

## 1. Introduction

Le  $\mu C$  16F84 possède un jeu de 35 instructions. Chaque instruction est codée sur un mot de 14 bits qui contient le code opération (OC) ainsi que l'opérande. A part les instructions de saut, toutes les instructions sont exécutées en un cycle d'horloge. Sachant que l'horloge fournie au  $\mu C$  est prédivisée par 4, si on utilise par exemple un quartz de 4 MHz, on obtient donc 1000000 cycles/seconde, cela nous donne une puissance de l'ordre de 1 MIPS (1 Million d'Instructions Par Seconde). Avec un quartz de 20 MHz, on obtient une vitesse de traitement d'autant plus rapide.

## 2. Organisation des instructions

### 21. Instructions « orientées octets » (adressage direct) : Figure 1

Ce sont des instructions qui manipulent les données sous forme d'octets. Elles sont codées de la manière suivante :

- ☑ 6 bits pour l'instruction : c'est logique, car comme il y a 35 instructions, il faut 6 bits pour pouvoir les coder toutes.
- ☑ 1 bit (d) pour indiquer si le résultat obtenu doit être conservé dans le registre de travail (accumulateur) W de l'unité de calcul (W pour Work) ou sauvé dans un registre f (f pour file).
- ☑ Reste 7 bits pour encoder l'adresse de l'opérande, mais 7 bits ne donnent pas accès à la mémoire RAM totale, c'est ainsi qu'on utilise le bit RPO du registre STATUS pour compléter le 8<sup>ème</sup> bit.

🔔 **Remarque** : (W, f ?d) signifie que le résultat est stocké soit dans W si d=0, soit dans f si d=1.

Figure 1

Instructions opérant sur un registre (adressage direct)		indicateurs	Cycles
ADDWF f,d	$W + f \rightarrow (W, f ?d)$	C, DC, Z	1
ANDWF f,d	$W \text{ And } f \rightarrow (W, f ?d)$	Z	1
CLRF f	Effacer f	Z	1
CLRWF	Effacer W	Z	1
CLRWDT	Effacer watchdog timer	TO, PD	1
COMF f,d	Complémenter f $\rightarrow (W, f ?d)$	Z	1
DECF f,d	Décrémenter f $\rightarrow (W, f ?d)$	Z	1
DECFSZ f,d	Décrémenter f $\rightarrow (W, f ?d)$ , sauter si 0		1(2)
INCF f,d	Incrémenter f $\rightarrow (W, f ?d)$	Z	1
INCFSZ f,d	Incrémenter f $\rightarrow (W, f ?d)$ , sauter si 0		1(2)
IORWF f,d	$W \text{ Or } f \rightarrow (W, f ?d)$	Z	1
MOVF f,d	$f \rightarrow (W, f ?d)$	Z	1
MOVWF f	$W \rightarrow f$		1
RLF f,d	Rotation à gauche de f à travers C $\rightarrow (W, f ?d)$	C	1
RRF f,d	Rotation à droite de f à travers C $\rightarrow (W, f ?d)$	C	1
SUBWF f,d	$f - W \rightarrow (W, f ?d)$	C, DC, Z	1
SWAPF f,d	Permuter les deux quartets de f $\rightarrow (W, f ?d)$		1
XORWF f,d	$W \text{ Xor } f \rightarrow (W, f ?d)$	Z	1

## 22. Instructions « orientées bits » : Figure 2

Ce sont des instructions destinées à manipuler directement les bits d'un registre d'une case mémoire. Elles sont codées de la manière suivante :

- 4 bits pour l'instruction.
- 3 bits pour indiquer le numéro du bit à manipuler (de 0 à 7).
- 7 bits pour indiquer l'opérande.

Figure 2

Instructions opérant sur un bit d'un registre		indicateurs	Cycles
BCF f,b	Mettre à 0 le bit b du registre f		1
BSF f,b	Mettre à 1 le bit b du registre f		1
BTFSC f,b	Tester le bit b de f, sauter une instruction si 0		1(2)
BTFSS f,b	Tester le bit b de f, sauter une instruction si 1		1(2)

## 23. Instructions opérant sur une donnée (adressage immédiat) : Figure 3

Ce sont des instructions qui manipulent des données qui sont codées dans l'instruction directement. Elles sont codées de la manière suivante :

- L'instruction est codée sur 6 bits.
- Elle est suivie d'une valeur immédiate codée sur 8 bits (donc de 0 à 255).

Figure 3

Instructions opérant sur une donnée (adressage immédiat)		indicateurs	Cycles
ADDLW k	$W + k \rightarrow W$	C, DC, Z	1
ANDLW k	$W \text{ And } k \rightarrow W$	Z	1
IORLW k	$W \text{ Or } k \rightarrow W$	Z	1
MOVLW k	$k \rightarrow W$		1
SUBLW k	$k - W \rightarrow W$	C, DC, Z	1
XORLW k	$W \text{ Xor } k \rightarrow W$	Z	1

## 24. Instructions de saut et appel de procédures : Figure 4

Ce sont des instructions qui provoquent une rupture dans la séquence de déroulement du programme. Elles sont codées de la manière suivante :

- Les instructions sont codées sur 3 bits.
- La destination est codée sur 11 bits.

Figure 4

Instructions générales		indicateurs	Cycles
CALL L	Branchement à un sous programme de label L		2
GOTO L	Branchement à la ligne de label L		2
NOP	Pas d'opération		1
RETURN	Retourner d'un sous programme		2
RETFIE	Retourner d'une procédure d'interruption		2
RETLW k	Retourner d'un sous programme avec k dans W		2
SLEEP	Mettre le $\mu$ C en mode de veille	TO, PD	1