第五章 输入输出和中断系统

信息科学与工程学院自动化系

5.4 AT89C52的中断系统

- AT89C52 的中断系统的结构

5．4．1 AT89C52的中断源和中断标志

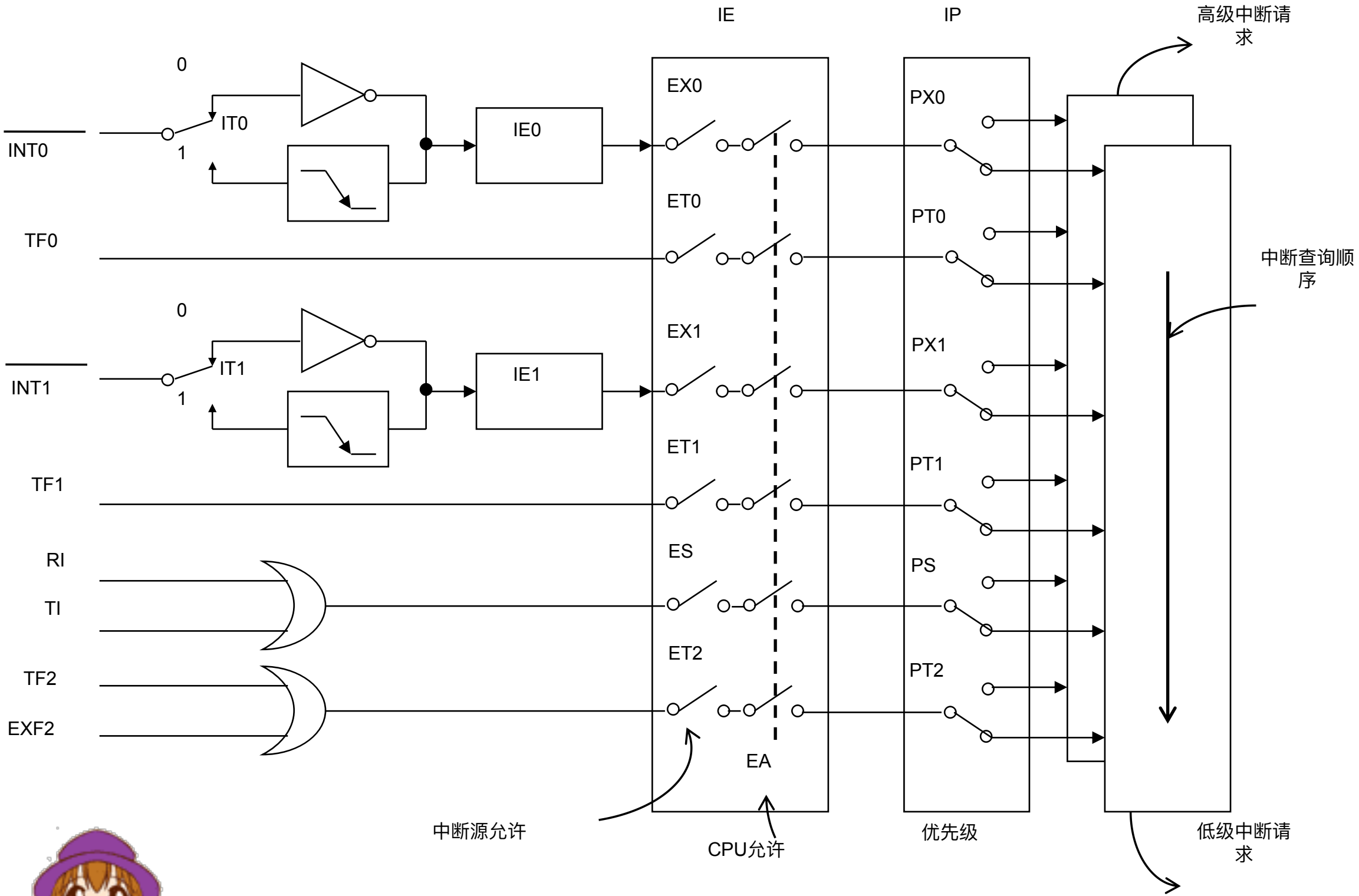
5．4．2 AT89C52对中断请求的控制

5．4．3 AT89C52对中断的响应和撤除

5．4．4 AT89C52中断系统的初始化及应用

- 中断程序设计举例

AT89C52 的中断系统的结构



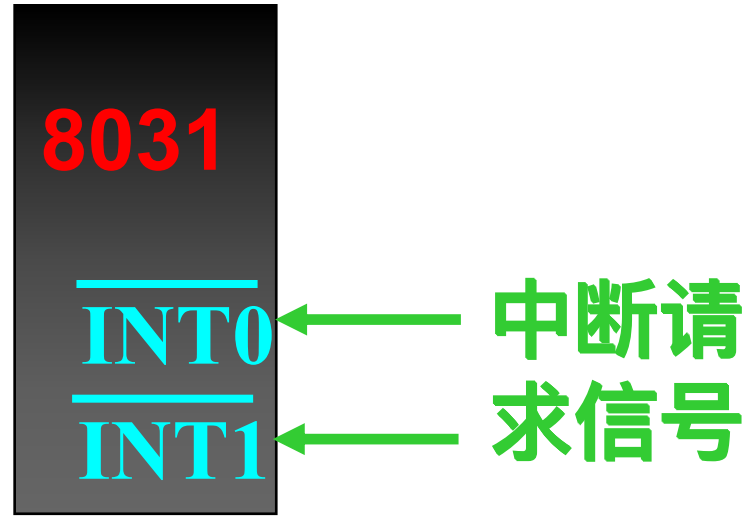
5.4.1 AT89C52的中断源和中断标志

AT89C52 共有6个中断源

- 外部中断0
- 外部中断1
- T0溢出中断
- T1溢出中断
- T2溢出中断
- 串行口中断

2个外部中断请求；
3个片内定时器/计数器T0、T1和T2中断请求；
1个串行口中断请求。

外部中断请求0、1



- 中断请求信号输入引脚：

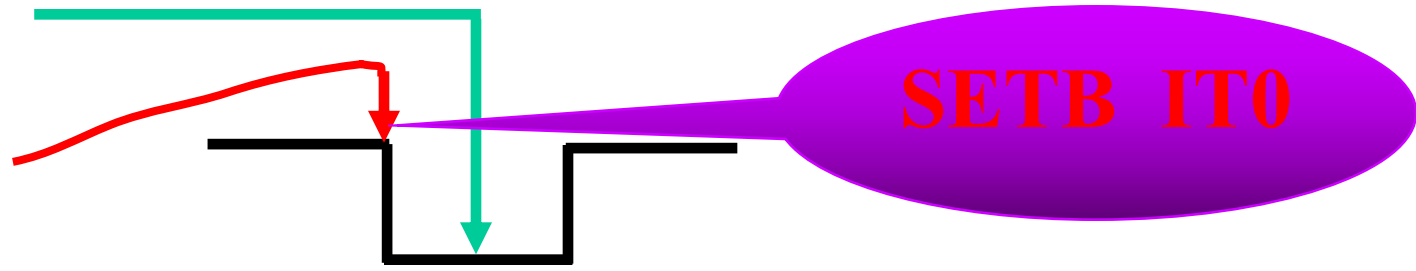
外部中断0请求引脚：$\overline{\text{INT0}}$(P3.2)

外部中断1请求引脚：$\overline{\text{INT1}}$(P3.3)

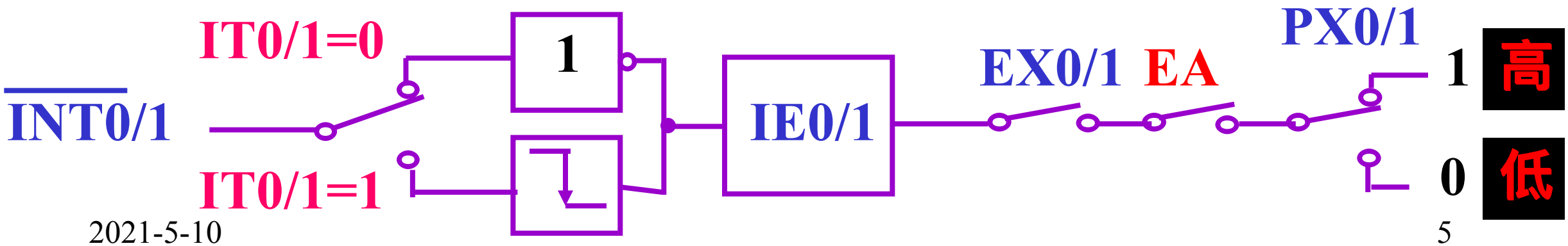
- 外部中断请求信号触发方式选择（IT0/1位）

电平触发：低电平

边沿触发：负边沿



- 外部中断请求标志位、中断允许位、优先级选择位：



定时器/计数器控制寄存器TCON（也锁存外部中断请求标志）

TCON (88H)

D7 8FH		8DH		8BH	8AH	89H	D0 88H
TF1		TF0		IE1	IT1	IE0	IT0

- IT0（IT1）：外部中断请求$\overline{INT0}$($\overline{INT1}$)的触发方式选择
  - IT0=0 电平触发方式，低电平有效；
  - IT0=1 边沿触发方式，高到低的负跳变有效
- IE0(IE1)：外部中断请求0(1)的中断申请(请求)标志位
  - IT0=0 CPU每个机器周期采样/INT0，若/INT0=1 则IE0=0 否则/INT0=0， IE0=1申请中断
  - IT0=1 若第一个机器周期/INT0=1，第二个机器周期/INT0=0，则IE0=1申请中断，否则IE0=0
- 转向中断服务时边沿触发方式下IE由硬件清零
- 转向中断服务时电平触发方式下硬件不清IE，待申请信号撤消。

T0、T1溢出中断

TCON （88H）

D7 8FH		8DH		8BH	8AH	89H	D0 88H
TF1		TF0		IE1	IT1	IE0	IT0

- TF0：片内定时器/计数器0溢出中断请求标志。
  - 定时/计数器0产生溢出中断（全“1”变全“0”）时，TF0由硬件置位（置“1”），向CPU申请中断；
  - 当CPU 响应中断时，由硬件自动清零TF0，中断申请撤除；
  - TF0 也可软件查询，复位清零。
- TF1： 片内定时器/计数器1溢出中断请求标志

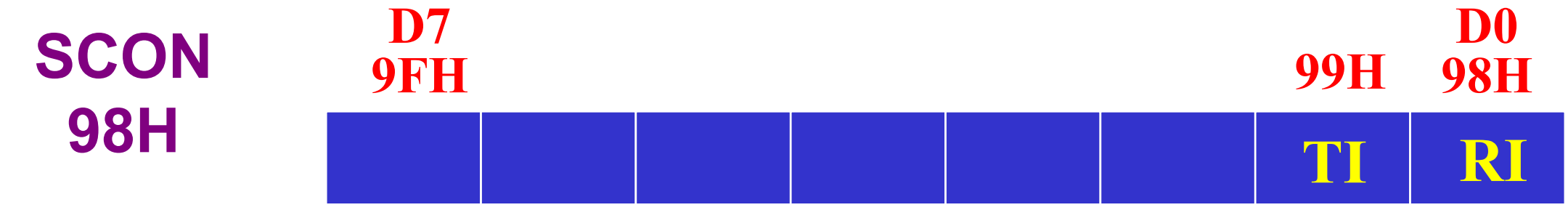
T2CON (C8H)

CFH	CEH	CDH	CCH	CBH	CAH	C9H	C8H
TF2	EXF2						

- TF2： 片内定时/计数器2溢出中断请求标志
- EXF2： 片内定时/计数器2外部中断请求标志

串行口中断

串行口控制寄存器SCON



- RI：串行口**接收**中断标志，RI=1串行口接收中断

当允许串行口接收数据时，每接收完一个串行帧，由硬件置位RI。响应中断后，RI必须由软件清除。

- TI：串行口**发送**中断标志，TI=1串行口发送中断

当发送完一个串行帧，由硬件置位TI。CPU响应中断时不能自动清除TI，TI必须由软件清除。

- RI、TI 由硬件置位

必须由软件清零

5.4.2 AT89C52 对中断请求的控制

	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H
IE (A8H)	EA		ET2	ES	ET1	EX1	ET0	EX0

中断允许控制：对应位=0，不允许中断
对应位=1，允许中断

	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H
IP (B8H)			PT2	PS	PT1	PX1	PT0	PX0

中断优先级控制：对应位=0，低优先级
对应位=1，高优先级

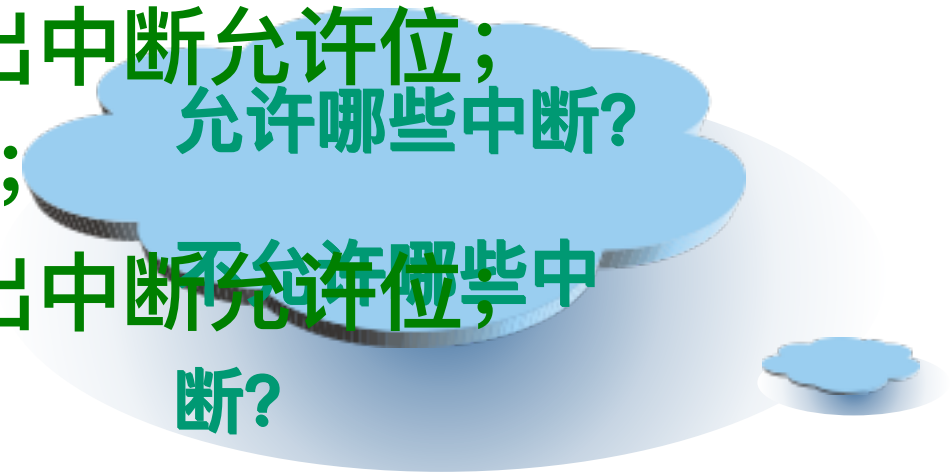
1、对中断允许的控制

- 中断允许控制寄存器IE

IE	D7 AFH	D6	D5 ADH	D4 ACH	D3 ABH	D2 AAH	D1 A9H	D0 A8H
A8H	EA	---	ET2	ES	ET1	EX1	ET0	EX0

1：允许
0：禁止

- EX0(IE.0)：外部中断0允许位；
- EX1(IE.2)：外部中断1允许位；
- ET0(IE.1)：定时/计数器T0溢出中断允许位；
- ET1(IE.3)：定时/计数器T1溢出中断允许位；
- ES (IE.4)：串行口中断允许位；
- ET2(IE.5)：定时/计数器T2溢出中断允许位；



- EA (IE.7)：CPU中断允许总控位=1,开放中断;允许中断=0,禁止任何中断

例1：假如允许两个外部中断源/INT0和/INT1申请中断，如何设置?

方法一

```
MOV    IE，#85H
```

或

```
MOV    0A8H，#85H
```

方法二

```
SETB   EA ；开中断总控位
SETB   EX0；允许INT0中断
SETB   EX1；允许INT1中断
```

IE	D7 EA	D6	D5 ET2	D4 ES	D3 ET1	D2 EX1	D1 ET0	D0 EX0
A8H	1	---				1		1

2、对中断优先级的控制

- 中断优先级控制寄存器 IP

IP	D7	D6	D5	D4	D3	D2	D1	D0
			BDH	BCH	BBH	BAH	B9H	B8H
B8H	-	-	PT2	PS	PT1	PX1	PT0	PX0

1：优先
0：滞后

每个中断源的中断优先级都是由中断优先级寄存器IP中的相应位的状态来规定的 。

- PX0（IP.0）：外部中断0优先级控制位；
- PT0（IP.1）：定时/计数器T/C0优先级控制位；
- PX1（IP.2）：外部中断1优先级控制位；
- PT1（IP.3）：定时/计数器T1优先级控制位；
- PS （IP.4）：串行口中断优先级控制位；
- PT2（IP.5）：定时/计数器T2优先级控制位。

优先级对中断响应的影响：

- CPU同时接收到几个中断时，首先响应优先级别最高的中断请求。
- 正在进行的中断过程不能被新的同级或低优先级的中断请求所中断。
- 正在进行的低优先级中断服务，能被高优先级中断请求所中断。这称之为“中断嵌套”。

总结中断源的各个触发器

中断源	中断标志1/0	中断允许1/0	中断级别1/0
外部$\overline{INT0}$	IE0 (TCON.1)	EX0(IE.0)	PX0 (IP.0)
外部$\overline{INT1}$	IE1(TCON.3)	EX1(IE.2)	PX1 (IP.2)
定时器0	TF0(TCON.5)	ET0(IE.1)	PT0 (IP.1)
定时器1	TF1 (TCON.7)	ET1(IE.3)	PT1 (IP.3)
串行口	RI (SCON.0) TI (SCON.1)	ES (IE.4)	PS (IP.4)
定时器2	TF2 (T2CON.7)	ET2(IE.5)	PT2 (IP.5)
CPU标志		EA (IE.7)	

寄存器：TCON、SCON 、IE 、IP 复位后，这四个寄存器均为00H。

若设置串行口和定时器/计数器1为高级中断：

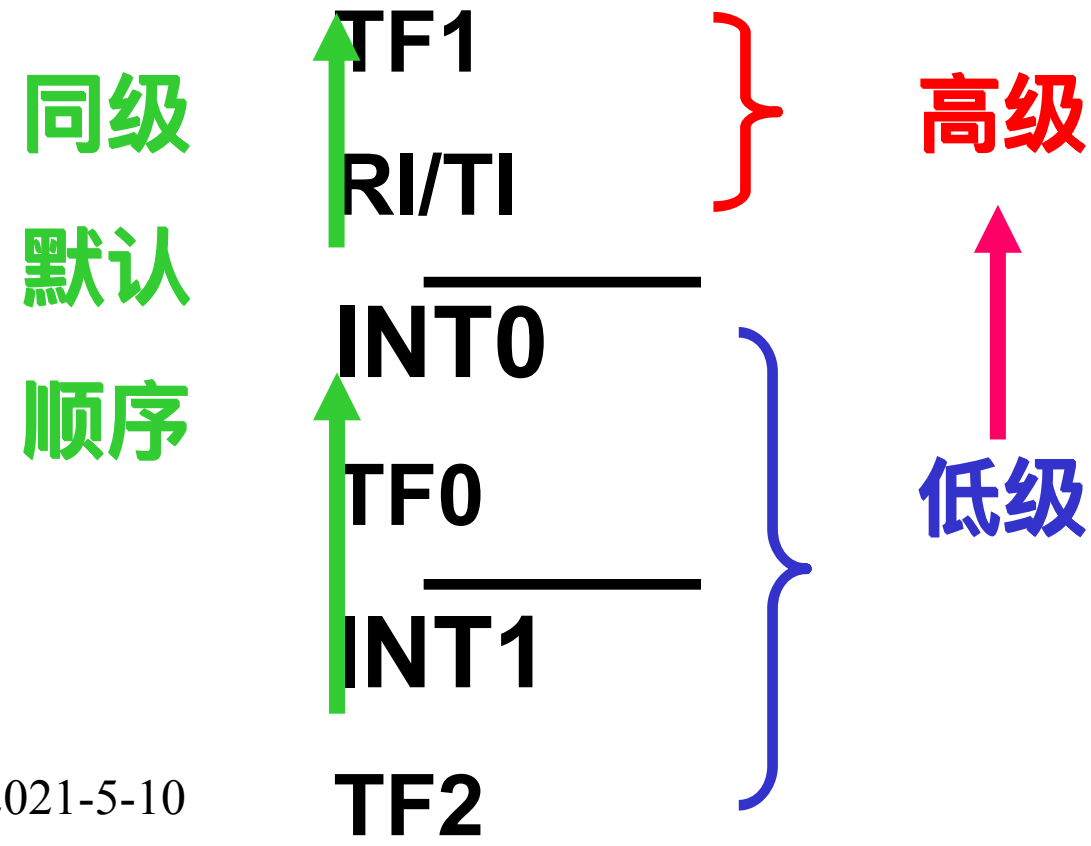
SETB　PS

SETB　PT1

问：设置后，那个中断源的优先级最高?

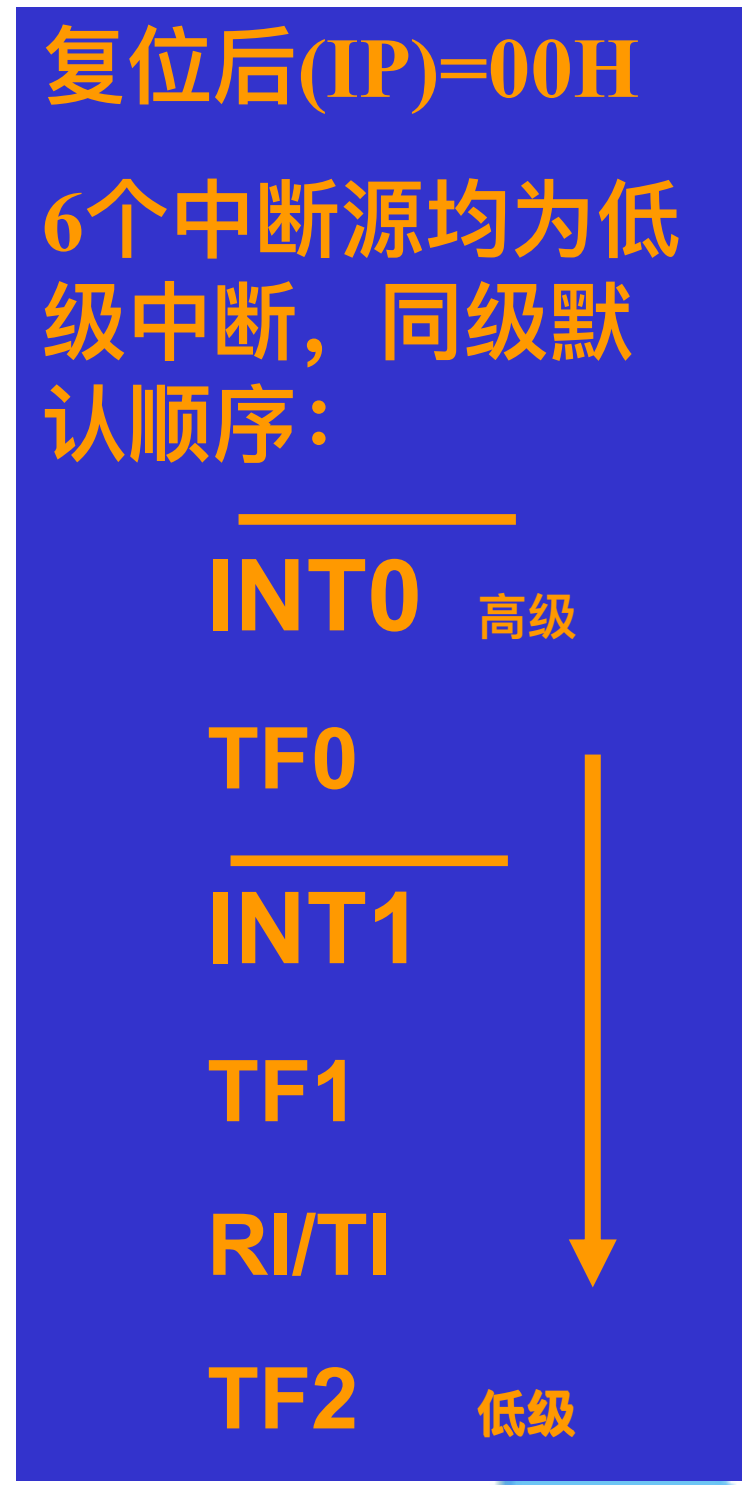
那个中断源的优先级最低?

答：　设置后，优先级的顺序为：



复位后(IP)=00H

6个中断源均为低级中断，同级默认认顺序：



5.4.3 AT89C52对中断的响应与撤消

- 中断响应的基本条件
- 中断响应过程
- 中断响应时间
- 中断请求的撤消

AT89C52中断响应的基本条件

中断响应的基本条件：

- 首先要有中断源发出中断申请；
- 中断总允许位EA=1，即CPU允许所有中断源申请中断；
- 在中断源寄存器TCON和SCON中，申请中断的中断允许位为1，即此中断源可以向CPU申请中断。

CPU执行程序过程中，在每个机器周期的S5P2期间，中断系统对各个中断源进行采样。这些采样值在下一个机器周期内按优先级和内部顺序被依次查询。

CPU 中断受阻的条件：

- CPU 正在处理相同或更高级中断
- 现行的机器周期不是所执行指令的最后一个机器周期
- 正在执行的指令是访问IE、IP或RETI指令，在执行上述指令后至少再执行一条指令，才可能响应中断。

AT89C52 响应中断的过程

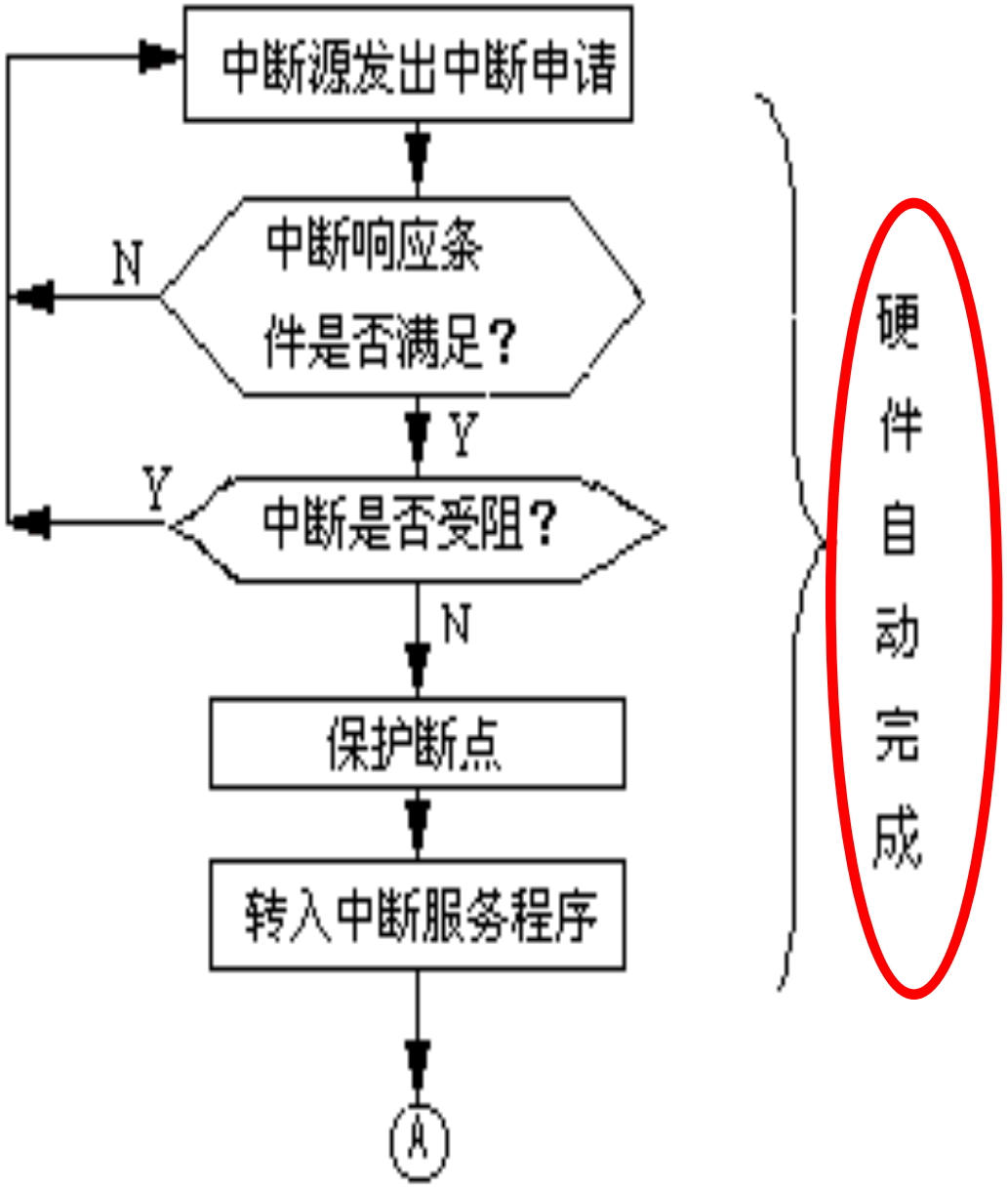
A CPU自动完成：

- CPU先在每个机器周期的S5P2期间，对各中断源重复进行查询，并设置相应的中断标志位 。
- 如果中断响应条件满足，且不存在中断阻断的情况，则CPU就响应中断。
- 硬件生成长调用指令自动地把断点地址压入堆栈保护，并随之将对应的中断入口装入程序计数器PC，使程序转向该入口地址，以执行中断服务程序。

用户必须完成：

在这些入口地址存放一条无条件跳转指令，使程序跳转到用户安排的中断服务程序起始地址上去 。

AT89C52 的中断响应过程



中断入口地址表

中断源	入口地址	同级中断优先级
IE0	0003H	
TF0	000BH	
IE1	0013H	
TF1	001BH	
TI/RI	0023H	
TF2	002BH	

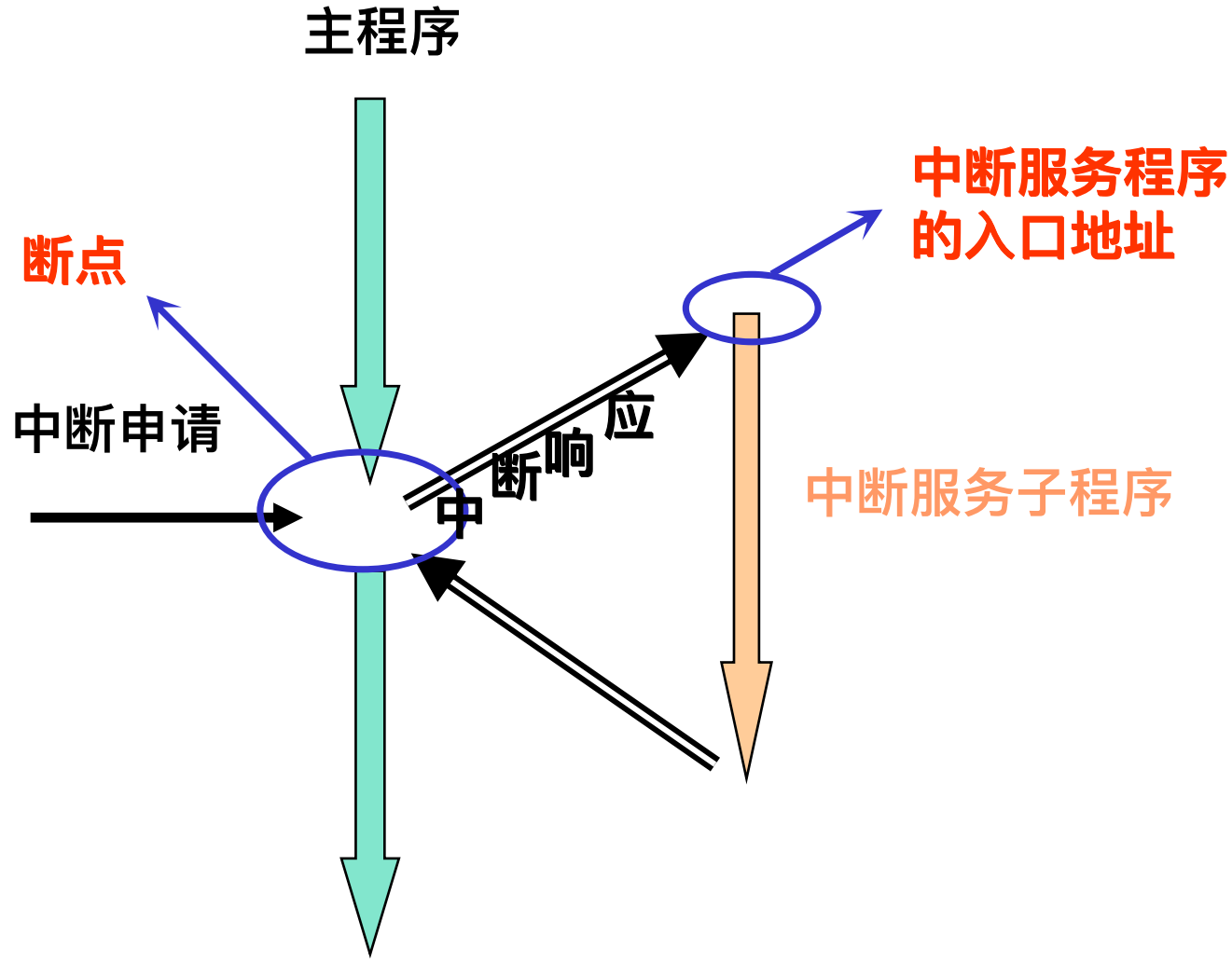
- 一旦CPU决定响应中断，则CPU会去执行相应的中断服务程序。
- 中断响应的过程包括：
  - 保护断点地址
  - 程序转至中断服务程序的入口地址
  - 执行中断服务子程序

系统自动放入堆栈保护！

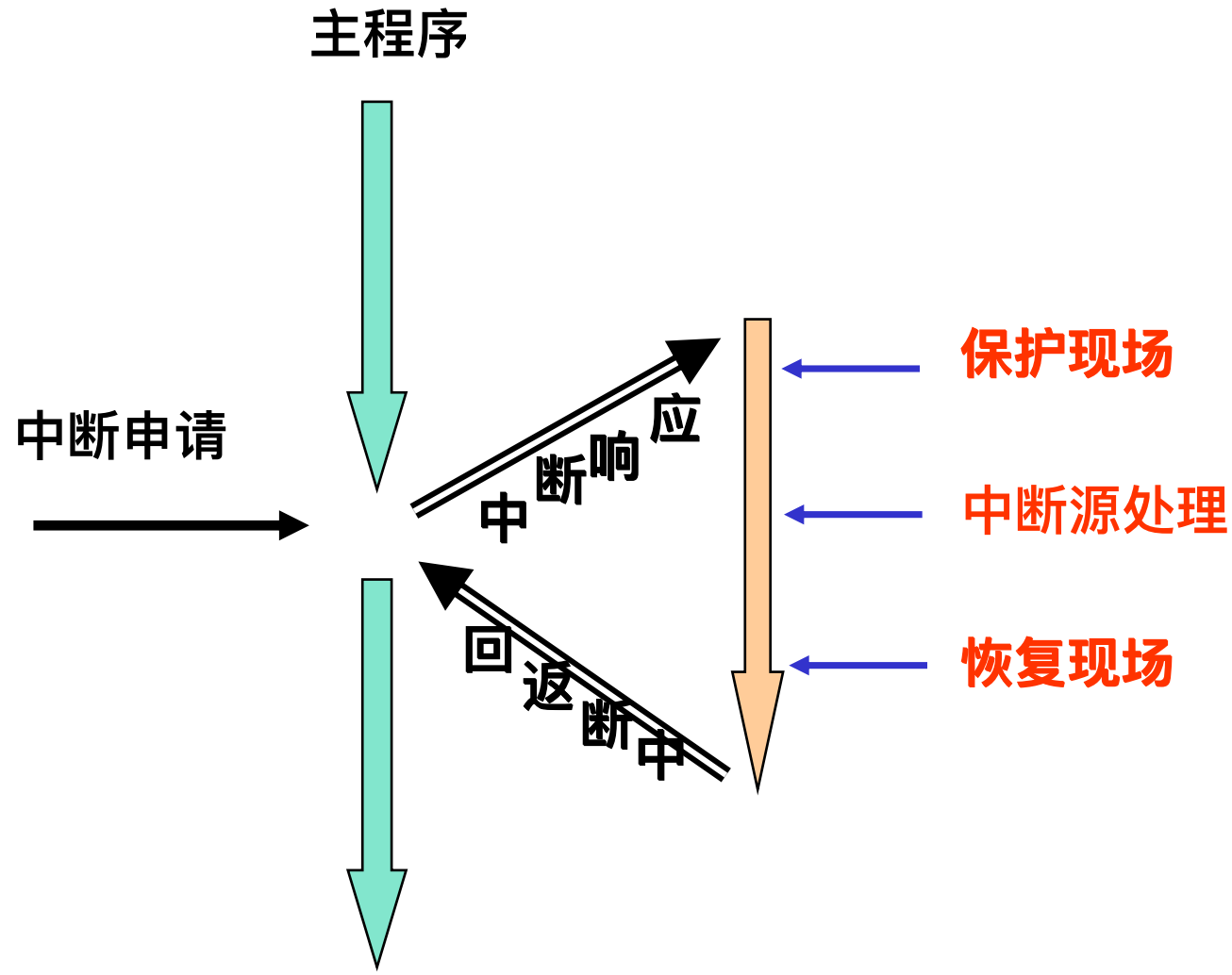
复现场

请求

哪里去呢？



- **中断响应——中断服务**
  - 保护现场
  - 处理中断源要求
  - 恢复现场



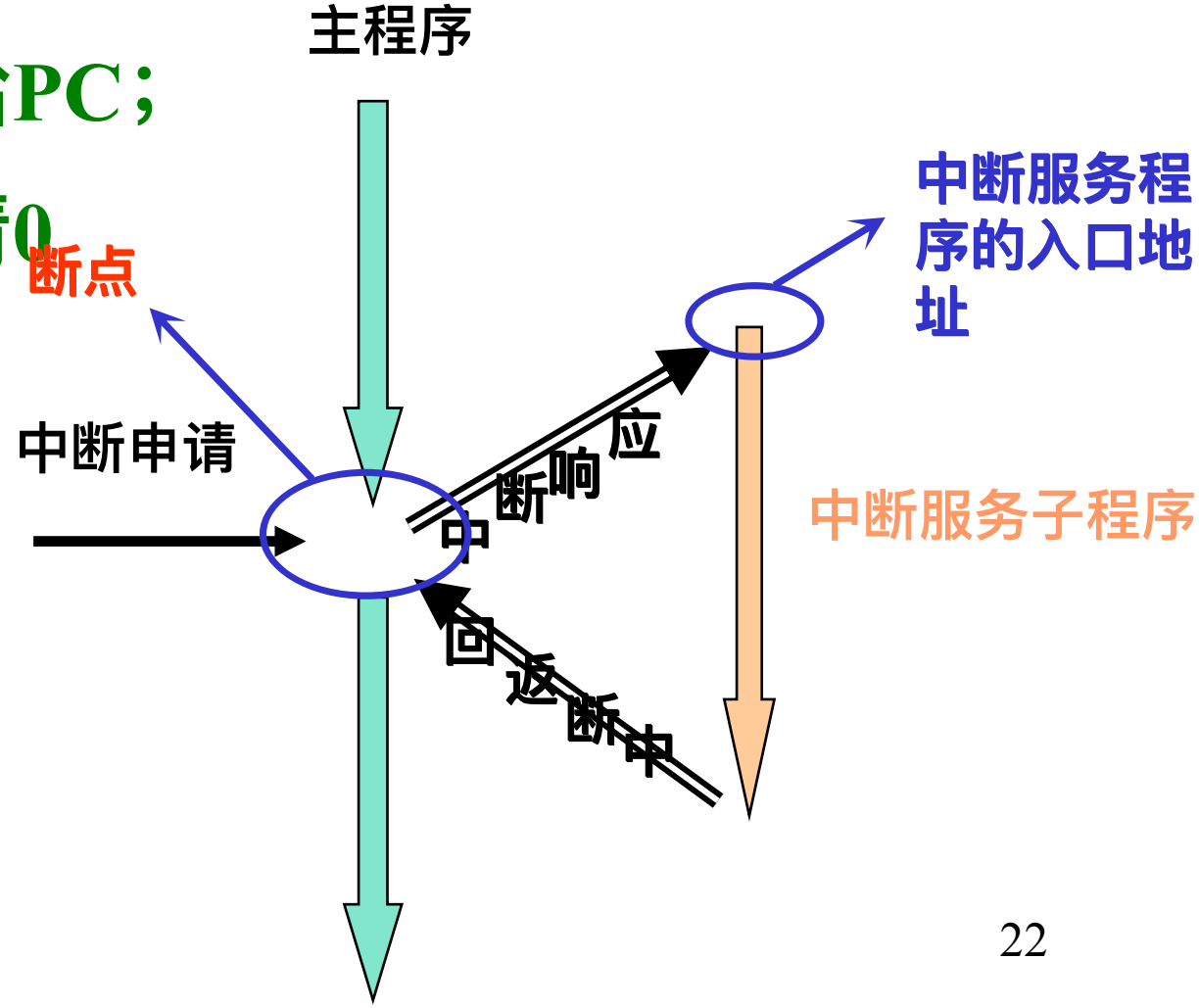
中断返回

- CPU完成中断服务后，返回到原来暂停的位置(即断点)，继续执行原来的程序。

➤ CPU返回方式：使用RETI指令返回

➤ RETI指令的任务：

- 取出保存的断点地址给PC；
- 将优先级状态触发器清0



RETI指令的具体功能是：

- **将中断响应时压入堆栈保存的断点地址从栈顶弹出送回PC，CPU从原来中断的地方继续执行程序；**
- **将相应的中断优先级状态触发器清0，通知中断系统，中断服务程序已执行完毕。**

注意：不能用RET指令代替RETI指令。

在中断服务程序中PUSH指令与POP指令必须成对使用，否则不能正确返回断点 。

多级中断系统的处理过程

CLR EA

用户编程设计

允许更高级中断打断此中断服务程序

低级中断源的中断服务程序

在恢复现场时，CPU不响应新的中断请求，使数据不被破坏。

SETB EA

中断服务子程序

AT89C52中断响应时间

1、一个单级的中断系统中必要响应时间

从中断源发出中断请求信号到CPU响应中断请求并转向中断入口地址所需要的时间就是中断响应时间，至少要经历三个完整的机器周期：

第一个机器周期用于查询中断标志位；

第二和第三个机器周期用于保护断点并自动转入执行一条长转移LCALL指令 ,进入中断服务程序入口地址。

2、附加等待时间

逢RETI或其它访问IE、IP指令（2周期）时，则需要把当前指令执行完以后，再继续执行一条指令后才能响应中断。

而指令所需的最长时间（MUL或DIV指令）是4个机器周期。

得出结论: 在一个单级的中断系统中，MCS-51响应中断的时间一般在3-8个机器周期之间。

3、中断受阻引起响应时间延长

当一个同级或更高级的中断正在进行，则附加的等待时间取决于正在进行的中断程序。

中断请求的撤除

CPU必须在中断返回前，把它的响应中断标志位复位成“0”状态。单片机的6个中断源，撤除中断请求的方法是不相同。

1. 内部硬件自动清除中断标志

T0/T1：TF0/TF1；

外部中断0/1（下降沿触发）： IE0/IE1

2. 软件清除中断标志

T2：TF2/EXF2；

串行口中断：RI/TI

3. 外加硬件结合软件清除中断请求

外部中断请求（电平触发）：IE0/IE1可复位为“0”

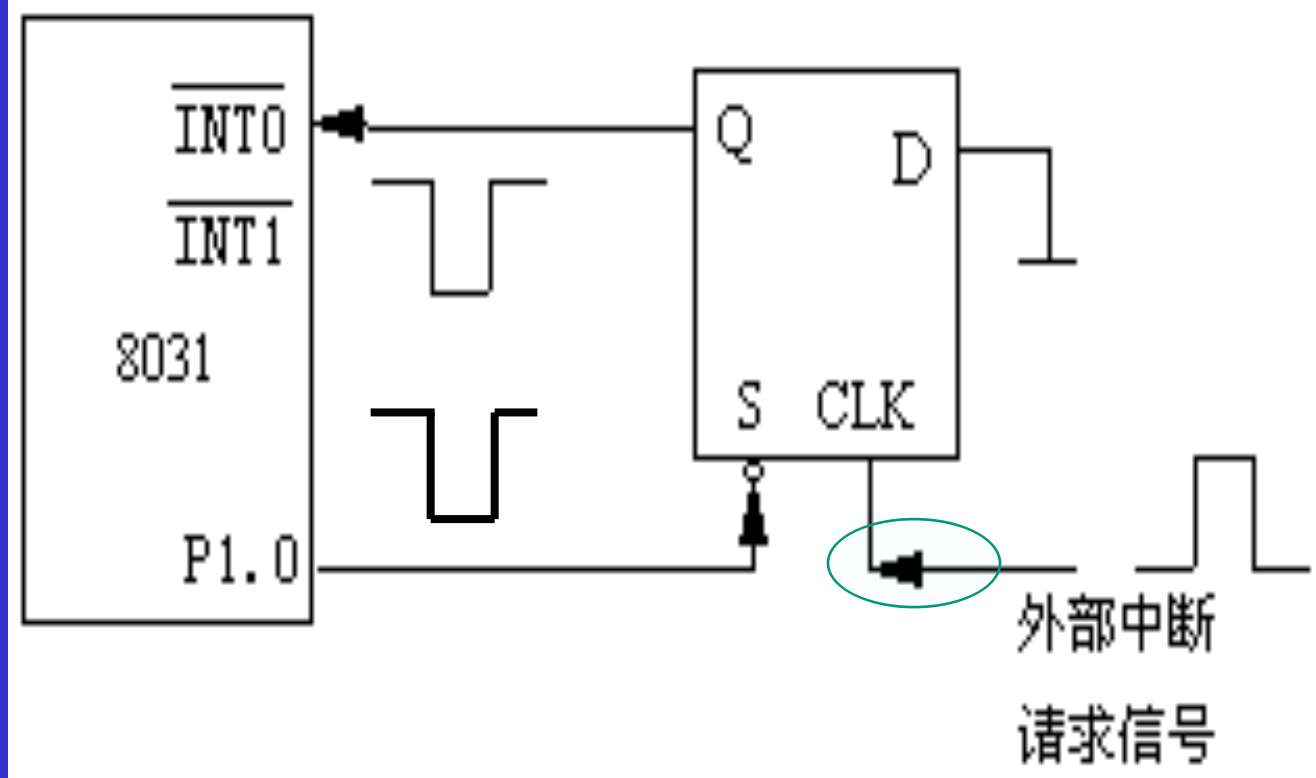
如果不能及时撤除有效电平，再次申请中断。所以可加硬件电路。

外部中断请求的撤除

对于负边沿触发的外部中断，CPU在响应中断后，是用硬件自动清除有关的中断请求标志IE0或IE1，来撤除INT0或INT1上的中断请求。

对于电平触发的外部中断，IE0或IE1是依靠CPU检测INT0或INT1上低电平而置位。尽管CPU响应中断时，IE0或IE1可用硬件自动复位成“0”状态，但若外部中断源不能及时撤除INT0或INT1上低电平，就会再次使已经变“0”的中断标志IE0或IE1置位，这是绝对不能允许的。所以，对于电平触发的外部中断请求的撤除，必须随着其中断被响应，使INT0或INT1上低电平变为高电平。

```
INSVR：ANL   P1，#0FEH
                ；   使S端有效
        ORL   P1，#01H
                        ;S端无效
          ┆
        RETI
        END
```



用D触发器锁存外部中断请求信号。响应中断后，转入服务程序。利用直接置1端S来实现撤消请求信号。只要输出一个负脉冲,使D触发器置1,从而撤消中断请求信号。(可在中断服务程序开始使/INT0或/INT1电平变高，以撤除中断请求。)

- 执行第一条指令使P1.0输出为负，其持续时间为2个机器周期，足以使D触发器置位，INT0上电平变高，从而撤除中断请求。

- 第二条指令使P1.0变为1，否则，D触发器的S端始终有效，INT0端终为1，无法再次中断。

MCS-51 中断请求的撤除

中断源	撤消中断标志的方法
外部中断0/1	当边沿触发中断，则硬件置位，硬件清除
	当低电平触发中断，硬件置位， 必须人为撤消中断请求信号
定时/计数器 0/1	由硬件置位和复位，硬件自动撤消
串行口RI/TI	由硬件置位， 必须由软件复位 CLR RI ,CLR TI

5．4．4 AT89C52中断系统的初始化及应用

1、中断系统的初始化

中断系统初始化步骤为：

1）CPU开中断或关中断；

2）某中断源中断请求的允许或禁止（屏蔽）；

3）设定所用中断的中断优先级；

4）若为外部中断，则应规定低电平还是负边沿的中断触发方式。

[例] 请写出/INT1为低电平触发的中断系统初始化程序

解： (1)采用位操作指令

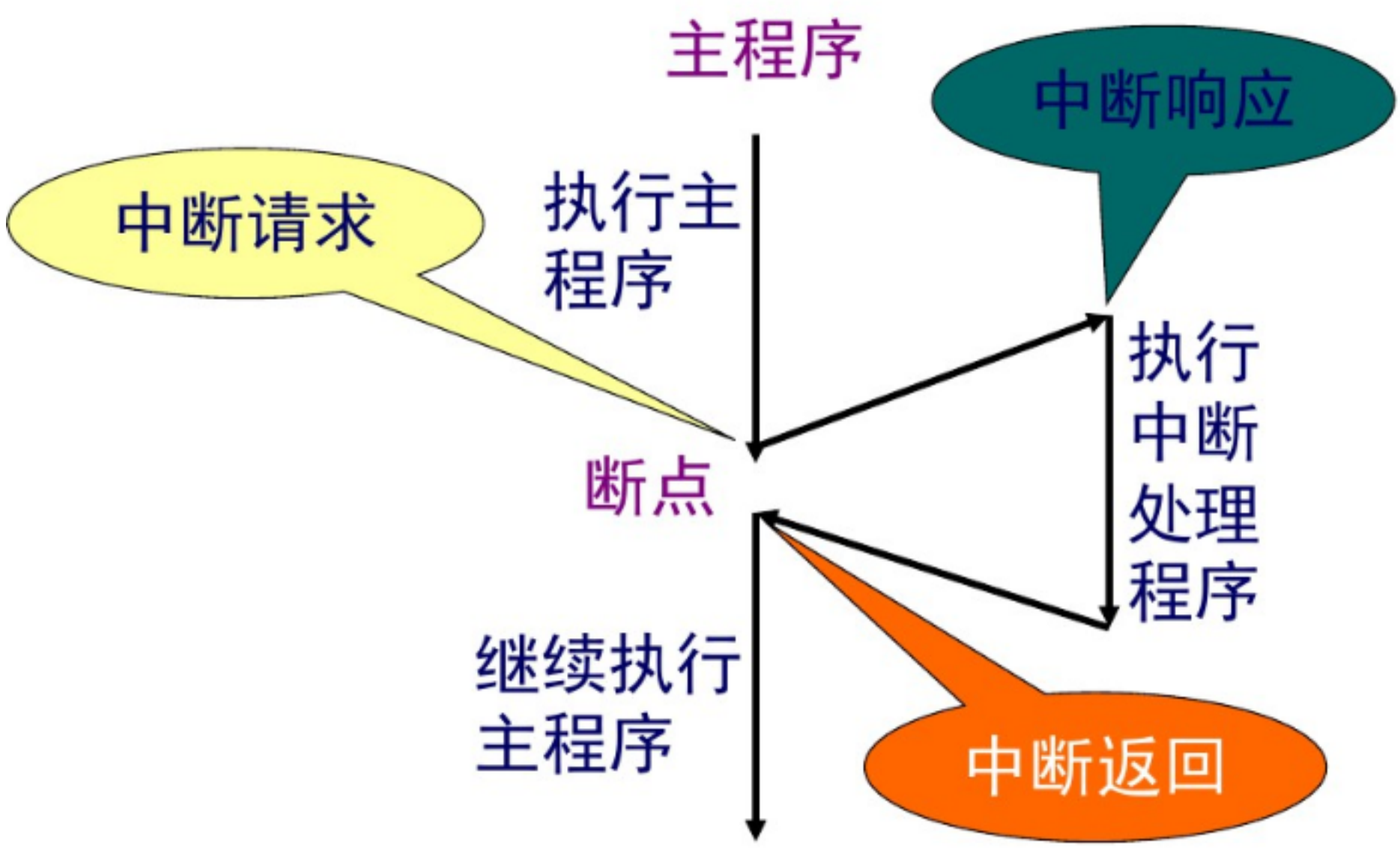
```
SETB  EA   ；CPU开中断
SETB  EX1  ；开/INT1中断
SETB  PX1  ；令/INT1为高优先级
CLR   IT1  ；令/INT1为电平触发
```

(2)采用字节型指令

```
MOV  IE，#84H   ；开/INT1中断
ORL   IP，#04H  ；令/INT1为高优先级
ANL  TCON，#OFBH   ；令/INT1为电平触发
```

IT1=0

2、程序设计方法



(1)主程序

主程序除了用来对MCS-51**本身**中断系统初始化外，还经常用来对具有中断功能的**外部扩展**的可编程I/O接口、可编程内部定时器/计数器T0、T1和可编程内部串行口等初始化，程序如下：

```
        ORG   0000H
        AJMP  MAIN
        ┆
MAIN:                    ;  MAIN为主程序首地址
        ┆
```

必须在0000H单元放一条跳转到主程序的跳转指令，实现了上电自动转向主程序

复位后，其PC的内容被强迫置成0000H

上电启动，CPU就执行0000H单元的指令

（2）中断服务程序

RETI

中断服务程序是一种具有特定功能的独立程序段，它为中断源的特定要求服务，其一般**低级中断源程序**编写格式如下：

```
        ORG  ZDRK
        AJMP  INTVS
          ┆
INTVS:CLR   EA
        PUSH  ACC
        PUSH  01H
          ┆
        SETB  EA
          ┆
中断服务
          ┆
        CLR  EA
        POP  01H
        POP  ACC
        SETB  EA
        RETI
```

嵌套

中断服务程序入口

地址：ZDRK

0003H

000BH

0013H

001BH

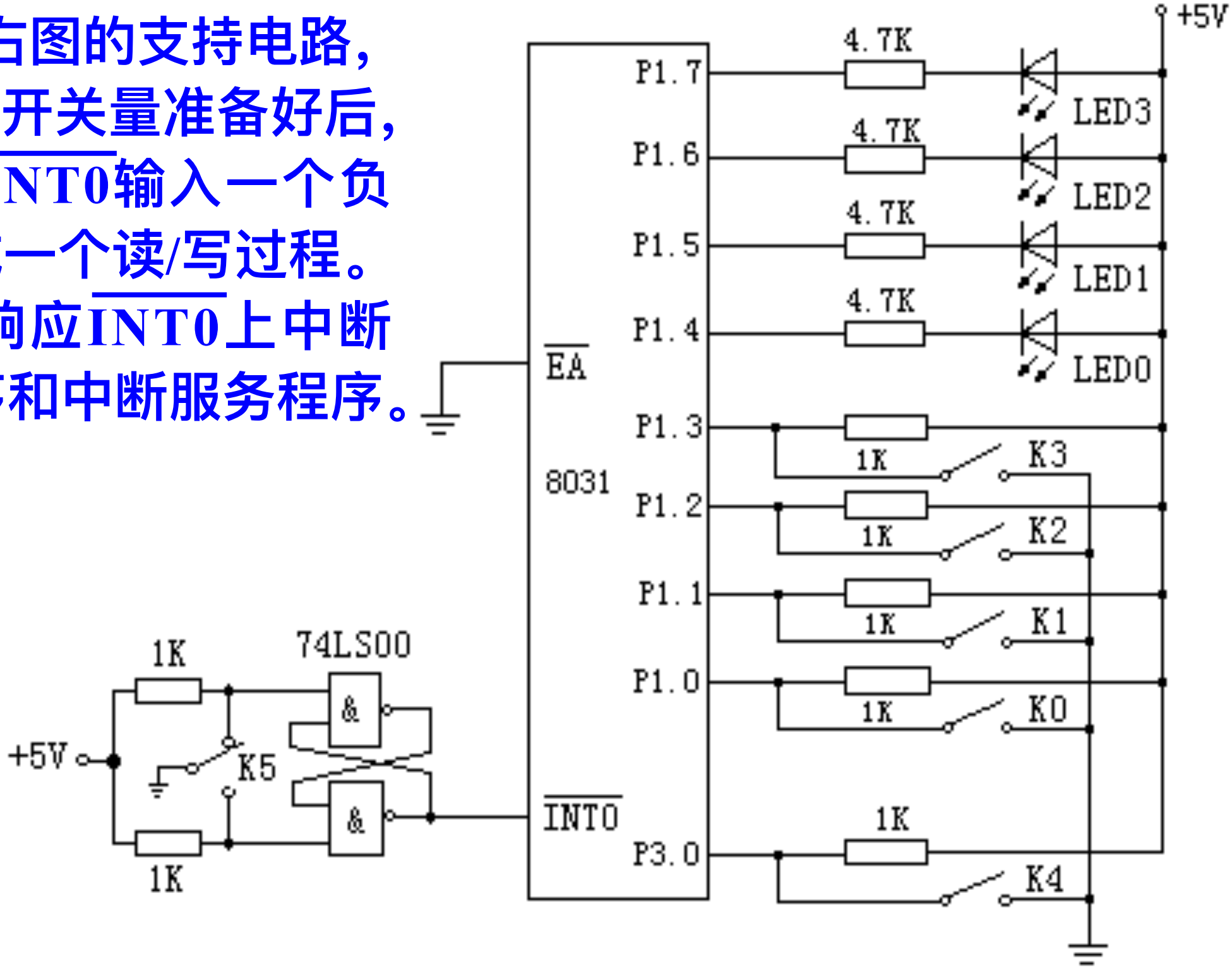
0023H

中断服务程序的设计

```
GJZHD:   CLR EA
         保护现场
         ……
         中断处理
         ……
         ……
         ……
         ……
         恢复现场
         SETB  EA
         RETI
```

AT89C52 的中断程序设计举例

[例] 根据右图的支持电路，当K0-K3的开关量准备好后，按下K5向$\overline{\text{INT0}}$输入一个负脉冲，完成一个读/写过程。编出CPU响应$\overline{\text{INT0}}$上中断时的主程序和中断服务程序。



```
ORG 0000H
AJMP MAIN
ORG 0003H
LJMP EX1SVR
ORG 0100H
MAIN:MOV  SP#6FH
     SETB IT0;令/INT0边沿触发
     SETB EX0;开
     SETB EA ;开中断
     SJMP $;   等待中断
```

初始化部分

```
EX1SVR: PUSH PSW
        PUSH ACC
        MOV A,#0FFH; “1”
        MOV P1,A;  设输入态
        MOV A,P1;  读P1,开关数
        SWAP A   ；换位
        MOV P1,A; 写入P1，输出
        POP ACC
        POP PSW
        RETI
        END
```

问：响应中断后什么地址压入堆栈?

答：将 SJMP $ 指令的转移目标地址压入堆栈。即 中断在本指令处产生，本指令的目标地址是本指令地址。

问：若中断在CALL指令执行后产生，响应中断后什么地址压入堆栈?

答：将子程序入口地址压入堆栈，因为它是CALL指令的转移目标地址， 而不是CALL的下一条指令地址。

问：若要控制中断的次数，在主程序中采用CJNE R7,#10,NEXT指令，NEXT标号应写在什么语句上?

答：将NEXT标号写在本指令上。

即：NEXT:CJNE R7,#10,NEXT；等待中断

必须在中断服务程序中用INC R7指令更新R7的值。

R7的初值在主程序的初始化部分置入。

子程序与中断服务程序的区别

- 对强迫中断的服务程序具有随机性
  - 要考虑可能在程序的什么指令处发生，
  - 要保护什么内容，才能保证返回断点后正常工作。
- 对人为设置的软件中断与子程序调用的区别
  - 返回指令不同
    - 子程序返回用RET
    - 中断服务程序返回用RETI
  - 处理内容不同，中断一般处理I/O操作。

第五章结束