

单片机中断系统设计与开发

中断技术是计算机中的重要技术之一,它既和硬件有关,也和软件有关。正因为有了中断,才使得计算机的工作更灵活,效率更高。本节将介绍中断的概念,并以 80C51 系列单片机的中断系统为例介绍中断的处理过程及应用。

3.1 51 单片机的中断系统结构

3.1.1 中断系统的基本概念

1. 中断的概念

中断是通过硬件来改变 CPU 运行方向的一种技术,它既和硬件有关,也和软件有关。计算机在执行程序的过程中,由于单片机内部或外部的某种原因,有必要尽快终止当前程序的执行,而去执行相应的处理程序,待处理结束后,再回来继续执行被终止了的源程序。这种程序在执行过程中由于外界的原因而被中间打断的情况被称为“中断”。

中断之后所执行的处理程序,通常称为“中断服务”或“中断处理子程序”,原来运行的程序称为“主程序”。主程序被断开的位置(地址)称为“断点”。引起中断的原因或能发出中断申请的来源,称为“中断源”。中断源要求服务的请求称为“中断请求(或申请)”。

调用中断服务程序的过程类似于程序设计中的调用子程序,其主要区别在于调用子程序指令在程序中是事先安排好的;而何时调用中断服务程序事先却无法确知。因为中断的发生是由外部因素决定的,程序中无法事先安排调用指令,因而调用中断服务程序的过程是由硬件自动完成的。

2. 引进中断技术的优点

计算机引进中断技术之后,主要具有如下优点。

(1) 分时操作

在计算机与外部设备交换信息时,存在着高速 CPU 和低速外设(如打印机等)之间的矛盾。若采用软件查询方式,则不但占用了 CPU 操作时间,而且相应速度慢。中断功能解决了高速 CPU 与低速外设之间的矛盾。此时,CPU 在启动外设工作后,继续执行主程序,同时外设也在工作。每当外设做完一件事,就发出中断申请,请求 CPU 中断它正

在执行的程序,转去执行中断服务程序(一般是处理输入/输出数据)。中断处理完成后,CPU 恢复执行主程序,外设仍然继续工作。这样,CPU 可以命令多个外设(如键盘、打印机等)同时工作,从而大大提高了 CPU 的工作效率。

(2) 实时处理

在实时控制中,现场的各个参数、信息是随时间和现场情况不断变化的。有了中断功能,外界的这些变化量可根据要求随时向 CPU 发出中断请求,要求 CPU 及时处理,CPU 就可以马上响应(若中断响应条件满足)并加以处理。这样的及时处理在查询方式下是做不到的,从而大大缩短了 CPU 的等待时间。

(3) 故障处理

计算机在运行过程中,难免会出现一些无法预料的故障,如存储出错、运算溢出和电源突跳等。有了中断功能,计算机就能自行处理,而不必停机。

3. 中断源

发出中断请求的来源一般统称为“中断源”。中断源有多种,最常见的有以下 4 种。

(1) 外部设备中断源

计算机的输入/输出设备,如键盘、磁盘驱动器、打印机等,可通过接口电路向 CPU 申请中断。

(2) 故障源

故障源是产生故障信息的来源。它作为中断源,使得 CPU 能够以中断方式对已发生的故障及时进行处理。

计算机故障源有内部和外部之分。内部中断源一般是指执行指令时产生的错误情况,如除法中除数为零等,通常把这种中断源称为“内部软件中断”(注意:目前多数 80C51 系列单片机没有内部软件中断功能);外部故障源主要有电源掉电等情况,在电源掉电时可以接入备用的电池供电电路,以保存存储器中的信息。当电压因掉电而降到低一定值时,即发出中断申请,由计算机的中断系统自动响应,并进行相应处理。

(3) 控制对象中断源

当计算机作实时控制时,被控对象常常用作中断源。例如电压、电流、温度等超过其上限或下限时,以及继电器、开关闭合断开时都可以作为中断源向 CPU 申请中断。

(4) 定时/计数脉冲中断源

定时/计数脉冲中断源也有内部和外部之分。内部定时中断是由单片机内部的定时器/计数器溢出而自动产生的;外部计数中断是由外部脉冲通过 CPU 的中断请求输入线或定时/计数器的输入线而产生的。

要求每个中断源所发出的中断请求信号符合 CPU 响应中断的条件,例如电平高/低、持续时间、脉冲幅度等。

3.1.2 中断系统的结构

中断过程是在硬件基础上再配以相应的软件而实现的。不同计算机的硬件结构和软件指令是不完全相同的,因而中断系统结构一般是不相同的。但同一系列的单片机即使型号不同,中断系统的基本结构也是类似的,只是中断源个数不完全一样。

1. 中断系统的结构

51 单片机中断系统主要由几个与中断有关的特殊功能寄存器、中断入口、顺序查询逻辑电路等组成。中断系统框图如图 3-1 所示。51 单片机有 5 个中断源,可提供两个中断优先级,即可实现二级中断嵌套。与中断有关的特殊功能寄存器有 4 个,分别为中断源寄存器(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共同决定。5 个中断源对应 5 个固定的中断入口地址。

下面分别对 51 单片机的中断源及专用寄存器等进行介绍。

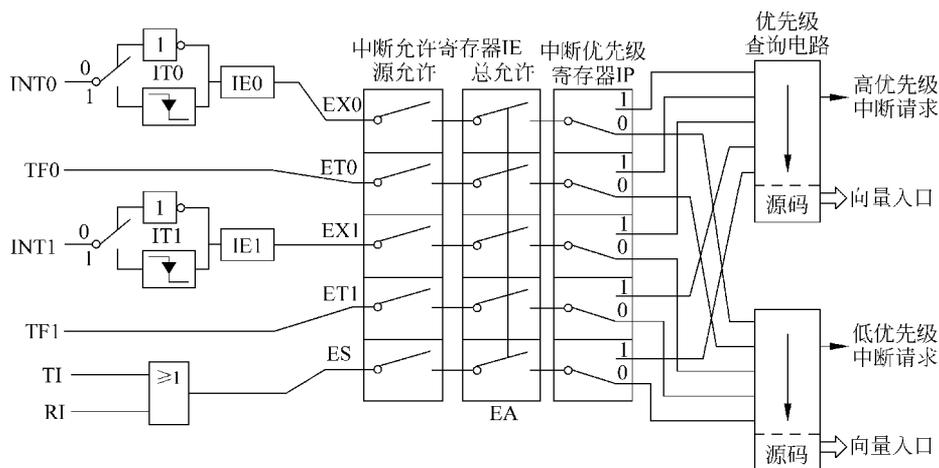


图 3-1 51 单片机的中断系统框图

2. 中断源及入口

(1) 中断源

51 的中断源可以分为 3 类,即外部中断、定时中断和串行口中断。由图 3-1 所示的系统结构可知,51 单片机有 5 个中断请求源,分别为两个外部输入中断源 $\overline{\text{INT0}}$ (P3.2) 和 $\overline{\text{INT1}}$ (P3.3),两个片内定时器 T0 和 T1 的溢出中断源 TF0(TCON.5)和 TF1(TCON.7)。下面分类介绍。

① 外部中断类。外部中断是由外部原因引起的,包括外部中断 0 和外部中断 1。这两个中断请求信号分别通过两个固定引脚即 $\overline{\text{INT0}}$ (P3.2)脚和 $\overline{\text{INT1}}$ (P3.3)脚输入。

外部中断请求信号有两种信号输入方式,即电平方式和脉冲方式。在电平方式下为低电平有效,即在 $\overline{\text{INT0}}$ (P3.2)脚或 $\overline{\text{INT1}}$ (P3.3)脚出现有效低电平时,外部中断标志下降沿有效时,外部中断标志 $\overline{\text{INT0}}$ 或 $\overline{\text{INT1}}$ 就置为 1。

注意: 在脉冲方式下,中断请求信号的高、低电平状态都应该至少维持 1 个机器周期。

中断请求信号是低电平有效还是下降沿有效,必须通过 TCON 寄存器中的 IT0(TCON.0)或 IT1(TCON.2)位来决定。一旦输入信号有效,则向申请中断,并使相应的中断标志位 IE0 或 IE1 置 1。

② 定时中断类。定时中断是为满足定时或计数溢出处理的需要而设置的。

定时方式的中断请求是由单片机内部发生的,输入脉冲是内部产生的周期固定的脉冲信号(1个机器周期),无须在芯片外部设置输入端。

计数方式的中断请求是由单片机内部发生的,脉冲信号由 T0(P3.4)或 T1(P3.5)引脚输入,脉冲下降沿为计数有效信号。这种脉冲周期是不固定的。

当时/计数器中的计数值发生溢出时,表明定时时间或计数值已到。这时以计数溢出信号作为中断请求使溢出标志位置 1,即 T0 中断请求标志 TF0=1,或 T1 中断请求标志 TF1=1。如果允许中断,则请求中断处理。

③ 串行口中断类。串行口中断是为满足串行数据的传送需要而设置的。每当串行口由 TxD(P3.1)端发送串行口中断请求标志 RI 或 TI 置 1,并请求中断。

由图 3-1 可以看出,当这些中断源的中断标志为 1 时,并不一定能引起中断,而是还需要经过 IE 寄存器的控制,才能进入中断请求。

(2) 中断入口

当 CPU 响应某中断源的中断申请之后,CPU 将此中断源的入口地址装入 PC,中断服务程序即从此地址开始执行,因而将此地址称为“中断入口”,亦称为“中断矢量”。在 AT89C51 单片机中,各中断源以及与之对应的入口地址(由单片机硬件电路决定)分配如表 3-1 所示。

表 3-1 各中断源及入口地址的对应关系

中 断 源	入口地址	中 断 源	入口地址
外部中断 0	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	0013H		

所有的 80C51 系列单片机都有上述 5 个中断源。89 系列单片机也下例外,有些型号与其完全相同,如 Philips 公司的 P89C51;有些则增加了新的中断源,如 AT89S52 增加了定时器 T2 中断源,入口地址为 002BH;还有一些型号的中断源多达 9 个,其入口地址按 8 字节一个中断源顺序往下排,可以表达为入口地址=8n+3,n 为中断自然优先级。

3.2 51 单片机的中断控制

中断功能虽然是硬件和软件结合的产物,但用户不必了解中断硬件电路和发生过程。对于用户来说,重点是怎样通过软件管理和应用中断功能。为此,首先应该掌握与中断控制和管理有关的几个寄存器。下面分别对其进行介绍。

3.2.1 中断允许控制寄存器 IE

专用寄存器 IE 是 8051 单片机中的中断允许寄存器,由它控制对中断的开放或关闭。通过向 IE 写入中断控制字,可以实现对中断的二级控制。这里所说的“二级”是指控制字中

有一个中断总允许位 EA。当 EA 为 0 时, CPU 将屏蔽所有的中断申请; 而当 EA 为 1 时, 虽然 CPU 已经开放中断, 但还必须设置相应中断源的控制位, 才可确定允许哪个中断源中断。

IE 各位的格式如下:

IE 地址 (A8H)	AFH	ADH	ACH	ABH	AAH	A9H	A8H	
	EA	—	ET2	ES	ET1	EX1	ET0	EX0

其各位名称及作用如下:

EA(IE. 7) —— 中断允许总控制位。

当 EA=1 时, CPU 开放中断, 每个中断源是被允许还是被禁止, 分别由其各自的允许位决定。

当 EA=0 时, CPU 屏蔽所有的中断申请, 称为“关中断”。

(IE. 6) —— 未使用, 默认为 0。

ET2(IE. 5) —— T2 中断允许位(仅 AT89S52/C52 或类似型号单片机有 ET2)。

当 ET2=1 时, 允许 T2 中断。

当 ET2=0 时, 禁止 T2 中断。

ES(IE. 4) —— 串行口中断控制位。

当 ES=1 时, 允许串行口中断。

当 ES=0 时, 禁止串行口中断。

ET1(IE. 3) —— T1 中断控制位。

当 ET1=1 时, 允许 T1 中断。

当 ET1=0 时, 禁止 T1 中断。

EX1(IE. 2) —— 外部中断 1 控制位。

当 EX1=1 时, 允许外部中断 1 中断。

当 EX1=0 时, 禁止外部中断 1 中断。

ET0(IE. 1) —— T0 中断控制位。

当 ET0=1 时, 允许 T0 中断。

当 ET0=0 时, 禁止 T0 中断。

EX0(IE. 0) —— 外部中断 0 控制位。

当 EX0=1 时, 允许外部中断 0 中断。

当 EX0=0 时, 禁止外部中断 0 中断。

51 单片机复位后, IE 中各中断允许位均被清 0, 即禁止所有中断。

例如, 假定要开放外中断 1 和 T1 的溢出中断, 屏蔽其他中断, 则对应的中断允许控制寄存器内容应为 10001100B, 即中断允许控制字为 8CH。

使用字节操作, 可用一条指令完成。

```
IE = 0x8c;
```

使用位操作指令, 则需以下 3 条指令实现。

```
IE. 2 = 1; IE. 3 = 1; IE. 7 = 1;
```

3.2.2 中断请求标志寄存器 TCON/SCON

当有中断源发出请求时,由硬件将相应的中断标志位置 1。在中断请求被响应前、相应中断标志位被锁存在特殊功能寄存器 TCON 或 SCON 中。

1. 定时器控制寄存器 TCON

TCON 为定时器 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 $\overline{\text{INT0}}$ 和 $\overline{\text{INT1}}$ 的中断标志等。

TCON 中与中断有关的各位如下。

其各位名称及作用如下:

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

(1) TF1(TCON. 7) —— T1 溢出中断标志。

当 T1 开始工作,并且计数值产生溢出时,由硬件使 $\text{TF1}=1$,在中断工作方式下向 CPU 请求中断。此标志一直保持到 CPU 响应中断后,才由硬件自动清 0;也可用软件查询该标志,并由软件清 0。

如果 T1 不工作,或者在工作但没有产生溢出,则 $\text{TF1}=0$ 。

(2) TF0(TCON. 5) —— T0 溢出中断标志。

其操作功能类似于 TF1。

(3) IE1(TCON. 3) —— $\overline{\text{INT1}}$ 外部中断 1 标志。

当 $\text{IE1}=1$ 时,表明外部中断 1 向 CPU 申请中断。

当 $\text{IE1}=0$ 时,表明外部中断 1 没有向 CPU 申请中断。

(4) IT1(TCON. 2) —— 外部中断 1 触发方式控制位。

当 $\text{IT1}=0$ 时,外部中断 1 设置为电平触发方式。在这种方式下,CPU 在每个机器周期的 S5P2 期间对 $\overline{\text{INT1}}$ (P3. 3)引脚采样。若采样为低电平,则认为有中断申请,随即使 IE1 标志置 1;若为高电平,则认为无中断申请或中断申请已撤除,随即清除 IE1 标志。在电平触发方式中,CPU 响应中断后不能自动清除 IE1 标志,也不能由软件清除 IE1 标志,故在中断返回前,必须撤销 $\overline{\text{INT1}}$ 引脚上的低电平,否则将会引起再次中断而出错。

当 $\text{IT1}=1$ 时,外部中断 1 设置为边沿触发方式。CPU 在每个机器周期的 S5P2 期间采样引脚,若在连续两个机器周期采样到先高电平后低电平,则使 IE1 标志置 1,此标志一直保持到 CPU 响应中断时,才由硬件自动清除。在边沿触发方式中,为保证 CPU 在两个机器周期内检测到先高后低的负跳变,则输入高/低电平的持续时间起码要保持 12 个时钟周期。

(5) IE0(TCON. 1) —— $\overline{\text{INT0}}$ 外部中断 0 标志位。

其操作功能与 IE1 类似。

(6) IT0(TCON. 0) —— 外部中断 0 触发方式控制位。

其操作功能与 IT1 类似。

TCON.6 和 TCON.4——中断中未使用。

2. 串行口控制寄存器 SCON

SCON 是串行口控制寄存器,其低两位 TI 和 RI 锁存串行口的接收中断和发送中断标志。

SCON 中与中断有关的各位如下:

SCON							99H	98H
(98H)	—	—	—	—	—	—	TI	RI

其各位名称及作用如下。

(1) TI(SCON.1)——串行发送中断标志。

当 TI=1 时,说明 CPU 将 1 字节数据写入发送缓冲器 SBUF,并且已发送完 1 个串行帧,此时,硬件使 TI 置 1。在中断工作方式下,可以向 CPU 申请中断,在中断和查询工作方式下都不能自动清除 TI,必须由软件清除标志。

当 TI=0 时,说明没有进行串行发送,或者串行发送未完成。

(2) RI(SCON.0)——串行接收中断标志。

当 RI=1 时,在串行口允许接收后,每接收完 1 个串行帧,硬件使 RI 置 1。同样,在中断和查询工作方式下都不会自动清除 RI,必须由软件清除标志。

当 RI=0 时,说明没有进行串行接收,或者串行接收未完成。

注意: SCON 的高 6 位在中断中不用。

3.2.3 中断优先级寄存器 IP

80C51 单片机中断优先级的设定由专用寄存器 IP 统一管理。

它具有两个中断优先级,由软件设置每个中断源为高优先级中断或低优先级中断,可实现二级中断嵌套。

高优先级中断源可中断正在执行的低优先级中断服务程序,除非在执行低优先级中断服务程序时设置了 CPU 关中断或禁止某些高优先级中断源的中断。同级或低优先级的中断源不能中断正在执行的中断服务程序。为此,在 51 中断系统中,内部有两个(用户不能访问的)优先级状态触发器,它们分别指示 CPU 是否在执行高优先级或低优先级中断服务程序,从而决定是否屏蔽所有的中断申请或同一级的其他中断申请。

专用寄存器 IP 为中断优先级寄存器,用于选择各中断源优先级,用户可用软件设定。其各位格式如下:

IP			BDH	BCH	BBH	BAH	B9H	B8H
(B8H)	—	—	PT2	PS	PT1	PX1	PT0	PX0

其各位名称及作用如下。

(1) PT2(IP.5)——T2 中断优先级选择位(仅 AT89S52/C52 或类似型号单片机有)。

当 $PT2=1$ 时,设置定时器 T2 为高优先级中断。

当 $PT2=0$ 时,设置定时器 T2 为低优先级中断。

(2) $PS(IP.4)$ —— 串行口中断优先级选择位。

当 $PS=1$ 时,设定串行口为高优先级中断。

当 $PS=0$ 时,设定串行口为低优先级中断。

(3) $PT1(IP.3)$ —— T1 中断优先级选择位。

当 $PT1=1$ 时,设定定时器 T1 为高优先级中断。

当 $PT1=0$ 时,设定定时器 T1 为低优先级中断。

(4) $PX1(IP.2)$ —— 外部中断 1 中断优先级选择位。

当 $PX1=1$ 时,设定外部中断 1 为高优先级中断。

当 $PX1=0$ 时,设定外部中断 1 为低优先级中断。

(5) $PT0(IP.1)$ —— T0 中断优先级选择位。

当 $PT0=1$ 时,设定定时器 T0 为高优先级中断。

当 $PT0=0$ 时,设定定时器 T0 为低优先级中断。

(6) $PX0(IP.0)$ —— 外部中断 0 中断优先级选择位。

当 $PX0=1$ 时,设定外部中断 0 为高优先级中断。

当 $PX0=0$ 时,设定外部中断 0 为低优先级中断。

当系统复位后,IP 全部清 0,将所有中断源设置为低优先级中断。

如果几个相同优先级的中断源,同时向 CPU 申请中断,CPU 通过内部硬件查询逻辑按自然优先级顺序确定响应哪个中断请求。其自然优先级由硬件形成,排列如表 3-2 所示。

表 3-2 各中断源及其自然优先级

序号	中 断 源	自然优先级
1	外部中断 0	最高级 ↓ 最低级
2	定时器 T0 中断	
3	外部中断 1	
4	定时器 T1 中断	
5	串行口中断	

这种排列顺序在实际应用中很方便、合理。如果重新设置了优先级,则顺序查询逻辑电路将会相应改变排队顺序。例如,如果给 IP 中设置的优先级控制字为 09H,则 $PT1$ 和 $PX0$ 均为高优先级中断,但当这两个中断源同时发出中断申请时,CPU 将首先响应自然优先级较高的 $PX0$ 的中断申请。

例如,某软件中对寄存器 IE、IP 设置如下:

```
IE = 0x8f;
```

```
IP = 0x06;
```

那么,此时在该系统中,CPU 中断允许,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如下。

定时器/计数器 0 > 外部中断 1 > 外部中断 0 > 定时器/计数器 1

3.3 51 单片机的中断处理过程

中断处理过程可分为 3 个阶段,即中断响应、中断处理和中断返回。所有计算机的中断处理都有这样 3 个阶段,但不同的计算机由于中断系统的硬件结构不完全相同,因而中断响应的方式有所不同。在此仅以 51 系列单片机为例来介绍中断处理过程。

3.3.1 中断响应

中断响应是在满足 CPU 的中断响应条件之后,CPU 对中断源中断请求的回答。在这一阶段,CPU 要完成执行中断服务以前的所有准备工作。这些准备工作包括保护断点和把程序转向中断服务程序的入口地址(通常称为“矢量地址”)。

计算机在运行时,并不是任何时刻都会去响应中断请求,而是在中断响应条件满足之后才会响应。

1. CPU 的中断响应条件

CPU 响应中断的条件主要有以下几点。

- (1) 有中断源发出中断请求。
- (2) 中断总允许位 $EA=1$,即 CPU 允许所有中断源申请中断。
- (3) 申请中断的中断源的中断允许位为 1,即此中断源可以向 CPU 申请中断。

以上是 CPU 响应中断的基本条件。若满足,则 CPU 一般会响应中断,但如果有下列任何一种情况存在,则中断响应就会受到阻断。

- (1) CPU 正在执行一个同级或更高级的中断服务程序。

(2) 当前的机器周期不足正在执行指令的最后一个周期,即正在执行的指令完成前,任何中断请求都得不到响应。

(3) 正在执行的指令是返回(RETI)指令或者对专用寄存器 IE、IP 进行读/写的指令,此时在执行 RETI 或者读/写 IE 或 IP 之后,不会马上响应中断请求。至少要再执行一条其他指令,才会响应中断。

存在上述任何一种情况,CPU 都不会马上响应中断。此时,将该中断请求锁存在对应中断源的中断标志位中,然后在下一个机器周期再按顺序查询。

在每个机器周期的 S5P2 期间,CPU 对各中断源采样,并设置相应的中断标志位。CPU 在下一个机器周期 S6 期间按优先级顺序查询各中断标志,如查询到某个中断标志为 1,将在下一个机器周期 S1 期间按优先级进行中断处理。中断查询在每个机器周期中重复执行,如果中断响应的基本条件已满足,但由于上述原因之一而未被及时响应,待封锁中断的条件撤销之后,由于中断标志还存在,故仍会响应。

2. 中断响应过程

如果中断响应条件满足,且不存在中断阻断的情况,则 CPU 将响应中断。

在 51 单片机的中断系统中有两个优先级状态触发器,即“高优先级状态”触发器和

“低优先级状态”触发器。这两个触发器是由硬件自动管理的,用户不能对其编程。当CPU响应中断时,它首先使优先级状态触发器置位,这样可以阻断同级或低级的中断;然后,中断系统自动把断点地址压入堆栈保护(但不保护状态寄存器PSW及其他寄存器内容),再由硬件执行一条长调用指令将对应的中断入口装入程序计数器PC,使程序转向该中断入口地址,并执行中断服务程序。

3.3.2 中断处理

中断处理程序(又称“中断服务”或“中断子程序”)从入口地址开始执行,直到返回指令RETI为止。这个过程称为“中断处理”。此过程主要用于处理中断源的请求,但由于中断处理程序是由随机事件引起的实时响应,从而使得它与一般的子程序存在一定差别。

在编写中断服务程序时,必须注意以下几点。

(1) 注意保护现场和恢复现场,因为一般主程序和中断服务程序都可能会用到累加器、PSW寄存器及其他一些寄存器。CPU进入中断服务程序后,在用到上述寄存器时,就会破坏它原来存在寄存器中的内容,一旦中断返回,将会造成主程序的混乱,因而在进入中断服务程序后,一般要先保护现场,然后再执行中断处理程序,并在返回主程序以前恢复现场。对于要保护的内容一定要全面考虑,不能遗漏。

(2) 在CPU响应中断,使程序转向该中断入口地址后,通常不能从此地址开始运行中断服务程序,因为各入口地址之间只相隔8字节,一般的中断服务是容纳不下的。因此,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,使程序跳转到用户安排的中断服务程序起始地址上去。这样,可使中断服务程序灵活地安排在64KB程序存储器的任何空间。

(3) 若要在执行当前中断程序时禁止更高优先级中断源产生的中断,应先用软件关闭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。

(4) 在保护现场和恢复现场时,为了不使现场数据受到破坏或者造成混乱,一般规定此时CPU不响应新的中断请求。这就要求在编写中断服务程序时,注意在保护现场之前要关中断,在恢复现场之后开中断。如果在中断处理时,允许有更高级的中断打断它,则在保护现场之后再开中断,恢复现场之前关中断。

3.3.3 中断返回

中断返回是指中断服务完成后,计算机返回到断点(即原来断开的位置),继续执行原来的程序。

中断返回由专门的中断返回指令RETI实现。该指令的功能是将断点地址取出,送回到程序计数器PC中。另外,它还通知中断系统已完成中断处理,清除优先级状态触发器,并使部分中断源标志(除TI、RI)清0。在中断服务程序中,要特别注意不能用RET指令代替RETI指令。

3.3.4 中断请求的撤销

CPU响应某中断请求后,在中断返回前应撤销该中断请求,否则会引起另一次中断。