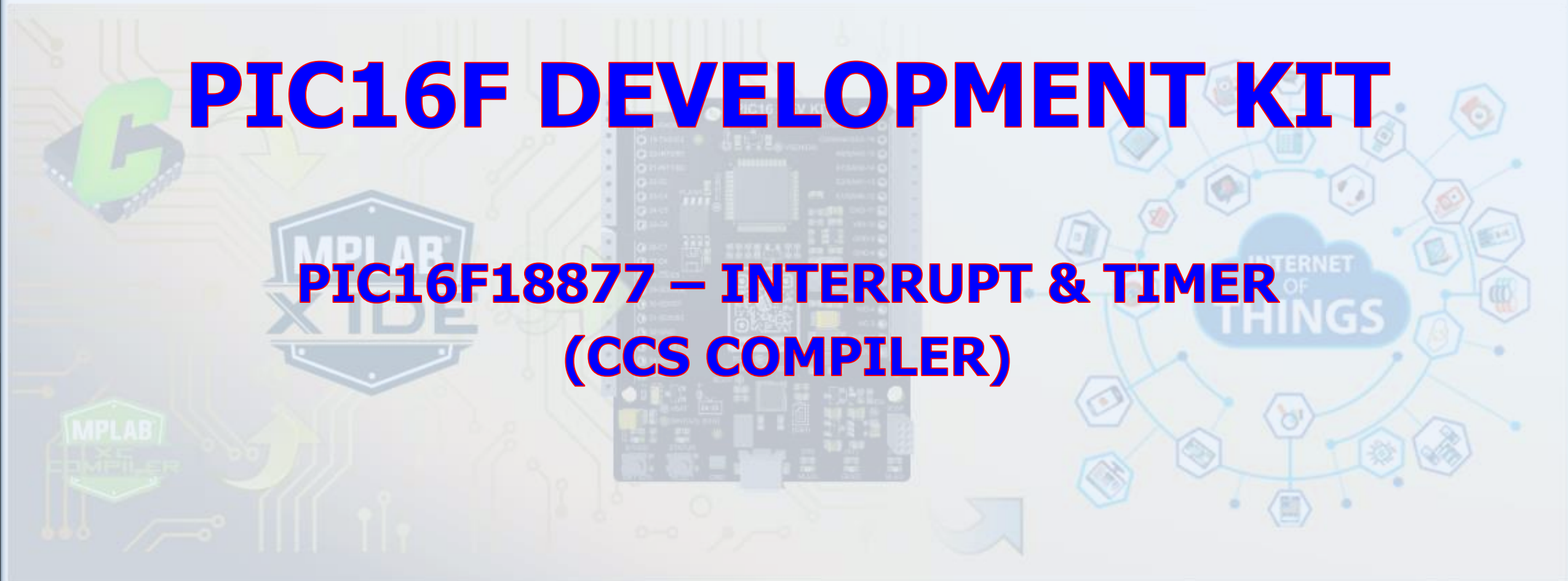


# **PIC16F DEVELOPMENT KIT**

**PIC16F18877 – INTERRUPT & TIMER  
(CCS COMPILER)**





# **NỘI DUNG**

**INTERRUPT**

**TIMER**



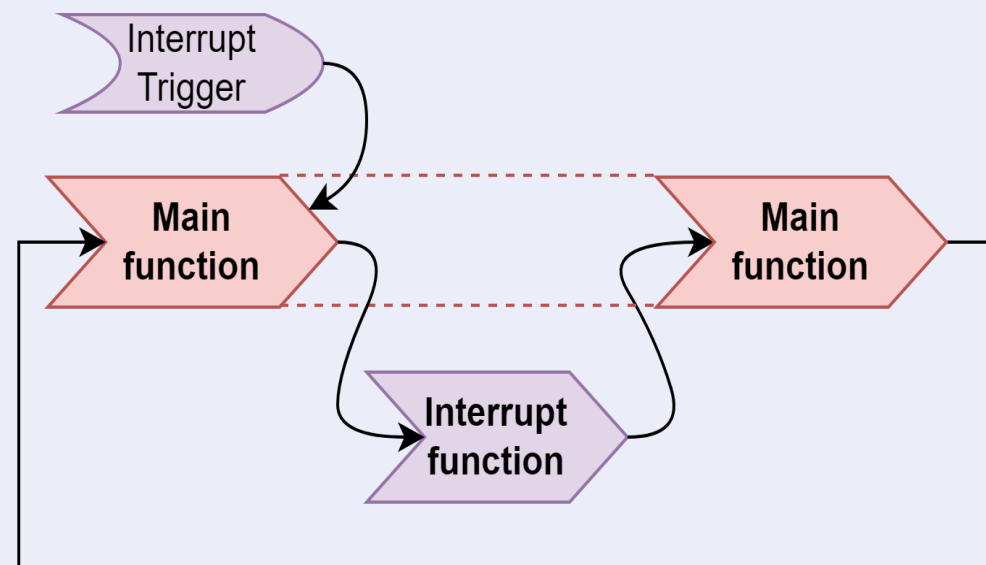
# INTERRUPT

## 1. TỔNG QUAN

INTERRUPT

TIMER

- ❖ Ngắt thực thi một chương trình ưu tiên hơn chương trình chính (main function).
- ❖ Ngắt xử lý một sự kiện không xác định thời điểm xảy ra.
- ❖ Ngắt xử lý sự kiện theo chu kỳ.





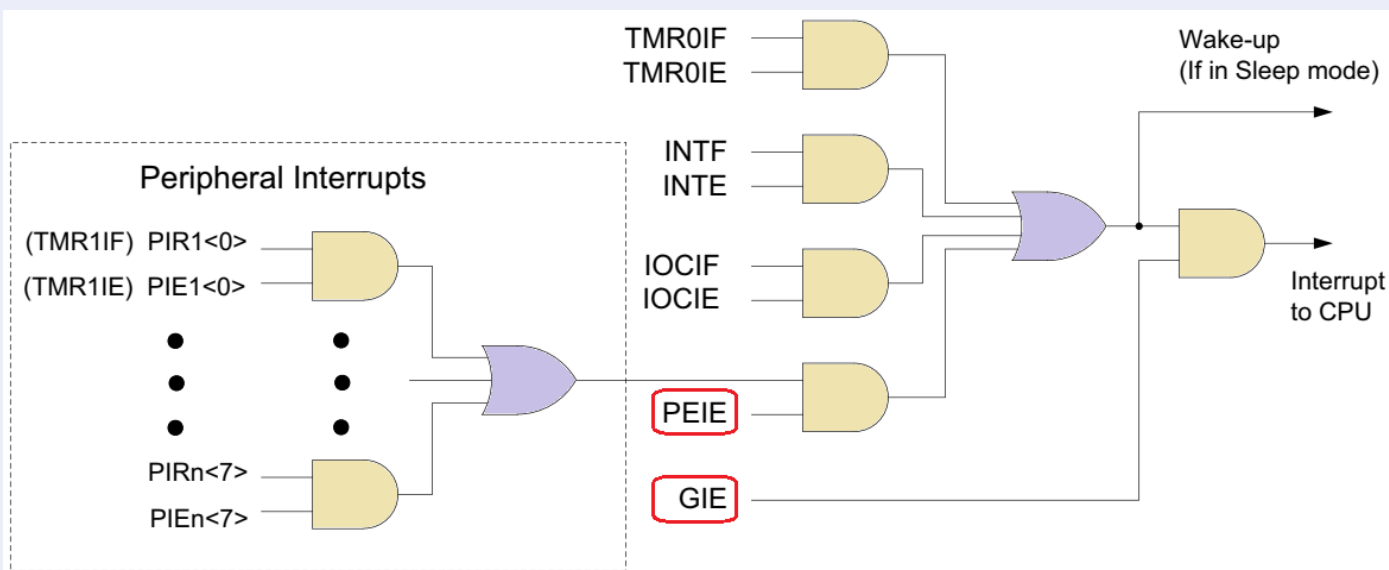
# INTERRUPT

## 2. CÁC NGUỒN NGẮT

INTERRUPT

TIMER

- ❖ PIC16F18877 có 1 vector ngắt.
  - ❖ Có 2 nhóm ngắt:
    - Ngắt ngoại vi (peripheral interrupts), điều khiển bởi bit PEIE.
    - Ngắt ngoài ngoại vi, điều khiển bởi bit GIE.
- Bit này cũng đồng thời điều khiển tất cả các nguồn ngắt khác.





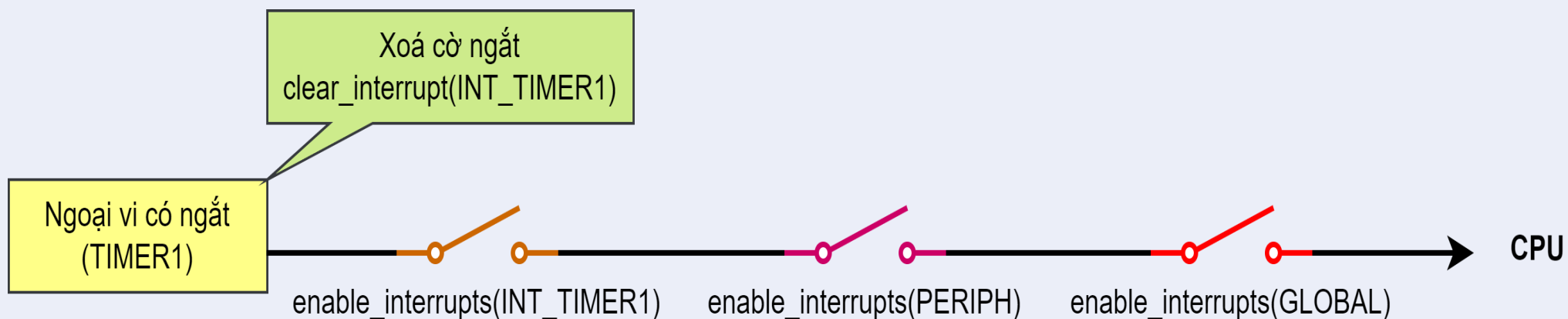
# INTERRUPT

## 3. NGẮT TRONG CCS

INTERRUPT

TIMER

- ❖ Bật ngắt *enable\_interrupts(tên nguồn ngắt)*.
- ❖ Tắt ngắt *disable\_interrupts(tên nguồn ngắt)*.
- ❖ Xóa cờ ngắt *clear\_interrupt(tên nguồn ngắt)*.
- ❖ Cấu trúc khối điều khiển ngắt:





# INTERRUPT

## 3. NGẮT TRONG CCS

INTERRUPT

TIMER

❖ Các nguồn ngắt thông dụng:

GLOBAL	Ngắt toàn cục	INT_OSC_FAIL	Ngắt lỗi dao động
PERIPH	Ngắt ngoại vi	INT_CLCx	Ngắt cổng logic
INT_IOC_x	Ngắt thay đổi trạng thái IO	INT_CCPx	Ngắt CCP
INT_EXT	Ngắt ngoài	INT_TIMERx	Ngắt timer
INT_AD	Ngắt chuyển đổi ADC	INT_RDA	Ngắt UART
INT_SSPx	Ngắt SPI/I2C	INT_ZCD	Ngắt điểm 0



# INTERRUPT

## 4. CHƯƠNG TRÌNH NGẮT

INTERRUPT

TIMER

❖ Chương trình ngắt đặt bất kỳ vị trí nào trong file source.

❖ Cấu trúc:

#<TÊN NGUỒN NGẮT>

void FuncName(void)

```
{  
    clear_interrupt(Tên nguồn ngắt);  
    // Chương trình người dùng  
}
```

```
3      #INT_TIMER2  
4  
5      void TIMER2_isr(void)  
6      {  
7          clear_interrupt(INT_TIMER2);  
8          output_toggle(BLED);  
9      }
```



# INTERRUPT

## 4. CHƯƠNG TRÌNH NGẮT

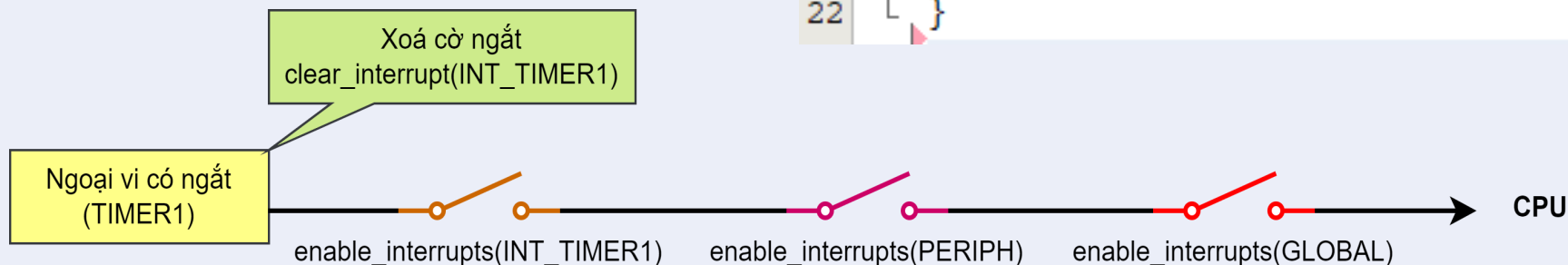
INTERRUPT

TIMER

### ❖ Cấu hình ngắt:

- Cấu hình module ngoại vi.
- Xóa cờ ngắt module ngoại vi.
- Cho phép ngắt module ngoại vi.
- Cho phép ngắt ngoại vi.
- Cho phép ngắt toàn cục.

```
11 void main(void)
12 {
13     setup_timer_2(T2_CLK_LFINTRC|T2_DIV_BY_2
14                 |T2_START_IMMEDIATELY, 154, 10);
15     clear_interrupt(INT_TIMER2);
16     enable_interrupts(INT_TIMER2);
17     enable_interrupts(PERIPH);
18     enable_interrupts(GLOBAL);
19
20     while(TRUE)
21         sleep();
22 }
```







# TIMER

## 1. TỔNG QUAN

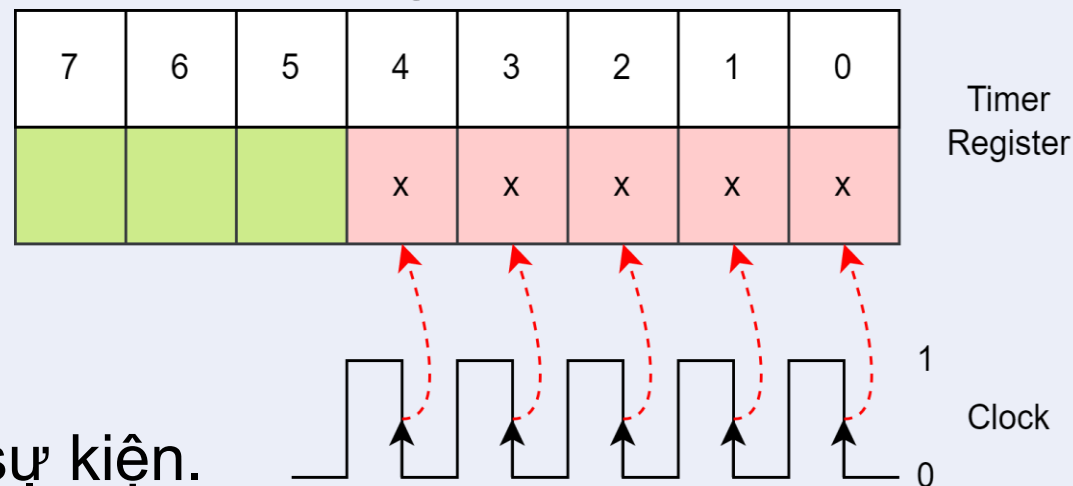
INTERRUPT

TIMER

- ❖ Timer/counter thực tế là một bộ đếm với giá trị đếm tăng 1 đơn vị tương ứng với mỗi xung clock cấp vào.
- ❖ Timer/counter bản chất là 1 module, chúng chỉ khác nhau về nguồn clock.

➤ Timer: Xung clock xác định được tần số, sử dụng để đếm thời gian.

➤ Counter: Xung clock không xác định tần số, sử dụng để đếm sự kiện.





# TIMER

## 1. TỔNG QUAN

INTERRUPT

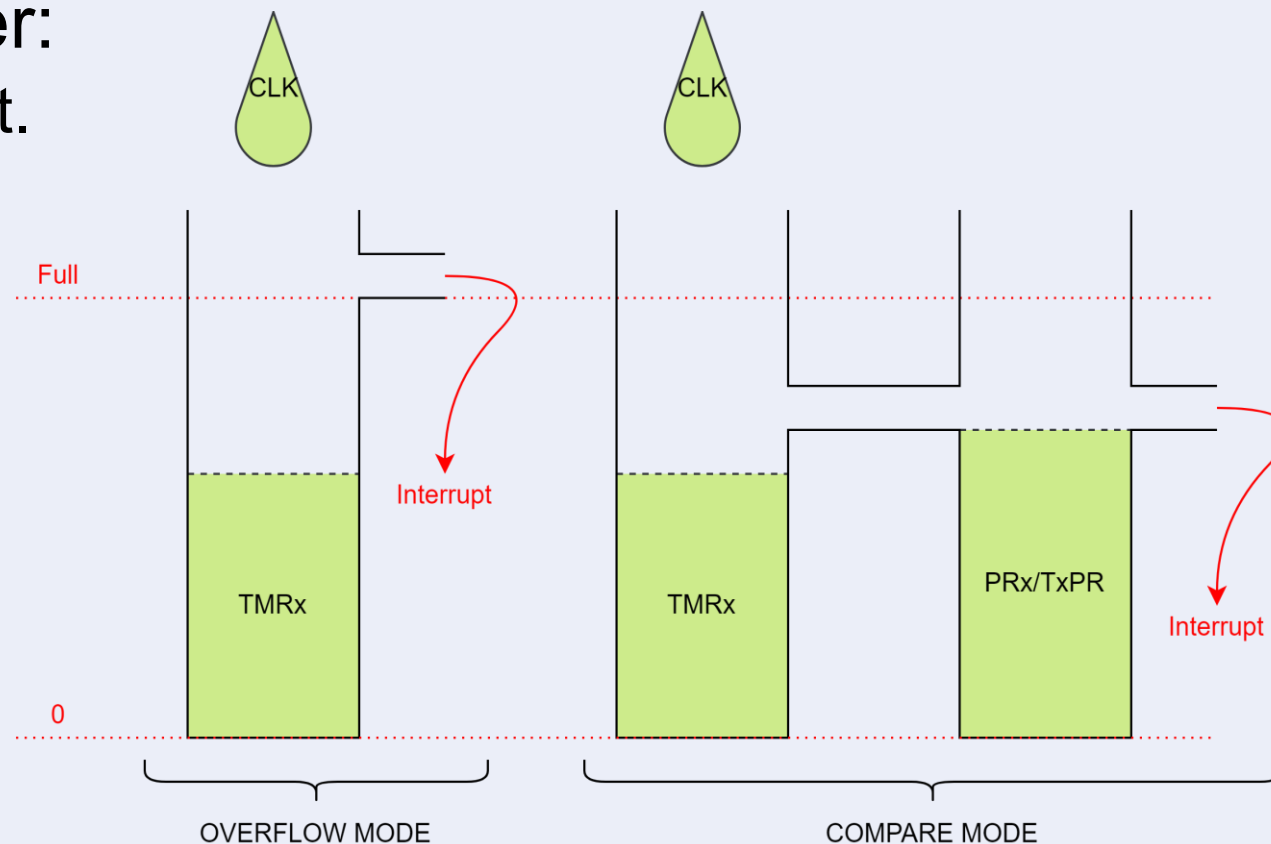
TIMER

### ❖ PIC16F18877 có 7 timer:

- Timer 0: 8-bit hoặc 16-bit.
- Timer 1/3/5: 16-bit.
- Timer 2/4/6: 8-bit.

### ❖ Hai chế độ hoạt động:

- Tràn (Overflow).
- So sánh (Compare).





# TIMER

## 2. CHẾ ĐỘ TRÀN

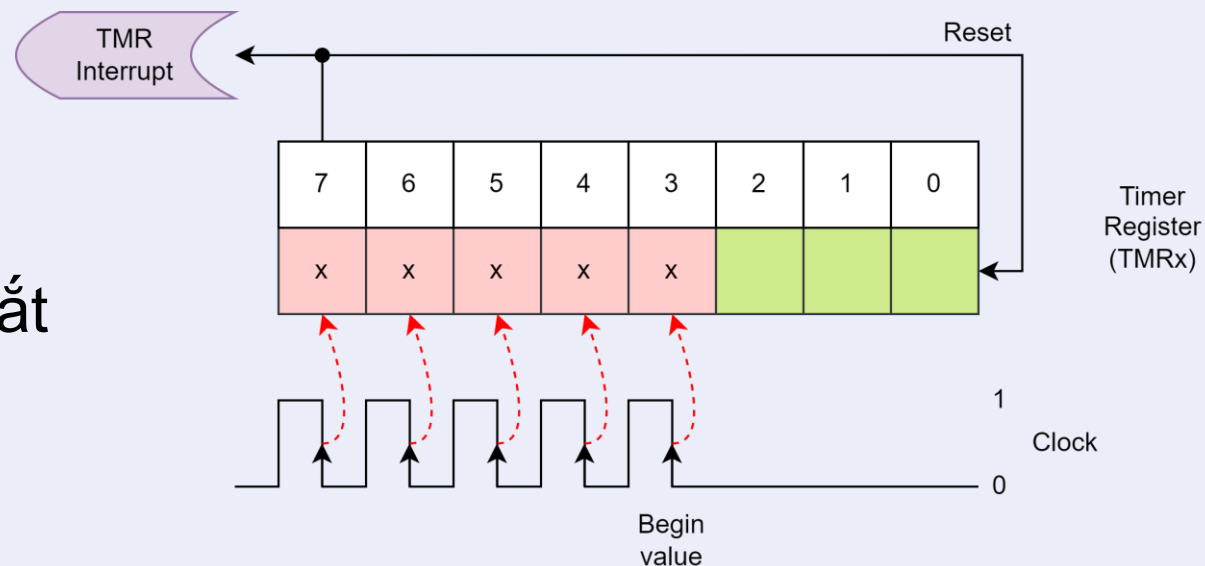
INTERRUPT

TIMER

❖ Áp dụng cho timer 16-bit (TMR0/1/3/5).

❖ Hoạt động:

- Ứng với mỗi xung cạnh lên, bộ đếm sẽ tăng 1 giá trị, khi đạt giá trị tối đa sẽ sinh ngắt và reset timer về 0.
- Giá trị bắt đầu là tùy chọn.





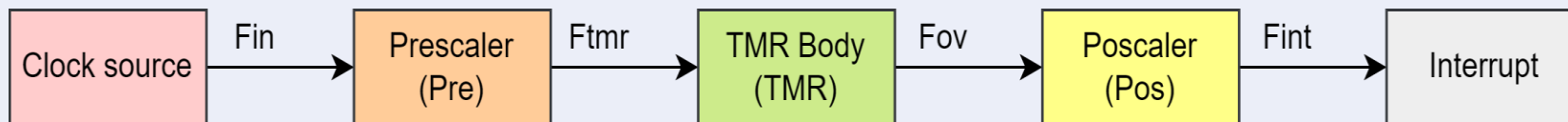
# TIMER

## 2. CHẾ ĐỘ TRÀN

INTERRUPT

TIMER

❖ Tính toán:



Tần số xung cấp vào timer:  $F_{tmr} = \frac{F_{in}}{Pre}$

Khoảng định thời: từ  $\frac{1000}{F_{tmr}}$  đến  $\frac{1000}{F_{tmr}} \times 65536$  (ms)

Tần suất tràn:  $F_{ov} = \frac{F_{tmr}}{65536 - TMR} = \frac{F_{in}}{Pre \times (65536 - TMR)}$

Tần suất ngắt (TMR=0, Pos>1):  $F_{int} = \frac{F_{ov}}{Pos} = \frac{F_{in}}{Pre \times Pos \times 65536}$

Tần suất ngắt (Pos=1):  $F_{int} = F_{ov} = \frac{F_{in}}{Pre \times (65536 - TMR)} \Rightarrow TMR = 65536 - \frac{F_{in}}{F_{int} \times Pre}$



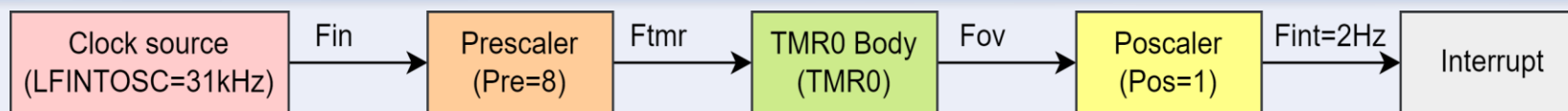
# TIMER

## 2. CHẾ ĐỘ TRÀN

INTERRUPT

TIMER

❖ Ví dụ:



Tần số xung cấp vào timer:  $F_{tmr} = \frac{F_{in}}{Pre} = \frac{31000}{8} = 3875Hz$

Khoảng định thời: từ  $\frac{1000}{3875} = 0.258ms$  đến  $\frac{1000}{3875} \times 65536 = 16912.516ms$

Giả sử ta muốn ngắt sau mỗi 500ms ( $Pos=1 \Rightarrow Fov=Fint=2Hz$ ):

Giá trị TMR theo Fint:  $TMR = 65536 - \frac{31000}{2 \times 8} = 63599.$

MCU 8-bit nên thanh ghi 16-bit sẽ tách thành 2 thanh ghi 8bit:

8-bit thấp:  $TMRL = TMR \% 256 = 111$

8-bit cao:  $TMRH = \frac{TMR}{256} = 248$



# TIMER

## 2. CHẾ ĐỘ TRÀN

INTERRUPT

TIMER

### ❖ Cấu hình TMR trên CCS:

- Cấu hình Fin, Pre, Pos, ...
- Cấu hình giá trị timer.
- Cấu hình interrupt.

```
13      setup_timer_0(T0_LFINTOSC|T0_INPUT_NOT_SYNCRONIZED|T0_DIV_8|T0_16_BIT);
14      set_timer0(63599);
15      clear_interrupt(INT_TIMER0);
16      enable_interrupts(INT_TIMER0);
17      enable_interrupts(GLOBAL);
```



# TIMER

## 2. CHẾ ĐỘ TRÀN

INTERRUPT

TIMER

### ❖ Ngắt TMR trên CCS:

- Xoá cờ ngắt.
- Đặt lại giá trị bắt đầu của TMR (nếu có).
- Thực hiện chương trình người dùng.'

```
3  #INT_TIMER0
4  void TIMER0_isr(void)
5  {
6      clear_interrupt(INT_TIMER0);
7      set_timer0(63599);
8      output_toggle(BLED);
9  }
```

→ Lưu ý: Tránh sử dụng biến hoặc function trùng với chương trình chính để tránh sự xung đột.



# TIMER

## 3. CHẾ ĐỘ SO SÁNH

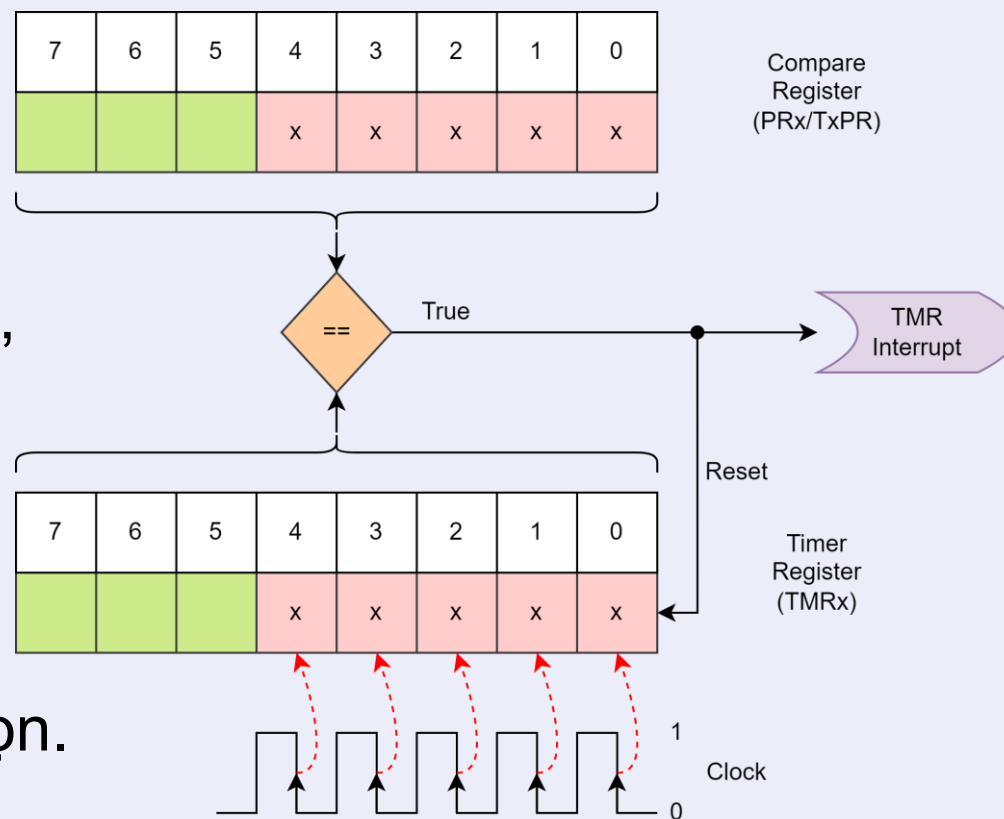
INTERRUPT

TIMER

❖ Áp dụng cho timer 8-bit (TMR0/2/4/6).

❖ Hoạt động:

- Ứng với mỗi xung (cạnh tùy chọn), bộ đếm sẽ tăng 1 giá trị, khi đạt giá trị của thanh ghi so sánh sẽ sinh ngắt và reset timer về 0.
- Giá trị TMR bắt đầu là 0.
- Giá trị thanh ghi so sánh là tùy chọn.







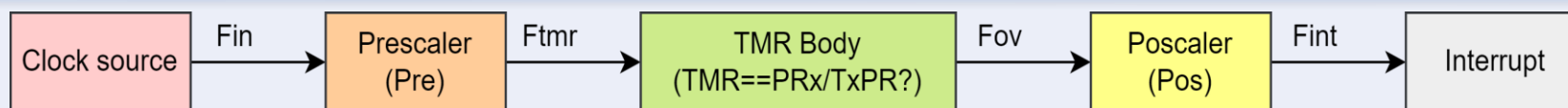
# TIMER

## 3. CHẾ ĐỘ SO SÁNH

INTERRUPT

TIMER

❖ Tính toán:



Tần số xung cấp vào timer:  $F_{tmr} = \frac{F_{in}}{Pre}$

Khoảng định thời: từ  $\frac{1000}{F_{tmr}}$  đến  $\frac{1000}{F_{tmr}} \times 256$  (ms)

Tần suất tràn:  $F_{ov} = \frac{F_{tmr}}{TMR+1} = \frac{F_{in}}{Pre \times (TxPR+1)} \Rightarrow TxPR = \frac{F_{in}}{F_{ov} \times Pre} - 1$

Tần suất ngắt:  $F_{int} = \frac{F_{ov}}{Pos} = \frac{F_{in}}{Pre \times Pos \times (TxPR+1)} \Rightarrow TxPR = \frac{F_{in}}{F_{int} \times Pre \times Pos} - 1$



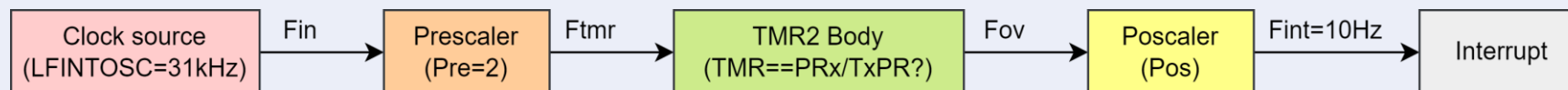
# TIMER

## 3. CHẾ ĐỘ SO SÁNH

INTERRUPT

TIMER

❖ Ví dụ:



Tần số xung cấp vào timer:  $F_{tmr} = \frac{F_{in}}{Pre} = \frac{31000}{2} = 15500Hz$

Khoảng định thời: từ  $\frac{1000}{15500} \approx 0.065ms$  đến  $\frac{1000}{15500} \times 256 \approx 16.516ms$

Giả sử ta muốn ngắt sau mỗi 100ms (>16.516ms), ta có  $F_{int} = 10Hz$ , chọn  $Pos = 10$

$$\Rightarrow F_{ov} = F_{int} \times Pos = 10 \times 10 = 100Hz$$

Giá trị  $TxPR$  theo  $F_{ov}$ :  $TxPR = \frac{31000}{100 \times 2} - 1 = 154.$



# TIMER

## 3. CHẾ ĐỘ SO SÁNH

INTERRUPT

TIMER

### ❖ Cấu hình TMR trên CCS:

- Cấu hình Fin, Pre, Pos, ...
- Cấu hình giá trị timer.
- Khởi động timer.
- Cấu hình interrupt.

```
13  setup_timer_2(T2_CLK_LFINTRC|T2_DIV_BY_2
14          |T2_START_IMMEDIATELY, 154, 10);
15  clear_interrupt(INT_TIMER2);
16  enable_interrupts(INT_TIMER2);
17  enable_interrupts(PERIPH);
18  enable_interrupts(GLOBAL);
```



# TIMER

## 3. CHẾ ĐỘ SO SÁNH

INTERRUPT

TIMER

### ❖ Ngắt TMR trên CCS:

- Xoá cờ ngắt.
- Đặt lại giá trị bắt đầu của TMR (nếu có).
- Thực hiện chương trình người dùng.

→ Lưu ý: Tránh sử dụng biến hoặc function trùng với chương trình chính để tránh sự xung đột.

```
3  #INT_TIMER2
4
5  void TIMER2_isr(void)
6  {
7      clear_interrupt(INT_TIMER2);
8      output_toggle(BLED);
9  }
```