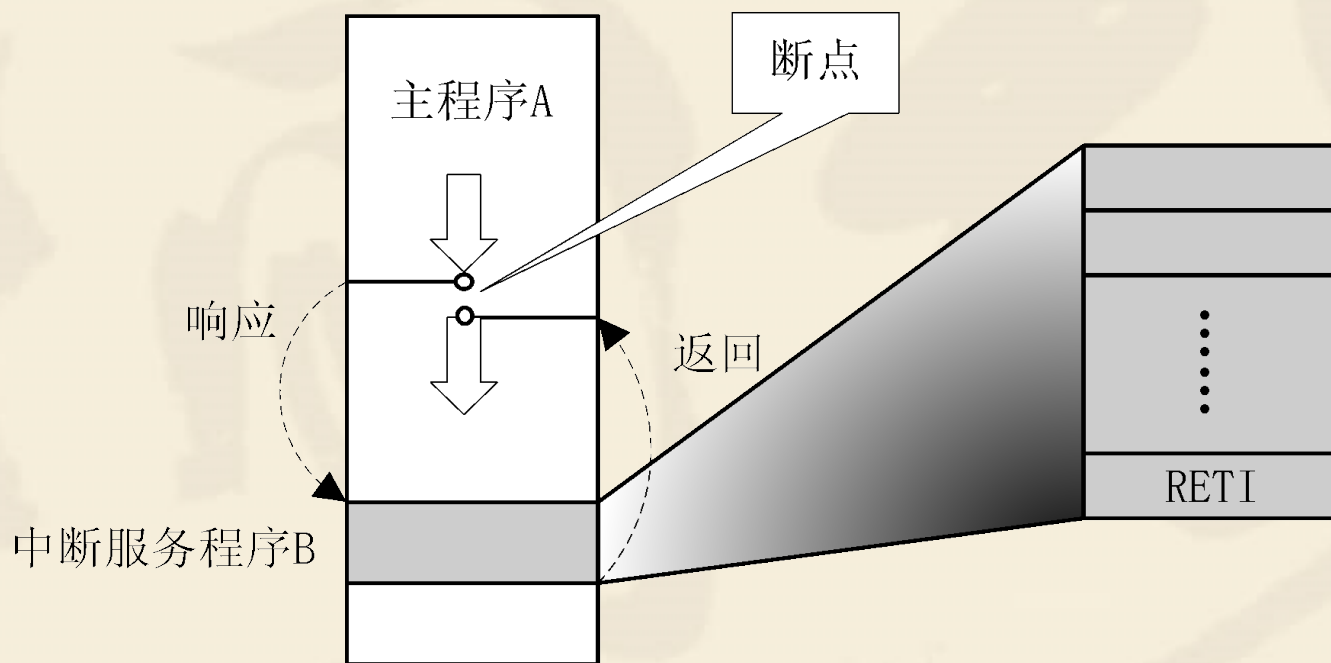


第5章-80C51单片机的中断系统



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）。

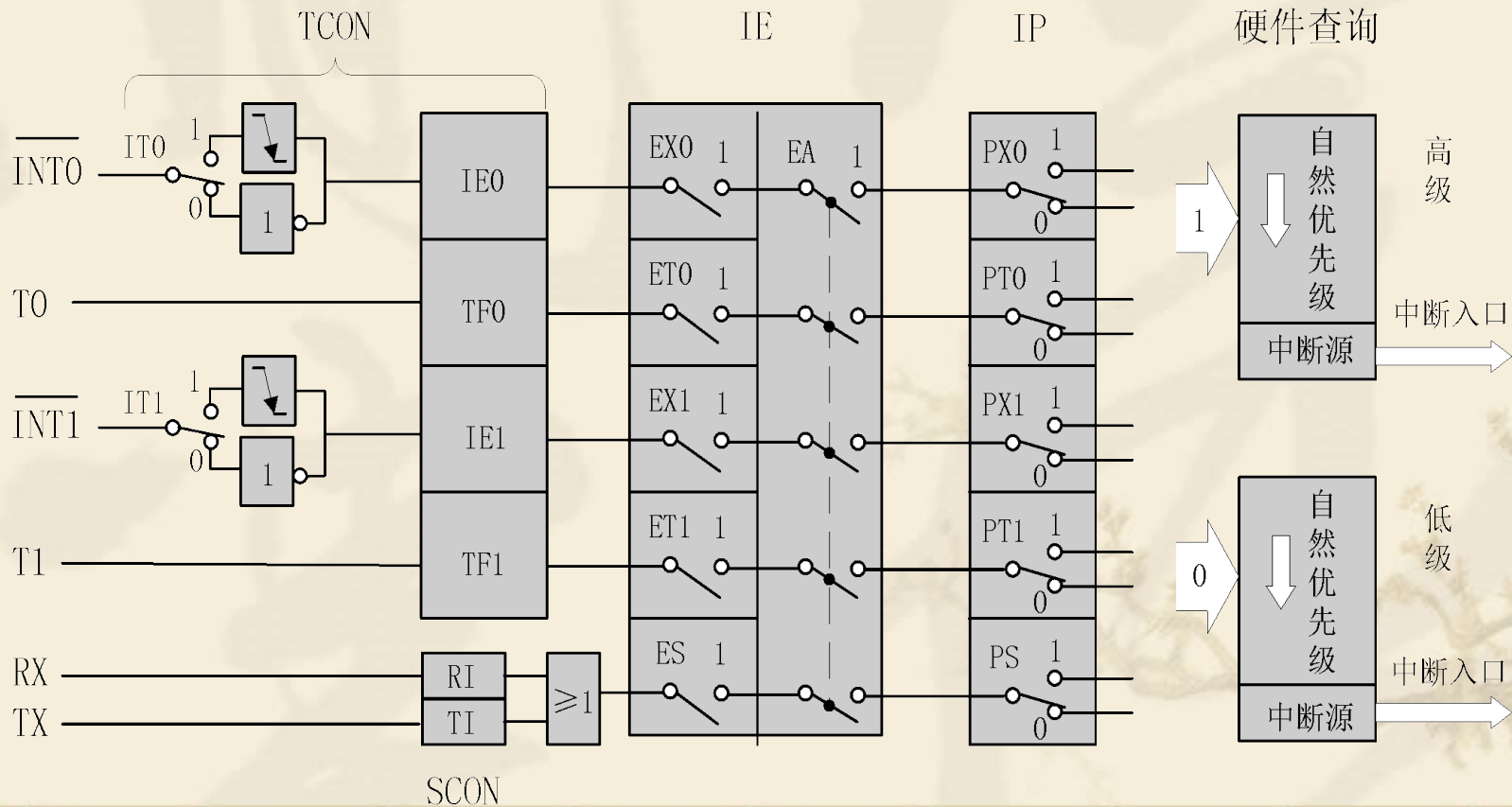
。

随着计算机技术的应用，人们发现中断技术不仅解决了快速主机与慢速I/O设备的数据传送问题，而且还具有如下优点：

- ❖ **分时操作**。CPU可以分时为多个I/O设备服务，提高了计算机的利用率；
- ❖ **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；
- ❖ **可靠性高**。CPU具有处理设备故障及掉电等突发性事件能力，从而使系统可靠性提高。

二、80C51中断系统的结构

80C51的中断系统有5个中断源，2个优先级，可实现二级中断嵌套。



5.1.2 80C51的中断源

一、中断源

1、 $\overline{\text{INT0}}$ (P3.2)。可由IT0 (TCON.0) 选择其为低电平有效还是下降沿有效。当CPU检测到P3.2引脚上出现有效的中断信号时，中断标志IE0 (TCON.1) 置1，向CPU申请中断。

2、 $\overline{\text{INT1}}$ (P3.3)。可由IT1 (TCON.2) 选择其为低电平有效还是下降沿有效。当CPU检测到P3.3引脚上出现有效的中断信号时，中断标志IE1 (TCON.3) 置1，向CPU申请中断。

I	位	7	6	5	4	3	2	1	0	
字节地址: 88H		TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

3、**TF0** (TCON.5)，片内定时/计数器T0溢出中断请求标志。当定时/计数器T0发生溢出时，置位TF0，并向CPU申请中断。

4、**TF1** (TCON.7)，片内定时/计数器T1溢出中断请求标志。当定时/计数器T1发生溢出时，置位TF1，并向CPU申请中断。

5、**RI** (SCON.0) 或**TI** (SCON.1)，串行口中断请求标志。当串行口接收完一帧串行数据时置位RI或当串行口发送完一帧串行数据时置位TI，向CPU申请中断。

I 位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

位	7	6	5	4	3	2	1	0	
字节地址: 98H							TI	RI	SCON

二、中断请求标志

在中断系统中，采用哪种中断，选择哪种触发方式，要由TCON和SCON相应位规定。

1、TCON(定时/计数器控制寄存器)的中断标志

I 位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

IT0 (TCON.0) ，外部中断0触发方式控制位。

当IT0=0时，为低电平触发方式。CPU在每个机器周期的S5P2采样INT0#，为0，置IE0=1，表示向CPU请求中断；为1，清零IE0，表示没有中断请求。

低电平触发方式下，IE0状态完全由INT0#决定，响应中断时并不自动清除IE0！要由软件清除！CLR IE0

当IT0=1时，为边沿触发方式（下降沿有效）。CPU在每个机器周期的S5P2采样INT0#，如果在连续两个机器周期检测到INT0#引脚由高电平变到低电平，则置IE0=1，产生中断请求。

边沿触发方式下，CPU响应中断时由硬件自动清IE0标志。



1、TCON(定时/计数器控制寄存器) 的中断标志

位	7	6	5	4	3	2	1	0	
字节地址: 88H	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0	TCON

IE0 (TCON. 1) , 外部中断0中断请求标志位。

IT1 (TCON. 2) , 外部中断1触发方式控制位。

IE1 (TCON. 3) , 外部中断1中断请求标志位。

TF0 (TCON. 5) , 定时/计数器T0溢出中断请求标志位。T0启动后就开始由初值加1计数, 直至最高位产生溢出使TF0=1向CPU发出中断请求。CPU响应中断时, TF0会自动清0。

TF1 (TCON. 7) , 定时/计数器T1溢出中断请求标志位。

TCON的低4位用于控制外部中断, TCON的高4位用于控制定时/计数器的启动和中断申请。

2、SCON（串行口控制寄存器）的中断标志

位 [↕]	7 [↕]	6 [↕]	5 [↕]	4 [↕]	3 [↕]	2 [↕]	1 [↕]	0 [↕]	[↕]
字节地址：98H [↕]	[↕]	[↕]	[↕]	[↕]	[↕]	[↕]	TI [↕]	RI [↕]	SCON [↕]

❖与中断有关的是其低2位

❖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。

❖单片机复位后，TCON和SCON各位清0。

5.1.3 80C51中断的控制

一、中断允许控制

CPU对中断系统所有中断以及某个中断源的开放和屏蔽是由**中断允许寄存器IE**控制的。IE由软件设置。某位设为**1**，相应的中断源**允许中断**；某位设为**0**，相应的中断源**中断屏蔽**。CPU复位时，**IE=0**，禁止所有中断。

位	7	6	5	4	3	2	1	0	
字节地址: A8H	EA			ES	ET1	EX1	ET0	EX0	IE

- ❖ **EX0** (IE. 0)，INT0#中断允许位；
- ❖ **ET0** (IE. 1)，定时/计数器T0中断允许位；
- ❖ **EX1** (IE. 2)，INT1#中断允许位；
- ❖ **ET1** (IE. 3)，定时/计数器T1中断允许位；
- ❖ **ES** (IE. 4)，串行中断允许位；
- ❖ **EA** (IE. 7)，CPU中断允许（总允许）位。

二、中断优先级控制

80C51单片机有两个中断优先级，即可实现二级中断服务嵌套。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设定的。某位设定为1，则相应中断源为高优先级；某位设定为0，则相应中断源为低优先级。单片机复位时，IP=0，各中断源同为低优先级。

位	7	6	5	4	3	2	1	0	
字节地址: B8H				PS	PT1	PX1	PT0	PX0	IP

- ❖PX0 (IP. 0) ， 外部中断0优先级设定位；
- ❖PT0 (IP. 1) ， 定时/计数器T0优先级设定位；
- ❖PX1 (IP. 2) ， 外部中断1优先级设定位；
- ❖PT1 (IP. 3) ， 定时/计数器T1优先级设定位；
- ❖PS (IP. 4) ， 串行口优先级设定位。

同一优先级中的中断申请不止一个时，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，其排列如图所示：

各中断源响应优先级及中断服务程序入口表

中断源	中断标志	中断服务程序入口	优先级顺序
外部中断 0 ($\overline{\text{INT0}}$)	IE0	0003H	高
定时/计数器 0 (T0)	TF0	000BH	↓
外部中断 1 ($\overline{\text{INT1}}$)	IE1	0013H	↓
定时/计数器 1 (T1)	TF1	001BH	↓
串行口	RI 或 TI	0023H	低

80C51单片机的中断优先级有三条原则：

- ❖ 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。

\\为了实现上述后两条原则，中断系统内部设有用户不能寻址的高、低两个优先级的状态触发器。高优先级状态触发器置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；低优先级状态触发器置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。优先级状态触发器的复位由中断返回指令RETI控制。





❖ 中断服务子程序必须用RETI结尾，否则后续中断将被屏蔽。

5.2 80C51单片机中断处理过程

5.2.1 中断响应条件和时间

一、中断响应条件

- ❖ 中断源有中断请求；
- ❖ 此中断源的中断允许位为1；
- ❖ CPU开中断（即EA=1）。

同时满足时，CPU才有可能响应中断。

中断服务的进入：

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。

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，转向被称作中断向量的特定地址单元，进入相应的中断服务程序。

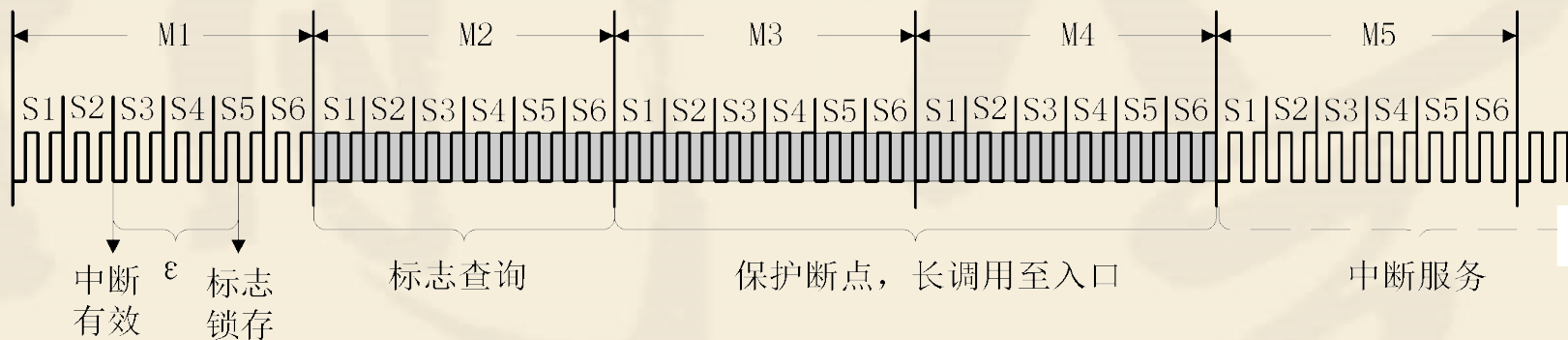
遇以下任一条件，硬件将受阻，不产生LCALL指令：

- ❖ CPU正在处理同级或高优先级中断；
- ❖ 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；
- ❖ 正在执行的指令为RET、RETI或任何访问IE或IP寄存器的指令。即只有在这些指令后面至少再执行一条指令时才能接受中断请求。

若由于上述条件的阻碍中断未能得到响应，当条件消失时该中断标志却已不再有效，那么该中断将不被响应。就是说，中断标志曾经有效，但未获响应，查询过程在下一个机器周期将重新进行。

二、中断响应时间

某中断的响应时序如图：



- ❖ 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。
- ❖ 80C51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。

5.2.2 中断响应过程

- ❖ 将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）。
- ❖ 执行一条硬件LCALL指令，即把程序计数器PC的内容压入堆栈保存，再将相应的中断服务程序的入口地址送入PC。
- ❖ 执行中断服务程序。

中断响应过程的前两步是由中断系统内部自动完成的，而中断服务程序则要由用户编写程序来完成。

❖ 编写中断服务程序时注意：

- ❖ （1）由于80C51系列单片机两个相邻中断源中断服务程序入口地址相距才8个单元，一般的中断服务程序放不下，通常在相应的中断服务程序入口地址单元放一条长跳转指令LJMP，这样可以使中断服务程序能灵活地安排在64kB ROM的任何地方。若在2kB范围内转移，则可以用AJMP。
- ❖ （2）硬件LCALL指令，只负责将PC内的断点地址压入堆栈保护，而对其他寄存器（如PSW，A等）的内容不作保护处理。所以，在中断服务程序中，首先用软件保护现场，在中断服务之后，中断返回前恢复现场，以防止中断返回后，丢失原来寄存器的内容。

5.2.3 中断返回

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。

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。

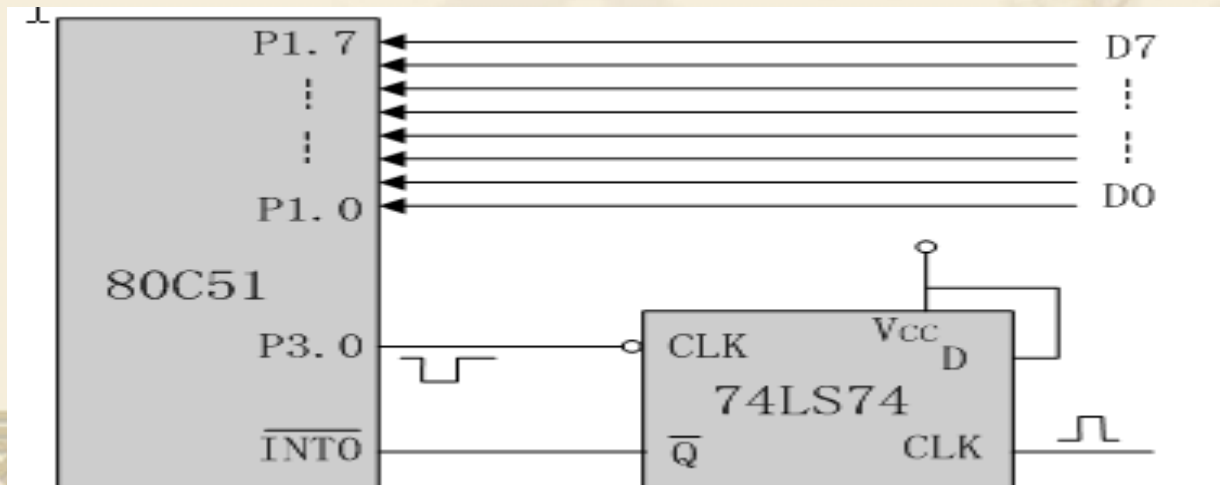
电平触发方式适合于外部中断输入以低电平输入且中断服务程序能清除外部中断请求源的情况。例如，并行接口芯片8255的中断请求线在接受读或写操作后即被复位，因此，以其去请求电平触发方式的中断比较方便。

5.2.4 中断程序举例

例 单外部中断源示例。

图为采用单外部中断源的数据采集系统示意图。将 P1 口设置成数据输入口，外围设备每准备好一个数据时，发出一个选通信号（正脉冲），使 D 触发器 Q 端置 1，经 \overline{Q} 端向 $\overline{INT0}$ 送入一个低电平中断请求信号。如前所述，采用电平触发方式时，外部中断请求标志 IE0（或 IE1）在 CPU 响应中断时不能由硬件自动清除。因此，在响应中断后，要设法撤除 $\overline{INT0}$ 的低电平。撤除 $\overline{INT0}$ 的方法是，将 P3.0 线与 D 触发器复位端相连，只要在中断服务程序中，由 P3.0 输出一个负脉冲，就能使 D 触发器复位， $\overline{INT0}$ 无效，从而清除 IE0 标志。

P3.0输出负脉冲，D触发器复位，D=0, Q#=INT0#1无效，IE0清零。



以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/398110077003006134>