

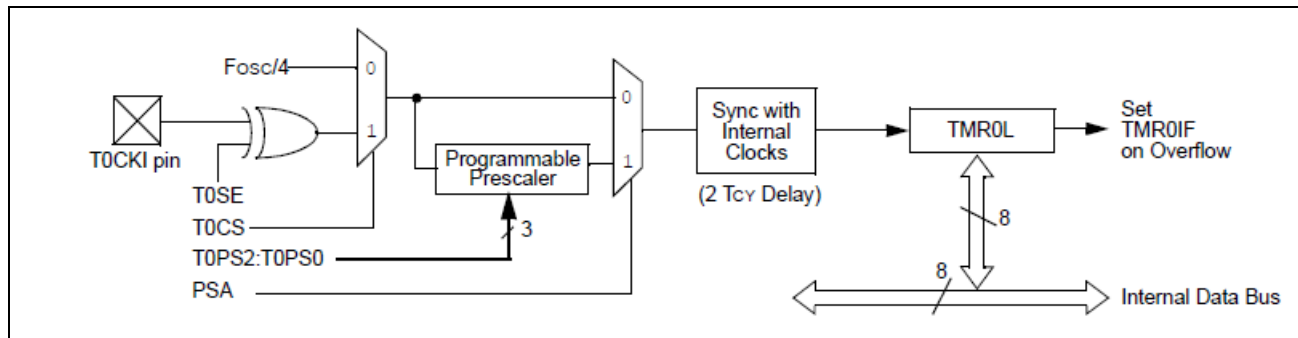
## 9 – Les Timers

La plupart des microcontrôleurs possèdent 1 ou plusieurs « Timer » 8bits, 16bits ou même 32bits. Ces dispositifs peuvent être employés pour compter des événements ou générer des signaux temporels.

### 1. Le principe du Timer0 :

Le « Timer » compte des événements suivant différentes sources ; (TMR0IF) indique le dépassement du registre, il peut déclencher une interruption sur dépassement (si TMR0IE=1).

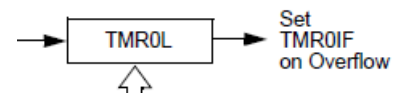
En 8 bits → 255+1 et en 16 bits → 65535+1.



### 2. Les Sources :

- L'horloge interne (Fosc/4) avec ou sans pré-diviseur (Programmable Prescaler).
- Une horloge spécialisée (non représenté ici).
- Des impulsions sur des broches d'entrée dédiées. (T0CKI) avec la possibilité du choix du front (T0SE).

### 3. Générateur de temps ou « Horloge Temps Réel » :



Il faut donc provoquer un dépassement pour activer un événement (une Interruption)

$Val = Max - (T/H)$  où  $Max = 256$  si en 8bits (ou 65 536 si en 16bits) ;  $T$  = la période souhaitée et  $H$  = la période de l'horloge d'entrée.

En chargeant « Val » dans le registre (TMR0L) une valeur équivalente à le registre basculera en dépassement à un nombre  $T$  d'impulsion d'Horloge.

Register 13-1: T0CON: Timer0 Control Register

R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1	R/W-1
TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0
bit 7							bit 0

bit 7 **TMR0ON:** Timer0 On/Off Control bit

1 = Enables Timer0  
0 = Stops Timer0

bit 6 **T08BIT:** Timer0 8-bit/16-bit Control bit

1 = Timer0 is configured as an 8-bit timer/counter  
0 = Timer0 is configured as a 16-bit timer/counter

bit 5 **T0CS:** Timer0 Clock Source Select bit

1 = Transition on T0CKI pin is clock (counter mode)  
0 = Internal instruction cycle is clock (timer mode)

bit 4 **T0SE:** Timer0 Source Edge Select bit

1 = Increment on high-to-low transition on T0CKI pin  
0 = Increment on low-to-high transition on T0CKI pin

bit 3 **PSA:** Timer0 Prescaler Assignment bit

1 = Timer0 prescaler is NOT assigned. Timer0 clock input bypasses prescaler.  
0 = Timer0 prescaler is assigned. Timer0 clock input comes from prescaler output.

bit 2-0 **T0PS2:T0PS0:** Timer0 Prescaler Select bits

These bits are ignored if  $PSA = 1$

111 = 1:256 prescale value

110 = 1:128 prescale value

101 = 1:64 prescale value

100 = 1:32 prescale value

011 = 1:16 prescale value

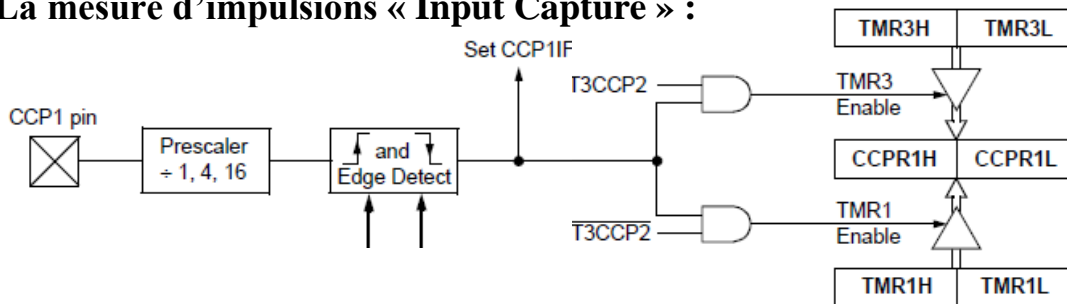
010 = 1:8 prescale value

001 = 1:4 prescale value

000 = 1:2 prescale value

## 9 – Les Timers

### 4. La mesure d'impulsions « Input Capture » :



Un compteur au choix issu du Timer1 ou du Timer3 tourne en permanence suivant la vitesse programmée. Les impulsions que l'on désire mesurer arrivent sur l'entrée (CCP1) ;

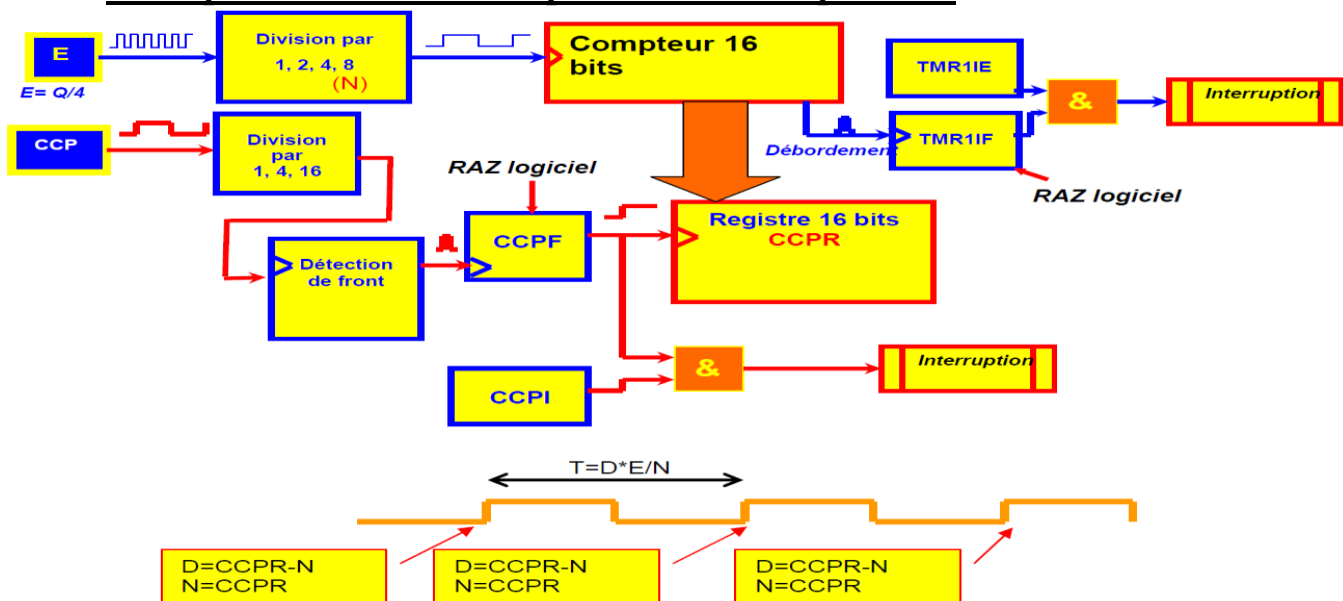
Elles peuvent être divisées par /1 /4 ou /16 (Prescaler).

On peut choisir le front sur lequel on fera la mesure (Edge Detect).

Une interruption peut marquer cet événement et le compteur sélectionné est recopié dans le registre (CCPR1).

En faisant la différence avec la valeur précédemment relevée dans CCPR1 La période du signal sur (CCP1) est égal au nombre de coups horloge du Timer de référence.

### 5. Principe de la mesure de la période d'une impulsion :

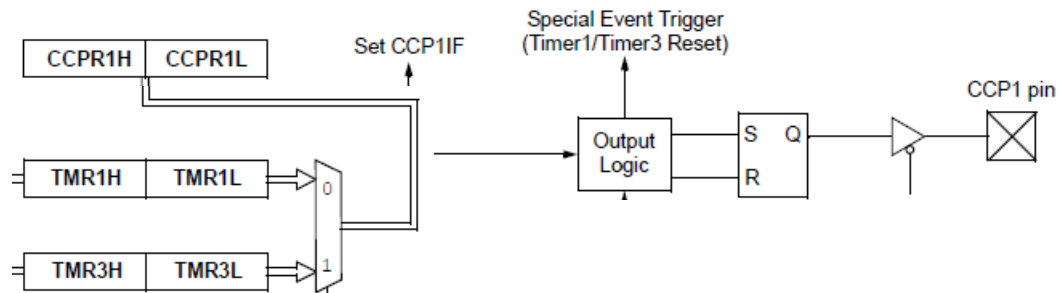


A chaque front montant TIMER1 est recopié dans CCPR, une interruption est générée, le sous programme d'IT calcul  $D = CCPR - N$  puis  $N = CCPR$ . D représente la période du signal d'entrée

```
// Programme test de la fonction capture. Le nombre d'impulsions comptées
// entre deux fronts montants de CCPR1 est rangée dans la variable duree
#include <p18f452.h>
unsigned int duree; // représente le comptage entre 2 fronts
// sous programme d'interruption
#pragma interrupt itcomp
void itcomp(void)
{
    unsigned static int ancien;
    if (PIR1bits.CCP1IF)
    {
        {duree=CCPR1-ancien; // 1'IT provient d'une capture
        ancien=CCPR1; } // comptage entre les deux front
        PIR1bits.CCP1IF=0; //efface le drapeau d'IT
    }
}
#pragma code interruption=0x8
void fontion (void)
{
    _asm goto itcomp _endasm
}
#pragma code
void main(void)
{
    // configure PORTC CCP1
    DDRCbits.RC2=1; // RC2/CCP1 en entree
    // configure le TIMER1
    T1CONbits.RD16=0; // TMR1 mode simple (pas de RW)
    T1CONbits.TMR1CS=0; // compte les impulsions sur internal clock
    T1CONbits.T1CKPS1=1; // prédiviseur = 1/8 periode sortie = 8uS
    T1CONbits.T1CKPS0=1;
    T1CONbits.T1SYNC=1; // pas de synchronisation sur sleep/Reset
    T1CONbits.TMR1ON=1; // TMR1 Activé
    // configure le mode capture sur le TIMER1 avec IT sur CCP1
    T3CONbits.T3CCP2=0; // mode comparaison entre TMR1 et CCPR1
    CCP1CON=0x05; // capture mode sur fronts montants
    PIE1bits.CCP1IE=1; // active IT sur mode capture/comparaison CCP1
    RCONbits.IPEN=1; // Interruption prioritaires activées
    INTCONbits.GIE=1; // Toutes les IT démasquées autorisées
    while(1) ;
}
}
```

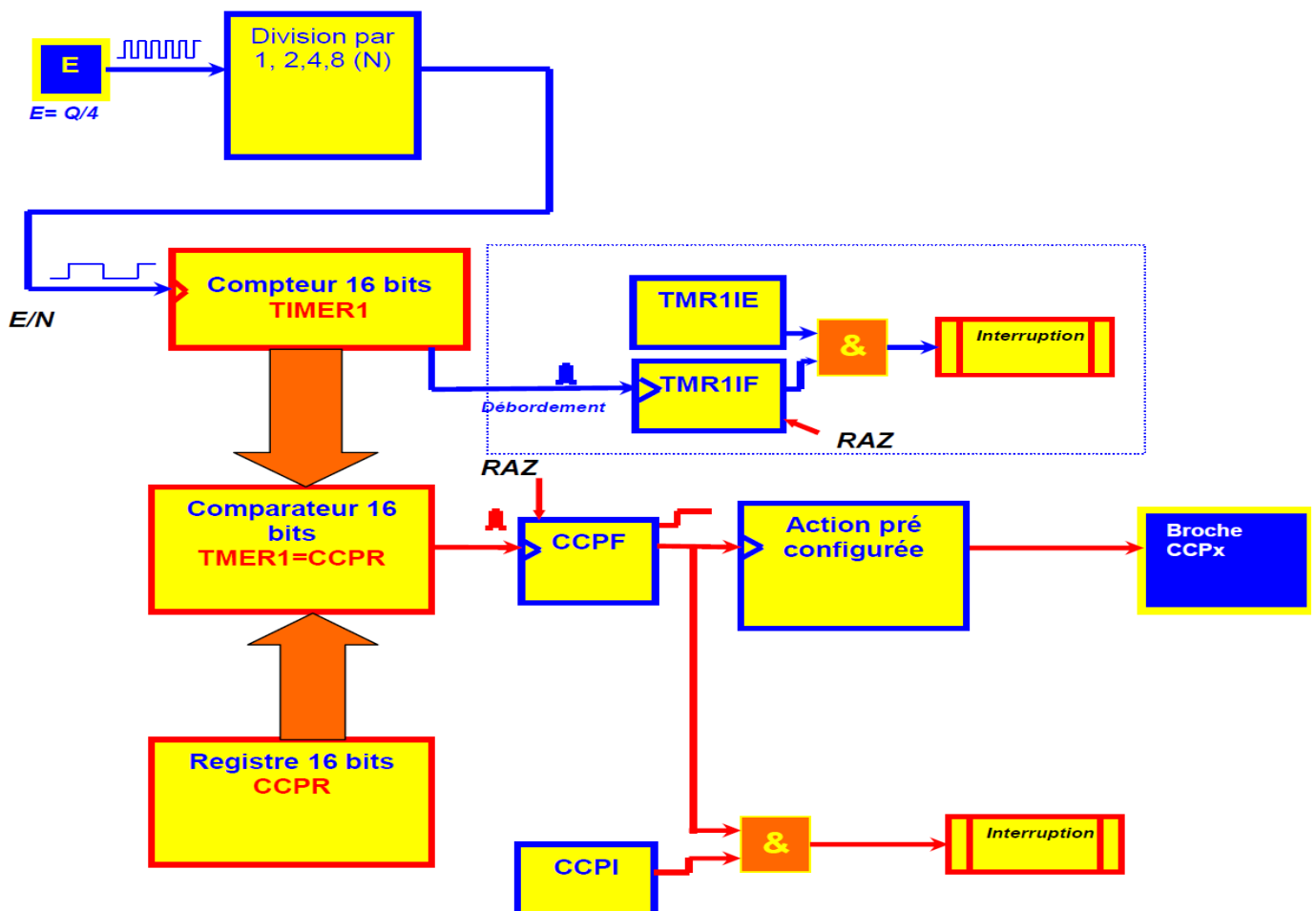
## 9 – Les Timers

### 6. Le générateur d'impulsions « Output Compare » :



Un compteur au choix issu du Timer1 ou du Timer3 tourne en permanence suivant la vitesse programmée. Une valeur a été introduite dans le comparateur (CCPR1) ; Lors de l'égalité de ces 2 nombres un évènement est généré (CCP1IF) qui provoquera le changement d'état de la bascule de sortie sur la broche (CCP1). La période du signal de sortie correspondra à la valeur du comparateur multiplié par la base de temps du Timer de référence.

### 7. Principe de la génération d'impulsions avec le Timer1 :



Valuer ajoutée à CCPR entre deux événements :  $D = \frac{\text{durée}}{ExN}$

Lors de chaque coïncidence la broche CCP évolue. Une interruption est générée, CCPR est incrémenté de la durée voulue.

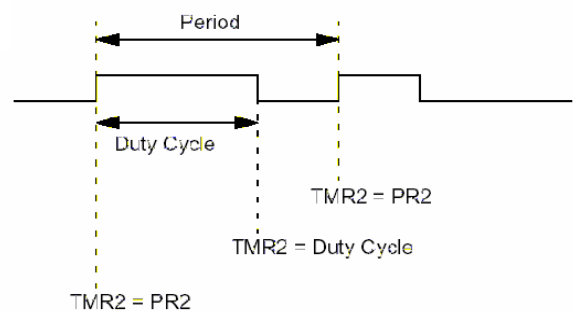
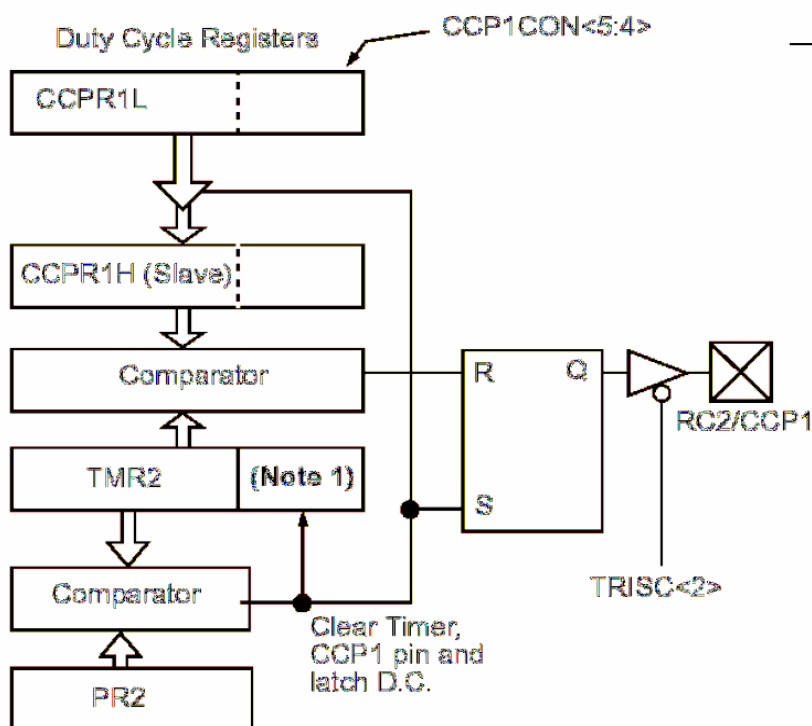
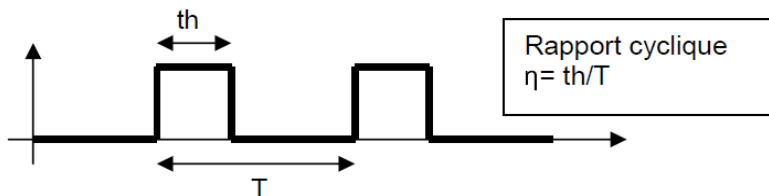
## 9 – Les Timers

### 8. Le Modulateur de Largeur d'Impulsions « PWM » :

La modulation de largeur d'impulsion (ou de rapport cyclique) est couramment utilisée dans la commande des hacheurs (pour commander un moteur à courant continu par exemple).

La sortie CPP1 peut être configurée en PWM (TRISC<2> =0).

TMR2 cadence le processus, CCPR1 représente la durée de l'état haut et PR2 la période voir configuration de CCP1CON/CCP2CON (page 22)



CCP1=0  
Lorsque TMR2=PR2  
CCP1=1  
TMR2=0  
CCPR1H=CCPR1L  
Lorsque TMR2=CCPR1H  
CCP1=0  
Lorsque TMR2=PR2  
Etc...

PR2 représente la période T  
CCPR1L représente th

**Note:** 8-bit timer is concatenated with 2-bit internal Q clock or 2 bits of the prescaler to create 10-bit time-base.

#### Exemples :

Fréquence PWM	2.44kHz	9.77kHz	39.06kHz	156.25kHz	312.50kHz	416.67kHz
Prédiviseur (1,4,16)	16	4	1	1	1	1
Valeur PR2	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
Résolution Max bits)	14	12	10	8	7	6.58

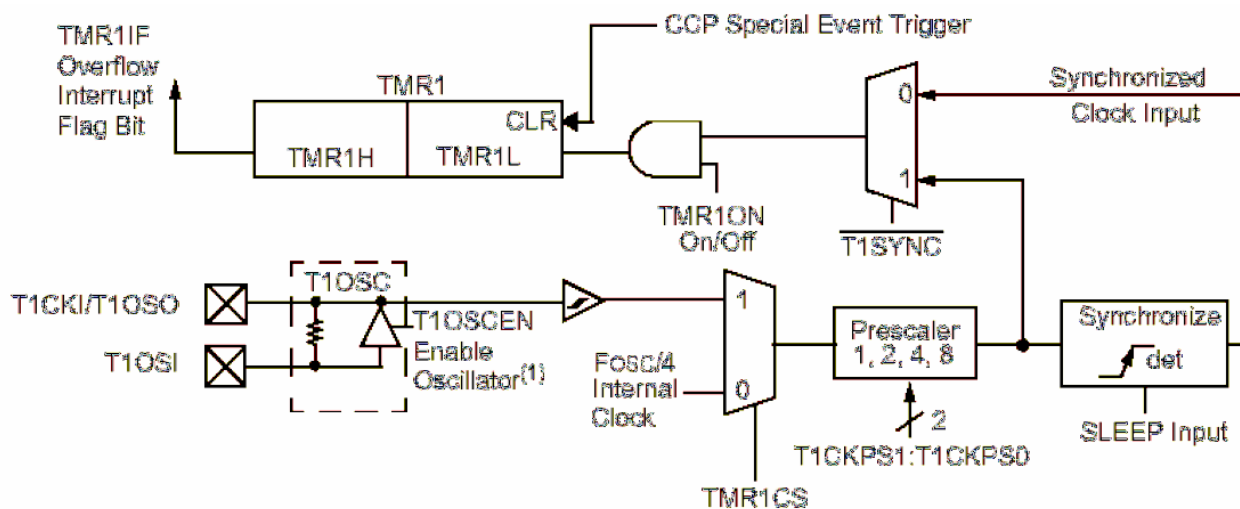
Période PWM = (PR2) + 1] • 4 • TOSC • (TMR2 valeur du prédiviseur)

Rapport cyclique PWM= (CCPR1L:CCP1CON<5:4>) • TOSC • (TMR2 valeur du prédiviseur)

$$\text{Résolution(PWM) en bits} = \frac{\log \left( \frac{\text{FOSC}}{\text{FPWM}} \right)}{\log(2)}$$

## 9 – Les Timers

### 9. Le Timer1 :



#### REGISTRE T1CON

7	6	5	4	3	2	1	0
RD16	—	T1CKPS1	T1CKPS0	T1OSCEN	T1SYNC	TMR1CS	TMR1ON

**RD16:** 16-bit Read/Write Mode Enable bit

1 = Accès à TMR1 par 16 bits

0 = Accès à TMR1 en 2 fois 8 bits

**T1CKPS1:T1CKPS0:** Valeur du prédiviseur

11 = 1:8

10 = 1:4

01 = 1:2

00 = 1:1

**T1OSCEN:** Validation de l'oscillateur (entrées sur T1OSO et T1OSI)

1 = activé

0 = désactivé

**T1SYNC:** Synchronisation avec l'horloge externe (pour le mode sleep)

Quand TMR1CS = 1:

1 = Ne pas synchroniser l'horloge externe

0 = Synchroniser l'horloge externe

**TMR1CS:** Choix de l'horloge

1 = Horloge externe

0 = Horloge interne (Fosc/4)

**TMR1ON:** Validation du TIMER1

1 = TIMER1 activé

0 = TIMER1 arrêté

#### Remarques :

En plaçant un quartz de 32.768KHz sur T1OS, il est possible d'obtenir une base de temps de 1S.

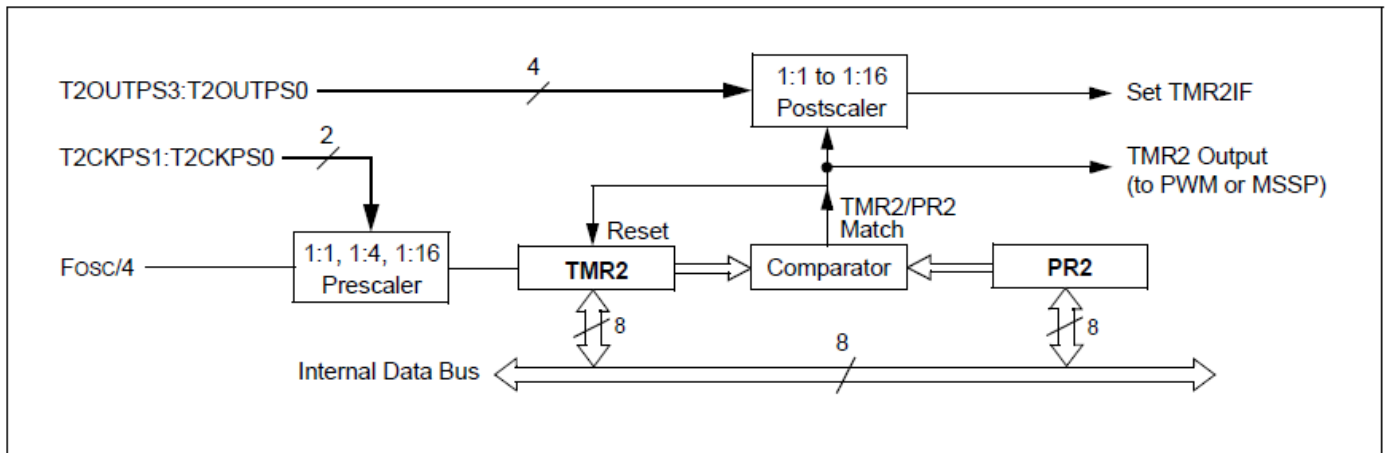
Validation de l'IT de débordement par PIE1 <TMR1IE>

Drapeau d'IT PIR1<TMR1IF>

## 9 – Les Timers

### 10. Le Timer2 :

Le Timer2 peut servir de base de temps soit au système PWM soit au périphérique série.



**REGISTER 14-1: T2CON: TIMER2 CONTROL REGISTER**

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	T2OUTPS3	T2OUTPS2	T2OUTPS1	T2OUTPS0	TMR2ON	T2CKPS1	T2CKPS0
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-3 **T2OUTPS3:T2OUTPS0:** Timer2 Output Postscale Select bits

0000 = 1:1 Postscale

0001 = 1:2 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

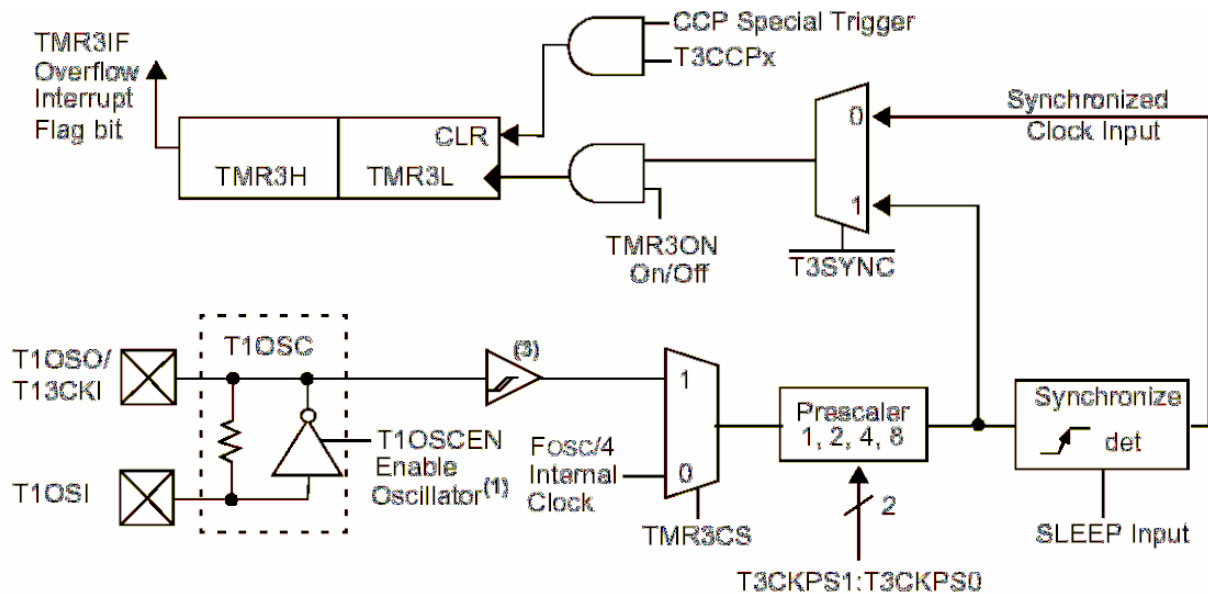
01 = Prescaler is 4

1x = Prescaler is 16

## 9 – Les Timers

### Le Timer3 :

L'horloge du Timer3 peut être générée par un deuxième oscillateur externe ou par l'horloge interne /4.



REGISTRE TCON3

7	6	5	4	3	2	1	0
RD16	T3CCP2	T3CKPS1	T3CKPS0	T3CCP1	T3SYNC	TMR3CS	TMR3ON

**RD16:** Lecture/ écriture 16-bit Read/Write

1 = Accès à TMR3 sur 16 bits

0 = Accès à TMR3 sur 2x8 bits

**T3CCP2:T3CCP1:** Liaisons Timer3 et Timer1 et CCPx

1x = Timer3 est l'horloge du module compare/capture CCP

01 = Timer3 est l'horloge du module compare/capture CCP2,

Timer1 est l'horloge du module compare/capture CCP1

00 = Timer1 est l'horloge du module compare/capture CCP

**T3CKPS1:T3CKPS0:** Prédiviseur

11 = 1:8

10 = 1:4

01 = 1:2

00 = 1:1

**T3SYNC:** Synchronisation avec l'horloge externe (pour le mode sleep)

Quand TMR1CS = 1:

1 = Ne pas synchroniser l'horloge externe

0 = Synchroniser l'horloge externe

**TMR3CS:** Choix de l'horloge

1 = Horloge externe

0 = Horloge interne (Fosc/4)

**TMR3ON:** Validation du TIMER1

1 = TIMER1 activé

0 = TIMER1 arrêté

Validation de l'IT de débordement par PIE2 <TMR3IE>

Drapeau d'IT PIR2<TMR3IF>

## 9 – Les Timers

### 11. Mots clés :

- Horloge
- Pré diviseur
- Interruption par dépassement
- Input Capture
- Output-Compare
- PWM
- Période
- Duty Cycle