

手把手教你学单片机(三)

[套件供应]

■周兴华

本讲介绍汇编语言程序设计及指令。

要进行单片机设计,首先要学会编写程序。单片机它必须配合各种各样的软件才能发挥其运算和控制功能。汇编语言是一种采用助记符表示指令、数据和地址来设计程序的语言。例如MOV A,#0AH表示取一个(立即)数0A(十六进制,如转成二进制为00001010)传送到A累加器。汇编语言指令的一般格式如下:

[标号:] 操作码助记符 [第一操作数] [,第二操作数] [,第三操作数] [;注释]

其中带方括号 [...]的部分为可选项。

标号是表示该指令位置的符号地址。它是以英文字母开始的由1~6个字母或数字组成的字符串,并以“:”结尾。通常在子程序入口或转移指令的目标地址处才赋予标号。

操作码助记符是表示指令操作功能的英文缩写。每条指令都有操作码。它是指令的核心部分。

操作数字段表示指令操作所需要的操作数或操作数的地址。操作数字段的表达形式与寻址方式有关。指令的操作数可以是1个、2个或3个,有些指令可能没有操作数。操作数与操作数之间以“,”分隔,操作码与操作数之间以空格分隔。

注释字段是用户给该条指令或该段程序的功能说明,是为了方便阅读程序的一种标注。注释以“;”为开始。注释部分不影响指令的执行。

在学习指令之前,先必须了解

一下单片机的基本知识。

单片机的基本结构组成中包含有中央处理器CPU,程序存储器、数据存储器、输入/输出接口部件,还有地址总线,数据总线和控制总线等。

中央处理器CPU是单片机中的核心部分,其中包含算术逻辑部件(ALU)、控制器、寄存器、累加器A等。算术逻辑部件(ALU)功能是完成算术运算和逻辑运算,算术运算包括加法、减法、加1、减1等操作。逻辑运算包括“与”、“或”、“异或”等操作。控制器的功能是按时间顺序协调各部分的工作,在控制器的控制下,单片机可对指令进行读取、译码,形成各种操作动作,使各个部件之间能协调工作。寄存器包含有程序计数器(PC),PC是专门用来控制指令执行顺序的一个寄存器,可以放16位二进制数码,用来存放指令在内存中的地址。当一个地址码被取出后,PC会自动加1,作好取下一个指令地址码的

存指令执行结果的标志,供程序查询和判别。在PSW的8位中有7个标志位,格式如下:

7	6	5	4	3	2	1	0
CY	AC	FO	RS1	RS0	OV	-	P

P:这是PSW的第0位,它是累加器A的奇偶标志位。P=1表示累加器A中的数为奇数,P=0为偶数。

OV:这是PSW的第2位,称OV为溢出标志,对于带符号的数,在操作时,OV=1表示有溢出,OV=0表示无溢出。

RS1、RS0:这是PSW中的第4位和第3位,用来指示当前使用的工作寄存器区。

FO:用户标志位。可作为软件标志,它的作用和内部RAM位寻址区的各位相似。

AC:辅助进位(半进位)标志。

CY:进位标志。在最高位有进位(做加法运算时)或有借位(做减法时),CY=1,否则CY=0。片内工作寄存器共有4x8=32个,这32个寄存器的地址编号为00H到1FH,分成四个区,每区8个寄存器都用R0~R7来标称。当前使用到的工作寄存器区,可由PSW中的RS1、RS0位指示出来。

下面先学习数据传送指令。

	0区		1区		2区		3区	
	RS1	RS0	RS1	RS0	RS1	RS0	RS1	RS0
	0	0	0	1	1	0	1	1
R0	00H		00H		00H		00H	
R1	01H		01H		01H		01H	
R2	02H		02H		02H		02H	
R3	03H		03H		03H		03H	
R4	04H		04H		04H		04H	
R5	05H		05H		05H		05H	
R6	06H		06H		06H		06H	
R7	07H		07H		07H		07H	

准备工作。累加器A是8位寄存器,它和算术逻辑部件ALU一起完成各种算术逻辑运算,既可以存放运算前的原始数据,又可以存放运算的结果,它是使用最为频繁的一个器件。

程序状态字寄存器PSW是一个非常重要的标志寄存器,用来保

数据传送指令在程序中占有极大比例,非常重要。我们按寻址方式来分类学习数据传送指令,那么什么是寻址呢?获得操作数地址的方式,称为操作数地址的寻址方式,简称寻址方式。

1.立即数寻址。这种寻址方式要传送的操作数在指令的第二字

节中直接给出。因此叫立即数寻址。例如:MOV A,#data #为立即数标志,立即数data的范围是0~FFH。

2. 直接寻址。例如:MOV direct1,direct2 这条指令的含义是将地址direct2单元中的内容传送给地址direct1单元中。

3. 寄存器寻址。例如MOV A,Rn n=0,1,...7。这条指令的含义是将工作寄存器Rn中的内容传送给累加器A。

4. 寄存器间接寻址。例如:MOV A,@Ri i=1,2。这条指令的含义是将Ri中的内容取出,作为另一个单元的地址,将此单元中的内容取出传送到累加器A。

5. 位寻址。例如:MOV C,07H 这条指令的功能是将内部RAM20H单元的D7位(位地址为07H)的内容传送到累加器C中。

6. 变址寻址。变址寻址方式是以DPTR或PC作基址寄存器,以累加器A作变址寄存器,并以两者内容相加形成的16位地址作为操作数的地址。例如:MOVC A,@A+DPTR;其功能是把DPTR和A的内容相加,再把所得到的程序存储器地址单元的内容送A。变址寻址方式只能对程序存储器进行寻址,或者说它是专门针对程序存储器的寻址方式。

7. 相对寻址。相对寻址以程序计数器PC的当前值作为基址,与指令中给定的相对偏移量rel进行相加,把所得之和作为程序的转移地址。例如:JZ 30H 当A=0时,PC←PC+2+rel;若A≠0,则PC←PC+2。

讲了那么多,可能有的读者会说,我还是云里雾里,一头雾水,舍也搞不懂。那也没关系,学单片机一开始就是这样,还是让我们结合程序与实验吧。以后实验做多了,再结合指令学习,就会逐步搞明白的。

打开Keil C51 Windows集成开发环境,按讲座(二)的方法,先在我的文档中建立一个文件目录(如S1),然后建立一个S1.uv2的工程项目,最后建立源程序文件(如S1.asm)。输入以下源程序:

```

序号:1  ORG 0000H;
      2  LJMP MAIN;
      3  ORG 030H;
      4  MAIN: MOV P1,#0DBH;
      5  ACALL DEL;
      6  MOV P1,#06DH;
      7  ACALL DEL;
      8  MOV P1,#0B6H;
      9  ACALL DEL;
     10  AJMP MAIN;
     11  DEL:  MOV R7,#0FFH;
     12  DEL1: MOV R6,#0FFH;
     13  DEL2: DJNZ R6,DEL2;
     14  DJNZ R7,DEL1;
     15  RET;
     16  END;
  
```

编译通过后,将其烧录到89C51芯片中,将芯片插入到S1型LED输出试验板上(配套器材的供应消息见本刊1、2月的配文广告),通电后会发现P1口的8个发光管每隔2个右循环点亮。现在我们结合试验情况来详释一下程序。

序号1(程序解释,以下同):程序开始,ORG是一条伪指令,表示程序从地址0000H开始运行。

序号2:跳转到MAIN主程序处。

序号3:主程序MAIN从地址0030H开始。

序号4:主程序开始后的第一条指令是将立即数DBH传送到P1口,DBH的二进制数为11011011,其中低电平0可将发光管点亮。冒号前的MAIN为标号,表示主程序开始。

序号5:调用延时子程序DEL,维持发光管点亮(延时结束时从子程序返回)。ACALL xx这条调用指令我们还未学到,以后再解释。

序号6:将立即数6DH传送到P1口,6DH的二进制数为

01101101,结果点亮的发光管右移一位。

序号7:调用延时子程序DEL,维持发光管点亮。

序号8:将立即数B6H传送到P1口,B6H的二进制数为11011011,点亮的发光管继续右移一位。

序号9:调用延时子程序DEL,维持发光管点亮

序号10:跳转到主程序处循环。AJMP xx这条转移指令我们还未学到,以后再解释。

序号11:延时子程序开始,将立即数FFH传送给寄存器R7。冒号前的DEL为标号。

序号12:将立即数FFH传送给寄存器R6。同理DEL1为标号。

序号13:将R6中的内容减1后判断,若为0则程序向下执行;若不为0则跳转到DEL2处(即本身)执行。

序号14:将R7中的内容减1后判断,若为0则程序向下执行;若不为0则跳转到DEL处执行。这样将总共做FFH×FFH(即256×256=65536)次减1,用于增大延时时间。

序号15:返回调用子程序处。

序号16:程序结束。END也是一条伪指令,表示程序从这里结束。

现在我们弄明白了,只要改变传送给P1口的立即数,即可控制使哪个发光管亮、哪个发光管灭。改变第一次、第二次、第三次传送给P1口的立即数数字,就可控制点亮的发光管实现右循环或左循环。改变传送给R6、R7寄存器的立即数就可调整延时时间。你可以试一下改变这些立即数,看看有什么变化,看能不能控制一组彩灯。

下面给出MCS-51内部的RAM和特殊功能寄存器SFR的数据传送指令。

1. 以累加器为目的操作数

MOV A,Rn 表示将寄存

器Rn中的内容送累加器A。

MOV A,direct 表示将直接地址direct中的内容送累加器A。

MOV A,@Ri 表示将寄存器Ri中内容作为地址的单元内容送累加器A。

MOV A,#data 表示将立即数data送累加器A。

2. 以寄存器为目的的操作数

MOV Rn,A 表示将累加器A中的内容送寄存器Rn中。

MOV Rn,direct 表示将直接地址单元direct中的内容送寄存器Rn中。

MOV Rn,#data 表示将立即数data送寄存器Rn中。

3. 以直接地址为目的的操作数

MOV direct,A 表示将累加器A中的内容送直接地址单元direct中。

MOV direct,Rn 表示将寄存器Rn中的内容送直接地址单元direct中。

MOV direct2,direct 1 表示将直接地址单元direct1中的内容送直接地址单元direct2中。

MOV direct,@Ri 表示将寄存器Ri中内容作为地址的单元内容送直接地址单元direct中。

MOV direct,#data 表示将立即数data送直接地址单元direct中。

4. 以寄存器间接地址为目的的操作数

MOV @Ri,A 表示将累加器A中的内容送间接地址Ri中。

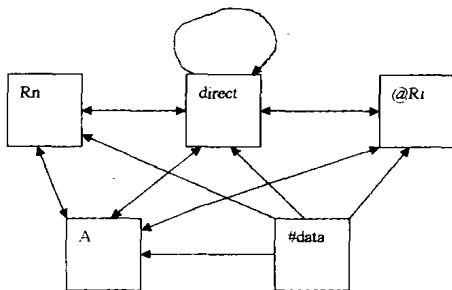
MOV @Ri,direct 表示将直接地址单元direct中的内容送间接地址Ri中。

MOV @Ri,#data 表示将立即数data送间接地址Ri中。

5. 16位数据传送

MOV DPTR,#data16 表示将16位二进制数传送到DPTR寄存器中。

为了学习方便,下面给出MOV



指令中操作数的各种组合关系。

刚才我们做了单片机的输出实验,是不可控的。那么能不能由输入信号去控制输出信号,让单片机受人的控制产生输出信号,做到可控呢?答案是肯定的,下面我们再做一个实验。输入以下源程序。

```

序号:1   ORG 0000H
2       LJMP MAIN;
3       ORG 030H;
4       MAIN: MOV P3,#0C0H;
5       ACALL DEL4MS;
6       MOV P3,#0F0H;
7       MOV A,P3;
8       CJNE A,#0F0H,F1;
9       AJMP MAIN;
10      F1:  MOV P3,#0F9H;
11      ACALL DEL1S;
12      MOV P3,#0A4H;
13      ACALL DEL1S;
14      MOV P3,#0B0H;
15      ACALL DEL1S;
16      MOV P3,#099H;
17      ACALL DEL1S;
18      MOV P3,#092H;
19      ACALL DEL1S;
20      MOV P3,#082H;
21      ACALL DEL1S;
22      MOV P3,#0F8H;
23      ACALL DEL1S;
24      MOV P3,#080H;
25      ACALL DEL1S;
26      MOV P3,#090H;
27      ACALL DEL1S;
28      AJMP MAIN;
29      DEL4MS:MOV R7,#04H;
30      DL0:  MOV R6,#0FFH;
31      DL1:  DJNZ R6,DL1;
32      DJNZ R7,DL0;
33      RET;
34      DEL1S:MOV R5,0FFH;
35      F2:  ACALL DEL4MS;
36      DJNZ R5,F2;
37      RET;
38      END

```

编译通过后,将其烧录到89C51芯片中,将芯片插入到S2型LED数码管输出试验板上,通电后左边第一个数码管显示“0”。这时若按动S1~S12按键的一个,则数码

管以1秒的间隔显示“1、2...9”九个数字,最后回停到“0”。这样就能实现输出的数字由人工输入触发。

下面我们解释一下程序。

序号1(程序解释,以下同):程序开始。

序号2:跳转到MAIN主程序处。

序号3:主程序MAIN从地址0030H开始。

序号4:P3口输出C0H,数码管显示“0”。

序号5:调用4mS延时子程序,维持发光管点亮。

序号6:向P3口送F0H,准备读P3口的输入状态。

序号7:将P3口的输入状态读入累加器A。

序号8:若A的内容(即读入的P3口状态)等于F0H,说明无按键按下,程序向下执行;反之有键按下则程序跳转到F1。

序号9:跳转到主程序处循环。

序号10:将立即数F9H传送给P3口,数码管显示“1”。

序号11:调用延时1S子程序。

序号12:将立即数A4H传送给P3口,数码管显示“2”。

序号13:调用延时1S子程序。

序号14:将立即数B0H传送给P3口,数码管显示“3”。

序号15:延时1S子程序。

序号16:将立即数99H传送给P3口,数码管显示“4”。

序号17:调用延时1S子程序。

序号18:将立即数92H传送给P3口,数码管显示“5”。

序号19:调用延时1S子程序。

序号20:将立即数82H传送给P3口,数码管显示“6”。

序号21:调用延时1S子程序。

序号22:将立即数F8H传送给P3口,数码管显示“7”。

序号23:调用延时1S子程序。

序号24:将立即数80H传送给P3口,数码管显示“8”。

序号25:调用延时1S子程序。

序号26:将立即数90H传送给P3口,数码管显示“9”。

序号27:调用延时1S子程序。

序号28:跳转到主程序处循环。

序号29~33:延时4mS子程序。

序号34~37:延时1S子程序。

序号38:程序结束。

(下一讲继续介绍指令的学习及实验)。

有关Keitc51软件及编译器邮购信息请见上期介绍或打电话向021-64066571咨询。◀