

中断过程与调用子程序过程的比较教学法研究

程启明

(盐城工专高教研究室,盐城,224003)

摘 要 中断过程和调用子程序过程是《微机原理与应用》课程教学的重点和难点。提出把两者结合进来,采用比较的教学方法。详细地对比两者的异同点,实践证明这种方法具有明显的教学效果。

关键词 中断过程 调用子程序过程 比较教学法

分类号 TP39

一、定义与作用

子程序是微机基本结构中的一种,基本程序结构包括顺序(简单)、分支(判断)、循环、子程序和查表共五种。

子程序结构是把一个大程序中的相同程序段抽出称为一个子程序,而抽出的剩余程序称为主程序,并在原抽出处放上一条调用(子程序)指令(LCALL 或 ACALL),子程序最后面加上一条返回(主程序)指令(RET)。这样,微机首先执行主程序,碰到调用指令就转去执行子程序,子程序完后则返回主程序断点,继续执行没有处理完的主程序,这一过程叫做(主程序)调用子程序过程。

子程序结构可简化程序,防止重复书写错误,并可节省内存空间。计算机中经常把常用的各种通用的程序段编成子程序库,提供给用户使用。用户在自己编写的程序中,只要会调用这些子程序,就可大大减化编程的麻烦。

中断是计算机处理单元 CPU 与外设 I/O 交换数据的一种方式,除此方式外,还有无条件、条件(查询)和存储器直接存取(DMA)三种传送方式。由于无条件方式不可靠,条件方式效率低,DMA 方式硬件复杂,而中断方式 CPU 效率高,因此,一般大多采用中断方式。

中断概念是当计算机正在执行某一(主)程序时,收得一外设准备好中断请求,如中断响应条件成立,计算机就把正在执行某一(主)程序暂停一下,去响应处理这一请求,执行中断服务程序,处理完服务程序后,中断返回指令(RET)使计算机返回原来还没有执行完的程序断点处继续执行,这一过程称为中断过程。有了中断,计算机才能具有并行处理、实时处理和故障处理等重要功能。

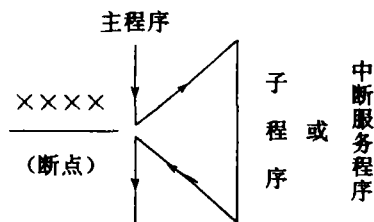
二、联系与区别

中断与调用子程序两过程属于完全不同的概念,但它们也有不少相似之处。两者执行的示意图都可用图来表示。两者都需要保护断点(即下一条指令地址)、跳至子程序或中断服务程

序、保护现场、子程序或中断处理、恢复现场、恢复断点(即返回主程序)。正是由于这些表面上的相似处,很容易使学生把两者混淆起来,特别是把中断也看作为子程序,这就大错特错了。

中断过程与调用子程序过程相似点是表面的,从本质上讲两者是完全不一样的。

两者的根本区别主要表现在服务时间与服务对象不一样上。首先,调用子程序发生时的时间是已知和固定的,即在主程序中的调用指令执行时发生主程序调用子程序,调用指令所在位置是已知和固定的;而中断过程发生的时间一般是随机的,CPU 在执行某一主程序时收到中断源提出的中断申请时就发生中断过程,而中断申请一般由硬件电路产生,申请提出时间是随机的。其次,子程序完全为主服务而存在,主程序是主导者,子程序是辅助者;而主程序与中断服务程序两者一般是完全无关和相互独立的。



三、过程比较

1、调用子程序过程

主程序调用子程序过程应包括保护断点、跳至子程序首址、保护现场、子程序处理、恢复现场、恢复断点(子程序返回)共六步。这六步都是用软件指令完成的,其中前两者保护断点和跳至子程序由调用指令 LCALL 或 ACALL 完成。保护断点其实就是把断点地址推入堆栈中保护起来,要注意的是保护断点应是紧跟着调用指令的下一条指令地址,而不是调用指令本身地址否则会引起重复调用“死循环”,关于这点必须强调一下,提醒学生注意。LCALL 与 ACALL 是 MCS-51 单片机的两条调用指令,它们分别称为长调用和绝对调用,两指令主要区别在于 LCALL 后面操作数直接就是 16 位二进制的子程序入口地址;而 ACALL 后面操作数只是 11 位二进制数,子程序入口地址是把这 11 位数作为低位地址,再加上 ACALL 的下一条指令地址(即断点地址)的高 5 位作为高位地址,从而形成 16 位的子程序入口地址。由此可见,两条调用指令的范围是不一样的,LCALL 为 64KB 程序存储器任何空间范围,而 ACALL 只在高 5 位地址不变的同一页 2KB 范围内。

由于主程序与子程序可能会使用相同的寄存器和存储单元存放数据,如果两者这些数据无关,那么子程序在使用这些相同寄存器和存储单元之前,就必须先把主程序存放在这些区间里的数据现场保护起来,子程序在使用这些区间结束后,再把主程序现场恢复出来,以便子程序返回主程序后,主程序继续使用这些区间。现场保护与恢复一般采用堆栈推入指令 PUSH 与堆栈弹出指令 POP 来实现。当然也可采用改变工作寄存器 R0-R7 区间指针等方法来实现。子程序返回只需用 RET 指令就可把保护在堆栈中的断点恢复出来,继续执行主程序。此外,主程序与子程序是密切联系的,它们之间存在着入口参数与出口参数的数据传递问题,也就是主程序如何把输入参数传达给子程序,子程序又如何把处理结果的输出参数带回给主程序,两者参数传递一般可采用寄存器或存储单元、堆栈、数据指针等方法实现。

子程序再调用子程序叫做子程序嵌套,由于 MCS-51 单片机只能把内部 RAM 地址为 00-7FH 的 128 字节单元作为堆栈使用,断点地址保护到堆栈中需占 2 字节单元,所以子程序最多可嵌套 64 级,当然实际使用中子程序嵌套一般 2-3 级即可,否则程序结构就显得太复

杂了。

2、中断过程

中断过程远比调用子程序过程要复杂,它包括中断请求(或申请)、中断排队、中断响应、中断服务(或处理)和中断返回共五大步。这五大步是由硬件和软件结合实现的。

中断请求由中断源向 CPU 提出。MCS-51 单片机只有定(时)/计(数器)0(溢出)、定(时)/主(数器)1(溢出)、串行口(发送与接收)、外(部)中(断)0 和外(部)中(断)1 等 5 个硬件中断源,其中前三个为内部中断源,后两个为外部中断源。

由于 CPU 在某一时刻只能响应一个中断请求,为处理执行主程序时,同时来了多个中断请求和正在处理某一中断请求时又来了新的中断请求两种情况,计算机采用硬件或软件给各个中断源按优先权大小进行中断排队,从多个中断请求中选出一个级别最高中断请求而响应之,这个过程叫做中断排队。MCS-51 单片机用户可用指令设置高、低两个优先级,而且同级中还有五个固定的隐含优先级,从高到低的优先级顺序分别中外中 0、定/计 0、外中 1、定/计 1、串行口。这样在执行主程序时,同时来了两个以上的中断请求,则先按高低两级选择高优先级,如只有一个高级就直接选之,如有两个以上都是高级或无高级而有两个以上都是低级,则按同级隐含优先级,选择其中一个隐含高级别。如正在执行一个低优先级,又来了一个高优先级,则高优先级中断低优先级,这就形成了两级中断嵌套。如正在执行低的又来了另一个低的或者正在执行高的,又来了一个低的或高的,则不予理睬,不会引起中断嵌套,也就是说同级隐含原则对不同时来的同级中断是无效的。

CPU 在执行任何机器指令时,在每一个机器周期 T_{cy} 都要抽点时间(MCS-51 单片机为 S_3P_2 状态节拍),来采集查询有无中断请求,如没有,则继续执行原程序机器指令,如有中断请求,则先从中选出级别最高者,在中断响应条件成立时,去处理响应此中断请求。

MCS-51 单片机的中断响应必要条件是中断屏蔽总开关 EA 和中断请求相应的屏蔽分开关都必须闭合。因为只有这样,中断请求信号才能送到 CPU。此外,还不能碰到以下三种情况,即有优先级更高的中断请求同时提出或者正在执行同级或高级中断、正在执行的指令还没执行完、正在执行 RETI 或访问中断屏蔽寄存器 IE 和中断排队寄存器 IP 的指令,不碰到这三种情况是中断响应的充分条件。第一种情况中断排队轮不到它响应;第二种情况必须等到现行指令执行完,再响应中断,指令周期分为 $1T_{cy}$ 、 $2T_{cy}$ 、 $4T_{cy}$ 三种机器周期,主计算机在每 T_{cy} 末仅查看一次中断请求,因此 $2T_{cy}$ 和 $4T_{cy}$ 机器指令有可能没有执行完。如果现行指令没有执行完就响应中断,则中断处理返回后,计算机很难把分断开执行的同一条指令贯穿起来;第三种情况处理与第二种情况处理类似,但它除了要把现行的 RETI 指令或访问 IE、IP 指令执行完外,还需再执行下面至少一条指令,方能响应中断,利用这个原理,很容易实现单步操作。

中断服务和中断返回又由关(禁止)同级和低级中断、中断请求撤除、保护断点、跳至中断服务程序入口、保护现场、中断处理、恢复现场、开(允许)同级和低级中断、恢复断点(中断返回)等九小步组成。这九步中断服务过程与前面介绍的六步调用子程序过程类似,只是增加了开、关同级和低级中断和中断请求撤除三步,其余六步完全相似。开、关同级和低级中断目的是使不同时来中断请求时,同级隐含排队原则都无效,此时只能实现高级嵌套低级的两级中断嵌套。在中断处理返回前,清除中断请求原因是防止同一次中断申请被重复响应。

(下转第 33 页)

持吸合,则图4所示单元的SCR的门极通过二极管与阳极相连,SCR相当于一个二极管。三相晶闸管整流桥就相当于一个二极管桥,对PT输出电压,桥路完全开放。

只要发电机端电压高于欠电压继电器的动作值,继电器动作,断开触点。晶闸管的ON、OFF由控制系统决定。

三、应用实例

某自备电源系统中,电源容量 $P_N=120\text{KW}$ 、 $U_N=400/230\text{V}$ 、 $I=216\text{A}$ 、 $n_N=1500\text{rpm}$ 、 $\cos\varphi=0.8$ (滞后)。三只电流互感器经三相二极管桥整流,三只电压互感器经三相桥式全控整流电路整流,两者并联向发电机提供励磁。发电机空载时的励磁电压为32V,励磁电流为10A,满载时励磁电流增加到26A。该电流是由单片机来控制晶闸管的触发角得到的,以实现电机端电压的调节。同时系统中还设有电机欠速超速 $\pm 15\text{rpm}$ 报警电路。该电路在某单位使用后,系统运行稳定,电压调整率 $\Delta U < 2\%$,满足生产要求。

参考文献

- 1 王鉴光主编. 电机控制系统. 机械工业出版社
- 2 单片机原理及应用. 机械工业出版社

(上接第25页)

中断服务过程的前四步是由硬件自动实现的,这点完成不同于调用子程序过程。当然串行口和外中0或外中1的电平触发方式还需用户分别用软件和硬件清除中断请求。MCS-51单片机的中断服务程序入口地址是固定的,它们分别是外中0为0003H、定/计0为000BH、外中1为0013H、定/计1为001BH、串行口为0023H。断点现场保护与恢复也是通过堆栈推入与弹出来实现的。主程序与中断服务程序之间一般无任何关系,两者间不存在参数传递问题,这点也不同于主程序与子程序之间关系。

中断处理完后,就应返回断点继续执行主程序,开同级和低级中断、恢复断点两步可用中断返回指令RETI实现。要注意子程序返回指令RET只能实现恢复断点,而不能打开同级与低级中断,必须把两条返回指令功能区别开来。

中断过程与调用子程序过程既有联系又有区别。本文从多个不同角度详细地对比和比较了它们的异同。实践证明采用这种比较教学方法,教学效果能显著提高。

参考文献

- 1 何立民. MCS-51单片机应用系统设计——系统配置与接口技术. 北京航空航天大学出版社. 1990
- 2 程启明. 微机原理及其应用教学内容和教学方法研究. 全国高校计算机基础教育研究会'94年学术年会论文集. 清华大学出版社. 1994