**الفصل العاشر**

**مجموعة تعليمات متحكمات PIC**

* 1. **تصنيف تعليمات متحكمات PIC**

يمكننا أن نصنف مجموعة التعليمات هذه وفق عدة تصنيفات, إذ يمكن التمييز بين تعليمات حسابية وأخرى منطقية أو تعليمات خانة وتعليمات تحكم, والتصنيف الأساسي المعتبر هنا هو كيفية توزيع شيفرة التعليمة في كل مجموعة من مجموعات التعليمة, واعتمادا على هذا المعيار يمكن أن نصنف مجموعة تعليمات متحكمات PIC وفق ثلاث مجموعات رئيسية:

 1.مجموعة تعليمات البايت (byte-oriented operations)

 2.مجموعة تعليمات الخانة (bit-oriented operations)

 3.مجموعة تعليمات التحكم وتعليمات الثوابت (literal and control operations)

بهدف تسهيل التعامل مع تعليمات متحكمات PIC سنقدم هنا ثلاث جداول:

الجدول الأول:الجدول(10-1) يحتوي على جميع تعليمات المتحكم PIC الخاصة بعائلة المجال الوسطي (Mid-Range) ويبين هذا الجدول صيغة كل تعليمة والأعلام المتأثرة بها.ويعتبر هذا الجدول مرجعا سريعا لأخذ معلومات شاملة عن التعليمات ككل.

الجدول الثاني:الجدول (10-2) يحتوي على وصف للرموز المستخدمة في التعليمات.

الجدول الثالث :الجدول (10-3) يصنف الشيفرات المختلفة لمجموعات التعليمات .

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Notes** | **Status Affected** | **14-Bit instruction word** | **Cycles** | **Description**  | **Mnemonic,****Operands**  |
| **MSB LSB** |
| **BYTE-ORIENTED FILE REGISTER OPERATIONS** |
| 1,21,221.21.21.2.31.21.2.31..21.21.21.21.21.21.21.2 | C,D,C,ZZZZZZZZZCCC,D,C,ZZ |  00 0111 dfff ffff 00 0101 dfff ffff 00 0001 ifff ffff 00 0001 0xxx xxxx 00 1001 dfff ffff 00 0011 dfff ffff 00 1011 dffff ffff 00 1010 dfff ffff 00 1111 dfff ffff 00 0100 dfff ffff 00 1000 dfff ffff 00 0000 ifff ffff 00 0000 0xx0 0000 00 1101 dfff ffff 00 1100 dfff ffff 00 0010 dfff ffff 00 1110 dfff ffff 00 0110 dfff ffff | 111111111111111111 | Add Wand fAND W WITH fClear fClear WComplement f Decrement fDecrement f, Skip if 0Increment fIncrement f. Skip if 0Inclusive or W with fMove fMove W to fNO operation Rotate left f through carryRotate Right f through carrySubtract W from fSwap nibbles in fExclusive OR W with f  | ADDWF f,dANFWF f,dCLRF fCLRW \_COMF f,dDECF f,dDECFSZ f,dINCF f,dINCFSZ f,dIORWF f,dMOVF f,dMOVWF fNOP \_RLF f,dRRF f,dSUBWF f,dSWAPF f,dXORWF f,d |
| **BIT-ORIENTED FILE REGISTER OPERATIONS** |
| 1.21.233 |  | 1. 00bb bfff ffff
2. 01bb bfff ffff
3. 10bb bfff ffff

01 11bb bfff ffff | 1111 | Bit Clear fBit Set fBit Test f, Skip if ClearBit Test f, Skip if Set | BCF f,bBSF f,bBTFCS f,bBTFSS f,b |
| **LITRAL AND CONTROL POERATION** |
|  | C.D.C.ZZZC,D,C,ZZ | 1. 111x kkkk kkkk
2. 1001 kkkk kkkk
3. 0kkk kkkk kkkk
4. 0000 0110 0100

10 1kkk kkkk kkkk11 1000 kkkk kkkk11 00xx kkkk kkkk00 0000 0000 100111 01xx kkkk kkkk00 0000 0000 000000 0000 0110 001111 110x kkkk kkkk11 1010 kkkk kkkk | 1121211222111 | Add literal and WAND literal with WCall subroutineClear WatchdogGo to addressInclusive OR literal with WMove literal to WReturn from interrupt Return with literal in WReturn from SubroutineGo into standby modeSubtract W from literalExclusive OR literal with W  | ADDLW KANDLW K CALL KCLRWDT -GOTO KIORLW KMOVLW KRETFILE -RETLW KRETURN -SLEEP -SUBLW KXORLW K |

**الجدول ( 10-1)** تعليمات المتحكم PIC

|  |  |
| --- | --- |
|  **Description** | **Field** |
| Register file address (0x00 to 0x7F) | f |
| Working register (accumulator) | w |
| Bit address within an 8-bit file register ( 0 to 7) | b |
| Literal field ,constant date or label (may be either an 8-bit or an 11-bit value) | k |
| Don t care (0 or 1)The assembler will generate code with x=0 it is the recommended from of use for compatibility with all Microchip software tools | x |
| Destination select; d=0: store result in W, d=1 : store result in file register f: | d |
| Destination either the W register or the specified register file Location | dest |
| Label name | label |
| Top of stack | TOS |
| Program counter  | PC |
| Program counter high latch | PCLATH |
| Global interrupt enable bit | GIE |
| Watchdog timer | WDT |
| Time \_out bit |  |
| Power –down bit |  |
| Optional | {} |
| Contents | () |
| Assigned to |  |
| Register bit field | <> |
| In the set of | E |
| User defined term(font is courier) | italics |



**الجدول(10-2)** وصف للرموز المستخدمة في التعليمات

 **Byte –oriented** file register operations

13 8 7 6 0

|  |  |  |
| --- | --- | --- |
| f(FILE#) | d | OPCODE |

 d=0 for destination W

 d=1 for destination f

 f=7-bit file register address

**Bit –oriented** file register operations

13 8 7 6 0

|  |  |  |
| --- | --- | --- |
| f(FILE#) | b(BIT#) | OPCODE |

 b =3- bit bit address

 f =7- bit file register address

**Literal and control** **operations**

 **General**

 13 8 7 0

|  |  |
| --- | --- |
|  OPCODE | K(literal) |

 K=8- bit literal (immediate) value

 **CALL and GOTO instruction only**

 13 11 10 0

|  |  |
| --- | --- |
|  OPCODE | K(literal) |

 K=11-bit literal (immediate) value

**الجدول (10-3)** الشيفرات المختلفة لمجموعات التعليمات

**10-2 زمن تنفيذ تعليمات متحكمات PIC وشيفراتها الأساسية**

**10-2-1 زمن التعليمة**:

ذكرنا في بداية رحلتنا مع المتحكم PIC أن أحد مظاهر قوته البارزة , أن زمن تنفيذ جميع تعليماته إنما هو دورة ساعة واحدة باستثناء تعليمات تفريغ البرنامج سواء التفريغ المشروط كتعليمة BTFSS أو التفريغ غير المشروط كتعليمة Call وتعليمة go to أو بكلمات أخرى فإن التعليمات التي تؤدي إلى تغير قيمة عداد البرنامج تحتاج إلى دورتين من دورات التعليمة 2TCY.

وحيث أن أي تعليمة من تعليمات متحكمات PIC ذات دورة TCY واحدة أو دورتين 2Tcy فسيمكنك ببساطة أن تقوم بحساب الزمن الكلي لتنفيذ البرامج خصوصا إذا كان الهزاز ذو تردد 4MHZ. وكما نعلم أن دورة التعليمة Tcy تتضمن أربع أدوار من أدوار الهزاز (1÷MHZ=0.25µS) وبالتالي فإن Tcy=4\*0.25=1µs ، أي أن زمن تنفيذ التعليمة هو 1µs أو µ2 حسب نوع التعليمة.

ومن المفيد أن نذكر هنا أن تصور صغر قيمة هذا الزمن يعطينا مرونة برمجية, حيث ستكون عطالتنا البشرية عند تنفيذنا للأعمال حاجزاً أمام تصور كيفية إنجاز عدد كبير من التعليمات التي تخدم أكثر من غرض, وخصوصاً أثناء مسح لوحة المفاتيح والشاشة والقنوات التشابهية وغير ذلك كلها بنفس الفترة القصيرة بالنسبة لدينا (نحو 5 ms وهو زمن لا يذكر لدى حواسنا البشرية), في حين سيكون هذا الزمن كبير جداً جداً بالنسبة لإمكانية المتحكم PIC (حيث يستطيع أن ينفذ خلال الزمن المذكور 5 ms نحو 5000 تعليمة).

وعند دراستك لتطبيق مقياس الجهد التشابهي ذو الأربع قنوات قياس ولوحة المفاتيح 4×4 والشاشة ذات الأربع وحدات من نوع 7- segment ستجد أهمية التصور, وأنصح بالتركيز الشديد على هذا التطبيق حيث سيكون بحق اللبنة الأساسية عندك لاستثمار صناعي وخدمي أفضل للمتحكم PIC .

**10-2-2 شيفرات تعليمات المتحكم PIC**

يبين الجدول (10-1) وصفاً للشيفرات الأساسية لكل مجموعة من التعليمات .

* **شيفرة تعليمات البايت**: تحدد الخانات السبع الأولى فيها عنوان المسجل حيث بإمكانها عنونة  مسجلاً في المجال (00 – 7F ), أما الخانة الثامنة d فهي تحدد أين ستوضع نتيجة التعليمة أو أين هدفها.

 فعندما d= 0 ستوضع النتيجة حصراً في المراكم W .

 وعندما : d=1 ستوضع النتيجة في الملف المعين بالتعليمة نفسها f.

 أما الخانات الست الأخيرة فهي تشفر نوع العملية المنفذة على المسجل f.

* **شيفرة تعليمات الخانة**: كما في تعليمات البايت فإن أول سبع خانات محجوزة لعنوان المسجل المراد تطبيق التعليمة على أحد خاناته, أما الخانات الثلاث التالية d فهي التي تحدد عنوان الخانة الهدف داخل المسجل f, والخانات الأربع الأخيرة تشفر نوع العملية المطبقة على الخانة b من المسجل f.
* **شيفرة تعليمات التحكم وتعليمات الثوابت**: من أجل تعليمتي تفريغ البرنامج GOTO, CALL تحجز أول إحدى عشرة خانة لقيمة الثابت K الذي يمثل عنوان لافتة LABEL ويعطي فكرة عن مجال هاتين التعليمتين (الذي هوتعليمة), وتمثل الخانات الباقية شيفرة التعليمة أهي قفز GOTO أم استدعاء CALL.

أما باقي تعليمات المجموعة فمنها التعليمات SLEEP ,RETURN ,RETFILE ,CLRWDT التي تعتبر كتعليمات مباشرة لا علاقة لها بالثابت k لذا فجميع الخانات هي شيفرة.

والتعليمات الأخرى تحتوي على قيمة الثابت k ثماني الخانات الذي يعبر عن أحد أطراف عملية حسابية أو منطقية أو تحميل.

* + 1. **دورة تنفيذ التعليمة**

تشتمل دورة التعليمة Tcy على أربع فترات Q وهي الدوراتQ1 Q2 وكل منها هو نفسه دورة هزاز الشريحة Tosc ,وتزود الدورات Q المتحكم بالقاعدة الزمنية التي تضبط تنفيذ فعاليته المتعددة (القراءة –معالجة المعطيات –الكتابة.....)

إن دورة كل تعليمة Tcy تتألف من أربع دورات Q,وبإمكاننا توزيع تنفيذ التعليمة خلال كامل دورتها Tcy كما يلي :

خلال Q1:دورة فك الشيفرة (Decode cycle) أو عدم وجود أي عملية (no -operation)

خلال Q2:دورة قراءة التعليمة (Read cycle) أو عدم وجود أي عملية (No -operation)

خلال Q3:معالجة المعطيات Date.

خلال Q4:دورة كتابة (Write cycle) أو عدم وجود أي عملية (no - operation).

* 1. **مجموعة تعليمات البايت byte oriented operations**

**التعليمة** **MOVF**

**الصيغة** MOVF f , d

**المعاملات** 127 f0

(0, 1)  d

**العملية** d (f)

ا**لأعلام المتاثرة** z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1000 | 00 |

**وصف التعليمة** تقوم بنقل محتويات المسجل f إلى الهدف d (destination) الذي هو المراكم w عندما d=0 أو المسجل f نفسه عندما d=1 ,وفائدة التعليمة في هذه الحالة هي فحص محتويات المسجل f عن طريق العلم z.

 ***الأمثلة:***

***مثال 1:***

 MOVF FSR ,0

قبل تنفيذ التعليمة:

 × 00 W=0 FSR= 0 × C2

 بعد تنفيذ التعليمة:

 C2 W= 0 × C2 × Z=0 FSR= 0

***مثال 2:***

 MOVF INDF, 0 (عنونة غير مباشرة)

قبل تنفيذ التعليمة:

FSR (FSR) = 0 × 00 FSR = 0 × C2 W = 0 ×17 هذه القيمة محتواة في المسجل الذي عنوانه بعد تنفيذ التعليمة:

 (FSR) = 0 × 00 FSR = 0 × C2 W = 0 × 00

 Z = 1

***مثال 3:***

 MOVF FSR, 1 (فحص محتويات المسجل عن طريق العلم Z)

الحالة الأولى

 قبل تنفيذ التعليمة:

 FSR = 0 × 43

 بعد تنفيذ التعليمة:

 Z = 0 FSR = 0 × 43

الحالة الثانية

 قبل تنفيذ التعليمة :

 FSR = 0 × 00

 بعد تنفيذ التعليمة:

 Z = 1 FSR = 0 × 00

**التعليمة MOVWF**

الصيغة MOVWF F

وصف التعليمة: نقل المعطيات من المركم إلى المسجل F.

**التعليمة ADDWF**

**الصيغة** ADDWF f, d

**المعاملات**  0

 

**العملية** d W+(F)

**الأعلام المتأثرة** C, DC, Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 0111 | 00 |

**وصف التعليمة** تقوم هذه التعليمة بعملية جمع حسابية لمحتويات المراكم W مع محتويات المسجل f وتوضع النتيجة في المراكم لأجل d =0 أو توضع في المسجل f نفسه لأجل d =1

 ***الأمثلة:***

***مثال1***:

 ADDWF FSR, 0

 قبل تنفيذ التعليمة:

FSR=0×c2 w=0×17 

بعد تنفيذ التعليمة:

FSR=0×c2 w=0×D9

***مثال 2***:

 ADDW INDF, 1 (عنونة غير مباشرة)

قبل تنفيذ التعليمة:

(FSR)=0×37 FSR=0×c2 w=0×17

بعد تنفيذ التعليمة:

 (FSR)=0×37 FSR=0×c2 w=0×17

***مثال3:***

 ADDWF PCL

**الحالة الأولى**

 قبل تنفيذ التعليمة:

C=x PCL=0×37 W=0×10

بعد تنفيذ التعليمة:

C=x PCL=0×47 W=0×10

**الحالة الثانية :**

قبل تنفيذ التعليمة:

C=x PCH=0×08 PCL=0×F7 W=0×10

 بعد تنفيذ التعليمة:

 C=1 PCH=0×08 PCL=0×07 W=0×10

**التعليمة SUBWF**

 **الصيغة**  SUBWF f, d

**المعاملات**  0

 (0, 1)  d

**العملية**-W d (F)

**الأعلام المتأثرة** C, DC, Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 0010 | 00 |

**وصف التعليمة**

تقوم التعليمة بطرح محتويات المراكم Wمن محتويات المسجل f وتوضع النتيجة في المراكم عندما d=0 أوعندما توضع في المسجل f نفسه عندما d=1

***الأمثلة***: (يتم إنجاز الطرح وفق عملية المتمم الثنائي(complement method))

***مثال 1:***

SUBWF REG1, 1

**الحالة الأولى**

قبل تنفيذ التعليمة:

Z = X C = X W = 2 REG1= 3

بعد تنفيذ التعليمة:

 Z= 0 C= 1 W= 2 REG1= 1

وهذه النتيجة موجبة positive

**الحالة الثانية**

قبل تنفيذ التعليمة:

Z= X C= X W= 2 REG1=2

بعد تنفيذ التعليمة:

Z= 1 C= 1 W= 2 REG1= 0 النتيجة صفر

**الحالة الثالثة**

بعد تنفيذ التعليمة:

Z= X C= X W= 2 REG1= 1

بعد تنفيذ التعليمة:

 Z= 0 C= 0 W= 2 REG1= 1

النتيجة سالبة قيمتها المطلقة هي متمم محتويات REG1= 0000 0001

بعد الإتمام -1=0000 0001 -0000 0010

 +1111 1110

 1111 1111 =FF

**ملاحظة:** مراجعة التعليمة SUBLW بشأن علم الحمل/الاستعارة.

**التعليمة ANDWF**

**الصيغة** ANDWF f, d

**المعاملات**  0

 (0, 1)  d

**العملية** (w).AND (F) d

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 0101 | 00 |

**وصف التعليمة**

 تقوم بإجراء عملية AND المنطقية بين محتويات المراكم W والمسجل f ووضع النتيجة في المراكم عدما d=0أو في المسجل f نفسه للأجل d=1

***الأمثلة***:

***مثال 1:***

ANOWD FSR, 1

قبل تنفيذ التعليمة:

FSR=0×c2 w=0×17 

بعد تنفيذ التعليمة:

FSR=0×02 w=0×17

***مثال 2****:*

 ANOWD FSR , 0

قبل تنفيذ التعليمة:

FSR=0×c2 w=0×17 

بعد تنفيذ التعليمة:

FSR=0×c2 w=0× 02

0001 0111 (0×17)

 AND 1100 0010 (0×c2)

 0000 0010 0×02

***مثال 3:***

 ANDWF INDF, 1

قبل تنفيذ التعليمة:

 (FSR)= 0×5A FSR= 0×C2 W= 0×17

بعد تنفيذ التعليمة:

(FSR)= 0×15 FSR= 0×C2 W= 0×17

(نلاحظ أن طرفي عملية AND هو مسجل العنونة غير المباشرة INDF الذي نشير إليه بالعنوان الموجود في المسجل FSR, ومحتويات هذا العنوان 0×5A).

 **التعليمة IORWF**

**الصيغة** IORWF f, d

**المعاملات**  0≤f≤127

 (0,1) d

**العملية**  W.OR. (f) d

**الأعلام المتأثرة**  Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | ffff | 0001 | 00 |

**وصف التعليمة**  تقوم بإجراء عملية OR المنطقية بين محتويات المراكم w ووضع النتيجة (Result) في المراكم عندما d= 0, أو في المسجل f نفسه لأجل d= 1 .

 **الأمثلة:**

***مثال 1***:

 IORWF Result, 0

قبل تنفيذ التعليمة:

 Result= 0×13 (مسجل أغراض عامة) w= 0×91

بعد تنفيذ التعليمة:

Z= 0 W= 0×93 Result= 0×13

***مثال 2****:*

IORWF RESULT , 1

**الحالة الأولى**

قبل تنفيذ التعليمة:

W=0×91 Result=0×13

بعد تنفيذ التعليمة:

W=0×91 Result=0×93 z=0

**الحالة الثانية**

قبل تنفيذ التعليمة:

W=0×00 Result=0×00

بعد تنفيذ التعليمة:

W=0×00 Result=0×93 z=0

**التعليمة XORWF**

 **الصيغة** XORWF f, d

 **المعاملات** 0≤f≤127

 (0,1) d

 **العملية** d w.XOR.(F)

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 0110 | 00 |

**وصف التعليمة**  إجراء عملية XOR المنطقية بين محتويات المراكم W ومحتويات المسجل f

ووضع النتيجة في المراكم عندما d=0 أو في المسجل f عندما d=1

**الأمثلة:**

***مثال1:***

 XORWF REG, 1

قبل تنفيذ التعليمة:

W=0×B5 REG=AF

بعد تنفيذ التعليمة:

W=0×00 REG=0×1A

***مثال 2*:**

 XORWF REG/ 0

قبل تنفيذ التعليمة:

W= 0×B5 REG= 0×AF

بعد تنفيذ التعليمة:

W= 0×1A REG= 0×AF

***مثال 3:***

XORWF INDF, 1

قبل تنفيذ التعليمة :

(FSR) 0×AF FSR =0×c2 W=0×B5

بعد تنفيذ التعليمة :

(FSR)=0×1A FSR=0×c2 W=0×B5

**التعليمة COMF**

**الصيغة**  COMF f, d

**المعاملات** 0≤f≤127

 (0.1) d

**العملية** d f

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1001 | 00 |

**وصف التعليمة**

 تقوم بإيجاد المتمم الأحادي للمسجل f , أي إبدال كل 1 بـ 0 والعكس ووضع النتيجة في

المراكم عندما d=0 أو وضعها في المسجل f نفسه عندما d=1.

**الأمثلة:**

***مثال 1***:

COMF REG1, 0

قبل تنفيذ التعليمة :

W=0×xx REG1=0×13

بعد تنفيذ التعليمة:

W=0×EC REG1=0×13

***مثال2:***

COMF INDF, 1 (عنونة غير مباشرة)

قبل تنفيذ التعليمة :

(FSR)=0×FF FSR=0×c2

بعد تنفيذ التعليمة:

 (FSR)=0×00 FSR=0×c2 Z=1

 **التعليمة DECF f,d**

**الصيغة** DECF f, d

**المعاملات** 0≤f≤127

(0.1) d

**العملية** (الهدف) (f)-1 d

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 0011 | 00 |

**وصف التعليمة**

تقوم بإنقاص القيمة المخزنة في المسجل f بمقدار واحد ثم تخزن النتيجة في المراكم عندما

d =0, أو في المسجل fعندما d=1

**الأمثلة:**

***مثال1:***

DECF CNT, 1

قبل تنفيذ التعليمة :

 Z=0 CNT=0×01

بعد تنفيذ التعليمة:

 Z=1 CNT=0×00

***مثال2:***

 DECF CNT, 0

 قبل تنفيذ التعليمة :

 W=x Z=0 CNT=0×01

بعد تنفيذ التعليمة:

 W=0×0F Z=1 CNT=0×00

**التعليمة INCF**

**الصيغة** INCF f, d

**المعاملات** 0≤f≤127

(0.1) d

**العملية** (الهدف) (f)+ 1 d

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1010  | 00 |

**وصف التعليمة**

تقوم بزيادة القيمة المخزنة في المسجل f بمقدار واحد ثم تخزن النتيجة في المراكم عندما d=0 أو في المسجل f نفسه عندما d=1

**الأمثلة:**

مثال1:

 INCF CNT ,1

قبل تنفيذ التعليمة :

 Z=0 CNT=0×FF

بعد تنفيذ التعليمة:

 Z=1 CNT=0×00

***مثال2***:

(عنونة غير مباشرة) INCF INDF,1

قبل تنفيذ التعليمة :

Z=0 (FSR)=0×ff FSR=0×c2

بعد تنفيذ التعليمة:

Z=1 (FSR)=0×00 FSR=0×c2

***مثال3:***

 INCF CNT, 0

قبل تنفيذ التعليمة :

Z=0 w=x CNT=0×10

بعد تنفيذ التعليمة:

Z=1 w=0×11 CNT=0×10

**التعليمة DECFSZ**

**الصيغة** DECFSZ f,d

**المعاملات** 0≤f≤127

(0.1) d

**العملية** (الهدف) (f)-1 d

تجاوز التعليمة التالية عندما تصبح النتيجة 0

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1011  | 00 |

**وصف التعليمة**

تقوم بإنقاص القيمة المخزنة في المسجل f بمقدار واحد ثم تخزن النتيجة في المراكم عندما d=0 أو في المسجل f نفسه عندما d=1

فإذا كانت النتيجة 0 تهمل التعليمة التالية للتعليمة DECFSZ وتنفذ كتعليمة NOP وتصبح التعليمة بدورتين 2Tcy حيث يتم عندها تفرع البرنامج وتغيير قيمة المسجل PC,أما إذا كانت النتيجة ليست صفرا تنفذ التعليمة التالية ولا يوجد قفز.

***الأمثلة:***

***مثال 1:***

HERE DECFSZ CNT, 1

GOTO LOOP

 CONTINUE .

 .

قبل تنفيذ التعليمة:

CNT= 0×01 PC= address HERE

بعد تنفيذ التعليمة:

 CNT= 0×00 PC= address CONTINUE تم تحقيق شرط القفز

**الحالة الثانية**

قبل تنفيذ التعليمة:

CNT= 0×02 PC= address HERE

بعد تنفيذ التعليمة:

CNT= 0×01 PC= address HERE + 1 لم يتحقق شرط القفز

**التعليمة INCFSZ**

**الصيغة** INCFSZ f, d

**المعاملات** 0≤f≤127

(0.1) d

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1111  | 00 |

**وصف التعليمة**

تقوم بزيادة القيمة المخزنة في المسجل f بمقدار واحد ثم تخزن النتيجة في المراكم عندما d=0 أو في المسجل f نفسه عندما d=1

فإذا كانت النتيجة 0 تهمل التعليمة التالية للتعليمة INCFSZ وتنفذ كتعليمة NOP وتصبح التعليمة بدورتين 2Tcy حيث يتم عندها تفرع البرنامج وتغيير قيمة المسجل PC,أما إذا كانت النتيجة ليست صفرا تنفذ التعليمة التالية ولا يوجد قفز.

***الأمثلة:***

**مثال 1:**

HERE INCFSZ CNT, 1

GOTO LOOP

 CONTINUE .

 .

.

**الحالة الأولى**

قبل تنفيذ التعليمة:

CNT= 0×FF PC= address HERE

بعد تنفيذ التعليمة:

 CNT= 0×00 PC= address CONTINUE

**الحالة الثانية**

قبل تنفيذ التعليمة:

CNT= 0×00 PC= address HERE

بعد تنفيذ التعليمة:

CNT= 0×01 PC= address HERE + 1

**التعليمة RRF**

**الصيغة** RRF f,d

**المعاملات** 0≤f≤127

 (0,1) d

**العملية**  C bit7;bit7 bit6....;bit0 C

**الأعلام المتأثرة** C

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1100 | 00 |

**وصف التعليمة** يتم تدوير محتويات المسجل f بمقدار خانة واحدة باتجاه اليمين عبر علم الحمل C (أي دوران بتسع خانات), وتوضع النتيجة في المراكم w عندما d=0, أو في المسجل نفسه عندما d= 1, والشكل التالي يبين كيفية تنفيذ هذه التعليمة.

 المسجل f

C

***الأمثلة:***

مثال 1:

 RRF REG1, 0

قبل تنفيذ التعليمة:

C= 0 W= xxxx xxxx REG1= 1110 0110

بعد تنفيذ التعليمة:

C= 0 W=0111 0011 REG1= 1110 0110

***مثال 2:***

 RRF INDF, 1

**الحالة الأولى**

 قبل تنفيذ التعليمة:

C= 1 (FSR )=0011 1010 FSR=0 ×c2 W= xxxx xxxx

بعد تنفيذ التعليمة:

C= 0 (FSR) =1001 1101 FSR=0 ×c2 W= 0 ×17

**الحالة الثانية**

قبل تنفيذ التعليمة:

C= 0 (FSR) =0011 1001 FSR=0 ×c2 W= xxxx xxxx

بعد تنفيذ التعليمة:

C= 1 (FSR) =0001 1100 FSR=0 ×c2 W= 0 ×17

**التعليمة RLF**

**الصيغة**  RLF f,d

**المعاملات** 0≤f≤127

 (0,1) d

**العملية** C biy0, bit0 bit1….; bit7 C

**الأعلام المتأثرة** C

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1101 | 00 |

**وصف التعليمة** يتم تدوير محتويات المسجل f بمقدار خانة واحدة باتجاه اليسار عبر علم الحمل C (أي دوران بتسع خانات), وتوضع النتيجة في المراكم w عندما d=0, أو في المسجل نفسه عندما d= 1, والشكل التالي يبين كيفية تنفيذ هذه التعليمة.

 المسجل f

C

***الأمثلة***

***مثال 1:***

 PLF REG1, 0

قبل تنفيذ التعليمة:

C= 0 REG1= 1110 0110

 بعد تنفيذ التعليمة:

C= 1 W=1100 1100 REG1= 1110 0110

***مثال 2:***

RLF INDF, 1 (عنونة غير مباشرة)

**الحالة الأولى**

قبل تنفيذ التعليمة:

C= 1 (FSR)= 0011 1010 FSR= 0×C2

بعد تنفيذ التعليمة:

C= 0 (FSR) = 0111 1010 FSR= 0×C2

**الحالة الثانية**

قبل تنفيذ التعليمة :

C=0 (FSR)= 1011 1001 FSR= 0×C2

بعد تنفيذ التعليمة:

C=1 (FSR)= 0111 0010 FSR= 0×C2

**التعليمة SEAPF**

**الصيغة**  SWAPF f,d

**المعاملات** 0≤f≤127

 (0,1) d

**العملية** (d< 7:4> الهدف) النيبل العلوي (f<3:0>) النيبل السفلي

 **(**d<3:0> الهدف**)** النيبل السفلي (f<7:4>) النيبل العلوي

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | dfff | 1110 | 00 |

**وصف التعليمة**

النبيل (Nibble) هو مصطلح يعني نصف البايت أي أربع خانات ,والتعليمة SWAPF تقوم بتبد يل النبيل العلوي مع النبيل السفلي والعكس .ويتم وضع نتيجة التبديل في المراكم w عدما d=0 أو المسجل

 ***الأمثلة***

***مثال 1:***

 SWAPF REG1, 0

قبل تنفيذ التعليمة:

W=0× xx, REG1= 0 ×A5

 بعد تنفيذ التعليمة:

W=0× 5A, REG1= 0 ×A5

***مثال2:***

 SWAPF INDF, 1

قبل تنفيذ التعليمة:

 (FSR) =0×20 FSR=0×c2 W=0×17

بعد تنفيذ التعليمة:

 (FSR) =0×02 FSR=0×c2 W=0×17

**التعليمة CLRF**

**الصيغة**  f CLRF

**المعاملات** 0≤f≤127

**العملية** f 00h

 Z 1

**الأعلام المتأثرة** z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | 1fff | 0001 | 00 |

**وصف التعليمة** يتم مسح محتويات المسجل fمما يؤدي إلى توضيع العلم z

**الأمثلة:**

***مثال 1:***

CLRF FLAG-REG

قبل تنفيذ التعليمة:

 FLAG-REG=0×5A (المسجل أغراض عامة)

بعد تنفيذ التعليمة:

 FLAG-REG=0×00

Z=1

***مثال2***:

CLRF INDF

 قبل تنفيذ التعليمة:

 (FSR) =0×AA FSR=0×c2

 بعد تنفيذ التعليمة:

 FSR=0×c2

 (FSR)=0×00

 Z=1

**التعليمة CLRW**

**النص** CLRW

**العاملات** لا يوجد

**العملية** 00h w

 Z 1

**الأعلام المتأثرة** z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| xxxx | 0xxx | 0001 | 00 |

**وصف التعليمة** يتم مسح محتويات المراكم wمما وتفعيل (توضيع) العلم z

***الأمثلة:***

***مثال1:***

 CLRW

 قبل تنفيذ التعليمة:

 W=0×5A

 بعد تنفيذ التعليمة:

W=0×00

Z=1

**التعليمة NOP**

**الصيغة** NOP

**المعاملات** لا يوجد

**العملية** لا يوجد

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 0000 | 0xx0 | 0000 | 00 |

**وصف التعليمة** لا تنفذ هذه التعليمة أي عملية.

**الأمثلة**

***مثال 1:***

HERE NOP

قبل تنفيذ التعليمة:

PC= address HERE

بعد تنفيذ التعليمة

 PC= address HERE+1

**10-4 مجموعة تعليمات الخانة** bit oriented operation

**التعليمة BCF**

**الصيغة**  BCF f, d

**المعاملات**  0≤f≤127

0≤b≤7

**العملية**  0 →f<b>

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | bfff | 00bb | 01 |

**وصف التعليمة** يتم تصفير الخانة رقم b من المسجل f.

***الأمثلة:***

***مثال 1:***

BC F FLAG-REG, 7

قبل تنفيذ التعليمة :

FLAG-REG= 0×C7h= 1100 0111

بعد تنفيذ التعليمة :

FLAG-REG= 0×47h= 0100 0111

***مثال 2:***

 BCF INDF, 3

قبل تنفيذ التعليمة:

(FSR)= 0×2F FSR= 0×C2; W= 0×17

بعد تنفيذ التعليمة:

(FSR)= 0×27 FSR= 0×C2 W= 0×17

**التعليمة BSF f, b**

**الصيغة BSF f ,b**

**المعاملات**  0≤f≤127

0≤b≤7

**العملية**  0 →f<b>

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | bfff | 01bb | 01 |

**وصف التعليمة** تقوم بتفعيل (set) الخانة رقم b من المسجل f.

**الأمثلة :**

***مثال1:***

BSF FLAG-REG, 7

قبل تنفيذ التعليمة:

FLAG-REG=0×0A=(0000 1010) B

بعد تنفيذ التعليمة :

FLAG-REG=0×8A = (0000 1010)

***مثال2:***

(عنونة غير مباشرة) BSF INDF ,3

قبل تنفيذ التعليمة:

(FSR)=0×20 FSR=0×c2 W=0×17

بعد تنفيذ التعليمة:

(FSR)=0×28 FSR=0×c2 W=0×17

**التعليمة BTFSC**

**الصيغة**  BTFSC f, b

**المعاملا ت** 0≤f≤127

 0≤b≤7

**العملية**  إذا كان (f<b>=0) يتم تجاوز التعليمة التالية

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff | bfff | 10bb | 01 |

**وصف التعليمة**

إذا كانت الخانة رقم b من المسجل f مساوية للصفر "0" عندها يتم تجاوز التعليمة التالية وتنفذ كتعليمة NOP, أي تصبح تعليمة BTFSC كتعليمة تفريغ للبرنامج ذّات دورتي تعليمة (2Tcy) عند التنفيذ.

أما إذا كانت هذه الخانة "1" فيتم تنفيذ التعليمة التالية مباشرة BTFSC .

**الأمثلة :**

***مثال 1***

 HERE BTFSC FLAG,4

 FALSC Go To PROCESS\_CODE

 TRUE

 .

 .

 الحالة الأولى

قبل تنفيذ التعليمة :

FLAG =xxx0xxx PC=address HERE (مسجل أغراض عامة)

بعد تنفيذ التعليمة :

 وحيث أن الخانة FLAG<4> مساوية للصفر فإن عداد البرنامج سيأخذ القيمة

PC= address TRUE

الحالة الثانية

قبل تنفيذ التعليمة :

 FLAG =xxx1xxxx PC = address [HERE]

بعد تنفيذ التعليمة :

 وحيث أن الخانة FLAG(4) مساوية للواحد فإن عداد البرنامج سيأخذ القيمة

PC = address (FALSE)

**التعليمة BTFSS**

**الصيغة**  BTFSS f, b

**المعاملات** 0≤f≤127

  0≤b≤7

ا**لعملية**  إذا كان (f<b>) = 1 يتم تجاوز التعليمة التالية

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| ffff  | bfff  | 11bb  |  01  |

**وصف التعليمة** إذا كانت الخانة رقم b من المسجل f مساوية للواحد "1" عندها يتم تجاوز التعليمة التالية التي ستنفذ كتعليمة NOP, أي تصبح BTFSS تعليمة تفريغ للبرنامج ذات دورتي تعليمة 2Tcy عند التنفيذ

أما إذا كانت هذه الخانة "0" فيتم تنفيذ التعليمة التالية مباشرة لتعليمة BTFSS

***الأمثلة:***

***مثال1:***

HERE BTFSS FLAG, 4

FLASE Go To PROCESS-CODE

TRUE

الحالة الأولى

قبل تنفيذ التعليمة :

FLAG = xxx0xxxx PC = address [HERE]

بعد تنفيذ التعليمة :

 وحيث أن FLAG <4> = 0 فإن عداد البرنامج سيأخذ القيمة

PC = address <false>

الحالة الثانية

قبل تنفيذ التعليمة :

FLAG = xxx1 xxxx PC = address [HERE]

بعد تنفيذ التعليمة :

 وحيث أن FLAG <4> =1 فإن عداد البرنامج سيأخذ القيمة

PC= address <TRUE>

**10-5 مجموعة تعليمات التحكم وتعليمات الثوابت**

 **Literal and Control Operation**

**التعليمة MOVLW**

 **الصيغة**  MOVLW k

 **المعاملات** 0≤k≤255

  **العملية** w k

 **الأعلام المتأثرة** لا يوجد

 **شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 00xx  | 11  |

**وصف التعليمة :** تقوم بتحميل القيمة الثابتة k ذات الثمانية خانات في مسجل العمل w

***الأمثلة:***

***مثال 1:***

MOVLW 0×5A

بعد تنفيذ التعليمة :

w =0×5A

***مثال 2:***

MOVLW MYREG

قبل تنفيذ التعليمة :

w = 0×10

Address of MYREG = 0×37 حيث أن MYREG هو نموذج لموقع في ذاكرة المعطيات (data memory)والهدف من التعليمة تحميل عنوان موقع ما في المراكم w

بعد تنفيذ التعليمة :

w = 0×37

***مثال 3***

MOVLW HIGH (LU- TABLE)

قبل تنفيذ التعليمة :

w =0×10

عنوان بداية جدول البحث Lookup table هوTABLE وهو LU- TABLE =0×9375

بعد تنفيذ التعليمة :

w = 0×93

**ملاحظة**: جداول التفتيش (البحث) Lookup table هي قطع برمجية فيها مجموعة من التعليمات, سندرسها بالتفصيل في وقتها.

**التعليمة ADDLW**

 **الصيغة** ADDLW k

 **المعاملات** 0≤k≤255

 **العملية** w w + k

 **الأعلام المتأثرة** C, DC, Z

 **شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 111x  | 11  |

 **وصف التعليمة**: تقوم هذه التعليمة بجمع محتويات المراكم w مع الثابت ثماني الخانات k وتوضع النتيجة في المراكم w حصراً .

 ***الأمثلة:***

***مثال 1*:**

 ADDLW 0×15

قبل تنفيذ التعليمة :

W=0×10

بعد تنفيذ التعليمة :

w = 0×25

***مثال2:***

 ADDLW MYREG (جمع W مع عنوان موقع ذاكري للمعطيات)

قبل تنفيذ التعليمة :

W=0×10

Address of MYREG=0×37

حيث MYREG هو عنوان موقع ذاكري في ذاكرة المعطيات

بعد تنفيذ التعليمة :

w = 0×47

 ***مثال 3:***

ADDLW HIGH (LU\_TABLE) (جمع wمع عنوان موقع ذاكرة البرنامج)

قبل تنفيذ التعليمة :

W=0×10

Address LU-TABLE =0×9375

حيث LU- TABLE وهي لافتة تقابل عنوان في ذاكرة البرنامج.

بعد تنفيذ التعليمة :

W=0×A3

***مثال 4:***

 ADDLW PCL

قبل تنفيذ التعليمة:

 W=0×10 عنوان المسدل 0×02 =PCL

بعد تنفيذ التعليمة :

W=0×10

**التعليمة SUBLW**

**الصيغة** SUBLW k

**المعاملات** 0≤k≤255

**العملية** w k-w

**الأعلام المتأثرة** C, DC, Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 110x  | 11  |

**وصف التعليمة** يتم طرح محتويات المراكم w من الثابتk ثماني الخانة وفق طريقة المتمم الثنائي (2's Complement Method)ويتم وضع نتيجة الطرح في المراكم w .

***الأمثلة:***

***مثال 1:***

 SUBLW 0×02

**الحالة الأولى:**

قبل تنفيذ التعليمة :

Z =x c = x W =0×01

بعد تنفيذ التعليمة :

Z=0 c=1 w =0×01 (النتيجة موجبة)

**الحالة الثانية :**

قبل تنفيذ التعليمة :

Z =x c = x W =0×02

بعد تنفيذ التعليمة :

 Z=1 c=1 w =0×00 (النتيجة صفر)

**الحالة الثالثة:**

 قبل تنفيذ التعليمة :

Z =x c = x W =0×03

بعد تنفيذ التعليمة :

Z=0 c=0 w =0×FF (النتيجة سالبة)

***مثال 2:***

SUBLW MYREG

قبل تنفيذ التعليمة :

W =0×10

 عنوان مسجل المعطيات MYREG هو 0×73

بعد تنفيذ التعليمة :

C = 1 W =0×27 الناتج سالب

**ملاحظة**: من المفيد هنا أن نذكر أن العلم C المسمى بعلم الاستعارة (الطرح ) / الحمل (الجمع) ذو منطق متعاكس Carry / borrow أي عندما يكون هناك حمل سيكون C=1 وعندما لا تحدث استعارة سيكون C=1 والعكس بالعكس, وذلك يعود لأن أصل عملية الطرح تتم بالجمع من المتمم الثنائي. لذا وخروجا من أللبث في هذا الموضوع نقدم الجدول التالي الذي يبين نتيجة عملية الطرح:

|  |  |  |
| --- | --- | --- |
| **النتيجة** | **Z** | **C** |
| صفر | 1 | X |
| سالب | 0 | 0 |
| موجب | 1 | 0 |

**التعليمة ANDLW**

**الصيغة** ANDLW k

**المعاملات** 0≤k≤255

**العملية** w →w. AND. k

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 1001  | 11  |

**وصف التعليمة** : يتم إنجاز عملية AND المنطقية بين محتويات المراكم W والثابت k ثماني الخانة ثم توضع النتيجة في المراكم w .

***الأمثلة:***

***مثال 1:***

ANDLW 0×5F

قبل تنفيذ التعليمة :

W =0×3A 0×5F 0101 1111 0×3A 1010 0011 بعد تنفيذ التعليمة : 0×03 0000 0011

W =0×03

***مثال 2:***

ANDLW MYREG

قبل تنفيذ التعليمة :

 0×37 = MYREG عنوان الموقع W =0×3A

بعد تنفيذ التعليمة :

W =0×23

 **التعليمة IORLW**

**الصيغة** IORLW k

**المعاملات** 0≤k≤255

**العملية** w →w. OR .k

**الأعلام المتأثرة** Z

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 1000  | 11  |

**وصف التعليمة** : تنفذ هذه التعليمة عملية OR المنطقية بين محتويات المراكم w وخانات الثابت k الثمان , أما نتيجة العملية فتوضع في المراكم w .

***الأمثلة:***

***مثال 1:***

IORLW 0×25

قبل تنفيذ التعليمة :

 W =0×9A

بعد تنفيذ التعليمة :

Z =0 W =0×BF

***مثال 2:***

IORLW MYREG

قبل تنفيذ التعليمة:

W= 0×9A عنوان الموقع 0×37 MYREG

بعد تنفيذ التعليمة:

Z= 0 W= 0×9F

***مثال 3:***

IORLW HIGH (LU, TABLE)

قبل تنفيذ التعليمة:

 عنوان تعليمة في ذاكرة البرنامج المقابلة للافتة LU-TABLE هو 0×9375

 W= 0×9A

بعد تنفيذ التعليمة:

Z= 0 W= 0×9B

***مثال 4:***

IORLW 0×00

قبل تنفيذ التعلية:

W= 0×00

بعد تنفيذ التعليمة:

Z= 1 W= 0×00

**التعليمة XORLW**

 **الصيغة** XORLW k

 **المعاملات** 0≤k≤255

 **العملية** w →w × OR k

 **الأعلام المتأثرة**  Z

 **شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 1010  | 11  |

**وصف التعليمة** تنفذ هذه التعليمة عملية OR المنطقية بين محتويات المراكم w وخانات الثابت k الثمان , وتوضع النتيجة في المراكم w .

***الأمثلة:***

***مثال 1:***

XORLW 0×AF

قبل تنفيذ التعليمة :

W =0×B5

بعد تنفيذ التعليمة :

Z=0 W =0×1A

**التعليمة CLRWDT**

**الصيغة**  CLRWDT

**المعاملات** لا يوجد

**العملية** 00→WDT

 0→WDT prescaler count

 مقسم المؤقت WDT

1→TO

1→PD

**الأعلام المتأثرة** PD TO

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 0100  | 0100  | 0000  | 00  |

**وصف التعليمة** : تقوم هذه التعليمة بتصفير مؤقت المراقبة , بالإضافة إلى تصفير خانة المقسم الخاص بالمؤقت WDT وتقوم هذه التعليمة أيضا بتفعيل الخانتين PD, TO

***الأمثلة:***

***مثال 1:***

CLRWDT

قبل تنفيذ التعليمة:

 WDT counter= x

WDT prescaler= 1:128 (نسبة القسم)

بعد تنفيذ التعليمة:

WDT counter = 0×00

WDT prescaler count = 0

= 1 

 =1

WDT perscaler = 1:128

**التعليمة SLEEP**

**الصيغة**  SLEEP

**المعاملات** لا يوجد

**العملية** 00h→WDT

0h→WDT prescaler count

 1→

 0→

**الأعلام المتأثرة** 

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 0011  | 0110  | 0000  | 11  |

**وصف التعليمة** يتم تصفير كل من مؤقت المراقبة (watchdog timer) وعداد المقسم (prescaller count) ويتم تصفير خانة حالة انخفاض الطاقة 

power down status git) ) أما خانة الحالة (time-out) فيتم تفعيلها.

ويتم وضع المعالج في نمط الـ (الراحة) SLEEP حيث سيتوقف الهزاز (OSCILLATOR).

***الأمثلة:***

***مثال 1:***

SLEEP

**التعليمة CALL**

**الصيغة** CALL K

**المعاملات** 0≤k≤2047

**العملية** PC+1→TOS

 قمة المكدس TOP OF STACK

→ PC (10:0) K

**(**PCLATCH<4:3>**)**→PC<12:11>

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| kkkk  | kkkk  | 0kkk  | 10  |

**وصف التعليمة** عندما يستدعي برنامج فرعي ما (subroutine) أولاً يتم دفع عنوان العودة (pc+1) إلى المكدس stack, وهو العنوان الذي سيتم تنفيذ تعليمته بعد العودة من البرنامج الفرعية, ويتم تحميل الإحدى عشرة خانة (الثابت k) إلى الخانات الأولى من عداد البرنامج أي pc<10:0>. أما الخانتان العلويتان لعداد البرنامج pc <12:11> فيتم الحصول عليهما من الخانتين PC LATH<4:3>, وتذكر بأن هذه التعليمة هي ذات دورتي ساعة Tcy .

***الأمثلة :***

مثال 1:

HERE CALL THERE

قبل تنفيذ التعليمة:

PC= address (HERE)

بعد تنفيذ التعليمة:

TOS= HERE + 1

PC= address (THERE)

**التعليمة GOTO**

**الصيغة**  GOTO k

**المعاملات** 0≤k≤2047

**العملية** → PC<10:0> K

PC LATCH<4:3>→PC <12:11>

**الأعلام المتأثرة**  لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| Kkkk  | Kkkk  | 1kkk  | 10  |

**وصف التعليمة** تقوم هذه التعليمة بعملية تفريغ للبرنامج بشكل غير مشروط unconditional branch, ويتم تحيل الثابت k ذو الإحدى عشرة خانة إلى خانات إعداد البرنامج pc<0:10>

أما الخانتان العلويتان لعداد البرنامج pc<11:12> فتحمل من خانتي المسجل PCLATCH <4:3> ونذكر بأن هذه التعليمة هي تعليمة ذات دورتي تعليمة (2TCY).

***الأمثلة:***

***مثال 1:***

GOTO THERE

بعد تنفيذ التعليمة:

PC = address (THERE)

 **التعليمة RETURN**

**الصيغة** RETURN

**المعاملات**  لا يوجد

**العملية** TOS →PC

 حيث TOS هي مصطلح مختصر للعبارة Top Of Stake

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 1000  | 0000  | 0000  | 00  |

**وصف التعليمة** هذه التعليمة مخصصة للعودة من برنامج فرعي (Subbroutione)

وعند تنفيذ هذه التعليمة يتم سحب القيمة المخزنة أخيرا في قمة المكدس والتي تمثل عنوان العودة الذي تم دفعه عند تنفيذ عملية الاستدعاء, حيث يعاد تحميلها إلى عداد البرنامج PC

وهذه التعليمة ذات دورتي تعليمة (2Tcy) عند التنفيذ .

**الأمثلة:**

***مثال:***

HERE RETURN

بعد تنفيذ التعليمة :

PC = TOS

**التعليمة RETFIF**

**الصيغة** RETFIE

**المعاملات** لا يوجد

**العملية** TOS →PC ( قمة المكدس )

 GIE → 1 ( GIEخانة تمكين المقاطعة العامة)

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 1001  | 0000  | 0000  | 00  |

**وصف التعليمة:** هذه التعليمة هي تعليمة عودة من المقاطعة Interrupt .

 ويتم عند تنفيذها نقل العنوان ذو الـ 13 bit الموجود في قمة المكدس (TOS) إلى عداد البرنامج PC , بالإضافة إلى ذلك يتم تفعيل خانة تمكين المقاطعة العامة (Global Interrupt Enable bot ) الموجودة في المسجل <7> option وهي تعليمة ذات دورتين (2Tcy).

**الأمثلة:**

***مثال:***

 RETFIE

بعد تنفيذ التعليمة :

PC = TOS

GIE =1

**التعليمة RETLW**

**الصيغة** RETLW k

**المعاملات** 0≤k≤255

**العملية** w → k

 ( قمة المكدس) TOS → PC

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| KKKK  | KKKK  | 01KK  | 11  |

**وصف التعليمة** تقوم هذه التعليمة بتحميل الثابت K ثماني الخانة إلى المراكم W وكذلك تسحب (POP) قيمة عداد البرنامج من المكدس (TOS) والذي يشير إلى التعليمة التي ستنفذ بعد العودة وهي تعليمة ذات دورتي Tcy وكثيرا ما تستخدم هذه التعليمة مع جداول التفتيش

(Lookuptables).

**الأمثلة:**

**مثال1:**

HEAR CALL TABEL

 هنا يتم تحميل المراكم W بقيمة .

 ضبط Offset .

 .

TABLE ADDWF PC; W=Offset

 RETLW K1; W=0 قيمة معينة تقابل K

 RETLW K2; W=1 قيمة معينة تقابل K

 RETLW K3;

 .

 .

 .

 .

 RETLW Kn;

**ملاحظة**

يرجى لتحقيق فهم أعمق لهذه التعليمة مراجعة جداول التفتيش في فصل تنظيم وتصميم البرنامج.

قبل تنفيذ التعليمة:

W= 0×02

بعد تنفيذ التعليمة:

 PC= TOS= HERE + 1 W=K4

**10-6 تعليمات خاصة**

**التعليمة Option**

 **ملاحظة** لا تستخدم هذه التعليمة مع متحكمات عائلة PIC16CXX بل مع متحكمات عائلةPIC16C5X .

 **الصيغة** Option

**المعاملات**  لا يوجد

**العملية** w →option

**الأعلام المتأثرة** لا يوجد

 **شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 0010  | 0110  | 0000  | 00  |

**وصف التعليمة**

يتم نسخ محتويات المراكم w في المسجل option ,وهذه التعليمة تستخدم مع العائلة PIC16C5X حيث إن المسجل option هو مسجل قابل للقراءة والكتابة (readable/writeable) ويمكن للمستخدم أن يعنونه بشكل مباشر.

**ملاحظة** :لتأمين الموافقة مع أفراد عائلة PIC16CXX ينصح دائما بعدم استخدام هذه التعليمة والتي نستعيض عنها بالتعليمتين :

MOVLW B'0000 0111'

MOVWF option

**الأمثلة:**

***مثال 1:***

Vovlw B'0000 0111'

 OPTIPN

بعد تنفيذ التعليمة:

 المسجل Option بأخذ القيمة 0000111

**التعليمة TRIS**

**ملاحظة** :تستخدم هذه التعليمة مع العائلة PIC16C5X .

**الصيغة**  TRIS f

**المعاملات** 5≤f≤7

**العملية** w TRIS register

 حيث TOS هي مصطلح مختصر للعبارة Top Of Stake

**الأعلام المتأثرة** لا يوجد

**شيفرة التعليمة**

|  |  |  |  |
| --- | --- | --- | --- |
| 1000  | 0000  | 0000  | 00  |

**وصف التعليمة**

تقوم هذه التعليمة بنقل محتويات المراكم w إلى مسجل التحكم بالنافذة f حيث f=5,6,7 وهي قيم تمثل العناوين المقابلة للمسجلات PORTA,PORTB,BORTC على الترتيب, وهذه التعليمة خاصة بعائلة PIC16CSX .

**ملاحظة**

للموافقة مع تعليمات العائلة PIC16cx ينصح بعدم استخدام هذه التعليمة.

**الأمثلة :**

***مثال1:***

TRIS PORTB

قبل تنفيذ التعليمة :

W =0×A5

بعد تنفيذ التعليمة :

TRISE=0×A5