第3章 单片机的中断系统、定时/计数器和串行口

本章将详细介绍单片机的中断系统功能、定时/计数器功能和串行口功能。

3.1 单片机的中断系统

3.1.1 中断系统概念

当中央处理器 CPU 正在处理某事件时外界发生了更为紧急的请求，要求 CPU 暂停当前的工作，转而去处理这个紧急事件，处理完毕后，再回到原来被中断的地方，继续原来的工作，这样的过程称为中断，如图 3-1 所示。实现这种功能的部件称为中断系统（中断机构），产生中断的请求源称为中断源。中断源向 CPU 提出的处理请求，称为中断请求或中断申请。CPU 暂时中止自身的事务，转去处理事件的过程，称为 CPU 的中断响应过程。对事件的整个处理过程，称为中断服务（或中断处理）。处理完毕，再回到原来被中止的地方，称为中断返回。

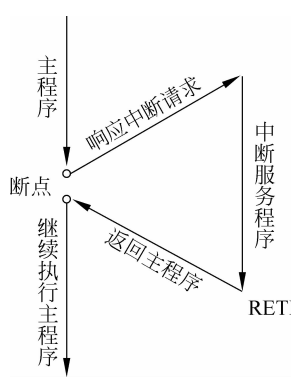

图 3-1 中断流程

MCS-51 系列单片机有 5 个中断源，52 系列单片机有 6 个中断源。单片机的中断系统一般允许多个中断源，当几个中断源同时向 CPU 请求中断时，就存在 CPU 优先响应哪一个中断源请求的问题。

通常根据中断源的轻重缓急排队，优先处理最紧急事件的中断请求源，即规定每一个中断源有一个优先级别，CPU 总是最先响应级别最高的中断。它可分为两个中断优先级，即高优先级和低优先级；可实现两级中断嵌套。用户可以用关中断指令（或复位）来屏蔽所有的中断请求，也可以用开中断指令使 CPU 接收中断申请。即每一个中断源的优先级都可以由程序来设定。

当 CPU 正在处理一个中断源请求时，发生了另一个优先级比它高的中断源请求。如果 CPU 能够暂停对原来的中断源的处理程序，转而去处理优先级更高的中断源请求，处理完以后，再回到原来的低级中断处理程序，这样的过程称为中断嵌套。具有这种功能的中断系统称为多级中断系统，没有中断嵌套功能的则称为单级中断系统。

具有二级中断服务程序嵌套的中断过程如图 3-2 所示。

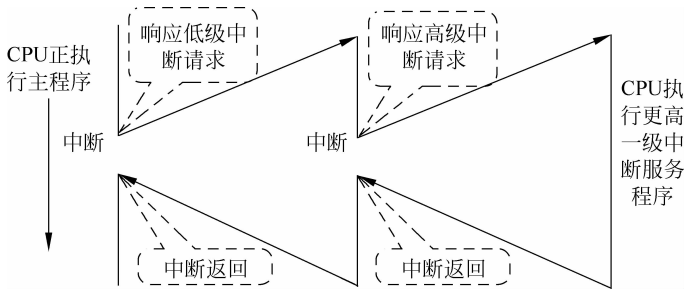

图 3-2 二级中断嵌套示意图

程序控制传送方式中(如查询传送方式),由于是 CPU 主动要求传送数据,而它又不能控制外设的工作速度,因此只能用等待的方式来解决速度匹配的问题。中断方式则是外设主动提出数据传送的请求,CPU 在收到这个请求以前,执行本身的程序(主程序),只是在收到外设希望进行数据传送的请求之后,才中断原有主程序的执行,暂时去与外设交换数据。由于 CPU 工作速度很快,交换数据所花费的时间很短。对于主程序来讲,虽然中断了一个瞬间,由于时间很短,对计算机的运行也不会有什么影响。

中断方式完全消除了 CPU 在查询方式中的等待现象,大大提高了 CPU 的工作效率。

中断方式的另一个应用领域是实时控制。将从现场采集到的数据通过中断方式及时传送给 CPU,经过处理后就可立即作出响应,实现现场控制。而采用查询方式就很难做到及时采集,实时控制。

由于外界异步事件中断 CPU 正在执行的程序(只要允许的话)是随机的,CPU 转去执行中断服务程序时,除了硬件会自动把断点地址(16 位程序计数器的值)压入堆栈之外,用户还得注意保护有关工作寄存器、累加器、标志位等信息(称为保护现场),以便在完成中断服务程序后,恢复原工作寄存器、累加器、标志位等的内容(称为恢复现场)。最后执行中断返回指令,自动弹出断点地址到 PC,返回主程序,继续执行被中断的程序。

3.1.2 MCS-51 中断系统结构

MCS-51 中不同型号单片机的中断源是不同的,最典型的 80C51 单片机有 5 个中断源(8052 有 6 个),具有两个中断优先级,可以实现二级中断嵌套。5 个中断源的排列顺序由中断优先级控制寄存器 IP 和顺序查询逻辑电路(图 3-3 中的硬件查询)共同决定。5 个中断源对应 5 个固定的中断入口地址,也称矢量地址。与中断系统有关的特殊功能寄存器有中断源寄存器(即专用寄存器 TCON、SCON 的相关位)、中断允许控制寄存器 IE 和中断优先级控制寄存器 IP。5 个中断源的中断请求是否会得到响应,要受中断允许寄存器 IE 各位的控制,它们的优先级分别由 IP 各位来确定。MCS-51 基本的中断系统结构如图 3-3 所示。

中断是计算机的一个重要功能。采用中断技术能实现以下的功能。

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,返回到原来的断点处继续运行主

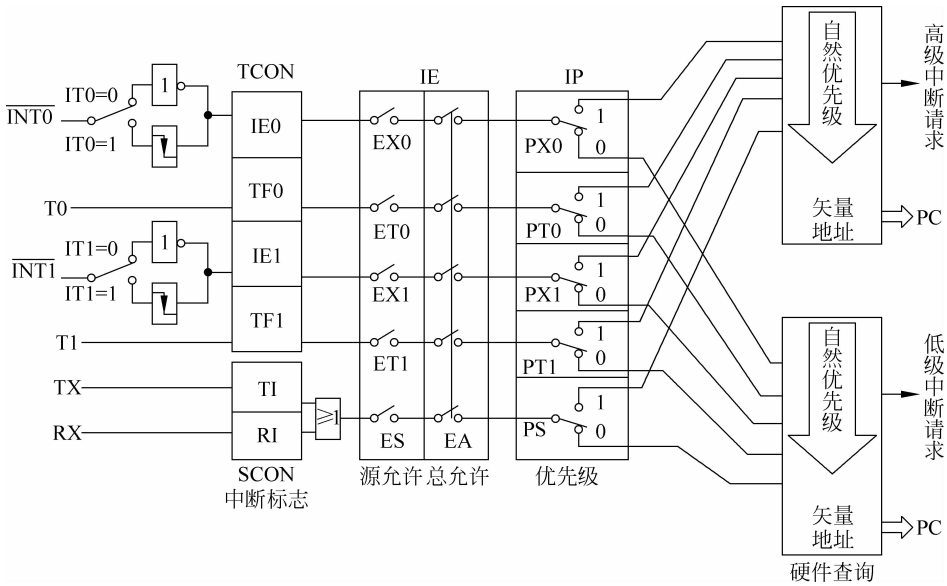

图 3-3　MCS-51 单片机的中断系统

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，从而大大提高了 CPU 的利用率和输入输出的速度。

② 实时处理。当计算机用于实时控制时，请求 CPU 提供服务是随机发生的。有了中断系统，CPU 就可以立即响应并加以处理。

③ 故障处理。计算机在运行时往往会出现一些故障，如电源断电、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。

1. 中断源与中断请求标志

MCS-51 单片机设有 5 个中断源，分别是 2 个外部中断$\overline{\text{INT0}}$、$\overline{\text{INT1}}$，2 个内部定时器/计数器溢出中断 TF0、TF1 和 1 个内部串行口中断 TI 或 RI。这些中断请求分别由特殊功能寄存器 TCON 和 SCON 的相应位锁存。

1）定时器/计数器控制寄存器 TCON

TCON 为定时器/计数器的控制器，它也锁存外部中断请求标志，其格式如下：

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
TCON	TF1		TF0		IE1	IT1	IE0	IT0	字节地址 88H
位地址	8FH		8DH		8BH	8AH	89H	88H	

现对各位说明如下：

(1) TF1(TCON.7)为定时器/计数器 T/C1 的溢出中断请求标志位，位地址为 8FH。T/C1 被启动后，从初始值开始加 1 计数。当 T/C1 产生溢出中断（全“1”变为全“0”）时，TF1 由硬件自动置位（置“1”），向 CPU 申请中断；当 T/C1 的溢出中断为 CPU 响应后，TF1 由硬件自动复位（置“0”或清“0”），中断申请撤除。也可用软件查询 TF1 标志，并由软

件复位。

(2) TF0(TCON.5)为定时器/计数器 T/C0 的溢出中断请求标志位,位地址为 8DH,作用和 TF1 类似。

(3) IE1(TCON.3)为外部中断 1($\overline{INT1}$)的中断请求标志位,位地址为 8BH。当 IT1=0 即电平触发方式时,CPU 在每个机器周期的 S5P2 期间采样$\overline{INT1}$引脚。若$\overline{INT1}$为低电平,则认为有中断申请,随即 IE1 由硬件自动置位;若$\overline{INT1}$为高电平,则认为无中断请求或中断申请已撤除,随即 IE1 由硬件自动复位。当 IT1=1 即边沿触发方式时,CPU 在每个机器周期的 S5P2 期间采样$\overline{INT1}$引脚。若在连续两个机器周期采样到先高电平后低电平,则 IE1 由硬件自动置位,向 CPU 申请中断。当 CPU 响应此中断后,IE1 由硬件自动复位,中断申请撤除。

(4) IT1(TCON.2)为外部中断 1($\overline{INT1}$)的触发控制标志位,位地址为 8AH。IT1 可由软件置位或清 0(SETB IT1 或 CLR IT1)。当 IT1=0,为电平触发方式,$\overline{INT1}$低电平有效;当 IT1=1,$\overline{INT1}$为边沿触发方式,$\overline{INT1}$输入脚上的高到低的负跳变有效。

(5) IE0(TCON.1)为外部中断 0($\overline{INT0}$)的中断请求标志位,位地址为 89H,作用和 IE1 类似。

(6) IT0(TCON.0)为外部中断 0($\overline{INT0}$)的触发控制标志位,位地址是 88H,作用和 IT1 类似。

2) 串行口控制寄存器 SCON

SCON 为串行口的控制器,它的低二位锁存串行口的接收中断和发送中断标志格式如下:

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
SCON							TI	RI	字节地址 98H
位地址							99H	98H	

现对各位说明如下:

(1) TI(SCON.1)为串行口发送中断标志位,位地址为 99H。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(SCON.0)为串行口接收中断标志位,位地址为 98H。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,必须用软件对其清 0。

综上所述,MCS-51 的 5 个中断源的 6 个中断申请标志位是 TF1、TF0、IE1、IE0、TI 和 RI。在 CPU 响应与之对应的中断后,TF1 和 TF0 可由硬件自动复位,TI 和 RI 须在中断服务程序中由软件复位,IE0 和 IE1 只有在中断为边沿触发时,由硬件自动复位。

应当指出:MCS-51 系统复位后,TCON 和 SCON 中各位被复位成“0”状态,应用时要注意各位的初始状态。

2. MCS-51 的中断控制

CPU 对中断源的开放和屏蔽,以及每个中断源是否被允许中断,都受中断允许寄存器 IE 控制。每个中断源优先级的设定则由中断优先级寄存器 IP 控制。寄存器状态可通过程

序由软件设定。

1）中断的开放和屏蔽

MCS-51 没有专门的开中断和关中断指令，中断的开放和关闭是通过中断允许寄存器 IE 进行两级控制的。所谓两级控制是指有一个中断允许总控制位 EA，配合各中断源的中断允许控制位共同实现对中断请求的控制。这些中断允许控制位集成在中断允许寄存器 IE 中。格式如下：

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
IE	EA	×	ET2	ES	ET1	EX1	ET0	EX0	字节地址 A8H
位地址	AF		ADH	ACH	ABH	AAH	A9H	A8H	

中断允许寄存器 IE 的单元地址是 A8H，各控制位（位地址为 A8H～AFH）可以进行字节寻址也可位寻址。所以既可以用字节传送指令又可以用位操作指令来对各个中断请求加以控制。

IE 各位的作用如下。

（1）EA：为 CPU 中断总允许位。EA＝0 时，CPU 关中断，禁止一切中断；EA＝1 时，CPU 开放中断，而每个中断源是开放还是屏蔽分别由各自的允许位确定。

（2）EX0：为外部中断 0（$\overline{\text{INT0}}$）的中断允许位。EX0＝1 时，允许外部中断 0 中断；否则禁止中断。

（3）ET0：为定时器 0（T0 溢出中断）的中断允许位。ET0＝1 时，允许 T0 中断；否则禁止中断。

（4）EX1：为外部中断 1（$\overline{\text{INT1}}$）的中断允许位。EX1＝1 时，允许外部中断 1 中断；否则禁止中断。

（5）ET1：为定时器 1（T1 溢出中断）的中断允许位。TE1＝1 时，允许 T1 中断；否则禁止中断。

（6）ES：为串行口中断允许位。ES＝1 时，允许串行口的接收和发送中断；ES＝0 时，禁止串行口中断。

（7）ET2：为定时器 2 中断允许位。仅用于 52 子系列单片机中，ET2＝1 时，允许定时器 2 中断，否则禁止中断。

（8）×：保留位。

例如，可以采用如下字节传送指令来开放定时器 T0 的溢出中断：

```
MOV  IE,＃82H
```

也可以用位寻址指令，则需采用如下两条指令实现同样功能：

```
SETB  EA
SETB  ET0
```

在 MCS-51 复位后，IE 各位被复位成“0”状态，CPU 处于关闭所有中断的状态。所以，在 MCS-51 复位以后，用户必须通过程序中的指令来开放所需中断。

2）中断优先级别的设定

MCS-51 系列单片机具有两个中断优先级。对于所有的中断源，均可由软件设置为高优先级中断或低优先级中断，并可实现两级中断嵌套。

一个正在执行的低优先级中断服务程序，能被高优先级中断源所中断。同级或低优先级中断源不能中断正在执行的中断服务程序。每个中断源的中断优先级都可以通过程序来设定，由中断优先级寄存器 IP 统一管理，格式如下：

	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	
IP	×	×	PT2	PS	PT1	PX1	PT0	PX0	字节地址 B8H
位地址			BDH	BCH	BBH	BAH	B9H	B8H	

IP 各位的作用如下。

(1) PX0：为外部中断 0($\overline{\text{INT0}}$)优先级设定位。PX0＝1 时，外部中断 0 为高优先级；否则为低优先级。

(2) PT0：为定时器 0(T0)优先级设定位。PT0＝1 时，T0 为高优先级；否则为低先级。

(3) PX1：为外部中断 1($\overline{\text{INT1}}$)优先级设定位。PX1＝1 时，外部中断 1 高优先级；否则为低优先级。

(4) PT1：为定时器 1(T1)优先级设定位。PT1＝1 时，T1 为高优先级；否则为低优先级。

(5) PS：为串行口优先级设定位。PS＝1 时，串行口为高优先级；否则为低优先级。

(6) PT2：为定时器 2 优先级设定位。仅适用于 52 子系列单片机。PT2＝1 时，设定为高优先级；否则为低优先级。

(7) ×：保留位。

当系统复位后，IP 各位均为 0，所有中断源设置为低优先级中断。IP 也是可进行寻址和位寻址的特殊功能寄存器。

在单片机工作时，中断控制由程序来实现，也就是通过程序对上述两个寄存器进行设置，以确定开中断/关中断和每一中断源的优先级别。例如 CPU 开中断可由以下两条指令来实现：

```
SETB  0AFH                ; EA 置 1
```

或

```
ORL IE,＃80H              ; 按位"或",EA 置 1
```

CPU 关中断可由以下两条指令来实现：

```
CLR  0AFH                 ; EA 清 0
```

或

```
ANL IE,＃7FH              ; 按位"与",EA 清 0
```

又如设置外部中断源$\overline{\text{INT0}}$为高优先级，外部中断源$\overline{\text{INT1}}$为低优先级，可由下面指令来实现：

```
SETB  0B8H                ; PX0 置 1
CLR  0BAH                 ; PX1 清 0
```

或

```
MOV  IP,＃000××0×1B       ; PX0 置 1,PX1 清 0
```

3）优先级结构

中断优先级只有高低两级。所以在工作过程中必然会有两个或两个以上中断源处于同一中断优先级。若出现这种情况，内部中断系统对各中断源的处理遵循以下两条基本原则。

① 低优先级中断可以被高优先级中断所中断，反之不能。

② 一种中断（不管是什么优先级）一旦得到响应，与它同级的中断不能再中断它。

为了实现这两条规则，中断系统内部包含两个不可寻址的“优先级激活”触发器。其中一个指示某高优先级的中断正在得到服务，所有后来的中断都被阻断。另一个触发器指示某低优先级的中断正在得到服务，所有同级的中断都被阻断，但不阻断高优先级的中断。

当CPU同时收到几个同一优先级的中断请求时，哪一个的请求将得到服务，取决于内部的硬件查询顺序，CPU将按自然优先级顺序确定应该响应哪个中断请求。其自然优先级由硬件形成，排列如下：

中断源	同级自然优先级
外部中断0	最高级
定时器0中断	↓
外部中断1	⋮
定时器1中断	↓
串行口中断	最低级
定时器2中断	最低级（52系列单片机中）

【例3-1】 设80C51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：（1）用字节操作指令

```
MOV  IP,#05H
```

或

```
MOV  0B8H,#05H
```

（2）用位操作指令

```
SETB  PX0
SETB  PX1
CLR   PS
CLR   PT0
CLR   PT1
```

3. 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。由于各计算机系统的中断系统硬件结构不同，中断响应的方式也有所不同。

1）中断响应

（1）CPU响应中断的条件如下。

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 若现行指令为 RETI 或者是访问 IE 或 IP 指令,该指令以及紧接着的另一条指令已执行完。

例如,CPU 对外部中断的响应,当采用边沿触发方式时,CPU 在每个机器周期的 S5P2 期间采样外部中断输入信号$\overline{\text{INTX}}$(X=0,1),如果在相邻的两次采样中,第一次采样到的$\overline{\text{INTX}}$=1,紧接着第二次采样到的$\overline{\text{INTX}}$=0,则硬件将特殊功能寄存器 TCON 中的 IEX(X=0,1)置 1,请求中断。IEX 的状态可一直保存下去,直到 CPU 响应此中断,进入到中断服务程序时,才由硬件自动将 IEX 清 0。由于外部中断每个机器周期被采样一次,因此,输入的高电平或低电平至少必须保持 12 个振荡周期(一个机器周期),以保证能被采样到。

(2) 中断响应操作过程如下。MCS-51 的 CPU 在每个机器周期的 S5P2 期间顺序采样每个中断源,CPU 在下一个机器周期 S6 期间按优先级顺序查询中断标志,如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相应的中断矢量地址装入 PC,以便进入相应的中断服务程序。

MCS-51 单片机的中断系统中有两个不可编程的“优先级生效”触发器。一个是“高优先级生效”触发器,用以指明已进行高级中断服务,并阻止其他一切中断请求;一个是“低优先级生效”触发器,用以指明已进行低优先级中断服务,并阻止除高优先级以外的一切中断请求。80C51 单片机一旦响应中断,首先置位相应的中断“优先级生效”触发器,然后由硬件执行一条长调用指令 LCALL,把当前 PC 值压入堆栈,以保护断点,再将相应的中断服务程序的入口地址(如外中断 0 的入口地址为 0003H)送入 PC,于是 CPU 接着从中断服务程序的入口处开始执行。

对于有些中断源,CPU 在响应中断后会自动清除中断标志,如定时器溢出标志 TF0、TF1 和边沿触发方式下的外部中断标志 IE0、IE1;而有些中断标志不会自动清除,只能由用户用软件清除,如串行口接收中断标志 RI、发送中断标志 TI。在电平触发方式下的外部中断标志 IE0 和 IE1 则是根据引脚$\overline{\text{INT0}}$和$\overline{\text{INT1}}$的电平变化的,CPU 无法直接干预,需在引脚外加硬件(如 D 触发器)使其自动撤销外部中断请求。

CPU 执行中断服务程序之前,自动将程序计数器的内容(断点地址)压入堆栈保护起来(但不保护状态寄存器 PSW 的内容,也不保护累加器 A 和其他寄存器的内容),然后将对应的中断矢量装入程序计数器 PC,使程序转向该中断矢量地址单元中,以执行中断服务程序。各中断源及与之对应的矢量地址见表 3-1。

表 3-1 中断源及其对应的矢量地址

中 断 源	中断矢量地址	中 断 源	中断矢量地址
外部中断 0($\overline{\text{INT0}}$)	0003H	定时器 T1 中断	001BH
定时器 T0 中断	000BH	串行口中断	0023H
外部中断 1($\overline{\text{INT1}}$)	0013H		

由于 MCS-51 系列单片机的两个相邻中断源中断服务程序入口地址相距只有 8 个单元，一般的中断服务程序是容纳不下的，通常是在相应的中断服务程序入口地址中放一条长跳转指令 LJMP，这样就可以转到 64KB 的任何可用区域了。若在 2KB 范围内转移，则可存放 AJMP 指令。

中断服务程序从矢量地址开始执行，一直到返回指令“RETI”为止。“RETI”指令的操作，一方面告诉中断系统该中断服务程序已执行完毕，另一方面把原来压入堆栈保护的断点地址从栈顶弹出，装入程序计数器 PC，使程序返回到被中断的程序断点处继续执行，如图 3-2 所示。

在编写中断服务程序时应注意以下 3 点。

① 在中断矢量地址单元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中断，在中断返回前再开放中断。

（3）中断响应时间。在实时控制系统中，为了满足控制速度要求，常要弄清 CPU 响应中断所需的时间。

图 3-4 所示为某中断的最快响应过程，C1 周期的 S5P2 前某中断生效，在 S5P2 期间中断请求被锁存到相应标志中去。下一个机器周期 C2 恰逢某指令的最后一个机器周期，且该指令并非 RETI 或任何访问 IE、IP 的指令。于是后面两个机器周期 C3 和 C4 执行硬件 LCALL 指令，C5 周期便进入了中断服务程序。由此可见，MCS-51 从外部中断请求有效（CPU 响应中断）到开始执行中断服务程序的第一条指令（中断入口地址处指令）为止，至少要经历三个完整的机器周期，即第一个机器周期用于查询中断标志位，第二和第三个机器周期用于保护断点自动转入执行一条长转移 LCALL 指令。

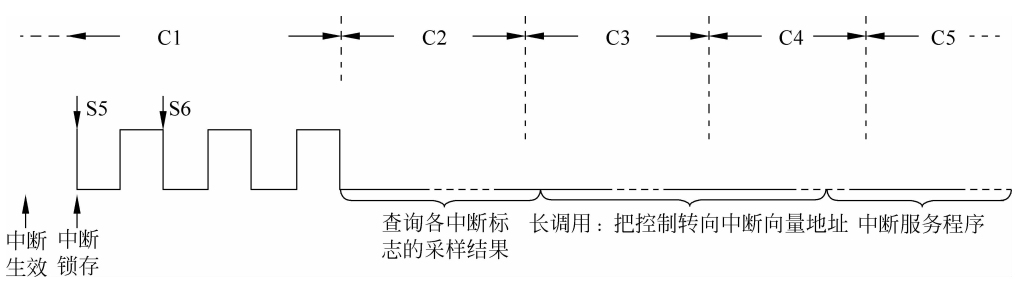

图 3-4　中断响应时序

如果遇到中断受阻的情况，则需要更长的响应时间，例如以下情况。

① 当一个同级或更高级的中断正在进行，则附加的等待时间取决于正在进行的中断程序。

② 如果查询周期不是正在执行的指令的最后一个机器周期，则附加的等待时间为 1～3 个机器周期，因为执行时间最长的指令 MUL 和 DIV 也只有 4 个机器周期。

③ 抛开第一种情况不谈，最不利的情况莫过于查询周期恰逢 RETI 或其他访问 IE、IP

指令的第一个周期，而这类指令后面又跟随着 MUL 或 DIV 指令，由此所引起的附加等待时间不会超过 5 个机器周期（一个周期完成正在进行的指令，再加 MUL 或 DIV 的 4 个周期）。

至此，可以得出结论，在一个单级的中断系统中，MCS-51 响应中断的时间一般在 3～8 个机器周期之间。

2）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。

现场通常有 PSW、工作寄存器和 SFR 等。如果在中断服务程序中要用这些寄存器，则在进入中断服务之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。

中断服务针对中断源的具体要求进行相应的处理。

编写中断服务程序时，需要注意以下几点。

（1）各中断源的入口矢量地址之间只相隔 8 个单元，一般的中断服务程序是容纳不下的，因而最常用的方法是在中断入口地址单元处存放一条无条件转移指令，转至存储器其他的任何空间。

（2）若在执行当前中断程序时禁止更高优先级中断，应用软件关闭 CPU 中断或屏蔽更高级中断源的中断，在中断返回前再开放中断。

（3）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。这样，在编写中断服务程序时，应注意在保护现场之前要关闭中断，在保护现场之后若允许高优先级中断嵌套，则应开中断。同样，在恢复现场之前应关闭中断，恢复之后再开中断。

3）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。

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 执行此指令时，一方面清除中断响应时所置位的“优先级生效”触发器，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。若用户在中断服务程序中进行了压栈操作，则在 RETI 指令执行前应进行相应的出栈操作，使栈顶指针 SP 与保护断点后的值相同。也就是说，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。

3.1.3 中断系统的初始化及应用

1. 中断系统的初始化

MCS-51 中断系统是可以通过 4 个与中断有关的特殊功能寄存器 TCON、SCON、IE 和 IP