



MICROCONTROLEUR

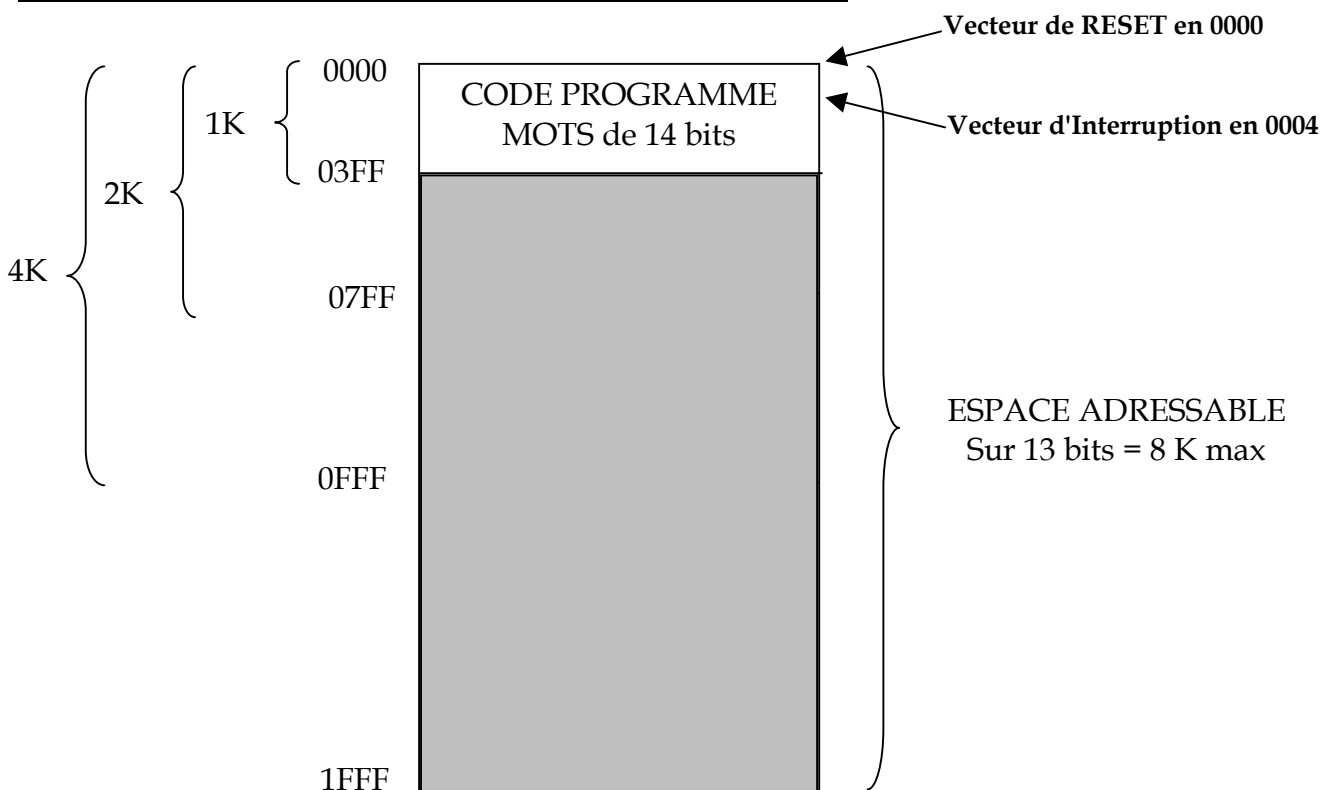
PIC 16F84

Documentation interne ELE
D. MENESPLIER 2000

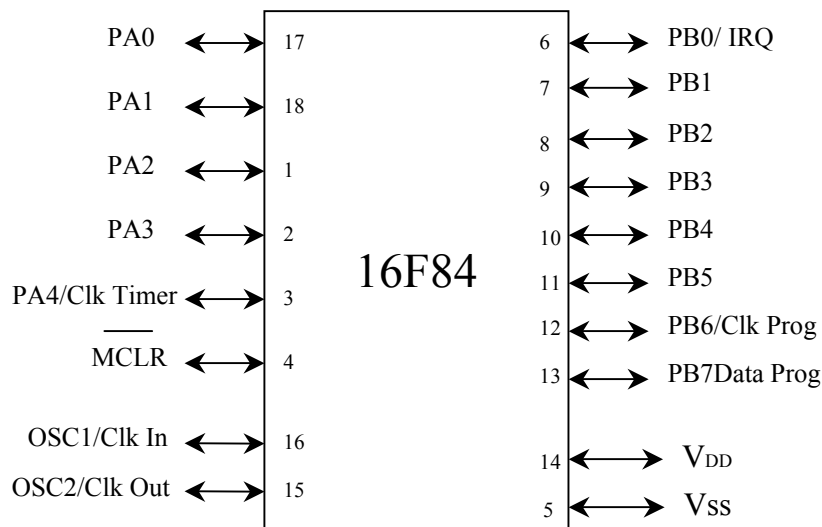
DESCRIPTION :

- Consommation : 2mA sous 5V à 4 MHz.
- Architecture RISC : 35 instructions de durée 1 ou 2 cycles.
- Durée du cycle : Période de l'oscillateur quartz divisée par 4 soit 400 ns pour un quartz de 10 MHz.
- Deux bus distincts pour le code programme et les data.
- Code instruction : mot de 14 bits et compteur programme (PC) sur 13 bits, ce qui permet d'adresser 8 K mots (de h'0000' à h'1FFF')
- Bus DATA sur 8 bits.
- 13 Ports Entrée-Sortie bidirectionnels pouvant produire 25 mA par sortie. PORTA = 5 bits et PORTB=8bits.
- 4 sources d'interruption :
 - Externe par la broche partagée avec le Port B : PB₀
 - Par changement d'état des bits du Port B: PB4 PB5 PB6 ou PB7
 - Quand l'écriture de Data en EEPROM est terminée.
 - Par débordement du Timer.
- Compteur 8 bits avec pré diviseur programmable.
- 68 Octets de RAM
- 64 Octets d'EEPROM Data.
- 1K mots de 14 bits en EEPROM FLASH pour le programme (h'000' à h'3FF').
- 1 registre de travail : W et un registre fichier : F permettant d'accéder à la RAM ou aux registres internes du PIC. Tous les deux sont des registres 8 bits.

ORGANISATION de l'ESPACE PROGRAMME :



BROCHAGE :



ORGANISATION de la MEMOIRE DATA :

Elle est constituée de deux zones :

- La zone SFR (Special function registers) qui contient les registres de configuration et de fonction du microcontrôleur.
- La zone GPR (General Purpose Register) qui contient la RAM.

Elle est découpée en deux pages : Bank 0 et Bank 1. L'accès à la Bank 0 se fait en positionnant le bit 5 du registre STATUS (RP0) à "0". Pour l'accès à la Bank 1 on doit mettre ce bit à "1".

Chaque page contient 128 bits (de 00 à 7f), et nécessite donc 7 bits d'adresse. Les 12 premiers octets de chaque page sont réservés pour la zone SFR. Le reste est attribué à la zone GPR pour de la RAM.

ZONE GPR:

Elle contient des mots de 8 bits. On peut y accéder par adressage direct ou indirect (en utilisant le registre FSR qui sera vu plus tard).

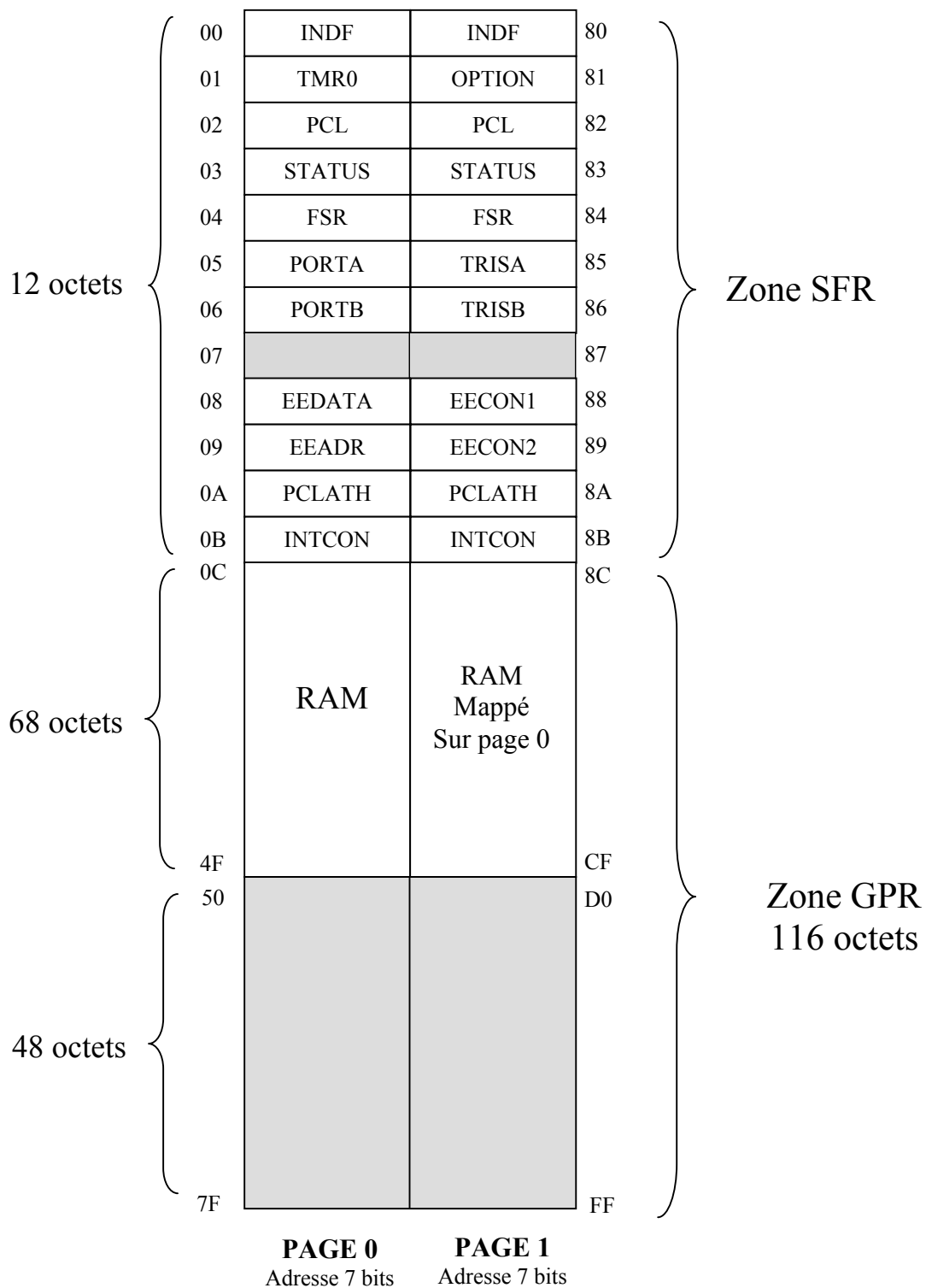
Les adresses GPR en page 1 sont confondues avec celles de la page 0.

On atteint la même adresse en RAM quand on adresse en h'0C' ou en h'8C'.

ZONE SFR:

Elle contient les registres utilisés par le microcontrôleur et les périphériques internes pour contrôler le fonctionnement du circuit. Ces registres sont du type RAM Statique. Ils sont répartis dans les deux pages BANK 0 et BANK 1. Ils occupent les 12 premiers octets de ces deux pages.

MAPPING de la mémoire DATA:



REGISTRES en PAGE 1: OPTION TRISA TRISB EECON1 EECON2

ADRESSAGE DIRECT :

Accès à la zone DATA par adresse absolue sur 7 bits. Donc il faut au préalable choisir la page concernée : Bank 0 avec RP0=0 de STATUS et Bank 1 avec RP0=1 pour accéder aux registres de la zone SFR. Pour la RAM en zone GPR, ce n'est pas nécessaire car la Bank 1 est confondue avec la Bank 0.

ADRESSAGE INDIRECT:

Accès aux registres et RAM de la zone DATA par utilisation du registre FSR sur 8 bits. On écrit dans ce registre l'adresse de la zone SFR ou GPR que l'on veut atteindre. La donnée à lire ou à écrire dans le registre sélectionné indirectement est accessible par le registre INDF à l'adresse h'00' ou h'80'.

Le registre INDF n'est pas un registre physique. Quand on adresse INDF, on accède au registre dont les 8 bits d'adresse sont contenus dans le registre FSR en h'04', qui se comporte comme un pointeur.

Exemple :

- Le registre à l'adresse h'05' contient la valeur : h'10'.
- Le registre à l'adresse h'06' contient la valeur : h'0A'.
- On écrit la valeur h'05' dans FSR.

Si on lit le registre INDF à l'adresse h'00' on obtient la valeur : h'10'.

On incrémente FSR de 1 et on relit INDF en h'00' on obtient la valeur : h'0A'

La lecture du registre INDF lui même, en mettant la valeur h'00' dans FSR donne la valeur h'00'.

L'écriture dans INDF lui même se traduit par un NOP (non opération).

REGISTRE de CONFIGURATION en h'2007' :

Dans la zone Programme à l'adresse h'2007', se trouve un mot de 14 bits qui permet de configurer les particularités du PIC . Cette adresse se trouve juste sous les 8 K max (de 0000 à 1FFF) adressable du programme par le compteur ordinal. On ne peut accéder à l'adresse h'2007' qu'en phase de programmation..

- Protection du code.
- Mise en service du Watch Dog.
- Power Up Timer.
- Choix du type d'oscillateur.

B ₁₃										B ₀			
CP	CP	CP	CP	CP	CP	CP	CP	CP	CP	$\overline{\text{PWRTE}}$	WDTE	FOSC 1	FOSC 0

Bits 4 à 13 : **CP** = Protection Code.

- Si "0" toute la zone mémoire de programme est protégée.
- Si "1" pas de protection du code.

Bit 3 : **PWRTE** = Power Up Timer Enable. Délai fixe de 72 ms, qui maintient l'UC au Reset et permet ainsi à la tension d'alimentation de bien se stabiliser.

- Si "0" le Power Up Timer est autorisé.
- Si "1" le Power Up Timer est non activé.

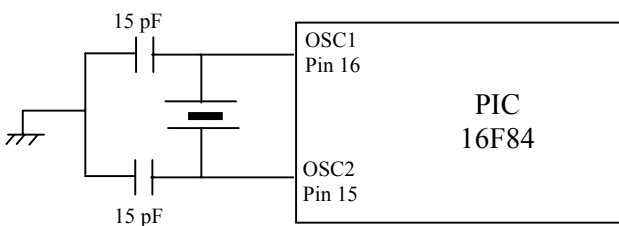
Bit 2 : **WDTE** = Watchdog Timer Enable.

- Si "0" le Watchdog n'est pas activé.
- Si "1" le Watchdog est activé.

Bit 1 et Bit 0 : **FOSC1 et FOSC2** = Oscillator Selection.

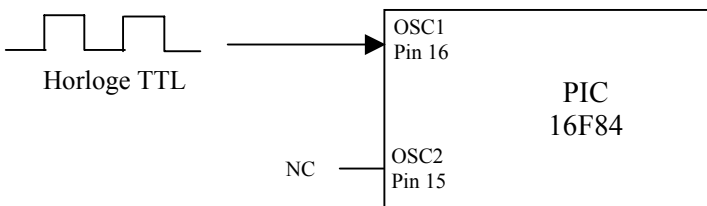
- Si "11" : RC oscillator : oscillateur avec résistance et capacité (<4MHz).
- Si "10" : HS oscillator : quartz ou résonateur haute fréquence (>4 Mhz).
- Si "01" : XT oscillator : quartz ou résonateur standard (<4MHz).
- Si "00" : LP oscillator : quartz basse fréquence (<200KHz) pour applications à faible consommation.

Montage avec un Quartz ou un Filtre céramique : MODE: XT, HS ou LP

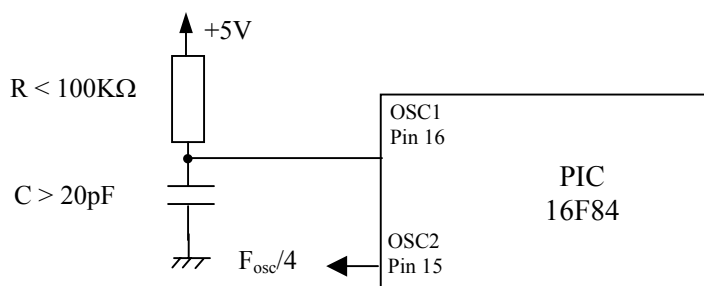


$$\text{Fréquence Cycle machine} = \frac{F_{osc}}{4}$$
$$\text{Cycle machine} = 4 \times T_{osc}$$

Montage avec une Horloge Externe : MODE: XT, HS ou LP



Montage avec oscillateur RC : MODE: RC



ATTENTION
Dans ce mode ne pas entrer un signal externe par la pin 16.
Risque de destruction du circuit

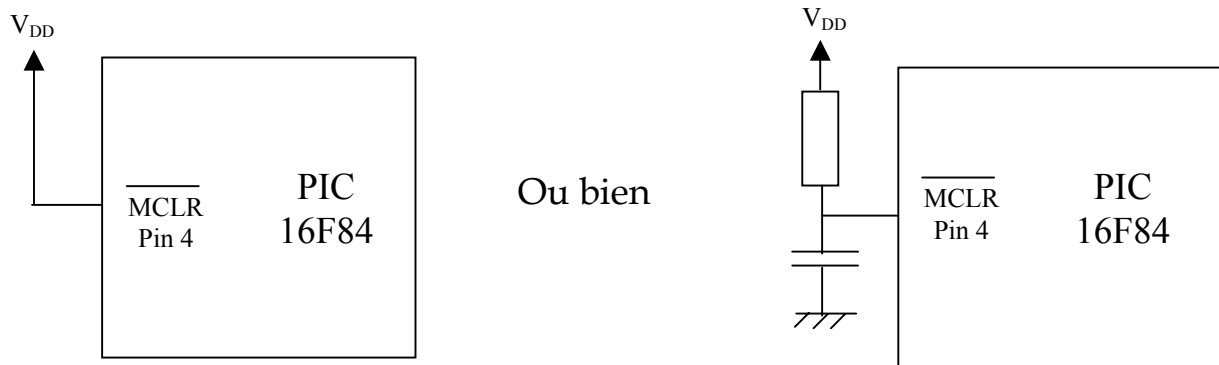
POWER ON RESET: POR

Un pulse interne (POR) est généré dans le Chip quand une montée de la tension d'alimentation V_{DD} est détectée (entre 1,2V et 1,7V). Le délai du Timer Power Up de 72 ms est lancé. Le délai Oscillateur Time Out interne est alors lancé (1024 cycles). Il permet d'attendre que l'oscillateur soit bien stabilisé.

C'est à partir de cet instant que le Reset interne a lieu. A condition que la broche MCLR soit bien à V_{DD} .

Il suffit donc en général de tirer la broche MCLR (pin 4) à la tension d'alimentation soit directement soit à travers une résistance.

Si le taux de montée de V_{DD} est trop faible, il faudra installer le circuit habituel RC sur la pin 4 pour avoir un Reset correct.



LES REGISTRES INTERNES

STATUS REGISTER : (h'03' ou h'83').

On accède indifféremment à ce registre par une quelconque de ces 2 adresses.

Bit 7

Bit 0

IRP	RP1	RP0	$\overline{\text{TO}}$	$\overline{\text{PD}}$	Z	DC	C
-----	-----	-----	------------------------	------------------------	---	----	---

Au reset : STATUS = 00011XXX

Bit 7 : **IRP** = permet la sélection des pages en adressage indirect.

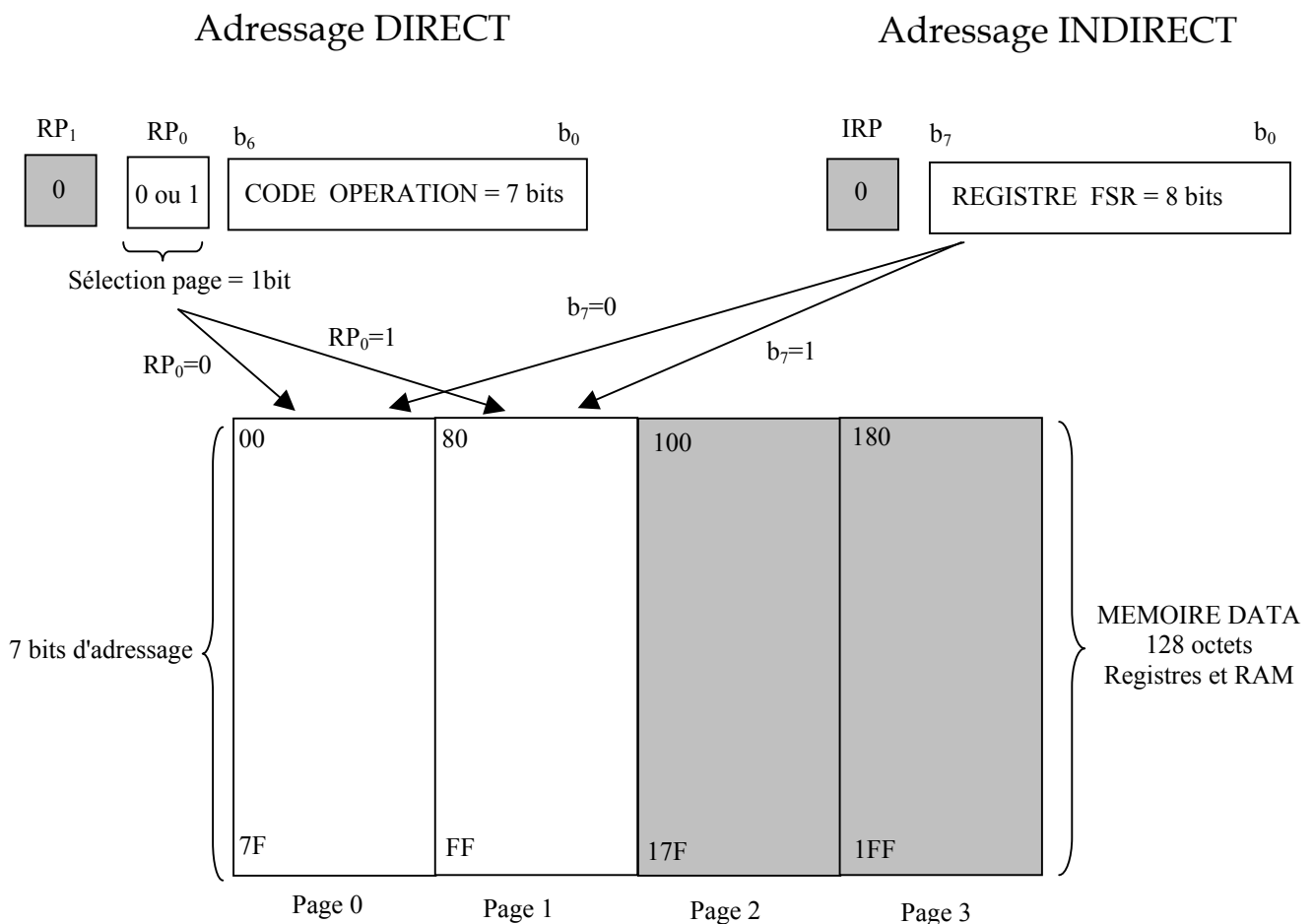
Pour les deux pages possibles du PIC 16F84 (de 00 à 7F et de 80 à FF) ce bit doit être laissé à "0". Mis à "1" il permettra d'atteindre les pages 3 et 4 (de 100 à 17F et de 180 à 1FF) pour les futurs produits de Microchip.

Bits 6 et 5 : **RP1 et RP0** = permettent la sélection des pages en adressage direct.

RP1	RP0	Page sélectionnée
0	0	PAGE 0 de 00 à 7F
0	1	PAGE 1 de 80 à FF
1	0	PAGE 2 de 100 à 17F
1	1	PAGE 3 de 180 à 1FF

Pour les deux pages possibles du PIC 16F84 (de 00 à 7F et de 80 à FF) le bit RP1 doit être laissé à "0". Mis à "1" il permettra d'atteindre les pages 3 et 4 (de 100 à 17F et de 180 à 1FF) pour les futurs produits de Microchip.
Le seul bit actif est RP0 qui permet de choisir la page de 128 bits désirée.

Exemple: `BSF STATUS,5` ; permet d'accéder à la Page 1 car RP0=1
`BCF STATUS,5` ; permet l'accès à la Page 0 car RP0=0



Bit 4 : $\overline{\text{TO}}$ = Time Out bit.

Bit en lecture seulement.

1 = Après une mise sous tension, après une RAZ du watchdog (CLRWDT) ou bien après l'instruction SLEEP.

0 = Signifie qu'un Time Out du timer de watchdog est survenu.

Bit 3 : $\overline{\text{PD}}$ = Power Down bit.

1 = Après une mise sous tension ou bien après une RAZ du Watchdog.

0 = Après l'instruction SLEEP.

Bit 2 : Z = Zero bit.

1 = Le résultat d'une opération arithmétique ou logique est zéro.

0 = Le résultat d'une opération arithmétique ou logique est différent de zéro.

Bit 1 : DC = Digit Carry bit.

1 = Une retenue sur le 4eme bit des poids faible est survenue après les instructions : ADDWF et ADDLW.

0 = Pas de retenue sur le 4eme bit des poids faible.

Bit 0 : C = Carry bit / Borrow .

1 = Une retenue sur le bit MSB est survenue après les instructions ADDWF et ADDLW.

0 = Pas de retenue sur le bit MSB.

Pour une Borrow, l'information est inversée (après SUBLW et SUBWF)

OPTION REGISTER : (h'81').

Ce registre en lecture écriture permet de configurer les prédiviseurs du Timer et du Watchdog, la source du Timer, le front des interruptions et le choix du Pull ups sur le Port B..

Bit 7

Bit 0

$\overline{\text{RBPU}}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
--------------------------	--------	------	------	-----	-----	-----	-----

Au reset : OPTION = 11111111

Bit 7 : $\overline{\text{RBPU}}$ = Pull up Enable bit on Port B.

1 = Pull up désactivé sur le Port B.

0 = Pull up activé.

Bit 6 : INTEDG = Interrupt Edge select bit.

1 = Interruption si front montant sur la broche PB0/IRQ (pin 6).

0 = Interruption si front descendant sur PB0/IRQ.

Bit 5 : TOCS = Timer TMR0 Clock Source select bit.
 1 = L'horloge du Timer est l'entrée PA4/Clk (pin 3).
 0 = Le Timer utilise l'horloge interne du PIC.

Bit 4 : TOSE = Timer TMR0 Source Edge select bit.
 1 = Le Timer s'incrémente à chaque front montant de la broche PA4/Clk.
 0 = Le Timer s'incrémente à chaque front descendant de la broche PA4/Clk.

Bit 3 : PSA = Prescaler Assignment bit.
 1 = Le prédiviseur est affecté au watchdog..
 0 = Le prédiviseur est affecté au Timer TMR0.

Bits 2 à 0 : PS2 PS1 PS0 = Prescaler Rate Select bits.

PS2	PS1	PS0	Prédiv Timer	Prédiv Watchdog
0	0	0	2	1
0	0	1	4	2
0	1	0	8	4
0	1	1	16	8
1	0	0	32	16
1	0	1	64	32
1	1	0	128	64
1	1	1	256	128

Quand le prédiviseur est affecté au Watchdog (PSA=1), TMR0 est prédivisé par 1.

INTCON REGISTER : (h'0B' ou h'8B').

Ce registre en lecture écriture permet de configurer les différentes sources d'interruption.

Bit 7

Bit 0

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

Au reset : INTCON = 0000000X

Bit 7 : GIE = Global Interrup Enable bit
 1 = Autorise toutes les interruptions non masquées.
 0 = Désactive toutes les interruptions.

Bit 6 : **EEIE** = EEPROM Write Complete Interrup Enable bit.

1 = Autorise l'interruption causée par la fin d'écriture en EEPROM.

0 = Désactive l'interruption causée par la fin d'écriture en EEPROM.

Bit 5 : **TOIE** = Timer TMR0 Overflow Interrup Enable bit.

1 = Autorise les interruptions du Timer TMR0.

0 = Désactive les interruptions du Timer TMR0.

Bit 4 : **INTE** = RB0/Int Interrup Enable bit.

1 = Autorise les interruptions sur la broche : PB0/IRQ (pin6).

0 = Désactive les interruptions sur la broche : PB0/IRQ (pin6).

Bit 3 : **RBIE** = RB Port Change Interrup Enable bit.

1 = Autorise les interruptions par changement d'état du Port B (PB4 à PB7).

0 = Désactive les interruptions par changement d'état du Port B (PB4 à PB7).

Bit 2 : **TOIF** = Timer TMR0 Overflow Interrup Flag bit.

1 = Le Timer à débordé. Ce flag doit être remis à zéro par programme.

0 = Le Timer n'a pas débordé.

Bit 1 : **INTF** = RB0/Int Interrup Flag bit.

1 = Une interruption sur la broche PB0/IRQ (pin 6) est survenue.

0 = Pas d' interruption sur la broche PB0/IRQ (pin 6).

Bit 0 : **RBIF** = RB Port Change Interrup Flag bit. Ce flag doit être remis à zéro par programme. Ceci n'est possible qu'après une lecture du Port B.

1 = Quand au moins une entrée du port B (de PB4 à PB7) a changé d'état.

0 = Aucune entrée de PB4 à PB7 n'a changé d'état.

PCL REGISTER : (h'02' ou h'82').

Le compteur de programme est sur 13 bits. Les 8 bits de poids faible sont dans le registre PCL qui est en lecture/écriture.

PILE DE SAUVEGARDE (STACK) :

Le PIC 16F84 possède 8 niveaux physiques de pile (sur 13 bits) pour la sauvegarde du compteur programme avant un saut à sous programme (CALL) ou quand une interruption apparaît.

Les 13 bits de la valeur du compteur sont restitués à l'instruction de retour sous programme ou retour d'interruption (RETURN et RETLWou RETFIE).

La pile est un buffer circulaire. Quand la pile a été sauvée 8 fois, la 9^{eme} valeur prend la place de la 1^{ere} valeur stockée.

LES PORTS ENTREE / SORTIE

PORTA : (h'05')

Il comporte 5 bits. PA4 est une entrée trigger de Schmitt et une sortie à drain ouvert. Elle peut également servir d'entrée pour le décomptage du Timer TMR0.

Les autres bits PA0 à PA3 sont des entrées/sorties compatibles TTL .

Chaque broches peut être configurée en entrée ou en sortie grâce à un registre de direction : TRISA.

TRISA : (h'85')

1 = Le bit correspondant est configuré en ENTREE. Le driver de sortie correspondant est placé en mode haute impédance.

0 = Le bit correspondant est configuré en SORTIE. Le contenu du Latch correspondant est chargé sur la broche de sortie.

PORTB : (h'06') et TRISB : (h'86')

Il comporte 8 bits. Le registre de direction correspondant est TRISB.

Si on écrit un "1" dans le registre TRISB, le driver de sortie correspondant passe en haute impédance. Si on écrit un "0", le contenu du Latch de sortie correspondant est recopié sur la broche de sortie.

Chaque broche du PORT B est munie d'un tirage au $+V_{DD}$ que l'on peut mettre ou non en service en mode entrée uniquement. On active cette fonction par la mise à "0" du bit 7 dans le registre OPTION en h'81'.

Au reset, le tirage est désactivé.

Il est inactif quand le port est configuré en sortie.

Les 4 broches PB7 PB6 PB5 et PB4 provoquent une interruption sur un changement d'état si elles sont configurées en ENTREE.

On doit remettre à zéro le Flag de cette interruption (bit 0 du registre INTCON en h'0B') dans le programme d'interruption.

Cette possibilité d'interruption sur un changement d'état associé à la fonction de tirage configurable sur ces 4 broches, permet l'interfaçage facile avec un clavier. Cela rend possible le réveil du PIC en mode SLEEP par un appui sur une touche du clavier.

Au RESET: PORT A et PORT B configurés en ENTREE (TRISA et TRISB = 1)
PORT B : Tirage désactivé.

LA MEMOIRE DE DONNEE EEPROM

Cette mémoire est en lecture/écriture. Elle n'est pas placée dans la zone de mémoire DATA et sera donc adressée indirectement par 4 registres de la zone SFR, il s'agit des registres suivant :

- EECON1 en h'88'
- EECON2 en h'89'
- EEDATA en h'08'
- EEADR en h'09'

Le PIC 16F84 possède 64 octets d'EEPROM dont les adresses vont de 00 à 3F.

EEADR : Registre d'adresse.

Ce registre permet d'adresser jusqu'à 256 octets d'EEPROM mais seulement les 64 premiers octets sont implémentés.

Les deux bits de poids forts doivent donc être positionné à "0" pour être certain de bien accéder aux 64 premiers octets.

EEDATA : Registre données.

C'est le registre dans lequel transitent les données à écrire ou lues.

EECON1 : Registre de contrôle.

Ce registre de contrôle possède 5 bits.

Bit 7

Bit 0

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

Bits 7 à 5 : non implémentés.

Bit 4 : **EEIF** : EEPROM Write Operation Interrup Flag bit.

1 = L'opération d'écriture est terminée. Ce Flag doit être remis à zéro par soft.

0 = L'opération d'écriture n'est pas terminée ou n'a pas encore démarré.

Bit 3 : **WRERR** : EEPROM Error Flag bit.

1 = Une opération d'écriture s'est terminée prématurément à cause d'un reset ou de l'entrée en action du watchdog pendant la phase d'écriture.

0 = L'opération d'écriture s'est passé correctement.

Bit 2 : **WREN** : EEPROM Write Enable bit.

1 = Autorise les cycles d'écriture.

0 = Interdit l'écriture dans l'EEPROM.

Bit 1 : **WR** : Write Control bit

1 = Démarre un cycle d'écriture. Ce bit ne peut qu'être mis à "1" par soft. Il est remis à "0" par hard quand l'écriture est terminée.

0 = Le cycle d'écriture en EEPROM est terminé.

Bit 0 : **RD** : Read Control bit.

1 = Débute un cycle de lecture de l'EEPROM. Ce bit ne peut qu'être mis à "1" par soft. Il est remis à "0" par hard.

0 = Ne débute pas un cycle de lecture.

EECON2 : Registre de contrôle.

Ce n'est pas un registre physique. Il est utilisé uniquement pendant les séquences d'écriture en EEPROM.

LECTURE de l'EEPROM:

- 1 - Ecrire l'adresse désirée dans le registre d'adresse EEADR.
- 2 - Mettre le bit RD de EECON1 à "1" pour démarrer la lecture.
- 3 - Lire la donnée disponible dans le registre data EEDATA.

Exemple:

BCF	STATUS,5	; Accès page 0.
MOVLW	adresse	; adresse est une valeur entre 00 et 3F.
MOVWF	EEADR	; écriture adresse dans registre Adresse.
BSF	STATUS,5	; accès page 1.
BSF	EECON1,0	; bit RD = 1 pour débiter lecture.
BCF	STATUS,5	; Accès page 0.
MOVF	EEDATA,0	; lecture Data et mise dans W.

ECRITURE dans l' EEPROM:

- 1 - Ecrire l'adresse désirée dans le registre d'adresse EEADR.
- 2 - Mettre la donnée dans le registre data EEDATA.
- 3 - Séquence d'initialisation : Il est nécessaire d'exécuter ensuite une séquence spécifique pour lancer la procédure d'écriture, pour chaque octet.

Il faut écrire la valeur h'55' dans EECON2, puis écrire la valeur h'AA' dans ce même registre EECON2 et ensuite faire passer le bit de démarrage du cycle d'écriture WR (b1 de EECON1) à "1".

NB:

- On doit tout d'abord mettre le bit d'autorisation d'écriture WREN (b2 de EECON1) à "1". Il faudra le remettre à "0" en fin d'écriture car contrairement au bit WR il n'y repasse pas automatiquement.
- Il faut inhiber les interruptions pendant la séquence d'initialisation.
- La fin du cycle d'écriture est détecté par la montée à "1" du flag EEIF (b4 de EECON1). Il faut le remettre à "0" par programme.

Exemple:

```

BCF      STATUS,5 ; Accès page 0.
MOVLW   adresse  ; Adresse est une valeur entre 00 et 3F.
MOVWF   EEADR    ; Ecriture adresse dans registre Adresse
MOVLW   data     ; Data est une valeur entre 00 et FF.
MOVWF   EEDATA   ; Ecriture data dans le registre de DATA.
BSF     STATUS,5 ; Accès page 1.
BCF     INTCON,7 ; Interdit les IT par bit GIE = 0.
BSF     EECON1,2 ; Autorise écriture par bit WREN = 1.

```

Séquence d'initialisation	MOVLW	h'55'	; Début de la séquence d'initialisation.
	MOVWF	EECON2	; Ecriture valeur h"55" dans EECON2.
	MOVLW	h'AA'	
	MOVWF	EECON2	; Ecriture valeur h"AA" dans EECON2.
	BSF	EECON1,1	; Démarrage cycle écriture par bit WR = 1.

```

TESTFL  BTFSS    EECON1,4 ; Test du flag "fin d'écriture".
        GOTO    TESTFL   ; pas fini car flag EEIF = 0.
        BCF     EECON1,4 ; C'est fini, RAZ du flag EEIF obligatoire.
        BCF     EECON1,2 ; On interdit l'écriture par bit WREN = 0.

```

VERIFICATION ECRITURE :

Après avoir écrit en EEPROM il se doit de vérifier si la valeur écrite est bien celle désirée.

Exemple:

```

BCF      STATUS,5 ; Accès page 0.
MOVWF   EEDATA,0 ; W = data que l'on vient d'écrire.
READ    BSF     STATUS,5 ; Accès page 1
        BSF     EECON1,0 ; bit RD = 1 : pour lancer cycle lecture.
        BCF     STATUS,5 ; Accès page 0.
        SUBWF  EEDATA,0 ; W = W - EEDATA
        BTFSS  STATUS,2 ; Test du bit Z. (si Z=1 c'est OK )
        GOTO   ERREUR   ; Z = 0 : donc data écrite # Data à écrire.

```

Suite programme car écriture est OK.

LE TIMER

Le compteur/Timer TMR0 a les caractéristiques suivantes :

- Compteur sur 8 bits.
- Lecture / écriture de TMR0.
- Prédiviseur 8 bits programmable.
- Choix de l'horloge : interne en Timer et externe en compteur.
- Interruption au débordement (passage de FF à 00).
- Choix du front de l'horloge en mode horloge externe.

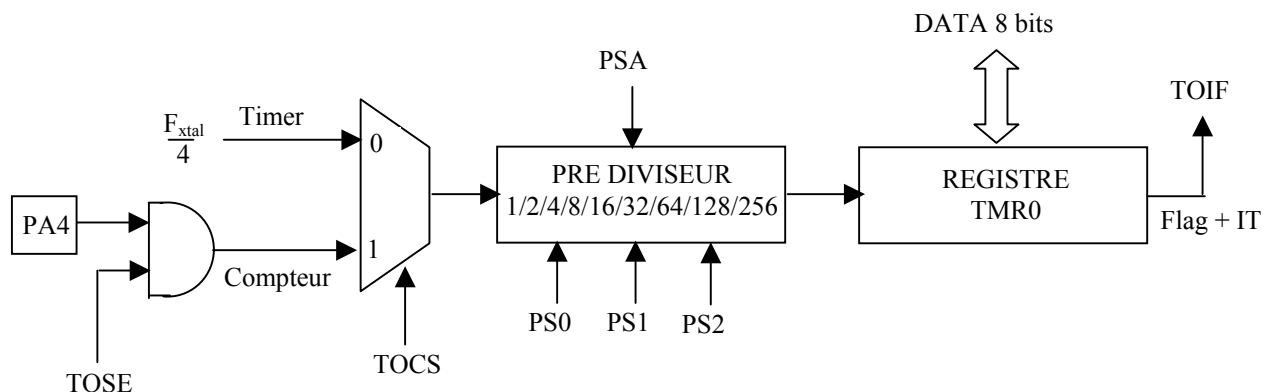
Tous les bits de configuration de TMR0 sont dans le registre OPTION en h'81'.
Le registre TMR0 est à l'adresse h'01'.

Mode TIMER : Le choix de ce mode se fait par : TOCS = 0 (b5 de OPTION).
TMR0 est incrémenté à chaque cycle instruction ($F_{osc}/4$), en considérant le prédiviseur avec un rapport de 1.

Mode COMPTEUR : Ce mode est sélectionné si TOCS = 1. TMR0 est alors incrémenté à chaque front montant ou descendant sur la broche PA4/CLK (pin3). Le choix du front est fait par le bit TOSE (b4 de OPTION).
Si TOSE = 0 le compteur s'incrémente à chaque front montant.
Si TOSE = 1 c'est le front descendant qui incrémente le compteur.

LE PREDIVISEUR : Il est partagé entre le Watchdog et TMR0.
L'affectation se fait par le bit PSA (b3 de OPTION).
Si PSA = 0 le prédiviseur est affecté à TMR0. Le choix du rapport de division se fait avec les bits PS2, PS1 et PS0 (b2, b1 et b0 de OPTION).
Si PSA = 1 le prédiviseur est affecté au Watchdog et le rapport de division pour TMR0 est fixé à 1.

INTERRUPTION : Elle est générée quand TMR0 passe de la valeur FF à 00.
Le Flag TOIF (b2 de INTCON) passe à "1". On peut masquer la génération de l'interruption en mettant le bit TOIE (b5 de INTCON) à "0".
Le Flag TOIF doit être remis à zéro par soft dans le sous programme d'interruption, avant de re-autoriser cette interruption.



PROGRAMMATION

Le PIC 16F84 peut être programmé en circuit. C'est le procédé de Microchip : ICSP = In Circuit Serial Programming.

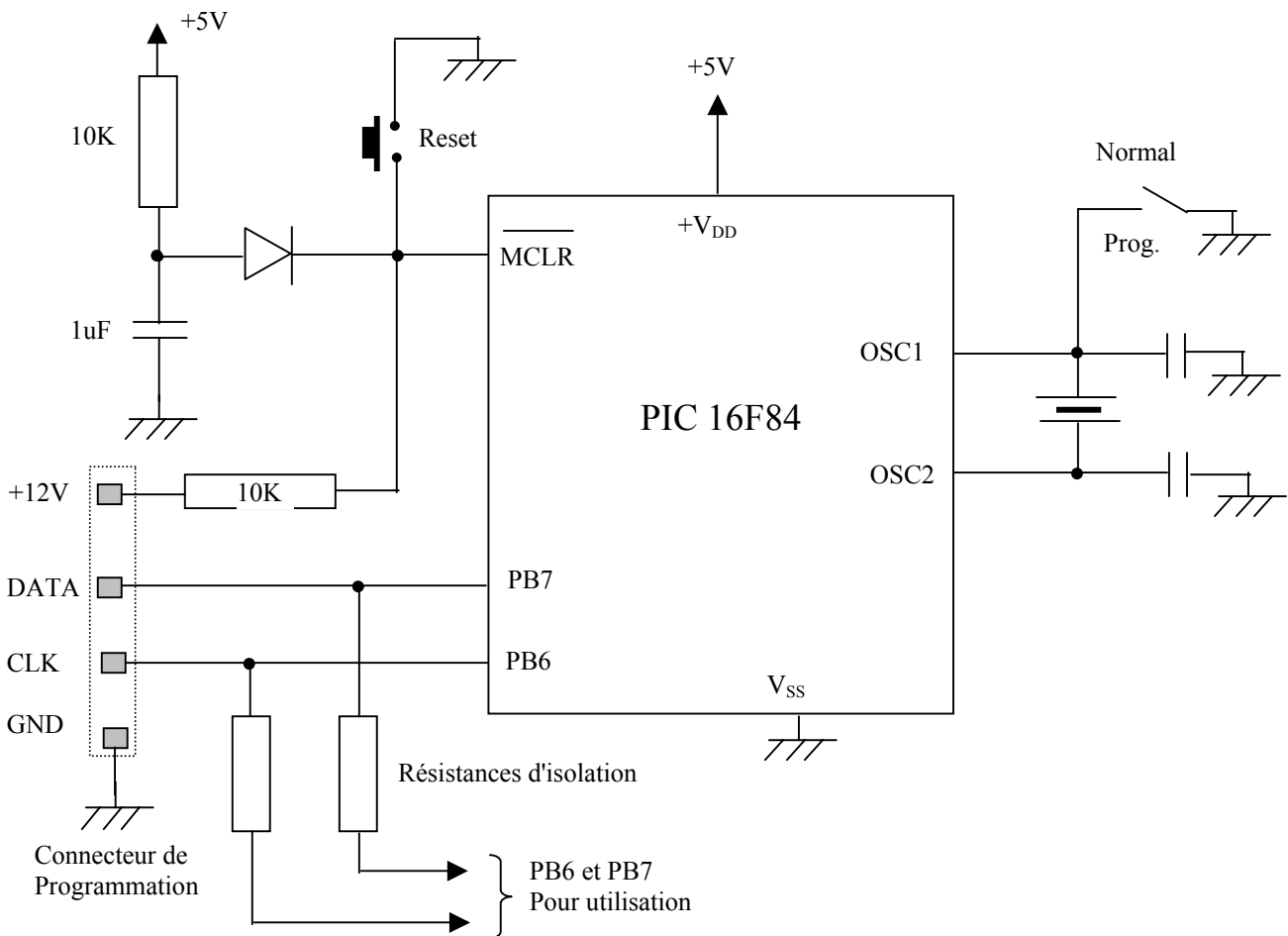
En mode normal la broche MCLR est tiré au $+V_{DD}$. Pour faire un reset du circuit, il suffit de la mettre à la masse.

Le PIC détecte l'entrée en mode programmation quand sa broche MCLR passe de 0 V à $V_{PP} = 12$ à 14 V avec les entrées PB₆ et PB₇ = "0".

Il faut que pendant ce temps l'oscillateur interne ne démarre pas. C'est le cas après un Power on Reset, car l'oscillateur attend 1024 cycles d'horloge : c'est le OST time out en mode XT LP ou HS (quartz). Mais pas en mode RC.

Si la montée de V_{PP} est trop lente, ou en mode RC, l'oscillateur risque de démarrer avant que la broche MCLR ne soit à 12 V. La programmation ne s'effectuera pas alors correctement.

Pour éviter ce problème, il suffit de bloquer l'oscillateur en mode programmation en mettant la pin 16 (OSC1) à la masse avant d'appliquer V_{PP} .



ASSEMBLEUR pour PIC

Notation en Hexadécimal :

h'4F'
ou : 04F
ou : 4F (Il faut un zéro devant le chiffre héxa s'il débute par une lettre).

Notation en Décimal :

d'28'

Notation en Binaire :

b'01010011'

Notation en Octal :

o'777'

Notation en ASCII :

'chaîne de caractères'
ou : A'chaîne de caractères'

Fichiers générés après assemblage : on part de : FILE.asm

FILE.lst = fichier listing.
FILE.hex = fichier objet en code machine à charger dans le PIC.
FILE.err = fichier des erreurs.

Directives assembleur :

	ORG	Adresse en héxa	→	Positionne adresse.
ETIQ	EQU	0F	→	Affecte la valeur 0F à ETIQ.
	END		→	Fin d'assemblage.
	;		→	Commentaires.

Ecriture de DATA en EEPROM Programme : DA

```

ORG     ADRESSE: mémoire programme (1 k octets de 0000 à 03FF)
ETIQ DA  "Texte"
  
```

Rangement en mémoire programme (mots de 14 bits) des caractères ASCII de la chaîne "texte" en 2 fois 7 bits (2 caractères de 7 bits à une même adresse). Les caractères ASCII sont toujours codés sur 7 bits (bit 7 =0).

Exemple:

```

          ORG     0010
ETIQ DA  "DANI"
  
```

Cela donne :

ADRESSE	DATA (14 bits)	Significations
0010	2241	1000100 1000001 └───┬───┘ └───┬───┘ "D"=44 "A"=41
0011	2749	1001110 1001001 └───┬───┘ └───┬───┘ "N"=4E "I"=49

Ecriture de DATA en EEPROM DATA : DE

```

ORG     ADRESSE mémoire Data ( 64 octets entre H2100 et H 213F)
ETIQ DE  "Texte"
  
```

Rangement en mémoire data (mots de 8 bits) des caractères ASCII de la chaîne "texte".

Exemple:

```

          ORG     2100
ETIQ DE  "DA"
  
```

Cela donne :

ADRESSE EEPROM DATA	DATA (8 bits)	Significations
00	44	01000100 └───┬───┘ "D"=44
01	41	01000001 └───┬───┘ "A"=41

DEFINE :

#DEFINE Nom Chaîne de caractères

Quand le Nom est rencontré par l'assembleur, il est remplacé par la chaîne de caractères ainsi définie.

Exemple: PORTB EQU 05
 #DEFINE LED PORTB,1

Chaque fois que l'assembleur rencontre LED il le remplace par: PORTB,1.

Autre exemple: STATUS EQU 03
 #DEFINE PAGE0 BCF STATUS,5
 #DEFINE PAGE1 BSF STATUS,5

Chaque fois que l'on écrira PAGE0 on se positionnera en Page 0, c'est à dire que le bit 5 du registre STATUS sera forcé à "0". De la même manière PAGE1 forcera le bit 5 de STATUS à "1" et positionnera l'adressage en page 1.

#INCLUDE :

#INCLUDE "Nom de fichier"

Exemple: #INCLUDE "MENES.INC"

Le fichier MENES.INC contient toutes les adresses des PORTS et registres du PIC 16F84, ainsi que les noms des bits particuliers de configuration de certains registres. Il peut contenir des directives assembleur telles que: MACRO, #DEFINE, etc ..

MACRO :

C'est une suite d'instructions que l'on pourra ensuite insérer dans le code source, en appelant simplement cette macro par son étiquette.

Exemple: PAGE1 MACRO
 BCF STATUS,6
 BSF STATUS,5
 ENDM

Cette macro positionne l'accès au registre F en page 1 par: RP1=0 et RP0=1.

MEMOIRE EEPROM FLASH

Elle est organisée en mots de 14 bits et contient :

- Le Programme de 0000 à 1FFF maximum.
- L'identification du programmeur : 4 adresses de 2000 à 2003.
- Le registre de configuration en 2007.
- La zone EEPROM DATA à partir de 2100 (mots de 8 bits seulement).

On ne peut y accéder qu'en mode programmation.

