浅谈 CPU 状态字寄存器

雷浩（无锡市北辰自动化技术有限公司）

1. 引言

CPU 寄存器状态字的各位给出了有关指令状态或结果的信息以及所出现的错误，我们可以将二进制逻辑操作状态位信号状态直接集成到程序中，以控制程序执行的流程。

2. 状态字寄存器

先简单介绍一下 CPU 中状态字。

● 首次检查位：状态字的 0 位称作首次检查位，如果/FC 位的信号状态为“0”，则表示伴随着下一条逻辑指令，程序中将开始一个新的逻辑串。FC 前面的斜杠表示对 FC 取反。

● 逻辑运算结果：状态字的第 1 位为 RLO 位（RLO= “逻辑运算结果”），在二进制逻辑运算中用作暂时存储位。比如，一串逻辑指令中的某个指令检查触点的信号状态，并根据布尔逻辑运算规则将检查的结果（状态 位）与 RLO 位进行逻辑门运算，然后逻辑运算结果又存在 RLO 位中。

● 状态位：状态位（第 2 位）用以保存被寻址位的值。状态位总是向扫描指令（A,AN,O,…）或写指令（=,S,R,）显示寻址位的状态（对于写指令，保存的寻址位状态是本条写指令执行后的该寻址位的状态）。

● OR 位：在用指令 OR 执行或逻辑操作之前，执行与逻辑操作的时候，就需要用到 OR 这一状态位。OR 位表示先前执行的与逻辑操作产生的值为“1”，于是， 逻辑操作或的执行结果就已被确定为“1”。

● OV 位：溢出表示算术或比较指令执行时出现了错误。根据所执行的算术或逻辑指令结果对该位进行设置。

● OS 位：溢出存储位是与 OV 位一起被置位的，而且在更新算术指令之后，它能够保持这种状态，也就是说，它的状态不会由于下一个算术指令的结果而改变。 这样，即使是在程序的后面部分，也还有机会判断数字区域是否溢出或者指令是否含有无效实数。OS 位只有通过如下这些命令进行复位：JOS（若 OS = 1，则跳转）命令，块调用和块结束命令。

● CC1 及 CC0 位：CC1 和 CC0（条件代码）位给出有关下列结果的相关信息：

- 算术指令结果
- 比较指令结果
- 字逻辑指令
- 在移位功能中，移出位相关信息。

可以用以下指令来检查条件代码 CC1 和 CC0。

CC1 CC0 检查完成后，如果：

0 0 A == 0 结果 =0

1 0 A > 0 结果 > 0

0 1 A < 0 结果 < 0

● BR 位：状态字的第 8 位称为二进制结果位。它将字处理程序与位处理联系起来，在一段既有位操作又有字操作的程序中，用于表示字逻辑是否正确。将 BR 位加入程序后，无论字操作结果如何，都不会造成二进制逻辑链中断。在梯形图的方块指令中，BR 位与 ENO 位有对应关系，用于表明方块指令是否被正确执行：如果执行出现了错误，BR 位为 0，ENO 位也为 0；如果功能被正确执行，BR 位为 1，ENO 位也为 1。在用户编写的 FB/FC 程序中，应该对 BR 位进行管理，功能块正确执行后，使 BR 位为 1，否则使其为 0。使用 SAVE 指令将 RLO 存入 BR 中，从而达到管理 BR 位目的。

状态字的 9-15 位未使用。

3. 具体使用

下面我们结合 STEP7 中的指针编程来具体介绍条件码 CC0/CC0 的用法。

不同的指令在 CPU 中执行时间是不同的。浮点数比定点数执行时间要长；字逻辑指令比位逻辑指令执行时间要长；在某些程序中适当使用状态字来进行编程可以减少 CPU 程序的执行时间。

例 1：比如说要比较一个 DB 中块的 DBB0-DBB99 这 100 个字节是正数是负数还是 0，正数用 1 来表示；负数用-1 来表示；0 用 0 来表示。并且将对应结果存入 MB200 开始的 100 个字节中。我们通常的做法可能为：

```
        OPN   DB    10
        L     0
        T     MD     0
        L     P#200.0
        T     MD     4
        L     100
CYC:    T     MB   100
        L     DBB [MD 0]
        L     0
        >I
        JNB   NP
        L     1
        T     MB [MD 4]
NP:     NOP   0
        L     DBB [MD 0]
        L     0
        <I
        JNB   NN
        L     -1
        T     MB [MD 4]
NN:     NOP   0
        L     DBB [MD 0]
        L     0
        ==I
        JNB   NO
        L     0
        T     MB [MD 4]
NO:     NOP   0
        L     MD     0
        SLD   3
        T     MD     0
        L     MD     4
        SLD   3
        T     MD     4
        L     MB   100
        LOOP  CYC
```

如果利用条件码来进行编程，既可以减少程序的大小还会减少一定的指令执行时间，我们只需要将中间的比较程序加以优化，即可以达到目的。

无锡市滴翠路100号国家工业设计园创意园A幢903室　　邮编：214072
电话：（0510）85166813，85166823　传真：（0510）85166813　Email：wxbca@163.com

```
        L     DBB [MD 0]
        L     0
        >I
        JNB   NP
        L     1
        T     MB [MD 4]
NP:     NOP   0
        A     <0
        JCN   NN
        L     -1
        T     MB [MD 4]
NN:     NOP   0
        A     ==0
        JCN   NN
        L     0
        T     MB [MD 4]
NO:     NOP   0
```

例 2：根据状态位 C0 和 CC1 的状态而跳转的跳转功能指令 JZ 不改变任何状态位的状态，而且逻辑操作结果 RLO 值也会“随着”该跳转功能带到跳转程序段中，供用户程序其它逻辑操作之用（不改变/FC 状态）。

示例 两个整数相减并需进行连续判断：

```
L MW2
L MW8
-I
JZ ZERO // 如果结果等于“0”，则跳转至标号 ZERO 处
        // 结果不等于“0”时所执行的指令
ZERO:   // 结果等于“0”时，所要执行的指令
```

如果用户不熟悉 JZ 指令和状态位 C0 和 CC1 的具体含义，编程时就需要通过比较指令将比较结果存入一个二进制位中，再根据这个二进制位通过 JC/JCN 指令来控制程序的执行了。

例 3：我们实际应用中可能要利用某些协议转换网关（比如说 Hilscher 公司的 NTTAP 系列网关）来和某些串口协议的仪表进行通信时，会遇到 CRC 校验的问题，关于 CRC 校验时需要判断溢出位是否为 1 的问题来进行程序的进一步计算。我们以 EURO2408 的 MODBUS 通信时需要的 CRC 校验为例说明 CRC 校验的步骤：

1、装载 16#FFFF 到一个 16 位 CRC 寄存器；
2、将 CRC 寄存器的高 8 位字节与信息中的第一个 8 位字节相异或，结果返回到 CRC 寄存器中；
3、将 CRC 寄存器数据向右移动一位；
4、如果溢出的位等于 1，则将 CRC 寄存器与 16#A001 相异或，结果返回到 CRC 寄存器中；
4、如果溢出的位等于 0，则重复第 3 步；
5、重复第 3、4 步骤，直到已经移位了 8 次；
6、将 CRC 寄存器的高 8 位字节与信息中的下一个 8 位字节相异或，结果返回到 CRC 寄存器中；
7、重复第 3 步到第 6 步，直到信息中所有字节都与 CRC 寄存器相异或，并都移位了 8 次；
8、最后的 CRC 寄存器中的结果即为 CRC 校验码，最后被添加到信息（数据）的末尾（交换！低 8 位在前，高 8 位在后；）

在第 4 步中需要判断溢出的位是否为 1，如何判断对于整个程序有着重要的影响。我们可以用 A>0 指令来判断这个条件，具体代码的编写，有兴趣时大家可以根据上面的步骤编写一个自己的 CRC 程序。

4.结束语

在一般情况下，我们不必考虑这些状态位，但在某些情况下，利用这些状态位并结合一定的指令，可以给我们的编程带来更大的灵活性，同时对于进一步提高自己的编程水平也有一定的作用。

5.参考文献

[1].SIEMENS AG.STL 编程手册 V5.3。
[2].SIEMENS AG.S7-300 指令及执行时间。
[3].廖常初.S7-300/400PLC 应用技术。