

第 6 章 MCS-51 中断系统.....	1
6.1 中断概述.....	1
6.1.1 中断概念.....	1
6.1.2 中断分类.....	2
6.1.3 中断优先权及CPU响应中断的原则	2
6.2 中断处理过程.....	3
6.2.1 中断请求.....	3
6.2.2 中断响应.....	3
6.2.3 中断处理.....	4
6.2.4 中断返回.....	4
6.3 MCS-51 单片机中断系统.....	4
6.3.1 中断源.....	5
6.3.2 中断的控制.....	5
6.3.4 中断处理过程.....	7
6.4 中断程序举例	9
6.4.1 主程序.....	9
6.4.2 中断服务程序.....	9
思考题与习题.....	10

第 6 章 MCS-51 中断系统

中断技术是计算机在实时处理和实时控制中不可缺少的一个很重要的技术,而中断系统使计算机能对外界随机发生的事件能够及时给予响应和处理。

6.1 中断概述

6.1.1 中断概念

1. 中断

所谓中断,是指当计算机执行正常程序时,系统中出现了某些急需处理的异常情况和特殊请求。这时 CPU 暂时中止现行程序,转去对随机发生的更紧迫事件进行处理(执行中断服务程序),待该事件处理完毕,CPU 自动地回到原来被中断的程序继续执行。这个过程称为中断。

2. 中断源及中断源种类

(1) 中断源

把能够向 CPU 发出中断请求的来源,称为中断源。它是引起 CPU 中断的原因。

(2) 中断源的种类

① 外部设备中断

在微型机系统中,输入/输出设备可成为中断源,称为设备中断。如键盘、打印机、A/D 转换器等。例如,主机要将处理的结果打印输出,须首先启动打印机,当打印机准备就绪后向主机发出中断请求,然后主机响应请求,打印一行信息。

② 硬件故障中断

在微型机系统中,硬件有时会发生故障,此时由硬件故障引起的中断成为故障中断源。例如,微型机使用的内存 RAM 是采用半导体存储器,当电源掉电时,信息将丢失。为了避免信息的丢失,在电源掉电时,向 CPU 请求中断,执行相应的服务程序,将一些重要的参数保存起来,并接入备用电源供电。

③ 外部事件中断

在微机控制系统中如有外部事件出现,例如在定时控制或定时数据采集系统中,由外部时钟电路定时,一旦达到规定的时间,时钟电路就立即向 CPU 提出中断申请,请求处理。

④ 程序性中断

一般在程序调试过程中设置的断点、单步操作等。

3. 中断系统

为实现中断功能而设置的各种硬件和软件统称为中断系统。

4. 中断的作用

(1) 实现并行操作

采用中断方式实现输入/输出操作时,可使 CPU 和外部设备(绝大部分时间内)并行工作。在 CPU 执行程序的过程中,当需要进行输入/输出操作时,CPU 启动相应的外部设备,然后 CPU 继续执行原来的程序。而此时,被启动的外部设备独立地进行自己的操作,当它需要与 CPU 进行数据交换时,向 CPU 发出中断请求,CPU 转去为设备进行中断服务。当中断服务完毕后又返回到断点处,继续执行下去。而外设也照样继续工作,这样就解决了快速的 CPU 与慢速的外设之间的矛盾,CPU 可以和多个外设同时工作,从而大大地提高了

CPU 的效率。

(2) 实现实时处理

所谓实时，就是指物理事件发生的真实时间。实时处理，就是指计算机对外来信号的响应要及时，否则将丢失信息，产生错误的处理。计算机在用于实时控制时，现场的各种参数、状态信息发生异常情况时，均可发出中断请求，要求 CPU 及时进行处理。

(3) 故障处理

在计算机运行过程中，有时会出现一些事先无法预料的情况或故障，如电源掉电、运算溢出、传输错误等，此时可利用中断进行相应的处理而不必停机。

6.1.2 中断分类

1. 根据中断是否可屏蔽（禁止），分为不可屏蔽中断和可屏蔽中断

前者是不能被程序禁止的中断，当 CPU 检测到有不可屏蔽中断请求时，它一定要响应，这是为一些特别重要的中断请求而设置的一种中断方法；后者是可被程序禁止的中断，这是为一般的中断请求而设置的。在开中断条件下允许 CPU 响应该中断请求；在关中断条件下禁止 CPU 响应该中断请求。在 MCS-51 系列单片机中，中断均为可屏蔽中断，没有不可屏蔽中断。

2. 根据中断源的位置不同可分为内中断和外中断

内中断是由 CPU 或单片机内部中断源提出的中断请求；外中断是由 CPU 或单片机外部中断源提出的中断请求。

3. 根据产生中断的原因不同，分为软件中断和硬件中断

所谓软件中断是指人们可以通过相应的指令使 CPU 响应中断，在 Intel8086 / 8088CPU 中内部中断即为软件中断；而外部中断即为硬件中断。

6.1.3 中断优先权及 CPU 响应中断的原则

1. 中断优先权及其确定原则

在微型机应用系统中通常有多个中断源，当多个中断源同时向 CPU 申请中断时，要求 CPU 既能区分各个中断的请求，又能确定首先为哪一个中断服务。为了解决这一问题，首先确定中断服务的次序，其原则是把这些如不及时处理将造成严重后果的中断请求排在最先处理的位置；把那些仅要求在一定的时间范围内进行处理的请求排在其次处理位置；而把那些对处理时间没有严格要求的请求排在最后处理位置。给中断源指定处理的次序就是给中断源确定中断优先权（或称中断优先级）。

2. CPU 响应中断的顺序

在中断源较多的微型机应用系统中，由于中断情况复杂，CPU 如何安排响应中断请求的顺序呢，经分析不外乎以下几种：

(1) 当不同优先级的中断源同时申请中断时：此时 CPU 响应中断请求的顺序为，从优先级高的中断源到优先级低的中断源。

(2) 当相同优先级的中断源同时申请中断时：此时 CPU 按事先规定的顺序进行处理。例如：在 MCS-51 系列单片机中规定在相同优先级条件下，CPU 响应及处理的顺序如下：外部中断 0、T0 溢出中断、外部中断 1、T1 溢出中断、串行口中断。

(3) 当 CPU 正在处理某中断，又有中断源提出中断请求时，后提出中断请求的中断源的优先级高，此时在开中断的条件下 CPU 暂停现在执行的中断服务程序而转去为优先级高的中断源服务，服务完毕就返回原服务程序继续执行下去。后提出中断请求的中断源的优先

级等于或低于正在为其服务的中断源的优先级时，此时 CPU 不理睬后提出的中断请求，仍执行原来的中断服务程序，当处理完原中断请求后再响应此中断请求。

6.2 中断处理过程

在微型机系统中，一个完整的中断处理过程应包括：中断请求、中断允许控制、中断优先级排队、中断响应、中断处理及中断返回 6 个步骤。因此，中断处理过程中所要解决的基本问题是：

- 中断源怎样向 CPU 申请中断？CPU 又如何检测中断请求信号？
- 对中断源的中断请求，CPU 是不是都必须响应？怎样对它们进行控制？
- 当多个中断源同时申请中断时，CPU 如何确定服务对象？怎样实现中断优先级排队及高级中断源中断低级中断处理的功能？
- CPU 怎样响应中断，怎样转入中断服务程序的入口地址？又怎样从中断服务程序返回？

6.2.1 中断请求

中断请求就是中断源向 CPU 发出的请求中断的要求。例如：单片机的中断既可由外部中断源产生，也可由内部中断源产生，不同的中断源产生中断的条件也不尽相同。如外部设备中断源向 CPU 发出中断请求必须满足下列两个条件：

1. 外部设备准备就绪

当执行输入操作时，输入设备接口中的数据寄存器已装入数据，当执行输出操作时，输出设备接口中的数据寄存器已空，这时外设便向 CPU 提出中断请求。

2. 中断未被屏蔽

在实际系统中往往有多个中断源，为了增加控制的灵活性，通常每个中断源还设置了一个中断允许触发器（外设中断请求的标志），只有当该触发器为 1 时，中断源的中断请求信号才能送到 CPU。否则，即使中断源有请求，也不能向 CPU 发出请求信号。这时称中断被屏蔽。

6.2.2 中断响应

1. 中断优先级排队

一个 CPU 通常可以和多个中断源相连，某一瞬间可能会发生 2 个或 2 个以上中断源同时请求中断的情况，这就要求按轻重缓急给每一个中断源的中断请求赋予一个中断优先级。这样当多个中断源同时向 CPU 请求中断时，CPU 就可以通过中断优先级排队电路率先响应中断优先级高的中断请求而把中断优先级低的中断请求暂时搁置起来，等到处理完优先级高的中断请求后再来响应优先级低的中断。

2. 中断响应

经 CPU 排队后，即响应其中最高优先级的中断请求，先将 PC 当前值压入堆栈保存（即断点入栈），以备返回原程序（即被中断的程序）。接着将相应中断处理程序的入口地址（中断处理程序的首地址）送 PC，于是转入该处理程序的入口，进行相应的处理。

6.2.3 中断处理

1. 保护现场

在执行中断处理程序时，首先应将在中断处理程序中要使用的累加器、通用寄存器等入栈保护，进行现场保护。有的机器现场保护是由硬件自动完成的，有的机器则需用软件实现现场保护。在保护现场的过程中，应关中断（禁止中断）。当现场保护好后应开中断（允许中断）。

2. 中断服务

中断服务程序要根据具体情况编制。通常，在中断服务时允许 CPU 响应优先级比其高的中断请求。

3. 恢复现场

在中断服务结束后，应立即关中断，以保证在恢复现场过程中不受干扰。恢复现场即把原来压入堆栈的、标志寄存器、累加器、通用寄存器等内容弹出至相应部件。现场恢复后，应立即开中断，以便响应更高级的中断请求。

6.2.4 中断返回

在中断处理程序中最后一条指令必须是 RETI（返回主程序指令），于是硬件自动将存放在堆栈中的断点地址弹出至 PC，使 CPU 继续执行原来的程序。

6.3 MCS-51 单片机中断系统

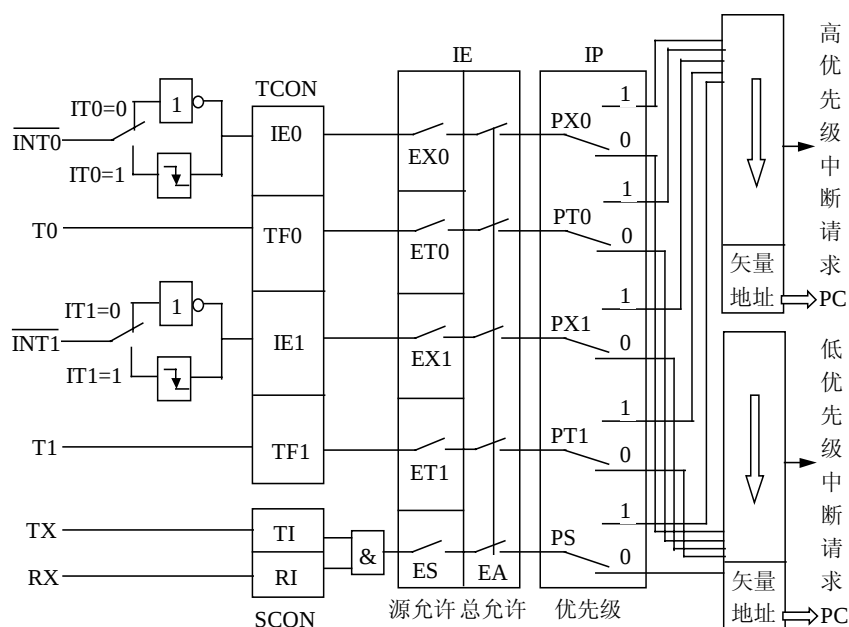


图 6-1 MCS-51 中断系统结构框图

MCS-51 单片机的中断系统是 8 位单片机中功能较强的一种，可以提供 5 个中断源，具有 2 个中断优先级，可实现 2 级中断服务程序嵌套。MCS-51 单片机的中断系统结构示意图如图 6-1 所示。它有 4 个用于中断控制的寄存器 IE、IP、TCON 和 SCON，用来控制中断的类型、中断的开 / 关和各种中断的优先级别。

6.3.1 中断源

8031 单片机的 5 个中断源为：

- INT0——外部中断 0 请求。通过 P3.2 引脚输入，中断标志 IE0。
- INT1——外部中断 1 请求。通过 P3.3 引脚输入，中断标志 IE1。
- T0——定时器 / 计数器 0 溢出中断请求，中断标志 T0。
- T1——定时器 / 计数器 1 溢出中断请求，中断标志 T1。
- TX / RX ——串行口中断请求，TX 为串行口发送中断请求，RX 为串行口接收中断请求。串行口接收中断标志 RI，串行口发送中断标志 TI。

6.3.2 中断的控制

8031 中断系统有以下 4 个特殊功能寄存器：

- 定时器控制寄存器 TCON（用六位）
- 串行口控制寄存器 SCON（用两位）
- 中断允许寄存器 IE
- 中断优先级寄存器 IP

其中，TCON 和 SCON 只有一部分位用于中断控制。通过对以上各特殊功能寄存器的各位进行置位或复位等操作，可实现各种中断控制功能。

1. 中断请求标志

(1) TCON 中的中断标志位

TCON 为定时器 / 计数器 T0 和 T1 的控制寄存器，同时也锁存 T0 和 T1 的溢出中断标志及外部中断 0 和外部中断 1 的中断标志等。与中断有关的位如图 6-2 所示。

位地址	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
TCON(88H)	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

图 6-2 TCON 中的中断标志位

各标志位的含义如下：

TF1——定时器 / 计数器 T1 的溢出中断请求标志位。当启动 TI 计数以后，T1 从初值开始计数，计数器产生溢出时，由硬件使 TF1 置 1，并向 CPU 申请中断，当 CPU 响应中断时，硬件将自动对 TF1 清零。

TF0——定时器 / 计数器 T0 的溢出中断请求标志位。含义同 TF1。

IE1——外部中断 1 的中断请求标志。当 CPU 检测到外部中断引脚 1 有中断请求时，由硬件将 IE1 置 1。当 CPU 转向其中断处理程序时，由硬件使 IE1 清零。

IT1——外部中断 1 的中断触发方式控制位。

IT1=0 时，外部中断 1 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 1 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；若外部中断 1 请求为高电平，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 1 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 请求为高电平，紧接下一个机器周期采样到外部中断 1 请求为低电平，则使 IE1 置 1。直到 CPU 响应该中断时，才由硬件使 IE1 清 0。

IE0 ——外部中断 0 的中断请求标志。含义与 IE1 相同。

IT0——外部中断 0 的中断触发方式控制位。含义与 IT1 相同。

TR1、TR0 定时/计数器 T1、T0 启停位。

(2) 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 中的 TI 和 RI 的格式如图 6-3 所示。

位地址	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
SCON(98H)	SM0	SM1	SM2	REN	TB8	RB8	TI	RI

图 6-3 SCON 中的中断标志位

各控制位的含义如下:

TI——串行口发送中断标志。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,由中断系统的硬件自动将 TI 置 1。但 CPU 响应中断时,并不清除 TI,必须在中断处理程序中用指令将 TI 清 0。

RI——串行口接收中断标志。在允许串行口接收时,每接收完一个字符后,中断系统的硬件自动将 RI 置 1,同样,CPU 响应中断处理程序时并不自动将 RI 复位,必须用指令将其清 0。

SCON 中其余位用于串行口方式设定和串行口发送/接收控制。

2. 中断允许控制

8051 对中断源的开放或屏蔽是由中断允许寄存器 IE 控制的。IE 的格式如图 6-4 所示。

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

图 6-4 中断允许控制位

中断允许寄存器 IE 对中断的开放和关闭实现两级控制。所谓两级控制,就是有一个总的开关中断控制位 EA (IE.7),当 EA=0 时,屏蔽所有的中断申请,即任何中断申请都不接受;当 EA=1 时,CPU 开放中断,但 5 个中断源还要由 IE 的低 5 位的各对应位的状态进行中断允许控制(见图 6-1)。IE 中各位的含义如下:

EA——中断允许总控制位。EA=0 时,CPU 屏蔽所有的中断请求;EA=1 时,CPU 开放中断。此时每个中断源的中断是否允许,还要取决于各中断源的中断允许控制位的状态。

ES——串行口中断允许位。ES=1,允许串行口中断;否则禁止其中断。

ET1——定时器 / 计数器 T1 的溢出中断允许位。ET1=1,允许定时器 / 计数器 T1 溢出时提出中断请求;否则,禁止其中断。

EX1——外部中断 1 中断允许位。EX1=1,允许外部中断 1 中断;否则,禁止其中断。

ET0——定时器 / 计数器 T0 的溢出中断允许位。ET0=1,允许定时器 / 计数器 T0 溢出时提出的中断请求;否则,禁止其中断。

EX0——外部中断 0 中断允许位。EX0=1,允许外部中断 0 中断;否则,禁止中断。

MCS-51 单片机复位后将 IE 寄存器清 0,所以单片机是处于关中断的状态。若要开放中断,必须使 EA 位为 1 且相应中断允许位也为 1。开、关中断既可使用位操作指令,也可使用字节操作指令实现。

[例 6-1] 假设允许片内定时器 / 计数器中断,禁止其他中断。试根据假设条件置 IE 的相应值。

解:方法 1

用字节操作指令 MOV IE, #8AH 或 MOV A8H, #8AH

方法 2

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

3. 中断优先级控制

MCS-51 单片机有两个中断优先级，即高优先级和低优先级。每个中断源的优先级由 IP 的状态决定，通过对中断优先级寄存器 IP(字节地址为 B8H)赋值来设定各个中断源的优先级或为高优先级或为低优先级。IP 的格式如图 6-5 所示，各位的意义如下：

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

图 6-5 中断优先级寄存器 IP 的控制位

IP 中的低 5 位为各中断源优先级的控制位，可用软件来设置。各位的含义如下：

PS——串行口中断优先级控制位。PS=1，串行口指定为高中断优先级；否则，为低中断优先级。

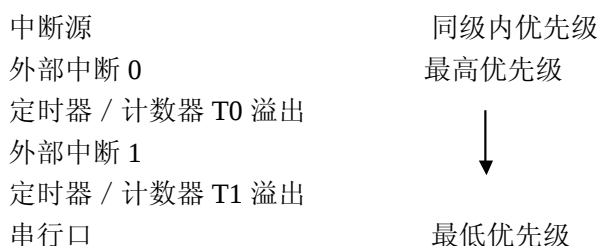
PT1——定时器 / 计数器 T1 中断优先级控制位。PT1=1，定时器 / 计数器 T1 指定为高中断优先级；否则，为低中断优先级。

PX1——外部中断 1 中断优先级控制位。PX1=1，外部中断 1 指定为高中断优先级；否则，为低中断优先级。

PT0——定时器 / 计数器 T0 中断优先级控制位。PT0=1，定时器 / 计数器 T0 指定为高中断优先级；否则，为低中断优先级。

PX0——外部中断 0 中断优先级控制位。PX0=1，外部中断 0 指定为高中断优先级；否则，为低中断优先级。

在 MCS-51 单片机中，当两个不同优先级的中断源同时提出中断请求时，CPU 先响应优先级高的中断请求，后响应优先级低的中断请求，当几个同级的中断源同时提出中断请求时，CPU 将按如下的顺序依次响应。



当 CPU 正在执行一个低优先级中断处理程序时，它会被高优先级的中断源所中断，但不会被同级中断源所中断。

[例 6-2] 设 8031 的片外中断为高优先级，片内中断为低优先级。试设置 IP 相应值。

解：用位操作指令

SETB PX0

SETB PX1

CLR PS

CLR PT0

CLR PT1

6.3.4 中断处理过程

中断处理过程可分为三个阶段，即中断响应、中断处理和中断返回。

1. 中断响应的条件

- (1) 有中断源发出中断请求。
- (2) 中断总允许位 EA=1，即 CPU 开中断。

- (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) 若先行指令为 RETI 或者是访问 IE 或 1P 指令时，该指令以及紧接着的另一条指令已执行完。

2. 中断响应操作过程

CPU 执行程序的过程中，在每个机器周期的 S5P2 期间顺序采样每个中断源，这些采样值在下一个机器周期内，将按优先级或内部顺序依次查询，若查询到某个中断标志为 1，将在接下来的机器周期 S1 期间按优先级进行中断处理。中断系统通过硬件自动将响应的中断矢量地址装入 PC，以便进入响应的中断服务程序。

响应中断时首先把被中断程序的断点压入堆栈，然后执行一条由中断系统提供的硬件 LCALL 指令，即程序转至相应中断处理程序入口。5 个中断源相应中断处理程序入口地址如表 6-1 所示。从表可知，两相邻的中断处理程序入口地址的间隔为 8 个单元。也就是说，若要在其中存放相应的处理程序，其长度不得超过 8 个字节。通常，中断处理程序的长度不止 8 个字节。此时，可以在相应的中断处理程序入口地址中放一条长跳转指令 LJMP，这样，中断处理程序的长度就不受 8 个字节的限制了。

表 6-1 MCS-51 单片机中断处理程序入口地址表

中 断 源	中断处理程序入口地址
外部中断源 0	0003H
T0 溢出中断	000BH
外部中断源 1	0013H
T1 溢出中断	001BH
串行口（接收/发送）	0023H

3. 中断处理

CPU 响应中断后即转至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。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。一般情况下，中断处理包括两部分内容：一是保护现场，二是为中断源服务。

所谓现场，是指被中断的程序中使用了 PSW、工作寄存器和 SFR 等，如果在中断服务程序中也要使用这些寄存器，则在中断服务程序前应将它们的内容保护起来（保护现场），在中断结束，执行 RETI 指令前应恢复现场。

中断服务是针对中断源的具体要求进行相应的处理。我们在编写中断服务程序时应注意：

- (1) 中断程序入口处放一条无条件转移指令(如 LJMP XXXXH)，使中断服务程序可灵活地安排在 64KB 程序存储器的任何空间。
- (2) 在中断服务程序中，用户应注意用软件保护现场，以免中断返回后丢失原寄存器、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，在中断返回前再开放中断。
- (4) 在保护现场和恢复现场时，为了不使现场信息受到破坏或造成混乱，一般应关闭 CPU 中断，使 CPU 暂不响应新的中断请求。

4. 中断返回

当某一中断源发出中断请求时，CPU 能决定是否响应这个中断请求。若响应此中断请求，CPU 必须在现行（假设）第 K 条指令执行完后，把断点地址（第 K+1 条指令的地址）

即现行 PC 值压入堆栈中保护起来（保护断点）。当中断处理完后，再将压入堆栈的第 K + 1 条指令的地址弹到 PC（恢复断点）中，程序返回到原断点处继续运行。

6.4 中断程序举例

中断程序的结构及内容与 CPU 对中断的处理过程密切相关，通常分为两大部分。

6.4.1 主程序

1. 主程序的起始地址

MCS-51 系列单片机上电或复位后，(PC) = 0000H，而 0003H~002BH 分别为各中断源的入口地址。所以，编程时应在 0000H 处写一跳转指令（一般为长跳转指令），使 CPU 在执行程序时，从 0000H 跳过各中断源的入口地址。主程序则是以跳转的目标地址作为起始地址开始编程。

2. 主程序的初始化内容

所谓初始化，是对将要用到的 MCS-51 系列单片机内部部件或扩展芯片进行初始工作状态设定。MCS-51 系列单片机复位后，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 CPU 中断，允许某些中断源中断和设置中断优先级等。

6.4.2 中断服务程序

1. 中断服务程序的起始地址

当 CPU 接收到中断请求信号并予以响应后，CPU 把当前的 PC 内容压入栈中进行保护，然后转入相应的中断服务程序入口处执行。MCS-51 系列单片机中断系统对 5 个中断源分别规定了各自的入口地址（见表 6-1），但这些入口地址相距很近（仅 8 个字节），如果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；若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。

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（PUSH）与弹栈（POP）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。

【例 6-3】如图 6-6 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 8031 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（若输入为高电平则相应的 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：如图 6-6 所示，采用外部中断 0，中断申请从 $\overline{INT\ 0}$ 输入，并采用了去抖动电路。

当 P1.0~P1.3 的任何一位输出 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：

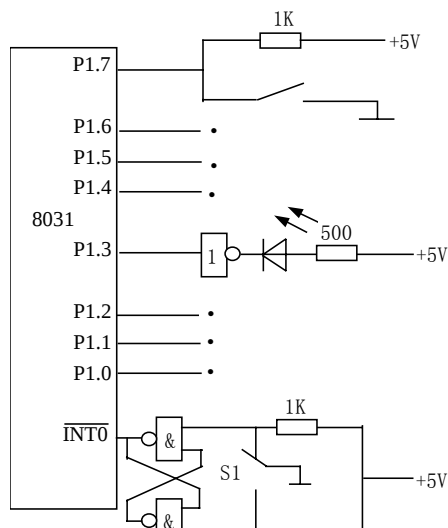


图 6-6 外部中断实验

```

ORG      0000H
AJMP     MAIN          ; 上电，转向主程序
ORG      0003H          ; 外部中断 0 入口地址
AJMP     INSERT        ; 转向中断服务程序
ORG      0100H          ; 主程序
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
         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 4 位互换
         MOV     P1, A   ; 输出驱动 LED 发光
         RETI           ; 中断返回
END

```

思考题与习题

- 6-1 什么是中断和中断系统？其主要功能是什么？计算机采用中断有什么好处？
- 6-2 8051 共有哪些中断源？对其中断请求如何进行控制？
- 6-3 什么是中断优先级？中断优先处理的原则是什么？
- 6-4 说明外部中断请求的查询和响应过程。
- 6-5 8051 在什么条件下可响应中断？
- 6-6 简述 8051 单片机的中断响应过程。
- 6-7 当正在执行某一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？
- 6-8 8051 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？

6-9 在 MCS-51 中，哪些中断可以随着中断被响应而自动撤除？哪些中断需要用户来撤除？撤除方法是什么？