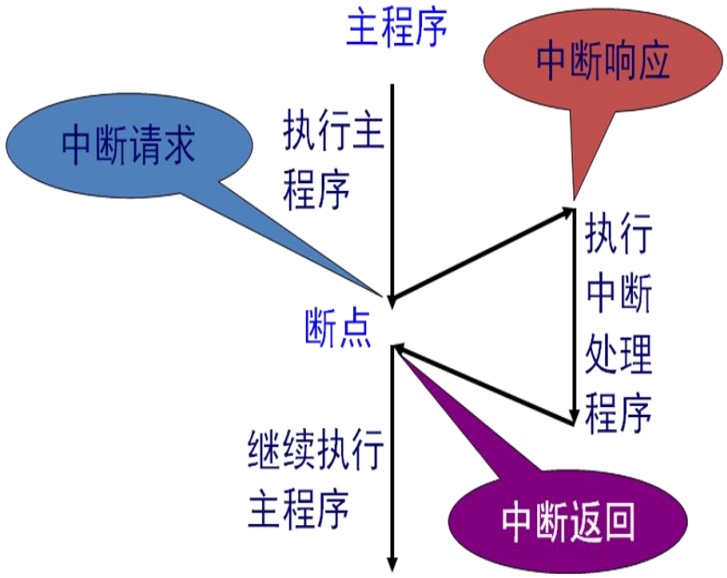
2.5 中断系统

2.5.1 中断概念

一、中断

CPU在处理某一事件A时，发生了另一事件B请求CPU迅速去处理（中断发生）； CPU暂时中断当前的工作，转去处理事件B（中断响应和中断服务）；待CPU将事件B处理完毕后，再回到原来事件A被中断的地方继续处理事件A（中断返回），这一过程称为中断。

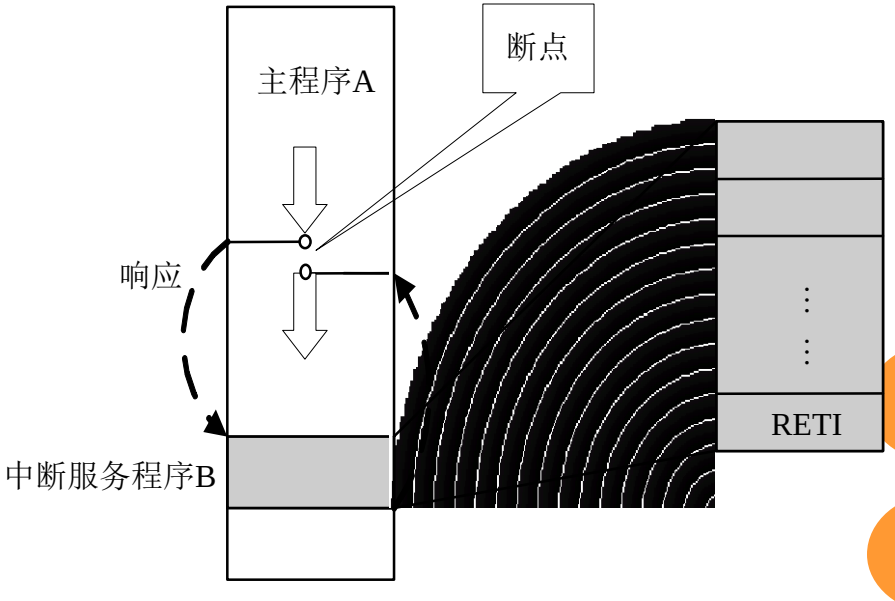


中断过程的示意图

2.5 中断系统

引起CPU中断的根源，称为中断源。中断源向CPU提出的中断请求。CPU暂时中断原来的事务A，转去处理事件B。对事件B处理完毕后，再回到原来被中断的地方（即断点），称为中断返回。实现上述中断功能的部件称为中断系统（中断机构）。

中断是指计算机暂时停止原程序的执行转而为外部设备服务，并在服务完以后自动返回原程序执行的过程。一个资源（CPU）面对多项任务，但由于资源有限，因此就可能出现资源竞争的局面，即几项任务来争夺一个CPU。而中断技术就是解决资源竞争的有效方法，采用中断技术可以使多项任务共享一个资源，中断技术实质上就是一种资源共享技术。



中断响应过程示意图

中断技术是计算机在实时处理和实时控制中不可缺少的重要技术，采用中断技术能极大的提高工作效率和处理问题的灵活性。

2.5 中断系统

二、中断系统

中断系统是计算机中实现中断功能的各种软、硬件的总称。中断系统是计算机的重要组成部分。实时控制、故障自动处理时往往用到中断系统，计算机与外部设备间传送数据及实现人机联系也常常采用中断方式。单片机的中断系统需要解决以下基本问题：

1.实现中断

当单片机内部或外部有中断申请时，CPU能及时响应中断，停下正在执行的任务，转去处理中断服务子程序，中断服务处理后能回到原断点处继续处理原先的任务。

2.中断源

中断源是中断请求信号的来源。包括中断请求信号的产生及该信号怎样被CPU有效地识别。而且要求中断请求信号产生一次，只能被CPU接收处理一次，即不能一次中断申请被CPU多次响应。这就涉及到中断请求信号的及时撤除问题。

2.5 中断系统

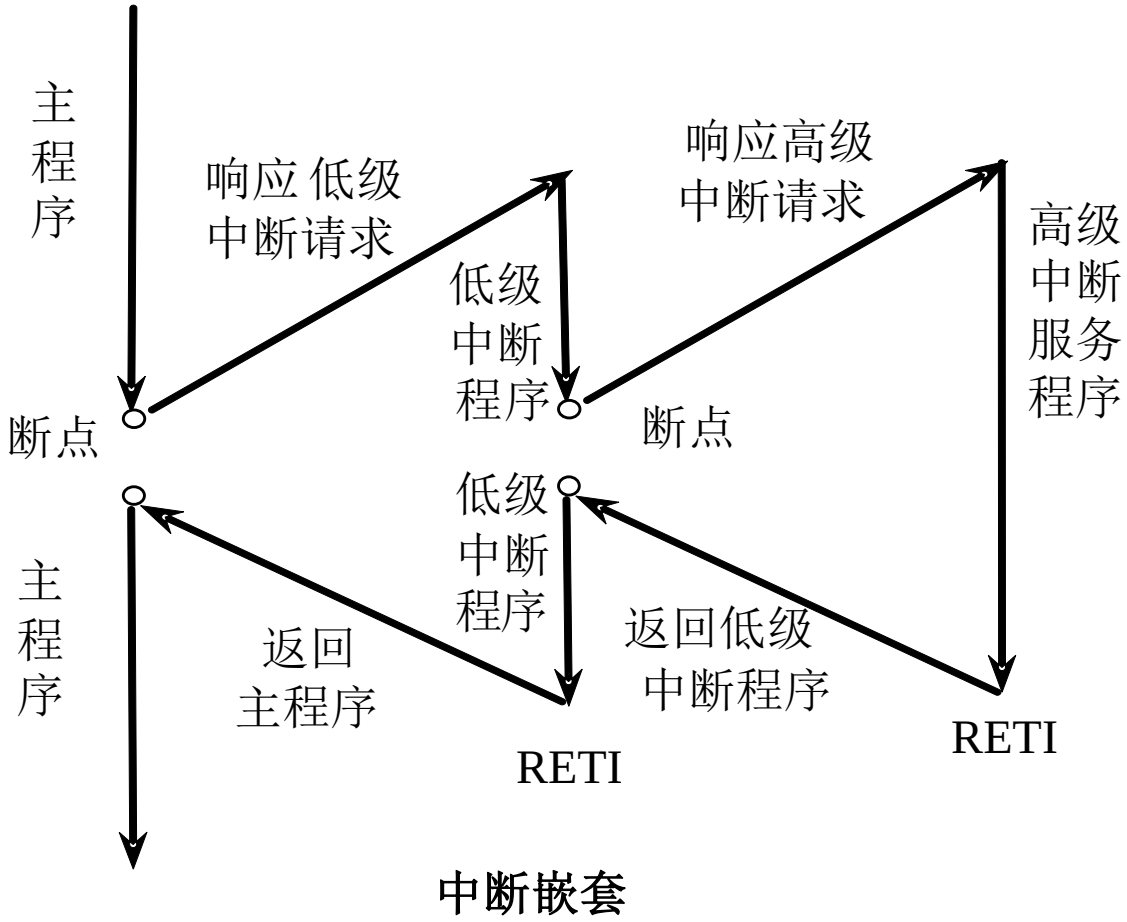
3.实现中断优先级的控制

当有几个中断源同时向CPU发出中断请求，或者CPU正在处理某中断源服务程序时，又有另一中断源申请中断，那么CPU既要能够区分每一个中断源，且要能够确定优先处理哪一个中断源，即中断的优先级。通常首先为优先级最高的中断源服务，再响应级别较低的中断源。按中断源级别高低依次响应的过程称为优先级排队。这个过程可以由硬件电路实现，也可以通过软件查询来实现。

4.实现中断嵌套

当低优先级中断源正在享用中断服务时，若这时优先级比它高的中断源也申请中断，要求能停下低优先级中断源的服务程序转去执行更高优先级中断源的服务程序，实现中断嵌套，并能逐级正确返回原断点处。

2.5 中断系统


主程序
响应低级中断请求
响应高级中断请求
高级中断服务程序
低级中断程序
断点
断点
低级中断程序
主程序
返回主程序
返回低级中断程序
RETI
RETI

中断嵌套

2.5 中断系统

5.自动响应中断

当某一个中断源发出中断请求时，CPU将根据有关条件（是否允许中断、中断的优先级等）进行相应的判断，以决定是否响应该中断请求。若响应该中断请求，CPU在执行完当前指令后，再把断点处的PC值压入堆栈保存起来，这个过程称为保护断点，由硬件自动完成。在中断服务程序开始，由用户把相关寄存器和标志位的状态也压入堆栈保存起来，这称为保护现场。随后开始执行中断服务程序。

6.实现中断返回

执行中断服务程序到最后时，需要从堆栈中恢复相关寄存器和标志位的状态，这称为恢复现场。再执行RETI指令，恢复PC值，即恢复断点，继续执行主程序。

2.5 中断系统

中断处理包括4个步骤：中断请求、中断响应、中断处理和中断返回。

1.中断请求 ■

中断请求由中断源向CPU发出，中断可以人为设定，也可以是为响应突发性随机事件而设置。通常有I/O设备、实时控制系统中的随机参数和信息故障源等中断源发出中断请求。

2.中断响应、处理和返回 ■

(1) 在每条指令结束后，系统都自动检测中断请求信号，如果有中断请求，且CPU处于开中断状态下，则响应中断。 ■

(2) 保护现场，在保护现场前，一般要关中断，以防止现场被破坏。保护现场一般是用堆栈指令将原程序中用到的寄存器推入堆栈。

(3) 中断服务，即为相应的中断源服务。 ■

(4) 恢复现场，用堆栈指令将保护在堆栈中的数据弹出来，在恢复现场前要关中断，以防止现场被破坏。在恢复现场后应及时开中断。 ■

(5) 返回，此时CPU将推入到堆栈的断点地址弹回到程序计数器，从而使CPU继续执行刚才被中断的程序。

2.5 中断系统

三、引入中断技术的优点

1. 实现CPU与外部设备的速度配合

协调快速CPU与慢速外部设备之间的工作。分时操作。CPU可以分时为多个I/O设备服务，提高了CPU的工作效率，实现了CPU和外部设备的并行工作。

2. 实现实时控制

实时控制，计算机能及时地响应被控对象提出的分析、计算和控制等请求，使被控对象保持在最佳工作状态，以达到预定的控制效果。由于这些控制参数的请求都是随机发出的，而且要求单片机必须作出快速响应并及时处理，对此，只有靠中断技术才能实现实时响应。CPU能够及时处理应用系统的随机事件，系统的实时性大大增强。

3. 实现故障的及时发现及处理

外界干扰、硬件或软件设计中存在问题等因素，实际运行中会出现硬件故障、运算错误、程序运行故障等，有了中断技术，单片机就能及时发现故障并自动处理。便于突发故障（如硬件故障、运算错误、电源掉电、程序故障等）的及时发现，提高系统可靠性。提高了计算机处理故障与应变的能力。

4. 实现人机联系

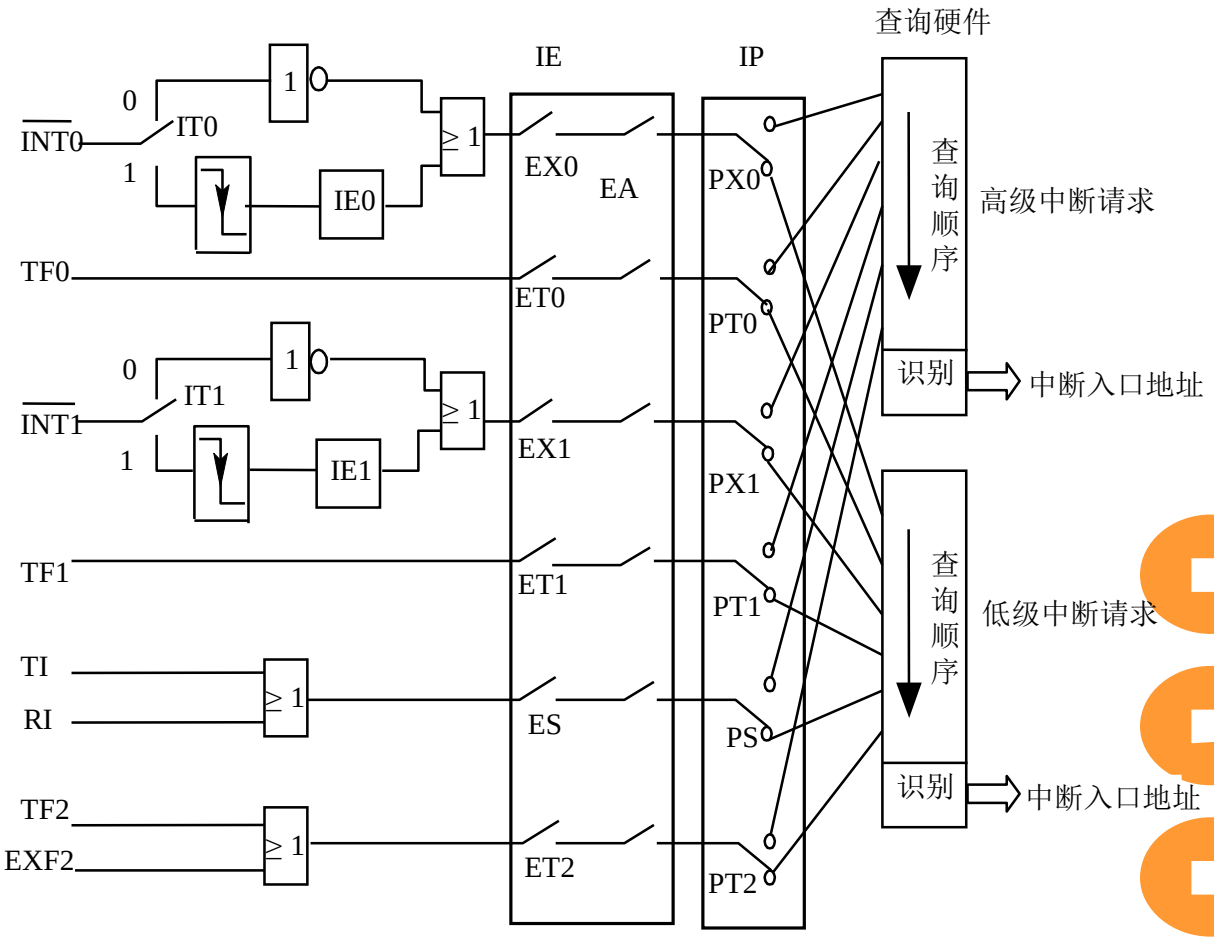
能使用户通过键盘向单片机发出中断请求，可以实时干预计算机的工作。

2.5 中断系统

2.5.2 89C52中断系统

一、 89C52中断源

向CPU发出中断请求的来源称之为中断源。89C52是一个多中断源的单片机，共6个中断源，分别是外部中断2个，定时中断3个和串行中断1个。其中，定时中断和串行中断均属内部中断。



89C52中断系统逻辑结构

2.5 中断系统

1.外部中断

外部中断是由外部信号引起的，是指从单片机外部引脚 $\overline{\text{INT0}}$、$\overline{\text{INT1}}$ 输入中断请求信号的中断，即外部中断源有两个，外部中断0和外部中断1。输入/输出的中断请求、实时事件的中断请求、掉电和设备故障的中断请求都可以作为外部中断源，从引脚 $\overline{\text{INT0}}$ 、$\overline{\text{INT1}}$ 输入。

外部中断请求 $\overline{\text{INT0}}$、$\overline{\text{INT1}}$有两种触发方式：电平触发及脉冲触发方式。这两种触发方式可以通过对特殊功能寄存器中的定时器控制寄存器（**TCON**）编程来选择。当设定为电平触发方式时，若 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚上采样到有效的低电平，则向**CPU**提出中断请求；当设定为脉冲触发方式时，若$\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚上采样到有效负跳变，则向**CPU**提出中断请求。

2.5 中断系统

定时器控制寄存器（TCON）

用于保存外部中断请求以及定时器的计数溢出。TCON的字节地址为98H。TCON各位定义格式如下：

D7	D6	D5	D4	D3	D2	D1	D0
TF1	—	TF0	—	IE1	IT1	IE0	IT0

TCON中与中断有关的定义位

- IT0和IT1

外部中断0(或1)触发方式控制位。IT0(或IT1)被设置为0，则选择外部中断为电平触发方式，低电平有效；IT0(或IT1)被设置为1，则选择外部中断为脉冲触发方式，后沿负跳有效。IT0(或IT1)由软件置“1”或清“0 " 。

2.5 中断系统

- **IE0和IE1**

外部中断0(或1)的中断请求标志位。当IT0(或IT1)=0，即电平触发方式时，CPU在每个机器周期的S5P2采样 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚。若 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚为低电平，将直接触发外部中断。脉冲触发方式时，若第一个机器周期采样到 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚为高电平，第二个机器周期采样到 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚为低电平时，由硬件置位IE0(或IE1)，并以此负跳变信号向CPU请求中断。当CPU响应中断响应完成后转向中断服务程序时由硬件将IE0(或IE1) 自动清零。

2.内部中断

内部中断是单片机芯片内部产生的中断。89C52内部中断有定时器/计数器T0、T1和T2的溢出中断，串行口的发送/接收中断。

1）定时中断：为满足定时或计数的需要而设置。当计数结构发生计数溢出时，即表明定时时间到或计数值已满，这时可以向CPU申请中断。由于请求是在单片机芯片内部发生的，无需在芯片上设置引入端，故为内部中断。

2.5 中断系统

TCON中与定时中断有关的定义位：

TF0（TCON.5），定时/计数器T0溢出中断请求标志位。

TF1（TCON.7），定时/计数器T1溢出中断请求标志位。

当定时器/计数器T0、T1的定时或计数到由硬件自动置位TCON的TF0或TF1，便向CPU申请中断。CPU响应中断而转向中断服务程序时，由硬件自动将TF0或TF1清零，即CPU响应中断后能自动撤除中断请求信号。

TF2和EXF2：在特殊功能寄存器T2CON中，逻辑或后作为一个中断源。CPU响应中断时不清零，须由软件清零。

定时/计数器溢出标志位的使用有两种情况：采用中断方式时，作中断请求标志位来使用；采用查询方式时，作查询状态位来使用。

2.5 中断系统

2）串行中断：为串行数据传送的需要而设置。每当串行口接收或发送完一组串行数据时，就产生一个中断请求。请求是在单片机芯片内部自动发生的，不需在芯片上设置引入端。

串行中断的中断标志在特殊功能寄存器SCON中。

RI（SCON.0），串行口接收中断标志位。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。同样，RI必须由软件清除。

TI（SCON.1），串行口发送中断标志位。当CPU将一个发送数据写入串行口发送缓冲器时，就启动了发送过程。每发送完一个串行帧，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。

当串行口发送完或接收完一帧信息，由接口硬件自动置位SCON的TI或RI，以此向CPU申请中断，CPU响应中断后，接口硬件不能自动将TI或RI清零，即CPU响应中断后不能自动撤除中断请求信号，需用户采用软件方法将TI或RI清零，来撤除中断请求信号。

2.5 中断系统

二、中断控制

89C52有 6个中断源, 为了使每个中断源都能独立地被允许或禁止, 以便用户能灵活使用, 它在每个中断信号的通道中设置了一个中断屏蔽触发器。只有该触发器无效, 它所对应的中断请求信号才能进入CPU, 即此类型中断开放。否则, 即使其对应的中断标志位置1, CPU也不会响应中断, 即此类型中断被屏蔽了。同时CPU内还设置了一个中断允许触发器, 它控制CPU能否响应中断。

1.中断允许寄存器（IE）

单片机中没有专设的开中断和关中断指令，对各中断源的中断开放或关闭是由内部的中断允许寄存器IE的各位来控制的。IE寄存器的字节地址为0A8H，各位地址为0AFH～0A8H。IE各位的定义如下：

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

2.5 中断系统

1）EA：中断允许总控制位。

EA=0，中断总禁止，禁止所有的中断请求；EA=1，中断总允许，开放中断。EA的作用是使中断允许形成两级控制。即各中断源首先受EA位的控制；其次还要受各中断源自己的中断允许控制位控制。EA中断总允许后，中断的禁止或允许由各中断源的中断允许控制位进行设置。

2）ET2：定时器/计数器T2的溢出中断允许位。

ET2=0，禁止T2中断；ET2=1，允许T2中断。

3）ES：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。

2.5 中断系统

4）ET1：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。

5）EX1：外部中断1($\overline{\text{INT1}}$)的中断允许位。

EX1=0，禁止外部中断1中断；EX1=1，允许外部中断1中断。

6）ET0：定时器/计数器T0的溢出中断允许位。

7）EX0：外部中断0($\overline{\text{INT0}}$)的中断允许位。

EX0=0，禁止外部中断0中断；EX0=1允许外部中断0中断。

2.5 中断系统

2.中断优先级控制寄存器IP

89C52的6个中断源有两个用户可控的中断优先级，从而可实现二级中断嵌套。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。IP寄存器字节地址为0B8H，各位地址为0BFH～0B8H。

IP寄存器各位定义如下：

D7	D6	D5	D4	D3	D2	D1	D0
—	—	PT2	PS	PT1	PX1	PT0	PX0

PX0外部中断0优先级设定位；

PT0定时器T0中断优先级设定位，

PX1外部中断1优先级设定位；

PT1定时器T1中断优先级设定位；

PS串行中断优先级设定位；

PT2定时器T2中断优先级设定位。

为“0”的位优先级为低；为“1”的位优先级为高。

2.5 中断系统

中断优先级控制原则和控制逻辑

89C52具有两级优先级，具备两级中断服务嵌套的功能。中断优先级的控制原则：

- CPU同时接收到几个中断请求时，首先响应优先级最高的中断请求。
- 低优先级中断请求不能打断高优先级的中断服务；但高优先级中断请求可以打断低优先级的中断服务，从而实现中断嵌套。
- 如果一个中断请求已被响应，则同级的其它中断服务将被禁止。即同级不能嵌套。

如果同级的多个中断请求同时出现，则有中断优先权排队问题。同一优先级的中断优先权排队，由中断系统硬件确定的自然优先级形成，CPU按查询次序（自然优先级）确定那个中断请求被响应。

为了实现上述原则，中断系统内部设有两个用户不能寻址的优先级状态触发器。其中一个置1，表示正在响应高优先级的中断，它将阻断后来所有的中断请求；另一个置1，表示正在响应低优先级中断，它将阻断后来所有的低优先级中断请求。

2.5 中断系统

中断控制举例

例如，某软件中对寄存器IE、IP设置如下：

MOV IE, #8FH

MOV IP, #06H

则此时该系统中：

CPU中断总允许；允许外部中断0、外部中断1、 定时器/计数器0、定时器/计数器1提出的中断申请；

允许中断源的中断优先次序为：定时器/计数器0>外部中断1>外部中断0>定时器/计数器1。

中 断 源	同级的中断优先级
外部中断0 定时器/计数器0中断 外部中断1 定时器/计数器1中断 串行口中断 定时器/计数器2中断	最高 ↓ 最低

同级中断请求的中断优先级（自然顺序）

2.5 中断系统

三、中断响应过程

CPU中断响应过程分三步完成：中断响应、执行中断服务程序及中断返回。

1.中断响应的条件

下述三个中断响应的条件同时满足时，CPU才有可能响应中断。

- **有中断源提出中断请求；**
- **中断总允许位EA=1，即CPU开放中断；**
- **申请中断的中断源的中断允许位为1（中断允许寄存器IE相应位置1），即没有被屏蔽。**

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的S1期间按优先级和内部顺序被依次查询。如果某个中断标志在上一个机器周期的S5P2时被置成了1，那么它将于现在的查询周期中及时被发现。接着CPU便执行一条由中断系统提供的硬件LCALL指令，转向被称作中断入口的特定地址单元，进入相应的中断服务程序。

2.5 中断系统

遇到以下三种情况的任意一种，硬件将受阻，不产生LCALL指令：

1）CPU正在处理同级或高优先级中断；

2）当前查询的机器周期不是所执行指令的最后一个机器周期。即在完成所执行指令前，不会响应中断，从而保证指令在执行过程中不被打断；

3）正在执行的指令为RET、RETI或任何访问特殊功能寄存器IE或IP的指令。换言之, 在RET、RETI或读写IE或IP之后，不会马上响应中断请求，而至少再执行一条其它指令之后才能接受中断请求。

若由于上述情况的阻碍中断未能得到响应，当情况消失时该中断标志却已不再有效，那么该中断将不被响应。也就是说，中断标志曾经有效，但未获响应，查询过程在下个机器周期将重新进行。

2.5 中断系统

2.中断响应过程

CPU响应中断后，由硬件自动执行如下的功能操作：

(1) 根据响应的中断请求源的优先级高低，对相应的优先级状态触发器置1（以阻断后来的同级或低级的中断请求）。

(2) 保护断点，执行硬件中断服务子程序调用LCALL指令，并把当前程序计数器PC的内容压入堆栈保存。也就是将CPU本来要取用的指令地址暂存到堆栈中保护起来，以便中断结束时，CPU能找到原来程序的断点处，继续执行下去。

(3) 保护现场时关闭中断，以防其他中断信号干扰。此时，中断系统关闭该中断源接收电路，其他中断请求均被禁止。这一措施需用指令完成。

(4) 清内部硬件可清除的中断请求标志位(IE0、IE1、TF0、TF1)，串行口中断请求标志RI和TI除外。

(5) 把被响应的中断源所对应的中断服务程序的入口地址（中断矢量）送入PC，从而转入相应的中断服务程序执行。

2.5 中断系统

3.中断服务程序

当CPU响应中断时，由硬件直接产生一个固定的地址，即入口（矢量）地址，由入口地址指出每个中断源设备的中断服务程序的入口。当CPU识别出某个中断源时，由硬件直接给出一个与该中断源相对应的入口地址，从而转入各自中断服务程序。在中断响应后，计算机调用的子程序称为中断服务程序。这是专门为外部设备或其他内部部件中断源服务的程序段，其结尾必须是中断返回指令RETI。中断服务程序要由用户编写程序来完成。

2.5 中断系统

中断服务程序入口地址：

中断源	入口地址
外部中断0	0003H
定时器T0	000BH
外部中断1	0013H
定时器T1	001BH
串行口中断	0023H
定时器T2	002BH

2.5 中断系统

4.中断返回

计算机在中断响应时执行到RETI指令时，立即结束中断并从堆栈中自动取出在中断响应时压入的PC当前值，从而使CPU返回原程序中断点继续进行下去。

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。

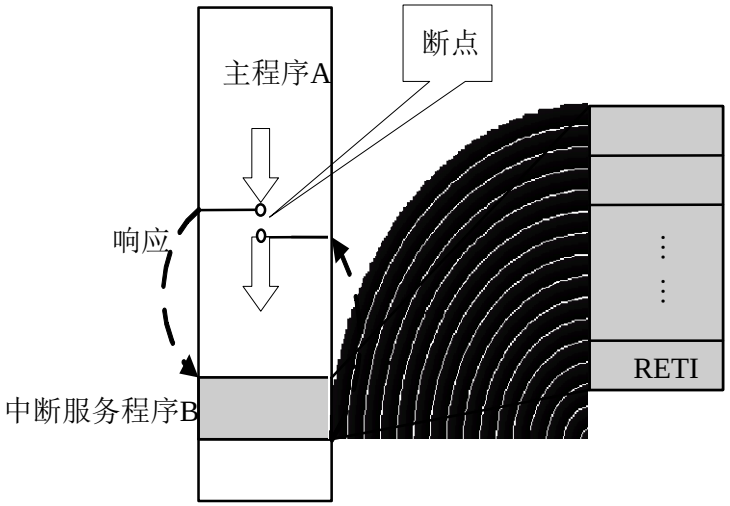
2.5 中断系统

中断响应过程举例

例如，现有外部中断1提出申请，且主程序中有R0、R1、DPTR、累加器A需保护，则编制程序应为：

```
        ORG  0000H   ■
        AJMP  MAIN ■
        ORG  0013H ■
        LJMP  INT1 ■
        … ■
        ORG  0100H ■
MAIN: …                ;    主程序 ■
        … ■
        ORG  1000H
■
```

```
INT1: PUSH  ACC
                ；中断服务程序
      PUSH  DPH ■
      PUSH  DPL ■
      PUSH  0 ■
      PUSH  1 ■
      POP  1 ■
      POP  0 ■
      POP  DPL ■
      POP  DPH ■
      POP  ACC ■
      RETI ■
```



2.5 中断系统

编程中：

(1）在0000H放一条跳转到主程序的跳转指令, 这是因为单片机复位后, PC的内容变为0000H，程序从0000H 开始执行，紧接着0003H是中断程序入口地址，故在此中间只能插入一条转移指令；

(2）响应中断时，先自动执行一条隐指令“LCALL 0013H”，而0013H至 001BH（定时器 1 溢出中断入口地址）之间可利用的存储单元不够，故放一条无条件转移指令；

(3）在中断服务程序的末尾，必须安排一条中断返回指令RETI，使程序自动返回主程序。

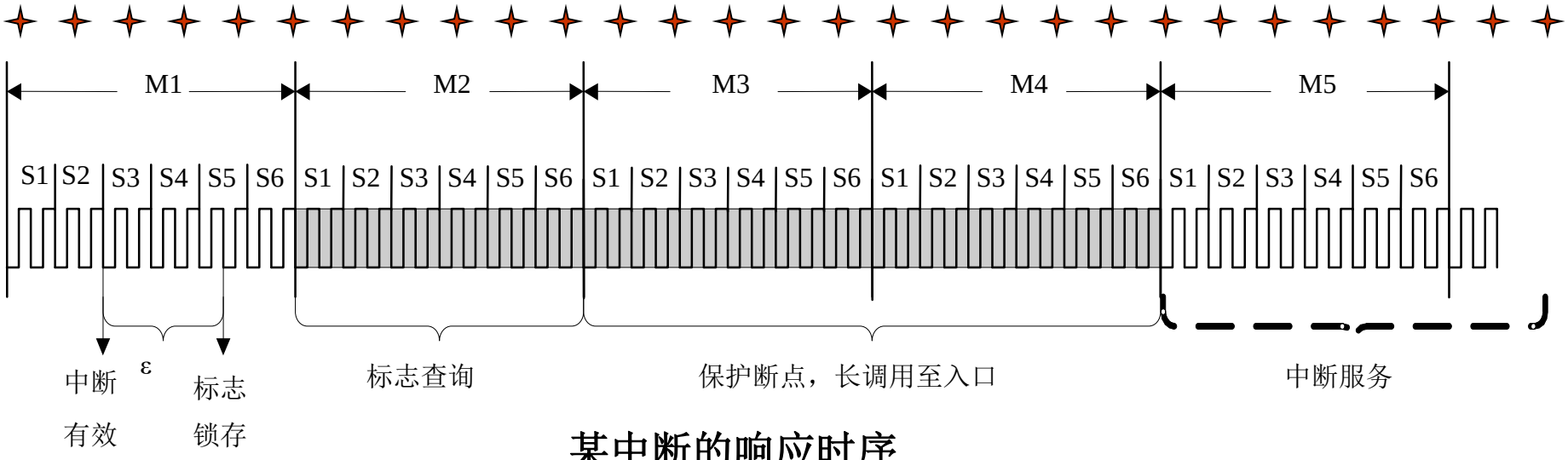
2.5 中断系统

四、中断响应时间

所谓中断响应时间是指CPU检测到中断请求信号到转入中断服务程序入口所需要的机器周期数。了解中断响应时间对设计实时测控应用系统有重要意义。

51单片机响应中断的最短时间为3个机器周期。若CPU检测到中断请求信号时间正好是一条指令的最后一个机器周期，则不需等待就可以立即响应。所谓响应中断就是由内部硬件执行一条长调用指令，需要2个机器周期，加上检测需要1个机器周期，一共需要3个机器周期才开始执行中断服务程序。

2.5 中断系统



某中断的响应时序

若M1周期的S5P2前某中断生效，在S5P2期间其中断请求被锁存到相应的标志位中去；M2恰逢指令的最后一个机器周期，且该指令不是RETI或访问IE、IP的指令。于是，M3和M4便可以执行硬件LCALL指令，M5周期将进入了中断服务程序。51单片机的中断响应时间（从标志置1到进入相应的中断服务），至少要3个完整的机器周期。

2.5 中断系统

中断响应的最长时间由下列情况决定：

若中断检测时正在执行RETI或访问IE或IP指令的第一个机器周期，这样包括检测在内需要2个机器周期(以上三条指令均需两个机器周期)；若紧接着要执行的指令恰好是执行时间最长的乘除法指令，其执行时间均为4个机器周期；再用2个机器周期执行一条硬件长调用LCALL指令才转入中断服务程序。这样，总共需要8个机器周期。一般情况下的中断响应时间在3-8个机器周期之间。

2.5 中断系统

2.5.3 外部中断触发方式选择

外部中断请求有两种信号方式，即电平方式和脉冲方式。可通过设置定时器控制寄存器（TCON）的有关控制位进行定义。

一、电平触发方式

电平方式的中断请求是低电平有效。只要单片机在中断请求引入端上采样到有效的低电平时，就激活外部中断。若外部中断定义为电平触发方式，中断标志位的状态随CPU在每个机器周期采样到的外部中断输入引脚的电平变化而变化，这样能提高CPU对外部中断请求的响应速度。但外部中断源若有请求，必须把有效的低电平保持到请求获得响应时为止，不然就会漏掉；而在中断服务程序结束之前，中断源又必须撤消其有效的低电平，否则中断返回之后将再次产生中断。

对于电平触发的外部中断，由于CPU对 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 引脚没有控制作用，也没有相应的中断请求标志位，因此需要外接电路来撤除中断请求信号。

2.5 中断系统

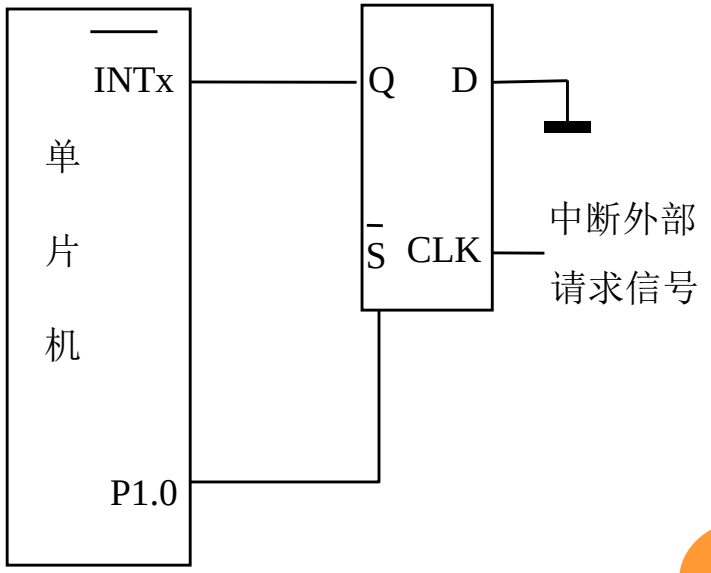
外部中断请求信号通过D触发器加到单片机引脚上。当外部中断请求信号使D触发器的CLK端发生正跳变时，由于D端接地，Q端输出0，向单片机发出中断请求。CPU响应中断后，利用一根口线，如P1.0作应答线，在中断服务程序中用两条指令：

ANL P1.0 #0FEH

ORL P1.0 #01H

来撤除中断请求。第一条指令使P1.0为0，而P1口其它各位的状态不变。由于P1.0与直接置1端相连，故D触发器Q=1，撤除了中断请求信号 。第二条指令将P1.0变成1，从而 $\overline{S}$ =1，使以后产生的新的外部中断请求信号又能向单片机申请中断。

电平触发方式适合于外部中断输入以低电平输入且中断服务程序能清除外部中断请求源的情况。



中断请求信号撤除电路

2.5 中断系统

二、边沿触发方式

脉冲方式，其中断请求在脉冲的后沿负跳有效。CPU在两个相随机器周期对中断请求引入端进行的采样中，如前一次为高电平，后一次为低电平，即为有效中断请求。若外部中断定义为边沿触发方式，在相继连续的两次采样中，一个周期采样到外部中断输入为高电平，下一个周期采样到为低电平，则在IE0或IE1中将锁存一个逻辑1。即便是CPU暂时不能响应，中断申请标志也不会丢失，直到CPU响应此中断时才清零。这样，为保证下降沿能被可靠地采样到，外部中断引脚上的高低电平（负脉冲的宽度）均至少要保持一个机器周期（若晶振为12MHz时，为1微秒）。

边沿触发方式适合于以负脉冲形式输入的外部中断请求。

2.5 中断系统

中断系统应用举例

例2-1 单步操作的中断实现。 ■

把一个外部中断（设为 $\overline{\text{INT0}}$）设置为电平触发方式。其中断服务程序的末尾写上如下几条指令： ■

```
JNB  P3.2, $  ;  在INT0 变高前原地等待(死循环) ■
JB   P3.2, $  ;  在 INT0 变低前原地等待(死循环) ■
RETI          ;  返回并执行一条指令 ■
```

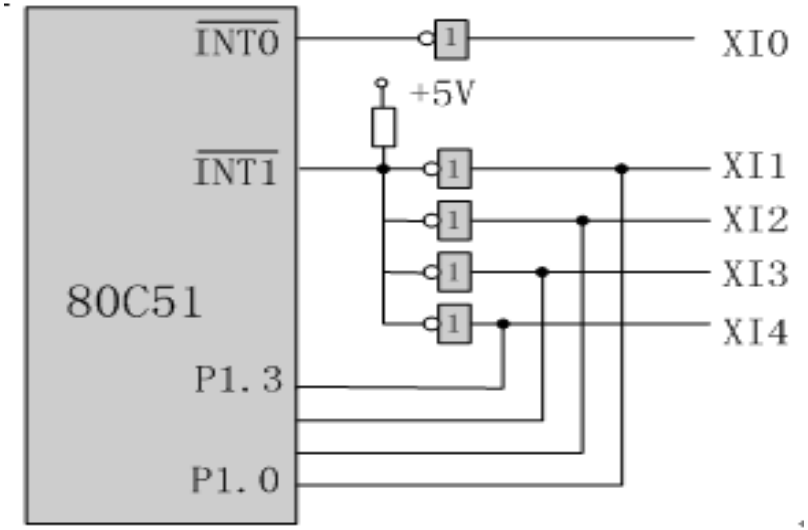
若 $\overline{\text{INT0}}$ 保持低电平，且允许 $\overline{\text{INT0}}$ 中断，则CPU就进入外部中断0服务程序，由于有上述几条指令，它就会停在JNB处，原地等待。当 $\overline{\text{INT0}}$ 端出现一个正脉冲（由低到高，再到低）时，程序就会往下执行，执行RETI后，将返回主程序，往下执行一条指令，然后又立即响应中断，以等待 $\overline{\text{INT0}}$ 端出现的下一个正脉冲。这样在 $\overline{\text{INT0}}$ 端每出现一个正脉冲，主程序就执行一条指令，实现了单步执行的目的，要注意的是，这个正脉冲的高电平持续时间不小于3个周期，以确保CPU能采集到高电平值。

2.5 中断系统

例2-2 多中断源。■

51 单片机只有两个外部中断输入端，当有2个以上中断源时，它的中断输入端就不够了。此时，可以采用中断与查询相结合的方法来实现。可以使每个中断源都接在同一个外部中断输入端上，同时利用输入口线作为多中断源情况下各中断源的识别线。设有5个外部中断源，中断优先级排队顺序为：XI0、XI1、XI2、XI3、XI4。设计它们与80C51单片机的接口。

相应的程序如下：



2.5 中断系统

```
ORG  0003H
LJMP INSE0 ；转外部中断0服务程序入口
ORG  0013H
LJMP INSE1 ；转外部中断1服务程序入口
… …
… …
INSE0：PUSH PSW    ；XI0中断服务程序
       PUSH ACC
       … …
       … …
       POP  ACC
       POP  PSW
       RETI
```

```
INSE1：PUSH PSW      ；中断服务程序
       PUSH ACC
JB P1.0，DV1   ；P1.0为1，转XI1中断服务程序
JB P1.1，DV2   ；P1.1为1，转XI2中断服务程序
JB P1.2，DV3   ；P1.2为1，转XI3中断服务程序
JB P1.3，DV4   ；P1.3为1，转XI4中断服务程序
INRET：POP ACC
       POP PSW
       RETI
  DV1：… …          ；XI1中断服务程序
       AJMP INRET
  DV2：… …          ；XI2中断服务程序
       AJMP INRET
  DV3：… …          ；XI3中断服务程序
       AJMP INRET
  DV4：… …          ；XI4中断服务程序
       AJMP INRET
```