第 5 章

CHAPTER 5

数据处理、运算指令

PLC 产生初期主要用于在工业控制中以逻辑控制来代替继电器控制。随着计算机技术与 PLC 技术的不断发展与融合，PLC 增加了数据处理功能，使其在工业应用中功能更强，应用范围更广。在当今自动化程度越来越高的加工生产线中，仅仅具备基本指令的功能是远远不够的，还应该具备数据处理和运算的功能。

5.1 数据处理指令

数据处理指令涉及对数据的非数值运算操作，主要包括数据传送、字节交换、存储器填充、字节立即读写、移位、转换等指令。

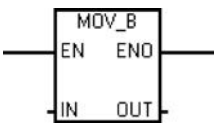
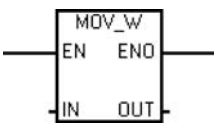
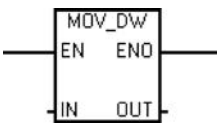
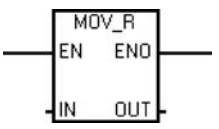
5.1.1 数据传送指令

该类指令用来完成各存储单元之间一个或者多个数据的传送。可分为单个数据传送指令和数据块传送指令。

1. 字节、字、双字、实数单个数据传送(MOV)指令

单个数据传送指令用来传送单个的字节、字、双字、实数。指令格式及功能如表 5-1 所示。

表 5-1 MOV 指令格式

LAD				
STL	MOVB IN,OUT	MOVW IN,OUT	MOVD IN,OUT	MOVR IN,OUT
操作数及数据类型	**IN**：VB,IB,QB,MB,SB,SMB,LB,AC 及常量； **OUT**：VB,IB,QB,MB,SB,SMB,LB,AC	**IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC 及常量； **OUT**：VW,T,C,IW,QW,SW,MW,SMW,LW,AC,AQW	**IN**：VD,ID,QD,MD,SD,SMD,LD,HC,AC 及常量； **OUT**：VD,ID,QD,MD,SD,SMD,LD,AC	**IN**：VD,ID,QD,MD,SD,SMD,LD,AC 及常量； **OUT**：VD,ID,QD,MD,SD,SMD,LD,AC
	数据类型：字节	**数据类型**：字、整数	**数据类型**：双字、双整数	**数据类型**：实数

续表

功能	使能输入有效时，即 EN＝1 时，将一个输入 IN 的字节、字/整数、双字/双整数或实数送到 OUT 指定的存储器输出。在传送过程中不改变数据的大小。传送后，输入存储器 IN 中的内容不变

提示 使 ENO＝0，即使能输出断开的错误条件是：SM4.3（运行时间）、0006（间接寻址错误）。

【例 5-1】 单个数据传送指令 MOV 程序举例。

（1）将数据 255 传送到 VB1 里面。程序如图 5-1 所示。

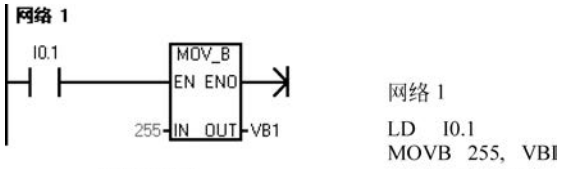


图 5-1 MOV_B 指令（例 5-1 题图）

设计分析：当 I0.1 接通时，MOV_B 指令将数据 255 传给 VB1，传送后，VB1＝255，此后，即使 I0.1 断开，VB1 里的数据保持 255 不变。

（2）将变量存储器 VW10 中的内容送到 VW100 中。程序如图 5-2 所示。

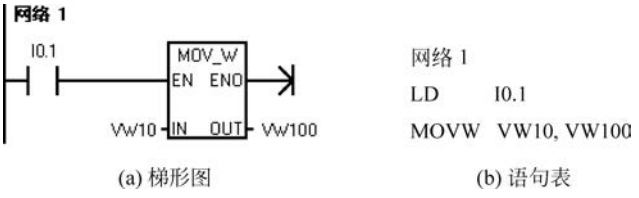


图 5-2 MOV_W 指令（例 5-1 题图）

（3）在 I0.1 控制开关导通时，将 VD100 中的双字数据传送到 VD200 中。程序如图 5-3 所示。

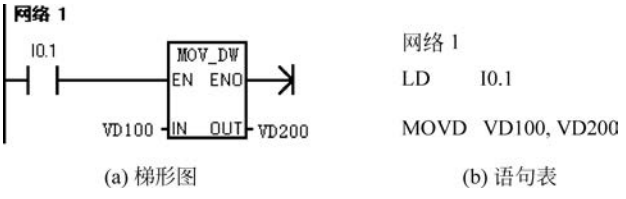


图 5-3 MOV_DW 指令（例 5-1 题图）

（4）在 I0.1 控制开关导通时，将常数 3.14 传送到双字单元 VD200 中。程序如图 5-4 所示。

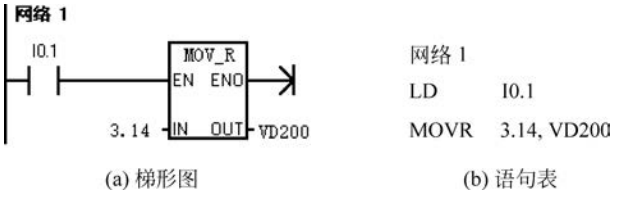
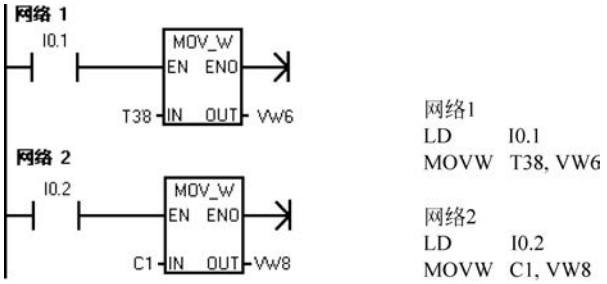


图 5-4 MOV_R 指令（例 5-1 题图）

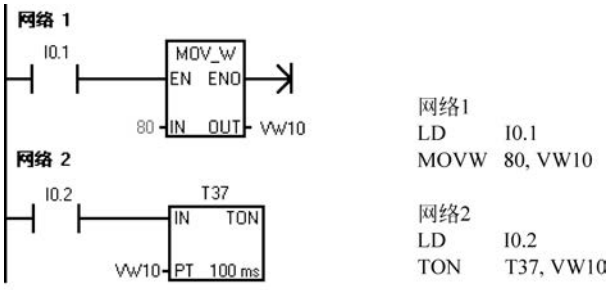
(5) 定时器及计数器当前值的读取。程序如图 5-5 所示。



(a) 梯形图　　(b) 语句表

图 5-5　定时器及计数器当前值的读取(例 5-1 题图)

(6) 定时器(计数器)设定值的间接指定。程序如图 5-6 所示。



(a) 梯形图　　(b) 语句表

图 5-6　定时器设定值的间接指定(例 5-1 题图)

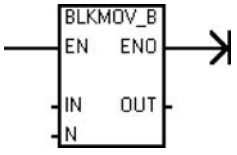
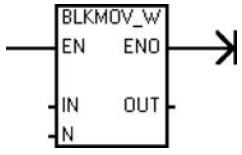
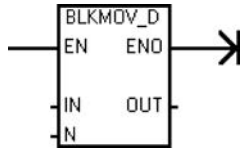
提示　由于定时器及计数器的数据类型都为整数型,因此使用传送指令时一定要用 MOV_W。

提示　功能指令涉及的数据类型多,编程时应保证操作数在合理范围内。S7-200 PLC 不支持完全数据类型检查。操作数的数据类型应与指令标识符相匹配。

2. 字节、字、双字、实数数据块传送(BLKMOV)指令

该类指令可用来进行一次多个(最多 255)数据的传送。数据块传送指令将从输入地址 IN 开始的 N 个数据传送到输出地址 OUT 开始的 N 个单元中,N 的范围为 1～255,N 的数据类型为字节。指令格式及功能如表 5-2 所示。

表 5-2　BLKMOV 指令格式

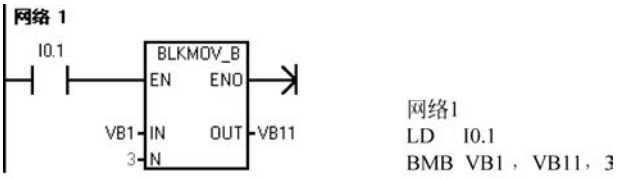
LAD			
STL	BMB　IN,OUT	BMW　IN,OUT	BMD　IN,OUT

续表

操作数及数据类型	**IN**：VB，IB，QB，MB，SB，SMB，LB； **OUT**：VB，IB，QB，MB，SB，SMB，LB； **数据类型**：字节	**IN**：VW，IW，QW，MW，SW，SMW，LW，T，C，AIW； **OUT**：VW，IW，QW，MW，SW，SMW，LW，T，C，AQW； **数据类型**：字	**IN/OUT**：VD，ID，QD，MD，SD，SMD，LD； **数据类型**：双字
	N：VB，IB，QB，MB，SB，SMB，LB，AC 及常量； **数据类型**：字节； **数据范围**：1～255		
功能	使能输入有效时，即 EN＝1 时，把从输入 IN 开始的 *N* 个字节（字、双字）传送到以输出 OUT 开始的 *N* 个字节（字、双字）中		

提示 使 ENO＝0 的错误条件：0006（间接寻址错误）、0091（操作数超出范围）。

【例 5-2】 块传送指令 BLKMOV 程序举例。将变量存储器 VB1 开始的 3 个字节（VB1～VB3）中的数据移至 VB11 开始的 3 个字节中（VB11～VB13）。程序如图 5-7 所示。



(a) 梯形图　　(b) 语句表

图 5-7　例 5-2 题图

5.1.2　字节交换、存储器填充与字节立即读写指令

1. 字节交换与存储器填充指令

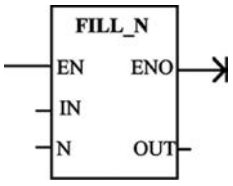
字节交换指令用来交换输入字 IN 的最高位字节和最低位字节，交换结果仍存在输入端（IN）指定的地址中。

存储器填充指令在 EN 端口执行条件存在时，用 IN 指定的输入值填充从 OUT 指定的存储单元开始的 *N* 个字的存储空间。多用于字数据存储区填充及对空间的清零。指令格式如表 5-3 所示。

表 5-3　字节交换指令使用格式及功能

LAD	STL	功能及说明
SWAP EN　ENO IN	SWAP IN	**功能**：使能输入 EN 有效时，将输入字 IN 的高字节与低字节交换，结果仍放在 IN 中； **IN**：VW，IW，QW，MW，SW，SMW，T，C，LW，AC； 数据类型：字

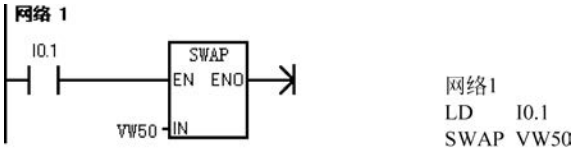
续表

LAD	STL	功能及说明
	FILL IN,OUT,N	**功能**：将字型输入数据从 OUT 开始的 *N* 个字存储单元中； **IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC,常数,＊VD,＊AC,＊LD； **OUT**：VW,IW,QW,MW,SW,SMW,LW,T,C,AQW,＊VD,＊AC,＊LD； **N**：VB,IB,QB,MB,SB,SMB,LB,AC,常数,＊VD,＊AC,＊LD； **数据类型**：IN、OUT 为字型,N 为字节型,取值范围为 1～255 的整数

提示　ENO＝0 的错误条件：0006(间接寻址错误)、SM4.3(运行时间)。

【例 5-3】　字节交换和存储器填充指令应用举例,如图 5-8～图 5-10 所示。

(1) 字节交换指令。

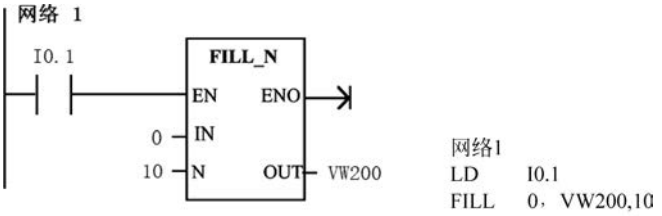


(a) 梯形图　　(b) 语句表

图 5-8　字节交换指令(例 5-3 题图)

分析：指令执行之前 VW50 中的字为 D6 C3；指令执行之后 VW50 中的字为 C3 D6。

(2) 存储器填充指令。

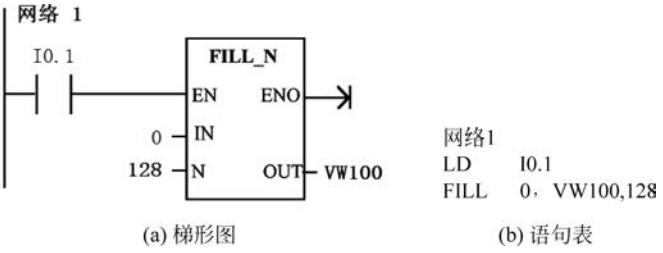


(a) 梯形图　　(b) 语句表

图 5-9　VM200～VM219 中全部清 0(例 5-3 题图)

分析：指令执行之后,VW200～VW219 中全部清 0。

另外,如果将 VW100 开始的 256 字节全部清 0。*N* 怎么给?



(a) 梯形图　　(b) 语句表

图 5-10　VM100 开始的 256 字节全部清 0(例 5-3 题图)

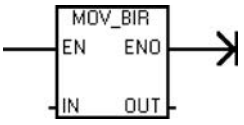
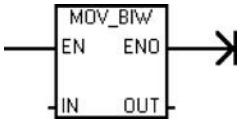
分析：在 I0.1 控制开关导通时，将 VW100 开始的 256 字节全部清 0。

2. 字节立即读写指令

字节立即读(MOV-BIR)指令在 EN 端口执行条件存在时，读取实际物理输入端 IN 给出的 1 字节的数值，并将结果写入 OUT 所指定的存储单元，但输入映像寄存器未更新。

字节立即写(MOV-BIW)指令在 EN 端口执行条件存在时，从输入 IN 所指定的存储单元中读取 1 字节的数值并写入实际输出 OUT 端的物理输出点，同时刷新对应的输出映像寄存器。指令格式及功能如表 5-4 所示。

表 5-4 字节立即读写指令格式

LAD	STL	功能及说明
	BIR IN,OUT	**功能**：字节立即读 **IN**：IB **OUT**：VB,IB,QB,MB,SB,SMB,LB,AC **数据类型**：字节
	BIW IN,OUT	**功能**：字节立即写 **IN**：VB,IB,QB,MB,SB,SMB,LB,AC,常量 **OUT**：QB **数据类型**：字节

提示 使 ENO=0 的错误条件：0006(间接寻址错误)、SM4.3(运行时间)。注意字节立即读写指令无法存取扩展模块。

5.1.3 移位指令

移位指令分为左、右移位和循环左、右移位及寄存器移位指令三大类。前两类移位指令按移位数据的长度又分字节型、字型、双字型 3 种。常用于顺序动作的控制。

1. 左、右移位指令

左、右移位数据存储单元与 SM1.1(溢出)端相连，移出位被放到特殊标志存储器 SM1.1 位。移位数据存储单元的另一端补 0。移位指令格式见表 5-5。

移位指令使用时应注意以下几点。

(1) 被移位的数据在字节操作时是无符号的；对于字和双字操作，当使用有符号数据类型时，符号位也将被移动。

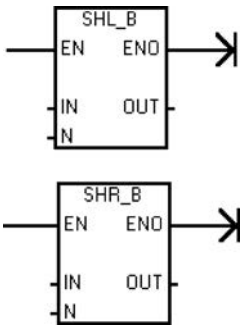
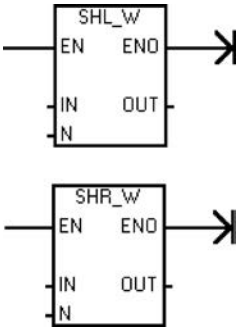
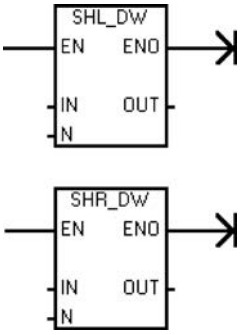
(2) 在移位时，存放被移位数据的编程元件的移出端与特殊继电器 SM1.1 相连，移出位送 SM1.1，另一端补 0。

(3) 移位次数 N 为字节型数据，它与移位数据的长度有关。如 N 小于实际的数据长度，则执行 N 次移位；如 N 大于数据长度，则执行移位的次数等于实际数据长度的位数。

(4) 左、右移位指令对特殊继电器的影响：结果为零置位 SM1.0，结果溢出置位 SM1.1。

(5) 运行时刻出现不正常状态则置位 SM4.3，ENO=0。

表 5-5　移位指令格式及功能

LAD			
STL	SLB　OUT,N SRB　OUT,N	SLW　OUT,N SRW　OUT,N	SLD　OUT,N SRD　OUT,N
操作数及数据类型	**IN**：VB,IB,QB,MB,SB,SMB,LB,AC 及常量； **OUT**：VB,IB,QB,MB,SB,SMB,LB,AC； **数据类型**：字节	**IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC 及常量； **OUT**：VW,IW,QW,MW,SW,SMW,LW,T,C,AC； **数据类型**：字	**IN**：VD,ID,QD,MD,SD,SMD,LD,AC,HC 及常量； **OUT**：VD,ID,QD,MD,SD,SMD,LD,AC； **数据类型**：双字
	N：VB,IB,QB,MB,SB,SMB,LB,AC 及常量； **数据类型**：字节； **数据范围**：$N \leqslant$数据类型(B、W、D)对应的位数		
功能	SHL：字节、字、双字左移 N 位；SHR：字节、字、双字右移 N 位		

(1) 左移位指令。

使能输入有效时，将输入 IN 的无符号数字节、字或双字中的各位向左移 N 位后(右端补 0)，将结果输出到 OUT 所指定的存储单元中。如果移位次数大于 0，则最后一次移出位保存在“溢出”存储器位 SM1.1。如果移位结果为 0，则零标志位 SM1.0 置 1。

(2) 右移位指令。

使能输入有效时，将输入 IN 的无符号数字节、字或双字中的各位向右移 N 位后，将结果输出到 OUT 所指定的存储单元中，移出位补 0，最后一移出位保存在 SM1.1。如果移位结果为 0，零标志位 SM1.0 置 1。

(3) 使 ENO=0 的错误条件：0006(间接寻址错误)、SM4.3(运行时间)。

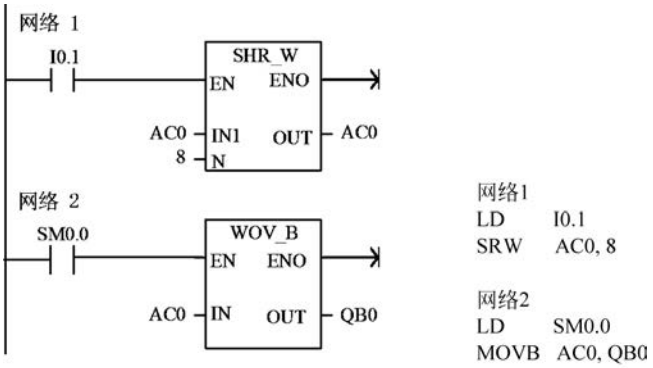
【例 5-4】　移位指令程序应用举例。将 AC0 字数据的高 8 位右移到低 8 位，输出给 QB0。程序如图 5-11 所示。

提示　在 STL 指令中，若 IN 和 OUT 指定的存储器不同，则须首先使用数据传送指令 MOV 将 IN 中的数据送入 OUT 所指定的存储单元。如：

```
MOVB IN,OUT
SLB OUT,N
```

2. 循环左、右移位指令

循环移位将移位数据存储单元的首尾相连，同时又与溢出标志 SM1.1 连接，SM1.1 用来存放被移出的位。指令格式见表 5-6。



(a) 梯形图　　(b) 语句表

图 5-11　例 5-4 题图

表 5-6　循环左、右移位指令格式及功能

LAD	ROL_B / ROR_B	ROL_W / ROR_W	ROL_DW / ROR_DW
STL	RLB　OUT,N RRB　OUT,N	RLW　OUT,N RRW　OUT,N	RLD　OUT,N RRD　OUT,N
操作数及数据类型	**IN**：VB，IB，QB，MB，SB，SMB，LB，AC 及常量； **OUT**：VB，IB，QB，MB，SB，SMB，LB，AC； **数据类型**：字节	**IN**：VW，IW，QW，MW，SW，SMW，LW，T，C，AIW，AC 及常量； **OUT**：VW，IW，QW，MW，SW，SMW，LW，T，C，AC； **数据类型**：字	**IN**：VD，ID，QD，MD，SD，SMD，LD，AC，HC 及常量； **OUT**：VD，ID，QD，MD，SD，SMD，LD，AC； **数据类型**：双字
	N：VB，IB，QB，MB，SB，SMB，LB，AC 及常量； **数据类型**：字节		
功能	ROL：字节、字、双字循环左移 *N* 位；ROR：字节、字、双字循环右移 *N* 位		

(1) 循环左移位指令 ROL。

使能输入有效时，将 IN 输入无符号数(字节、字或双字)循环左移 *N* 位后，将结果输出到 OUT 所指定的存储单元中，移出的最后一位的数值送溢出标志位 SM1.1。当需要移位的数值是零时，零标志位 SM1.0 为 1。

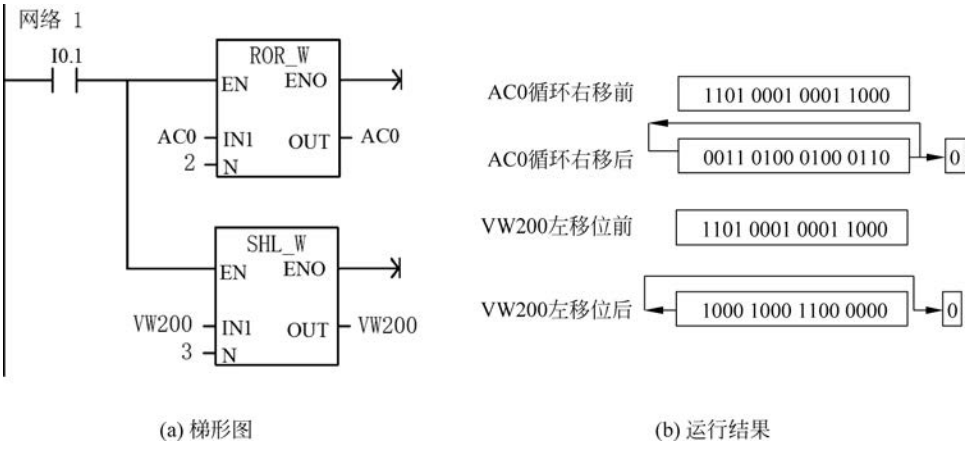
(2) 循环右移位指令 ROR。

使能输入有效时，将 IN 输入无符号数(字节、字或双字)循环右移 *N* 位后，将结果输出到 OUT 所指定的存储单元中，移出的最后一位的数值送溢出标志位 SM1.1。当需要移位

的数值是零时，零标志位 SM1.0 为 1。

提示　移位次数 $N \geqslant$ 数据类型(B、W、D)时：如果操作数是字节，当移位次数 $N \geqslant 8$ 时，则在执行循环移位前，先对 N 进行模 8 操作(N 除以 8 后取余数)，其结果 0～7 为实际移动位数；如果操作数是字，当移位次数 $N \geqslant 16$ 时，则在执行循环移位前，先对 N 进行模 16 操作(N 除以 16 后取余数)，其结果 0～15 为实际移动位数；如果操作数是双字，当移位次数 $N \geqslant 32$ 时，则在执行循环移位前，先对 N 进行模 32 操作(N 除以 32 后取余数)，其结果 0～31 为实际移动位数。使 ENO＝0 的错误条件：0006(间接寻址错误)，SM4.3(运行时间)。

【例 5-5】　移位指令程序应用举例。将 AC0 中的字循环右移 2 位，将 VW200 中的字左移 3 位。程序及运行结果如图 5-12 所示。



(a) 梯形图　　(b) 运行结果

图 5-12　例 5-5 题图

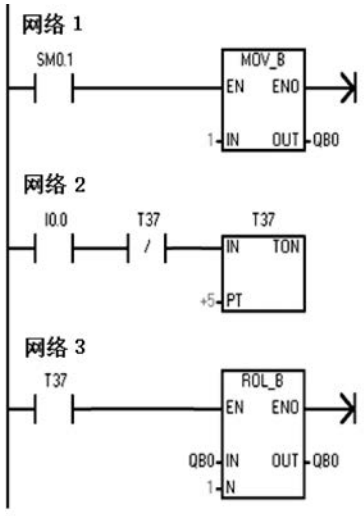
【例 5-6】　移位指令程序应用举例。

(1) 用 I0.0 控制接在 Q0.0～Q0.7 上的 8 个彩灯循环移位，从左到右以 0.5s 的时间间隔依次点亮，保持任意时刻只有一个指示灯亮，到达最右端后，再从左到右依次点亮。

设计分析：8 个彩灯循环移位控制，可以用字节的循环移位指令。根据控制要求，首先应置彩灯的初始状态为 QB0＝1，即左边第一盏灯亮；接着灯从左到右以 0.5s 的时间间隔依次点亮，即要求字节 QB0 中的 1 用循环左移位指令每 0.5s 移动一位，因此须在 ROL-B 指令的 EN 端接一个 0.5s 的移位脉冲(用定时器指令实现)。梯形图程序和语句表程序如图 5-13 所示。

(2) 用 I0.0 控制 16 个彩灯循环移位，从左到右以 2s 的时间间隔依次 2 个为一组点亮；保持任意时刻只有 2 个灯亮，到达最右端后，再依次点亮，按下 I0.1 后，彩灯循环停止。

设计分析：16 个彩灯分别接 Q0.0～Q1.7，可以用字的循环移位指令，进行循环移位控制。根据控制要求，首先应置彩灯的初始状态为 QW0＝3，即左边第 1、2 盏灯亮；接着灯从左到右以 2s 的时间间隔依次点亮，即要求字节 QW0 中的 11 用循环左移位指令每 2s 移动两位，因此须在 ROL-W 指令的 EN 端接一个 2s 的移位脉冲。梯形图程序和语句表程序如图 5-14 所示。



(a) 梯形图

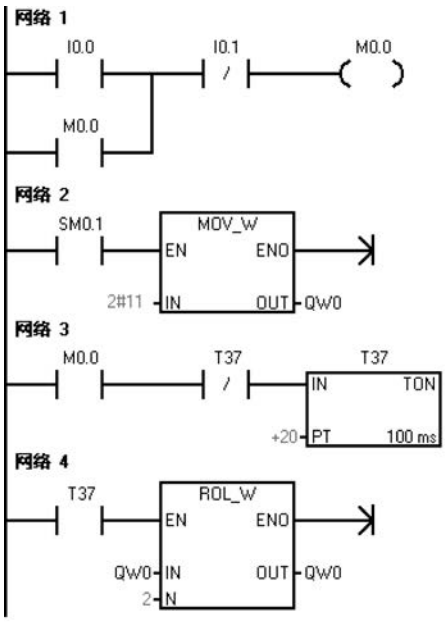
```
网络1
LD      SM0.1     //首次扫描时
MOVB    1, QB0    //置8位彩灯初态

网络2
LD      I0.0      //T37产生周期为
AN      T37       //0.5s的移位脉冲
TON     T37, +5

网络3
LD      T37       //每来一个脉冲
RLB     QB0, 1    //彩灯循环左移1位
```

(b) 语句表

图 5-13　例 5-6 题图(1)



(a) 梯形图

```
网络1   起动停止标志
LD      I0.0
O       M0.0
AN      I0.1
=       M0.0

网络2   首次扫描置彩灯初态
LD      SM0.1
MOVW    2#11, QW0

网络3   T37产生周期为2s的移位脉冲
LD      M0.0
AN      T37
TON     T37, +20

网络4   每来一个脉冲彩灯循环左移2位
LD      T37
RLW     QW0, 2
```

(b) 语句表

图 5-14　例 5-6 题图(2)

3. 移位寄存器指令

移位寄存器指令是可以指定移位寄存器的长度和移位方向的移位指令，其指令格式如表 5-7 所示。

在梯形图中，EN 为使能输入端，连接移位脉冲信号，每次使能有效时，整个移位寄存器移动 1 位。

表 5-7　移位寄存器指令指令格式

LAD	STL	说　明
SHRB EN　ENO DATA S_BIT N	SHRB DATA,S_BIT,N	**DATA 和 S_BIT**：I,Q,M,SM,T,C,V,S,L,数据类型为 BOOL 变量； **N**：VB,IB,QB,MB,SB,SMB,LB,AC 及常量,数据类型为字节

移位寄存器指令 SHRB 将 DATA 数值移入移位寄存器,并进行移位。DATA 为数据输入端,连接移入移位寄存器的二进制数值,执行指令时将该位的值移入寄存器。

移位寄存器是由 S_BIT 和 *N* 决定的。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,移位寄存器的最大长度为 64 位,*N* 为正值表示左移位,输入数据(DATA)移入移位寄存器的最低位(S_BIT),并移出移位寄存器的最高位。移出的数据被放置在溢出内存位(SM1.1)中。*N* 为负值表示右移位,输入数据移入移位寄存器的最高位中,并移出最低位(S_BIT)。移出的数据被放置在溢出内存位(SM1.1)中。

提示　使 ENO=0 的错误条件：0006(间接地址)、0091(操作数超出范围)、0092(计数区错误)。移位指令影响特殊内部标志位：SM1.1(为移出的位值设置溢出位)。

【例 5-7】 移位寄存器指令程序举例。在输入触点 I0.1 的上升沿,从 VB100 的低 4 位(自定义移位寄存器)由低向高移位,I0.2 移入最低位,其梯形图、时序图如图 5-15 所示。

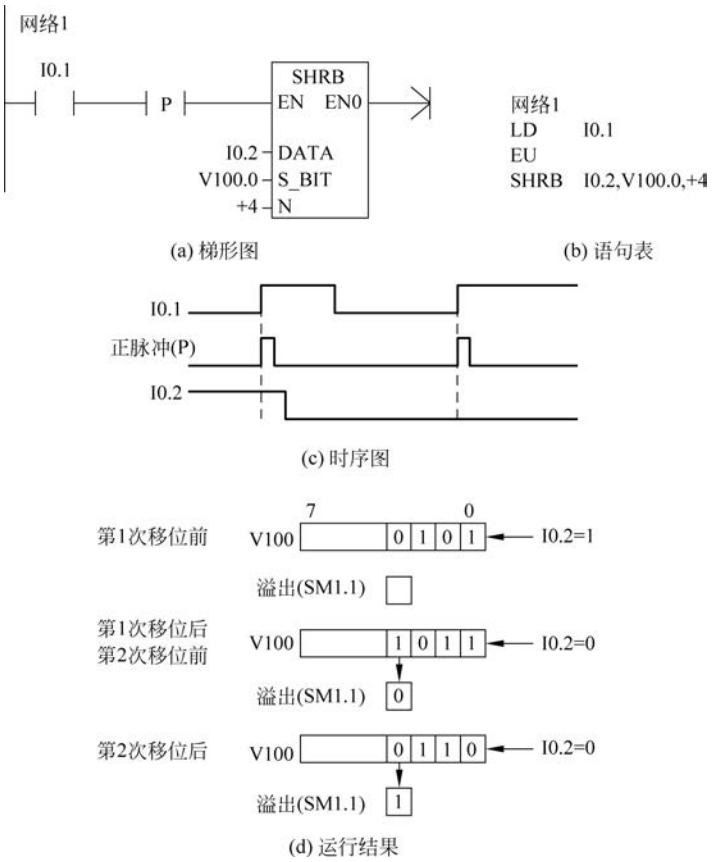


图 5-15　例 5-7 题图

设计分析：建立移位寄存器的位范围为 V100.0～V100.3，长度 $N=+4$。在 I0.1 的上升沿，移位寄存器由低位向高位移位，最高位移至 SM1.1，最低位由 I0.2 移入。移位寄存器指令对特殊继电器影响为结果为零置位 SM1.0、溢出置位 SM1.1；运行时刻出现不正常状态置位 SM4.3，ENO=0。

【例 5-8】 用 PLC 实现模拟喷泉的控制。用灯 L1～L12 分别代表喷泉的 12 个喷水注。

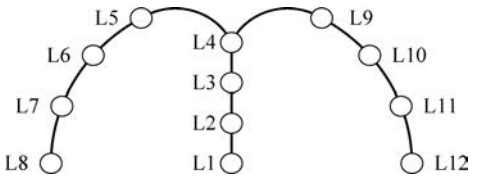


图 5-16 喷泉控制示意图(例 5-8 题图)

控制要求：按下起动按钮后，隔灯闪烁，L1 亮 0.5s 后灭，接着 L2 亮 0.5s 后灭，接着 L3 亮 0.5s 后灭，接着 L4 亮 0.5s 后灭，接着 L5、L9 亮 0.5s 后灭，接着 L6、L10 亮 0.5s 后灭，接着 L7、L11 亮 0.5s 后灭，接着 L8、L12 亮 0.5s 后灭，L1 亮 0.5s 后灭，如此循环下去，直至按下停止按钮，如图 5-16 所示。

设计分析：

(1) I/O 分配。

输入	输出	
起动按钮：I0.0；	L1：Q0.0；	L5、L9：Q0.4；
停止按钮：I0.1；	L2：Q0.1；	L6、L10：Q0.5；
	L3：Q0.2；	L7、L11：Q0.6；
	L4：Q0.3；	L8、L12：Q0.7

(2) 梯形图程序。

利用移位寄存器实现模拟控制。移位寄存器的位与输出对应关系设置：根据喷泉模拟控制的 8 位输出(Q0.0～Q0.7)，须指定一个 8 位的移位寄存器(M10.1～M11.0)，移位寄存器的 S-BIT 位为 M10.1，并且移位寄存器的每一位对应一个输出。在移位寄存器指令中，EN 连接移位脉冲，每来一个脉冲的上升沿，移位寄存器移动一位。移位寄存器应 0.5s 移一位，因此需要设计一个 0.5s 产生一个脉冲的脉冲发生器(由 T38 构成)，如图 5-17 所示。

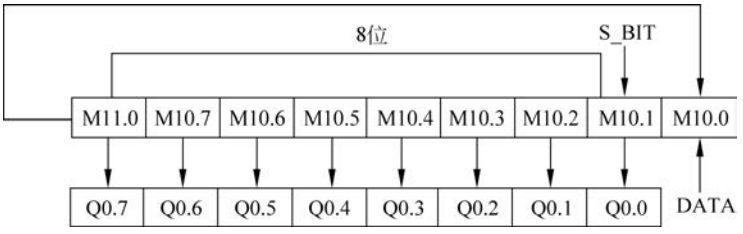


图 5-17 移位寄存器的位与输出对应关系图(例 5-8 题图)

M10.0 为数据输入端 DATA，根据控制要求，每次只有一个输出，因此只需要在第 1 个移位脉冲到来时由 M10.0 送入移位寄存器 S_BIT 位(M10.1)一个 1，第 2～8 个脉冲到来时由 M10.0 送入 M10.1 的值均为 0，这里时间继电器 T38 构成 0.5s 产生一个机器扫描周期脉冲的脉冲发生器，如图 5-18 所示。

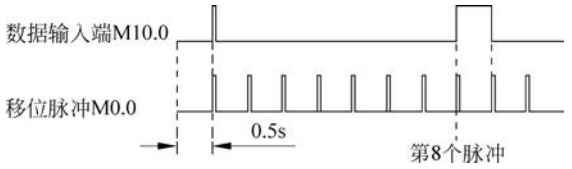


图 5-18　喷泉模拟控制移位脉冲时序图(例 5-8 题图)

在程序中由定时器 T37 延时 0.5s 导通一个扫描周期实现,第 8 个脉冲到来时 M11.0 置位为 1,同时通过与 T37 并联的 M11.0 常开触点使 M10.0 置位为 1,在第 9 个脉冲到来时由 M10.0 送入 M10.1 的值又为 1,如此循环下去,直至按下停止按钮。按下停止按钮(I0.1),触发复位指令,使 M10.1～M11.0 的 8 位全部复位。梯形图程序如图 5-19 所示。

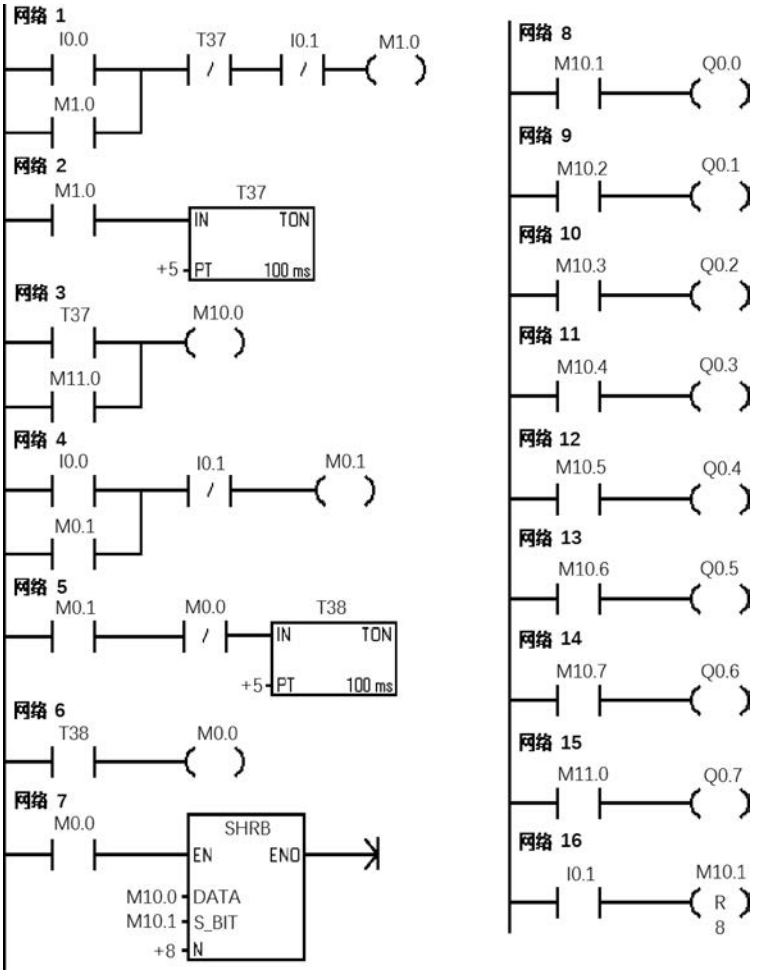


图 5-19　喷泉模拟控制梯形图(例 5-8 题图)

5.1.4　转换指令

在实际控制过程中,经常要对不同类型的数据进行运算,数据运算指令中要求参与运算的数值为同一类型,为了实现数据处理时的数据匹配,所以要对数据格式进行转换。

转换指令是指对操作数的不同类型进行转换，并输出到指定目标地址中。转换指令包括数据的类型转换、数据的编码和译码指令以及字符串类型转换指令。

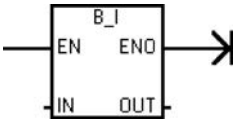
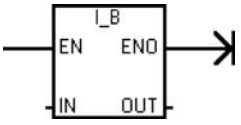
不同功能的指令对操作数要求不同。类型转换指令可实现字节与字整数之间的转换、整数与双整数的转换、双字整数与实数之间的转换、BCD码与整数之间的转换等。

在S7-200中，转换指令是指对操作数的不同类型及编码进行相互转换的操作，以满足程序设计的需要。

1. 字节与字整数之间的转换

字节与字整数之间转换的转换格式、功能及说明如表5-8所示。

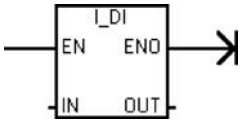
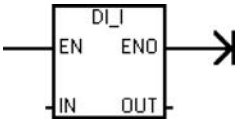
表5-8 字节与字整数之间的转换指令

LAD		
STL	BTI IN,OUT	ITB IN,OUT
操作数及数据类型	**IN**：VB,IB,QB,MB,SB,SMB,LB,AC及常量，数据类型为字节； **OUT**：VW,IW,QW,MW,SW,SMW,LW,T,C,AC，数据类型为整数	**IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC及常量，数据类型为整数； **OUT**：VB,IB,QB,MB,SB,SMB,LB,AC，数据类型为字节
功能及说明	BTI指令将字节数值(IN)转换成整数值，并将结果置入OUT指定的存储单元。因为字节不带符号，所以无符号扩展	ITB指令将字整数(IN)转换成字节，并将结果置入OUT指定的存储单元。输入的字整数0～255被转换。超出部分导致溢出，SM1.1=1。输出不受影响
ENO=0的错误条件	0006 间接地址； SM4.3 运行时间	0006 间接地址； SM1.1 溢出或非法数值； SM4.3 运行时间

2. 字整数与双字整数之间的转换

字整数与双字整数之间的转换格式、功能及说明如表5-9所示。

表5-9 字整数与双字整数之间的转换指令

LAD		
STL	ITD IN,OUT	DTI IN,OUT
操作数及数据类型	**IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC及常量； **数据类型**：整数； **OUT**：VD,ID,QD,MD,SD,SMD,LD,AC； **数据类型**：双整数	**IN**：VD,ID,QD,MD,SD,SMD,LD,HC,AC及常量； **数据类型**：双整数； **OUT**：VW,IW,QW,MW,SW,SMW,LW,T,C,AC； **数据类型**：整数

续表

功能及说明	ITD 指令将整数值(IN)转换成双整数值，并将结果置入 OUT 指定的存储单元。符号被扩展	DTI 指令将双整数值(IN)转换成整数值，并将结果置入 OUT 指定的存储单元。如果转换的数值过大，则无法在输出中表示，产生溢出 SM1.1＝1，输出不受影响
ENO＝0的错误条件	0006　间接地址； SM4.3　运行时间	0006　间接地址； SM1.1　溢出或非法数值； SM4.3　运行时间

3. 双整数与实数之间的转换

双整数与实数之间进行转换的转换格式、功能及说明如表 5-10 所示。

表 5-10　双整数与实数之间的转换指令

LAD	DI_R EN　ENO IN　OUT	ROUND EN　ENO IN　OUT	TRUNC EN　ENO IN　OUT
STL	DTR　IN,OUT	ROUND　IN,OUT	TRUNC　IN,OUT
操作数及数据类型	**IN**：VD，ID，QD，MD，SD，SMD，LD，HC，AC 及常量，数据类型为双整数； **OUT**：VD，ID，QD，MD，SD，SMD，LD，AC，数据类型为实数	**IN**：VD，ID，QD，MD，SD，SMD，LD，AC 及常量，数据类型为实数； **OUT**：VD，ID，QD，MD，SD，SMD，LD，AC，数据类型为双整数	**IN**：VD，ID，QD，MD，SD，SMD，LD，AC 及常量，数据类型为实数； **OUT**：VD，ID，QD，MD，SD，SMD，LD，AC，数据类型为双整数
功能及说明	DTR 指令将 32 位带符号整数 IN 转换成 32 位实数，并将结果置入 OUT 指定的存储单元	ROUND 指令按小数部分四舍五入的原则，将实数(IN)转换成双整数值，并将结果置入 OUT 指定的存储单元	TRUNC(截位取整)指令按将小数部分直接舍去的原则，将 32 位实数(IN)转换成 32 位双整数，并将结果置入 OUT 指定存储单元
ENO＝0的错误条件	0006　间接地址； SM4.3　运行时间	0006　间接地址； SM1.1　溢出或非法数值； SM4.3　运行时间	0006　间接地址； SM1.1　溢出或非法数值； SM4.3　运行时间

提示　不论是四舍五入取整，还是截位取整，如果转换的实数数值过大，无法在输出中表示，则产生溢出，即影响溢出标志位，使 SM1.1＝1，输出不受影响。

【例 5-9】　双整数与实数之间的转换应用举例。将 VW10 中的整数 100 和 VD100 中的实数 190.5 相加。梯形图和语句表如图 5-20 所示。

【例 5-10】　整数、双整数与实数之间的转换应用举例。要求：将 in 转换为 cm，已知 VW100 的当前值为 in 的计数值，1in＝2.54cm。

设计分析：VW100 中的整数值 in→双整数 in→实数 in→实数 cm→整数 cm。梯形图和语句表如图 5-21 所示。

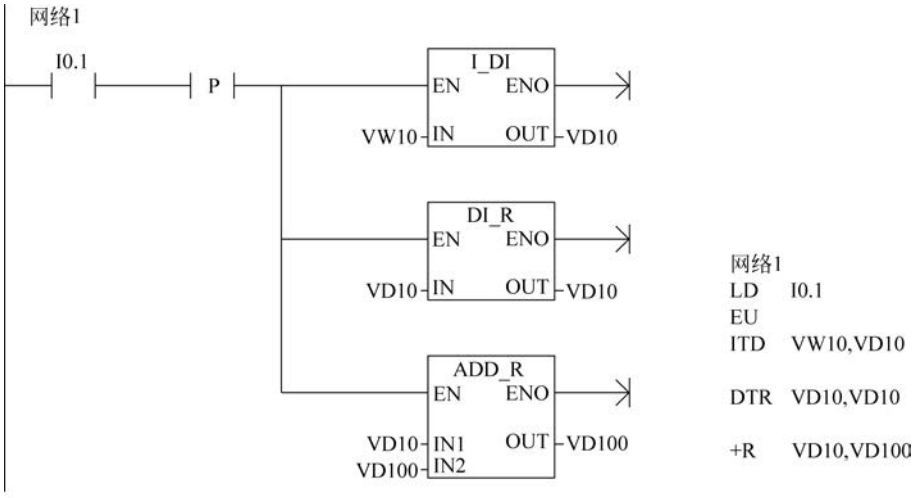


图 5-20　例 5-9 题图

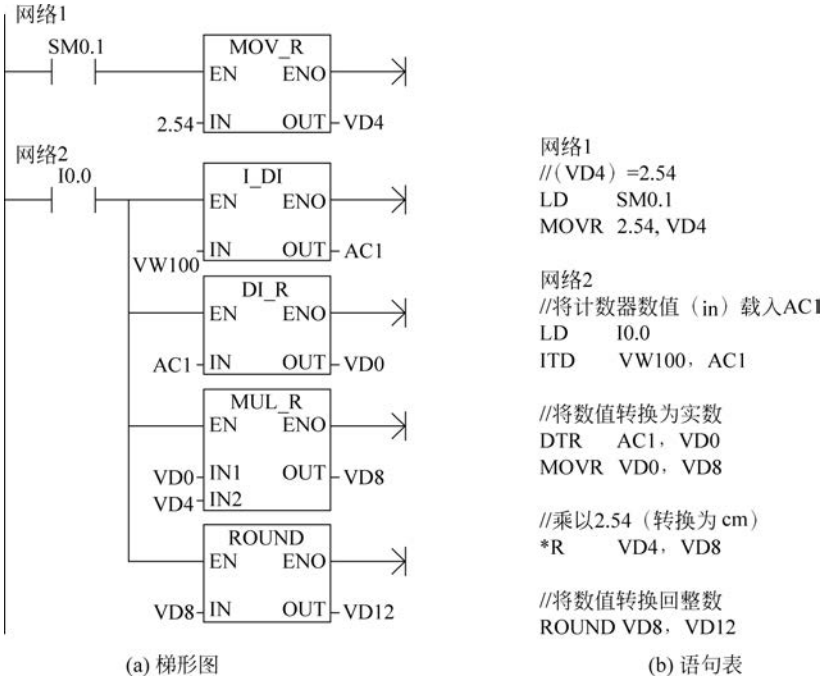
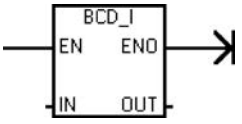
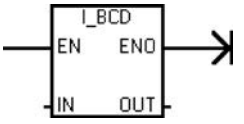


图 5-21　例 5-10 题图

4. BCD 码与整数的转换

BCD 码与整数之间进行转换的指令格式、功能及说明如表 5-11 所示。

表 5-11　BCD 码与整数之间的转换指令

LAD		
STL	BCDI　OUT	IBCD　OUT
操作数及数据类型	**IN**：VW,IW,QW,MW,SW,SMW,LW,T,C,AIW,AC 及常量； **OUT**：VW,IW,QW,MW,SW,SMW,LW,T,C,AC； **IN/OUT 数据类型**：字	
功能及说明	BCD_I 指令将二进制编码的十进制数 IN 转换成整数，并将结果送入 OUT 指定的存储单元。IN 的有效范围是 BCD 码 0～9999	I_BCD 指令将输入整数 IN 转换成二进制编码的十进制数，并将结果送入 OUT 指定的存储单元。IN 的有效范围是 0～9999
ENO=0 的错误条件	0006：间接地址；SM1.6：无效 BCD 数值；SM4.3：运行时间	

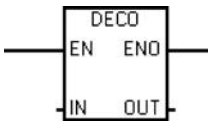
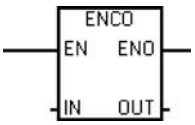
提示　①数据长度为字的 BCD 格式的有效范围为：0～9999(十进制)，0000～9999(十六进制)，0000 0000 0000 0000～1001 1001 1001 1001(BCD 码)。②指令影响特殊标志位 SM1.6(无效 BCD)。③在表 5-11 的 LAD 和 STL 指令中，IN 和 OUT 的操作数地址相同。若 IN 和 OUT 操作数地址不是同一个存储器，则对应的语句表指令为

```
MOV　IN　OUT
BCDI　OUT
```

5. 译码和编码指令

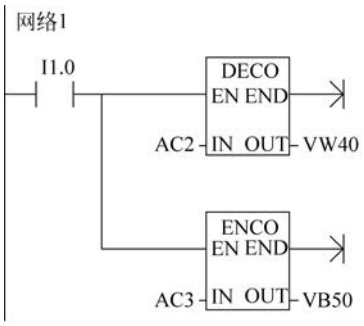
译码和编码指令的格式和功能如表 5-12 所示。

表 5-12　译码和编码指令的格式和功能

LAD		
STL	DECO IN,OUT	ENCO IN,OUT
操作数及数据类型	**IN**：VB，IB，QB，MB，SMB，LB，SB，AC，常量，数据类型为字节； **OUT**：VW，IW，QW，MW，SMW，LW，SW，AQW，T，C，AC，数据类型为字	**IN**：VW，IW，QW，MW，SMW，LW，SW，AIW，T，C，AC，常量，数据类型为字； **OUT**：VB，IB，QB，MB，SMB，LB，SB，AC，数据类型为字节
功能及说明	译码指令根据输入字节(IN)的低 4 位表示的输出字的位号，将输出字相对应的位置位为 1，输出字的其他位均置位为 0	编码指令将输入字(IN)最低有效位(其值为 1)的位号写入输出字节(OUT)的低 4 位中
ENO=0 的错误条件	0006　间接地址；SM4.3　运行时间	

【例 5-11】 译码编码指令程序应用举例。

设计分析：若(AC2)＝2，执行译码指令，则将输出字 VW40 的第 2 位置 1，VW40 中的二进制数为 2＃0000 0000 0000 0100；若(AC3)＝2＃0000 0000 0000 0100，执行编码指令，则输出字节 VB50 中的错误码为 2。梯形图和语句表如图 5-22 所示。



(a) 梯形图

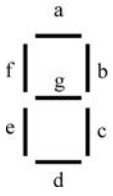
```
网络1
LD      I1.0
DECO    AC2, VW40   //译码
ENCO    AC3, VB50   //编码
```

(b) 语句表

图 5-22 译码编码指令程序应用举例(例 5-11 题图)

6. 七段显示译码指令

七段显示器的 abcdefg 段分别对应于字节的第 0～6 位，字节的某位为 1 时，其对应的段亮；输出字节的某位为 0 时，其对应的段暗。将字节的第 7 位补 0，则构成与七段显示器相对应的 8 位编码，称为七段显示码。数字 0～9、字母 A～F 与七段显示码的对应如图 5-23 所示。



IN	段显示	(OUT) -gfe dcba
0	0	0001 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

IN	段显示	(OUT) -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 5-23 与七段显示码对应的代码

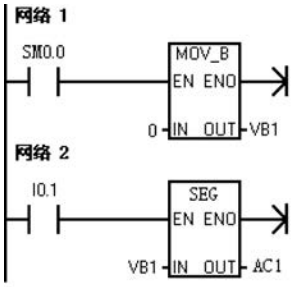
七段译码指令 SEG 将输入字节 16＃0～F 转换成七段显示码。指令格式如表 5-13 所示。

表 5-13 七段显示译码指令

LAD	STL	功能及操作数
SEG EN ENO IN OUT	SEG IN，OUT	**功能**：将输入字节(IN)的低 4 位确定的十六进制数(十六＃0～F)产生相应的七段显示码，送入输出字节 OUT； **IN**：VB，IB，QB，MB，SB，SMB，LB，AC，常量； **OUT**：VB，IB，QB，MB，SMB，LB，AC； **IN/OUT 数据类型**：字节

提示　使 ENO=0 的错误条件：0006(间接地址)、SM4.3(运行时间)。

【例 5-12】　编写实现用七段显示码显示数字 0 的程序。梯形图和语句表如图 5-24 所示。



(a) 梯形图

```
网络1
LD      SM0.0
MOVB    0,VB1

网络2
LD      I0.1
SEG     VB1, AC1
```

(b) 语句表

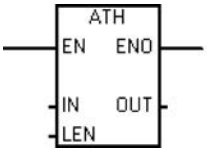
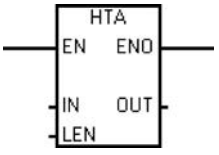
图 5-24　例 5-12 题图

设计分析：程序运行结果为 AC1 中的值为 16#3F(2#0011 1111)。

7. ASCII 码与十六进制数之间的转换指令

ASCII 码与十六进制数之间的转换指令的格式和功能如表 5-14 所示。

表 5-14　ASCII 码与十六进制数之间的转换指令的格式和功能

LAD		
STL	ATH IN,OUT,LEN	HTA IN,OUT,LEN
操作数及数据类型	**IN/OUT**：VB,IB,QB,MB,SB,SMB,LB,数据类型为字节； **LEN**：VB,IB,QB,MB,SB,SMB,LB,AC 及常量;数据类型为字节,最大值为 255	
功能及说明	ASCII 至 HEX(ATH)指令将从 IN 开始的长度为 LEN 的 ASCII 字符转换成十六进制数,放入从 OUT 开始的存储单元	HEX 至 ASCII (HTA)指令将从输入字节(IN)开始的长度为 LEN 的十六进制数转换成 ASCII 字符,放入从 OUT 开始的存储单元
ENO=0 的错误条件	0006　间接地址；SM4.3　运行时间；0091　操作数范围超界； SM1.7　非法 ASCII 数值(仅限 ATH)	

提示　合法的 ASCII 码对应的十六进制数包括 30H～39H、41H～46H。如果在 ATH 指令的输入中包含非法的 ASCII 码,则终止转换操作,特殊内部标志位 SM1.7 置位为 1。

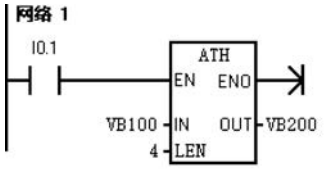
【例 5-13】　编程将 VB100～VB103 中存储的 4 个 ASCII 码转换成十六进制数。已知(VB100)= 33,(VB101)=32,(VB102)=41,(VB103)=45。梯形图和语句表如图 5-25 所示。

设计分析：程序运行结果如下。

执行前：(VB100)=33,(VB101)=32,(VB102)=41,(VB103)=45。

执行后：(VB200)=32,(VB201)=AE。

可见将 VB100～VB103 中存放的 4 个 ASCII 码 33、32、41、45 转换成了十六进制数 32 和 AE，放在 VB200 和 VB201 中。



(a) 梯形图

```
网络1
LD    I0.1
ATH   VB100, VB200, 4
```

(b) 语句表

图 5-25　例 5-13 题图

5.2　算术运算、逻辑运算指令

随着控制领域中新型控制算法的出现和复杂控制对控制器计算能力的要求，新型 PLC 中普遍增加了较强的计算功能。数据运算指令分为算术运算和逻辑运算两大类。

算术运算指令包括加、减、乘、除运算和数学函数变换，逻辑运算指令包括逻辑与或非指令等。

5.2.1　算术运算指令

1. 整数与双整数加减法指令

整数加法(ADD-I)和减法(SUB-I)指令是使能输入有效时，将两个 16 位符号整数相加或相减，并产生一个 16 位的结果输出到 OUT。

双整数加法(ADD-D)和减法(SUB-D)指令是使能输入有效时，将两个 32 位符号整数相加或相减，并产生一个 32 位结果输出到 OUT。

整数与双整数加减法指令格式如表 5-15 所示。

表 5-15　整数与双整数加减法指令格式

LAD	ADD_I EN ENO IN1 OUT IN2	SUB_I EN ENO IN1 OUT IN2	ADD_DI EN ENO IN1 OUT IN2	SUB_DI EN ENO IN1 OUT IN2
STL	MOVW IN1,OUT +I IN2,OUT	MOVW IN1,OUT −I IN2,OUT	MOVD IN1,OUT +D IN2,OUT	MOVD IN1,OUT −D IN2,OUT
功能	IN1＋IN2＝OUT	IN1－IN2＝OUT	IN1＋IN2＝OUT	IN1－IN2＝OUT
操作数及数据类型	**IN1/IN2**：VW，IW，QW，MW，SW，SMW，T，C，AC，LW，AIW，常量，＊VD，＊LD，＊AC； **OUT**：VW，IW，QW，MW，SW，SMW，T，C，LW，AC，＊VD，＊LD，＊AC； **数据类型**：整数		**IN1/IN2**：VD，ID，QD，MD，SMD，SD，LD，AC，HC，常量，＊VD，＊LD，＊AC； **OUT**：VD，ID，QD，MD，SMD，SD，LD，AC，＊VD，＊LD，＊AC； **数据类型**：双整数	
ENO=0 的错误条件	0006　间接地址；SM4.3　运行时间；SM1.1　溢出			

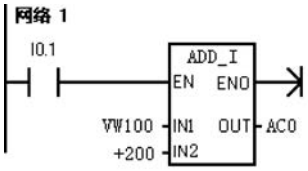
注意：

（1）当 IN1、IN2 和 OUT 操作数的地址不同时，在 STL 指令中，首先用数据传送指令将 IN1 中的数值送入 OUT，然后再执行加、减运算，即 OUT＋IN2＝OUT，OUT－IN2＝OUT。为了节省内存，在整数加法的梯形图指令中，可以指定 IN1 或 IN2＝OUT，这样可以不用数据传送指令。如指定 IN1＝OUT，则语句表指令为：＋I　IN2，OUT；如指定 IN2＝OUT，则语句表指令为：＋I　IN1，OUT。在整数减法的梯形图指令中，可以指定 IN1＝OUT，则语句表指令为：－I　IN2，OUT。这个原则适用于所有的算术运算指令，且乘法和加法对应，减法和除法对应。

（2）整数与双整数加减法指令影响算术标志位 SM1.0（零标志位）、SM1.1（溢出标志位）和 SM1.2（负数标志位）。

提示　在梯形图编程和指令表编程时对存储单元的要求是不同的，所以在使用时一定要注意存储单元的分配。梯形图编程时，IN2 和 OUT 指定的存储单元可以相同也可以不同；指令表编程时，IN2 和 OUT 要使用相同的存储单元。

【例 5-14】 求 300 加 200 的和，300 在数据存储器 VW100 中，结果放入 AC0。梯形图和语句表如图 5-26 所示。



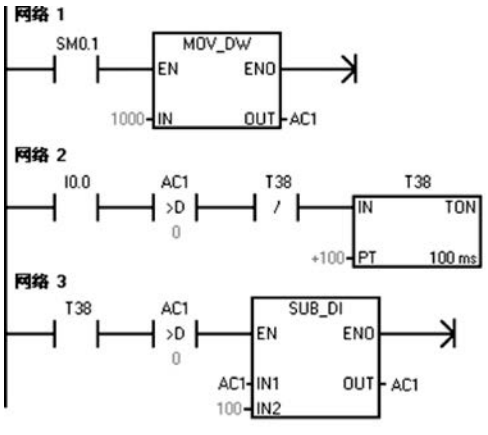
(a) 梯形图

```
网络1
LD      I0.1
MOVW    VW100, AC0   // VW100→AC0
+I      +200, AC0    // VW100+200=AC0
```

(b) 语句表

图 5-26　例 5-14 题图

【例 5-15】 在程序初始化时，设 AC1 为 1000，合上 I0.0 开关，AC1 的值每隔 10s 减 100，一直减到 0 为止。梯形图和语句表如图 5-27 所示。



(a) 梯形图

```
网络1  程序初始化
LD      SM0.1
MOVD    1000, AC1
网络2  启动定时
LD      I0.0
A       D>AC1,0
AN      T38
TON     T38,100
网络3  AC1减100
LD      T38
A       D>AC1,0
-D      100,AC1
```

(b) 语句表

图 5-27　例 5-15 题图

2. 整数乘除法指令

整数乘法(MUL-I)指令是在使能输入有效时，将两个16位符号整数相乘，并产生一个16位积，从OUT指定的存储单元输出。

整数除法(DIV-I)指令是在使能输入有效时，将两个16位符号整数相除，并产生一个16位商，从OUT指定的存储单元输出，不保留余数。如果输出结果大于一个字，则溢出位SM1.1置位为1。

双整数乘法(MUL-D)指令是在使能输入有效时，将两个32位符号整数相乘，并产生一个32位乘积，从OUT指定的存储单元输出。

双整数除法(DIV-D)指令是在使能输入有效时，将两个32位整数相除，并产生一个32位商，从OUT指定的存储单元输出，不保留余数。

整数乘法产生双整数(MUL)指令是在使能输入有效时，将两个16位整数相乘，得出一个32位乘积，从OUT指定的存储单元输出。

整数除法产生双整数(DIV)指令是在使能输入有效时，将两个16位整数相除，得出一个32位结果，从OUT指定的存储单元输出。其中高16位放余数，低16位放商。

整数乘除法指令格式如表5-16所示。

表5-16 整数乘除法指令格式

LAD	MUL_I EN ENO IN1 OUT IN2	DIV_I EN ENO IN1 OUT IN2	MUL_DI EN ENO IN1 OUT IN2	MUL_DI EN ENO IN1 OUT IN2	MUL EN ENO IN1 OUT IN2	DIV EN ENO IN1 OUT IN2
STL	MOVW IN1, OUT *I IN2,OUT	MOVW IN1, OUT /I IN2,OUT	MOVD IN1, OUT *D IN2,OUT	MOVD IN1, OUT /D IN2,OUT	MOVW IN1, OUT MUL IN2, OUT	MOVW IN1, OUT DIV IN2, OUT
功能	IN1 * IN2= OUT	IN1/IN2= OUT	IN1 * IN2= OUT	IN1/IN2= OUT	IN1 * IN2= OUT	IN1/IN2= OUT

说明：

整数、双整数乘除法指令操作数及数据类型和加减运算的相同。

整数乘法、除法产生双整数指令的操作数。

IN1/IN2：VW,IW,QW,MW,SW,SMW,T,C,LW,AC,AIW,常量,*VD,*LD,*AC。

数据类型：整数。

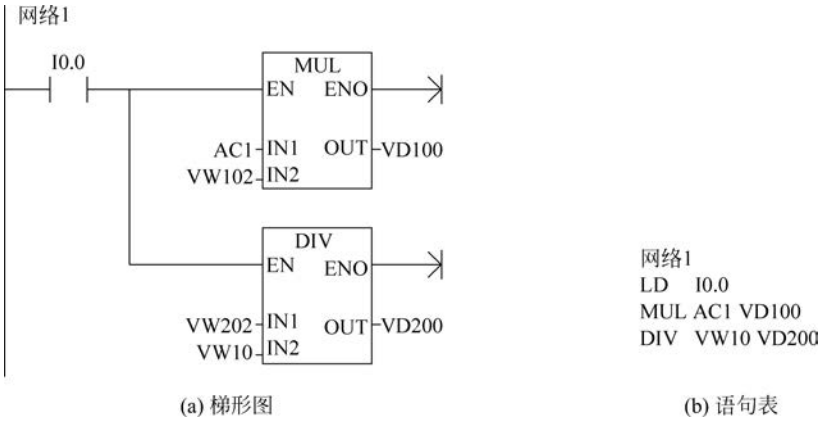
OUT：VD,ID,QD,MD,SMD,SD,LD,AC,*VD,*LD,*AC。

数据类型：双整数。

使ENO=0的错误条件：0006(间接地址)、SM1.1(溢出)、SM1.3(除数为0)。

对标志位的影响：SM1.0(零标志位)、SM1.1(溢出)、SM1.2(负数)、SM1.3(被0除)。

【例 5-16】 乘/除法指令应用举例，梯形图和语句表如图 5-28 所示。



(a) 梯形图　　(b) 语句表

图 5-28　例 5-16 题图

提示　因为 VD100 包含 VW100 和 VW102 两个字，VD200 包含 VW200 和 VW202 两个字，所以在语句表指令中不需要使用数据传送指令。

3. 实数加减乘除指令

实数加法(ADD-R)、减法(SUB-R)指令是将两个 32 位实数相加或相减，并产生一个 32 位实数结果，从 OUT 指定的存储单元输出。

实数乘法(MUL-R)、除法(DIV-R)指令是在使能输入有效时，将两个 32 位实数相乘(除)，并产生一个 32 位积(商)，从 OUT 指定的存储单元输出。

操作数 IN1/IN2：VD，ID，QD，MD，SMD，SD，LD，AC，常量，* VD，* LD，* AC。

OUT：VD，ID，QD，MD，SMD，SD，LD，AC，* VD，* LD，* AC。

数据类型：实数。

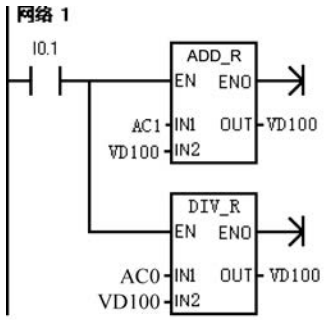
指令格式如表 5-17 所示。

表 5-17　实数加减乘除指令

LAD	ADD_R EN ENO IN1 OUT IN2	SUB_R EN ENO IN1 OUT IN2	MUL_R EN ENO IN1 OUT IN2	DIV_R EN ENO IN1 OUT IN2
STL	MOVD IN1，OUT +R　IN2，OUT	MOVD IN1，OUT −R　IN2，OUT	MOVD IN1，OUT *R　IN2，OUT	MOVD IN1，OUT /R　IN2，OUT
功能	IN1+IN2=OUT	IN1−IN2=OUT	IN1 * IN2=OUT	IN1/IN2=OUT
ENO=0 的错误条件	0006　间接地址；SM4.3　运行时间；SM1.1　溢出		0006　间接地址；SM1.1　溢出；SM4.3　运行时间；SM1.3　除数为 0	
对标志位的影响	SM1.0 零；SM1.1 溢出；SM1.2 负数；SM1.3 被 0 除			

【例 5-17】 实数运算指令的应用。梯形图和语句表如图 5-29 和图 5-30 所示。

（1）实数加/减法运算指令的应用。



(a) 梯形图

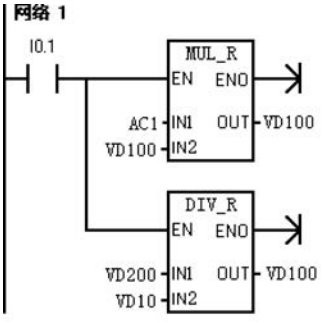
```
网络1
LD   I0.0
+R   AC1, VD100

/R   VD100, AC0
```

(b) 语句表

图 5-29 实数加/减法运算指令（例 5-17 题图）

（2）实数乘/除法运算指令的应用。




```
网络1
LD    I0.1
*R    AC1, VD100

MOVR  VD200, VD100
/R    VD10, VD100
```

(a) 梯形图　　(b) 语句表

图 5-30 实数乘/除法运算指令（例 5-17 题图）

4. 数学函数变换指令

数学函数变换指令包括平方根、自然对数、自然指数、三角函数等。

（1）平方根（SQRT）指令。对 32 位实数（IN）取平方根，并产生一个 32 位实数结果，从 OUT 指定的存储单元输出。

（2）自然对数（LN）指令。对 IN 中的数值进行自然对数计算，并将结果置于 OUT 指定的存储单元中。

求以 10 为底数的对数时，用自然对数除以 2.302 585（约等于 10 的自然对数）。

（3）自然指数（EXP）指令。将 IN 取以 e 为底的指数，并将结果置于 OUT 指定的存储单元中。

将“自然指数”指令与“自然对数”指令相结合，可以实现以任意数为底、任意数为指数的计算。求 y^x，可输入以下指令：EXP（x * LN（y））。

例如：求 2^3 为 EXP(3 * LN(2))＝8；求 27 的 3 次方根 $27^{1/3}$ 为 EXP(1/3 * LN(27))＝3。

（4）三角函数指令。将一个实数的弧度值 IN 分别求 SIN、COS、TAN，得到实数运算结果，从 OUT 指定的存储单元输出。

函数变换指令格式及功能如表 5-18 所示。

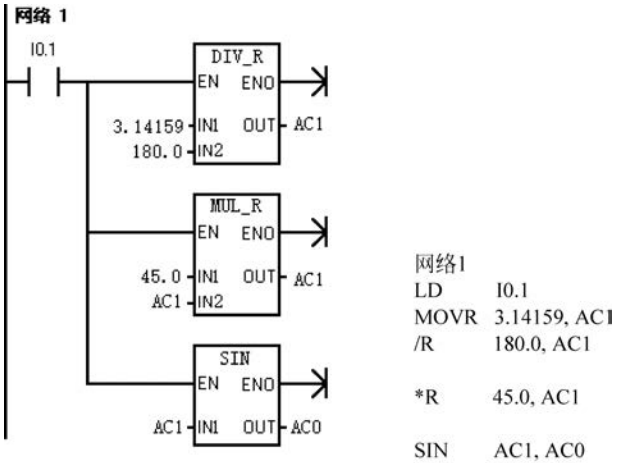
表 5-18　函数变换指令格式及功能

LAD	SQRT EN ENO IN OUT	LN EN ENO IN OUT	EXP EN ENO IN OUT	SIN EN ENO IN OUT	COS EN ENO IN OUT	TAN EN ENO IN OUT
STL	SQRT IN,OUT	LN IN,OUT	EXP IN,OUT	SIN IN,OUT	COS IN,OUT	TAN IN,OUT
功能	SQRT(IN)＝OUT	LN(IN)＝OUT	EXP(IN)＝OUT	SIN(IN)＝OUT	COS(IN)＝OUT	TAN(IN)＝OUT
操作数及数据类型	**IN**：VD,ID,QD,MD,SMD,SD,LD,AC,常量,＊VD,＊LD,＊AC； **OUT**：VD,ID,QD,MD,SMD,SD,LD,AC,＊VD,＊LD,＊AC； **数据类型**：实数					

提示　使 ENO＝0 的错误条件：0006(间接地址)、SM1.1(溢出)、SM4.3(运行时间)。对标志位的影响：SM1.0(零)、SM1.1(溢出)、SM1.2(负数)。

【例 5-18】　利用函数变换指令求 45°正弦值。

设计分析：先将 45°转换为弧度，为(3.141 59/180)＊45，再求正弦值。梯形图和语句表如图 5-31 所示。



(a) 梯形图　　(b) 语句表

图 5-31　例 5-18 题图

【例 5-19】　利用函数变换指令求 65°的正切值。

设计分析：先将 65°转换为弧度，即(3.141 59/180)＊65，再求正切值。梯形图和语句表如图 5-32 所示。

5.2.2　逻辑运算指令

逻辑运算是对无符号数按位进行与、或、异或和取反等操作。操作数的长度有 B、W、DW。指令格式如表 5-19 所示。

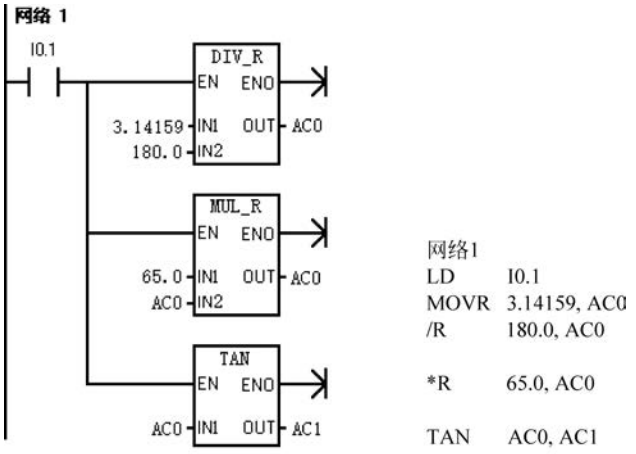


图 5-32　例 5-19 题图

表 5-19　逻辑运算指令格式

LAD		WAND_B WAND_W WAND_DW	WOR_B WOR_W WOR_DW	WXOR_B WXOR_W WXOR_DW	INV_B INV_W INV_DW
STL		ANDB IN1,OUT ANDW IN1,OUT ANDD IN1,OUT	ORB IN1,OUT ORW IN1,OUT ORD IN1,OUT	XORB IN1,OUT XORW IN1,OUT XORD IN1,OUT	INVB OUT INVW OUT INVD OUT
功能		IN1,IN2 按位相与	IN1,IN2 按位相或	IN1,IN2 按位异或	对 IN 取反
操作数	**B**	**IN1/IN2**：VB,IB,QB,MB,SB,SMB,LB,AC,常量,* VD,* AC,* LD; **OUT**：VB,IB,QB,MB,SB,SMB,LB,AC,* VD,* AC,* LD			
	W	**IN1/IN2**：VW,IW,QW,MW,SW,SMW,T,C,AC,LW,AIW,常量,* VD,* AC,* LD; **OUT**：VW,IW,QW,MW,SW,SMW,T,C,LW,AC,* VD,* AC,* LD			
	DW	**IN1/IN2**：VD,ID,QD,MD,SMD,AC,LD,HC,常量,* VD,* AC,SD,* LD; **OUT**：VD,ID,QD,MD,SMD,LD,AC,* VD,* AC,SD,* LD			

(1) 逻辑与(WAND)指令。将输入 IN1、IN2 按位相与,得到的逻辑运算结果放入 OUT 指定的存储单元。

(2) 逻辑或(WOR)指令。将输入 IN1、IN2 按位相或,得到的逻辑运算结果放入 OUT 指定的存储单元。

(3) 逻辑异或(WXOR)指令。将输入 IN1、IN2 按位相异或,得到的逻辑运算结果放入 OUT 指定的存储单元。

(4) 取反(INV)指令。将输入 IN 按位取反,将结果放入 OUT 指定的存储单元。

提示

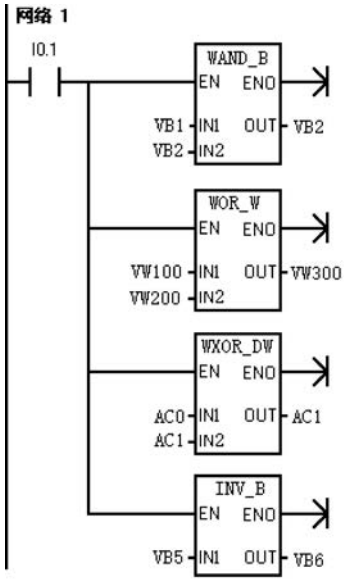
(1) 在逻辑运算指令中,在梯形图指令中设置 IN2 和 OUT 所指定的存储单元相同,这样对应的语句表指令如表 5-19 中所示。若在梯形图指令中,IN2(或 IN1)和 OUT 所指定的存储单元不同,则在语句表指令中须使用数据传送指令,将其中一个输入端的数据先送入 OUT,再进行逻辑运算,如

```
MOVB IN1,OUT
ANDB IN2,OUT
```

(2) ENO=0 的错误条件:0006(间接地址)、SM4.3(运行时间)。

(3) 对标志位的影响:SM1.0(零)。

【例 5-20】 逻辑运算指令编程举例。梯形图和语句表如图 5-33 所示。



(a) 梯形图

```
网络1
//字节与操作
LD      I0.1
ANDB    VB1, VB2

//字或操作
MOVW    VW100, VW300
ORW     VW200, VW300

//双字异或操作
XORD    AC0, AC1

//字节取反操作
MOVB    VB5, VB6
INVB    VB6
```

(b) 语句表

图 5-33　例 5-20 题图

设计分析:

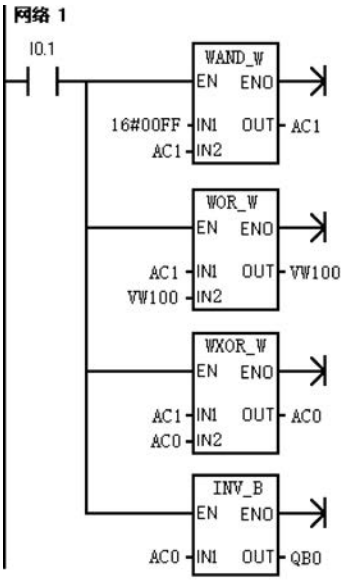
运算过程如下。

```
VB1                          VB2                          VB2
0001 1100          WAND      1100 1101        →           0000 1100

VW100                        VW200                        VW300
0001 1101 1111 1010  WOR     1110 0000 1101 1100→         1111 1101 1111 1110
```

```
VB5                          VB6
0000 1111          INV       1111 0000
```

【例 5-21】 利用逻辑运算指令编程。要求：屏蔽 AC1 的高 8 位；然后 AC1 与 VW100 或运算结果送入 VW100；AC1 与 AC0 进行字异或，结果送入 AC0；最后，AC0 字节取反后输出给 QB0。梯形图和语句表如图 5-34 所示。



(a) 梯形图

```
网络1
//字与操作
LD      I0.1
ANDW    16#00FF, AC1

//字或操作
ORW     AC1, VW100

//字异或操作
XORW    AC1, AC0

//字节取反操作
MOVB    AC0, QB0
INVB    QB0
```

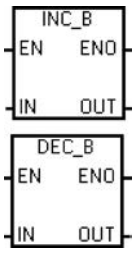
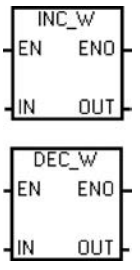
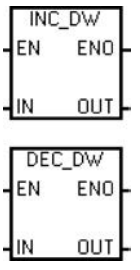
(b) 语句表

图 5-34　例 5-21 题图

5.2.3　递增、递减指令

递增、递减指令用于对输入无符号数字节、符号数字、符号数双字进行加 1 或减 1 的操作。指令格式如表 5-20 所示。

表 5-20　递增、递减指令格式

LAD						
STL	INCB OUT	DECB OUT	INCW OUT	DECW OUT	INCD OUT	DECD OUT
功能	字节加 1	字节减 1	字加 1	字减 1	双字加 1	双字减 1

续表

操作及数据类型	**IN**：VB，IB，QB，MB，SB，SMB，LB，AC，常量，* VD，* LD，* AC； **OUT**：VB，IB，QB，MB，SB，SMB，LB，AC，* VD，* LD，* AC； **数据类型**：字节	**IN**：VW，IW，QW，MW，SW，SMW，AC，AIW，LW，T，C，常量，* VD，* LD，* AC； **OUT**：VW，IW，QW，MW，SW，SMW，LW，AC，T，C，* VD，* LD，* AC； **数据类型**：整数	**IN**：VD，ID，QD，MD，SD，SMD，LD，AC，HC，常量，* VD，* LD，* AC； **OUT**：VD，ID，QD，MD，SD，SMD，LD，AC，* VD，* LD，* AC； **数据类型**：双整数

(1) 递增字节(INC-B)/递减字节(DEC-B)指令。

递增字节和递减字节指令在输入字节(IN)上加 1 或减 1，并将结果置入 OUT 指定的变量中，递增和递减字节运算不带符号。

(2) 递增字(INC-W)/递减字(DEC-W)指令。

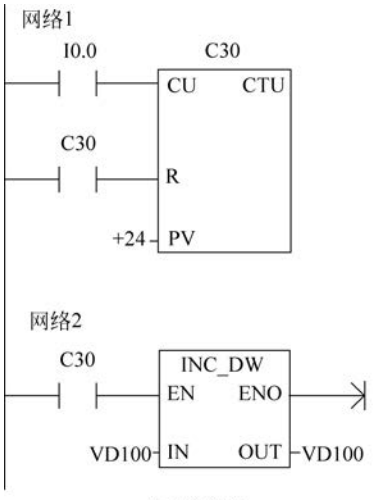
递增字和递减字指令在输入字(IN)上加 1 或减 1，并将结果置入 OUT。递增和递减字运算带符号(16#7FFF > 16#8000)。

(3) 递增双字(INC-DW)/递减双字(DEC-DW)指令。

递增双字和递减双字指令在输入双字(IN)上加 1 或减 1，并将结果置入 OUT。递增和递减双字运算带符号(16#7FFFFFFF > 16#80000000)。

提示　①使 ENO=0 的错误条件：SM4.3(运行时间)、0006(间接地址)、SM1.1(溢出)。②影响标志位：SM1.0 (零)、SM1.1(溢出)、SM1.2(负数)。③在梯形图指令中，IN 和 OUT 可以指定为同一存储单元，这样可以节省内存，在语句表指令中不需要使用数据传送指令。

【例 5-22】 利用递增、递减指令编程。控制要求：食品加工厂对饮料生产线上的盒装饮料进行计数，每 24 盒为一箱，要求能记录生产的箱数。梯形图和语句表如图 5-35 所示。



(a) 梯形图

```
网络1
//增脉冲输入
LD    I0.0
LD    C30
CTU   C30,+24

网络2
//双字增操作
LD    C30
INCD  VD100
```

(b) 语句表

图 5-35　例 5-22 题图

5.3 表功能指令

数据表是用来存放字型数据的表格，如图 5-36 所示，通过专设的表功能指令可以方便地实现对表中数据的各种操作。表格的第一个字地址即首地址，为表地址，首地址中的数值是表格的最大长度(TL)，即最大填表数。表格的第二个字地址中的数值是表的实际长度(EC)，指定表格中的实际填表数。每次向表格中增加新数据后，EC 加 1。从第三个字地址开始存放数据(字)。表格最多可存放 100 个数据(字)，不包括指定最大填表数(TL)和实际填表数(EC)的参数。

VW200	0006	TL(最大填表数)
VW202	0002	EC(实际填表数)
VW204	1234	d0数据0
VW206	5678	d1数据1
VW208	××××	
VW210	××××	
VW212	××××	
VW214	××××	

图 5-36 数据表

要建立表格，首先须确定表的最大填表数。梯形图和语句表如图 5-37 所示。

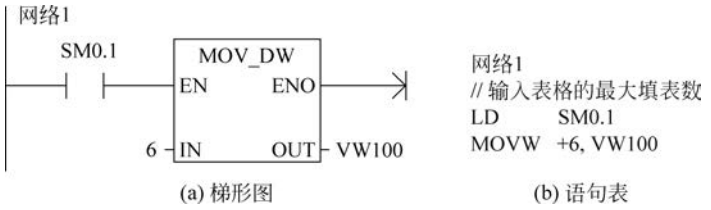


图 5-37 输入表格的最大填表数

确定表格的最大填表数后，可用表功能指令在表中存取字型数据。S7-200 PLC 表功能指令包括填表指令、表取数指令、表查找指令、字填充指令。表功能指令包括所有的表格读取和表格写入指令，必须用边缘触发指令激活。

5.3.1 填表指令

填表(ATT)指令向表格(TBL)中增加一个字(DATA)。填表指令的格式如表 5-21 所示。

表 5-21 填表指令格式

LAD	STL	功能及操作数
AD_T_TBL EN ENO DATA TBL	ATT DATA,TBL	**功能**：当 EN 有效时，将输入的字型数据填写到指定的表格中。在填表时，新数据填写到表格中最后一个数据的后面； **DATA**：VW，IW，QW，MW，SW，SMW，LW，T，C，AIW，AC，常量，* VD，* LD，* AC，数据类型为整数； TBL 为表格的首地址：VW，IW，QW，MW，SW，SMW，LW，T，C，* VD，* LD * AC，数据类型为字

提示　①表 5-21 中的第一个字存放表的最大长度(TL)；第二个字存放表内实际的项数(EC)。②每填入一个新数据 EC 自动加 1。表最多可以装入 100 个有效数据(不包括 LTL 和 EC)。③使 ENO＝0 的错误条件：0006(间接地址)、0091(操作数超出范围)、SM1.4(表溢出)、SM4.3(运行时间)。④填表指令影响特殊标志位 SM1.4(填入表的数据超出表的最大长度时，SM1.4＝1)。

【例 5-23】　利用填表指令编程。将 VW100 中的数据 5678，填入首地址为 VW200 的数据表中。

设计分析：

(1) 首地址为 VW200 的表存储区，表中数据在执行本指令前已经建立，表中第一字单元存放表的长度为 5，第二字单元存放实际数据项 2 个，表中两个数据项为 1234 和 4321。

(2) 将 VW100 单元的字数据 5678 追加到表的下一个单元(VW208)中，且 EC 自动加 1。

梯形图、语句表及运行结果如图 5-38 所示。

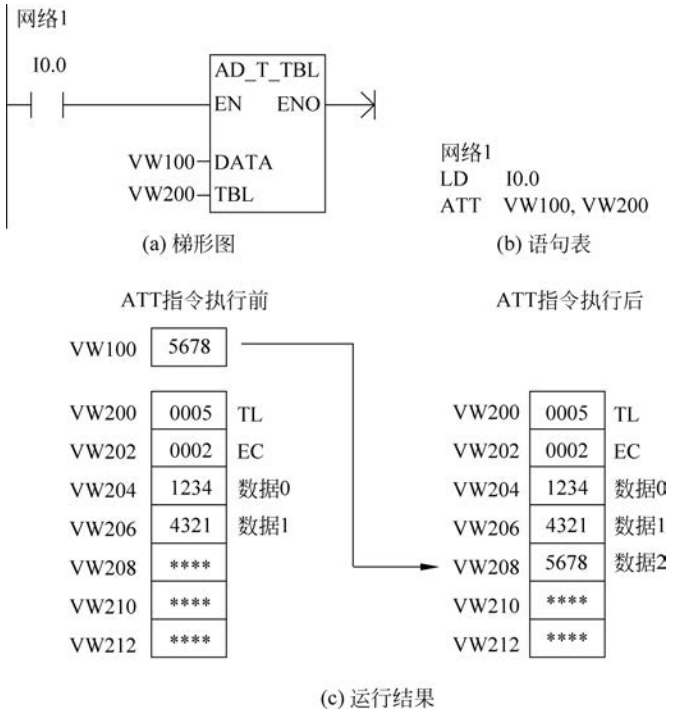


图 5-38　例 5-23 题图

5.3.2　表取数指令

从数据表中取数有先进先出(FIFO)和后进先出(LIFO)两种。执行表取数指令后，实际填表数 EC 值自动减 1。

(1) 先进先出指令。移出表格(TBL)中的第一个数(数据 0)，并将该数值移至 DATA 指定存储单元，表格中的其他数据依次向上移动一个位置。

(2) 后进先出指令。将表格(TBL)中的最后一个数据移至输出端 DATA 指定的存储单元，表格中的其他数据位置不变。

表取数指令格式如表 5-22 所示。

表 5-22　表取数指令格式

LAD	FIFO EN　ENO TBL　DATA	LIFO EN　ENO TBL　DATA
STL	FIFO　TBL,DATA	LIFO　TBL,DATA
说明	输入端 TBL 为数据表的首地址,输出端 DATA 为存放取出数值的存储单元	
操作数及数据类型	**TBL**：VW,IW,QW,MW,SW,SMW,LW,T,C,＊VD,＊LD,＊AC,数据类型为字； **DATA**：VW,IW,QW,MW,SW,SMW,LW,AC,T,C,AQW,＊VD,＊LD,＊AC,数据类型为整数	

提示　使 ENO＝0 的错误条件：0006(间接地址)、0091(操作数超出范围)、SM1.5(空表)、SM4.3(运行时间)。对特殊标志位的影响：SM1.5(试图从空表中取数,SM1.5＝1)。

【例 5-24】　利用表取数指令编程。

(1) 先进先出指令应用。

设计分析：

① 表首地址 VW200 单元,内容 0006 表示表的长度,数据 3 项,表中数据从 VW204 单元开始。

② 在 I0.0 有效时,将最先进入表中的数据 3256 送入 VW300 单元,下面数据依次上移,EC 减 1。梯形图、语句表及运行结果如图 5-39 所示。

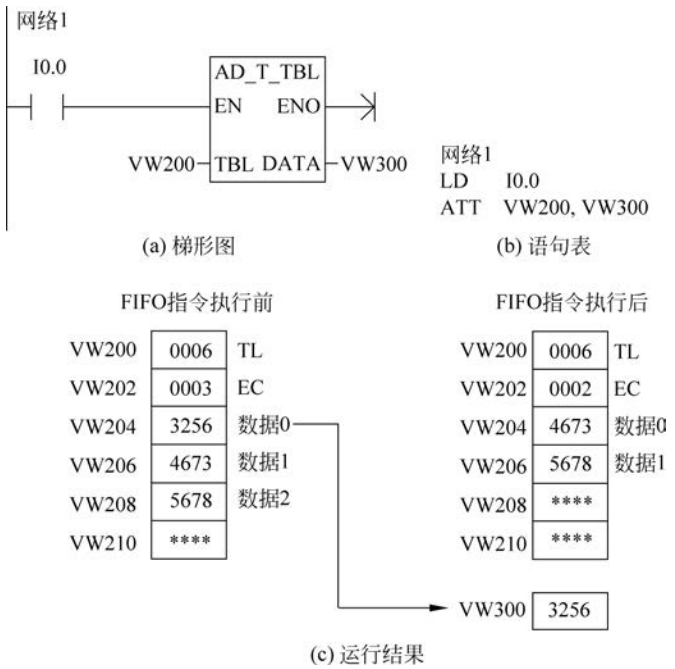


图 5-39　先进先出指令(例 5-24 题图)

(2) 后进先出指令应用。

设计分析：

① 表首地址 VW100 单元，内容 0006 表示表的长度，数据 3 项，表中数据从 VW104 单元开始。

② 在 I0.0 有效时，将最后进入表中的数据 3721 送入 VW200 单元，EC 减 1。

梯形图、语句表及运行结果如图 5-40 所示。

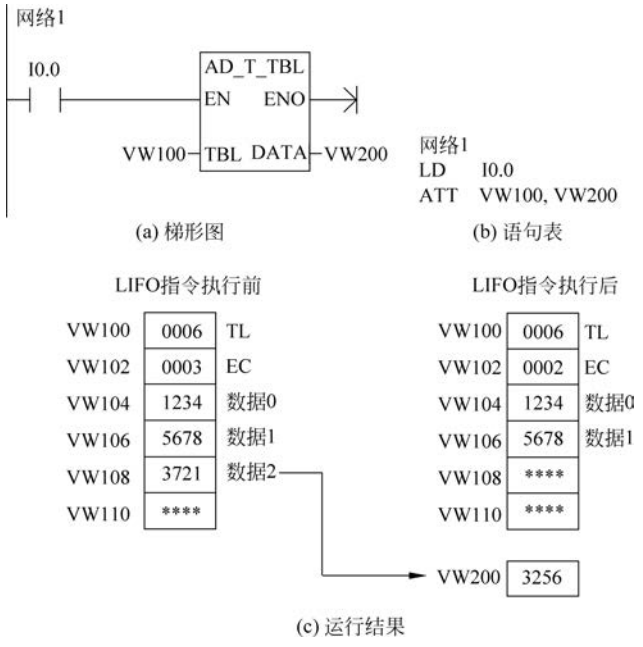


图 5-40　后进后出指令(例 5-24 题图)

(3) 表取数指令综合应用。

设计分析：在图 5-36 的数据表中，用 FIFO、LIFO 指令取数，将取出的数值分别放入 VW300、VW400 中。

梯形图、语句表及运行结果如图 5-41 所示。

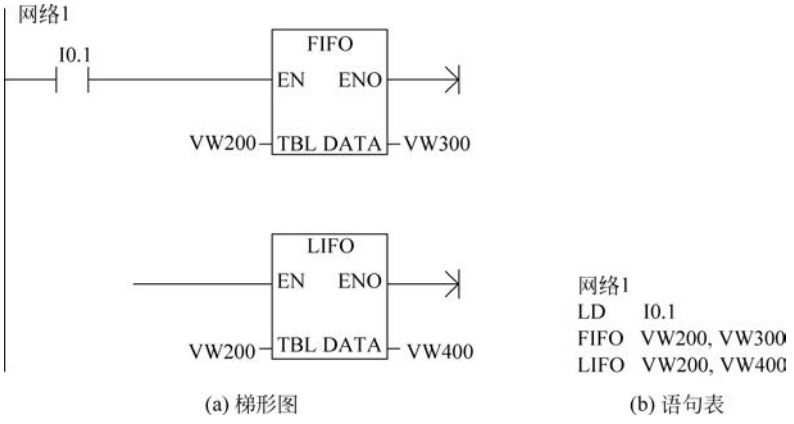
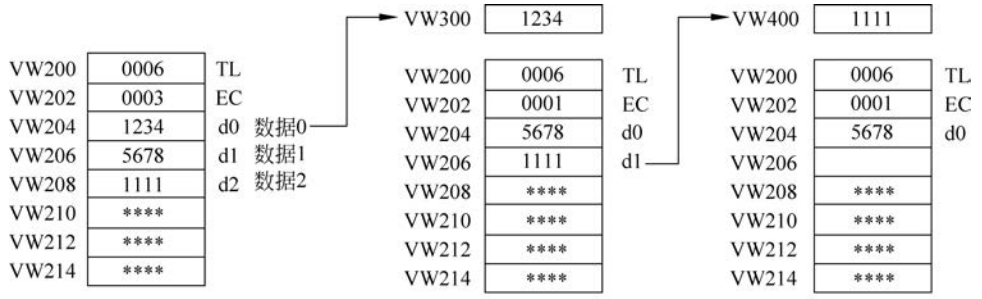


图 5-41　表取数指令综合(例 5-24 题图)



(c) 运行结果

图 5-41 （续）

5.3.3 表查找指令

表查找(TBL-FIND)指令用于在表格(TBL)中搜索符合条件的数据在表中的位置(用数据编号表示,编号范围为 0～99)。其指令格式如表 5-23 所示。

表 5-23 表查找(TBL-FIND)指令格式

LAD	STL	功能及操作数
TBL_FIND EN ENO TBL PTN INDX CMD	FND= TBL,PATRN,INDX FND<> TBL,PATRN,INDX FND< TBL,PATRN,INDX FND> TBL,PATRN,INDX	**功能**：在执行查表指令前,首先对 INDX 清 0,当 EN 有效时,从 INDX 开始搜索 TBL,查找符合 PTN 且 CMD 所决定的数据,每搜索一个数据项,INDX 自动加 1；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。 **TBL**：为表格的实际填表数对应的地址(第二个字地址),即高于对应的“增加至表格”“后入先出”或“先入先出”指令 TBL 操作数的一个字地址(2 字节)。TBL 操作数为 VW,IW,QW,MW,SW,SMW,LW,T,C,* VD,* LD,* AC。数据类型为字。 **PTN**：用来描述查表条件时进行比较的数据。PTN 操作数为 VW,IW,QW,MW,SW,SMW,AIW,LW,T,C,AC,常量,* VD,* LD,* AC。数据类型为整数。 **INDX**：搜索指针,即从 INDX 所指的数据编号开始查找,并将搜索到的符合条件的数据的编号放入 INDX 所指定的存储器。INDX 操作数为 VW,IW,QW,MW,SW,SMW,LW,T,C,AC,* VD,* LD,* AC。数据类型：字。 **CMD**：比较运算符,其操作数为常量 1～4,分别代表 =、<>、<、>。数据类型为字节

说明：

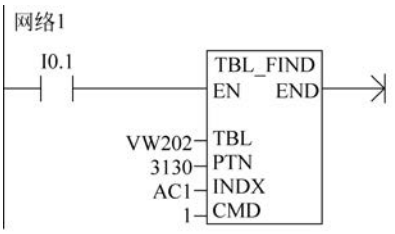
(1) 表查找指令搜索表格时，从 INDX 指定的数据编号开始，寻找与数据 PTN 的关系满足 CMD 比较条件的数据。参数如果找到符合条件的数据，则 INDX 的值为该数据的编号。要查找下一个符合条件的数据，再次使用“表格查找”指令之前须将 INDX 加 1。如果没有找到符合条件的数据，则 INDX 的数值等于实际填表数 EC。一个表格最多可有 100 个数据，数据编号范围为 0～99。将 INDX 的值设为 0，则从表格的顶端开始搜索。

(2) 使 ENO＝0 的错误条件：SM4.3(运行时间)、0006(间接地址)、0091(操作数超出范围)。

提示　查表指令不需要 ATT 指令中的最大填表数 TL。因此，查表指令的 TBL 操作数比 ATT 指令的 TBL 操作数高 2 字节。例如，ATT 指令创建的表 TBL＝VW200，对该表进行查找指令时的 TBL 应为 VW202。

【例 5-25】　利用查表指令编程。从 EC 地址为 VW202 的表中查找等于 3030 数据的位置存入 AC1 中，设表中数据均为十进制数表示。

设计分析：为了从表格的顶端开始搜索，AC1 的初始值＝0，查表指令执行后 AC1＝1，找到符合条件的数据 1。继续向下查找，先将 AC1 加 1，再激活表查找指令，从表中符合条件的数据 1 的下一个数据开始查找，第二次执行查表指令后，AC1＝4，找到符合条件的数据 4。继续向下查找，将 AC1 再加 1，再激活表查找指令，从表中符合条件的数据 4 的下一个数据开始查找，第三次执行表查找指令后，没有找到符合条件的数据，AC1＝6(实际填表数)。梯形图、语句表及运行结果如图 5-42 所示。



(a) 梯形图

```
网络1
LD     I0.1
FND=   VW202,3130,AC1
```

(b) 语句表

执行前

AC1	0	
VW202	0006	EC
VW204	4542	数据0
VW206	4142	数据1
VW208	3030	
VW210	3234	
VW212	3235	

执行后

AC1	2	
VW202	0006	EC
VW204	4542	数据0
VW206	4142	数据1
VW208	3030	
VW210	3234	
VW212	3235	

(c) 运行结果

图 5-42　例 5-25 题图

执行过程如下。

（1）表首地址 VW202 单元，内容 0006 表示表的长度，表中数据从 VW204 单元开始；

（2）若 AC1＝0，在 I0.1 有效时，从 VW204 单元开始查找；

（3）在搜索到 PTN 数据 3030 时，AC1＝2，其存储单元为 VW208。

习题与思考题

5-1 已知 VB10＝18，VB20＝30，VB21＝33，VB32＝98。将 VB10，VB30，VB31，VB32 中的数据分别送到 AC1，VB200，VB201，VB202 中。写出梯形图及语句表程序。

5-2 试用传送指令编写梯形图程序。要求控制 Q0.0～Q0.7 对应的 8 个指示灯，在 I0.0 接通时，使输出隔位接通，在 I0.1 接通时，输出取反后隔位接通。

5-3 编制检测上升沿变化的程序。每当 I0.0 接通一次，使存储单元 VW0 的值加 1，则如果计数达到 5，则输出 Q0.0 接通显示，用 I0.1 使 Q0.0 复位。

5-4 用数据类型转换指令实现将 cm 转换为 in。已知 1in＝2.54cm。

5-5 编写输出字符 A 的七段显示码程序。

5-6 彩灯的循环移位控制。假设有 8 个指示灯，从右到左以 0.5s 的速度依次点亮，任意时刻只有一个指示灯亮，到达最左端，再从右到左依次点亮。

5-7 炫丽彩灯灯光的模拟控制。控制要求：L1、L2、L9→L1、L5、L8→L1、L4、L7→L1、L3、L6→L1→L2、L3、L4、L5→L6、L7、L8、L9→L1、L2、L6→L1、L3、L7→L1、L4、L8→L1、L5、L9→L1→L2、L3、L4、L5→L6、L7、L8、L9→L1、L2、L9→L1、L5、L8……循环下去。按下面的 I/O 分配编写程序。

输入	输出	
起动按钮：I0.0	L1：Q0.0	L6：Q0.5
停止按钮：I0.1	L2：Q0.1	L7：Q0.6
	L3：Q0.2	L8：Q0.7
	L4：Q0.3	L9：Q1.0
	L5：Q0.4	

5-8 用算术运算指令完成下列的运算。①$8^4$；②求 COS60°。

5-9 将 VW100 开始的 10 个字的数据送到 VW200 开始的存储区。