

Đại học Quốc Gia TP HCM
Trường Đại học Khoa Học Tự Nhiên
Khoa Vật Lý – Vật Lý Kỹ Thuật
Bộ môn Vật Lý Tin Học

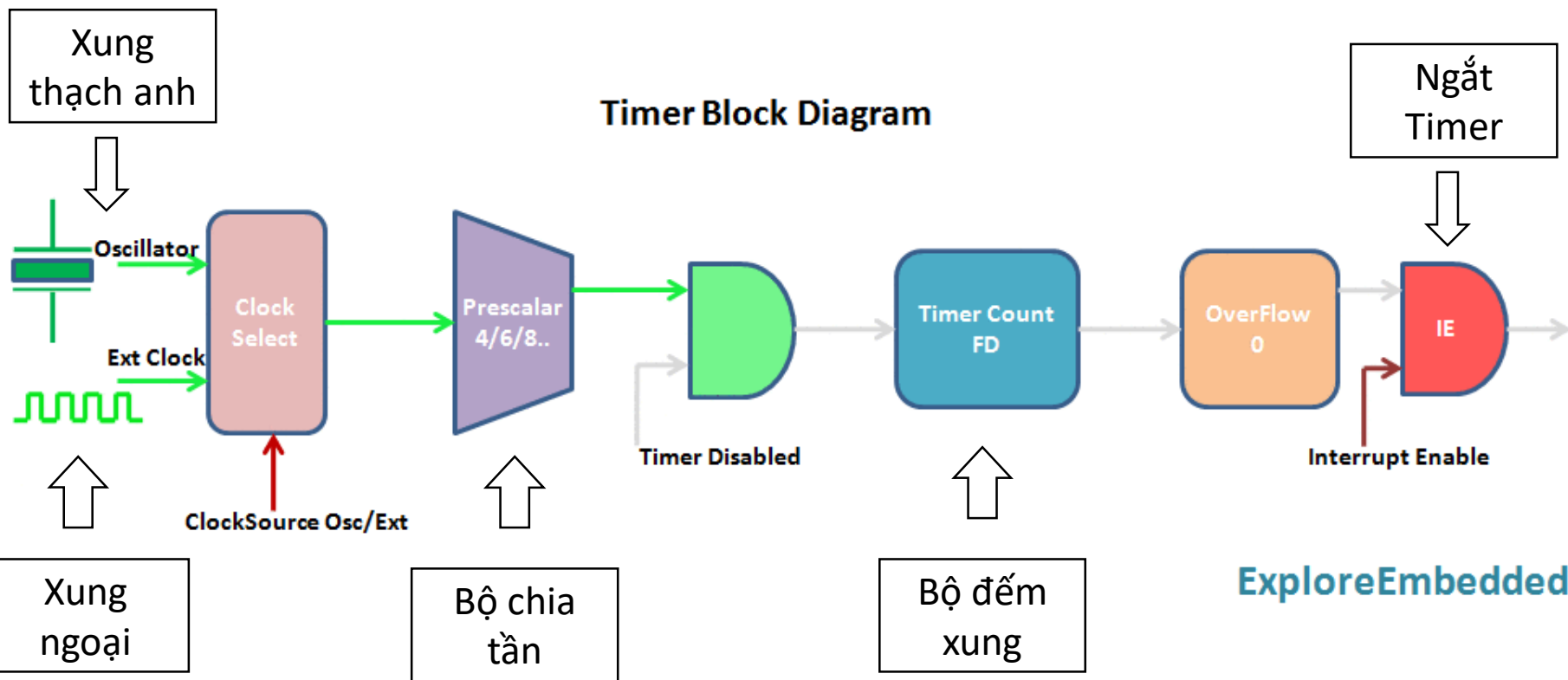


THỰC HÀNH VI ĐIỀU KHIỂN (PHY10605)

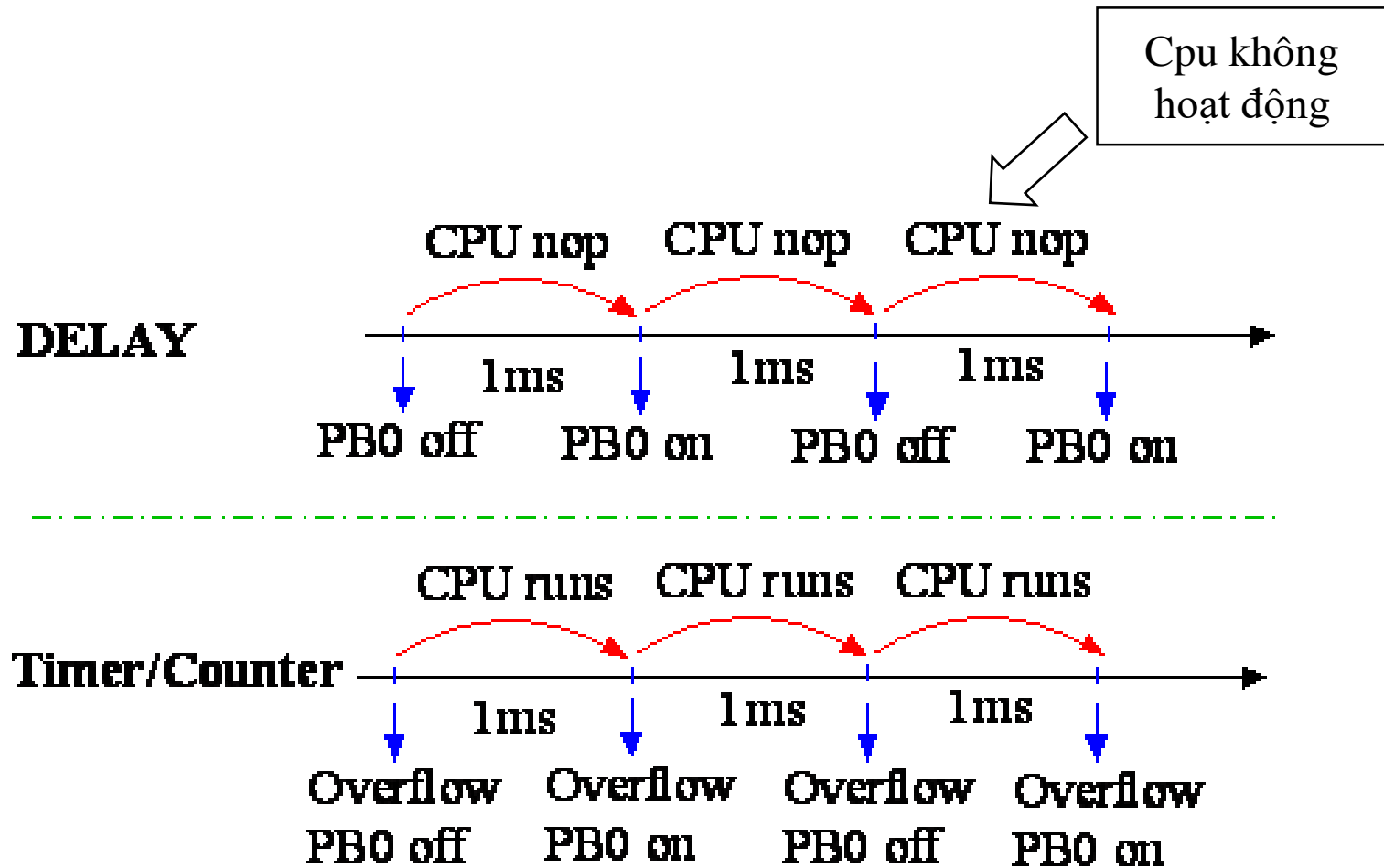
CBHD: Võ Hoàng Thủy Tiên
vhttien@hcmus.edu.vn
0937649914

Huỳnh Quốc Việt
hyqviet@hcmus.edu.vn
0349043204

1. Giới Thiệu TIMER



1. Giới Thiệu TIMER

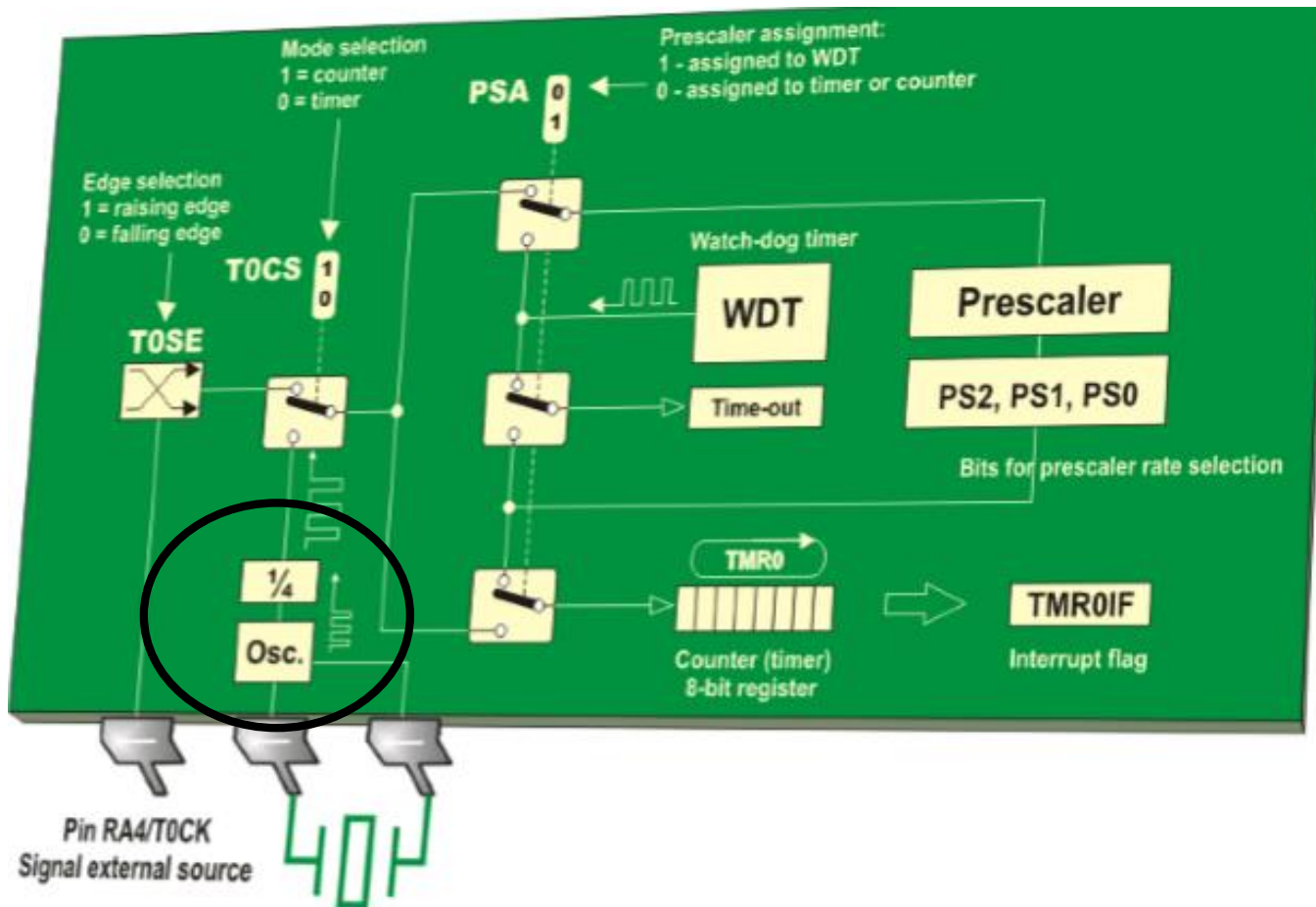


1. Giới Thiệu TIMER

| Timer | Size | Control Register | Count Register | Min Delay | Max Delay |
|---------------|--------|------------------|----------------|-----------|-----------|
| TIMER0 | 8-bit | OPTION_REG | TMR0 | 0.2usec | 13.107ms |
| TIMER1 | 16-bit | T1CON | TMR1H TMR1L | 0.2usec | 104.857ms |
| TIMER2 | 8-bit | T2CON | TMR2 | 0.2usec | 819usec |

- TMR0, TMR1H, TMR1L, TMR2 là các thanh ghi sẽ lưu giá trị đếm của bộ timer
- OPTION_REG, T1CON và T2CON: thanh ghi điều khiển chức năng của các Timer

2. Khảo Sát Timer0



2.Khảo Sát Timer0



- **T0CS:** bit chọn lựa xung vào
1-Dùng xung clock ngoại nối với chân T0CKI
0-Dùng xung clock nội trong chip
- **T0SE:** Bit xác định dạng đếm lên của TMR0
1-Giá trị đếm sẽ tăng theo cạnh lên của xung
0-Giá trị đếm sẽ tăng theo cạnh xuống của xung
- **PSA:**Bit xác định bộ chia tần
1-Bộ chia tần theo WDT (sẽ không tìm hiểu ở môn này)
0-Bộ chia tần theo TIMER0

2.Khảo Sát Timer0



- **PS2-PS0:** Bit chọn tỉ lệ chia

| 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 |

2.Khảo Sát Timer0

Ví dụ: Thiết kế Timer0 sử dụng xung clock nội và đếm theo cạnh lên của xung, sử dụng bộ chia tần của Timer0 và có bộ chia tần là 64.



→ Thanh ghi `OPTION_REG`=**00010101**

2. Khảo Sát Timer0



Thanh ghi INTCON

GIE: Bit cho phép sử dụng ngắt toàn cục

1-Cho phép ngắt toàn cục

0-Không cho phép ngắt toàn cục

TMR0IE: Bit ngắt tràn của TMR0

1-Cho phép ngắt

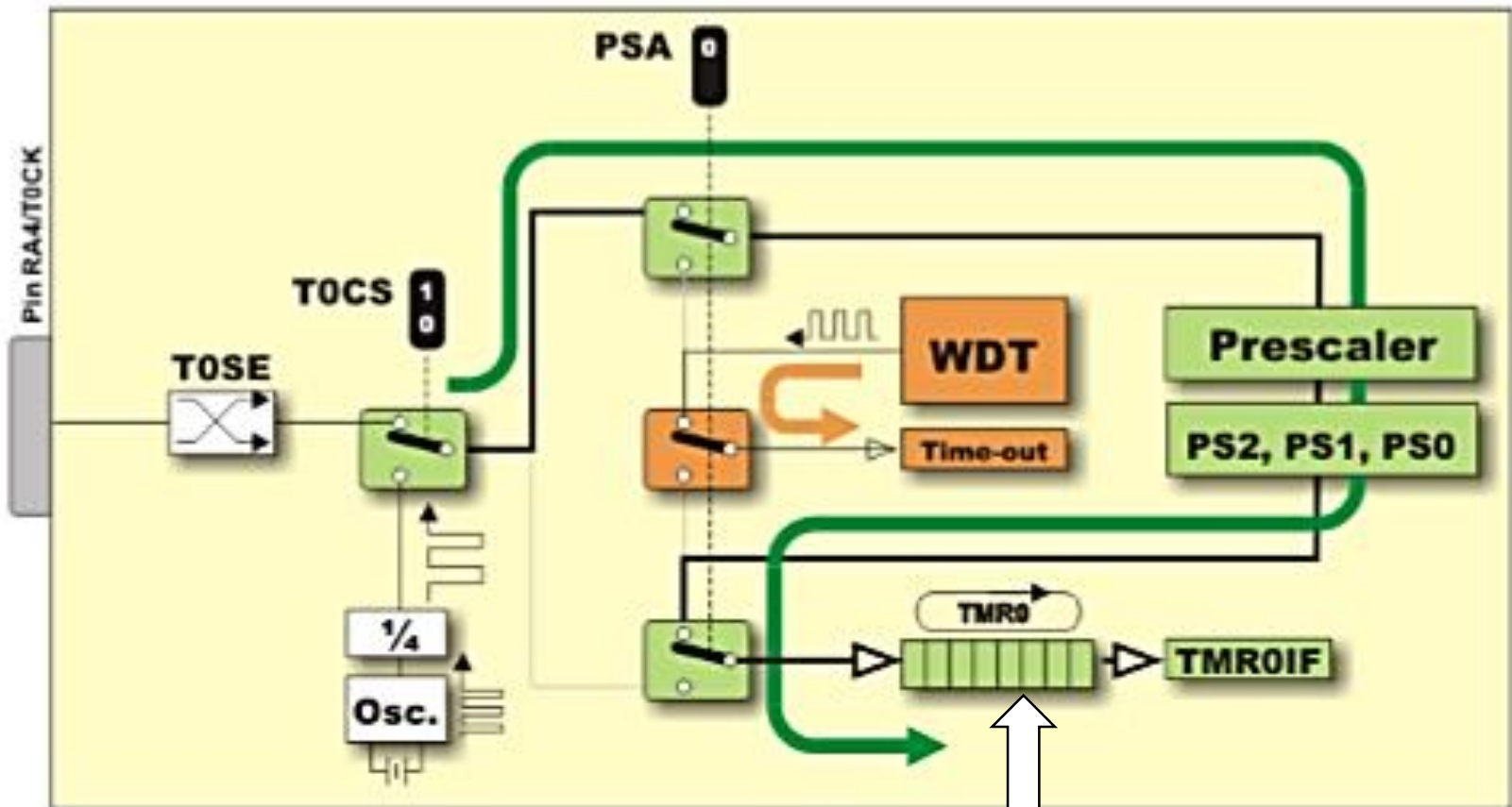
0-Không cho ngắt

TMR0IF: Bit báo cờ tràn của TMR0

1-Giá trị của TMR0 đã tràn (phải clear trước khi sử dụng Timer)

0-Giá trị của TMR0 chưa tràn

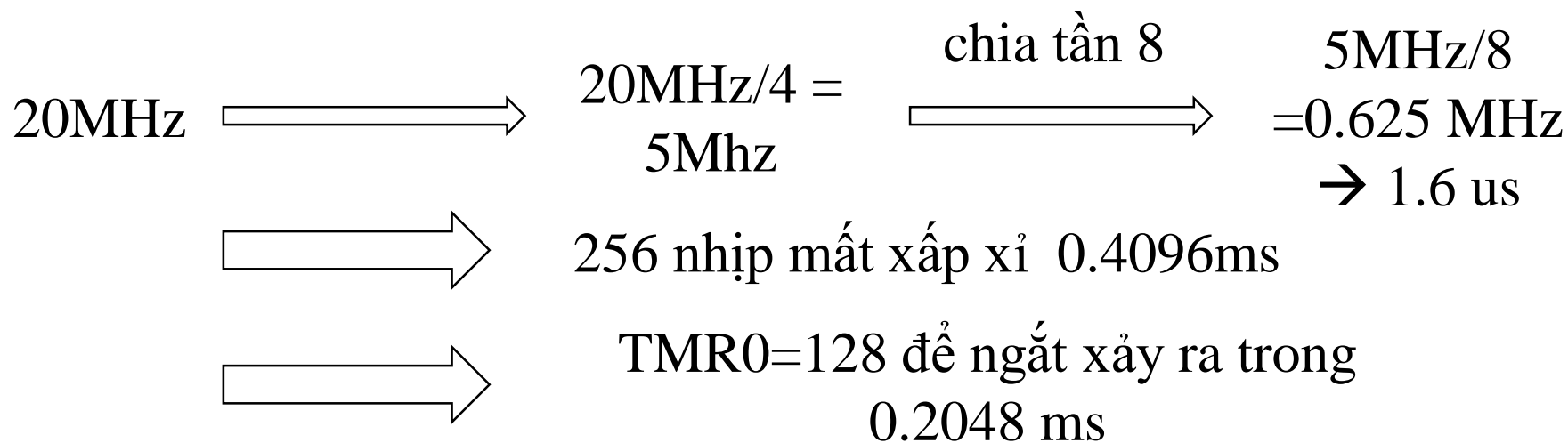
2. Khảo Sát Timer0



Thanh ghi lưu giá trị đếm

2. Khảo Sát Timer0 (Bộ chia tần)

- Vi điều khiển có tần số là 20MHz, nếu sử dụng bộ chia tần là 8 thì mất bao lâu để Timer0 có thể ngắt được 1 lần và set TMR0 bằng 128 thì ngắt xảy ra trong ???

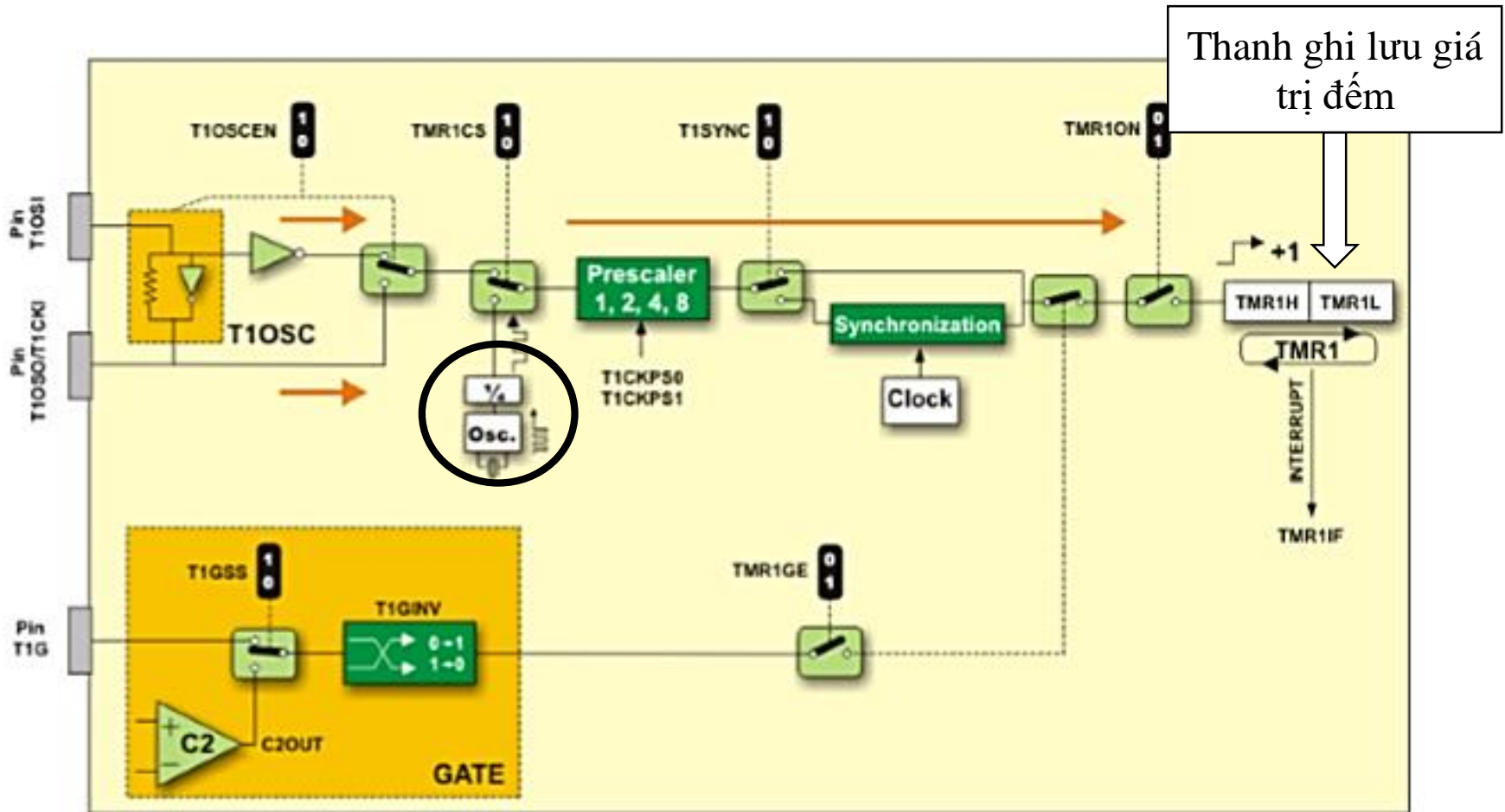


2.Khảo Sát Timer0 (Bộ chia tần)

B1) Vi điều khiển có tần số là 20MHz, nếu sử dụng bộ chia tần là 16 thì TIMER0 sẽ có thể mất tối đa là bao nhiêu giây để làm đầy thanh ghi TMR0. (Ghi rõ các giá trị thanh ghi PS2-PS0)

B2) Vi điều khiển có tần số là 20MHz, nếu sử dụng bộ chia tần là 64 thì TIMER0 sẽ có thể mất tối đa là bao nhiêu giây để làm đầy thanh ghi TMR0. (Ghi rõ các giá trị thanh ghi PS2-PS0)

3. Khảo Sát Timer1



3. Khảo Sát Timer1



T1CKPS1:T1CKPS0: Chọn bộ chia tần số

| T1CKPS1:T1CKPS0 | Prescale |
|-----------------|----------|
| 11 | 1:8 |
| 10 | 1:4 |
| 01 | 1:2 |
| 00 | 1:1 |

3. Khảo Sát Timer1



T1OSCEN: Bit điều khiển cho phép bộ dao động của Timer 1

1- Cho phép bộ tạo dao động hoạt động

0- Không cho phép bộ tạo dao động hoạt động

T1SYNC: Bit điều khiển đồng bộ ngõ vào xung clock bên ngoài
(chỉ sử dụng khi TMR1CS=1)

1- Không sử dụng xung ngoài là xung đồng bộ

0- Sử dụng xung ngoài là xung đồng bộ

3. Khảo Sát Timer1



TMR1CS: Bit lựa chọn nguồn xung của Timer1

0- Chọn xung nội bên trong

1- Chọn xung ngoại bên ngoài ở chân RC0/T10S0/T1CKI

TMR1ON: Bit điều khiển Timer1

1- Cho phép sử dụng Timer1

0- Không cho phép sử dụng Timer1

3. Khảo Sát Timer1

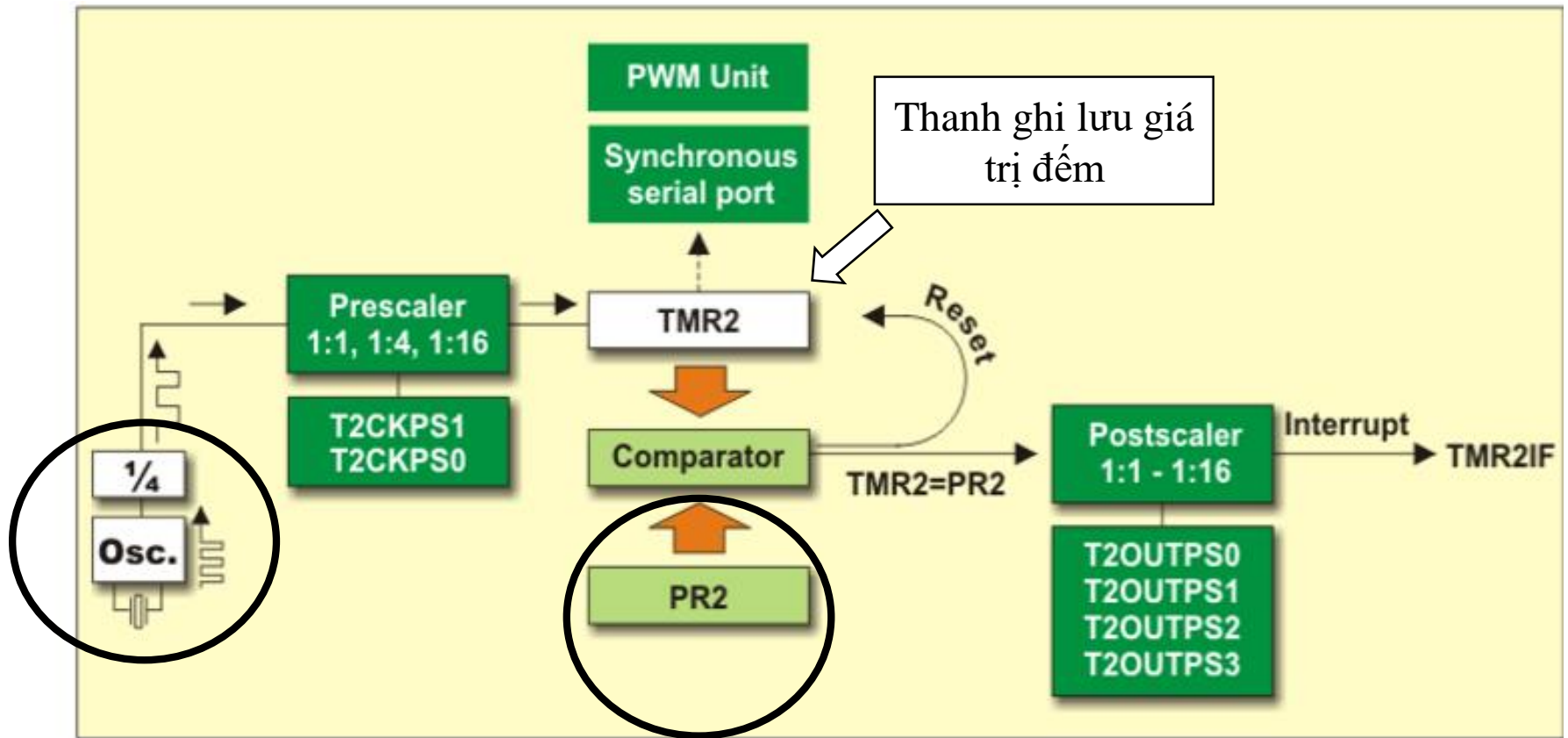
Timer1 ở chế độ đồng bộ (chỉ sử dụng khi TMR1CS=1)

- Chỉ được sử dụng khi /T1SYNC=0: đồng bộ xung ngoại với xung bên trong.
- Khi CPU ở chế độ ngủ thì Timer sẽ không hoạt động

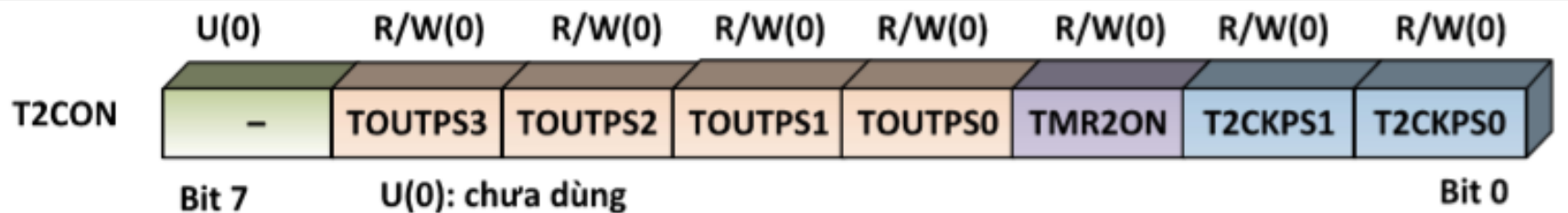
Timer1 ở chế độ bất đồng bộ (chỉ sử dụng khi TMR1CS=1)

- Chỉ được sử dụng khi /T1SYNC=1: không đồng bộ xung ngoại với xung bên trong.
- Khi CPU ở chế độ ngủ thì Timer vẫn sẽ hoạt động khi tràn bộ đếm đạt giá trị cực đại sẽ phát sinh ngắt và đánh thức CPU

4. Khảo Sát Timer2



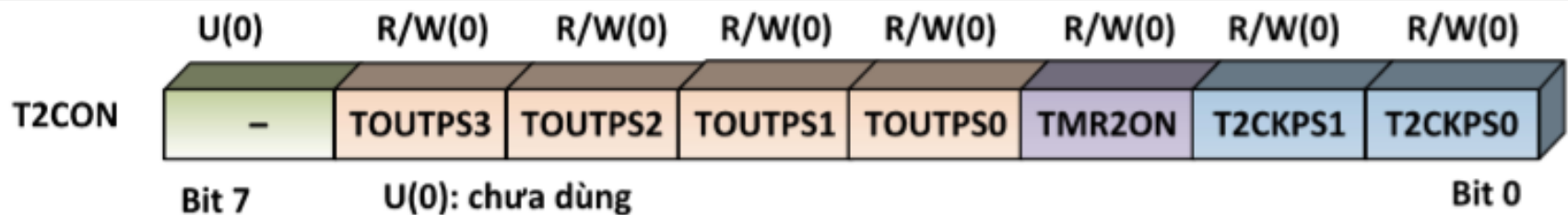
4. Khảo Sát Timer2



TOUTPS3:TOUTPS0: Bit chọn tỉ lệ chia sau:

| | | | |
|-----------|-----------|-----------|-----------|
| 0000=1:1 | 0001=1:2 | 0010=1:3 | 0011=1:4 |
| 0100=1:5 | 0101=1:6 | 0110=1:7 | 0111=1:8 |
| 1000=1:9 | 1001=1:10 | 1010=1:11 | 1011=1:12 |
| 1100=1:13 | 1101=1:14 | 1110=1:15 | 1111=1:16 |

4. Khảo Sát Timer2



TMR2ON: Bit điều khiển TIMER2

1- Cho phép Timer2 hoạt động

0- Không cho phép Timer2 hoạt động

T2CKPS1:T2CKPS0: Bit lựa chọn kiểu chia trước

| T2CKPS1:T2CKPS0 | Prescaler |
|-----------------|-----------|
| 00 | 1:1 |
| 01 | 1:4 |
| 1x | 1:16 |

5. Một số lệnh để sử dụng Timer trong PIC

| Cú Pháp | Chức năng |
|---------------------|---|
| Setup_Timer_X(Mode) | Định cấu hình cho Timer X (X=0,1,2) |
| Mode: | |
| Chọn loại xung | T0_INTERNAL T0_EXT_L_TO_H T0_EXT_H_TO_L |
| Bộ chia tần | T0_DIV_1 T0_DIV_2 T0_DIV_4 T0_DIV_8 T0_DIV_16 T0_DIV_32 T0_DIV_64 T0_DIV_128 T0_DIV_256 |

5. Một số lệnh để sử dụng Timer trong PIC

| Cú Pháp | Chức năng |
|--------------------------------|-------------------------------------|
| Set_timerx(Vaule) | Thiết lập giá trị bắt đầu cho Timer |
| Vaule=Get_timerX() | Đọc giá trị thanh ghi TMRX |
| Ngắt Timer: | |
| enable_interrupts(INT_TIMERX); | Kích hoạt ngắt của Timer |
| enable_interrupts(GLOBAL); | Cho phép ngắt toàn cục |

```
#include <>

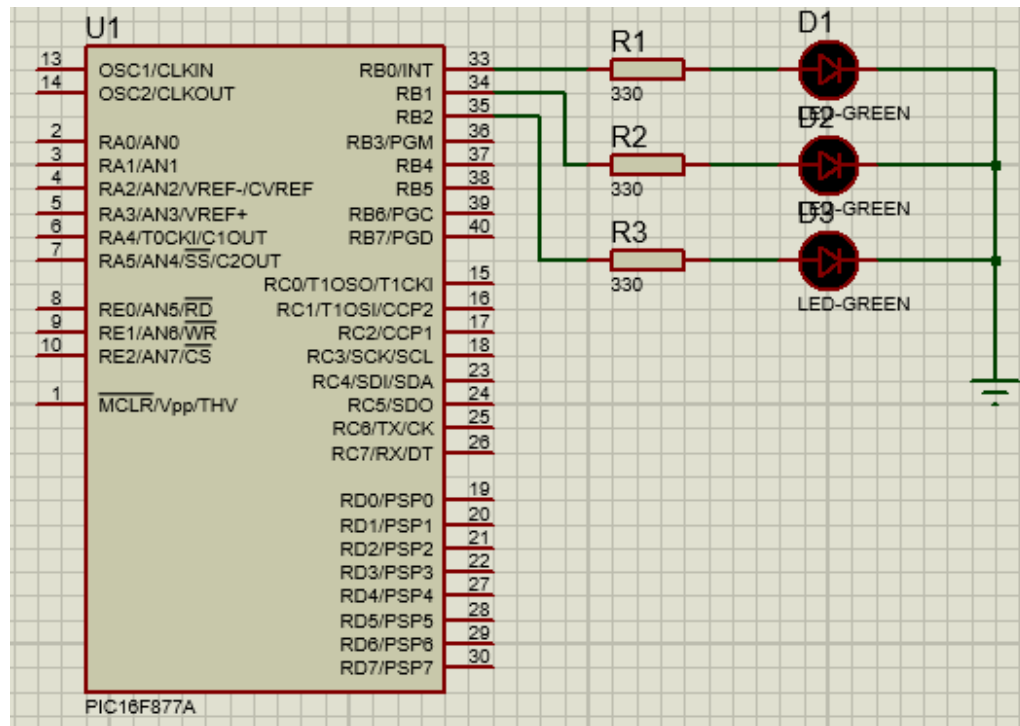
#INT_TIMER1
void Ngat_timer1()
{
    Code hoạt động
}
```

```
void main()
{
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_256);
    set_timer1(Vaule);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);
    while(TRUE)
    {

    }
}
```

3. Thực Hành

Bài 1: Dùng vi điều khiển PIC 16F877A điều khiển 3 led đơn sáng tắt với chu kì là 200ms (100ms sáng/100ms tắt) sử dụng Timer1



3.Thực Hành

```
#include <>
unsigned int8 X=0;

#INT_TIMER1
void Ngat_timer1()
{
    X=~X;
    output_b(X);
    set_timer1(???);
}
```

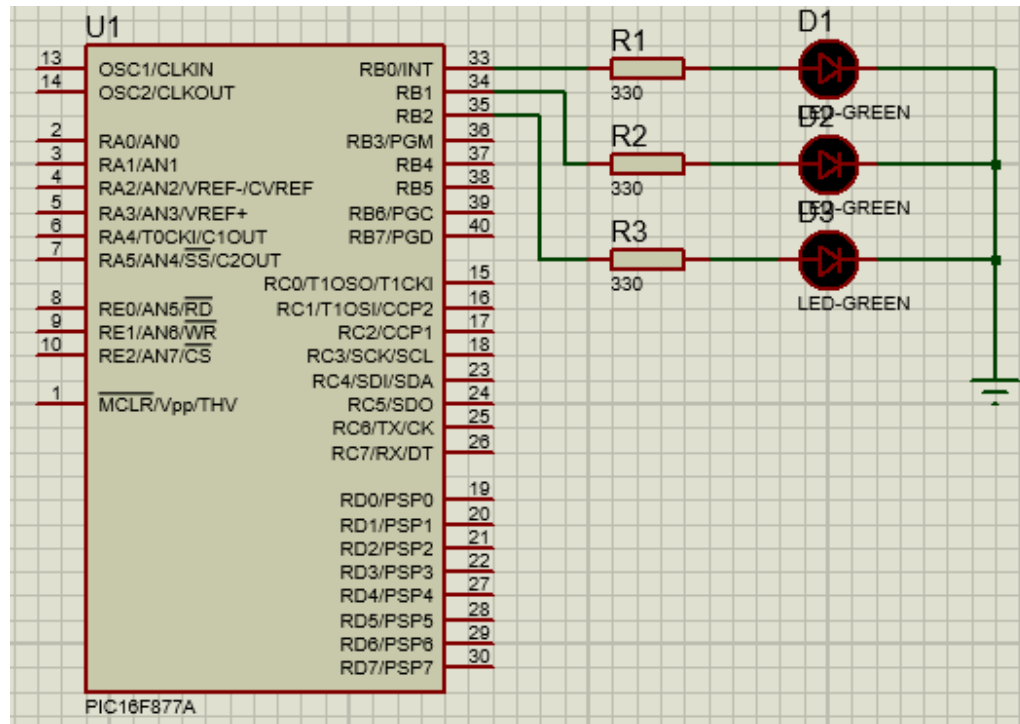
```
void main()
{
    set_tris_b(0x00);
    output_b(X);

    setup_timer_1(T1_INTERNAL|T1_DIV_B
Y_8);
    set_timer1(???);
    enable_interrupts(INT_TIMER1);
    enable_interrupts(GLOBAL);
    while(TRUE)
    {

    }
}
```


3. Thực Hành

Bài 2: Dùng vi điều khiển PIC 16F877A điều khiển 3 led đơn sáng tắt với chu kì là 2s (1s sáng/1s tắt) sử dụng Timer1



3.Thực Hành

```
#include <bai1.1.h>
```

```
unsigned int8 X=0,Count=1;
```

```
#INT_TIMER1
```

```
void Ngat_timer1()
```

```
{
```

```
    Count++;
```

```
    if (Count==10)
```

```
    {
```

```
        X=~X;
```

```
        output_b(X);
```

```
        Count=0;
```

```
    }
```

```
    set_timer1(3036);
```

```
}
```

```
void main()
```

```
{
```

```
    set_tris_b(0x00);
```

```
    output_b(X);
```

```
    setup_timer_1(T1_INTERNAL|T1_DIV_BY_8);
```

```
    set_timer1(3036);
```

```
    enable_interrupts(INT_TIMER1);
```

```
    enable_interrupts(GLOBAL);
```

```
    while(TRUE)
```

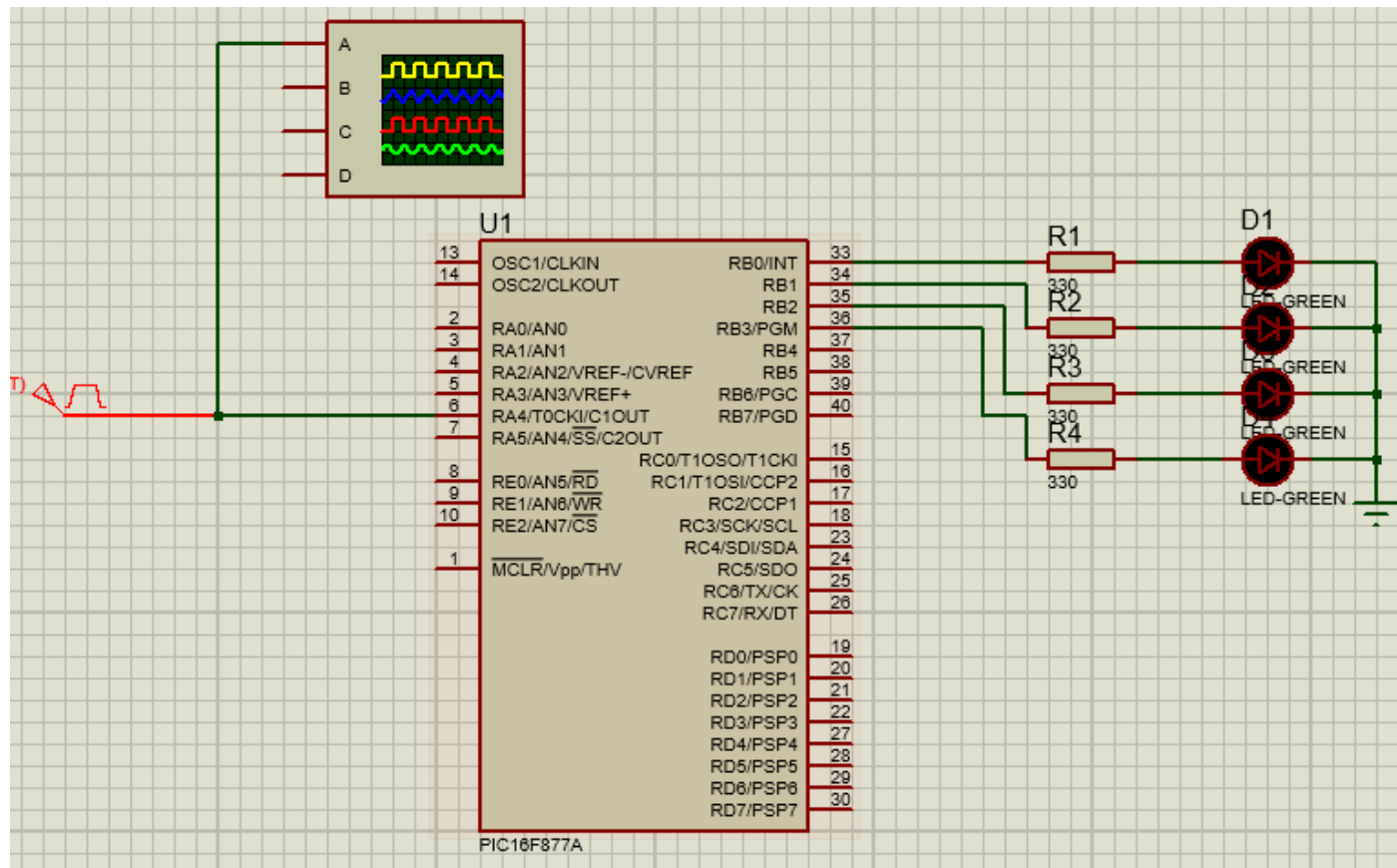
```
    {
```

```
    }
```

```
}
```

3. Thực Hành

Bài 3: Dùng vi điều khiển PIC 16F877A điều khiển 3 led đơn hiển thị giá trị đếm của Timer0 (chỉ đếm đến 10 rồi reset)



3.Thực Hành

```
#include <>
Khai báo biến Count
void main()
{
    Cài đặt port B là output
    Khai báo Timer0 với chế độ sử dụng xung ngoài và bộ chia tần là 1
    Đặt giá trị thanh ghi TMR0=0
    while(TRUE)
    {
        //TODO: User Code
        If (count==???)
        {
            Đặt lại thanh ghi TMR0
        }
        Đọc giá trị của thanh ghi TMR0
        Bật tắt led theo giá trị Count
    }
}
```