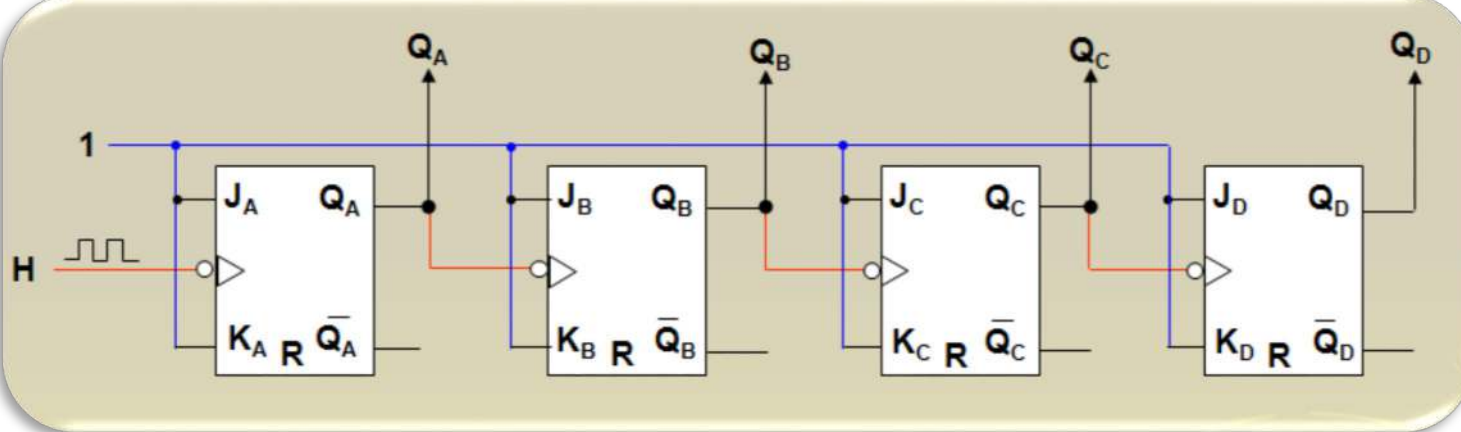
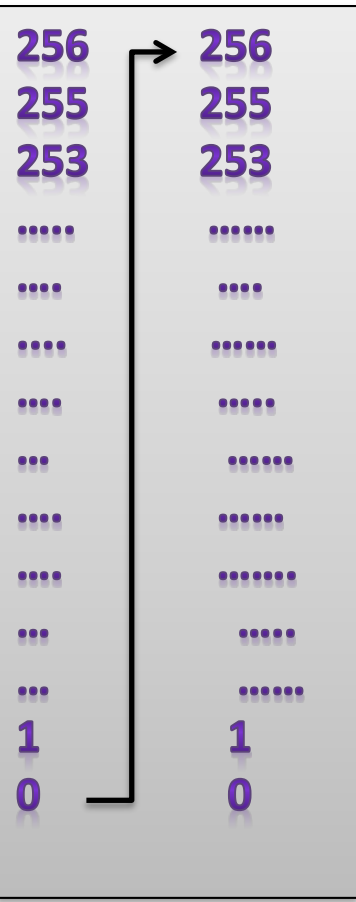
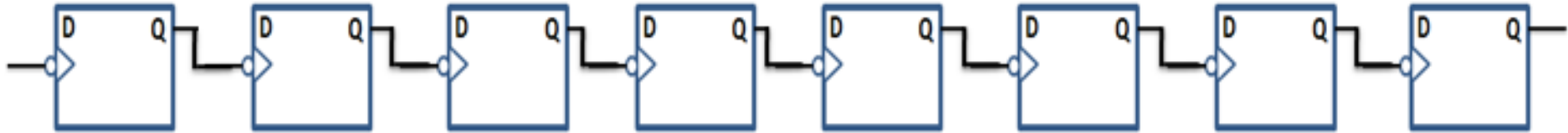


TMRO التّأجيل بواسطة



عداد التصاعدي بدورة كاملة (4قلابات) يعد إلى 15 ثم يعود إلى 0 تلقائيا

15	→	15
14		14
13		13
12		12
11		11
10		10
9		9
8		8
7		7
6		6
5		5
4		4
3		3
2		2
1		1
0		0



عداد التصاعدي بدورة كاملة (8 قلابات) يعد إلى 256 ثم يعود إلى 0 تلقائيا

File Address File Address

00h	Indirect addr. ⁽¹⁾	Indirect addr. ⁽¹⁾	80h
01h	TMRO	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h			87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 ⁽¹⁾	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch			8Ch
	36 General Purpose registers (SRAM)	Mapped (accesses) in Bank 0	
2Fh			AFh
30h			B0h
7Fh			FFh
	Bank 0	Bank 1	

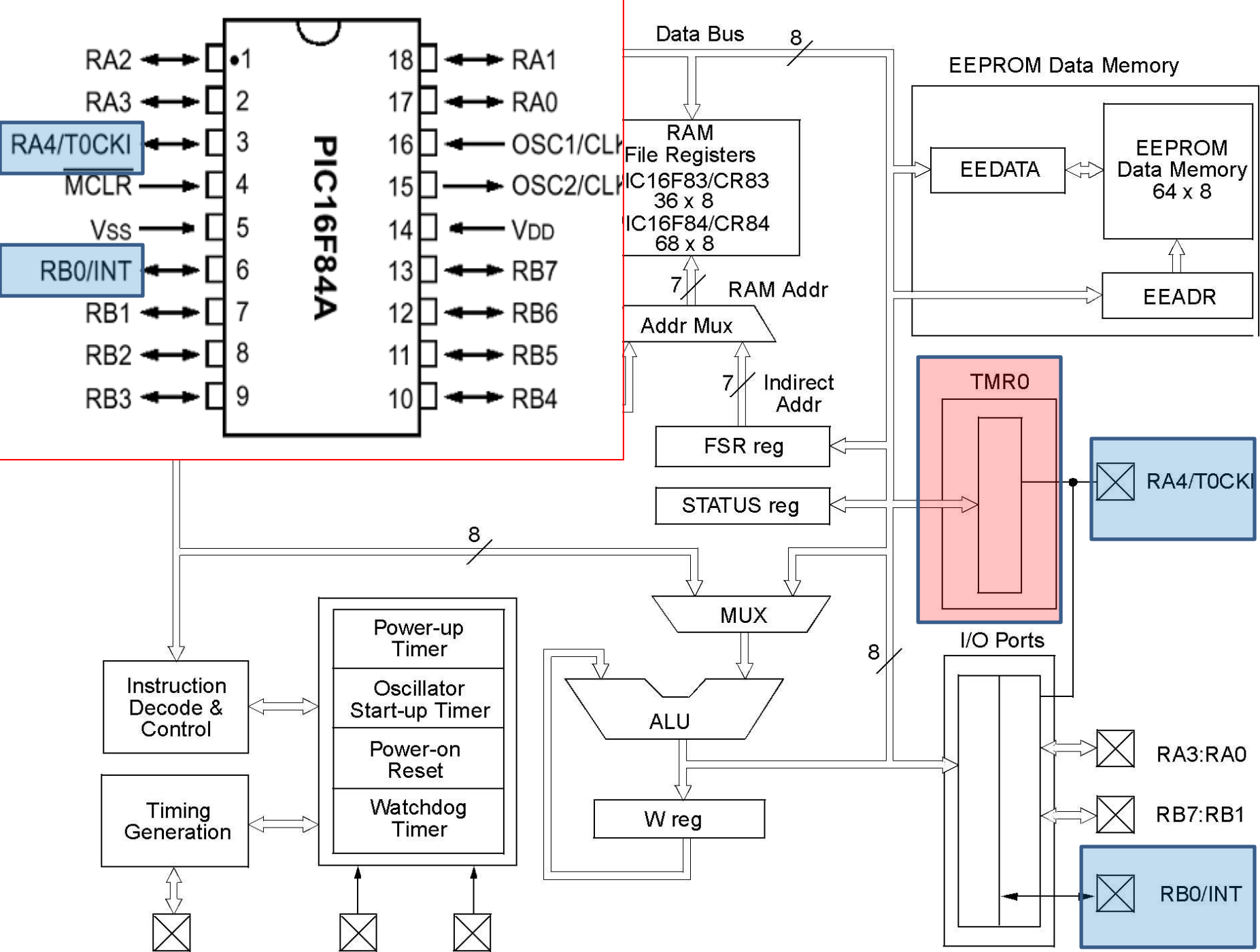
موقع السجل
TMRO

TMRO هو سجل مشفر على 8 بيت

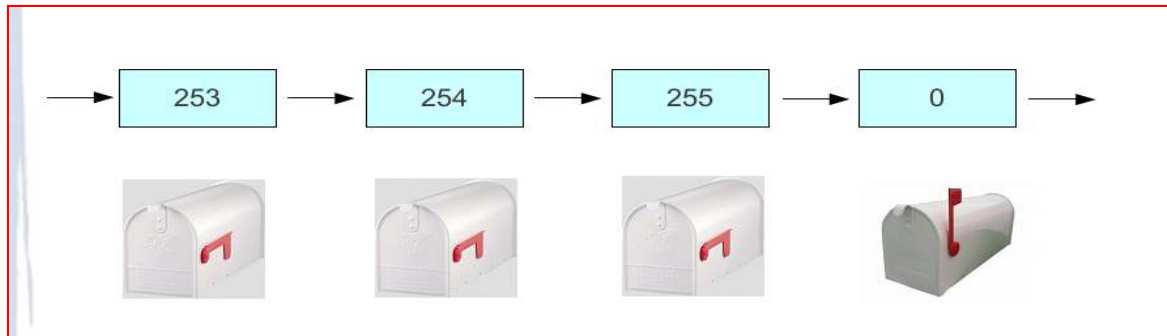
موقع السجل
INTCON

■ Unimplemented data memory location; read as '0'.

Note 1: Not a physical register.



TMRO هو عبارة عن عداد يعد حتى 256 بدورة كاملة اي انه في كل مرة يعد 256 ثم يعود إلى 0 ثم يبدأ العد من جديد



كل خصائص ومميزات السجل TMRO موجودة في سجلين آخرين هما:
- سجل OPTION-REG
- سجل INTCON

- *Registre OPTION_REG* (adresse : 81h ; 181h) : (même que le pic 16F84)

RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

- *Registre INTCON* (adresse : 0Bh ; 8Bh ; 10Bh ; 18Bh) : (même que le pic 16F84)

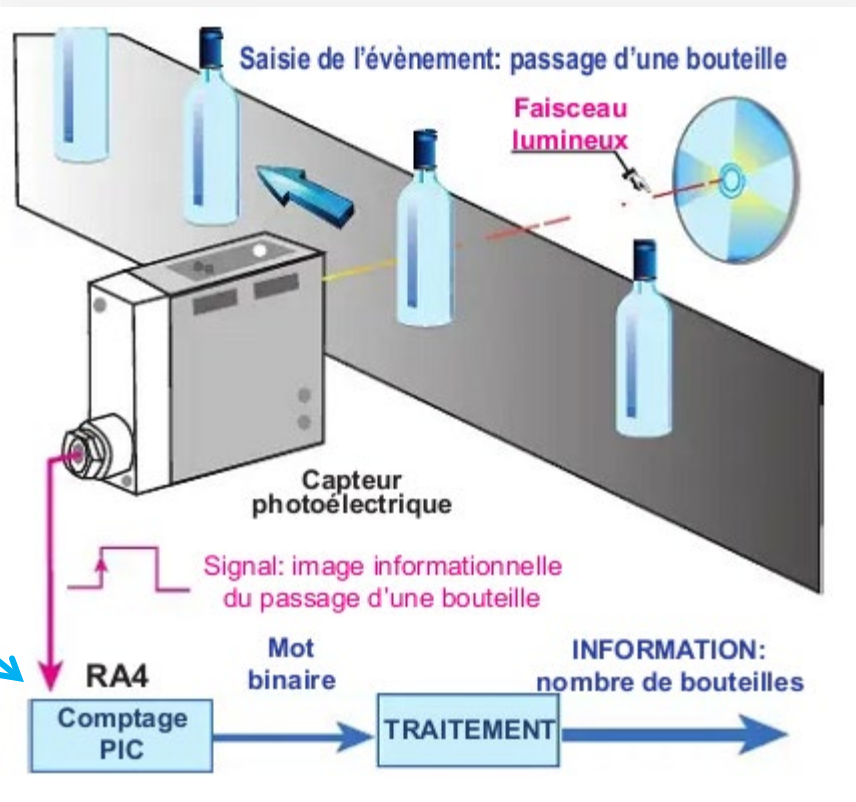
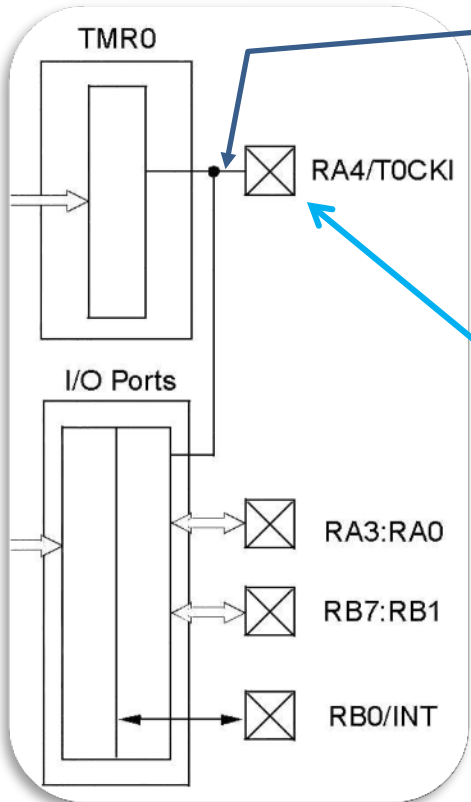
GIE	PEIE	T0IE	INTE	RBIE	T0IF	INTF	RBIF
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

RBPUR	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

استعمال إشارة ساعة داخلية
 استعمال إشارة ساعة خارجية

استعمال إشارة ساعة خارجية

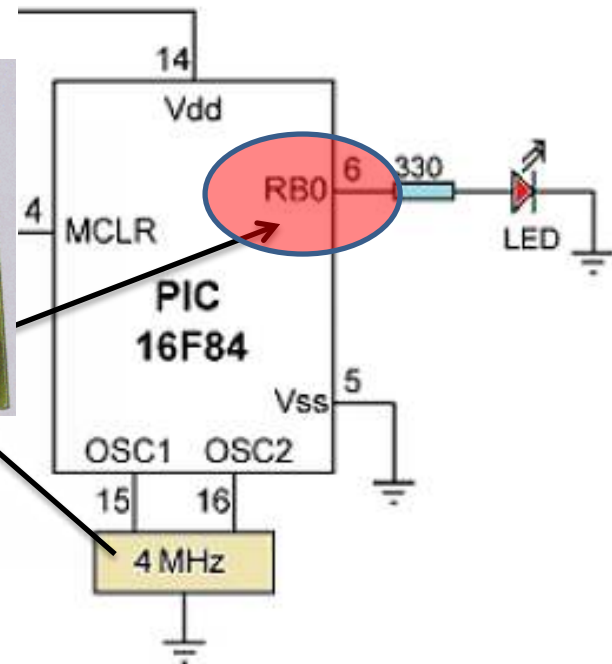
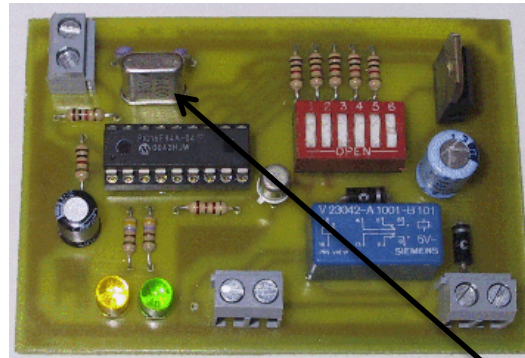
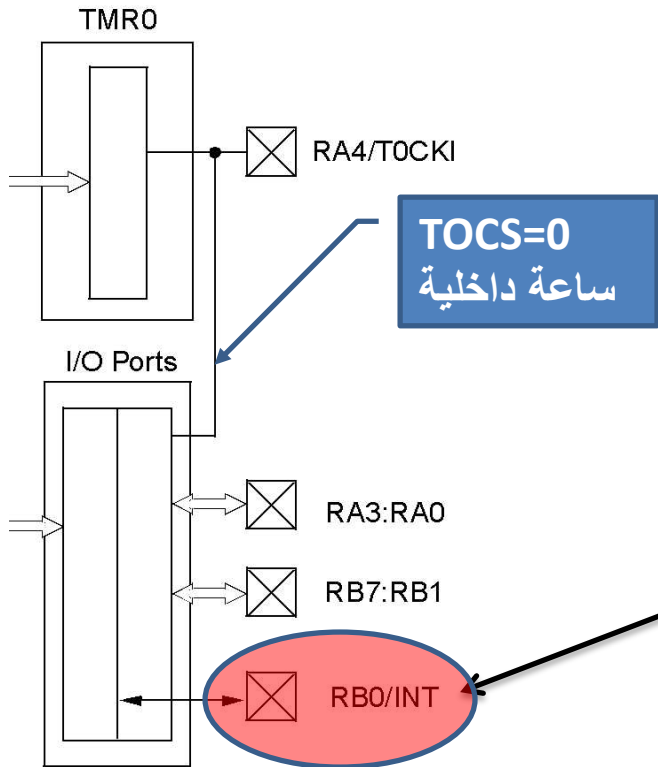
TOCS=1
 ساعة خارجية



RBPUR	INTEDG	TOCS	T0SE	PSA	PS2	PS1	PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

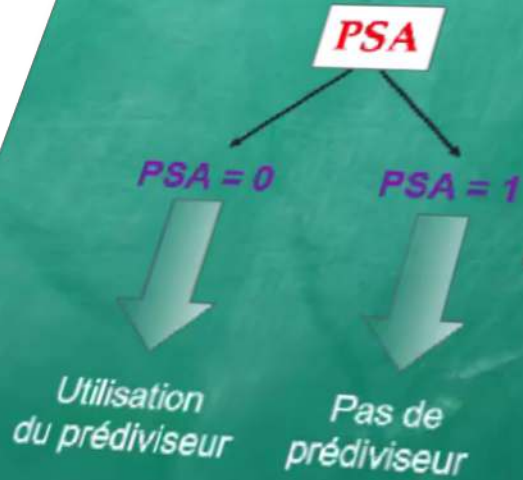
استعمال إشارة ساعة داخلية
 استعمال إشارة ساعة خارجية

استعمال إشارة ساعة داخلية



Registre OPTION_REG :

RBPV	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
------	--------	------	------	-----	-----	-----	-----



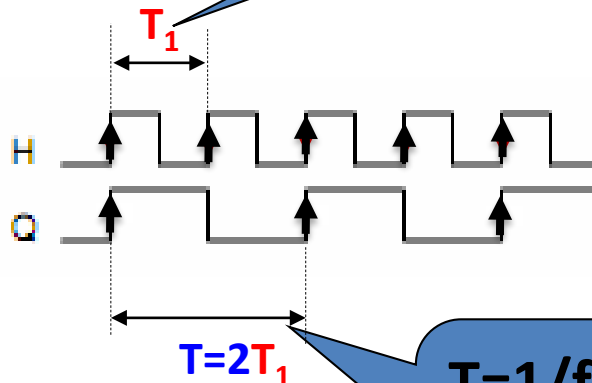
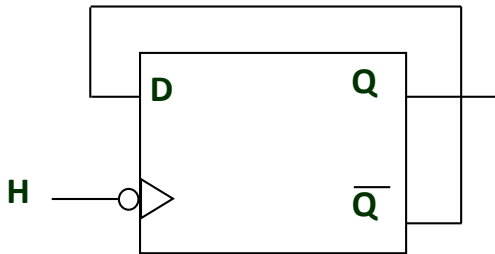
PS2	PS1	PS0	DIV
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

استعمال قاسم التردد
 PSA=0
 عدم استعمال قاسم التردد
 PSA=1

تذكير

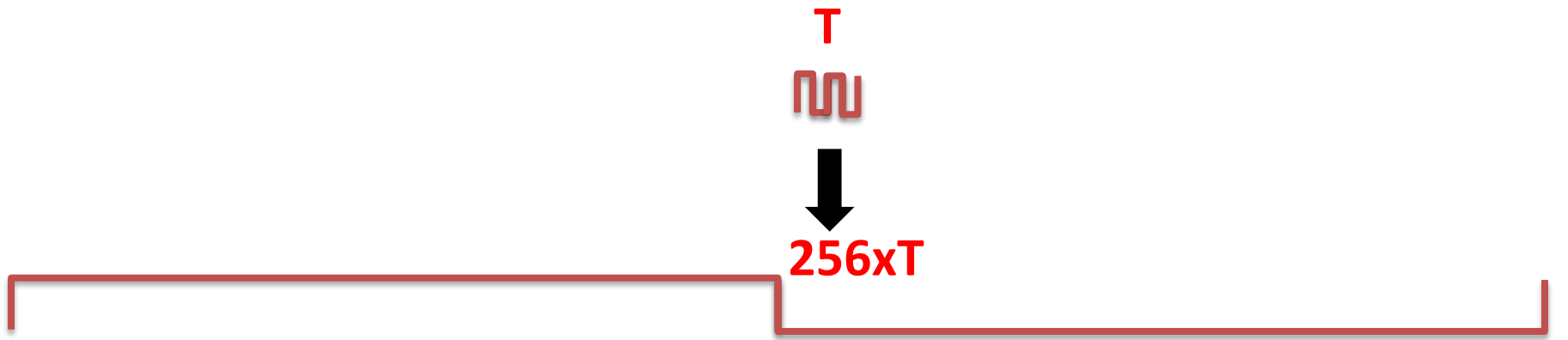
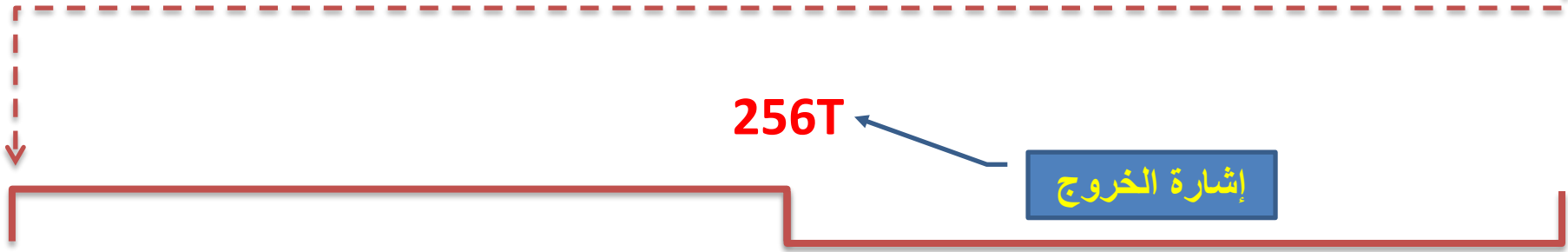
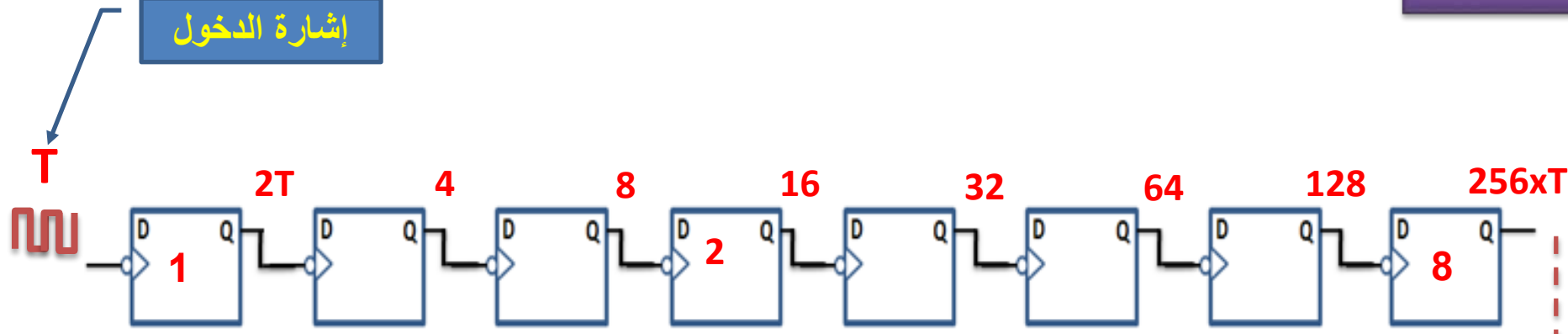
القلاب D D Bascule

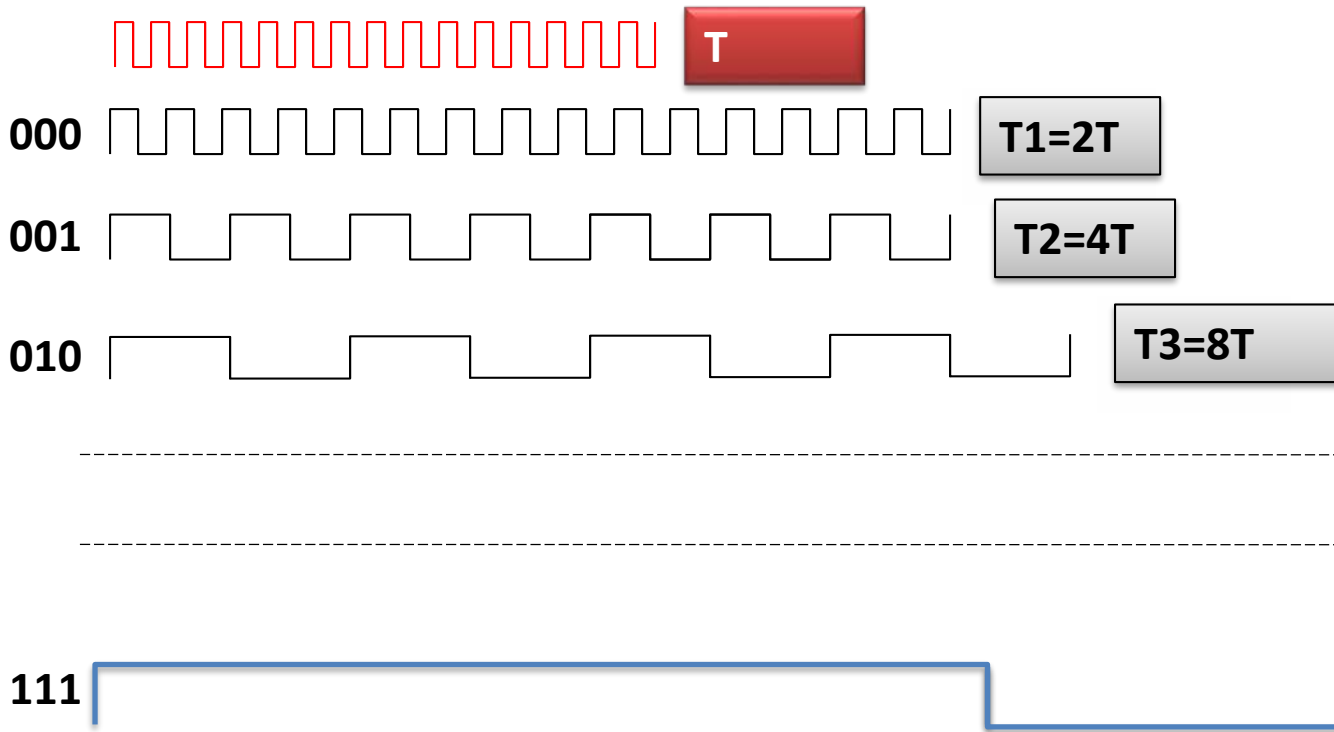


$T_1 = 1/f_1$

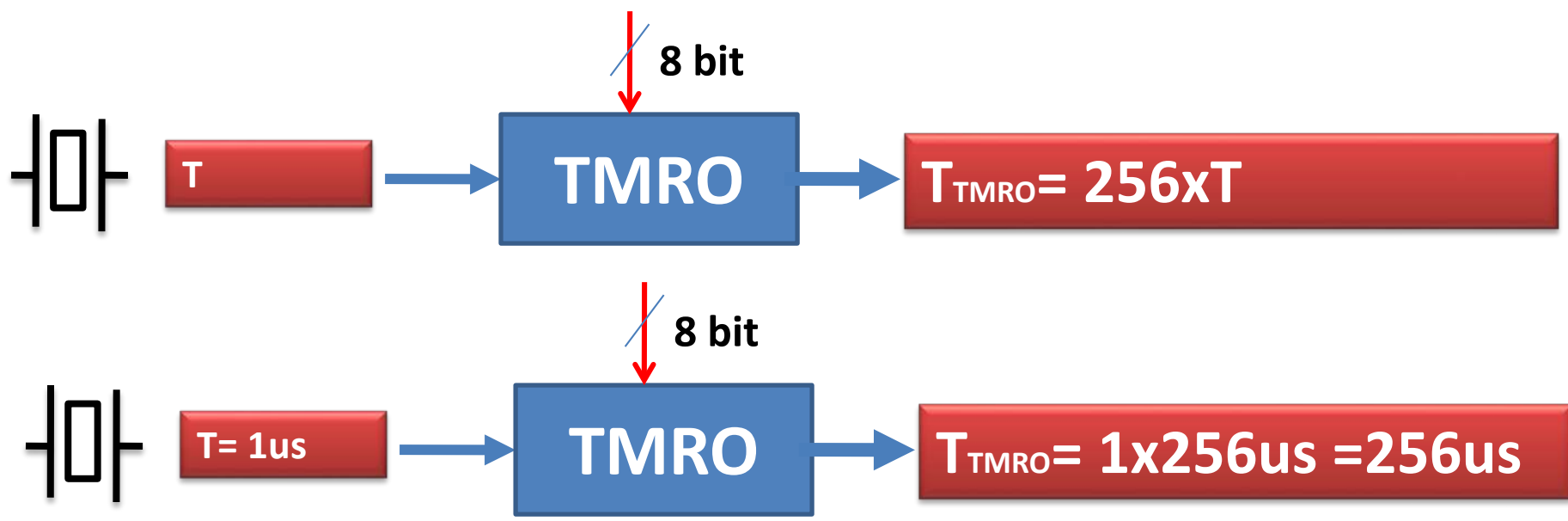
$T = 1/f \Rightarrow f = 1/T$
 $\Rightarrow f = 1/2T_1 \Rightarrow f = 1/[2(1/f_1)]$
 $\Rightarrow f = f_1/2$

إشارة الدخول





كل دور T يرفق بتوفيقية منطقية



RBP\overline{U}	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0



استعمال قاسم التردد
PRE-SCALER



PS2	PS1	PS0	TTMRO
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

Le Timer 0: Le prédiviseur

Le prédiviseur est un diviseur d'événements situé AVANT l'entrée de comptage du timer0.

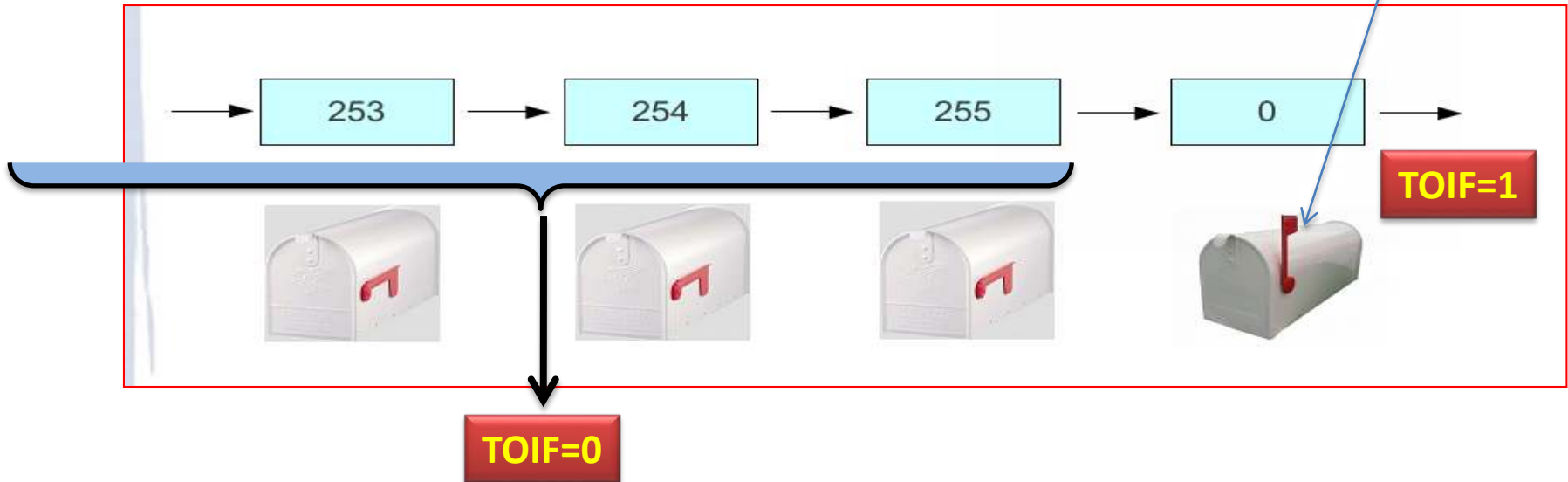
<i>PS2</i>	<i>PS1</i>	<i>PS0</i>	<i>/TIMER0</i>	<i>Temps Timer0 Qz = 4Mhz</i>	<i>Temps Timer0 Qz = 8Mhz</i>
0	0	0	2	512 μ s	256 μ s
0	0	1	4	1024 μ s	512 μ s
0	1	0	8	2048 μ s	1024 μ s
0	1	1	16	4096 μ s	2048 μ s
1	0	0	32	8192 μ s	4096 μ s
1	0	1	64	16384 μ s	8192 μ s
1	1	0	128	32768 μ s	16384 μ s
1	1	1	256	65536 μ s	32768 μ s

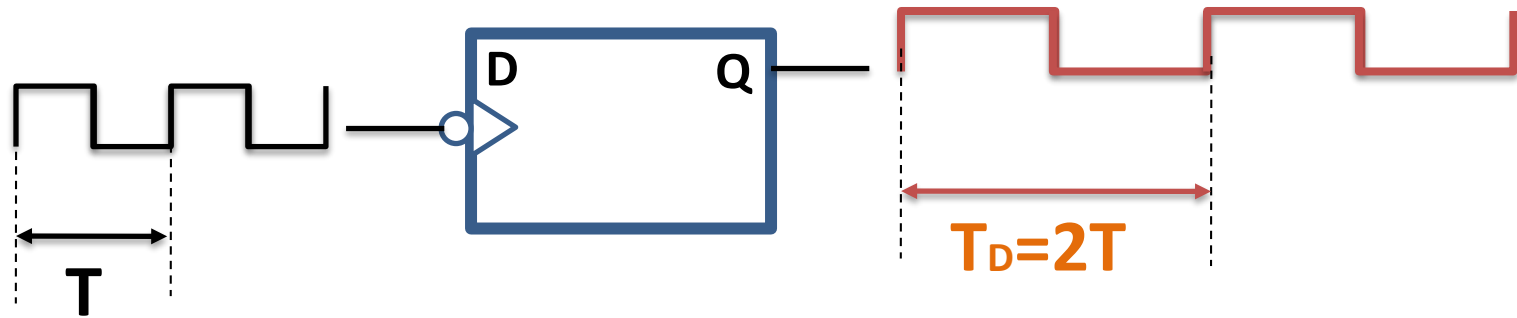
GIE	PEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0

انتهاء العد لسجل TMR0 ورفع العلم (FLAG) $TOIF=1$
 عدم انتهاء العد $TOIF=0$

رفع العلم (FLAG)

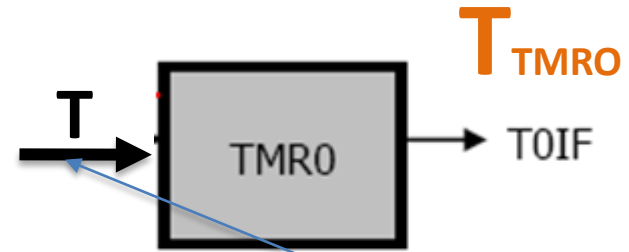
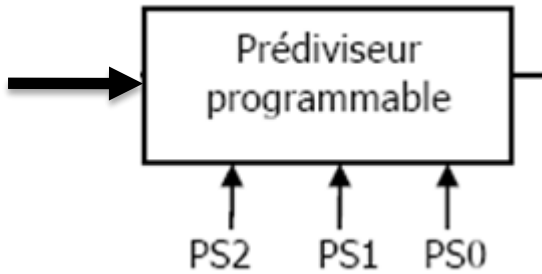
السجل TMR0



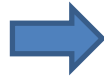


- $T=1\mu S$ \longrightarrow $T_D=2 \times 1\mu S = 2\mu S$
- $T=2\mu S$ \longrightarrow $T_D=2 \times 2\mu S = 4\mu S$
- $T=3\mu S$ \longrightarrow $T_D=2 \times 3\mu S = 6\mu S$

كلما كبر زمن إشارة التوقيت كلما كبر زمن مخرج القلاب D

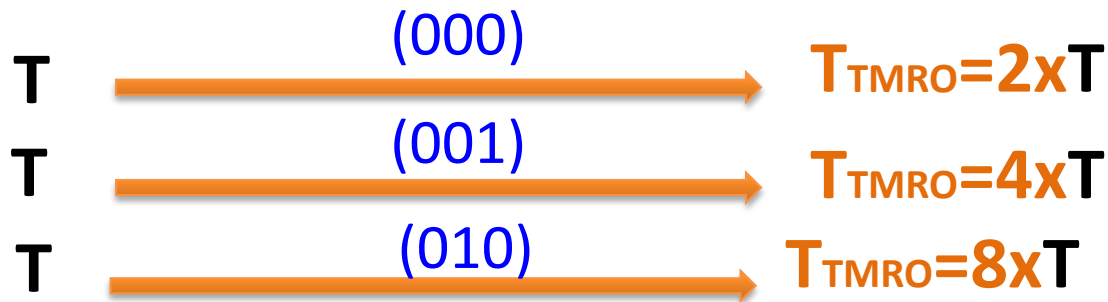


استعمال قاسم التردد
PRE-SCALER



PS2	PS1	PS0	T _{TMRO}
0	0	0	2
0	0	1	4
0	1	0	8
0	1	1	16
1	0	0	32
1	0	1	64
1	1	0	128
1	1	1	256

يعتبر قاسم التردد
كإشارة ساعة
للسجل TMRO



T $\xrightarrow{(011)}$ $T_{\text{TMRO}}=16xT$

T $\xrightarrow{(100)}$ $T_{\text{TMRO}}=32xT$

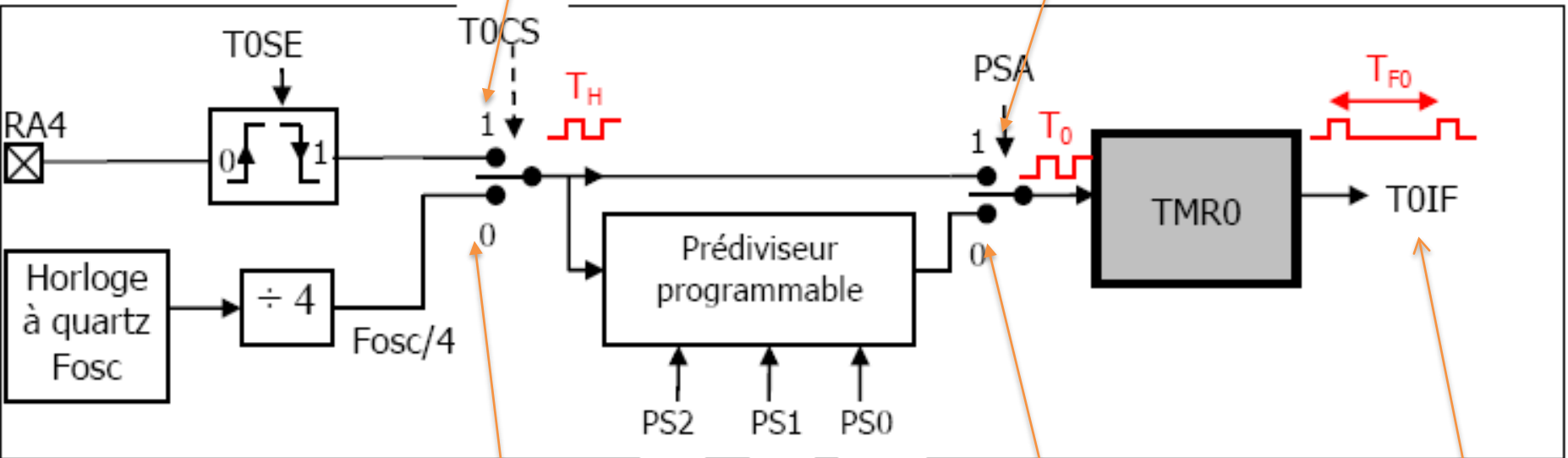
T $\xrightarrow{(101)}$ $T_{\text{TMRO}}=64xT$

T $\xrightarrow{(110)}$ $T_{\text{TMRO}}=128xT$

T $\xrightarrow{(111)}$ $T_{\text{TMRO}}=256xT$

TOCS=1
استعمال إشارة ساعة خارجية

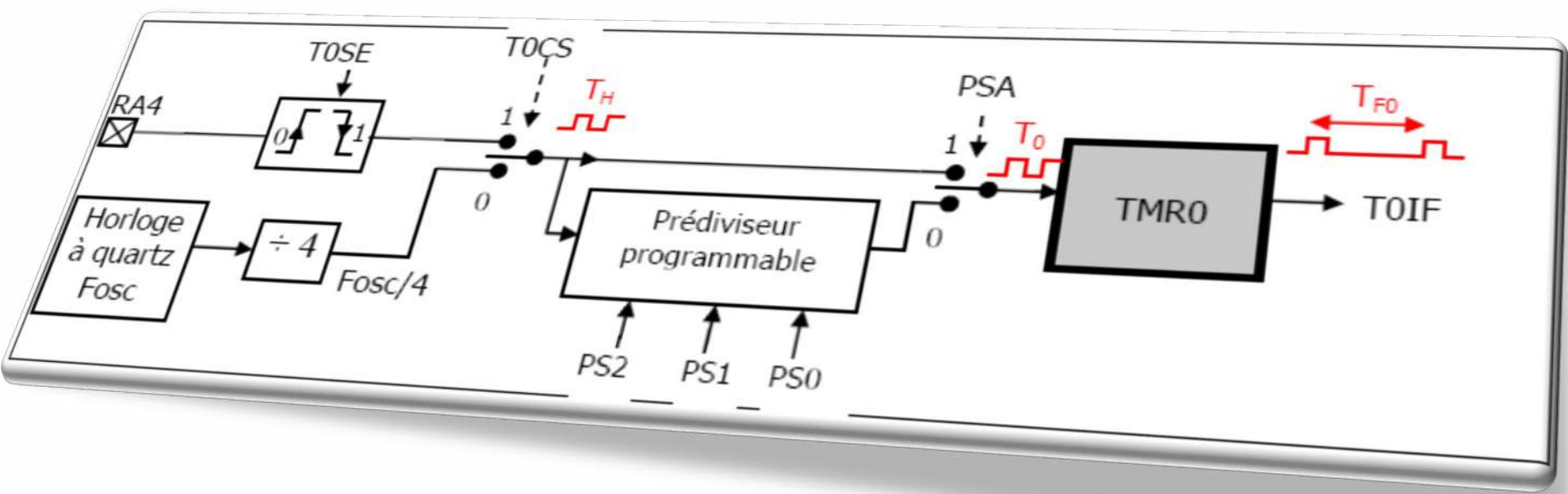
PSA=1
عدم استعمال قاسم التردد

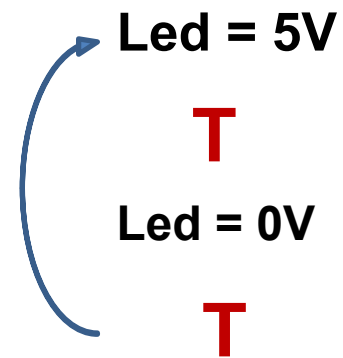
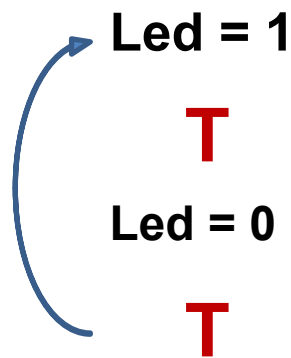
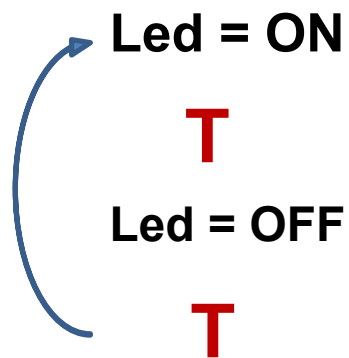
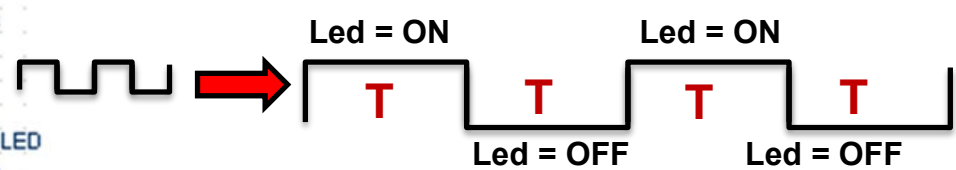
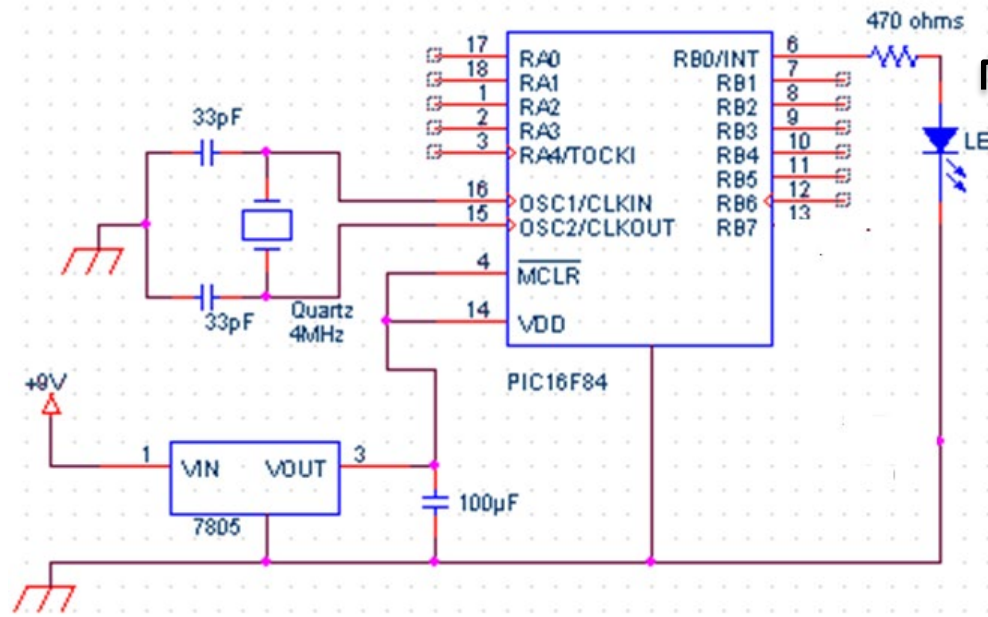


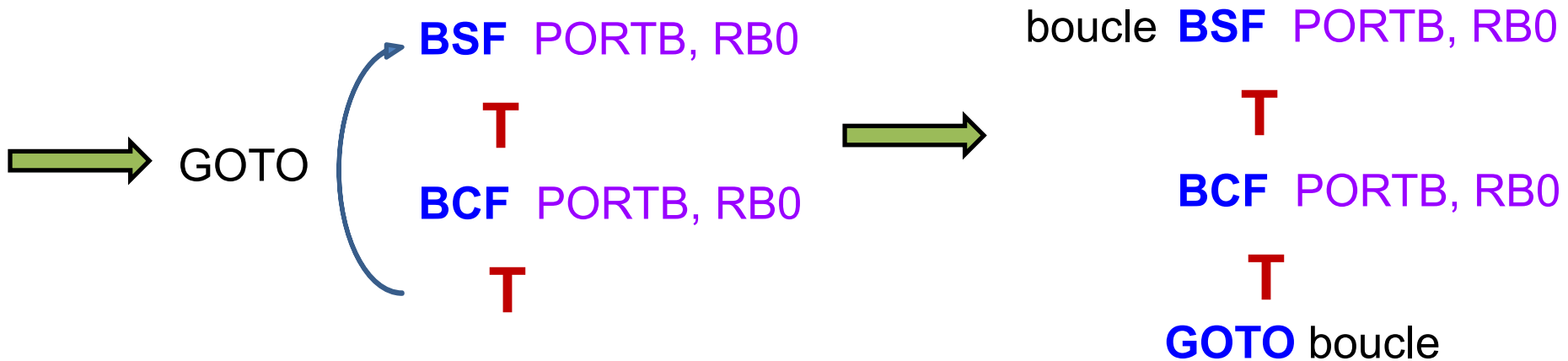
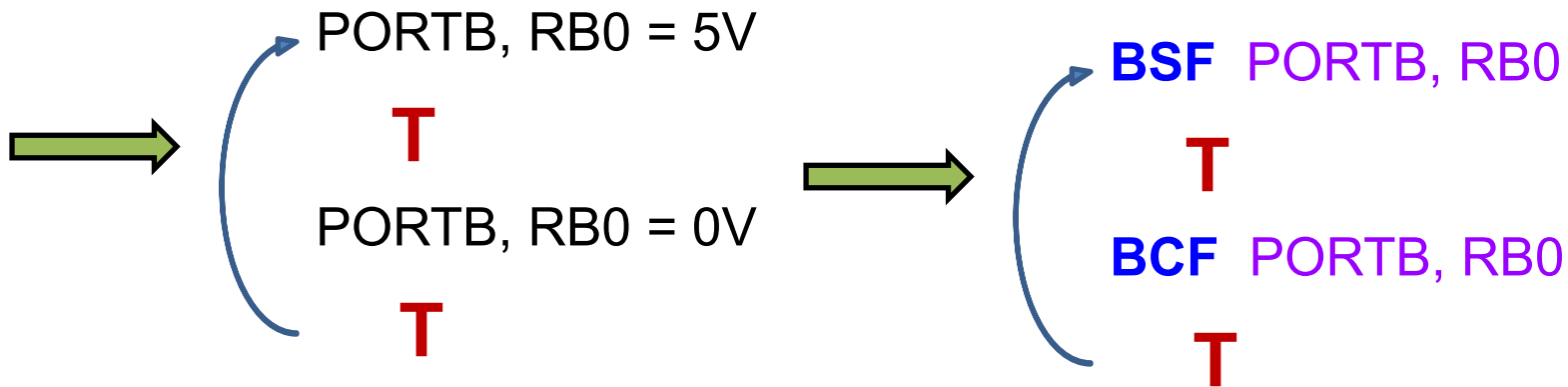
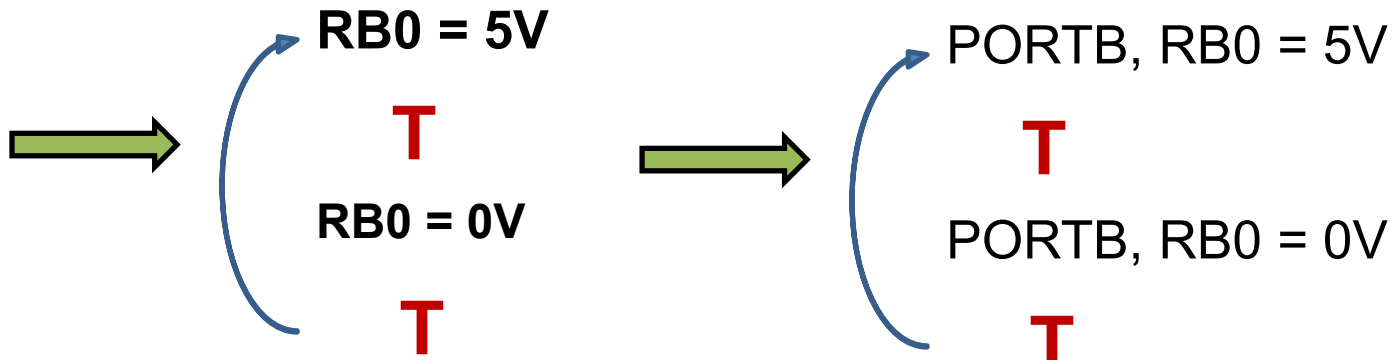
TOCS=0
استعمال إشارة ساعة داخلية

PSA=0
استعمال قاسم التردد

TOIF=1 (FLAG) ورفع العلم **TMRO** انتهاء العد لسجل
TOIF=0 عدم انتهاء العد







boucle **BSF** PORTB, RB0

T

BCF PORTB, RB0

T

GOTO boucle

T= Tempo

boucle **BSF** PORTB, RB0

Tempo

BCF PORTB, RB0

Tempo

GOTO boucle

Tempo



```
movlw    0xFC
movwf    TMRO
boucle
btfss    INTCON, TMROIF
goto     boucle
bcf      INTCON, TMROIF
return
```

boucle **BSF** PORTB, RB0

CALL Tempo

BCF PORTB, RB0

CALL Tempo

GOTO boucle

خطوات كتابة برنامج التأجيل باستخدام TMRO

1- شحن السجل OPTION-REG حسب ما نختار من قيم قاسم التردد (PRE-SCALER).

2- مسح بيت TOIF لسجل INTCON حتى يبدأ التأجيل.

3- اختبار بيت TOIF لسجل INTCON في كل مرة والقفز إذا كان يساوي 1 (عند القفز أي أن المدة الزمنية للتأجيل انتهت وبالتالي يتم تنفيذ التعليمة `BCF PORTB , RB0` وبالتالي ينطفئ المصباح أو ينفذ التعليمة `BSF PORTB , RB0` وبالتالي يشتعل المصباح في البرنامج الرئيسي).

4- عند انتهاء التأجيل (انقضاء المدة الزمنية للتأجيل $TOIF=1$) يعاد من جديد مسح بيت TOIF لسجل INTCON حتى يبدأ التأجيل من جديد.

TEMPO MOVLW B'00000111 ; شحن سجل OPTION-REG بقيمة معينة حسب قيمة PER-SCALER

MOVWF OPTION_REG ;

BCF INTCON, TOIF; مسح بيت TOIF لسجل INTCON حتى يبقى العد ساري المفعول

B2 BTFSS INTCON, TOIF; اختبار بيت TOIF لسجل INTCON والقفز إذا كان يساوي 1

GOTO B2;

BCF INTCON, TOIF; إذا أصبح بيت TOIF لسجل INTCON يساوي 1 عملية العدد انتهت ويعاد مسح بيت TOIF من جديد

TEMPO MOVLW B'00000111 ;

شحن سجل OPTION-REG بقيمة معينة حسب قيمة PER-SCALER

MOVWF OPTION_REG ;

BCF INTCON, TOIF;

مسح بيت TOIF لسجل INTCON حتى يبقى العد ساري المفعول

B2 BTFS INTCON, TOIF;

اختبار بيت TOIF لسجل INTCON والقفز إذا كان يساوي 1

GOTO B2;

BCF INTCON, TOIF;

إذا أصبح بيت TOIF لسجل INTCON يساوي 1 عملية العد انتهت ويعاد مسح بيت TOIF من جديد

TEMPO MOVLW B'00000111 ; شحن سجل OPTION-REG بقيمة معينة حسب قيمة PER-SCALER

MOVWF OPTION_REG ;

BCF INTCON, TOIF; مسح بيت TOIF لسجل INTCON حتى يبقى العد ساري المفعول

B2 BTFSS INTCON, TOIF; اختبار بيت TOIF لسجل INTCON والقفز إذا كان يساوي 1

GOTO B2;

BCF INTCON, TOIF; إذا أصبح بيت TOIF لسجل INTCON يساوي 1 عملية العدد انتهت ويعاد مسح بيت TOIF من جديد

ملاحظة هامة:

- 1- لا يظهر السجل TMRO في البرنامج إذا لم يشحن بقيمة ابتدائية (لأنه يعد إلى 256 ثم يعود إلى 0 تلقائيا كما هو الحال في العدادات بدورة كاملة).
- 2- يظهر السجل TMRO في البرنامج إذا شحن بقيمة ابتدائية (لأنه يجب معرفة قيمة الشحن الابتدائية).

تعطى علاقة التأجيل بالعبارة التالية:

$$T = T_{\text{quartz}} \times \text{Valeur du prédiviseur} \times (TMR0 - N_{\text{init}})$$

$$T = T_{\text{quartz}} \times \text{Valeur du pré-diviseur} \times (\text{TMR0-N}_{\text{init}})$$

$$T_{\text{quartz}} = 4\text{MHz}/4 = 1\text{MHz} = 1/T \Rightarrow T = 1\mu\text{s}$$

ملاحظة: تنفيذ التعليمة يمر بـ 4 مراحل لذلك يقسم التردد على 4 .

Valeur du pré-diviseur \Rightarrow (111) = 256 (قيمة مختارة على سبيل المثال)

ملاحظة: قيمة pre-scaler تأخذ من الجدول الخاص به .

(TMR0-N_{init})

$$\left. \begin{array}{l} \text{TMR0}_{\text{max}} = 256 \\ \text{N}_{\text{init}} = 0 \end{array} \right\} \Rightarrow (\text{TMR0-N}_{\text{init}}) = (256-0)$$

$$T = 1\mu\text{s} \times 256 \times (256-0)$$

$$T = 65536\mu\text{s}$$

```
START    BSF PORTB, RBO    ; LED = ON

          CALL TEMPO

          BCF PORTB, RBO    ; LED = OFF

          CALL TEMPO

          GOTO START

TEMPO    MOVLW B'00000111    ;

          MOVWF OPTION_REG    ;

          BCF INTCON, TOIF;

B2       BTFSS INTCON, TOIF;

          GOTO B2;

          BCF INTCON, TOIF;

          RETURN
```

```
START    BSF PORTB, RBO    ; LED = ON

        CALL TEMPO

        BCF PORTB, RBO    ; LED = OFF

        CALL TEMPO

        GOTO START

TEMPO    MOVLW B'00000111    ;
        MOVWF OPTION_REG    ;
        BCF INTCON, TOIF;
B2       BTFSS INTCON, TOIF;
        GOTO B2;
        BCF INTCON, TOIF;
        RETURN
```

ملاحظة: تعتبر القيمة 65536us اكبر قيمة يمكن الحصول عليها.

- من اجل الحصول القيمة اقل من 65536us هناك طريقتان :
- 1- تخفيض من قيم قاسم التردد PRE-SCALER
 - 2- شحن السجل TMRO بقيمة ابتدائية.

```
MOVLW B'00000011 ;
```

```
MOVWF OPTION_REG ;
```

```
MOVLW 0X 8F
```

```
MOVW TMRO
```

```
BCF INTCON, TOIF;
```

```
B2 BTFSS INTCON, TOIF;
```

```
GOTO B2;
```

```
BCF INTCON, TOIF;
```

يظهر السجل TMRO في البرنامج إذا شحن بقيمة ابتدائية (لأنه يجب معرفة قيمة الشحن الابتدائية).

$T = T_{\text{quartz}} \times \text{Valeur du prédiviseur} \times (\text{TMRO} - \text{Ninit})$

$T = 1\mu\text{s} \times 16 \times (256 - 143) \Rightarrow T = 1808\mu\text{s}$

$(8F)_{16} = (143)_{10}$

إذا أردنا الحصول على زمن أكبر من 65536us
فهناك طريقتان :

1- إما استعمال TMR1

2- استعمال حلقة مع TMRO

2- استعمال حلقة مع TMRO

```
START BSF PORTB, RBO ; LED = ON  
CALL TEMPO = 1s
```

```
BCF PORTB, RBO ; LED = OFF  
CALL TEMPO = 1s
```

```
GOTO START
```

$$T = T_{\text{quartz}} \times \text{Valeur du prédiviseur} \times (TMR0 - N_{\text{init}}) \cdot N(\text{Boucle})$$

```
TEMPO MOVLW 0x15 ;  
MOVWF COUNT ;
```

$$T = 1\mu\text{s} \times 256 \times (256 - 0) \cdot 15$$

```
BSF STATUS, RPO ;  
MOVLW B'00000111 ;  
MOVWF OPTION_REG ;
```

$$T = 983040\mu\text{s} = 0.98\text{s}$$

```
BCF INTCON, TOIF;  
B2 BTFS INTCON, TOIF;
```

للحصول على 1S بالضبط يمكن استعمال التعليمة NOP داخل الحلقة.

```
GOTO B2;  
BCF INTCON, TOIF;
```

```
DECFSZ COUNT, 1  
GOTO B2;
```

```
RETURN
```

```
    movlw    0xFC
    movwf    TMRO
```

```
boucle
```

```
    btfss   INTCON, TMROIF
    goto    boucle
    bcf     INTCON, TMROIF
    return
```

```
;*****
```

```
;                               Programme principal
```

```
;*****
```

```
main    bsf     PORTB, 0
        call    tempo
        bcf     PORTB, 0
        call    tempo
        goto    main
```