

ĐẠI HỌC CÔNG NGHIỆP TP. HỒ CHÍ MINH
KHOA CÔNG NGHỆ ĐIỆN TỬ
BỘ MÔN ĐIỆN TỬ MÁY TÍNH

CHƯƠNG 6

MODUL ĐỊNH THỜI

(TIMER)

CHƯƠNG 5: MODUL ĐỊNH THỜI

Nội dung bao gồm:

- **Tổng quan về bộ định thời**
- **Giới thiệu sơ lược về Timer 0**
- **Giới thiệu sơ lược về Timer 1**
- **Giới thiệu sơ lược về Timer 2**
- **Ví dụ minh họa và bài tập ứng dụng.**

TỔNG QUAN BỘ ĐỊNH THỜI

➤ Các bộ định thời được dùng cho nhiều chức năng:

- **Định thời gian để kích hoạt sự kiện**
- **Đếm số lượng sự kiện (Đếm xung)**
- ...

➤ PIC16F887 có 3 bộ định thời:

- Timer 0
- Timer 1
- Timer 2.



TỔNG QUAN BỘ ĐỊNH THỜI

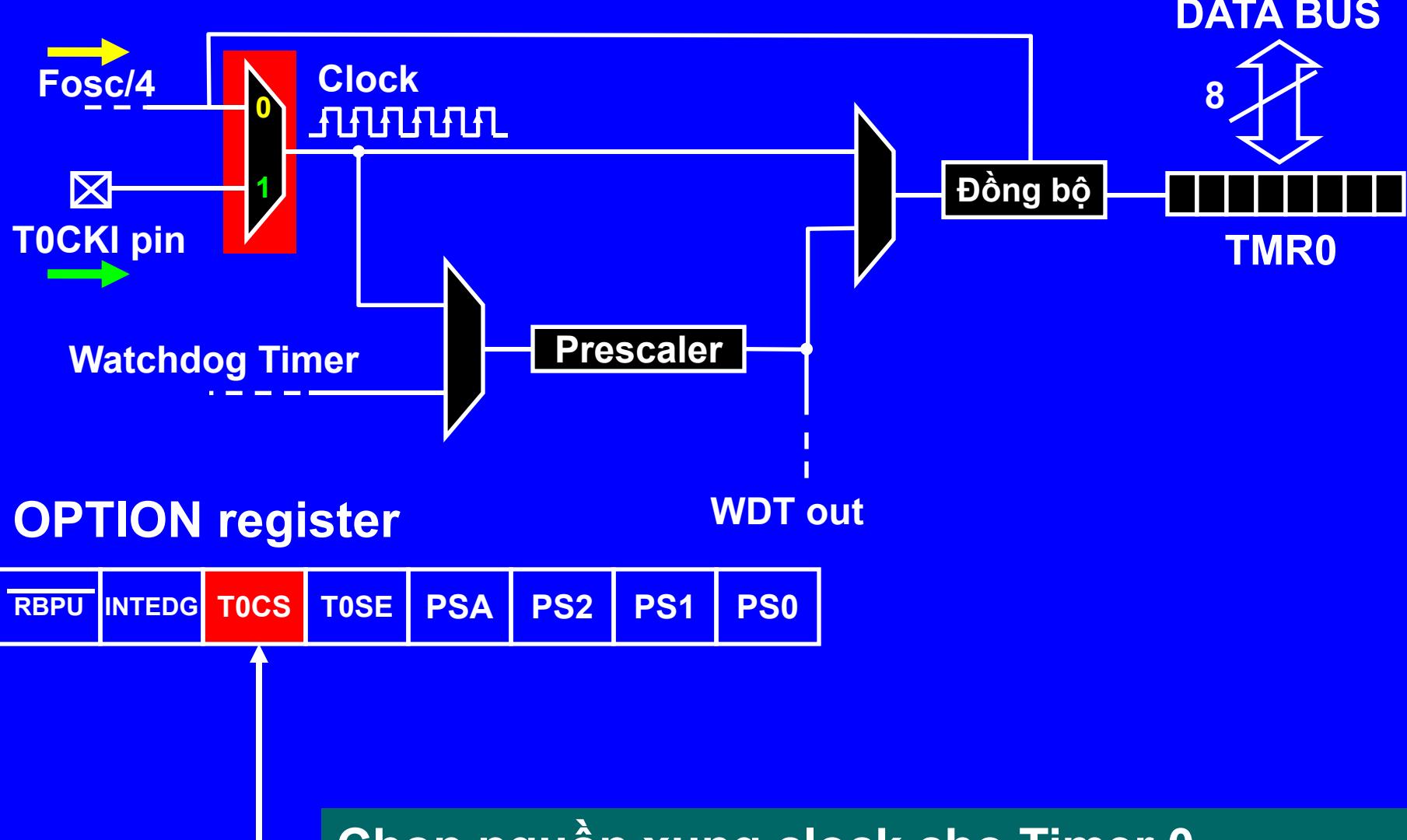
| | TIMER 0 | TIMER 1 | TIMER 2 |
|--|---|---|--|
| Kích thước thanh ghi | 8 bit (TMR0) | 16 bit (TMR1H:TMR1L) | 8-bit (TMR2) |
| Nguồn xung (Bên trong) | $F_{osc}/4$ | $F_{osc}/4$ | $F_{osc}/4$ |
| Nguồn xung (Bên ngoài) | Chân T0CKI | Chân T1CKI hoặc bộ dao động Timer 1 (T1OSC) | Không có |
| Tỉ lệ xung clock khả dụng (Độ phân giải) | Prescaler 8 bit (1:2 → 1:256) | Prescaler 3 bit (1:1, 1:2, 1:4, 1:8) | Prescaler (1:1, 1:4, 1:8) Postscaler (1:1 → 1:16) |
| Sự kiện ngắt và vị trí cò ngắt | Khi bị tràn FFh → 00h (TMR0IF của INTCON) | Khi bị tràn FFFFh → 0000h (TMR1IF của PIR1) | TMR2 bằng với PR2 (TMR2IF của PIR2) |
| Có thẻ "đánh thức" PIC từ chế độ "ngủ" | Không | Có | Không |

TIMER 0

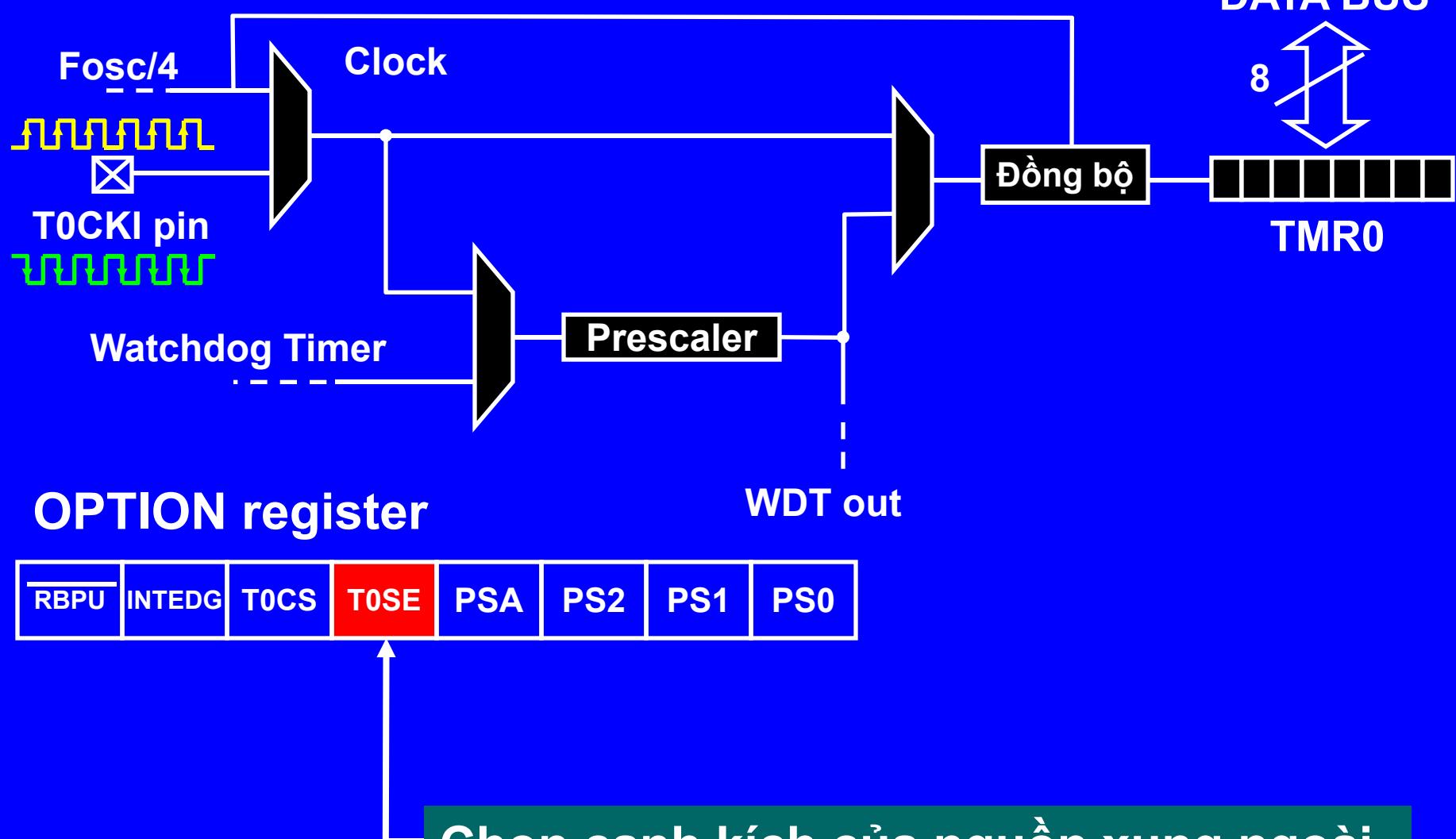
➤ Modul Timer 0 là một bộ định thời/đếm 8 bit với các tính năng:

- Bộ định thời/đếm 8 bit (8-bit Timer/Counter)
- Có bộ chia tỉ lệ (Prescaler) 8 bit, dùng chung với bộ định thời Watchdog
- Tạo tín hiệu ngắt khi Timer 0 bị tràn
- Có khả năng lập trình để sử dụng nguồn xung clock bên trong hoặc bên ngoài
- Có khả năng lập trình để chọn cạnh kích của xung clock kích bên ngoài (cạnh lên hay cạnh xuống).

SƠ ĐỒ KHỐI TIMER 0



SƠ ĐỒ KHỐI TIMER 0

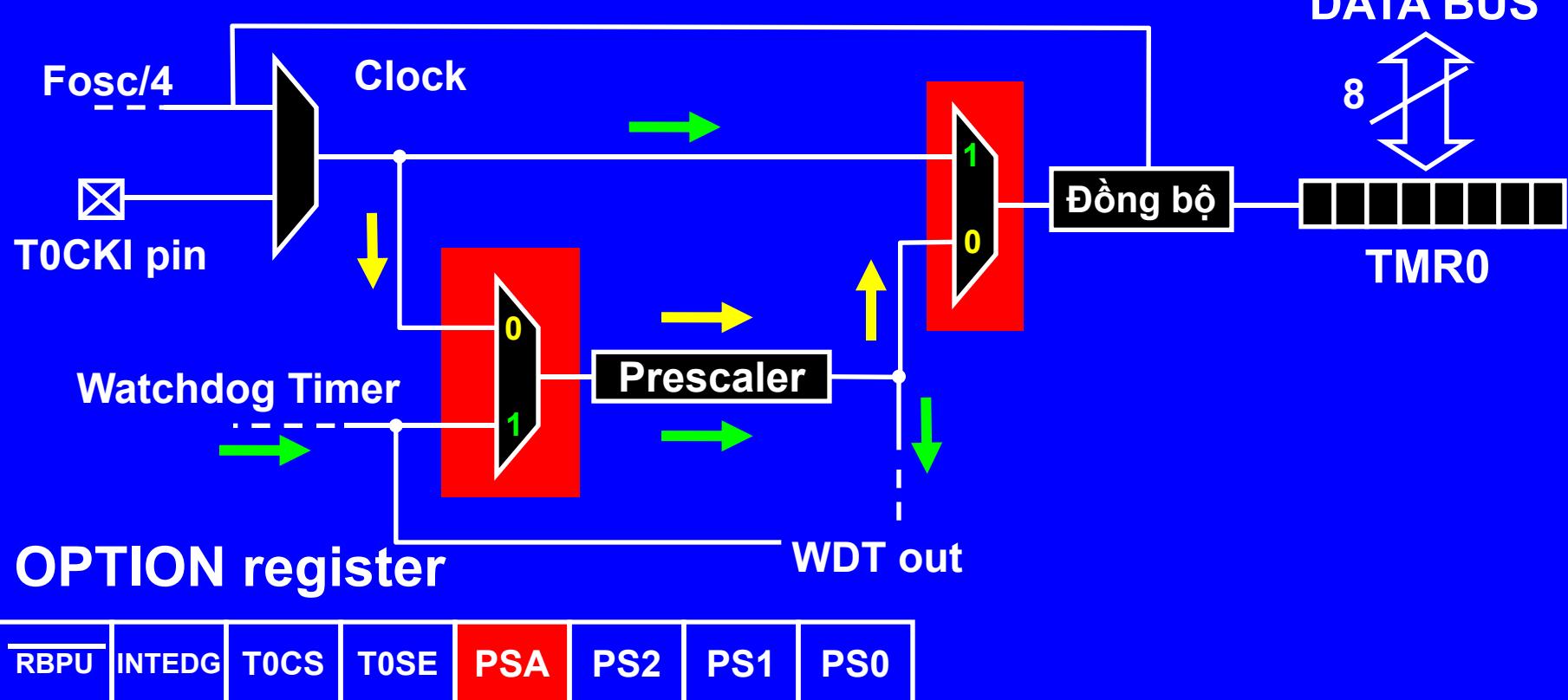


Chọn cạnh kích của nguồn xung ngoài

1 = Tăng TMR0 tại cạnh lên (L → H)

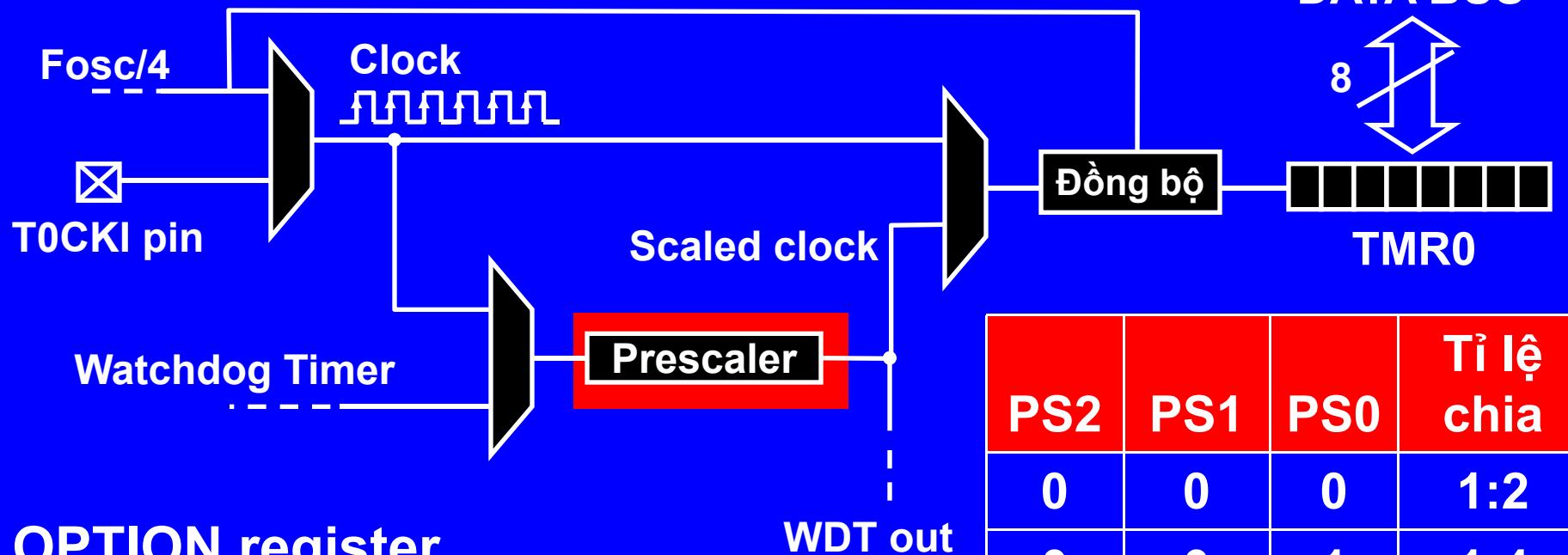
0 = Tăng TMR0 tại cạnh xuống (H → L)

SƠ ĐỒ KHÓI TIMER 0



Chỉ định Prescaler
1 = Prescaler dùng cho WDT
0 = Prescaler dùng cho Timer 0

SƠ ĐỒ KHÓI TIMER 0



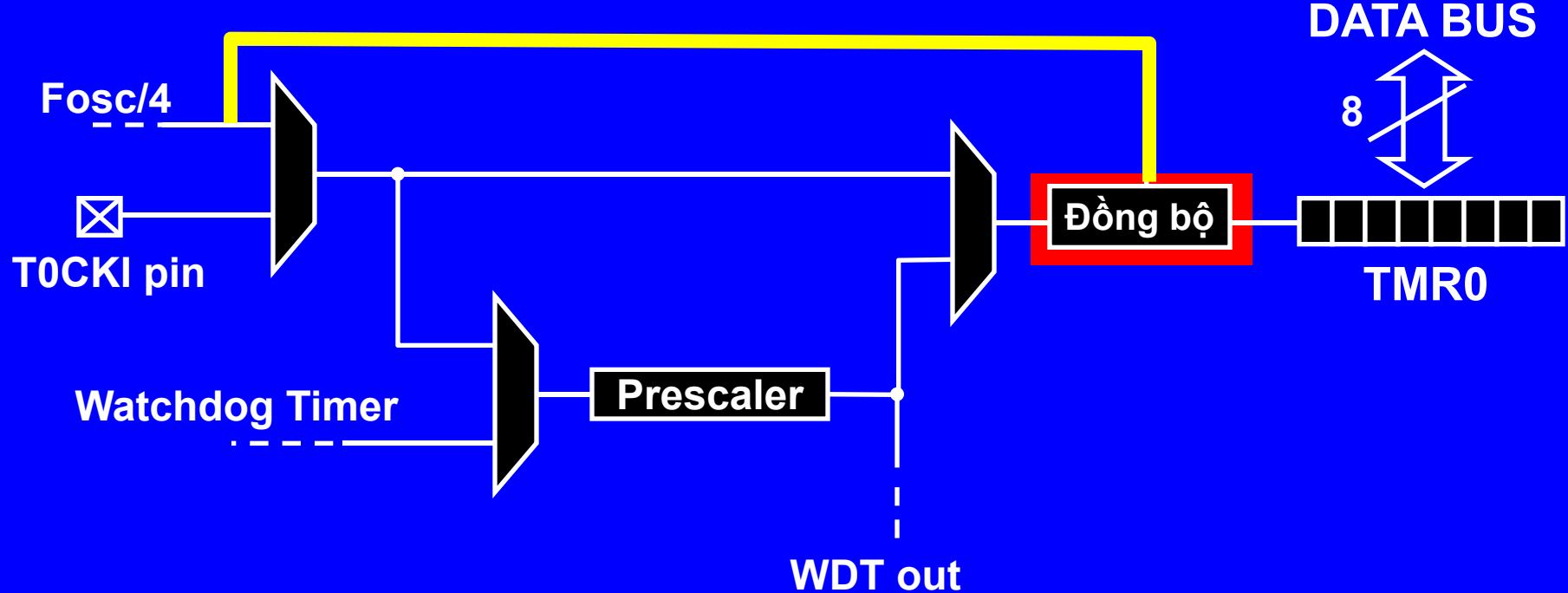
OPTION register

| | | | | | | | |
|------|--------|------|------|-----|-----|-----|-----|
| RBPU | INTEDG | T0CS | T0SE | PSA | PS2 | PS1 | PS0 |
|------|--------|------|------|-----|-----|-----|-----|

Các bit chọn tỉ lệ
cho Prescaler

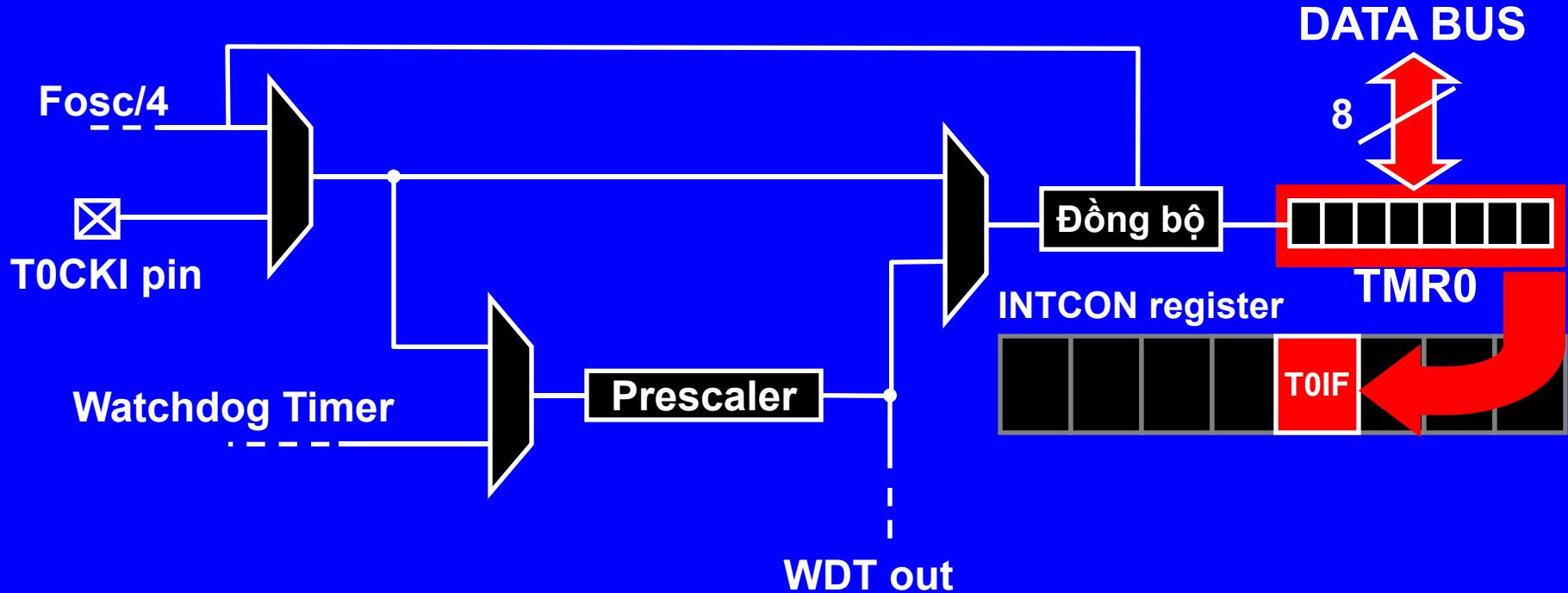
| PS2 | PS1 | PS0 | Tỉ lệ chia |
|-----|-----|-----|------------|
| 0 | 0 | 0 | 1:2 |
| 0 | 0 | 1 | 1:4 |
| 0 | 1 | 0 | 1:8 |
| 0 | 1 | 1 | 1:16 |
| 1 | 0 | 0 | 1:32 |
| 1 | 0 | 1 | 1:64 |
| 1 | 1 | 0 | 1:128 |
| 1 | 1 | 1 | 1:256 |

SƠ ĐỒ KHỐI TIMER 0



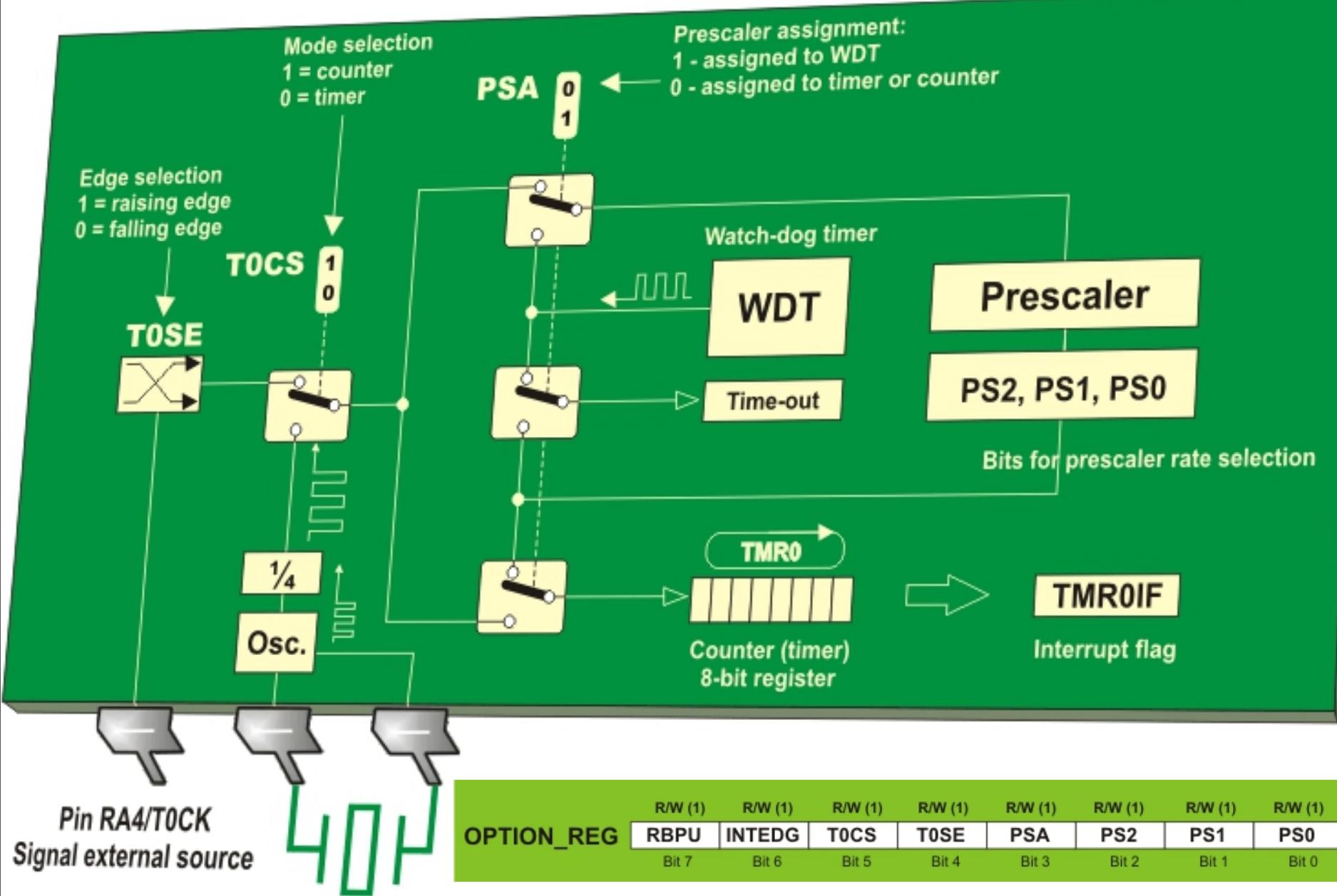
Nếu nguồn xung clock cấp cho Timer 0 được lấy từ chân **T0CKI** thì nó sẽ được đồng bộ hóa với nguồn xung clock bên trong

SƠ ĐỒ KHỐI TIMER 0



- TMR0 có thể đọc hoặc ghi
 - Thao tác ghi sẽ làm cấm việc tăng dữ liệu của Timer 0 trong 2 chu kỳ máy
- TOIF trong INTCON sẽ được đặt ($TOIF = 1$) khi Timer 0 bị tràn ($TMR0 = FFh \rightarrow 00h$).

SƠ ĐỒ KHÓI TIMER 0



TIMER 0: MỘT SỐ ĐIỂM CẦN LƯU Ý

- Khi Prescaler được gán cho Timer thì bất cứ thao tác GHI dữ liệu vào TMR0 sẽ làm cho Prescaler bị xóa
- Khi Prescaler được gán cho WDT thì lệnh CLRWDT sẽ làm cho Prescaler + WDT bị xóa
- Sự kiện ngắt khi Timer 0 tràn không thẻ "đánh thức" vi điều khiển khỏi chế độ "ngủ"
- Khi Timer được sử dụng như 1 bộ đếm (Counter) thì chiều dài xung (khoảng thời gian giữa 2 xung liên tiếp) tối thiểu là:
 - Không có Prescaler: $2 \times T_{osc} + 20\text{ns}$
 - Có Prescaler: 10ns

TIMER 0: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 1:** Ghi vào thanh ghi TMR0 giá trị tương ứng cho thời gian cần định thời

$$t_{DELAY} = (256 - [TMR0]) \times 4 \times \frac{1}{fosc} \times Pre$$

t_{DELAY} : Thời gian cần định thời (μs)

$fosc$: Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1; 2, 4, ..., 256)

[TMR0]: Giá trị cần ghi vào thanh ghi TMR0

$$0 \leq [TMR0] \leq 255$$

- **BƯỚC 2:** Xóa cờ báo tràn (cờ ngắt) TMR0IF.

TIMER 0: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 3:** Chọn chế độ hoạt động của Timer 0
 - Chế độ định thời gian (Timer)
 - Kích hoạt hoặc vô hiệu hóa ngắt (tùy chọn)
 - Gán Prescaler cho Timer và chọn giá trị tỉ lệ của Prescaler (tùy chọn)
- **BƯỚC 4:** Xác định thời điểm Timer 0 bị tràn (hay nói cách khác là thời điểm đủ thời gian cần định thời của Timer) bằng cách
 - Kiểm tra cờ TMR0IF (nếu dùng thăm dò)
 - Xử lý ISR của Timer 0 (nếu dùng ngắt).

TIMER 0: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để đếm các sự kiện (Counter):

- **BƯỚC 1:** Xóa giá trị trong thanh ghi TMR0 (hoặc đặt giá trị ban đầu của bộ đếm)
- **BƯỚC 2:** Xóa cờ báo tràn (cờ ngắt) TMR0IF
- **BƯỚC 3:** Chọn chế độ hoạt động của Counter 0
 - Chế độ đếm sự kiện (Counter)
 - Kích hoạt hoặc vô hiệu hóa ngắt (tùy chọn)
 - Gán Prescaler cho Timer và chọn giá trị tỉ lệ của Prescaler (tùy chọn)
 - Chọn loại cạnh kích của nguồn xung đếm bên ngoài
- **BƯỚC 4:** Đọc về và xử lý số xung đếm được trong thanh ghi TMR0
 - Dựa vào cờ báo tràn TMR0IF để xử lý các trường hợp số xung đếm vượt quá 255.

VÍ DỤ MINH HỌA

- VÍ DỤ 1 : Viết chương trình con tạo trễ $t_{delay} = 100\mu s$ dùng Timer 0, $f_{osc} = 4MHz$, chọn prescaler =2.

VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

Void DELAY100us(void)

{

//Bước 1: giá trị cần ghi vào thanh ghi TMR0

TMR0 = 206;

// Bước 2: Xóa cờ báo tràn

TMR0IF = 0;

// Bước 3: Chọn chế độ hoạt động của Timer 0

T0CS = 0; // Chế độ định thời

GIE=PEIE =TMR0IE = 0; //cảm ngắt

PSA = 0; PS2 = 0; PS1 = 0; PS0 = 0; // chọn prescaler =2

// Bước 4 : Kiểm tra cờ TMR0IF

while(TMR0IF==0);

}

VÍ DỤ MINH HỌA

- VÍ DỤ 2 : Viết chương trình con tạo trễ $t_{delay} = 250\mu s$ dùng Timer 0, $f_{osc} = 4MHz$, chọn prescaler =4
- VÍ DỤ 3 : Viết chương trình con tạo trễ $t_{delay} = 300\mu s$ dùng Timer 0, $f_{osc} = 4MHz$, chọn prescaler =8

VÍ DỤ 4 : Viết chương trình con tạo trễ

$t_{\text{delay}} = 0,3\text{s}$ dùng Timer 0, $f_{\text{OSC}} = 8\text{MHZ}$, chọn prescaler =4

```
void delay100us(unsigned int n)
```

```
{     while(n--)
```

```
{
```

```
    TMR0 = 206;
```

```
    TMR0IF = 0;
```

```
    T0CS = 0;
```

```
    GIE=PEIE =TMR0IE = 0;
```

```
    PSA = 0; PS2 = 0; PS1 = 0; PS0 = 1; // chon prescaler =4
```

```
    while(TMR0IF==0); }
```

```
}
```

```
void delay0.3s(void)
```

```
{
```

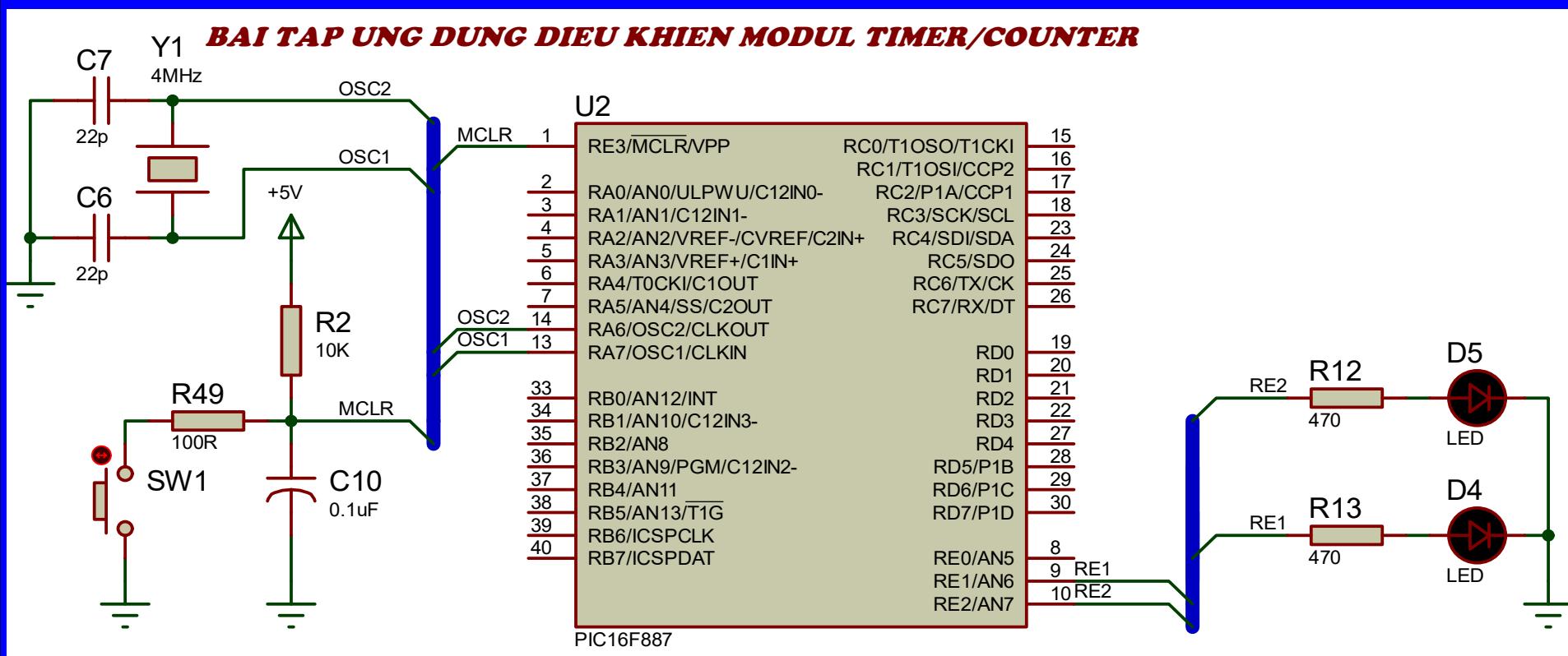
```
    delay100us(3000);
```

```
}
```

VÍ DỤ MINH HỌA

➤ **Ví dụ 5:** Dựa vào sơ đồ, viết chương trình điều khiển tạo sóng vuông có tần số 50Hz tại chân RE2 và 1KHz tại chân RE1. Sử dụng modul Timer 0 để tạo xung.

- **Sơ đồ nguyên lý:**



Tính toán giá trị thanh ghi ví dụ 5

$$T_1 = \frac{1}{f_1} = \frac{1}{50} = 0.02s$$

$$T_2 = \frac{1}{f_2} = \frac{1}{1000} = 0,001s$$

$$\frac{T_1}{T_2} = 20 \rightarrow T_1 = 20T_2 \quad \text{Nhận thấy } T_2 < T_1 \text{ suy ra :}$$

$$t_{delay} = \frac{T_2}{2} = \frac{0.001}{2} = 0.0005s = 500\mu s$$

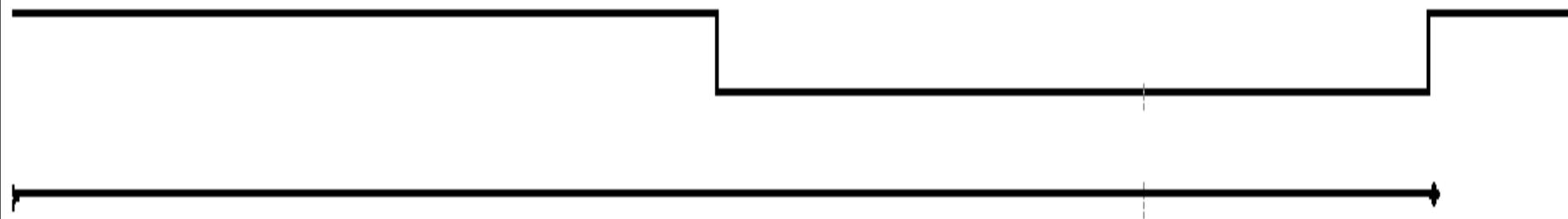
Từ đó suy ra sẽ viết hàm tạo trễ 500us dùng Timer 0

Tính toán giá trị thanh ghi ví dụ 5

$T_2=0.001\text{s}$

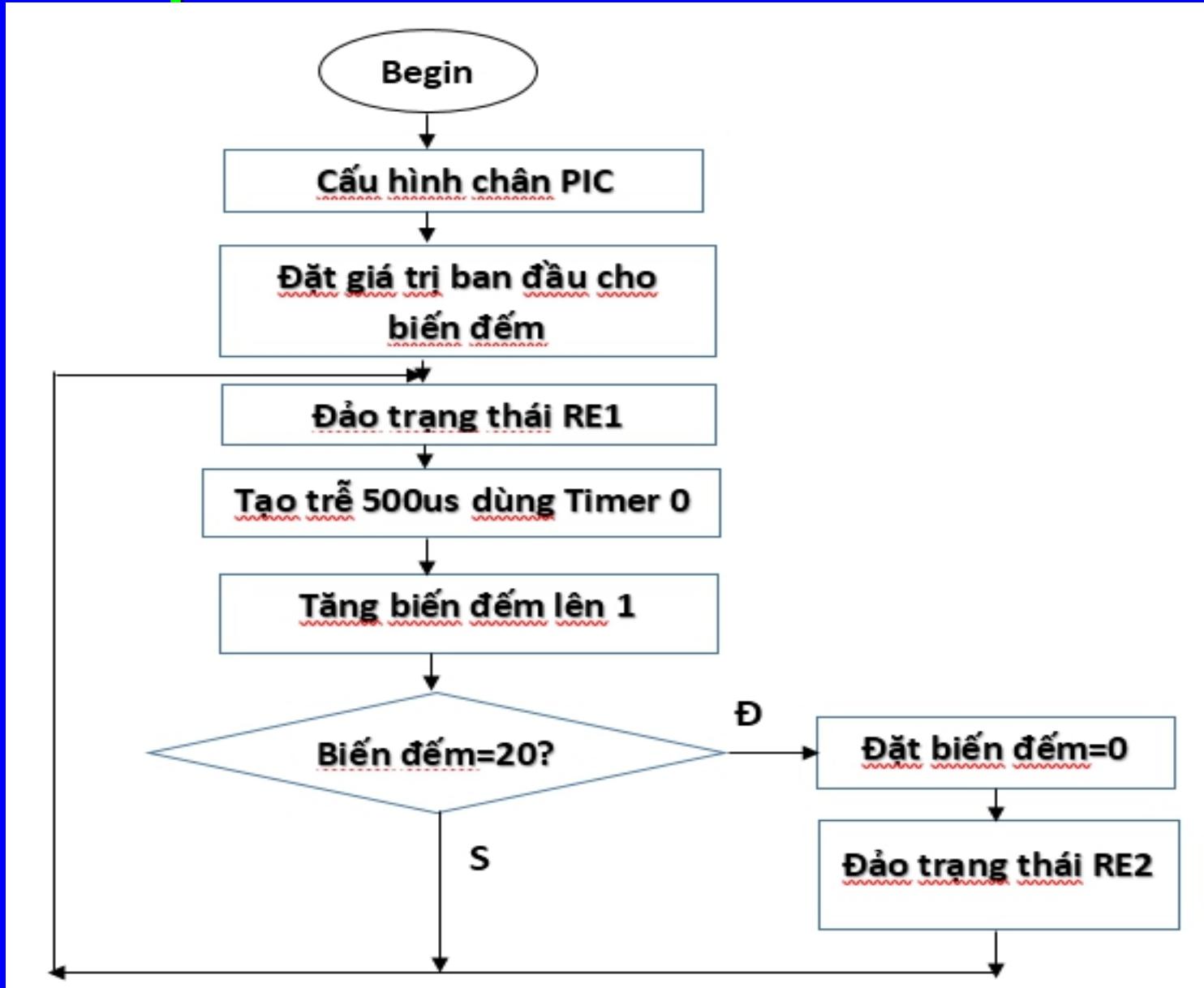


$$t_{delay} = T_2/2$$



$$T_1=0.02\text{s}=20T_2$$

Giải thuật



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

```
// Chuong trinh con tao tre 100us dung Timer 0
void delay100us(unsigned char n)
{
    while(n--)
    {
        TMR0 = 206; // buoc 1
        TMR0IF = 0; // buoc 2
        T0CS = 0; // buoc 3
        GIE=PEIE =TMR0IE = 0; //cam ngat
        PSA = 0; PS2 = 0; PS1 = 0; PS0 = 0; //prescaler=2
        while(TMR0IF==0); // buoc4
    }
}
```

//Chuong trinh chinh

void main (void)

{

ANSEL=ANSELH=0;

TRISE = 0;

unsigned char count=0;

while(1)

{

RE1=!RE1;

delay100us(5);

count=count+1;

if(count==20)

{

count=0;

RE2=!RE2;

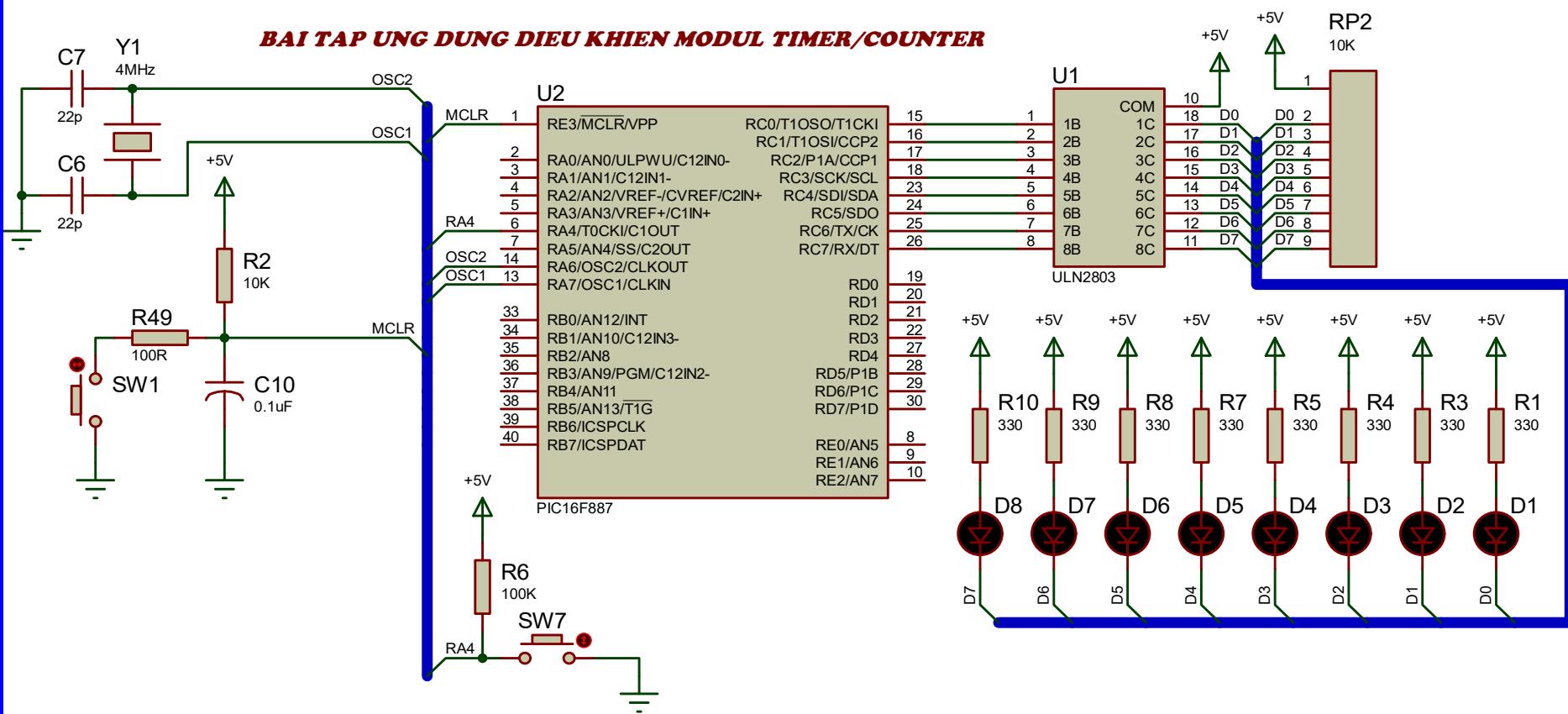
}

}

}

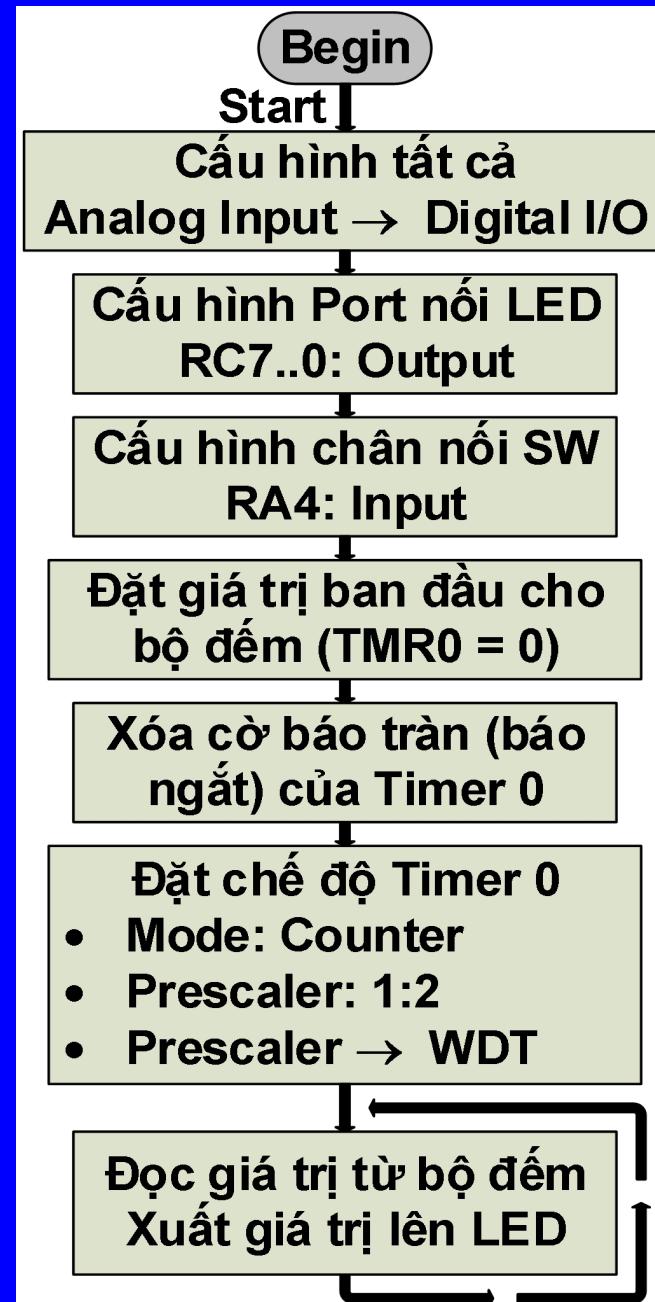
VÍ DỤ MINH HỌA

- **Ví dụ 6:** Dựa vào sơ đồ, viết chương trình điều khiển đếm số lần nhấn-nhả nút SW7 (0 → 255) được nối với ngõ vào của Counter 0 (T0CKI) và hiển thị giá trị này trên 8 LED đơn. Sử dụng modul Counter 0 để đếm.
- **Sơ đồ nguyên lý:**



VÍ DỤ MINH HỌA

- **Giải thuật:**



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

Source: C5_BT2_CodeC.c

SIM: C5_BT2_CodeC.pdsprj

- Chương trình (Hi-Tech C):

```
void main (void)
```

```
{
```

```
ANSEL = 0;
```

```
ANSELH = 0;
```

```
TRISC = 0;
```

```
TRISA4 = 1;
```

```
TMR0 = 0;
```

```
TMR0IF = 0;
```

```
T0CS = 1;
```

```
T0SE = 0;
```

```
PSA = 1;
```

```
PS2 = 0;
```

```
PS1 = 0;
```

```
PS0 = 0;
```

```
while(1)
```

```
{
```

```
PORTC = TMR0;
```

```
}
```

```
}
```

BÀI TẬP ỨNG DỤNG TIMER0

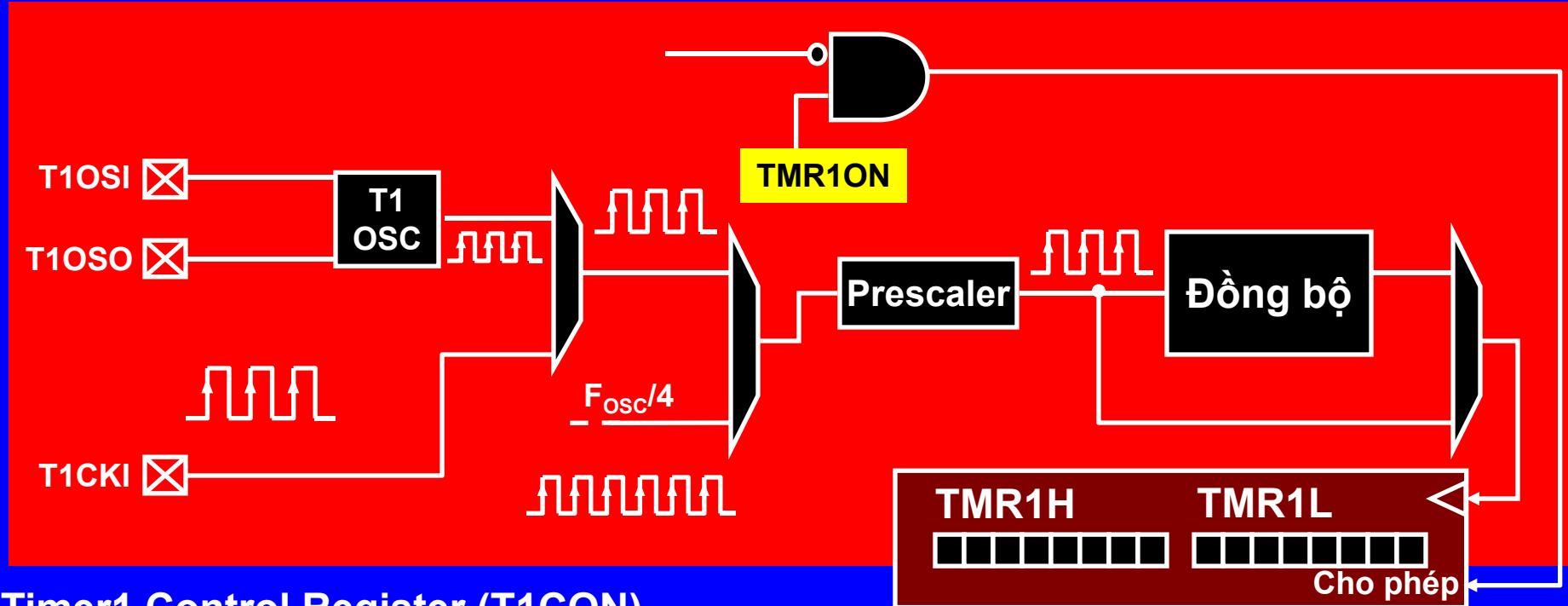
- **Bài tập 1:** Dựa vào sơ đồ của Ví dụ 1, viết chương trình điều khiển tạo sóng vuông có tần số 1Hz tại chân RE2 và 10KHz tại chân RE1. Sử dụng modul Timer 0 để tạo xung.

- **Bài tập 2:** Dựa vào sơ đồ của Ví dụ 2, viết chương trình điều khiển đếm giảm dần theo số lần nhấn nút SW7 ($100 \rightarrow 0$) được nối với ngõ vào của Counter 0 (T0CKI) và hiển thị giá trị này trên 8 LED đơn. Sử dụng modul Counter 0 để đếm.

TIMER 1

- Modul Timer 1 là một bộ định thời / đếm 16 bit với các tính năng:
 - Bộ định thời/đếm 16 bit (16-bit Timer/Counter)
 - Bộ chia tỉ lệ (Prescaler) 3 bit
 - Tạo tín hiệu ngắt khi Timer 1 bị tràn
 - Có khả năng lập trình để sử dụng nguồn xung clock bên trong hoặc bên ngoài
 - Có khả năng "đánh thức" vi điều khiển từ chế độ "ngủ" nhờ có thêm bộ dao động bên ngoài

SƠ ĐỒ KHÓI TIMER 1



Timer1 Control Register (T1CON)

Kích hoạt Timer 1
1 = Cho phép Timer 1
0 = Cấm Timer 1

| T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON |
|--------|--------|---------|---------|---------|--------|--------|--------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

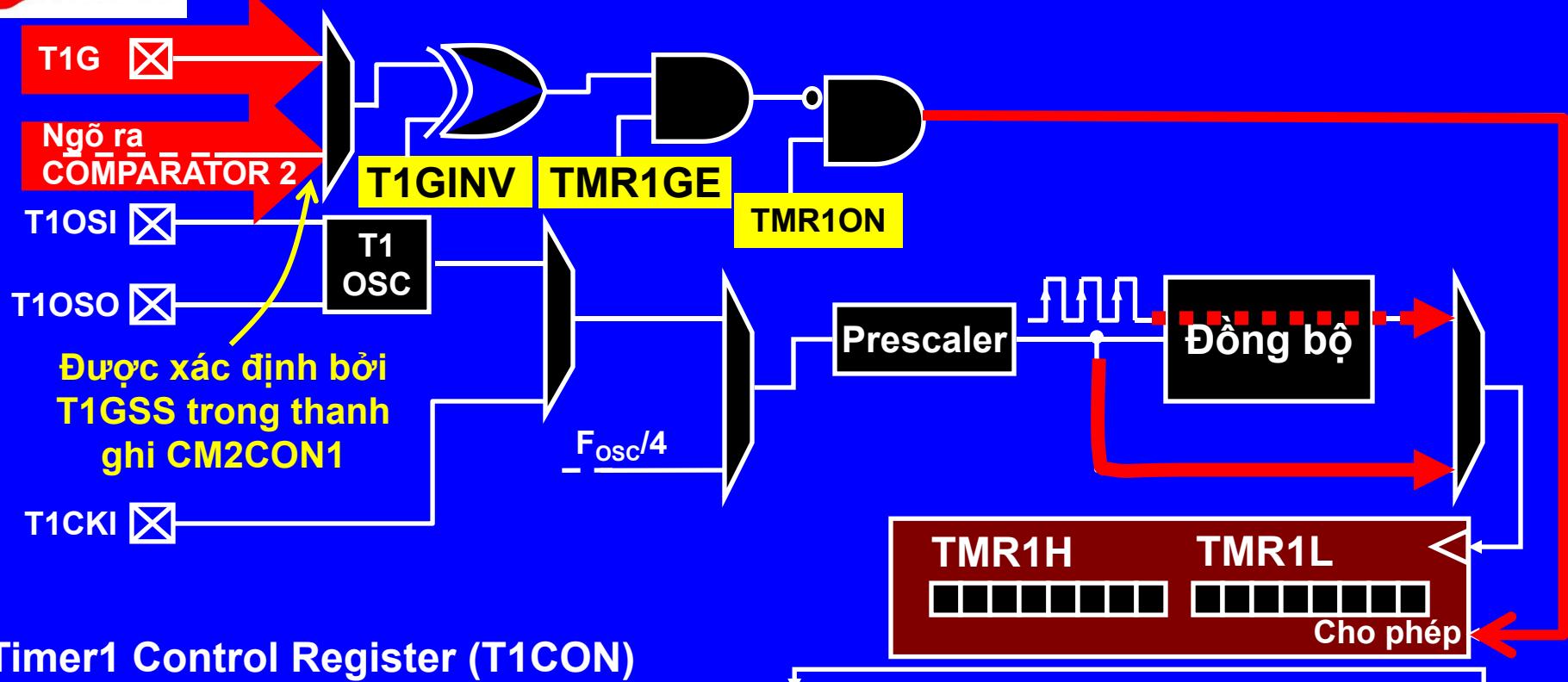
T1CKPS1 T1CKPS0 Tỉ lệ

| | | |
|---|---|-----|
| 1 | 1 | 1:8 |
| 1 | 0 | 1:4 |
| 0 | 1 | 1:2 |
| 0 | 0 | 1:1 |

Chọn nguồn xung clock
1 = Bên ngoài (cạnh lên tại T1CKI)
0 = Bên trong ($F_{osc}/4$)

Cho phép bộ dao động LP
1 = Chọn T1OSC
0 = Chọn T1CKI

SƠ ĐỒ KHÓI TIMER 1



Timer1 Control Register (T1CON)

| | | | | | | | |
|--------|--------|---------|---------|---------|--------|--------|--------|
| T1GINV | TMR1GE | T1CKPS1 | T1CKPS0 | T1OSCEN | T1SYNC | TMR1CS | TMR1ON |
|--------|--------|---------|---------|---------|--------|--------|--------|

Cho phép kích cỗng Timer 1
 1 = Timer 1 bật nếu cỗng kích hoạt
 0 = Timer 1 luôn luôn đếm (cầm kích cỗng)

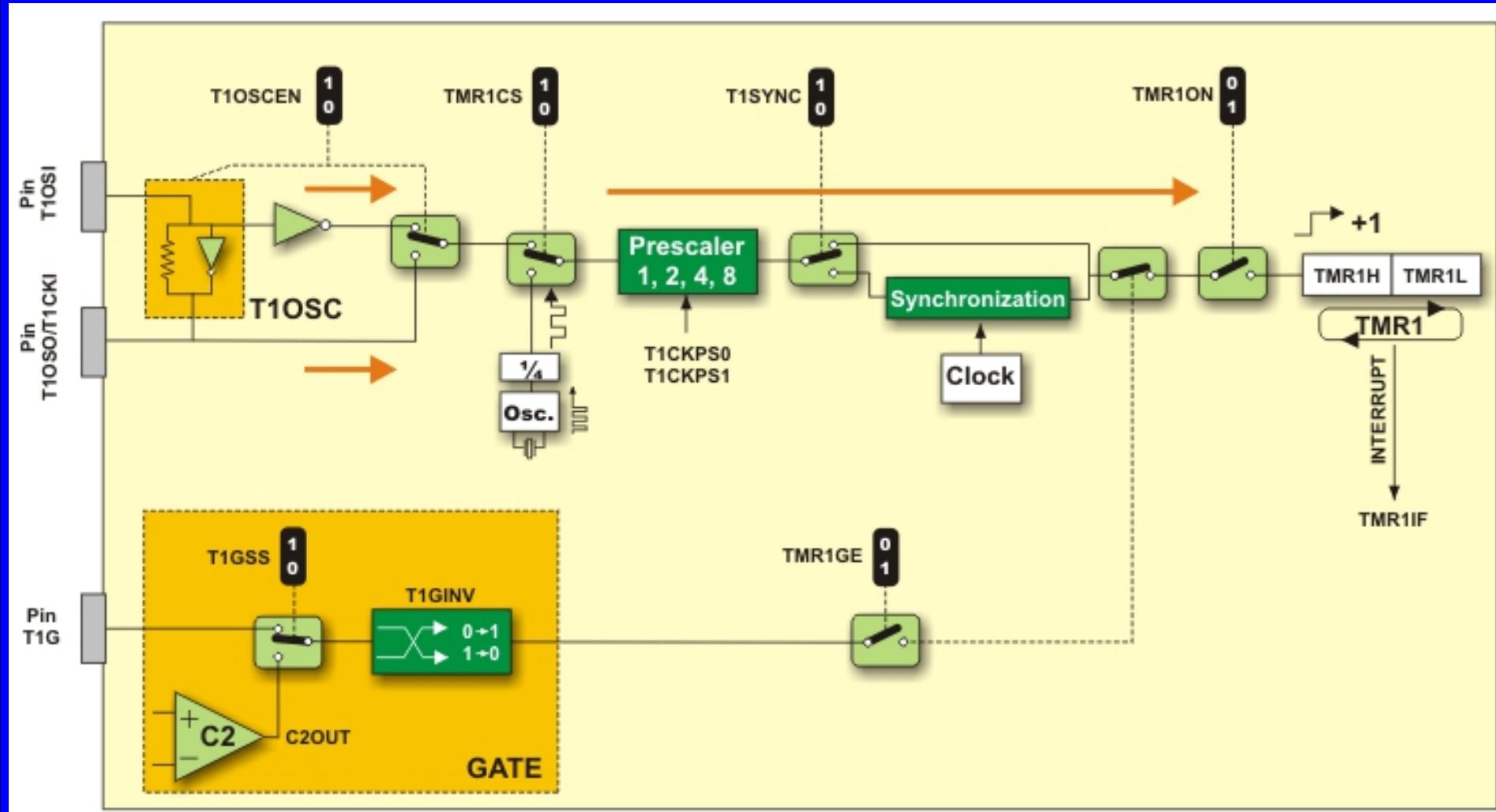
Nếu TMR1ON = 0 thì
 bỏ qua bit TMR1GE

Đảo mức kích cỗng Timer 1

1 = Timer 1 đếm khi cỗng có mức CAO
 0 = Timer 1 đếm khi cỗng có mức THẤP

Đồng bộ xung clock bên ngoài Timer 1
 với xung clock bên trong ($F_{osc}/4$)
 1 = Không đồng bộ
 0 = Đồng bộ

SƠ ĐỒ KHÓI TIMER 1

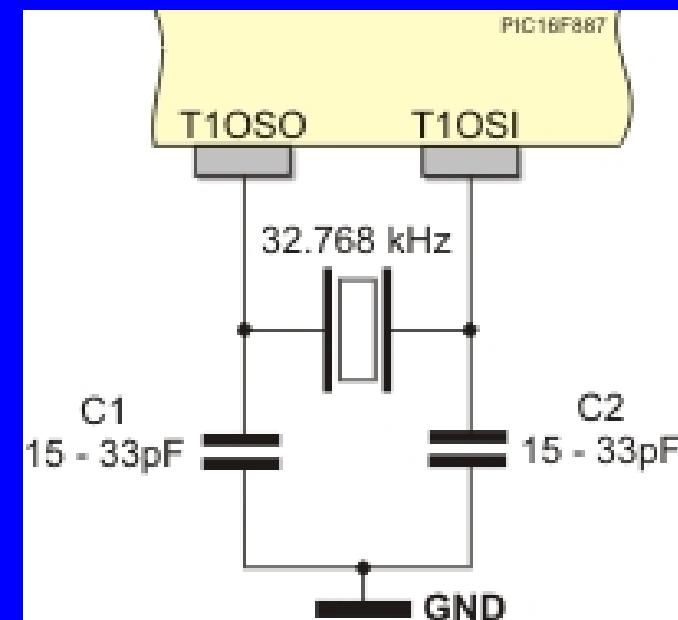


SƠ ĐỒ KHỐI TIMER 1

➤ Bộ dao động của Timer 1:

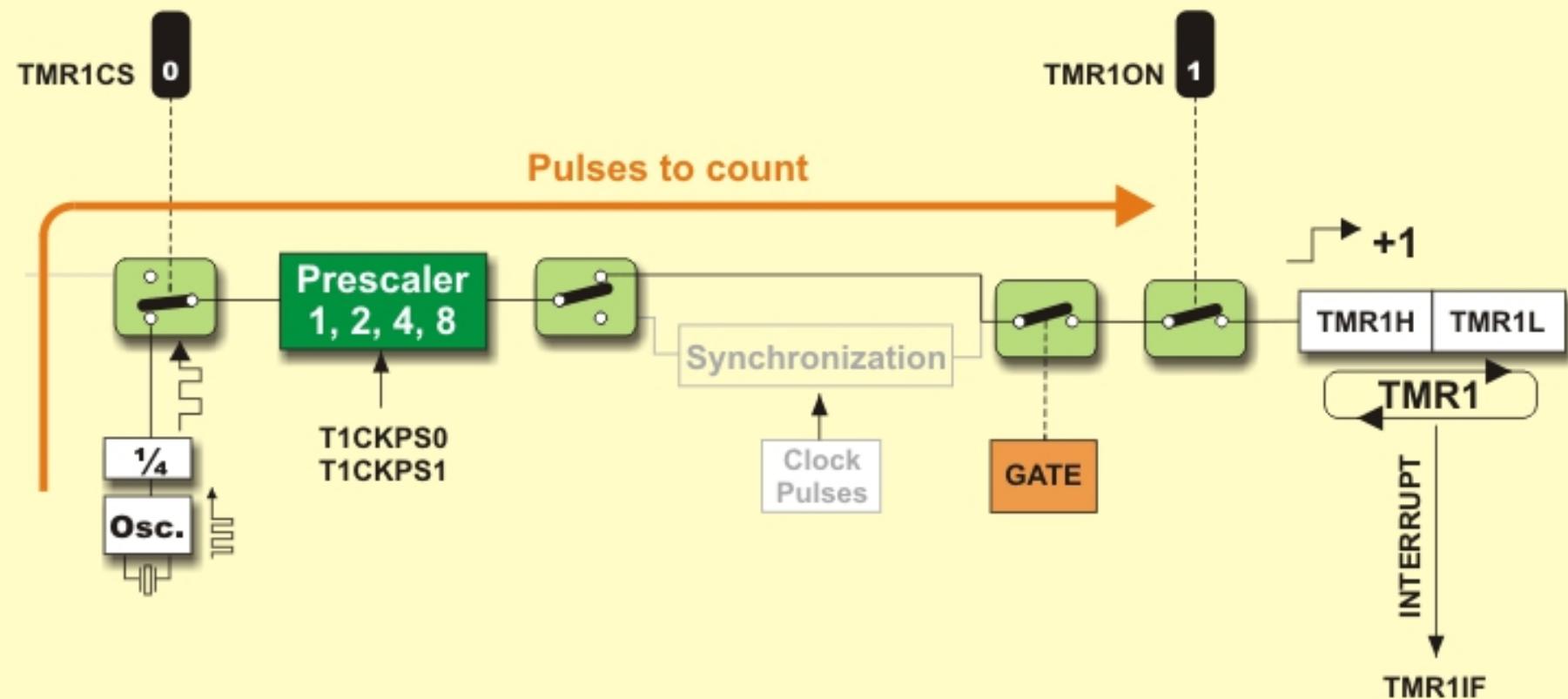
- Được kết nối thông qua chân RC0 và RC1
- Mạch điện bổ sung (xem hình) được thiết kế chủ yếu cho các hoạt động ở tần số thấp
- Không phụ thuộc vào xung clock bên trong nên có thể hoạt động ngay cả trong chế độ "ngủ"
- Sau khi được kích hoạt thì người sử dụng phải chờ khoảng vài ms để bộ dao động hoạt động ổn định.

| OSCILLATOR | FREQUENCY | C1 | C2 |
|------------|-----------|-------|-------|
| LP | 32 kHz | 33 pF | 33 pF |
| | 100 kHz | 15 pF | 15 pF |
| | 200 kHz | 15 pF | 15 pF |



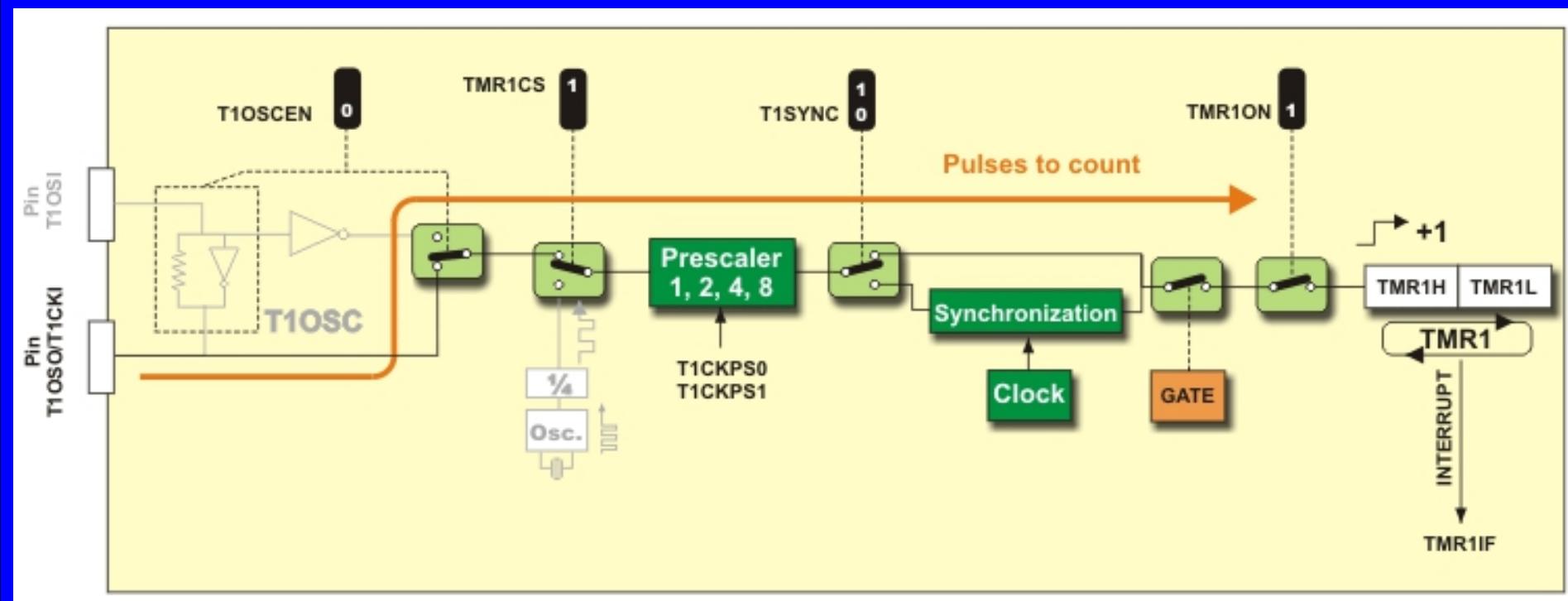
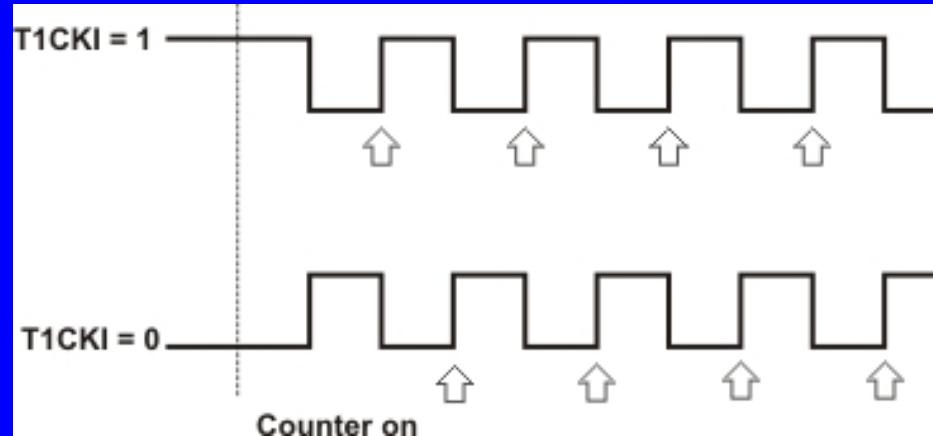
SƠ ĐỒ KHỐI TIMER 1

- Timer 1 hoạt động trong chế độ định thời gian (Timer)



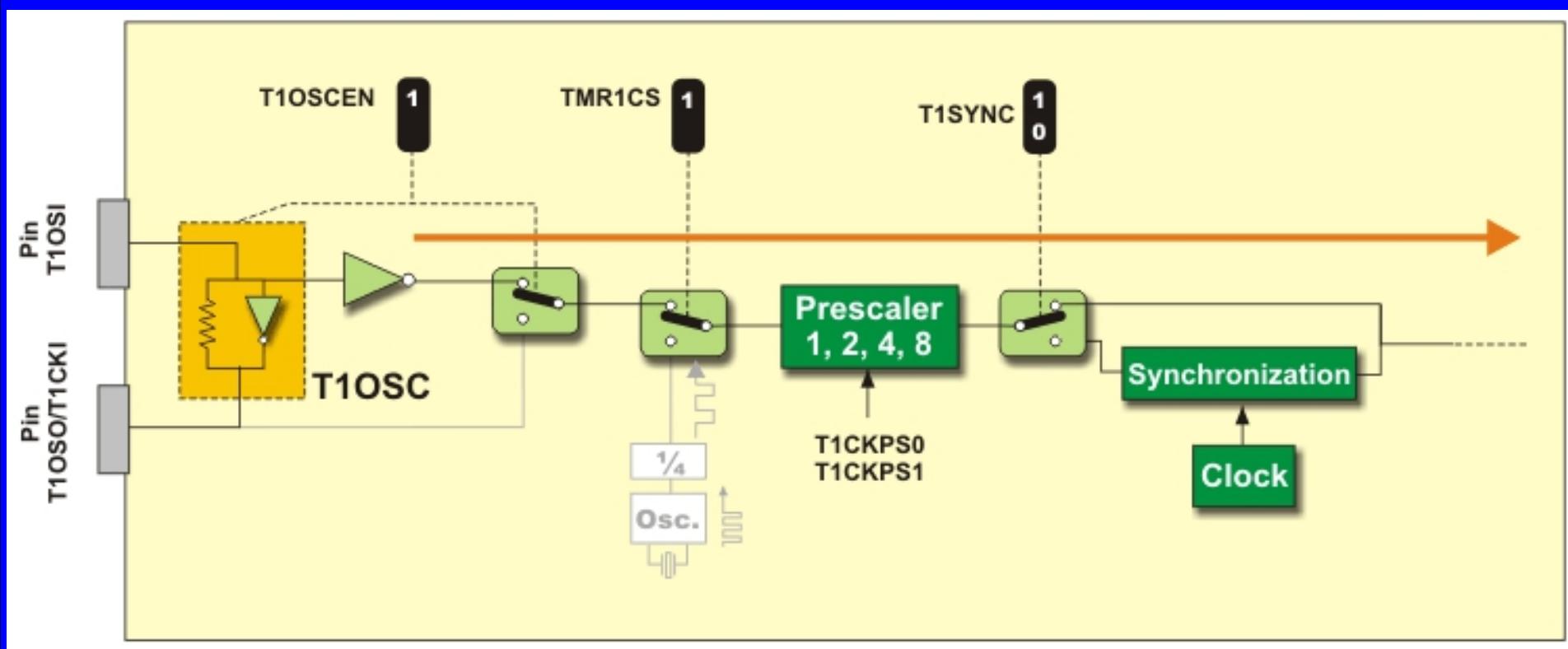
SƠ ĐỒ KHỐI TIMER 1

- Timer 1 hoạt động trong chế độ đếm sự kiện (Counter)



SƠ ĐỒ KHỐI TIMER 1

- Timer 1 hoạt động trong chế độ sử dụng bộ dao động bên ngoài (Timer Oscillator)



TIMER 1: MỘT SỐ ĐIỂM CẦN LƯU Ý

- Không thể tắt Prescaler mà chỉ có thể thay đổi giá trị tỉ lệ của nó
- Prescaler sẽ bị xóa bằng cách ghi dữ liệu vào các thanh ghi TMR1H và TMR1L
- Sự kiện ngắt khi Timer 1 tràn chỉ có thể "đánh thức" vi điều khiển khỏi chế độ "ngủ" trong trường hợp **sử dụng nguồn xung clock bên ngoài và không sử dụng tính năng đồng bộ.**

TIMER 1: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 1:** Ghi vào thanh ghi TMR1 (bao gồm TMR1H:TMR1L) giá trị tương ứng cho thời gian cần định thời

$$t_{DELAY} = (65536 - [TMR1]) \times 4 \times \frac{1}{f_{osc}} \times Pre$$

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 2, 4, 8)

[TMR1]: Giá trị cần ghi vào thanh ghi TMR1

$$0 \leq [TMR1] \leq 65535$$

$TMR1H =$ Byte cao của [TMR1]

$TMR1L =$ Byte thấp của [TMR1]

TIMER 1: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- Lưu ý:

- Công thức tính t_{DELAY} trên áp dụng cho trường hợp sử dụng nguồn xung clock bên trong
- Trong trường hợp sử dụng nguồn xung clock bên ngoài lấy từ bộ dao động Timer 1 (T1OSI và T1OSO) thì công thức tính t_{DELAY} sẽ như sau

$$t_{DELAY} = (65536 - [TMR1]) \times \frac{1}{f_{osc(T1)}} \times Pre$$

t_{DELAY} : Thời gian cần định thời (μs)

$f_{osc(T1)}$: Tần số bộ dao động Timer 1 (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 2, 4, 8)

[TMR1]: Giá trị cần ghi vào thanh ghi TMR1

TIMER 1: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 2:** Xóa cờ báo tràn (cờ ngắt) TMR1IF.
- **BƯỚC 3:** Chọn chế độ hoạt động của Timer 1
 - Chế độ định thời gian (Timer)
 - Kích hoạt hoặc vô hiệu hóa ngắt (tùy chọn)
 - Chọn giá trị tỉ lệ của Prescaler
- **BƯỚC 4:** Cho phép Timer 1 bắt đầu hoạt động
- **BƯỚC 5:** Xác định thời điểm Timer 1 bị tràn (đủ thời gian cần định thời của Timer) bằng cách
 - Kiểm tra cờ TMR1IF (nếu dùng thăm dò)
 - Xử lý ISR của Timer 1 (nếu dùng ngắt).

TIMER 1: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để đếm các sự kiện (Counter):

- **BƯỚC 1:** Xóa giá trị trong thanh ghi TMR1 (hoặc đặt giá trị ban đầu của bộ đếm)
- **BƯỚC 2:** Xóa cờ báo tràn (cờ ngắt) TMR1IF
- **BƯỚC 3:** Chọn chế độ hoạt động của Counter 1
 - Chế độ đếm sự kiện (Counter)
 - Kích hoạt hoặc vô hiệu hóa ngắt (tùy chọn)
 - Chọn giá trị tỉ lệ của Prescaler
 - Chọn tính năng đồng bộ hoặc không đồng bộ xung
- **BƯỚC 4:** Cho phép Counter 1 bắt đầu hoạt động
- **BƯỚC 5:** Đọc về và xử lý số xung đếm được trong thanh ghi TMR1 (TMR1H:TMR1L)
 - Dựa vào cờ báo tràn TMR1IF để xử lý các trường hợp số xung đếm vượt quá 65535.

MINH HỌA NGẮT CỦA TIMER 1

TMR1H

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

TMR1L

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
|---|---|---|---|---|---|---|---|

Main Code
Start

; Bắt đầu bằng việc xóa cờ ngắt Timer 1

banksel PIR1

bcf PIR1, TMR1IF

; Cho phép ngắt Timer 1

banksel PIE1

bsf PIE1, TMR1IE

; Cho phép ngắt toàn cục và ngắt ngoại vi

bsf INTCON, GIE

bsf INTCON, PEIE

Timer 1 tràn ↴

PIR1

| | | | | | | | |
|--|--|--|--|--|--|--|---|
| | | | | | | | 1 |
|--|--|--|--|--|--|--|---|

TMR1IF

PIE1

| | | | | | | | |
|--|--|--|--|--|--|--|---|
| | | | | | | | 1 |
|--|--|--|--|--|--|--|---|

TMR1IE

INTCON

| | | | | | | | |
|---|---|--|--|--|--|--|--|
| 1 | 1 | | | | | | |
|---|---|--|--|--|--|--|--|

GIE PEIE

;

MINH HỌA KHỞI ĐỘNG TIMER 1

; Xóa thanh ghi TMR1

| | |
|---------|-------|
| banksel | TMR1H |
| clrf | TMR1H |
| clrf | TMR1L |

; Xóa cờ báo tràn (cờ ngắt) TMR1IF trong PIR1

| | |
|---------|------------|
| banksel | PIR1 |
| bcf | PIR1,TMRIF |

; Thiết lập giá trị thanh ghi T1CON để Timer 1

; sử dụng xung clock bên trong với Prescaler
; là 1:8, T1 OSC bị cấm và Timer 1 bị dừng

| | |
|-------|-------------|
| movlw | b'00110000' |
| movwf | T1CON |

; Timer 1 bắt đầu hoạt động (giá trị tăng lên)

| | |
|-----|---------------|
| bsf | T1CON, TMR1ON |
|-----|---------------|

; Không dùng ngắt Timer 1, thực hiện thăm

; dò đối với bit báo tràn TMR1IF

| | |
|-------|--------------|
| btfss | PIR1, TMR1IF |
| goto | \$-1 |

< Tiếp theo >

TMR1H

TMR1L

TIMER 1 TRÀN !!!

INTCON

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

PIR1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

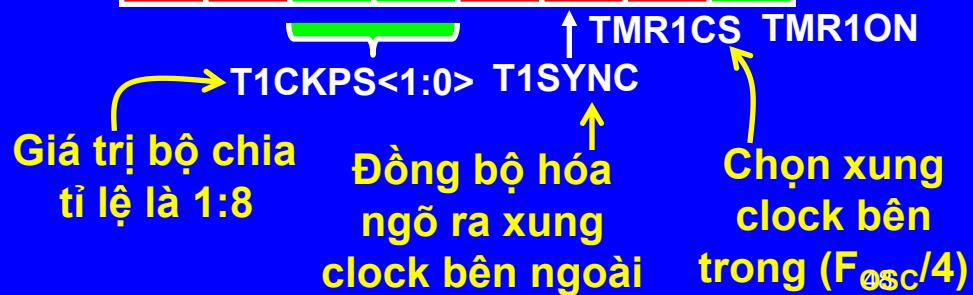
PIE1

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
|---|---|---|---|---|---|---|---|

TMR1IE

T1CON

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
|---|---|---|---|---|---|---|---|

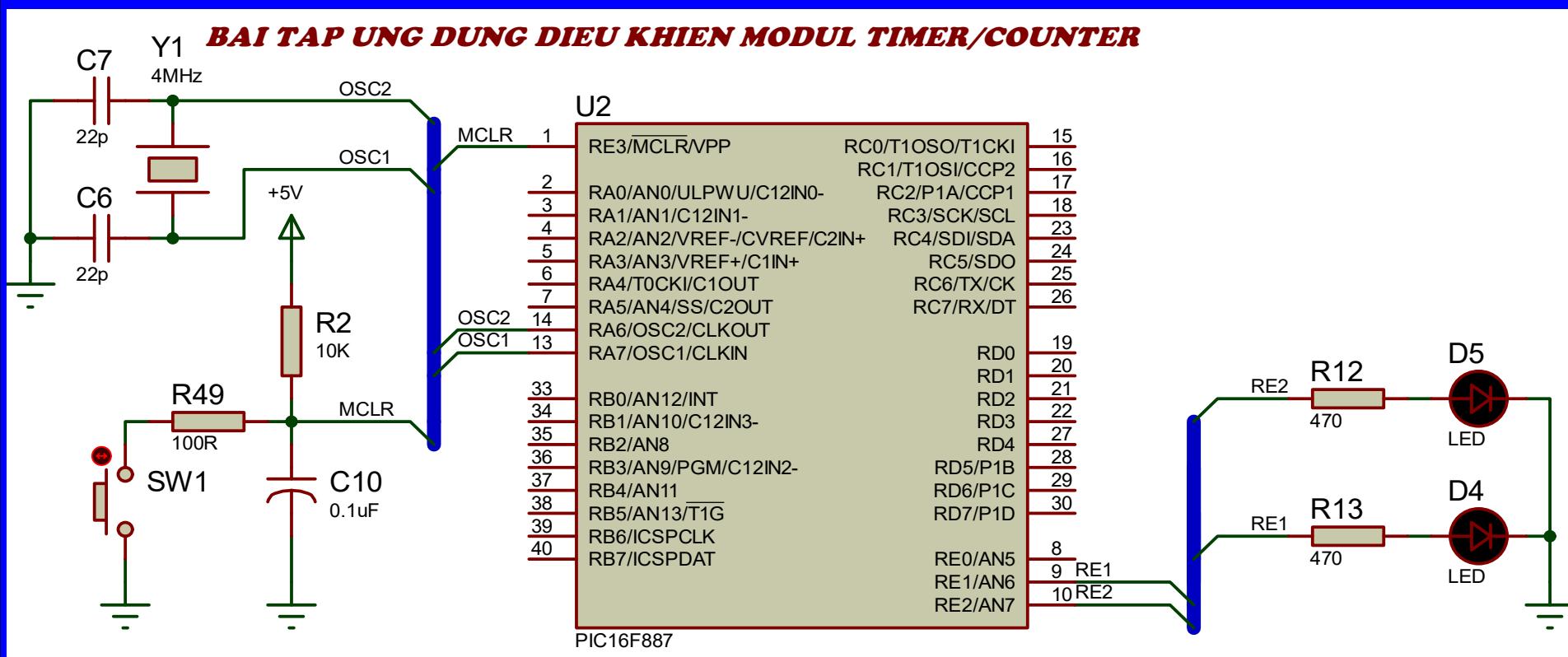


CHẾ ĐỘ ĐỊNH THỜI GIAN

VÍ DỤ MINH HỌA

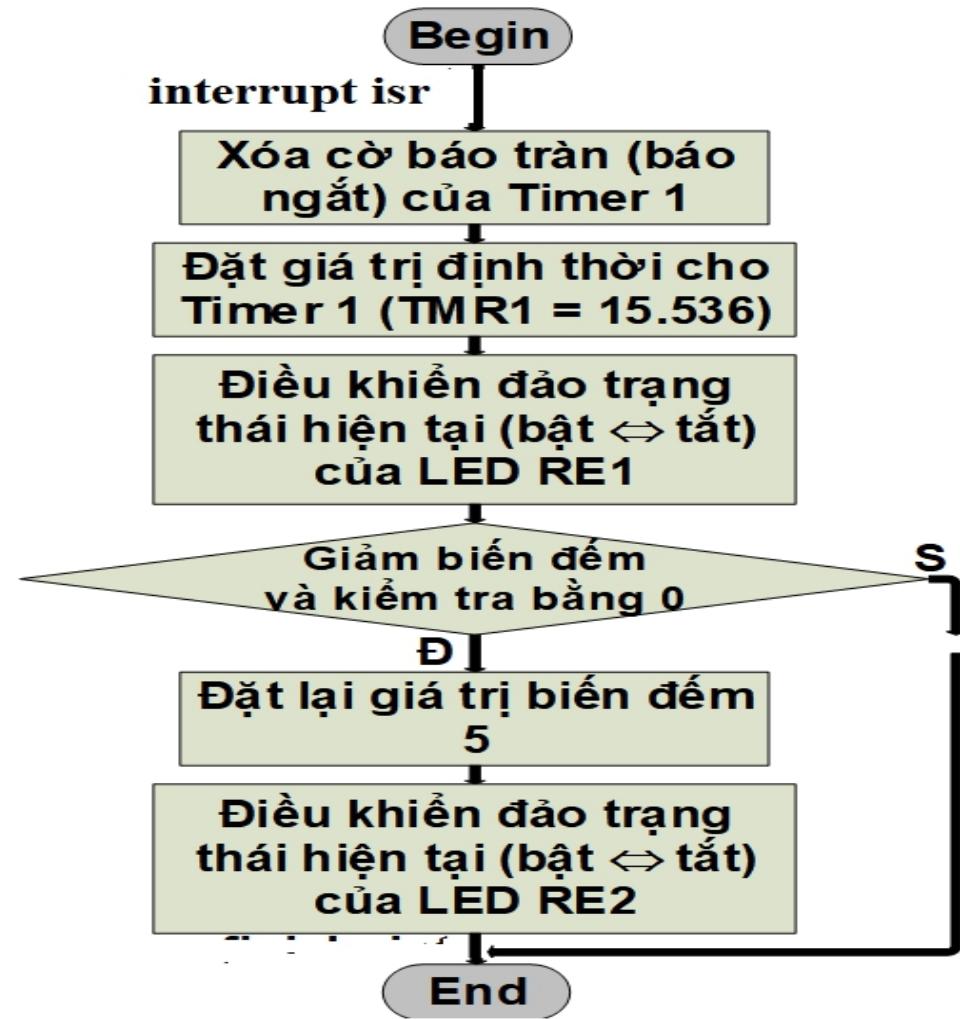
➤ **Ví dụ 1:** Dựa vào sơ đồ, viết chương trình điều khiển tạo sóng vuông có tần số 1Hz tại chân RE2 và 5Hz tại chân RE1. Sử dụng modul Timer 1 để tạo xung.

- **Sơ đồ nguyên lý:**



VÍ DỤ MINH HỌA

- Giải thuật:



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

unsigned char count;
void interrupt isr(void)

```
{  
    TMR1IF = 0;  
    TMR1H = 0x3C;  
    TMR1L = 0xB0;
```

```
    RE1 = !RE1;  
    count--;  
    if (count == 0)  
    {  
        count = 5;  
        RE2 = !RE2;  
    }  
}
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void main (void)
```

```
{
```

```
ANSEL = 0;
```

```
ANSELH = 0;
```

```
TRISE1 = 0;
```

```
TRISE2 = 0;
```

```
count = 5;
```

```
TMR1H = 0x3C;
```

```
TMR1L = 0xB0;
```

```
TMR1IF = 0;
```

```
T1CKPS0 = 1;
```

```
T1CKPS1 = 0;
```

```
TMR1CS = 0;
```

```
TMR1ON = 1;
```

```
TMR1IE = 1;
```

```
PEIE = 1;
```

```
GIE = 1;
```

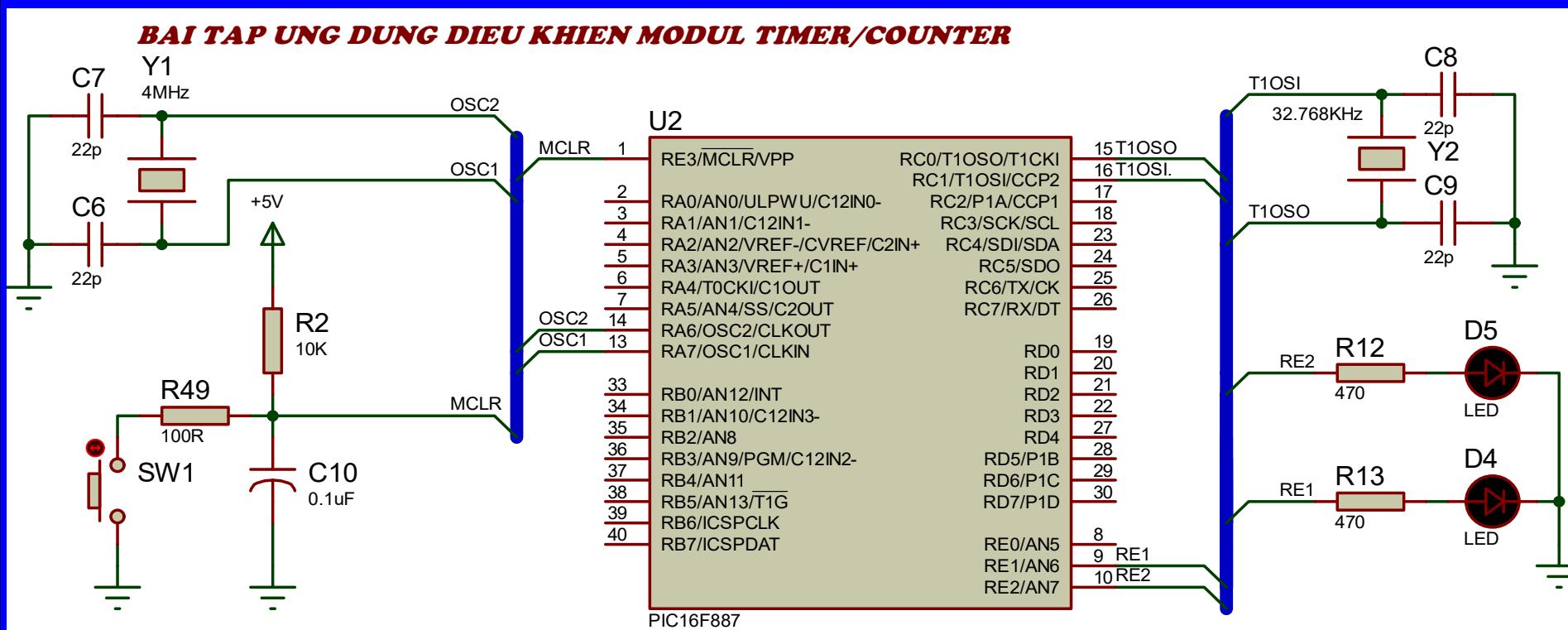
```
while(1);
```

```
}
```

VÍ DỤ MINH HỌA

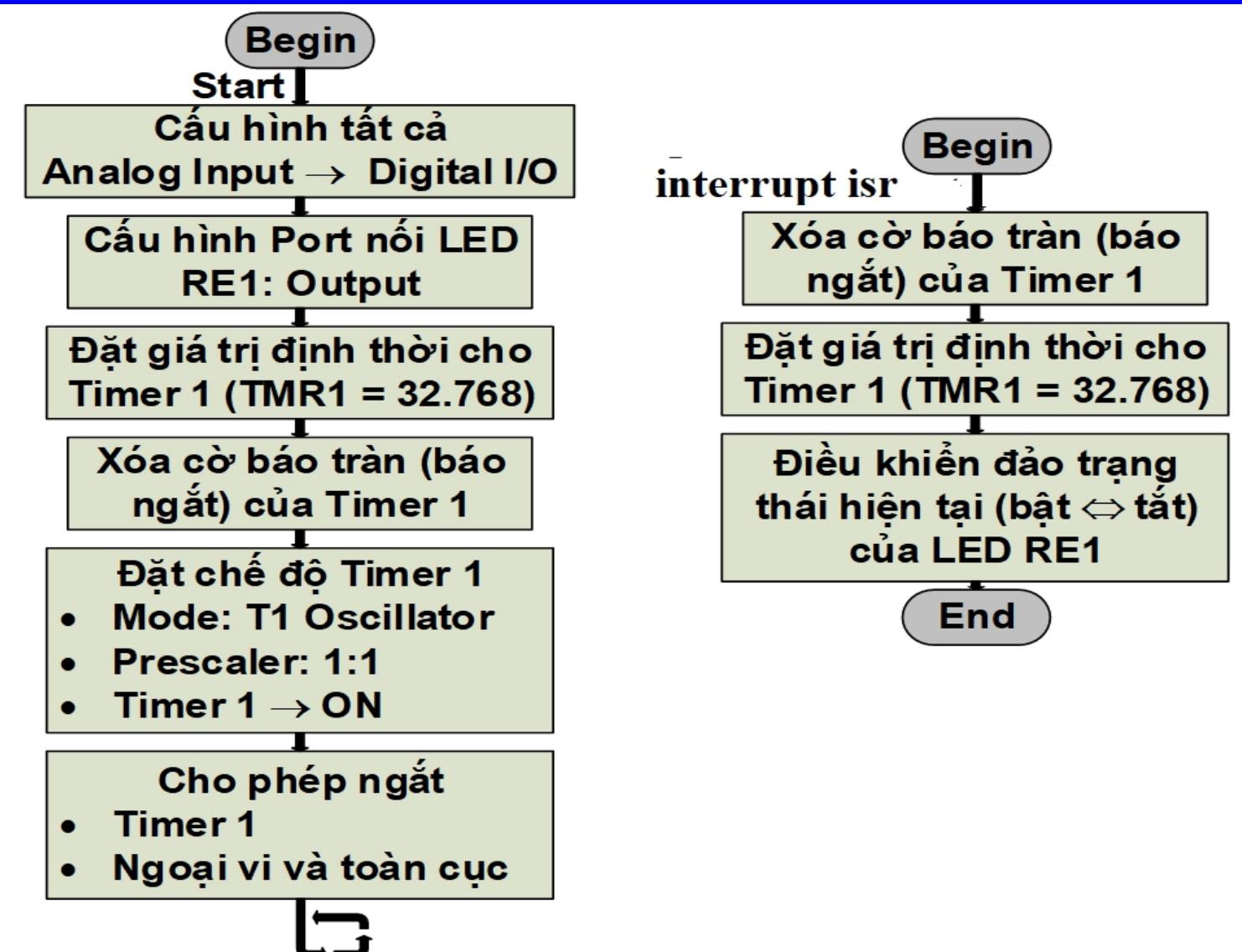
➤ **Ví dụ 2:** Dựa vào sơ đồ, viết chương trình điều khiển tạo sóng vuông có tần số 0,5Hz tại chân RE1. Sử dụng modul Timer 1 kết hợp với bộ dao động bên ngoài (T1 Oscillator) có tần số 32.768Hz.

- **Sơ đồ nguyên lý:**



VÍ DỤ MINH HỌA

- Giải thuật:



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void interrupt isr(void)
{
    TMR1IF = 0;
    TMR1H = 0x80;
    TMR1L = 0x00;

    RE1 = !RE1;
}

void main (void)
{
    ANSEL = 0;
    ANSELH = 0;

    TRISE1 = 0;
```

```
TMR1H = 0x80;
TMR1L = 0x00;

TMR1IF = 0;
TMR1CS = 1;
T1OSCEN = 1;
T1SYNC = 1;
T1CKPS0 = 0;
T1CKPS1 = 0;
TMR1ON = 1;
TMR1IE = 1;
PEIE = 1;
GIE = 1;
while(1);
}
```

BÀI TẬP ỨNG DỤNG TIMER1

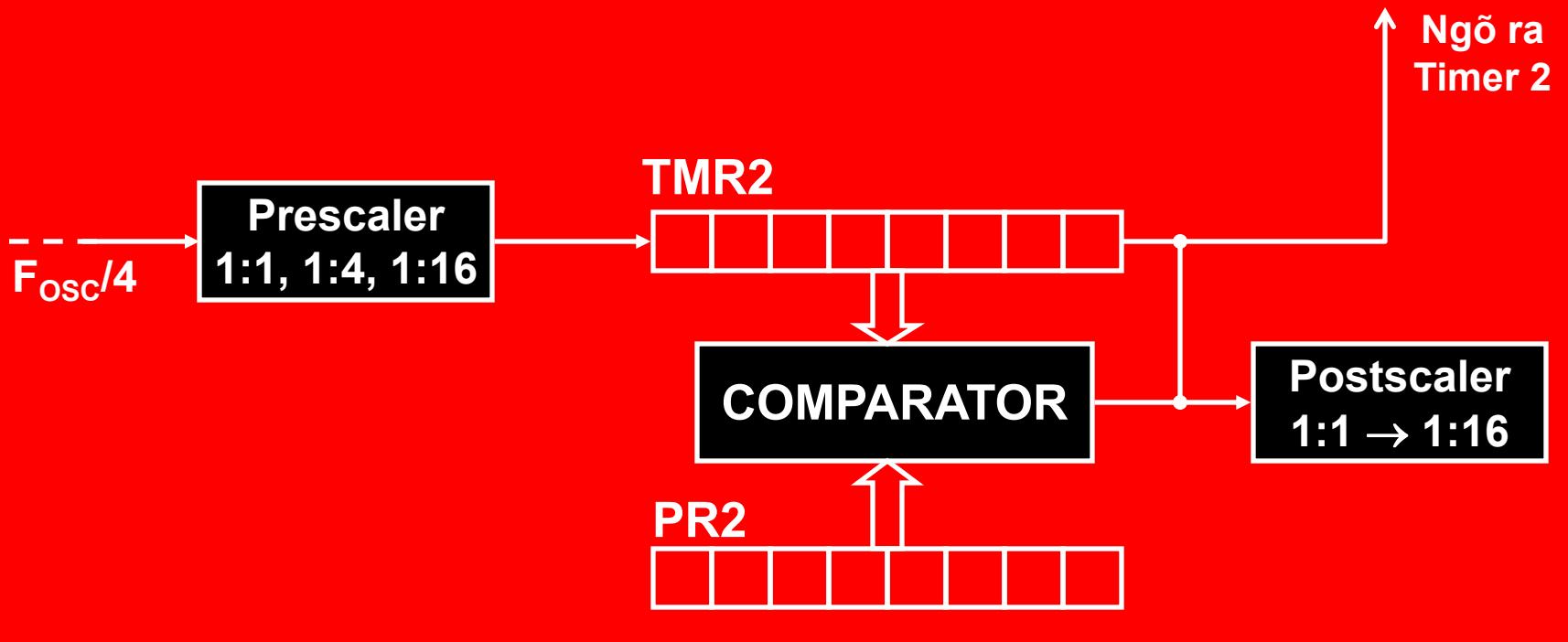
- **Bài tập 1:** Viết chương trình con tạo thời gian trễ có $t_{DELAY} = 100ms$, biết rằng $f_{osc} = 4MHz$. Sử dụng modul Timer 1 để tạo thời gian trễ.
- **Bài tập 2:** Dựa vào sơ đồ của Ví dụ 2, viết chương trình điều khiển đếm số lần nhấn-nhả nút SW7 ($0 \rightarrow 200$) được nối với ngõ vào của Counter 1 (T1CKI) và hiển thị giá trị này trên 8 LED đơn. Sử dụng modul Counter 1 để đếm.
- **Bài tập 3:** Viết chương trình con tạo thời gian trễ có $t_{DELAY} = 1s$. Sử dụng modul Timer 1 kết hợp với bộ dao động bên ngoài có tần số $32.768Hz$ (T1 Oscillator).

TIMER 2

➤ Modul Timer 2 là

- Bộ định thời 8 bit (8-bit Timer) hoạt động theo một cách thức mang tính chuyên biệt
- Bộ chia tỉ lệ trước (Prescaler) 4 bit
- Bộ chia tỉ lệ sau (Postscaler) 4 bit
- Tạo tín hiệu ngắt khi giá trị của TMR2 bằng với giá trị của PR2
- Không khả năng lập trình để sử dụng nguồn xung clock bên ngoài.

SƠ ĐỒ KHỐI TIMER 2



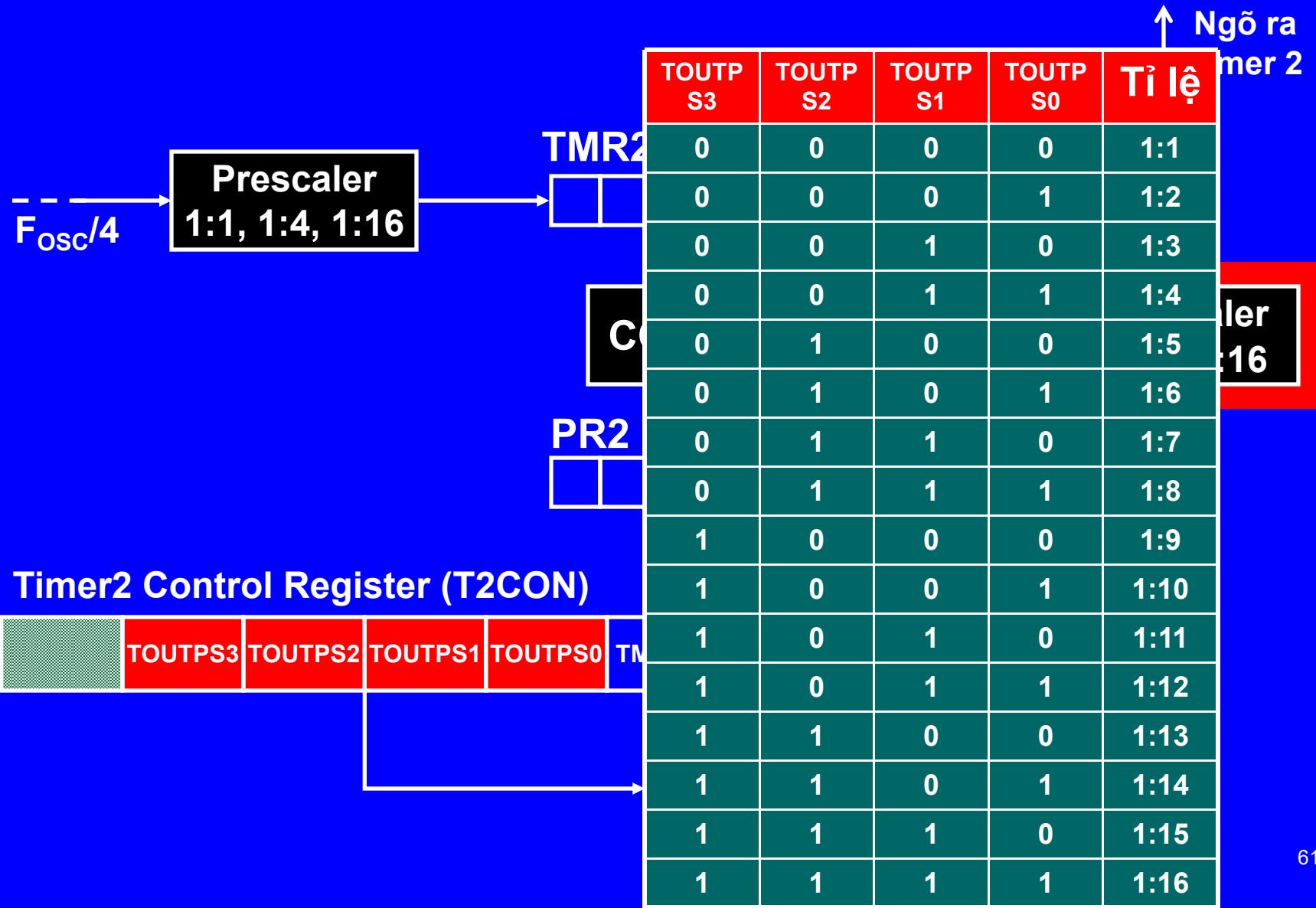
Timer2 Control Register (T2CON)



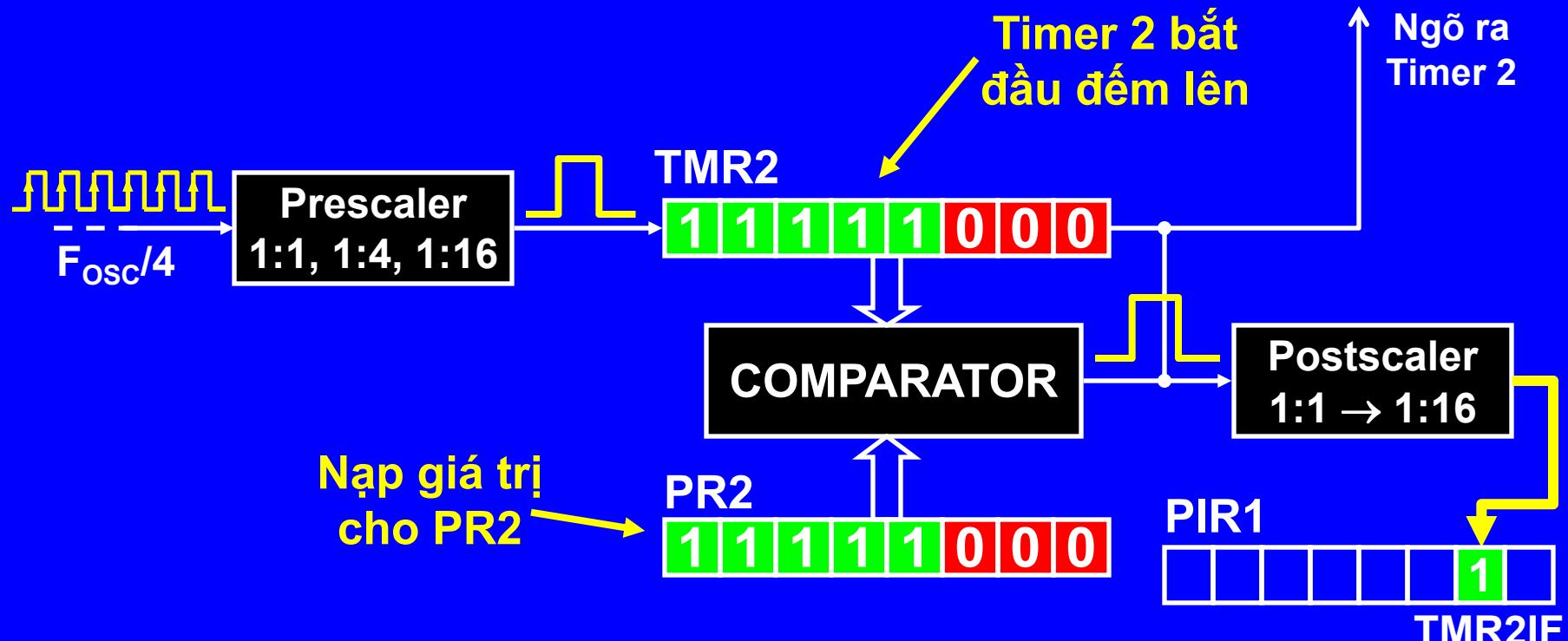
Kích hoạt Timer 2
1 = Cho phép Timer 2
0 = Cấm Timer 2

| T2CKPS1 | T2CKPS0 | Tỉ lệ |
|---------|---------|-------|
| 0 | 0 | 1:1 |
| 0 | 1 | 1:4 |
| 1 | X | 1:16 |

SƠ ĐỒ KHÓI TIMER 2



SƠ ĐỒ KHÓI TIMER 2

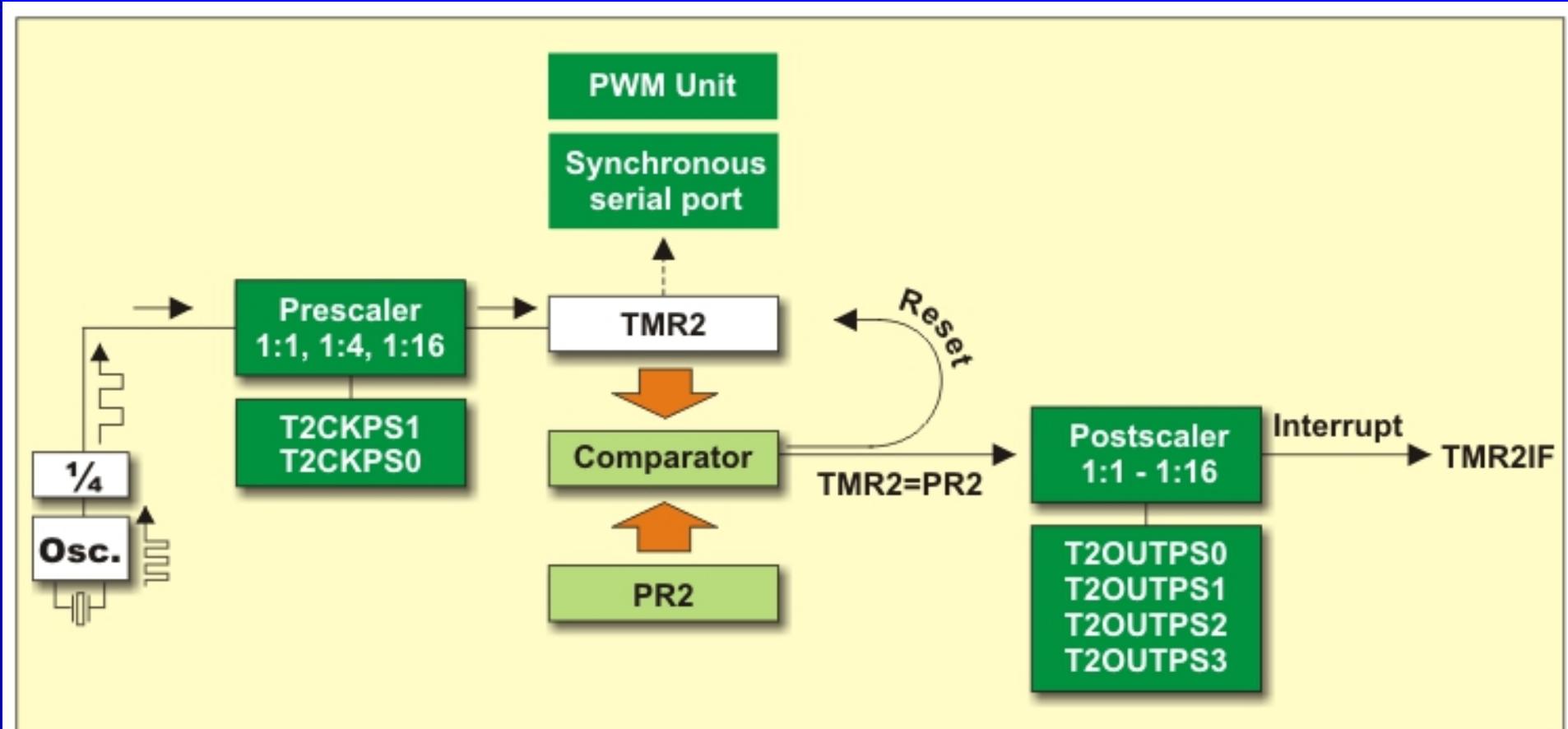


Timer2 Control Register (T2CON)



Còn được đặt (TMR2IF = 1) ngay khi giá trị bằng nhau nếu Postscaler = 1:1

SƠ ĐỒ KHỐI TIMER 2



TIMER 2: MỘT SỐ ĐIỂM CẦN LƯU Ý

- Khi mới cấp nguồn thì giá trị PR2 = FFh
- Prescaler và Postscaler sẽ bị xóa khi:
 - Ghi dữ liệu vào thanh ghi TMR2
 - Ghi dữ liệu vào thanh ghi T2CON
 - Bất cứ sự kiện reset nào (WDT, POR, BOR,...)
- Khi reset vi điều khiển thì Prescaler và Postscaler sẽ bị xóa
- TMR2 sẽ bị xóa khi ghi giá trị vào T2CON
- TMR2 sẽ tăng giá trị từ 00h đến khi bằng với giá trị trong PR2, sau đó TMR2 sẽ được reset về 0 vào chu kỳ tăng kế tiếp.

TIMER 2: MỘT SỐ ĐIỂM CẦN LƯU Ý

- Sự kiện ngắt khi giá trị TMR2 bằng giá trị PR2 không thể "đánh thức" vì điều khiển khởi chế độ "ngủ".
- Timer 2 ít khi được dùng để định thời gian hoặc đếm sự kiện như Timer 0,1. Nó thường dùng cho modul CCP để tạo xung PWM.

TIMER 2: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 1:** Ghi vào thanh ghi PR2 giá trị tương ứng cho thời gian cần định thời

$$t_{DELAY} = ([PR2] + 1) \times 4 \times \frac{1}{f_{osc}} \times Pre \times Post$$

t_{DELAY} : Thời gian cần định thời (μs)

f_{osc} : Tần số dao động (MHz)

Pre: Giá trị hệ số Prescaler (Pre = 1, 4, 16)

Post: Giá trị hệ số Postscaler (Pre = 1, 2, ..., 16)

[TMR2]: Giá trị cần ghi vào thanh ghi TMR2

[PR2]: Giá trị cần ghi vào thanh ghi PR2

$$0 \leq [PR2] \leq 255 \text{ và } [TMR2] = 0$$

TIMER 2: MỘT SỐ ĐIỂM CẦN LƯU Ý

➤ Để định thời gian (Timer):

- **BƯỚC 2:** Xóa cờ báo tràn (cờ ngắt) TMR2IF.
- **BƯỚC 3:** Chọn chế độ hoạt động của Timer 2
 - Kích hoạt hoặc vô hiệu hóa ngắt (tùy chọn)
 - Chọn giá trị tỉ lệ của Prescaler
 - Chọn giá trị tỉ lệ của Postscaler
- **BƯỚC 4:** Cho phép Timer 2 bắt đầu hoạt động
- **BƯỚC 5:** Xác định thời điểm Timer 2 bị tràn (đủ thời gian cần định thời của Timer) bằng cách
 - Kiểm tra cờ TMR2IF (nếu dùng thăm dò)
 - Xử lý ISR của Timer 2 (nếu dùng ngắt).

MINH HỌA KHỞI ĐỘNG TIMER 2

; Cắt ngắt Timer 2 trong thanh ghi PIE1. Đảm bảo
; rằng cờ ngắt Timer 2 trong PIR1 đã được xóa

```
banksel      PIE1
bcf          PIE1,TMR2IE
banksel      PIR1
bcf          PIR1,TMR2IF
```

; Khởi tạo giá trị thanh ghi T2CON để
; Postscaler = 1:15, Prescaler = 1:16, Timer 2 dừng

```
movlw      b'01110010'
movwf      T2CON
```

; Đảm bảo rằng thanh ghi TMR2 đã được xóa

```
banksel      TMR2
clrf        TMR2
```

; Đặt giá trị cho thanh ghi PR2

```
banksel      PR2
movlw      b'10000000'
movwf      PR2
```

; Cho Timer 2 chạy

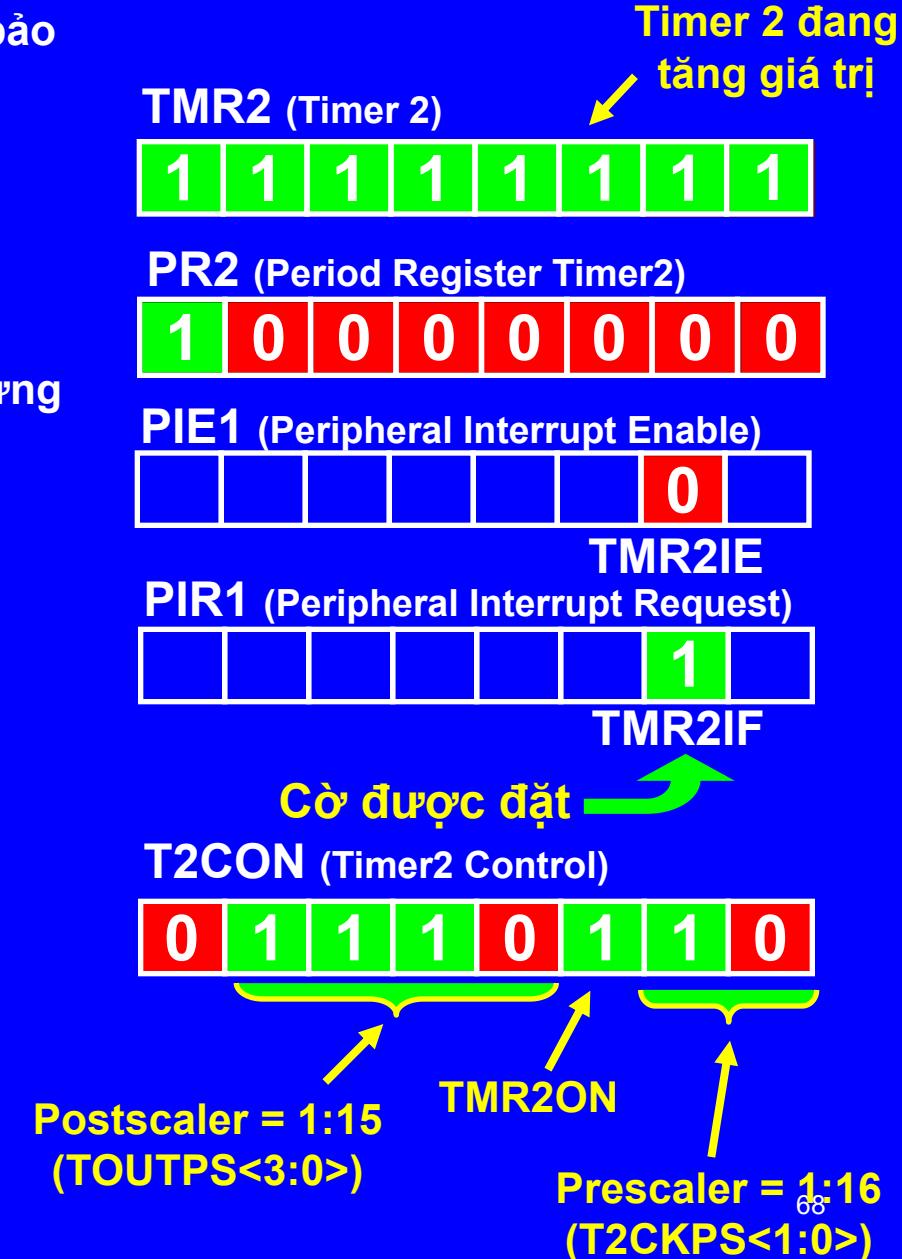
```
banksel      T2CON
bsf         T2CON,TMR2ON
```

; Không dùng ngắt Timer 2, thực hiện thăm

; dò đối với bit báo tràn TMR2IF

```
btfss      PIR1,TMR2IF
goto       $-1
```

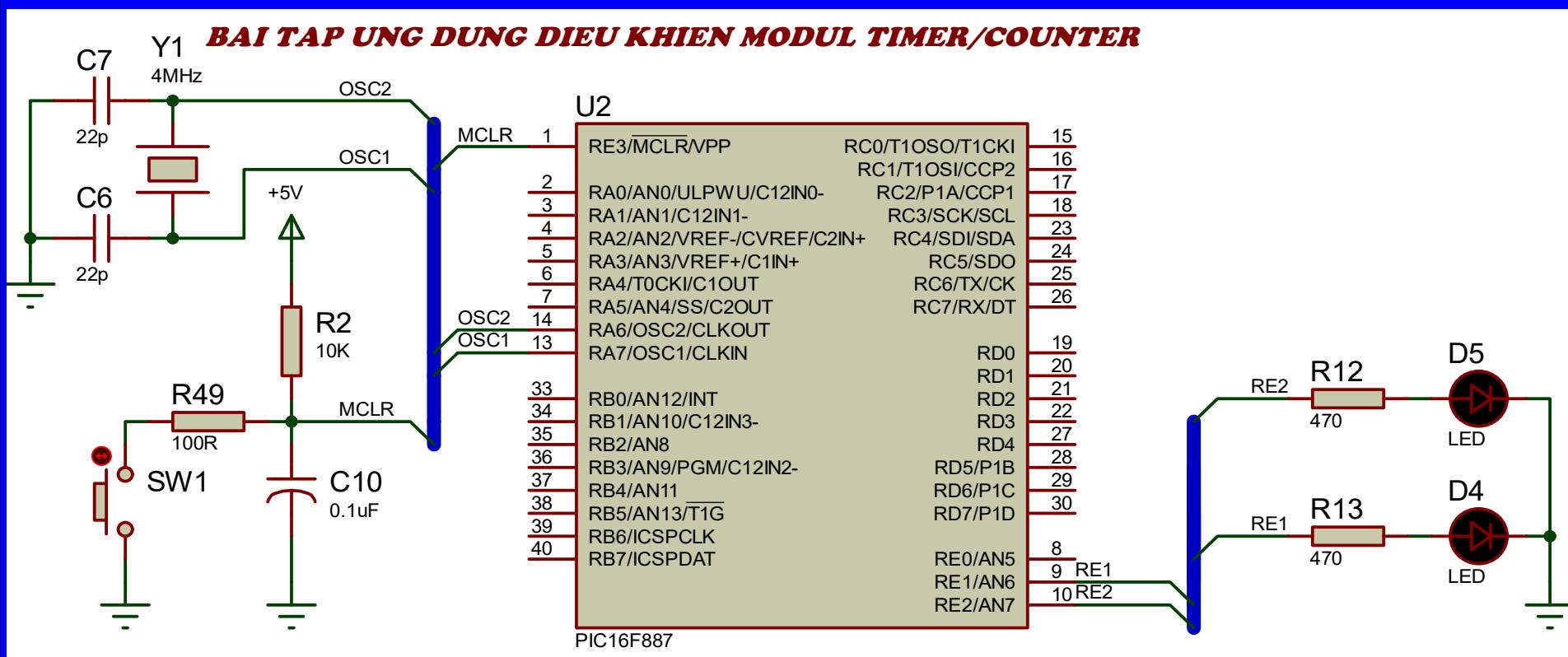
< Tiếp theo >



VÍ DỤ MINH HỌA

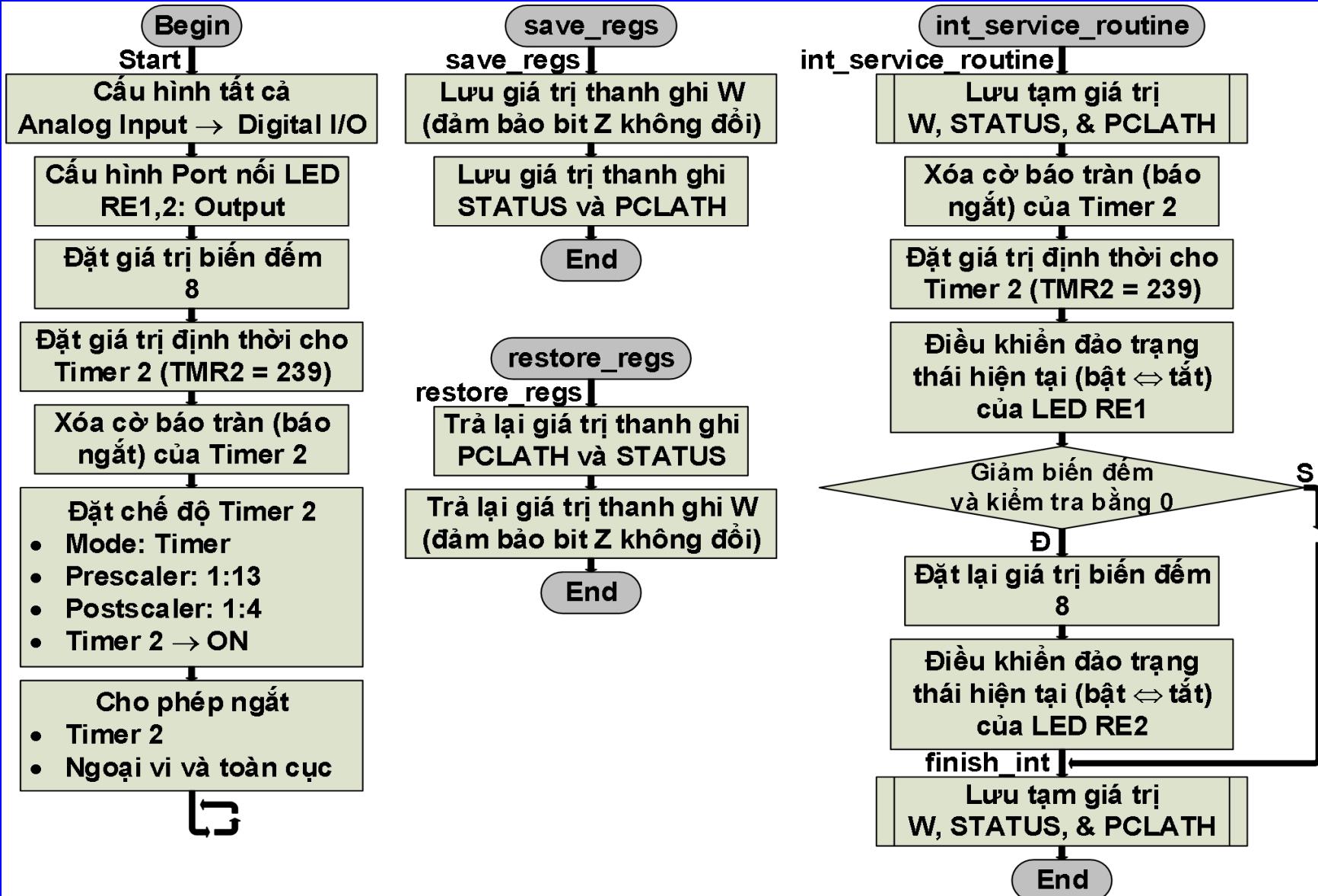
➤ **Ví dụ 1:** Dựa vào sơ đồ, viết chương trình điều khiển tạo sóng vuông có tần số 5Hz tại chân RE2 và 40Hz tại chân RE1. Sử dụng modul Timer 2 để tạo xung.

- **Sơ đồ nguyên lý:**



VÍ DỤ MINH HỌA

- Giải thuật:



VÍ DỤ MINH HỌA

- Cấu hình (Hi-Tech C):

```
_CONFIG(FOSC_HS & WDTE_OFF & PWRTE_ON &  
MCLRE_ON & CP_OFF & CPD_OFF & BOREN_OFF &  
IESO_OFF & FCMEN_OFF & LVP_OFF & DEBUG_ON);
```

```
#define _XTAL_FREQ 4000000
```

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

unsigned char count;
void interrupt isr(void)

{

TMR2IF = 0;

RE1 = !RE1;

count--;

if (count == 0)

{

count = 8;

RE2 = !RE2;

}

}

VÍ DỤ MINH HỌA

- **Chương trình (Hi-Tech C):**

```
void main (void)
```

```
{
```

```
ANSEL = 0;
```

```
ANSELH = 0;
```

```
TRISE1 = 0;
```

```
TRISE2 = 0;
```

```
count = 8;
```

```
PR2 = 239;
```

```
TMR2IF = 1;
```

```
TOUTPS3 = 1;
```

```
TOUTPS2 = 1;
```

```
TOUTPS1 = 0;
```

```
TOUTPS0 = 0;
```

```
T2CKPS1 = 0;
```

```
T2CKPS0 = 1;
```

```
TMR2ON = 1;
```

```
TMR2IE = 1;
```

```
PEIE = 1;
```

```
GIE = 1;
```

```
while(1);
```

```
}
```

BÀI TẬP ỨNG DỤNG TIMER2

- **Bài tập 1:** Viết chương trình con tạo thời gian trễ có $t_{DELAY} = 1\text{ms}$, biết rằng $f_{osc} = 4\text{MHz}$. Sử dụng modul Timer 2 để tạo thời gian trễ.
- **Bài tập 2:** Viết chương trình con tạo thời gian trễ có t_{DELAY} lớn nhất có thể đạt được, biết rằng $f_{osc} = 4\text{MHz}$. Sử dụng modul Timer 2 để tạo thời gian trễ.
- **Bài tập 3:** Dựa vào sơ đồ của Ví dụ 1, viết chương trình điều khiển tạo sóng vuông tần số 20Hz (chu kỳ nhiệm vụ $D = 40\%$) tại chân RE1. Sử dụng modul Timer 2 để tạo xung.