

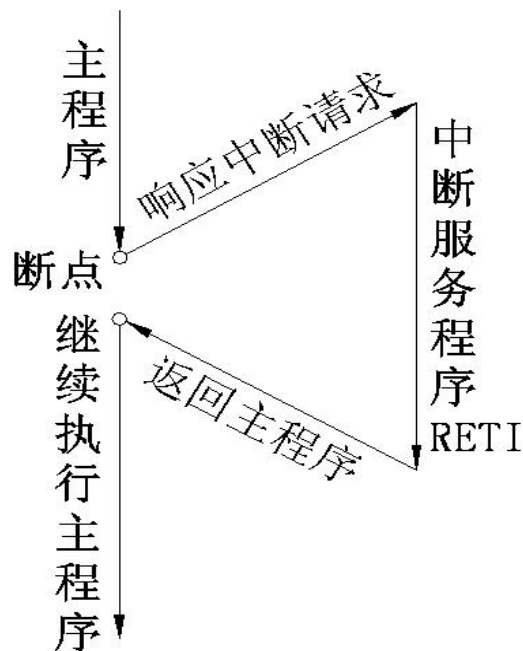
中断的概念

当CPU正在处理某事件的时候，外部发生的某一事件请求CPU迅速去处理，于是CPU暂时中止当前的工作，转去处理所发生的事件。中断服务处理完该事件后，再返回到原来被中止的地方继续原来的工作，这样的过程称为中断。

保护现场，恢复现场，中断返回

中断作用：

- 1) 中断系统提高了CPU对外界异步事件的处理能力，解决了快速CPU与慢速的外部设备之间的矛盾，大大提高了CPU的利用率。
- 2) 中断系统使CPU能够及时地处理内部和外部的随机信息，提高了响应的实时性和故障处理能力。

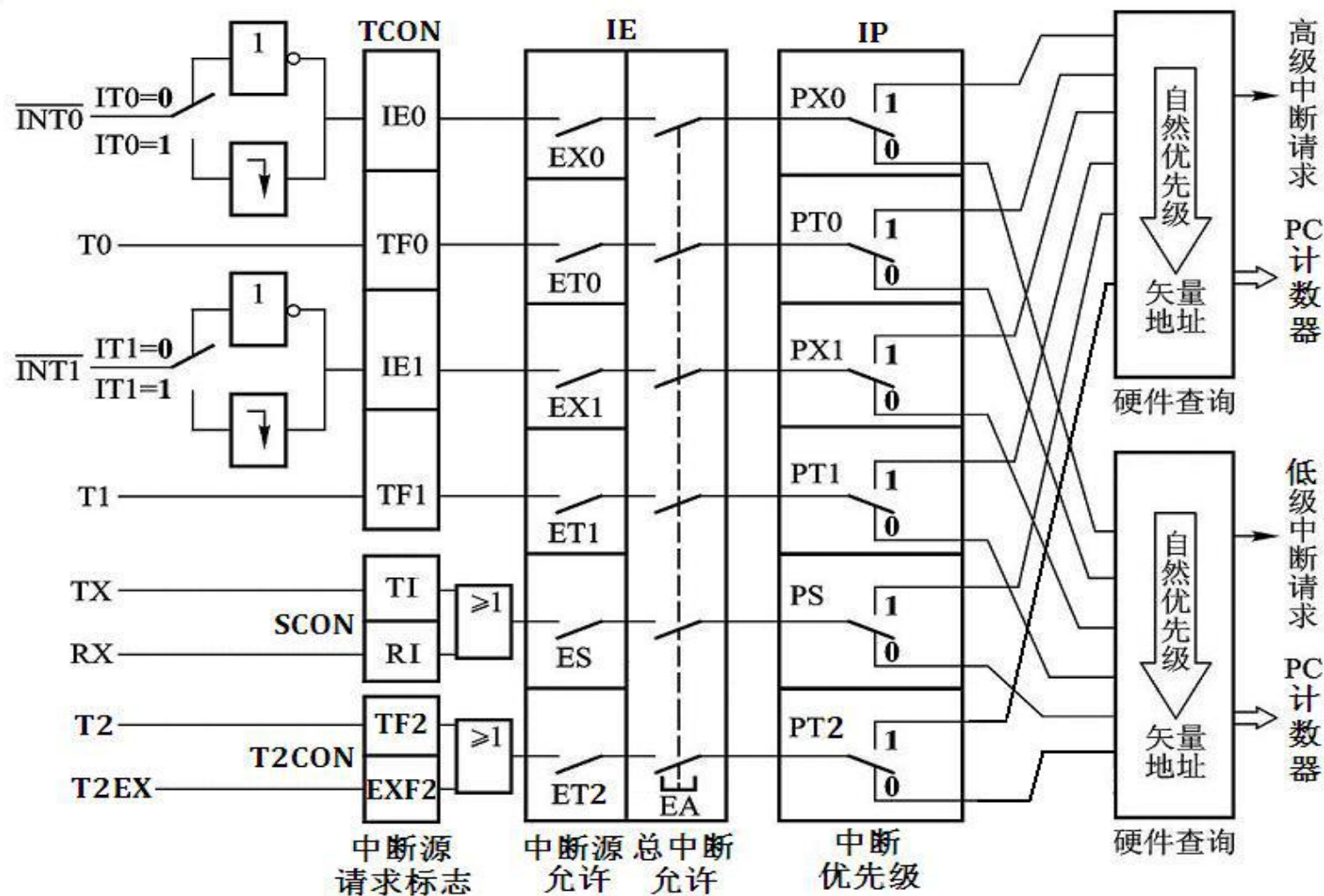


中断系统结构

- ◆ 8031系列单片机有5个中断源——2个外部中断源 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ ，2个片内定时器 / 计数器（T0和T1）溢出中断源，1个片内串行口中断源。**8032系列单片机有6个中断源，增加了一个定时器/计数器T2中断源。**
- ◆ 每一个中断源都可独立设置为开/关中断。
- ◆ 分为两级——高级中断和低级中断。其中任何一个中断源的优先级均可由软件设定为高级或低级，能实现两级中断服务程序嵌套。

第5部分 MCS-51的中断系统

中断系统结构



中断请求源

- 1) 外部中断0请求，由P3.2 ($\overline{\text{INT0}}$) 引脚输入。可由用户设定为两种触发方式：电平触发或边沿触发。一旦输入信号有效，则向CPU申请中断，并且将中断标志IE0置1。
- 2) 外部中断1请求，由P3.3 ($\overline{\text{INT1}}$) 引脚输入。亦可由用户设定为电平触发方式还是边沿触发方式。一旦输入信号有效，则向CPU申请中断，并将中断标志IE1置1。
- 3) 片内定时器T0溢出中断请求。当定时器T0产生溢出时，T0中断请求标志TF0置1，请求中断处理。
- 4) 片内定时器T1溢出中断请求。当定时器T1产生溢出时，T1中断请求标志TF1置1，请求中断处理。
- 5) 片内串行口发送 / 接收中断请求。当通过串行口发送或接收完一帧串行数据时，串行口中断请求标志TI或RI置1，请求中断处理。
- 6) 片内定时器T2的中断请求。含有计数溢出 (TF2) 和“捕捉” (EXF2) 两种中断请求标志，经或门共用一个中断矢量。两种中断触发是由T2的两种不同工作方式决定的。

中断请求源

特殊功能寄存器TCON

TCON是专用寄存器，字节地址为88H，它锁存了外部的中断请求标志及T0和T1的溢出中断请求标志。复位后TCON为00H。

	D7	D6	D5	D4	D3	D2	D1	D0	
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H
位地址	8FH		8DH		8BH	8AH	89H	88H	

- ◆ **IT0**: 选择外部中断0 ($\overline{\text{INT0}}$) 触发方式控制位。IT=0, $\overline{\text{INT0}}$ 低电平触发中断；IT=1, $\overline{\text{INT0}}$ 负跳沿触发中断。
- ◆ **IE0**: 外部中断0请求标志位。IE0 = 1, 外部中断0向CPU申请中断。
- ◆ **IT1**: 选择外部中断1 ($\overline{\text{INT1}}$) 触发方式控制位。操作功能类似IT0位。
- ◆ **IE1**: 外部中断1请求标志位。IE1 = 1时, 外部中断1向CPU申请中断。

中断请求源

特殊功能寄存器TCON

TCON是专用寄存器，字节地址为88H，它锁存了外部的中断请求标志及T0和T1的溢出中断请求标志。复位后TCON为00H。

	D7	D6	D5	D4	D3	D2	D1	D0	
TCON	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	88H
位地址	8FH		8DH		8BH	8AH	89H	88H	

- ◆ **TF0**: 片内定时器T0溢出中断请求标志。T0被启动后，从初始值开始进行加1计数，当最高位产生溢出时置TF0=1，向CPU申请中断。
- ◆ **TF1**: 片内定时器T1溢出中断请求标志，其操作功能与TF0类同。

中断请求源

特殊功能寄存器T2CON

T2CON是专用寄存器，字节地址为0C8H，可位寻址（0C8H~0CFH）。复位后T2CON为00H。

	D7	D6	D5	D4	D3	D2	D1	D0	
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{T2}$	CP/ $\overline{RL2}$	C8H

T2CON中的最高两位为定时器/计数器T2的中断请求标志位TF2和EXF2。

- **TF2:** 当T2的计数器（TL2、TH2）计数计满溢出回0时，由内部硬件置位TF2，向CPU发出中断请求。但是当RCLK位或TCLK位为1时将不予置位。本标志位必须由软件清0。

中断请求源

特殊功能寄存器T2CON

	D7	D6	D5	D4	D3	D2	D1	D0	
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/ $\overline{\text{T2}}$	CP/ $\overline{\text{RL2}}$	C8H

- **EXF2**: 当由引脚T2EX (P1.1脚) 上的负跳变引起“捕捉”或“重新装载”且EXEN2位为1, 则置位EXF2标志位 (寄存器T2CON.6), 向CPU发出中断请求。

上述两种中断请求, 在满足中断响应条件时, CPU都将响应其中断请求, 转向同一个中断矢量地址进行中断处理。因此, 必须在T2的中断服务程序中对TF2和EXF2两个中断请求标志位进行查询判别, 然后正确转入对应的中断处理程序。

中断结束后, 中断请求标志位TF2或EXF2必须由软件清0。

中断请求源

特殊功能寄存器SCON

SCON是串行口控制寄存器，与中断有关的是它的低两位TI和RI。

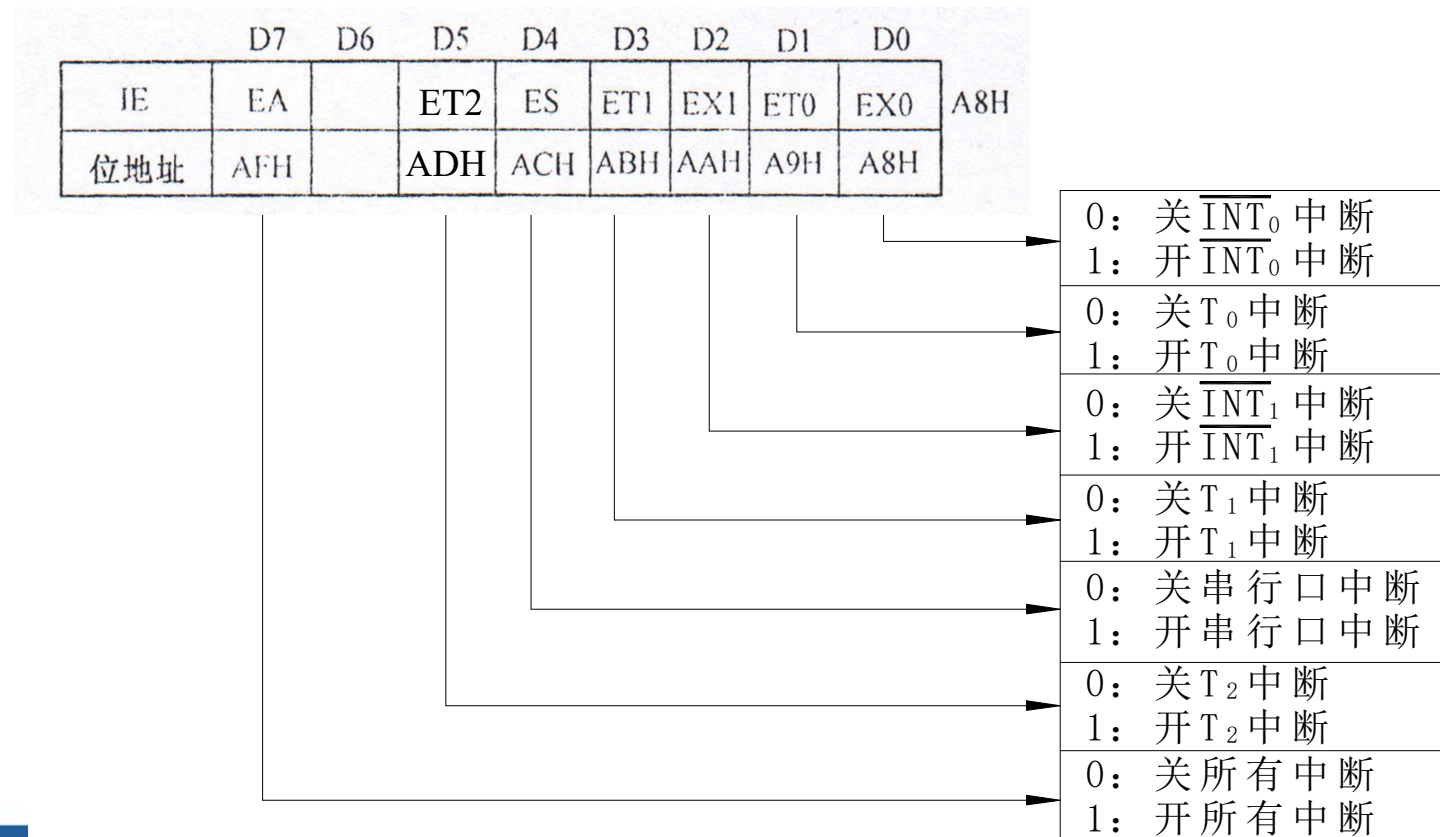
	D7	D6	D5	D4	D3	D2	D1	D0	
SCON	SM0	SM1	SM2	REN	TB8	RB8	TI	RI	98H
位地址							99H	98H	

- ◆ **TI (SCON.1)**：串行口发送中断请求标志位。CPU每将一个字节写入发送缓冲器SBUF，启动发送完一个串行帧数据时，由硬件置位TI，请求中断。
- ◆ **RI (SCON.0)**：串行口接收中断请求标志位，当允许串行口接收数据时，每接收完一个串行帧数据，由硬件置位RI，请求中断。
- ◆ CPU在响应串行中断时，并不自动清零标志位TI和RI，需在中断服务程序中通过软件清零。

中断控制

中断允许寄存器IE

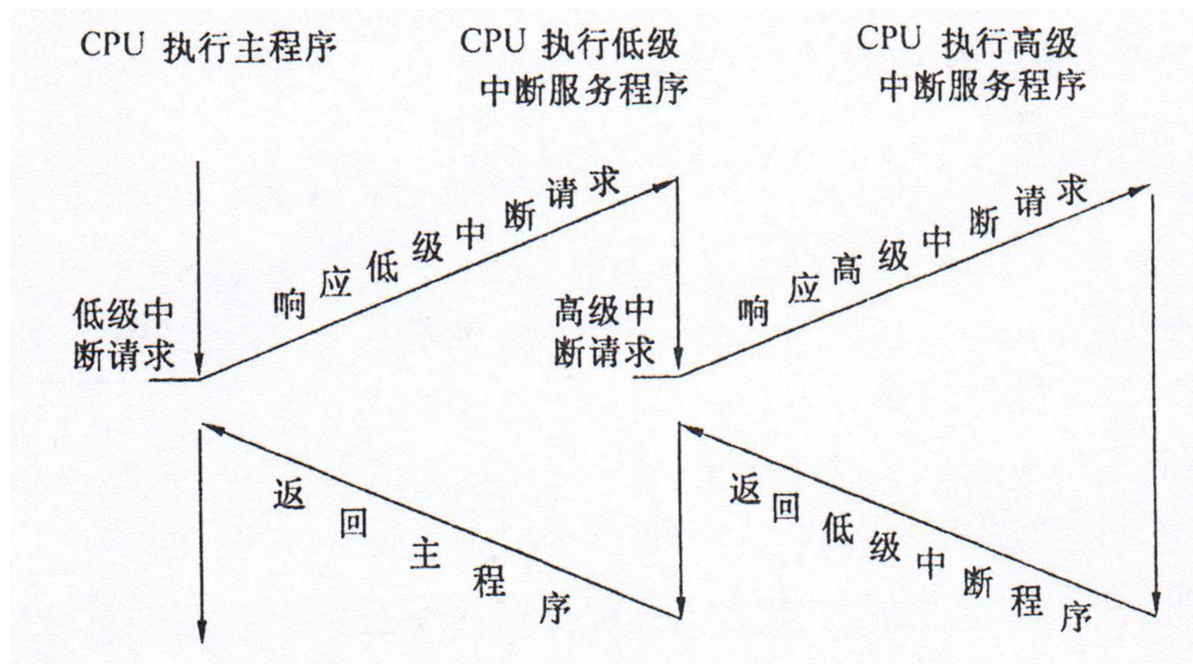
中断允许寄存器IE对中断的开放和关闭实现两级控制，第一级是总的开关中断控制位EA，另一级是个中断源对应的中断请求允许位。



中断控制

中断优先级寄存器IP

MCS-51单片机的中断系统有两个中断优先级，每一个中断请求源可以由软件设置为高优先级中断或低优先级中断。可实现两级中断嵌套，执行低优先级中断服务程序时，可被高优先级中断请求所中断。



中断控制

IP			PT ₂	PS	PT ₁	PX ₁	PT ₀	PX ₀	B8H
位地址			BD	BC	BB	BA	B9	B8	

不用

中断优先级寄存器IP

INT ₀ 中断优先级控制	0: 低优先级 1: 高优先级
T ₀ 中断优先级控制	0: 低优先级 1: 高优先级
INT ₁ 中断优先级控制	0: 低优先级 1: 高优先级
T ₁ 中断优先级控制	0: 低优先级 1: 高优先级
串行口中断优先级控制	0: 低优先级 1: 高优先级
T ₂ 中断优先级控制	0: 低优先级 1: 高优先级

中断控制

中断优先基本规则

- 1) 低优先级可被高优先级中断，反之则不能。
- 2) 任何一种中断（不管是高级还是低级），一旦得到响应，不会再被同级中断源中断。
- 3) 同时收到几个同级的中断请求时，哪一个中断请求能获得响应，取决于内部的硬件查询顺序（同级顺序）。

表 4-1 同一优先级中断的查询次序

中断源	中断级别
外部中断 0	最高 ↓ 最低
T0 溢出中断	
外部中断 1	
T1 溢出中断	
串行口中断	
T2 中断	

响应中断请求的条件

中断响应的必要条件

- 1) CPU开中断，即IE寄存器中的中断总允许位EA=1。
- 2) 该中断源的中断允许位为“1”，即该中断没有被屏蔽。
- 3) 该中断源发出中断请求，即该中断源对应的中断请求标志为“1”。
- 4) 无同级或更高级的中断正在被服务，并且无同级顺序更高的中断源正在请求中断。

表 4-2 中断入口地址表

中断源	中断入口地址
外部中断 0	0003H
定时器/计数器 T0	000BH
外部中断 1	0013H
定时器/计数器 T1	001BH
串行口中断	0023H
定时器/计数器 T2 (T2+EXF2)	002BH

右表是各中断服务程序入口地址

通常在各中断服务程序入口地址处放置一条无条件转移指令，使程序转向中断服务程序。

响应中断请求的条件

中断被推迟响应的条件（中断封锁条件）

- 1) CPU正在处理同级或更高优先级的中断；
- 2) 现行机器周期不是正在执行的指令的最后一个机器周期，即现行指令完成前，不响应任何中断请求；
- 3) 正在执行的是中断返回指令RETI或访问专用寄存器IE或IP的指令。也就是说，在执行RETI或是访问IE、IP的指令后，至少需要再执行一条其他指令，才会响应中断请求。

中断查询在每个机器周期中的S5P2重复执行，在S6进行优先级排序。如果某中断标志被置位，但因上述三种情况未能即时响应中断，该中断将继续等待。但是若封锁条件撤消后该中断的标志位已不再存在（被软件清零），被拖延的中断将不再被响应。

外部中断的响应时间

MCS-51单片机对于外部中断的响应时间是3~8个机器周期，具体时间取决于中断请求时正在执行的指令。

外部中断的触发方式选择

- 低电平方式：外部中断采用低电平触发方式时，中断服务程序返回之前，外部中断请求的电平必须恢复到高电平，否则CPU返回主程序后将再次响应中断。
- 负跳沿触发方式：外部中断采用负跳沿触发方式时，负脉冲的低电平至少保持12个时钟周期。中断请求触发器能锁存中断输入线上的负跳变，即使CPU暂时不能响应，中断请求标志也不会丢失；直到CPU响应此中断，该标志才自动清零。

中断请求的撤销

1) 定时器/计数器T0、T1中断请求的撤销：中断响应后，硬件自动清除中断请求标志位TF0或TF1，中断请求自动撤销。

2) 外部中断请求的撤销

跳沿方式：中断响应后，硬件自动清除中断请求标志位IE0或IE1，外部中断的负跳沿信号是瞬态过程，不会维持。所以外部中断请求是自动撤销。

电平方式：硬件自动清除中断请求标志位IE0或IE1，但中断请求的低电平信号可能继续维持。可在系统中增加硬件电路解决（参见P97）。

中断请求的撤销

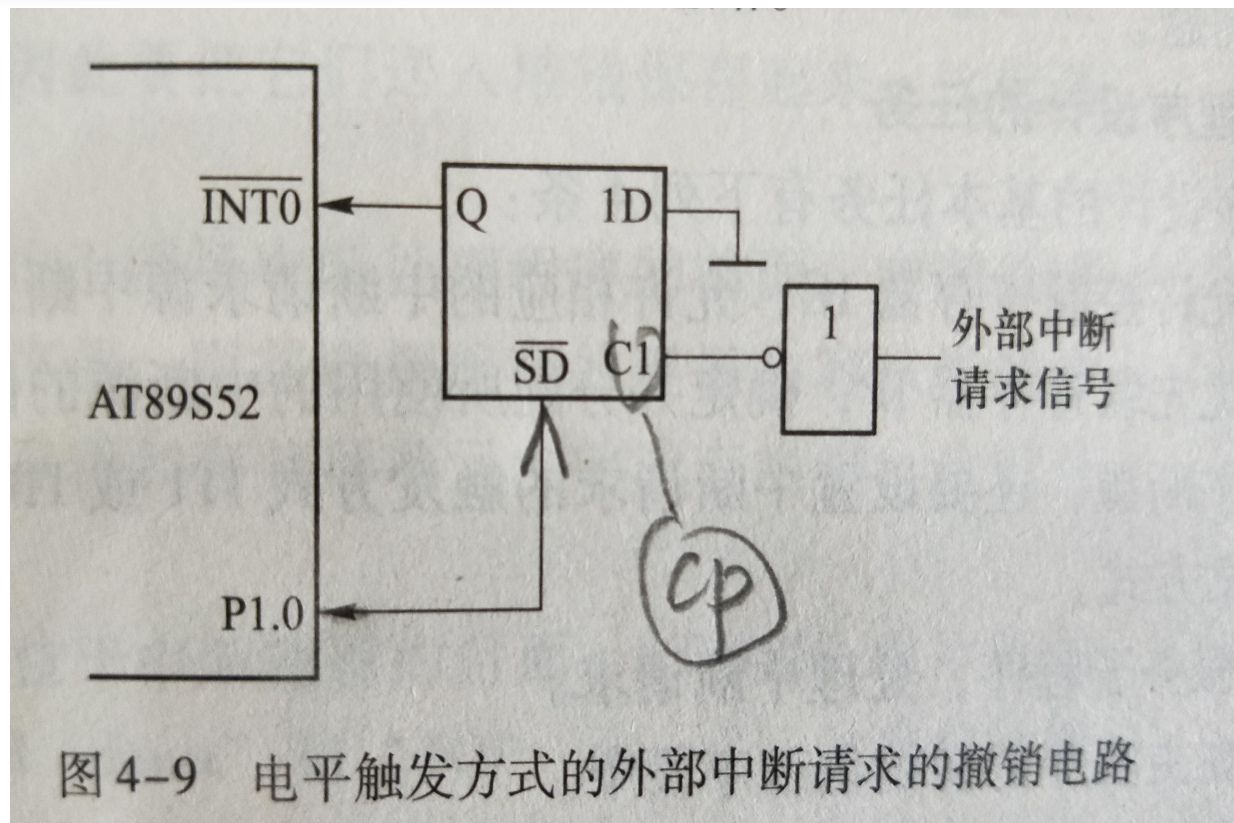


图 4-9 电平触发方式的外部中断请求的撤销电路

中断请求的撤销

3) 串行口中断请求的撤销: CPU响应后, 硬件不能自动清除TI和RI标志位, 因此CPU响应中断后, 必须在中断服务程序中, 用软件来清除相应的中断标志位, 以撤消中断请求。

4) 定时器2中断请求的撤销: 定时器2的中断请求是由TF2或EXF2产生, 它们被置位后不能由硬件清除, 必须在中断服务程序中, 用软件来清除TF2或EXF2的中断标志位, 以撤消中断请求。

中断服务程序的设计

中断服务程序设计的任务：

- 1) 设置中断允许控制寄存器IE，允许相应的中断请求源中断
- 2) 设置中断优先级寄存器IP，确定并分配所使用的中断源的优先级。
- 3) 若是外部中断源，还要设置中断请求的触发方式IT1或IT0，以决定采用电平触发方式还是跳沿触发方式。

例：SETB EA ; EA位置“1”，CPU开中断
SETB EX0 ; EX0位置“1”，允许外部中断0产生中断
SETB PX0 ; PX0位置“1”，外部中断0为高级中断
SETB IT0 ; IT0位置“1”，外部中断0为跳沿触发方式

中断服务程序的设计

中断时的主程序结构：

- 1) 主程序从起始地址0000H开始执行，通常在0000H处采用无条件跳转指令进入主程序。主程序初始化中要设置堆栈指针。
- 2) 各中断服务程序的入口地址是固定的，通常在各中断服务程序的入口地址采用无条件跳转指令进入各自的中断服务程序。

例：

```
ORG    0000H
LJMP   MAIN

ORG    中断入口地址
LJMP   INT
```

MAIN: 主程序

INT: 中断服务程序

中断服务程序的设计

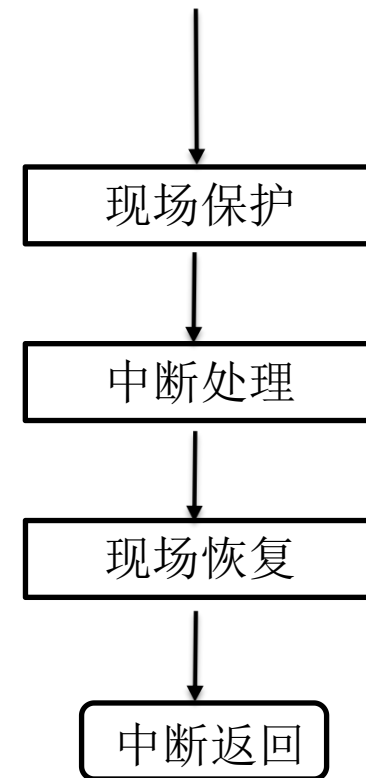
中断服务程序的流程：

1) 现场保护和现场恢复。

- ◆ 现场：中断服务程序内部使用的通用寄存器、存储器的数据和状态。
- ◆ 进入中断服务程序立即保护现场，中断返回之前恢复现场。
- ◆ 现场保护的顺序和现场恢复的顺序相反，**PUSH**和**POP**指令成对使用。

2) 中断处理

3) 中断返回。必须用**RETI**指令，不能用**RET**指令。

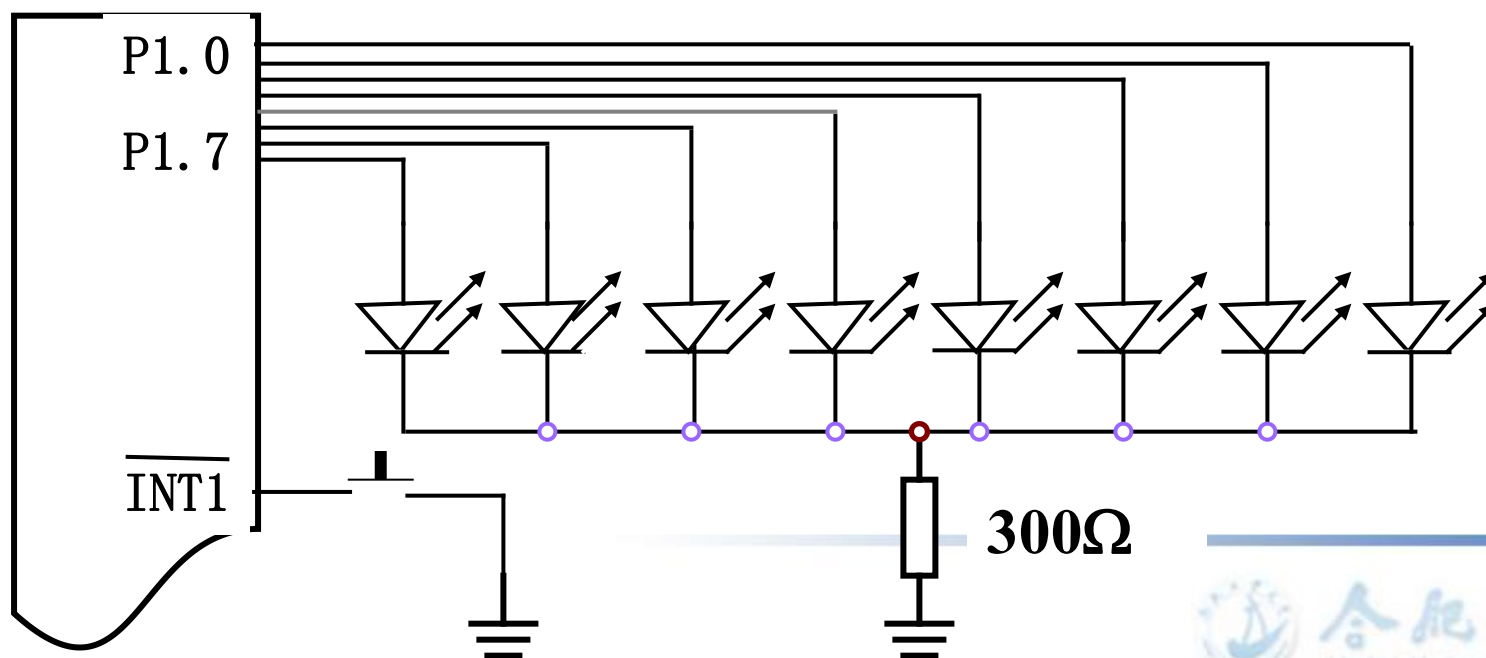


第5部分 MCS-51的中断系统

举例

通过外部中断1，在中断服务中将B寄存器里的内容左环移一位。
已知: (B)=01h, 要求采用边沿触发, 低优先级。

此例的实际意义: 在INT1引脚接一个按钮开关到地, 每按一下按钮就申请一次中断, 中断服务则是: 依次点亮八盏灯中的一盏。



```

        ORG 0000H
        LJMP MAIN
        ORG 0013H      ; 中断矢量
        LJMP INT
        ORG 0100H
MAIN:    MOV SP, #60H
        SETB EA        ; CPU开中断
        SETB EX1       ; 外部中断1中断允许
        CLR  PX1       ; 0 优先级（复位状态，可省略）
        SETB IT1       ; 边沿触发
        MOV  P1, #00H  ; 初始化将所有灯熄灭
        MOV  B, #01H   ; 给 B 寄存器赋初值
HERE:    .....
        SJMP HERE      ; 等待中断申请

```

```

        INT:  PUSH ACC
               PUSH B
               MOV  A, B      ; 自B寄存器中取数
               (MOV P1, A)    ; 输出到P1口
               RL  A          ; 左环移一次
               MOV  B, A      ; 存回B
               POP B
               POP  ACC
               RETI           ; 中断返回

```

中断服
务程序

多外部中断源系统设计

定时器/计数器作为外部中断源的方法

- 51系列单片机中的定时器/计数器T0和T1可以扩展为外部中断源。
- 定时器/计数器T0和T1具有2个内部中断标志和外部计数输入引脚。当定时器设置为计数方式、工作在模式2（自动恢复常数方式）时，若计数初值设为0FFH，一旦外部信号从计数器引脚输入一个负跳变信号，计数器加1产生溢出中断。
- 可以把定时器/计数器T0和T1外部中断源做为边沿触发输入信号，接至定时器的T0（P3.4）或T1（P3.5）引脚上，利用该定时器的溢出中断标志及中断服务为外部中断源服务。

多外部中断源系统设计

定时器/计数器作为外部中断源的方法

```
ORG    0000H
AJMP   IINI    ; 跳到初始化程序
ORG    000BH
AJMP   T0_P

.....

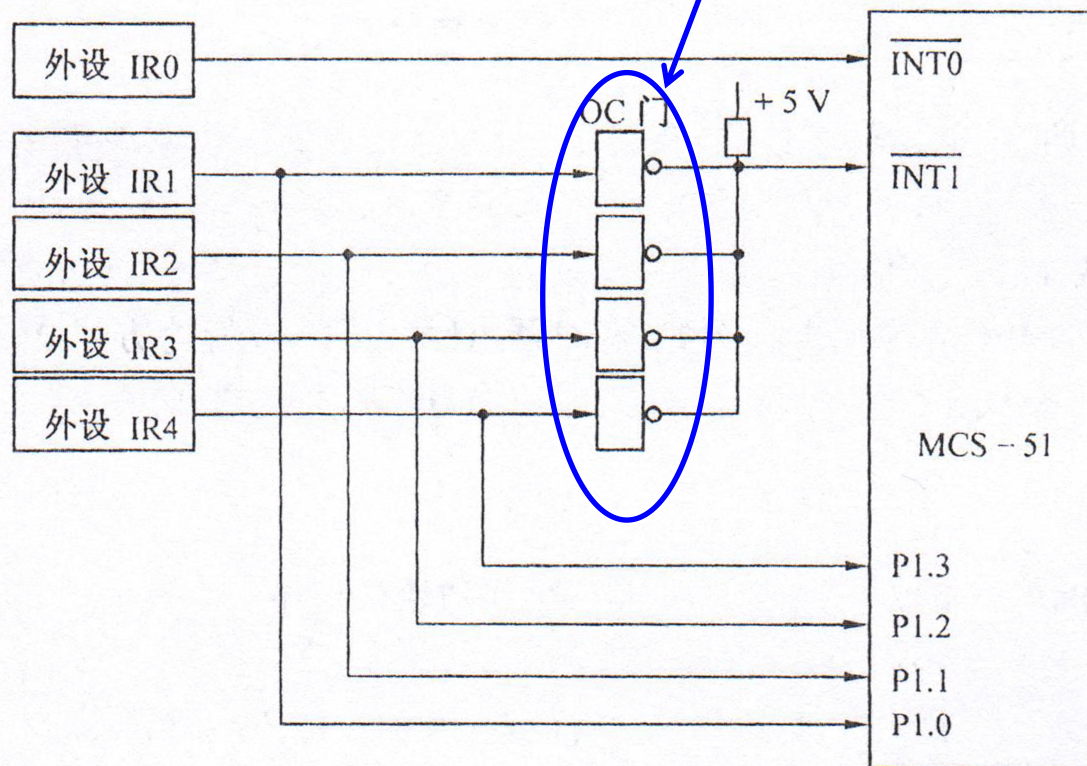
IINI:   MOV     TMOD, #06H    ;设置T0的工作方式寄存器
        MOV     TL0, #0FFH   ;设置计数器重装值
        MOV     TH0, #0FFH   ;设置计数器初值
        SETB    EA           ;CPU开中断
        SETB    ET0          ;允许T0中断
        SETB    TR0          ;启动T0计数
        .....
        SJMP    $

T0_P:   .....
```


多外部中断源系统设计

中断和查询结合的方法

- 外设IR0为高优先级中断，低电平或者负跳沿触发中断。
- 外设IR1~IR4为低优先级中断，高电平触发中断，并且能够通过软件撤除中断电平信号。
- 外设IR1~IR4都可引发中断，具体是哪一个中断源需通过中断服务程序中检查P1.0~P1.3上的电平信号判断。



多外部中断源系统设计

中断和查询结合的方法

```
ORG    0013H
LJMP   INT1
.....
ORG    0100H
INT1:  MOV    P1, #0FFH
        MOV    A, P1
        JB     ACC.0, IR1
        JB     ACC.1, IR2
        JB     ACC.2, IR3
        JB     ACC.3, IR4
INTIR:  RETI
```

```
IR1:    IR1的中断处理
        LJMP   INTIR
IR2:    IR2的中断处理
        LJMP   INTIR
IR3:    IR3的中断处理
        LJMP   INTIR
IR4:    IR4的中断处理
        LJMP   INTIR
```

在IR1~IR4中，IR1具有较高的同级中断响应优先权。

第5部分 MCS-51的中断系统

作业

第五部分 作业

提示：若用到相对跳转的指令，请注意指令的跳转范围。



- 一、MCS-51 有哪些中断源？各中断标志是如何产生的？又是如何清除的？
- 二、MCS-51 单片机响应外部中断的典型时间是多少？在哪些情况下，CPU 将推迟对外部中断请求的响应？
- 三、某系统有三个外部中断源 IR1、IR2、IR3，当某一中断源低电平时，便要求 CPU 进行处理，它们的优先处理次序由高到低为 IR3、IR2、IR1。三个外部中断源 IR1、IR2、IR3 对应的中断处理程序的入口地址分别为 1000H，1100H，1200H。试画出电路连接示意图，并编写主程序及中断服务子程序（转至相应的中断处理程序的入口即可）。
- 四、如下图所示，外部数据经 P1 口输入单元，每准备好一个数据，便发出选通信号，使触发器输出“1”，再经非门得“0”，输入至 INT0，向 CPU 发出中断请求，CPU 响应中断请求后，在中断处理程序中先撤除中断请求信号（通过 P3.0=0，使 INT0=1），再由 P1 口输入数据到单片机内部。请写出初始化程序和中断服务程序。

