

## 手把手教你学单片机(七)

■周兴华

## 位操作指令学习

MCS-51 单片机内部有一个性能优异的位处理器,实际上是一个一位微处理器,它有自己的位变量操作运算器、位累加器(借用进位标志 CY)和存储器(位寻址区中的各位)等。MCS-51 指令系统加强了对位变量的处理能力,具有丰富的位操作指令,可以完成以位变量为对象的传送、运算、控制转移等操作。位操作指令的操作对象是内部 RAM 的位寻址区,即字节地址为 20H~2FH 单元中连续的 128 位(位地址为 00H~7FH),以及特殊功能寄存器中可以进行位寻址的各位。

位条件转移指令也是位操作指令的子集,已在讲座(六)控制转移类指令中介绍过。下面介绍位变量传送、控制和运算指令。

## 1. 位数据传送指令:

主要用于对位操作累加器 C 进行数据传送,均为双字节指令。

**MOV C,bit** 该指令的功能是将直接寻址位的内容送入位累加器 CY 中,不影响其他标志。

**MOV bit,C** 该指令是将 CY 的内容传送到直接寻址位。

## 2. 位控制修正指令

这类指令的功能分别是清除、取反、置位进位标志 C 或直接寻址位,执行结果不影响其他标志。

**CLR C** 位累加器 CY 清零。

**CLR bit** 直接寻址位清零。

**SETB C** 位累加器 CY 置位。

**SETB bit** 直接寻址位置位。

**CPL C** 位累加器 CY 取反。

**CPL bit** 直接寻址位取反。

## 3. 位逻辑运算指令

这类指令的功能是把进位 C 的内容及直接位地址的内容逻辑“与”、“或”后的操作结果送回到 C 中。斜杠“/”表示对该位取反后再参与运算,但不改变原来的数值。

**ANL C,bit** 位累加器 CY 逻辑“与”直接寻址位,其结果送位累加器 CY 中。

**ANL C,/bit** 直接寻址位取反后同位累加器 CY 逻辑“与”,其结果送位累加器 CY 中。

**ORL C,bit** 位累加器 CY 逻辑“或”直接寻址位,其结果送位累加器 CY 中。

**ORL C,/bit** 直接寻址位取反后同位累加器 CY 逻辑“或”,其结果送位累加器 CY 中。

需要说明的是,在汇编语言级指令格式中,位地址 bit 有多种表示方式:

(1) 直接(位)地址方式,如: 0DH。

(2) 字节地址. 位方式,如: 21H.5。

(3) 寄存器名. 位方式,如 ACC.7,但不能写成 A.7。

(4) 位定义名方式,如:RS0。

(5) 用伪指令 BIT 定义位名方式,如:F1 BIT PSW.1,经定义后,允许在指令中用 F1 来代替 PSW.1。

接下来做实验,感性认识所学这些指令的作用。

现在我们做实验,在 S1 板上实现:将 P1.0 置低电平后,点亮 P2.0 的 LED (即实现将 P1.0 的状态传送到 P2.0)。在我的文档中建

立一个文件目录(S11),然后建立 S11.uv2 的工程项目,最后建立源程序文件(S11.asm)。

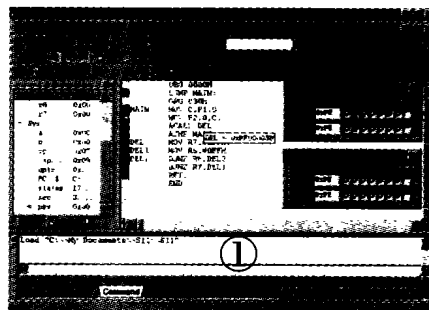
输入下面的程序:

```

序号: 1      ORG 0000H;
        2      LJMP MAIN;
        3      ORG 030H;
        4      MAIN:MOV C,P1.0;
        5      MOV P2.0,C;
        6      ACALL DEL;
        7      AJMP MAIN;
        8      DEL:MOV R7,#0FFH;
        9      DEL1:MOV R6,#0FFH;
       10      DEL2:DJNZ R6,DEL2;
       11      DJNZ R7,DEL1;
       12      RET;
       13      END

```

编译通过后,进行模拟仿真,打开 P1、P3 口,将 P1.0 的打勾取消(即为低电平),按下键盘的 F10,发现 P2.0 也为低电平。若将 P1.0 置高电平,则 P2.0 也为高电平。这样,实现了将 P1.0 的状态传送到 P2.0。软件模拟仿真输出结果见图 1。进行实验时,将 S11 文件夹中的 S11.hex 文件烧录到 89C51 芯片中,将芯片插入到 S1 型 LED 输出试验板上,将 P1.0 口处的短路块拔出(用作输入时,需将输出的短路块拔出),在 S1 实验板上通电运行后,一开始 P2.0 处的 LED 并不亮。将试验线一端插“0”电平插针,另一端插 P1.0 脚外相连的插针,LED 点亮。拿开试验线,LED 又不亮。



我们对程序进行分析解释。

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: 将 P1.0 的外部输入内容传送给位累加器 CY 中。

序号 5: 将位累加器 CY 的内容传送给 P2.0 中。

序号 6: 调用延时子程序, 维持发光管点亮。

序号 7: 跳转到 MAIN 处反复执行。

序号 8~12: 延时子程序。

序号 13: 程序结束。

再做一个实验, 在 S1 板上实现: 比较 P1 口与 P3 口输入的两个数大小。若 P1 数小把 P0.0 点亮; 若 P3 数小则把 P0.7 点亮; 若两数相等, 把 P0.4 点亮。在我的文档中建立一个文件目录(S12), 然后建立 S12.uv2 的工程项目, 最后建立源程序文件(S12.asm)。

输入下面的程序:

```

序号: 1      ORG 0000H
        2      LJMP MAIN;
        3      ORG 030H
4 MAIN:  MOV P0,#0FFH;
        5      MOV P1,#0FFH;
        6      MOV P2,#0FFH;
        7      MOV P3,#0FFH;
        8      MOV A,P1;
        9      CJNE A,P2,L1;
       10      CLR P0.4;
       11      ACALL DEL;
       12      AJMP MAIN;
13 L1:   JC L2;
       14      CLR P0.7;
       15      ACALL DEL;
       16      AJMP MAIN;
17 L2:   CLR P0.0;
       18      ACALL DEL;
       19      AJMP MAIN;
20 DEL:  MOV R7,#0FFH;
21 DEL1: MOV R6,#0FFH;
22 DEL2: DJNZ R6,DEL2;
23      DJNZ R7,DEL1;
24      RET
25      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, 将 P1 口、P3 口处的短路块全部拔出 (此时 P1、P3 的输入均为 FFH), S1 实验板上通电运行。一开始 P0.4 处的 LED 点亮。将试验线一端插“0”电平插针, 另一端插 P1.0 脚外相连的插针, P1 的输入变成 FEH, P1 < P3, P0.0 处的 LED 点亮。反之, 将试验线插 P3.0 脚外相连的插针, P3 的输入变成 FEH, P3 < P1, P0.7 处的 LED 点亮。读者可以自己选择输入大小进行更多的比较实验。程序详解。

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 4~7: P0~P3 口初始化 (均置高电平)。

序号 8: 将 P1 口内容传送给累加器 A。

序号 9: 比较 P2 口内容与累加器 A 中内容之大小。若 P2=A 中内容, 则程序顺序执行, 位累加器 (也称进位位) CY=0。P2≠A 中内容, 则程序跳转至 L1 处, 若 P2<A 中内容, CY=0; 若 P2>A 中内容, CY=1。

序号 10: P0.4 置零, 点亮对应的 LED。

序号 11: 调用延时子程序, 维持 LED 点亮。

序号 12: 跳转到 MAIN 主程序处循环执行。

序号 13: 判位累加器, 若为 1, 跳转至 L2 处; 若为 0, 顺序执行。

序号 14: P0.7 置零, 点亮对应的 LED。

序号 15: 调用延时子程序, 维持 LED 点亮。

序号 16: 跳转到 MAIN 主程序处循环执行。

序号 17: P0.0 置零, 点亮对应的 LED。

序号 18: 调用延时子程序, 维持 LED 点亮。

序号 19: 跳转到 MAIN 主程序处循环执行。

序号 20~24: 延时子程序。

序号 25: 程序结束。

学习、实验做到这个阶段, 大部分的读者已对所学指令有了感

性认识, 也学会了初步的编程。有读者来信问, 是否每行指令结束一定要加分号? 在这里, 我们说, 如你不加注释 (中文或英文), 则不需分号。但若加注释, 则一定要加分号。因为汇编器对分号及后面的该行内容不作编译。通常, 为了使一个程序具有良好的可读性, 注释是必不可缺的。分号是英文的标点符号, 而注释通常是中文的。我们也可以预先在每行结束处加上分号, 以便随时添加注释。总之, 加不加分号应依是否有注释而定, 加了也没错。

另有读者来信来电问: S2 试验板接上 5V 稳压电源后怎么不工作? 回答是: 由于 S2 板上已设计安装了 7805 三端稳压器, 因此输入插座的电压应大于 8V, 而我们提供的 TOP851 编程器 (邮购代号 B1) 配带的电源适配器正好是 9V, 因此只需将编程器 9V 电源插入 S2 板的电源插座即可正常工作。如要用 5V 稳压电源对 S2 板供电还需再焊出引线插座, 反而显得麻烦。S1 板使用 5V 稳压电源供电。

(下一讲继续介绍指令的学习及实验)。

配文优惠 (邮购每次邮费保价费 12 元): Keil 51 Windows 集成开发环境 (已汉化光盘, 邮购代号: K1): 46 元。TOP851 多功能编程器 (邮购代号: B1): 400 元。LED 输出试验板 (邮购代号: S1): 90 元。LED 数码管输出试验板 (邮购代号: S2): 140 元。5V 高稳定专用稳压电源 (邮购代号: D1): 35 元。邮购时只需在附言栏中写明邮购代号及数量并附上联系电话即可。邮购地址: 201103 上海市闵行区莲花路 2151 弄 57 号 201 室

联系人: 吕超亚

电话: 021-64066571

13044152947

技术支持 E-mail: zzh2151@sohu.com