

Présentation du

microcontrôleur PIC 16F84

Avertissement

Cette présentation du PIC 16F84 ne dispense pas de consulter la documentation technique

Sommaire	1
Architecture	2 , 3
Structure interne	4
Broches	5
Horloge	5
Fichier des registres	6 , 7
Ports	8 , 9
Timers	10, 11
Reset	12
Interruptions	13
Compteur de programme et pile	13
Mémoire EEPROM de données	14
Configuration et programmation	15
Résumé des instructions	16

Bibliographie

Documentations constructeur et notes d'application sur www.microchip.com

Ouvrages en français

Microcontrôleurs PIC à structure RISC de C.F. URBAIN Edition PUBLITRONIC

Les microcontrôleurs PIC de C. TAVERNIER Editions DUNOD

Certaines illustrations de ce document proviennent de la documentation constructeur de Microchip.

Architecture

Type Von Neumann (6811 entre autres)

La ROM contient les codes des instructions et les constantes.

Exemple : **Ldaa #10** ; charger la constante 10 dans l'accumulateur A

Occupe deux adresses en ROM pour y loger les octets \$96 et 10

Nécessite plusieurs cycles d'horloge :

- adresse AD sur le bus d'adresse, code \$96 sur le bus de données
- adresse AD+1 sur le bus d'adresse, valeur 10 sur le bus de données

Type Harvard sur PIC16F84

Un seul cycle d'horloge par instruction :

1. L'opérande est intégré à l'instruction.

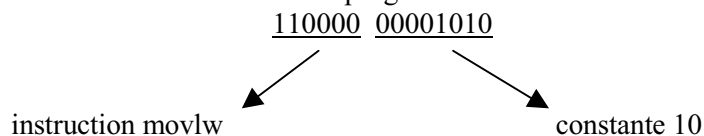
Exemple : **movlw 10** ; charger le registre de travail avec la constante 10.

[to move : déplacer , l (literal) : constante, w (work) registre de travail]

Cette instruction comprend :

- Le code de l'instruction sur 6 bits (33 instructions)
- La constante 10 sur 8 bits

On trouvera donc en mémoire programme un mot de 14 bits



Terminologie :

- Jeu d'instructions complexe (Complex Instructions Set Computer: **CISC**)
- Jeu d'instructions réduit (Reduced Instructions Set Computer: **RISC**)

2. Les mémoires Programme et Données et les bus correspondants sont séparés

Ceci permet, au même instant, de

- Exécuter l'instruction correspondant à l'adresse courante
- Extraire l'instruction suivante

Matériel : ROM Programme → registre tampon → décodeur d'instruction

Ces 3 composants constituent un pipeline. Voir figure 1 page suivante.

Séquencement : L'horloge est quadriphasée. Voir figure 2

Exception : Sauts et branchements : deux cycles d'horloge sont nécessaires. figure 3.

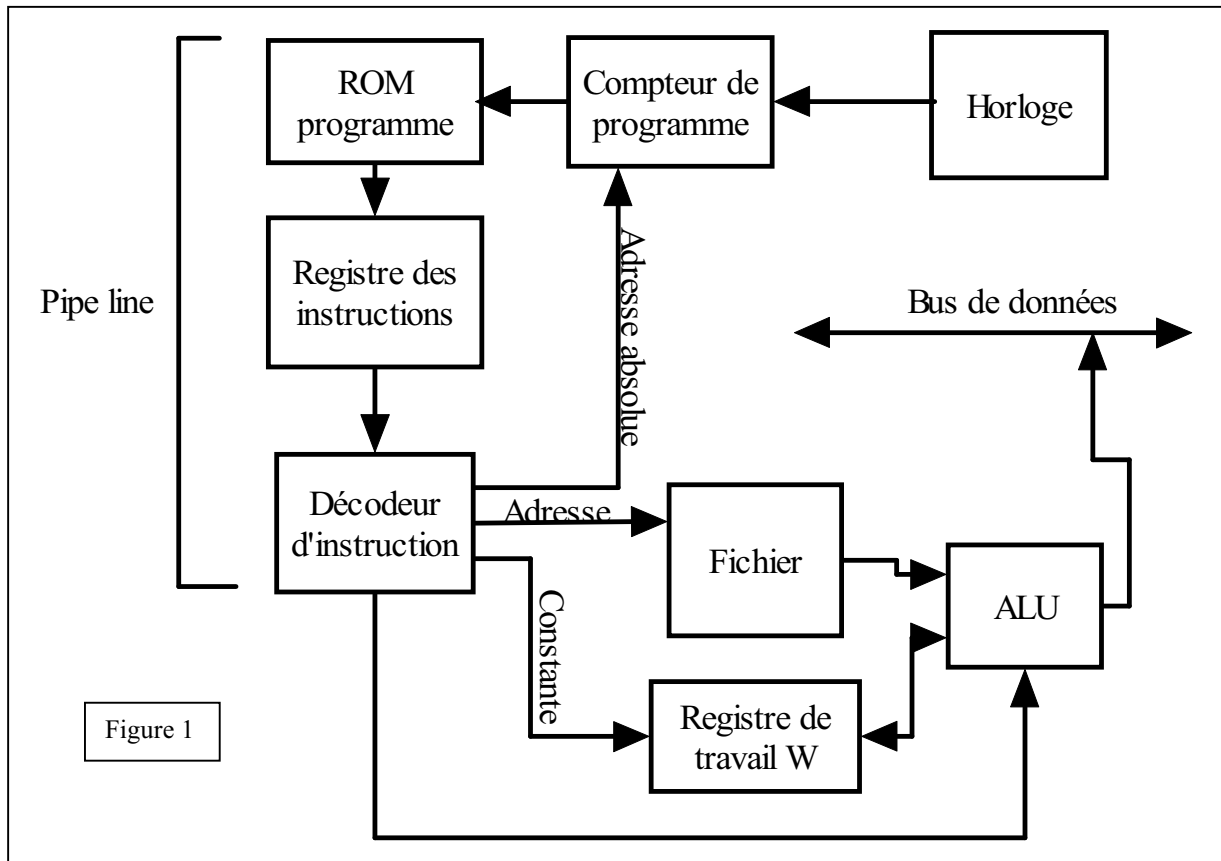


Figure 1

CLOCK/INSTRUCTION CYCLE

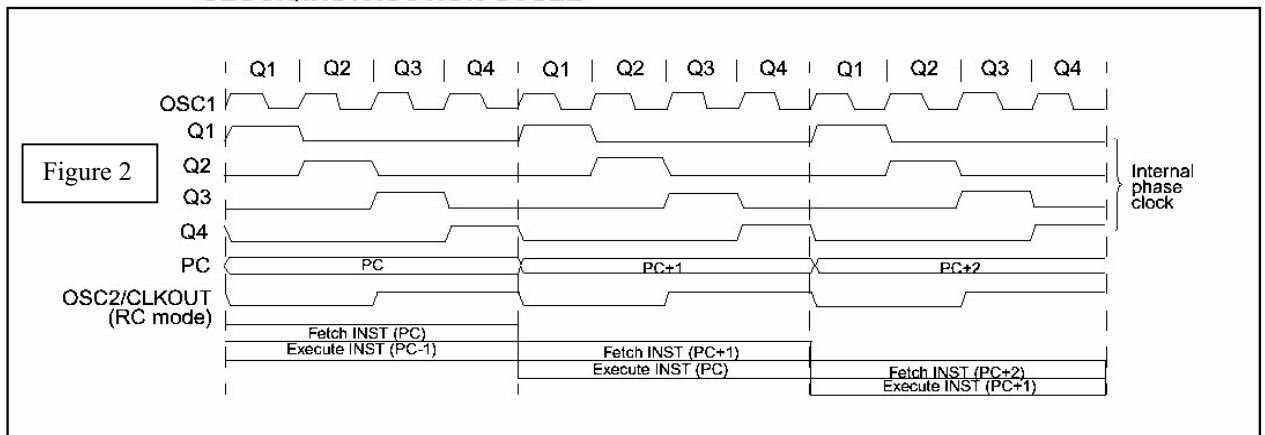


Figure 2

INSTRUCTION PIPELINE FLOW

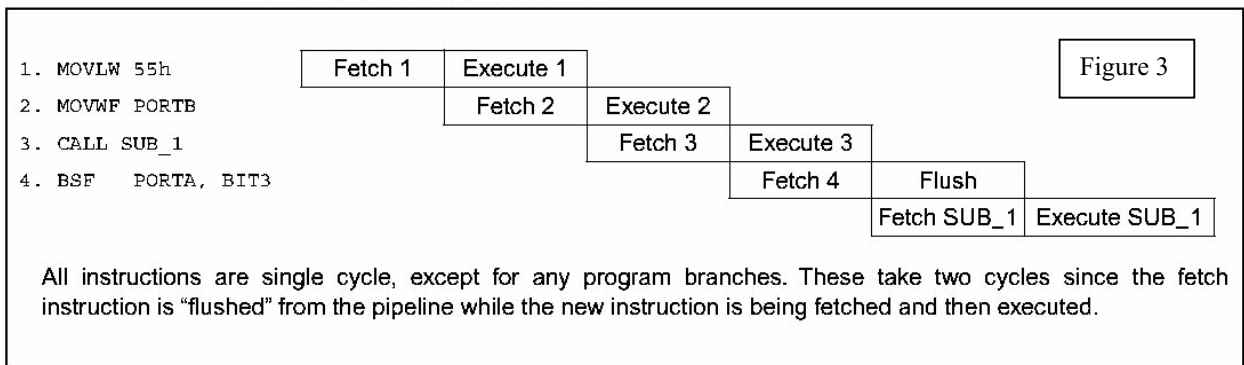
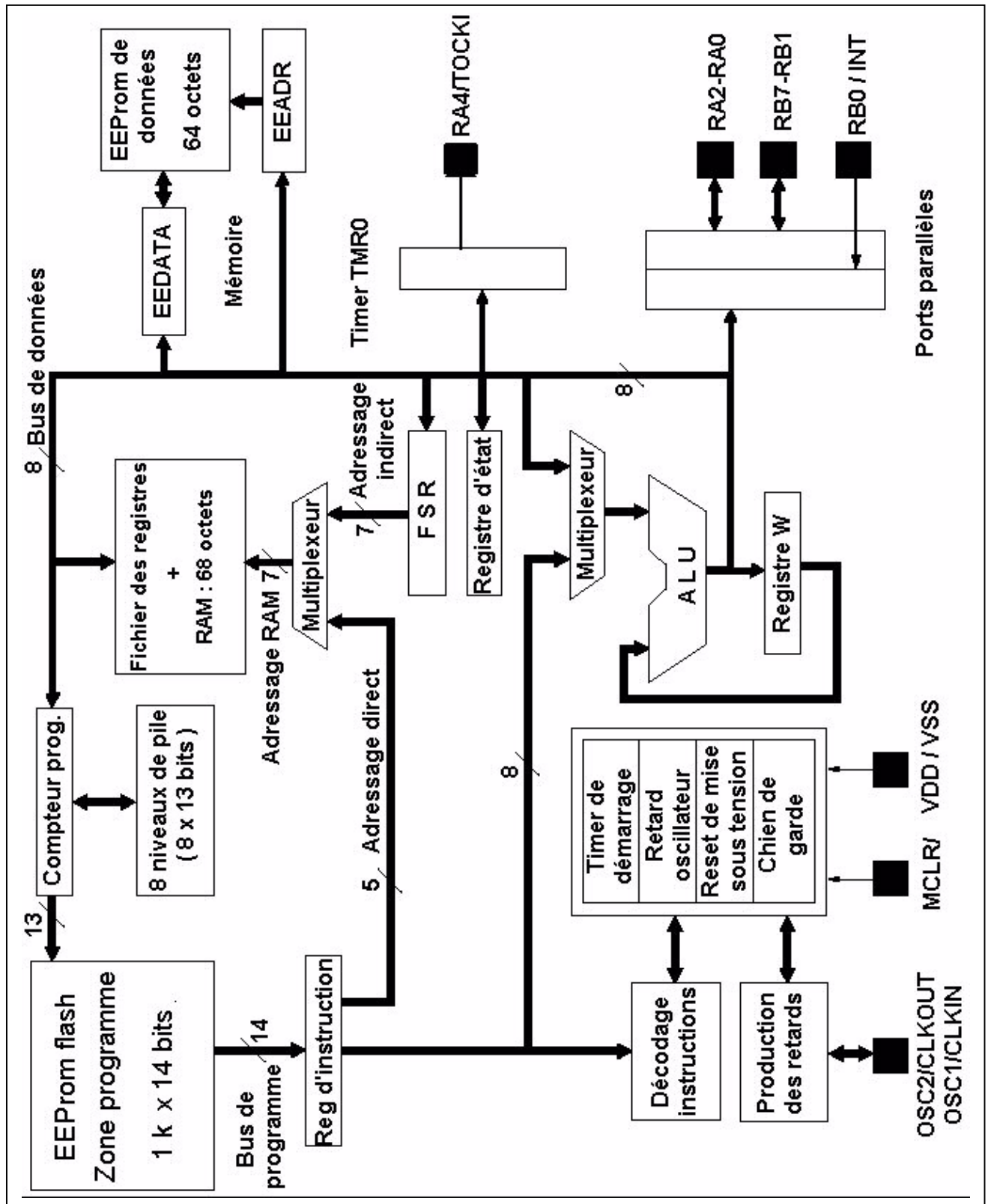


Figure 3

All instructions are single cycle, except for any program branches. These take two cycles since the fetch instruction is "flushed" from the pipeline while the new instruction is being fetched and then executed.



Structure interne



Broches du PIC 16F84

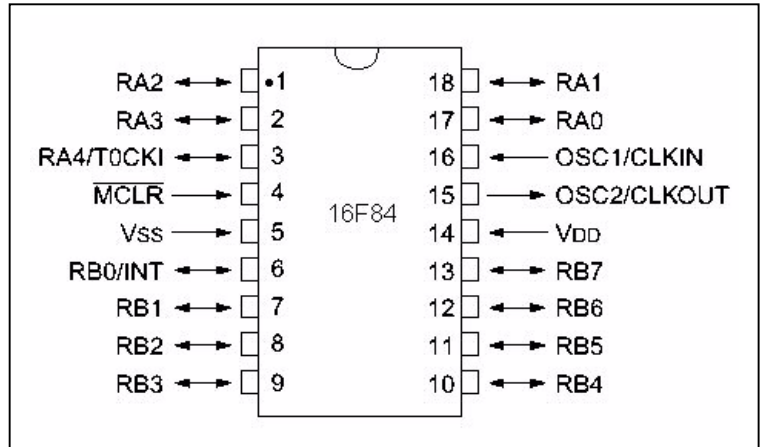
Le microcontrôleur est réalisé en technologie CMOS. Les signaux sont compatibles TTL.

Vss et Vdd : broches d'alimentation
(3 à 5,5 V)

OSC1, OSC2 : signaux d'horloge. Peuvent recevoir un circuit RC ou un résonateur.
CLKIN peut être connectée à une horloge externe. (0 à 4 , 10 ou 20 MHz)

MCLR : Reset (Master Clear).

RA0..RA4 : 5 E/S du port A
RB0..RB7 : 8 E/S du port B



T0CKI : Entrée d'horloge externe du timer TMR0.

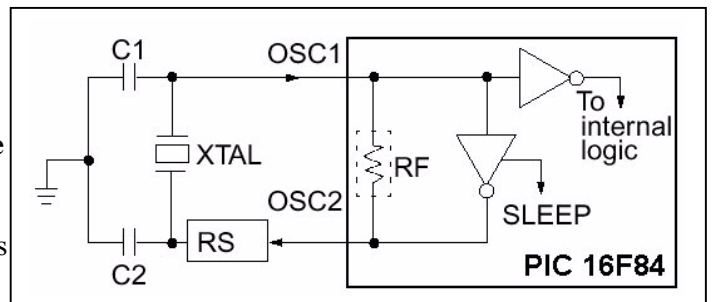
INT : Entrée d'interruption externe

Horloge interne

0 à 4 , 10 ou 20 MHz selon le type de μC .

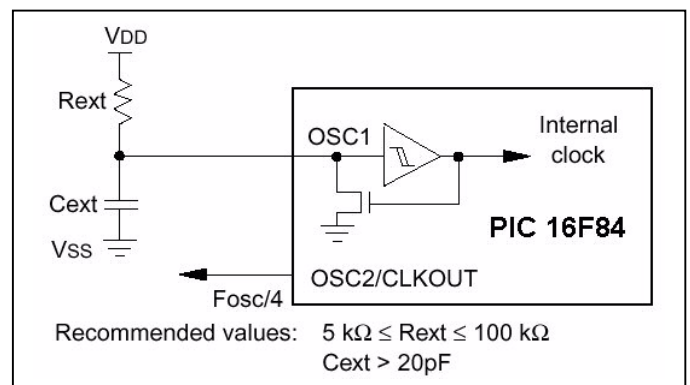
Utilisation d'un résonateur quartz ou céramique
(oscillateur Colpitts).

RF assure la polarisation de l'inverseur CMOS.
Le filtre passe bas RS,C2 limite les harmoniques dus à l'écrêtage (le résonateur vibre en sinus) et Réduit l'amplitude de l'oscillation.



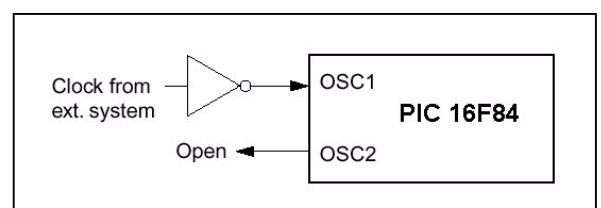
Utilisation d'un circuit RC
(Relaxateur)

Lorsque la tension aux bornes de Cext atteint le seuil haut de la bascule de Schmitt, le transistor MOS devient fortement conducteur et décharge Cext.
Quand la tension au point OSC1 atteint le seuil bas, le transistor MOS se bloque et Cext se (re)charge.



Horloge externe

Dans certains cas, une horloge externe au microcontrôleur peut être utilisée pour synchroniser le PIC sur un processus particulier.
La broche OSC2 est alors inutilisée.



Fichier des registres

Le fichier est un groupe de registres répartis sur deux pages (banques). Il comprend :

- En banque 0, des registres fondamentaux (01 – 0B)
- En page 1, des registres associés (81 – 8B)
- En page 0, des emplacements RAM à usage général pour y loger des variables.

Les adresses 00 et 07 ne contiennent aucune mémoire.

Pour les registres PCL, STATUS, FSR, PCLATH, INTCON, le bit b7 de l'adresse n'existe pas.

La pile (8 registres) n'est pas dans ce fichier.

	Banque 0	Banque 1	
00	Adressage indirect ←		80
01	RTCC	OPTION	81
02	PCL	<i>PCL</i>	82
03	STATUS	<i>STATUS</i>	83
04	FSR	<i>FSR</i>	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07			87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH	<i>PCLATH</i>	8A
0B	INTCON	<i>INTCON</i>	8B
0C – 4F	Usage général ←		8C - CF

00 Cette adresse ne contient pas de registre physique.

01 RTCC (ou TMR0)

Contenu du Timer (8 bits). Il peut être incrémenté par l'horloge ($f_{osc}/4$) ou la broche RA4.

02 PCL

8 bits de poids faible du compteur ordinal PC. Les 5 (13-8) bits de poids forts sont dans PCLATH.

03 STATUS

Registre d'état contenant :

- 5 bits, témoins (drapeaux) caractérisant le résultat de l'opération réalisée par l'ALU. (lecture seule).
- 1 bit de sélection de page (Ecriture/lecture)

RP0	TO/	PD/	Z	DC	C
-----	-----	-----	---	----	---

Au reset, seul le bit RP0 de sélection de page est fixé (0 : page 0)

TO/ (Time Out) : débordement du timer WDT

PD/ : (Power Down) caractérise l'activité du chien de garde WDT

Z (zéro) résultat nul pour une opération arithmétique et logique.

DC (digit carry) retenue sur un quartet (4 bits)

C (carry) retenue sur un octet (8 bits).

04 FSR

Registre de sélection de registre : contient l'adresse d'un autre registre (adressage indirect)

05 PORTA

06 PORTB

Registres de données des ports parallèles (voir détail page 8)

08 EEDATA

Contient un octet lu ou à écrire dans l'EEPROM de données.

09 EEADR

Contient l'adresse de la donnée lue ou écrite dans l'EEPROM de données.

0A PCLATH

Voir 02 PCL.

0B INTCON

Contrôle des 4 interruptions

GIE	EEIE	TOIE	INTE	RBIE	RTIF	INTF	RBIF
-----	------	------	------	------	------	------	------

Masques :**GIE** : (Global Interrupt Enable) : masque global d'inter.**EEIE** : (EEPROM Interrupt Enable)**TOIE** : (Timer 0 Interrupt Enable)**INTE** : (Interrupt Enable) Entrée d'inter sur broche RB0/INT**RBIE** : (RB Interrupt Enable) Entrées d'inter sur broches RB4-RB7.**Drapeaux :****TOIF** : (Timer 0 Interrupt Flag) débordement du timer**INTF** (Interrupt Flag) interruption provoquée par la broche RB0/INT**RBIF** (RB Interrupt Flag) interruption provoquée par les broches RB4-RB7.**81 OPTION**

8 bits (tous à 1 au RESET) affectant le comportement des E/S et des timers.

RBPU/	INTEDG	RTS	RTE	PSA	PS2	PS1	PS0
-------	--------	-----	-----	-----	-----	-----	-----

RBPU/ (RB Pull Up) Résistances de tirage à Vdd des entrées du port B.**INTEDG** (Interrupt Edge) Front actif sur RB0 (1 pour front montant).**RTS** (Real Timer Source) Signal alimentant le timer 0 : 0 pour horloge interne, 1 pour RA4/T0CLK**RTE** (Real Timer Edge) front actif du signal timer (0 pour front montant).**PSA** (Prescaler assignment) 0 pour Timer 0 et 1 pour chien de garde WDT.**PS2..0** (Prescaler 210) Valeur du Diviseur de fréquence pour les timers.**85 TRISA****86 TRISB**

Direction des données pour les ports A et B : 0 pour sortir et 1 pour entrer.

88 EECON1

Contrôle le comportement de l'EEPROM de données.

EEIF	WRERR	WREN	WR	RD
------	-------	------	----	----

EEIF (EEPROM Interrupt Flag) passe à 1 quand l'écriture est terminée.**WRERR** (Write Error) 1 si erreur d'écriture.**WREN** (Write Enable) : 0 pour interdire l'écriture en EEPROM de données.**WR** (Write) 1 pour écrire une donnée. Bit remis automatiquement à 0**RD** (Read) : 1 pour lire une donnée. Bit remis automatiquement à 0**89 EECON2**

Registre de sécurité d'écriture en EEPROM de données.

Une donnée ne peut être écrite qu'après avoir écrit successivement 0x55 et 0xAA dans ce registre :

```

movlw Adresse
movwf EEADR           ;définition de l'adresse

movlw Donnee
movwf EEDATA        ;définition de la donnée

movlw 0x55           ;)
movwf EECON2        ;) sécurité
movlw 0xAA           ;)
movwf EECON2        ;)
bsf EECON1,WR       ; écriture

```

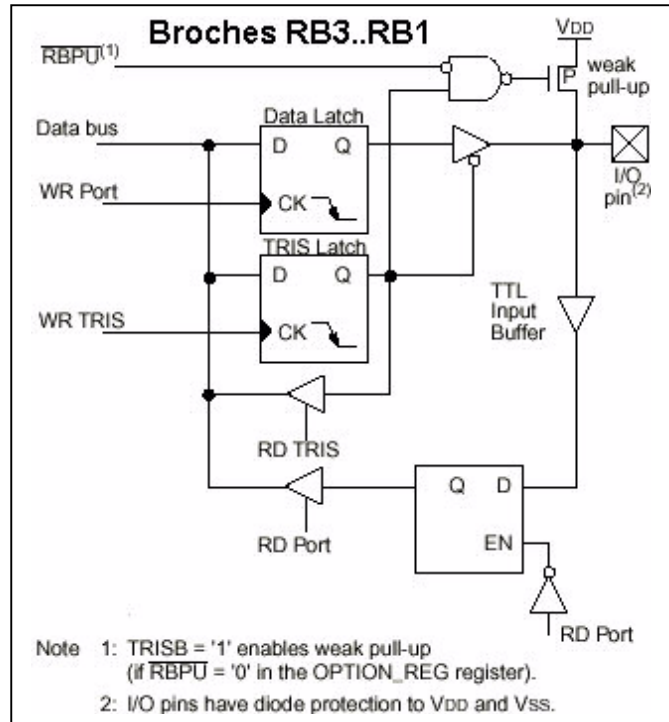
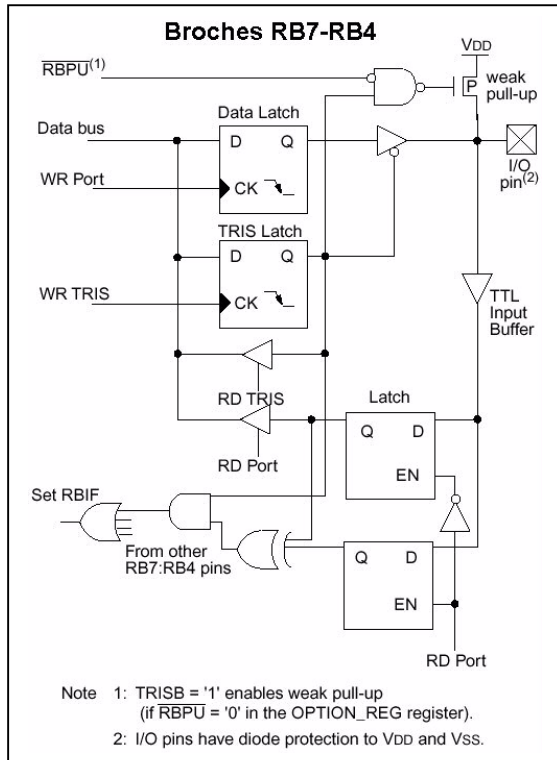

Particularités du port B

Toutes les sorties du port B sont complémentaires et compatibles TTL.

Toutes les entrées du port B sont de technologie CMOS et compatibles TTL. Elles peuvent être pourvues de « résistances » de tirage (weak pull up) validées par le bit RBPU du registre OPTION.

En entrée, une quelconque des lignes RB7..RB4 peut affecter le drapeau RBIF pour demander une interruption si celle-ci n'est pas masquée par RBIE. (voir schéma).

En entrée, la ligne RB0 peut déclencher l'interruption externe RB0/INT si celle-ci est autorisée par les bits INTE et GIE du registre OPTION. Le front déclenchant cette interruption peut être montant ou descendant selon le bit INTEDG de OPTION.



Exemple d'utilisation des ports : Produire un signal sur RB0 dès que RA0 passe à 1

```

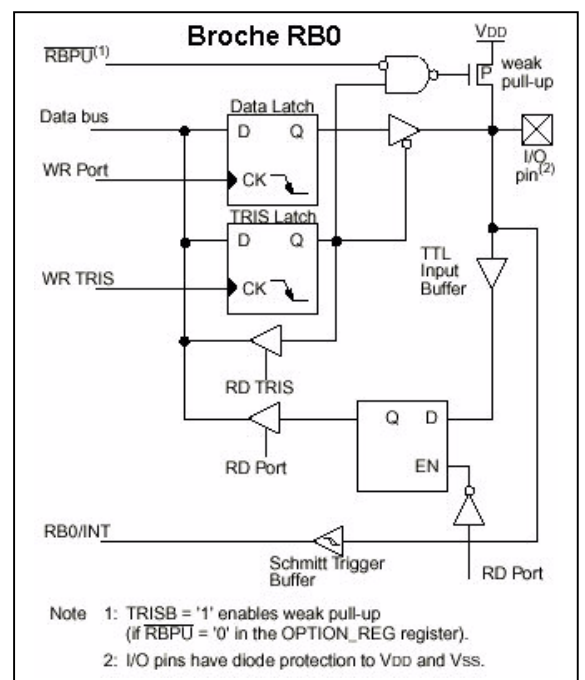
include "registres.h" ; adresses registres

org 0 ;Programme lancé à 0x000

bsf STATUS,RP0 ;passer en page 1
movlw 0xFF
movwf TRISA
movlw ~1 ;RB0 en
movwf TRISB ;sortie
bcf STATUS,RP0 ;repasser en page 0

AttendUnSurRA0 ;attendre que RA0 passe à 1
btfss PORTA,0
goto AttendUnSurRA0

movlw 1
ChangerRB0
xorwf PORTB,F ;changer l'état de RB0
goto ChangerRB0
    
```



Timers

Timer du chien de garde (WDT : WatchDog Timer)

C'est un compteur incrémenté en permanence (free running) par une horloge RC indépendante. Ce compteur, lorsqu'il arrive à FF (à $t = \text{WDT TimeOut}$), est capable de Réinitialiser (RESET MCLR) le microcontrôleur.

En cas de blocage du microcontrôleur, ceci permet d'éviter des processus indésirables.

Le programme doit donc remettre le WDT à 0 périodiquement pour que WDT TimeOut ne soit jamais atteint en fonctionnement normal. Ce Time Out vaut en principe 18 ms. Il est cependant réglable au moyen d'un pré diviseur introduit par programme.

Les instructions qui réinitialisent le WDT ne doivent pas se trouver dans un sous-programme d'interruption ou dans un processus aléatoire.

En mode veille (SLEEP) Le WDT continue à fonctionner mais le μC sortira du mode SLEEP quand le Time Out sera atteint.

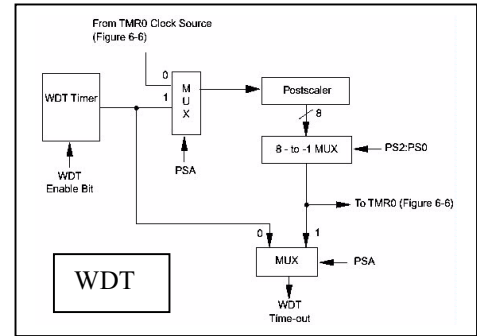
WDT est validé ou non, lors de la programmation de l'EEPROM de configuration du PIC.

Timer « utilisateur » TMR0

C'est aussi un compteur 8 bits libre (incrémenté en permanence par l'horloge interne $F_{osc}/4$) ou commandé par la broche TMR0/RA4 du port A. Dans ce cas, le bit RTE du registre OPTION sélectionne le front actif.

Le contenu de TMR0 est accessible dans le fichier des registres (adresse 01). Toute écriture inhibe l'incrémentation pendant deux cycles d'horloge.

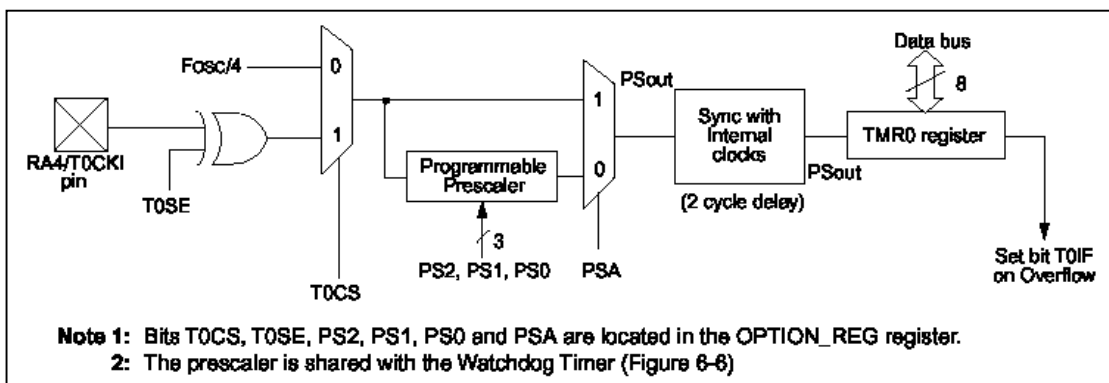
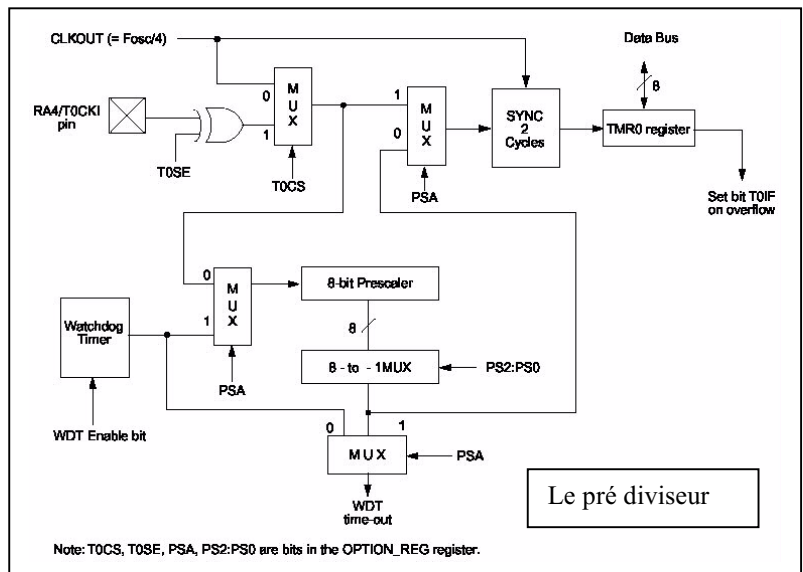
Lorsque son contenu passe de FF à 0, une interruption via le drapeau TOIF peut intervenir.



bit 3: **PSA**: Prescaler Assignment bit
 1 = Prescaler assigned to the WDT
 0 = Prescaler assigned to TMR0

bit 2-0: **PS2:PS0**: Prescaler Rate Select bits

Bit Value	TMR0 Rate	WDT Rate
000	1:2	1:1
001	1:4	1:2
010	1:8	1:4
011	1:16	1:8
100	1:32	1:16
101	1:64	1:32
110	1:128	1:64
111	1:256	1:128



```

; Interruption provoquée par le timer 'temps réel'
; Pour plus de détails sur le traitement des interruption,
; consulter la note AN566
;
;       include "registres.h"
;
;       org 0
;       goto Init
;
;       org 4           ;Inter toujours à l'adresse 4
;
;       btfss INTCON,TOIF ;Vérifier que l'inter a bien
;       goto AutresInter ;été provoquée par le timer
;
;       movlw 2
;       xorwf PORTB,F     ;Complémenter RB1
;       bcf INTCON,TOIF  ;drapeau à 0
;       retfie           ;retour au PP
;
AutresInter           ;Traitement éventuel des autres interruptions
;       nop             ;inutile
;       nop             ;inutile
FinInter             retfie

;----- le programme commence ici -----

; Initialisation
Init                 bsf STATUS,RP0           ;passer en page 1
;
;       movlw ~(1+2)    ;RB0 et RB1
;       movwf TRISB    ;en sortie
;
;       movlw (1<<RBPU) | (1<<PS0) ;predivision par 4 et
;       movwf OPTIO    ;pas de pull Up sur port B
;
;       bcf STATUS,RP0 ;repasser en page 0
;
;       movlw(1<<TOIE) | (1<<GIE) ;Validation locale et
;       movwf INTCON   ;globale inter Timer
;
;déroulement du programme principal
Boucle              movlw 1
;       xorwf PORTB,F   ;Complémenter RB0
;       goto Boucle
;
;       end

```

RESET

Le RESET peut avoir plusieurs causes :

- Mise sous tension
- Etat 0 sur broche MCLR/
- Débordement du timer du chien de garde.

Lorsque le RESET intervient, le μ C peut être

- En fonctionnement normal
- En mode SLEEP.

Cause	Etat	TO/	PD/	PC
Mise sous tension		1	1	000
0 sur broche MCLR/	Normal	NA	NA	000
	Sleep	NA	0	000
Chien de garde	Normal	0	1	000
	Sleep	0	0	PC+1

NA : non affecté

Les 2 bits d'état (registre STATUS) TO et PD peuvent être affectés par le RESET.

Le compteur de programme PC est en général remis à 0.

Les autres registres peuvent être affectés par le RESET, selon le tableau :

Register	Address	Power-on Reset	MCLR Reset during: – normal operation – SLEEP WDT Reset during normal operation	Wake-up from SLEEP: – through interrupt – through WDT Time-out
W	—	xxxx xxxx	uuuu uuuu	uuuu uuuu
INDF	00h	---- ----	---- ----	---- ----
TMR0	01h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCL	02h	0000h	0000h	PC + 1 ⁽²⁾
STATUS	03h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	04h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PORTA	05h	---x xxxx	---u uuuu	---u uuuu
PORTB	06h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEDATA	08h	xxxx xxxx	uuuu uuuu	uuuu uuuu
EEADR	09h	xxxx xxxx	uuuu uuuu	uuuu uuuu
PCLATH	0Ah	---0 0000	---0 0000	---u uuuu
INTCON	0Bh	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾
INDF	80h	---- ----	---- ----	---- ----
OPTION_REG	81h	1111 1111	1111 1111	uuuu uuuu
PCL	82h	0000h	0000h	PC + 1
STATUS	83h	0001 1xxx	000q quuu ⁽³⁾	uuuq quuu ⁽³⁾
FSR	84h	xxxx xxxx	uuuu uuuu	uuuu uuuu
TRISA	85h	---1 1111	---1 1111	---u uuuu
TRISB	86h	1111 1111	1111 1111	uuuu uuuu
EECON1	88h	---0 x000	---0 q000	---0 uuuu
EECON2	89h	---- ----	---- ----	---- ----
PCLATH	8Ah	---0 0000	---0 0000	---u uuuu
INTCON	8Bh	0000 000x	0000 000u	uuuu uuuu ⁽¹⁾

Legend: u = unchanged, x = unknown, - = unimplemented bit read as '0',
q = value depends on condition.

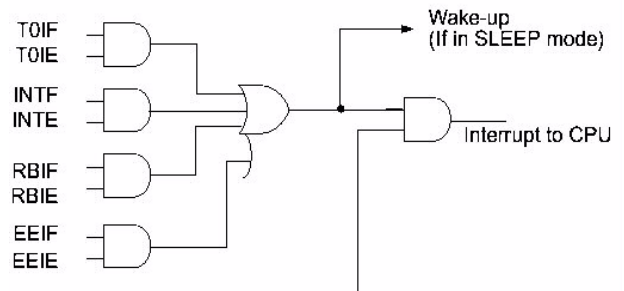
Note 1: One or more bits in INTCON will be affected (to cause wake-up).
2: When the wake-up is due to an interrupt and the GIE bit is set, the PC is loaded with the interrupt vector (0004h).
3: Table 8-3 lists the reset value for each specific condition.

Interruptions

4 sources d'interruption validées

- Globalement par le bit GIE du registre INTCON
- Localement par les bits EEIE, RTIE, INTE et RBIE du registre INTCON

4 drapeaux pouvant provoquer l'interruption correspondante : EEIF, RTIF, INTF, RBIF.



Il n'y a qu'un seul vecteur d'interruption : il contient l'adresse 004.

Le service d'une interruption comporte les actions suivantes :

Début

- Le drapeau correspondant passe à 1.
- PC est empilé puis affecté de l'adresse 004.
- Le masque GIE est automatiquement mis à 0 pour interdire d'autres interruptions.

Fin (instruction RETI)

- PC est dépilé
- GIE est remis à 1

Il incombe au programme :

- d'identifier la source de l'interruption en consultant les drapeaux.
- De remettre à 0 le drapeau qui a provoqué l'interruption.
- De sauvegarder éventuellement certains registres.

Compteur de programme (PC)

Ce registre de 10 bits utiles contient l'adresse (en mémoire programme) de l'instruction en cours .

Une fois l'instruction exécutée, PC est incrémenté d'une unité. Les instructions **call**, **goto** et **return** affectent directement le contenu de PC :

- **Goto** place une valeur (adresse) dans PC
- **Call** empile PC et place une valeur (adresse) dans PC
- **Return** dépile PC

Les huit bits de poids faible de PC sont accessibles à l'adresse 02 du fichier des registres.

La pile

Groupe de 8 registres de 13 bits (Les 3 bits de poids fort ne sont pas utilisés actuellement).

Son rôle est de sauver temporairement le compteur de programme PC (l'adresse courante) lors de l'appel d'un sous-programme ou du service d'une interruption.

La pile n'est pas dans le fichier des registres : elle n'est donc pas manipulable manuellement.

C'est une mémoire tampon circulaire : le 9^{ème} empilement écrase le 1^{er} . Aucune indication sur l'état de la pile n'est accessible.

EEProm de données

Une EEPROM (Electrically Erasable Prom) de 64 octets est disponible pour y stocker des données. Contrairement à l'EEProm de programme, elle n'est pas adressée par le compteur de programme PC mais par un registre séparé EEADR se trouvant à l'adresse 09 dans le fichier des registres. Les données sont accessibles par le registre EEDATA d'adresse 08. Deux octets de contrôle sont associés à cette EEPROM : EECON1 et EECON2 d'adresse 88 et 89.

La durée d'écriture d'un octet étant de l'ordre de 10 ms, un Timer associé à l'EEProm place le bit EEIF de EECON1 à 1 en fin d'écriture.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on Reset	Value on all other resets
08h	EEDATA	EEPROM data register								xxxx xxxx	uuuu uuuu
09h	EEADR	EEPROM address register								xxxx xxxx	uuuu uuuu
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	---0 q000
89h	EECON2	EEPROM control register 2								-----	-----

Legend: x = unknown, u = unchanged, - = unimplemented read as '0', q = value depends upon condition. Shaded cells are not used by Data EEPROM.

Lecture d'une donnée

- Placer l'adresse de la donnée à lire dans EEADR
- Mettre le bit RD de EECON1 à 1
- Lire le contenu du registre EEDATA

EXAMPLE 7-1: DATA EEPROM READ

```
BCF STATUS, RP0 ; Bank 0
MOVLW CONFIG_ADDR ;
MOVWF EEADR ; Address to read
BSF STATUS, RP0 ; Bank 1
BSF EECON1, RD ; EE Read
BCF STATUS, RP0 ; Bank 0
MOVF EEDATA, W ; W = EEDATA
```

Ecriture d'une donnée

- Placer l'adresse de la donnée à écrire dans EEADR
- Placer la donnée à écrire dans EEDATA
- Placer 0x55 dans EECON2 (1)
- Placer 0xAA dans EECON2 (1)
- Mettre le bit WR de EECON1 à 1
- Attendre que le bit EEIF de EECON1 soit à 1. (10 ms) On peut utiliser l'interruption produite par EEIF en la validant par le bit EEIE de INTCON. Ne pas oublier de remettre EEIF à 0.

EXAMPLE 7-1: DATA EEPROM WRITE

```
BSF STATUS, RP0 ; Bank 1
BCF INTCON, GIE ; Disable INTs.
BSF EECON1, WREN ; Enable Write
MOVLW 55h ;
MOVWF EECON2 ; Write 55h
MOVLW AAh ;
MOVWF EECON2 ; Write AAh
BSF EECON1, WR ; Set WR bit
; begin write
BSF INTCON, GIE ; Enable INTs.
```

EXAMPLE 7-1: WRITE VERIFY

```
BCF STATUS, RP0 ; Bank 0
: ; Any code can go here
: ;
MOVF EEDATA, W ; Must be in Bank 0
BSF STATUS, RP0 ; Bank 1
READ
BSF EECON1, RD ; YES, Read the
; value written
BCF STATUS, RP0 ; Bank 0
;
; Is the value written (in W reg) and
; read (in EEDATA) the same?
;
SUBWF EEDATA, W ;
BTFSZ STATUS, Z ; Is difference 0?
```

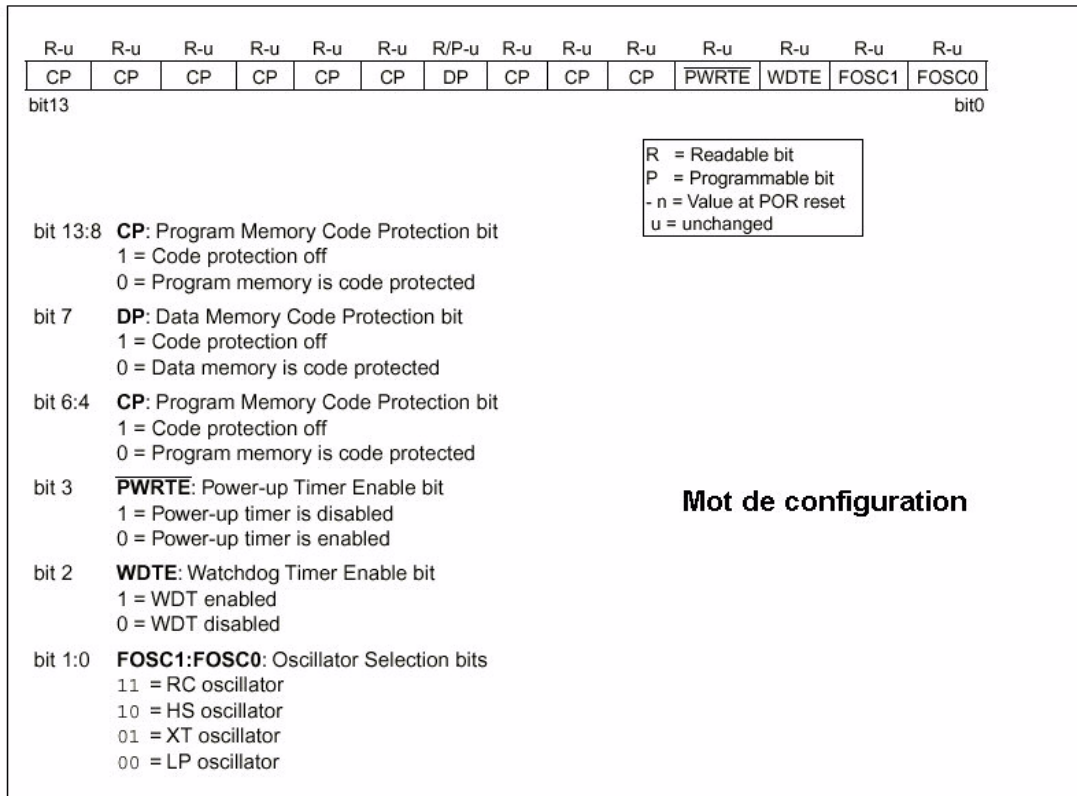
(1) processus de sécurité pour éviter une écriture accidentelle

Configuration

Au cours de la programmation du PIC, des bits de configuration sont écrit dans une EEPROM particulière d'une largeur de 14 bits.

On y trouve :

- **4 mots d'identification (ID)** à partir de l'adresse 0x2000 pouvant contenir un repérage quelconque.
- **1 mot de configuration** à 0x2007 qui permet
 - de choisir le composant utilisé pour l'horloge (résonateur ou RC)
 - de valider ou non les timers TMR0 et WDT
 - d'interdire la lecture des mémoires EEPROM de programme et de données.



Le PIC 16F84 est programmable In Situ au moyen de signaux sériels dont l'organisation est proche du protocole I2C.

Pendant la programmation,

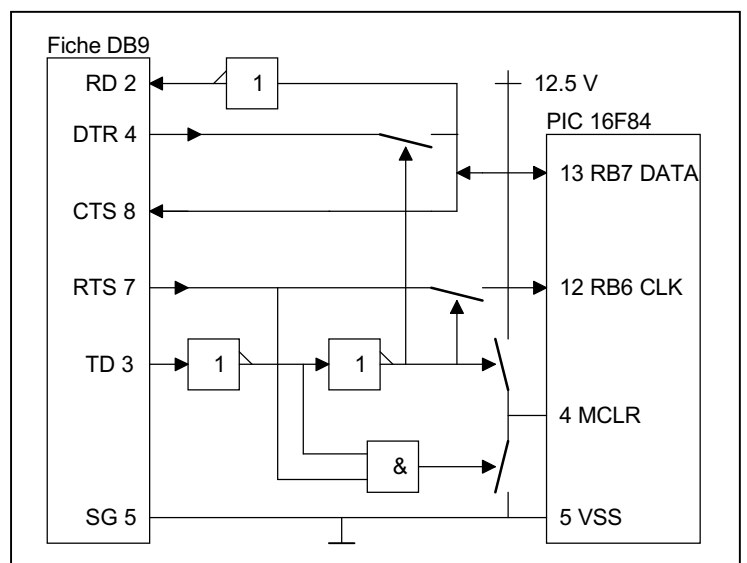
- Tension de 12..14 V appliquée sur MCLR
- RB6 reçoit l'horloge de communication
- RB7 reçoit ou envoie les données série.

Le protocole de programmation est décrit dans la note AN566.

De nombreux schémas de structure de programmeurs ont été publiés

Exemple de programmeur placé sur le port série d'un ordinateur (principe).

Le logiciel gère les signaux de programmation.



Résumé des instructions (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 zéro dans f
clrw	place zéro dans W et fait passer Z à 1
clrwdt	initialise le timer du chien de garde
comf f,d	complémente f à 1. Résultat dans W si d=0 ou dans f si d=1
decf f,d	décrémente f. Résultat dans W si d=0 ou dans f si d=1
decfsz f,d	décrémente f. Si f=0, incrémente PC. Résultat dans W ou dans f
goto k	charge une adresse dans PC
incf f,d	incrémente F
incfsz f,d	incrémente f. Résultat dans W si d=0 ou dans f si d=1
iorlw k	effectue un OU inclusif 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
movlw k	charge une constante k dans W
movwf f	stocke W vers f
nop	ne fait rien en gaspillant une période d'horloge
retfie	dépile PC pour retour d'une interruption
retlw k	dépile PC pour retour d'un sous programme et place la cste k dans W
return	dépile PC pour retour d'un sous programme
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	soustrait 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	permuté les deux quartets de f. Résultat dans W (d=0) ou dans f (d=1)
xorlw k	OU exclusif entre une constante k et W. Résultat dans W
xorwf k	OU exclusif entre W et f. Résultat dans W si d=0 ou dans f si d=1