

SOMMAIRE

1) INTRODUCTION.....	4
2) CARACTÉRISTIQUES DES MICROCONTROLEURS.....	5
2) ORGANISATIONS INTERNES.....	6
2.1) LES μ CS 16F873 ET 16F876.....	6
2.2) LES μ CS 16F874 ET 16F877.....	7
3) DESCRIPTION DES DIFFERENTES BROCHES.....	8
3.1) LES μ CS 16F873 ET 16F876.....	8
3.2) LES μ CS 16F874 ET 16F877.....	9
4) BROCHAGES PHYSIQUES DES DIFFERENTES VERSIONS DE μCS.....	10
5) LES BROCHES DU μC.....	11
5.1) MCLR.....	11
5.2) OSCILLATEUR : OSC1 ET OSC2 OU CLKIN ET CLOUT.....	12
5.3) ALIMENTATION : VDD ET VSS.....	13
5.4) L'INTERRUPTION : RBO/INT.....	13
6) L'UNITÉ CENTRALE.....	14
6.1) ORGANISATION MEMOIRE.....	14
6.1.1) <i>Plan Mémoire pour les instructions (code programme)</i>	14
6.1.2) <i>Plan Mémoire pour les données et registres internes (SFR : Special Function Register)</i>	15
6.1.2) <i>L'UAL : L'unité Arithmétique et Logique</i>	17
7) JEU D'INSTRUCTIONS.....	18
8) LES MODES D'ADRESSAGES.....	19
8.1) ADRESSAGE INHERENT OU IMPLICITE.....	19
8.2) ADRESSAGE IMMEDIAT.....	19
8.3) ADRESSAGES DIRECT ET ETENDU.....	19
8.4) ADRESSAGE RELATIF.....	20
8.5) ADRESSAGE INDIRECT OU ENCORE INDEXE.....	20
8.6) MANIPULATION DE BITS.....	21
8.6.1) <i>Forçage de bits</i>	21
8.6.2) <i>Test de bits</i>	21
9) LES PORTS D'ENTREES SORTIES.....	22
9.1) GENERALITES.....	22
9.2) CONFIGURATION DES PORTX , LES REGISTRES PORTX ET TRISX.....	23
9.3) LE PORT A.....	24
9.3) LE PORT B.....	25
9.4) LE PORT C.....	26
9.5) LES PORT D ET E.....	27
10) LE CONVERTISSEUR ANALOGIQUE NUMERIQUE.....	28
10.1) ORGANISATION INTERNE.....	28
10.2) FONCTIONNEMENT DU CONVERTISSEUR.....	29
10.3) LE REGISTRE ADCON1.....	30
10.4) LE REGISTRE ADCON0.....	31
10.5) EXEMPLE D'UTILISATION.....	32

11) LES TIMERS.....	33
11.1) LE TIMER 0.....	33
11.1.1) <i>Présentation</i> :.....	33
11.1.2) <i>Fonctionnement</i> :.....	34
11.1.3) <i>Configuration et registres associés</i> :.....	34
11.2) LE TIMER 1.....	35
11.2.1) <i>Présentation</i> :.....	35
11.2.2) <i>Fonctionnement</i> :.....	35
11.2.3) <i>Configuration et registres associés</i> :.....	36
11.3) LE TIMER 2.....	37
11.3.1) <i>Présentation</i> :.....	37
11.3.2) <i>Fonctionnement</i> :.....	37
11.3.3) <i>Configuration et registres associés</i> :.....	38
11.4) LES MODULES CCP1 ET CCP2 (C.C.P. :CAPTURE COMPARE PWM).....	39
11.4.1) <i>Le mode CAPTURE</i> :.....	39
12) LA LIAISON SERIE USART OU SCI (SERIAL COMMUNICATION INTERFACE).....	40
12.1) PRESENTATION :.....	40
12.2) LA TRANSMISSION :.....	41
12.2.1) <i>Présentation</i> :.....	41
12.2.2) <i>Fonctionnement</i> :.....	41
12.2.3) <i>Configuration et registres associés</i> :.....	42
12.3) LA RECEPTION :.....	43
12.3.1) <i>Présentation</i> :.....	43
12.3.2) <i>Fonctionnement</i> :.....	43
12.3.3) <i>Configuration et registres associés</i> :.....	44
12.4) LE GENERATEUR D'HORLOGE:.....	45
12.4.1) <i>Présentation et fonctionnement</i> :.....	45
12.4.2) <i>Configuration et registres associés</i> :.....	46
13) LES INTERRUPTIONS.....	47
13.1) PRESENTATION.....	47
13.2) FONCTIONNEMENT.....	47
13.3) DEROULEMENT D'UNE INTERRUPTION.....	48
13.4) CONFIGURATION ET REGISTRES ASSOCIES :.....	49

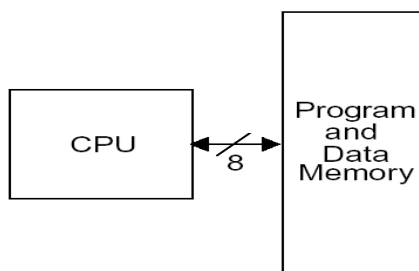
LES MICROCONTROLEURS PIC 16F87X

1) INTRODUCTION.

Ces microcontrôleurs méritent bien une introduction. En effet, ils ont été conçus sur une architecture dite **HARVARD (RISC)** et non sur un modèle **VON NEUMANN (COMPLEX)**.

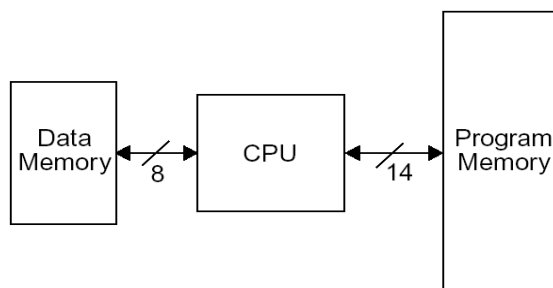
- L'architecture **VON NEUMANN** employée par la plupart des microcontrôleurs actuels (**INTEL80XX, motorola HC05, HC08 et HC11, ou ZILOG Z80**) est basée sur un bus de données unique. Celui-ci véhicule les **instructions** et les **données**.

von-Neumann



- L'architecture **HARVARD** utilisée par les microcontrôleurs **PICs** est basée sur deux bus de données. Un bus est utilisé pour les **données** et un autre pour les **instructions**.

Harvard



Avantages et inconvénients.

	Architecture VON NEUMANN (MOTOROLA, INTEL, ZILOG, ..)	Architecture HARVARD (RISC) (MICROCHIP PICs)
Avantages	<ul style="list-style-type: none"> - Jeu d'instructions riches. - Accès à la mémoire facile. 	<ul style="list-style-type: none"> - Jeu d'instructions pauvre, mais facile à mémoriser. - Le codage des instructions est facile, chaque instruction est codée sur un mot et dure un cycle machine. - Le code est plus compact.
Inconvénients	<ul style="list-style-type: none"> - Le temps pour exécuter une instruction est variable. - Le codage des instructions se fait sur plusieurs octets. 	<ul style="list-style-type: none"> - Le jeu d'instruction est très pauvre, par exemple pour effectuer une comparaison il faut faire une soustraction. - Les accès aux registres internes et la mémoire sont très délicats.

Remarque : La taille mémoire spécifiée pour un **PICs** s'exprime en **Kilo Mots (14 bits)** pour la famille **16F87X** et non en **kilo octets**. Comme chaque instruction est codée par un mot de **14 bits**, comparées aux microcontrôleurs classiques (**1,2 ou 3 octets** par instruction), les **PICs** ont un code plus compact et ils utilisent moins de mémoire.

2) CARACTÉRISTIQUES DES MICROCONTROLEURS.

Ces microcontrôleurs appartiennent à la famille des **PICs**. Le **16** signifie qu'ils font partie de la famille des **16F** de **MICROCHIP** et le **PIC16F876** est une version **28** broches alors que le **16F877** est une version **40** broches.

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

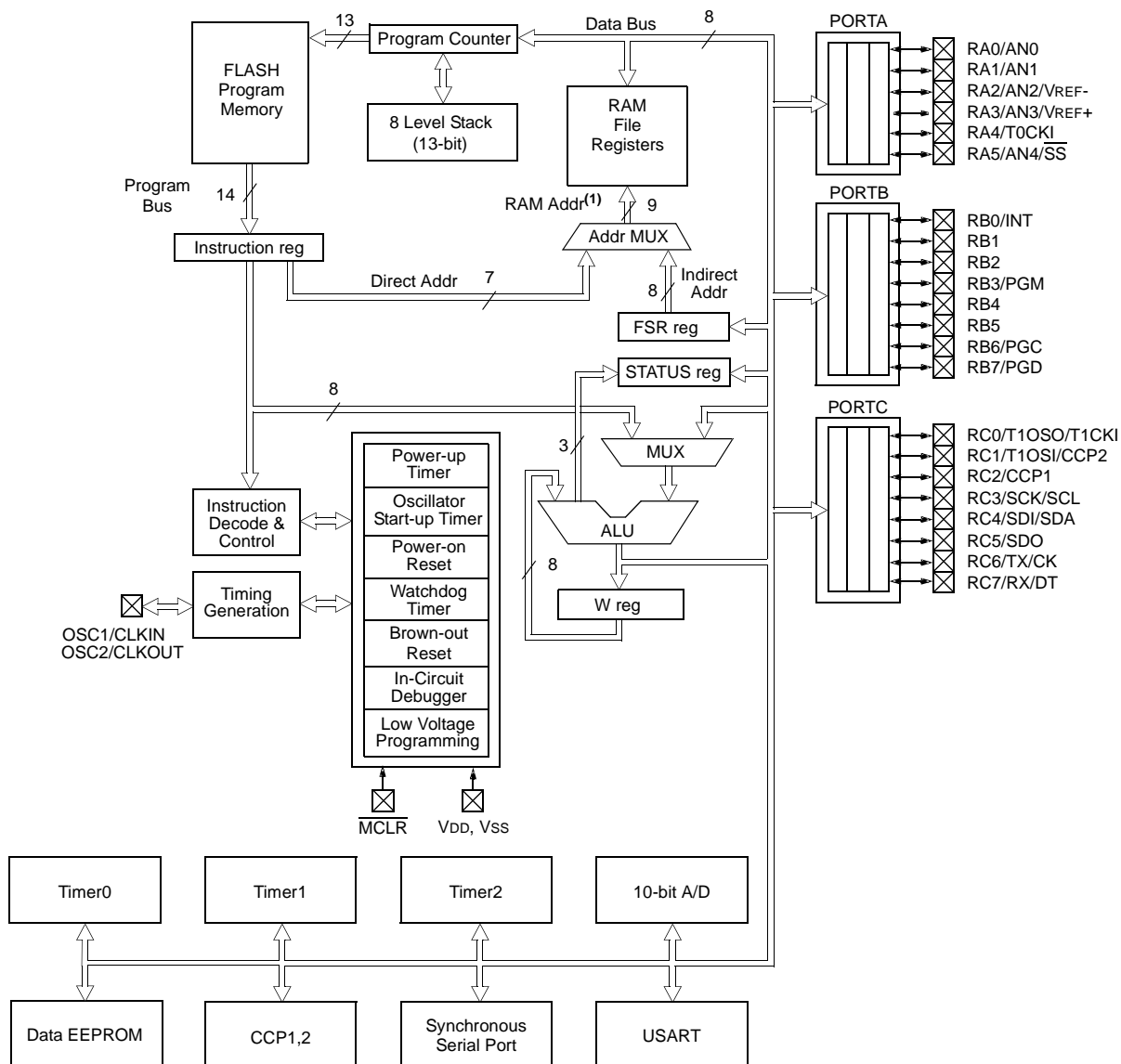
Caractéristiques communes :

- Programmable sur site **ICSP (In Circuit Serial Programming)**.
- Débuggable sur site **ICD**.
- Fréquence de fonctionnement élevée, jusqu'à **20 Mhz**.
- Une mémoire vive de **192 à 368** octets.
- Une mémoire **EEPROM** pour sauver des paramètres de **128 à 256** octets.
- Une mémoire morte de type **FLASH** de **4 Kmots à 8 Kmots (1mot = 14 bits)**.
- Chien de garde **WDT**.
- Surveillance d'horloge **OST**.
- Surveillance de tension d'alimentation **BOR**.
- De **21 à 32** Entrées et sorties suivant le type de micro contrôleur.
- Chaque sortie peut sortir un courant maximum de **25mA**.
- **3** Temporisateurs : **TIMER0** (8 bits avec pré diviseur), **TIMER1** (16 bits avec pré diviseur avec possibilité d'utiliser une horloge externe réseau **RC** ou **QUARTZ**) et **TIMER2** (8 bits avec pré diviseur et post diviseur)
- **2** entrées de captures et de comparaison avec **PWM** (Modulation de largeur d'impulsions).
- Convertisseur analogique numérique **10** bits avec de **5 à 8** entrées multiplexées maximum.
- Une interface de communication série asynchrone et synchrone. (**USART/SCI**).
- Une interface de communication série synchrone. (**SSP/SPI et I2C**).
- Plusieurs modes de fonctionnements faible consommation.
- Une seule tension d'alimentation **2** ou **5V**.
- Conservation des informations en mémoire vive jusqu'à **1.5V**.
- Faible consommation :
 - **<2mA** à **5V** pour **Fquartz=4Mhz**.
 - **20µA** à **3V** pour **Fquartz à 32Khz**.
 - **<1µA** pour en mode sommeil.

2) ORGANISATIONS INTERNES

2.1) Les μ Cs 16F873 et 16F876.

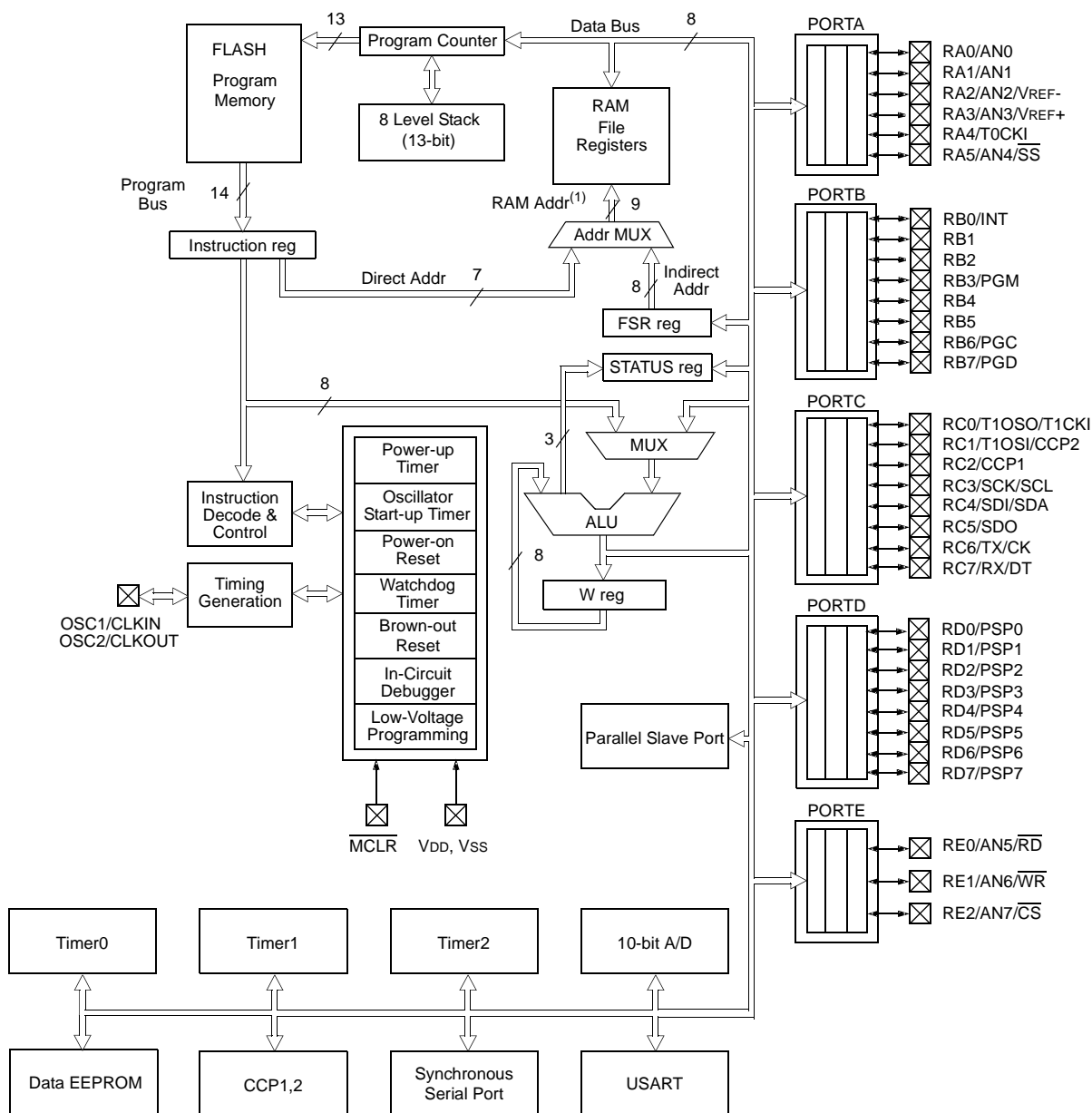
Device	Program FLASH	Data Memory	Data EEPROM
PIC16F873	4K	192 Bytes	128 Bytes
PIC16F876	8K	368 Bytes	256 Bytes



LES MICROCONTROLEURS PIC 16F87X

2.2) Les µCs 16F874 et 16F877.

Device	Program FLASH	Data Memory	Data EEPROM
PIC16F874	4K	192 Bytes	128 Bytes
PIC16F877	8K	368 Bytes	256 Bytes



Remarque : Les 16F873 et 16F874 présentent peu d'intérêt par rapport aux 16F876 et 16F877, en effet ils possèdent moins de mémoires programmes et ils sont à peu près au même prix.

LES MICROCONTROLEURS PIC 16F87X

3) DESCRIPTION DES DIFFERENTES BROCHES

3.1) Les μ Cs 16F873 et 16F876.

Pin Name	DIP Pin#	SOIC Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	9	9	I	ST/CMOS ⁽³⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	10	10	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, the OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	1	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	2	I/O	TTL	PORTA is a bi-directional I/O port. RA0 can also be analog input0. RA1 can also be analog input1. RA2 can also be analog input2 or negative analog reference voltage. RA3 can also be analog input3 or positive analog reference voltage. RA4 can also be the clock input to the Timer0 module. Output is open drain type. RA5 can also be analog input4 or the slave select for the synchronous serial port.
RA1/AN1	3	3	I/O	TTL	
RA2/AN2/VREF-	4	4	I/O	TTL	
RA3/AN3/VREF+	5	5	I/O	TTL	
RA4/T0CKI	6	6	I/O	ST	
RA5/SS/AN4	7	7	I/O	TTL	
RB0/INT	21	21	I/O	TTL/ST ⁽¹⁾	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs. RB0 can also be the external interrupt pin. RB3 can also be the low voltage programming input. Interrupt-on-change pin. Interrupt-on-change pin. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock. Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.
RB1	22	22	I/O	TTL	
RB2	23	23	I/O	TTL	
RB3/PGM	24	24	I/O	TTL	
RB4	25	25	I/O	TTL	
RB5	26	26	I/O	TTL	
RB6/PGC	27	27	I/O	TTL/ST ⁽²⁾	
RB7/PGD	28	28	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	11	11	I/O	ST	PORTC is a bi-directional I/O port. RC0 can also be the Timer1 oscillator output or Timer1 clock input. RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output. RC2 can also be the Capture1 input/Compare1 output/PWM1 output. RC3 can also be the synchronous serial clock input/output for both SPI and I ² C modes. RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode). RC5 can also be the SPI Data Out (SPI mode). RC6 can also be the USART Asynchronous Transmit or Synchronous Clock. RC7 can also be the USART Asynchronous Receive or Synchronous Data.
RC1/T1OSI/CCP2	12	12	I/O	ST	
RC2/CCP1	13	13	I/O	ST	
RC3/SCK/SCL	14	14	I/O	ST	
RC4/SDI/SDA	15	15	I/O	ST	
RC5/SDO	16	16	I/O	ST	
RC6/TX/CK	17	17	I/O	ST	
RC7/RX/DT	18	18	I/O	ST	
Vss	8, 19	8, 19	P	—	Ground reference for logic and I/O pins.
VDD	20	20	P	—	Positive supply for logic and I/O pins.

Legend: I = input O = output I/O = input/output P = power
 — = Not used TTL = TTL input ST = Schmitt Trigger input

- Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
 3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

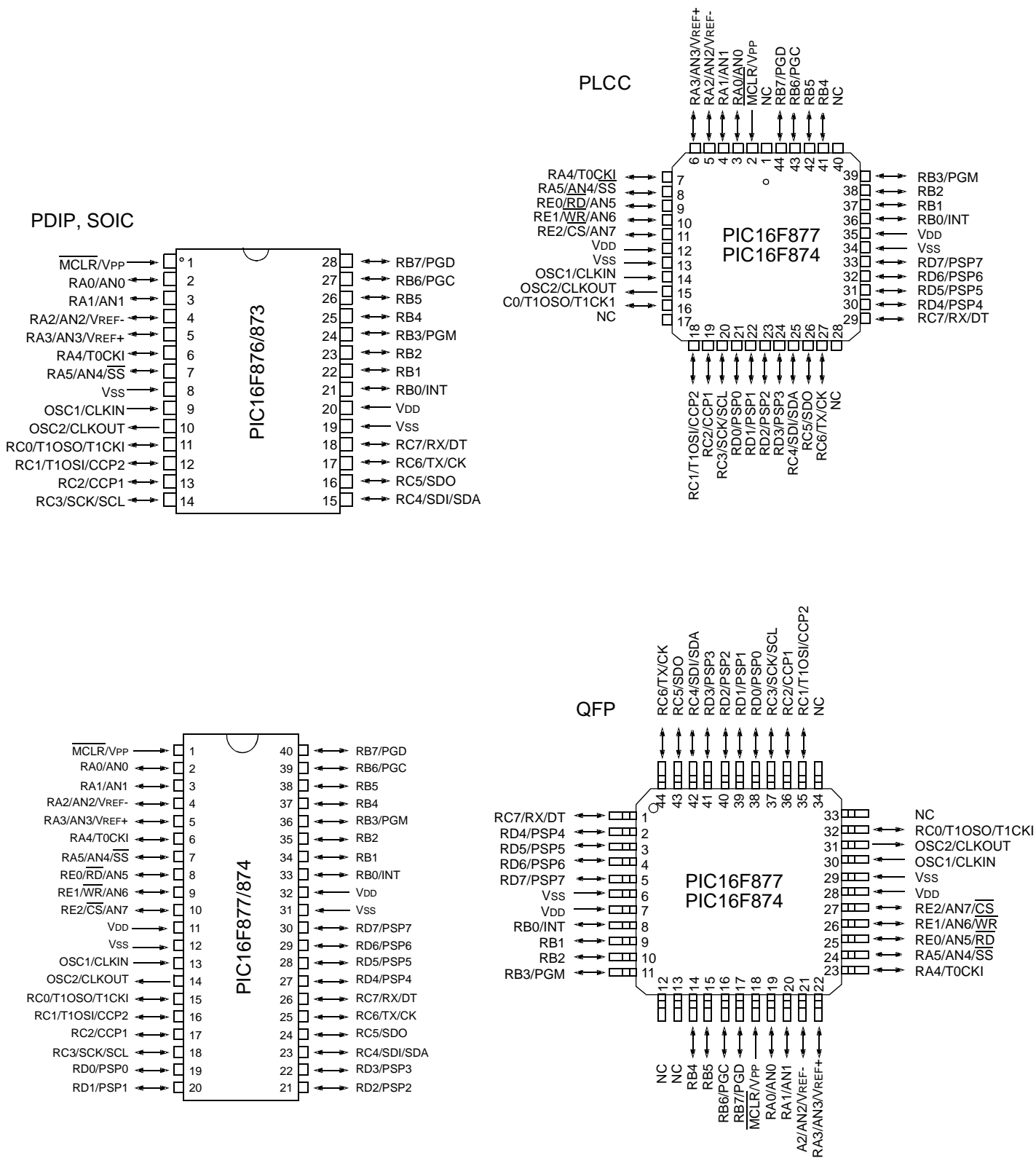
LES MICROCONTROLEURS PIC 16F87X

3.2) Les μ Cs 16F874 et 16F877.

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS ⁽⁴⁾	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
RA0/AN0	2	3	19	I/O	TTL	<p>PORTA is a bi-directional I/O port.</p> <p>RA0 can also be analog input0.</p> <p>RA1 can also be analog input1.</p> <p>RA2 can also be analog input2 or negative analog reference voltage.</p> <p>RA3 can also be analog input3 or positive analog reference voltage.</p> <p>RA4 can also be the clock input to the Timer0 timer/counter. Output is open drain type.</p> <p>RA5 can also be analog input4 or the slave select for the synchronous serial port.</p>
RA1/AN1	3	4	20	I/O	TTL	
RA2/AN2/VREF-	4	5	21	I/O	TTL	
RA3/AN3/VREF+	5	6	22	I/O	TTL	
RA4/T0CKI	6	7	23	I/O	ST	
RA5/SS/AN4	7	8	24	I/O	TTL	
RB0/INT	33	36	8	I/O	TTL/ST ⁽¹⁾	<p>PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.</p> <p>RB0 can also be the external interrupt pin.</p> <p>RB3 can also be the low voltage programming input.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.</p> <p>Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.</p>
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	
RB4	37	41	14	I/O	TTL	
RB5	38	42	15	I/O	TTL	
RB6/PGC	39	43	16	I/O	TTL/ST ⁽²⁾	
RB7/PGD	40	44	17	I/O	TTL/ST ⁽²⁾	
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	<p>PORTC is a bi-directional I/O port.</p> <p>RC0 can also be the Timer1 oscillator output or a Timer1 clock input.</p> <p>RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.</p> <p>RC2 can also be the Capture1 input/Compare1 output/PWM1 output.</p> <p>RC3 can also be the synchronous serial clock input/output for both SPI and I²C modes.</p> <p>RC4 can also be the SPI Data In (SPI mode) or data I/O (I²C mode).</p> <p>RC5 can also be the SPI Data Out (SPI mode).</p> <p>RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.</p> <p>RC7 can also be the USART Asynchronous Receive or Synchronous Data.</p>
RC1/T1OSI/CCP2	16	18	35	I/O	ST	
RC2/CCP1	17	19	36	I/O	ST	
RC3/SCK/SCL	18	20	37	I/O	ST	
RC4/SDI/SDA	23	25	42	I/O	ST	
RC5/SDO	24	26	43	I/O	ST	
RC6/TX/CK	25	27	44	I/O	ST	
RC7/RX/DT	26	29	1	I/O	ST	
RD0/PSP0	19	21	38	I/O	ST/TTL ⁽³⁾	<p>PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.</p>
RD1/PSP1	20	22	39	I/O	ST/TTL ⁽³⁾	
RD2/PSP2	21	23	40	I/O	ST/TTL ⁽³⁾	
RD3/PSP3	22	24	41	I/O	ST/TTL ⁽³⁾	
RD4/PSP4	27	30	2	I/O	ST/TTL ⁽³⁾	
RD5/PSP5	28	31	3	I/O	ST/TTL ⁽³⁾	
RD6/PSP6	29	32	4	I/O	ST/TTL ⁽³⁾	
RD7/PSP7	30	33	5	I/O	ST/TTL ⁽³⁾	
RE0/RD/AN5	8	9	25	I/O	ST/TTL ⁽³⁾	<p>PORTE is a bi-directional I/O port.</p> <p>RE0 can also be read control for the parallel slave port, or analog input5.</p> <p>RE1 can also be write control for the parallel slave port, or analog input6.</p> <p>RE2 can also be select control for the parallel slave port, or analog input7.</p>
RE1/WR/AN6	9	10	26	I/O	ST/TTL ⁽³⁾	
RE2/CS/AN7	10	11	27	I/O	ST/TTL ⁽³⁾	
Vss	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28,40	12,13,33,34		—	These pins are not internally connected. These pins should be left unconnected.

LES MICROCONTROLEURS PIC 16F87X

4) BROCHAGES PHYSIQUES DES DIFFERENTES VERSIONS DE μ CS.



5) LES BROCHES DU μC .

5.1) MCLR.

Cette broche sert à initialiser le μC .

Le μC dispose de plusieurs sources de **RESET** :

- **POR.**
- **EXTERNAL RESET.**
- **WDT.**
- **BOR.**

- **POR: (POWER ON RESET) Mise sous tension.**

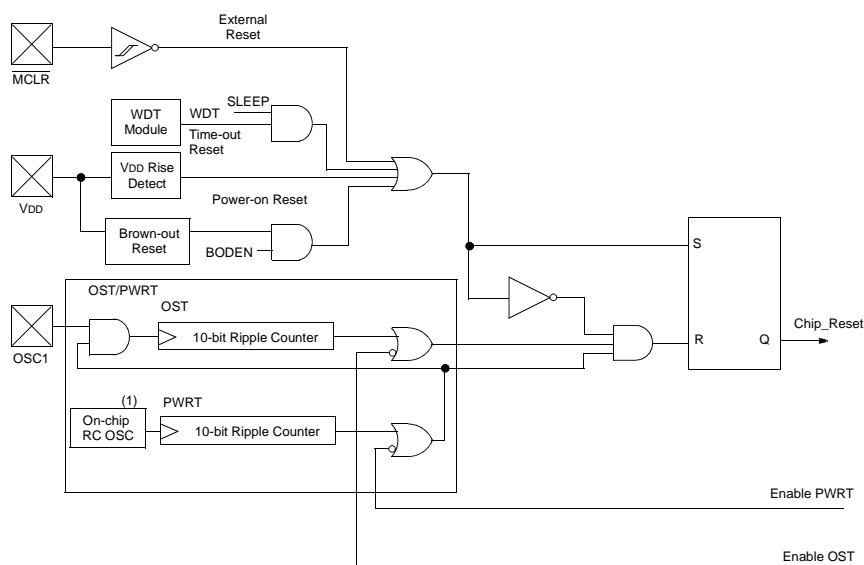
Un front montant sur **MCLR** déclenche l'initialisation du μC . Le temps nécessaire est au minimum de **72mS** et au maximum de **72mS+1024*Tosc**. Le μC dispose en interne d'un circuit de détection de niveau, quand la tension **VDD** est comprise entre **1.2V** et **1.7V**, il démarre une procédure d'initialisation.

Cette broche peut être simplement reliée à **VDD** si on n'a pas besoin de **RESET** externe. Par contre si on souhaite implanter un bouton de remise à zéro, on pourra câbler un simple réseau **RC** sur la broche **MCLR**.

Remarque importante : On peut se passer de circuit **RC** à la seule condition que le temps de montée de **VDD** soit suffisamment rapide (au minimum **50mV/ms**). Si le temps de montée est inférieur à **50mV/ms**, il faut rajouter un réseau **RC**.

- **EXTERNAL RESET (Mise à l'état bas de MCLR).** Remise à zéro extérieure. Il faut appliquer un niveau bas sur l'entrée **RESET** pendant au moins **2 μS** pour que l'initialisation soit prise en compte.
- **WDT:** Chien de garde.
Si le **WDT** arrive à la fin du temps de garde sans avoir été rafraîchi il y aura alors une initialisation du μC .
- **BOR: Baisse de l'alimentation.**
Si la tension **VDD** chute en dessous de **4V** pendant **100 μS** au moins, le microcontrôleur peut générer un **RESET**.

Schéma structurel du câblage de la broche MCLR.



5.2) Oscillateur : OSC1 et OSC2 ou CLKIN et CLOUT.

Ces broches permettent de faire fonctionner l'oscillateur interne du **PIC**.

On peut utiliser **3** types d'oscillateurs :

- Un quartz ou résonateur céramique.

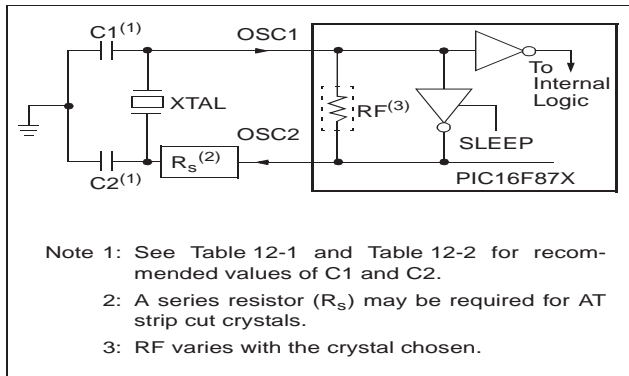


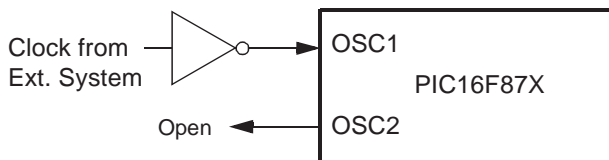
TABLE 12-1: CERAMIC RESONATORS

Ranges Tested:			
Mode	Freq.	OSC1	OSC2
XT	455 kHz	68 - 100 pF	68 - 100 pF
	2.0 MHz	15 - 68 pF	15 - 68 pF
	4.0 MHz	15 - 68 pF	15 - 68 pF
HS	8.0 MHz	10 - 68 pF	10 - 68 pF
	16.0 MHz	10 - 22 pF	10 - 22 pF

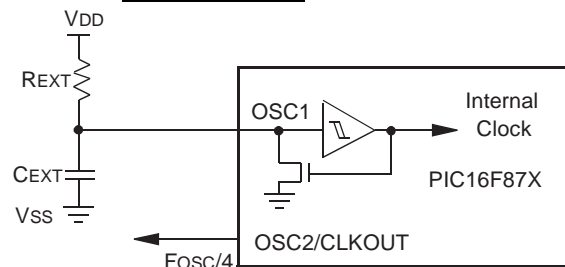
TABLE 12-2: CAPACITOR SELECTION FOR CRYSTAL OSCILLATOR

Osc Type	Crystal Freq.	Cap. Range C1	Cap. Range C2
LP	32 kHz	33 pF	33 pF
	200 kHz	15 pF	15 pF
XT	200 kHz	47-68 pF	47-68 pF
	1 MHz	15 pF	15 pF
	4 MHz	15 pF	15 pF
HS	4 MHz	15 pF	15 pF
	8 MHz	15-33 pF	15-33 pF
	20 MHz	15-33 pF	15-33 pF

Un oscillateur externe.



Un réseau RC.



Recommended values: 3 k R_{EXT} 100 k
 $C_{EXT} > 20\text{pF}$

Remarque : Les instructions standards durent **1** cycle machine (sauf les instructions de sauts **2** cycles). Le μC utilise **4** coups d'horloge pour réaliser un cycle machine.

Si la fréquence du **QUARTZ** est de **20MHz (T=50nS)**, une instruction sera exécutée toutes les **200nS**. Dans ce cas là, le μC a une puissance de calcul de **5MIPS (5 Millions d'instructions par secondes !!!)**.

La fréquence **MAX** est de **20MHz** pour les μC dont les références se terminent par **-20**.

Par exemples : **16F876-20 (20MHz max)** et **16F876-04 (4MHz max)**.

La fréquence **MIN** est le continu.

Remarque : La consommation du circuit sera d'autant plus faible que la fréquence sera petite, cela peut être intéressant pour des applications de faible consommation (alimentation autonome).

Pour des applications faible consommation, on peut utiliser les séries **LF (Low Frequency and Low Power)**.

5.3) Alimentation : VDD et VSS.

Ce sont les broches d'alimentation du circuit. Les tensions qui peuvent être appliquées vont :

- De **4,5V à 6V** pour la gamme standard **F**.
- De **2 à 6V** pour la gamme étendue **LF**.

L'intensité du courant consommé peut aller de **1µA à 10mA**.

La consommation du **µC** sera fonction de :

- La tension d'alimentation.
- La fréquence interne.
- Le mode de fonctionnement.

De plus ces bornes doivent être découplées par deux condensateurs :

- **1µF** électrolytique.
- **10nF** céramique.

5.4) L'Interruption : RBO/INT.

Cette broche a une double fonction elle peut être utilisée comme une broche standard **RBO** ou comme une entrée d'interruption **INT**.

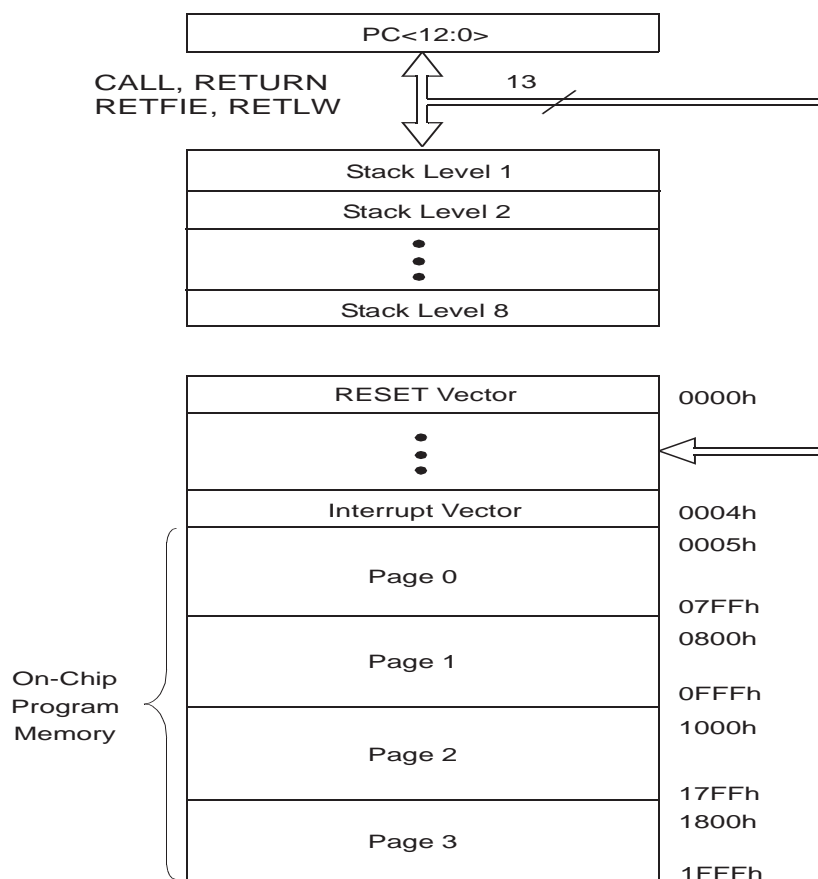
Si cette broche est utilisée comme une entrée d'interruption externe, elle doit être maintenue à un niveau haut par l'intermédiaire de résistances de **10 kΩ** pour ne pas déclencher d'interruptions imprévues, cela permet aussi de relier plusieurs sources d'interruptions sur une même ligne (**OU CABLE**).

6) L'UNITÉ CENTRALE.

6.1) Organisation mémoire.

Comme les **PICs** utilisent un bus pour les instructions et un bus pour les données, il faut considérer deux plans mémoire l'un pour les instructions et l'autre pour les données ainsi que les registres internes.

6.1.1) Plan Mémoire pour les instructions (code programme).



Le plan mémoire est linéaire les adresses vont de **0000h** à **1FFFh** (**8k mots de 14 bits**), par page de **2K mots** . On peut remarquer, le vecteur de reset est figé en **0000h**.

Les **PICs** n'ont qu'un seul vecteur d'interruption en **0004h**. Lors d'une interruption, le sous programme associé devra déterminer quel périphérique a demandé une interruption.

La pile utilisée par les sous programmes n'est pas implantée en mémoire de donnée comme avec les microcontrôleurs classiques, mais dans la mémoire programme. Elles sont utilisées lors d'appels de sous programmes, on ne peut pas imbriquer plus de **8** sous programmes (Ce qui est déjà beaucoup ! !).

LES MICROCONTROLEURS PIC 16F87X

6.1.2 Plan Mémoire pour les données et registres internes (SFR : Special Function Register).

Le plan mémoire des données et des registres internes est découpé en 4 zones ou bank de 128 octets, pour accéder à une zone il faut positionner les bits **RP0 (bit 5)** et **RP1 (bit 6)** du registre **STATUS**.

RP1 : RP0	Zone sélectionnée (BANK)
0 0	De 00h à 7Fh : BANK 0
0 1	De 80h à FFh : BANK 1
1 0	De 100h à 17Fh : BANK 2
1 1	De 180h à 1FFh : BANK 3

File Address		File Address		File Address		File Address							
Indirect addr. ^(*)	00h	Indirect addr. ^(*)	80h	Indirect addr. ^(*)	100h	Indirect addr. ^(*)	180h						
TMR0	01h	OPTION_REG	81h	TMR0	101h	OPTION_REG	181h						
PCL	02h	PCL	82h	PCL	102h	PCL	182h						
STATUS	03h	STATUS	83h	STATUS	103h	STATUS	183h						
FSR	04h	FSR	84h	FSR	104h	FSR	184h						
PORTA	05h	TRISA	85h		105h		185h						
PORTB	06h	TRISB	86h	PORTB	106h	TRISB	186h						
PORTC	07h	TRISC	87h		107h		187h						
PORTD ⁽¹⁾	08h	TRISD ⁽¹⁾	88h		108h		188h						
PORTE ⁽¹⁾	09h	TRISE ⁽¹⁾	89h		109h		189h						
PCLATH	0Ah	PCLATH	8Ah	PCLATH	10Ah	PCLATH	18Ah						
INTCON	0Bh	INTCON	8Bh	INTCON	10Bh	INTCON	18Bh						
PIR1	0Ch	PIE1	8Ch	EEDATA	10Ch	EECON1	18Ch						
PIR2	0Dh	PIE2	8Dh	EEADR	10Dh	EECON2	18Dh						
TMR1L	0Eh	PCON	8Eh	EEDATH	10Eh	Reserved ⁽²⁾	18Eh						
TMR1H	0Fh		8Fh	EEADRH	10Fh	Reserved ⁽²⁾	18Fh						
T1CON	10h		90h		110h		190h						
TMR2	11h	SSPCON2	91h	General Purpose Register 16 Bytes	111h	General Purpose Register 16 Bytes	191h						
T2CON	12h	PR2	92h		112h		192h						
SSPBUF	13h	SSPADD	93h		113h		193h						
SSPCON	14h	SSPSTAT	94h		114h		194h						
CCPR1L	15h		95h		115h		195h						
CCPR1H	16h		96h		116h		196h						
CCP1CON	17h		97h		117h		197h						
RCSTA	18h	TXSTA	98h		118h		198h						
TXREG	19h	SPBRG	99h		119h		199h						
RCREG	1Ah		9Ah		11Ah		19Ah						
CCPR2L	1Bh		9Bh		11Bh		19Bh						
CCPR2H	1Ch		9Ch		11Ch		19Ch						
CCP2CON	1Dh		9Dh		11Dh		19Dh						
ADRESH	1Eh	ADRESL	9Eh		11Eh		19Eh						
ADCON0	1Fh	ADCON1	9Fh		11Fh		19Fh						
	20h		A0h				120h		1A0h				
General Purpose Register 96 Bytes	7Fh	General Purpose Register 80 Bytes	EFh	General Purpose Register 80 Bytes	16Fh	General Purpose Register 80 Bytes	1EFh						
								accesses 70h-7Fh	F0h	accesses 70h-7Fh	170h	accesses 70h - 7Fh	1F0h
									FFh		17Fh		1FFh

■ Unimplemented data memory locations, read as '0'.

* Not a physical register.

Note 1: These registers are not implemented on the PIC16F876.

Note 2: These registers are reserved, maintain these registers clear.

Les registres appelés **General Purpose Register** ne sont ni plus ni moins que des cases mémoires pour stocker les données.

Remarque : Les 4 banques mémoires ne facilitent pas la gestion de la mémoire. Par exemple, pour accéder à la case mémoire **1A0h (BANK3)**, il ne faut pas oublier de positionner correctement les bits **RP0** et **RP1** du registre **STATUS**.

Code : Transfert du contenu de l'adresse **1A0h** dans le registre **W**.

```
BSF    STATUS,5      ; RP0 = 1 Sélection de la BANK 3
BSF    STATUS,6      ; RP1 = 1 Sélection de la BANK 3
MOVF   0x20,0        ; Transfert du contenu de l'adresse 1A0h dans W.
```

Pourquoi les concepteurs des **PICs** ont été obligés de scinder en 4 banques mémoires ? L'adresse des données est codée sur 7 bits, donc les adresses vont de **00h** à **7Fh**. Les 2 bits **RP0** et **RP1** sont 2 bits d'adresses supplémentaires et extérieurs, ce qui porte à 2⁹ adresses possibles ou **512** registres.

Code binaire de MOVF.

001000 d fff ffff *d : destination 0 si W et 1 si F*
 fff fff : adresse mémoire.

Pour l'exemple ci-dessus :

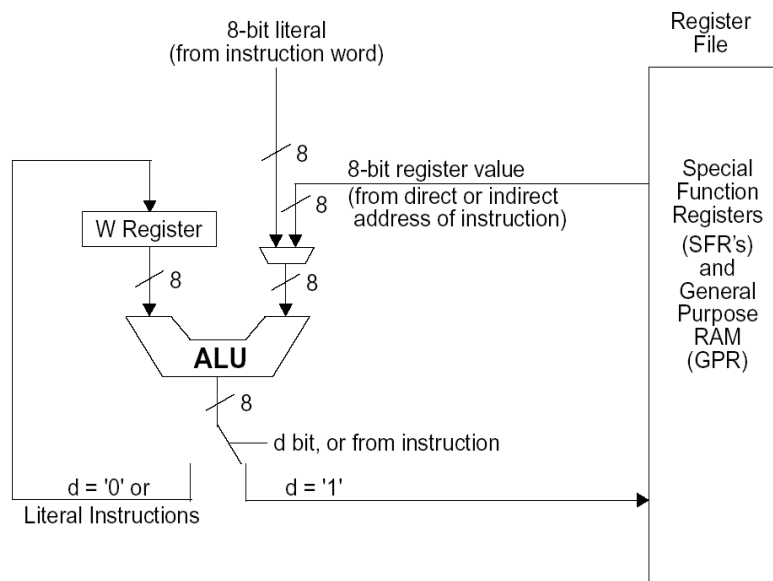
```
MOVF   0x20,0
d=0 et fff fff = 010 0000
```

Donc le code de MOVF 0x20,0 est : 001000 0 010 0000

En conclusion, la gestion de la mémoire avec les **PICs** n'est pas très facile, la programmation demande de la rigueur.

La programmation des **PICs** avec des langages de hauts niveaux tels que **BASIC**, **PASCAL** et **langage C** permet de s'affranchir de la gestion des pages mémoires.

6.1.2) L'UAL : L'unité Arithmétique et Logique.



Elle est composée :

- D'un accumulateur 8 bits **W : WORKING** (travail), c'est lui qui effectue toutes les opérations arithmétiques et logiques.
- Un registre d'état 8 bits **STATUS**.

R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x
IRP	RP1	RP0	\overline{TO}	\overline{PD}	Z	DC	C
bit 7					bit 0		

bit 7 IRP: Register Bank Select bit (used for indirect addressing)
 1 = Bank 2, 3 (100h - 1FFh)
 0 = Bank 0, 1 (00h - FFh)

bit 6-5 RP1:RP0: Register Bank Select bits (used for direct addressing)
 11 = Bank 3 (180h - 1FFh)
 10 = Bank 2 (100h - 17Fh)
 01 = Bank 1 (80h - FFh)
 00 = Bank 0 (00h - 7Fh)
 Each bank is 128 bytes

bit 4 \overline{TO} : Time-out bit
 1 = After power-up, CLRWD instruction, or SLEEP instruction
 0 = A WDT time-out occurred

bit 3 \overline{PD} : Power-down bit
 1 = After power-up or by the CLRWD instruction
 0 = By execution of the SLEEP instruction

bit 2 Z: Zero bit
 1 = The result of an arithmetic or logic operation is zero
 0 = The result of an arithmetic or logic operation is not zero

bit 1 DC: Digit carry/borrow bit (ADDWFADDLW, SUBLW, SUBWF instructions)
 (for borrow, the polarity is reversed)
 1 = A carry-out from the 4th low order bit of the result occurred
 0 = No carry-out from the 4th low order bit of the result

bit 0 C: Carry/borrow bit (ADDWFADDLW, SUBLW, SUBWF instructions)
 1 = A carry-out from the Most Significant bit of the result occurred
 0 = No carry-out from the Most Significant bit of the result occurred

Note: For borrow, the polarity is reversed. A subtraction is executed by adding the two's complement of the second operand. For rotate (RRF, RLF) instructions, this bit is loaded with either the high, or low order bit of the source register.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

LES MICROCONTROLEURS PIC 16F87X

7) JEU D'INSTRUCTIONS.

Mnemonic, Operands	Description	Cycles	14-Bit Opcode				Status Affected	Notes	
			MSb	LSb					
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	df ff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	df ff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	lf ff	ffff	Z	2
CLRWF	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	df ff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	df ff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	df ff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	df ff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	df ff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	df ff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	df ff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	lf ff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	df ff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	df ff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	df ff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	df ff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	df ff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bf ff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bf ff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bf ff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bf ff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add literal and W	1	11	111x	kk kk	kkkk	C,DC,Z	
ANDLW	k	AND literal with W	1	11	1001	kk kk	kkkk	Z	
CALL	k	Call subroutine	2	10	0kkk	kk kk	kkkk		
CLRWDI	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}, \overline{PD}$	
GOTO	k	Go to address	2	10	1kkk	kk kk	kkkk		
IORLW	k	Inclusive OR literal with W	1	11	1000	kk kk	kkkk	Z	
MOVLW	k	Move literal to W	1	11	00xx	kk kk	kkkk		
RETFIE	-	Return from interrupt	2	00	0000	0000	1001		
RETLW	k	Return with literal in W	2	11	01xx	kk kk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into standby mode	1	00	0000	0110	0011	$\overline{TO}, \overline{PD}$	
SUBLW	k	Subtract W from literal	1	11	110x	kk kk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR literal with W	1	11	1010	kk kk	kkkk	Z	

- Note 1: When an I/O register is modified as a function of itself (e.g., MOVF PORTB, 1), the value used will be that value present on the pins themselves. For example, if the data latch is '1' for a pin configured as input and is driven low by an external device, the data will be written back with a '0'.
- 2: If this instruction is executed on the TMR0 register (and, where applicable, d = 1), the prescaler will be cleared if assigned to the Timer0 module.
- 3: If Program Counter (PC) is modified, or a conditional test is true, the instruction requires two cycles. The second cycle is executed as a NOP

LEGENDE DES TABLEAUX D'INSTRUCTIONS

d	Détermine la destination : d=0 Registre Travail d=1 Registre Mémoire SFR ou RAM.
f	Adresse SFR ou RAM en hexa.
fff ffff	Adresse SFR ou RAM en binaire.
k	Valeur immédiate sur 8bits ou Adresse de destination sur 11 bits.
b	Valeur décimale, elle détermine le bit à modifier ou à tester.
bbb	Valeur binaire, elle détermine le bit à modifier ou à tester.

Remarque: toutes les instructions ne durent qu'un seul cycle machine, sauf les instructions de sauts tels que GOTO, CALL.

8) LES MODES D'ADRESSAGES

8.1) Adressage inhérent ou implicite

8.1.1) Description.

Le mnémonique de l'instruction mentionne la donnée sur laquelle porte l'opération (contenu des registres), ou aucune donnée n'est nécessaire.

8.1.2) Syntaxe.

MNEMONIQUE

8.1.3) Exemples.

CLRW ; Mise à zéro de W
NOP ; aucune opération (temporisation)
SLEEP ; Mise en sommeil du µC

8.2) Adressage immédiat.

8.2.1) Description.

L'instruction porte sur une valeur constante indiquée immédiatement après le mnémonique.

8.2.2) Syntaxe.

MNEMONIQUE **constante**

8.2.3) Exemples.

MOVLW 255 ; charge 0xFF dans W
ADDLW 0x20 ; additionne 32 avec W et met le résultat dans W

8.3) Adressages direct et étendu.

8.3.1) Description.

Les **PICs** ne disposent pas vraiment de modes d'adressages **DIRECT** et **ETENDU**, l'adressage de la mémoire de données se fait dans la page sélectionnée par les **BIT 5 (RP0)** et **BIT 6 (RP1)** du registre **STATUS**.

8.3.2) Syntaxe.

MNEMONIQUE *f,d*
d=0 Registre W comme destination (*WORKING*)
d=1 Registre *f* comme destination (un des registers *SFR*)

8.3.3) Exemple.

```

PORTB          EQU    0x06
VARIABLE       EQU    0x20      ; page 0

; S'assurer que nous sommes bien en BANK 0
BCF            STATUS,5        ; RP0 = 0 Sélection de la BANK 0
BCF            STATUS,6        ; RP1 = 0 Sélection de la BANK 0

; PORTB <- VARIABLE
MOVWF         VARIABLE,0      ; Transfert le contenu de VARIABLE
                                   ; dans W
MOVWF         PORTB          ; Transfert le contenu de W
                                   ; dans le registre PortB
    
```

8.4) Adressage relatif.

8.4.1) Description.

Ce mode d'adressage n'existe pas vraiment, mais des instructions permettent de réaliser des sauts de programme, ceux sont les instructions **GOTO** et **CALL**.

8.4.2) Syntaxe.

MNEMONIQUE *Adresse*

8.4.3) Exemple N°1.

Deux cas sont à considérer :

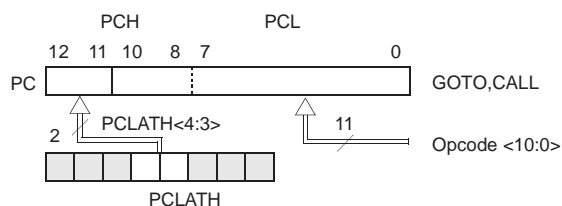
1) Les sauts dans la même page mémoire :

```
CALL TEMPO      ; Appel du SP TEMPO
GOTO FIN        ; Branchement à l'étiquette FIN
```

2) Les sauts dans une page mémoire différente, dans ce cas il faut positionner correctement les bits 4 et 3 du registre **PCLATCH**, pour accéder à la bonne page mémoire.

```
BSF PCLATCH,3   ; 1 et 1 Sélection de la BANK 3 de 1800h à 1FFFh
BSF PCLATCH,4
CALL CONV
```

En effet pourquoi **PCLATCH** et pas **PCH** ? **PCH** n'est pas accessible directement il faut passer par **PCLATCH** pour que les bits 4 et 3 de celui-ci soient recopiés dans **PCH**.



Remarque : Pour le retour de sous programme, il n'est pas nécessaire de se préoccuper de ces bits car la valeur du **PC** est mémorisée sur 13 bits dans la pile.

8.5) Adressage indirect ou encore indexé.

8.5.1) Description.

Les **PICs** disposent à travers les registres **INDF(ou f0)** et **FSR(ou f4)** d'un mode d'adressage indexé, la structure est un peu particulière, **FSR** est le registre d'index et **INDF** permet d'accéder à son contenu.

8.5.2) Syntaxe.

MNEMONIQUE *INDF,d*

8.5.3) Exemple.

```
PORTB <- TAB_VAL[4]
Récupérer le 4ème élément d'une table TAB_VAL ;
```

```
; PORTB <- TAB_VAL[4]
MOVLW    TAB_VAL      ; W <- Adresse de TAB_VAL
ADDLW    4             ; W <- W + 4
MOWF    FSR           ; Adresse + 4 dans le registre d'index FSR
MOVWF    INDF,0       ; Transfert du contenu de TABLE[4] dans W
MOVWF    PORTB        ; Transfert du contenu de W sur le PORTB
```

8.6) Manipulation de bits.

8.6.1) Forçage de bits.

8.6.1.1) Description.

Il s'agit de **2** instructions permettant de mettre à **0** ou **1** un bit d'un octet de l'espace mémoire **SFR**. Elles sont le plus souvent utilisées pour positionner des bits des registres du **µC**.

8.6.1.2) Syntaxe.

BSF *f,b* *pour mettre à 1*
ou *BCF* *f,b* *pour mettre à 0*

8.6.1.3) Exemples.

```
BCF            PORTA,2            ; Mise à 0 du bit 2 du PORTA
BSF            STATUS,0           ; Mise à 1 du bit 0 du registre STATUS
                         ; C'est-à-dire la CARRY
```

8.6.2) Test de bits.

8.6.2.1) Description.

Il s'agit de **2** instructions permettant de tester un bit d'un octet de l'espace mémoire **SFR**. Elles sont le plus souvent utilisées pour déterminer l'état des bits des registres du **µC**.

En fonction du résultat du test :

- le programme se poursuit avec l'instruction suivante (résultat du test faux)
- le programme saute l'instruction qui suit le test.

8.6.2.2) Syntaxe.

BTFSS *f,b*
ou *BTFSC* *f,b*

8.6.2.3) Exemple.

```
MOVF            CMP,1            ; Transfert du contenu CMP dans CMP
                         ; Cela permet de tester si le contenu
                         ; de CMP est nul en positionnant le bit Z
BTFSS            STATUS,Z           ; Test du bit Z ?
GOTO            SINON           ; Z=0 alors exécuter le code pour SINON
MOVLW           0xFF            ; Alors CMP <- 0xFF
MOVWF           CMP
GOTO            FSI           ; allez en fin de si
SINON    DECWF        CMP,1           ; CMP <- CMP -1 Décrémenter CMP
.....
FSI            .....
```

9) LES PORTS D'ENTREES SORTIES

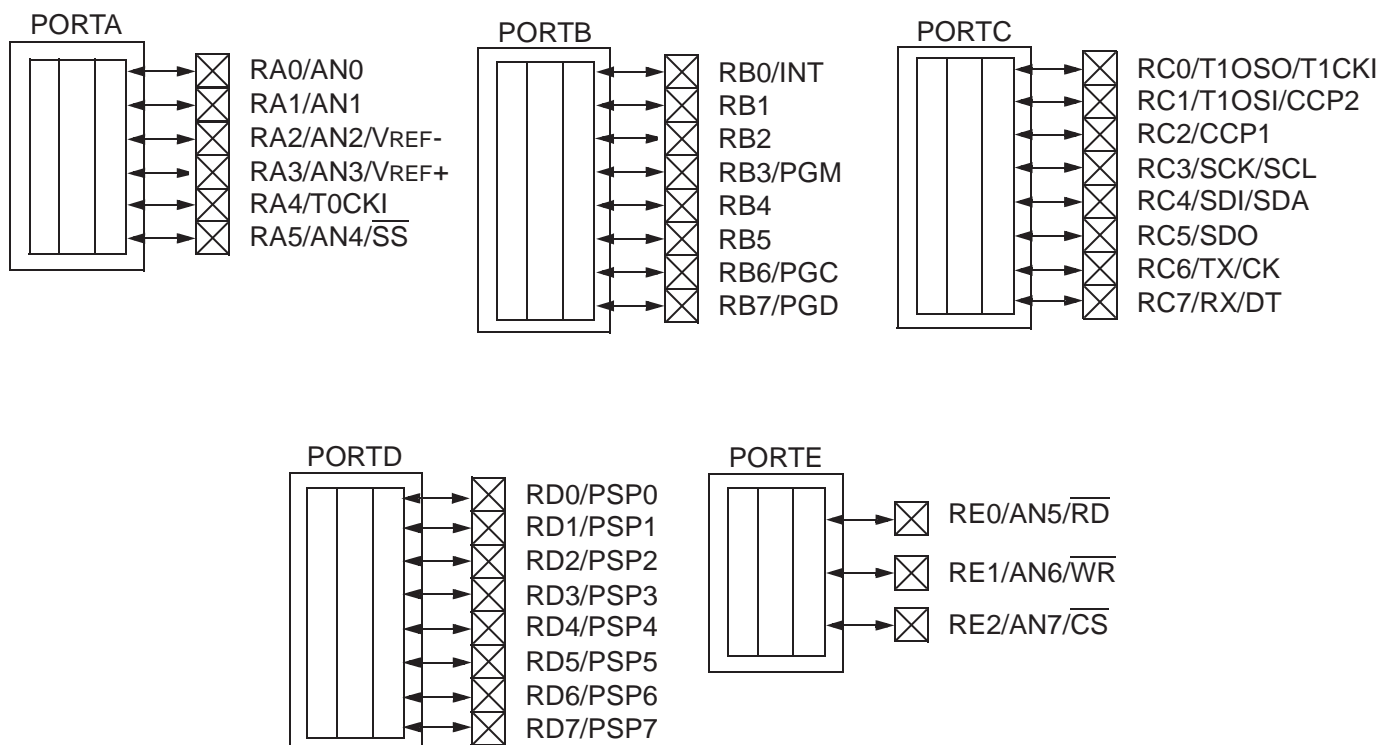
9.1) Généralités.

Le μC dispose de **3 PORTS (A,B et C)** pour le **16F876** et **5 PORTS (A,B,C,D et E)** pour le **16F877**.
Tous les ports d'entrées sorties **Input/ Output** sont bidirectionnels.

La plupart des lignes de **PORTs** ont une double fonction.

- Le **PORT A (5 bits) I/O** pure et/ou convertisseur analogique et/ou **TIMER 0**.
La broche **RA4** du **PORT A** (Entrée du **TIMER 0 T0CKI**) est du type **DRAIN OUVERT**.
- Le **PORT B (8 bits) I/O** pure et/ou programmation in situ **ICSP/ICD** (Broche **RB3/PGM**, **RB6/PGC** et **RB7/PGD**) et l'entrée d'interruption externe **RB0/INT**.
Remarque : Si le **PIC** est utilisé en mode **ICSP/ICD** il faut laisser libre les broches **RB3/PGM**, **RB6/PGC** ainsi que **RB7/PGD**) et les configurer en entrée.
- Le **PORT C (8 bits) I/O** pure et/ou **TIMER 1** et/ou **SPI / I2C** et/ou **USART**.
- Le **PORT D (8 bits) I/O** pure et/ou port parallèle **8 bits** associé au **PORT E**.
- Le **PORT E (3 bits) I/O** pure et/ou pilotage du **PORT E RE0/RD**, **RE1/WR** et **RE2/CS**.

Toutes les lignes de **PORTs** peuvent fournir un courant de **25mA** par ligne de **PORT**. Une limite de **40mA** par **PORT** doit être respectée pour des questions de dissipation.



9.2 Configuration des PORTx , les registres PORTx et TRISx.

Tous les ports sont pilotés par deux registres :

- Le registre de **PORTx**, si le **PORT x** ou certaines lignes de **PORT X** sont configurées en sortie, ce registre détermine l'état logique des sorties.
- Le registre **TRISx**, c'est le registre de direction. Il détermine si le **PORTx** ou certaines lignes de port sont en entrée ou en sortie. L'écriture d'un **1** logique correspond à une **entrée** (**1** comme Input) et l'écriture d'un **0** logique correspond à une sortie (**0** comme Output).
Au **RESET** toutes les lignes de ports sont configurées en entrées.

Remarque : Les registres **TRISx** appartiennent à la **BANQUE 1** des **SFR**.

Lors de l'initialisation du **µC** il ne faut pas oublier de changer de page mémoire pour les configurer.

Exemple : On souhaite obtenir la configuration suivante des **PORTA** et **PORTB**.

SENS	NC	NC	S	E	S	S	S	S	E
PORTA	NC	NC	RA5	RA4	RA3	RA2	RA1	RA0	

SENS	E	E	S	S	E	S	S	S	E
PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	

Programme en assembleur.

Remarque toutes les lignes de sorties des **PORTs** sont mises à zéro.

```

; Mise à zéro des registres de données des ports A et B
clrf PORTA
clrf PORTB
; Configuration des PORTA et B

; Accès aux registres TRISx (Banque mémoire 1)
bsf STATUS,RP0 ; RP0 = 1
bcf STATUS,RP1 ; RP1 = 0

; Configuration des registres de direction
; Configuration du PORTA X X S E S S S E
movlw B'11010001' ; valeur binaire 1 1 0 1 0 0 0 1
movwf TRISA
; Configuration du PORTB E E S S E S S E
movlw B'11001001' ; valeur binaire 1 1 0 0 1 0 0 1
movwf TRISB

; Retour en banque mémoire 0
bcf STATUS,RP0 ; RP0 = 0
bcf STATUS,RP1 ; RP1 = 0

.....
.....

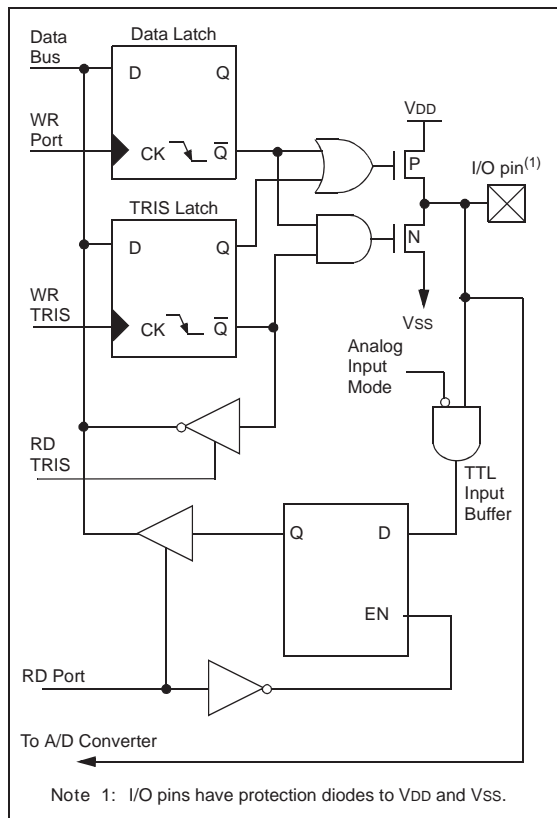
```

9.3) Le PORT A.

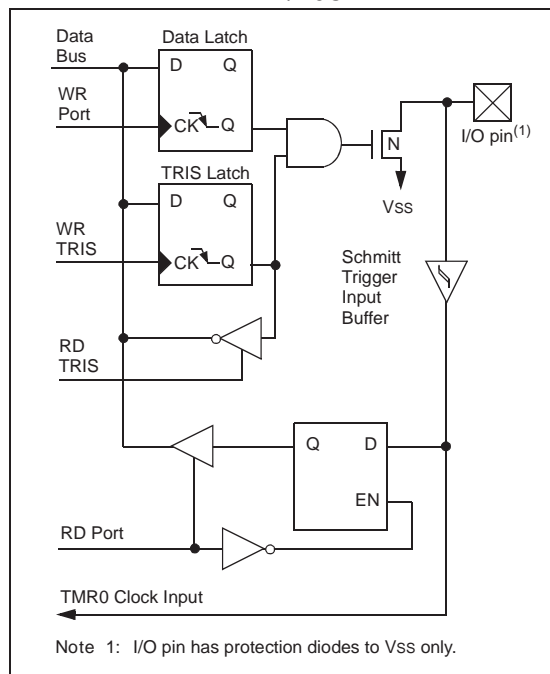
Le PORT A (5 bits) I/O pure et/ou convertisseur analogique et/ou TIMER 0.

Attention à PA4 (Entrée du TIMER 0 T0CKI), elle est de type DRAIN OUVERT.

BLOCK DIA GRAM OF
RA3:RA0 AND RA5 PINS



BLOCK DIA GRAM OF
RA4/T0CKI PIN



PORTA FUNCTIONS

Name	Bit#	Buffer	Function
RA0/AN0	bit0	TTL	Input/output or analog input.
RA1/AN1	bit1	TTL	Input/output or analog input.
RA2/AN2	bit2	TTL	Input/output or analog input.
RA3/AN3/VREF	bit3	TTL	Input/output or analog input or VREF.
RA4/T0CKI	bit4	ST	Input/output or external clock input for Timer0. Output is open drain type.
RA5/SS/AN4	bit5	TTL	Input/output or slave select input for synchronous serial port or analog input.

Legend: TTL = TTL input, ST = Schmitt Trigger input

SUMMARY OF REGISTERS ASSOCIATED WITH PORTA

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
05h	PORTA	—	—	RA5	RA4	RA3	RA2	RA1	RA0	--0x 0000	--0u 0000
85h	TRISA	—	—	PORTA Data Direction Register						--11 1111	--11 1111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.

Shaded cells are not used by PORTA.

LES MICROCONTROLEURS PIC 16F87X

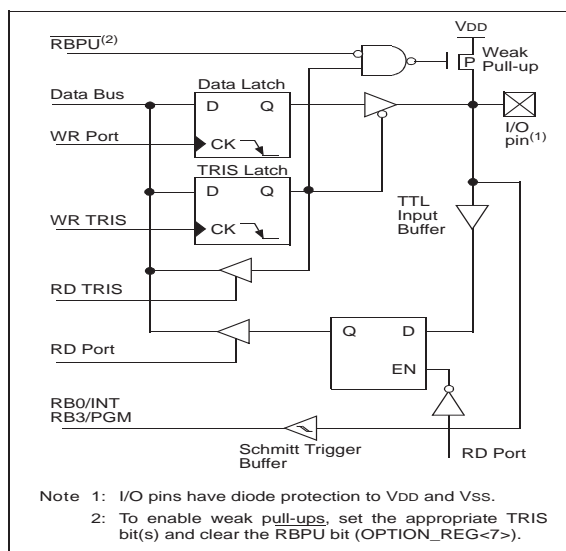
9.3) Le PORT B.

Le **PORT B** dispose de **(8 bits) I/O** pure et/ou programmation in situ **ICSP/ICD** (Broche **RB3/PGM, RB6/PGC** et **RB7/PGD**) et une entrée d'interruption externe **RB0/INT**.

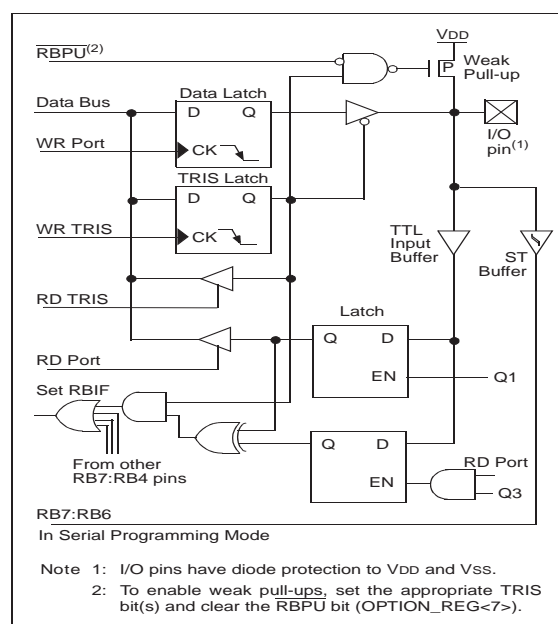
Il dispose de deux fonctions spéciales :

- La première c'est la possibilité de configurer toutes ses lignes avec une résistance de **PULL-UP** en configurant le bit **RBP** à **0** du registre **OPTION**.
- La deuxième, c'est la possibilité de générer une interruption sur un changement d'état des broches **RB4** à **RB7**. C'est très pratique pour la gestion des claviers matricés.

BLOCK DIAGRAM OF RB3:RB0 PINS



BLOCK DIAGRAM OF RB7:RB4 PINS



PORTB FUNCTIONS

Name	Bit#	Buffer	Function
RB0/INT	bit0	TTL/ST ⁽¹⁾	Input/output pin or external interrupt input. Internal software programmable weak pull-up.
RB1	bit1	TTL	Input/output pin. Internal software programmable weak pull-up.
RB2	bit2	TTL	Input/output pin. Internal software programmable weak pull-up.
RB3/PGM ⁽³⁾	bit3	TTL	Input/output pin or programming pin in LVP mode. Internal software programmable weak pull-up.
RB4	bit4	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB5	bit5	TTL	Input/output pin (with interrupt-on-change). Internal software programmable weak pull-up.
RB6/PGC	bit6	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming clock.
RB7/PGD	bit7	TTL/ST ⁽²⁾	Input/output pin (with interrupt-on-change) or In-Circuit Debugger pin. Internal software programmable weak pull-up. Serial programming data.

Legend: TTL = TTL input, ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.

Note 2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.

Note 3: Low Voltage ICSP Programming (LVP) is enabled by default, which disables the RB3 I/O function. LVP must be disabled to enable RB3 as an I/O pin and allow maximum compatibility to the other 28-pin and 40-pin mid-range devices.

SUMMARY OF REGISTERS ASSOCIATED WITH PORTB

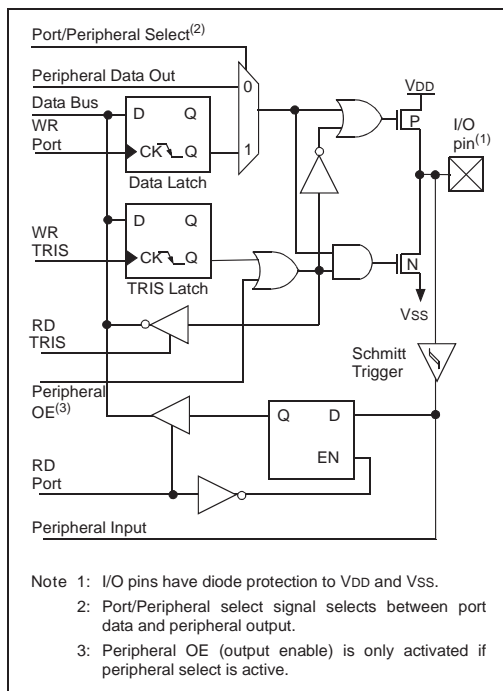
Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
06h, 106h	PORTB	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	xxxx xxxx	uuuu uuuu
86h, 186h	TRISB	PORTB Data Direction Register								1111 1111	1111 1111
81h, 181h	OPTION_REG	RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

Legend: x = unknown, u = unchanged. Shaded cells are not used by PORTB.

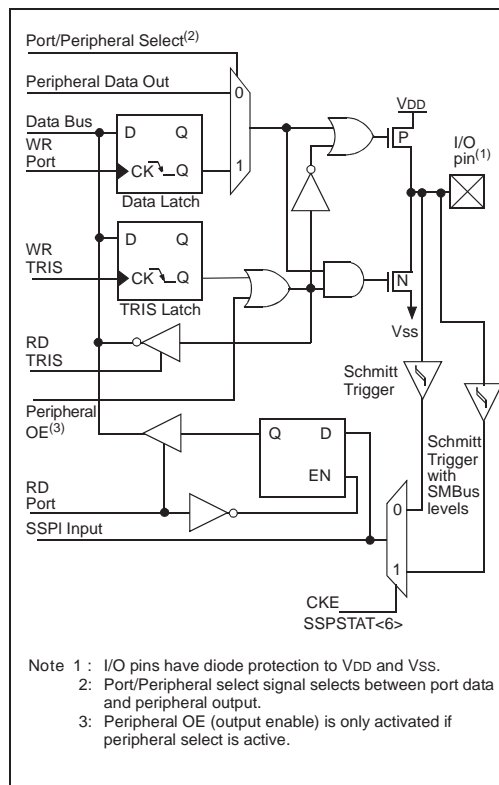
9.4) Le PORT C.

Le **PORT C (8 bits) I/O** pure qu'il partage avec le **TIMER 1**, la liaison **SPI / I2C** et l'**USART**.

**PORTC BLOCK DIAGRAM
(PERIPHERAL OUTPUT
OVERRIDE) RC<2:0>,
RC<7:5>**



**PORTC BLOCK DIAGRAM
(PERIPHERAL OUTPUT
OVERRIDE) RC<4:3>**



PORTC FUNCTIONS

Name	Bit#	Buffer Type	Function
RC0/T1OSO/T1CKI	bit0	ST	Input/output port pin or Timer1 oscillator output/Timer1 clock input.
RC1/T1OSI/CCP2	bit1	ST	Input/output port pin or Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	bit2	ST	Input/output port pin or Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	bit3	ST	RC3 can also be the synchronous serial clock for both SPI and I ² C modes.
RC4/SDI/SDA	bit4	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I ² C mode).
RC5/SDO	bit5	ST	Input/output port pin or Synchronous Serial Port data output.
RC6/TX/CK	bit6	ST	Input/output port pin or USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	bit7	ST	Input/output port pin or USART Asynchronous Receive or Synchronous Data.

Legend: ST = Schmitt Trigger input

SUMMARY OF REGISTERS ASSOCIATED WITH PORTC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
07h	PORTC	RC7	RC6	RC5	RC4	RC3	RC2	RC1	RC0	xxxx xxxx	uuuu uuuu
87h	TRISC	PORTC Data Direction Register								1111 1111	1111 1111

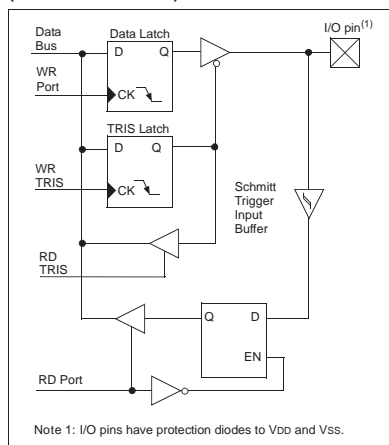
Legend: x = unknown, u = unchanged

LES MICROCONTROLEURS PIC 16F87X

9.5) Les PORT D et E.

Le **PORT D (8 bits) I/O** et **PORT E (3 bits)** utilisent la même type structure interne.

**PORTD BLOCK DIAGRAM
(IN I/O PORT MODE)**



PORTD FUNCTIONS

Name	Bit#	Buffer Type	Function
RD0/PSP0	bit0	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit0.
RD1/PSP1	bit1	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit1.
RD2/PSP2	bit2	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit2.
RD3/PSP3	bit3	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit3.
RD4/PSP4	bit4	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit4.
RD5/PSP5	bit5	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit5.
RD6/PSP6	bit6	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit6.
RD7/PSP7	bit7	ST/TTL ⁽¹⁾	Input/output port pin or parallel slave port bit7.

Legend: ST = Schmitt Trigger input, TTL = TTL input

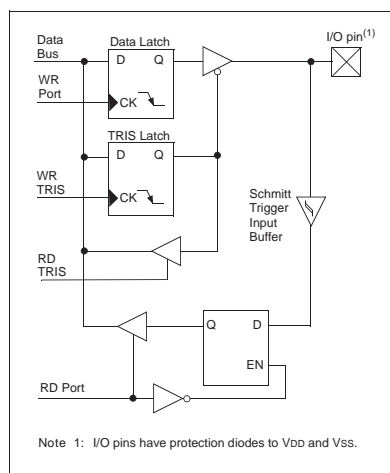
Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

SUMMARY OF REGISTERS ASSOCIATED WITH PORTD

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
08h	PORTD	RD7	RD6	RD5	RD4	RD3	RD2	RD1	RD0	xxxx xxxx	uuuu uuuu
88h	TRISD	PORTD Data Direction Register								1111 1111	1111 1111
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTD.

**PORTE BLOCK DIAGRAM
(IN I/O PORT MODE)**



PORTE FUNCTIONS

Name	Bit#	Buffer Type	Function
RE0/RD [−] /AN5	bit0	ST/TTL ⁽¹⁾	I/O port pin or read control input in Parallel Slave Port mode or analog input: RD 1 = Idle 0 = Read operation. Contents of PORTD register are output to PORTD I/O pins (if chip selected)
RE1/WR [−] /AN6	bit1	ST/TTL ⁽¹⁾	I/O port pin or write control input in Parallel Slave Port mode or analog input: WR 1 = Idle 0 = Write operation. Value of PORTD I/O pins is latched into PORTD register (if chip selected)
RE2/CS [−] /AN7	bit2	ST/TTL ⁽¹⁾	I/O port pin or chip select control input in Parallel Slave Port mode or analog input: CS 1 = Device is not selected 0 = Device is selected

Legend: ST = Schmitt Trigger input, TTL = TTL input

Note 1: Input buffers are Schmitt Triggers when in I/O mode and TTL buffers when in Parallel Slave Port mode.

SUMMARY OF REGISTERS ASSOCIATED WITH PORTE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
09h	PORTE	—	—	—	—	—	RE2	RE1	RE0	--- -xxx	--- -uuu
89h	TRISE	IBF	OBF	IBOV	PSPMODE	—	PORTE Data Direction Bits			0000 -111	0000 -111
9Fh	ADCON1	ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0	--0- 0000	--0- 0000

Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by PORTE.

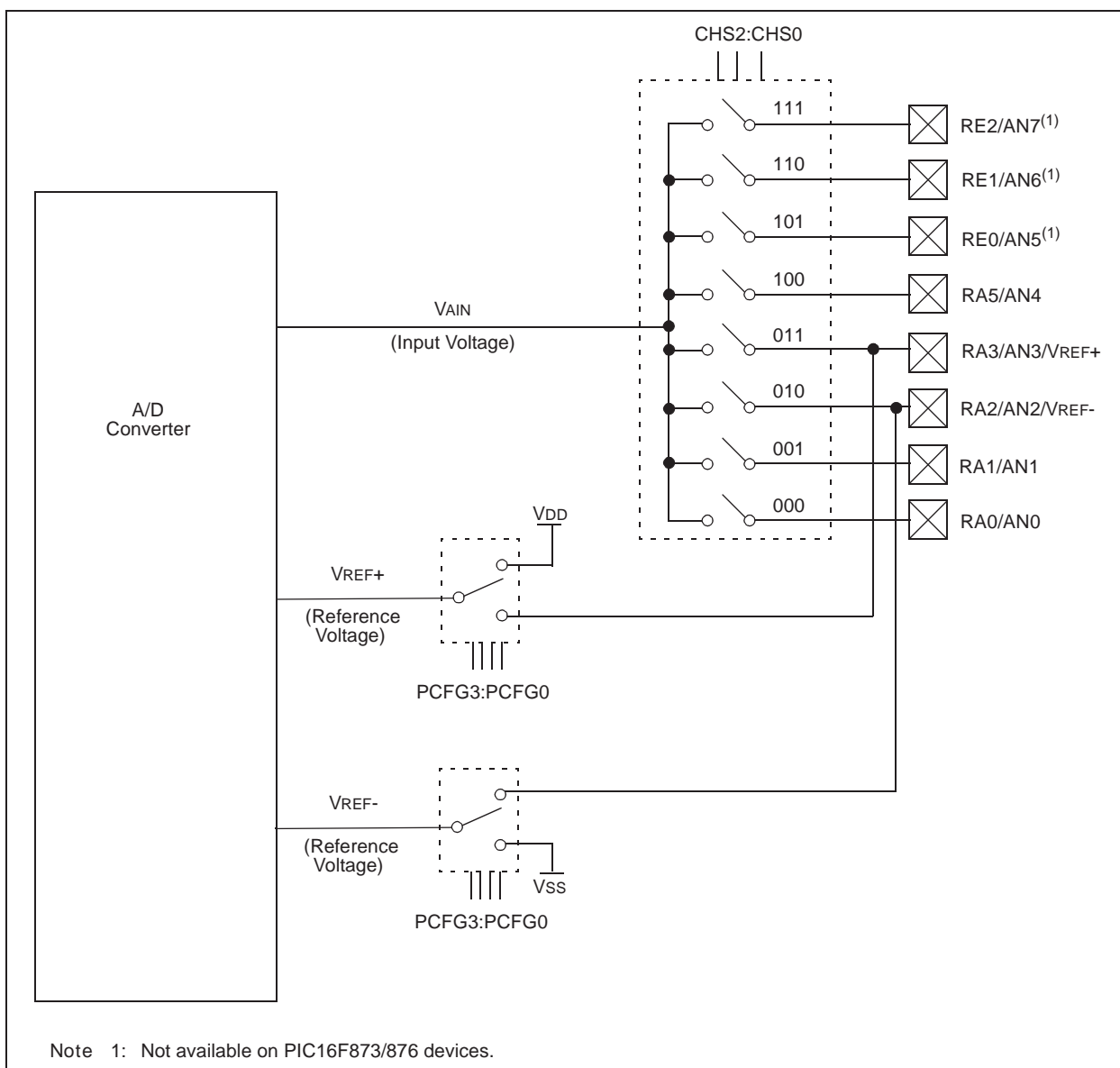
10) LE CONVERTISSEUR ANALOGIQUE NUMERIQUE.

Le convertisseur analogique numérique est à approximations successives et il possède une résolution de 10 bits. Il est composé de :

- Un multiplexeur analogique **5 voies (PIC16F876)** ou **8 voies PIC16F877**).
- Un échantillonneur bloqueur.
- Un Converteur Analogique Numérique de **10 bits**.

10.1) Organisation interne.

A/D BLOCK DIAGRAM



Remarque : Les entrées analogiques **RE1/AN5**, **RE2/AN6** et **RE3/AN2** sont disponibles avec les **PICs 16F877** et **16F874**.

De plus la broche **RA4** n'est pas concernée pas le convertisseur.

10.2) Fonctionnement du convertisseur.

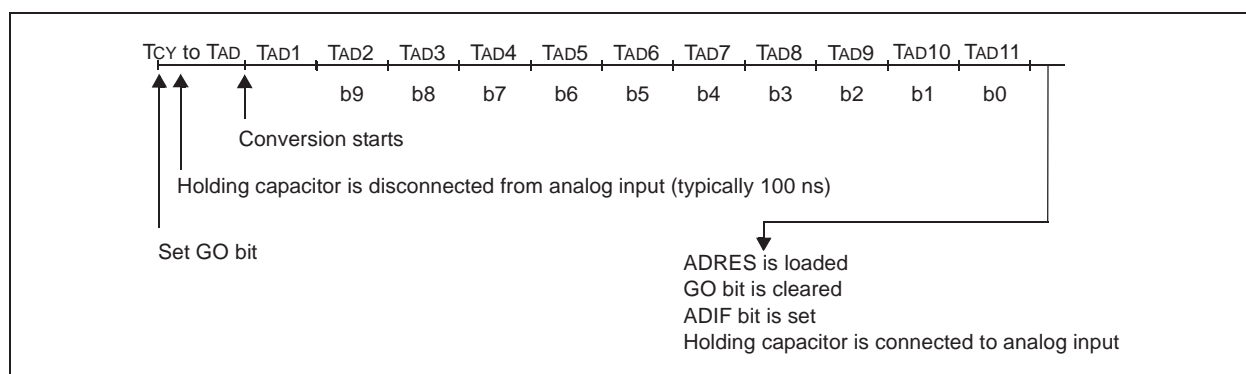
La conversion se passe en 2 temps :

- **1^{er} temps** le signal à convertir est appliqué sur l'entrée à convertir, ce signal doit être présent au moins pendant le temps **Tacq (temps d'acquisition environ 20µS pour 5V)**.
- **2^{ème} temps** la conversion, approximations successives.

Le temps de conversion minimum est de **12 TAD** (**TAD** c'est le temps de conversion dépendant de l'horloge interne, typiquement **1,6µS**).

Une conversion commence toujours par la mise à **1** du bit **GO/DONE** du registre **ADCON0**. Lorsque la conversion est terminée se bit repasse à **0**.

Donc pour pouvoir lire le résultat dans les registres **ADRESL** et **ADRESH** il suffit d'attendre que le bit **GO/DONE** passe à **0**.



La valeur résultante **N** de la conversion **ADRESH:ADRESL** est égale à :

$$N \text{ (valeur numérisée)} = ((VIN - VREF-) / (VREF+ - VREF-)) * 1023$$

Si **VREF+ = VDD = 5V** et **VREF- = VSS = 0V** alors

$$N \text{ (valeur numérisée)} = 1023 * (VIN / 5)$$

Mais avant de réaliser une conversion il faut définir la configuration du convertisseur :

- Le nombre d'entrées analogiques.
- Le nombre d'entrées logiques.
- Le type de tension de référence :
 - Interne **VREF = VDD - VSS**.
 - Externe, soit **VREF = VREF+ - VSS** ou **VREF = VREF+ - VREF-**.

Cette configuration se fait à travers le registre **ADCON1**, voir page suivante.

10.3) Le registre **ADCON1**.

Il permet de choisir une configuration parmi les **16** proposées.

Remarque : La configuration de ce registre **ADCON1** ne dispense pas de configurer les registres de directions des **PPORTA** et **PORTE** respectivement **TRISA** et **TRISE**.

ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7				bit 0			

- bit 7 ADFM: A/D Result Format Select bit
1 = Right justified. 6 Most Significant bits of ADRESH are read as '0'.
0 = Left justified. 6 Least Significant bits of ADRESL are read as '0'.
- bit 6-4 Unimplemented: Read as '0'
- bit 3-0 PCFG3:PCFG0: A/D Port Configuration Control bits:

PCFG3: PCFG0	AN7 ⁽¹⁾ RE2	AN6 ⁽¹⁾ RE1	AN5 ⁽¹⁾ RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs ⁽²⁾
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

A = Analog input D = Digital I/O

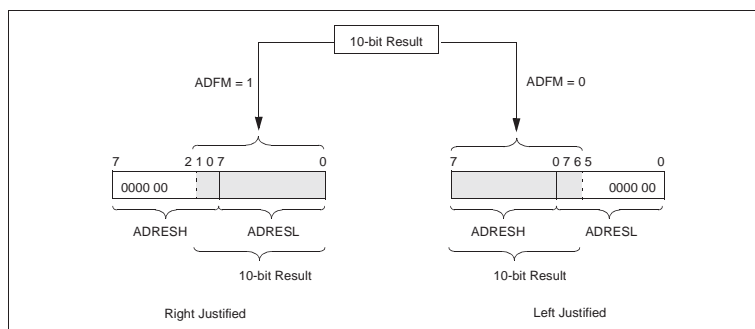
- Note 1: These channels are not available on PIC16F873/876 devices.
- Note 2: This column indicates the number of analog channels available as A/D inputs and the number of analog channels used as voltage reference inputs.

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

De plus le bit **ADFM** permet de choisir entre deux types de justification pour le résultat.

- Si **ADFM=1** alors le résultat sera justifié à **droite** dans les registre **ADRESH** et **ADRESL**, c'est-à-dire **ADRESL** contient les bits **7 à 0** du résultat de la conversion et **ADRESH** contient **6** zéros suivi des bit **9 et 8** du résultat de la conversion.
- Si **ADFM=0** alors le résultat sera justifié à **gauche**, c'est-à-dire **ADRESH** contient les bits **9 à 2** du résultat de la conversion et **ADRESL** contient les bit **1 et 0** du résultat de la conversion suivi de **6** zéros.

A/D RESULT JUSTIFICATION



10.4) Le registre **ADCON0**.

Ce registre permet de définir l'horloge de conversion (bit **ADCS1** et **ADCS0**), le canal à convertir (**CHS2**, **CHS1** et **CHS0**) et **ADON** bit de mise en fonctionnement.

Remarque : Lors de la mise en fonctionnement du **CAN** par le bit **ADON**, Le bit **GO/DONE** ne doit pas être modifié en même temps, c'est-à-dire dans la même instruction. .

ADCON0 REGISTER (ADDRESS : 1Fh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
	ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
bit 7								bit 0

bit 7-6 ADCS1:ADCS0: A/D Conversion Clock Select bits

00 = FOSC/2
 01 = FOSC/8
 10 = FOSC/32
 11 = FRC(clock derived from the internal A/D module RC oscillator)

bit 5-3 CHS2:CHS0: Analog Channel Select bits

000 = channel 0, (RA0/AN0)
 001 = channel 1, (RA1/AN1)
 010 = channel 2, (RA2/AN2)
 011 = channel 3, (RA3/AN3)
 100 = channel 4, (RA5/AN4)
 101 = channel 5, (RE0/AN5)⁽¹⁾
 110 = channel 6, (RE1/AN6)⁽¹⁾
 111 = channel 7, (RE2/AN7)⁽¹⁾

bit 2 GO/DONE: A/D Conversion Status bit

If **ADON = 1**:
 1 = A/D conversion in progress (setting this bit starts the A/D conversion)
 0 = A/D conversion not in progress (this bit is automatically cleared by hardware when the A/D conversion is complete)

bit 1 Unimplemented: Read as '0'

bit 0 ADON: A/D On bit

1 = A/D converter module is operating
 0 = A/D converter module is shut-off and consumes no operating current

Note 1: These channels are not available on PIC16F873/876 devices.

Legend:

R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared x = Bit is unknown

Le choix de l'horloge est déterminé par les bit **ADCS1** et **ADCS0**, sachant que le temps **TAD** doit être au minimum de **1,6µS**.

TAD vs. MAXIMUM DEVICE OPERATING FREQUENCIES (STANDARD DEVICES (C))

AD Clock Source (TAD)		Maximum Device Frequency
Operation	ADCS1:ADCS0	Max.
2Tosc	00	1.25 MHz
8Tosc	01	5 MHz
32Tosc	10	20 MHz
RC ^(1, 2, 3)	11	(Note 1)

Note 1: The RC source has a typical TAD time of 4 µs, but can vary between 2-6 µs.
 2: When the device frequencies are greater than 1 MHz, the RC A/D conversion clock source is only recommended for SLEEP operation.
 3: For extended voltage devices (LC), please refer to the Electrical Characteristics (Sections 15.1 and 15.2).

10.5) Exemple d'utilisation.

Exemple : On souhaite obtenir la configuration suivante avec un **PIC16F877** :

RE2 : Sortie logique **RE1** : Sortie logique **RE0** : Entrée logique
RA5 : Sortie logique **RA4** : Entrée Logique **RA3** : Entrée analogique
RA2 : Entrée logique **RA1** : Entrée analogique **RA0** : Entrée analogique
Tension de référence **VREF = VDD – VSS = 5V** et fréquence du quartz égale à **12MHz**.

Programme en assembleur.

Remarque : Toutes les lignes de sorties des **PORTs** sont mises à zéro.

```

; Mise à zéro des registres de données des ports A et E
clrf  PORTA
clrf  PORTE

; Configuration des registres de directions des PORT A et E
; Accès aux registres TRISx (Banque mémoire 1)
bsf   STATUS,RP0   ; RP0 = 1
bcf   STATUS,RP1   ; RP1 = 0

; Configuration des registres de directions
; Configuration du PORTA X X S E E E E E
movlw B'11011111' ; valeur binaire          1 1 0 1 1 1 1 1
movwf TRISA
; Configuration du PORTE 0 0 0 0 0 S S E
movlw B'00000001' ; valeur binaire          0 0 0 0 0 0 0 1
movwf TRISE

; Configuration du registre ADCON1 Page 1
; ADFM = 1 justification à droite du résultat
; PCFG 3:0 0100 => RE3:RE0 Type D comme Digitale
;                => RA5 : D, RA3 : A comme Analogique
;                => RA2 : D, RA1 : A et RA0 : A
movlw B'10000100' ; valeur binaire          1 0 0 0 0 1 0 0
movwf ADCON1

; Retour en banque mémoire 0
bcf   STATUS,RP0   ; RP0 = 0
bcf   STATUS,RP1   ; RP1 = 0

; Configuration du registre ADCON0 Page 0
; ADCS1 et ADSC0 = 1 0 Fréquence Max 20MHz
; ADON = 1 Mise en route du CAN
; 0 pour les autres bits
movlw B'10000001' ; valeur binaire          1 0 0 0 0 0 0 1
movwf ADCON0

; Conversion du canal RA3
; Sélection du canal 3 avec les bits CHS2, CHS1 et CHS0 : 0 1 1
; GO/DONE = 1 Lancement d'une conversion
bcf   ADCON0,CHS2
bsf   ADCON0,CHS1
bsf   ADCON0,CHS0
bsf   ADCON0,GO   ; Déclenchement de la conversion
ATT  btfsc ADCON0,GO_DONE ; attendre la fin de conversion
goto  ATT
; fin de conversion, lecture du résultat
movf  ADRESH,0   ; Partie haute
movwf RES_HAUT,1
bsf   STATUS,RP1 ; Passage en page 1
movf  ADRESL,0   ; Partie basse
bsf   STATUS,RP0 ; Passage en page 0
movwf RES_BAS,1 ;

```


11) LES TIMERS.

Les **PICs 16F87x** disposent de **3 timers** :

- **Le timer 0 (8 bits)**: Il peut être incrémenté par des impulsions extérieures via la broche (**TOCKI/ RA4**) ou par l'horloge interne (**Fosc/4**).
- **Le timer 1 (16 bits)**: Il peut être incrémenté soit par l'horloge interne, par des impulsions sur la broche **T1CKI/RC0** ou par un oscillateur (**RC ou quartz**) connecté sur les broches **TOSO/RCO** et **TIOSI/RC1**.
- **Le timer 2 (8 bits)** : Il est incrémenté par l'horloge interne, celle peut être pré divisée.

Tous ces **timers** peuvent déclencher une interruption interne, si ils ont été autorisés.

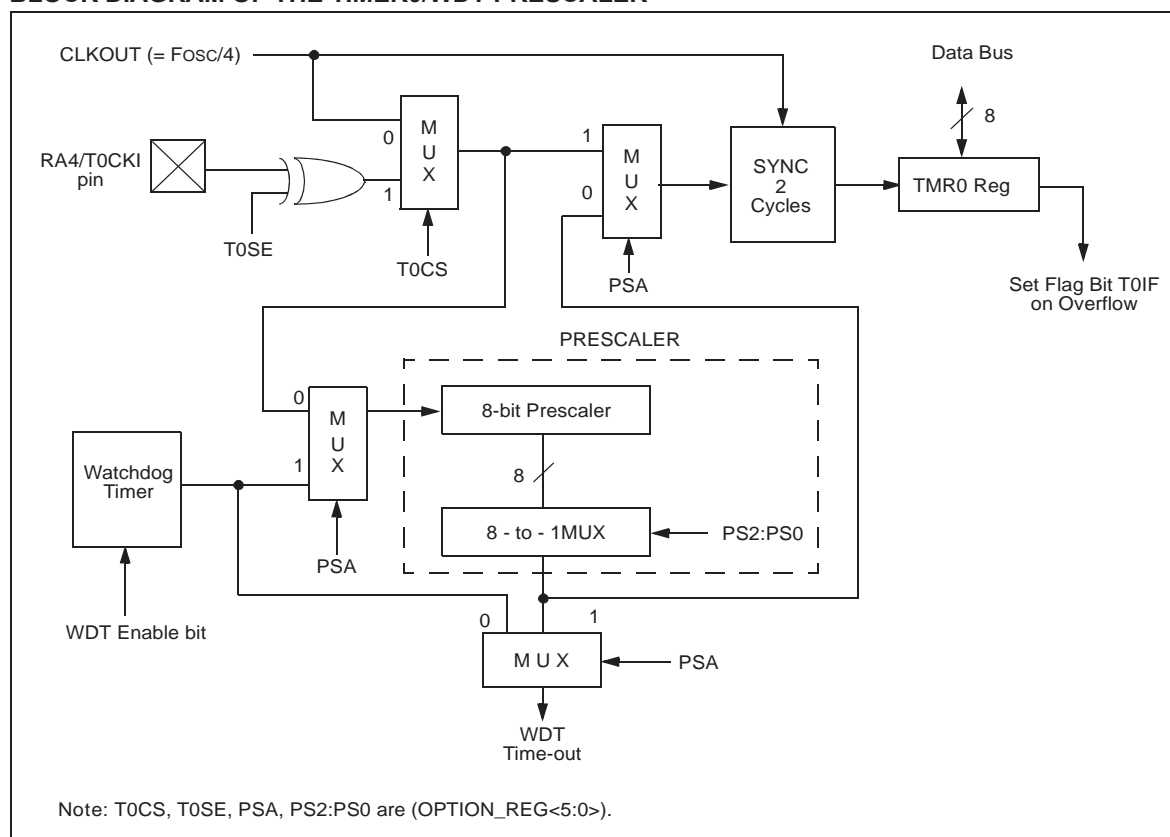
11.1) Le timer 0.

11.1.1) Présentation :

C'est le plus ancien des **timers** implantés dans les **PICs**, son ancienne appellation était **RTC**, pour **Real Time Clock** (horloge temps réelle). On peut se servir de celui-ci pour générer des événements périodiques, comme le rafraîchissement d'afficheurs multiplexés ou l'incrémentation de variables (secondes, minutes).

Celui-ci est incrémenté soit par l'horloge interne (**Fosc/4**) ou par une horloge appliquée sur la broche **TOCKI/ RA4**.

BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

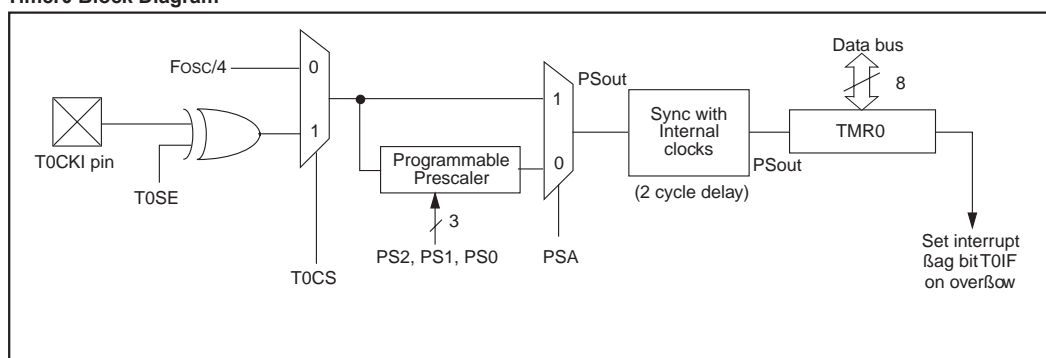


Comme on peut le constater sur ce schéma le **timer 0** partage avec le chien de garde **Watchdog** le pré diviseur. Celui-ci est affecté à l'un ou à l'autre, suivant la valeur du bit **PSA (0 : Timer0 et 1 : chien de garde)**.

LES MICROCONTROLEURS PIC 16F87X

On peut obtenir un schéma simplifié du fonctionnement du **timer0** sans le chien de garde.

Timer0 Block Diagram



11.1.2) Fonctionnement :

Le bit **T0CS** permet de choisir l'horloge, interne (**Fosc/4**) ou externe **TOCKI/RA4**. Dans ce dernier cas l'incrémentation du **timer 0** peut se faire soit sur front montant ou descendant suivant la valeur du bit **TOSE**. Le bit **PSA** de choisir si horloge permet de pré diviser l'horloge d'un rapport allant de **2 à 256**. La valeur de pré division est fixée par les bits **PS2, PS1 et PS0**.

Quand le contenu du **timer 0** passe de **FF** à **00** le bit **TOIF** passe à **1** pour signaler un débordement, si le bit **TOIE** est à **1** alors une interruption **timer 0** est déclenchée.

Le contenu du **timer 0** peut être modifié à tout instant, à une condition près, la nouvelle valeur inscrite dans le registre **TMR0** sera prise en compte après **3** cycles machines.

Remarque importante : Lorsque le bit **TOIF** passe à **1** lors du passage de la valeur **FFh** à **00h** du registre **TMR0**, il doit être remis à **0** de façon logicielle par une instruction du type : `bcf INTCON, TOIF`

11.1.3) Configuration et registres associés :

La configuration du **timer 0** passe par les registres **TMR0** (adresse **01h**), **OPTION_REG** (adresse **81h page 1**) et **INTCON** (adresse **0Bh** : toutes les pages).

OPTION_REG REGISTER (@ 81h or 181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0
bit 7	RBPU							
bit 6	INTEDG							
bit 5	T0CS: TMR0 Clock Source Select bit 1 = Transition on TOCKI pin 0 = Internal instruction cycle clock (CLKOUT)							
bit 4	T0SE: TMR0 Source Edge Select bit 1 = Increment on high-to-low transition on TOCKI pin 0 = Increment on low-to-high transition on TOCKI pin							
bit 3	PSA: Prescaler Assignment bit 1 = Prescaler is assigned to the WDT 0 = Prescaler is assigned to the Timer0 module							
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					

Legend:
 R = Readable bit W = Writable bit U = Unimplemented bit, read as '0'
 - n = Value at POR '1' = Bit is set '0' = Bit is cleared x = Bit is unknown

REGISTERS ASSOCIATED WITH TIMER0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
01h, 101h	TMR0	Timer0 Module's Register								xxxx xxxx	uuuu uuuu
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF	0000 000x	0000 000u
81h, 181h	OPTION_REG	RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0	1111 1111	1111 1111

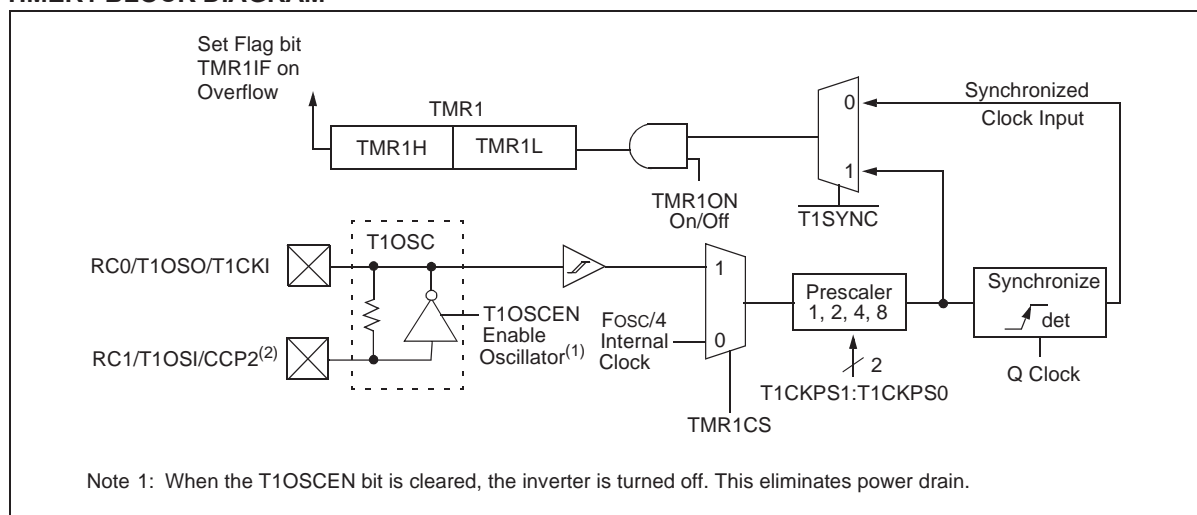
Legend: x = unknown, u = unchanged, - = unimplemented locations read as '0'.
 Shaded cells are not used by Timer0.

11.2) Le timer 1.

11.2.1) Présentation :

Il fonctionne sur le même principe que le **timer 0**, mais il est plus moderne dans sa conception. C'est un compteur **16 bits**.

TIMER1 BLOCK DIAGRAM



11.2.2) Fonctionnement :

Le bit **TMR1CS** permet de choisir l'horloge soit interne (**Fosc/4**), externe **T1CKI** ou un oscillateur à quartz connecté sur les broches **T1OSO** et **T1OSI**.

Les bits **T1CKPS1** et **T1CKPS0** permettent de choisir la valeur de la pré division à appliquer à l'horloge choisie, de **1** à **8**.

Le bit **T1SYNC** permet de choisir si l'horloge de sortie du pré diviseur doit être synchrone avec l'horloge du microcontrôleur. Dans le cas où l'on choisit l'horloge interne **Fosc/4**, il n'est pas nécessaire de la synchroniser.

Le bit **TMR1ON** active ou désactive le **timer 1**. Si ce bit est à **1** alors le **timer 1** est en fonctionnement et les registres **TMR1H :TMR1L** sont incrémentés à chaque coup d'horloge.

Quand le contenu du **timer 1** passe de **FFFF** à **0000** le bit **TMR1IF** passe à **1** pour signaler un débordement, de plus si le bit **TMR1IE** est à **1** alors une interruption **timer 1** est déclenchée.

Remarque importante : Lorsque le bit **TMR1IF** passe à **1** lors du passage de la valeur **FFFF** à **0000** des registres **TMR1H :TMR1L**, il doit être remis à **0** de façon logicielle par une instruction du type :

```
bcf PIR1,TMR1IF
```

LES MICROCONTROLEURS PIC 16F87X

11.2.3) Configuration et registres associés :

La configuration du **timer 1** passe par les registres : **PIR1** (adresse **0Ch**), **PIE1** (adresse **8Ch page 1**) le registres **TMR1L** et **TMR1H** (adresses **0Eh** et **0Fh**), **T1CON** (adresse **10h page 0**) et **INTCON** (adresse **0Bh** : toutes les pages).

T1CON: TIMER1 CONTROL REGISTER (ADDRESS 10h)

U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	
—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	
bit 7								bit 0

- bit 7-6 Unimplemented: Read as '0'
- bit 5-4 T1CKPS1:T1CKPS0: Timer1 Input Clock Prescale Select bits
11 = 1:8 Prescale value
10 = 1:4 Prescale value
01 = 1:2 Prescale value
00 = 1:1 Prescale value
- bit 3 T1OSCEN: Timer1 Oscillator Enable Control bit
1 = Oscillator is enabled
0 = Oscillator is shut-off (the oscillator inverter is turned off to eliminate power drain)
- bit 2 T1SYNC: Timer1 External Clock Input Synchronization Control bit
When TMR1CS = 1:
1 = Do not synchronize external clock input
0 = Synchronize external clock input
When TMR1CS = 0:
This bit is ignored. Timer1 uses the internal clock when TMR1CS = 0.
- bit 1 TMR1CS: Timer1 Clock Source Select bit
1 = External clock from pin RC0/T1OSO/T1CKI (on the rising edge)
0 = Internal clock (Fosc/4)
- bit 0 TMR1ON: Timer1 On bit
1 = Enables Timer1
0 = Stops Timer1

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

REGISTERS ASSOCIATED WITH TIMER1 AS A TIMER/COUNTER

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh, 18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
0Eh	TMR1L	Holding Register for the Least Significant Byte of the 16-bit TMR1 Register							xxxx xxxx	uuuu uuuu	
0Fh	TMR1H	Holding Register for the Most Significant Byte of the 16-bit TMR1 Register							xxxx xxxx	uuuu uuuu	
10h	T1CON	—	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON	--00 0000	-- uu uuuu

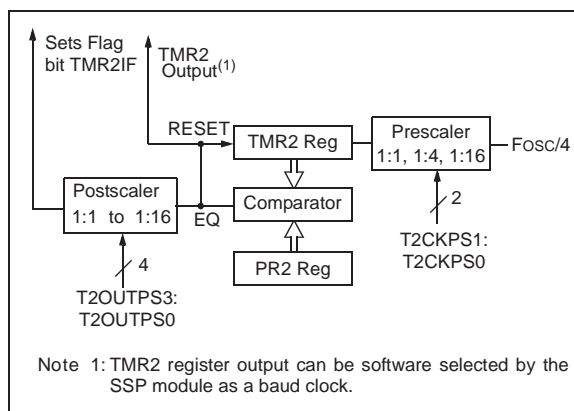
Legend: x = unknown, u = unchanged, - = unimplemented, read as '0'. Shaded cells are not used by the Timer1 module.
 Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

11.3) Le timer 2.

11.3.1) Présentation :

C'est un **timer 8 bits**, son horloge ne peut être que l'horloge interne divisée par **4 (Fosc/4)**

TIMER2 BLOCK DIAGRAM



11.3.2) Fonctionnement :

Il est incrémenté par l'horloge interne (**Fosc/4**) pré divisée ou non. Les bits **T2CKPS1** et **T2KPS0** permettent de choisir la valeur de la pré division (**1,4 ou 16**).

Le contenu du registre incrémenté **TMR2** et il est comparé au registre **PR2**, si ces deux registres sont égaux alors une impulsion d'horloge est générée et le contenu de **TMR2** est remis à **00h**. Celle-ci peut servir d'horloge pour piloter les liaisons **I2C** et **SPI** du module **SSP** ou encore être divisée par un post diviseur appelé : **POSTSCALER**.

Son rapport de division peut être de **1 à 16**. Les bits **T2OUTPS0**, **T2OUTPS1**, **T2OUTPS2**, et **T2OUTPS3**, permettent de choisir la valeur de la post division **1, 2, 3, 4, 5 16**.

Quand la sortie du post diviseur passe à **1** le bit **TMR2IF** est positionné, celui-ci peut déclencher une interruption si celle-ci a été autorisée (Bit **TMR2IE** à **1** du registre **PIE1**).

Remarque importante : Lorsque le bit **TMR2IF** passe à **1** lorsqu'une impulsion est générée en sortie du **POSTSCALER**, il doit être remis à **0** de façon logicielle par une instruction du type : `bcf PIR1, TMR2IF`

11.3.3) Configuration et registres associés :

La configuration du **timer 2** passe par les registres : **PIR1** (adresse **0Ch**), **PIE1** (adresse **8Ch page 1**), **TMR2** (adresse **11h page 0**), **PR2** (adresse **92h page 1**), **T2CON** (adresse **12h page 0**) et **INTCON** (adresse **0Bh** : toutes les pages).

T2CON: TIMER2 CONTROL REGISTER (ADDRESS 12h)

	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	TOUTPS3	TOUTPS2	TOUTPS1	TOUTPS0	TMR2ON	T2CKPS1	T2CKPS0	
bit 7							bit 0	

- bit 7 Unimplemented: Read as '0'
- bit 6-3 TOUTPS3:TOUTPS0: Timer2 Output Postscale Select bits
 - 0000 = 1:1 Postscale
 - 0001 = 1:2 Postscale
 - 0010 = 1:3 Postscale
 -
 - 1111 = 1:16 Postscale
- bit 2 TMR2ON: Timer2 On bit
 - 1 = Timer2 is on
 - 0 = Timer2 is off
- bit 1-0 T2CKPS1:T2CKPS0: Timer2 Clock Prescale Select bits
 - 00 = Prescaler is 1
 - 01 = Prescaler is 4
 - 1x = Prescaler is 16

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

11.4) Les modules CCP1 et CCP2 (C.C.P. :Capture Compare Pwm).

Ces deux modules peuvent fonctionner dans l'un des trois modes ci-dessous :

- Mode **capture (CAPTURE)** : Ce mode permet en outre d'effectuer des mesures de temps.
- Mode **comparaison (COMPARE)** : Ce mode permet en outre de générer des événements périodiques.
- Mode **PWM (PULSE WITH MODULATION)**: Ce mode permet de générer des signaux dont le rapport cyclique est variable.

Ces modules sont associés aux broches **RC2/CCP1** et **RC1/T1OSI/CCP2**. Suivant le mode choisit, les **timers 1** ou **2** vont être utilisés. Les modes **Capture** et **Comparaison** utilise le **timer 1**, tandis que le mode **PWM** utilise le **timer 2**.

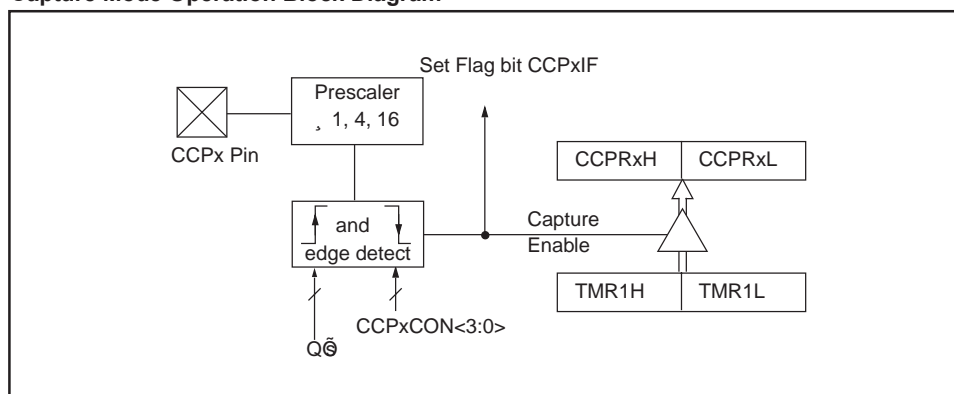
11.4.1) Le mode CAPTURE :

Il mémorise la valeur du **timer 1** dans les registres **CCP1R1H : CCP1R1L** ou **CCP1R2H : CCP1R2L** quand un événement se produit sur une des broches **CCP1** ou **CCP2**.

Cette mémorisation peut avoir lieu :

- Tous les fronts montants.
- Tous les fronts descendants.
- Tous les **4** fronts montants.
- Tous les **16** fronts montants.

Capture Mode Operation Block Diagram



12) LA LIAISON SERIE USART OU SCI (SERIAL COMMUNICATION INTERFACE).

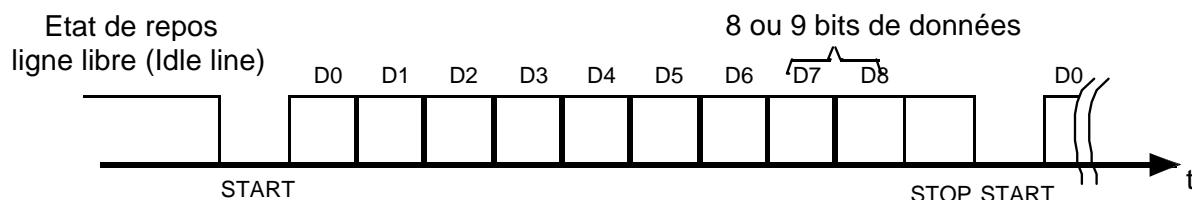
12.1) Présentation :

Remarque : La liaison **USART** du **PIC** peut fonctionner soit en mode synchrone ou asynchrone, seul le mode asynchrone sera étudié.

La liaison série **SCI** est une interface série asynchrone de type **START / STOP**. Elle permet d'effectuer des communications avec d'autres systèmes ou objets techniques sur de longues distances (quelques mètres à quelques kilomètres).

Elle dispose des fonctionnalités suivantes:

- Fonctionnement en Full Duplex, c'est à dire émission et réception de données en même temps.
- Transmission et réception de données (compatibles avec la norme **RS232** en utilisant une fonction d'adaptation de niveaux).
- Contrôle des erreurs de transmission et de réception.
- Transmission sur 8 ou 9 bits.
- Mode réveil automatique lors de la réception de signaux valides.
- 4 Sources d'interruptions possibles.



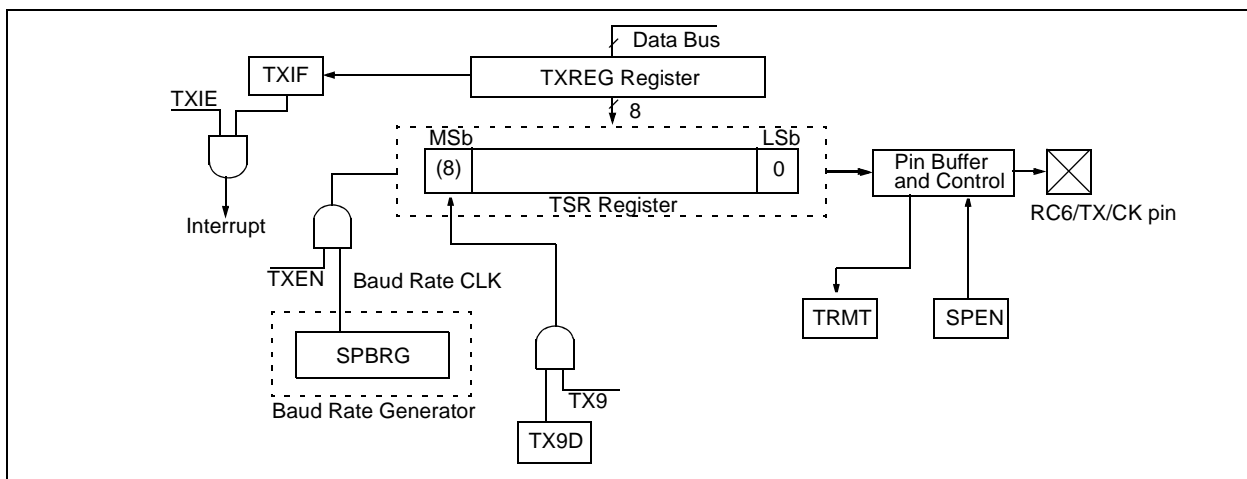
Cette interface est composée de 3 fonctions :

- La transmission.
- La réception.
- Le générateur d'horloge (choix de la vitesse de transmission et de réception).

12.2) La transmission :

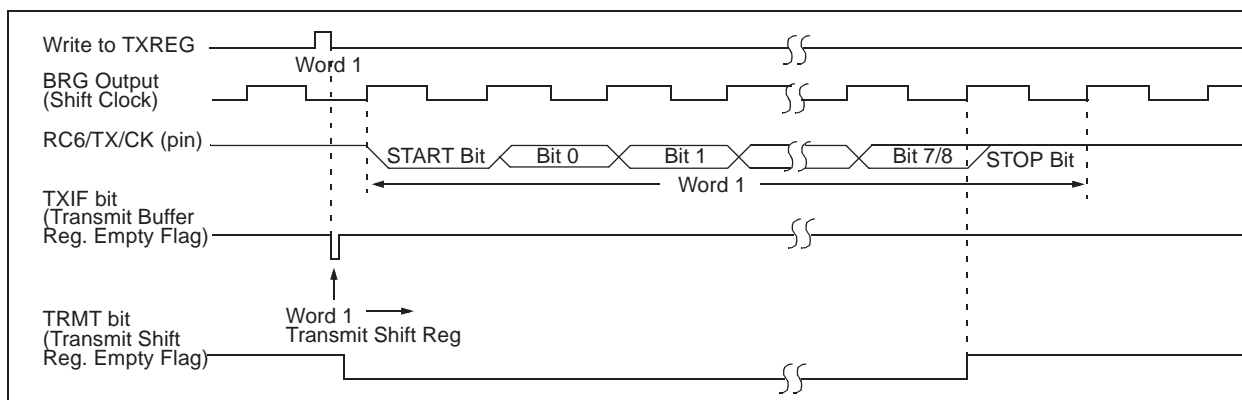
12.2.1) Présentation :

USART TRANSMIT BLOCK DIAGRAM



12.2.2) Fonctionnement :

ASYNCHRONOUS MASTER TRANSMISSION



Cette fonction utilise un registre à décalage pour transmettre les **8** ou **9** bits de l'information du registre **TXREG**.

Pour que cette fonction soit opérationnelle, il faut que la broche **RC6** du **PORTC** soit configurée en sortie, positionner à **1** le bit **SPEN** du registre **RCSTA** et le bit **TXEN** du registre **TXSTA**.

Dans le cas où l'on utilise une transmission sur **9** bits (Bit de parité par exemple), il faut autoriser la transmission sur **9** bits via le bit **TX9** du registre **TXSTA**, la valeur du neuvième bit doit être mise dans le bit **TX9D** du registre **TXSTA**.

Avant de transmettre une information, il faut s'assurer que le registre de transmission soit libre à travers le bit **TXIF** (**1** libre et **0** occupée). Le bit **TRMT** du registre **TXSTA** indique si la transmission est complètement terminée (**1** terminée et **0** occupée).

LES MICROCONTROLEURS PIC 16F87X

12.2.3) Configuration et registres associés :

La configuration de la transmission de la **SCI** passe par les registres : **PIR1** (adresse **0Ch**), **PIE1** (adresse **8Ch page 1**), **RCSTA** (adresse **18h page 0**), **TXREG** (adresse **19h page 0**), **TXSTA** (adresse **98h page 1**), **SPBRG** (adresse **99h page 1**) et **INTCON** (adresse **0Bh** : toutes les pages).

REGISTERS ASSOCIATED WITH ASYNCHRONOUS TRANSMISSION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN		FERR	OERR	RX9D	0000 -00x	0000 -00x
19h	TXREG	USART Transmit Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC		BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous transmission.

Note 1: Bits PSPIE and PSPIF are reserved on the PIC16F873/876; always maintain these bits clear.

Le registre le plus important pour la transmission c'est le registre **TXSTA**.

TXSTA: TRANSMIT STATUS AND CONTROL REGISTER (ADDRESS 98h)

	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R-1	R/W-0
	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D
	bit 7							bit 0

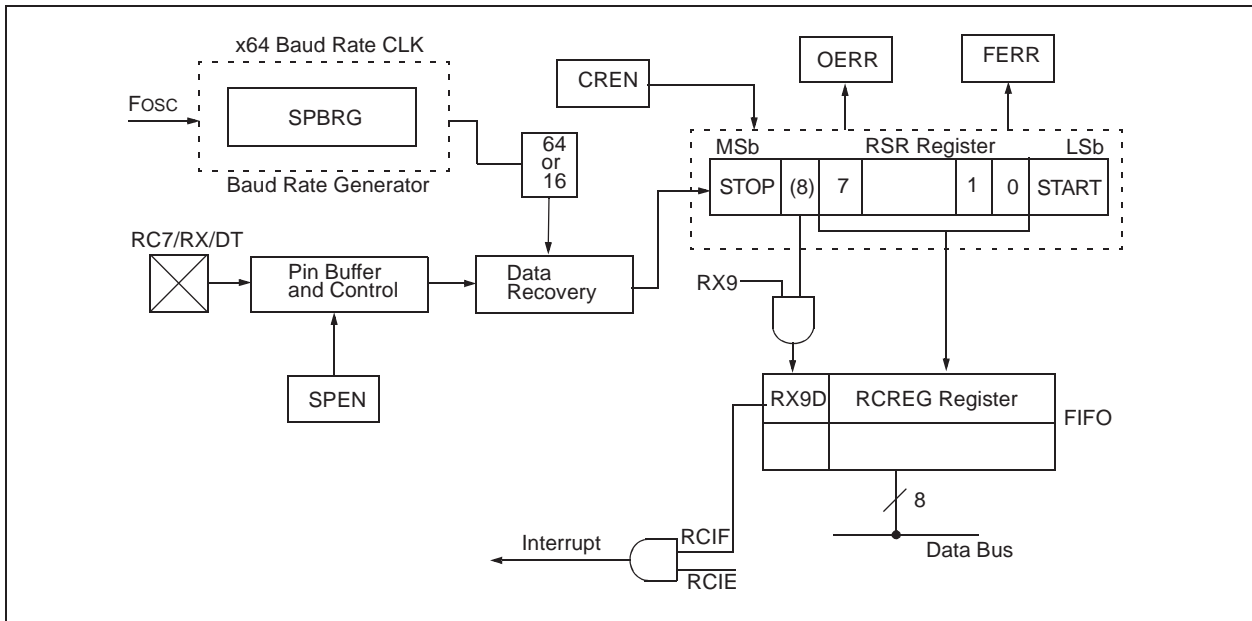
- bit 7 CSRC: Clock Source Select bit
Asynchronous mode:
Don't care
Synchronous mode:
1 = Master mode (clock generated internally from BRG)
0 = Slave mode (clock from external source)
- bit 6 TX9: 9-bit Transmit Enable bit
1 = Selects 9-bit transmission
0 = Selects 8-bit transmission
- bit 5 TXEN: Transmit Enable bit
1 = Transmit enabled
0 = Transmit disabled
- bit 4 Note: SREN/CREN overrides TXEN in SYNC mode.
SYNC: USART Mode Select bit
1 = Synchronous mode
0 = Asynchronous mode
- bit 3 Unimplemented: Read as '0'
- bit 2 BRGH: High Baud Rate Select bit
Asynchronous mode:
1 = High speed
0 = Low speed
Synchronous mode:
Unused in this mode
- bit 1 TRMT: Transmit Shift Register Status bit
1 = TSR empty
0 = TSR full
- bit 0 TX9D: 9th bit of Transmit Data, can be parity bit

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

12.3) La réception :

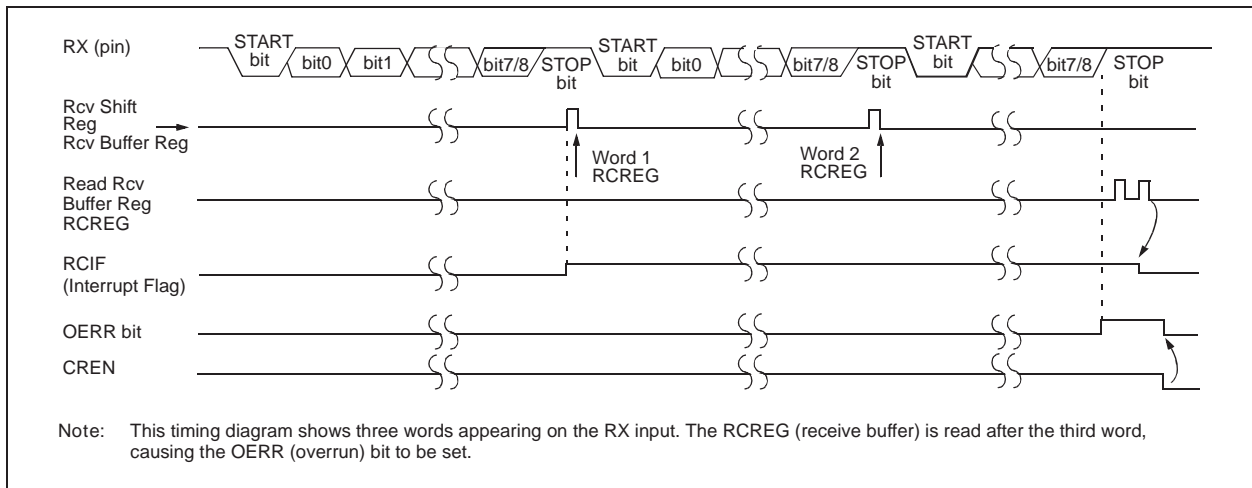
12.3.1) Présentation :

USART RECEIVE BLOCK DIAGRAM



12.3.2) Fonctionnement :

ASYNCHRONOUS RECEPTION



Cette fonction utilise un registre **RSR** à décalage pour les **8** ou **9** bits de l'information à recevoir, une fois la réception terminée la valeur est stockée dans le registre **RCREG**.

Pour que cette fonction soit opérationnelle, il faut que la broche **RC7** du **PORTC** soit configurée en entrée et positionnée à **1** le bit **SPEN** du registre **RCSTA**.

Dans le cas où l'on utilise une réception sur **9** bits, il faut autoriser la réception sur **9** bits via le bit **RX9** du registre **RCSTA**, la valeur du neuvième bit est récupérée dans le bit **RX9D** du registre **RCSTA**.

Avant de lire une information dans le registre **RCREG**, il faut s'assurer que l'information est bien reçue en testant le bit **RCIF** (1 Donnée reçue), ce bit est remis à **0** lors de la lecture du registre **RCREG**.

Les bits **FERR** et **OERR** peuvent indiquer respectivement une erreur de format et une erreur « **over run** ».

LES MICROCONTROLEURS PIC 16F87X

12.3.3) Configuration et registres associés :

La configuration de la transmission de la **SCI** passe par les registres : **PIR1** (adresse **0Ch**), **PIE1** (adresse **8Ch page 1**), **RCSTA** (adresse **18h page 0**), **RCREG** (adresse **1Ah page 0**), **TXSTA** (adresse **98h page 1**), **SPBRG** (adresse **99h page 1**) et **INTCON** (adresse **0Bh** : toutes les pages).

REGISTERS ASSOCIATED WITH ASYNCHRONOUS RECEPTION

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
0Bh, 8Bh, 10Bh,18Bh	INTCON	GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	R0IF	0000 000x	0000 000u
0Ch	PIR1	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	0000 0000	0000 0000
18h	RCSTA	SPEN	RX9	SREN	CREN	—	FERR	OERR	RX9D	0000 -00x	0000 -00x
1Ah	RCREG	USART Receive Register								0000 0000	0000 0000
8Ch	PIE1	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE	0000 0000	0000 0000
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented locations read as '0'. Shaded cells are not used for asynchronous reception.

Note 1: Bits PSPIE and PSPIF are reserved on PIC16F873/876 devices; always maintain these bits clear.

Le registre le plus important pour la transmission c'est le registre **RCSTA**.

RCSTA: RECEIVE STATUS AND CONTROL REGISTER (ADDRESS 18h)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-0	R-x
SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D
bit 7				bit 0			

- bit 7 SPEN: Serial Port Enable bit
1 = Serial port enabled (configures RC7/RX/DT and RC6/TX/CK pins as serial port pins)
0 = Serial port disabled
- bit 6 RX9: 9-bit Receive Enable bit
1 = Selects 9-bit reception
0 = Selects 8-bit reception
- bit 5 SREN: Single Receive Enable bit
Asynchronous mode:
Don't care
Synchronous mode - master:
1 = Enables single receive
0 = Disables single receive
This bit is cleared after reception is complete.
Synchronous mode - slave:
Don't care
- bit 4 CREN: Continuous Receive Enable bit
Asynchronous mode:
1 = Enables continuous receive
0 = Disables continuous receive
Synchronous mode:
1 = Enables continuous receive until enable bit CREN is cleared (CREN overrides SREN)
0 = Disables continuous receive
- bit 3 ADDEN: Address Detect Enable bit
Asynchronous mode 9-bit (RX9 = 1):
1 = Enables address detection, enables interrupt and load of the receive buffer when RSR<8> is set
0 = Disables address detection, all bytes are received, and ninth bit can be used as parity bit
- bit 2 FERR: Framing Error bit
1 = Framing error (can be updated by reading RCREG register and receive next valid byte)
0 = No framing error
- bit 1 OERR: Overrun Error bit
1 = Overrun error (can be cleared by clearing bit CREN)
0 = No overrun error
- bit 0 RX9D: 9th bit of Received Data (can be parity bit, but must be calculated by user firmware)

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
- n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

LES MICROCONTROLEURS PIC 16F87X

12.4) Le générateur d'horloge:

12.4.1) Présentation et fonctionnement:

C'est lui qui fixe la vitesse de la réception et transmission de l'USART.

Il faut utiliser la formule ci-dessous pour calculer la valeur à mettre dans le registre **SPBRG**. Si la fréquence du quartz du microcontrôleur est supérieure à **10MHz**, il est conseillé de positionner le bit **BGRH** à **1**.

BAUD RATE FORMULA

SYNC	BRGH = 0 (Low Speed)	BRGH = 1 (High Speed)
0	(Asynchronous) Baud Rate = $F_{osc}/(64(X+1))$	Baud Rate = $F_{osc}/(16(X+1))$
1	(Synchronous) Baud Rate = $F_{osc}/(4(X+1))$	N/A

X = value in SPBRG (0 to 255)

Tableaux de vitesses :

BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 0)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	1.221	1.75	255	1.202	0.17	207	1.202	0.17	129
2.4	2.404	0.17	129	2.404	0.17	103	2.404	0.17	64
9.6	9.766	1.73	31	9.615	0.16	25	9.766	1.73	15
19.2	19.531	1.72	15	19.231	0.16	12	19.531	1.72	7
28.8	31.250	8.51	9	27.778	3.55	8	31.250	8.51	4
33.6	34.722	3.34	8	35.714	6.29	6	31.250	6.99	4
57.6	62.500	8.51	4	62.500	8.51	3	52.083	9.58	2
HIGH	1.221	-	255	0.977	-	255	0.610	-	255
LOW	312.500	-	0	250.000	-	0	156.250	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	0.300	0	207	0.3	0	191
1.2	1.202	0.17	51	1.2	0	47
2.4	2.404	0.17	25	2.4	0	23
9.6	8.929	6.99	6	9.6	0	5
19.2	20.833	8.51	2	19.2	0	2
28.8	31.250	8.51	1	28.8	0	1
33.6	-	-	-	-	-	-
57.6	62.500	8.51	0	57.6	0	0
HIGH	0.244	-	255	0.225	-	255
LOW	62.500	-	0	57.6	-	0

BAUD RATES FOR ASYNCHRONOUS MODE (BRGH = 1)

BAUD RATE (K)	Fosc = 20 MHz			Fosc = 16 MHz			Fosc = 10 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-	-	-	-
1.2	-	-	-	-	-	-	-	-	-
2.4	-	-	-	-	-	-	2.441	1.71	255
9.6	9.615	0.16	129	9.615	0.16	103	9.615	0.16	64
19.2	19.231	0.16	64	19.231	0.16	51	19.531	1.72	31
28.8	29.070	0.94	42	29.412	2.13	33	28.409	1.36	21
33.6	33.784	0.55	36	33.333	0.79	29	32.895	2.10	18
57.6	59.524	3.34	20	58.824	2.13	16	56.818	1.36	10
HIGH	4.883	-	255	3.906	-	255	2.441	-	255
LOW	1250.000	-	0	1000.000	-	0	625.000	-	0

BAUD RATE (K)	Fosc = 4 MHz			Fosc = 3.6864 MHz		
	KBAUD	% ERROR	SPBRG value (decimal)	KBAUD	% ERROR	SPBRG value (decimal)
0.3	-	-	-	-	-	-
1.2	1.202	0.17	207	1.2	0	191
2.4	2.404	0.17	103	2.4	0	95
9.6	9.615	0.16	25	9.6	0	23
19.2	19.231	0.16	12	19.2	0	11
28.8	27.798	3.55	8	28.8	0	7
33.6	35.714	6.29	6	32.9	2.04	6
57.6	62.500	8.51	3	57.6	0	3
HIGH	0.977	-	255	0.9	-	255
LOW	250.000	-	0	230.4	-	0

12.4.2) Configuration et registres associés :

La configuration du générateur d'horloge de la **SCI** passe par les registres : **TXSTA** (adresse **98h page 1**), **RCSTA** (adresse **18h page 0**) et **SPBRG** (adresse **99h page 1**).

REGISTERS ASSOCIATED WITH BAUD RATE GENERATOR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Value on: POR, BOR	Value on all other RESETS
98h	TXSTA	CSRC	TX9	TXEN	SYNC	—	BRGH	TRMT	TX9D	0000 -010	0000 -010
18h	RCSTA	SPEN	RX9	SREN	CREN	ADDEN	FERR	OERR	RX9D	0000 000x	0000 000x
99h	SPBRG	Baud Rate Generator Register								0000 0000	0000 0000

Legend: x = unknown, - = unimplemented, read as '0'. Shaded cells are not used by the BRG.

13) LES INTERRUPTIONS.

13.1) Présentation.

Le μC dispose de plusieurs sources d'interruptions.

- Une interruption externe, action sur la broche **INT/RB0**.
- Débordement du **TIMER0**.
- Changement d'état logique sur une des broches du **PORTB (RB4 à RB7)**.
- Une interruption d'un des périphériques (**PEIE**).
 - Fin de programmation d'une case mémoire de l'**EEPROM**.
 - Changement d'état sur le **PORTD (PSPIE)**.
 - Fin de conversion analogique numérique (**ADIE**).
 - Réception d'une information sur la liaison série (**RCIE**).
 - Fin d'émission d'une information sur la liaison série (**TXIE**).
 - Interruption **SPI** ou **I2C** du module **MSSP (SSPIE)**.
 - Interruption du registre de capture et/ou de comparaison 1 (**CCPI1E**).
 - Interruption du registre de capture et/ou de comparaison 2 (**CCPI2E**).
 - Débordement du **TIMER1 (TMR1E)**.
 - Débordement du **TIMER2 (TMR2E)**.
 - Collision de **BUS (BCLIE)**

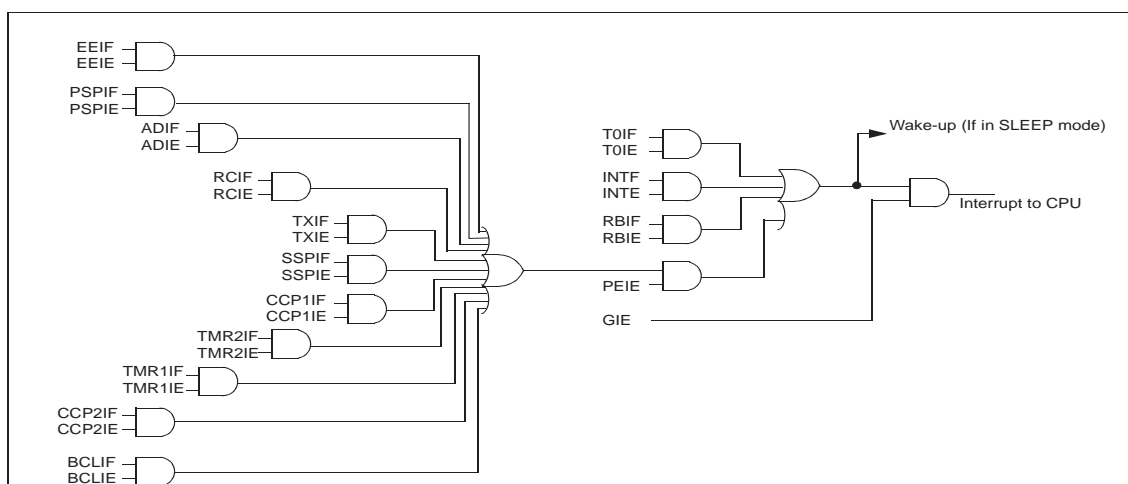
13.2) Fonctionnement.

Lors d'un événement dans un ou plusieurs des circuits périphériques (**ADC, EEPROM, USART-SCI, MSSP-I2C-SPI, TIMER1, TIMER2**) comme par exemple : la fin de conversion, la fin de programmation d'un octet dans l'**EEPROM**, la réception d'une information, la détection d'un front, etc... et si le bit de l'interruption concernée a été autorisée (**EEIE, PSPIE, ADIE, RCIE, TXIE, SSPIE, CCP1IE, TMR2IE, TMR1IE, CCP2IE** ou **BCLIE : Registres PIE1 et PIE2**) alors une interruption périphérique est déclenchée. Pour que celle-ci soit prise en compte il faut que le bit d'autorisation des interruptions périphériques soit positionné à **1 (PEIE)** ainsi que le bit **GIE** d'autorisation globale des interruptions du registre **INTCON**.

Pour qu'une interruption du type **TIMER0** ou **INT/RB0** ou **PORTB** soit prise en compte il suffit que le bit local d'autorisation d'interruption soit positionné à **1 (TOIE ou INTE ou RBIE)** ainsi que le bit **GIE** d'autorisation globale des interruptions du registre **INTCON**.

Dans ces conditions le programme en cours d'exécution est interrompu et le microcontrôleur exécute le programme d'interruption à partir de l'adresse **0x0004**. Au début de celui-ci il faut que le logiciel vérifie quel périphérique a déclenché l'interruption.

INTERRUPT LOGIC



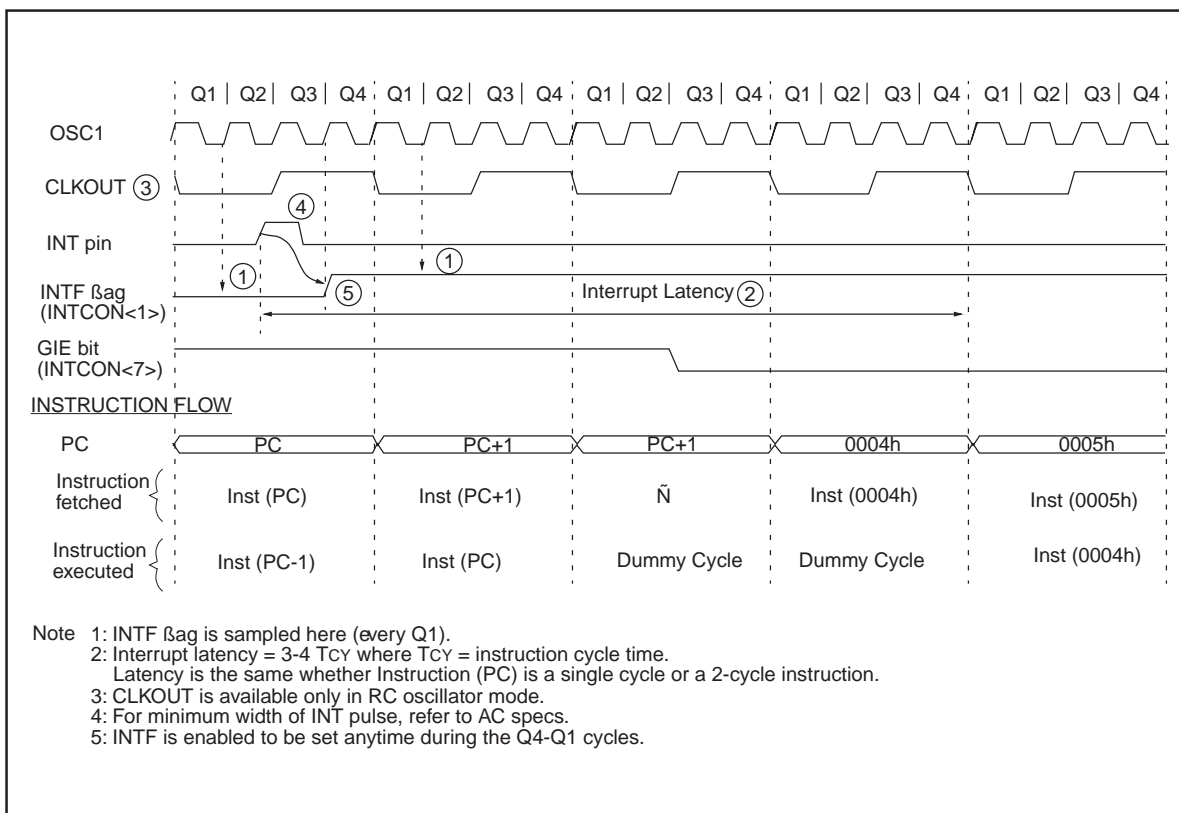
The following table shows which devices have which interrupts.

Device	TOIF	INTF	RBIF	PSPIF	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF	EEIF	BCLIF	CCP2IF
PIC16F876/873	Yes	Yes	Yes	—	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
PIC16F877/874	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

13.3 Déroulement d'une interruption.

Chronogramme de la prise en compte d'une interruption.

INT Pin and Other External Interrupt Timing



Au début d'une interruption le sous programme d'interruption doit sauvegarder le contexte et le restituer à la fin, c'est à dire les valeurs des registres **W**, **PCLATH** et **STATUS**.

Cela permet au processus interrompu de retrouver ses registres intacts.

Pour respecter ce principe il faut ajouter au début du sous programme d'interruption quelques instructions pour sauvegarder les registres **W**, **PCLATH** et **STATUS**. A la fin du sous programme on ajoute des instructions pour restaurer ces valeurs.

SAVING STATUS, W, AND PCLATH REGISTERS IN RAM

```

MOVWF  W_TEMP           ;Co py W to TEMP register
SWAPF  STATUS,W         ;Swap status to be saved into W
CLRF   STATUS           ;ba nk 0, regardless of current bank, Clears IRP,RP1,          RP0
MOVWF  STATUS_TEMP     ;Sa ve status to bank zero STATUS_TEMP register
MOVF   PCLATH, W       ;On ly required if using pages 1, 2 and/or 3
MOVWF  PCLATH_TEMP     ;Sa ve PCLATH into W
CLRF   PCLATH          ;Pa ge zero, regardless of current page
:
: (ISR)                 ;(I nsert user code here)
:
MOVF   PCLATH_TEMP, W  ;Re store PCLATH
MOVWF  PCLATH          ;Move W into PCLATH
SWAPF  STATUS_TEMP,W  ;Swap STATUS_TEMP register into W
:                          ;(s ets bank to original state)
MOVWF  STATUS          ;Move W into STATUS register
SWAPF  W_TEMP,F        ;Swap W_TEMP
SWAPF  W_TEMP,W        ;Swap W_TEMP into W
    
```


13.4) Configuration et registres associés :

- Le registre **OPTION** permet de choisir le type de front pour l'interruption **INT/RB0**.

OPTION_REG REGISTER (ADDRESS 81h, 181h)

	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
	RBP	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7								bit 0

bit 6 **INTEDG:** Interrupt Edge Select bit
 1 = Interrupt on rising edge of RB0/INT pin
 0 = Interrupt on falling edge of RB0/INT pin

- Le registre **INTCON** permet d'autoriser les interruptions globales (**GIE**), les interruptions des périphériques (**PEIE**), L'interruption **TIMER0 (TOIE)**, l'interruption extérieure (**INT/RB0**), l'interruption de changement d'état du **PORTB (RBIE)** et les indicateurs associés des interruptions (**TIMER0, INT/RB0** et du changement d'état du **PORTB : RBIF**).

INTCON REGISTER (ADDRESS 0Bh, 8Bh, 10Bh, 18Bh)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-x
	GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7								bit 0

bit 7 **GIE:** Global Interrupt Enable bit
 1 = Enables all unmasked interrupts
 0 = Disables all interrupts

bit 6 **PEIE:** Peripheral Interrupt Enable bit
 1 = Enables all unmasked peripheral interrupts
 0 = Disables all peripheral interrupts

bit 5 **TOIE:** TMR0 Overflow Interrupt Enable bit
 1 = Enables the TMR0 interrupt
 0 = Disables the TMR0 interrupt

bit 4 **INTE:** RB0/INT External Interrupt Enable bit
 1 = Enables the RB0/INT external interrupt
 0 = Disables the RB0/INT external interrupt

bit 3 **RBIE:** RB Port Change Interrupt Enable bit
 1 = Enables the RB port change interrupt
 0 = Disables the RB port change interrupt

bit 2 **TOIF:** TMR0 Overflow Interrupt Flag bit
 1 = TMR0 register has overflowed (must be cleared in software)
 0 = TMR0 register did not overflow

bit 1 **INTF:** RB0/INT External Interrupt Flag bit
 1 = The RB0/INT external interrupt occurred (must be cleared in software)
 0 = The RB0/INT external interrupt did not occur

bit 0 **RBIF:** RB Port Change Interrupt Flag bit
 1 = At least one of the RB7:RB4 pins changed state; a mismatch condition will continue to set the bit. Reading PORTB will end the mismatch condition and allow the bit to be cleared (must be cleared in software).
 0 = None of the RB7:RB4 pins have changed state

LES MICROCONTROLEURS PIC 16F87X

- Les registres d'autorisations des interruptions périphériques **PIE1** et **PIE2**.

PIE1 REGISTER (ADDRESS 8Ch)

	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
	PSPIE ⁽¹⁾	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
bit 7								bit 0

- bit 7 PSPIE⁽¹⁾: Parallel Slave Port Read/Write Interrupt Enable bit
 1 = Enables the PSP read/write interrupt
 0 = Disables the PSP read/write interrupt
- bit 6 ADIE: A/D Converter Interrupt Enable bit
 1 = Enables the A/D converter interrupt
 0 = Disables the A/D converter interrupt
- bit 5 RCIE: USART Receive Interrupt Enable bit
 1 = Enables the USART receive interrupt
 0 = Disables the USART receive interrupt
- bit 4 TXIE: USART Transmit Interrupt Enable bit
 1 = Enables the USART transmit interrupt
 0 = Disables the USART transmit interrupt
- bit 3 SSPIE: Synchronous Serial Port Interrupt Enable bit
 1 = Enables the SSP interrupt
 0 = Disables the SSP interrupt
- bit 2 CCP1IE: CCP1 Interrupt Enable bit
 1 = Enables the CCP1 interrupt
 0 = Disables the CCP1 interrupt
- bit 1 TMR2IE: TMR2 to PR2 Match Interrupt Enable bit
 1 = Enables the TMR2 to PR2 match interrupt
 0 = Disables the TMR2 to PR2 match interrupt
- bit 0 TMR1IE: TMR1 Overflow Interrupt Enable bit
 1 = Enables the TMR1 overflow interrupt
 0 = Disables the TMR1 overflow interrupt

PIE2 REGISTER (ADDRESS 8Dh)

	U-0	R/W-0	U-0	R/W-0	R/W-0	U-0	U-0	R/W-0
	—	Reserved	—	EEIE	BCLIE	—	—	CCP2IE
bit 7								bit 0

- bit 7 Unimplemented: Read as '0'
- bit 6 Reserved: Always maintain this bit clear
- bit 5 Unimplemented: Read as '0'
- bit 4 EEIE: EEPROM Write Operation Interrupt Enable
 1 = Enable EE Write Interrupt
 0 = Disable EE Write Interrupt
- bit 3 BCLIE: Bus Collision Interrupt Enable
 1 = Enable Bus Collision Interrupt
 0 = Disable Bus Collision Interrupt
- bit 2-1 Unimplemented: Read as '0'
- bit 0 CCP2IE: CCP2 Interrupt Enable bit
 1 = Enables the CCP2 interrupt
 0 = Disables the CCP2 interrupt

LES MICROCONTROLEURS PIC 16F87X

- Le registre des indicateurs d'événements **PIR1**.

PIR1 REGISTER (ADDRESS 0Ch)

	R/W-0	R/W-0	R-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0
	PSPIF ⁽¹⁾	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
bit 7								bit 0

- bit 7 PSPIF⁽¹⁾: Parallel Slave Port Read/Write Interrupt Flag bit
 1 = A read or a write operation has taken place (must be cleared in software)
 0 = No read or write has occurred
- bit 6 ADIF: A/D Converter Interrupt Flag bit
 1 = An A/D conversion completed
 0 = The A/D conversion is not complete
- bit 5 RCIF: USART Receive Interrupt Flag bit
 1 = The USART receive buffer is full
 0 = The USART receive buffer is empty
- bit 4 TXIF: USART Transmit Interrupt Flag bit
 1 = The USART transmit buffer is empty
 0 = The USART transmit buffer is full
- bit 3 SSPIF: Synchronous Serial Port (SSP) Interrupt Flag
 1 = The SSP interrupt condition has occurred, and must be cleared in software before returning
 from the Interrupt Service Routine. The conditions that will set this bit are:
 SPI
 - A transmission/reception has taken place.
 I²C Slave
 - A transmission/reception has taken place.
 I²C Master
 - A transmission/reception has taken place.
 - The initiated START condition was completed by the SSP module.
 - The initiated STOP condition was completed by the SSP module.
 - The initiated Restart condition was completed by the SSP module.
 - The initiated Acknowledge condition was completed by the SSP module.
 - A START condition occurred while the SSP module was idle (Multi-Master system).
 - A STOP condition occurred while the SSP module was idle (Multi-Master system).
 0 = No SSP interrupt condition has occurred.
- bit 2 CCP1IF: CCP1 Interrupt Flag bit
 Capture mode:
 1 = A TMR1 register capture occurred (must be cleared in software)
 0 = No TMR1 register capture occurred
 Compare mode:
 1 = A TMR1 register compare match occurred (must be cleared in software)
 0 = No TMR1 register compare match occurred
 PWM mode:
 Unused in this mode
- bit 1 TMR2IF: TMR2 to PR2 Match Interrupt Flag bit
 1 = TMR2 to PR2 match occurred (must be cleared in software)
 0 = No TMR2 to PR2 match occurred
- bit 0 TMR1IF: TMR1 Overflow Interrupt Flag bit
 1 = TMR1 register overflowed (must be cleared in software)
 0 = TMR1 register did not overflow

Note 1: PSPIF is reserved on PIC16F873/876 devices; always maintain this bit clear.