单片机中断系统

信息工程学院　　陈梅

STC89C52单片机中断系统 要点

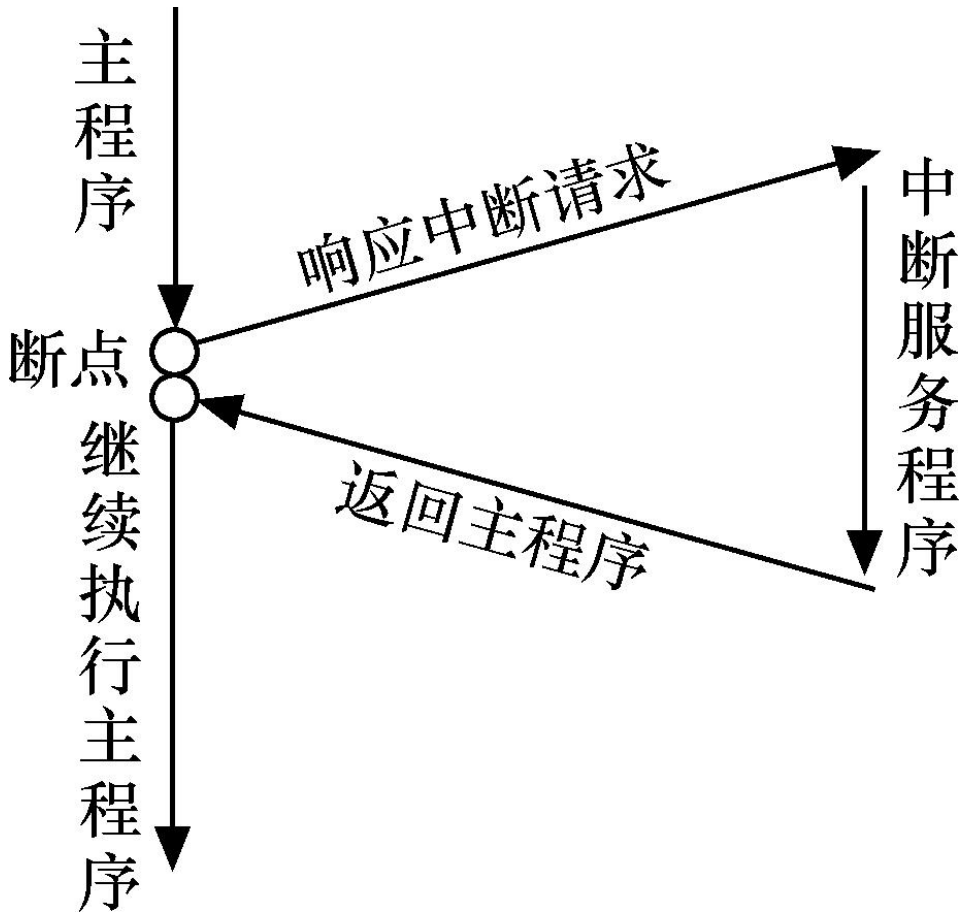
要点：

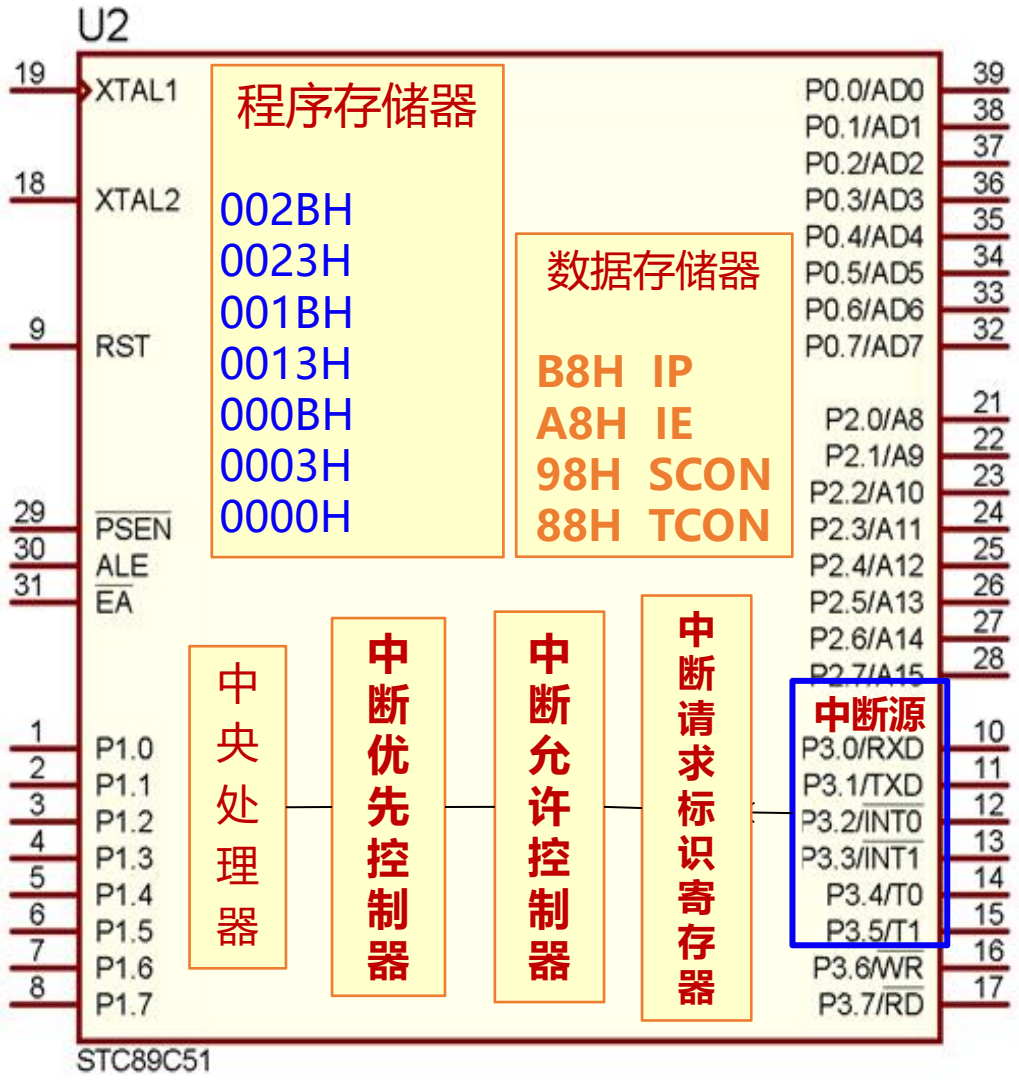
- 中断的相关概念
- STC89C52单片机中断系统的组成
- 中断相关的特殊功能寄存器各位功能和作用
- 中断响应的硬件处理过程、中断响应的条件、
- 外部中断响应时间、中断请求撤销的方法

- 中断服务子程序设计要考虑的几个问题
- 采用中断时的主程序结构、中断服务子程序的流程
- 边沿触发的外部中断仿真示例

中断过程

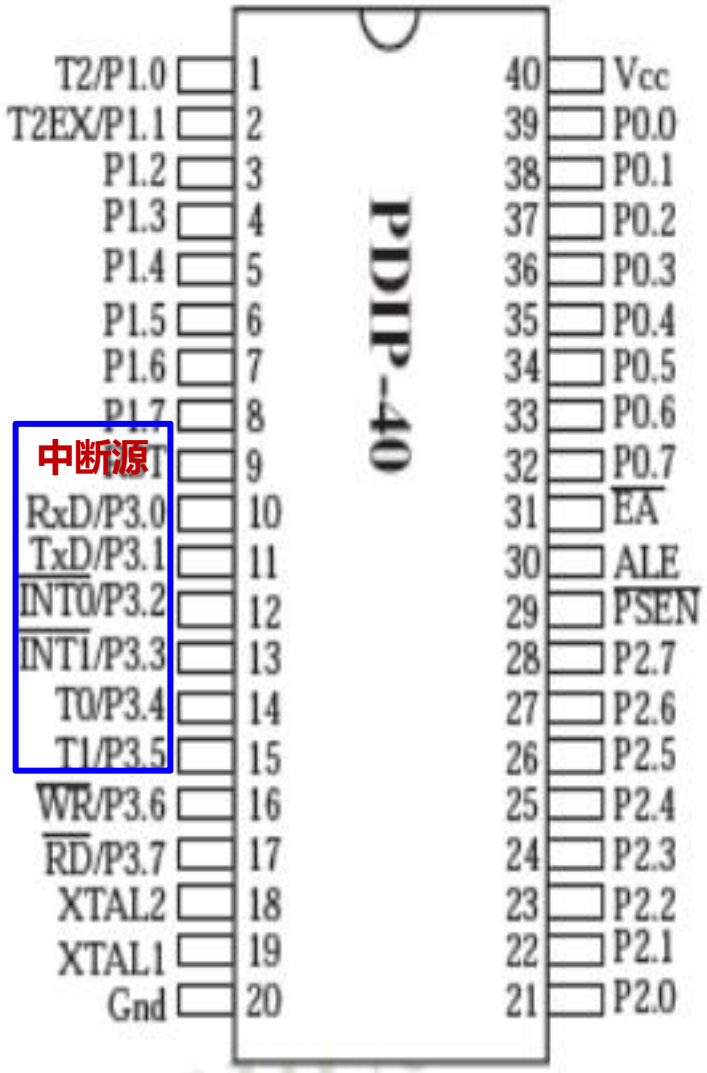
当机器正在执行程序的过程中，一旦遇到一些异常或特殊请求时，就停止正在执行的程序，而转入必要的处理，并在处理完毕后，立即返回断点继续执行。

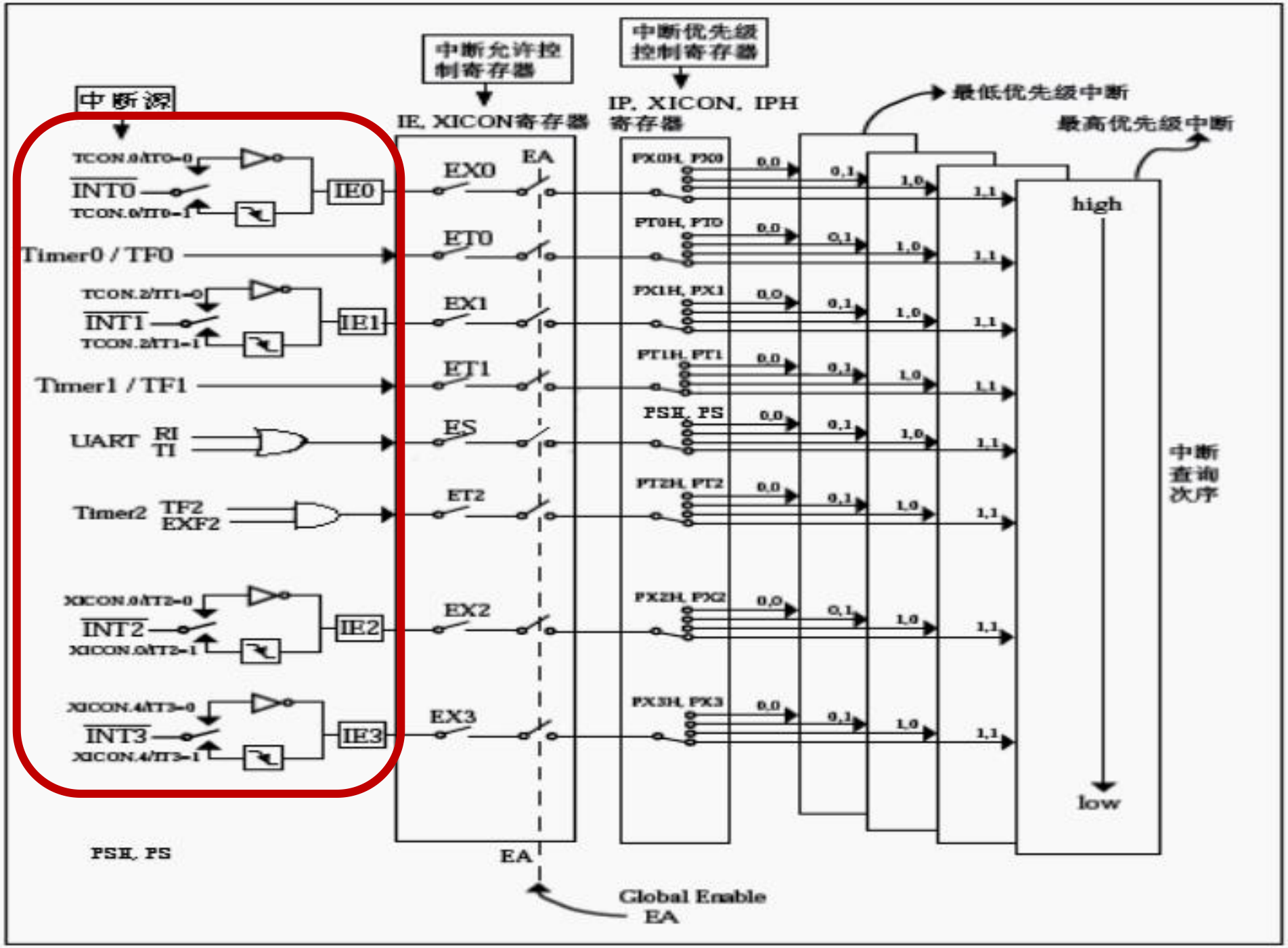


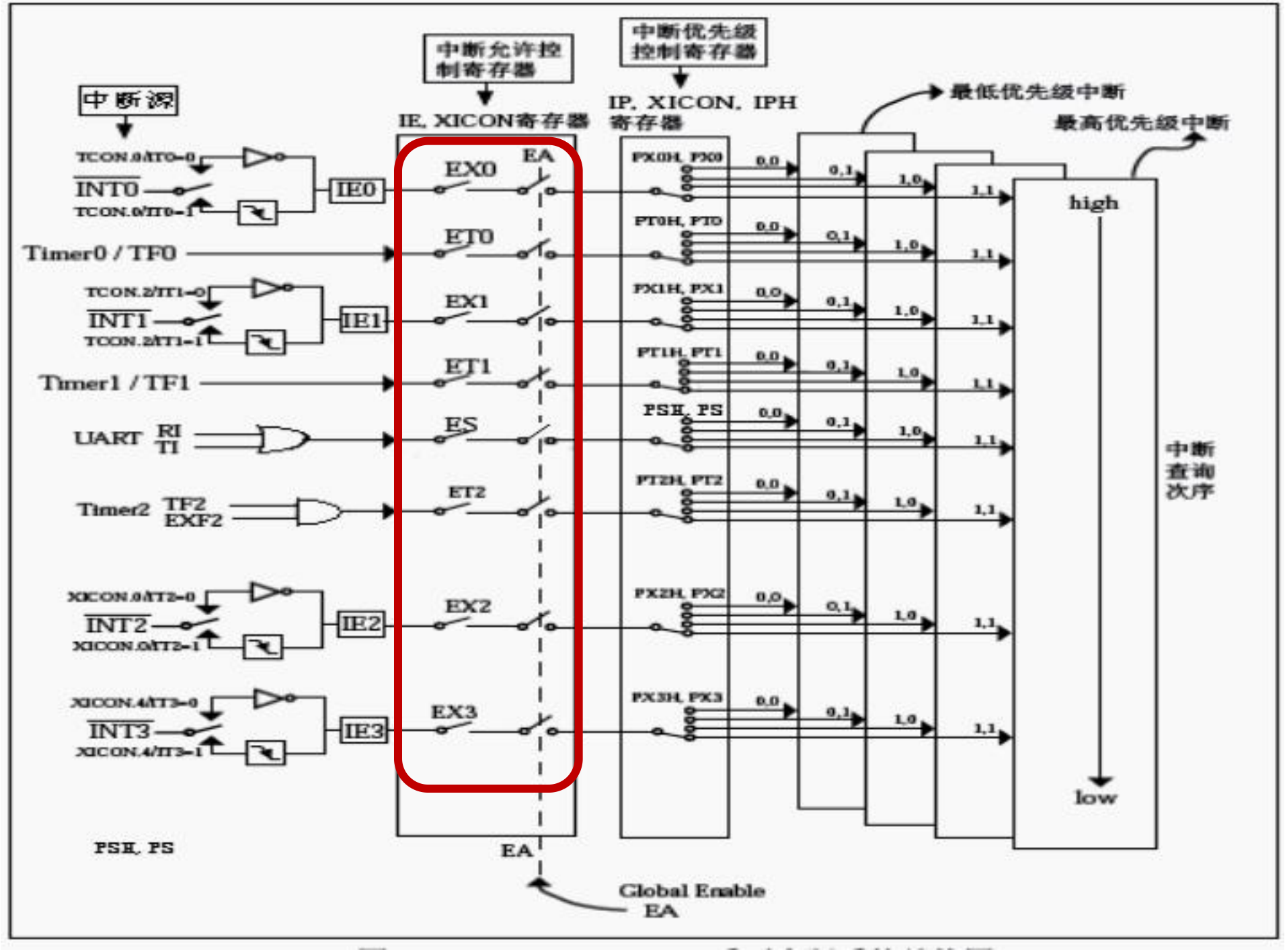


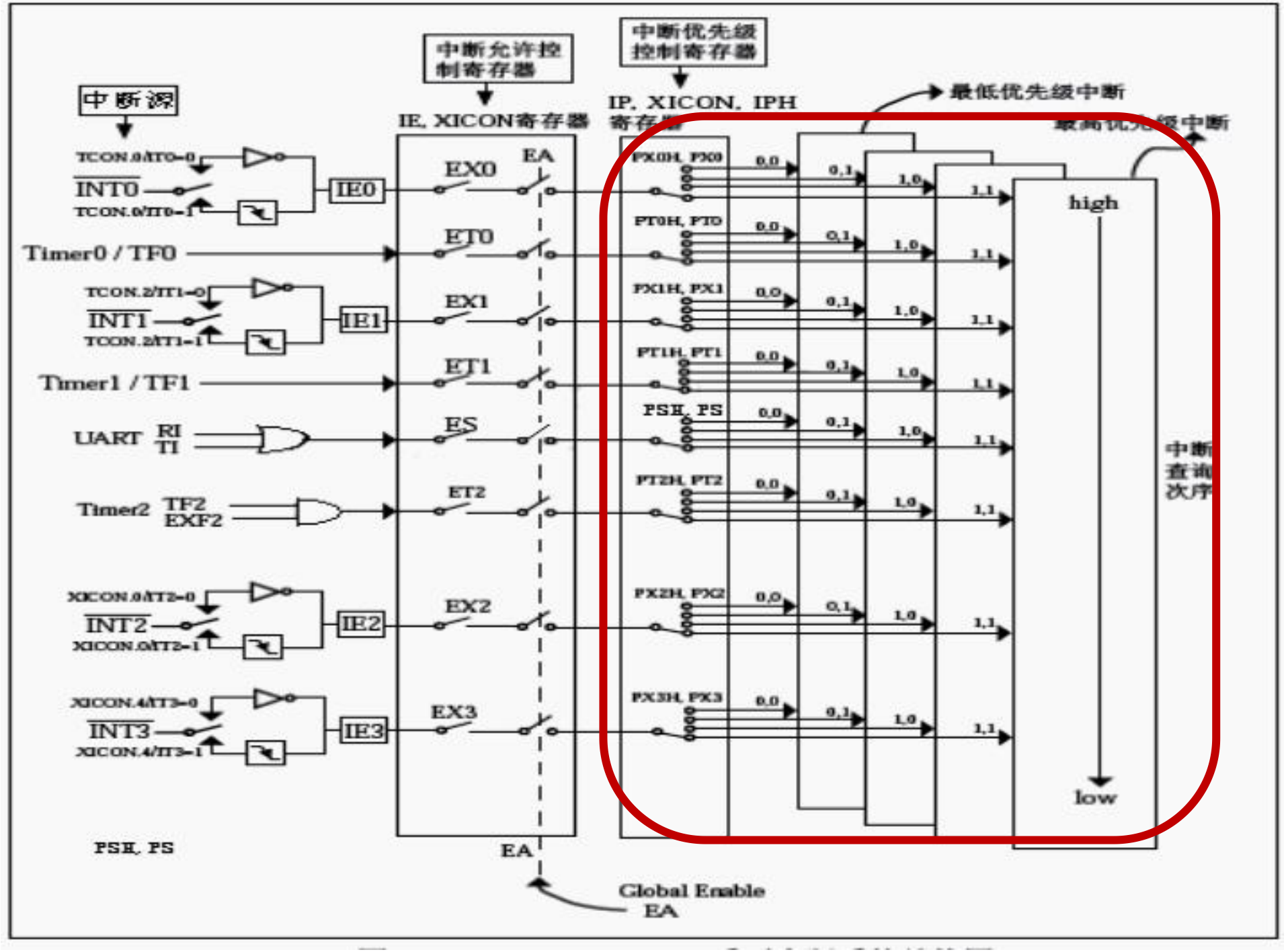
三类中断源：

- 外部中断
  /INT0、/INT1
- 定时中断
  T0、T1
- 串口中断
  RXD、TXD




中断允许控制寄存器
中断优先级控制寄存器
中断源
IE, XICON寄存器
IP, XICON, IPH寄存器
最低优先级中断
最高优先级中断
high
low
中断查询次序
EX0
EA
ET0
EX1
ET1
ES
ET2
EX2
EX3
IE0
IE1
IE2
IE3
Timer0 / TF0
Timer1 / TF1
UART RI TI
Timer2 TF2 EXF2
INT0
INT1
INT2
INT3
PSH, PS
Global Enable EA


中断源
中断允许控制寄存器
IE, XICON寄存器
中断优先级控制寄存器
IP, XICON, IPH寄存器
最低优先级中断
最高优先级中断
high
low
中断查询次序
EA
IE0
IE1
IE2
IE3
EX0
ET0
EX1
ET1
ES
ET2
EX2
EX3
Timer0 / TF0
Timer1 / TF1
UART RI TI
Timer2 TF2 EXF2
INT0
INT1
INT2
INT3
PSH, PS
Global Enable EA


中断源
中断允许控制寄存器
IE, XICON寄存器
中断优先级控制寄存器
IP, XICON, IPH寄存器
最低优先级中断
最高优先级中断
INT0
Timer0 / TF0
INT1
Timer1 / TF1
UART RI TI
Timer2 TF2 EXF2
INT2
INT3
IE0
IE1
IE2
IE3
EX0
ET0
EX1
ET1
ES
ET2
EX2
EX3
EA
PSH, PS
high
low
中断查询次序
Global Enable EA

- TCON（定时/计数器的控制寄存器）88H 可位寻址

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

- SCON（串行口控制寄存器）98H 可位寻址

SCON							TI	RI

中断请求标识寄存器

- IE 中断允许控制寄存器 A8H 可位寻址

IE	EA	–	ET2	ES	ET1	EX1	ET0	EX0

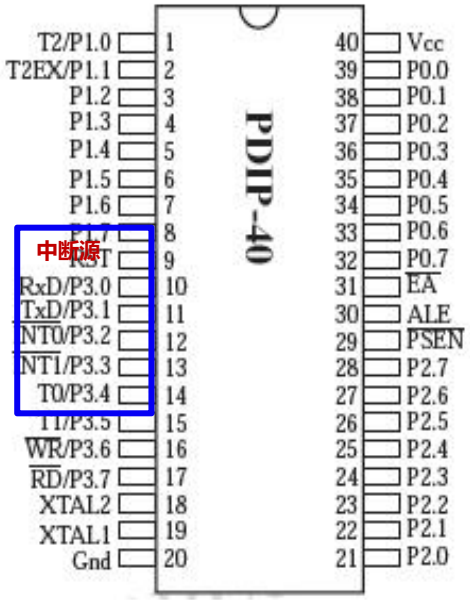
中断允许控制寄存器

- **中断优先级寄存器IP、IPH、附加控制寄存器XICON**

IP	可位寻址	–	–	PT2	PS	PT1	PX1	PT0	PX0
IPH	不可位寻址	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H
XICON	可位寻址	PX3	–	–	–	PX2	–	–	–

中断优先控制寄存器

外部中断相关寄存器–只有外部中断需要设置触发方式



➢ TCON（定时/计数器的控制寄存器）88H 可位寻址

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IT0、IT1: 外部中断的触发方式控制位。

- **IT0或IT1=0，电平触发方式；（默认）**
- **IT0或IT1=1，跳沿触发方式。（由程序设置）**

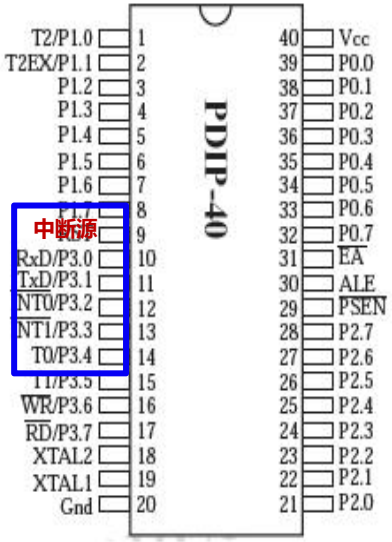
汇编语句：SETB IT0 或 SETB IT1

C51语句： IT0=1; 或 IT1=1;

IE0、IE1: 外部中断的请求标识位。

- **IE0、IE1=0，外部中断INT0、INT1无请求;（默认）**
- **IE0、IE1=1，外部中断INT0、INT1有请求;（由硬件自动置位）**

定时中断、串口中断请求标志寄存器



➢ TCON（定时/计数器的控制寄存器）88H 可位寻址

TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1、TF0：定时中断请求标识位

- TF0、TF1=0，定时计数器T0、T1溢出无请求;（默认）
- TF0、TF1=1，定时计数器T0、T1溢出有请求;（由硬件自动置位）

➢ SCON（串行口控制寄存器）98H 可位寻址

SCON							TI	RI

TI、RI：串行发送和中行接收中断请求标识位。

- TI=0、RI=0，无发送、无接收中断请求（默认）
- TI=1、RI=1，串口发送、接收完一帧数据后请求中断；（由硬件自动置位）

IE 中断允许控制寄存器 A8H 可位寻址

	D7	D6	D5	D4	D3	D2	D1	D0
IE	EA	—	ET2	ES	ET1	EX1	ET0	EX0

IE对中断的开放和关闭实现**两级控制：总开关&分开关**

- 总开关**EA=0**屏蔽所有中断（默认）；**EA=1**开放所有中断（由程序设置）

  SETB EA 或 EA=1

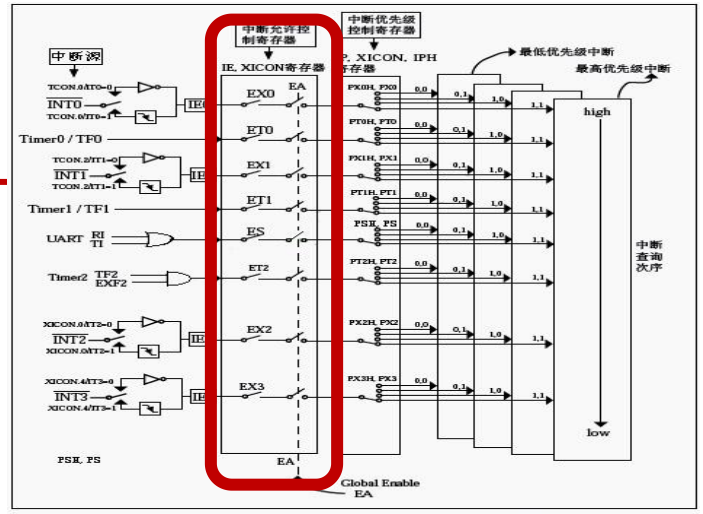
- 分开关：**=0**禁止相应中断（默认）；**=1**开放相应中断（由程序设置）

  ES—串行口中断允许位 **SETB ES 或 ES=1**

  EX1/EX0—外部中断1和0中断允许位 **SETB EXi 或 EXi=1**

  ET2/ET1/ET0—定时器/计数器溢出中断允许位 **SETB ETi 或 ETi=1**

请问下列两组代码的区别？

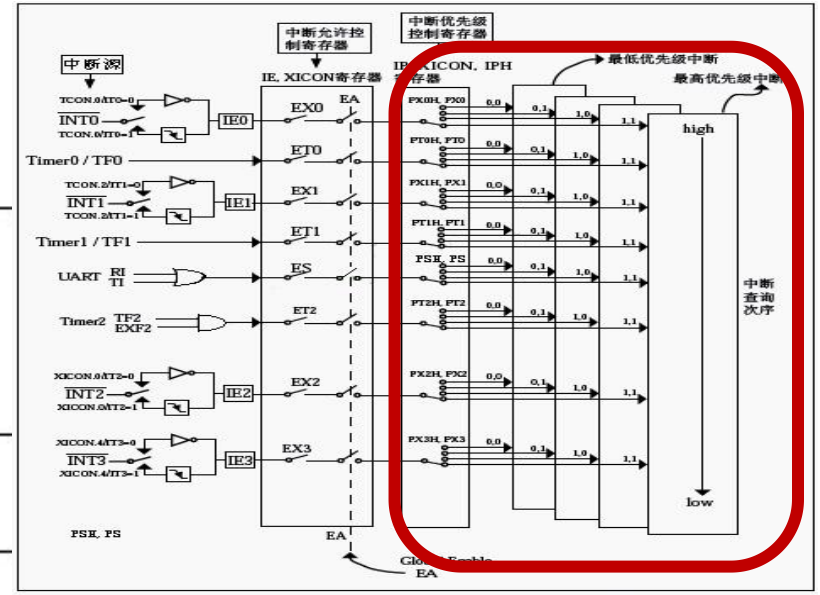


MOV	IE,#0AAH	或	IE=0xAA;

SETB	EA		EA=1;
SETB	ET0		ET0=1;
SETB	ET1		ET1=1;
SETB	ET2	或	ET2=1;
CLR	ES		ES=0;
CLR	EX0		EX0=0;
CLR	EX1		EX1=0;

(IE)=1010 1010，即打开定时器0、1、2的中断，屏蔽了外部0、1中断和串行口中断；

中断源	中断向量地址	中断查询次序	中断优先级设置	优先级0最低	优先级1	优先级2	优先级3最高	
$\overline{INT0}$	0003H	0(最优先)	PX0H PX0	0 0	0 1	1 0	1 1	
Timer 0	000BH	1	PT0H PT0	0 0	0 1	1 0	1 1	TF0
$\overline{INT1}$	0013H	2	PX1H PX1	0 0	0 1	1 0	1 1	IE1
Timer 1	001BH	3	PT1H PT1	0 0	0 1	1 0	1 1	TF1
UART	0023H	4	PSH PS	0 0	0 1	1 0	1 1	RI + TI
Timer 2	002BH	5	PT2H PT2	0 0	0 1	1 0	1 1	TF2 + EXF2
$\overline{INT2}$	0033H	6	PX2H PX2	0 0	0 1	1 0	1 1	IE2
$\overline{INT3}$	003BH	7(最低)	PX3H PX3	0 0	0 1	1 0	1 1	IE3



中断优先级寄存器IP、IPH——默认全0为最低优先级

IP	可位寻址	—	—	PT2	PS	PT1	PX1	PT0	PX0
IPH	不可位寻址	PX3H	PX2H	PT2H	PSH	PT1H	PX1H	PT0H	PX0H

中断源	中断优先级设置	优先级0 最低	优先级1	优先级2	优先级3 最高
$\overline{INT0}$	PX0H PX0	0 0	0 1	1 0	1 1
Timer 0	PT0H PT0	0 0	0 1	1 0	1 1
$\overline{INT1}$	PX1H PX1	0 0	0 1	1 0	1 1
Timer 1	PT1H PT1	0 0	0 1	1 0	1 1
UART	PSH PS	0 0	0 1	1 0	1 1
Timer 2	PT2H PT2	0 0	0 1	1 0	1 1
$\overline{INT2}$	PX2H PX2	0 0	0 1	1 0	1 1
$\overline{INT3}$	PX3H PX3	0 0	0 1	1 0	1 1

```
MOV   IP,#05H     ;即IP=0x05
MOV   IPH,#0FH    ;即 IPH=0x0F
```

根据指令知：

IP = 0000 0101

IPX= 0000 1111

PX0H PX0=11即INT0为最高优先级3级

PX1H PX1=11即INT1为最高优先级3级

PT0H PT0=10即T0为优先级2级

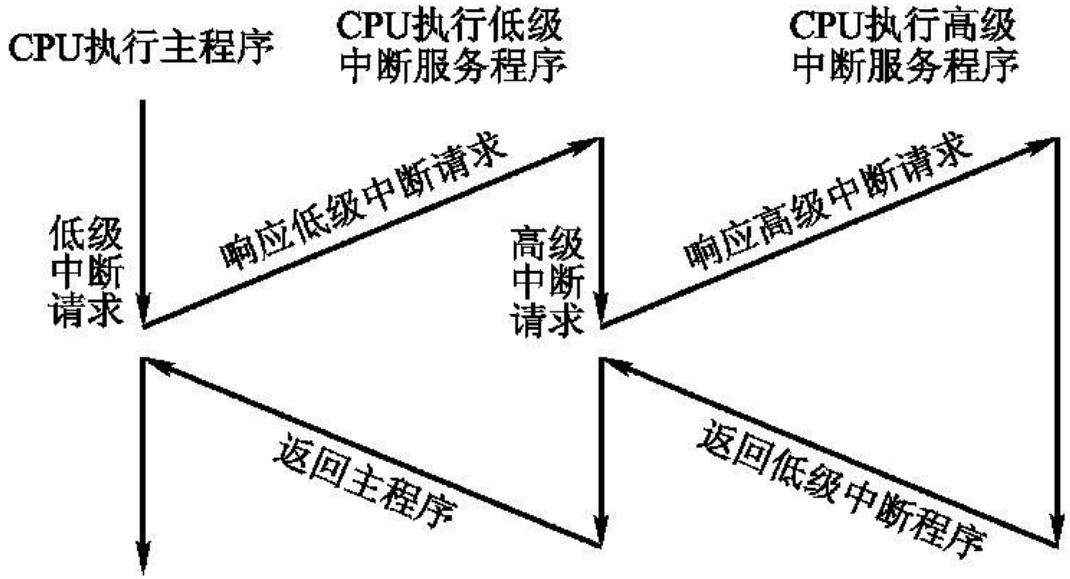
PT1H PT1=10即T1为优先级2级

中断源优先级关系的基本规则

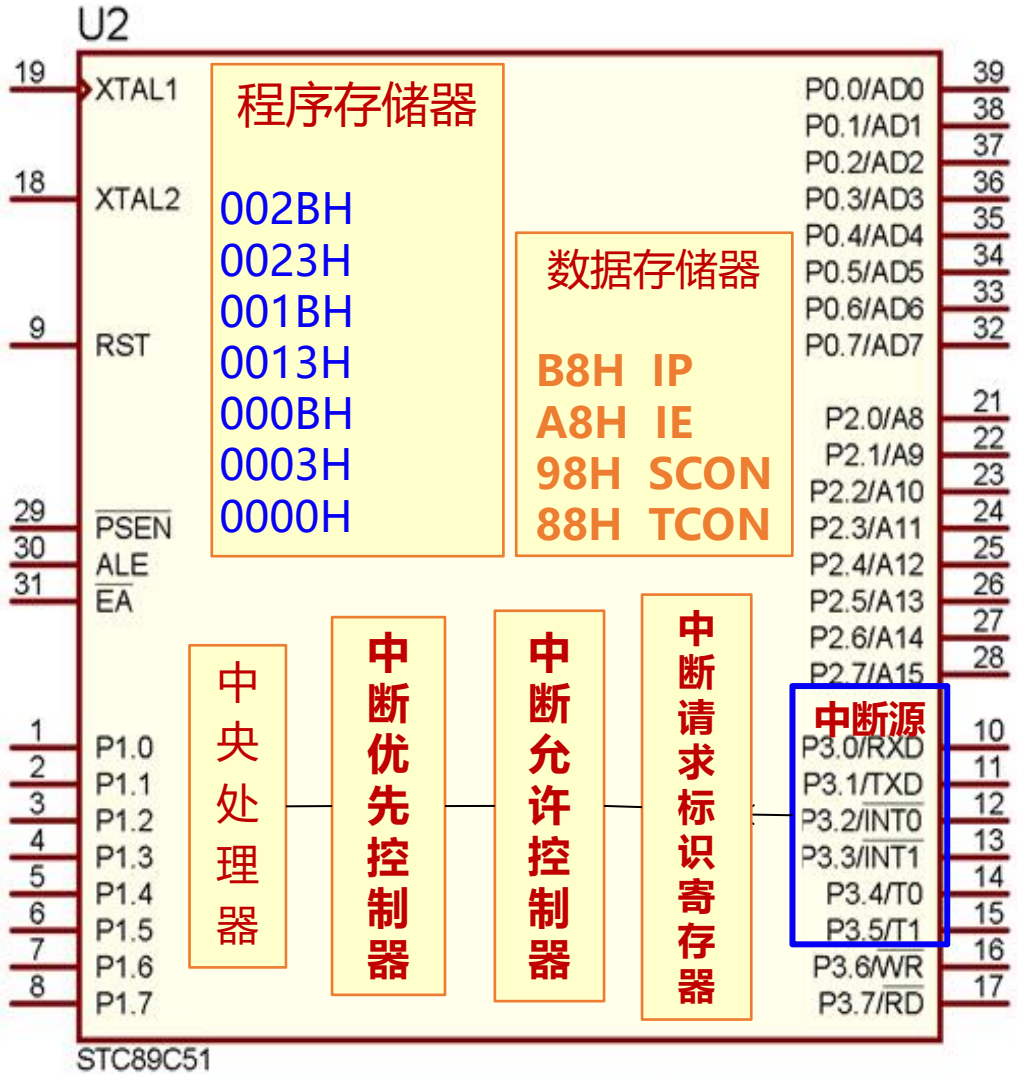
（1）低优先级可被高优先级中断，高优先级不能被低优先级中断。如果某一中断源被设置为高优先级中断，则不能被任何其他的中断源的中断请求所中断。

（2）任何一种中断一旦得到响应，不会再被它的同级中断源所中断。

（3）当两个相同优先级中断同时产生时，将由查询次序来决定系统先响应哪个中断。



中断请求的必要条件



响应中断请求的必要条件:

1) 中断源有请求：TCON或SCON对应请求标志位为1
2) 中断开放：IE中的EA位为1开放总中断，
   IE中对应中断源允许位为1开某中断源
3) 无同级或更高级中断正在被服务，即对IP的设置
4) 当前指令执行完毕。

（且当前不是RETI指令或是对IE,IP操作的指令，
因为这两类指令还需要再执行完一条指令才响应中断）

中断请求的撤销——①中断标志位清0②中断信号的撤消

1) 定时计数器中断请求的撤消

该中断响应后,硬件会自动把中断请求标志位(TF0或TF1)清0.

2) 串行口中断请求的撤消

该中断响应后,硬件不会自动清中断请求标志位 (TI或RI),

必须用软件指令清0: CLR TI 或 CLR RI

3) 外部中断请求的撤消

该中断响应后,硬件会自动把中断标志位(IE0或IE1)的清0。

- 跳沿方式——由于跳沿信号过后就消失，所以请求信号也是自动撤消。
- 电平方式——中断请求信号的低电平可能继续存在,将会重新把清0的标位置位.所以还必须外接撤消电路和一段软件代码配合撤消.

C51程序结构——以外部中断1为例

```
#include<reg52.h>
void  main()
{
        ……;  //设置SP堆栈指针
        ……;  //设置IT外部中断触发方式
        ……;  //设置IE开放总中断
        ……;  //设置IP和IPH中断优先级
        while(1){ ;}    //踏步等待中断
 }

void exint1(void) interrupt  2  using 0
 {
        ……;  //外部中断1服务函数
   }
```

中断号	中断源
0	INT0外部中断
1	T0定时计数中断
2	INT1外部中断
3	T1定时计数中断
4	串口中断
5	T2定时计数中断

响应中断服务的执行过程

- 主程序在等待中断发生时由硬件自动生成一条LCALL指令，将断点地址压入堆栈保护，然后找到相应中断源的入口跳转到相应的服务程序即响应中断请求。

汇编程序结构——以外部中断1为例

```
        ORG    0000H
        LJMP   MAIN
        ORG    0013H     ；0013H为外部中断1的入口地址
        LJMP   INT
        ORG    0100H     ；主程序入口
MAIN：  ……;    //设置SP堆栈指针
        ……;    //设置IT外部中断触发方式
        ……;    //设置IE开放总中断
        ……;    //设置IP和IPH中断优先级
        SJMP $
INT：   ……                ；中断服务子程序
        RETI
        END
```

中断源	向量地址
$\overline{\text{INT0}}$	0003H
Timer 0	000BH
$\overline{\text{INT1}}$	0013H
Timer 1	001BH
UART	0023H
Timer 2	002BH
$\overline{\text{INT2}}$	0033H
$\overline{\text{INT3}}$	003BH

中断返回的执行过程

- 当中断服务程序的运行结束时，必须将中断响应时压栈保护的断点值弹出恢复，返回断点处继续，并清除中断响应时被设置的中断优先级状态。

子程序结构与中断服务程序结构的区别?

- **中断返回指令　　RETI**

  执行断点保护的操作与RET指令相同，

  不同之处：该指令同时还清除被置1的中断优先级状态

```
        ORG    0000H
        LJMP   MAIN
        ORG    X1X2X3X4H ；中断入口地址
        LJMP   INT

        ORG    Y1Y2Y3Y4H ；主程序
MAIN：  ……
        SJMP $

INT：   ……              ；中断服务程序
        RETI
        END
```

- 子程序返回指令　RET

  (SP)→PCH，(SP)-1→SP；

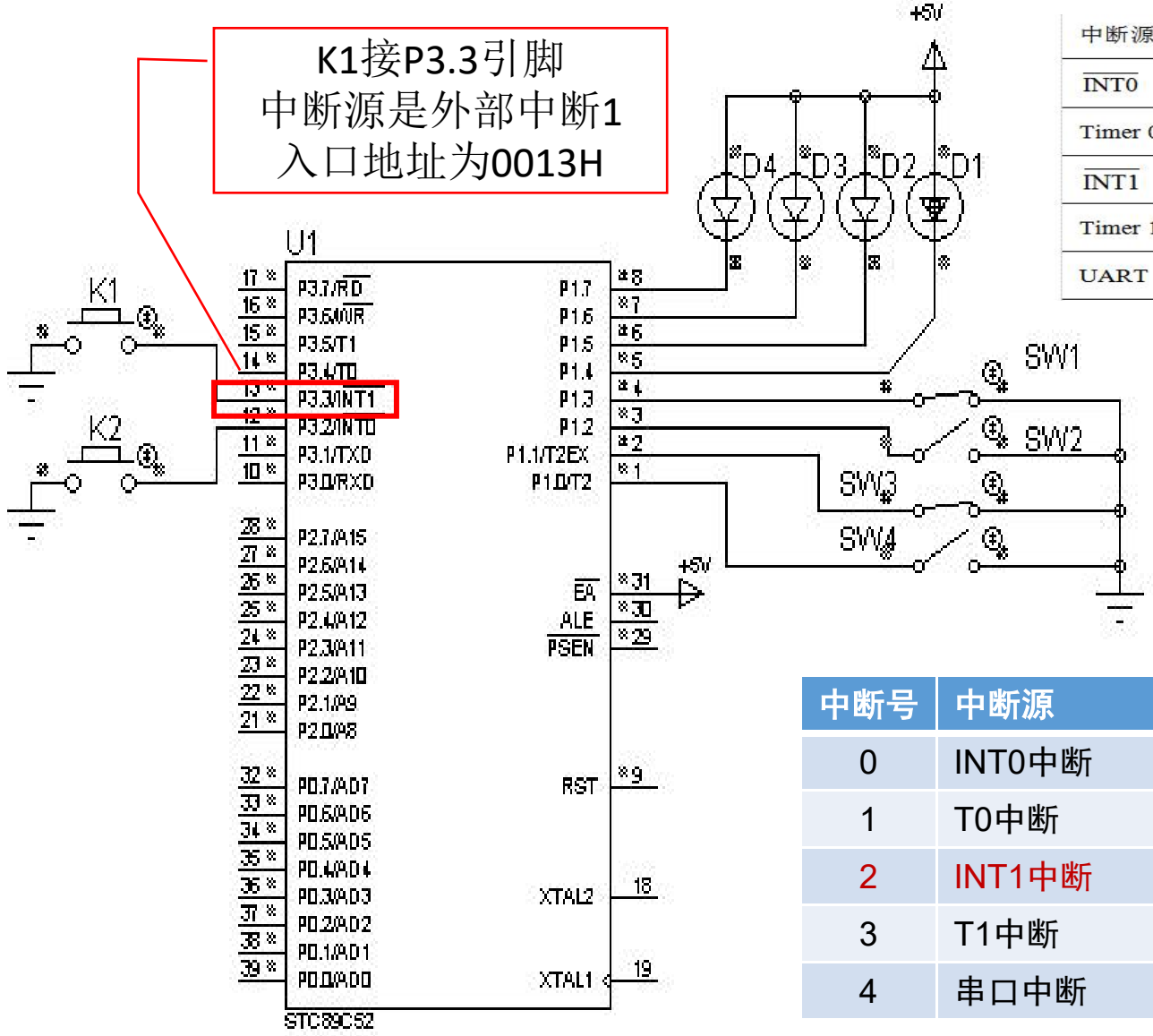
  (SP)→PCL，(SP)-1→SP；

```
        ORG    0000H
        LJMP   MAIN


        ORG    Y1Y2Y3Y4H ；主程序
MAIN:   ……
        LCALL DELAY
        ……

DELAY:  ……
        RET
        END
```

【例5-3】外部中断1为边沿触发的外部中断源，当按下P3.3所接的K1产生中断请求单片机读取P1.0-P1.3引脚状态，将采样到信号转换为输出信号去驱动P1.4-P1.7的发光二极管的亮灭，单片机的工作频率为11.0592MHz，编写驱动程序。

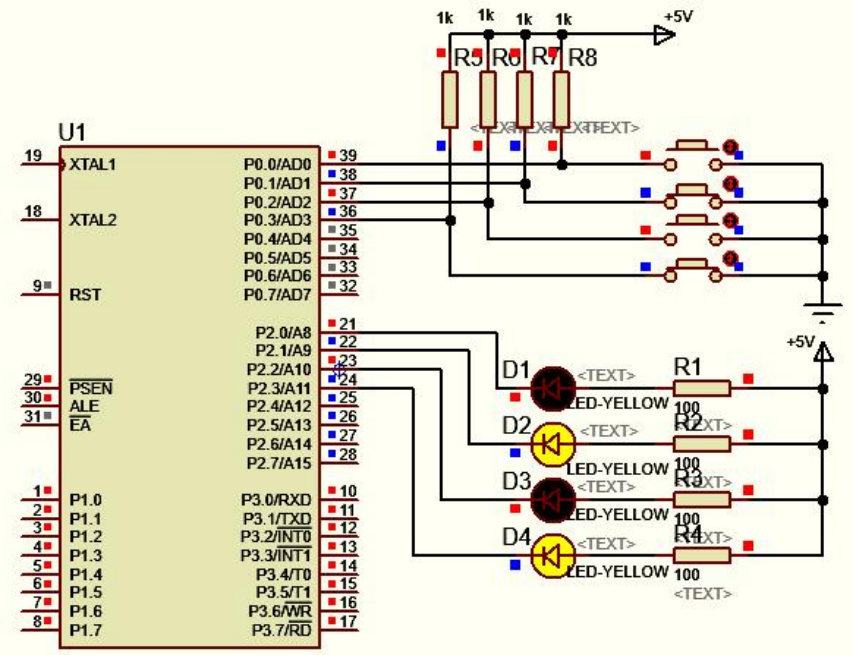


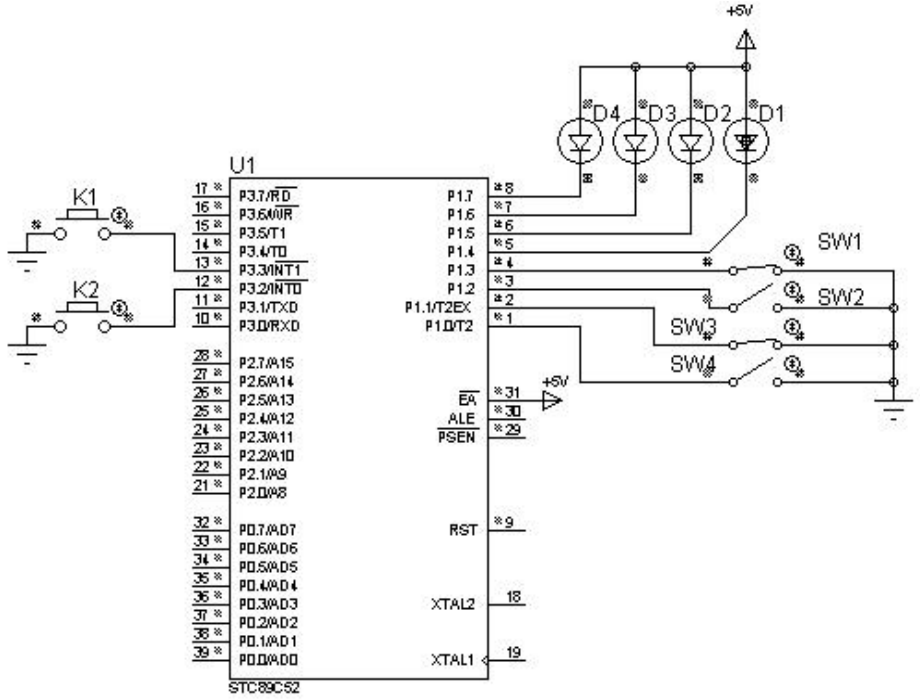
中断源	向量地址
$\overline{INT0}$	0003H
Timer 0	000BH
$\overline{INT1}$	0013H
Timer 1	001BH
UART	0023H

中断号	中断源
0	INT0中断
1	T0中断
2	INT1中断
3	T1中断
4	串口中断

```
        ORG     0000H
        LJMP    MAIN
        ORG     0013H  ；中断入口地址
        LJMP    INT
        ORG     0100H ；主程序
MAIN：  ……
        SJMP $
INT：   ……              ；中断服务程序
        RETI
        END
```

```
#include<reg52.h>
void  main()
{
        ……;
        while(1){ ;}
}
void exint1(void) interrupt  2
{
        ……;     //中断服务函数
  }
```






```
        ORG   0000H
        MOV   P2,#0FFH
LOOP:   MOV   P0,#0FFH
        MOV   A,P0
        MOV   P2,A
        LJMP  LOOP
        END
```

```
INT1:   MOV   P1,#0FFH
        MOV   A,P1
        SWAP  A
        MOV   P1,A
        RETI
```

主程序任务：

- 设置堆栈指针：　　MOV　SP,#60H
- 设置外部中断边沿触发：SETB　IT1
- 开放外部中断1：　　SETB　EX1
- 开总中断：　　SET　 EA
- 等待中断：　　SJMP　$

中断服务程序任务：

- 读取P1口低4位值：　　MOV　P1,#0FFH
  　　MOV　A,P1
- 高、低4位交换：　　SWAP　 A
- P1高4位输出驱动LED：MOV　P1，A

```
        ORG    0000H
        LJMP   MAIN
        ORG    0013H ；中断入口地址
        LJMP   INT

        ORG    0100H；主程序
MAIN：  ……;    //设置SP堆栈指针
        ……;    //设置IT外部中断触发方式
        ……;    //设置IE开放总中断
        ……;    //设置IP和IPH中断优先
        SJMP $

INT：   ……              ；中断服务程序
        RETI
        END
```

主程序任务：

- 设置堆栈指针：　　MOV　SP,#60H
- 设置外部中断边沿触发：SETB　IT1
- 开放外部中断1：　　SETB　EX1
- 开总中断：　　SET　 EA
- 等待中断：　　SJMP　$

中断服务程序任务：

- 读取P1口低4位值：　　MOV　P1,#0FFH
  MOV　A,P1
- 高、低4位交换：　　SWAP　 A
- P1高4位输出驱动LED：MOV　P1，A

```
        ORG   0000H
        LJMP  MAIN
        ORG   0013H        ;入口地址
        LJMP  INT1

        ORG   0100H
MAIN:   MOV   SP,#60H
        SETB  IT1
        SETB  EX1
        SETB  EA
        SJMP  $

INT1:   MOV    P1,#0FFH
        MOV    A,P1
        SWAP   A
        MOV    P1,A
        RETI
        END
```

```c
#include<reg52.h>
#define uchar unsigned char
void  main()
{
        SP=0X60;
        IT1=1;
        EX1=1;
        EA=1;
        while(1){ ; }
 }

void exint1(void) interrupt  2
{   uchar p1_Value=0;
    P1=0xff;
    p1_Value=P1 & 0x0f;
    p1_Value=p1_Value<<4;
    P1=p1_Value;
}
```

```
        ORG   0000H
        LJMP  MAIN
        ORG   0013H       ;入口地址
        LJMP  INT1

        ORG   0100H
MAIN:   MOV   SP,#60H
        SETB  IT1
        SETB  EX1
        SETB  EA
        SJMP  $

INT1:   MOV    P1,#0FFH
        MOV    A,P1
        SWAP   A
        MOV    P1,A
        RETI
        END
```

第5讲　中断系统要点

重点

- STC89C52单片机中断系统的组成
- 中断相关的特殊功能寄存器各位功能和作用
- 中断响应的处理过程、中断请求响应的条件、中断请求撤销的方法

- 采用中断时的主程序结构、
- 中断服务子程序的执行过程