

الميكرو مراقب PIC16F84A

- (١) عموميات
- (٢) الهيكلية القاعدية للميكرو مراقب PIC16F84A
- (٣) تنظيم ذاكرة الميكرو مراقب (PIC16F84A)
- (٤) السجلات الأساسية
- (٥) لغة مجمع الميكرو مراقب
- (٦) هيكلية برنامج بلغة مجمع الميكرو مراقب
- (٧) كتابة برنامج: للتهيئة (برمجة المداخل والمخارج) و برنامج فرعي للتأجيل
- (٨) برنامج رئيسي للتحكم في مصابيح (للتحكم على مصباح بزر ضاغط)

عموميات

- ميكرو مراقبات شركة MICROCHIP تأخذ التسمية PIC والتي يمكن تأخذ عدة معاني و معناها حسب الصانع: مراقبة الربط الخارجي/التحكم في الأجهزة المحيطة :

Peripheral Interface Controller

الميكرو مراقبات لها RISC (Reduced Instruction Set Computer)
بنية قاعدية تسمى -

(هذه البنية الهدف منها تقليل التعليمات لزيادة سرعة التنفيذ لكن علي حساب عوامل أخرى كقوة المعالجة مثلا ...)

الـ: PIC له عدد من التعليمات محدود بـ : ٣٣ أو ٣٥.

- تعيين الـ: PIC من الشكل : xxXXyy-zz

: العائلة حاليا “ ١٢ ، ١٤ ، ١٦ ، ١٧ ، ١٨ ” بحيث ١٦ : تشير الى فئة المدى
xx المتوسط

mid Range

yy : نوع الميكرو مراقب

XX : نوع ذاكرة البرنامج

C- EEPROM أو EPROM

PROM : R-

FLASH : F-

zz : قيمة التواتر الأعظمي للتشغيل

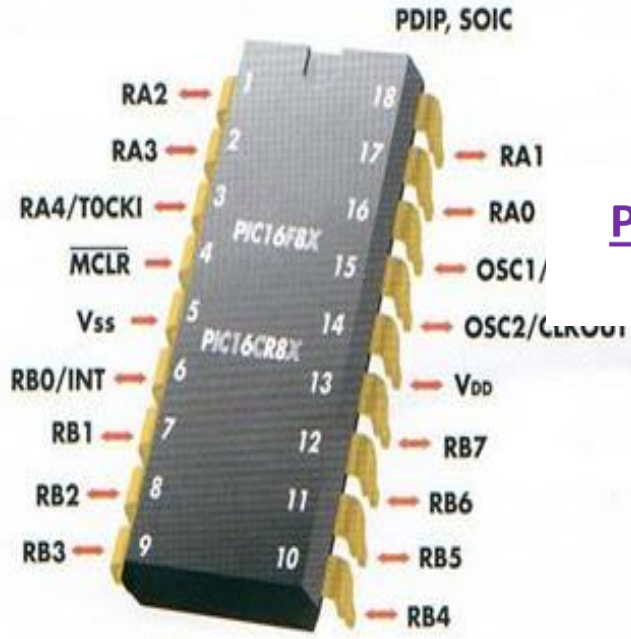
A تمثل 20 Mhz

مثال PIC 16F 84 -10 : قيمة التواتر الأعظمي للتشغيل 10Mhz ←

لدراسة و إستغلال الميكرو مراقبات PIC ننتقل من دراسة 16F84 الذي يعتبر كإنبطاقة لدراسة باقي الأنواع

- هو عنصر تقني إلكتروني مندمج، يظهر الوظائف التي لها دور معالجة المعلومات : عمليات حسابية (جمع، ضرب، ...) أو منطقية (أو، و...) بين إشارات دخول مختلفة التي تسمح بتوليد إشارات خروج .
- يمكن إنجاز هذه الوظائف بواسطة دارات مندمجة تماثلية أو منطقية. لكن عندما يصبح العنصر التقني معقد، ويصبح من الضروري إنجاز معالجة معلومات كثيرة، من الأبسط استعمال الميكرو مراقب.

الميكرو مراقب هو جيل جديد و مطور من الميكرو معالج وأن جميع ملحقات المعالج تم وضعها في شريحة واحدة ومن هنا بناء دارة تحتوي على ميكرو مراقب تكون بسيطة وصغيرة وليست معقدة.



التركيب الخارجى للـ PIC 16F84

الأرجل : RA0, RA1, RA2, RA3 & RA4 عبارة عن ← PORTA

الأرجل : RB0, RB1, RB2, RB3, RB4, RB5, RB6 & RB7 عبارة عن ← PORTB

الأرجل : OSC1 (CLKIN) & OSC2 (CLKOUT) ← عبارة عن طرفي توصيل دارة اهتزازية خارجية :

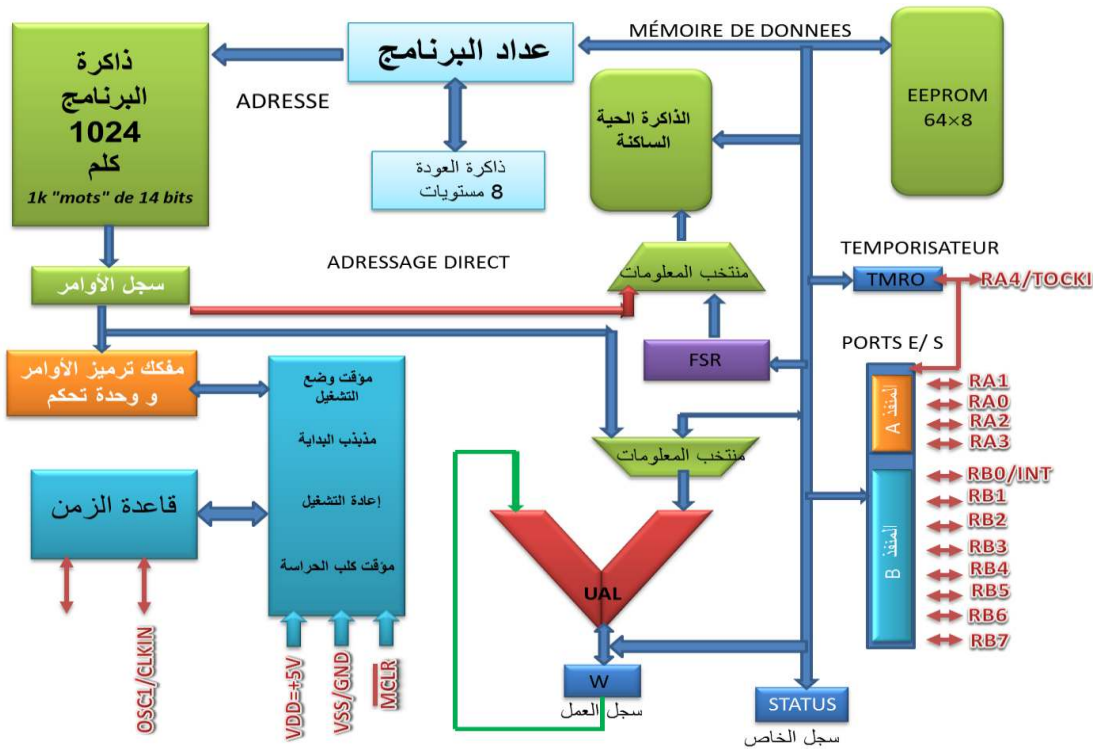
كريستاله Quartz تردد لها 4 MHz

الرجل : V_{DD} عبارة عن ← مدخل الطاقة وتوصل على 5 فولت

الرجل : V_{SS} عبارة عن ← الأرضي

الرجل : MCLR ← عبارة عن RESET لإعادة تشغيل البرنامج

التركيب الداخلى للـ PIC 16F84

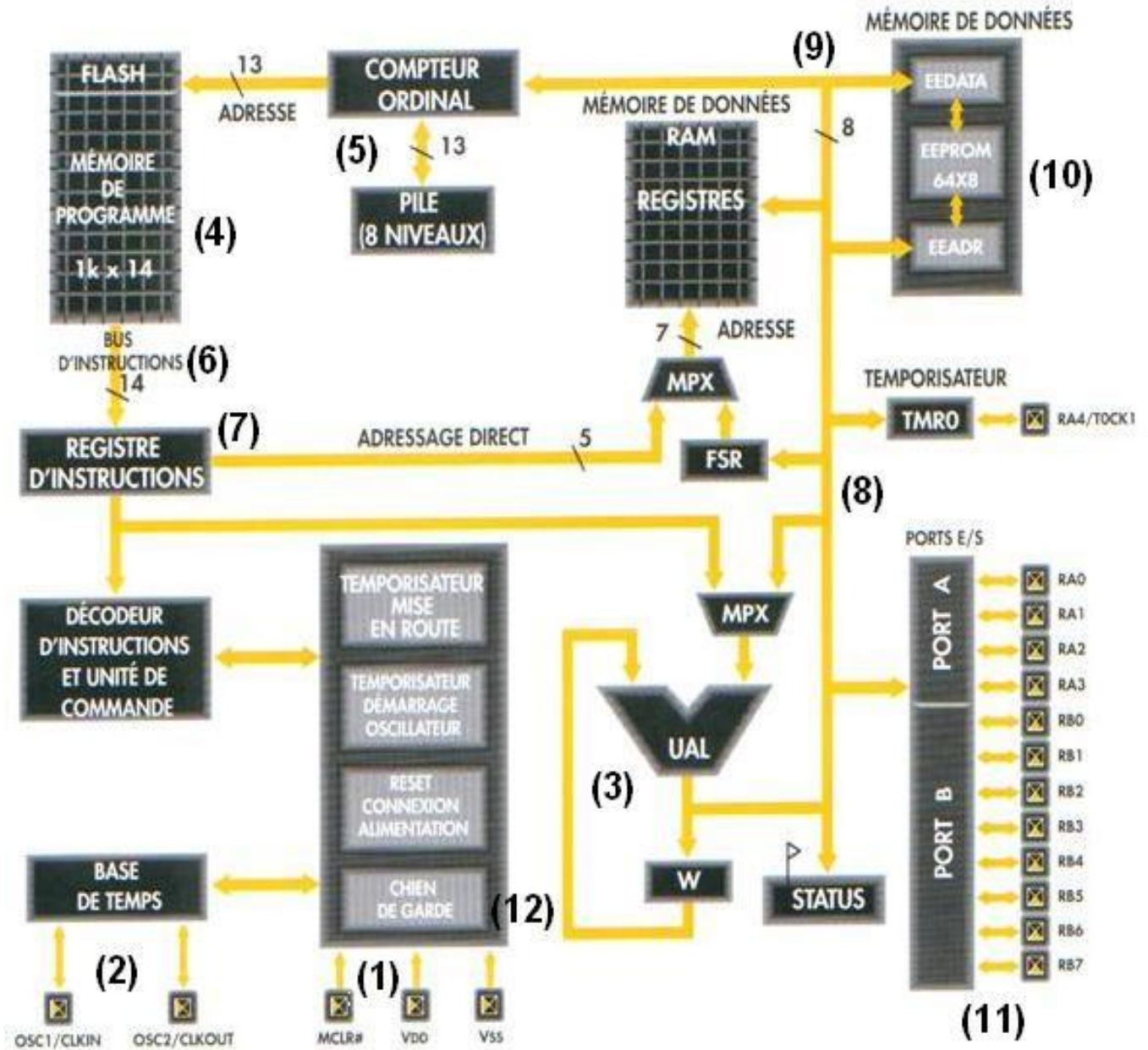


يتكون الميكرومراقب أساسا من العناصر التالية

- (١) – دارة التغذية و إعادة التهيئة : V_{SS} , V_{DD} & MCLR
- (٢) – إشارة الساعة : Quartz 4 Mhz
- (٣) – وحدة المعالجة او الوحدة المنطقية الحسابية: UAL
- (٤) – ذاكرة البرنامج : FLASH تحتوي البرنامج
- (٥) – عداد البرنامج : Compteur Ordinaire
- (٦) – ناقل أوامر البرنامج Bus d'Instructions
- (٧) – السجل الحامل للأمر في طور المعالجة
- (٨) – ناقل المعلومات Bus de Données
- (٩) – الذاكرة : RAM تحتوي على السجلات الخاصة و ٦٨ خانة للمعلومات
- (١٠) – ذاكرة المعطيات : EEPROM تخزن بها المعطيات بشكل دائم
- (١١) – منفذان : المرفأ A والمرفأ B

تلخص وظيفة الميكرو مراقب في ما يلي

- الذهاب للبحث عن التعليمة في الذاكرة (نزوده بعنوان الخلية المعنية و إشارة القراءة).
- فك ترميز التعليمة وتنفيذها.
- تحقيق انتقال المعطيات من أو إلى الذاكرة أو نحو المداخل و المخرج.
- حساب عنوان التعليمة الموالية و ضمان تتابع البرنامج .



تنظيم ذاكرة الميكرو مراقب:

تنقسم ذاكرة الميكرو مراقب الى :

- ذاكرة البرنامج - الذاكرة RAM - ذاكرة المعطيات

ذاكرة البرنامج FLASH : تحتوي على أوامر وتعليمات البرنامج وأيضا مؤشر الوضع في الصفرو مؤشر المقاطعة .

الذاكرة RAM : تحتوي على ٦٨ بايت للمعطيات وعلى جميع السجلات اللازمة لتسيير النظام واستعمال الملحقات الداخلية والمنافذ.

ذاكرة البرنامج FLASH

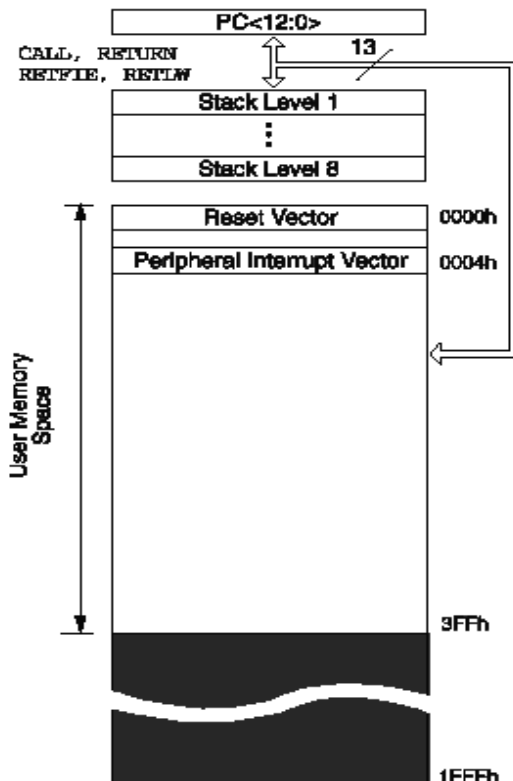
- تتكون ذاكرة البرنامج من ١٤ كلمة ذات ١٤ بيت.

مثال : `movlw 0x0A` 110000 00001010



- RESET $\Leftarrow$ الميكرو مراقب يبدأ من جديد من الموضع 0000H (مؤشر الوضع في الصفرو).

- INTERRUPTION $\Leftarrow$ الميكرو مراقب يذهب إلى الموضع 0004H (مؤشر المقاطعة)



الذاكرة RAM

مقسمة الى منطقتين

تتكون الأولى من ٢٤ خانة (٨ بيت) خاصة بسجلات تسيير النظام، ١٢ منها في صفحة 0 BANK ذات العناوين من ٠٠ إلى 0B و ١٢ في صفحة 1 BANK ذات العناوين من ٨٠ إلى 8B.

تتكون الثانية من ٦٨ خانة معنونة من 0C إلى 4F وتدعى سجلات التشغيل.

File Address	بنك 0	بنك 1	File Address
00h	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	80h
01h	TMR0	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ⁽¹⁾	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	36 General Purpose registers (SRAM)	Mapped (accesses) in Bank 0	
4Fh			CFh
30h			B0h
7Fh			FFh
	Bank 0	Bank 1	

■ Unimplemented data memory location; read as '0'.

Note 1: Not a physical register.

المساحة المتبقية باللون الرمادي فهي غير مستعملة.

PIC 16F84 لسجلات الأساسية للميكرومراقب

توجد أغلب هذه السجلات في الذاكرة RAM قسم منها

في الصفحة 0 Bank و القسم الآخر في الصفحة 1 Bank

كما أن بعضها يوجد في الصفحتين معا لتسهيل الوصول إليها

و عددها ١٦ وهي ذات وظائف مميزة

دورها في تسيير الميكرومراقب، أهمها :

سجل الحالة

/السجلان TRISA & TRISB/السجل المنفذان A B/

سجل العمل

سجل الخيارات /OPTION REG/

سجل الإعدادات الداخلية /INTCON/

سجل الإعدادات المادية _CONFIG

١/ سجل الحالة: (STATUS) Registre d'état

يعطي دلالات متعددة عن نتائج العمليات الحسابية أو حالة الميكرومراقب، كما يسمح بتحديد Bank0 و Bank1 للوصول إلى سجلات الإعداد.

٢/ السجلان : TRISA & TRISB

يعينان اتجاه كل بيت (خط) في المنفذين A & B، بحيث يبرمج كل بيت كمدخل إذا أعطى القيمة ١ و كمخرج إذا أعطى القيمة ٠.

٣/ السجل المنفذان PORTA & PORTB

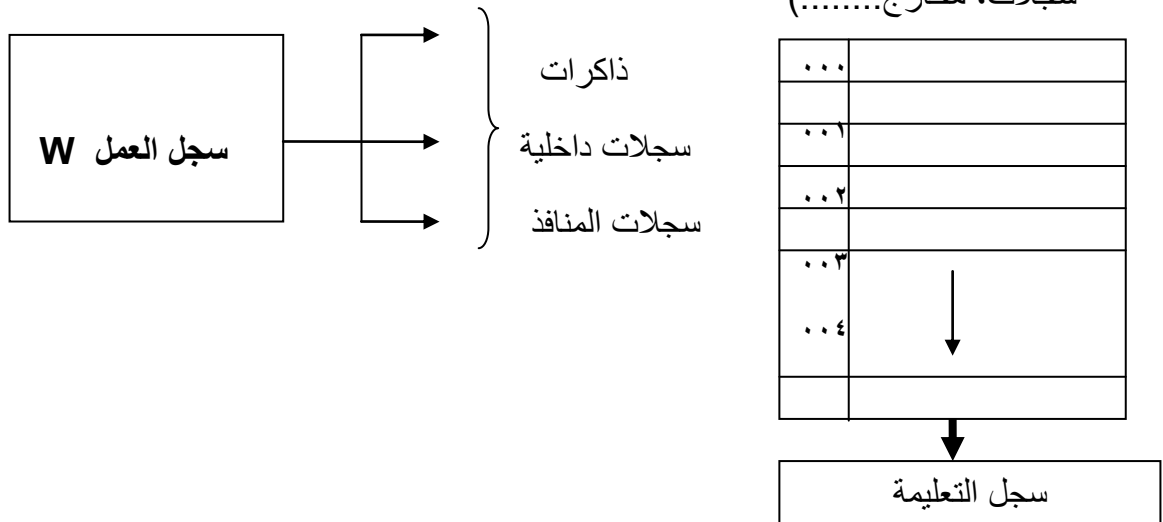
يمكن الاتصال الخارجي وتحتوي على ١٣ بيت (خط) من المداخل والمخارج موزعة كما يلي :
٥ على A : RA0,.....,RA4 و ٨ على B : RB0,.....,RB7.

يقوم TRISA & TRISB بتحديد اتجاه كل منهما كما سبق ذكره.

٤/ سجل العمل : Registre de Travail (W)

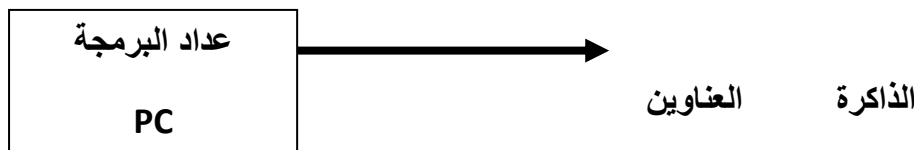
سجل العمل W: يستعمل كوسيط يلجأ إليه خاصة في العمليات الحسابية.

بواسطة التعليمات يقوم هذا السجل بنقل المعلومات إلى جميع العناصر الداخلية (ذاكرات، سجلات، مخارج.....)



٥, ٤: عداد البرمجة :

- يحتوي على عنوان التعليمة من أجل البحث عليها في ذاكرة البرنامج
- ترسل إلى سجل التعليمات لفكها أي معرفة طبيعتها
- التعليمة تطبق و العداد يتقدم بخطوة



Registre d'état (STATUS) سجل الحالة:

7	6	5	4	3	2	1	0
		RPO	$\overline{TO}$	$\overline{PD}$	Z	DC	C

Register page zero : يسمح بتحديد إحدى الصفحتين Bank0 , Bank1 بحيث من أجل RP0=0 تحدد الصفحة Bank0

Time out : = ٠ في حالة انقضاء زمن التأجيل الذي يفرضه الحارس watch dog

Power Down : يشير إلى حالة سبات الميكرو مراقب (بواسطة التعليلة sleep).

Carry : يشير إلى الإحتفاض خلال عملية حسابية.

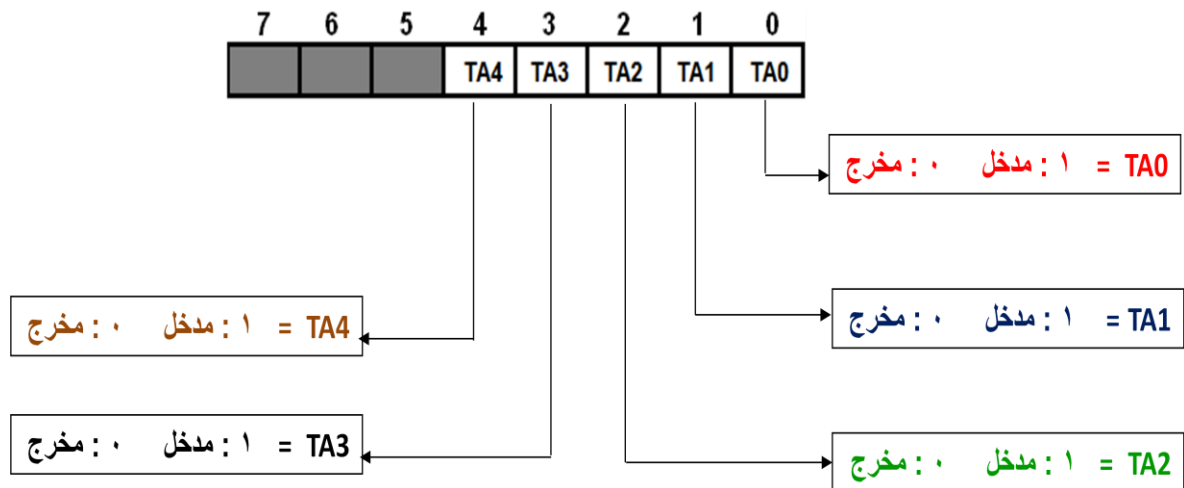
Digit carry : يشير إلى الإحتفاض على الأبيات الأربعة الأولى.

Zéro : = ١ إذا كانت نتيجة العملية تساوي ٠.

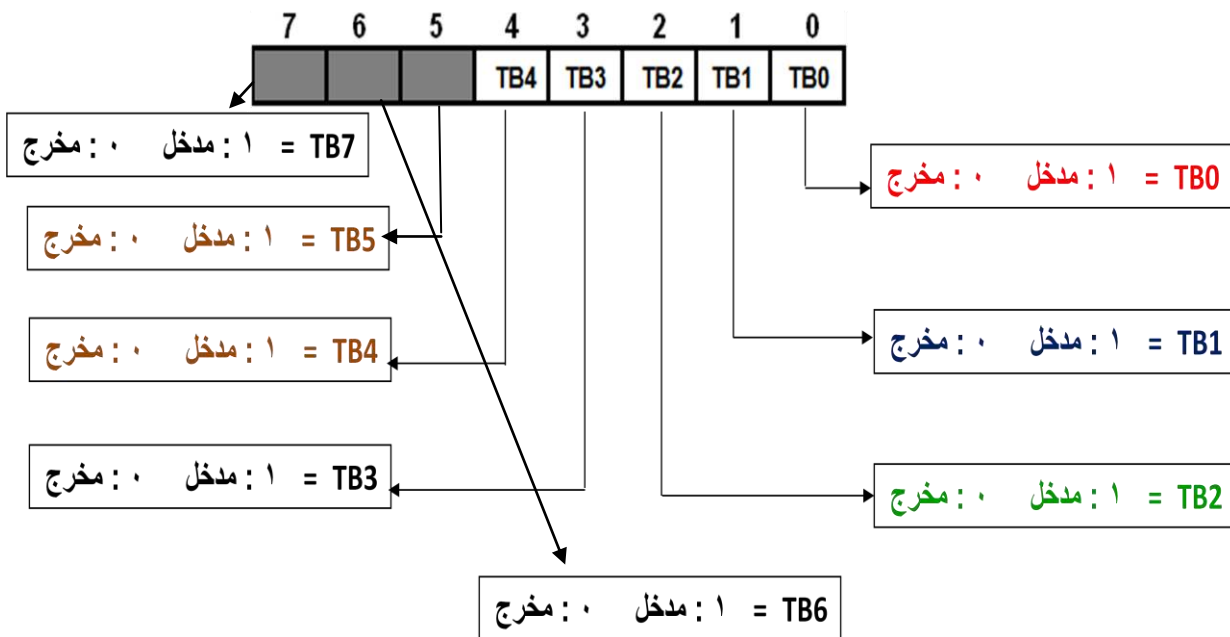
ملاحظة :

- الببت ٤, ٣, ٢, ١, ٠ تعطي دلالات عن نتائج العمليات الحسابية أو حالة الميكرو مراقب ولا يمكننا تغييرها.
- الببت ٥ يسمح لنا تحديد الصفحة التي نريدها ويمكننا تغييرها بواسطة البرنامج.

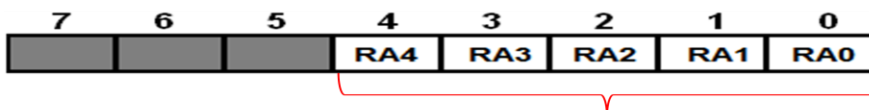
السجل TRISA إتجاه المنفذ A (PORTA)



السجل TRISB إتجاه المنفذ B (PORTB)

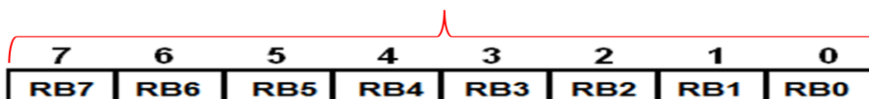


السجل المنفذان PORTA & PORTB



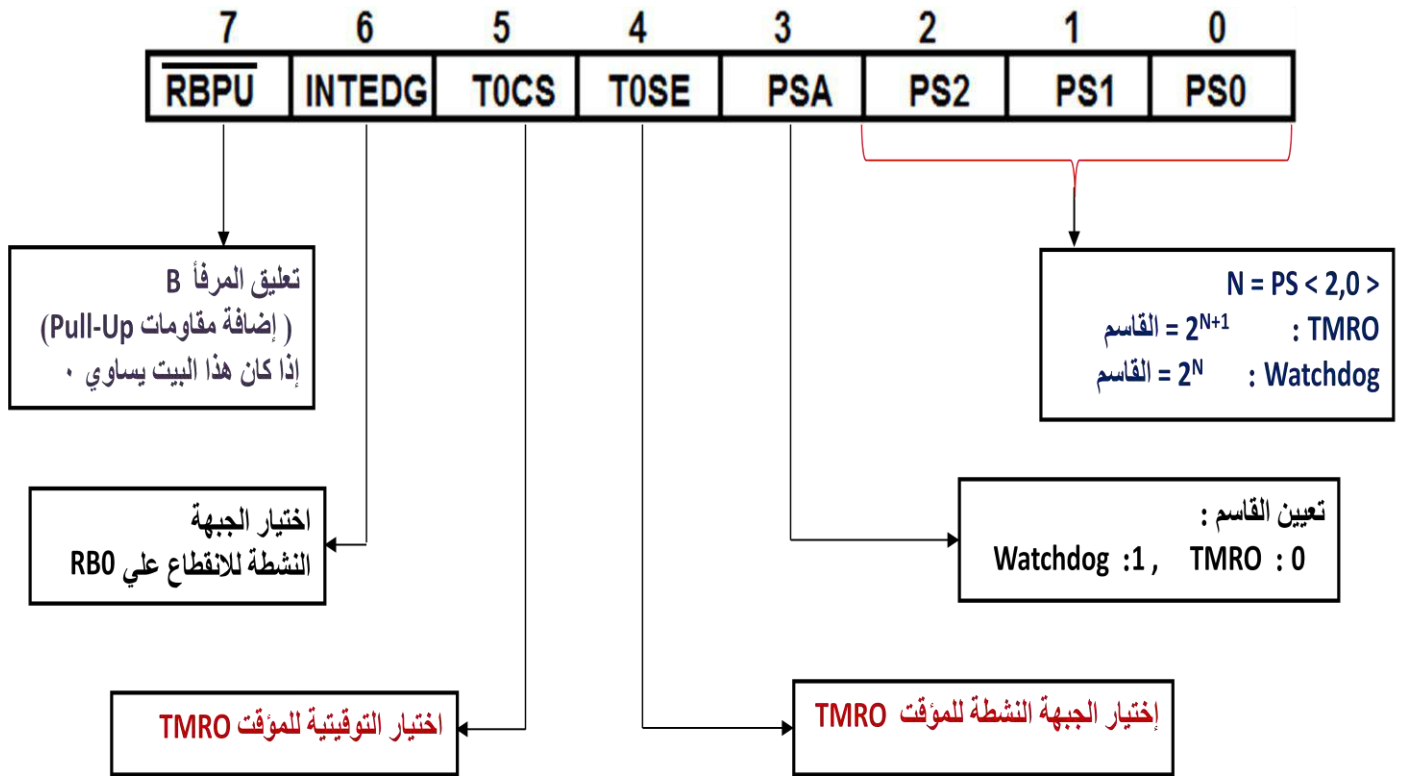
السجل المنفذ PORTA A

المدخل والمخرج

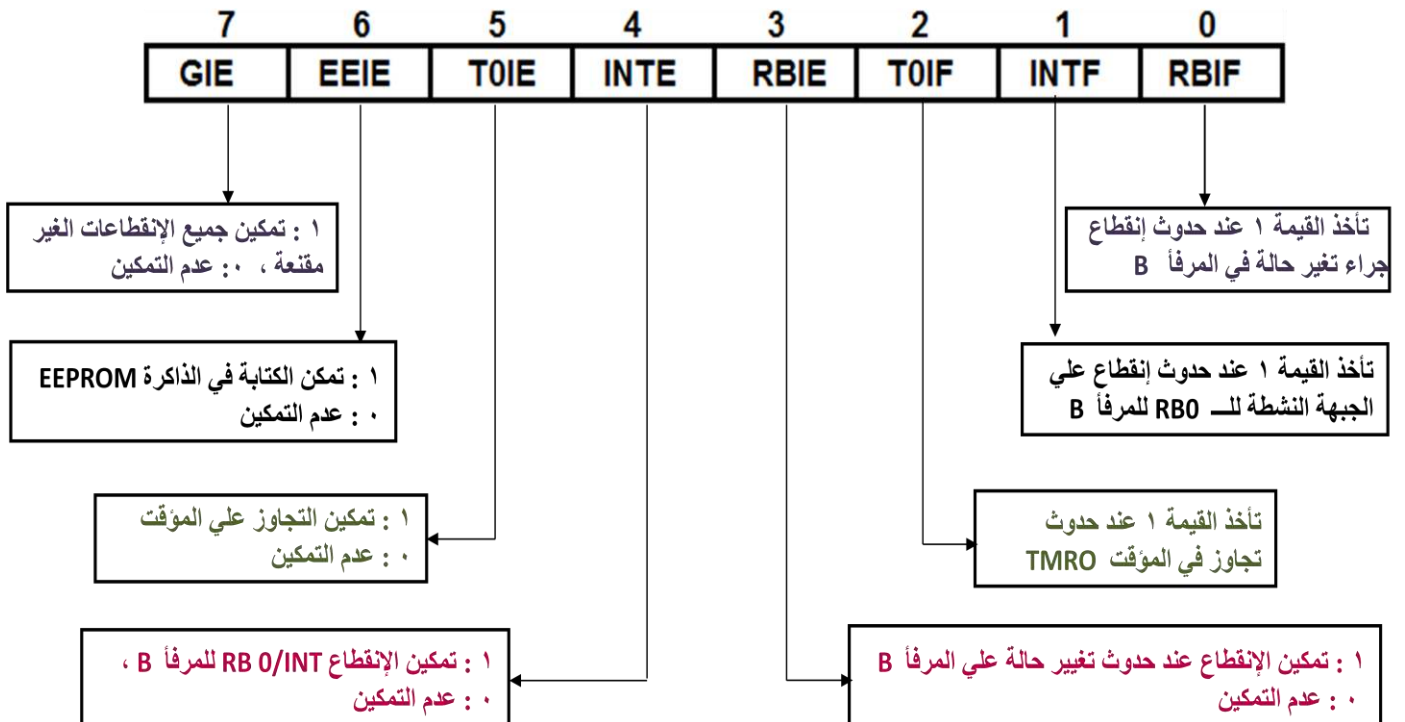


السجل المنفذ PORTB B

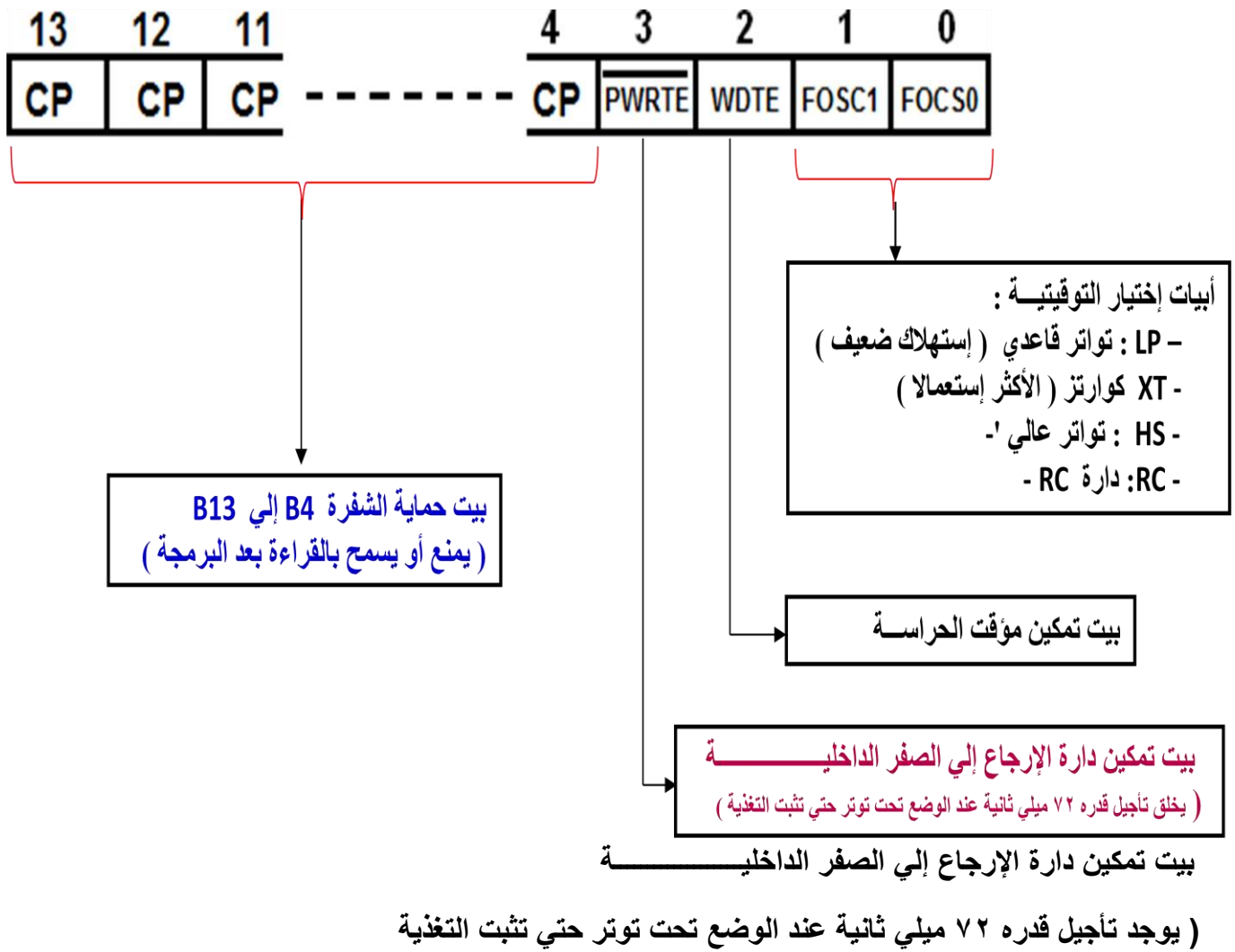
OPTION REG سجل الخيارات



INTCON سجل الإعدادات الداخلية



سجل الإعدادات المادية _CONFIG



ملاحظة:

نقوم بهذه الإعدادات أثناء البرمجة عن طريق سجل يتواجد في العنوان 2007h يتم برمجته بواسطة تعليمة خاصة بالمجمع compilateur

السجل OPTION الخيارات : وهو السجل الذي يتحكم في تشغيل الساعة الداخلية للميكرو المراقب TMR0 حسب السجل السابق يعين مصدر الاشارة المراد استعماله كساعة داخلية TMR0 يصبح TOCS=0 او ساعة خارجية يصبح TOCS=1 الساعة الخارجية تربط في القطب RA4/TOCK14 و البيت PSA يستعمل لتوجيه قاسم التردد الى الساعة TMR0 او الى WDT

بالنسبة للبيت INTEDG يعين الجبهة المستعملة لطلب قطع مطلوب من الخارج على القطب RB0/INT6 $\uparrow 1/\downarrow 0$

نشاط باك ٢٠١٩ :

إعدادات السجل OPTION_REG للميكرو مراقب PIC16F84A :

RBP $\bar{U}$	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
---------------	--------	------	------	-----	-----	-----	-----

ملخص معطيات الصانع

PS2	PS1	PS0	المعامل
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

T0CS : اختيار نوع الساعة (0 : ساعة داخلية ، 1 : ساعة خارجية)

T0SE : اختيار نوع الجبهة (0 : جبهة نازلة ، 1 : جبهة صاعدة)

PSA : اسناد قاسم التردد

(0 : قاسم التردد لـ TMR0 ، 1 : قاسم التردد لـ WDT)

PS2, PS1, PS0 : معامل قاسم التردد حسب الجدول التالي :

س15. عين الهيكل المادي الذي يجسد وظيفة الترابط المنسجم بين التكنولوجيا الكهربائية و الهوائية في هذه الدارة. نريد تغيير دارة المؤجل السابقة بدارة أخرى منجزة بالميكرو مراقب PIC16F84A حيث نستعمل مذبذب (ساعة) خارجي نشط على الجبهة الصاعدة و بقاسم التردد على 128 .

س16. أنقل على ورقة إجابتك ثم أكمل ملء اعدادات السجل OPTION_REG الموالي مستعينا بملخص معطيات الصانع في الملحق (الصفحة 21/7).

اعدادات السجل OPTION_REG

1	0			0			
---	---	--	--	---	--	--	--

ج16. ملء السجل OPTION_REG :

1	0	1	1	0	1	1	0
---	---	---	---	---	---	---	---

مبدأ تشغيل الميكروالمراقب :

دورة الالة هو الزمن اللازم لتنفيذ تعليمة و يمثل اربع دورات لإشارة الساعة

نشاط: استعمال كوارتز 4MHZ احسب زمن تنفيذ التعليمة

$$T_{CYCLE}=4T_H \Rightarrow T_{CYCLE}=4 \times 1/f_H = 4 \times 1/4 \times 10^6 = 1\mu S$$

زمن تنفيذ التعليمة هو ١ ميكرو ثانية .

كلما كان الكوارتز ذو تردد كبير كان تنفيذ العملية أسرع للتعليمة

تشغيل البرنامج :

عند تغذية الميكروالمراقب فإنه يبحث عن التعليمة الأولى المتواجدة عند العنوان ٠٠٠٠ لذاكرة البرنامج و ينفذها يبحث عن التعليمة الموالية المتواجدة عند العنوان ٠٠٠١ و ينفذها و هكذا يتعاقب التشغيل

العنوان تتم ببرنامج بواسطة عداد البرنامج PC CONTEUR PROGRAMME

عداد البرنامج يأخذ القيمة ٠ عند تغذية الميكرو المراقب ثم يزداد محتواه ب ١ عند كل دورة . العنوان ٠٠٠٠ مخصص لشعاع إعادة التعيين RESET أي تهيئة PIC

بواسطة MRCL فإننا نتواجد في هذا العنوان

العنوان ٠٠٠٤ مخصص لشعاع الانقطاع .

list p=16f84

include <p16f84.inc<

__config h'3ff1'

org 0x00

goto debut

debut

org 0x05

call init

bcf STATUS,5

boucle bsf PORTB,0

call tempo

bcf PORTB,0

call tempo

goto boucle

;;

‘initialisation des ports et du

registre OPTION_REG‘

;;

init

bcf STATUS,5 ‘ aller la banque0

clrf PORTB ;effacer PORTB

```

    bsf STATUS,5           ;aller la banque1

    movlw 0x00              ;ts les ports B en sortie

    movwf TRISB

    movlw h'03'            ;initialiser OPTION_REG ==>
PSA=0(prescalaire OK);PS2-PS1-PS0=111(prescalaire a 256);
TOCS=0(horloge interne(

    movwf OPTION_REG

    bcf STATUS,5

    return

                                ;sous programme de temporisation
                                ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

tempo

    clrf TMR0

lab   btfss INTCON,2        ;tester le bit2 (TOIF)si c'est 1 donc
le TMR0 a deborde

    goto lab

    bcf INTCON,2           ;remettre a zero le bit2 du registre
INTCON (indispensable(

    return

    end

```

البرمجة باللغة التجميعية

كل برنامج بلغة المجمع يضم العناصر التالية

العنوان : لتعريف وظيفة البرنامج

قائمة التعليمات (Les Instructions): يمثل البرنامج الفعلي للتطبيق

- التعليقات (Les Commentaires): هي وصف لكل عملية من البرنامج

(١) برمجة المداخل والمخارج

bsf STATUS,5	نضع ١ في البيت ٥ من	للذهاب الى الصفحة الثانية من الذاكرة ;
MOVLW 0x00	نضع ٠٠٠٠٠٠٠٠ في سجل العمل ;	
MOVWF TRISB	نضع محتوى سجل العمل في المنفذ	لبرمجته كمنفذ خروج ;
MOVLW 0x1F	نضع ١١١١١ في سجل العمل ;	
MOVWF TRISA	نضع محتوى سجل العمل في المنفذ	لبرمجته كمنفذ دخول ;
bcf STATUS,5	نضع ٠ في البيت ٥ من	للعودة الى الصفحة الأولى من الذاكرة ;

tempo	البرنامج الفرعي للتأجيل ;
MOVLW 0xFF	نضع ٢٥٥ في سجل العمل W ;
MOVWF retar2	نضع محتوى سجل العمل (٢٥٥) في المتغير retar2 ;
boucle2	
MOVWF retar1	نضع محتوى سجل العمل (٢٥٥) في المتغير retar1 ;
boucle1	
DECFSZ retar1,f	ننقص من المتغير retar1 ١ ونقفز التعليمة الموالية إذا retar1=0 وإلا فالتعليمة الموالية ;
GOTO boucle1	إذهب إلى boucle1 ;
DECFSZ retar2,f	ننقص من المتغير retar2 ١ ونقفز التعليمة الموالية إذا retar2=0 وإلا فالتعليمة الموالية ;
GOTO boucle2	إذهب إلى boucle2 ;
RETURN	إرجع إلى البرنامج الأساسي بعد التعليمة CALL ;

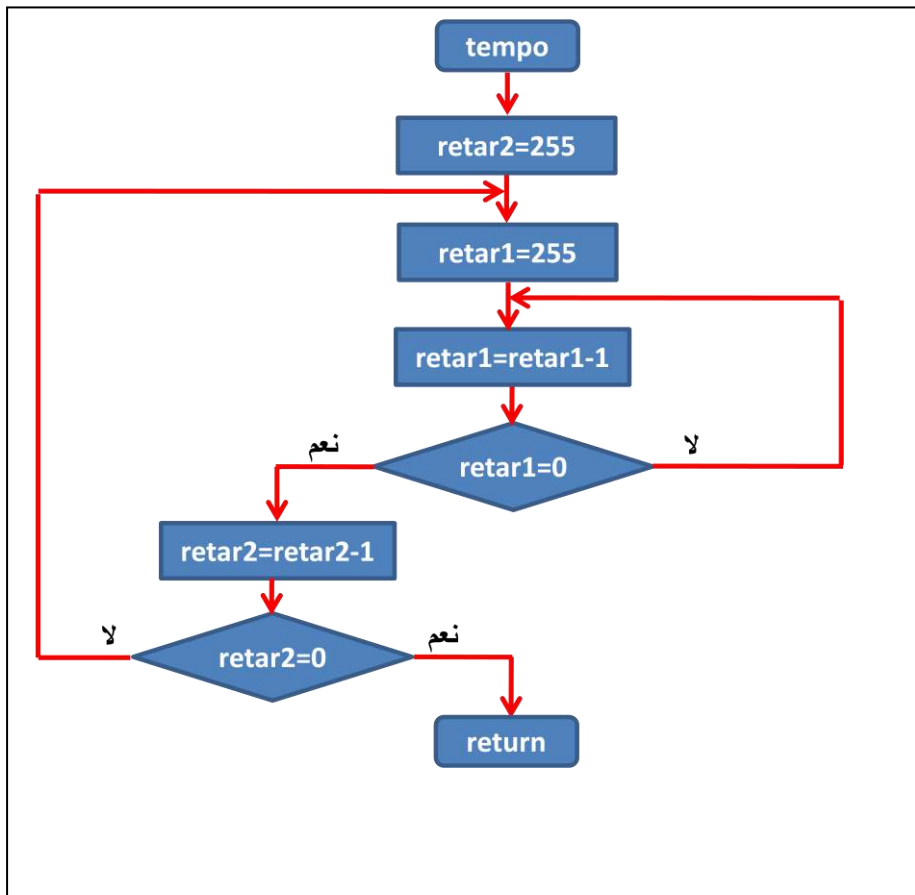
ملاحظة :

- البرنامج الرئيسي يستدعي البرنامج الفرعي بواسطة التعليمة CALL : مثال
CALL tempo

- إن تواتر الساعة الداخلية للميكرو مراقب = ١١/٤ تواتر الساعة الخارجية، أي إذا استعملت ساعة خارجية تنتج 4Mhz، $\Leftarrow \Leftarrow$ الميكرو مراقب ينفذ تعليمة خلال ١ ميكروثانية (أي أن دور الساعة الداخلية يتم خلال ١ ميكروثانية).

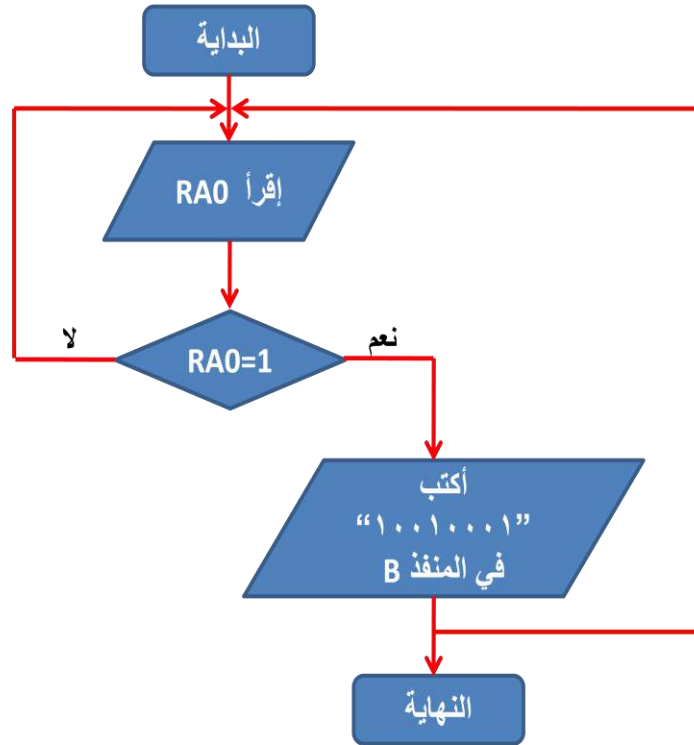
سؤال :

(١) أنجز الخوارزمية البيانية لهذا البرنامج الفرعي للتأجيل



تطبيقات : برنامج رئيسي للتحكم على مصابيح

نريد كتابة برنامج نجعل فيه RA0 كمدخل يتحكم في RB0 ، RB4 ، RB7 كمخارج ونستعمل للمدخل قاطعة وللمخارج ثنائيات ضوئية كدليل للاشتغال.



```

;*****
; هذا البرنامج عبارة عن تطبيق لبرمجة الميكرومراقب بلغة التجميع
;*****
; اسم البرنامج : التحكم في ثلاثة مخارج باستعمال مدخل واحد
; إنجاز : ..... التاريخ : .....
;
;***** التحكم *****
LIST p=16F84A
#include <p16f84a.inc>
; تعريف متغيرات النظام
_CONFIG_CP_OFF & _WDT_OFF & _PWRTE_OFF & _XT_OSC
;
;***** التشغيل في حالة الوضع في الصفر *****
ORG 0x00
GOTO init
;***** البداية *****
init
BSF STATUS,5 ; نضع ١ في الببت ٥ من STATUS للذهاب الى الصفحة الثانية من الذاكرة
MOVLW 0x00 ; نضع ٠٠٠٠٠٠٠٠ في سجل العمل
MOVWF TRISB ; نضع محتوى سجل العمل في المنفذ B لبرمجته كمنفذ خروج
MOVLW 0x1F ; نضع ١١١١١ في سجل العمل
MOVWF TRISA ; نضع محتوى سجل العمل في المنفذ A لبرمجته كمنفذ دخول
BCF STATUS,5 ; نضع ٠ في الببت ٥ من STATUS للعودة الى الصفحة الاولى من الذاكرة
GOTO debut ; اقفز الى البرنامج الرئيسي
;*****

```

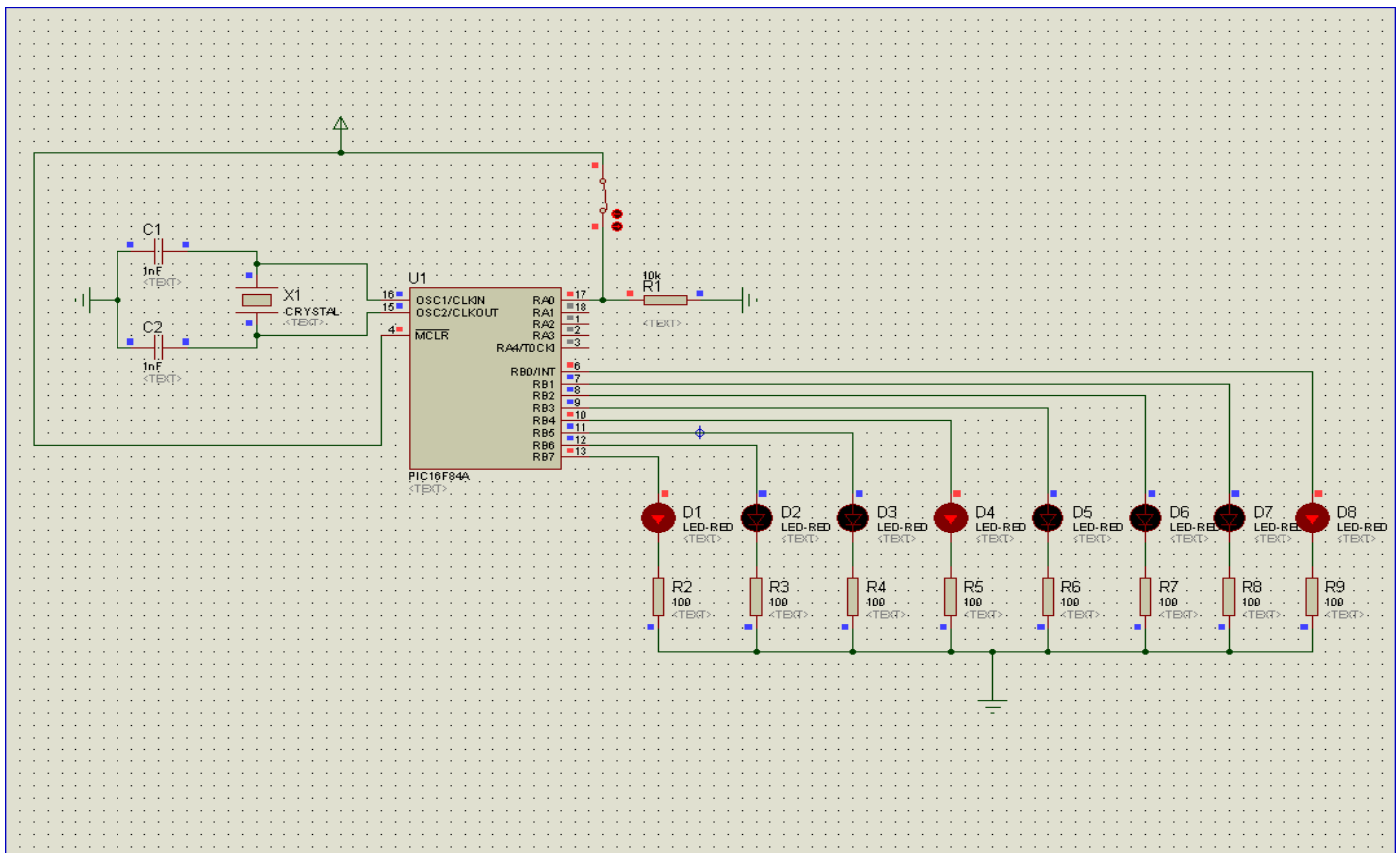
```

; ***** البرنامج الرئيسي *****
debut
    MOVLW B'00000000' ; نضع ٠٠٠٠٠٠ في سجل العمل
    MOVWF PORTB ; نضع محتوى سجل العمل في المنفذ ( كل الثنائيات منطفئة )
start
    BTFSS PORTA, 0 ; القاطعة مغلوقة أكمل البرنامج
    GOTO debut ; و إلا ارجع إلى
    MOVLW B'10010001' ; RB0=1 & RB4=1 & RB7=1
    MOVWF PORTB ; نضع محتوى سجل العمل في المنفذ الخروج
    GOTO start ; ضع كل ذلك في حلقة
END ; نهاية البرنامج

```

٢/٢

تطبيق البرنامج بعد تحويله إلى لغة الآلة HEX باستعمال MPLAB وتقليده
بواسطة MULTISIM



```

*****
;
;      * هذا البرنامج عبارة عن تطبيق لبرمجة الميكرومراقب بلغة التجميع
;
*****
;
;      اسم البرنامج : التحكم في ثلاثة مخارج باستعمال مدخل واحد
;
;      إنجاز : ..... التاريخ : .....
;
;
;      ***** التحكم *****
;
LIST p=16F84A      ; تعريف الميكرومراقب
#include <p16f84a.inc>
_CONFIG_CP_OFF & _WDT_OFF & _PWRTE_OFF & _XT_OSC      ; تعريف متغيرات النظام
;
;      *****
;
retar1 EQU 0x0C      ; ضع متغير في العنوان 0x0C
retar2 EQU 0x0F      ; ضع متغير في العنوان 0x0F
;
;      ***** التشغيل في حالة الوضع في الصفر *****
;
ORG 0x00      ; عنوان الإنطلاق بعد الرجوع إلى الصفر
GOTO init
;
;      ***** البداية *****
;
init
    BSF STATUS,5      ; نضع 1 في الببت 5 من STATUS للذهاب الى الصفحة الثانية من الذاكرة
    MOVLW 0x00      ; نضع 00000000 في سجل العمل W
    MOVWF TRISB      ; نضع محتوى سجل العمل في المنفذ B لبرمجته كمنفذ خروج
    MOVLW 0x1F      ; نضع 11111 في سجل العمل W
    MOVWF TRISA      ; نضع محتوى سجل العمل في المنفذ A لبرمجته كمنفذ دخول
    BCF STATUS,5      ; نضع 0 في الببت 5 من STATUS للعودة الى الصفحة الأولى من الذاكرة
    GOTO debut      ; أقفز الى البرنامج الرئيسي
;
;      *****

```

```

; ***** البرنامج الرئيسي *****
debut
    MOVLW B'00000000' ; نضع ٠٠٠٠٠٠٠٠ في سجل العمل
    MOVWF PORTB ; نضع محتوى سجل العمل في المنفذ ( كل الثنائيات منطفئة )

start
    BTFSS PORTA, 0 ; القاطعة مغلقة أكمل البرنامج
    GOTO debut ; و إلا ارجع إلى
    MOVLW B'10010001' ; RB0=1 & RB4=1 & RB7=1
    MOVWF PORTB ; نضع محتوى سجل العمل في المنفذ الخروج
    CALL tempo ; نفذ البرنامج الفرعي tempo
    MOVLW B'00000000' ; نضع ٠٠٠٠٠٠٠٠ في سجل العمل
    MOVWF PORTB ; نضع محتوى سجل العمل في المنفذ ( كل الثنائيات منطفئة )
    CALL tempo ; نفذ البرنامج الفرعي tempo
    GOTO start ; ضع كل ذلك في حلقة

tempo
    MOVLW 0xFF ; نضع ٢٥٥ في سجل العمل
    MOVWF retar2 ; نضع محتوى سجل العمل (٢٥٥) في المتغير retar2

boucle2
    MOVWF retar1 ; نضع محتوى سجل العمل (٢٥٥) في المتغير retar1

boucle1
    DECFSZ retar1,f
    GOTO boucle1
    DECFSZ retar2,f
    GOTO boucle2
    RETURN
END ; نهاية البرنامج

```