

## TD : Jeux D'instructions

### Exercice 1 : Jeux d'instructions

Quel est le contenu de var et de w après l'exécution de ces instructions ? et précisez l'état des bits modifiés au niveau du registre STATUS : Z et C

Sachant qu'on a :

CBLOCK 0x0C

Var :1

endC

Compléter le tableau ci-après

Instruction	VAR	W	C	Z
movlw 0x05	--	0X05	--	--
movwf var	0X05	0X05	--	--
BCF var,2	0X01	0X05	--	--
incf var,1	0X02	0X05	--	0
decf var, w	0X02	0X01	--	0
movlw d'255'	0X02	0XFF	--	0
addlw 1	0X02	0X00	1	1
movlw d'12'	0X02	0X0C	1	1
movwf var	0X0C	0X0C	1	1
movlw d'25'	0X0C	0X19	1	1
addwf var,f	0X25	0X19	0	0
movlw 0x02	0X25	0X02	0	0
sublw 0x01	0X25	0XFF	0	0
movlw 0x20	0X25	0X20	0	0
movwf var	0X20	0X20	0	0
movlw 0x1F	0X20	0X1F	0	0
subwf var,w	0X20	0X01	1	0
movwf var	0X01	0X01	1	0
movlw B'11001101'	0X01	0XCD	1	0
andlw B'11110000'	0X01	0XC0	1	0
movlw B'11000101'	0X01	0XC5	1	0
xorlw B'00001111'	0X01	0XCA	1	0
BCF status,c	0X01	0XCA	1	0
Movlw B'10010111'	0X01	0X97	1	0
Movwf var	0X97	0X97	1	0
RLF var,f	0X2E	0X97	1	0
Movlw 0xC5	0X2E	0XC5	1	0
Movwf var	0XC5	0XC5	1	0
Swapf var,	0X5C	0XC5		

## Exercice 2 : Analyse bout de code

Commenter le bout de code suivant :

```

Movlw d'253' .....
Movwf portb .....
Movlw 3 .....
Movwf compteur .....
Movlw 0x2 .....
Boucle .....
Addwf var,f .....
Decfsz compteur,f .....
Goto boucle .....
Movf portb,w .....
Compléter le tableau ci-dessous :

```

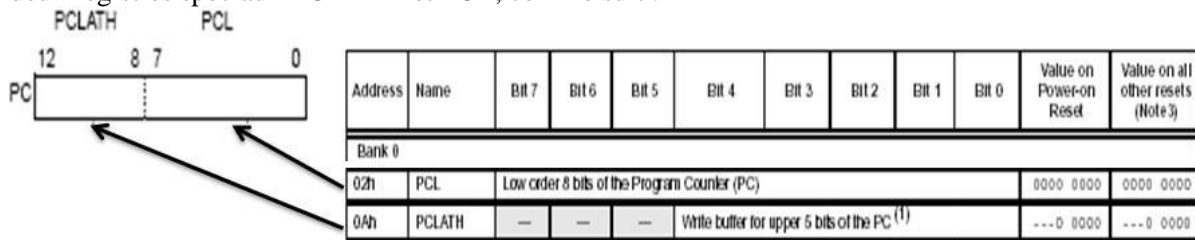
portb	w	compteur	Z	C

## Exercice 3 : PC/PCL

1. Quel est le rôle du compteur ordinal (dit Program Counter : PC)

Le compteur ordinal contient l'adresse de la prochaine instruction à exécuter

2. La taille du compteur ordinal (dit Program Counter PC) est 13 bits, sa valeur est lue dans les deux registres spéciaux PCLATH et PCL, comme suit :



Sachant qu'avec le PIC 16F84, le nombre d'instructions maximal tolérable est de 1024, quelles sont les valeurs possibles du registre PCLATH? Expliquer.

Les valeurs possibles sont quatre : 00, 01, 10 et 11. Pour notre PIC16F84, la longueur d'adresse d'une instruction est de 10 bits : le PCL (PC low) contient les 8 bits de poids faible et le PCLATH contient les 2 bits de poids fort.

3. Commentez et analysez le bout de code suivant et dire c'est quoi le résultat final des registres W, PCL et resultat (resultat est une variable déjà déclarée) :

```

Movlw 0x02           ; W ← 0x02
Call tab             ; appel de la sous-routine « tab »
Movwf resultat       ; resultat ← W ; resultat ← 0x0C
Tab                  ; nom de la sous-routine
Addwf PCL,f          ; PCL ← PCL + W ; PCL ← PCL + 0x02
Retlw 0x0A           ; Return et mettre 0x0A dans W
Retlw 0x0B           ; Return et mettre 0x0B dans W
Retlw 0x0C           ; Return et mettre 0x0C dans W
Retlw 0x0D           ; Return et mettre 0x0D dans W

```

```

;..... PCL ← PCL+ 0x02 ; W ← 0x0C : resultat ← 0x0C
;.....
;.....

```

Rappel : L'instruction « retlw » assure le retour d'une sous-routine (équivalent à l'instruction return), mais en plus, elle affecte à W une valeur spécifiée :

Syntaxe : retlw k ; (w) = k puis return.

#### Exercice 4 : Comparaison

Soient trois variables A , B et resultat, écrire le code assembleur commenté qui permet de les comparer :

- Si A>B, resultat =1
- Si A=B, resultat=2
- Si A<B, resultat=3

1- Déclarer les variables suivantes dans l'ordre (resultat, A, B) chacune de taille 1 octet

```

.. CBLOCK 0x0C
.. Resultat : 1 ; 0x0C
.. A : 1 ; 0x0D
.. B : 1 ; 0x0E
.. EndC

```

Quelle est l'adresse de la variable B au niveau de la RAM ?

```

..... 0x0E

```

2- Code

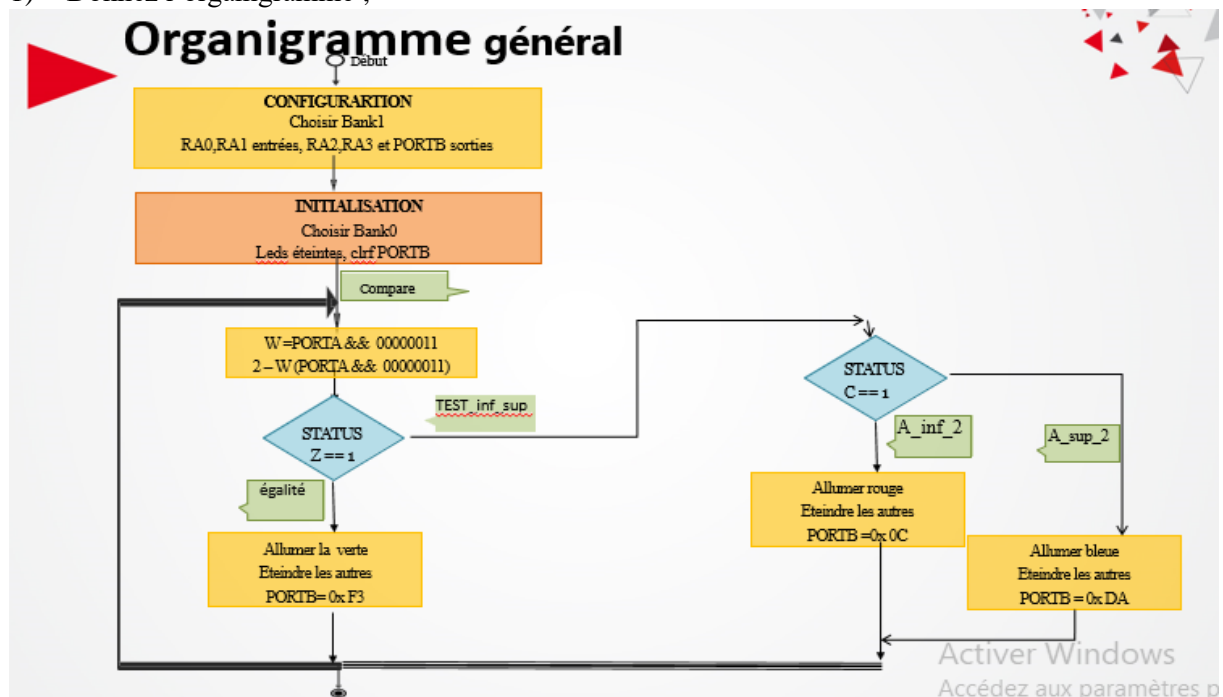
<pre> Compare MOVF B,W ; W ← B           SUBWF A,W ; W ← A- W  BTFSC STATUS,Z   GOTO Egalite BTFSS STATUS,C   GOTO A_inf_B   GOTO A_sup_B </pre>	<pre> Egalite MOVLW 0x02         MOVWF resultat         GOTO Compare A_inf_B MOVLW 0x03         MOVWF resultat         GOTO Compare A_sup_B MOVLW 0x01         MOVWF resultat         GOTO Compare </pre>
--------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## Application : MPLab + ISIS

On se propose d'implémenter sur un PIC16F84 un comparateur ayant 1 entrée de 2 bits. Le fonctionnement est le suivant :

- Si l'entrée (A1,A0)2 est égale à 2 alors la diode verte s'allume et l'afficheur 7 Segment affiche E
- Si l'entrée (A1,A0)2 est inférieure à 2 alors la diode rouge s'allume et l'afficheur 7 Segment affiche I
- Si l'entrée (A1,A0)2 est supérieure à 2 alors la diode bleue s'allume et l'afficheur 7 Segment affiche S

1) Donnez l'organigramme ;



2) Proposer un code en assembleur répondant au fonctionnement décrit

3) Réaliser le câblage correspondant (figure 1) sur ISIS et simuler le fonctionnement

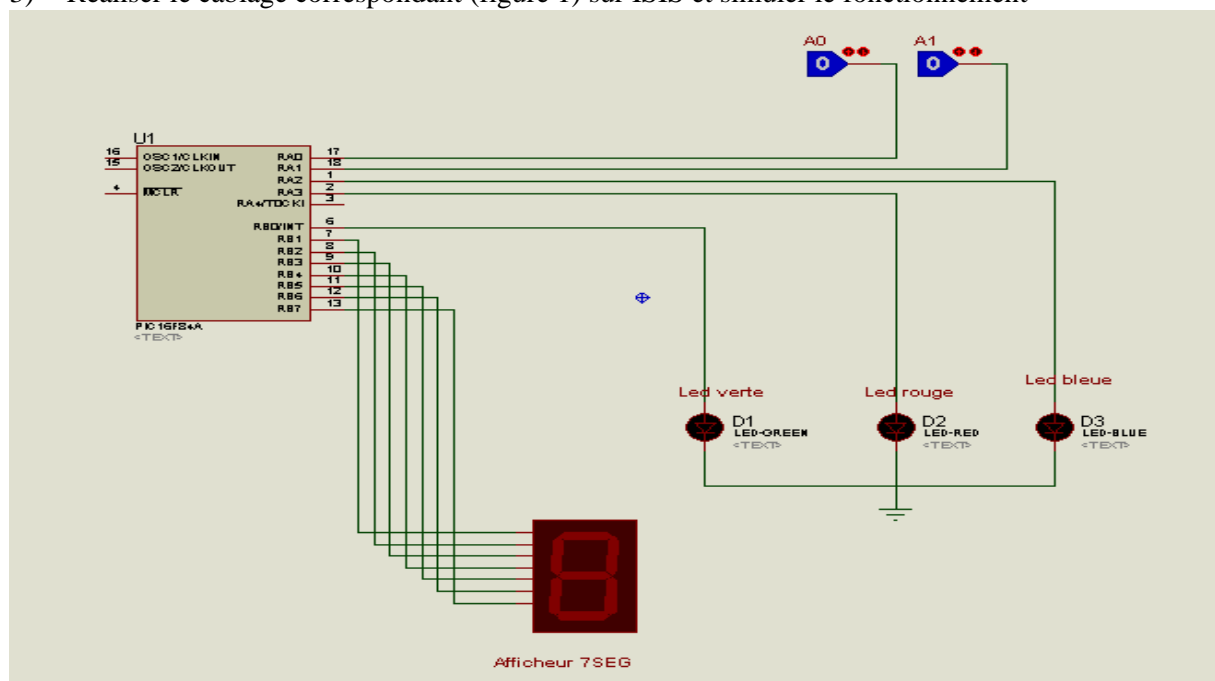


Figure 1