微机原理和接口技术

第六讲 指令系统与汇编程序5

提 纲

1. 指令系统概述
2. 寻址方式
3. 数据传送类指令
4. 算术运算类指令
5. 逻辑运算类指令
6. 控制转移类指令
7. 位操作指令
8. 查表指令的应用
9. 堆栈操作指令的应用
10. 十进制调整指令的应用
11. 逻辑指令与字节状态操作
12. 转移指令的应用

提 纲

10. 十进制调整指令的应用

十进制调整指令的应用

- 在实际应用中，通常要进行BCD码加法运算。但由于计算机只能进行二进制运算，BCD码相加时会出现错误的结果，因此必须对结果进行修正。8051MCU通过十进制调整指令对运算结果修正，可得到正确的BCD加法结果。
- **十进制调整指令： DA A**

功能：对两个压缩BCD码（一个字节存放2位BCD码）数相加的结果进行十进制调整

- **注意：**

1）只能用在ADD和ADDC指令之后，对相加后存放在A中的结果进行修正。

2）两个压缩BCD码按二进制数相加之后，必须经过此指令的调整才能得到正确的BCD码累加和结果。

- **调整的条件和方法：**

1）若（A0~3）>9或（AC）＝1，则（A0~3）←（A0~3）＋6，即低位加6调整。

2）若（A4~7）>9或（Cy）＝1，则高位加6调整。

十进制调整指令的应用

- **例3-16：**2个单字节压缩BCD码相加，(A)=19H, (R0)=19H，试分析程序执行结果。
- ADD A,R0 ;(A)=32H
- DA A
-
- 执行结果： 0001 1001
- 0001 1001
- 0000 0110
- 0011 0010（A）=32H，AC=1
- 0011 1000
- 修正：因为AC=1，所以低4位要+6调整，调整后结果为 38H，得到了正确的BCD码加法结果。

十进制调整指令的应用

- **例3-17**：2个单字节压缩BCD码相加，(A)=89H, (R0)=23H，试分析程序执行结果。

```
ADD     A,R0            ;(A)=ACH
DA      A
```

执行结果： 10001001

00100011

10101100 （A）=ACH，Cy=0，AC=0；

- 修正：由于低4位和高4位均大于9，所以均要+6调整：

10101100

01100110（+66H）

1 00010010

- 调整结果：（A）=12H，Cy=1，调整指令使Cy置1。得到2个BCD数相加的正确结果为112。

提 纲

11. 逻辑指令与字节状态操作

逻辑指令与字节状态操作

在8051MCU指令系统中有许多位操作指令，然而，对于不可位寻址的内存单元来说，要对其中某些位进行清零、置位、取反等操作时，则要借助于逻辑运算指令。

1. 逻辑“与”操作的位屏蔽

“ANL”操作常用来**屏蔽字节中的某些位**，要保留的位用1去“与”（X和1相与为X），**要清除的位用0去“与”**（X和0相与为0）。

例：若(A)=68H，执行ANL A,#0FH指令后，（A）=08H，实现了 高4位清0，低4位保留。

2. 逻辑“或”操作的置位

“ORL”操作常用来对字节中的**某些位置位**，要保留的位用0去“或”（X和0相或为X），**置1的位用1去“或”**(X和1相或为1）。

例：若(A)=68H，执行 ORL A,#0FH指令后，（A）=6FH，实现了高4位保留，低4位置1。

3. 逻辑“异或”操作的求反

“XRL”操作常用来对字节中的**某些位求反**，要保留的位用0去“异或”（X和0相异或为X），**要求反的位用1去异或**(X和1相异或为1）。

例：若(A)=68H，执行 XRL A,#0FH指令后，（A）=67H，实现了高4位保留，低4位求反。

提 纲

12. 转移指令的应用

转移类指令应用

- 8051 MCU指令系统中具有无条件相对转移SJMP和多种条件转移指令。这类指令都是相对PC当前值跳过一个偏移量rel，转移到目的地址执行程序。在实际程序设计中，通常转移的目的地址是确定的，因此要计算转移指令到目的地址的偏移量rel。

rel=目的地址-（转移指令所在地址+转移指令字节数）

rel是一个带符号数的8位二进制补码数，其范围为（**-128**）-（**+127**）

比较麻烦，可以少用。

转移类指令应用

- 散转指令是一条无条件转移指令，JMP @A+DPTR可代替众多的判别跳转指令，具有散转功能。该指令的基址寄存器是DPTR，变址寄存器是A，由于A内容的不同，使程序转移到相对于DPTR偏移量为A内容的地址处，执行分支程序。

散转指令的典型应用是键盘的散转，即根据按键的键值，使程序转去执行该按键的处理程序。例如，对于有16个按键的4x4行列式键盘，每一个按键都有一个对应的键操作程序，当某键被按下时，程序应立即转移到该键对应的处理程序中，因此有16个分支转移的散转操作。

转移类指令应用

散转指令： JMP @A+DPTR

例：利用散转指令，根据按键键值实现散转到各按键的处理程序。

设16个按键的键值为00H-0FH(存放在A中)，设计一个存放16个按键入口地址，表中依次存放16个键的无条件转移指令LJMP KPRGi(i=0 15)。根据A的内容散转到表格的不同位置，再由LJMP KPRGi指令转移到相应的按键处理程序中。

```
KJMP:   MOV    DPTR,#KPRG   ;散转入口地址表的首地址赋给基址寄存器DPTR
        MOV    B,#03H       ;给每个入口地址展宽3字节，以便放3字节LJMP指令
        MUL    AB
        JMP    @A+DPTR      ;散转到入口地址表中
KPRG:   LJMP   KPRG0        ;散转入口地址表，依次存放16个按键程序的转移指令
        LJMP   KPRG1        ; LJMP是3字节指令
        ……
        LJMP   KPRG15
KPRG0:  ……                  ;0号按键操作程序
KPRG1:  ……                  ;1号按键操作程序
……
KPRG15:  ……                 ;15号按键操作程序
```

转移类指令应用

- 8051指令系统具有丰富的比较不等转移指令。在这类指令中，两操作数相比较，如果不相等，则程序跳转一个偏移量到目的地址执行程序。

- **数值比较转移指令（4条）：**

```
CJNE   A,     direct, rel
CJNE   A,     #data，rel
CJNE   Rn,    #data，rel
CJNE   @Ri,   #data，rel
```

- 利用比较指令对进位标志C的影响，可以实现两操作数大小的比较转移。
- 若Cy=1表示第1操作数＜第2操作数；
- 若Cy=0表示第1操作数≥第2操作数。

转移类指令应用

- **例3-22**：某温度控制系统，A中是实际温度Ts，(20H)=温度下限值T20，(30H)=温度上限值T30。若Ts＞T30，程序转降温JW，若Ts＜T20，程序转升温SW，若T30≥Ts≥T20程序转保温BH。

- PROG:　CJNE　A,30H　LOOP
- 　　SJMP　BH　　；等于T30，保温
- LOOP：　JNC　JW　　；大于T30，降温
- 　　CJNE　A,20 H,LOOP1
- 　　SJMP　BH　　；等于T20，保温
- LOOP1：　JC　SW　　；小于T20，升温
- BH：　　------　　；保温
- JW：　　------
- SW：　　------

Thank you!