

## Le microcontrôleur PIC16f84

### I. Définition :

Le pic est un microcontrôleur du fabricant Microchip. Les principales caractéristiques du microcontrôleur sont:

### Séparation des mémoires de programme et de données (architecture Harvard) :

La séparation des bus instructions et données permet d'avoir des instructions codées en un seul mot de 14 bits, tout en manipulant des données d'une largeur classique de 8 bits.

Communication avec l'extérieur seulement par des ports : il ne possède pas de bus d'adresses, de bus de données et de bus de contrôle comme la plupart des microprocesseurs.

Utilisation d'un jeu d'instructions réduit, d'où le nom de son architecture : RISC (Reduced Instructions Set Construction). Les instructions sont ainsi codées sur un nombre réduit de bits, ce qui accélère l'exécution : 1 cycle machine (ou temps de cycle) par instruction sauf pour les sauts qui requièrent 2 cycles. En revanche, leur nombre limité oblige à se restreindre à des instructions basiques, contrairement aux systèmes d'architecture CISC (Complex Instructions Set Construction) qui proposent plus d'instructions donc codées sur plus de bits mais réalisant des traitements plus complexes.

Le temps de cycle est égal à quatre fois la période d'horloge (période du quartz). Ainsi, si un PIC est piloté à partir d'un quartz de 10 MHz (période de 100 ns), le temps de cycle  $T_H$  sera de 400 ns. On pourra donc exécuter  $1/400 \times 10^{-9} = 2\,500\,000$  instructions par seconde (à l'exception des instructions de branchement), ce qui correspond à une puissance de calcul de 2,5 MIPS.

Grâce à cette architecture RISC, un programme nécessite environ deux fois moins d'espace mémoire et « tourne » environ deux fois plus vite que sur les systèmes à architecture classique.

Il existe trois familles de PIC :

- 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

Un PIC est identifié par un numéro de la forme suivant : xxXXyy-zz

- xx : Famille du composant (12, 14, 16, 17, 18)
- XX : Type de mémoire de programme
  - C - EPROM ou EEPROM
  - CR - PROM
  - F - FLASH
- yy : Identification
- zz : Vitesse maximum du quartz

Exemple : PIC 16F84 –10, soit :

- 16 : Mid-Line
- F : FLASH
- 84 : Type
- 10 : Quartz à 10MHz au maximum

## II. Le PIC 16F84

Il s'agit d'un microcontrôleur 8 bits à 18 pattes. Principales caractéristiques :

- 35 instructions
- Instructions codées sur 14 bits
- Données sur 8 bits
- 1 cycle machine par instruction, sauf pour les sauts (2 cycles machine)
- Vitesse maximum 10 MHz soit une instruction en 400 ns (1 cycle machine = 4 cycles d'horloge)
- 4 sources d'interruption
- 1000 cycles d'effacement/écriture pour la mémoire flash, 10.000.000 pour la mémoire de donnée EEPROM

### 1. Brochage et fonction des pattes

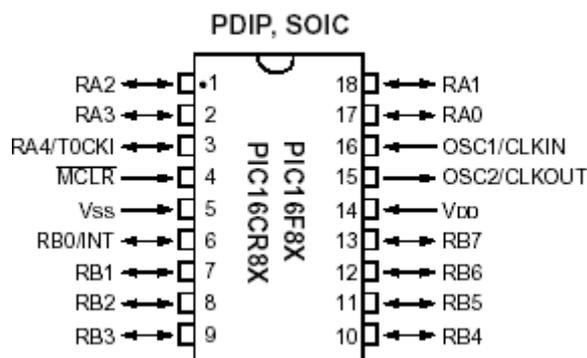
Les fonctions des pattes sont les suivantes :

- $V_{SS}$ ,  $V_{DD}$  : Alimentation
- OSC1,2 : Horloge
- RA0-4 : PORTA
- RB0-7 : PORTB
- T0CKI : Entrée de comptage
- INT : Entrée d'interruption
- MCLR : Reset : 0V

Choix du mode

programmation : 12V - 14V

exécution : 4.5V - 5.5V



Brochage du circuit.

## 2. Architecture générale

### Définition

**Bus** : un bus est un ensemble de lignes véhiculant une information sur n bits.

**data bus** : c'est le bus de données, celles-ci sont transmises sur un bus de n bits

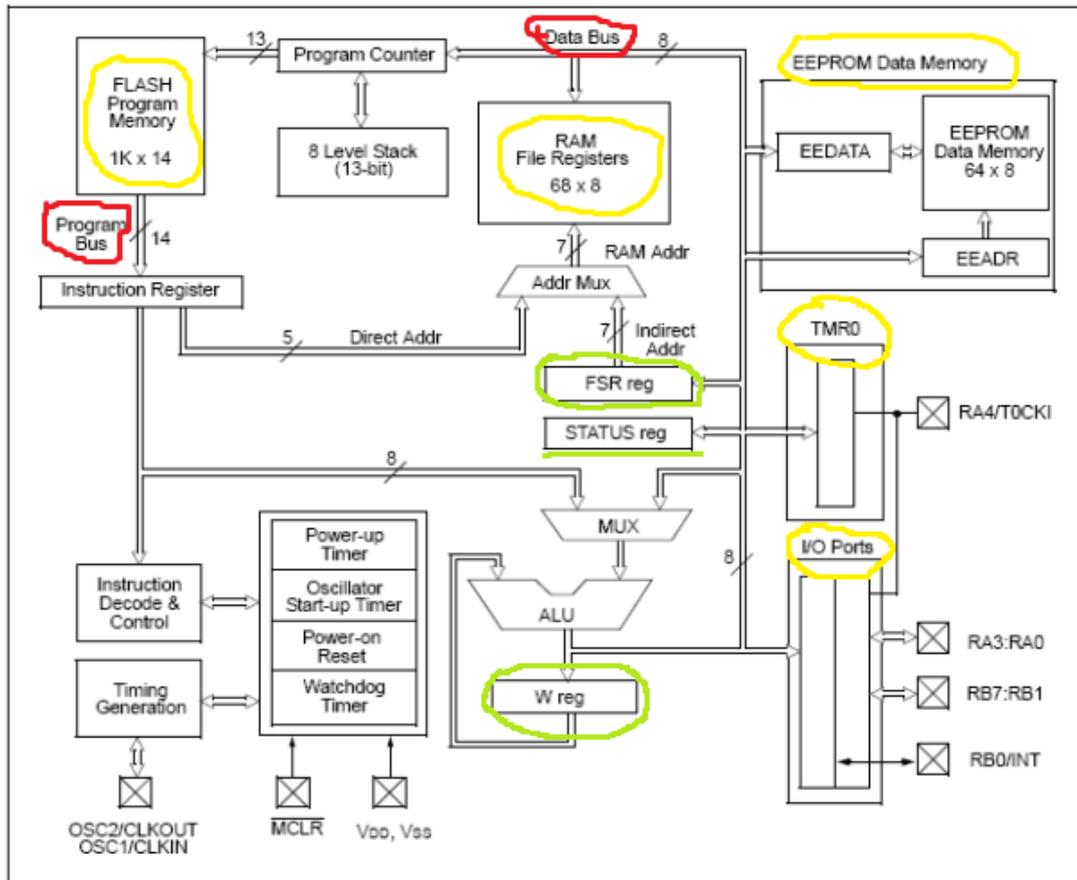
données sur 8 bits : n=8 lignes

**adresse bus** : c'est le bus d'adresses, les adresses sont transmises sur un bus de n bits

adresses sur 16 bits : n=16 lignes

L'architecture interne du pic16f84 est constituée des éléments suivants :

- un système d'initialisation à la mise sous tension (power-up timer, ...)
- un système de génération d'horloge à partir du quartz externe (timing génération)
- une unité arithmétique et logique (ALU)
- une mémoire flash de programme de 1k "mots" de 14 bits
- un compteur de programme (Program Counter : registre PC) et une pile (stack)
- un bus spécifique pour le programme (program bus) : pour les instructions
- un registre contenant le code de l'instruction à exécuter
- un bus spécifique pour les données (data bus)
  
- une mémoire RAM contenant :
  - les registres (SFR)
  - 68 octets de données : espace mémoire libre pour l'utilisateur
- une mémoire EEPROM de 64 octets de données
- 2 ports d'entrées/sorties : PORTA et PORTB
- un compteur (timer)
- un chien de garde (watchdog)



Architecture interne du PIC 16F84.

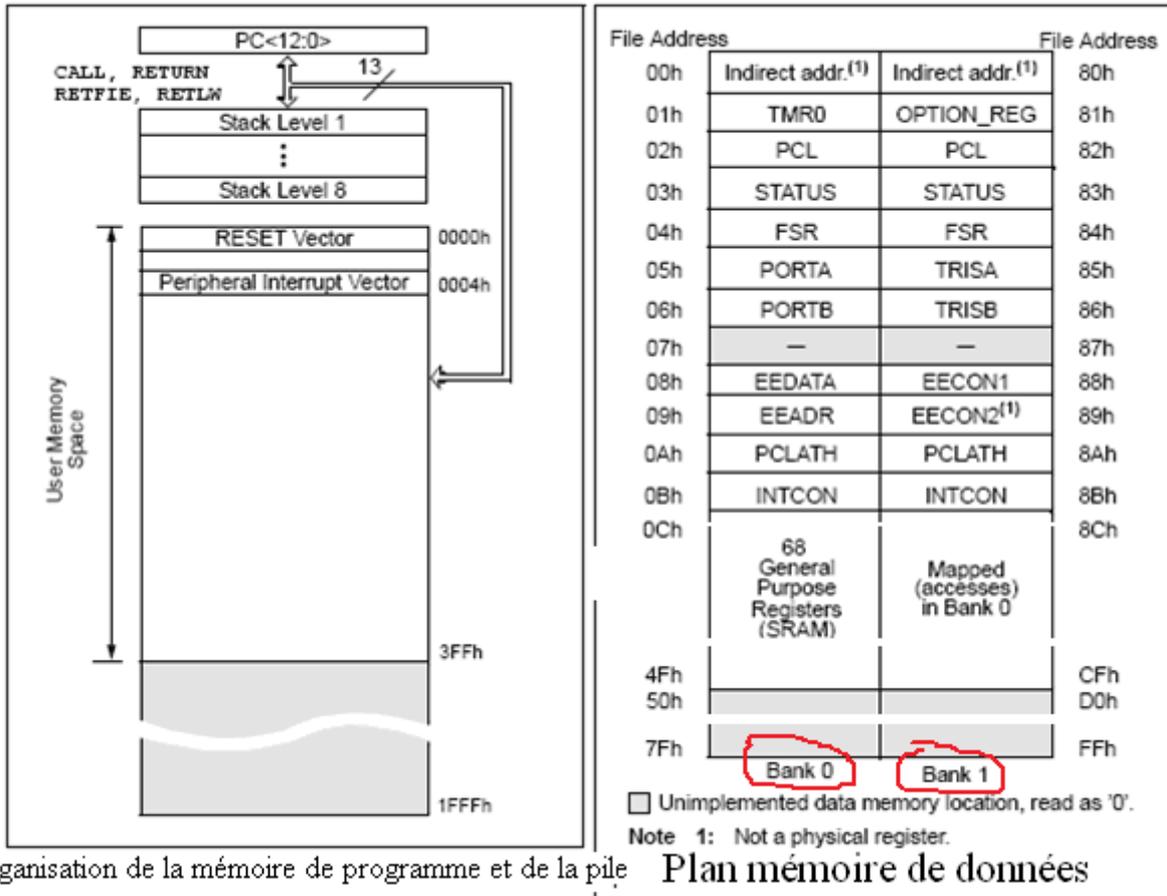
### III. Organisation de la mémoire

Le PIC contient de la mémoire de programme et de la mémoire de données. La structure Harvard des PICs fournit un accès séparé à chacune. Ainsi, un accès aux deux est possible pendant le même cycle machine.

#### 1. Mémoire de programme

Elle contient le programme à exécuter. Ce dernier est téléchargé par liaison série. C'est une mémoire flash de 1k "mots" de 14 bits dans le cas du PIC 16F84, même si le compteur de programme (PC) de 13 bits peut en adresser 8k. L'adresse 0000h contient le vecteur du reset, l'adresse 0004h est l'unique vecteur d'interruption du PIC.

La pile contient 8 valeurs. Elle n'a pas d'adresse dans la plage de mémoire. C'est une zone réservée par le système.



Organisation de la mémoire de programme et de la pile Plan mémoire de données

## 2. Mémoire de données

Elle se décompose en deux parties de RAM et une zone EEPROM.

La première partie de RAM contient les SFRs (Special Function Registers) qui permettent de contrôler les opérations sur le circuit (le pic). La seconde partie de RAM contient des registres généraux (68 octets), libres pour l'utilisateur. La zone EEPROM contient 64 octets.

Les instructions du pic orientées octets ou bits contiennent une adresse sur 7 bits pour désigner l'octet avec lequel l'instruction doit travailler. D'après le plan mémoire de données, l'accès au registre TRISA d'adresse 85h, par exemple, est impossible avec une adresse sur 7 bits. C'est pourquoi le constructeur a défini deux banques : Bank0 et Bank1

Le bit RP0 du registre d'état (STATUS, bit5) permet de choisir entre les deux banks. Ainsi, une adresse sur 8 bits est composée de RP0 en poids fort et des 7 bits provenant de l'instruction à exécuter.

### 2.1. Registres généraux

Il s'agit de l'espace de la mémoire RAM d'adresse 0x0C et qui est accessible soit **directement** soit **indirectement** à travers les registres FSR et INDF. C'est l'espace mémoire réservé à l'utilisateur. Cet espace va de l'adresse **0x0C** à l'adresse **0x4F** (68 octets).

### 2.2. Registres spéciaux - SFRs

Ils permettent la gestion du circuit. Certains ont une fonction générale, d'autre une fonction spécifique attachée à un périphérique donné. Ils sont situés de l'adresse 00h à l'adresse 0Bh dans la bank0 et de l'adresse 80h à l'adresse 8Bh dans la bank1 (voir figure : Plan mémoire). Les registres 07h et 87h n'existent pas.

Addr	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on Power-on RESET	Details on page
<b>Bank 0</b>											
00h	INDF	Uses contents of FSR to address Data Memory (not a physical register)								---- ----	11
01h	TMR0	8-bit Real-Time Clock/Counter								xxxx xxxx	20
02h	PCL	Low Order 8 bits of the Program Counter (PC)								0000 0000	11
03h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
04h	FSR	Indirect Data Memory Address Pointer 0								xxxx xxxx	11
05h	PORTA <sup>(4)</sup>	—	—	—	RA4/T0CKI	RA3	RA2	RA1	RA0	---x xxxx	16
06h	PORTB <sup>(5)</sup>	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT	xxxx xxxx	18
07h	—	Unimplemented location, read as '0'								—	—
08h	EEDATA	EEPROM Data Register								xxxx xxxx	13,14
09h	EEADR	EEPROM Address Register								xxxx xxxx	13,14
0Ah	PCLATH	—	—	—	Write Buffer for upper 5 bits of the PC <sup>(1)</sup>				---0 0000	11	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	10
<b>Bank 1</b>											
80h	INDF	Uses Contents of FSR to address Data Memory (not a physical register)								---- ----	11
81h	OPTION_REG	RBPU	INTEG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	9
82h	PCL	Low order 8 bits of Program Counter (PC)								0000 0000	11
83h	STATUS <sup>(2)</sup>	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C	0001 1xxx	8
84h	FSR	Indirect data memory address pointer 0								xxxx xxxx	11
85h	TRISA	—	—	—	PORTA Data Direction Register				---1 1111	16	
86h	TRISB	PORTB Data Direction Register								1111 1111	18
87h	—	Unimplemented location, read as '0'								—	—
88h	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD	---0 x000	13
89h	EECON2	EEPROM Control Register 2 (not a physical register)								---- ----	14
0Ah	PCLATH	—	—	—	Write buffer for upper 5 bits of the PC <sup>(1)</sup>				---0 0000	11	
0Bh	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	10

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

- Note 1:** The upper byte of the program counter is not directly accessible. PCLATH is a slave register for PC<12:8>. The contents of PCLATH can be transferred to the upper byte of the program counter, but the contents of PC<12:8> are never transferred to PCLATH.
- 2:** The  $\overline{TO}$  and  $\overline{PD}$  status bits in the STATUS register are not affected by a  $\overline{MCLR}$  Reset.
- 3:** Other (non power-up) RESETS include: external RESET through MCLR and the Watchdog Timer Reset.
- 4:** On any device RESET, these pins are configured as inputs.
- 5:** This is the value that will be in the port output latch.

## Description des SFR.

**INDF** (00h - 80h) : Utilise le contenu de FSR pour l'accès indirect à la mémoire.

**TMR0** (01h) : Registre lié au compteur.

**PCL** (02h - 82h) : Contient les poids faibles du compteur de programmes (PC). Le registre **PCLATH** (0Ah-8Ah) contient les poids forts.

**STATUS** (03h - 83h) : Il contient l'état de l'unité arithmétique et logique ainsi que les bits de sélection des banks.

**FSR** (04h - 84h) : Permet l'adressage indirect

**PORTA** (05h) : Donne accès en lecture ou écriture au port A de 5 bits : RA0 à RA4 ; la ligne **RA4** peut être utilisé en entrée de comptage.

**PORTB** (06h) : Donne accès en lecture ou écriture au port B : 8 bits ; la ligne **RB0** peut être utiliser en entrée d'interruption.

**EEDATA** (08h) : Permet l'accès aux données dans la mémoire EEPROM.

**EEADR** (09h) : Permet l'accès aux adresses de la mémoire EEPROM.

**PCLATCH** (0Ah - 8Ah) : Donne accès en écriture aux bits de poids forts du compteur de programme.

**INTCON** (0Bh - 8Bh) : Masque d'interruptions.

**OPTION\_REG** (81h) : Contient des bits de configuration pour divers périphériques.

**TRISA** (85h) : Indique la direction (entrée ou sortie) du port A.

**TRISB** (86h) : Indique la direction (entrée ou sortie) du port B.

**EECON1** (88h) : Permet le contrôle d'accès à la mémoire EEPROM.

**EECON2** (89h) : Permet le contrôle d'accès à la mémoire EEPROM.

### **2.3. Mémoire EEPROM**

Le PIC 16f84 possède une zone EEPROM de 64 octets accessibles en lecture et en écriture par le programme. **On peut y sauvegarder des valeurs, qui seront conservées même si l'alimentation est éteinte, et les récupérer lors de la mise sous tension.** Leur accès est spécifique et requiert l'utilisation de registres dédiés. La lecture et l'écriture ne peut s'exécuter que selon des séquences particulières.

#### **Remarques**

**Vecteur reset** : L'adresse 0x000h de la mémoire flash s'appelle vecteur reset car lorsqu'on appuie sur le bouton reset, le pic démarre instantanément l'exécution du programme à partir de l'adresse 0x000h de la mémoire flash.

**Vecteur d'interruption** : L'adresse 0x004h de la mémoire flash s'appelle vecteur d'interruption, car lorsqu'une interruption est générée le pic abandonne l'exécution du programme principale qu'il était en train d'exécuter et démarre l'exécution du sous-programme d'interruption à partir de l'adresse 0x004h.