

实用单片机讲座:

□周兴华

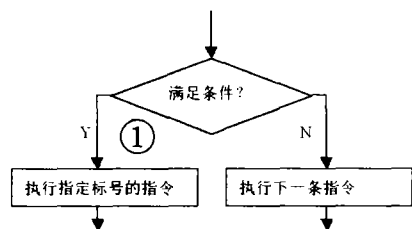
# 手把手教你学单片机(十三)

### 四、分支程序设计

分支程序即根据条件对程序的执行进行判断,满足指定条件则进行程序分支转移。分支程序又可分单分支程序及多分支程序。

#### 1. 单分支程序

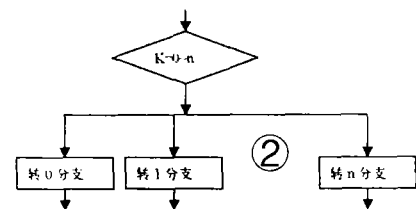
在 MCS-51 指令系统中,通过条件判断实现程序单分支的指令有 JZ、JNZ、CJNZ 和 DJNZ 等,此外还有以位状态作为条件实现程序分支的指令,如:JC、JNC、JB、JNB 和 JBC 等。使用这些指令,可以完成对 0、正负、大小、溢出、状态等各种条件判断。图 1 为单分支程序的状态流程。



#### 2. 多分支程序

多分支程序又称散转程序,在单片机应用程序开发中经常要用到程序多分支散转的情况,因此,散转程序设计技术也是开发者需要掌握的。

散转程序通常是按照数值进行转移,假定分支转移值为 K,则其分支转移状态流程见图 2。

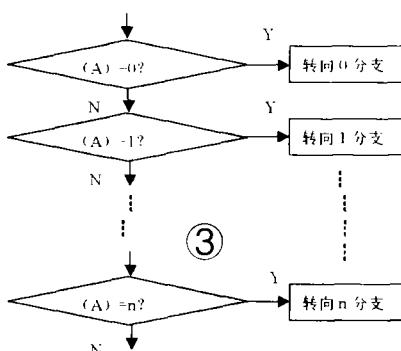


MCS-51 指令系统中并没有多分支转移指令,无法使用一条指令完成多分支转移,要实现多分支转移,可根据情况采用不同的方法,下面分别介绍。

#### 1). 使用 CJNE 指令,通过逐次比较,以转向不同的分支入口。

假定分支转移值在 A 中,则可使用 CJNE A,#data,rel 指令,其分支流程

如图 3 所示。



这种多分支方法的优点是层次清晰,程序简单易懂。但这种方法分支速度较慢,特别是层次较多时。此外分支入口地址应在 8 位偏移量(256)有效范围之内。

#### 2). 使用 JMP

@A+DPTR 指令,转向不同的分支入口。

它是一条无条件程序转移指令,采用基址 + 变址的间接寻址方式。基址为 DPTR,相当于指令转移的基地址,A 为偏移地址,A 中的内容不同,使程序转移到相对于 DPTR 偏移量为 A 中内容的地址处。从该指令看出,散转程序设计的 key 之处在于,在程序中建立一个子程序散转地址表,使数据寄存器 DPTR 指向表的入口地址,然后根据变址寄存器 A 中的内容,用判断指令确定程序跳转到散转地址表中的那一个子程序处。

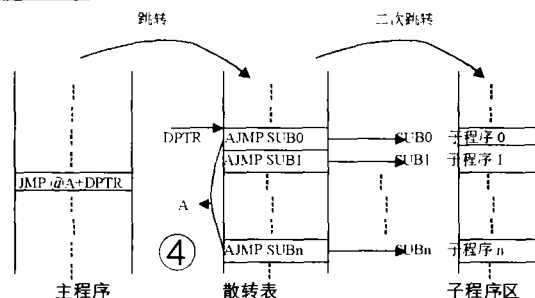
还有一种方法,将 A 清零,根据 DPTR 的值来决定程序转向的目标地址。DPTR 的内容可以通过查表或其它方法获得。

散转子程序设计一般有两个步骤:首先,用 AJMP 或 LJMP 指令构成一个子程序散转地址表,随后用 JMP @A+DPTR 指令使程序跳转到散转地址表相应的位置,从而实现相应子程序跳转(见图 4)。

实际上,散转编程技术与中断编程技术类似,中断是由中断地址表实现应用系统对外界不同随机事件的相应处

理。而散转是用散转表实现对外界随机事件的相应处理。所不同的是中断机制有硬件支持部分,而散转却是由纯软件实现正确的程序转移。所以,也可以说,程序散转是用纯软件实现程序中断的处理。

下面在 S1 板做一个实验,采用散转程序设计。定义 P2.0、P2.1、P2.2、P2.3 为四则运算中的 +、-、×、÷ 命令键,P2.7



为输出显示键,操作数从 P1、P3 口输入,按下输出显示键后结果由 P1、P3 口输出并显示。

在我的文档中建立一个文件目录(S24),然后建立一个 S24.uv2 的工程项目,最后建立源程序文件(S24.asm)。

输入下面的程序:

```

序号: 1   ORG 0000H
      2   AJMP MAIN
      3   ORG 030H
      4   MAIN: MOV R0,#0FFH
      5         MOV DPTR,#TAB
      6   START: JNB P2.0,L0
      7         JNB P2.1,L1
      8         JNB P2.2,L2
      9         JNB P2.3,L3
      10        JNB P2.7,L7
      11        CJNE R0,#0FFH, NEXT
      12        AJMP START
      13   NEXT: MOV A,R0
      14         RL A
      15         JMP @A+DPTR
      16   L0:  ACALL DEL10MS
      17         JB P2.0,START
      18         MOV R0,#00H
      19         AJMP START
      20   L1:  ACALL DEL10MS
      21         JB P2.1, START
      22         MOV R0,#01H
      23         AJMP START
      24   L2:  ACALL DEL10MS
      25         JB P2.2, START
    
```

## MCU Application

```

26      MOV R0,#02H
27      AJMP START
28  L3:  ACALL DEL10MS
29      JB P2.3, START
30      MOV R0,#03H
31      LJMPP START
32  L7:  LCALL DEL10MS
33      JB P2.7, START
34      MOV R0,#04H
35      AJMP START
36  TAB: AJMP SUADD
37      AJMP SUSUB
38      AJMP SUMUL
39      AJMP SUDIV
40      AJMP DIS
41  SUADD:MOV A,P1
42      ADD A,P3
43      MOV R1,A
44      CLR A
45      ADDC A,#00H
46      MOV R3,A
47      AJMP START
48  SUSUB:MOV A,P1
49      CLR C
50      SUBB A,P3
51      MOV R1,A
52      CLR A
53      RLC A
54      MOV R3,A
55      AJMP START
56  SUMUL:MOV A,P1
57      MOV B,P3
58      MUL AB
59      MOV R1,A
60      MOV R3,B
61      AJMP START
62  SUDIV:MOV A,P1
63      MOV B,P3
64      DIV AB
65      MOV R1,A
66      MOV R3,B
67      AJMP START
68  DIS:MOV P1,R1
69      MOV P3,R3
70      ACALL DEL10MS
71      AJMP START
72  DEL10MS:MOV R5,#0BH
73  F1:  MOV R7,#02H
74  F2:  MOV R6,#0FFH
75  F3:  DJNZ R6,F3
76      DJNZ R7,F2
77      DJNZ R5,F1
78      RET
79      END

```

编译通过后,将 S24 文件夹中的 hex 文件烧录到 89C51 芯片中,将芯片插入到 S1 型数码管试验板上,将 P1 口及 P3 口外接的 16 个短路块取下,同时取下 P2.0、P2.1、P2.2、P2.3 及 P2.7 的短路块。接上 5V 电源,用试验线一端接“0”电平,另一端碰触一下 P2.0,进行加法运算。然后将 P1 口及 P3 口外接的 16 个短路块插上(注意 5V 电源不要断开,

让单片机工作)。最后再用试验线碰触一下 P2.7,激活输出,这时 P3、P1 口输出为:0000000111111110。结果正确吗?我们核对一下。P3 口及 P1 口外接的 16 个短路块取下后,由于内部弱上拉电阻的作用,输入均为高电平“0”,即 P3、P1 口的输入均为十进制数 255,作加法后结果为 510,而 P3、P1 口输出 0000000111111110,换算为十进制数正是 510。读者朋友也可自己进行加、减、乘、除的试验。由此实验我们也可看出,自行设计制作一台能作简单的加、减、乘、除的计算器也并不是一件难事,只要将操作数从键盘输入单片机运算并输出驱动数码管显示即可。

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:寄存器 R0 初始化工作,置 FFH。  
 序号 5:散转表首地址送 DPTR。  
 序号 6:若 P2.0 为低电平转 L0。  
 序号 7:若 P2.1 为低电平转 L1。  
 序号 8:若 P2.2 为低电平转 L2。  
 序号 9:若 P2.3 为低电平转 L3。  
 序号 10:若 P2.7 为低电平转 L7。  
 序号 11:R0 内容不等于 FFH 则转 NEXT。  
 序号 12:跳转回 START 处重复执行。  
 序号 13:将 R0 内容送累加器 A。  
 序号 14:A 中内容左移一位,即乘 2。  
 序号 15:根据 A 的内容进行散转。  
 序号 16:调用 10ms 延时子程序。  
 序号 17:再判 P2.0 是否为低电平,若为低电平顺序执行;不为低电平转 START。  
 序号 18:R0 送立即数 00H。  
 序号 19:跳转回 START 处。  
 序号 20:调用 10ms 延时子程序。  
 序号 21:再判 P2.1 是否为低电平,若为低电平顺序执行;不为低电平转 START。  
 序号 22:R0 送立即数 01H。  
 序号 23:跳转回 START 处。  
 序号 24:调用 10ms 延时子程序。  
 序号 25:再判 P2.2 是否为低电平,若为低电平顺序执行;不为低电平转 START。  
 序号 26:R0 送立即数 02H。  
 序号 27:跳转回 START 处。  
 序号 28:调用 10ms 延时子程序。  
 序号 29:再判 P2.3 是否为低电平,若为低电平顺序执行;不为低电平转 START。  
 序号 30:R0 送立即数 03H。  
 序号 31:跳转回 START 处。  
 序号 32:调用 10ms 延时子程序。  
 序号 33:再判 P2.7 是否为低电平,若为低电平顺序执行;不为低电平转 START。  
 序号 34:R0 送立即数 04H。  
 序号 35:跳转回 START 处。  
 序号 36-40:散转地址表,分别散转至加、减、乘、除子程序。  
 序号 41:加法子程序开始,将 P1 口内容读入 A。  
 序号 42:P3 口内容与 A 内容相加,相加结果

在 A 内。  
 序号 43:相加结果(低 8 位)送 R1 暂存。  
 序号 44:清除 A。  
 序号 45:将 A 内容、立即数 00H、进位标志相加。  
 序号 46:相加结果(高 8 位)送 R3 暂存。  
 序号 47:跳转回 START 处。  
 序号 48:减法子程序开始,将 P1 口内容读入 A。  
 序号 49:清除借位标志。  
 序号 50:A 内容减 P3 口内容(即 P1 减 P3),结果存 A。  
 序号 51:A 送 R1 暂存。  
 序号 52:清除 A。  
 序号 53:A 左环移一位,以得到借位标志结果。  
 序号 54:借位结果作为高 8 位显示送 R3 暂存。  
 序号 55:跳转回 START 处。  
 序号 56:乘法子程序开始,将 P1 口内容(被乘数)读入 A。  
 序号 57:将 P3 口内容(乘数)送寄存器 B。  
 序号 58:A 乘 B。  
 序号 59:积的低 8 位暂存 R1。  
 序号 60:积的高 8 位暂存 R3。  
 序号 61:跳转回 START 处。  
 序号 62:除法子程序开始,将 P1 口内容(被除数)读入 A。  
 序号 63:将 P3 口内容(除数)送寄存器 B。  
 序号 64:A 除 B。  
 序号 65:整数商存于 A 中,再送 R1 暂存。  
 序号 66:余数存于 B 中,再送 R3 暂存。  
 序号 67:跳转回 START 处。  
 序号 68:显示子程序开始,将 R1 内容送 P1 口显示。  
 序号 69:将 R3 内容送 P3 口显示。  
 序号 70:调用延时子程序,以便观察清楚。  
 序号 71:跳转回 START 处。  
 序号 72-78:延时子程序。  
 序号 79:程序结束。

## 五、查表程序设计

单片机是一种控制能力强、运算能力有限的计算机,若控制系统要求进行较复杂的数据处理,则可采用事先建立系统的数学模型的方法。如建立  $y=f(x)$  数学模型,并且能估计出数据范围,则可以将数据离散化后建立一个数据表,利用查表指令,求得结果。这样一来,将复杂的数学运算变为简单的查表运算,具有程序简单、执行速度快的优点,弥补了单片机数据运算能力有限的缺点。事实上,有些情况下必须采用查表来进行数据处理,例如,无法建立  $y=f(x)$  的解析式的情况下,就必须用查表方式代替数学运算来进行数据处理(如通过查表得到 LED 七段字段码)。因此查表程序设计技术是单片机重要的软件技术之一。

查表就是根据变量  $x$ ,在表中寻找  $y$ ,使  $y=f(x)$ ,查表程序设计如下:

首先将  $y=f(x)$  函数离散化,在程序存储器中建立一个线性数据表,  $y_0, y_1,$

...,yn。该表一般为最常用的线性表结构,就是用一组连续的存储单元依次存储线性表中的各个元素,查表时根据元素的序号来取出对应存储地址中的数据元素即可完成查表运算,数据表一般存放在程序存储器中。

MCS-51 系列单片机主要用下面两条指令来完成程序存储器查表运算:

```
MOVC A,@A+PC
MOVC A,@A+DPTR
```

两条指令都采用基址加变址的间接寻址方式访问表格中的数据,使用 @A+PC 基址加变址寻址时,PC 是下一条指令的地址,而 A 是从下一条指令首地址到线性数据表中被访问元素的偏移量,因此数据表只能位于当前指令后面的 256 个字节内。使用 @A+DPTR 基址加变址寻址时,DPTR 为常数,是数据表的首地址,A 为从数据表首地址到被访问数据元素的偏移量,线性数据表可以位于程序存储器的任意位置,最具有通用性。

再在 S1 板做个实验,采用查表程序设计。使单片机演奏一段音乐。

单片机演奏音乐的原理是:通过控制定时器的定时来产生不同频率的方波,驱动喇叭后便发出不同音阶的声音,再利用延迟来控制发音时间的长短,即可控制音调中的节拍。把乐谱中的音符和相应的节拍变换为定常数和延迟常数,作成数据表格存放在存储器中,由程序查表得到定时常数和延迟常数,分别用以控制定时器产生方波的频率和发出该频率方波的持续时间。当延迟时间到时,再查下一个音符的定时常数和延迟常数。依次进行下去,就可自动演奏出悦耳动听的音乐。

下面是歌曲“新年好”中的一段简谱:

1=C 1115|3331|1355|432 - |

用定时器 T0 方式 1 来产生歌谱中各音符对应频率的方波,由 P1.7 输出驱动喇叭发声。节拍的控制可通过调用延时子程序 D200(延时 100ms)次数来实现,以每拍 400ms 的节拍时间为例,那么一拍需要循环调用 D200 延时子程序 4 次。同理,半拍就需要调用 2 次。设单片机晶振频率为近似 12MHz,乐曲中的音符、频率及定时常数三者的对应关系如下:

在我的文档中建立一个文件目录 (S25),然后建立一个 S25.uv2 的工程项

C 调音符	5	6	7	1	2	3	4	5	6	7
频率 (Hz)	392	440	494	524	588	660	698	784	880	988
半周期 (ms)	1.28	1.14	1.01	0.95	0.85	0.76	0.72	0.64	0.57	0.51
定时值	FD80	FDC6	FE07	FE25	FE57	FE84	FE98	FEC0	FEE3	FF01

目,最后建立源程序文件 (S25.asm)。

输入下面的程序:

```
序号: 1 ORG 0000H
2 LJMP MAIN
3 ORG 0BH
4 MOV TH0,R1
5 MOV TLO,R0
6 CPL P1.7
7 RETI
8 ORG 0100H
9 MAIN:MOV TMOD,#01H
10 MOV IE,#82H
11 MOV DPTR,#TAB
12 LOOP:CLR A
13 MOVC A,@A+DPTR
14 MOV R1,A
15 INC DPTR
16 CLR A
17 MOVC A,@A+DPTR
18 MOV R0,A
19 ORL A,R1
20 JZ NEXT0
21 MOV A,R0
22 ANL A,R1
23 CJNE A,#OFFH,NEXT
24 SJMP MAIN
25 NEXT:MOV TH0,R1
26 MOV TLO,R0
27 SETB TR0
28 SJMP NEXT1
29 NEXT0:CLR TR0
30 NEXT1:CLR A
31 INC DPTR
32 MOVC A,@A+DPTR
33 MOV R2,A
34 LOOPI:ACALL D200
35 DJNZ R2,LOOPI
36 INC DPTR
37 AJMP LOOP
38 D200:MOV R3,#81H
39 D200B:MOV A,#OFFH
40 D200A:DEC A
41 JNZ D200A
42 DEC R3
43 CJNE R3,#00H,D200B
44 RET
45 TAB:DB 0FEH,25H,02H,0FEH,25H,02H
46 DB 0FEH,25H,04H,0FDH,80H,04H
47 DB 0FEH,84H,02H,0FEH,84H,02H
48 DB 0FEH,84H,04H,0FEH,25H,04H
49 DB 0FEH,25H,02H,0FEH,84H,02H
50 DB 0FEH,0C0H,04H,0FEH,0C0H,04H
51 DB 0FEH,98H,02H,0FEH,84H,02H
52 DB 0FEH,57H,08H,00H,00H,04H
53 END
```

编译通过后,将 S25 文件夹中的 hex 文件烧录到 89C51 芯片中,将芯片插入到

S1 型数码管试验板上,接上 5V 电源,蜂鸣器中立即奏响“新年好”的音乐。

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:定时器 T0 中断入口地址。  
 序号 4-5:定时器 T0 重装初值。  
 序号 6:P1.7 取反。  
 序号 7:定时中断子程序返回。  
 序号 8:主程序从地址 0100H 开始。  
 序号 9:定时器 T0 方式 1。  
 序号 10:允许 T0 中断。  
 序号 11:数据表格首地址。  
 序号 12:清除 A。  
 序号 13:查表。  
 序号 14:定时器初值高 8 位存 R1。  
 序号 15:DPTR 加 1。  
 序号 16:清除 A。  
 序号 17:查表。  
 序号 18:定时器初值低 8 位存 R0。  
 序号 19:A 中内容与 R1 相或。  
 序号 20:若相或结果全 0 为休止符。  
 序号 21:将 R0 内容送 A。  
 序号 22:A 中内容与 R1 相与。  
 序号 23:若相与结果全 1 表示乐曲结束。  
 序号 24:从头开始,循环演奏。  
 序号 25-26:装入定时初值。  
 序号 27:启动定时器 T0。  
 序号 28:跳转至 NEXT1 处。  
 序号 29:关闭定时器 T0,停止发声。  
 序号 30:清除 A。  
 序号 31:DPTR 加 1。  
 序号 32:查延迟常数。  
 序号 33:A 内容送 R2。  
 序号 34:调用 100ms 延时子程序。  
 序号 35:控制延迟次数。  
 序号 36:DPTR 加 1。  
 序号 37:处理下一个音符。  
 序号 38-44:延时 100ms 子程序。  
 序号 45-53:数据表格。  
 序号 54:程序结束。  
 (下一讲学习单片机的键盘接口技术)。

配文优惠供应 (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:zxh2151@sdu.com ◀