



武汉纺织大学  
WUHAN TEXTILE UNIVERSITY

# 单片机原理及应用

课程性质：必修（考试） 学时：48 学分：3  
考核方式：闭卷考试



# 第五章 中断系统

- § 5.1 微机的输入/输出方式
- § 5.2 中断的概念
- § 5.3 89C51中断系统结构及中断控制
- § 5.4 中断响应及中断处理过程
- § 5.5 中断程序举例
- § 5.6 思考题与习题



## § 5.1 微机的输入/输出方式

§ 5.1.1 无条件传送方式

§ 5.1.2 查询传送方式

§ 5.1.3 直接存储器存取（DMA）方式



## § 5.1.1 无条件传送方式

- ▶ CPU总是认为外设在任何时刻都处于“准备好”的状态。
- ▶ 这种传送方式不需要交换状态信息，只需在程序中加入访问外设的指令，数据传送便可以实现。
- ▶ 此种方法很少使用。



## § 5.1.2 查询传送方式（条件传送）

- 一、什么是查询传送方式
- 二、查询方式程序流程图
- 三、查询方式的过程
- 四、查询方式的特点

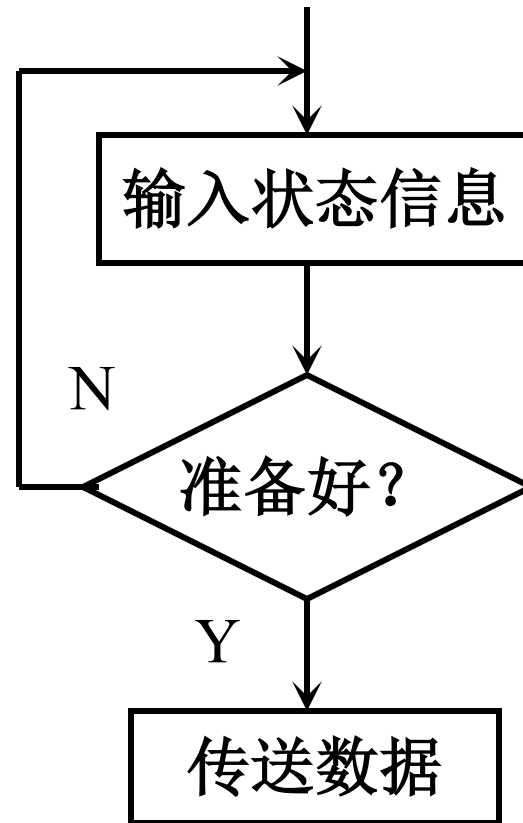


## 一、什么是查询传送方式

- 在输入时，需要查询外设的输入数据是否准备好；
- 在输出时，需要查询外设是否把上一次CPU输出的数据处理完毕。
- 查询传送方式：通过查询外设的状态信息，确信外设已处于“准备好”，微机才发出访问外设的指令，实现数据的传送。
- 状态信息：一般为1位二进制码。

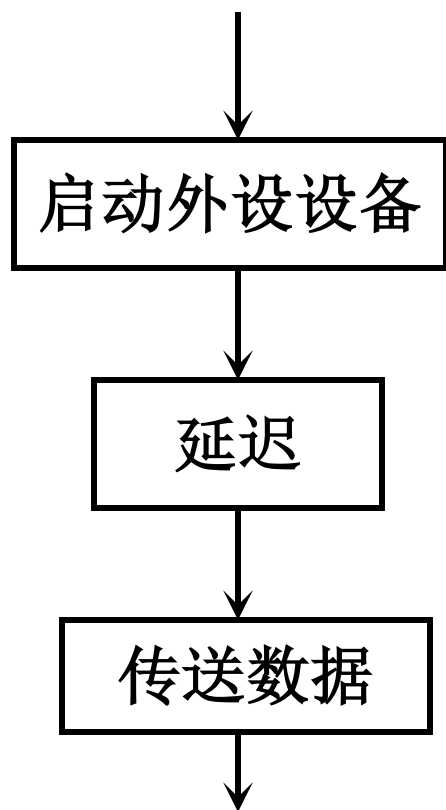


## 二、查询方式程序流程图





### 三、查询方式的过程



查询方式的过程:

查询——等待——数据传送，  
待到下一次数据传送时则重复  
上述过程。





## 四、查询方式的特点

- 优点：通用性好，可以用于各类外设和CPU间的数据传送。
- 缺点：CPU在完成一次数据传送后要等待很长时间才能进行下一次的传送。在等待过程中，CPU不能进行其他操作，所以效率比较低。



### § 5.1.3 直接存储器存取（DMA）方式

- **DMA: Direct Memory Access**
- **CPU让出数据总线（悬浮状态）**，使外设和存储器之间直接传送（不通过CPU）数据的方式。
- 适用于外设和存储器之间有大量的数据需要传送及外设工作速度很快的情况。



## § 5.2 中断的概念

- 一、中断系统
- 二、中断源
- 三、中断请求
- 四、中断响应过程
- 五、中断服务
- 六、中断返回
- 七、中断方式的特点

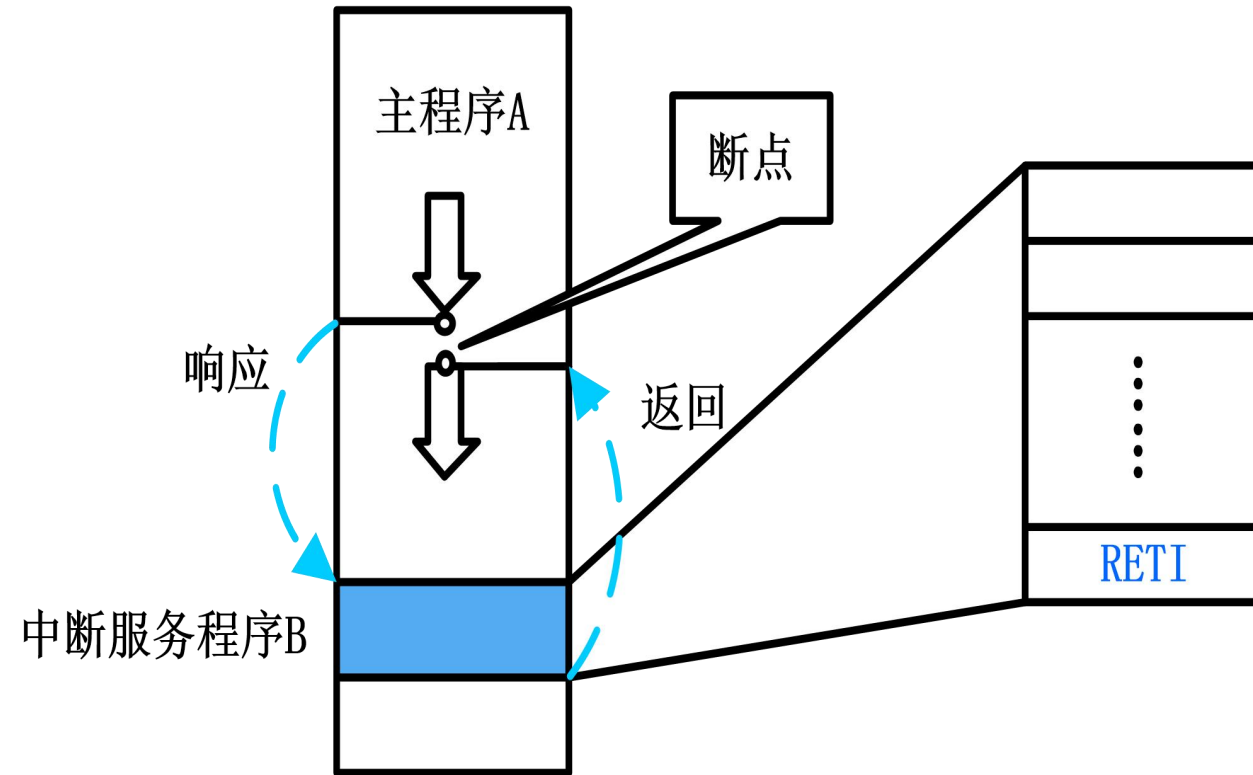


# 一、中断系统

- 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。
- 中断流程如图5-2所示。



图5-2 中断流程





## 一、中断系统

- 实现中断功能的部件称为**中断系统**，又称中断机构。



## 二、中断源

- 产生中断的请求源称为**中断源**。



### 三、中断请求

- 中断源向CPU提出的处理请求，称为中断请求或中断申请。





## 四、中断响应过程

- CPU暂时中止自身的事物，转去处理事件的过程，称为CPU的**中断响应过程**。



## 五、中断服务

- 对事件的整个处理过程，称为中断服务（或中断处理）。



## 六、中断返回

- 中断处理完毕，在返回到原来被中止的地方，称为中断返回。



## 七、中断方式的特点

- 1、中断方式消除了CPU在查询方式中的等待现象，大大提高了CPU的工作效率。
- 2、将从现场采集的数据通过中断方式及时传送给CPU，经过处理后可立即作出反应，实现现场控制。采用查询方式很难做到及时采集，实时控制。



## § 5.3 89C51中断系统结构及中断控制

---

§ 5.3.1 89C51中断源

§ 5.3.2 中断控制



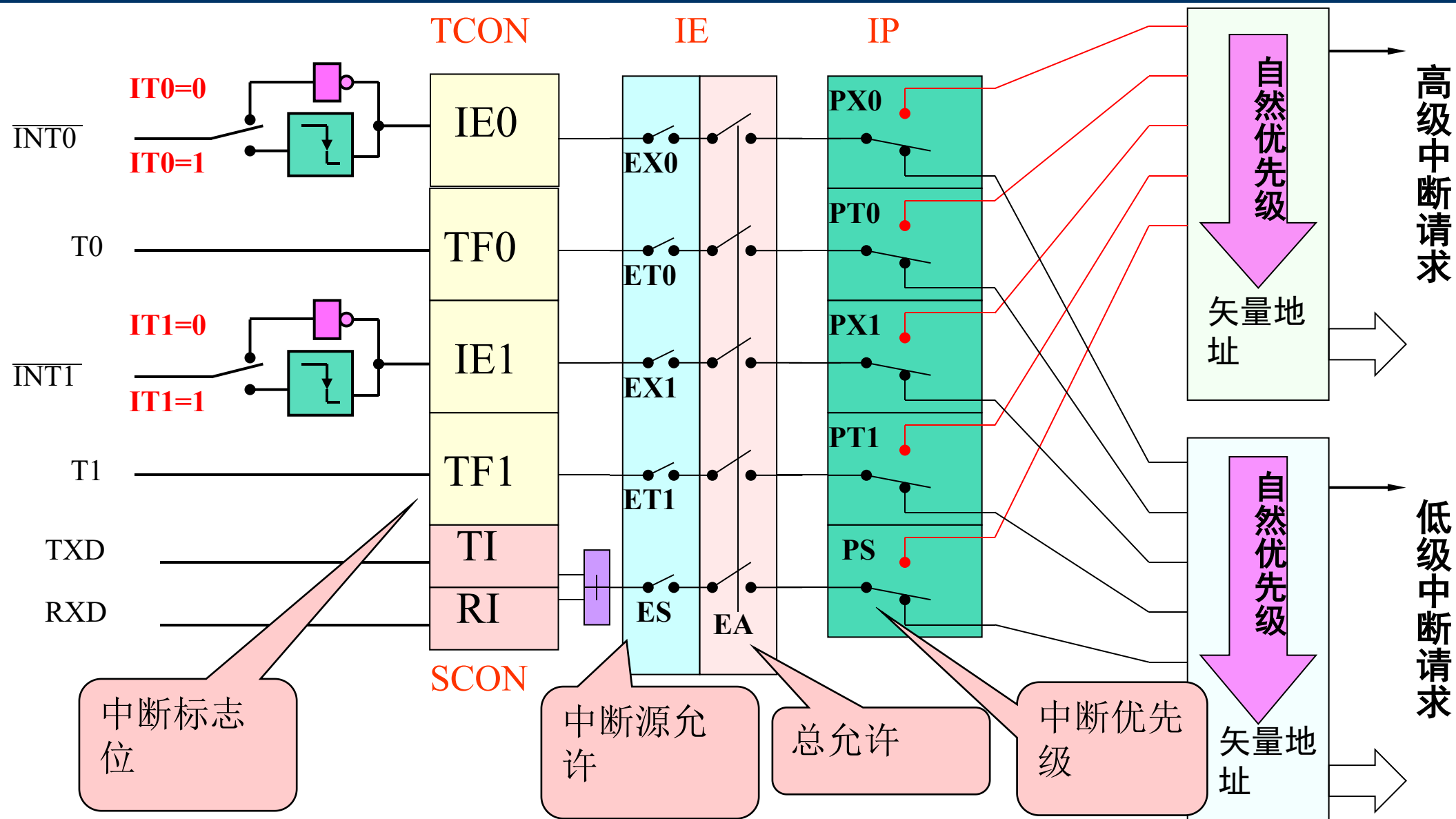
# 中断系统结构

---

- 89C51的中断系统的结构如图5-3所示。



图5-3 中断系统结构





# 1、分时操作

- 计算机的中断系统可以使CPU与外设同时工作。
- CPU在启动外设后，便继续执行主程序；而外设被启动后，开始进行准备工作。当外设准备就绪时，就向CPU发出中断请求，CPU响应该中断请求并为其服务完毕后，返回原来的断点处继续运行主程序。外设在得到服务后，也继续进行自己的工作。
- 因此，CPU可以使多个外设同时工作，并分时为各外设提供服务，从而提高了CPU的利用率和输入和输出的速度。





## 2、实时处理

- 当微机用于实时控制时，请求CPU提供服务时随机发生的。有了中断系统，CPU就可以立即响应并加以处理。



### 3、故障处理

- 微机在运行时往往会出现一些故障，如断电、存储器奇偶校验出错、运算溢出等。
- 有了中断系统，当出现上述情况时，CPU可及时转去执行故障处理程序，自行处理故障而不必停机。



## § 5.3.1 89C51中断源

---

一、89C51中断系统的五个中断源

二、通常情况的中断源



## 一、89C51中断系统的五个中断源

- 1、**INT0**——外部中断0请求，低电平有效。通过P3.2引脚输入。
- 2、**INT1**——外部中断1请求，低电平有效。通过P3.3引脚输入。
- 3、**T0**——定时器/计数器0溢出中断请求。
- 4、**T1**——定时器/计数器1溢出中断请求。
- 5、**TX/RX**——串行口中断请求。当串行口完成一帧数据的发送或接收时，便请求中断。



## 二、通常情况的中断源

1. I/O外设
2. 硬件故障
3. 实时时钟
4. 为调试程序而设定的中断源



## § 5.3.2 中断控制

- 一、中断请求标志
- 二、中断允许控制
- 三、中断优先级控制



## 一、中断请求标志

1、定时器控制寄存器TCON中的中断标志位

2、串行口控制寄存器SCON中的中断标志位



## 1、TCON中的中断标志位

- **TCON为定时器/计数器T0和T1的控制器，同时也锁存T0和T1的溢出中断标志及外部中断0和1的中断标志等。**
- 与中断有关的位如**图5-4**所示。

	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
TCON (88H)	TF1		TF0		IE1	IT1	IE0	IT0

- 各控制位的含义。





## 各控制位的含义

1、TF1：定时器/计数器T1溢出中断请求标志位。

当启动T1计数后，T1从初值开始加1计数，计数器最高位产生溢出时，由硬件使TF1置1，并向CPU发出中断请求。当CPU响应中断时，硬件将自动对TF1清0。

2、TF0：定时器/计数器T0溢出中断请求标志位。含义与TF1类同。

3、IE1：外部中断1的中断请求标志。INT1 (P3.3)。

当检测到外部中断引脚1 上存在有效的中断请求信号时，由硬件使IE1置1。

4、IE0：外部中断0的中断请求标志。INT0 (P3.2)。其含义与IE1类同。



## 各控制位的含义

### 5、IT1：外部中断1的中断触发方式控制位。

IT1=0时，外部中断1程控为电平触发方式。

CPU在每一个机器周期S5P2期间采样外部中断请求引脚的输入电平。

若外部中断1请求为低电平，则使IE1置1；若为高电平，则使IE1清0。

IT1=1时，外部中断1程控为边沿触发方式。

CPU在每一个机器周期S5P2期间采样外部中断请求引脚的输入电平。

如果在相继的两个机器周期采样过程中，一个机器周期采样到外部中断1请求引脚为高电平，接着的下一个机器周期采样到为低电平，则使IE1置1。直到CPU响应该中断时，才由硬件使IE1清0。

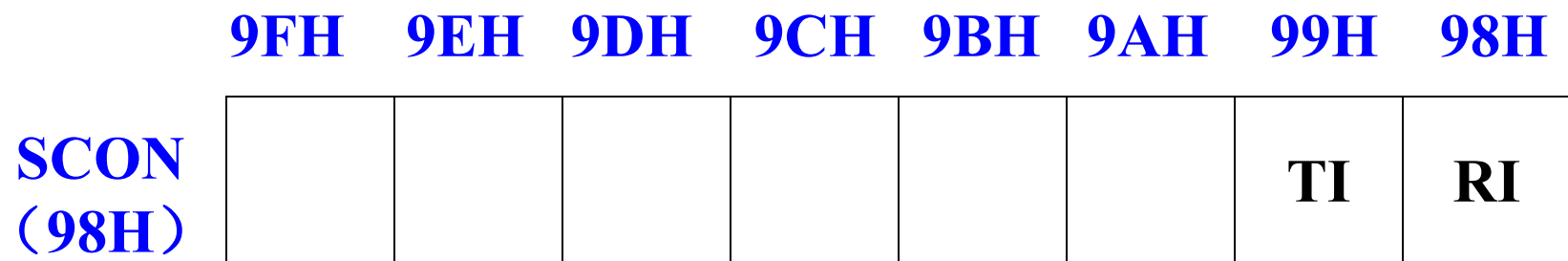
### 6、IT0：外部中断0的中断触发方式控制位。

其含义与IT1类同。



## 2、串行口控制寄存器SCON中的中断标志位

- **SCON**为串行口控制寄存器，其低2位锁存串行口的接收中断和发送中断标志**RI**和**TI**。
- **SCON**中**TI**和**RI**的格式如图5-5所示。



- 其控制位的含义



## 1、TI：串行口发送中断请求标志。

CPU将一个数据写入发送缓冲器SBUF时，就启动发送。每发送完一帧串行数据后，硬件置位TI。但CPU响应中断时，并不清除TI，必须在中断服务程序中由软件对TI清0。

## 2、RI：串行口接收中断请求标志。

在串行口允许接收时，每接收完一个串行帧，硬件置位RI。同样，CPU响应中断时不会清除RI，必须在中断服务程序中由**软件对RI清0**。



## 二、中断允许控制

### 1、中断允许寄存器IE的格式。

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

### 2、中断允许寄存器IE中各位的含义。



## 2、中断允许寄存器IE中各位的含义。

### ①EA—中断允许总控制位。

**EA=0**，屏蔽所有的中断请求；

**EA=1**，CPU开放中断。对各中断源的中断请求是否允许，还要取决于各中断源的中断允许控制位的状态。这就是所谓的两级控制。

### ②ES—串行口中断允许位。

**ES=0**，禁止串行口中断；

**ES=1**，允许串行口中断。

### ③ET1—定时器/计数器T1的溢出中断允许位。

**ET1=0**，禁止T1中断；

**ET1=1**，允许T1中断。



## 2、中断允许寄存器IE中各位的含义。

④EX1—外部中断1的溢出中断允许位。

EX1=0，禁止外部中断1中断； EX1=1，允许外部中断1中断。

⑤ET0—定时器/计数器T0的溢出中断允许位。

ET0=0，禁止T0中断； ET0=1，允许T0中断。

⑥EX0—外部中断0的溢出中断允许位。

EX0=0，禁止外部中断0中断； EX0=1，允许外部中断0中断。



例5-1：假设允许片内定时器/计数器中断，禁止其他中断。试根据假设条件设置IE的相应值。

**解：** (IE) = 10001010B = 8AH

**(a) 用字节操作指令**

**MOV IE, #8AH; 或 MOV A8H, #8AH;**

**(b) 用位操作指令**

**SETB ET0; 定时器/计数器0允许中断**

**SETB ET1; 定时器/计数器1允许中断**

**SETB EA ; CPU开中断**





## 三、中断优先级控制

- 1、89C51的中断优先级
- 2、中断优先级寄存器IP
- 3、中断优先级排列顺序
- 4、中断优先级控制实现的功能



# 1、89C51的中断优先级

89C51有两个中断优先级。

每个中断请求源均可编程为高优先级中断或低优先级中断。

中断系统中有两个不可寻址的“优先级生效”触发器，分别指出CPU正在执行的高、低优先级的中断服务程序。当其为1时则分别屏蔽所有的中断请求。



## 2、中断优先级寄存器IP

### 中断优先级寄存器IP的格式

				BCH	BBH	BAH	B9H	B8H
IP (B8H)				PS	PT1	PX1	PT0	PX0

### 中断优先级寄存器IP各控制位的含义



# 中断优先级寄存器IP各控制位的含义


- ①PS—串行口中断优先级控制位
- ②PT1—定时器/计数器T1中断优先级控制位
- ③PX1—外部中断1中断优先级控制位
- ④PT0—定时器/计数器T0中断优先级控制位
- ⑤PX0—外部中断0中断优先级控制位

若某控制位为1，则相应的中断源规定为高级中断；反之，为0，则相应的中断源规定为低级中断。



### 3、中断优先级排列顺序

- 当同时接收到几个同一优先级的中断请求时，响应哪个中断源则取决于内部硬件查询顺序。
- 其优先级顺序排列如图示。

中断源	同级内的
中断优先级	
外部中断0中断	 高级
定时器/计数器T0溢出中断	
外部中断1	
定时器/计数器T1溢出中断	
串行口中断	低级



## 4、中断优先级控制实现的功能

### (1) 按内部查询顺序排队

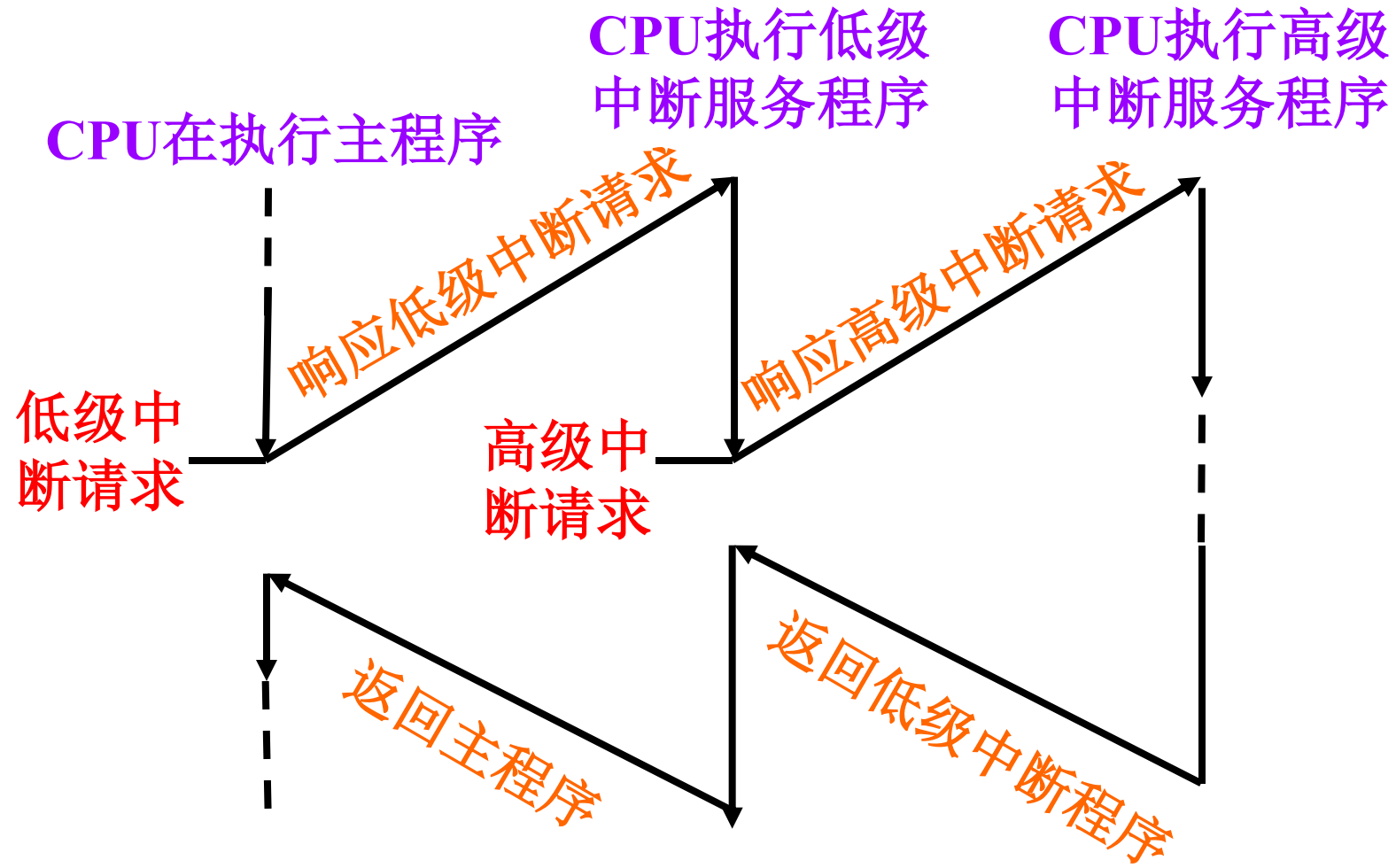
- 当数个中断源同时向CPU发出中断请求时，CPU根据设计者事先确定的中断源顺序号的次序，依次响应其中断请求。

### (2) 实现中断嵌套

- 当CPU正在处理一个中断请求时，又出现了另一个优先级比它高的中断请求，这时，CPU就暂时中止执行对原来优先级较低的中断源的服务程序，保护当前断点，转去响应优先级更高的中断请求，并为它服务。待服务结束，再继续执行原来较低级的中断服务程序。该过程称为中断嵌套。该中断系统称为多级中断系统。



图5-9 二级中断嵌套





例5-2： 设80C51的片外中断为高优先级，片内中断为低优先级。设置IP相应值。

解： (IP) = 00000101B = 05H

(a) 用字节操作指令

MOV IP, #05H; 或 MOV 0BH, #05H;

(b) 用为操作指令

SETB PX0

SETB PX1

CLR PS

CLR PT0

CLR PT1





## § 5.4 中断响应及中断处理过程

§ 5.4.1 中断响应

§ 5.4.2 中断处理

§ 5.4.3 中断返回

§ 5.4.4 关于具体的中断服务程序



# 中断处理过程的三个阶段

- 中断处理过程可分为：
- 中断响应
- 中断处理
- 中断返回
- 中断处理的大致流程如图5-10所示。

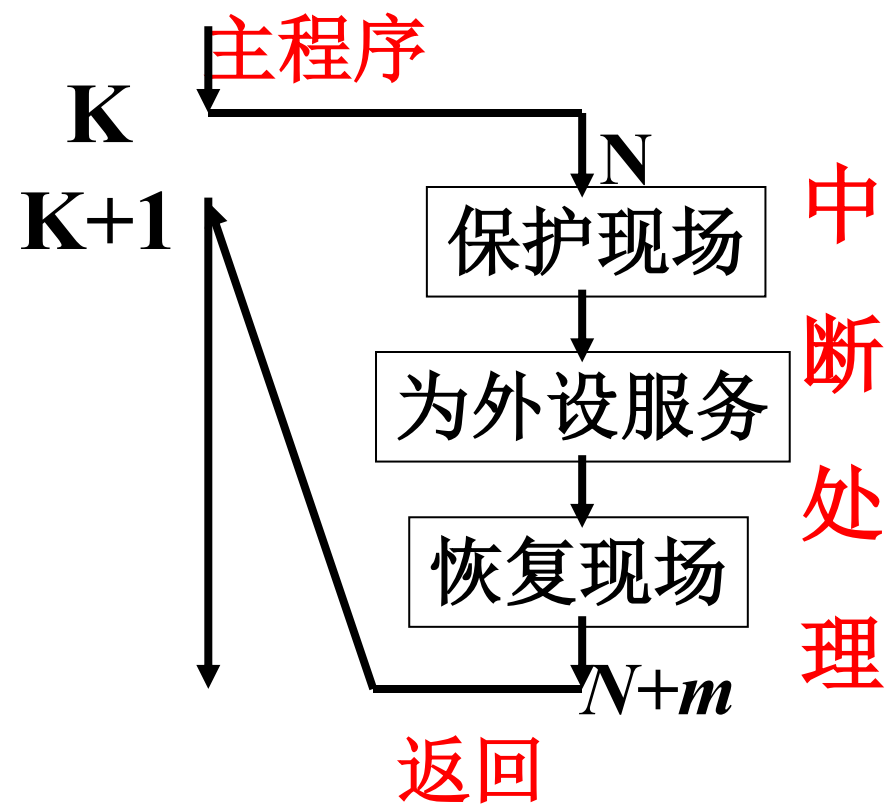


图5-10 中断处理的大致流程



## § 5.4.1 中断响应

- 一、中断响应条件
- 二、中断响应操作过程
- 三、中断响应时间



# 一、中断响应条件

- ①有中断源发出中断请求。
- ②中断总允许位EA=1，即CPU中断。
- ③申请中断的中断源的中断允许位为1，即中断没有被屏蔽。
- ④无同级或更高级中断正在被服务。
- ⑤当前的指令周期已经结束。
- ⑥若现行指令为RETI或是访问IE或IP指令时，该指令以及紧接着的另一条指令已执行完。



## 二、中断响应操作过程

### 1、中断响应操作过程

- (1) 把当前PC值压入堆栈，保护断点。
- (2) 将相应的中断服务程序的入口地址送入PC。
- (3) 对有些中断源，CPU会自动清除中断标志。
- (4) 执行中断服务程序。
- (5) 执行到返回指令RETI，中断服务程序结束，将堆栈内容弹出到PC，返回到原来断点继续执行。

### 2、各中断源及其对应的矢量地址

中断源	中断矢量
外部中断 0 (INT0)	0003H
定时器 T0 中断	000BH
外部中断 1 (INT1)	0013H
定时器 T1 中断	001BH
串行口中断	0023H



### 三、中断响应时间

- CPU不是在任何情况下都对中断请求予以响应，而不同的情况下对中断响应的时间是不同的。
- 对于顺利的中断响应，其最短的响应时间为3个机器周期。
- 对于受阻的中断响应，则响应时间会更长一些。
- 若中断系统只有一个中断源，则响应时间为3~8个机器周期之间。



## § 5.4.2 中断处理

- 一、什么是中断处理
- 二、中断处理包括的内容
- 三、编写中断服务程序时的注意问题



## 一、什么是中断处理

- CPU响应中断后即转入中断服务程序的入口，执行中断服务程序。从中断服务程序的第一条指令开始到返回指令为止，这个过程称为中断处理或中断服务。





## 二、中断处理包括的内容

- 不同的中断源服务的内容及要求各不相同，其处理过程也就有所区别。

一般情况，中断处理包括两部分内容：

**1、保护现场：**如在中断服务程序中要用到PSW、工作寄存器和SFR等寄存器时，则在进入中断服务之前应将它们的内容保护起来，在中断结束、执行RETI指令前应恢复现场。

**2、为中断源服务：**针对中断源的具体要求进行相应的处理。



### 三、编写中断服务程序时的注意问题

- ①在中断矢量地址单元处放一条无条件转移指令，使中断服务程序可灵活地安排在64KB ROM的任何空间。
- ②在中断服务程序中，应注意用软件保护现场，以免中断返回后丢失原来寄存器、累加器中的信息。
- ③若要在执行当前中断程序时禁止更高优先级中断，可以先用软件关闭CPU中断或禁止某中断源中断，在中断返回前再开放中断。



### § 5.4.3 中断返回

- 在中断服务程序中，最后一条指令必须为中断返回指令**RETI**。
- **CPU**执行该指令时，一方面清除中断响应时所置位的“优先级生效”触发器，另一方面从当前栈顶弹出断点地址送入程序计数器**PC**，从而返回主程序。
- 注意在中断服务程序中，**PUSH**和**POP**指令必须成对使用，否则，不能正确返回断点。



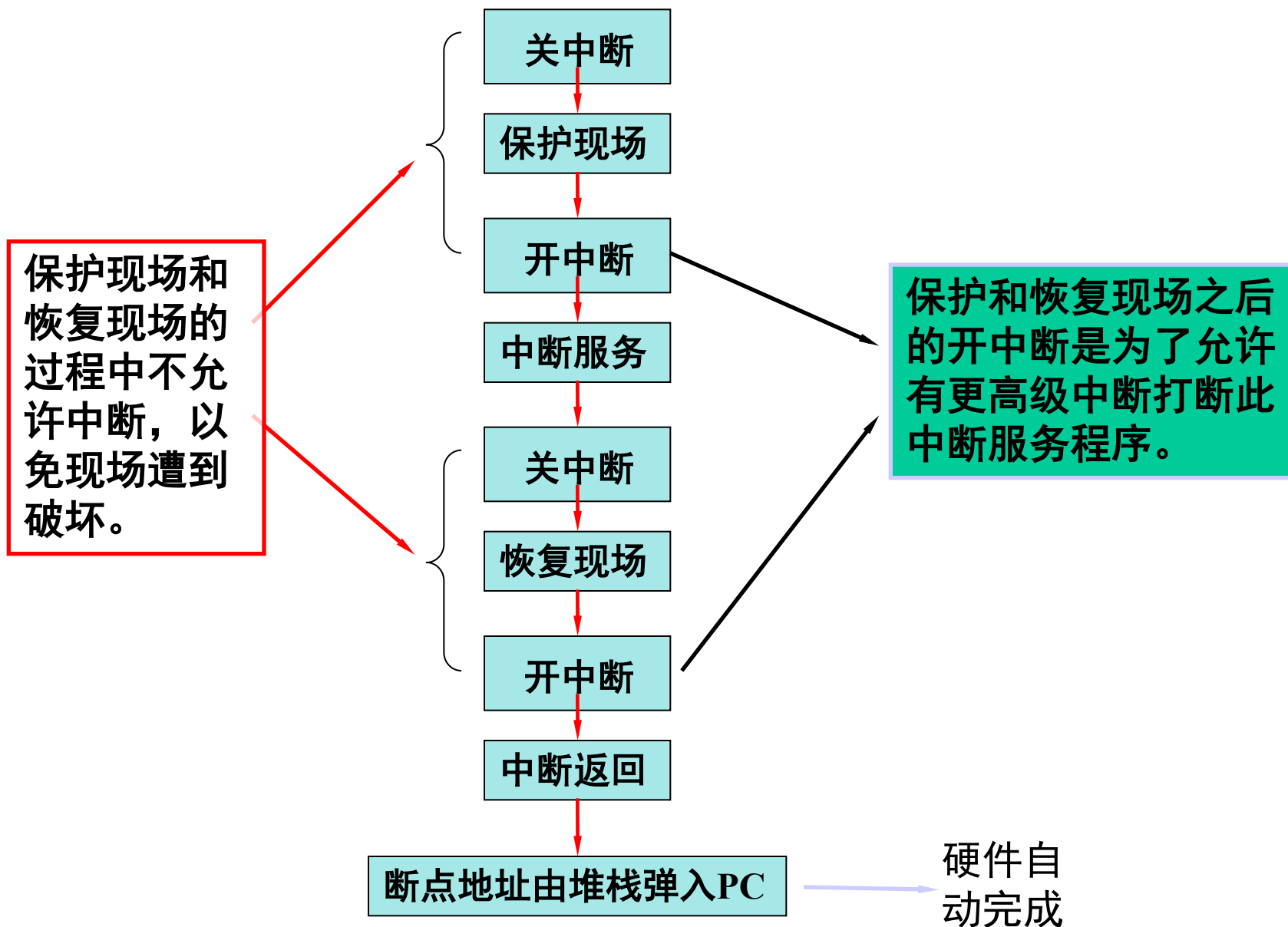
## § 5.4.4 关于具体的中断服务程序

- 中断服务程序从入口地址开始执行，直到返回指令“RETI”为止，包括两部分内容：
  - (1) 保护现场，
  - (2) 处理中断源的请求。



## § 5.4.4 关于具体的中断服务程序

### 中断服务程序





## § 5.4.4 关于具体的中断服务程序

**INTT0: PUSH ACC**  
**PUSH DPH**  
**PUSH DPL**  
**PUSH PSW**

保护现场

中断源服务

**POP PSW**  
**POP DPL**  
**POP DPH**  
**POP ACC**

恢复现场

**RETI**



## § 5.5 中断程序举例

§ 5.5.1 主程序

§ 5.5.2 中断服务程序

§ 5.5.3 中断服务程序举例



## § 5.5.1 主程序

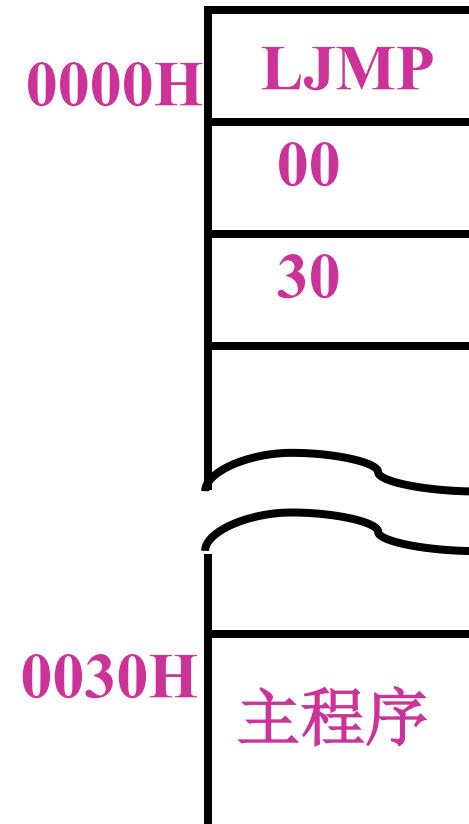
- 1、主程序的起始地址
- 2、主程序的初始化内容





## 1、主程序的起始地址

- 89C51系列单片机复位后，(PC) = 0000H
- 而0003H~0023H分别为各中断源的入口地址。
- 编程时应在0000H处写一条跳转指令（一般为长跳转指令），主程序是以跳转的目标地址作为起始地址开始编写，一般从003H开始，如图5-12所示。





## 2、主程序的初始化内容

- 初始化：将用到的内部部件或扩展芯片进行初始工作状态设定。
- 单片机复位后，特殊功能寄存器IE、IP内容均为00H，所以应对IE、IP进行初始化编程，以开放中断，允许某些中断源中断和设置中断优先级等。



## § 5.5.2 中断服务程序

- 当CPU接收到中断请求信号并予以响应后，CPU把当前的PC内容压入堆栈进行保护，然后转入响应的中断服务程序入口处执行。

1、中断服务程序的起始地址

2、中断服务程序编制中的注意事项



## 1、中断服务程序的起始地址

- 中断系统对五个中断源分别规定了各自的入口地址，但这些入口地址相距很近（8个字节）。
- 如中断服务程序的指令代码少于8个字节，则可从规定的中断服务程序入口地址开始，直接编写中断服务程序；
- 如中断服务程序的指令代码大于8个字节，则应采用与主程序相同的方法，在相应的入口处写一条跳转指令，并以跳转指令的目标地址作为中断服务程序的起始地址进行编程。
- 以INT0为例，中断矢量地址为0003H，中断服务程序从0200H开始。如图5-13所示。

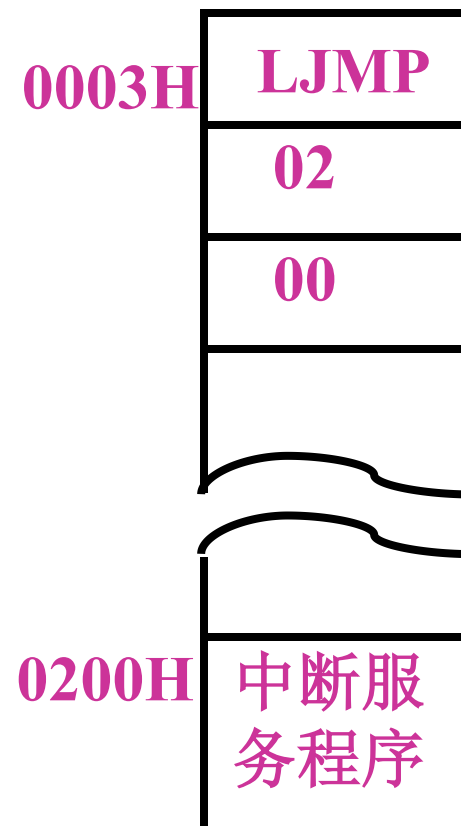


图5-13中断服务程序地址



## 2、中断服务程序编制中的注意事项

- ① 视需要确定是否保护现场。
- ② 及时清除那些不能被硬件自动清除的中断请求标志，以免产生错误的中断。
- ③ 中断服务程序中的压栈与弹栈指令必须成对使用，以确保中断服务程序的正确返回。
- ④ 主程序和中断服务程序之间的参数传递与主程序和子程序的参数传递方式相同。



## § 5.6.3 中断服务程序举例

例5-3 外部中断实验

例5-4 利用中断显示系统故障



### 例5-3外部中断实验:

如图5-14所示，将P1口的**P1.4~P1.7**作为**输入**位，**P1.0~P1.3**作为**输出**位。要求利用89C51将开关所设的数据读入单片机内，并依次通过P1.0~P1.3输出，驱动发光二极管，以检查P1.4~P1.7输入的电平情况（若输入为**低**电平则相应的LED亮）。

现要求采用中断边沿触发方式，每中断一次，完成一次读/写操作。

解:

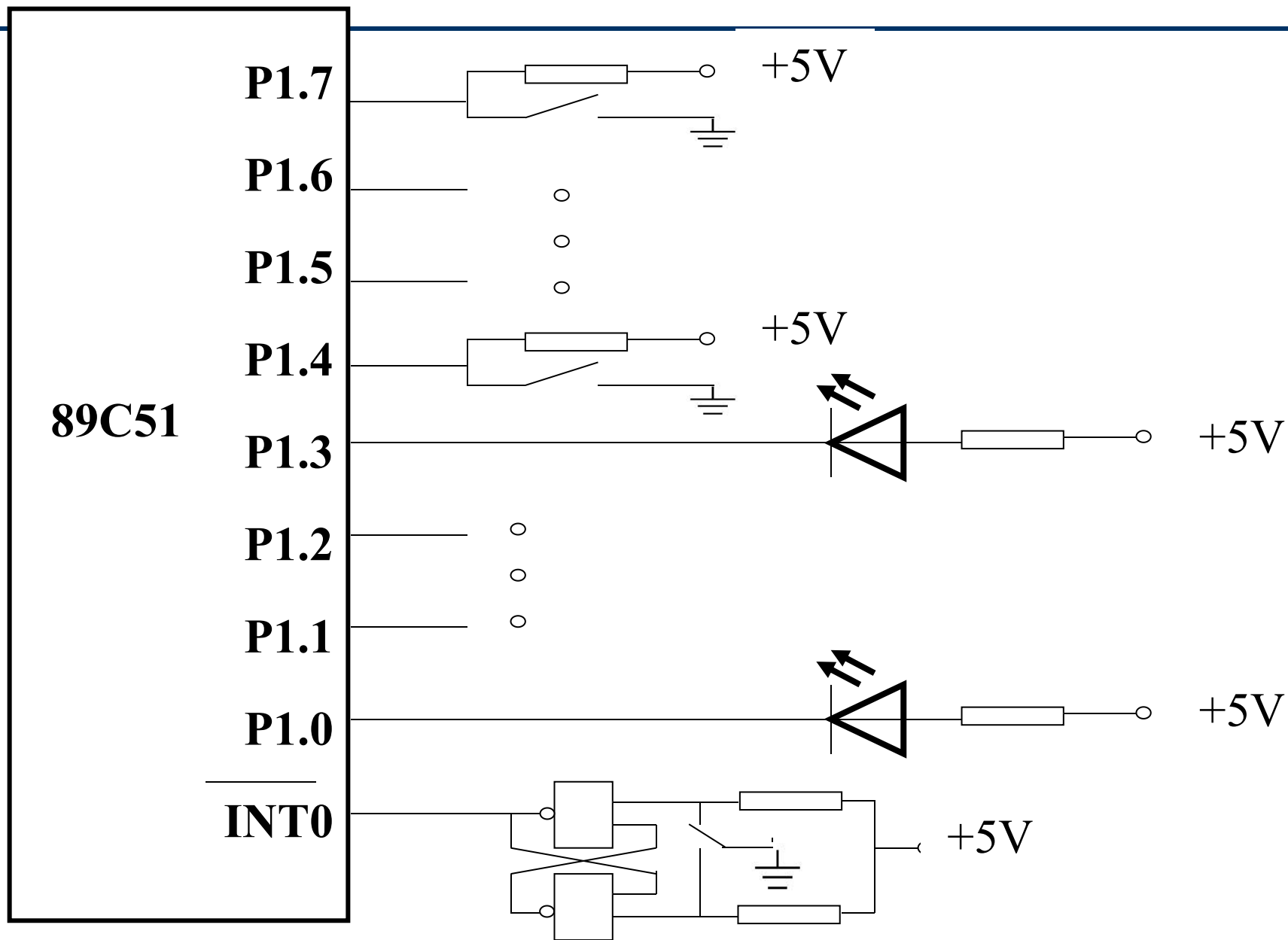


图 1-4 外部中断实验





**例5-3解：**如图所示，采用外部中断0，中断申请从INT0输入，并采用了去抖动电路。当P1.0～P1.3的任何一位输出为0时，相应的发光二极管就会发光。当开关S1闭合时，发出中断请求。中断服务程序的矢量地址为0003H。源程序如下。

源程序如下：

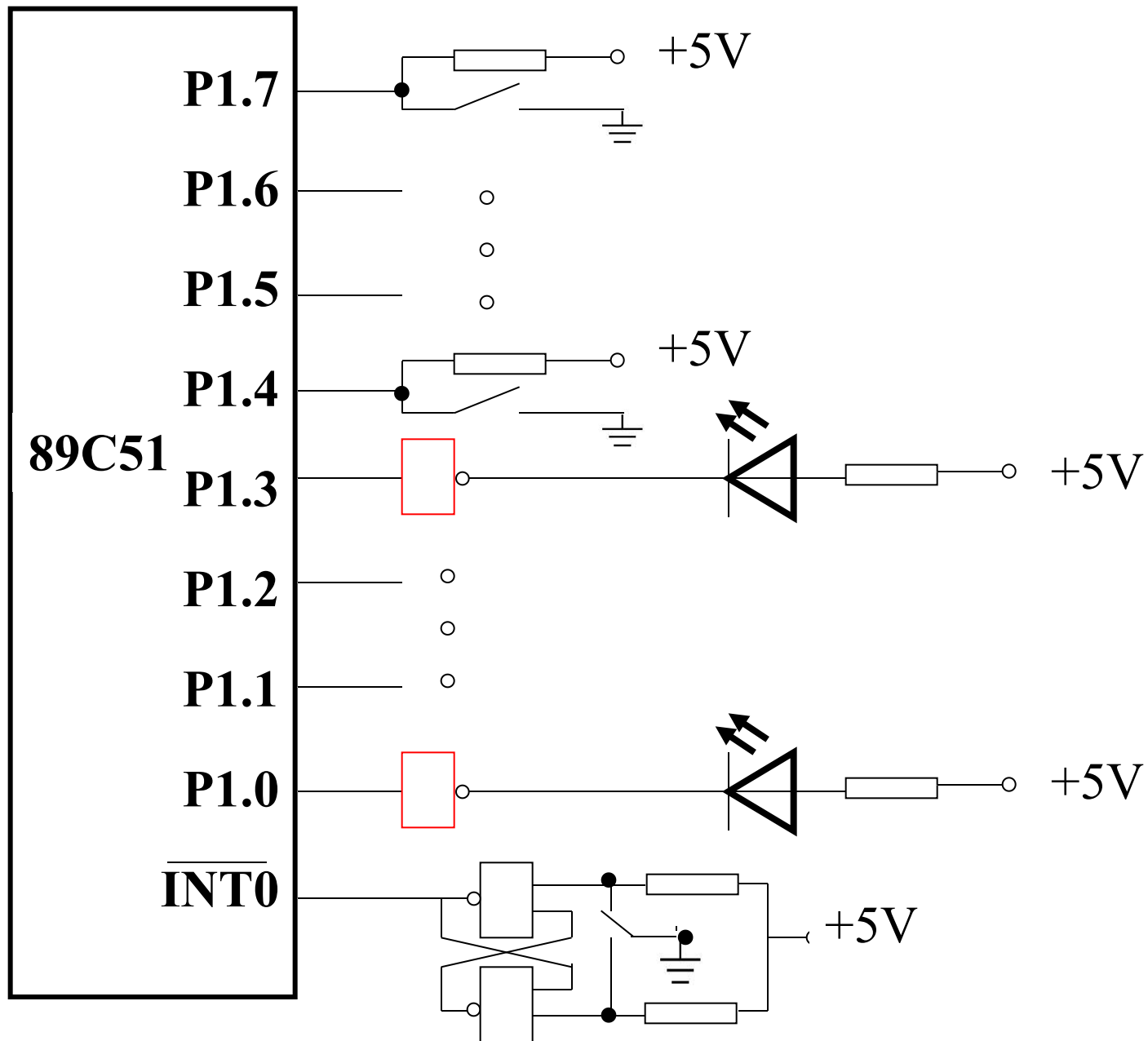
```
ORG    0000H
LJMP   MAIN    ; 上电，转向主程序
ORG    0003H   ; 外部中断0入口地址
LJMP   INSER   ; 转向中断服务程序
ORG    0030H   ; 主程序
MOV    SP #60H
MAIN:  SETB    EX0      ; 允许外部中断0中断
      SETB    IT0      ; 选择边沿触发方式
      SETB    EA       ; CPU开中断
      HERE:  SJMP   HERE ; ; 等待中断
```



**例5-3解：**如图所示，采用外部中断0，中断申请从INT0输入，并采用了去抖动电路。

当P1.0~P1.3的任何一位输出为1时，相应的发光二极管就会发光。当开关 S1 闭合时，发出中断请求。中断服务程序的矢量地址为0003H。源程序如下。

```
ORG 0200H    ; 中断服务程序  
INSERT: MOV A, #0F0H  
  
MOV P1, A    ; 设P1.4~P1.7为输入  
  
MOV A, P1    ; 取开关数  
  
SWAP A       ; A的高、低四位互换  
  
MOV P1, A    ; 输出驱动LED发光  
  
RETI         ; 中断返回  
  
END
```



## 外部中断实验

当P1.0~P1.3的任何一位输出为1时，相应的发光二极管就会发光。



## 例5-4 利用中断显示系统故障:

如图5-15所示，此中断电路可实现系统的故障显示。当系统的各部分正常工作时，四个故障源的输入均为低电平，显示灯全不亮。当有某个部分出现故障时，则相应的输入线由低电平变为高电平，相应的发光二极管亮。

解：



四个故障源

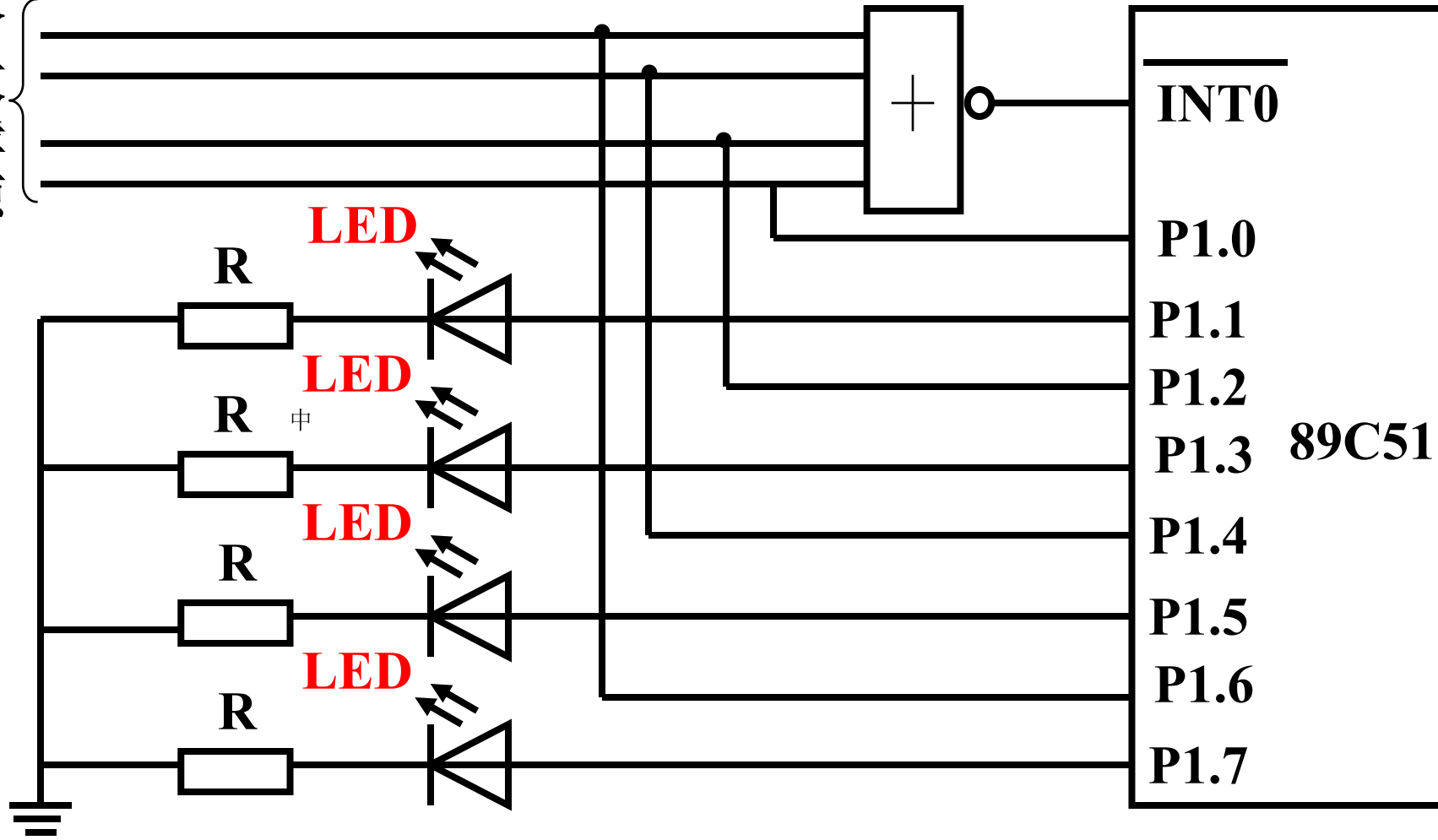


图5-15 利用中断显示系统故障



例5-4解：如图所示，当某一个故障信号输入线由低电平变为高点平时，会通过 INT0线引起89C51 中断（边沿触发方式）。在中断服务程序中，应将各故障源的信号读入，并加以查询，以进行相应的发光显示。

源程序如下：

```
ORG 0000H
LJMP MAIN; 上电，转向主程序
ORG 0003H; 外部中断0入口地址
LJMP INSERT; 转向中断服务程序
MOV SP #60H

MAIN: ANL P1,#55H
      ; P1.0,P1.2,P1.4,P1.6为输入
      ; P1.1,P1.3,P1.5,P1.7为输出
SETB EX0 ; 允许外部中断0中断
SETB IT0 ; 选择边沿触发方式
SETB EA ; CPU开中断

HERE: SJMP HERE ; 等待中断
```

```
INSERT: JNB P1.0, L1
        ; 查询中断源，（P1.0）=0，转L1
        SETB P1.1
        ; 是P1.0引起的中断，使相应的二极管亮

L1:     JNB P1.2, L2; 继续查询
        SETB P1.3 ;

L2:     JNB P1.4, L3
        SETB P1.5

L3:     JNB P1.6, L4;
        SETB P1.7

L4:     RETI
        END
```



## § 5.6 思考题与习题

- 1、什么是中断和中断系统？其主要功能是什么？
- 2、试编写一段对中断系统初始化的程序，使之允许 $\overline{\text{INT0}}$ ， $\overline{\text{INT1}}$ ，T0，串行口中断，且使T0中断为高优先级中断。
- 3、在单片机中，中断能实现哪些功能？
- 4、89C51共有哪些中断源？对其中端请求如何进行控制？
- 5、什么是中断优先级？中断优先处理的原则是什么？
- 6、说明外部中断请求的查询和响应过程。



- 7、89C51在什么条件下可响应中断？简述89C51单片机的中断响应过程。
- 8、在89C51内存中，应如何安排程序区？
- 9、试述中断的作用及中断的全过程。
- 10、当正在执行某一个中断源的中断服务程序时，如果有新的中断请求出现，试问在什么情况下可响应新的中断请求？在什么情况下不能响应新的中断请求？





## § 5.7 思考题与习题

11、89C51单片机有五个中断源，但只能设置两个中断优先级，因此，在中断优先级安排上受到一定的限制。试问以下几种中断优先顺序的安排（级别由高到低）是否可能：若可能，则应如何设置中断源的中断级别：否则，请简述不可能的理由。

- (1) 定时器0，定时器1，外中断0，外中断1，串行口中断。
- (2) 串行口中断，外中断0，定时器0，外中断1，定时器1。
- (3) 外中断0，定时器1，外中断1，定时器0，串行口中断。
- (4) 外中断0，外中断1，串行口中断，定时器0，定时器1。
- (5) 串行口中断，定时器0，外中断0，外中断1，定时器1。
- (6) 外中断0，外中断1，定时器0，串行口中断，定时器1。
- (7) 外中断0，定时器1，定时器0，外中断1，串行口中断。



## § 5.7 思考题与习题

- 12、简述89C51单片机的中断响应过程。
- 13、89C51单片机外部中断源有几种触发中断请求的方法？如何实现中断请求？
- 14、以中断方法设计单片机秒、分脉冲发生器。假定P1.0每秒产生一个机器周期的正脉冲， P1.1每分钟产生一个机器周期的正脉冲。



第五章结束