51单片机汇编语言教程：第14课-单片机条件转移指令

（基于 HJ-1G、HJ-3G 实验板）

条件转移指令是指在满足一定条件时进行相对转移。

判 A 内容是否为0转移指令

JZ rel

JNZ rel

第一指令的功能是：如果(A)=0，则转移，不然次序执行（执行本指令的下一条指令）。转移到什么地方去呢？如果按照传统的办法，就要算偏移量，很麻烦，好在现在我们能借助于机器汇编了。因此这第指令我们能这样理解：JZ 标号。即转移到标号处。下面举一例说明：

```
MOV A,R0
JZ L1
MOV R1,#00H
AJMP L2
L1: MOV R1,#0FFH
L2: SJMP L2
END
```

在执行上面这段程序前如果 R0中的值是0的话，就转移到 L1执行，因此最终的执行结果是 R1中的值为0FFH。而如果 R0中的值不等于0，则次序执行，也就是执行 MOV R1，#00H 指令。最终的执行结果是 R1中的值等于0。

第一条指令的功能清楚了，第二条当然就好理解了，如果 A 中的值不等于0，就转移。把上面的那个例程中的 JZ 改成 JNZ 试试吧，看看程序执行的结果是什么?

比较转移指令

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

第一条指令的功能是将 A 中的值和立即数 data 比较，如果两者相等，就次序执行（执行本指令的下一条指令），如果不相等，就转移，同样地，我们能将 rel 理解成标号，即：CJNE A,#data,标号。这样利用这条指令，我们就能判断两数是否相等，这在很多场合是非常有用的。但有时还想得知两数比较之后哪个大，哪个小，本条指令也具有这样的功能，如果两数不相

等，则 CPU 还会反映出哪个数大，哪个数小，这是用 CY（进位位）来实现的。如果前面的数（A 中的）大，则 CY=0，不然 CY=1，因此在程序转移后再次利用 CY 就可判断出 A 中的数比 data 大还是小了。

例：

```
MOV A,R0
CJNE A,#10H,L1
MOV R1,#0FFH
AJMP L3
L1: JC L2
MOV R1,#0AAH
AJMP L3
L2: MOV R1,#0FFH
L3: SJMP L3
```

上面的程序中有一条单片机指令我们还没学过，即 JC，这条指令的原型是 JC rel,作用和上面的 JZ 类似，但是它是判 CY 是0，还是1进行转移，如果 CY=1，则转移到 JC 后面的标号处执行，如果 CY=0则次序执行（执行它的下面一条指令）。

分析一下上面的程序，如果（A）=10H，则次序执行，即 R1=0。如果（A）不等于10H，则转到 L1处继续执行，在 L1处，再次进行判断，如果（A）>10H，则 CY=1，将次序执行，即执行 MOV R1，#0AAH 指令，而如果（A）<10H，则将转移到 L2处指行，即执行 MOV R1，#0FFH 指令。因此最终结果是：本程序执行前，如果（R0）=10H，则（R1）=00H，如果（R0）>10H，则（R1）=0AAH，如果（R0）<10H，则（R1）=0FFH。

弄懂了这条指令，其它的几条就类似了，第二条是把 A 当中的值和直接地址中的值比较，第三条则是将直接地址中的值和立即数比较，第四条是将间址寻址得到的数和立即数比较，这里就不详谈了，下面给出几个对应的例程。

```
CJNE A,10H ;把 A 中的值和10H 中的值比较（注意和上题的区别）
CJNE 10H，#35H ;把10H 中的值和35H 中的值比较
CJNE @R0,#35H ;把 R0中的值作为地址，从此地址中取数并和35H 比较
```

循环转移指令

```
DJNZ Rn,rel
DJNZ direct,rel
```

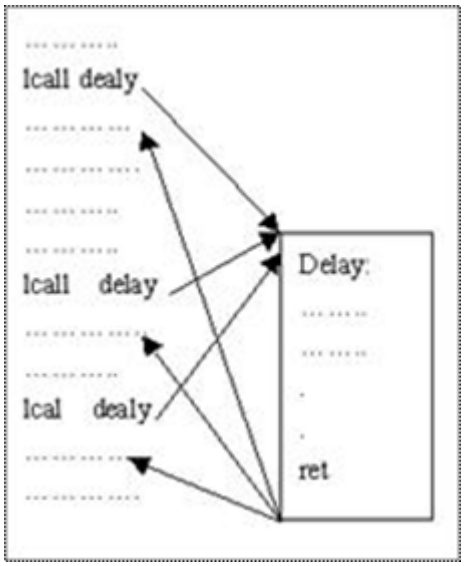
第一条指令在前面的例程中有详细的分析，这里就不多谈了。第二条指令，只是将 Rn 改成直接地址，其它一样，也不多说了，给一个例程。

DJNZ 10H，LOOP

3．调用与返回指令

（1）主程序与子程序 在前面的灯的实验中，我们已用到过了子程序，只是我们并没有明确地介绍。子程序是干什么用的，为什么要用子程序技术呢？举个例程，我们数据老师布置了10道算术题，经过观察，每一道题中都包含一个（3*5+2）*3的运算，我们能有两种选择，第一种，每做一道题，都把这个算式算一遍，第二种选择，我们能先把这个结果算出来，也就是51，放在一边，然后要用到这个算式时就将51代进去。这两种办法哪种更好呢？不必多言。设计程序时也是这样，有时一个功能会在程序的不一样地方反复使用，我们就能把这个功能做成一段程序，每次需要用到这个功能时就“调用”一下。

（2）调用及回过程：主程序调用了子程序，子程序执行完之后必须再回到主程序继续执行，不能“一去不回头”，那么回到什么地方呢？是回到调用子程序的下面一条指令继续执行（当然啦，要是还回到这条指令，不又要再调用子程序了吗？那可就没完没了了……）。参考图1



调用指令

LCALL addr16 ;长调用指令

ACALL addr11 ;短调用指令

上面两条指令都是在主程序中调用子程序，两者有一定的区别，但在开始学习单片机的这些指令时，能不加以区别，而且能用 LCALL 标号，ACALL 标号，来理解，即调用子程序。

（5）返回指令则说了，子程序执行完后必须回到主程序，如何返回呢？只要执行一条返回指令就能了，即执行　ret 指令

4．空操作指令

nop　　就是 空操作，就是什么事也不干，停一个周期，一般用作短时间的延时。

51 实验板推荐(点击下面的图片可以进入下载资料链接)

HJ-1G	HJ-3G