

# Chương 6: Hoạt động ngắt

# 6.1 Giới thiệu

- Phương pháp lặp
- Phương pháp ngắt

## 6.2 Tổ chức ngắt của 8051

<b>BIT</b>	<b>SYMBOL</b>	<b>BIT ADDRESS</b>	<b>DESCRIPTION (1 = ENABLE, 0 = DISABLE)</b>
IE.7	EA	AFH	Global enable/disable
IE.6	-	AEH	Undefined
IE.5	ET2	ADH	Enable Timer 2 Interrupt (8052)
IE.4	ES	ACH	Enable serial port interrupt
IE.3	ET1	ABH	Enable Timer 1 interrupt
IE.2	EX1	AAH	Enable external 1 interrupt
IE.1	ET0	A9H	Enable Timer 0 interrupt
IE.0	EX0	A8H	Enable external 0 interrupt

## 6.2 Tổ chức ngắt của 8051

- VD:

Cho phép ngắt của bộ định thời 1

Setb ET1

Setb EA

# Ưu tiên ngắt

<b>BIT</b>	<b>SYMBOL</b>	<b>BIT ADDRESS</b>	<b>DESCRIPTION</b> (1 = HIGHER LEVEL, 0 = LOWER LEVEL)
IP.7	-	-	Undefined
IP.6	-	-	Undefined
IP.5	PT2	0BDH	Priority for Timer 2 interrupt (8052)
IP.4	PS	0BCH	Priority for serial port interrupt
IP.3	PT1	0BBH	Priority for Timer 1 interrupt
IP.2	PX1	0BAH	Priority for external 1 interrupt
IP.1	PT0	0B9H	Priority for Timer 0 interrupt
IP.0	PX0	0B8H	Priority for external 0 interrupt

# Chuỗi vòng

- Nếu 2 ngắt có cùng mức ưu tiên xuất hiện đồng thời:
  - ➔ chuỗi vòng xác định ngắt nào được ưu tiên
  - ➔ Chuỗi vòng: ngắt ngoài 0, ngắt cho bộ định thời 0, ngắt ngoài 1, ngắt cho bộ định thời 1, ngắt do port nối tiếp, ngắt do bộ định thời 2.

# Các bit cờ tạo ra ngắt

<b>INTERRUPT</b>	<b>FLAG</b>	<b>SFR REGISTER AND BIT POSITION</b>
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	T1	SCON.1
Serial port	RI	SCON.0
Timer 2	TF2	T2CON.7 (8052)
Timer 2	EXF2	T2CON.6 (8052)

## 6.3 Xử lý ngắt

<b>INTERRUPT</b>	<b>FLAG</b>	<b>VECTOR ADDRESS</b>
System reset	RST	0000H
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial port	RI or TI	0023H
Timer 2	TF2 or EXF2	002BH



<b>Interrupt Number</b>	<b>Description</b>	<b>Address</b>
0	EXTERNAL INT 0	0003h
1	TIMER/COUNTER 0	000Bh
2	EXTERNAL INT 1	0013h
3	TIMER/COUNTER 1	001Bh
4	SERIAL PORT	0023h

## 6.4 Thiết kế chương trình sử dụng ngắt ngoài

```
#include "regx51.h"           /* khai bao thu vien AT89c51 */
unsigned char giatri;        /* Bien trung gian */
void main (void)            /* Chuong trinh chinh */
{
    EA=1; /* cho phep su dung ngat*/
    EX0=1;           /* cho phep su dung ngat ngoai 0 (int0)*/
    IE1=1; //ngat khi tin hieu 1=>>0
    // IE1=0; Ngat khi tin hieu 0 muc 0

    while (1)
    {
        P2 = 0xFF; /* Port 2 nhap */
        giatri = P2; /* nhap gia tri Port 2 vao bien giatri */
        P1 = giatri; /* xuat gia tri vao Port 1 */
    }
}

void External_Int0(void) interrupt 0 using 2
{
    unsigned int i,j;
    P1=0;
    for(i=0;i<25555;i++)
    {
        j++;
    }
    P1=0xff;
}
```

## 6.5 Thiết kế chương trình sử dụng ngắt timer

```
#include "regx51.h"          /* khai bao thu vien AT89c51 */
unsigned char giatri;      /* Bien trung gian */
void main (void)          /* Chuong trinh chinh */
{
    EA=1;                /* cho phép sử dụng ngắt*/
    ET0=1;               /* cho phép sử dụng ngắt timer 0*/
    TMOD &= 0xF0; // Xoa 4 bit thap của thanh ghi TMOD
    TMOD |= 0x01; // cho bit M0 =1
    TF0 = 0;            // xoa co tran
    TH0=0xD8;
    TL0=0xEF;
    TR0=1;              // khoi dong timer
    while (1)          {
    }
}
void Timer0(void) interrupt 1 using 2
{
    TR0=0;              // stop timer
    TH0=0xD8;
    TL0=0xEF;
    if (P1==255)
        P1=0;
    else
        P1=255;
    TR0=1;              // khoi dong timer
}
```

## 6.6 Thiết kế chương trình sử dụng ngắt Serial

```
#include "regx51.h"          /* khai bao thu vien AT89c51 */
unsigned char giatri,x,c;    /* Bien trung gian */
void main (void)            /* Chuong trinh chinh */
{
    EA=1;                   /* cho phep su dung ngat*/
    ES=1;                   /* cho phep su dung ngat serial*/
    SCON = 0x50;            /* SCON: mode 1, 8-bit UART, enable rcvr */
    TMOD |= 0x20;          /* TMOD: timer 1, mode 2, 8-bit reload */
    TH1 = 231;             /* TH1: reload value for 1200 baud @ 11.059MHz */
    TR1 = 1;               /* TR1: timer 1 run */
    TI = 1;                /* TI: set TI to send first char of UART */
    while (1)
    {
    }
}
void Timer0(void) interrupt 4 using 2
{
    c = SBUF;                // read character
    RI = 0;                  // clear interrupt
    request flag
    TI=0;
    x=c-0x30;
    P1=x;
}
```