول الأدنى c 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 جميع التراكيب في الجدول بشكل يتطابق مع الترتيب التصاعدي للترميز الثنائي للأرقام العشرية). تقابل كل تركيبة للمتحولات على الدخل سطرًا واحدًا من الجدول، ويدعى الرقم العشري المقابل بدليل الحالة S (Index State) ويأخذ قيم من 0 حتى $2^n - 1$. يكتب التابع المنطقي المقابل للتابع الزوجي الموصف بالجدول أعلاه وفق التالي: تكتب تراكيب الدخل المقابلة لقيم الواحد فقط على الخرج، بحيث إذا كانت قيمة متحول الدخل صفر يكتب بالشكل المنفي وإذا كانت بالقيمة واحد يكتب بالشكل المباشر وبناء على ذلك يكون التابع المنطقي بالجدول 4-3 كالتالي: $F(a,b,c) = a'b'c' + a'bc' + ab'c' + abc'$ وهكذا الحال بالنسبة لأي تابع منطقي آخر

4-2-1 أشكال التابع المنطقي

يعبر عن كل تابع منطقي بقيم تقابل حالة الإمكانية التي يعبر عنها بالقيمة "1" أو عدم الإمكانية بتحقيق التابع والتي يعبر عنها بالقيمة "0". ويمكن بالنسبة لأي تابع منطقي إظهار حالة الإمكانية بالحدوث من حالة عدم الإمكانية بأسلوبين مختلفين: فمثلاً، بالنسبة لإنسان ما إن كان طويل القامة يقال عنه انه طويل، وهو تعبير مباشر أو أنه غير قصير (أي طويل) وهو تعبير غير مباشر. وبناء على ذلك يمكن أن تكون الصياغة الجبرية للتابع المنطقي بواحد من شكلين أساسيين، أحدهما متمم للآخر (الشكل الجامع والضارب). الأول- الشكل الجامع للتابع المنطقي: يمكن توضيح هذا المفهوم من خلال مثال محدد وليكن لدينا تابع منطقي من الشكل: $F_1(a,b,c) = a' + b'c + a'bc'$ يسمى كل من a' و $b'c$ و $a'bc'$ حد جداء P-Term وإذا كان أحد هذه الحدود مثل الحد $a'bc'$ يحوي كافة متحولات التابع (بالشكل المباشر أو المنفي) يسمى عندئذٍ بحد أصغري minterm ويكون التابع المعطى بالشكل غير القياسي. يصبح التابع بالشكل القياسي Standard Form إذا كانت جميع حدود التابع حدود أصغرية، بالتالي لتحويل الشكل غير القياسي لتابع منطقي

إلى الشكل القياسي يجب تحويل كل حد P-Term إلى حد أصغري minterm term وننتقل في ذلك من قوانين بول بإضافة المتحولات الناقصة كالتالي:

$$F_1(a,b,c) = a' + b'c + a'bc' = a'(b + b')(c + c') + (a + a')b'c + a'bc'$$

وبالاصلاح ينتج $F_1(a,b,c) = a'(bc + bc' + b'c + b'c') + ab'c + a'b'c + a'bc'$ وبأخذ حد واحد من الحدود المتكررة عملاً بالقانون $A + A = A$ وإلغاء كل حد فيه جداء متحول مع متممه نحصل على الشكل التالي:

$$F_1(a,b,c) = a'bc + a'bc' + a'b'c + a'b'c' + ab'c + a'b'c + a'bc' = \\ = a'bc + a'bc' + a'b'c + a'b'c' + ab'c$$

وهو الشكل القياسي للتابع المعطى. ويمكن أيضاً كتابة الشكل القياسي للتابع بالشكل المختصر بعد ترميز المتحولات بالنظام الثنائي وكتابة المكافئ العشري وفق التالي:

$$F_1(a,b,c) = a'bc + a'bc' + a'b'c + a'b'c' + ab'c \\ = 011 + 010 + 001 + 000 + 101 = \sum_m(0, 1, 2, 3, 5)$$

هذا الشكل النهائي الناتج للتابع المنطقي والذي يقابل حالة الامكانية يسمى بالشكل الجامع "Sum of Product" اختصاراً SOP وتتم 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.

الثاني- الشكل الضارب للتابع المنطقي: بإسلوب مماثل لما سبق يمكننا توضيح هذا المفهوم من خلال مثال محدد وليكن لدينا تابع منطقي معطى بالشكل:

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

حد مجموع S-Term وإذا كان أحد هذه الحدود مثل الحد $(a' + b + c')$ يحوي كافة متحولات التابع (بالشكل المباشر أو المنفي) يسمى عندئذٍ بحد أعظمي Maxterm ويكون التابع المعطى بالشكل غير القياسي. يصبح التابع بالشكل القياسي Standard Form إذا كانت جميع حدود التابع حدود أعظمية، بالتالي لتحويل الشكل غير القياسي لتابع منطقي إلى

الشكل القياسي يجب تحويل كل حد S-Term إلى حد أعظمي Maxterm وننتقل في ذلك من قوانين بول بإضافة المتحولات الناقصة كالتالي:

$$F_2(a,b,c) = a'(b+c')(a' + b + c') = (a' + bb' + cc')(aa' + b + c')(a' + b + c')$$

وبالاصلاح قد ينتج مثلاً تابع منطقي بصيغة من الشكل التالي:

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

وهو الشكل القياسي للتابع المعطى. ويمكن أيضاً كتابة هذا الشكل القياسي للتابع بالشكل المختصر بعد ترميز المتحولات بالنظام الثنائي وكتابة المكافئ العشري بصيغة الضرب وبحيث إن كل متحول بصيغة النفي يقابل الواحد والمتحول بالصيغة المباشرة يقابل الصفر،

$$F_2(a,b,c) = (a' + b + c)(a' + b' + c)(a' + b' + c') \quad \text{أي:}$$

$$= (1 + 0 + 0)(1 + 1 + 0)(1 + 1 + 1) = \Pi_M(4, 6, 7)$$

هذا الشكل النهائي الناتج للتابع المنطقي والذي يقابل حالة عدم الامكانية يسمى بالشكل الضارب "Product of Sum" اختصاراً POS وتتم كتابة هذه الصيغة انطلاقاً من الأسس

- 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 جميع متحولات الدخل.

وهنا نشير إلى أنه غالباً ما يكون التابع المنطقي بمرحلة التنجيز على شكل دائرة منطقية بالصيغة المختصرة التي لا تقابل الشكل القياسي، لكن يوجد بعض التطبيقات التي يلزم فيها التعامل مع الشكل القياسي للتابع. 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: أوجد باستخدام جدول الحقيقة الشكل القياسي للتابع $y = a' + bc' + abc$ ؟

الجواب: بما أن التابع المعطى هو بثلاثة متحولات ومعرف من أجل ثلاثة حدود فقط هي: a' أو bc' أو abc ، يكون جدول الحقيقة المقابل له من الشكل المبين أدناه.

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

لمعرفة أين يكون المتحول a في الشكل المنفي مثلاً، أي بالقيمة صفر؟ ينظر إلى الجدول، فيلاحظ أن ذلك يظهر بالأسطر الأربعة الأولى. وتوضع قيمة 1 لـ y في هذه الأسطر للتعبير عن أن التابع معرف من أجل هذه التراكيب. وبنفس الطريقة يتابع من أجل الحدين bc' و abc، والشكل القياسي للتابع يكون مجموع الحدود التي في كل منها y = 1 أي:

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

$$= 000 + 001 + 010 + 011 + 110 + 111 =$$

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

4-2-2 التحويل بين الأشكال القانونية للتابع المنطقية

يسمى الشكلان القياسان للتعبير عن التابع المنطقي (الشكل الجامع والشكل الضارب) بالصيغ القانونية للتابع المنطقي Canonical Form، وانطلاقاً من أن الشكلين الجامع والضارب لتابع ما متماثلان لبعضهما البعض، لا يمكن لدليل حالة ما أن يكون مشتركاً بين الشكلين بأن واحد. وهذا يعني أنه عند معرفة أحد أشكال التابع سواء أكان بالصيغة العادية أم القياسية يمكن بسهولة استنتاج الشكل الآخر. لذلك سنتعامل من الآن وصاعداً مع شكل واحد فقط للتابع المنطقي هو الشكل الجامع (الأكثر استخداماً) ما لم يذكر عكس ذلك. فمن أجل تابع منطقي بثلاثة متحولات تكون مجمل الحدود (الشكل القياسي) وفق ما هو مبين بالجدول 4-4 حيث المتمم بالشكل الجامع يعني صفر وبالشكل الضارب يعني الواحد وعوضاً عن إشارة الجمع بين حدود الجامع تستخدم إشارة الضرب بين حدود الضارب. وبالتالي لتحويل إحدى الصيغتين إلى الأخرى ننتقل من مثال محدد لظهور آلية التحويل، وليكن تحويل الشكل الجامع لتابع منطقي إلى الشكل الضارب المتمم له.

الجدول 4-4: الحدود الصغرى والعظمى الممكنة لتابع منطقي بثلاث متحولات

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	$m_6 = a bc'$	$M_6 = a' + b' + c$
1 1 1	$m_7 = a b c$	$M_7 = a' + b' + c'$

بفرض أن التابع معطى بالشكل الجامع بالصيغة المختصرة $F = \sum_m (1, 4, 6)$ ، ويطلب إيجاد الشكل الضارب المقابل له.

الجواب: بما أن التابع معطى بالشكل الجامع يعني أن الأرقام الواردة بين قوسين تمثل حالات التابع التي تكون فيها قيمته تساوي الواحد، وسيضم الحدود التالية:

$$F(a,b,c) = 1 + 4 + 6 = 001 + 100 + 110 = a'b'c + ab'c' + abc'$$

لتحويل هذه الصورة إلى الشكل الضارب تتبع الخطوات التالية:

1- 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' ، أي:

$$F' (a,b,c) = \sum_m (0,2,3,5,7) = a'b'c' + a'bc' + a'bc + ab'c + abc$$

2- يؤخذ نفي التابع الجديد مرة أخرى أي نفي F' للحصول على $F'' = F$ وفق

$$\text{الشكل: } F'' (a,b,c) = F = (a'b'c' + a'bc' + a'bc + ab'c + abc)'$$

العبارة الناتجة يتم الحصول على الشكل التالي:

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

3-4 البوابات المنطقية

لتحقيق (تنفيذ) العمليات المنطقية بتابع منطقي، تستخدم العناصر المنطقية التي تمثل أبسط كينونة فيزيائية لتنفيذ العمليات المنطقية. تسمى العناصر المنطقية أيضا بالبوابات المنطقية Logic gates. يرمز لكل بوابة منطقية برمز محدد. وتتميز كل بوابة منطقية عن غيرها بأربعة جوانب تعتمد عند دراستها هي: الاسم، والرمز المنطقي، وجدول الحقيقة، والتابع المنطقي وفق ما هو مبين بالشكل 3-4 لأول ثلاث بوابات منطقية من المستوى واحد وبوابة رابعة Buffer من المستوى صفر.

المستوى	الاسم	الرمز	جدول الحقيقة			التابع
صفر	1- دارئ Buffer	F a	دخل		خرج	F = a
			a	F		
			0	0		
			1	1		
	2-العاكس NOT	F a	دخل		خرج	F = a'
			a	F		
			0	1		
			1	0		
الأول	3-الضرب المنطقي AND	F a b	دخل		خرج	F = a . b
			a	b	F	
			0	0	0	
			0	1	0	
			1	0	0	
			1	1	1	
	4- الجمع المنطقي OR	F a b	دخل		خرج	F = a + b
			a	b	F	
			0	0	0	
			0	1	1	
			1	0	1	
			1	1	1	

الشكل 3-4: البوابات المنطقية الأساسية من المستويين صفر وواحد

يقصد هنا بالمستوى الذي تنتمي له البوابة عدد العمليات المنطقية التي تقوم البوابة بتنفيذها. فمثلاً، بوابة دارى 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. وتنتمي هذه البوابة للمستوى الأول، لأنها تقوم بعملية منطقية واحدة، هي عكس القيمة الموجودة على الدخل. ويمكن من جدول الحقيقة صياغة شكل التابع الذي يمثل التوصيف المنطقي للبوابة. ويلاحظ من الرموز المنطقية للبوابات، أن بعض البوابات مثل الدارى والعاكس يمتاز بوجود مدخل واحد ومخرج واحد، وأن للبعض الآخر مثل AND و OR يمكن أن يكون لكل منهما مدخلان أو أكثر لنتنتج إشارة خرج واحدة. الشكل الممكن لقيم الدخل والخرج على شكل إشارات ثنائية المستوى لكل من الدارى والعاكس وبوابات AND و OR (بمدخلين) مبين في الشكل 4-4.



الشكل 4-4: شكل إشارات الدخل والخرج لبعض البوابات المنطقية الأساسية

إضافة للبوابات المنطقية الاساسية يوجد بعض البوابات المنطقية المركبة. ولتحديد هذه البوابات المنطقية المركبة نطلق من أنه يمكن من دائرة منطقية بمدخل واحد (ممثلاً

بالمتحول الثنائي (a) تشكيل تابعين على الخرج كل منهما بقيمة مغايرة عن الآخر، هما التابعين: $F_0 = a'$ و $F_1 = a$ (أي 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 $2^n = 2^{2^n}$. 2^n 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 4-5.

الجدول 4-5: جدول الحقيقة للتوابع الستة عشر التي يمكن تعريفها باستخدام متحولين

Inputs	a	0	0	1	1	قيمة التابع	اسم التابع
	b	0	1	0	1		
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

يوجد من مجمل التوابع المبينة في الجدول أعلاه، تابعان F_0 و F_{15} تمثل الثوابت 0 و 1 وتقابل تسميات Null: for binary constant 0 و Identity: for binary constant 1 ولا يقابل أي منهما رمزاً لبوابة منطقية. تمثل البوابات الأساسية ستة من التوابع المبينة في الجدول 4-5، هي التوابع المنطقية: $F_1, F_3, F_5, F_7, F_{10}, F_{12}$. اثنان منها (F_1, F_7)

يقابلان البوابتين AND و OR والأربعة المتبقية تقابل بوابتي Buffer و NOT (حيث إن بعض البوابات تستخدم لتتجزئ تابعين مختلفين، كل منهما يؤدي وظيفة مماثلة لوظيفة الآخر، مثل عملية النقل Transfer التي تقابل تابعين منطقيين، أحدهما للمتحول a والثاني للمتحول b ونفس الشيء للمتمم). بقية التوابع المنطقية في الجدول أعلاه تقابل بوابات منطقية مركبة، بمعنى أن البوابة الواحدة تنجز أكثر من عملية منطقية. تصنف البوابات المركبة بمجموعتين، هما:

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

- اثنتان 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، "إذا" يوجد الآخر. لا يستخدم لهذه التوابع بوابات برموز محددة. الثانية: البوابات التي تقابل توابع قياسية، وتمثل بقية التوابع المبينة في الجدول أعلاه وعددها أربع بوابات، هي: F_6, F_8, F_9, F_{14} . تمثل هذه التوابع بأربع بوابات قياسية تسمى NAND و NOR و XOR و XNOR ويمكن بحسب عدد العمليات المنطقية التي تنفذها كل بوابة على حدة، تصنيف البوابات القياسية إلى ثلاثة مستويات هي:

- بوابات المستوى الأول، وتضم البوابات AND و OR و NOT.

- بوابات المستوى الثاني، وتضم البوابتين NAND و NOR.

- بوابات المستوى الثالث، وتضم البوابتين XOR و XNOR.

يظهر الشكل 4-5 رموز بوابات المستويين الثاني والثالث (البوابات المنطقية المركبة القياسية) مع الرمز وجدول الحقيقة والتابع المنطقي المقابل لكل منها على حدة. تُعدّ بوابات المستوى الثاني من أكثر أنواع البوابات انتشاراً، حيث كل منهما تقوم بعمليتين بأن 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) فتعني NOT OR، وتقوم في المرحلة الأولى بالجمع المنطقي وبنفي للناتج في المرحلة الثانية.

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

الشكل 4-5: البوابات المنطقية المركبة (من المستويين الثاني والثالث)

يمكن باستخدام بوابات المستوي الثاني رسم أية دائرة منطقية يتم تصميمها، وتصنع هذه البوابات المنطقية من قبل شركات متخصصة على شكل دارات متكاملة جاهزة للاستخدام في مختلف التطبيقات. يتم اللجوء بكثير من التطبيقات في أثناء التحيز الفعلي لبعض الدارات المنطقية إلى تشكيل بوابات منطقية غير متوافرة من بوابات أخرى متوافرة، ولإظهار ذلك يمكننا الانطلاق من قوانين بول: $a + a = a$ ، $a \cdot a = a$ وتشكيل بوابات NOT و AND و OR من بوابات NOR أو من بوابات NAND، حيث يعتمد بوابات بمدخلين أو أكثر وقصر (وصل) هذين المدخلين معاً، يمكن الحصول على بوابة بمدخل واحد تكافئ بالوظيفة البوابة NOT، كما هو مبين في الشكل 4-6. وبأسلوب مماثل يمكن تشكيل البوابة OR

من NAND و NOR، كما هو مبين في الشكل 4-6 ب، وتشكيل البوابة AND من NAND و NOR، كما هو مبين في الشكل 4-6 ج.

الشكل 4-6: تشكيل البوابات NOT و AND و OR من البوابتين NAND و NOR.

يمكن أيضاً تشكيل بوابات NOR من بوابات NAND، وكذلك تشكيل بوابات NAND من بوابات NOR وفق ما هو مبين في الشكل 4-7.

$$\frac{a+b}{2} = \frac{a+b}{2}$$

$$\frac{a+b}{2} = \frac{a+b}{2}$$

$$\frac{a+b}{2}$$

$$\frac{a+b}{2} = \frac{a+b}{2}$$

الشكل 4-7: تشكيل بوابة NOR من بوابات NAND وبوابة NAND من بوابات NOR

أما بوابات المستوى الثالث فتقوم كل منهما بالعمليات المنطقية الثلاث بأن واحد. وبالتالي كل منهما تمثل دائرة مركبة من بوابات المستويات الأدنى، حيث:

- البوابة XOR، تعني Exclusive-OR وهي تشبه التابع OR، إلا أنها تستثني التركيبة التي يكون فيها كل من المدخلين مساوياً للواحد، بمعنى أن XOR تكون بحالة 1 إذا كان أحد المداخل (a مثلاً) يساوي الواحد. ويظهر الشكل 4-8 التركيبة الداخلية لبوابة XOR باستخدام بوابات منطقية من المستوى الأول.



الشكل 4-8: التركيبة الداخلية للبوابة XOR

تمثل عملية XOR الجمع بالمود 2 ويرمز لها بالرمز $\oplus$ ، حيث: $a \oplus b = ab' + a'b$

بالتالي: $a \oplus a' = 1$ ، $a \oplus a = 0$ ، $a \oplus 1 = a'$ ، $a \oplus 0 = a$

وتجدر هنا الإشارة إلى أن العملية XOR هي تبادلية وتجميعية، بمعنى أن:

$$(A \oplus B) \oplus C = A \oplus (B \oplus C) = A \oplus B \oplus C$$

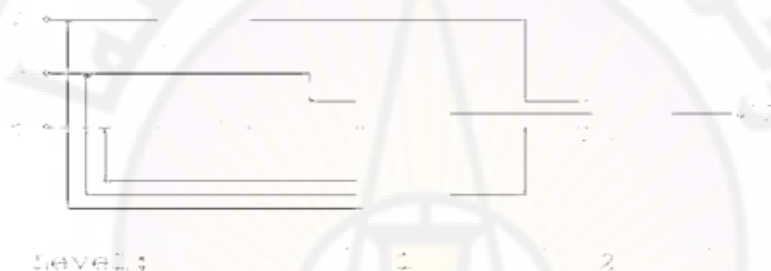
- البوابة XNOR، تعني التكافؤ Equivalence وهي تمثل التابع NOT XOR، أي متمم XOR وتكتب وفق الشكل التالي: $(a \oplus b)' = ab + a'b'$ بمعنى أن خرج هذه العملية يكون بحالة 1 عندما يكون المدخلان متماثلين سواء أكانت قيمتهما صفراً أم واحداً

4-4 الشبكات المنطقية

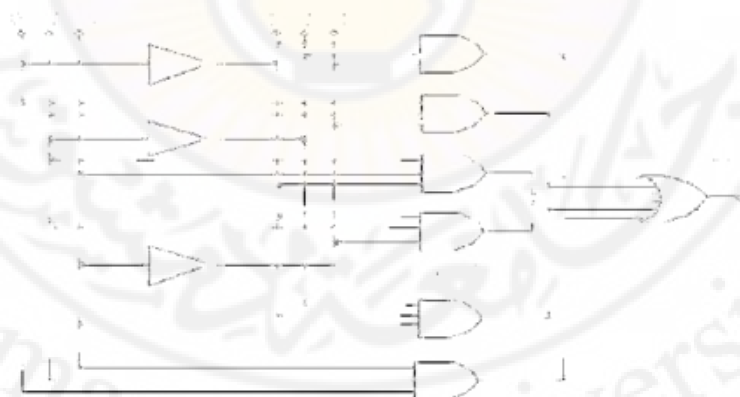
يسمى الشكل الذي يُظهر البوابات المنطقية المشكلة للدائرة المنطقية بالمخطط أو الشبكة المنطقية. تتكون الشبكة المنطقية من بوابة منطقية أو أكثر موزعة على أكثر من مستوى. تعرف الشبكة المنطقية Logic net بأنها مخطط منطقي مكون من عدة فروع. يطلق على عدد البوابات المنطقية المستخدمة في الفرع الواحد بطول الفرع. تبنى الشبكات المنطقية بمستويات متعددة، ولا يُعد المستوى الذي يرد عادة في البداية، ويضم العواكس فقط مستوى من مستويات الشبكة (لأنه جزء من الدخل). يمكن توضيح ذلك من خلال المثال التالي، وليكن رسم الشبكة المنطقية التي تحقق تابع منطقي من الشكل: $y = a' + b(c' + ac)$. الجواب: يتضح من شكل التابع أنه مكون من ثلاثة متحولات منطقية a, b, c وأن الشكل المعطى به التابع غير قياسي. باستخدام البوابات المنطقية المناسبة يمكن رسم الدائرة المنطقية التي تحقق التابع المعطى بعد مراعاة ترتيب الأولويات في تنفيذ العمليات المنطقية وفق ما هو مبين في الشكل 4-9 أ، بحيث إن الدائرة تحتاج إلى بوابتي NOT (لنفي المتحولين a, c) وبوابتي AND كل منهما بمدخلين وبوابتي OR كل منهما بمدخلين أيضاً موزعة على أربعة مستويات. يرسم بالمستوى (المرحلة) الأول بوابة AND التي تقابل الحد ac وبالمستوى الثاني رسمت بوابة OR لجمع خرج المستوى السابق (الأول) مع المتحول c' . وقد تم بالمستوى الثالث رسم بوابة AND أخرى تقابل ضرب ناتج المستوى السابق بالمتحول b وأخيراً بالمستوى الرابع تم رسم بوابة OR لجمع خرج ما سبق مع المتحول a' . وبالتالي يكون طول أطول فرع 4 وهو الفرع الذي إحدى مداخله c .



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



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



(ج) الدارة المقابلة للشكل $y = a'bc + a'bc' + a'b'c + a'b'c' + abc' + abc$

الشكل 4-9: أشكال مختلفة لدارات منطقية تقابل نفس التابع المنطقي

لإيجاد الصورة القياسية للتابع المعطى أعلاه، يعدل بالخطوة الأولى، الشكل العام للتابع بفك الأقواس، ليتم الحصول على $y = a' + bc' + abc$ وفق الشكل الجامع، ولكن ليس

بالصيغة القياسية. يضم الشكل الجديد للتابع المعطى حداً أصغرياً واحداً m-Term وحين نوع P-term. ويصبح المخطط العام للدائرة المقابلة لهذا الشكل الجديد للتابع كما هو مبين في الشكل 4-9 ب، حيث انخفض عدد المستويات من أربعة إلى اثنين فقط، وبالتالي هو بالمقارنة مع الشكل 4-9 أ (والذي يمثل نفس التابع) أفضل بكثير لاعتبار آخر يتمثل في أن انجاز تابع بمستويين يرافقه، كما سنجد لاحقاً قيمة تأخير أقل لانتشار الإشارات عبر البوابات المنطقية من الدخول إلى الخرج النهائي. ولإيجاد الشكل القياسي للتابع يتم اللجوء إلى قوانين جبر بول لاستكمال أي متحول ناقص. وبحسب الشكل المعطى به التابع تستخدم قوانين محددة من جبر بول وفق التالي:

أولاً- إذا كان التابع معطى بالشكل الضارب، تستخدم القوانين التالية:

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

ثانياً- إذا كان التابع معطى بالشكل الجامع، تستخدم القوانين التالية:

$$A + A = A, \quad A + A' = 1, \quad A \cdot 1 = A$$

وبما أنه يكفي الامام بإحدى شكلي التابع، نبين آلية التحويل بالنسبة للشكل الجامع، فيصبح الشكل $y = a' + bc' + abc$ للتابع المعطى على النحو التالي:

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

وتكون الدارة المنطقية التي تقابل هذا الشكل الأخير للتابع وفق ما هو مبين بالشكل 4-9 ج،

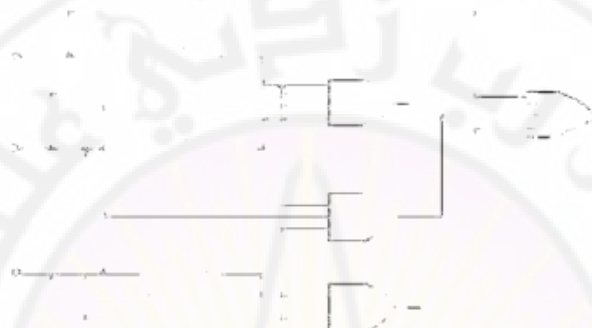
4-4-1 الشبكات المنطقية المقابلة للأشكال القياسية للتوابع المنطقية

تقابل الصيغ القياسية للتوابع المنطقية شبكات منطقية ثنائية المستوى لإنجاز التوابع المنطقية Two-Level implementation، هما:

- شبكات منطقية بالتركيبة AND-OR، وهي تقابل الشكل الجامع للتابع المنطقي.
 - شبكات منطقية بالتركيبة OR-AND، وهي تقابل الشكل الضارب للتابع المنطقي.
- نبين فيما يلي المواصفات العامة لكل من هاتين التركيبتين من خلال الأمثلة.

مثال: صمم الشبكة المنطقية التي تحقق التابع $F(a,b,c,d) = a + bc' + a'cd' + a'b'cd$

الجواب: يتضح من الشكل المعطى به التابع أنه يقابل التركيبية AND-OR، لأن كل حد بالتابع يمثل ببوابة AND والقيمة النهائية للتابع تعطى بخرج بوابة OR واحدة تمثل مجموع الجداءات، وبالتالي الشكل العام للشبكة المنطقية المقابلة يكون وفق ما هو مبين في الشكل 4-10.



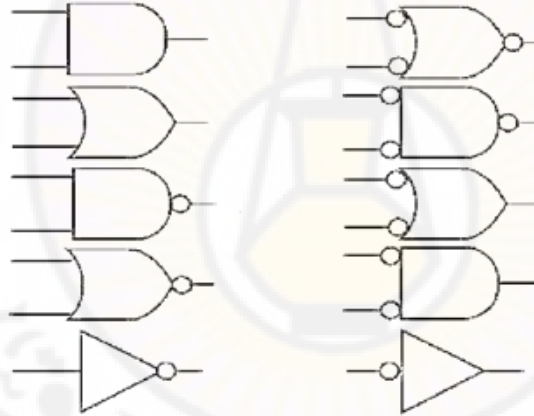
الشكل 4-10: شبكة منطقية بالتركيبية AND-OR (تقابل التابع المنطقي

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

بالطبع إن لم يكن شكل التابع المنطقي بالصيغة المعطاة أعلاه، يجب لرسم التابع وفق التركيبية AND-OR تحويله إلى شكل مماثل بصيغة الشكل الجامع للتابع، بحيث بعد الانتهاء من مرحلة العواكس، تمثل جميع الجداءات AND بالمرحلة الأولى والمجاميع OR بالمرحلة الثانية. ويمكن بأسلوب مماثل تمثيل الشكل الضارب المقابل للشكل الجامع للتابع المعطى بشبكة منطقية تقابل التركيبية OR-AND، بحيث تظهر في المستوى الأول بوابات منطقية من نوع OR وفي المستوى الثاني بوابات منطقية من نوع AND. بالإضافة إلى التراكيب التي تستخدم بوابات المستوى الأول لتجزير التوابع المنطقية، يصادف تراكيب أخرى أكثر انتشاراً لتنفيذ الشكلين الأساسيين للتابع المنطقي، تستخدم فيها بوابات منطقية من المستوى الثاني فقط.

2-4-4 شبكات NAND و NOR

تُعدّ الشبكات المنطقية بالتركيبة AND-NOT والتي تدعى اختصاراً بشبكات NAND وشبكات OR-NOT التي تدعى بشبكات NOR من أوسع الشبكات انتشاراً لاعتبارين: الأول: يمكن من بوابات المستوى الثاني تشكيل كافة بوابات المستوى الأول. الثاني: اعتماد بوابات المستوى الثاني معياراً قياسياً لتصنيع الدارات المتكاملة. يوجد إجراءات محدّدة لتحويل أي مخطط منطقي يستخدم بوابات من المستوى الأول بالتركيبتين AND-OR و OR-AND إلى مخططات منطقية (شبكات) مكافئة بالتركيبة NAND و NOR. وتُعمد لعملية التحويل إلى منطق NAND فقط أو NOR فقط رموز جديدة وفق ما هو مبين في الشكل 4-11.



الشكل 4-11: الرموز البديلة لكل من بوابتي NAND و NOR

تسمى البوابات الناتجة بالبوابات المعدلة، حيث الدائرة التي تدل على النفي يمكن ان تظهر على الدخل. وتصبح: - بوابة NAND العادية مكافئة لـ NOT OR
- بوابة NOR العادية مكافئة لـ NOT AND.

وبالنظر إلى أن طرق تصميم الشبكات المنطقية باستخدام عناصر NAND فقط، تعتمد على التركيبة AND-OR، يمكن باستخدام الطريقة المباشرة، تحويل التركيبة AND-OR إلى التركيبة NAND فقط وفق ما هو مبين بالمثل التالي:

مثال: صمم الشبكة المنطقية التي تحقق التابع المنطقي المعطى بالشكل الجامع
 $F(a,b,c,d) = a'd + ab'c + a'bcd'$ وفق التركيبة NAND.

الجواب: يمكن باستخدام قوانين ديمورغان ان نكتب:

$$F(a,b,c,d) = (F(a,b,c,d))' = (a'd + ab'c + a'bcd')' \\ = ((a'd)' (ab'c)' (a'bcd')')'$$

أي أن كلاً من الأقواس الداخلية في الشكل الذي تم التوصل إليه بالنسبة للتابع المعطى أصبح يمثل العملية المنطقية NAND والتابع بين الأقواس الخارجية يمثل بوابة NAND واحدة بثلاثة مداخل. ودوماً من الآن وصاعداً يقصد بالتركيبة NAND فقط، ألا يستخدم في رسم الشبكة إلا بوابات NAND وفق ماهو مبين في الشكل 4-12 بعد رسم بوابات NOT بقصر مداخل بوابات NAND.



الشكل 4-12: الشبكة المنطقية للتابع $F(a,b,c,d) = a'd + ab'c + a'bcd'$ بالتركيبة NAND فقط

وباعتماد الرموز الجديدة للبوابات المنطقية (التركيبة المكافئة) يمكن تنفيذ كل شبكة منطقية لتابع منطقي بمستويين، بواحد من ثلاثة أشكال أساسية، نبيّنها من خلال المثال التالي: مثال: ارسم أشكال الشبكات المنطقية الثلاثة الممكنة لتابع منطقي معطى بالصيغة $F(a, b, c, d) = ac + bd$

الجواب: يمكن رسم الشبكة المنطقية المقابلة لهذا التابع بواحدة من التراكيب الثلاثة المبينة في الشكل 4-13. حيث الشكل 4-13 أ يقابل التركيبة القياسية AND-OR. والشكل 4-13 ب يقابل التركيبة المعدلة NAND-NAND باستخدام بوابات NAND فقط والشكل 4-13 ج يقابل التركيبة المختلطة (Mixed Notation)، أي NAND – NAND

التي تستخدم الرموز القياسية لـ NAND بالمرحلة الأولى والرموز المعدلة لـ NAND بالمرحلة الثانية، وفق ما هو مبين بالشكل أعلاه.



$$\text{NAND - OR} = \text{NAND - NAND}$$

الشكل 4-13: الشبكات المنطقية الثلاث الممكنة لتابع منطقي معطى بالشكل الجامع

$$(F(a, b, c, d) = ac + bd)$$

يمكن نظرياً من أربع بوابات منطقية مختلفة (NAND, NOR, AND, OR)، تشكيل $2^4 = 16$ تركيبة مختلفة لشبكات كل منها ينفذ بمرحلتين، بحيث يوجد بالمرحلة الأولى إحدى البوابات وبالمرحلة الثانية توجد بوابة ثانية. طبعاً يمكن لبوابة واحدة أن تكون نفسها بالمرحلتين. وبالتالي ثمانية من هذه التراكيب يمكن اختصارها لعملية واحدة لا جدوى منها، مثل التركيبة OR - OR، لأن النتيجة النهائية ستكون OR لجميع متحولات الدخل. أما التراكيب الثمانية الباقية فيمكن استخدامها برسم الشبكات المنطقية المقابلة لتتابع معطاة بالشكل الجامع أو الضارب، إذ إن التابع الواحد يمكن رسمه بأكثر من تركيبة.

مثال: ارسم الأشكال الممكنة للشبكات المنطقية التي تقابل تابعاً منطقياً معطى بالشكل

$$F(a, b, c, d) = (a(c + d)(a + b + c))'$$

الجواب: يمكن تنفيذ التابع المعطى بواحدة من التراكيب المبينة بالشكل 4-14 بعد تعديل

شكل التابع المعطى إلى تراكيب جديدة مثل: OR-NAND أو NOR-OR وغيرها.

$$F(a, b, c, d) = (a(c + d)(a + b + c))' = a' + (c + d)' + (a + b + c)'$$



OR - Invert OR - OR - NAND

شكل 4-14: الشبكات المنطقية الثلاث الممكنة لتابع منطقي معطى بالشكل الضارب

$$(F(a, b, c, d) = (a(c + d)(a + b + c))')$$

3-4-4 التأخير الزمني في الشبكات المنطقية

يستخدم عادة المخطط التوقيتي للإشارات الكهربائية لتوصيف عمل الشبكات المنطقية بإظهار قيمة إشارة الخرج بكل لحظة بناء على القيم الموجودة للإشارات على الدخل. يظهر المخطط الزمني أيضاً مقدار التأخير الناتج من عبور الإشارات بالعناصر المنطقية المكونة للشبكة. نعرض فيما يلي مثلاً، لنظهر من خلاله مسألتين:

الأولى - كيفية كتابة (استخراج) التابع المنطقي لشبكة أو دائرة منطقية معطاة.

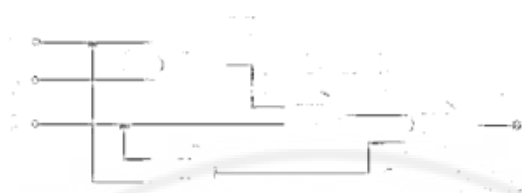
الثانية - إظهار مقدار التأخير الزمني خلال عبور الإشارات من الدخل إلى الخرج.

مثال: بفرض معطى شبكة منطقية غير قياسية بثلاث مراحل لتنفيذ تابع منطقي وفق ما هو

مبين في الشكل 4-15، بحيث تضم الشبكة بالمرحلة الأولى البوابة OR بالرقم I والبوابة

AND بالرقم II. وفي المرحلة الثانية تضم البوابة NAND فقط بالرقم III وفي المرحلة

الثالثة تضم البوابة NOR بالرقم IV.



شكل 4-15: شبكة منطقية بثلاثة مراحل

بفرض أن زمن تأخير مرور الإشارة في كل من هذه البوابات هو بحدود 20ns. يطلب:

- كتابة التابع المنطقي المقابل للشبكة المعطاة.

- رسم المخطط الزمني للشبكة وتحديد مقدار تأخير الإشارة.

الجواب: لكتابة التابع المنطقي، نرقم مداخل البوابات، كما هو مبين في الشكل أعلاه بالأرقام من 1 حتى 6، بحيث المدخل a يقابل الرقم 6 والمدخل b يقابل الرقم 5 والمدخل c

يقابل الرقم 4 يصبح المخطط المعطى

$$y = (1+2)' = ((3.4)' + 4.6)' = (((5+6).4)' + ac)' = ((ac+bc)' + ac)'$$

بتعديل هذا الشكل، يمكن الوصول إلى إحدى الأشكال القياسية للتابع. يمثل التابع المعطى

إحدى الحالات الخاصة، لأنه بعد فك الأقواس والإصلاح يمكن الحصول على الشكل

المختصر $y = a'bc$ ، أي إن التابع معرّف بحد واحد. أما بخصوص الطلب الثاني حول

رسم المخطط الزمني وحساب التأخير، فيمكن إعطاء قيم عشوائية للمداخل a, b, c وتتبع

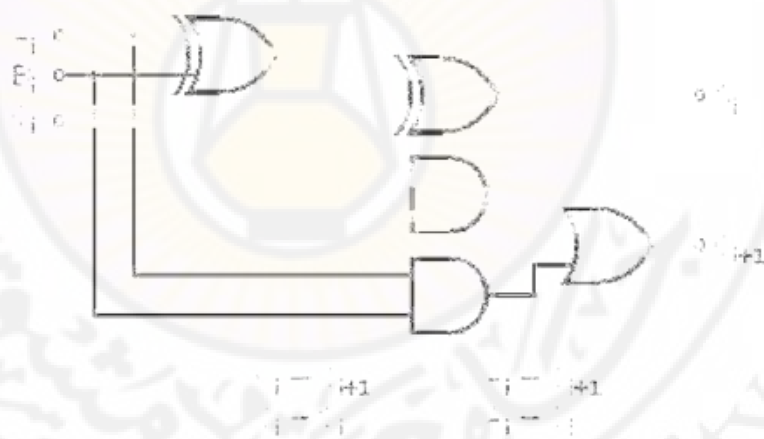
قيمة الخرج بكل لحظة. فإذا تم اعتماد الترتيب cba لمتحولات الدخل، يكون المخطط الزمني

للشبكة المبينة بالشكل 4-15 وفق ما هو مبين في الشكل 4-16. ومنه يتضح كيف يمكن



شكل 4-16: المخطط الزمني للشبكة التي تقابل التابع $y = a'bc$

توصيف الشبكة المنطقية باستخدام المخطط الزمني، حيث إن قيمة الخرج y تكون بحالة "1" فقط من أجل التركيبة (110) التي تقابل نفس الشكل الذي سبق وتم التوصل إليه بالطريقة الجبرية وهو $y = cba'$. ولحساب التأخير الزمني، يجب الأخذ بعين الاعتبار إن الشبكة المنطقية المعطاة تنفذ التابع على ثلاث مراحل. وبالتالي إجمالي التأخير يصل إلى 60ns من لحظة تطبيق القيم على الدخل إلى حين ظهور صدى ذلك على الخرج. يسمى التأخير الزمني أيضاً بتأخير التوليد (أو تأخير العبور Propagation delay) وهو تعبير تقني يملك عدة مفاهيم بحسب النص. وبشكل عام يمثل طول الزمن اللازم للوصول إلى الهدف. أي طول الفترة الزمنية التي تبدأ عندما تستقر القيم على مداخل البوابات المنطقية إلى حين استقرارها على الخرج. بالتالي تحديد زمن تأخير التوليد في الدارات المنطقية التركيبية يتطلب حساب زمن عبور الإشارة بأطول فرع من الدخل وحتى الخرج النهائي. فمثلاً بالشكل أدناه:



إشارة C_{i-1} تمثل تأخير ثلاث بوابات منطقية من لحظة تطبيق القيم على المداخل A_i, B_i

5-4 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

1- العملية المنطقية باستخدام البوابة NOT تعني:

- a - first complement b- difference
c- second complement d- None of all

2- وفق قانون الامتصاص العكسي، فإن العبارة $a'b' + b = a' + b$ تعادل

- a- $a + b$ b- $a' + b'$ c- $a' + b$ d- none of all

3- ليكن x متحول منطقي، حدد أي من العبارات المنطقية التالية غير صحيحة؟

- a- $x + 1 = x$ b- $x \cdot x' = 0$ c- $x + x' = 1$ d- $(x')' = x$

4- أي من العبارات المنطقية توصف خرج الدارة x المبينة أدناه؟



- a- $AB + C$ b- $A + BC$ c- $A(B + C')$ d- $A + B + C$

5- العبارة المنطقية $a \oplus a$ يعني (أو يساوي)

- a- 1 b- a c- 0 d- a'

6- إذا كانت x, y, z متحولات منطقية، أي من العبارات التالية تعتبر صحيحة؟

- a- $x \cdot (y + z) = (x \cdot y) + (x \cdot z)$ b- $x + x \cdot y = x$
c- $x \cdot (x' + y) = x \cdot y$ d- All the above

7- أي مما يلي ليست قاعدة صحيحة من قواعد الجبر البولياني؟

- a- $A + 1 = 1$ b- $A = A'$
c- $AA = A$ d- $A + 0 = A$

8- باعتماد قوانين دي مورغان، أي من المساواة التالية صحيحة؟

- a- $(AB)' = A' + B'$ b- $(XYZ)' = X' + Y' + Z'$
c- $(A + B + C)' = (ABC)'$ d- all of these

9- التابع المنطقي لبوابة التكافؤ (exclusive-NOR) بالمدخلين a, b يكون:

- a- $y = a'b + ab'$ b- $y = a'b' + ab$
c- $(a'b' + ab)'$ d- None of the above

10- المكافئ لصيغة الضارب POS من صيغة الجامع SOP لتابع منطقي معطى

a- $\Pi (0, 1, 2, 4, 5, 7)$

b- $\Pi (3,6)$

بالشكل $\Sigma(3,6)$ هو

c- $\Sigma(0, 1, 2, 4, 5, 7)$

d- none of all

ثانياً: أسئلة متفرقة

1- باستخدام طريقتي جدول الحقيقة والمفاتيح الكهربية، أثبت صحة قانون الامتصاص العكسي $a'b' + b = a' + b$ ؟

2- باستخدام الطريقة الجبرية اختصر العبارة المنطقية التالية:

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

3- أوجد جدول الحقيقة المقابل لتابع منطقي معطى بالشكل التالي:

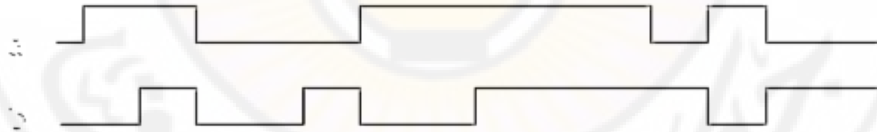
$F = a' + a'd + ab'c + bcd'$

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

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

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

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



a) $a \oplus b$

b) $a \text{ NAND } b$ أوجد المخطط المقابل لكل من

6- ليكن معطى تابع منطقي من الشكل: $Y = a + dd' + ab'c + ac'd$ يطلب:

- إيجاد جدول الحقيقة المقابل للتابع المعطى

- تحويل شكل التابع من صيغة SOP إلى صيغة POS

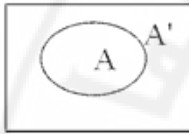
- رسم المخطط المنطقي المقابل للصيغة POS

الفصل الخامس: تمثيل التوابع المنطقية واختصارها

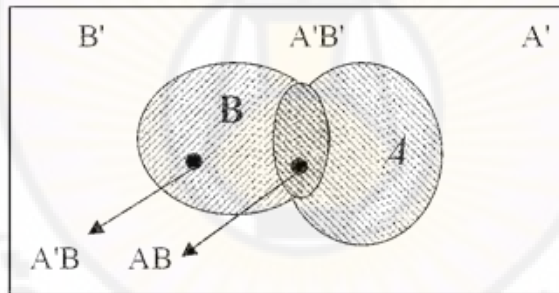
يتطرق هذا الفصل إلى طريقة تخطيطية للتعبير عن التوابع المنطقية، تتمثل باستخدام مخطط كارنوف Karnaugh Map لتمثيل التابع المنطقي بشكل تصويري واختصاره.

5-1 تمثيل المتحولات المنطقية باستخدام المخططات

انطلاقاً من مخطط فين Venn diagram يمكن توضيح طريقة المخططات (الخرائط) لاستخدامها في تمثيل التابع المنطقي. إذا تم باستخدام مخطط فين التعبير عن جميع الحالات الممكنة لمتحول ما مثل A بدائرة ضمن مستوٍ محدد، كما هو مبين في الشكل جانباً، فإن كل ما هو خارج الدائرة وضمن المستوي يمثل متمم المتحول A ويرمز له بـ A' .



وبالتالي يمكن تمثيل متحولين مثل A و B بمخطط فين، وفق ما هو مبين في الشكل 5-1.



الشكل 5-1: مخطط فين لمتحولين

يعدّ مخطط كارنوف (k-map) شكلاً منتظماً لمخطط فين، ويستخدم للتعبير عن العلاقة بين المتحولات المنطقية في جبر بول تخطيطياً. يمثل K-map صيغة بيانية لجدول الحقيقة بترتيب جديد، ويُعدّ من أكثر الوسائط استخداماً في تمثيل التوابع المنطقية وتصميم الدارات المنطقية المقابلة لها، كما أنه يمثل أداة فعالة في تمثيل التوابع المنطقية واختصارها، حيث يستخدم K-map بشكل أساسي كما سنجد لاحقاً في تبسيط التوابع المنطقية. يحتوي المخطط بدلاً من الأسطر في جدول الحقيقة على عدد من الخلايا (المربعات) يساوي عدد تراكيب جدول الحقيقة، بحيث إن كل خلية تقابل حداً أصغرياً واحداً من الحدود الأصغرية

للتابع أو حداً أعظماً. وهكذا إن كان التابع مكوناً من متغير منطقي واحد فإن عدد خلايا المخطط المناسبة لتمثيله هو 2، بحيث يعطى لنفي المتحول نفس الحيز المعطى للمتحول نفسه. فمثلاً، لتمثيل تابع بمتحول واحد a يمكن اعتماد المخطط المبين جانباً. يرمز بالمخطط للحيز المخصص للمتحول a بالقيمة 1 وللحيز المخصص لمتعمه بالقيمة 0.

0	a'
1	a

إذا كان التابع يحتوي على متحولين فإن المخطط سيحتوي على أربع خلايا بمعدل خليتين لكل من المتحولين وخليتين لمتعم كل منهما. وبشكل عام إن كان التابع مكوناً من n متحول فإن عدد خلايا المخطط المقابل له يساوي 2^n . يتم عند إنشاء مخطط كارنوف وفق ما هو متعارف عليه، تقسيم متحولات التابع إلى مجموعتين: الأولى - تضم نصف عدد المتحولات وتمثل بشكل أفقي.

والثانية- تضم النصف الآخر (إذا كان عدد المتحولات زوجياً أو النصف + 1 إن كان فردياً) لتمثيلها بشكل شاقولي.

لذلك مخطط كارنوف سيكون دوماً مربع الشكل إذا كان عدد المتحولات زوجياً أو مستطيلاً إذا كان العدد فردياً. فمثلاً، من أجل تابع منطقي بمتحولين a, b يقابله k -map مكون من $2^2 = 2^2 = 4$ خلية. قام كارنوف بتوزيع هذه الخلايا الأربع المتساوية بالمساحة بين المتحولين a و b ومتعميهما بالتساوي. ولتحقيق ذلك بحالة تابع بمتحولين، تقسم متحولات التابع المعطى إلى مجموعتين، بحيث الأولى تضم المتحول a الذي سيمثل هو ومتعمه أفقياً، والثانية تضم المتحول b لتمثيله هو ومتعمه شاقولياً، كما في الشكل 2-5.

a'	a
b'	b
b'	b
b'	b

الشكل 2-5: مخطط كارنوف لتابع منطقي بمتحولين

وبالتالي أي خلية تمثل تركيبة معينة من تركيبات متحولات الدخل، أي حد أصغري. يعد مخطط كار نوف مهما كان عدد متحولات التابع مخصصاً بكامله لتمثيل متحول واحد بغض النظر عن بقية المتحولات، بحيث يخصص نصف خلايا المخطط للمتحول، والنصف الآخر لمتممه. وبالتالي يمكننا إعادة إعداد رسم المخطط أعلاه بعد ترميزه ثنائياً، بحيث يرمز لكل متحول في الشكل العادي بالقيمة 1 ولتممه بالقيمة 0، كما هو مبين في المخطط بالشكل 3-5.

		b	
		0	1
a	0	00	01
	1	10	11

الشكل 3-5: الترميز الثنائي لمتحولين في مخطط كارنوف

يعتمد عند كتابة أي حد من المخطط، المباشرة بكتابة الرمز الموجود بالسطر ابتداء من اليسار، بمعنى إن التركيبة 10 مثلاً، تقابل الترميز ab' (أي إن قيمة المتحول $a = 1$ ، وقد كتب بأقصى اليسار وقيمة $b = 0$ ، أي b'). ومن أجل تابع منطقي بثلاثة متحولات منطقية $F(a, b, c)$ سيكون مخطط كار نوف مستطيل الشكل ومكوناً من ثماني خلايا. وبخصوص الترميز الثنائي للمتحولات المنطقية نشير إلى نقطتين:

الأولى: يقابل الترميز الثنائي لكل مربع رقم عشري يمثل دليل حالة للتابع المراد تمثيله. الثانية: يتم ترميز الخلايا في مخطط كار نوف وفق رماز كروي، وهذا يعني أن كل خليتين مختلفتين عن بعضهما ببت واحد في المخطط تكونان متجاورتين، لذلك عند ترميز متحولات العمود الثالث يعتمد الترميز $(11)_2$ للمحافظة على التجاور وفق ما هو مبين بالشكل 4-5.

		bc			
		00	01	11	10
a	0	$m_0 \sim 000$ $= (0)_{10}$	$m_1 \sim 001$ $= (1)_{10}$	$m_3 \sim 011$ $= (3)_{10}$	$m_2 \sim 010$ $= (2)_{10}$
	1	$m_4 \sim 100$ $= (4)_{10}$	$m_5 \sim 101$ $= (5)_{10}$	$m_7 \sim 111$ $= (7)_{10}$	$m_6 \sim 110$ $= (6)_{10}$

الشكل 4-5: الترميز الثنائي لثلاث متحولات في مخطط كارنوف وفق رماز كروي

ومن أجل تابع منطقي بأربعة متحولات من الشكل $F(a, b, c, d)$ ، يكون مخطط كارنوف وفق ما هو مبين في الشكل 5-5، وفيه يظهر شكل التجاور بين كل خليتين من خلاياه.

cd \ ab	00	01	11	10
00				
01				
11				
10				

$a'b'cd'$
 $0010 = (2)_{10}$
 هذان الحدان متجاوران لوجود
 تغير بيت واحد بين الحدين
 $ab'cd' \sim 1010 = (10)_{10}$

الشكل 5-5: مخطط كارنوف لتابع منطقي بأربع متحولات

تمثل المجموعة الأولى بالمتحولين ab أفقياً بالتسلسل التالي: 00 01 11 10 والمجموعة الثانية بالمتحولين cd عمودياً بنفس التسلسل السابق. ويقابل كل خلية في المخطط المتحولات الأربعة للتابع بتراكيبها المختلفة ابتداء من الشكل المنفي لجميع المتحولات $a'b'c'd' = 0000$ وصولاً إلى الشكل المباشر لكل المتحولات $abcd = 1111$ ، أي أن المخطط من أجل أربعة متحولات يضم ست عشرة تركيبة مختلفة في كل منها يمثل المتحول d (في أقصى اليمين) البت ذا الوزن الأدنى و a بأقصى اليسار بكان البت ذي الوزن الأعلى. ويمكن تمثيل تابع منطقي بخمسة متحولات من الشكل $F(a,b,c,d,e)$ بمخطط مكون من $2^5 = 32$ خلية، إما على شكل مستطيل واحد متصل (يمثل فيه المتحولان a و b بشكل أفقي والمتحولات الثلاث c, d, e بشكل عمودي)، أو على شكل مخططين كل منهما ب 16 خلية، وفق ما هو مبين في الشكل 5-6، حيث إن الجزء الأيسر (16 خلية) في المخطط أعلاه ب 32 خلية، مخصص للمتحول e ، والجزء الأيسر (16 خلية أخرى) لمتهمه أي e' ، أي لابد من حدوث تداخل بالحيز المخصص لمتحول ما مع الحيز المخصص لمتحول آخر أو مع الحيز المخصص لمتهمه متحول آخر. المهم هو مراعاة عدم وقوع متحول ما بكاملة بمنطقة متحول آخر أو بمنطقة متممه، ولذلك يلاحظ أن الحيز المخصص لمتحول ما، يمكن أن يكون متصلاً أو منفصلاً.

cd \ ab	00	01	11	10
00				
01				
11				
10				

$e' = 0$

cd \ ab	00	01	11	10
00				
01				
11				
10				

$e = 1$

هاتان الخليتان متجاورتان لأن الخلية الأولى (من اليسار) تقابل الحد الأصغري $a'bc'de$ أي الرمز 01011، والثانية تقابل الحد $a'bc'de'$ أي الرمز 01010، والاختلاف هو بمتحول واحد (الخانة التي تقابل المتحول e)

الشكل 5-6: مخطط كارنوف لتابع منطقي بخمس متحولات

يمكن تغيير ترتيب الرموز بمخطط كارنوف أيضاً ولا يؤثر ذلك على قيمة التابع شرط ان تتم المحافظة عند كتابة حدود التابع على نفس الترتيب الذي يعتمد من البداية حتى النهاية، بمعنى أنه من أجل تابع منطقي بخمسة متحولات يمكن اعتماد الترتيب abcde أو الترتيب edcba لمتحولات نفس التابع.

5-1-1 مخطط كارنوف والأشكال القياسية للتابع المنطقي

يقابل تمثيل التابع المنطقي بمخطط كارنوف الشكل القياسي للتابع سواء أكان بصيغة الشكل الجامع أم الضارب. وبالتالي كل خلية بمخطط كارنوف تقابل حد أصغري إذا كان التابع معطى بالشكل الجامع، أو حد أعظمي من حدود التابع المقابل للمخطط المعتمد إذا كان التابع معطى بالشكل الضارب. يُمثل التابع المنطقي بمخطط كارنوف إذا كان بالشكل الجامع (حالة الامكانية) بكتابة (1) مقابل كل حد أصغري معرّف به التابع و (0) في الخلايا التي تقابل الحدود التي لا يكون التابع معرّف بها. يصادف في أثناء التعامل مع مخطط كارنوف حالتين أساسيتين هما:

الأولى: استخدام المخطط لتمثيل تابع منطقي معطى بصيغته الجبرية القياسية.
مثال: ليكن معطى تابع منطقي بالشكل الجامع وبالصيغة: $F = \Sigma (1, 4, 5, 6)$
والمطلوب تمثيله باستخدام مخطط كارنوف.

الجواب: بداية يحدّد شكل مخطط كارنوف المناسب وذلك بتحديد عدد كل من الأسطر والأعمدة، وهذا يتم بالاتّفاق من أكبر دليل حالة معطى بالتابع. ويلاحظ بالمثال المعطى، أن أكبر دليل حالة هو الرقم (6)، ولتمثيل هذا الرقم العشري بشكل ثنائي يلزم ثلاثة متحولات. وبفرض أن هذه المتحولات هي a, b, c. يلزم لتمثيل التابع مخطط بـ $2^3 = 8$ خلية. ويتم تمثيل التابع بكتابة 1 في خلايا المخطط التي لها نفس دلالات الحالة المعطاة بالتابع وبقيّة الخلايا يكتب فيها 0 وفق ما هو مبين في الشكل 5-7.

a \ bc	00	01	11	10
0	0	1	0	0
1	1	1	0	1

الشكل 5-7 مخطط كارنوف المقابل لتابع منطقي معطى بالصيغة المبسطة

يمكن تمثيل التابع إذا كان معطى بالشكل الضارب بأسلوب مماثل، مع فارق أن قيمة التابع بدليل الحالة المعطى تكون صفراً.

الثانية: استخدام مخطط كارنوف للحالة المعاكسة، أي كتابة شكل التابع الممثل في مخطط كارنوف سواء أكان التابع معطى بالشكل الجامع أم بالشكل الضارب. وكمثال على ذلك، يطلب كتابة الصيغة الجبرية للتابع المعطى بمخطط كارنوف المبين في الشكل 5-8.
الجواب: إذا طلب كتابة التابع بالشكل الجامع، يلاحظ من المخطط وجود خمس خلايا كل منها بالقيمة واحد. بالتالي تكتب الحدود المقابلة للخلايا التي قيمة كل منها واحد فيكون الشكل الجبري للتابع: $F = a'b'c'd' + a'b'cd' + a'bc'd' + abc'd' + abcd$ وبالصيغة المبسطة يكتب الشكل النهائي للتابع المعطى بالشكل: $F(a,b,c,d) = \Sigma_m(0,2,5,12,15)$
أما إذا افترضنا أن نفس المخطط يقابل تابع منطقي معطى بالشكل الضارب يكون الشكل

النهائي للتابع المعطى ممثلاً بجداء جميع دلائل الحالة التي قيمة كل منها صفراً، أي
 $F(a,b,c,d) = \prod_M(1,3,4,6,7,8,9,10,11,13,14).$

cd \ ab	00	01	11	10
00	1	0	0	1
01	0	1	0	0
11	1	0	1	0
10	0	0	0	0

الشكل 5-8: مخطط كارنوف لتابع منطقي بأربع متحولات

وإذا طلب كتابة التابع بالشكل الضارب، فيكون الناتج

$$F(a,b,c,d) = \prod_M(1,3,4,6,7,8,9,10,11,13,14)$$

يمكن باستخدام مخطط كارنوف إجراء عملية التحويل بين الشكلين الجامع والضارب للتابع المنطقي بأسلوب أسهل وأسرع من الطريقة التي سبق وتمت الإشارة إليها بالفصل الرابع. مثال: ليكن معطى تابع منطقي بالصيغة المبسطة للشكل الجامع $F = \sum (1, 4, 5, 6)$ ، يطلب باستخدام مخطط كارنوف إيجاد الشكل الضارب لنفس التابع؟

الجواب: بما أن الشكل الضارب للتابع هو متمم للشكل الجامع لنفس التابع، فإن الشكل الضارب المطلوب يمثل جميع الحدود التي قيمة كل منها في المخطط صفر، أي الحدود بدلائل الحالة (0, 2, 3, 7). وبالتالي يمكن كتابة الشكل الضارب للتابع مباشرة وفق الشكل

$$F = \prod_M(0, 2, 3, 7)$$

5-1-2 حالات عدم الاكتراث (التعين)

يصادف ببعض التوابع المنطقية تراكيب على الدخل لا تكون من أجلها قيمة الخرج محددة بقيمة معينة. تدعى مثل هذه التراكيب (الحالات) التي تكون فيها قيمة الخرج غير محددة، بحالات عدم الاكتراث Don't Care. ويمكننا إظهار هذا المفهوم من خلال الأمثلة. مثال: صمم دائرة منطقية للتحكم بآلية تعبئة مسبح (خزان) بالماء بواسطة مضختين وفق

الشروط التالية: بفرض أن مسيح بعمق ثلاثة أمتار مزود بمضختين، الأولى M_1 باستطاعة 10 أنش والثانية M_2 باستطاعة 5 أنش وبثلاثة حساسات لسويات الماء في المسبح (فواشات) مرمزة بالأحرف a, b, c ، بحيث الرمز a يدل على عمق المتر الأول من أسفل المسبح (ويأخذ القيمة $a = 0$ إذا كان المتر الأول من المسبح لم يمتلئ بالماء بعد و $a = 1$ مجرد أن أصبح ممتلئاً). والمتحول b يرمز لعمق المتر الثاني من المسبح (أي مسافة متر آخر فوق a). والمتحول c يدل على مسافة متر آخر فوق b أو ما يعادل سطح المسبح، والآن: إذا كان المسبح فارغاً، يطلب أن تعمل المضختان M_1, M_2 معاً لتعبئته بسرعة. وإذا كان مستوى الماء في المسبح بين الموضعين a و b تعمل فقط المضخة M_1 . وعندما يصبح مستوى الماء بين الموضعين b و c تعمل فقط المضخة M_2 . وعندما يصبح مستوى الماء عند الموضع c يجب توقيف كلتا المضختين M_1, M_2 . أي أن المسألة تحولت إلى تصميم دائرة منطقية بثلاثة مداخل a, b, c وبمخرجين M_1, M_2 كل منهما يأخذ قيمتين (0 وتعني حالة عدم تشغيل أي من المضختين و 1 تعني التشغيل). لتصميم مثل هذه الدارة، يتم الانطلاق من جدول الحقيقة الذي يعكس فهمنا للمسألة المعطاة بالشكل الموصف أعلاه. وانطلاقاً من الترميز الوارد بنص السؤال يكون جدول الحقيقة للمسألة وفق الجدول 5-1.

الجدول 5-1: جدول الحقيقة لدائرة منطقية بوجود حالات عدم تعيين

Index	Inputs			Outputs	
S	c	b	a	M_1	M_2
0	0	0	0	1	1
1	0	0	1	1	0
2	0	1	0	d	d
3	0	1	1	0	1
4	1	0	0	d	d
5	1	0	1	d	d
6	1	1	0	d	d
7	1	1	1	0	0

حيث من أجل التركيبية ذات دليل الحالة (2) مثلاً، التي تقابل الترميز $cba = 010$ ، تم وضع الرمز d (من Don't Care) مقابل قيمة كل من الخرجين M_1 و M_2 للإشارة إلى

حالة عدم الاكتراث (التحديد)، لأنه من غير المعقول أن يكون المتر الأوسط (الثاني) من مسبح ما مليء بالماء والمتر الذي يقع تحته غير مليء، وهذا ما يقصد بحالة عدم الاكتراث (حالة من دون معنى) لا يمكن توصيفها بـ 0 أو 1. وتكرر في هذا المثال القيمة d بالنسبة لدلائل الحالة 6, 5, 4 أيضاً ولنفس الأسباب. عند كتابة توابع الخرج من جدول الحقيقة وبالشكل الجامع مثلاً، نعبّر عن حالات عدم التعين بشكل منفصل تحت الرمز Σ بالدليل d أي من الشكل Σ_d ، وتضاف هذه الحالات إلى الحالات التي يكون فيها التابع معرفاً، وفق ما هو مبين أدناه بالنسبة للتابعين M_1 و M_2 حيث:

$$M_1(a, b, c) = \Sigma_m(0, 1) + \Sigma_d(2, 4, 5, 6)$$

$$M_2(a, b, c) = \Sigma_m(0, 3) + \Sigma_d(2, 4, 5, 6)$$

ويمكن تمثيل كل من التابعين بمخطط كار نوف بوجود حالات عدم الاكتراث d كالتالي:

a	bc			
	00	01	11	10
0	1	1	0	d
1	d	d	0	d

التابع M_1

a	bc			
	00	01	11	10
0	1	0	1	d
1	d	d	0	d

التابع M_2

2-5 اختصار التوابع المنطقية

يفترض بالخطوة الأولى لتصميم أي دائرة منطقية تحديد الشكل العام للتابع المنطقي واختصاره قبل المباشرة بتتجيز الدارة إلى أبسط شكل ممكن، لما لموضوع الاختصار من أهمية سواء على الصعيد المادي أو على صعيد تبسيط درجة تعقيد الدارة. وتعني عملية اختصار التابع المنطقي جعل عدد حدود التابع أقل ما يمكن وجعل عدد المتحولات بكل حد أقل ما يمكن خاصة المنفي منها. يوجد عدة طرق لاختصار التوابع المنطقية من أهمها:

- الطريقة الجبرية
- طريقة المخططات
- الطريقة الجدولية

5-2-1 الطريقة الجبرية

يقصد بالطريقة الجبرية في اختصار التوابع المنطقية وفق ما أشير إلى ذلك في الفصل الرابع، استخدام قوانين جبر بول في إعادة صياغة التابع بأبسط شكل ممكن دون التغير بقيمته. تُعدّ الطريقة الجبرية من أقلّ الطرائق استخداماً بحكم إنها طويلة ومملة وتحتاج إلى الالمام بكافة قوانين جبر بول، وبخاصة أن كان التابع المنطقي كبيراً. يمكن تلخيص أهم القواعد التي تتبع في تبسيط العبارات المنطقية عند استخدام الطريقة الجبرية بالتالي:

1- دمج كل حدين يضمنان متحول منطقي مشترك في حد واحد، عدد المتحولات فيه يكون أقلّ بواحد من عدد المتحولات في كلّ من الحدين الأساسيين، وذلك باستخدام القانون

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

2- حذف حد باستخدام أحد القوانين: - قانون التماثل $a + a = a$

- قانون التبسيط $a + ab = a(1 + b) = a$

- قانون الحد الفائض $ab + bc + a'c = ab + a'c$

3- حذف بعض الرموز في حد ما ضمن العبارة باستخدام قانون الامتصاص

$$a + a'b = a + b$$

4- استخدام متحولات منطقية أو حدود غير موجودة بالعبارة المنطقية مثل الحد الفائض بقصد اختصار متحولات أو حدود أخرى بالعبارة المعطاة، أو استخدام قوانين مثل:

$$a + a' = 1 \quad \text{أو} \quad a \cdot a' = 0$$

مثال - أوجد أبسط شكل ممكن للعبارة المنطقية التالية: $a \cdot b + a' \cdot c + b \cdot c$

الجواب: باستخدام قوانين بول وإجراء بعض التعديلات، يمكن إعادة صياغة العبارة

المعطاة وفق الشكل التالي:

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

$$= a.b + a'.c + a.b.c + a'.b.c$$

$$= a.b + a'.c.(1 + b) + a'.b.c$$

$$= a.b.(1 + c) + a'.c$$

$$= a.b + a'.c$$

يتضح من هذا المثال أنه تم الاستغناء عن حد بكامله $(b.c)$ من دون أن تتغير قيمة التابع.

5-2-2 طريقة المخططات

تمتاز طريقة المخططات (تحديداً استخدام مخطط كار نوف) في إنها تساعد بشكل فعال على تبسيط التتابع المنطقية إلى أقل عدد ممكن من الحدود وعدد المتحولات في الحد الواحد بشكل سريع، حيث يمكن بعد تمثيل التابع المعطى بمخطط كار نوف، إيجاز مراحل اختصار أي تابع باستخدام هذه الطريقة بالخطوات التالية:

- إحاطة جميع الوحدات الممثلة في المخطط من دون استثناء بحلقات، إذ إن كل حلقة مهما كان عدد الوحدات فيها ستقابل في الشكل المختصر للتابع حداً واحداً.
- دمج الخلايا (الحدود) المتجاورة في المخطط والتي في كل منها قيمة التابع تساوي 1.
- المحافظة على أن يكون عدد الوحدات في كل حلقة من رتبة 2^n ، أي من أجل $n = 0, 1, 2$ ، سيكون عدد الوحدات في الحلقة الواحدة 1 أو 2 أو 4 ، أي لا يمكن وضع ست وحدات مثلاً بحلقة واحدة.
- جعل حجم الحلقة الواحدة أكبر ما يمكن (أي أن يكون عدد الوحدات المتجاورة فيها أكبر ما يمكن)، لأن ذلك سينعكس على عدد المتحولات ضمن الحد الواحد.
- جعل عدد الحلقات أقل ما يمكن، لأن عدد الحلقات سينعكس على عدد الحدود. بتطبيق هذه الخوارزمية حول اختصار التابع المنطقي باستخدام مخطط كار نوف، يلاحظ أن أي خلية تحتوي على قيمة واحد، يمكن أن تشارك في أكثر من حلقة، وبالتالي تمتاز كل حلقة من الحلقات التي يمكن الحصول عليها بالمخطط بحسب عدد الواحدات بالخصائص التالية:
- الحلقة الأحادية: تحتوي على واحد فقط، وهذا يعني أن الحد المقابل لها غير متجاور مع أي حد آخر. ويقال عادة عن العنصر المقابل لهذه الحلقة إنه عنصر معزول. إذن الحلقة الأحادية لا يمكن منها اختصار أي شيء، وسيحتوي الحد المقابل لها على جميع متحولات التابع بشكلها العادي أو المتمم، وهي تمثل حداً أصغرياً أو حداً أعظمياً حسب شكل التابع.
- حلقات ثنائية: وهي حلقات تحتوي على واحدتين، وتسمح مثل هذه الحلقة باختصار متحول واحد من الحد الذي سيقابلها.
- حلقات رباعية: وهي حلقات تحتوي على 4 من الوحدات المتجاورة. وبالاختصار ستسمح مثل هذه الحلقة باختصار متحولين. وهكذا الحال بالنسبة للحلقات الأكبر.

يمكن اختصار عدد من المتحولات بحسب حجم الحلقة، بحيث إذا كان حجم الحلقة 2^n يمكن اختصار n متحول من الحد المقابل. ويمكن التعبير عن الحلقات أو كتابة الحدود المقابلة لكل منها على النحو التالي: أي متحول يقع جزء من قيمته في الشكل العادي وجزء من متممه ضمن نفس الحلقة يلغى من الحد، وتكتب فقط المتحولات التي تقع بكاملها ضمن الحلقة الواحدة. وفيما يلي سيتم عرض مجموعة من الأمثلة حول هذه المفاهيم المتعلقة باختصار التوابيع المنطقية باستخدام مخططات كارنوف ابتداء من أبسطها

		b	0	1
a	0	1	1	
	1	0	1	

مثال: بفرض معطى تابع منطقي بمخطط كارنوف المبين في الشكل جانباً. يطلب اختصار هذا التابع إلى أبسط شكل.

الجواب: يلاحظ من المخطط المعطى أن الشكل العام للتابع

المعطى قبل الاختصار هو: $F = a'b' + a'b + ab$.

وباختصار هذا الشكل باستخدام الطريقة الجبرية يكون الناتج:

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

أما باستخدام مخطط كارنوف لاختصار نفس التابع، يلاحظ أنه بتطبيق المبادئ الواردة سابقاً حول هذه الطريقة، يمكن وضع كل واحد من متجاورين بحلقة واحدة كما هو مبين في الشكل التالي:

		b	0	1
a	0	1	1	1
	1	0	1	1

حلقه 1

حلقه 2

تُشكل حلقتان 1 و 2 بالوحدات الموجودة في المخطط، وبوساطتهما يمكن الحصول على أبسط شكل للتابع، بحيث:

حلقة 1 + حلقة 2 = F . ويمكن من الحلقة 1 الحصول على الحد a' فقط، لأن كامل الحلقة تقع بالحيز المخصص للمتحول a' . أما المتحول b فلا يجوز أن يظهر بهذا الحد لأن نصف الحلقة يقع بالحيز المخصص لـ b ، والنصف الآخر يقع بالحيز المخصص لـ b' ، وبالتالي يلغى هذا المتحول عند كتابة هذه الحلقة. بأسلوب مماثل، يمكن من الحلقة 2 الحصول على الحد b فقط. وهكذا أبسط شكل للتابع سيكون: $F = a' + b$ وهو نفس الشكل الذي تم الحصول عليه بالطريقة الجبرية، لكن بشكل أبسط.

ويمكن الانتقال إلى توابع منطقية بثلاث متحولات، لإظهار مختلف الحالات حول التشكيل الصحيح للحلقات من خلال الأمثلة الميَّنة أدناه.

مثال 1: بفرض معطى تابع منطقي بثلاثة متحولات، وفق مخطط كار نوف المبين في الشكل التالي:

a \ bc	00	01	11	10
0	0	1	1	1
1	0	1	1	1

يطلب إيجاد أبسط شكل ممكن للتابع المقابل.

الجواب: عند كتابة التابع المعطى في المخطط بصيغته الجبرية، سيكون الناتج وفق الطريقة المعروفة بكتابة التابع على النحو التالي:

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

ولاختصار هذا التابع بالطريقة الجبرية قد يطول الحل. أما باستخدام مخطط كار نوف، فيمكن اختصار التابع بسرعة كبيرة، لأن جميع الوحدات الممثلة في المخطط على الرغم من أن عددها ستة يمكن وضعها بحلقتين وفق ما هو مبين في الشكل أدناه.

a \ bc	00	01	11	10
0	0	1	1	1
1	0	1	1	1

حلقة 1 → (01, 11, 10) for a=0
 حلقة 2 → (01, 11, 10) for a=1

وهذا يعني أن الشكل المختصر للتابع سيكون مكوناً من حدين فقط (بعدد الحلقات) وكل حد سيكون مكوناً من متحول واحد، لأن الحلقة الرباعية تختصر متحولين من أصل ثلاثة وهكذا فإن ناتج الحلقة الأولى هو المتحول c فقط. وناتج الحلقة الثانية هو المتحول b أي أن: $F(a, b, c) = b + c$. في حال عدم الاختيار المناسب لعدد الحلقات أو لحجمها قد لا يكون الجواب خاطئاً، وإنما شكل الناتج لا يكون هو أبسط شكل ممكن.

مثال 2: بفرض معطى تابع منطقي بثلاثة متحولات، وفق المخطط المبين أدناه. يطلب إيجاد أبسط شكل ممكن للتابع المقابل؟

a \ bc	00	01	11	10
0	1	0	0	1
1	1	0	0	1

الجواب: يمكن ضم أول واحدين بأقصى اليسار (في العمود الأول) بحلقة مستقلة، لأنهما متجاوران، كما أنه يمكن ضم الواحدين بأقصى اليمين (في العمود الرابع) بحلقة واحدة. وتكون نتيجة الحلقة الأولى $b'c'$ ونتيجة الحلقة الثانية bc' ، أي أن الناتج هو $F(a, b, c) = b'c' + bc'$ ، ولكن هذه النتيجة لا تمثل أبسط شكل للتابع، لأنه لم يتم اختيار الحلقات بشكل صحيح. وللاختيار الصحيح للحلقات، يمكننا تصور مخطط كارنوف مثل كرة مفتوحة، وبالتالي الطرف الأيمن للمخطط مجاور للطرف الأيسر وجميع أطراف المخطط متناظران بالنسبة للمحاور. وهذا يعني أن الواحد الموجود في الخلية الواقعة بالسطر الأول والعمود الأول مجاور للواحد الموجود في الخلية الواقعة بالسطر الأول والعمود الرابع، وهذا بدوره يعني أن الوحدات الأربع متجاورة ويمكن ضمها بحلقة واحدة مكونة من جزأين وفق ما هو مبين بالشكل أدناه.

a \ bc	00	01	11	10
0	1	0	0	1
1	1	0	0	1

يتضح عند كتابة الحد الذي يقابل الحلقة المبينة في الشكل أعلاه، إن المتحولين a و a' ظهرا بنفس الحلقة لذا يلغيان. نفس الشيء بالنسبة للمتحول b و b' ، أما المتحول c فلا يظهر بالحلقة أبداً. إذاً كامل الحلقة مغطاة بمتحول واحد لم يغير قيمته هو المتحول c' ، لذلك فإن نتيجة الاختصار ستكون $F = c'$ ، وهو أبسط شكل ممكن للتابع المعطى.

مثال 3: يطلب إيجاد أبسط شكل ممكن للتابع المعطى بمخطط كارنوف المبين بالشكل

	bc	00	01	11	10
a	0	1	1	0	1
	1	0	1	0	1

الجواب: يوجد في مثل هذه الأمثلة إكاثنتان

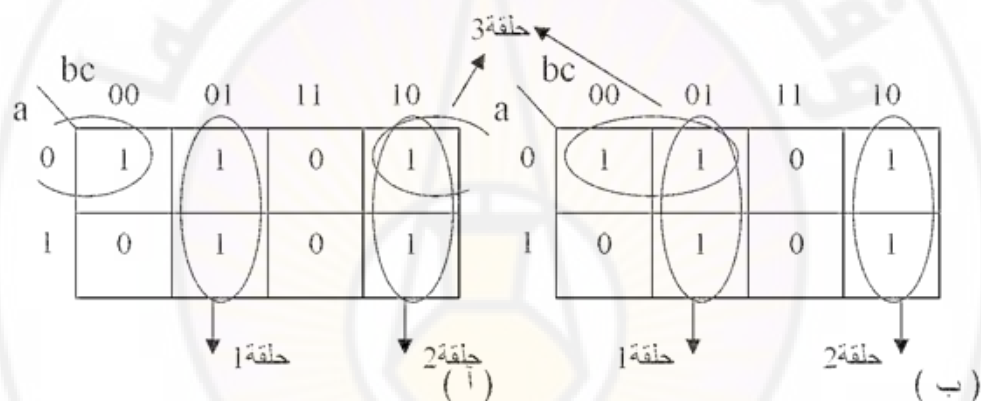
لتشكيل الحلقات، تنتج من شكل توزع

الوحدات في المخطط وهما في هذا المثال:

الأولى: أن يوضع الواحد الموجود بالخلية

المقابلة للتركيبية 000 مع الواحد بالخلية

المقابلة للتركيبية 010 بحلقة واحدة وفق ما هو مبين في الشكل 5-9 (أ).



الشكل 5-9: شكلان مختلفان ومتكافئان لتشكيل حلقات لنفس التابع في مخطط كارنوف

الثانية: أن يوضع الواحد بالخلية المقابلة للتركيبية 000 مع الواحد بالخلية المقابلة للتركيبية

001 بحلقة واحدة وفق ما هو مبين في الشكل 5-9 (ب). بالتالي يلزم بالحد الأدنى ثلاث

حلقات، هي: الحلقتان 1 و 2 ثابتتان بالحالتين، والحلقة الثالثة يمكن أن تتغير وفق واحدة

من الحالتين المبينتين أعلاه. وتكون نتيجة الاختصار بحسب المخطط (أ) على الشكل

التالي: $F(a, b, c) = a'c' + b'c + bc'$ وبحسب المخطط (ب) على الشكل التالي:

$F(a, b, c) = a'b' + b'c + bc'$. الجواب في الحالتين مقبول، لأنه في حقيقة الأمر

الجوابان متكافئان على الرغم من اختلافهما الظاهري بالحددين $a'b'$ و $a'c'$. ويفهم ذلك بأن

الحد $a'c'$ مع الحدين $(b'c + bc')$ يكافئ الحد $a'b'$ مع نفس الحدين. ويمكن إثبات ذلك

بتحويل كل من النتيجتين للشكل القياسي للتابع، حيث كلاً من النتيجتين يكافئ الشكل:

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

يلاحظ فيما إن تم أخذ الحلقتين 1 و 2 فقط واعتبار الواحد بالخلية المقابلة للتركيبة 000 بحلقة منفردة، تكون النتيجة من الشكل $F(a, b, c) = a'b'c' + b'c + bc'$ غير خاطئة، ولكنها لا تمثل أبسط شكل للتابع.

مثال 4: بفرض معطى تابع منطقي بثلاثة متحولات أيضاً وفق مخطط كار نوف

a \ bc	00	01	11	10
0	1	0	1	0
1	0	1	0	0

المبين في الشكل جانباً. يطلب إيجاد

أبسط شكل ممكن للتابع المقابل.

الجواب: لا يمكن في هذا المثال دمج أي من الوحدات المعطاة بالتابع مع الآخر لعدم وجود تجاور بين أي منها، وبالتالي كل من هذه الوحدات يمثل عنصراً معزولاً وله حلقة

a \ bc	00	01	11	10
0	1		1	
1		1		

مستقلة، وفق ما هو مبين في الشكل التالي:

يقال عن مثل هذا التابع إنه غير قابل

للاختصار، وسيبقى بثلاثة حدود، وفي كل

حد سيكون ثلاثة متحولات وفق الصيغة:

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

بالطبع عندما تكون جميع خلايا المخطط تضم وحدات، وهي حالة خاصة جداً، يمكن تغطية كامل المخطط بحلقة واحدة كما هو مبين بالشكل أدناه

a \ bc	00	01	11	10
0	1	1	1	1
1	1	1	1	1

حلقة واحدة

تلغى في هذه الحالة كل متحولات التابع وتكون نتيجة التابع المقابل هي واحد مهما يكن عدد متحولاته، أي في هذا المثال $F(a, b, c) = 1$. يمكن بأسلوب مماثل لما سبق، عرض بعض الأمثلة عن توابع منطقية بأربعة متحولات.

مثال: أوجد أبسط شكل ممكن لتابع منطقي بأربعة متحولات معطى بالشكل الجامع

$$F = \Sigma(0, 2, 4, 6, 8, 9, 10, 12, 13)$$

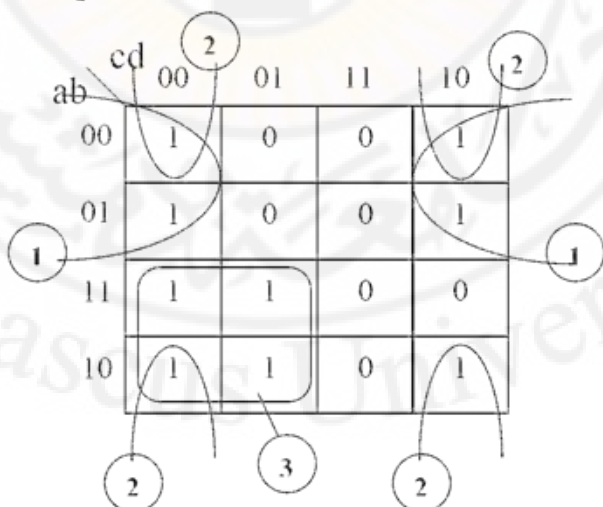
وبالصيغة المبسطة التالية: $F = \Sigma(0, 2, 4, 6, 8, 9, 10, 12, 13)$

الجواب: بتمثيل التابع المعطى بمخطط كارنوف وفق ما هو مبين بالشكل أدناه

ab \ cd	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	1	1	0	0
10	1	1	0	1

وبرسم الحلقات المناسبة كما هو مبين بالشكل 5-10 لكافة الواحدات المتجاورة، حيث إن الواحدات المتواجدة في الزوايا الأربعة للمخطط متجاورة ولذلك توضع جميعها في حلقة واحدة (وكانها نقطة واحدة من كرة مفتوحة). ويصبح الشكل المختصر للتابع المعطى بثلاث حلقات مكون من ثلاث حدود كل منها بمتحولين فقط ، وفق التالي:

$$F(a, b, c, d) = a'd' + b'd' + ac'$$



الشكل 5-10: تمثيل تابع منطقي بأربع متحولات بمخطط كارنوف ورسم الحلقات المناسبة

5-2-2-1 استخدام حالات عدم الاكتراث باختصار التابع المنطقي

تساعد حالات عدم الاكتراث في اختصار التابع المنطقي، علماً أنه عند رسم المخطط المنطقي لتابع ما يضم حالات عدم تعيين لا تظهر الحدود التي تقابل حالة عدم الاكتراث d في المخطط المنطقي. توضع حالة عدم الاكتراث d بقيمة 1 إن كانت ضمن حلقة ما أو يسند لها القيمة 0 إن كانت خارجها وذلك لبناء أقل عدد ممكن من الحلقات وبحجم اكبر ما يمكن للحلقة الواحدة، وهذا يتضح من خلال المثال التالي:

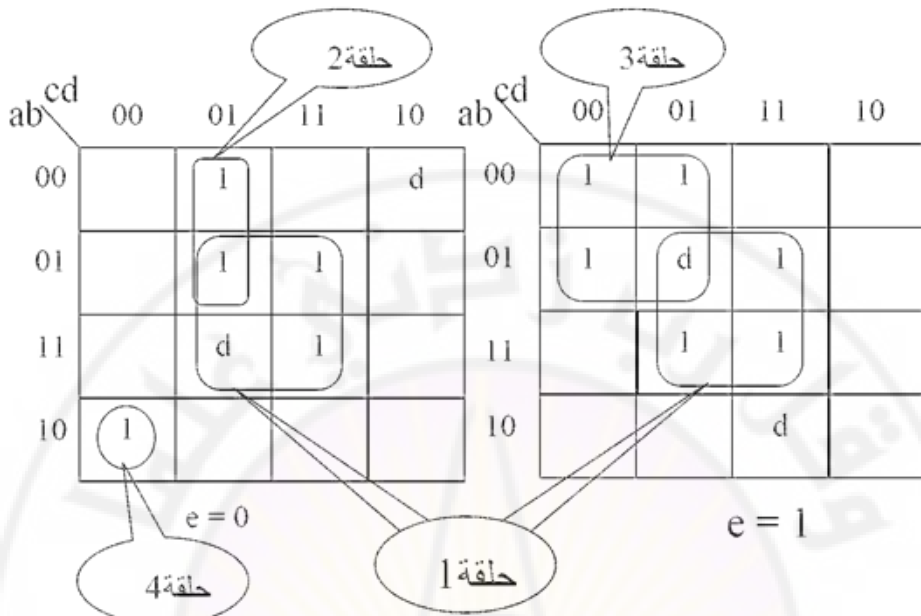
مثال : أوجد أبسط شكل ممكن للتابع المنطقي المعطى بصيغة الشكل الجامع التالية:

$$F(a, b, c, d) = \sum_m(0, 3, 12, 15) + \sum_d(6, 7, 11, 13, 14)$$

الجواب: تحدّد بمخطط كار نوف الوحدات المعطاة بالتابع والد d وفق ما هو مبين في المخطط أدناه.

cd \ ab	00	01	11	10
00	1	0	1	0
01	0		d	d
11	1	d	1	d
10	0	0	d	0

ترسم الحلقات المناسبة التي تغطي بشكل أساسي الوحدات ويراعى وجود الـ d ، بحيث إذا كان هناك بعض حالات عدم الاكتراث d التي يفيد استخدامها باختصار التابع (من حيث زيادة عدد الوحدات المتجاورة بالحلقة الواحدة) تُعدّ قيمتها واحداً، وإن لم تكن تخدم عملية الاختصار بشيء تجعل قيمتها صفراً. ويمكن بهذا المثال الاستفادة من أربعة من الـ d الممثلة في المخطط، بحيث يصبح بالإمكان تشكيل حلقات أكبر (لأنه لا يسمح بوجود أصفار ضمن الحلقة). وتبقى d واحدة لا تخدم بشيء، لذا تجعل قيمتها صفراً. ويصبح أبسط شكل ممكن للتابع وفق هذه الشروط بالصيغة $F = a'b'c'd' + cd + ab$. بالطبع عند رسم المخطط المنطقي المقابل للتابع المعطى قبل الاختصار، سيظهر في المخطط فقط العناصر المنطقية المقابلة للحدود $(a, b, c, d) = \sum_m(0, 3, 12, 15)$ أما العناصر



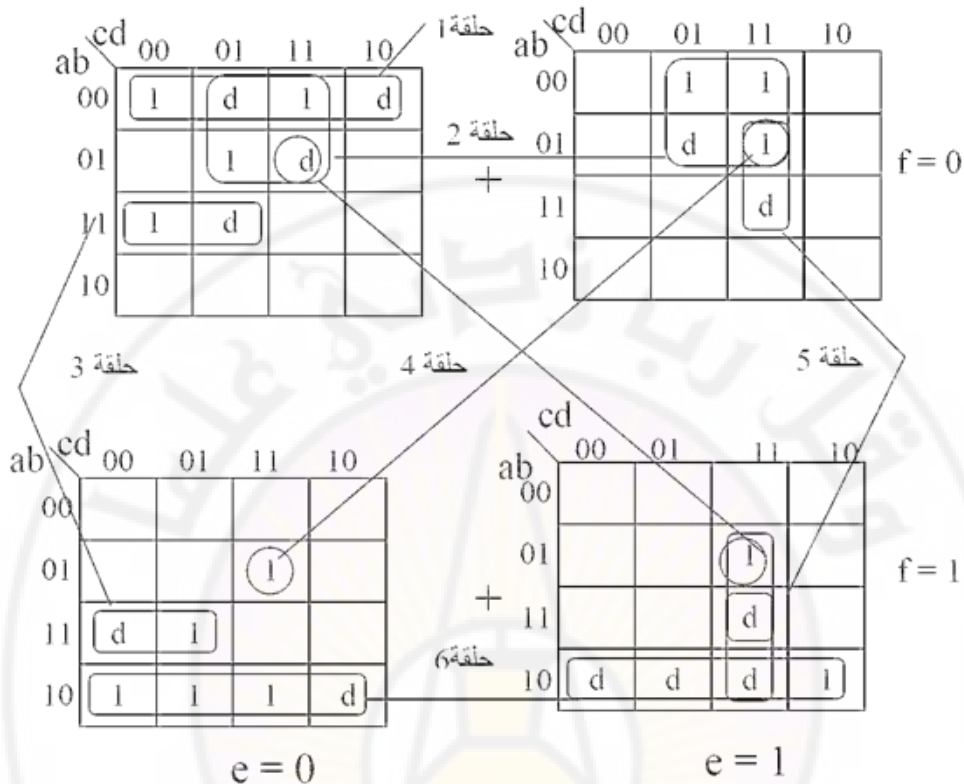
الشكل 5-11: تمثيل تابع منطقي بخمس متحولات بمخطط كارنوف ورسم الحلقات المناسبة

يكون الناتج النهائي: $F = e'ab'c'd' + e'a'c'd + ea'ce + bd$

وأخيراً يمكن الإشارة إلى آلية اختصار تابع منطقي بستة متحولات وبوجود حالات عدم تعين، حيث لا يفضل استخدام كارنوف لتمثيل توابع منطقية بأكثر من ستة متحولات. مثال: أوجد أبسط شكل ممكن للتابع المنطقي المعطى بالصيغة المبسطة التالية:

$$F = \sum_m (0, 3, 5, 12, 17, 19, 23, 39, 40, 41, 43, 45, 55, 58) + \sum_d (1, 2, 7, 13, 15, 21, 42, 44, 56, 57, 59, 63)$$

الجواب: يتضح من دلائل الحالة المعطى بها التابع أنه بستة متحولات، لأن أكبر دليل حالة معطى هو (63) يقابل ستة خانات $(11111)_2$. لتكن هذه المتحولات: a, b, c, d, e, f . يمكن تشكيل الحلقة الواحدة ضمن مخطط واحد من المخططات الأربعة المقابلة للتابع، ويمكن أن تكون بمخططين متقابلين أو في المخططات الأربعة معاً، وفق ما هو مبين في الشكل 5-12. حيث إن المتحول e يمثل عمودياً والمتحول f يمثل أفقياً للتركيبة الأساسية للمتحويلات الأربعة الأساسية a, b, c, d ويكون الشكل المختصر للتابع المعطى هو $F = a'b'e'f' + a'bcd + bcde + abc'e' + a'df' + ab'f$ وهو المطلوب.



الشكل 5-12: تمثيل تابع منطقي بستة متحولات بمخطط كارنوف ورسم الحلقات المناسبة

3-2-5 الطريقة الجدولية

تسمى الطريقة الجدولية في اختصار التوابع المنطقية بطريقة كوين ماكلوسكي Quine Mccluskey Method. ويبدأ اختصار التوابع المنطقية وفق هذه الطريقة بتحديد لائحة بالحدود الصغرى للتابع، ومن ثم يتابع العمل وفق خطوات منتظمة. تكمن الأهمية الرئيسية لهذه الطريقة في إمكانية برمجة خطوات العمل بها بسبب انتظامها إلا أنها طويلة خاصة عندما يكون عدد متحولات التابع كبيراً. ولإستعراض خطوات الاختصار باستخدام هذه الطريقة ننطلق من المثال التالي:

مثال: اختصر التابع المنطقي المعطى بالشكل:

الجدولية (ماكلوسكي) إلى أبسط شكل، مبيناً خطوات الاختصار؟ باستخدام الطريقة

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

الجواب: قبل إستعراض خطوات الاختصار المنتظمة لهذه الطريقة، نحاول بداية إظهار الشكل المختصر للتابع المعطى باستخدام واحدة من طرائق السابقة، لنتمكن لاحقاً من المقارنة بين هذه الطريقة وغيرها من الطرائق الأخرى. فمثلاً، باستخدام طريقة كارنوف، يمكن اعتماد واحد من شكلين لترتيب متحولات الدخل:

- الترتيب abcd وتصبح الصيغة المبسطة للتابع على شكل مجموع جداءات وفق التالي:

$$F(a, b, c, d) = \sum_m (0, 2, 3, 5, 7, 8, 10, 11, 13, 14, 15)$$
 ويكون الشكل

$$F = F(a, b, c, d) = ac + bd + b'd' + cd$$

- الترتيب dcba وتصبح الصيغة المبسطة للتابع المعطى على شكل مجموع جداءات وفق التالي:

$$F(d, c, b, a) = \sum_m (0, 1, 4, 5, 7, 10, 11, 12, 13, 14, 15)$$
 ويكون الشكل المختصر المقابل

$$F = F(d, c, b, a) = ac + bd + b'd' + cb'$$

يمكن اعتماد أي من الترتيبين. وبمراقبة الشكلين يلاحظ ان كافة الحدود بكلا الشكلين متماثلة باستثناء الحد cd من الحالة الأولى والحد cb' من الحالة الثانية، إذ يوجد اختلاف، ولكن في حقيقة الأمر ان كلا الشكلين متكافئان، لأنه لو تم من الشكل الأول إستبدال الترتيب dcba بالمقابلات لها في الشكل الثاني abcd ، أي تعويض كل رمز a في الشكل الأول بالرمز المقابل له وهو d (سواء في الشكل المباشر أو المنفي) و b بـ c و c بـ b و d بـ a يمكن الحصول على نفس النتيجة التي تمثل الشكل المختصر للتابع. يمكن الحصول على نفس النتيجة باستخدام الطريقة الجديدة بإتباع الخطوات التالية: الخطوة الأولى: يحول التابع المعطى إلى صورة مجموع حدود أصغريه. سبق وبيننا ذلك سواء بالترتيب بالترتيب abcd أو الترتيب dcba.

الخطوة الثانية: توزع الحدود الصغرى في مجموعات وفق عدد الواحدات الموجودة في كل منها وترتب في جدول بشكل تصاعدي بحسب عدد الواحدات، بحيث الحدود التي يحتوي كل منها على واحد فقط تشكل مجموعة مستقلة والحدود التي تحتوي على واحدین تشكل مجموعة أخرى تأتي بعد السابقة بالترتيب وهكذا حتى نهاية الجدول (m_0 سينفرد بمجموعة لأنه الحد الوحيد الذي لا يحتوي على واحدات).

الخطوة الثالثة: يظهر من الجدول أدناه شكل توزيع الحدود الصغرى للتابع المعطى على أربع مجموعات. وتبدأ هذه الخطوة بعملية مقارنة الحدود الصغرى الموجودة في المجموعات المتجاورة فقط، بدمج الحدين المختلفين بمتحول واحد $ab + ab' = a$. توضع إشارة (✓) مقابل كل من الحدين اللذين تم دمجهما وينقل الحد الجديد إلى العمود المجاور في الجدول كما هو مبين بالعمود الثاني.

العمود الثالث	العمود الثاني	العمود الأول	مج
0-0 → 0,2,8,10	0 ✓, 00-0 2	0 ✓ 0000	0
0-0 → 0,8,2,10	0 ✓, 000- 8	0 ✓ 0010	1
تُلغى لأنها مكررة	2 ✓, 001- 3	8 ✓ 1000	
	2 ✓, 010- 10	0 ✓ 0011	2
	8 ✓, 10-0 10	5 ✓ 0101	
		10 ✓ 1010	
01- → 2,3,10,11	3 ✓, 0-11 7	0 ✓ 0111	3
01- → 2,10,3,11	3 ✓, 011- 11	0 ✓ 1011	
تُلغى	5 ✓, 01-1 7	13 ✓ 1101	
	5 ✓, 101- 13	14 ✓ 1110	
	10 ✓, 101- 11	15 ✓ 1111	4
11- → 3,7,11,15	10 ✓, 1-10 14		
تُلغى 11- → 3,11,7,15	7 ✓, 111- 15		
1-1 → 5,7,13,15	11 ✓, 1-11 15		
تُلغى 1-1 → 5,13,7,15	13 ✓, 11-1 15		
1-1 → 10,11,14,15	14 ✓, 111- 15		
تُلغى 1-1 → 10,14,11,15			

بعد الانتهاء من مقارنة مجموعات الحدود في عمود ما، يتم الانتقال لمقارنتها في العمود المجاور الذي نتج عن عملية المقارنة في العمود الذي يسبقه. تكرر العملية السابقة إلى حين ظهور آخر عدد لا يحتوي على حدود أصغرية مختلفة في متحول واحد. عند تكرار

ظهور حد ما في أحد الأعمدة ، يترك واحداً منها ويهمل الباقي بحسب الترتيب تصاعدياً. وبعد الانتهاء من المقارنة فإن الحدود في العمود الأخير التي لا يوجد بجوارها الإشارة (√) تمثل الحدود الرئيسية (التي لا يمكن حذف أي من حروفها). وبالتالي الحدود الرئيسية التي يمكن ان تظهر في الصورة المبسطة للتابع. يلاحظ من الجدول ان الحدود الرئيسية

$$(\text{عددها } 5) \text{ هي: } 0, 2, 8, 10 \rightarrow -0 - 0 = a'c'$$

$$2, 3, 10, 11 \rightarrow -01 - = bc'$$

$$3, 7, 11, 15 \rightarrow - - 11 = ab$$

$$5, 7, 13, 15 \rightarrow -1 - 1 = ac$$

$$10, 11, 14, 15 \rightarrow 1 - 1 = bd$$

أي ان: $F(a, b, c, d) = a'c' + bc' + ab + ac + bd$ ، لكن هذه النتيجة لا تمثل أبسط شكل لوجود حدود فائضة. الخطوة الرابعة : وفيها يتم التخلص من الحدود الفائضة (مثلاً، يوجد بين الحدين $a'c'$ و ab حد فائض هو bc') بوضع الحدود الرئيسية في خارطة الحدود الرئيسية عمودياً وأرقام الحدود الصغرى أفقياً وفق ما هو مبين في الشكل أدناه. يوضع مقابل نقاط تقاطع كل من الحدود الرئيسية وأرقام الحدود الصغرى التي يشملها الحد الرمز x .

	0	2	3	5	7	8	10	11	13	14	15
0											
2											
3											
5											
7											
8											
10											
11											
13											
14											
15											

الخطوة الخامسة: تحدد الحدود الرئيسية الأكيدة التي لابد من ظهورها في الصورة المبسطة للتابع وإيجاد مثل هذه الحدود على الخارطة يتم بسهولة ، حيث توضع دائرة على الإشارات x التي توجد بمفردها في العمود. وهذا يعني ان مثل هذه الحدود، يمكن شملها بحد واحد فقط هو الحد الموجود في نفس السطر الموجودة به x .

الخطوة السادسة: ترسم خطوطاً أفقية تصل بين الدوائر والحدود المقابلة لها ثم ترسم أعمدة من كافة إشارات x . بالتالي يكتب من كل سطر حداً واحداً وهو يغطي كافة الحدود الصغرى الموجودة فيه، هذا بالنسبة للأسطر التي يوجد بها إشارة x ضمن دائرة. أما الخطوط التي لا يوجد بها إشارة x ضمن دائرة، يرسم واحداً منها فقط، الذي فيه أكبر عدد من الحدود الصغرى، لذلك تم رسم الخط الذي يغطي bc' و لم يرسم الخط الذي يغطي ab ، حيث كان يمكن أخذ ذلك أيضاً. وهكذا يكون الشكل المختصر للتابع:

$$F(a, b, c, d) = a'c' + bc' + ac + bd$$

فيه ثلاثة حدود رئيسية أكيدة وحد رئيسي غير أكيد، لأنه كان يمكن ان يؤخذ الحد ab . أي ان الحدود الرئيسية الأكيدة تشارك في جميع صور التابع، بينما الحدود غير الأكيدة يشارك بعضها في أحد الصور والبعض الآخر في صور أخرى ولكن في نهاية الأمر فان الدارة المنطقية الناتجة عن تصميم أي منها تكافئ الدوائر الأخرى وهكذا الحدود الفائضة تبقى من الحدود الرئيسية غير الأكيدة وقد تم أخذ السطر الذي يضم bc' وليس ab لأن كلاً من هذين الحدين متساو في عدد الحروف. وبمقارنة هذه النتيجة مع النتيجة التي تم التوصل إليها وفق كارنوف يلاحظ انها متطابقة والفرق هو انه يمكن بهذه الطريقة تحديد الحدود الأكيدة والحدود الرئيسية غير الأكيدة، الشيء الذي لا يمكن فعله وفق كارنوف. ونشير أخيراً إلى وجود طرق أخرى لاختصار التوابع المنطقية غير الطرق المشار إليها أعلاه، لكن نكتفي بما تم عرضه، ولن ندخل بتفاصيل أي طريقة أخرى، لأن جميع الطرق الأخرى مثل طريقة المضامين الأولية Prime Implicants الموصفة أيضاً بالنسخة الأولى لهذا الكتاب (نسخة 2008) تعد أقل استخداماً من طريقة كارنوف.

5-3 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

1- لتمثيل تابع منطقي بثلاثة متحولات من الشكل $f(a, b, c)$ بمخطط كارنوف يلزم

- a. map with 4 squares b. map with 6 squares
c. map with 8 squares d. map with 16 squares

2- مخطط كارنوف بـ 32 خلية يكفي لتمثيل تابع منطقي بـ

- a. 6 variables b. 4 variables
c. 5 variables d. none of all

3- أي من الخلايا الواردة أدناه في مخطط كارنوف متجاورة مع الخلية بالدليل 0100 ؟

- a. 1100 b. 0101 c. 0110 d. all the above

4- عدد المتحولات التي يمكن اختصارها من حلقة بـ 16 cells هو

- a. 3 Variables b. 4 Variables c. 8 Variables d. none of all

5- عدد الوحدات التي يمكن وضعها بحلقة واحدة لتابع بثلاثة متحولات ممثل بمخطط

- كارنوف يمكن أن يكون: a- 3 b- 1 c- 6 d- none of all

ثانياً: أسئلة متفرقة

1- بسط تابع منطقي معطى بالشكل $F = \sum_m(0, 2, 4, 5, 6, 7, 8, 10, 13, 15)$

2- بسط توابع منطقية إذا كانت معطاة بالشكل:

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

$$b) F = a'b'ce' + a'b'c'd' + b'd'e' + b'cd' + cde' + bde'$$

3- أوجد الحدود الأصغرية للتوابع التالية باستخدام مخطط كارنوف.

$$F = c'd + abc' + abd' + a'b'd$$

4- ليكن معطى تابع منطقي في الشكل $F(a, b, c, d) = a'c + ac'd'$. إذا علمت أن

التركيبية $a = c = 1$ لا يمكن أن تحدث أبداً على مدخل الدارة المحققة للتابع، يطلب تمثيل

التابع بمخطط كارنوف.

5- اختصر تابعاً منطقياً معطى بالشكل المختصر التالي:

$$F(a, b, c, d) = \sum_m(1, 3, 7, 11, 13, 15) + \sum_d(0, 4, 8, 12, 14)$$

الفصل السادس: التنجيز الفيزيائي للعناصر المنطقية

تُصنع الدارات الرقمية Digital Circuits وفق تقانات Technologies مختلفة على شكل دارات متكاملة. سيتم بهذا الفصل التعرف على أهم تقانات تصنيع العناصر المنطقية المكونة للدارات الرقمية وآلية تشكيلها من المكونات الإلكترونية الأساسية. يرمز للعناصر المنطقية برموز خاصة، تمثل أشكالاً رمزية للتعبير عن وظيفة محدّدة للعنصر المنطقي مثل الرموز المبينة في الشكل 1-6.

			
للعنصر OR	والرمز NOT للعنصر	الرمز	

الشكل 1-6: رموز منطقية لبعض العناصر المنطقية

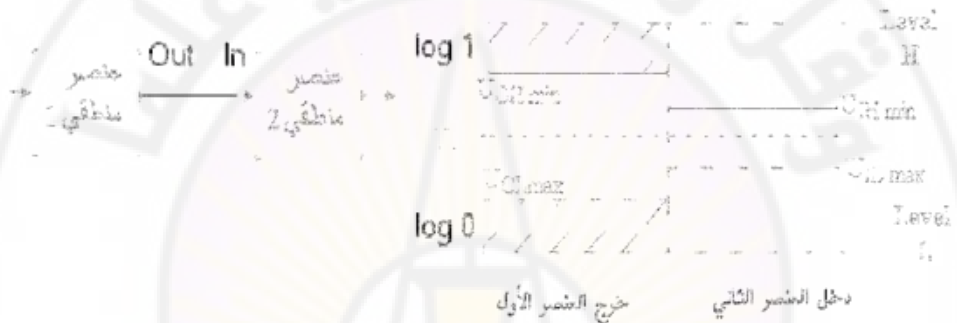
بمعنى أنه سيتم التركيز على إظهار البنية الداخلية لكل من هذه الرموز، وآلية تحقيقها من الناحية الفيزيائية لفهم الآلية التي يتحقق بها أي تابع منطقي. ويمكن أن تكون العناصر المنطقية محققة إما على شكل مفاتيح وصل مثل الري ليه أو على شكل مفاتيح من دون اتصال، كما هو الحال بالنسبة للعناصر نصف الناقل Semiconductors.

1-6 الدارات التلامسية Contact's Logic Circuits

تحقق (تنفذ) العناصر المنطقية التلامسية (أي العناصر الاتصالية التي تعتمد مبدأ مفاتيح التلامس) عادة من الريليات الكهرومغناطيسية بتركيبتها المختلفة. والمواصفة المميزة للريليه هي الفصل (العزل) الكهربائي لدارة متحولات الدخل عن متحولات الخرج. تستخدم تقانات الريليات في تصنيع العناصر المستخدمة في أنظمة الاتصالات الهاتفية، حيث يطلب تحقق معدل وصل عال يصل إلى 10^{13} (يقصد بمعدل الوصل النسبة بين ممانعات مأخذ الوصل ما بين حالتَي الوصل والفصل). يصل معدل الوصل بالنسبة لأنصاف النواقل إلى ما يقارب 10^7 فقط. وبالنظر لعدم انتشار هذه التقنية برسم العناصر والشبكات المنطقية بالوقت الحاضر، تكفي هذه الإشارة لوجود مثل هذه التقنية من دون الدخول بأية تفاصيل أخرى حولها، علماً أنها كانت تستخدم بتشكيل العناصر المنطقية.

6-2 شروط ربط العناصر المنطقية

تتكون العناصر المنطقية من عناصر تعتمد في عملها المبادئ الإلكترونية (يوجد عناصر أخرى مبنية على مبادئ كهر وضوئية أو طبية وغيرها). تربط العناصر المنطقية مع بعضها لتشكل دارات متكاملة لوظيفة محددة. تستخدم عملية الربط المتبادل للعناصر المنطقية المواصفات العامة لعناصر الوصل نصف الناقلية. ويمكن توضيح آلية ربط عنصرين منطقيين وشروط تعاملهما معاً وفق ما هو مبين في الشكل 6-2.



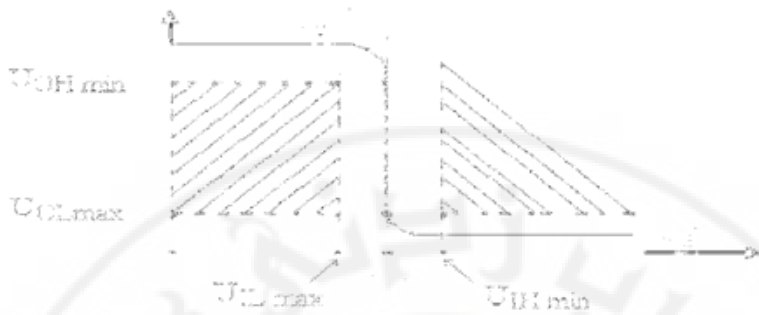
الشكل 6-2: شروط ربط عنصرين منطقيين

يرمز بمخطط المستويات للعناصر المنطقية وفق الشكل أعلاه إلى كافة قيم (نطاق) $\log 0$ بالمستوى L وإلى كافة قيم $\log 1$ بالمستوى H. وبذلك الرمز U_{OHmin} في المخطط يرمز إلى أصغر قيمة لجهد الواحد منطقي على خرج العنصر الأول، وهكذا تقرأ بقية الرموز في المخطط أعلاه. أما U_I فيمثل جهد الاتزان أو الفصل بين المستويين (0 و 1)، ويختلف عادة عرض هذا النطاق من عنصر إلى آخر ويسمى نطاق الانتقال. وبالتالي لربط عنصرين منطقيين (متماتلين، مختلفين) يفترض تحقيق شرطين أساسيين هما:

الأول- أن تحقق مداخل ومخارج العنصرين المتراجحتين التاليتين: $U_{OHmin} > U_{IHmin}$

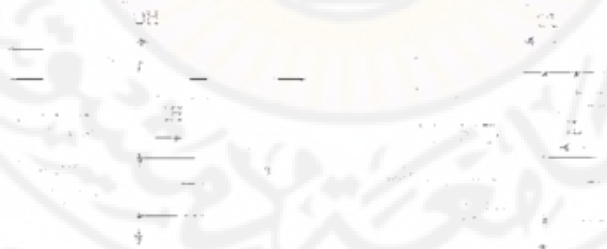
$U_{OLmax} < U_{ILmax}$

يمكن فهم المتراجحتين من خلال مميزة العبور الأساسية لعنصر منطقي مثل العاكس NOT المبينة في الشكل 6-3، بحيث في أقصى حدود الإنتاج، يجب أن لا تصل هذه المميزة إلى المناطق المهيشرة.



الشكل 3-6: مميزة العبور للعنصر المنطقي NOT

الثاني - عدم تجاوز حمل الخرج الأعظمي أو قدرة الخرج على الإمداد Fan-out، يعرف الحمل الاعظمي القياسي بكمية التيار الذي تحتاجه مدخل بوابة أخرى من نفس العائلة. تحدّد قدرة الخرج على الإمداد عدد الأحمال القياسية التي يمكن وصلها إلى مخرج بوابة من دون أن يؤثر ذلك على عملها النظامي. يستخدم أحياناً مفهوم التحميل Loading بدلاً من قدرة الخرج على الإمداد (التفرع الخرجي). يمكن توضيح مفهوم قدرة الخرج على الإمداد عند وصل مخرج بوابة مثل NAND إلى مداخل عدة بوابات أخرى وفق ما هو مبين في الشكل 4-6.



الشكل 4-6: قدرة المخرج على الإمداد (Fan-out)

نميز بين حالتين:

الأولى: إذا كان خرج البوابة بالمستوى "1" سيكون بمثابة منبع تيار Current source لكل مداخل البوابات المنطقية الأخرى المربوطة به.

الثانية: إذا كان خرج البوابة بالمستوى المنخفض " 0 " سيكون بمثابة مصرف للتيار Current sink لكل مداخل البوابات المنطقية الأخرى المربوطة به.

في حال تجاوز قيمة الخرجية لأي بوابة يصبح عملها غير مضمون النتائج، لأنها تصبح محملة بأكثر مما هو مسموح به Overloaded. ويستهلك كل مدخل وفق الشكل 4-6 آ تيار I_{IH} من التيار الخارج من مخرج البوابة I_{OH} . بينما وفق الشكل 4-6 ب يقدم كل مدخل من مداخل البوابات الموصلة تياراً I_{IL} للتيار الذي تتحمله البوابة I_{OL} . وتحسب عادة قدرة مخرج البوابة على الإمداد بأخذ الأصغر من النسبتين I_{OL} / I_{IL} أو I_{OH} / I_{IH} ، حيث قيمة قدرة الخرج على الإمداد بالنسبة لدارات TTL مثلاً هي 10) وهذا يعني أن عدد المداخل التي يمكن وصلها إلى مخرج البوابة الواحدة بعائلة TTL يجب ألا يزيد عن عشرة مداخل لبوابات أخرى من نفس العائلة). أما قدرة الدخل على التحميل Fan-in فتعطي بعدد المداخل المتوافرة بالبوابة. يوجد شروط أخرى لتعامل عنصرين منطقيين مع بعضهما تتعلق بتقانات تصنيع العناصر كل منهما.

3-6 خصائص العناصر المنطقية

بالنظر إلى صعوبة الوصول إلى بعض العناصر المنطقية، ولأسيما العناصر المتكاملة ضمن دارات محددة لقياس خصائصها، يتم التركيز فقط على دراسة الخصائص الديناميكية للعناصر المنطقية، والتي تعرف بشكل مختلف عما هو معروف لدى العناصر الإلكترونية الأساسية مثل الترانزستور، لأن العنصر المنطقي قد يكون مكوناً من عدة ترانزستورات بتوصيلات مختلفة وأنواع مختلفة. وبما أن الأساس الذي تبنى بواسطته جميع المكونات الأساسية لأي دارة رقمية هو الترانزستور، سيتم فيما يلي قبل المباشرة بتناول الخصائص العامة للعناصر المنطقية عرض أهم المفاهيم العائدة للترانزستور من وجهة نظر تحليل الدارات الرقمية. يصادف في الدارات الرقمية نوعان أساسيان من الترانزستورات: - الترانزستور ثنائي القطبية BJT (Bipolar Junction Transistor) الذي يبنى من مواد نصف ناقلة (الجرمانيوم أو السيلكون)، ويعتمد عمله على نوعين من الحوامل الإلكترونية (الإلكترونات والثقوب)، حيث يكون بنوعين هما: PNP و NPN.

- الترانزستور أحادي القطبية Unipolar، ويسمى بالترانزستور ذي الأثر الحقل FET (Field-Effect Transistor). ويعتمد عمله على نوع واحد من الحوامل الأكثرية، حيث إن ترانزستورات أحادي القطبية تصنف بدورها بمجموعتين أساسيتين، هما:

أ- ترانزستور نصف الناقل معدن - أوكسيد (MOS).

- ترانزستور الأثر الحقل ذو الوصلة Junction FET (JFET).

تسمى مجموعة الدارات المشكلة من ترانزستورات BJT بدارات ثنائي القطبية Bipolar. وتسمى مجموعة الدارات المشكلة من ترانزستورات FET بدارات ثنائي القطبية Bipolar. بما أن ترانزستورات JFET تستخدم في الدارات الخطية، ستقتصر دراستنا للدارات الرقمية على استخدام ترانزستورات من نوع MOS فقط. يمكن الحصول على المعطيات الأساسية حول الترانزستور التي تلزم لتحليل الدارات الرقمية من منحنيات الخواص لوصلة باعث مشترك Common-Emitter. فمثلاً، بالنسبة لترانزستورات BJT، إذا كان U_{BE} أقل من 0.6V فإن الترانزستور يكون في حالة قطع Cut off وبهذه الحالة لا يمر أي تيار في القاعدة (كما أن التيار الذي يمر بالمجمع يكون مهملًا. وبالتالي تسلك دائرة مجمع - باعث سلوك الدارة المفتوحة). أما عندما تكون وصلة القاعدة - باعث منحازة أمامياً (أي أن $U_{BE} > 0.6V$) يصبح الترانزستور بحالة توصيل ويبدأ تيار القاعدة بالتزايد بسرعة كبيرة مقابل تغير قليل بالجهد U_{BE} الذي نادراً ما يزيد عن 0.8V. نكتفي هنا بهذه الإشارة المختصرة حول ترانزستورات BJT. ونستعرض فيما يلي مفهوم أهم البارامترات Parameters التي تعكس خصائص العنصر المنطقي من تأخير انتشار الإشارة الذي يختلف بين عنصر وآخر، وبعض الخصائص الأخرى، وفق التالي:

أولاً- تأخير الانتشار Propagation delay

تأخذ عادة الإشارات التي تنتشر من مداخل بوابة منطقية إلى مخرجها مدة معينة، تعرف بتأخير الانتشار للبوابة وتقاس بالنانو ثانية (حيث $1 \text{ ns} = 10^{-9} \text{ s}$). وأزمنة تأخير الانتشار للبوابات والدارات المنطقية تعطى عادة من الشركة المصنعة للدارة بكتاب المعطيات الخاص بالشركة. ويحسب زمن تأخير الانتشار الوسطي لبوابة منطقية من شكل موجتي

الدخل والخرج للبوابه. وهنا نشير إلى إن الشكل العام للإشارة الرقمية التي سيتم التعامل معها والتي تسمى نبضة Pulse ليست مربعة أو مستطيلة تماماً، وإنما حوافها تأخذ بعض الميلان وفق ما هو مبين في الشكل 5-6. أي أن زمن انتقال الإشارة من مستوى لآخر (من المستوى 1 إلى 2 أو من المستوى 3 إلى 4) على الرغم من قصره بالمقارنة مع زمن استمرار النبضة في المستوى (0 أو 1) لا يهمل في بعض التطبيقات. كما أن تأخير الانتشار يختلف بين بوابة وأخرى باختلاف تقانة التصنيع المستخدمة للبوابه.

الشكل 5-6: شكل النبضة وزمن تأخير انتشار الإشارة بالبوابه NOT

يرمز لزمن انتقال الإشارة من المستوى العالي إلى المستوى المنخفض بـ t_{pHL} ويقاس عادة عند القيمة 50% من الوجه الأمامي للنبضة Leading edge، وبالعكس يرمز للانتقال من المستوى المنخفض إلى المستوى العالي بـ t_{pLH} . ليس بالضرورة أن يساوي زمن صعود الإشارة Rise time زمن هبوط الإشارة Decay time، وبالتالي لحساب الزمن الوسطي لتأخير الإشارة ببوابه مثل NOT يؤخذ وسطى الزمنين t_{pHL} و t_{pLH} المبيينين بالميزه الفولطية للبوابه وفق ما هو مبين في الشكل 5-6 ب، أي أن: $t_p = (t_{pHL} - t_{pLH}) / 2$. وهكذا أصبح بالإمكان تصور الإشارات التي تمر بدارة منطقية من مداخلها وصولاً إلى مخارجها بأنها قد تمر عبر سلسلة من البوابات المنطقية التي قد تكون موزعة على أكثر من مستوى، وأن زمن التأخير الكلي للدارة سيكون بهذه الحالة مساوياً إلى مجموع أزمنة التأخير للبوابات الموجودة بأطول فرع في الدارة.

ثانياً - تبديد القدرة Power dissipation

تتطلب كل دائرة إلكترونية كمية محددة من القدرة (الاستطاعة) لكي تعمل بشكل صحيح. والاستطاعة المبددة هي باراميتير (متوسط) Parameter يمثل كمية الاستطاعة التي تحتاجها البوابة. ولحساب هذه الاستطاعة، نشير بداية إلى أن التيار المستجّر من وحدة التغذية الرئيسية لبوابة منطقية يرمز له بالرمز I_{CCH} عندما يكون خرج البوابة بالمستوى العالي للجهد، و I_{CCL} عندما يكون خرج البوابة بالمستوى المنخفض للجهد. يرمز لمتوسط التيار بالرمز $I_{CC} (avg)$ ، حيث: $I_{CC} (avg) = (I_{CCH} + I_{CCL}) / 2$. وبالتالي متوسط الاستطاعة المبددة يعطى بالعلاقة: $P (avg) = I_{CC} (avg) \times U_{CC}$. وهكذا عند تصميم وحدة تغذية لمنظومة تضم عدة دوائر منطقية، يجب الأخذ بالحسبان أن الاستطاعة المبددة اللازم أخذها بعين الاعتبار تمثل المجموع الكلي للاستطاعة المبددة بجميع الدوائر المستخدمة بالمنظومة

ثالثاً - هامش الضجيج Noise margin

يستخدم مصطلح الضجيج Noise للدلالة على إشارة خارجية غير مرغوب فيها تتراكب على الإشارة الفعلية. ويجب بالنسبة للدوائر المنطقية أخذ نوعين من الضجيج بالحسبان - ضجيج التيار المستمر DC Noise الذي ينتج عن تغير مستوى جهد الإشارة. - ضجيج التيار المتناوب AC Noise الذي ينتج من تغيرات إشارات أخرى. يمثل هامش الضجيج Noise margin الجهد الأعظمي للضجيج الذي يضاف إلى إشارة دخل دائرة منطقية من دون أن يؤثر بخرج الدارة بشكل غير مرغوب فيه. وباستخدام الرموز المبينة أعلاه لجهد الدخل والخرج لبوابة منطقية بالمستويين العالي والمنخفض، يحسب هامش الضجيج من معرفة إشارة الجهد على خرج البوابة وإشارة الجهد المطلوبة في مدخل البوابة، أي بأخذ الأصغر من الفرقين: $U_{IL} - U_{OL}$ أو $U_{OH} - U_{IH}$. وهو يقاس بالفولط. للتخلص من الضجيج غير المرغوب به في الدوائر الرقمية يستخدم العديد من الفلاتر Filters التي نستغني هنا الدخول بالتفاصيل حولها، لأنها من خارج سياق هذا الكتاب.

4-6 تقانات تشكيل العناصر المنطقية

يوجد العديد من تقانات التصنيع التي تتفاوت فيما بينها من حيث الحجم والسرعة والاستطاعة. سنحاول فيما يلي إظهار تركيبة بعض التقانات وتركيبه بعض البوابات المنطقية المشكلة بها مع بعض المواصفات الفنية المميزة لكل تقانة على حدة ابتداء من التقانات الأقدم. تصنيف التقانات التي سيتم تناولها بثلاث مجموعات وفق التالي:

أولاً- التقانتان DL و DTL

تُعدّ التقانتان DL (Diode Logic) و DTL (Diode Transistor Logic) من أقدم التقانات، خاصة كون كل منهما يعتمد على الديودات في تشكيل العناصر المنطقية. تمثل التقانتان DL و DTL البداية لمجمل التقانات المعتمدة على أنصاف النواقل. سيتم التعرف بهاتين التقانتين على الرغم من قدمهما، لأنهما تمثلان نقطة بداية مفيدة في تفسير العمليات الأساسية التي تتم بالبوابات المنطقية المشكلة بتقانات أخرى أكثر استخداماً، حيث يمكن تعميم المفاهيم المعتمدة بهما على كثير من التقانات الأخرى.

1- العناصر المنطقية بالتقانة (العائلة) DL

يمكن وفق المنطق DL تصنيع بعض العناصر المنطقية مثل AND و OR ، وليس كل العناصر المنطقية، لأنه يصعب بهذا المنطق مثلاً تحقيق العنصر NOT. وهذه هي واحدة من أهم سلبياتها. ويعني المنطق DL أن جميع مكونات الدارة المشكلة للعنصر المنطقي هي مشكلة من الديودات حصراً. فمثلاً، أبسط دارة تقابل العنصر المنطقي OR بمدخلين يمكن تحقيقها بالديودات، وفق ما هو مبين في الشكل 6-6.



الشكل 6-6: بوابة OR بمدخلين وفق المنطق DL

يتضح من الشكل أعلاه انه مثلاً، إذا كان على المدخل a القيمة 1 منطقي وعلى b قيمة صفر منطقي، فإن الجهد المقابل للواحد منطقي على المدخل a سيهبط بكامله على المقاومة R ويمكن قراءة قيمة 1 على الخرج، وكأن: $y = 1 + 0 = 1$. وينفس الأسلوب يمكن التأكد من جميع التراكيب الأخرى للمتحويلين a, b المبينة بجدول الحقيقة العائد لهذا العنصر، حيث الخرج دوماً سيكون واحداً باستثناء الحالة التي يكون فيها على كل من المدخلين a و b قيمة الصفر. بالطبع من أجل عنصر منطقي من نوع OR بثلاثة مداخل أو أكثر يستخدم المبدأ نفسه أيضاً، يكفي أن نزيد عدد أفرع الدارة على الدخل بعدد المتحولات. يمكن وفق المنطق DL تحقيق AND أيضاً وفق ما هو مبين بالشكل 6-7.



الشكل 6-7: بوابة AND بمدخلين بالمنطق DL

وبسبب وجود المقاومة R مع الثنائيات بالتقانة DL، تسمى هذه التقانة أيضاً بالاسم RDL من Resistor Diode Logic.

2- العناصر المنطقية بالتقانة DTL

لتشكيل عنصر منطقي مثل NOT، يلزم وجود ترانزستور، ليقوم بتوصيلة الباعث المشترك بعكس الإشارة وتضخيمها. يمكن إظهار تركيبة عنصر النفي NOT وفق ما هو مبين في الشكل 6-8 والذي يمكن توصيفه كالتالي: يلاحظ من الشكل، أنه عند تطبيق إشارة بالقيمة واحد على مدخل الترانزستور (أي +5V)، سيفتح الترانزستور وما هو موجود على مقاومة الحمل R بدارة الخرج سيهبط إلى الأرض وتصبح قيمة الخرج صفراً، وبالعكس عندما تكون قيمة الدخل a مساوية للصفر، سيصبح الخرج بالواحد لوجود كامل الجهد على الحمل R، لأن الترانزستور يكون بحالة القطع.

الشكل 8-6: تركيبة العنصر NOT وفق التقنية DTL

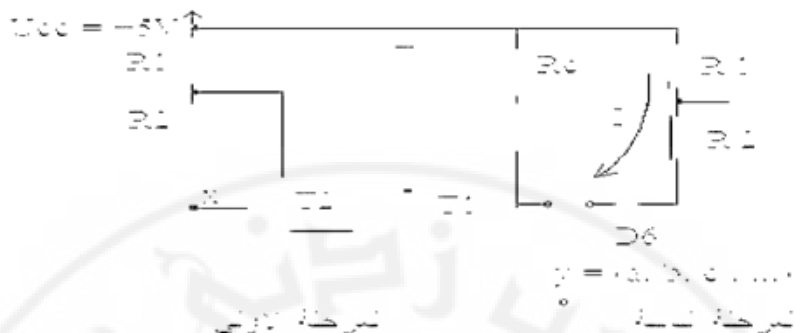
يعد المنطق DTL شكل مطور للمنطق DL. وبالتالي خلال تصنيع العناصر المنطقية بهذه المنطق غالباً ما تدمج الديودات مع الترانزستورات مشكلة منطق الديود - ترانزستور لتحقيق العنصر المنطقي. وتقوم الديودات بالمنطق DTL بأداء التابع المطلوب والترانزستورات بتقوية الإشارة وإتمام قيمتها. وبما أن البوابتين NAND و NOR تمثلان الدارات الأساسية بتركيبة كل دائرة متكاملة من أي عائلة منطقية، سيتم الإشارة بالأمثلة من الآن وصاعداً إلى تركيبة هاتين البوابتين. فمثلاً، تركيبة البوابة NAND بمدخلين سيكون وفق المنطق DTL، كما هو مبين في الشكل 9-6 وهو يقابل التركيبة AND-NOT.

د

الشكل 9-6: تركيبة بوابة NAND بمدخلين وفق المنطق DTL

يتوافق كل مدخل بالمنطق DTL مع ديود خاص فيه. بالتالي الديودات D_1 و D_2 والمقاومة R_1 تشكل عنصر ضرب، أي البوابة AND بالمدخل a, b (طبعاً يمكن زيادة عدد

المداخل بحسب الطلب). أما الترانزيستور T فيمثل العنصر الفعال في هذا المنطق ويقوم بتأمين عملية تقوية الإشارة وإتمامها مشكلاً بذلك نفي الجداء. تمثل الديودات D_4 و D_5 في تركيبة DTL عنصر ربط بين عنصر الضرب وعنصر النفي. تعتمد ديودات الربط لتجنب وقوع أخطاء بسبب عمليات وصل مفاتيح التغذية وفصلها، ولتأمين عمل الدارة بشكل سليم عند التغيرات السريعة لحالات الدخل. توجد المقاومة R التي توصل إلى الأرض للتحكم بإغلاق الترانزيستور أو إلى جهد سالب لامتصاص المركبات السالبة في الجهد إن وجدت. ويمكن تلخيص عمل الدارة في الشكل 6-9 بالتالي: إذا كان أي مدخل من مداخل البوابة بحالة منخفضة (أي بالقيمة صفر حتى 0.2V)، فإن الديود المقابل سيكون بحالة انحياز أمامي يمرر التيار من المنبع U_{cc} عبر R_1 إلى عقدة الدخل بالنقطة x، ويصبح الجهد بالنقطة x مساوياً لجهد الدخل 0.2V مضافاً إليه 0.7V مقدار هبوط الجهد على الديود المقابل. وهذا لا يكفي لجعل الترانزيستور T بحالة توصيل، لأن الجهد المثبت بالنقطة x على القيمة 0.9V بوجود أحد ديودات المداخل بحالة الوصل هو أصغر بكثير من الجهد الموجود على المعبر Base- Emitter بالنسبة لـ T، أي U_{BE} والبالغ 1.8V (بمعدل 0.6V على كل من المعبر B-E وعلى كل من الديودين D_4 و D_5). سيكون بهذه الحالة ترانزيستور الخرج بحالة قطع وكامل جهد الخرج على حمل الخرج R_c وهو يتراوح ما بين 4V إلى 5V الذي يقابل واحداً منطقياً. وتقابل هذه الحالة جميع التراكيب التي يكون فيها واحد على الأقل من المداخل a,b بالمستوى المنخفض. عندما تكون جميع مداخل البوابة بمستوياتها العالية، يكون الجهد على مهبط كل ديود مساوياً للجهد على المصعد وهذا يعني أن الجهد بالنقطة x يصبح مساوياً لمجموع U_{BE} والجهد على الديود بين D_4 و D_5 بقيمة 0.7V لكل منها أو ما مجموعه $2.1V = 3 \times 0.7$. وهو كاف لقيادة الترانزيستور T إلى حالة الإشباع. يكون هبوط الجهد في حالة الإشباع للترانزيستور T بدارة الخرج U_{CE} من رتبة 0.2V، وهو ما يقابل المستوى 0. ويتبين من المراجع العلمية أن الدارة بالمنطق DTL تمتاز بالقيمة (8) لقدرة الخرج على الإمداد. يمكن في التطبيقات المعدلة، زيادة قدرة الخرج على الإمداد إلى (10) بإبدال أحد الديودات المربوطة بقاعدة ترانزيستور الخرج، بترانزيستور موصول بوظيفة ديود، وفق ما هو مبين في الشكل 6-10.



الشكل 6-10: تركيبة NAND بثلاثة مداخل وفق المنطق DTL المعدل

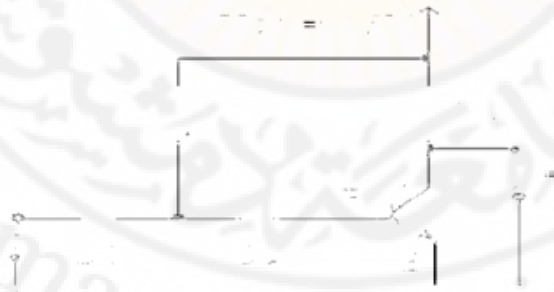
وبما أن الديود في الدارات المتكاملة يتكون عادة من ترانزستور يوصل فيه المجمع إلى القاعدة ، تم كما هو موضح في الشكل أعلاه، استبدال الديود D_1 بترانزستور T_2 (يعمل نفس عمل الديود) للحصول على قدرة أكبر للمخرج على الإمداد. وعندما يفتح T_2 سيعمل في المنطقة الفعالة وليس في منطقة الإشباع بسبب أن التيار المار في R_2 يؤدي إلى انحياز عكسي لوصلة مجمع T_2 (كونه من نوع NPN)، بينما T_1 سيكون في حالة الإشباع. وبما أن تيار باعث T_2 يمثل تيار القاعدة لـ T_1 أي أن تيار قاعدة T_1 أكبر بكثير من تيار قاعدة T_2 ، يؤدي ذلك إلى زيادة تيار المجمع بحيث يصبح أكبر، علاوة على ذلك ترانزستور الخرج من المرحلة الأولى يعمل مصرفاً للتيار القادم من مداخل بوابات المرحلة الثانية والتيار I القادم من المرحلة التالية يضاف إلى تيار مجمع الترانزستور عندما يكون الترانزستور في حالة الإشباع ($y = 0$). وتسمح الزيادة في تيار الإشباع المسموح به للمجمع بزيادة الأحمال التي يمكن وصلها إلى المخرج وبالتالي زيادة قدرة المخرج على الإمداد. وكما بينا التركيبة العامة لبوابة NAND وفق المنطق DTL. نبين فيما يلي التركيبة المبسطة للبوابة الأساسية الثانية NOR وفق هذا المنطق أيضاً في الشكل 6-11 ، حيث يتضح من الشكل أدناه، إذا كان كل من المدخلين a و b بالمستوى المنخفض، سيكون كل من الترانزستورين T_1 و T_2 مغلقاً، وبالتالي يقاس في النقطة x على الحمل R كامل U_{cc} ، أو ما يعادل واحداً منطقياً. مجرد أن يصبح أحد المدخلين بحالة 1، سيفتح الترانزستور المقابل ويهبط ما هو على الحمل R إلى الأرض ويقاس بالنقطة x ما يعادل 0

وهذا ما يقابل التابع NOR



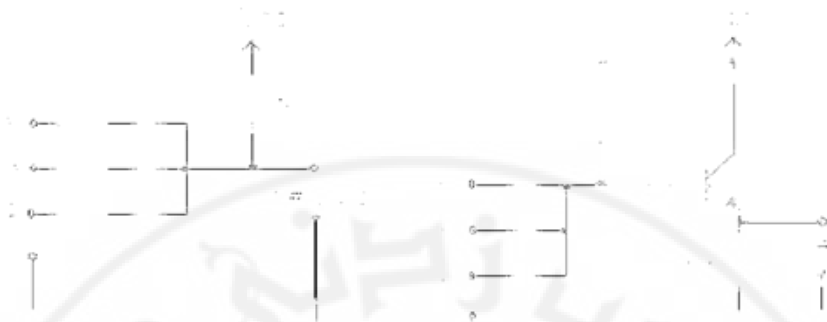
الشكل 6-11: تركيبة NOR وفق المنطق DTL

يمكن تشكيل بوابات المستوى الأول (NOT, AND, OR) من البوابتين NAND و NOR تقنياً وفق التالي: إذا انطلقنا من تركيبة NAND مثلاً وفق المنطق DTL، يمكن تشكيل العنصر NOT وفق ما هو مبين في الشكل 6-12 ومنه يتضح أنه عندما تكون قيمة المدخل a بالمستوى H سيكون D_1 مغلقاً وبهذه الحالة سيمر التيار بالديود D_2 وسيغذي قاعدة T ويجعله في حالة الإشباع، بمعنى أن التيار سيمر عبر وصلة الباعث. ويبقى الجهد على المجمع (الحمل R_c) قريباً من الصفر.



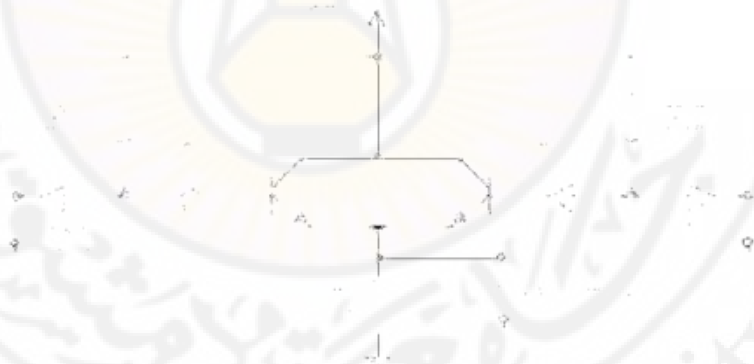
الشكل 6-12: تركيبة البوابة NOT وفق المنطق DTL

أما تركيبة العنصر AND، فيمكن الحصول عليها باستبدال موقع مقاومة الحمل بدارة الخرج من دارة المجمع إلى دارة المشع، وفق ما هو مبين في الشكل 6-13.



الشكل 6-13: تركيبة البوابة AND وفق المنطق DTL

وبالتالي لتصنيع مثل هذه البوابة بالمنطق DTL، يجب دمج الدارة التي تمثل الديودات بعنصر الضرب مع الدارة التي تمثل الترانزيستور باعتباره عاكساً. وهنا نقاس قيمة الخرج على المقاومة R_e بدارة الباعث لتحقيق عنصر AND. يمكن بالمنطق DTL تحقيق التابع OR وذلك بالانطلاق من تركيبة التابع NOR وفق ما هو مبين في الشكل 6-14.



الشكل 6-14: تركيبة البوابة OR وفق المنطق DTL

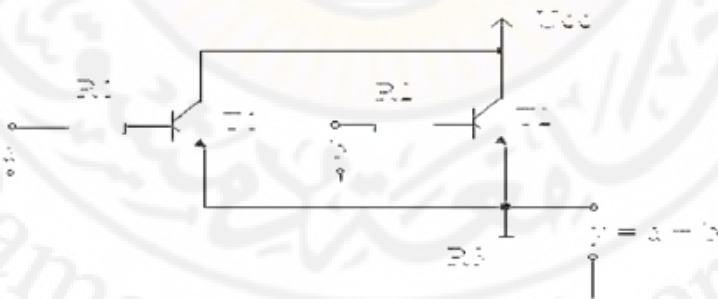
تركيبه البوابة OR بالمنطق DTL مماثلة لتركيبه البوابة NOR في الشكل 6-11 بعد تعديل مكان قياس جهد الخرج، أي أن الشكل نفسه يمكن اعتماده لأكثر من عنصر منطقي (بوابة) وذلك بتغيير مكان قياس الخرج النهائي.

ثانياً- التقانات RTL و DCTL و RCTL و ECL

تمثل العناصر المنطقية المشكلة بهذه التقانات أشكالاً معدلة لتركيبية أساسية مشتركة بين جميع الأنواع المصنعة بهذه المجموعة. وبالنظر لمحدودية انتشار هذه التقانات، نستعرض فيما يلي باختصار تركيبية بعض العناصر المشكلة بها.

1- العناصر المنطقية بالتقانة RTL

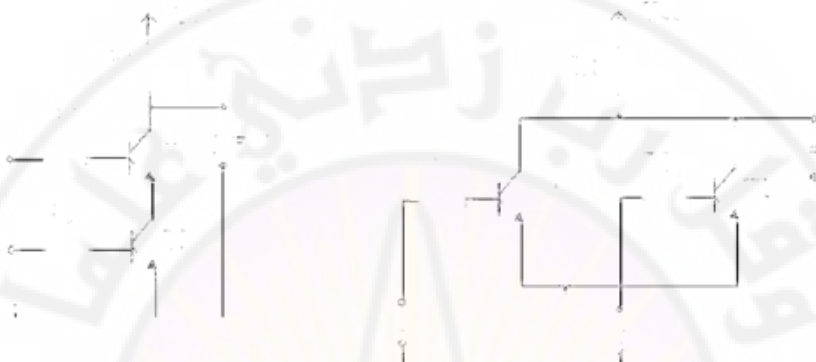
يستخدم منطق مقاومة - ترانزستور (Resistor Transistor Logic) RTL، كما يتضح من اسمه، مقاومات وترانزستورات في تشكيل العناصر المنطقية. وتعد هذه العائلة مع عائلة DTL من أقدم العائلات المعتمدة بتصنيع العناصر المنطقية، وأول العائلات التي استخدمت تجارياً على نطاق واسع. يضاف بالمنطق RTL لمداخل البوابة المنطقية مقاومة على التسلسل مع كل مدخل (على قاعدة الترانزستور) لتعديل ضعف الربح المنطقي. وهذا بعد ذاته يعدّ سلبية من سلبيات هذا المنطق، لأن هذه المقاومة تخفض السرعة العملية، حيث سعة الدخل بين مأخذ الترانزستور يجب أن تشحن وتفريغ عبر هذه المقاومات. يظهر الشكل 6-15 البوابة OR بمدخلين وفق المنطق RTL.



الشكل 6-15: تركيبية البوابة OR بالمنطق RTL

عندما يكون أحد المدخلين a أو b بالمستوى العالي، فإن الترانزستور المقابل من T_1 أو T_2 سيكون في حالة الإشباع، وبالتالي التيار سيعبر من جهته إلى R_3 وستكون النتيجة بدارة الخرج واحداً منطقياً. وعندما يكون كلا المدخلين بالقيمة صفر، يصبح T_1 و T_2 مغلقين،

وبالتالي الجهد على R_3 سيكون 0V، أي أن الخرج $y = 0 + 0 = 0$ ، يمكن بالمنطق RTL أيضاً تحقيق توابع منطقية أخرى. يظهر الشكل 6-16 التركيبة العامة لكل من البوابتين الأساسيتين NAND و NOR بهذا المنطق.



الشكل 6-16: التركيبة العامة لكل من NAND و NOR بالمنطق RTL

2- التفاضلتين DCTL و RCTL

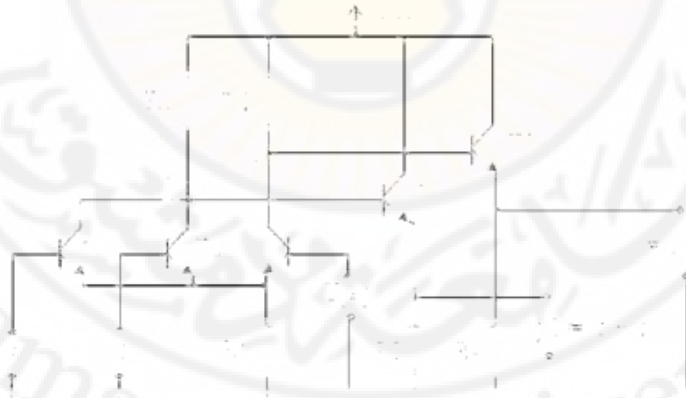
تُعدّ التقانة DCTL (Direct Coupled Transistor Logic) تقانة منطق الترانزستورات المترابطة مباشرة. والتقانة RCTL (Resistor Capacitor Transistor Logic) منطق الترانزستورات بمقاومات ومكثفات من أبسط التقانات المشكّلة بترانزستورات BJT، لأنّ التوابع تتشكل فيهما بالربط المباشر للترانزستورات، وهي تتمثل شكلاً معدّلاً للتقانة RTL. يمثل المنطق RCTL نفس المنطق RTL بوجود مكثفات على الدخل، بحيث المكثفات تربط عادة على التفرع مع مقاومات الدخل بقصد زيادة السرعة (للتخلص من السلبية الرئيسية لل RTL). يظهر الشكل 6-17 التركيبة العامة لبوابة NOR بمدخلين وفق المنطق RCTL. وقد اعتمدت على سبيل المثال التقانة RCTL من قبل Texas Instrument تحت الاختصار SN منذ العام 1961، حيث الاستطاعة قد انخفضت من 12 mW بالتقانة RTL إلى 2 mW بالتقانة RCTL بسبب الإجراء الإضافي لتركيبة RTL باستخدام مكثفات على التوازي مع المقاومات على الدخل.

$$L_{oo} \leftarrow R_o$$

الشكل 6-17: تركيبة البوابة NOR وفق المنطق RCTL

2- تقنية المشعات المترابطة ECL

تُعدّ تقنية المشعات المترابطة ECL - Emitter Coupled Logic من أسرع التقنيات، وقد اعتمدت في تصنيع كثير من الدارات الرقمية من قبل أكثر من شركة، مثل شركات General Electronic و Motorola وغيرها. ويمكن بالمنطق ECL تصنيع مختلف العناصر المنطقية. فمثلاً، الدارة التي تقابل البوابتين OR و NOR معاً بنفس الدارة مبينة في الشكل 6-18.



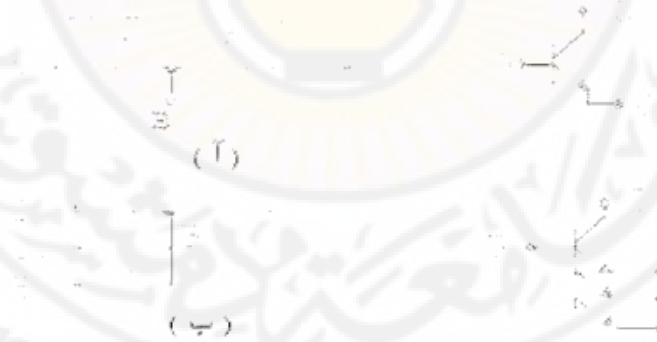
الشكل 6-18: تركيبة البوابتين OR و NOR بالمنطق ECL

تربط بالمنطق ECL مشعات الترانزستورات التي تمثل الدخل مع بعضها البعض. يستخدم المنطق ECL جهدي تغذية. جهد رئيسي $U_{cc} = 5V$ والآخر معياري $U_{ref} = 4V$. وهذه

هي واحدة من أهم سببياته. والجهد المعياري U_{ref} هو الذي يعرف حد التميز بين المستويين المنخفض L والمرتفع H للإشارات الأولية والتي تعرف بهذا المنطق وفق التالي: $4.4V$ للمستوى H و $3.6V$ للمستوى L . يلاحظ من الشكل أعلاه أن الترانزستورين T_1 و T_2 يمثلان المدخلين a و b للبوابة المنطقية. توصل مشعات الترانزستورات T_1 , T_2 مع مشع الترانزستور T_3 إلى مقاومة مشع واحدة Re بالتالي عندما تكون متحولات الدخل a و b بالمستوى L مثلاً، تكون هذه الترانزستورات مغلقة والترانزستور T_3 مفتوح (لأنه يوجد على المشع المشترك الموصل بالأرض سوية جهد لإشارة الدخل أدنى بالقيمة من إشارة القاعدة لترانزستور T_3). وبهذا الشكل يقاس في النقطة z جهداً أقل من $4.4V$ وكان T_5 مغلق، أي على خرج من دائرة الباعث سيكون صفراً، وهذا يقابل التابع المنطقي OR . بينما في النقطة x يقاس كامل U_{cc} ، لأن الترانزستورين T_1 و T_2 مغلقان، وكان T_4 مفتوح، بالتالي يقاس على مقاومة مشعة القيمة $5V$ أو ما يقابل واحداً منطقياً، وهذا ما يقابل خرج التابع NOR . في الحالة المعاكسة عندما يكون أي من الترانزستورين T_1 أو T_2 مفتوحاً سيصبح T_3 مغلقاً، لأنه يوجد على المقاومة Re كامل $U_{cc} = 5V$ ، وهو جهد أعلى من الجهد المعياري المطبق على قاعدة T_3 ، بالتالي يصبح في النقطة z ما يقارب $5V$ أي T_5 يفتح ويصبح الخرج $y_1 = 1$ ، بينما في النقطة x سيكون الجهد أقل من $4.4V$ و T_4 سيغلق ويقاس على المقاومة الموجودة في مشعة القيمة صفر، أي أن $y_2 = 0$. وتوصل الدارة في الشكل 6-18 تيار المشع بحسب جهد الدخل إما من الترانزستور T_3 أو من ترانزستورات الدخل T_1 و T_2 (المقابلة لمتحولات الدخل) وتعطي تابع الخرج في الشكل المتمم أيضاً وهو ما يقابل العنصرين OR و NOR بنفس الدارة. بالطبع قيمة الجهد بالنقطتين x و z بالقيم أقل وأكبر من $4.4V$ (الخرج) لا يمكن استخدامه في تحريض مداخل عنصر لاحق مربوط به، لذلك يوجد في دائرة الخرج للدائرة الأساسية الترانزستورات T_4 و T_5 لتعديل الجهد للمجال المطلوب. وتجدر الإشارة هنا إلى أن المنطق ECL لا يعدّ من عائلات المنطق الرقمي القياسية، لأن الترانزستورات المكونة له لاتصل إلى درجة الإشباع، وبالتالي لهذا المنطق قدرة الوصول إلى تأخيرات انتشار منخفضة من رتبة 5 ns وأقل.

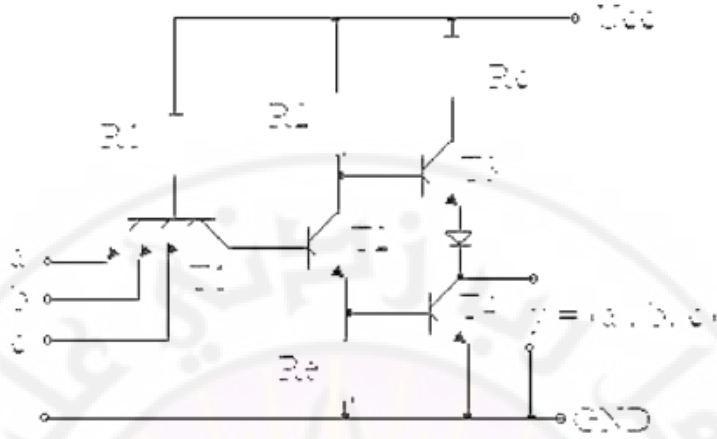
5-6 التفاتة (العائلة) TTL

تتمتع العائلة TTL أو T^2L (Transistor -Transistor Logic) بالمقارنة مع غيرها من تقانات مجموعة ثنائي القطبية بخصائص عدة جعلت منها بكل تراكيبها المختلفة الأوسع انتشاراً. ويمكن تشكيل مختلف التوابع المنطقية الأساسية والمركبة بالمنطق TTL. يُعدّ المنطق TTL شكلاً معدّلاً للمنطق DTL، حيث يستعاض عن الديودات بالمنطق DTL بترانزستورات متعددة البواعث. ولتوضيح مفهوم الترانزستورات متعددة البواعث، نعود للتركيبية الأساسية للترانزستور العادي BJT من النوع NPN، ونذكر باختصار أن هذا النوع من الترانزستورات، كما هو موضح في الشكل 6-19 أ مكون من معبرين هما: الأول NP في دائرة الدخل والثاني PN في دائرة الخرج للترانزستور الواحد موصولين بعكس بعضهما. وبما أن كل معبر PN يمثل ديوداً، يمكن في دائرة الدخل، إضافة معبر (فرع) آخر والحصول على ترانزستور بمعبرين (بمشعين). وهكذا يمكن إلى نفس الترانزستور إضافة عدد من المعابر PN (ديودات) في دائرة الدخل بعدد المداخل ليتم الحصول على ترانزستور متعدد البواعث، كما مبين بالشكل 6-19 ب الذي يمثل الرمز الكهربائي لترانزستور بثلاث مشعات.



الشكل 6-19: الرمز الكهربائي للترانزستور متعدد البواعث

يستخدم المنطق TTL على نطاق واسع في تصميم النظم الرقمية. ولإظهار أولى إصدارات TTL ننطلق من التركيبية الأساسية لبوابة NAND المبينة في الشكل 6-20 التي تُعدّ بنفس الوقت التركيبية القياسية للعائلة TTL.



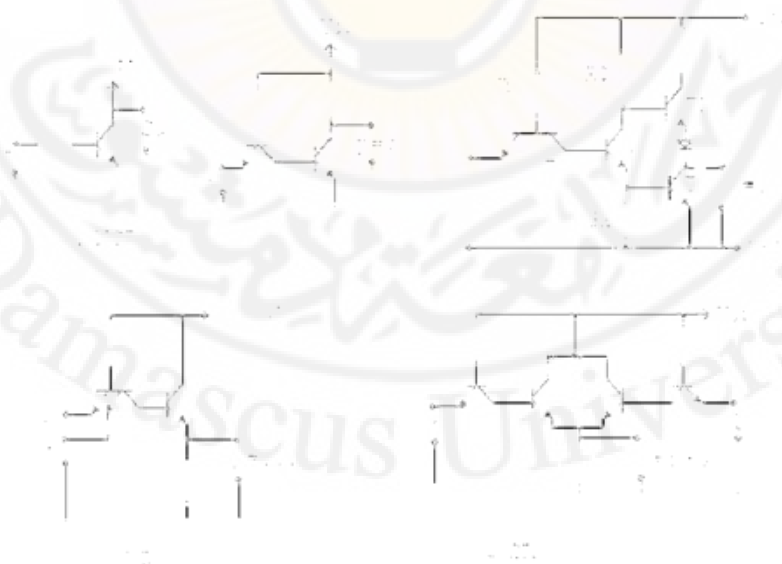
الشكل 20-6: التركيبية القياسية للمنطق TTL (بوابة NAND بثلاثة مداخل)

القيم النموذجية للمقاومات الداخلة بالتركيبية الداخلية للبوابة NAND وفق المنطق TTL القياسي على النحو التالي: $R_1 \sim 4 \text{ K}\Omega$, $R_2 \sim 1,6 \text{ K}\Omega$, $R_c \sim 130 \text{ }\Omega$, $R_e \sim 1 \text{ K}\Omega$. يتلخص مبدأ عمل الدارة المبينة في الشكل أعلاه عن بوابة NAND بثلاثة مداخل أنه عندما تكون المداخل $a = b = c = 1$ فإن الترانزستور T_1 سيكون مغلقاً، وبالتالي سيفتح الترانزستور T_2 . وهذا يؤدي بدوره إلى هبوط جهد التغذية U_{cc} على المقاومة R_e الذي يؤدي بدوره إلى فتح ترانزستور الخرج T_4 ، لكن على مجمع T_4 توجد قيمة جهد أقل من الجهد الذي يعادل الواحد منطقي، لأن الترانزستور T_3 سيكون مغلقاً لعدم وجود جهد كافٍ على قاعدته لجعله بحالة التوصيل. بالتالي يقاس على الخرج جهد يعادل صفراً منطقياً. وعندما يصبح أحد المداخل a, b, c (أو كلها معاً) في حالة الصفر، سيفتح الترانزستور T_1 وبالتالي يغلق الترانزستور T_2 . وعندما يكون T_2 مغلقاً سيفتح الترانزستور T_3 ويغلق T_4 ، ويقاس على مجمع T_4 الجهد الذي يعادل واحد منطقي. والميزات الإيجابية لهذه التقانة عديدة نذكر منها، أن ممانعة خرج البوابة هي حمل مقاومي وسعوي. بمعنى أن ممانعة الخرج في الحالتين L و H منخفضة، وبهذا الشكل يمكن حصر الضجيج بصورة أفضل. كما يمكن الحصول على تحميل سعوي مناسب، علاوة على أنه بتخفيض ممانعة المجمع في دارة الخرج يمكن الحصول على تقصير للثوابت الزمنية لهذا المنطق. تبرز بالمقابل

مشكلة تتمثل بأنه بالمقاومة الصغيرة للمجمع ترتفع الاستطاعة اللازمة للدارة. وانطلاقاً من مطلب الحصول على أقصر ثابت زمني، يطلب أن يكون خلال فتح الترانزيستور T_3 في دارة الخرج ممانعة المجمع عالية وأن تكون الممانعة خلال عملية الإغلاق صغيرة. وكحل وسط ملائم ما بين استهلاك القدرة والتأخير الزمني، تحقق هذه الوظيفة لممانعة المجمع للترانزيستور T_3 بوصل ديود في مجمع T_4 لتبقي قيمة R_c ثابتة خلال عملية الإنتاج. بمعنى أن هبوط جهد على الديود في ممر الخرج لدارة المنطق TTL يضمن أن يكون الترانزيستور T_3 بحالة قطع متى كان T_4 مشبوعاً. كما يوصل عادة في دارة TTL القياسية ديود اضافي بين كل مدخل والارضى.

6-5-1 العناصر المنطقية الأساسية بالمنطق TTL

إذا تم استبدال الديودات المقابلة للمداخل بالمنطق DTL لكل من البوابات الأساسية NOT و AND و OR المبينة بالأشكال 6-12 و 6-13 و 6-14 بما يقابلها من ترانزستورات متعددة المشعات. يمكن الحصول وفق ما هو مبين بالشكل 6-21 على التركيبة الأساسية لكل من هذه البوابات الثلاثة وفق المنطق TTL.



الشكل 6-21: تركيبة كل من البوابات المنطقية الأساسية وفق المنطق TTL العادي

تم في الشكل 6-21a رسم التركيبة البسيطة للبوابة NOT وما يقابلها وفق المنطق TTL انطلاقاً من التركيبة القياسية للبوابة NAND وفق TTL مع تعديل بعدد المداخل (مدخل واحد)، بمعنى عندما تكون معروفة التركيبة الأساسية لكل من البوابتين الرئيسيتين NAND و NOR بأي منطق، يمكن ببساطة معرفة تركيبة بقية العناصر المنطقية بالمنطق المقابل. من الميزات الإيجابية للمنطق TTL يمكن أن نذكر بأن قدرة الخرج على الإمداد تصل إلى 10 ومدة انتشار الإشارة لا تتجاوز 20ns وهو زمن مقبول مقارنة مع المنطق الأسرع ECL، خاصة وأنه يكفي المنطق TTL جهد تغذية وحيد. وبالنظر إلى أن تأخير الانتشار يعتمد على: - زمني الانتقال والتخزين

- ثابت الزمن للمقاومة والمكثف

فإن إنقاص زمن التخزين Storage time يخفّض تأخير الانتشار. كما أن إنقاص قيم المقاومة بالدارة يخفّض الثابت الزمني، وهذا يؤدي بدوره إلى إنقاص تأخير الانتشار، لذلك أضيف تحسينات على التركيبة الأساسية لـ TTL من دون تعديل بتوصيلات التركيبة. تضمّنت التحسينات استخدام قيم مختلفة للمقاومات الداخلة بالتركيبة الأساسية، بحيث أصبحت البوابة الناتجة بتبديد استطاعة أقل (كلما زادت القيم النموذجية للمقاومات، انخفضت قيمة التيار المستجّر من وحدة التغذية). وبسرعة أعلى (السرعة تتناسب عكساً مع تأخير الانتشار، ونتم بتخفيض القيم النموذجية للمقاومات). تتضمن العائلة TTL الأكثر استخداماً بالإضافة للعائلة القياسية Standard TTL عائلات فرعية أخرى نذكر منها:

- العائلة المنخفضة القدرة Low power TTL

- العائلة عالية السرعة TTL High speed TTL

- عائلة شوتكي Shottky clamped TTL

حيث إن العائلة منخفضة القدرة مشابهة لمثيلاتها في العائلة القياسية ولكنها مختلفة بقيم المقاومات ($R1=40K$, $R2=20K$, $R3=12K$, $R4=0.5K$) وبعدم وجود الشائي، وبالتالي يكون التيار المار أقل وفقد القدرة أقل. وهذه العائلة الفرعية لها زمن تأخير قدره 33 ns، والفقد في القدرة للبوابة الواحدة هو 1 mW وأقصى تردد هو 3 MHz. أما العائلة العالية السرعة والتي يتم فيها إضافة زوج دارلينجتون Q3/Q4 - Darlington

وتقليل قيم المقاومات يسرع الانتقال من حالة منطقية إلى أخرى. ولهذه العائلة الفرعية زمن تأخير قدره 6 ns وقدرة مفقودة بمقدار 0.22W وأقصى تردد يمكن العمل عليه 50 MHz وأخيراً عائلة الربط بالشوتكي هي أسرع من أي عائلة TTL أخرى ويرمز لها بـ TTLS وتختلف عن أي عائلة أخرى في الـ TTL في أنها تستخدم ترانزستور معدل، بحيث توصل قاعدة الترانزستور بمجمعه عن طريق موحد شوتكي (المكون من وصلة معدن مع نصف ناقل)، وأن الجهد على طرفي الديود الموصل يكون بقيمة 0.4V فقط (بالمقارنة مع 0.7V في الديود التقليدي). يعمل الديود على تحويل تيار القاعدة عندما يكون الترانزستور في حالة غلق الدائرة ON، وذلك حتى يمنع من تخزين الشحنات التي تبطل من سرعته عندما يتحول من الحالة ON إلى حالة فتح الدائرة OFF (أي يعمل على تصريف الاختناق الذي يقود لحالة الإشباع مما ينعكس إيجاباً على أزمنة انتشار الإشارة وبالتالي رفع سرعة الفتح والحصول) وهذا يسرع الأداء العام للدائرة ويعطي للبوابات زمن انتقال يساوي 3 ns ومعدل استهلاك للطاقة 19 mW وتردد عمل قدره 125 MHz. تنتج هذه التركيبة المعدلة لـ TTL باستبدال كافة الترانزستورات بتركيبة TTL القياسية بترانزستورات شوتكي باستثناء الترانزستور T_3 (ليقائه المستمر بالمنطقة الفعالة) بالرمز 74S...

6-5-2 تراكيب المنطق TTL الخاصة

يوجد بالإضافة إلى التركيبة الأساسية للمنطق TTL، تراكيب عدة معدلة تساعد في الحصول على خصائص جديدة وتحقيق توابع منطقية إضافية. نستعرض منها ثلاثة عناصر أساسية فقط نصادفها بكثرة في بعض التطبيقات العملية، مثل: العناصر بمجمع مفتوح، والعناصر التوسيعية، والعناصر بثلاث حالات.

6-5-2-1 عناصر TTL بمجمع مفتوح

بالعودة إلى التركيبة القياسية لبوابة NAND بثلاثة مداخل في الشكل 6-20، يلاحظ أن من أهم سلباتها، هي أنه لا يمكن بناؤها بتشكيل عنصر تجريبي يسمح بوصل عدة مخارج مع بعضها وفق التوصيلة المبينة في الشكل 6-22. ولا يمكن وصل خرجي عنصرين من المنطق TTL معاً في الشكل أدناه، لأن مثل هذا الوصل قد يؤدي إلى قصر على الخرج

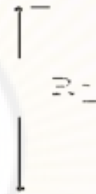
النهائي للدائرة ونزاع ترانزيستور الخرج ومكونات الدائرة.

1

2

الشكل 6-22: وصلة غير مسموح بها لخرج بوابات بالمنطق TTL العادي

علماً أنه قد يلزم في حالات كثيرة تطبيق مثل هذا الشكل من التوصيل. ولإيجاد عناصر منطقية تحقق التوصيلة أعلاه لتنفيذ وظيفة محددة، تم تعديل تركيبة TTL بالاستغناء عن العناصر التي يمكن أن تتلف بحال وصل بوابات منطقية وهي مقاومة الخرج والترانزيستور المرتبط معها. ويصبح الشكل الجديد لبوابة NAND، وفق ما هو مبين في الشكل 6-23.



الشكل 6-23: تركيبة بوابة NAND بثلاثة مداخل بالمنطق TTL بمجمع مفتوح

ولكي لا تتصرف الدائرة الناتجة تصرف الدائرة المفتوحة، يجب أن توصل المقاومة R_L بشكل خارجي إلى جيد التغذية U_{cc} لرفع الخرج إلى المستوى العالي عندما يكون الترانزيستور T_1 بحالة القطع. يسمى العنصر الجديد الناتج من دون ترانزيستور الخرج، عنصر TTL بمجمع مفتوح TTL with Open Collector (TTL-OC). وهكذا لم يعد لدى البوابة NAND وفق هذه التركيبة الجديدة ما يتلف فيما لو وصل خرجها مع خرج أي دائرة أخرى، ويصبح رمزها، وفق ما هو مبين في الشكل 6-24 أ. يوصل بالرمز المنطقي للعنصر TTL - OC مقاومة خارجية مثل R_L بين التغذية ومجمع الترانزيستور كي لا يبقى مجمع

الترانزيستور T_4 في الهواء، وحتى يتمكن أن يصبح بالقيمة 1 عند الطلب. يمكن باستخدام عناصر مجمع مفتوح ربط مخارج عدة عناصر منطقية مع بعضها، وفق ما هو مبين في الشكل 6-24 ب، حيث نقطة تلاقي خرج أكثر من بوابة بنقطة مثل x يعني الجداء المنطقي AND (أي بنفس الوقت).

—
—
—
—

∴

∴ -

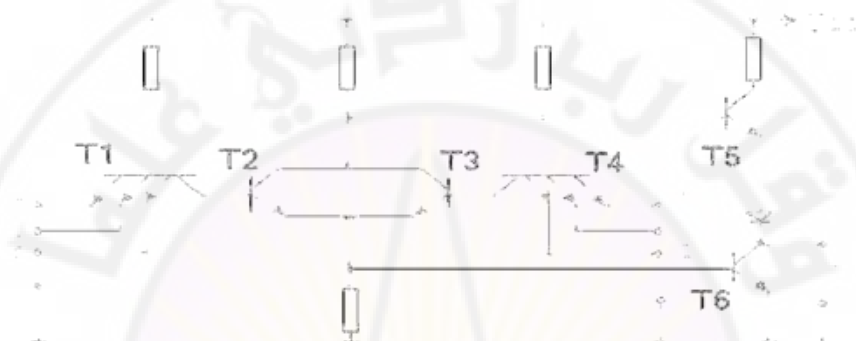
الشكل 6-24: الرمز المنطقي لعناصر TTL بمجمع مفتوح

يمكن وصل خرج بوابتين من نوع NAND بمجمع مفتوح، بحيث الأولى تمثل التابع $y_1 = (a.b.c)'$ والثانية تمثل التابع $y_2 = (xy.z)'$ مع بعضهما، ليصبح الخرج النهائي F جداء الخرجين، أي: $F = (a.b.c)' . (xy.z)' = (a.b.c + xyz)'$. يقابل مثل هذا التابع واحدة من التراكيب: NAND-AND أو AND-OR-NOT المعروفة بـ AND-NOR (بمستويين)، وفق ما هو مبين في الشكل 6-25.

الشكل 6-25: تمثيل التركيبة AND - NOR باستخدام عناصر بمجمع مفتوح

المخطط التفصيلي للدائرة التي تمثل التركيبة AND - NOR المبينة أعلاه وفق المنطق TTL، يبدو كما هو مبين في الشكل 6-26، حيث عندما يفتح أحد الترانزيستور T_2 أو T_3

أو الاثنين معاً (وهذا يتم عندما تكون $a = b = c = 1$ أو عندما $x = y = z = 1$) يصبح الخرج F بالقيمة صفر. وعندما يغلق الترانزيستورين T_2 و T_3 معاً (بوجود 0 على أي من مشعات الترانزيستور T_1 وأي من مشعات T_4) سيغلق الترانزيستور T_6 ويصبح كامل U_{cc} ، أي ما يعادل واحداً منطقياً عبر الترانزيستور T_5 على الخرج F .



الشكل 6-26: الدارة الالكترونية لتتجيز التابع المنطقي $F = (a.b.c + xyz)'$

يعرض الشكل 6-27 واحداً من التطبيقات المباشرة لعناصر TTL بمجمع مفتوح، الذي يسمح بالوصل المتناوب لعدة مولدات إشارة متباينة الخرج، والعمل على ناقل واحد وفق الدخل الذي يحدّد للمدخلين K و L للتحكم بالعملية.



الشكل 6-27: تطبيق مباشر على استخدام منطق TTL بمجمع مفتوح

عندما يكون كل من K و L بحالة الصفر، يعني ذلك أن كلا المولدين خارج الخدمة ويوجد بهذه الحالة واحد منطقي على المسرى المشترك من U_{cc} المطبقة أصلاً على R_L الموجودة

على خرج العنصر بالمجمع المفتوح. وبمجرد أن يصبح أحد المدخلين فقط K أو L بحالة الواحد، سيتم انتقاء المولد المرتبط معه. ويبقى خرج المولد الثاني وكأنه غير موجود، لأنه سيرسل على المسرى المشترك القيمة واحد الموجودة من الأساس بسبب استخدام بوابات بمجمع مفتوح. ويصبح على المسرى المشترك خرج المولد الذي تم انتقاؤه. بالطبع التركيبية $K = L = 1$ غير مسموح بها.

2-2-5-6 عناصر TTL التوسعية

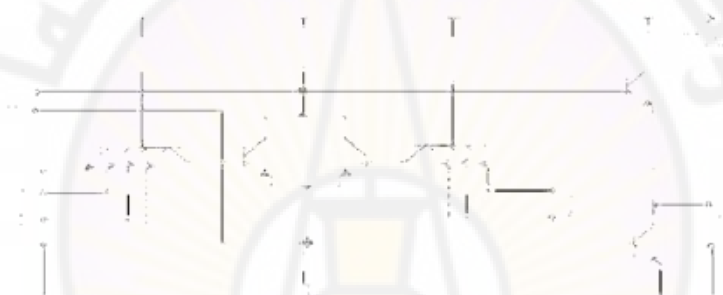
يمكن ببعض التطبيقات زيادة عدد مداخل دائرة منطقية باستخدام ما يعرف الموسع Expander. يمثل الموسع دائرة تركيبية تستخدم لتحقيق توابع منطقية مركبة بأشكال مختلفة اعتماداً على تراكيب محددة مثل التركيبية AND-NOR.

مثال: بفرض معطى تابع منطقي F من الشكل $F = (abc + de + x_1x_2x_3x_4)'$ والمطلوب تبجيز الدارة المنطقية المقابلة له.

الجواب: يمكن تنجيز مثل هذا التابع بواسطة عدة بوابات منطقية موزعة على أكثر من مستوى، كما مر معنا سابقاً، لكن لتنفيذ هذا التابع بشكل مبسط وبوثوقية عالية، يمكن أن نصادف كثيراً من الدارات المتكاملة التي صنعت بتراكيب مختلفة تحت اسم " موسع " . يمكن بهذا المثال فرض وجود تابع جزئي مثل y ، بحيث يمكن تحقيقه بدارة منطقية وفق منطق TTL، كما هو مبين في الشكل 6-28 أ

الشكل 28-6: رموز العنصر التوسيعي Expander

وبالعودة للتركيبة AND-NOR، يمكن إضافة مثل هذا التابع y (الذي يقابل تابعاً منطقياً من الشكل: $y = x_1x_2x_3x_4$) أو متممه $y' = (x_1x_2x_3x_4)'$ إلى أي تركيبة أساسية أخرى للحصول على تابع أكثر تعقيداً، حيث إن الشكل العام للدائرة المقابلة للتابع المعطى بشكل صندوق مبين في الشكل 6-28 ب. والشبكة التفصيلية التي تقابل التابع المعطى يمكن إظهارها في الشكل 6-29، بحيث عندما يكون التابع $y = x_1x_2x_3x_4 = 1$ فإن $y' = 0$ وهذا يعني أن الترانزستور بدارة الخرج مغلق، وبالتالي الخرج النهائي للدائرة F سيكون بالقيمة بالصفر بغض النظر عن قيم المداخل الأخرى a, b, c, d, e .

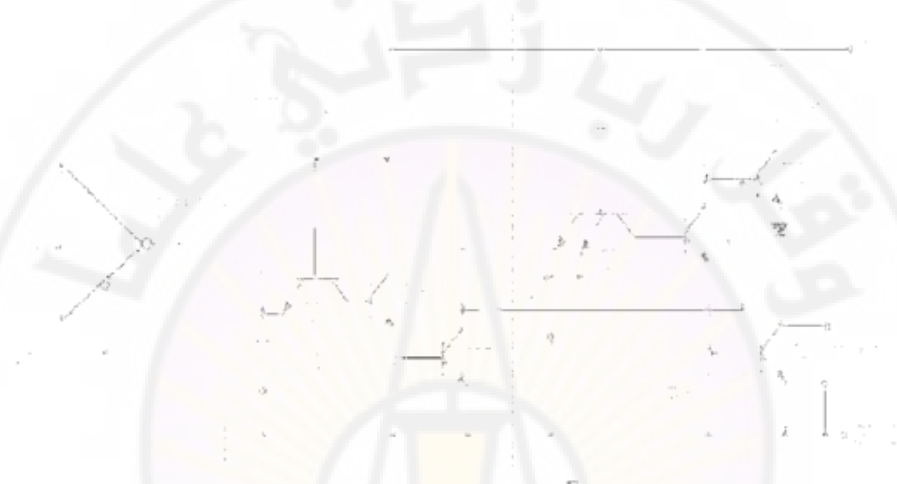


الشكل 6-29: التركيبة الداخلية لدائرة موسع Expander

3-2-5-6 عناصر TTL بثلاث حالات

تلغي العناصر المنطقية المشكلة بالمنطق TTL بمجمع مفتوح أهم الإيجابيات التي تمتاز بها العناصر المنطقية المشكلة بهذا المنطق حول أزمنة الفتح والغلق، لأن الترانزستور والديود بدارة الخرج لـ TTL القياسي المسؤولين عن تأمين ذلك، قد ألغيت بالتركيبة TTL - OC. وكما نعلم في التركيبة القياسية لعناصر TTL جميع الترانزستورات تصل إلى حالة الإشباع باستثناء ترانزستور الخرج T_3 . ولكي تبقى ميزة سرعة الفتح والغلق موجودة لدى عناصر TTL بوجود إمكانية تشكيل عنصر تجريبي (إنتاجي) التي تقدمها تركيبة المجمع المفتوح، تستخدم عناصر منطقية بثلاث حالات Three State التي تجمع الإيجابيتين المذكورتين أعلاه، حيث بتعديل تركيبة العنصر المنطقي المشكل بالمنطق TTL، يمكن جعله يمر بالإضافة للحالتين اللتين يمر بهما أي عنصر منطقي، أن يمر

بحالة معزولة (لا تُعدّ صفراً ولا واحداً)، تدعى بالحالة الثالثة للعنصر. وبالتالي سيكون لمثل هذه العناصر تركيبة داخلية خاصة أكثر تعقيداً. ولتوضيح التركيبة العامة لعناصر TTL بثلاث حالات، يمكن أن نستعرض فيما يلي مثلاً محدداً عن العنصر NOT الذي يأخذ رمزاً خاصاً وفق ما هو مبين في الشكل 6-30 أ.



الشكل 6-30: رمز وتركيبية عنصر منطقي مشكل بالمنطق TTL وبثلاث حالات

التركيبة الداخلية لهذا العنصر بالمنطق TTL وبثلاث حالات مبينة في الشكل 6-30 ب التي تمثل بوجود إشارة التحكم الخارجية I (جهد التحجيب U_{mask})، بوابة منطقية بثلاث حالات تعمل كالتالي: إذا كان I فعالاً وهذا يقابل أن $U_{mask} = U_{ss} = 0V$ يفتح بدارة التحجيب الترانزستور T_5 ويغلق كل من الترانزستورين T_6 و T_7 وتحرر التوصيلة الأساسية للبوابة NOT وكان شيئاً غير موجود، أي بوابة NOT تعمل بالحالتين المعروفتين بوجود الترانزستور T_3 والديود المسئولين عن تأمين سرعة فتح وإغلاق ترانزستور الخرج. أما إذا لم يكن I فعالاً (أي $I = 1$ أو $U_{mask} = U_{cc} = 5V$) يصبح خرج البوابة معزولاً عن دخلها، لأن كلا من الترانزستورين T_6 و T_7 سيفتح، وهذا يعني بوجود ديود بمجمع الترانزستور T_7 المنحاز أمامياً أنه لا يصل أي تغيير على الدخل إلى دائرة الخرج، أي كأن الدارة مفتوحة (أو البوابة معزولة). بمعنى آخر أن خرج الدارة لن يتغير حتى ولو تغيرت قيم الدخل. هذه الحالة الجديدة التي يعزل بها الخرج عن الدخل تسمى بالحالة الثالثة التي تعادل

حالة ممانعة عالية High Impedance. تم في المخطط الإلكتروني للبوابة عزل الخرج عن الدخل بوصل قاعدة الترانزستور T_3 مع الديود المرفق معه عبر ديود بالانحياز الأمامي إلى مجمع T_7 بدارة التحجيب.

3-5-6 مبادئ التعامل مع عناصر العائلة TTL

يوجد العديد من الشروط والمبادئ التي يفترض مراعاتها عند التعامل مع عناصر مصنعة بالمنطق TTL، وخاصة أثناء ربط عنصرين أو أكثر. وفيما يلي نستعرض بعضاً منها:

1- الالتزام بالقيم النموذجية لجهود الدخل والخرج لعنصر TTL:

بالعودة إلى الشكل 2-6 حول آلية ربط عنصرين منطقيين مع بعضهما البعض يجب تحقيق القيم المناسبة (النموذجية) لعملية الربط وهي: $U_{IHmin} \geq 2V$, $U_r \sim 1,5V$, $U_{OLmax} \leq 0,4V$, $U_{OHmin} > 2,4$, $U_{ILmax} \leq 0,8V$

2- مبادئ وصل المداخل غير المستخدمة:

بالرغم من أن أي مدخل لعنصر منطقي مشكل وفق المنطق TTL غير الموصل بأي شيء يتصرف كأنه بالمستوى H، لا ينصح بترك هذه المدخل حراً في الهواء (كي لا يقوم بتحريض جهود تشويش على بقية المداخل). بمعنى أن كل مدخل لـ TTL يجب وصله وتحديد مستواه بشكل لا يؤثر على الوظيفة المطلوبة. ويمكن في الحالات التي يكون فيها بعض مداخل TTL غير مستخدمة في التصميم (كاستخدام مدخلين فقط من بوابة NAND بثلاثة مداخل) التعامل مع هذه المداخل وفق الحالتين التاليتين:

الأولى: وفيها يمكن وصل المداخل الحرة مع مصدر التغذية، ولكن هذا يهدد بخطر الانهيار خلال الهزات اللحظية بسعات عالية للتغذية على خطوط النقل. ولتلافي هذه الحالة يمكن وصلها عبر مقاومات بقيمة عالية من رتبة عدة كيلو أومات إلى التغذية، وبما أن ذلك سيعقد الدارة لا ينصح به أيضاً.

الثانية: وهي الأفضل، وفيها يمكن وصل المداخل الحرة (غير المستخدمة) مع مداخل مجاورة مستخدمة ماثلة بالقيمة.

6-6 تقانات أحادي القطبية بتشكيل العناصر المنطقية

نستعرض فيما يلي بعض تقانات أحادي القطبية في تشكيل نفس العناصر المنطقية التي سبق وتم التعرف عليها بتقانات ثنائي القطبية. نعلم بأن جملة كاملة من الترانزستورات القنوية أحادية القطبية FET قد طورت إلى أن ظهرت ترانزستورات لا تختلف عن بعضها بالتركيب الميكانيكية فقط وإنما أيضاً بأسلوب العمل. يرد في المراجع العلمية تصنيفات عديدة لهذا التطور، المهم منها هو ترانزستورات MOSFET أي الترانزيستور القنوي ذي البوابة (أو المزلاج) المعزولة، حيث إن الفارق الجوهرى بين ترانزستورات BJT و MOS هو أن مقاومة الدخل في MOS على عكس BJT عالية جداً من رتبة الميغا أوم وهي قريبة من قيمة مقاومة الدخل للصمامات الإلكترونية. كما أنها تعمل عند ترددات عالية من رتبة الـ GHz وأقل ضجيجاً من ترانزستورات BJT. ومن الميزات الأخرى الهامة لتقانات أحادي القطبية هو إمكانية استخدام ترانزستورات MOS بوظيفة مقاومة متغيرة يمكن التحكم بقيمتها عن طريق الجهد، وهذا له انعكاس إيجابي على درجة التكامل، لأن الترانزيستور خلال عملية التكامل يأخذ حجماً أصغر بكثير مما تأخذه المقاومة وهذه الميزة غير المتوافرة لدى تقانات ثنائي القطبية تسمح بتحقيق درجة تكامل عالية لدى تقانات أحادي القطبية. يستخدم بترانزيستور MOS تسميات مختلفة للأقطاب الرئيسية، حيث:

- القطب الذي تتجه منه حاملات الشحنة الأساسية (الثقوب Holes أو الإلكترونات - Electrons) لتصب في القناة يسمى بالمنبع S - Source.
 - القطب الثاني المقابل للمجمع Collector يسمى بالمصرف D- Drain (حيث المنبع والمصرف يمثلان المهبط والمصعد لدى الصمامات الإلكترونية).
 - قطب التحكم وهو يناظر القاعدة لدى BJT ، ويسمى أيضاً بالبوابة G-Gate.
- وهنا تجدر الإشارة إلى أن السعة بين البوابة والمنبع وما بين البوابة والمصرف هي من رتبة أحاد البيكوفاراد وهذا يفسر سبب البطء لدى ترانزستورات MOSFET بالمقارنة مع BJT لأنها تحتاج لزمّن معين (غير مهمل) لعملية شحن هذه المكثفات غير موجود مثيل له لدى ترانزستورات BJT.

يوجد أربعة أنواع أساسية من بني MOS بحسب نوع القناة إن كانت من نوع P أو N وبحسب نمط العمل إن كان أغنياً Enhancement أو أفقارياً Depletion. وإذا كانت الحوامل الأكثرية ثقوباً، تكون القناة من نوع P ويتم الحصول على عنصر PMOS. أما إذا كانت الحوامل الأكثرية إلكترونات، فتكون القناة من نوع N، ويتم الحصول على عنصر NMOS. ويعتمد تحديد نمط العمل، على الحالة التي تكون فيها منطقة القناة عندما يكون الجهد المطبق على البوابة صفراً. إذا كانت القناة موصلة (بنتيجة إشابة القناة في البداية بقليل من الشوائب نوع P)، يسمى النمط أفقارياً، لأن التيار يسري في هذا النمط ما لم تستنزف القناة بتطبيق حقل بوابة. أما إذا تركت المنطقة تحت البوابة في البداية غير مشحونة، يجب عندئذ تضمين القناة في حقل البوابة قبل أن يتمكن التيار من الجريان، حيث يكون التيار قد تعزز بوساطة جهد البوابة، ويتم الحصول بذلك على النمط الأغنياً. يظهر الشكل 6-31 الرموز الكهربائية الأربعة لترانزستورات MOS.

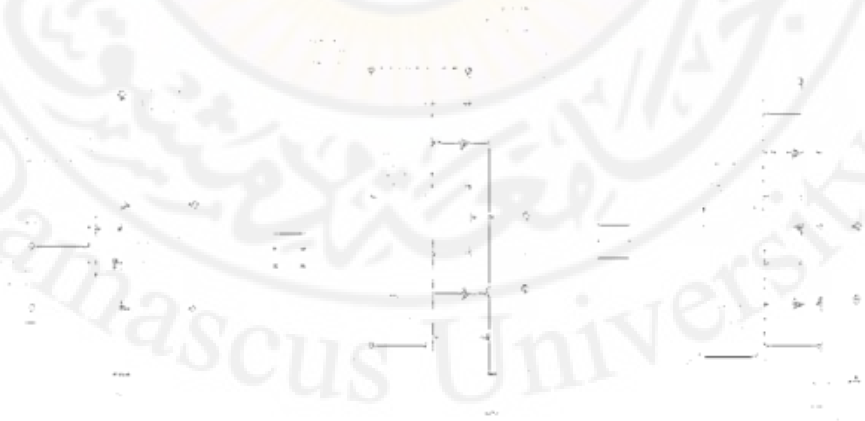


الشكل 6-31: الرموز الكهربائية الأربعة لترانزستور MOS

قد يظهر بترانزستورات MOSFET قطب رابع يمثل الشريحة الحاملة أو القاعدة B (Substrate) وهذا القطب إما أن يترك للوصل بشكل خارجي ويكون لترانزستور أربعة أقطاب، أو أن يقصر داخلياً مع S ويصبح لترانزستور ثلاثة أقطاب. ويدل الرمز بالخط المقطع الواصل بين المنبع والمصرف على النوع المعزز (الأغنياً). وفي هذا الرمز توصل الشريحة الحاملة مع المنبع. ويوجد رمز بديل يهمل الشريحة الحاملة، ويوضع بدلاً من ذلك سهم في طرف المنبع ليعين الاتجاه الموجب لسريان التيار.

6-6-1 الأحمال بتقانات أحادي القطبية

تم اللجوء لتحقيق درجة التكاملية العالية والاستطاعة المنخفضة بالنسبة للعناصر المصنعة بتقانات أحادي القطبية إلى استخدام أحمال غير ظاهرية. وللتوضيح نستعرض فيما يلي شكل الأحمال في تقانة مثل PMOS باستخدام عنصر مثل NOT كما في الشكل 6-32. ويشبه الشكل 6-32 أ دارة عاكس باستخدام ترانزستور ثنائي القطبية، لكن المقاومة R تشغل عند تكامل الترانزستور في دارة متكاملة مساحة تساوي تقريباً 20 مرة من المساحة التي يشغلها العنصر الفعال. وبهذا الشكل تفقد ميزة الـ MOSFET في الكثافة العالية بالتكامل، لذلك نادراً ما تستخدم مثل هذه المقاومة، ويستعاض عنها كما هو واضح في الشكل 6-32 ب بترانزستور ثنائي Q_2 من نوع MOSFET يقوم بوظيفة مقاومة غير قطبية بدلاً من المقاومة الخطية R الظاهرية. وبما أن ممانعة الدخل للترانزستور Q في حالة الفتح في الشكل 6-32 أ كبيرة فهذا يعني أن مقاومة التحميل يجب أن تكون كبيرة أيضاً، بالتالي تم استبدالها بترانزستور MOS آخر يعمل بدور مقاومة، ويؤدي إلى تخفيض استطاعة العنصر أيضاً. وتؤمن هذه الطريقة في الاستبدال تحقيق نظم عالية التكامل باستخدام ترانزستورات ذات الأثر الحفلي فقط من دون الحاجة لمركبات أخرى مثل الديودات والمقاومات والمكثفات التي تأخذ عادة حجماً كبيراً عند التكامل.



الشكل 6-32: بوابة NOT بنوعين من الأحمال بالتقانة PMOS

يؤدي وجود مولدي تغذية في الشكل 6-32 ب إلى تعقيد الدارة المتكاملة، وهذا بدوره قاد إلى الشكل 6-32 ج الذي يوصل فيه القطب القيادي (أي البوابة) بالمجمع وتحويل نظام عمل الترانزيستور إلى العمل في منطقة الإشباع (كمولد تيار). وهو أسلوب التحميل الذي سيعتمد خلال التعريف بالتقانات PMOS, NMOS, CMOS. وبهذا الشكل الأخير يكون الترانزيستور الفعّال Driver من النوع الأغنائي، أما الحمل فهو من النوع الإفقاري.

6-6-2 التقانات PMOS, NMOS, CMOS

أولاً- التقانة PMOS: يمكن بهذه التقانة تحقيق عناصر منطقية متعددة. نظهر فيما يلي وفق الشكل 6-33 تركيبة كل من البوابتين NAND, NOR بهذه التقانة، حيث يجب وصل الأساس لجميع الترانزستورات بالأرض حتى لو لم يرد ذلك في المخططات (حرصاً على عدم تعقيد الشكل). وبمتابعة كل شكل على حدة يمكن التأكد من جدول الحقيقة العائد للعنصر انطلاقاً من أن الوصل التفرعي يعني الجمع والتسلسلي يعني الضرب المنطقي.

الشكل 6-33: بوابات NOR و NAND بالتقانة PMOS

تستخدم العناصر المنطقية بالتقانة PMOS وفق المنطق الموجب القيمتين: $U(0) = 0$ و $U(1) = -U_D$ وهي بطيئة، لأن مدة انتشار الإشارة فيها تزيد عن الـ 100 ns. كما أن عملية تكاملها صعبة، وتأخذ حجماً لا يمكن تصغيره عن حد معين نظراً لجهود التغذية العالية التي تحتاجها، وبالتالي عند التصغير يخشى من انهيار القناة. يمكن بالإطار العام الحصول على بوابة AND مثلاً بربط عاكس بعد بوابة NAND والحصول على بوابة

OR بربط عاكس على خرج بوابة NOR. يمكن بالتقانة PMOS مثل تقانات ثنائي القطبية تشكيل بعض التوابع المركبة على شكل موسع Expander، مثل $y = (a \cdot b + c)'$ بالتركيبة AND - OR - NOT وفق ما هو مبين في الشكل 6-34.



الشكل 6-34: التابع المنطقي $y = (A.B + C)'$ بالتقانة PMOS (Expander)

وبشكل عام تشغل بوابة مثل NAND بثلاثة مداخل بالتقانة MOS (مع الحمل بدارة الخرج) ما يعادل 1/3 الحجم الذي يشغله الترانزستور الواحد إن كان من النوع BJT خلال عملية التكامل على الشريحة، وهذا يفسر الفرق بين درجة تكامل كل من تقانات أحادي وثنائي القطبية. لا تصلح التقانة PMOS للتكاملات العالية، بل لتشكيل العناصر الرخيصة منخفضة درجة التكامل فقط.

ثانياً- **التقانة NMOS**: تستخدم هذه التقانة ترانزستورات MOS بالقناة N، ويمكن الحصول عليها من التقانة PMOS بتبديل القطبية، بحيث يصبح U_D بدلاً من $-U_D$ وعندها يصبح $U(0) = 0$ و $U(1) = U_{DD}$ ، أي أنها متمم للتقانة PMOS. وبالنظر إلى الحركية الأكبر للإلكترونات في العناصر المصنعة بالتقانة NMOS بالمقارنة مع الثقوب (والتي تصل إلى 2.4 مرة زيادة)، وإمكانية تشغيل هذه القناة بجهد منخفض 5V+. يمكن بحسب مسارات إنتاج الترانزستورات بالقناة N تخفيض مدة انتشار الإشارة وتصغير أبعاد الترانزستور نفسه وتحقيق درجة تكامل أعلى بالمقارنة مع التقانة PMOS. يمكن تشكيل بعض العناصر المنطقية بـ NMOS، وفق الشكل 6-35.

الشكل 35-6: بوابات NOT و NOR و NAND بالنقانة NMOS

ثالثاً- النقانة CMOS: إذا تم تصنيع ترانزستور MOS بالنقانة P وآخر بالنقانة N على نفس الحامل يمكن الحصول على MOSFETs المتمم Complementary MOS، اختصاراً CMOS. تبدأ عملية تصنيع عنصر وفق النقانة CMOS بالطبقة الحاملة نوع N مثلاً، ثم تجرى حفرة عمودية لتحتوي النوع الآخر، أي P بحيث يتشكل ترانزستورين بأن واحد على نفس الحامل، أحدهما يعمل متمماً للثاني. تمتاز النقانة CMOS بالمقارنة مع غيرها من التقانات التي سبق ومرت معنا بتبديد منخفض للاستطاعة، حيث إن تبديد الاستطاعة بحالة السكون لبوابة CMOS هو من رتبة 0,1 mW وفي حالة التبديل (إذا كان معدل التبديل 1 MHz) بحدود 1 mW وقد يصل إلى 5 mW عندما يكون التبديل 10 MHz. يمكن بالنقانة CMOS مثل التقانات الأخرى تشكيل معظم العناصر المنطقية، حيث يحدّد عمل العنصر الواحد بوحدة تغذية وحيدة ضمن المجال من 3V حتى 18 V وتكون قيمة U_{DD} عادة 5 V. يوجد بالتقانات المندرجة تحت تقانات أحادية القطبية، بالإضافة إلى التقانات الثلاث المذكورة أعلاه العديد من التقانات الأحدث والأسرع، مثل التقانات: VMOS و FAMOS و HMOS وغيرها.

6-6-2-1 بوابة NOT بتقانة MOS

الشكل العام لدارة العاكس بالتقانة CMOS مبين في الشكل 6-36.



الشكل 6-36: تركيبة العنصر NOT وفق التقانة CMOS

تم وصل البوابات G_1 و G_2 إلى بوابة مشتركة G ويبقى كل من S_1 و S_2 القطبين الأساسيين، بحيث يوصل أحدهما إلى الأرض والثاني للتغذية. وبمعنى آخر يوجد في نفس التركيبة ترانزستورين مختلفان، أحدهما سيكون بالقناة P ، والآخر بالقناة N . وبالتالي عوضاً عن إضافة ترانزستور خارجي بوظيفة حمل أومي، يمكن بنفس التركيبة استخدام الترانزستور بالقناة N بوظيفة العنصر الفعال والترانزستور الآخر بالقناة P بوظيفة الحمل. يتضح من الشكل، أن الترانزستورين موصولان على التسلسل عن طريق ربط المصارف مع بعضهما، ويؤخذ الخرج الإجمالي من نقطة وصل المصرفين معاً في النقطة D . وعندما يتغير جهد الدخل U_I على البوابة المشتركة G من $U(0) = 0$ إلى $U(1) = U_D$ يتغير جهد الخرج U_O من $U_O(1) = U_D$ إلى $U_O(0) = 0$ ، لأنه عندما يكون $U_I = 0$ سيكون $U_{GS1} = 0$ ويكون Q_1 بحالة قطع. أما الجهد U_{GS2} سيكون سالباً $U_{GS2} = -U_D$ ، لأن الترانزستور PMOS يعمل في منطق معاكس للترانزستور NMOS. وبالنظر إلى أن الترانزستورين Q_1 و Q_2 موصولان على التسلسل فإن التيار الذي يمر عبر Q_2 يساوي التيار المار عبر Q_1 ، ولكن Q_1 مغلق لا يمر به أي تيار، وهذا يعني أن $I_D = 0$ وبهذه الحالة $U_{GS2} = -U_D$ يقتضي $U_{DS2} = 0$ و $U_O = U_D$ وهذا يمثل عمل العاكس.

6-2-6-2 البوابتين الأساسيتين NAND و NOR بتقانات MOS

تركيبية كل من البوابتين المنطقيتين NAND و NOR بمدخلين لكل منهما بالتقانة CMOS مبينة في الشكل 6-37.

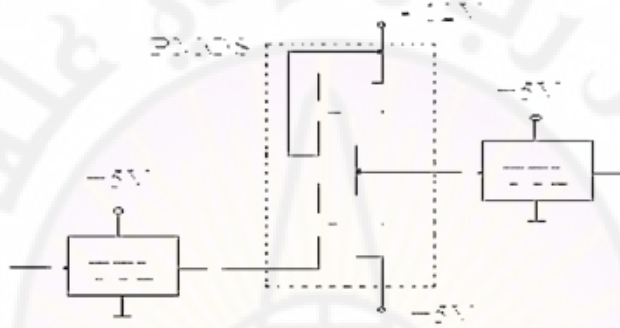


الشكل 6-37: تركيبية البوابتين NAND, NOR بالتقانة CMOS

تحتاج البوابة NAND بمدخلين بالتقانة CMOS إلى أربعة ترانزستورات (بمعدل ترانزستور واحد لكل مدخل مترافق مع ترانزستور واحد بوظيفة حمل بخلاف ما مر معنا سابقاً بالنسبة لبوابة NAND بمدخلين بكل من التقنيتين PMOS و NMOS والتي كل منهما تحتاج إلى ثلاثة ترانزستورات لإنجاز التابع المنطقي المقابل. يحتوي عنصر CMOS ضمناً لكل مدخل على ترانزستورين أحدهما فعال والآخر حمل. وبالتالي يلاحظ في الشكل 6-37 أنه عندما يكون $A = B = L$ ، فإن T_1 و T_2 يكونان مغلقين T_3 و T_4 يكونان موصولين، ويصبح $y = U_D = 1$. أما عندما يكون $A = B = H$ فإن T_1 و T_2 تصبح بحالة توصيل، ولكن T_3 و T_4 تغلق ويصبح $y = 0$. وفي حال أن أحد الترانزستورين T_1 أو T_2 كان في حالة الفتح فإن U_D سيظهر على الخرج y نتيجة الربط التسلسلي للترانزستورات الفعالة والربط التفرعي للأحمال. وهذا يقابل التابع NAND. وبخصوص البوابة NOR في الشكل 6-37 ب تماثل البوابة NAND، إذ يمكن تشكيل أي عنصر آخر بالتقانة CMOS، حيث تم في الشكل أعلاه اعتماد ترانزستورات من النمط الأغنائي (ويمكن أن تستخدم الأحمال من النمط الإفقاري).

6-7 الموائمة بين عناصر ثنائية القطبية وأحادية القطبية

بالنظر إلى أن التغذية الكيرائية اللازمة للتقانة PMOS سالبة لا يمكن ربطها مباشرة مع عناصر مثل TTL لعدم توافقهما حتى بقيم الجهود وضرورة ربط مثل هذه العناصر ببعض التطبيقات، يلزم استخدام توصيلة إصلاح Interface للموائمة بين مصدرين مختلفين للتغذية، وفق ما هو مبين في الشكل 6-38.



الشكل 6-38: الموائمة بين تقانتي أحادي وثنائي القطبية

تقوم دارات تحويل أو نقل المستويات المنطقية للتقانة TTL في الشكل أعلاه بإصلاح الجهد بين كل من المنطق TTL و PMOS بالاتجاهين، بمعنى التحويل من التغذية بـ +5V المعمول بها لتغذية TTL إلى الجهد المناسب لتغذية دارات PMOS. يمكن تغذية العناصر المصنعة بالتقانتين NMOS و CMOS مباشرة بجهد وحيد +5V وبهذا الشكل الموائمة بين العناصر المشكلة بكل من التقانات TTL و NMOS و CMOS محققة. يمكن أن تعمل العناصر بالتركيبة CMOS أيضاً بجهد تغذية منخفضة حتى 1V وبسحب تيار محدود $5 \mu A$ مما جعل منها أكثر تقانات MOS انتشاراً. وهنا تجدر الإشارة إلى أنه يجب اتخاذ احتياطات كبيرة عند لمس عناصر CMOS أو تركيبها (خاصة لمس مأخذ الدارة المتكاملة) حتى لا يتسبب ذلك في احتمالات تلفها نتيجة زيادة الجهود الساكنة. كما يجب الحذر عند استعمال المواد التي تولد جهود استاتيكية (مثل الأقمشة والبلاستيك والنايلون). ويفضل عند توصيل الدارة وصل أطراف التغذية أولاً قبل أي مأخذ آخر، كما أنه يجب توصيل المداخل غير المستخدمة في الدارة بالشكل المناسب.

نختم هذا الفصل بإجراء مقارنتين بسيطتين هما:

الاولى: مقارنة بين العناصر المشكلة بتقانات MOS وتقانات TTL ومنها نذكر الفوارق التي تتعلق بـ :

- السرعة: تمتاز عناصر TTL بسرعة أعلى من عناصر MOS.
- الحجم: تشغل عناصر MOS حجماً أصغر بكثير من الحجم الذي تشغله عناصر TTL ، وبالتالي يمكن بالتقانة MOS تأمين درجة تكامل أعلى.
- استهلاك الطاقة: يلزم عناصر TTL طاقة أعلى بكثير من عناصر MOS
- الوثوقية: تمتاز عناصر MOS بوثوقية أعلى من عناصر TTL، نظراً لصغر حجمها، وما يترتب على ذلك من تقليل لدور الإنسان بتشكيل المخططات.

الثانية: مقارنة بين تقانات أحادية القطبية نفسها، حيث إن أهم المواصفات لكل من التقانات الثلاث الرئيسية المندرجة تحت تقانات أحادية القطبية هي:

- تقانة PMOS: بالنظر إلى السرعة المنخفضة للتقانة PMOS في الفتح والغلق (الاستجابة) وقدمها، لا تُعدّ هذه التقانة من التقانات الجيدة، لذلك اقتصر وجودها على معالجات الجيل الأول وبعض دارات الدعم الخاصة بها.
- تقانة NMOS: استخدمت بكثرة في تصنيع مختلف معالجات الجيلين الثاني والثالث، وكذلك ذواكر ROM و RAM عالية السعة. وبالنظر إلى سرعة هذه التقانة التي تزيد بما يقارب ثلاث مرات عن سرعة PMOS وتوافقها مع دارات TTL أصبحت الأساس لبعض التقانات الأخرى المشهورة مثل:
- تقانة CMOS: تستخدم هذه التقانة بكثير من التطبيقات وبكل من الدارات الرقمية والتمثيلية، وتُعدّ الأوسع انتشاراً بسبب المدى الواسع لجهد التغذية الممكن استخدامه والممانعة الجيدة ضد الضوضاء (حيث يمكن أن تعمل عناصر CMOS بشكل صحيح، حتى لو بلغت نسبة الضجيج 40% من U_D)، علاوة على استهلاكها المنخفض للطاقة والاستقرار الحراري والقدرة على الإمداد على الخرج إلى 30.

8-6 أسئلة وتمارين

اختر الجواب الصحيح لكل من التساؤلات التالية:

- 1- The number of level in a digital signal is:
a- one b- two c- four d- ten
- 2- A pure sine wave is :
a- a digital signal b- can be digital or analog signal
c- analog signal d- neither digital nor analog signal
- 3- The abbreviation TTL means
a- transistor - transceiver latch. b- three - transistor logic.
c- transistor- transistor logic d- Two - transistor logic.
- 4- Fan out of DTL gates can be
a- 8 b- 10 c-30 d- none of all
- 5- Which of the following consume minimum power
a- TTL b- CMOS c-DTL d- RTL
- 6- Which of the following is the fastest logic
a- TTL b- ECL c-LSI d- CMOS
- 7- Which TTL logic gate is used for wired ANDing
a- Open collector output b- Totem pole
c- Tri state output d- ECL gates
- 8- A family of logic devices designed for extremely high speed applications is called:
a- NMOS. b- ECL. c- PMOS. d- TTL.
- 9- The logic 0 level of CMOS logic device is approximately
a- 1.2 V b- 0.4 V c-5 V d- 0 V
- 10-CMOS circuits consume power
a- Equal to TTL b- Less than TTL
c- Twice of TTL d- Thrice of TTL
- 11-CMOS circuits are extensively used for ON-chip computers mainly because of their extremely
a- Low power dissipation b- High noise immunity
c- Large paking density d- Low cost
- 12- The unused input for a NOR gate should be tied to
a- HIGH. b- ground.
c- another unused input. d- both B and C

- 13- The primary advantage of ECL over TTL is that ECL integrated circuits have
- a- very low power dissipation.
 - b- low noise margins.
 - c- very short propagation delay times.
 - d- both A and C
- 14- Which one of the following families has less delay time
- a- Standard TTL Family
 - b- TTL High speed TTL Family
 - c- Low power TTL Family
 - d- Shottky clamped TTL Family
- 15- The major advantage of TTL logic circuits over CMOS is
- a- lower propagation delay.
 - b- the ability to output higher voltages.
 - c- more modern design.
 - d- very low power consumption.
- 16- The major advantage of CMOS logic circuits over TTL is
- a- very low power consumption.
 - b- the ability to produce several output voltage levels.
 - c- lower propagation delay.
 - d- much higher propagation delay.

الفصل السابع: الدارات المتكاملة

نشأت الدارات المتكاملة Integrated Circuits (اختصاراً IC) على مختلف أنواعها أثر تطور وسائل تصنيع الترانزيستور. وقد خضعت تكنولوجيا تصنيع الدارات المتكاملة إلى عدة مراحل، وأصبح يوجد دارات متكاملة تمثيلية Analog circuits مثل دارات التكبير، ودارات رقمية Digital circuits تشمل دارات التحويل وتخزين المعلومات ومعالجتها. سيتم بهذا الفصل التعرف على الدارات المتكاملة، أشكال تواجدها، وعلى آلية استخدام الدارات الرقمية منها في انجاز المخططات المنطقية من دون الدخول بتفاصيل تصنيعها. تعرف الدارة المتكاملة على أنها شريحة أو رقاقة Chip، توصل العناصر الالكترونية بما فيها البوابات المنطقية المختلفة داخلها، والرقاقة توضع بدورها ضمن حاوية Container. تصنع الحاوية عادة من السيراميك أو البلاستيك أو المعدن ويخرج منها مغارز (مآخذ) Pins. توصل مختلف الإشارات من الرقاقة إلى نقاط توصيل تلحم إلى المغارز. تخصص كل دارة متكاملة بحسب العناصر المكونة لها لتحقيق وظيفة (وظائف) محددة. تصنع الدارات المتكاملة تجارياً باعتبارها منتجاً جاهزاً للاستخدام عند الطلب. وكانت معظم الدارات المتكاملة مع أول ظهور لها ذات تركيبة بسيطة لعوامل تتعلق بتكنولوجيا تصنيعها، حيث كانت تظهر على شكل تكرار أو نقل مباشر للدارات المصنعة من عناصر منفصلة. ومع تطور المايكرو إلكترونيك Microelectronics أصبح يوجد دارات متكاملة أكثر تعقيداً، حيث تم تجاوز مرحلة تكرار النماذج المبنية من وحدات منفصلة إلى مرحلة تشكيل منظومات متكاملة على شكل دارة متكاملة واحدة.

7-1 تصنيف الدارات المتكاملة

تصنف الدارات المتكاملة وفق معايير عدة تتعلق بمبدأ العمل وشكل التغليف وتقانة التصنيع ودرجة التكامل وغيرها. وفيما يلي نعرض باختصار من مجمل معايير تصنيف الدارات المتكاملة بعض التفاصيل المتعلقة بمعياري درجة التكامل فقط. يقصد بدرجة تكامل دارة متكاملة عدد العناصر المنطقية المجمعة في الشريحة الواحدة، حيث يمكن تقسيم الدارات المتكاملة وفق هذا المعيار إلى المجموعات التالية:

1- دارات متكاملة قليلة أو منخفضة درجة التكامل (دارات ذات تكامل ضيق النطاق) يرمز لها اختصاراً SSI (Small Scale Integration). يصل عدد البوابات المجمعة ضمن الشريحة الواحدة من دارات هذه المجموعة إلى 10 بوابات. وبالتالي فإن هذه المجموعة تشمل الدارات المتكاملة البسيطة فقط مثل البوابات المنطقية العادية.

2- دارات متوسطة نطاق التكامل MSI (Medium Scale Integration)، وفيها يتراوح عدد البوابات المجمعة ضمن الشريحة الواحدة من 11 حتى 100، وبالتالي دارات هذه المجموعة تشمل دارات منطقية من النوع التركيبي والتابعي مثل الجوامع والموزعات والمشفرات والعدادات والسجلات

3- دارات عالية نطاق التكامل LSI (Large Scale Integration)، وفيها يصل عدد البوابات إلى ما يقارب 1000 بوابة. تضم دارات هذه المجموعة عادة دارات مشكلة من شرائح SSI و MSI مجمعة مع بعضها على شريحة واحدة، مثل بعض الذواكر ومحولات الإشارة الرقمية إلى تمثيلية وبالعكس.

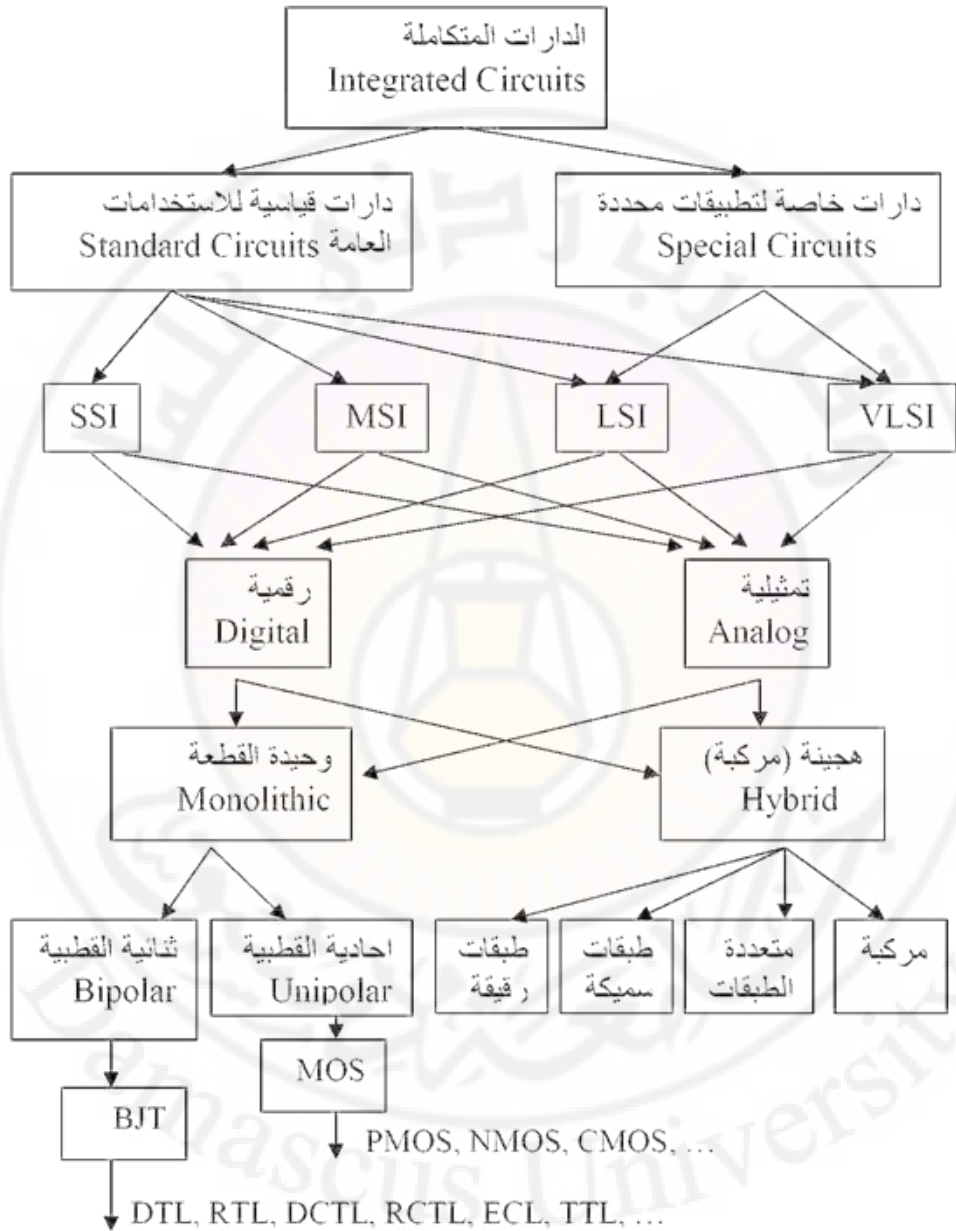
4- دارات عالية التكامل جداً VLSI (Very Large Scale Integration)، وفيها يزيد عدد العناصر عن 1000، كما هو الحال في بعض المعالجات والمتحكمات الصغيرة. يبين الشكل 1-7 أهم التصنيفات المعتمدة للدارات المتكاملة، حيث يظهر من المستوى الأول للمخطط أن أي دائرة متكاملة يمكن أن نصادفها، قد تكون:

- إما قياسية Standard، وبالتالي وظيفتها محددة أينما وجدت، ويمكن استخدامها للتطبيقات العامة وتوصيلها يكون بشكل قياسي، مثل الدارات الخاصة بالعمليات الحسابية، والترميز وغيرهما.

- وإما دارات غير قياسية للتطبيقات الخاصة، مثل بعض دارات التحكم التي يمكن مصادفتها ببعض الأغراض المنزلية أو السيارات الحديثة والتي تختلف تركيبها من شركة مصنعة إلى أخرى.

وبما أن عملية تصنيع كل دائرة متكاملة تمر بسلسلة طويلة من الإجراءات إلى حين ظهورها منتجاً جاهزاً للاستخدام، فإننا نادرًا ما نصادف دارات متكاملة بنطاق تكامل منخفض أو حتى متوسط للتطبيقات الخاصة. يلاحظ بالمستوى الثاني للمخطط أن الدارات الخاصة غير

القياسية، تصنع بدرجة تكامل عالية LSI وما فوق.



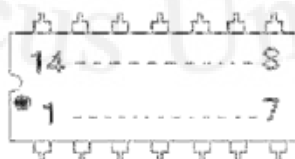
الشكل 1-7: تصنيفات الدارات المتكاملة

المستوى الثالث من المخطط مرتبط بنوع الإشارة التي تعالج بالدارة، حيث يصادف نوعان من الدارات المتكاملة، تصنف بحسب النوعين الأساسيين للإشارات التي يمكن مصادفتها في التطبيقات العملية وهي: الإشارات التمثيلية Analog signals التي تعالج في الدارات التمثيلية وفيها قيم جهود الدخل والخرج تتغير ضمن مجال من القيم (مثل إشارات المجسات البيولوجية). والإشارات المتقطعة Discrete signals التي تعالج في الدارات الرقمية، وفيها قيم جهود الدخل والخرج تأخذ واحدة من قيمتين فقط (مثل إشارات مفتاح كهربائي). توجد الدارات المتكاملة بنوعيها الرقمي والتمثيلي على شكل دارات هجينة (دارة مركبة من أكثر من مكون) أو دارات وحيدة القطعة (البلورة). تصنع الدارات الهجينة Hybrid Circuits على شكل طبقات رقيقة وسميكة أو متعددة الطبقات، أما الدارات أحادية البلورة Monolithic فتوجد إما على شكل دارات أحادية أو ثنائية القطبية، ومنهما تتبثق التقانات المندرجة تحت BJT و MOS التي سبق ومرت معنا بالفصل السابق. من أكثر أشكال التغليف للدارات المتكاملة وجوداً، هو التغليف DIL (Dual Integrated Line) ذو الخطين المتكاملين. فمثلاً، دارة متكاملة بالتغليف DIL-14 تعني دارة بأربعة عشر مأخذاً، موزعة على صفين متوازيين، على كل طرف سبعة مغارز. تصنع الدارات المتكاملة خاصة الدارات من رتبة SSI و MSI بأعداد زوجية للمأخذ ... 14, 16, 18. يظهر الشكل 2-7 المسقط الرأسي لدارة متكاملة بالتغليف DIL - 14 وآلية ترقيم المغارز، حيث ينتوء بمنتصف وجه الدارة يساعد في تحديد المأخذ رقم 1. الأبعاد العائدة لأي دارة متكاملة إذا كانت تحتوي على 14 حتى 18 مأخذ مثلاً، هو: - عرض الدارة 7,5 mm.

- سماكة الدارة 3,8 mm.

- عرض المغرز الواحد 0,55 mm.

- البعد بين مغرزين متجاورين 2,55 mm.



الشكل 2-7: المسقط الرأسي لدارة متكاملة بالتغليف DIL-14

أما طول الدارة وعرضها، فيختلف بحسب عدد المغارز. وبما أن التعامل مع الدارات المتكاملة يتم من خلال المغارز ، نشير إلى أن التركيبة DIL لا تستخدم في الدارات التي تضم عدداً كبيراً من المغارز مثل المعالجات الصغيرة الحديثة.

7-2 خصائص الدارات المتكاملة

انطلاقاً من أن أي دائرة متكاملة سنصادفها في التطبيقات العملية تكون على شكل قطعة مصممة، وأن المسارات المتعلقة بعملية تكامل العناصر الأساسية المكونة للدائرة ضمن الغلاف وغيرها من الأمور المتعلقة بإنتاجها لا تدخل في صلب الموضوع ضمن هذا الكتاب، نكتفي هنا بالإشارة إلى الخصائص العامة للدارات المتكاملة من خلال ما يعرف بورقة البيانات التي ترد عادة بكتالوجات الشركة المصنعة (تحت اسم الخواص الكهربائية للدائرة). تضم ورقة البيانات للدائرة المتكاملة المفهوم العام لإشارات كل مأخذ من مأخذ الدائرة إضافة إلى وظيفتها وخواصها الكهربائية (الجهود والاستطاعة وهامش الضجيج وغيرها من العوامل التي تميز الدارة) وشروط ربطها مع غيرها من الدارات الأخرى ضمن مخطط متكامل. فبخصوص وظيفة الدارة المتكاملة، قد يكون لبعض الدارات المتكاملة أكثر من وظيفة، لكن للتمييز بين دائرة وأخرى يرمز لكل دائرة متكاملة برقم يمثل عنوان هذه الدارة. تتم عملية كتابة عنوان أو وظيفة الدارة وفق أسلوب محدد. فمثلاً، بالنسبة للدارات المصنعة بالمنطق TTL تم اعتماد أرقام محددة لتحديد السلسلة التي تتبع لها الدارة، يلي ذلك أرقام أخرى لتحديد الوظيفة، حيث تم اعتماد الرقمين 74 للإشارة إلى العائلة TTL القياسية. والأرقام التي تأتي بعد هذين الرقمين من اليمين، هي لتحديد وظيفة الدارة. تُعد الدارة ذات الرقم 7400 أول عنصر في هذه العائلة. بالإضافة إلى الرقم 74 يصادف ضمن عائلة TTL أيضاً دارات مصنعة تحت السلسلة 84، وكذلك السلسلة 54 وهي تسميات مرتبطة بمواصفات فنية لا بالوظيفة. فمثلاً، الدارات 7400, 8400, 5400 تؤدي نفس الوظيفة، لأن كل من هذه الدارات الثلاث تمثل أربع بوابات NAND بمداخل لكل منها. يتمثل الاختلاف بين الدارات الثلاثة ببعض المواصفات التقنية من حيث درجة الحرارة التي يمكن أن تعمل بها كل من هذه الدارات أو مقدار جهد التغذية المسموح به أو درجة الوثوقية. توزيع بوابات

NAND الأربعة للدائرة المتكاملة 7400 مثلاً، بالتغليف DIL-14 مبين في الشكل 7-3، حيث يمكن استخدام كل بوابة ضمن الدارة على حدة بمعزل عن البوابات الأخرى.



الشكل 7-3: التوزيع الداخلي للبوابات المنطقية ضمن الدارة المتكاملة 7400

وبالنظر إلى أن المواصفات المتعلقة بدرجة الحرارة مثلاً لكل دائرة متكاملة ضمن كل سلسلة من السلاسل 54, 84, 74 هي نفسها أياً كان رقم الدارة. نستعرض فيما يلي القيم العائدة لكل سلسلة على حدة وفق التالي:

السلسلة 74: تصلح للعمل في وسط بدرجات حرارة من 0 حتى 74°C

السلسلة 84: فتستخدم للعمل في وسط بدرجات حرارة $84^{\circ}\text{C} \div -25$

السلسلة 54: تستخدم للعمل في وسط بدرجات حرارة $125^{\circ}\text{C} \div -55$

بالطبع هذا المؤشر يدل بشكل مبدئي على أن السلسلة 54 هي الأفضل، يليها في ذلك السلسلة 84، وأخيراً 74 من حيث الوثوقية ومجالات التطبيق. ورقم السلسلة (أقصى خانتين من اليسار) يحدد إضافة لدرجة الحرارة المناسبة لعمل الدارة، أيضاً التغيرات المسموح بها في جهد التغذية للدارة، حيث إن السلسلتان 74, 84 تعملان بجهد تغذية بالمجال 5 ± 0.25 بينما السلسلة 54 تصلح للعمل بالمجال أوسع قليلاً $U_{cc} = +5 \pm 0.5\text{ V}$. يمكن اعتبار أن أهم الخصائص التي ترد في ورقة البيانات للدائرة المتكاملة التي يجب مراعاتها عند اختيار الدارة لتطبيق ما هي: تقانة التصنيع، نوع التغليف، سرعة التشغيل (تأخير الانتشار)، درجة حرارة التشغيل، نطاق تكامل الدارة، هامش الضجيج (ممانعة الضوضاء)، مقدار تبديد الاستطاعة (استهلاك الطاقة)، قدرة الخرج على الإمداد وغيرها من البارامترات.

تُعَدُّ بعض هذه الخصائص متداخلة مع بعضها، لأن تأخير انتشار الإشارة مثلاً مرتبط بشكل أو بآخر بتبديد الاستطاعة أيضاً، وبالتالي لا يوجد أي أهمية لترتيب هذه الخواص، لأنه من أجل تطبيق معين قد يكون العامل المتعلق بالسرعة مثلاً هو الأهم وفي تطبيق آخر قد يكون الأهم هو هامش الضجيج. وبما أنه سبق وأشرنا في الفصل السابق إلى بعض خصائص العناصر المنطقية، وبالتالي إلى مواصفات الدارات المتكاملة. نشير فيما يلي إلى بعض التفاصيل الإضافية حول بعض الخصائص لتحديد دور كل منها في اختيار دارة متكاملة عن غيرها لتطبيق محدد ومنها نذكر:

٥- تقانة التصنيع: تحدد تقانة المستخدمة في تصنيع العناصر المكونة لدارة متكاملة (إن كانت من مجموعة ثنائي القطبية أو أحادي القطبية) بعض الخصائص الأخرى المتعلقة بدرجة التكامل والسرعة والاستهلاك وغيرها، خاصة وأنها قد تصادف عناصر منطقية مثل NAND مصنعة بتقانات ثنائي القطبية وكذلك بتقانات أحادي القطبية. يمكن أن يصادف نفس العنصر المصنوع بالتقانة ثنائي القطبية مشكلاً وفق المنطق ECL أو TTL (ونفس الشيء بالنسبة لعناصر أحادي القطبية). فإذا كان المطلوب بتطبيق ما استخدام دارة متكاملة بتبديد منخفض للقدرة وبهامش واسع للضجيج، يتم اللجوء لاستخدام عناصر من مجموعة أحادي القطبية، أما إذا كان المطلوب بالدرجة الأولى السرعة، فيتم اللجوء إلى استخدام عناصر من مجموعة ثنائي القطبية، وهكذا يلعب نوع تقانة التصنيع دوراً هاماً في اختيار الدارة المتكاملة.

٥- سرعة التشغيل: يقصد بها تأخير الانتشار أو ما يعرف أحياناً بـ زمن الاستجابة (الزمن اللازم لانتشار الإشارة المنطقية من مدخل العنصر حتى ظهور صدى الإشارة على خرجه). يتأثر تأخير انتشار الإشارة بالعديد من العوامل التي سبق وتتمت الإشارة إلى البعض منها في الفصل السابق. وهنا سنشير إلى أثر عامل آخر يتعلق بالمادة نصف الناقلة المكونة للعنصر ودوره في تأخير الانتشار، لإظهار أن كل خاصية للدارة المتكاملة تتشعب بأكثر من اتجاه. يشمل تأخير الانتشار نوعين من الأزمنة هما: زمن التخزين وزمن الانتقال، ولإظهار ذلك ننطلق من تركيبة أبسط عنصر الكتروني، حيث كما نعلم على مستوى الديو

عندما يطبق جهداً في الاتجاه الأمامي، ويتغير هذا الجهد فجأة ليصبح جهداً في الاتجاه العكسي، فإن انخفاض قيمة التيار المار في الديود لا يصل إلى الصفر في نفس اللحظة ولا حتى إلى قيمة تيار التسريب العكسي في لحظة عكس الجهد المؤثر على الديود، لأن حاملات الشحنة لا بد أن تتحد أولاً مع الذرات وتخفي الشحنات الحرة. تمر خلال هذه الفترة نبضة من التيار العكسي، كما هو مبين في الشكل 4-7، حتى يتضاءل التيار العكسي إلى قيمة تيار التسريب العكسي.



الشكل 4-7: زمن تأخير الانتشار بالديود العادي

يتسبب التيار العكسي خلال هذه الفترة الزمنية في تخزين شحنة في الديود، بذلك تبعد حاملات الشحنة الحرة من المعبر PN (زمن التخزين). وعند إتمام ذلك يبدأ التيار العكسي في التضاؤل لتصبح قيمته مساوية لتيار التسريب العكسي (زمن الانتقال). وتكون مدة تأخير الإشارة مجموع الزمنين. ما ينطبق على الديود بخصوص زمن الانتشار ينطبق على أي عنصر منطقي يدخل في تركيبه نصف ناقل. يتراوح زمن الانتشار بمعظم أنواع الدارات المتكاملة المصنعة بتقانات ثنائية القطبية ما بين 2 حتى 100 ns. في بعض التطبيقات قد يكون التأخير في زمن انتقال الإشارة بحدود واحد من الألف من الثانية مقبولاً، أما في الحواسيب الرقمية فيلزم زمن انتشار من رتبة ns. تسمى عادة القيم التي تتأثر بها الدارات المتكاملة بقيم الحدود. وتنص المواصفات على أخذ أصغر حد للقيمة العليا وأكبر حد للقيمة المنخفضة.

*- قدرة الخرج على الإمداد: سبق وأشرنا إلى أن قدرة الخرج على الإمداد للعناصر المشكلة بالمنطق TTL هي 10 كرقم قياسي. والسؤال هو: من أين أتى هذا الرقم؟ للإجابة على هذا السؤال لابد من توضيح مفهوم التيارات المنصبة والتيارات النابعة في الدارات المتكاملة. عندما تغذي بوابة من نوع TTL بوابة أو بوابات أخرى، يجب مراعاة حدود تيارات التغذية التي يحددها الصانعون للدارة، فعندما يكون خرج الدارة في الحالة L يكون مسار التيار كما في الشكل 5-7 أ، وتعمل البوابة المغذية (تحديدا الترانزيستور T_4) كمصبب للتيار للبوابة المغذاة.



الشكل 5-7: التيارات المنصبة والنابعة بدارة متكاملة مصنعة بالمنطق TTL

تحدّد المواصفات عادة بأن التيار المنصب لا يجب أن يزيد عن 1.6 mA لكل بوابة يتم تغذيتها، وبما أنه يمكن أن يصل مجموع التيارات المنصبة في البوابة المغذية إلى 16 mA، فإن قدرة الخرج على الإمداد $Fan-out = 16 / 1.6 = 10$ ويدل هذا الرقم على أقصى عدد ممكن من البوابات التي يمكن تغذيتها ببوابة TTL. يمثل الشكل 5-7 ب الحالة التي يكون فيها الخرج في الحالة عالية المستوى H، حيث البوابة هنا تمثل منبعاً للتيار وأن أقصى حد للتيار النابع هو $40 \mu A$ لكل حمل والمجموع الكلي هو $400 \mu A$ ، ويكون $Fan-out = 400/40 = 10$.

*- تبديد القدرة: سبق وأشرنا في الفصل السادس إلى أنه تم تصميم وإنتاج عدد كبير وأنواع مختلفة من دارات TTL لتقليل تبديد القدرة وتأخير الانتشار وزيادة قدرة الخرج على الإمداد وغيرها من البارامترات المميزة للدائرة المتكاملة، حيث إن قيم البارامترات الأساسية بالنسبة لـ TTL القياسي مبيّنة في الجدول 1-7. يوجد أشكال مختلفة لتسميات الدارات المتكاملة، حيث يتعلق بعض هذه التسميات بخاصية تبديد القدرة، كما أن تحسين أي بارامتر من بارامترات الدائرة المتكاملة قد يكون على حساب بارامتر آخر.

الجدول 1-7: قيم البارامترات الأساسية لمختلف سلاسل العائلة TTL

اسم السلسلة من العائلة TTL	رمز السلسلة	تأخير الانتشار	تبديد القدرة	الخرجية
قياسي	74	9 ns	10 mW	10
قدرة منخفضة	74L	33 ns	1 mW	20
سرعة عالية	74H	6 ns	22 mW	10
شوتكي	74S	3 ns	19 mW	10
شوتكي منخفض القدرة	74LS	5,9 ns	2 mW	20
شوتكي متطور	74AS	5,1 ns	10 mW	40
متطور منخفض القدرة	74ALS	4 ns	1 mW	20

إضافة إلى تركيبة TTLS يمكن أن يصادف أيضاً تراكيب أخرى لـ TTL مختلفة عن بعضها باختلاف قيم مكونات تركيبها الأساسية. فمثلاً، يرمز للدارات المتكاملة التي تمثل عناصر منطقية باستطاعات منخفضة بالرمز L TTL، ويمكن الحصول عليها بوضع قيم مرتفعة للمقاومات الداخلة في تركيبها، وتأخذ بعد ذلك الدائرة المتكاملة الرمز 74L... على حساب تطويل مدة الإشارة. وبالعكس توجد عناصر سريعة يرمز لها بـ H TTL بتأخير انتشار قصير على حساب زيادة القدرة المبددة بخفض قيم المقاومات، وتأخذ بعد ذلك الدائرة المتكاملة الرمز 74H... يمكن من التراكيب H TTL و L TTL و S TTL تشكيل تراكيب مختلفة من الدارات المتكاملة مثل التراكيب: HS TTL و LS TTL وهذه التراكيب

تصنع تحت سلاسل مختلفة مثل السلسلة: 74LS..., 84LS..., 54LS..., لدرجة أن أول دائرة متكاملة بالعائلة TTL على سبيل المثال يمكن مصادفتها تحت كل من التسميات التالية: 7400, 74H00, 74L00, 74S00, 74LS00, 74AS00, 74ALS00. تمثل جميع الأرقام السبعة الواردة أعلاه دوائر متكاملة لنفس الوظيفة لكن بتقنيات مختلفة. وبحسب الرمز الوارد يختلف مقدار استهلاك الطاقة أو زمن انتشار الإشارة بين دائرة وأخرى. قيم البارامترات الأساسية المميزة لكل من هذه السلاسل مبينة في الجدول 1-7. يستخدم في إنشاء ذكر اسم دائرة متكاملة فقط الرقم الأساسي الذي يدل على وظيفة الدارة. وعلى المستخدم (المصمم) بعد ذلك البحث عن السلسلة المناسبة بحسب نوع التطبيق، خاصة وأن معظم الدارات المتكاملة لا تصنع بالأشكال السبعة الواردة أعلاه، بل في بعض منها. وكما أشرنا أعلاه إلى تقانة TTL من مجموعة تقانات ثنائي القطبية للدارات المتكاملة، نشير هنا أيضاً إلى التقانة CMOS من مجموعة تقانات أحادي القطبية. يصل هامش الضجيج عند تشغيل دارات CMOS إلى قرابة 40% من جهد التغذية. أما قدرة الخرج على الإمداد فقد تصل إلى $Fan-out = 30$ عند تردد 1 MHz. وتتناقص قدرة الخرج على الإمداد بتزايد تردد العمل. يعتمد بتسمية الدارات المتكاملة من نوع أحادي القطبية أرقام أخرى للسلاسل. فمثلاً، السلسلة ... 40 تشير إلى العائلة CMOS. يوجد ضمن عائلات CMOS سلاسل تبدأ بالرقم 74 أيضاً وتأخذ لتمييزها عن غيرها من سلاسل TTL الأخرى الرمز C من CMOS بعد رقم السلسلة ليصبح رمزها 74C. السلاسل التي تبدأ بـ 74C تكون متوافقة Compatible من جهة المغارز والوظيفة مع دارات TTL التي تحمل نفس الرقم. فمثلاً، الدارة المتكاملة التي تحمل الرقم 74C00 هي مثل الدارة 7400 من TTL تمثل أربع بوابات NAND كل منها بمدخلين، متوافقة معها ومصنعة بالتغليف DIL-14 لكن بالتقانة CMOS.

- توجد سلسلة أخرى من CMOS بالرمز 74CH للسرعات العالية، وهي شكل محسن للسلسلة 74C، وتزيد بها بعشرة أضعاف في سرعة التبديل.
- السلسلة 74CT هي سلسلة من CMOS متوافقة مع العائلة TTL كهربائياً، أي يمكن وصل دارات هذه السلسلة مع دارات TTL من دون الحاجة إلى دارات موائمة بينها.

- يوجد إصدارات أخرى من CMOS تبدأ بالرقمين 74 من اليسار مثل السلسلة 74VHC وهي عالية السرعة. تعد السلسلة 74VHCT من السلاسل المتوافقة مع التقنية TTL. في حال ورود رقم لدائرة متكاملة مثل 7400 من دون أية رموز أخرى، يعني ذلك أن الدارة من نوع TTL القياسية وأي رمز يضاف لهذا الرقم يفسر بحسب موضعه. وما ينطبق على العائلة TTL ينطبق على ما يخص العائلة CMOS أيضاً. أخيراً نشير هنا باختصار إلى آلية ترقيم أو تسمية الدارات المتكاملة، حيث يكون اسم الدارة المتكاملة مكتوب على السطح الخارجي للدائرة على شكل مجموعة من الرموز والأحرف والأرقام. وقد يصادف على الدارة الواحدة رقمين:

الأول: ويعني الرقم التسلسلي لإنتاج القطعة Serial number وهذا خاص بالمصنع.
الثاني: ويمثل الاسم الفعلي للدائرة، وقد يبدو لدائرة من العائلة TTL بترتيب وفق التالي:

حروف أو رموز	أرقام	حروف تحديداً	أرقام	حروف أو رموز
يبين الرمز أدناه رقم واحدة من الدارات المتكاملة				
M	00	H	74	SN

يفهم هذا الترتيب أو التسمية ابتداء من اليسار كالتالي:

القسم الأول: الرمز SN ويدل على اسم الشركة الصانعة للدائرة المتكاملة.

القسم الثاني: الرقم 74 ويدل على رقم السلسلة ويمكن أن يكون بالنسبة لـ TTL أيضاً أي من الرقمين 84 أو 54.

القسم الثالث: الحرف H أو S (من High أو Shottky) أو أية حروف في هذا الموقع

تدل على مواصفات فنية خاصة بالدائرة مثل السرعة والاستطاعة وحتى تقنية

التصنيع (مثل C للعائلة CMOS).

القسم الرابع: الرقم 00 أو أي رقم بهذا الموضع يدل على وظيفة الدائرة وهو بالحد الأدنى مكون من خانتين.

القسم الخامس: الرمز أو الحرف M يدل بهذا الموقع على مواصفات فنية إضافية خاصة بالدائرة من حيث التغليف أو الوثوقية، مثل M من Military أي للتطبيقات العسكرية (وثوقية عالية).

7-3 تغليف الدارات المتكاملة

كثير من التوابع والبوابات المنطقية تكون متاحة ضمن غلاف واحد، يعرف بالدارة المتكاملة بحيث تمتاز الدارة الناتجة بعد التغليف بمواصفات منها: حجمها يكون أصغر، وثوقيتها أعلى، واستهلاكها للطاقة منخفض. كل عنصر منطقي (أو أكثر) يكامل ضمن دارة متكاملة فمثلاً أربع بوابات XOR بالترتيب المبين بالشكل 7-6 تقابل دارة متكاملة بالاسم 74136 .



الشكل 7-6: العناصر المنطقية الداخلة في تركيب الدارة المتكاملة 74136

الشكل العام للدارات المتكاملة البسيطة ابتداء من البوابة المنطقية بتركيباتها الالكترونية كما هو الحال بالنسبة لبوابة NAND بالتقانة TTL القياسية وبمدخلين يبدو بالشكل 7-7. يظهر الشكل أيضاً تغليف أربع بوابات منطقية (NAND) بدارة تكاملية واحدة من رتبة SSI



الشكل 7-7: شكل لدارات متكاملة من البوابة إلى التغليف من رتبة SSI

يبرز من كل قطعة مصممة مأخذ، تدعى Pins مرتبة وفق شكل محدد. يظهر على السطح الخارجي لكل دائرة دائرة صغيرة أو علامة للإشارة إلى المأخذ رقم 1. أما من الداخل فتتكون كل دائرة متكاملة من شريحة chip يركب عليها كل العناصر التي تؤدي غرضاً ما، وقد يكون في الدائرة الواحدة أكثر من عنصر من نفس النوع كما هو مبين بالنسبة للدائرة في الشكل 6-7. تغلف الدائرة بأنوع متعددة من المواد البلاستيكية أو الزجاجية أو السيراميكية وغيرها. توصل مأخذ الدائرة الواحدة سواء كانت دخل أو خرج مع الشريحة الأساسية المكونة للدائرة المتكاملة. يظهر الشكل 8-7 مظهر الوصل الداخلي للشريحة مع المأخذ الخارجية.



الشكل 8-7: مظهر وصل مأخذ الدائرة المتكاملة بالـ Chip الداخلي

للدوائر المتكاملة بحسب نوعها أساليب مختلفة للرزم IC Package styles تشكل كل منها دائرة ثابتة الوظيفة. من مجمل التغليف المعروفة على مستوى الدارات منخفضة ومتوسطة درجة التكامل، نعرض فيما يلي أشكال الرزم الأكثر انتشاراً وهي: DIP, SOIC, FP, PLCC, LCCC. يظهر الشكل 9-7 بأجزائه الخمسة أهم أشكال التغليف المعمول بها.

التغليف Dual in-Line Package – DIL a-

وهو التغليف الأكثر انتشاراً بالنسبة لدارات TTL (من الرتبة SSI, MSI, LSI)

b- Small-outline IC – SOIC التغليف

c- Flat pack IC – FP التغليف

d- Plastic-leaded chip carrier – PLCC التغليف

التغليف LCCC - Plastic-Leaded chip carrier - e

الشكل 9-7: أشكال مختلفة لتغليف الدارات المتكاملة

بالطبع تغليف الدارات عالية التكامل، كما هو الحال بالنسبة للمعالجات الصغيرة يختلف عن تغليف الدارات منخفضة ومتوسطة درجة التكامل. فمثلاً الشكل العام لتغليف إحدى معالجات البنتيوم من Intel، يبدو كما هو مبين بالشكل أدناه والمأخذ متوضعة بالجهات الأربع وعلى أكثر من صف بكل جهة.

7-4 الدارات المتكاملة التمثيلية

من مجمل الدارات التمثيلية نشير فيما يلي كعينة لواحدة منها فقط هي الدارة المتكاملة 72555 المعروفة بالاسم 555، نظراً لاستخداماتها الواسعة بالأنظمة الرقمية لتوليد النبضات الساعية. تصنع الدارة 555 بالتركيبة DIL-8 والشكل العام لتوزيع إشاراتها على المآخذ مبينة في الشكل 7-10 أ، والشكل 7-10 ب يظهر المخطط المنطقي (التركيبية الداخلية) للدارة.

DIS THIR IN

2 1 1 1

8 7 6 5

1 2 3 4

1 2 3 4

END FOUR

D.S. Discharge

تفريغ

THR. Threshold

عتبة

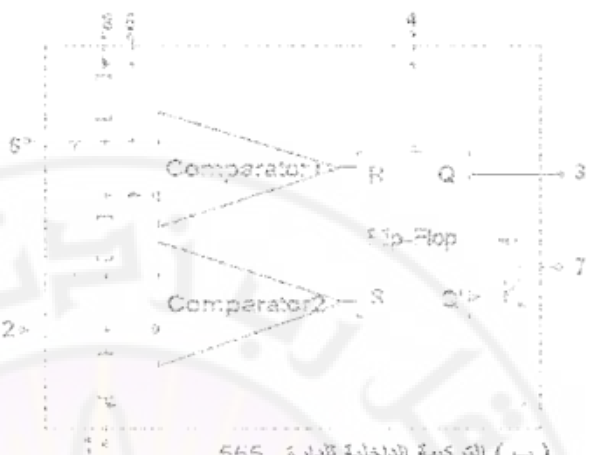
T. Trigger

قذح

R. Reset

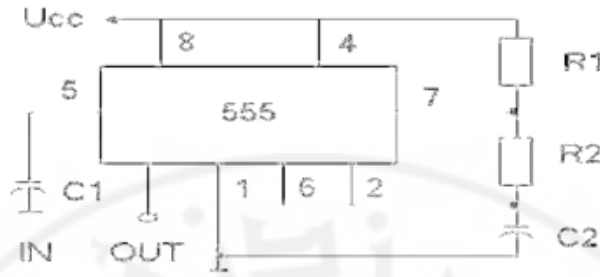
تصفير

555 الدارة (أ)



الشكل 7-10: الدارة المتكاملة 555

تُعرف الدارة 555 بأنها مؤقت زمني مصنع على شكل دائرة متكاملة IC timer لتوليد نبضات الساعة Clock pulse generator. يتضح من التركيبة الداخلية للدائرة أنها مكونة بالإضافة للمقاومات الداخلية من قلاب ومقارنين للجهد Voltage comparators. تقوم المقاومات الداخلية التي تشكل مجزئاً للجهد بتقسيم الجهد $U_{cc} = +5V$ إلى الثلث أو الثلثين، بحيث المقاومة الأولى تعطي 3.3V والثانية 1.7V من U_{cc} لتغذية المدخلين الثابتين بالمقارنين. يتلخص مبدأ عمل الدارة بالتالي: عندما يتجاوز جهد العتبة للدائرة Threshold input على المدخل 6 للمقارن الأول القيمة 3.3V يصبح المدخل R للقلاب RS بالقيمة واحد، وهذا يؤدي إلى تصفير القلاب وجعل الخرج على المغرز 3 بالقيمة 0، ويكون T منحاز أمامياً. وعندما ينتقل مدخل القذح على المدخل 2 للمقارن الثاني إلى مادون 1.7V يصبح المدخل S للقلاب RS بالقيمة واحد، وهذا يؤدي إلى نقل خرج الدارة إلى ما يقارب 5V أي حالة الواحد ويصبح الترانزيستور T بحالة القطع. يمكن التحكم بالدائرة بواسطة دائرة RC خارجية لإنتاج تأخيرات زمنية (ترددات) متفاوتة فيما بينها وبدقة Precision عالية بحسب قيم R و C . التوصيل القياسي للدائرة 555 لتعمل مولداً للنبضات الساعية مبين بالشكل 7-11 بالطبع R_1 , R_2 والمكثفتان C_1 , C_2 ليست جزءاً من الدارة وهي تتغير بحسب الطلب.



الشكل 11-7: توصيل الدارة 555 بوصفها مولداً للن نبضات الساعية

يشحن المكثف C_2 عبر المقاومتين R_1 و R_2 عندما يكون T ضمن الدارة 555 بحالة قطع. وعندما يصبح الترانزستور منحازاً أمامياً يفرغ شحنته عبر R_2 . وبهذا الشكل عندما يصبح جهد الشحن على طرفي المكثف بالقيمة $3.3V$ يؤدي جهد العتبة على المدخل 6 إلى تصفير القلاب ضمن الدارة ونقل الترانزستور إلى حالة التوصيل. وعندما يبلغ جهد تفريغ الشحنة القيمة $1.7V$ يؤدي مدخل القذح على المأخذ 2 إلى وضع القلاب بحالة الوضع وإغلاق الترانزستور من جديد. وهكذا يتناوب خرج القلاب باستمرار مابين مستويين من الجهد، بحيث يبقى المخرج في المستوى العالي لمدة تساوي مدة شحن المكثف، ويبقى بالمستوى المنخفض لمدة تساوي زمن التفريغ. يحدد زمن الشحن بعلاقات تجريبية:

$$t_H = 0.963(R_1 + R_2).C_2$$

$$t_L = 0.963R_2C_2$$

5-7 استخدام الدارات المتكاملة

بخصوص استخدام الدارات المتكاملة، نشير إلى أنه يلزم لتحقيق (أو تنجيز) أي مخطط منطقي بشكل عملي تحديد أنواع الدارات المتكاملة اللازمة لانجاز التابع الذي ينفذه المخطط وإعادة رسم المخطط متكاملأ على شكل دائرة مطبوعة لتنفيذه عملياً. بالطبع لا يمكن حصر استخدام الدارات المتكاملة بمجال محدد، لأننا نصادفها بمختلف أنواع التطبيقات وهي تتدرج بالتعقيد ابتداء من البوابات المنطقية البسيطة وحتى أحدث وأضخم المعالجات الصغيرة.

نختتم هذا العرض حول الدارات المتكاملة بعينة من بعض دارات TTL في الجدول 7-2 التي تساعد لاحقاً بانتقاء ما يلزم من دارات مناسبة خلال عرض بعض الدارات والأمثلة

للتطبيق الفعلي. يمكننا أيضاً الاستفادة من بعض الجداول الواردة في هذا الكتاب لتنفيذ بعض التطبيقات البسيطة. لايتسع المجال هنا لعرض تطبيقات على استخدام الدارات المتكاملة، وسنكتفي بما سيتم عرضه من تجارب بالفصل الخامس عشر.

الجدول 2-7: بعض دارات TTL التي تمثل بوابات منطقية

رقم الدارة	وظيفة (تركيبية) الدارة	ملاحظات
74 00	أربع بوابات NAND كل منها بمدخلين	Normal
7401	أربع بوابات NAND كل منها بمدخلين	with OC
74 02	أربع بوابات NOR كل منها بمدخلين (7428 &)	Normal
7403	أربع بوابات NAND كل منها بمدخلين	with OC
74 04	ست بوابات NOT	Normal
7405	ست بوابات NOT	with OC
7406	ست بوابات NOT , buffer/driver	with OC 30V out
7407	hex buffer/driver	with OC 30V out
74 08	أربع بوابات AND كل منها بمدخلين	Normal
74 09	أربع بوابات AND كل منها بمدخلين	with OC
74 10	ثلاث بوابات NAND كل منها بثلاثة مداخل	Normal
74 11	ثلاث بوابات AND كل منها بثلاثة مداخل	Normal
...		...
74 20	بوابتا NAND كل منهما بأربعة مداخل (7440 &)	Normal
74 21	بوابتا AND كل منهما بأربعة مداخل	Normal
74 27	ثلاث بوابات NOR كل منها بثلاثة مداخل	Normal
74 32	أربع بوابات OR كل منها بمدخلين	Normal
74 33	أربع بوابات NOR كل منها بمدخلين	with OC
74 86	أربع بوابات XOR كل منها بمدخلين	Normal

6-7 أسئلة وتمارين

اختر الجواب الصحيح لكل من التساؤلات الآتية:

- 1- CMOS stands for
 - a- complementary metal oxide semiconductor
 - b- complex metal oxide semiconductor
 - c- complex material oxide semiconductor
 - d- None of the above
- 2- The number of logic gates in LSI package is
 - a- 10
 - b- 100
 - c- 1000
 - d- None of all
- 3- Which one of the following is not ICs Package
 - a- LCCC
 - b- SOIC
 - c- DDIL
 - d- PLCC
- 4- ICs are generally made of
 - a- Silicon
 - b- Germanium
 - c- Copper
 - d- None of all
- 5- An audio amplifier is an example of
 - a- Digital IC
 - b- Linear IC
 - c- Both digital and linear IC
 - d- None of the above
- 6- The active components in an IC are
 - a- Resistors
 - b- Capacitors
 - c- Transistors and diodes
 - d- None of the above
- 7- We use ICs in computers
 - a- Digital
 - b- Linear
 - c- Both digital and linear
 - d- None of the all
- 8- The most popular types of ICs are
 - a- Thin-film
 - b- Hybrid
 - c- Thick-film
 - d- Monolithic
- 9- Which of the following is most difficult to fabricate in an IC?
 - a- Diode
 - b- Transistor
 - c- FET
 - d- Capacitor
- 10- ICs are used in
 - a- Linear devices only
 - b- Digital devices only
 - c- Both linear and digital devices
 - d- None of the above

الفصل الثامن: الدارات المنطقية التركيبية Combinational logic circuits

تصنف الدارات المنطقية بحسب نوعها أو آلية عملها بواحدة من المجموعتين التاليتين:

- مجموعة الدارات المنطقية التركيبية Combinational Logic Circuits

- مجموعة الدارات المنطقية التتابعية Sequential Logic Circuits .

تعرف كل دائرة منطقية تركيبية (تجميعية)، بأنها جملة مكونة من مجموعة عناصر منطقية موصولة مع بعضها البعض لتحقيق تابع منطقي. لا تحتوي الدارات المنطقية التركيبية على أي عنصر تخزين، وبالتالي تتحدد قيم مخارج الدارات التركيبية في أي لحظة بالقيم الموجودة على مداخلها بتلك اللحظة. بمعنى عندما تتغير قيم الدخل يتغير الخرج مباشرة. يمكن بناء على تعريف الدارة المنطقية التركيبية الحكم على أن كل بوابة منطقية من البوابات التي سبق وأشرنا لها، هي دائرة منطقية من النوع التركيبي. أما الدارات المنطقية التتابعية فتحتوي على عناصر ذاكرة ويقاد عملها بالساعة Clock، وبالتالي نميز بالنسبة للدارات

التتابعية بين كل من: - الدارات التتابعية المتزامنة Synchronous Circuit

- والدارات غير المتزامنة Asynchronous circuit .

تختلف آلية عمل الدارات المنطقية التتابعية عن الدارات المنطقية التركيبية، بأن قيمة خرج الدارة المنطقية التركيبية يعتمد على القيم اللحظية الموجودة على الدخل، بينما خرج الدارة التتابعية لا يعتمد على قيم مداخل الدارة فقط وإنما يأخذ بعين الاعتبار الحالة السابقة للدارة.

8-1 تصميم الدارات المنطقية التركيبية

يمكن تصميم أي دائرة منطقية من النوع التركيبي بطريقتين أساسيتين، هما:

الأولى: بمعونة الحاسب باستخدام إحدى لغات توصيف العتاديات، وهذا ما سنتطرق له بالفصل الثالث عشر.

الثانية: باتباع خطوات محددة وفق الشكل التقليدي.

بالطبع كلتا الطريقتين تؤدي إلى نفس النتيجة، إلا أن الطريقة الأولى أحدث، وتساعد في اختبار التصميم قبل تنجزيه.

أما الطريقة الثانية فتمر بسلسلة من الخطوات التي يجب اتباعها بتسلسل مّحدد، وهي:

- توصيف عمل الدارة المطلوب تصميمها بتحديد متحولات الدخّل ومتحولات الخرج وبناء جدول الحقيقة المقابل الذي يحدّد علاقة الخرج بالدخّل. تتوقف صحة التصميم على هذه النقطة الهامة، بمعنى إن لم يكن فهم المسألة تاماً، ولم توصف بالشكل الصحيح ستكون بقية الخطوات مغلوطة أيضاً.

- صياغة التابع المنطقي الذي يقابل المسألة من جدول الحقيقة الذي يتم التوصل إليه، بصيغة الإمكانية (الشكل الجامع)، أو عدم الإمكانية (الشكل الضارب).

- اختصار التابع المنطقي الناتج بوحدة من طرق الاختصار إلى أبسط شكل ممكن.

- رسم الدارة المنطقية وفق أحد الأشكال الممكنة بأقل عدد ممكن من المستويات. وفيما يلي نستعرض مثلاً يبيّن مراحل التصميم أعلاه، ويصلح للتعميم على تصميم أي دارة منطقية من النوع التركيبي.

مثال: صمم الدارة المنطقية التي تشير للأعداد الأولية فقط من جملة الأعداد العشرية 0 حتى 15 (علماً أن 0 و 1 غير أولية، أما العدد 2 فهو أول عدد أولي)

الجواب: يكون التصميم باتّباع الخطوات المشار إليها أعلاه وفق التالي:

1- بناء جدول الحقيقة: يبنى جدول الحقيقة المقابل لهذه المسألة انطلاقاً من أعلى عدد عشري معطى هو 15 وأنه يقابل في النظام الثنائي التركيبية $(1111)_2$ ، بالتالي يلزم لتمثيل الدخّل أربع متحولات. بفرض أن هذه المتحولات المنطقية الأربعة على الدخّل هي: a, b, c, d وأن الخرج هو y ، يكون جدول الحقيقة المقابل، مكوناً من $2^4 = 16$ سطرًا. تسند قيمة واحد لـ y مقابل كل تركيبة تعادل قيمتها عدداً أولياً من الأعداد المحددة ضمن المجال 0 حتى 15 (أي أن التابع محقق) ونصيغ مجمل هذه المعطيات وفق ما هو مبين بالجدول 8-1.

2- يكتب من جدول الحقيقة الناتج، التابع المنطقي المقابل بإحدى صيغ التابع المنطقي (الجامع أو الضارب)، حيث بإعتماد الشكل الجامع، يكون التابع المقابل لجدول الحقيقة بالصيغة المختصرة وفق التالي: $y = \sum_m(2, 3, 5, 7, 11, 13)$ ويمكن نشر هذا التابع كالتالي: $y = a'b'cd' + a'b'cd + a'bc'd + a'bed + ab'cd + abc'd$

الجدول 1-8: الأعداد الأولية ما بين 0 والـ 15

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

يتضح من شكل التابع ان الشبكة المنطقية التي تقابل التركيبية القياسية AND – OR للتابع الناتج تكون وفق ما هو مبين في الشكل 1-8، أي أنه يلزم لرسم الشبكة بمستويين أربع بوابات NOT وستة بوابات AND كل منها بأربع مداخل وبوابة OR بستة مداخل، وهي شبكة مكلفة نسبياً، لذلك يفضل اللجوء إلى الخطوة الثالثة حول اختصار التابع.



الشكل 1-8: الدارة المنطقية للتابع الذي يحدّد الأعداد الأولية من 0 حتى 15.

3- تختصر الصيغة التي تم التوصل إليها للتابع إلى أبسط شكل ممكن إن كان قابلاً للاختصار، حيث بالنسبة لهذا المثال، يمكن باستخدام خارطة كارنوف

تمثيل التابع المعطى وفق المخطط التالي:

ab \ cd	00	01	11	10	
00	0	0	1	1	1
01	0	1	1	0	3
11	0	1	0	0	4
10	0	0	1	0	2

واختصاره إلى الشكل التالي: $y = a'b'c + b'cd + a'bd + bc'd$ ، قد يكون للتابع شكل مختصر آخر مكافئ لهذا الشكل يحدد بالحلقات التي تعتمد بالمخطط للاختصار. بكل الحالات يضم الشكل المختصر أربع حدود فقط (عوضاً عن ستة) وكل حد مكون من ثلاث متحولات (عوضاً عن أربعة) وهذا يعني أن الشكل المختصر للتابع يكفيهِ ثلاث بوابات NOT وأربع بوابات AND كل منها بثلاث مداخل وبوابة OR بأربع مداخل لرسم الدارة المناسبة. يمكن رسم الشبكة المنطقية المقابلة للشكل المختصر بمستويين بنفس الأسلوب المشار إليه أعلاه.

8-2 تطبيقات متفرقة باستخدام الدارات المنطقية التركيبية

يصادف في التطبيقات العملية كثيراً من العناصر المنطقية التي توجد عادة بشكل قياسي والذي كل منها يقوم بتنفيذ وظيفة محدّدة، ويصنع على شكل دارات متكاملة للاستخدامات العامة. وتتشكل هذه العناصر المكونات الرقمية الأساسية Digital Components لأي نظام رقمي. سنحاول بهذا الفصل التعرف على بعض هذه العناصر واسعة الانتشار والتي كل منها يمثل دائرة منطقية من النوع التركيبي، مثل المرمز للترميز والمقارن للمقارنة وغيرها. أما بالنسبة للتطبيقات الخاصة غير القياسية نتبع خطوات التصميم المشار إليها أعلاه لتصميم الدارة المناسبة، وهذا ماسنوضحه من خلال بعض التطبيقات التي تبين مراحل الوصول إلى المخطط المنطقي لكل منها.

أولاً- مولد وفاحص النّدية

بالنظر إلى إن مفهوم مولد أو فاحص النّدية يعتمد على مفهوم التابع الفردي، نشير بداية إلى مفهوم كل من التابع الفردي والتابع الزوجي Odd & Even Function، انطلاقاً من عملية XOR. بما أن عملية الـ XOR هي تبادلية وتجميعية، هذا يعني أنه يمكن تقييم XOR بثلاثة متحولات وفق أي ترتيب، لأنه يمكن مبادلة مدخلي XOR من دون أن تتأثر نتيجة العملية. يمكننا التعبير عن ذلك رياضياً وفق التالي:

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

وسيكون 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 بالقيمة 1. بمعنى أنه في الحالة التي يكون فيها عدد المتحولات ثلاثة أو أكثر، يجب أن يكون عدد المتحولات التي تساوي 1 فردياً. وهذا يقود إلى التأكيد بأن العملية XOR المتعددة المتحولات تمثل تابعاً فردياً. يتضح أيضاً أن عدد الوحدات في كل دليل حالة معرّف به التابع XOR هو دوماً فردي وعدد الوحدات بالحدود الصغرى المقابلة لدلائل الحالة غير المتضمنة في التابع، أي (0, 3, 5, 6) هو دوماً زوجي. أي أن متمم التابع الفردي هو تابع زوجي. فمثلاً، تابع XOR بأربعة متحولات سيضم 16 حداً أصغرياً، نصفها (1, 2, 4, 7, 8, 11, 13, 14)، يمتلك عدداً فردياً من الوحدات ونصف دلائل الحالة، أي (3, 5, 6, 8, 9, 10, 12, 15) سيتملك عدداً زوجياً من الوحدات. تستخدم بوابات XOR في المنظومات التي تتطلب كشف الخطأ، بتوليد بتّ للنّدية، وتسمى الدارة التي تولد بتّ النّدية لدى المرسل بمولد النّدية Parity Generator (PG) وتسمى الدارة التي تفحص الرسالة مع النّدية بالجبهة المقابلة لدى المستقبل، بفاحص النّدية Parity Checker (PC). لتصميم مثل هذه الدارات، يمكن الانطلاق من مثال بسيط، يتمثل برسم الدارة المناسبة لتوليد النّدية. وهنا يجب أن يتم التمييز بين النّدية الزوجية والنّدية الفردية. لتصميم دارة لتوليد نّدية زوجية مثلاً، لمراقبة معطيات بطول 3bits فقط وأخرى لفحصها، يجب الانطلاق من أن المعطيات المراد إرسالها (أو تخزينها في الذاكرة) هي من الشكل abc، بالتالي المتحولات a,b,c تصبح بمثابة مداخل

للدائرة المطلوبة وبتّ النّدية PG هو خرجها. وباعتماد النّدية الزوجية، يصبح جدول الحقيقة لمولد 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 أن المعطيات على الدخّل يمكن أن تكون بأيّ تركيبة من 000 حتى 111 يفترض وضع الخرج PG بالقيمة 0 أو 1، بحيث يكون العدد الكلي للوحدات بكلّ تركيبة بما في ذلك قيمة PG زوجياً. وبكتابة التابع المنطقي $PG(a,b,c)$ من جدول الحقيقة أعلاه يمكن الحصول على الصيغة التالية:

$PG = a'b'c + a'bc' + ab'c' + abc = a'(b'c+bc') + a(b'c'+ab)c = a \oplus b \oplus c$
يسمى هذا التابع الناتج أيضاً بتابع النّدية الفردية، لأن قيمة التابع PG هي دوماً 1 عند كل حد تتضمن قيمته الثنائية عدداً فردياً من الوحدات. ويكون المخطط المنطقي المقابل لمثل هذا التابع بوصفه مولد ندية زوجية، وفق ما هو مبين في الشكل 8-2 أ.

الشكل 8-2: المخطط المنطقي لمولد وفاحص النّدية

يرسل بتّ الندية المولد بالمنبع Source (سواء أكان زوجياً أم فردياً) مع الرسالة الأساسية إلى الهدف Destination. والرسالة التي يتم استلامها بطول 4 bits تدخل إلى دائرة فاحص الندية للكشف عن وقوع أي خطأ. وبالنظر إلى أن المعطيات مرسلّة بنّدية زوجية، سيكون خرج فاحص الندية PC بحالة الواحد في حال وقوع عدد فردي من الأخطاء (والخرج سيتممه إلى زوجي). وهكذا جدول الحقيقة لفاحص الندية سيكون مكوناً من أربعة مداخل وخرج وحيد، وفق ما هو مبين في الجدول 2-8.

الجدول 2-8: جدول الحقيقة لفاحص الندية (الزوجية) لرسالة مكونة من 4 bits

القسم الأول من الجدول			القسم الثاني من الجدول		
التراكيب الأساسية بعدد زوجي من الوحدات			التراكيب الإضافية الجديدة بعدد فردي للوحدات		
دليل الحالة	البتات المستقبلية	فاحص الندية	دليل الحالة	البتات المستقبلية	فاحص الندية
S	a b c PG	PC	S	a b c PG	PC
0	0 0 0 0	0	1	0 0 0 1	1
3	0 0 1 1		2	0 0 1 0	
5	0 1 0 1		4	0 1 0 0	
6	0 1 1 0		7	0 1 1 1	
9	1 0 0 1		8	1 0 0 0	
10	1 0 1 0		11	1 0 1 1	
12	1 1 0 0		13	1 1 0 1	
15	1 1 1 1		14	1 1 1 0	

أي من أجل ثماني كلمات مرسلّة سيتم فحص ثمانية كلمات أخرى جديدة (أي ضعف العدد). وهي الكلمات التي يكون فيها عدد الوحدات فردياً. يمثل الجدول الجديد التابع المنطقي: $PC = a \oplus b \oplus c \oplus PG$ ، ويقابل المخطط المنطقي المبين بالشكل 2-8 ب، حيث يمكن جعل الدارة في هذا الشكل لتوليد الندية، إذا وصل المدخل PG إلى الصفر (وتصبح نفس الدارة صالحة لتوليد الندية وفحصها).

ثانياً- إظهار الأرقام العشرية بوساطة السبع مقاطع المضيئة

لتصميم الدارة التي تعمل على إظهار الأرقام العشرية (من 0 وحتى 9) بوساطة العناصر ذات السبع قطع المضيئة Seven Segments وفق ما هو مبين بالشكل (a) أدناه، نتابع تحليل المسألة وفق التالي:



نعلم أنه حتى يضيء الرقم 6 مثلاً، يجب أن تضيء المقاطع المبينة في الشكل (b)، أي جميع المقاطع عدا المقطعين B و A ($A = B = 0$). ولذلك أول خطوة بتصميم الدارة المطلوبة، هي رسم جدول الحقيقة الذي يوصف ما هو مطلوب، بحيث إذا وضع أحد المقاطع بالقيمة 0، يعني أنه في حالة OFF مطفي (غير مضاء). وبما أنه يلزم لتمثيل الأرقام العشرية بشكل ثنائي أربعة متحولات منطقية، وأن كل رقم مكون من سبعة مقاطع، فإن كل مقطع على الخرج من هذه المقاطع السبعة سيمثل تابع منطقي بأربعة متحولات. بفرض أن متحولات الدخل هي X_0, X_1, X_2, X_3 . وأن توابع الخرج السبعة المقابلة هي: A, B, C, D, E, F, G. يكون جدول الحقيقة للتطبيق المعطى وفق ما هو مبين في الجدول 3-8. تم في الجدول إضافة عمود للتعبير عن كل رقم بما يقابله بالستة عشري (1 Byte)، بحيث إن الرقم 5 مثلاً يقابل القيمة 6DH. لإضاءة رقم ما، يوضع بالجدول أدنا مقابل التركيبية الثنائية التي تمثلها على الدخل القيمة 1 مقابل كل مقطع من المقاطع السبعة التي يفترض أن تضاء بالرقم. فمثلاً، الرقم العشري 6 يمثل على الدخل بشكل ثنائي على أربع خانات بالشكل (0110). وبالتالي لإظهار مثل هذا الرقم على الخرج (المقاطع السبعة)، يجب وضع التركيبية $7CH = (0111\ 1100)_2$ على المقاطع والتي تعني إضاءة جميع المقاطع باستثناء المقطعين A و B.

الجدول 3-8: جدول الحقيقة لإظهار الأرقام العشرية على سبع مقاطع مضينة

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

يمثل بمخطط كارنوف التابع المنطقي المقابل لكل مقطع من المقاطع السبعة على حدة من الجدول بقصد الاختصار. فمثلاً، يمكن تمثيل التابع المقابل للمقطع A كالتالي:

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

		X_1X_0			
X_3X_2		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 بأربعة متحولات على الدخول يمكن تمثيل 16 تركيبة مختلفة، يلزم منها فقط عشرة تراكيب، لذلك يلاحظ بمخططات كارنوف أن الأعداد العشرية من 10 حتى 15 تقابل حالات عدم اكتراث (تعيين) d. وباختصار التابع $A(X_0, X_1, X_2, X_3)$ نجد أن: $A(X_0, X_1, X_2, X_3) = A = X'_2X'_1 + X_0$. بنفس الأسلوب يتم تمثيل التوابع الستة المتبقية بمخططات كارنوف، وفق ما هو مبين أدناه. يكتب الشكل المختصر لكل تابع على حدة، للحصول على التوابع الستة المتبقية وهي:

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

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

$X_3X_2 \backslash X_1X_0$		X_1X_0			
		00	01	11	10
X_3X_2	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_1X_0			
		00	01	11	10
X_3X_2	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	

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

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

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

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

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

$$B = X_2 + X_0 X_1 + X'_0 X'_1;$$

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

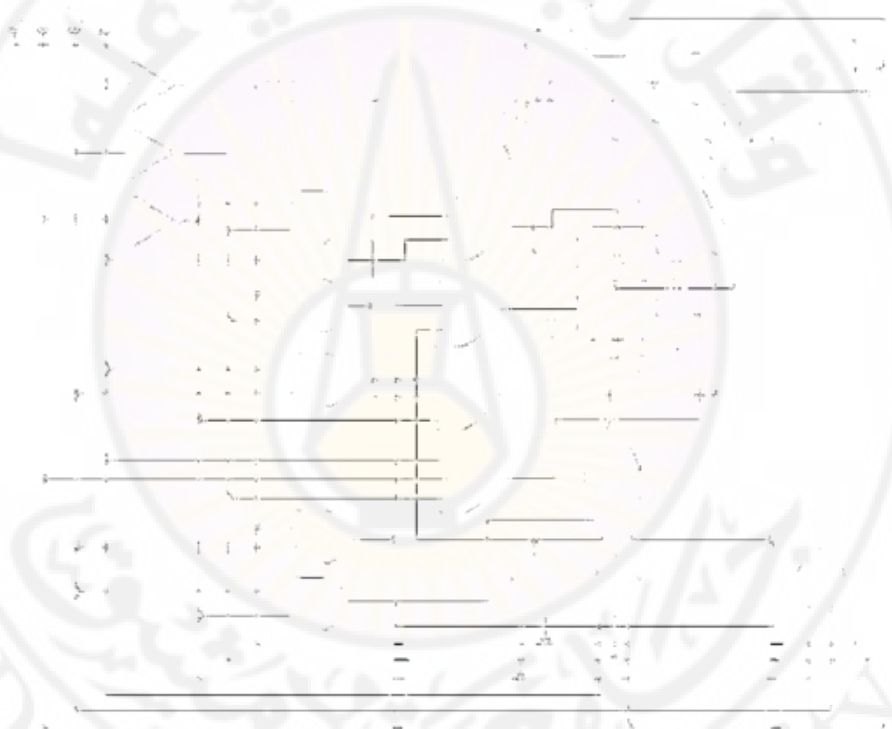
$$D = X'_0 X'_2 + X'_0 X_1 + X_1 X'_2 + X_0 X'_1 X_2;$$

$$E = X'_0 X_1 + X'_0 X'_2;$$

$$F = X'_0 X'_1 + X'_0 X_2 + X'_1 X_2 + X_3;$$

$$G = X'_0 X_1 + X_1 X'_2 + X'_1 X_2 + X_3$$

ويرسم هذه المعادلات السبع تكون الدارة المطلوبة، وفق ما هو مبين في الشكل 8-3.



الشكل 8-3: دائرة إظهار الأرقام العشرية على عنصر إظهار من سبعة مقاطع

3-8 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. ويقال عن الدارة الرقمية التي تقوم بجمع خانتين ثنائيتين فقط، إنها 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". تصنع مثل دارات الجمع والطرح على شكل دارات متكاملة قياسية، وسنحاول فيما يلي التعرف على بعض هذه الدارات القياسية الواسعة الانتشار، ابتداء من الدارات الخاصة بعمليات الجمع والطرح. وسنقوم خلال التعريف بالدارة، الإشارة إلى أربعة جوانب أساسية لكل دارة على حدة، هي: - الرمز المنطقي Logic symbol.

- جدول الحقيقة Truth table.

- التابع المنطقي Logic function.

- والمخطط المنطقي Logic diagram.

1-3-8 نصف الجامع Half Adder

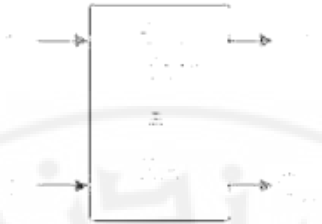
يمثل نصف الجامع أو الجامع البسيط HA وحدة رقمية لها مدخلان فقط لإدخال المعطيات الثنائية المراد جمعها ومخرجان أحدهما يمثل المجموع S والآخر يمثل المنقول C_0 (إن كان هناك من نقل) للمرحلة التالية، وبالتالي نصف الجامع يمثل دارة تقوم بجمع القيم على المدخل بشكل حسابي. ويظهر الشكل 4-8 الرمز المنطقي لنصف الجامع. ترمز المتحولات الثنائية a, b للقيم المراد جمعها ويرمز للخروج بالمتحولين:

- S لتمثيل المجموع (Summation).

- C_{out} Carry Out اختصاراً C_0 لتمثيل المنقول.

لمعرفة التركيبة الداخلية لدارة نصف الجامع، تعتمد نفس الأسس المتبعة بتصميم أي دارة منطقية تركيبية، ابتداءً من تحديد جدول الحقيقة الذي يظهر علاقة المخارج بالمداخل وصولاً

إلى رسم المخطط المنطقي.



الشكل 8-4: الرمز المنطقي لنصف الجامع

يظهر الجدول 8-4 جدول الحقيقة لنصف الجامع.

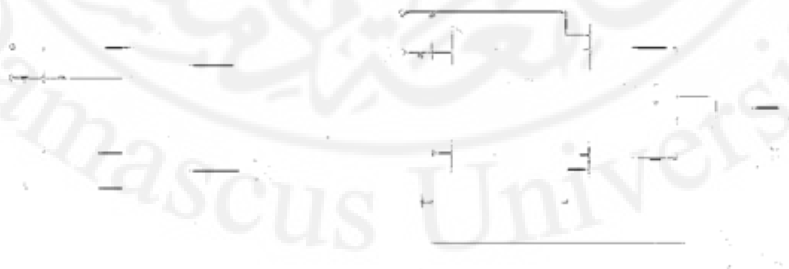
الجدول 8-4: جدول الحقيقة لدارة نصف جامع

Inputs		Outputs	
a	b	S	C _o
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

ومنه يتضح أن التوابع المنطقية المقابلة له هي: $S = ab' + a'b = a \oplus b$

$$C_o = ab$$

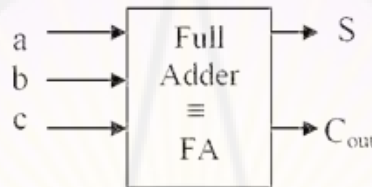
المخطط المنطقي الإجمالي الذي يقابل المعادلتين أعلاه وبالتالي بنية الصندوق الأسود يكون وفق ما هو مبين في الشكل 8-5.



الشكل 8-5: المخطط المنطقي لنصف الجامع

8-3-2 Full Adder الجامع التام

يمثل الجامع التام أو الكامل FA وحدة رقمية بثلاثة مداخل لإدخال المعطيات، اثنتين منهما a, b لإدخال البيانات الخارجية والثالث يمثل المنقول من المرحلة السابقة (CI اختصاراً c) الذي ينتج من عملية جمع سابقة. ويرمز لمخارج الجامع التام بنفس رموز مخارج نصف الجامع، ويصبح الرمز المنطقي للجامع التام، وفق ما هو مبين في الشكل 8-6. وبما أن لمخارج الجامع التام نفس المفهوم لدى نصف الجامع، يكون جدول الحقيقة للجامع التام وفق ما هو مبين بالجدول 8-5.



الشكل 8-6: الرمز المنطقي للجامع التام

الجدول 8-5: جدول الحقيقة 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

يمكن من جدول الحقيقة للجامع التام كتابة التوابع المنطقية S و C_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 وإمكانية اختصار C_o بمتحول واحد من حدين. يعدل

الشكل العام لكل من التابعين أعلاه إلى الصيغة التالية:

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

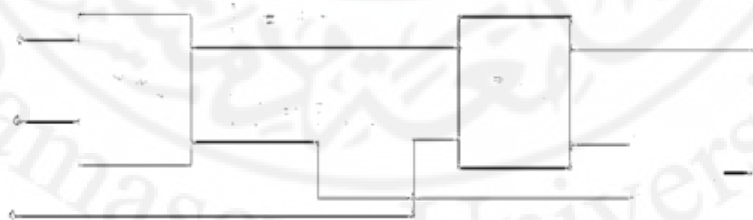
$$C_o = (a'b + ab')c + ab(c' + c) = (a \oplus b) \cdot c + ab$$

بحيث يصبح من السهل تذكرهما ويكون المخطط المنطقي المقابل للجامع التام، وفق ما هو مبين في الشكل 8-7. تم بالقسم الأيسر من الشكل الإشارة إلى رمز كل من المتحول البيني $P_i = a_i \oplus b_i$ والذي يسمى ناشر المنقول Carry propagate والمتحول G_i الذي يسمى مولد المنقول Carry generate لتوضيح مفهوم المنقول المستبق Carry Look ahead كما سنجد لاحقاً.



الشكل 8-7: المخطط المنطقي للجامع الكامل

يمكن من نصفي جامع وبوابة OR تشكيل جامعاً كاملاً وذلك بربط نصفي جامع ببعضهما، وفق ما هو مبين في الشكل 8-8، بحيث يحافظ على قيم التتابع S و C_o من دون تغيير.



الشكل 8-8: تشكيل دائرة جامع كامل باستخدام دارتي نصف جامع

3-3-8 الجامع الثنائي المتوازي Parallel binary adder

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

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

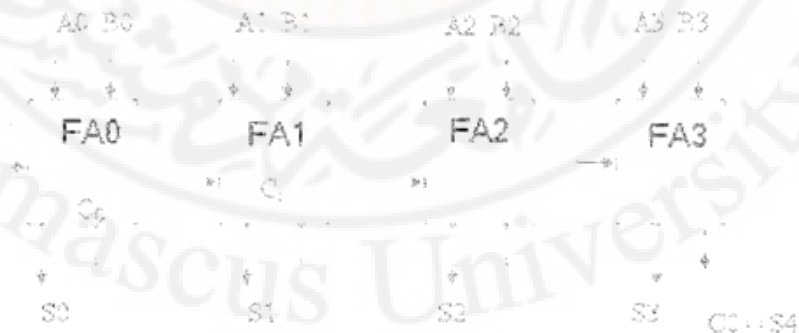
الجواب: بالإطار العام من أجل عددين A و B طول كل منهما 4 bits مثلاً، سيكون حاصل جمعهما معاً عدداً بخمسة بتات وفق الشكل التالي:

$$A = A_3A_2A_1A_0 = (13)_{10} = (1101)_2$$

$$B = B_3B_2B_1B_0 = (11)_{10} = (0111)_2$$

$$A+B = S_4S_3S_2S_1S_0 = (24)_{10} = (11000)_2$$

يكون عدد خانات الناتج النهائي أكبر بخانة واحدة عن عدد خانات العدد الأطول من بين العددين المراد جمعهما، وتستخدم هذه الخانة لتمثل المنقول الأخير. وبالتالي يلزم نصف جامع للخانتين الأوليتين وثلاثة جوامع كاملة لبقية الخانات، وينتج جامع على التوازي بطول 4 bits، وفق ما هو مبين في الشكل 8-9. لتوحيد العناصر المستخدمة في بنية الجامع المتوازي يمكن عوضاً عن نصف الجامع الأول استخدام جامع كامل بعد جعل منقول الدخل العائد له صفر $C_1 = 0$.



الشكل 8-9: دارة جامع ثنائي متوازي بطول 4 bits

4-8 دارات الطرح Subtraction circuits

بنفس الآلية التي تم التعرف بها على دارات الجمع بأنواعها الثلاثة، يمكن التعرف فيما يلي على دارات الطرح بأنواعها الثلاثة أيضاً مع فارق واحد بأنه في عملية الطرح يستخدم عوضاً عن مفهوم المنقول Carry بالجمع، مفهوم اقتراض Borrow.

أولاً: نصف الطارح Half Subtractor

يظهر الشكل 8-10 الرمز المنطقي لنصف الطارح HS، حيث الرمز D يعني الفرق (Difference)، أما B_{out} (Borrow out) فتعني الاستعارة.

الشكل 8-10: الرمز المنطقي لنصف الطارح

يظهر الجدول 8-6 جدول الحقيقة لنصف الطارح

الجدول 8-6: جدول الحقيقة لنصف الطارح

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

يمكننا من الجدول أعلاه كتابة التوابع المنطقية العائدة لنصف الطارح وفق التالي:

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

$$B_o = a'b$$

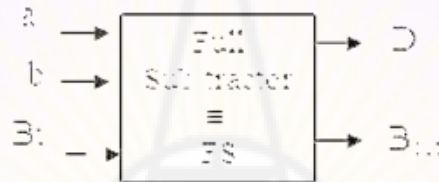
وهي مثل معادلات نصف الجامع مع فارق واحد هو أن قيمة الاقتراض تكون لدى نصف الطارح $B_o = a'b$ بدلاً من قيمة النقل $C_o = ab$ لدى نصف الجامع، حيث إن التركيبية 1 - 0 تقابل الناتج $D = 1$ والإعارة $B_o = 1$ ، وتصبح العملية بعد اقتراض واحد، وكأنها

$D = 10 - 1 = 1$. ويكون المخطط المنطقي وفق الشكل 8-11.

الشكل 8-11: المخطط المنطقي لنصف الطارح HS

ثانياً: الطارح التام Full Subtractor

يظهر الشكل 8-12 الرمز المنطقي للطارح التام، والجدول 8-7 جدول الحقيقة العائد له



الشكل 8-12: الرمز المنطقي للطارح التام

الجدول 8-7: جدول الحقيقة للطارح التام

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

بأسلوب مماثل لاستنتاج معادلات الجامع التام تكون المعادلات المنطقية للطارح التام

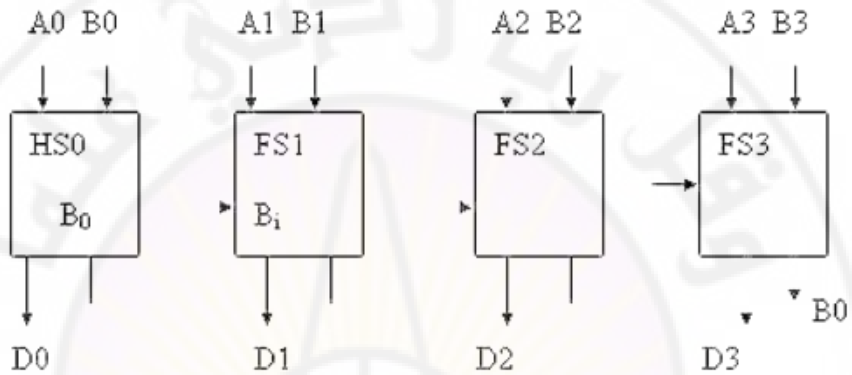
$$D = a'b'B + a'bB' + ab'B' + abB = a \oplus b \oplus B \quad \text{وفق التالي:}$$

$$B_o = a'b'B + a'bB' + a'bB + abB = (a \oplus b)' \cdot B + a'b$$

يمكن من هذه المعادلات رسم المخطط المنطقي لدارة الطارح التام بسهولة.

ثالثاً: الطارح الثنائي المتوازي Parallel binary subtractor

يبدو الشكل العام لطارح متوازي بطول 4 bits مثلاً، كما هو مبين في الشكل 8-13. وهو يماثل المخطط المنطقي للجامع على التوازي باستثناء استخدام FS عوضاً عن FA في المخطط وعوضاً عن مفهوم المنقول يستخدم مفهوم الإعارة.



الشكل 8-13: المخطط الصندوقي للطارح المتوازي بطول 4 bits

5-8 الجامع / الطارح الثنائي المتوازي

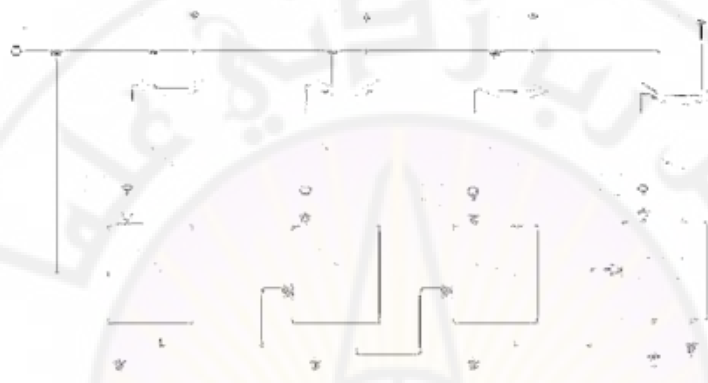
Parallel binary adder / subtractor

بالنظر إلى أن عمليتي جمع أو طرح عددين ثنائيين، تعد من أكثر العمليات الحسابية أهمية وهي متلازمة في كثير من العمليات التي تنفذ بالحواسيب الرقمية خلال معالجة المعطيات، فإنه عوضاً عن تصميم دارات خاصة بالجمع وأخرى خاصة بالطرح على التوازي، غالباً ما يتم تصميم دارات للجمع والطرح على التوازي بدارة واحدة وبالطول المناسب، تسمى بدارات الجامع/الطارح على التوازي. انطلاقاً من أن طرح أي عددين تعني جمع المطروح منه مع المتمم للاثنيين للمطروح، يمكن تحويل عملية الطرح إلى جمع، حيث من أجل عددين A و B كل منهما 4 bits مثلاً، تصبح عملية الطرح كالتالي:

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

يلغى V_4 لأن الجمع يتم مع المتمم للواحد (أي طرح). وانطلاقاً من أن: $B \oplus 0 = B$ و $B \oplus 1 = B'$ ، يمكن باستخدام بوابات XOR تحقيق دارة جمع / طرح بطول 4 bits،

وفق ما هو مبين في الشكل 8-14. تم في دائرة الجامع / الطارح وصل مكونات العدد الأول A مباشرة إلى دائرة الجامع التام. أما المدخل الثاني للجامع المخصص لـ B فيتم وصله مع إشارة التحكم C_i عن طريق بوابة XOR لتحقيق الوظيفة المزدوجة للدائرة.



الشكل 8-14: دائرة جامع طارح على التوازي بطول 4 bits

تتحكم إشارة التحكم الخارجية C_i في الدائرة بتحديد وظيفة الدائرة. فإذا كانت $C_i = 0$ مثلاً، تكون الدائرة للجمع، لأن بوابة XOR تكون وكأنها غير موجودة وأن قيمة المنقول الداخل في دائرة الجامع التام الأول صفر أيضاً. أما إذا كانت $C_i = 1$ ، فتتحول وظيفة الدائرة إلى الطرح، لأن بوابة XOR تؤمن المتمم للواحد للعدد B ليتم جمعه مع العدد A. ولتحويل المتمم للواحد إلى متمم للثنتين، فإن الواحد اللازم لذلك موجود على مدخل أول جامع تام بالدائرة ويجمع مع أول خانة من النتيجة. ويمكن تشكيل مثل هذه الدائرة بالطول المطلوب، وهنا تجدر الإشارة إلى أنه خلال التعامل مع جامع / طارح على التوازي بأي طول كان، يمكن

مصادفة موضوعين متباينين، هما: - الأول يتعلق بانتشار المنقول
- والثاني يتعلق بالطرح أو الفيض

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

أولاً- انتشار المنقول Carry propagation

يعدّ زمن انتشار المنقول عاملاً يحدّ من سرعة جمع عددين بدارة الجامع/الطارح على التوازي، لأن النتيجة لن تكون صحيحة ما لم تعطى الإشارات زمناً كافياً يسمح لها بالانتشار عبر البوابات ما بين مداخل الدارة ومخارجها. ولإيضاح، نبيّن فيما يلي آلية عمل الدارة في الشكل 7-8 من خلال عملية الجمع، وفق التالي: بعد وصل جميع بتّات المضاف Addend والمضاف إليه Augends إلى مداخل الدارة، يجب أن تنتشر الإشارة عبر البوابات المكونة للدارة قبل أن يصبح المجموع على المخارج صحيحاً. وزمن انتشار الإشارة في الدارة هو الزمن الذي يأخذه المنقول لينتشر خلال جميع الجوامع الكاملة المكونة للدارة. وبما أن كل بتّ من بتّات الناتج يعتمد على قيمة منقول الدخل فان قيمة أي مرحلة من مراحل الجامع، مثل $S_i = S_3$ ، ستكون في حالتها المستقرة النهائية بعد أن ينتشر منقول الدخل لهذه المرحلة مباشرة، بمعنى رغم وجود A_3 و B_3 على مدخلي الجامع الثالث فإن منقول الدخل C_3 لن يثبت على قيمته النهائية قبل توافر المنقول C_2 من المرحلة السابقة وبالمثل من أجل C_2 يجب الانتظار حتى يستقر C_1 وهكذا دواليك حتى يتم الوصول إلى C_0 ، أي أن الناتج النهائي لن يستقر على القيمة الصحيحة إلا بعد أن يتموج Ripple المنقول عبر جميع المراحل. ولما كانت جميع العمليات الحسابية الأخرى تتجزّ بالجمع المتتالي، فان الزمن اللازم لعملية الجمع يكون حرجاً جداً. بالتالي لابد من إنقاص زمن انتشار المنقول. وتوجد عدة تقنيات لخفض زمن انتشار المنقول في الجامع، والطارح المتوازي، من أهمها: - استخدام بوابات منطقية سريعة.

- استخدام مبدأ المنقول المستبق

ولتوضيح مبدأ طريقة المنقول المستبق Carry Look ahead، نعود للشكل الذي يمكن أن يمثل دارة جامع تام بالمرحلة i من جامع على التوازي، حيث يمكن التعبير عن خرج المجموع S والمنقول C لأي جامع تام (i) ضمن الجامع المتوازي بالعلاقتين التاليتين:

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

وبكتابة التوابع المنطقية التي تمثل مخارج المنقول لكل مرحلة بالجامع المتوازي على حدة،

وتعويض كل منقول C_i بقيمته، يمكن الحصول على ما يلي:

$$C_0 = \text{Input Carry}$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 (G_0 + P_0 C_0) = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 (G_1 + P_1 G_0 + P_1 P_0 C_0) = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0$$

وهكذا تستمر العملية بحسب عدد خانات الأعداد المراد جمعها. ينتج مولد المنقول G_i منقولاً بقيمته واحد عندما يكون $a_i = b_i = 1$ بغض النظر عن قيمة منقول الدخل C_i . وبرسم التوابيع المنطقية الممثلة لـ C_i يمكن الحصول على المخطط المبين بالشكل 8-15، وفيه يظهر أن جميع توابيع المنقول عبر عنها بصيغة مجموع جداءات. وهذا يعني أن كل تابع سينفذ بمستوى واحد من بوابات AND متبوعاً ببوابة واحدة من نوع OR. تمثل الدارة في الشكل 8-15 دارة مولد المنقول المستبق، وهي تظهر مثلاً، أن المنقول C_3 ليس بحاجة لانتظار انتشار المنقول الذي يسبقه، أي C_2 أو C_1 ، لأنه في حقيقة الأمر C_3 ينتشر بنفس الوقت الذي ينتشر فيه كل من C_1 و C_2 .

الشكل 8-15: المخطط المنطقي لمولد المنقول المستبق

يمكن ضمّ الدارة أعلاه للدارة القياسية للجامع على التوازي (بطول 3 bits مثلاً) لجعل زمن تأخير الانتشار لجميع المخارج من S_1 حتى S_3 متساوياً.

ثانياً- الفيض Overflow

عند جمع أو طرح عددين ثنائيين بذاكرة رقمية مثل الجامع / الطارح على التوازي يكون كل من هذين العددين مخزنًا عادةً بسجل. فإذا كان طول سجلات الوحدة المستخدمة بجمع عددين بطول n bits وكان ناتج عملية الجمع بطول $n + 1$ bits ، ويقال إنه حدث فيض Overflow وهذا يعني أن قسماً من النتيجة لا يظهر لعدم قدرة السجلات على استيعاب الناتج. ويتم تجنب هذه الظاهرة غير المرغوب فيها باستخدام سجلات بطول كلمة أكبر ولو بخانة واحدة على الأقل من عدد الخانات اللازمة لتمثيل أكبر قيمة عددية يمكن التعامل معها، أو بإضافة منطق خاص للدارة التي تقوم بعملية الجمع بقصد الكشف عن الفيض عند حدوثه، بحيث يتمكن المستخدم بواسطة هذا المنطق بعد إجراء كل عملية جمع من معرفة ما إن كان قد حدث فيض أم لا، لأخذ ذلك بعين الاعتبار. بالطبع عند جمع عددين أحدهما موجب والآخر سالب لا يمكن أن يحدث فيض، لأن الناتج سيكون دوماً أصغر من العدد الأكبر من العددين الأساسيين. يمكن أن تحدث مشكلة الفيض، عندما يكون العددان المضافان لبعضهما البعض موجبين فقط أو سالبين فقط. وبما أن المنقول في حال جمع أعداد ثنائية أو عشرية موجبة أو سالبة هو دوماً واحد فإنه يمكن تخزين قيمة الفيض في حال حدوثه بذاكرة طولها 1 bit لنتمكن من فحصها بعد كل عملية جمع. تدعى مثل هذه الذاكرة، كما سنجد لاحقاً قلاب Flip-Flop يمر بوحدة من حالتين، إما 0 وتعني Reset أو 1 وتعني Set. وبالنسبة للأعداد غير المؤشرة لا يوجد مشكلة لأن كل البتات المستخدمة بتمثيل العدد تمثل معلومات مفيدة، ولا يوجد أي قيد على تخصيص بت للإشارة. سبق وأشرنا إلى أن مجال الأعداد غير المؤشرة على مستوى 8 bits ، هو من 0 حتى 255، وبالتالي إذا تم جمع قيمتين وكان ناتجهما أكبر من 255 سيحدث فيض.

مثال : ناتج جمع العددين: $200 + 127$ عشرياً هو 327 وهذا يكافئ ثنائياً:

$$\begin{array}{r} 127 = 0111\ 1111 + \\ 200 = 1100\ 1000 \\ \hline 327 = 1\ 0100\ 0111 \end{array}$$

حدث فيض، ويكشف عنه بوجود منقول أخير End carry out من الموضع الأكثر دلالة MSB، وهذا ما سبق وعبرنا عنه بوجود قيمة لـ S_4 في الشكل 14-8. أما بالنسبة للأعداد

المؤشرة، فإن المشكلة هي أن أقصى بتّ من اليسار دوماً يمثل الإشارة، وهو يعامل عند الجمع مع عدد آخر مؤشر معاملة باقي البتّات في الأعداد التي تتم عليها عملية الجمع وكأنه جزء من العدد، وبالتالي المنقول هنا لا يدل على الفيض، لأن العدد السالب يكون أصلاً ممثلاً بمتّمه للاثنتين. وبما أن مجال الأعداد المؤشرة على مستوى 1 byte هو من 128- حتى 127+ فإن ناتج جمع أي عددين، سينتج فيضاً إذا كان خارج هذا المجال. مثال: باعتماد العددين 75 و 75 على أنهما مؤشران مرة إيجاباً ومرة سلباً، سنجد أنه في كلتا الحالتين يوجد فيض، لأن $75 + 75 = +150$ خارج المجال المسموح به على مستوى 8 bits وأن $75 - 75 = -150$ هو أيضاً خارج المجال المسموح به، وهذا ما يمكن إظهاره ثانياً وفق التالي:

$$\begin{array}{rcl} \text{الحالة أ:} & +75 = & 0101\ 0011 \\ & +75 = & 0101\ 0011 \\ \hline & +150 = & 1010\ 0110 \end{array}$$

يلاحظ في الناتج وجود قيمة واحد مكان وجود الإشارة، مما يدل على أن الناتج سالب وهذا مستحيل. إذاً هناك فيض، لأن 8 bits لم تتسع للناتج.

$$\begin{array}{rcl} \text{الحالة ب:} & -75 = & 1010\ 1101 \\ & -75 = & 1010\ 1101 \\ \hline & -150 = & 1\ 0101\ 1010 \end{array}$$

يلاحظ أن ناتج جمع عددين سالبين على مستوى 8 bits هو عدد موجب لوجود قيمة الصفر بالبت الثامن موضع الإشارة، وهذا مستحيل، إذاً هناك فيض أيضاً. كان بالإمكان تدارك هذه المشكلة لو تم تمثيل كل من العددين من البداية بسجلات 9 bits، واعتبار البت التاسع موضعاً للإشارة. يمكن بكلا الحالتين المبينتين أعلاه، الكشف عن حالة الفيض بمراقبة المنقول إلى موضع بتّ الإشارة ومنقول الناتج عن موضع بتّ الإشارة. يحدث الفيض إذا اختلف المنقولان. وللمقارنة بين المنقولين تستخدم البوابة المنطقية XOR وهكذا أصبح بإمكاننا رسم الدارة النهائية لجمع أو طرح أي عددين ثنائيين كل منهما 3 bits من دون إشارة ومع إشارة وبوجود مولد المنقول المستبق ومنطق لكشف الفيض، وفق ما هو مبين في الشكل 8-16.

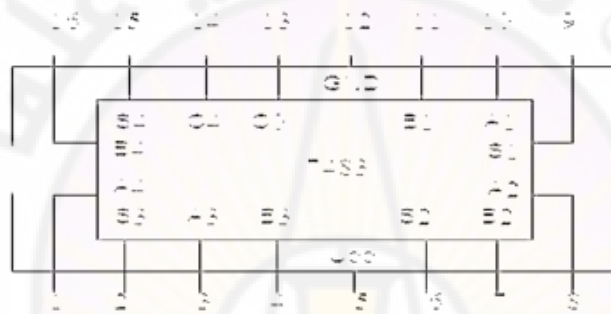


الشكل 8-16: الدارة الكاملة لجامع / طارح ثنائي بطول 3 bits

يتضح من المنطق المخصص لفحص الفيض أن الأعداد المعطاة إذا كانت غير مؤشرة فإن الخرج C يدل على المنقول. أما إذا كان العددين بإشارة، فإن الخرج O هو الذي يحدّد وجود فيض من عدمه، وبالتالي من أجل عددين فيهما الأكبر بطول n bits، سيكون الناتج بطول $n + 1$ bits والبت $n + 1$ هو الإشارة الفعلية بعد أن أزيح باتجاه اليسار. وهنا نشير أيضاً إلى أنه يمكن ببعض التطبيقات البسيطة الاستغناء عن كتلة المنقول المستبق بالنظر إلى أن الوقت اللازم لانتشار المنقول الإيجابي Carry أو السلبي Borrow من وحدة كاملة إلى أخرى قليل جداً نسبة إلى الوقت الذي تستهلكه عملية الطرح أو الجمع.

8-6 تطبيقات على الجمع / الطرح الثنائي

تصنع دارات متكاملة قياسية لانجاز عمليات الجمع والطرح الثنائي، حيث يمكن استخدام الدارة 7483 التي تمثل دائرة جامع/ طارح بطول 4 bits بالتغليف DIL-16 لاجراء عمليات الجمع أو الطرح لأي عددين مهما يكن طول أي منهما وبالتالي رسم المخططات المنطقية المقابلة. وبالنظر للاستخدام الواسع لهذه الدارة، نبين في الشكل 8-17 توزيع الإشارات العائدة لهذه الدارة.



الشكل 8-17: توزيع إشارات الدارة 7483 (إشارات الدخل والخرج والتحكم والتغذية)

تضم الدارة بالإضافة إلى مأخذي التغذية أربعة مأخذ A_1, A_2, A_3, A_4 لتمثيل العدد الأول وأربعة مأخذ B_1, B_2, B_3, B_4 لتمثيل العدد الثاني وأربعة مأخذ للخرج (الناتج) في حالتي الطرح والجمع وهي S_1, S_2, S_3, S_4 ويبقى للدائرة مأخذين هما:

- المأخذ C_4 ، ويمثل قيمة آخر منقول في حالة الجمع إن كان موجوداً وكأنه S_5 (بينما قيم المنقول بين جامع تام وآخر ضمن الدارة لا تظهر في الشكل).

- المأخذ C_0 لتمثيل إشارة التحكم التي تحدد طبيعة عمل الدارة، إن كانت للجمع أو للطرح أي كأنها تمثل الإشارة التي تربط العددين A و B . بمعنى، إذا كانت:

$(+) \sim C_0 = 0$ فإن الدارة تمثل دائرة جامع على التوازي بطول 4 bits، وإذا كانت

$(-) \sim C_0 = 1$ فإن الدارة تمثل دائرة طارح على التوازي بطول 4 bits.

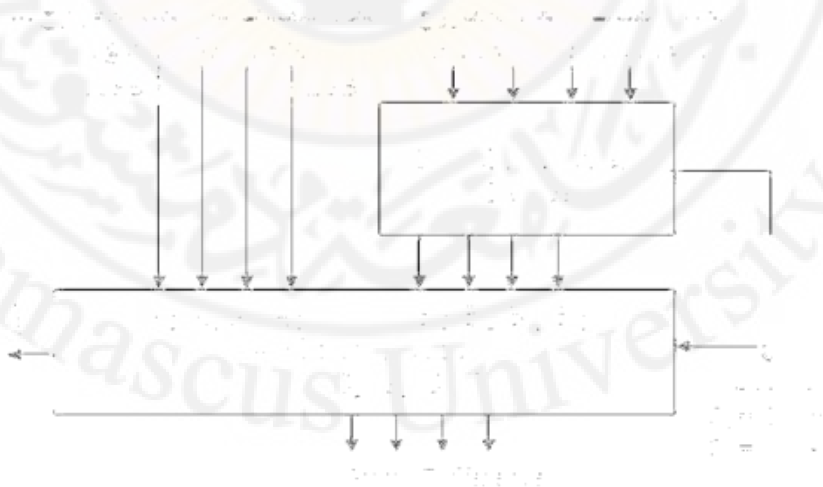
ويقرأ على المخارج S_1, S_2, S_3, S_4 الناتج بكلتا الحالتين.

أولاً: عمليات الجمع والطرح الثنائي

نعرض فيما يلي بعض التطبيقات البسيطة عن عمليات الجمع والطرح على الأعداد الثنائية باستخدام الدارات المتكاملة المناسبة، من خلال الأمثلة وفق التالي:

مثال: صمم الدارة المنطقية التركيبية المناسبة لإجراء عملية الجمع أو الطرح على عددين عشريين، مثل 14 و 5 (أي 14 ± 5).

الجواب: بتحويل الأعداد المعطاة إلى ما يقابلها في النظام الثنائي، يلاحظ وجود اختلاف بأطوال الأعداد المعطاة. وبالتالي ستكون أول خطوة في التصميم، هي مساواة الأعداد بالطول، أي إلى الشكل 0101 ± 1110 الذي يقابل الصيغة: $A = 14 = A_3A_2A_1A_0$ للعدد الأول (1110) و $B = 5 = B_3B_2B_1B_0$ للعدد الثاني (0101)، بحيث يكون المجموع $S = S_4S_3S_2S_1S_0$ (والفرق $D = D_3D_2D_1D_0$). وبالنظر إلى سهولة عملية الجمع، نشير بهذا المثال لعملية الطرح فقط. يتم التعامل خلال عملية الطرح مع المطروح على أنه عدد سالب، ويمثل باستخدام طريقة المتمم لاثنتين. ويتم جمع الناتج مع المطروح منه جمعاً عادياً، حيث يمكن تصور المخطط العام لطرح عددين كل منهما بطول 4 bits، وفق ما هو مبين في الشكل 8-18.



الشكل 8-18: المخطط العام لجمع وطرح عددين ثنائيين بطول 4 bits على التوازي

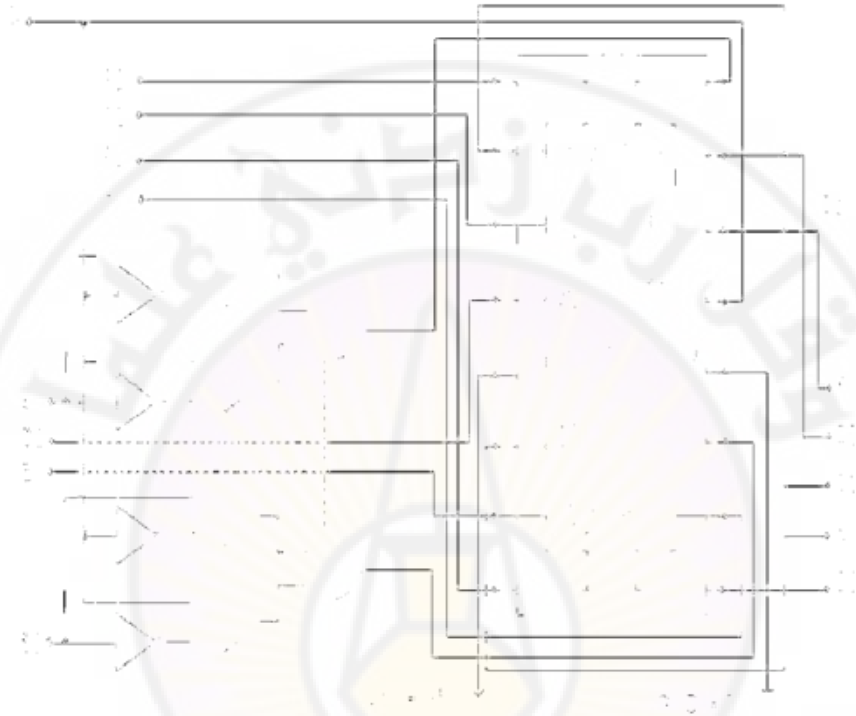
بالطبع إذا زاد طول العددين المراد جمعهما أو طرحهما عن 4 bits، يمكن استخدام دارات بطول كلمة أكبر أو بزيادة عدد الدارات المستخدمة بطول 4 bits بحسب أطوال الأعداد المراد جمعها وربطها مع بعضها على التعاقب (التتالي). ففي المثال المعطى لإجراء عملية الطرح، يتم بالخطوة الأولى الجمع مع المتمم الأحادي كالتالي $1000 + 1010 = 1110$ وبإضافة واحد للنتائج بعد حذف الفائض، يصبح الناتج النهائي: $1001 + 1 = 1000$. يظهر الشكل أعلاه آلية تنجيز العملية باستخدام الدارة المتكاملة 7483، حيث يمكن تعميم العملية على أي عددين آخرين مهما كان طول كل منهما. ويكفي بهذا المثال دائرة واحدة 7483 لتقوم بجمع العدد 14 مع المتمم لأثنين للعدد 5، على أن يتم أولاً إيجاد المتمم للواحد للعدد المطروح بمنطق خارجي وإضافة الواحد اللازم عن طريق المدخل C_i لإتمام عملية الطرح. أما بخصوص المنطق المناسب لإيجاد المتمم للواحد للمطروح هناك عدة إمكانيات تقنية نذكر منها اثنتان فقط هما: أولاً استخدام دائرة مثل 7404 من دون أية إشارة تحكم خارجية، على أن يتم التحكم بـ C_i للدائرة 7483 بشكل مستقل وذلك بإسناد $0 \leq C_i$ في حال الجمع أو 1 في حال الطرح. وثانياً اختيار منطق محدد مع إشارة التحكم لتأمين المتمم للواحد لعدد ما، مثل المنطق المبين في الشكل 19-9 الذي يظهر أن الخرج لمتحول الدخل B هو نفسه B إذا كان $C_i = 0$ وأنه المتمم للواحد B' إذا كان $C_i = 1$.



الشكل 19-8: منطق لتشكيل المتمم للواحد بوجود إشارة تحكم خارجية C_i .

بالطبع يوجد تقنيات أخرى كاستخدام دارات متعددة الوظائف، مثل الدارة التي يمكن أن تقوم بأربعة وظائف مختلفة، أي دائرة تظهر على الخرج بالإضافة إلى المتمم للواحد ثلاثة وظائف أخرى، هي: Buffer، Set، Reset. وتوصف مثل هذه الدارات بجدول وظائف يحدد وظيفة الدارة بناء على قيم إشارات التحكم الخاصة بالدائرة. وباستخدام الخيار الثاني، يكون

المخطط العام للدائرة التي تجمع أو تطرح العددين المعطيين مبين في الشكل 8-20.



الشكل 8-20: دائرة جمع / طرح لعددين بطول 4 bits (5 ± 14) باستخدام الدارة 7483

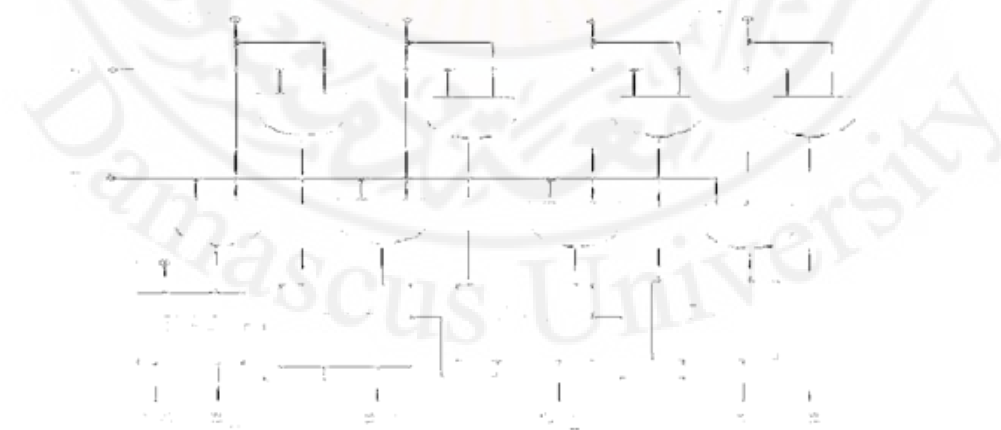
تم في المخطط إظهار توصيل كل من المداخل والمخارج بشكل فعلي في المثال المعطى، وقد تم الاقتصار على إظهار المنطق الخاص بتشكيل المتمم للواحد للبتين LSB و MSB، كما أن القيمة على مأخذ الإشارة C_i (الـ Pin رقم 13) تكون بالقيمة صفر، وتمثل النقل Carry في حال الجمع. أما C_o فتستخدم للربط مع C_i بالدائرة اللاحقة إن كانت الأعداد المراد جمعها أو طرحها أطول من 4 bits.

ثانياً: عمليات الضرب والقسمة الثنائية

يمكن إجراء عمليات الضرب والقسمة بطرائق مختلفة، منها ما يتم بوساطة العداد ومنها ما يتم بوساطة البرامج. يتم اللجوء إلى الطرائق التي تستخدم العداد عندما يطلب سرعة عالية بالحصول على النتيجة. ويمكن أن يصادف بالنسبة لعمليتي الضرب والقسمة، أدوات تركيبية وتتابعية، سنشير هنا للدارات التركيبية فقط. تتم عملية الضرب تقنياً بوساطة دارة، تسمى بالضارب Multiplier، وفيها تتحول عملية الضرب إلى مجموعة عمليات متتالية من الإزاحة والجمع. نعرض فيما يلي مثلاً عاماً عن جداء عديدين، الأول مكون من 4 bits، والثاني مكون من 2 bits، كما هو موضح في الشكل التالي:

$$\begin{array}{r} 1011 \\ \times 10 \\ \hline 10110 \\ 101100 \\ \hline 110010 \end{array}$$

تضم دارة الضارب بالإضافة إلى بوابات AND بعض دارات الجمع أيضاً. الشكل العام لدارة الضرب المقابلة للشكل المبين أعلاه تبدو كما في الشكل 8-21.



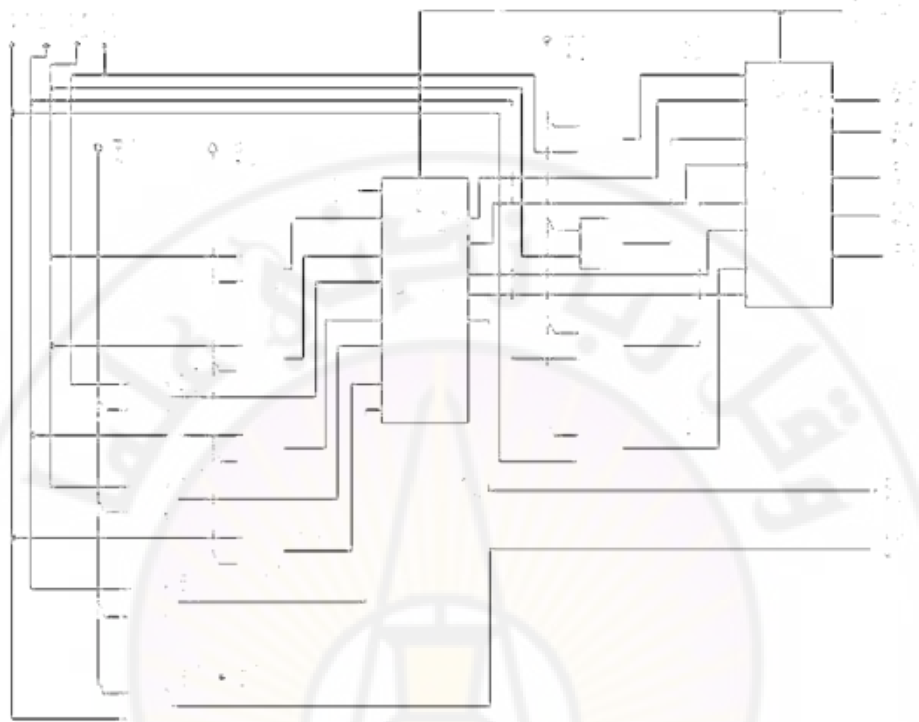
الشكل 8-21: الشكل العام للضارب الثنائي

يتضح من الشكل أعلاه صعوبة وكلفة تشكيل الضارب بشكل تقني مقابل السرعة المطلوبة، خاصة أن شكل الدارة غير قياسي، ويتغير الشكل كل مرة بحسب أطوال الأعداد المضروبة ببعضها البعض، علاوة على زيادة طول الناتج بكل ما يترتب على ذلك من دارات إضافية. وبالنسبة لعملية القسمة، فإن العملية تحول إلى شكل من أشكال الضرب. فمثلاً، لقسمة عدد ما مثل A على عدد آخر مثل B ، يتم انطلاقاً من الشكل العام للقسمة $Y = A/B$. بتعويض المقدار $1/B$ بمتحول آخر مثل X تتحول عملية القسمة إلى ضرب وفق الشكل $Y = A \cdot X$ والضرب بدوره يحول إلى جمع. أما لإيجاد قيمة المقدار $1/B$ فتوجد عدة طرق، منها ما هو معتمد على الدارات المنطقية التركيبية وجداول الحقيقة ومنها ما هو معتمد على طرق أخرى لإيجاد القيمة الأقرب من ناتج $X = 1/B$ بإحدى طرق التقريب Approximation مثل استخدام تتابعية نيوتن Iteration formula وذلك بعد اختيار دقيق وجيد لنقطة البداية X_0 . وهنا سنكتفي بالإشارة إلى عملية الضرب للأعداد الثنائية من خلال مثال محدّد يظهر التركيبية العامة للضارب الثاني.

مثال: ارسم الدارة المنطقية التي تنجز عملية ضرب العددين 5 و 14. الجواب: ليكن $A = 1110 = A_3A_2A_1A_0$ و $B = 101 = B_2B_1B_0$ ولنجري

$$\begin{array}{r} A = A_3A_2A_1A_0 \\ B = B_2B_1B_0 \\ \hline \begin{array}{cccc} & & & a_3 \ a_2 \ a_1 \ a_0 \\ & & & b_3 \ b_2 \ b_1 \ b_0 \\ \hline & & c_3 \ c_2 \ c_1 \ c_0 \\ S_6 \ S_5 \ S_4 \ S_3 \ S_2 \ S_1 \ S_0 \ (= a_0) \end{array} \end{array}$$

يلاحظ بعملية الضرب أن طول الناتج سيزداد عما هو الحال في عمليتي الطرح والجمع ويصبح الناتج بالنسبة لضرب عدد بطول 4 bits مع عدد آخر بطول 3 bits بطول 7 bits. يمكن انجاز الخوارزمية المبينة أعلاه عملياً باستخدام ما يلزم من دارات الضرب المنطقي AND (مثل الدارة 7408)، وما يلزم من دارات الجمع باستخدام الدارة 7483، فيكون الشكل العام للدارة المقابلة للمثال المعطى (أي دارة لضرب أي عددين من 0 حتى 15 بالنسبة للأول ومن 0 حتى 7 بالنسبة للثاني) مبين في الشكل 22-8.



الشكل 22-8: دائرة لضرب عددين ثنائيين بطول 4 bits للأول و 3 bits للثاني

7-8 الجامع / الطارح العشري Decimal adder/ subtractor

سبق وأشرنا إلى عمليات الجمع/ الطرح الثنائي ووجدنا باستخدام دائرة جامع/ طارح ثنائي بعرض 4 bits مثل الدارة 7483 كيفية اجراء هذه العمليات واظهار الناتج دون أي تعديل. أما التجهيزات الرقمية التي تنفذ العمليات الحسابية الأساسية بنظام الأعداد العشرية تعتمد رماز معين وتعطي النتائج بنفس الرماز. وبما أن الرماز المعتمد في مختلف الحواسيب الإلكترونية هو ال BCD، سنظهر فيما يلي باستخدام الدارات المتكاملة، كيفية تصميم الدارات المناسبة لإجراء عمليتي الجمع والطرح بالرماز BCD كلاً على حدة، تحت عنوان دائرة " الجامع/الطارح العشري المرمز ثنائياً "، حيث في حال التعامل مع الأعداد العشرية، قد يلزم تعديل النتيجة لاظهارها بالشكل الصحيح، ويمكن اجراء عمليات الجمع/ الطرح العشري بنفس الدارات المتكاملة التي استخدمت مع الأعداد الثنائية.

أولاً- الجمع باستخدام الرمز BCD

لجمع الأعداد العشرية في الدارات المنطقية، ترمز الأرقام العشرية ثنائياً بالرمز BCD وتتم عملية الجمع مع مراعاة تصحيح النتيجة إن كانت القيمة الأولية لحاصل الجمع بأي خانة تزيد عن الرقم 9. وفيما يلي نوضح هذه العملية من خلال مثال محدد كما يلي:

مثال: ارسم الدارة المناسبة لإيجاد ناتج جمع العددين العشريين: $278 + 495$ ؟

الجواب: بتمثيل العددين وفق الرمز BCD، يمكن جمعهما ثنائياً وفق التالي:

$$\begin{array}{r} 495 \sim 0100 \ 1001 \ 0101 \\ 278 \sim 0010 \ 0111 \ 1000 \\ \hline 773 \sim 0110 \ 10000 \ 1101 \end{array}$$

يلاحظ أن الناتج بكل من خانتي الآحاد والعشرات ليس بال BCD. يعدّل الناتج السابق إلى الشكل الأولي 0111 0000 1101، ويضاف الرقم 0110 لكل من مرتبتي الآحاد والعشرات للحصول على المكافئ بال BCD، فيصبح الناتج النهائي وفق التالي:

$$\begin{array}{r} 0111 \ 0000 \ 1101 \\ + \quad 0110 \ 0110 \\ \hline 0111 \ 0111 \ 0011 \\ 10^2 \quad 10^1 \quad 10^0 \end{array}$$

وهو ما يعادل الجواب الحقيقي. 3 7 7

الذي يمكن تنفيذ المرحلة الأولى منه بدارات متكاملة وفق ما هو مبين في الشكل 8-23.



الشكل 8-23: جزء من دارة جمع عددين عشريين ممثلين بالرمز BCD

حيث إن الشكل أعلاه يشير إلى آلية جمع الخانة الأولى فقط (خانة الآحاد من كل من العددين). يتم الجمع الأولي للرقمين بدارة 7483، وتصحيح النتيجة إلى الـ BCD بدارة ثانية 7483 في واحدة من ثلاث حالات:

- الأولى: إذا كان C_0 من الدارة 7483 الأولى بحالة الواحد، أي أن الناتج يعادل العدد 16 وما فوق.

- الثانية: إذا كان كل من البتّين الثالث والرابع (S_2 و S_3) بحالة الواحد أي من الشكل 11xx، حيث بغض النظر عن قيم x سيكون الناتج أكبر، أو يساوي 12، وهو ليس BCD وهذا ما تتفحصه بوابة AND1.

- الثالثة: إذا كانت قيمة كل من البتّين الثاني والرابع (S_1 و S_2) بحالة الواحد، أي من الشكل 1x1x، لأن الناتج سيكون أيضاً أكبر من 9 وبالتالي ليس بالرمز BCD. أي أن خرج البوابة OR سيكون بحالة الواحد إن حدثت أي من الحالات الثلاث الواردة أعلاه، وهذا يعني تشكيل هذا الواحد بصيغة الرقم 0110 وإضافته للناتج الأولي لتصحيحه إلى شكل BCD، وبنفس الوقت نقله لعملية الجمع بالخانة التالية بوظيفة نقل C_0 من مرحلة سابقة وربطه بـ C_1 التالي. إذا كان كل من العددين مكون من أكثر من خانة يلزم دارتين بتوصيلة مشابهة للتوصيلة بالشكل أعلاه لكل رقمين متقابلين من نفس الخانة واحدة للجمع والثانية لتصحيح النتيجة إلى صيغة الـ BCD. وبشكل عام يمكن تمثيل الدارة العامة للجامع العشري لأي عددين مهما كان طول كل منهما صندوقاً بالمخطط المبين في الشكل أدناه.

ثانياً - الطرح باستخدام الرمز BCD

تتم عملية الطرح باستخدام الرمز BCD بإيجاد المتمم للرقم 9 للمطروح وإضافته للمطروح منه، ومن ثم إضافة واحد للنتائج. ويوجد أكثر من طريقة لإيجاد المتمم التساعي للعدد العشري وتحقيق ذلك على شكل دارات رقمية، نستعرض فيما يلي طريقتين أساسيتين: الأولى: وتعتمد على إيجاد التوابع المنطقية التي تمثل المتمم التساعي بالرمز BCD للرقم انطلاقاً مما هو مبين بالجدول 8-8.

الجدول 8-8: المتمم التساعي للأعداد العشرية

True		9's complement				
Dec.	Input BCD w x y z	Dec.	Output BCD B ₃ B ₂ B ₁ B ₀			
0	0 0 0 0	9	1	0	0	1
1	0 0 0 1	8	1	0	0	0
2	0 0 1 0	7	0	1	1	1
3	0 0 1 1	6	0	1	1	0
4	0 1 0 0	5	0	1	0	1
5	0 1 0 1	4	0	1	0	0
6	0 1 1 0	3	0	0	1	1
7	0 1 1 1	2	0	0	1	0
8	1 0 0 0	1	0	0	0	1
9	1 0 0 1	0	0	0	0	0

يتضح من الجدول أعلاه، أن: $B_0(w,x,y,z) = \sum_m(0, 2, 4, 6, 8)$

$B_1(w,x,y,z) = \sum_m(2, 3, 6, 7)$

$B_2(w,x,y,z) = \sum_m(2, 3, 4, 5)$

$B_3(w,x,y,z) = \sum_m(0, 1)$

ولاختصار هذه التوابع، يمكن بتمثيل كل منها على حدة بمخطط كارنوف للحصول على الأشكال التالية: $B_0 = z'$, $B_1 = y$, $B_2 = x \oplus y$, $B_3 = x'y'w'$ حيث يمكن بعد ذلك رسم المخطط المنطقي المقابل لهذه التوابع الأربعة مجتمعة.

الثانية: تتمثل باستخدام جامع لإضافة العدد $(1010)_2 = (10)_{10}$ للمطروح مع إهمال المنقول بكل خانة. لتوضيح هذه الطريقة ننتقل من مثال محدد.

مثال: ارسم الدارة المناسبة لإيجاد ناتج طرح العددين العشريين: 278 - 547.

الجواب: نبيّن فيما يلي مراحل الحل وفق الخطوات التالية:

1- يمثل المطروح بال BCD وفق الشكل: $0010\ 0111\ 1000 \sim 278$

2- يتم إيجاد المتمم إلى واحد للشكل الثنائي للمطروح أي: $1101\ 1000\ 0111$

3- للحصول على المتمم التساعي للمطروح، تضاف القيمة 1010 لكل خانة من خانات

العدد ويكمل بكل خانة النقل الحاصل، وفق التالي:

$$\begin{array}{r} 1101\ 1000\ 0111 \\ + 1010\ 1010\ 1010 \\ \hline 1\ 0111\ 1\ 0010\ 1\ 0001 \end{array}$$

بإهمال المناقل يتم الحصول على الشكل: $0111\ 0010\ 0001$ أو ما يقابل العدد 721

وهو المتمم التساعي للمطروح. ويمكن إظهار المسار المبين أعلاه لرقم واحد باستخدام دائرة

جامع / طراح (الدارة المتكاملة 7483) وفق المخطط المبين في الشكل 8-24.

4- يجمع المطروح منه $0101\ 0100\ 0111 \sim 547$

مع المتمم التساعي للمطروح $+ 0111\ 0010\ 0001$

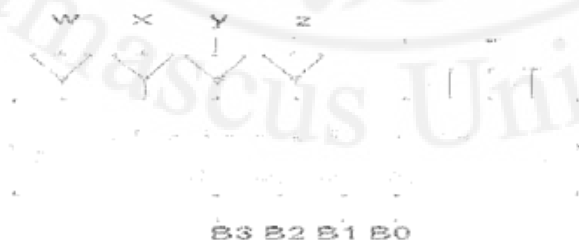
فيتم الحصول على الناتج الأولي: $1100\ 0110\ 1000$

5- يحول الناتج إلى BCD: $1100\ 0110\ 1000$

$+ 0110$

$1\ 0010\ 0110\ 1000$ وبإهمال المنقول الناتج

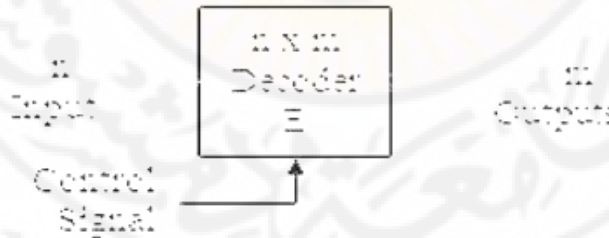
ينتج بشكل أولي العدد 268، وبإضافة واحد له يصبح الناتج 269 وهو الجواب الفعلي.



الشكل 8-24: آلية تشكيل المتمم التساعي لرقم عشري

8-8 مفك الترميز Decoder

تأتي أهمية دارات فك الترميز Decoders مثلاً، من ارتباط عمل هذه الدارات بعمل كثير من الدارات الأخرى، مثل المعالج الصغري، حيث إن المعالج الذي يشكل العنصر الرئيسي لأي حاسوب إلكتروني، يتعامل ضمن المنظومة الواحدة مع العديد من العناصر الرقمية من ذواكر على مختلف أنواعها، بوابات دخل وخرج وغيرها من دارات الدعم الأخرى. وبما أن المعالج الصغري بلحظة ما، لا يمكنه التعامل إلا مع دائرة واحدة مهما يكن عدد الدارات المتكاملة ضمن المنظومة الواحدة، يتحكم المعالج في الدارات المتممة له بالمنظومة بواسطة مفك الترميز، حيث يحدد المعالج عنوان الدارة التي سيتعامل معها وعن طريق مفك ترميز يتم انتقاؤها عن غيرها بإشارة تحكم خاصة، تدعى Chip Select إشارة الانتقاء CS. تسمى دائرة فك الترميز أيضاً بمحلل الشفرة وهي عبارة عن دائرة منطقية تركيبية تحوي بشكل عام على n مدخل و m مخرج، بحيث إن $m = 2^n$. بمعنى أنه من أجل n متحول على الدخل، يقوم مفك الترميز بتوليد 2^n تركيبة مختلفة على الخرج، بحيث إن خرج واحد من مجمل المخارج يكون فعالاً. يمثل كل خرج لمفك الترميز حداً أصغرياً minterm مشكلاً من متحولات الدخل. ويظهر الشكل 8-25 الرمز المنطقي لمفك الترميز الذي يشار إليه بالتركيبة nxm.



الشكل 8-25: الرمز المنطقي لمفك الترميز

لرسم المخطط المنطقي لمفك الترميز، يمكن الانطلاق من جدول الحقيقة الذي يختلف من مفك ترميز إلى آخر بحسب تركيبته. فمثلاً، بالنسبة لمفك ترميز 2x4 Decoder (أي مفك ترميز عدد مداخله 2 وعدد مخارجه 4) يكون جدول الحقيقة كالتالي:

الجدول 8-9: جدول الحقيقة لمفكك ترميز 2x4

Inputs		Outputs
a	b	
0	0	$0 \sim a'b'$
0	1	$1 \sim a'b$
1	0	$2 \sim ab'$
1	1	$3 \sim ab$

المخطط المنطقي الذي يحقق جدول الحقيقة أعلاه، مبين في الشكل 8-26، حيث يمكن تشكيل المخطط باستخدام بوابات AND ويكون الخرج فعالاً بالواحد.



الشكل 8-26: مفكك ترميز 2 إلى 4 خطوط (باستخدام بوابات AND)

يمكن تشكيل مفكك الترميز نفسه باستخدام بوابات NAND كما هو مبين بالشكل 8-27، ليصبح الخرج (المنتقى Selected) فعالاً بالصفر.

الشكل 8-27: 2x4 Decoder مع مدخل تأهيل (باستخدام بوابات NAND)

تم بالشكل أعلاه اضافة إشارة تأهيل فعالة بالصفير للمخطط المنطقي لـ 2x4 Decoder وهنا نشير إلى أنه لبعض الدارات المنطقية المصنعة على شكل دائرة متكاملة وخاصة الدارات من رتبة MSI مثل مفككات الترميز، توجد إشارة تحكم إضافية لتأهيل الدارة، تسمى سماحية Enable أو تأهيل (اختصاراً E). تلعب إشارة التأهيل (وقد يكون عددها بالدائرة أكثر من إشارة) دور المفتاح الرئيسي لعمل الدارة. وتكون إشارة التأهيل فعالة بالصفير أو بالواحد (وهذا عائد لمصنع الدارة، ولا يؤثر في عملها أياً كانت قيمتها). ويصبح جدول الحقيقة بوجود إشارة التأهيل، وفق ما هو مبين في الجدول 8-10.

الجدول 8-10: جدول الوظائف (العمل) لمفكك ترميز 2x4

Enable	Inputs		Outputs			
E	a	b	CS' ₀	CS' ₁	CS' ₂	CS' ₃
1	d	d	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

يتضح من الجدول بأنه إذا كانت إشارة التأهيل $E = 1$ ، فإن الدارة تكون غير مؤهلة Disable بغض النظر عن القيم الموجودة على المدخلين a و b وذلك لأن خرج العاكس العائد لإشارة التأهيل E سيصبح 0، وبالتالي مخارج بوابات NAND ستصبح بالقيمة 1، أي لا يتم انتقاء أي خرج. أما عندما تكون $E = 0$ سيصبح خرج العاكس واحداً، وبالتالي خرج كل بوابة NAND سيصبح مرتبطاً بالقيم المطبقة على كل من المدخلين a و b، وبما أن خرج مفكك الترميز بالشكل أعلاه فعالاً بالصفير فإن خرجاً واحداً سيكون بالصفير وبقية المخارج ستكون قيمة كل منها بالقيمة 1، بمعنى أن خرجاً واحداً من مخارج مفكك الترميز سيكون منطقي وتفعيل الدارة المرتبطة به فقط.

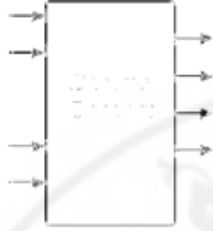
8-9 المرمز Encoder

يمثل المرمز دائرة رقمية تعمل بعكس عمل دائرة مفكك الترميز، بالتالي إذا كان المرمز من النوع القياسي Standard، فهو يمثل دائرة تركيبية بـ 2^n مدخل و n مخرج، بحيث يكون بكل لحظة أحد المداخل فعالاً وعلى المخرج يوجد الرمز الثنائي Binary code المقابل له، أي أن مرمز بثمانية مداخل يكفيه ثلاثة مخرج لترميز أي مدخل يتم انتقاؤه. بالإضافة إلى المرمز القياسي، يمكن أن نصادف أيضاً رموز غير قياسية، بمعنى أن عدد مداخلها ليس من رتبة 2^n ، مثل المرمز العشري الذي يمثل دائرة من عشرة مداخل تقابل الأرقام من 0 - 9 وأربعة مخرج A, B, C, D يقرأ عليها الرمز المقابل للمدخل الفعال بالأوزان 8 4 2 1. يظهر الجدول 8-11 جدول الحقيقة للمرمز العشري.

الجدول 8-11: جدول الحقيقة للمرمز العشري

الرقم Index	Inputs المداخل										Outputs المخرجات			
	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	D	C	B	A
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	0	0	0	1	1
.....														
8	0	0	0	0	0	0	0	0	1	0	1	0	0	0
9	0	0	0	0	0	0	0	0	0	1	1	0	0	1

يظهر الشكل 8-28 الرمز المنطقي والمخطط المنطقي للمرمز العشري، حيث يمكن رسم المخطط المنطقي باستخدام بوابات OR، لأنه كما يبدو من الجدول 8-11، أنه من أجل المداخل العشرة، يتولد أربعة مخرج كل منها يمثل تابعاً منطقياً، بحيث التابع A مثلاً، معرّف من أجل الأرقام الفردية 1,3,5,7,9 لوجود قيمة الواحد لهذا التابع مقابل كل من هذه الأرقام $A = I_1 + I_3 + I_5 + I_7 + I_9$. ونفس الأسلوب يمكن كتابة بقية التوابيع، إذ أن D مثلاً معرّف من أجل الرقمين 8 و 9 فقط وبالتالي: $D = I_8 + I_9$.



الشكل 28-8: الرمز المنطقي والمخطط المنطقي للرمز العشري

يلاحظ من جدول الحقيقة للرمز أيضاً نقطتان أساسيتان:

الأولى: عندما تكون جميع مداخل المرمز أصفاراً، يكون الخرج نفس الخرج الذي يتم الحصول عليه عندما يكون المدخل $I_0 = 1$ ، أي يوجد غموض Ambiguity، علاوة على أن المدخل I_0 لا يظهر في المخطط المنطقي. يمكن حل مثل هذا الغموض بإضافة مخرج يشير إلى وجود دخل واحد على الأقل يساوي الواحد.

الثانية: وجود بعض القيود على جدول الحقيقة تتبع من طبيعة عمل المرمز نفسه، وتتمثل في أنه بأي لحظة يمكن تفعيل أي من مداخله. فإذا كان لسبب ما قد تم تفعيل مدخلين معاً بأن واحد، فإن الخرج ينتج تركيبة غير معروفة. مثل الحالة التي يكون فيها كل من المدخلين I_6 و I_9 مثلاً بحالة 1، عندها سيكون خرج المرمز التركيبية 1111، نظراً لأن المخارج B و C ستكون بالقيمة 1 من تفعيل I_6 والمخارج A و D بالقيمة 1 من تفعيل I_9 ولكن التركيبية 1111 لا تقابل تفعيل المدخل I_6 بمفرده ولا المدخل I_9 بمفرده، بل حتى لا يوجد مدخل يقابل مثل هذه التركيبية، أي حدث غموض. للتخلص من مثل هذا الغموض، يجب تحديد سلم أفضليات للدخل. فإذا تم اعتماد الأفضلية العليا للمداخل ذات الأدلة الأكبر، يمكن بالحالة المبينة أعلاه عن وجود 1 على كل من المدخلين I_6 و I_9 بأن هذا الواحد يقابل التركيبية 1001 المقابلة لـ I_9 على الخرج، لأن أفضلية المدخل I_9 هي أعلى من أفضلية I_6 .

8-10 تطبيقات على استخدام الرموز ومفككات الترميز

يمكن استخدام مفكك الترميز بالإضافة إلى وظيفته الأساسية بعدة تطبيقات، نذكر منها: أولاً: خلال تصميم ما، قد يلزم 3x8 Decoder مثلاً وعند التنفيذ قد لا تصادف مثل هذا المفكك للترميز، بل قد يتوافر على سبيل المثال 4x16 Decoder. يمكن بهذه الحالة حل المشكلة باستخدام دائرة 4x16 وباستخدام ثلاثة من مداخلها فقط، ووصل المدخل الرابع بالثالث، بحيث يصبح معرفاً فقط ثمانية مخارج، أي من السهولة تشكيل مفكك ترميز صغير من مفكك ترميز كبير.

ثانياً: قد يلزم مثلاً، تشكيل 4x16 Decoder من مفككات ترميز متوافرة بالتركيبة 2x4 ويلزم وفق هذه الحالة بالحد الأدنى أربعة مفككات ترميز لتأمين العدد المطلوب من المخارج. بفرض أن خرج مفككات الترميز المتوافرة فعالة بالصفر، يمكن بعد ذلك باستخدام إحدى إشارات التأهيل الحصول على المطلوب، وفق ما هو مبين في الشكل 8-29.



الشكل 8-29: تشكيل دائرة 4x16 Decoder من دوائر 2x4 Decoder

يظهر من الشكل ضرورة استخدام مفكك ترميز خامس لتحديد أي مفكك ترميز يتم انتقاؤه بلحظة ما عن غيره من المفككات الأربعة، لأنه لا يمكن بلحظة ما أن يكون مفعلاً أكثر من

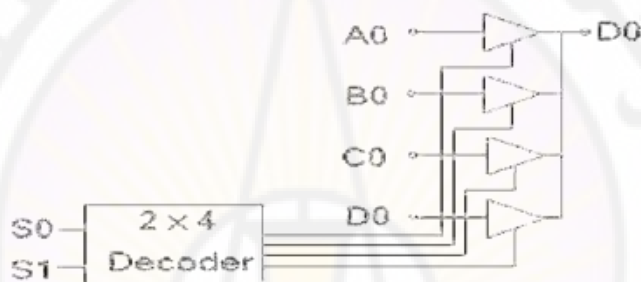
مفكك ترميز واحد. وهكذا تصبح الدارة الناتجة تمثل مفكك ترميز 4×16 لأنها مكونة من أربعة مداخل a, b, c, d وستة عشر مخرجاً من 0 حتى 15. فمثلاً، إذا كان على مداخل مفكك الترميز الخامس (الاضافي) التركيبية $dc = 00$ ، سيكون الخرج 0 هو الفعّال، وهذا يعني أنه تم انتقاء مفكك الترميز رقم 1 بمخارجه الأربعة من 0 حتى 3. وبالتالي إذا كان $ba = 10$ سيتم تفعيل الخرج رقم 2 من مفكك الترميز الأول. وهكذا يتم من أجل كل تركيبة للمدخلين c و d انتقاء أحد مفككات الترميز الأربعة وكل منها بأربعة مخارج. ولتفعيل خرج محدد مثل الخرج رقم 13 مثلاً، يجب تأمين التركيبية 1101 على المداخل $dcba$ والتي تقابل التركيبية $dc = 11$ والتركيبية $ba = 01$.

ثالثاً: يستخدم مفكك الترميز أيضاً برسم التوابع المنطقية. فمثلاً، باستخدام مفكك الترميز وبوابة OR واحدة يمكن رسم الدارة التي تحقق أي تابع منطقي مثل تابع معطى بالشكل $y(a, b, c) = b' + a'c + ab'c'$ وفق التالي: يتم الانطلاق من شكل التابع، حيث يتضح من الصياغة الجبرية أن التابع يضم ثلاثة متحولات، وجدول الحقيقة يبدو كما هو مبين في الجدول بالشكل 8-30 أي $y(a, b, c) = \sum_m(0, 1, 3, 4, 5)$. وبما أن كل خرج لمفكك الترميز يمثل حداض أصغرياً، يلزم لتمثيل الحدود المعرفة بالتابع استخدام 3×8 Decoder. ويكون المخطط المنطقي التابع المعطى باستخدام مفكك ترميز وبوابة OR خارجية، وفق ما هو مبين في الشكل أدناه.

a	b	c	y
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	0

الشكل 8-30: دائرة لتنفيذ تابع منطقي باستخدام مفكك ترميز وبوابة OR

رابعاً: يستخدم مفكك الترميز للحصول على مسرى ذي ثلاث حالات Three-state bus بوصل عدة مخارج من النوع ذي الثلاث حالات مع بعضها، بحيث في أي لحظة يكون أحد مداخل التحكم مؤهلاً في حين أن جميع المخارج الباقية تكون بحالة ممانعة عالية. فمثلاً، من أجل أربع سجلات A, B, C, D كل منها بطول 4bits يمكن تصميم مسرى مشترك باستخدام بوابات ذي ثلاث حالات ومفكك ترميز، وفق ما هو مبين بالشكل 8-31، بحسب القيم الموجودة على خطوط الانتقاء S_0 و S_1 ، تؤهل خرج واحد من المخارج الأربعة التي كل منها يمثل مفتاحاً لدارئ بثلاث حالات.



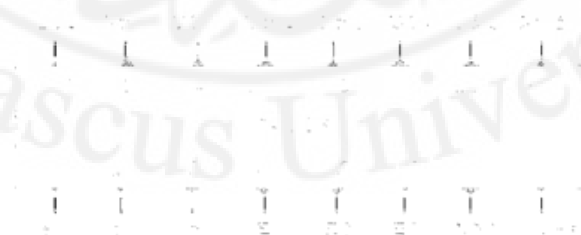
الشكل 8-31: تشكيل مسرى (ممر Bus) ذي ثلاث حالات باستخدام مفكك ترميز

يوصل البت ذي الوزن الأدنى من كل سجل إلى خط المعطيات D_0 ذو الوزن الأدنى من مسرى المعطيات (ويكرر ثلاث مرات أخرى لتشكيل البتات D_1 و D_2 و D_3 من البتات المقابلة من السجلات الأربعة). الميزة المهمة للبوابات ذات الحالات الثلاث هي أن تأخير تأهيل الخرج أطول من تأخير عدم تأهيله. وبالتالي إذا كان يوجد أكثر من بوابة مربوطة مع بعضها على مسرى مشترك وتم تأهيل إحداها فإن البوابات غير المؤهلة تدخل إلى حالة الممانعة العالية قبل أن تتأهل البوابة المنتقاة وهذا يلغي أثر الحالة التي يمكن أن تكون فيها أكثر من بوابة فعالة بأن واحد. يوجد بعض مفككات الترميز بغير الشكل القياسي، مثل الـ BCD decoder الذي يمثل دائرة من أربعة مداخل (تكفي لترميز 16 مخرج) يستخدم منها فقط عشرة مخارج تتوافق مع الرماز BCD. يوجد في الجدول 8-12 أرقام بعض الدارات المتكاملة التي تمثل مفككات ترميز مناسبة لمختلف التطبيقات، مثل الدارات المتكاملة: 7442, 7445, 74138, 74139, 74154 وغيرها.

الجدول 8-12: بعض الدارات المتكاملة من العائلة TTL التي تمثل مكونات رقمية أساسية

رقم الدارة	وظيفة (تركيبية) الدارة	ملاحظات
74 42	BCD-Decimal Decoder	DIL – 16
74 45	BCD-Decimal Decoder with OC	DIL – 16
74 46	BCD-Seven Seg. Decoder with OC	DIL – 16
74 82	2 bits binary full adder	DIL – 14
74 83	4 bits binary full adder	DIL – 14
74 85	4 bits Comparator	DIL – 16
74138	3-Line to 8-Line Decoder/Demultip.	DIL – 16
74139	Dual 2-Line to 4-Line Dec./Demultiplexer.	DIL – 16
74147	10 to 4-line Priority Encoder	DIL – 16
74148	8 to 3-line Priority Encoder	DIL – 16
74151	8-bits Data Selector	DIL – 24
74153	Dual 4 to 1 Line Data Selector	DIL – 16
74154	4 to 16-Line Decoder (& 74159)	DIL – 24
74155	Dual 2 To 4 Decoder	DIL – 16
74157	Quad 2 to 1 Line Data Selector	DIL – 16
74184	BCD-to-Binary Converter with OC	DIL – 16
74185	Binary-to-BCD Converter with OC	DIL – 16

وبالنظر لورود الدارة المتكاملة 74138 التي تمثل 3x8 Decoder في معظم التصاميم المنطقية، نبين في الشكل 8-32 توزيع الإشارات على مآخذ الدارة، خاصة أنها تُعد أيضاً مثالاً عن الدارات التي تضم أكثر من إشارة سماحية. فمثلاً، من أجل التركيبية 101 على الدخل يصبح الخرج CS'_5 بالقيمة صفر، عندما $E_3E_2E_1 = 100$.



الشكل 8-32: توزيع إشارات الدارة المتكاملة 74138

من التطبيقات الممكنة باستخدام المرمز، نشير فيما يلي إلى تطبيق واحد يمثل مرمز يتضمن تابع الأفضلية، ويسمى " مرمز الأفضلية Priority encoder".
مثال: صمم دائرة مرمز ذا أفضلية بأربعة مداخل.

بداية، يتم الانطلاق من جدول الحقيقة الذي يوصف عمل الدارة المطلوبة. بما أن المرمز المطلوب يحتوي على أربعة مداخل، لتكن I_0, I_1, I_2, I_3 ، يكفي مخرجان لترميز هذه المداخل، لتكن O_0, O_1 . ويقصد تجاوز حالة الغموض بالنسبة للمرمز، يضاف بجدول الحقيقة مخرجاً ثالثاً F للإشارة إلى صلاحية الدخل Valid bit indicator، بحيث تكون قيمة هذا الخرج "1" عندما تكون قيمة مدخل واحد أو أكثر من مداخل المرمز تساوي "1". وبالعكس ذلك، عندما قيمة F تساوي "0" فلا معنى للمخارج، لأن الدخل يكون غير مقبول. إذاً يجب بالخطوة الأولى للتصميم، تحديد الأفضليات Input priority. فإذا اعتمد مبدأ الأفضلية العليا للمداخل ذات الأدلة الأكبر يكون I_3 هو صاحب الأفضلية العليا، ويمكن باستخدام حالات عدم الاكتراث d ، اختصار جدول الحقيقة العائد للدارة المطلوبة بأربعة مداخل والمكون من 16 سطراً إلى الأسطر المبينة في الجدول 8-13. يلاحظ أن تركيبة الدخل $I_3I_2I_1I_0 = 1 d d d$ التي تقابل القيمة 1 للخروج O_3 ، تعني التراكيب التالية: 10 00, 10 01, 10 10, 10 11, 11 00, 11 01, 11 10, 11 11.

الجدول 8-13: جدول الحقيقة لرمز ذا أفضلية بأربعة مداخل

Inputs المداخل				Outputs المخارج		
I_3	I_2	I_1	I_0	F	O_1	O_0
0	0	0	0	0	d	d
0	0	0	1	1	0	0
0	0	1	d	1	0	1
0	1	d	d	1	1	0
1	d	d	d	1	1	1

وبتمثيل توابع الخرج بمخططات كارنوف، وفق ما هو مبين أدناه.

		المخرج O_0						المخرج O_1			
$I_2 I_1$	I_0	00	01	11	10	$I_3 I_2$	$I_1 I_0$	00	01	11	10
		00	01	11	10			00	01	11	10
00				1	1	00					
01						01		1	1	1	1
11		1	1	1	1	11		1	1	1	1
10		1	1	1	1	10		1	1	1	1

خريطة 1

خريطة 1

خريطة 1

خريطة 2

وباعتبار أن التابع F معرف من أجل كل قيمة واحد على أي من المداخل الأربعة يمكن الحصول على المعادلات التالية:

$$F = I_0 + I_1 + I_2 + I_3$$

$$O_0 = I_3 + I_2 I_1;$$

$$O_1 = I_3 + I_2$$

وهي تقابل المخطط المنطقي المبين في الشكل 8-33.



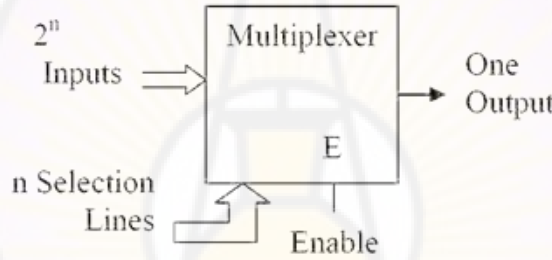
الشكل 8-33 : دائرة مرمرز أفصلية بأربعة مداخل

11-8 المضمّمات والمفرقات Multiplexers / Demultiplexer

تُعدّ المضمّمات والمفرقات من الدارات المنطقية التركيبية واسعة الانتشار أيضاً، وهي تقوم بتوجيه المداخل إلى المخرجات اعتماداً على عنوان محدد يتم اختياره وفق إشارات تحكم Control signal (تسمى خطوط الانتخاب Selection lines)، ويمكن لهذه الدارات أن تؤدي العديد من الوظائف، مثل اختيار البيانات المدخلة وتجزئة الفترات الزمنية ورسم التتابع المنطقية. بالاطار العام كما هو الحال بالنسبة للمرمرز ومفكك الترميز، يُعدّ عمل المضمّم Multiplexer معاكساً تماماً لعمل المفرّق Demultiplexer.

أولاً: المضمّم Multiplexer

يمثل المضمّم دائرة تركيبية تنتقي المعلومات الثنائية بلحظة ما من مدخل واحد من بين عدة مداخل وتوجهها إلى المخرج الوحيد. يتم التحكم بعملية اختيار خط دخل معين عن غيره من خطوط الدخل الأخرى بواسطة عنوان خاص بالدخل الذي تم انتقاؤه. وتطبق العناوين العائدة لكل دخل على حدة على مجموعة من خطوط الانتقاء، بحيث من أجل n خط انتقاء يمكن أن يصل عدد خطوط الدخل للمضمّم إلى 2^n . يسمى المضمّم بالناخب Data selector أيضاً، لأن دارته تقوم بانتخاب خط واحد من خطوط الدخل وإظهاره على المخرج، كما يسمى بالمجمّع لأن الدارة تقوم بتجميع خطوط الدخل بخط وحيد على المخرج. يظهر الشكل 8-34 الرمز المنطقي للمضمّم.



الشكل 8-34: الرمز المنطقي للمضمّم

يشار إلى المضمّم في المخططات بالرمز MUX، كأن يقال مضمّم 4×1 MUX، وهذا يعني ان للمضمّم أربعة مداخل ومخرجاً واحداً. يمكن بدارة المضمّم تصور أن المعطيات موجودة على المداخل، وبعد تفعيل إشارة التأهيل E، يظهر على المخرج المعطيات الموجودة على أحد المداخل الذي يميز عن غيره بالعنوان الموجود على خطوط الانتخاب. بالطبع إذا كانت إشارة التأهيل للمضمّم غير فعالة، لا يظهر على المخرج قيمة أي من المداخل. يوصف عمل المضمّم بجدول وظائف Function table. فمثلاً، من أجل 4×1 MUX يمكن تحديد شكل المضمّم وعدد خطوط الانتقاء Address Lines وبالتالي جدول الوظائف العائد له وفق ما هو مبين بالجدول 8-14.

الجدول 8-14: جدول الحقيقة (الوظائف) لمضمم بالتركيبة 4x1 مع رمزه

الرمز	الوظيفة	الرمز	الوظيفة
D_0	الدخول	D_1	الدخول
D_2	الدخول	D_3	الدخول
D_4	الدخول	D_5	الدخول
D_6	الدخول	D_7	الدخول
D_8	الدخول	D_9	الدخول
D_{10}	الدخول	D_{11}	الدخول
D_{12}	الدخول	D_{13}	الدخول
D_{14}	الدخول	D_{15}	الدخول
D_{16}	الدخول	D_{17}	الدخول
D_{18}	الدخول	D_{19}	الدخول
D_{20}	الدخول	D_{21}	الدخول
D_{22}	الدخول	D_{23}	الدخول
D_{24}	الدخول	D_{25}	الدخول
D_{26}	الدخول	D_{27}	الدخول
D_{28}	الدخول	D_{29}	الدخول
D_{30}	الدخول	D_{31}	الدخول

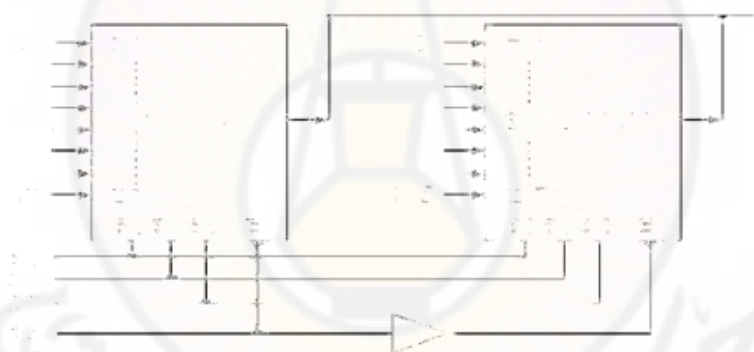
حيث إن قيمة 00 على خطوط الانتقاء تقابل مرور الدخل D_0 على الخرج، وهكذا يكون الأمر بالنسبة لبقية التراكيب. يمكن رسم المخطط المنطقي للمضمم المقابل لجدول الوظائف أعلاه، وفق ما هو مبين في الشكل 8-35.



الشكل 8-35: المخطط المنطقي لمضمم 4 x 1

يتضح من الشكل أعلاه أنه من أجل التركيبة $S_1S_0 = 00$ مثلاً، سيكون خرج البوابات AND ذوات الأرقام 2, 3, 4 صفراً، والبوابة رقم 1 تكون مفتوحة، بمعنى مداخلها الثاني والثالث في حالة واحد وخرجها يعتمد فقط على القيمة الموجودة على D_0 ، بحيث إذا كان على D_0 واحد سيظهر هذا الواحد على خرج البوابة OR، ويكون الخرج y بحالة الواحد. وهكذا الحال بالنسبة للتراكيب الأخرى. إذ أن كل تركيبة S_1S_0 تقابل ظهور أحد المداخل على الخرج النهائي. وكما مر معنا في الفقرة السابقة حول ربط عدة مفكات ترميز مع بعضها لزيادة عدد المخارج، يمكن أيضاً باستخدام إشارات التأهيل ربط عدد من المضمّمات

مع بعضها لزيادة عدد المداخل. ويشترط في حال ربط المضمّمات مع بعضها أن تكون المضمّمات المستخدمة بخرج ثلاثي الحالة Three state، لأن المضمّمات التي ستربط مع بعضها يجب أن يكون لها خرج نهائي وحيد مهما بلغ عددها، بالتالي لابد من قصر المخارج مع بعضها لتشكيل خرج واحد (ويمكن استخدام بوابة OR لربط المخارج). مثال: صمم دائرة 16×1 MUX باستخدام ما يلزم من مضمّمات 8×1 MUX. الجواب: يلزم دارتان 8×1 MUX، وبهذه الحالة يجب استخدام إشارات التأهيل E لربط المضمّمات 8×1 MUX بقصد تشكيل مضمّم بالتركيبة 16×1 MUX. بفرض أن إشارة التأهيل للمضمّم المتوافر فعالة بالقيمة 0، عندئذ يكون المخطط العام لربط المضمّمين مع بعضهما، وفق ما هو مبين في الشكل 8-36.



الشكل 8-36: تشكيل مضمّم 16×1 MUX من مضمّمين 8×1 MUX

تم وصل المضمّمات بشكل قياسي، أي أن لكل مضمّم بالتركيبة 8×1 MUX ثلاثة خطوط انتقاء S_0, S_1, S_2 توصل مع بعضها بشكل قياسي وثمانية مداخل من D_0 حتى D_7 ومخرج واحد. وتوصل المخارج مع بعضها لتشكيل مخرج واحد إن كان المضمّم مشكلاً ببوابات ثلاثية الحالة. وباستخدام خط التأهيل في كل من المضمّمين وبوابة NOT يمكن تشكيل مضمّم ستة عشر مدخلاً من I_0 حتى I_{15} وبأربعة خطوط انتقاء، بحيث خط الانتقاء الرابع S_3 يؤهل المضمّم MUX_0 عندما يكون بالصفّر ويكون MUX_1 غير مؤهل لوجود قيمة 1 على المأخذ E. وعندما يصبح بالقيمة واحد يعمل بالعكس، أي يؤهل MUX_1 ويتجاهل

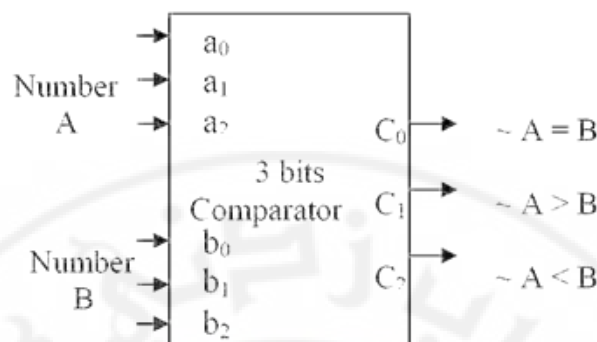
MUX_0 . وبما أن المضمّم يولد على الخرج حدوداً أصغريه أيضاً، ويحتوي ضمناً على بوابة OR، هذا يعني أنه يمكن استخدامه لتنفيذ التوابع المنطقية من دون بوابة OR، بالتالي التركيبية العامة لتنفيذ التابع باستخدام المضمّم ستكون أبسط.

ثانياً: المفترقات Demultiplexer

يصادف في بعض التطبيقات دارات تركيبية بوظيفة تعاكس وظيفة المضمّم تسمى بالمفترقات. إذا المفترق هو دائرة تركيبية تستقبل المعطيات من مدخل وحيد وتوجهها إلى مخرج واحد من بين 2^n خرج إذا كان عدد خطوط الانتقاء n خطاً. والتركيبية العامة لدائرة المفترق هي بعكس تركيبية دائرة المضمّم. وبالتالي المفترق يحاكي الدارة التي تقوم بتحويل المعطيات التسلسلية إلى تفرعية. ويوجد بدارة المفترق دوماً مخرج واحد منتقى (فعال) وبقية المخارج تكون غير فعالة، أي حالة خاصة لمفكك الترميز، حتى أن مفكك الترميز المجزئ بمدخل للتأهيل يمكن أن يعمل مفترقاً. ولما كان بالإمكان الحصول على عمل كل من مفكك الترميز والمفترق بنفس الدارة، يمكن الإشارة إلى مفكك الترميز المجزئ بمدخل للتأهيل على أنه مفكك ترميز / مفترق.

8-12 المقارن Comparator

يمثل المقارن دائرة منطقية من النوع التركيبي، تسمح بمقارنة عددين لإظهار فيما إذا كان العددين متساويين أو غير متساويين وأي منهما أكبر إن كانا مختلفين. ولإظهار التركيبية العامة للمقارن يمكن الانطلاق من مثال فيه كل من المدخلين A و B مكون من 3 bits ($A = a_2a_1a_0$ و $B = b_2b_1b_0$)، فيكون الرمز المنطقي للمقارن، وفق ما هو مبين في الشكل 8-37، حيث إن العنصر الرئيسي في تركيبية المقارن هو بوابة التكافؤ NOT XOR لأن خرج البوابة XNOR بالمدخلين A و B يكون بحالة الواحد عندما تكون $A = B$. يكون العددين متساويين إذا كانت جميع البتات المتقابلة فيما بينهما متساوية. ولتحديد أي من العددين هو الأكبر، يتم الانطلاق بعملية المقارنة من أعلى بتين متقابلتين. يمكن بالانطلاق من جدول الوظائف للمقارن تصميم دائرة مقارن تضم كافة التراكيب الممكنة على الدخل، وفق ما هو مبين في الجدول 8-15.



الشكل 8-37: الرمز المنطقي لمقارن بين عددين بطول 3 bits لكل منهما.

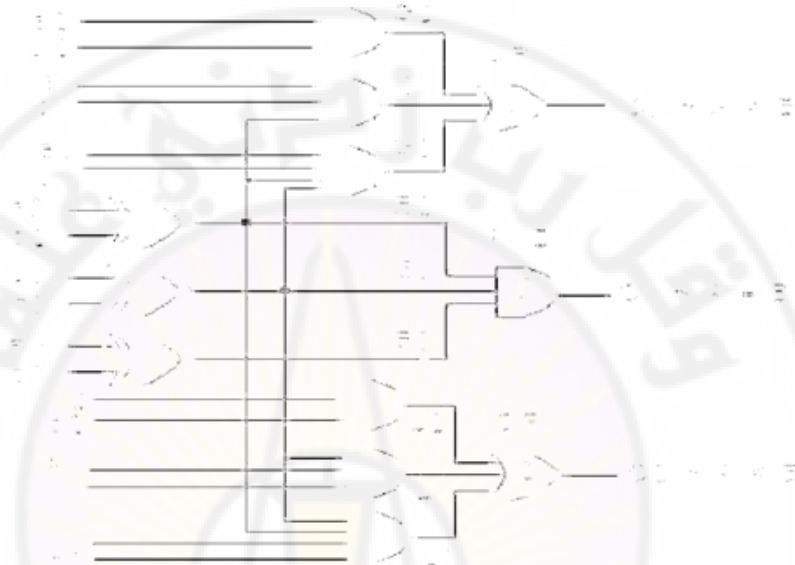
تحدّد نتيجة المقارنة بثلاثة متحولات ثنائية: يرمز الأول C_0 لحالة $A = B$ والثاني C_1 لحالة $A > B$ والثالث C_2 لحالة $A < B$.

الجدول 8-15: جدول بالتراكيب الموافقة لعمل المقارن

Comparing Inputs			Outputs		
$A_2 B_2$	$A_1 B_1$	$A_0 B_0$	$A > B$	$A < B$	$A = B$
$A_2 > B_2$	d	d	H	L	L
$A_2 < B_2$	d	d	L	H	L
$A_2 = B_2$	$A_1 > B_1$	d	H	L	L
$A_2 = B_2$	$A_1 < B_1$	d	L	H	L
$A_2 = B_2$	$A_1 = B_1$	$A_0 > B_0$	H	L	L
$A_2 = B_2$	$A_1 = B_1$	$A_0 < B_0$	L	H	L
$A_2 = B_2$	$A_1 = B_1$	$A_0 = B_0$	L	L	H

يمكن رسم المخطط المنطقي الذي يقابل الجدول أدناه حول عمل المقارن، وفق ما هو مبين في الشكل 8-38، حيث يجب أن يكون $G_2 = 1$ في الحالة التي يكون فيها الخرج $C_1 \sim (A > B) = 1$ ، أي يجب أن يكون $a_2 = b'_2 = 1$ وهذا يعني أن a_2 يجب أن يكون 1 و b_2 يجب أن يكون 0، أي $A > B$ ، وهذا يقابل الشكل: $a_2 a_1 a_0 \quad b_2 b_1 b_0$
 $1 \quad x \quad x \quad 0 \quad x \quad x$
وبأسلوب مماثل، الحالة التي يكون فيها الخرج $C_2 = 1$ يعني أن $L_2 = 1$ وهذا لا يتم إلا عندما يكون $a'_2 = b_2 = 1$ ، أي أن $a_2 = 0$ و $b_2 = 1$ ، $(A < B)$.

بعد الانتهاء من مقارنة أعلى بتّين، تستمر عملية المقارنة بفرض أن $a_2 = b_2$ ($E_2=1$) وأن الخرج C_0 لا يزال صفراً، أي أن الاختلاف هو في موضع آخر أدنى.



الشكل 8-38: المخطط المنطقي لمقارن بين عددين بطول 3 bits لكل منهما

تصنع المقارنات على شكل دارات متكاملة، وكمثال عنها نورد من الجدول 8-12 الدارة 7485 التي تمثل 4bit Comparator، حيث باستخدام أكثر من دارة، يمكن وفق توصيل محدّد أن تتم المقارنة بين أي عددين مهما يكن طول كل منهما.

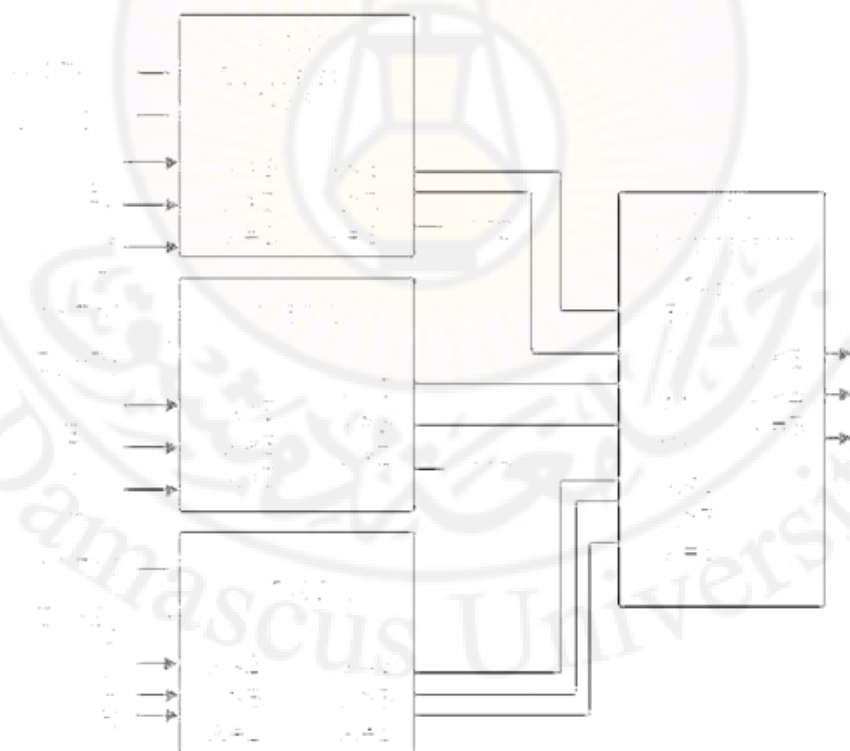
مثال: ارسم باستخدام الدارة 7485 المخطط العام للمقارنة بين عددين كل منهما 12 bits لتحديد إن كانا متساويين أو لتحديد الأكبر منهما إن كانا مختلفين.

الجواب: بالانطلاق من جدول الوظائف المبين بالجدول 8-16، يلاحظ أن الدارة 7485 تضم بالإضافة إلى المداخل الثمانية لإدخال المعطيات (بمعدل أربعة مداخل للعدد الأول وأربعة للثاني) والمخارج الثلاثة الرئيسية التي تشير إلى الحالات: $A > B$ و $A = B$ و $A < B$ وجود ثلاثة مداخل أخرى للربط مماثلة بالتسمية، تدعى Cascading inputs. يكون الشكل النهائي لدارة مقارنة العددين المعطيين، وفق ما هو مبين في الشكل 8-39.

الجدول 8-16: جدول الوظائف لمقارنة عددين كل منهما بطول 12 bits

الوظائف	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة
الوظائف	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة
الوظائف	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة
	المقارنة	المقارنة	المقارنة

يتم تحقيق المطلوب باستخدام أربع دارات 7485، بحيث تستخدم ثلاث دارات لمقارنة مكونات العددين ودائرة رابعة لربط مخارج الدارات الثلاث الأولى.



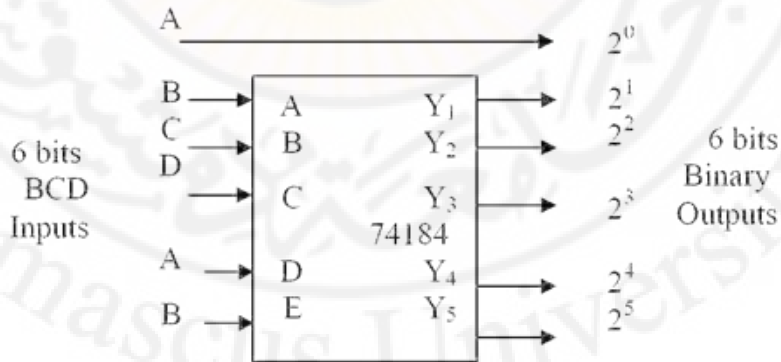
الشكل 8-39: دائرة لمقارنة عددين كل منهما بطول 12 bits باستخدام الدارة 7485

8-13 محولات الأرقام Converters

تستخدم محولات الأرقام لتحويل الأعداد من شكل لآخر. وبالرغم من وجود العديد من المحولات لتتجز التتابع المنطقية، نستعرض فيما يلي تركيبة نوعين فقط من المحولات، هما المحولات من الرماز BCD إلى النظام الثنائي وبالعكس.

1- المحولات من الثنائي المرمز عشرياً إلى الثنائي BCD to Binary converts

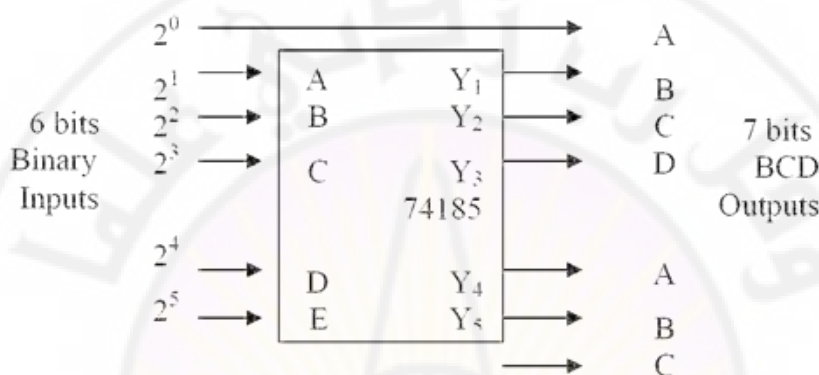
يقوم هذا النوع من المحولات بتحويل الأعداد من الرماز BCD إلى ما يقابلها في النظام الثنائي، وتصنع مثل هذه المحولات على شكل دارات متكاملة أيضاً. وتمثل الدارة المتكاملة 74184 مثالاً عن محول يقوم بتحويل الأعداد التي يصل طولها إلى 6 bits بالرماز BCD إلى ما يقابلها في النظام الثنائي. بالتالي يستخدم مثل هذا المحول لتحويل أي عدد على الدخل من أصغر قيمة بالـ BCD وهي 00 0000 حتى أعلى قيمة بالـ BCD وهي 11 1001 (أي ما يعادل رقم ونصف) إلى ما يقابله في النظام الثنائي، أي حتى العدد الثنائي $(100\ 111)_2$ الذي يقابل العدد العشري $(39)_{10}$. وبقية التراكيب الممكنة على الخرج من التركيبة (101000) حتى (111111) غير موجودة. وبالنظر إلى أن الخانة الأولى بكل من الرماز BCD والرماز الثنائي هي نفسها، تسمح الدارة 74184 بوصل خمسة مداخل بشكل مباشر، كما هو مبين في الشكل 8-40.



الشكل 8-40: المخطط العام لدارة المحول 74184

2- المحولات من الثنائي إلى الثنائي المرمز عشرياً Binary to BCD converts

يمثل المحول من ثنائي إلى BCD محولاً معاكساً للمحول السابق. وباعتماد الدارة 74185 التي تمثل مثالاً لمثل هذا المحول، يظهر الشكل 8-41 المخطط الصندوقي لاستخدام مثل هذا النوع من المحولات باستخدام الدارة 74185.

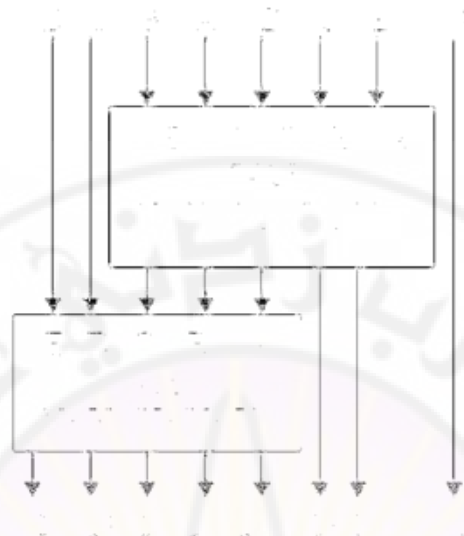


الشكل 8-41: المخطط العام لاستخدام المحول 74185 بتحويل الأعداد

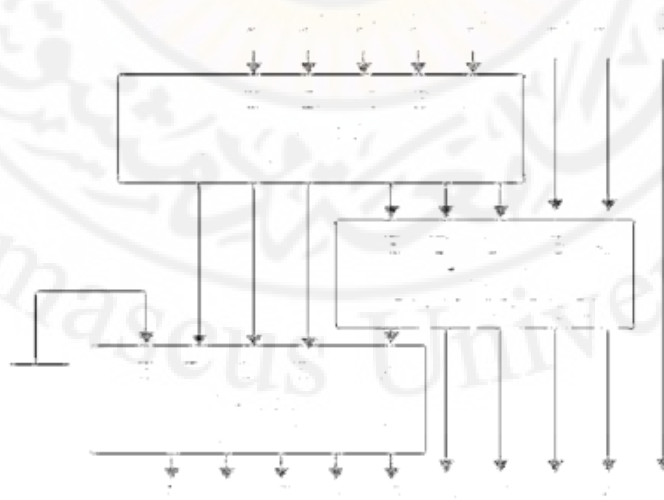
يلاحظ أن الدارة مكونة من ستة مداخل لإدخال قيمة بالرمز الثنائي وسبعة مخارج لإظهار القيمة المقابلة بالـ BCD. بالطبع أكبر عدد ثنائي يمكن تمثيله بـ 6 bits هو من الشكل $(11\ 1111)_2$ وهو يعادل أعداداً عشرية من 0 حتى 63 أو ما يقابل بالـ BCD أعداداً من 7 bits، لأن العدد 63 يمثل بالرمز BCD الشكل $(110\ 0011)$. ونعرض فيما يلي مثالاً واحداً عن كيفية استخدام كل نوع من محولات الأعداد.

مثال: صمم الدارة التي تقوم بتحويل أي عدد معطى بالـ BCD ومكون من رقمين ضمن المجال $(00 \div 99)$ إلى ما يقابلها في النظام الثنائي.

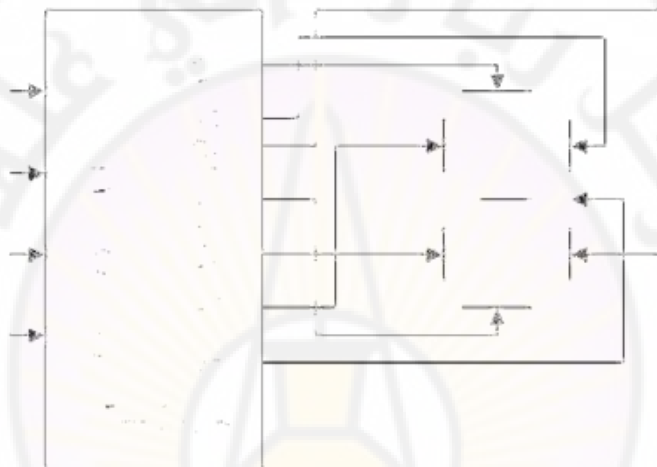
الجواب: باستخدام الدارة 74184 يمكن تحقيق المطلوب بدارتين، وفق ما هو مبين بالشكل أدناه، بحيث يكون عدد المداخل ثمانية $(1001\ 1001)_{BCD} = (99)_{10}$ وعدد المخارج ثمانية (يستخدم منها سبعة فقط بالنظر إلى أن العدد 99 يقابل ثنائياً $(110\ 0011)_2 = (99)_{10}$)



مثال: صمم دائرة لتحويل أي عدد ثنائي مكون من 8 bits إلى ما يقابله بالـ BCD.
 الجواب: عدد ثنائي بـ 8 bits من الشكل 1111 1111 يقابل عشرياً العدد 256 ، وسيكون
 المقابل بالـ BCD من الشكل 0 0101 0101 أي عدد من 10 bits. ويظهر الشكل أدناه
 مخطط التحويل باستخدام ثلاث دوائر من النوع 74185 (مخطط تحويل عدد ثنائي بطول
 8 bits إلى ما يقابله بالـ BCD).



وأخيراً نشير إلى أن استخدام النوعين المبيّنين أعلاه من المحولات قد يؤدي في كثير من التطبيقات إلى تبسيط التصميم. فمثلاً، الدارة المتكاملة 7446 بالرغم من أنها تصنف تحت فئة مفككات الترميز، تمثل شكلاً من أشكال المحولات، لأنها تقوم بتحويل أي رقم بالرماز BCD إلى سبعة مقاطع Seven Segment. حيث إن الشكل العام للدائرة وشكل ربطها بعنصر العرض السباعي، مبين في الشكل التالي:



يمكن رسم الدارة المنطقية، بغض النظر عن المعادلات المنطقية التي سبق وتوصلنا إليها ومخططات كارنوف باستخدام المحول 7446، وفق ما هو مبين في الشكل 8-42.



الشكل 8-42: مخطط دائرة إظهار الأرقام العشرية باستخدام السبعة مقاطع المضئنة

14-8 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

- 1- الدارة المنطقية بشكل عام
 - a- contains only logic elements
 - b- describe logic function
 - c- is a combinational and sequential
 - d- all the above
- 2- تضم الدارة المنطقية التركيبية
 - a- memory
 - b- no memory
 - c- flip-flops
 - d- counters
- 3- خرج الدارة المنطقية التركيبية يعتمد على
 - a- the its inputs only
 - b- the its previous state only
 - c- the its inputs and previous state
 - d- all the above
- 4- أي من العناصر المنطقية التالية يمثل دارة منطقية متتابعة؟
 - a- decoder
 - b- multiplexer
 - c- R-S Flip-Flop
 - d- comparator
- 5- خرج الجامع التام S (مع المدخلين a, b) هو
 - a. $S = a + b$
 - b. $S = a + b + c_{in}$
 - c. $S = a \oplus b \oplus c_{in}$
 - d. $S = a \oplus b$
- 6- عدد الجوامع التامة التي تلزم لجمع عددين كل منهما مكون من n رقم هو .
 - a. $N + 1$
 - b. $N - 1$
 - c. N
 - d. $2N$
- 7- المخطط الصندوقي لجامع / طارح على التوازي بعرض 4 bits مثلاً يحتوي
 - a- 4 full adders & 4 full Subtractor
 - b- 4 full adders
 - c- 4 full Subtractors
 - d- All the above
- 8- المعاقب Multiplexer دارة تركيبية تملك
 - a- 1 input many output
 - b- many inputs 1 output
 - c- many input many output
 - d- one input one output
- 9- من أجل أربع خطوط انتقاء يمكننا بناء معاقب
 - a. 4×1
 - b. 16×1
 - c. 1×4
 - d. 8×1
- 10- مفكك الترميز Decoder يمثل دارة تركيبية بـ
 - a- n- inputs and n- outputs
 - b- 2^n inputs and n- outputs
 - c- n- inputs and 2^n - outputs
 - d- n- inputs and one outputs

ثانياً: أسئلة متفرقة

- 1- صمم (ارسم) الدارة المنطقية التي تشير فقط للأعداد التي تقبل القسمة على 3 من مجمل الأعداد 0 حتى 25.
- 2- صمم دارة تركيبية بأربعة مداخل A, B, C, D وأربعة مخارج W, X, Y, Z، بحيث إذا كانت قيمة الدخل عدداً زوجياً، تكون قيمة الخرج أكبر بمقدار واحد. وإذا كانت قيمة الدخل عدداً فردياً، تكون قيمة الخرج أصغر بمقدار واحد.
- 3- أوجد جدول الحقيقة لتابع منطقي من الشكل: $F = ab + ab' + b'c$ وعبر عنه بصيغة: أ- الشكل الجامع. ب- الشكل الضارب.
- 4- صمم دارة تركيبية تعمل على تحويل أي رقم عشري من 0 حتى 9 إلى ما يقابله برماز كري Gray Code.
- 5- صمم بالدارات المتكاملة التي تعرفها دارة آلة حاسوبية بسيطة لإجراء عمليات الجمع والطرح والضرب على أعداد عشرية، كل منها مكون من خانتين.
- 6- باستخدام الرموز المنطقية فقط، ارسم شكل مفكك ترميز 5 x 32 Decoder بما يلزم من مفككات ترميز 3 x 8 Decoders (مدخل التأهيل فعال بالواحد).
- 7- صمم الدارة التركيبية المعرفة بالتتابع المنطقية:
$$F1 = ab'c' + a'b$$
$$F2 = a'b'c + ab$$
 باستخدام مفكك ترميز المناسب وما يلزم من بوابات منطقية خارجية .
- 8- صمم دارة جامع كامل بما يلزم من مضمّمات 4 x 1 MUX .
- 9- صمم دارة تركيبية بسيطة لمقارنة عددين طول كل منهما 5 bits بحيث الخرج سيكون 1 إذا كان العددان متساويين، وغير ذلك يكون الخرج 0 (عدم تساوي).
- 10- بين بشكل صندوقي فقط، وباستخدام المحولات المناسبة آلية تحويل:
أ- عدد مكون من أربع خانات بال BCD إلى ما يقابله في النظام الثنائي.
ب- عدد ثنائي مكون من 16 bits إلى ما يقابله بال BCD .

الفصل التاسع: الدارات المنطقية المتتابعية Sequential Circuits

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 المتتابعية.

9-1 المفهوم العام للدارات المتتابعية

للتعريف بالدارات المتتابعية، يمكن الانطلاق من مثال بسيط، كالدارة المبينة بالشكل أدناه التي تتكون من عنصرين منطقيين (كل منهما من النوع التركيبى) 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 تحديد 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:

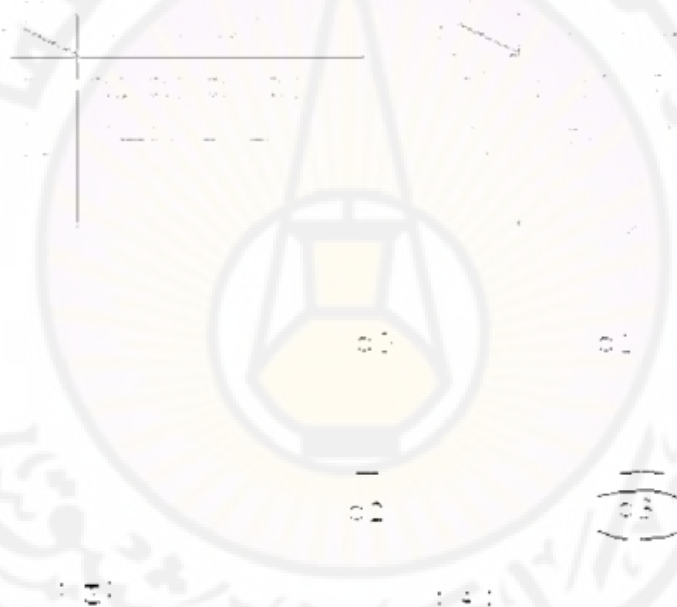
الأولى: إذا كانت القيمة السابقة لـ F صفراً تصبح القيمة الجديدة لـ F صفراً أيضاً.
 الثانية: إذا كانت القيمة السابقة لـ F واحداً تصبح القيمة الجديدة لـ F واحداً أيضاً.
 بالتالي جدول الحقيقة لم يعد يصلح لتوصيف مثل هذه الدارة ولابد من استبداله بجدول آخر، لأن خرج الدارة لا يمكن تحديده بالقيمة الموجودة على 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

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 Previous أو Present) 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) ويغير عن حالة داخلية للدارة Internal state مخزنة بالتغذية الراجعة. بالتالي الدارة أعلاه تتصرف كأن لديها ذاكرة للحالة السابقة للدارة. وهكذا، إذا رمز للتراكيب: 00 01 10 11 الممكنة لمتحولات الدخل a و b بالرموز: X_0, X_1, X_2, X_3 على التوالي، وللقيمتين 0 و 1 للمتحول الداخلي ب Q_p ، يصبح الشكل الجديد لجدول الحقيقة، وفق ما هو مبين أدناه.



يسمى الشكل الجديد لجدول الحقيقة بجدول الحالات الداخلية أو جدول العبور وهو الجدول الذي يعبر 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 $X_1, X_2, \dots, X_n$ ولقيم الخرج المقابلة (أبجدية الخرج) بـ $Y_1, Y_2, \dots, Y_m$ وللحالات الداخلية (السابقة) للدارة بـ $q_1, q_2, \dots, q_p$ ، يمكن

توصيف الدارة التتابعية بأربعة أشكال، وفق ما هو مبين في الشكل 9-1. بحيث يمثل الشكل 9-1 أ جدول عبور والنتاج هو قيمة الحالة اللاحقة (Q_n) للدارة، أما الشكل 9-1 ب، فيمثل جدول عبور والنتاج فيه هو قيمة الخرج y للدارة. يمكن أن يعبر عن الحالة اللاحقة للدارة والخرج في جدول واحد، كما هو مبين في الشكل 9-1 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. يُعبر عن الشكل 9-1 ج بشكل آخر وفق ما هو مبين في الشكل 9-1 د ويسمى بمخطط الحالة.



الشكل 9-1: 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 ، وتمت 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 المتبقية. 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 المتتابعةية على أنها دارة تركيبية بممرات للتغذية الخلفية، كل منها يمثل عنصر تخزين. وبحسب التركيبة العامة لممر التغذية الخلفية أو نظام فعالية الدارة المتتابعةية، تصنف هذه الدارات إلى نوعين أساسيين هما دارات متتابعةية متزامنة بوجود Clock ودارات متتابعةية غير متزامنة. أهم ما يميز الدارات المتتابعةية على مختلف أنواعها عنصر التخزين (الذاكرة)، ومن أكثر عناصر الذاكرة استعمالاً في الدارات المتتابعةية 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 نبين تركيبية الدارة ثنائية الاستقرار.

9-2 الدارات ثنائية وأحادية وعديمة الاستقرار

بالإضافة إلى الدارات أحادية وعديمة الاستقرار التي تستخدم بتوليد إشارات بتواترات مختلفة، يمكن أن تصادف 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 المتتابعةية.

أولاً- 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 9-2 أ. بالنظر إلى دارة عديم الاستقرار، يلاحظ أنه إذا كان الترانزستور T_2 مفتوحاً في الحالة الابتدائية و T_1 مغلقاً، يشحن بهذه الحالة المكثف C_2 بكامل جهد المصدر U_{cc} . 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.

الشكل 2-9: دائرة أحادي وعديم الاستقرار

تتكمّل 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 $t_2 \approx 0.7 C_2 R_2$ ، ولا تستقر الدارة بأي حالة.

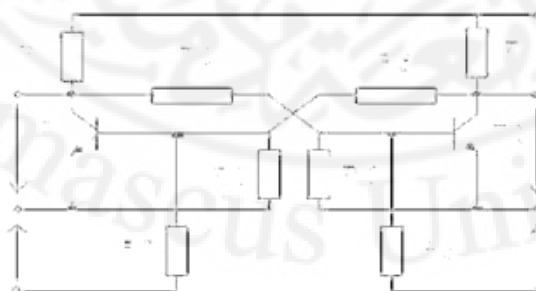
ثانياً - الدارات أحادية الاستقرار

بأسلوب مماثل لما سبق، نجد أن توصيلة دائرة أحادية الاستقرار Monostable، مبيّنة 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 و R_5 على T_1 في حالة القطع. يشحن المكثف C عملياً بكامل جهد المصدر U_{cc} 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 $t_1 = 0.7 R_3 C$. 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 U_{cc} وينتج دائرة أحادية الاستقرار، مستقرة في وضع (خرج T_2) وغير مستقرة في الوضع الآخر (خرج T_1).

ثالثاً - دائرة ثنائي الاستقرار

تتضمن دوائر ثنائي الاستقرار Bistable Circuit ترانزستورين يعملان بنظام الإشباع ويشكلان من أجل تنظيم محدد للروابط المتبادلة ما يسمى بدائرة الوصل ثنائية الاستقرار، كما هو مبين في الشكل 3-9، حيث بمجرد وصل التغذية الكهربائية يمكن أن يفتح أي من الترانزستورين T_1 أو T_2 (ولكن لأمر 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 و T_2 مغلقاً إلى فترة غير محدودة (أي أن الدائرة تستقر بواحدة من حالتين)، حتى يتغير الجهد u_1 (أو u'_1) إلى الحالة المعاكسة التي فيها يصبح T_2 مفتوحاً و T_1 مغلقاً. تنتقل الدائرة من حالة إلى أخرى بتطبيق جهد سالب على الدخل U_1 (أو جهد موجب على U'_1). يظهر هذا التغيير على المخارج 2 و 2' للدائرة ثنائية الاستقرار. تشكل هذه التوصيلة الأساس في تشكيل القلايات 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.



الشكل 3-9: التوصيلة الأساسية لدائرة ثنائي استقرار

3-9 القلابات Flip-Flops

يمثل القلاب دائرة تناظرية 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 للدوائر التناظرية 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 التناظرية بالمعادلتين، تابع النقل: $Q_{s,t+1} = \phi(X_{i,t}, q_{k,t})$ 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 $Q_{s,t+1} = \phi(X_{i,t}, q_{k,t})$ وتابع الخرج: $Y_{j,t} = \Psi(q_{k,t})$ ويستخدم عند التعامل مع هذين النموذجين تعبير آلات الحالة المحدودة Finite State Machines اختصاراً FSM. يتم تزامن المخارج في الدوائر التناظرية من نوع مور بنبضة الساعة، أما المخارج في الدوائر التناظرية 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 دوائر تناظرية نموذج مور وكل منها يمثل ذاكرة بطول 1bit. يوجد أربعة أنواع للقلابات هي: R-S و J-K و T و D بعضها له مدخل واحد عدا مدخل الـ Clock (مثل T, D) والبعض الآخر له مدخلين (مثل RS, JK) لكن لكل من القلابات الأربع مخرجان أحدهما مقيم للأخر ويعطي الحالة اللاحقة للقلاب. تُعد القلابات بشكل عام عناصر أساسية في تصميم الأجهزة الإلكترونية الرقمية من عدادات، وسجلات، وغيرها من أجهزة التحكم، وبالتالي 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 دائرة 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 القلايات 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 بوصفه دائرة منطقية من النوع التتابعي من الأوجه العشرة المذكورة أعلاه، نظراً لإمكانية تعميم كثير من النقاط لهذا القلاب على القلايات 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 الإلكتروني بالشكل المبين أدناه والذي يمثل قلاب (من أجل قيم

تقريبية للمكونات (SR latch) ($R1, R2 = 1\text{ k}\Omega$, $R3, R4 = 10\text{ k}\Omega$) 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 9-4.



الشكل 9-4: المخطط المنطقي للقلاب RS

يضم المخطط المنطقي جزءاً تركيبياً وخطاً راجعاً من المخرج 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 9-4 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 للدائرة. 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 للدائرة 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 9-1. بالطبع يمكن بأسلوب مماثل تشكيل الجدول الوظيفي للقلاب RS من البوابات NAND. يتضح من المخطط المنطقي للقلاب RS في الشكل 9-4 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.

الجدول 9-1: 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 مبين بالجدول الوظيفي. وبالنسبة 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 للدائرة 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 $R = S = 1$ غير مسموح بها، لأنها تعني للقلاب أن يكون بنفس اللحظة بكل من حالتي التعتيئة والتصفير وهذا ليس له معنى (علاوة على أن قيمة كل من Q_n و Q'_n ستكون 0 وهذا غير ممكن)، ويعبر عن ذلك في الجدول الوظيفي بحالة d. وهنا نشير إلى أنه إذا كان القلاب RS مبنياً بالبوابات NAND فإن الجدول الوظيفي يتغير، بحيث التركيبة $R = S = 1$ تمثل حالة الذاكرة والتركيبة $R = S = 0$ 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 أحد 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

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

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

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

$$Q_n = S + R'Q_p$$

خامساً- جدول التحريض Excitation table

يقصد بجدول التحريض (أو جدول التهييج) بالنسبة لأي قلاب تحديد 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 ما بين الحالة السابقة Q_p والحالة اللاحقة Q_n للقلاب، فإن كل جدول من هذا النوع لكل قلاب سيتكون من أربعة أسطر، يحدد فيه كل سطر وضعية محددة ما بين الحالتين السابقة واللاحقة. وبالنسبة للقلاب RS تحديداً فإن جدول التحريض يبدو كما هو مبين في الجدول 9-2.

الجدول 9-2: 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

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 بالصفـر وسيبقى بالصفـر. 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 تركيبان ممكنان على الدخل في كل منهما $S = 0$ ، أما R فيمكن أن تأخذ أيًا 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 S-R 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 (CP) Clock Pulse. 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 تعيين 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 دائرة القلاب الأساسية، كما هو مبين في الشكل 5-9 (البوابات 3 و 4) للتحكم بالقلاب وحمايته من التغيرات الطارئة.

الشكل 5-9: 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 خرجة إلا عندما يقدر باستخدام مفتاح خارجي يمثل مدخلاً للتحكم (وهكذا تتم حماية خرج الدارة من أثر أي تغير غير مرغوب به على الدخل).

الشكل 6-9: الرموز المعتمدة للاستجابة للساعة (المستوى والحافة)

تتبدل حالة القلاب المتزامن عند حدوث تغيير في مدخل التحكم ويدعى هذا التغيير 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 المحدد 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 أ. 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 للدائرة التتابعية، يظهر بعض المشاكل، لأن أي تغيير 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 الفعال. 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 للدائرة 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

للوّجه الهابط للنّبضة 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 النّبضة عادة عند قيمة الـ 50% من الإشارة أي بالنقطة التي تتوسط أطراف النّبضة وبالتالي يمكننا كما هو واضح من المخطط تعريف زمن التهيئة، بأنه الفترة الزمنية من بداية تطبيق نبضة المعلومات إلى لحظة تبدل مستوى النّبضة الساعية من المستوى العالي إلى المستوى المنخفض. أما زمن الإمساك، فيمثل الفترة الزمنية بين لحظة تبدل مستوى النّبضة الساعية وبين نهاية نبضة المعلومات (الزمن المتبقي من نبضة المعلومات).

الشكل 7-9: 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 بالنّبضة.
 - الزمن الاصغري الذي يدعى بزمن الإمساك الذي يجب فيه أن تستمر القيمة على الدخل من دون تغيير بعد تطبيق الانتقال الموجب للساعة.
 - زمن تأخير الانتشار Propagation delay time للقلاب والذي يعرف بالمدة التي تفصل بين حافة الفدح ولحظة استقرار الخرج عند الحالة الجديدة.
- يمكن التعبير عن القلاب بمخطط زمني يلحظ تغيرات الحالات الداخلية، ويظهر تأخير الإشارة. فمن أجل قلاب RS متزامن، يستجيب للوجه الصاعد للنّبضة 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 يبين 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 القدرح يجب أن يكون لها عرض أصغري معين، وأن الفترة بين التغيرات المتتالية لمتحول واحد يجب ألا 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 ألا 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 القدرح الأصغري عادة من المواصفات الأساسية التي تعطي 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 القدرح، وغالباً ما يعطى على شكل تردد أعظمي مثل 10MHz، وهذا يقابل 100 ns كفترة أصغرية بين نبضات الإطلاق، وبشكل عام يكون الزمن الأصغري الفاصل بين النبضات أكبر بكثير من العرض الأصغري 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 الساعية 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

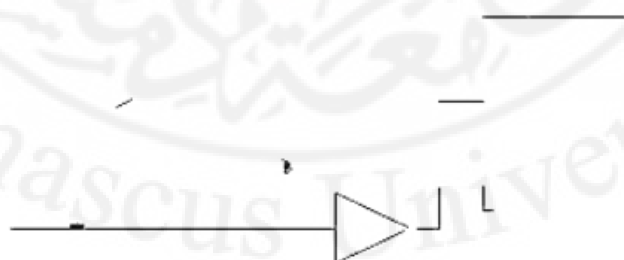
نبيّن فيما يلي المفهوم العام لدارة القلاب من نوع قائد-مقاد، على أن نعود لإظهار المفهوم العام للقلاب الذي يقدح على الجبهة (الطرفي) لاحقاً. ويمكن تجاوز مشاكل التزامن المتعلقة بعرض النبضة Pulse width وزمن تأخير الإشارة Delay time باستخدام قلابات أكثر تعقيداً، هي قلابات القائد - مقاد، وهي قلابات تستخدم عادة 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 لحمايتها من أي تغير غير مرغوب به. الشكل العام لدارة قائد - مقاد، مبينة في الشكل 9-9. يجب لضبط عملية التزامن بشكل صحيح ألا تظهر قيم جديدة على RS أثناء وجود نبضة الساعة، وإنما يجب تغيير قيم المداخل قبل ظهور المستوى العالي للنبضة.



الشكل 9-9: المخطط المنطقي لقلاب RS نوع قائد - مقاد

يمكن التعبير عن دارة قلاب من نوع قائد/ مقاد صندوقياً، وفق ما هو مبين بالشكل 9-10، 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.



الشكل 9-10: 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 دائرة لقلب من نوع قائد - مقاد، يقاد بالأحداث لالانبضة الساعية، أي غير متزامن، وفق ما هو مبين في الشكل 9-11، حيث يتضح من دائرة 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:

الشكل 9-11: المخطط المنطقي للقلب RS غير المتزامن

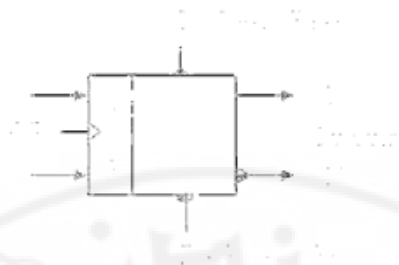
$R = 1, S = 0$ و $R = 0, S = 1$. كما أن الحالتين المتبقيتين: $R = S = 1$ و $R = S = 0$ غير ممكنتين لأن هذه المداخل تمثل مخارج زوج البوابات المتصالب للقائد. وبما أنه لا يمكن أن يكون $Q_n = Q'_n$ يختصر جدول العبور من أجل جميع قيم الوضع والإرجاع إلى قيمتين تقابلان الحالات التي يكون فيها:

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

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

تاسعاً - القلب بالمداخل المباشرة F.F with direct inputs

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



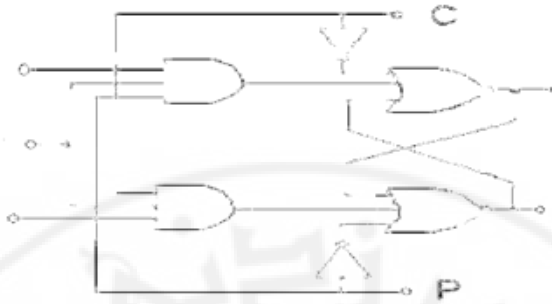
الشكل 9-12: قلاب RS بمدخل خارجية مباشرة

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 فعالين بالمستوى "0" أو بالمستوى "1" بحسب نوع الدارة. بالطبع لا يجوز أن يفعل المدخلين (بالصفر) معاً. وبالتالي جدول الوظائف لهذا النوع من القلابات يبدو كما في الجدول 9-3.

الجدول 9-3: جدول الوظائف للقلاب RS بمدخل 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

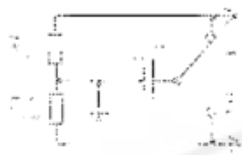
المخطط المنطقي الذي يمكن أن يحقق جدول الوظائف أعلاه، يمكن أن يكون كما هو مبين في الشكل 9-13. يتضح من المخطط المنطقي، إذا كان $P = C = 1$ (غير فعالين) سيكون خرج كل من العاكسين صفراً، وبالتالي خرج كل من بوابتي NOR المشكلتين للقلاب يصبح مرتبطاً بالقيم الموجودة على كل من المدخلين R و S .



الشكل 9-13: المخطط المنطقي لقلاب RS بمدخل 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 بالدائرة الكهربائية بالشكل 9-14 أ. 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، دائرة 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 9-14 ب، لتوليد معطيات ثنائية والانتقال من 0 إلى 1 مرة واحدة عند ضغط المفتاح.



الشكل 14-9: دائرة 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. إذا أصبح $R = 0$ و $S = 1$ ، وينتقل الخرج إلى القيمة واحد ($Q_n = 1$). الاهتزاز المتكرر في النقطة B وتوليد عدة نبضات لم يعد 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 $Q_n = 1$ ، 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 كدائرة متكاملة جاهزة للتوصيل والاستخدام المباشر على عكس غيره من القلابات الأخرى وبخاصة القلابين D و JK التي تتواجد بكثرة كدوائر متكاملة، وذلك لعدم الحاجة لوجود RS كدائرة 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 دائرة 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

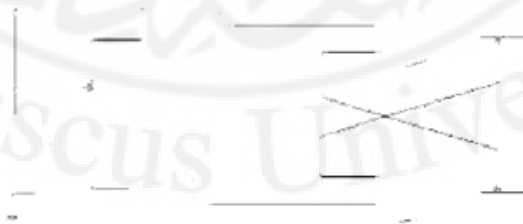
يمكن عرض الرمز المنطقي للقلاب D بوجود مدخل خاص للساعة وآخر خاص بالمحو غير المتزامن R (فعال بالصفر مثلاً)، وفق ما هو مبين في الشكل أدناه.



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

2- المخطط المنطقي

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



الشكل 9-15: المخطط المنطقي للقلاب D

3- الجدول الوظيفي

يمتاز القلاب D انطلاقاً 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 9-4.

الجدول 9-4: 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 9-5.

الجدول 9-5: 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 المتتابعية، أصبح بإمكاننا التعريف بالطريقة الثانية التي سبق وأشرنا إليها، 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 التغيير 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 9-16، التي تُعد أكثر فعالية للقلاب D، لأنها تقدح بالحافة الموجبة لثلاثة مواسك RS بأن واحد، اثنين منهما يستجيبان للمداخل D و CP والثالث يمثل مخرجاً للقلابين الأوليين.



الشكل 9-16: قلاب D يقدح بالحافة

تم في المخطط المنطقي للقلاب D الذي يقدح بوجه النبضة استخدام بوابات NAND، وبالتالي التركيبية التي تقابل $R = S = 0$ غير مسموح بها. ولنظير 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 للدائرة $R = 1$ 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 المساعد للنبضة ظير 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 المساعد 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. وأخيراً 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 دائرة 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، وب نفس الوقت تعطي إنذاراً 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 9-17.

قيمة - صفر



الشكل 9-17: دائرة مراقبة بسيطة باستخدام القلاب D

يتضح من الشكل 9-17 الذي يضم فقط قلاباً واحداً من نوع D مع بوابتين أنه طالما قيمة الإشارة القادمة من حساس الحرارة هي صفر (الوضع سليم) ستبقى اللبة 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 به 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 9-18.

الشكل 9-18: الرمز المنطقي للقلاب T

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

الجدول 9-6: 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:

$$Q_n = T \cdot Q_p' + T' \cdot Q_p = T \oplus Q_p$$

أما جدول التحويل الذي يحدد القيمة التي يفترض تطبيقها على المدخل T للانتقال من حالة إلى حالة أخرى مبين في الجدول 9-7، 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 $T = 0$.

الجدول 7-9: 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 بواسطتها 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 بالضغط الثانية على نفس الكبسة أن يفعل 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 كل 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 يبدل حالته، ويصبح Q_0 مساوياً للواحد. وهذا الواحد عبر العاكس سيصبح صفراً، وبالتالي خرج NAND سيصبح 1 ويعمل الإنذار الصوتي.



الشكل 19-9: نظام تحكم صوتي 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، سيبدل 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 (أ) أدناه. ويظهر الشكل (ب)، تشكيل القلاب J-K بالبوابات NAND بوجود مدخل مباشر للمحو غير المتزامن R.

(أ)

(ب)

يُعدّ القلاب J-K معرّفاً من أجل جميع التراكيب على الدخل ولا توجد أية تركيبية محظورة، كما هو الحال بالنسبة للقلاب RS. وصمم القلاب J-K، بحيث التركيبية $J = K = 1$ تصبح مسموحة. وتكون الحالة اللاحقة للقلاب من أجل هذه التركيبية على الدخل بعكس الحالة السابقة. ولكي تكون التركيبية $J = K = 1$ مقبولة، وتعطي خرجاً $Q_n = Q'_p$ ، أخذت تركيبية 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 9-20. 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 تركيبيتين عندما تكون $J = K = 1$ و $Q_p = 0$ أو $Q_p = 1$.

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.

الجدول 9-9: 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 القلايات الأخرى، ويظهر الشكل 9-21 مثالاً عن دائرة 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.

الشكل 9-21: شكل توزع مأخذ القلايات J-K في الدارة 7476

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

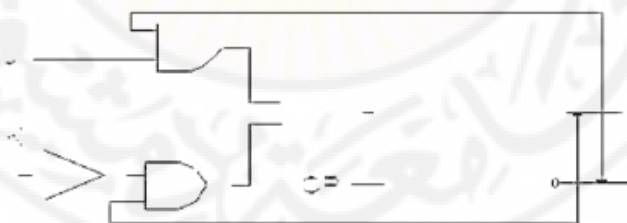
$$T = Q \oplus D$$

الشكل 9-22: 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 9-23.

الشكل 9-23: 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 9-24.



الشكل 9-24: تشكيل القلاب $J-K$ من القلاب D

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

الجدول 9-10: أرقام بعض الدارات المتكاملة المصنعة بالتقانة TTL والتي تمثل قلابات

رقم الدارة	وظيفة (تركيبية) الدارة	ملاحظات
74 70	J – K Flip-Flop AND Inputs (74 72)	DIL – 14
74 73	Dual J – K Flip-Flop (74 78 & 74 103& 74 107& 74 108..)	DIL – 14
74 74	Dual D-Type Flip-Flop	DIL – 14
74 75	Quad Latch = 4x D-Type Flip-Flop	DIL – 16
7476	Dual J – K Flip-Flop(74 106& ..)	DIL – 16
74 101	J – K Flip-Flop	DIL – 14
74 110	J – K Flip – Flop Data Lockout	DIL – 14
74 111	Dual J – K Flip – Flop Data Lockout	DIL – 16
74 116	Dual Quad Latch	DIL - 24
74 118	Hex S – R Latch	DIL – 16
74 119	Hex S – R Latch	DIL - 24
74 121	Monostable Multivibrator	DIL – 14
74 123	Dual Monostable Multivibrator	DIL – 16
74 174	Hex D –Type Flip- Flop	DIL – 16
74 175	Quad D –Type Flip- Flop	DIL – 16
74 273	Octal D –Type Flip- Flop(& 74 374)	DIL - 20
74 276	Quad J – K Flip – Flop (& 74 376)	DIL - 20
74279	Quad R – S Latch	DIL – 16

4-9 أسئلة وتمارين

اختر الجواب الصحيح لكل من التساؤلات التالية:

1- أي من العناصر المنطقية التالية يمثل دائرة تتابعية؟

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

2- القلاب يمثل دائرة

- 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

- a. $R = S = d, C = P = 0$ واحد، عندما:
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 العاشر: السجلات Registers

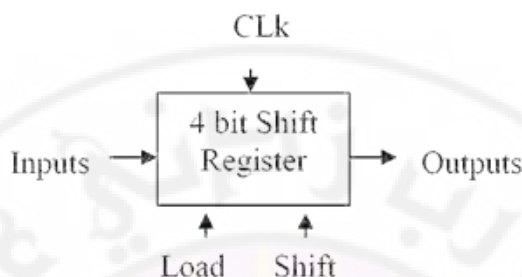
تُعدّ السجلات واحدة من أهم الدارات المنطقية التتابعية، نظراً لاستخداماتها العديدة في مختلف الأنظمة الرقمية. والسجل Register هو عبارة عن دائرة تتابعية مكونة من مجموعة ثنائيات استقرار (قلابات)، بالإضافة إلى بعض البوابات المنطقية. يستخدم السجل بشكل رئيسي لتخزين الأرقام الثنائية، بحيث كل خلية بالسجل تمثل 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 البتات التي يتشكل منها. فمثلاً، لتخزين كلمة مكونة من أربعة أرقام ثنائية مثل الكلمة $(1110)_2$ ، يلزم سجل مكون من أربعة قلابات موصولة مع بعضها بطريقة ما، تجعل من هذه البتات تشكل مجموعة متكاملة عند التعامل معها. نعرف فيما يلي على التركيب العامة للسجلات من خلال أمثلة على سجلات بطول كلمة 4 bits فقط، بحيث بتات السجل الواحد تكون مرتبة بجانب بعضها، وفق ما هو مبين في الشكل 10-1 بالترتيب من أقصى اليمين إلى أقصى اليسار.

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

الشكل 10-1: سجل بطول 4 bit

يمكن تعميم كل ما ينطبق على سجلات بطول 4bits على سجلات بأطوال كلمة مختلفة. وبالنظر إلى أن 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. يظهر الرمز المنطقي للسجل في الشكل 10-2 طول كلمة السجل وبعض الإشارات القيادية،

حيث يلاحظ من الرمز المنطقي للسجل أنه خلال التعامل مع السجل، نصادف بالإضافة إلى الاسم (الذي يدل على الوظيفة) طول الكلمة، ومجموعة إشارات أخرى مثل:



الشكل 2-10: الرمز العام للسجل

- المدخل Inputs: قد تكون مدخلاً واحداً يستخدم لإدخال المعطيات للسجل أو عدة مدخل مساوية لطول السجل.
- المخرج Outputs: تستخدم لإظهار محتويات السجل وعددها قد يكون مخرجاً واحداً أو عدة مخرج بعدد خلايا السجل.
- نبضات الساعة CLK، حيث يوجد لكل قلاب يدخل في ترقية السجل مدخل خاص لنبضات الساعة التي تقود عمل السجل.
- إشارات تحكم أخرى مثل Load, Shift, Reset, ... وقد توجد كلها بالسجل الواحد.

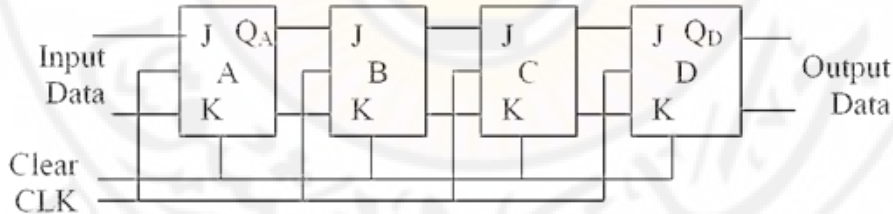
1-10 سجلات الإزاحة

يسمى السجل الذي يستخدم لإزاحة البيانات، بسجل الإزاحة Shift Register وهو ذو تركيبة بسيطة. يمكن أن نصادف في التطبيقات العملية سجلات بأطوال قياسية مثل سجلات: 4, 8, 16, 32, 64, ... bit انطلاقاً من الأطوال المتعارف عليها لطول كلمة الحاسوب. وبحسب آلية التوصيل الداخلية للسجل نميز بين ثلاثة أنواع من سجلات الإزاحة، هي:

- سجلات الإزاحة نحو اليمين.
- سجلات الإزاحة نحو اليسار.
- سجلات الإزاحة الحلقية.

10-1-1 سجلات الإزاحة نحو اليمين

يرتبط مفهوم إزاحة البيانات أو نقلها بواسطة سجل بطريقة إدخال البيانات وإخراجها. وبالتالي لتوضيح التركيبة العامة لسجلات الإزاحة نحو اليمين (وغيرها من السجلات الأخرى) ننطلق من الترميز المعتمد للبيانات بالسجلات. إن الترميز المعياري للبيانات ضمن سجل هو أن البتّ بأقصى اليمين دوماً يمثل البتّ ذا الوزن الأدنى LSB، والبتّ بأقصى اليسار يمثل الـ MSB. فمثلاً، عند تمثيل المعطيات $(1110)_2$ بسجل طول كلمته 4 bits مثل السجل المبين في الشكل 10-1، يجب أن نحرص على ترميز $D_0 = 0$ بأنها الـ LSB وبالتالي يجب أن 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 ما هو موجود بالخانة MSB نحو اليمين وإخراج القيمة الموجودة بالخانة التي تقابل الـ LSB خارج السجل. والتركيبية العامة لمثل هذا السجل بطول 4bits، مبينة في الشكل 10-3. والذي يتضح منه أن كل خانة من السجل خصصت بقلاب، تم ترميز القلابات الأربعة بالأحرف A, B, C, D، بحيث إن القلاب A يقابل الخانة LSB والقلاب D يقابل الخانة MSB.



الشكل 10-3: سجل إزاحة نحو اليمين بطول 4 bits.

استخدام القلابات J-K في الشكل أعلاه، علماً أنه كان بالإمكان استخدام أي نوع آخر من القلابات. تُعدّ دائرة السجل أعلاه بسيطة، لأنها مكونة من القلابات فقط، وتسمح بإزاحة البيانات نحو اليمين عند كل إدخال للمعطيات من القلاب A (متوافقاً مع ورود النبضة الساعية) من دون الحاجة لوجود أي بوابة منطقية إضافية. تقدح الساعة المشتركة CLK جميع القلابات بأن واحد. توصل جميع مداخل التفسير للقلابات الأربعة بأن واحد إلى

مدخل واحد غير متزامن Clear الذي يمكن أن يكون فعالاً بالصفر أو بالواحد. بمجرد تفعيل إشارة Clear، تصفر كافة القلايات دفعة واحدة، تستخدم عادة هذه الإشارة قبل بدء التعامل مع السجل. تتراح المعطيات المخزنة بالسجل باتجاه اليمين مع كل ورود للنبضة الساعية بتطبيق قيمة الواحد أو الصفر على المدخل الرئيسي للقلايا.

10-1-2 سجلات الإزاحة نحو اليسار

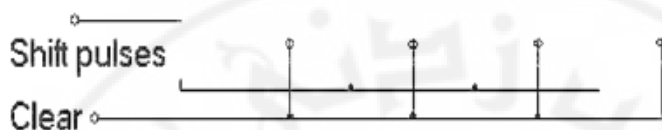
يسمح هذا النوع من السجلات Shift- Left Registers بإدخال بيانات الإزاحة من البت الذي يقابل خانة الـ LSB، ويقوم بإزاحة البيانات المخزنة بالسجل باتجاه البت MSB (اليسار). وبالتالي التركيبية العامة تمثل هذا النوع من السجلات بطول 4bits مبنية في الشكل 10-4. وهنا تم من باب التنويع استخدام قلايات من نوع D. كما أنه حدد بأن القلايات المستخدمة بالسجل تستجيب للوجه الصاعد للنبضة. وأن إشارة المسح الموصولة مع إشارة التصفير فعالة بالصفر.

الشكل 10-4: سجل إزاحة نحو اليسار بطول 4 bits

10-1-3 سجلات الإزاحة الحلقية

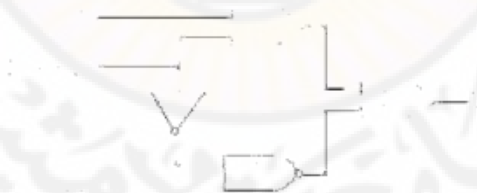
يلزم بسجلات الإزاحة نحو اليمين أو اليسار إدخال معطيات خارجية من مدخل النقل التسلسلي للسجل وإخراج بيانات من مخرج النقل التسلسلي حتى تتم عملية الإزاحة. أما بالنسبة للسجلات الحلقية Shift-Around Registers، فتدور البيانات التي تكون مخزنة مسبقاً بهذا النوع من السجلات من طرف إلى آخر من دون أي فقد بالبيانات. يمكن مع كل نبضة ساعية إجراء عملية الدوران وأخذ عينة عن المخرج. وهنا نميز أيضاً بين نوعين من سجلات الدوران، سجلات دوران باتجاه اليمين أو سجلات دوران باتجاه اليسار.

الشكل العام لسجل حلقي نحو اليمين بطول 4bits يبدو كما هو مبين في الشكل 5-10.



الشكل 5-10: سجل دوران (حلقي) نحو اليمين بطول 4 bits

يوجد وصلة راجعة من الخرج إلى الدخل، وبمعكس توصيل القلايات يصبح السجل حلقياً نحو اليسار. وهنا نشير إلى أنه في كثير من التطبيقات قد يلزم سجل واحد يقوم بأكثر من وظيفة، مثل إزاحة البيانات نحو اليمين أو اليسار في أثناء النقل التسلسلي للبيانات مثلاً، إضافة إلى تدوير البيانات. يمكن تحقيق مثل هذه الأنواع من السجلات تقنياً، بإضافة دارات تحكم خارجية للتركيب الأساسية للسجل لتعمل على انتقاء الوظيفة المطلوبة. وكمثال على هذه الدارات نورد الدارة المبينة في الشكل 6-10 التي تعمل على اختيار واحد من المدخلين A_1 و A_2 الذي يمكن أن يكون كل منهما خرجاً لسجل تسلسلي.



الشكل 6-10: منطق للتحكم باختيار واحد من مدخلين

تقوم الدارة بالشكل أعلاه بإظهار الدخل A_1 أو a_2 على الخرج Y بحسب إشارة التحكم وبغض النظر عن الوظيفة التي يؤديها السجل المختار وذلك وفق الآتي: إذا كان $Control = 0$ فإن $Y = A_2$ وإذا كان: $Control = 1$ فإن $Y = A_1$ يمكن توصيل مثل هذه الدارات مع سجل إزاحة بوظيفة للحصول منه على أكثر من وظيفة.

10-2 الأنواع الأساسية للسجلات

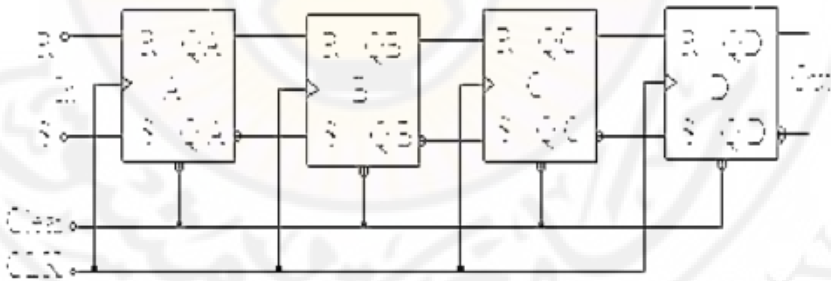
تصنف السجلات بحسب طرق إدخال البيانات وإخراجها إلى أربعة أنواع أساسية

- 1- سجل بدخل تسلسلي - خرج تسلسلي Serial In- Serial Out اختصاراً SISO.
 - 2- سجل بدخل تسلسلي - خرج تفرعي Serial In-Parallel Out اختصاراً SIPO.
 - 3- سجل بدخل تفرعي - خرج تفرعي Parallel In - Parallel Out اختصاراً PIPO.
 - 4- سجل بدخل تفرعي - خرج تسلسلي Parallel In - Serial Out اختصاراً PISO.
- وفيما يلي نعرض باختصار التركيبية العامة لكل من هذه الأنواع الأربعة .

10-2-1 سجل بدخل تسلسلي - خرج تسلسلي

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

مثال: من أجل سجل نوع SISO بطول 4bit وخلاياه مكونة من القلابات RS يكون المخطط المنطقي، وفق ما هو مبين في الشكل 10-7.



الشكل 10-7: المخطط المنطقي لسجل SISO بطول 4 bits

يتضح من الشكل أن القلابات المشكلة للسجل موصولة على التسلسل، بحيث خرج كل قلاب يكون موصولاً مع مدخل القلاب الذي يليه. وإن إشارة التحكم Clear للتصفير هي فعالة بالصففر وموصولة مع Reset جميع القلابات لتتم عملية التصفير لكامل السجل دفعة واحدة. توصل إشارة الساعة CLK إلى كل القلابات بأن واحد. تدخل البيانات بالتوصيلية المبينة أعلاه إلى السجل عن طريق نقطة دخول واحدة هي S وتخرج منه عن طريق نقطة

خروج واحدة Q_D . يتم التحكم بعملية إدخال البيانات وإخراجها بتاً بعد بتّ عن طريق نبضات الساعة. لنحاول الآن إظهار آلية إدخال القيمة $(1101)_2 = (13)_{10}$ مثلاً، لتخزينها في السجل المعطى بعد تصفيره. يبين الجدول 10-1 كيفية إدخال هذه القيمة في القلابات المرمزة A, B, C, D المكونة للسجل وإخراجها، حيث إن إشارة المحو عندما تكون فعالة لا يكون هناك أي دور للقيم الموجودة على المداخل R و S (ويعبر عن ذلك بوضع d تحت كل من R و S) وتصفّر جميع القلابات.

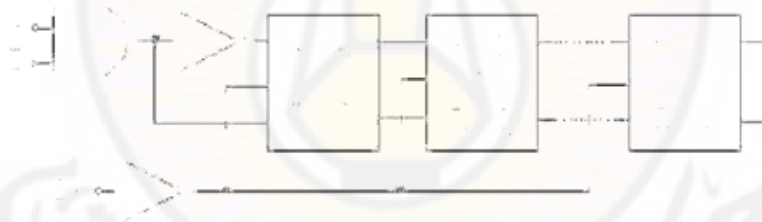
الجدول 10-1: آلية إدخال المعطيات إلى مسجل RS من نوع دخل تسلسلي - خرج تسلسلي

Input	R	S	A	B	C	D	Out
Clear	d	d	0	0	0	0	0
1	0	1	1	0	0	0	0
0	1	0	0	1	0	0	0
1	0	1	1	0	1	0	0
1	0	1	1	1	0	1	0 →

لإدخال قيمة واحد مثلاً بالقلاب RS نجعل $S = 1$ و $R = 0$ ولإدخال قيمة صفر نجعل $S = 0$ و $R = 1$. يلاحظ من الجدول 10-1، أن أي بتّ سيدخل للسجل سيتم من نقطة محددة هي مدخل القلاب A، ومنه مع كل نبضة ساعية يتم الانتقال إلى القلابات الأخرى بالتدريج. بالطبع في حال عدم وجود نبضة ساعية يحافظ السجل على حالته السابقة من دون تغيير. كما أن إخراج أي بتّ من السجل سيتم من نقطة محددة هي خرج القلاب D. وكمثال فعلي على هذا النوع من السجلات نورد للتطبيقات العملية في الجدول 10-2 بعض السجلات المصنعة على شكل دارات متكاملة. يظهر الجدول أرقام بعض الدارات المتكاملة التي يمثل كل منها نوع من السجلات، حيث الدارة ذات الرقم 7491A مثلاً، تمثل سجلاً بطول 8 bits، وهي بحسب ما هو وارد في كاتالوك الشركات الصانعة مكونة من 14 pin، كما هو مبين في الشكل 10-8 أ، علماً أنه باستثناء التغذية (U_{ss} , U_{cc}) والساعة CP لا يوجد من الـ 14 مأخذ لهذه الدارة مستخدم سوى أربعة مأخذ، كما هو مبين بالرمز العام للدارة في الشكل 10-8 ب.

الشكل 8-10: توزيع مأخذ الدارة المتكاملة 7491A ورمزها العام

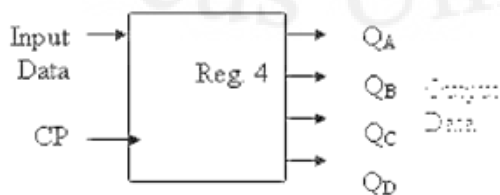
يتم إدخال قيمة 1 في السجل بتطبيق التركيبة $AB = 11$. ومن أجل أي تركيبة أخرى لـ AB سيتم إدخال الصفر. أما الخرج فيمكن أخذه إما من المخرج Q بشكله المباشر أو من Q' بصيغة المتمم. المخطط الداخلي (المنطقي) للدارة المذكورة، مبين في الشكل 9-10.



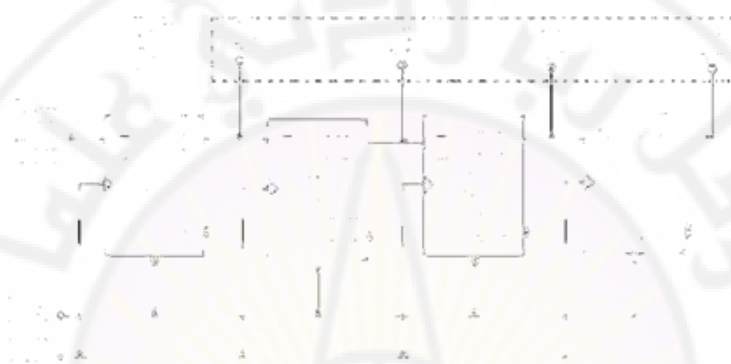
الشكل 9-10: المخطط العام للتركيبة الداخلية للدارة المتكاملة 7491A

10-2-2 سجل بدخل تسلسلي - خرج تفرعي

يختلف هذا النوع من السجلات عن سابقه، بأن له مدخلاً وحيداً وعدة مخرجات كما هو مبين في الشكل أدناه بالنسبة لسجل بطول 4 bits (بغض النظر عن نوع القلايات المستخدمة في تركيبته). أي أن البيانات تدخل بشكل تسلسلي، وتخرج من جميع القلايات المكونة للسجل بنفس الوقت.



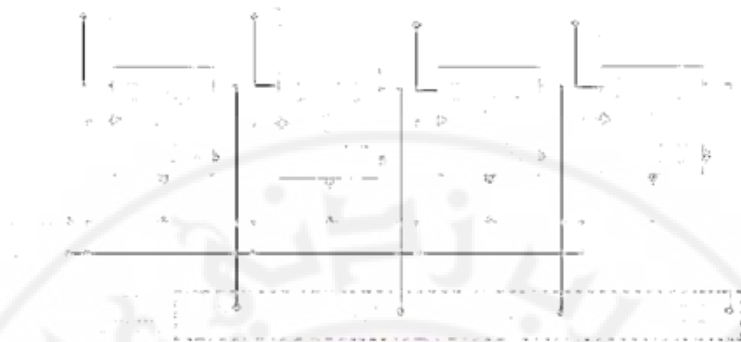
المخطط المنطقي لسجل من نوع SIPO باستخدام قلابات D يبدو كما هو مبين في الشكل 10-10. بالطبع كان بالإمكان أخذ الخرج من المخارج Q' لكل قلاب على حدة. وكمثال من الدارات المتكاملة على هذا النوع من السجلات، نورد الدارة المتكاملة ذات الرقم 74164 التي تمثل 8 bit Shift Register SIPO.



الشكل 10-10: المخطط المنطقي لسجل بطول 4 bits من نوع SIPO

10-2-3 سجل بدخل تفرعي - خرج تفرعي

يتم إدخال جميع البيانات بسجلات دخل تفرعي - خرج تفرعي (أو دخل توازي - خرج توازي) بلحظة واحدة. كما يتم إخراجها أيضاً من جميع خلايا السجل بنبضة ساعية واحدة. وتُعدّ سجلات دخل تفرعي - خرج تفرعي PIPO من أكثر أنواع السجلات استخداماً وخاصة مع المعالجات الصغيرة (بحيث يكون طول هذه السجلات عادة بطول كلمة المعالج، وبالتالي يمكن استخدامها كبوابات للدخل أو الخرج). يمكن باستخدام القلابات D مثلاً، إظهار المخطط المنطقي لسجل من هذا النوع بطول 4 bits، وفق ما هو مبين في الشكل 10-11. وكمثال من الدارات المتكاملة، نورد الدارة المتكاملة ذات الرقم 74195 المبنية من القلابات J-K. وهي تمثل سجلاً من نوع PIPO بطول 4 bits فقط. بمعنى أن مدخل كل قلاب من القلابات المكونة للسجل ومخرجه يكون مستقلاً عن مدخل ومخرج أي من القلابات الأخرى.



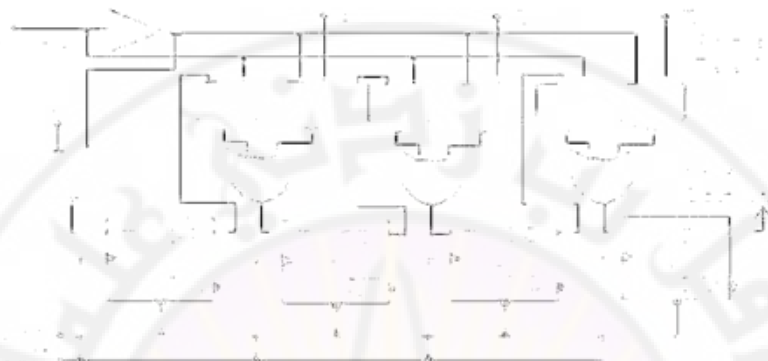
الشكل 10-11: المخطط المنطقي لسجل من نوع PIPO بطول 4 bits

ترتبط القلايات مع بعضها بإشارات التحكم فقط، مثل: CLK, Clear. بمجرد ورود النبضة الساعية، تدخل المعطيات الموجودة على المداخل الخارجية A, B, C, D بنفس اللحظة على التوازي. ونفس الشيء بالنسبة للخروج.

10-2-4 سجل بدخل تفرعي - خرج تسلسلي

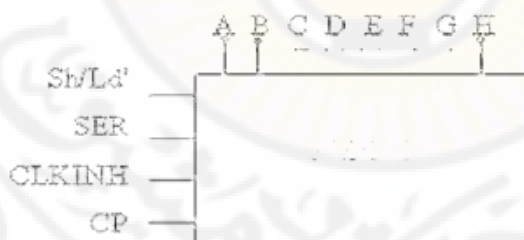
تُعدّ التركيبة العامة لسجلات هذه المجموعة من الناحية التقنية هي الأعقد. ولها استخدامات متعددة كتحويل المعطيات التفرعية من حاسوب ما إلى تسلسلية لنقلها عبر خط هاتف. يسمح هذا النوع من السجلات بإدخال المعطيات على التوازي بنبضة ساعية واحدة، أما إخراج البيانات فيتم من نقطة واحدة بإخراج كل بت بعد بت على التسلسل. يلزم لإخراج كامل البيانات المخزنة بالسجل من هذا النوع PISO عدد من النبضات الساعية يساوي عدد القلايات المكونة للسجل. يمكن أن تكون التركيبة العامة لسجل من نوع PISO بطول 4 bits مبني من القلايات D كما هو مبين في الشكل 10-12، حيث يلزم منطق إضافي للتحكم بعملية تعبئة البيانات في السجل في المرحلة الأولى، ومن ثم إزاحتها في المرحلة الثانية. يمكن تركيب إشارتي الإدخال Load (اختصاراً Ld) والإزاحة Shift (اختصاراً Sh) على ناقل واحد، بحيث تكون القيمة الفعالة لإحدهما بعكس الأخرى، وتكتب بالشكل Sh/Ld' . بالتالي إذا كانت قيمة الإشارة $Sh/Ld' = 0$ ، يعني ذلك شحن السجل بالقيمة المطلوب تخزينها الموجودة على مداخل السجل A, B, C, D دفعة واحدة، أما إذا كانت

القيمة $Sh/Ld' = 1$ ، فهذا يعني إزاحة القيمة المخزنة بالسجل، أي إخراج البيانات المخزنة عن طريق المخرج Q_D بت بعد بت والقيمة التي تؤدي للإزاحة تُطبق على المدخل A.



الشكل 10-12: التركيبية الداخلية لسجل من نوع PISO بطول 4 bits

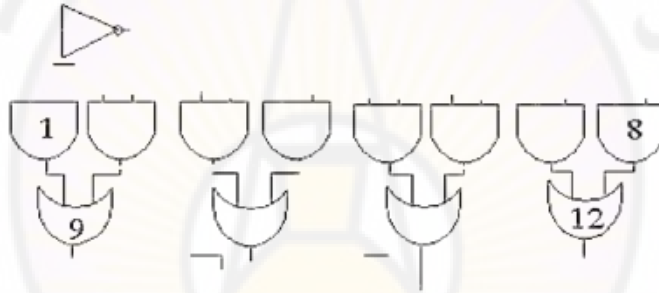
وكمثال من الدارات المتكاملة عن هذا النوع من السجلات، نورد الدارة 74165 ، التي تمثل سجل 8 bit Shift Register PISO. وبالنظر لتعدد خصائص هذه الدارة، نورد في الشكل 12-13 الرمز العام للدارة.



الشكل 10-13: الرمز المنطقي للدارة المتكاملة 74165

10-3 سجلات الإزاحة ذات الاتجاهين

تمتاز سجلات الإزاحة بالاتجاهين Bi-directional shift registers بإمكانية إزاحة البيانات المخزنة فيها من اليمين إلى اليسار وبالعكس، وذلك بالاعتماد على قيمة إشارة للتحكم، تسمى مثل هذه الإشارة الخارجية يمين/يسار، اختصاراً R/L'. الشكل العام لسجل من هذا النوع بطول 4 bits مبني بقلا بات من نوع D وبوجود مثل هذه الإشارة مبين في الشكل 10-14، حيث يمكن تتبع عمل مثل هذه السجلات من خلال المخطط. بعد التصغير وورود أول نبضة ساعية إلى جميع القلابات المشكلة للسجل بوجود معطيات على مدخل المعطيات التسلسلي، يمكن أن تتم واحدة من عمليتين:

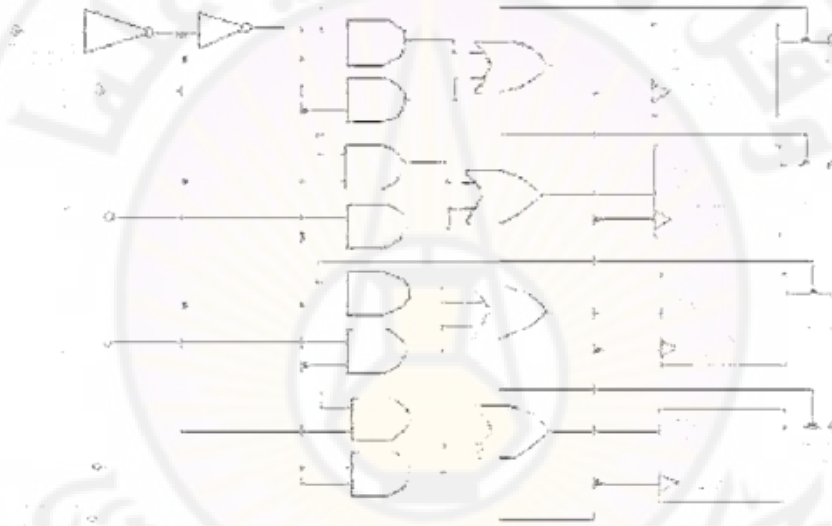


الشكل 10-14: سجل إزاحة بالاتجاهين بطول 4 bits.

الأولى: إذا كان $R/L' = 0$ ، فإن خرج البوابات AND ذات الأرقام 1, 3, 5, 7 سيصبح 0 وبالتالي القيمة الموجودة على مدخل المعطيات التسلسلي ستدخل عبر بوابة (AND8) وبوابة OR12 إلى القلاب D الذي يمثل أدنى خانة LSB. القيمة التي كانت موجودة بالقلاب D ستنتقل عبر بوابة AND6 وبوابة OR11 إلى القلاب C. الثانية: بنفس الأسلوب، إذا كان $R/L' = 1$ ستتم الإزاحة نحو اليمين، أي أن التركيبة العامة للسجل تبقى كما هي من دون تغيير وعن طريق إشارة تحكم خارجية يمكن التحكم باتجاه الإزاحة بحسب القيمة الموجودة عليها.

10-4 السجلات بالتحميل المتوازي

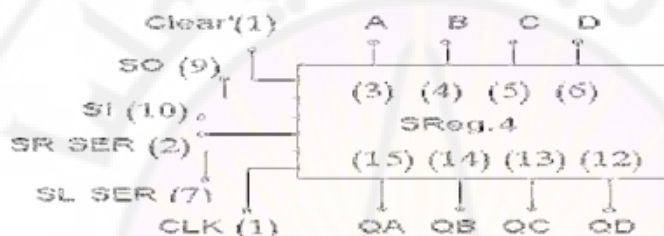
تُعدّ السجلات ذي التحميل المتوازي Register with Parallel Load حالة خاصة من سجلات " الدخّل على التوازي - الخرج على التوازي"، وفيها يوجد إشارة تحكم خارجية بالاسم "تعبئة Load". يسمح هذا النوع من السجلات بتعبئة قيمة ما في السجل دفعة واحدة، ويمكن قراءة أي قيمة مخزنة في السجل بأي لحظة. تبدو التركيبية الداخلية لسجل من هذا النوع بطول 4 bits كما في الشكل 10-15.



الشكل 10-15: التركيبية العامة لسجل بدخّل على التوازي بطول 4 bits

بتفعيل إشارة التعبئة، يمكن شحن السجل بالقيمة المطبقة على المداخل الخارجية دفعة واحدة. يظهر في الشكل منطق مكون من بوابات NOT و AND و OR يؤمن عملية إدخال ما هو موجود على المداخل I_0, I_1, I_2, I_3 إذا كانت إشارة التحميل فعّالة بلحظة ورود النبضة. علماً أنه في حال وجود معطيات على المداخل I_0, I_1, I_2, I_3 وإشارة التحميل غير فعّالة لن يتغير محتوى السجل مع ورود النبضة الساعية، لأن ما هو موجود على مخرج كل قلاب يعود ليدخل من جديد إلى مدخله (حالة ذاكرة). يمكن بناء سجلات قادرة على تنفيذ عدد أكثر من العمليات (إزاحة، تدوير وغيرها) بسجل واحد تسمى بالسجلات

متعددة الوظائف. يطلق على مثل هذه السجلات أيضاً اسم سجلات إزاحة عامة Universal registers، وهي سجلات تمتاز بإمكانية إزاحة البيانات بالاتجاهين، وبإمكانية التحميل على التوازي. يلزم لمثل هذه السجلات جدول يوضح كيفية تنفيذ هذه العمليات التي يقوم بها السجل، يسمى بجدول الوظائف Function table. نورد فيما يلي الدارة المتكاملة 74194 مثالاً على هذا النوع من السجلات، وهي تمثل سجل عام بطول 4 bits. يظهر الشكل 10-16 رمز هذه الدارة مع توزيع إشاراتها.



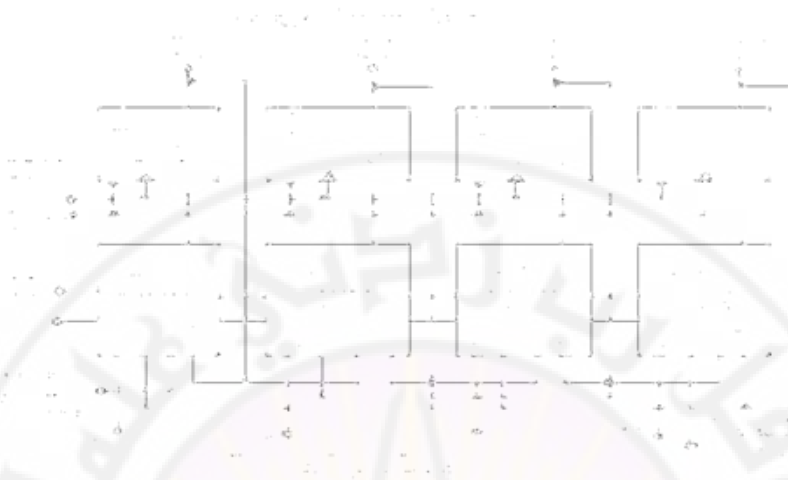
الشكل 10-16: الرمز المنطقي لسجل عام (تحديداً الدارة 74194)

تكون التركيبة الداخلية لمثل هذا السجل، وفق ما هو مبين في الشكل 10-17، انطلاقاً من جدول وظائف الدارة المبين في الجدول 10-3.

الجدول 10-3: جدول الوظائف للسجل 74194

Mode Control		Register Operation
S_1	S_0	عملية السجل
0	0	No Change لا تغيير
0	1	Shift Right إزاحة إلى اليمين
1	0	Shift Left إزاحة إلى اليسار
1	1	Parallel Load تحميل على التوازي

يلزم لمثل هذه السجلات المركبة إضافة إلى القلايات بعض البوابات المنطقية والدارات التركيبية مثل المضمّم لتنفيذ جدول الوظائف المعتمد.



الشكل 10-17: التركيبة الداخلية لسجل متعدد الأغراض بطول 4 bits.

يحقق الشكل أعلاه أربعة وظائف مختلفة، حيث تم استخدام مضمّم بالتركيبة 4×1 لاختيار وظيفة محددة من أجل كل تركيبة على خطوط الانتقاء S_1S_0 . عندما يكون $S_1S_0 = 00$ ، فإن ما هو موجود على المدخل 0 لكل مضمّم، سيظهر على خرجه ويدخل إلى القلاب المقابل، لكن وفق هذه التركيبة بفضل الخط الرابع من خرج القلاب إلى مدخل المضمّم توجد الحالة السابقة للقلاب وبالتالي تصبح الحالة اللاحقة للقلاب هي نفس الحالة السابقة، أي أن وضع القلاب وبالتالي السجل لم يتغير. وبنفس الطريقة تدرس بقية التراكيب المبينة بجدول الوظائف، حيث تم استخدام مدخلين للمعطيات التسلسلية، أحدهما مخصص لإزاحة البيانات المخزنة بالسجل باتجاه اليمين، والآخر باتجاه اليسار.

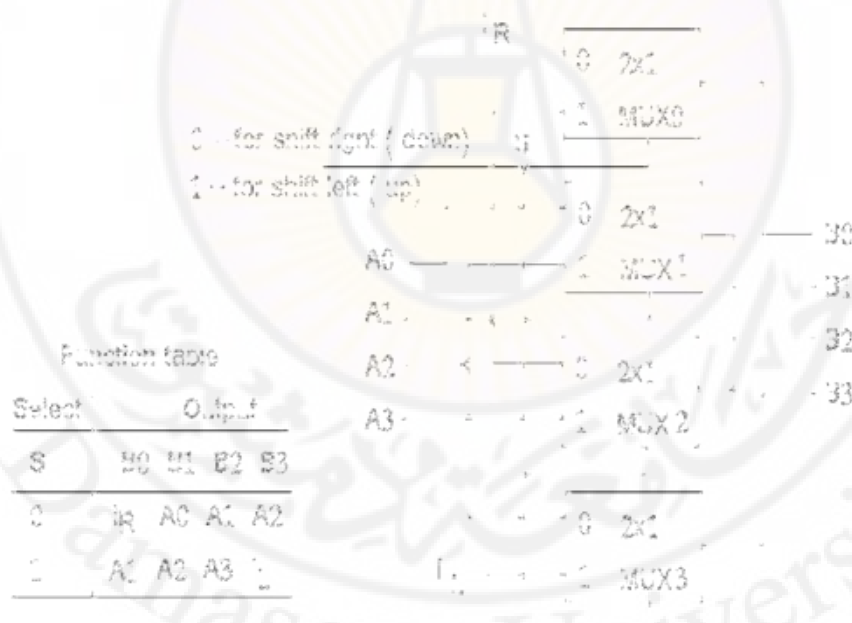
5-10 تطبيقات على استخدام السجلات

يوجد العديد من التطبيقات التي تستخدم السجلات وبخاصة التطبيقات ذات الطابع البرمجي كما سنجد لاحقاً بمقرر المعالجات الصغيرة، انطلاقاً من التقسيمات المتعارف عليها وهي: سجلات بيانات: يتم فيها التعامل مع البيانات للتخزين وإجراء العمليات الحسابية والمنطقية. سجلات العناوين: يتم فيها تخزين العناوين المختلفة. سجل الحالات: ويحتوي على حالة المعالج بعد تنفيذ أمر محدد.

نورد فيما يلي من مجمل التطبيقات الممكنة للسجلات، تطبيقين فقط ذات طابع تقني هما:

الأول- المزيج Shifter

بالرغم من أن السجل يمثل دائرة تتابعية، يمكن النظر إلى المزيج على أنه دائرة تركيبية تتميز بطول الكلمة والوظيفة، وهو يدخل بشكل أساسي في التركيبية الداخلية لوحدة الحساب والمنطق. يفهم المزيج على أنه سجل إزاحة بالاتجاهين وبمداخل على التوازي. يعمل على إدخال البيانات من المداخل المباشرة، ومن ثم وفق بعض إشارات التحكم تراح هذه البيانات باتجاه اليمين أو اليسار بحسب الطلب. وإذا رمز لمدخل المعطيات التسلسلي لسجل من اليمين بالرمز I_R وللمدخل من اليسار بالرمز I_L ، يمكن انطلاقاً من جدول وظائف محدد، رسم المخطط العام لدائرة تركيبية تمثل مزيج بطول 4bits باستخدام المصمّمات، وفق ما هو مبين في الشكل 10-18.



الشكل 10-18: الدارة التركيبية لمزيج بطول 4 bits مع جدول الوظائف المقابل

يمثل المزيج التنفيذ التقني للميكرو عمليات الخاصة بالإزاحة، بحيث عندما يكون خط الانتقاء للمصمّم $S = 0$ يتم انتقاء المدخل بالدليل 0 من المصمّم ليظهر على الخرج.

وهذا يعني إدخال القيمة I_R في أول مضمم من اليسار وإظهارها على المخرج B_0 ، وبالتالي القيمة A_0 تظهر على المخرج B_1 وهكذا دواليك حتى النهاية ويخرج من السجل البت ذو الوزن LSB. أما عملية الإزاحة نحو اليسار فتتم بالعكس، حيث يتم بخانة LSB إدخال ما هو موجود على المدخل التسلسلي وإخراج القيمة الموجودة بخانة MSB. وهنا تجدر الإشارة إلى وجود ثلاثة أنواع من الميكروعمليات الخاصة بالإزاحة، هي:

- الإزاحة المنطقية Logical shift، وتعني الإزاحة الناتجة من إدخال القيمة صفر من المدخل التسلسلي للسجل وإزاحة البيانات المخزنة إما باتجاه اليمين ويرمز لها عادة بالرمز shr، وإما باتجاه اليسار ويرمز لها بالرمز shl.

- الإزاحة الحلقية Circular shift، وتعرف أيضاً بالدورانية Rotate operation ويستخدم الرمز cir للرمز إلى الإزاحة الدائرية باتجاه اليمين والرمز cil للإزاحة الدائرية باتجاه اليسار. لا يوجد بهذا النوع للإزاحة إضاعة للمعلومات المخزنة.

- الإزاحة الحسابية Arithmetic shift وتستخدم لإزاحة الأعداد الثنائية المؤشرة باتجاه اليمين، ويرمز لها بالرمز ashr أو باتجاه اليسار ويرمز لها بالرمز ashل. يجب أن تحافظ عملية الإزاحة الحسابية على إشارة القيمة المزاحة. يتم خلال عملية الإزاحة الحسابية باتجاه اليسار تطبيق صفر على المدخل التسلسلي وإزاحة جميع البتات باتجاه اليسار، كما هو الحال بالإزاحة المنطقية باتجاه اليسار، لكن في الإزاحة الحسابية يمكن أن يحدث بعض الأخطاء تسمى الفيض. يحدث الفيض عندما يكون آخر بتين بالسجل قبل إجراء عملية الإزاحة مختلفين بالقيمة. أما للإزاحة الحسابية للبيانات باتجاه اليمين، ينظر إلى قيمة البت MSB وينسخ كما هو في البت الأدنى وكل بت ينقل بدوره إلى البت الأدنى ويخرج من السجل القيمة التي توجد بالخانة LSB. بالطبع القيمة 0 بخانة الإشارة المقابلة للخانة MSB تعني إشارة موجب والقيمة 1 تقابل الإشارة السالبة. تعني عملية الإزاحة ashل ضرب القيمة المخزنة بالعدد 2، وعملية ashr فتعني قسمة القيمة المخزنة على 2. تستخدم الميكروعمليات لغة تصويرية لتوصيف عمل العداد مثل الرمز ← للإزاحة. يمكن إظهار كافة الأشكال المبينة أعلاه من ميكروعمليات الإزاحة. يعبر عن إزاحة السجل R باتجاه اليسار منطقياً مثلاً بالشكل: $R \leftarrow \text{shl } R$ وهكذا بالنسبة لبقية العمليات.

الثاني - الجامع التسلسلي

يقوم الجامع التسلسلي Serial adder بجمع مكونات أي عددين من البتات الثنائية بشكل تسلسلي. سبق وتناولنا بالفصل التاسع التركيبية العامة للجامع الفرعي ووجدنا أنه لجمع عددين كل منهما بطول n bits ، يلزم n جامع تام، بينما يكفي للجامع التسلسلي استخدام جامع تام واحد مهما يكن طول كل من العددين المراد جمعهما. لتوضيح التركيبية العامة للجامع التسلسلي، نطلق من مثال محدد وفق التالي: بفرض أنه يطلب تصميم دائرة جامع تسلسلي لجمع أي عددين عشرين مثل العددين: $68 + 57$ فيكون الجواب كالتالي: ليكن العدد الأول 68 الذي يقابل القيمة $(100\ 0100)_2$ مخزن بسجل A بطول 8 bits ، والعدد الثاني 57 الذي يقابل القيمة $(11\ 1001)_2$ مخزناً بسجل B، وليكن المجموع الناتج 125 الذي يقابل القيمة الثنائية $(0111\ 1101)_2$ مخزن بسجل ثالث C بطول 8 bits (علماً أنه بالإمكان حفظ الناتج بواحد من السجلين A, B). يلزم عملية الجمع بالشكل التسلسلي لمحاكاة عملية الجمع العادية سجلات إزاحة من النوع " دخل تسلسلي - خرج تسلسلي " لتخزين الأرقام المراد جمعها، بحيث بعد كل عملية جمع لبتين متقابلين تتم الإزاحة إلى البتتين التاليتين وبكل مرة يحسب الحمل، ويخزن بذاكرة مؤقتة (قلاب) لاستخدامه بعملية الجمع اللاحقة. وباستخدام سجلات A, B, C مزودة بإشارات التحكم التالية:

- إشارة خارجية لإدخال المعطيات إلى السجل بشكل تسلسلي تطبق على مأخذ خاص بالاسم SI (Serial In).

- إشارة لإزاحة البيانات المدخلة إلى اليمين بالاسم Shift Right (SR) .

- إشارة تزامن تطبق على مأخذ خاص بالاسم CP (Clock Pulse).

بفرض أن الخرج التسلسلي SO من السجل A هو a ومن السجل B هو b، يمكن رسم الشكل العام للدائرة المطلوبة، وفق ما هو مبين في الشكل 19-12. تتكون الدائرة بالإضافة إلى السجلات من دائرة جامع تام واحد وقلاب واحد يمثل ذاكرة بطول 1 bit لحفظ قيمة النقل التي يمكن أن تنتج من جمع أي بتين متقابلين من العددين المراد جمعهما.

الشكل 10-19: التركيبة العامة للجامع التسلسلي

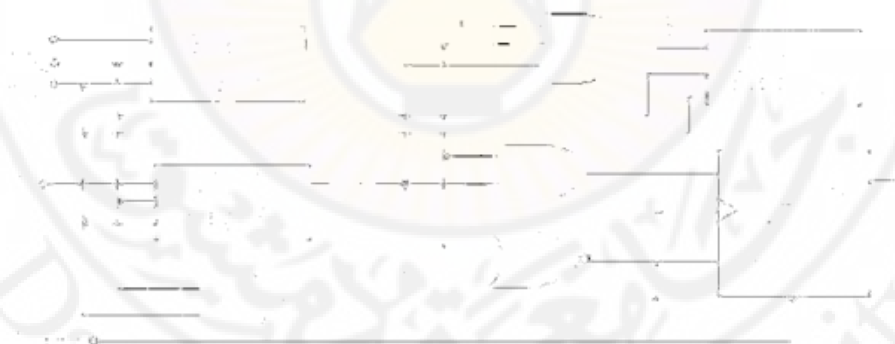
يتلخص عمل الدارة في الشكل أعلاه بالتالي: بعد الانتهاء من إدخال كل من العددين إلى السجلات باستخدام المداخل SI بوساطة عدد من نبضات الساعة، يساوي طول العدد الأطول منهما، تبدأ عملية الجمع بتطبيق إشارة SR مع ورود الوجه الصاعد للنبضة الساعية CP على كل من السجلين A, B بأن واحد، فتتشكل النبضة الساعية للقلاب المستخدم الذي يوجد على مدخله قيمة النقل من المرحلة السابقة (القيمة البدائية 0). يخرج من كل سجل أدنى بت ومن القلاب يخرج النقل السابق. تجمع هذه القيم مع بعضها لتشكيل المجموع S الذي يعطى بالعلاقة: $S = A \oplus B \oplus C_{in}$ الذي يخزن بالبت المقابل بالسجل C ويحسب C_{out} وفق العلاقة: $C_{out} = (A \oplus B) C_{in} + A.B$ الذي يخزن بالقلاب للمرحلة التالية. وهكذا يستمر العمل إلى حين الانتهاء من جمع كل البتات المتقابلة بالعددين بانتهاء تأهيل إشارات التحكم بالإزاحة SR الذي يتم بعد أخذ كافة البتات بكل من العددين. بعد أن تم رسم دارة الجامع التسلسلي بالطريقة المباشرة، بناء على الوصف الكلامي، نعود لتصميم نفس الدارة وفق الإجراءات المتبعة بتصميم الدارات التتابعية التي مرت معنا، انطلاقاً من جدول الحالة وفق التالي: بالاستفادة من الرموز المستخدمة أعلاه بالنسبة للجامع التسلسلي وباستخدام قلاب J-K يمكن بناء جدول الحالة للجامع المطلوب، وفق ما هو مبين في الجدول 10-4. تصبح عملية التصميم منحصرة باستخراج المعادلات الأساسية للدارة:

$$J = a . b \quad ; \quad K = a'b' = (a + b)' \quad ; \quad S = a \oplus b \oplus q$$

الجدول 4-10: جدول الحالة للجامع التسلسلي بوجود قلايات من نوع JK

الحالة السابقة	المدخل		الحالة اللاحقة	الخرج	مداخل القلايات	
	Inputs		Next State	Output	Flip-Flop Input	
q	a	b	Q	S	J	K
0	0	0	0	0	0	d
0	0	1	0	1	0	d
0	1	0	0	1	0	d
0	1	1	1	0	1	d
1	0	0	0	1	d	1
1	0	1	1	0	d	0
1	1	0	1	0	d	0
1	1	1	1	1	d	0

يرسم هذه المعادلات نحصل على المخطط المنطقي المبين في الشكل 20-10 الذي يقابل المخطط المبين في الشكل 19-10 بفارق أن المخطط في الشكل 20-10 يظهر التركيب التفصيلية للجامع التام المستخدم.



الشكل 20-10: دائرة الجامع التسلسلي

يلاحظ بمقارنة الجامع التسلسلي مع الجامع التفرعي أن الجامع التفرعي أسرع بكثير (ولكن مقابل هذه السرعة يلزم تجهيزات أكثر). علاوة على أن الجامع التفرعي يستخدم سجلات PIPO وعدد من دوائر الجامع التام يساوي عدد البتات في الأعداد المطلوب جمعها، بينما الجامع التسلسلي يستخدم سجلات SISO، ويكفيه جامع تام واحد. ويمكن النظر إلى الجامع

التفرعي على أنه دائرة تركيبية، بينما الجامع التسلسلي يمثل دائرة تتابعية، لأن النتيجة النهائية بالعمليات التسلسلية تعتمد على القيم السابقة. يمكن أن نصادف تطبيقات تقنية أخرى على استخدام السجلات كتصميم وحدة الممرات المشتركة Common bus وغيرها من التطبيقات الهامة.

الجدول 10-2 : أرقام بعض الدارات المتكاملة التي تمثل أنواعا مختلفة من السجلات

74 91	8-bits Shift Register SISO	DIL – 14
74 94	4-bits Shift Register PISO	DIL – 16
74 95	4-bits Shift Register PIPO	DIL – 14
7496	5 bit Shift Register (PIPO)	DIL – 16
7499	4 bit Right – Shift, Left – Shift Reg.	DIL – 16
74164	8-bits Shift Register SIPO	DIL – 14
74165	8-bits Shift Register PISO	DIL – 16
74173	Quad 3-State Latch	DIL – 16
74194	4-bits Shift Register PIPO (universal)	DIL – 16
74198	8-bits Shift Register PIPO	DIL – 24
74241	Octal Buffer 3-State	DIL – 20
74244	Octal 3-State Driver	DIL – 20
74245	Octal Bus Transceiver	DIL – 20
74259	8 bit Addressable Latch	DIL – 16
74291	8-bit Bi-Directional Counter Shift Reg.	DIL – 20
74295	4-bit Right Shift Left Shift 3-State Reg.	DIL – 14
74322	8 bit S / P Register	DIL – 20
74388	4-bit Register Standard & 3-State out	DIL – 16
74396	8-bit Storage Register	DIL – 16

10-6 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من تساؤلات التالية:

- 1- تتمثل وظيفة السجل في
a- Storage data b- Shift data c- Rotate data d- All the above
- 2- الخاصية الأساسية لأي سجل هي:
a- function of register b- word length of register
c- structure of register d- all the above
- 3- بالسجل ثنائي الاتجاه، يمكن للمعطيات أن تزاح
a- right or left b- up or down
c- only from right to left d- only from left to right
- 4- عملية الضرب باستخدام المسجلات، تعني
a- Shift to left b- Shift to right
c- Rotate to left d- Rotate to right
- 5- يمثل الموديم سجل لتحويل البيانات من
a- Serial to parallel b- Parallel to serial
c- Serial to serial d- Parallel to parallel

ثانياً: أسئلة متفرقة

- 1- ليكن على المدخل التسلسلي لسجل بطول 4 bits القيمة الثنائية (11010)، يطلب إزاحة محتوى السجل خمس مرات إلى اليمين، وإظهار محتوى السجل المستخدم بعد كل عملية إزاحة؟
- 2- صمم باستخدام القلايات D سجل إزاحة بطول 5 bits وبمدخل تحميل على التوازي، بحيث تنقل المعطيات الجديدة إلى داخل السجل عندما يكون مدخل التحميل Load = 1 ومدخل الإزاحة Shift = 0.
- 3- صمم دائرة طارح تسلسلي (المخطط المنطقي) لطرح أي عشرين ضمن المجال 00 حتى 99.

الفصل الحادي عشر: العدادات Counters

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

- إعطاء عدد النبضات المطبقة على الدخل.
- القسمة، حيث يستطيع العدّاد إعطاء نبضات في الخرج بتعدد أصغر من تردد إشارة الدخل، بالتالي العدّاد يعمل مقسماً للتردد Frequency divider. ويمكن تمثيل أشكال استخدام العدادات بالرموز التوضيحية المبينة في الشكل 11-1.

؛

؛ - ؛

الشكل 11-1: الرموز العامة للعدّاد

يكون دخل العدّاد دوماً نبضات ساعية، في الوقت الذي يكون فيه الخرج إما قيمة مرمزة كما في الشكل 11-1 أ، أو نبضة بتعدد معين كما في الشكل 11-1 ب. نميز بحسب طريقة وصل الساعة إلى القلايات المكونة للعدّاد بين نوعين رئيسيين من العدّادات هما:

- العدادات غير المتزامنة Asynchronous Counter، أو غير المتوافقة، والاسم الشائع لها هو العدادات التموجية (Ripple Counters عدادات ريبيل).

- العدادات المتزامنة Synchronous Counter، أو العدادات المتوافقة.

تستخدم العدادات بنوعيهما في النظم الرقمية لتعداد عدد حدوث حدث ما، ولتوليد إشارات توقيت Timing Signals للتحكم بتتابع عمليات محددة في هذه الأنظمة. يمكن أن يكون تتابع حالات العداد عدداً ثنائياً، فينتج في هذه الحالة عداد ثنائي، ويمكن أن يكون أي تتابع عددي آخر غير الثنائي. ولفهم آلية العد باستخدام العدادات الثنائية مثلاً، نفترض أنه 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 2-11 أ، أي سيغير القلاب حالته (إلى الحالة المعاكسة) مع كل ورود جديد للنبضة الساعية. يتتبع عمل هذه الدارة من خلال المخطط الزمني المبين في الشكل 2-11 ب، يلاحظ أنه بعد تفعيل المدخل Clear سيصبح محتوى القلاب $Q = 0$.

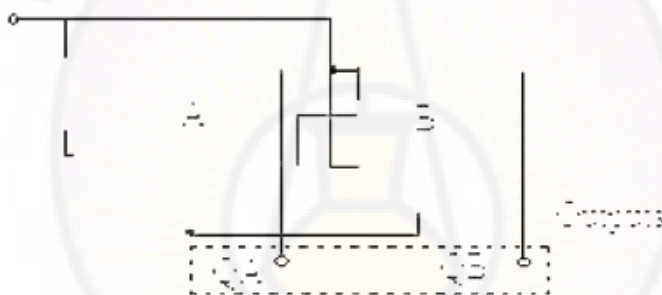
: : : - :

الشكل 2-11: عداد ثنائي يعد قيمتين

مع ورود النبضة الساعية رقم 1 سيقبّل القلاب أو يغير حالته من 0 إلى 1 ويبقى بهذه الوضعية إلى حين ورود النبضة الساعية رقم 2 ، حيث يعود فور ورود النبضة الثانية من جديد ويغير حالته من 1 إلى 0 وهكذا دواليك، أي أن للقلاب إمكانية التعداد، حيث يستطيع القلاب الواحد العد إلى اثنين لأن الخرج Q يصبح بالحالة 1 بعد كل عدد مفرد من نبضات المولد. لزيادة إمكانية التعداد بالعداد المشكل من القلابات، يجب زيادة عدد القلابات الداخلة

بتركيبته، حيث من أجل عدّاد مشكل من قلابين مثلاً، يمكن العدّ حتى أربعة، لأن كل قلاب يستطيع أن يعد قيمتين، ولكن كل قيمة للقلاب الثاني تعادل قيمتين من القلاب الأول، أي أن الناتج يصبح $2 \times 2 = 4$. وهكذا من أجل عدّاد بـ n قلاب يمكن عدّ حتى 2^n قيمة. نميز بين شكلين مختلفين لربط القلابات ضمن العدّاد الواحد هما:

الأول- الربط غير المتزامن للقلابات: وفيه يتم، كما هو مبين في الشكل 3-11 لعدّاد مكون من قلابين مثلاً، وصل الساعة من مصدر الساعة الخارجية إلى أول قلاب فقط والساعة اللازمة لكل من القلابات الأخرى ضمن العدّاد تؤخذ من مخرج القلاب الذي يسبقه، بحيث كل قلاب يغير حالته بمعزل عن الآخر. يسمى العدّاد الناتج من طريقة الربط هذه بالعدّاد غير المتزامن (اللاتزامني).



الشكل 3-11: دائرة عدّاد لأربع قيم - غير متزامن

تعمل القلابات بالعدّاد غير المتزامن بصيغة القلب، لذا توصل مداخل القلابات A و B المشكلة للعدّاد بالمستوى H. ونحصل على الخرج النهائي من Q_A و Q_B بشكل ثنائي وفق الترتيب $Q_B Q_A$. ويظهر المخطط الزمني للعدّاد المبين في الشكل 4-11 كيف ينتقل العدّاد بالتسلسل بين أربع حالات ابتداء من 0 حتى القيمة 3، حيث بعد تصفير كل من القلابين المشكّلين للعدّاد A و B يصبح محتوى (خرج) كل من القلابين $Q_A = Q_B = 0$. مع ورود الوجه الصاعد لأول نبضة ساعية (النبضة رقم 1) إلى القلاب A يغير القلاب حالته إلى الواحد، ويصبح خرج $Q_A = 1$ ، ويبقى بهذه الوضعية إلى حين ورود النبضة رقم 2، فيغير حالته من جديد، وينتقل للحالة 0 مع ورود الوجه الصاعد للنبضة الثانية. ومع ورود النبضة رقم 3 ينتقل القلاب A من جديد للحالة 1، وهكذا باستمرار يغير حالته مع كل نبضة إلى

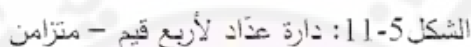
عكس الحالة التي يكون فيها.



الشكل 4-11: المخطط الزمني لعداد غير متزامن يعدّ أربع قيم

لمتابعة خرج القلاب الثاني B ننتقل من الخرج المتم للقلاب A الذي تم الحصول عليه بالمرحلة السابقة، ونعدّه الساعة الجديدة للقلاب B. فنجد بنفس الأسلوب السابق أن خرج القلاب B مكون من نبضة واحدة تقابل نبضتي ساعة من Q'_A ، وهذه النبضة على خرج B تقابل النبضات الأربعة المطبقة على الدخل الرئيسي، مما يدل على أن الدارة قادرة على عدّ أربع قيم، وفق ما هو مرمز أسفل المخطط الزمني. يؤخذ الخرج النهائي للعداد من المخارج Q (أو من المخارج المتممة Q') لكل قلاب على حدة بترتيب محدّد، بحيث إن خرج القلاب الذي يوصل بالساعة الرئيسية ويغير وضعه مع كل نبضة ساعية يقابل قيمة الـ LSB من الناتج (أي أن العداد يبدأ من الحالة $Q_B Q_A = 00$ ، ويمر بعدة حالات ويعود بالنهاية للحالة 00 وفق الترتيب التالي: $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ ، يستمر العداد بالعدّ بهذا الشكل دواراً نياً مادام وجود للنبضات الساعية على القلاب الأول.

الثاني - الربط التزامني للقلابات: وفيه يتم بالنسبة لعداد مكون من قلابين مثلاً، كما هو مبين في الشكل 5-11 وصل الساعة من المصدر الرئيسي إلى جميع القلابات المشكلة للعداد بأن واحد، بحيث مع كل ورود للنبضة الساعية تغيّر كلّ القلابات محتواها كل بحسب القيمة الموجودة على مدخله.



الشكل 6-11: المخطط الزمني لعداد متزامن يعد أربع قيم

329

بعد مرور النبضة الأولى يصبح $Q_A = 1$ وبالتالي مدخل القلاب B يصبح واحداً أيضاً (حالة القلاب). فور ورود الوجه الصاعد للنبضة رقم 2 يقلب كل من A و B، لكن القلاب A يقلب للصفر، بينما القلاب B يقلب للواحد. ويستمر الوضع إلى حين ورود النبضة رقم 3. حيث يلاحظ، أن القلاب A يقلب كالعادة، بينما B يستمر بالحالة التي هو فيها لوجود قيمة 0 على مدخله من المرحلة السابقة. وأخيراً مع النبضة رقم 4 يقلب القلابين معاً. يصل العداد إلى نهاية مجال العد، وفق ما هو مرمز بأسفل المخطط الزمني. يمكن تمثيل العدد المشار إليه أعلاه كدائرة تناظرية بمخطط الحالة وجدول العبور المبين بالشكل 7-11.



الشكل 7-11: مخطط الحالة وجدول العبور لعداد بأربع حالات

11-1 العدادات غير المتزامنة Asynchronous Counters

لا يعني مفهوم اللاتزامني بالنسبة للعدادات عدم وجود نبضات ساعية، لأنه بكل النوعين للعدادات (المتزامنة وغير المتزامنة) توجد النبضات الساعية. تسمى مجموعة من الأحداث بأنها متزامنة إذا وقعت في نفس الوقت. ويقال إنها غير متزامنة إذا حدثت في أوقات مختلفة. مفهوم التزامن أو عدم التزامن بالعدادات مرتبط بأوقات تغيير الحالة في القلابات المكونة للعداد، لذا يسمى العداد غير متزامن إذا كانت القلابات المكونة له تبدل حالتها في أوقات مختلفة على عكس العداد المتزامن الذي فيه جميع القلابات المكونة للعداد تبدل حالتها بنفس الوقت. بالتالي الساعة الرئيسية (من مولد النبضات الساعية) توصل عادة في العدادات غير المتزامنة إلى القلاب الأول الذي يقابل الخانة LSB، بينما الساعة اللازمة لكل قلاب من مكونات العداد توصل من خرج القلاب الذي يسبقه وذلك بحسب نوع العداد إن كان سيعد بشكل تصاعدي أو تنازلي.

11-1-1 العدادات غير المتزامنة التصاعدية

يقال عن العداد غير المتزامن المبين في الشكل 11-3 بأنه تصاعدي لأنه يبدأ العد من القيمة 0 وصولاً إلى القيمة 3 ثم يعود من جديد. ويكرر دورة العد بشكل حلقي إلى حين إيقافه بتوقيف النبضات الساعية القادمة على المدخل CP. يعد العداد بشكل تصاعدي إذا توافر عدة شروط مجتمعة في تركيبته الداخلية مثل:

- اعتماد قلابات تستجيب للوجه الصاعد للنبضة.
 - أخذ الخرج النهائي للعداد من المخارج Q للقلابات وليس من Q'.
 - وصل ساعة كل قلاب (باستثناء الأول) من Q' للقلاب الذي يسبقه.
- وللتوضيح ننطلق من مثال آخر حول تركيبية عداد يعد مثلاً حتى ثماني قيم. وجدنا في المثال السابق المبين في الشكل 11-3 أنه بقلابين يمكن تشكيل عداد يعد أربع قيم. لو أضفنا قلاباً ثالثاً بنفس أسلوب التوصيل لتركيبية العداد أعلاه، حصلنا على عداد يعد حتى $2^3 = 8$. يكون الشكل العام لمثل هذا العداد غير المتزامن التصاعدي، كما هو مبين في الشكل 11-8.



الشكل 11-8: التركيبية الداخلية لعداد غير متزامن تصاعدي يعد ثماني قيم

لفهم آلية عمل العداد بشكل أفضل نلجأ عادة إلى المخططات الزمنية المقابلة له. فالمخطط الزمني للعداد المبين في الشكل 11-8، يبدو كما هو في الشكل 11-9. يتضح من المخطط الزمني أن القلابات المكونة للعداد تستجيب للوجه الصاعد للنبضة الساعية. يلاحظ من الخرج النهائي للعداد المعطى على المآخذ $Q_C Q_B Q_A$ أن القلاب الأول A يغير حالته بعد

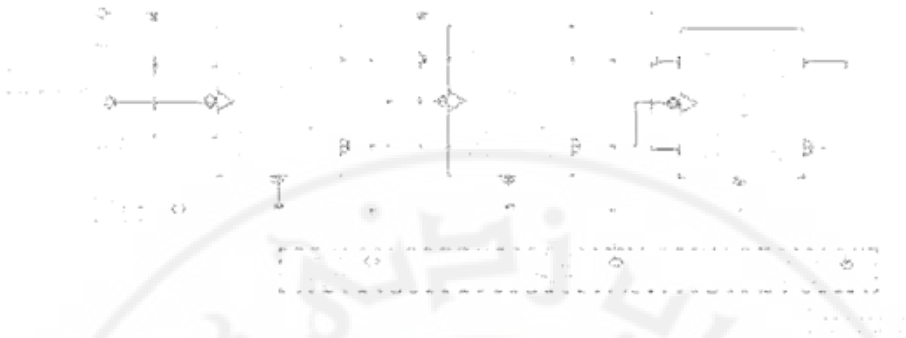
ورود كل نبضتين من الساعة على المدخل CP. يبدأ القلاب B بتغير حالته بعد كل انتهاء للقلاب A من المرور بالحالات التي يمكن أن يمر بها، أي بعد كل نبضتين من خرج A وهذا ما يعادل أربع نبضات من الدخل الرئيسي. وأخيراً القلاب C لا يباشر بالقلب إلا بعد انتهاء كل من القلابين B و A من القلب، حيث يلاحظ على خرج القلاب الأخير Q_C بالعداد نبضة واحدة مقابل ثمان نبضات على الدخل، أي أنه يعد ثمان قيم.

الشكل 9-11: المخطط الزمني لعداد غير متزامن تصاعدي (يعد حتى ثمانية قيم)

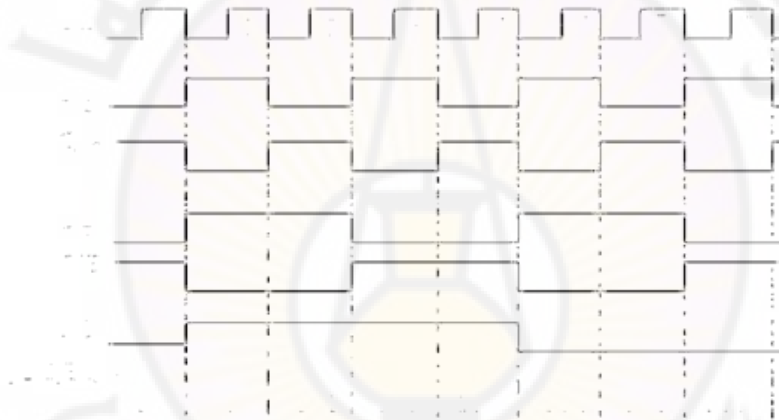
بترميز الخرج بشكل ثنائي، يلاحظ أن العداد يعد بشكل تصاعدي ابتداء من القيمة 0 التي تقابل التركيبة الثنائية 000 وصولاً إلى القيمة 7 المقابلة للتركيبة 111.

11-1-2 العدادات غير المتزامنة التنازلية

بالعودة إلى الشروط الثلاثة المبينة أعلاه للعداد التصاعدي، يلاحظ أنه إذا أُخلى بواحد من هذه الشروط يصبح العداد تنازلياً. فمثلاً، في الشكل 8-11 الذي يمثل عداداً تصاعدياً غير متزامن، إذا استبدلنا القلابات المستخدمة بالعداد بقلابات أخرى تستجيب للوجه الهابط للنبضة، كما هو مبين في الشكل 10-11. وأبقينا الشرطين الآخرين على حالهما، يتم الحصول على عداد يعد ثمان قيم لكن بشكل تنازلي، أي من $(111)_2 = (7)_{10}$ حتى 000، وفق ما هو مبين في المخطط الزمني في الشكل 11-11.



الشكل 10-11: التركيبة الداخلية لعداد غير متزامن تنازلي يعد ثماني قيم



الشكل 11-11: المخطط الزمني لعداد غير متزامن تنازلي (يعد حتى ثماني قيم)

باعتقاد الخرج النهائي للعداد من Q' بدلاً من Q لكل قلاب على حدة، والمحافظة على الشرطين الآخرين يمكن الحصول على نفس النتيجة وهي عداد غير متزامن تنازلي، حيث بدلاً من البداية بـ 000 في العداد التصاعدي ستتم البداية بالحالة 111 بالعداد التنازلي وهي متمم الحالة 000. تغيير الشرط الثالث بأخذ الساعة لكل قلاب من Q القلاب الذي يسبقه (عوضاً عن Q') مع المحافظة على الشرطين الآخرين يؤدي كما هو واضح من المخططات الزمنية المبينة أعلاه إلى نفس النتيجة أيضاً. بالطبع إذا أخلينا بشرطين من الشروط الثلاثة المبينة أعلاه بأن واحد يصبح العداد تصاعدياً.

11-1-3 العدادات غير المتزامنة التصاعدية - التنازلية

يمكن تصميم عدادات تعدّ بالاتجاهين، أي بشكل تصاعدي وتنازلي. لتصميم مثل هذه العدادات الأكثر استخداماً يلزم لدارة العداد الأساسية إضافة منطق معين مع إشارة بمفهومين مختلفين أو إشارتين مستقلتين هما:

الأولى: للأعلى UP لجعل العداد يعدّ بشكل تصاعدي عند تفعيلها.

والثانية: للأسفل Down لجعل العداد يعدّ بشكل تنازلي عند تفعيل هذه الإشارة .

لتوضيح التركيبة العامة لمثل هذه العدادات ننتقل من دون الدخول بمزيد من التفاصيل من مثال محدد وفق التالي:

مثال: صمم دارة عداد غير متزامن تصاعدي- تنازلي يعدّ حتى 8 قيم.

الجواب: يلزم للتركيبة الأساسية للعداد ثلاثة قلابات، لتكون من النوع T التي تستجيب للوجه الهابط للنبضة مثلاً. ومن المعايير الأساسية لتحديد نوعية العداد هو تحديد ساعة كل قلاب بالعداد إن كانت موصولة بالخرج Q أو Q' للقلاب الذي يسبقه. وبالتالي لتشكيل عدادين بتركيبة واحدة يستخدم إشارتين إضافيتين بالاسم UP للعدّ تصاعدياً وعكسها Down للعدّ تنازلياً مع منطق بسيط لتشكيل مدخل الساعة المناسب، وفق ما هو مبين في الشكل 11-12.



الشكل 11-12: التركيبة الداخلية لعداد غير متزامن يعدّ ثماني قيم

بشكل تصاعدي تنازلي معاً

يتضح من الشكل أعلاه، إذا كانت إشارة التحكم $UP/Down' = 0$ ، يعني ذلك أن الإشارة الفعالة هي Down وبالتالي العداد يجب أن يعدّ بشكل تنازلي لوجود خلل بالشرط المتعلق بشكل الاستجابة للساعة (وجه هابط). هذا الشرط محقق، لأن ساعة كل قلاب أصبحت مأخوذة من Q' وخرج العداد مأخوذ من مخارج الفلابات Q بالترتيب $Q_C Q_B Q_A$ مما يجعل العداد الناتج تنازلي. أما إذا كانت إشارة التحكم $UP/Down' = 1$ ، يعني ذلك أن الإشارة الفعالة هي UP أي تصاعدي، وهذا محقق أيضاً، لأن ساعة كل قلاب أصبحت مأخوذة من الخرج Q (حيث اختلا شرطان).

11-1-4 العدادات مقاس N

يطلق ببعض المراجع العلمية على عدد الحالات التي يستقر بها العداد اسم Modulus (اختصاراً M)، ويقصد بها عدد القيم المراد عدّها بالعداد، وهي دوماً أعلى بواحد من أكبر قيمة يمكن أن يظهرها العداد . بمعنى على سبيل المثال أن الـ Modulus بالنسبة للعداد الذي يعدّ ثماني قيم من 0 حتى 7 هو $M = 8$. يقصد بالعداد القياسي (سواء أكان متزامناً أم غير متزامن) كل عداد يعدّ لقيمة من مرتبة 2^n من أجل $n = 1, 2, 3, \dots$ (حيث n تمثل عدد قلابات العداد). بمعنى أنه يمكن ببساطة تصميم عداد يعدّ حتى 16 قيمة وذلك باستخدام أربعة قلابات ووصلها مع بعضها بأسلوب مماثل لما مر سابقاً. لكن ببعض التطبيقات العملية، قد يلزم عدادات لا تعدّ إلى قيمة قياسية من رتبة 2^n ، مثل العداد العشري الذي يعدّ عشر قيم فقط من 0 حتى 9. وتسمى مثل هذه العدادات بالعدادات مقاس N (Modified modulus counters) أو العدادات غير القياسية، حيث N هي قيمة الـ Modulus. فمثلاً، العداد مقاس 5 هو عداد يعدّ خمس قيم، ويفترض أن يبدأ العدّ من القيمة 0 حتى القيمة 4 وبعد ذلك يعود للصفر من جديد. وبما أنه لا يمكن الحصول على القيمة 5 من الصيغة 2^n ، نأخذ أقرب عدد مناسب للقلابات اللازمة لتشكيل العداد المطلوب. ففي مثال عداد لخمس قيم لا يكفي استخدام قلابين، لأن القلابين يستطيعان العدّ حتى 3 فقط (أي أربع قيم)، وبالتالي يلزم لتشكيل العداد المطلوب ثلاثة قلابات، لكن ثلاثة قلابات تسمح بعدّ ثماني قيم ، يلزم منها خمس قيم فقط، بهذه الحالة يشكل عداداً بثلاثة قلابات،

يرمز لها بالرموز A, B, C، توصل مع بعضها بالطريقة المعروفة لتشكيل العداد. وبمنطق خارجي محدد يمكن جعل العداد الناتج يعود للقيمة صفر مجرد وصوله بالعد إلى الرقم 4. لتصميم مثل هذا العداد ننطلق من جدول الحالة المبين في الجدول 11-1 والمراحل المبينة بالفقرة التالية.

الجدول 11-1: جدول الحالة لعداد خمس قيم من أصل ثماني



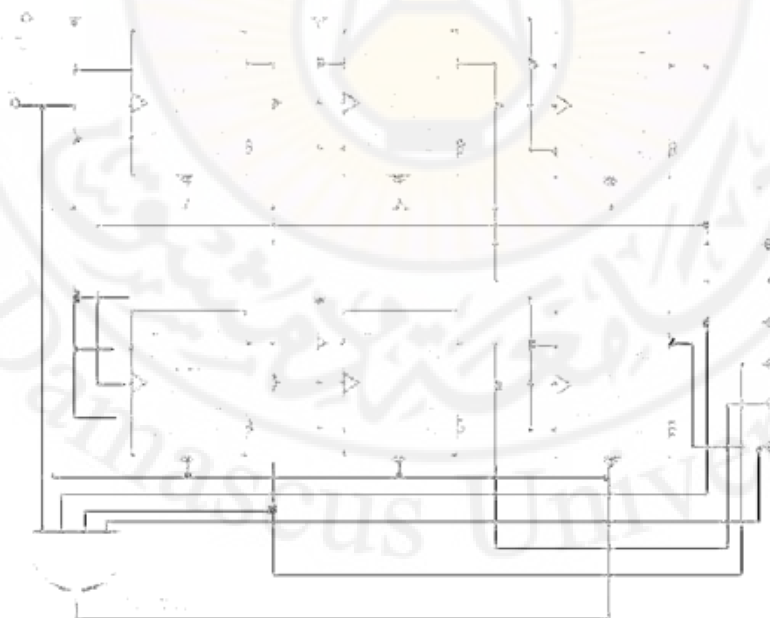
11-1-5 مراحل تصميم العدادات غير المتزامنة

يلاحظ من الفقرة السابقة أن العدادات غير القياسية تخضع لشروط خاصة أثناء التصميم. نستعرض فيما يلي أهم الخطوات المفترض إتباعها خلال تصميم عداد غير متزامن قياسي أو غير قياسي، بغض النظر عن نوعه إن كان تصاعدياً أو تنازلياً وعدد حالاته من خلال مثال يصلح للتعميم على تصميم أي عداد آخر.

مثال: يصادف بالقرب من بعض الإشارات الضوئية على الطرق العامة في المدينة لوحة إظهار، تظهر عليها قيمة ما عند إضاءة إشارة من الإشارات الضوئية كالحمراء مثلاً (أو الإشارة الخضراء)، وهذه القيمة التي تظهر تأخذ بالتناقص بشكل منتظم، بحيث عندما تصل إلى الصفر تضاء الإشارة المعاكسة. سوف لن نهتم حالياً بهذا المثال بآلية إظهار الأرقام وتحويلها للصيغة العشرية ولا بتركيبة دارة التحويل بين الإشارات (عبر الإشارة الصفراء). وسيتم التركيز فقط على آلية توليد مثل هذا العدد وكيفية تناقصه (أو تزايد) لتشغيل إشارة عن غيرها. ليكن المطلوب إذاً تصميم دارة تسمح من لحظة إضاءة الإشارة الخضراء الاستمرار بالإضاءة لمدة ثلاثة أرباع الدقيقة، ولنترك الآن الفترة المطلوبة لمدة إضاءة

الإشارة الحمراء (التي قد تكون أكثر أو أقل). يلزم إذاً، تصميم دائرة عدّاد يعدّ 45 قيمة بشكل تنازلي من 44 حتى 0. والخطوات الأساسية لتصميم مثل هذا العدّاد هي:

1- تحديد عدد القلايات اللازمة لبناء العدّاد من العلاقة $n = \text{INT} [\log_2 M]$ ، حيث M - تمثل عدد حالات العدّاد التي يمكن أن يمر بها العدّاد، وهي تساوي في هذا المثال القيمة 45. و $n -$ عدد القلايات اللازمة لتشكيل العدّاد. التابع INT - يعني أصغر رقم صحيح أكبر، أو يساوي القيمة الناتجة عن حساب $\log_2 M$ ، حيث على سبيل المثال $\text{INT} (5, 322) = 6$. ومن دون الخوض في هذه المعادلة، يتضح أنه يلزم بهذا المثال 6 قلايات لتشكيل العدّاد المطلوب، لأن الرقم 45 يقع بين الرقمين القياسيين 32 (الذي يقابله خمس قلايات) و 64 (الذي يقابله 6 قلايات)، أي أن رقم 6 هو أقرب رقم يكفي من القلايات. لنسم هذه القلايات بالأسماء $F_A, F_B, F_C, F_D, F_E, F_F$ على أن يقابل القلاب F_A أدنى خانة بالناتج أي الـ LSB والقلاب F_F الخانة MSB، كما هو مبين بالتركيبة العامة للعدّاد المطلوب في الشكل 11-13.



الشكل 11-13: التركيبة الداخلية لعدّاد غير متزامن تنازلي يعدّ 45 قيمة

2- تحديد نوع القلابات التي ستستخدم بتشكيل العدّاد وطريقة توصيلها مع بعضها بحسب نوع العدّاد المطلوب. في هذا المثال حول تصميم عدّاد تنازلي، ستم استخدام

- قلابات J-K. تستجيب للوجه الصاعد للنبضة ونوصلها بصيغة القلب.

- اعتماد الخرج النهائي للعدّاد من المخارج Q للقلابات المشكلة له وليس من Q'

- وصل ساعة كل قلاب بالمخرج Q للقلاب الذي يسبقه.

أي يوجد بالشروط الثلاثة للعدّاد التصاعدي خلل، هو الشرط الثالث. والعدّاد بهذه التركيبة والوصل القياسي للقلابات المشكلة له سيعد $2^6 = 64$ قيمة بشكل تنازلي، بالتالي لجعل العدّاد يتوقف عن العدّ عند وصوله للصفر والعودة من جديد (بعد تأخير زمني نضعه بحسب الطلب) للمباشرة بالعدّ من القيمة 44 حتى الصفر، نمثل ثنائياً المقدار (M-1) لتحديد أماكن توضع الوحدات فيه. وبهذا التطبيق $2(101100) = 44 = 45 - 1 = M - 1$ ، يوجد ثلاث وحدات بالخانات الثالثة والرابعة والسادسة (من اليمين).

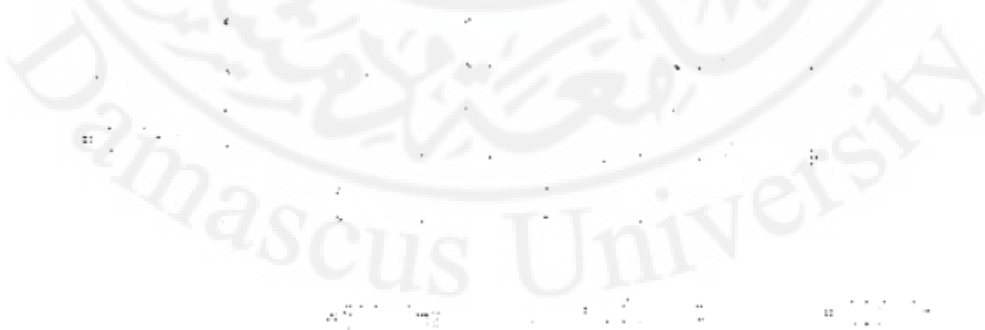
3- ننظر بالخطوة الأخيرة للتصميم إلى المستوى التي تكون فيه إشارة التصفير Reset فعالة لكل قلاب مستخدم بتركيبة العدّاد.

- فإذا كانت هذه الإشارة فعالة بالصفر، نصل خرج كل قلاب يقابل الخانة التي فيها القيمة واحد بالمقدار (M-1) مع إشارة الساعة إلى مداخل بوابة NAND، ونصل خرج هذه البوابة إلى إشارة Reset لجميع القلابات.

- وإذا كانت إشارة التصفير للقلابات المستخدمة فعالة بالواحد نصل خرج كل قلاب يقابل الخانة التي فيها القيمة واحد بالمقدار (M-1)، مع إشارة الساعة إلى مداخل بوابة AND، ونصل خرج هذه البوابة إلى إشارة Reset.

يمكن دمج خطوات التصميم المبينة أعلاه مع بعضها بمخطط واحد للعدّاد المطلوب، ويمكن التأكد من آلية عمل العدّاد باستخدام المخطط الزمني العائد له أيضاً. لكن بسبب كبر المخطط الذي يفترض به أن يظير 45 نبضة ساعية، يتعذر رسمه هنا. يضم جدول الحالات العائد للعدّاد 45 تركيبة من أصل 64 تركيبة مختلفة، حيث يوجد تركيبة واحدة تقابل العدد 44 وهي التركيبة 101100 التي تعني أن خرج كل من القلابات C, D, F سيكون بحالة الواحد وخرج بقية القلابات بحالة الصفر. ومجرد ورود النبضة الساعية عندما

يصل العدّاد لهذه التركيبة، سيصبح خرج البوابة NAND صفراً. وبهذا الشكل نضمّن أن العدّاد لن يصل إلى أي قيمة أكبر من 44، لأنه وفق التركيبة العامة لهذا العدّاد، سيتنقل وفق التتابع : $101100, \dots \rightarrow 101100 \rightarrow 000000 \rightarrow \dots \rightarrow 101011 \rightarrow 101100$ يمكن خلال تشكيل العدّادات باستخدام طرق التوصيل المبينة أعلاه للقلابات الحصول أيضاً على قاسم للترددات، لأن تردد الإشارة f_{Out} لأي قلاب يعادل نصف تردد إشارة الدخل f_{Inp} . وبالعودة إلى المخطط الزمني في الشكل 2-11 ب، يلاحظ أن دور إشارة الخرج يعادل ضعف دورة إشارة الدخل، أي أن القلاب قسم تردد إشارة الدخل على 2. وإذا كان دور إشارة الدخل 200 ns، أي بتردد $f = 1/T = 1/200 \cdot 10^{-9} = 5 \text{ MHz}$ ، فإن دور إشارة الخرج سيكون 400 ns، أي $f = 1/T = 1/400 \cdot 10^{-9} = 2,5 \text{ MHz}$ وهو النصف (أي أن العداد المكون من قلاب واحد قام بالتقسيم على 2). وبالنظر إلى المخطط الزمني لعداد مكون من قلابين كما هو مبين في الشكل 4-11 أو الشكل 6-11، يلاحظ أن تردد إشارة الخرج هو ربع تردد إشارة الدخل. وهذا واضح من أن كل 4 نبضات على الدخل يقابلها على خرج آخر قلاب في العدّاد نبضة واحدة، أي كأن عدّاد يعدّ أربع قيم، يقوم بتقسيم التردد على 4 (بمعدل كل قلاب من العداد يقسم على 2 وبالتالي بوجود قلابين في العداد الواحد تستطيع أن تقسم على 2×2). وانطلاقاً من ذلك، يمكن استخدام العدّادات مع مولدات إشارة للحصول على إشارات بتردد محدّد. فمثلاً بالشكل 14-11، إذا كان تردد ساعة الدخل CP هو 1 MHz فإن تردد اشارات الخرج يكون كالتالي:



الشكل 14-11: دائرة للحصول على اشارات بترددات مختلفة من اشارة دخل بتردد 1MHz

11-1-6 تطبيقات على استخدام العدادات غير المتزامنة

يوجد العديد من التطبيقات العملية التي يمكن بها استخدام العدادات غير المتزامنة لدرجة أنه من غير المجدي هنا عرض تطبيق عن غيره بسبب كثرة هذه التطبيقات. الغاية من عرض هذه الفقرة، هي الإشارة إلى أن العدادات غير المتزامنة تصنع أيضاً على شكل دارات متكاملة، قد يكون لكل منها أكثر من استخدام. وانطلاقاً من ذلك نورد بعض الدارات المتكاملة في الجدول 11-2 التي تمثل عدادات للاستخدامات العامة.

الجدول 11-2: بعض الدارات المتكاملة التي تمثل بعض العدادات والمهتزازات

74 90	Decode Counter (74176, 74196,...)	DIL-14
7492	Divide by 12 Counter	DIL-14
7493	4 bit Binary Counter (74293)	DIL-14
74160	Synchronous Decode Counter (74162)	DIL-16
74161	Synchronous Binary Counter(74163)	DIL-16
74168	Synchronous Binary Up – Down Counter	DIL-16
74169	Synchronous Decade Up – Down Counter	DIL-16
74177	Binary Counter (74197)	DIL-14
74190	Reversible Decade Counter (74192)	DIL-16
74191	Reversible Binary Counter (74193)	DIL-16
74291	Bi-Directional Counter Shift Register	DIL-20
74390	Dual 4-bit Decade Counter	DIL-16
74393	Dual 4-bit Binary Counter	DIL-14
74490	Dual Decade Counter	DIL-16
74668	Synchronous Binary Up – Down Counter	DIL-16
74669	Synchronous Decade Up – Down Counter	DIL-16
74120	Dual Pulse Synchronizer	DIL-16
74121	Monostable Multivibrator	DIL-14
74124	Dual Voltage Controlled Oscillator	DIL-16
74325	Dual V.C. Oscillator 2ph. Outputs	DIL-16

نتناول من هذا الجدول الدارة 7493A للإشارة إلى كيفية استخدامها، نظراً لكثرة وجودها في التطبيقات العامة. يمكن النظر للدارة 7493A الواحدة على أنها عداد يعدّ على الأكثر 16 قيمة، وبالتالي فهي مكونة من أربعة قلابات، لكن لهذه الدارة بنية خاصة كونها مقسمة داخلياً إلى جزأين منفصلين عن بعضيهما ولكل منهما ساعة خاصة. الجزء الأول مكون من قلاب واحد بمدخل ساعة CLKA مستقل ويسمح بعد قيمتين، والجزء الثاني مكون من ثلاثة قلابات بمدخل ثاني للساعة CLKB يسمح بعد 8 قيم. ويمكن الحصول من هذه الدارة على عداد يعدّ 16 قيمة بوصل خرج الجزء الأول مع دخل الجزء الثاني (ليصبح 2×8). يوجد بالدارة بعض المغارز غير المستخدمة، يشار إليها بالرمز x لأنها غير موصولة بشيء. يرمز لها أيضاً بـ NC. الشكل العام للدارة 7493 المصنع بالتغليف DIL-14، وتوزع الإشارات الرئيسية على المغارز مبين في الشكل 11-15.



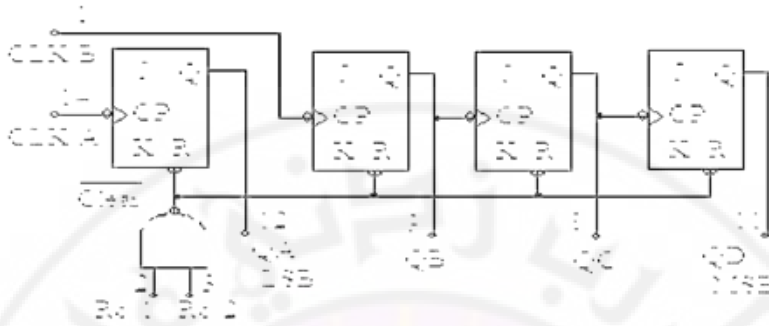
الشكل 11-15: توزيع مآخذ الدارة المتكاملة 7493A

ومن كاتالوك الشركة الصانعة للدارة نجد أن جدول الوظائف لهذه الدارة معطى بجدول مثل الجدول 11-3 (حيث الرمز d في الجدول يعني (Don't Care)).

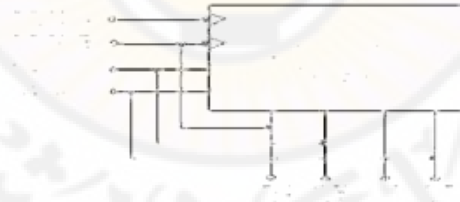
الجدول 11-3: جدول الوظائف للدارة المتكاملة 7493A

مدخل التصفير		المخارج			
$R_0(1)$	$R_0(2)$	Q_D	Q_C	Q_B	Q_A
H	H	0	0	0	0
0	d	العداد يعدّ			
d	0				

كما أن التركيبة الداخلية لهذه الدارة مبينة في الشكل التالي:



يظهر بوضوح من التركيبة الداخلية للدارة الجزءان المكونان لها (حيث تمت تجزئة الدارة إلى جزأين بقصد زيادة مرونة التعامل معها، إذ يمكن النظر إليها على أنها دارتان بقطعة واحدة)، ويمكن استخدامها عداداً لقسمة تردد إشارة الساعة على 2، إذا تم استخدام المدخل CLK A والمخرج Q_A بشكل مستقل، أو إذا تم استخدام عداد ذي ثلاث منازل، أي بثنائي حالات أو استخدام عداد ذي 4 منازل و 16 حالة. بالطبع يمكن من هذه الدارة تشكيل عدادات أخرى غير قياسية باستخدام مبادئ التصميم التي سبق ومرت معنا. فعلى سبيل المثال، يمكن في الشكل 11-16 باستخدام هذه الدارة إظهار التوصيلة العامة لعداد عشري.



الشكل 11-16: عداد عشري باستخدام الدارة 7493

وهذا يعني أنه حتى نتمكن من عد عشر قيم (من 0 حتى 9)، لابد من استخدام الجزأين معاً، وهذا يتم بالاستغناء عن مدخل الساعة الثاني أي CLK B، والاكتفاء باستخدام مدخل الساعة الأول CLK A، شرط أن يربط خرج القلاب الأول Q_A باعتباره مدخل ساعة للجزء الثاني من الدارة. ولكن التوصيل بهذا الشكل يجعل الدارة تعد 16 قيمة. وللوقوف عند القيمة العاشرة بالعد، يجب مجرد الوصول إلى $Q_D Q_C Q_B Q_A = 1001$ ، أي عندما تصبح

$Q_D = Q_B = 1$ أن يصفر العداد، ولذلك تم في الشكل أعلاه وصل كل من Q_D , Q_B بمدخل البوابة NAND، لأن إشارة المسح للدائرة Clear فعالة بالصفر. بالطبع يمكن بنفس الدارة تشكيل أي عداد آخر لعد أي قيمة من 2 حتى 14 أيضاً.

11-2 العدادات المتزامنة

بالعودة للشكل 3-11 الذي يمثل التركيبة الداخلية لعداد غير متزامن ويعد أربع قيم، يلاحظ أن إشارة الساعة تؤثر مباشرة على القلاب A، بينما لا يصل هذا التأثير إلى القلاب B إلا بعد فترة تأخير كافية لمرور الإشارة عبر القلاب A. وهذا التأخير يزداد كلما كانت تركيبة العداد أوسع. فمن أجل عداد يعد ثماني قيم مثلاً، يوجد بالإضافة للتأخير السابق (في القلاب A) تأخير آخر في القلاب B قبل وصول الإشارة إلى القلاب C، وهكذا تتراكم فترات التأخير، بحيث في عداد مكون من n خانة يمكن (بافتراض أن جميع القلابات المكونة للعداد لها نفس زمن التأخير) حساب فترة التأخير الكلية من العلاقة $T = n.t$ ، حيث t هي فترة التأخير في الخانة أو المنزلة الواحدة. يشكل هذا التأخير عقبة لا يستهان بها في كثير من التطبيقات بشكل يحد من الانتشار الواسع للعدادات غير المتزامنة، نظراً لكبر قيمة هذا التأخير في بعض الحالات. يفترض في المخطط الزمني لعمل العداد إظهار أزمنة الاستجابة والتأخير لكل منزلة، لكن في الأمثلة السابقة لم تعتمد هذه التأخيرات خلال رسم المخططات، لأنها قد تشكل عائقاً في فهم المخطط نفسه. إذاً من أهم سلبيات العدادات غير المتزامنة هو طول الزمن اللازم لكي تغير جميع مراحل العداد وضعها من حالة إلى أخرى. تستخدم العدادات المتزامنة للتخلص من السلبيات التي يمكن أن نصادفها بالعدادات غير المتزامنة، لأنه في العدادات المتزامنة تعمل جميع مراحل العداد بنفس الوقت. يتحقق هذا بوصل مدخل الساعة في جميع قلابات العداد مع إشارة الساعة الأساسية. وهكذا لتصميم عداد متزامن مكون من ثلاثة منازل، ويعد 8 حالات من: 000 إلى 111....، ومن ثم يعود إلى الـ 000، ننتقل من الجدول 4-11 الذي يمثل جدول الحالات. بما أن إشارتي الساعة والمحو في العدادات المتزامنة تكون موصولة بنفس الوقت إلى جميع القلابات المكونة للعداد، يكفي لرسم الشكل العام للعداد أن نحدد المعادلات المنطقية التي توصف مدخل كل قلاب يدخل بتركيبة

العَدَّاد على حدة. ننتقل في ذلك وفق الطريقة المباشرة من جدول الحالة الذي يتضح منه أن القلاب الأول A بالخرج Q_A يبدل حالته (انتقال من الحالة 0 إلى الحالة 1 أو بالعكس) مع كل نبضة ساعية.

الجدول 4-11: جدول الحالة لعَدَّاد متزامن يَعدُّ ثماني قيم

CP	Q_C	Q_B	Q_A
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

هذا يعني إذا كان القلاب المستخدم بتركيبية العَدَّاد من النوع J-K يجب أن يكون هذا القلاب موصولاً بحالة القلب، أي أن مداخله $J_A = K_A = 1$. أما القلاب الثاني كما يبدو من جدول الحالات، فإنه يبدل حالته بعد كل نبضتين من Q_A (أربع نبضات CP لأنه يَعدُّ حتى 4). بمعنى آخر القلاب الثاني B بالخرج Q_B يبدل حالته عندما يكون $Q_A = 1$. وبما أن القلاب J - K حتى يغير حالته يجب أن يكون $J = K = 1$ ، تصبح معادلة المدخلين J_B و K_B كالتالي: $J_B = K_B = 1 = Q_A$. وبالنسبة للقلاب الثالث C بالخرج Q_C ، يلاحظ من الجدول أعلاه أنه يبدل حالته عندما يكون كل من Q_A و Q_B بحالة الواحد، وهذا ما يغير عنه بالشكل: Q_B . $J_C = K_C = Q_A \cdot Q_B$. ويرسم هذه المعادلات يكون المخطط المنطقي لمثل هذا العَدَّاد، وفق ما هو مبين في الشكل 11-17. هذا هو المبدأ العام في تشكُّل أي عداد متزامن قياسي، حيث من أجل عَدَّاد بأربع مراحل، يفترض إضافة شرط رابع للقلاب الرابع مماثلاً لما سبق، وهو أن هذا القلاب الرابع D سيغير حالته عندما يكون كل من Q_A, Q_B, Q_C بحالة الواحد، ويعبر عن ذلك بالعلاقة: $J_D = K_D = Q_A \cdot Q_B \cdot Q_C$.



100.0
0.0

—

الشكل 11-17: التركيبة الداخلية لعداد متزامن قياسي، يعد ثماني قيم

11-2-1 مراحل تصميم العدادات المتزامنة

يتبع عند تصميم عداد متزامن غير قياسي (عداد مقاس N) يعد إلى القيمة N التي ليست من رتبة 2^n ، عدة مراحل نلخصها بالتالي:

- 1- بناء جدول لحالات العداد الحالية واللاحقة.
 - 2- تحديد نوع القلايات التي تستخدم في بناء العداد وجدول التحريض المقابل لتحديد قيم مداخل القلايات.
 - 3- تحديد عدد القلايات اللازمة وشكل استجابتها للنمضة الساعية.
 - 4- رسم الهيكل العام للعداد، بحيث توصل إشارتا الساعة والمحو لكل القلايات.
 - 5- تحديد العلاقات المنطقية التي تحدد قيم مداخل القلايات، وتبسيطها.
 - 6- جمع الخطوات الخمس السابقة، برسم المخطط المنطقي للعداد بشكله النهائي.
- لتوضيح آلية تطبيق هذه الخطوات في أثناء تصميم عداد متزامن غير قياسي نعتد مثلاً محدداً، وليكن تصميم دائرة عداد عشري متزامن.
- أولاً- بناء جدول لحالات العداد الحالية واللاحقة: بما أن الرقم الذي سيأتي بعد الصفر في عملية العد هو الواحد والرقم الذي سيأتي بعد التسعة هو الصفر، فإن جدول حالات العد الحالية واللاحقة للعداد المطلوب يكون وفق ما هو مبين بالجدول 11-5 (الأعمدة الثلاثة الأولى من جدول حالات العداد).

ثانياً: بفرض أننا سنستخدم القلابات J-K في بناء هذا العدّاد. ننتقل من جدول التحريض العائد للقلاب J-K ونعيد تشكيل الجدول 5-11 بإضافة عمود جديد (الرابع) ليحدّد قيم مداخل القلابات، انطلاقاً من الحالة التي سينتقل لها العدّاد مع ورود النبضة الساعية. حيث نجد على سبيل المثال بالسطر الأول من الجدول الجديد، أنه بعد ورود النبضة الأولى ينتقل العدّاد من الحالة $q = 0$ أي (qdqcqbqa) إلى $Q = 1$ ($Q_D Q_C Q_B Q_A$).

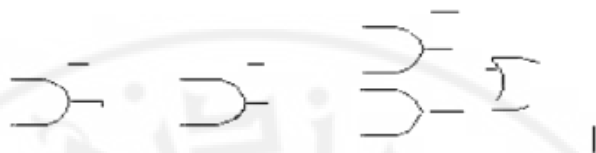
الجدول 5-11: جدول الحالات لعدّاد عشري مع القيم المفترضة لمداخل القلابات

الساعة Clock	الحالة السابقة Present States	الحالة اللاحقة Next States	مداخل القلابات Flip-Flop Inputs
CP	qd qc qb qa	Q_D Q_C Q_B Q_A	$J_D K_D J_C K_C J_B K_B J_A K_A$
0	0 0 0 0	0 0 0 1	0 d 0 d 0 d 1 d
1	0 0 0 1	0 0 1 0	0 d 0 d 1 d d 1
2	0 0 1 0	0 0 1 1	0 d 0 d d 0 1 d
3	0 0 1 1	0 1 0 0	0 d 1 d d 1 d 1
4	0 1 0 0	0 1 0 1	0 d d 0 0 d 1 d
5	0 1 0 1	0 1 1 0	0 d d 0 1 d d 1
6	0 1 1 0	0 1 1 1	0 d d 0 d 0 1 d
7	0 1 1 1	1 0 0 0	1 d d 1 d 1 d 1
8	1 0 0 0	1 0 0 1	d 0 0 d 0 d 1 d
9	1 0 0 1	0 0 0 0	d 1 0 d 0 d d 1

ينتقل القلاب الأول A من $qa = 0$ إلى $Q_A = 1$ ، عندما يكون $J = 1$ و $K = d$. وبنفس الأسلوب ينتقل القلاب الثاني B من الحالة 0 إلى الحالة 0، وفق ما هو مبين في الجدول أعلاه عندما يكون $J = 0$ و $K = d$ ، وهكذا نستمر بكتابة بقية التراكيب.

ثالثاً: يحدّد عدد القلابات n بالعدادات المتزامنة اللازمة للعدّاد من العلاقة المستخدمة في العدّادات غير المتزامنة، وباعتبار أن $M = 10$ ، يتضح أن عدد القلابات اللازمة هو 4. وهذا يعني كما هو الحال بالنسبة للعدادات غير المتزامنة تشكيل منطق خارجي يعمل على تفسير العدّاد بعد عدّه للقيمة 1001. وهنا سنفترض أن القلابات المختارة لبناء العدّاد تستجيب للوجه الصاعد للنبضة.

رابعاً: توصيل كل من الإشارتين Clock و Clear لكل القلابات بأن واحد بالهيكل العام للعداد، وفق ما هو مبين في الشكل 11-18.



الشكل 11-18: التركيبة الداخلية لعداد عشري متزامن

خامساً: تحدّد العلاقات المنطقية لقيم مداخل القلابات في كل مرحلة من العמוד الثاني والعمود الأخير (الرابع) من الجدول 11-5 واختصارها. وبما أن هذه الخطوة هي الأهم والأصعب في عملية التصميم، نبينها بطريقتين.

أ- يلاحظ بالنظر إلى جدول الحالات أعلاه تحت التسمية $J_A K_A$ وجود إما القيم (d 1) أو (1 d)، حيث d يمكن أن تأخذ القيمة 0 أو 1، ولكن قيمة d هنا حتماً هي الواحد، لأنه يفترض أن يكون المدخل J للقلاب موصولاً بالمدخل K في حال استخدامه بتشكيل العداد. وبالتالي كامل العמוד سيحتوي تحت التسمية $J_A K_A$ القيم (1 1)، للدلالة على أن القلاب الأول في العداد سيعمل دوماً بصيغة القلب، وتكون معادلته $J_A = K_A = 1$. بنفس الأسلوب يمكن من الجدول أعلاه تحديد المعادلات التي تحدد قيم بقية المداخل وفق التالي: القلاب B يغير حالته عندما يكون $Q_A = 1$ و $Q_D = 0$ أي أن معادلة مداخله هي: $Q_D' = Q_A \cdot J_B = K_B$ وهذا يتحقق ببوابة AND بمدخلين. أما القلاب C فيغير حالته عندما يكون $Q_A = Q_B = 1$ أي أن $Q_B \cdot Q_A = J_C = K_C$ وهذا يتحقق أيضاً باستخدام بوابة AND وأخيراً القلاب D يقلب من حالة إلى أخرى بحالتين هما:

عندما يكون $QA = QB = QC = 1$ أو عندما يكون $QA = QD = 1$. وهذا يتحقق ببوابتي AND وبوابة OR لربط مخارج بوابات AND، وفق الشكل 11-18.

ب- يتمثل قيم مداخل القلابات بالارتباط مع الحالة السابقة مباشرة بمخططات كارنوف، نظير فيما يلي آلية الحصول على معادلة مدخل أحد القلابات، وليكن القلاب A مثلاً وفق التالي: من أجل المدخل JA من الجدول أعلاه يمكن الحصول بوجود أربعة متحولات تمثل الحالة السابقة (qdqcqbqa)، على المخطط المبين في الشكل 11-19 أ، حيث أربعة متحولات تقابل مخطط من 16 خلية بالرماز من 0000 حتى 1111. وبما أن العداد سوف لن يمر بالحالات التي هي أكبر من 1001 تُعدّ الخلايا بالرماز من 1010 حتى 1111 بالقيمة d. وتتكملة قيم الخلايا كما هو مبين في الشكل 11-19 ب، يلاحظ أن كامل المخطط يقابل حلقة واحدة، يلغى فيها جميع المتحولات ويكون الناتج هو نفسه الذي تم الحصول عليه سابقاً وهو أن $JA = 1$.

$Q_D Q_C$	$Q_D Q_C$	00	01	11	10	$Q_D Q_C$	$Q_D Q_C$	00	01	11	10
00		1	d	d	1	00		1	d	d	1
01		1	d	d	1	01		1	d	d	1
11		d	d	d	d	11					
10		1	d	d	d	10		1	d		

(-)

(+)

(أ)

(ب)

الشكل 11-19: مخطط كارنوف لحساب قيمة مدخل القلاب A (J_A)

يمكن بنفس الأسلوب تمثيل المدخل K_A ، والحصول على $K_A = 1$. ويمكن الحصول على معادلات بقية مداخل القلابات الداخلة في تركيب العداد. نكتفي بعرض المخطط المقابل لحساب قيمة المدخل J_A ، من دون عرض بقية المخططات الأخرى، نظراً لكثرتها (ثمانية مخططات لأربعة قلابات).

سادساً: يَجْمَل كل ما سبق في مخطط منطقي واحد لإظهار التركيبة الداخلية للعداد بناء على المعادلات التي تم التوصل إليها.

11-2-2 العدادات المتزامنة التصاعدية - التنازلية

يمكن تصنيف العدادات المتزامنة بحسب طريقة العدّ إلى ثلاثة أنواع:

- عدادات تصاعدية فقط وهي العدادات التي تزداد القيمة المخزنة فيها عند قدوم كل نبضة، وتمثل بذلك عملية الزيادة Increment، أي: $S = S + 1$.
- عدادات تنازلية فقط، وهي العدادات التي تتناقص القيمة المخزنة فيها عند قدوم كل نبضة، وتمثل بذلك عملية التناقص Decrement، أي: $S = S - 1$.
- عدادات تصاعدية - تنازلية Up-Down Counters، وهي عدادات يمكن أن تعدّ تصاعدياً أو تنازلياً بحسب إشارة تحكم خارجية.

نشير فيما يلي إلى طريقة تصميم النوع الثالث من خلال مثال حول تصميم عدّاد مكون من ثلاث خانات فقط (ليعدّ من 0 وحتى 7). يتم الانطلاق بعملية التصميم لأيّ عدّاد متزامن من جدول الحالات للعدّاد المطلوب. ويستخدم فيه إشارتان إضافيتان هما: UP (للأعلى) تصاعدي و Down (للأسفل) تنازلي. يلاحظ من جدول الحالات للعدّاد المطلوب والمبين في الجدول 11-6 التغيرات التالية:

الجدول 11-6: جدول الحالة لعدّاد من ثلاث خانات (يعدّ 8 قيم)

UP	DOWN	Q ₂	Q ₁	Q ₀
0	1	0	0	0
1	1	0	0	1
2	1	0	1	0
3	1	0	1	1
4	1	1	0	0
5	1	1	0	1
6	1	1	1	0
7	1	1	1	1

- تتبدل قيمة خرج القلاب الأول Q_A مع كل نبضة قادمة في حالة العدّ التصاعدي والعدّ التنازلي على السواء، أي حالة القلاب، وهذا يعني أن مدخلي هذا القلاب J و K يجب أن تكون موصولة للواحد $J_A = K_A = 1$.

- تتغير قيمة Q_B في كل لحظة تكون فيها قيمة $Q_A=1$ في حالة العدّ التصاعدي و $Q_A=0$ في حالة التنازلي، ويمكن التعبير عن ذلك بالعلاقة: $J_B = K_B = Q_A \cdot UP$ للتصاعدي و $J_B = K_B = Q'_A \cdot Down$ للتنازلي، بحيث تصبح العلاقة النهائية لهذه الخانة: $J_B = K_B = Q_A \cdot UP + Q'_A \cdot Down$.

- تتغير قيمة الخانة الثالثة Q_C في كل لحظة تكون فيها $Q_A=1$ و $Q_B=1$ في حالة العدّ التصاعدي أي $J_C = K_C = Q_A \cdot Q_B \cdot UP$ وكذلك عندما تكون $Q_A=0$ وكذلك $Q_B=0$ في حالة العدّ التنازلي، أي $J_C = K_C = Q'_A \cdot Q'_B \cdot Down$ ، وبهذا الشكل تصبح العلاقة النهائية لهذه الخانة كالتالي: $J_C = K_C = Q_A \cdot Q_B \cdot UP + Q'_A \cdot Q'_B \cdot Down$.

برسم هذا المنطق الإضافي الذي تم التوصل إليه إضافة إلى التركيبية الأساسية للعداد، يمكن الحصول على التركيبية الداخلية لعداد مترامن تنازلي - تصاعدي بثلاث خانات يعدّ ثماني قيم، وفق ما هو مبين في الشكل 11-20.



الشكل 11-20: المخطط العام لإدارة عداد تصاعدي تنازلي يعدّ ثماني قيم

11-2-3 العدادات ذات الدخل المتوازي

يلزم ببعض التطبيقات استخدام عدادات قادرة على أن تكمل عملية العدّ من قيمة معينة بشكل تصاعدي أو تنازلي. وبالتالي يجب لمثل هذه العدادات، إضافة إلى القلايات المشكلة للعداد، إضافة منطق يجعل العداد قادراً على إدخال البيانات في المرحلة الأولى قبل بدء عملية العدّ وإجراء عملية العدّ في المرحلة التالية ابتداء من القيمة التي تم إدخالها. يسمى

هذا النوع بالعَدَّات ذات الدخل المتوازي Counter with parallel load ويضم هذا النوع من العدادات عادة إشارتين قياديتين إضافيتين هما:

- إشارة الشحن (أو التعبئة) Load، وتستخدم لإدخال البيانات إلى العدَّاد.
 - إشارة Count (العد)، وتستخدم لجعل العدَّاد يعدّ ابتداءً من القيمة المخزنة.
- بقصد التبسيط نعرض فيما يلي مثلاً عن عدَّاد ب 3bit، يستخدم قلابات J - K. يمكن بإضافة منطق محدّد إلى التوصيلية القياسية للقلابات المكونة للعدَّاد الحصول على التركيبية الداخلية لمثل هذا العدَّاد، وفق ما هو مبين في الشكل 11-21.



الشكل 11-21: التركيبية الداخلية لعدَّاد، يعدّ ثمانية قيم وبداخل متوازٍ

يتضح من المخطط العام لمثل هذا العدَّاد وجود مداخل خارجية لإدخال معطيات على التوازي وإخراج نتيجة العدّ على مخرجه على التوازي أيضاً. فمن أجل عدَّاد مكون من ثلاثة قلابات، يوجد ثلاثة مداخل خارجية I_1, I_2, I_3 وثلاثة مخرج O_1, O_2, O_3 لقراءة القيمة المخزنة بالعدَّاد على التوازي. يُحدّد عمل مثل هذه العدَّادات بجدول الوظائف، الذي يمكن أن يكون مثل الجدول 11-7.

الجدول 7-11: جدول الوظائف لعداد بدخل متواز

Function الوظيفة	العداد Load Count	التعبئة Clear	المسح
تصفير العداد	d	d	0
لا تغيير No Change	0	0	1
إدخال Load	d	1	1
عدّ Count	1	0	1

بالاعتماد على القيم الموجودة على إشارتي التحكم الأساسيتين Load و Count، بجدول الوظائف تتحدد الوظيفة التي سيقوم بها العداد. يتضح من السطر الأول بجدول الوظائف أنه مجرد تفعيل إشارة المسح، فإن العداد سيصفر بمعزل عن وجود النبضات الساعية، ولا يصبح هناك أي دور لبقية الإشارات الأخرى مثل Load و Count. تم التعبير عن ذلك بوضع الرمز d في الجدول أعلاه مقابل كل منهما. أما عندما لا تكون إشارة المسح فعالة، فيلاحظ ثلاثة أنماط لعمل العداد هي: إدخال وعدّ وذاكرة، نبيّنها على مستوى الخانة الأولى. الأول: أن لا يغير العداد حالته، وهذا يقابل حالة عدم تفعيل كل من Load و Count. يتصرف العداد بهذه الحالة وكأنه سجل يحتفظ بقيمة معينة، لأن $Load\ Count = 0\ 0$ تقابل عبر بوابة AND رقم 1 والعواكس رقم 6 و 7 والبوابتين AND2 و OR4 أن يكون $J = 0$ وعبر AND1، AND3 و OR5 أن يكون $K = 0$ حالة ذاكرة بالنسبة لكل قلاب من القلابات المشكلة للعداد ($J=K=0$).

الثاني: الإدخال، أي عندما يكون $Load = 1$ لا يوجد أي دور لقيمة إشارة التحكم Count، التي يمكن أن تكون 0 أو 1 (أي d)، بحيث ما سيطبق على I يدخل على J ومتممه عبر العاكس رقم 8 سيطبق على K أي حالة إدخال للقلاب.

الثالث: العدّ، وهذا يقابل الوضع الذي يكون فيه $Count = 1$ و $Load = 0$ وبهذه التركيبة يلغى أي أثر للبيانات الموجودة على المداخل الخارجية I، لأن خرج البوابات AND رقم 2 و 3 سيكون دوماً بحالة الصفر ومدخلي كل قلاب $J = K = 1$ ، أي حالة القلب بالنسبة للقلاب والعدّ بالنسبة للعداد. أضيف للتركيبة الداخلية للعداد بدخل متواز المبين في الشكل

أعلاه البوابة AND رقم 10 بخرج خاص يسمى Carry Out بالخرج المنقول، بحيث يبقى هذا الخرج في حالة 0 طالما العداد يعد في مجال أصغر من مجاله، أي أقل من 7 في المثال المبين في الشكل أعلاه، ويتحول إلى 1 فقط عندما يصل العداد إلى أكبر قيمة يستطيع عدّها، وهي $2^{(111)}$. ويستخدم هذا المخرج المهم لربط عدة عدادات لتكوين عداد بسعة أكبر. وقد تم استخدام العاكس رقم 9 للتعبير عن استخدام الوجه المعاكس للنبضة .

3-11 قطر العدادات Counter Cascading

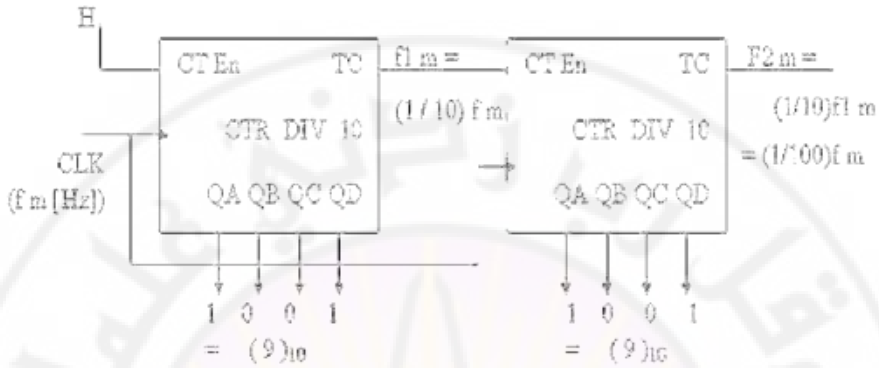
لإظهار آلية قطر العدادات واستخداماتها، ننتقل من تطبيق محدد. وبالنظر إلى واحدة من لوحات الإظهار المستخدمة مع مضخة في محطة للوقود مثلاً، والتي تظهر المبلغ المقابل لكمية البنزين التي يتم تعبئتها والمكونة من ثلاث خانات، يلاحظ أن مثل هذه اللوحة تظهر أرقاماً عشرية ضمن المجال 000 حتى 999 مثلاً. ويمكن تصوّر العملية كالتالي: كل خانة يقابلها عداد عشري يعدّ من 0 حتى 9. ويجب أن تكون هذه العدادات مربوطة مع بعضها البعض، بحيث عندما يعدّ أول عداد عشر قيم يصفر ويبدأ العداد الثاني بعد قيمة واحدة، ثم يستمر العدادان الأول والثاني بالعدّ إلى حين الوصول إلى القيمة 99، وبعد ذلك يمكن أن يبدأ العداد الثالث بعد أول قيمة. وهكذا حتى يتم الوصول إلى المبلغ المقابل. تستخدم إشارة الخرج المنقول المبينة في الشكل 11-20، لوصل العدادات الثلاثة مع بعضها بكتلة واحدة لتحقيق التوصيف أعلاه، بحيث يوصل الخرج المنقول لكل عداد والذي يسمى TC (من Terminal count) مع مدخل Count للعداد اللاحق. تستخدم هذه العملية لهدفين أساسيين: - بناء عدادات ذات ساعات أكبر .

- استخدام العداد لتجزئة الترددات العالية (مقسم).

يبين الشكل 11-22 آلية قطر العدادات من خلال مثال لربط عدادين عشريين للحصول على عداد يعدّ 100 قيمة (00 حتى 99)، وبنفس الوقت يقسم تواتر إشارة الدخل f_m على 100 أيضاً. يتم ربط عدادين (أو أكثر) في الشكل أعلاه وفق التالي:

- 1- يوصل مدخل الساعة لكل عداد بإشارة Clock الرئيسية.
- 2- يوصل مدخل طلب العدّ CTEn من Count Enable في العداد الأول إلى

المستوى العالي High للمباشرة بالعدّ فور ورود النبضات الساعية.
3 - يوصل مدخل طلب العدّ CTEn لكل عدّاد مع الخرج TC للعدّاد الذي يسبقه.



الشكل 11-22: المخطط الصندوقي لقطر عدّادين عشريين

ويحسب المجال الكلي للعدّاد الناتج بضرب ساعات جميع العدّادات المكونة له، حيث السعة الكلية للعدّاد المبين أعلاه هي: $10 \times 10 = 100$. يكون المخرج TC في حالة "1" فقط عندما يصل العدّاد إلى أعلى قيمة يستطيع عدّها، وتظهر بالعدّاد العشري القيمة 1 على المخرج TC مقابل كل عشرة نبضات ساعية CP على المدخل. وهكذا يمكننا تشكيل المقسم الذي نرغب به باستخدام مبدأ قطر العدادات.

11-4 العدّادات الخاصة

تصمم العدّادات لتوليد أي تتابع من الحالات، حيث إن العدّادات التي سبق وأشرنا إليها على مختلف أنواعها، كانت تتابع تتابع العدّ الثنائي، يسمى هذا النوع التقليدي من العدّادات التي يمر فيها كل عدّاد من حالة إلى أخرى من دون قفز بالعدّ، بالعدّاد المرحلي أو التمرجي Ripple Counter. إضافة لهذا النوع من العدادات الواسع الانتشار، يمكن مصادفة عدادات تتابع تتابع آخر في العدّ غير التتابع الثنائي. نتعرف فيما يلي من مجمل الأنواع الخاصة للعدادات على ثلاثة أنواع فقط نصادفها بكثرة في بعض التطبيقات الخاصة مثل: العداد الحلقة، وعداد جونسون، ومولد الأعداد العشوائية، الذي يمثل سجل إزاحة يولد أعداداً لا تتابع أي نظام منطقي.

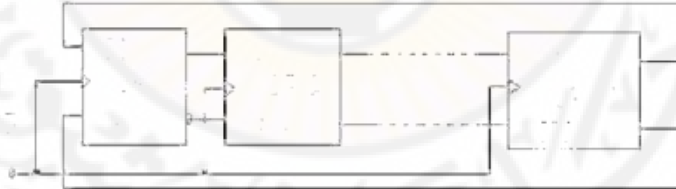
أولاً- العداد الحلقي Ring Counter

يمثل هذا النوع من العدادات سجل إزاحة دوراني، بحيث يكون في لحظة ما قيمة خرج قلاب واحد بحالة الواحد وقيمة خرج باقي القلابات أصفاراً. فمثلاً، جدول الحالات لعداد حلقي مكون من عشرة قلابات يبدو كما في الجدول 11-8.

الجدول 11-8: جدول الحالات لعداد حلقي

CP	Q ₀	Q ₁	Q ₂	Q ₃	Q ₄	Q ₅	Q ₆	Q ₇	Q ₈	Q ₉
0	1	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0
7	0	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	0	1	0
9	0	0	0	0	0	0	0	0	0	1

يظهر الشكل 11-23 التركيبة العامة للعداد المقابل، الذي يستخدم لتوليد تتابع إشارات توقيت



الشكل 11-23: المخطط المنطقي للعداد الحلقي

ثانياً- عداد جونسون Johnson Counter

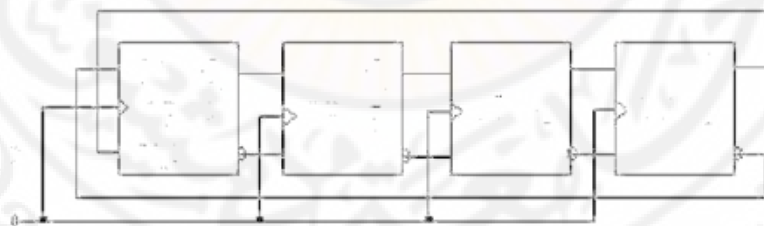
يمتاز هذا النوع من العدادات (نسبة لجونسون) بتركيبة خاصة. لأنه يمثل عداد مع إزاحة Shift Counter، بحيث إن العداد المكون من n bits يدور بتاً وحيداً من بين قلاباته ليولد n حالة مميزة. يمكن مضاعفة عدد الحالات إذا بدلنا طريقة وصل سجل الإزاحة ليصبح عداداً حلقياً مبدل النهاية Switch-tail ring counter والذي فيه يمر العداد من أجل n

قلاّب بتتابع مؤلف من $2n$ حالة. يوصل بعدّاد جونسون مبدل النهاية المخرج المتمم Q' من القلاّب الأخير إلى مدخل القلاّب الأول. يظهر الجدول 11-9 جدول الحالات لعدّاد من هذا النوع مكوناً من أربعة قلاّبات.

الجدول 11-9: جدول الحالات لعدّاد جونسون مكون من أربعة قلاّبات

CP	Q_A	Q_B	Q_C	Q_D
0	0	0	0	0
1	1	0	0	0
2	1	1	0	0
3	1	1	1	0
4	1	1	1	1
5	0	1	1	1
6	0	0	1	1
7	0	0	0	1
عودة من جديد للحالة 0 0 0 0				

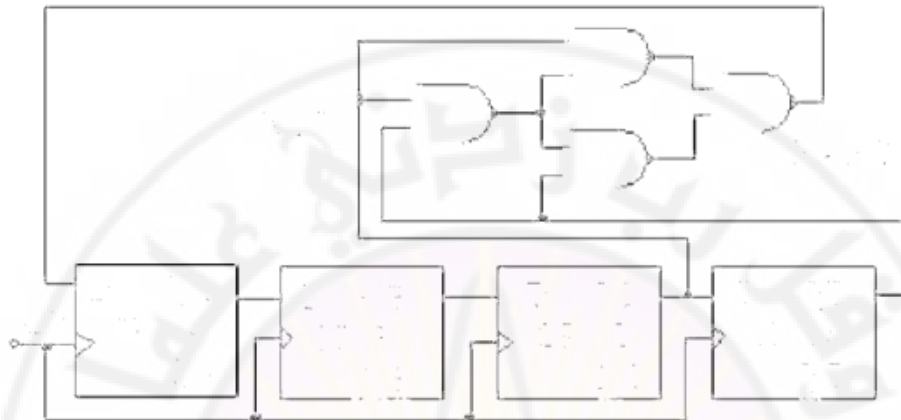
بالتالي انطلاقاً من الحالة الأولى بعد التصفير، ينتقل العدّاد عبر تتابع من ثماني حالات، بحيث كل عملية إزاحة نحو اليمين تدخل 1 من اليسار إلى أن يمتلئ السجل بالوحدات، والتركيبية الداخلية لعدّاد جونسون المقابلة لجدول الحالات أعلاه مبينة في الشكل التالي:



ثالثاً - مولد الأعداد العشوائية

يمثل مولد الأعداد العشوائية دائرة من نوع خاص تحاكي برامج انتقاء عدد عشوائي بمساعدة الحاسوب من مجموعة أعداد. يمكن اعتبار مثل هذه الدارات من العدّادات الخاصة أيضاً. يُعدّ مولد الأعداد العشوائية بالأساس سجل إزاحة ينتج أعداد ثنائية بترتيب عشوائي. توجد طرق عدة لتحقيق مثل هذه الدارة، منها استخدام سجل إزاحة تتم تغذية مداخله من شبكة

منطقية خاصة (بأشكال مختلفة)، تستنبط إشاراتهما من القلايات التي يتكون منها السجل.
يمكن اعتبار الشكل 11-24 مثالاً لمولد الأعداد العشوائية.



الشكل 11-24: مولد للأعداد العشوائية الزائفة

تمثل الدارة في الشكل أعلاه، توصيلة بسيطة لا تولد أرقاماً عشوائية تماماً، لأنه إذا عرفت الحالة الأولية لسجل الإزاحة، يمكن التنبؤ بتسلسل الأعداد المولدة، لذلك يطلق على مثل هذه التوصيلة "مولد الأعداد العشوائية الزائفة". علاوة على أن الدارة المبينة في الشكل أعلاه لا يمكن أن تحتوي على التركيبة 0000، وبالتالي فإن عدد الحالات المختلفة يصبح 15 حالة تولد من دون أي ضابط. يمكن أن يكون جدول الحالات لمثل هذه الدارة وفق ما هو مبين في الجدول 11-10. تتضمن الشبكة المنطقية المستخدمة في الشكل 11-24 مع سجل مكون من أربعة قلايات أن تبدأ الدارة بالقيمة 1111. تستخدم بوابة XOR (حيث إن أربع بوابات NAND بالتوصيلة المبينة بالشكل أعلاه تقابل وظيفياً بوابة XOR) لمقارنة حالة المخارج للقلايين D و C، بحيث تغذي المدخل D_A بالقيمة المنطقية 1 عندما تكون كل من Q_D و Q_C مختلفتين، وتغذي D_A بالقيمة 0 عندما تكون كل من Q_D و Q_C متساويتين. بالطبع عند أخذ مخارج أخرى من السجل بالشكل أعلاه كمدخل للبوابة XOR (تراكيب مختلفة)، ينتج مولد مغاير للمولد المشار إليه والذي فيه كل قيمة مضبوطة على الدخل تقابل قيمة عشوائية على الخرج.

الجدول 10-11: جدول الحالات لإدارة توليد أعداد عشوائية

القيمة المتولدة	حالة القلابات $Q_A Q_B Q_C Q_D$	القيمة العشرية
15	1 1 1 1	0
7	0 1 1 1	1
3	0 0 1 1	2
1	0 0 0 1	3
8	1 0 0 0	4
4	0 1 0 0	5
2	0 0 1 0	6
9	1 0 0 1	7
12	1 1 0 0	8
6	0 1 1 0	9
11	1 0 1 1	10
5	0 1 0 1	11
10	1 0 1 0	12
13	1 1 0 1	13
14	1 1 1 0	14

11-5 مولدات النبضات الساعية

نشير فيما يلي إلى موضوع يتعلق بكل دائرة تتابعية، خاصة المترامنة منها، هو موضوع مولدات النبضات الساعية الخارجية Clock Generators المسؤولة عن توليد النبضات CP اللازمة لعمل الدارات المنطقية التتابعية. تصنع مولدات النبضات الساعية بشكل مستقل، ومن دون الدخول بالتفاصيل حول آليات التشكيل، سنقتصر من مجمل الأشكال المستخدمة لتوليد النبضات الساعية على عرض المولدات التي تتشكل باستخدام البوابات المنطقية للتطبيقات البسيطة ومنها:

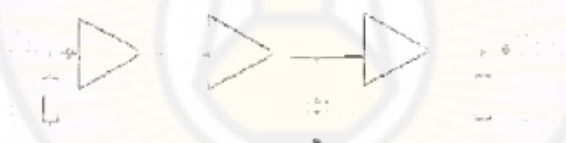
1- مولد نبضات ساعية بسيط باستخدام بوابة NOT. يمكن تشكيل مثل هذا المولد باستخدام الدارة المتكاملة 7404 التي تضم ستة بوابات NOT. التوصيلية الأساسية لمثل هذا المولد مبينة في الشكل 11-25. لتشكيل مثل هذا المولد يلزم عدد فردي غير محدد من

البوابات NOT موصولة في الحلقة ويكون التواتر f معطى بالعلاقة: $f = (1/n) \tau$ ، حيث n عدد العواكس و τ هو زمن تأخير عبور الإشارة في العاكس الواحد.



الشكل 11-25: مولد بسيط للنبضات الساعية

لا تُعدّ التوصيلة المبينة في الشكل أعلاه مناسبة لكثير من التطبيقات، لأن التواتر مرتبط مع زمن التأخير τ الذي يرتبط بدوره مع درجة حرارة الوسط. وبما أن المطلب الأساسي من تصميم مولد نبضات منطقي، هو استبعاد أثر ارتباط f مع τ فإن ذلك يمكن تحقيقه بوصل عدد فردي من العواكس وعنصر RC بثابت زمني، كما هو مبين بالشكل 11-26. تواتر المذبذب Oscillator في الشكل أدناه يعطى بالعلاقة التقريبية التالية:

$$f = 1/(2R_2C)((0.405R_1/(R_1 + R_2)) + 0.693)$$


الشكل 11-26: مولد للنبضات الساعية مع عنصر RC بثابت زمني كبير

2- مولد نبضات ساعية باستخدام عاكسين.

يمكن تشكيل مولد منطقي بشكل سهل ومألوف عن طريق عاكسين فقط وفق ما هو مبين في الشكل جانباً.



السلية الأساسية لهذه التوصيلة هي أن المولد لا يتذبذب من أجل سعة معينة للمكثف وهي $C = 0$ وهنا تجدر الإشارة إلى أننا قد نصادف في بعض المذبذبات بوابات NAND (باستخدام الدارة 7400) بدلاً من NOT . بالطبع هذا لا يغير بالموضوع أي شيء.

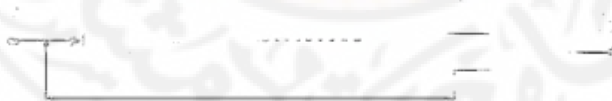
3- مولد Schmitt Inverter Waveform

دون الدخول بمزيد من التفاصيل حول هذا المولد، يمكن أساس هذا النوع من الدارات عديمة الاستقرار المبين بالشكل أدناه، تعديله بإضافة بعض العناصر لتوليد تواترات مختلفة.



4- مولد نبضات بتواتر مضاعف

يمكن تشكيل مولد نبضات بتواتر مضاعف باستخدام بوابة XOR مع العواكس بتوصيلة محددة، وفق ما هو مبين في الشكل 11-27. يتضح من خلال دراسة المخطط الزمني للدارة المبينة في الشكل أدناه أن وظيفة المضاعف مبنية على التأخير الزمني للعاكس.



الشكل 11-27: مضاعف التواتر

والإشارة المتأخرة $f(t)$ مع إشارة الدخل f متداخلة مع بعضها عبر XOR وعرض نبضة الخرج مرتبط بعدد العواكس الموصولة. نكتفي عند هذا الحد بما تم عرضه من مولدات للنبضات الساعية، خاصة أننا أشرنا بنهاية الفصل السابع لمود نبضات ساعية باستخدام الدارة 555 الذي يصلح لمختلف التطبيقات.

11-6 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

1- أي من العبارات التالية صحيحة

- a- Asynchronous events do not occur at the same time
- b- Asynchronous events are controlled by a clock
- c- Synchronous events do not need a clock to control them
- d- Only asynchronous events need a control clock.

2- لماذا يمكن لعداد نوع synchronous أن يعمل بتواتر أعلى من ripple counter

- a- The flip-flops change one after the other
- b- The flip-flops change at the same time
- c- A synchronous counter cannot operate at high frequen.
- d- A ripple counter is faster

3- كم حالة طبيعية يوجد بـ 4-bit ripple counter

- a- 4
- b- 8
- c- 16
- d- 32

4- بأربعة قلابات من نوع JK يمكننا بناء عداد يستطيع عدّ

- a- 20 values
- b- 16 values
- c- 32 values
- d- None of the above

5- لعدّ عشرين قيمة (من 0 حتى 19) يلزم

- a- 20 Flip-Flops
- b- 2 Flip-Flops
- c- 5 Flip-Flops
- d- 4 Flip-Flops

6- كم دائرة تكاملية من النوع 7490 يجب وصلها بشكل متتابع حتى تكون قادرة على العدّ

- a- 4
- b- 3
- c- 2
- d- 1

إلى القيمة 999.

7- أي مما يلي مثلاً لعداد بموديولا منقطع truncated modulus (غير قياسي)

- a- 8
- b- 13
- c- 16
- d- 32

ثانياً: أسئلة متفرقة

- 1- ما هو عدد القلايات اللازم لعد 100 قيمة بشكل ثنائي، وما هي القيمة التي يمكن عدّها بعدد يضم سبعة قلايات؟
- 2- ارسم المخطط المنطقي لعداد تموجي ثنائي ذي 4 bits يعد تنازلياً ويستخدم:
(أ) قلايات تقدح بالحافة الموجبة للساعة
(ب) قلايات تقدح بالحافة السالبة للساعة
- 3- صمم دائرة عداد عشري لاتزامني يعد بشكل تصاعدي وتنازلي
- 4- صمم دائرة عداد يعدّ ستين قيمة: - مرة تصاعدي بشكل متزامن.
- ومرة تنازلي بشكل غير متزامن.
- 5- صمم دائرة عداد BCD باستخدام كل من القلايات T, D, and JK واكتب معادلات الدخل المقابلة لاستخدام كل من القلايات، وبين أي من التصاميم الثلاثة هو الأكثر كفاءة.
- 6- صمم دائرة لتوليد إشارة بتردد 4 KHz من منبع تردد ب 1KHz
- 7- من مولد ساعة بتردد 16 MHz لنظام رقمي، يطلب باستخدام العدادات:
* إنتاج نبضات ساعة لإحدى الدارات ب زمن 25 ns للدورة الواحدة
* تأمين 800 KHz اللازمة لعمل دائرة أخرى.

الفصل الثاني عشر: تحليل الدارات التتابعية وتصميمها

يعبر عن الشكل العام للدارة التتابعية بنموذج يختلف عن النموذج العام للدارة التركيبية. يشير نموذج الدارة التتابعية إلى قيمة الحالة السابقة للدارة التي توجد عادة في الرابطة العكسية من الخرج إلى الدخل. وتحدد قيم الخرج في الدارة التتابعية بالقيم الحاضرة على الدخل والقيم من المرحلة السابقة. تتكون أي دارة تتابعية من جزأين أساسيين، وفق ما هو مبين في النموذج العام للدارة التتابعية بالشكل 12-1، هما: الأول - الجزء التركيبي Combinational part الثاني - جزء الذاكرة Memory part.



الشكل 12-1: النموذج العام للدارة التتابعية

تكون كل المداخل والمخارج للجزء التركيبي من الدارة المستقل عن الزمن على شكل إشارات مستوى (أي يمكنها أن تأخذ قيم 0 و 1 لفترات زمنية غير محدودة). ويمكن أن تتغير المخارج والمحرضات في أي لحظة استجابة لتغيرات مداخل الدارة. أما عناصر الذاكرة فتتكون عادة من وحدات لها القدرة على تخزين البيانات الثنائية، وبالتالي لها مداخلها ومخارجها الخاصة، بحيث إن البيانات المخزنة في الذاكرة بلحظة ما تحدد حالة الدارة التتابعية التي تضم في تركيبها مثل هذه العناصر. وبحسب العلاقة بين مداخل قسم الذاكرة ومخارجها، نميز بين نوعين من الدارات:

الأول: الدارات التتابعية نموذج الساعة Clock Mode Sequential Circuits وهي دارات يمكن تعيين سلوكها من معرفة إشاراتها في لحظات متقطعة زمنياً. يتكون قسم

الذاكرة في هذا النوع من الدارات من مصفوفة عناصر الذاكرة (التي تتشكل بشكل أساسي من الفلابات المتزامنة). وتسمى أيضاً بالدارات التتابعية المتزامنة.

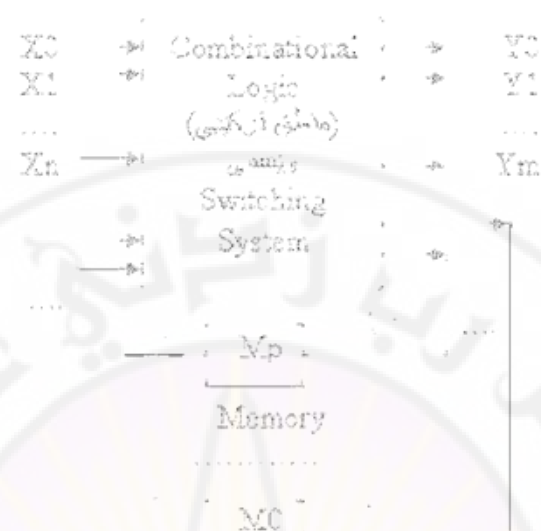
الثاني: الدارات التتابعية نموذج المستوى Level Mode Sequential Circuits يعتمد سلوك هذا النوع من الدارات على إشارات الدخل في أي لحظة من الزمن وعلى الترتيب الذي تتغير فيه المداخل. ويتكون قسم الذاكرة من مجموعة خطوط التغذية الراجعة، حيث يمكن أن تبقى الإشارات على إحدى القيمتين (0 أو 1) لفترة زمنية. ويمكن أن تتغير في أوقات غير محددة أيضاً وتسمى هذه الدارات بالدارات التتابعية غير المتزامنة. وإذا كان سلوك الدارات التتابعية المتزامنة يعتمد على قيم الإشارة في لحظات زمنية محددة، فإن عمل الدارات التتابعية غير المتزامنة وسلوكها يعتمد على الترتيب العام لتغير القيم على الدخل والذي يمكن أن يحدث في أي لحظة من الزمن.

12-1 الدارات التتابعية المتزامنة

يُعبّر عن مفهوم التزامن في الدارات التتابعية المتزامنة بوساطة ساعة توقيت من مولد خارجي يقوم بتوليد سلسلة من النبضات منتظمة الدور Period ، تتوزع في الدارة بشكل يجعل عناصر الذاكرة تتأثر مع وصول هذه النبضات. ويمكن تعديل النموذج العام للدارات التتابعية في الشكل 12-1 وتمثيل أي دارة منطقية تتابعية متزامنة، وفق الشكل المعدل أدناه



وبالانطلاق في دراسة الدارات التتابعية المتزامنة من نموذج عام أكثر تفصيلاً، مثل النموذج المبين في الشكل 12-2 الذي يظهر مداخل الدارة ومخارجها والحالات الداخلية لها، سنحاول إظهار العلاقة بين هذه الباروميترات الأساسية للدارة التتابعية.



الشكل 2-12: المداخل والمخارج والحالات الداخلية للدارة التتابعية المتزامنة

عند تطبيق دخل معين $X_0, X_1, \dots, X_n$ على مدخل النموذج العام للدارة التتابعية يتشكل $Y_0, Y_1, \dots, Y_m$ خرج، وبنفس الوقت يتشكل $Q_0, Q_1, \dots, Q_n$ متحول تحريض، ويخزن كل منها بعنصر ذاكرة مستقل $M_0, M_1, \dots, M_p$ لتحديد الحالة التالية ولتولد بدورها $q_0, q_1, \dots, q_p$ متحول حالة داخلية تمثل مجمل الحالة الحالية للدارة. وتشكل متحولات الحالة مع القيم الجديدة على المداخل X إجمالي دخل الدارة التتابعية. ولتحليل عمل الدارة التتابعية المتزامنة، يعتمد المبدأ التالي: تفتح جميع خطوط التغذية العكسية ويترك واحد منها فقط موصولاً، وتدرس الدارة بناءً على الخط الموصول بمفرده، ثم تأخذ كل التركيب الناتجة بالنسبة للدارة. بالانطلاق من الحالة الداخلية الأولى للدارة q_0 وتركها موصولة بالوقت الذي تفصل به بقية الحالات، ينتج بوجود المداخل الخارجية X وهذه الحالة q_0 تابع منطقي من الشكل: $Y_0 = F_0(X_1, X_2, \dots, X_n, q_0)$ من أجل قيمة ثابتة لـ q_0 . تكتب كل تركيبة خرج مقابلة لتركيبه الدخل ضمن هذه الشروط وكأنها دارة تركيبية. تؤخذ الحالة الداخلية الثانية للدارة q_1 وتفصل بقية الحالات وبنفس الأسلوب يكتب: $Y_1 = F_1(X_1, X_2, \dots, X_n, q_1)$ من أجل قيمة ثابتة لـ q_1 ، وهكذا حتى تؤخذ جميع الحالات الداخلية الممكنة للدارة.

يتم الحصول على الجدول العام الذي يوصف عمل الدارة، بحيث كل عمود في الجدول يرمز لتركيبية متحول دخل وكل سطر لحالة داخلية واحدة، وبالتالي كل خلية في الجدول ناتجة من تقاطع السطر مع العمود تشكل حالة عامة. ويُعبّر عن ذلك من أجل حالة بلحظة زمنية t بالعلاقة التالية: $Y_{j,t} = \Psi(X_{i,t}, q_{k,t})$ من أجل $i = 0, 1, 2, \dots, n$ (المدخل).
 $j = 0, 1, 2, \dots, m$ (المخارج).
 $k = 0, 1, \dots, p$ (الحالات الداخلية).

ويسمى التابع Ψ بتابع الخرج الذي يأخذ الشكل العام :

$$Y_j = F_j (X_0, X_1, \dots, X_n, q_0, q_1, \dots, q_p)$$

بالطبع جدول من هذا النوع غير تام لتوصيف الدارة التتابعية، لأنه لم يلحظ المتحولات الداخلية التي ستحدد الحالة اللاحقة للدارة. ويمكن تصوير هذه العلاقة بجدول آخر تكون فيه كل خلية ممثلة لحالة عامة تتضمن قيمة الدخل الآتية وقيمة الحالة السابقة وهذا ما يمكن التعبير عنه رياضياً بالشكل: $Q_{s,t+1} = \varphi(X_{i,t}, q_{k,t})$ باللحظة $t + 1$ ، حيث: $s = 0, 1, \dots, n$. ويسمى التابع φ بتابع النقل (العبور) أو التحويل، الذي يأخذ الشكل العام التالي: $Q_s = Q_n = F_j(X_0, X_1, \dots, X_n, q_0, q_1, \dots, q_p)$. غالباً ما يدمج الجدولان السابقان بجدول واحد أو أن يعبر عن الدارة بمخطط حالة. تتميز دارات نموذج الساعة بوجود خط منفصل للساعة، ويمكن تعيين سلوكها من معرفة إشاراتها في لحظات متقطعة زمنياً Discrete Instance of Time. وتضم ممرات التغذية العكسية بهذا النوع من الدارات تجهيزات قادرة على تخزين المعلومات، ومن بين هذه التجهيزات تستخدم القلايات باعتبارها عناصر أساسية لتخزين المعلومات. وبالعودة للشكل المعدل للدارة التتابعية المتزامنة، يلاحظ أن الدارات التتابعية المتزامنة تستخدم بالإضافة إلى المدخل الرئيسية إشارات أخرى تؤثر في عناصر التخزين بلحظات زمنية متقطعة. تسمى الدارات التتابعية التي تستخدم نبضات الساعة على مداخل عناصر التخزين "الدارات المحكومة بنبضات الساعة Clocked Sequential Circuits". ويحدد سلوك الدارة المحكومة بالساعة بالمدخل والمخارج وحالة القلايات، حيث كل تركيبة لقيم المدخل والحالة الراهنة للدارة يقابلها خرج وحالة لاحقة جديدة. والخرج النهائي للدارة سيكون إما من الجزء التركيبي

للدائرة فقط أو من القلايات فقط أو من الاثنين معاً. أما دخل القلايات المستخدمة فيأتي من مصدرين، هما: - الجزء التركيبي للدائرة.

- ومولد النبضات الساعية الذي يعطي

نبضات بفواصل زمنية ثابتة، أي أن خرج القلاب لا يتغير حتى ولو تغيرت قيم مداخله القادمة من الجزء التركيبي ما لم تكن النبضة الساعية من مولد النبضات جاهزة أيضاً. ويعود هذا الخرج عبر حلقة التغذية الخلفية في اللحظة التي لا تكون فيها النبضة الساعية على مدخله فعالة ليشكل مع المداخل الخارجية للجزء التركيبي المدخل الإجمالي للدائرة التتابعية. ولذلك هذا النوع من الدارات نادراً ما يبدي عدم استقرار، وبالتالي يُعدّ عملياً النوع الأكثر انتشاراً من الدارات التتابعية.

12-1-1 تحليل الدارات التتابعية المتزامنة

لتحليل الدارات التتابعية المتزامنة، لابد من إتباع الإجراءات التالية:

1- بناء جدول الحالات State table

2- تحديد معادلات الحالات State equations

3- رسم مخطط الحالات State diagram

4- بناء جداول التهييج للقلايات Excitation tables

5- اختصار الحالات وإسنادها State reduction and assignment

وفيما يلي نوضح باختصار من خلال مثال محدّد مفهوم كل من هذه الخطوات، وليكن تحليل الدارة التتابعية المبينة في الشكل 3-12 التي تتكون من عنصري ذاكرة، كل منهما يمثل قلاب RS، ولها مدخل واحد x ومخرج واحد y . حيث يتم الانطلاق من المخطط المنطقي، ويجب اتباع نفس التسلسل المبين أعلاه لخطوات تحليل الدارة التتابعية المتزامنة. يرمز بداية لكل قلاب في المخطط برمز مثل $A, B, \dots$ ويرمز للحالة الحالية المقابلة لكل قلاب برمز $q_A, q_B, \dots$ وللحالة اللاحقة بـ $Q_A, Q_B, \dots$.

الشكل 3-12: مثال لدارة نتابعة بقصد التحليل

12-1-1-1 بناء جدول الحالة State table

تعدّ أول خطوة بتحليل الدارات النتابعة المحكومة بنبضات الساعة، هي الحصول على مخطط للتتابع الزمني لكل من الدخل والخرج والحالات الداخلية وتمثيله بجدول، يسمى أيضاً بجدول الانتقال Transition table. وبما أن لكل قلاب حالة راهنة q باللحظة t وحالة لاحقة Q بعد دور ساعة واحد عند الزمن $t + 1$ ، يتكون جدول الحالة من ثلاثة أجزاء رئيسية هي:

الأول: باسم الحالة السابقة الممكنة لمختلف القلابات المكونة للدارة.

الثاني: باسم الحالة اللاحقة للقلابات عند مختلف القيم الممكنة للدخل.

الثالث: باسم الخرج عند مختلف القيم الممكنة للدخل، كما هو مبين في الجدول 1-12، حيث تتطلب عملية استخراج جدول الحالة إدراج جميع التراكيب الممكنة لكل من الحالة السابقة والقيمة الآنية على الدخل.

الجدول 1-12: الشكل العام لجدول الحالة

الحالة الراهنة (السابقة)	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
$q_A q_B$	$Q_A Q_B$	$Q_A Q_B$	y	y
0 0	0 0	0 1	0	1
0 1	1 0	0 1	0	1
1 0	0 0	1 1	0	1
1 1	1 0	0 0	1	1

ففي هذا المثال من أجل دخل واحد وقلابين يوجد ثمانية تراكيب مختلفة من 000 ولغاية 111، قسمت على عمودين كل منهما بأربعة تراكيب، أحدها من أجل الدخل $x = 0$ والثاني من أجل $x = 1$.

2-1-1-12 State equations تحديد معادلات الحالة

يمكن توصيف سلوك أي دائرة منطقية تتابعية متزامنة بواسطة معادلات الحالة التي تسمى أيضاً بمعادلات الانتقال. ويمكن من معادلات الحالة أو المخطط المنطقي للدائرة تحديد الحالة اللاحقة للدائرة المحكومة بالساعة أو الخرج. وهكذا يمكن تحديد الحالة اللاحقة لكل من القلابين (من نوع RS مثلاً) A و B بالدائرة المنطقية التتابعية المعطاة بالمعادلتين:

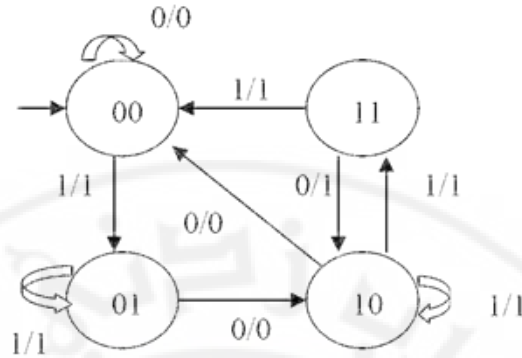
$$Q_A = q_A \cdot x + q_B \cdot x'$$

$$Q_B = q_A' \cdot x + q_B' \cdot x$$

يمكن استخراج تابع الخرج للدائرة المعطاة، وهو: $y = q_A \cdot q_B + x$.

3-1-1-12 State diagram رسم مخطط الحالة

يمكن تمثيل المعلومات الموجودة بجدول الحالة بيانياً Graphically ، على هيئة مخطط يدعى بمخطط الحالة. وتُمثل كل حالة للدائرة بدائرة في المخطط، ويشار إلى الانتقال بين الحالات بخطوط موجهة من حالة المنطلق إلى حالة الهدف. فمثلاً، مخطط الحالة للدائرة التتابعية المعطاة بالمثال أعلاه يمكن تمثيله وفق ما هو مبين في الشكل 4-12. وقد تم عرض الشكل أدناه لمخطط الحالة بوصفه صيغة ثانية لتوصيف عمل الدائرة.



الشكل 4-12: مخطط حالة لدارة تتابعية بأربع حالات

يمثل الرمز الثنائي ضمن الدائرة رقم الحالة. وتمثل القيمة (Label) على كل سهم بين حالتين قيمة كل من الدخل والخرج، بحيث في كل زوج يدل الرقم الذي يقع فوق الخط (/) على قيمة الدخل أثناء الحالة السابقة، والرقم تحت الخط المائل على الخرج.

4-1-1-12 بناء جداول التهيج للقلابات Excitation table

يولد القسم التركيبي في الدارة المنطقية التتابعية المخارج الخارجية، ويوصف جبرياً بوساطة مجموعة من معادلات الخرج. أما قسم الدارة الذي يولد مداخل القلابات، فيوصف بوساطة مجموعة من التوابع المنطقية التي تدعى بمعادلات التحريض. ويقصد ببناء جداول التهيج للقلابات المستخدمة تحديد معادلات الدخل. ويمكن من أجل المدخل الوحيد x والمخرج الوحيد y بالمثل المعطى، التعبير عن المخطط المنطقي للدارة جبرياً بوساطة معادلات دخل القلابات ومعادلة الخرج كالتالي:

$$S_A = R'_A = q_A \cdot x + q_B \cdot x'$$

$$S_B = R'_B = q'_A \cdot x + q'_B \cdot x$$

$$Y = q_A \cdot q_B + x$$

وهنا نشير إلى أنه في حال عدم وجود مخرج محدد للدارة، يتم رسم مخطط الحالة المقابل للمخطط المنطقي بكتابة قيمة وحيدة على كل انتقال مابين حالة وأخرى. ولتوضيح ذلك نعرض فيما يلي مثالا آخر حول تحليل دارات تتابعية متزامنة باستخدام قلاب آخر مثل القلاب J-K، ونظهر من خلاله أيضاً خوارزمية استخراج الحالة اللاحقة للدارة، ليكن المثال ممثلاً للمثال السابق، بحيث يصبح باستخدام القلابات J-K، وبوجود مدخل وحيد x

وفق ما هو مبين في الشكل أدناه.

ولنجر الآن التحليل بوجود القلاب JK (Analysis with JK Flip-Flop). يبدأ التحليل من جدول الحالة. وبالطبع الجزءان المتعلقان بالحالة الراهنة والمداخل يمكن إنجازهما فوراً بكتابة جميع التراكيب الثنائية الممكنة لهما. أما جزء المخارج فيحدّد عادة من معادلات الخرج. ولتحديد الحالة اللاحقة للقلاب، لابد من العودة إلى الجدول الوظيفي. نتلخص خوارزمية استخراج الحالة اللاحقة بالخطوات التالية:

- 1- تعيين معادلات دخل القلابات بدلالة الحالة السابقة ومتحولات الدخل.
 - 2- كتابة القيم الثنائية لكل معادلة من معادلات الدخل.
 - 3- استخدام الجدول الوظيفي المقابل للقلاب المستخدم لتعيين قيم الحالة اللاحقة.
- وبتطبيق هذه الخوارزمية على المخطط في الشكل أعلاه، نجد بداية أنه لا يوجد للدائرة مخارج، وهذا يعني (في مثل هذه الحالات) أنه لا يوجد حاجة لوجود عمود خرج في جدول الحالة (حيث مخارج القلابات تكون هي نفسها مخارج الدارة). ويمكن توصيف الدارة بمعادلات دخل القلابات المستخدمة فقط، وهي:

$$K_A = q_B \cdot x', \quad J_A = q_B, \quad K_B = q'_A \cdot x + q_A \cdot x', \quad J_B = x', \quad J_B = x'$$

ويصبح جدول الحالة المقابل للدائرة بالشكل المبين في الجدول أدناه، حيث يتم الحصول على القيم الثنائية في الجدول مباشرة من معادلات الدخل الأربعة بطريقة مشابهة لتلك المستخدمة بالحصول على جدول الحقيقة من تعبير منطقي.

الشكل 5-12: مخطط الحالة لدارة تتابعية متزامنة

بالطبع من أجل n قلاب يمكن توليد 2^n حالة، وبالتالي مخطط بست حالات يقابل دارة بثلاثة قلابات على الأقل. وتبسيط الدارة يعني تقليل عدد القلابات الداخلة من دون التأثير على الوظيفة العامة للدارة. وهذا ما سنحاول إظهاره من خلال تخفيض عدد حالات الدارة. بداية نبين أنه يوجد عدد غير منته من تتابعات الدخل التي يمكن تطبيقها على الدارة، بحيث كل منها يعطي تتابع خرج وحيد. فمثلاً، التتابع 011001000 على الدخل ابتداء من حالة الدخل البدائية q_0 ينتج من أجل كل دخل بالقيمة 0 أو 1 خرجاً قيمته 0 أو 1 وينقل الدارة إلى حالة جديدة. يمكن الحصول من مخطط الحالة أعلاه من أجل الكلمة (أو السلسلة String) 011001000 تتابع كل من الخرج والحالة وفق التالي: إذا كانت الدارة في الحالة الابتدائية q_0 ، وتم من شريط الدخل قراءة القيمة 0 مثلاً تنتقل الدارة من الحالة q_0 إلى الحالة q_1 ، وتنتج خرجاً بالقيمة 1. وهكذا يستمر التنقل من أجل كل قيمة على الدخل بين الحالات الداخلية للدارة، ويمكن بكل مرة إظهار قيمة الخرج المقابل حتى يتم الوصول إلى الشكل النهائي الذي يقابل الدخل المعطى وفق الترتيب التالي:

State : $q_0 \ q_1 \ q_2 \ q_2 \ q_5 \ q_1 \ q_2 \ q_5 \ q_1 \ q_4$

Input : 0 1 1 0 0 1 0 0 0

Output : 1 0 0 1 1 0 1 1 0

بحيث تظهر في كل عمود الحالة الراهنة وقيمة كل من الدخل والخرج، ويقال أن كلمة الدخل 011001000 تقابل الخرج 100100100. وبما أن المهم هو تحديد تتابعات الخرج التي يمكن الحصول عليها نتيجة تطبيق تتابعات دخل ما بغض النظر عن الحالات التي تمر بها الدارة، سنحاول فيما يلي اختصار عدد حالات الدارة شرط ألا يؤثر ذلك على تتابعات الدخل-الخرج والحصول على دارة مكافئة بعدد حالات أقل. بمعنى إننا سنحاول الحصول على دارة ثانية بتركيبية أبسط مكافئة للدارة الأولى، بحيث إذا طبقنا تتابعات دخل محدد على الدارتين، نحصل على نفس تتابعات الخرج. ولإظهار خوارزمية إنقاص عدد الحالات، يفضل الانطلاق من جدول الحالة للدارة عوضاً عن التعامل مع مخطط الحالة. وللمباشرة باختصار الدارة، نعرف بداية مفهوم التكافؤ بالنسبة لحالات الدارة. ويقال عن حالتين من مخطط حالة إنهما متكافئتان، إذا أعطت كل منهما لكل عنصر في مجموعة المداخل نفس الخرج، ونقلتا الدارة إلى نفس الحالة اللاحقة. أما خوارزمية الاختصار فتتلخص بالتالي: يتم البحث بكل خطوة عن الحالات المتكافئة ويتم حذف إحداها. ويستمر هذا المسار حتى يتم الوصول إلى الوضع الذي لا يمكن فيه إيجاد حالات متكافئة، فيكون الجدول الناتج هو الجدول الذي يقابل أبسط مخطط حالة للدارة. ويمكن من آخر مخطط حالة يتم التوصل إليه تحديد عدد القلايات اللازم، وبالتالي الشكل المختصر للدارة. وجدول الحالة المقابل لمخطط الحالة في الشكل 12-5 يمكن تشكيله، وفق الجدول 12-2.

الجدول 12-2: جدول الحالة المقابل لمخطط الحالة بالشكل 12-5

الحالة الراهنة	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
q	Q	Q	y	Y
q ₀	Q ₁	Q ₂	1	1
q ₁	Q ₄	Q ₂	0	0
q ₂	Q ₅	Q ₂	1	0
q ₃	Q ₁	Q ₄	1	1
q ₄	Q ₃	Q ₂	1	0
q ₅	Q ₁	Q ₄	1	1

ويتضح من الجدول أن كلاً من الحالتين الراهنتين q_3 و q_5 تذهبان إلى نفس الحالة اللاحقة لكل من تركيبيتي الدخل، وتعطيان نفس الخرج (عندما تكون الدارة بالحالة q_3 وتتم قراءة $x = 0$ تنتقل الدارة للحالة الجديدة q_1 ويكون الخرج $y = 1$ وهكذا). أي أن هاتين الحالتين متكافئتان، وبالتالي يمكن بالخطوة التالية حذف إحداهما (ولتكن q_5)، وفق الجدول 12-3.

الجدول 12-3: الشكل المعدل لجدول الحالة بالجدول 12-2 بدون الحالة q_5

الحالة الراهنة	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
q	Q	Q	y	y
q_0	Q_1	Q_2	1	1
q_1	Q_4	Q_2	0	0
q_2	Q_3	Q_2	1	0
q_3	Q_1	Q_4	1	1
q_4	Q_3	Q_2	1	0

لأن حذف إحداها لن يؤثر على نتابعات الدخل - الخرج. ويشكل جدول حالة جديد من دون السطر الذي يضم الحالة q_5 وبنفس الوقت أينما ترد q_5 في الجدول يتم استبدالها بـ q_3 المكافئة. ويلاحظ من الشكل الجديد للجدول أيضاً أن الحالتين q_2 و q_4 أصبحتا متكافئتين لنفس الأسباب المبينة أعلاه، وبالتالي يمكن حذف إحداهما (ولتكن الحالة q_4) وتشكيل شكل جديد للجدول بنفس الشروط المبينة أعلاه، فنحصل على جدول بعدد أسطر أقل بسطر واحد ومن دون أي وجود للحالة q_4 وفق ما هو مبين في الجدول 12-4.

الجدول 12-4: الشكل المعدل لجدول الحالة بالجدول 12-3 بدون الحالة q_4

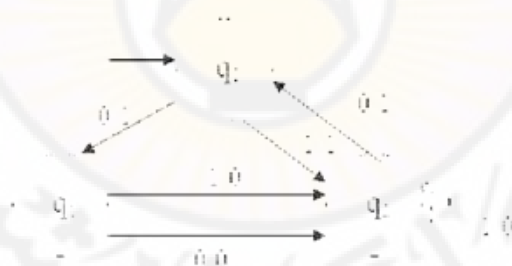
الحالة الراهنة	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
q	Q	Q	y	y
q_0	Q_1	Q_2	1	1
q_1	Q_2	Q_2	0	0
q_2	Q_3	Q_2	1	0
q_3	Q_1	Q_2	1	1

حيث يلاحظ من الجدول أيضاً أن الحالتين q_0 و q_3 أصبحتا متكافئتين، يحذف إحداهما ولتكن الحالة q_3 ، فنحصل على جدول حالة وفق ما هو مبين في الجدول 12-5.

الجدول 12-5: الشكل المعدل لجدول الحالة بالجدول 12-4 بدون الحالة q_3

الحالة الراهنة	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
q	Q	Q	y	y
q_0	Q_1	Q_2	1	1
q_1	Q_2	Q_2	0	0
q_2	Q_0	Q_2	1	0

ويمثل الجدول الأخير أبسط شكل يمكن الحصول عليه بالنسبة للمثال المعطى، لأنه لم يعد يوجد حالات متكافئة، وبالتالي هذا الجدول هو الذي يقابل أبسط شكل ممكن للدائرة المعطاة. ويلاحظ من مخطط الحالة الجديد انخفاض عدد حالات الدائرة المعطاة من ست إلى ثلاث حالات فقط من دون أن يؤثر ذلك على تتابعات الدخل-الخرج. ويصبح مخطط الحالة المكافئ للمخطط الاساسي المعطى، وفق ما هو مبين في الشكل أدناه.



وهذا يعني أنه يكفي للشكل المختصر للدائرة المعطاة قلابين فقط بستة تنقلات. وللتأكد من صحة عملية الاختصار، يمكن إعادة تطبيق نفس الكلمة التي طبقت على الدائرة الأساسية قبل تخفيض عدد الحالات، أي السلسلة 011001000 على الدائرة بشكلها الجديد، فنجد أن:

State : q_0 q_1 q_2 q_2 q_0 q_1 q_2 q_0 q_1 q_2
 Input : 0 1 1 0 0 1 0 0 0 0
 Output : 1 0 0 1 1 0 1 1 0

وهو نفس الخرج بالنسبة لنفس الدخل. ويبقى أن نشير هنا إلى إن إنقاص عدد الحالات في

جدول الحالة العائد لدارة تتابعية، يؤدي إلى دارة بعدد عناصر وتقلات أقل، لكن ليس بالضرورة أن يؤدي اختصار عدد الحالات إلى توفير بعدد البوابات التركيبية. ويتداخل هذا الموضوع المتشعب مع مواضيع أخرى تتعلق بمقرر آخر حول " نظرية الحسابات Computation Theory " (أو اللغات الصورية) التي تُعدّ مدخلاً لتصميم المترجمات Compilers، ولا مجال هنا للدخول بمزيد من التفاصيل حولها. وأخيراً بعد أن يتم إنقاص عدد الحالات لدارة، يتم الانتقال إلى مرحلة تحقيقها بمكوناتها الفيزيائية، وهذا يعني إسناد قيمة مرمزة لكل حالة من حالات الدارة. وبما أنه تمّ بالمثال المعطى تقيص عدد الحالات من ست حالات إلى ثلاث يكفي 2 bits لترميز حالات الدارة. من طرف آخر 2 bits قادرة على ترميز $2^2 = 4$ حالة، وهذا يعني أن حالة رابعة ستبقى غير مستخدمة، تعامل على أنها حالة عدم اكتراث. ويمكن اعتماد أي من طرق الترميز التي سبق ومرت معنا بالفصل الثالث باستخدام الرماز المناسب. وبالطبع يوجد طرق أخرى لترميز الحالات لم يشر إليها سابقاً، مثل ترميز "موجة الواحد". وفي حال استخدام مثل هذا الترميز، يجب التقيد بوجود بت واحد بقيمة الواحد في كل تركيبة، بحيث ينتقل هذا الواحد من أدنى خانة حتى أعلى خانة بالتدريج مشكلاً موجة من الوحدات. يلزم وفق هذا الترميز عدد من البتات يساوي عدد حالات الدارة.

12-1-2 تصميم الدارات التتابعية المتزامنة

يتم تحديد الدارة التركيبية بوساطة جدول الحقيقة، ويجري تحديد مجموعة الخصائص المميزة للدارة التتابعية المتزامنة من جدول الحالة للدارة. وبالتالي الخطوة الأولى في تصميم الدارات المنطقية التتابعية المتزامنة هي الحصول على جدول الحالة أو أي تمثيل آخر مكافئ له مثل مخطط الحالة. وبما أن الدارة التتابعية تتكون من قلابات وبوابات منطقية، يقوم تصميم الدارة التتابعية التي تحقق مواصفات محددة باختيار نوع القلابات اللازمة وعددها، وإيجاد الجزء التركيبي للدارة (إن كان موجوداً). ويستخرج الجزء التركيبي عادة من جدول الحالة (عن طريق تقييم معادلات القلابات ومعادلات الخرج). أما عدد القلابات اللازم فيحدّد من عدد الحالات التي تحتاجها الدارة. وبالتالي بعد تحديد عدد ونوع القلابات اللازمة تنتقل عملية التصميم من مسألة تصميم دارة تتابعية إلى مسألة تصميم دارة تركيبية.

يمكن تلخيص إجرائية تصميم الدارة التتابعية المتزامنة بالخطوات التالية:

1- إنشاء مخطط الحالة (بناء على الوصف الكلامي ومواصفات العملية المطلوبة).

2- اختصار عدد الحالات إن كان ذلك ممكناً.

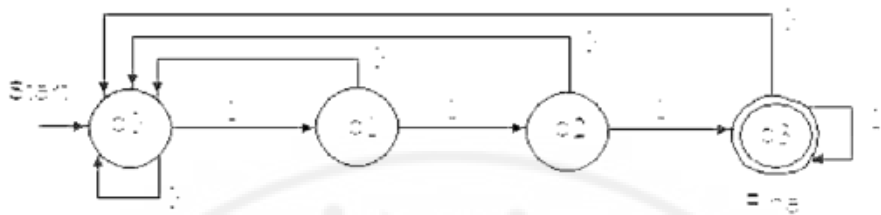
3- إسناد قيم ثنائية لكل حالة وإعادة تشكيل جدول الحالة من جديد مرمز ثنائياً.

4- اختيار نوع القلابات وإيجاد المعادلات المبسطة لمداخل القلابات والمخارج.

5- رسم المخطط المنطقي الذي يمثل الهدف من عملية التصميم.

وباستثناء الخطوة الأولى التي لا يمكن تحديدها بشكل مسبق، يمكن تجزئة بقية الخطوات وأتمتها آلياً عن طريق خوارزمية دقيقة. ويشار عادة إلى جزء التصميم الذي يتبع إجرائية معروفة " بالتركيب Synthesis". وفيما يلي نعرض مثلاً، نبين من خلاله آلية التصميم. مثال: صمم دائرة منطقية تتابعية متزامنة (أوت وميتون Automaton)، تعمل على كشف ورود ثلاث واحدات متتالية أو أكثر من شريط الدخل.

الجواب: تكون البداية من التوصيف الكلامي لما هو مطلوب ببناء جدول الحالة المقابل، بحيث يرمز لكل حالة بدائرة يكتب بها اسم أو رقم الحالة. ويتم الانطلاق من حالة بدائية، ولتكن q_0 وتكتب قيمة الدخل على الخط الواصل بين حالة وأخرى. ويتم في كل مرة التساؤل وفق التالي: إذا كانت الدارة بالحالة q_0 ، وكان أول بت في شريط الدخل 0 يفترض أن تبقى الدارة بالحالة q_0 ، لأن الصفر غير مقبول. أما إذا كانت قيمة أول دخل 1 فيفترض بالدارة أن تنتقل إلى حالة تالية ولتكن q_1 ، مشيرة بذلك إلى أنه تم كشف أول 1 من السلسلة المطلوبة. وهكذا بعد أن تصبح الدارة بالحالة q_1 ، نسأل من جديد، هل القيمة التالية من شريط الدخل هي 0؟ وإذا كان الجواب نعم يفترض بالدارة أن تعود للحالة q_0 ، لأن الكلمة المدخلة مرفوضة وفق ما هو مطلوب من الدارة. أما إذا كان الجواب كلا، فيفترض بالدارة أن تنتقل إلى حالة تالية، ولتكن q_2 للدلالة على كشف وصول واحدتين على التوالي. وهكذا تستمر المناقشة وتمثيل التوصيف الكلامي للمسألة بمخطط الحالة المبين في الشكل أدناه (الذي يمثل مخطط حالة لدارة تتابعية من نوع مور).



يمثل مخطط الحالة أعلاه دورة تنبؤية من نوع مور لعدم وجود مخارج أثناء التنقل من حالة لأخرى (الخرج الوحيد هو 1 عندما تكون الدارة بالحالة q_3). وبعد أن تم استنتاج (اشتقاق) مخطط الحالة، يتبع بالمرحل المتبقية من التصميم إجرائية تركيب مباشر باستخدام أي من القلايات المتوافرة. وفيما يلي نظهر من خلال هذا المثال خطوات التصميم المبينة أعلاه خطوة بخطوة. وبالنظر لعدم إمكانية اختصار حالات الدارة، ننقل إلى الخطوة الثالثة، أي تسنيد ترميزات ثنائية إلى حالات الدارة وإدراجها في جدول لتشكيل جدول الحالة المقابل. ويمكن استنتاج جدول الحالة من مخطط الحالة، وفق الجدول 12-6.

الجدول 12-6: جدول يبين استخراج جدول الحالة من مخطط الحالة

States	Inputs	
	0	1
q_0	Q_0	Q_1
q_1	Q_0	Q_2
q_2	Q_0	Q_3
q_3	Q_0	Q_3

تضم الدارة التي تم التوصل إليها بحسب جدول الحالة أربع حالات $q_0, \dots, q_3$. ويمكن ترميز هذه الحالات بمحولين فقط بالتركييب: 11, ..., 00. لفرمز لهذين المتحولين بالرمزين A و B التي تمثل أسماء القلايين المطلوبين. يمكن اختيار أي نوع من القلايات، وسنرمز لخرجيهما Q_A, Q_B . ونظهر فيما يلي تكملة التصميم باستخدام نوعين من القلايات كلاً على حدة وفق التالي:

أولاً: تركيب الدارة باستخدام قلابات D،

يظهر من مخطط الحالة للدارة أن للدارة مدخلاً وحيداً x (شريط الدخل) ومخرجاً وحيداً y (من حالة القبول q_3). ننطلق عند استخدام قلابات من نوع D من المعادلة المميزة للقلاب، أي $Q_n = D$. ويعدل جدول الحالة إلى الشكل المبين في الجدول 12-7.

الجدول 12-7: جدول حالة بوجود الدخل x والخرج y

الحالة الراهنة	الدخل	الحالة التالية	الخرج
q_p	x	Q_n	y
q_0	0	Q_0	0
q_0	1	Q_1	0
q_1	0	Q_0	0
q_1	1	Q_2	0
q_2	0	Q_0	0
q_2	1	Q_3	0
q_3	0	Q_0	1
q_3	1	Q_3	1

يظهر الجدول 12-8 ترميز جدول الحالة أعلاه، ويمكن من هذا الجدول الأخير إيجاد معادلات مداخل القلابات مباشرة من أعمدة الحالة التالية لكل من القلابين A و B، والتعبير عنها بصيغة مجموع حدود أصغريه وفق التالي: $Q_A = \sum (3, 5, 7)$ و $Q_B = \sum (6, 7)$

الجدول 12-8: جدول حالة مرمز ثنائياً

Dec.	الحالة الراهنة	الدخل	الحالة التالية	الخرج
	$q_A \ q_B$	x	$Q_A \ Q_B$	y
0	0 0	0	0 0	0
1	0 0	1	0 1	0
2	0 1	0	0 0	0
3	0 1	1	1 0	0
4	1 0	0	0 0	0
5	1 0	1	1 1	0
6	1 1	0	0 0	1
7	1 1	1	1 1	1

وباستخدام مخطط كارنوف لتبسيط قيمة كل من Q_B و Q_A ، نحصل على المعادلات التالية:

$$Q_A = q_A \cdot x + q_B \cdot x, \quad Q_B = q_A \cdot x + q'_B \cdot x$$

المطلوبة، وفق ما هو مبين في الشكل أدناه.

ثانياً: تركيب الدارة باستخدام قلابات J-K

بداية لابد لتحديد معادلات مداخل القلابات من استنتاج علاقة وظيفية بين جدول الحالة ومعادلات المداخل. ولهذه الغاية نظهر فيما يلي تكملة عملية التصميم باستخدام قلاب مثل J-K لإظهار كيفية استخدام جداول التحريض. ويصبح جدول الحالة المرمز للمثال المعطى باستخدام القلاب D عند استخدام J-K من الشكل المبين في الجدول 9-12.

الجدول 9-12: جدول حالة مرمز بوجود القلاب JK

الحالة الراهنة		الدخل	الحالة التالية		مداخل القلابات			
q_A	q_B	x	Q_A	Q_B	J_A	K_A	J_B	K_B
0	0	0	0	0	0	d	0	d
0	0	1	0	1	0	d	1	d
0	1	0	0	0	1	d	d	1
0	1	1	1	0	0	d	d	0
1	0	0	0	0	d	0	0	d
1	0	1	1	1	d	0	1	d
1	1	0	0	0	d	0	d	0
1	1	1	1	1	d	1	d	1

ويتم اشتقاق مداخل القلابات من جدول الحالة بالترافق مع جدول التحريض للقلاب المستخدم. فمثلاً، السطر الأول من الجدول أعلاه يفهم كالتالي: حتى ينتقل القلاب A من الحالة الراهنة صفر إلى الحالة اللاحقة صفر يفترض (وفق جدول التحريض) أن يكون المدخل $J = 0$ و $K = d$ ، وهكذا تعالج بقية الأسطر. وتبسط المعادلات بواسطة مخططات كارنوف، فنحصل على المعادلات التالية:

$$J_A = q_B \cdot x' ; \quad K_A = q_B \cdot x ; \quad J_B = x ; \quad K_B = (q_A \oplus x)'$$

وبرسم هذه المعادلات التي تمثل قيم المداخل للقلابات الداخلة في تركيبة الدارة، يمكن الحصول على المخطط المنطقي التالي:

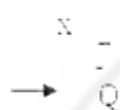
12-2 الدارات التتابعية غير المتزامنة

يعتمد سلوك الدارات التتابعية غير المتزامنة بأي لحظة على إشارات الدخل وعلى الترتيب الذي تتغير فيه، لأن هذا النوع من الدارات لا يملك مدخلاً خاصاً للساعة. وتستخدم في الدارات غير المتزامنة عناصر تخزين تكون عادة من تجهيزات تأخير زمني أو قلابات غير متزامنة. وعناصر التخزين في معظم الدارات التتابعية غير المتزامنة هي من نوع البوابات التي لا تحتاج إلى أية تجهيزات إضافية في حلقة التغذية الراجعة للدارة، لأن التأخير الناتج من مرور الإشارات بهذه البوابات يكفي في معظم التطبيقات للانتقال من الحالة الراهنة إلى الحالة اللاحقة للدارة وتخزين المعلومات. يكمن الفارق الرئيسي ما بين الدارات المتزامنة وغير المتزامنة بإمكانية وقوع الدارات التتابعية غير المتزامنة في حالات عدم الاستقرار.

يحدث تغيير الحالات الداخلية في الدارات المتتابعة غير المتزامنة بحسب تركيبة الدارة، وتتم معالجة إشارة الدخل بعد تطبيقها وتستمر فعالية الدارة من دون ارتباط مع أي تحكم خارجي. وبالتالي خلال تحليل الدارة غير المتزامنة يمكن مصادفة:

- الحالة الداخلية اللاحقة للدارة متوافقة مع الحالة الآتية، وفي هذه الحالة تبقى الدارة في حالة الاستقرار، ونعبر عن ذلك من أجل المدخل X وفق الشكل التالي،

حيث Q هنا تمثل كل من الحالة الراهنة واللاحقة للدارة.



- الحالة اللاحقة للدارة غير متطابقة مع الحالة الآتية. وبهذه الحالة نضع أثناء تحليل الدارة الحالة اللاحقة مكان الحالة الراهنة، ونتابع العبور من حالة لأخرى. وهنا يمكن أن نصادف واحدة من تركيبتين مختلفتين:

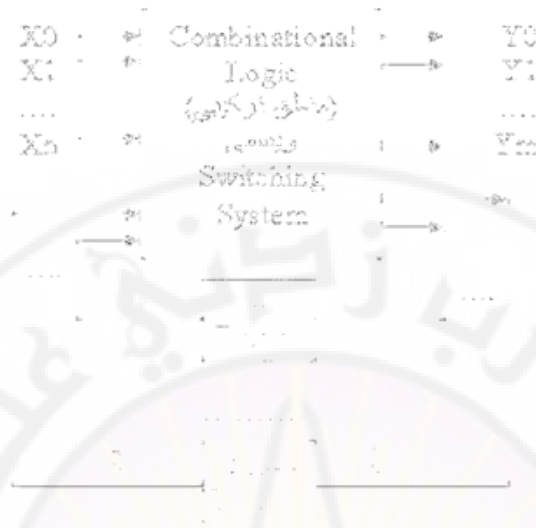
أن تصل الدارة في النهاية إلى إحدى حالات الاستقرار، كما في الشكل أدناه.



(b) أن تصل الدارة إلى حالة سبق وقد مرت بها، وفق ما هو مبين في الشكل أدناه.



ويحصل بالحالة b اهتزاز في الدارة المتتابعة، أو عدم استقرار، وهي حالة غير مطلوبة، ويطلب إيجاد حل لتجاوزها. ولتحليل الدارات المنطقية المتتابعة غير المتزامنة، ننطلق من النموذج العام للدارة المتتابعة المبين في الشكل 1-12. ويتعدّل هذا النموذج بالنسبة للدارة المتتابعة غير المتزامنة إلى النموذج المبين في الشكل 6-12 يمكن من الشكل المعدّل ملاحظة أن الدارة المتتابعة غير المتزامنة تتكون من دارة تركيبية بتغذية خلفية. وهنا نميز بين مدخل عنصر التأخير الخارج من الجزء التركيبي للدارة (الذي يمثل الحالة اللاحقة للدارة) والذي يرمز له بـ Q ويسمى بمتحول التحريض Excitation variables ومخرج عنصر التأخير الداخل إلى الجزء التركيبي للدارة والذي يرمز له بـ q ويسمى بمتحول ثانوي Secondary variable.



الشكل 6-12: النموذج العام للدارة التتابعية غير المتزامنة

عند تغير متحول الدخل x ، فإن المتحولات الثانوية q لا تتغير بنفس اللحظة، وإنما تحتاج إلى مدة معينة تكفي لانتشار الإشارة من مدخل الجزء التركيبي للدارة إلى متحولات التحريض لتشكل قيماً جديدة للحالة التالية، وهو ما نعتبر عنه بانتقال الدارة من حالة إلى أخرى والذي يتم بناء على تغير قيمة الدخل من دون ارتباط مع الساعية. ولضمان صحة عمل الدارة غير المتزامنة، يجب تأمين وصول الدارة إلى حالة مستقرة قبل حدوث تغير جديد في قيمة الدخل، لذلك يمنع عادة في الدارات غير المتزامنة تغير متحولين أو أكثر في اللحظة ذاتها. بمعنى أن الزمن الفاصل بين تغير مدخلين يجب أن يكون أكبر من الزمن الذي تستغرقه الدارة للوصول إلى الحالة المستقرة، وهذا ما يعرف بالنمط الأساسي لتشغيل الدارة غير المتزامنة والذي يفترض فيه أن إشارة واحدة من إشارات الدخل تتغير في لحظة ما عندما تكون الدارة في وضع مستقر. ولهذه الاعتبارات المتعلقة بالتوقيت التي لا تصادفها في الدارات التتابعية المتزامنة، يُعد تحليل الدارات التتابعية غير المتزامنة وتصميمها أصعب بكثير من تحليل الدارات التتابعية المتزامنة وتصميمها. ويتضمن تحليل الدارات التتابعية غير المتزامنة الحصول على جدول أو مخطط يصف تتابع الحالات الداخلية والمخارج باعتبارها توابع لمتحولات الدخل.

- نميز بين نوعين من المخططات المنطقية للدارة التتابعية غير المتزامنة هما:
- مخطط يحتوي على حلقة أو أكثر من حلقات التغذية الخلفية من دون قلابات.
- مخطط يحتوي على قلابات غير متزامنة.
- وفيما يلي نبين من خلال الأمثلة آلية تحليل كل من هذين النوعين على حدة.

12-2-1 تحليل الدارات التتابعية غير المتزامنة بوجود التغذية الخلفية

تتضمن إجرائية التحليل Analysis procedure بالنسبة للدارات التتابعية غير المتزامنة التي تستخدم حلقات التغذية الخلفية بناء كل مما يلي ودراسته.

- جدول الانتقال Transition table.
- جدول الانسياب Flow table.
- شروط السباق Race condition.
- اعتبارات الاستقرار Stability consideration.

أولاً- جدول الانتقال Transition table

لإظهار مفهوم ودور جدول الانتقال في تحليل الدارات التتابعية غير المتزامنة، ننتقل من مثال محدّد عن دارة تتابعية غير متزامنة بمدخل واحد x وحالتين داخليتين q_1 و q_2 ، وتضم بوابات منطقية، وفق ما هو مبين في الشكل أدناه.



ويمكن عند الحصول على جدول الانتقال تحليل سلوك الدارة بمراقبة انتقال الحالة باعتبارها تابعاً لتغيرات متحولات الدخل. والإجرائية الواجب اتباعها للحصول على جدول الانتقال تنطلق من مخطط الدارة المعطى، وفق التالي:

1- تحديد جميع حلقات التغذية الخلفية في الدارة: يلاحظ في الشكل أعلاه وجود حلقتين للتغذية الخلفية من مخرجي OR و AND رجوعاً إلى مدخل بوابتي NAND و NOR.

2- تخصيص متحول Q_i لكل مخرج حلقة من حلقات التغذية الخلفية ومتحول موافق له q_i للدخل لكل $i = 1, 2, \dots, k$ (حيث k يمثل عدد حلقات التغذية الخلفية). يلزم الشكل أعلاه متحولاً تحريض Q_1 و Q_2 للحالات الداخلية ومتحولان ثانويان مقابلان q_1 و q_2 . ويتم الحصول على التأخير المرتبط بحلقة التغذية الخلفية من تأخير الانتشار بين الدخل q والمخرج الموافق له Q ، إذ إن كل بوابة في ممر انتشار الإشارة تسبب (إن كانت TTL) زمن تأخير يتراوح بين 15ns حتى 25 ns يضاف إلى ذلك التأخير الناتج من انتقال الإشارات الكهربائية بالأسلاك (والذي يقدر بـ 1ns لكل قدم من الأسلاك). وليس من الضروري إضافة عناصر تأخير خارجية إذا كانت الدارة التركيبية والأسلاك المستخدمة بممر التغذية الخلفية تؤمنان التأخير الكافي.

3- اشتقاق التوابع المنطقية لجميع المخارج Q باعتبارها توابع للمداخل الخارجية ولجميع المداخل q_i . وهنا يبدأ تحليل الدارة بجعل متحولات التحريض باعتبارها توابع لكل من متحولات الدخل الخارجية والمتحولات الثانوية. يمكن الحصول على هذه المعادلات من المخطط المنطقي المبين في الشكل أعلاه وفق التالي:

$$Q_1 = (q_1'x)' + (q_2 + x)' = (q_1 + x) + q_1'x'$$

$$Q_2 = (q_1 + x). (q_1'x) = q_1'x$$

4- رسم جدول لكل تابع Q باستخدام المتحولات q والمداخل الخارجية x ، يسمى جدول الانتقال. وترمز بجدول الانتقال أرقام الحالات ثنائياً في الشكل الذي يحقق التجاور بين الأسطر. وتمثل الحالات السابقة q بالأسطر والمداخل x بالأعمدة. وتدمج جداول الانتقال للتوابع Q_1 و Q_2 المبينة في الجداول 10-12 (أ و ب) في جدول واحد يبين قيمة $Q = Q_1Q_2 \dots Q_k$ النهائية. ويتشكل جدول الانتقال في الجدول 10-12 ج من الجدولين (أ و ب). وبجميع القيم الرقمية الموجودة في المربعات الموافقة، بحيث البت الأول من Q يؤخذ من Q_1 والثاني من Q_2 .

الجدول 10-12: جداول الانتقال للدائرة المبينة في الشكل 7-12

(أ) جدول التابع Q1 (ب) جدول التابع Q2 (ج) جدول الانتقال

إحاطة قيمة Q ($Q = Q_1 Q_2$) بدائرة في كل مربع تتساوى فيه قيمة Q مع قيمة q من نفس السطر لتدل على شرط الاستقرار. ويظهر هنا الفرق بين كل من الدارات التتابعية المتزامنة وغير المتزامنة، حيث في الدارات المتزامنة تتحدد الحالة الراهنة تماماً بوساطة قيم القلابات، ولا تتغير بتغير المداخل في غياب نبضة الساعة، في حين أنه يمكن أن تتغير الحالة الداخلية للدائرة غير المتزامنة مباشرة بعد تغير الدخل، ولهذا السبب تدمج أحياناً الحالة الداخلية للدائرة مع قيمة الدخل لتشكل ما يعرف بالحالة الكلية للدائرة. ويظهر بجدول الانتقال بالمثال أعلاه حالتان كليتان مستقرتان $q_1 q_2 x = 100, 101$.

ثانياً- جدول الانسياب Flow table

يستخدم جدول الانسياب في تصميم الدارات التتابعية غير المتزامنة، وهو مشابه لجدول الانتقال بفارق أن ترميز الحالات الداخلية للدائرة يكون بأحرف أبجدية (بدلاً من الأعداد الثنائية)، ويتضمن أيضاً قيم خرج الدائرة. فمثلاً، بالنسبة للمثال المعطى، يلاحظ أن جدول الانتقال 10-12 ج يضم أربعة تراكيب مختلفة للحالات اللاحقة للدائرة هي بالترتيب: 00, 01, 11, 10. ويتميز هذه التراكيب بالمتحولات a, b, c, d على التوالي، يمكن الحصول على جدول الانسياب المبين في الجدول 11-12.

الجدول 12-11: جدول الانسياب المقابل لجدول الانتقال من الجدول 12-10 ج.

$q \backslash x$	0	1
a	d	c
b	a	c
c	d	d
d	d	d

ولتوضيح بعض المفاهيم الأخرى المرتبطة بجدول الانسياب نعرض فيما يلي مثالين:
 الأول- يكون جدول الانسياب المقابل لدارة معطاة بجدول الانتقال المبين بالجدول 12-12 أ
 مثل الجدول 12-12 ب، حيث يمكن أن يكون بسطر جدول الانسياب أكثر من حالة مستقرة.

الجدول 12-12: جدول الانتقال والانسياب لدارة تناظرية غير متزامنة

$q \backslash x$	0	1
0	0	1
1	1	1

$q \backslash x$	0	1
0	0	1
1	1	1

ويقصد من هذا المثال الإشارة إلى أن جدول الانسياب يسمى " بجدول الانسياب الأولي
 Primitive flow table"، إذا كان بكل سطر من أسطره توجد حالة مستقرة واحدة فقط،
 كما في الجدول 12-12.

الثاني: يمثل الحالة المعاكسة للحالة المبينة أعلاه، أي رسم المخطط المنطقي المقابل
 لجدول انسياب معطى. وتبدأ عملية رسم الدارة التناظرية غير المتزامنة التي تقابل جدول
 انسياب بتخصيص قيمة ثنائية لكل حالة في الجدول. فمثلاً، جدول الانسياب المقابل لدارة
 غير متزامنة تمر بحالتين a, b ولها مدخلان x_1, x_2 ومخرج واحد z، يمكن أن يكون كما
 هو مبين في الجدول 12-13.

الجدول 12-13: جدول انسياب يتضمن قيم خرج دارة تناظرية غير متزامنة

	00	01	11	10
0	0	0	0	0
1	0	0	1	1

يرمز للقيمة الثنائية لمتحول الخرج داخل المربع الذي يرمز للحالة اللاحقة، ويكتب بعد رمز الحالة مفصول عنه بفاصلة. ويمكن توصيف سلوك الدارة الممثلة بجدول الانسياب أعلاه كالتالي: إذا كان الدخل $x_1x_2 = 11$ مثلاً، يمكن للدارة أن تكون بالحالة a والخرج يكون 0 ويمكن أن تكون بالحالة b ويكون الخرج 1. وهكذا يمكن الانتقال بعد ذلك من التركيبة 11 إلى تركيبة الدخل 01، 10. أما الانتقال من 00 إلى 11 في اللحظة الزمنية ذاتها فلا يسمح به لأنه في النمط الأساسي لا يسمح بتغيير متحولين بأن واحد. ولرسم المخطط المنطقي المقابل للدارة الموصفة بجدول الانسياب 12-13 يلاحظ أن الدارة تمر بحالتين فقط هما a, b . بالتالي العملية تبدأ بإسناد الحالة a بالقيمة 0 والحالة b بالقيمة 1، ويتحول جدول الانسياب إلى جدول انتقال كما هو مبين في الجدول 12-14.

الجدول 12-14: الحصول على جدول الانتقال لدارة تناظرية من جدول الانسياب

	00	01	11	10
0	0	0	0	0
1	0	0	1	1

تصبح عملية رسم المخطط المنطقي من جدول الانتقال واضحة، خاصة إذا عدنا إن جدول الانتقال أعلاه يمثل مخطط كارنوف لتابع بثلاثة متحولات. ويمكن بعد الاختصار الحصول على تابع انتقال من الشكل: $Q = x_1x_2' + x_1q$. ويمكن من طرف آخر الحصول على جدول الخرج من جدول الانسياب أيضاً، بكتابة قيم الخرج المبينة بجدول الانسياب، وفق ما هو مبين في الجدول 12-15.

الجدول 15-12: الحصول على جدول الخرج لدارة تناظرية من جدول الانسياب

	00	01	11	10
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

يكون تابع الخرج المقابل للجدول أعلاه من الشكل: $y = x_1x_2q$. ودمج توابع الانتقال والخرج الناتجة بمخطط واحد، نحصل على الدارة الموصفة بجدول الانسياب أعلاه كالتالي:

يستخدم جدول التضمين Implication table لاختصار جداول الحالة والانسياب التي يتم التوصل إليها بالنسبة لتطبيق ما. ويُعدّ جدول التضمين أداة فعالة في اختصار الحالات بجدول حالة غير معرّف تماماً. وسبق أن أشرنا إلى إجرائية اختصار عدد الحالات الداخلية في الدارات التناظرية المتزامنة، وهي مشابهة لإجرائية الاختصار بالنسبة للدارات التناظرية غير المتزامنة مع بعض التعديلات. وتوجد في الدارات التناظرية غير المتزامنة حالات لا يكون فيها لزوج الحالات نفس الحالات التالية، وعلى الرغم من ذلك تذهب إلى حالات تالية متكافئة. ويمكن بواسطة جدول التضمين إذا كان عدد الحالات بجدول الحالة كبيراً فحص كل زوج من الحالات بحثاً عن احتمال وجود تكافؤ. ويمثل جدول التضمين مخططاً مؤلفاً من مجموعة من المربعات، بمعدل مربع واحد لكل زوج من الحالات، وبالتالي فهو يوفر مكاناً لكتابة أزواج الحالات المتضمنة الممكنة وتحديد كل أزواج الحالات المتكافئة. ولتوضيح الخوارزمية المعدلة لاختصار الدارات التناظرية غير المتزامنة باستخدام جدول التضمين، نشير فيما يلي إلى حالتين:

الأولى: تمثل اختصار حالات دائرة تناهية معطاة بجدول حالة معرّف تماماً. بفرض أنه تم لتطبيق ما التوصل إلى جدول الحالات المبين في الجدول 12-16.

الجدول 12-16: جدول حالة يطلب اختصاره

الحالة الراهنة	الحالة اللاحقة		الخرج	
	X = 0	X = 1	X = 0	X = 1
q	Q	Q	y	y
a	d	b	0	0
b	e	a	0	0
c	g	f	0	1
d	a	d	1	0
e	a	d	1	0
f	c	b	0	0
g	a	e	1	0

يلاحظ في الجدول أعلاه أن للحالتين a, b مثلاً ذات الخرج لذات الدخل والحالات التالية لهما هي d, e من أجل $x = 0$ و b, a من أجل $x = 1$. وإذا استطعنا أن نثبت أن الحالتين (d, e) متكافئتان، عندهما ستكون الحالتان (a, b) متكافئتين. وعندما نتحقق مثل هذه العلاقة، يقال إن زوج الحالات (a, b) تتضمن زوج الحالات (d, e). ومن خصائص الحالات المتكافئة أنه إذا تضمن الزوج (a, b) الزوج (d, e) وتضمن الزوج (d, e) الزوج (a, b) عندها يكون كلاهما متكافئاً، أي يمكن دمج a, b بحالة واحدة، وكذلك d, e تكونان متكافئتين، ويمكن دمجهما في حالة ثانية. وجدول التضمين المقابل لجدول الحالات المبين أعلاه سيتكون من ستة أسطر وستة أعمدة، وفق الجدول 12-17، حيث يبنى جدول التضمين كالتالي: يكتب على الجانب الأيسر للجدول جميع الحالات المعزفة في جدول الحالة شاقولياً باستثناء الحالة الأولى وتكتب بالترتيب من الأعلى إلى الأسفل، أي من b حتى g. ويكتب في أسفل الجدول جميع الحالات المعزفة في جدول الحالة أفقياً باستثناء الحالة الأخيرة وتكتب بالترتيب من اليسار إلى اليمين، أي من a حتى f، بحيث تظهر جميع التراكيب الممكنة من حالتين مختلفتين مع مربع يشغل تقاطع السطر مع العمود لاختبار تكافؤ الحالتين.

يوضع x في المربع الموافق للحالتين غير المتكافئتين و $\checkmark$ بالمربع الموافق للحالتين المتكافئتين.

الجدول 12-17: جدول التضمين

b	d, e $\checkmark$					
c	X	X				
d	X	X	X			
e	X	X	X	$\checkmark$		
f	c, d X	c, e X a, b	X	X	X	
g	X	X	X	d, e $\checkmark$	d, e $\checkmark$	X
	a	b	c	d	e	f

تحتوي بعض المربعات على مدخلات الحالات المتضمنة التي يجب زيادة التحري فيها لتحديد إن كانت متكافئة أم لا؟. وفيما يلي نستعرض الإجراءات المتبعة لتعبئة الجدول. ونضع في الخطوة الأولى x في كل مربع يحوي حالات لا تتساوى مخارجها لكل دخل وبناء على ذلك نجد مثلاً أن خرج الحالة c مختلف عن مخارج جميع الحالات الأخرى، لذلك يوضع x في جميع مربعات السطر والعمود بالاسم c. ويوجد في جدول التضمين تسعة مربعات أخرى من هذه الفئة. وبعد ذلك نضع في كل مربع من المربعات المتبقية أزواج الحالات المتضمنة في زوج الحالات الذي يمثل هذا المربع. وتتم تعبئة الجدول ابتداء من أعلى مربع في العمود الأيسر باتجاه الأسفل، ثم يتم الانتقال إلى العمود التالي على اليمين. فمثلاً، الزوج (a, b) يتضمن (d, e)، يسجل (d, e) في المربع المعرف بالعمود a والسطر b. وينفسر الأسلوب نتابع حتى يمثل كامل الجدول. وبما أن الحالتين (d, e) متكافئتان، لأنهما تقودان إلى ذات الحالة التالية ولهما الخرج ذاته، نضع علامة $\checkmark$ في المربع المعرف بالعمود d والسطر e ليدل بذلك على أن الحالتين متكافئتان ومستقلتان عن أي زوج متضمن. والخطوة التالية، هي إعادة فحص المربعات من جديد لمعرفة فيما إن كان يجب وضع علامات x إضافية فيها أم لا؟. توضع علامة x في المربع إذا تضمن زوجاً

واحداً متضمناً على الأقل غير متكافئ (ويشطب المربع). فمثلاً، بالمربع المعرف f, a بوضع علامة x كونه يتضمن الحالتين c, d ، لأن الزوج (c, d) يعرف مربعاً مشطوباً (يحوي العلامة x). ويكرر هذا الإجراء حتى لا تبقى مربعات إضافية تقبل وضع x . وأخيراً توضع إشارة $\sqrt{}$ في كل المربعات التي لا تحوي x ، وتكون هي المربعات التي تعرف أزواج الحالات المتكافئة. ويظهر بالمثال المبين أعلاه أن أزواج الحالات المتكافئة هي: $(a, b), (d, e), (d, g), (e, g)$. بعد معرفة أزواج الحالات المتكافئة يتم دمج الحالات في مجموعات أوسع إن أمكن ذلك. ويمكن دمج الحالات الثلاث الأخيرة في مجموعة من ثلاث حالات متكافئة (d, e, g) ، لأن كل حالة فيها تكافئ الحالتين الأخيرتين في المجموعة. أما التقسيم الأخير للحالات فيتألف من الحالات المتكافئة التي تم إيجادها في جدول التضمين، وجميع الحالات المتبقية في جدول الحالة التي لا تكافئ أيّاً من الحالات الأخرى، أي: $(a, b), (d, e, g), (c), (f)$. وهذا يعني أن جدول الحالة السابق المكون من سبع حالات يمكن اختزاله إلى أربع حالات، تقابل كل منها عضواً من التصنيف المبين أعلاه. ويتم ذلك بإبدال الحالة b بالحالة a وكل من الحالتين d, e بالحالة d ، وفق ما هو مبين في الجدول 12-18.

الجدول 12-18: جدول حالة مختزل من سبع إلى أربع حالات

الحالة الراهنة	الحالة اللاحقة		الخروج	
	$X = 0$	$X = 1$	$X = 0$	$X = 1$
q	Q	Q	y	Y
a	d	a	0	0
c	d	f	0	1
d	a	d	1	0
f	c	a	0	0

الثانية: تمثل اختصار حالات دائرة تناظرية معطاة بجدول حالة غير معرف تماماً. وتوجد حالات يكون فيها جدول الحالة للدائرة التناظرية غير معرف تماماً، وهذا يحدث عندما يكون من غير الممكن حدوث بعض تراكيب الدخل أو تتابعاته نتيجة قيود داخلية أو خارجية. وفي هذه الحالة، تكون الحالات التالية والمخارج التي كان من المفروض حدوثها

لو كانت جميع تراكيب الدخل مسموحة هي حالات عدم اكتراث. وبالتبع يمكن أحياناً تمثيل الدارات التتابعية المتزامنة بجداول حالة غير معرفة تماماً، لكن هنا سنشير فقط للدارات التتابعية غير المتزامنة. ويمكن دمج الحالات غير المعرفة تماماً لاختصار عدد الحالات في جدول الانسياب، ولكن لا يمكن اعتبار مثل هذه الحالات متكافئة، لأن تعريف التكافؤ يتطلب أن تكون المخارج والحالات التالية محددة لجميع المداخل.

ثالثاً- شروط السباق Race conditions

نظراً لعدم تساوي فترات التأخير بحلقات التغذية الخلفية في الدارات التتابعية غير المتزامنة بسبب اختلاف أطوال المسارات التي تسلكها الإشارات بالدارة، تحدث حالات سباق Race تؤدي بدورها إلى تغيير متحولات الحالة بطريقة لا يمكن التنبؤ بها بشكل مسبق. وبما أن جدول الانتقال للمثال المعطى أعلاه لا يظهر حالات سباق، نورد فيما يلي أمثلة أخرى تظهر حالات سباق، لنبين من خلالها مفهوم السباق بالنسبة للدارات التتابعية غير المتزامنة. وليكن معطى دارة تتابعية غير متزامنة بمخطط حالة يضم أربع حالات، وفق ما هو مبين في الشكل أدناه (حول أوت وميتون محدد Deterministic FA معرف فوق أبجدية الدخل $\{0, 1\}$).



وبترميز حالات الدارة q_0, q_1, q_2, q_3 بشكل ثنائي وفق التراكيب 00, 01, 10, 11 على التوالي، يمكن الحصول على جدول الانتقال المبين في الشكل أدناه. يظهر في الجدول أدناه للدارة المعطاة حالتا استقرار (الحالات التي فيها السهم من أجل دخل معين ينطلق من الحالة ويعود لنفس الحالة، وتسمى هذه العملية Closure). وتنتج شروط السباق في الدارة من تغيير قيمة متحولين أو أكثر من متحولات الحالة استجابة لتغيير أحد متحولات الدخل.

فمثلاً عند الانتقال من الحالة $Q = q_1q_2 = 00$ من أجل $x = 0$ إلى الحالة $Q = q_1q_2 = 11$ من أجل $x = 1$ قد نصادف بسبب التأخيرات المتباينة في التغذية الخلفية للدائرة الحالات التالية:

- أن يتغير المتحول الأول q_1 بشكل أسرع من تغير المتحول الثاني q_2 ، وبالتالي تتابع تغير المتحولات سيصبح من الشكل: $00 \rightarrow 10 \rightarrow 11$
 - أن يتغير المتحول الثاني q_2 بشكل أسرع من تغير المتحول الأول q_1 ، وبالتالي تتابع تغير المتحولات سيصبح من الشكل: $00 \rightarrow 01 \rightarrow 11$. وبالنظر إلى أن الترتيب الذي تتغير فيه متحولات الحالة قد لا يكون معروفاً بشكل مسبق نميز بين نوعين من السباق هما:
 الأول: إذا كانت الحالة المستقرة التي تصل إليها الدارة لا تعتمد على الترتيب الذي

تتغير فيه متحولات الحالة، فإن السباق يكون غير حرج No critical race
 الثاني: إذا كانت الحالة المستقرة النهائية التي تصل إليها الدارة تعتمد على الترتيب الذي تتغير فيه متحولات الحالة (أي إذا كان من الممكن الانتهاء إلى حالتين مستقرتين أو أكثر بحسب الترتيب الذي يحدث فيه تغيرات متحولات الحالة)، فإن السباق يكون حرجاً. ويمثل المثال أعلاه حالة سباق غير حرجة، لأنه يبدأ بالحالة الكلية $q_1q_2x = 000$ ، ويتغير متحول الدخل x من 0 إلى 1، تنتقل الدارة من الحالة 00 إلى الحالة 11، وهنا توجد حالة سباق، لأنه يوجد ثلاث إمكانيات يمكن أن تتغير فيها المتحولات هي:

- إما أن تتغير أنيا معاً من $00 \rightarrow 11$
 - وإما أن تتغير وفق التتابع $00 \rightarrow 10 \rightarrow 11$
 - وإما أن تتغير وفق التتابع $00 \rightarrow 01 \rightarrow 11$
- تنتهي الدارة بالحالات الثلاث المبينة أعلاه بنفس الحالة المستقرة، وبذلك السباق غير حرج، حيث الدارة تستقر بالإمكانيات الثلاث بالحالة النهائية $q_1q_2x = 111$. ويمكن أن نأخذ مثلاً آخر لدارة بجدول انتقال يضم ثلاث حالات استقرار وفق الجدول 12-19.

الجدول 12-19: جدول انتقال لدارة تتابعية بثلاث حالات استقرار وسباقات حرجة

- يلاحظ في الدارة سباقات حرجة للاعتبارات التالية:
- إذا بدأت الدارة من الحالة الكلية المستقرة $q_1q_2x = 000$ فإنه مع تغير الدخل x من 0 إلى 1 تنتقل الدارة إلى واحدة من الحالات التالية:
- أن تتغير متحولات الحالة من $00 \rightarrow 11$ بنفس اللحظة، وبهذه الحالة تستقر الدارة بالحالة المستقرة 11 .
 - أو أن تتغير $00 \rightarrow 01 \rightarrow 11$ إذا كان تغير q_2 أسرع من تغير q_1 ، وتستقر الدارة بالحالة المستقرة 11.
 - أو أن تتغير وفق التسلسل $00 \rightarrow 10 \rightarrow 10$ إذا كان تغير q_1 أسرع من تغير q_2 وتستقر الدارة بالحالة المستقرة 10، وهي حالة مغايرة للحالة السابقة التي استقرت بها نفس الدارة، لذلك السباق الحرج، لأن الدارة تذهب إلى أكثر من حالة مستقرة مختلفة عن بعضها بحسب ترتيب التتابع الذي تتغير وفقه متحولات الحالة، حيث الانتقال الأول والثاني يفقدان الدارة إلى الحالة الكلية المستقرة 111، بينما الانتقال الثالث يقودها إلى حالة كلية مستقرة

أيضاً هي الحالة 101. وكي تعمل الدارة بشكل صحيح يجب عدم السماح بحدوث السباقات الحرة فيها.

ولتجنب حدوث حالات السباق الحرجة، يجب توجيه الدارة للعبور عبر حالات بينية غير مستقرة بحيث يتغير فيها متحول حالة وحيد. فمثلاً، بالنسبة للدارة المعطاة بالمثال أعلاه، يمكن توجيه الدارة أثناء الانتقال من الحالة 00 عند تغير المتحول x من 0 إلى 1 إلى حالة بينية غير مستقرة مثل الحالة 01 شرط أن تؤدي هذه الحالة غير المستقرة في النهاية إلى نفس الحالة المستقرة للدارة. ويقال عن الدارة عندما تنتقل عبر تتابع وحيد من الحالات غير المستقرة إنها تمر بدورة Cycle. وبالطبع يجب توخي الحذر عند استخدام دورة تنتهي في حالة مستقرة. أما إذا لم تنتهِ الدورة بحالة مستقرة فإن الدارة تواصل التنقل من حالة غير مستقرة إلى أخرى غير مستقرة، وهكذا دواليك، جاعلة بذلك الدارة بكاملها غير مستقرة كما هو الحال بالنسبة لدارة معطاة بجدول انتقال مثل الجدول 20-12، حيث الدارة بعد انتقالها من الحالة المستقرة 00 ووصولها إلى الحالة غير المستقرة 01 بفعل تغير الدخل x من القيمة 0 إلى القيمة 1 تواصل التنقل بشكل متكرر من 00 إلى 10 عبر 11 باستمرار.

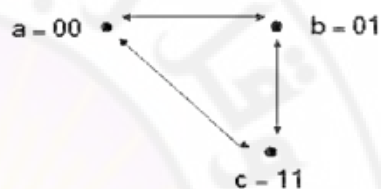
الجدول 20-12: جدول انتقال لدارة غير مستقرة

يمكن تجنب حالات السباق أيضاً بإسناد ترميزات ثنائية إلى متحولات الحالة بطريقة مناسبة كأن تخصص أعداد ثنائية لمتحولات الحالة، بحيث يتغير متحول واحد فقط عند الانتقال من حالة إلى أخرى بجدول الانسياب. وتتعلق طريقة إسناد خال من السباق من جدول الانسياب للدائرة بتخصيص متحولات ثنائية لكل حالة مستقرة وصولاً إلى جدول الانتقال المكافئ. وتعتمد الطريقة في أثناء إعطاء الترميزات لحالات الدارة على إعطاء ترميزات متجاورة

للحالات التي تحدث بها انتقالات. وبما أن تخصيص متحول وحيد في جدول انسياب يحوي سطرين لا يسبب مشكلة سباق خرج، نعلم مثال آخر بجدول انسياب يتطلب تخصيص متحولين ثنائيين، ويحتوي ثلاثة أسطر، وفق الشكل 12-7 أ. يظهر في الشكل 12-7 ب مخطط الانتقال الذي يمثل رسوماً جميع التغيرات بين السطور في جدول الانسياب.

x_1x_2		00	01	11	10
q					
a		a	b	c	a
b		a	b	b	c
c		a	c	c	c

(أ) جدول الانسياب

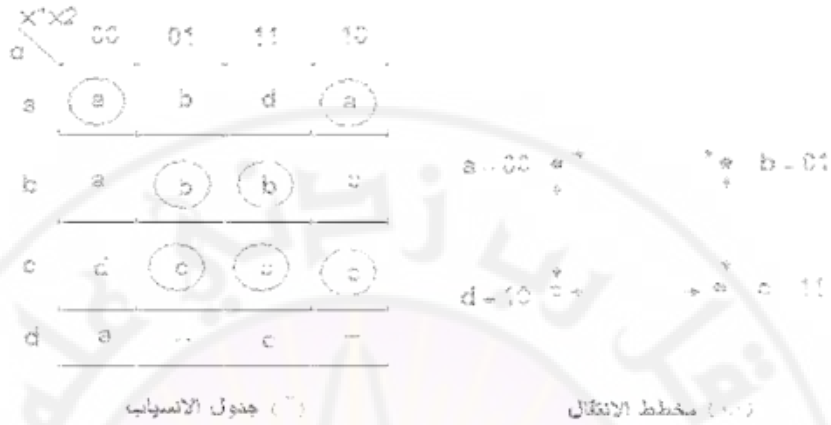


(ب) مخطط الانتقال

الشكل 12-7: جدول انسياب بثلاثة أسطر

باعتدال رماز كري بترميز حالات الجدول للمحافظة على التجاور، يتم تخصيص الحالات a, b, c بالترميزات $00, 01, 11$ على التوالي. وبسبب هذا التخصيص المعتمد سيتم سباق خرج عند الانتقال من الحالة a إلى c ، عندما يكون الدخل $x_1x_2 = 11$ (السطر الأول، العمود الثالث في الجدول أعلاه) نظراً لتغير المتحولين الثنائيين للحالة بأن واحد، والانتقال إلى حالتين مستقرتين متباينتين. ويوجد شرط سباق آخر في الجدول أعلاه عند انتقال الدارة من الحالة c إلى الحالة a لكن ليس خرجاً، لأنه في الحالتين ينتهي بحالة كلية مستقرة $qx_1x_2 = 000$. وللحصول على تخصيص خالٍ من السباقات الحرجة تضاف أسطر جديدة إلى جدول الانسياب المختزل لتسمح بتشكيل دورات بين الحالات المستقرة. بمعنى أنه عوضاً عن الانتقال من الحالة a إلى c عندما يكون $x_1x_2 = 11$ ، ويمكن الانتقال إلى حالة مجاورة مثل 10 بالاسم d ، ومن ثم الانتقال من d إلى الحالة المجاورة c . وبما أن استخدام سطر رابع لا يزيد من عدد المتحولات الثنائية، ويسمح بتشكيل دورات بين حالتين مستقرتين، يمكن رسم جدول الانسياب المعدل مع مخطط الانتقال، وفق ما هو مبين في

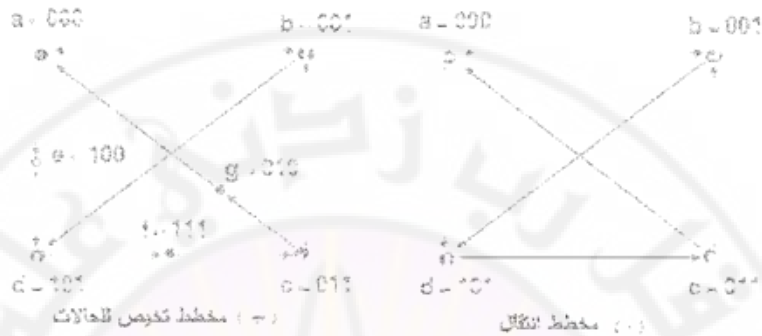
الشكل 8-12.



الشكل 8-12: جدول انسياب بسطر إضافي لدارة خالية من السباقات

يلاحظ في الجدول أعلاه إضافة سطر رابع بالاسم d واستعاضة الحالات التي كانت تسبب شرط السباق a, c بالحالة d التي خصص لها القيمة 10 في الشكل التي تجاور فيه كل من الحالتين 00, 11 المقابلتين لـ a, c على التوالي لتجنب السباق الحرج، حيث الانتقال من a إلى c يجب أن يمر بالحالة d. ولا يخصص السطر الإضافي لأي حالة محددة مستقرة. ويمثل الرمز (-) في السطر الرابع حالة غير معرفة، يمكن اعتبارها حالة عدم تعيين. وبالطبع الأسطر الثلاثة الأولى تمثل الشروط نفسها الموجودة في جدول الانسياب الأصلي ذي الأسطر الثلاثة. وباستخدام الترميزات الثنائية المشار إليها أعلاه يمكن الحصول على جدول الانتقال الموافق لجدول الانسياب الناتج. وقد لا يكون في بعض الأحيان سطرًا واحدًا كافيًا لتجنب السباقات الحرجة، وبهذه الحالة قد يكون من الضروري إضافة سطرين أو أكثر في جدول الانسياب. فمثلاً، جدول انسياب بأربعة أسطر يلزمه على الأقل متحولين للحالة، لكن متطلبات إضافة أسطر إضافية لتجنب السباق الحرج ببعض الحالات يفرض استخدام ثلاثة متحولات حالة ثنائية. فإذا كان مخطط الانتقال لدارة ما وفق الشكل 9-12 أ يحتوي على انتقال قطري واحد أو انتقالين لا يوجد طريقة لتخصيص القيم الثنائية للحالات بمتحولين ثنائيين تحقق متطلبات التجاور، إذ يلزم على الأقل ثلاثة متحولات ثنائية (إذا لم يكن هناك انتقالات في الاتجاهات القطرية من b إلى d أو من c إلى a، ويمكن إيجاد

ترميزات متجاورة للانتقالات الأربعة المتبقية). يبين الشكل 9-12 ب مخطط تخصيص للحالات يناسب أي جدول انسياب من 4 أسطر

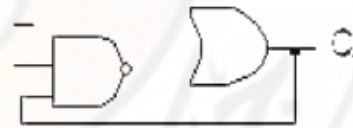


الشكل 9-12: اختيار أسطر إضافية لجدول انسياب بأربعة أسطر

يتم تخصيص الحالة b بالقيمة 001 لمجاورة الحالات الثلاث الأصلية الأخرى (a ~ 000 ، c ~ 011 ، d ~ 101). أما للانتقال من الحالة a إلى الحالة d غير المجاورة لها، فيجب أن يمر بحالة إضافية تجاور الحالتين هي الحالة e ~ 100. ولانتقال من الحالة d إلى الحالة c، يجب أن يمر بالحالة الإضافية f ~ 111 المجاورة للحالتين. وأخيراً من c إلى a يجب المرور بالحالة الإضافية g ~ 010. وباستخدام التخصيصات المبينة أعلاه يتوسع جدول الانسياب المقابل ليصبح مكوناً من سبعة أسطر، بحيث إن كل سطر يقابل حالة واحدة من الحالات السبع وترميز يقابل ترميز كري وفق الشكل: 000, 001, 011, 010, 110, 111, 101, 100 التي تقابل على التوالي الرموز التالية: a, b, c, g, f, d, e. أما التركيبة 110 فلا حاجة لها، وتوضع من باب تكملة جميع التراكيب من أجل ثلاثة متحولات، وبالتالي تقابل الرمز (-). وبالطبع الأسطر الثلاثة الإضافية تحتوي على حالات غير محاطة بدوائر، تستخدم لتوفير انتقالات بين الحالات المستقرة خالية من السباقات الحرجة. إضافة لما هو مبين أعلاه، بخصوص الاسنادات الخالية من السباق.

رابعاً - اعتبارات الاستقرار Stability consideration

يُعرف عدم استقرار دائرة تناظرية غير متزامنة بانتقال الدارة بين حالات غير مستقرة، أي اهتزاز الدارة باستمرار بين حالتين أو أكثر من دون الاستقرار بأي منها. ومن الطرائق المستخدمة في أثناء تحليل الدارة للكشف عن حدوث عدم الاستقرار استخدام جدول الانتقال. ويمكن كشف شرط عدم الاستقرار في معظم الحالات مباشرة من المخطط المنطقي. قد يكون لمفهوم عدم الاستقرار بعض التطبيقات المفيدة، مثل تصميم مولد نبضات مربعة. فمثلاً، بإجراء تعديل بسيط على الدارة التناظرية بتغيير البوابة AND إلى NAND واستخدام رموز جديدة للدخل والخرج، يمكن الحصول على دائرة جديدة وجدول انتقال جديد، وفق ما هو مبين في الشكل 10-12. وتابع التحريض الذي ينتج جدول الانتقال المبين أعلاه هو: $Q = x_1 + x'_2 + q'$. ويتضح من الجدول العائد لهذه الدارة أنه بالعمود الذي يقابل المدخلات $x_1x_2 = 01$ لا يوجد أي دائرة، وهذا يعني أنه من أجل هذا الدخل لن تتساوى الحالة اللاحقة للدارة مع الحالة السابقة، لأنه عندما تكون $q = 0$ والمدخل يساوي 01، تنتقل الدارة إلى السطر الثاني في الجدول.



الشكل 10-12: دائرة تناظرية مستقرة مع جدول الانتقال

وعندما يتم الانتقال للسطر الثاني من أجل $q = 1$ ، ستنقل الدارة إلى السطر الأول مجرد وصول الدخل 01. وهكذا باستمرار (الدارة تمر بحالة عدم استقرار من أجل دخل محدد)، أي أن الدارة لاتصل إلى أي حالة مستقرة كي تستقر بها، ما يؤدي بدوره إلى تناوب قيمة متحول الحالة مابين 1 و 0 طيلة المدة التي يكون فيها المدخل $x_1x_2 = 01$ ، أي حالة مهتز بتواتر معين ترتبط قيمته بقيمة التأخير بالتغذية الخلفية للدارة. فإذا فرضنا أن زمن تأخير انتشار الإشارة عبر كل بوابة من البوابات المشكلة للدارة 10 ns مثلاً (بما في ذلك التأخير

ضمن الأسلاك) يكون التأخير في البوابتين من أجل الحالة $Q = 0$ بحدود 20 ns، ومن أجل $Q = 1$ بحدود 20 ns، وهذا يمثل إشارة مربعة بدور 40 ns، وتواتر الدارة سيكون $f = 1/(40 \cdot 10^{-9}) = 25 \text{ MHz}$.

12-2-2 الدارات التتابعية غير المتزامنة بوجود القلايات

يمكن بناء الدارات التتابعية غير المتزامنة من دون مسارات تغذية خلفية خارجية وذلك باستخدام المواسك غير المحكومة بالساعة (علماً أنه يوجد حلقة تغذية خلفية داخل الماسك نفسه). وتكون عناصر الذاكرة في الدارات غير المتزامنة التي تتضمن المواسك واضحة. يولد استخدام المواسك في الدارات غير المتزامنة شكلاً منتظماً للمخطط المنطقي. ولكي نتمكن من تحليل الدارة غير المتزامنة باستخدام جدول الانتقال الذي يعتمد بناؤه على وجود التغذية الخلفية، ويعدل المخطط المنطقي للماسك RS غير المتزامن لإظهار ممر التغذية الخلفية (في حال استبدال الماسك RS المبني ببوابات NOR بأخر مبني بـ NAND يكون المخطط مماثلاً، ويصبح $Q = S' + Rq$ عندما $S'R' = 0$).

وباعتماد مواسك مبنية بالبوابات NOR وبالتابع المنطقي:

$SR = 0$ عندما $Q = [(S + q)' + R]' = (S + q)'' \cdot R' = SR' + R'q = S + R'q$ (للتعبير على أنه لا يسمح بتطبيق 0 على S و R في اللحظة ذاتها). يمكن أن يعبر عن جدول الانتقالات بالمخطط التالي:

SR	00	01	11	10
0	0	0	0	:
1	1	:	:	1

ولاستعراض إجرائية تحليل دائرة تستخدم الماسك SR، ننتقل من مثال محدد مثل الدارة المبينة في الشكل أدناه، والتي تتضمن ماسيكن، فتكون الخطوات كما يلي:

1- تسمية مخرج كل ماسك Q_i وممر التغذية الخلفية المرتبط به q_i بحيث $i = 1, 2$ ،
بالمثال المعطى. يوجد بالمثال المذكور مخرجان Q_1, Q_2 ومداخلان x_1, x_2 وحلقا تغذية
خلفية بالمتحولين الثانويين q_1, q_2 .

2- استخراج التوابع المنطقية للمدخلين S_i و R_i لكل ماسك i ، حيث إن عملية تحليل
الدائرة تتطلب أولاً الحصول على التوابع المنطقية للمداخل R و S لكل ماسك. وبالنظر

$$\text{للدائرة المعطاة يلاحظ أن: } S_1 = x_1 q_2 \quad \text{و} \quad R_1 = x'_1 x'_2$$

$$S_2 = x_1 x_2 \quad \text{و} \quad R_2 = x'_2 q_1$$

3- التحقق من الشرط $SR = 0$ للمواسك المبينة من البوابات NOR، لأنه في حال
عدم تحقق هذا الشرط من المحتمل أن تعمل الدارة بشكل غير صحيح. يتم الاختبار

$$\text{بالتعويض وفق التالي:} \quad S_1 R_1 = x_1 q_2 x'_1 x'_2 = 0 \quad \text{لأن } x_1 x'_1 = 0$$

$$S_2 R_2 = x_1 x_2 x'_2 q_1 = 0 \quad \text{لأن } x_2 x'_2 = 0$$

4- استخراج جداول الانتقال للدائرة بإيجاد قيمة Q لكل ماسك، وتحديد الحالات المستقرة في
الجدول بوضعها ضمن دائرة. يجب بالمثال المعطى لاستخراج جدول الانتقال، بداية

$$\text{حساب توابع التحريض من العلاقة: } Q = S + R'q \quad (\text{لأن } Q \text{ تابع لـ } q \text{ و } x)$$

$$Q_1 = S_1 + R'_1 q_1 = x_1 q_2 + (x_1 + x_2) q_1 = x_1 q_2 + x_1 q_1 + x_2 q_1$$

$$Q_2 = S_2 + R'_2 q_2 = x_1 x_2 + (x_2 + q'_1) q_2 = x_1 x_2 + x_2 q_2 + q'_1 q_2$$

يمكن تشكيل جدول مركب لـ $Q = Q_1Q_2$ كما هو مبين في الجدول أدناه، حيث تخصص الأسطر لتمثيل المتحولات الثانوية $q = q_1q_2$ الأعمدة لتمثيل $x = x_1x_2$.

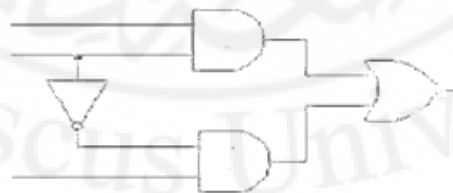
	00	01	11	10
00	○	○	○	○
01	○	○	○	○
11	○	○	○	○
10	○	○	○	○

3-2-12 Hazards المخاطر

تعرف ظاهرة الخطر Hazard بأنها حالة تحدث عندما يؤدي تغير متحول واحد بدارة منطقية إلى تغير لحظي في قيمة الخرج في الوقت الذي يجب أن تبقى قيمته ثابتة. تحدث مثل هذه المشاكل فقط في الدارات المتتابعة غير المتزامنة، بالتالي عند تصميم الدارات غير المتزامنة يجب ضمان عمل الدارة وفق النمط الأساسي الذي لا يسمح فيه بتغير أكثر من دخل واحد في اللحظة نفسها، ويجب أن تكون الدارة خالية من السباقات الحرجة.

أولاً: المخاطر في الدارات التركيبية

لإظهار المخاطر المحتمل حدوثها في الدارات التركيبية، ننتقل من دارة محددة وبسيطة مثل الدارة المبينة في الشكل 12-11.



الشكل 12-11: دارة تركيبية ذات مخاطر

بفرض أن قيم كل من المداخل الثلاثة a, b, c تساوي واحداً، هذا يجعل مخرج البوابة

AND1 مساوياً للواحد ومخرج البوابة AND2 مساوياً للصفر والمخرج النهائي للدائرة مساوياً للواحد. بفرض أن المدخل b قد تغير من 1 إلى 0 عندها يتغير خرج البوابة AND1 إلى 0 وخرج AND2 إلى 1 تاركاً قيمة خرج الدارة عند القيمة 1. ومع ذلك يمكن أن ينتقل الخرج النهائي مؤقتاً إلى القيمة 0 إذا أخذنا بالحسبان تأخير الانتشار للعاكس. يمكن أن يسبب التأخير في العاكس إلى تغير مخرج AND1 إلى 0 قبل تغير خرج البوابة AND2 إلى 1. يصبح في هذه الحالة كل من مدخلي البوابة OR لحظياً بالقيمة "0"، وهذا يسبب تغير قيمة خرج الدارة إلى القيمة "0" لمدة قصيرة (زمن تأخير انتشار إشارة b في العاكس). تتجزأ الدارة في الشكل أعلاه التابع المنطقي $Y = ab + b'c$ ويمكن أن يسبب نمط التحفيز انتقال المخرج إلى 0 في الوقت الذي ينبغي أن يبقى 1. تدعى هذه الحالة بالخطر الساكن Static-1 Hazard . يوجد بالإضافة للخطر 1 الساكن والخطر 0 الساكن، نوع ثالث هو الخطر الديناميكي Dynamic Hazard الذي يسبب تغير الخرج أكثر من مرة عندما يتغير من 0 إلى 1 أو بالعكس عندما يتغير من 1 إلى 0. تتضمن عملية إزالة الخطر ضمان عدم حدوث المخاطر الساكنة أو الديناميكية. لتوضيح ذلك ننطلق من مخطط كارنوف المقابل للتابع المعطى والذي يقابل الشكل:

		bc			
		00	01	11	10
a	0		1		
	1	0	1	1	1

$$Y = ab + b'c \quad (أ)$$

		bc			
		00	01	11	10
a	0		1		
	1	1	1	1	1

$$Y = ab + b'c + ac \quad (ب)$$

إن تغير b من 1 إلى 0 ينقل الدارة من الحد الأصغري 111 إلى الحد 101. ونظراً لأن تغير الدخل ينتج عنه حد جداء مختلف يشمل الحدين الأصغرين، يوجد خطر. الحد الأصغري 111 مغطى بحد الجداء المحقق بالبوابة AND1، والحد الأصغري 101 بحد الجداء المحقق بالبوابة AND2. كلما كان على الدارة أن تنتقل من حد إلى آخر، يكون من

المحتمل حدوث فاصل لحظي لا يكون فيه أي من الحدين مساوياً للقيمة 1 متيحاً الفرصة إلى حدوث خرج بقيمة 0 غير مرغوب فيه. العلاج لإزالة الخطر هو بجمع الحدين الأصغرين اللذين سببا المشكلة في حد جداء آخر يتقاطع مع كلتا المجموعتين ac، كما هو مبين في مخطط كارنوف (ب). ويبين الشكل 12-12 الدارة المقابلة الخالية من المخاطر التي يمكن الحصول عليها بتوليد حد الجداء ac وتنجزه ببوابة إضافية.



الشكل 12-12: دارة خالية من المخاطر

ثانياً: المخاطر في الدارات التتابعية

نكتفي هنا بدراسة المخاطر في الدارات غير المتزامنة، بالنظر إلى أن المخاطر التي تحدث بالجزء التركيبي المرتبط في الدارات التتابعية المتزامنة لا يؤخذ عادة بالحسبان، نظراً لأن الإشارات اللحظية الخاطئة غير مزعجة عموماً. بأسلوب مماثل للدارات التركيبية، لإظهار المخاطر المحتملة حدوثها في الدارات التتابعية غير المتزامنة، ننتقل من دارة محدّدة مثل الدارة المبينة في الشكل 12-13. وإذا كانت الدارة في حالة كلية مستقرة $qab = 111$ وتغير الدخل b من 1 إلى 0، ويجب أن تكون الحالة الكلية المستقرة التالية 110. لكن بسبب المخاطر يمكن للخرج q أن يتغير لحظياً إلى 0. فإذا غدت هذه الإشارة الخاطئة (تغذية خلفية) AND2 قبل أن ينتقل NOT إلى 1، فإن خرج البوابة AND2 يبقى عند 0. وتنتقل الدارة إلى الحالة الكلية المستقرة غير الصحيحة 010.



الشكل 12-13: دارة تتابعية غير متزامنة ذات مخاطر

يمكن إزالة هذا الخطر بإضافة بوابة إضافية، كما في الشكل 12-13. يوجد طريقة أخرى لتلافي المخاطر الساكنة باستخدام مواسك SR، لأن تطبيق إشارة 0 لحظية على S و R في ماسك بالبوابات NOR لا يؤثر على حالة الدارة. يوجد نوع آخر من المخاطر يسمى بالخطر الأساسي الذي يحدث نتيجة تأخيرات غير متساوية على طول مسارين أو أكثر تنطلق من نفس المدخل. يجب لتجنب المخاطر الأساسية معالجة كل حلقة من حلقات التغذية الخلفية بمفردها لنضمن أن زمن التأخير في ممر التغذية الخلفية كافٍ مقارنة بتأخيرات الإشارات الأخرى القادمة من أطراف المداخل. ولا يمكن تصحيح المخاطر الأساسية بإضافة بوابات إضافية كما هو الحال بالنسبة للمخاطر الساكنة. بل يمكن تصحيح هذه المشكلة بضبط مقدار التأخير في المسار المتأثر.

11-2-4 تصميم الدارات التتابعية غير المتزامنة

رغم تشعب الخطوات الأساسية التي تمر بها عملية تصميم الدارات التتابعية غير المتزامنة (خاصة وان البعض منها يتطلب تطبيق إجراءات منهجية) وصعوبة تصميمها بالمقارنة مع الدارات التتابعية المتزامنة، تُعد دراسة تصميم هذا النوع من الدارات مجدية عندما تكون السرعة هي المعامل الأهم، بمعنى عندما يطلب من الدارة الاستجابة بسرعة من دون الحاجة إلى الانتظار لورود النبضة الساعية. كما وتُعد الدارات التتابعية غير المتزامنة أفضل من الناحية الاقتصادية وبخاصة بالأنظمة الصغيرة التي لا تتطلب عدداً كبيراً من المكونات لعدم حاجتها لوجود مولد لنبضات الساعة الذي قد تكون تركيبته بنفس درجة تعقيد تركيب الدارة رغم عدم وجود أي دور له في المعالجة. والأهم مما سبق هو انها قد تكون مجدية أكثر في التطبيقات التي تتغير فيها إشارات الدخل بأوقات غير منتظمة وبشكل مستقل عن عامل الزمن. تبدأ عملية تصميم الدارات التتابعية غير المتزامنة مثل الدارات التتابعية المتزامنة من الوصف الكلامي للتطبيق بالحصول على جدول الانسياب الأولي (وهذا هو الجزء الأصعب في التصميم، لأنه يعتمد على الحدس والخبرة للوصول إلى التفسير الصحيح لتوصيف التطبيق). يتم بعد الوصول إلى أبسط شكل لجدول الانسياب ترميز الحالات ثنائياً للحصول على جدول الانتقال الذي يتم منه رسم المخطط المنطقي للجزء التركيبي للدارة إما مع تغذية

خلفية أو مع ماسك. نستعرض فيما يلي خطوات تصميم دارة تتابعيه غير متزامنة من خلال مثال بسيط لنبيين من خلاله بعض المفاهيم المرتبطة بهذه الخطوات، نظراً لبعض التباينات بينها وبين ما ورد بالنسبة للدارات التتابعية المتزامنة. ليكن المثال، تصميم دارة ماسك مقاد بوابات له مدخلان D للمعطيات و G للبوابة ومخرج وحيد Q ضمن الشروط التالية: يحول الماسك المعطيات على الدخل D كما هي إلى المخرج Q، طالما $G = 1$. وعند انتقال قيمة مدخل التحكم G إلى الصفر يحتفظ المخرج Q بقيمة المعطيات التي كانت على المدخل لحظة انتقال G من 1 إلى 0 وبعد ذلك لا تؤثر التغيرات التي تحصل على المدخل D بقيمة المخرج Q. الخطوة الأولى بتصميم الدارة المطلوبة بعد التوصيف الكلامي المبين أعلاه للمسألة هو بناء جدول الانسياب المناسب.

1- جدول الانسياب الأولي

سبق وأشرنا إلى ان جدول الانسياب الأولي لأي تطبيق، يمثل جدول يوجد بكل سطر من سطوره حالة مستقرة واحدة. يمكن بناء مثل هذا الجدول من جدول آخر يتضمن جميع الحالات المستقرة الكلية الممكنة للتطبيق. لبناء جدول بالحالات المستقرة الكلية للدارة، ننطلق من المبدأ العام وهو انه وفق النمط الأساسي Fundamental mode لعمل الدارات التتابعية غير المتزامنة، لا يسمح بتغيير مدخلين (أو متحولين من متحولات الدخل) بأن واحد، مثل الانتقال من التركيبة 00 إلى 11 أو من 10 إلى 01. لنبدأ بدراسة الحالات التي يكون فيها $G = 1$. بفرض اننا سننطلق من التركيبة $DG = 01$ على الدخل والتي تقابل ان $D = 0$ و $G = 1$ ، يفترض ان يكون $Q = 0$ وفق ماهو مطلوب من الدارة، لأنه عندما يكون $G = 1$ يكون $Q = D$. لنسند لهذه التركيبة أو الشرط الرمز a للإشارة إلى الحالة الداخلية الأولى للدارة. التركيبة $DG = 11$ ستقابل المخرج $Q = 1$ لنفس الأسباب وليرمز لها بالحالة b. وبما انه يمكن الانتقال من التركيبة 01 إلى 10 عبر التركيبة 00، وكذلك من التركيبة 11 إلى 00 عبر التركيبة 10 (أي الحالات التي يتغير فيها متحول دخل واحد فقط). نكمل بدراسة الحالات التي تنتقل فيها قيمة G إلى الصفر. بما ان قيمة المخرج تعتمد على القيمة الأخيرة لـ D نصادف أربعة تراكيب بمعدل اثنتان لكل من الحالتين السابقتين

هما: عندما تكون الدارة بالحالة a المقابلة للتركيبة $DG = 01$ ويتم الانتقال للتركيبة $DG = 00$ سيكون الخرج $Q = 0$ ، لأن آخر قيمة لـ D كانت صفر. لنسند لهذه التركيبة التي تأتي بعد الحالة a الرمز c، أي ان الحالة c أصبحت تقابل التركيبة $DGQ = 000$ ، لكن من التركيبة $DG = 00$ على الدخل يمكن الانتقال إلى التركيبة 10 أيضاً من أجل نفس قيمة D السابقة، أي 0 (لأن $G = 0$) وبالتالي التركيبة الجديدة بعد الحالة c هي $DGQ = 100$ ليرمز لها بالرمز d (بالتالي الحالة c بالتركيبة 000 يمكن ان تأتي بعد التركيبة 010 التي تقابل الحالة a أو بعد التركيبة 100 التي تقابل الحالة d لتغير متحول دخل واحد بكل من الحالتين). وهكذا نتابع من أجل الحالتين المتبقيتين، فنحصل على جدول الحالات الكلية للدائرة، وفق ما هو مبين بالشكل أدناه.

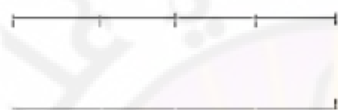
الحالة	المداخل D G	المخرج Q	تعليق
a b	0 1 1 1	0 1	$D = Q$ لأن $G = 1$
c d	0 0 1 0	0 0	بعد الحالة a أو d بعد الحالة c
e f	1 0 0 0	1 1	بعد الحالة b أو f بعد الحالة e

يظهر بكل سطر من الجدول حالة كلية واحدة تتكون من حرف يمثل الحالة الداخلية وتركيبة الدخل الممكنة من D و G ويظهر الخرج Q الموافق لكل حالة كلية. يوجد في الجدول فرق بالترميز 00 على الدخل، إذا كان الخرج 0 أو 1، حيث تعبر التركيبة الأولى عن الحالة c والتركيبة الثانية عن الحالة f. مخطط أو جدول الانسياب الأولي المقابل للجدول أعلاه يتكون من ستة أسطر وأربعة أعمدة وفق ما هو مبين في الشكل أدناه، حيث يخصص عمود لكل تركيبة دخل ومربع واحد من كل سطر لتمثيل حالة مستقرة. أما بخصوص الخرج فيكتب بعد رمز الحالة مفصلاً عنه بفاصلة ان كان موجوداً وان لم يكن موجود خرج يكتب الرمز (-). يوجد إجرائية محدّدة لتعبئة مربعات الجدول يجب التقيد بها وتتمثل بالتالي:

تبدأ عملية تعبئة جدول الانسياب الأولي بتعبئة مربع واحد من كل سطر وهو المربع الذي يقابل الحالة المستقرة بذلك السطر. تكتب هذه المعلومات بالعمود الثاني من السطر الأول بجدول الانسياب. وبنفس الأسلوب تملأ بقية المربعات الموافقة للحالات المستقرة الخمسة الأخرى مع مخارجها في أعمدة الدخل الموافقة. بعد ذلك تتم تعبئة مربع ثاني من كل سطر بالمعلومة (- ، -) التي تعني لا حالة ولا خرج، حيث يوضع خط (-) Dash بالمربع من كل سطر يختلف بمتحولين أو أكثر من متحولات الدخل المرتبطة بالحالة المستقرة. فمثلاً، في المربع الناتج من تقاطع السطر الأول بالرمز 01 مع العمود الرابع بالرمز 10 يعني تغير متحولي دخل بان واحد وهذا غير مسموح به، لذا يكتب الرمز (-). أما الخرج فيشار له بالخط (-) للدلالة على حالة عدم الاكتراث، حيث يشار إلى جميع المخارج المرتبطة بحالات غير مستقرة بخط (-) للدلالة على حالات عدم الاكتراث. بالتالي نضع خطين بالمربع الواحد بالعمود 10 من السطر a بالرمز 01 وهذا ينتج كما سنجد لاحقاً حالة عدم تعين للحالة التالية وللمخرج في هذا المربع. بقي أخيراً من الضروري إيجاد القيم المناسبة للمربعين الآخرين بكل سطر، حيث كلاً منهما سيمثل حالة غير مستقرة مع حالة عدم اكتراث للخروج. يلاحظ في المربع الأول من السطر الأول وجود المعلومة (- ، c) وهذا ناتج وفق ما هو مبين بجدول الحالات الكلية للدائرة . يمكن الوصول إلى الدخل 00 بالعمود الأول إما من تحول التركيبية 01 التي تقابل الحالة a وإما من تحول التركيبية 10 التي تقابل الحالة d. يتم الوصول إلى c بالتركيبية 00 على الدخل بعد حدوث تغير في الدخل انطلاقاً

من الحالة a أو الحالة d وهذا يقابل في جدول الانسياب ظهور حالة غير مستقرة c في العمود 00 والسطرين a و d من العمود نفسه. يجري تعيين قيم الحالات غير المستقرة في باقي مربعات الجدول بنفس الطريقة.

اختصار جدول الانسياب



(أ) حالات مرشحة للدمج

DG	00	01	11	10
a, c, d	(c), 0	(a), 0	b, -	(d), 0
b, e, f	(f), 1	a, -	(b), 1	(e), 1

(ب) الجدول المختصر (إكاثيتان)، جدول التضمين المقابل لجدول الانسياب الأولي

b	✓			
c	✓		d, e X	
d	✓		d, e X	✓
e		c, f X	✓	d, e X
f		c, f X	✓	d, e X
a				
b				
c				
d				
e				

جدولي الانتقال والخرج للماسك المقود

DG	00	01	11	10
y				
0	0	0	1	0
1	1	0	1	1

(أ)

DG	00	01	11	10
y				
0	0	0	-	0
1	1	-	1	1

(ب)

DG	00	01	11	10
y				
0	0	0	1	0
1	1	0	1	1

(ج)

12-3 أسئلة وتمارين

أولاً: أسئلة متعددة الإجابات

اختر الجواب الصحيح لكل من التساؤلات التالية:

- 1- Table that is not a part of asynchronous analysis procedure
 - a- transition table.
 - b- state table.
 - c- flow table.
 - d- excitation table.
- 2- In asynchronous circuits changes occur with change
 - a- inputs.
 - b- outputs.
 - c- clock pulses.
 - d- time.
- 3- In synchronous circuits present state is determined by
 - a- unclocked flip-flops.
 - b- clocked flip-flops.
 - c- flip-flops.
 - d- latches.
- 4- Race in which stable state depends on order is called
 - a- critical race.
 - b- identical race.
 - c- non critical race.
 - d- defined race.
- 5- Transition table that terminates in total stable state gives
 - a- sequence.
 - b- series.
 - c- unique sequence.
 - d- unique series
- 6- Effect of change of input to more than one state is called
 - a- undefined condition.
 - b- race condition.
 - c- reset condition.
 - d- ideal condition.
- 7- Asynchronous sequential logic circuits do not use
 - a- inputs.
 - b- outputs.
 - c- clock pulses.
 - d- time.
- 8- Race condition is present in
 - a- synchronous logic circuit.
 - b- ideal logic circuit.
 - c- asynchronous logic circuit.
 - d- both a and c.
- 9- First step of making transition table is
 - a- determining feedback loop.
 - b- deriving functions of Y
 - c- designating output of loops.
 - d- plotting Y.
- 10- Table having one state in each row is called
 - a- transition table.
 - b- state table.
 - c- flow table.
 - d- primitive flow table.

الفصل الثالث عشر: استخدام لغة Verilog HDL في تصميم الدارات المنطقية

يتناول هذا الفصل التعريف بإحدى لغات البرمجة لوصف العتاديات HDL (اختصاراً لـ Hardware Description Language) واستخدام مثل هذه اللغات في تصميم الدارات المنطقية واختبارها بمعونة الحاسوب قبل تجزئتها بشكل فعلي.

1-13 لغة Verilog HDL

يوجد العديد من لغات HDL المدعومة من جمعية IEEE. وفيما يلي سنشير باختصار لواحدة منها، هي لغة Verilog لتتعارف على طريقة ثانية ذات طابع برمجي لتحليل وتصميم الدارات المنطقية بمعونة الحاسب CAD (Computer Aided Design). تستخدم لغات HDL في تصميم النظم الرقمية، وهي لغات برمجية بصيغ نصية Textual Form. وتعدّ من التطورات المهمة التي طرأت على تصميم النظم الرقمية، وهي تستخدم بشكل أساسي من قبل المصممين والمنتجين للنظم الرقمية. وتشبه HDL لغات البرمجة العادية، وهي لغة وثائقية تستخدم لتمثيل النظم الرقمية بصيغة يمكن قراءتها من قبل الإنسان والحاسوب معاً. يمكن تخزين واسترداد محتوى هذه اللغة ومعالجتها بواسطة برمجيات حاسوبية. يوجد تطبيقان أساسيان للغة توصيف العتاديات هما:

- محاكاة المنطق Logic Simulation: يقصد به إمكانية تمثيل بنية النظام المنطقي باستخدام الحاسوب والتنبؤ بسلوك العتاديات قبل تصنيعها، حيث تسمح المحاكاة بكشف الأعطال الوظيفية في التصميم من دون حاجة لتنفيذ الدارة فيزيائياً.

- تركيب المنطق Logic Synthesis: يقصد به عملية استخراج قائمة المكونات وتوصيلاتها البينية Interconnection من نموذج النظام الرقمي الموصف بلغة HDL، وهو يشبه إلى حد كبير عملية ترجمة برنامج في لغات البرمجة التقليدية مع فارق أساسي، أنه بدلاً من إنتاج برنامج بلغة الآلة Object Code بالنسبة للغات عالية المستوى، ينتج قاعدة معطيات مع تعليمات عن كيفية صنع قطعة فيزيائية لإدارة رقمية تتجزأ عبارة منطقية برموز اللغة HDL.

تُعدّ اللغة Verilog HDL لغة وثائقية Documentation language، وهي من لغات البرمجة الموجهة بشكل خاص لوصف بنية العتاديات وسلوكها. ويمكن استخدامها لتمثيل المخططات المنطقية والتعابير البوليانية والدارات الرقمية. وتستخدم لغة Verilog ما يقارب الـ 100 كلمة مفتاحية Key Word وهي تتحسس لحالة الأحرف Case Sensitive، بمعنى أنه يوجد تميز بين الأحرف الكبيرة Capital Letters والصغيرة Small Letters. سيتم التعريف بلغة Verilog HDL من خلال التعريف التدريجي بأهم الكلمات المفتاحية للغة وكيفية استخدامها في توصيف الدارات المنطقية.

يعدّ المجتزأ Module اللبنة الأساسية باللغة، ويمثل كتلة بناء يعلن عنها بالكلمتين المفتاحيتين module في البداية و end module في النهاية. ويطلق (خلال عملية المحاكاة) اسم منصة اختبار Test Bench على المحفّز Stimulus الذي يختبر التصميم من الناحية الوظيفية، وهو يكتب بلغة HDL. وتسمى (خلال عملية التركيب) قائمة المكونات لنموذج النظام الرقمي الموصف بلغة HDL وتوصيلاتها البينية بقائمة الشبكة Net list وهي تستخدم على مستوى البوابة المنطقية لتصنيع الدارة المتكاملة أو لترتيب مخطط لوحة دارة مطبوعة. ولتوضيح هذه المفاهيم وغيرها من المفاهيم الأساسية المرتبطة بلغة Verilog HDL نستعرض فيما يلي مثلاً محدداً حول محاكاة عمل دارة منطقية باستخدام لغة فيريلوغ. وبالعودة للمخطط المنطقي الذي يمثل تركيبية بوابة XOR باستخدام بوابات المستوى الأول يمكن بعد إعادة ترميز المخطط من جديد، وفق ما هو مبين في الشكل أدناه.



التعبير عن هذا المخطط ومحاكاته بلغة Verilog HDL يكون وفق الشكل الآتي:

```

module ex_xor (a, b, F);
input a, b;
output F;
wire e1, e2, e3, e4;
not
g1(e1, b),
g2(e2, a);
    and
        g3(e3, e1, a),
        g4(e4, e2, b);
    or    g5(F, e3, e4);
endmodule

```

يتضح من هذا المثال البسيط أن السطر الأول يصرح عن المجتزأ بالكلمة المفتاحية **module**، وفيه يذكر أسماء المنافذ وقائمتها. بحيث الاسم (بهذا المثال ex_xor) يمثل معرفاً Identifier يمكن الرجوع إليه في التصميم. وتشكل المعرفات (التي تمثل أسماء تعطي للمتحولات) من محارف أبجدية رقمية ومن الخط التحتي (_) Underscore بحيث تبدأ بحرف أو بخط سفلي ولا يسمح أن تبدأ برقم. أما قائمة المنافذ فتشمل مأخذ الاتصال بالمحيط الخارجي، وتوضع بعد الاسم بين قوسين، وهي تمثل مداخل الدارة ومخارجها. تفصل عناصر القائمة عن بعضها بفواصل عادية وتستخدم الفاصلة المنقوطة لإنهاء العبارة. يعرف ابتداء من السطر الثاني الإعلان عن الدخل **input** والخرج **output** لتحديد أي من المنافذ هو دخل وأي منها هو خرج، بحيث تكتب هذه الكلمات المفتاحية بأحرف صغيرة حصراً وبخط غامق بقصد التمييز. ويعلن في الشكل أعلاه عن التوصيلات الداخلية بين عناصر المخطط بواسطة أسلاك Wires مثل e₁, e₂ ويعبر عنها بالكلمة المفتاحية **wire**. يتألف الإعلان عن كل بوابة من اسم اختياري، مثل: g₁, g₂, ... يليه خرج البوابة ومداخلها بين قوسين ومفصولة عن بعضها بفواصل عادية، بحيث يذكر الخرج أولاً تتبعه المداخل، كما هو موضح بالمثال أعلاه. وبالطبع أسماء البوابات الأساسية **not**, **and**, **or** تُعدّ كلمات مفتاحية أيضاً.

ينتهي وصف المجتزأ بالكلمة المفتاحية **endmodule** من دون فاصلة منقوطة. ويلزم عند استخدام لغات HDL بمحاكاة مخططات منطقية، تحديد مقدار التأخيرات الزمنية في البوابات Gate delays من الدخل إلى الخرج. وتستخدم Verilog HDL بعض الرموز الخاصة لتوضيح مثل هذه الحالات والتركيبية العامة للبرنامج، ومن الرموز الخاصة نذكر:

- الرمز // خطان مائلان: للدلالة على الملاحظة Comment، بمعنى أن أي نص يرد بعد هذا الرمز يفهم على أنه تعليق أو توضيح لا يترجم.

- الرمز # : لتحديد التأخير بدلالة وحدات زمنية Time Units، حيث يجري ربط وحدة الزمن بالزمن الحقيقي باستخدام أمر المترجم timescale.

وبالمناسبة، فإن جميع أوامر مترجم اللغة Verilog HDL تبدأ بالرمز (`) الفاصلة العلوية المقلوبة Back quote. فمثلاً، الأمر timescale 1 ns / 100ps يمثل مثال على موجه Directive مقياس الزمن وفيه يحدد: الرقم الأول وحدة قياس زمن التأخير والثاني الدقة التي تدور إليها التأخيرات.

بفرض أن زمن التأخير المحدد لكل بوابة في المخطط أعلاه هو 10, 20, 30 لكل من بوابات NOT, OR, AND على التوالي، فإن توصيف الدارة أعلاه مع تغير المداخل ab من 00 حتى 11 مروراً بالتركيبتين 01 و 10 يصبح عند محاكاة الدارة باستخدام لغة Verilog كالتالي: يتغير مخرج الدارة مع كل تغير للدخل بوجود التأخير المحدد أعلاه وفق ما هو مبين في الجدول التالي:

وحدة الزمن	الخرج (من أجل الدخل ab = 01) e ₁ e ₂ e ₃ e ₄ F	الخرج (من أجل الدخل ab = 10) e ₁ e ₂ e ₃ e ₄ F	الخرج (من أجل الدخل ab = 11) e ₁ e ₂ e ₃ e ₄ F
-	00 1 1 0 0 0	00 1 1 0 0 0	00 1 1 0 0 0
-	01 1 1 0 0 0	10 1 1 0 0 0	11 1 1 0 0 0
10 ns	01 0 1 0 0 0	10 1 0 0 0 0	11 0 0 0 0 0
20 ns	01 0 1 0 0 0	10 1 0 0 0 0	11 0 0 0 0 0
30 ns	01 0 1 0 1 0	10 1 0 1 0 0	11 0 0 0 0 0
40 ns	01 0 1 0 1 0	10 1 0 1 0 0	11 0 0 0 0 0
50 ns	01 0 1 0 1 1	10 1 0 1 0 1	11 0 0 0 0 0

يظهر من الجدول أن خرج العواكس e_1 أو e_2 مثلاً يتغير كل منهما من 1 إلى 0 بعد تأخير زمني 10 ns في كل من g_1 و g_2 ، لأن كلا من العاكسين يعمل بمعزل عن الأخرى. أما بوابة AND ذات التأخير المعطى بقيمة 30 ns يتغير خرجها (e_3) بالنسبة للبوابات g_3 و e_4 بالنسبة للبوابات (g_4) من 0 إلى 1 بعد تأخير 30ns للبوابتين، لأنهما تقعان بنفس المستوى، وكل منهما تعمل بمعزل عن الأخرى. وضمن هذه الفترة الأخيرة، أي بعد مرور أول 20 ns من 30 ns يمكن أن يتغير خرج البوابات OR لكن مداخلها دائماً ضمن هذه الفترة تكون بالقيمة 0 لذا بقي خرجها النهائي 'F' مساوياً للصفر، ويستمر بهذه القيمة حتى مرور 20 ns أخرى بعد آخر تغير لقيمة خرج البوابات AND أي 50 ns، حيث يتغير الخرج F بحسب القيم الموجودة على مداخل OR. يصبح الشكل العام للبرنامج:

// Description of circuit with delay

وصف الدارة مع التأخير

```

module ex_xor (a, b, F);
input a, b;
output F;
wire e1, e2, e3, e4;
not # (10) g1(e1, b),
not # (10) g2(e2, a);
    and # (30) g3(e3, e1, a),
    and # (30) g4(e4, e2, b);
or # (20) g5(F, e3, e4);
endmodule

```


13-1-1 مؤثرات لغة Verilog

يبين الجدول 13-1 مؤثرات Operators لغة Verilog لوصف العتاديات.

الجدول 13-1: مؤثرات لغة Verilog HDL

نوع المؤثر	الرمز	العملية المنفذة
حساب	+, -, *, /, %	جمع، طرح، جداء، قسمة معامل باقي القسمة
منطق (bit-wise أو اختزال)	~ & ^	إتمام NOT (بت مع بت) AND (بت مع بت) OR (بت مع بت) XOR (بت مع بت)
منطقي	! && 	نفي AND OR
إزاحة	>> << { , }	إزاحة إلى اليمين إزاحة إلى اليسار ضم (سلسلة)
علاقاتي	> < == != >= <=	أكبر من أقل من مساواة عدم تكافؤ أكبر من أو يساوي أصغر من أو يساوي
شرطي	?:	إذا ... و

يوجد تمييز بين العمليات الحسابية والمنطقية، لذلك استخدمت رموز مختلفة لكل منها. يستخدم رمز المساواة إشارتي (=) من دون ترك فراغ بينهما من الشكل (==) لتمييزه من تعليمات الإسناد التي تستخدم رمز (=) واحد. ويوفر مؤثر الضم آلية لمحاذاة عدة معاملات بعضها إلى جانب بعض، بمعنى أنه يمكن مثلاً ضم معاملين كل منهما مؤلف من بتين ليكونا معاً معاملاً جديداً بأربعة بتات وهكذا دواليك. وتفيد المؤثرات المنطقية والعلاقاتية Relational في تحديد شروط التحكم. تنجز العمليات الحسابية بأعداد ثنائية وتمثل الأعداد السالبة بتمماتها إلى اثنين. فمثلاً، مؤثر المعامل modulus % ينتج الباقي من قسمة عددين، إذ أن نتيجة العملية 5 % 12 هو الباقي 2. ويشار إلى الأعداد الثنائية بلغة Verilog بالحرف b مسبقاً بإشارة فتحة أي b'. ويذكر حجم العدد أولاً ثم قيمته، وهكذا فإن 2'b01 تحدد عدداً ثنائياً مؤلفاً من بتين قيمته 01. وهكذا 1'b0 تمثل رقماً ثنائياً واحداً بقيمة صفر و 1'b1 تمثل رقماً ثنائياً واحداً بقيمة واحد. أما لتمثيل قيمة من الشكل 1101 فيكتب 4'b1101، وهكذا الحال بالنسبة لأي عدد آخر. ويمكن باستعمال الرموز 'h, 'd, 'o, 'b تحديد الأعداد بالأنظمة: الستة عشري أو العشري أو الثماني أو الثنائي. في حال عدم ذكر نوع العدد فإنه يُعدّ عشرياً. وإذا لم يذكر حجم العدد فيُعدّ مكوناً من 32 bits. ويوجد بلغة Verilog نوعان من المؤثرات المنطقية، هما:

- بت بت bit-wise : تنفذ مؤثرات البت بت العملية المنطقية على معاملين بت بت - bit-by-bit، أي يؤخذ البتّان المتقابلان في كل من المعاملين، وتجرى عليهما العملية.

- طريقة الاختزال (التخليص) Reduction: وبينما ينفذ مؤثر الاختزال العملية المنطقية على معامل Operand واحد، تقوم هذه المؤثرات بتنفيذ العملية المنطقية على بتات المعامل الواحد بتاً بتاً بدءاً من اليمين باتجاه اليسار، ويمكن بنتيجة العملية الحصول على نتيجة مكونة من بت واحد، فمثلاً، نتيجة تطبيق مؤثر الاختزال (~ |) الذي يعني NOT OR على المعامل 00101 هو 0، بينما عند تطبيق نفس المعامل على 00000 تكون النتيجة 1. ويمكن أن تأخذ المؤثرات المنطقية والعلاقاتية بوظيفة معاملات، ومتحولات Variables أو تعابير Expressions.

تستخدم بصورة أساسية لتحديد حالات الصح أو الخطأ بمعنى إذا كان الشرط محققاً (صح) يقيّم بواحد، و يقيّم بصفر عند عدم تحقق الشرط. أما إذا كان الشرط غير واضح فيقيم بالقيمة d. وعندما يكون المعامل عدداً، تعطى قيمة 0 إذا كان العدد يساوي الصفر، في حين تعطى القيمة 1 إذا كان العدد مغايراً للصفر. فمثلاً، من أجل $A = 1101$ و $B = 0000$ ، يؤخذ عندها A واحداً، لأنه لا يساوي الصفر، أما B فيؤخذ صفراً. وتكون نتائج العمليات الأخرى لهذه القيم كالتالي:

$$A \&\& B = 0$$

$$A || B = 1$$

$$! A = 0$$

$$(A > B) = 1$$

$$(A == B) = 0$$

ترتيب مؤثرات الإزاحة معامل المتجه Vector Operand إلى اليمين أو إلى اليسار عدداً محدداً من البتات. تملأ البتات التي تشغّر نتيجة عملية الإزاحة باصفار. فمثلاً، إذا كان $R = 11010$ ، تؤدي العبارة $R = R >> 1$ إلى إزاحة R موضعاً واحداً إلى اليمين، لتصبح النتيجة $R = 01101$. وأخيراً المؤثر الشرطي (?) يأخذ ثلاث معاملات وفق التالي: Condition? true expression : false expression: ، و يقيم الشرط على النحو التالي: إذا كانت النتيجة 1 (الشرط محققاً) ينفذ التعبير true expression، أما إذا كانت النتيجة 0، فينفذ التعبير false expression، وهذه العملية مكافئة للحالة if-else.

13-1-2 تقنيات النمذجة

بداية نشير من دون الدخول بالتفاصيل إلى أن كل من البوابات المنطقية القياسية تأخذ بلغة HDL كلمة مفتاحيه بنفس اسم البوابة مثل ... and, not, وتعرف على أنها أوليات النظام System primitives. ويمكن للمستخدم أن يوجد أوليات إضافية يعرفها على شكل جداول (وبعد جدول الحقيقة من إحدى الطرائق التي بموجبها تحدد الدارة الرقمية جدولياً) وتسمى أوليات معرفة من قبل المستخدم UDP (User-Defined Primitives). لا يستخدم UDP الكلمات المفتاحية module ... endmodule، بل يعلن عنها باستخدام الكلمتين primitive ... end primitive وضمنها يعرف أي جدول بالكلمتين المفتاحيتين table

end table ويمكن وصف المجتزأ بلغة Verilog HDL بواسطة أي من تقنيات النمذجة Modeling Techniques التالية:

- 1- النمذجة على مستوى البوابة Gate-Level Modeling باستخدام انتساخ Instantiation البوابات والوحدات الأولية المعرفة من قبل المستخدم. تصف النمذجة على مستوى البوابة الدارة عن طريق تعيين البوابات وكيفية وصلها ببعضها ببعض.
- 2- النمذجة بطريقة تدفق (جريان) المعطيات Dataflow Modeling باستخدام عبارات إسناد مستمرة Continuous Assignment Statements. يستخدم هذا النوع من تقنيات النمذجة غالباً في وصف الدارات التركيبية.
- 3- النمذجة السلوكية Behavioral Modeling باستخدام عبارات إجرائية Procedural Assignment Statements مع الكلمة المفتاحية always. وتستخدم في وصف النظم الرقمية على مستوى عال من التجريد.
- 4- النمذجة على مستوى التبديل Switch-Level Modeling ويوفر هذا النوع من النمذجة إمكانية التصميم حتى على مستوى الترانزستور. وفيما يلي نستعرض باختصار مفهوم كل من هذه التقانات.

1- النمذجة على مستوى البوابات المنطقية

تميز لغة Verilog اثني عشرة بوابة أساسية لتكون بوابات أولية معرفة سلفاً، أربع منها من نوع البوابات ثلاثية الحالة، والثماني المتبقية تمثل بوابات عادية، يصرح عنها عادة بأحرف صغيرة بالكلمات المفتاحية: buf, not, and, or, nand, nor, xor, xnor. يخصص عند محاكاة البوابات منطق بأربع قيم لكل بوابة على حدة هي: 0, 1, d, z حيث d - تعني قيمة غير معروفة Unknown، ويقصد بها عدم اكتراث. وتُعَدّ قيمة ما خلال المحاكاة d، عندما يكون الدخل أو الخرج غامضاً.

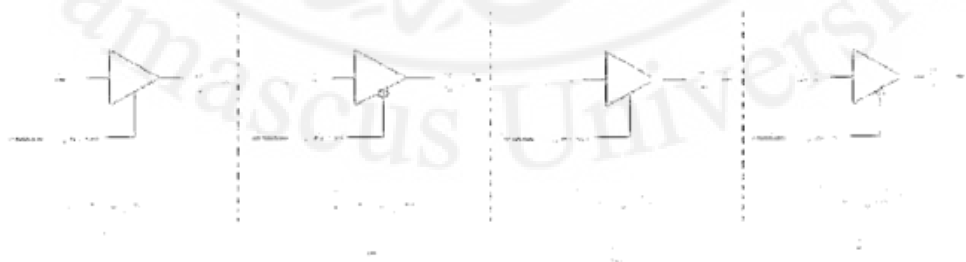
z - وتعني ممانعة عالية High Impedance، وتحدث فقط في خرج البوابة ثلاثية الحالة إذا ترك سلك ما سهواً من دون توصيل. وجدول الحقيقة لكل من البوابات NOT, AND, OR, XOR مبينة في الجدول 13-2.

وجداول الحقيقة للبوابات الأربع المتبقية هي مماثلة مع فارق أن المخارج متممة، بحيث إن Buf هو متمم لـ NOT و NAND متمم لـ AND وهكذا.

الجدول 2-13: جداول الحقيقة للبوابات الأولية المعرفة سلفاً

الرمز	الاسم	الرمز	الاسم
	AND		NAND
	OR		NOR
	XOR		XNOR
	Buf		NOT

تحدد الدارة المنطقية بالنمذجة على مستوى البوابة (وفق المثال حول محاكاة البوابة XOR) بواسطة بواباتها المنطقية وتوصيلاتها مع بعضها. عند مكاملة Incorporated بوابة ضمن مجتزأ ما، يقال أن البوابة انتسخت Instantiated ضمن المجتزأ، بالتالي منتسخت العناصر هي عبارات تدل على عناصر ذات مستوى أدنى في التصميم. أما بخصوص البوابات ثلاثية الحالة فيكون لكل منها مدخل تحكم إضافي يتم بوساطته وضع البوابة بحالة الممانعة العالية. يصادف في الدارات المنطقية أربعة أنواع من البوابات ثلاثية الحالة، وفق ما هو مبين في الشكل 1-13، حيث كل من هذه البوابات الأربعة تدخل حالة الممانعة العالية Z.



الشكل 1-13: الأشكال الأربعة الممكنة للبوابات ثلاثية الحالة

إذا لم تتحقق القيمة المناسبة لمدخل التحكم. بالتالي يوفر هذا النوع من التمثيل وصفاً نصياً للمخطط، ويتم انتساخ كل بوابة بواسطة العبارة (gate name (output, input, control ويمكن أن يكون اسم البوابة أي من الأسماء الأربعة للبوابات ذات الثلاث حالات، ويمكن أن تكون قيمة الخرج z , 1, 0. فمثلاً، لانتساخ البوابة في الشكل 13-1 أ ضمن مجزأ ما يمكن أن تكتب الصيغة التالية (bufif1 (out, A, control، حيث بواسطة الكلمة المفتاحية bufif1 تنتقل القيمة الموجودة على المدخل A كما هي إلى المخرج عندما يكون مدخل التحكم $control = 1$. وينتقل المخرج إلى الحالة z عندما يكون $control = 0$. أما الأمر (not0 (Y, B, enable المقابل للشكل 13-1 د فيعني: الخرج $Y = z$ إذا كان مدخل التأهيل $enable = 1$. ويكون $Y = \text{not} (B)$ عندما يكون $enable = 0$.

2- النمذجة بطريقة تدفق المعطيات

تستخدم النمذجة بطريقة تدفق أو جريان المعطيات Dataflow Modeling عدداً من المؤثرات التي تؤثر في المعاملات للحصول على النتائج المطلوبة. وتستخدم طريقة نمذجة تدفق المعطيات إسنادات مستمرة مع الكلمة المفتاحية assign. الإسناد المستمر هو عبارة يتم بموجبها إسناد قيمة إلى شبكة. واستخدمت الشبكة نوع المعطيات Data Type Net لتمثيل الوصل الفيزيائي بين عناصر الدارة. وتعرف الشبكة مخرج بوابة مصرح عنه إما بالكلمة المفتاحية output أو wire. وتحدد القيمة المسندة لشبكة ما بواسطة تعبير يحتوي معاملات ومؤثرات. وتقدم النمذجة بطريقة تدفق المعطيات أيضاً وسيلة لتوصيف الدارات المنطقية (وبخاصة التركيبية منها) اعتماداً على وظيفة هذه الدارات بدلاً من البوابات.

3- النمذجة السلوكية

تمثل النمذجة السلوكية Behavioral Modeling الدارات الرقمية اعتماداً على المستوى الوظيفي والخوارزمي. وكما تستخدم هذه التقنية بوصف الدارات التركيبية، فهي غالباً ما تستعمل لوصف الدارات التتابعية. تستعمل النمذجة السلوكية الكلمة المفتاحية always متبوعة بعبارات أوامر إسناد إجرائية Procedural Assignment Statements. يجب أن يكون الخرج الهدف من هذه الاسنادات الإجرائية من نوع معطيات سجل.

4- النمذجة على مستوى التبديل

يمثل هذا النوع من النمذجة نوعاً خاصاً يستخدم في التصميم على مستوى الترانزستور MOS. وبالطبع ترانزستورات MOS يمكن اعتبارها مفاتيح تكون إما موصولة Conduct أو مفتوحة Open. يوصف نوعاً مفتاحي MOS بالكلمتين المفتاحيتين nmos, pmos. وينسخ هذان النوعان بتحديد أطراف Terminals الترانزستور الثلاثة بالترتيب التالي: nmos (drain, source, gate); pmos (drain, source, gate); وعند تصميم دارات من عائلة MOS يجب تحديد التوصيلات إلى منبع التغذية U_{dd} والأرض. وتعرف التغذية والأرضي بالكلمتين المفتاحيتين supply1 و supply0 وتعيان بالتعليمتين التاليتين: supply1 PWR; supply0 GND; توصيف دائرة العاكس بالتقانة CMOS يصبح وفق الشكل التالي:

// Description of CMOS Inverter

module inverter (Y, A);

input A;

output Y;

supply1 PWR;

supply0 GND;

pmos (Y, PWR, A);

// (Drain, Source, Gate)

nmos (Y, GND, A);

endmodule

يفهم من التوصيف أعلاه، بأنه تم في البداية التصريح عن الدخل A والخرج Y ومنبعي التغذية $U_{dd} = \text{log1 PWR}$ والأرضي GND. بعد ذلك ينتسخ المجتزأ ترانزستور PMOS و ترانزستور NMOS. والمخرج Y مشترك لكلا الترانزستورين عند مصرفيهما (لأن المصرفين متصلان) والدخل أيضاً مشترك للترانزستورين عند بوابتيهما (لأن البوابتين موصولتان معاً).

13-1-3 تمثيل التعابير البوليانية

حددت التعابير البوليانية بلغة Verilog HDL باستخدام عبارة تخصيص مستمرة تتألف من الكلمة المفتاحية assign متبوعة بتعبير بولياني. فمثلاً، تابع XOR بالنسبة للمتحولين a, b يمكن تمثيله وفق الشكل التالي: $assign F = (\sim a \& b) | (a \& \sim b)$; وبالعكس، إذا كان لدينا تعبير محدد يمكن بلغة Verilog توصيف الدارة المقابلة له.

مثال: بين توصيف الدارة المحددة بالتعبيرين البوليانيين:

$$X = A + BC + B'D$$

$$Y = B'C + B C' D'$$

الجواب: يوجد للدارة المعطاة بهذين التعبيرين مخرجان هما X, Y وأربعة مداخل هي: A, B, C, D يمكن التعبير عنها بعبارة assign وتوصيفها كما لو قمنا برسم مخططها.

توصيف دارة محددة بتعابير بوليا نية: // Description of Circuit specified

```
module cir_boolean (X, Y, A, B, C, D);  
input A, B, C, D;  
output X, Y;  
assign X = A | (B & C) | (~B & D);  
assign Y = (~B & C) | (B & ~C & ~D);  
endmodule
```

13-2 وظائف Verilog للإظهار

يمكن أيضاً إظهار مخارج رقمية باستخدام مهمات نظام Verilog التي تكون مبيتة في وظائف النظام والتي يمكن تمييزها باستخدام كلمات مفتاحية تبدأ بالرمز S. ومن مهمات النظام Verilog System Tasks المفيدة للإظهار نستعرض فيما يلي:

- Sdisplay: وهي من إحدى المهام التي تقوم بإظهار قيمة واحدة للمتحولات في لحظة زمنية مع عودة لنهاية السطر End-of-Line Return.
- Swrite: وهي مثل Sdisplay لكن من دون عودة إلى السطر التالي.
- Smonitor: تظهر متحولات عند تغير قيمة متحولات ما خلال المحاكاة.
- Stime: تظهر زمن المحاكاة.
- Sfinish: تنهي عملية المحاكاة.

الصيغة القوا عدية Syntax للعبارات Sdisplay, Swrite, Smonitor هي كالتالي:
 Task name (format specification, argument list);
 Task name اسم المهمة
 Sdisplay, Swrite, Smonitor: أما مواصفات الصيغة فتضمن
 أساس العدّ للأعداد المظهرة باستخدام الرمز %. فمثلاً، العبارة Sdisplay (%d %o %b
 %h %h, A, B, C, D, E); تحدد إظهار A بالعمشري لوجود %d بوصفه رمزاً مقابل له
 من decimal، و B بالثمانى لنفس الأسباب، و C بالثنائى و D, E بالستة عمشري. لا
 يوجد بمواصفات الصيغة فواصل وأن مواصفات الصيغة تفصل عن قائمة المتحولات
 Argument List بفاصلة وأن المتحولات في لائحة المتحولات (الأدلة) مفصولة عن
 بعضها البعض بفاصل أيضاً. يمكن أن تتضمن مواصفات الصيغة سلسلة محصورة بين
 معترضتين (") أيضاً. فمثلاً، الشكل التالي:

Sdisplay ("time = &0d A= %b B = %h", Stime, A, B);

يحدد مجموعة محصورة بين معترضتين، حيث (time =) و (A =) و (B =) هي أجزاء
 من المجموعة المطلوب إظهارها. والصيغة %0d و %b و %h تحدد الأساس لكل من
 Stime و A و B. يفضل عند إظهار قيمة الزمن استعمال الصيغة %0d بدلا من %d

3-13 الحلقات Loops

يوجد بلغة HDL أربعة أنواع من الحلقات Loop Statements التي تسمح بتنفيذ العبارات
 الإجرائية التكرارية هي: كرر Repeat، إلى الأبد Forever، مادام While، من أجل
 For. يجب أن تظهر جميع عبارات الحلقات داخل كتلة initial و always. ولتوضيح
 آلية تنفيذ كل من هذه الحلقات، ننطلق من مثال ذي طابع تكراري، مثل توليد قطار من
 النبضات الساعية المنتظمة وفق الشكل المبين أدناه.



ونبين فيما يلي آلية تنفيذ كل من هذه الأنواع الأربعة للحلقات على حدة وفق التالي: -
 تنفذ الحلقة " كرر Repeat " العبارات المرافقة لها عدداً محدداً من المرات. وفيما يلي نبين
 طريقتين ممكنتين لإنتاج نبضات الساعة باستخدام العبارة Initial، هما:

الأولى - وتأخذ الشكل التالي:

```
initial
begin
Clock = 1'b0;
repeat (30)
#10clock = ~ clock;
end
```

توضع الساعة على القيمة 0 ($t = 0$). تتَم الساعة كل 10 وحدات زمنية، وتكرر هذه العملية 30 مرة، فينتج 15 دورة ساعة، وزمن كل منها 20 وحدة زمنية.
الثانية - وتأخذ الشكل التالي:

```
initial
begin
Clock = 1'b0;
#300 Sfinish;
end
always
#10 clock = ~ clock;
```

تضع الكتلة initial الساعة على 0 باللحظة $t = 0$ ، وبعد 10 وحدات زمنية تتم العبارة always حالة الساعة تكرارياً كل 10 وحدات زمنية منتجة بذلك ساعة بدورة زمنية مقدارها 20 وحدة زمنية. وتتوقف المحاكاة استجابة لأمر النظام الذي يعبر عنه بالكلمة المفتاحية finish عند الزمن 300 وحدة زمنية.

٥- تسبب الحلقة " إلى الأبد Forever " تنفيذاً متتالياً مستمراً للعبارة الإجرائية. فمثلاً، لإنتاج قطار مستمر من نبضات الساعة، يمكن استخدام الصيغة التالية:

```
Initial
begin
Clock = 1'b0;
Forever #5 clock = ~ clock;
end
```

°- تتفّذ الحلقة " While " مادام "While" كتلة من التعليمات بصفة متكررة طول المدة التي يكون فيها محقق شرط ما، ولا تتفّذ أي شيء إذا كان الشرط غير محقق. فمثلاً، بالحلقة التالية:

```
Integer I;
Initial
begin
  I = 0;
  While ( I < 100)
    I = I + 1;
end
```

تزيد قيمة العدّ من 0 ولغاية 99. ويتم الخروج من الحلقة عند العدّ 100. وعند التعامل مع عبارات الحلقات، يفضل استعمال معطيات من نوع الأعداد الصحيحة، التي يصرح عنها بالكلمة المفتاحية **Integer** كما في المثال أعلاه. ويمكن أيضاً استعمال الكلمة المفتاحية **reg** للمتحوّلات، وبهذه الحالة تخزن المتحوّلات التي يصرح عنها على أنها معطيات من نوع reg بوصفها أعداداً غير مقيدة بإشارة. أما الأعداد التي يصرح عنها على أنها معطيات من النوع Data Type Integer فتخزن كأعداد مقيدة بإشارة بصيغة متمماتها إلى اثنين 2's Complement.

°- الحلقة " من أجل For " تتضمن ثلاثة أجزاء تفصل بينها فواصل منقوطة:

- شرط ابتدائي Initial Condition

- تعبير لفحص شرط الانتهاء Terminating Condition

- عملية إسناد لتغيير قيمة متحول التحكم Control Variable

الشكل العام لحلقة For هو: عبارات إجرائية .. (I = 0; I < 100; I + 1) for

4-13 منصة الاختبار Test Bench

تمثل منصة الاختبار Test Bench برنامج HDL يستعمل لتطبيق محفز Stimulus على تصميم HDL بغية فحص استجابته خلال عملية المحاكاة. ويمكن أن تكون برامج منصات الاختبار طويلة ومعقدة وقد تأخذ وقتاً لتطويرها أطول من التصميم موضوع الاختبار. ولكن هنا سنقتصر على وصف نموذجي لمجترّات المحفز بلغة وصف العتاديات HDL stimulus modules. وتستعمل منصات الاختبار عبارتي always و initial

لتحفيز الدارة الموضوعية تحت الاختبار. وتنفذ عبارة always تكرارياً في حلقة، في حين تنفذ عبارة initial مرة واحدة عند زمن بدء عملية المحاكاة. وقد تستمر مع أي عملية تأخرت بعدد من وحدات الزمن باستخدام الرمز #. لنأخذ على سبيل المثال كتلة initial التالية:

```
Initial
begin
A = 0; B = 0;
#10 A = 1;
#20 A = 0; B = 1;
end
```

وتحصر الكتلة بين الكلمتين المفتاحيتين begin و end. عند الزمن $t = 0$ يوضع كل من A, B على الصفر. وبعد انقضاء عشر وحدات زمنية Time units تتغير A لتصبح واحداً، وبعد 20 وحدة زمنية أخرى (أي عند $t = 30$) تتغير A لتصبح 0 و B لتصبح 1. ويمكن باستخدام كتلة initial توليد المداخل الثلاثة في جدول الحقيقة كالتالي:

```
initial
begin
D = 3'b000;
repeat ( 7 )
#10 D = D + 3'b001; end
```

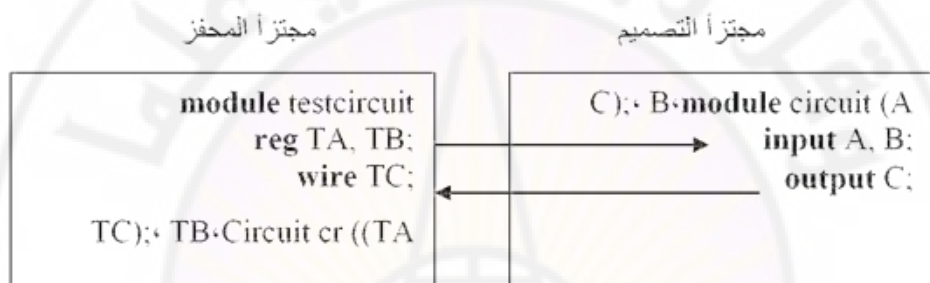
يعطي المتجه D المؤلف من ثلاثة بتات قيمة ابتدائية 000 عند الزمن $t = 0$ وتحدد الكلمة المفتاحية repeat عبارة حلقة Looping Statement يضاف فيها كل مرة واحد إلى قيمة المتجه D لسبع مرات، مرة كل عشر وحدات زمنية للوصول إلى تتابع الأعداد الثنائية من 000 حتى 111. مجتزأ المحفز stimulus module هو برنامج HDL له البنية التالية:

- مجتزأ اسم الاختبار module test name

- التصريح عن المحددات المحلية من نوع wire, reg
- انتساخ مجتزأ التصميم المراد فحصه (المجتزأ الواصف للتصميم)
- توليد المحفز stimulus للنظام باستخدام العبارتين initial و always
- إظهار استجابة الخرج للنظام وأخيراً نهاية المجتزأ endmodule.

لا يوجد عادة لمجترأ الاختبار مداخل أو مخارج. ويتم التصريح في مجترأ المحفز عن الإشارات المطبقة كمداخل لمجترأ التصميم بغرض المحاكاة باعتبارها معطيات من نوع سجل محلي local reg data type.

يصرح عن المخارج في مجترأ التصميم التي تظهر من أجل الفحص في مجترأ المحفز باعتبارها معطيات من نوع سلك محلي local wire data type. وبعدئذ ينتسخ المجترأ موضوع الاختبار باستخدام Local Identifier. يبين الشكل أدناه هذه العلاقة.



يولد المجترأ المحفز مداخل لمجترأ التصميم بالتصريح عن المعرفات TA, TB كنوع سجل reg، ويفحص خرج مجترأ التصميم بواسطة المعرف TC من نوع wire. تستعمل عندئذ المعرفات المحلية لانتساخ مجترأ التصميم موضوع الاختبار. تظهر استجابة المحفز المولدة بواسطة الكلمتين initial و always على خرج المحاكى Simulator بوصفها مخططات زمنية. وتمثل محاكاة المنطق طريقة سريعة ودقيقة لتحليل الدارات المنطقية (التركيبية) للتحقق أنها تعمل بوجه صحيح. يوجد نوعان من التحقق هما:

تحقق وظيفي: يدرس فيه العمل المنطقي للدارة بعيداً عن الاعتبارات الزمنية. وهذا ما يمكن عمله باستنتاج جدول الحقيقة.

تحقق زمني: ويدرس فيه عمل الدارة بعد أخذ التأخيرات الزمنية للبوابات بالحسبان، وهذا يتم بملاحظة أشكال الموجات على مخارج البوابات عند الاستجابة. وفيما يلي نعرض مثلاً عن التحقق الزمني الذي يأخذ بالحسبان تأخير البوابات المنطقية. بالعودة إلى المثال السابق حول توصيف دارة XOR بوجود التأخير، ويمكن كتابة منصة الاختبار لمحاكاة الدارة مع التأخير وفق الشكل المبين أدناه والذي أصبح يتضمن مجترأين:

الأول- مجتزأ المحفز Stimulus Module

والثاني- مجتزأ وصف الدارة وفق التالي:

محفز لدارة XOR من أجل التركيبة $ab = 11$ ، // Stimulus for XOR circuit

```
module stimcret;
reg a, b;
wire e1, e2, e3, e4;
ex_xor cwd (a, b, F);
initial
begin
a = 1'b0; b = 1'b0;
#100
a = 1'b1; b = 1'b1;
#100 Sfinish;
end
endmodule
```

// Description of circuit with delay

وصف الدارة مع التأخير

```
module ex_xor (a, b, F);
input a, b;
output F;
wire e1, e2, e3, e4;
not # (10) g1(e1, b),
not # (10) g2(e2, a);
and # (30) g3(e3, e1, a),
and # (30) g4(e4, e2, b);
or # (20) g5(F, e3, e4);
endmodule
```

يلاحظ من الشكل أعلاه أن بمجتزأ المحفز timeret لا يوجد منافذ. وقد أعلن عن المداخل إلى الدارة بالكلمة المفتاحية reg، وعن المخرج بالكلمة المفتاحية wire. ولقد استسخت ex_xor مع الاسم cwd. وتعين العبارة initial المداخل بين الكلمتين المفتاحيتين begin و end. بداية نضع $ab = 00$ وبعد 100 ns تتغير قيمة المداخل إلى القيمة $ab = 01$ مثلاً. وتنتهي المحاكاة بعد 100 ns أخرى بوجود مهمة النظام Sfinish.

13- 5 لغة وصف العتاديات للدارات التركيبية

نكمل توصيف الدارات التركيبية باستخدام لغة Verilog HDL، حيث يمكن توصيف الـ Module بلغة Verilog HDL بوساطة واحدة من أربع تقنيات للنمذجة:

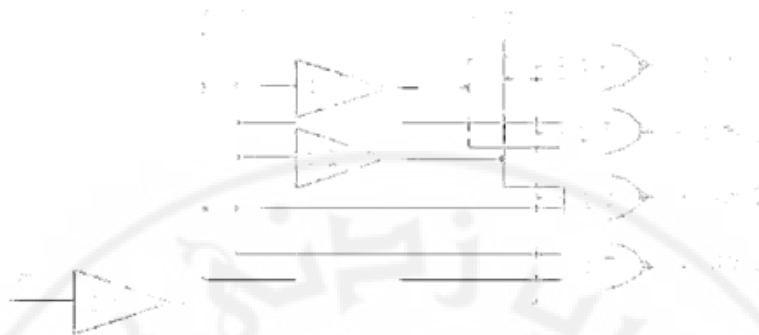
- النمذجة على مستوى البوابة
- النمذجة بطريقة تدفق المعطيات
- النمذجة السلوكية
- النمذجة على مستوى التبديل

ونستعرض فيما يلي توصيف الدارات المنطقية التركيبية باستخدام التقانات المناسبة لتوصيفها من خلال بعض الأمثلة المحددة من دون الدخول بمزيد من التفاصيل، علماً أن كل من تقنيات النمذجة تناسب نوعاً من الدارات.

أولاً- النمذجة على مستوى البوابات المنطقية:

بداية نشير إلى آلية تعامل لغة Verilog HDL مع أعداد مختلفة من البتات والتي تدعى بالأسعة (المتجهات) Vectors. ويحدّد المتجه بوساطة أقواس متوسطة يوضع بداخلها عدنان مفصولان عن بعضهما بنقطتين، فمثلاً، التعبير $A[0:7]$ output يصرّح عن متجه خرج A مؤلف من ثمانية بتات مرقمة من 0 حتى 7. وتحدّد مختلف البتات المنفردة ضمن قوسين مربعين أيضاً. فمثلاً، التعبير $A[3]$ يحدّد البت الثالث من المتجه A، كما يمكن عنوانه جزء من المتجهات. فمثلاً، التعبير $A[2:0]$ يحدّد من المتجه A البتات الثلاثة الأقل وزناً. نبيّن فيما يلي مثالين للنمذجة على مستوى البوابات Gate-Level، يستخدم كل منهما أعداداً مختلفة من البتات.

1- حول نمذجة مفكك الترميز 2x4 Decoder، حيث بعد إعادة رسم الشكل وتسمية البوابات الداخلة في تركيبة مفكك الترميز السبعة، بالأسماء من g_1 حتى g_7 وفق ما هو مبين في الشكل أدناه:



يكون التوصيف المقابل بلغة Verilog HDL وفق التالي:

// Gate-Level Description of a 2-to-4-Line Decoder

```
module decoder_gate (a, b, E, CS);
```

```
  input a, b, E;
```

```
  output [0:3] CS;
```

```
  wire anot, bnot, Enot;
```

```
  not
```

```
    g1(anot, a),
```

```
    g2(bnot, b),
```

```
    g3(Enot, E);
```

```
  nand
```

```
    g4(CS[0], anot, bnot, Enot),
```

```
    g5(CS[1], a, bnot, Enot),
```

```
    g6(CS[2], anot, b, Enot),
```

```
    g7(CS[3], a, b, Enot);
```

```
endmodule
```

يستخدم التصريح سلك wire للتوصيلات الداخلية، وتنتج البوابات not متممات المدخلين a و b ومدخل التأهيل E، وتوفر البوابات nand مخارج CS. ويلاحظ أن الكلمات المفتاحية not و nand كتبت لمرة واحدة ولا حاجة لتكرارها بعدد البوابات شرط أن تضاف فواصل عادية عند نهاية كل بوابة باستثناء الإسناد الأخير. ويمكن جمع مجتزئين أو أكثر لبناء وصف لتصميم ما، حيث يوجد نوعان من منهجيات التصميم:

- الأول، ويعتمد منهجية التصميم من الأعلى إلى الأسفل Top-Down
- الثاني، ويعتمد منهجية التصميم من الأسفل إلى الأعلى Bottom-Up.

تعرّف كتلة مستوى القمة Top-level block في التصميم من الأعلى إلى الأسفل، وبعد ذلك تعرف الكتل الجزئية اللازمة لبناء كتلة مستوى القمة. أما في التصميم من الأسفل إلى الأعلى فتعرّف كتل البناء أولاً، ثم تجمع لبناء كتلة مستوى القمة. فمثلاً، الجامع الثنائي يمكن اعتباره عنصر كتلة القمة، بني من أربع كتل جامع كامل، وكتلة الجامع الكامل مبنية بدورها من كتلتي نصف جامع. ففي طريقة التصميم من القمة إلى القاعدة يعرّف الجامع لأربعة بتات أولاً، ثم يوصف نصفاً الجامع، بعدئذ يبني الجامع لأربعة بتات من أربعة جوامع كاملة. أما في طريقة التصميم من القاعدة إلى القمة فيعرّف نصف الجامع أولاً، ثم يبني الجامع الكامل بوساطة نصفي جامع. وبالاتفاق من مثال محدّد وليكن توصيف جامع ثنائي بطول أربعة بتات، نبدأ بتجميع الأشكال حول دائرة نصف الجامع والجامع الكامل وفق الشكل أدناه. يكون التوصيف بلغة Verilog HDL كالتالي:

```
// Gate-Level hierarchical description of 4-bit adder
// Description of half adder (see Fig 45-a)
module halfadder (S, C, x, y);
    input x, y;
    output S, C;
// Instantiate primitive gates
    xor (S, x, y);
    and (C, x, y);
endmodule
```

```

// Description of full adder
module full_adder (S, C, x, y, z);
    input x, y, z;
    output S, C;
    wire S1, D1, D2;
// instantiate the half adder
    Halfadder HA1 (S1, D1, x, y),
                HA2 (S, D2, S1, z);
    or g1 (C, D2, D1);
endmodule
// Description of 4-bit adder
module _4bit_adder (S, C4, A, B, C0);
    input [3:0] A, B;
    input C0;
    output [3:0] S;
    output C4;
    wire C1, C2, C3;
// instantiate the full adder
    Fulladder FA0 ( S[0], C1, A[0], B[0], C0),
                FA1 ( S[1], C2, A[1], B[1], C1),
                FA2 ( S[2], C3, A[2], B[2], C2),
                FA3 ( S[3], C4, A[3], B[3], C3);
endmodule

```

يلاحظ في التوصيف أعلاه، أنه تم تعريف نصف الجامع بانتساخ البوابات الأولية أولاً. يصف المجتزأ التالي الجامع الكامل بانتساخ نصفى جامع، ويصف المجتزأ الثالث جامع الأربع بتات بانتساخ أربعة جوامع كاملة. ولقد جرت عملية انتساخ المجتزأ باستخدام اسم المجتزأ الذي تم انتساخه مع مجموعة أسماء منافذ port جديدة. تم انتساخ نصف الجامع HA₁ الموجود ضمن مجتزأ الجامع الكامل بالمنافذ: S₁, D₁, x, y وهذا ينتج نصف جامع مخرجه S و D₁ ومداخله x و y. وبما أن الطريقة الوحيدة لتضمين مجتزأ في آخر هي انتساخه، لهذا تنتسخ المجتزآت ضمن مجتزآت أخرى لإيجاد وصف هرمي للتصميم، حيث لا يمكن بـ Verilog حشر مجتزأ آخر ضمن الكلمتين module و endmodule.

ثانياً - النمذجة بطريقة تدفق المعطيات:

سبق وأشرنا إلى أن هذه التقنية تستخدم إسنادات مستمرة مع الكلمة المفتاحية assign، حيث على سبيل المثال، توصيف مضّم بسيط 2x1 MUX بمدخلين للمعطيات A و B ومدخل للانتقاء S ومخرج Y يمكن التعبير عنه بالإسناد المستمر:

```
assign Y = ( A & S ) | ( B & ~S);
```

ومنه يفهم أن الهدف، أي المخرج Y يأتي مباشرة بعد الكلمة assign متبوعاً بإشارة الإسناد (=) يليها التعبير البولياني. وهذا يكافئ بلغة العتاديات توصيل مخرج البوابة OR أي (|) إلى السلك Y. ونبين فيما يلي نفس المثالين اللذين تم تناولهما في الفقرة أعلاه حول آلية النمذجة باستخدام طريقة تدفق المعطيات.

الأول: حول نمذجة 2x4 Decoder بطريقة تدفق المعطيات. وتعرف الدارة وفق هذه الطريقة بأربع عبارات إسناد (واحدة لكل مخرج) باستخدام التعبيرات البوليانية وفق التالي

```
// Dataflow Description of a 2-to-4 Line Decoder
```

```
module decoder_DF (A, B, E, D);
```

```
    input A, B, E;
```

```
    output [0:3] D;
```

```
    assign D[0] = ~(~A & ~B & ~E);
```

```
        D[1] = ~(~A & B & ~E);
```

```
        D[2] = ~(A & ~B & ~E);
```

```
        D[3] = ~(A & B & ~E);
```

```
endmodule
```

الثاني: الوصف بطريقة تدفق المعطيات لجامع أربعة بتات. يوصف الجمع المنطقي هنا باستخدام مؤثر الجمع ومؤثر الضم بعبارة واحدة

```
// Dataflow Description of 4-bit adder
```

```
module binary_adder (A, B, Cin, S, Cout);
```

```
    input [3:0] A, B;
```

```
    input Cin;
```

```
    output [3:0] S;
```

```
    output Cout;
```

```
    assign { Cout, S } = A + B + Cin;
```

```
endmodule
```

يحدّد رمز الجمع (+) عملية الجمع الثنائي للبتّات الأربعة من A مع البتّات الأربعة من B والبتّ الوحيد للمنقول C_{in} . والخرج الهدف هو ناتج ضم منقول الخرج Cout مع البتّات الأربعة للمجموع S. ويعبّر عن عملية ضم المعاملات بوضعها ضمن أقواس كبيرة مع فواصل تفصل بينها، لذلك يمثّل التعبير {Cout, S} الناتج المؤلف من خمسة بتّات. ونعرض فيما يلي مثلاً آخر حول توصيف مقارن لعددين كل منهما مكون من 3bits.

// Dataflow Description of a 3-bit Comparator

module Comp (A, B, ALTB, AGTB, AEQB);

input [2:0] A, B;

output ALTB, AGTB, AEQB;

assign ALTB = (A < B),

AGTB = (A > B),

AEQB = (A == B);

Endmodule

ويظهر الوصف بطريقة تدفق المعطيات أعلاه، كيف يسهل التصميم المنطقي للدائرة. يحدّد

المجتزأ مدخلين A, B كل منهما مكون من 3bit ويحدّد ثلاثة مخارج هي:

- المخرج ALTB الذي يأخذ القيمة 1 عندما يكون $A < B$.

- المخرج AGTB الذي يأخذ القيمة 1 عندما يكون $A > B$.

- المخرج AEQB الذي يأخذ القيمة 1 عندما يكون $A = B$.

ثالثاً - النمذجة السلوكية:

سبق وأشرنا إلى أن هذه الطريقة تستخدم بشكل أساسي لتوصيف الدارات المتتابعة، إلا أنه

يمكن أيضاً استخدامها بتوصيف الدارات المنطقية التركيبية. ونعرض فيما يلي مثلاً عن

استخدام هذه الطريقة بتوصيف الدارات التركيبية. وليكن مثلاً حول الوصف السلوكي

لمضمّن بسيط 2x1MUX.

بما أن المتحول Y هو مخرج هدف للمضمّن، يجب التصريح عنه بوصفه معطيات سجل

reg data. تتخذ عبارات الإسناد الإجرائية داخل كتلة always في كل مرة يحدث فيها تغيير

في أحد المتحولات المذكورة بعد الرمز @ وهي في هذه الحالة متحولات الدخل A, B

ومدخل الانتقاء select. بالإضافة لعبارة السلوك always يوجد بلغة Verilog HDL

عبارة أخرى للسلوك أيضاً، هي عبارة initial. وينفذ السلوك الابتدائي Initial behavior بمجرد البدء عند اللحظة $t = 0$. تنفذ عبارة initial مرة واحدة فقط، حيث يبدأ تنفيذها بمجرد بدء عملية المحاكاة وتنتهي بعد أن ينتهي تنفيذ جميع العبارات. وينفذ السلوك always بصورة متكررة إلى أن ينتهي الزمن المقرر للمحاكاة. ولا يوجد في التوصيف المبين أدناه لمضمّن 2x1MUX فاصلة منقوطة في نهاية العبارة always، كما أن الكلمة المفتاحية or تستعمل بين المتحولات بدلاً من " | " .

// Behavioral Description of 2-to-1 Line Multiplexer

```
module mux2x1_Bh (A, B, select, Y);
    input A, B, select;
    output Y;
    reg Y;
    always @ (select or A or B)
        if (select == 1) Y = A;
        else Y = B;
```

endmodule

تؤمن العبارة الشرطية if-else قراراً يعتمد على قيمة مدخل الانتقاء، ويمكن كتابة العبارة if من دون رمز المساواة، كما في الشكل: $Y = A; \text{ if (select)}$ ويعني هذا الإسناد أنه يفحص ما إذا كانت قيمة select مساوية 1. وباستعمال المؤثر الشرطي يوصف نفس المثال أعلاه باستخدام طريقة تدفق المعطيات على الشكل التالي:

// Dataflow Description of 2-to-1 Line Multiplexer

```
module mux2x1_Df (A, B, select, Y);
    input A, B, select;
    output Y;
    assign Y = select ? A : B;
endmodule
```


13-6 لغة وصف العتاديات للدارات التتابعية

يستخدم كما أشرنا سابقاً لوصف الدارات المنطقية التركيبية بلغة وصف العتاديات، تقنيات مستوى البوابات أو تعابير جريان المعطيات، بينما يستخدم لوصف عمل القلابات تقنية النمذجة السلوكية. وبما أن الدارات التتابعية تصنع من القلابات والبوابات، فإنه يمكن توصيفها بمزيج من التعابير السلوكية وتدفق المعطيات، بحيث يوصف الجزء التركيبي للدارة التتابعية بعبارات إسناد ومعادلات بوليا نية ويوصف القلاب بعبارات السلوك. يوجد بلغة Verilog HDL نوعان من عبارة السلوك هما: `initial` و `always`. يمكن للمجترأ الواحد أن يتضمن عدداً اختيارياً من عبارات `initial` و `always`. ويمكن أن تنفذ هذه العبارات على التوازي فيما بينها بدءاً من اللحظة $t = 0$. يمكن بالعبارة `always` التحكم بتأخيرات تنتظر تحقق شرط ما (حدوث حادثة ما). الشكل العام لصيغة عبارة " تعبير حادثة التحكم" هي من الشكل التالي:

(event control expression) `@ always`، وسنشير إلى نوعين من الحوادث:

- الحوادث التي تتحسس للمستوى Level-Sensitive

- الحوادث التي تتحسس لحافة القذح Edge-Sensitive

فمثلاً، العبارة `@ (A or B or Reset) always` تسبب تنفيذ العبارات الإجرائية الموجودة في كتلة `always` إذا حدث تغير في A أو B أو Reset. تميز لغة Verilog HDL بين القذح بالحافة الموجبة أو السالبة وتخصص لهما الكلمتان المفتاحيتان `posedge` و `negedge`.

يوجد نوعان من الإسنادات الإجرائية: تكتيلي Blocking وغير تكتيلي Non-blocking. ويميز بينهما بالرمز المستخدم. تستخدم الإسنادات التكتيلية الرمز `(=)` كمؤثر إسناد، وتستخدم الإسنادات غير التكتيلية الرمز `(=>)`. تنفذ العبارات التكتيلية تتابعياً بنفس ترتيب ورودها في الكتلة التتابعية، في حين تحسب العبارات غير التكتيلية العبارة الموجودة في الجانب الأيمن، لكن لا تتم عملية الإسناد إلى الجانب الأيسر قبل أن يتم حساب جميع العبارات. أول ما يلزم عند محاكاة دارة تتابعية هو توليد النبضات الساعية لقذح القلابات التي قد تكون موجودة بالدارة، وبالتالي لابد من استخدام عبارات التكرار أو الحلقات، حيث

يجب أن تظهر جميع عبارات الحلقات داخل كتلة initial و always. نكتفي بهذا الحد من التوصيف النظري للغة وصف العتاديات، وننتقل إلى التطبيق بعرض أمثلة حول الدارات التتابعية التي تعرفنا عليها وذلك من خلال مثالين: الأول: لتوصيف قلاب ويتضمن شكلين لمسألة واحدة، لإظهار السبل الممكنة للتوصيف:

الشكل الأول- يتضمن وصفاً لمجترأ الماسك D بمدخلين (مدخل للمعطيات ومدخل للتحكم) ومخرج وحيد Q. وبما أن المخرج Q يقيم بعبارة إجرائية، يجب الإعلان عنه باعتباره نوع سجل reg type. تستجيب المواسك إلى مستويات إشارة الدخل، لذلك أدرج المدخلان من دون حافة تأهيل في تعبير التحكم في الحدث الذي يلي الرمز @ في Always، وفق ما هو مبين في التوصيف أدناه.

```
// Description of D Latch
module D_Latch ( Q, D, Control);
output Q;
input D, Control;
reg Q;
always @ (Control or D)
if (Control) Q = D; // Same as: if ( Control ==1)
endmodule
```

يوجد في التوصيف المبين أعلاه، عبارة إسناد إجرائية تكتيلية واحدة فقط تحدد نقل الدخل D إلى المخرج Q إذا كان التحكم محققاً (المنطق 1). وتنفذ العبارة في كل مرة يطرأ فيها تغيير في قيمة D في الوقت الذي يكون فيه التحكم 1.

الشكل الثاني- ويتضمن توصيفاً لقلابين من نوع D يقدر كل منهما بالحوافة الموجبة، وبالتالي يلزم مجترأين. يستجيب الأول إلى الساعة فقط، ويتضمن الثاني مدخل تصفير غير متزامن. وفيه يجب التصريح عن المخرج Q من نوع معطيات reg إضافة إلى كونه قد أدرج باعتباره مخرجاً، وذلك لأنه مخرج هدف في عبارة إسناد إجرائية.

```
// Description of D Flip Flop
module D_Latch ( Q, D, CLK);
output Q;
input D, CLK;
reg Q;
```

```

always @ (posedge CLK)
Q = D;
endmodule
// Description of D F.F with asynchronous Reset
module DFF ( Q, D, CLK, RST);
output Q;
input D, CLK, RST;
reg Q;
always @ (posedge CLK or negedge RST)
if (~ RST) Q = 1'b0; // Same as: if ( RST ==0)
else Q = D;
endmodule

```

تعني الكلمة المفتاحية `posedge` أن انتقال الدخل D إلى Q سيحدث خلال الانتقال الموجب لحافة الساعة، وأن التغيرات التي تحصل في الدخل D في أي وقت آخر لا تغير من قيمة Q. ويتضمن المجتزأ الثاني مدخلاً تصفيرياً غير متزامن، إضافة إلى ساعة التزامن. ويمكن أن يكون لتغير الحدث الذي يلي الرمز @ في العبارة `always`، أي عدد من الأحداث ذات الحافة الموجبة أو السالبة. ويجب أن يكون أحد هذه الأحداث هو حدث الساعة. وما تبقى يحدد الشروط التي ينفذ بناءً عليها منطق غير متزامن. فكل عبارة `if` أو `else if` في عبارات الإسناد الإجرائية توافق حدثاً غير متزامن. توافق عبارة `else` الأخيرة حدث الساعة. يوجد في المجتزأ الثاني حدثاً حافة. فحدث الحافة السالبة `RST negedge` هو غير متزامن باعتباره يماثل عبارة `if (~ RST)`. وتبقى قيمة Q صفراً ما دامت قيمة `RST = 0`. وعندما تكون قيمة `RST = 1` يمكن لحدث الحافة الموجب للـ CLK أن تنقل D إلى Q تزامنياً.

الثاني: يوصف مخطط حالة لدارة تتابعية من الدارات التي تم توصيفها بهذا الفصل ويتضمن شكلين لمسألة واحدة أيضاً. والشكلان اللذان سيتم إظهارهما أدناه يمثلان السبل الممكنة لإظهار توصيف عمل الدرة التتابعية باستخدام لغة فيريلوغ بالصيغة ذاتها التي تتم بمخطط الحالة للدارة، أيأ كان نوعها ، بحيث:

الشكل الأول- يتضمن وصفاً لمخطط حالة لدارة تتابعية نموذج ميلي. ويكون التوصيف بلغة

HDL وفق التالي:

// Description Mealy State Diagram

```
module Mealy ( x, y, CLK, RST);  
output y;  
input x, CLK, RST;  
reg y;  
reg [1:0] prstate, nxtstate;  
parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;  
always @ (posedge CLK or negedge RST)  
if (~ RST) prstate = s0; // initialize to state s0  
else prstate = nxtstate; // Clock Operation  
always @ (prstate or x); // Determine Next State  
case (prstate)  
s0: if (x) nxtstate = s1;  
else nxtstate = s0;  
s1: if (x) nxtstate = s3;  
else nxtstate = s0;  
s2: if (~x) nxtstate = s0;  
else nxtstate = s2;  
s3: if (x) nxtstate = s2;  
else nxtstate = s0;  
endcase  
always @ (prstate or x) // Evaluate output  
case (prstate)  
s0: y = 0;  
s1: if (x) y = 1'b0; else y = 1'b1;  
s2: if (x) y = 1'b0; else y = 1'b1;  
s3: if (x) y = 1'b0; else y = 1'b1;  
endcase  
endmodule
```

وقد صرح عن الدخل والخرج والساعة والتصفير (الإرجاع) بالطريقة المعتادة، وعن حالات الدارة بالمعرفات prstate و nxtstate، حيث لغة Verilog تسمح بتعريف الثوابت في المجتزأ بواسطة الكلمة المفتاحية **parameter**. وبالتالي يلاحظ إسناد القيم الثنائية 00 حتى 11 للحالات من s_0 إلى s_3 بالصيغة $s_2 = 2'b10$ (الذي يظهر بتنين لتخزين الثابت)

عوضاً عن الشكل $s_2 = 2$ (الذي يظهر عدداً ثنائياً ذا 32 بتاً). يستخدم في المثال أعلاه ثلاث كتل always تنفذ على التوازي وتتفاعل فيما بينها بواسطة متحولات مشتركة، بحيث ترجع عبارة always الأولى الدارة إلى الحالة الابتدائية $s_0 = 00$ وتحدد العملية المتزامنة المحكومة بساعة. وتنفذ العبارة $prstate = nxtstate$ استجابة إلى انتقال الحافة الموجبة للساعة فقط. وهذا يعني أن أي تغير يحدث في قيمة الحالة التالية $nxtstate$ في كتلة always الثانية، ينقل إلى الحالة الراهنة $prstate$ نتيجة الحدث $posedge$ (الحافة الموجبة). تعين كتلة always الثانية الانتقال للحالة التالية بوصفها تابعاً للحالة الراهنة والدخل. ويتبع شرط التفريع المتعدد المسالك التابع المحدد بمخطط الحالة المعطى. تحسب كتلة always الثالثة الخرج كتابع للحالة الراهنة والدخل.

الشكل الثاني- ويتضمن وصفاً لمخطط حالة لدارة تتابعيه نموذج مور. يكون التوصيف بلغة

HDL وفق التالي:

```
// Description Moore State Diagram
module Moore ( x, AB, CLK, RST);
output [1:0]AB;
input x, CLK, RST;
reg [1:0] state;
parameter s0 = 2'b00, s1 = 2'b01, s2 = 2'b10, s3 = 2'b11;
always @ (posedge CLK or negedge RST)
if (~ RST) state = s0; // initialize to state s0
else case (state)
s0: if (~x) state = s1; else state = s0;
s1: if (x) state = s2; else state = s3;
s2: if (~x) state = s3; else state = s2;
s3: if (~x) state = s0; else state = s3;
endcase
assign AB state; // output of F.F
endmodule
```

وفيما يلي نعرض وصف العتاديات لكل من السجلات والعدادات بوصفهما دارات تتابعية.

أولاً- لغة وصف العتاديات للسجلات

يمكن وصف السجلات بلغة وصف العتاديات HDL على مستوى السلوك أو على مستوى البنية. فعلى مستوى السلوك يوصف السجل بوصف مختلف العمليات التي يؤديها بما يشبه جدول الوظائف. أما الوصف على مستوى البنية، فيعرض الدارة بدلالة مجموعة من المكونات الأساسية (بوابات وقلابات ومضّمات). وتستسخ Instantiated المكونات المختلفة لتشكل وصفاً هرمياً للتصميم يشبه التمثيل بواسطة المخطط المنطقي. ونبين كلا النوعين من التوصيف من خلال توصيف السجل متعدد الأغراض.

أولاً- على مستوى السلوك: يكون التوصيف السلوكي الذي يقابل جدول وظائف لسجل بطول 4 bits وبمدخلين للانتخاب (s_0, s_1) ومدخلين لإدخال معطيات الإزاحة من اليمين واليسار وأربعة مداخل $I_0, \dots, I_3$ وأربعة مخارج $O_0, \dots, O_3$ وفق ما هو مبين أدناه.

// Behavioral Description of Universal Shift Register

```
module shftreg (s1, s0, lin, rin, pin, O, CLK, CLR);
    input s1, s0; // Selection lines of Multiplexer
    input lin, rin; // Serial inputs, where lin: means Left In and
// rin: means Right in
    input CLK, CLR; // Clock and Clear Inputs
    input [3:0] pin; // Four Pins for Parallel Inputs  $I_0, \dots, I_3$ 
    output [3:0] O; // Four Parallel Output of Register  $O_0, \dots, O_3$ 
    reg [3:0] O;
    always @ (posedge CLK or negedge CLR)
        if (~ CLR) O = 4'b0000;
        else
            case ({s1, s0})
                2'b00: O = O; // No Change
                2'b01: O = {rin, O[3:1]}; // Shift Right
                2'b10: O = {O[2:0], lin}; // Shift Left
                2'b11: O = pin; // Parallel Load Inputs
            endcase
endmodule
```


ثانياً - على مستوى البنية: يمكن وصف بنية السجل المكون من أربعة مضمّات وأربعة قلابات وفق التالي:

```
// Structural Description of Universal Shift Register
module SHFTREG (I, select, lin, rin, O, CLK, CLR);
    input [3:0] I; //Parallel input
    input [1:0] select; // Mode Select
    input lin, rin, CLK, CLR;
    output [3:0] O;
    // Instantiate the four stages
    Stage ST0 (O[0], O[1], lin, I[0], O[0], select, CLK, CLR);
    Stage ST1 (O[1], O[2], O[0], I[1], O[1], select, CLK, CLR);
    Stage ST2 (O[2], O[3], O[1], I[2], O[2], select, CLK, CLR);
    Stage ST3 (O[3], rin, O[2], I[3], O[3], select, CLK, CLR);
endmodule

// one stage of Shift Register
module stage (i0, i1, i2, i3, Y, select, CLK, CLR);
    input i0, i1, i2, i3, CLK, CLR;
    input [1:0] select;
    output Y;
    reg Y;
    reg D;

    // 4x1 Multiplexer
    always @ (i0 or i1 or i2 or i3 or select)
        case (select)
            2'b00: D = i0;
            2'b01: D = i1;
            2'b10: D = i2;
            2'b11: D = i3;
        endcase

    // D flip Flop
    always @ (posedge CLK or negedge CLR)
        if (~ CLR) Y = 1'b0;
        else Y = D;

endmodule
```


ثانياً- لغة وصف العتاديات للمعدات

يمكن وصف العتادات بلغة وصف العتاديات HDL، مثلما مر معنا بالنسبة لتوصيف السجلات، وذلك على مستوى السلوك أو على مستوى البنية. وفيما يلي نكتفي بعرض مثال واحد حول توصيف عداد على مستوى السلوك من دون الدخول بالتفاصيل. وليكن العداد الذي يمثل عدداً متزامناً بدخل متوازٍ يعد ثماني قيم فقط. ولتوصف مختلف العمليات التي يقوم بها العداد، يكتب:

// Behavioral Description of Binary Counter with Parallel Load.

```
module counter (count, load, IN, CLK, CLR, O, CO);
    input count, load, CLK, CLR;
    input [2:0] IN; // Three Parallel Inputs Data, I0, ..., I2
    output CO; // Carry Out
    output [2:0] O; // Data Output
    reg [2:0] O;
    assign CO = count & ~ load & ( O == 3'b111);
    always @ (posedge CLK or negedge CLR)
        if (~ CLR) O = 3'b000;
        else if (load) O = IN;
        else if (count) O = O + 1'b1;
        else O = O;
endmodule
```

وتحدد الكتلة البرمجية always العملية المطلوب تنفيذها اعتماداً على قيم كل من المداخل Clear, Load, Count. إذا كان CLR غير فعال (CLR = 1)، فستنفذ عملية واحدة من بين ثلاث عمليات عند ورود الوجه الهابط للنبضة الساعية. وتتخذ العبارات البرمجية if و else if و else القرارات كما يلي:

If CLR = 0 , Clear O to 0.

Else if (CLR = 1 and) Load = 1 , Load Inputs to O.

Else if (CLR = 1 and Load = 0 and) Count = 1 , Increment O.

Else (CLR = 1 and Load = 0 and Count = 0) , No Change in O.

الفصل الرابع عشر: المنطق القابل للبرمجة

تُعدّ ذاكرة ROM المخصصة للقراءة فقط واحدة من التجهيزات المنطقية القابلة للبرمجة العتادية PLD (Programmable Logic Device). تشير كلمة برمجة هنا إلى الإجرائية العتادية التي تحدّد البتات الثنائية المدخلة ضمن التشكيلة العتادية للتجهيزية. وتحدّد المعلومات الثنائية المخزنة بداخل التجهيزات المنطقية القابلة للبرمجة بطريقة ما، ثم تضمّن Embedded في العتاديات وتعرف هذه العملية ببرمجة التجهيزية. وأصبحت التجهيزات المنطقية القابلة للبرمجة على ضوء التطور التقني المتسارع لمختلف الدارات المنطقية الأخرى محدودة الانتشار، وعلى الرغم من ذلك يفرد لها بهذا الكتاب فصلاً قصيراً من باب إتمام مواضيع الكتاب، ونميز هنا بين نوعين من هذه التجهيزات هي:

- التجهيزات التركيبية المنطقية القابلة للبرمجة Combinational PLD
- التجهيزات التتابعية المنطقية القابلة للبرمجة Sequential PLD

14-1 التجهيزات التركيبية المنطقية القابلة للبرمجة

تمثل التجهيزية المنطقية القابلة للبرمجة دائرة متكاملة، يوجد بداخلها بوابات موصولة ببعضها بواسطة ممرات الكترونية، يشبه سلوكها سلوك الفواصم Fuses. تكون في الحالة الأصلية للتجهيزية جميع المنصيرات (الفواصم) بحالة توصيل. وتتضمن عملية برمجة التجهيزية حرق الفواصم الموجودة في الممرات الواجب إزالتها للحصول على التشكيلة Configuration الخاصة للتابع المنطقي المطلوب. تتألف التجهيزات المنطقية التركيبية القابلة للبرمجة من بوابات منطقية فقط. فمثلاً، تجهيزية منطقية تركيبية مثل PROM تمثل دائرة متكاملة مكونة من بوابات قابلة للبرمجة مقسمة إلى صفيقة من البوابات AND وصفيقة من البوابات OR لتتجزى مجموع جداءات AND-OR. ويُعدّ مصطلح مصفوفة أو صفيقة هنا مجازياً وهو مستمد من مصطلح Array. يوجد ثلاثة أنواع من التجهيزات المنطقية القابلة للبرمجة التركيبية، تختلف فيما بينها بطريقة توضع الوصلات القابلة للبرمجة في صفيقة البوابات AND-OR، هي:

١٠ - ذاكرة قراءة فقط قابلة للبرمجة PROM

١١ - منطق صفيقة قابلة للبرمجة PLA من Programmable Logic Array

١٢ - صفيقة منطقية قابلة للبرمجة PAL من Programmable Array Logic

- يتضمن النوع الأول، أي ذاكرة PROM صفيقة ثابتة من البوابات AND تمثل مفكك الترميز وصفيقة من بوابات OR قابلة للبرمجة. وتنجز البوابات OR القابلة للبرمجة التتابع البوليانية بصيغة مجموع حدود جداءات أصغريه.

- يُعد النوع الثاني، أي PLA من أكثر التجهيزات القابلة للبرمجة مرونة، لأنه يمكن فيه برمجة كل من صفيقتي البوابات AND والبوابات OR.

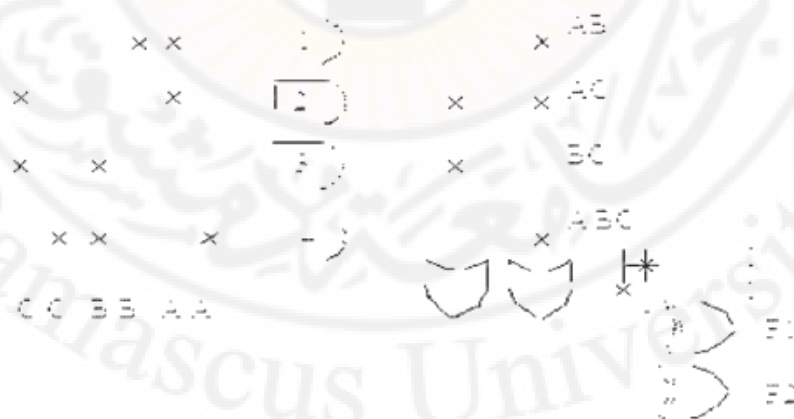
- يوجد في النوع الثالث PAL صفيقة من البوابات AND قابلة للبرمجة، وصفيقة ثابتة من بوابات OR غير قابلة للبرمجة. تبرمج البوابات AND، لتعطي حدود الجداء للتتابع البوليانية التي تجمع منطقيا في كل بوابة OR. ونتعرف فيما يلي على النوعين PLA و PAL بعد أن سبق وتعرفنا على النوع الأول.

1-1-14 التشكيل PLA

يشبه المنطق PLA الذاكرة PROM باستثناء أن PLA لا يقوم بتفكيك ترميز لكامل المتحولات ولا بتوليد جميع الحدود الأصغرية، حيث استبدل مفكك الترميز بصفيقة (مصفوفة) من بوابات AND التي يمكن برمجتها لتوليد أي حد من حدود جداء متحولات الدخل، توصل حدود الجداء بعدئذ إلى بوابات OR لتنتج مجموع الجداءات للتتابع المنطقية المطلوبة. وهنا نشير إلى أن التجهيزات المنطقية القابلة للبرمجة والتي تستخدم بتصميم النظم الرقمية يمكن أن تتضمن مئات إلى ملايين البوابات التي ترتبط فيما بينها بمئات إلى آلاف الممرات الداخلية. وبالتالي كي نتمكن من بيان المخطط الداخلي بأسلوب مبسط يجب استخدام رموز خاصة لتمثيل البوابات القابلة للتطبيق في منطق المصفوفة. فمثلاً، بالنسبة لبوابة AND أو OR متعددة المداخل يستخدم الرمز التالي:



وهو الرمز التقليدي لبوابة AND متعددة المداخل (مثلاً بأربعة مداخل) المستخدم بالمنطق PLA، حيث يستبدل التمثيل الأفقي للمداخل مهما كان عددها بخط أفقي واحد على المدخل وعليه يرسم بشكل عمودي عدد المداخل، بحيث تتصل خطوط الدخل مع البوابة من خلال منصهرات داخلية Fuses. يمكن أن يكون للدائرة النموذجية المتكاملة لمصفوفة منطقية قابلة للبرمجة 16 مدخل و 48 حد جداء و 8 مخارج، إذ إن المهم دوماً بالنسبة للمنطق PLA تحديد حجم المصفوفة الذي يحدّد بعدد المداخل وعدد خطوط الجداء وعدد المخارج. فإذا كان عدد المداخل n وعدد خطوط الجداء k وعدد المخارج m ، فإن البنية الداخلية لـ PLA تتألف من n بوابة (عازل ومتمم بأن واحد) و k بوابة AND و m بوابة OR و m بوابة XOR. يوجد $k \times 2^n$ وصلة بين المداخل وبين مصفوفة البوابات AND و $n \times k$ وصلة بين مصفوفة البوابات AND ومصفوفة OR و m وصلة مترافقة مع البوابات XOR. ولتوضيح مخططات PLA وإظهار آلية تنجزها للمعادلات المنطقية، نستعرض مثالاً حول المنطق الداخلي لمصفوفة منطقية قابلة للبرمجة PLA مكوناً من ثلاثة مداخل وأربعة حدود جداء ومخرجين، كما هو مبين بالشكل 14-1.



الشكل 14-1: المنطق الداخلي لمصفوفة منطقية قابلة للبرمجة PLA مكوناً من ثلاثة مداخل وأربعة حدود جداء ومخرجين

يستخدم المخطط في الشكل 14-1 رغم بساطته نفس الرموز التي تستخدم عادة في الدارات المعقدة، حيث يمرر كل دخل عبر عازل وعاكس بأن واحد أي رمز مركب له مخرجان أحدهما عادي والآخر معكوس. ويوصل كل مدخل ومعكوسة إلى مداخل كل بوابة AND وتوصل مخارج البوابات AND إلى مداخل كل بوابة من بوابات OR. وتوصل مخارج البوابات OR إلى أحد مدخلي بوابة XOR التي يمكن برمجتها مدخلها الآخر ليستقبل إشارة 0 أو 1. يتم الحصول على معكوس خرج البوابة OR إذا كانت قيمة المدخل الآخر للبوابة XOR هو الواحد منطقي (ويحافظ المخرج على قيمته إن كان المدخل 0). والتابعان المنطقيان اللذان تحققهما دائرة PLA المبينة في الشكل أعلاه هما:

$$F_1 = AB' + AC + A'BC' , F_2 = (AC + BC)'$$

يمكن تعيين مخطط الفواصم لتجهيزية المصفوفة PLA على شكل جدول يسمى بجدول البرمجة، مثل الجدول 14-1، بحيث يتألف الجدول من أربعة أجزاء:

الجدول 14-1: جدول لبرمجة مصفوفة PLA

الجداء	حدود جاء	المداخل Inputs A B C			المخارج (T) (C) F ₁ F ₂	
		A	B	C	F ₁	F ₂
AB'	1	1	0	-	1	-
AC	2	1	-	1	1	1
BC	3	-	1	1	-	1
A'BC'	4	0	1	0	1	-

يتضمن الجزء الأول حدود الجداء والثاني لائحة بأرقام حدود الجداء ويحدد الجزء الثالث الممرات المطلوبة بين المداخل والبوابات AND، ويعرف الجزء الرابع الممرات بين مخارج البوابات AND والبوابات OR. ويمكن أن يظهر كل متحول من متحولات الخرج وفق أحد شكلين: إما حقيقي (T) أو متمم (C)، وهو الذي يحدد برمجة البوابة XOR. ويوضع تحت كل متحول من متحولات الدخل واحدة من ثلاث قيم هي: 0 أو 1 أو شحطة (-) Dash مقابل كل جداء، بحيث يوضع 1 تحت المتحول الذي يظهر بشكله الحقيقي في حد الجداء، ويوضع 0 تحت المتحول الذي يظهر بشكله المتمم، بينما الشحطة (-) يتم

وضعياً تحت المتحول الذي لا يظهر في حد الجداء. وتحدد الممرات بجدول البرمجة بين المداخل والبوابات AND بالعمود الثالث الذي يحمل ترويسة مداخل، حيث يدل 1 في عمود الدخل على وصل متحول الدخل إلى البوابة AND. ويدل 0 على وصل متمم متحول الدخل إلى البوابة AND، بينما تدل الإشارة (-) على عدم صهر الفاصمتين العائدتين لكل من المتحول ومتممه، أي عدم ظهور المتحول في حد الجداء (مدخل بوابة AND المفتوح أي غير الموصول بشيء ويتم التعامل معه وكأنه 1). وعرفت الممرات بين البوابات AND و OR بالعمود الأخير من جدول البرمجة تحت الترويسة مخارج ووضعت القيمة 1 تحت متحول الخرج مقابل حدود الجداء التي تظهر في تابع الخرج (يتطلب كل حد جداء لديه 1 في عمود الخرج ممراً من مخرج البوابة AND إلى مدخل البوابة OR). وتدل الإشارة (-) في قائمة المخارج على الفواصم التي تم صيرها. وتعد الوصلة المفتوحة (الفاصمة المقطوعة) في مداخل البوابة OR صفراً منطقياً. ويدل الحرف T في عمود الخرج على أن المدخل الآخر للبوابة XOR الموافقة موصول إلى صفر منطقي، على حين يدل الحرف C (Complement) على أنه موصول إلى 1 منطقي. ولا يوجد حاجة عند تصميم نظام منطقي باستخدام PLA لإظهار الوصلات الداخلية للوحدة وكل ما يلزم هو جدول لبرمجة المصفوفة PLA الذي يمكن بالاستفادة منه برمجة المصفوفة المنطقية القابلة للبرمجة لتقديم المنطق المطلوب. وكما هو الحال بالنسبة للذاكرة ROM يمكن برمجة الصيغة PLA بطريقتين:

- برمجة القناع: يجب على المستخدم وفق هذه الطريقة إرسال جدول برمجة الصيغة PLA إلى المصنع لتولي إنتاج تجهيزية PLA المقابلة.
- البرمجة الحقلية FPLA (Field Programmable Logic Array)، حيث يمكن برمجة الصيغة المنطقية القابلة للبرمجة حقلياً من قبل المستخدم باستخدام وحدة برمجة.

قبل تنفيذ الدارة التركيبية باستخدام PLA يجب إيجاد أقل عدد ممكن من حدود الجداء المميزة. وهذا يتم بتبسيط كل تابع من التوابع المنطقية إلى أقل عدد ممكن من حدود الجداء (عدد الرموز في الجداء غير مهم، نظراً لتوافر جميع متحولات الدخل).

14-1-2 التشكيل PAL

بالنظر إلى أن صيغة البوابات AND هي فقط القابلة للبرمجة (وصيفة بوابات OR ثابتة) في المنطق PAL ، فإن برمجة منطق الصيغة القابلة للبرمجة PAL تكون أسهل ، إلا أنها أقل مرونة من المنطق PLA. تحتوي دارة PAL المتكاملة النموذجية (المتوافرة تجارياً) على ثمانية مداخل وثمانية مخارج وثمانية مقاطع، يتألف كل منها من صيغة مؤلفة من ثماني بوابات AND متبوعة بـ OR. تقاد المخارج فيها أحياناً بوساطة عوازل أو عواكس ذات ثلاث حالات. ويجب عند التصميم باستخدام PAL تبسيط التوابع البوليانية لكي تلائم كل مقطع. ولا يمكن بالوحدة PAL (على عكس PLA) أن يشترك في حد الجداء بوابتان أو أكثر من البوابات OR، بالتالي يمكن تبسيط كل تابع بصورة مستقلة عن الآخر. عدد حدود الجداء التي يمكن تشكيلها في كل مقطع ثابت، وبالتالي إذا كان عدد الحدود في التابع أكبر من عدد حدود الجداء التي يمكن تشكيلها بوساطة مقطع واحد، يمكن استخدام مقطعين لتحقيق تابع واحد. وفيما يلي نبين من خلال مثال محدد استخدام PAL في تصميم دارة تركيبية.

مثال: صمم باستخدام التجهيزية PAL الدارة المنطقية التركيبية التي تحقق التوابع

$$F_1(a, b, c, d) = \sum (2, 12, 13) \quad \text{البوليانية:}$$

$$F_2(a, b, c, d) = \sum (7, 8, 9, 10, 11, 12, 13, 14, 15)$$

$$F_3(a, b, c, d) = \sum (0, 2, 3, 4, 5, 6, 7, 8, 10, 11, 15)$$

الجواب: نبدأ بتبسيط التوابع المعطاة، فنحصل على الأشكال

$$F_1 = abc' + a'b'cd'$$

$$F_2 = a + bcd$$

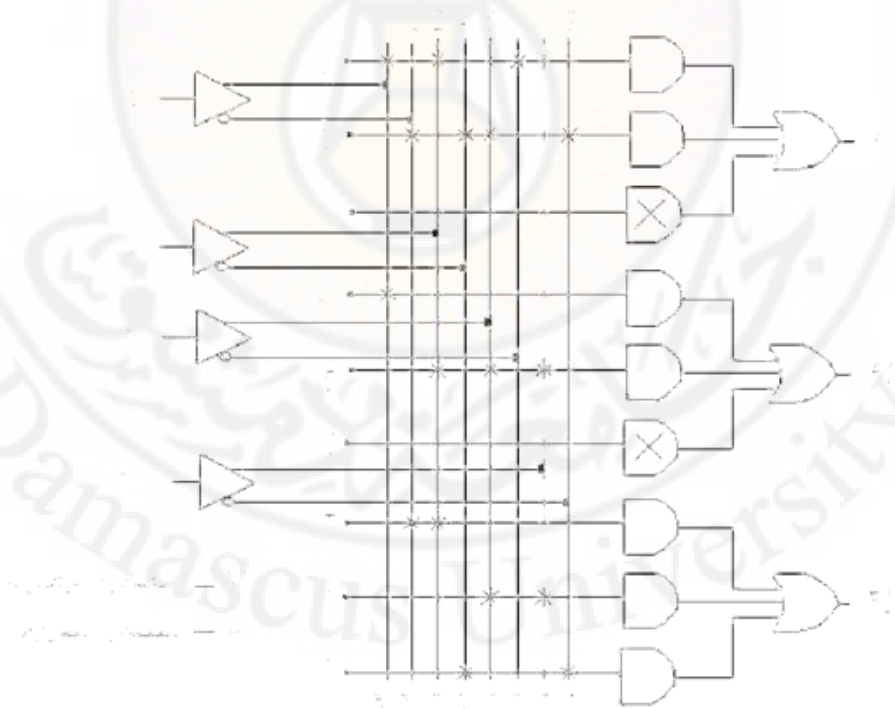
$$F_3 = a'b + cd + b'd'$$

يبني جدول برمجة PAL المقابل للتوابع المعطاة، كما هو مبين في الجدول 2-14، بحيث يشبه جدول برمجة PLA ماعدا مداخل AND التي تحتاج إلى برمجة. ويتضمن جدول البرمجة لـ PAL للتوابع المنطقية الثلاثة المعطاة، ثلاثة مقاطع، يتألف كل منها من ثلاثة حدود جداء (ليتوافق مع منطق الصيغة القابلة للبرمجة).

الجدول 14-2: جدول لبرمجة PAL

المخارج	مداخل البوابة AND a b c d	حدود جاء
$F_1 = abc' + a'b'cd'$	1 1 0 -	1
	0 0 1 0	2
	- - - -	3
$F_2 = a + bcd$	1 - - -	4
	- 1 1 1	5
	- - - -	6
$F_3 = a'b + cd + b'd'$	0 1 - -	7
	- - 1 1	8
	- 0 - 0	9

يبين الشكل 14-2 مخطط الفواصم لمنطق PAL، كما تم تحديده في جدول البرمجة.



الشكل 14-2: مخطط الفواصم لمنطق PAL، كما تم تحديده في جدول البرمجة

يعلم التقاطع الموافق لكل من 1 أو 0 على المخطط بالرمز :

X- للمنصورة الموصولة، حيث جميع المنصهرات الموصولة تقابل الصفر .

+ - للمنصورة (أو الفاصمة) المحروقة.

ويعلم على المخطط التقاطع المقابل لكل إشارة (-) بجداول البرمجة بفاصمة محروقة للمداخل ومتمماتها. وتترك جميع مداخل البوابات AND غير المستخدمة على حالها (الفواصم موصولة ويكون مخرج AND دائماً صفراً).

14-2 التجهيزات التتابعية المنطقية القابلة للبرمجة

تضم التجهيزات التتابعية القابلة للبرمجة بالإضافة للبوابات، القلايات أيضاً. يمكن برمجة التجهيزات القابلة للبرمجة لانجاز العديد من وظائف الدارات التتابعية. ويتوافر العديد من أنواع التجهيزات التتابعية القابلة للبرمجة بشكل تجاري، إلا أن المنطق الداخلي لهذه التجهيزات معقد، وسنكتفي هنا بوصف مختصر للأنواع الرئيسية لهذه التجهيزات وهي:

- التجهيزات المنطقية القابلة للبرمجة البسيطة SPLD من

Simple Programmable Logic Device

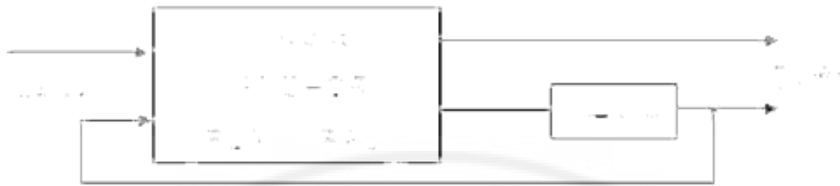
- التجهيزات المنطقية القابلة للبرمجة المعقدة CPLD من

Complicated Programmable Logic Device

- صفيقة البوابات القابلة للبرمجة حقلياً FPGA من

Field Programmable Gate Array

يشار أحياناً إلى التجهيزية المنطقية القابلة للبرمجة التتابعية باسم تجهيزه منطقية قابلة للبرمجة بسيطة لتمييزها عن التجهيزية المنطقية القابلة للبرمجة المعقدة. جميع الأسماء المشار إليها أعلاه مثل PAL و PLA وغيرها صدرت من مصنعين مختلفين خلال عملية تطوير تجهيزات المنطق القابل للبرمجة. تتضمن تجهيزات SPLD الموجودة بداخل الدارة المتكاملة قلايات إضافة إلى صفيقة البوابات AND-OR. والنتيجة هي دارة تتابعية، مثل الدارة المبينة في الشكل 3-14. وتعزل التجهيزية PAL أو PLA بتضمينها عدداً من القلايات توصل لتكون سجلاً. ويمكن أخذ مخارج الدارة من مخارج البوابات OR، أو من مخارج القلايات.



الشكل 3-14: دائرة تتبعية تتضمن تجهيزات SPLD

تتوافر توصيلات قابلة للبرمجة إضافة لتضمين مخارج القلايات في حدود الجداء المكونة بصفيقة البوابات AND. ويمكن أن تكون القلايات من النوع D أو J-K. تتضمن تجهيزية SPLD النموذجية من 8 إلى 10 خلايا ماكروية Macro cell ضمن حافظة دائرة متكاملة واحدة (حيث كل مقطع في الـ SPLD يسمى خلية ماكروية، وهي دائرة تحتوي على مجموع جداءات لوظائف المنطق التركيبي وعدد اختياري من القلايات). وتوصل جميع القلايات في SPLD إلى ساعة عامة، ويجري التحكم بجميع بوابات دائري بواسطة مدخل تأهيل OE. وتتألف CPLD من عدد من التجهيزات القابلة للبرمجة تتصل فيما بينها عن طريق مصفوفة مفاتيح قابلة للبرمجة، وفق ما هو مبين في الشكل 4-14.



الشكل 4-14: مصفوفة مفاتيح قابلة للبرمجة

تمثل التجهيزية CPLD، مجموعة من التجهيزات المنطقية القابلة للبرمجة مجتمعة في دائرة متكاملة واحدة وهي أكثر اقتصادية في التطبيقات التي تتطلب وصل مجموعة من التجهيزات لتحقيق جميع المواصفات. ويقاد كل مغرز من مغارز الدخل والخرج بواسطة بوابة عازل ذي ثلاث حالات، ويمكن برمجتها لتعمل كمدخل أو كمخرج. وتستقبل مصفوفة المفاتيح

المداخل وتوجيهها إلى الخلايا المايكروية المستقلة، وبالمثل ترسل المخارج المنتقاة من الخلايا المايكروية إلى المخارج بحسب الحاجة. وتحتوي كل تجهيزية PLD على 8 إلى 16 خلية مايكروية متصلة مع بعضها، بحيث إذا كان لدى إحدى الخلايا بعض حدود الجداء غير مستخدمة يمكن استخدامها من الخلايا المجاورة. يبرمج قلاب الخلية المايكروية ليعمل كقلاب J-K و T أو D. وأخيراً بخصوص صفيحة البوابات القابلة للبرمجة حقلياً FPGA، تتألف من جداول بحث Look-Up Tables، ومن مضمّمات وبوابات وقلابات. وجداول البحث هي جداول حقيقة مخزنة في ذواكر RAM تنتج توابع الدارات التركيبية للكتلة المنطقية. فمثلاً، يمكن تخزين جدول الحقيقة لدارة تركيبية ذات أربعة مداخل ومخرجين في ذاكرة RAM بالحجم 16x2. ميزة استخدام ذاكرة RAM لتخزين جدول الحقيقة بدلاً من ذاكرة ROM، هي أنه يمكن برمجة الجدول بالكتابة في الذاكرة. أما السلبية فهي أن RAM من النوع المؤقت وهذا يعني الحاجة إلى إعادة تحميل محتوى الجدول في حال انقطاع التغذية. ويتطلب التصميم باستخدام الأنواع الرئيسية للتجهيزات SPLD و CPLD و FPGA أدوات تصميم بمعونة الحاسوب CAD واسعة النطاق (مثل لغات وصف البني العتادية HDL و VHDL و ABEL) لتسهيل إجرائية التركيب.

الفصل الخامس عشر: القسم العملي

يتضمن القسم العملي عرض لبعض التجارب العملية للمواضيع النظرية الواردة في كتاب الدارات المنطقية، بالتالي يلزم لإجراء بعض التجارب بكلفة منخفضة تأمين لوحات لإجراء التجارب تسمى Bread-board، حيث يمكن غرز أي دائرة تكاملية أو أي عنصر آخر بأي مساحة فارغة من اللوحة. يلزم كل لوحة وحدة تغذية تؤمن 5V DC، بحيث يصبح الشكل العام للوحة بعد وضع بعض العناصر عليها كما هو مبين في الشكل أدناه.

يلزم بعض التجارب بحسب اسم التجربة عناصر محددة من دارات متكاملة يحددها نوع التجربة وبعض العناصر الأخرى مثل المقاومات، أسلاك توصيل ومفاتيح (8bit switch) كوححدات دخل للمعطيات ولمبات إشارة LEDs كوححدات خرج لإظهار الناتج وأرقام Seven Segments وقد يلزم بعض التجارب المتقدمة مولد نبضات ساعية وغيرها. وقيل المباشرة بعرض أسماء بعض التجارب، نعرض فيما يلي توزيع مأخذ بعض الدارات المتكاملة التي تغطي معظم التجارب المقترحة، والتي تساعد في توصيل وتنفيذ التجربة بسرعة دون العودة إلى أي كاتالوك آخر، حيث من مجمل الدارات المتكاملة المصنعة بالتقانة TTL التي يمكن استخدامها في التجارب، نعرض توصيف مجموعة من الدارات المتكاملة وفق ما هو مبين بالاشمال أدناه، مضافاً لها توصيف الستة دارات التي سبق وتم توصيفها بالقسم النظري وهي

- 1- الدارة 7400: وتمثل أربع بوابات NAND كل منها بمدخلين (الشكل 3-7 الصفحة P190)
- 2- الدارة 7483: وتمثل دارة جامع / طارح ثنائي بعرض 4 bit (الشكل 17-8 الصفحة P230)
- 3- الدارة 74138: وتمثل مفكك ترميز بثلاث مداخل 3x8 Decoder (الشكل 32-8 الصفحة P249)
- 4- الدارة 7476: وتمثل قلابين بدارة واحدة كل منهما من النوع J-K (الشكل 21-9 الصفحة P298)
- 5- الدارة 7491: وتمثل سجل بطول 8 bit من نوع SISO (الشكل 8-10 الصفحة P310)
- 6- الدارة 7493: التي يمكن أن تستخدم كعداد إلى 2 وإلى 8 وإلى 16 (الشكل 15-11 الصفحة P340)

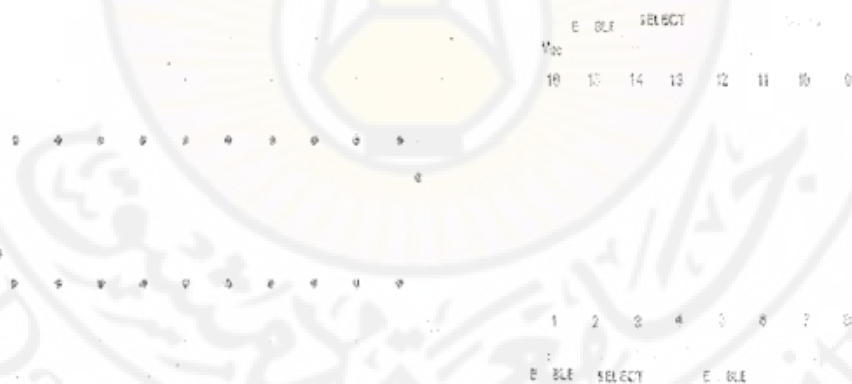


الشكل 1-15: الدارة 7404 - ست بوابات NOT
والدارة 7407 - ست بوابات Buffer

الشكل 2-15: الدارة 7432 - أربع بوابات OR كل منها بمدخلين
والدارة 7408 - أربع بوابات AND كل منها بمدخلين

الشكل 3-15: الدارة 7411 - ثلاث بوابات AND كل منها بثلاثة مداخل
والدارة 7402 - أربع بوابات NOR كل منها بمدخلين

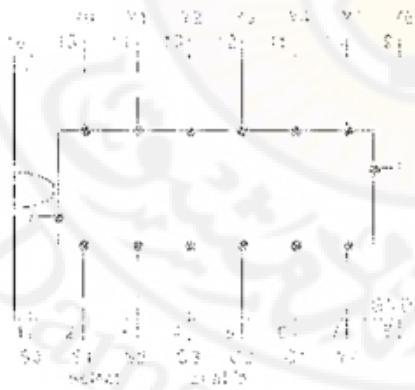
الشكل 4-15: الدارة 7427 - ثلاث بوابات NOR كل منها بثلاثة مداخل
والدارة 7410 - ثلاث بوابات NAND كل منها بثلاثة مداخل



الشكل 5-15: الدارة 74139 - تمثل مفككي ترميز كل منهما 2x4 Decoder
والدارة 74154 - تمثل مفكك ترميز بالتركيبة 4x16 Decoder



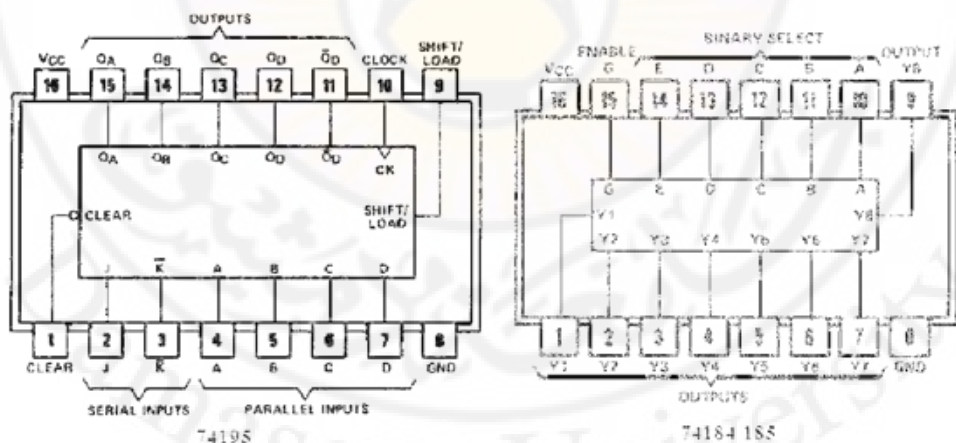
الشكل 6-15: الدارة 7446 - تمثل BCD to 7- Segment Decoder



الشكل 7-15: الدارة 74147 - تمثل مرمر 10x4 Encoder

والدارة 74153 - تمثل معاقب 8x1 Multiplexer

الشكل 8-15: الدارة 7485 - مقارن بين عددين بطول 4 bits
والدائرة 7486 (74136) - أربع بوابات XOR كل منها
بمدخلين



الشكل 9-15: الدارة 74184/185 - تمثل BCD-to-Binary Converter
Binary-to-BCD Converter with OC
والدائرة 74195 - تمثل 4-bits Shift Register PIPO



الشكل 10-15: الدارة 74390 - تمثل عدادين من نوع D

والدارة 74390 - تمثل عدادين (Ripple Counter) الشكل 10-15: الدارة 7474 - تمثل قلابين من نوع D

نكتفي حالياً بعرض أسماء التجارب دون الدخول بتفاصيل المخطط المنطقي علماً إن بعض التجارب يمكن استخدامها أيضاً لمقرر آخر هو تنظيم الحاسوب.

التجربة الأولى: وتخصص للتحقق من سلوك البوابات المنطقية من المستويات الثلاثة: NOT, AND, OR من المستوى الأول و NAND, NOR من المستوى الثاني و XOR, XNOR من المستوى الثالث. وتشكيل البوابات المنطقية الأساسية من البوابات القياسية NAND, NOR والتحقق من جداول الحقيقة العائدة لكل من هذه البوابات على حدة. وقد يلزم مثل هذه التجربة الدارات المتكاملة التالية: 7408, 7404, 7402, 7400, 7486, 7332 بمعدل دائرة واحدة من كل نوع للمجموعة الواحدة.

التجربة الثانية: التحقق من صحة قوانين ديمورغان بتطبيق دارات تقابل العلاقتين المنطقيتين: $(X + Y)' = X' \cdot Y'$ و $(X \cdot Y)' = X' + Y'$ وذلك بإجراء تجربة وفق التالي: التحقق بداية من العبارة $(X + Y)'$ برسم المخطط المنطقي المقابل لها وفق ما هو مبين في الشكل 11-15، ومن ثم

التحقق من المخطط المنطقي المقابل للعبارة $X'Y'$ ، حيث إن المخططين على الرغم من اختلافهما بالشكل يجب أن يؤديا نفس النتيجة بالعبارتين.

الشكل 11-15: المخطط المنطقي للعبارتين المتساويتين $(X+Y)'$, $X'Y'$

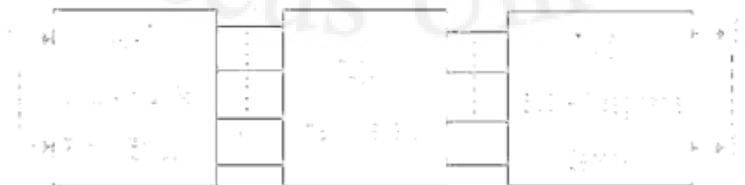
التجربة الثالثة: تنفيذ دارة الجامع التام أو تحقيق دارة جامع أو طارح تام باستخدام دارتي نصف جامع أو نصف طارح والتحقق من جداول الحقيقة العائدة لكل منها.

التجربة الرابعة: تصميم وتجريب دارة ضارب لـ 4 bit بـ 3 bit مثلاً باستخدام البوابات المنطقية أو باستخدام دارات جامع ثنائي بطول 4 bit مثل الدارة 7483.

التجربة الخامسة: تصميم وتجريب دارة جامع (طارح) BCD لجمع (أو لطرح) عشرين كل منهما مكون من رقمين باستخدام دارة جامع ثنائي 4 bit.

التجربة السادسة: تصميم دارة مفكك ترميز 2x4 Decoder والتحقق من جداول الحقيقة العائدة له. بالطبع يوجد دارة متكاملة تضم دارتي 2x4 Decoder ولكن الغاية من التجربة هي التأكد من صحة المخطط المنطقي لمفكك ترميز.

التجربة السابعة: تطبيق على المرمز لإظهار الأرقام العشرية باستخدام السبع مقاطع المضئية، واختبار المخطط العام بالدارات المتكاملة المبينة بالشكل أدناه:



التجربة الثامنة: حول المضمم والمفرق (Multiplexer & Demultiplexer) والتحقق من جداول الحقيقة العائدة لكل منهما. وكذلك تصميم وتجريب تشكيل مضمم كبير غير متوفر باستخدام مضممات أصغر متوفرة (مثل تصميم 8x1 MUX باستخدام مضممات من رتبة 4x1 MUX).

التجربة التاسعة: تصميم وتجريب دائرة لمقارنة معطيات بطول 2bit، وذلك باستخدام البوابات الأساسية وفق ما هو مبين بالشكل أدناه. وكذلك تصميم وتجريب مقارن لمعطيات 8 bit باستخدام دارات متكاملة مثل الدارة 7485 والتحقق من جدول الحقيقة الذي يظهر حالات التساوي أو الأكبر أو الأصغر بين عددين كل منهما 8 bits.

التجربة العاشرة: تصميم دائرة منطقية من النوع التركيبي Combinational لتحقيق وظيفة معينة مثل الدارة التي تحدد الأعداد الأولية من جملة الأعداد من 0 وحتى 15 والتي تنتهي بعد الاختصار إلى تحقيق تابع منطقي من الشكل $y = a'b'c + a'bd + bc'd + b'cd$. أو التحقق من صحة بعض التصاميم مثل تركيبية دائرة الجامع/ الطارح المتوازي.

التجربة الحادية عشر: تطبيق على القلابات بمختلف أنواعها، كبناء واختبار قلاب نوع R-S أو JK بوجود ساعة وبتوصيلات مختلفة للقلابات مثل توصيلة القلاب D مثلاً، كما في الشكل أدناه، والتحقق من جدول الحالات.



التجربة الثانية عشر: تطبيق على أساسيات مسجلات الإزاحة وتركيب سجلات بأنواع مختلفة (SISO, SIPO, PISO, PIPO) بطول 4 bits مثلاً،

التجربة الثالثة عشر: تطبيق على العدادات بتجريب استخدام نفس الدارة المتكاملة مثل الدارة 7493 مرة كعداد ثنائي ومرة أخرى كعداد عشري وفق ما هو مبين بالشكل



وكذلك تصميم وتجريب دارة عداد لا متزامن بعرض 4 bits مثلاً للعد بالشكل القياسي أو بال mod10 مثلاً، وإظهار آلية التقسيم باستخدام العدادات.

التجربة الرابعة عشر: تطبيق آخر على العدادات بتجريب عدادات متزامنة بطول 3 bit مثلاً وعدادات غير متزامنة بطول محدد تعدّ بشكل تصاعدي/تنازلي.

التجربة الخامسة عشر: تصميم وتجريب دارة منطقية تركيبية للمراقبة والتحكم من خلال تصميم الدارة المناسبة لمراقبة حالة سيارة عادية والتحكم بإقلاعها ضمن شروط محددة.

نحاول فيما يلي اظهار تحليل شامل لمثل هذه التجربة وفق التالي: يمكننا تصميم بعض الدارات المنطقية التركيبية لوظيفة المراقبة والتحكم مثلاً بشكل مماثل للدارة التي تم تصميمها للاشارة إلى الأعداد الأولية لكن بدرجة أكثر تعقيداً. تتضمن المسألة المعطاة بعض التفاصيل التي نشير إليها فيما يلي بالتحليل وصولاً إلى مخطط التوصيل والتجريب. نعلم أن السيارة العادية تحتاج للعمل بشكل نظامي إلى 4 L زيت للمحرك و 8 L ماء للتبريد وأن تكون البطارية بوضع جيد (مشحونة). نرغب من مجمل الباراميترات المتعلقة بالسيارة مراقبة هذه الباراميترات الثلاثة فقط وهي: البطارية B والزيت O والماء W، بحيث إن الخل في أي من هذه الباراميترات الثلاثة يحيل دون اقلاع السيارة، أي أن الخرج Start يكون صفر بحال وجود خلل (للحيلولة دون اقلاع المحرك) ويصبح بحالة واحد بعد تدارك الخلل (المحرك جاهز للاقلاع). بالتالي جدول الحقيقة المقابل لمثل هذه المسألة وفق الشروط التي نضعها يكون وفق ما هو مبين بالجدول التالي (تم مبدئياً وضع قيم وسطية بالجدول ويمكن تغيير المعطيات والشروط وفق ما يطلب). الشرط الأساسي بعد مراقبة حالة السيارة أن لايقطع المحرك في حال وجود نقص حاد في أي من المتحولين O و B، ويمكن السماح للمحرك بالاقلاع في حال وجود نقص بسيط بأي من المتحولين المذكورين. يمثل الجدول 1-15 تابع بثلاث متحولات من الشكل S(B, O, W) لكن ليست كلها متحولات منطقية، لأنه يمثل هذه المسائل يلاحظ أن بعض المتحولات قد يجوز لها قيمتين أو أكثر.

الجدول 1-15: ترميز تابع بمتحولات غير منطقية

Input			Output
B	O	W	S
0	d	d	0
1	أقل تماماً من 2L	أقل تماماً من 4L	0
1	أقل تماماً من 2L	أكثر تماماً من 4L	0
1	أكثر تماماً من 2L	أقل تماماً من 4L	0
1	أكثر تماماً من 2L	أكثر تماماً من 4L	1

فمثلاً وضع البطارية B يمكن أن يكون فقط 0 للإشارة إلى إن البطارية غير مشحونة وغير كافية لاقلاع المحرك، ويمكن أن تكون 1 للإشارة إلى أن وضع البطارية جيد (أي أن B متحول منطقي لأنه يأخذ قيمتين فقط). أما بالنسبة للزيت O، فيمكننا عن طريق حساسات جعله يأخذ ثلاث قيم مختلفة حسب كمية الزيت الموجودة بالمحرك وتقسمها وفق ما هو مبين بالجدول التالي:

O	O2	O1
0 - وتعني وجود 2L وما دون	0	0
1 - وتعني وجود حتى 3L	0	1
2 - وتعني وجود حتى 4L	1	0

يرمز المتحول O بحسب قيمته إلى أكثر من متحول منطقي. في هذه المسألة نجعل O مكون من متحولين منطقيين $O = O_2O_1$ لأن أعلى قيمة ممكنة لـ O هي 2 ولترميز 2 يكفي متحولين منطقيين، علماً أنه بمتحولين يمكن ترميز أربع قيم عشرية، بالتالي حالة $O_2O_1 = 11$ الممكنة وغير الواردة في الجدول والتي تقابل عشراً القيمة $O = 3$ يمكن في حال ورودها التعامل معها كحالة عدم تعين $d = \text{Don't care}$. وبنفس الطريقة يمكننا جعل المتحول W يأخذ أربع قيم مختلفة وفق الجدول التالي:

W	W2	W1
0 - وتعني وجود 2L وما دون	0	0
1 - وتعني وجود حتى 4L	0	1
2 - وتعني وجود حتى 6L	1	0
3 - وتعني وجود حتى 8L	1	1

وهكذا تتحول المسألة من تابع بثلاث متحولات وعدة أسطر لجدول التوصيف لعلاقة الخرج بالدخل إلى تابع بخمس متحولات منطقية $S(B, W_2, W_1, O_2, O_1)$ له $2^5 = 32$ تركيبة معرفة وفق ما هو مبين بالجدول أدناه. بكتابة التابع S وفق القيم المعرف بها في الجدول الجديد كما تم اختيارها يمكن اختصاره والوصول إلى الشبكة المنطقية المناسبة للتطبيق المشار إليه وتجريبها.

No.	B	W2	W1	O2	O1	S
0 to	0	0	0	0	0	0
15		...	...	...	...	
16	1	0	0	0	0	0
17	1	0	0	0	1	0
18	1	0	0	1	0	0
19	1	0	0	x = 1	x = 1	0
20	1	0	1	0	0	0
.	...	...	...	...	...	...
28	1	1	1	0	0	0
29	1	1	1	0	1	1
30	1	1	1	1	0	1
31	1	1	1	x = 1	x = 1	1

التجربة السادسة عشر: تصميم دائرة منطقية من النوع التتابعي Sequential لتحقيق

وظيفة معينة كنظام تحكم ضوئي وصوتي باستخدام الفلاب T

يمكننا تحليل مثل هذه المسألة بأسلوب مماثل لتحليل المسألة السابقة ويمكننا مباشرة رسم الدارة المنطقية المقابلة مباشرة بناء على العرض الوارد بالفقرة 3-3-9 كما في الشكل التالي:



نختم هذا الفصل بعرض مشروع تطبيقي يظهر آلية تنفيذ Implementation مخطط منطقي ورسم الدارة المطبوعة المقابلة له، حيث يمكن تنفيذ أي مشروع آخر بأسلوب مماثل.

مشروع تطبيقي

يطلب انتاج دارة منطقية تعمل على تحويل أي رقم بالرماز BCD مثلاً إلى ما يقابله بالرماز 84-2-1 مبنياً آلية التصميم من البداية ولغاية رسم الدارة المطبوعة المقابلة للمخطط المنطقي بشكله النهائي، بحيث تكون الدارة قابلة للإنتاج باستخدام العناصر المطلوبة. التصميم: بناء على ما هو مطلوب، يتم في المرحلة الأولى ترميز الدخل المعطى بالرماز BCD بالمتحولات B_0, B_1, B_2, B_3 والخرج المقابل للرماز 84-2-1 بالمتحولات Y_0, Y_1, Y_2, Y_3 لبناء جدول الحقيقة المناسب لتوصيف المطلوب وفق الجدول أدناه.

المتحولات	المتحولات	المتحولات	المتحولات	المتحولات	المتحولات
B_3	B_2	B_1	B_0	Y_3	Y_2

تمثل توابع الخرج الأربعة: Y_0, Y_1, Y_2, Y_3 بمخطط كارنوف بدلالة متحولات الدخل الأربعة B_0, B_1, B_2, B_3 ، وفق ما هو مبين في الشكل 12-15 بمعدل مخطط لكل تابع. أي أن كل تابع خرج من Y_0 حتى Y_3 سيكون من الشكل $Y = f(B_0, B_1, B_2, B_3)$. يوجد من أجل أربعة متحولات على الدخل 16 تركيبة، يستخدم منها فقط عشرة تراكيب لأنه بأربعة مداخل لا يمكن بالرماز BCD تمثيل إلا الأرقام من 0 حتى 9. بالتالي تستخدم حالات عدم الاكتراث d المقابلة للأعداد العشرية 10 حتى 15. وبرسم الشكل المختصر للتوابع الأربعة بمخطط واحد يكون الشكل النهائي للمخطط وفق ما هو مبين بالشكل 12-15

B_1B_0					
B_3B_2		00	01	11	10
	00	0	1	1	0
	01	0	1	1	0
	11	d	d	d	d
	10	0	1	d	d

B_1B_0					
B_3B_2		00	01	11	10
	00	0	1	0	1
	01	0	1	0	1
	11	d	d	d	d
	10	0	1	d	d

B_1B_0					
B_3B_2		00	01	11	10
	00	0	1	1	1
	01	1	0	0	0
	11	d	d	d	d
	10	0	1	d	d

B_1B_0					
B_3B_2		00	01	11	10
	00	0	0	0	0
	01	0	1	1	1
	11	d	d	d	d
	10	1	1	d	d



الشكل 12-15: المخطط المنطقي لإدارة التحويل بين الرمزين BCD و 8 4-2-1

يلاحظ من المخطط المنطقي المقابل أن المشروع يحتاج إلى ثلاث بوابات NOT وست بوابات AND كل منها بمدخلين و AND واحدة بثلاثة مدخل وبوابتي OR كل منها بثلاثة مدخل وبوابة OR واحدة بمدخلين. بانتقاء الدارات المتكاملة المناسبة، نجد مثلاً أنه يمكن تأمين البوابات NOT من الدارة المتكاملة ذات الرقم 7404 التي تحتوي على ست بوابات (يلزم منها ثلاث ويبقى ثلاث لأي استخدام لاحق). أما البوابات AND فيمكن تأمين أربع بوابات كل منها بمدخلين من الدارة المتكاملة ذات الرقم 7408. ولتأمين البوابتين المتبقيتين هناك إمكائتان: إما أن يستخدم لهما دارة 7408 إضافية، أو أن يتم تأمينهما من أي دارة أخرى مماثلة ومستخدمة في التصميم. ونظراً للحاجة إلى بوابة AND بثلاثة مدخل، تستخدم الدارة 7415 التي تضم ثلاث بوابات AND كل منها بثلاثة مدخل. يستخدم منها بوابة بثلاثة مدخل وبوابتين كل منهما بمدخلين بعد قصر المدخل الثالث مع أي من المدخلين الآخرين. يظهر الشكل أعلاه العناصر المستخدمة حتى بما في ذلك المقاومات من رتبة 220Ω . وبالنظر لعدم توفر دارة متكاملة تضم بوابات OR بثلاثة مدخل، تم البحث عن البديل وكان باستخدام الدارة 7427 التي تمثل ثلاث بوابات NOR كل منها بثلاثة مدخل، حيث يمكن استخدامها لتشكيل بوابتي OR بعد وصل NOT من الدارة 7404 على خرج كل منهما. يمكن استخدام مفتاح 4 bit switch لإدخال الرقم العشري وثنائيات مرسلة للضوء (LED) (Light Emitting Diode) لقراءة الناتج. يمكن باستخدام لوح مطلي بالنحاس رسم الدارة المطبوعة المقابلة للمخطط أعلاه، وفق الشكل 13-15 وتركيب العناصر والتجريب.

أ- الوجه السفلي للبطاقة Bottom

ب- الوجه العلوي للبطاقة Top

الشكل 13-15: الدارة المطبوعة المقابلة للمخطط المنطقي بالشكل 12-15

المراجع

- [1] التصميم الرقمي M. Morris Mano، المركز العربي للتعليم والترجمة والتأليف والنشر، دمشق، 2006.
- [2] معجم مصطلحات المعلوماتية من منشورات الجمعية العلمية السورية للمعلوماتية الطبعة الأولى، 2000.
- [3] عدة مواقع من الأنترنت يطول عرضها، ولكن يحدد معظمها بالبحث تحت اسم المفردة أو الفقرة نفسها الواردة في الكتاب (مثال ذلك المرجع [4])
- [4] Integrated Circuits: From Wikipedia, the free encyclopedia
- [5] Thomas L. Floyd, Digital Fundamental, 10th Edition, Published by Pearson International Edition, 2009.
- [6] A.P. Godse, Digital logic Design and Application, Publications Pune, 2008.
- [7] Paul lunde, The book of Codes, University of California press, 2009.
- [8] M. Morris Mano, Digital Design, Third Edition, Published by Prentice Hall Inc., 2002.
- [9] Jan, M. Rabaey, Anantha Chandrahassan, and Brivoje Nikolic, Digital Integrated Circuits – A Design Perspective, Published by Prentice Hall Electronics and VLSI Series, 2002.
- [10] Brown Stephan, Vranesic Zvonko, Fundamentals of Digital Logic with VHDL Design, 2nd Ed., Published by LAVOISIER Librarie, 2004.
- [11] Sharma Ashok K., Dinardo N. John, Advanced-Semiconductor- Memories-Architectures-Applications, Jon Wiley & sons, Inc. USA, 2003.
- [12] John F. Wakerly, Digital Design Principles and Practices, 4th Ed., Prentice Hall, USA, 2005.
- [13] Castinera Moreira Jurge, Farrell Patrick Guy, Essentials of Error-Control Coding, Lavoisier Librarie, 2006.

- [14] Zoran Salcic, Asim Smailagic, Digital System Design and Prototyping Using Field Programmable Logic and HDL, 2nd Ed., Kluwer Academic Publishers, 2000.
- [15] R. D. Dowsing and Ian Marshall, Computers: From Logic to Architecture, 2nd ed., Published by McGraw Hill, 2000.
- [16] Ramachadan S., Digital VLSI System Design: A design manual for Implementation of projects on FPGAs & ASICs using verilog, Published by Lavoisier Librarie, 2007.
- [17] David Harris, Digital Design and Computer Architecture, Morgan Kaufmann, 2007.
- [18] Frank Vahid, Digital Design, University of California, Wiley & sons, Inc. USA, 2006.
- [19] James O. Hamblen, Tyson S. Hall, Michael D.Furman, Rapid Prototyping Digital System, 2nd edition, Springer,2005.
- [20] Alexander Miczo: Digital Logic Testing and Simulation, 2nd edition, WILEY & sons, Inc. USA, 2003.
- [21] Michael D. Ciletti, Advanced Digital Design with Verilog HDL, 1th edition, Pearson Education, 2002.
- [22] Alan Clements, Principles of Computer Hardware, Oxford University Press, 2006.
- [23] Set of Web Sites About "Logic Design" - One of them is: [Http: en. Wikibooks. Org/ wiki/ Digital_Circuits.](http://en.wikibooks.org/wiki/Digital_Circuits)
- [24] Marcovitz, Alan B., Introduction to Logic and Computer Design, McGraw-Hill Companies, 2008
- [25] Jr. Charles H. ; Kinney, Larry L., Fundamentals of Logic Design, 7th ed., Cengage Learning, 2013

الملاحق

Page	الملحق الأول: قائمة بالجداول الواردة في الكتاب
P37	الجدول 1-2: ترميز إشارتي + و - في كل من أنظمة العد الأربعة الأساسية
P60	الجدول 1-3: أول عشرين قيمة عشرية بالرماز BCD
P62	الجدول 2-3: الرمازان الموزنان: 1 2 4 2 و 1 2 4 8
P64	الجدول 3-3: الرماز Excess-3
P65	الجدول 3-4: رماز كراي
P68	الجدول 3-5: رماز ASCII
P74	الجدول 3-6: آلية تقسيم مولد رماز 4 bit للحصول على D' وفق CRC
P78	الجدول 3-7: عدد الأخطاء التي يمكن كشفها وإصلاحها باستخدام هامينغ
P79	الجدول 3-8: توزع بتات المراقبة وفق كود هامينغ على كلمة بطول 4 bits
P89	الجدول 4-1: جدول الحقيقة بتراكيبه المختلفة لعبارة منطقية
P91	الجدول 4-2: التوابع المنطقية الأساسية
P92	الجدول 4-3: جدول الحقيقة لتابع منطقي بثلاث متحولات
P97	الجدول 4-4: الحدود الصغرى والعظمى الممكنة لتابع منطقي بثلاث متحولات
P100	الجدول 4-5: جدول الحقيقة للتوابع الستة عشر التي يمكن تعريفها بمتحولين
P124	الجدول 5-1: جدول الحقيقة لدارة منطقية بوجود حالات عدم تعيين
P194	الجدول 7-1: قيم البارامترات الأساسية لمختلف سلاسل العائلة TTL
P203	الجدول 7-2: بعض دارات TTL التي تمثل بوابات منطقية
P207	الجدول 8-1: الأعداد الأولية مابين 0 وال 15
P211	الجدول 8-2: جدول الحقيقة لفاحص الندية (الزوجية) لرسالة مكونة من 4 bits
P213	الجدول 8-3: جدول الحقيقة لإظهار الأرقام العشرية على سبع مقاطع مضيئة
P217	الجدول 8-4: جدول الحقيقة لدارة نصف جامع
P218	الجدول 8-5: جدول الحقيقة للجامع التام

- P221 الجدول 6-8: جدول الحقيقة لنصف الطارح
- P222 الجدول 7-8: جدول الحقيقة للطارح التام
- P239 الجدول 8-8: المتمم التساعي للأعداد العشرية
- P242 الجدول 9-8: جدول الحقيقة لمفكك ترميز 2×4
- P243 الجدول 10-8: جدول الوظائف (العمل) لمفكك ترميز 2×4
- P244 الجدول 11-8: جدول الحقيقة للمرمز العشري
- الجدول 12-8: بعض الدارات المتكاملة من العائلة TTL التي تمثل مكونات
- P249 رقمية أساسية بتركيبة الدارات المنطقية التركيبية.
- P250 الجدول 13-8: جدول الحقيقة لمرمز ذا أفضلية بأربعة مداخل
- P253 الجدول 14-8: جدول الحقيقة (الوظائف) لمضمم بالتركيبة 4×1 مع رمزه
- P256 الجدول 15-8: جدول بالتراكيب الموافقة لعمل المقارن
- P258 الجدول 16-8: جدول الوظائف لمقارنة عددين كل منهما بطول 12 bits
- P274 الجدول 1-9: جدول الوظائف للقلاب RS
- P276 الجدول 2-9: جدول التحريض للقلاب RS
- P285 الجدول 3-9: جدول الوظائف للقلاب RS بمدخل خارجية مباشرة
- P289 الجدول 4-9: جدول الوظائف للقلاب D
- P289 الجدول 5-9: جدول التحريض للقلاب D
- P293 الجدول 6-9: الجدول الوظيفي للقلاب T
- P294 الجدول 7-9: جدول التحريض للقلاب T
- P297 الجدول 8-9: الجدول الوظيفي للقلاب JK
- P298 الجدول 9-9: جدول التحريض للقلاب JK
- P300 الجدول 10-9: أرقام بعض الدارات المتكاملة بالتقانة TTL التي تمثل قلابات
- الجدول 1-10: آلية إدخال المعطيات إلى مسجل مكون من القلابات RS من
- P309 نوع دخل تسلسلي - خرج تسلسلي
- P323 الجدول 2-10: أرقام بعض الدارات المتكاملة التي تمثل أنواعا من السجلات

- الجدول 10-3: جدول الوظائف للسجل 74194 P316
- الجدول 10-4: جدول الحالة للجامع التسلسلي بوجود قلابات من نوع JK P322
- الجدول 11-1: جدول الحالة لعدد خمس قيم من أصل ثماني P336
- الجدول 11-2: بعض الدارات المتكاملة التي تمثل بعض العدادات والمهتزازات P340
- الجدول 11-3: جدول الوظائف للدائرة المتكاملة 7493A P341
- الجدول 11-4: جدول الحالة لعدد متزامن يعد ثماني قيم P344
- الجدول 11-5: جدول الحالات لعدد عشري مع القيم المفترضة لمداخل القلابات P346
- الجدول 11-6: جدول الحالة لعدد من ثلاث خانات (يعد 8 قيم) P349
- الجدول 11-7: جدول الوظائف لعدد يدخل متواز P352
- الجدول 11-8: جدول الحالات لعدد حلقي P355
- الجدول 11-9: جدول الحالات لعدد جونسون مكون من أربعة قلابات P356
- الجدول 11-10: جدول الحالات لدائرة توليد أعداد عشوائية P358
- الجدول 12-1: الشكل العام لجدول الحالة P369
- الجدول 12-2: جدول الحالة المقابل لمخطط الحالة بالشكل 12-5 P375
- الجدول 12-3: الشكل المعدل لجدول الحالة بالجدول 12-2 بدون الحالة q5 P375
- الجدول 12-4: الشكل المعدل لجدول الحالة بالجدول 12-3 بدون الحالة q4 P376
- الجدول 12-5: الشكل المعدل لجدول الحالة بالجدول 12-4 بدون الحالة q3 P376
- الجدول 12-6: جدول يبين استخراج جدول الحالة من مخطط الحالة P379
- الجدول 12-7: جدول حالة بوجود الدخل x والخرج y P380
- الجدول 12-8: جدول حالة مرمر ثنائياً P381
- الجدول 12-9: جدول حالة مرمر بوجود القلاب JK P382
- الجدول 12-10: جداول الانتقال للدائرة المبينة في الشكل 12-7 P387
- الجدول 12-11: جدول الانسياب المقابل لجدول الانتقال P388
- الجدول 12-12: جدول الانتقال والانسياب لدائرة تتابعية غير متزامنة P388
- الجدول 12-13: جدول انسياب يتضمن قيم خرج دائرة تتابعية غير متزامنة P389

- الجدول 12-14: الحصول على جدول الانتقال لدارة تتابعية من جدول الانسياب P390
- الجدول 12-15: الحصول على جدول الخرج لدارة تتابعية من جدول الانسياب P390
- الجدول 12-16: جدول حالة يطلب اختصاره P391
- الجدول 12-17: جدول التضمين P392
- الجدول 12-18: جدول حالة مختزل من سبع إلى أربع حالات P394
- الجدول 12-19: جدول انتقال لدارة تتابعية بثلاث حالات استقرار وسباقات حرجة P396
- الجدول 12-20: جدول انتقال لدارة غير مستقرة P398
- الجدول 13-1: مؤثرات لغة Verilog HDL P418
- الجدول 13-2: جداول الحقيقة للبوابات الأولية المعرفة سلفاً P422
- الجدول 14-1: جدول لبرمجة مصفوفة PLA P450
- الجدول 14-2: جدول لبرمجة PAL P453
- الجدول 15-1: ترميز تابع بمنحولات غير منطقية P467

Page	الملحق الثاني: قائمة بالأشكال الواردة في الكتاب
P16	الشكل 1-1: التركيبة العامة للحاسوب الرقمي
P18	الشكل 1-2: أنواع الإشارات الرقمية ومجالها
P69	الشكل 1-3: أهم تقنيات ضغط البيانات
P76	الشكل 2-3: التمثيل الهندسي للمتحولات المنطقية
P86	الشكل 1-4: الرمز الكيربائي لكل من قوانين بول لعبارة منطقية بمتحول واحد
P88	الشكل 2-4: طريقة المفاتيح الكيربائية بإثبات صحة قانون التوزيع
P98	الشكل 3-4: البوابات المنطقية الأساسية من المستويين صفر وواحد
P99	الشكل 4-4: شكل إشارات الدخل والخرج لبعض البوابات المنطقية الأساسية
P102	الشكل 4-5: البوابات المنطقية المركبة (من المستويين الثاني والثالث)
P103	الشكل 4-6: تشكيل البوابات NOT و AND و OR من NAND و NOR
P104	الشكل 4-7: تشكيل بوابة NOR من بوابات NAND
P104	الشكل 4-8: التركيبة الداخلية للبوابة XOR
P106	الشكل 4-9: أشكال مختلفة لدارات منطقية تقابل نفس التابع المنطقي
P106	الشكل 4-10: شبكة منطقية بالتركيبة AND-OR تقابل التابع المنطقي
P108	$F(a, b, c, d) = a + bc' + a'cd' + a'b'cd$
P109	الشكل 4-11: الرموز البديلة لكل من بوابتي NAND و NOR
P109	الشكل 4-12: الشبكة المنطقية للتابع $F(a,b,c,d) = a'd + ab'c + a'bcd'$
P110	بالتركيبة NAND فقط
P110	الشكل 4-13: الشبكات المنطقية الثلاث الممكنة لتابع منطقي معطى بالشكل
P111	الجامع ($F(a, b, c, d) = ac + bd$)
P111	الشكل 4-14: الشبكات المنطقية الثلاث الممكنة لتابع منطقي معطى بالشكل
P112	الضارب ($F(a, b, c, d) = (a(c + d)(a + b + c))'$)
P113	الشكل 4-15: شبكة منطقية بثلاثة مراحل
P113	الشكل 4-16: المخطط الزمني للشبكة التي تقابل التابع $y = a'bc$

- الشكل 1-5: مخطط فين لمتحولين P117
- الشكل 2-5: مخطط كارنوف لتابع منطقي بمتحولين P118
- الشكل 3-5: الترميز الثنائي لمتحولين في مخطط كارنوف P119
- الشكل 4-5: الترميز الثنائي لثلاث متحولات بمخطط كارنوف وفق رماز كراي P119
- الشكل 5-5: مخطط كارنوف لتابع منطقي بأربع متحولات P120
- الشكل 6-5: مخطط كارنوف لتابع منطقي بخمس متحولات P121
- الشكل 7-5: مخطط كارنوف لتابع منطقي معطى بالصيغة المبسطة P122
- الشكل 8-5: مخطط كارنوف لتابع منطقي بأربع متحولات P123
- الشكل 9-5: شكلان مختلفان ومتكافئان لتشكيل حلقات لنفس التابع بكارنوف P131
- الشكل 10-5: تمثيل تابع منطقي بأربع متحولات بمخطط كارنوف مع الحلقات P133
- الشكل 11-5: تمثيل تابع منطقي بخمس متحولات بمخطط كارنوف مع الحلقات P136
- الشكل 12-5: تمثيل تابع منطقي بستة متحولات بمخطط كارنوف مع الحلقات P137
- الشكل 1-6: رموز منطقية لبعض العناصر المنطقية P143
- الشكل 2-6: شروط ربط عنصرين منطقيين P144
- الشكل 3-6: مميزة العبور للعنصر المنطقي NOT P145
- الشكل 4-6: قدرة المخرج على الإمداد (Fan-out) P145
- الشكل 5-6: شكل النبضة وزمن تأخير انتشار الإشارة بالبوابة NOT P148
- الشكل 6-6: بوابة OR بمدخلين وفق المنطق DL P150
- الشكل 7-6: بوابة AND بمدخلين بالمنطق DL P151
- الشكل 8-6: تركيبية العنصر NOT وفق التقنية DTL P152
- الشكل 9-6: تركيبية بوابة NAND بمدخلين وفق المنطق DTL P152
- الشكل 10-6: تركيبية NAND بثلاثة مداخل وفق المنطق DTL المعدل P154
- الشكل 11-6: تركيبية NOR وفق المنطق DTL P155
- الشكل 12-6: تركيبية البوابة NOT وفق المنطق DTL P155
- الشكل 13-6: تركيبية البوابة AND وفق المنطق DTL P156

- الشكل 6-14: تركيبة البوابة OR وفق المنطق DTL P156
- الشكل 6-15: تركيبة البوابة OR بالمنطق RTL P157
- الشكل 6-16: التركيبة العامة لكل من NAND و NOR بالمنطق RTL P158
- الشكل 6-17: تركيبة البوابة NOR وفق المنطق RCTL P159
- الشكل 6-18: تركيبة البوابتين OR و NOR بالمنطق ECL P159
- الشكل 6-19: الرمز الكهربائي للترانزيستور متعدد البواعث P161
- الشكل 6-20: التركيبة القياسية للمنطق TTL (بوابة NAND بثلاثة مداخل) P162
- الشكل 6-21: تركيبة كل من البوابات المنطقية الأساسية وفق TTL العادي P163
- الشكل 6-22: وصلة غير مسموح بها لخرج بوابات بالمنطق TTL العادي P165
- الشكل 6-23: تركيبة بوابة NAND بثلاثة مداخل بالمنطق TTL بمجمع مفتوح P166
- الشكل 6-24: الرمز المنطقي لعناصر TTL بمجمع مفتوح P167
- الشكل 6-25: تمثيل التركيبة NOR – AND باستخدام عناصر بمجمع مفتوح P167
- الشكل 6-26: الدارة الالكترونية لتجزير التابع المنطقي $F = (a.b.c + xyz)'$ P168
- الشكل 6-27: تطبيق مباشر على استخدام منطق TTL بمجمع مفتوح P168
- الشكل 6-28: رموز العنصر التوسيعي Expander P169
- الشكل 6-29: التركيبة الداخلية لدارة موسع Expander P170
- الشكل 6-30: رمز وتركيبه عنصر منطقي مشكل بالمنطق TTL وبثلاث حالات P171
- الشكل 6-31: الرموز الكهربائية الأربعة لترانزيستور MOS P174
- الشكل 6-32: بوابة NOT بنوعين من الأحمال بالتقانة PMOS P175
- الشكل 6-33: بوابات NOR و NAND بالتقانة PMOS P176
- الشكل 6-34: التابع المنطقي $y=(A.B+C)'$ (Expander) بالتقانة PMOS P177
- الشكل 6-35: بوابات NOT و NOR و NAND بالتقانة NMOS P178
- الشكل 6-36: تركيبة العنصر NOT وفق التقانة CMOS P179
- الشكل 6-37: تركيبة البوابتين NAND, NOR بالتقانة CMOS P180
- الشكل 6-38: الموازنة بين تقانتي أحادي وثنائي القطبية P181

- P187 الشكل 7-1: تصنيفات الدارات المتكاملة
- P188 الشكل 7-2: المسقط الرأسي لإدارة متكاملة بالتغليف DIL-14
- P190 الشكل 7-3: التوزيع الداخلي للبوابات المنطقية ضمن الدارة المتكاملة 7400
- P192 الشكل 7-4: زمن تأخير الانتشار بالديود العادي
- P193 الشكل 7-5: التيارات المنصبة والنابعة بدارة متكاملة مصنعة بالمنطق TTL
- P197 الشكل 7-6: العناصر المنطقية الداخلة في تركيب الدارة المتكاملة 74136
- P197 الشكل 7-7: شكل لدارات متكاملة من البوابة إلى التغليف من رتبة SSI
- P198 الشكل 7-8: مظهر وصل مأخذ الدارة المتكاملة بالـ Chip الداخلي
- P200 الشكل 7-9: أشكال مختلفة لتغليف الدارات المتكاملة
- P201 الشكل 7-10: الدارة المتكاملة 555
- P202 الشكل 7-11: توصيل الدارة 555 بوصفها مولداً للن نبضات الساعية
- P207 الشكل 8-1: الدارة المنطقية للتابع الذي يحدد الأعداد الأولية من 0 حتى 15.
- P210 الشكل 8-2: المخطط المنطقي لمولد وفاحص الندية
- P215 الشكل 8-3: دارة إظهار الأرقام العشرية على عنصر إظهار من سبعة مقاطع
- P217 الشكل 8-4: الرمز المنطقي لنصف الجامع
- P217 الشكل 8-5: المخطط المنطقي لنصف الجامع
- P218 الشكل 8-6: الرمز المنطقي للجامع التام
- P219 الشكل 8-7: المخطط المنطقي للجامع الكامل
- P219 الشكل 8-8: تشكيل دارة جامع كامل باستخدام دارتي نصف جامع
- P220 الشكل 8-9: دارة جامع ثنائي متواز بطول 4 bits
- P221 الشكل 8-10: الرمز المنطقي لنصف الطارح
- P222 الشكل 8-11: المخطط المنطقي لنصف الطارح HS
- P222 الشكل 8-12: الرمز المنطقي للطارح التام
- P223 الشكل 8-13: المخطط الصندوقي للطارح المتوازي بطول 4 bits
- P224 الشكل 8-14: دارة جامع طارح على التوازي بطول 4 bits

- الشكل 8-15: المخطط المنطقي لمولد المنقول المستبق P226
- الشكل 8-16: الدارة الكاملة لجامع / طارح ثنائي بطول 3 bits P229
- الشكل 8-17: توضع إشارات الدارة 7483 (الدخل والخرج والتحكم والتغذية) P230
- الشكل 8-18: المخطط العام لجمع وطرح عددين بطول 4 bits على التوازي P231
- الشكل 8-19: منطق لتشكيل المتمم للواحد بوجود إشارة تحكم خارجية Ci. P232
- الشكل 8-20: دارة جمع / طرح لعددين 4 bits (5 ± 14) باستخدام 7483 P233
- الشكل 8-21: الشكل العام للضارب الثنائي P234
- الشكل 8-22: دارة لضرب عددين ثنائيين بطول 4 bits للأول و 3 bits للثاني P236
- الشكل 8-23: جزء من دارة جمع عددين عشريين ممثلين بالرماز BCD P237
- الشكل 8-24: آلية تشكيل المتمم التساعي لرقم عشري P240
- الشكل 8-25: الرمز المنطقي لمفك الترميز P241
- الشكل 8-26: مفك ترميز 2 إلى 4 خطوط (باستخدام بوابات AND) P242
- الشكل 8-27: 2x4 Decoder مع مدخل تأهيل (باستخدام بوابات NAND) P242
- الشكل 8-28: الرمز المنطقي والمخطط المنطقي للرمز العشري P245
- الشكل 8-29: تشكيل دارة 4x16 Decoder من دارات 2x4 Decoder P246
- الشكل 8-30: دارة لتنفيذ تابع منطقي باستخدام مفك ترميز وبوابة OR P247
- الشكل 8-31: تشكيل مسرى (ممر Bus) ذي ثلاث حالات باستخدام مفك ترميز P248
- الشكل 8-32: توضع إشارات الدارة المتكاملة 74138 P249
- الشكل 8-33: دارة مرمر أفضلية بأربعة مداخل P251
- الشكل 8-34: الرمز المنطقي للمضمم P252
- الشكل 8-35: المخطط المنطقي لمضمم 4 x 1 P253
- الشكل 8-36: تشكيل مضمم 16x1 MUX من مضممين 8x1 MUX P254
- الشكل 8-37: الرمز المنطقي لمقارن بين عددين بطول 3 bits لكل منهما. P256
- الشكل 8-38: المخطط المنطقي لمقارن بين عددين بطول 3 bits لكل منهما P257
- الشكل 8-39: دارة لمقارنة عددين بطول 12 bits باستخدام الدارة 7485 P258

- الشكل 8-40: المخطط العام لدارة المحول 74184 P259
- الشكل 8-41: المخطط العام لاستخدام المحول 74185 بتحويل الأعداد P260
- الشكل 8-42: مخطط دارة إظهار الأرقام العشرية باستخدام السبعة مقاطع P262
- الشكل 9-1: الأشكال الممكنة للتعبير عن اندارة التتابعية P267
- الشكل 9-2: دارة أحادي وعديم الاستقرار P269
- الشكل 9-3: التوصيلة الأساسية لدارة ثنائي استقرار P270
- الشكل 9-4: المخطط المنطقي للقلاب RS P273
- الشكل 9-5: المخطط المنطقي لقلاب RS المتزامن P277
- الشكل 9-6: الرموز المعتمدة للاستجابة للساعة (المستوى والحافة) P278
- الشكل 9-7: أزمنة التهيئة والوقوف P280
- الشكل 9-8: المخطط الزمني للقلاب RS P281
- الشكل 9-9: المخطط المنطقي لقلاب RS نوع قائد - مقاد P283
- الشكل 9-10: المخطط الصندوقي للقلاب المتزامن RS P283
- الشكل 9-11: المخطط المنطقي للقلاب RS غير المتزامن P284
- الشكل 9-12: قلاب RS بمدخل خارجية مباشرة P285
- الشكل 9-13: المخطط المنطقي لقلاب RS بمدخل خارجية مباشرة P286
- الشكل 9-14: دارة لإلغاء أثر اهتزاز مفتاح ميكانيكي باستخدام القلاب RS P287
- الشكل 9-15: المخطط المنطقي للقلاب D P288
- الشكل 9-16: قلاب D يقدح بالحافة P290
- الشكل 9-17: دارة مراقبة بسيطة باستخدام القلاب D P292
- الشكل 9-18: الرمز المنطقي للقلاب T P293
- الشكل 9-19: نظام تحكم ضوئي وصوتي باستخدام القلاب T P295
- الشكل 9-20: المخطط المنطقي للقلاب J-K P297
- الشكل 9-21: شكل توزيع مأخذ القلابات J-K في الدارة 7476 P298
- الشكل 9-22: تشكيل القلاب T من القلاب D P298

- الشكل 9-23: تشكيل القلابين D و T من القلاب J-K P299
- الشكل 9-24: تشكيل القلاب J-K من القلاب D P299
- الشكل 10-1: سجل بطول 4 bits P303
- الشكل 10-2: الرمز العام للسجل P304
- الشكل 10-3: سجل إزاحة نحو اليمين بطول 4 bits. P305
- الشكل 10-4: سجل إزاحة نحو اليسار بطول 4 bits P306
- الشكل 10-5: سجل دوران (حلقى) نحو اليمين بطول 4 bits P307
- الشكل 10-6: منطق للتحكم باختيار واحد من مدخلين P307
- الشكل 10-7: سجل إزاحة مع منطق تحكم للتبديل بين الوظائف P308
- الشكل 10-8: المخطط المنطقي لسجل SISO بطول 4 bits P310
- الشكل 10-9: توزيع مأخذ الدارة المتكاملة 7491A ورمزها العام P310
- الشكل 10-10: المخطط المنطقي لسجل بطول 4 bits من نوع SIPO P311
- الشكل 10-11: المخطط المنطقي لسجل من نوع PIPO بطول 4 bits P312
- الشكل 10-12: التركيبة الداخلية لسجل من نوع PISO بطول 4 bits P313
- الشكل 10-13: الرمز المنطقي للدارة المتكاملة 74165 P315
- الشكل 10-14: سجل إزاحة بالاتجاهين بطول 4 bits. P314
- الشكل 10-15: التركيبة العامة لسجل يدخل على التوازي بطول 4 bits P315
- الشكل 10-16: الرمز المنطقي لسجل عام (تحديداً الدارة 74194) P316
- الشكل 10-17: التركيبة الداخلية لسجل متعدد الأغراض بطول 4 bits. P317
- الشكل 10-18: الدارة التركيبية لمزيج بطول 4 bits مع جدول الوظائف المقابل P318
- الشكل 10-19: التركيبة العامة للجامع التسلسلي P321
- الشكل 10-20: دارة الجامع التسلسلي P322
- الشكل 11-1: الرموز العامة للعداد P325
- الشكل 11-2: عداد ثنائي يعد قيمتين P326
- الشكل 11-3: دارة عداد لأربع قيم - غير متزامن P327

- P328 الشكل 4-11: المخطط الزمني لعداد غير متزامن يعّد أربع قيم
- P329 الشكل 5-11: دائرة عداد لأربع قيم - متزامن
- P329 الشكل 6-11: المخطط الزمني لعداد متزامن يعّد أربع قيم
- P330 الشكل 7-11: مخطط الحالة وجدول العبور لعداد بأربع حالات
- P331 الشكل 8-11: التركيبة الداخلية لعداد غير متزامن تصاعدي يعّد ثماني قيم
- P332 الشكل 9-11: المخطط الزمني لعداد غير متزامن تصاعدي (يعّد حتى ثمانية)
- P333 الشكل 10-11: التركيبة الداخلية لعداد غير متزامن تنازلي يعّد ثماني قيم
- P333 الشكل 11-11: المخطط الزمني لعداد غير متزامن تنازلي (يعّد حتى ثمانية)
- الشكل 12-11: التركيبة الداخلية لعداد غير متزامن يعّد ثماني قيم بشكل تصاعدي تنازلي معاً
- P334
- P337 الشكل 13-11: التركيبة الداخلية لعداد غير متزامن تنازلي يعّد 45 قيمة
- P339 الشكل 14-11: دائرة للحصول على اشارات بترددات مختلفة من 1 MHz
- P341 الشكل 15-11: توزيع مأخذ الدارة المتكاملة 493A7
- P342 الشكل 16-11: عداد عشري باستخدام الدارة 7493
- P345 الشكل 17-11: التركيبة الداخلية لعداد متزامن قياسي، يعّد ثماني قيم
- P347 الشكل 18-11: التركيبة الداخلية لعداد عشري متزامن
- P348 الشكل 19-11: مخطط كارنوف لحساب قيمة مدخل القلاب A (JA)
- P350 الشكل 20-11: المخطط العام لدائرة عداد تصاعدي تنازلي يعّد ثماني قيم
- P351 الشكل 21-11: التركيبة الداخلية لعداد، يعّد ثماني قيم ويدخل متوازٍ
- P354 الشكل 22-11: المخطط الصندوقي لقطر عدادين عشريين
- P355 الشكل 23-11: المخطط المنطقي للعداد الحلقي
- P357 الشكل 24-11: مولد للأعداد العشوائية الزائفة
- P359 الشكل 25-11: مولد بسيط للنبضات الساعية
- P359 الشكل 26-11: مولد للنبضات الساعية مع عنصر RC بثابت زمني كبير
- P360 الشكل 27-11: مضاعف التواتر

- الشكل 1-12: النموذج العام للدارة التتابعية P363
- الشكل 2-12: المداخل والمخارج والحالات الداخلية للدارة التتابعية المتزامنة P365
- الشكل 3-12: مثال لدارة تتابعية بقصد التحليل P368
- الشكل 4-12: مخطط حالة لدارة تتابعية بأربع حالات P370
- الشكل 5-12: مخطط الحالة لدارة تتابعية متزامنة P373
- الشكل 6-12: النموذج العام للدارة التتابعية غير المتزامنة P384
- الشكل 7-12: جدول انسياب بثلاثة أسطر P398
- الشكل 8-12: جدول انسياب بسطر إضافي لدارة خالية من السباقات P399
- الشكل 9-12: اختيار أسطر إضافية لجدول انسياب بأربعة أسطر P400
- الشكل 10-12: دارة تتابعية مستقرة مع جدول الانتقال P402
- الشكل 11-12: دارة تركيبية ذات مخاطر P405
- الشكل 12-12: دارة خالية من المخاطر P406
- الشكل 13-12: دارة تتابعية غير متزامنة ذات مخاطر P407
- الشكل 1-13: الأشكال الأربع الممكنة للبوابات ثلاثية الحالة P422
- الشكل 1-14: المنطق الداخلي لمصفوفة منطقية قابلة للبرمجة PLA P499
- مكوناً من ثلاثة مداخل وأربعة حدود جداء ومخرجين P499
- الشكل 2-14: مخطط الفواصم لمنطق PAL، كما تم تحديده في جدول البرمجة P453
- الشكل 3-14: دارة تتابعية تتضمن تجهيزات SPLD P455
- الشكل 4-14: مصفوفة مفاتيح قابلة للبرمجة P455
- الشكل 1-15: الدارة 7404 - ست بوابات NOT P458
- والدارة 7407 - ست بوابات Buffer P458
- الشكل 2-15: الدارة 7432 - أربع بوابات OR كل منها بمدخلين P459
- والدارة 7408 - أربع بوابات AND كل منها بمدخلين P459
- الشكل 3-15: الدارة 7411 - ثلاث بوابات AND كل منها بثلاثة مداخل P459
- والدارة 7402 - أربع بوابات NOR كل منها بمدخلين P459

- الشكل 4-15: الدارة 7427 - ثلاث بوابات NOR كل منها بثلاثة مداخل
- P460 والدارة 7410 - ثلاث بوابات NAND كل منها بثلاثة مداخل
- الشكل 5-15: الدارة 74139 - تمثل مفككي ترميز كل منهما 2x4 Decoder
- P460 والدارة 74154 - تمثل مفكك ترميز بالتركيبة 4x16 Decoder
- الشكل 6-15: الدارة 7446 - تمثل BCD to 7-Segment Decoder
- P461 والدارة 74147 - تمثل مرمر 10x4 Encoder
- الشكل 7-15: الدارة 74153 - تمثل معاقب 8x1 Multiplexer
- P461 والدارة 7485 - مقارنة بين عددين بطول 4 bits
- الشكل 8-15: الدارة 7486 (74136) - أربع بوابات XOR كل منها بمدخلين
- P462 والدارة 7486 (74136) - أربع بوابات XOR كل منها بمدخلين
- الشكل 9-15: الدارة 74184/185 - تمثل BCD-to-Binary Converter/ Binary-to-BCD Converter with OC
- P462 والدارة 74195 - تمثل 4-bits Shift Register PIPO
- الشكل 10-15: الدارة 7474 - تمثل قلابين من نوع D
- P463 والدارة 74390 - تمثل عدادين عشريين (Ripple Counter)
- الشكل 11-15: المخطط المنطقي للعبارتين المتساويتين $(X+Y)'$, $X'Y'$
- P464
- الشكل 12-15: المخطط المنطقي للتحويل بين الرمزين BCD و 4-2-1 و 8
- P469
- الشكل 13-15: الدارة المطبوعة المقابلة للمخطط المنطقي بالشكل 13-15
- P470

الملحق الثالث: جدول بأهم التعابير والمصطلحات الانكليزية

Absolute value	القيمة المطلقة
Absorption law	قانون الامتصاص
Access time	زمن الوصول أو النفاذ
Addition circuits	دارات الجمع
Adjacent squares	المربعات المتجاورة
Ambiguity	غموض أو التباس
American Standard Code for Information Interchange (ASCII)	الرماز الأمريكي المعياري لتبادل المعلومات
Analog information	معلومات تمثيلية
Anode	مصعد
Approximation	تقريب
Architectural	معماري
Arithmetic Logic Unit (ALU)	وحدة الحساب والمنطق
Array logic	صفيفة منطقية
Assignment	إسناد
Associative law	قانون التجميع
Astable circuit	دائرة عديمة الاستقرار
Asynchronous logic circuit	دائرة منطقية غير متزامنة
Average current	متوسط التيار
Behavioral modeling	نمذجة سلوكية
Bi-directional	ذو اتجاهين
Binary Adder-Subtractor	جامع - طارح ثنائي
Binary Coded Decimal (BCD)	العشري المرمز ثنائيا
Binary counter	عداد ثنائي
Binary division	القسمة الثنائية

Binary multiplication	الضرب الثنائي
Binary multiplier	الضارب الثنائي
Binary pattern	تشكيل من المواضع الثنائية
Binary Ripple Counter	عداد تموجي ثنائي
Binary subtraction	طرح ثنائي
Bipolar Junction Transistor (BJT)	ترانزستور ثنائي القطبية
Bi stable circuit	دائرة ثنائية الاستقرار
Borrow	إعارة أو استقراض
Breadboard	لوحة اختبار
Buffer gate	بوابة دارى
Canonical form	صيغة قانونية (للتابع المنطقي)
Carry-in	المنقول من المرتبة الأدنى
Carry-out	المنقول إلى المرتبة الأعلى
Carry generate	مولد المنقول
Carry look ahead	المنقول المستبق
Carry propagation	انتشار المنقول
Case sensitive	حساس للحالة ...
Cathod	مهبط
Character	محرف
Check bit	بت اختبار
Chip	رقاقة أو شريحة
Chip Select (CS)	انتقاء الشيب أو الدارة
Clock generator	مولد نبضات الساعة
Clock Pulse (CP)	نبضة الساعة
Clock response	استجابة الساعة
Clocked sequential circuit	دائرة تتابعيه محكومة بنبضات الساعة

Closure	إغلاق (متكرر)
Coded information	معلومات مرمزة
Combinational logic circuit	الدارة المنطقية التركيبية
Commutative law	قانون التبديل
Comparator	المقارن
Compatibility	التوافق أو المواءمة
Complement	المتمم
Complementary MOS (CMOS)	إتمامي (من متمم)
Components	المكونات
Computer Aided Design (CAD)	تصميم بمساعدة الحاسوب
Concatenation	الإلحاق أو الضم جنباً لجنب
Condition register	سجل الشرط (الحالة)
Configuration	تشكيلة
Contact's logic circuits	الدارات المنطقية التلامسية
Converters	المحولات
Counter cascading	قطر الغدادات
Critical race	سباق حرج
Current Mode Logic (CML)	نمط المنطق التياراتي
Current sink	مصرف التيار
Current source	منبع التيار
Cut-off	حالة القطع
Cyclical Redundancy Check (CRC)	الفحص الحلقي التكراري
Dataflow modeling	نمذجة جريان المعطيات
Data representation	تمثيل المعطيات
Data selector	ناخب المعطيات
Debounce circuit	دائرة حذف الاهتزاز

Decade counter	عداد عشري
Decay time	زمن الهبوط
Decimal Adder / Subtractor	الجامع / الطارح العشري
Decimal arithmetic	الحساب العشري
Decimal system	النظام العشري
Decoder	مفكك الترميز
Demultiplexer	مفروق
Depletion mode	نمط إفقاري
Description	وصف أو توصيف
Design procedure	إجرائية التصميم
Destination	الهدف
Difference	الفرق أو ناتج طرح عددين
Digital logic fundamentals	أسس المنطق الرقمي
Diminished radix complement	متكم الأساس المخفض
Diode Logic (DL)	منطق الديودات
Diode Transistor Logic (DTL)	منطق الترانزيستور ديود
Direct access	ولوج مباشر
Direct Current (DC)	التيار المستمر
Directive	موجه
Discrete information	معلومات متقطعة
Discrete instance of time	لحظات متقطعة زمنياً
Distributive law	قانون التوزيع
Documentation language	لغة وثائقية
Don't Care condition (d)	شروط أو حالات عدم الاكتراث
Dual Integrated Lines (DIL)	صفيين من الخطوط المتكاملة
Duality	الازدواجية

Edge triggered flip - flop	قلاب مقدوح بالحافة
Efficiency	فعالية أو مردود
Embedded	مضمن أو تضمين
Emitter Coupled Logic (ECL)	منطق المشعات المترابطة
Encoder	مرمز
Enhancement mode	نمط أغنائي أو معزز
Error correction	تصحيح الخطأ
Error-detection properties	خاصية كشف الخطأ
Essential hazard	الخطر الأساسي
Even function	التابع الزوجي
Events	أحداث أو وقائع
Excitation table	جدول التحريض
Expander	عنصر توسيعي
Failure	عطل
Fan-out	قدرة الخرج على الإمداد (الخرجية)
Fault	خلل أو عطل
Feature	سمة أو ميزة
Field-Effect Transistor (FET)	ترانزستور الأثر الحقل
Finite State Machine (FSM)	آلة محدودة عدد الحالات
First complement	المتكم للواحد (الأحادي)
Fixed point	الفاصلة الثابتة
Flexibility	مرونة
Flip-Flop (FF)	قلاب أو نطايط أو رجاج
Floating point	الفاصلة العائمة (المتحركة)
Flow chart	مخطط الانسياب (التدفق أو الجريان)
Fraction part	الجزء الكسري

Frequency divider	مقسم التردد
Full Adder (FA)	جامع تام (كامل)
Full Subtractor (FS)	طارح تام
Fundamental mode	النمط الأساسي
Gate delay	تأخير البوابة
Gate implementation	تنجيز بواسطة البوابات
Gate level minimization	اختصار على مستوى البوابة
General- Purpose Digital Computer	الحاسب الرقمي متعدد الأغراض
Generation	جيل
Gigabyte (GB)	جيجا بايت
Global	شامل أو شمولي
Grid	شبكة
Hardware Description Language (HDL)	لغة وصف العتاديات
Hazards	المخاطر
Hexadecimal system	النظام الستة عشري
High impedance	ممانعة عالية
Hold time	زمن الإمساك
Hybrid	هجين
Identifier	معرف أو معين
Implementation	تنجيز أو تحقيق
Implication	التضمين
Incorporated	مكاملة
Index State	دليل الحالة
Individual	إفرادي أو خاص
Inhibition	المنع
Initial	بدئي، أي في البداية

Instantiation	انتساخ
Instruction register	سجل التعليمات
Integer part	الجزء الصحيح
Integrated Circuits (IC)	الدوائر المتكاملة
Interconnection	الوصل البيني
Interface	واجهة ترابط أو تواصل
Inverter	العاكس
Jumper	واصلات بين نقاط مختلفة
Junction Field- Effect Transistor	ترانزستور الأثر الحقل ذو القطبية
Karnaugh map	مخطط أو خارطة كارنوف
Large Scale Integration (LSI)	تكامل واسع النطاق
Latch	ماسك
Leading edge	الحافة القيادية أو الوجه الصاعد
Least Significant Bit (LSB)	البت ذو الوزن أو الدلالة الأدنى
Level mode sequential circuits	الدوائر التتابعية نموذج المستوى
Level of integration	مستوى التكامل
Light Emitting Diode (LED)	ديود ضوئي (مشع للضوء)
Logic simulation	محاكاة المنطق
Logic synthesis	التركيب المنطقي
Mantissa	الجزء العشري
Manufacturer	الصانع أو الشركات الصانعة
Master-Slave flip-flop	قلاب قائد - مقاد
Medium Scale Integration (MSI)	تكامل متوسط النطاق
Megabyte (MB)	ميغا بايت
Merging	دمج
Metal-Oxide Semiconductor (MOS)	معدن - أوكسيد نصف ناقل

Mixed notation	تعبير مختلط
Modeling techniques	تقنيات النمذجة
Module	مجزأ
Monolithic	وحيد البلورة أو القطعة
Monostable	أحادي الاستقرار
Most Significant Bit (MSB)	البت ذو الوزن الأعلى
Multiplexer (MUX)	المضخم
Multiplier	الضارب
Multivibrator	مهتز أو عديم الاستقرار
Negative edge	الوجه أو الحافة السالبة
Nibble	وحدة لقياس المعلومات بطول 4 bits
Noise margin	هامش الضجيج
No critical races	سباقات غير حرجية
Non-volatile	دائم (غير متطاير)
Normalized mantissa	الكسر المعياري
Not Connected (NC)	غير موصول
Null function	تابع معدوم
Octal system	النظام الثماني
Odd parity	النديّة الفردية
Open Collector (OC)	مجمع مفتوح
Over flow	الطفح أو الفيض
Overloaded	تحميل زائد
Parallel binary adder	جامع ثنائي متواز
Parallel load	تحميل أو شحن متواز
Parallel port	بوابة تفرعيه
Parity bit	بتّ النديّة أو بتّ التكافؤ

Parity checker	فاحص النّدية
Parity generator	مولد النّدية
Periodic train of clock	قطار من النبضات الساعية
Physical realization	التحقيق الفيزيائي
Polarity indicator	مؤشر القطبية
Polynomial	متعدد الحدود (كثير حدود)
Positional value	القيمة الموضعية
Positive edge	حافة موجبة
Power dissipation	تبديد الاستطاعة
Precedence	الأسبقية
Precision	الدقة
Present state	الحالة الحالية أو الراهنة
Previous	السابق
Priority Encoder	مرمز الأفضلية
Procedural assignment statements	عبارات إسناد إجرائية
Product of sum	جداء مجاميع
Product term	حد جداء
Programmable Array Logic (PAL)	منطق الصفيفة القابلة للبرمجة
Programmable Logic Array (PAL)	صفيفة منطقية قابلة للبرمجة
Programmable Logic Device (PLD)	تجهيزه منطقية قابلة للبرمجة
Propagation delay	تأخير الانتشار (للإشارة)
Pulse transition	انتقال النبضة
Pulse width	عرض النبضة
Quality	الجودة
Quantity	كمية
Queue	رتل

Race condition	شرط السباق
Radix	جذر أو القاعدة أو الأساس
Reduction	اختصار أو اختزال
Register	سجل
Reliability	الوثوقية
Reminder	الباقي
Resistor – Transistor Logic(RTL)	منطق ترانزستور – مقاومة
Ring counter	عداد حلقي
Ripple counter	عداد تموجي
Rise time	زمن الصعود
Saturation region	منطقة الإشباع
Selection lines	خطوط الانتقاء
Self complementing	إتمام ذاتي
Semiconductor	نصف ناقل
Sequential logic circuit	الدارة المنطقية المتتابعة
Serial adder-subtractor	جامع طارح تسلسلي
Serial addition	جمع تسلسلي
Set / Reset latch	ماسك وضع / إرجاع
Setup time	زمن التهيئة أو زمن الإعداد
Shift -Around register	سجل إزاحة حلقي
Signed binary number	أعداد ثنائية مؤشرة (مقيدة بإشارة)
Signed magnitude	مقدار مقيد بإشارة
Simplification	تبسيط أو اختصار
Simulation	محاكاة
Small Scale Integration (SSI)	تكامل ضيق النطاق
Special application programs	البرامج التطبيقية الخاصة

Special registers	السجلات الخاصة
Specification	الخصائص المميزة
Stability consideration	اعتبارات الاستقرار
Standard circuits	الدوائر القياسية أو المعيارية
Standard form	الشكل القياسي (للتابع المنطقي)
State assignment	إسناد الحالة
State reduction	اختصار الحالة
Stimulus module	مجتزأ المحفز
Storage element	عنصر تخزين
Storage time	زمن التخزين
Substrate	القاعدة أو الأساس
Subtraction circuits	دوائر الطرح
Sum of product	مجموع جداءات
Switch bounce	ارتداد المفتاح
Switch-level modeling	نمذجة على مستوى المفتاح
Switch-tail ring counter	عداد حلقي مبدل النهاية
Synchronous counter	عداد متزامن
Synchronous logic circuit	دائرة منطقية متزامنة
Syndrome	العرض
Synthesis	تركيب
Technology	تقانة
Terabyte (TB)	تيرا بايت
Test bench	منصة اختبار
Textual form	صيغة نصية
Three-level implementation	تنجيز بثلاث مستويات
Three state gate	بوابة بثلاثة حالات

Threshold	عتبة
Timing and control unit	وحدة التوقيت والتحكم
Timing waveforms	موجات توقيت
Toggle	القلب
Top-down design	تصميم من الأعلى إلى الأسفل
Trailing edge	الحافة الأخيرة أو الوجه الهابط
Transistor-Transistor Logic (TTL)	منطق ترانزستور - ترانزستور
Transition diagram	مخطط الانتقال
Trigger	قاذح
Two-level implementation	انجاز (أو تنفيذ) ثنائي المستوى
Two's or Second complement	المتكم لأثنين
Ultraviolet light	الأشعة فوق البنفسجية
Unipolar	وحيد القطبية
Unique pattern	نمط متميز أو منفرد
Unsigned number	عدد غير مؤشر (عديم الإشارة)
Unstable circuit	دائرة عديمة الاستقرار
Unused state	حالة غير مستخدمة
Up-down counter	عداد تصاعدي تنازلي
User-Defined Primitives (UDP)	أوليات معرفة من قبل المستخدم
Valid bit indicator	بت مؤشر الصلاحية
Very Large-Scale Integration (VLSI)	تكامل عال جدا
Volatile memories	ذاكر غير دائمة (أو متلاشية)
Waveforms	أشكال الموجات
Weighted code	الرمزة الموزنة أو المتقلة
Wire	سلك

اللجنة العلمية

الأستاذ المساعد الدكتور: مازن محاييري

الأستاذ المساعد الدكتور: بسام محمد

الأستاذ المساعد الدكتور: مصطفى الموالي

المدقق اللغوي:

الأستاذ الدكتور: رياض العوادة

حقوق الطبع والترجمة والنشر محفوظة لمديرية الكتب والمطبوعات