**الفصل التاسع**

**تنظيم الذاكرة**

**9-1 مقدمة Introduction**

يوجد في شرائح متحكمات Microchip قسمان أساسيان في الذاكرة هما , ذاكرة البرنامج (program mem) و ذاكرة المعطيات (data mem) ولكل قسم ممر (bus) خاص به , لذلك من الممكن الوصول إليهما معا خلال نفس دورة التعليمة وهذا ما تعرفنا عليه في بنية هارفارد (Harvard architecture)

إن ذاكرة المعطيات يمكن تقسيمها إلى :

1. ذاكرة RAM ذات أغراض عامة (general purpose RAM)
2. مسجلات الوظائف الخاصة SFRS(Special Function Registers)

وهذه المسجلات إما أن تتحكم بعمل نواة المتحكم (CORE) وهي ما سوف نناقشه في هذا الفصل أو أن تتحكم بعمل المحيطيات (peripherals) والتي سيتم مناقشته في الفصول القادمة المتعلقة بكل صنف من هذه المحيطيات .

**9-2 تنظيم ذاكرة البرنامج Program Memory Organization**

**9-2-1 عداد البرنامج PC Program Counter**

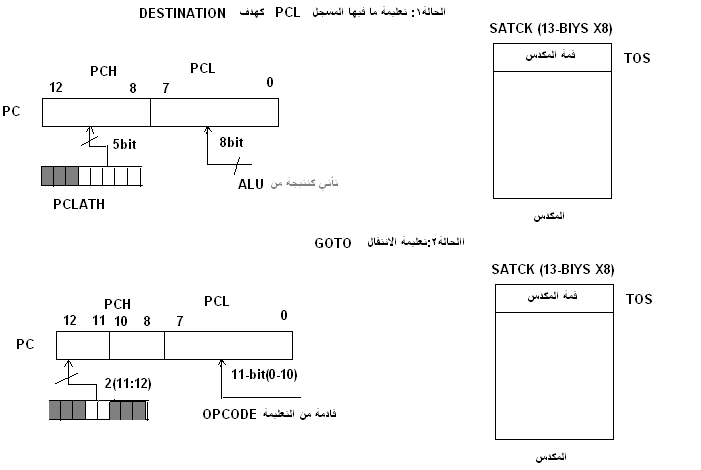
هو عداد ذو 13 bits وبالتالي يستطيع عنونة ذاكرة ذات عدد من المواقع قدره 2¹³=8.192=8k words حيث أن طول كلمة البرنامج كما أشرنا سابقا هو 14 bits.

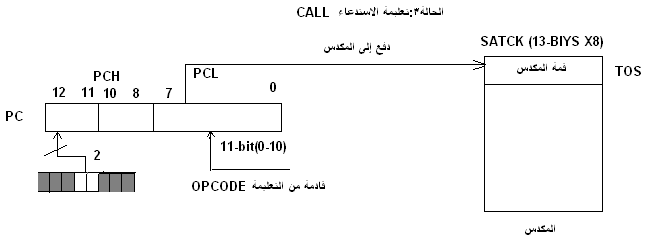
ولما كانت جميع التعليمات هي أحادية الكلمة فإن عدد التعليمات المتاحة لنا في برنامج المتحكم هو 8192 تعليمة.

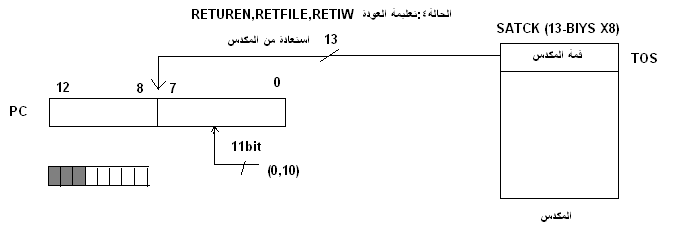
يتألف عداد البرنامج PC من قسمين :

1. قسم سفلي هو المسجلPCL وهو مسجل قابل للقراءة منه والكتابة فيه (readable\ writable)
2. قسم علوي هو المسجل PCH ولا يمكن القراءة منه أو الكتابة فيه مباشرة إنما يتم تحديثه دائما من خلال المسجل PCLATH الذي يمكن الوصول إليه قراءة أو كتابة.

نبين في الشكل (9-1) الحالات الأربع الممكنة للتعامل مع عداد البرنامج PC .







**الشكل** (9-1) تحميل المسجل PC في جميع الحالات الممكنة

**9-2-2 تقسيم ذاكرة البرنامج والمخطط البنيوي لها**

تقسم ذاكرة البرنامج إلى أربع صفحات كل صفحة بسعة 2K\*14 ويبين الجدول 9-1 مجال عنونة كل صفحة.

يتم الانتقال بين هذه الصفحات بتغيير الخانتين PC (11:12) بواسطة تغيير المسجل PCLATH وهو المسجل ماسك القسم العلوي للعداد PC(Program Counter Latch High) وهذا المسجل هو أحد مسجلات الوظائف الخاصة .

|  |  |  |
| --- | --- | --- |
| PC (12-11) | مجال العنونة | Page |
| 00  01  10  11 | 0000-07FF  0800-0FFF  1000-1FFF  1800-1FFF | Page0  Page1  Page2  Page3 |

**الجدول** 9-1 صفحات ذاكرة البرنامج

عند تنفيذ تعليمات البرنامج ونهاية إحدى صفحات البرنامج فإن العداد PC سينتقل آليا إلى الصفحة التالية بدون أي تدخل من المستخدم.

ومن أجل شرائح ذات ذاكرة برنامج أقل من 8 k word فإن الوصول إلى موقع ذاكري خارج مجال العنوان الفيزيائي المسموح سيسبب عملية التفاف (wraparound) والتي تعني ( من أجل شريحة ذات ذاكرة برنامج بسعة 4k word ) أن الوصول إلى العنوان 17ff يعنون فعليا 7ff, أما من أجل الشرائح ذات ذاكرة برنامج أقل من 2k word لا نحتاج إلى عملية تصفح لذاكرة البرنامج (paging).

نبين في الشكل 9-2 المخطط البنيوي لذاكرة البرنامج وكيف يتم التعامل بين عداد البرنامج وكل من المكدس وصفحات ذاكرة البرنامج .

**9-2-3 شعاع التصفير Reset Vector**

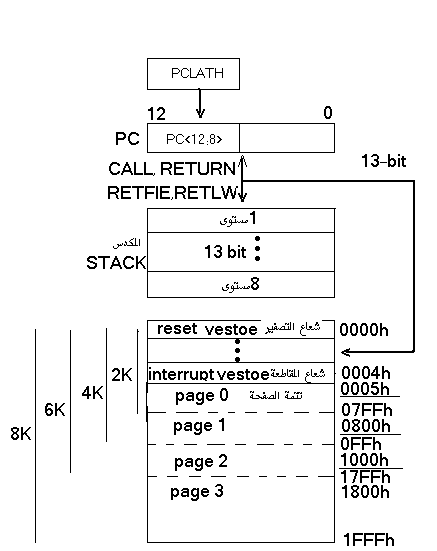
تقوم عملية تصفير(reset) الشريحة في جميع المتحكمات بتحميل عداد البرنامج PC بالقيمة 0000h ويدعى هذا العنوان عنوان شعاع التصفير (reset vector address), وهو العنوان الذي سينتقل إليه التنفيذ عند حدوث عملية تصفير للشريحة.

**ملاحظة** *: إن أي عملية تصفير ستؤدي إلى مسح محتويات المسجل PCLATH مما يعني أن أي عملية تفريع للتنفيذ (branch) (من خلال تعليمة Go To أو تعليمة استدعاء Call ) ستتم في الصفحة page 0 نفسها إلا إذا تم تغيير PCLATH قبل هذه التعليمة .*

**9-2-4 شعاع المقاطعة Interrupt Vector**

عند حدوث مقاطعة ما (interrupt) فإن عداد البرامج PC سوف يحمل مباشرة بالقيمة 0004h ويدعى هذا العنوان بعنوان شعاع المقاطعة (interrupt vector address ) ومن هذا الشعاع سيتم الانتقال إلى روتين خدمة المقاطعة ISR(interrupt service routine).علما أنه قبل تغيير قيمة PC إلى العنوان المفروض الانتقال إليه لتنفيذ الروتين ISR فإنه يجب تعديل محتويات المسجل PCLATH إلى القيمة التي توافق العنوان المطلوب الانتقال إليه (بداية البرنامجISR ).

ويجب الانتباه هنا إلى ضرورة تخزين محتويات المسجل PCLATH قبل التعديل السابق حتى يتسنى لنا استرجاعها في برنامج ISR عند الانتهاء منه والرغبة في العودة إلى التعليمة التي سبقتها المقاطعة.



**الشكل** (9-2 ) المخطط البنيوي لذاكرة البرنامج

***ملاحظة1****:لا تحتوي جميع الشرائح كامل حيز الذاكرة.*

***ملاحظة2****:إن معايرة المعلومات يمكن أن تبرمج في موقع ذاكرة البرنامج.*

**9-2-5 معايرة المعلومات Calibration Information**

إن بعض الشرائح تمتلك ميزة معايرة المعلومات المخزنة في ذاكرة البرنامج وهذه المعلومات تكون مبرمجة من قبل شركة Microchip أثناء عملية الفحص النهائي للشريحة حيث تسمح هذه الميزة بإعطاء نتائج أفضل وتتم عملية المعايرة هذه نموذجيا في نهاية البرنامج. وتنتج هذه العملية بشكل مشابه لتعليمة RETLW .

***ملاحظة****: من أجل الشرائح ذات النوافذ (WINDOWED DEVICE ) تكتب قيم المعايرة قبل إنجاز المسح (ERASING) مما يسمح باستعادة هذه القيم عندما يتم إعادة برمجة هذه الشريحة.*

**9-2-6 المكدس STACK**

إن المكدس في متحكمات Microchip هو عبارة عن مجموعة (combination) من ثمانية مستويات مما يسمح بحدوث عدد من المقاطعات (interrupts) والاستدعاءات (calls) حتى الثمانية وباختصار فإن المكدس يحمل عنوان العودة إلى البرنامج الرئيسي (المستدعي) .

وحيث أن المكدس يستخدم لتخزين قيم عداد البرنامج فكل مستوى من مستوياته هو بعرض 13 bit وبما أن عرض الموقع الذاكري للمكدس ليس 14 bit أو 8bitوبالتالي فالمكدس ليس جزءا من ذاكرة البرنامج أو ذاكرة المعطيات بل هو مستقل عنهما .

عند تنفيذ تعليمة الاستدعاء call لبرنامج فرعي ما فإنه يتم دفع push قيمة PC التي تعنون التعليمة التالية لتعليمة الاستدعاء إلى المكدس ونفس الأمر يحدث عند وقوع مقاطعة ما interruption. ويتم سحب قيمة PC من المكدس وذلك عند تنفيذ إحدى التعليمات RETURN, RETFE, RETLW. ويجب ملاحظة أن المسجل PCLATH لا يتم تعديله سواء عند الدفع PUSH إلى المكدس أو عند السحب POP منه.

عند امتلاء المكدس بثماني قيم عودة لعداد البرنامج فإن عملية الدفع التاسعة سوف تخزن قيمة PC فوق القيمة الأولى وعملية الدفع العاشرة سوف تخزن فوق القيمة الثانية وهكذا وبالتالي فإن المكدس هو محطة تخزين دائرية .

يبين الشكل (9-3) كيفية إنجاز هذا التخزين الدائري.

**ملاحظة** : *- لا يوجد أعلام تدل على طفحان المكدس .*

*- لا يوجد تعليمات POP, PUSH بل تتم تلقائيا من خلال تنفيذ التعليمات*:

PUSH: Call, Any Interrupt

|  |
| --- |
| Push1 Push9  Push2 Push10  Push3  Push4  Push5  Push6  Push7  Push8 |

POP: RETURN, RETLW, retfie

STACK

\_\_\_\_\_\_\_\_\_Top of STACK

**الشكل** (9-2)

**9-2-7 تقسيم ذاكرة البرنامج Program Memory Paging**

تمتلك تعليمات Call, Go to مجال عنونة بعرض 11 خانة أي يمكنها تفريغ البرنامج ضمن مجال حتى 2k .

إلا أن بعض الشرائح تكون فيها ذاكرة البرنامج ذات سعة أكبر من 2k لذا حتى يتسنى لتعليمتي CALL,GO TO عنونة كامل المجال الذاكري المحجوز لذاكرة البرنامج يجب أن يكون هناك خانتين لتحديد صفحة ذاكرة البرنامج .

يستخدم لعملية تحديد ذاكرة البرنامج paging الخانتين PCLATH(4:3) كما هو مبين في الجدول (9-1) والتي تبرمج لاختيار صفحة ذاكرة البرنامج الهدف.

عندما يتم تنفيذ تعليمة برنامج عودة ما فإن كامل الخانات الثلاث عشرة 13 BIT لعداد البرنامج PC سوف تسحب poped من المكدس لذا لا نحتاج إلى تعليمات برمجة للخانتين PCLATH(4:3) عند العودة من برنامج فرعي أو من روتين خدمة مقاطعة ISR.

**ملاحظة**:- *في الشرائح ذات ذاكرة البرنامج 2K\*14 وما دون يتم تجاهل خانتي التصفح Paging أي PCLATH < 4:3> حيث تستخدمان للوصول إلى ذاكرة برنامج فيها أكثر من صفحة.*

*- ويمكن استخدام هاتين الخانتين حينئذ كخانتي قراءة \ كتابة ذات أغراض عامة ألا أن ذلك لا ينصح به.*

*- إن الشرائح ذات ذاكرة برنامج4K,2K تتجاهل الخانة PCLATH<4> التي تستخدم للوصول إلى الصفحات Page2, Page3 .*

مثال استدعاء برنامج فعلي واقع في page 1من الصفحة page 0

* نلاحظ أن محتويات PCLATH تخزن ويعاد استعادتها (افتراضيا) بين بداية ونهاية البرنامج الفرعي أو روتين خدمة المقاطعة ISR .

ORG \*500

اختيار الصفحة BSF PCLATH/3/ (800FFF) 1

استدعاء البرنامج الفرعي CALL SUB1-P1

ORG 0\*900

البرنامج الفرعي المستدعى SUB1/

عودة إلى البرنامج المستدعى في الصفحة RETURN Page 0

**9-3 تنظيم ذاكرة المعطيات Data Memory Organization**

**9-3-1 تمهيد**

تتألف ذاكرة المعطيات من حيزين ذاكريين :

* 1. حيز مسجلات الوظائف الخاصة (Special Function Registers ) SFRs وهذه المسجلات SFRs تحكم عمليات الشريحة operation من خلال التحكم بوظائف نواة المتحكم الداخلية core أو من خلال التحكم بالوظائف المعدة للمحيطيات peripherals
  2. حيز مسجلات الأغراض العامة (General Purpose Registers ) GPRs الذي يستخدم لتخزين المعطيات وفق تعريف المستخدم وتلبية لإستراتيجيته البرمجية .

عملية تقسيم ذاكرة المعطيات تسمى <Banking> ويتم الانتقال بين البنوك المتاحة بواسطة خانات برمجية خاصة بذلك هي الخانات status<7:5> .

يمكن الوصول إلى أي موقع ذاكري في ذاكرة المعطيات بواسطة العنونة المباشرة (التي تستخدم الخانات (6:5) statusالمسماة (RP0,RP1) ) أو بواسطة العنونة غير المباشرة (باستخدام المسجل SFR المسمى مسجل اختيار الملف ) ويستخدم مع هذا المسجل الخانة (indirect register pointer)IRP التي تستخدم للوصول إما إلى BANK0/BANK1 أو إلى BANK2/BANK3 وسنقوم بشرح مفصل لكلا أسلوبي العنونة في الفقرات القادمة.

**9-3-2 مسجلات الأغراض العامة special function registers**

تستخدم هذه السجلات من قبل وحدة المعالجة المركزية CPU والوحدات المحيطية وذلك من أجل التحكم بالوظائف لإنجاز العمليات المطلوبة وتعمل هذه السجلات بتقنية (static RAM).

تقسم مسجلات الوظائف الخاصة SFRs إلى قسمين:

1. مسجلات تتحكم بوظائف النواة (CORE).
2. مسجلات تتحكم بالمحيطات (PERIPHERALS).

وفي هذا الفصل سندرس بالتفصيل المسجلات المتعلقة بالنواة أما الصنف الآخر فسيدرس بشكل تتابعي مع كل بحث قادم.

بعض المسجلات SFRs يتم عودتها إلى حالة بدائية بعد كل عملية تصفير POR (تصفير عند وصل التغذية) أو أي عملية تصفير , وبعض المسجلات الأخرى يتم تصفيرها و أخرى لا تتأثر(unaffected)

يمكن الوصول إلى هذه المسجلات بشكل مباشر أو غير مباشر بواسطة المسجل FSR.

**ملاحظة**: يمكن أن تحتوي منطقة مسجلات SFRs بعض مسجلات GPR.

**9-3-3 الانتقال بين بنوك ذاكرة المعطيات <Banking>**

تقسم ذاكرة المعطيات إلى أربعة أقسام كل منها نسميه بنك وكل بنك يحتوي على [0→7f]128 bites حيث تشغل مسجلات الوظائف الخاصة SFRs الحيز السفلي منه وفي الحيز العلوي منه مسجلات الأغراض العامة GPRs.

بعض المسجلات عالية الاستخدام (high use) الموجودة في البنك bank0 لها صور في البنوك الأخرى وذلك من أجل الاختصار في الشيفرة code وتسريع عملية الوصول .

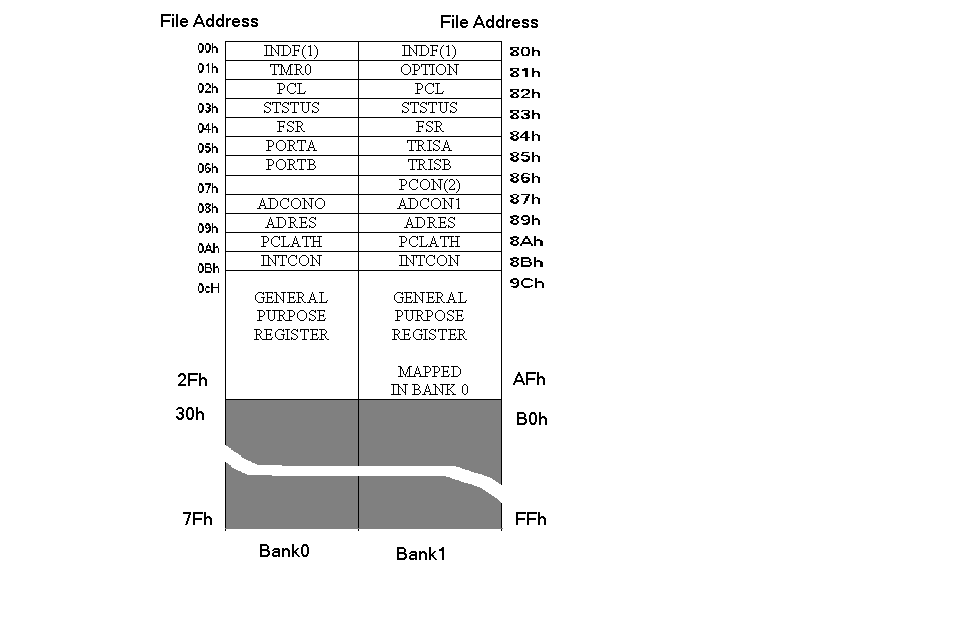
يتم الانتقال بين هذه البنوك الأربعة بطريقتين وذلك حسب نوع العنونة المختار , ففي حالة العنونة المباشرة يستخدم لذلك الخانتين RPO, RP1 أما في حالة العنونة غير المباشرة تستخدم الخانة IRP وهذه الخانات موجودة في مسجل الحالة (status reg) القادم شرحه في فقرات لاحقة .

يبين الجدول 9-2 قيم هذه الخانات والبنوك المختارة المقابلة بالنسبة لنمطي العنونة المباشرة وغير المباشرة .

|  |  |  |
| --- | --- | --- |
| Indirect IRP | Direct PR1: PR0 | Bank |
| 0  1 | 00  01  10  11 | 0  1  2  3 |

**الجدول** 9-2

إن وجود أربعة بنوك في ذاكرة المعطيات ليس متوفرا في جميع العائلات فبعض العائلات مثل PIC16C71X تحتوي فقط على اثنين من هذه البنوك الأربعة كما في الشكل (9-5) الذي يمثل مخطط مسجلات ذاكرة المعطيات في المتحكم PIC16C71.



**الشكل** (9-5 ) مخطط مسجلات ذاكرة المعطيات

**ملاحظة :** *في حال وجود بنكين فقط نحتاج إلى الخانة RPO فقط في العنونة المباشرة ولا نحتاج إلى الخانة IRP في العنونة غير المباشرة حيث ستعتبر قيمتها"0".*

يظهر في الجدول 9-3 ملخصا حول جميع المسجلات ذات الوظائف الخاصة مع عناوينها وقيمها عند أحوال مختلفة من التصفير وذلك في المتحكم PIC16C71 .

|  |  |  |
| --- | --- | --- |
| **Indirect**  **(IRP)** | **Direct**  **RP1:RP0** | **Accessed**  **Bank** |
| 0 | 00 | 0 |
| 01 | 1 |
| 1 | 10 | 2 |
| 11 | 3 |

**الجدول** 9-3 ملخص المسجلات ذات الوظائف الخاصة

**9-3-5 مسجل الحالة ومسجل الاختيار**

**أولا : مسجل الحالة (Status Register)**

وهو من أهم المسجلات يحتوي على أعلام الحالة للوحدة ALU بالإضافة إلى أعلام التصفير وخانات اختيار بنوك المعطيات , ويتم تعديل محتويات مسجل الحالة بعد تعليمات محددة , وهناك بعض التعليمات التي لا تغير من محتويات هذا المسجل مثل SWAPF, DECFSZ, NOP وينصح عند فحص خانات المسجل Status استخدام التعليمات PTEFC,PTFSS حيث أن مثل هذه التعليمات لا تؤثر على محتويات مسجل الحالة وبإمكانك للكتابة في المسجل Status استخدام التعليمات BCF, BSF, MOVW F

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **R/W-X** | **R/W-X** | **R/W-X** | **R-1** | **R-1** | **R/W-0** | **R/W-0** | **R/W-0** |
| **C** | **DC** | **Z** | **PD** | **TO** | **RP0** | **RP1** | **IRP** |

Bit1 Bit0

**دلائل الرموز** :

R = تعني إن الخانة قابلة للقراءة readable

W =تعني أن الخانة قابلة لكتابة writable

X =قيمة غير منتهية

U =خانات غير مستخدمة unimplement تقرأ 0

n -= القيمة عند حدوث التصفير POR

الخانة IRP : خانة اختيار بنك المعطيات (عند العنونة غير المباشرة )

0 =اختبار bank 0,1

1= اختبار BANK 2,3

الخانتين RP1:RP0: خانات اختيار بنك المعطيات (العنونة المباشرة )

00= bank0 (00-7fh)

01=bank1 (80-ffh)

10=bank2 (100-17fh

11=bank3 (180-1ffh)

الخانة T0 : انقضاء زمن تأخير المؤقت (Time-out ) WDT

0= انتهاء زمن تأخير مؤقت المراقبة WDT

1= بعد الإقلاع power up وبعد تعليمة CLRWDT وبعد تعليمة SLLEP

الخانة P0 : خانة الطاقة المنخفضة power-down bit

0= بعد تنفيذ التعليمة SLEEP

1= بعد الإقلاع power up وبعد التعليمة LLROWDT

**ملاحظة :**

***الخانتان PD,T0 سيتم مناقشتهما في الفصل الثالث من هذا الباب لذلك لا علينا أن ننشغل بهما في هذا الموضع بشمل كبير* .**

الخانة Z:علم الصفرية

0= نتيجة العليمة الرياضية أو المنطقية ليست صفرا.

1= نتيجة العملية الرياضية أو المنطقية هي صفر

الخانة DC : علم نصف الحمل /الاستعارة

0= لا يوجد حمل /استعارة من/إلى الخانة الرابعة (في النيبل السفلي)

1= حدوث حمل /استعارة من /إلى الخانة الرابعة (في النيبل السفلي)

الخانة C: علم الحمل/ الاستعارة

0= عدم حدوث حمل /استعارة من/إلى الخانة الأكثر أهمية (الثامنة )

1= حدوث حمل /استعارة من/إلى الخانة الأكثر أهمية (الثامنة )

**ملاحظة : *من أجل التعليمات RLF, RRF فإن العلم C يمكن أن يؤخذ إما القيمة LSB أو MSB حسب التعليمة .***

**ثانياً: مسجل الاختيار(option)**

R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1 R/W-1

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

الخانة : خانة تمكين الانحياز العلوي للنافذة B

1=عدم تمكين الانحياز العلوي(pull-up)

0= تمكين الانحياز(pull-up)

الخانة INTEDG: خانة اختيار جبهة المقاطعة على القطب RBO

1=حدوث المقاطعة عند الجبهة الصاعدة (rising edge)على القطب RBO.

0=حدوث المقاطعة عند الجبهة الهابطة(falling edge)على القطب RBO.

الخانة TOCS: خانة اختيار مصدر نبضات المؤقت TMRO

1=حدوث الانتقال على القطب RA4/TOCKI (عداد)

0=نبضات الساعة الخاصة بدورات التعليمة الداخلية

الخانة TOSE:خانة اختيار جبهة زيادة المؤقت على القطب RA4

1=الزيادة عند الجبهة الهابطة على القطب RA4/TOCKI

0=الزيادة عند الجبهة الصاعدة على القطب RA4/TOCKI

الخانة PSA:خانة تعيين تابعة المقسم(prescaler)

1=تابعة القسم إلى المؤقت WDT

0=تابعة القسم على المؤقت TIMER 0

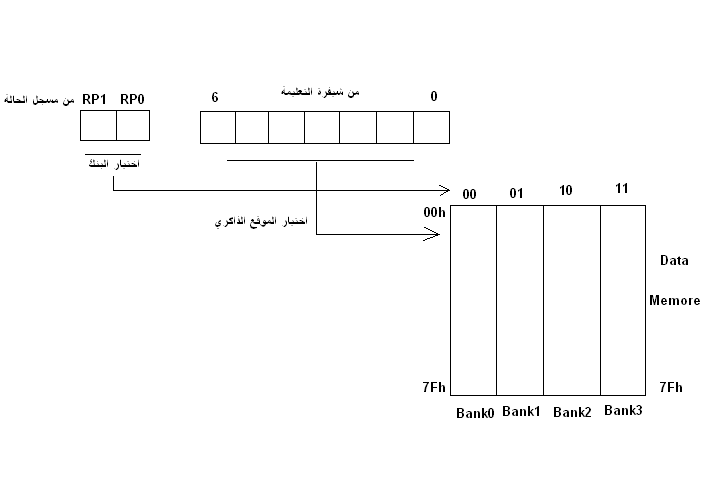
الخانة PS2:PSO: خانات اختيار نسبة المقسم (prescaler)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| الخانات | | | | | نسبة التقسيم للمؤقت TMRO | نسبة التقسيم للمؤقت WDT |
| PS0 | | PS1 | PS2 | |
| 0 | 0 | | | 0 | 1:2 | 1:1 |
| 1 | 0 | | | 0 | 1:4 | 1:2 |
| 0 | 1 | | | 0 | 1:8 | 1:4 |
| 1 | 1 | | | 0 | 1:16 | 1:8 |
| 0 | 0 | | | 1 | 1:32 | 1:16 |
| 1 | 0 | | | 1 | 1:64 | 1:32 |
| 0 | 1 | | | 1 | 1:128 | 1:64 |
| 1 | 1 | | | 1 | 1:256 | 1:128 |
|  |  | | |  |  |  |

**9-3-6 العنونة المباشرة Direct Addressing**

وهي مبينة في الشكل 9-6:

لاختيار موقع ذاكري (بايت) من أصل 128 بايت نحتاج إلى 7 خانات عنونة حيث أن 27=128 وهذه الخانات السبع نحصل عليها من شيفرة التعليمة نفسها والتي تحتوي على عنوان المسجل الهدف من خلال سبع خانات في شيفرة التعليمة من أصل 14 خانة أما الخانات الأخرى فهي لتشفير نوع العملية (تحميل- جمع).



**الشكل** (9-6) العنونة المباشرة

ولاختيار البنك الهدف يتم استخدام الخانتين RP1:RP0 من مسجل الحالة واللتان تستطيعان عنونة 22=4 بنوك , وبالعودة إلى الجدول 3-9 نجد قيم الخانات RP0:RP1 الموافقة لاختيار بنك من البنوك الأربعة .

**9-3-7 العنونة غير المباشرة: Indirect Addressing**

يستخدم هذا الأسلوب من العنونة عندما يكون عنوان ذاكرة المعطيات متغيراً باستمرار كما في حالة التعامل مع الكتل (blocks) الذاكرية, ونذكر انه عند التعامل مع الجداول الذاكرية كنا نقوم بتهيئة مؤشر الذاكرة على قيمة ابتدائية ثم نقوم بتعديل هذه القيمة لمسح مجال محدد من الذاكرة هو طول الكتلة الذاكرية .

يستخدم المجال FSR هنا كمؤشر علماً أن هذا المؤشر هو احد مسجلات الوظائف الخاصة وبالتالي يمكن تعديله أثناء البرنامج,والمسجل الذي عنوانه هو قيمة FSR يشار إليه بالمسجل INDF حيث سيتم التعامل مع هذا المسجل عند استخدام العنونة غير المباشرة .

نبين في الشكل (7.9)كيف تتم عملية العنونة غير المباشرة.

* حيث سنلاحظ أن البايت الهدف يتم اختياره بواسطة خانات المسجل FSR الأولى أما البنك الهدف فيتم اختياره بواسطة الخانتين FSR(7) و IRP من مسجل الحال

IRP FSR register

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

|  |
| --- |
|  |

اختيار البنك 00 01 10 11 00h

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

اختيار الموقع

7

DATA

Memory

Bank0 bank1 bank2 bank3

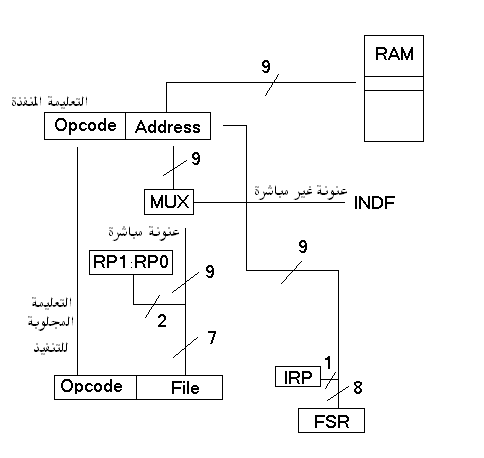
**الشكل** 9-7 العنونة غير المباشرة

**9-3-8 خلاصة في نمطي العنونة :**

كما هو مبين في الشكل 8-9 فإننا نحتاج دوما للوصول إلى موقع ما في ذاكرة المعطيات إلى تسع خانات لان سعة هذه الذواكر هي 29=152=128K .

وبما أن هذه الذواكر مقسمة إلى أربعة بنوك فإننا نحتاج إلى خانتين للتنقل بين هذه البنوك وهاتان الخانتان هما في أسلوب العنونة المباشرة RP1:RP0 من المسجل status , وفي لأسلوب العنونة غير المباشرة هماIRP من المسجل status والخانة FSR(7) .

الآن بعد اختيار البنك الهدف نجد بأنه لاختبار البايت الهدف من أصل 128 بايت نحتاج إلى سبع خانات حيث 27=128 ففي أسلوب العنونة المباشرة يتم الحصول على هذه الخانات السبع من شيفرة OPcode للتعليمة نفسها وفي حالة العنونة غير المباشرة تخزن هذه الخانات السبع في الخانات السبع الأولى من المسجل . FSR



**الشكل9**-8 طرق الوصول إلى الذاكرة RAM

ملاحظة هامة لأجل العنونة غير المباشرة:

*يمكن أن نستخدم في حالة العنونة غير المباشرة مسجلا خاصا INDF حيث أن أي تعليمة تستعمل المسجل*

*INDF ستؤدي فعليا إلى الوصول إلى المسجل المشار إليه بواسطة المسجل FSR* .

**مثال** RAM Initialization

سوف نبين في هذا المثال كيف تتم عملية تهيئة الذاكرة RAM وذلك بتصفير مسجلات الأغراض العامة في هذه الذاكرة والموجودة في البنوك الأربعة لهذه الذاكرة .

حيث تم استخدام نمط العنونة غير المباشرة بحيث تم في كل مرة تحميل عنوان أول مسجل من مسجلات الأغراض العامة الموجودة في البنك المختار وإجراء عملية التصفير الدورية بواسطة المسجل INDF ثم عملية اختبار انتهاء عملية تهيئة البنك وذلك إما باختبار الخانة السابعة من FSR والتي ستصبح 1 عند نهاية البنك ذات العنوان 7F=01111111 .

تصفير مسجل الحالة اختيار الـ CLRF Status /bank

تحميل أول عنوان من مسجلات الأغراض العامة MOVLW OX20 /

Bank0-LP

تصفير المسجل GPR المشار إليه بواسطة المسجل CLRF INDFO / FSR

زيادة FSR تمهيدا لاختيار مسجل GPR التالي لتصفيره HNCF FSR /

اختبار انتهاء bank0 عند العنوان BTFSS FSR/7 7F

GO TO bank0-LP

Bank 1- Select

تحميل أول عنوان من مسجلات الأغراض العامة المرادMOVLW OXA0 /

تصفيرها في bank1 إلى المؤشر MOVWF FSR / FSR

BANK 1-LP

تصفير المسجل لحق المشار إليه بواسطة CLRF INDFO /FSR

زيادة INCF FSR / FSR

? اختبار انتهاء BTFSS status /c (FER=00 /C = 1)

GO TO bank 1-LP

/ bank2 – select

اختيار أحد البنكين BSF status /IRP/ bank 2/ bank 3

تحميل أول عنوان من مسجلات الأغراض العامة MOVLW OX20 /

المراد تصفيرها في الـ bank2 إلى المؤشر MOVWF FSR / FSR

Bank 2-LP

تصفير المسجل GPR المشار إليه بواسطة CLRF INDF0 / FSR

زيادة المؤشر INCF FSR

؟ اختبار إنهاء BTFSS FSR7 bank 2

GO TO bank 2 –LP

Bank 3 – select

اختيار MOVLW OXAO bank3

MOVWF FSR

Bank 3 –LP

تصفير المسجل GPR المشار إليه بواسطة CLRF INDF0 FSR

زيادة المؤشر INCF FSR

؟ اختبار إنهاء الـ BTFSS status/c (fsr=00/C=1)

GO TO bank 3 –LP