

## TD 1 : PIC16F84A

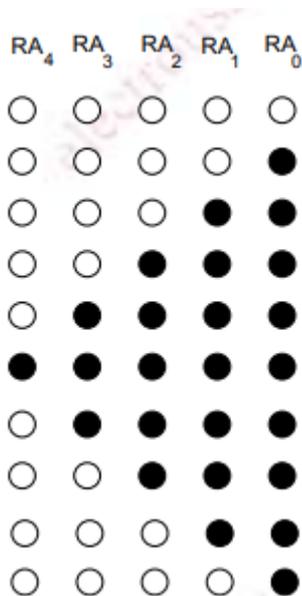
### Exercice 1

Dans un microcontrôleur PIC16F84A,

- quel est l'élément qui sert à indiquer l'adresse de la prochaine instruction à exécuter ?
- où doit se situer le programme qui est exécuté à la mise sous tension ?
- quelle la durée d'un cycle pour exécuter une instruction si le PIC16F84A est piloté par un cristal de 10 MHz ?
- où doivent se situer les données utilisées par le programme ?
- quelle est la longueur d'une instruction en bits ?
- quelle est la longueur d'une donnée dans la mémoire de donnée ?

### Exercice 2

On désire réaliser à l'aide du PIC 16F84A un jeu de lumière qui permet de faire les séquences suivantes :



- Donner le schéma électronique
- comment doit-être configuré le port A ? et quel sera le contenu du registre TRISA ?
- donner la suite des données qu'il faut envoyer sur le port A pour réaliser le jeu de lumière désiré.

### Exercice 3

Exécuter les instructions suivantes et donner les états successifs (la description des instruction est donnée plus bas).

```
movlw 0x05
movwf xdata
movlw 0x01
movwf ydata
movf xdata,0
incf ydata,0
addlw 0x01
sublw 0x05
decf xdata,1
```

### Exercice 4

Exécuter les instructions suivantes et donner les états successifs.

```
movlw 0xD5
movwf X
movlw 0x93
movwf Y
movlw 0xF0
andwf X,1
iorwf Y,1
comf Y,1
```

### Exercice 5

Soit le registre STATUS. On exécute la partie du programme suivante :

```
movlw 0x18
movwf STATUS
movlw 0xFF
addlw 0x01
```

Donner les états successifs du registre W et du registre STATUS.

Quelle est la durée de cette partie du programme, si le PIC16F84A est piloté par un quartz de 8 MHz?

### Exercice 6

Exécuter les instructions suivantes et donner les états successifs.

```

movlw 0x02
  movwf xdata
decsz xdata,1
  goto saut
movlw 0x0A
  movwf xdata
saut decfsz xdata,1
  goto saut
  decf xdata,1

```

### Exercice 7

Soit un sous-programme (appelé : retard) dont l'algorithme est le suivant :

```

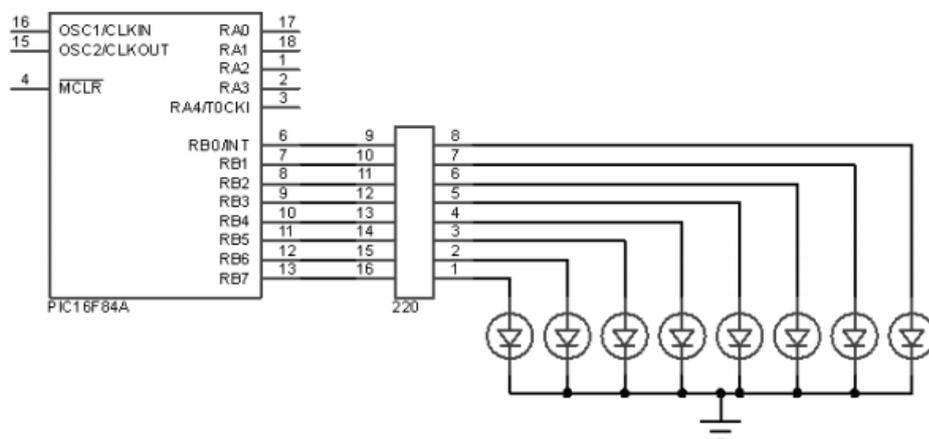
1) COUNT = 10
2) Décrémenter COUNT de 1 (COUNT = COUNT -1)
3) Si : COUNT = 0 ?
Alors : retour de sous-programme
Sinon : aller à 2

```

1. Donner l'organigramme de ce sous-programme
2. Ecrire le sous-programme en assembleur
3. Combien de cycles, dure ce sous-programme ? et quelle est la durée de ce sousprogramme si le PIC utilisé est piloté par un quartz de 8 Mhz ?

### Exercice 8

Soit le montage suivant :



On désire réaliser le jeu de lumière suivant :

RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	Code
1	0	0	0	0	0	0	1	0x81
0	1	0	0	0	0	1	0	0x42
0	0	1	0	0	1	0	0	0x24
0	0	0	1	1	0	0	0	0x18
0	0	1	0	0	1	0	0	0x24
0	1	0	0	0	0	1	0	0x42



**1** → led allumée **0** → led éteinte

1. Ecrire un programme en assembleur
2. Ecrire un programme en C

- **Jeu d'instructions**

W : registre de travail (accumulateur), taille 8 bits

k : valeur littérale, taille 8 bits

Mnémonique , opérande	Description	bit du <u>registre STATUS</u> affecté	nombre de cycles
<b>MOVLW k</b>	k (8 bits) est chargé dans (W)	-	1
<b>ADDLW k</b>	Additionne k (8 bits) et (W) et place le résultat dans (W)	C, DC , Z	1
<b>SUBLW k</b>	Soustrait W de k (8 bits) et place le résultat dans (W)  k - (W) -> (W)	C, DC , Z	1
<b>ANDLW k</b>	Réalise un ET logique entre k (8 bits) et (W), et place le résultat dans (W)	Z	1
<b>IORLW k</b>	Réalise un OU logique (inclusif) entre k (8 bits) et (W), et place le résultat dans (W)	Z	1
<b>XORLW k</b>	Réalise un OU exclusif entre k (8 bits) et (W), et place le résultat dans (W)	Z	1

L : label (étiquette)

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
<b>GOTO L</b>	Branchement à l'adresse L	-	2
<b>CALL L</b>	Appelle un sous-programme (subroutine) situé à l'adresse L	-	2
<b>RETURN</b>	Retour de sous-programme	-	2
<b>RETLW k</b>	Retour de sous-programme, avec chargement de la valeur littérale k (8 bits) dans (W)	-	2
<b>RETFIE</b>	Retour de sous-programme d'interruption	-	2
<b>CLRWDT</b>	Efface le Watchdog	/TO, /PD	1
<b>SLEEP</b>	Place le microcontrôleur en mode sommeil	/TO, /PD	1

f : registre (spécial ou d'usage général)

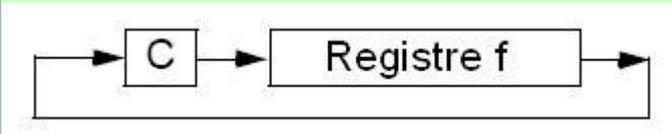
b : position du bit (0 à 7)

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
<b>BCF f , b</b>	Mise à 0 du b ème bit du registre f	-	1
<b>BSF f , b</b>	Mise à 1 du b ème bit du registre f	-	1
<b>BTFSC f , b</b>	Si le b ème bit du registre f est égal à 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
<b>BTFSS f , b</b>	Si le b ème bit du registre f est égal à 1, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2

f : registre (spécial ou d'usage général)

d : registre de destination (on peut choisir entre le *registre de travail W* et le *registre f*).

Mnémonique , opérande	Description	bit du registre STATUS affecté	nombre de cycles
<b>MOVWF f</b>	(W) est chargé dans (f)	-	1
<b>MOVF f , d</b>	(f) (8 bits) est chargé dans (destination)	Z	1
<b>ADDWF f , d</b>	Additionne le contenu du registre f (8 bits) et (W), et place le résultat dans (destination)	C, DC , Z	1
<b>SUBWF f , d</b>	Soustrait (W) de (f) (8 bits) et place le résultat dans (destination).  (f) - (W) ->(destination)	C, DC , Z	1
<b>ANDWF f , d</b>	Réalise un ET logique entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1
<b>IORWF f , d</b>	Réalise un OU logique (inclusif) entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1
<b>XORWF f , d</b>	Réalise un OU exclusif entre (f) (8 bits) et (W), et place le résultat dans (destination)	Z	1
<b>COMF f , d</b>	Réalise le complément logique de (f) (8 bits), et place le résultat dans (destination)	Z	1
<b>DECF f , d</b>	Décrémente (f) et place le résultat dans (destination).  (f) - 1 -> (destination)	Z	1

<b>DECFSZ f, d</b>	Décrémente (f) et place le résultat dans (destination). Si le résultat est 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
<b>INCF f, d</b>	Incrémente (f) et place le résultat dans (destination). (f) + 1 -> (destination)	Z	1
<b>INCFSZ f, d</b>	Incrémente (f) et place le résultat dans (destination). Si le résultat est 0, alors l'instruction suivante est ignorée, et une instruction NOP est exécutée à la place (soit 2 cycles)	-	1 ou 2
<b>CLRF f</b>	Efface le contenu du registre (f). Remarque : le bit Z est donc mis à 1.	Z	1
<b>CLRW</b>	Efface le contenu de l'accumulateur (W). Remarque : le bit Z est donc mis à 1.	Z	1
<b>RLF f, d</b>	Réalise une rotation circulaire à gauche :  Le résultat est placé dans (destination).	C	1
<b>RRF f, d</b>	Réalise une rotation circulaire à droite :  Le résultat est placé dans (destination).	C	1
<b>SWAPF f, d</b>	Les 4 bits de poids forts et les 4 bits de poids faibles de (f) sont échangés. Le résultat est placé dans (destination).	-	1
<b>NOP</b>	Cette instruction ne fait rien (durée 1 cycle).	-	1