BÀI GIẢNG ĐIỆN TỬ

Học phần: KĨ THUẬT VI XỬ LÝ

Bài 3. Ngắt ngoài

NGẮT NGOÀI

Mục tiêu:

- *Sinh viên nhớ và hiểu:*

- Tổ chức ngắt của PIC18F4520; Quá trình thực hiện ngắt của vi điều khiển;

  Các bit điều khiển ngắt ngoài; Các bước lập trình sử dụng ngắt ngoài.

- Các ứng dụng sử dụng ngắt ngoài.

- *Sinh viên vận dụng các kiến thức đã học để lập trình, mô phỏng hoạt động của ngắt ngoài, bao gồm:*

- 01 nguồn ngắt với mức ưu tiên cao.

- 02 nguồn ngắt.

(1) KHÁI QUÁT VỀ NGẮT

- **Khái niệm về ngắt:**

Là tạm thời dừng công việc hiện tại và chuyển sang thực hiện một nhiệm vụ khác (thường là cấp thiết, quan trọng hơn) sau đó lại quay lại thực hiện tiếp công việc cũ (đang thực hiện dở).

Ví dụ: Khi bạn đang đọc truyện (chẳng hạn vừa đọc hết dòng thứ 5), chuông điện thoại báo bạn có một cuộc điện thoại khẩn đến bất ngờ, bạn phải dừng đọc, đánh dấu lại vị trí đang đọc và nghe cuộc điện thoại, sau đó quay lại đọc tiếp từ dòng thứ 6.

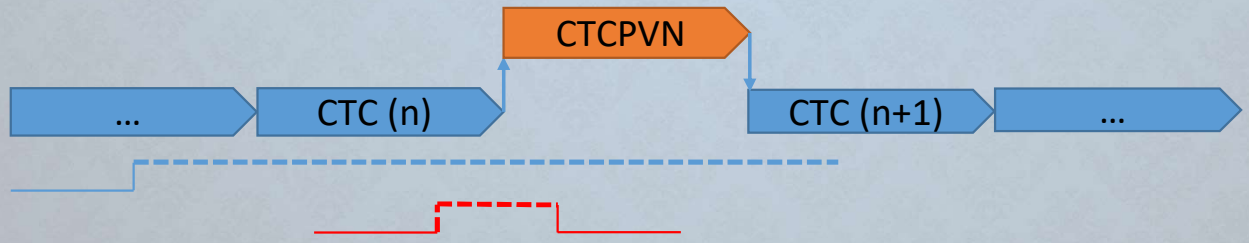
(1) KHÁI QUÁT VỀ NGẮT

Với vi điều khiển (VĐK):

Các lệnh tuần tự trong CTC | Dừng CTC | Tín hiệu ngắt | Lưu các thanh ghi vào ngăn xếp

Khi bạn đang đọc truyện (chẳng hạn vừa đọc hết dòng thứ 5), chuông điện thoại báo bạn có một cuộc điện thoại khẩn đến bất ngờ, bạn phải dừng đọc, đánh dấu lại vị trí đang đọc và nghe cuộc điện thoại, sau đó quay lại đọc tiếp từ dòng thứ 6.

Thực hiện các lệnh trong CTCPVN | Về CTC, lấy lại giá trị của các thanh ghi | Thực hiện tiếp các lệnh của CTC



Cho phép ngắt (---), Tín hiệu yêu cầu ngắt (---)

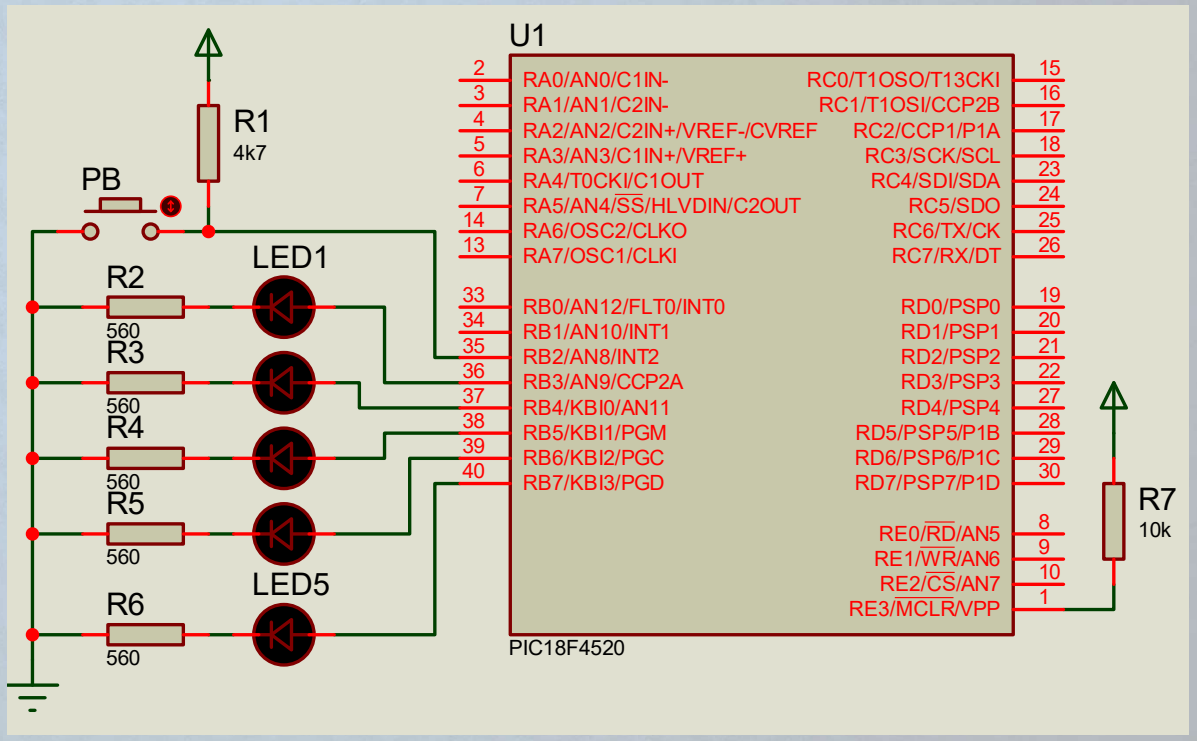
CTCPVN: Chương trình con phục vụ ngắt (Thuật ngữ tiếng Anh là Interrupt Service Routine –ISR)

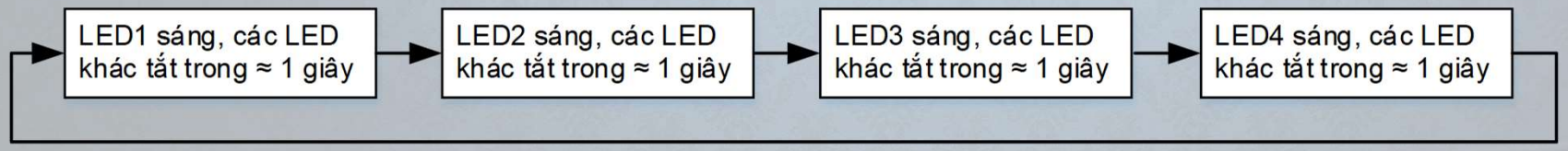
(1) KHÁI QUÁT VỀ NGẮT

- **Vì sao cần có ngắt?**

Giả thiết có mạch điện mô phỏng như hình bên. Viết chương trình điều khiển các LED1-4 sáng tuần tự theo chu trình như hình dưới; LED5 sáng (ngay lập tức) trong thời gian ~1s khi nhấn PB.

PB có thể được nhấn ở thời điểm bất kỳ.





(1) KHÁI QUÁT VỀ NGẮT

- **Vì sao cần có ngắt? (tiếp)**

Có thể thấy:

(1) Điều khiển LED1-4 sáng tuần tự:

```
#define LED5 PORTBbits.RB7
#define PB   PORTBbits.RB2
void main(void)
…
while(1)
        {
        PORTB=0b00001000; //LED1 sáng
        DelayKTCYx(100); // trễ
        …
        PORTB=0b01000000; //LED4 sáng
        DelayKTCYx(100); // trễ
        }
```

(2) Kiểm tra trạng thái nhấn của PB để điều khiển LED5 sáng:

```
if (PB==0){LED5=1; DelayKTCYx(100);}
```

Tuy nhiên các lệnh trên cần đặt ở điểm nào để có thể có thể thỏa mãn điều kiện "LED5 sáng (ngay lập tức) khi nhấn PB (ở thời điểm bất kỳ)"

?

(1) KHÁI QUÁT VỀ NGẮT

- **Vì sao cần có ngắt? (tiếp)**

```
...
while(1)
{
1.  PORTB=0b00001000; //LED1 sáng
2.  DelayKTCYx(100); // trễ
3.  PORTB=0b00010000; //LED2 sáng
4.  DelayKTCYx(100); // trễ
5.  PORTB=0b00100000; //LED3 sáng
6.  DelayKTCYx(100); // trễ
7.  PORTB=0b01000000; //LED4 sáng
8.  DelayKTCYx(100); // trễ
}
```

Đặt ngoài vòng lặp `while(1)`: Không kiểm tra được trạng thái của PB khi VĐK đã vào vòng lặp: Không được!!!

Nếu đặt trong vòng lặp `while(1)`: Không thể thỏa mãn điều kiện "LED5 sáng (ngay lập tức) khi nhấn PB (ở thời điểm bất kỳ)". Ví dụ: Giả sử đặt sau lệnh thứ 2, nếu VĐK đang thực hiện lệnh ở lệnh thứ 5 thì phải chờ sau khi VĐK thực hiện xong các câu lệnh thứ 5,6,7,8 và lặp lại thực hiện xong lệnh thứ 1,2 rồi mới đến lệnh kiểm tra PB.

Không được!!!

Đặt ngoài vòng lặp `while(1)`: Không kiểm tra được trạng thái của PB khi VĐK đã vào vòng lặp: Không được!!!

(1) KHÁI QUÁT VỀ NGẮT

- **Vì sao cần có ngắt? (tiếp)**

Vấn đề "điều khiển LED5 sáng ngay khi nhấn PB được giải quyết bằng ngắt thay vì các chương trình điều khiển tuần tự, cụ thể như sau:

Cho phép nhấn PB gây ra ngắt; các lệnh **`LED5=1; Delay10KTCYx(100);`** được đặt ở CTCPVN thay vì CTC

CTC

```
void main(void)
…
/*các lệnh cho phép nhấn PB gây ra ngắt*/
while(1)
      {
      PORTB=0b0000100; //LED1 sáng
      DelayKTCYx(100); // trễ
      …
      PORTB=0b0100000; //LED4 sáng
      DelayKTCYx(100); // trễ
      }
```

Nhấn PB

Nhấn PB

CTCPVN

```
void ctcpvn(void)
{
…
LED5=1; //LED5 sáng
DelayKTCYx(100);
}
```

(1) KHÁI QUÁT VỀ NGẮT

- **Vì sao cần có ngắt? (tiếp)**

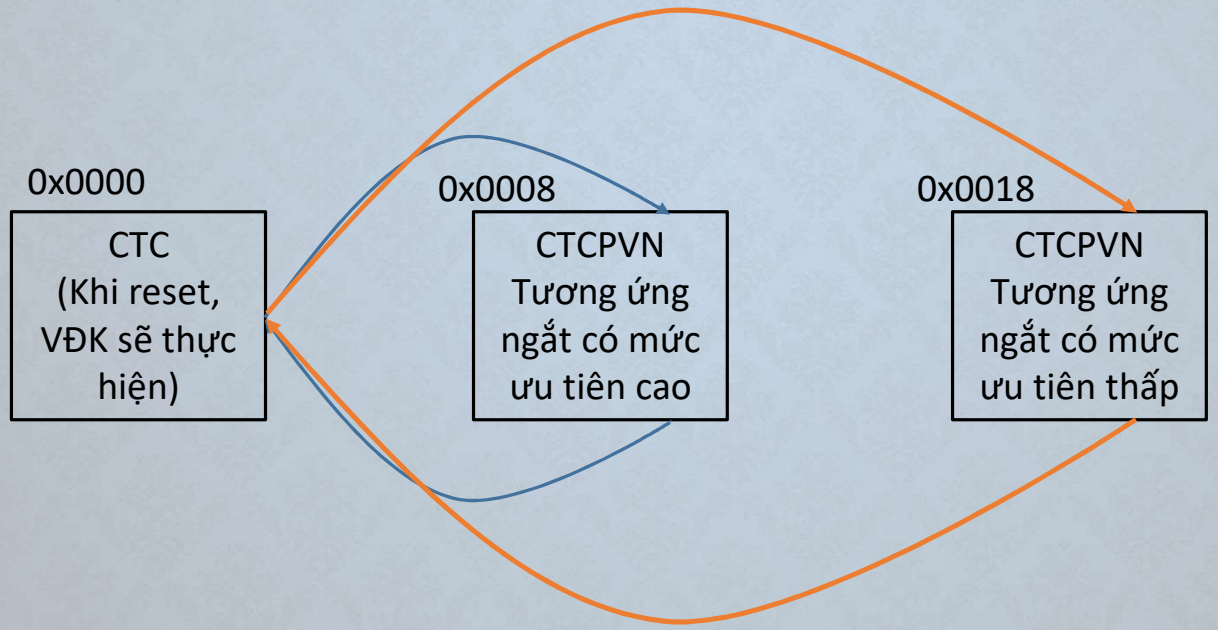
Thông qua kết quả mô phỏng có thể thấy:

- VĐK vừa có thể thực hiện các lệnh ở chương trình chính (điều khiển LED1-4 sáng tuần tự)
- Vừa có thể trao đổi dữ liệu (đọc trạng thái của PB và xuất mức "1" điều khiển LED5 sáng).

Điều này có được nhờ khối xử lý ngắt trên VĐK, nhờ khối này, vi điều khiển trở nên "đa nhiệm" hơn.

(2) NGẮT NGOÀI TRÊN PIC18F4520

- Vector ngắt: PIC18F4520 có 02 vector ngắt khác nhau (0x0008: ngắt ưu tiên cao; 0x0018: ngắt ưu tiên thấp)



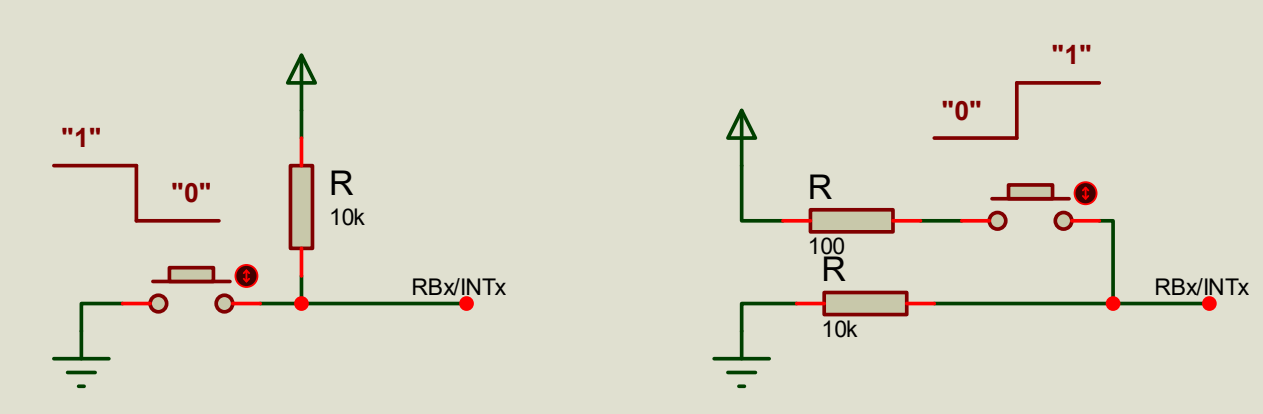
(2) NGẮT NGOÀI TRÊN PIC18F4520

- PIC18F4520 có 15 nguồn ngắt khác nhau được chia làm 2 loại:

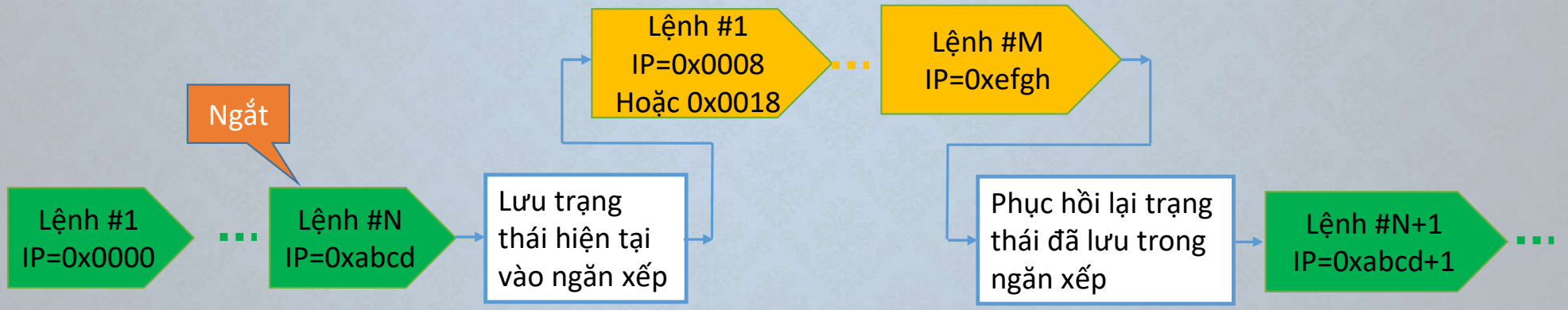
+ Ngắt trong (Internal Interrupt): Nguyên nhân gây ra ngắt được tạo ra từ bên trong VĐK, ví dụ: Tràn timer1, bộ biến đổi tượng tự - số (ADC) biến đổi xong…

+ Ngắt ngoài (External Interrupt): Nguyên nhân gây ra ngắt được tạo ra từ mạch điện bên ngoài VĐK. PIC184520 có 3 nguồn ngắt ngoài, gọi tắt là INT0, INT1 và INT2.

Để gây ra ngắt cần có mạch điện bên ngoài tạo ra sự chuyển mức logic từ “0” sang “1” hoặc sự chuyển mức logic từ “1” sang “0” đưa đến chân RB0/INT0 hoặc RB1/INT1 hoặc RB2/INT2



(3) QUÁ TRÌNH THỰC HIỆN NGẮT CỦA VI ĐIỀU KHIỂN



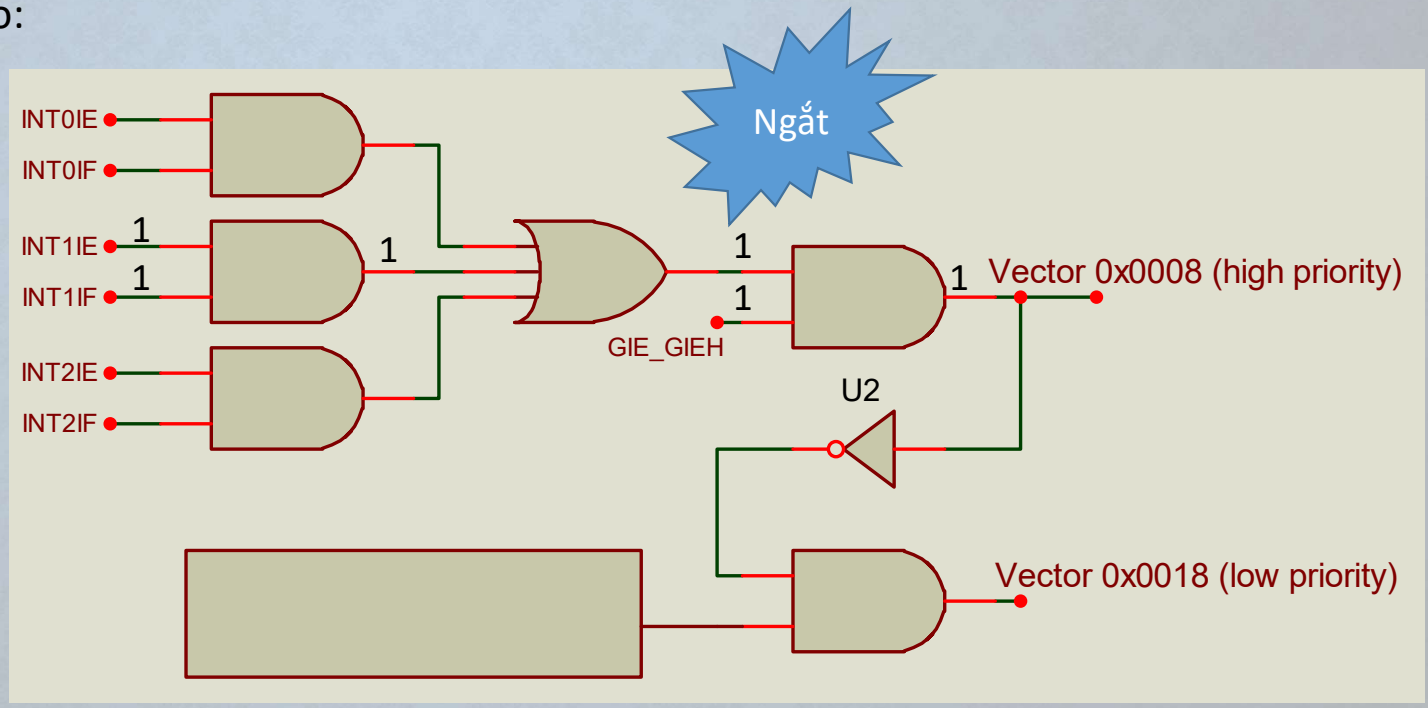
IP: Con trỏ lệnh (Instruction Pointer)

(4) CÁC BIT ĐIỀU KHIỂN NGẮT NGOÀI

Ở chế độ mặc định, 03 nguồn ngắt ngoài được đặt ở mức ưu tiên cao (hight priority). Vì vậy, để sử dụng các nguồn ngắt này chỉ cần khởi tạo:

- GIE=1: Cho phép ngắt toàn cục (Global Interupt Enable);
- INTxIE=1 (x=0, 1 hoặc 2): Cho phép nguồn ngắt x;
- INTEDGx=0 hoặc INTEDGx=1

Khi xảy ra ngắt, bit cờ ngắt của nguồn ngắt tương ứng (INTxIF) sẽ được đặt (set) bằng 1



(4) CÁC BIT ĐIỀU KHIỂN NGẮT NGOÀI

Loại bit \ Nguồn ngắt	INT0	INT1	INT2
Cho phép ngắt toàn cục (Global Interupt Enable)	**GIE_GIEH** Thanh ghi INTCON		
Cho phép nguồn ngắt	**INT0IE** Thanh ghi INTCON	**INT1IE** Thanh ghi INTCON3	**INT2IE** Thanh ghi INTCON3
Ngắt bằng sườn âm (=0) hoặc sườn dương (=1)	**INTEDG0** Thanh ghi INTCON2	**INTEDG1** Thanh ghi INTCON2	**INTEDG2** Thanh ghi INTCON2
Cờ ngắt	**INT0IF** Thanh ghi INTCON	**INT1IF** Thanh ghi INTCON3	**INT2IF** Thanh ghi INTCON3

(5) CÁC BƯỚC LẬP TRÌNH SỬ DỤNG NGẮT NGOÀI

1. Viết khung chương trình có sử dụng ngắt

2. Viết các lệnh của chương trình chính:

-Khởi tạo các PORT (ADCON1, TRIS)

-Set bit ngắt toàn cục INTCONbits. GIE_GIEH=1;

-Set bit cho phép nguồn ngắt tương ứng: INTCONbits.INT0IE hoặc INTCON3bits.INT1IE hoặc INTCON3bits.INT2IE

-Chọn ngắt bằng sườn âm/dương bằng cách xóa/set các bit tương ứng: INTCON2bits.INTEDG0=0 hoặc INTCON2bits.INTEDG1 =0 hoặc INTCON2bits.INTEDG2=0

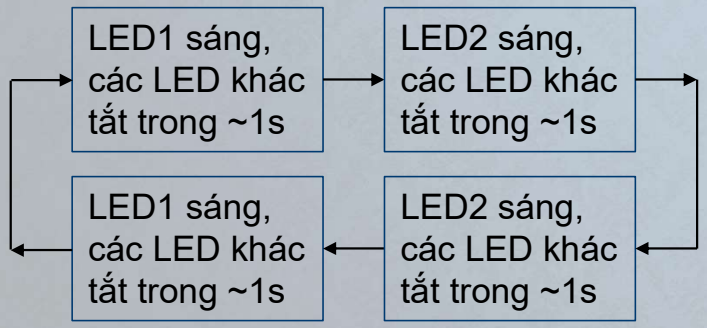
-Các lệnh theo đề bài

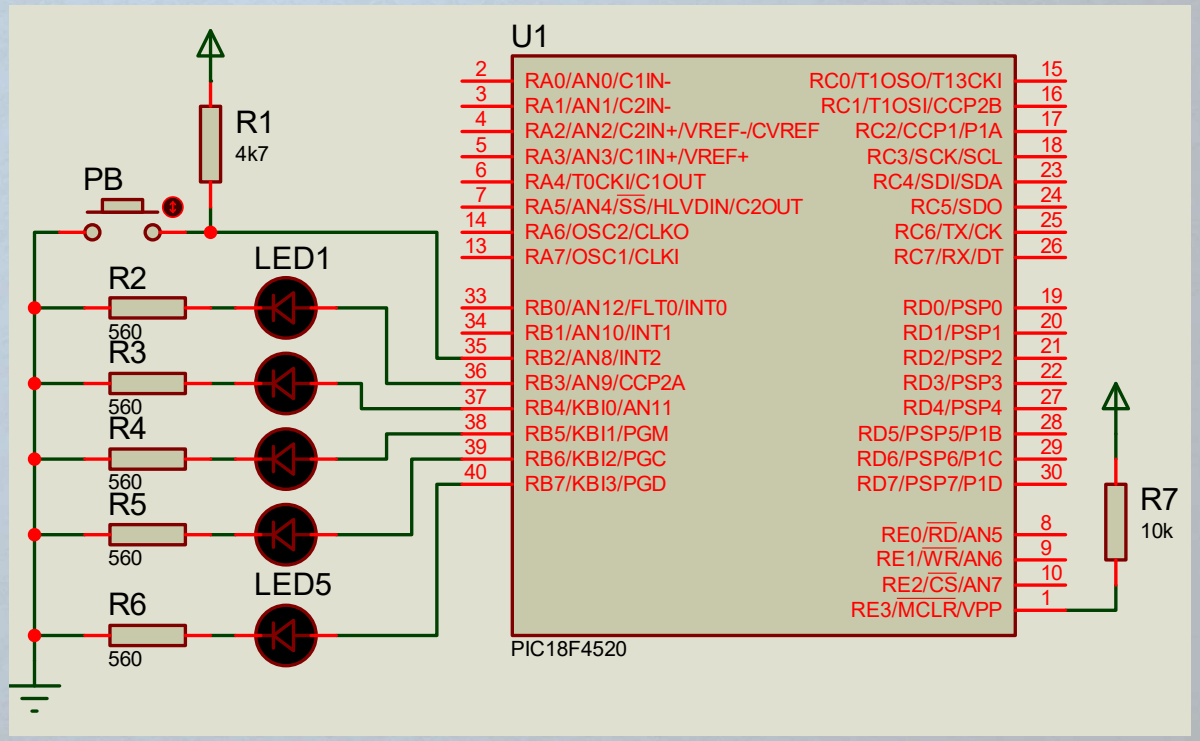
3. Viết các lệnh của CTCPVN:

-Xóa cờ ngắt tương ứng: INTCONbits.INT0IF=0 hoặc INTCON3bits.INT1IF=0 hoặc INTCON3bits.INT2IF=0

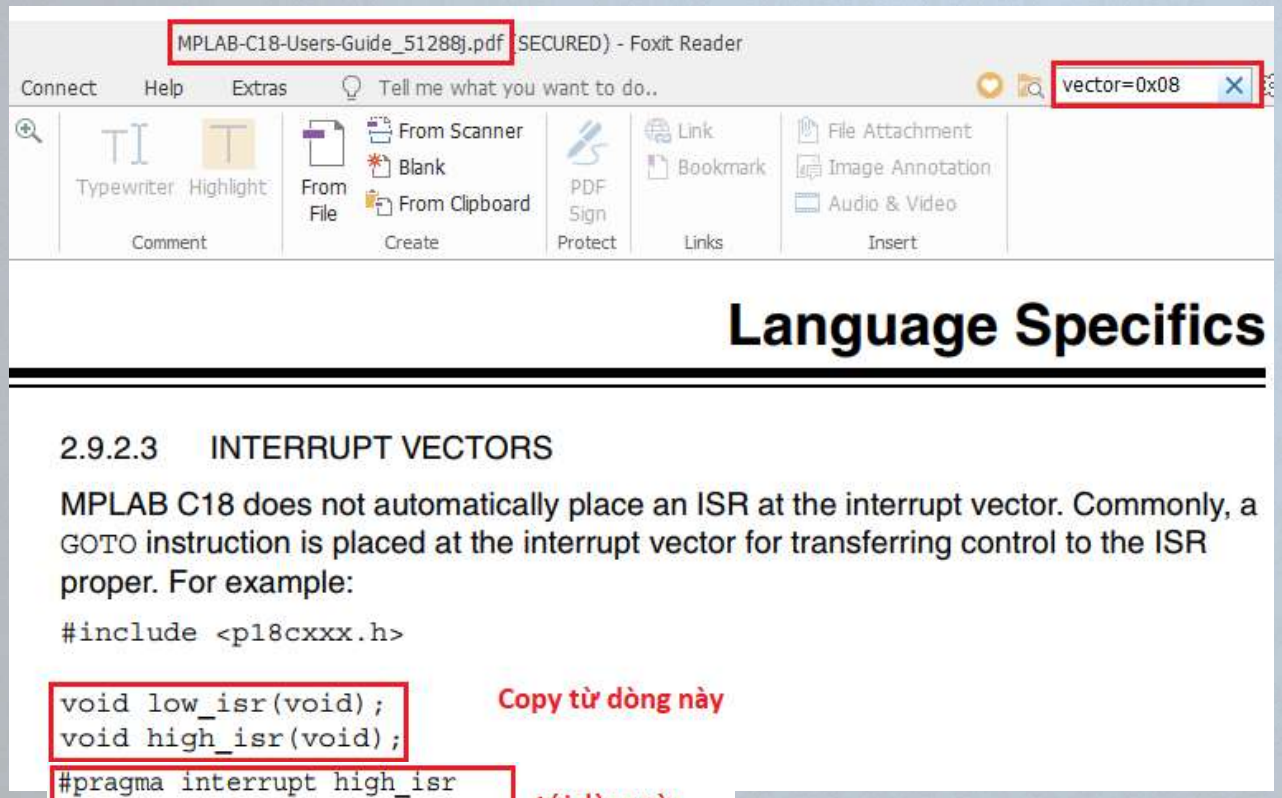
-Các lệnh theo đề bài

Giả thiết có mạch điện mô phỏng như hình bên. Viết chương trình điều khiển:
- Các LED1-4 sáng tuần tự theo chu trình như hình dưới;
- LED5 sáng trong thời gian ~1s khi nhấn PB.





1. Viết khung chương trình có sử dụng ngắt




```
void low_isr(void);
void high_isr(void);

#pragma code low_vector=0x18
void interrupt_at_low_vector(void)
{
_asm GOTO low_isr _endasm
}
#pragma code
#pragma interruptlow low_isr
void low_isr (void)
{
/* ... */
}
```

Đặt sau phần cấu hình (#pragma config)

```
#pragma code high_vector=0x08
void interrupt_at_high_vector(void)
{
_asm GOTO high_isr _endasm
}
#pragma code
#pragma interrupt high_isr
void high_isr (void)
{
/* ... */
}
```

2. Viết các lệnh của chương trình chính:

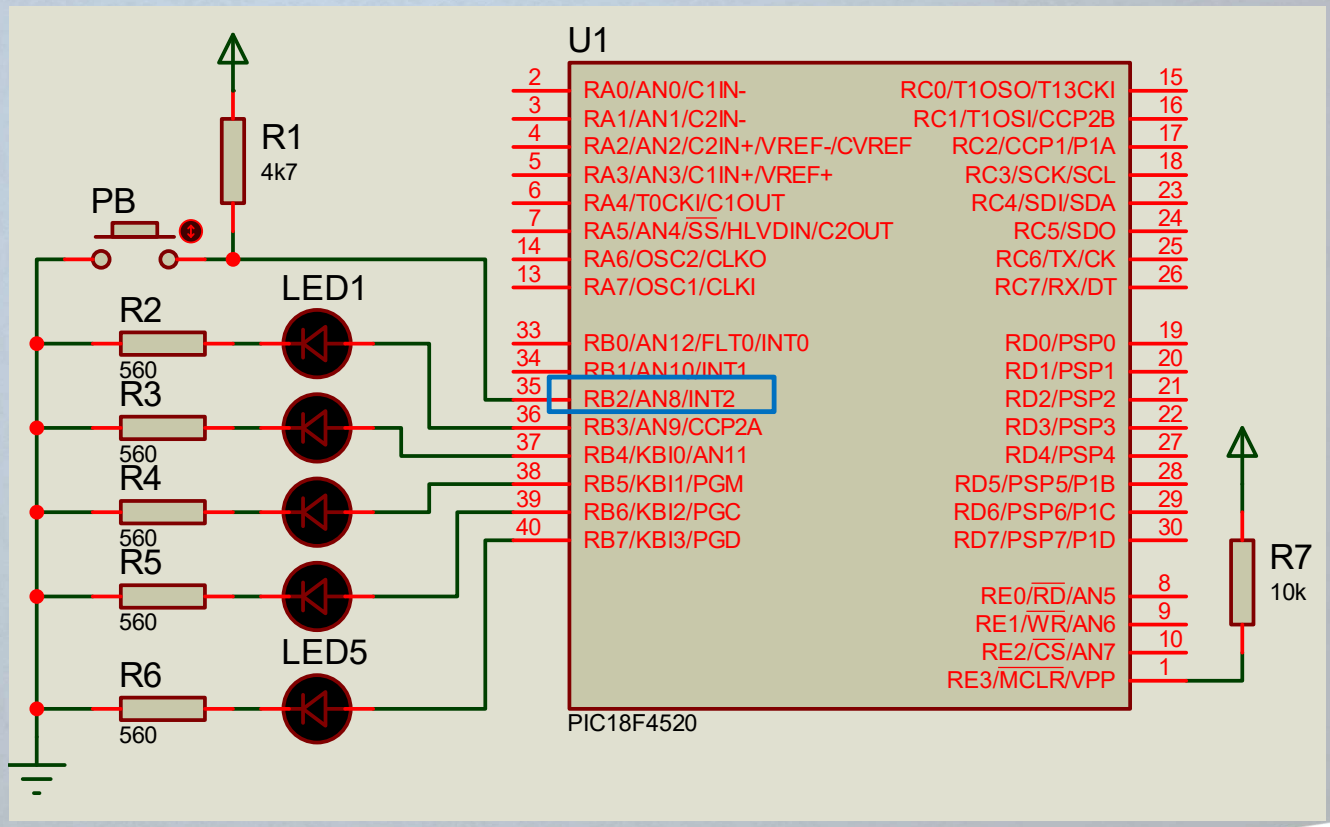
- ***Khởi tạo:***

+ Khởi tạo các PORT:

```
ADCON1=0x0F;
TRISB=0b00000100;
```

+ Khởi tạo ngắt INT2:

```
INTCONbits. GIE=1;
INTCON3bits.INT2IE=1;
INTCON2bits.INTEDG2=0;
```



2. Viết các lệnh của chương trình chính:

- ***Các lệnh theo đề bài***

Viết chương trình điều khiển:

CTC

- Các LED1-4 sáng tuần tự theo chu trình như hình dưới;

CTCPVN

- LED5 sáng trong thời gian ~1s khi nhấn PB.

```
while(1)
{
PORTB=0b00001000; //LED1 sáng
Delay10KTCYx(100); // trễ
PORTB=0b00010000; //LED2 sáng
Delay10KTCYx(100); // trễ
PORTB=0b00100000; //LED3 sáng
Delay10KTCYx(100); // trễ
PORTB=0b01000000; //LED4 sáng
Delay10KTCYx(100); // trễ
}
```

3. Viết các lệnh của CTCPVN

- ***Xóa cờ ngắt:***

```
INTCON3bits.INT2IF=0;
```

- ***Các lệnh theo đề bài***

Viết chương trình điều khiển:
- Các LED1-4 sáng tuần tự theo chu trình như hình dưới;
- LED5 sáng trong thời gian ~1s khi nhấn PB.

```
void high_isr (void)
{
INTCON3BITS.INT2IF=0;

LED5=1;
Delay10KTCYx(100);
LED5=0;
}
```

Các vấn đề cần ghi nhớ:

1. Khái niệm về ngắt, lý do cần sử dụng ngắt
2. Các vector ngắt (02 vector) và các nguồn ngắt ngoài trên PIC18F4520 (03 nguồn)
3. Quá trình thực hiện ngắt của vi điều khiển
4. Các bit điều khiển ngắt ngoài (04 loại)
5. Các bước lập trình sử dụng ngắt ngoài (03 bước)
6. Lập trình, mô phỏng hoạt động của ngắt INT2 với mức ưu tiên cao.