الفصل الثالث ٣

المتحكمات بيك بسيطة الإمكانيات

The Low Range PIC Microcontrollers

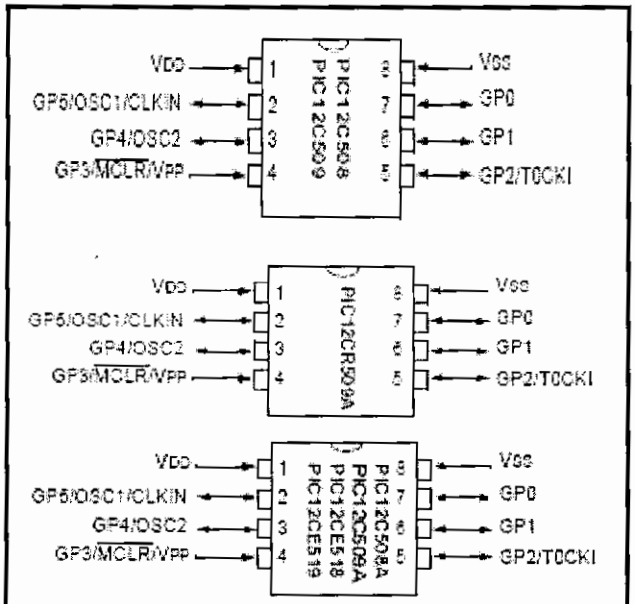
٣-١ مقدمة

بالرغم من وجود العديد من الشركات التى تنتج المتحكمات المختلفة ، إلا أن عائلة المتحكمات بيك PIC التى تنتجها شركة Microchip تعتبر الأكثر شيوعا من حيث الاستخدام هذه الأيام نتيجة تواجدها فى أكثر من إصدار تناسب العديد من التطبيقات وهذا يجعلها رخيصة الثمن لأنه مع التطبيقات البسيطة لن تكون مضطرا لشراء متحكم معقد وغالى الثمن ، ولكنك ستجد ضمن عائلة بيك أحد المتحكمات البسيطة التى تناسب هذا التطبيق . كل ذلك لأن متحكمات بيك تبدأ من الإصدارات بسيطة الإمكانيات مثل المتحكم 12C508 الذى له ٨ أطراف فقط وتمتد إلى الإصدار 17C766 الذى له ٨٤ طرف ويحتوى الكثير من الإمكانيات الأخرى كما سنرى .

٣-٢ الإصدارات البسيطة Low range, PIC12C5XX

شكل (٣-١) يبين بعض شرائح هذا الإصدار التى تشترك كلها فى الخواص العامة التالية :

- شرائح ذات ٨ أطراف .
- عدد ٣٣ أمر فقط ، كل منها يتكون من ١٢ بت وكلها تنفذ فى دورة ماكينة machine cycle واحدة .
- 512x12 أو 1024x12 مكان ذاكرة برمجة على حسب الشريحة المستخدمة ، لاحظ أن عدد بتات مكان البرمجة هو نفسه عدد بتات الأمر .



شكل (٣-١) الإصدارات البسيطة من عائلة PIC

- عدد سبعة مسجلات ذات أغراض خاصة ، سنعرف وظيفة كل منها بعد قليل .
- مكدسة stack من مستويين فقط ، كل منها ١٢ بت أيضا .
- مسار بيانات من ٨ بت لنقل البيانات بين المسجلات ووحدة الحساب وذاكرة البيانات .
- ذاكرة للبيانات مقدارها ٢٥ أو ٤١ بايت (٨بت) على حسب الشريحة المستخدمة.
- مذبذب ٤ ميجاهرتز داخلى للحصول على نبضات التزامن ، ويمكن توصيل التزامن من مصدر خارجى . دورة الأمر تتكون من ٤ نبضات تزامن ، وهذا يعطى سرعة تنفيذ للأمر الواحد تبلغ ١ ميكروثانية .
- مؤقت ٨ بت .
- مؤقت كلب حراسة watch dog timer داخلى له التزامن الخاص به .
- خمسة أطراف لإدخال وإخراج البيانات .
- جدول (٣-١) يبين مقدار الذاكرة وأنواعها فى كل شريحة من شرائح هذا الإصدار . الذاكرة ROM فى العمود الأوسط هى ذاكرة قراءة فقط تبرمج مرة واحدة ولا يمكن مسحها أو إعادة برمجتها مثل ال EPROM ، وهذه يكتب عليها البرنامج فى الغالب فى المصنع ولمرة واحدة ، وتستخدم فى حالة الإنتاج الكثير من نفس التطبيق حيث أنها تكون رخيصة بالمقارنة مع الأنواع الأخرى . الذاكرة فى العمود الأيمن هى ذاكرة بيانات من النوع EEPROM التى يمكن إعادة الكتابة عليها إلكترونيا (flash memory) وهى ٨ بت لتخزين البيانات المستديمة التى لا تضيع مع انقطاع القدرة مثل الجداول look up tables .

Device	Memory			
	EPROM Program	ROM Program	RAM Data	EEPROM Data
PIC12C508	512 x 12		25	
PIC12C508A	512 x 12		25	
PIC12C509	1024 x 12		41	
PIC12C509A	1024 x 12		41	
PIC12CE518	512 x 12		25	16
PIC12CE519	1024 x 12		41	16
PIC12CR509A		1024 x 12	41	

جدول (٣-١) الذاكرة بأنواعها ومقدارها فى الإصدارات البسيطة للمتحكم PIC

كل هذه الخواص تجعل هذا المتحكم مناسبا للكثير من التطبيقات البسيطة مثل التحكم فى الأجهزة المنزلية ودوائر الإنذار ودوائر التحكم عن بعد remote control ودوائر التوقيت .

إن التركيب الداخلى لهذه العائلة من المتحكمات يختلف عن باقى المتحكمات وحتى عن باقى المعالجات . فى كل المعالجات والمتحكمات الأخرى الشهيرة يوجد مسار واحد للبيانات تنتقل عليه البيانات وشفرات الأوامر وهذا المسار ٨ بت أو ٨ خطوط فى المعالجات والمتحكمات ٨ بت . فى المتحكمات PIC يوجد مسار خاص بالبيانات مكون من ٨ بت ، ومسار خاص بالبرامج أو أكواد الأوامر يتكون من ١٢ بت فى

الإصدارات البسيطة . لذلك توجد ذاكرة خاصة بالبيانات وأخرى خاصة بالبرامج يتم التعامل مع كل منها من خلال المسار الخاص به . هذه الطريقة فى التركيب الداخلى تسمى طريقة هارفارد Harvard architecture ، أما الطريقة الأولى ، مسار واحد للبيانات والبرامج ، وهى الطريقة الشائعة فتسمى طريقة نيومان Neumann architecture . استخدام طريقة هارفارد فى التنظيم الداخلى لعائلة PIC أتاح أن تكون شفرات الأوامر كلها من مقطع واحد ، وهذا المقطع مكون من ١٢ بت فى الإصدارات البسيطة وأتاح أيضا أن تنفذ كل الأوامر فى دورة أمر واحدة فقط ما عدا اوامر القفز فإنها تحتاج ٢ دورة أمر .

تحتوى متحكمات PIC على وحدة حساب ومنطق ALU تقوم بتنفيذ أوامر الحساب والمنطق على مسجل التشغيل Working register, W (وهو يقابل المركم Accumulator فى المعالجات والمتحكمات الأخرى) وأى مسجل آخر . وحدة الحساب والمنطق ومسجل التشغيل تتعامل مع بيانات من ٨ بت . سنتكلم عن هذه المسجلات بالتفصيل فيما بعد .

٣-٣ وظيفة الأطراف المختلفة فى المتحكمات 12C5XX

كما رأينا فى شكل (٣-١) فإن هذه الشريحة لها ٨ أطراف ، وسنرى وظيفة كل طرف من هذه الأطراف فى هذا الجزأ كما يلى :

الطرف ٧ GP0: هذا الطرف يستخدم كطرف إدخال أو إخراج للبيانات . يستخدم أيضا كطرف إدخال للبيانات التتابعية فى حالة برمجة المتحكم ، هذه البيانات بالطبع تمثل البرنامج . يمكن إضافة مقاومة وصل مع Vcc (Pull up resistance) فى حالة استخدامه كخط إخراج للبيانات ، ويتم ذلك برمجيا بتنشيط أحد الأعلام . يمكن استخدام هذا الطرف أيضا لإيقاظ المتحكم wake up من أى حالة نوم يدخل فيها حيث يستيقظ المتحكم عند حدوث أى تغيير فى جهد هذا الطرف .

الطرف ٦ GP1: هذا الطرف يستخدم كطرف إدخال أو إخراج للبيانات . يستخدم أيضا كطرف إدخال لنبضات التزامن فى حالة برمجة المتحكم . يمكن إضافة مقاومة جذب مع Vcc (Pull up resistance) فى حالة استخدامه كخط إخراج للبيانات ، ويتم ذلك برمجيا بتنشيط أحد الأعلام . يمكن استخدام هذا الطرف أيضا لإيقاظ المتحكم wake up من أى حالة نوم يدخل فيها حيث يستيقظ المتحكم عند حدوث أى تغيير فى جهد هذا الطرف .

الطرف ٥ GP2/T0CK: هذا الطرف يستخدم كطرف إدخال أو إخراج للبيانات . يستخدم أيضا لإدخال نبضات التزامن الخارجية للمؤقت T0 وسيأتى شرح مفصل لهذا المؤقت فيما بعد .

الطرف ٤ GP3/$\overline{MCLR}$/Vpp: خط إدخال فقط للبيانات ، خط إعادة وضع أو تصفير Master clear or reset ، كما يوضع عليه جهد البرمجة أثناء وضع أو كتابة البرامج على الشريحة . فى أثناء تنفيذ البرامج يجب ألا يتجاوز جهد هذا الطرف

جهد القدرة للشريحة Vdd وإلا فإن الشريحة يمكن أن تعتبر نفسها دخلت فى مرحلة حرق البرنامج . عند تشغيل هذا الطرف كمصفر عام للشريحة فإنه يكون نشط مع الجهد المنخفض (صفر) active low . يمكن استخدام هذا الطرف أيضا لإيقاظ المتحكم wake up من أى حالة نوم يدخل فيها حيث يستيقظ المتحكم عند حدوث أى تغيير فى جهد هذا الطرف .

الطرف ٣ GP3/OSC2: هذا الطرف يستخدم كطرف إدخال أو إخراج للبيانات ، أو خط إخراج لنبضات التزامن (المذبذب الداخلى) .

الطرف ٢ GP2/OSC1/CLKIN: هذا الطرف يستخدم كطرف إدخال أو إخراج للبيانات ، فى حالة توصيل مذبذب (كريستال) خارجية توصل بين هذا الطرف والطرف ٣ السابق ، كما يمكن إدخال نبضات تزامن من على هذا الطرف أيضا .

الطرف ١ : طرف جهد القدرة Vdd يتراوح من ٢,٥ فولت حتى ٥,٥ فولت مما يعنى إمكانية تشغيله من حجرين بطارية كل منهما ١,٥ فولت .

الطرف ٨ : طرف الأرضى .

٣-٤ انسيابية الأوامر Instruction Pipelining

سنقدم هنا فكرة جديدة تزيد سرعة تنفيذ الأوامر بدرجة كبيرة جدا وهذه الفكرة مستخدمة فى المتحكمات بيك . تخيل أن أى أمر من أوامر أى معالج أو متحكم يحتاج إلى خمس نبضات تزامن حتى يتم تنفيذه ، بحيث تتم عملية التنفيذ فى خلال الخمس نبضات بالخطوات التالية :

١- إحضار الأمر Fetch Instruction, FI
٢- تشفير الأمر Decode Instruction, DI
٣- إحضار المعاملات Fetch Operand, FO
٤- تنفيذ الأمر Execute Instruction, EX
٥- تخزين النتيجة Write Result, WR

هذه الخطوات الخمس يمكن تنفيذها بالتتابع على أى أمر ، وفى هذه الحالة فإننا سنحتاج إلى خمس نبضات تزامن لكى تتم عملية إحضار وتنفيذ أى أمر حيث يتم تنفيذ كل خطوة فى نبضة تزامن منفصلة . يمكن إسراع هذه العملية باستخدام فكرة انسياب الأوامر كما هى موضحة فى شكل (٣-٢) . نلاحظ من هذا الشكل أن كل أمر تم تقسيمه إلى خمس مراحل بحيث عندما يكون المعالج مشغولا فى تنفيذ مرحلة معينة لأمر معين فإن الأمر التالى يتم تنفيذه أيضا فى نفس الوقت ولكن فى مرحلة أخرى من مراحل التنفيذ . فمثلا عندما يكون الأمر رقم I فى مرحلة التخزين WR فإن الأمر I+1 يكون فى مرحلة التنفيذ EX ، ويكون الأمر I+2 فى مرحلة إحضار المعاملات ، والأمر I+3 فى مرحلة تشفير الأمر ، والأمر I+4 فى مرحلة إحضار الأمر ، وهكذا . أى أنه يكون هناك دائما خمسة أوامر موجودة داخل وحدة التنفيذ كل أمر منها يتم تنفيذ مرحلة معينة منه على حسب موقعه فى تتابع الأوامر داخل

الوحدة . نلاحظ من هذا الشكل أننا سنحصل من وحدة التنفيذ على أمر وقد تم تنفيذه فى نهاية كل نبضة تزامن (أمر لكل نبضة) . أى أن سرعة تنفيذ الأوامر قد زادت بمقدار خمس مرات ويمكن زيادتها أكثر من ذلك بزيادة عدد مراحل تنفيذ الأوامر . أى أن الأوامر تنساب فى مراحل التنفيذ فيما يشبه الأنبوبة أو خط الإنتاج وكل أمر يوجد فى مرحلة تنفيذ معينة كما فى خط إنتاج السيارات مثلا ، ولذلك سميت هذه الطريقة بالانسيابية أو Pipelining .

رقم الأمر	نبضات التزامن							
	1	2	3	4	5	6	7	8
I	FI	DI	FO	EX	WR			
I+1		FI	DI	FO	EX	WR		
I+2			FI	DI	FO	EX	WR	
I+3				FI	DI	FO	EX	WR
I+4					FI	DI	FO	EX
I+5						FI	DI	FO

شكل (٣-٢) الانسيابية Pipelining

من الواضح أنه لكى نستفيد من فكرة الانسيابية فإن جميع الأوامر لابد أن تكون لها نفس الطول أو نفس عدد المراحل ، وكل مرحلة لابد أن تنفذ فى نبضة تزامن واحدة ، فهل هذا محقق فى أوامر المتحكمات ؟ المتحكم بيك تم وضع جميع أوامره بحيث تكون كلها لها نفس الطول وتحتاج لنفس زمن التنفيذ إلا أوامر القفز كما سنرى . الكثير من المتحكمات والمعالجات لها أطوال أوامر مختلفة وتنفذ فى أعداد مختلفة من نبضات الساعة . وهذا يسوقنا إلى تقسيم المعالجات عامة إلى نوعين من حيث مجموعة أوامر كل منها .

النوع الأول يسمى المعالجات ذات مجموعة الأوامر المركبة ،

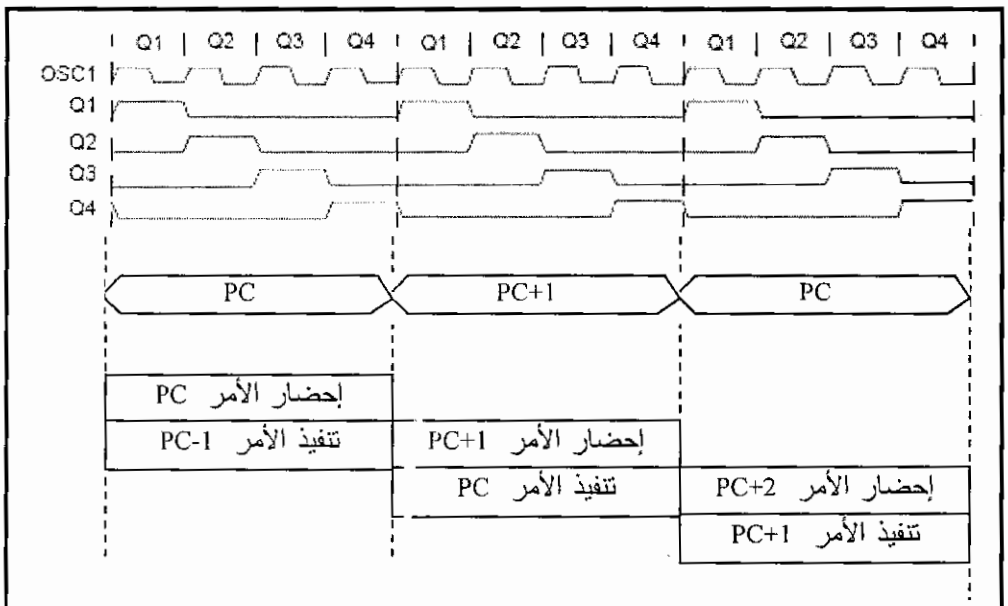
Complex Instruction Set Computers, CISC

النوع الثانى يسمى المعالجات ذات مجموعة الأوامر المخفضة ،

Reduced Instruction Set Computers, RISC

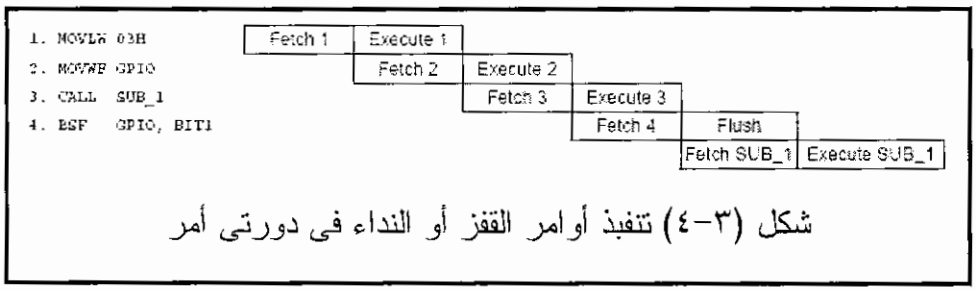
فى النوع الأول من المعالجات ، CISC ، تكون كمية الأوامر كبيرة جدا ، ٣٠٠ أمر على الأقل ويكون معظم هذه الأوامر أوامر مركبة . ولذلك فإن مشفر الأوامر Instruction decoder يكون معقدا جدا مما يتسبب أن الإشارة تأخذ وقتا كبيرا فى تخلل دائرة المشفر ، ولذلك فإن وحدة التحكم فى هذه المعالجات تكون معقدة . أيضا فإن الكثير من الأوامر المركبة يتم تنفيذها بطريقة البرمجيات الصغيرة Microprograms حيث ينفذ أمر الضرب مثلا بتنفيذ برمجية صغيرة تنفذ عملية

الضرب فى صورة مجموعة من أوامر ، وهذا بالطبع يستهلك الكثير من الوقت . كذلك فإنه نتيجة اختلاف أطوال الأوامر فى هذا النوع من المعالجات فإنه يكون من الصعب استخدام فكرة الانسيابية Pipelining . نتيجة لذلك ظهر التفكير فى النوع الثانى من المعالجات RISC حيث يكون كل شيء هنا مخفضا ، عدد الأوامر تم تخفيضه حتى أقل من ١٢٨ أمر ، وكذلك طرق عنونة الذاكرة Memory addressing تم تخفيضها حيث أن التعامل مع الذاكرة يستهلك الكثير من الوقت . مع بساطة عدد الأوامر فإن مشفر الأوامر ، وكذلك وحدة التحكم سيكونان أكثر بساطة مما سيوفر الكثير من وقت التنفيذ . كذلك فإن تخفيض عدد الأوامر سيقتصر على الأوامر ذات الأطوال الواحدة بقدر الإمكان والتى يمكن تنفيذها فى نفس عدد المراحل مما سيسهل استخدام طريقة الانسيابية Pipelining فى تنفيذ هذه الأوامر . مع تخفيض عدد الأوامر وبساطتها فى المعالجات RISC أمكن الاستغناء وبدرجة كبيرة على استخدام البرمجيات الصغيرة فى تنفيذ الأوامر حيث أمكن استخدام دوائر مبنية Hardware لتنفيذ هذه الأوامر مما وفر الكثير من وقت التنفيذ أيضا . كما رأينا فإن أنظمة RISC تتمتع بالكثير من المميزات مما جعل معظم المعالجات ابتداء من المعالج 80486 تقريبا يأخذ بهذه الفكرة وينفذها ، حيث أصبحت كل الأوامر تقريبا تنفذ فى نبضة تزامن واحدة ، وهذا النظام هو ما أخذت به متحكمات بيك .



شكل (٣-٣) دورة الأمر وانسيابية الأوامر Pipelining فى المعالجات بيك

فى المتحكمات بيك تم قسمة نبضات التزامن القادمة سواء من المصدر الداخلى أو الخارجى على ٤ للحصول على ما يسمى بدورة الأمر Instruction cycle التى يتكون كل منها من ٤ نبضات Q1 و Q2 و Q3 و Q4 كما فى شكل (٣-٣) . فى النبضة Q1 يتم زيادة عداد البرنامج PC ليشير للأمر التالى فى التنفيذ . بنهاية النبضة Q4 يتم إحضار الأمر من الذاكرة ووضعه فى مسجل الأوامر Instruction register وتشفيره عن طريق مشفر الأوامر Instruction decoder ، ولكنه لم ينفذ حتى الآن . أى أن إحضار الأمر وتشفيره وزيادة عداد البرنامج يتم فى دورة أمر . أما تنفيذ هذا الأمر فيتم فى دورة الأمر التالية ، إن أى أمر يتم إحضاره فى دورة أمر وتنفيذه فى الدورة التالية كما فى شكل (٣-٣) . كما نلاحظ من شكل (٣-٣) فإن كل أمر يتم إحضاره فى دورة أمر ، ثم ينفذ فى دورة الأمر التالية ، كما أن جميع الأوامر يتم تنفيذها فى دورة أمر واحدة ما عدا أوامر القفز أو النداء على برامج فرعية ، حيث فى هذه الحالة بعد إحضار الأمر ووضعه فى مسجل الأوامر وتشفيره يتم تفريغه flushing من محتوياته ليحضر مكانه الأمر الذى سيتم القفز إليه . شكل (٣-٤) يبين كيف أن أوامر القفز أو النداء سيتم تنفيذها فى دورتى أمر بدلا من واحدة كباقى الأوامر وذلك من خلال تنفيذ برنامج بسيط أحد أوامره هو نداء على برنامج فرعى حيث عند تنفيذ هذا الأمر تم تفريغ مسجل الأوامر وتحميله بأول أمر فى البرنامج الفرعى .



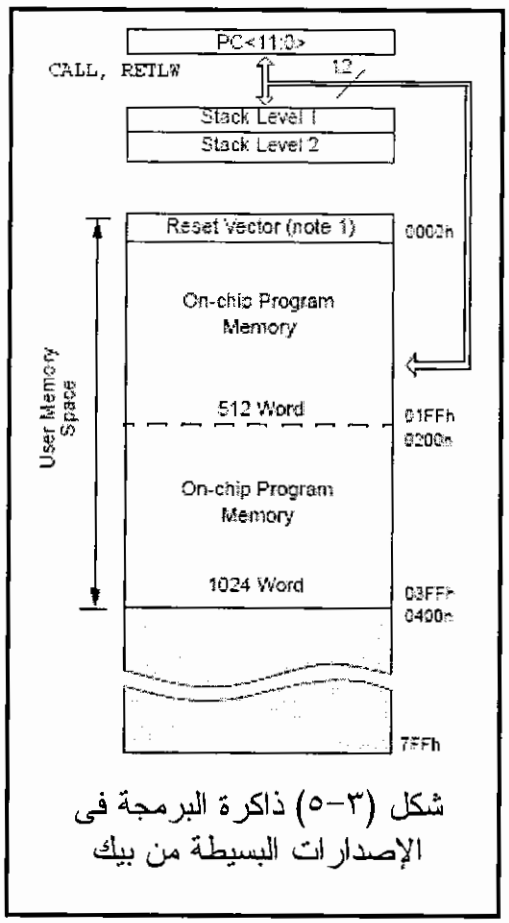
شكل (٣-٤) تنفيذ أوامر القفز أو النداء فى دورتى أمر

٣-٥ ذاكرة البرمجة فى المتحكمات بيك البسيطة

ذاكرة البرمجة فى هذا النوع من المتحكمات تكون إما ٥١٢×١٢ بايت (صفحة page واحدة) أو ١٠٢٤×١٢ بايت (صفحتان) ، أى أن البايت هنا تتكون من ١٢ بت إذا جاز لنا أن نطلق عليها بايت . أى أن جميع أوامر هذا الإصدار من بيك تتكون من ١٢ بت . عداد البرنامج PC يتكون أيضا من ١٢ بت ، أى أنه يمكنه التعامل مع ذاكرة مقدارها ٤ كيلوبايت×١٢ بت ، ولكن الموجود فقط داخل الشريحة هو ٥١٢ أو ١٠٢٤ بايت على حسب الإصدار . عداد البرنامج يحتوى عنوان الأمر التالى فى التنفيذ كما أشرنا سابقا . شكل (٣-٥) يبين رسم صندوقى للذاكرة فى هذه

المتحكمات . نلاحظ من شكل (٣-٥) أن المكدسة stack تتكون من مكانين فقط ، عرض كل منهما أيضا يتكون من ١٢ بت وسيأتى الحديث عن المكدسة فيما بعد . آخر مكان فى الذاكرة (رقم ٥١٢ أو رقم ١٠٢٤ على حسب الإصدار) يحتوى بيانات معايرة تردد المذبذب الداخلى للمتحكم ولذلك يجب عدم التسجيل فيه . عند عمل تصفير reset للمتحكم فإنه يذهب لهذا المكان ليقرأ محتوياته ثم يذهب للعنوان 0000H ليبدأ التنفيذ من هناك كما فى الشكل .

٣-٦ ذاكرة البيانات RAM فى المتحكمات بيك البسيطة

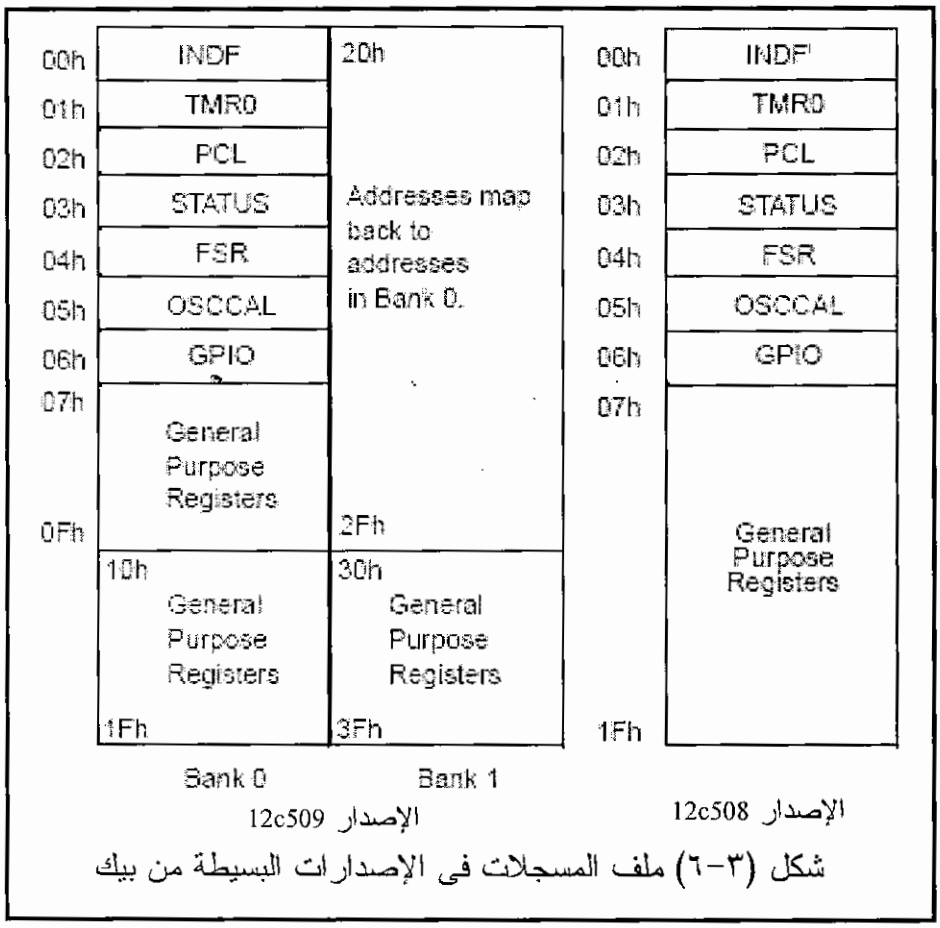


شكل (٣-٥) ذاكرة البرمجة فى الإصدارات البسيطة من بيك

ذاكرة البيانات فى هذا النوع من المتحكمات تتكون من مجموعة من البايتات أو المسجلات التى يطلق عليها فى العادة ملف المسجلات register file. ملف المسجلات يتكون من ٣٢ مسجل فى بعض الإصدارات (12c508, 12c508A) أو ٦٤ مسجل مقسمة على بانكين أو مجموعتين فى بعض الإصدارات الأخرى (12c509, 12c509A) . شكل (٣-٦) يبين شكل توضيحى لملف المسجلات فى كل من الإصدارين .

يمكن تقسيم ملف المسجلات فى كل من الإصدارين كما فى شكل (٣-٦) إلى قسمين ، الأول هو مجموعة من المسجلات ذات الوظائف الخاصة special function registers, SFR وتبدأ من العنوان 00h حتى العنوان 07h فى كل من الإصدارين وسيأتى الحديث عن كل منها على حده فيما بعد . القسم الثانى من المسجلات وهو المسجلات العامة general purpose registers وتبدأ من العنوان 08h حتى العنوان 1Fh وعددها ٢٥ مسجلا تستخدم فى أغراض البرمجة . كما فى شكل (٣-٦) فإن الإصدار 12c509 يحتوى نفس ملف المسجلات كما فى الإصدار 12c508 بالإضافة إلى مجموعة مماثلة حيث تسمى المجموعة الأولى بانك صفر Bank 0 والمجموعة الثانية تسمى بانك ١ Bank 1 كما فى الشكل ، حيث تمتد العناوين هنا إلى 3Fh . لاحظ أن عناوين المسجلات الخاصة هى نفسها فى الإصدارين بحيث أن البرامج المكتوبة لأى

إصدار تنفذ على الإصدار الآخر دون أى مشكلة . هذه المسجلات العامة يمكن التعامل معها بالطريقة المباشرة أو الغير مباشرة كما سنرى بالتفصيل . سنرى الآن وظيفة كل واحد من المسجلات الخاصة :



شكل (٣-٦) ملف المسجلات فى الإصدارات البسيطة من بيك

٣-٦-١ مسجل الحالة Status register, STATUS

هذا المسجل عنوانه هو 03h وهو يحتوى مجموعة أعلام flags حسابية تبين حالة المتحكم بعد كل عملية حسابية بالإضافة إلى بعض الأعلام الأخرى التى سيأتى شرحها الآن وكما هو موضح فى شكل (٣-٧) . هذا الشكل يبين إذا كانت كل بت أو علم يمكن قراءتها أو الكتابة فيها وحالة كل علم من الأعلام عند عمل تصفير reset للمتحكم :

- **العلم GPWUF** يمكن قراءته والكتابة فيه ، كما أنه يكون صفر عند تصفير المتحكم ، وهذا العلم يكون ١ عند صحيان المتحكم من حالة نوم نتيجة تغير الجهد

على أى طرف من أطراف الإدخال أو الإخراج . يكون صفر بعد تشغيل القدرة power واستقرارها .

- **العلم PA0** علم اختيار صفحة من صفحات ذاكرة البرمجة حيث الصفحة تتكون من ٥١٢ بايت . PA0=0 يعنى التعامل مع الصفحة صفر أو أول ٥١٢ بايت من العنوان 000h حتى العنوان 01FFh وذلك بالطبع للمتحكمات 12c508 . PA0=1 يعنى التعامل مع الصفحة واحد أو ثانى ٥١٢ بايت من العنوان 200h حتى العنوان 03FFh وذلك بالطبع للمتحكمات 12c509 .
- **العلم Time Out, $\overline{TO}$** وهو علم الأنتهاء من زمن تأخير ، حيث يكون واحد عند رجوع القدرة ، أو تنفيذ الأمر sleep للدخول فى النوم ، أو تنشيط أو بداية تشغيل مؤقت كلب الحراسة WDT . يكون صفر (نشط) عند تصفير مؤقت كلب الحراسة أى نفاذ زمن التأخير المخصص له .

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
GPWUF	—	PA0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
bit7	6	5	4	3	2	1	bit0

R = Readable bit
W = Writable bit
- n = Value at POR reset

شكل (٣-٧) مسجل الحالة

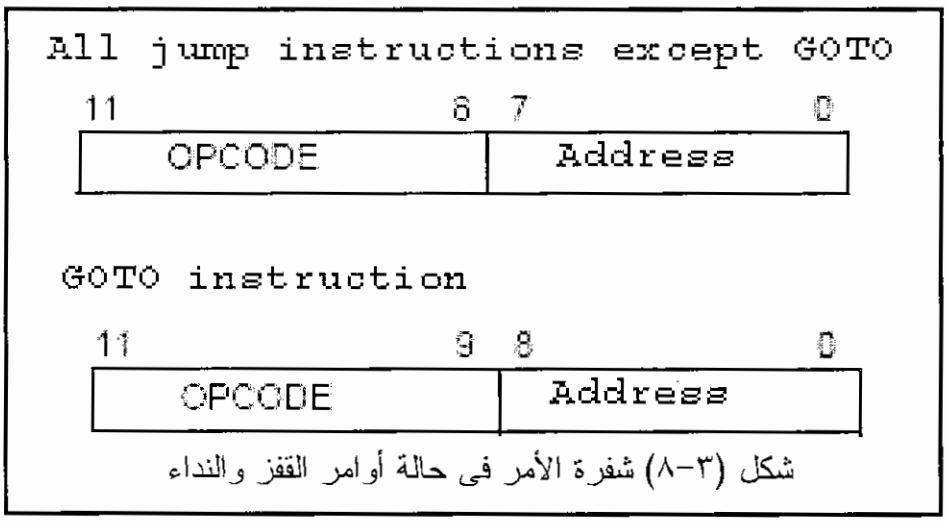
- **العلم Power Down, $\overline{PD}$** وهو علم بيان حالة القدرة على الشريحة حيث يكون واحد عند استقرار القدرة أو تصفير مؤقت كلب الحراسة ، ويكون صفر فى حالة تنفيذ الأمر sleep .
- **العلم Zero flag, Z** ، هذا العلم يكون واحد إذا كانت آخر عملية حسابية أو منطقية نفذها المتحكم تساوى صفر ، ويكون صفر فيما عدا ذلك .
- **العلم Digit Carry, DC** ، وهو علم الحمل النصفى Half carry flag ، وهذا العلم يكون واحد إذا كان هناك حمل من البت الثالثة للرابعة بعد إجراء عملية جمع ، أو استلاف من البت الرابعة للثالثة فى حالة إجراء عملية طرح . أى أن عملية الحمل أو الاستلاف هنا تكون بين الخانة الأولى للثانية عند التعامل مع الأرقام الستعشرية ويكون صفر فيما عدا ذلك .
- **العلم Carry, C** ، وهو علم الحمل Carry flag الذى يكون واحد عند حدوث حمل من آخر بت (السابعة) فى حالة إجراء عملية جمع ، أو استلاف إليها فى حالة إجراء عملية طرح . يكون صفر فيما عدا ذلك .

لاحظ أن البت السادسة فى مسجل الأعلام غير مستخدمة كما فى شكل (٣-٧) .

٣-٦-٢ مسجل ماسك عداد البرنامج Program Counter Latch, PCL

عداد البرنامج PC كالعادة يحتوى عنوان الأمر الذى عليه الدور فى التنفيذ ، وهو يتكون من ١٢ بت وبذلك يمكنه التعامل مع كمية من ذاكرة البرمجة تصل إلى ٤ كيلو . تذكر أن متحكمات بيك البسيطة التى معنا الآن تحتوى إما ٥١٢ بايت (صفحة واحدة) أو ١٠٢٤ بايت (صفحتان) ذاكرة برمجة فقط . كما نعلم أيضا فإن كل أوامر البيك تتكون من بايت واحدة ، لذلك فإنه مع تنفيذ كل أمر فإن عداد البرنامج يزداد بمقدار واحد فقط إلا مع الأوامر التى تغير من محتويات هذا العداد مثل أوامر القفز أو النداء على البرامج الفرعية .

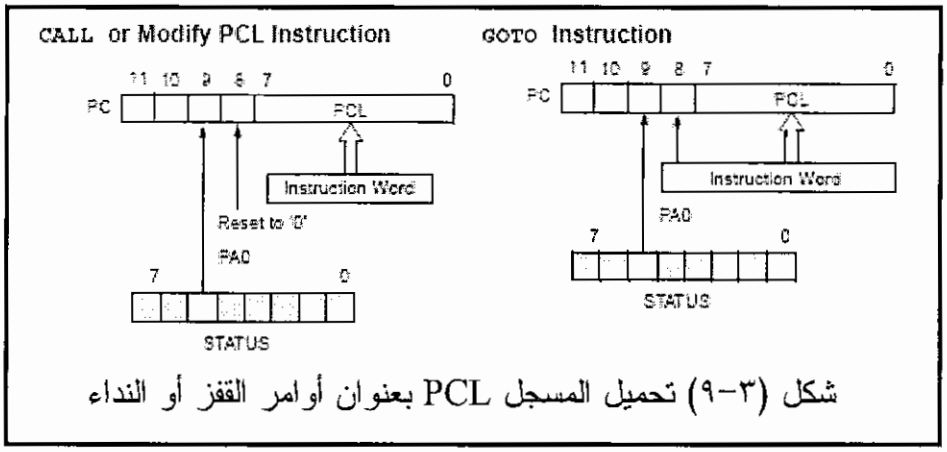
فى أوامر القفز أو النداء يحتوى الأمر نفسه على العنوان الذى سيتم القفز إليه . شكل (٣-٨) يوضح أن شفرة أوامر القفز كلها ما عدا الأمر GOTO تحتوى ٨ بت مخصصة للعنوان الذى سيتم القفز إليه ، بينما الأمر GOTO فيخصص ٩ بت للعنوان الذى سيتم القفز إليه كما فى الشكل .



شكل (٣-٨) شفرة الأمر فى حالة أوامر القفز والنداء

المسجل PCL الذى عنوانه 02h يوضع به عنوان الأمر الذى سيتم القفز إليه فى حالة أوامر القفز أو النداء كما فى شكل (٣-٩) . نلاحظ من هذا الشكل أنه فى حالة أوامر النداء أو القفز ما عدا الأمر GOTO فإن الثمانية بتات الخاصة بالعنوان تأتى من شفرة الأمر وتوضع فى المسجل PCL وتوضع البت رقم ٨ فى عداد البرنامج بصفر بينما تأتى البت التى تحدد الصفحة التى يتم التعامل معها من مسجل الحالة . فى حالة الأمر GOTO هناك كما ذكرنا ٩ بتات تمثل عنوان القفز ، ثمانية منها توضع فى المسجل PCL والتاسعة توضع فى بت ٨ من عداد البرنامج ، كذلك تأتى البت التى تحدد الصفحة من مسجل الأعلام كما فى شكل (٣-٩) . المسجل PCL يمكن التعامل مع محتوياته من خلال أوامر المتحكم كما سنرى عند دراسة هذه الأوامر .

لاحظ أن ذلك يعنى أن الأمر GOTO يمكن أن يقفز لأى مكان فى الصفحة (٥١٢ بايت) بينما باقى أوامر القفز والنداء فتقفز إل ٢٥٦ مكان فقط .
عند عمل تصفير reset للمتحكم فإن عداد البرنامج يصبح كله وحايد حيث يقفز لآخر مكان فى ذاكرة البرمجة وينفذ الأمر الموجود هناك وهو أمر خاص بمعايرة المذبذب وبعدها ينتقل لبداية ذاكرة البرمجة ليبدأ تنفيذ الأوامر الموجودة هناك .



شكل (٣-٩) تحميل المسجل PCL بعنوان أوامر القفز أو النداء

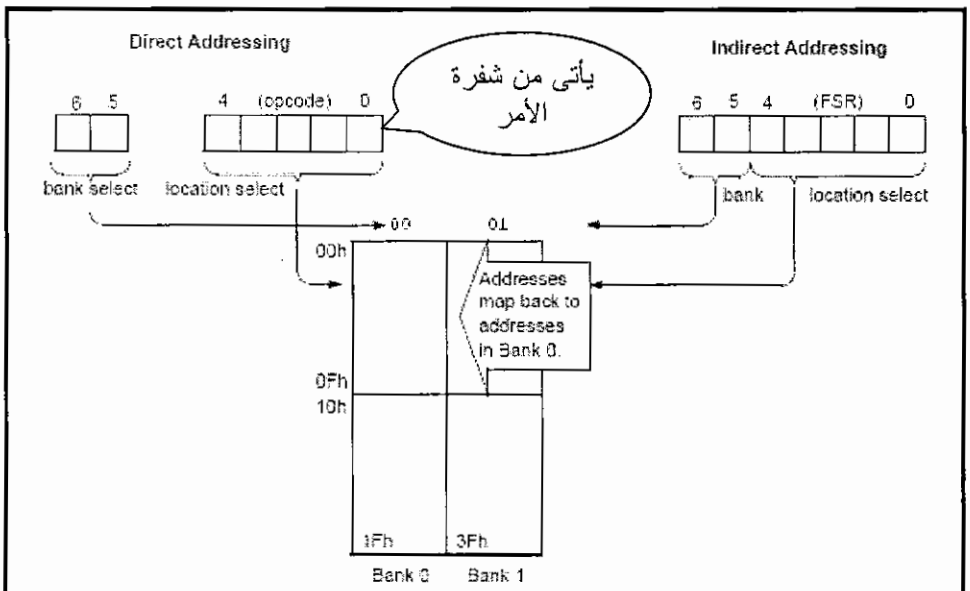
٣-٦-٣ المسجلان FSR, INDF

المسجل File Select Register, FSR يستخدم مع المسجل INDF للتعامل الغير مباشر مع ملف المسجلات . شكل (٣-١٠) يبين طريقة التعامل المباشر وغير المباشر مع ملف المسجلات . فى الطريقة المباشرة يكون عنوان المسجل موجود مباشرة فى الأمر نفسه . أما فى الطربقة الغير مباشرة فإن عنوان المسجل المراد التعامل معه يوضع فى المسجل FSR ويستخدم المسجل INDF فى الأمر للدلالة على أن هذا التعامل من النوع الغير مباشر . فمثلا الأمر CLRF INDF معناه تصفير المسجل الذى يوجد عنوانه فى المسجل FSR .
لاحظ أن المسجل INDF مسجل غير موجود أصلا داخل الشريحة ، ومحاولة القراءة منه تعطى أصفارا ، ومحاولة الكتابة فيه لا تسجل أى نتيجة أى كما لو نفذنا أمر لا يعمل شىء ، وهو موجود فقط للتعامل الغير مباشر مع ملف المسجلات بالتعاون مع المسجل FSR كما سنرى . المسجل FSR مستخدم منه ٥ بت فقط من بت 0 حتى بت ٤ للإشارة الغير مباشرة فى ملف المسجلات فى حالة المتحكم 12c508 أما فى حالة المتحكم 12c509 فتستخدم بت إضافية (البت رقم ٥) للإشارة للبانك الذى يتم التعامل معه . باقى بتات المسجل (بت ٦ و بت ٧) غير مستخدمة وعند قراءتها تقرأ القيمة واحد ..

مثال ٣-١ :
تصفير المسجلات التى عناوينها 10h حتى 1Fh بالطريقة الغير مباشرة .

```
        MOVLW  0x10
        MOVWF  FSR
Next    CLRF    INDF
        INCF    FSR,F
        BTFSC   FSR,4
        GOTO    Next
Continue
```

الأمر الأول يضع الثابت أو القيمة الفورية 10h فى المسجل W ، ثم ينقل هذه القيمة من المسجل W إلى المسجل fsr كما فى الأمر الثانى وبذلك أصبح المسجل fsr يشير على العنوان 10h . الأمر الثالث clrf indf يصفر بطريقة غير مباشرة المسجل الذى عنوانه فى المسجل fsr . الأمر الرابع يزيد محتويات المسجل fsr بمقدار واحد لتشير إلى العنوان التالى . الأمر الخامس bitt test and skip if clear, BTFSC يختبر البت الرابعة ليرى إذا كانت صفر يهمل الأمر التالى ولن ينفذ الأمر GOTO حيث عندها تكون محتويات المسجل fsr تساوى 20h ويستمر فى باقى البرنامج . أما إذا كانت البت الرابعة تساوى صفر فإنه ينفذ الأمر goto ويرجع إلى العلامة Next حيث يستمر فى تنفيذ الحلقة .



شكل (٣-١٠) التعامل المباشر وغير المباشر مع ذاكرة البيانات

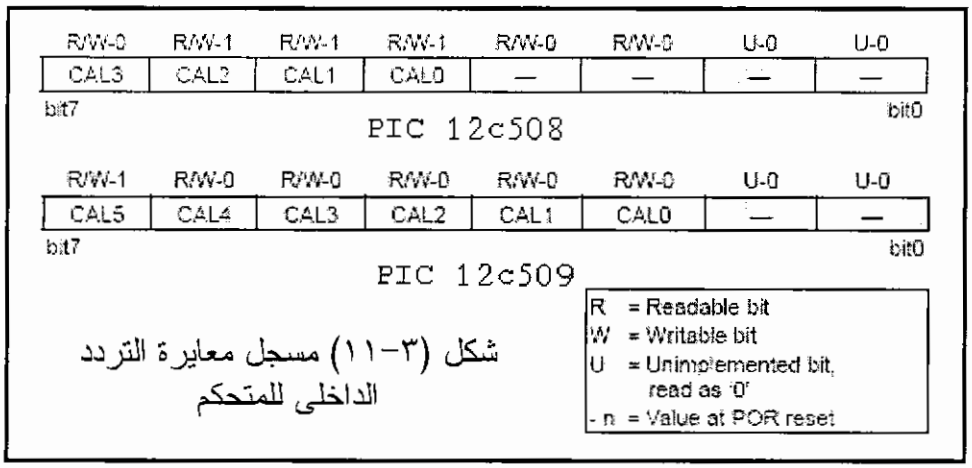
٣-٦-٤ المسجل Oscillator Calibration, OSCCAL

عنوانه هو 05h وهذا المسجل مستخدم منه ٤ بت فقط (بت ٤ حتى بت ٧) فى المتحكم 12c508 و ٦ بت فقط (من بت ٢ حتى بت ٧) فى المتحكم 12c509 ، وأما باقى البتات فغير مستخدمة . عند تشغيل المتحكم باستخدام المذبذب الداخلى (٤

ميجاهرتز) فإنه يمكن معايرة تردد هذا المذبذب بزيادته قليلا باستخدام الرقم الذى يوضع فى بتات هذا المسجل . شكل (٣-١١) يبين بتات هذا المسجل .

٣-٦-٥ المسجل GPIO وإدخال وإخراج البيانات

مسجل إدخال وإخراج البيانات وعنوانه هو 06h ، ويتكون من ٨ بت مستخدم منها ستة فقط لأن هذا الإصدار لا يحتوى إلا ستة خطوط لإدخال وإخراج البيانات كما رأينا ومنها الخط GP3 الذى يستخدم لإدخال البيانات فقط . البيانات المراد إخراجها توضع فى هذا المسجل . عند تحديد أى طرف ليعمل كطرف إخراج ، فإن البيانات المخرجه يتم مسكها على هذا الطرف إلى أن يتم الكتابة عليها مرة ثانية . عند قراءة هذه الأطراف باستخدام أمر مثل MOVF GPIO,W فإن الإشارة الموجودة على جميع الأطراف يتم قراءتها سواء كانت هذه الأطراف محددة مسبقا كأطراف دخل أو خرج .



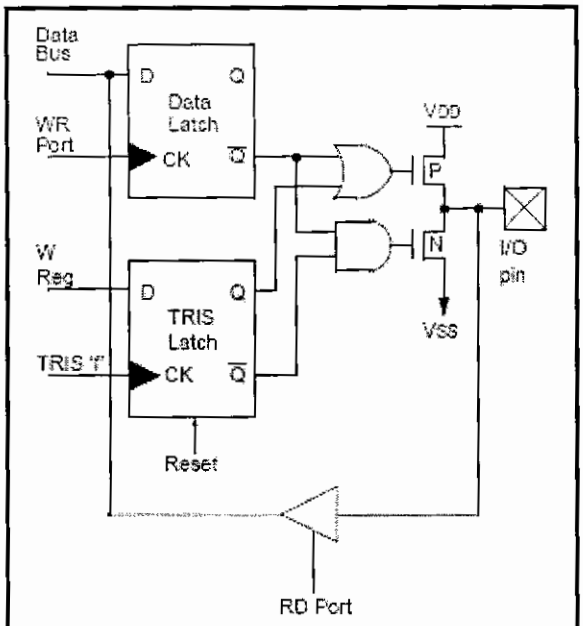
شكل (٣-١١) مسجل معايرة التردد الداخلى للمتحكم

٣-٦-٦ المسجل TRIS

هو أحد المسجلات الغير معنونة والذى يستخدم لتحديد أى الأطراف يكون دخل وأيها يكون خرج . بتسجيل واحد فى أى بت من بتات هذا المسجل فإن طرف الإدخال/الإخراج المقابل له سيكون طرف دخل ، وبتسجيل صفر فى بت من بتات المسجل TRIS فإن الطرف المقابل له سيكون طرف خرج . الأمر TRIS f حيث f=6 يحمل محتويات المسجل W فى المسجل TRIS .

شكل (٣-١٢) يبين دور المسجل TRIS فى تحديد إذا كان الطرف سيكون دخل أم خرج . نلاحظ من هذا الشكل أنه بوضع Q=1 فى أى ماسك فى المسجل TRIS فإن $\overline{Q} = 0$ وبالتالى فإن خرج البوابة آند يكون صفر وبالتالى سيكون الترانزستور الموصل عليها مفتوح open circuit ، وكذلك طالما أن Q=1 فإن خرج بوابة الأور يكون واحد ويكون الترانزستور الموصل عليها مفتوح أيضا . وعلى ذلك فإن طرف

الشريحة المقابل سيكون مفصولا open circuit وبالتالى يمكن قراءته بأمر القراءة الذى ينشط عازل الدخل عن طريق تنشيط الخط RD port كما فى الشكل .
بوضع Q=0 فى الماسك TRIS Latch فإن الطرف المقابل سيكون خرج يخرج البيانات الموجودة على ماسك البيانات Data Latch ، فإذا كانت هذه البيانات تساوى صفر سيتم إخراج صفر على الطرف المقابل ، بينما إذا كان ماسك البيانات يحتوى واحد ، فإن الطرف المقابل يصبح واحد . حاول متابعة ذلك فى شكل (٣-١٢) . عند عمل تصفير reset للشريحة فإن جميع خطوط إدخال/إخراج البيانات تكون خطوط إدخال .



شكل (٣-١٢) إدخال وإخراج البيانات

٣-٦-٧ المسجل TMR0 أو المؤقت T0 وملحقاته

تتم عملية التوقيت فى هذا المتحكم من خلال المسجل TMR0 الذى يتكون من ٨ بت والذى يستخدم كمؤقت أو عداد كما سنرى . هناك أيضا عداد من ٨ بت يستخدم لضبط أو قاسم لساعة المؤقت prescaler والمسجل Option الذى يستخدم فى ضبط أداء المؤقت .
المسجل TMR0 يمكن استخدامه كمؤقت يعد النبضات الداخلية والتى ترددها عبارة عن نبضة كل دورة أمر machine or instruction cycle والتى تساوى ربع نبضات التزامن Fosc/4 . فى حالة استخدامه كعداد فإنه يعد نبضات تزامن يتم إدخالها من خارج الشريحة على الطرف ٥ ، GP2/T0CKL . يتم ضبط أداء المؤقت أو العداد من خلال مجموعة من الأعلام الموجودة فى المسجل Option الذى سنقدمه فى الجزء التالى .

٣-٦-٨ المسجل Option

هذا المسجل غير معنون ، أى أنه ليس من ضمن المسجلات الخاصة الموجودة فى شكل (٣-٦) ولكن يتم التعامل معه من خلال الأوامر واسمه فقط باستخدام الأمر OPTION الذى يحمل هذا المسجل بمحتويات مسجل التشغيل W وهذا المسجل

يمكن الكتابة فيه فقط ولا يمكن قراءته . شكل (٣-١٣) يبين بتات هذا المسجل ، وأما وظيفة كل بت من هذه البتات فهى كما يلى :

البت ٧ $\overline{GPWU}$ لتنشيط خاصية الصحيان من النوم عند حدوث أى تغير فى جهد أطراف الإدخال GP0, GP1, GP3 . هذه المتحكمات يمكن إدخالها فى حالة نوم عن طريق الأمر SLEEP حيث يتوقف المتحكم عن تنفيذ البرامج وذلك لتخفيض القدرة . إذا دخل المتحكم فى حالة النوم فإنه يمكن إخراجه منها عن طريق حدوث أى تغير فى جهد أطراف الدخل السابق ذكرها . هذه الخاصية ، إيقاظ المعالج بتغير جهد أطراف الدخل كما ذكرنا ، تنشط بوضع البت ٧ فى المسجل Option تساوى صفر ، بوضع هذه البت تساوى واحد فإن هذه الخاصية لن تكون نشطة .

W-1	W-1	W-1	W-1	W-1	W-1	W-1	W-1
$\overline{GPWU}$	$\overline{GPPU}$	T0CS	T0SE	PSA	PS2	PS1	PS0
bit7	6	5	4	3	2	1	bit0

W = Writable bit
U = Unimplemented bit
- n = Value at POR reset

شكل (٣-١٣) المسجل Option

البت ٦ $\overline{GPPU}$ ، عند استخدام أحد أطراف إخراج البيانات فإنه يمكن أن نضيف له مقاومة توصل مع Vcc لتغذية الأحمال الخارجية تسمى Pull up resistance بوضع هذه البت تساوى صفر فإنه يتم توصيل هذه المقاومة على الأطراف المستخدمة كخرج ، وبوضع هذه البت بصفر فإن هذه الخاصية لن تكون نشطة .

Bit Value	Timer0 Rate	WDT Rate
000	1 : 2	1 : 1
001	1 : 4	1 : 2
010	1 : 8	1 : 4
011	1 : 16	1 : 8
100	1 : 32	1 : 16
101	1 : 64	1 : 32
110	1 : 128	1 : 64
111	1 : 256	1 : 128

جدول (٣-١) ضبط نسبة القسمة للمؤقت T0 أو مؤقت الحراسة .

البت ٥ T0CS ، هذه البت يتم عن طريقها اختيار مصدر نبضات التزامن للمؤقت T0 . بوضعها تساوى صفر فإن المؤقت يعمل على النبضات الداخلية والتى تساوى ربع نبضات الساعة Foscl/4 . بوضع هذه البت تساوى واحد فإن المؤقت يعمل على النبضات المدخلة له من الخارج على الطرف ٥ .

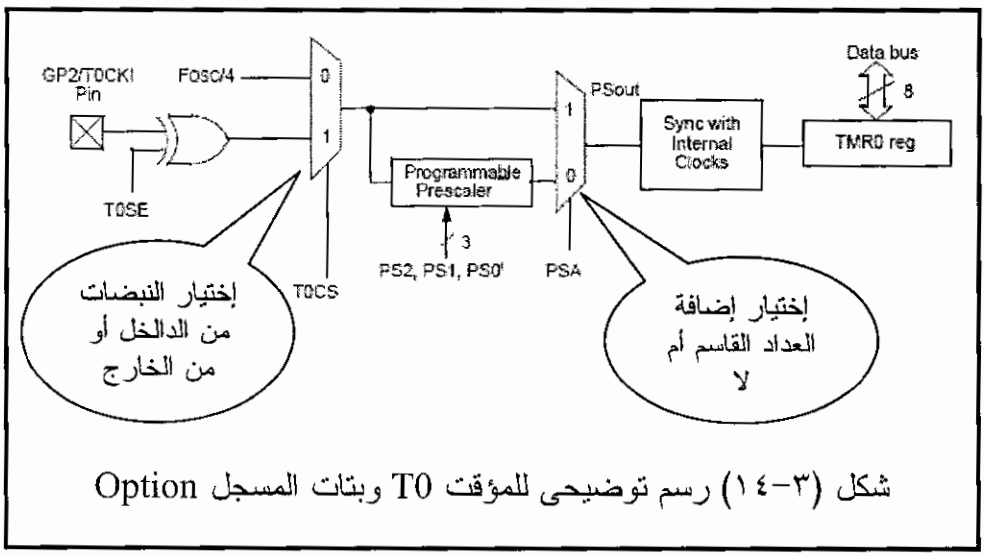
البت ٤ T0SE ، عند إدخال نبضات تزامن خارجية من على الطرف ٥ ليعمل عليها المؤقت T0 ، فإنه يمكن اختيار الحافة الفعالة لهذه النبضات عن طريق هذا العلم . بوضع هذه البت تساوى واحد فإن المؤقت سيعمل عند الحافة النازلة (واحد إلى صفر) من نبضات التزامن الخارجية ،

وعند وضع هذه البت بصفر فإن المؤقت سيعمل عند الحافة الصاعدة (صفر إلى واحد) لهذه النبضات .

البت ٣ PSA ، Prescaler assignment ، هناك عداد كما ذكرنا داخل المتحكم يستخدم كقاسم لنبضات التزامن حيث يتم قسمة نبضات التزامن على النسبة المضبوط عليها هذا العداد لإطالة زمن التوقيت . هذا العداد يمكن استخدامه مع المؤقت T0 أو مع مؤقت الحراسة WDT ولا يمكن استخدامه مع الإثنين معا فى وقت واحد . البت ٣ فى المسجل Option عند وضعها تساوى واحد فإن هذا العداد يستخدم كقاسم لنبضات تزامن مؤقت الحراسة WDT ، وعند وضعها بصفر فإن هذا العداد يستخدم كقاسم للمؤقت T0 .

البتات ٢ ، ١ ، وصفر ، هذه البتات تستخدم لضبط نسبة القسمة التى سيعل عندها العداد السابق . جدول (٣-١) يبين كيفية اختيار هذه النسبة سواء مع المؤقت T0 أو مع مؤقت الحراسة . لاحظ من هذا الجدول كيف يمكن إطالة زمن تأخير المؤقت T0 حتى نسبة ٢٥٦ مرة .

شكل (٣-١٤) يبين عملية تخصيص العداد القاسم prescaler للمؤقت T0 أو لمؤقت الحراسة وتأثير الأعلام المختلفة فى المسجل Option على أداء المؤقت .

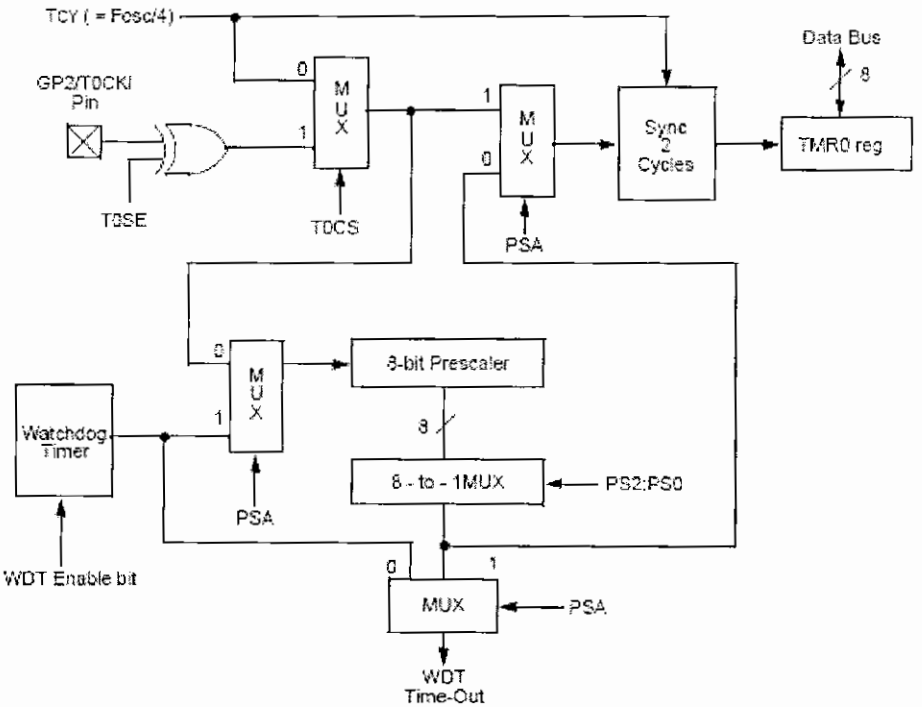


شكل (٣-١٤) رسم توضيحى للمؤقت T0 وبتات المسجل Option

لاحظ وجود فرصتى اختيار فى هذا الشكل ، الأولى نختار بها نبضات تزامن المؤقت T0 إما من الداخل (T0cs=0) أو من الخارج (T0cs=1) . الاختيار الثانى هو إما نضم العداد القاسم إلى T0 (PSA=0) ، أو ينضم إلى مؤقت الحراسة (PSA=1) . شكل (٣-١٥) يبين عملية تخصيص العداد القاسم هذه بتوضيح أكثر من شكل (٣-١٤) .

عندما يكون العداد القاسم مخصصا للمؤقت T0 فإن أى عملية كتابة فى محتويات هذا المؤقت T0 ستسبب تصفيرا للعداد القاسم . يمكن الكتابة فى المؤقت T0 بأوامر مثل CLRF 1 ، الذى يصفر محتويات المؤقت T0 الذى عنوانه هو 01 ، والأمر MOVWF 1 الذى ينقل محتويات المسجل W إلى T0 ، والأمر BSF 1,x الذى يضع البت رقم x فى المسجل T0 بواحد .

عندما يكون العداد القاسم مخصصا لمؤقت الحراسة WDT فإن أى عملية كتابة فى محتويات هذا المؤقت WDT ستسبب تصفيرا للعداد القاسم . يمكن الكتابة فى المؤقت WDT بالأمر CLRWDT الذى يصفر محتويات مؤقت الحراسة وبالتالى يصفر العداد القاسم .



شكل (٣-١٥) رسم تفصيلى لتخصيص العداد القاسم للمؤقت T0 أو مؤقت الحراسة وعلاقة ذلك بالمسجل Option

عملية تخصيص العداد القاسم للمؤقت T0 أو مؤقت الحراسة WDT تتم عن طريق البرمجة بعدد من الأوامر . مجموعة الأوامر التالية تنقل تبعية القاسم من المؤقت T0 إلى مؤقت الحراسة :

```
CLRWDT
CLRF  TMR0
```

```
MOVLW 'xxxx1xxx'b
OPTION
```

مجموعة الأوامر التالية تنقل تبعية القاسم إلى المؤقت T0 من مؤقت الحراسة :

```
CLRWDT
CLRF TMR0
MOVLW 'xxxx0xxx'b
OPTION
```

٣-٦-٩ ضبط مكونات المتحكم Configuration word

هذه الكلمة word تشغل آخر مكان فى ذاكرة البرمجة . أول ٢ بت فيها (بت 0 وبت 1) كما فى شكل (٣-١٦) يمكن بها اختيار واحد من ٤ أنواع للمذبذبات كما سنرى . البت ٢ عند وضعها بواحد يتم تنشيط مؤقت الحراسة ، وعند وضعها بصفر لا ينشط مؤقت الحراسة بل يكون مخمدا . البت ٣ تستخدم لحماية البرنامج الموجود فى ذاكرة البرمجة حيث بوضع هذه البت بواحد يكون البرنامج محمى وبوضعها بصفر يكون البرنامج غير محمى . البت ٤ هى بت تنشيط للطرف $\overline{MCLR}$ فعند وضع هذه البت بصفر يتم تنشيط طرف التصفير $\overline{MCLR}$ ، وعند وضعها بواحد لا يكون هذا الطرف نشطا . شكل (٣-١٦) يبين بتات هذه الورد . لاحظ أنها مكونة من ١٢ بت لأنها أحد بايتات ذاكرة البرمجة ، ومستخدم منها ٥ بت فقط التى تم ذكرها .

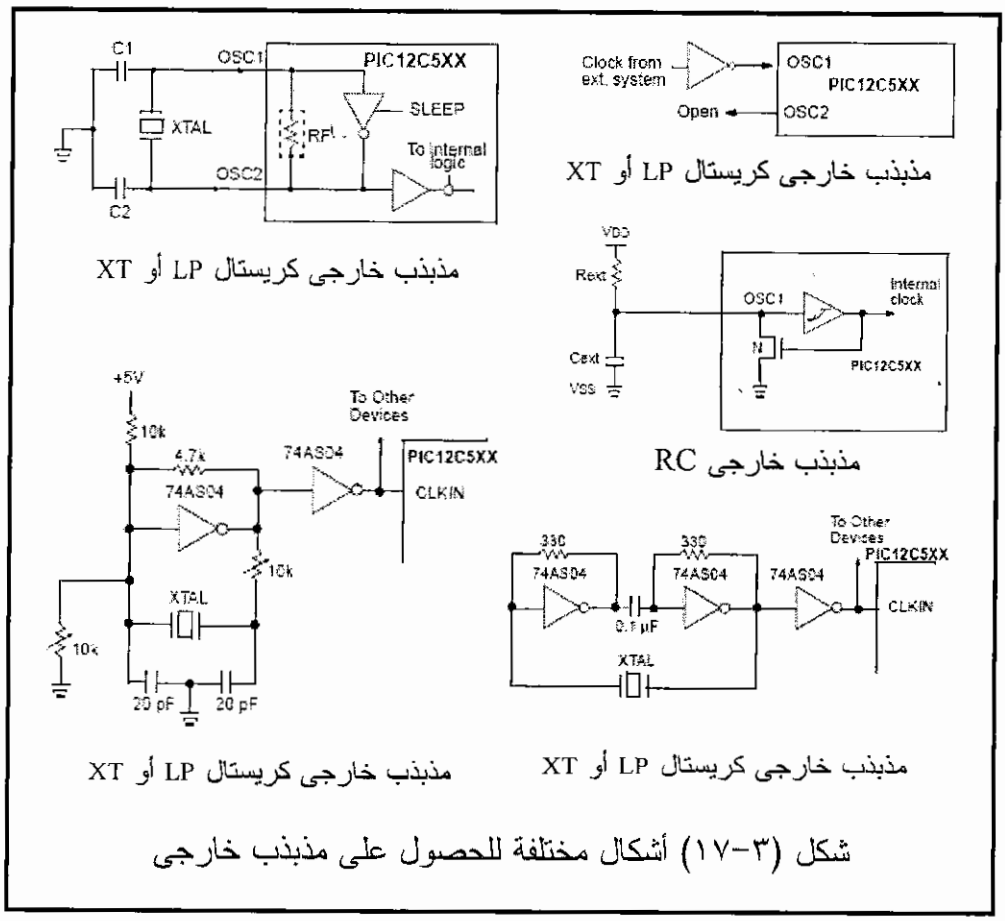
—	—	—	—	—	—	—	MCLRE	CP	WDTE	FOSC1	FOSC0
bit11	10	9	8	7	6	5	4	3	2	1	bit0

شكل (٣-١٦) ضبط مكونات المتحكم

يمكن إختيار واحد من ٤ أنواع من المذبذبات يعمل عليها المتحكم كما يلى :

نوع المذبذب	FOSC1	FOSC0
EXTRC مذبذب RC خارجى	1	1
INTRC مذبذب RC داخلى	1	0
XT مذبذب خارجى كريستال	0	1
LP مذبذب خارجى كريستال	0	0

شكل (٣-١٧) يبين الطرق المختلفة للحصول على مذبذب خارجى . عند اختيار المذبذب الداخلى يكون التردد هو ٤ ميجاهرتز ويمكن معايرته باستخدام المسجل OSCCAL .



مذبذب خارجى كريستال LP أو XT

مذبذب خارجى كريستال LP أو XT

مذبذب خارجى RC

مذبذب خارجى كريستال LP أو XT

مذبذب خارجى كريستال LP أو XT

شكل (٣-١٧) أشكال مختلفة للحصول على مذبذب خارجى

٣-٧ مجموعة أوامر المتحكم

سنترك شرح مجموعة أوامر هذا المتحكم لحين الانتهاء من شرح المتحكمات متوسطة الإمكانيات والمتمثلة فى المتحكم 16f84 فى الفصل القادم حيث أن مجموعة الأوامر هى نفسها لكل من هذين النوعين من المتحكمات .