

**Đại Học Công Nghiệp Thành Phố Hồ Chí Minh**

\*\*\*\*\*

# **Bài 3 : Hệ thống điều khiển Interrupt**

**Bộ môn: Điện Tử Máy Tính**

**GV: Lê Lý Quyên Quyên**

**Mail: [lelyquyenquyen@iuh.edu.vn](mailto:lelyquyenquyen@iuh.edu.vn)**

# MODUL NGẮT (INTERRUPT)

## Nội dung bao gồm:

- Tổng quan về ngắt
- Các nguồn ngắt của PIC16F887
- Cho phép / cấm các nguồn ngắt
- Các thanh ghi của modul ngắt
- Cấu trúc một chương trình có sử dụng ngắt
- Độ trễ của ngắt
- Sao lưu / phục hồi dữ liệu khi ngắt
- Ưu tiên ngắt
- Ví dụ minh họa và bài tập ứng dụng.

# TỔNG QUAN

- Trong thực tế chúng ta thường mong muốn bộ xử lý sẽ thực hiện một công việc nào đó khi có một sự kiện xảy ra
- Có 2 phương pháp để kiểm tra sự xuất hiện của một sự kiện và thực thi đáp ứng
  - **Hỏi vòng (Polling):** Phải kiểm tra liên tục tại các thời điểm khác nhau của chương trình chính để phát hiện sự kiện xảy ra và thực thi chương trình phục vụ sự kiện
  - **Ngắt (Interrupt):** Tự động tạm dừng chương trình chính và bắt đầu thực thi chương trình phục vụ sự kiện ngay khi sự kiện xảy ra.

# TỔNG QUAN

## ➤ Ngắt (Interrupt):

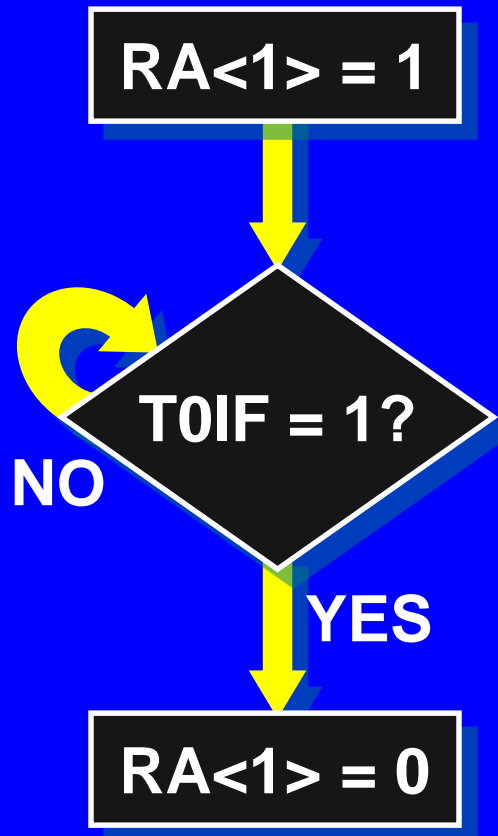
- Bộ xử lý tạm dừng công việc đang thực thi (A) để chuyển sang thực thi một công việc khác (B) nếu một sự kiện cụ thể xảy ra.
- Sau khi thực thi xong công việc đó (B) sẽ quay trở về tiếp tục thực thi công việc (A) đang bị tạm dừng

## ➤ Tín hiệu thông báo cho bộ xử lý về một sự kiện được gọi là tín hiệu ngắt (nguồn ngắt)

## ➤ Chương trình được thực thi để đáp ứng cho sự kiện được gọi là chương trình con phục vụ ngắt (ISR: Interrupt Service Routine).

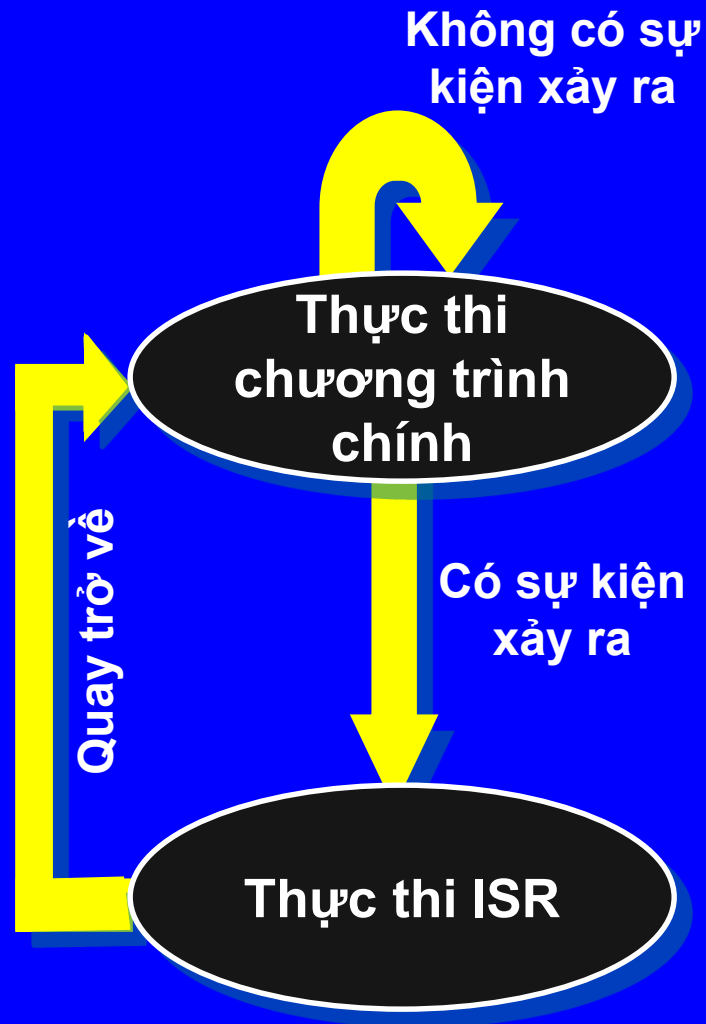
# TỔNG QUAN

## ➤ Phương pháp thăm dò (Polling):



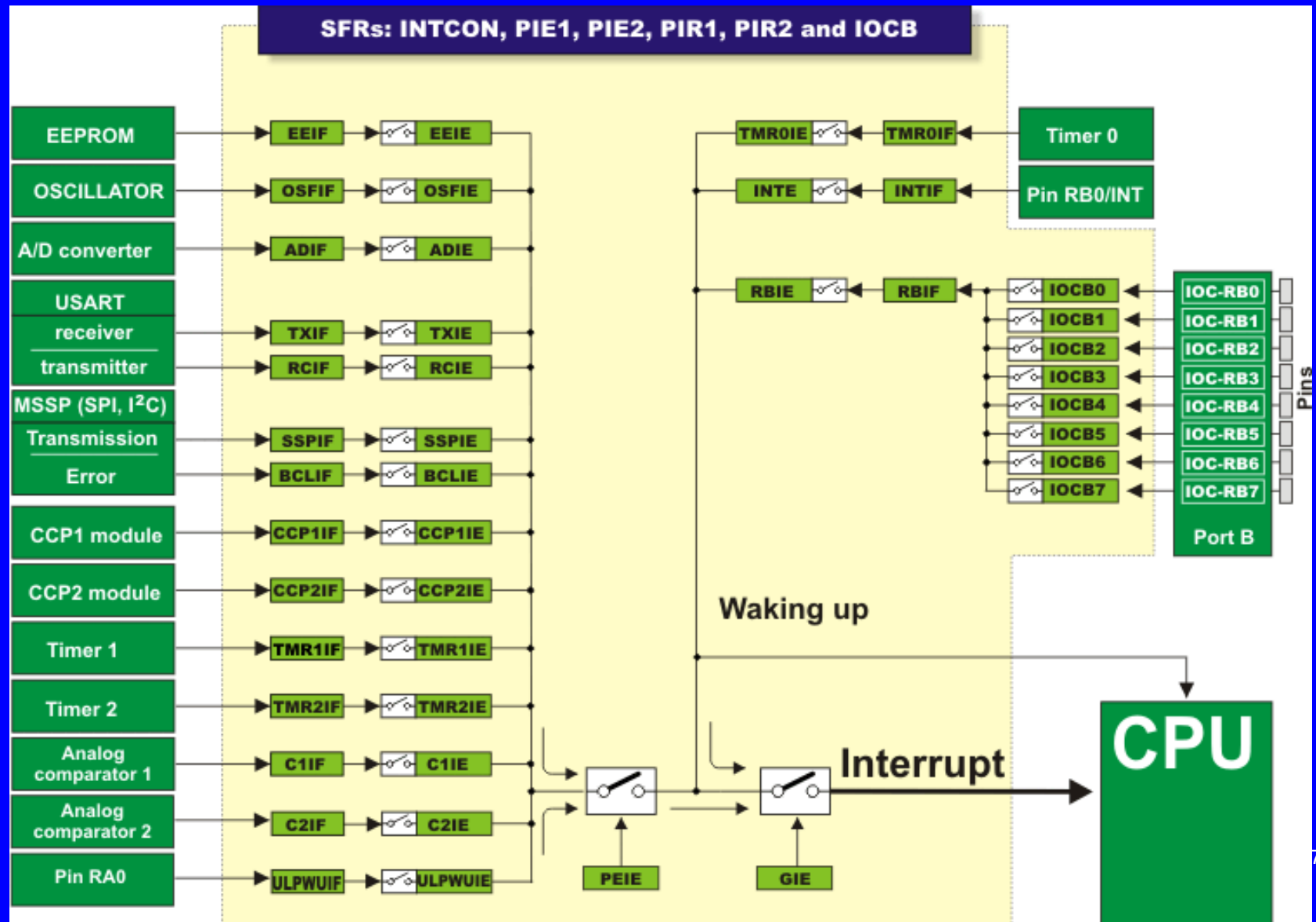
# TỔNG QUAN

## ➤ Phương pháp ngắt (Interrupt):



# CÁC NGUỒN NGẮT CỦA PIC16F887

➤ Có 15 nguồn ngắt.



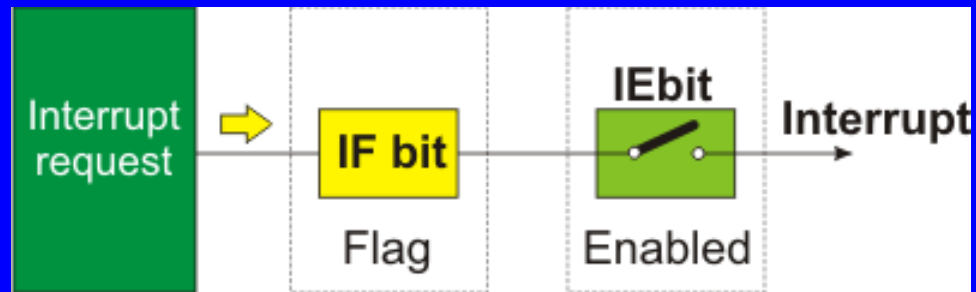
# CHO PHÉP / CẤM CÁC NGẮT

- Mặc định thì tất cả các ngắt đều bị cấm
- Mỗi ngắt được cho phép hoặc cấm bởi người sử dụng thông qua các bit cho phép (**IE: Interrupt Enable**) tương ứng
  - IE = 0: Cấm
  - IE = 1: Cho phép
- PIC16F887 cho phép các ngắt thông qua các bit trong một số thanh ghi sau:
  - Interrupt Control Register (**INTCON**)
  - Peripheral Interrupt Enable Register 1 (**PIE1**)
  - Peripheral Interrupt Enable Register 2 (**PIE2**).

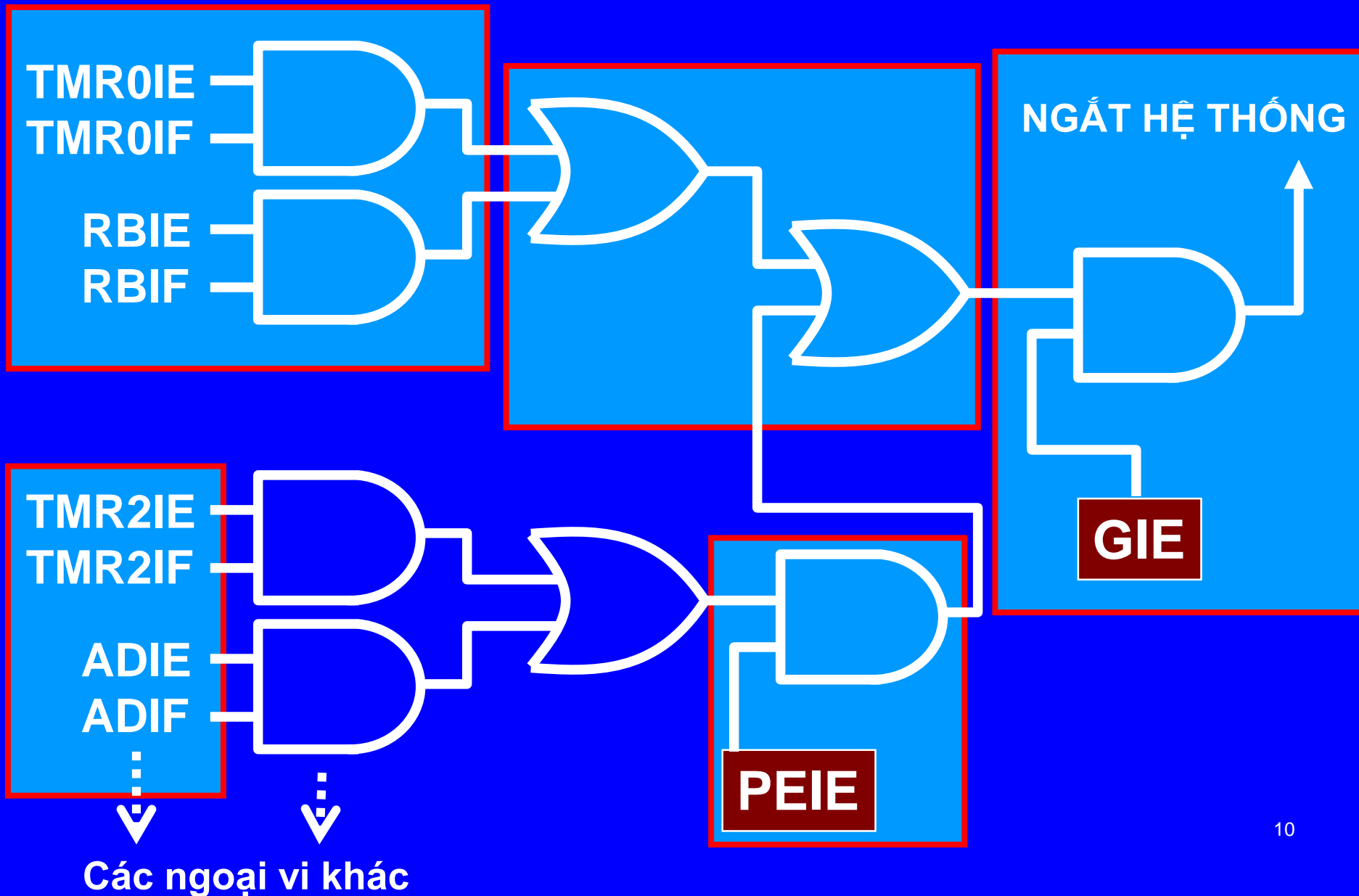


# CHO PHÉP / CẤM CÁC NGẮT

- Mỗi ngắt được kết hợp với 1 bit khác, dùng để chỉ thị rằng “sự kiện ngắt” đã xảy ra. Bit này được gọi là cờ ngắt (**IF: Interrupt Flag**)
  - IF = 0: Không có sự kiện
  - IF = 1: Có sự kiện
- Cờ ngắt IF sẽ có mức cao (IF = 1) ngay khi “sự kiện ngắt” tương ứng xảy ra, bất chấp ngắt đó có được cho phép hay không.
- Cờ ngắt IF phải được xóa (IF = 0) bởi người sử dụng trước khi thoát khỏi ISR của nó.



# CHO PHÉP / CẤM CÁC NGẮT



# CÁC THANH GHI CỦA MODUL NGẮT

## ➤ Thanh ghi **INTCON**

- Thanh ghi cho phép ngắt và yêu cầu ngắt (cờ ngắt) nội

## ➤ Thanh ghi **PIE1**

- Thanh ghi cho phép ngắt ngoại vi 1

## ➤ Thanh ghi **PIR1**

- Thanh ghi yêu cầu ngắt (cờ ngắt) ngoại vi 1

## ➤ Thanh ghi **PIE2**

- Thanh ghi cho phép ngắt ngoại vi 2

## ➤ Thanh ghi **PIR2**

- Thanh ghi yêu cầu ngắt (cờ ngắt) ngoại vi 2.

# NGẮT NỘI (CORE INTERRUPT)

## CÁC BIT CHO PHÉP

**GIE:** Cho phép ngắt toàn cục

**\* Phải được đặt để sử dụng BẤT KỲ ngắt nào của PIC**

**PEIE:** Cho phép ngắt ngoại vi

**\* Phải được đặt để sử dụng các ngắt ngoại vi**

**T0IE:** Cho phép ngắt Timer0 tràn

**INTE:** Cho phép ngắt ngoài INT

**RBIE:** Cho phép ngắt Port B thay đổi



**Các cờ ngắt vẫn sẽ  
được đặt ngay cả khi  
các ngắt không được  
cho phép**

## CÁC BIT CỜ

**T0IF:** Cờ ngắt Timer0 tràn

**INTF:** Cờ ngắt ngoài INT

**RBIF:** Cờ ngắt Port B thay đổi

# NGẮT NGOẠI VI (PERIPHERAL INTERRUPT)

## Thanh ghi PIE1 (Cho phép ngắt)

	ADIE	RCIE	TXIE	SSPIE	CCP1IE	TMR2IE	TMR1IE
--	------	------	------	-------	--------	--------	--------

## Thanh ghi PIR1 (Cờ ngắt)

	ADIF	RCIF	TXIF	SSPIF	CCP1IF	TMR2IF	TMR1IF
--	------	------	------	-------	--------	--------	--------

Cho phép	Cờ	Điều kiện
ADIE	ADIF	Hoàn thành chuyển đổi ADC
RCIE	RCIF	Đầy bộ đệm nhận EUSART
TXIE	TXIF	Đầy bộ đệm truyền EUSART
SSPIE	SSPIF	Ngắt I <sup>2</sup> C hoặc SPI
CCP1IE	CCP1IF	Hoàn thành so sánh hoặc bắt giữ cho Timer1
TMR2IE	TMR2IF	Giá trị thanh ghi Timer2 bằng giá trị thanh ghi PR2
TMR1IE	TMR1IF	Thanh ghi Timer1 bị tràn

**Các cờ ngắt vẫn sẽ được đặt ngay cả khi các ngắt không được cho phép**

# NGẮT NGOẠI VI (PERIPHERAL INTERRUPT)

## PIE2 Register (Interrupt Enables)

OSCFIE	C2IE	C1IE	EEIE	BCLIE	ULPWUIE		CCP2IE
--------	------	------	------	-------	---------	--	--------

## PIR2 Register (Interrupt Flags)

OSCFIF	C2IF	C1IF	EEIF	BCLIF	ULPWUIF		CCP2IF
--------	------	------	------	-------	---------	--	--------

Cho phép	Cờ	Điều kiện
OSCFIE	OSCFIF	Bộ dao động hệ thống bị lỗi
C2IE	C2IF	Ngõ ra bộ so sánh 2 bị thay đổi
C1IE	C1IF	Ngõ ra bộ so sánh 1 bị thay đổi
EEIE	EEIF	Thao tác ghi EEPROM hoàn tất
BCLIE	BCLIF	Xung đột bus xuất hiện ở chế độ MSSP I <sup>2</sup> C
ULPWUIE	ULPWUIF	Điều kiện “đánh thức” xảy ra
CCP2IE	CCP2IF	Hoàn thành so sánh hoặc bắt giữ cho Timer1

**Các cờ ngắt vẫn sẽ được đặt ngay cả khi các ngắt không được cho phép**

# ĐỘ TRỄ CỦA NGẮT

- **Độ trễ của ngắt (Interrupt Latency)**
  - Là khoảng thời gian tính từ lúc sự kiện xảy ra cho đến khi lệnh tại địa chỉ 0004h được thực thi
- **Độ trễ của các ngắt đồng bộ (thường là ngắt bên trong)**
  - Độ trễ khoảng 3 chu kỳ máy
- **Độ trễ của các ngắt không đồng bộ (thường là ngắt bên ngoài)**
  - Độ trễ khoảng 3 – 3.5 chu kỳ máy.

# SAO LƯU DỮ LIỆU KHI NGẮT

## ➤ Trong khi xảy ra ngắt:

- Chỉ có giá trị PC được lưu lại (vào ngăn xếp)
- Giá trị của các thanh ghi khác có thể bị thay đổi sau khi thực thi xong ISR

→ Do đó cần phải sao lưu và phục hồi giá trị các thanh ghi quan trọng

## ➤ Các thanh ghi mà người dùng nên lưu lại:

- Thanh ghi W
- Thanh ghi STATUS
- Thanh ghi PCLATH
- Thanh ghi được định nghĩa bởi người dùng.



# ƯU TIÊN NGẮT

- Vi điều khiển họ Mid-Range (PIC16F) thiết lập tất cả các ngắt có cùng một mức ưu tiên
- Việc thiết lập mức ưu tiên khác nhau cho các nguồn ngắt là do người sử dụng
- Để thiết lập mức ưu tiên:
  - Bước 1: Xác định các nguồn ngắt
  - Bước 2: Xác định thứ tự của các ISR.

# ƯU TIÊN NGẮT

- **Ví dụ:** Thiết lập thứ tự ưu tiên từ cao xuống thấp cho các ngắt: Port B, Timer 2 và Timer 1

```
void interrupt isr(void)
```

```
{  
  if (RBIE == 1) && (RBIF == 1)           ①  
    PORTB_ISR()
```

```
  if (TMR2IE == 1) && (TMR2IF == 1)         ②  
    Timer2_ISR()
```

```
  if (TMR1IE == 1) && (TMR1IF == 1)         ③  
    Timer1_ISR()
```

```
}
```

# Các lưu ý

➤ Ngắt nội:

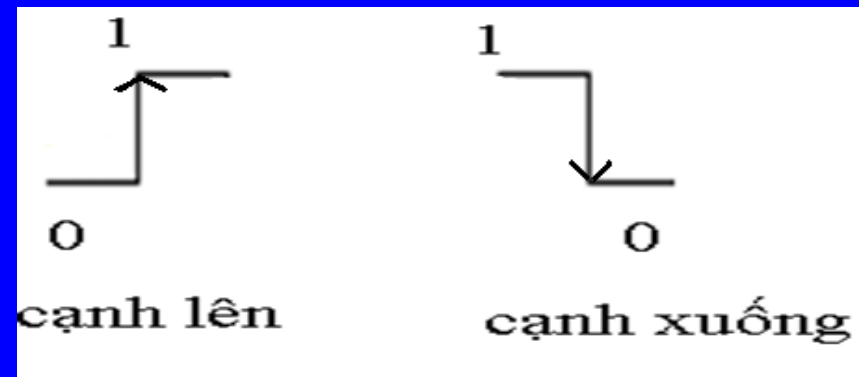
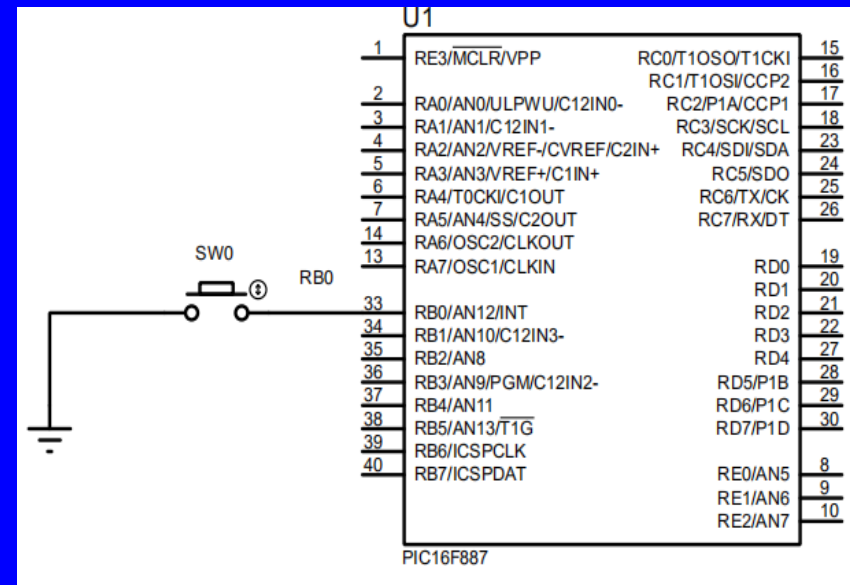
## 1. Ngắt ngoài : (sử dụng RB0)

GIE=1 : cho phép ngắt toàn cục

INTF=0 : xóa cờ ngắt ngoài

INTE=1: cho phép ngắt ngoài

INTEDG =1/0: chọn kích cạnh lên/cạnh xuống



# Các lưu ý

➤ Ngắt nội:

## 2. Ngắt PORTB

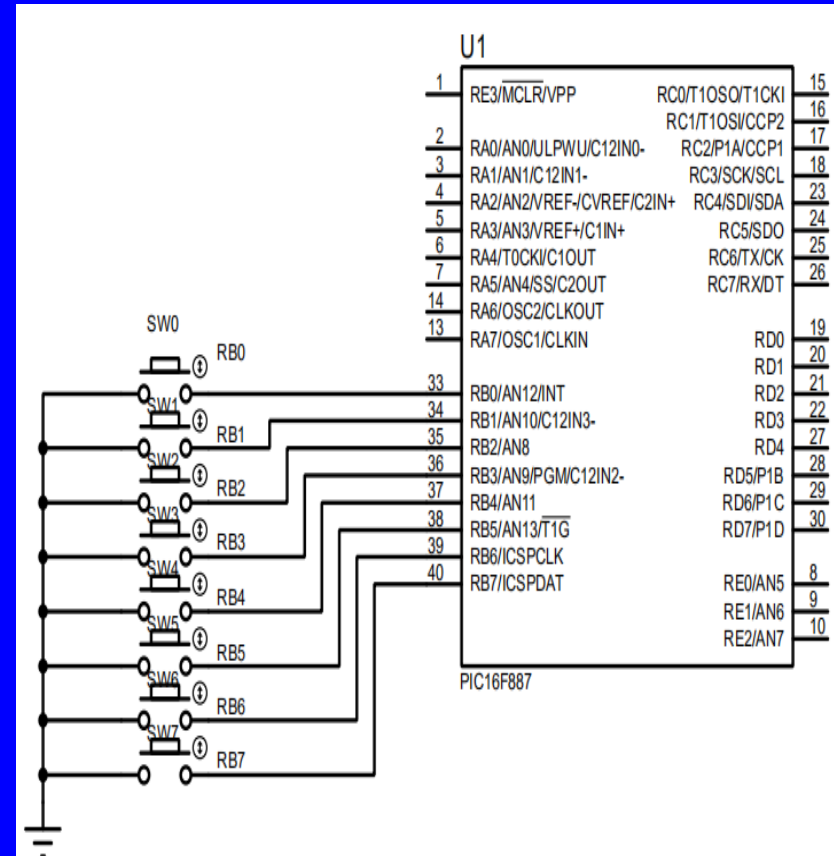
(sử dụng RB0 :RB7)

GIE=1 :Cho phép ngắt toàn cục

RBIF=0 : xóa cờ ngắt PORTB

RBIE=1: cho phép ngắt PORTB

**IOCBX = 1 (X =0:7):** cho phép ngắt ở chân RBX của PORTB

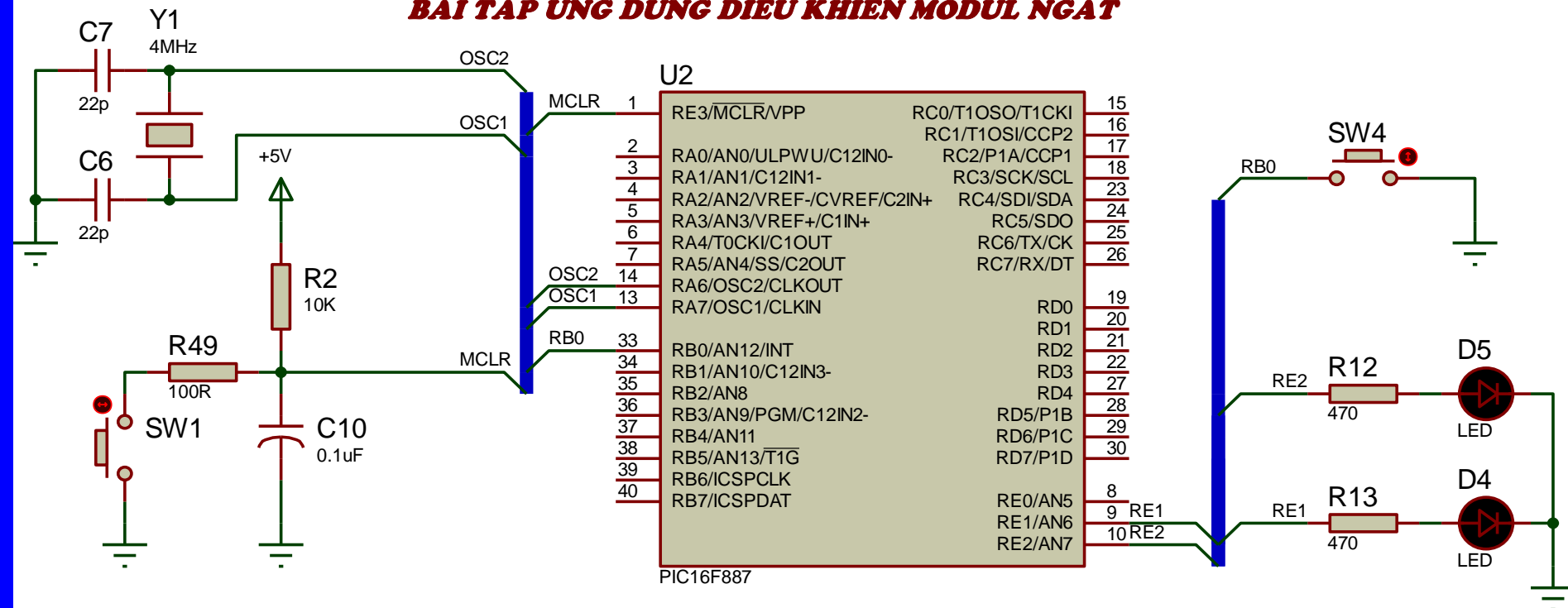


# VÍ DỤ MINH HỌA

➤ **Ví dụ 1:** Dựa vào sơ đồ, viết chương trình điều khiển LED D4, D5 sáng/tắt theo kiểu đếm lên nhị phân  $00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow \dots$  tương ứng cho mỗi lần nhấn SW4. Sử dụng tính năng ngắt ngoài.

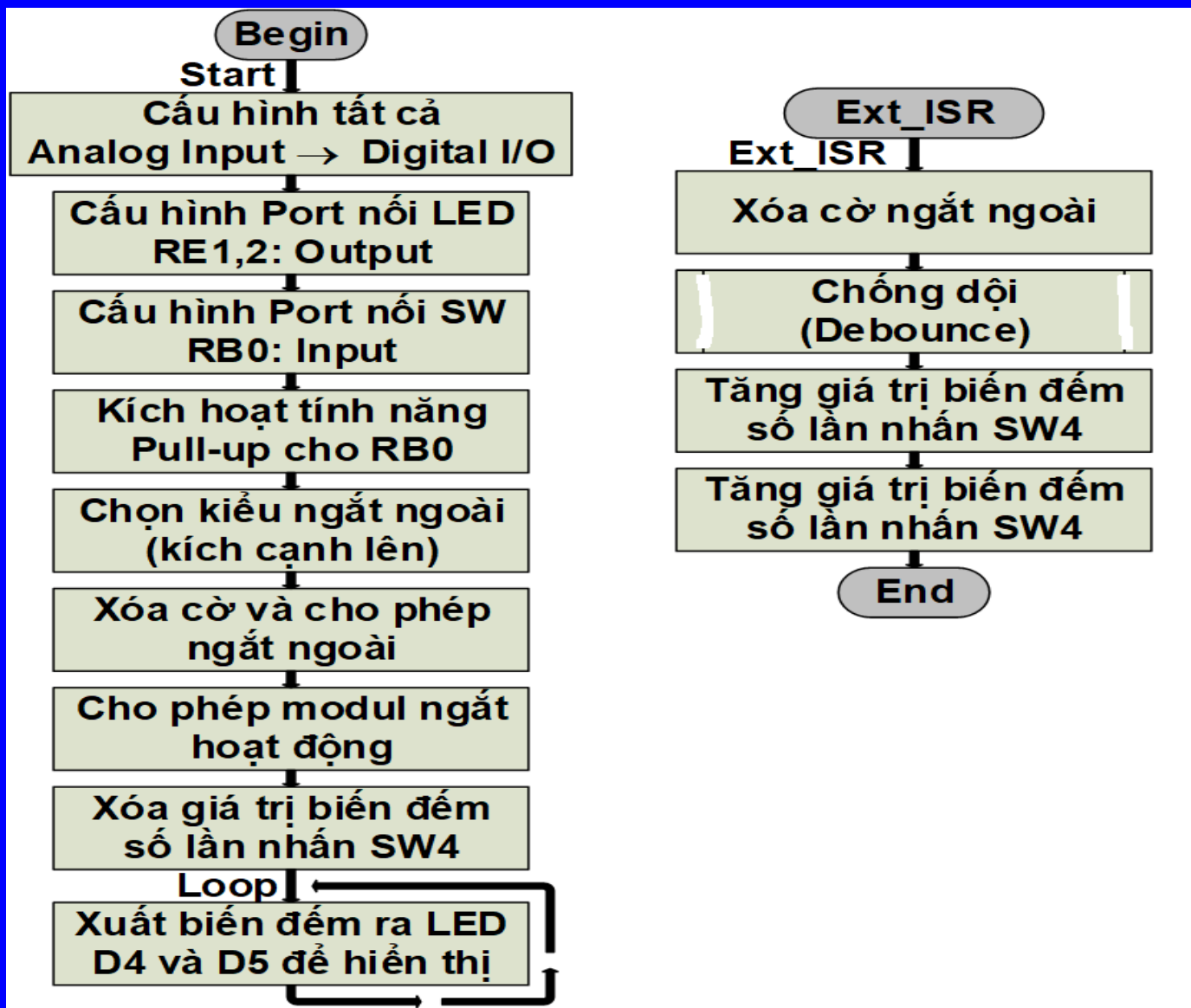
• **Sơ đồ nguyên lý:**

## BAI TAP UNG DUNG DIEU KHIEN MODUL NGAT



# VÍ DỤ MINH HỌA

## • Giải thuật:



## ➤ Bảng trạng thái

	RE2	RE1	RE0
	0	0	0
	0	0	1
	0	1	0
	0	1	1
	1	0	0
	1	0	1
	1	1	0
	1	1	1

# 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 push_count;
```

```
void interrupt isr(void)
```

```
{  
    __delay_ms(5); //chống dội
```

```
  
    push_count++; push_count++;  
    INTF = 0;  
}
```

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

```
void main (void)
```

```
{  
    ANSEL = ANSELH = 0;  
    TRISE1 = 0;  
    TRISE2 = 0;  
    PORTE = 0;
```

```
  
    TRISB0 = 1;  
    nRBPU = 0;  
    WPUB = 0x01;
```

```
  
    INTEDG = 1;  
    INTF = 0;  
    INTE = 1;  
    GIE = 1;
```

```
    push_count = 0;
```

```
    while(1)
```

```
    {  
        PORTE = push_count;  
    }
```

```
}
```

# Lưu ý

Đối với MPLABX 5.3 thay bằng cấu trúc

```
void __interrupt () my_isr_routine (void) {  
  
}
```

Đối với MPLABX 5.45 thay bằng cấu trúc

```
void __interrupt () isr(void) {  
  
}
```

Đối với MPLABX 6.0 thay bằng cấu trúc

```
void __interrupt() ISR(void)
```

Đối với MPLABX 6.5 thay bằng cấu trúc

```
void interrupt _isr(void)
```

Đối với MPLABX 6.05, 6.15 thay bằng cấu trúc

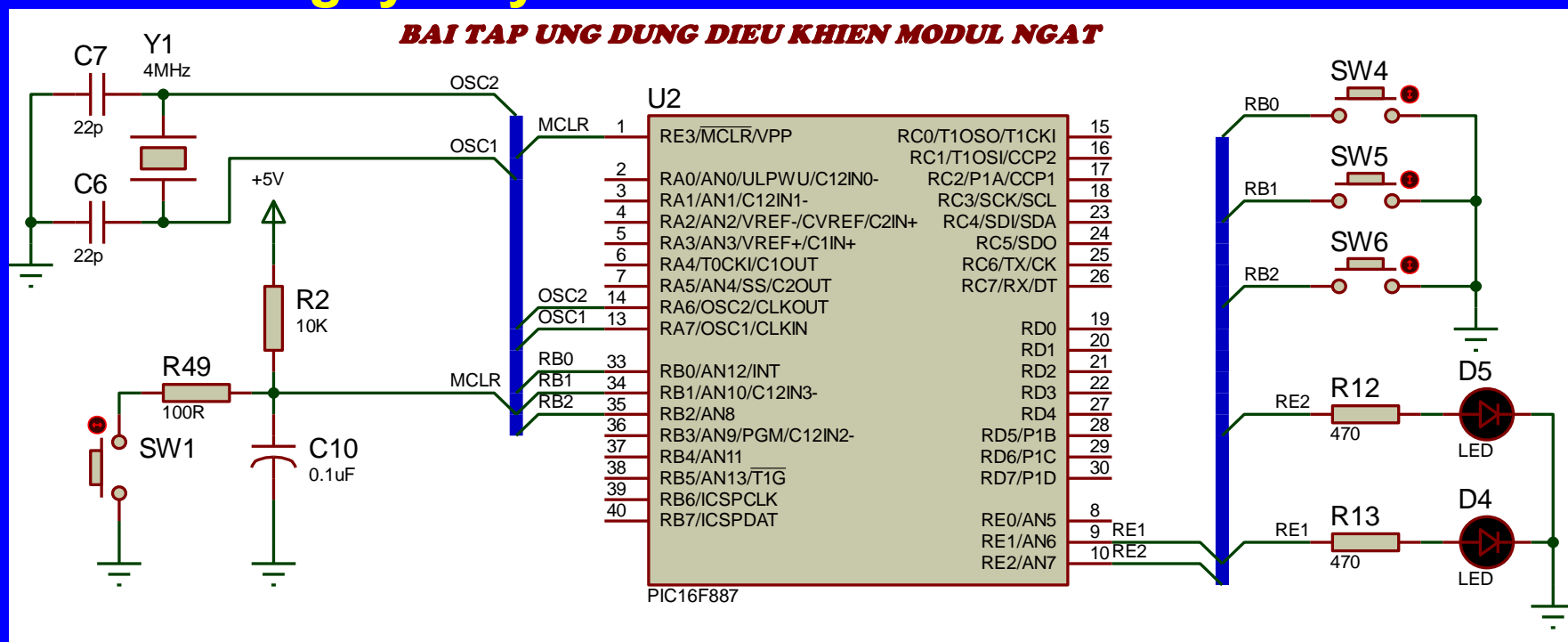
```
void __interrupt() isr(void)
```

# VÍ DỤ MINH HỌA

➤ **Ví dụ 2:** Dựa vào sơ đồ, viết chương trình điều khiển LED D4, D5:

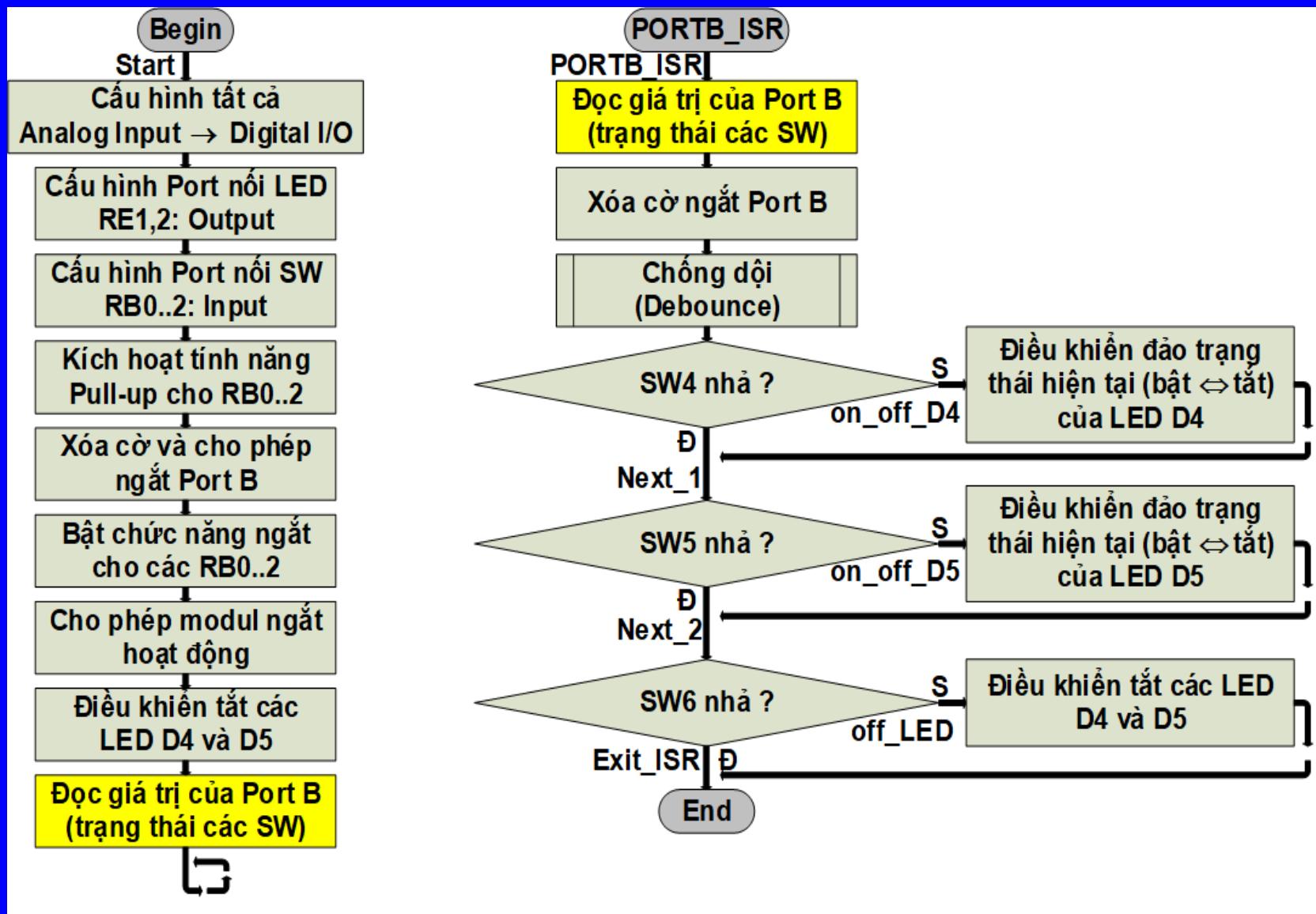
- Nhấn SW4 → D4 sáng hoặc tắt
- Nhấn SW5 → D5 sáng hoặc tắt
- Nhấn SW6 → D4, D5 tắt
- Sử dụng tính năng ngắt Port B

• **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)
{
    PORTB;
    RBIF = 0;
    __delay_ms(5); //chống dội

    if (!RB0)
        RE1 = !RE1;
    else if (!RB1)
        RE2 = !RE2;
    else if (!RB2)
        PORTE = 0x00;
}
```

# VÍ DỤ MINH HỌA

- Chương trình (Hi-Tech C):

```
void main (void)
```

```
{  
  ANSEL = 0;  
  ANSELH = 0;  
  
  TRISE1 = 0;  
  TRISE2 = 0;  
  
  TRISB0 = 1;  
  TRISB1 = 1;  
  TRISB2 = 1;  
  
  nRBPU = 0;  
  WPUB = 0x07;
```

```
  RBIF = 0;
```

```
  RBIE = 1;
```

```
  IOCB0 = 1;
```

```
  IOCB1 = 1;
```

```
  IOCB2 = 1;
```

```
  GIE = 1;
```

```
  PORTE = 0x00;
```

```
  PORTB; //đọc gtri PORTB
```

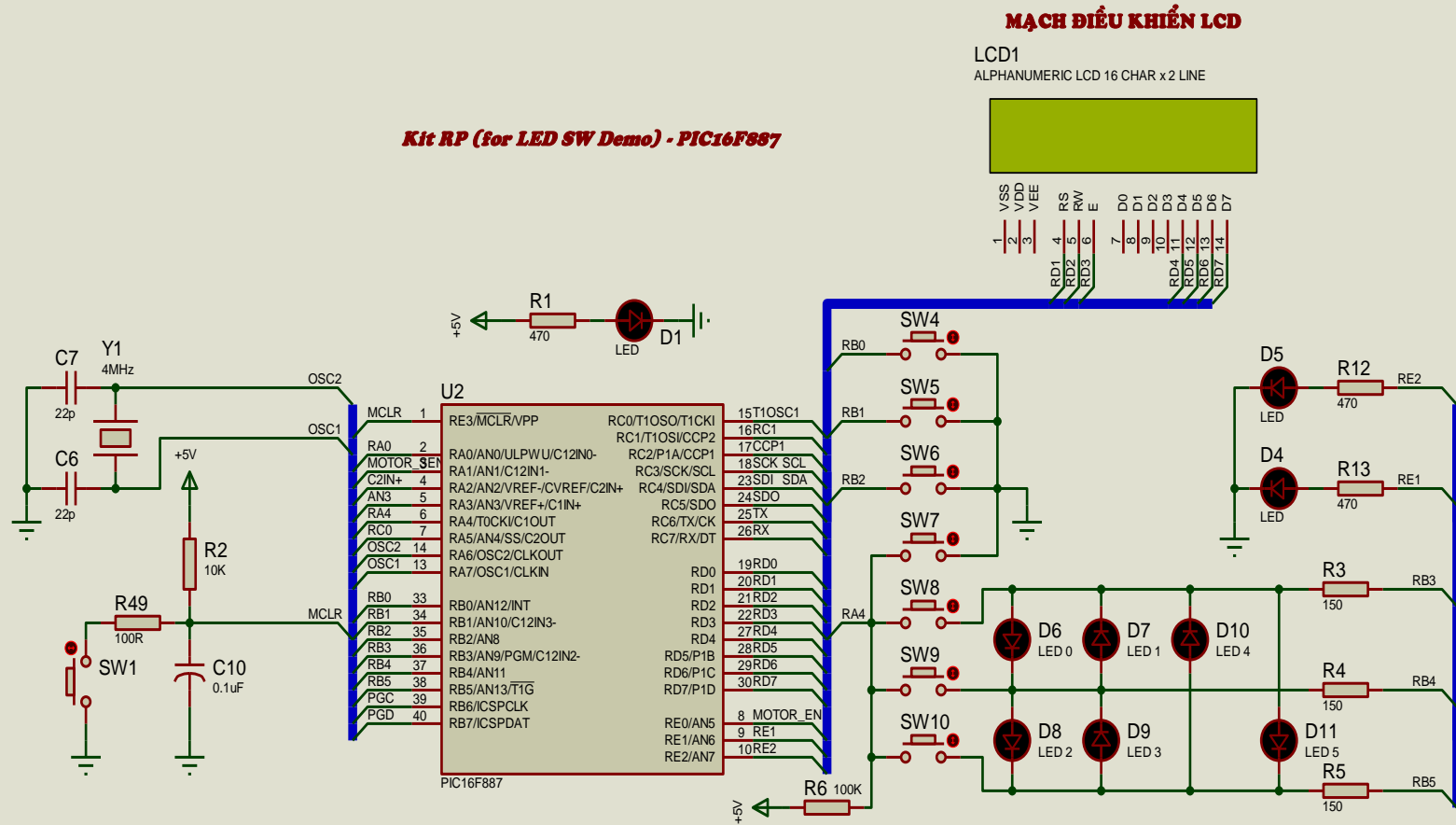
```
  while(1); //kết thúc ctrình
```

```
}
```



# BÀI TẬP ỨNG DỤNG

## Cho sơ đồ mạch kit RP như hình vẽ



# BÀI TẬP ỨNG DỤNG

## Dựa vào sơ đồ mạch kit RP

- **Bài tập 1:** Viết chương trình sử dụng ngắt ngoài để thực hiện việc tăng biến đếm khi nhấn\_nhả SW4. Nếu giá trị biến đếm bằng 20 thì reset giá trị biến đếm về 00 và hiển thị giá trị đếm lên LCD như sau:

Tên sinh viên dem

xx

xx: giá trị biến đếm (00÷20)

line 1\_LCD: tên T.Việt không dấu hiển thị sản trên LCD.

# BÀI TẬP ỨNG DỤNG

## Dựa vào sơ đồ mạch kit RP

➤ **Bài tập 2:** Viết chương trình gồm 3 bộ đếm độc lập S1, S2, S3 khi thay đổi trạng thái SW4(S1), SW5(S2), SW6(S3) thì giá trị mỗi bộ đếm tăng 1 đơn vị bằng cách sử dụng ngắt Port B. Nếu giá trị đếm bằng 10 thì reset giá trị đếm về 00 và hiển thị giá trị đếm lên LCD như sau:

S1	S2	S3
xx	xx	xx

xx: giá trị biến đếm (00÷10)

# BÀI TẬP ỨNG DỤNG

## Dựa vào sơ đồ mạch kit RP

➤ **Bài tập 3:** Viết chương trình hiển thị lên LCD nội dung như sau:

Bài tập ngắt  
Họ và tên sinh viên

**Yêu cầu :-** Nội dung LCD chớp tắt liên tục,  
- Nhấn\_nhả SW4 thì 6 led sáng đuôi từ D6÷D11 , sử dụng tính năng ngắt ngoài để nhận dạng trạng thái SW4.  
-Thời gian thay đổi trạng thái là 500ms.

# YÊU CẦU BUỔI HỌC

- Mỗi sinh viên tự viết code, chạy mô phỏng trên phần mềm Proteus đúng yêu cầu bài toán:

1. Bài tập 1 : 3đ

2. Bài tập 2 : 3đ

3. Bài tập 3 : 4đ

- Thời gian làm bài :

- Thời gian chấm bài :