

Chapitre 4: Microcontrôleurs

Objectifs du chapitre

- ✓ Découvrir le fonctionnement d'un microcontrôleur et avoir l'habilité de le programmer
- ✓ Différencier entre un μ -processeur et un μ -contrôleur

Contenu du chapitre

IV.1- Introduction

Les microcontrôleurs sont aujourd'hui implantés dans la plupart des applications grand public ou professionnelles, il en existe plusieurs familles.

La société Américaine Microchip Technologie a mis au point dans les années 90 un microcontrôleur CMOS : le PIC (**P**eripheral **I**nterface **C**ontroller). Ce composant encore très utilisé à l'heure actuelle, est un compromis entre simplicité d'emploi, rapidité et prix de revient.

- Les PIC existent dans plusieurs versions:
- les UVROM qui sont effaçable par une source de rayonnements ultraviolets
- les OTPROM programmable une seule fois
- les E²PROM et flash EPROM qui sont effaçables électriquement.

IV.2- Classification des PIC_s de Microchip

Actuellement les modèles Microchip, sont classés en 3 grandes familles, comportant chacune plusieurs références. Ces familles sont :

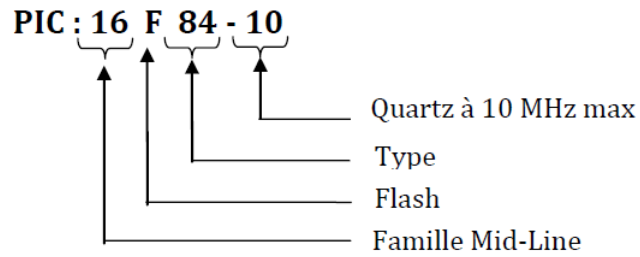
- Base-line : les instructions sont codées sur 12 bits.
- mid-line : les instructions sont codées sur 14 bits.
- High-End : les instructions sont codées sur 16 bits.

IV.3- Identification des PIC_s

Un PIC est généralement identifié par une référence de la forme suivante : **xx(L)XXyy-zz**

xx	:	famille du composant, actuellement « 12, 14, 16, 17 et 18 ».	
L	:	tolérance plus importante de la plage de tension	
XX	:	type de mémoire programme	
		C	: EPROM ou EEPROM
		CR	: PROM
		F	: Flash
yy	:	Identificateur.	
zz	:	vitesse maximale du quartz de pilotage.	

Exemple :



IV.4- Tableau comparatif

	Mem prog en octets	RAM en octets	EEPROM en octets	Fmax en MHz	E / S	Boîtier
12C508	512x12	25	-	4	6	8 broches
16C72A	2048x14	128	-	20	22	28 broches
16F84	1024x14	68	64	20	13	18 broches
16F628	2028x14	224	128	20	16	18 broches
16F876	8192x14	368	256	20	22	28 broches
16F877	8192x14	368	256	20	33	40 broches

Tableau IV.1 : Caractéristiques de quelques pics.

IV.5- Architecture interne

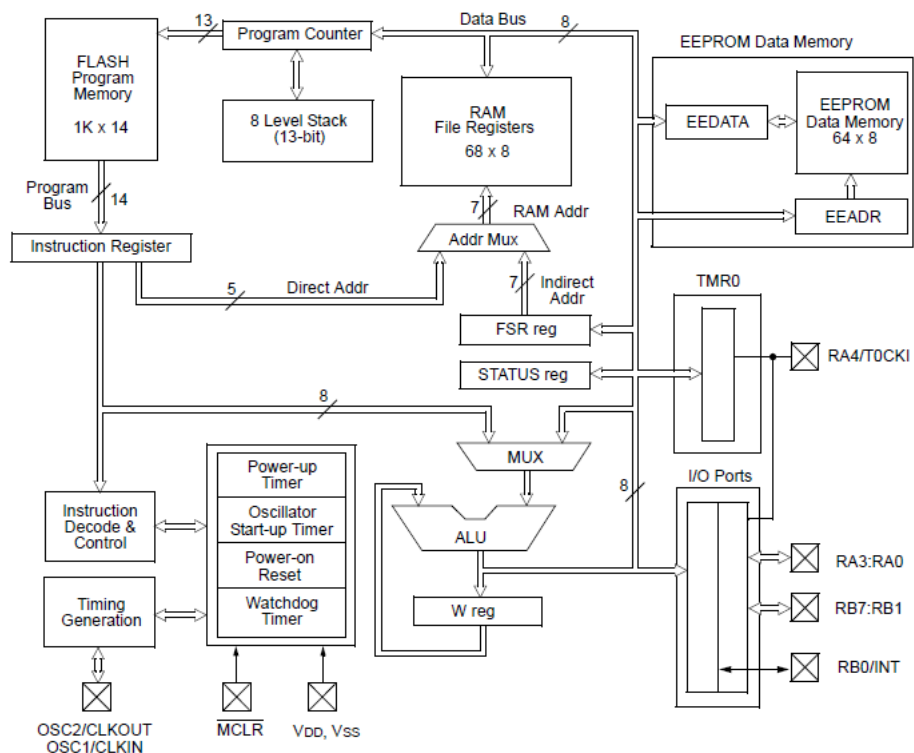


Figure IV.1 : Architecture interne du 16F84.

La **Figure IV.1** présente l'architecture du PIC 16F84, commune à la majorité des microcontrôleurs PIC. Le PIC 16F84 est un microcontrôleur 8 bits d'architecture de type RISC (*Reduced Instructions Set Computer*), signifie « calculateur à jeu réduit d'instructions ». Les instructions sont en effet au nombre de 35, soit une centaine de moins que le microcontrôleur 68HC11 de Motorola par exemple.

Parmi les principaux constituants, on remarque:

- ❖ la mémoire de type Flash ROM pour le programme;
- ❖ la mémoire de type RAM pour les registres;
- ❖ la mémoire de type EEPROM pour les données à sauvegarder;
- ❖ des registres particuliers: W, FSR et d'état;
- ❖ l'Unité Arithmétique et Logique (ALU) ;
- ❖ les ports d'entrées / sorties;
- ❖ la pile à 8 niveaux.

IV.5.1- La mémoire de programme

Comme son nom l'indique, c'est dans cette mémoire qu'est mémorisé le programme à exécuter. De type Flash ROM, elle est organisée en 1024 mots de 14 bits, ce qui permet de coder sur un seul mot l'instruction et l'opérande, à la différence des microcontrôleurs plus classiques à 8 bits où l'instruction et l'opérande sont écrits sur plusieurs adresses consécutives.

Chaque mot de cette mémoire programme est repéré par son adresse allant de 0 à 1023 (0 à 3FF en hexadécimal). Le Compteur Programme (PC) contient l'adresse de l'instruction à exécuter.

IV.5.2- La pile

Ce terme désigne les emplacements mémoire où sont empilées les adresses de retour lors des appels à des sous-programmes. La pile ne comportant que 8 niveaux, il ne faut pas imbriquer plus de 8 sous-programmes.

IV.5.3- Les registres

IV.5.3.1- Organisation

Adresses	Registres	Registres	Adresses
00	INDF	INDF	80
01	TMR0	OPTION	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85

06	PORTB	TRISB	86
07			87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C à 4F	68 mémoires disponibles	Accès aux mêmes mémoires que page 0	8C à CF
Page 0		Page 1	

Tableau IV.2 : Organisation des registres.

La mémoire RAM de 68 octets est utilisée pour stocker des données temporaires. Baptisé du nom de registre à usage général par Microchip, chacune de ces mémoires est adressable directement. À ce bloc de 68 registres à usage général, le fabricant a associé 11 registres internes et organisé les adresses selon le **Tableau IV.2**. On remarque que les registres sont définis sur deux pages dont nous expliquerons l'accès au cours du paragraphe sur le registre STATUS.

Certains figurent dans les deux pages, ce qui peut faciliter la programmation.

IV.5.3.2- Description des registres internes

*** Registre INDF**

Le registre INDF d'adresse 00 est utilisé pour l'adressage indirect. Pour expliquer ce mode d'adressage, il est plus facile d'expliquer le mode d'adressage direct.

Dans le mode direct, l'instruction comprend l'adresse effective de l'opérande. Exemple: ADDWF 8, 0 signifie additionner le contenu du registre W avec le contenu du registre d'adresse 08 et mettre le résultat dans W. Il est possible de faire la même opération en utilisant l'adressage indirect et le registre INDF par la ligne d'instruction ADDWF INDF, 0 en ayant mis préalablement 8 dans le registre FSR (*File Selection Register*) d'adresse 04.

*** Registre TMRO**

Le registre TMRO d'adresse 01 est lui aussi particulier puisqu'il constitue un compteur programmable incrémenté continuellement à une cadence constante indépendante de l'exécution du reste du programme. Après avoir atteint la valeur 255, le compteur reprend à 0. La cadence de l'incrémentation est proportionnelle à la fréquence de l'horloge appliquée au microcontrôleur ou à la fréquence du signal appliqué sur la broche 3 (RA4/TOCKI). Un pré diviseur permet de programmer cette proportion. Le fonctionnement de ce pré diviseur est géré dans le registre OPTION décrit dans le **Tableau IV. 3**.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
RBP	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0

Tableau IV. 3 : Registre OPTION.

À l'issue d'un *Reset*, tous les bits du registre OPTION sont à 1.

RBP : (PORTB Pull Up enable bit) : ce bit à 0 permet la validation des résistances de tirage.

Prédiviseur non actif (PreScaler Assignment bit, $PSA = 1$)

La cadence de comptage est égale à la fréquence $F_{osc}/4$ si le bit TOCS (TMRO *Clock Source*) est à 0. Cette cadence est celle du signal appliqué à TOCKI si le bit TOCS est à 1. De plus, le bit TOSE (TMRO *Source Edge*) permet de choisir si l'incrémentaion a lieu sur un front descendant ou sur un front montant (descendant si $TOSE = 0$ et montant si $TOSE = 1$).

Exemple:

Si nous utilisons un quartz à 4 MHz. $F_{osc}/4$ a donc une valeur de 1 MHz. Si $TOCS = 0$, la cadence de comptage est de 1 million d'incrémentations par seconde.

Pré diviseur actif ($PSA = 0$)

Le pré diviseur (tableau 4) permet de diviser la fréquence d'incrémentaion $F_{osc}/4$ ou TOCKI selon l'effet voulu.

PS2	PS1	PS0	Diviseur
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

Tableau IV.4 : Utilisation du pré diviseur avec le timer 0.

*** Registre PCL et PCLATH**

Le registre PCL d'adresse 02 contient les 8 bits de poids faible du PC qui est codé sur 13 bits. Lors d'opérations d'écriture dans le registre PCL, le nouveau compteur programme est calculé à partir des 8 bits du registre PCL et des 5 bits du registre PCLATH d'adresse 0A.

* Registres PORTA et PORTB

Les registres PORTA d'adresse 05 et PORTB d'adresse 06 sont les registres des ports d'entrées/sorties. Chaque broche de chaque port peut être utilisée en entrée ou en sortie: le sens de fonctionnement des ports est défini par le contenu des registres TRISA et TRISB : un bit à 0 programme la broche correspondante en sortie, un bit à 1 la programme en entrée. Quand une broche est en sortie, son état est déterminé par le bit correspondant du registre du port. Quand une broche est en entrée, c'est son niveau qui détermine la valeur du bit du registre correspondant. Lors d'un *Reset*, les bits des registres TRISA et TRISB étant mis à 1, toutes les broches des ports A et B sont initialement des entrées.

* Registre INTCON

Le registre INTCON d'adresse 0B, présenté **Tableau IV.5**, est utilisé pour le contrôle et la gestion des interruptions.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF

Tableau IV.5 : Registre INTCON.

Le bit RBIF, comme *RB Interrupt Flag*, est mis à 1 lors d'un changement d'état sur une des lignes RB4 à RB7.

- Le bit INTF (*INTerrupt Flag*), est mis à 1 si une interruption est générée sur RB0/INT.
- Le bit TOIF (*Timer 0 Interrupt Flag*), est mis à 1 en cas de débordement du *timer 0*.
- Le bit RBIE (*RB Interulpt Enable*), mis à 1, autorise les interruptions sur RB4 à RB7.
- Le bit INTE (*INTerrupt Enable*), mis à 1, autorise les interruptions sur RB0/INT. L'interruption a lieu sur le front montant de l'impulsion si le bit INTEDG, comme *INTerrupt Edge*, du registre OPTION est à 1 ; elle a lieu sur le front descendant si ce bit est à 0.
- Le bit TOIE (*Timer 0 Intmupt Enable*), mis à 1 autorise les interruptions dues au débordement du *timer 0*.
- Le bit EEIE (*EEPROM write completed Interrupt Enable*), autorise les interruptions de fin d'écriture dans l'EEPROM.
- Le bit GIE (*Global Interrupt Enable*), mis à 1 autorise toutes les interruptions non masquées par leur bit individuel.

À la lecture de la signification de ces bits, on notera que le 16F84 possède 4 sources potentielles d'interruptions:

- un changement d'état sur les broches RB4 à RB7;
- une source externe via la broche RB0/INT ;
- le débordement du *timer 0* ;
- la fin de l'écriture dans l'EEPROM de données.

Chaque indicateur de changement d'état RBIF, INTF, RTIF doit être remis à 0 par le logiciel dans le programme de traitement de l'interruption.

Lors d'un *Reset*, tous les bits du registre INTCON sauf RBIF sont mis à 0. RBIF garde son état précédent.

Un seul vecteur d'interruption étant disponible à l'adresse 0004, le programme d'interruption doit déterminer éventuellement quelle est la source d'interruption.

* Registres EEDATA, EEADR, EECON1 et EECON2

Les registres EEDATA, EEADR, EECON1 et EECON2 sont utilisés pour la lecture et l'écriture dans l'EEPROM. L'utilisation de quatre registres pour ces opérations est en fait assez simple.

Le registre EEDATA d'adresse 08 contient, selon l'opération, la donnée lue ou la donnée à écrire. Le registre EEADR d'adresse 09 contient son adresse. Le registre EECON1 d'adresse 88 définit le mode de fonctionnement de l'EEPROM. Le registre EECON2 d'adresse 89 n'est pas physiquement implanté et ne sert qu'à la sécurisation lors de la phase d'écriture. L'EEPROM ne contenant que 64 octets d'adresse 00 à 3F, les 2 bits de poids fort de EEADR sont donc toujours à 0.

Le registre EECON1 d'adresse 88 est décrit tableau 6.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
			EEIF	WRERR	WREN	WR	RD

Tableau IV.6 : Registre EECON1.

Le bit RD (*Read Data*), doit être à 1 pour lire une donnée. Il est automatiquement remis à 0 à l'issue.

Le bit WR (*WRite data*), doit être à 1 pour écrire la donnée. Il est automatiquement remis à 0 à l'issue.

- Le bit WREN (*WRite Enable*), doit être à 1 pour autoriser l'écriture.
- Le bit WRERR (*Write ERROR*), est mis à 1 en cas d'erreur d'écriture.
- Le bit EEIF, comme *EEPROM Interrupt Flag*, est mis à 1 quand l'écriture d'une donnée est terminée. Ce bit doit être remis à 0 par le programme.

* Registre STATUS

Le registre STATUS, ou registre d'état d'adresse 03, que nous allons détailler plus particulièrement, est décrit **Tableau IV.7**.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IRP	RP1	RP0	TO	PD	Z	DC	C

Tableau IV.7 : Registre STATUS.

Le bit C (*Carry*), est mis à 1 lors d'opérations avec retenue. Ce bit est également utilisé par les instructions de rotation.

- Le bit DC (*Digit Carry*), est mis à 1 quand une retenue apparaît après le bit 3.
- Le bit Z (*Zero*), est mis à 1 quand un résultat arithmétique ou logique est nul.
- Le bit PD (*Power Down*), est mis à 1 lors de la mise sous tension ou lors de l'exécution de CLRWDT. Ce bit est mis à 0 par l'instruction SLEEP.
- Le bit TO (*Time Out*), est mis à 1 lors de la mise sous tension ou lors de l'exécution des instructions CLRWDT et SLEEP. Ce bit est mis à 0 si le *timer* chien de garde déborde.
- Le bit RP0 sert à sélectionner la page des registres:
 - RP0 à 0 : page 0 active, il est alors possible d'accéder aux registres d'adresse 00 à 4F ;
 - RP0 à 1 : page 1 active, il est alors possible d'accéder aux registres d'adresse 80 à CF.

Les registres qui figurent dans les deux pages sont accessibles quel que soit l'état du bit RP0 Les bits RP1 et IRP ne sont pas utilisés par le 16F84 et il est conseillé de les laisser à 0.

*** Registre de travail W (*Working register*) et Unité Arithmétique et Logique (ALU)**

Le registre de travail est l'équivalent des accumulateurs des microcontrôleurs classiques. C'est à travers l'Unité Arithmétique et Logique que s'effectuent les opérations arithmétiques et booléennes entre les données d'un registre et le registre de travail W. Le registre de travail n'est pas adressable comme les autres registres. Dans les opérations à deux opérandes, un opérande est dans le registre W, l'autre est une constante ou est contenu dans un autre registre. Dans les opérations à un opérande, l'opérande est soit dans W, soit dans un autre registre. Les valeurs des bits C, DC et Z du registre d'état sont affectées selon les opérations effectuées par l'ALU.

IV.5.4- Le mot de configuration

Une partie des fonctionnalités du PIC est contrôlée par les bits du mot de configuration d'adresse 2007, décrit **Tableau VI.8 :**

Bit 13 à 4	Bit 3	Bit 2	Bit 1	Bit 0
CP	PWRTE	WDTE	FOSC1	FOSC0

Tableau VI.8 : Mot de configuration.

Les bits CP (*Code Protection*), mis à 1 protègent le PIC en lecture.

Le bit PWRTE (*Power-up Timer Enable*), mis à 1 active le *Power-up timer* et génère un délai de 72 ms après un *Reset*, permettant éventuellement à la tension V_{DD} d'atteindre un niveau acceptable.

Le bit WDTE (*WatchDog Timer Enable*), mis à 1 active le chien de garde.

Les bits FOSC1 et FOSC0 sont positionnés selon le type d'horloge externe choisie:

- 11 : oscillateur RC ;
- 10 : oscillateur quartz haute vitesse (4 à 20 MHz);
- 01 : oscillateur quartz standard (centaines de KHz à 4 MHz);
- 00 : oscillateur quartz basse consommation (centaines de KHz).

IV.6- Brochage du PIC 16F84

Ce microcontrôleur se présente sous la forme d'un boîtier DIL à 18 broches comme schématisé

Figure VI.2

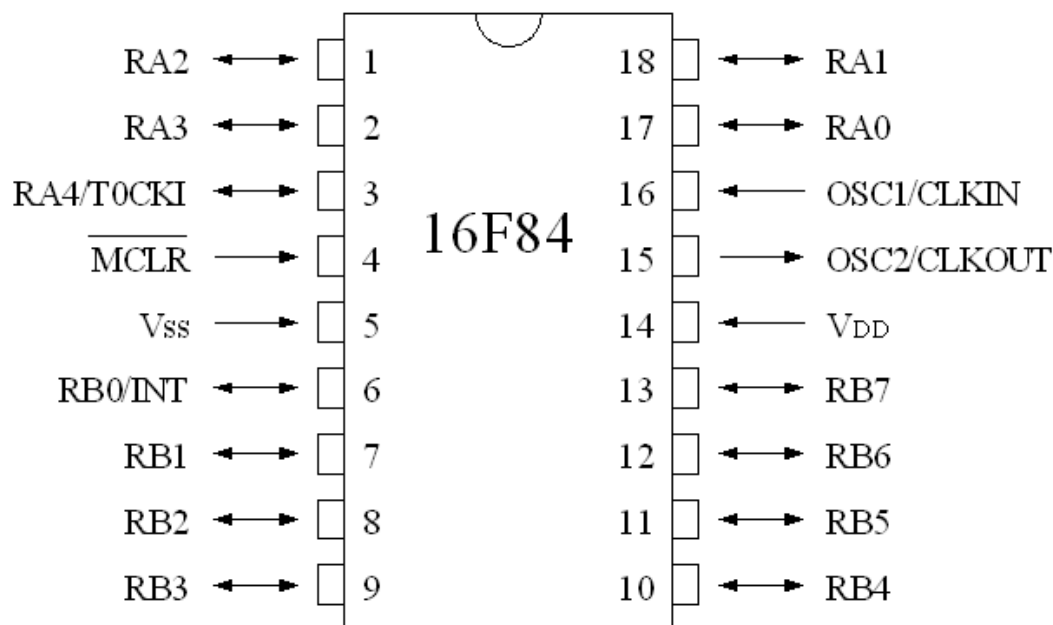


Figure VI.2 : Brochage du PIC 16F84.

- RA0 à RA4/T0CKI sont les pattes d'entrées/sorties du portA. RA4/T0CKI est commune avec l'entrée d'horloge externe du *timer 0*.
- RB0/INT à RB7 sont les pattes d'entrées/sorties du port B. RB0/INT est commune avec l'entrée d'interruption externe.

- Individuellement, chaque broche des ports A et B ne peut débiter plus de 20 mA ou absorber plus de 25 mA. Le total des intensités débitées par le port A ne peut dépasser 50 mA et par le port B, 100 mA. Le total des intensités absorbées par le port A ne peut dépasser 80 mA et par le port B, 150 mA.

OSC1/CLOCKIN et OSC2/CLOCKOUT sont les pattes d'horloges. Plusieurs types d'horloges peuvent être utilisés: externe, à quartz ou à circuit RC.

La fréquence de l'horloge interne du PIC est obtenue en divisant par 4 la fréquence de l'horloge externe. Pour un quartz à 4 MHz, la fréquence interne est donc de 1 MHz et la durée d'un cycle est de 1 μ s.

- MCLR/VPP est la patte de *Reset* et d'entrée de la tension de programmation. Les circuits PIC intégrant en interne le circuit de *Reset* automatique à la mise sous tension, cette broche doit être reliée à la V_{DD} en utilisation normale.
- V_{SS} et V_{DD} sont les pattes d'alimentation. V_{DD} doit être compris entre 2 et 6 V en utilisation. Lors de la programmation, V_{DD} doit être comprise entre 4,5 V et 5,5 V et V_{PP} comprise entre 12 V et 14 V.

IV.7- Jeu d'instructions

Le jeu d'instructions du PIC 16Fxx est présenté dans les tableaux, 9 (opérations sur des octets), 10 (opérations sur des bits) et 11 (opérations de contrôle et opérations sur les littéraux).

Mnémonique	Opérande	Description	Cycles	Bits affectés
ADDWF	f,d	Add W and F	1	C, DC, Z
ANDWF	f,d	AND W with f	1	Z
CLRF	f	Clear f	1	Z
CLRW		Clear W	1	Z
COMF	f,d	Complement f	1	Z
DECF	f,d	Decrement f	1	Z
DECFSZ	f,d	Decrement f, Skip if 0	1(2)	
INCF	f,d	Increment f	1	Z
INCFSZ	f,d	Increment f, Skip if 0	1(2)	
IORWF	f,d	Inclusive OR W with f	1	Z
MOVF	f,d	Move f	1	Z
MOVWF	f	Move W to f	1	
NOP		No Operation	1	
RLF	f,d	Rotate Left through Carry	1	C
RRF	f,d	Rotate Right through Carry	1	C

SUBWF	f,d	Subtract W from f	1	C, DC, Z
SWAPF	f,d	Swap nibbles in f	1	
XORWF	f,d	Exclusive OR W with f	1	Z

Tableau IV.9 : Opérations sur des octets.

Mnémonique	Opérande	Description	Cycles	Bits affectés
BCF	f,b	Bit Clear f	1	
BSF	f,b	Bit Set f	1	
BTFSC	f,b	Bit Test f, Skip if Clear	1(2)	
BTFSS	f,b	Bit Test f, Skip if Set	1(2)	

Tableau IV.10 : Opérations des bits.

Mnémonique	Opérande	Description	Cycles	Bits affectés
ADDLW	K	Add literal and W	1	C, DC, Z
ANDLW	K	AND literal and W	1	Z
CALL	K	Call subroutine	1	
CLRWDT		Clear Watchdog Timer	1	TOn, PDn
GOTO	K	Goto adress	1(2)	
IORLW	K	Inclusive OR literal with W	1(2)	
MOVLW	K	Move literal to W	1	Z
RETFIE		Return from interrupt	1	Z
RETLW	K	Return with literal in W	1	Z
RETURN		Return from subroutine	1	Z
SLEEP		Go into standby mode	1(2)	TOn, PDn
SUBLW	K	Subtract W from literal	1(2)	
XORLW	K	Exclusive OR literal with W	1	Z

Tableau IV.11 : Opérations de contrôle et opérations sur les littéraux.

Si le PC est modifié ou si un test est vrai, l'instruction dure 2 cycles.

Notations :

f	Registre (File Register).
d	Destination de l'opération (W si d=0 ou f si d=1).
K	Valeur immédiate (Literal value).

À titre d'exemple, les listings 1 et 2 montrent deux programmes types de lecture et d'écriture en assembleur.

Liste des mnémoniques

addlw	k	;ajoute une constante k à W. Résultat dans W
addwf	f,d	;ajoute W à f. Résultat dans W si d=0 ou dans f si d=1
andlw	k	;effectue un ET entre une la constante k et W. Résultat dans W
andwf	f,d	;effectue un ET entre W et f. Résultat dans W si d=0 ou dans f si d=1
bcf	f,b	;fait passer le bit b de f à 0
bsf	f,b	;fait passer le bit de f à 1
btfsc	f,b	;teste le bit b de f. Incrémente PC si b=0
btfss	f,b	;teste le bit b de f. Incrémente PC si b=1
call	k	;empile PC et affecte PC de l'adresse d'un sous programme
clrf	f	;place zero dans f
clrw		;place zero dans W et fait passer STATUS.Z à 1
clrwtd		;initialise le timer du chien de garde
comf	f,d	;complemente f à 1. Résultat dans W si d=0 ou dans f si d=1
decf	f,d	;décrémte f. Résultat dans W si d=0 ou dans f si d=1
decfsz	f,d	;décrémte f. Si f=0, incrémente PC. Résultat dans W ou dans f
goto	k	;charge une adresse dans PC
incf	f,d	;incrémte F
incfsz	f,d	;incrémte f. Résultat dans W si d=0 ou dans f si d=1
iorlw	k	;effectue un OU entre une constante et W
iorwf	f,d	;effectue un OU entre W et f. Résultat dans W si d=0 ou dans f si d=1
movf	f,d	;si d=0, écrit f dans W , sinon réécrit f dans f
rlf	f,d	;effectue une rotation de bits à gauche à travers Carry
rrf	f,d	;effectue une rotation de bits à droite à travers Carry
sleep		;fait passer le PIC en mode veille
sublw	k	;sustrait W de d'une constante k (W := k-W)
subwf	f,d	;soustrait W de f. Résultat dans W si d=0 ou dans f si d=1
swapf	f,d	;permutte les deux quartets de f. Résultat dans W si d=0 ou dans f si d=1
xorlw	k	;effectue un OU exclusif entre une constante et W. Resultat dans W
xorwf	f,d	;OU exclusif entre W et f. Résultat dans W si d=0 ou dans f si d=1
const		;est obsolète : ne plus l'utiliser