الفصل الثانى ٢

المتحكم إنتل Intel 8051

The Intel 8051 Microcontroller

obeikandl.com

٢-١ مقدمة

فى عام ١٩٨١ أعلنت شركة Intel عن منتج جديد من المتحكمات وهو المتحكم 8051 بالمواصفات التالية والتى كانت من أعلى المواصفات فى هذا الوقت :

- مسار بيانات ووحدة حساب ومنطق ٨ بت .
- سهولة التقابل معه (توصيل الدوائر الخارجية عليه) .
- سرعات من ١٢ حتى ٣٠ ميجاهرتز .
- أوامر متعددة لنقل البيانات والضرب والقسمة والتعامل على مستوى البت .
- طرق مختلفة للعنونة (التعامل مع الذاكرة) .
- ٤ كيلوبايت ذاكرة برمجة ROM
- ١٢٨ بايت RAM
- ١٢٨ من المسجلات الخاصة SFR .
- إدخال وإخراج للبيانات على التوالى Serial I/O .
- ٣٢ خط لإدخال وإخراج البيانات .
- مؤقتان/عدادان ١٦ بت 16 bit timer counters .

أطلق على هذا المتحكم فى وقته بأنه "نظام على شريحة" "System on a chip" ولقد سمحت شركة Intel فيما بعد للعديد من الشركات بإنتاج هذا المتحكم بإمكانيات مختلفة عن المتحكم الأصلى ولكن بشرط أن تكون هذه المتحكمات متطابقة من حيث مجموعة الأوامر مع المتحكم الأصلى 8051 بحيث إذا نفذ المستخدم أى تطبيق على هذه المتحكمات ، فإنه ينفذ دون أى مشاكل على المتحكم الأصلى . ولقد ساعد ذلك على شيوع هذا المتحكم لدى المستخدمين بدرجة كبيرة .

المتحكم 8051 يتبع متحكمات نيومان ، بمعنى أنه له مسار بيانات واحد (٨ بت) يتم نقل البيانات عليه كما يتم نقل شفرات الأوامر عليه أيضا . ولذلك فإن ذاكرة البرمجة هى ذاكرة البيانات تتكون أى منهما من بايتات كل منها ٨ بت . لذلك أيضا سنجد هنا أن شفرة الأمر الواحد ستتكون فى الغالب من أكثر من بايت واحدة وسيتم إحضارها بأكثر من عملية قراءة واحدة من الذاكرة .

لقد أنتجت شركة Intel إصداران آخران من هذا المتحكم وهما المتحكمان Intel 8031 و Intel 8052 وهما متطابقان تماما مع المتحكم 8051 من حيث الأطراف ولكنهما يختلفان فى التركيب الداخلى بدرجة بسيطة كما يوضح ذلك جدول ٢-١ . نلاحظ من هذا الجدول أن المتحكم 8052 يحتوى فقط مؤقت واحد زيادة عن المتحكم 8051 ، ولذلك فإننا يجب أن نتوقع أن البرامج المكتوبة للمتحكم 8051 تنفذ دون أى مشاكل على المتحكم 8052 وأما العكس فلن يكون بالضرورة صحيح . أما المتحكم 8031 فإنه يختلف عن المتحكم 8051 فى أنه لا يحتوى على ذاكرة برمجة ROM على الإطلاق ولابد من توصيل ذاكرة خارجية عليه حتى يمكن كتابة البرامج فيها .

عملية توصيل ذاكرة خارجية بهذا الشكل تستهلك اثنين من بوابات الإدخال والإخراج المتاحة ويتبقى بوابتان فقط وإن كان يمكن استعواض ذلك عن طريق توصيل شريحة PPI على المتحكم كما سنرى فيما بعد .
المتحكم 8051 يتم إنتاجه بأكثر من شركة والاختلاف الأساسى فى كل هذه المنتجات هو فى نوع ذاكرة البرمجة المستخدمة فى كل منها على النحو التالى :

جدول ٢-١

الخاصية	8051	8052	8031
ذاكرة البرمجة ROM الداخلية	4K bytes	8K bytes	0K bytes
ذاكرة قراءة وكتابة RAM	128 بايت	256 بايت	128 بايت
المؤقتات	2	3	2
خطوط الإخراج والإدخال	32	32	32
مخارج متوالية	1	1	1
مصادر المقاطعة	6	8	6

٢-١-١ المتحكم 8751

هذا المتحكم هو نفسه المتحكم 8051 بداخله 4K بايت ذاكرة برمجة Ultra Violet ROM, UV ROM ، التى يتم مسحها عن طريق تعريضها لأشعة فوق بنفسجية لمدة عشرون دقيقة على الأقل قبل أى عملية تسجيل فيها . ويتطلب ذلك بالطبع نزع شريحة المتحكم من مكانها فى الدائرة ووضعها فى جهاز المسح . بعد ذلك توضع شريحة المتحكم فى جهاز خاص يقوم بتسجيل أو حرق البرنامج عليها . وهذا يعتبر عيب أساسى مع هذه الشريحة حيث يسبب ذلك صعوبة فى التعامل معها .

٢-١-٢ إصدارات شركة أتمل من المتحكم 8051

منتجات شركة أتمل من المتحكمات 8051 متطابقة تماما مع المتحكم الأصلى من حيث مجموعة الأوامر مع الاختلاف فى عدد الأطراف فى بعض الإصدارات مع الاختلاف الأساسى وهو أنها تستخدم ذاكرة برمجة من النوع اللحظى أو flash ROM . هذه الذاكرة هى ذاكرة ROM يمكن مسحها إلكترونيا وفوريا حيث يستغرق مسحها حوالى ثانية واحدة فقط مقارنة مع 20 دقيقة فى الذاكرة UV ROM . عملية تسجيل البرنامج فى هذه الذاكرة ما زالت تحتاج لجهاز أو دائرة لحرق البرنامج على المتحكم ، مما يعنى أنه لابد من وضع المتحكم فى جهاز خاص يقوم بتسجيل أو حرق البرنامج . هذا يعنى أن الاستفادة الوحيدة من استخدام الذاكرة الفورية أو اللحظية هو تقليل زمن مسح محتويات الشريحة وبالتالى زمن إعداد التطبيق على المتحكم . جدول ٢-٢ يبين العديد من الإصدارات التى تنتجها شركة أتمل من هذا المتحكم . إقرأ هذا الجدول بعناية ولاحظ الاختلاف فى كمية ذاكرة البرمجة flash

ROM ، والذاكرة RAM وعدد خطوط إدخال وإخراج البيانات ، وبالتالى عدد أطراف الشريحة وعدد المؤقتات المستخدمة . لكن بالرغم من ذلك فإن مجموعة الأوامر متطابقة تماما بينها جميعا ومع المتحكم الأساسى 8051 .

جدول ٢-٢

Part Number	ROM Kbytes	RAM bytes	I/O pins	Timers	Interrupt	Vcc	Package
AT89C51	4	128	32	2	6	5	40
AT89LV51	4	128	32	2	6	3	40
AT89C1051	1	64	15	1	3	3	20
AT89C2051	2	128	15	2	6	3	20
AT89C52	8	128	32	3	8	5	40
AT89LV52	8	128	32	3	8	3	40

٢-١-٣ إصدارات شركة دالاس Dallas من المتحكم 8051

أنتجت شركة دالاس Dallas semiconductors إصدارها من المتحكم 8051 الذى يتميز باحتوائه على ذاكرة RAM قادرة على الاحتفاظ بمحتوياتها بالرغم من انقطاع القدرة عنها وتسمى Non Volatile RAM, NV RAM .هذه الذاكرة تحتوى على بطارية داخلها تستخدم للحفاظ على محتوياتها بعد انقطاع القدرة لمدة تصل حتى عشر سنوات ، هذا بجانب أنها مصنعة باستخدام تقنية CMOS التى تستهلك القدرة بأقل ما يمكن ، ولذلك فإن سعر هذه الذاكرة يكون مرتفعا وبالتالى فإن سعر المتحكم الذى يستخدمها يكون أغلى من المتحكمات الأخرى التى تستخدم الذاكرة UV ROM أو الذاكرة اللحظية Flash memory . جدول ٢-٣ يبين بعض إصدارات شركة دالاس من هذا المتحكم ونلاحظ أنها أعطته الرقم DS5000 بدلا من 8051 ولكنه مازال متطابق معه من حيث مجموعة الأوامر . الإصدار الذى يحتوى الحرف T يعنى أنه يحتوى ساعة تعطى الزمن الحقيقى .

لاحظ أن وجود ساعة تعطى الزمن الحقيقى يختلف عن وجود مؤقت وهو أن ساعة الزمن الحقيقى تعطى الوقت بالساعة والدقيقة والثانية كما تعطى اليوم والشهر والسنة وهذا يفيد فى الكثير من التطبيقات .

يتميز استخدام هذا النوع من الذاكرة NV RAM فى المتحكم أن عملية مسح البرنامج وإعادة تسجيله أصبحت أسهل بكثير ولا تحتاج لنزع شريحة المتحكم من الدائرة ، بل يمكن أن يتم ذلك فى أثناء وجود المتحكم فى مكانه على الدائرة . بل الأخطر من ذلك أنه يمكن تغيير بايت واحدة على الشريحة دون الحاجة لمسح محتويات الشريحة بالكامل كما كان يحدث فى الإصدارات السابقة .

جدول ٢-٣

Part number	NV RAM	RAM	I/O pins	Timers	Interrupts	Vcc	Package
DS5000-8	8K	128	32	2	6	5V	40
DS5000-32	32K	128	32	2	6	5V	40
DS5000T-8	8K	128	32	2	6	5V	40
DS5000T-32	32K	128	32	2	6	5V	40

٢-١-٤ المتحكمات التى تبرمج مرة واحدة One Time Programmable, OTP

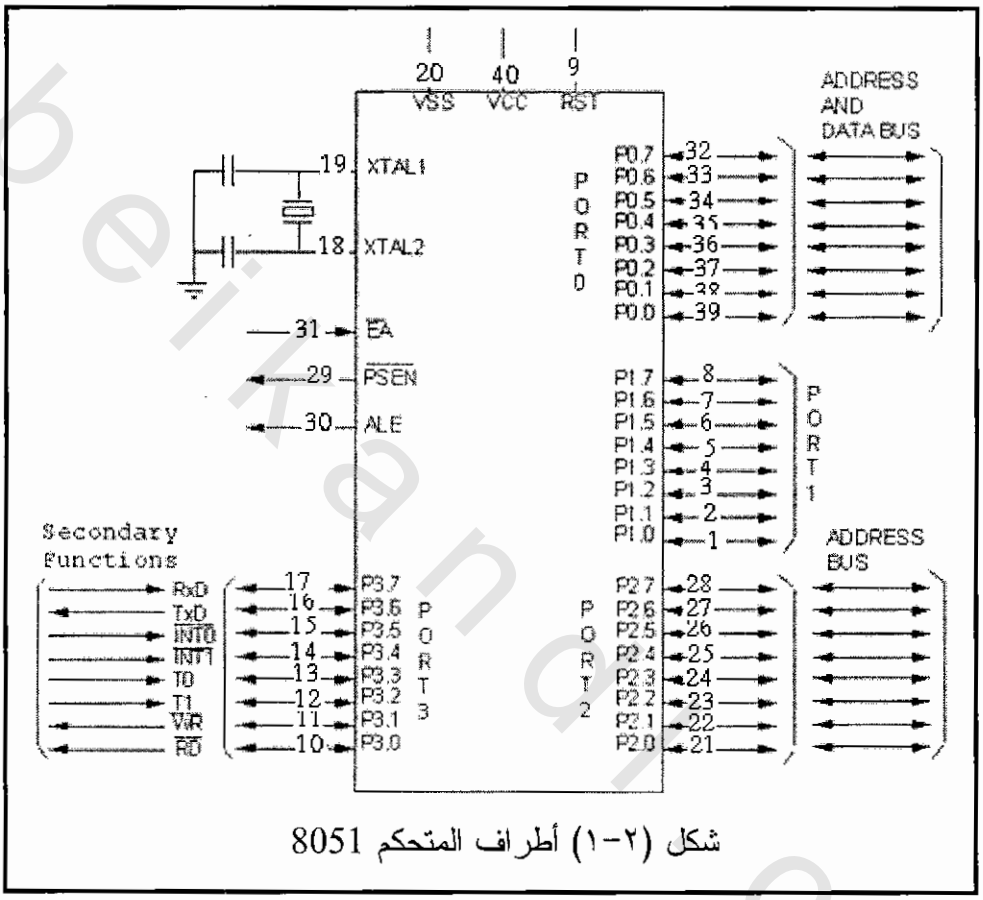
هذه الإصدارات تحتوى ذاكرة برمجة ROM يتم برمجتها مرة واحدة فقط ولا يمكن مسحها . تستخدم هذه الإصدارات عادة بعد تجربة البرنامج ووضعه فى صورته النهائية التى لا يكون هناك أى حاجة بعدها لأى تعديل فى البرنامج . هنا يتم حرق البرنامج على هذا النوع من الإصدارات التى يتم إنتاجها بكميات كبيرة تمهيدا لتسويقها حيث تكون التكلفة أقل فى هذه الحالة .

٢-٢ توصيف لأطراف الشريحة

شكل (٢-١) يبين أطراف المعالج 8051 (٤٠ طرف) مقسمة إلى مجموعات حتى يسهل دراستها . يمكن تقسيم هذه الأطراف إلى المجموعات التالية :

- ثمانية أطراف P0.0 حتى P0.7 وهى تمثل أطراف البوابة رقم صفر وهى تمثل خطوط إدخال وإخراج للبيانات . هذه الأطراف تستخدم أيضا كخطوط عناوين وبيانات Address/Data ، A/D0 حتى A/D7 فى حالة التعامل مع ذاكرة خارجية بالاستعانة بخط التحكم ALE وسيأتى شرح مفصل لذلك .
- ثمانية أطراف P1.0 حتى P1.7 وهى تمثل أطراف البوابة رقم واحد وهى تمثل خطوط إدخال وإخراج للبيانات وهى ليست مخلوطة مع وظائف أخرى كما فى حالة البوابة صفر .
- ثمانية أطراف P2.0 حتى P2.7 وهى تمثل أطراف البوابة رقم 2 وهى تمثل خطوط إدخال وإخراج للبيانات . هذه الأطراف تستخدم أيضا كخطوط عناوين للنصف الأعلى من مسار العناوين A0 حتى A7 فى حالة التعامل مع ذاكرة خارجية .
- ثمانية أطراف P3.0 حتى P3.7 وهى تمثل أطراف البوابة رقم 3 وهى تمثل خطوط إدخال وإخراج للبيانات . هذه الأطراف تستخدم أيضا كخطوط تحكم فى أداء المتحكم وسيأتى شرح تفصيلى لوظيفة كل خط من هذه الخطوط فيما يعد .

- خطوط التحكم $\overline{PSEN}$ $\overline{EA}$, ALE, RST وهذه أيضا سيأتى شرح تفصيلى لوظيفة كل منها عند استخدامه .
- طرفان XTAL2 ,XTAL1 ويتم توصيل بللورة crystal بينهما بالتردد المطلـوب الذى ستعمل عنده الشريحة .
- خطوط القدرة Vcc والأرضى Vss .

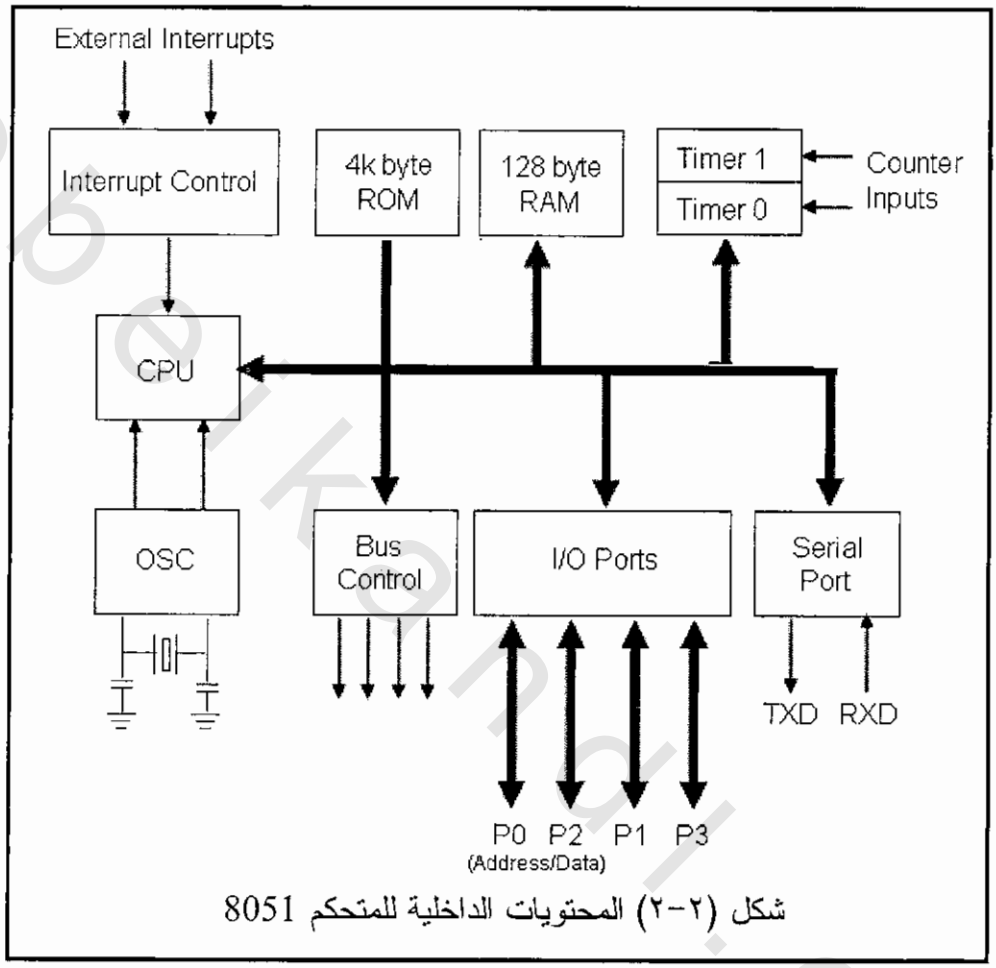


شكل (٢-١) أطراف المتحكم 8051

٢-٣ المحتويات الداخلية للمتحكم 8051

شكل (٢-٢) يبين رسما صندوقيا للتركيب الداخلى لهذه الشريحة . نلاحظ من هـذا الشكل وجود مسار بيانات من ٨ بت يصل بين وحدة المعالجة المركزية cpu وجميع المكونات بما فى ذلك الذاكرة RAM وذاكرة البرمجة ROM التى تحتوى شـفرات البرنامج التطبيقى . مهمة وحدة المعالجة المركزية هى إجراء العمليـات الحسـابية والمنطقية والتى منها الضرب والقسمة التى لم تكـن موجـودة فـى الكثيـر مـن المعالجات ٨ بت . يحتوى المتحكم 8051 على مركم accumulator أو المسـجل A والذى تتم من خلاله عمليات الإدخال والإخراج وكذلك عمليات الحساب والمنطق .

يحتوى المتحكم أيضا على مسجل عام اسمه المسجل B يستخدم مع المركم فى إجراء عمليات الضرب والقسمة كما سنرى عند دراسة هاتين العمليتين . جميع المكونات الأخرى سيتم الحديث التفصيلى عن كل منها فى جزء منفصل .

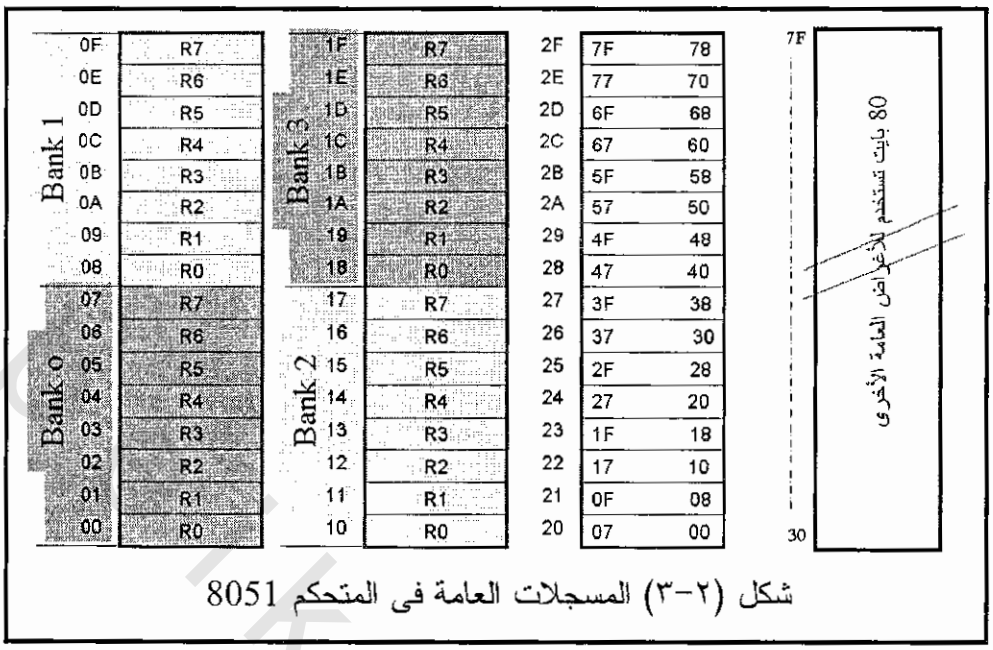


شكل (٢-٢) المحتويات الداخلية للمتحكم 8051

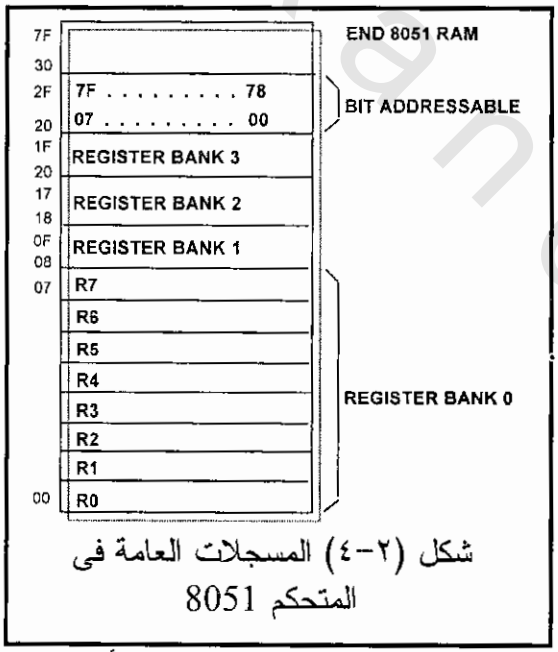
٢-٤ المسجلات العامة (RAM)

يحتوى المتحكم 8051 على 128 بايت من ذاكرة القراءة والكتابة يمكن النظر إليها على أنها مسجلات عامة داخل المتحكم . أول 32 بايت من هذه المسجلات تشغل العناوين من 00H إلى 1FH ويمكن التعامل مع كل منها من خلال هذا العنوان كما فى شكل (٢-٣) . يمكن أيضا التعامل مع هذه المسجلات من خلال اسم لكل منها R0 حتى R7 على حسب البلوك الذى يقع فيه كل مسجل أو كل بايت من خلال ٢ بت فى مسجل الحالة يستخدمان لتحديد بلوك من الأربعة بلوكات المراد التعامل معه كما سنرى عند شرح مسجل الحالة .

obeikandl.com



شكل (٢-٣) المسجلات العامة فى المتحكم 8051

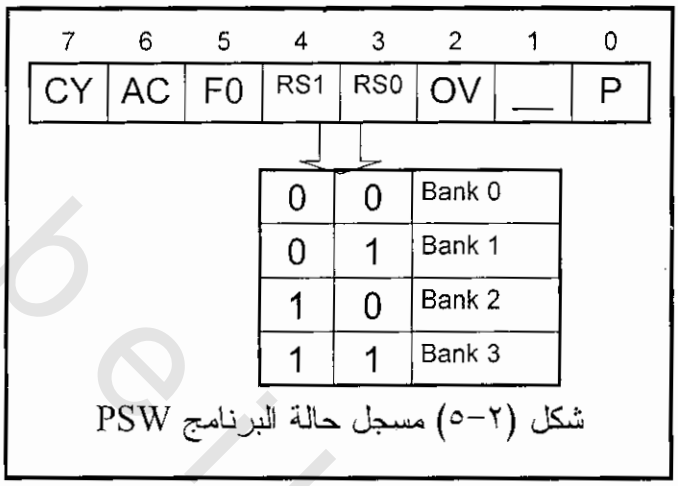


شكل (٢-٤) المسجلات العامة فى المتحكم 8051

المسجلات من 00H حتى 1FH (الأربع بلوكات) كلها يتم التعامل معها على مستوى البايت فقط ، أى أنه لا يمكن التعامل مع بت واحدة من بتات أى واحد من هذه المسجلات . المسجلات من 20H حتى 2FH وعددها ١٦ مسجل يمكن التعامل مع بتاتها منفردة لأن كل بت من بتات هذه المسجلات لها عنوان يميزها وعناوين هذه البتات كما فى شكل (٢-٣) يمتد من البت التى عنوانها 00H وهى أول بت فى المسجل 20H حتى البت التى عنوانها 7FH وهى آخر بت فى المسجل 2FH كما فى الشكل . هذه المسجلات يمكن التعامل معها أيضا على مستوى البايت إن أردنا من خلال العناوين الموضحة فى الشكل . باقى المسجلات من العنوان 30H حتى العنوان 7FH هى مسجلات عامة يتم التعامل معها على مستوى البايت فقط . شكل (٢-٤) يبين نفس مجموعة المسجلات الموجودة فى شكل (٢-٣) ولكن بطريقة أخرى أكثر شمولية .

٢-٥ مسجل حالة البرنامج Program Status Word, PSW

مثل جميع المعالجات والمتحكمات يحتوى المتحك 8051 على مسجل حالة يبين حالة آخر عملية نفذها المتحكم بجانب بعض المعلومات الأخرى كما سنرى فى هذا الجزء وكما هو موضح فى شكل (٢-٥) الذى يبين بتات هذا المسجل :



شكل (٢-٥) مسجل حالة البرنامج PSW

البت 7 CY, Carry fag وهى علم الحمل الذى يكون واحد إذا حدث حمل carry من آخر بت أو استلاف إلى آخر بت فى نتيجة أى عملية جمع أو طرح .

البت 6 AC, Auxiliary carry flag وهى علم الحمل النصفى half carry flag الذى يكون واحد عند حدوث حمل من البت D3 إلى البت D4 فى نتيجة أى عملية جمع ، أو استلاف من D4 إلى D3 فى حالة إجراء أى عملية طرح .

البت 5 F0, User defined flag ، وهو "علم مستخدم" ، يستخدمه لأى غرض من أغراض البرمجة .

البتات 4, 3 Register bank Selector, RS1,0 يستخدمها المستخدم لتحديد بنك المسجلات العامة الذى يريد التعامل معه حيث يمكن اختيار واحد من أربع بنكات يمكن للمستخدم أن يختار أيها عن طريق كتابة وحايد أو أصفار فى هذه البتات كما فى شكل (٢-٥) .

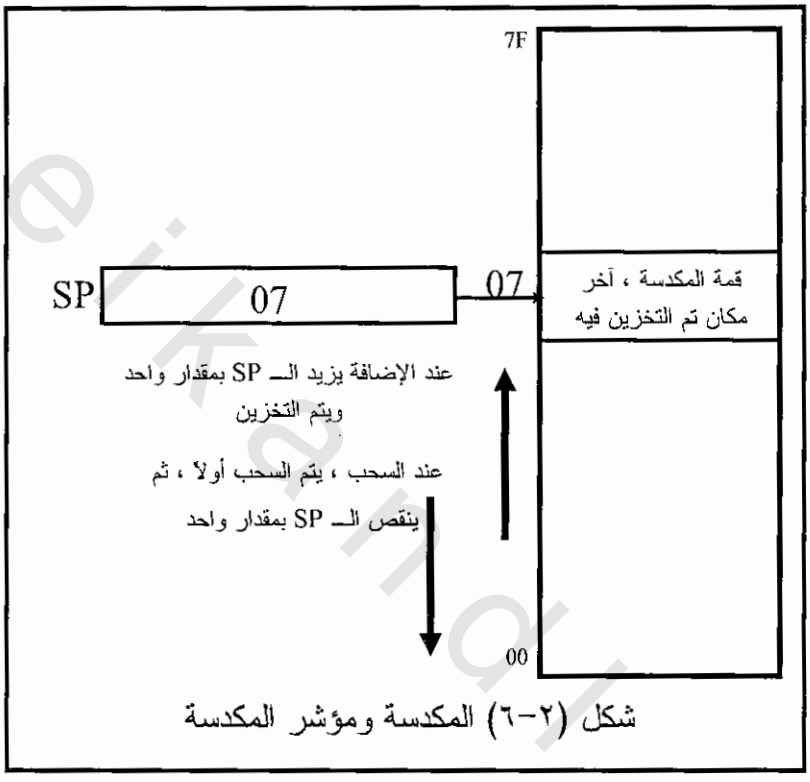
البت 2 OV, Over flow flag علم الفيضان الذى يصبح واحد عند حدوث فيضان على خانة الإشارة فى حالة التعامل مع أرقام ذات إشارة signed numbers وسيأتى الحديث عن هذا العلم بالتفصيل عند الحديث عن أو مر الحساب .

البت 0 Parity flag, P وهذا العلم يعكس عدد الوحايد فى المركم A ، فإذا كان عدد الوحايد فى هذا المسجل فردى odd فإن هذا العلم يكون واحد ، ويكون صفر بالطبع إذا كان عدد الوحايد زوجى even .

obeikandi.com

٢-٦ المكدسة ومؤشر المكدسة Stack and Stack Pointer

المكدسة stack هى بمعناها العام جزء من الذاكرة يستخدم لتخزين عناوين العودة عند النداء على برنامج فرعى subroutine أو حدوث مقاطعة Interrupt . قمة هذه المكدسة أو آخر مكان تم الدفع فيه يتم الإشارة إليه عن طريق مسجل مؤشر المكدسة stack pointer, SP كما فى شكل (٢-٦) . أى أن مؤشر المكدسة يحتوى عنوان آخر مكان فى المكدسة تم التخزين فيه .



شكل (٢-٦) المكدسة ومؤشر المكدسة

كما هو مبين فى شكل (٢-٦) فإن المؤشر SP يتكون من ٨ بت لذلك فإنه يمكنه الإشارة من العنوان 00H حتى العنوان 7FH وهو آخر عنوان متاح فى الذاكرة العامة . القيمة التلقائية default التى تكون موجودة فى المسجل SP عند بدأ تشغيل المتحكم هى العنوان 07H حيث يمكن تغيرها لأى قيمة أخرى عند الحاجة لذلك . مع الإضافة فى المكدسة فإن المسجل SP يزداد بمقدار واحد ثم يتم التخزين أو الدفع فى هذا المكان . أما عند السحب من المكدسة فإنه يتم السحب أولا من العنوان المشار إليه بالمؤشر SP ثم بعد ذلك يتم إنقاص المسجل SP بمقدار واحد . عملية زيادة أو إنقاص المؤشر SP بمقدار واحد تتم عن طريق المتحكم ولا يقوم بها المستخدم . يمكن دفع محتويات أى مسجل فى المكدسة باستخدام الأمر PUSH

obeikandi.com

بشرط استخدام العنوان المباشر للمسجل مثل PUSH 0E0 . وبنفس الطريقة يمكن الاستدعاء من المكدسة لأى مسجل بالأمر POP .

٢-٧ مسجلات الوظائف الخاصة

Special Function Registers, SFR

تبدأ عناوين هذه الجموعة من المسجلات من العنوان 80H (العنوان التالى لآخر عنوان فى ال RAM) حتى العنوان FFH . هذه المسجلات بجانب إمكانية استخدامها فى أغراض البرمجة المختلفة فإنها تستخدم فى بعض الأغراض الخاصة بأداء المتحكم نفسه كما سنرى . بعض هذه المسجلات يتم التعامل معها على مستوى البايت الكاملة ، وبعضها يمكن التعامل معه على مستوى البت أو البايت . جدول ٢-٤ يبين كل هذه المسجلات ووظيفة وعنوان كل منها . ليس كل العناوين من 80H حتى FFH مستخدمة بهذا النوع من المسجلات ، ولكن هناك بعض العناوين الغير مستخدمة ومحاولة الكتابة فيها غير مسموح ولن يكون معروف النتيجة . شكل (٢-٧) يبين ترتيب آخر لعناوين هذه المسجلات يتضح منه العناوين المستخدمة وغير المستخدمة .

جدول ٢-٤

اسم المسجل	عنوانه	وظيفة المسجل
مسجلات خاصة بوحدة المعالجة المركزية cpu		
A	0E0	المركم Accumulator
B	0F0	يساعد فى العمليات الحسابية
DPH	83	مؤشر البيانات الأعلى Data Pointer High
DPL	82	مؤشر البيانات الأدنى Data Pointer Low
SP	81	مؤشر المكدسة Stack Pointer
PSW	0D0	مسجل حالة البرنامج Program Status word
مسجلات الإدخال والإخراج		
P0	80	ماسك بيانات بوابة Input/Output Port latch
P1	90	ماسك بيانات بوابة Input/Output Port latch
P2	0A0	ماسك بيانات بوابة Input/Output Port latch
P3	0B0	ماسك بيانات بوابة Input/Output Port latch
مسجلات الاتصالات على التوالى		
SCON	98	مسجل تحكم بوابة البيانات المتوالية Serial port control
SBUF	99	ماسك البيانات المتوالية Serial port data buffer
PCON	87	التحكم فى القدرة Power Control
مسجلات المقاطعة		
IP	0B8	أولوية المقاطعة Interrupt Priority

IE	0A8	Interrupt Enable control التحكم فى تنشيط المقاطعة
مسجلات المؤقتات		
TMOD	89	Timer/Counter mode control التحكم فى حالة المؤقت
TCON	88	Timer/Counter control التحكم فى أداء المؤقت
TL0	8A	Timer 0 Low byte البايت الأدنى من المؤقت الأول
TH0	8C	Timer 0 High byte البايت الأعلى من المؤقت الأول
TL1	8B	Timer 1 Low byte البايت الأدنى من المؤقت الثانى
TH1	8D	Timer 1 High byte البايت الأعلى من المؤقت الثانى

F8								
F0	B							
E8								
E0	ACC							
D8								
D0	PSW							
C8								
C0								
B8	IP							
B0	P3							
A8	IE							
A0	P2							
98	SCON	SBUF						
90	P1							
88	TCON	TMOD	TL0	TL1	TH0	TH1		
80	P0	SP	DPH	DPL				PCON

ا لتعامل معهذه العناوين الغير
مستخدمة غير مسموح به ، و إذا
تم لا يكون معروف النتيجة !!

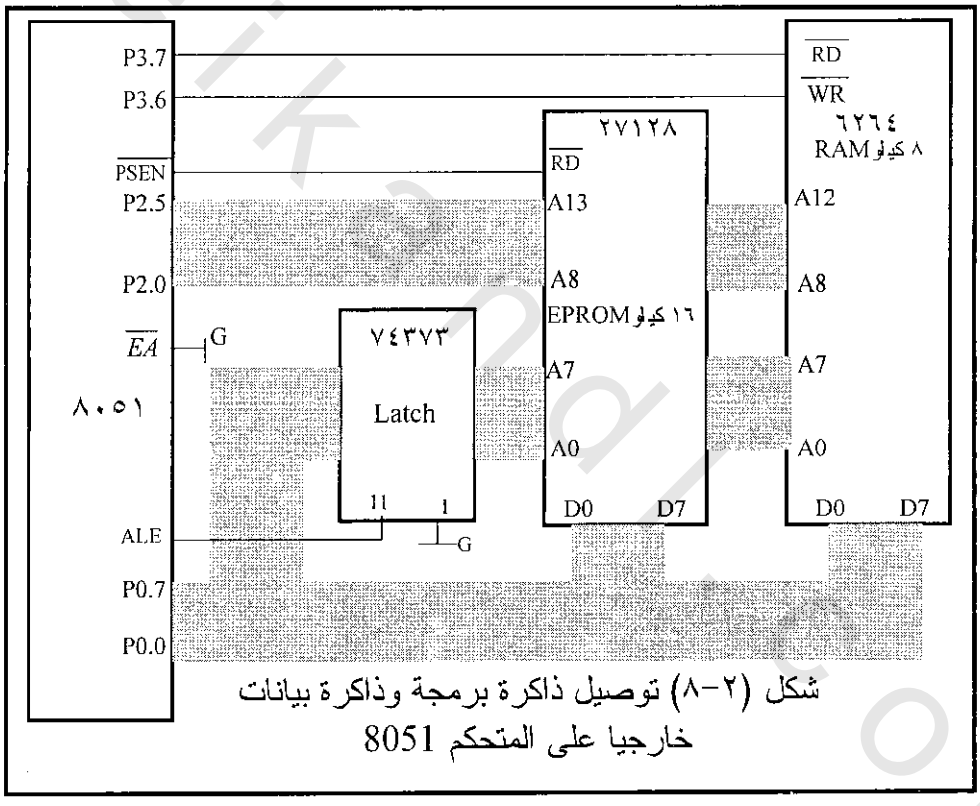
شكل (٢-٧) ترتيب آخر للمسجلات الخاصة فى المتحكم 8051

المسجلات الخاصة الموجودة فى شكل (٢-٧) يمكن التعامل معها بأسمائها مثل SP أو PSW وهكذا ، أو من خلال عناوينها . المسجلات الموجودة فى العمود الأيسر من الشكل هى فقط يمكن التعامل معها على مستوى البت أو البايت وأما باقى المسجلات الخاصة فيتم التعامل معها على مستوى البايت فقط .

obeikandl.com

٢-٨ التعامل مع ذاكرة خارجية فى المتحكم 8051

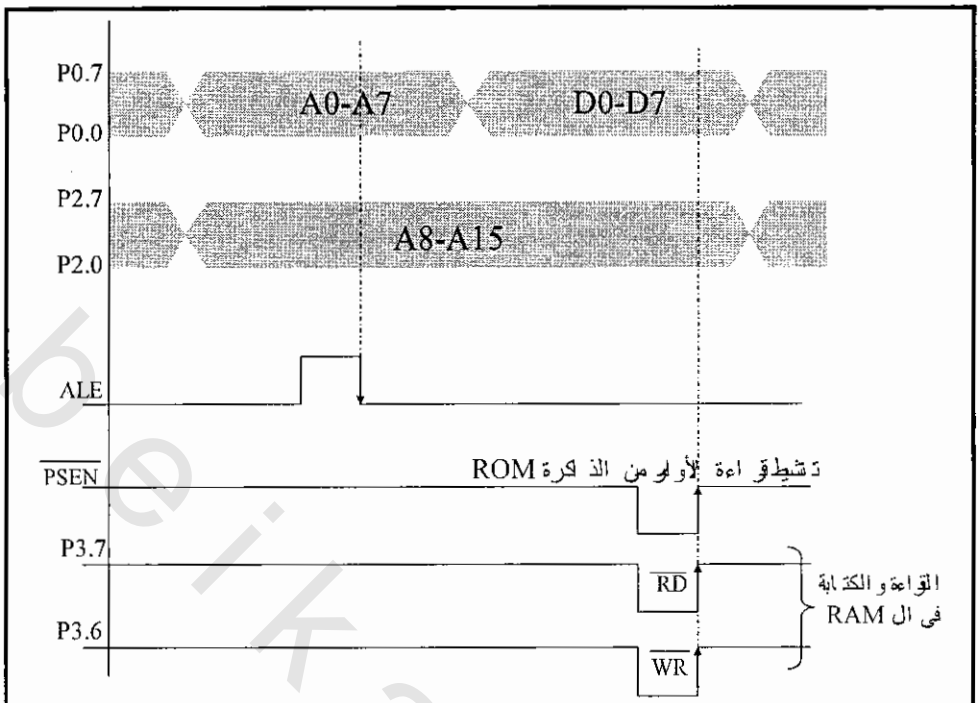
كما رأينا فإن المتحكم 8051 يحتوى 4K بايت ذاكرة برمجة ROM يمكن كتابة البرامج فيها ، ولكن قد يكون هناك بعض التطبيقات التى تحتاج لذاكرة برمجة أكثر من ذلك وهنا سنحتاج لتوصيل ذاكرة خارجية . الشيء الآخر هو أن المتحكم 8031 وهو يعتبر أحد إصدارات المتحكم 8051 لا يحتوى ذاكرة برمجة أصلا ، وهنا لابد من توصيل ذاكرة خارجية . هناك حقيقة مهمة يجب أن نتذكرها وهى أنه يمكننا توصيل 64K ذاكرة برمجة خارجية ROM ، كما يمكننا توصيل 64K ذاكرة بيانات خارجية RAM بالإضافة لذاكرة البرمجة . شكل (٢-٨) يبين طريقة توصيل 16K بايت ROM و 8K بايت RAM على المتحكم وسنقدم شرحا تفصيليا لذلك فى النقاط التالية :



شكل (٢-٨) توصيل ذاكرة برمجة وذاكرة بيانات خارجيا على المتحكم 8051

- خطوط البوابة P0.0 حتى P0.7 تحمل خليط من إشارة العناوين والبيانات للنصف الأول من مسار العناوين A/D0 حتى A/D7 ولابد من فصل إشارة العناوين وحدها حتى يمكن توصيلها على أطراف العناوين فى شرائح الذاكرة . يتم ذلك بالاستعانة بالطرف ALE .

- الطرف Address Latch Enable, ALE طرف يحمل إشارة خارجة من المتحكم ، هذا الطرف يكون دائما بصفر إلا عندما تكون الإشارة على خطوط البوابة P0 تمثل إشارة عناوين فإن المتحكم يضع الطرف ALE=1 . لذلك فإنه كما فى شكل (٢-٨) تم توصيل هذا الطرف على طرف تنشيط الماسك 74373 بحيث عند نزول الطرف ALE من واحد لصفر يتم مسك هذه الإشارة لتمثل إشارة العناوين A0 حتى A7 حيث يمكن توصيلها على خطوط عناوين الذاكرة المقابلة كما بالشكل .
- خطوط البوابة P0 تم توصيلها كخطوط بيانات مباشرة على أطراف البيانات لشرائح الذاكرة D0 حتى D7 كما بالشكل . وهنا ربما يسأل سائل ويقول إنه فى أثناء كون الإشارة على البوابة P0 تمثل عناوين فإن شرائح الذاكرة ستقرأها على أنها بيانات ، فما العمل؟ الإجابة هى أن ذلك لا يمكن أن يحدث نظرا لتوقيت الإشارة على خطوط التحكم $\overline{RD}$ و $\overline{WR}$ القادمة من الأطراف P3.7 و P3.6 وكذلك خط التحكم $\overline{PSEN}$.
- شكل (٢-٩) يبين التوقيت الحادث على خطوط التحكم $\overline{RD}$ و $\overline{WR}$ و $\overline{PSEN}$ مع الطرف ALE وكيف أن هذه الأطراف لن تكون فعالة إلا عندما تكون الإشارة الخارجة من البوابة P0 تمثل بيانات فعلية . وعلى ذلك فلا خوف من أن تقرأ شرائح الذاكرة إشارة العناوين على أنها بيانات .
- الطرف $\overline{PSEN}$ يقوم المتحكم بتنشيطه أثناء عملية إحضار شفرات الأوامر ، لذلك تم توصيله على خط تنشيط الخرج أو القراءة $\overline{RD}$ من ذاكرة البرمجة كما فى شكل (٢-٨) . إن هذا مما يجعل المدى العنوانى لذاكرة البرمجة 64K بايت مختلف تماما عن المدى العنوانى لذاكرة البيانات 64K بايت أخرى .
- الإشارتان $\overline{RD}$ و $\overline{WR}$ القادمتان من الطرفين P3.7 و P3.6 لا ينشطهما المتحكم إلا فى حالة القراءة أو الكتابة فى ذاكرة البيانات فقط . ولذلك فلا دخل لهما بالقراءة من ذاكرة البرمجة . وهذا يؤكد أيضا أن المدى العنوانى لذاكرة البرمجة منفصل عن المدى العنوانى لذاكرة البيانات .
- الطرف External Address, EA هذا الطرف يمثل طرف دخل للمتحكم يجب أن يوضع بصفر أى يوصل بالأرضى كما فى شكل (٢-٨) عند التعامل مع ذاكرة خارجية ، ويوصل على Vcc عند التعامل مع الذاكرة الداخلية . إذا كان البرنامج من الكبر بحيث أنه يشغل كل ذاكرة البرمجة الداخلية وجزء من الذاكرة الخارجية فماذا نفعل بالطرف EA؟ . فى هذه الحالة يتم توصيل الطرف EA على Vcc حتى يجبر المتحكم على البدأ بقراءة البرنامج من الذاكرة الداخلية ، وعند الانتهاء من الذاكرة الداخلية ينتقل المتحكم وحده إلى الذاكرة الخارجية . إن ذلك يعنى أن المدى العنوانى لذاكرة البرمجة يبدأ بالجزء الموجود داخل شريحة المتحكم من العنوان 0000H حتى العنوان 0FFFH وأما المدى من 1000H حتى FFFFH فهو الموجود خارج شريحة المتحكم .



شكل (٢-٩) توقيت التعامل مع الذاكرة

• أوامر التعامل مع الذاكرة الخارجية سندرسها بالتفصيل بعد قليل .

٢-٩ طرق العنونة Addressing Modes

يقصد بطرق العنونة هو كيف ستتمكن وحدة المعالجة المركزية cpu فى المتحكم من التعامل أو استدعاء البيانات سواء كانت هذه البيانات فى ذاكرة أو فى أحد المسجلات أو كانت عبارة عن ثوابت .

٢-٩-١ العنونة الفورية Immediate addressing

كما يوحى اسم هذه الطريقة فإن مصدر البيانات يكون فى البايت التالية لشفرة الأمر، وتستخدم هذه الطريقة فى تحميل المسجلات بثوابت . تستخدم علامة الشباك # للدلالة على هذه الطريقة كما فى الأمثلة التالية :

```
MOV A,#25 ;    تحميل المسجل A بالثابت 25
MOV R4,#62 ;   تحميل المسجل R4 بالثابت 62
```

الأوامر من هذا النوع تكتب فى 2 بايت . فى حالة التعامل مع مسجل 16 بت فإن الأمر يكتب فى 3 بايت كما يلى :

```
MOV DPTR,#33FF ;  تحميل المسجل DPTR بالثابت 33FF
```

obeikandl.com

لاحظ أنه يمكننا التعامل مع المسجل DPTR كمسجل ١٦ بت كما سبق ، أو كممسجلين كل منهما ٨ بت كما يلى :

```
MOV DPL,#55
MOV DPH,#77
```

حيث DPL و DPH هما البايت ذات القيمة الأدنى والبايت ذات القيمة العليا من المسجل DPTR .

٢-٩-٢ عنونة المسجلات Register addressing mode

هنا يكون طرفى التعامل مع البيانات مسجلات يتم ذكرها بإسمها بشرط أن يكون المسجل A واحد منهما كما فى الأمثلة التالية :

```
MOV A,R1  ; نسخ محتويات المسجل R1 فى المسجل A
MOV R7,A  ; نسخ محتويات المسجل A فى المسجل R7
ADD A,R3  ; جمع محتويات المسجل R3 مع المسجل A ووضع النتيجة فى A
```

لاحظ أنه لابد من تطابق مصدر وهدف البيانات من حيث عدد البتات فى كل منهما وتذكر أن هذه الأوامر تكتب فى بايت واحدة فقط .

٢-٩-٣ العنونة المباشرة Direct addressing mode

هنا يتم التعامل مع المسجلات العامة (ال 128 بايت فى ال RAM الداخلية من العنوان 00 حتى 7F) والمسجلات الخاصة SFR (من العنوان 80 حتى FF) من خلال عناوينها وليس من خلال أسمائها . لاحظ أن التعامل يكون مع البانك المحدد فى المسجل PSW بالبتات RS0 و RS1 . يمكن أن يكون أحد طرفى التعامل (إما مصدر المعلومة أو هدفها) من خلال عنوانه المباشر ، كما يمكن أن يكون الطرفين عنونة مباشرة . كما فى الأمثلة العامة التالية :

```
MOV A,add
MOV add,A
MOV Rr,add
MOV add,Rr
MOV add,#n
MOV add1,add2
```

أو الأمثلة المحددة التالية :

```
MOV A,80h
MOV 80h,A
MOV R0,12h
MOV 12,R0
MOV 3A,#3A
MOV 0A8h,77h
```

لاحظ أنه إذا كان أى رقم ستعشرى يبدأ بحرف مثل الرقم A8 فى المثال السابق فإنه لابد من وضع صفر أمامه ليكتب هكذا 0A8 . كما نرى فإن هذه الأوامر ستكتب فى ٢ بايت أو ٣ بايت على حسب طبيعة الأمر .

٢-٩-٤ العنونة الغير المباشرة Indirect addressing mode

الأمر MOV R1,#04 يضع الرقم 04 فى المسجل R1 . لو وضعنا الأمر MOV@R1,#35 فإن المتحكم سيضع الرقم 35 فى العنوان 04 الموجود فى المسجل R1 وليس فى المسجل R1 مباشرة . باختصار هنا يكون العنوان الذى سيتم التعامل معه موجود فى أحد المسجلين **R0 أو R1** فقط . هذه العناوين يجب أن تكون فى ال RAM من العنوان 00 حتى 7F . يستخدم الحرف @ للدلالة على هذا النوع من العنونة كما فى الأمثلة العامة التالية :

```
MOV @Rp,#n
MOV @Rp,add
MOV @Rp,A
MOV add,@Rp
MOV A,@Rp
```

أو الأمثلة المحددة التالية :

```
MOV A,@R0
MOV @R1,#35h
MOV 05h,@R1
MOV @R1,A
MOV @R0,80h
```

٢-١٠ مجموعات أوامر المتحكم 8051

٢-١٠-١ مجموعة أوامر نقل البيانات Data transfer

تختص هذه المجموعة بنقل البيانات من مصدر إلى هدف وكل من المصدر أو الهدف من الممكن أن يكون مسجلا أو بايت ذاكرة بأى طريقة من طرق العنونة السابقة . جدول ٢-٥ يبين هذه الأوامر . كل هذه الأوامر لها الشفرة الأسمبلى المعروفة فى الكثير من المعالجات وهى الأمر MOV . مصدر البيانات لا يمكن أن يكون معلومة فورية . فى العنونة المباشرة لابد أن يكون أحد طرفى الأمر هو المسجل A . عند التعامل مع مساحة الذاكرة المخصصة للمسجلات الخاصة (من 80 حتى FF) ، لابد أن يكون العنوان المستخدم هو عنوان لمسجل فعلى .

جدول ٢-٥

الأمر	وصف الأمر	بايتات	نبضات
MOV A,Rn	نقل محتويات مسجل Rn للمسجل A	1	12
MOV A,direct	نقل محتويات عنوان مباشر إلى المسجل A	2	12

MOV A,@Ri	نقل محتويات عنوان غير مباشر إلى المسجل A	1	12
MOV A,#data	نقل معلومة فورية للمسجل A	2	12
MOV Rn,A	نقل محتويات المسجل A إلى المسجل Rn	1	12
MOV Rn,direct	نقل محتويات عنوان مباشر إلى المسجل Rn	2	24
MOV RN,#data	نقل معلومة فورية للمسجل Rn	2	12
MOV direct,A	نقل محتويات المسجل A إلى عنوان مباشر	2	12
MOV direct,Rn	نقل محتويات مسجل Rn إلي عنوان مباشر	2	24
MOV direct,direct	نقل محتويات عنوان مباشر إلى عنوان مباشر	3	24
MOV direct,@Ri	نقل محتويات عنوان غير مباشر إلى عنوان مباشر	2	24
MOV direct,#data	نقل معلومة فورية إلى عنوان مباشر	3	24
MOV @Ri,A	نقل محتويات المسجل A إلى عنوان غير مباشر	1	12
MOV @Ri,direct	نقل محتويات عنوان مباشر إلى عنوان غير مباشر	2	24
MOV @Ri,#data	نقل معلومة فورية إلى عنوان غير مباشر	2	12
MOV DPTR,#data16	تحميل مؤشر البيانات بمعلومة من 16 بت	3	24
MOVC A,@A+DPTR	نقل معلومة مشار إليها بمؤشر البيانات والمسجل A إلى المسجل A	1	24
MOVC A,@A+PC	نقل معلومة مشار إليها بعداد البرنامج PC والمسجل A إلي المسجل A	1	24
MOVX A,@Ri	نقل معلومة مشار إليها بالمسجل Ri (8بت) فى الذاكرة الخارجية إلى المسجل A	1	24
MOVX A,@DPTR	نقل معلومة مشار إليها بالمسجل DPTR فى الذاكرة الخارجية إلى المسجل A	1	24
MOVX @Ri,A	نقل محتويات المسجل A إلى عنوان مشار إليه بالمسجل Ri (8بت) فى الذاكرة الخارجية	1	24
MOVX @DPTR,A	نقل محتويات المسجل A إلى عنوان مشار إليه بالمسجل DPTR فى الذاكرة الخارجية	1	24
PUSH direct	دفع محتويات عنوان مباشر للمكدسة	2	24
POP direct	سحب من قمة المكدسة إلى عنوان مباشر	2	24
XCH A,Rn	استبدال محتويات المسجل A مع المسجل Rn	1	12
XCH A,direct	استبدال محتويات المسجل A مع عنوان مباشر	2	12
XCH A,@Ri	استبدال محتويات المسجل A مع عنوان غير مباشر	1	12
XCHD A,@Ri	استبدال محتويات أول 4 بت فى المسجل A مع نظراؤهم في عنوان غير مباشر	1	12
MOV C,bit	نقل بت محددة بعنوان مباشر إلى علم الحمل	2	12
MOV bit,C	نقل علم الحمل إلى بت محددة بعنوان مباشر	2	24

- من أمثلة التعامل على مستوى البت كما فى آخر أمرين فى جدول ٢-٥ ما يلى :

MOV C,P0.0
MOV P1.0,C

حيث الأمر الأول ينقل أول بت (بت رقم صفر) فى البوابة رقم صفر إلى علم الحمل ، والأمر الثانى ينقل علم الحمل إلى البت المناظرة فى البوابة رقم واحد . التعامل هنا يكون مع علم الحمل فقط . من أمثلة ذلك أيضا الأمر :

MOV C,03

الذى ينقل البت التى عنوانها 03 فى مجموعة المسجلات التى نتعامل معها على مستوى البايت (البايت رقم 20 حتى 2F) إلى علم الحمل . لاحظ أن البت 03 ستكون البت الرابعة فى البايت رقم 20 ، أنظر شكل (٢-٣) .

- التعامل مع الذاكرة الخارجية لا يكون إلا بالطريقة الغير مباشرة ومن خلال المسجل A فقط .
- الدفع أو السحب إلى/من المكدسة لا يكون إلا بالطريقة المباشرة ، أى أن المسجلات تستخدم بعناوينها وليس بأسمائها .
- أوامر الاستبدال تكون مع المسجل A وأماكن أو بايتات ذاكرة داخل المتحكم .
- أمر الاستبدال XCHD A, @Ri يستبدل النصف الأدنى فى المسجل A مع النصف الأدنى فى المسجل أو العنوان المشار إليه بالمسجل Ri ، بينما النصف الأعلى فى المسجلين لا يتأثر .

٢-١٠-٢ مجموعة أوامر المنطق Logic Instructions

- أوامر المنطق لا تؤثر على الأعلام على العكس من كل المعالجات ، إلا الأوامر التى تتعامل مع علم الحمل فهى تؤثر عليه على حسب طبيعة الأمر .
- التعامل بأوامر المنطق يكون على المسجلات (الذاكرة) الداخلية فقط سواء المسجلات العامة أو الخاصة .
- مصدر البيانات يكون بأى طريقة من طرق العنونة .
- الهدف الذى ستذهب إليه البيانات يكون المسجل A أو عنوان مباشر .
- وجود المعامل bit فى أى أمر يعنى أحد بتات المسجلات التى يمكن التعامل معها على مستوى البت من خلال عنوان مباشر لهذه البت ، أو من خلال اسم لها مثل البت P0.3 التى يقصد بها البت رقم 3 فى البوابة رقم صفر .

جدول ٢-٦

الأمر	وصف الأمر	بايتات	نبضات
CLR C	تصفير علم الحمل	1	12
CLR bit	تصفير بت بعنوان مباشر	2	12
SETB C	وضع واحد فى علم الحمل	1	12

SETB bit	وضع واحد فى بت بعنوان مباشر	2	12
CPL C	عكس علم الحمل	1	12
CPL bit	عكس بت بعنوان مباشر	2	12
ANL C,bit	آند لعلم الحمل مع بت بعنوان مباشر والنتيجة توضع فى علم الحمل	2	24
ANL C,/bit	آند لعلم الحمل مع معكوس بت بعنوان مباشر والنتيجة توضع فى علم الحمل	2	24
ORL C,bit	أور لعلم الحمل مع بت بعنوان مباشر والنتيجة توضع فى علم الحمل	2	24
ORL C,/bit	أور لعلم الحمل مع معكوس بت بعنوان مباشر والنتيجة توضع فى علم الحمل	2	24
ANL A,Rn	آند للمسجل A مع المسجل Rn والنتيجة فى المسجل A	1	12
ANL A,direct	آند للمسجل A مع بايت مباشرة والنتيجة فى المسجل A	2	12
ANL A,@Ri	آند للمسجل A مع بايت غير مباشرة والنتيجة فى المسجل A	1	12
ANL A,#data	آند للمسجل A مع معلومة فورية والنتيجة فى المسجل A	2	12
ANL direct,A	آند للمسجل A مع بايت مباشرة والنتيجة فى البايت المباشرة	2	12
ANL direct,#data	آند لمعلومة فورية مع بايت مباشرة والنتيجة فى البايت المباشرة	3	24
ORL A,Rn	أور للمسجل A مع المسجل Rn والنتيجة فى المسجل A	1	12
ORL A,direct	أور للمسجل A مع بايت مباشرة والنتيجة فى المسجل A	2	12
ORL A,@Ri	أور للمسجل A مع بايت غير مباشرة والنتيجة فى المسجل A	1	12
ORL A,#data	أور للمسجل A مع معلومة فورية والنتيجة فى المسجل A	2	12
ORL direct,A	أور للمسجل A مع بايت مباشرة والنتيجة فى البايت المباشرة	2	12
ORL direct,#data	أور لمعلومة فورية مع بايت مباشرة والنتيجة فى البايت المباشرة	3	24
XRL A,Rn	إكس أور للمسجل A مع المسجل Rn والنتيجة فى المسجل A	1	12
XRL A,direct	إكس أور للمسجل A مع بايت مباشرة والنتيجة فى المسجل A	2	12

XRL A,@Ri	إكس أور للمسجل A مع بايت غير مباشرة والنتيجة فى المسجل A	1	12
XRL A,#data	إكس أور للمسجل A مع معلومة فورية والنتيجة فى المسجل A	2	12
XRL direct,A	إكس أور للمسجل A مع بايت مباشرة والنتيجة فى البايت المباشرة	2	12
XRL direct,#data	إكس أور لمعلومة فورية مع بايت مباشرة والنتيجة فى البايت المباشرة	3	24
CLR A	تصفير المسجل A	1	12
CPL A	عكس المسجل A	1	12
RL A	دوران المسجل A ناحية الشمال	1	12
RLC A	دوران المسجل A ناحية الشمال من خلال علم الحمل	1	12
RR A	دوران المسجل A ناحية اليمين	1	12
RRC A	دوران المسجل A ناحية اليمين من خلال علم الحمل	1	12
SWAP A	استبدال نصفى المسجل A الأدنى والأعلى	1	12

٢-١٠-٣ مجموعة أوامر الحساب Arithmetic Instructions

- المسجل A طرف فى كل عمليات الحساب .
- النتيجة تذهب دائما للمسجل A .
- كل العمليات الحسابية تجرى بجميع طرق العنونة .
- التأثير على الأعلام يختلف من أمر لآخر

جدول ٢-٧

الأمر	وصف الأمر	بايتات	نبضات
ADD A,Rn	جمع محتويات مسجل Rn مع A والنتيجة فى A	1	12
ADD A,direct	جمع محتويات عنوان مباشر مع A والنتيجة فى A	2	12
ADD A,@Ri	جمع محتويات عنوان غير مباشر مع A والنتيجة فى A	1	12
ADD A,#data	جمع قيمة فورية مع A والنتيجة فى A	2	12
ADDC A,Rn	جمع محتويات مسجل Rn مع علم الحمل مع A والنتيجة فى A	1	12
ADDC A,direct	جمع محتويات عنوان مباشر مع علم الحمل مع A والنتيجة فى A	2	12
ADDC A,@Ri	جمع محتويات عنوان غير مباشر مع علم الحمل مع A والنتيجة فى A	1	12
ADDC A,#data	جمع قيمة فورية مع علم الحمل مع A والنتيجة فى A	2	12
SUBB A,Rn	طرح محتويات مسجل Rn مع علم الحمل من A	1	12

	والنتيجة فى A		
SUBB A,direct	طرح محتويات عنوان مباشر مع علم الحمل من A والنتيجة فى A	2	12
SUBB A,@Ri	طرح محتويات عنوان غير مباشر مع علم الحمل من A والنتيجة فى A	1	12
SUBB A,#data	طرح قيمة فورية مع علم الحمل من A والنتيجة فى A	2	12
INC A	زيادة واحد على محتويات المسجل A	1	12
INC Rn	زيادة واحد على محتويات مسجل Rn	1	12
INC direct	زيادة واحد على محتويات عنوان مباشر	1	12
INC @Ri	زيادة واحد على محتويات عنوان غير مباشر	1	12
DEC A	إنقاص واحد من محتويات المسجل A	1	12
DEC Rn	إنقاص واحد من محتويات مسجل Rn	1	12
DEC direct	إنقاص واحد من محتويات عنوان مباشر	1	12
DEC @Ri	إنقاص واحد من محتويات عنوان غير مباشر	1	12
INC DPTR	زيادة واحد على محتويات المسجل DPTR	1	24
MUL AB	ضرب محتويات A فى محتويات B والنتيجة فى A (النصف الأدنى) و B (النصف الأعلى)	1	48
DIV AB	قسمة A على B والنتيجة فى A والباقى فى B	1	48
DA A	ضبط محتويات المسجل A عشريا	1	12

٢-١٠-٤ مجموعة أوامر القفز Jump Instructions

جدول ٢-٧

الأمر	وصف الأمر	بايتات	نبضات
JC rel	قفز إلى عنوان نسبى عندما C=1	2	24
JNC rel	قفز إلى عنوان نسبى عندما C=0	2	24
JB bit,rel	قفز إلى عنوان نسبى عندما bit=1	3	24
JNB bit,rel	قفز إلى عنوان نسبى عندما bit=0	3	24
JBC bit,rel	قفز إلى عنوان نسبى عندما bit=1 وبعدها يجعل C=0	3	24
ACALL addr11	نداء مطلق على برنامج فرعى عند عنوان من 11 بت	2	24
LCALL addr16	نداء بعيد على برنامج فرعى عند عنوان من 16 بت	3	24
RET	عودة من برنامج فرعى	1	24
RETI	عودة من برنامج خدمة مقاطعة	1	24
AJMP addr11	قفز مطلق إلى عنوان من 11 بت	2	24
LJMP addr16	قفز إلى عنوان بعيد من 16 بت	3	24
SJMP rel	قفز نسبى إلى عنوان قصير	2	24
JMP @A+DPTR	قفز غير مباشر نسبى للمسجلين A+DPTR	1	24
JZ rel	قفز نسبى إذا كان المسجل A=0	2	24

JNZ rel	قفز نسبي إذا كان المسجل لا يساوى الصفر	2	24
CJNE A, direct,rel	قارن المسجل A مع عنوان مباشر ويقز نسبيا إذا لم يكن هناك تساوى بينهما	3	24
CJNE A, #data,rel	قارن المسجل A مع معلومة فورية ويقز نسبيا إذا لم يكن هناك تساوى بينهما	3	24
CJNE RN, #data,rel	قارن المسجل Rn مع معلومة فورية ويقز نسبيا إذا لم يكن هناك تساوى بينهما	3	24
CJNE @Ri, #data,rel	قارن العنوان الغير مباشر @Ri مع معلومة فورية ويقز نسبيا إذا لم يكن هناك تساوى بينهما	3	24
DJNZ Rn,rel	انقص المسجل Rn بمقدار واحد واقفز نسبيا طالما أنه لا يساوى الصفر	2	24
DJNZ direct,rel	انقص عنوان مباشر بمقدار واحد واقفز نسبيا طالما أنه لا يساوى الصفر	3	24
NOP	لا تعمل شيء	1	12

٢-١١ إدخال وإخراج البيانات Input Output Data

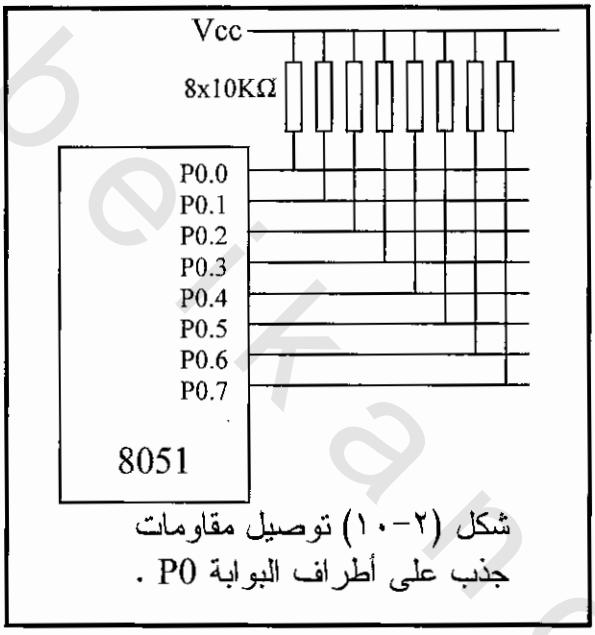
كما نعلم فإن المتحكم 8051 به أربع بوابات يمكن برمجة كل منها لتعمل كبوابة إدخال أو إخراج للبيانات ، وكل واحدة من هذه البوابات تتكون من ثمان بتات أو ثمانية أطراف يمكن التعامل مع كل طرف فيها على حده بحيث تكون بعض الأطراف داخل نفس البوابة تعمل كخطوط إدخال والباقى تعمل كخطوط إخراج . كل طرف من هذه الأطراف يقابله بداخل المتحكم ماسك D flip flop ، وكل 8 من هذه الماسكات تكون أحد البوابات المعروفة بعنوان معين . فمثلا البوابة P0 عنوانها هو 80H وهذا العنوان هو عنوان الثمانية ماسكات الخاصة بهذه البوابة . تذكر جيدا أن هذه الماسكات تختلف تماما عن أطراف الشريحة ، وأن البيانات الموجودة على أطراف الشريحة ليس بالضرورة أن تكون هى الموجودة فى هذه الماسكات . كما نعلم أيضا أن هذه البوابات ما عدا البوابة P1 لها وظائف أخرى تبادلية مع عملية الإدخال والإخراج ، وفى حالة عملها فى هذه الوظائف البديلة فإنه لا يمكن استخدامها فى عمليات الإدخال والإخراج . عند استخدام أحد هذه البوابات كبوابة إدخال لابد من كتابة واحد فى ماسك هذه البوابة أو الطرف المراد استخدامه كطرف إدخال . أما عند استخدام أى طرف كطرف إخراج فإن البيانات المراد إخراجها يتم كتابتها مباشرة فى ماسك هذا الطرف .

٢-١١-١ البوابة P0

أطراف هذه البوابة تستخدم كخطوط بيانات وعناوين فى حالة التعامل مع الذاكرة الخارجية . عند استخدام أطراف هذه البوابة فى إدخال وإخراج البيانات لابد من

توصيل مقاومة جذب pull up resistance قيمتها 10K أوم على جميع أطراف هذه البوابة كما فى شكل (٢-١٠) .

شكل (٢-١١) يبين التركيب الداخلى لأحد أطراف البوابة P0 وقد تم توصيل مقاومة جذب خارجيا على هذا الطرف كما ذكرنا . كل أطراف هذه البوابة تتصل بطرف الخرج من خلال اثنين ترانزستور كما فى الشكل . عند تسجيل واحد فى ماسك الطرف فإن دائرة التحكم الموجودة فى الشكل تخرج صفر على قاعدة كل واحد من الترانزستورين فتجعل كل منهما لا يعمل off وبالتالى يصبح طرف الخرج منفصل buffered تماما عن خرج الدائرة . فى هذه الحالة يمكن قراءة طرف الخرج عن طريق تنشيط ماسك الدخل I عن طريق البرنامج أو أمر القراءة حيث يتم إدخال الإشارة الموجودة على الطرف من خلال هذا الماسك إلى مسار البيانات . البرنامج التالى يقرأ البوابة P0 ، بعد تسجيل وحايد فى كل ماسكاتها ، ويخرج محتوياتها على البوابة P1 .



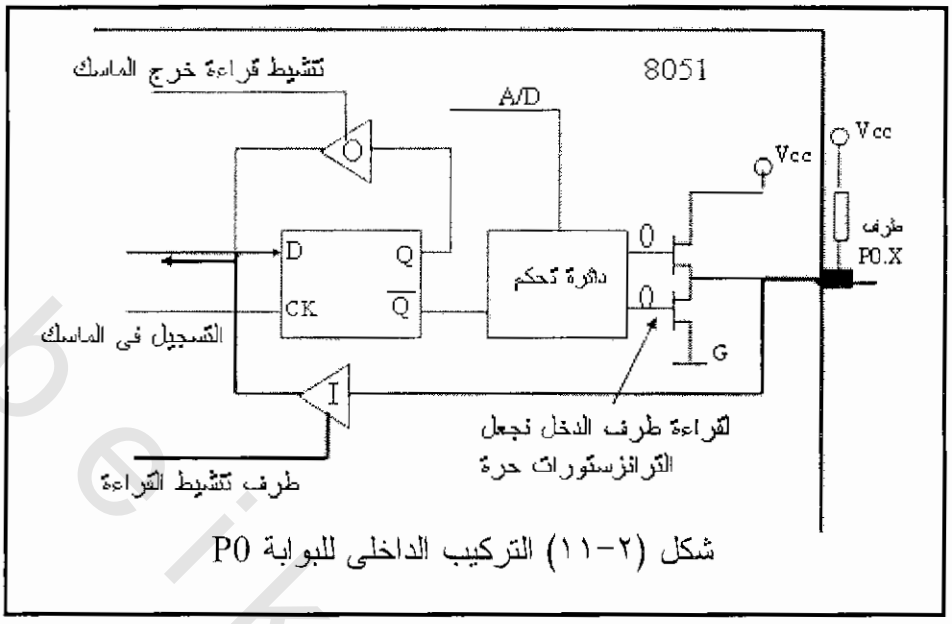
شكل (٢-١٠) توصيل مقاومات جذب على أطراف البوابة P0 .

```
        MOV A,#FFH
        MOV P0,A
Back:   MOV A,P0
        MOV A,P1
        SJMP Back
```

عند بداية تشغيل المتحكم reset فإن كل البوابات تكون بوابات إخراج . بعد ذلك يمكن جعلها بوابات إدخال أو استخدامها كبوابات إخراج مباشرة . البرنامج التالى يستخدم نفس البوابة ، P0 ، فى إنارة وإطفاء لمبات مركبة على هذه الأطراف باستخدام برنامج فرعى اسمه delay للحصول على زمن تأخير .

```
        MOV A, #FFH
Back:   MOV P0,A
        ACALL delay
        CPL A
        SJMP Back
```

obeikandi.com

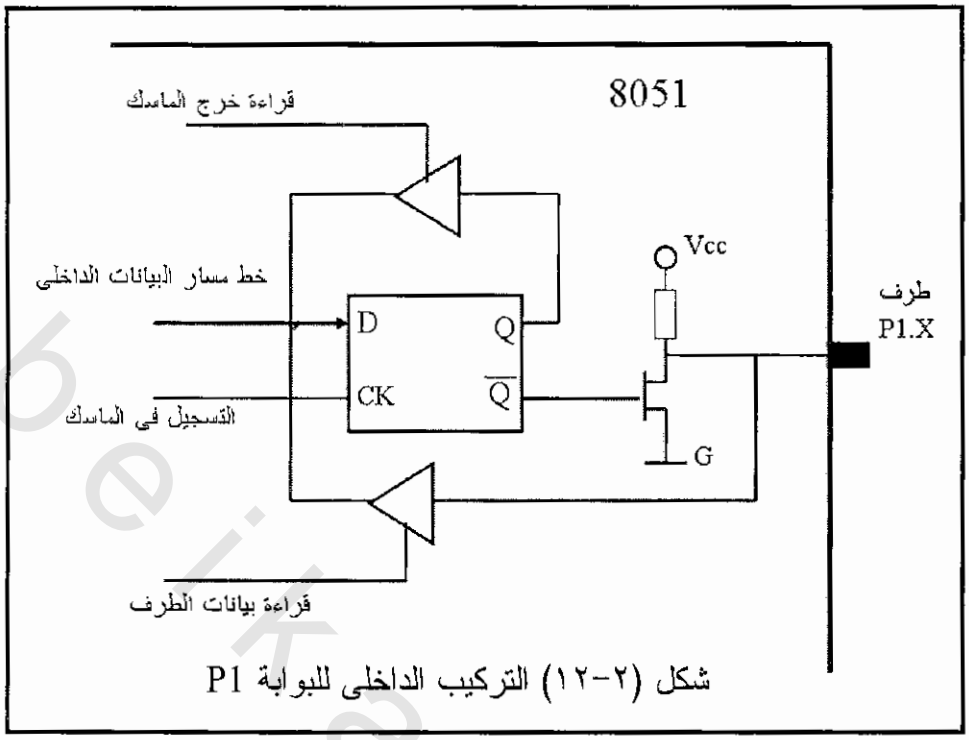


شكل (٢-١١) التركيب الداخلى للبوابة P0

عند استخدام أى طرف كطرف إخراج وتسجيل واحد فى ماسك هذا الطرف فإن هذا الواحد سيجعل ترانزستورات الخرج لا تعمل off وبالتالى يصبح الطرف الخارجى متصلا بالجهد Vcc من خلال مقاومة الجذب الموصلة على هذا الطرف . فى هذه الحالة يكون عازل الدخل I فاصلا لأنه لا يتم تنفيذ أمر قراءة فى هذه الحالة . أى أنه لإخراج واحد على أى طرف يكفى تسجيل هذا الواحد فى ماسك هذا الطرف .

عند تسجيل صفر فى ماسك أى طرف فإن هذا الصفر يجعل $\overline{Q}=1$ وبالتالى قاعدة الترانزستور الأسفل يكون عليها واحد وبالتالى يكون الترانزستور الأسفل نشط on مما يجعل طرف الخرج متصلا بالأرضى ، أى صفر من خلال هذا الترانزستور . الخلاصة من ذلك هى أنه عند استخدام أى طرف من أطراف هذه البوابة كطرف دخل ، نكتب واحد فى ماسك هذا الطرف ثم نقرأه ، وعند استخدام هذا الطرف كطرف خرج نكتب البيانات المراد إخراجها مباشرة فى ماسك الخرج .

باقى البوابات P1 و P2 و P3 كلها يمكن استخدامها كبوابات إدخال أو إخراج مباشرة دون الحاجة لتوصيل مقاومات جذب خارجية كما فعلنا مع البوابة P0 لأن جميع أطراف هذه البوابات موصل عليها داخليا مقاومات جذب كما فى شكل (٢-١٢) . فقط عند استخدام أى خط من خطوط هذه البوابات كطرف دخل نكتب واحد فى ماسك هذه الأطراف ثم نقرأها . وعند استخدامها كطرف خرج نكتب البيانات المراد إخراجها مباشرة كما فعلنا مع البوابة P0 .



شكل (٢-١٢) التركيب الداخلى للبوابة P1

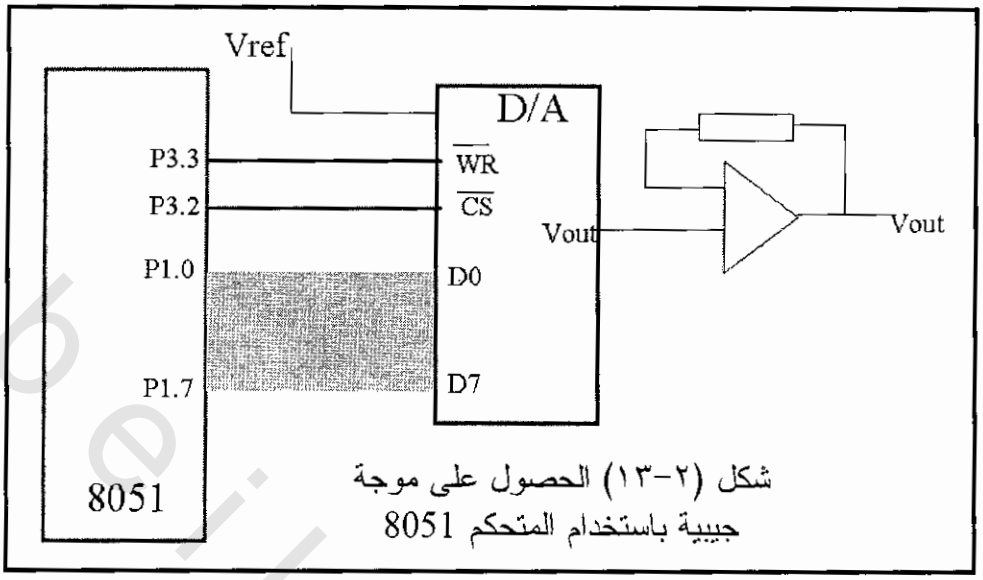
مثال ٢-١ :

مطلوب استخدام المتحكم 8051 فى توليد موجة جيبية sine wave يمكن التحكم فى ترددها . سيتم ذلك عن طريق تخزين قيم متوالية للموجة الجيبية تمثل دورة كاملة من دورات الموجة فى الذاكرة ثم النداء على هذه القيم بالتتابع وإخراجها على بوابة إخراج . بوابة الإخراج تتصل بمحول رقمى/انسيابى كما فى شكل (٢-١٣) ليقوم بتحويل هذه القيم الرقمية إلى قيم انسيابية .

القيم المخزنة فى الذاكرة تسمى جدول بحث look up table . يمكن حساب قيم هذا الجدول عن طريق تقسيم دورة كاملة من الموجة الجيبية (360 درجة) إلى أى عدد من الأقسام وليكن 150 جزء حيث ستكون قيمة كل جزء تساوى 360/150=2.4 درجة . نحسب قية الموجة الجيبية عند كل جزء من هذه الأجزاء ابتداء من الصفر n=0 وانتهاء عند القيمة n=150 كما هو موضح فى شكل (٢-١٤) وباستخدام العادلة التالية :

$$X=80+7F\cos(n*2.4), \quad n=0, 1, 2, \ldots 150$$

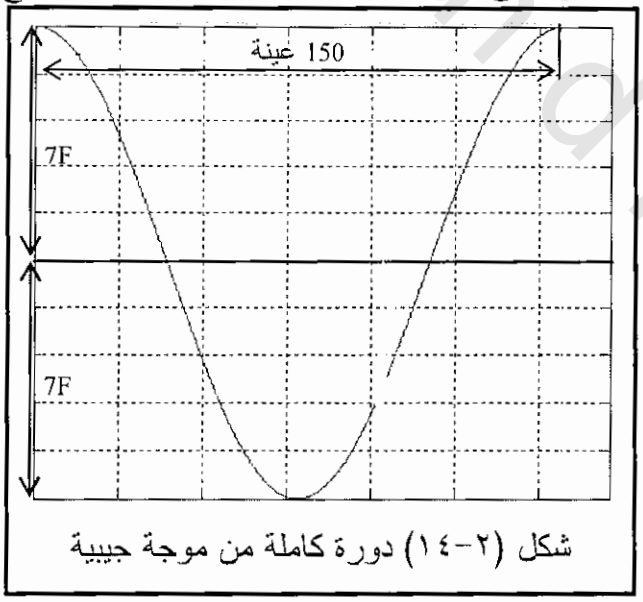
حيث X هى قيمة جيب تمام الموجة عند كل نقطة أو كل جزء من ال 150 جزء . لاحظ أننا فى المعادلة السابقة اعتبرنا أن قيم موجة جيب التمام تتراوح بين الصفر و FFH . بالتعويض فى المعادلة السابقة عن قيم n المختلفة نحصل على قيم X كالتالى:



شكل (٢-١٣) الحصول على موجة جيبية باستخدام المتحكم 8051

FF, FF, FF, FE, FD, FC,FB, ...FE, FF, FF

هذه القيم سيتم تخزينها فى الذاكرة ابتداء من عنوان سنسميه أو نشير إليه بالإسم table كما فى شكل (٢-١٥) الذى يوضح البرنامج بلغة تجميع المتحكم 8051 وهذا الجدول . بزيادة عدد العينات المأخوذة فى دورة الموجة الجيبية فإن موجة الخرج الناتجة ستكون أكثر دقة ونعومة . البرنامج الموضح فى شكل (٢-١٥) يقوم باختصار بقراءة قيم الجدول table على التتابع وإخراجها على بوابة الإخراج . فى هذا البرنامج تم وضع زمن تنفيذ كل أمر بين قوسين بحيث يمكن جمع كل هذه الأزمنة وحساب الزمن الفعلى الذى سيأخذه البرنامج بالكامل حيث سيمثل هذا الزمن أقل زمن دورى للموجة وبالتالى أكبر تردد لها . يمكن زيادة الزمن الدورى وبالتالى تقليل التردد عن طريق إضافة أزمنة تأخير فى البرنامج .



شكل (٢-١٤) دورة كاملة من موجة جيبية

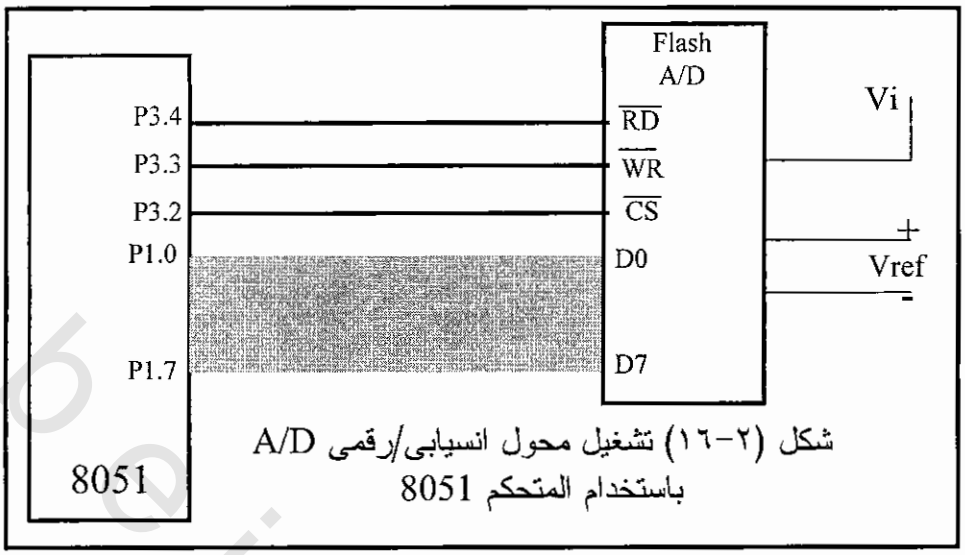
```
        ORIG 0000                                          Table: .db 00h
Start:  clr P3.2      تنشيط الطرف CS للمحول                      .db ffh
        mov dptr,table  الإشارة لبداية الجدول                    .db feh
Repeat:mov r1,#96H  (1) r1=150d عدد العينات فى الدورة            .db feh
Next:   mov a,r1 (1)                                             .db fdh
        movc a,@a+dptr  (2)                                      .....
        mov p1,a  (1)  P1 إرسال العينة إلى البوابة               .db 00h
        clr p3.3  (1)   WR=0 تنشيط الخط                         .....
        setb p3.3 (1)   WR=1                                     .db fdh
        djnz r1,next  (2)                                        .db feh
        sjmp repeat  (2)                                         .db feh
                                                                 .end
```

شكل (٢-١٥) برنامج توليد موجة جيبية باستخدام المتحكم 8051

فى شكل (٢-١٥) هناك بعض الأوامر الخاصة بالمجمع assembler مثل الأمر ORIG 0000 والذى يعنى بدأ البرنامج (أو الأوامر التالية) من العنوان 0000H ، وكذلك الأمر db nn والذى يعنى حجز بايت define byte فى الجدول table وتحميلها بالقيمة الستعشرية nnH . ننصح هنا بمراجعة أوامر المجمع فى أى كتاب خاص بلغة التجميع .

مثال ٢-٢ :

فى هذا المثال سنقوم بتشغيل محول انسيابى/رقمى A/D لتحويل موجة انسيابية Analog إلى صورتها الرقمية وقراءة ١٠٠٠ قراءة أو الف قيمة منها وتخزينها فى الذاكرة ابتداء من العنوان 4000H حتى العنوان 43E7H . شكل (٢-١٦) يوضح الدائرة المقترحة لذلك حيث استخدمت البوابة P1 كبوابة دخل نقرأ من خلالها خرج المحول . المحول المفترض من النوع الفورى flash مما يعنى أن زمن التحويل له صفر تقريبا . لذلك فإننا بمجرد تنشيط الشريحة عن طريق جعل الطرف $\overline{CS} = 0$ ، ثم بدأ عملية التحويل عن طريق جعل الطرف $\overline{WR} = 0$ فإننا نقرأ خرج المحول فورا عن طريق جعل الطرف $\overline{RD} = 0$ ولا نكون مضطرين للانتظار حتى ينتهى المحول من عملية التحويل . تتم عملية التحويل عن طريق تنشيط الطرف $\overline{CS} = 0$ من خلال الطرف P3.2 وبعدها يتم تنشيط الطرف $\overline{WR} = 0$ لبدأ عملية التحويل من خلال الطرف P3.3 . بما أن المحول من النوع الفورى فإنه يمكن قراءة خرج المحول فورا عن طريق تنشيط الطرف $\overline{RD} = 0$ من خلال الطرف P3.4 كما هو موضح فى شكل (٢-١٦) .



شكل (٢-١٦) تشغيل محول انسيابى/رقمى A/D
باستخدام المتحكم 8051

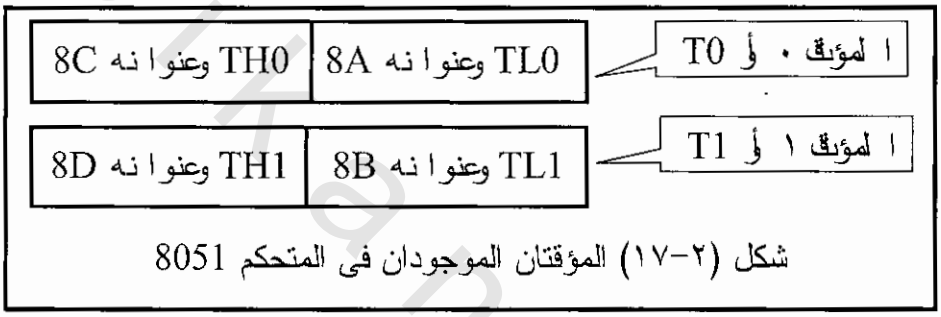
البرنامج التالى سيقوم بهذه العملية حيث سيتم تخزين البيانات ابتداء مـن العنـوان 4000H حتى العنوان 43E7H . شفرات البرنامج تبدأ من العنـوان 0000H وذلـك باستخدام الأمر org 0000 .

```
        .equ begin 4000h
        .equ delay 74h  زمن تأخير بين كل عينة والتالية
        .equ end1,43h
        equ end2,e8h
        .org 0000
Start:  mov dptr,#begin
        clr p3.2    CS=0
Next:   clr p3.3    WR=0  بدأ التحويل
        setb p3.3   WR=1
        clr p3.4    RD=0  تنشيط القراء
        mov a,p1    قراءة خرج المحول
        setb p3.4   RD=1  إنهاء عملية القراءة
        movx @dptr,a    تخزين القراءة
        inc dptr
        mov a,dph
        cjne a,#end1,wait  مقارنة بالبايت الصغرى من عنوان نهاية البيانات
        mov a,dpl
        cjne a,#end2,wait  مقارنة بالبايت الكبرى من عنوان نهاية البيانات
        sjmp done
Wait:   mov r1,delay
Here:   djnz r1,here
```

```
        sjmp next
Done:   sjmp done
        .end
```

٢-١٢ المؤقتات والعدادات Timers and Counters

يحتوى المتحكم 8051 على مؤقتين أو عدادين حيث المؤقت والعداد شيء واحد . عند استخدامه كمؤقت يتم تحميله بقيمة معينة تكافيء الزمن المطلوب ويتم إنقاص هذه القيمة حتى تصل إلى الصفر عند نهاية زمن التأخير المطلوب كما سنرى . عند استخدامه كعداد يتم توصيله بنبضات خارجية يتم عدها كما سنرى أيضا . كل مؤقت/عداد يتكون من ١٦ بت (٢ بايت) يتم التعامل مع كل بايت فى كل منها من خلال عنوان منفصل كما فى شكل (٢-١٧) الذى يبين عنوان كل بايت فى كل مؤقت .



شكل (٢-١٧) المؤقتان الموجودان فى المتحكم 8051

يستخدم المسجلان TCON الذى عنوانه 88H و TMOD الذى عنوانه 89H فى التحكم فى أداء المتحكم كالتالى :

٢-١٢-١ المسجل Timer CONtrol, TCON

يتكون هذا المسجل من ٨ بت كما فى شكل (٢-١٨) وكل بت تمثل حالة معينة من حالات المتحكم كما يلى :

١- **العلمان TF0 ,TF1** وهما علما فيضان لكل من المؤقتين حيث يصبح كل منهما يساوى واحد عندما يصل المؤقت إلى نهايته FFFFH حيث بعدها يصبح صفرا فى النبضة التالية .

٢- **بتات بدأ المؤقت Timer Run, TR1, TR0** تستخدم كل واحدة من هذه البتات فى تشغيل أو توقيف المؤقت المقابل لها عن طريق وضعها بواحد أو صفر عن طريق المستخدم . عند وضع أى بت بصفر يتوقف المؤقت المقابل لها عن العد أو العمل ولكن لا يتم تصفير المؤقت .

٣- **علما المقاطعة الخارجية External Interrupt flags, IE1, IE0** يحتوى المتحكم 8051 على طرفين للمقاطعة الخارجية وهذين الطرفين مشتركين

وظيفيا مع أطراف البوابة $P3.4 = \overline{INT1}$ و $P3.5 = \overline{INT0}$. عند حدوث أى مقاطعة خارجية على واحد من هذين الطرفين الخارجيين P3.4 أو P3.5 فإن العلم المقابل له IE1 أو IE0 يكون بواحد . لاحظ أن هذين العلمين لا علاقة لهما بالمؤقتات .

٤- **بتات تحديد كيفية المقاطعة IT1, IT0** تستخدم هذه البتات لجعل خطوط المقاطعة الخارجية $P3.4 = \overline{INT1}$ أو $P3.5 = \overline{INT0}$ تكون حساسة للحافة النازلة (واحد للصفر) عند وضع IT1=1 **أو** IT0=1 وعند وضع IT1=0 **أو** IT0=0 فإن طرف المقاطعة المقابل يكون حساس للمستوى صفر وليس لأى حافة .

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TCON
وعنوانه 88

شكل (٢-١٨) مسجل التحكم فى المؤقتات TCON

٢-١٢-٢ المسجل Timer MOD, TMOD تستخدم بتات هذا المسجل فى اختيار طريقة عمل المؤقت . يتكون هذا المسجل من ٨ بتات أربعة لكل مؤقت كما هو موضح فى شكل (٢-١٩) . وظيفة بتات هذا المسجل سنعرضها كالتالى :

7	6	5	4	3	2	1	0
Gate	C/T	M1	M0	Gate	C/T	M1	M0
مؤقت 1				مؤقت 0			

TMOD
وعنوانه 89

شكل (٢-١٩) مسجل اختيار طريقة عمل المؤقتات TMOD

١- **البت Gate** توضع بصفر لبدأ تشغيل المؤقت بشرط أن تكون البت المقابلة TR1 أو TR0 فى المسجل TCON بواحد .

٢- **البت** $C/\overline{T}$ بوضع هذه البت بواحد فإن المؤقت يعمل كعداد يعد النبضات الداخلة على الطرف P3.5 فى حالة المؤقت واحد والطرف P3.4 فى حالة المؤقت صفر . بوضع هذه البت بصفر فإن المؤقت يعمل كمؤقت وليس كعداد ، لاحظ شرطة النفى فوق الحرف T .

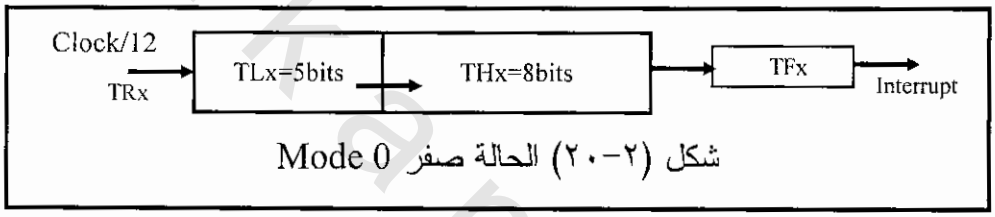
جدول ٢-٨

M1	M0	الحالة Mode
0	0	صفر 0
0	1	واحد 1
1	0	اثنان 2

٣- **البتات M1, M0** توضع كل منها واحد أو صفر عن طريق المبرمج لاختيار طريقة عمل المؤقت ، ويمكن اختيار واحدة من أربع حالات modes كما فى جدول ٢-٨ .

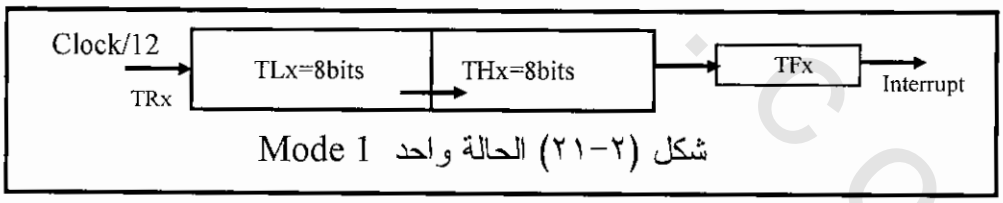
تفصيل هذه الحالات كالتالى :

- **الحالة صفر mode 0** وفيها يتكون المؤقت المختار من ١٣ بت ، ثمانية منها تمثل البايت العليا من المؤقت THx وخمسة من البايت الصغرى TLx . أى أن THx سيستقبل نبضة من TLx كل ٣٢ دورة أمر ، وعندما يصبح THx كله وحايد فإن العلم TF (علم المقاطعة أو علم الفيضان) فى المسجل Tx يصبح واحد . تذكر هنا أن المؤقت يعمل على نبضات داخلية وكل نبضة تمثل دورة أمر التى تتكون من ١٢ نبضة من نبضات الساعة (نبضات المذبذب oscillator) . شكل (٢-٢٠) يبين رسما توضيحيا لهذه الحالة .



شكل (٢-٢٠) الحالة صفر Mode 0

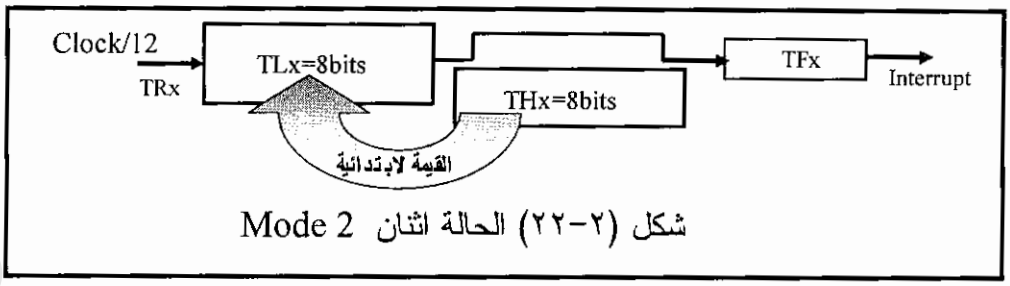
- **الحالة واحد Mode 1** مثل الحالة السابقة تماما سوى أن المؤقت المختار يتكون من ١٦ بت (البايت العليا والبايت الصغرى معا) . العلم TF=1 عندما تكون كل بتات المسجلين THx و TLx وحايد . شكل (٢-٢١) يبين رسما توضيحيا لهذه الحالة .



شكل (٢-٢١) الحالة واحد Mode 1

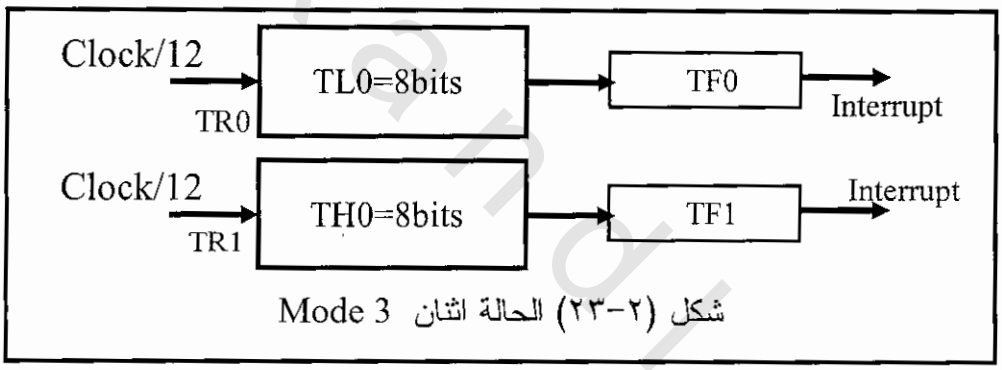
- **الحالة اثنان mode 2** فى هذه الحالة يتكون المؤقت من ٨ بت فقط هى البايت الصغرى TLx ولكن الجديد هنا أنه يتم تحميل هذه البايت TLx بقيمة ابتدائية يبدأ بها المؤقت عملية العد من البايت العليا THx . يصبح العلم TF=1 عندما تصبح البايت TLx كلها وحايد . شكل (٢-٢٢) يبين رسما توضيحيا لهذه الحالة .

obeikandi.com



شكل (٢-٢٢) الحالة اثنان Mode 2

- الحالة ثلاثة mode 3 المؤقت T0 فقط يمكنه أن يعمل فى الحالة الثالثة ، فى هذه الأثناء يمكن أن يكون T1 فى أى حالة ما عدا الحالة الثالثة . عندما يكون T0 فى الحالة الثالثة فإنه يعمل كمؤقتان منفصلان كل منهما ٨ بت ، الأول هو TL0 ويتحكم فيه TF0 و TR0 ، الثانى هو TH0 ويتحكم فيه TF1 و TR1 الخاصين بالمؤقت T1 . إذا استخدم T1 فى هذه الأثناء فى أى حالة أخرى فإنه يعمل دون الاعتماد على TF1 أو TR1 ، أو يعمل كعداد . شكل (٢-٢٣) يبين رسما توضيحيا لهذه الحالة .



شكل (٢-٢٣) الحالة اثنان Mode 3

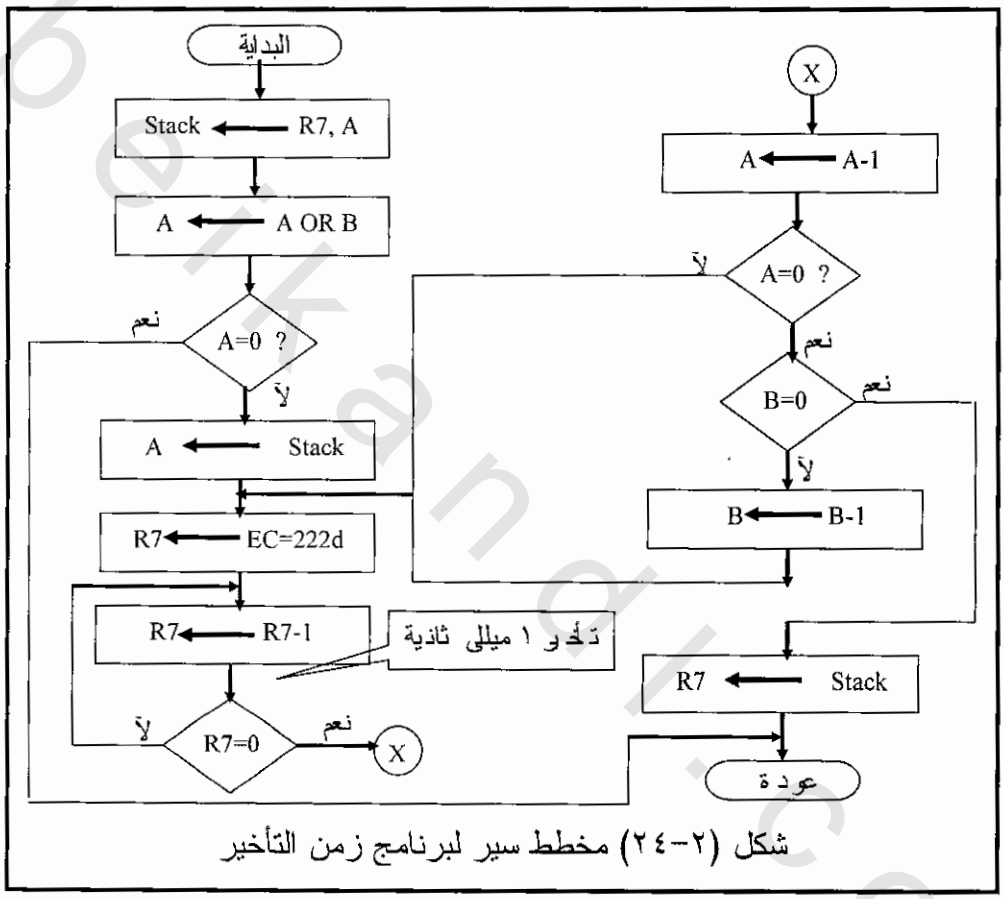
الفرق بين العداد والمؤقت هو مصدر النبضات ، فى المؤقت يكون المصدر هو نبضات الساعة الداخلية بعد قسمتها على ١٢ ، فى حالة العداد يتم إدخال النبضات من الخارج على الطرف P3.4 فى حالة T0 ، والطرف P3.5 فى حالة T1 . الطرف $C/\overline{T}$ يجب أن يكون ١ حتى تعمل المؤقتات كعدادات .

يقوم المتحكم بقراءة أطراف الدخل فى حالة العدادات مرة كل دورة ماكينة Machine cycle (١٢ نبضة ساعة) ، لذلك لابد أن يكون تردد النبضات على أطراف العد أقل من clock/24 وإلا فإن عملية العد لن تكون دقيقة .

مثال ٢-٣ :

برنامج فرعى عند النداء عليه يعطى زمن تأخير بقدر العدد الموجود فى المسجلين A و B ، حيث الوحدة هى الميللي ثانية . لذلك يبدأ البرنامج بتحميل المسجل R7

بالقيمة ECh=222d التى تعطى تأخير مقداره ١ ميللى ثانية مع فرض أن ساعة المتحكم ترددها هو ١٦ ميجاهرتز . أكبر زمن تأخير سيكون ٦٥٥٣٦ ميللى ثانية وهو أكبر رقم يمكن تخزينه فى المسجلين A و B . تذكر أن دورة الأمر تتم فى ١٢ نبضة ساعة ، بمعرفة عدد الدورات التى يأخذها كل أمر نستطيع أن نعرف زمن تنفيذه . سننفذ البرنامج مرة دون استخدام المؤقتات ومرة باستخدام المؤقتات كتدريب عليها . شكل (٢-٢٤) يبين مخطط سير للبرنامج يعقبه البرنامج باستخدام أوامر المتحكم 8051 .



شكل (٢-٢٤) مخطط سير لبرنامج زمن التأخير

كما نرى فى مخطط السير فإن البرنامج يبدأ بدفع المركم A والمسجل R7 فى المكدسة stack لحفظ أى بيانات ضرورية قد تكون هناك حاجة إليها على حسب التطبيق المقترح . العملية المنطقية OR على محتويات المسجلين A و B لنرى إذا كان كل منهما يساوى صفرا أم لا ، فإذا كان كل من المسجلين يحتوى صفر ففى هذه الحالة يتم وقف البرنامج . إذا كانت نتيجة ال OR لا تساوى صفر يتم استرجاع محتويات المسجل A من المكدسة (لأنها قد تغيرت بسبب إجراء عملية ال OR) حيث يبدأ إجراء زمن التأخير . يبدأ برنامج زمن التأخير بتحميل المسجل R7

بالقيمة العشرية 222 التى يتم إنقاصها واحد بواحد فى حلقة حتى تصل إلى الصفر حيث عندها يكون قد مر واحد ميللى ثانية على حسب نبضات الساعة التى تم اعتبارها ١٦ ميجاهرتز فى هذا البرنامج . بعد وصول R7 إلى الصفر يتم إنقاص المسجل A بمقدار واحد ثم نرجع لتحميل R7 مرة ثانية لنبدأ تأخير واحد ميللى ثانية مرة ثانية . يستمر البرنامج فى ذلك إلى أن يصبح المسجل A=0 . إذا وصل المسجل A إلى الصفر يتم اختبار المسجل B هل وصل للصفر أم لا ؟ إذا لم يصل B إلى الصفر يتم العودة للتأخير واحد ميللى ثانية ، ثم إنقاص A بواحد والذى يكافىء تحميله مرة ثانية بالقيمة FFH حيث 00H-01H=FFH وبعدها يستمر البرنامج حتى تصل A إلى الصفر مرة ثانية . تذكر أن A تحمل النصف الأعلى من الرقم و B تحمل النصف الأدنى منه . لذلك إذا بدأ البرنامج وكان A=FFH و B=FFH فإن زمن التأخير الناتج سيكون FFFFH=65536d . ينتهى البرنامج عندما يكون كل من A=B=0 حيث عندها يتم إرجاع قيمة المسجل R7 من المكدسة والعودة للبرنامج الأساسى . أوامر هذا البرنامج بدون استخدام المؤقتات ستكون كالتالى :

```
          .equ delay 0ech
Softime:  .org 0000h
          push o7
          push acc
          orl a,b
          cjne a,#00h,OK
          pop acc
          sjmp Done
OK:       pop acc
Timer:    mov R7,#delay
Mill:     nop
          djne R7,Mill
          nop
          djne acc,Timer
          cjne a,b,Bdown
          sjmp done
Bdown:    dec b
          sjmp Timer
Done:     pop 07h
          ret
          .end
```

البرنامج التالى هو نفس البرنامج السابق ولكن هذه المرة باستخدام المؤقتات . هنا تم تحميل المؤقت بالقيمة CBFAH التى تكافىء المتمم الثنائى للقيمة 1333d=0535H والتى تعطى واحد ميللى ثانية عند تردد ساعة مقداره ٦ ميجاهرتز هذه المرة . البرنامج هذه المرة سيكون كما يلى فحاول تتبعه :

Obeikandl.com

```
            .equ msech ,0fah
            .equ msecl,0cbh
            .org 0000h
Timer:      push tl0
            push th0
            cjne a,#00h,Go        is a=0, b=0?
            orl a,b
            jz done
            clr a
Go:         anl tcon,#0cfh        TF0=0, TR0=0
            anl tmod,#0f0h        Timer0 for Time, mode1
            orl tmod,#01h
Onem:       mov tl0,#msecl
            mov th0,#msech
            orl tcon,#10h
Wait:       jbc tf0,Dwnab
            sjmp wait
Dwnab:      anl tcon,#0efh
            djnz acc,Onem
            cjne a,b,Bdown
            sjmp done
Bdown:      dec b
            Sjmp Onem
Done:       pop th0
            pop tl0
            ret
            .end
```

١٣-٢ المقاطعة Interrupts

كما نعلم فإنه يمكن مقاطعة المتحكم فى أى لحظة إما من خطوط مقاطعة خارجية ، أو داخليا عندما يصل مؤقت إلى قيمته العظمى مثلا أو ينتهى المتحكم من إرسال أو استقبال بايت تتابعية كما سنرى عند دراسة التراسل التتابعى . عندما يستقبل المتحكم أى مقاطعة فإنه يذهب إلى مكان محدد فى الذاكرة يسمى متجه المقاطعة interrupt vector والذى يحتوى برنامج الخدمة المطلوب لهذه المقاطعة Interrupt Service Routine, ISR وكل المقاطعات الأخرى . إذا كان برنامج الخدمة قصيرا فإنه يكتب فى نفس متجه القاطعة ، أما إذا كان برنامج الخدمة كبيرا فإن يتم كتابة أمر قفز فقط فى متجه المقاطعة لبداية هذا البرنامج فى الذاكرة . عندما ينتهى المتحكم من تنفيذ برنامج خدمة المقاطعة يعود إلى نفس المكان الذى جاء منه قبل

حدوث طلب القاطعة . عند حدوث أى طلب للمقاطعة فإن المتحكم يتبع الخطـوات التالية :

١- ينتهى المتحكم من تنفيذ الأمر الحالى (الذى كان ينفذه وقت حدوث المقاطعة) ، ثم يدفع عنوان الأمر الذى عليه الدور فى التنفيذ (محتويات عداد البرنامج) فى المكدسة stack .

٢- يقفز المتحكم إلى عنوان محدد فى ذاكرة البرمجة ROM يسمى متجه المقاطعة كما فى جدول ٢-٩ . متجه المقاطعة يحتوى عنوان محدد لكل مقاطعـة يـتم القفز إليه من قبل المتحكم عند حدوث أى واحدة من الستة مقاطعات الممكنة فى المتحكم 8051 .

جدول ٢-٩

مصدر المقاطعة	عنوان القفز فى ذاكرة البرمجة ROM	طرف المقاطعة
الطرف Reset	0000H	9
المقاطعة الخارجية INT0	0003H	P3.2(12)
المقاطعة من المؤقت صفر TF0	000BH	
المقاطعة الخارجية INT1	0013H	P3.3(13)
المقاطعة من المؤقت واحد TF1	001BH	
المقاطعة من البيانات التتابعية R1, T1	0023H	

٣- العناوين الموجودة فى متجه المقاطعة متتابعة كما فى جدول ٢-٩ والمسافة بين كل عنوان والتالى له تكون عبارة عن عدد قليل جدا من البايتات التى لا تكفى لكتابة برنامج خدمة مقاطعة كامل للكثير من هذه المقاطعات . لذلك فإنـه فـى الغالب يتم كتابة شفرة أمر قفز واحد فقط لينقل عملية التنفيذ إلى حيـث يوجـد برنامج خدمة المقاطعة الكامل فى الذاكرة . برنامج الخدمة لأى مقاطعة لابد أن ينتهى بالأمر RETI والذى بتنفيذه يعود المتحكم إلى العنوان الذى جاء منه عند طلب القاطعة . كمثال على ذلك إفترض أننا أعطينا نبضة على الطرف Reset للمتحكم فإنه مباشرة يتوجه إلى العنوان 0000H فى متجـه المقاطعـة . عنـد العنوان 0000H نسجل شفرة الأمر LJMP add حيث add هو عنـوان خدمـة المقاطعة Reset والذى يحتوى البرنامج المطلوب تنفيـذه عنـد حـدوث هـذه المقاطعة .

٤- بعد تنفيذ الأمر RETI فإن المتحكم يسحب POP العنوان الموجـود فـى قمـة المكدسة ويضعه فى عداد البرنامج وبذلك يعود إلى حيث انتهى عنـد حـدوث المقاطعة .

كل هذه المقاطعات ما عدا المقاطعة Reset يمكن إخمادها أى منعها من التأثير أو تنشيطها وذلك من خلال مسجلين خاصين بالتحكم فى عملية المقاطعة كما سنرى الآن بالتفصيل . الكثير من المستخدمين لا ينظر للطرف Reset على أنه طرف مقاطعة لأنه لا يتم التحكم فى أداؤه من حيث الإخماد أو التنشيط من خلال مسجلات التحكم فى المقاطعة .

٢-١٣-١ مسجل تنشيط المقاطعة Interrupt Enable, IE

شكل (٢-٢٥) يبين محتويات هذا المسجل وأما وظيفة كل بت من بتاته فهى كما يلى:

7	6	5	4	3	2	1	0	
EA	—	ET2	ES	ET1	EX1	ET0	EX0	IE
								وعنوانه A8

شكل (٢-٢٥) مسجل تنشيط المقاطعة IE

- **بت تنشيط أو إخماد كل المقاطعات Enable All Interrupts, EA** عندما تكون هذه البت بصفر تخمد جميع المقاطعات ولا يسمح بأى منها ، وعندما تكون واحد يسمح لكل المقاطعات .
- **بت تنشيط مقاطعة البيانات المتتالية Enable Serial interrupt, ES** عندما تكون هذه البت بواحد يسمح بالمقاطعة من على بوابة البيانات المتوالية ، وعندما تكون صفر لا يسمح بالمقاطعة على هذه البوابة .
- **بتات تنشيط مقاطعة المؤقتات Enable Timer ET1, ET0** عندما تكون هذه البت بواحد ينشط علم المؤقتات TF1, TF0 وعندما تكون بصفر لا يسمح بالمقاطعهة من المؤقتات .
- **بتات تنشيط المقاطعة الخارجية Enable external interrupt, EX1, EX0** عندما تكون هذه البت بواحد يتم تنشيط المقاطعة الخارجية على الخطوط INT0 و INT1 ، عندما تكون بصفر لا يسمح بالمقاطعة على هذه الأطراف .
- **الطرفين ٥ و ٦** غير مستخدمة .

مما سبق يتضح أن بتات هذا المسجل خاصة بتنشيط أو إخماد أى واحدة من المقاطعات الستة منفردة ، أو إخماد كل المقاطعات أو تنشيطها كلها من خلال البت EA(7) . معنى ذلك أنه لكى ننشط أى مقاطعة لابد من وضع EA=1 ثم وضع البت المقابلة لهذه المقاطعة بواحد وباقى البتات أصفارا . عند عمل Reset للمتحكم فإن كل بتات هذا المسجل توضع بأصفار .

٢-١٣-٢ مسجل أولوية المقاطعة Interrupt Priority, IP

هذا المسجل يبين سلوك المتحكم فى حالة حدوث مقاطعة أو أكثر من أكثر من مصدر فى نفس الوقت . عند توصيل القدرة إلى المتحكم أو إعادة وضعه Reset فإن المتحكم يضع أولوية تلقائية لهذه المقاطعات كما هو مبين فى جدول ٢-١٠ . كما هو مبين فى هذا الجدول فإنه إن حدثت مقاطعة على الطرفين الخارجيين INT0 و INT1 مثلا فإن المقاطعة INT0 يتم خدمتها أولا .

جدول ٢-١٠

اسم المقاطعة	المقاطعات من الأولوية الأعلى للأدنى
INT0	القاطعة الخارجية INT0
TF0	المقاطعة من المؤقت صفر TF0
INT1	المقاطعة الخارجية INT1
TF1	المقاطعة من المؤقت واحد TF1
R1, T1	القاطعة من البيانات التتابعية

يمكن للمستخدم أن يغير من هذا الوضع التلقائى للأولويات عن طريق وضع بتات المسجل IP الموضح فى شكل (٢-٢٦) . نلاحظ من هذا الشكل أن كل واحدة من المقاطعات الخمس ممثلة ببت فى هذا المسجل . بوضع واحد فى البت المقابلة لأى مقاطعة فإن هذه المقاطعة تأخذ الأولوية الأعلى . أى مقاطعة يمكنها أن تأخذ القيمة واحد (الأولوية الأعلى) ، أو القيمة صفر (الأولوية الأدنى) فقط . إذا تم وضع أكثر من مقاطعة فى الأولوية واحد وحدث تنشيط لاثنين منها فى نفس الوقت ، فإن المتحكم يرتب هذه المقاطعات على حسب الوضع الموجودة فى جدول ٢-٩ .

وظيفة بتات المسجل IP يمكن بيانها كما يلى :

- بت أولوية مقاطعة البيانات المتتالية PS .
- بتات أولوية المقاطعة من المؤقتات PT1, PT0 .
- بتات أولوية المقاطعة من على الأطراف الخارجية PX1, PX0 .
- البتات 5 و 6 و 7 غير مستخدمة .

نذكر هنا أنه بوضع واحد فى أى بت من هذه البتات فإن المقاطعة المقابلة لها تأخذ الأولوية الأعلى ، وبوضعها بصفر فإن هذه المقاطعة تأخذ الأولوية الأدنى .

7	6	5	4	3	2	1	0	
—	—	PT2	PS	PT1	PX1	PT0	PX0	IP
								وعنوانه B8

شكل (٢-٢٦) مسجل أولوية المقاطعة

مثال ٢-٤ :

أكتب برنامجا يقرأ البوابة P0 ويرسل محتوياتها إلى البوابة P1 وذلك دائما أو إلى مالانهاية . فى هذه الأثناء مطلوب استخدام المؤقت T0 لتوليد موجة مربعة على

طرف الخرج P2.1 بزمن دورى مقداره ٢٠٠ ميكروثانية . استخدم بللورة ذات تردد 11.0592 ميجاهرتز كمولد لنبضات التزامن .
التردد 11.0592 ميجاهرتز بعد قسمته على ١٢ نحصل على زمن دورة أمر مقدارها 1.085 ميكروثانية . بقسة 100 (نصف زمن دورة الموجة المطلوبة) على 1.085 نحصل على الرقم 92 تقريبا . الآن يجب تحميل TH0 برقم بحيث عندما يضاف إليه الرقم 92 نحصل على الرقم FFH . لاحظ أن الرقم 92d=5CH ، لذلك فإن الرقم FF-5C=A4H سيكون هو الرقم المفروض أن نضعه فى TH0 لنبدأ العد منه . لذلك سنستخدم المؤقت فى الحالة اثنان mode 2 بحيث أنه مع تنشيط المؤقت تحمل القيمة A4H من TH0 إلى TL0 ويبدأ المؤقت فى العد منها إلى أن يصل للقيمة TL0=FFH حيث عندها يصبح العلم TF0=1 ويسبب مقاطعة للمتحكم . عند حدوث هذه المقاطعة يذهب المتحكم إلى العنوان 000BH فى متجه المقاطعة كما ذكرنا . عند هذا العنوان سنكتب برنامج خدمة هذه المقاطعة وهو برنامج قصير فى هذه الحالة بحيث يمكن كتابته ابتداء من هذا العنوان دون الدخول فى العنوان التالى فى متجه المقاطعة كما سنرى فى البرنامج التالى . لاحظ أن البرنامج يبدأ من العنوان 0000H بأمر قفز بحيث يمكن وضع البرنامج خارج نطاق متجه المقاطعة .

```
        ORG 0000H
        LJMP Start  ; للخروج من نطاق متجه المقاطعة
        ORG 000BH   ; بداية برنامج خدمة المقاطعة للمؤقت
        CPL P2.1    ; إعكس طرف الخرج
        RETI        ; عودة من برنامج خدمة المقاطعة
        ORG 0030H
Start:  MOV TMOD,#02H; تشغيل المؤقت صفر فى الحالة صفر
        MOV P0,#FFH ; لجعل البوابة صفر بوابة إخراج
        MOV TH0,#A4
        MOV IE,#82H ; تنشيط مقاطعة المؤقت صفر وأيضا كل المقاطعات
        SETB TR0    ; بدأ المؤقت صفر
Here:   MOV A,P0
        MOV P1,A
        SJMP Here
        END
```

٢-١٤ إدخال وإخراج البيانات التتابعية
Serial data input/output

سنتناول فى هذا الجزء كيفية الاتصالات التتابعية مع المتحكم 8051 فقط ويرجى لكى تكتمل الصورة فى ذهن القارىء أن يقوم بمراجعة الفصل الخاص بالاتصالات التتابعية . الشريحة 8051 لها طرفان يتم من خلالهما التراسل معها تتابعيا ، الطرف الأول وهو الطرف TXD الذى يتم من خلاله إخراج أو إرسال البيانات التتابعية إلى

خارج الشريحة وهذا الطرف هو الطرف رقم ١ فى البوابة ٣ أى الطـرف P3.1 . الطرف الثانى هو الطرف RXD الذى يتم من خلاله إدخـال أو اسـتقبال البيانـات التتابعية من خارج الشريحة وهذا الطرف هو الطرف رقـم 0 فـى البوابـة ٣ أى الطرف P3.0 . يتحكم فى عملية التراسل ثلاث مسجلات كما يلى :

٢-١٤-١ مسجل عزل البيانات Serial Buffer, SBUF

هذا المسجل يمثل عازل (مخزن مؤقت) للبيانات المرسلة أو المستقبلة ، لذلك يمكـن النظر إليه على أنه مسجلان لهما نفس الرقم أو العنوان وهو 99H . المسجل الأول يمثل عازل إرسال يتم فيه تسجيل أى بيانات يراد إرسالها تتابعيا خـارج الشـريحة على الطرف TXD . المسجل الثانى عازل استقبال يتم فيه استقبال أى بيانات تتابعية من الطرف RXD تمهيدا لقراءتها كبايت كاملة . عملية التراسل المتتـالى تسـتلزم إضافة بتات توضح بداية ونهاية كل بايت وغالبا ما تكون بت فى البدايـة start bit وبت فى نهاية البايت stop bit . من مهمة المسجل SBUF أيضا إضافة بتات البداية والنهاية فى حالة الإرسال ، وحذف هذه البتات المضافة مـن البيانـات التـى يـتم استقبالها لتبقى البايت الخاصة بالبيانات الحقيقية فقط . هذا المسجل يمكـن التعامـل معه كأى واحد من المسجلات كما فى الأوامر التالية :

```
MOV SBUF,#'D'   يحمل المسجل بالقيمة الأسكى للحرف وهى ٤٤
MOV SBUF,A
MOV A,SBUF
```

غالبا ما يتم التراسل المتتالى بين المتحكم والحاسب الشخصى من خـلال المخـرج المتتالى COM PORT والذى يعمل بالنظام القياسى RS232 الذى يعمل أيضا عنـد معدلات تراسل قياسية . لذلك فإنه لكى نضمن الكفاءة لعملية التراسل بين المـتحكم والحاسب لابد أن يعمل المتحكم عند معدل تراسل baud rate متوائم مع أحد معدلات تراسل الحاسب القياسية . المتحكم 8051 به وحدة خاصة بهذا النوع مـن التراسـل وهى الوحدة المسماة بوحدة "الاستقبال والإرسـال اللاتـوافقى العامـة" Universal Asynchronous Receiver Transmitter, UART كما نعلم فإن المتحكم يقسم تـردد الساعة على ١٢ للحصول على تردد دورة الأمر machine cycl . بعد ذلك تقـوم وحدة التراسل UART بقسمة هذا التردد على ٣٢ مرة أخرى قبل أن تـدخل علـى المؤقت T1 الذى يمكن برمجته للحصول على معدل التراسل المطلوب baud rate . كمثال على ذلك نفرض ان تردد الساعة هو 11.0592 ميجاهرتز ، بقسمة هذا التردد على ١٢ نحصل على 921.6 كيلوهرتز التى تمثـل تـردد دورة الأمـر . بقسـمة 921.6KHz على 32 نحصل على 28800 هرتز . هذا التردد يتم إدخاله على المؤقت T1 وضبطه ليعمل فى الطريقة اثنين mode 2 التى يتم فيها تحميل الجزء الأعلى من المؤقت TH1 بقيمة معينة يبدأ منها الجزء الأدنى TL1 عملية العـد (راجـع جـزء

المؤقتات) حتى يصل إلى القيمة (FFH)256 . لذلك فإن المؤقت يقسم تردد الدخل على الفرق بين القيمة العظمى 256 والقيمة التى يبدأ عملية العد من عندها والموجودة فى TH1 بعد إضافة واحد . لذلك لكى نحصل على معدل التراسل القياسى 9600 فإننا يجب أن نقسم التردد 28800Hz على 3 ، ويتم ذلك عن طريق تحميل TH1 بالقيمة FD=1+(FF-3) . جدول ٢-١١ يبين القاسم المطلوب استخدامه مع التردد 28800Hz للحصول على بعض معدلات التراسل القياسية والرقم المقابل له المفروض وضعه فى TH1 لتبدأ عملية العد منه .

جدول ٢-١١

TH1	القاسم المطلوب	معدل التراسل القياسى المطلوب
FD	3	9600
FA	6	4800
F4	12	2400
E8	24	1200

٢-١٤-٢ مسجل التحكم التتابعى Serial Control register, SCON

شكل (٢-٢٧) يبين بتات هذا المسجل وعنوانه وفيما يلى نقدم شرحا لوظيفة كل بت من بتات هذا المسجل :

جدول ٢-١٢

الطريقة mode	SM1	SM0
الحالة صفر mode 0	0	0
الحالة واحد mode 1	1	0
الحالة اثنين mode 2	0	1
الحالة ثلاثة mode 3	1	1

- **البتات 6 و 7** وهى بتات الحالة mode التى تعمل عندها الشريحة SM0, SM1 حيث يمكن للمتحكم أن يتمم عملية التراسل التتابعى فى واحدة من أربع طرق كما فى جدول ٢-١٢ وكلها تحدد طريقة وضع البيانات فى حزم packets تمهيدا لإرسالها .

7	6	5	4	3	2	1	0	
SM0	SM1	SM2	REN	TB8	RB8	TI	RI	SCON وعنوانه 98

شكل (٢-٢٧) مسجل التحكم فى التراسل التتابعى

<u>فى الحالة صفر mode 0</u> تكون طريقة التراسل توافقية وبدون أى بتات إضافية ، أى أنه يتم إرسال أو استقبال ٨ بت فقط ، ويتم إرسال أو استقبال البيانات على الطرف RxD(P3.0) وأما نبضات التزامن أو التوافق فتخرج على الطرف TxD(P3.1) . معدل التراسل فى هذه الحالة يكون هو نبضات دورة الأمر أى fosc/12 بت فى الثانية . عملية الإرسال transmission تبدأ بمجرد تحميل البيانات المراد إرسالها فى المسجل SBUF ، وأما عملية الاستقبال فتتم بمجرد تنشيط أعلام الاستقبال $\overline{RI} = 0$ و REN=1 فى المسجل SCON . كما نلاحظ أن

هذه الطريقة يمكن النظر إليها على أنها طريقة لتحويل البيانات من الصورة المتوازية إلى الصورة التتابعية والعكس بطريقة سريعة ولكنها غير مناسبة للتراسل مع الحاسب .

في الحالة واحد mode 1 وهى الأكثر شيوعا فى المتحكم 8051 والأنسب للتراسل مع الحاسب ، وفيها يتم التراسل لاتوافقيا حيث لا يكون هناك نبضات توافق كما فى الحالة صفر ، ولكن يتم إضافة بت تبين بدأ الإرسال start bit ثم ترسل البيانات (٨ بت) ثم بت تبين نهاية الإرسال stop bit . لذلك فإن البيانات ترسل فى هذه الحالة فى صورة حزم كل منها ١٠ بت حيث ترسل البت ذات القيمة الصغرى LSB أولا ، ويتم وضع العلم TI بواحد فى نهاية الإرسال بعد البت العاشرة . هنا يتم الإرسال على الطرف TxD بينما يتم الاستقبال على الطرف RxD ، لذلك فإن عمليات الإرسال والاستقبال يمكن أن تتم فى نفس التوقيت full duplex . تتم عملية الإرسال بمجرد وضع البيانات فى المسجل SBUF وأما عملية الاستقبال فتتم بمجرد نزول الطرف RxD من واحد إلى الصفر حيث تهمل بت البداية وبعدها يتم نقل البيانات ٨ بت إلى المسجل SBUF ثم يتم تخزين بت التوقف فى البت RB8 فى المسجل SCON وبعدها يتم وضع العلم RI=1 للدلالة على تمام استقبال بايت كاملة يمكن قراءتها . فى هذه الحالة يمكن ضبط معدل التراسل باستخدام المؤقت T1 عن طريق تشغيله فى الطريقة اثنين mode 2 كما ذكرنا وتحميل المسجل TH1 بالقيمة الابتدائية التى يبدأ العد من عندها تبعا للمعادلات التالية :

أولا فى حالة SMOD(PCON.7)=0

$$TH1 = 256 - \frac{f_{osc}}{32x12xBaud_rate}$$

ثانيا فى حالة SMOD(PCON.7)=1

$$TH1 = 256 - \frac{f_{osc}}{16x12xBaud_rate}$$

كما نلاحظ من المعادلتين السابقتين فإن معدل التراسل أيضا فى الحالات واحد واثنين وثلاثة يعتمد على محتويات آخر بت فى المسجل PCON الذى سنتكلم عنه بعد قليل .

فى الحالة اثنين mode 2 يتم وضع البيانات فى حزم مثل الحالة واحد ولكن الحزمة فى هذه الحالة تكون ١١ بت بدلا من ١٠ بت حيث يتم إرسال بت البداية start bit ، ثم البيانات ٨ بت حيث يتم إرسال البت الأدنى أولا ، ثم البت التاسعة وهى البت TB8 فى المسجل SCON فى حالة الإرسال ثم بت التوقف stop bit التى تبين نهاية الحزمة . فى حالة الاستقبال يتم وضع البت التاسعة فى البت RB8 فى المسجل SCON . معدل التراسل فى هذه الحالة لا يتم التحكم فيه عن طريق المؤقت T1 كما فى الحالة الأولى ولكنه يكون معدل ثابت يعطى بالعلاقة التالية :

Buad rate=fosc/64 فى حالة SMOD(PCON.7)=0
Buad rate=fosc/32 فى حالة SMOD(PCON.7)=1

كما نلاحظ فإن هذه الحالة غير مناسبة للتراسل مع الحاسبات .

<u>فى الحالة ثلاثة mode 3</u> مثل الحالة اثنين تماما يتم تكوين الحزمة من ١١ بت ولكن الاختلاف يكون فى معدل التراسل حيث يكون فى هذه الحالة مثله مثل الحالة واحد التى يكون فيها معدل التراسل باستخدام المؤقت T1 .

- **البت 5 ، SM2** : بت التراسل بين المتحكمات بعضها البعض فى الحالتين اثنين وثلاثة فقط . فى هاتين الحالتين إذا كانت هذه البت بواحد فإنه إذا كانت البت التاسعة التى تم استقبالها بواحد فإن المتحكم يولد طلب مقاطعة عن طريق وضع العلم RI=1 ، وإذا كانت البت التاسعة بصفر فلن يتولد طلب المقاطعة . هذه البت ليس لها أى تأثير فى الحالتين صفر وواحد لذلك يتم وضعها بصفر .
- **البت 4 ، REN** توضع هذه البت بواحد لتنشيط عملية الاستقبال ، وتوضع بصفر لمنع أو إخماد هذه العملية .
- **البت 3 ، Transmitted bit 8, TB8** فى الحالتين اثنين وثلاثة من حالات التراسل التتابعى يضع المستخدم هذه البت بواحد أو صفر لتنشيط عملية المقاطعة فى حالة التراسل بين متحكمين .
- **البت 2 ، Received bit 8, RB8** فى الحالتين اثنين وثلاثة من حالات التراسل التتابعى توضع البت التاسعة التى تم استقبالها فى هذه البت ، فى الحالة واحد توضع بت التوقف stop bit .
- **البت 1 ، Transmit Interrupt, TI** علم مقاطعة الإرسال الذى يوضح نهاية إرسال بايت ، يصبح واحد عند نهاية إرسال البت ٧ فى الحالة صفر ومع بداية إرسال بت التوقف فى الحالات الأخرى . بعد أن تصبح واحد لابد من تصفيرها عن طريق المستخدم .
- **البت 0 ، Receive Interrupt, RI** علم مقاطعة الإستقبال الذى يوضح أن هناك بايت تم استقبالها فى المسجل SBUF ولابد من قراءتها قبل أن تفقد باستقبال بيانات أخرى ، يصبح واحد عند نهاية إستقبال البت ٧ فى الحالة صفر ومع استقبال بت التوقف فى الحالات الأخرى . بعد أن تصبح واحد لابد من تصفيرها عن طريق المستخدم . المسجل SCON يمكن التعامل معه على مستوى البت مثل SCON0 أو SCON5 وهكذا .

مثال ٢-٥ :

برنامج يرسل كلمة hello على التتابع بمعدل إرسال 9600 بت فى الثانية باستخدام بت توقف واحدة وبت للبداية وبيانات من ٨ بت (الحالة واحد) . هذا البرنامج سيكون كما يلى :

```
        MOV TMOD,#20H     المؤقت ١ فى الحالة ٢
        MOV TH1, #-3      للحصول على معدل التراسل ٩٦٠٠
        MOV SCON,#50H     تراسل تتابعى فى الحالة واحد مع تنشيط الاستقبال
        SETB TR1          إبدأ تشغيل المؤقت واحد
Here1:  MOV A,#"h"
        ACALL Transmit
        MOV A,#"e"
        ACALL Transmit
        MOV A,#"l"
        ACALL Transmit
        MOV A,#"l"
        ACALL Transmit
        MOV A,#"o"
        ACALL Transmit
        SJMP Here1
Transmit: MOV SBUF,A      بدأ عملية إرسال بايت
Here:   JNB TI,Here       إنتظر حتى نهاية عملية إرسال البايت
        CLR TI            لابد من تصفير هذا العلم عن طريق البرنامج
        RET
```

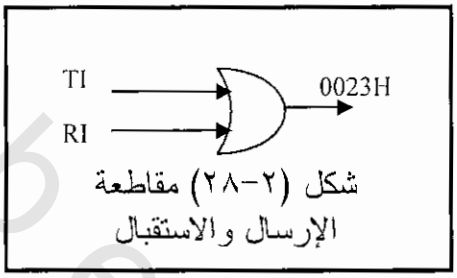
بعد هذا المثال نؤكد على أهمية العلم TI فى عملية الإرسال وكيف أنه يصبح واحد عن طريق المتحكم فى نهاية إرسال أى بايت وعلى المستخدم أو المبرمج أن يجعله بصفر قبل البدأ فى إرسال البايت التالية . على المبرمج أيضا ألا يبدأ فى إرسال بايت جديدة قبل التأكد من أن العلم TI أصبح واحد دليل على إتمام إرسال البايت السابقة وأن المسجل SBUF فاضى وجاهز للبايت التالية .

مثال ٢-٦ :

برنامج يستقبل البيانات المتتالية على الطرف RxD ويضعها على البوابة P1 .

```
        MOV TMOD,#20H     المؤقت ١ فى الحالة ٢
        MOV TH1, #-3      للحصول على معدل التراسل ٩٦٠٠
        MOV SCON,#50H     تراسل تتابعى فى الحالة واحد مع تنشيط الاستقبال
        SETB TR1          إبدأ تشغيل المؤقت واحد
        CLR RI            تصفير علم الاستقبال تمهيدا لاستقبال بايت جديدة
Here:   JNB RI, Here      إنتظر وصول حرف جديد
        MOV A,SBUF
        MOV P1,A
        CLR RI
        SJMP Here
```

فى المثالين السابقين وبالتحديد فى الحلقة Here فى كل منهما كان المتحكم ينتظر إما العلم TI أو العلم RI أن يصبح واحدا ، وفى أثناء فترة الانتظار هذه فإن المتحكم لا يفعل شيء سوى الانتظار ونقول فى هذه الحالة أن المتحكم منجذب pulled بهذا العلم . هذا بالطبع يعتبر إهدار لوقت المتحكم حيث يمكن فى أثناء هذه الفترة أن يفعل أى شيء آخر وعندما يصبح أى علم منهما بواحد فإنه يقاطع المتحكم حيث عندها يذهب المتحكم إلى برنامج خدمة معين خاص بهذا العلم . كما فى شكل (٢-٢٨) فإنه عندما يصبح أى واحد من العلمين TI أو RI بواحد فإنه يذهب إلى العنوان 0023H فى متجه المقاطعة . لابد فى هذه الحالة أن تكون بت تنشيط المقاطعة التتابعية ES وهى البت رقم ٤ فى مسجل تنشيط المقاطعة (IE.4) بواحد . عندما يذهب المتحكم إلى برنامج الخدمة لابد من عمل اختبار لمعرفة مصدر المقاطعة وهل هى مقاطعة إرسال بناء على العلم TI أم مقاطعة استقبال بناء على العلم RI . المثال التالى يوضح ذلك .



شكل (٢-٢٨) مقاطعة الإرسال والاستقبال

مثال ٢-٧ :

برنامج يقرأ البيانات الموجودة على بوابة الإدخال P1 ويرسلها باستمرار إلى البوابة P2 ، بينما البيانات التتابعية التى يتم قراءتها يرسلها إلى البوابة P0 باستخدام المقاطعة . تردد الساعة هو 11.0592MHz ومعدل التراسل المطلوب هو 9600 بت فى الثانية .

```
        ORG 00
        LJMP Main       قفز لبداية البرنامج الأساسى ليعبر متجه المقاطعة
        ORG 23H         عنوان المقاطعة التتابعية فى متجه المقاطعة
        LJMP Serial     قفز لبرنامج خدمة المقاطعة التتابعية
        ORG 30H         بداية البرنامج الأساسى
Main:   MOV P1,#0FFH    وضع البوابة واحد كبوابة إدخال
        MOV TMOD,#20H   وضع المؤقت واحد فى الحالة اثنين
        MOV TH1,#0FDH   للحصول على معدل التراسل ٩٦٠٠ بت فى الثانية
        MOV SCON,#50H   التراسل التتابعى فى الحالة واحد مع تنشيط الاستقبال
        MOV IE,90H      تنشيط المقاطعة التتابعية فى مسجل تنشيط المقاطعة
        SETB TR1        بدأ المؤقت واحد
Back:   MOV A,P1        حلقة لا نهائية يقرأ البوابة واحد ويخرجها على البوابة اثنين
        MOV P2,A
        SJMP Back
-------------------------------------------------------------------------
        ORG 100H        بداية برنامج خدمة المقاطعة
```

```
Serial:    JB TI, Transmit  إذا كانت المقاطعة نتيجة علم الإرسال إذهب للجزء الخاص بذلك
           MOV A, SBUF      نقل البيانات التى تم استقبالها إلى المركم
           MOV P0,A         ومنه إلى البوابة صفر
           CLR RI           ثم لا ننسى تصفير علم الاستقبال
           RETI             ثم نرجع للبرنامج الأساسى
Transmit:  CLR TI           هنا المقاطعة نتيجة علم الإرسال حيث يتم تصفيره
           RETI             ثم العودة إلى البرنامج الأساسى
           END
```

٢-١٤-٣ مسجل التحكم فى القدرة Power Mode Control register, PCON

هذا المسجل لا يمكن التعامل معه على مستوى البت . شكل (٢-٢٩) يبين محتويات هذا المسجل حيث نلاحظ وجود بت واحدة (بت ٧) خاصة بالتحكم فى التراسل التتابعى ، واثنين بت للتحكم فى حالة القدرة ، واثنان بت تستخدمان كأعلام عامة ، وباقى البتات غير مستخدمة ، وتفاصيل هذه البتات كما يلى :

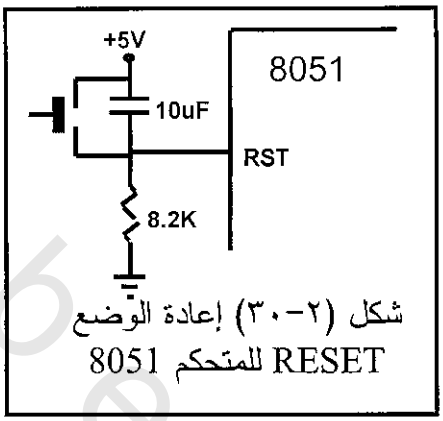
7	6	5	4	3	2	1	0	
SMOD	–	–	–	GF1	GF0	PD	IDL	PCON وعنوانه ٨٧

شكل (٢-٢٩) محتويات المسجل PCON

- **البت 7 Serial baud rate modify, SMOD bit,** بوضع هذه البت بواحد يتم مضاعفة معدل التراسل التتابعى فى الحالات 1 و 2 و 3 التى تعتمد على المؤقت T1 فى تحديد معدل التراسل كما رأينا سابقا عند الحديث عن المسجل SCON . بوضعها بصفر يتحدد معدل التراسل فى هذه الحالات تبعا للمؤقت T1 فقط .
- **البتات 2 و 3 General purpose user flags, GF1, GF2** هاتين البتين تستخدمان كعلمين تحت تصرف المبرمج للأغراض العامة .
- **البت 1 Power down bit, PD,** توضع بواحد لإدخال المتحكم فى حالة تخفيض القدرة . فى هذه الحالة يقوم المتحكم بحفظ محتويات ال RAM وكل المسجلات ، ثم يوقف المذبذب Oscillator حيث تتجمد جميع مكونات المتحكم وتتوقف عن العمل ، ويظل المتحكم كذلك إلى أن يقوم المستخدم بعمل RESET للمتحكم .
- **البت 0 Idle mode bit, IDL,** توضع بواحد لإدخال المتحكم فى حالة الخمول Idle . فى هذه الحالة تتوقف وحدة الحساب عن العمل وإجراء أى عمليات حسابية ، ولكنه يسمح لأنظمة المقاطعة والمؤقتات والتخزين فقط فى ال RAM بالعمل .

٢-١٥ طرف إعادة الوضع للمتحكم RESET

هذا الطرف هو الطرف ٩ فى الشريحة وهو عالى الفعالية بمعنى أن هذا الطرف يجب أن يكون صفر دائما إلا فى حالة الرغبة فى عمل إعادة وضع للشريحة فإنه يوصل على جهد القدرة Vcc . شكل (٢-٣٠) يبين دائرة مقترحة للحصول على ذلك .



شكل (٢-٣٠) إعادة الوضع RESET للمتحكم 8051

جدول ٢-١٣ يبين محتويات جميع مسجلات المتحكم عند إعادة وضعه . هذه القيم مهم أن يعرفها المبرمج حتى يعرف ماذا سيكون هناك فى أى مسجل عند إعادة وضع المتحكم أو توصيل القدرة إليه .

جدول ٢-١٣

المسجل Register	القيمة Value
PC	0000h
DPTR	0000h
A	00h
B	00h
SP	07h
PSW	00h
P0-P3	FFh
IP	xxx00000h
IE	0xx00000h
TCON	00h
TMOD	00h
TH0	00h
TL0	00h
TH1	00h
TL1	00h
SCON	00h
SBUF	xxh

٢-١٦ الإصدارات الأخرى للمتحكم 8051

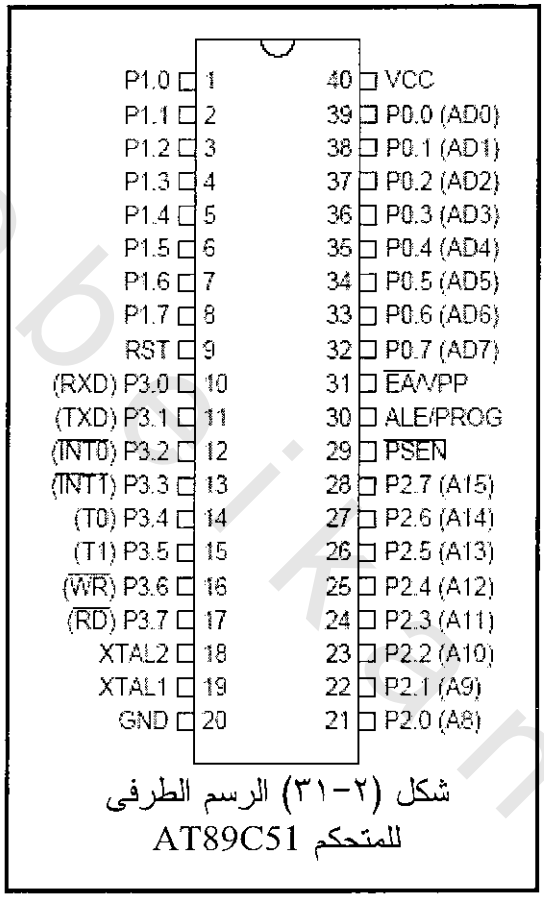
كما ذكرنا فى بداية الفصل أن هناك بعض الشركات الأخرى التى تنتج المتحكم 8051 ولكن مع بعض الاختلافات البسيطة فى المواصفات ولكن مجموعة الأوامر تبقى كما هى فى كل هذه الإصدارات ، وسنقدم هنا بعض هذه الإصدارات مع ذكر خواص كل منها فقط لنرى كيف أن الاختلافات بسيطة . من أهم هذه الإصدارات إصدارات شركة أنتل من هذا المتحكم .

٢-١٦-١ المتحكم AT89C51

شكل (٢-٣١) يبين أطراف هذه الشريحة وهى كما نرى من الشكل متطابقة تماما مع المتحكم Intel8051 طرفا بطرف . مواصفات هذا الإصدار يمكن تلخيصها فيما يلى :

- متوافق تماما مع المتحكم الأصلى Intel8051 .
- 4 كيلوبايت ذاكرة لحظية flash memory قابلة للمسح والبرمجة ١٠٠٠ مرة . وهذا هو الاختلاف الأساسى فى هذا الإصدار حيث أن هذه الذاكرة فى المنتج الأساسى كانت EPROM .

- نبضات ساعة من صفر حتى ٢٤ ميجاهرتز .
- ذاكرة بيانات 128x8 RAM بايت تماما مثل المنتج الأصلى .
- ٣٢ طرف إدخال/إخراج للبيانات.
- مؤقتان كل منهما ١٦ بت .
- ستة مصادر للمقاطعة (تماما مثل المنتج الأصلى) .
- استقبال وإرسال البيانات على التوالى .
- إمكانية تخفيض القدرة power down أو حالة الخمول Idle .

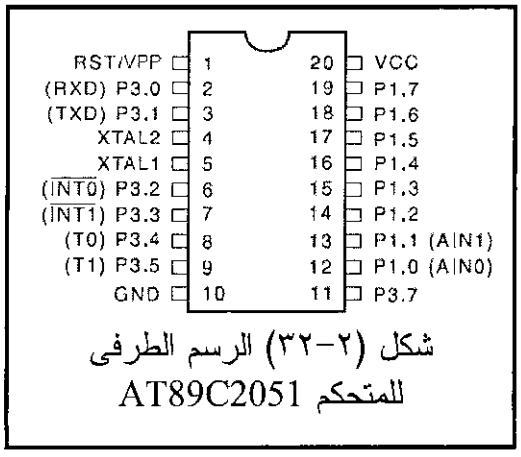


شكل (٢-٣١) الرسم الطرفى
للمتحكم AT89C51

٢-١٦-٢ المتحكم AT89C2051

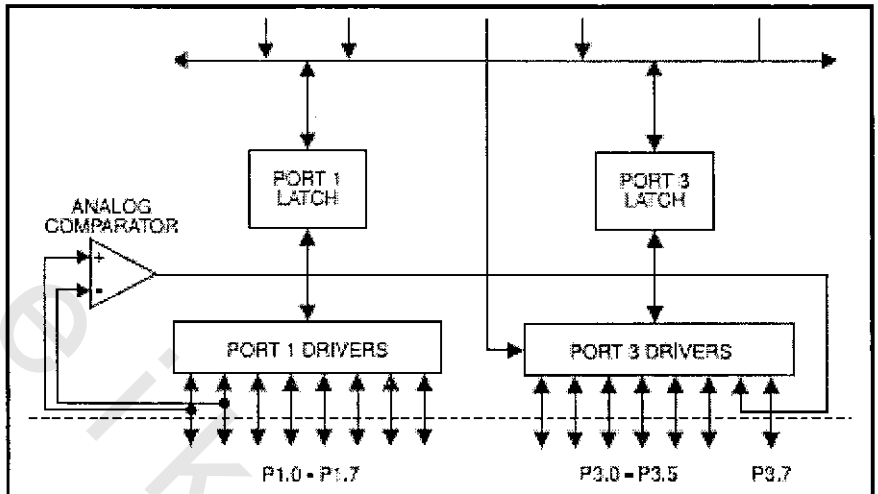
شكل (٢-٣٢) يبين الرسم الطرفى لهذا المتحكم حيث نلاحظ أنه يتكون من ٢٠ طرفا بدلا من ٤٠ كما كان فى المنتج الأصلى ، ولكن مجموعة الأوامر كما هى تماما لكل من المتحكمين وسنقدم مواصفات هذا المتحكم فيما يلى :

- متوافق تماما مع المتحكم الأصلى Intel8051 من حيث الأوامر .
- ٢ كيلوبايت ذاكرة لحظية flash memory قابلة للمسح والبرمجة ١٠٠٠ مرة بدلا من ٤ كيلوبايت .
- نبضات ساعة من صفر حتى ٢٤ ميجاهرتز .
- ذاكرة بيانات 128x8 RAM بايت تماما مثل المنتج الأصلى .
- ١٥ طرفا إدخال/إخراج للبيانات .
- مؤقتان كل منهما ١٦ بت .
- ستة مصادر للمقاطعة (تماما مثل المنتج الأصلى) .



شكل (٢-٣٢) الرسم الطرفى
للمتحكم AT89C2051

- استقبال وإرسال البيانات على التوالى .
- إمكانية تخفيض القدرة power down أو حالة الخمول Idle .
- مقارن انسيابى analog comparator .



شكل (٢-٣٣) وضع المقارن الانسيابى داخل المتحكم AT89C2051

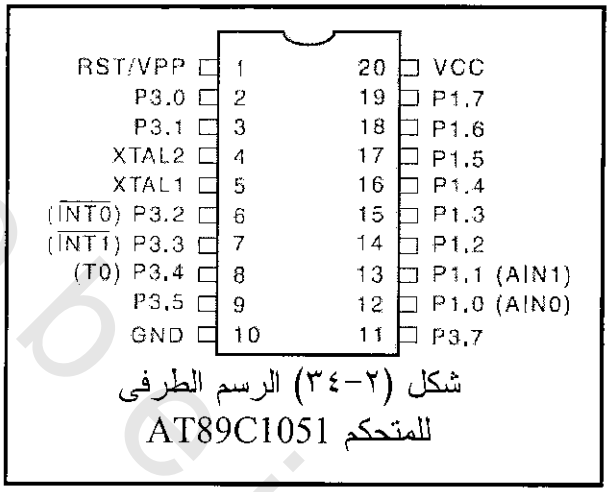
كما نلاحظ فى شكل (٢-٣٣) فإن طرفا الدخل للمقارن الانسيابى هما الطرفين 12(P1.0) و 13(P1.1) فى الشريحة ، وأما خرج المقارن فموصل كطرف دخل على البوابة 3 وهو الطرف P3.6 . وجود مثل هذا المقارن يفيد جدا فى الكثير من التطبيقات حيث يمكن باستخدام محول رقمى انسيابى D/A وبرنامج بسيط ينفذه المتحكم أن نحصل على محول انسيابى رقمى A/D .

٢-١٦-٣ المتحكم AT89C1051

شكل (٢-٣٤) يبين الرسم الطرفى لهذا المتحكم حيث نلاحظ أنه يتكون من ٢٠ طرفا بدلا من ٤٠ كما كان فى المنتج الأصلى ، ولكن مجموعة الأوامر كما هى تماما لكل من المتحكمين وهو أقرب ما يكون من المتحكم السابق AT89C2051 وسنقدم مواصفات هذا المتحكم فيما يلى :

- متوافق تماما مع المتحكم الأصلى Intel8051 من حيث الأوامر .
- 1 كيلوبايت ذاكرة لحظية flash memory قابلة للمسح والبرمجة ١٠٠٠ مرة بدلا من ٤ كيلوبايت .
- جهد قدرة من 2.3V حتى 6V وهذه ميزة عظيمة حيث يمكن تشغيله من بطاريات .

- نبضات ساعة من صفر حتى 24 ميجاهرتز .
- ذاكرة بيانات RAM ٦٤ بايت .
- ١٥ طرف إدخال/إخراج للبيانات .
- مؤقت واحد ١٦ بت .
- ثلاثة مصادر للمقاطعة .
- إمكانية تخفيض القدرة power down أو حالة الخمول Idle .
- مقارن انسيابي analog comparator .

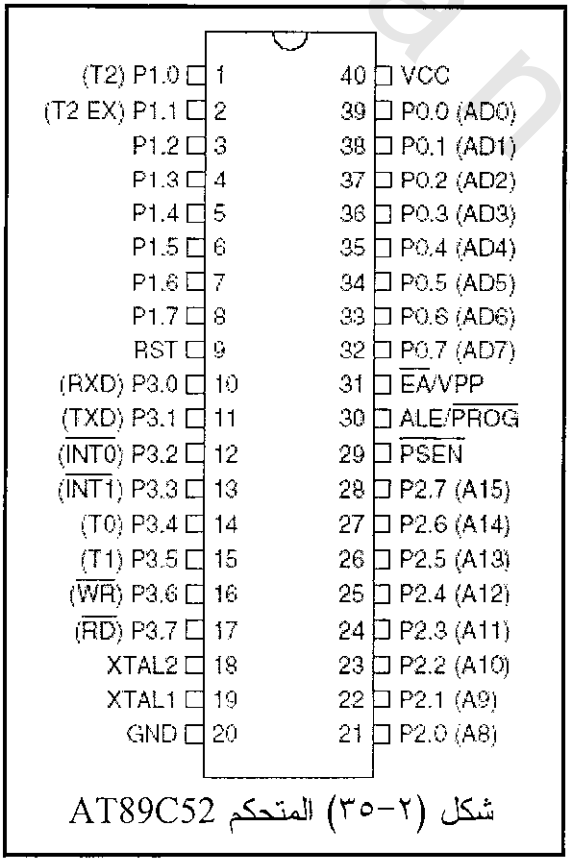


شكل (٢-٣٤) الرسم الطرفى
للمتحكم AT89C1051

٢-١٦-٤ المتحكم AT89C52

شكل (٢-٣٥) يبين أطراف هذه الشريحة وهى كما نرى من الشكل متطابقة تماما مع المتحكم Intel8051 طرفا بطرف فيما عدا بعض الإضافات . مواصفات هذا الإصدار يمكن تلخيصها فيما يلى :

- متوافق تماما مع المتحكم الأصلى Intel8051 وبالذات من حيث الأوامر .
- ٨ كيلوبايت ذاكرة لحظية flash memory قابلة للمسح والبرمجة ١٠٠٠ مرة .
- نبضات ساعة من صفر حتى ٢٤ ميجاهرتز .
- ذاكرة بيانات 256x8 RAM بايت .
- ٣٢ طرف إدخال/إخراج للبيانات .
- ثلاثة مؤقتات/عدادات كل منها ١٦ بت .
- ثمانية مصادر للمقاطعة .



شكل (٢-٣٥) المتحكم AT89C52

obeikandl.com

- استقبال وإرسال البيانات على التوالى .
- إمكانية تخفيض القدرة power down أو حالة الخمول Idle .

بالطبع فإن هناك إصدارات من هذا المتحكم من شركات أخرى ولكن لن نتكلم عنها هنا نظرا لعدم شيوعها .

obeikandl.com