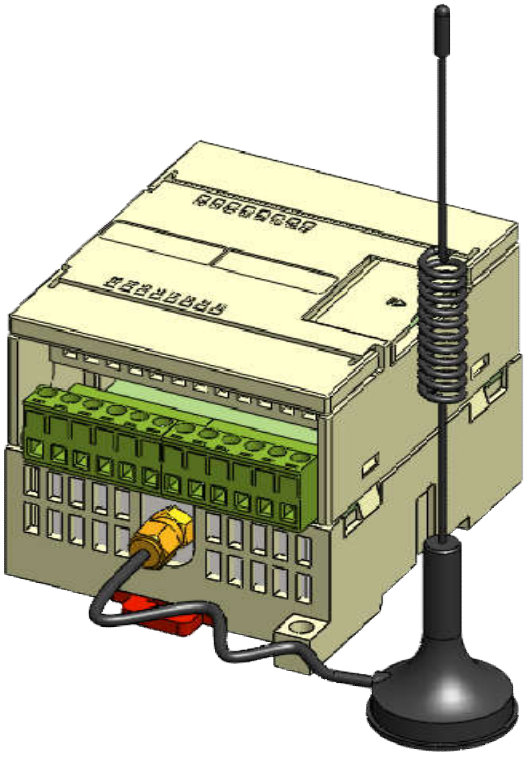
USER'S MANUAL

用户编程手册

梯形图编程版



远程通无线 PLC

F0022E

www.T50rtu.com

T50rtu@sina.com

北京捷麦顺驰科技有限公司

版权声明

北京捷麦顺驰科技有限公司版权所有，并保留对本手册及本声明的最终解释权和修改权。

本手册的版权归北京捷麦顺驰科技有限公司所有。未得到北京捷麦顺驰科技有限公司的书面许可，任何人不得以任何方式或形式对本手册内的任何部分进行复制、摘录、备份、修改、传播、翻译成其它语言、将其全部或部分用于商业用途。

免责声明

本手册依据现有信息制作，其内容如有更改，恕不另行通知。

北京捷麦顺驰科技有限公司在编写该手册的时候已尽最大努力保证其内容准确可靠，但不对本手册中的遗漏、不准确或印刷错误导致的损失和损害承担责任。

我们会经常对手册中的数据进行检查，并在后续的版本中进行必要的更正。欢迎您提出宝贵意见。

技术支持

北京捷麦顺驰科技有限公司建立了以总部技术支持中心、区域技术支持中心和本地技术支持中心为主体的完善的服务体系，并提供电话热线服务。

您在产品使用过程中遇到问题时可随时与北京捷麦顺驰科技有限公司技术支持服务热线联系。

此外，您还可以通过北京捷麦顺驰科技有限公司网站及时了解最新产品动态，以及下载需要的技术文档。

北京捷麦顺驰科技有限公司：

地址：北京市丰台区芳城园一区日月天地B座1505

邮编：100017

电话：010-58076471/2/3

传真：010-58076471

E-mail: support@t50rtu.com

网站：http://www.t50rtu.com

目 录

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

第1章 基本工作原理

- 无线 PLC 的组成
- 工作原理
- 支持的编译语言

1.1 无线 PLC 组成

传统 PLC 通过输入输出电路与现场设备直接连接，并在内部寄存器存储了这些状态，程序直接访问这些寄存器，从而达到利用软件程序寄存器（变量）代替现场硬件状态，再发挥其 CPU 的逻辑计算能力，以完成复杂的控制功能。PLC 组成如图 1-1 无线 PLC 的组成中的灰色部分所示。

无线 PLC 无线 PLC 是在传统的 PLC 基础上在硬件上添加了短信、GPRS 和电台的无线 PLC 信功能，在软件上添加了采集管理，信道管理和驱动的功能。用户可以不用关心无线通信及组网的情况下，实现将现场设备的状态或用户程序变量值数据通过无线通信功能与远端的中心站软件直接对接，中心站软件就可以显示或控制现场设备的状态以及访问无线 PLC 用户程序变量的目的。无线 PLC 无线 PLC 组成如图 1-1 无线 PLC 的组成所示。

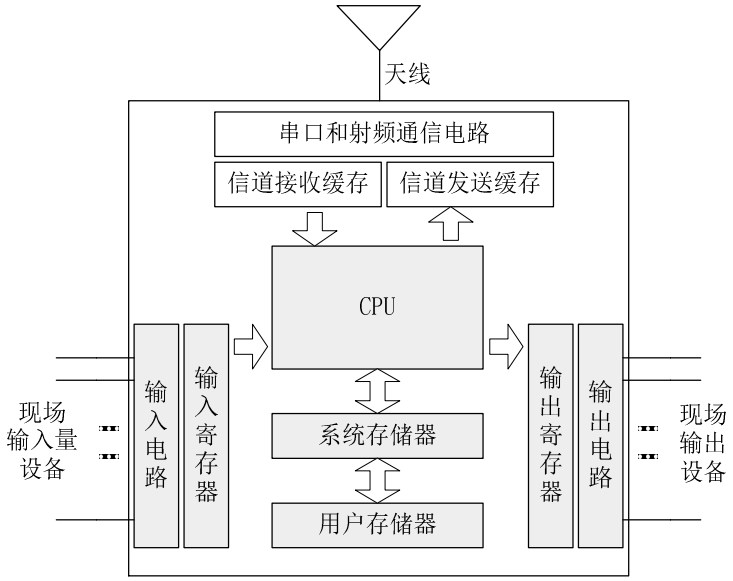


图 1-1 无线 PLC 的组成

✧ **输入寄存器**

输入寄存器的每一个变量对应着一个输入通道物理量，其值反映了输入通道物理量的状态，值的改变由输入电路驱动，并保持一个扫描周期。CPU 可以读其值，但不可以写或者进行修改。

无线 PLC 的输入寄存器有开关量、整形和实数类型。开关量输入寄存器反映当前输入通道的开关量的开关状态；整形输入寄存器反映当前输入通道的计数脉冲的累计值；实数输入寄存器反映当前输入通道的电压、电流或者温度。

✧ **输出寄存器**

输出寄存器的每一个变量值都表明了 PLC 在下一个时间段的输出值（继电器的输出状态），在程序的执行过程中，CPU 可以读其值，并作为条件参加运算，也可以修改其值，而中间的变换仅仅影响寄存器的值，只有程序执行到一个程序尾部时值才影响到下一时间段的输出，即只有最后的修改才对输出节点（继电器）的真实值产生影响。

✧ **存储器**

存储器分为系统存储器和用户存储器。系统存储器存储的是系统程序，它是由厂家开发固化好了的，用户不能更改，PLC 要在系统程序的管理下运行。用户存储器中存放的是用户程序和运行所需要的资源。

✧ **CPU 单元**

CPU 单元控制着 I/O 输入输出寄存器的读写、对存储器单元中程序的解释执行工作、完成无线通信与中心站的对接工作，是 PLC 的大脑，是这些周期任务的执行机构。

✧ **信道通信数据缓存区**

信道通信数据缓存区分为信道接收缓存和信道发送缓存，信道又分成短信、GPRS 和串口（无线电台）三种。当各种信道接收到数据后（收到短信、收到 GPRS 数据包和收到串口数据包），无线 PLC 会将数据提取到对应信道的接收缓冲区中；当各种信道发送数据时（发送短信，发送 GPRS 数据包和发送串口数据），无线 PLC 会将要发送的数据缓存到对应信道的发送缓冲区中，等到通信扫描周期再将数据发送出去。

1.2 工作原理

1.2.1 循环扫描

CPU 连续执行用户程序、任务的循环序列称为扫描。如图 1-2 循环扫描周期所示，CPU 的扫描周期包括读输入、执行程序、处理通信请求、执行 CPU 自诊断测试、写输出和采集管理等内容。

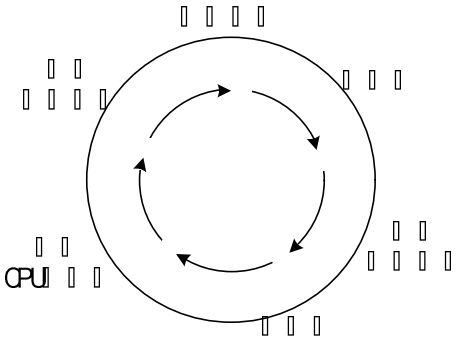


图 1-2 循环扫描周期

PLC 可被看成是在系统软件支持下的一种扫描设备。它一直周而复始地循环扫描并执行由系统软件规定好的任务，用户程序只是扫描周期的一个组成部分，用户程序不运行时，PLC 也在扫描，只不过在一个周期中去除了用户程序执行这个部分内容。PLC 在一个周期中完成了 6 个扫描过程.

✧ **执行 CPU 自诊断测试**

为保证设备的可靠性，及时反映所出现的故障，无线 PLC 具备自监视功能。自监视功能主要由时间监视器（WDT，看门狗）完成。看门狗是一个硬件定时器，每一个扫描周期开始前都被复位（重装）。看门狗的定时值是固定的为 3000ms，当扫描周期中某一个任务执行的时间超过这个定时值，无线 PLC 就会认为设备出现故障，进行相应的故障处理（重启无线 PLC）。

✧ **处理通讯请求**

在扫描周期的通信处理阶段，CPU 将处理有关信道的任务，这一过程用于 PLC 之间及 PLC 与上位机计算机或终端设备之间的通信。

✧ **执行用户程序**

用户的程序为了三个部分，分别为主程序、事件程序和子程序。在扫描周期的执行用户程序阶段，CPU 从头至尾执行用户的主程序。事件程序并不作为正常扫描周期的一部分来执行，而是当事件发生时才执行。子程序是被调用时才执行的。

✧ **处理采集管理**

无线 PLC 无线 PLC 与传统的 PLC 最大的区别就是可以直接构建远程测控（报警）系统，一套完整稳定的远程测控系统，至少具备与 PLC 或分站终端设备之间的采集功能，具备与上位机主站通信交互功能，具备数据超时重发和校验的功能，具备当某通信信道发送故障自动切换到其他通信信道上的功能。

无线 PLC 在正常运行状态下，每一个扫描周期内都包含处理采集管理这个过程。即使用户程序中没有编写任何内容，也不影响无线 PLC 成为一个远程测控系统中的 DTU 设备。

✧ **读输入、写输出**

CPU 在处理用户程序时，使用的输入值不是直接从物理输入点读取的，运算的结果也不直接送至实际物理输出点，而是在内存中设置了两个映射寄存器（系统变量）：一个为输入映射寄存器，另外一个为输出映射寄存器。用户程序中所用的输入值是输入寄存器的值，运算结果也放在输出寄存器中。在输入扫描过程中，CPU 把实际输入点的状态锁入到输入映射寄存器；在输出扫描过程中，CPU 把输出映射寄存器的值锁定到实际物理输出点。

下图 1-3 信号传递过程描述了信号从输入端子到输出端子的传递过程 。

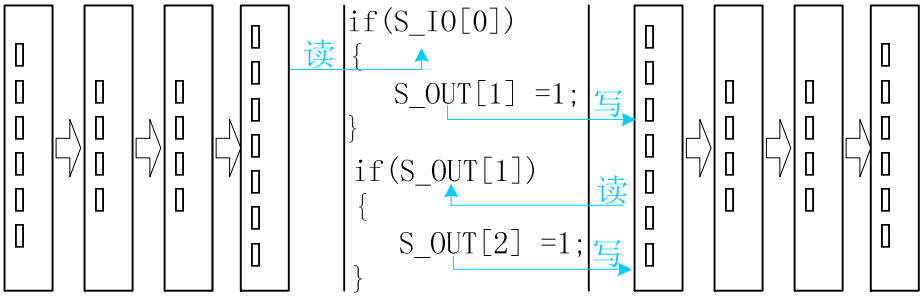


图 1-3 信号传递过程

在读输入阶段，CPU 对各个输入端子进行扫描，通过输入电路将各输入点的状态进行锁入输入映射寄存器中，转入用户程序执行阶段后，CPU 按照先上后下的顺序对每条语句（指令）进行扫描，根据输入映射寄存器和输出映射寄存器的状态执行用户程序，同时将执行结果写入输出映射寄存器中。在用户程序执行期间，即使输入端子的状态发生变化，输入状态寄存器的内容也不会改变（输入状态变化只能在下一个工作周期的输入阶段才能被集中输入）。在写输出阶段，将输出隐射寄存器的状态通过输出电路传递到输出端子。

1.2.2 I/O 响应时间

由于 PLC 采用循环扫描的工作方式，而且对输入和输出信号只在每个扫描周期的固定时间集中

输入/输出，所以必然会产生输出信号相对输入信号滞后的现象。扫描周期越长，滞后现象越严重。对慢速控制系统这是允许的，当控制系统对实时性要求较高时，编程者就必须要面对这个问题。

从输出映射寄存器到输出端子所经历的时间称为输出延迟，它是由硬件决定的，当执行完用户程序后，下一步工作就是将输出映射寄存器的值输出到物理端子上，无线 PLC 的输出延时时间在 0.1ms 以内。

程序执行时间主要由用户程序长短来决定的，对一个实际的控制程序，编程人员须对此部分进行现场测算，使 PLC 的响应时间控制在系统允许的范围内。

在最有利的情况下，输入状态经过一个扫描周期在输出得到响应的时间，称为最小 I/O 响应时间，在最不利的情况下，输入点的状态恰好错过了输入的锁入时刻，造成在下一个输入锁定才能被响应，这就需要两个扫描周期时间，称为最大 I/O 响应时间。它们是由 PLC 的扫描执行方式决定的，与编程方法无关。因此，输入状态要想得到响应，开关量信号宽度至少要大于一个扫描周期才能保证被 PLC 采集到。

1.3 编程语言

无线 PLC 无线 PLC 可支持用户通过“简易 C 语言”、“梯形图”和“STL”三种语言编程的用户程序。

1.3.1 简易 C 语言

我公司结合无线 PLC 自身的特点去掉了一些标准 C 语言中不实用和不易理解的语法和语句，将这种简化的 C 语言称之为简易 C 语言。简易 C 语言去掉了标准 C 语言中指针、for 循环语句和 switch-case 选择语句，其他的语法及语句完全遵守标准 C 语言，简易 C 语言是标准 C 语言的子集。

如果用户熟悉 C 语言编程，那么只需要在无线 PLC 的编程中，不使用指针，将使用 for 循环语句的程序段替换成 while 循环语句，将使用 switch-case 选择语句的程序段替换成 if-else 选择语句即可。

如果用户没有 C 语言的基础，可以通过学习我们提供的《简易 C 语言程序说明》，可在短的时间上手无线 PLC 的编程。

```
if (SysInitFlag ==1)// 该变量为系统初始化标志，在首次扫描时为 1，后自动变成 0（SM0.1）
{//调用串口发送字符串系统函数，打印欢迎界面
        UartSendStr("welcome to the world of CM01P !");
          S_Time31Set   = 20;//设定时间值(定时器 31 的分辨率为 100ms),20*100 = 2000mS
         S_Time31Mode = 1 ; //使用自动重装，循环计数
         S_Time31Able = 1 ; //开始计时
```

```
}
```

注意：

简易 C 语言不区分大小写，因此语句、变量和函数等在编程的时不用考虑大小写的问题。

1.3.2 梯形图编程

梯形图与电器控制系统的电路图很相似，具有直观易懂的优点，很容易被工厂电气人员掌握，特别适用于开关量逻辑控制。

逻辑控制是分段的，程序在同一时间执行一段，从左到右，从上到下。下图给出了梯形图程序的一个例子。不同的指令用不同的图形符号表示。它包括三种基本形式。

触点代表逻辑输入条件，例如；开关、按钮或者内部条件等。

线圈通常表示逻辑输出结果，例如：灯负载、电机启动器、继电器或者内部输出条件。

盒表示其它一些指令，例如：定时器、计数器、通信或者数学运算指令。

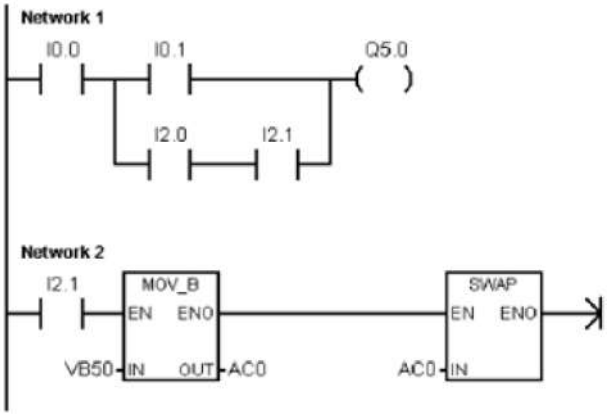


图 1-4 梯形图示例代码

1.3.3 STL 语言

STL 编辑器按照文本语言的形式显示程序。STL 语言可以创建用梯形图无法创建的程序。这是因为您在使用无线 PLC 的梯形图进行编程，为了正确地画出图形，必须遵守一些规则，而有些程序语句是不满足这些规则的。

STL 编写的程序下图 1-5STL 语言示例代码所示，文本方式与汇编语言的编程方式十分相象。

```
LD    I0.0    //读入一个输入
A     I0.1    //和另一个输入进行 “与”
=     Q1.0    //向输出1写入值
```

图 1-5STL 语言示例代码

无线 PLC 从上到下按照程序的次序执行每一条指令，然后回到程序的开始重新执行。STL 使用一个逻辑堆栈来分析控制逻辑。您插入 STL 指令来处理堆栈操作。

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

第2章 无线 PLC 的基本概念（梯形图）

2.1 数据的存取

无线 PLC 将信息存于不同的存储器单元，每个单元都有唯一的地址。您可以明确指出要存取的存储器地址。这就允许用户程序直接存取这个信息。下表列出了不同长度的数据所能表示的数值范围。

表 2-1 不同长度的数据表示的十进制和十六进制数范围

数制	字节（B）	字（W）	双字（D）
无符号整数	0到255 0到FF	0到65,535 0到FFFF	0到4,294,967,295 0到FFFF FFFF
符号整数	-128到＋127 80到7F	-32,768到 +32,767 8000到7FFF	-2,147,483,648到 +2,147,483,647 8000 0000到7FFF FFFF
实数IEEE 32位浮点数	不用	不用	+1.175495E-38到 +3.402823E+38（正数） -1.175495E-38到 -3.402823E+38（负数）

若要存取存储区的某一位，则必须指定地址，包括存储器标识符、字节地址和位号。图 4-3 是一个位寻址的例子（也称为"字节.位寻址）。在这个例子中，存储器区、字节地址（I 代表输入，3 代表字节 3）和位地址（第 4 位）之间用点号（"."）相隔开。

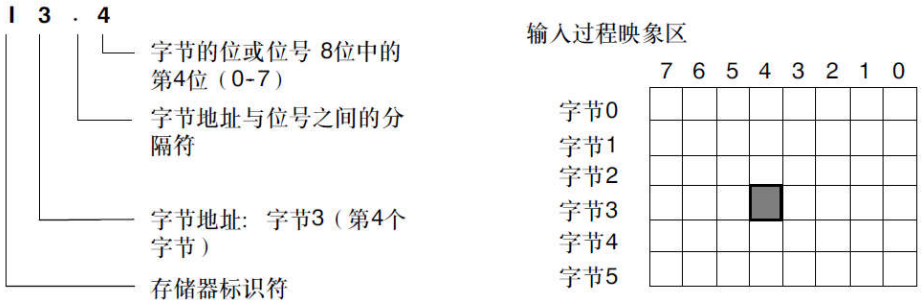


图 2-1 字节.位寻址

使用这种字节寻址方式，可以按照字节、字或双字来存取许多存储区（V、I、Q、M、S、L 及 SM）中的数据。若要存取一个字节、字或双字数据，则必须以类似位寻址的方式给出地址，包括存储器标识符、数据大小以及该字节、字或双字的起始字节地址，如下图所示。

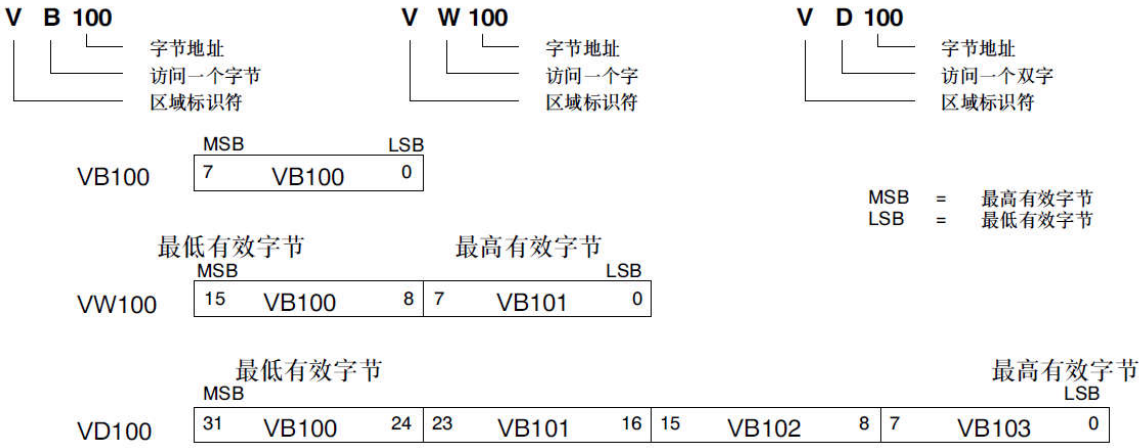


图 2-2 对同一地址进行字节，字和双字存取操作的比较。

其它存储区（如 T，C，HC 和累加器）中存取数据使用的地址格式包括区域标识符和设备号。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

2.1.1 存储器的范围及特性

表 2-2 无线 PLC 存储器范围及特性

存取方式		范围
位存取（字节.位）	I	0.0 -- 15.7
	Q	0.0 -- 15.7
	V	0.0 -- 2047.7
	M	0.0 -- 31.7
	SM	0.0 -- 165.7
	S	0.0 -- 31.7
	T	0 -- 255
	C	0 -- 255
	L	0.0 -- 63.7
字节存取	IB	0 -- 15
	QB	0 -- 15
	VB	0 -- 2047
	MB	0 -- 31
	SMB	0 -- 165
	SB	0 -- 31
	LB	0 -- 63
	AC	0 -- 3
	KB（常数）	KB（常数）
字存取	IW	0 -- 14
	QW	0 -- 14
	VW	0 -- 2046
	MW	0 -- 30
	SMW	0 -- 164
	SW	0 -- 30
	T	0 -- 255
	C	0 -- 255
	LW	0 -- 62
	AC	0 -- 3
	AIW	0 -- 30
	AQW	0 -- 30
	A KB（常数）	KB（常数）
双字存取	ID	0 -- 12
	QD	0 -- 12
	VD	0 -- 2044
	MD	0 -- 28
	SMD	0 -- 162
	SD	0 -- 28
	LD	0 -- 60
	AC	0 -- 3
	HC	0 -- 5
	SAID	0 -- 60
	SAQD	0 -- 60
	KD（常数）	KD（常数）

存储器	范围
用户程序大小 带运行模式下编辑 不带运行模式下编辑	8192 字节 8192 字节
用户数据大小	8192 字节
输入映像寄存器	I0.0 -- I15.7
输出映像寄存器	Q0.0 -- Q15.7
模拟量输入（只读）	AIW0 -- AIW30
特殊模拟量输入（只读）	SAIW0 –SAIW60
模拟量输出（只写）	AQW0 -- AQW30
特殊模拟量输出（只写）	SAQW0 – SAQW60
变量存储器（V）	VB0 -- VB2047
局部存储器（L）1	LB0 -- LB63
位存储器（M）	M0.0 -- M31.7
特殊存储器（SM） 只读	SM0.0 -- SM179.7 SM0.0 -- SM29.7
定时器 有记忆接通延迟 1ms 10ms 100ms 接通/关断延迟 1ms 10ms 100m	256（T0 -- T255） T0,T64 T1 -- T4, T65 -- T68 T5 -- T31, T69 -- T95 T32,T96 T33 -- T36, T97 -- T100 T37 -- T63, T101 -- T255
计数器	C0 -- C255
高速计数器	HC0 – HC7
顺序控制继电器（S）	S0.0 -- S31.7
累加寄存器	AC0 -- AC3
跳转/标号	0 -- 255
调用/子程序	0 -- 63
中断程序	0 -- 127
正 /负跳变	256
PID 回路	-

端口	端口 0

2.1.2 存储区数据的存取

- **输入过程映象寄存器：I**

在每次扫描周期的开始，CPU 对物理输入点进行采样，并将采样值写入输入过程映象寄存器中。可以按位、字节、字或双字来存取输入过程映象寄存器中的数据：

位:	I[字节地址].[位地址]	I0.1
字节、字或双字:	I[长度][起始字节地址]	IB4

- **输出过程映象寄存器：Q**

在每次扫描周期的结尾，CPU 将输出过程映象寄存器中的数值复制到物理输出点上。可以按位、字节、字或双字来存取输出过程映象寄存器：

位:	Q[字节地址].[位地址]	Q1.1
字节、字或双字:	Q[长度][起始字节地址]	QB5

- **变量存储区：V**

您可以用 V 存储器存储程序执行过程中控制逻辑操作的中间结果， 也可以用它来保存与工序或任务相关的其它数据。并且可以按位、字节、字或双字来存取 V 存储区中的数据：

位:	V[字节地址].[位地址]	V10.2
字节、字或双字:	V[长度][起始字节地址]	VW100

- **位存储区：M**

可以用位存储区作为控制继电器来存储中间操作状态和控制信息。并且可以按位、字节、字或双字来存取位存储区：

位:	M[字节地址].[位地址]	M26.7
字节、字或双字:	M[长度][起始字节地址]	MD20

- **定时器存储区：T**

无线 PLC 中，定时器可用于时间累计，其分辨率（时基增量）分为 1ms、10ms 和 100ms 三种。

定时器有两个变量：

- 当前值：16 位有符号整数，存储定时器所累计的时间。
- 定时器位：按照当前值和预置值的比较结果置位或者复位。预置值是定时器指令的一部分。

可以用定时器地址（T＋定时器号）来存取这两种形式的定时器数据。究竟使用哪种形式取决于所使用的指令： 如果使用位操作指令则是存取定时器位；如果使用字操作指令，则是存取定时

器当前值。如下图所示，常开触点指令是存取定时器位；而字移动指令则是存取定时器的当前值。

格式： T[定时器号] T24

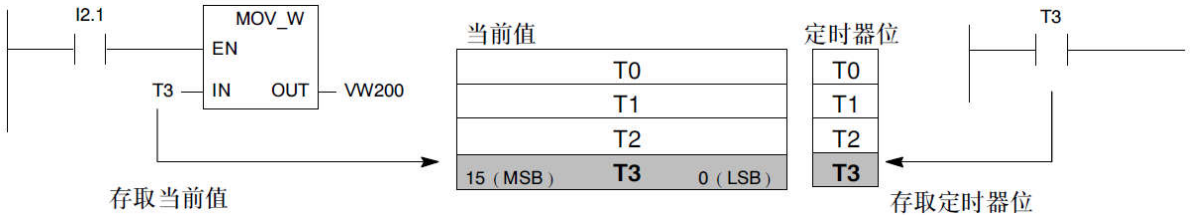


图 2-3 存取定时器位或者定时器的当前值

➢ 计数器存储区：C

在无线 PLC 中，计数器可以用于累计其输入端脉冲电平由低到高的次数。无线 PLC 提供了三种类型的计数器：一种只能增计数；一种只能减计数；另外一种既可以增计数，又可以减计数。计数器有两种形式：

- 当前值：16 位有符号整数，存储累计值。
- 计数器位：按照当前值和预置值的比较结果置位或者复位。预置值是计数器指令的一部分。

可以用计数器地址（C＋计数器号）来存取这两种形式的计数器数据。究竟使用哪种形式取决于所使用的指令：如果使用位操作指令则是存取计数器位；如果使用字操作指令，则是存取计数器当前值。如下图所示，常开触点指令是存取计数器位；而字移动指令则是存取计数器的当前值。

格式： C[计数器号]C24

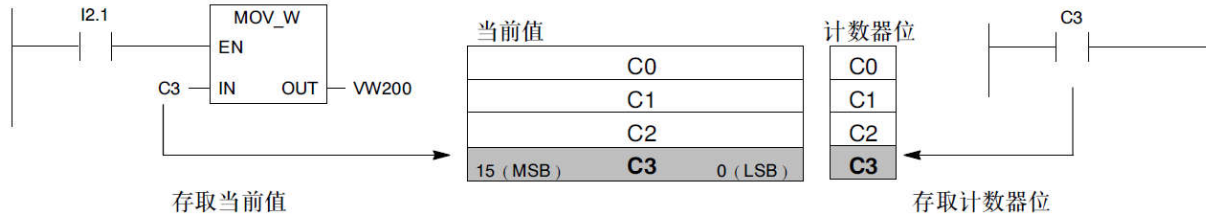


图2-4存取计数器位或者计数器的当前值

➢ *高速计数器：HC

高速计数器对高速事件计数，它独立于 CPU 的扫描周期。高速计数器有一个 32 位的有符号整数计数值（或当前值）。若要存取高速计数器中的值，则应给出高速计数器的地址，即存储器类型（HC）加上计数器号（如 HC0）。高速计数器的当前值是只读数据，仅可以作为双字（32 位）来寻址。

格式： HC[高速计数器号] HC1

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

*注意：高速计数器记录的是硬件输入通道的脉冲计数值，HC0 代表的是通道 0 的脉冲计数，HC7 代表的是通过 7 的脉冲计数。脉冲计数器是无线 PLC 的基本功能之一，无论用户是否使用该功能，高速计数器都进行工作。

➢ 累加器：AC

累加器是可以象存储器一样使用的读写设备。例如，可以用它来向子程序传递参数，也可以从子程序返回参数，以及用来存储计算的中间结果。无线 PLC 提供 4 个 32 位累加器（AC0，AC1，AC2 和 AC3）。并且您可以按字节、字或双字的形式来存取累加器中的数值。

被访问的数据长度取决于存取累加器时所使用的指令。如下图所示，当以字节或者字的形式存取累加器时，使用的是数值的低 8 位或低 16 位。当以双字的形式存取累加器时，使用全部 32 位。

格式： AC[累加器号] AC0

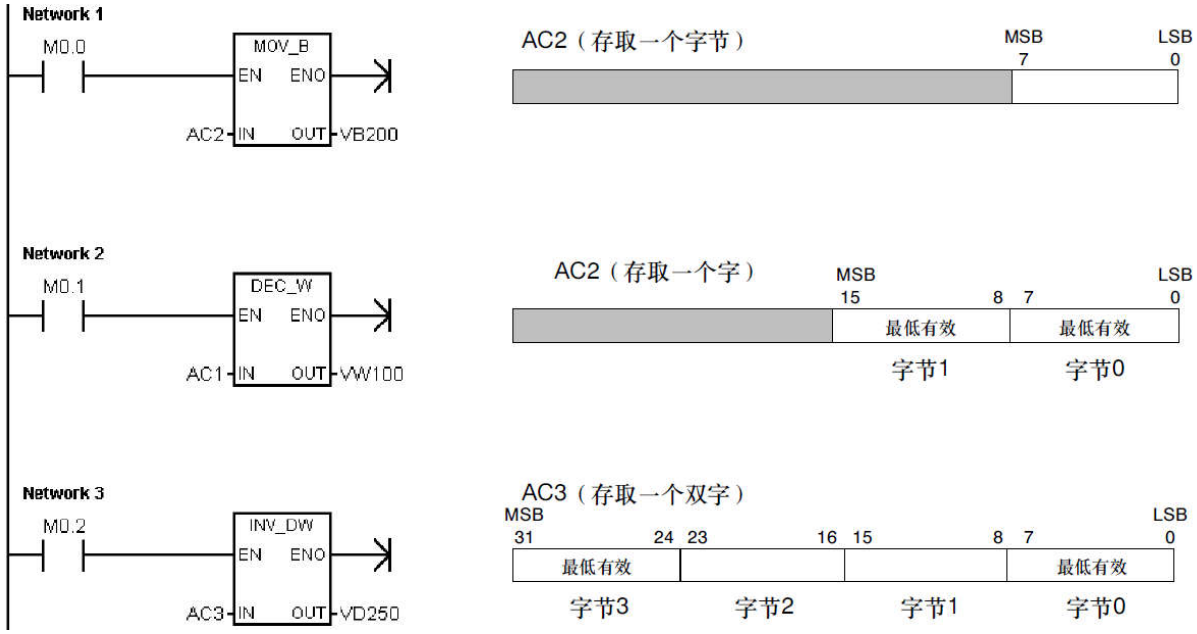


图 2-5 存取累加器

➢ 特殊存储器：SM

SM 位为 CPU 与用户程序之间传递信息提供了一种手段。可以用这些位选择和控制无线 PLC CPU 的一些特殊功能。例如：首次扫描标志位、按照固定频率开关的标志位或者显示数学运算或操作指令状态的标志位。（有关 SM 位的详细信息参见附录）。并且可以按位、字节、字或双字来存取 SM 位：

位:	SM[*字节地址*].[*位地址*]	SM0.1
字节、字或者双字:	SM[*长度*][*起始字节地址*]	SMB86

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

➢ 局部存储器：L

无线 PLC 有 64 个字节的局部存储器，其中 60 个可以用作临时存储器或者给子程序传递参数。(最后的 4 个字节的局部变量在梯形图编程中保留)。

局部存储器和变量存储器很相似，但有一处区别：变量存储器是全局有效的，而局部存储器只在局部有效。全局是指同一个存储器可以被任何程序存取（包括主程序、子程序和中断服务程序）。局部是指存储器区和特定的程序相关联。无线 PLC 给主程序分配 64 个局部存储器；给每一级子程序嵌套分配 64 个字节局部存储器；同样给中断服务程序分配 64 个字节局部存储器。

子程序或者中断服务程序不能访问分配给主程序的局部存储器。子程序不能访问分配给主程序、中断服务程序或者其它子程序的局部存储器。同样的，中断服务程序也不能访问分配给主程序或子程序的局部存储器。

无线 PLC 根据需要分配局部存储器。也就是说，当主程序执行时，分配给子程序或中断服务程序的局部存储器是不存在的。当发生中断或者调用一个子程序时，需要分配局部存储器。新的局部存储器地址可能会覆盖另一个子程序或中断服务程序的局部存储器地址。

局部存储器在分配时 PLC 不进行初始化，初值可能是任意的。当在子程序调用中传递参数时，在被调用子程序的局部存储器中，由 CPU 替换其被传递的参数的值。局部存储器在参数传递过程中不传递值，在分配时不被初始化，可能包含任意数值。

位:	L[*字节地址*].[*位地址*]	L0.0
字节、字或双字:	L[*长度*] [*起始字节地址*]	LB33

➢ 模拟量输入：AI

无线 PLC 将模拟量值（如温度或电压）转换成 1 个字长（16 位）的数字量。可以用区域标识符（AI）、数据长度（W）及字节的起始地址来存取这些值。因为模拟输入量为 1 个字长，且从偶数位字节（如 0，2，4）开始，所以必须用偶数字节地址（如 AIW0，AIW2，AIW4）来存取这些值。模拟量输入值为只读数据。

格式： AIW[起始字节地址] AIW4

➢ ^特殊模拟量输入：FAI

无线 PLC 将模拟量值（如温度、电压或电流）转换成 4 个字长（32 位）浮点型的数字量（实数值）。可以用区域标识符（FAI）及通道的起始地址来存取这些值。通道起始地址 0 就是指模拟量输入通道 0（硬件的 IN0 输入通道）。特殊模拟量输入值为只读数据。

格式： FAI[起始字节地址] FAI3

^**注意**：特殊模拟量输入（FAI）与普通模拟量输入（AI）都是指向同一个输入通道，他们之间的值是可以相互转换的，只是模拟输入量（AI）采用 0~4096 范围内的双字节整形数据来表示 0~20mA 电压、0~5V 或 10V 电压和温度，而特殊模拟输入量（FAI）采用实数四字节来直接表达电流电压或温度。例如模拟输入通道 0 的电流为 10.0mA 时，模拟量输入 AI0 的值为 2048；特殊模拟量输入 FAI0 的值为 10.0。

➢ 模拟量输出：AQ

无线 PLC 把 1 个字长（16 位）数字值按比例转换为电流或电压。可以用区域标识符（AQ）、数据长度（W）及字节的起始地址来改变这些值。因为模拟量为一个字长，且从偶数字节（如 0，2，4）开始，所以必须用偶数字节地址（如 AQW0，AQW2，AQW4）来改变这些值。模拟量输出值是只写数据。

格式： AQW[起始字节地址] AQW4

➢ ^特殊模拟量输出：FAQ

无线 PLC 把两个字长（32 位）数字值（实数）按比例转换为电流或电压。可以用区域标识符（FAQ）及通道的起始地址来改变这些值。特殊模拟量输出值是只写数据。

格式： FAQ[起始字节地址] FAQ3

^**注意**：特殊模拟量输出（FAQ）与普通模拟量输出（AQ）都是指向同一个输出通道，他们之间的值是可以相互转换的，只是模拟输出量（AQ）采用 0~4096 范围内的双字节整形数据来表示 0~20mA 电压、0~5V 或 10V 电压和温度，而特殊模拟输出量（FAQ）采用实数四字节来直接表达电流电压或温度。例如模拟输出通道 0 的电流为 10.0mA 时，模拟量输入 AQ0 的值为 2048；特殊模拟量输入 FAQ0 的值为 10.0。

➢ 顺控继电器存储器：S

顺控继电器位（S）用于组织机器操作或者进入等效程序段的步骤。SCR 提供控制程序的逻辑分段。可以按位、字节、字或双字来存取 S 位。

位： S[字节地址].[位地址] S3.1

字节、字或者双字： S[长度][起始字节地址] SB4

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

2.1.3 实数的格式

实数（浮点数）由 32 位单精度数表示，其格式按照 ANSI/IEEE 754--1985 标准中所描述的形式，参见下图。实数按照双字长度来存取。

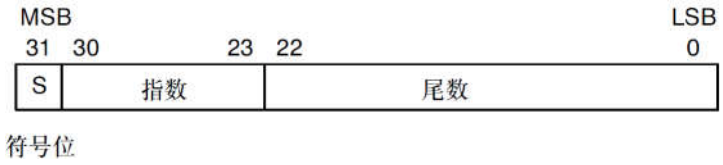


图 2-6 实数的格式

对于无线 PLC 来说，浮点数精确到小数点后第六位。因而当您使用一个浮点数常数时，您最多可以指定到小数点后第六位。

在计算中涉及到非常大和非常小的数，则有可能导致计算结果不精确。例如数值相差 10 的 x 次方倍，而 x＞6 时。

例如： 100 000 000+1=100 000 000

2.1.4 字符串的格式

字符串指的是一系列字符，每个字符以字节的形式存储。字符串的第一个字节定义了字符串的长度，也就是字符的个数。下图给出了一个字符串的格式。一个字符串的长度可以是 0 到 254 个字符，再加上长度字节，一个字符串的最大长度为 255 个字节。而一个字符串常量的最大长度为 126 字节。

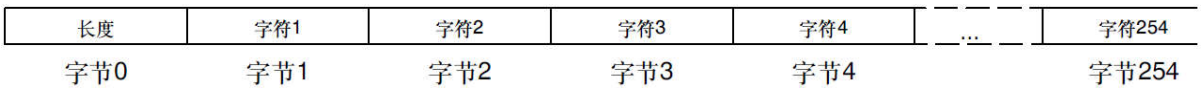


图 2-7 字符串的格式

2.1.5 在指令中输入常数值

在无线 PLC 的许多指令中，都可以使用常数值。常数可以是字节、字或者双字。无线 PLC 以二进制数的形式存储常数，可以分别表示十进制数、十六进制数、ASCII 码或者实数（浮点数）。无线 PLC 支持不同格式的字符串常量，采用“编码区别符#内容”。例如”GB#我是无线 PLC” 、”UTF#你好”和”UNICODE#捷麦”。中文字符常量前没有编码区别符（GB#\UTF#\UTF#）时，默认为 UNICODE 的中文编码格式，例如用户可以直接输入 ”捷麦” 的中文字符，它与”UNICODE#捷麦”一样。

见下表。

表 2-3 常数表示法

数制	格式	举例
十进制	[十进制值]	20047
十六进制	16 # [十六进制值]	16 # 4E4F
二进制	2 # [二进制数]	2#1010_0101_1010_0101
ASCII码	'[ASCII码文本]'	'ABCD'
实数	ANSI/IEEE 754-1985	+1.175495E-38（正数）-1.175495E-38（负数）
字符串	"[字符串文本]"	"ABCDE"

提示

无线 PLCCPU 不支持数据类型检测（例如指定常数存储为一个整数、有符号整数或者双整数）。例如：可以在加法指令中使用 VW100 中的值作为有符号整数，同时也可以在异或指令中将 VW100 中的数据当作无符号的二进制数。

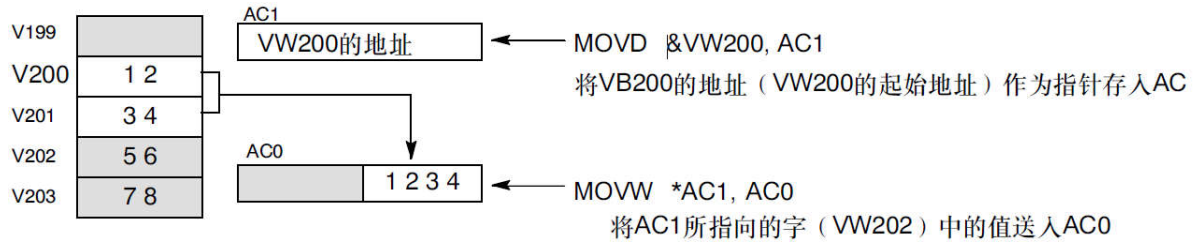
2.1.6 用指针对存储区间接寻址

间接寻址是指用指针来访问存储区数据。指针以双字的形式存储其它存储区的地址。只能用 V 存储器、L 存储器或者累加器寄存器（AC1、AC2、AC3）作为指针。要建立一个指针，必须以双字的形式，将需要间接寻址的存储器地址移动到指针中。指针也可以作为参数传递到子程序中。

无线 PLC 允许指针访问以下存储区： I、Q、V、M、S、AI、AQ 和 C（仅限于当前值）。您无法用间接寻址的方式访问单独的位，也不能访问 HC 或者 L 存储区。

要使用间接寻址，您应该用"&"符号加上要访问的存储区地址来建立一个指针。指令的输入操作数应该以"&"符号开头来表明是存储区的地址，而不是其内容将移动到指令的输出操作数（指针）中。

当指令中的操作数是指针时，应该在操作数前面加上"*"号。如下图所示，输入*AC1 指定 AC1 是一个指针，MOVW 指令决定了指针指向的是一个字长的数据。在本例中，存储在 VB200 和 VB201 中的数值被移动到累加器 AC0 中。



北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

图 2-8 创建和使用指针

下图所示，您可以改变一个指针的数值。由于指针是一个 32 位的数据，要用双字指令来改变指针的数值。简单的数学运算，如加法指令或者增加指令，可用于改变指针的数值。

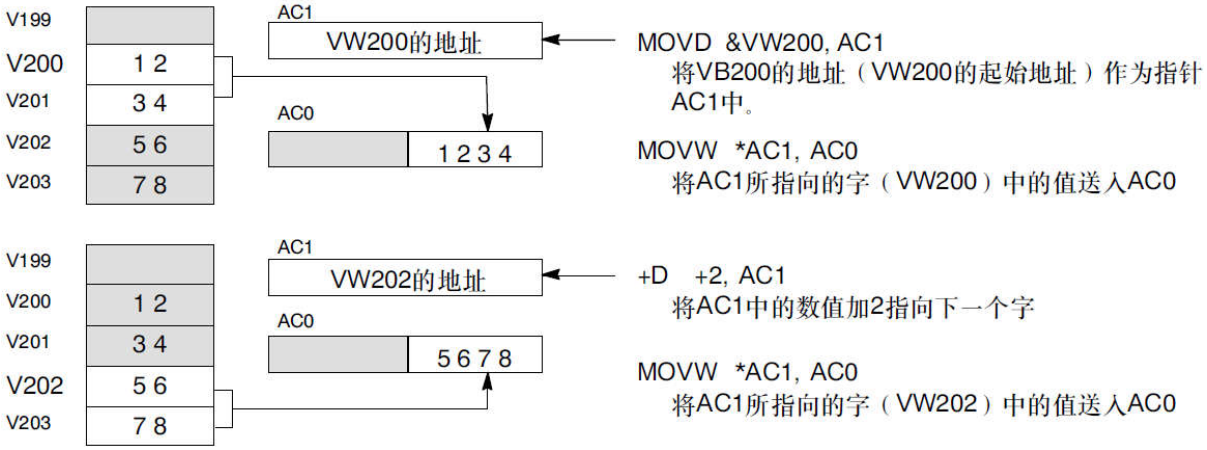


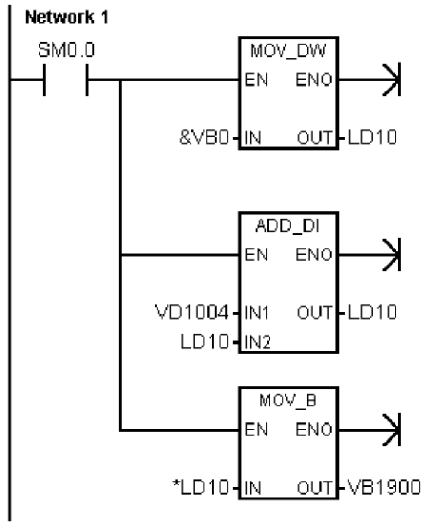
图 2-9 改变指针

提示

按照所访问的数据长度使用不同的指令：当访问字节时，使用增加指令使指针值加 1；当访问字或者计数器、定时器的当前值时，用加法或者增加指令使指针值加 2；当访问双字时，使用加法或者增加指令使指针值加 4。

用地址偏移量来访问 V 存储区数据的例子程序

本例中用 LD10 作为 VB0 的地址指针。然后可以利用 VD1004 中存储的地址偏移量来改变指针值。经过改变后，LD10 指向 V 区中的另外一个地址（VB0＋偏移量）。然后将 LD10 指向的 V 区地址中存储的数值复制到 VB1900 中。通过改变 VD1004 中的数值，您可以访问 V 存储器中的任意单元。

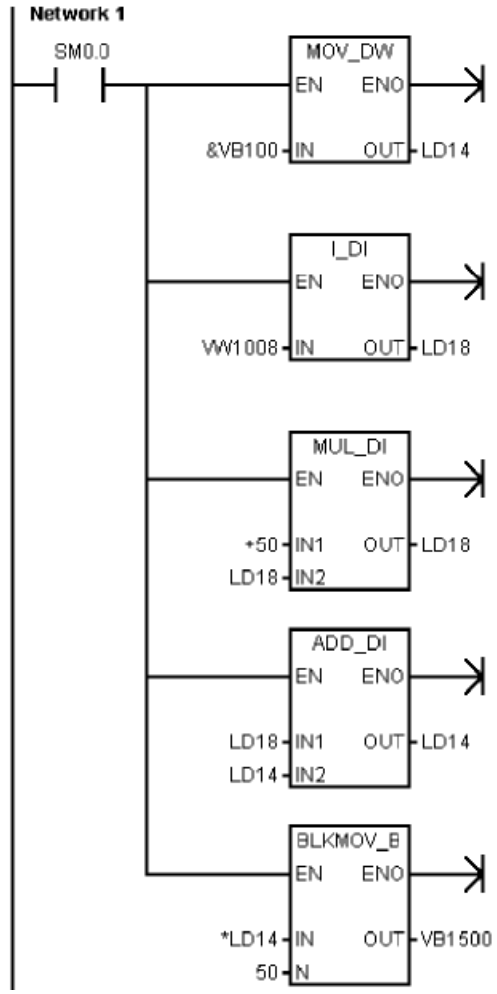



```
Network1
          //如何利用偏移量访问
          //V 存储器中的任意单元：

          //1.将 V 区的起始地址装载到指针中。
          //2.将偏移量加到指针上。
          //3.将 V 存储器中的数值复制到 VB1900 中。
          //
LD     SM0.0
MOVD   &VB0, LD10
+D          VD1004, LD10
MOVB   *LD10, VB1900
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

用指针访问数据表的例子程序
本例中用 LD14 作为指向一个配方表的指针，配方表的起始地址为 VB100。在本例中 VW1008 用来存储一个指定的配方在表中的索引号。如果每条配方的长度为 50 个字节，则用这个索引号乘以 50 就可以得到这条配方起始地址的偏移量。用指针加上偏移量，您就可以访问表中的每一条配方。在本例中，配方被复制到从 VB1500 开始的 50 个字节中。



Network1

//如何从配方表中传递一条配方：
//每条配方的长度为 50 个字节。
//索引值（VW1008）指定装载哪一条配方。

//1. 建立一个指针指向配方表的起始地址。
//2. 将配方的索引值转换为双字。
//3. 乘以每条配方的长度算出偏移量。
//4. 将偏移量加到指针上。
//5. 将配方传递至 VB1500 到 VB1549 中。

```
LD      SM0.0
MOVD    &VB100, LD14
ITD     VW1008, LD18
*D      +50, LD18
+D      LD18, LD14
BMB     *LD14, VB1500, 50
```

2.2 理解保存和存储数据

无线 PLC 提供了多种安全措施来确保用户程序、程序数据和组态数据不丢失。

- 保持数据存储器-- 由用户选定的数据存储区，在一次上电周期中，只要超级电容和可选电池卡不放电，该存储器的数据就不会改变。在所有存储区中，只有 V、M、定时器和计数器存储区能被组态为保持存储区。无线 PLC 的 V1.0 版本不支持该功能。
- 永久存储器-- 不可变存储器，用来储存程序块、数据块、系统块、强制值、组态为掉电保存的 M 存储器和在用户程序的控制下写入的指定值。

- 存储卡-- 可拆卸的不可变存储器，用来储存程序块、数据块、系统块、配方、数据归档和强制值。无线 PLC 的 V1.0 版本不支持该功能。

通过无线 PLC 资源管理器，您可以将文档文件（doc、text、pdf 等）储存在存储卡内，也可以将普通文件保留在存储卡中（复制、删除、创建目录和放置文件）。无线 PLC 的 V1.0 版本不支持该功能。

要安装存储卡，应先从无线 PLCCPU 上取下塑料盖，然后将存储卡插入槽中。

无线 PLC 的 V1.0 版本不支持资源管理器的功能。

2.2.1 下载和上传用户程序

用户程序包括以下几个部分：

- 程序块
- 数据块（可选）
- 系统块（可选）
- 配方（可选）
- 数据归档组态（可选）

当您下载程序时，出于安全考虑，程序块、数据块和系统块将储存在永久存储器中。而配方和数据归档组态将储存在存储卡中，并更新原有的配方和数据归档。那些不涉及下载操作的程序部分也将保留在永久存储器和存储卡中，保持不变。

如果程序下载涉及到配方或数据归档组态，则存储卡就必须一直装在无线 PLC 上，否则程序可能无法正确运行。

将用户程序下载至无线 PLC 中：

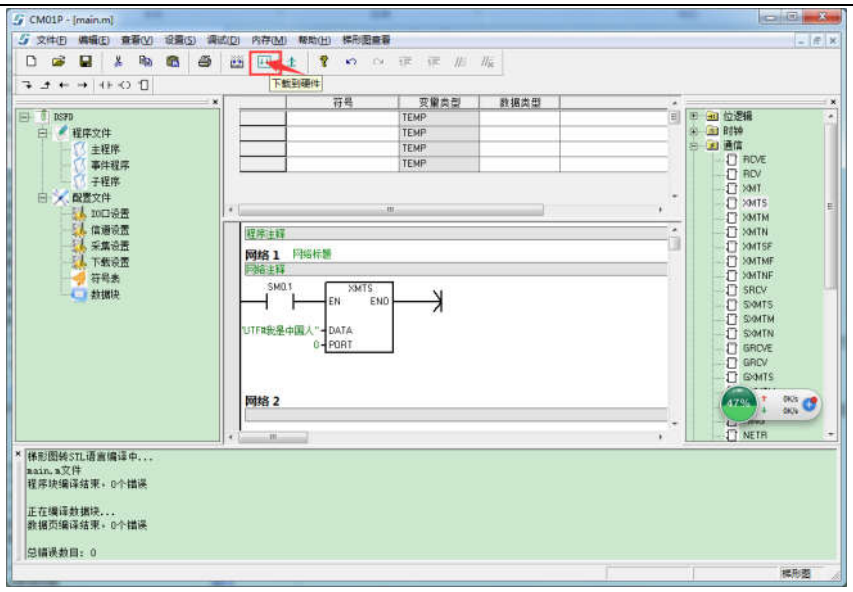


图 2-10 下载程序到无线 PLC 操作示意图

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

2.2.2 开机后数据的恢复

上电之后，无线 PLC 将从永久存储器中恢复程序块和系统块。然后无线 PLC 将检查可选电池卡（如果有的话）是否正确保存了 RAM 存储器中的数据。如果数据保存正确，则用户数据存储器的保持区将保持不变。而 V 存储器的非保持部分将根据永久存储器中的数据块内容来恢复。其它存储区的非保持部分则被清空。

如果 RAM 中的内容已经丢失（比如较长时间的掉电），则无线 PLC 将清除所有用户数据区，将保持数据丢失存储器位（SM0.2）置位，并读取永久存储器的数据块内容来恢复 V 存储器，如果 M 存储器的前 14 位已预设为保持，则无线 PLC 还将读取永久存储器恢复这些位的内容。

由于无线 PLC 的 V1.0 版本没有涉及可选电池卡和超级电容等储能装备，因此掉电 RAM 内容丢失的时间非常快，基本上完全断电后，RAM 的内容全部丢失，清除所有用户数据区，丢失存储器位（SM0.2）置位，从永久存储器中恢复 V 存储器和预设 M 存储器的内容。

2.2.3 通过编程方式将 V 存储器保存至永久存储器

可以将储存在 V 存储器中的数据（字节、字或双字）存储到永久存储器中。一般来说，一个写永久存储器的操作会使扫描周期增加 10 到 15 ms。通过保存操作所写入的数据会覆盖先前永久存储器中 V 存储区的数据。

保存至永久存储器的操作并不更新存储卡中的数据。具体操作详见存储指令。

提示

由于保存至永久存储器（EEPROM）的操作次数是有限的（最少 10 万次，典型值为 100 万次），所以请注意只在必要时才进行保存操作。否则，EEPROM 可能会失效，从而引起 CPU 故障。一般来说，当特定事件发生时，才执行存储操作，而特定事件是不很频繁发生的。

例如，如果无线 PLC 扫描周期为 50ms，一个数据在每个扫描周期保存一次，则 EEPROM 最短只能工作 5,000 秒，还不到一个半小时。另一方面，如果一个数据每小时保存一次，则 EEPROM 至少可以工作 11 年。

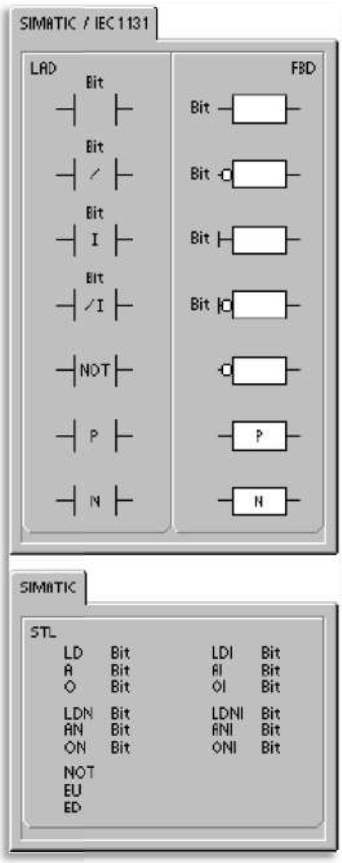
第3章 指令集

3.1 位逻辑指令

3.1.1 触点

➢ 标准触点

常开触点指令（LD、A 和 O）与常闭触点指令（LDN、AN 和 ON）从存储器或者过程映象寄存器中得到参考值。标准触点指令从存储器中得到参考值。(如果数据类型是 I 或 Q，则从过程映象寄存器中得到参考值。）当位值为 1 时，常开触点闭合；当位值为 0 时，常闭触点闭合。



在 STL 中，常开指令 LD、A 或 O 将相应地址位的位值存入栈顶；而常闭指令 LDN、AN 或 ON 则将相应地址位的位值取反，再存入栈顶。

➢ ~立即触点

立即触点并不依赖于无线 PLC 的扫描周期刷新，它会立即刷新。常开立即触点指令（LDI、AI 和 OI）和常闭立即触点指令（LDNI、ANI 和 ONI）在指令执行时得到物理输入值，但过程映像寄存器并不刷新。

当物理输入点状态为 1 时，常开立即触点闭合；当物理输入点状态为 0 时，常闭立即触点闭合。常开立即指令 LDI、AI 或 OI 将物理输入值存入栈顶，而常闭立即指令 LDNI、ANI 或 ONI 将物理输入的值取反，再存入栈顶。

注意：无线 PLC 无线 PLC 的 V1.0 版本不支持立即触点指令。

➢ 取反指令

取反指令（NOT）改变能流输入的状态（也就是说，它将栈顶值由 0 变为 1，由 1 变为 0）。

北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

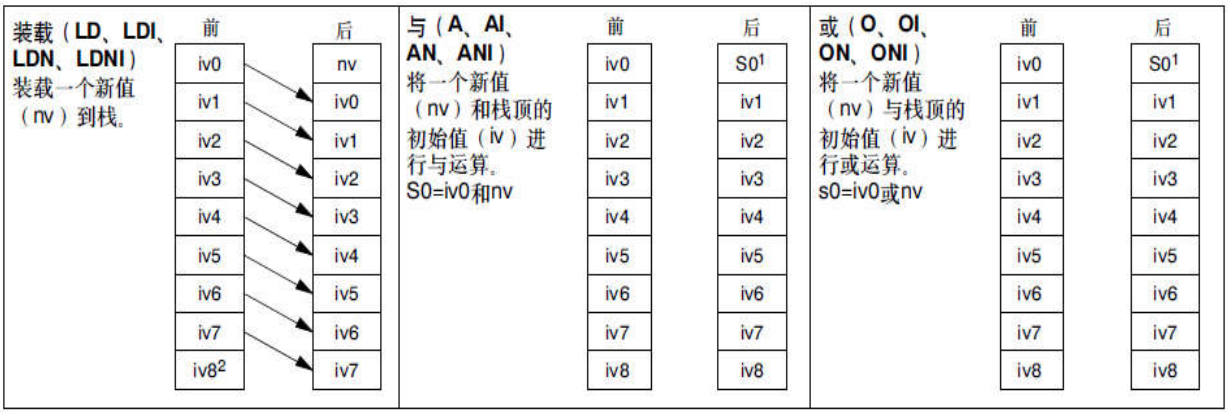
➢ 正、负跳变指令

正跳变触点指令（EU）检测到每一次正跳变（由 0 到 1），让能流接通一个扫描周期。负跳变触点指令（ED）检测到每一次负跳变（由 1 到 0），让能流接通一个扫描周期。对于正跳变指令，一旦发现有正跳变发生（由 0 到 1），该栈顶值被置为 1，否则置 0。对于负跳变指令，一旦发现有负跳变发生（由 1 到 0），该栈顶值被置为 1，否则置 0。

表 3-1 位逻辑输入指令的有效操作数

输入/输出	数据类型	操作数
位	BOOL	I、Q、V、M、SM、S、T、C、L、能流
位（立即）	BOOL	I

如下图所示，无线 PLC 用逻辑堆栈来决定控制逻辑。在本例中，"iv0"到"iv7"表示逻辑堆栈的初始值，"nv"表示指令提供的一个新值，S0 表示逻辑堆栈中存储的计算值。



1 S0 表示存储在逻辑栈中的计算值。

2 在装载指令执行之后，值 iv8 丢失

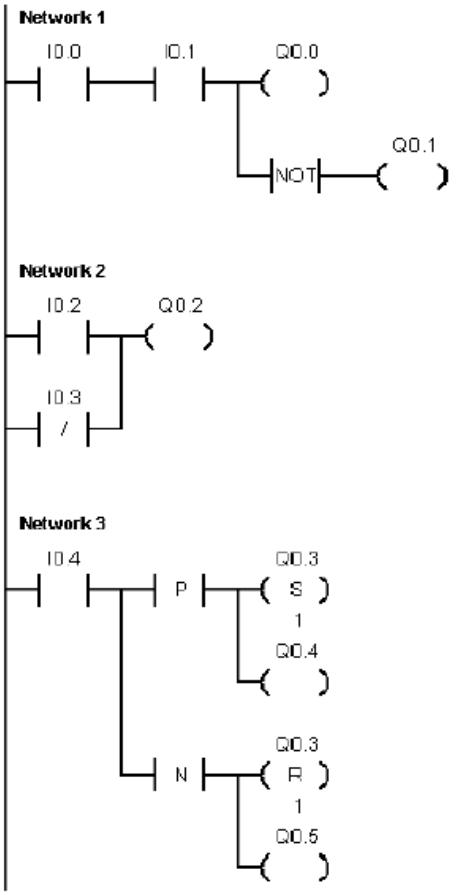
图 3-1 触点指令的操作。

提示

由于正跳变指令和负跳变指令要求由 1 到 0 或者由 0 到 1 的变化，您不能在第一个扫描周期中检测到上升沿或者下降沿的变化。在第一个扫描周期，无线 PLC 利用这些指令储存指定位的状态。在接下来的扫描周期中，这些指令能够检测到指定位的变化。

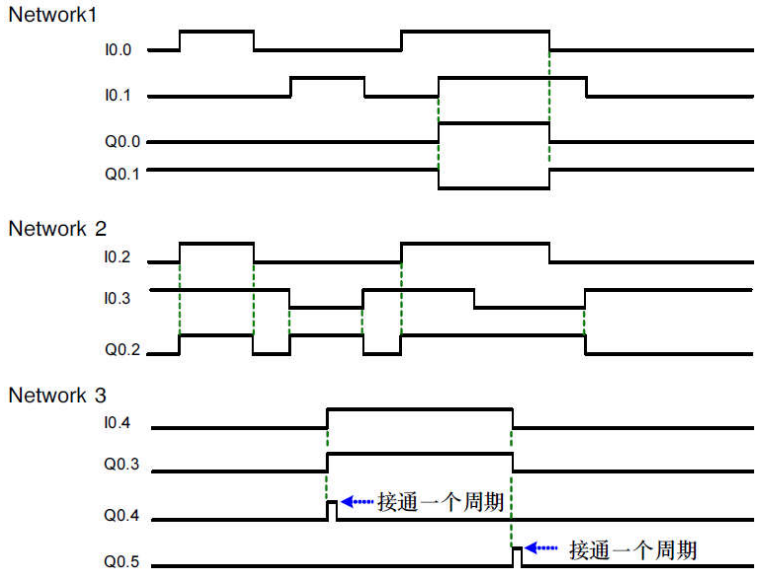
北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

示例：线圈指令




```
Network 1       //要想激活Q0.0，常开触点I0.0
                //和I0.1必须为接通（闭合）。NOT
                //指令作为一个反向器使用。在RUN
                //模式下，Q0.0和Q0.1具有相反的逻辑状态。
LD      I0.0
A       I0.1

=       Q0.0
NOT
=       Q0.1

Network 2       //要想激活Q0.2，常开触点I0.2必须为on或者常闭
                //触点I0.3必须为off。
                //要想激活输出，并行LAD分支
                //（或逻辑输入）中应该有一个或
                //多个逻辑值为真。
LD      I0.2
ON      I0.3
=       Q0.2

Network 3       //在P触点的一个上升沿或者在N触点的一个下降
                //沿出现时，一个扫描周期内输出一个脉冲。
                //在RUN模式，Q0.4和Q0.5的脉冲状态变化太快
                //以至于在程序中无法用状态图监视。
                //置位和复位指令将
                //Q0.3的状态变化锁存，
                //使程序可以监视。
LD      I0.4
LPS
EU
S       Q0.3，1
=       Q0.4
LPP
ED
R       Q0.3，1
=       Q0.5
```

时序图

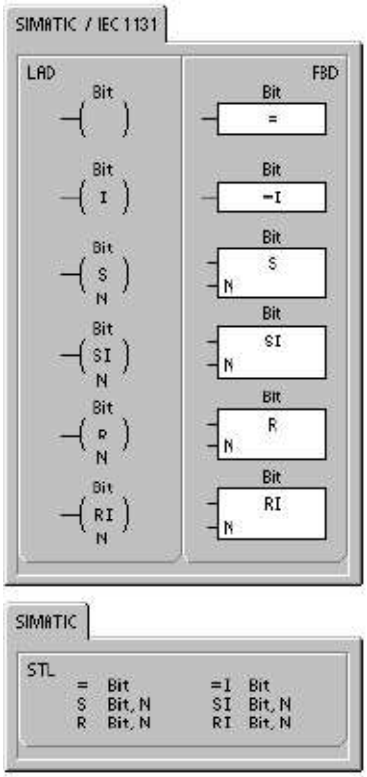


北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

3.1.2 线圈

➢ 输出

输出指令（=）将新值写入输出点的过程映象寄存器。当输出指令执行时，无线 PLC 将输出过程映象寄存器中的位接通或者断开。在 LAD 中，指定点的值等于能流。在 STL 中，栈顶的值复制到指定位。



➢ 立即输出

当指令执行时，立即输出指令（=I）将新值同时写到物理输出点和相应的过程映象寄存器中。当立即输出指令执行时，物理输出点立即被置为能流值。在 STL 中，立即指令将栈顶的值立即复制到物理输出点的指定位上。"I"表示立即，当指令执行时，新值会同时被写到物理输出和相应的过程映象寄存器。这一点不同于非立即指令，只把新值写入过程映象寄存器。

➢ 置位和复位

置位（S）和复位（R）指令将从指定地址开始的 N 个点置位或者复位。您可以一次置位或者复位 1--255 个点。

如果复位指令指定的是一个定时器位（T）或计数器位（C），指令不但复位定时器或计数器位，而且清除定时器或计数器的当前值。

使 ENO=0 的错误条件：

0006（间接寻址）

0091（操作数超出范围）

➢ 立即置位和立即复位

立即置位(SI)和立即复位(RI)指令将从指定地址开始的 N 个点立即置位或者立即复位。您可以一次置位或复位 1 到 128 个点。

"I"表示立即，当指令执行时，新值会同时被写到物理输出和相应的过程映象寄存器。这一点不同于非立即指令，只把新值写入过程映像寄存器。

使 ENO=0 的错误条件：

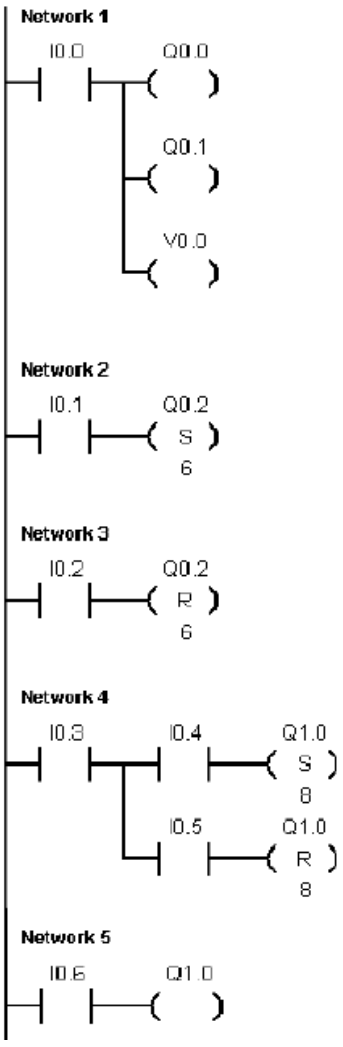
0006（间接寻址）

0091（操作数超出范围）

表 3-2 位逻辑输出指令的有效操作数

输入/输出	数据类型	操作数
位	BOOL	I、Q、V、M、SM、S、T、C、L
位（立即）	BOOL	Q
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数

示例：线圈指令

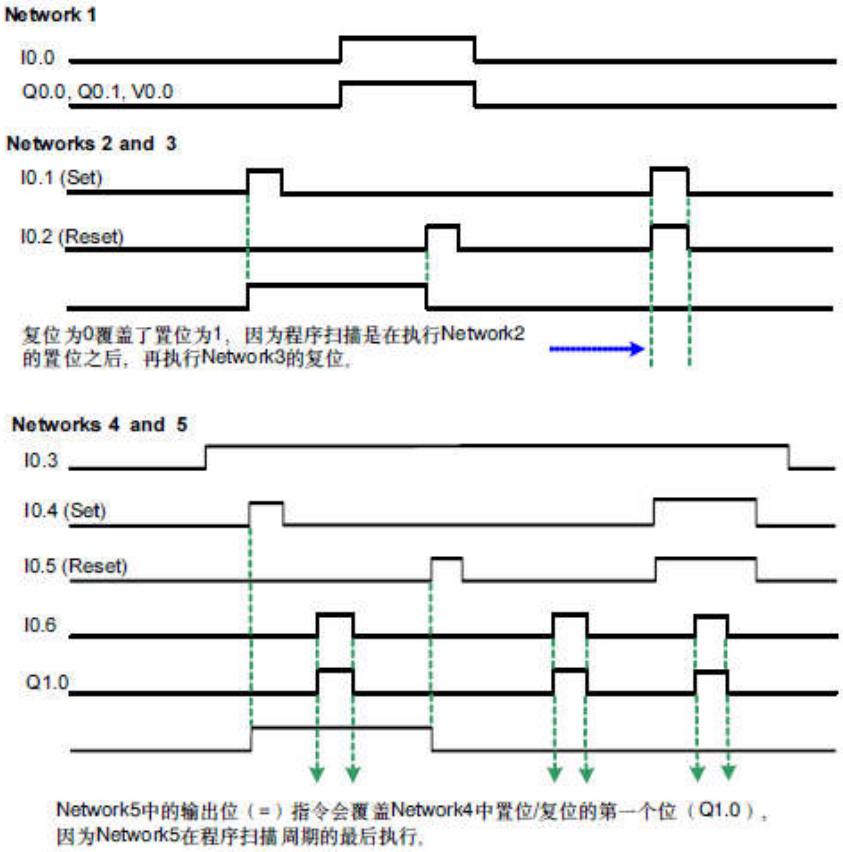



```
Network1 //输出指令为外部I/O（I、Q）和内部存储
         //器（M、SM、T、C、V、S、L）指定位值。
LD       I0.0
=        Q0.0
=        Q0.1
=        V0.0

Network2 //连续将一组6位置为1。
         //指定起始地址和置位的个数。当第一位
         //（Q0.2）的值为1时，置位指令
         //的程序状态指示器为ON。
LD       I0.1
S        Q0.2，6

Network3 //连续将一组6位置为0。
         //指定起始地址和复位的个数。
         //当第一位（Q0.2）的值为0时，复位指
         //令的程序状态指示器为ON。
LD       I0.2
R        Q0.2，6

Network4 //置位和复位一组8个输出位（Q1.0~Q1.7）。
LD       I0.3
LPS
A        I0.4
S        Q1.0，8
LPP
A        I0.5
R        Q1.0，8

Network5 //置位和复位指令实现锁存器功能。
         //完成置位/复位功能，必须确保这些
         //位没有在其它指令中被改写。在本例中，
         //Network4置位和复位一组
         //8个输出位（Q1.0~Q1.7）。在RUN模式
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

	//下Network5会覆盖Q1.0的值，从而
	//控制Network4中的程序状态显示器。
	LD I0.6
	= Q1.0

时序图

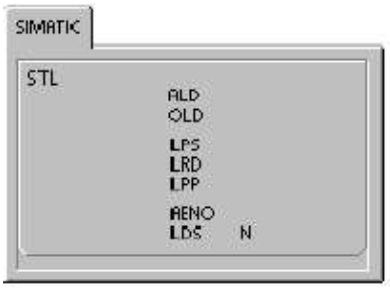


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.1.3 逻辑堆栈指令

➢ 栈装载与

栈装载与指令（ALD）对堆栈中第一层和第二层的值进行逻辑与操作，结果放入栈顶。执行完栈装载与指令之后，栈深度减 1。



➢ 栈装载或

栈装载或指令（OLD）对堆栈中第一层和第二层的值进行逻辑或操作，结果放入栈顶。执行完栈装载或指令之后，栈深度减 1。

➢ 逻辑推入栈

逻辑推入栈指令（LPS）复制栈顶的值，并将这个值推入栈。栈底的值被推出并消失。

➢ 逻辑读栈

逻辑读栈指令（LRD）复制堆栈中的第二个值到栈顶。堆栈没有推入栈或者弹出栈操作，但旧的栈顶值被新的复制值取代。

➢ 逻辑弹出栈

逻辑弹出栈指令（LPP）弹出栈顶的值，堆栈的第二个栈值成为新的栈顶值。

➢ ENO 与

ENO 与指令（AENO）对 ENO 位和栈顶的值进行逻辑与操作，其产生的效果与 LAD 或者 FBD 中盒指令的 ENO 位相同。与操作结果成为新的栈顶。

ENO 是 LAD 和 FBD 中盒指令的布尔输出。如果盒指令的 EN 输入有能流并且执行没有错误，则 ENO 将能流传递给下一元素。您可以把 ENO 作为指令成功完成的使能标志位。ENO 位被用作栈顶，影响能流和后续指令的执行。STL 中没有 EN 输入。条件指令要想执行，栈顶值必须为逻辑 1。在 STL 中也没有 ENO 输出。但是在 STL 中，那些与 LAD 和 FBD 中具有 ENO 输出的指令相应的指令，存在一个特殊的 ENO 位。它可以被 AENO 指令访问。

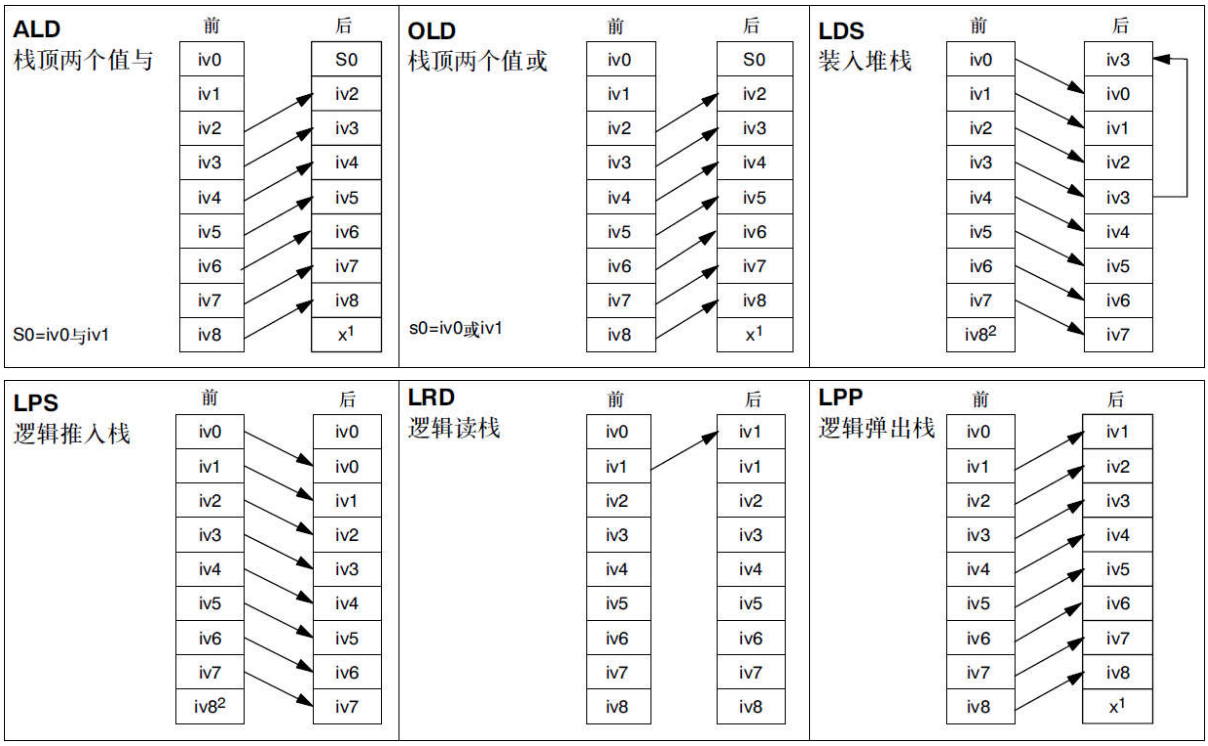
➢ 装入堆栈

装入堆栈指令（LDS）复制堆栈中的第 N 个值到栈顶。栈底的值被推出并消失。

表 3-3 装入堆栈指令的有效操作数

输入/输出	数据类型	操作数
N	BYTE	常数（0 到 8）

如下图所示，无线 PLC 用逻辑堆栈来决定控制逻辑。在本例中，“iv0”到“iv7”表示逻辑堆栈的初始值，“nv”表示指令提供的一个新值，而“S0”表示逻辑堆栈中存储的计算值。



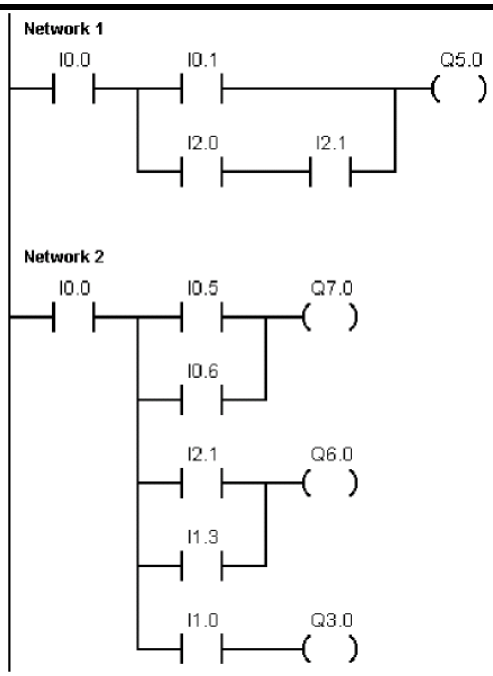
1 数值是不确定的（可以是 0，也可以是 1）

2 在逻辑入栈或者装入堆栈指令执行后，iv8 的值丢失。

图 3-2 逻辑堆栈指令的操作

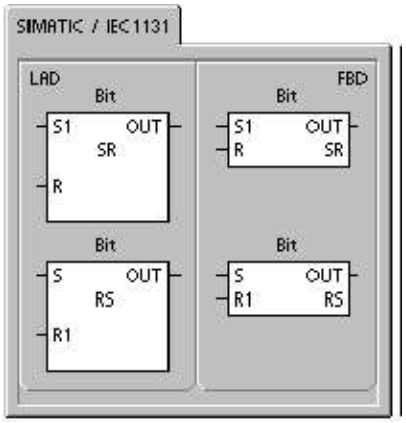
示例：逻辑堆栈指令

北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com




```
Network 1
LD      I0.0
LD      I0.1
LD      I2.0
A       I2.1
OLD
ALD
=       Q5.0

Network 2
LD      I0.0
LPS     I0.5
LD      I0.6
O
ALD
=       Q7.0
LRD
LD      I2.1
O       I1.3
ALD
=       Q6.0
LPP
A       I1.0
=       Q3.0
```

3.1.4 RS 触发器指令



置位优先触发器是一个置位优先的锁存器。当置位信号（S1）和复位信号（R）都为真时，输出为真。

复位优先触发器是一个复位优先的锁存器。当置位信号（S）和复位信号（R1）都为真时，输出为假。

Bit 参数用于指定被置位或者复位的布尔参数。可选的输出反映 Bit 参数的信号状态。

表 3-4RS 触发器指令的有效操作数

输入/输出	数据类型	操作数
S1、R	BOOL	I、Q、V、M、SM、S、T、C、能流
S、R1、OUT	BOOL	I、Q、V、M、SM、S、T、C、L、能流
Bit	BOOL	I、Q、V、M、S

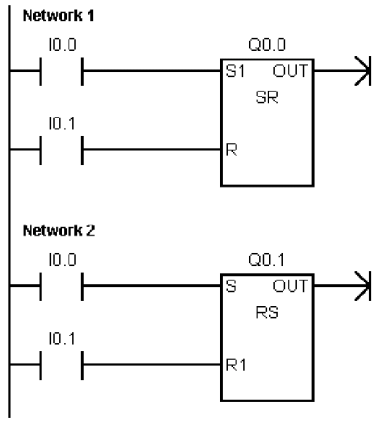
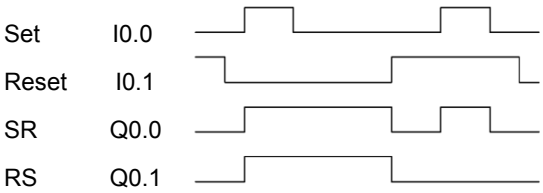
下表给出了例子程序的真值表。

表 3-5RS触发器指令真值表

指令	S1	R	输出（Bit）
置位优先触发器指令（SR）	0	0	保持前一状态

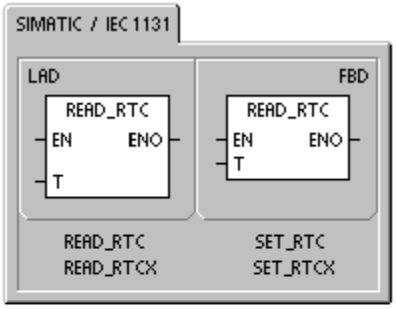
北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

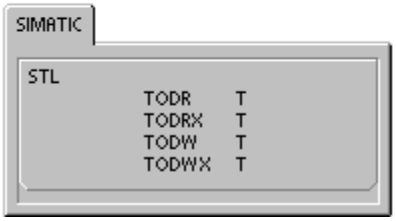
位优先触发器指令	0	1	0
	1	0	1
	1	1	1
指令	S	R1	输出（Bit）
复位优先触发器指令（RS） 复位优先触发器指令	0	0	保持前一状态
	0	1	0
	1	0	1
	1	1	0

示例：RS 触发器指令	
	时序图 

3.2!!!时钟指令

3.2.1 读实时时钟和写实时时钟





读实时时钟（TODR）指令从硬件时钟中读当前时间和日期，并把它装载到一个 6 字节，起始地址为 T 的时间缓冲区中。

写实时时钟（TODW）指令将当前时间和日期写入硬件时钟，当前时钟存储在以地址 T 开始的 6 字节时间缓冲区中。所有的日期和时间值采用标准值表达（例如：用 14 表示 2014 年，06 表示 6 月份）。下图给出了时间缓冲区（T）的格式。时间日期（TOD）时钟在电源掉电或内存丢失后，保持最后断电时的时间值。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0007（TOD 数据错误），只对写实时时钟指令有效。
- 000C（时钟模块不存在）

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

表 3-6 时钟指令的有效操作数

输入/输出	数据类型	操作数
T	BYTE	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC

T	T+1	T+2	T+3	T+4	T+5
年 00~99	月 01~12	日 01~31	小时 00~23	分钟 00~59	秒 00~59

图 3-3 6 字节时间缓冲区的格式

提示

无线 PLCCPU 会自动识别星期。设置时间时不用设置星期信息，可以通过指令获得某时间时星期几，

无效日期 February 30（2 月 30 日）可能被接受。故必须确保输入的数据是正确的。

PLC 只使用年信息的后两位，不会受到世纪跨越的影响。但是，用到年份进行计算或比较的用户程序必须考虑两位的表示方法和世纪的变化。在 2099 年之前闰年可正确处理。

3.2.2 读网络时钟

读实时时钟（TODRN）指令从网络中读当前北京时间和日期，并把它装载到一个 6 字节，起始地址为 T 的时间缓冲区中。用法与读实时时钟类似，只是网络时钟的读取不是立刻获得是，需要等待网络响应，等待的时间依据网络通信环境不同而不同。

3.2.3 时间校准

时间校准(TADJN)指令是从网络中读取当前标准北京时间和日期，然后自动校准当前的硬件实时时钟。默认情况下，如果使用了 GPRS 服务，那么注册 GPRS 服务时系统会自动校准时间，如果不使用自动校准功能，请在 CM01P 软件中关闭该功能。

3.2.4 ~扩展读实时时钟

扩展读实时时钟（TODRX）指令从 PLC 中读取当前时间、日期和夏令时组态，并装载到从由 T 指定的地址开始的 19 字节缓冲区内。

说明

无线 PLC 的 V1.0 版本的硬件不支持扩展读实时时钟的功能。

3.2.5 ~扩展写实时时钟

扩展读实时时钟（TODWX）指令写当前时间、日期和夏令时组态到 PLC 中由 T 指定的地址开

始的 19 字节缓冲区内。您必须按照 BCD 码的格式编码所有的日期和时间值（例如：用 16＃02 表示 2002 年）。表 6-9 给出了 19 字节时间缓冲区（T）的格式。

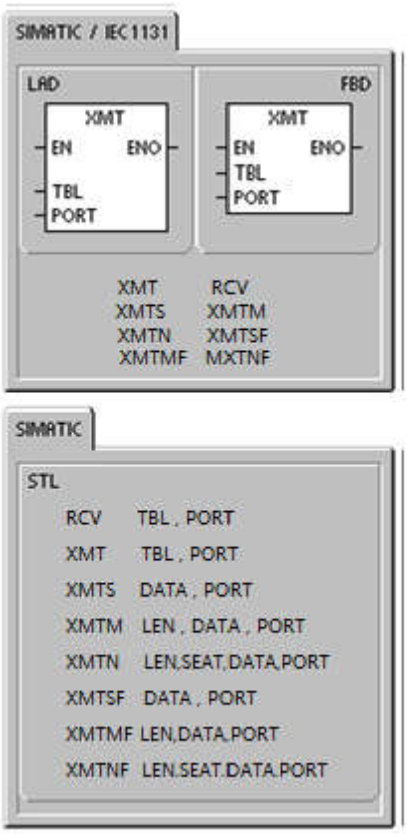
说明

无线 PLC 的 V1.0 版本的硬件不支持扩展写实时时钟的功能。

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.3 串口通信指令

无线 PLC 具有串口 S 通信功能。



3.3.1 *串口收发

发送指令（XMT）用于依靠串口发送一包数据。接收指令（RCV 或者 RCVE）从串口接收到的一包数据，将包信息被存储在数据缓冲区（TBL）中。数据缓冲区的前两个数据指明了（发送）接收到的字节数。

- 使 ENO=0 的错误条件：
- 0006（间接寻址）
- 0009（在 Port0 同时发送和接收）
- 000B（在 Port1 同时发送和接收）

表 3-7 发送和接收指令的有效操作数

输入/输出	数据类	操作数
TBL	BYTE	IB、QB、VB、MB、SMB、SB、*VD、*LD、*AC
PORT	BYTE	常数 0~1
DATA	BYTE	VB, 常数字符串, LB, *VD, *LD, *AC
LEN	INT	IW,QW,VW,MW,SMW,SW,LW,T,C,AC,AIW,*VD,*LD,*AC 及常数
SEAT	BYTE	IB,QB,VB,MB,SMB,SB,LB,AC,*VD,*LD,*AC 及常数

➢ **配置串口通信参数**

您可以通过 CM01P 的参数设置功能配置串口通信的参数，为串口选择波特率、校验和数据位数，操作如下图所示：

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

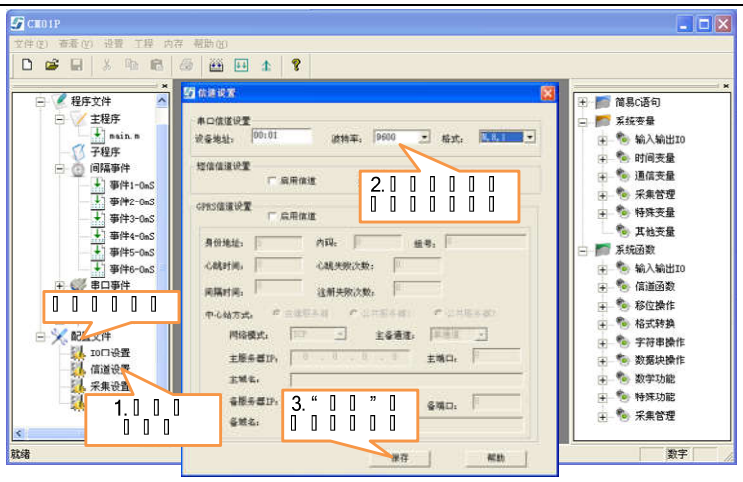


图 3-4 通过 CM01P 配置串口参数

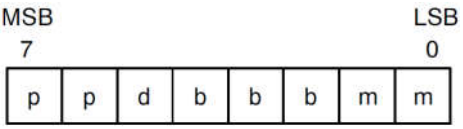
也可以通过 SMB30 和 SMB130 特殊存储器分别配置通讯口 0 和通讯口 1 选择波特率、校验和数据位数。串口通信的配置字节信息如下图所示。

简易串口接收指令 RCVN

当使用简易 RCVN 指令时，串口将依据 CM01P 配置的参数进行初始化（SM30 和 SMB130 配置的串口参数无效），并且自动开放全局中断（ENI）。

标准串口接收指令 RCV

当使用 RCV 指令时，当程序中通过 SM30 和 SMB130 对串口进行配置后，那么通过 CM01P 配置的串口参数无效，也就说特殊寄存器配置串口参数优先级比 CM01P 编程软件设置参数高。



SMB30 = 端口 0

SMB130 = 端口 1

pp： 校验选择

00= 不校验

01= 偶校验

10= 不校验

11= 奇校验

d： 每个字符的数据位

0= 每个字符 8 位

1= 每个字符 7 位

bbb：自由口波特率

000=38,400 波特

001=19,200 波特

010=9,600 波特

011=4,800 波特

100=2,400 波特

101=1,200 波特

110=115.2K 波特

111=57.6K 波特

mm：协议选择

保留（缺省设置自由口模式）

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

图 3-5 用于串口配置的 SM 控制字节（SMB30 或 SMB130）

➢ 发送数据

您可以通过任意发送数据指令来发送一包串口数据，串口发送数据有两种模式：中断式和非中断式。

串口中断式发送数据指令的执行和发送过程是不同步的。采用中断的方式发送串口数据时，执行中断式发送数据指令后，不等到数据全部发送完成，就认为这条语句已经执行完成，继续执行这条语句的下一条语句。当前的这个单字节发送成功后，无线 PLC 会中断当前正在执行的语句，开始发送下一个数据，然后再接着执行中断前被打断的语句。依次类推，直到所有需要发送的数据发送完成。当最后一个串口数据也发送完成后，产生一个串口发送完成事件。使用中断式发送数据时，当要数据还没有发送结束时，SM4.5（串口 0）和 SM4.6（串口 1）为 0（信道忙），当包发送成功后，SM4.5 和 SM4.6 为 1（信道不忙），SM7.1 和 SM7.7 为 1（通过中断式发送完成一包数据）。SM7.1 和 SM7.7 置位后，不会自动复位，需要您通过指令复位。

非中断式语句的执行和发送过程是同步的。语句执行和发送过程同步是指在执行发送串口数据的程序语句时，无线 PLC 开始发送串口数据，直到数据发送完成后，这条语句才执行完成，然后再执行下一条语句。采用非中断式发送数据，无论 SM4.5 或 SM4.6 在指令执行前是那种状态，执行完非中断式发送数据后，SM4.5 或 SM4.6 都为 1（信道空闲），SM7.1 或 SM7.7 不做任何变化（不会产生串口中断发送完成）。

注意，如果向通过串口发送中文，由于中文的常用编格式有三种：GB2312、unicode 码和 UTF-8，CM01P 编程环境默认的格式是 Unicode 码，如果使用 GB2312 中文格式字符串，则需要在字符串前加上“GB#”,如果使用 UTF-8 中文格式字符串，则需要在字符串前加上“UTF#”。例如"GB#北京捷麦顺驰" 和"UTF#北京捷麦顺驰"分别表示 GB2312 格式和 UTF-8 格式的“北京捷麦顺驰”。

XMT 标准串口发送数据

XMT TBL , PORT

使用中断方式的发送，TBL 前两个字节为长度，后面为具体的内容，发送时，前两个字节的长度不发送。PORT 为串口号。

图 3-6 发送缓冲区的格式

XMTS 串口发送字符串数据

XMTS　DATA　, PORT

使用中断方式的发送，DATA 为发送的字符串内容，PORT 为串口号。发送时，字符串的第一个字节长度不发送。

XMTM 串口发送任意大小的数据

XMTM　　LEN ,DATA ,PORT

使用中断方式的发送，LEN 为发送的数据内容长度（1-65535），DATA 为发送的内容，PORT 为串口号。

XMTN 串口发送任意位置和大小的数据

XMTN　LEN ,SEAT, DATA ,PORT

使用中断方式的发送，LEN 为发送的数据内容长度（1-65535），SEAT 为发送的开始位置是在 DATA 中的那个位置上（0-255），DATA 为发送的内容，PORT 为串口号。

*XMTSF 串口发送字符串数据

XMTSF　DATA　, PORT

使用非中断方式的发送，DATA 为发送的字符串内容，PORT 为串口号。发送时，字符串的第一个字节长度不发送。

XMTMF 串口发送任意大小的数据

XMTMF　　LEN ,DATA ,PORT

使用非中断方式的发送，LEN 为发送的数据内容长度（1-65535），DATA 为发送的内容，PORT 为串口号。

XMTNF 串口发送任意位置和大小的数据

XMTNF　LEN ,SEAT, DATA ,PORT

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

使用非中断方式的发送，发送的大小由参数 1 决定（1-65535），位置由参数 2（0-255），内容的首字节为参数 3。

如果有一个中断服务程序连接到发送结束事件（发送完成事件）上，在发送完缓冲区中的最后一个字符时，则会产生一个中断（对端口 0 为中断事件 1，对端口 1 为中断事件 7）。

➢ 接收数据

接收指令使您能够接收一个字节或多个字节的缓冲区，最多为 1024 个字节。下图给出了接收缓冲区的格式。

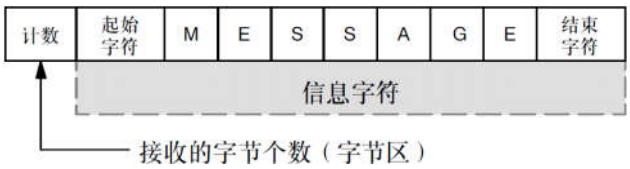


图 3-7 接收缓冲区的格式

计数的个数为两个字节，范围从 0 到 1024，后面为接收到的串口数据。

当无线 PLC 的串口收到数据时，将收到的数据先放着接收缓存区中（此时并不通知用户收到数据了），如果后面还有数据，会将这些数据依次按照先后顺序存放在这个缓存区中，当接收超过 3.5 的字节时间还没有数据时，就认为一包数据接收完毕，即包结束的判断标准是 3.5 个字节无数据传输。传输一包数据至少要求传输数据之前和之后有 3.5 个字节的空闲时间（没有数据传输），如下图所示：（无线 PLC 不允许您选择信息的启始和结束条件）

如果有一个中断服务程序连接到接收信息完成事件上，串口接收到一包数据时，无线 PLC 会产生一个中断（对端口 0 为中断事件 0，对端口 1 为中断事件 6）。

您可以不使用中断，通过监视 SM27.0（端口 0）或者 SM27.6（端口 1）来接收信息。当接收到一包串口数据时，这一标志位置位。注意，该标志不会自动复位，需要用户可以通过指令清该标志。

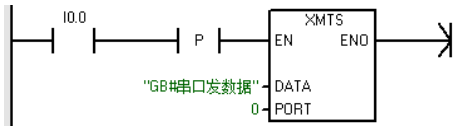
表 3-8 特殊存储器字节 SMB27

SM位	描述（只读）
格式	MSB LSB 7 0 U2T | U2R | — | — | — | — | R1T | U1R

SM27.0	串口(uart),0收到一包数据
SM7.7	串口(uart)1通过中断的方式发送完成一包数据（非中断发送不会置位）

串口示例1：发送串口指令

MAIN

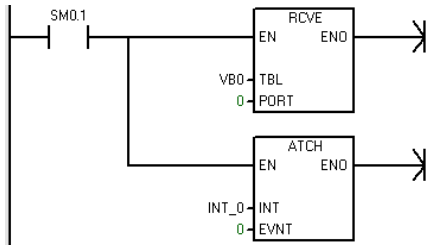



```
Network1          //本程序发送一条串口数据
                  //填写发送的短信内容
                  //调用发送短信指令
LD      I0.0
EU
XMTS    "GB#串口发数据",0
```

串口示例2：串口数据控制线圈

MAIN

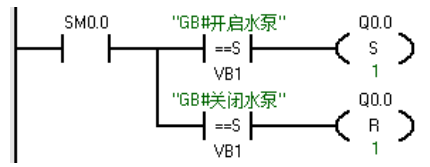
Network1




```
Network1 //本程序接收串口数据
         //当串口内容为“开启水泵”时,控制线圈0置位
         //当串口内容为“开启水泵”时,控制线圈0复位
LD      SM0.1
LPS
SRCV    VB0
LPP
ATCH    INT_0,0
```

INT_0

Network1

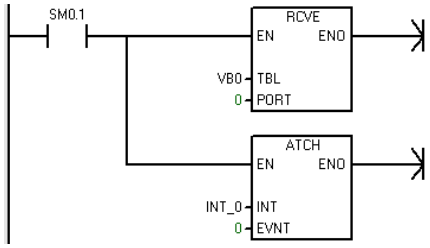



```
Network1
LD      SM0.0
LPS
AS=     "GB#开启水泵",VB1
S       Q0.0,1
LPP
AS=     "GB#关闭水泵",VB1
R       Q0.0,1
```

串口示例3：串口发送和接收指令

MAIN

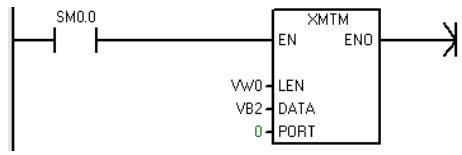
Network1




```
Network1          //本程序接收一包数据
LD      SM0.1
LPS
SRCV    VB0
LPP
ATCH    INT_0,0
```

INT_0

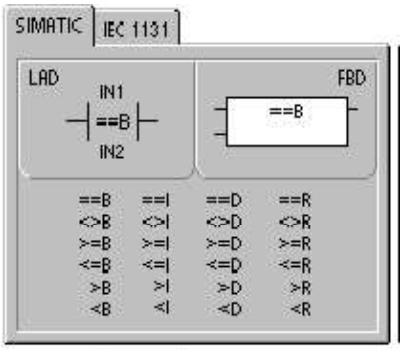
Network1

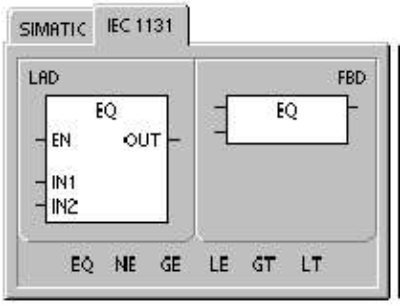


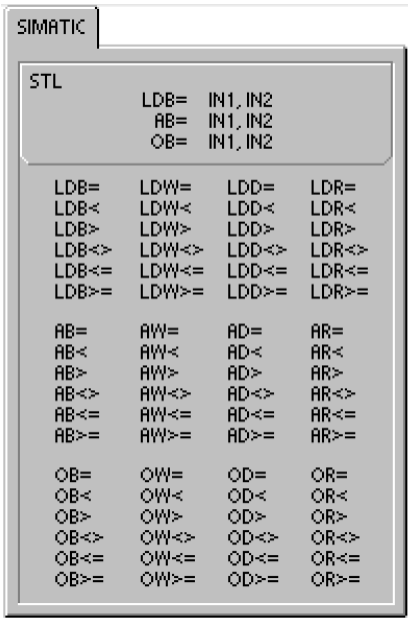

```
Network1          ///向用户回送串口信息
//VW0：接收到的串口内容长度
//VB2~ ：串口的内容
LD      SM0.0
XMTM    VW0,VB2,0
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.4 比较指令







3.4.1 数值比较

比较指令用于比较两个数值：

IN1 = IN2IN1 >= IN2IN1 <= IN2

IN1 > IN2IN1 < IN2IN1 <> IN2

字节比较操作是无符号的。

整数比较操作是有符号的。

双字比较操作是有符号的。

实数比较操作是有符号的。

对于 LAD 和 FBD：当比较结果为真时，比较指令使触点闭合（LAD）或者输出接通（FBD）。

对于 STL：当比较结果为真时，对 1 进行 LD，A 或 O 操作，并置入栈顶。

当您使用 IEC 比较指令时，您可以使用各种数据类型作为输入。但是，两个输入的数据类型必须一致。

注意

下列情况是致命错误，并且会导致无线 PLC 立即停止执行用户程序：

- 非法的间接地址（任意比较指令）
- 非法的实数（例如：NAN），（实数比较指令）

为了避免这些情况的发生，在执行比较指令之前，要确保合理使用了指针和存储实数的数值单元。

不管能流的状态如何，比较指令都会被执行。

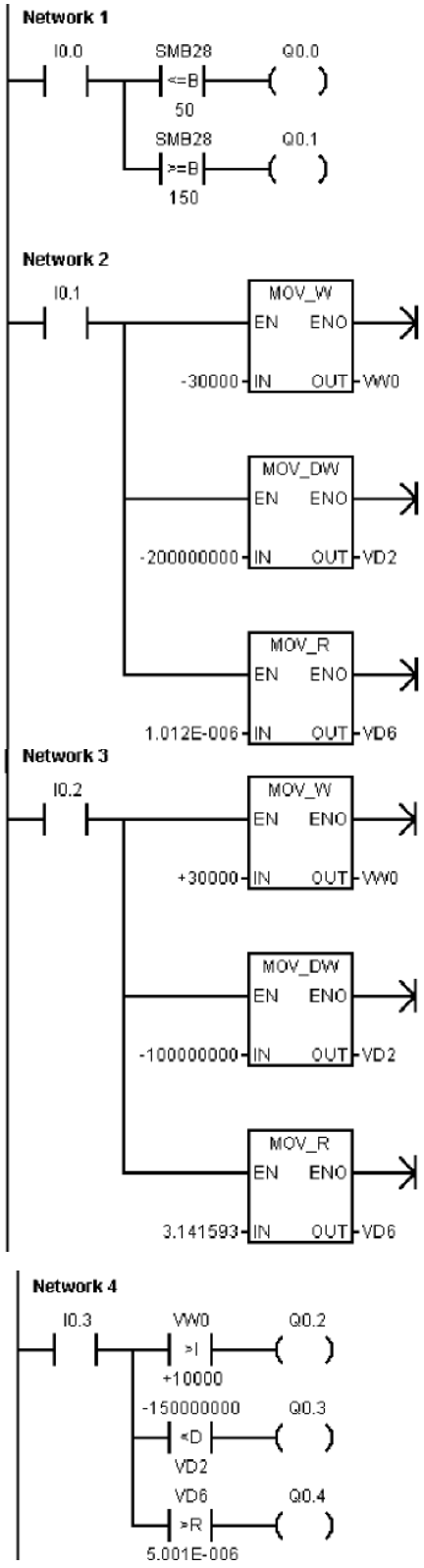
表 3-9 比较指令的有效操作数

输入/输出	类型	操作数
IN1、IN2	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

OUT	BOOL	I、Q、V、M、SM、S、T、C、L、能流

示例：比较指令



Network1 //调节模拟调节电位器0
//来改变SMB28的数值。
//当SMB28中的数值小于等于50时，
//Q0.0输出
//当SMB28中的数值大于等于150时，
//Q0.1输出
//当比较结果为真时，
//状态指示器点亮。

```
LD      I0.0
LPS
AB<=    SMB28，50
=       Q0.0
LPP
AB>=    SMB28，150
=       Q0.1
```

Network2
//在V存储器地址中装载较小的数值，
//使比较结果为假并且关闭状态指示器。

```
LD      I0.1
MOVW    --30000，VW0
MOVD    --200000000，VD2
MOVR    1.012E--006，VD6
```

Network3 //在V存储器地址中装载较大的数值，
//使比较结果为真并且点亮状态指示器。

```
LD      I0.2
MOVW    +30000，VW0
MOVD    -100000000，VD2
MOVR    3.141593，VD6
```

Network4 //整数字比较检测VW0>+10000是否为真。
//在程序中使用常数是为了显示不同的数据类型
//您也可以比较两个存储在程序存储区中的数值
//例如：VW0>VW100

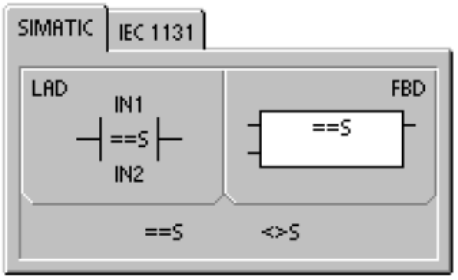
```
LD      I0.3
LPS
AW>     VW0，+10000
=       Q0.2
LRD
AD<     --150000000，VD2
=       Q0.3
LPP
```

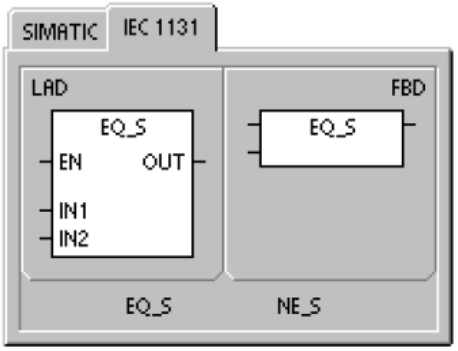

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

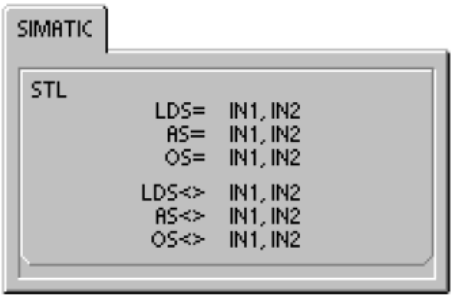
	AR> VD6，5.001E--006 = Q0.4

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.4.2 字符串比较







字符串比较指令比较两个字符串的 ASCII 码字符：

IN1=IN2　　IN1<>IN2

当比较结果为真时，比较指令使触点闭合（LAD）或者输出接通（FBD），或者对 1 进行 LD，A 或 O 操作，并置入栈顶（STL）。

注意

下列情况是致命错误，并且会导致无线 PLC 立即停止执行用户程序：

- 非法的间接地址（任意比较指令）
- 字符串的长度超过 254 个字符（字符串比较指令）
- 一个字符串的起始地址和长度使它不适合所指定的存储区（字符串比较指令）

为了避免这些情况的发生，在执行比较指令之前，要确保合理使用了指针和保存 ASCII 码字符串的存储区。确保一个保存 ASCII 码字符串的缓冲区能够在指定的存储区完整的保留。

无论使能位状态如何，"比较"指令均会执行。

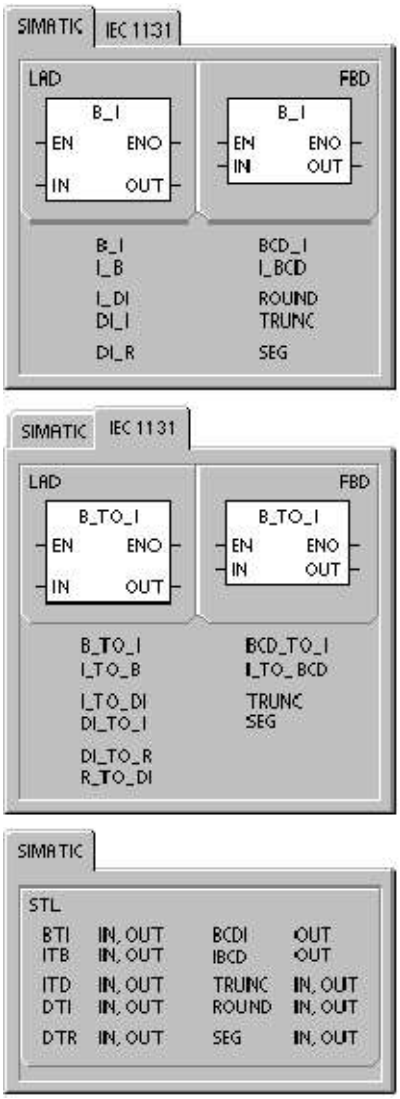
表 3-10字符串比较指令的有效操作数

输入/输出	定时器类型	操作数
IN1	STRING	VB，LB，*VD，*LD，*AC，常数
IN2	STRING	VB、LB、*VD、*LD、*AC
OUT	BOOL	I、Q、V、M、SM、S、T、C、L、能流

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.5 转换指令

3.5.1 标准转换指令



数字转换

字节转为整数（BTI）、整数转为字节（ITB）、整数转为双整数（ITD）、双整数转为整数（DTI）、双整数转为实数（DTR）、BCD 码转为整数（BCDI）和整数转为 BCD 码（IBCD）。以上指令将输入值 IN 转换为指定的格式并存储到由 OUT 指定的输出值存储区中。例如：您可以将双整数值转为实数值；您也可以在整数和 BCD 码格式之间相互转换。

四舍五入和取整

四舍五入指令（ROUND）将一个实数转为一个双整数值，并将四舍五入的结果存入 OUT 指定的变量中。

取整指令（TRUNC）将一个实数转为一个双整数值，并将实数的整数部分作为结果存入 OUT 指定的变量中。

包络段数

段码指令（SEG）允许您产生一个点阵，用于点亮七段码显示器的各个段。

表 3-11 标准转换指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、AC、*VD、*LD、*AC、常数
	DINT	ID、QD、VD、MD、SMD、SD、LD、HC、AC、*VD、*LD、*AC、常数
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
	DINT、REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

➢ BCD 码转为整数和整数转为 BCD 码的操作(BCDI)

BCD 码转整数指令（BCDI）将一个 BCD 码 IN 的值转换成整数值，并且将结果存入 OUT 指定的变量中。IN 的有效范围是 0 到 9999 的 BCD 码。

整数转 BCD 码指令（IBCD）将输入的整数值 IN 转换成 BCD 码，并且将结果存入 OUT 指定的变量中。IN 的有效范围是 0 到 9999 的整数。

使 ENO=0 的错误条件：

- SM1.6（无效的 BCD 码）
- 0006（间接寻址）

受影响的 SM 标志位：

- SM1.6（无效的 BCD 码）

➢ 双整数转为实数指令的操作(DTR)

双整数转实数指令（DTR）将一个 32 位，有符号整数值 IN 转换成一个 32 位实数，并将结果存入 OUT 指定的变量中。

使 ENO=0 的错误条件：

- 0006（间接寻址）

➢ 双整数转为整数指令的操作(DTI)

双整数转整数指令（DTI）将一个双整数值 IN 转换成一个整数值，并将结果存入 OUT 指定的变量中。

如果所转换的数值太大以致于无法在输出中表示则溢出标志位置位并且输出不会改变。

使 ENO=0 的错误条件：

- SM1.1（溢出）
- 0006（间接寻址）

受影响的 SM 标志位：

- SM1.1（溢出）

➢ 整数转为双整数指令的操作(ITD)

整数转双整数指令（ITD）将整数值 IN 转换成双整数值，并且存入 OUT 指定的变量中。符号

位扩展到高字节中。

使 ENO=0 的错误条件：

- 0006（间接寻址）

➢ 字节转为整数指令的操作(BTI)

字节转整数指令（BTI）将字节值 IN 转换成整数值，并且存入 OUT 指定的变量中。字节是无符号的，因而没有符号位扩展。

使 ENO=0 的错误条件：

- 0006（间接寻址）

➢ 整数转为字节指令的操作(ITB)

整数转字节指令（ITB）将一个字的值 IN 转换成一个字节值，并且存入 OUT 指定的变量中。只有 0 到 255 中的值被转换，所有其他值会产生溢出并且输出不会改变。

使 ENO=0 的错误条件：

- SM1.1（溢出）
- 0006（间接寻址）

受影响的 SM 标志位：

- SM1.1（溢出）

提示

如果想将一个整数转换成实数，先用整数转双整数指令，再用双整数转实数指令。

➢ 四舍五入取整和取整指令的操作

四舍五入取整指令（ROUND）将实数值 IN 转换成双整数值，并且存入 OUT 指定的变量中。如果小数部分大于等于 0.5，则数字向上取整。

取整指令（TRUNC）将一个实数值 IN 转换成一个双整数，并且存入 OUT 指定的变量中。只有实数的整数部分被转换，小数部分舍去。

使 ENO=0 的错误条件：

- SM1.1（溢出）

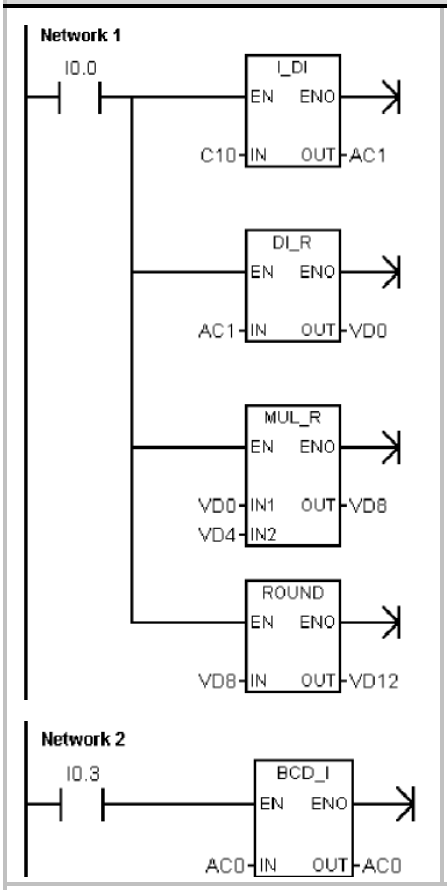
- 0006（间接寻址）

受影响的 SM 标志位：

- SM1.1（溢出）

如果所转换的不是一个有效的实数，或者其数值太大以致于无法在输出中表示，则溢出标志位置位并且输出不会改变。

示例1：电话呼叫指令



Network1 //转换英寸为厘米：
//1. 装载计数值（英寸）到AC1。
//2. 将该值转换为实数。
//3. 乘以2.54（转换为厘米）。
//4. 将数值转回整数。

```
LD I0.0
ITD C10，AC1
DTR AC1，VD0
MOVR VD0，VD8
*R VD4，VD8
ROUND VD8，VD12
```

Network2 //将BCD码转为整数

```
LD I0.3
BCDI AC0
```

双字整数转实数和四舍五入取整

C10	101	计数值=101英寸
VD0	101.0	计数值（实数形式）
VD4	2.54	2.54常数（英寸到厘米）
VD8	256.54	256.54厘米数（实数形式）
VD12	257	257厘米数（双整数形式）

BCD转整数

AC0	1234

BCDI

AC0	04D2

➢ 段码指令的操作

要点亮七段码显示器中的段，可以使用段码指令。段码指令将 IN 中指定的字符（字节）转换生成一个点阵并存入 OUT 指定的变量中。

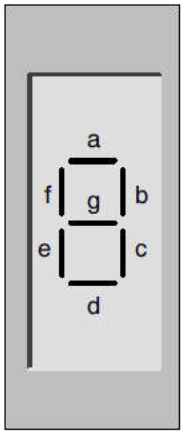
北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

点亮的段表示的是输入字节中低 4 位所代表的字符。下图给出了段码指令使用的七段码显示器的编码。

使 ENO=0 的错误条件：

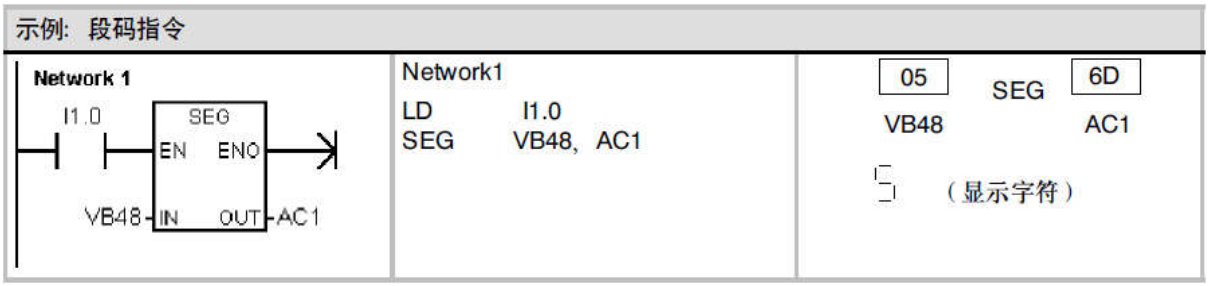
- 0006（间接寻址）

输入 LSD	七段码显示器	输出 -gfe dcba
0	0	0011 1111
1	1	0000 0110
2	2	0101 1011
3	3	0100 1111
4	4	0110 0110
5	5	0110 1101
6	6	0111 1101
7	7	0000 0111

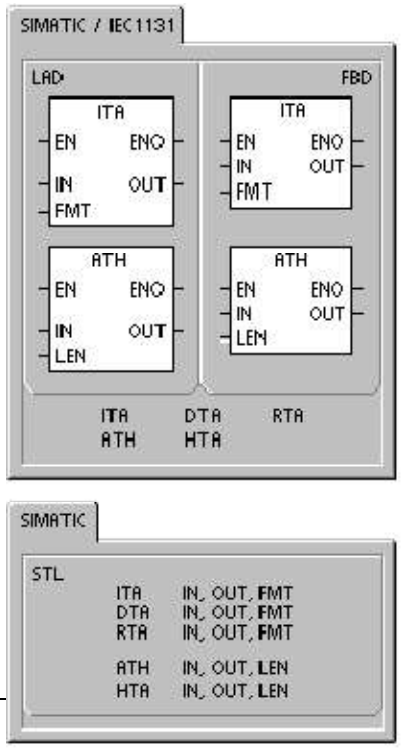


输入 LSD	七段码显示器	输出 -gfe dcba
8	8	0111 1111
9	9	0110 0111
A	A	0111 0111
B	b	0111 1100
C	C	0011 1001
D	d	0101 1110
E	E	0111 1001
F	F	0111 0001

图 3-8 七段码显示器的编码



3.5.2 ASCII 码转换指令



有效的 ASCII 码字符为十六进制的 30 到 39 和 41 到 46。

在 ASCII 码和十六进制数之间相互转换 ASCII 码转十六进制数指令（ATH）将一个长度为 LEN 从 IN 开始的 ASCII 码字符串转换成从 OUT 开始的十六进制数。十六进制数转 ASCII 码指令（HTA）将从输入字节 IN 开始的十六进制数，转换成从 OUT 开始的 ASCII 码字符串。被转换的十六进制数的位数由长度 LEN 给出。

能够被转换的 ASCII 码字符串或者十六进制数的最大数量为 255。有效 ASCII 码输入有效的 ASCII 码输入字符是 0 到 9 的

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

十六进制数代码值 30 到 39，和大写字符 A 到 F 的十六进制数代码值 41 到 46 这些字母数字字符。

使 ENO=0 的错误条件：

- SM1.7（非法的 ASCII 码）只对 ATH 有效
- 0006（间接寻址）
- 0091（操作数超出范围）

受影响的 SM 标志位：

- SM1.7（非法的 ASCII 码）

➢ 将数值转为 ASCII 码

整数转 ASCII 码（ITA）、双整数转 ASCII 码（DTA）和实数转 ASCII 码（RTA）指令，分别将整数、双整数或实数值转换成 ASCII 码字符。

表 3-12ASCII 码转换指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC
	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
LEN、FMT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC

➢ 整数转 ASCII 码指令的操作数

整数转 ASCII 码（ITA）指令将一个整数字 IN 转换成一个 ASCII 码字符串。格式 FMT 指定小数点右侧的转换精度和小数点是使用逗号还是点号。转换结果放在 OUT 指定的连续 8 个字节中。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 非法的格式
- nnn>5

ASCII 码字符串始终是 8 个字节。

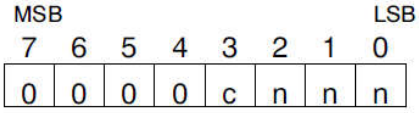
整数转 ASCII 码指令的格式操作数如图 6-15 所示。输出缓冲区的大小始终是 8 个字节，nnn 表

示输出缓冲区中小数点右侧的数字位数。nnn 的合理范围是 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。对于 nnn 大于 5 的情况，输出缓冲区会被空格键的 ASCII 码填冲。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。高 4 位必须为 0。

下图中给出了一个数值的例子，其格式为使用点号（c=0），小数点右侧有三位小数（nnn=011）。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（--）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 数值在输出缓冲区中是右对齐的。

FMT



c=逗号（1）或者点号（0）
nnn=小数点右侧的位数

	输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6	输出+7
输入=12				0	.	0	1	2
输入=-123			-	0	.	1	2	3
输入=1234				1	.	2	3	4
输入=-12345		-	1	2	.	3	4	5

图 3-9 整数转 ASCII 码（ITA）指令的 FMT 操作数

➢ 双整数转 ASCII 码指令操作

双整数转 ASCII 码（DTA）指令将一个双字 IN 转换成一个 ASCII 码字符串。格式操作数 FMT 指定小数点右侧的转换精度。转换结果存储在从 OUT 开始的连续 12 个字节中。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 非法的格式
- nnn>5

输出缓冲区的大小总是 12 个字节。

下图描述了双整数转 ASCII 码指令的格式操作数。nnn 表示输出缓冲区中小数点右侧的数字位数。nnn 的合理范围是 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。对于 nnn 大于 5 的情况，输出缓冲区会被空格键的 ASCII 码填冲。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。高 4 位必须为 0。

下图给出了一个数值的例子，其格式为使用点号（c=0），小数点右侧有四位小数（nnn=100）。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（--）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 数值在输出缓冲区中是右对齐的。

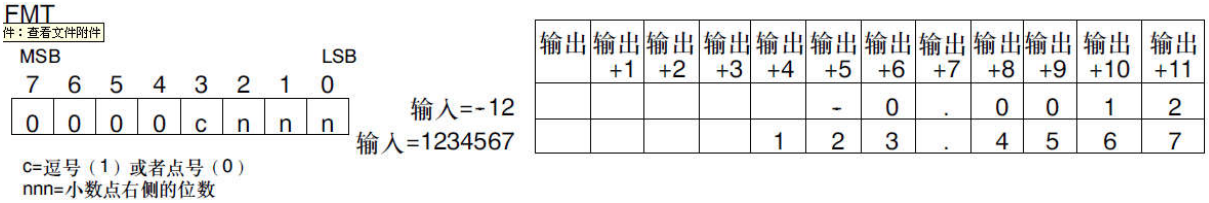


图 3-10 双整数转 ASCII 码（DTA）指令的 FMT 操作数

➢ 实数转 ASCII 码指令操作

实数转 ASCII 码指令（RTA）将一个实数值 IN 转为 ASCII 码字符串。格式操作数 FMT 指定小数点右侧的转换精度，小数点是用逗号还是用点号表示和输出缓冲区的大小。

转换结果存储在从 OUT 开始的输出缓冲区中。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- nnn>5
- ssss<3
- ssss<OUT 中的字符个数

结果 ASCII 码字符的位数（或长度）就是输出缓冲区的大小，它的值可以在 3 到 15 字节或字符之间。

无线 PLC 的实数格式支持最多 7 位小数。试图显示 7 位以上的小数会产生一个四舍五入错误。

下图是对 RTA 指令中格式操作数 FMT 的描述。ssss 表示输出缓冲区的大小。0、1 或者 2 个字节的大小是无效的。nnn 表示输出缓冲区中小数点右侧的数字位数。nnn 的有效范围为 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。对于 nnn 大于 5 或者指定的输出缓冲区太小以致于无法存储转换值的情况，输出缓冲区会被空格键的 ASCII 码填冲。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。

下图中给出了一个数值的例子，其格式为：使用点号（c=0）、小数点右侧有 1 位小数（nnn=001）和 6 个字节的缓冲区大小（ssss=0110）。输出缓冲区的格式符合以下规则：

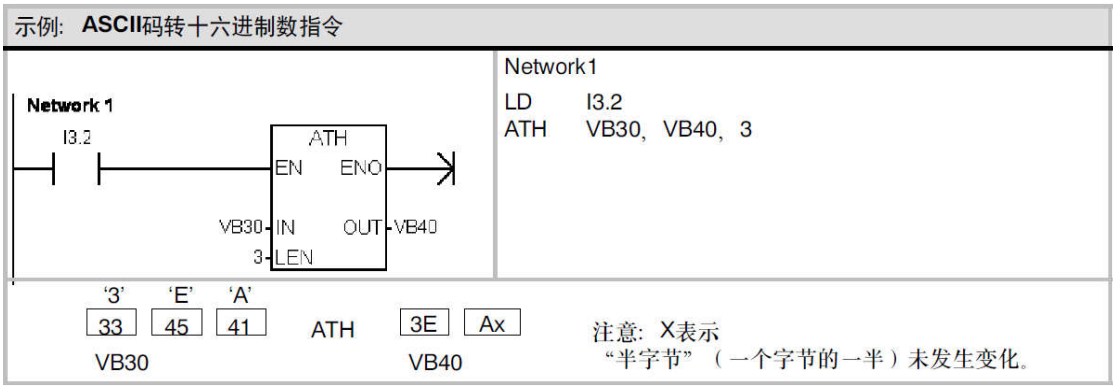
- 正数值写入输出缓冲区时没有符号位。

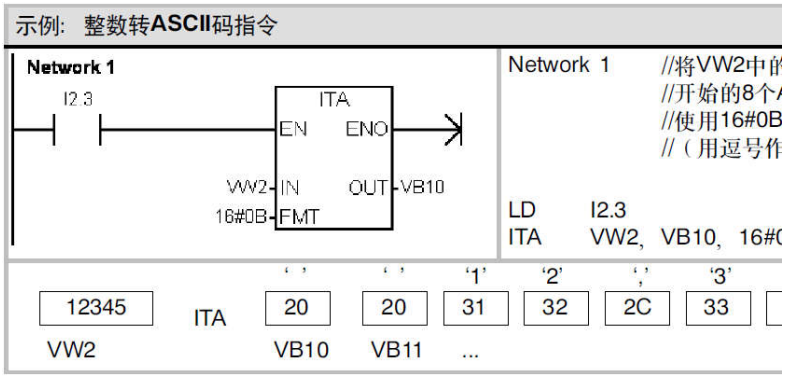
北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

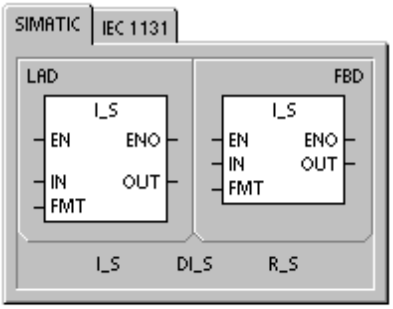
- 负数值写入输出缓冲区时以负号（--）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 小数点右侧的数值按照指定的小数点右侧的数字位数被四舍五入。
- 输出缓冲区的大小应至少比小数点右侧的数字位数多三个字节。
- 数值在输出缓冲区中是右对齐的。

	输出	输出+1	输出+2	输出+3	输出+4	输出+5
输入=1234.5	1	2	3	4	.	5
输入=-0.0004				0	.	0
输入=-3.67526			-	3	.	7
输入=1.95				2	.	0

图 3-11 实数转 ASCII 码（RTA）指令的 FMT 操作数



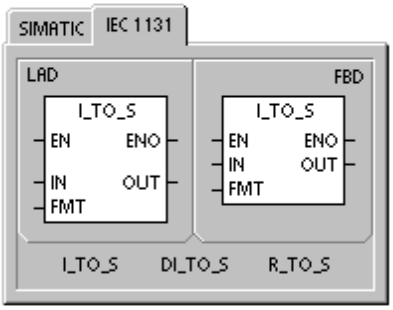


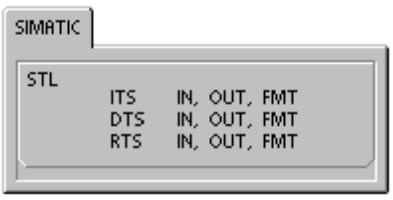


3.5.3 字符串转换指令

➢ 将数值转换为字符串

整数转字符串（ITS）、双整数转字符串（DTS）和实数转字符串（RTS）指令，将整数、双整数或实数值（IN）转换成 ASCII 码字符串（OUT）。





北京捷麦顺驰科技有限公司 地址：北京市
电传：（010）58076471 网址：http://w

➢ 整数转字符串的操作

整数转字符串指令（ITS）将一个整数字 IN 转换为 8 个字符长的 ASCII 码字符串。格式操作数 FMT 指定小数点右侧的转换精度和使用逗号还是点号作为小数点。结果字符串被写入从 OUT 开始的 9 个连续字节中。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 非法格式（nnn>5）

下图是对整数转字符串指令中格式操作数的描述。输出字符串的长度总是 8 个字符。nnn 表示输出缓冲区中小数点右侧的数字位数。nnn 的合理范围为 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。如果 nnn 的值大于 5，输出是由 8 个空格键的 ASCII 码组成的字符串。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。格式操作数的高 4 位必须为 0。

下图给出了一个数值的例子，其格式为：使用点号（c=0）并且小数点后保留 3 位小数。OUT 的值为字符串的长度。

- 输出缓冲区的格式符合以下规则：
- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（-）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 数值在输出缓冲区中是右对齐的。

3-13 数值转字符串指令的有效操作数表

输入/输出	数据类型	操作数
IN	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AIW、*VD、*LD、*AC、常数
	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
FMT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	STRING	VB、LB、*VD、*LD、*AC

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c=逗号（1）或者点号（0）
nnn=小数点右侧的位数

	输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6	输出+7	输出+8
输入=12	8				0	.	0	1	2
输入=-123	8			-	0	.	1	2	3
输入=1234	8				1	.	2	3	4
输入=-12345	8		-	1	2	.	3	4	5

图 3-12 整数转字符串指令的 FMT 操作数

➢ 双整数转字符串指令操作(DTS)

双整数转字符串指令(DTS)将一个双整数 IN 转换为一个长度为 12 个字符的 ASCII 码字符串。格式操作数 FMT 指定小数点右侧的转换精度和使用逗号还是点号作为小数点。结果字符串被写入从 OUT 开始的连续 13 个字节。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 非法格式（nnn>5）

下图是对整数转字符串指令中格式操作数的描述。输出字符串的长度总是 8 个字符。nnn 表示输出缓冲区中小数点右侧的数字位数。nnn 的合理范围为 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。如果 nnn 的值大于 5，输出是由 12 个空格键的 ASCII 码组成的字符串。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。格式操作数的高 4 位必须为 0。

下图中给出一个数值的例子，其格式为：使用点号（c=0）并且小数点后保留 4 位小数。OUT 的值为字符串的长度。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（-）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 数值在输出缓冲区中是右对齐的。

FMT

MSB							LSB
7	6	5	4	3	2	1	0
0	0	0	0	c	n	n	n

c=逗号（1）或者点号（0）
nnn=小数点右侧的位数

	输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6	输出+7	输出+8	输出+9	输出+10	输出+11	输出+12
输入=12	12					“	”	0	.	0	0	1	2
输入=-1234567	12					1	2	3	.	4	5	6	7

图 3-13 整数转字符串指令的 FMT 操作数

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

➢ 实数转字符串指令操作

实数转字符串指令（RTS）将一个实数值 IN 转换为一个 ASCII 码字符串。格式操作数 FMT 指定小数点右侧的转换精度和使用逗号还是点号作为小数点。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）

转换结果放在从 OUT 开始的一个字符串中。结果字符串的长度由格式操作数给出，它可以是 3 到 15 个字符。

无线 PLC 的实数格式支持最多 7 位小数。试图显示 7 位以上的小数会产生一个四舍五入错误。

下图是对实数转字符串指令中格式操作数的描述。ssss 表示输出字符串的长度。0、1 或者 2 个字节的大小是无效的。nnn 表示输出缓冲区中小数点右侧的数字位数。nnn 的有效范围为 0 到 5。将小数点右侧的位数定为 0，使得所显示的数值没有小数点。对于 nnn 大于 5 或者指定的输出缓冲区太小以致于无法存储转换值的情况，输出缓冲区会被空格键的 ASCII 码填冲。c 指定是用逗号（c=1）或者点号（c=0）作为整数和小数的分隔符。

非法的格式：

- nnn > 5
- ssss < 3
- ssss < 要求的字符数

下图中给出了一个数值的例子，其格式为：使用点号（c=0），小数点右侧有 1 位小数（nnn=001）和 6 个字节的缓冲区大小（ssss=0110）。OUT 的值为字符串的长度。输出缓冲区的格式符合以下规则：

- 正数值写入输出缓冲区时没有符号位。
- 负数值写入输出缓冲区时以负号（-）开头。
- 小数点左侧的开头的 0（除去靠近小数点的那个之外）被隐藏。
- 小数点右侧的数值按照指定的小数点右侧的数字位数被四舍五入。
- 输出缓冲区的大小应至少比小数点右侧的数字位数多三个字节。
- 数值在输出缓冲区中是右对齐的。

FMT

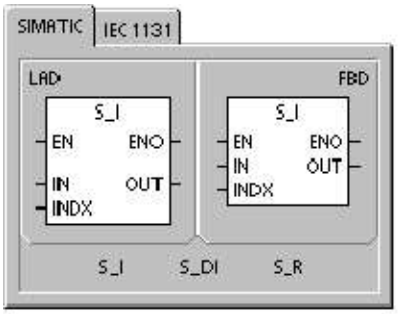
MSB							LSB
7	6	5	4	3	2	1	0
s	s	s	s	c	n	n	n

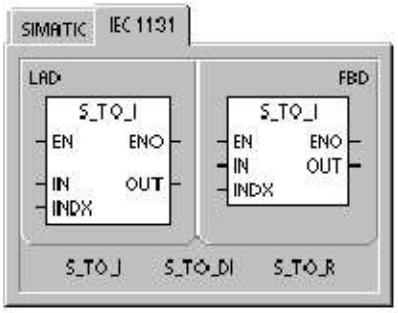
ssss=输出字符串长度
c=逗号（1）或者点号（0）
nnn=小数点右侧的位数

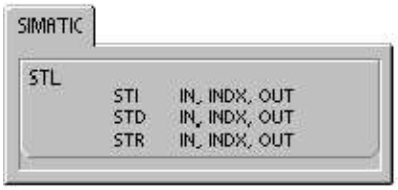
	输出	输出+1	输出+2	输出+3	输出+4	输出+5	输出+6
输入=1234.5	6	1	2	3	4	.	5
输入=-0.0004	6				0	.	0
输入=-3.67526	6			-	3	.	7
输入=1.95	6				2	.	0

图 3-14 实数转字符串指令的 FMT 操作数

➢ 将子字符串转换为数字值







子字符串转整数（STI）、子字符串转双整数（STD）和子字符串转实数（STR）指令，将从偏移量 INDX 开始的字符串值 IN 转换成整数/双整数或实数值 OUT。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（偏移量=0）
- SM1.1（溢出）

子字符串转整数和子字符串转双整数指令将字符串转换为以下格式：

[空格][+或-][数字 0 - 9]

子字符串转实数指令将字符串转换为以下格式：

[空格][+或-][数字 0 - 9][. 或，][数字 0 - 9]

INDX 值通常设置为 1，从字符串的第一个字符开始转换。

INDX 可以被设置为其它值，从字符串的不同位置进行转换。

这可以被用于字符串中包含非数值字符的情况。例如：输入字符串为“Temperature：77.8”，您可以将 INDX 设为 13，这样就可以跳过字符串开头的“Temperature：”。

子字符串转实数指令不能用于转换以科学计数法或者指数形式表示实数的字符串。指令不会产生溢出错误（SM1.1），但是它会将字符串转换到指数之前，然后停止转换。例如：字符串“1.234E6”转换为实数值 1.234，并且没有错误提示。

当到达字符串的结尾或者遇到第一个非法字符时，转换指令结束。非法字符是指任意非数字（0 - 9）字符。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

当转换产生的整数值过大以致输出值无法表示时，溢出标志（SM1.1）会置位。例如：当输入字符串产生的数值大于 32767 或者小于-32768 时，子字符串转整数指令会置位溢出标志。

当输入字符串中并不包含可以转换的合法数值时，溢出标志（SM1.1）也会置位。例如：如果输入字符串的“A123”，转换指令会置位 SM1.1（溢出）并且输出值保持不变。

表 3-14 子字符串转换为数值指令的有效操作数

输入/输出	数据类型	操作数
IN	STRING	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC、常数
INDX	BYTE	VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	I NT	VW、IW、QW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC
	DINT、REAL	VD、ID、QD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

对于整数和双整数
合法的输入字符串

输入字符串	输出整数
‘123’	123
‘-00456’	-456
‘123.45’	123
‘+2345’	2345
‘000000123ABCD’	123

对于实数
合法的输入字符串

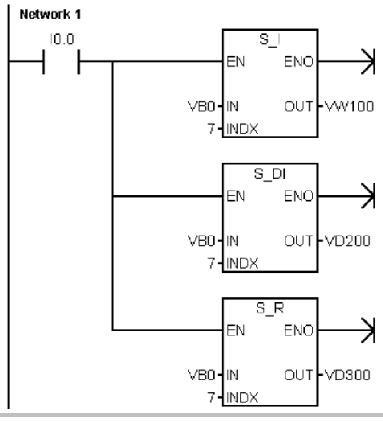
输入字符串	输出实数
‘123’	123.0
‘-00456’	-456.0
‘123.45’	123.45
‘+2345’	2345.0
‘00.000000123’	0.000000123

非法的输入字符串

输入字符串
‘A123’
‘ ’
‘++123’
‘+-123
‘+ 123’

图 3-15 合法和非法的输入字符串的实例

示例：字符串转换：双整数和实数指令程序举例




```
Network1                //将数字字符串转为整数。
                        //将数字字符串转为双整数。
                        //将数字字符串转为实数。
LD I0.0
STI VB0,7,VW100
STD VB0,7,VD200
STR VB0,7,VD300
```

VB0											VB11
11	’T’	’e’	’m’	’p’	’ ’	’ ’	’9’	’8’	’.’	’6’	’F’

程序执行后:
VW100（整数）=98
VD200（双整数）=98
VD300（实数）=98.6

3.5.4 编码和解码指令

编码

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

编码指令（ENCO）将输入字 IN 的最低有效位的位号写入输出字节 OUT 的最低有效“半字节”（4 位）中。

➢ 译码

译码指令（DECO）根据输入字节（IN）的低四位所表示的位号置输出字（OUT）的相应位为 1，输出字的所有其他位都清 0。

➢ SM 标志位和 ENO

对于编码和译码指令，下列条件影响 ENO。

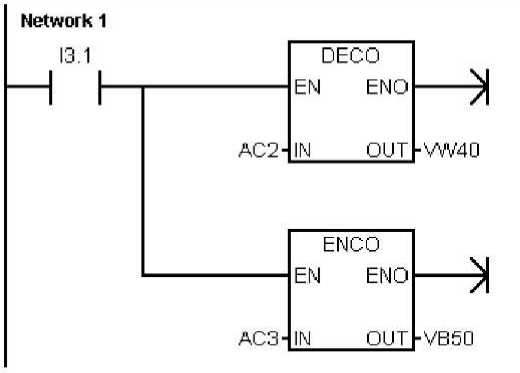
使 ENO=0 的错误条件：

- 0006（间接寻址）

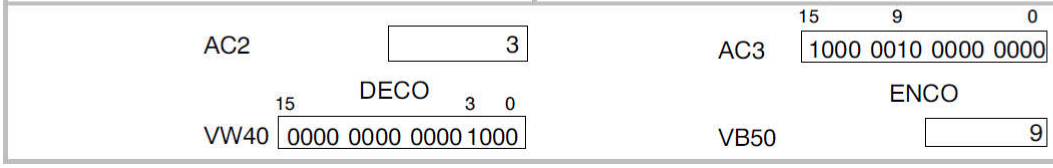
表 3-15 编码和解码指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE WORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
OUT	BYTE WORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC

示例：解码和编码指令

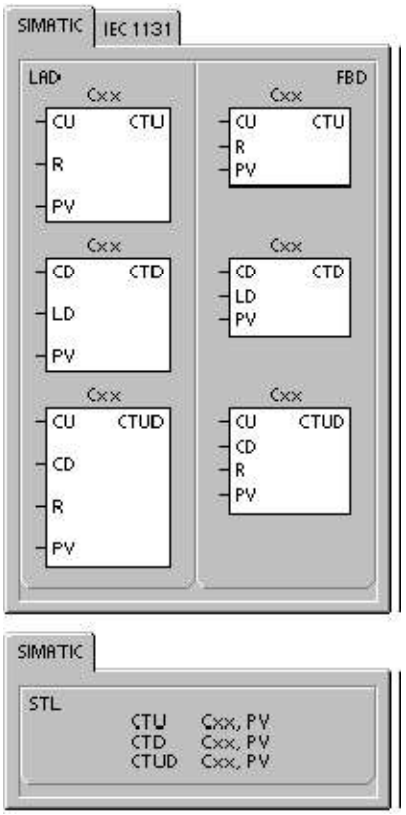



```
Network1      //AC2中包含错误检测位
              //1. 译码指令将VW40中相应位置位。
              //2. 编码指令将最低有效位转换为
              //   错误代码，存入VB50中。

LD      I3.1
DECO    AC2, VW40
ENCO    AC3, VB50
```



北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.6 计数器指令



3.6.1 增计数器

增计数指令（CTU）从当前计数值开始，在每一个（CU）输入状态从低到高时递增计数。当 CXX 的当前值大于等于预置值 PV 时，计数器位 CXX 置位。当复位端（R）接通或者执行复位指令后，计数器被复位。当它达到最大值（32，767）后，计数器停止计数。

STL 操作：

复位输入：栈顶

计数输入：其值被装载在第二个堆栈中。

3.6.2 减计数器

减计数指令（CTD）从当前计数值开始，在每一个（CD）输入状态的低到高时递减计数。当 CXX 的当前值等于 0 时，计数器位 CXX 置位。当装载输入端（LD）接通时，计数器位被复位，并将计数器的当前值设为预置值 PV。当计数值到 0 时，计数器停止计数，计数器位 CXX 接通。

STL 操作：

装载输入：栈顶

计数输入：其值被装载在第二个堆栈中。

3.6.3 增/减计数器

增/减计数指令（CTUD），在每一个增计数输入（CU）的低到高时增计数，在每一个减计数输入（CD）的低到高时减计数。计数器的当前值 CXX 保存当前计数值。在每一次计数器执行时，预置值 PV 与当前值作比较。

当达到最大值(32767)时，在增计数输入处的下一个上升沿导致当前计数值变为最小值(-32768)。当达到最小值(-32768)时，在减计数输入端的下一个上升沿导致当前计数值变为最大值(32767)。

当 CXX 的当前值大于等于预置值 PV 时，计数器位 CXX 置位。否则，计数器位关断。当复位端（R）接通或者执行复位指令后，计数器被复位。当达到预置值 PV 时，CTUD 计数器停止计数。

STL 操作：

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

复位输入：栈顶

减计数输入：其值被装载在第二栈位中。

增计数输入：其值被装载在第三栈位中。

表 3-16 计数器指令的有效操作数

输入/输出	数据类型	操作数
Cxx	WORD	常数（C0到C255）
CU、CD、LD、R	BOOL	I、Q、V、M、SM、S、T、C、L、能流
PV	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、 *LD、*AC、常数

提示

由于每一个计数器只有一个当前值，所以不要多次定义同一个计数器。（具有相同标号的增计数器、增/减计数器、减计数器访问相同的当前值。）

当使用复位指令复位计数器时，计数器位复位并且计数器当前值被清零。计数器标号既可以用来表示当前值，又可以用来表示计数器位。

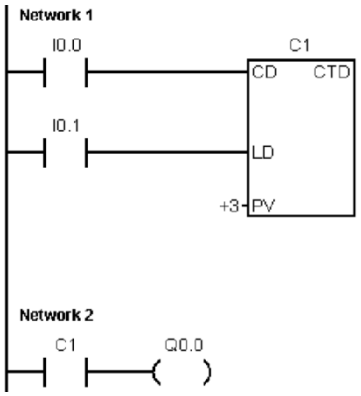
表 3-17 计数器指令的操作

类型	操作	计数器位	上电周期/首次扫描
CTU	CU使当前值递增， 当前值持续递增直至32767	当前值 >= 预设值时， 计数器位接通。	计数器位关断。 当前值可以保留。1
CTUD	CU使当前值递增 CD使当前值递减 当前值持续递增或递减除非计数 器被复位	当前值 >= 预设值时， 计数器位接通。	计数器位关断。 当前值可以保留。1
CTD	CD使当前值递减直至当前值为0.	当前值 >= 预设值时 当前值=0	计数器位关断。 当前值可以保留。1

1. 您可以选择计数器的当前值是否掉电保持。

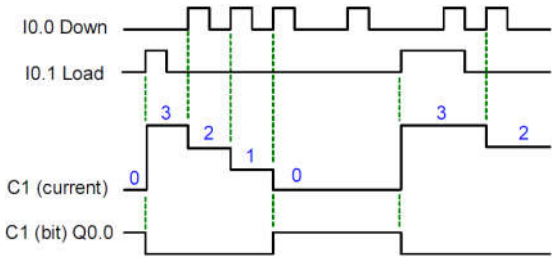
示例：SIMATIC减计数器指令

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

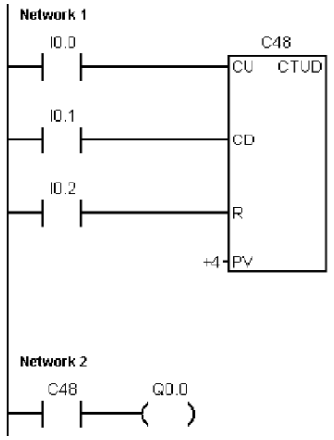



```
Network1        //当I0.1断开时，
                //减计数器C1的当前
                //值从3变到0。I0.0的上升沿使C1的
                //当前值递减。I0.1接通时装载预置值3。
LD              I0.0
LD              I0.1
CTD             C1，+3
Network 2       //当计数器C1的当前值=0时，C1接通。
LD              C1
=               Q0.0
```

时序图



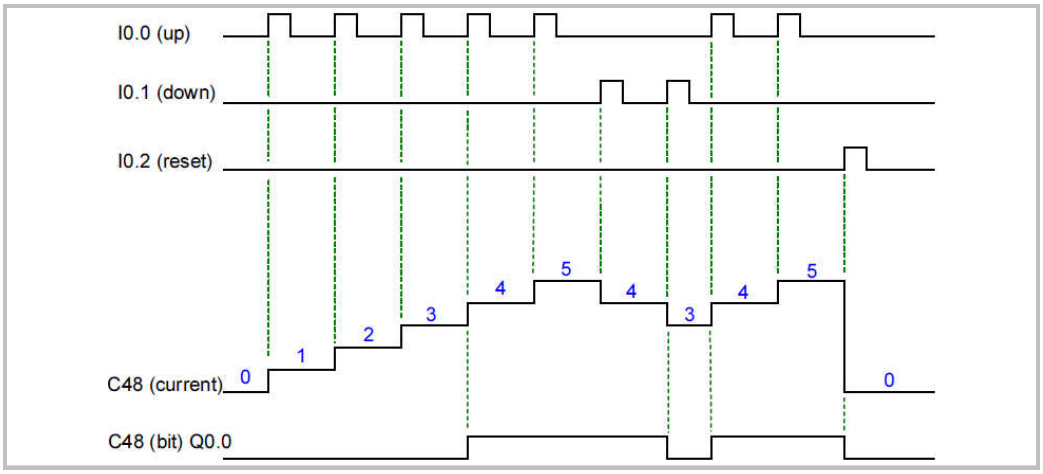
示例：SIMATIC增/减计数器指令程序举例




```
Network 1       // I0.0增计数
                // I0.1减计数
                // I0.2将当前值复位为0
LD              I0.0
LD              I0.1
LD              10.2
CTUD            C48，+4
Network2        //当当前值=4时，
                //将增/减计数器C48接通。
LD              C48
=               Q0.0
```

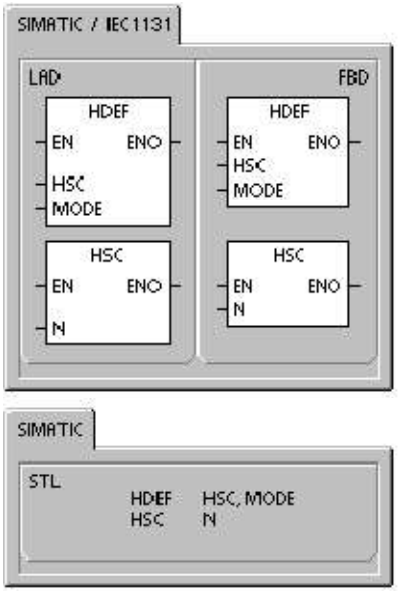
时序图

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com



3.7~高速计数器指令

3.7.1 ~定义高速计数器



定义高速计数器指令（HDEF）为指定的高速计数器（HSCx）选择操作模式。模块的选择决定了高速计数器的时钟、方向、启动和复位功能。

对于每一个高速计数器使用一条定义高速计数器指令。

使 ENO=0 的错误条件：

- 0003（输入点冲突）
- 0004（中断中的非法指令）
- 000A（HSC 重复定义）

3.7.2 ~高速计数器

高速计数器指令（HSC）在 HSC 特殊存储器位状态的基础上，配置和控制高速计数器。参数 N 指定高速计数器的标号。

高速计数器可以被配置为 12 种模式中的任意一种。

每一个计数器都有时钟、方向控制、复位、启动的特定输入。对于双相计数器，两个时钟都可以运行在最高频率。在正交模式下，您可以选择一倍速（1x）或者四倍速（4x）计数速率。所有计数器都可以运行在最高频率下而互不影响。

使 ENO=0 的错误条件：

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

- 0001（在 HDEF 指令之前执行 HSC 指令）
- 0005（同时执行 HSC/PLS）

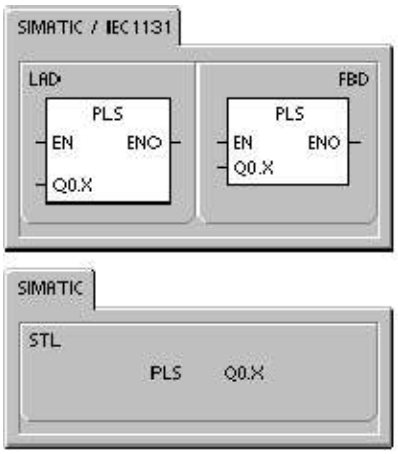
说明

无线 PLC 的 V1.0 版本的硬件不支持高速计数器指令。

无线 PLC 的 V1.0 版本的高速计数器记录的是硬件输入通道的脉冲计数值，HC0 代表的是通道 0 的脉冲计数，HC7 代表的是通过 7 的脉冲计数。脉冲计数器是无线 PLC 的基本功能之一，无论用户是否使用该功能，高速计数器都进行工作。

3.8~脉冲输出指令

脉冲输出指令（PLS）用于在高速输出（Q0.0 和 Q0.1）上控制脉冲串输出（PTO）和脉宽调制（PWM）功能。



改进的位控向导可以创建为您的应用程序定制的指令，这可以简化您的编程任务并充分利用无线 PLC CPU 的特有特性。

可以继续使用旧的 PLS 指令创建您自己的运动应用，但是只有改进的位控向导创建的指令才支持 PTO 上的线性斜坡。PTO 可以输出一串脉冲（占空比 50%），用户可以控制脉冲的周期和个数。

PWM 可以输出连续的、占空比可调的脉冲串，用户可以控制脉冲的周期和脉宽。

表 3-18 脉冲输出指令的有效操作数

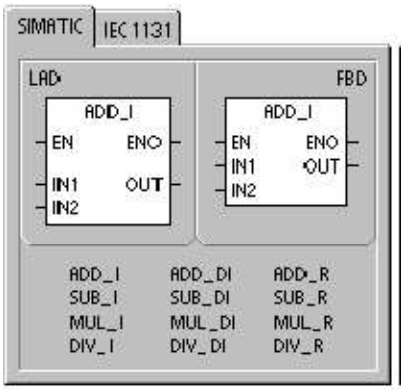
输入/输出	数据类型	操作数
Q0.X	WORD	常数：0（=Q0.0） 或者 1（=Q0.1）

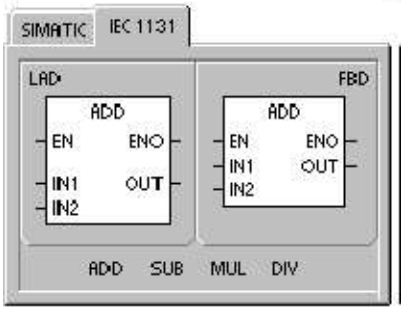
说明

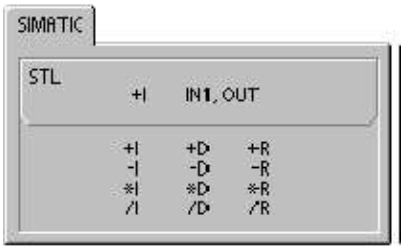
无线 PLC 的 V1.0 版本的硬件不支持脉冲输出指令的功能。

3.9 数字运算指令

3.9.1 加、减、乘、除指令







加法减法

IN1+IN2=OUT IN1--IN2=OUT

IN1+OUT=OUT OUT--IN1=OUT

整数加法（+I）或者整数减法（-I）指令，将两个 16 位整数相加或者相减，产生一个 16 位结果。双整数加法（+D）或者双整数减法（--D）指令，将两个 32 位整数相加或者相减，产生一个 32 位结果。实数加法（+R）和实数减法（--R）指令，将两个 32 位实数相加或相减，产生一个 32 位实数结果。

乘法除法

IN1*IN2=OUT IN1/IN2=OUT LAD

IN1*OUT=OUT OUT/IN1=OUT

整数乘法（*I）或者整数除法（/I）指令，将两个 16 位整数相乘或者相除，产生一个 16 位结果。（对于除法，余数不被保留。）双整数乘法（*D）或者双整数除法（/D）指令，将两个 32 位整数相乘或者相除，产生一个 32 位结果。（对于除法，余数不被保留。）实数乘法（*R）或实数除法（/R）指令，将两个 32 位实数相乘或相除，产生一个 32 位实数结果。

SM 标志位和 ENO

SM1.1 表示溢出错误和非法值。如果 SM1.1 置位，SM1.0 和 SM1.2 的状态不再有效而且原始输入操作数不会发生变化。如果 SM1.1 和 SM1.3 没有置位，那么数字运算产生一个有效的结果，同时 SM1.0 和 SM1.2 有效。在除法运算中，如果 SM1.3 置位，其它数学运算标志位不会发生变化。

使 ENO=0 的错误条件：

- SM1.1（溢出）
- SM1.3（被 0 除）
- 0006（间接寻址）

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

受影响的特殊存储器位：

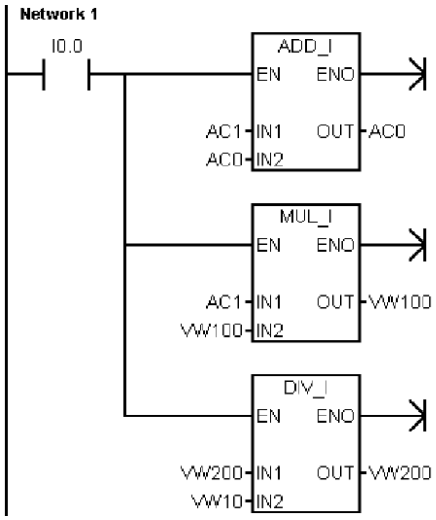
- SM1.0（结果为 0）
- SM1.1（溢出，运算过程中产生非法数值或者输入参数非法）
- SM1.2（结果为负）
- SM1.3（被 0 除）

表 3-19 加、减、乘、除指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	INT DINT REAL	IIW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、*LD、常数 ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数 ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	INT DINT、REAL	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、*VD、*AC、*LD ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

实数（或者浮点数）的表示格式采用 ANSI/IEEE 754--1985 标准（单精度）

示例：整数运算指令




```
Network 1
LD      I0.0
+I      AC1，AC0
*I      AC1，VW100
/I      VW10，VW200
```

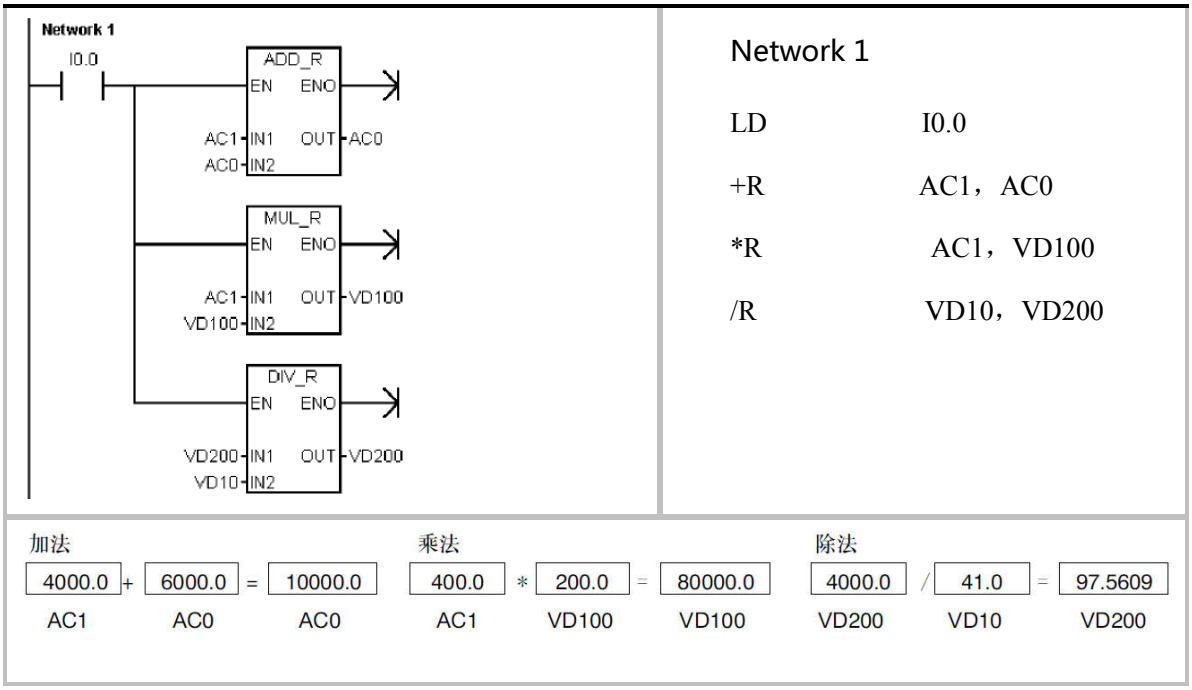
加法：40 (AC1) + 60 (AC0) = 100 (AC0)

乘法：40 (AC1) * 20 (VW100) = 800 (VW100)

除法：4000 (VW200) / 40 (VW10) = 100 (VW200)

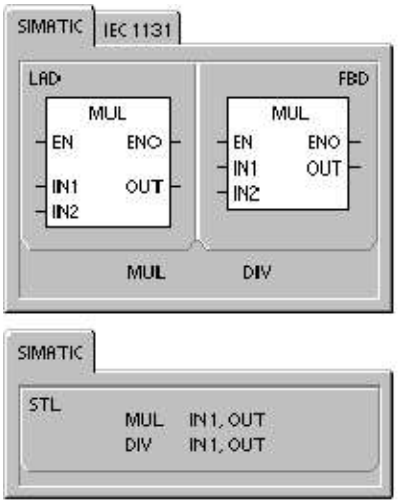
示例： 实数运算指令

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com



3.9.2 整数乘法产生双整数和带余数的整数除法

➢ 整数乘法产生双整数



IN1 * IN2 = OUT

IN1 * OUT = OUT

整数乘法产生双整数指令（MUL），将两个 16 位整数相乘，得到 32 位结果。在 STL 的 MUL 指令中，OUT 的低 16 位被用作一个乘数。

➢ 带余数的整数除法

IN1 / IN2 = OUT

OUT / IN1 = OUT

带余数的整数除法指令（DIV），将两个 16 位整数相除，得到 32 位结果。其中 16 位为余数（高 16 位字中），另外 16 位为商（低 16 位字中）。

在 STL 的 DIV 指令中，OUT 的低 16 位被用作除数。

➢ SM 标志位和 ENO

对于在本页中介绍的两条指令，特殊存储器（SM）标志位表示错误和非法值。如果在除法指令执行时，SM1.3（被 0 除）置位，其它数字运算标志位不会发生变化。否则，当数字运算完成时，

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

所有支持的数字运算状态位都包含有效状态。

使 ENO=0 的错误条件：

- SM1.1（溢出）
- SM1.3（被 0 除）
- 0006（间接寻址）

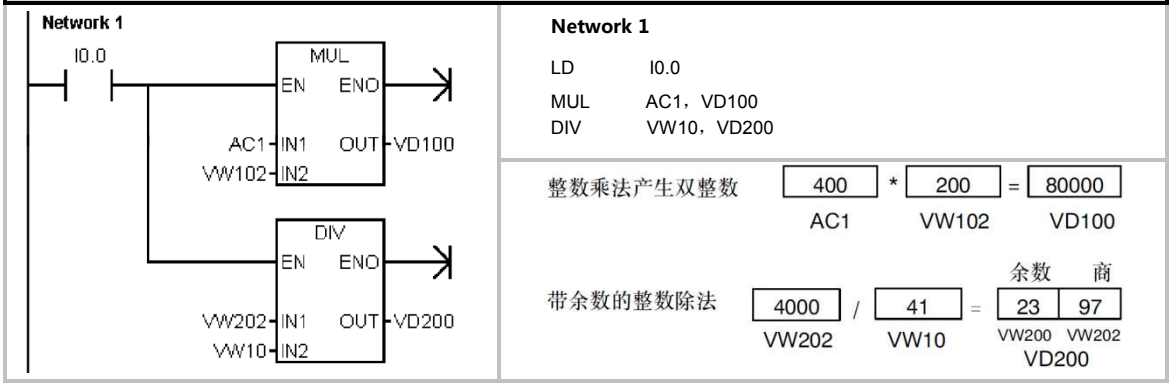
受影响的特殊存储器位：

- SM1.0（结果为 0）
- SM1.1（溢出）
- SM1.2（结果为负）
- SM1.3（被 0 除）

表3-20生双整数和带余数的整数除法指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
OUT	DINT	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

示例：整数乘法产生双整数和带余数的整数除法指令程序举例

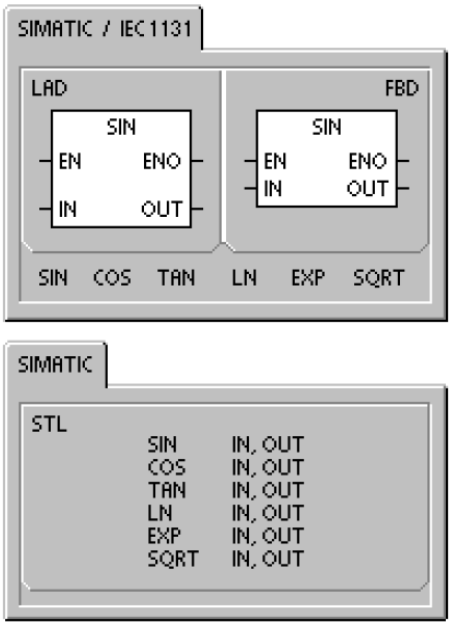


注意：VD100 包含：VW100 和 VW102，而 VD200 包含：VW200 和 VW202。

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

3.9.3 ~数学功能指令

➢ ~正弦、余弦和正切



正弦（SIN）、余弦（COS）和正切（TAN）指令计算角度值 IN 的三角函数值，并将结果存放在 OUT 中。输入角度值是弧度值。

SIN(IN)= OUTCOS(IN)= OUT　TAN(IN)= OUT

要将角度从度数变为弧度，可以使用 MULR（*R）指令，将度数乘以 1.745329E--2（接近 π /180）即可。

➢ ~自然对数和自然指数

自然对数指令（LN）计算输入值 IN 的自然对数，并将结果存放到 OUT 中。

自然指数指令（EXP）计算输入值 IN 的自然指数值，并将结果存放到 OUT 中。

LN （IN） = OUT EXP （IN）= OUT

要从自然对数计算出以 10 为底的对数值， 可以使用除法指令，将自然对数值除以 2.302585（接近 10 的自然对数）即可。

要计算任意实数的任意实数次方，包括分数形式的指数，需要将自然对数指令和自然指数指令结合在一起使用。例如：要计算 X 的 Y 次方，使用以下公式：EXP （Y * LN （X））。

平方根

平方根指令（SQRT）计算实数（IN）的平方根，并将结果存放到 OUT 中。

SQRT（IN）=OUT

如果要求其它次数的方根值：

5 的立方= 5^3 = EXP（3*LN（5）） = 125

125 的立方根= 125^（1/3） = EXP（（1/3）*LN（125））= 5

5 的立方的平方根= 5^（3/2） = EXP（3/2*LN（5）） = 11.18034

数学功能指令的 SM 位和 ENO

对于本页中描述的所有指令，SM1.1 用来表示溢出错误或者非法的数值。如果 SM1.1 置位，SM1.0 和 SM1.2 的状态不再有效而且原始输入操作数不会发生变化。如果 SM1.1 没有置位，那么数字运算

产生一个有效的结果，同时 SM1.0 和 SM1.2 状态有效。

使 ENO=0 的错误条件：

- SM1.1（溢出）
- 0006（间接寻址）

受影响的特殊存储器位：

- SM1.0（结果为 0）
- SM1.1（溢出）
- SM1.2（结果为负）

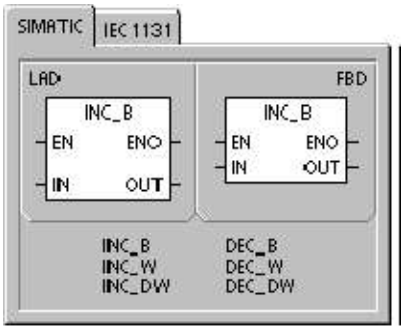
表 3-21 数学功能指令的有效操作数

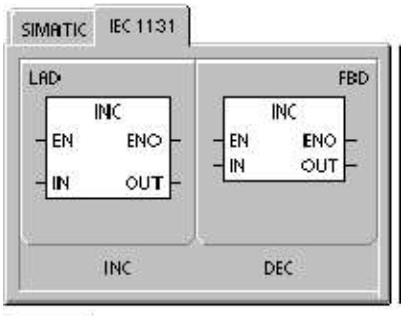
输入/输出	数据类型	操作数
IN	REAL	D、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、常数
OUT	REAL	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

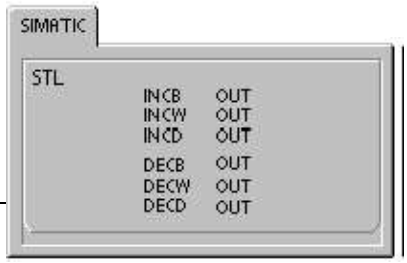
说明

无线 PLC 的 V1.0 版本的硬件不支持正弦、余弦和正切以及自然对数和自然指数指令的功能。

3.9.4 递增和递减指令







递增递减

IN+1=OUT IN - 1=OUT

OUT+1=OUT OUT - 1=OUT

递增或者递减指令将输入 IN 加 1 或者减 1，并将结果存放在 OUT 中。

字节递增（INCB）和字节递减（DECB）操作是无符号的。

字递增（INCW）和字递减（DECW）操作是有符号的。

双字递增（INCD）和双字递减（DECD）操作是有符号的。

使 ENO=0 的错误条件：

SM1.1（溢出）

0006（间接寻址）

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

受影响的特殊存储器位：

SM1.0（结果为 0）

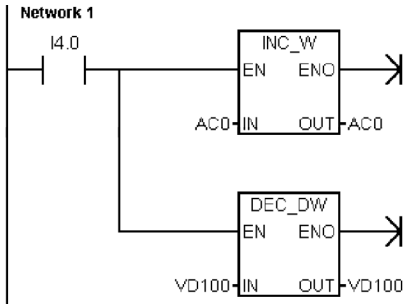
SM1.1（溢出）

SM1.2（结果为负）对于字和双字操作有效

表3-22递增和递减指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE INT DINT	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数 ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
OUT	BYTE INT DINT	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD IW、QW、VW、MW、SMW、SW、T、C、LW、AC,*VD、*AC、*LD、 ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

示例：递增和递减指令



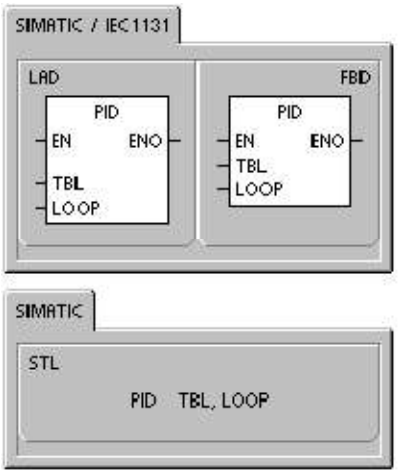

```
Network 1

LD      I4.0
INCW    AC0
DECD    VD100
```

字递增 125 + 1 = 126（AC0 / AC0）

双字递减 128000 - 1 = 127999（VD100 / VD100）

3.10 ~比例/积分/微分（PID）回路控制指令

PID 回路控制指令（PID）根据输入和表（TBL）中的配置信息，对相应的 LOOP 执行 PID 回路计算。



使 ENO=0 的错误条件：

SM1.1（溢出）

0006（间接寻址）

受影响的特殊存储器位：

SM1.1（溢出）

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

PID 回路指令（包含比例、积分、微分回路）可以用来进行 PID 运算。但是，可以进行这种 PID 运算的前提条件是逻辑堆栈栈顶（TOS）值必须为 1。该指令有两个操作数：TBL 和 LOOP。其中 TBL 是回路表的起始地址；LOOP 是回路号，可以是 0 到 7 的整数。

在程序中最多可以用 8 条 PID 指令。如果两个或两个以上的 PID 指令用了同一个回路号，那么即使这些指令的回路表不同，这些 PID 运算之间也会相互干涉，产生不可预料的结果。

回路表包含 9 个参数，用来控制和监视 PID 运算。这些参数分别是过程变量当前值（PVn），过程变量前值（PVn--1），给定值（SPn），输出值（Mn），增益（Kc），采样时间（Ts），积分时间（TI），微分时间（TD）和积分项前值（MX）。

为了让 PID 运算以预想的采样频率工作，PID 指令必须用在定时发生的中断程序中，或者用在主程序中被定时器所控制以一定频率执行。采样时间必须通过回路表输入到 PID 运算中。

表 3-23PID 回路控制指令的有效操作数

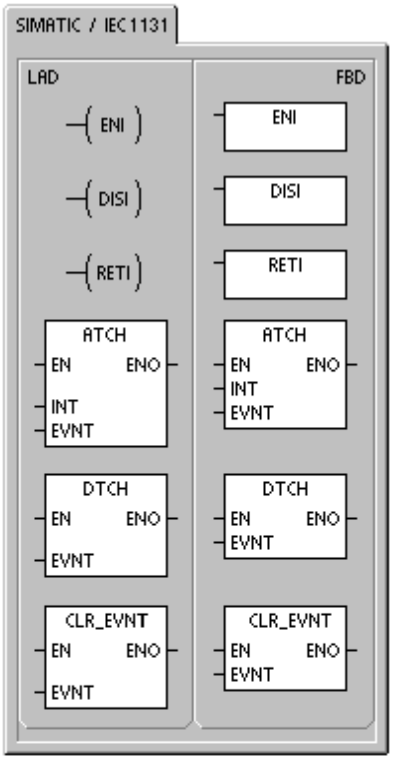
输入/输出	数据类型	操作数
TBL	BYTE	VB
LOOP	BYTE	常数（0到7）

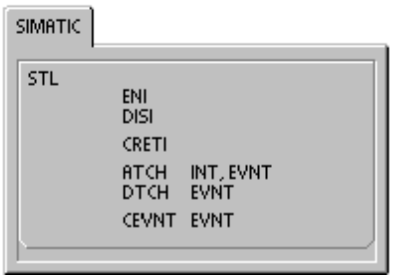
说明

无线 PLC 的 V1.0 版本的硬件不支持比例/积分/微分（PID）回路控制指令的功能。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.11 中断指令





中断允许和中断禁止

中断允许指令（ENI）全局地允许所有被连接的中断事件。中断禁止指令（DISI）全局地禁止处理所有中断事件。

当进入 RUN 模式时，初始状态为禁止中断。在 RUN 模式，您可以执行全局中断允许指令（ENI）允许所有中断。全局中断禁止指令（DISI）不允许处理中断服务程序，但中断事件仍然会排队等候。

使 ENO=0 的错误条件：

- 0004（试图在中断服务程序中执行 ENI、DISI 或者 HDEF 指令。）

中断条件返回

中断条件返回指令（CRETI）用于根据前面的逻辑操作的条件，从中断服务程序中返回。

中断连接

中断连接指令（ATCH）将中断事件 EVNT 与中断服务程序号 INT 相关联，并使能该中断事件。

使 ENO=0 的错误条件：

- 0002（与 HSC 的输入分配相冲突）

中断分离

中断分离指令（DTCH）将中断事件 EVNT 与中断服务程序之间的关联切断，并禁止该中断事件。

清除中断事件

清除中断事指令从中断队列中清除所有 EVNT 类型的中断事件。使用此指令从中断队列中清除不需要的中断事件。如果此指令用于清除假的中断事件，在从队列中清除事件之前要首先分离事件。否则，在执行清除事件指令之后，新的事件将被增加到队列中。

实例说明了处于正交模式的高速计数器如何使用 CLR_EVNT 指令清除中断事件。如果光电传感器正好处在从明亮过渡到黑暗的边界位置，那么在新的 PV 值装载之前，小的机械振动将生成实

际并不需要的中断。

表 3-24 中断指令的有效操作数

输入/输出	数据类型	操作数
INT	BYTE	常数（0到127）
EVNT	BYTE	常数 （0到37）

➢ 对中断连接和中断分离指令的理解

在激活一个中断程序前，必须在中断事件和该事件发生时希望执行的那段程序间建立一种联系。中断连接指令（ATCH）指定某中断事件（由中断事件号指定）所要调用的程序段（由中断程序号指定）。多个中断事件可调用同一个中断程序，但一个中断事件不能同时指定调用多个中断程序。

当把中断事件和中断程序连接时，自动允许中断。如果采用禁止全局中断指令不响应所有中断，每个中断事件进行排队，直到采用允许全局中断指令重新允许中断，如果不用允许全局中断指令，可能会使中断队伍溢出。

可以用中断分离指令（DTCH）截断中断事件和中断程序之间的联系，以单独禁止中断事件。中断分离指令（DTCH）使中断回到不激活或无效状态。

表 3-25 中断事件

事件号	中断描述	无线PLC-V1.0版本
0	uart1rx 串口0包接收完成	√
1	uart1tx串口0包发送完成	√
2	Smgrx短信包接收完成	√
3	smgtx 短信包发送完成	√
4	gprsrx GPRS包接收完成	√
5	gprstx GPRS包发送完成	√
6	uart2rx串口1包接收完成	√
7	uart2tx 串口1包发送完成	√
8	uart3rx 串口2包接收完成	×
9	uart3tx 串口2包发送完成	×
10	掉电事件	√
11	掉电恢复	√
12	定时中断0 SMB34	√
13	定时中断1 SMB35	√
14	定时器T32 CT=PT中断	√
15	定时器T96 CT=PT中断	√

➢ 理解中断服务程序的处理

执行中断服务程序用于响应与其相关的内部或者外部事件。一旦执行完中断服务程序的最后一条指令，控制权会回到主程序。您可以执行中断条件返回指令（CRETI）退出中断服务程序，下表对于在应用程序中使用中断服务程序给出了一些指导和限定。

表3-26使用中断服务程序的指导和限定

指导
中断处理提供了对特殊的内部或外部事件的响应。用户应当优化中断程序以执行一个特殊的任务，然后把控制返回主程序。 应当使中断程序短小而简单，执行时对其他处理也不要延时过长。如果做不到这些，意外的条件可能会引起由主程序控制的设备操作异常。对中断而言，其格言是"越短越好"。
限定
在中断程序中不能使用DISI、ENI、HDEF、LSCR和END指令。

系统对中断的支持

由于中断指令影响触点、线圈和累加器逻辑，所以系统保存和恢复逻辑堆栈、累加寄存器以及指示累加器和指令操作状态的特殊存储器标志位（SM）。这避免了进入中断程序或从中断程序返回对主用户程序造成破坏。

在主程序和中断程序间共享数据

您可以在主程序和一个或多个中断程序间共享数据。例如，用户主程序的某个地方可以为某个中断程序提供要用到的数据，反之亦然。如果用户程序共享数据，必须考虑中断事件异步特性的影响，这是因为中断事件会在用户主程序执行的任何地方出现。共享数据一致性问题的解决要依赖于主程序被中断事件中断时中断程序的操作。使用中断服务程序的局部变量表，这样可以保证中断服务程序只使用临时内存，而不会覆盖程序的其它地方使用的数据

这里有几种可以确保在用户主程序和中断程序间正确共享数据的编程技巧。这些技巧或限制共享存储器单元的访问方式，或让使用共享存储器单元的指令序列不会被中断。

- STL 程序共享单个变量：如果共享数据是单个字节、字、双字变量，而且用户程序用 STL 编写，那么通过把对共享数据操作得到的中间值只存储到非共享的存储器单元或累加器中，可以保证正确的共享访问。
- LAD 程序共享单个变量：如果共享数据是单个字节、字或双字变量，而且用户程序用梯形图编写，那么通过建立只用 Move 指令（MOVB、MOVW、MOVD、MOVR）访问共享存储器单元的约定，可以保证正确的共享访问。这些 Move 指令由执行时不受中断事件影响

的单条 STL 指令组成，而其它许多梯形图指令是由可被中断的 STL 指令序列组成的。

- STL 或 LAD 程序共享多个变量：如果共享数据由一些相关的字节、字或双字组成，那么可以用中断禁止/允许指令（DISI 和 ENI）来控制中断程序的执行。在用户程序开始对共享存储器单元操作的地方禁止中断，一旦所有影响共享存储器单元的操作完成后，再允许中断。在访问共享存储器单元期间，中断被禁止，中断程序不能执行，因而也无法访问共享存储器单元，但这种方法导致了对中断事件响应的延迟。

在中断服务程序中调用子程序

您可以在一个中断服务程序中调用一个子程序。中断服务程序与被调用的子程序共享累加器和逻辑堆栈。

➢ 支持的中断类型

支持下列类型的中断服务程序：

通讯口中断：串口、短信和 GPRS。

I/O 中断： 无线 PLC 的 V1.0 版本暂不支持 I/O 中断指令。

时基中断：产生使您的程序在指定的间隔上起作用的事件。

通讯口中断

PLC 的串行通讯口可由 LAD 或 STL 程序来控制。通讯口的这种操作模式称为自由端口模式。在自由端口模式下，用户可用程序定义波特率、每个字符位数和奇偶校验等。利用接收和发送中断可简化程序对通讯的控制。

PLC 的短信通信功能可以有 LAD 或 STL 程序来控制，当发生完成和接收完成短信数据后，会产生接收完成中断和接收完成中断，利用接收和发送完成中断可简化程序对短信的操作。

PLC 的 GPRS 通信功能可以有 LAD 或 STL 程序来控制，当发生完成和接收完成 GPRS 数据后，会产生接收完成中断和接收完成中断，利用接收和发送完成中断可简化程序对 GPRS 的操作。

I/O 中断

I/O 中断包含了上升沿或下降沿中断、高速计数器中断和脉冲串输出（PTO）中断。CPU 可用输入 I0.0 至 I0.3 的上升沿或下降沿产生中断。上升沿事件和下降沿事件可被这些输入点捕获。这些上升沿/下降沿事件可被用于指示当某个事件发生时必须引起注意的条件。

高速计数器中断允许响应诸如当前值等于预置值、相应于轴转动方向变化的计数方向改变和计数器外部复位等事件而产生的中断。每种高速计数器可对高速事件实时响应，而 PLC 扫描速率对这

些高速事件是不能控制的。

脉冲串输出中断给出了已完成指定脉冲数输出的指示。脉冲串输出的一个典型应用是步进电机。

可以通过将一个中断程序连接到相应的 I/O 事件上来允许上述的每一个中断。

说明

无线 PLC 的 V1.0 版本的硬件不支持 I/O 中断的功能。

时基中断

时基中断包括定时中断和定时器 T32/T96 中断。CPU 可以支持定时中断。可以用定时中断指定一个周期性的活动。周期以 1ms 为增量单位，周期时间可从 1ms 到 255ms。对定时中断 0，必须把周期时间写入 SMB34；对定时中断 1，必须把周期时间写入 SMB35。

每当定时器溢出时，定时中断事件把控制权交给相应的中断程序。通常可用定时中断以固定的时间间隔去控制模拟量输入的采样或者执行一个 PID 回路。

当把某个中断程序连接到一个定时中断事件上，如果该定时中断被允许，那就开始计时。在连接期间，系统捕捉周期时间值，因而后来的对 SMB34 和 SMB35 的更改不会影响周期。为改变周期时间，

首先必须修改周期时间值，然后重新把中断程序连接到定时中断事件上。当重新连接时，定时中断功能清除前一次连接时的任何累计值，并用新值重新开始计时。

一旦允许，定时中断就连续地运行，指定时间间隔的每次溢出时执行被连接的中断程序。如果退出 RUN 模式或分离定时中断，则定时中断被禁止。如果执行了全局中断禁止指令，定时中断事件会继续出现，每个出现的定时中断事件将进入中断队列（直到中断允许或队列满）。请参见定时中断的例子程序。

定时器 T32/T96 中断允许及时地响应一个给定的时间间隔。这些中断只支持 1ms 分辨率的延时接通定时器（TON）和延时断开定时器（TOF）T32 和 T96。T32 和 T96 定时器在其它方面工作正常。一旦中断允许，当有效定时器的当前值等于预置值时，在 CPU 的正常 1ms 定时刷新中，执行被连接的中断程序。首先把一个中断程序连接到 T32/T96 中断事件上，然后允许该中断。

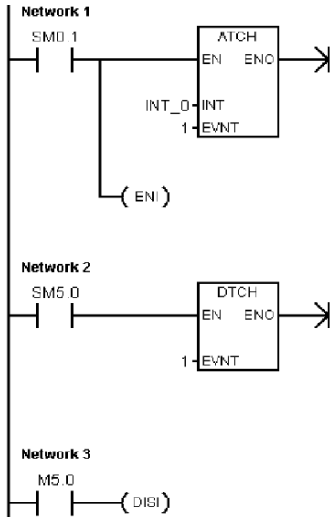
中断优先级和中断队列

无线 PLC 的中断无优先级，CPU 按先来先服务的原则处理中断。任何时间点上，只有一个用户中断程序正在执行。一旦中断程序开始执行，它要一直执行到结束。而且不会被别的中断程序，甚至是更高优先级的中断程序所打断。当另一个中断正在处理中，新出现的中断需要排队，等待处

理。

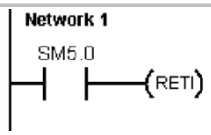
示例：中断指令

MAIN




```
Network 1   //首次扫描
              //1. 定义I0.0 的下降沿
              //中断服务程序为INT_0
              //2. 全局中断允许。
    LD          SM0.1
    ATCH        INT_0，1
    ENI

Network 2   //如果检测到I/O错误，
            //禁止I0.0的下降沿中断。
             //该程序段是可选的。
    LD          SM5.0
    DTCH        1

Network 3   //当M5.0接通时，禁止所有
            //中断。
    LD          M5.0
    DISI
```

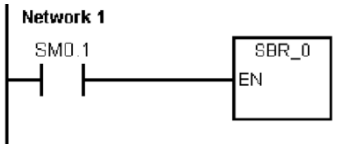
INT0




```
Network 1   //I0.0的下降沿中断服务
            //程序：
            //当有I/O错误时返回。
    LD          SM5.0
    CRETI
```

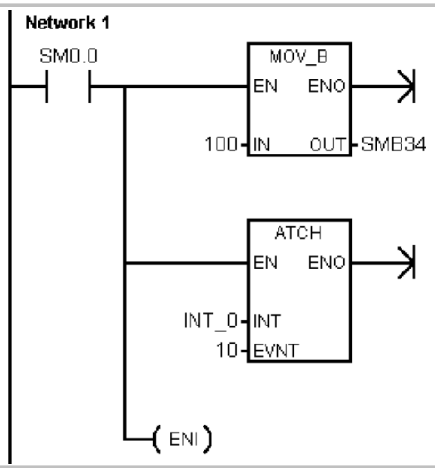
示例：用定时中断读取模拟量的数值程序举例

MAIN



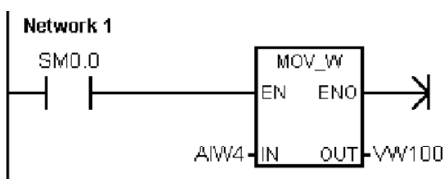

```
Network 1   //首次扫描，调用子程序0。
    LD          SM0.1
    CALL        SBR_0
```

SBR0




```
Network 1   //1. 设置定时中断的时间
            //间隔为100ms。
            //2. 连接INT_0到定时中
            //断0（事件10）。
            //3. 全局中断允许。
    LD          SM0.0
    MOVB        100，SMB34
    ATCH        INT_0，10
    ENI
```

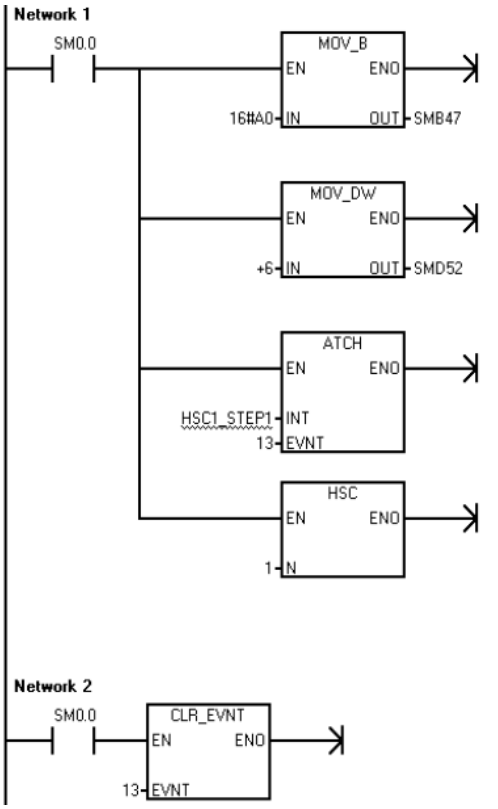
INT0




```
Network 1   //每100ms读AIW4的值
    LD          SM0.0
    MOVW        AIW4，VW100
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

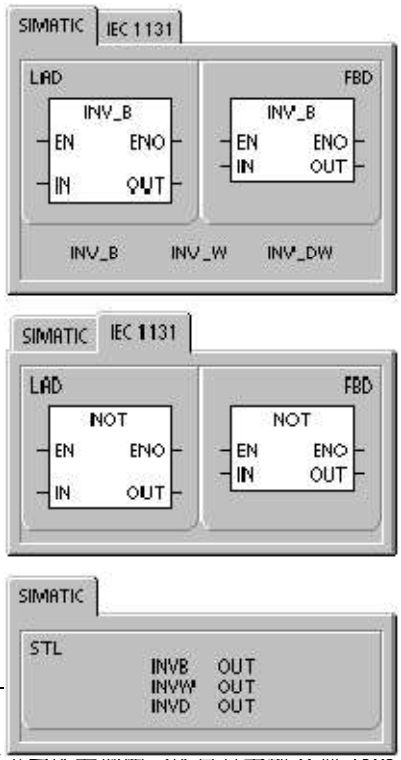
示例： 清除中断事件指令




```
Network 1            // 指令向导HSC

LD                   SM0.0
MOVB    16#A0，SMB47
                     //设置控制字节：
                     //写预设值;

MOVD    +6，SMD52    //PV = 6;

ATCH    HSC1_STEP1，13 //中断

HSC1_STEP1： CV = PV 用于HC1


Network 2    //清除由机械振动引起的
             //不需要的中断

LD                   SM0.0
CEVNT                13
```

3.12 逻辑操作指令

3.12.1 取反指令



➢ 字节、字和双字取反

字节取反（INVB）、字取反（INVW）和双字取反（INVD）指令将输入 IN 取反的结果存入 OUT 中。

使ENO=0的错误条件：

- 0006（间接寻址）

受影响的SM标志位：

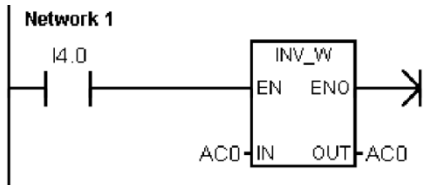
- SM1.0（结果为 0）

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区百万庄地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

表 3-27 取反指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
	WORD	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

示例：取反指令




```
Network1
  LD      I4.0
  INVW    AC0
```

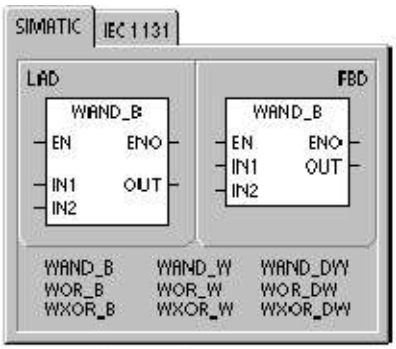
字取反 AC0 1101 0111 1001 0101

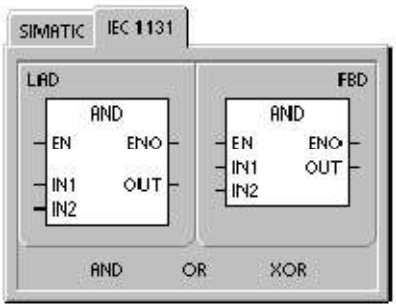
执行后

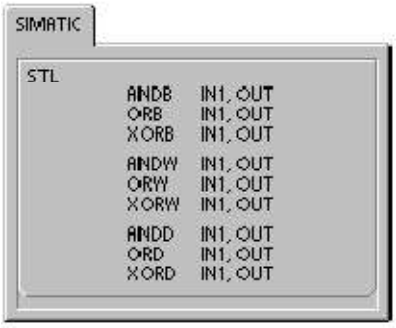
AC0 0010 1000 0110 1010

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.12.2 与、或和异或指令







➢ 字节与、字与和双字与

字节与（ANDB）、字与（ANDW）和双字与（ANDD）指令将输入值 IN1 和 IN2 的相应位进行与操作，将结果存入 OUT 中。

➢ 字节或、字或和双字或

字节或（ORB）、字或指令（ORW）和双字或（ORD）指令将两个输入值 IN1 和 IN2 的相应位进行或操作，将结果存入 OUT 中。

➢ 字节异或、字节或和双字异或

字节异或（XROB）、异或（XORW）和双字异或（ORD）指令将两个输入值 IN1 和 IN2 的相应位进行异或操作，将结果存入 OUT 中。

➢ SM 标志位和 ENO

对于本页中描述的所有指令，下列情况影响 SM 位和 ENO。

使ENO=0的错误条件：

- 0006（间接寻址）

受影响的SM标志位：

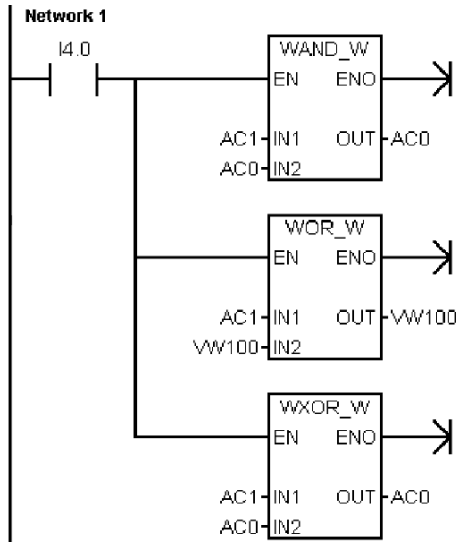
- SM1.0（结果为 0）

表 3-28 与、或和异或指令的有效操作数

输入/输出	数据类型	操作数
IN1、IN2	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
	WORD	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*AC、*LD
	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*AC、*LD
	DWORD	ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

示例：与、或和异或指令



Network1

```
LD      I4.0
ANDW    AC1 , AC0
ORW     AC1 , VW100
XORW    AC1 , AC0
```

字与

AC1 0001 1111 0110 1101

AND

AC0 1101 0011 1110 0110

等于

AC0 0001 0011 0110 0100

字或

AC1 0001 1111 0110 1101

或

VW100 1101 0011 1010 0000

等于

VW100 1101 1111 1110 1101

字异或

AC1 0001 1111 0110 1101

XOR

AC0 0001 0011 0110 0100

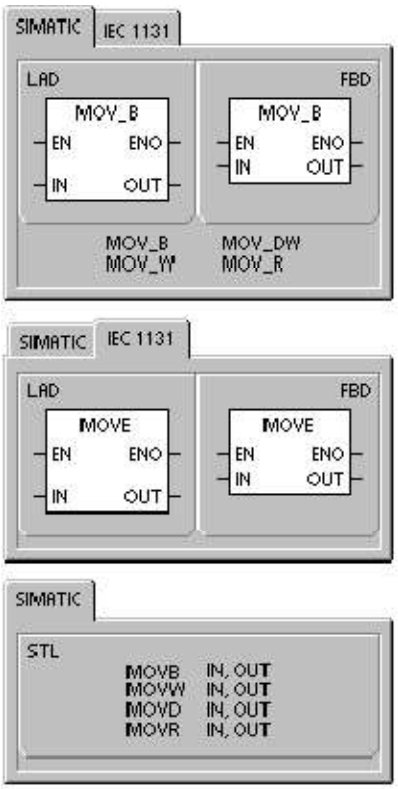
等于

AC0 0000 1100 0000 1001

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.13 传送指令

3.13.1 字节、字、双字或者实数传送



- 字节传送（MOVB）、字传送（MOVW）、双字传送

（MOVD）和实数传送指令在不改变原值的情况下将 IN 中的值传送到 OUT。

使用双字传送指令可以创建一个指针。。

对于 IEC 传送指令，输入和输出的数据类型可以不同，但数据长度必须相同。

使ENO=0的错误条件：

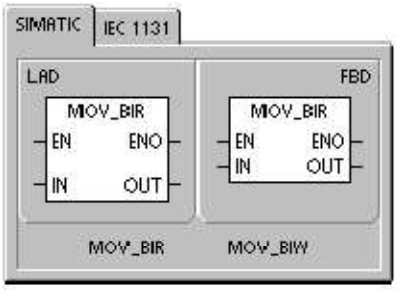
- 0006（间接寻址）

表 3-29 传送指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE WORD、INT DWORD、DINT REAL	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、*AC、*LD、常数 ID、QD、VD、MD、SMD、SD、LD、HC、&VB、&IB、&QB、&MB、&SB、&T、&C、&SMB、&AIW、&AQW、AC、*VD、 *LD、*AC、Constant ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC、 常数
OUT	BYTE WORD、INT DWORD、 DINT 、 REAL	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.13.2 字节立即传送（读和写）



字节立即传送指令允许您在物理 I/O 和存储器之间立即传送一个字节数据。

字节立即读（BIR）指令读物理输入（IN），并将结果存入内存地址（OUT），但过程映像寄存器并不刷新。

字节立即写指令（BIW）从内存地址（IN）中读取数据，写入物理输出（OUT），同时刷新相应的过程映像区。

使ENO=0的错误条件：

- 0006（间接寻址）
- 不能访问扩展模块

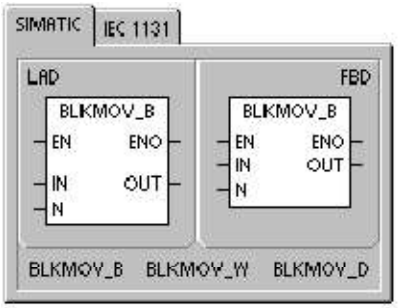
表 3-30 字节立即读指令的有效操作数

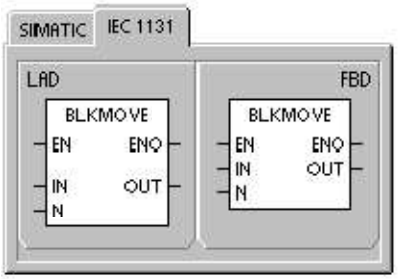
输入/输出	数据类型	操作数
IN	BYTE	iB、*VD、*LD、*AC
OUT	BYTE	iB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

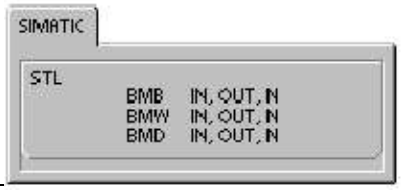
表3-31　字节立即写指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	BYTE	QB、*VD、*LD、*AC

3.13.3 块传送指令







➢ 字节、字、双字的块传送

字节块传送（BMB）、字块传送（BMW）和双字块传送（BMD）指令传送指定数量的数据到一个新的存储区，数据的起始地址 IN，数据长度为 N 个字节、字或者双字，新块的起始地址为 OUT。

N 的范围从 1 到 255。

使ENO=0的错误条件：

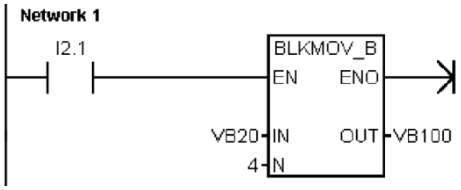
- 0006（间接寻址）
- 0091（操作数超出范围）

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

表 3-32 表传送指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	B、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC
	WORD、INT	IW、QW、VW、SMW、SW、T、C、LW、AIW、*VD、*LD、*AC
	DWORD、DINT	ID、QD、VD、MD、SMD、SD、LD、*VD、*LD、*AC
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、*VD、*LD、*AC
	WORD、INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AQW、*VD、*LD、*AC
	DWORD、DINT	ID、QD、VD、MD、SMD、SD、LD、*VD、*LD、*AC
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、常数、*VD、*LD、*AC

示例：块传送指令

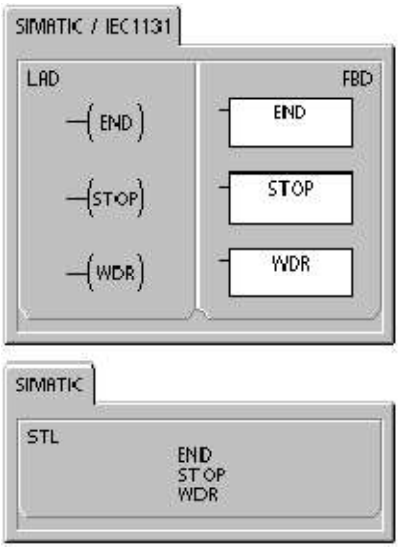



```
Network1        //将数组1（VB20到VB23）
                //传送至数组2（VB100到VB103）
    LD          12.1
    BMB         VB20，VB100，4
```

	VB20	VB21	VB22	VB23
数组1	30	31	32	33
数组2	VB100	VB101	VB102	VB103
	30	31	32	33

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.14 程序控制指令



3.14.1 条件结束

条件结束指令（END）根据前面的逻辑关系终止当前扫描周期。可以在主程序中使用条件结束指令，但不能在子程序或中断服务程序中使用该命令。

3.14.2 停止

停止指令（STOP）导致 CPU 从 RUN 到 STOP 模式从而可以立即终止程序的执行。

如果 STOP 指令在中断程序中执行，那么该中断立即终止，并且忽略所有挂起的中断，继续扫描程序的剩余部分。完成当前周期的剩余动作，包括主用户程序的执行，并在当前扫描的最后，完成从 RUN 到 STOP 模式的转变。

3.14.3 看门狗复位

看门狗复位指令（WDR）允许 S7--200 CPU 的系统看门狗定时器被重新触发，这样可以在不引起看门狗错误的情况下，增加此扫描所允许的时间。

使用 WDR 指令时要小心，因为如果您用循环指令去阻止扫描完成或过度的延迟扫描完成的时间，那么在终止本次扫描之前，下列操作过程将被禁止：

- I/O 更新（立即 I/O 除外）
- 强制更新
- SM 位更新（SM0，SM5～SM29 不能被更新）
- 运行时间诊断
- 由于扫描时间超过 25 秒，10ms 和 100ms 定时器将不会正确累计时间。
- 在中断程序中的 STOP 指令
- 带数字量输出的扩展模块也包含一个看门狗定时器，如果模块没有被无线 PLC 写，则此看门狗定时器将关断输出。在扩展的扫描时间内，对每个带数字量输出的扩展模块进行立即写操作，以保持正确的输出。请参考这段描述之后的实例。

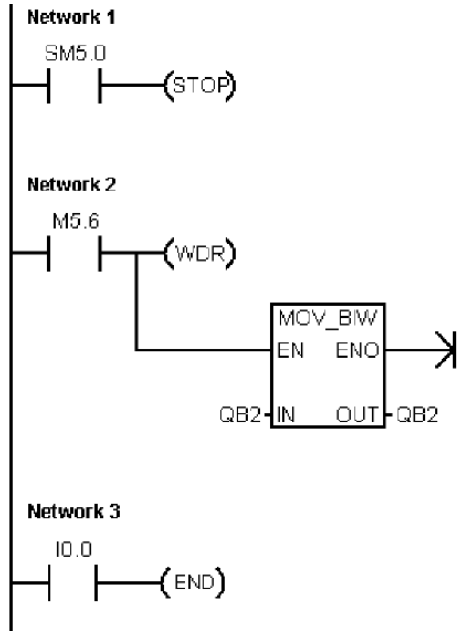
提示

如果希望程序的扫描周期超过 500ms，或者在中断事件发生时有可能使程序的扫描周期超过 500ms 时，您应该使用看门狗复位指令来重新触发看门狗定时器。

每次使用看门狗复位指令，您应该对每个扩展模块的某一个输出字节使用一个立即写指令来复位每个扩展模块的看门狗。

如果您使用了看门狗复位指令允许程序的执行有一个很长的扫描时间，此时将无线 PLC 的模式开关切换到 STOP 位置，则在 1.4 秒内，CPU 转到 STOP 方式。

示例：停止、结束和看门狗复位指令程序举例



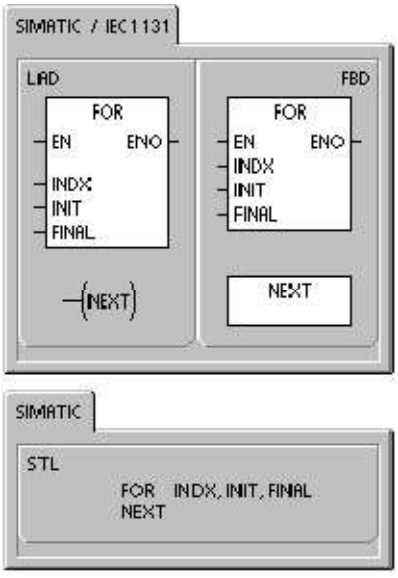

```
Network 1          //当检测到I/O错误时，
                   //强制切换到STOP模式。
LD                 SM5.0
STOP

Network2           //当M5.6接通时，允许扫描、
                   //周期扩展：
                   //1. 重新触发CPU的看门狗。
          //2. 重新触发第一个输出模块的看门狗
LD                 M5.6
WDR
BIW                QB2，QB2

Network 3
//当I0.0接通时，终止当前扫描周期。
LD                 I0.0
END
```

3.14.4 For--Next 循环指令

FOR 和 NEXT 指令可以描述需重复进行一定次数的循环体。每条 FOR 指令必须对应一条 NEXT 指令。For--Next 循环嵌套（一个 For--Next 循环在另一个 For--Next 循环之内）深度可达 8 层。

FOR--NEXT 指令执行 FOR 指令和 NEXT 指令之间的指令。必须指定计数值或者当前循环次数 INDX、初始值（INIT）和终止值（FINAL）。

NEXT 指令标志着 FOR 循环的结束。



使 ENO=0 的错误条件：

- 0006（间接寻址）

如果允许 FOR/NEXT 循环，除非在循环内部修改了终值，循环体就一直循环执行直到循环结束。

当 FOR/NEXT 循环执行的过程中可以修改这些值。当循环再次允许时，它把初始值拷贝到 INDX 中（当前循环次数）。

当下一次允许时，FOR/NEXT 指令复位它自己。例如，给定初值（INIT）为 1，终值（FINAL）为 10，那么随着当前计数值（INDX）从 1 增加到 10，FOR 与 NEXT 之间的指令被执行 10 次：1，2，3，...10.

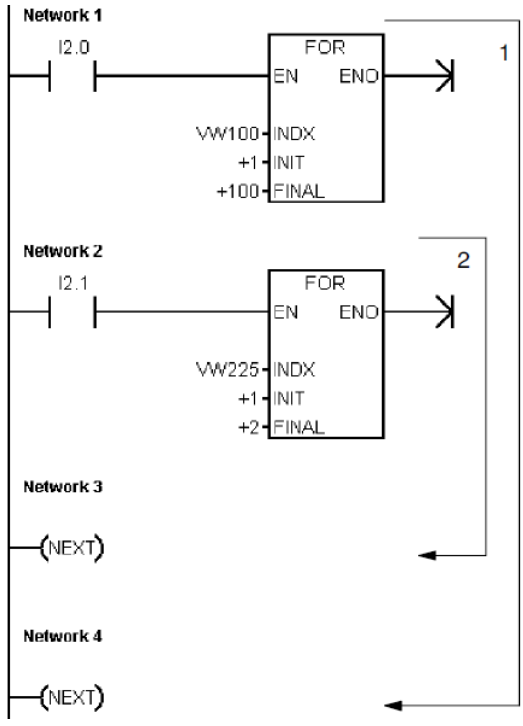
如果初值大于终值，那么循环体不被执行。每执行一次循环体，当前计数值增加 1，并且将其结果同终值作比较，如果大于终值，那么终止循环。

如果程序进入 FOR--NEXT 循环时，栈顶值为 1，则当程序退出 FOR--NEXT 循环时，栈顶值也将为 1。

表 3-33FOR--NEXT 指令的有效操作数

输入/输出	数据类型	操作数
INDX	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、 *AC
INIT、FINAL	INT	VW、IW、QW、MW、SMW、SW、T、C、LW、AC、AIW、*VD、 *AC、常数

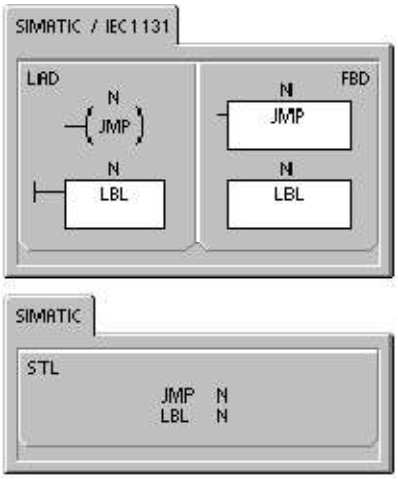
示例：For- Next循环指令




```
Network 1          //当I2.0接通时，外循环（标识
                   //1）执行100次。
LD                 2.0
F                  或 VW100，+1，+100
Network2           //当I2.1
                   //接通时，外循环
                   //每执行一次，内循环执行两次。
LD                 I2.1
F或                VW225，+1，+2
Network 3          //回路2结束。
NEXT
Network 4          //回路1结束。
NEXT
```


北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.14.5 跳转指令



跳转到标号指令（JMP）执行程序内标号 N 指定的程序分支。

标号指令标记跳转目的地的位置 N。

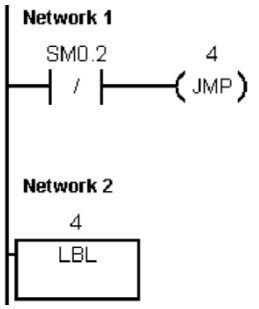
您可以在主程序、子程序或者中断服务程序中，使用跳转指令。跳转和与之相应的标号指令必须位于同一段程序代码（无论是主程序、子程序还是中断服务程序）。

不能从主程序跳到子程序或中断程序，同样不能从子程序或中断程序跳出。

可以在 SCR 程序段中使用跳转指令，但相应的标号指令必须也在同一个 SCR 段中。

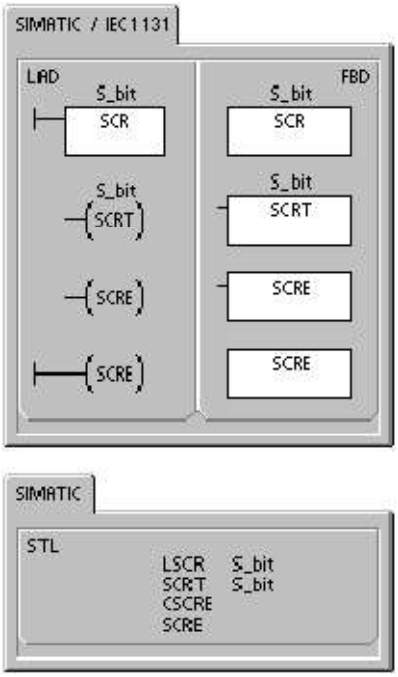
表 3-34 跳转指令的有效操作数

输入/输出	数据类型	操作数
N	WORD	常数（0到255）

示例：跳转到标号指令	
	**Network 1** //如果掉电保持的数据没有丢失，跳 //转到LBL4。 LDN SM0.2 JMP 4 **Network 2** LBL 4

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.14.6 顺控继电器（SCR）指令



SCR 指令使您能够按照自然工艺段在 LAD、FBD 或 STL 中编制状态控制程序。

只要您的应用中包含的一系列操作需要反复执行，就可以使用 SCR 使程序更加结构化，以至于直接针对应用。这样可以使得编程和调试更加快速和简单。

装载 SCR 指令（LSCR）将 S 位的值装载到 SCR 和逻辑堆栈中。

SCR 堆栈的结果值决定是否执行 SCR 程序段。SCR 堆栈的值会被复制到逻辑堆栈中，因此可以直接将盒或者输出线圈连接到左侧的能流线上而不经过中间触点。

限定

当使用 SCR 时，请注意下面的限定：

- 不能把同一个 S 位用于不同程序中。例如：如果在主程序中用了 S0.1，在子程序中就不能再使用它。
- 在 SCR 段之间不能使用 JMP 和 LBL 指令，就是说不允许跳入、跳出。可以在 SCR 段附近使用跳转和标号指令或者在段内跳转。
- 在 SCR 段中不能使用 END 指令。

表 3-35 顺控继电器指令的有效操作数

输入/输出	数据类型	操作数
S_bit	BOOL	S

下图给出了 S 堆栈和逻辑堆栈以及执行 LSCR 指令产生的影响。以下是对顺控继电器指令的正确理解：

装载 SCR 指令（LSCR）标志着 SCR 段的开始，SCR 结束指令则标志着 SCR 段的结束。在装载 SCR 指令与 SCR 结束指令之间的所有逻辑操作的执行取决于 S 堆栈的值。而在 SCR 结束指令和下一条装载 SCR 指令之间的逻辑操作则不依赖于 S 堆栈的值。

SCR 传输指令（SCRT）将程序控制权从一个激活的 SCR 段传递到另一个 SCR 段。

执行 SCRT 指令可以使当前激活的程序段的 S 位复位，同时使下一个将要执行的程序段的 S 位

置位。在 SCRT 指令指行时，复位当前激活的程序段的 S 位并不会影响 S 堆栈。SCR 段会一直保持能流直到退出。

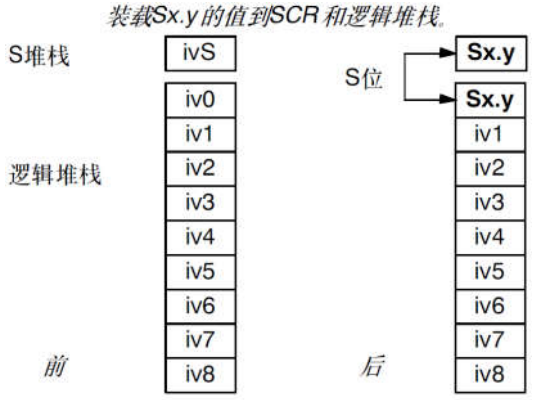


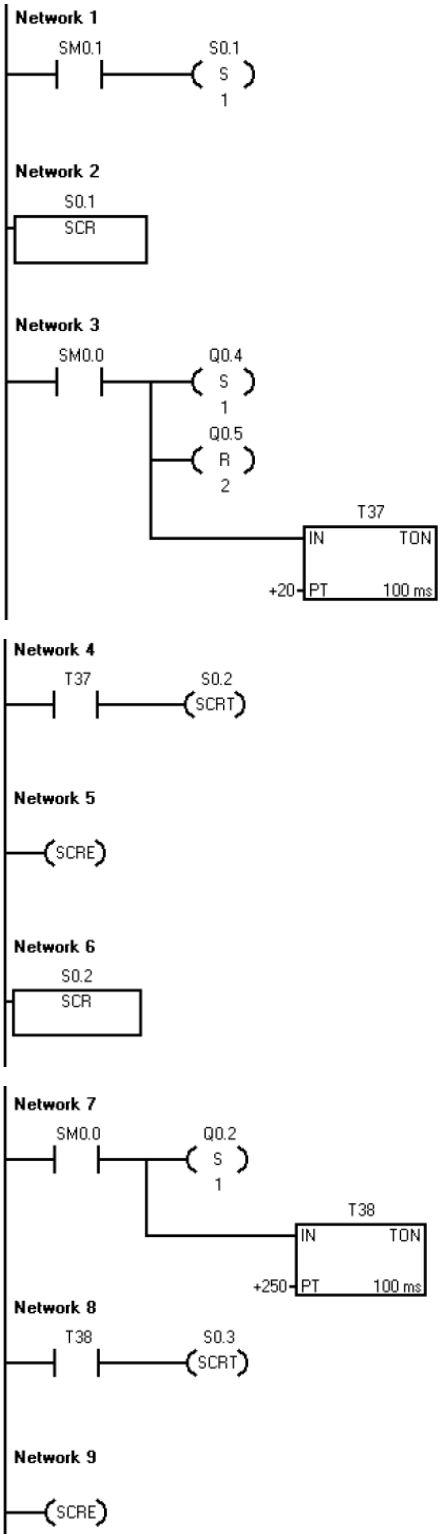
图 3-16LSCR 对逻辑堆栈的影响

SCR 条件结束指令（CSCRE）可以使程序退出一个激活的程序段而不执行 CSCRE 与 SCRE 之间的指令。CSCRE 指令不影响任何 S 位，也不影响 S 堆栈。

在以下实例中，首次扫描位 SM0.1 置位 S0.1，从而在首次扫描中，激活状态 1。延时 2 秒后，T37 导致切换到状态 2。切换使状态 1 停止，激活状态 2。

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

示例：顺控继电器指令



Network 1 //在首次扫描使能状态1。

LD SM0.1

S S0.1，1

Network 2 //状态1控制开始。

LSCR S0.1

Network 3 //控制第一条街的信号：

//1.置位：接通红灯。
//2.复位：关断黄灯和绿灯。
//3.启动2秒定时器。

LD SM0.0

S Q0.4，1
R Q0.5，2
TON T37，+20

Network 4 //延时2秒后，切换到状态2。

LD T37
SCRT S0.2

Network 5 //状态1的SCR区结束。

SCRE

Network 6 //状态2的控制区开始。

LSCR S0.2

Network 7 //控制第二条街的信号：

//1. 置位： 接通绿灯。
//2. 启动25秒定时器。

LD SM0.0

S Q0.2，1
TON T38，+250

Network 8 //延时25秒后，切换到状态3。

LD T38

SCRT S0.3

Network 9 //状态2的SCR区结束。

SCRE

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

➢ 分支控制

在许多实例中，一个顺序控制状态流必须分成两个或多个不同分支控制状态流。当一个控制状态流分离成多个分支时，所有的分支控制状态流必须同时激活，如下图所示：

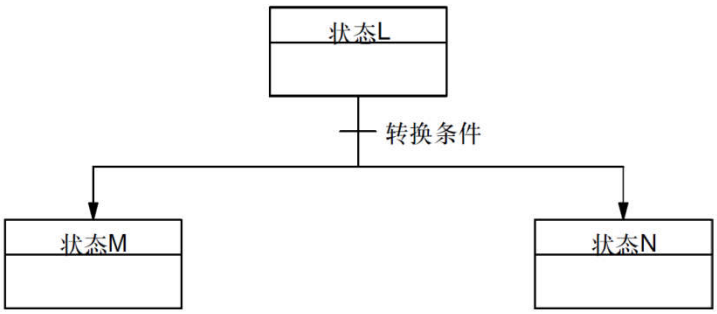
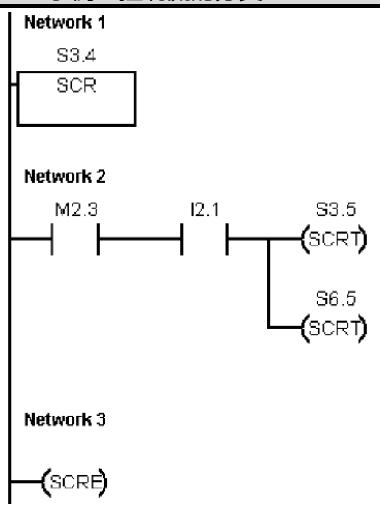


图 3-17 控制流的分支

使用多条由相同转移条件激活的 SCRT 指令，可以在一段 SCR 程序中实现控制流的分支，如下面的实例所示。

示例：控制流的分支




```
Network 1          //状态L控制区开始
LSCR               S3.4

Network 2
LD                 M2.3
A                  I2.1
SCRT               S3.5      //切换到状态M
SCRT S6.5      //切换到状态N

Network 3                    //状态L的状态区结束
SCRE
```

➢ 合并控制

与分支控制的情况类似，两个或者多个分支状态流必须合并为一个状态流。当多个状态流汇集成一个时，我们称之为合并。当控制流合并时，所有的控制流必须都完成，才能执行下一个状态。下图给出了两个控制流合并的示意图。

在 SCR 程序中，通过从状态 L 转到状态 L’，以及从状态 M 转到状态 M’的方法实现控制流的合并。当状态 L’、M’的 SCR 使能位为真时，即可激活状态 N，如下例所示：

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

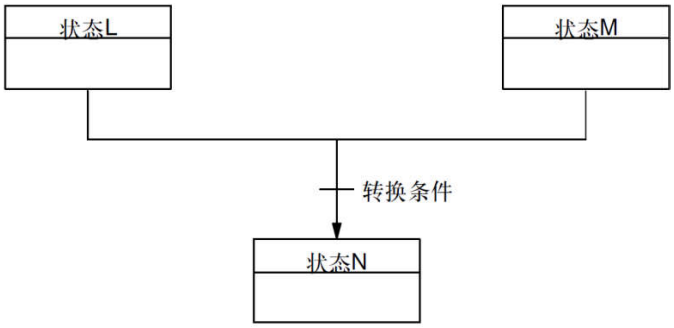
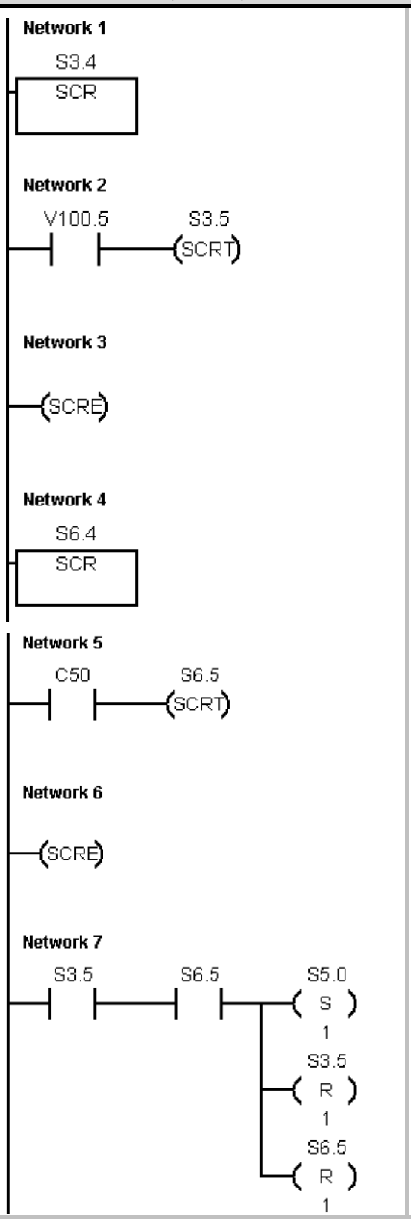


图 3-18 控制流的合并

示例：控制流的合并




```
Network 1      //状态L控制区开始

LSCR           S3.4

Network2       //切换到状态L
LD             V100.5
SCRT           S3.5

Network 3      //状态L SCR区的结束

SCRE

Network 4      //状态M控制区开始

LSCR       S6.4

Network5       //切换到状态M

LD             C50

SCRT           S6.5

Network 6      //状态M SCR区的结束

SCRE

Network7       //当状态L和M同时激活时：

//1. 使能状态N（S5.0）
//2. 复位状态L（S3.5）
//3. 复位状态M（S6.5）
LD             S3.5
A              S6.5
S              S5.0，1
R              S3.5，1
R              S6.5，1
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

在有些情况下，一个控制流可能转入多个可能的控制流中的某一个。到底进入哪一个，取决于控制流前面的转移条件，哪一个首先为真，如下图所示。

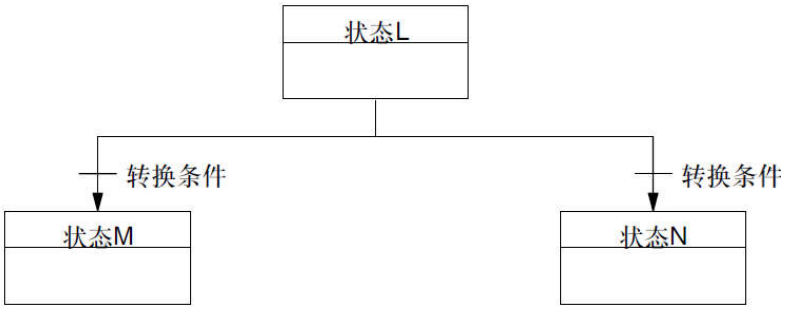
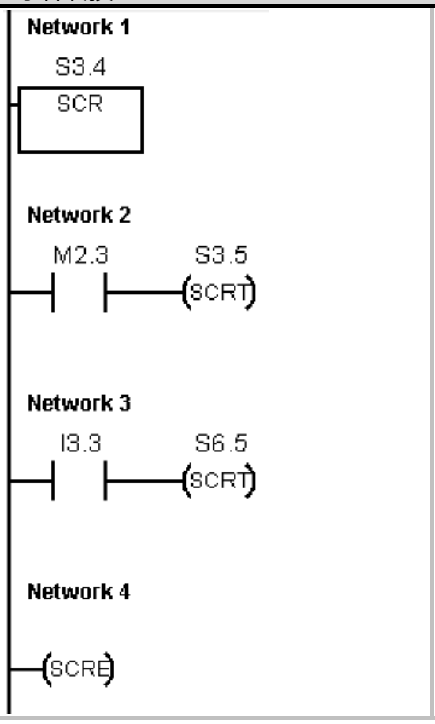


图3-19 条件转换控制流分支

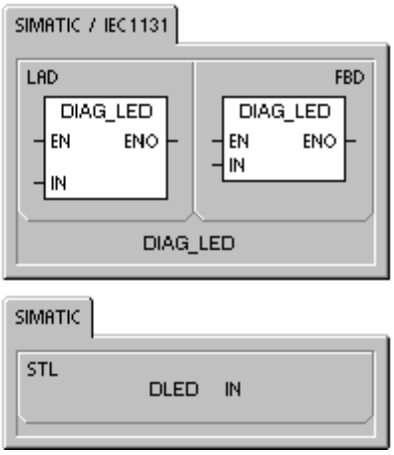
示例：条件转换




```
Network 1        //状态L控制区开始
LSCR             S3.4
Network 2        //切换到M状态
LD               M2.3
SCRT             S3.5
Network 3        //切换到N状态
LD               I3.3
SCRT             S6.5
Network 4        //状态L的SCR区结束
SCRE
```


北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.14.7 ~诊断 LED 指令



如果输入参数 IN 的值为零，就将诊断 LED 置为 OFF。如果输入参数 IN 的值大于零，就将诊断 LED 置为 ON（黄色）。

当系统块中指定的条件为真或者用非零 IN 参数执行 DIAG_LED 指令时，CPU 发光二极管（LED）标注的 SF/DIAG 可以被配置用于显示黄色。

系统块（配置 LED）复选框选项：

- 当有一项在 CPU 内被强制时，SF/DIAGLED 为 ON（黄色）
- 当模块有 I/O 错误时，SF/DIAGLED 为 ON（黄色）

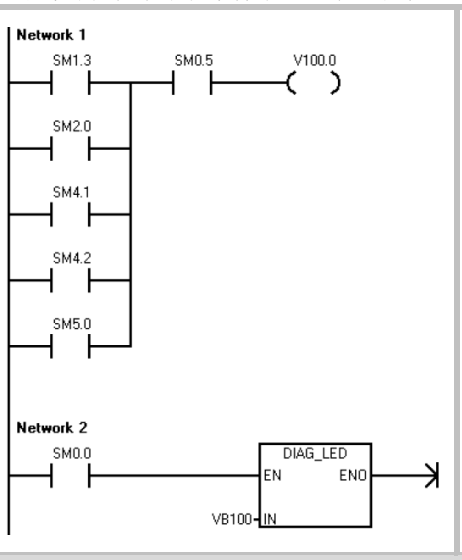
两个配置 LED 选项都不选中，将使 SF/DIAG 黄光只受 DIAG_LED 指令控制。CPU 系统故障（SF）用红光指示。

表 3-36 诊断 LED 指令的有效操作数

输入/输出	数据类型	操作数
IN	BYTE	VB、IB、QB、MB、SB、SMB、LB、AC、常数、 * VD、 * LD、 * AC

实例1诊断LED指令

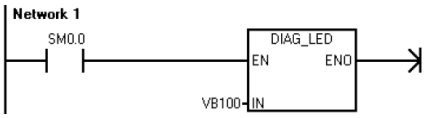
当检测到错误时，诊断LED闪烁。
只要检测到5个错误条件中的一个，诊断LED就闪烁。




```
Network1
LD      SM1.3
O       SM 2.0
O       SM4.1
O       SM4.2
O       SM5.0
A       SM0.5
=       V100.0
```

实例2诊断LED指令

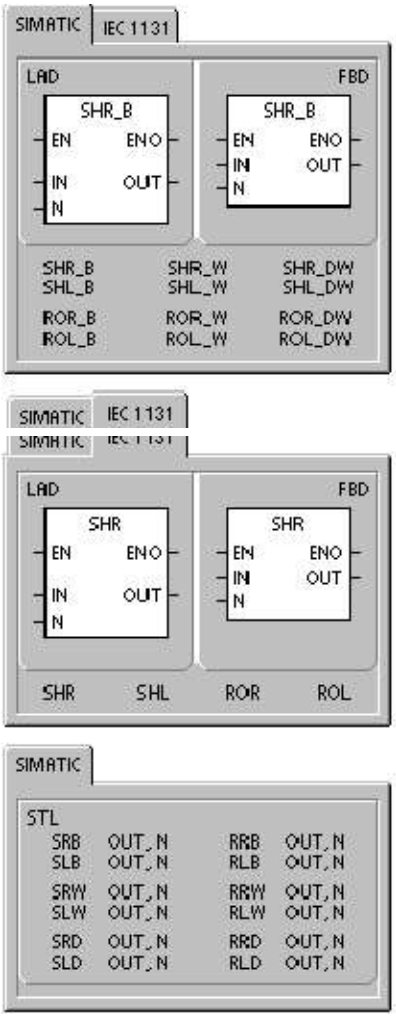
北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

当错误消失时，接通诊断LED。 当有错误代码在VB100中报告时，接通诊断LED	
	**Network1** LD SM0.0 DLED VB100

说明

无线 PLC 的 V1.0 版本的硬件不支持诊断 LED 指令的功能

3.15 移位和循环指令



3.15.1 右移和左移指令

移位指令将输入值 IN 右移或左移 N 位，并将结果装载到输出 OUT 中。

移位指令对移出的位自动补零。如果位数 N 大于或等于最大允许值（对于字节操作为 8，对于字操作为 16，对于双字操作为 32），那么移位操作的次数为最大允许值。如果移位次数大于 0，溢出标志位（SM1.1）上就是最近移出的位值。如果移位操作的结果为 0，零存储器位（SM1.0）置位。

字节操作是无符号的。对于字和双字操作，当使用有符号数据类型时，符号位也被移动。

使 ENO=0 的错误条件：

- 0006（间接寻址）

受影响的 SM 标志位：

- SM1.0（结果为 0）
- SM1.1（溢出）

3.15.2 循环右移和循环左移指令

循环移位指令将输入值 IN 循环右移或者循环左移 N 位，并将输出结果装载到 OUT 中。循环移位是圆形的。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

如果位数 N 大于或者等于最大允许值（对于字节操作为 8，对于字操作为 16，对于双字操作为 32），无线 PLC 在执行循环移位之前，会执行取模操作，得到一个有效的移位次数。移位位数的取模操作的结果，对于字节操作是 0 到 7，对于字操作是 0 到 15，而对于双字操作是 0 到 31。

如果移位次数为 0，循环移位指令不执行。如果循环移位指令执行，最后一个移位的值会复制到溢出标志位（SM1.1）。

如果移位次数不是 8（对于字节操作）、16（对于字操作）和 32（对于双字操作）的整数倍，最后被移出的位会被复制到溢出标志位（SM1.1）。当要被循环移位的值是零时，零标志位（SM1.0）被置位。

字节操作是无符号的。对于字和双字操作，当使用有符号数据类型时，符号位也被移位。

使 ENO=0 的错误条件：

- 0006（间接寻址）

受影响的 SM 标志位：

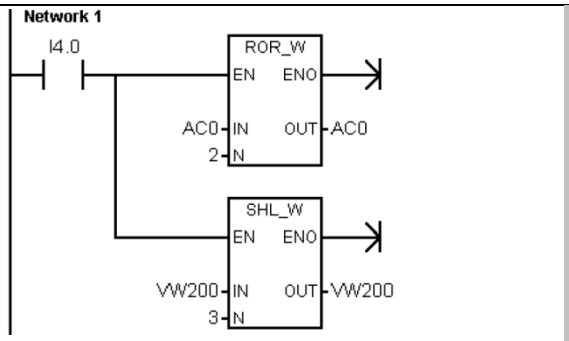
- SM1.0（结果为 0）
- SM1.1（溢出）

表 3-37 移位和循环移位指令的有效操作数

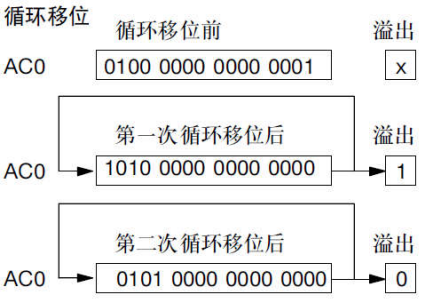
输入/输出	数据类型	操作数
IN	BYTE WORD DWORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数 IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、 *AC、常数 ID、QD、VD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC、常数
OUT	BYTE WORD DWORD	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC ID、QD、VD、MD、SMD、SD、LD、AC、*VD、*LD、*AC
N	BYTE	B、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数

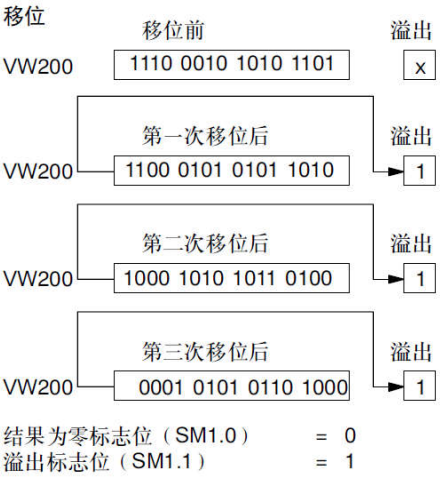
示例：移位和循环指令

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com



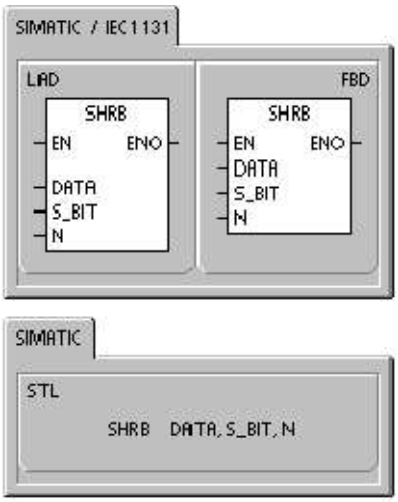

```
Network1

LD      I4.0
RRW     AC0 , 2
SLW     VW200 , 3
```





北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

3.15.3 移位寄存器指令



移位寄存器指令将一个数值移入移位寄存器中。移位寄存器指令提供了一种排列和控制产品流或者数据的简单方法。使用该指令，每个扫描周期，整个移位寄存器移动一位。

移位寄存器指令把输入的 DATA 数值移入移位寄存器。其中，S_BIT 指定移位寄存器的最低位，N 指定移位寄存器的长度和移位方向（正向移位=N，反向移位=-N）。

SHRB 指令移出的每一位都被放入溢出标志位（SM1.1）。

这条指令的执行取决于最低有效位（S_BIT）和由长度（N）指定的位数。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 0092（计数区错误）

受影响的 SM 标志位：

- SM1.1（溢出）

表 3-38 移位寄存器指令的有效操作数

输入/输出	数据类型	操作数
DATA、S_BIT	BOOL	I、Q、V、M、SM、S、T、C、L
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数

位移位寄存器的最高位（MSB.b）可通过下面公式计算求得：

MSB.b = [（S_BIT 的字节号） + （[N] -- 1 + （S_BIT 的位号）） / 8].[除 8 的余数]

例如：如果 S_BIT 是 V33.4，N 是 14，那么 MSB.b 是 V35.1，或：

MSB.b =V33+（[14]--1+4）/8

=V33+17/8

=V33+2（余数为 1）

=V35.1

当反向移动时，N 为负值，输入数据从最高位移入，最低位（S_BIT）移出。移出的数据放在溢出标志位（SM1.1）中。当正向移动时，N 为正值，输入数据从最低位（S_BIT）移入，最高位移出。移出的数据放在溢出标志位（SM1.1）中。

移位寄存器的最大长度为 64 位，可正可负。下图给出了 N 为正和负两种情况下的移位过程。

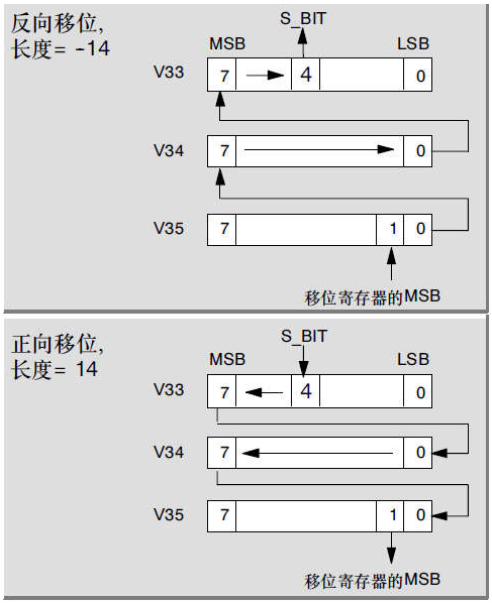
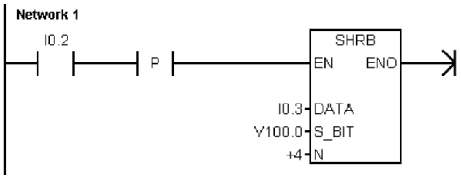
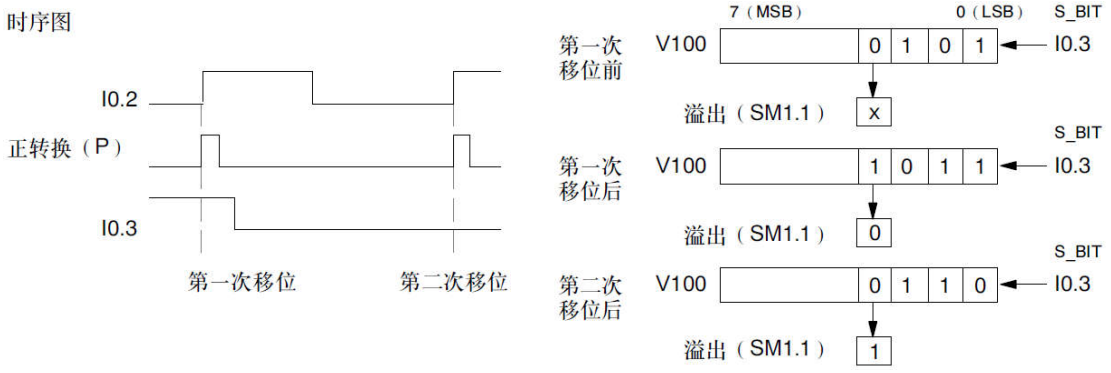


图 3-20 移位寄存器的入口和出口

示例： 移位寄存器指令

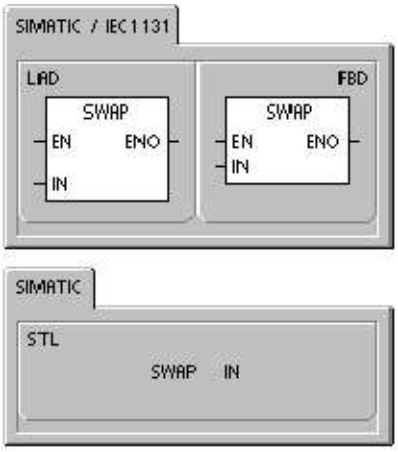



```
Network1

LD      I0.2
EU
SHRB    I0.3，V100.0，+4
```



北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

3.15.4 字节交换指令



字节交换指令用来交换输入字 IN 的高字节和低字节。

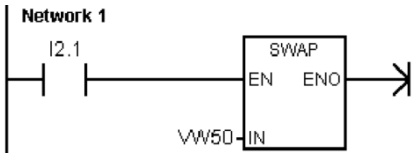
使 ENO=0 的错误条件：

- 0006（间接寻址）

表 3-39 字节交换指令的有效操作数

输入/输出	数据类型	操作数
IN	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC

示例：交换指令

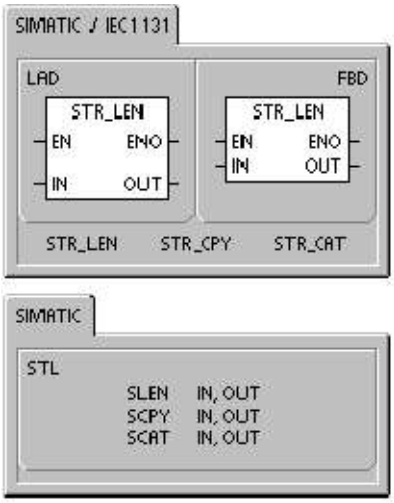



```
Network 1
LD      I2.1
SWAP    VW50
```

SWAP VW50 D6 C3 → VW50 C3 D6

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.16 字符串指令



3.16.1 字符串长度

字符串长度指令（SLEN）返回 IN 中指定的字符串的长度值。

3.16.2 字符串复制

字符串复制指令（SCPY）将 IN 中指定的字符串复制到 OUT 中。

3.16.3 字符串连接

字符串连接指令（SCAT）将 IN 中指定的字符串连接到 OUT 中指定字符串的后面。

SM 标志位和 ENO

对于字符串长度、字符串复制和字符串连接指令，下列条件影响 ENO。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）

表 3-40 字符串长度指令的有效操作数

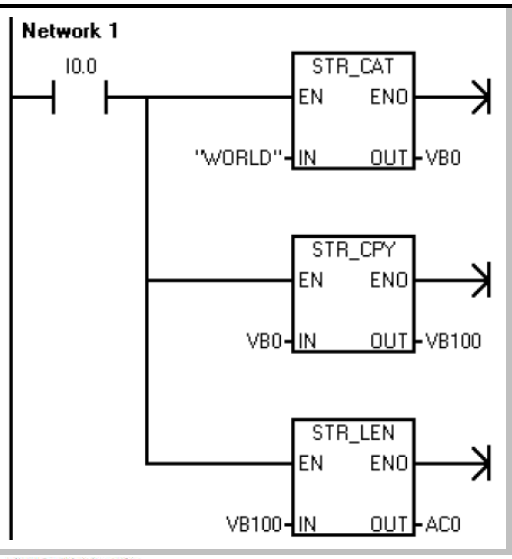
输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

表3-41字符串复制和字符串连接

输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	STRING	VB、LB、*VD、*AC、*LD

示例：字符串连接、字符串复制和字符串长度指令程序举例

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com




```
Network 1     //1.  将"WORLD"上的字符串
              //     附加到VB0上的字符串之后
              //2.  将VB0中的字符串复制到VB100中。
              //3.  得到VB100中存储的字符串的长度。
LD            I0.0
SCAT          "WORLD"，VB0
STRCPY        VB0，VB100
STRLEN        VB100，AC0
```

程序执行前

VB0						VB6
6	'H'	'E'	'L'	'L'	'O'	' '

程序执行后

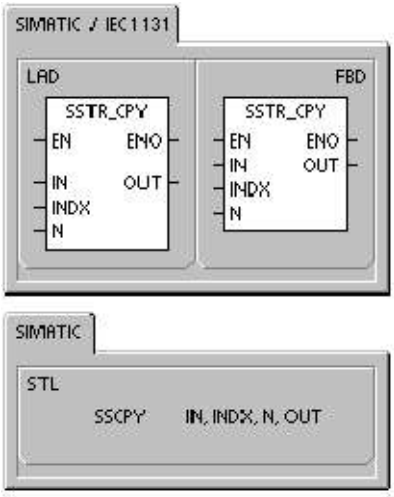
VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

VB100											VB111
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

AC0
11

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.16.4 从字符串中复制子字符串



从字符串中复制子字符串指令（SSCPY）从 INDX 指定的字符号开始，将 IN 中存储的字符串中的 N 个字符复制到 OUT 中。

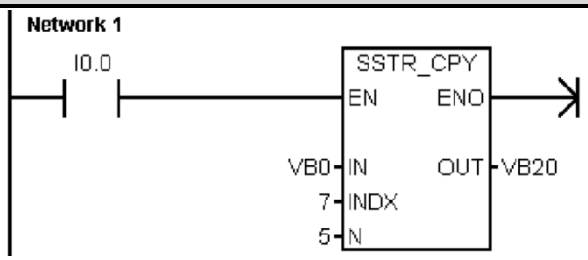
使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（INDX=0）

表 3-42 从字符串中复制子字符串指令

输入/输出	数据类型	操作数
IN	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	STRING	VB、LB、*VD、*LD、*AC
INDX、N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数

示例：复制子字符串指令




```
Network1        //从VB0中字符串的第7个字符开始，
                //复制5个字符到VB20开始的新字符串。
    LD          I0.0
    SSCPY       VB0，7，5，VB20
```

程序执行前

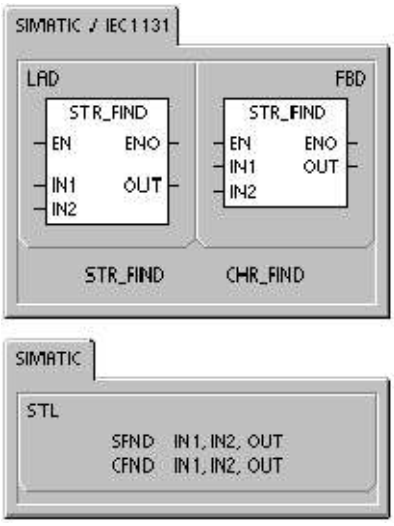
VB0											VB11
11	'H'	'E'	'L'	'L'	'O'	' '	'W'	'O'	'R'	'L'	'D'

程序执行后

VB20					VB25
5	'W'	'O'	'R'	'L'	'D'

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.16.5 字符串搜索



字符串搜索指令（SFND）在 IN1 字符串中寻找 IN2 字符串。从由 OUT（必须位于字符串长度的范围中）指定的开始位置进行搜索。如果在 IN1 中找到了与 IN2 中字符串相匹配的一段字符，则 OUT 中会存入这段字符中首个字符的位置。如果没有找到，OUT 被清 0。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（INDX=0）

➢ 字符搜索

字符搜索指令（CFND）在 IN1 字符串中寻找 IN2 字符串中的任意字符。从起始位置 OUT（必须位于字符串长度的范围 1 中）开始进行搜索。如果找到了匹配的字符，字符的位置被写入 OUT 中。如果没有找到，OUT 被清 0。

使 ENO=0 的错误条件：

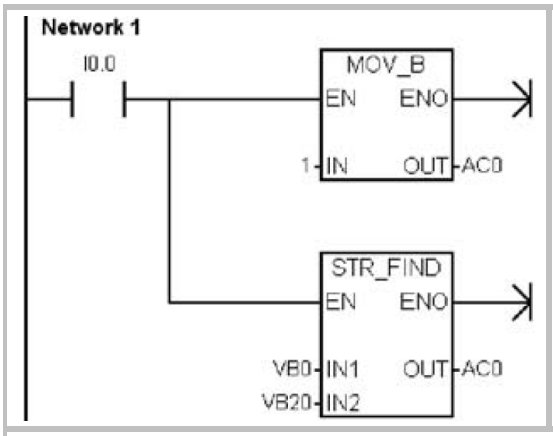
- 0006（间接寻址）
- 0091（操作数超出范围）
- 009B（INDX=0）

表 3-43 字符串搜索和字符搜索指令的有效操作数

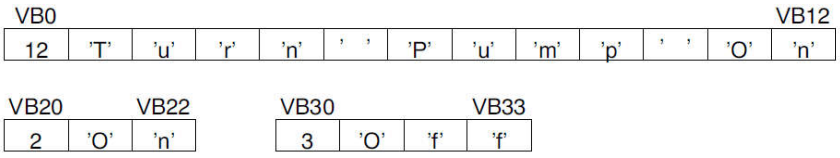
输入/输出	数据类型	操作数
IN1、IN2	STRING	VB、LB、*VD、*LD、*AC、字符串常数
OUT	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC

示例：字符串搜索
在以下例子中，用存储在VB0中的字符串作为泵的启/停命令。字符串'On'存储在VB20中，字符串'Off'存 储在VB30中。搜索结果在AC0中（OUT参数）。如果结果不是0，就说明在命令字符串中找到了字符串'On' （VB12）。

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com




```
Network 1                    //1.  置AC0为1。
                             //      （AC0用作OUT
参数。）
                             //2.  在VB0字符串中
搜索VB20

                             //字符串（'On'），
                             //从首个字符开始
                             //（AC0=1）。
LD          I0.0
MOVB        1,AC0
SFND        VB0，VB20，AC0
```

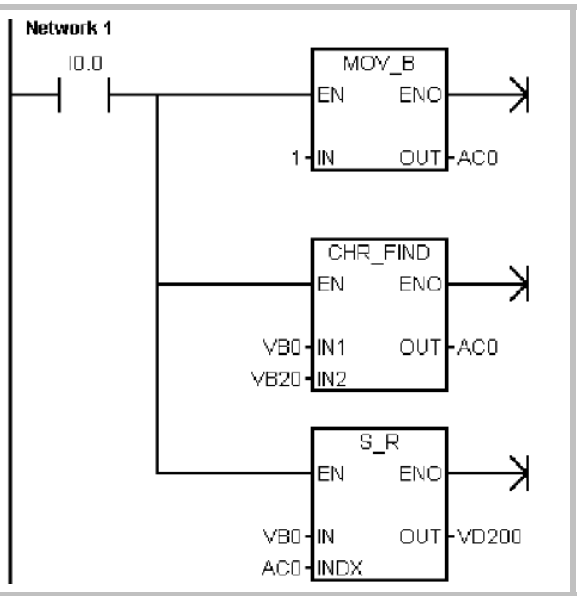


如果找到VB20中的字符串: AC0 = 11

如果没有找到VB20中的字符串: AC0 = 0

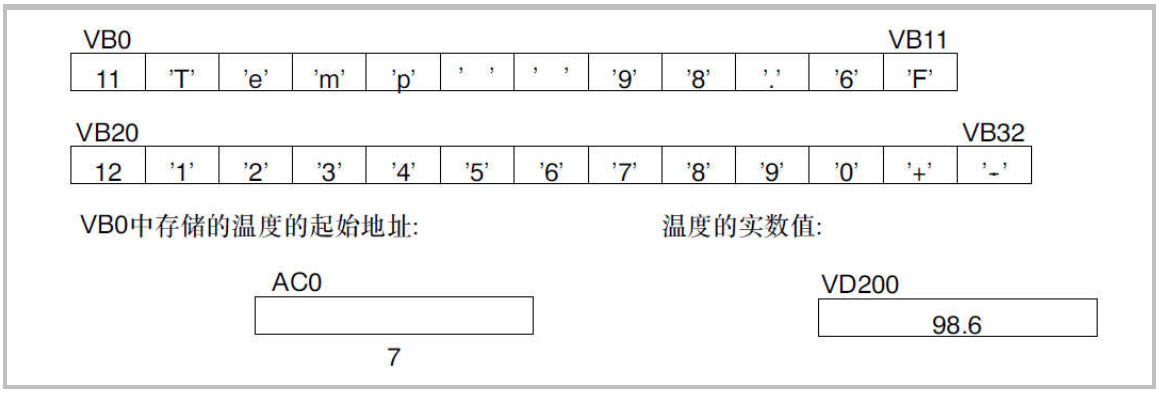
示例：字符串搜索指令

在以下例子中，存储在VB0的字符串包含温度值。存储在VB20中的字符串包括所有的数字（包括+和-），用于 识别字符串中的温度值。该例子程序在字符串中找到数字的启始位置，并将其转换为实数，温度值存放在VD200中。

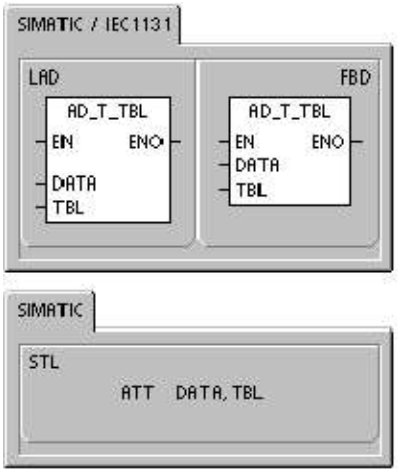



```
Network 1
//1.  置AC0为1。
//AC0用作OUT参数并指向字符串的首个字符

//2.  在VB0字符串中寻找数字字符。

//3.  将字符串转换为实数。

LD          I0.0
MOVB        1,AC0
CFND        VB0，VB20，AC0
STR         VB0，AC0，VD200
```


北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com



3.17 表指令



3.17.1 填表

ATT 指令向表（TBL）中增加一个数值（DATA）。表中第一个数是最大填表数（TL），第二个数是实际填表数（EC），指出已填入表的数据个数。新的数据填加在表中上一个数据的后面。每向表中填加一个新的数据，EC 会自动加 1。

一个表最多可以有 100 条数据。

使 ENO=0 的错误条件：

- SM1.4（表溢出）
- 0006（间接寻址）
- 0091（操作数超出范围）

受影响的 SM 标志位：

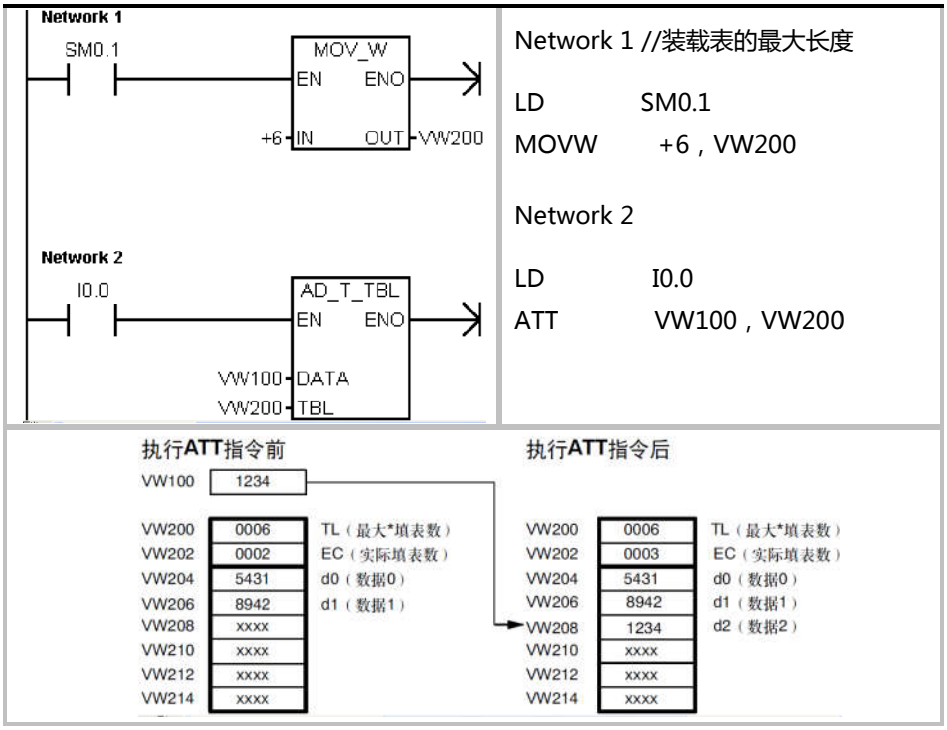
- 如果表出现溢出，SM1.4 会置 1。

表 3-44 指令的有效操作数

输入/输出	数据类型	操作数
DATA	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、 *AC、常数
TBL	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、*VD、*LD、*AC

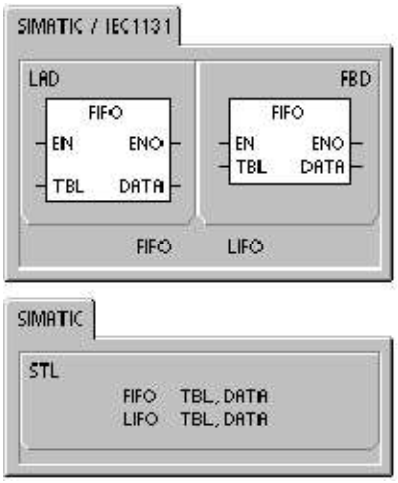
示例：填表指令

北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com



北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

3.17.2 先进先出和后进先出



一个表可以有最多 100 条数据。

➢ 先进先出

先进先出（FIFO）指令从表（TBL）中移走第一个数据，并将此数输出到 DATA。剩余数据依次上移一个位置。每执行一条本指令，表中的数据数减 1。

➢ 后进先出

后进先出（LIFO）指令从表（TBL）中移走最后一个数据，并将此数输出到 DATA。每执行一条本指令，表中的数据数减 1。

使 ENO=0 的错误条件：

- SM1.5（表空）
- 0006（间接寻址）
- 0091（操作数超出范围）

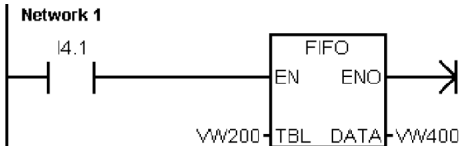
受影响的 SM 标志位：

- 当您试图从一个空表中删除一条数据时，SM1.5 会置 1。

表 3-45 先进先出和先进后出指令的有效操作数

输入/输出	数据类型	操作数
TBL	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、*VD、*LD、*AC
DATA	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、AQW、*VD、*LD、*AC

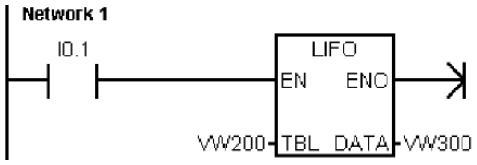
示例：先进先出指令




```
Network 1
LD      I4.1
FIFO    VW200，VW400
```


北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

示例：后进先出指令



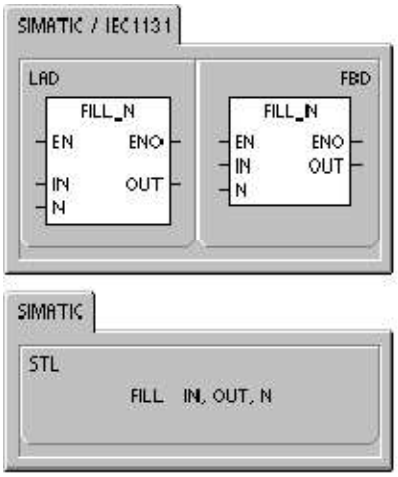
Network 1

LD I0.1

LIFO VW200，VW300

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

3.17.3 内存填充



存储器填充指令（FILL）用输入值（IN）填充从输出（OUT）开始的 N 个字的内容。

N 的范围从 1 到 255。

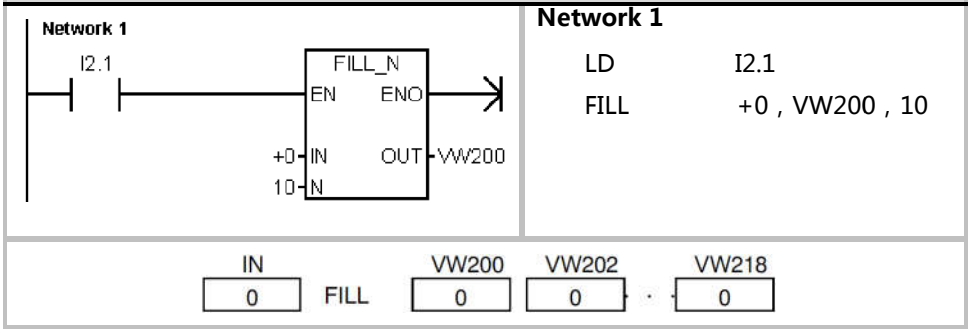
使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）

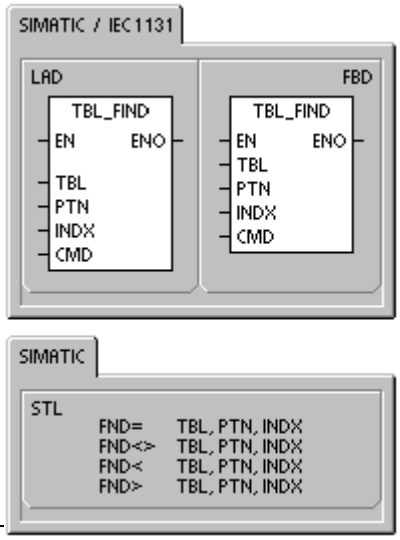
表 3-46 内存填充指令的有效操作数

输入/输出	数据类型	操作数
IN	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
N	BYTE	IB、QB、VB、MB、SMB、SB、LB、AC、*VD、*LD、*AC、常数
OUT	INT	IW、QW、VW、MW、SMW、SW、T、C、LW、AQW、*VD、*LD、*AC

示例：内存填充指令



3.17.4 查表



查表指令（FND）搜索表，以查找符合一定规则的数据。查表指令从 INDX 开始搜索表（TBL），寻找符合 PTN 和条件（=、<>、<或>）的数据。命令参数 CMD 是一个 1～4 的数值，分别代表=、<>、<和>。

如果发现了一个符合条件的数据，那么 INDX 指向表中该数的位置。为了查找下一个符合条件的数据，在激活查表指令前，

必须先对 INDX 加 1。如果没有发现符合条件的数据，那么 INDX 等于 EC。

一个表可以有最多 100 条数据。数据条标号从 0 到 99。

使 ENO=0 的错误条件：

- 0006（间接寻址）
- 0091（操作数超出范围）

表 3-47 查表指令的有效操作数

输入/输出	数据类型	操作数
TBL	WORD	IW、QW、VW、MW、SMW、T、C、LW、*VD、*LC、*AC
PTN	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数
INDX	WORD	IW、QW、VW、MW、SMW、SW、T、C、LW、AC、*VD、*LD、*AC
CMD	BYTE	（常数）1：等于（=），2：不等于（<>），3：小于（<），4：大于（>）

提示

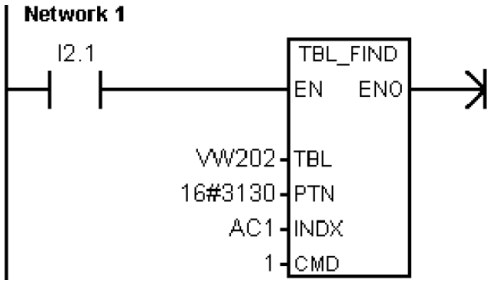
当您用 FND 指令查找由指令 ATT、LIFO 和 FIFO 生成的表时，实际填表数（EC）和输入数据相符，直接对应。最大填表数（TL）对 ATT、LIFO 和 FIFO 指令是必需的，但 FND 指令并不需要它。

因此，FND 指令的操作数 SRC 是一个字地址（指向 EC），比相应的 ATT、LIFO 或 FIFO 指令的操作数 TABLE 要高 2 个字节。

图3-21 FND指令与ATT、LIFO和FIFO指令所使用的表格式上的差异

示例：查表指令

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com



Network 1

LD I2.1

FND= VW202，16#3130，AC1

当I2.1接通时，搜索表，寻找和3130 HEX相等的值。

VW202	0006	EC（实际填表数）
VW204	3133	d0（数据0）
VW206	4142	d1（数据1）
VW208	3130	d2（数据2）
VW210	3030	d3（数据3）
VW212	3130	d4（数据4）
VW214	4541	d5（数据5）

如果表是用ATT、LIFO和FIFO指令创建的，VW200包含了允许的最大填表数，而Find指令不需要它。

AC1	0	从表头开始查找，AC1必须置为0。
执行查表 AC1	2	AC1中保存了第1个符合查表条件的数据编号d2）。
AC1	3	查表中剩余数据前，INDX加1。
执行查表 AC1	4	AC1中保存了第2个符合查表条件的 数据编号（d4）。
AC1	5	查表中剩余数据前，INDX加1。
执行查表 AC1	6	AC1中保存了已填表数。整个表已经查完，没有发现另外的匹配数据。
AC1	0	再次查表前，INDX的值必须复位到0。

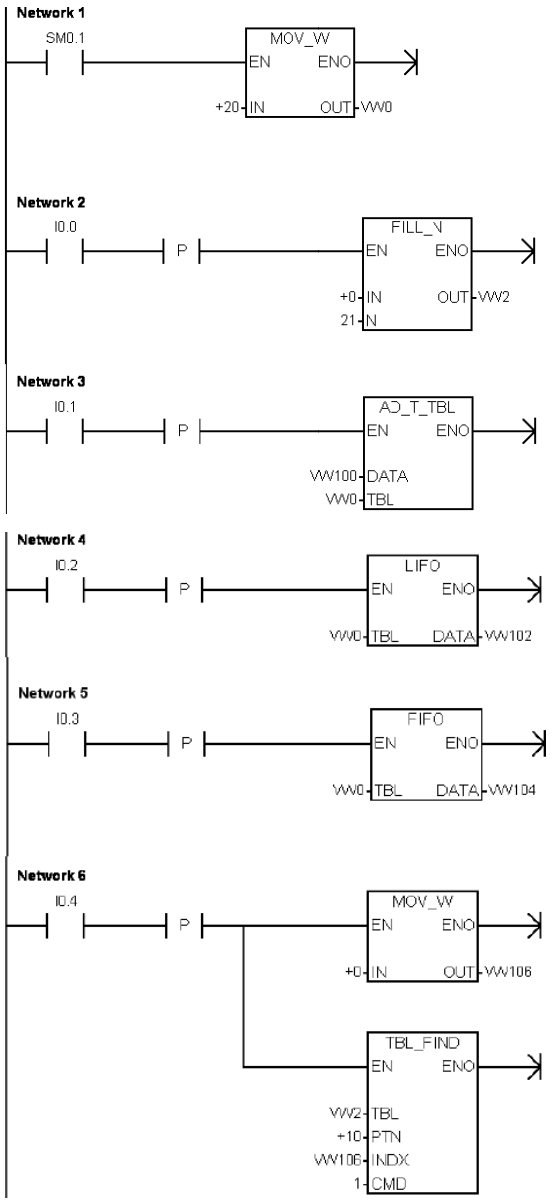
示例：创建一个表

下列程序创建一个包含20条数据的表。存储区中的第一个数据为表的长度（在本例中为20）。存储区中的第二个数据为表中数据的实际个数。其它存储区单元为数据。一个表可以有最多100条数据。其中不包括定义表长度和实际数据个数的两个单元（在本例中为VW0和VW2）。当CPU执行第一条指令时，表中的实际数据个数（VW2）会自动增或者减。

在使用表之前，必须为表指定数据的最多个数。否则您将无法在表中插入数据。同时，要确保使用边沿触发来激活读写指令。

在查表之前，INDX（VW106）必须清0。如果找到匹配的数据，INDX中会存入表中的条目号；如果没有找到，INDX中为实际数据个数（VW2）。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

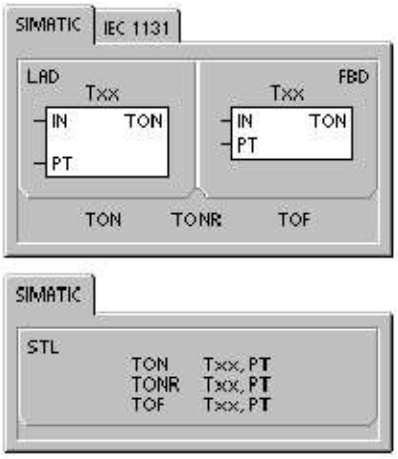



```
Network 1   //从VW4开始创建一个20
            //条数据的表。
            //1.  在首次扫描定义表的最大长度。
LD              SM0.1
MOVW             +20，VW0

Network 2   //用I0.0将表清0。
             //当I0.0产生上升沿时，
             //从VW2开始清0。
LD              I0.0
EU
FILL             +0，VW2，21

Network 3        //用I0.1向表中写数据。
                 //当I0.1产生上升沿时，
                 //将VW100的数
                 //据写入表中。
LD               I0.1
EU
ATT              VW100，VW0

Network 4   //用I0.2
            //读表中的最后一个数据。
            //将表中最后一个数据值移入VW102
            //这会减少条目的数量。
            //当I0.2产生上升沿时，
            //将表中最后一个数
            //据移入VW102中。
LD                  I0.3
EU
LIFO                VW0，VW104

Network 5 //用I0.3
            //读表中的最后一个数据。
            //将表中第一个数据值移入VW104中。
            //这会减少条目的数量。
            //当I0.3产生上升沿时，
            //将表中第一个数
            //据移入VW104中。
LD I0.3
EU
FIFO VW0，VW104

Network 6    //在表中搜索是否有数据为10。
            //1.  当I0.4产生上升沿时，
            //将INDX指针清0。
            //2.  在表中搜索等于10的数据。
LD                  I0.4
EU
MOVW                +0，VW106
FND=                VW2，+10，VW106
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.18 定时器指令

3.18.1 接通延时定时器



➢ 有记忆的接通延时定时器

接通延时定时器（TON）和有记忆的接通延时定时器在使能输入接通时记时。定时器号（Txx）决定了定时器的分辨率，并且分辨率现在已经在指令盒上标出了。

3.18.2 断开延时定时器

断开延时定时器用于在输入断开后延时一段时间断开输出。定时器号（Txx）决定了定时器的分辨率，并且分辨率现在已经在指令盒上标出了。

表 3-48SIMATIC 定时器指令的有效操作数

输入/输出	数据类型	操作数
TXX	WORD	常数（T0到T255）
IN	BOOL	I、Q、V、M、SM、S、T、C、L、能流
PT	INT	IW、QW、VW、MW、SMW、SW、LW、T、C、AC、AIW、*VD、*LD、*AC、常数

提示

不能将同一个定时器号同时用作 TOF 和 TON。例如，不能够既有 TON T32 又有 TOF T32。

如下表所示，三类定时器用于执行不同类型的定时任务：

- 接通延时定时器（TON）用于单一间隔的定时
- 有记忆接通延时定时器（TONR）用于累计许多时间间隔
- 断开延时定时器（TOF）用于关断或者故障事件后的延时（例如：在电机停后，需要冷却电机）

表 3-49 定时器指令的操作数

定时器类型	当前值>=预设值	使能输入（IN）的状态	上电周期/首次扫描
TON	定时器位ON，当前连续计数到32767	ON：当前值计数时间 OFF：定时器位OFF，当前值=0	定时器位OFF，当前值=0
TONR	定时器位ON，当前连续计数到3276	ON：当前值计数时间 OFF :定时器位和当前值保持最后状态	定时器位OFF，当前值保持1

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

TOF	定时器位OFF， 当前值=预设值， 停止计数	ON：定时器位ON，当前值=0 OFF :发生ON到OFF的跳变之后， 定时器计数	定时器位OFF 当前值=0

当使能输入接通时，接通延时定时器和有记忆接通延时定时器开始计时，当定时器的当前值（Txxx）大于等于预设值时，该定时器位被置位。

- 当使能输入断开时，清除接通延时定时器的当前值，而对于有记忆接通延时定时器，其当前值保持不变。
- 可以用有记忆接通延时定时器累计输入信号的接通时间，利用复位指令（R）清除其当前值。
- 当达到预设时间后，接通延时定时器和有记忆接通延时定时器继续计时，一直计到最大值32767。

断开延时定时器（TOF）用来在输入断开后延时一段时间断开输出。当使能输入接通时，定时器位立即接通，并把当前值设为 0。当输入断开时，定时器开始定时，直到达到预设的时间。

- 当达到预设时间时，定时器位断开，并且停止计时当前值。当输入断开的时间短于预设时间时，定时器位保持接通。
- TOF 指令必须用输入信号的接通到断开的跳变启动计时。
- 如果 TOF 定时器在顺控（SCR）区，而且顺控区没有启动，TOF 定时器的当前值设置为 0，定时器位设置为断开，当前值不计时。

提示

可以只使用复位（R）指令来复位 TONR。还可以使用复位指令去复位 TON 或 TOF 中的任何一个。

复位指令执行如下的操作：

定时器位=OFF

定时器当前位置=0

TONR 定时器只能通过复位指令进行复位操作。复位后，为了再启动，TOF 定时器需要使能输入有一个从 ON 到 OFF 的跳变。

为定时器选择分辨率

定时器对时间间隔记数。定时器的分辨率（时基）决定了每个时间间隔的时间长短。例如：一个以 10ms 为时基的延时接通定时器，在使能位接通后，以 10ms 的时间间隔计数，10ms 的定时器计数值为 50 代表 500ms。SIMATIC 定时器有三种分辨率：1ms、10ms 和 100ms。如表 6-74 所示，定时器号决定了定时器的分辨率。

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

提示

为确保时间间隔的最小值，预置值必须比它大 1。例如：为确保最小时间间隔 2100ms，要将 100ms 定时器的预置值 PV 设为 22。

表 3-50 定时器号和分辨率

定时器类型	用毫秒（ms）表示的分辨率	用秒（s）表示的最大值	定时器号
TONR	1ms	32.767s	T0,T64
	10ms	327.67s	T1 -- T4，T65 -- T68
	100ms	327.67s	T5 -- T31，T69--T95
TON、TOF	1ms	32.767s	T32,T96
	10ms	32.767s	T33 -- T36，T97 -- T100
	100ms	32.767s	T37 -- T63，T101 -- T255

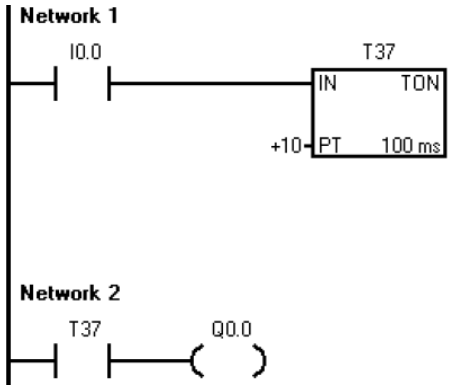
分辨率对定时器的影响

对于 1ms 分辨率的定时器来说，定时器位和当前值的更新不与扫描周期同步。对于大于 1 ms 的程序扫描周期，定时器位和当前值在一次扫描内刷新多次。

对于 10ms 分辨率的定时器来说，定时器位和当前值在每个程序扫描周期的开始刷新。定时器位和当前值在整个扫描周期过程中为常数。在每个扫描周期的开始会将一个扫描累计的时间间隔加到定时器当前值上。

对于 100ms 分辨率的定时器来说，定时器位和当前值在指令执行时刷新。因此，为了使定时器保持正确的定时值，要确保在一个程序扫描周期中，只执行一次 100ms 定时器指令。

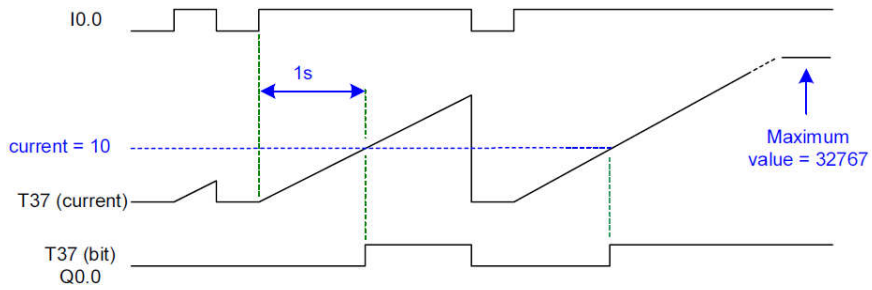
示例：接通延时定时器




```
Network 1     //100 ms 定时器T37在
              // ( 10 x 100 ms = 1s ) 后到时。
              //I0.0 ON=T37 使能，
              //I0.0 OFF=禁止并复位T37
LD            I0.0
TON           T37，+10
Network 2     //定时器T37控制Q0.0。
LD            T37
=             Q0.0
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

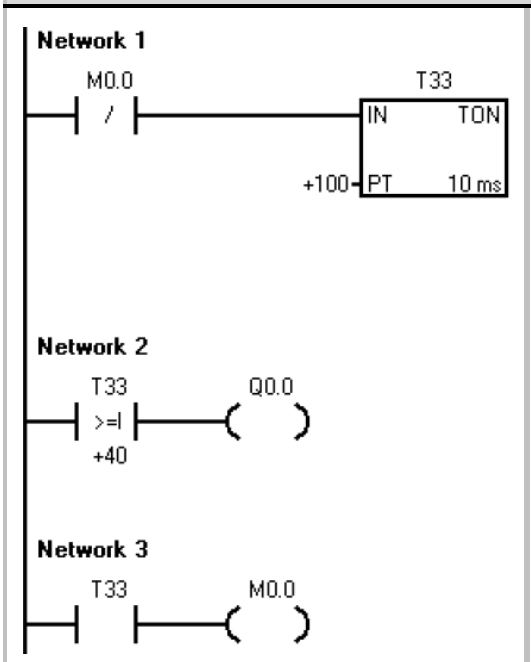
时序图



提示

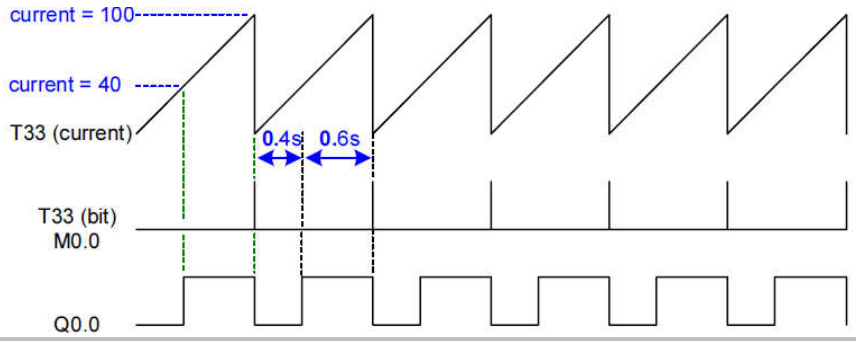
为了确保在每一次定时器达到预设值时，自复位定时器的输出都能接通一个程序扫描周期，用一个常闭触点来代替定时器位作为定时器的使能输入

示例：SIMATIC自复位接通延时定时器



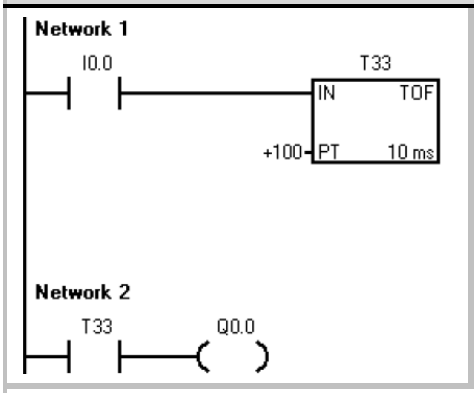

```
Network 1       //10 ms 定时器T33在
     // ( 100 x 10 ms = 1s ) 后到时。
     //M0.0脉冲过快，以致在状态视图中
     //无法监视
LDN                 M0.0
TON                 T33，+100

Network 2       /比较指令为真的时间较长，
                //可以在状态表中
                //监视 ,Q0.0的占空比为40%
LDW>=               T33，+40
=                   Q0.0

Network 3       //T33（位）的脉冲过窄，
                //在状态表中无法监视。
                //在1秒后复位M0.0。
LD                  T33
=                   M0.0
```

时序图



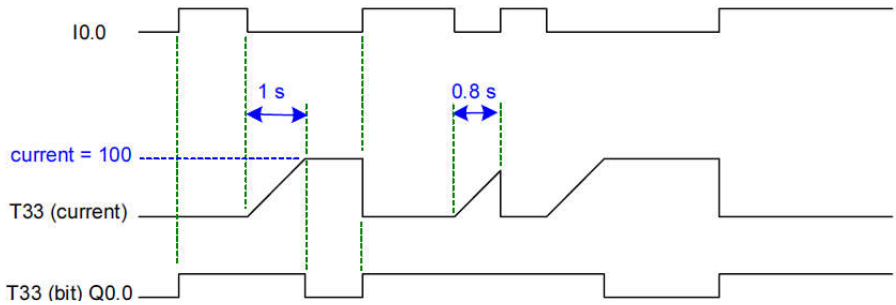
北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

示例：SIMATIC断开延时定时器

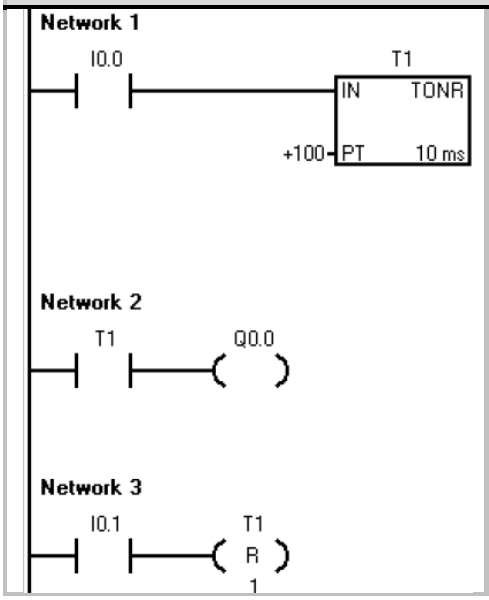



```
Network 1       //10ms定时器T33在1秒后到时。
                //I0.0关断使能T33。
                //I0.0接通T33复位。
LD              I0.0
TOF             T33，+100
Network 2       //定时器T33用其输出位控制0.0。
    LD          T33
    =           Q0.0
```

时序图

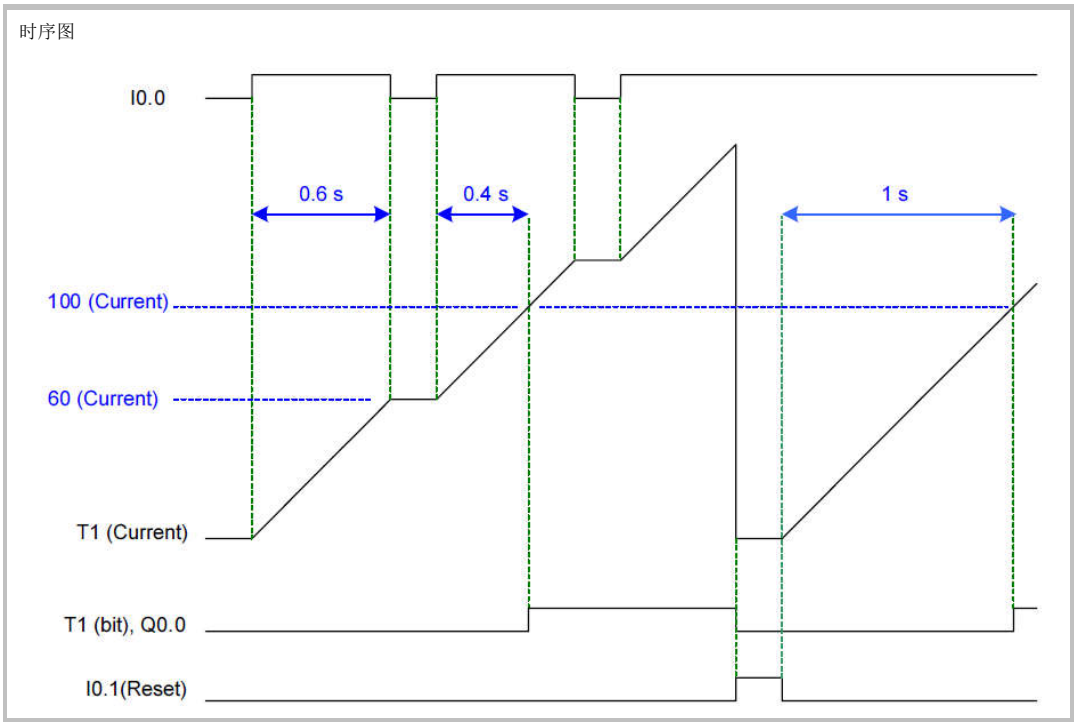


示例：SIMATIC有记忆的接通延时定时器




```
Network 1       //10 ms TONR定时器T1在
                //PT=( 100 x 10 ms=1s )后到时。
LD              I0.0
TONR            T1，+100
Network 2       //T1位控制Q0.0。
                //1秒后T1使Q0.0接通。
LD              T1
=               Q0.0
Network 3       //TONR定时器必须用复位
                //指令才能复位。
                //当I0.1接通时，复位T1。
LD              I0.1
R               T1
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com



3.18.3 时间间隔定时器

触发时间间隔

触发时间间隔（BITIM）指令读内置的 1 毫秒计数器的当前值，并将此值存储到 OUT 中。双字毫秒值的最大定时间隔是 2 的 32 次幂或 49.7 天。

计算时间间隔

计算时间间隔（CITIM）指令计算当前时间和 IN 提供的值之间的时间差。时间差被存储在 OUT 中。双字毫秒值的最大定时间隔是 2 的 32 次幂或 49.7 天。依据于 BITIM 指令执行的时间，CITIM 自动处理在最大间隔内发生的 1 毫秒定时器翻转。

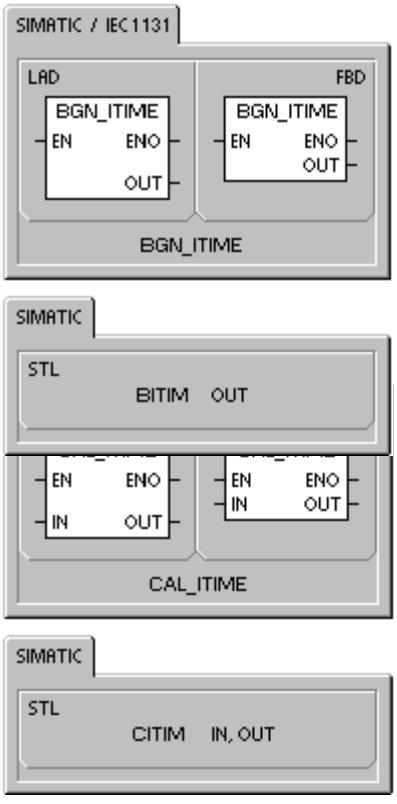


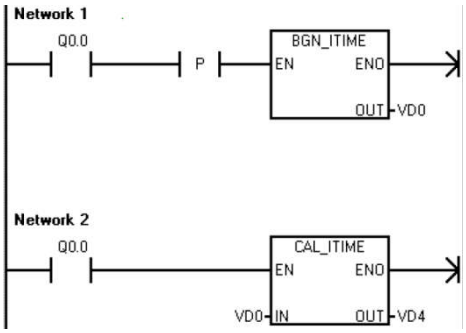
表 3-51 间隔定时器指令的有效操作数

输入/输出	数据类型	操作数

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com

IN	DWORD	VD、ID、QD、MD、SMD、SD、LD、HC、AC、*VD、*LD、*AC
OUT	DWORD	VD、ID、QD、MD、SMD、SD、LD、AC、*VD、*AC、*LD

示例：SIMATIC触发时间间隔和计算时间间隔




```
Network 1          //捕获Q0.0接通的时刻。
LD                 Q0.0
EU
BITIM              VD0
Network 2          // 计算Q0.0已经接通的时间。
LD                 Q0.0
CITIM              VD0，VD4
```


北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

3.19 子程序指令

子程序调用指令（CALL）将程序控制权交给子程序 SBR_N。

调用子程序时可以带参数也可以不带参数。子程序执行完成后，控制权返回到调用子程序的指令的下一条指令。

子程序条件返回指令（CRET）根据它前面的逻辑决定是否终止子程序。

使 ENO=0 的错误条件：

- 0008（超过子程序嵌套最大限制）
- 0006（间接寻址）

在主程序中，可以嵌套调用子程序（在子程序中调用子程序），最多嵌套 8 层。在中断服务程序中，不能嵌套调用子程序。

在被中断服务程序调用的子程序中不能再出现子程序调用。不禁止递归调用（子程序调用自己），但是当使用带子程序的递归调用时应慎重。

表 3-52 子程序指令的有效操作数

输入/输出	数据类型	操作数
SBR_N	WORD	常数 0到127
IN	BOOL BYTE WORD、INT DWORD DINT STRING	V、I、Q、M、SM、S、T、C、L、能流 VB、IB、QB、MB、SMB、SB、LB、AC、*VD、*LD、*AC 1、常数 VW、T、C、IW、QW、MW、SMW、SW、LW、AC、AIW、*VD、*LD、*AC 1、常数 VD、ID、QD、MD、SMD、SD、LD、AC、HC、*VD、*LD、*AC 1 VB、&IB、&QB、&MB、&T、&C、&SB、&AI、&AQ、&SMB、常数 *VD、*LD、*AC、常数
输入/输出	BOOL BYTE WORD、INT DWORD、 DINT	V、I、Q、M、SM 2、S、T、C、L VB、IB、QB、MB、SMB 2、SB、LB、AC、*VD、*LD、*AC1 VW、T、C、IW、QW、MW、SMW 2、SW、LW、AC、*VD、*LD、*AC1 VD、ID、QD、MD、SMD 2、SD、LD、AC、*VD、*LD，*A
OUT	BOOL BYTE WORD、INT DWORD、 DINT	V、I、Q、M、SM 2、S、T、C、L VB、IB、QB、MB、SMB 2、SB、LB、AC、*VD、*LD、*AC1 VW、T、C、IW、QW、MW、SMW 2、SW、LW、AC、AQW、*VD、*LD、*AC1 VD、ID、QD、MD、SMD 2、SD、LD、AC、*VD、*LD，*AC1

当有一个子程序被调用时，系统会保存当前的逻辑堆栈，置栈顶值为 1，堆栈的其他值为零，把控制交给被调用的子程序。当子程序完成之后，恢复逻辑堆栈，把控制权交还给调用程序。

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

因为累加器可在主程序和子程序之间自由传递，所以在子程序调用时，累加器的值既不保存也不恢复。

当子程序在同一个周期内被多次调用时，不能使用上升沿、下降沿、定时器和计数器指令。

带参数调用子程序

子程序可以包含要传递的参数。参数在子程序的局部变量表中定义。参数必须有变量名（最多 23 个字符）、变量类型和数据类型。一个子程序最多可以传递 16 个参数。

局部变量表中的变量类型区定义变量是传入子程序（IN）、传入和传出子程序（IN_OUT）或者传出子程序（OUT）。表 6-79 中描述了一个子程序中的参数类型。要加入一个参数，把光标放到要加入的变量类型区（IN、IN_OUT、OUT）。点击鼠标右键可以得到一个菜单选择。选择插入选项，然后选择下一行选项。这样就出现了另一个所选类型的参数项。

表 3-53 子程序的参数类型

参数	中断描述
IN	参数传入子程序。如果参数是直接寻址（如：VB10），指定位置的值被传递到子程序。如果参数是间接寻址（如：*AC1），指针指定位置的值被传入子程序；如果参数是常数（如：16＃1234），或者一个地址（如：&VB100），常数或地址的值被传入子程序。
IN_OUT	指定参数位置的值被传到子程序，从子程序的结果值被返回到同样地址。常数（如：16＃1234）和地址（如：&VB100）不允许作为输入/输出参数。
OUT	从子程序来的结果值被返回到指定参数位置。常数(如 :16＃1234）和地址（如：&VB100） 不允许作为输出参数。由于输出参数并不保留子程序最后一次执行时分配给它的数值，所以必须在每次调用子程序时将数值分配给输出参数。注意：在电源上电时，SET和RESET指令只影响布尔量操作数的值。
TEMP	任何局部存储器都不能用来传递参数，只能在子程序内部暂时存贮数据。

局部变量表中的数据类型区定义了参数的大小和格式。参数类型如下所示：

布尔：该数据类型用于单独位的输入和输出。下例中的 IN3 是布尔输入。

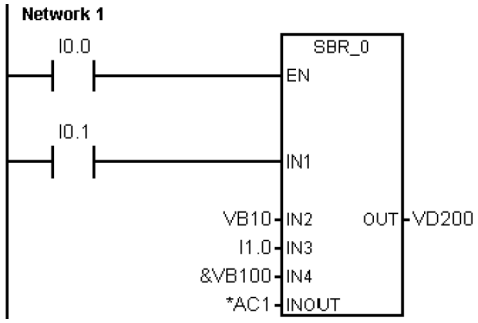
- 字节、字和双字：这些数据类型分别指明一个 1、2 或者 4 个字节的无符号输入或输出参数。
- 整数、双整数：这些数据类型分别指明 1、2 或者 4 个字节的有符号输入或输出参数。
- 实数：该数据类型指明一个（4 字节）IEEE 浮点值。
- STRING：该数据类型用作一个指向字符串的四字节指针。

- 能流：布尔能流仅允许对位输入操作。在局部变量表中布尔能流输入必须出现在其它类型的前面。只有输入参数可以这样使用。下例中的使能输入（EN）和 IN1 输入使用布尔逻辑。

示例：子程序调用

以下有两个STL程序。第一个程序只能在STL编辑器中以STL的形式显示，因为用作能流输入的BOOL参数没有 存储在L存储区中。

第二个程序能够在LAD和FBD编辑器中显示，因为使用了L存储器来存储用作能流输入的BOOL输入参数。



只能显示STL：

```
Network1
LD I0.0
CALL SBR_0，I0.1，VB10，I1.0，&VB100，*AC1，VD200
```

可以在LAD和Network1 FBD中正确显示：

```
LD I0.0
= L60.0
LD I0.1
= L63.7
LD L60.0
CALL SBR_0，L63.7，VB10，I1.0，&VB100，*AC1，VD200
```

地址参数（如 IN4 处的&VB100）以一个双字（无符号）的值传送到子程序。在带常数调用程序时必须指明常数类型。例如，把值为 12345 的无符号双字作为参数进行传递，常数参数必须用 DW＃12345 指明。如果参数中缺少了常数描述符，常数可能被当作不同的类型。

输入或输出参数上没有自动数据类型转换功能。例如，如果局部变量表明一个参数具有实型，而在调用时使用一个双字，子程序中的值就是双字。

当给子程序传递值时，它们放在子程序的局部存储器中。局部变量表的最左列是每个被传递参数的局部存储器地址。当子程序调用时，输入参数值被拷贝到子程序的局部存储器。当子程序完成时，从局部存储器区拷贝输出参数值到指定的输出参数地址。

数据单元的大小和类型用参数的代码表示。在子程序中局部存储器的参数值的分配如下所示：

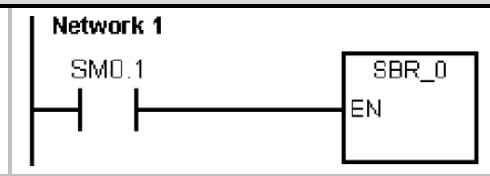
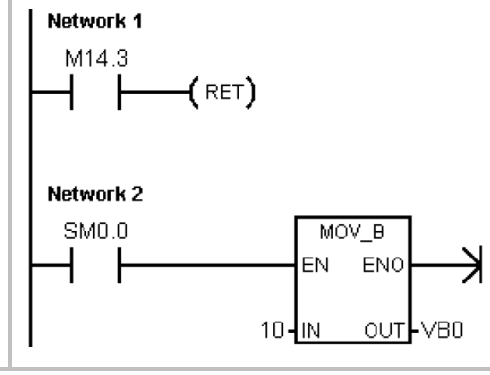
- 按照子程序指令的调用顺序，参数值分别给局部存储器，起始地址是 L0。
- 1 到 8 连续位参数值分配一个字节，从 Lx.0 到 Lx.7。
- 字节、字和双字值按照所需字节分配在局部存储器中（LBx、LWx 或 LDx）。

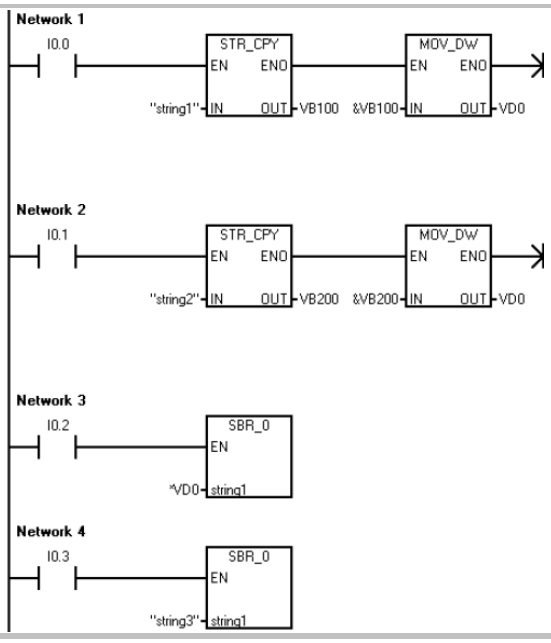
在带参数调用子程序指令中，参数必须按照一定顺序排列，输入参数在最前面，其次是输入/输出参数，然后是输出参数。

北京捷麦顺驰科技有限公司　　地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471　　网址：http://www.T50rtu.com

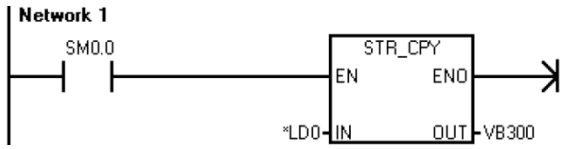
如果用语句表编程，CALL 指令的格式是：

CALL 子程序号，参数 1，参数 2，…，参数

示例：子程序和子程序返回指令程序举例		
MAIN		Network 1 //在首次扫描，调用初始化子程序0。 LD SM0.1 CALL SBR_0
SBR0		Network 1 //你可以使用条件返回指令在子程序 //结束之前返回。 LD M14.3 CRET Network 2 //如果M14.3接通 本段程序会被跳过。 LD SM0.0 MOVB 10，VB0

示例：带字符串的子程序调用		
该实例依据给定的输入，复制不同的字符串文字到单独的地址。字符串的单独地址被保存。然后通过使用直接地址，将字符串地址传递给子程序。子程序输入参数的数据类型是字符串。然后子程序移动字符串到不同位置。 字符串文字也可以被传递给子程序。子程序内的字符串引用一直是相同的。		
MAIN		**Network 1** // LD I0.0 SSCPY “string1”，VB100 AENO MOVD &VB100，VD0 **Network2** // LD I0.1 SSCPY “string2”，VB200 AENO MOVD &VB200，VD0 Network3 // LD I0.2 CALL SBR_0，*VD0

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

SBR0		**Network 1** // LD SM0.0 SSCPY *LD0 , VB300

北京捷麦顺驰科技有限公司
地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471
网址：http://www.T50rtu.com

附录

A:特殊存储器（SM）标志位

特殊存储器标志位提供大量的状态和控制功能，并能起到在 CPU 和用户程序之间交换信息的用。特殊存储器标志位能以位、字节、字或双字使用。

➢ SMB0：状态位

如下表所示，SMB0 有 8 个状态位，在每个扫描周期的末尾，由无线 PLC01 更新这些位。

表0-1特殊存储器字节SMB0（SM0.0至SM0.7）

SM位	描述（只读）
SM0.0	该位始终为1 √
S0.1	该位在首次扫描时为1，用途之一是调用初始化子程序 √
SM0.2	若保持数据丢失，则该位在一个扫描周期中为1。该位可用作错误存储器位，或用来调用特殊启动顺序功能。
SM0.3	开机后进入RUN方式，该位将ON一个扫描周期，该位可用作在启动操作之前给设备提供一个预热时间
SM0.4	该位提供了一个时钟脉冲，30秒为1，30秒为0，周期为一分钟，它提供了一个简单易用的延时或 1分钟的时钟脉冲 √
SM0.5	该位提供了一个时钟脉冲，0.5秒为1，0.5秒为0，周期为1秒钟。它提供了一个简单易用的延时或1秒钟的时钟脉冲 √
SM0.6	该位为扫描时钟，本次扫描时置1，下次扫描时置0。可用作扫描计数器的输入 √
SM0.7	该位指示CPU工作方式开关的位置（0为TERM位置，1为RUN位置）。当开关在RUN位置时，用该位可使自由端口通信方式有效，那么当切换至TERM位置时，同编程设备的正常通讯也会有效。 无线PLC01的V1.0版本开关一直在RUN位置上，该值为1 √

➢ SMB1：状态位

SMB1 包含了各种潜在的错误提示。这些位可由指令在执行时进行置位或复位。

表 0-2 特殊存储器字节 SMB1（SM1.0 至 SM1.7）

SM位	描述（只读）
SM1.0	当执行某些指令，其结果为0时，将该位置1。 √
SM1.1	当执行某些指令，其结果溢出或查出非法数值时，将该位置1。 √
SM1.2	当执行数学运算，其结果为负数时，将该位置1。 √
SM1.3	试图除以零时，将该位置1。 √
SM1.4	当执行ATT（Add to Table）指令时，试图超出表范围时，将该位置1。 √

SM1.5	当执行LIFO或FIFO指令，试图从空表中读数时，将该位置1。 √
SM1.6	当试图把一个非BCD数转换为二进制数时，将该位置1。 √
SM1.7	当ASCII码不能转换为有效的十六进制数时，将该位置1。 √

➢ SMB4：队列溢出

SMB4 包含中断队列溢出位，中断是否允许标志位及发送空闲位。队列溢出表明要么是中断发生的频率高于 CPU，要么是中断已经被全局中断禁止指令所禁止。

表 0-3 特殊存储器字节 SMB4（SM4.0 至 SM4.7）

SM位	描述（只读）
SM4.0	当通信中断队列溢出时，将该位置1。
SM4.1	当输入中断队列溢出时，将该位置1。
SM4.2	当定时中断队列溢出时，将该位置1。
SM4.3	在运行时刻，发现编程问题时，将该位置1。
SM4.4	该位指示全局中断允许位，当允许中断时，将该位置1。
SM4.5	当（口0）发送空闲时，将该位置1。
SM4.6	当（口1）发送空闲时，将该位置1。
SM4.7	当发生强置时，将该位置1

➢ SMB5：I/O 状态

SMB5 包含 I/O 系统里发现的错误状态位。这些位提供了所发现的 I/O 错误的概况。

表 0-4 特殊存储器字节 SMB5（SM5.0 至 SM5.7）

SM位	描述（只读）
SM5.0	当有I/O错误时，将该位置1
SM5.1	当I/O总线上连接了过多的数字量I/O点时，将该位置1。
SM5.2	当I/O总线上连接了过多的模拟量I/O点时，将该位置1。
SM5.3	当I/O总线上连接了过多的智能I/O模块时，将该位置1。
SM5.4至 SM5.7	保留

➢ SMB27：事件状态标志 1

SMB7包含各事件状态位，当事件条件触发时，想对应的事件状态标志位被置为1，需要注意，事件状态标志位不会自动复位，需要通过用户通过指令复位。

表 0-5 特殊存储器字节 SMB7

SM位	描述（只读）

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

格式	MSB LSB 8 0 U2T | U2R | -- | -- | -- | -- | R1T | U1R
SM27.0	串口(uart),0收到一包数据
SM27.1	串口(uart)0通过中断的方式发送完成一包数据（非中断发送不会置位）
SM27.2--SM27.5	不同的产品有不同的定义
SM7.6	串口(uart)1收到一包数据（针对有双串口的产品）
SM7.7	串口(uart)1通过中断的方式发送完成一包数据（非中断发送不会置位）

➢ SMB30 和 SMB130：自由端口控制寄存器

SMB30 控制自由端口 0 的通讯方式，SMB130 控制自由端口 1 的通讯方式。您可以对 SMB30 和 SMB130 进行写和读。这些字节设置自由端口通讯的操作方式，并提供自由端口或者系统所支持的协议之间的选择。

表0-6特殊存储器字节SMB30

口0	口1	描述
SMB30的格式	SMB130的格式	自由口模式控制字节 MSB LSB 7 0 p | p | d | b | b | b | m | m
SM30.0和 SM30.1	SM130.0和 SM130.1	mm：协议选择 00=点到点接口协议 （PPI/从站模式） 01=自由口协议 10=PPI/主站模式 11=保留（缺省是PPI/从站模式） 注意：当选择mm=10（PPI主站），PLC将成为网络的一个主站，可以执行 NETR和NETW指令。在PPI模式下忽略2到7位。
SM30.2到 SM30.4	SM130.2到 SM130.4	bbb：自由口波特率 000=38,400波特 100=2,400波特 001=19,200波特 101=1,200波特 010=9,600波特 110=115,200波特 011=4,800波特 111=57,600波特
SM30.5	SM130.5	d：每个字符的数据位 0=8位/字符 **停止位1** 1=7位/字符 **停止位2**
SM30.6和	SM130.6和	pp：校验选择

北京捷麦顺驰科技有限公司 地址：北京市丰台区芳城园一区日月天地 B 座 1505
电传：（010）58076471 网址：http://www.T50rtu.com

SM30.7	SM130.7	00=不校验 11=奇校验	10=不校验 01=偶校验

➢ SMB34 和 SMB35：定时中断的时间间隔寄存器

SMB34 分别定义了定时中断 0 和 1 的时间间隔，可以在 1ms~255ms 之间以 1ms 为增量进行设定。若定时中断事件被中断程序所采用，当 CPU 响应中断时，就会获取该时间间隔值。若要改变该时间间隔，您必须把定时中断事件再分配给同一或另一中断程序，也可以通过撤销该事件来终止定时中断事件。

表0-7特殊存储器字节SMB34和SMB35

SM位	描述
SMB34	定义定时中断0的时间间隔（从1ms~255ms，以1ms为增量） √
SMB35	定义定时中断1的时间间隔（从1ms~255ms，以1ms为增量） √

北京捷麦顺驰科技有限公司
电传：（010）58076471
地址：北京市丰台区芳城园一区日月天地 B 座 1505
网址：http://www.T50rtu.com