Bài 5. Ngắt quãng (interrupt).

Giới thiệu

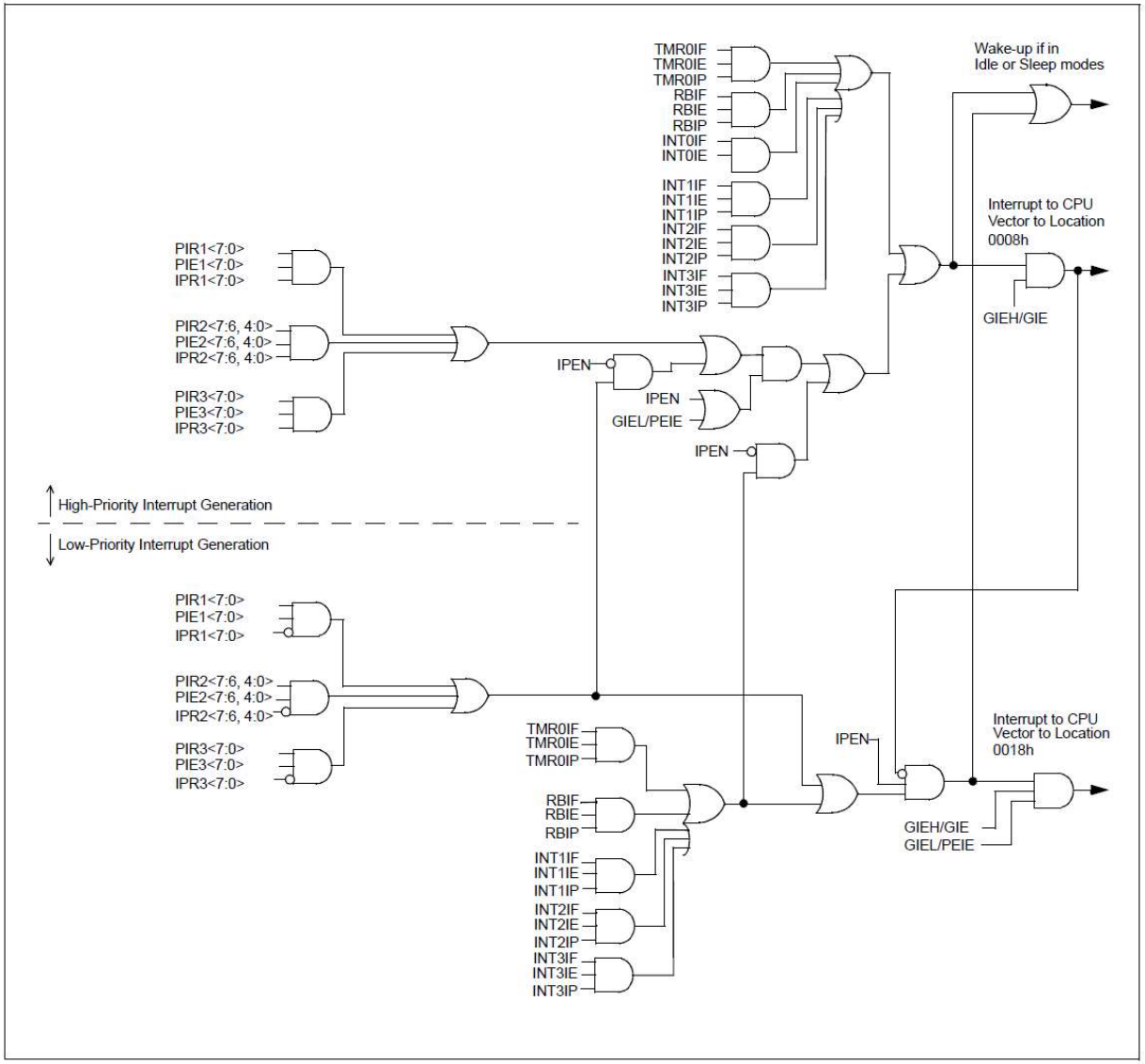
- Ngắt quãng là gì?
  - Ngắt quãng là sự kiện (event) một hoạt động xảy ra đột xuất cần thiết tạm ngưng chương trình đang chạy để chuyển sang chạy một chương trình khác phuc vụ cho yêu cầu đột xuất đó (Interrupt Service Routine ISR or Interrupt Handler).
  - Ngắt quãng thường được dùng cho các thiết bị ngoại vi để thay thế cho chế độ kiểm tra trạng thái (polling mode) của CPU khi trao đổi thông tin với thiết bị ngoại vi.
  - Chế độ kiểm tra trạng thái làm lãng phí thời gian CPU (trong hàng triệu lần kiểm tra mới có 1 lần thiết bị ngoại vi đáp ứng), làm giảm "tuổi thọ" CPU.

Giới thiệu (tt)

- PIC18F8722 có nhiều nguồn ngắt và có 2 cấp độ ưu tiên (priority) cao và thấp cho tất cả nguồn ngắt.
  - Ngắt có độ ưu tiên cao : dùng vector địa chỉ 0008h
  - Ngắt có độ ưu tiên thấp : dùng vector địa chỉ 0018h
- Ngắt quãng ưu tiên cao có thể tạm dừng ngắt quãng ưu tiên thấp đang thi hành.
- Ngắt quãng ưu tiên thấp sẽ không được phục vụ khi đang phục vụ cho một ngắt quãng ưu tiên cao.

Giới thiệu (tt)

FIGURE 10-1: PIC18F8722 FAMILY INTERRUPT LOGIC



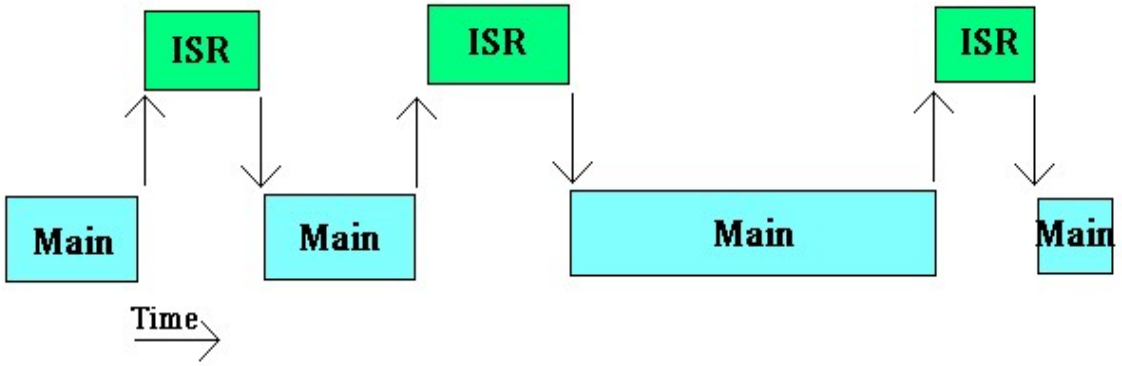
Các nguồn ngắt quãng

- Ngắt ngoài RB0/INT0, RB1/INT1, RB2/INT2, RB3/INT3.
- Timer n (n=0-4): cờ TMRnIF=1 khi đếm tràn:
  - Đếm 8 bit : FFh → 00h
  - Đếm 16 bit : FFFFh → 0000h
- PortB interrupt-on-change.
- Capture/Compare/PWM interrupt.
- USART, MSSP.
- ADC.
- Comparator.
- HLVPD.
- . . .

Cách thức hoạt động

Program execution without intrrupts :

Program execution with intrrupts :



ISR : Intrrupt Service Routin

Thiết kế chương trình sử dụng ngắt quãng

- Polling: chương trình chính kiểm tra các cờ, đợi cho có sự kiện xảy ra (không hiệu quả trong một số trường hợp).
- Interrupt driven: CPU có thể thực hiện những công việc khác, không cần chờ các sự kiện xảy ra (hiệu quả trong các ứng dụng điều khiển).
- I/O processor: các processor chuyên dụng xử lý các sự kiện I/O, không can thiệp vào hoạt động của CPU. (tốt nhất nhưng chi phí cao).

Xử lý ngắt

- Khi một nguồn ngắt được chấp nhận thì:
  - CPU hoàn tất lệnh hiện tại.
  - Cờ GIE (hoặc GIEH/GIEL) bị xóa để ngăn cản ngắt quãng cùng mức ưu tiên khác xảy ra.
  - Lưu PC vào stack.
  - Nạp địa chỉ 0008h hoặc 0018h vào PC để nhảy đến các ISR.

ISR

- Trong ISR, ta cần thực các việc:
  - Kiểm tra nguồn ngắt thông qua các cờ IF.
  - Xóa cờ ngắt IF (tránh xảy ra đệ qui).
  - Thực hiện xuất/nhập với thiết bị nguồn ngắt.
  - Chuẩn bị cho phép ngắt xảy ra tiếp lần sau.
  - Kết thúc bằng lệnh RETFIE.
- Lệnh RETFIE sẽ lập các cờ GIE (hoặc GIEH/GIEL) để cho phép ngắt xảy ra tiếp, khôi phục PC từ stack.

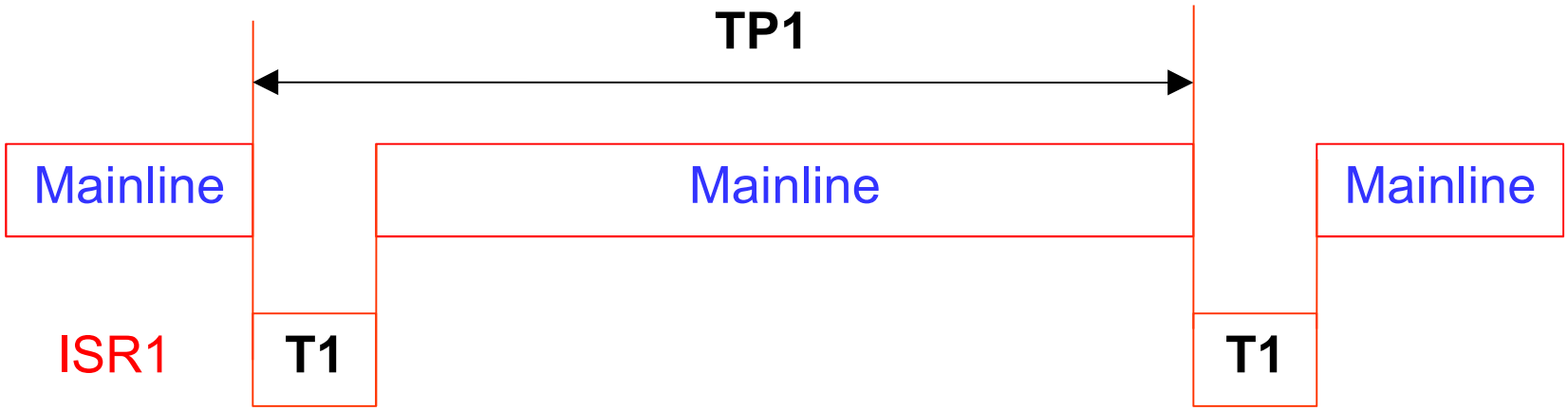
Thời gian ngắt (interrupt timing)

- Sử dụng cho ngắt có độ ưu tiên thấp.
  - TPi: Thời khoảng giữa các lần ngắt quãng xảy ra.
  - Ti : Thời gian thực hiện xong ISR.

$$Ti < TPi$$

Thời gian ngắt (interrupt timing)

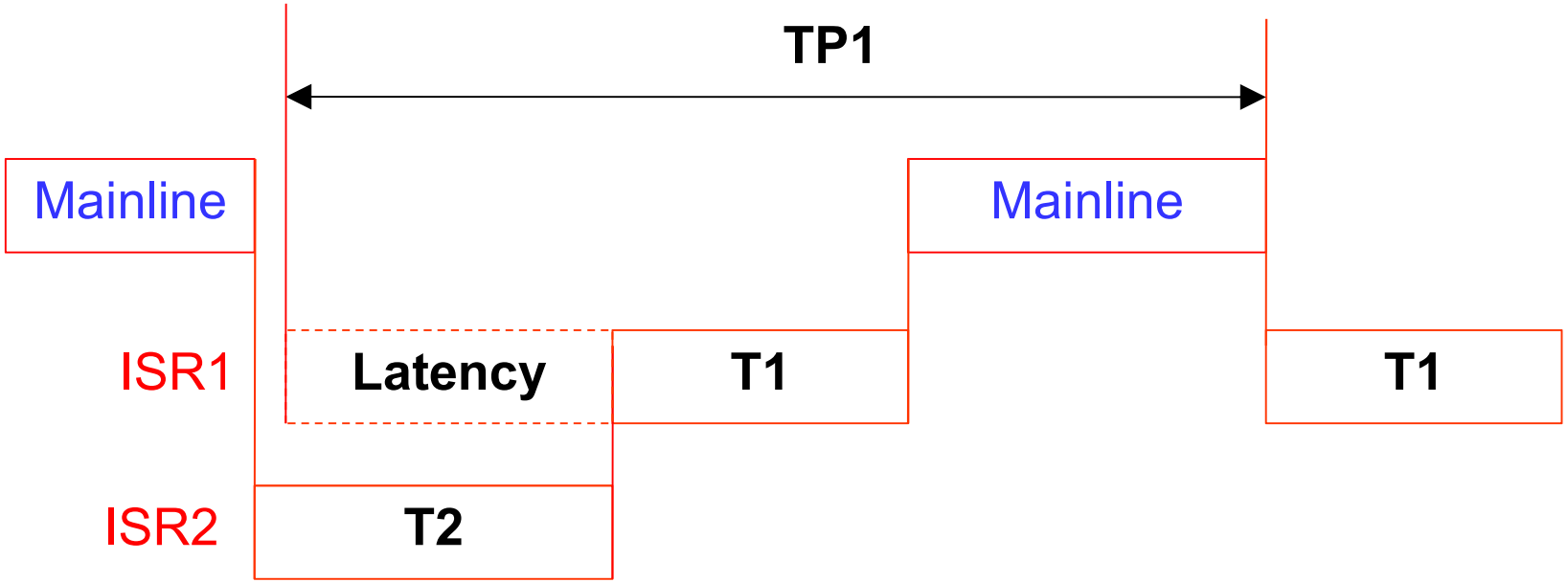
- Thời gian ngắt cho 1 nguồn ngắt



$$T1 < TP1$$

Thời gian ngắt (interrupt timing)

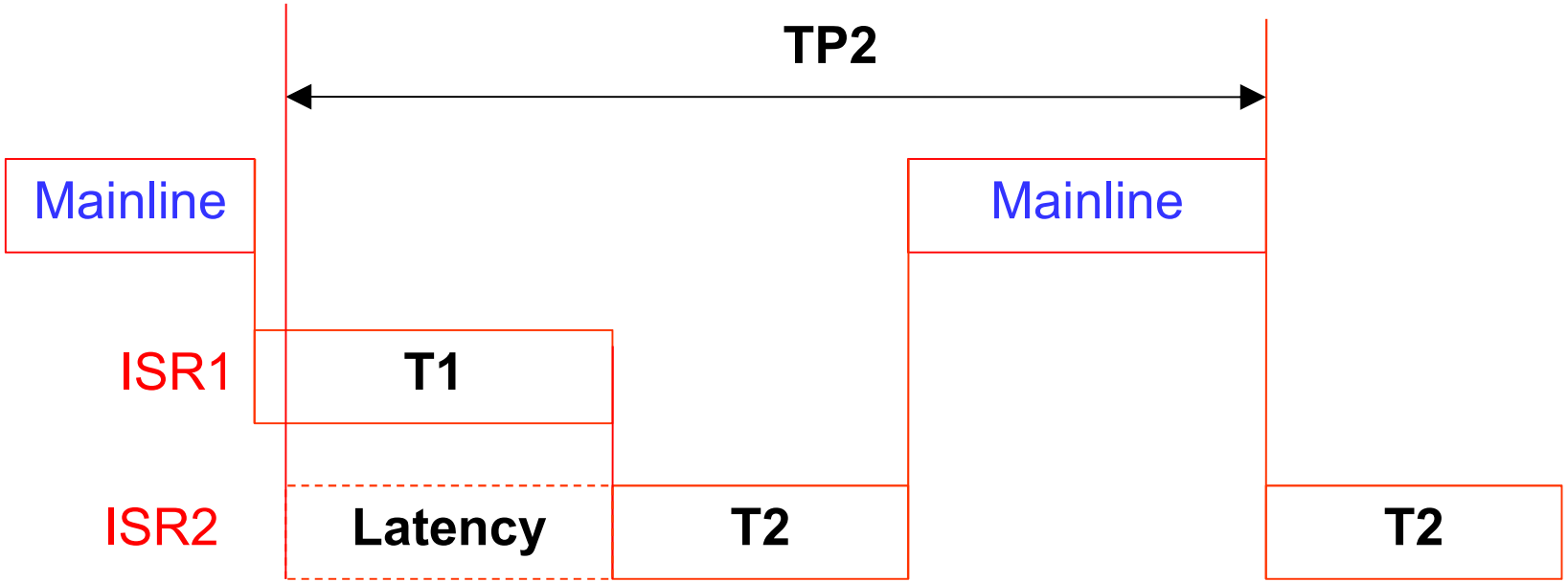
- Thời gian ngắt cho 2 nguồn ngắt.
- Trường hợp xấu nhất của ISR1 (đợi ISR2 xong).



$$T2 + T1 < TP1$$

Thời gian ngắt (interrupt timing)

- Thời gian ngắt cho 2 nguồn ngắt.
- Trường hợp xấu nhất của ISR2 (đợi ISR1 xong).



$$T1 + T2 < TP2$$

Cho phép và cấm các nguồn ngắt

- IPEN (RCON<7>): cho phép dùng độ ưu tiên cao/thấp.
- IPEN: enable (=1) - dùng ưu tiên cao/thấp
  - GIEH (INTCON<7>): cho phép các ngắt có độ ưu tiên cao.
  - GIEL (INTCON<6>): cho phép các ngắt có độ ưu tiên thấp.
- IPEN: disable (=0) - dùng ngắt theo ngoại vi
  - GIE (INTCON<7>): cho phép tất cả các nguồn ngắt xảy ra.
  - PEIE (INTCON<6>): cho phép tất cả các nguồn ngắt từ ngoại vi.

Các thanh ghi dùng trong ngắt quãng

- RCON : chỉ dùng bit 7 (IPEN)
- INTCON
- INTCON2
- INTCON3
- PIR1, PIR2, PIR3 : các bit IF
- PIE1, PIE2, PIE3 : các bit IE
- IPR1, IPR2, IPR3 : các bit IP

Các bit dùng trong các loại ngắt

Name	Priority bit	Local Enable bit	Local Flag bit
INT0 external interrupt	(luôn luôn cao)	INTCONbits.INT0IE	INTCONbits.INT0IF
INT1 external interrupt	INTCON3bits.INT1IP	INTCON3bits.INT1IE	INTCON3bits.INT1IF
INT2 external interrupt	INTCON3bits.INT2IP	INTCON3bits.INT2IE	INTCON3bits.INT2IF
INT3 external interrupt	INTCON2bits.INT3IP	INTCON3bits.INT3IE	INTCON3bits.INT3IF
RB port change interrupt	INTCON2bits.RBIP	INTCONbits.RBIE	INTCONbits.RBIF
TMR0 overflow interrupt	INTCON2bits.TMR0IP	INTCONbits.TMR0IE	INTCONTbits.TMR0IF
TMR1 overflow interrupt	IPR1bits.TMR1IP	PIE1bits.TMR1IE	PIR1bits.TMR1IF
TMR2 to match PR2 interrupt	IPR1bits.TMR2IP	PIE1bits.TMR2IE	PIR1bits.TMR2IF
TMR3 overflow interrupt	IPR2bits.TMR3IP	PIE2bits.TMR3IE	PIR2bits.TMR3IF
TMR4 to match PR4 interrupt	IPR3bits.TMR4IP	PIE3bits.TMR4IE	PIR3bits.TMR4IF
CCP1 interrupt	IPR1bits.CCP1IP	PIE1bits.CCP1IE	PIR1bits.CCP1IF
CCP2 interrupt	IPR2bits.CCP2IP	PIE2bits.CCP2IE	PIR2bits.CCP2IF
A/D converter interrupt	IPR1bits.ADIP	PIE1bits.ADIE	PIR1bits.ADIF
USART receive interrupt	IPR1bits.RC1IP	PIE1bits.RCIE	PIR1bits.RCIF
USART transmit interrupt	IPR1bits.TX1IP	PIE1bits.TXIE	PIR1bits.TXIF
Sync. Serial port interrupt	IPR1bits.SSP1IP	PIE1bits.SSPIE	PIR1bits.SSPIF

Xử lý "vùng tranh chấp"

- Vùng tranh chấp là các tài nguyên có thể bị thay đổi bởi cả chương trình chính lẫn ISR.
- Khi chương trình chính thay đổi nội dung của vùng tranh chấp, cần theo qui trình sau:
  - Cấm tất cả nguồn ngắt (xóa GIE/GIEH).
  - Thực hiện thay đổi vùng tranh chấp.
  - Cho phép ngắt xảy ra lại (lập GIE/GIEH).
  - Có thể sử dụng các lệnh read-modify-write để không cho ngắt quãng chen ngang.

Khởi tạo ngắt quãng (tt)

- Có 3 bit điều khiển hoạt động của mỗi nguồn ngắt.
  - Chọn IPEN hay không.
    - Nếu IPEN=1, chọn tiếp Priority bit (xxxIP) :
      - =1 ưu tiên cao,
      - =0 ưu tiên thấp.
  - Flag bit (xxxIF) : cần xóa về 0.
  - Enable interrupt bit (xxxIE) : cần lập lên 1.

Ví dụ khởi tạo ngắt Timer 0

```
; cho phép ngắt xảy ra với độ ưu tiên thấp.
   RCONbits.IPEN=1;
   INTCON2bits.TMR0IP=0;
;  xoá cờ ngắt
   INTCONbits.TMR0IF=0;
; cho phép ngắt Timer0 xảy ra
   INTCONbits.TMR0IE=1;
```

Khởi tạo ngắt Timer 0

```
void timer0_init()
{    RCONbits.IPEN=1;          //cho phep dung uu tien
     INTCON2bits.TMR0IP=0;     //Timer0 uu tien thap
     INTCONbits.TMR0IF=0;      //xoa co ngat
     INTCONbits.TMR0IE=1;      //cho phep ngat Timer0
     INTCONbits.GIEH=1;        //cho phep ngat uu tien cao
     INTCONbits.GIEL=1;        //cho phep ngat uu tien thap
     T0CON = 0;                //dung Fosc/4, prescaler=2
     timer0_reset();
}
```

Chương trình phục vụ ngắt quãng Timer 0

```
#define SODEM10MS    -12500
#define SODEM500MS   50
unsigned char dem500ms;
void timer0_reset(void)
{    TMR0ON=0;
     TMR0=SODEM10MS;
     TMR0ON=1;
}
```

```
void xuly500ms()
{    led++;}
void interrupt low_priority timer0_isr(void)
{    if (TMR0IE & TMR0IF)
     {    TMR0IF=0;              //xoa co ngat
          timer0_reset();        //nap lai so dem
          if ((--dem500ms)==0)
          { xuly500ms();     //lam cong viec dinh ky theo timer
            dem500ms=SODEM500MS;
          }
     }
}
```