Raynen

X1 系列可编程逻辑控制器用户手册软件篇

Raynen 睿能®

股票代码：603933

目录

目录 1
一、X 系列内存详解 8
1.1. X1 资源配置 8
1.2. X1 特殊内存配置 11
1.2.1 特殊辅助寄存器 SM8（M8） 11
1.2.2 特殊数据寄存器 SD8（D8） 17
二 、指令篇 23
2.1 基本指令 32
2.1.1 LD、LDI/取、取反指令 34
2.1.2 OUT/输出指令 36
2.1.3 AND、ANI/与、与反转指令 38
2.1.4 OR、ORI/或、或反转指令 39
2.1.5 LDP、LDF、ANDP、ORP、ORF/沿触发类指令 40
2.1.6 ORB/回路块或指令 42
2.1.7 ANB/回路块与指令 42
2.1.8 MPS、MRD、MPP/堆栈类指令 43
2.1.9 MC、MCR/主控、主控复位指令 44
2.1.10 INV/反转指令 46
2.1.11 MEP、MEF/运算结果脉冲化指令 47
2.1.12 PLS、PLF/脉冲输出指令 47
2.1.13 SET、RST/置位、复位指令 51
2.1.14 NOP/空指令 52
2.1.15 END/程序结束指令 52
2.1.16 指令步数和指令软元件 53
2.2 流程指令 54
2.2.1 CJ/程序跳转指令 54
2.2.2 CALL/子程序调用指令 56
2.2.3 SRET/子程序返回指令 59

2.2.4 IRET/中断返回指令..... 60
2.2.5 EI/允许中断指令..... 60
2.2.6 DI/禁止中断指令..... 61
2.2.7 FEND/主程序结束指令指令..... 61
2.2.8 WDT/看门狗定时器指令..... 62
2.2.9 FOR/循环范围开始指令..... 63
2.2.10 NEXT/循环范围结束指令..... 64
2.3 传送比较指令..... 65
2.3.1 CMP/比较指令..... 65
2.3.2 ZCP/区间比较指令..... 66
2.3.3 MOV/传送指令..... 68
2.3.4 SMOV/位移动指令..... 70
2.3.5 CML/反转指令..... 71
2.3.6 BMOV/成批传送指令..... 72
2.3.7 FMOV/多点传送指令..... 74
2.3.8 XCH/交换指令..... 75
2.3.9 BCD/BCD 转换指令 77
2.3.10 BIN/BIN 转换指令 78
2.4 四则逻辑运算指令..... 80
2.4.1 ADD/加法运算指令..... 80
2.4.2 SUB/减法运算指令..... 81
2.4.3 MUL/乘法运算指令..... 83
2.4.4 DIV/除法运算指令..... 85
2.4.5 INC/加一指令..... 86
2.4.6 DEC/减一指令..... 87
略。..... 88
2.4.7 WAND/逻辑与指令..... 89
2.4.8 WOR/逻辑或指令..... 90
2.4.9 WXOR/逻辑异或指令..... 91
2.4.10 NEG/补码指令..... 93

2.5 循环移位指令..... 94
2.5.1 ROR/循环右移指令..... 94
2.5.2 ROL/循环左移指令..... 96
2.5.3 RCR/带进位循环右移指令..... 97
2.5.4 RCL/带进位循环左移指令..... 99
2.5.5 SFTR/位右移指令..... 100
2.5.6 SFTL/位左移指令..... 102
2.5.7 WSFR/字右移指令..... 104
2.5.8 WSFL/字左移指令..... 107
2.5.9 SFWR/移位写入指令..... 110
2.5.10 SFRD/移位读出指令..... 111
2.6 数据处理指令..... 113
2.6.1 ZRST/成批复位指令..... 113
2.6.2 DECO/译码指令..... 114
2.6.3 ENCO/编码指令..... 116
2.6.4 SUM/计算 ON 位数指令..... 118
2.6.5 BON/ON 位的判定指令 120
2.6.6 MEAN/平均值指令..... 121
2.6.7 ANS/信号报警置位指令..... 123
2.6.8 ANR/信号报警复位指令..... 124
2.6.9 SQR/开方运算指令..... 125
2.6.10 FLT/BIN 整数→2 进制浮点数转换指令..... 127
2.7 高速处理指令..... 128
2.7.1 REF/输入输出刷新指令..... 128
2.7.2 REFF/带滤波的输入输出刷新指令..... 129
2.7.3 HSCS/高速比较置位指令..... 130
2.7.4 HSCR/高速比较复位指令..... 133
2.7.5 HSZ/高速区间比较指令..... 134
2.7.6 SPD/脉冲密度指令..... 142
2.7.7 PLSY/脉冲输出指令..... 144

2.7.8 PWM/脉宽调制指令..... 146
2.7.9 PLSR/带加减速的脉冲输出指令..... 147
2.8 通信指令..... 150
2.8.1 RS2/串行数据传送 2 指令..... 150
2.8.2 ADPRW/MODBUS 通信主站指令 150
2.9 浮点运算指令..... 150
2.9.1 ECMP/2 进制浮点数比较指令 150
2.9.2 EZCP/2 进制浮点数区间比较指令 152
2.9.3 EMOV/2 进制浮点数传送指令 154
2.9.4 ESTR/2 进制浮点数→字符串的转换指令 155
2.9.5 EVAL/字符串→2 进制浮点数的转换指令 157
2.9.6 EBCD/2 进制浮点数→10 进制浮点数的转换指令..... 160
2.9.7 EBIN/10 进制浮点数→2 进制浮点数的转换指令..... 161
2.9.8 EADD/2 进制浮点数加法运算指令 163
2.9.9 ESUB/2 进制浮点数减法运算指令 164
2.9.10 EMUL/2 进制浮点数乘法运算指令 165
2.9.11 EDIV/2 进制浮点数除法运算指令 167
2.9.12 EXP/2 进制浮点数指数运算指令 168
2.9.13 LOGE/2 进制浮点数自然对数运算指令 169
2.9.14 LOG10/2 进制浮点数常用对数运算指令 170
2.9.15 ESQR/2 进制浮点数开方运算指令 171
2.9.16 ENEG/2 进制浮点数符号翻转指令 172
2.9.17 INT/2 进制浮点数→BIN 整数的转换指令..... 173
2.9.18 SIN/2 进制浮点数 SIN 运算指令 174
2.9.19 COS/2 进制浮点数 COS 运算指令 175
2.9.20 TAN/2 进制浮点数 TAN 运算指令 176
2.9.21 ASIN/2 进制浮点数 SIN-1 运算指令 177
2.9.22 ACOS/2 进制浮点数 COS-1 运算指令 178
2.9.23 ATAN/2 进制浮点数 TAN-1 运算指令 179
2.9.24 RAD/2 进制浮点角度→弧度的转换指令 180

2.9.25 DEG/2 进制浮点弧度→角度的转换指令 181
2.10 数据处理 2 指令..... 182
2.10.1 WSUM/算出数据合计值指令..... 182
2.10.2 WTOB/字节单位的数据分离指令..... 184
2.10.3 BTOW/字节单位的数据结合指令..... 185
2.10.4 UNI/16 位数据的 4 位结合指令 187
2.10.5 DIS/16 位数据的 4 位分离指令 188
2.10.6 SWAP/高低字节互换指令..... 189
2.10.7 SORT2/数据排序 2 指令..... 191
2.11 定位指令..... 194
2.11.1 DSZR/带 DOG 搜索的原点回归指令..... 197
2.11.2 DVIT/中断定位指令..... 200
2.11.3 ZRN/原点回归指令..... 203
2.11.4 PLSV/可变速脉冲输出指令..... 204
2.11.5 DRVI/相对定位指令..... 206
2.11.6 DRVA/绝对定位指令..... 208
2.12 时钟运算指令..... 210
2.12.1 TCMP/时钟数据比较指令..... 210
2.12.2 TZCP/时钟数据区间比较指令..... 212
2.12.3 TADD/时钟数据加法运算指令..... 213
2.12.4 TSUB/时钟数据减法运算指令..... 214
2.12.5 HTOS/时、分、秒数据的秒转换指令..... 215
2.12.6 STOH/秒数据的（时、分、秒）转换指令..... 217
2.12.7 TRD/读出时钟数据指令..... 218
2.12.8 TWR/写入时钟数据指令..... 219
2.12.9 HOUR/计时表指令..... 221
2.13 格雷码指令..... 222
2.13.1 GRY/格雷码的转换指令..... 222
2.13.2 GBIN/格雷码的逆转换指令..... 224
2.14 数据块指令..... 225

2.14.1 BK+/数据块的加法运算指令..... 225
2.14.2 BK-/数据块的减法运算指令..... 227
2.14.3 BKCMP=、>、<、<=、>=/数据块的比较指令..... 229
2.15 字符串控制指令..... 232
2.15.1 STR/BIN→字符串的转换指令..... 232
2.15.2 VAL/字符串→BIN 的转换指令 235
2.15.3 $+/字符串的结合指令..... 237
2.15.4 LEN/检测出字符串的长度指令..... 239
2.15.5 RIGHT/从字符串的右侧开始取出指令..... 240
2.15.6 LEFT/从字符串的左侧开始取出指令..... 242
2.15.7 MIDR/从字符串的任意取出指令..... 243
2.15.8 MIDW/从字符串的任意替换指令..... 245
2.15.9 INSTR/字符串的检索指令..... 247
2.15.10 $MOV/字符串的传送指令..... 249
2.16 数据处理 3 指令..... 250
2.16.1 FDEL/数据表的数据删除指令..... 250
2.16.2 FINS/数据表的数据插入指令..... 252
2.16.3 POP/读取后入的数据指令..... 253
2.16.4 SFR/16 位数据 n 位右移（带进位）指令 255
2.16.5 SFL/16 位数据 n 位左移（带进位）指令 256
2.17 触点比较指令..... 258
2.17.1 LD=、>、<、<>、<=、>=/触点比较指令..... 258
2.17.2 AND=、>、<、<>、<=、>=/触点比较指令..... 260
2.17.3 OR=、>、<、<>、<=、>=/触点比较指令..... 263
2.18 数据表处理指令..... 265
2.18.1 LIMIT/上下限位控制指令..... 265
2.18.2 BAND/死区控制指令..... 267
2.18.3 ZONE/区域控制指令..... 269
2.18.4 SCL/定坐标（不同点坐标数据）指令..... 270
2.18.5 DABIN/10 进制 ASCII→BIN 的转换指令 274

2.18.6 BINDA/BIN→10 进制 ASCII 的转换指令 276
2.18.7 SCL2/定坐标 2（X/Y 坐标数据）指令..... 278
2.19 扩展文件寄存器指令..... 281
2.19.1 LOADR/读出扩展文件寄存器指令..... 281
2.19.2 SAVER/成批写入扩展文件寄存器指令..... 282
2.19.3 INITR/扩展寄存器的初始化指令..... 284
2.19.4 LOGR/登录到扩展文件寄存器指令..... 285
2.19.5 RWER/扩展文件寄存器的写入指令..... 287
2.19.6 INITER/扩展文件寄存器的初始化指令..... 288
2.20 方便指令..... 290
2.20.1 ABSD/凸轮控制绝对方式指令..... 290
2.20.2 INCD/凸轮控制相对方式指令..... 291
2.20.3 ALT/交替输出指令..... 293
2.20.4 SORT/数据排序指令..... 294
三、应用篇..... 298
3.1 子程序..... 298
3.2 中断程序..... 300
3.3 高速计数..... 306
3.4 串口通讯部分..... 308
3.4.1 监控下载 308
3.6 以太网功能..... 329
3.7 BD 板模块的使用 337
3.7.1 BD 板的 ID 及版本说明 337
3.7.2 数字量 BD 板使用说明..... 338
3.7.3 模拟量 BD 板使用说明..... 338
3.7.4 模拟量 BD 板地址对照表..... 339
四、附录..... 351
4.1 错误码内容..... 351

一、X 系列内存详解

1.1.X1 资源配置

名称	内容	软元件类型	内容
X	256 点	位元件	X0 到 X377（8 进制）
Y	256 点	位元件	Y0 到 Y377（8 进制）
M	7680 点	位元件	M0 到 M7679
SM	512 点	位元件	SM8000~SM8511
S	4096 点	位元件	S0~S4095
V	8 点	字元件	V0~V7
Z	8 点	字元件	Z0~Z7
T	512 点	字元件	T0~T511
C	200 点	16 位字元件	C0~C199
C（32 位）	56 点	32 位字元件	C200~C255
D	8000 点	字元件	D0~D7999
SD	512 点	字元件	SD8000~SD8511
R	32768 点	字元件	R0~R32767
输入中断	6 点		I00*~I50*
定时器中断	3 点		I6**~I8**
计数器中断	6 点		I010~I060
主控用嵌套	8 点		N0~N7
指针 P	1024 点		P0~P1023
常数 K			16 位范围-32768~+32767 32 位范围 -2147483648~+2147483647
常数 H			16 位范围 0~FFFF 32 位范围 0~FFFFFFFF
实数 E			-1.0×2^{128}～ -1.0×2^{-126}, 0, 1.0×2^{-126}～ 1.0×2^{128}（单精度浮点数范围）
字符串			以“”框起来的半角字符 最长 32 个字符

特别说明：

（1）定时器分类

100ms 型 0.1~3276.7s	10ms 型 0.01~327.67s	1ms 累计型 0.001~32.767s	100ms 累计型 0.1~3276.7s	1ms 型 0.001~32.767s
T0~T199 200 点	T200~T245 46 点	T246~T249 4 点 执行中断 保持用	T250~T255 6 点 保持用	T256~T511 256 点
T192~T199 子程序专用				

其中子程序专用和执行中断保持用的用法比较特殊，普通定时器仅在执行子程序时才计时，如果不执行子程序，计时会中断。在子程序中启动了专用定时器，即使子程序不执行了。定时器仍然继续计时，这样就保证了计时的准确性。

（2）使用计数器时注意掉电保持部分的寄存器

掉电保持的计数器被复位后，会保存改计数器的复位状态，在下次使用改计数器时请在梯形图中加入一条复位改计数器的指令，否则改计数器仍然处于复位状态，将不能正常计数。

（3）V、Z 寄存器

当 32 位指令的操作数进行变址操作时，V 为高 16 位，Z 为低 16 位；对 X，Y 软元件进行变址时，要将 V，Z 值转换成 8 进制后进行加法运算。

（4）掉电保持范围

	符号	进制	点数	起始地址	结束地址	锁存起始	锁存结束	锁存设置范围
辅助继电器	M	10	7680	0	7679	500	1023	0-1023
状态	S	10	4096	0	4095	500	999	0-999
定时器	T	10	512	0	511	T246~T255 掉电保持		
计数器 （16 位）	C	10	200	0	199	100	199	0-199
计数器 （32 位）	C	10	56	200	255	220	255	200-255
数据寄存器	D	10	8000	0	7999	200	511	0-511
扩展寄存器	R	10	32768	0	32767	R0~R32767 掉电保持		

其中：

- 锁存设置范围没有设定时：默认掉电不保持。
- 锁存起始~锁存结束：可以选择，选择将不能保持的范围变为保持的。

- 锁存结束~结束地址：默认掉电保持。
- X、Y 默认不保存，T 是默认只保持 T246~T255，R 寄存器都保持。

（5）字软元件的位指定 Dx.y

指定字软元件的位，可以将其作为位数据使用，指定字软元件的位时，请使用字软元件编号和位编号(16 进制数)进行设定，其中 x=0~7999，y=0~F。 当做位软元件使用时，Dx.y 不支持变址操作。

（6）组合位元件

位软元件，通过组合使用后也可以处理数值。这种情况下，以位数 Kn 和起始软元件的编号组合来表示，类似 KnX、KnY、KnS、KnM。K1～K4 可表示成 16 位数据，K1～K8 可表示成 32 位数据，K 每增加 1 对应的是 4 个位元件，位数和 16 位或 32 位数据位数不匹配时，舍弃高位。组合位元件支持变址操作，比如 MOV K2M0V0 D0，V0 = 2 时，表示将 M2~M9 的数据组合成 16 位数移动到 D0。

（4）掉电保持范围

	符号	进制	点数	起始地址	结束地址	锁存起始	锁存结束	锁存设置范围
辅助继电器	M	10	7680	0	7679	500	1023	0-1023
状态	S	10	4096	0	4095	500	999	0-999
定时器	T	10	512	0	511	T246~T255 掉电保持		
计数器（16 位）	C	10	200	0	199	100	199	0-199
计数器（32 位）	C	10	56	200	255	220	255	200-255
数据寄存器	D	10	8000	0	7999	200	511	0-511
扩展寄存器	R	10	32768	0	32767	R0~R32767 掉电保持		

其中：

- 锁存设置范围没有设定时：默认掉电不保持。
- 锁存起始~锁存结束：可以选择，选择将不能保持的范围变为保持的。
- 锁存结束~结束地址：默认掉电保持。
- X、Y 默认不保存，T 是默认只保持 T246~T255，R 寄存器都保持。

（5）字软元件的位指定 Dx.y

指定字软元件的位，可以将其作为位数据使用，指定字软元件的位时，请使用字软元件编号和位编号(16 进制数)进行设定，其中 x=0~7999，y=0~F。 当做位软元件使用时，Dx.y 不支持变址操作。

（6）组合位元件

位软元件，通过组合使用后也可以处理数值。这种情况下，以位数 Kn 和起始软元件的编号组合来表示，类似 KnX、KnY、KnS、KnM。K1～K4 可表示成 16 位数据，K1～K8 可表示成 32 位数据，K 每增加 1 对应的是 4 个位元件，位数和 16 位或 32 位数据位数不匹配时，舍弃高位。组合位元件支持变址操作，比如 MOV K2M0V0 D0，V0 = 2 时，表示将 M2~M9 的数据组合成 16 位数移动到 D0。

1.2.X1 特殊内存配置

1.2.1 特殊辅助寄存器 SM8（M8）

寄存器号	功能	说明
SM8000	RUN 监控 a 触点：RUN 时一直为 ON	
SM8001	RUN 监控 a 触点：RUN 时一直为 OFF	
SM8002	初始脉冲 a 触点：仅第一个扫描周期为 ON	
SM8003	初始脉冲 b 触点：仅第一个扫描周期为 OFF	
SM8004	错误发生标志位	
SM8011	10ms 时钟	
SM8012	100ms 时钟	
SM8013	1s 时钟	
SM8014	1min 时钟	
SM8015	停止计时以及预置	实时时钟相关
SM8016	时间读出后的显示被停止	
SM8017	±30 秒的修正	
SM8018	检测出安装(一直为 ON)	
SM8019	实时时钟(RTC)错误	
SM8020	零标志位	加减法运算结果为 0 时置位
SM8021	借位标志位	减法运算结果超过最大的负值时置位

SM8022	进位标志位	加法运算结果发生进位时，或者移位结果发生溢出时置位
SM8024	指定 BMOV 方向	
SM8029	指令动作结束标志	一般用于多周期才能动作结束的指令
SM8030	电池 LED 灭灯指示	驱动 SM8030 后，即使电池电压低，可编程控制器面板上的 LED 也不点亮
SM8031	非保持内存全部清除	特殊 SM8，SD8 不被清除
SM8032	保持内存全部清除	
SM8033	内存保持停止	
SM8034	禁止所有输出	
SM8035	强制 RUN 指令	
SM8036	强制 RUN 指令	
SM8037	强制 STOP 指令	
SM8039	恒定扫描模式	
SM8046	STL 状态动作（在执行 END 指令时处理）	当 SM8047 接通时，S0~S899、S1000~S4095 中任意一个为 ON 则接通
SM8047	STL 监控有效（在执行 END 指令时处理）	SM8047 接通后，SD8040~SD8047 有效
SM8048	信号报警器动作	当 SM8049 接通时，S900~S999 中任意一个为 ON 则接通
SM8049	信号报警器有效	SM8049 接通后，SD8049 的动作有效
SM8050	禁止输入中断 I00□	run→stop 清除
SM8051	禁止输入中断 I10□	
SM8052	禁止输入中断 I20□	
SM8053	禁止输入中断 I30□	
SM8054	禁止输入中断 I40□	

SM8055	禁止输入中断 I50□	
SM8056	禁止定时器中断 I6□□	
SM8057	禁止定时器中断 I7□□	
SM8058	禁止定时器中断 I8□□	
SM8059	计数器中断禁止 I010~I060	
SM8061	PLC 硬件错误	
SM8062	[COM0]串行通信出错锁存	
SM8063	[COM1]串行通信出错锁存	
SM8064	参数错误	
SM8065	语法错误	
SM8066	回路错误	
SM8067	运算错误	
SM8068	运算出错锁存	
SM8080	MODBUS TCP IP1 的通信超时标志	
SM8081	MODBUS TCP IP2 的通信超时标志	
SM8082	MODBUS TCP IP3 的通信超时标志	
SM8083	MODBUS TCP IP4 的通信超时标志	
SM8084	MODBUS TCP IP5 的通信超时标志	
SM8085	MODBUS TCP IP6 的通信超时标志	
SM8086	MODBUS TCP IP7 的通信超时标志	
SM8087	MODBUS TCP IP8 的通信超时标志	
SM8090	BKCMP 指令 块比较信号	
SM8091	BINDA 指令输出字符数切换信号	
SM8099	高速环形计数器（0.1ms 单位，16 位）动作	
SM8120	[COM0]RS2 发送等待标志位	
SM8121	[COM0]MODBUS 通信中；[COM0]RS2 发送请求	
SM8122	[COM0]MODBUS 通信发生错误；[COM0]RS2 接收结束标志位	

SM8123	[COMO]MODBUS 通信出错锁存	
SM8128	[COMO]MODBUS 主站发生重发事件的判断标志	
SM8129	[COMO]超时的判断标志位	
SM8130	HSZ(FNC55)指令表格比较模式	
SM8131	HSZ 指令表格比较执行结束标志位	
SM8132	HSZ、PLSY 指令频率控制模式	
SM8133	HSZ、PLSY 指令频率控制执行结束标志位	
SM8160	XCH 的 SWAP 功能	DXCH 交换功能不行，双字应该是交换各个字软元件的高低位，而实际执行指令之后，高位字变为 0，只有低位字在交换
SM8165	SORT2(FNC 149)指令降序排列	run→stop 清除
SM8168	SMOV(FNC 13)处理 HEX 数据的功能	run→stop 清除
SM8170	X000 脉冲捕捉(X1PLC 暂不支持)	run→stop 清除
SM8171	X001 脉冲捕捉(X1PLC 暂不支持)	
SM8172	X002 脉冲捕捉(X1PLC 暂不支持)	
SM8173	X003 脉冲捕捉(X1PLC 暂不支持)	
SM8174	X004 脉冲捕捉(X1PLC 暂不支持)	
SM8175	X005 脉冲捕捉(X1PLC 暂不支持)	
SM8179	BD 板使能配置	
SM8180	BD 板使能命令	
SM8195~SM8199	C251~C255 的 4 倍频切换	置 ON 时，切换为 4 倍频
SM8200~SM8234	C200~C234 计数模式控制位 ON:减计数模式 OFF:增计数模式	run→stop 不清除，断电清除
SM8235~SM8245	高速 C235~C245 的计数模式控制位 ON:减计数模式 OFF:增计数模式	run→stop 不清除，断电清除

SM8246~SM8255	高速计数器增减计数状态位 ON：减计数状态 OFF：增计数状态	run→stop 不清除，断电清除
SM8259	模块离线配置清除标志	置 ON 清除，清除后会复位
SM8261~SM8280	模拟量特殊适配器	
SM8304	零位，乘除运算结果为 0 时，置 ON	
SM8306	进位，除法运算结果溢出时，置 ON	
SM8329	指令执行异常结束	
SM8338	PLSY 指令加减速标志	
SM8336	DVIT 指令中断输入指定功能有效	
SM8340	[Y000] 脉冲输出中监控(ON:BUSY/OFF:READY)	
SM8341	[Y000] 清除信号输出功能有效	
SM8343	[Y000] 正转限位	
SM8344	[Y000] 反转限位	
SM8345	[Y000] 近点 DOG 信号逻辑反转	
SM8346	[Y000] 零点信号逻辑反转	
SM8347	[Y000] 中断信号逻辑反转	
SM8348	[Y000] 定位指令驱动中	
SM8349	[Y000] 脉冲输出停止指令	
SM8460	DVIT 指令[Y000] 用户中断输入指令	
SM8350	[Y001] 脉冲输出中监控(ON:BUSY/OFF:READY)	
SM8351	[Y001] 清除信号输出功能有效	
SM8352	[Y001] 指定原点回归方向	
SM8353	[Y001] 正转限位	
SM8354	[Y001] 反转限位	
SM8355	[Y001] 近点 DOG 信号逻辑反转	
SM8356	[Y001] 零点信号逻辑反转	

SM8357	[Y001] 中断信号逻辑反转	
SM8358	[Y001] 定位指令驱动中	
SM8359	[Y001] 脉冲输出停止指令	
SM8461	DVIT 指令[Y001] 用户中断输入指令	
SM8398	1ms 的环形计数器（32 位 SD8399,SD8398）动作	
SM8401	[COM1]MODBUS 通信中；[COM1]RS2 发送等待标志位	
SM8402	[COM1]MODBUS 通信发生错误；[COM1]RS2 发送请求	
SM8403	[COM1]MODBUS 通信出错锁存；[COM1]RS2 接收结束标志位	
SM8408	[COM1]MODBUS 主站发生重发事件的判断标志	
SM8409	[COM1]超时的判断标志	
SM8421	[COM2]MODBUS 通信中；[COM2]RS2 发送等待标志位	
SM8422	[COM2]MODBUS 通信发生错误；[COM2]RS2 发送请求	
SM8423	[COM2]MODBUS 通信出错锁存；[COM2]RS2 接收结束标志位	
SM8428	[COM2]MODBUS 主站发生重发事件的判断标志	
SM8429	[COM2]超时的判断标志	
SM8438	[COM2]串行通信出错锁存	
SM8499	设定延迟时间用的触点	和 SD8499（设定延时时间）配合使用
SM8500	MODBUS TCP 通信时从站不在线标志位	
SM8501	MODBUS TCP 通信时主站网线未连接或断开标志位	

SM8502	以太网功能控制位	
SM8503	MODBUS TCP 功能控制位	
SM8511	上位机形式的 MODBUS 配置生效标志	

1.2.2 特殊数据寄存器 SD8（D8）

寄存器号	功能	说明
SD8000	看门狗定时器（默认 200），单位 ms	默认 200
SD8001	PLC 类型及系统版本（兼容三菱用）	默认 24320
SD8002	内存容量 ，固定为 4（兼容三菱用）	默认 4（32K 步）
SD8004	错误 M 编码	
SD8005	电池电压，单位 0.1v	默认 33
SD8006	检测出电池电压低的等级，单位 0.1v	默认 27
SD8010	当前扫描时间（0.1ms）	
SD8011	最小扫描时间（0.1ms）	
SD8012	最大扫描时间（0.1ms）	
SD8013	实时时钟（秒）	
SD8014	实时时钟（分）	
SD8015	实时时钟（时）	
SD8016	实时时钟（日）	
SD8017	实时时钟（月）	
SD8018	实时时钟（年）	
SD8019	实时时钟（星期）	
SD8020	输入滤波器值	默认 10
SD8028	Z0 寄存器的内容	
SD8029	V0 寄存器的内容	
SD8039	恒定扫描时间设定（ms）	
SD8040	ON 状态编号 1	状态 S0~S899,S1000~S4095 中正在动作的状态的最小
SD8041	ON 状态编号 2	

SD8042	ON 状态编号 3	编号保存到 SD8040 中。其次为 ON 的状态保存到紧接着的寄存器，最大八点，默认-1
SD8043	ON 状态编号 4	
SD8044	ON 状态编号 5	
SD8045	ON 状态编号 6	
SD8046	ON 状态编号 7	
SD8047	ON 状态编号 8	
SD8049	ON 状态最小编号（SM8049 有效时）	SM8049 为 ON 时，保存信号报警继电器 S900~S999 中为 ON 的状态的最小编号 ，默认-1
SD8061	PLC 硬件错误的错误代码编号	
SD8062	COM0 串行通信错误代码编号	
SD8063	COM1 串行通信错误代码编号	
SD8064	参数错误的错误代码编号	
SD8065	语法错误的错误代码编号	
SD8066	梯形图错误的错误代码编号	
SD8067	运算错误的错误代码编号	
SD8068	发生运算错误的步编号的锁存	
SD8069	语法、回路、运算错误的错误步编号	
SD8093	[COM0]请求间延迟	
SD8094	[COM0]设定进行重发的次数	
SD8095	[COM0]从站本站号	
SD8099	0~32767（0.1ms 单位 16 位）的递增动作的环形计数器	
SD8101	PLC 类型及系统版本（兼容三菱用）	默认 16320
SD8102	内存容量 2N 固定为 16（兼容三菱用）	默认 64
SD8110	XPLC 软件版本号	默认 1xxx（1 表示 X1 系列，xxx 表示固件版本）
SD8111	XPLC 点数	高八位是 X 点数，低八位是

		Y 点数
SD8120	[COM0]通信格式设定	用于 COM 口的通信配协议置，详细使用参考 3.4 章节
SD8121	[COM0]协议	
SD8122	[COM0]MODBUS 通信错误寄存器；[COM0]RS2 发送数据的剩余点数	
SD8123	[COM0]MODBUS 通信错误锁存寄存器；[COM0]RS2 接收点数的监控	
SD8124	[COM0]RS2 报头 1、2	
SD8125	[COM0]RS2 报头 3、4	
SD8126	[COM0]RS2 报尾 1、2	
SD8127	[COM0]RS2 报尾 3、4	
SD8128	[COM0]显示主站当前的重发次数	
SD8129	[COM0]设定响应超时时间	
SD8130	HSZ 比较表格计数器	(X1PLC 暂不支持)
SD8131	HSZ、PLSY 频率表格计数器	(X1PLC 暂不支持)
SD8132	HSZ、PLSY 频率控制模式频率低位	(X1PLC 暂不支持)
SD8133	HSZ、PLSY 频率控制模式频率高位	(X1PLC 暂不支持)
SD8134	HSZ、PLSY 频率控制模式目标脉冲数低位	(X1PLC 暂不支持)
SD8135	HSZ、PLSY 频率控制模式目标脉冲数高位	(X1PLC 暂不支持)
SD8140	PLSR、PLSY 输出到 Y0 脉冲数低位	
SD8141	PLSR、PLSY 输出到 Y0 脉冲数高位	
SD8142	PLSR、PLSY 输出到 Y1 脉冲数低位	
SD8143	PLSR、PLSY 输出到 Y1 脉冲数高位	
SD8182	Z1 寄存器的内容	
SD8183	V1 寄存器的内容	
SD8184	Z2 寄存器的内容	
SD8185	V2 寄存器的内容	
SD8186	Z3 寄存器的内容	

SD8187	V3 寄存器的内容	
SD8188	Z4 寄存器的内容	
SD8189	V4 寄存器的内容	
SD8190	Z5 寄存器的内容	
SD8191	V5 寄存器的内容	
SD8192	Z6 寄存器的内容	
SD8193	V6 寄存器的内容	
SD8194	Z7 寄存器的内容	
SD8195	V7 寄存器的内容	
SD8219~SD8238	BD 板使用的 SD8 寄存器	
SD8312	发生运算出错的步编号的锁存（32bit 低位）	
SD8313	发生运算出错的步编号的锁存（32bit 高位）	
SD8314	SM8065~SM8067 的错误步编号（32bit 低位）	
SD8315	SM8065~SM8067 的错误步编号（32bit 高位）	
SD8336	DVIT 中断输入指定（32bit 低位）	
SD8337（60 点）	DVIT 中断输入指定（32bit 高位）	
SD8340	[Y000]当前寄存器值（32bit 低位）	
SD8341	[Y000]当前寄存器值（32bit 高位）	
SD8342	[Y000]基底速度	默认 800（HZ）
SD8343	[Y000]最高速度（32bit 低位）	默认 200000（HZ）
SD8344	[Y000]最高速度（32bit 高位）	
SD8345	[Y000]爬行速度	默认 1000（HZ）
SD8346	[Y000]原点回归速度（32bit 低位）	默认 50000（HZ）
SD8347	[Y000]原点回归速度（32bit 高位）	
SD8348	[Y000]最大加速时间	默认 100（ms）
SD8349	[Y000]最大减速时间	默认 100（ms）

SD8350	[Y001]当前寄存器值（32bit 低位）	
SD8351	[Y001]当前寄存器值（32bit 高位）	
SD8352	[Y001]基底速度	默认 800（HZ）
SD8353	[Y001]最高速度（32bit 低位）	默认 200000（HZ）
SD8354	[Y001]最高速度（32bit 高位）	
SD8355	[Y001]爬行速度	默认 1000（HZ）
SD8356	[Y001]原点回归速度（32bit 低位）	默认 50000（HZ）
SD8357	[Y001]原点回归速度（32bit 高位）	
SD8358	[Y001]最大加速时间	默认 100（ms）
SD8359	[Y001]最大减速时间	默认 100（ms）
SD8398	0~2147483647（1ms 单位）的递增环形计数器	
SD8399		
SD8400	[COM1]通信格式设定	
SD8401	[COM1]协议	
SD8402	[COM1]MODBUS 通信错误寄存器；[COM1]RS2 发送数据的剩余点数	
SD8403	[COM1]MODBUS 通信错误锁存寄存器；[COM1]RS2 接收点数的监控	
SD8405	[COM0]RS2 接收和校验(接收数据)	
SD8408	[COM1]显示主站当前的重发次数	
SD8409	[COM1]设定响应超时时间	
SD8410	[COM1]RS2 报头 1、2	
SD8411	[COM1]MODBUS 请求间延迟；[COM1]RS2 报头 3、4	
SD8412	[COM1]MODBUS 设定进行重发的次数；[COM1]RS2 报尾 1、2	
SD8413	[COM1]RS2 报尾 3、4	
SD8414	[COM1]MODBUS 从站本站号；[COM1]RS2 接收和校验(接收数据)	
SD8415	[COM1]RS2 接收和校验(计算结果)	

SD8416	[COM1]RS2 发送和校验	
SD8418	[COM0]RS2 接收和校验(计算结果)	
SD8419	[COM0]RS2 发送和校验	
SD8420	[COM2]通信格式设定	
SD8421	[COM2]协议	
SD8422	[COM2]MODBUS 通信错误寄存器； [COM2]RS2 发送数据的剩余点数	
SD8423	[COM2]MODBUS 通信错误锁存寄存器； [COM2]RS2 接收点数的监控	
SD8428	[COM2]显示主站当前的重发次数	
SD8429	[COM2]设定响应超时时间	
SD8430	[COM2]RS2 报头 1、2	
SD8431	[COM2]MODBUS 请求间延迟； [COM2]RS2 报头 3、4	
SD8432	[COM2]MODBUS 设定进行重发的次数； [COM2]RS2 报尾 1、2	
SD8433	[COM2]RS2 报尾 3、4	
SD8434	[COM2]MODBUS 从站本站号； [COM2]RS2 接收和校验(接收数据)	
SD8435	[COM2]RS2 接收和校验(计算结果)	
SD8436	[COM2]RS2 发送和校验	
SD8438	[COM2]错误代码编号	
SD8449	特殊模块错误代码	
SD8464	FNC.150.156 指令 Y000 指定清除信号软元件	默认 6（Y6）
SD8465	FNC.150.156 指令 Y001 指定清除信号软元件	默认 7（Y7）
SD8499	设定延迟时间（ms 单位）	
SD8500	显示 PLC 当前分配协议连接个数	以太网专用寄存器，具体使用见 3.6 章节
SD8501	不在线从站对应的 IP（第四段）	
SD8502	显示 MODBUS TCP 错误码）	

SD8503	设定以太网主站接收超时时间（ms 单位）	
SD8504	显示当前 TCP 连接数量	
SD8505	MAC 地址烧录用（用户不可用）	
SD8506	存储接收超时的从站 IP 及错误位置	
SD8507	尝试重连时间间隔（ms 单位）	
SD8508	本机 IP 第一段	
SD8509	本机 IP 第二段	
SD8510	本机 IP 第三段	
SD8511	本机 IP 第四段	

特别说明：

（1）通信专用寄存器

其中 MODBUS 和 RS2 通信寄存器有的是共用的，具体使用可查看 3.4 章节。

（2）标注 60 点只有 60 点 PLC 才能使用，其它点数的 PLC 中无效。

二 、指令篇

X3 支持使用的指令集如下表。

基本指令		
指令	功能号	指令说明
LD	/	装载
LDI	/	装载取反
LDP	/	取脉冲上升沿
LDF	/	取脉冲下降沿
AND	/	与
ANI	/	与非
ANDP	/	与脉冲上升沿
ANDF	/	与脉冲下降沿
OR	/	或

ORI	/	或非
ORP	/	或脉冲上升沿
ORF	/	或脉冲下降沿
MEP	/	运算结果上升沿脉冲化指令
MEF	/	运算结果下降沿脉冲化指令
ANB	/	回路块与
ORB	/	回路块或
MPS	/	存储进栈
MRD	/	存储读栈
MPP	/	存储出栈
INV	/	取反
OUT	/	输出
SET	/	置位
RST	/	复位
PLS	/	上升沿脉冲检测
PLF	/	下降沿脉冲检测
MC	/	主控
MCR	/	主控复位
NOP	/	空操作
END	/	程序结束
STL	/	步进梯形图开始
RET	/	步进梯形图结束
程序流程指令		
指令	功能号	说明
CJ	00	条件跳转
CALL	01	子程序调用
SRET	02	子程序返回
IRET	03	中断返回
EI	04	允许中断

DI	05	禁止中断
FEND	06	主程序结束
WDT	07	看门狗定时器
FOR	08	循环范围的开始
NEXT	09	循环范围的结束
传送比较指令		
指令	功能号	说明
CMP	10	比较
ZCP	11	区间比较
MOV	12	传送
SMOV	13	移位
CML	14	取反传送
BMOV	15	成批传送
FMOV	16	多点传送
XCH	17	交换
BCD	18	BCD 转换
BIN	19	BIN 转换
四则逻辑运算指令		
指令	功能号	说明
ADD	20	BIN 加法
SUB	21	BIN 减法
MUL	22	BIN 乘法
DIV	23	BIN 除法
INC	24	BIN 加 1
DEC	25	BIN 减 1
WAND	26	逻辑与
WOR	27	逻辑或
WXOR	28	逻辑异或
NEG	29	求补码

循环移位指令		
指令	功能号	说明
ROR	30	循环右移
ROL	31	循环左移
RCR	32	带进位循环右移
RCL	33	带进位循环左移
SFTR	34	位右移
SFTL	35	位左移
WSFR	36	字右移
WSFL	37	字左移
SFWR	38	移位写入
SFRD	39	移位读出
数据处理指令		
指令	功能号	说明
ZRST	40	成批复位
DECO	41	译码
ENCO	42	编码
SUM	43	ON 位总数
BON	44	ON 的判定
MEAN	45	平均值
ANS	46	信号报警器置位
ANR	47	信号报警器复位
SQR	48	BIN 开方
FLT	49	BIN 整数→2 进制浮点数转换
高速处理指令		
指令	功能号	说明
REF	50	输入输出刷新
REFF	51	滤波器调整
HSCS	53	比较置位（高速计数器）

HSCR	54	比较复位（高速计数器）
HSZ	55	区间比较（高速计数器）
SPD	56	脉冲密度 （X1PLC 不支持）
PLSY	57	脉冲输出
PWM	58	脉宽调制
PLSR	59	带加减速的脉冲输出
通信指令		
指令	功能号	指令说明
RS2	87	自由口通信
ADPRW	276	MODBUS 主站通信指令
浮点运算指令		
指令	功能号	说明
ECMP	110	2 进制浮点数比较
EZCP	111	2 进制浮点数区间比较
EMOV	112	2 进制浮点数数据传送
ESTR	116	2 进制浮点数→字符串的转换
EVAL	117	字符串→2 进制浮点数的转换
EBCD	118	2 进制浮点数→10 进制浮点数转换
EBIN	119	10 进制浮点数→2 进制浮点数转换
EADD	120	2 进制浮点数加法运算
ESUB	121	2 进制浮点数减法运算
EMUL	122	2 进制浮点数乘法运算
EDIV	123	2 进制浮点数除法运算
EXP	124	2 进制浮点数指数运算
LOGE	125	2 进制浮点数自然对数运算
LOG10	126	2 进制浮点数常用对数运算
ESQR	127	2 进制浮点数开方运算
ENEG	128	2 进制浮点数符号翻转
INT	129	2 进制浮点数→BIN 整数的转换

SIN	130	2 进制浮点数 SIN 运算
COS	131	2 进制浮点数 COS 运算
TAN	132	2 进制浮点数 TAN 运算
ASIN	133	2 进制浮点数 SIN^{-1} 运算
ACOS	134	2 进制浮点数 COS^{-1} 运算
ATAN	135	2 进制浮点数 TAN^{-1} 运算
RAD	136	2 进制浮点数角度→弧度的转换
DEG	137	2 进制浮点数弧度→角度的转换
数据处理 2		
指令	功能号	指令说明
WSUM	140	算出数据合计值
WTOB	141	字节单位的数据分离
BTOW	142	字节单位的数据结合
UNI	143	16 位数据的 4 位结合
DIS	144	16 位数据的 4 位分离
SWAP	147	高低字节互换
SORT2	149	数据排序 2
定位		
指令	功能号	指令说明
DSZR	150	带 dog 搜索的原点回归
DVIT	151	中断定位
ZRN	156	原点回归
PLSV	157	可变速的脉冲输出
DRVI	158	相对位置控制
DRVA	159	绝对位置控制
时钟运算		
指令	功能号	说明
TCMP	160	时钟数据比较
TZCP	161	时钟数据区间比较

TADD	162	时钟数据加法运算
TSUB	163	时钟数据减法运算
HTOS	164	时、分、秒数据的秒转换
STOH	165	秒数据的[时、分、秒]转换
TRD	166	时钟数据的读出
TWR	167	时钟数据的写入
HOUR	169	长时间计时
格雷码		
指令	功能号	指令说明
GRY	170	格雷码的转换
GBIN	171	格雷码的逆转换
数据块指令		
指令	功能号	指令说明
BK+	192	数据块的加法运算
BK-	193	数据块的减法运算
BKCMP=	194	数据块比较
BKCMP>	195	
BKCMP<	196	
BKCMP<>	197	
BKCMP<=	198	
BKCMP>=	199	
字符串控制指令		
指令	功能号	指令说明
STR	200	BIN→字符串的转换
VAL	201	字符串→BIN 的转换
$+	202	字符串的结合
LEN	203	检测出字符串的长度
RIGHT	204	从字符串的右侧开始取出
LEFT	205	从字符串的左侧开始取出

MIDR	206	从字符串中的任意取出
MIDW	207	字符串中的任意替换
INSTR	208	字符串的检索
$MOV	209	字符串的传送
数据处理指令 3		
指令	功能号	指令说明
FDEL	210	数据表的数据删除
FINS	211	数据表的数据插入
POP	212	读取后入的数据[先入后出控制用]
SFR	213	16 位数据 n 位右移(带进位)
SFL	214	16 位数据 n 位左移(带进位)
触点比较指令		
指令	功能号	说明
LD=	224	(S1)=(S2)
LD>	225	(S1)>(S2)
LD<	226	(S1)<(S2)
LD<>	228	(S1)!=(S2)
LD<=	229	(S1)<=(S2)
LD>=	230	(S1)>=(S2)
AND=	232	(S1)=(S2)
AND>	233	(S1)>(S2)
AND<	234	(S1)<(S2)
AND<>	236	(S1)!=(S2)
AND<=	237	(S1)<=(S2)
AND>=	238	(S1)>=(S2)
OR=	240	(S1)=(S2)
OR>	241	(S1)>(S2)
OR<	242	(S1)<(S2)
OR<>	244	(S1)!=(S2)

OR<=	245	(S1)<=(S2)
OR>=	246	(S1)>=(S2)
数据表处理		
指令	功能号	指令说明
LIMIT	256	上下限限位控制
BAND	257	死区控制
ZONE	258	区域控制
SCL	259	定坐标(不同点坐标数据)
DABIN	260	10 进制 ASCII→BIN 的转换
BINDA	261	BIN→10 进制 ASCII 的转换
SCL2	269	定坐标 2(X/Y 坐标数据)
扩展文件寄存器		
指令	功能号	指令说明
LOADR	290	读出扩展文件寄存器
SAVER	291	成批写入扩展文件寄存器
INITR	292	扩展寄存器的初始化
LOGR	293	登录到扩展寄存器
RWER	294	扩展文件寄存器的删除・写入
INITER	295	扩展文件寄存器的初始化
方便指令		
指令	功能号	指令说明
ALT	66	交替输出
SORT	69	数据排序

注意：

（1）若使用三菱上位机编程（原则上不允许，但是 XPLC 兼容 GX 上位机），程序中出现未实现的指令会报停机错误 6506。

（2）指令后面+P 代表边沿触发，指令前面+D 代表 32 位指令。

（3）当指令操作数为 32 位时，其本身组成的字元件（低 16 位）和相邻的一个字元件（高 16 位）共同组合成这个 32 位数，在指令使用过程中一定注意区别，且 32 位指令最好以偶数地址作为起始元件。

2.1 基本指令

以梯形图方式编程时，有些基本指令是不可见的，只有转换成指令列表才可以查看。

基本指令大致分五类：

1、触点指令：一般放在梯形图中紧接母线的地方，根据触发方式的不同来作为后面指令执行的一个动作开关。

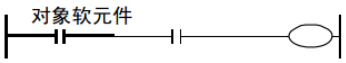
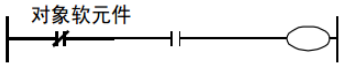
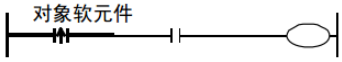
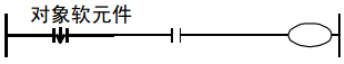
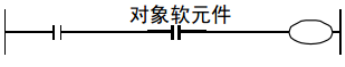
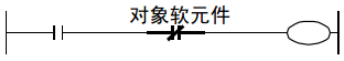
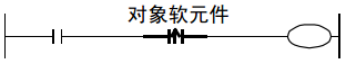
2、结合指令：连接梯形图形成多个执行逻辑。

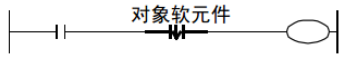
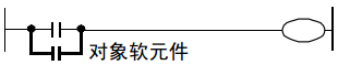
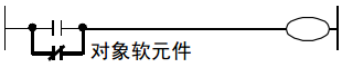
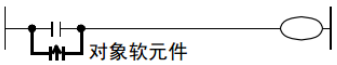
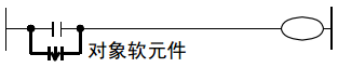
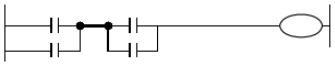
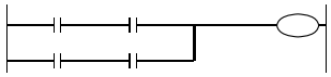
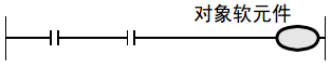
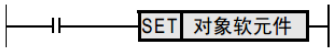
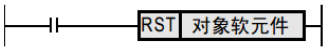
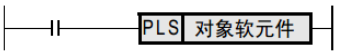
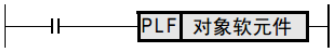
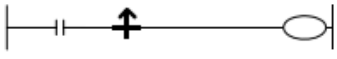
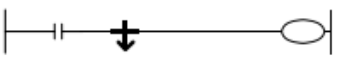
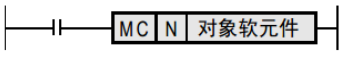
3、输出指令：根据输出方式给对象软元件赋初值。

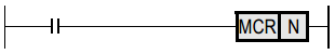
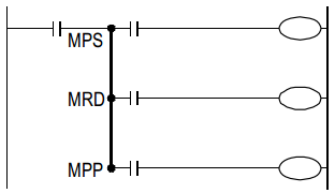
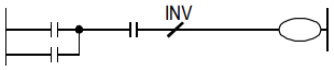
4、主控指令：控制一个程序块的开启关闭。

5、结束指令：END，放于梯形图末尾；FEND 用于主程序结束，会执行与 END 指令相同的输出处理、输入处理、看门狗定时器的刷新，然后返回到 0 步的程序，在编写子程序和中断程序时需要使用 FEND 指令。

基本指令表如下（a 触点：常开触点；b 触点：常闭触点）：

记号	称呼	符号	功能	对象软元件
LD	取		a 触点的逻辑运算开始	X,Y,M,S,T,C,Dx.y
LDI	取反		b 触点的逻辑运算开始	X,Y,M,S,T,C,Dx.y
LDP	取脉冲上升沿		检测到上升沿运算开始	X,Y,M,S,T,C,Dx.y
LDF	取脉冲下降沿		检测到下降沿运算开始	X,Y,M,S,T,C,Dx.y
AND	与		串联 a 触点	X,Y,M,S,T,C,Dx.y
ANI	与反转		串联 b 触点	X,Y,M,S,T,C,Dx.y
ANDP	与脉冲上升沿		上升沿检出的串联连接	X,Y,M,S,T,C,Dx.y

ANDF	与脉冲下降沿		下降沿检出的串联连接	X,Y,M,S,T,C,Dx.y
OR	或		并联 a 触点	X,Y,M,S,T,C,Dx.y
ORI	或反转		并联 b 触点	X,Y,M,S,T,C,Dx.y
ORP	或脉冲上升沿		上升沿检出的并联连接	X,Y,M,S,T,C,Dx.y
ORF	或脉冲下降沿		下降沿检出的并联连接	X,Y,M,S,T,C,Dx.y
ANB	回路块与		回路块的串联连接	-
ORB	回路块或		回路块的并联连接	-
OUT	输出		线圈驱动	Y,M,S,T,C,Dx.y
SET	置位		动作保持	Y,M,S,Dx.y
RST	复位		解除保持的动作，清除当前值及寄存器	Y,M,S,T,C,,Dx.y D,R,V,Z
PLS	脉冲		上升沿微分输出	Y,M
PLF	下降沿脉冲		下降沿微分输出	Y,M
MEP	运算结果上升沿脉冲化指令		运算结果的上升沿时为 ON	-
MEF	运算结果下降沿脉冲化指令		远算结果的下降沿时为 ON	-
MC	主控		连接到公共触点	Y,M

MCR	主控复位		解除连接到 公共触点	-
MPS	存储进栈		压入堆栈	-
MRD	存储读栈		读取堆栈	-
MPP	存储出栈		弹出堆栈	-
INV	反转		运算结果的反转	-
NOP	空操作		无处理	-
EDN	结束		程序结束以及 输入输出处理 和返回 0 步	-

2.1.1 LD、LDI/取、取反指令

1、概要

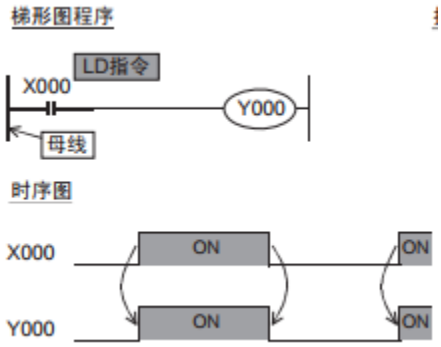
与母线相连的常开触点和常闭触点。

2、操作数可用元件类型

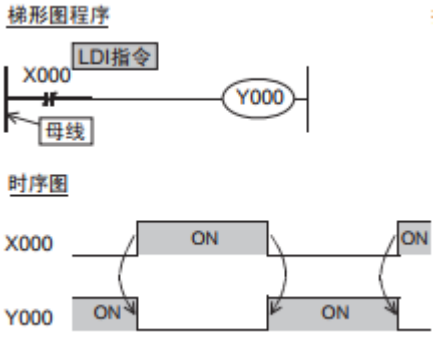
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
LD	*	*	*	*	*	*	*											*					
LDI	*	*	*	*	*	*	*											*					

3、功能和动作说明

（1）LD 指令（常开触点）

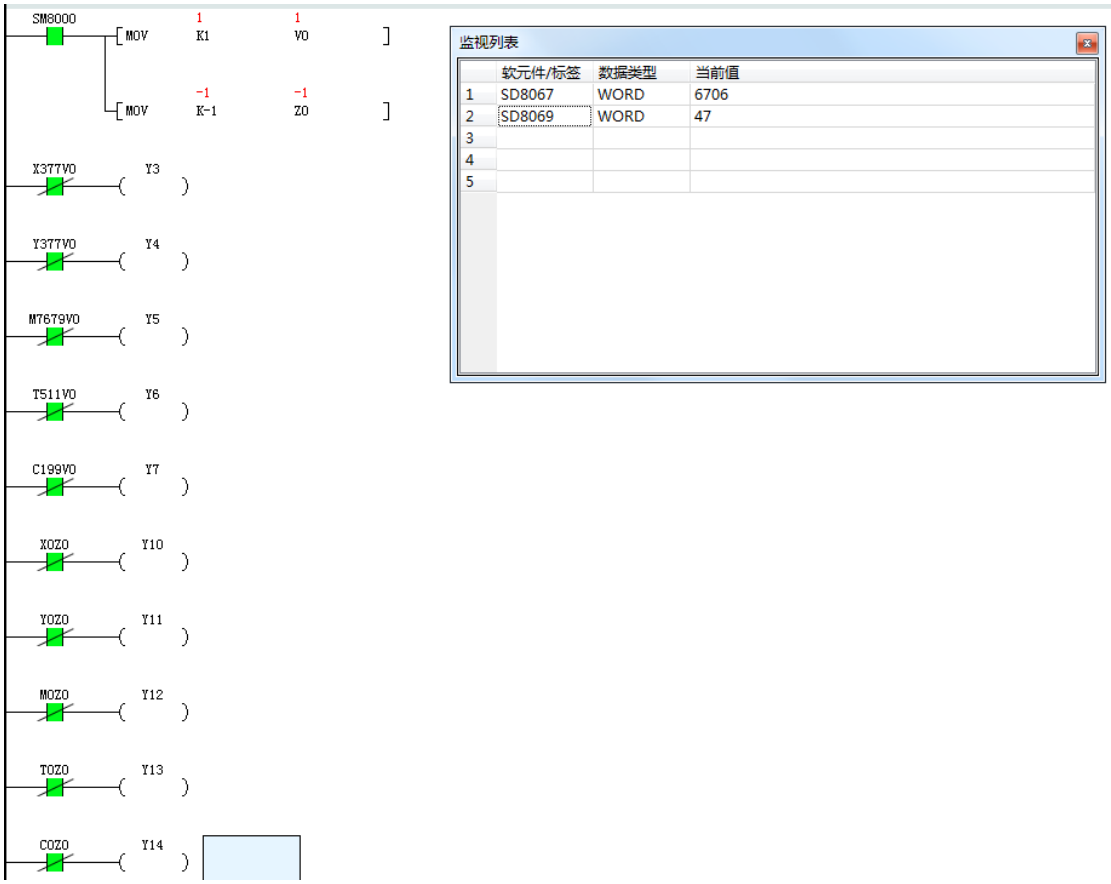


（2）LDI 指令（常闭触点）



4、注意事项

使用触点类指令，要注意变址修饰越界的情况，以 LDI 为例，LDI+位软元件变址操作，当位软元件变址操作越界后，XPLC 监控返回 0,但在执行 LDI 时报错返回，不执行取反操作。（XPLC 对于指令中出现的错误问题，只要出错则指令不执行，即输出 0）



此梯形图为变址越界时的执行情况，输出 Y 没有点亮，正常情况下 Y 会点亮。

2.1.2 OUT/输出指令

1、概要

OUT 指令是对输出继电器(Y)、辅助继电器(M)、状态(S)、定时器(T)、计数器(C)进行线圈驱动的指令。

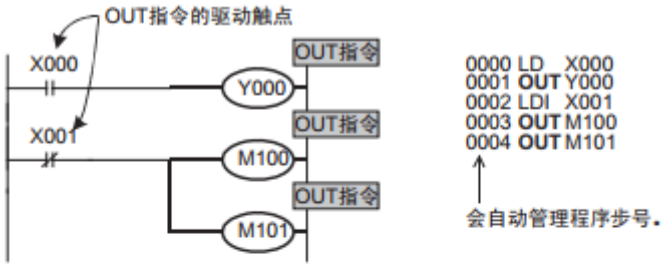
2、操作数可用元件类型

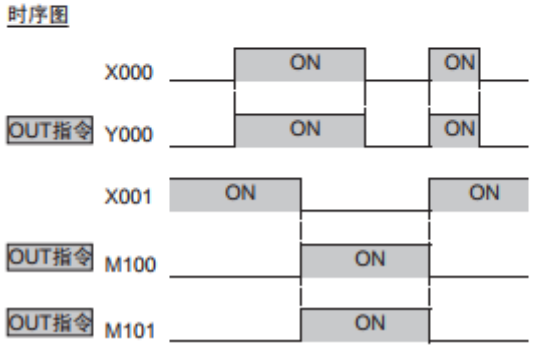
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
OUT		*	*	*	*	*	*											*					
设定值														*	*			*	*				

3、功能和动作说明

（1）使用位元件时

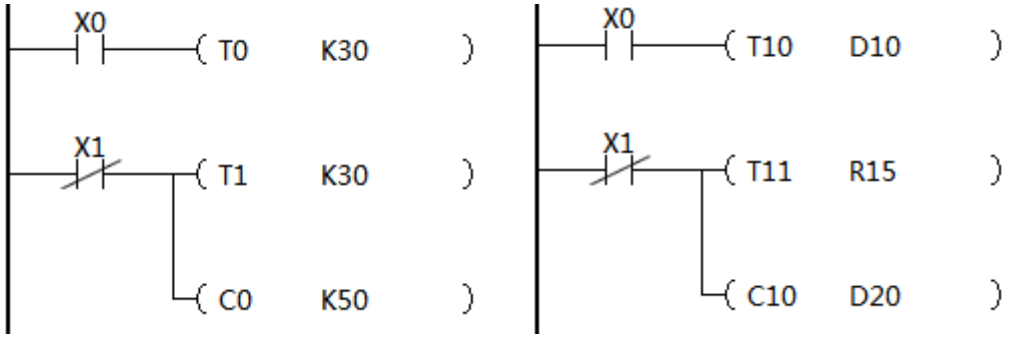
用 OUT 指令编写的软元件，根据驱动触点的状态执行 ON/OFF。





（2）使用定时器和计数器

在针对定时器的计时线圈和计数器的计数线圈的 OUT 指令后需要加上设定值。设定值可以使用 10 进制数(K)直接指定，也可以使用数据寄存器(D)或扩展寄存器(R)间接指定。



4、注意事项

（1）定时器和计数器的设定值的设定范围以及实际的定时器常数(包含设定值)参照 1.1 章节的资源配置表。

（2）当变址后对应的软元件不存在时报错 6706，且变址后的软元件不能为对应的高速计数器。

2.1.3 AND、ANI/与、与反转指令

1、概要

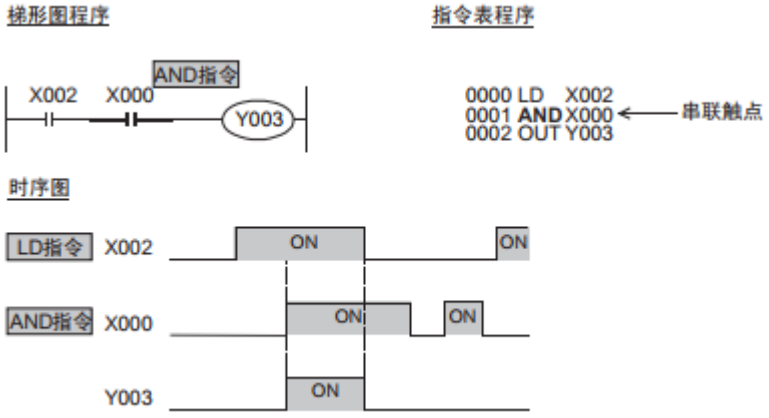
AND、ANI 指令用于串联连接触点。串联触点的数量没有限制，该指令可以连续多次使用。

2、操作数可用元件类型

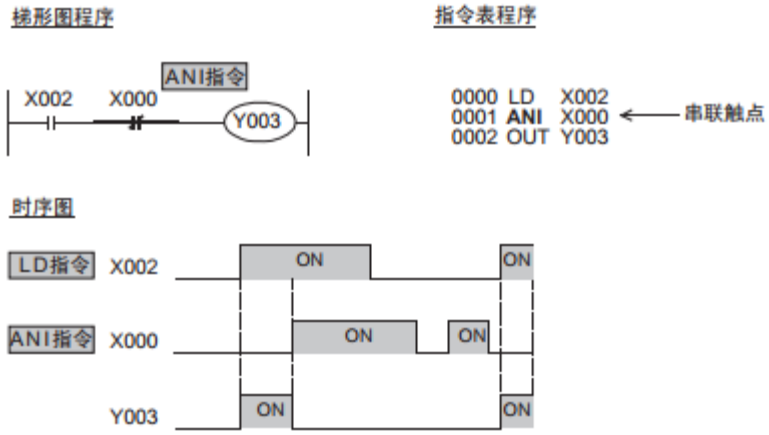
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
AND	*	*	*	*	*	*	*											*					
ANI	*	*	*	*	*	*	*											*					

3、功能和动作说明

（1）AND 指令（串联常开触点）



（2）ANI 指令（串联常闭触点）



4、注意事项

要注意变址修饰越界的情况，参考 LDI 指令。

2.1.4 OR、ORI/或、或反转指令

1、概要

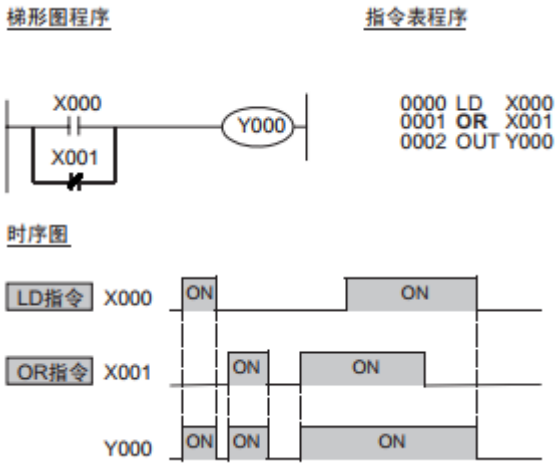
OR、ORI 指令用于并联连接触点。

2、操作数可用元件类型

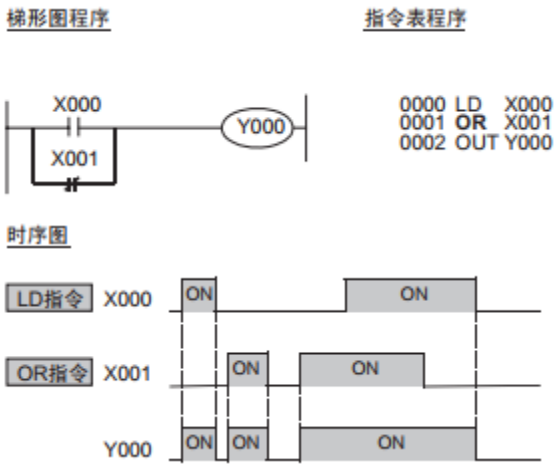
指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
OR	*	*	*	*	*	*	*											*					
ORI	*	*	*	*	*	*	*											*					

3、功能和动作说明

（1）OR 指令（并联常开触点）



（2）ORI 指令（并联常闭触点）



4、注意事项

要注意变址修饰越界的情况，参考 LDI 指令。

2.1.5 LDP、LDF、ANDP、ORP、ORF/沿触发类指令

1、概要

LDP、ANDP、ORP 指令是检测上升沿的触点指令，仅在指定位软元件的上升沿(从 OFF 改变到 ON 的时候)时，接通 1 个运算周期。

LDF、ANDF、ORF 指令是检测下降沿的触点指令，仅在指定位软元件的下降沿(从 ON 改变到 OFF)时，接通 1 个运算周期。

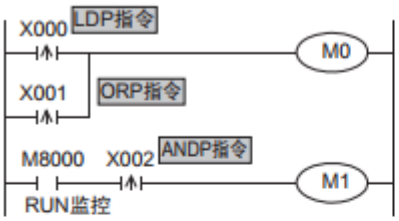
2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
LDP	*	*	*	*	*	*	*																
LDF	*	*	*	*	*	*	*																
ANDP	*	*	*	*	*	*	*																
ANDF	*	*	*	*	*	*	*																
ORP	*	*	*	*	*	*	*																
ORF	*	*	*	*	*	*	*																

3、功能和动作说明

（1）LDP、ANDP、ORP 指令（检测到上升沿时运算开始、串联连接、并联连接）

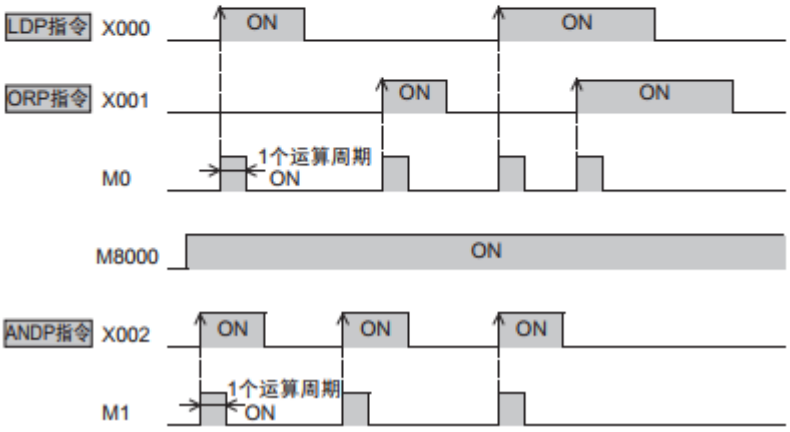
梯形图程序



指令表程序

```
0000 LDP  X000
0002 ORP  X001
0004 OUT  M0
0005 LD   M8000
0006 ANDP X002
0008 OUT  M1
```

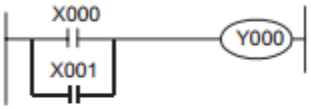
时序图



当 X000～X002 从 OFF 变成 ON 时，M0 或 M1 只维持 1 个运算周期为 ON。

（2）LDF、ANDF、ORF 指令（检测到下升沿时运算开始、串联连接、并联连接）

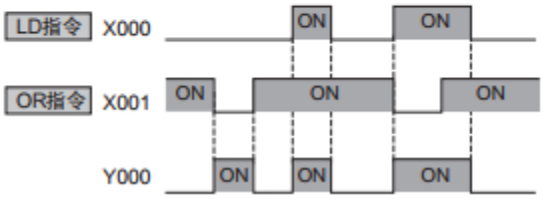
梯形图程序



指令表程序

```
0000 LD   X000
0001 ORI  X001
0002 OUT Y000
```

时序图



当 X000～X002 从 ON 变成 OFF 时，M0 或 M1 只维持 1 个运算周期为 ON。

4、注意事项

（1）要注意变址修饰越界的情况，参考 LDI 指令。

（2）脉冲触发类指令一般不适用于中断子程序中。

（3）脉冲触发指令一般不能在同一行梯形图中连续使用。

2.1.6 ORB/回路块或指令

1、概要

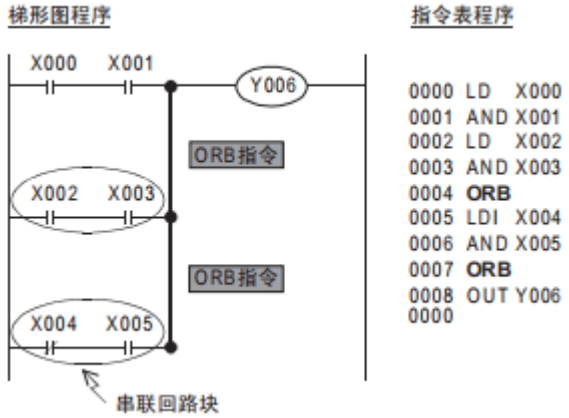
由 2 个以上的触点串联连接的回路称为串联回路块。

2、操作数可用元件类型

无。

3、功能和动作说明

并联连接串联回路块时，分支的起点使用 LD、LDI 指令，分支的结束使用 ORB 指令。ORB 指令，与后述的 ANB 指令等相同，都是不带软元件编号的独立指令。有多个并联回路时，在每个回路块中使用 ORB 指令，从而连接。



4、注意事项

ORB 指令可以无限使用，一般都是和 LD、LDI 连用，但是 LD、LDI 指令重复次数在 8 次以下，因此要注意。

2.1.7 ANB/回路块与指令

1、概要

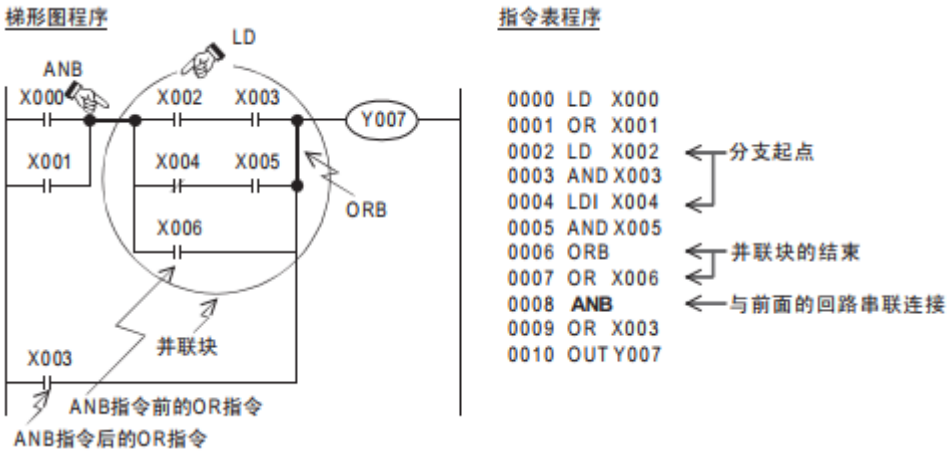
当分支回路(并联回路块)与前面的回路串联连接时，使用 ANB 指令。

2、操作数可用元件类型

无。

3、功能和动作说明

分支的起点使用 LD、LDI 指令，并联回路块结束后，可以使用 ANB 指令和前面的回路串联连接。有多个并联回路的时候，对每个回路块使用 ANB 指令，从而连接。



4、注意事项

ANB 指令可以无限使用，一般都是和 LD、LDI 连用，但是 LD、LDI 指令重复次数在 8 次以下，因此要注意。

2.1.8 MPS、MRD、MPP/堆栈类指令

1、概要

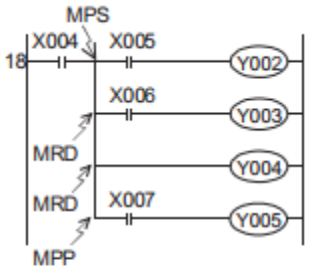
堆栈的内存，用于记忆运算的中间结果(ON 或 OFF)。三个指令分别表示压入堆栈、读取堆栈、弹出堆栈。

2、操作数可用元件类型

略。

3、功能和动作说明

这些指令是用于编写多重分支输出回路的便捷指令。

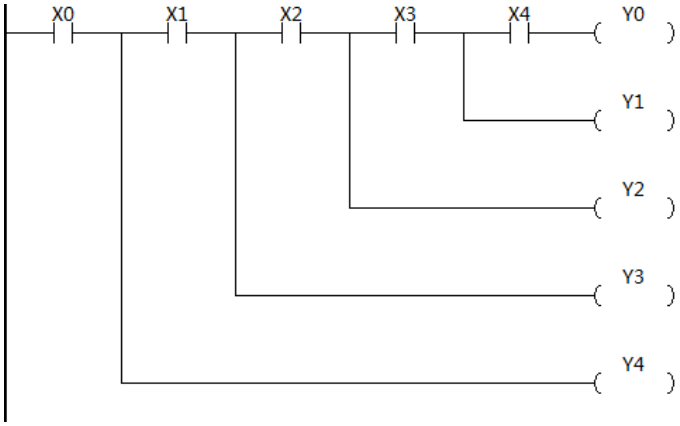


①使用 MPS 指令存储运算的中间结果后，驱动输出 Y002。
②使用 MRD 指令读取该存储内容后，驱动输出 Y003，MRD 指令可以多次编程。
③在最终输出回路中使用 MPP 指令替代 MRD 指令，从而在读出上述存储内容的同时将其复位。

4、注意事项

使用堆栈指令时要注意梯形图的执行顺序，一般情况下（无程序跳转指令和中断发生）都是从上到下，从左到右的顺序执行梯形图。

比如：



当 X1 置 ON 时，如果此时置位 X0，则应该 OUT Y3 先执行，OUT Y4 后执行。

2.1.9 MC、MCR/主控、主控复位指令

1、概要

执行 MC 指令后，母线(LD、LDI 点)移动到 MC 触点之后，开始执行 MC 和 MCR 之间的程序。使用 MCR 指令，可以将其返回原来的母线位置。

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
MC		*	*																				

3、功能和动作说明

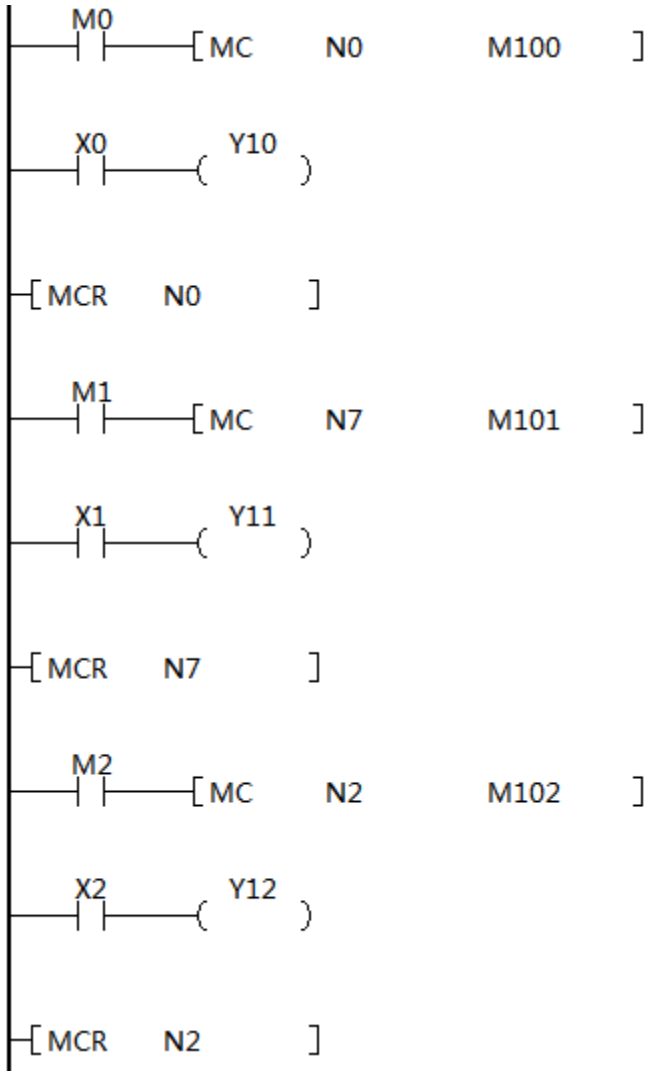
执行 MC 指令后，母线移动到 MC 触点之后。MC 触点后的母线上连接的驱动指令，只在 MC 指令执行时才执行各个动作，不执行 MC 指令时，无论驱动指令之前的触点状态如何，均为 OFF 执行(与触点 OFF 时的动作相同)。

在使用了主控的回路中，触点指令中如果有不需要触发条件的指令(FOR、NEXT、EI、DI 指令等)，则执行该指令，而与 MC 指令的执行指令无关。在如下所示的程序举例中，当输入条件为 ON 时，则执行从 MC 到 MCR 的指令，但是当输入条件为 OFF 时，各个驱动软元件的动作如下。

变为 OFF 的软元件：定时器(累计定时器除外)，用 OUT 指令驱动的软元件
保持状态的软元件：累计定时器、计数器、用 SET/RST 指令驱动的软元件

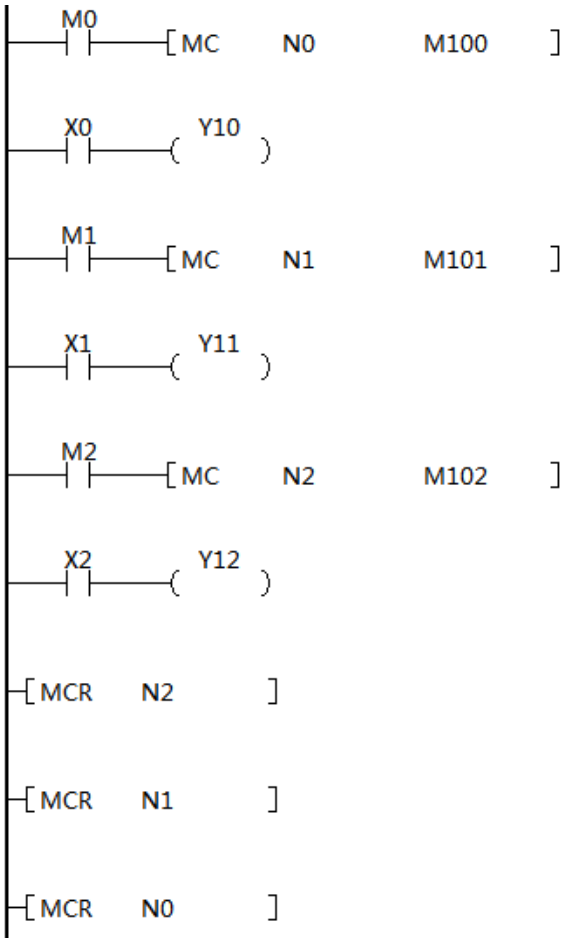
4、注意事项

（1）主控的无嵌套使用



无嵌套时 N 的编号可以任意排列，且使用次数无限制。

（2）主控的嵌套使用



有嵌套时最大允许 8 层嵌套，执行 MC 指令时，嵌套等级 N 编号依次增大，NO->N7，返回时，执行 MCR 指令，从大的嵌套等级开始解除，N7->NO。

2.1.10 INV/反转指令

1、概要

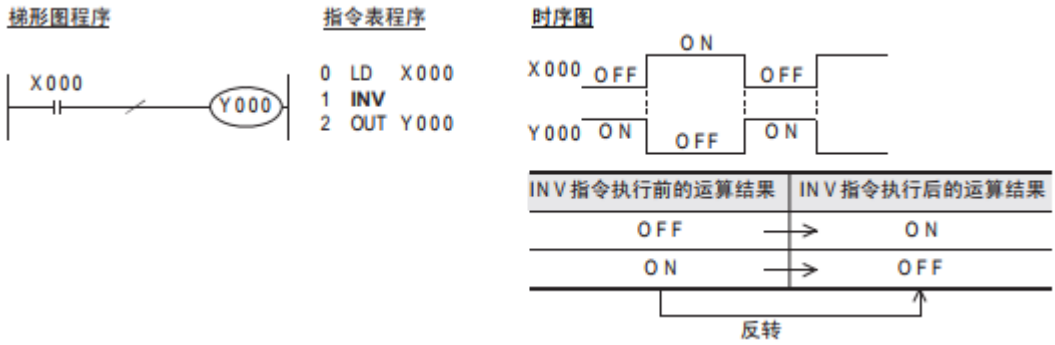
将 INV 指令执行前的运算结果反转的指令，无需指定软元件编号。

2、操作数可用元件类型

略。

3、功能和动作说明

运算结果反转。



4、注意事项

注意 INV 指令的作用范围，是将前面程序块的输出结果进行反转。

2.1.11 MEP、MEF/运算结果脉冲化指令

1、概要

MEP、MEF 指令是使运算结果脉冲化的指令，不需要指定软元件编号。

2、操作数可用元件类型

略。

3、功能和动作说明

（1）MEP 指令（运算结果的上升沿时为 ON）



（2）MEF 指令（运算结果的下升沿时为 ON）



4、注意事项

注意指令的作用范围，是根据前面程序块的运算结果而动作的。

2.1.12 PLS、PLF/脉冲输出指令

1、概要

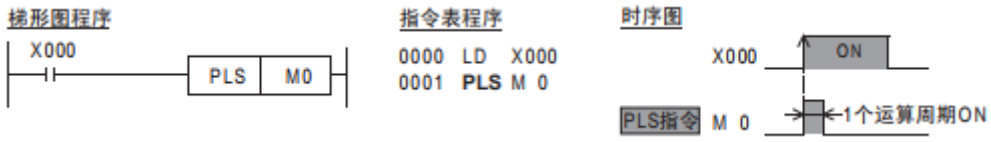
使用 PLS 指令后，仅在驱动输入 ON 以后的 1 个运算周期内，对象软元件动作。
使用 PLF 指令后，仅在驱动输入 OFF 以后的 1 个运算周期内，对象软元件动作。

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
PLS		*	*															*					
PLF		*	*															*					

3、功能和动作说明

（1）PLS 指令（上升沿的微分输出）



X000 从 OFF 变为 ON 时，只有一个运算周期的 M0 为 ON。

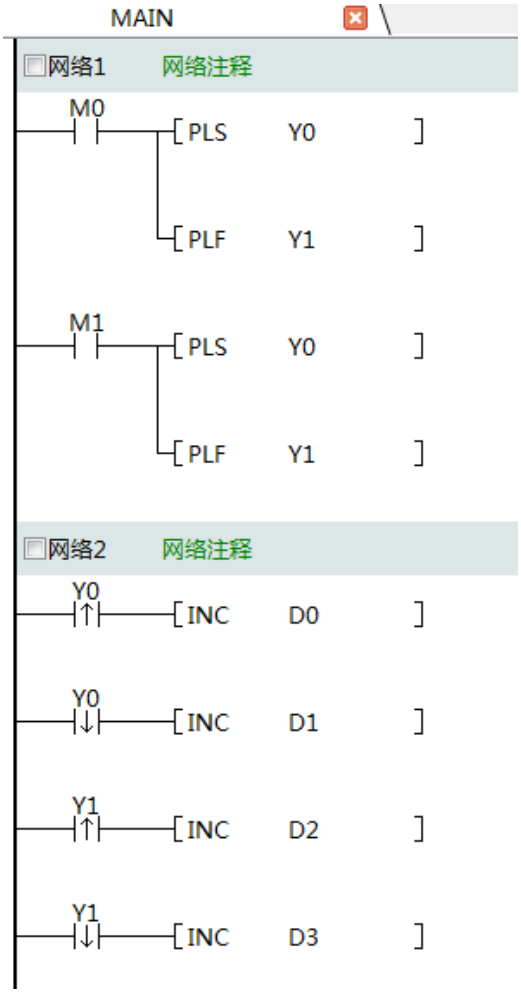
（2）PLF 指令（下升沿的微分输出）



X000 从 ON 变为 OFF 时，只有一个运算周期的 M1 为 ON。

4、注意事项

（1）同一个线圈对应多个 PLS 或 PLF 指令的执行情况，此时只会执行梯形图中最后面的 PLS 或 PLF。



Y0 对应 2 个 PLS 指令，Y1 对应 2 个 PLF 指令。

M0 置位后，Y0，Y1 无动作；M0 复位后，Y0，Y1 无动作。

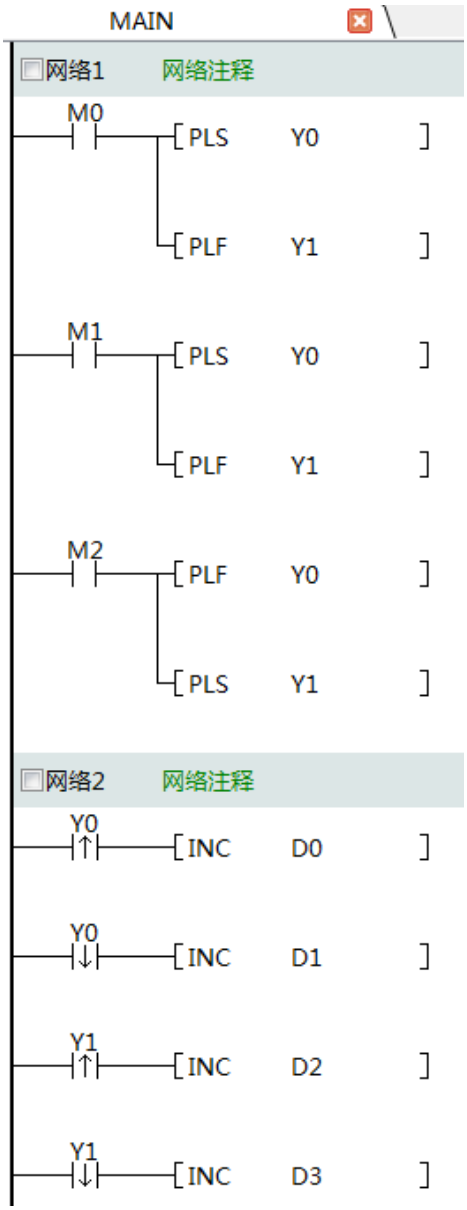
M1 置位后，会产生 Y0 上升沿和 Y0 下降沿，D0 加 1、D1 加 1。M1 复位后，会产生 Y1 的上升沿和 Y1 的下降沿，D2 加 1、D3 加 1。

详细动作分析：

M1 置位后，本次梯形图扫描周期 PLS 执行时置位 Y0，产生 Y0 上升沿,，D0 加 1；下个梯形图扫描周期 PLS 不会执行，复位 Y0，导致 Y0 这里再次产生下降沿，所以 D1 加 1。这两个动作分别是在两个梯形图扫描周期完成的；

M1 复位后，本次梯形图扫描周期 PLF 执行时置位 Y1，产生 Y1 上升沿,，D2 加 1；下个梯形图扫描周期 PLF 不会执行，复位 Y1，导致 Y1 这里再次产生下降沿，所以 D3 加 1。这两个动作分别是在两个梯形图扫描周期完成的；

（2）同一个线圈 Y 同时对应 PLS 和 PLF 指令的执行情况，此时只会执行梯形图中最后面的 PLS 或 PLF。



Y0 对应 2 个 PLS、1 个 PLF 指令，Y1 对应 2 个 PLF、1 个 PLS 指令。

M0 置位后，Y0，Y1 无动作。

M1 置位后，Y0，Y1 无动作。

M2 置位后，会产生 Y1 上升沿和 Y1 下降沿，D2 加 1、D3 加 1。M2 复位后，会产生 Y0 的上升沿和 Y0 的下降沿，D0 加 1、D1 加 1。

详细动作分析：

M2 置位后，本次梯形图扫描周期 PLS 执行时置位 Y1，产生 Y1 上升沿,，D2 加 1；下个梯形图扫描周期 PLS 不会执行，复位 Y1，导致 Y1 这里再次产生下降沿，所以 D3 加 1。这两个动作分别是在两个梯形图扫描周期完成的；

M1 复位后，本次梯形图扫描周期 PLF 执行时置位 Y0，产生 Y0 上升沿,，D0 加 1；下个梯形图扫描周期 PLF 不会执行，复位 Y0，导致 Y0 这里再次产生下降沿，所以 D1 加 1。这两个动作分别是在两个梯形图扫描周期完成的；

2.1.13 SET、RST/置位、复位指令

1、概要

（1）SET 置位位元件

（2）RST 复位位元件

（3）RST 清除字元件的当前值

2、操作数可用元件类型

指令	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
SET		*	*			*	*											*					
RST		*	*	*	*	*	*					*	*	*	*	*	*	*					

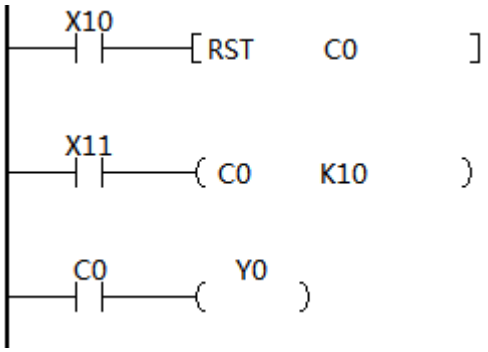
3、功能和动作说明

（1）SET 指令

对输出继电器(Y)、辅助继电器(M)、状态(S)以及数据寄存器(D)的指定位进行线圈驱动的指令。

（2）RST 指令

①复位普通定时器和累计型计数器



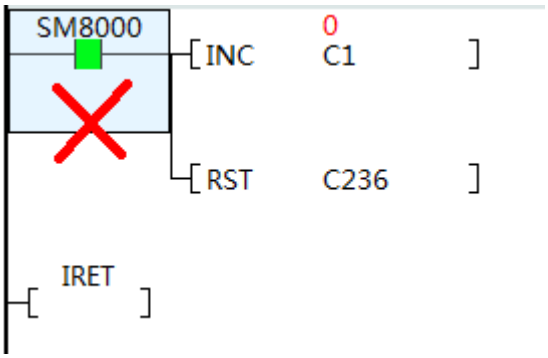
由 C0 对 X011 的 OFF→ON 的次数增计数，计数结果达到设定值 K10 的时候，输出触点 C0 动作。此后，即使 X011 从 OFF 变为 ON，计数器的当前值也不改变，输出触点也保持 动作。为了清除这些，恢复输出触点，要使 X010 为 ON。OUT C0 指令后面，需要指定常数 K 或间接设定用的数据寄存器编号。停电保持(保持)用计数器的场合，即使停电也能保持当前值以及输出触点的动作状态和复位状态。

②复位普通为元件或者字元件清 0

4、注意事项

操作特殊寄存器时，指令会执行，但结果不一定会置位或复位，比如 RST SM8000，SM8000 仍然为 ON。

用 RST 指令操作计数器 C 或者高速计数器时，RST 前面的触点不能使用常闭，否则一直使得计数器处于复位状态，而不能计数。即使是在中断服务函数里也不可以。



2.1.14 NOP/空指令

1、概要

空操作的指令。

2、操作数可用元件类型

略。

3、功能和动作说明

在一般的指令和指令之间加入 NOP 时，可编程控制器会无视其存在而继续运行。如在程序的中间加入 NOP，当需要更改、增加程序的时候，只需要很小地变动步编号就能实现，但是要求程序有余量。此外，若将已经写入的指令换成 NOP 指令，则回路会发生变化，请务必注意。

4、注意事项

在编程时，NOP 会占用程序步，但是不会影响程序运行。

2.1.15 END/程序结束指令

1、概要

表示程序结束的指令，在梯形图程序最后面，无显示。

2、操作数可用元件类型

略。

3、功能和动作说明

（1）程序结束处理

可编程控制器重复执行输入处理->执行程序->输出处理，若在程序的最后执行 END 指令，直接进行输出处理，FEND 同 END。

（2）重要参数刷新

执行 END 时会刷新看门狗时间，刷新报错寄存器。

4、注意事项

UNISYS 编程时，会隐藏 END 指令，自动添加到程序结束位置并下载至 PLC。

2.1.16 指令步数和指令软元件

基本指令的指令步数如下所示。

软元件		指令						
		LD, LDI, AND, ANI, OR, ORI	OUT	SET	RST	PLS, PLF	LDP, LDF, ANDP, ANDF, ORP, ORF	MC
位软元件	X000～X377	1	—	—	—	—	2	—
	Y000～Y377	1	1	1	1	2	2	3
	M0～M1535	1	1	1	1	2	2	3
	M1536～M3583	2	2	2	2	2	2	3
	M3584～M7679	3	3	3	3	3	3	4
	S0～S1023	1	2	2	2	—	2	—
	S1024～S4095	2	2	2	2	—	2	—
	T0～T191, T200～T245	1	3	—	2	—	2	—
	T192～T199, T246～T511	1	3	—	2	—	2	—
	C0～C199	1	3	—	2	—	2	—
	C200～C255	1	5	—	2	—	2	—
	特殊辅助继电器 M8000～M8255	1	2	2	2	—	2	—
	特殊辅助继电器 M8256～M8511	2	2	2	2	—	2	—
带变址的位软元件	X000～X357	3	—	—	—	—	—	—
	Y000～Y357	3	3	3	3	3	—	—
	M0～M7679	3	3	3	3	3	—	—
	T0～T511	3	4	—	—	—	—	—
	S0～S4095	—	—	—	—	—	—	—
	C0～C199	3	4	—	3	—	—	—
	C200～C255	—	—	—	—	—	—	—
	特殊辅助继电器 M8000～M8511	3	3	3	3	—	—	—
字软元件	D0～D7999, 特殊数据寄存器 D8000～D8511	—	—	—	3	—	—	—
	R0～R32767	—	—	—	3	—	—	—
带变址的字软元件	D0～D7999, 特殊数据寄存器 D8000～D8511, R0～R32767	—	—	—	—	—	—	—
字软元件的位指定	Dx.y 特殊辅助继电器	3	3	3	3	—	3	—

2.2 流程指令

用于控制梯形图的逻辑执行顺序（流程）。

2.2.1 CJ/程序跳转指令

普通程序跳转，跳转至对应的 P 程序位置。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC0	3	[CJ(P) D]

操作数说明

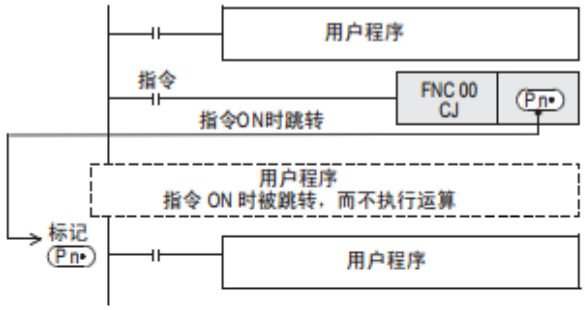
操作数	内容	类型
D	跳转目标标记的指针编号(P)	指针编号

操作数为指针标号 P0~P1023，其中 P63 为 END 所在步序，不需标记。指针标号允许用变址寄存器修改。

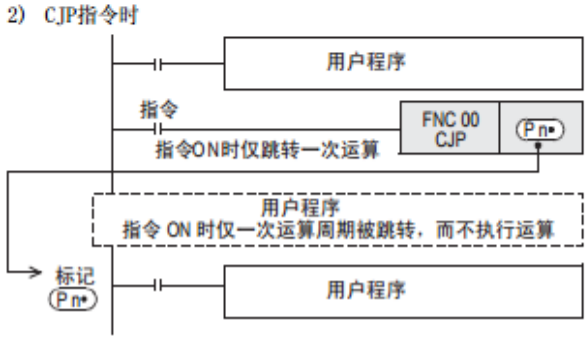
2、功能和动作说明

当指令输入为 ON 时，执行指定标记(指针编号)的程序。

1) CJ指令时



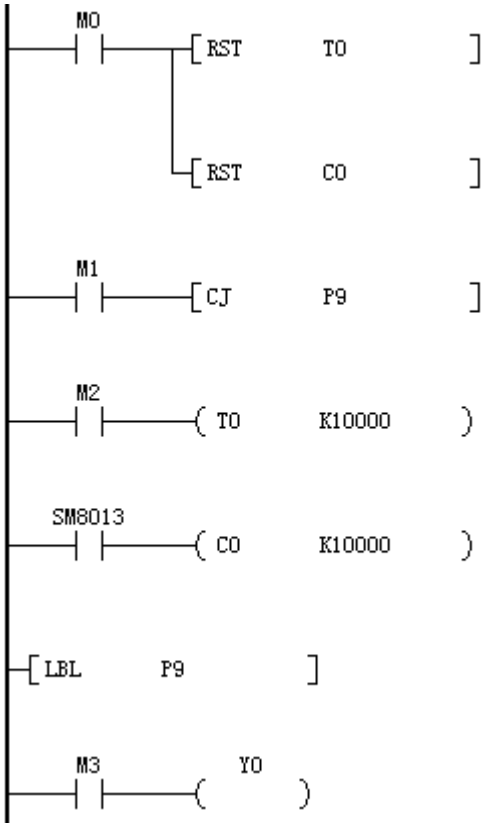
2) CJP指令时



3、程序举例

条件跳转指令 CJ 的步骤：

如图所示，当 M1 接通时，则由 CJ P9 指令跳到标号为 P9 的指令处开始执行，跳过了程序的一部分，减少了扫描周期。如果 M1 断开，跳转不会执行，则程序按原顺序执行。



4、注意事项

（1）在一个程序中一个标号只能出现一次，否则将出错；

（2）在跳转执行期间，即使被跳过程序的驱动条件改变，但其线圈（或结果）仍保持跳转前的状态，因为跳转期间根本没有执行这段程序。

（3）如果在跳转开始前定时器和计数器已在工作，则在跳转执行期间它们将停止工作，到跳转条件不满足后又继续工作。但对于正在工作的高速计数器 C235~C255 或者子程序专用定时器不管有无跳转仍连续工作。

（4）若积算定时器和计数器的复位（RST）指令在跳转区外，即使它们的线圈被跳转，但对它们的复位仍然有效。

（5）跳转指令可以根据特定的需求减少程序周期，跳过不需要运行的程序段。

2.2.2 CALL/子程序调用指令

用于子程序跳转（调用）。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC1	3	[CALL(P) D]

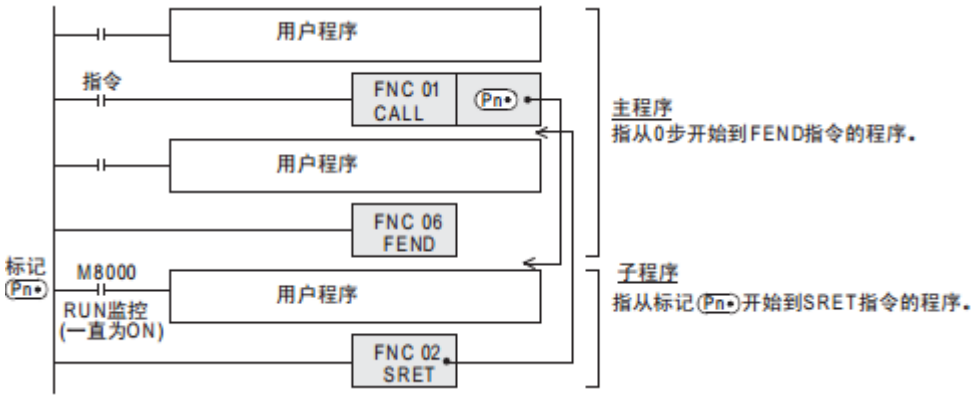
操作数说明

操作数	内容	类型
D	跳转目标标记的指针编号(P)	指针编号

操作数为指针标号 P0~P1023

2、功能和动作说明

当指令输入为 ON 时，执行 CALL 指令，向标记 Pn 的步跳转。接着，执行标记 Pn 的子程序。执行 SRET 指令后，返回到 CALL 指令的下一步。

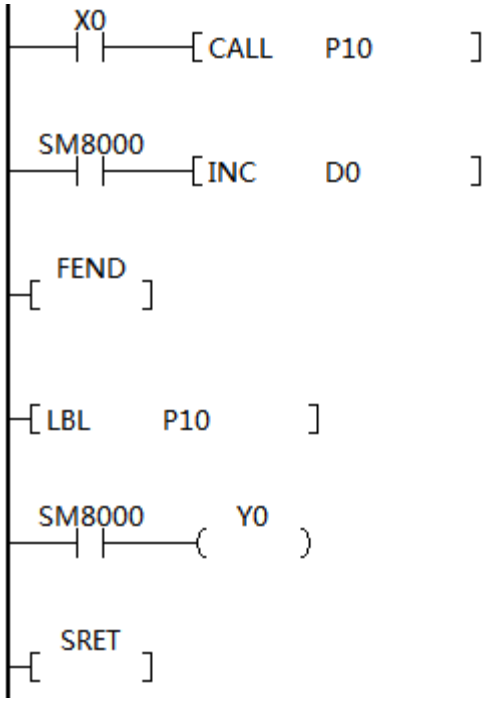


3、程序举例

（1）CALL 无嵌套的使用

主程序：X0 为 ON，则向标记 P10 的步跳转

子程序：执行子程序后，通过执行 SRET 指令，可以返回到原来的步+1 的位置。



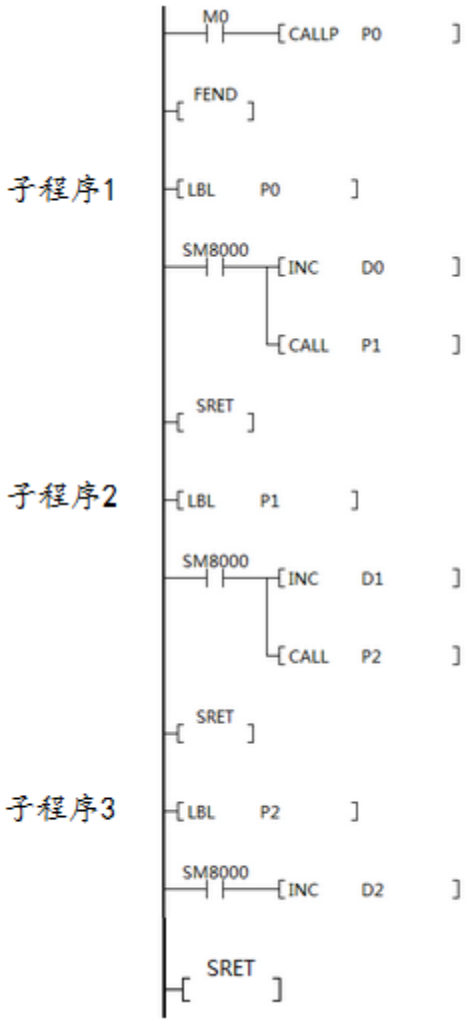
（2）CALL 多层嵌套的使用

主程序：M0 每 OFF→ON 一次，向标记 P0（子程序 1）跳转一次。

子程序 1:执行 INC 指令后 D0 加 1，然后向标记 P1（子程序 2）的步跳转。

子程序 2:执行 INC 指令后 D1 加 1，然后向标记 P2（子程序 3）的步跳转。

子程序 3:执行 INC 指令后 D2 加 1，然后 SRET 返回到子程序 2，然后再返回到子程序 1，然后再返回到 CALLP P0 下方。



4、注意事项

（1）在主程序的最后用 FEND 指令编程，没有 FEND 指令运行报 6637 停机错误。

（2）CALL 指令用的标记(P)，在 FEND 指令后编程，否则运行报 6637 停机错误。

（3）CALL 嵌套最多允许 6 层，超过报 6634 停机错误。

2.2.3 SRET/子程序返回指令

使用 CALL 指令调用的子程序，子程序结束后使用 SRET 返回。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC2	1	SRET

2、功能和动作说明

主要用于 CALL 指令调用后返回，用法参照 CALL 指令中例程。

2.2.4 IRET/中断返回指令

中断程序结束后用 IRET 返回到主程序调用位置的下一步位置。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC3	1	IRET

2、功能和动作说明

在处理主程序过程中如果产生中断(输入、定时器、计数器)，则跳转到中断(I)程序，然后使用 IRET 指令返回主程序。

用法说明参照 3.2 章节。

2.2.5 EI/允许中断指令

中断允许控制用，在需要使用中断的位置编写此指令

1、指令格式

指令编号	指令所占程序步	指令格式
FNC4	1	-[EI]

2、功能和动作说明

可编程控制器通常为禁止中断状态。使用 EI（FNC4）指令，可以使可编程控制器变为允许中断的状态。使用输入中断和定时器中断，计数器中断功能的时候，请使用该指令。中断子程序的用法说明参照 3.2 章节。

2.2.6 DI/禁止中断指令

中断禁止控制用，在不需要中断响应的位置编写此指令，使得中断在 EI 允许的位置处执行。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC5	1	-[DI]

2、功能和动作说明

在改为允许中断后，使用 DI 指令可以再次更改为禁止中断，可用此指令设定中断禁止的区域。用法说明参照 3.2 章节。

2.2.7 FEND/主程序结束指令指令

主程序结束专用指令。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC6	1	FEND

2、功能和动作说明

FEND 指令是主程序结束命令，功能同 END 指令，只是作为主程序专用的结束指令，在编写子程序和中断程序时需要使用 FEND 指令。

2.2.8 WDT/看门狗定时器指令

通过此指令刷新看门狗时间。

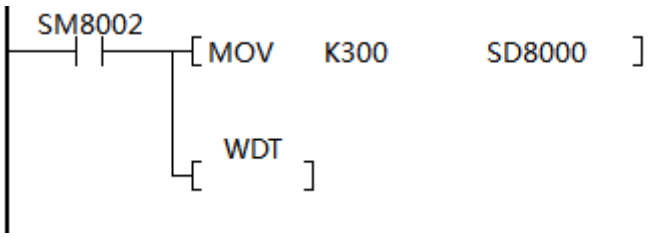
1、指令格式

指令编号	指令所占程序步	指令格式
FNC7	1	WDT(P)

2、功能和动作说明

可编程控制器的运算周期(0～END 或 FEND 指令的执行时间)如要超出 200ms 时，可编程控制器会出现看门狗定时器错误(检测出运算异常)，然后 ERROR(ERR) LED 灯亮后停止。类似这样的运算周期较长的情况，在程序中间插入 WDT 指令，可以避免出现这样的错误。

3、程序举例



设置看门狗定时器时间 300ms；

看门狗定时器刷新：没有编写 WDT 指令的时候，END 处理时 SD8000 的值变为有效。

4、注意事项

避免看门狗定时器超时的方法：

1. 控制梯形图运行周期。
2. 梯形图中 FOR、NEXT 这类流程指令使用次数不要太多。
3. 当梯形图中有运行起来比较耗时的程序时，提前刷新看门狗定时器或者更改看门狗定时器的时间。

2.2.9 FOR/循环范围开始指令

用于程序的重复执行操作。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC8	3	-[FOR S]

操作数说明

操作数	内容	类型
S	FOR~NEXT 指令之间的重复次数(1~32767,-32768~0 时，作为 1 处理)	BIN16 位

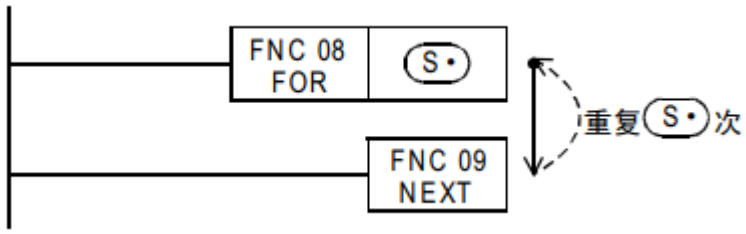
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D								*	*	*	*	*	*	*	*	*	*	*	*	*			

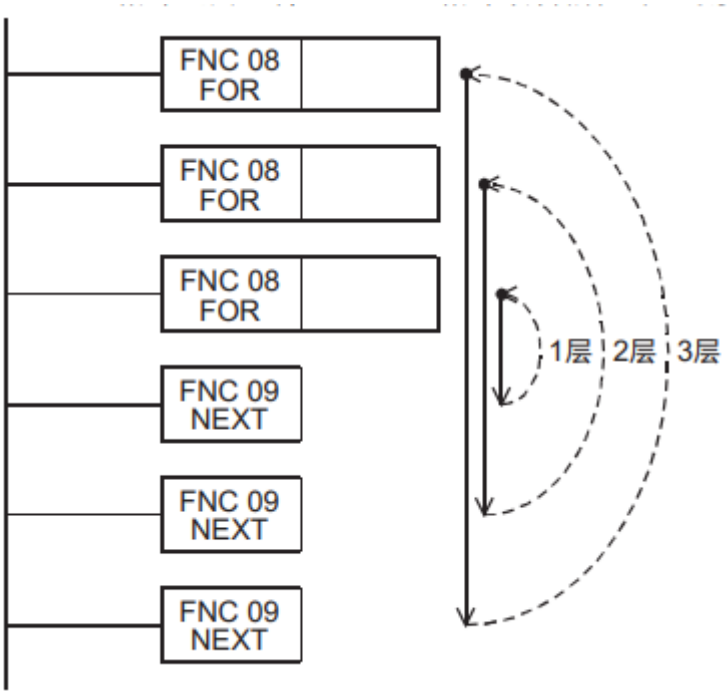
2、功能和动作说明

从 FOR(FNC 08)指令开始到 NEXT 指令之间的程序按指定次数重复运行。

（1）单层 FOR-NEXT



（2）多层 FOR-NEXT 嵌套



3、程序举例

参考上面的功能动作说明。

4、注意事项

（1）FOR-NEXT 嵌套使用最多允许 6 层，超出报 6636 停机错误。

（2）使用多层嵌套时注意看门狗超时时间，可能会引发超时，必要时要刷新看门狗时间。

（3）FOR-NEXT 对应个数不匹配会报 6637 停机错误。

（4）FOR 后面的次数是指一个梯形图扫描周期该循环执行的次数。

2.2.10 NEXT/循环范围结束指令

用于 FOR 循环的程序段结束。

1、指令格式

指令编号	指令所占程序步	指令格式
FNC9	1	-[NEXT]

2、功能和动作说明

和 FOR 指令结合使用，作为循环的结束，具体使用方法参考 FOR 指令。

2.3 传送比较指令

用于基本的数据传送、比较等操作指令。

2.3.1 CMP/比较指令

比较 2 个值，根据结果置位输出位元件。

1、指令格式

FNC10	指令所占程序步	指令分类	指令格式
16 位	7	CMP	-[(D)CMP(P) S1 S2 D]
		CMPP	
32 位	13	DCMP	
		DCMPP	

操作数说明

操作数	内容	类型
S1	比较值	BIN16/32 位
S2	比较源	BIN16/32 位
D	输出比较结果的起始位元件	位

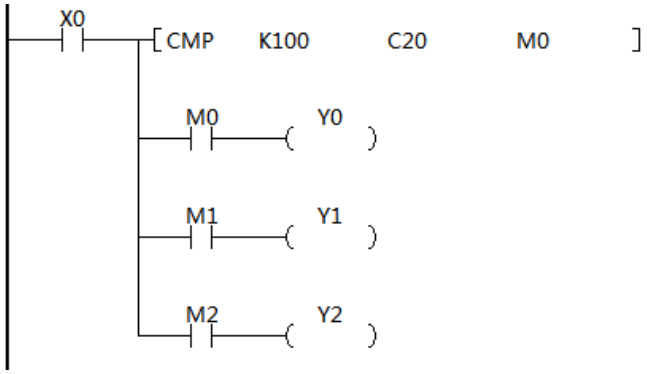
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*	*											*					

2、功能和动作说明

对比较值 S1 和比较源 S2 的内容进行比较，根据其结果(小、一致、大)，使 D、D+1、D+2 中的一个为 ON。

3、程序举例



C20 当前值 < 100 时，M0 置 ON，Y0 点亮；

C20 当前值 = 100 时，M1 置 ON，Y1 点亮；

C20 当前值 > 100 时，M2 置 ON，Y2 点亮。

4、注意事项

（1）D 中指定的软元件个数是连续的三个，注意不要与其他正在使用的软元件重复。

（2）即使是指令输入为 OFF,CMP 指令不执行时,D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

2.3.2 ZCP/区间比较指令

针对 2 个值构成的区间，将比较源的值与此区间比较，根据结果(区间左侧、区间内，区间右侧)置位输出位软元件。

1、指令格式

FNC11	指令所占程序步	指令分类	指令格式
16 位	9	ZCP	[(D)ZCP(P) S1 S2 S D]
		ZCPP	
32 位	17	DZCP	
		DZCPP	

操作数说明

操作数	内容	类型
S1	左区间的比较值	BIN16/32 位
S2	右区间的比较值	BIN16/32 位
S	比较源	BIN16/32 位
D	输出比较结果的起始位软元件	位

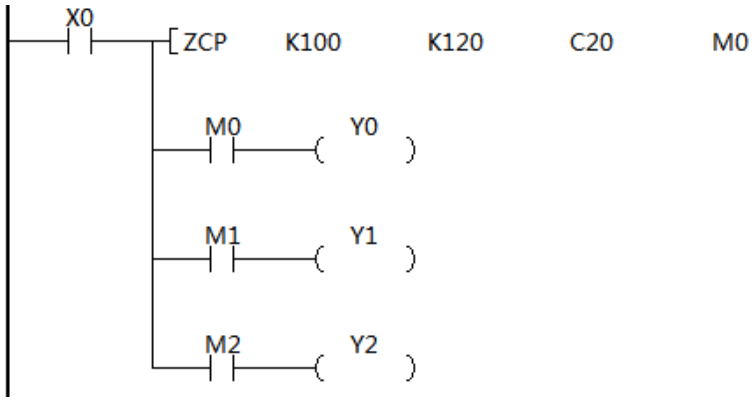
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*	*											*					

2、功能和动作说明

将比较源 S 的内容与左区间比较值 S1 和右区间比较值 S2 进行比较，根据其结果(区间左侧、区间内，区间右侧)，使 D、D+1、D+2 其中一个为 ON。

3、程序举例



C20 当前值 < 100 时，M0 置 ON，Y0 点亮；

100 ≤ C20 当前 ≤ 120 时，M1 置 ON，Y1 点亮；

C20 当前值 > 120 时，M2 置 ON，Y2 点亮。

4、注意事项

（1）D 中指定的软元件个数是连续的三个，注意不要与其他正在使用的软元件重复。

（2）即使是指令输入为 OFF,CMP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

2.3.3 MOV/传送指令

将源软元件的内容传送(复制)到目的软元件中的指令。

1、指令格式

FNC12	指令所占程序步	指令分类	指令格式
16 位	5	MOV	[(D)MOV(P) S D]
		MOVP	
32 位	9	DMOV	
		DMOVP	

操作数说明

操作数	内容	类型
S	传送源	BIN16/32 位

D	传送目标	BIN16/32 位

操作数可用元件类型

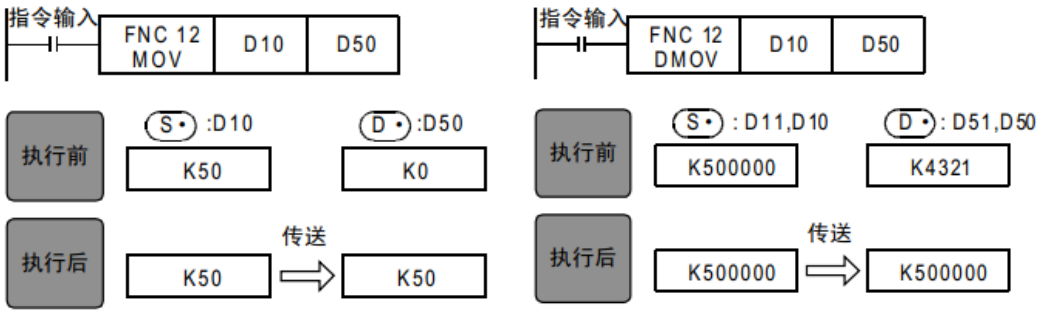
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

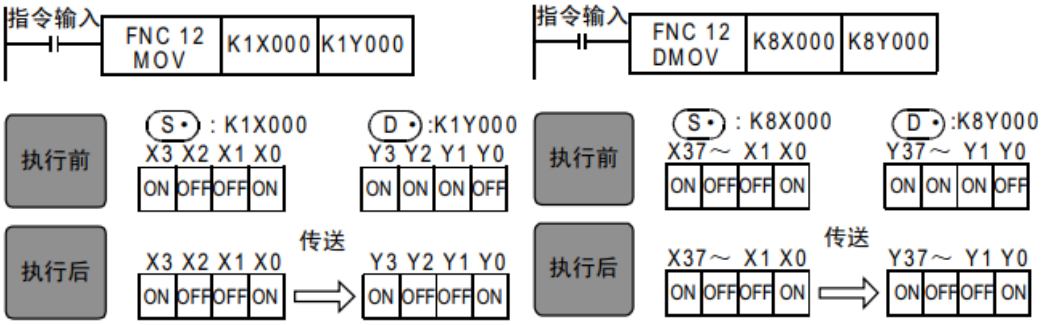
将传送源 S 的 1 点数据内容传送给目标 D。

3、程序举例

（1）传字软元件



（2）传组合位元件



4、注意事项

如果传送的是组合位元件，16 位指令最多传送 16 个（4 的倍数）位软元件，32 位指令最多传送 32 个（4 的倍数）位软元件。

2.3.4 SMOV/位移动指令

以位数为单位(4 位)进行数据分配合成的指令。

1、指令格式

FNC13	指令所占程序步	指令分类	指令格式
16 位	11	SMOV SMOVP	SMOV(P) S m1 m2 D n

操作数说明

操作数	内容	类型
S	进行位移动的数据	BIN16 位
m1	移动的起始位的位置	BIN16 位
m2	移动的位的个数	BIN16 位
D	位移动后的目标数据	BIN16 位
n	移动目标的起始位的位置	BIN16 位

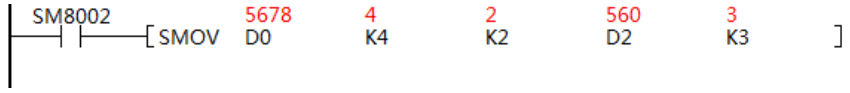
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
m1																			*	*			
M2																			*	*			
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

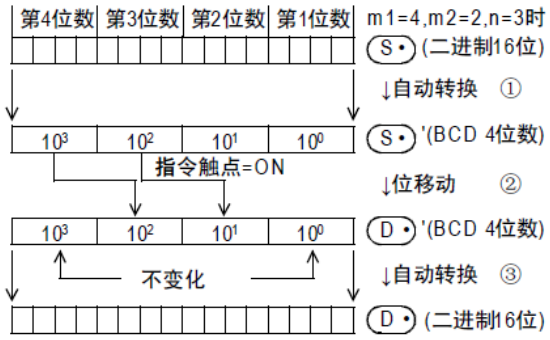
2、功能和动作说明

传送源 S 和传送目标 D 的内容转换(0000～9999)成 4 位数的 BCD，m1 位数起的低 m2 位数部分被传送(合成)到传送目标 D 的 n 位数起始处，然后转换成 BIN，保存在传送目标 D 中。

3、程序举例



D0	0	0	0	1	0	1	1	0	0	0	1	0	1	1	1	0	5678
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	1	0	0	0	1	1	0	0	0	0	560



① (S·)从BIN转换为BCD。

②从第 m1 位数起的低 m2 位数部分的数据，被传送（合成）到(D·)'的第 n 位数起始 m2位数。
(D·)'的10^3位数，被10^0位数在执行来自(S·)'的传送时不受任何影响。

③合成的数据(BCD)转换成BIN后，保存到(D·)中。

4、注意事项

（1）原数据取值范围是 0~9999。

（2）m1，m2，n 的范围是 1~4。

（3）将 M8168 置 ON 后，执行 SMOV 指令时，则不能进行 BIN→BCD 转换，位移动以 4 位为单位执行。

2.3.5 CML/反转指令

将源数据的所有位取反然后进行传送的指令。

1、指令格式

FNC14	指令所占程序步	指令分类	指令格式
16 位	5	CML	[(D)CML(P) S D]
		CMLP	
32 位	9	DCML	
		DCMLP	

操作数说明

操作数	内容	类型

S	执行反转的原数据	BIN16/32 位
D	反转后目标数据	BIN16/32 位

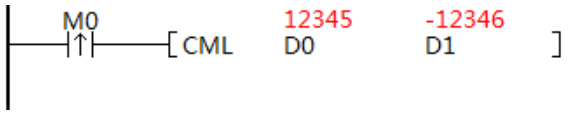
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 中指定的软元件的各位反转(0→1,1→0)后，传送至 D。

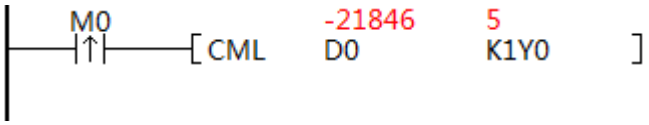
3、程序举例



D0	0	0	1	1	0	0	0	0	0	1	1	1	0	0	1	12345
D1	1	1	0	0	1	1	1	1	1	1	0	0	0	1	1	0 -12346

4、注意事项

指定的是位组合元件时，根据实际的位数来反转传送，但是这个位数不能超过指令位数且为 4 的倍数，下面的例子是指定位数为 4 的情况。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	-21846

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
Y0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5

2.3.6 BMOV/成批传送指令

对指定点数的多个数据进行成批传送(复制)。

1、指令格式

FNC15	指令所占程序步	指令分类	指令格式
16 位	7	BMOV BMOVP	-[BMOV(P) S D n]

操作数说明

操作数	内容	类型
S	传送源	BIN16
D	传送目标	BIN16
n	传送点数[1≤n≤512]	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*					*	*			

2、功能和动作说明

将 S 开始的 n 点的数据成批传送到 D 开始的 n 点中。超出软元件编号范围时，在可能的范围内传送。

3、程序举例

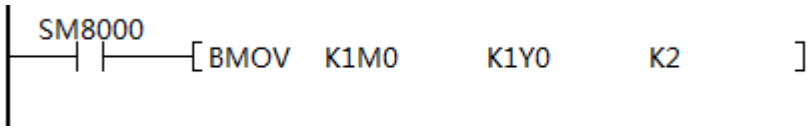


4、注意事项

（1）无论传送源的数据有无传送，为了防止数据源没有传送就被改写，采用编号重叠的方法，按上面的①～③的顺序自动传送。

（2）双向传送功能，当 SM8024 标志位置 ON 时，传送方向变为 D->S。

（3）带有位数指定的位软元件的情况下，S 和 D 要采用相同的 K 系数



2.3.7 FMOV/多点传送指令

将同一数据传送到多点软元件中的指令。

1、指令格式

FNC16	指令所占程序步	指令分类	指令格式
16 位	7	FMOV	(D)FMOV(P) S D n
		FMOVP	
32 位	13	DFMOV	
		DFMOVP	

操作数说明

操作数	内容	类型
S	传送源	BIN16/32 位
D	传送目标的起始字软元件	BIN16/32 位
n	传送点数[1≦n≦512]	BIN16

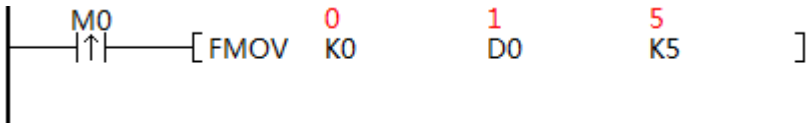
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*			*					
n																			*	*			

2、功能和动作说明

将 S 的内容传送到以 D 起始的 n 点的软元件中，n 点的软元件内容都相同，以 n 指定的个数超出了软元件编号范围时，在可能的范围内传送。

3、程序举例



执行前

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3
D3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4
D4	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5

执行后

D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4、注意事项

n 指定的个数超出了软元件编号范围时，在可能的范围内传送。

2.3.8 XCH/交换指令

在 2 个软元件之间进行数据交换

1、指令格式

FNC17	指令所占程序步	指令分类	指令格式
16 位	5	XCH	[(D)XCH(P) D1 D2]
		XCHP	
32 位	9	DXCH	
		DXCHP	

操作数说明

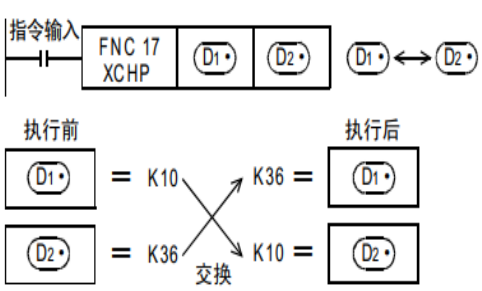
操作数	内容	类型
D1	要交换数据	BIN16/32 位
D2	要交换数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D1									*	*	*	*	*	*	*	*	*	*					
D2									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

D1 和 D2 相互之间进行数据交换。



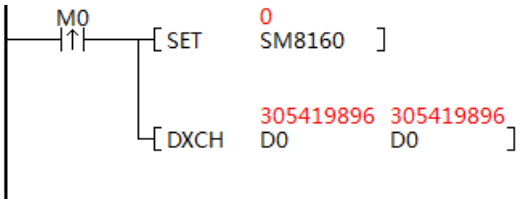
3、程序举例

无

4、注意事项

（1）在 SM8160 为 ON 的状态下执行指令时，交换字软元件的高 8 位(字节)和低 8 位(字节)。该动作和 SWAP(FNC 147)指令为相同的动作。

（2）32 位运算时，交换各个字软元件的高 8 位(字节)和低 8 位(字节)。



执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	1	0	1	0	1	1	0	0	1	1	1	1	0	0	0	5678
D1	0	0	0	1	0	0	1	0	0	0	1	1	0	1	0	0	1234

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	1	1	1	1	0	0	0	0	1	0	1	0	1	1	0	7856
D1	0	0	1	1	0	1	0	0	0	0	0	1	0	0	1	0	3412

（3）SM8160 为 ON 时，D1 和 D2 的软元件编号不一致的时候会报 6706 运算错误。

2.3.9 BCD/BCD 转换指令

将 BIN(2 进制数)转换成 BCD(10 进制数)的指令。

1、指令格式

FNC18	指令所占程序步	指令分类	指令格式
16 位	5	BCD	[(D)BCD(P) S D]
		BCDP	
32 位	9	DBCD	
		DBCDP	

操作数说明

操作数	内容	类型
S	源(2 进制数)数据	BIN16/32 位
D	目标(10 进制数)数据	BIN16/32 位

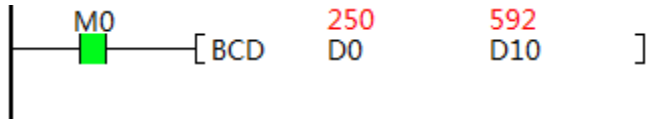
操作数可用元件类型

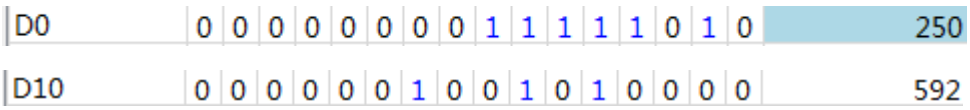
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 的 BIN(2 进制数)数据转换成 BCD(10 进制数)数据后传送到 D 中。16 位指令时 S 的数据可以转换成 K0~K9999 的 BCD(10 进制数)，32 位指令时 S 的数据可以转换成 K0~K99999999 的 BCD(10 进制数)。

3、程序举例





D0 写入数据 250，250 转换为 BCD 码为 0010（2） 0101（5） 0000（0）。

4、注意事项

注意源操作数的取值范围。

2.3.10 BIN/BIN 转换指令

将 10 进制数(BCD)转换成 2 进制数(BIN)的指令。

1、指令格式

FNC19	指令所占程序步	指令分类	指令格式
16 位	5	BIN	[(D)BIN(P) S D]
		BINP	
32 位	9	DBIN	
		DBINP	

操作数说明

操作数	内容	类型
S	源数据(10 进制数)	BIN16/32 位
D	目标数据(2 进制数)	BIN16/32 位

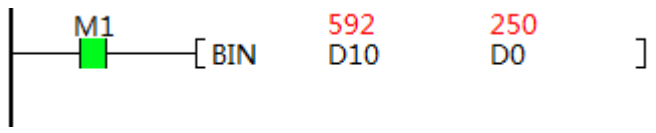
操作数可用元件类型

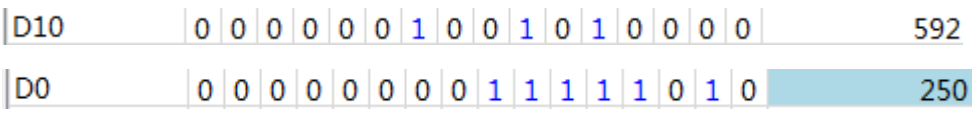
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S 的 BCD(10 进制数)数据转换成 BIN(2 进制数)数据后传送到 D 中。16 位指令时 S 可以在 0～9999(BCD)的范围内转换，32 位指令时 S 可以在 0～99999999(BCD)的范围内转换。

3、程序举例





D10 写入数据 592，592 对应的是 250 的 BCD 数据，250 转换为 2 进制数为 1111 1010。

4、注意事项

注意源操作数的取值范围。

2.4 四则逻辑运算指令

用于基本的加减乘除、逻辑运算。

2.4.1 ADD/加法运算指令

将两个数据进行加法运算的指令。

1、指令格式

FNC20	指令所占程序步	指令分类	指令格式
16 位	7	ADD	-[(D)ADD(P) S1 S2 D]
		ADDP	
32 位	13	DADD	
		DADDP	

操作数说明

操作数	内容	类型
S1	进行加法运算的数据 1	BIN16/32 位
S2	进行加法运算的数据 2	BIN16/32 位
D	加法运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制加法运算后传送到 D 中。

2、程序举例

略。

4、注意事项

标志位的动作及数值的正负的关系：

软元件	名称	内容
SM8020	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8021	借位	ON：运算结果小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时，借位标志位动作。 OFF：运算结果不小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时
SM8022	进位	ON：运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF：运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时

2.4.2 SUB/减法运算指令

将两个数据进行减法运算的指令。

4. 指令格式

FNC21	指令所占程序步	指令分类	指令格式
16 位	7	SUB	-[(D)SUB(P) S1 S2 D]
		SUBP	
32 位	13	DSUB	
		DSUBP	

操作数说明

操作数	内容	类型
S1	进行减法运算的数据 1	BIN16/32 位
S2	进行减法运算的数据 2	BIN16/32 位
D	减法运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制减法运算后传送到 D 中。

3、程序举例

略。

4、注意事项

标志位的动作及数值的正负的关系：

软元件	名称	内容

SM8020	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8021	借位	ON：运算结果小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时，借位标志位动作。 OFF：运算结果不小于-32,768(16 位运算)或是-2,147,483,648(32 位运算)时
SM8022	进位	ON：运算结果大于 32,767(16 位运算) 或者 2,147,483,647(32 位运算) 时，进位标志位动作。 OFF：运算结果不大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时

2.4.3 MUL/乘法运算指令

将两个数据进行乘法运算的指令。

1、指令格式

FNC22	指令所占程序步	指令分类	指令格式
16 位	7	MUL	[(D)MUL(P) S1 S2 D]
		MULP	
32 位	13	DMUL	
		DMULP	

操作数说明

操作数	内容	类型
S1	进行乘法运算的数据 1	BIN16/32 位
S2	进行乘法运算的数据 2	BIN16/32 位
D	乘法运算结果	BIN32/64 位

操作数可用元件类型

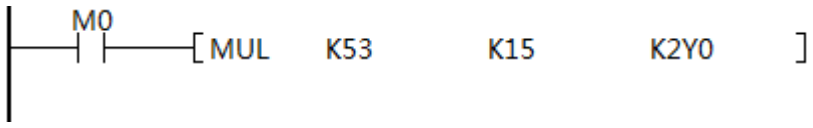
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

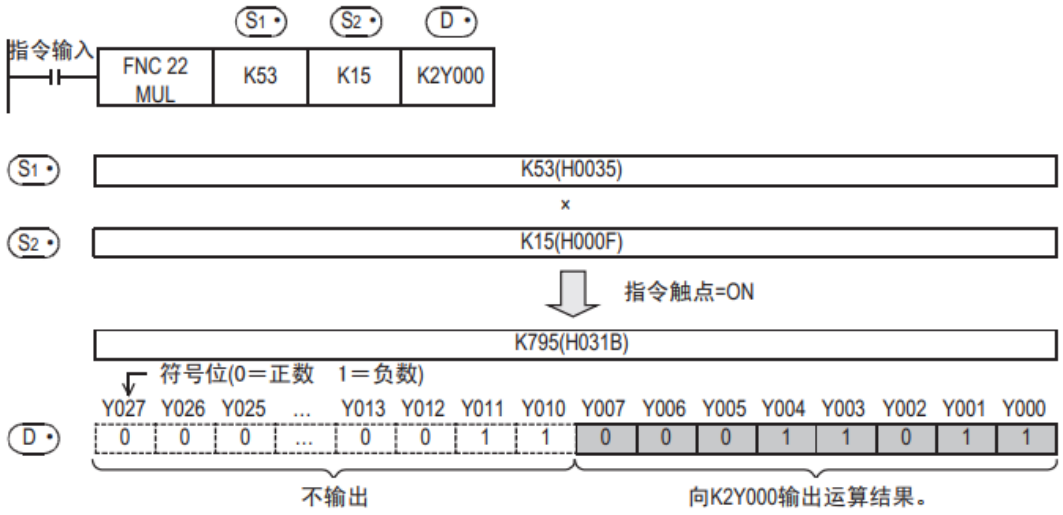
2、功能和动作说明

16 位指令时，将 S1 和 S2（S1、S2 均为单字）的内容进行二进制乘法运算后传送到 D 开始的 32 位数据中。

32 位指令时，将 S1 和 S2（S1、S2 均为双字）的内容进行二进制乘法运算后传送到的 64 位(四个连续的字软元件)中。

3、程序举例





（D+1，D）指定位数(K1～8)时，例如，指定 K2 时，只能得到乘积(32 位)中的低 8 位。

4、注意事项

（1）使用组合位时，得到的运算结果为 K 系数乘以 4 个位元件组合而成的数据。

（2）当运算结果为 0 时，SM8304 置 ON，其他结果为 OFF。

（3）进行 32 位指令运算时，目的操作数不能为 Z 寄存器。

（4）32 位指令计算出来得到的 64 位数据上位机并不能直接监控，只能通过多个寄存器自己组合出结果，要想更直观的显示，建议使用浮点数运算。

（5）注意 16 位指令和 32 位指令，操作数代表的实际寄存器范围

2.4.4 DIV/除法运算指令

将两个数据进行除法运算的指令。

5. 指令格式

FNC23	指令所占程序步	指令分类	指令格式
16 位	7	DIV	[(D)DIV(P) S1 S2 D]
		DIVP	
32 位	13	DDIV	
		DDIVP	

操作数说明

操作数	内容	类型
S1	被除数	BIN16/32 位
S2	除数	BIN16/32 位
D	除法运算结果	BIN32/64 位

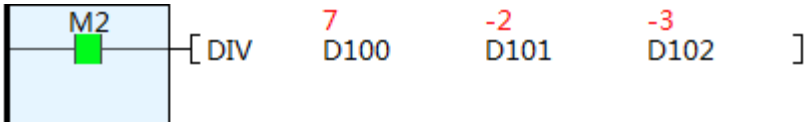
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 S1 和 S2 的内容进行二进制除法运算后，商传送到 D 中，余数传送到 D+1 中。16 位指令时，D 占用 2 个字元件，32 位指令时，D 占用 4 个字元件。

3、程序举例



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D100	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	7
D101	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-2
D102	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	-3
D103	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

4、注意事项

（1）有关运算结果：商和余数的最高位显示为正(0)、负(1)的符号。当被除数或除数其一为负时，商为负。当被除数为负时，余数为负。

（2）通过指定位数来指定位软元件时，不能得出余数。使用 32 位运算(DDIV、DDIVP)时，不能指定使用 Z 寄存器。

（3）除数 S2 为 0 时，发生运算错误，运算结果超出指令位数对应的值时，也会报错，且进位标志置 ON。

软元件	名称	内容
SM8304	零位	ON：运算结果为 0 时 OFF：运算结果为 0 以外时
SM8306	进位	ON ：运算结果超过 32,767(16 位运算)或者 2,147,483,647(32 位运算)时，进位标志位动作。 OFF ：运算结果为 32,767(16 位运算)或者 2,147,483,647(32 位运算)以下时

2.4.5 INC/加一指令

数据自增一指令。

1、指令格式

FNC24	指令所占程序步	指令分类	指令格式
16 位	3	INC	-[(D)INC(P) D]
		INCP	
32 位	5	DINC	
		DINCP	

操作数说明

操作数	内容	类型
D	进行自增一的数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 D 的内容加一后传送到 D 中。

3、程序举例

略。

4、注意事项

（1）16 位运算：+32,767 上加 1 后，变为-32,768，但是标志位(零、借位、进位)不动作。

（2）32 位运算：+2,147,483,647 上加 1 后，变为-2,147,483,648，但是标志位(零、借位、进位)不动作。

2.4.6 DEC/减一指令

数据自减一指令。

1、指令格式

FNC25	指令所占程序步	指令分类	指令格式
16 位	3	DEC	[(D)DEC(P) D]
		DECP	
32 位	5	DDEC	
		DDECP	

操作数说明

操作数	内容	类型
D	进行自减一的数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

D 的内容减一运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

（1）16 位运算：-32,768 上减一后，变为+32,767，但是标志位(零、借位、进位)不动作。

（2）32 位运算：-2,147,483,648 上减一后，变为+2,147,483,647，但是标志位(零、借位、进位)不动作。

2.4.7 WAND/逻辑与指令

两个数据进行逻辑与运算的指令。

1、指令格式

FNC26	指令所占程序步	指令分类	指令格式
16 位	7	WAND WANDP	-[WAND(P) S1 S2 D]
32 位	13	DAND DANDP	-[DAND(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	进行逻辑与运算的数据 1	BIN16/32 位
S2	进行逻辑与运算的数据 2	BIN16/32 位
D	逻辑与结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑与运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑与运算以位位单位，如下表中(1∧1=1　0∧1=0　1∧0=0　0∧0=0)所示变化。

	S1	S2	D
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	1	1	1

2.4.8 WOR/逻辑或指令

两个数据进行逻辑或运算的指令。

1、指令格式

FNC27	指令所占程序步	指令分类	指令格式
16 位	7	WOR	-[WOR(P)　S1　S2　D　]
		WORP	
32 位	13	DOR	-[DOR(P)　S1　S2　D　]
		DORP	

操作数说明

操作数	内容	类型
S1	进行逻辑或运算的数据 1	BIN16/32 位
S2	进行逻辑或运算的数据 2	BIN16/32 位
D	逻辑或结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑或运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑或运算以位为单位，如下表中(1∨1=1　0∨1=1　0∨0=0　1∨0=1)所示变化。

	S1	S2	D
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

2.4.9 WXOR/逻辑异或指令

两个数据进行逻辑异或运算的指令。

1、指令格式

FNC28	指令所占程序步	指令分类	指令格式
16 位	7	WXOR WXORP	-[WXOR(P) S1 S2 D]
32 位	13	DXOR DXORP	-[DXOR(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	进行逻辑异或运算的数据 1	BIN16/32 位
S2	进行逻辑异或运算的数据 2	BIN16/32 位
D	逻辑异或结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

S1 和 S2 的内容以各位为单位进行逻辑异或运算后，传送到 D 中。

3、程序举例

略。

4、注意事项

逻辑异或运算以位为单位，如下表中(1∀1=0　0∀0=0　1∀0=1　0∀1=1)所示变化。

	S1	S2	D
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

2.4.10 NEG/补码指令

求出数值的 2 进制补码(各位反转+1 后的值)的指令。使用这个指令后，可以反转数值的符号。

1、指令格式

FNC29	指令所占程序步	指令分类	指令格式
16 位	3	NEG	[(D)NEG(P) D]
		NEGP	
32 位	5	DNEG	
		DNEGP	

操作数说明

操作数	内容	类型
D	求补码的数据或运算结果	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

将 D 内容中的各位反转(0→1、1→0)后加 1 的结果保存到原先的软元件中。

3、程序举例

略。

4、注意事项

使用连续执行型(NEG、DNEG)指令时，每个扫描周期(各运算周期)都执行，请务必注意。

2.5 循环移位指令

使位数据或字数据按指定的方向或位数进行移动。

2.5.1 ROR/循环右移指令

使指定位数部分的位信息右移、循环的指令，不带进位标志。

1、指令格式

FNC30	指令所占程序步	指令分类	指令格式
16 位	5	ROR	-[(D)ROR(P) D n]
		RORP	
32 位	9	DROR	
		DRORP	

操作数说明

操作数	内容	类型
D	进行循环右移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

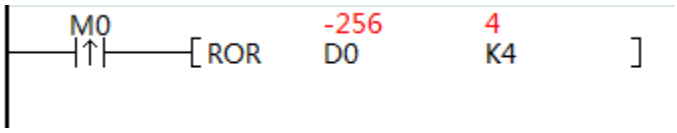
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

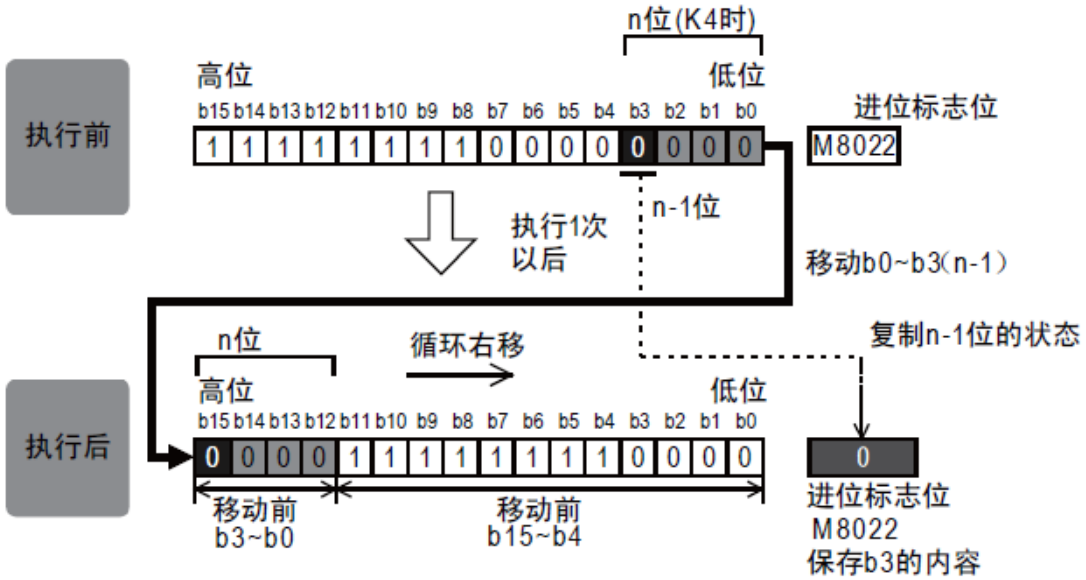
2、功能和动作说明

D 的所有位数中的 n 位循环右移。最后的位保存在进位标志位(M8022)中。

3、程序举例

将 D0 = -256 = 0xFF00 进行 4 位循环右移





4、注意事项

（1）相关软元件：M8022，进位标志，最后从最低位移出的位为 1 时置 ON。

（2)连续执行型(ROR、DROR)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3）在 D 中指定组合位元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.2 ROL/循环左移指令

使指定位数部分的位信息左移、循环的指令，不带进位标志。

1、指令格式

FNC31	指令所占程序步	指令分类	指令格式
16 位	5	ROL	[(D)ROL(P) D n]
		ROLP	
32 位	9	DROL	
		DROLP	

操作数说明

操作数	内容	类型
D	进行循环左移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

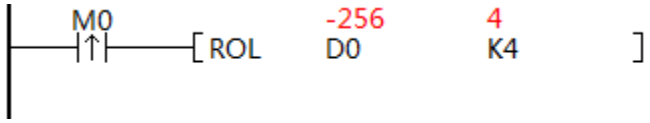
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

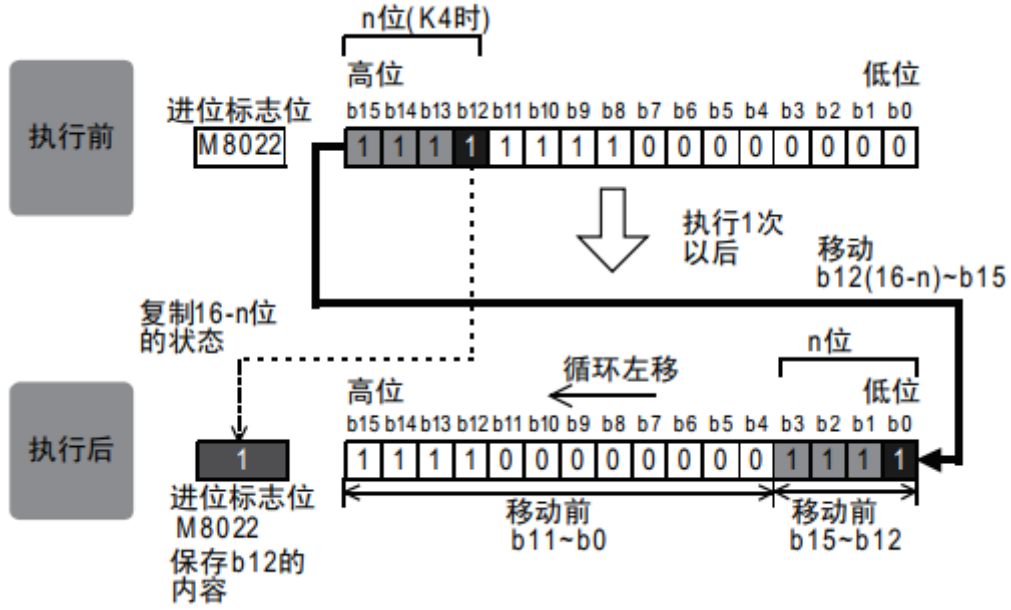
2、功能和动作说明

D 的所有位数中的 n 位循环左移。最后的位保存在进位标志位(SM8022)中。

3、程序举例

将 D0 = -256 = 0xFF00 进行 4 位循环左移





4、注意事项

（1）相关软元件：M8022，进位标志，最后从最高位移出的位为 1 时置 ON。

（2)连续执行型(ROL、DROL)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3)在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.3 RCR/带进位循环右移指令

使指定位数部分的位信息右移、循环的指令，带进位标志。

1、指令格式

FNC32	指令所占程序步	指令分类	指令格式
16 位	5	RCR	[(D)RCR(P) D n]
		RCRP	
32 位	9	DRCR	
		DRCRP	

操作数说明

操作数	内容	类型
D	进行循环右移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

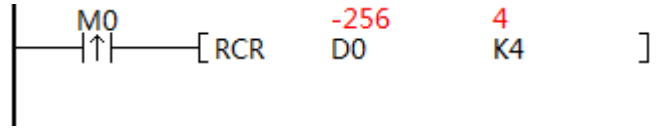
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

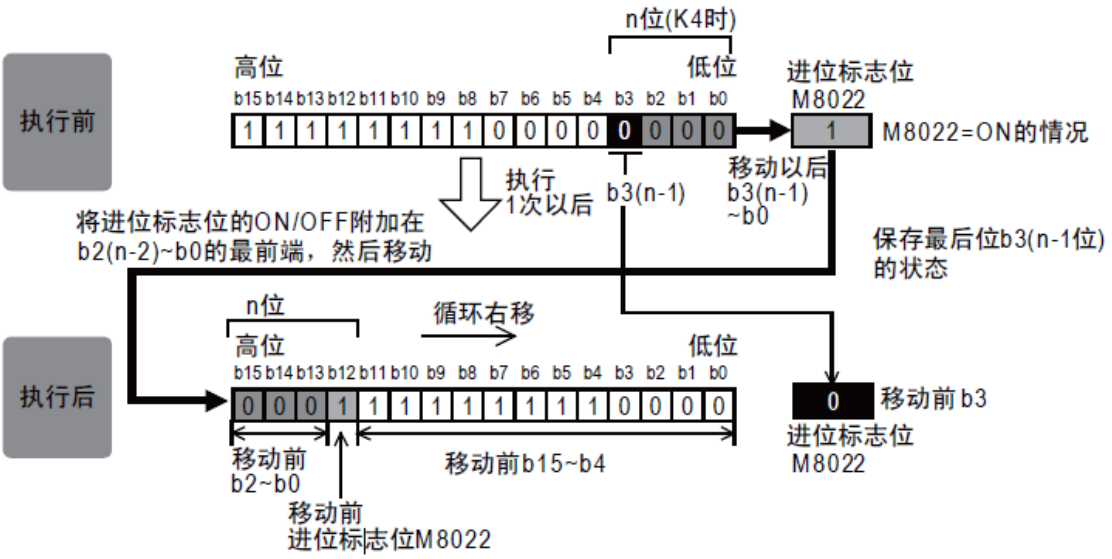
2、功能和动作说明

D 的所有位+1 位(进位标志位 M8022)向右移动 n 位。因为循环回路中有进位标志位，所以如果执行循环移位指令之前 M8022 就先 ON 或 OFF，则会被送入目标操作数中。

3、程序举例

将 D0 = -256 = 0xFF00 进行带进位的 4 位循环右移





4、注意事项

（1）相关软元件：M8022，进位标志，最后从最低位移出的位为 1 时置 ON。

（2)连续执行型(RCR、DRCR)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3)在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0) 。

2.5.4 RCL/带进位循环左移指令

使指定位数部分的位信息左移、循环的指令，带进位标志。

1、指令格式

FNC33	指令所占程序步	指令分类	指令格式
16 位	5	RCL	[(D)RCL(P) D n]
		RCLP	
32 位	9	DRCL	
		DRCLP	

操作数说明

操作数	内容	类型
D	进行循环左移的数据或移位后的结果	BIN16/32 位
n	循环移动的位数[1≤n≤16(16 位指令)，1≤n≤32(32 位指令)]	BIN16/32 位

操作数可用元件类型

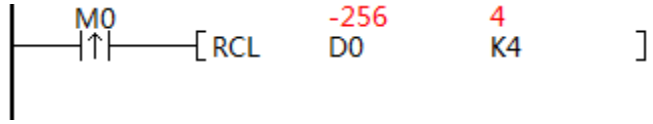
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

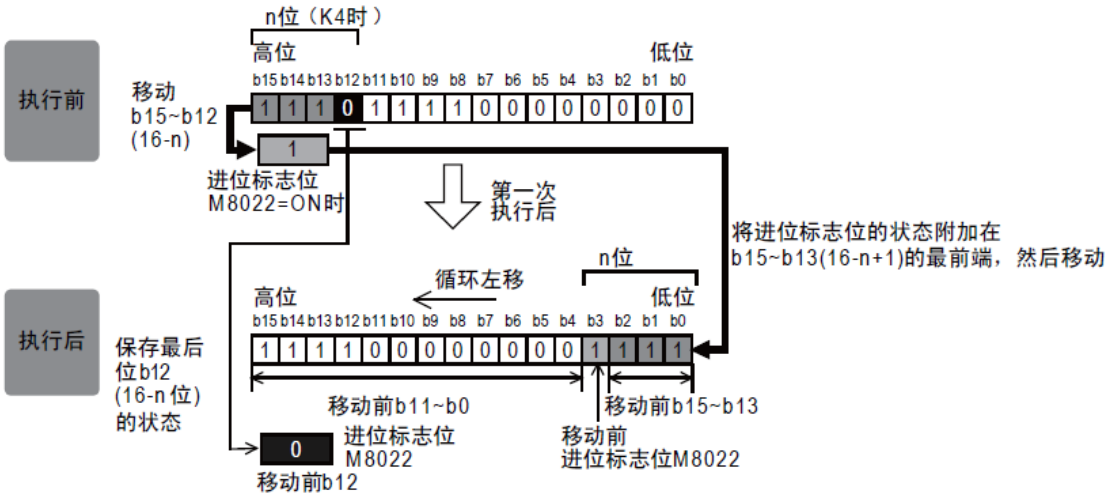
2、功能和动作说明

D 的所有位+1 位(进位标志位 M8022)向左移动 n 位。因为循环回路中有进位标志位，所以如果执行循环移位指令之前 M8022 就先 ON 或 OFF，则会被送入目标操作数中。

3、程序举例

将 D0 = -256 = 0xFF00 进行带进位的 4 位循环左移





4、注意事项

（1）相关软元件：M8022，进位标志，最后从最高位移出的位为 1 时置 ON。

（2)连续执行型(RCL、DRCL)指令的场合请注意每个扫描周期(运算周期)都会执行循环移位。

（3)在 D 中指定位数指定软元件时，仅 K4(16 位指令)或 K8(32 位指令)有效。(例如 K4Y010、K8M0)。

2.5.5 SFTR/位右移指令

右移指定的位长度的指令。移动后，从最高位开始传送 n2 点长度的 S 位软元件。

1、指令格式

FNC34	指令所占程序步	指令分类	指令格式
16 位	9	SFTR	-[SFTR(P) S D n1 n2]
		SFTRP	

操作数说明

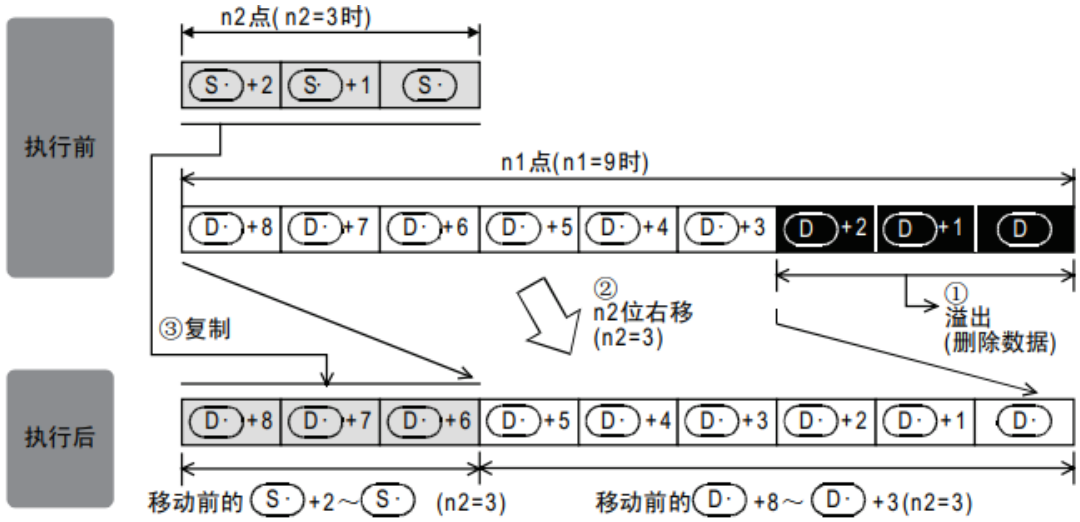
操作数	内容	类型
S	传送的位数据起始位元件	位
D	右移位的数据起始位软元件	位
n1	右移数据的位长度 n2≦n1≦1024	BIN16
n2	右移的位数 n2≦n1≦1024	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*	*											*					
D		*	*			*												*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 位数据，右移 n2 位(下记的①、②)。移位后，将 S 开始 n2 位数据传送（下记的③）到从 D+n1-n2 开始的 n2 位中。



3、程序举例



将 Y10~Y17 这 8 点位元件右移 3 位后，将 Y0~Y2 传入高位 Y15~Y17。

执行前

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	0	0	0	1	1	1	1	1

执行后

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	0	1	0	0	0	1	1

4、注意事项

（1）每次当指令输入从 OFF 变为 ON 时，执行 n2 位移位，指令输入状态一直为 ON 时，每个扫描周期(运算周期)都执行移位。

（2）传送源 S 和移位数据 D 重复时，报错 6710。

2.5.6 SFTL/位左移指令

左移指定的位长度的指令。移动后，从最低位开始传送 n2 点长度的 S 位软元件。

1、指令格式

FNC35	指令所占程序步	指令分类	指令格式
16 位	9	SFTL	-[SFTL(P) S D n1 n2]
		SFTLP	

操作数说明

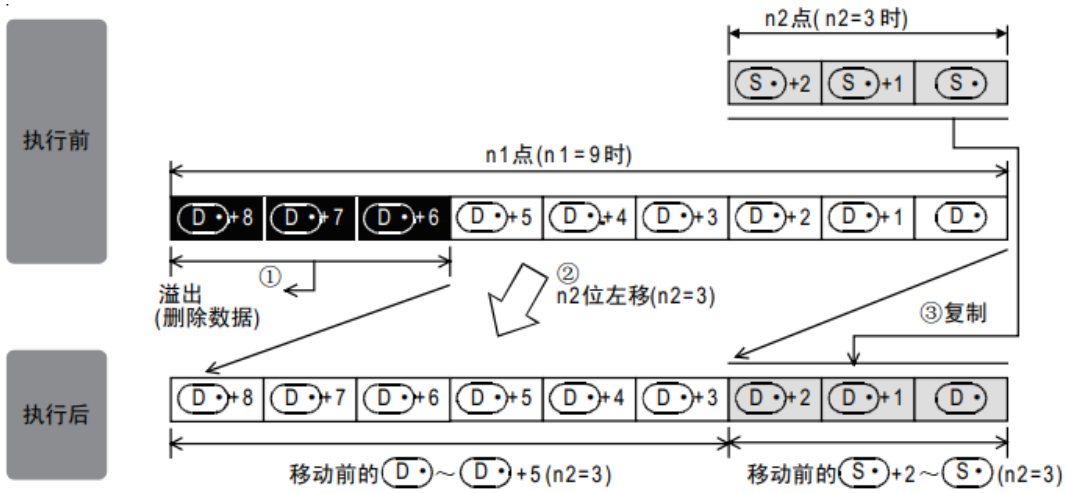
操作数	内容	类型
S	传送的位数据起始位元件	位
D	左移位的数据起始位软元件	位
n1	左移数据的位长度 n2≦n1≦1024	BIN16
n2	左移的位数 n2≦n1≦1024	BIN16

操作数可用元件类型

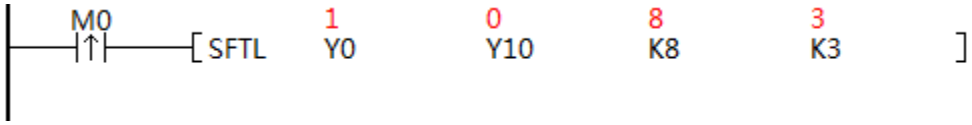
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*	*											*					
D		*	*			*												*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 位数据，左移 n2 位(下记的①、②)。移位后，将 S 开始 n2 位数据传送（下记的③）到从 D 开始的 n2 位中。



3、程序举例



将 Y10~Y17 这 8 点位元件左移 3 位后，将 Y0~Y2 传入低位 Y10~Y12。

执行前

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	1	1	1	1	0	0	0

执行后

软元件	7	6	5	4	3	2	1	0
Y0	0	0	0	0	0	1	0	1
Y10	1	1	0	0	0	1	0	1

4、注意事项

（1）每次当指令输入从 OFF 变为 ON 时，执行 n2 位移位，指令输入状态一直为 ON 时，每个扫描周期(运算周期)都执行移位。

（2）传送源 S 和移位数据 D 重复时，报错 6710。

2.5.7 WSFR/字右移指令

将 n1 个字软元件右移 n2 个字的指令。

1、指令格式

FNC36	指令所占程序步	指令分类	指令格式
16 位	9	WSFR	-[WSFR(P) S D n1 n2]
		WSFRP	

操作数说明

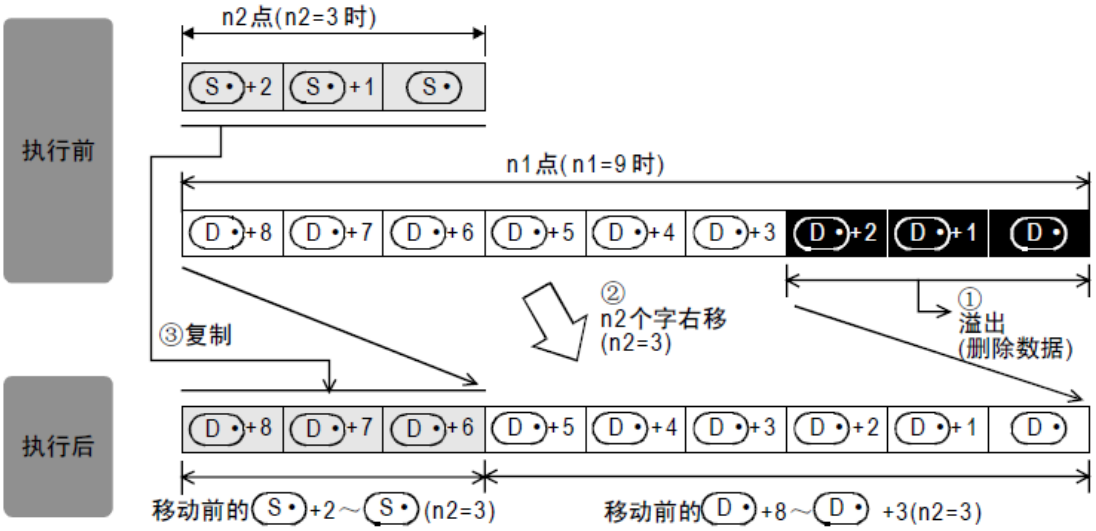
操作数	内容	类型
S	传送的字数据的起始字元件	BIN16
D	右移数据的起始字软元件	BIN16
n1	右移数据的字长度 n2≦n1≦512	BIN16
n2	右移的字数 n2≦n1≦512	BIN16

操作数可用元件类型

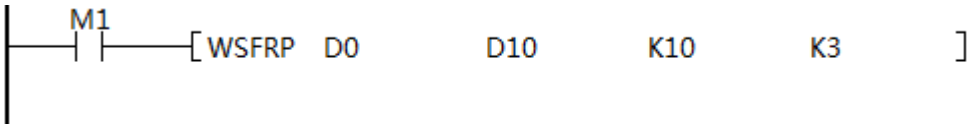
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

对于以 D 起始的 n1 个字软元件，右移 n2 个字(下面的①、②)。移位后，将 S 开始的 n2 点数据传送(下面的③)到从（D+n1-n2）开始的 n2 点中。



3、程序举例



将 D10~D19 这 10 点位元件右移 3 位后，将 D0~D2 传入 D17~D19。

执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D13	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19
D17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2

4、注意事项

（1）WSFRP 指令中，每次当指令输入从 OFF 变为 ON，就执行 n2 个字的移位，但是请注意 WSFR 指令中，每个运算周期都执行移位。

（2）传送源 S 和移位软元件 D 重复的时候，发生运算错误。(错误代码：K6710)

2.5.8 WSFL/字左移指令

将 n1 个字数据左移 n2 个字的指令。

1、指令格式

FNC37	指令所占程序步	指令分类	指令格式
16 位	9	WSFL	[WSFL(P) S D n1 n2]
		WSFLP	

操作数说明

操作数	内容	类型
S	传送的字数据起始字元件	BIN16

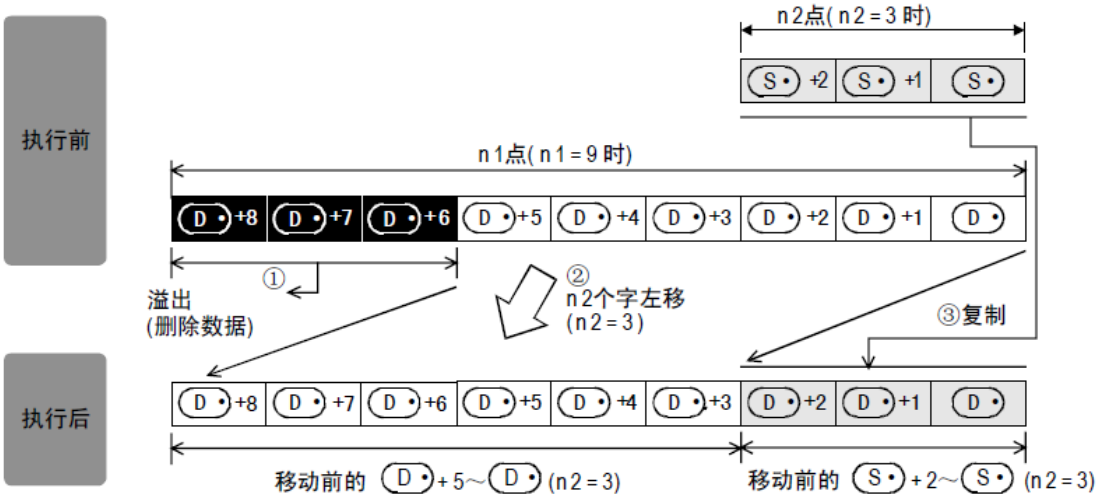
D	左移数据的起始字软元件	BIN16
n1	左移数据的字长度 n2≦n1≦512	BIN16
n2	左移的字数 n2≦n1≦512	BIN16

操作数可用元件类型

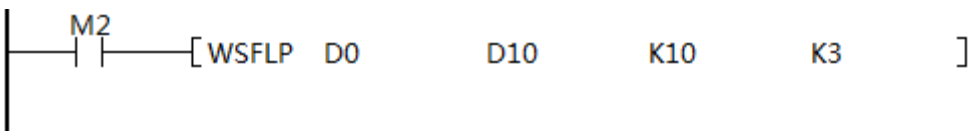
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n1																			*	*			
n2														*	*				*	*			

2、功能和动作说明

将以 D 起始的 n1 个字软元件，左移 n2 个字(下面的①、②)。移位后，将 S 开始的 n2 点传送(下面的③)到从 D 开始的 n2 点中。



3、程序举例



将 D10~D19 这 10 点位元件左移 3 位后，将 D0~D2 传入 D10~D12。

执行前

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D11	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D12	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D16	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16
D17	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	17
D18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	18
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	19

执行后

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
D13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10
D14	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11
D15	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	12
D16	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	13
D17	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	14
D18	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	15
D19	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	16

4、注意事项

（1）WSFLP 指令中，每次当指令输入从 OFF 变为 ON，就执行 n2 个字的移位，但是请注意 WSFL 指令中，每个运算周期都执行移位。

（2）传送源 S 和移位软元件 D 重复的时候，发生运算错误。(错误代码：K6710)

2.5.9 SFWR/移位写入指令

为实现先入先出或先入后出控制功能，先将准备的数据写入的指令，可用 SFRD 或 POP 指令读出数据。

1、指令格式

FNC38	指令所占程序步	指令分类	指令格式
16 位	7	SFWR SFWRP	[SFWR(P) S D n]

操作数说明

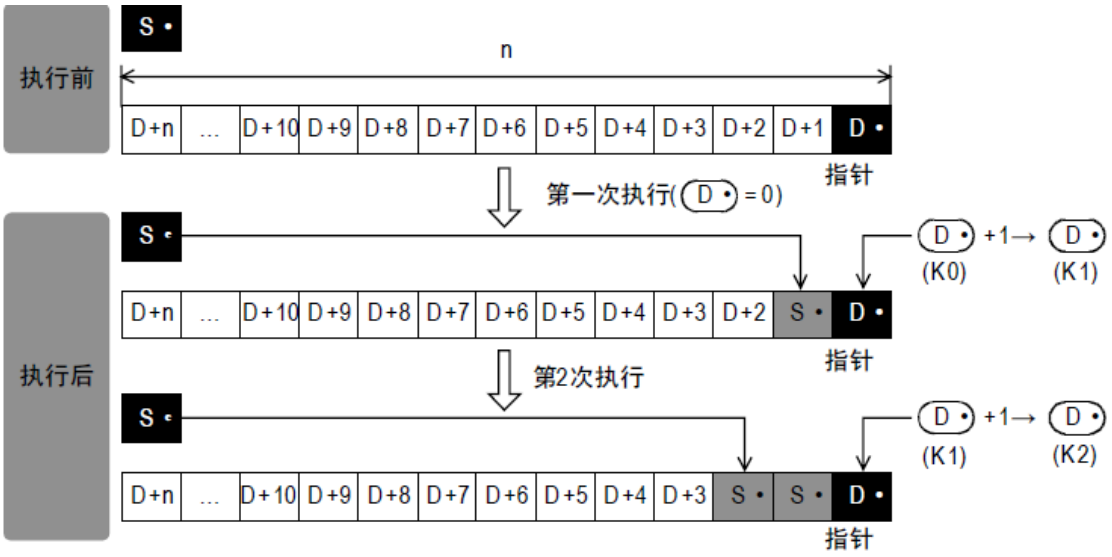
操作数	内容	类型
S	写入的数据	BIN16
D	保存数据并移位的起始字软元件 (D 保存数据点数，D+1 开始保存写入的数据)	BIN16
n	保存的数据点数+1 的值 2≦n≦512	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*			*					
n																			*	*			

2、功能和动作说明

在 D+1 开始的 n－1 点中依次写入 S 的内容，并对 D 中保存的数据数+1。



（1）X000 从 OFF 变为 ON 时， S 的内容被保存到 D+1 中， D+1 的内容变为 S 的值。

（2）S 的内容变化后再次执行输入从 OFF 变为 ON 后， S 的内容被保存到 D+2 中，D+2 的内容变为 S。(由于用连续执行型指令 SFWR，每个运算周期都依次被保存，因此请用脉冲执行型指令 SFWRP 编程。)

（3）从右端依次执行，D 中存储保存的数据点数。

3、程序举例

无

4、注意事项

（1）M8022 进位标志位的动作，指针 D 的内容超过 n－1 时，变为无处理(不写入)，且进位标志位 M8022 置 ON。

（2）请注意每个扫描周期(运算周期)都会依次保存(改写)。

2.5.10 SFRD/移位读出指令

先入先出形式的数据读出指令。

1、指令格式

FNC39	指令所占程序步	指令分类	指令格式
16 位	7	SFRD	[SFRD(P) S D n]
		SFRDP	

操作数说明

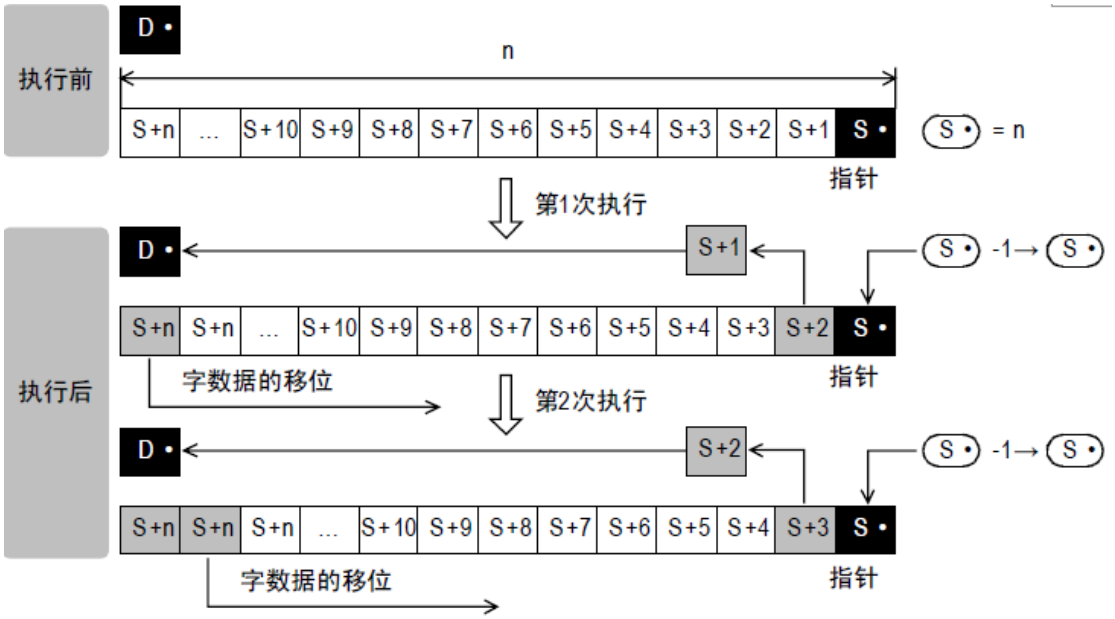
操作数	内容	类型
S	要读出的数据的起始字元件 (S 是数据点数，数据是从 S+1 开始的)	BIN16
D	读出的数据	BIN16
n	保存的数据的点数+1 的值。2≦n≦512	BIN16

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

2、功能和动作说明

使用 SFRD(FNC 38)指令被依次写入的 S1+1 传送(读出)到 D 中后，从 S1+1 开始的 n-1 点逐字右移。S 中保存的数据数-1。



（1）指令触点为 ON 时， S+1 的内容传送(读出)到 D 中。

（2）与此同时，S 的内容减少，左侧的数据逐字右移。

3、程序举例

无

4、注意事项

（1）SM8020 零位标志位的动作，数据的读出，通常从 S+1 开始执行，但是指针 S 的内容为 0 时，零标标志位 SM8020 动作。

（2）S+n 的内容不会因为读出而变化。

（3）每个扫描周期(运算周期)都会执行依次读出，但 S+n 的内容不会变化。

（4）S 为 0 时，不执行指令。

2.6 数据处理指令

处理复杂的数据或满足特殊的数据用途。

2.6.1 ZRST/成批复位指令

成批数据的复位（清零）指令。

1、指令格式

FNC40	指令所占程序步	指令分类	指令格式
16 位	5	ZRST ZRSTP	ZRST(P) D1 D2

操作数说明

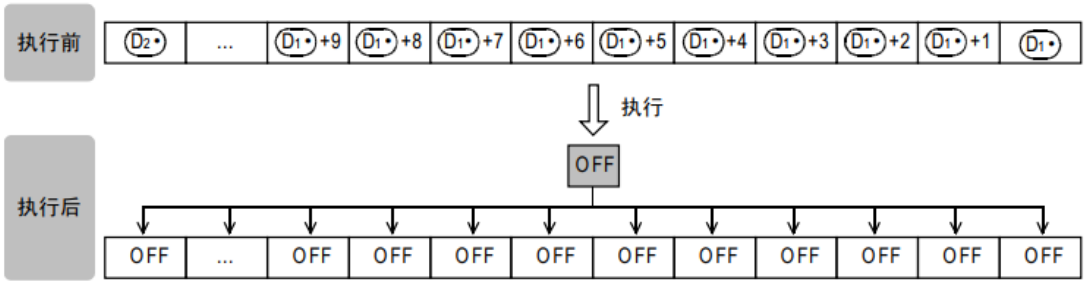
操作数	内容	类型
D1	成批复位的开始元件	BIN16 位
D2	成批复位的结束元件	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D1		*	*			*						*	*	*	*			*					
D2		*	*			*						*	*	*	*			*					

2、功能和动作说明

将同一种类的D1~D2 全部复位，位元件置OFF，字元件清零。



3、程序举例

略。

4、注意事项

（1）D1，D2 必须指定为同一种类的软元件，且编号 D1≦D2 编号，当 D1 编号大于 D2 时,D1 中指定的软元件只复位 1 点。

（2） ZRST 作为 16 位处理指令。也可以在 D1，D2 中指定 32 位计数器，但是不能出现 D1 和 D2 的寄存器位数不一样的情况。

2.6.2 DECO/译码指令

将数字数据中任意一个转换成 1 点的 ON 位的指令。根据 ON 位的位置可以将位编号读成数值。

1、指令格式

FNC41	指令所占程序步	指令分类	指令格式
16 位	7	DECO	-[DECO(P) S D n]
		DECOP	

操作数说明

操作数	内容	类型
S	要译码的数据	BIN16 位
D	译码结果	BIN16 位
n	保存译码结果的软元件的位点数(n=1～8)(n=0 时为不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*						*	*	*	*	*	*	*	*	*			
D		*	*			*						*	*	*	*			*					
n																			*	*			

2、功能和动作说明

S 的值相对应的 D～D+2^n-1 中的第 S 个位置 ON。

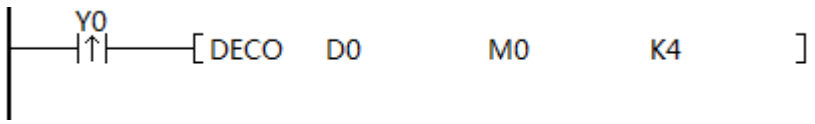
6. D 为位软元件时：1≦n≦8，S 中指定的软元件的 n 位数(1≦n≦8)，在 D 中被译码。S 都为 0 时，位软元件 D 为 ON。n=8 时， 最大 2^8=256 点。

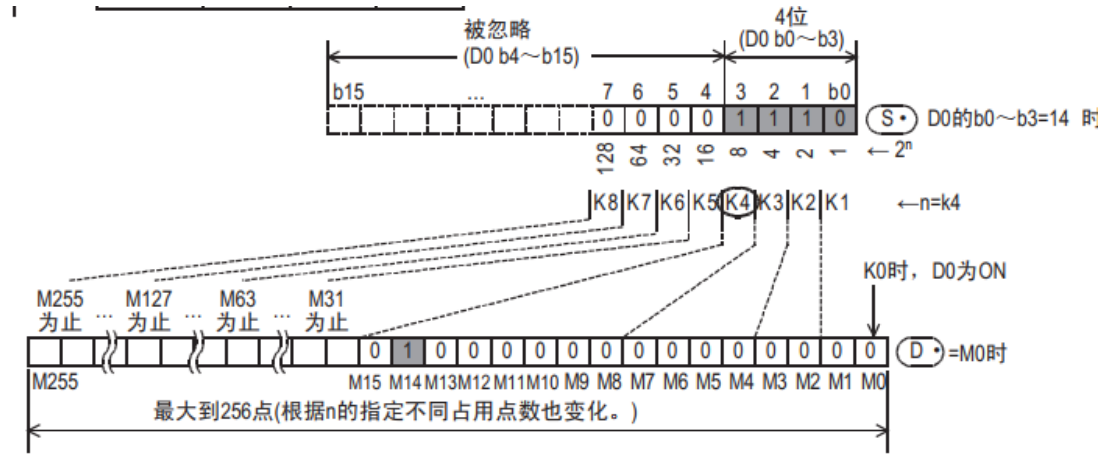
7. D 为字软元件时：1≦n≦4，S 的低 n 位在 D 中被译码。S 都为 0 时，位软元件 D 的 b0 为 ON。

3、程序举例

（1）D 为位软元件

D0 的值(当前值取 14)在 M0～M15 中译码，M14 置 ON，其它位为 0。

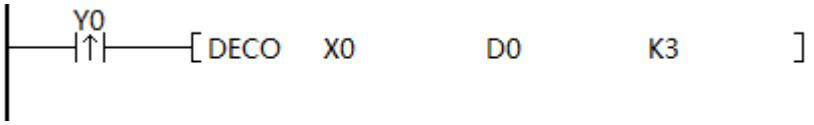


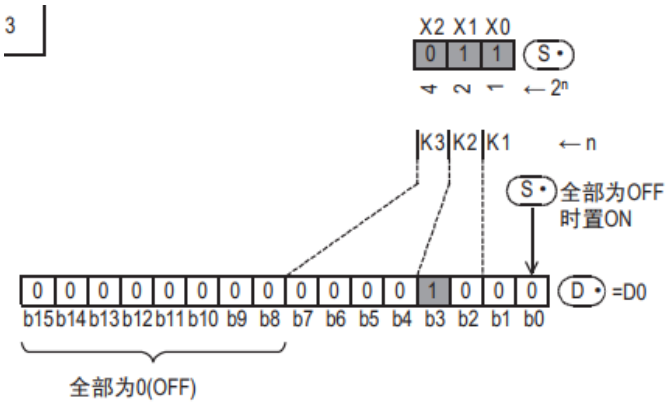


- D0 的 b0～b3 的值为 14(0+2+4+8)时，M0 开始的第 15 号的 M14 为 1(ON)。
- D0=0 时，M0 为 1(ON)。
- 使 n=K4，根据 D0(0～15)的数值，M0～M15 中任意一个为 1 点(ON)。
- 若使 n 在 K1～K8 之间变化，D0 就可以对应 0～255 的数值。

（2）D 为字软元件

X000～X002 的值(X000、X001 为 ON，X002 为 OFF)在 D0 中译码。





- X000～X002 的值为 3(1+2+0)时，b0 开始的第 4 号的 b3 为 1(ON)。
- X000～X002 都为 0(OFF)时，b0 为 1(ON)。

4、注意事项

（1）n 在 K1～K8 之间变化时，注意目的寄存器 D 为位软元件的范围不能与其他控制重复。

（2）n 为 0 时，指令不处理。

2.6.3 ENCO/编码指令

求出在数据中 ON 位的位置的指令。

1、指令格式

指令格式见下表

FNC42	指令所占程序步	指令分类	指令格式
16 位	7	ENCO	-[ENCO(P) S D n]
		ENCOP	

操作数说明

操作数	内容	类型
S	要编码的数据	BIN16 位
D	编码结果	BIN16 位
n	保存编码结果的软元件的位点数(n=1～8)(n=0 时为不处理)	BIN16 位

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S	*	*	*			*						*	*	*	*	*	*	*					
D												*	*	*	*	*	*	*					
n																			*	*			

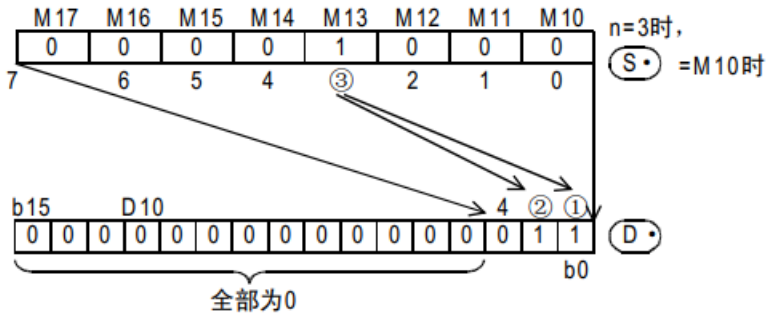
操作数可用元件类型

2、功能和动作说明

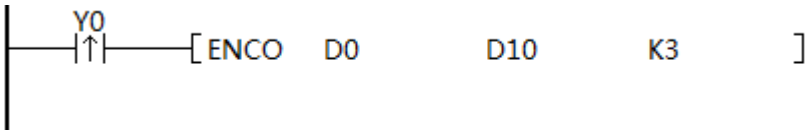
在 D 中保存 S 的 2^n 位编码后的值，编码就是将 ON 位的位置值转换成 BIN 数据。

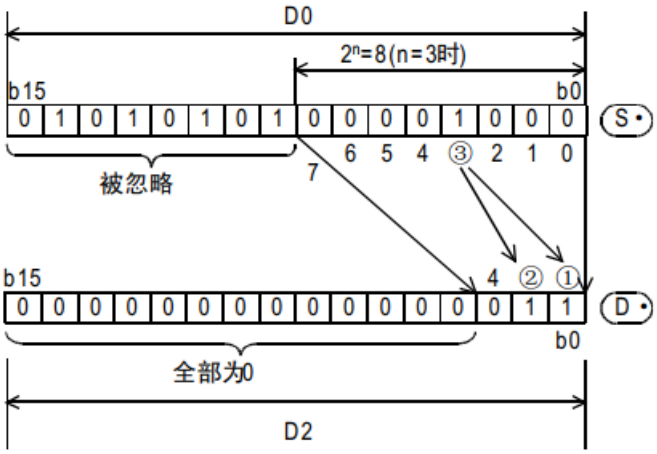
（1）S 为位软元件：1≦n≦8，S 开始 2^n 个的 ON 位位置，在 D 中编码。n=8 时，S 为最大 256 点。D 的编码结果从高位到低位 n 位，全部为 0(OFF)。





（2）S 为字软元件：1≦n≦4，到 S 中指定的软元件的位 2^n 个为止的 ON 位位置，都在 D 中编码。D 的编码结果从高位到低位 n 位，全部为 0(OFF)。





3、程序举例

无

4、注意事项

S 的数据中多个位为 ON 的情况下，忽略低位侧，对高位侧的 ON 位进行编码。

2.6.4 SUM/计算 ON 位数指令

计算指定的的数据中有多少个位为“1”(ON)的指令。

1、指令格式

FNC43	指令所占程序步	指令分类	指令格式
16 位	5	SUM	(D)SUM(P) S D
		SUMP	
32 位	9	DSUM	
		DSUMP	

操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位
D	目标数据	BIN16/32 位

操作数可用元件类型

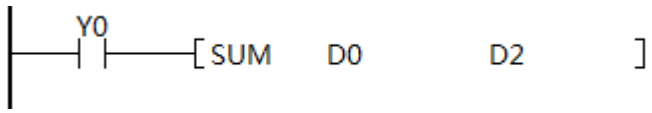
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

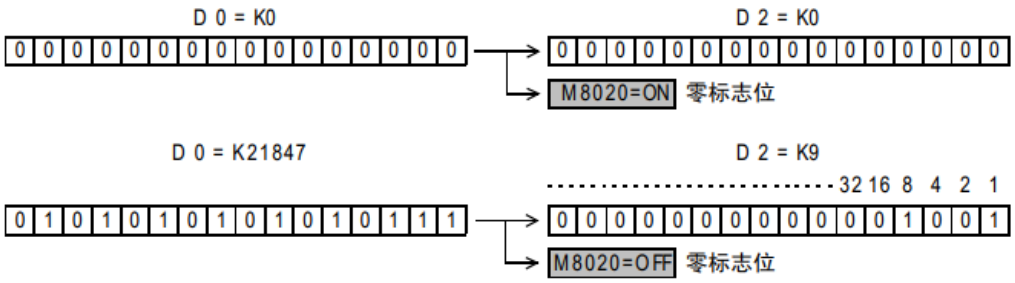
2、功能和动作说明

对 S 中为 ON 的位进行计数后保存到 D 中。S 都为 0（OFF 时），零位标志位 M8029 为 ON。

3、程序举例

Y0 为 ON 时，对 D0 中 ON 的位数计数后保存到 D2 中





4、注意事项

相关标志位 M8020 的动作。

2.6.5 BON/ON 位的判定指令

检测指定位的位置为 ON 还是 OFF 的指令。

1、指令格式

FNC44	指令所占程序步	指令分类	指令格式
16 位	7	BON	[(D)BON(P) S D n]
		BONP	
32 位	13	DBON	
		DBONP	

操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位
D	目标数据	位
n	要判定的位位置[n: 0～15(16 位指令)，n: 0～31(32 位指令)]	BIN16/32 位

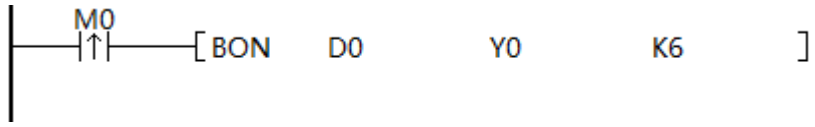
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*	*			*	*											*					
n														*	*				*	*			

2、功能和动作说明

S 的 n 位的状态（ON/OFF）输出到 D。

3、程序举例



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	75

D0 的第 6 位为 1 时，Y0=ON

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11

D0 的第 6 位为 0 时，Y0=OFF

4、注意事项

将 D、R 指定为 32 位指令的 n 时，[n+1,n]的 32 位值便生效，敬请注意。
DBON D0 M0 R0 时，则 n=[R1, R0]

2.6.6 MEAN/平均值指令

求数据的平均值的指令。

1、指令格式

FNC45	指令所占程序步	指令分类	指令格式
16 位	7	MEAN	[(D)MEAN(P) S D n]
		MEANP	
32 位	13	DMEAN	
		DMEANP	

操作数说明

操作数	内容	类型
S	求平均值的源数据	BIN16/32 位
D	平均值	BIN16/32 位
n	求平均值数据个数(n=1～64)	BIN16/32 位

操作数可用元件类型

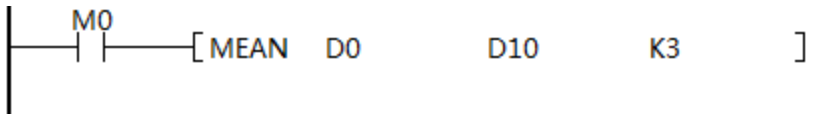
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n														*	*				*	*			

2、功能和动作说明

将 S 开始的 n 个 16 位或 32 位（根据指令类型）数据的平均值保存到 D 中，余数舍去。

3、程序举例

将 D0、D1、D2 的数据相加，除以 3 后求得的值保存到 D10 中



D0 = 33， D1 = 44， D2 = 66， M0 置位后 D10 = （D0 + D1 + D2）/ 3 = 47。

4、注意事项

（1）软元件编号溢出时，在可能的范围内将 n 作为较小的值处理。

（2）注意 n 的范围(n=1～64)，超出报错。

2.6.7 ANS/信号报警置位指令

对信号报警器用的状态(S900～S999)进行置位用的指令。

1、指令格式

FNC46	指令所占程序步	指令分类	指令格式
16 位	7	ANS ANSP	ANS S m D

操作数说明

操作数	内容	类型
S	计时定时器	BIN16/32 位
m	判定时间[m=1～32,767(100ms 单位)]	BIN16/32 位
D	信号报警器	位

操作数可用元件类型

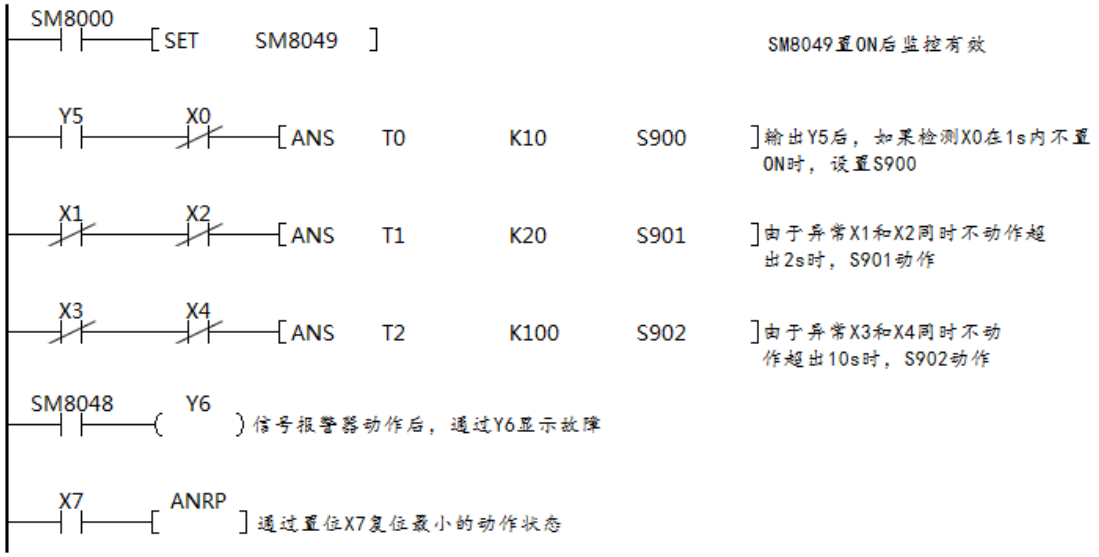
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*						*					
m														*	*				*	*			
D						*												*					

2、功能和动作说明

指令输入超出判定时间[m×100ms，定时器 S]以上持续为 ON 时，D 置位。指令输入在不满判定时间[m×100ms]就已 OFF 的情况下，复位判定用定时器 S 的当前值,不置位 D。此外，指令输入 OFF 后，判定用定时器复位。

3、程序举例

如下所示，编写诊断外部故障用的程序，如监控 SD8049(ON 状态最小编号)的内容时，会显示 S900～S999 中为 ON 的状态的最小编号。同时发生多个故障时，排除了最小编号的故障后可以得知下一个故障编号。



4、注意事项

（1）S 为 T 软元件时，范围为 T00～T199，报警用状态软元件范围是 S900～S999。

（2）注意与 S900～S999 相关的特殊寄存器的动作。

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON,状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

2.6.8 ANR/信号报警复位指令

对信号报警器(S900～S999)中已经置 ON 的小编号进行复位。

1、指令格式

FNC47	指令所占程序步	指令分类	指令格式
16 位	1	ANR	ANR(P)
		ANRP	

2、功能和动作说明

16 位运算指令，ANR 指令输入为 ON 后，将信号报警器 S900～S999 中运行中的状态复位。如有多个状态动作时，复位编号最小的一个状态。再次使指令输入为 ON 后，在动作的信号报警器用状态(S900～S999)中，下一个最小的编号被复位。

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON,状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

3、程序举例

可参考 ANS 指令。

4、注意事项

（1）相关寄存器的动作:

软元件	名称	内容
SM8049	信号报警器有效	SM8049 置 ON 后，下面的 SM8048 和 SD8049 工作。
SM8048	信号报警器动作	SM8049 为 ON,状态 S900～S999 中任一动作的时候，SM8048 置 ON。
SD8049	ON 状态最小编号	保存 S900～S999 中动作的最小编号。

（2）使用 ANR 指令时，每个运算周期都依次复位最小标号，ANPR 指令只执行 1 个运算周期，根据实际需求来选择对应的指令以复位对应的报警器。

2.6.9 SQR/开方运算指令

求平方根(开根号)的指令。

1、指令格式

FNC48	指令所占程序步	指令分类	指令格式
16 位	5	SQR	-[(D)SQR(P) S D]
		SQRP	
32 位	9	DSQR	
		DSQRP	

操作数说明

操作数	内容	类型
S	源数据	BIN16/32 位
D	目标数据	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*	*	*			
D														*	*			*					

2、功能和动作说明

计算出 S 的数据的平方根后，保存到 D 中。

3、程序举例

略。

4、注意事项

（1）当运算结果为小数时，舍弃小数部分，将整数结果填入 D，SM8021 置 ON。

（2）结果为 0 时，SM8020 置 ON。

2.6.10 FLT/BIN 整数→2 进制浮点数转换指令

将 BIN 整数值转换成 2 进制浮点数(实数)的指令。

1、指令格式

FNC49	指令所占程序步	指令分类	指令格式
16 位	5	FTL	[(D)FTL(P) S D]
		FTLP	
32 位	9	DFTL	
		DFTLP	

操作数说明

操作数	内容	类型
S	BIN 整数	BIN16/32 位
D	2 进制浮点数(实数)	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

将 S 的 BIN 整数值数据转换成 2 进制浮点数(实数)值后，保存在 D 中。

3、程序举例

略。

4、注意事项

16 位和 32 位指令中的 D 都是占用 2 个字软元件。

2.7 高速处理指令

可以按最新的输入输出信息进行程序控制，并能有效利用数据高速处理能力进行中断处理。

2.7.1 REF/输入输出刷新指令

获得最新的输入(X)，以及将输出(Y)扫描结果立即输出的指令。

1、指令格式

FNC50	指令所占程序步	指令分类	指令格式
16 位	5	REF	[REF(P) D n]
		REFP	

操作数说明

操作数	内容	类型
D	刷新的位软元件(X、Y)	位
n	刷新点数(8～256 中 8 的倍数)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D	*	*																					
n																			*	*			

2、功能和动作说明

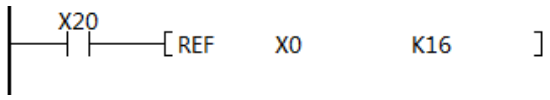
将 D 地址开始的 n 个元件状态进行立即更新，由于 PLC 访问端口是按字节访问的特性，故要求 S 的地址为 X0、X10、.....Y0、Y10、.....等最低位为 0 的编号元件。且 n 的值必须是 8 的整数倍（n=8~256）。

正常情况下，输入端口 X 的状态读取在每次程序开始执行扫描之前，实际的输入端口状态变化延迟决定于输入元件的滤波时间，X0~X7，X10~X17 有数字滤波功能，滤波时间在 0~60ms 可设（REFF 指令，默认滤波时间为 10ms）；输出端口 Y 的状态刷新则在每次程序执行扫描

完毕（执行到 END）之后批次执行，这样 IO 处理会有一定的延迟。若应用中需要最新的输入信号以及希望立即输出运算结果时，可以使用立即刷新指令 REF。

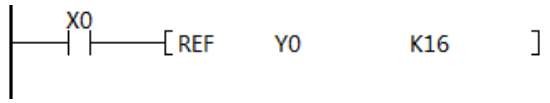
3、程序举例

（1）立即刷新输入信号



X20 为 ON 时，会立即读取 X0~X17 的输入点状态，更新输入信号，没有产生输入延迟。

（2）立即刷新输出信号



X0 为 ON 时，会立即将 Y0~Y17 的状态进行刷新，输出信号立即变更，不必到 END 指令才输出。

4、注意事项

根据实际的应用需求应用此指令，可在当前位置立即刷新输入输出，而不是按照梯形图的运行流程来进行输入输出的刷新。

2.7.2 REFF/带滤波的输入输出刷新指令

输入 X0~X17 的输入滤波器为数字式滤波器，采用 REFF 指令和 SD8020 可以更改滤波器的时间。

1、指令格式

FNC51	指令所占程序步	指令分类	指令格式
16 位	3	REFF	REFF(P) n
		REFFP	

操作数说明

操作数	内容	类型
n	滤波器的时间[0~60ms]	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
n														*	*				*	*			

2、功能和动作说明

根据输入的数字式滤波器时间刷新输入 X000～X017 的 16 点映像存储区。

关于滤波器时间在不同的使用情境下，会有所区别，当前以 X3-32 点为例说明：

（1）当 n 设定为 0 时，实际对应的输入滤波器滤波时间如下

输入编号	输入滤波常数设定为 0 时的滤波值
X0~X7（高速端子）	3us （只对 X1 系列 PLC）
X10~X17（普通端子）	1ms

（2）用于高速或中断的输入端子，在高速或中断程序运行时，一律按最小的滤波时间 1.5us 进行滤波器设定。

3、程序举例

略。

4、注意事项

注意高速中使用的输入和输入中断功能中指定的中断指针的输入，此情景中 REFF 指令刷新 SD8020 的滤波器时间不能正常处理，而是按上述说明进行滤波器设定。

2.7.3 HSCS/高速比较置位指令

每次执行时，都将高速计数器的计数值和指定值做比较，如果两个值相等，立即置位比较输出的指令。

1、指令格式

FNC53	指令所占程序步	指令分类	指令格式
32 位	13	DHSCS	-[DHSCS S1 S2 D]
		\	

操作数说明

操作数	内容	类型
S1	高速比较设定值	BIN32 位
S2	高速计数器[C235～C255]	BIN32 位
D	被置位的对象元件	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2													*					*					
D		*	*			*	*											*					*

2、功能和动作说明

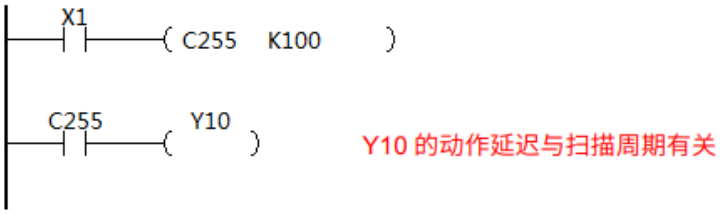
（1）当 S2 计数器的当前值等于设定值 S1 时，立即置位 D。S2 变量必须为高速计数器 C235~C255。

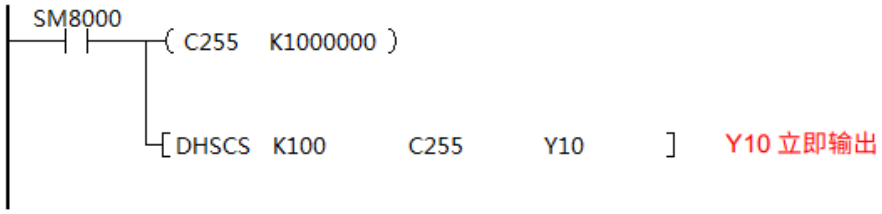
（2）D 为普通位元件时：当为本体的输出范围端口时为立即输出；当为本体以后的端口时，会等本次用户程序扫描完毕才会输出；当为 M、S、Dx.y 变量时，立即刷新。

（3）D 为调用计数中断子程序指针时：当 D 为 I010~I060 时，即为调用高速计数器中断 0~5 的子程序。当然必须写好相应的中断子程序、开启相应中断允许标志和全局中断允许标志等，才能正常相应计数器中断。SM8059 置 ON 则禁止了所有的高速计数器中断（I010~I060）。

3、程序举例

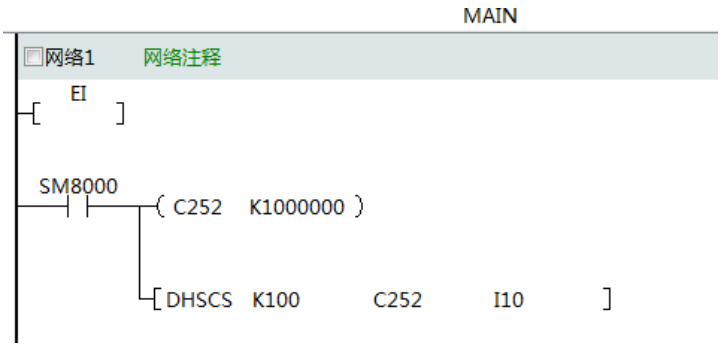
（1）置位普通位元件





（2）调用计数中断子程序

主程序：



I010 中断子程序：



4、注意事项

（1）使用 HSCS 指令时，应保证所使用的计数器已被启用，否则计数器的值将不会变化；

（2）计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即置位。例如上面例一中，若 C255 的当前值变为 99->100 或 101->100 时，Y10 立即置位（如果是属于本体的输出端子才会立即置位），且一直保持该状态，之后即使 C255 与 K100 的比较结果变成不相等，Y10 仍然保持 ON 状态，除非有另外的复位操作。

（3）指令的比较输出结果只取决于脉冲输入时的比较结果，即使采用 DMOV、DADD 等指令改写 C235~C255 的内容，若没有脉冲输入，比较输出也不会变化。

（4）当 HSCS 指令的输出目标为计数器中断 I010~I060 时，每个中断号只能使用 1 次，不可重复。

（5）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在每一路高速输入在 6 条指令之内，4 路高速输入最大 24 条。但是 HSZ 指令特殊模式（高速表格比较模式、频率控制模式）在梯形图里仅能同时驱动 1 个指令，也不影响其他路的 6 条。

2.7.4 HSCR/高速比较复位指令

每次执行时，都将高速计数器的计数值和指定值做比较，如果两个值相等，立即复位比较输出的指令。

1、指令格式

FNC54	指令所占程序步	指令分类	指令格式
32 位	13	DHSCR \	[DHSCR S1 S2 D]

操作数说明

操作数	内容	类型
S1	高速比较设定值	BIN32 位
S2	高速计数器[C235～C255]	BIN32 位
D	被复位的位元件	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2													*					*					
D		*	*			*	*						*					*					

2、功能和动作说明

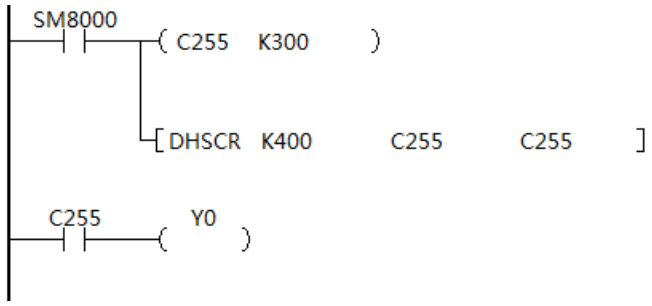
（1）当 S2 计数器的当前值等于设定值 S1 时，立即复位 D。S2 变量必须为高速计数器 C235~C255。

（2）D 为普通位元件时：当为本体的输出范围端口时为立即复位；当为本体以后的端口时，会等本次用户程序扫描完毕才会复位；当为 M、S、Dx.y 变量时，立即复位。

（3）D 为计数器时：可以复位任意计数器，常用于复位本体高速计数器。

3、程序举例

C255 的当前值变为 400 后，立即执行 C255 的自我复位，当前值为 0，输出触点为 OFF。



4、注意事项

（1）使用 HSCSR 指令时，应保证所使用的计数器已被启用，否则计数器的值将不会变化；

（2）计数器是以中断方式响应计数器的输入信号，及时比较，若本次比较时满足匹配关系，比较输出立即复位。

（3）指令的比较输出结果只取决于脉冲输入时的比较结果，即使采用 DMOV、DADD 等指令改写 C235~C255 的内容，若没有脉冲输入，比较输出也不会变化。

（4）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在每一路高速输入在 6 条指令之内，4 路高速输入最大 24 条。但是 HSZ 指令特殊模式（高速表格比较模式、频率控制模式）在梯形图里仅能同时驱动 1 个指令，也不影响其他路的 6 条。

2.7.5 HSZ/高速区间比较指令

将高速计数器的当前值和 2 个区间值进行比较，根据比较结果置位输出。该指令可设置成高速表格比较模式和频率控制模式。

1、指令格式

FNC55	指令所占程序步	指令分类	指令格式
32 位	17	DHSCZ \	-[DHSZ S1 S2 S D]

操作数说明

操作数	内容	类型
S1	比较区间下限	BIN32 位
S2	比较区间上限	BIN32 位
S	高速计数器[C235～C255]	BIN32 位
D	输出比较结果	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*		*	*	*	*			
S2								*	*	*	*	*	*	*	*		*	*	*	*			
S													*					*					
D		*	*			*	*																

2、功能和动作说明

（1）普通模式

• S1 为区间下限，S2 为区间上限，满足 S1<=S2；

• S 必须为 C235~C255；

• D 占用三个连续地址单元，当为 Y0~Y17 端口范围时，立即输出，其他 Y 端口在本次用户程序扫描完毕才输出；当为 M、S、Dx.y 变量时，为立即刷新。

当 S1 ＞ S 时，D 置位；

当 S1 ≤ S ＜ S2 时，D+1 置位；

当 S ≥ S2 时，D+2 置位；

（2）高速表格比较模式

当参数 D 为特殊辅助继电器 M8130 时，即指定高速表格比较模式，此模式下，注意操作数的意义。

- S1 表示比较表格的起始地址，只能是数据寄存器 D；
- S2 表示表格的行数（仅 K、H）限制为 K1~K128/H1~H80；
- S 必须为 C235~C255；
- D 为 M8130，指定高速表格比较模式。

比较表格的形式如下：

比较数据（32 位）	输出 Y 编号	SET/RST	表格计数器（D8130）
S1+1，S1	S1+2	S1+3	0 ↓
S1+5，S1+4	S1+6	S1+7	1 ↓
S1+9，S1+8	S1+10	S1+11	2 ↓
...	...	...	...
S1+(S2*4-3),S1+(S2*4-4)	S1+(S2*4-2)	S1+(S2*4-1)	S2-1 ↓ 从 0 开始重复

其中：每一行表格需要占用 4 个软元件；输出 Y 编号用 16 进制数指定，Y0~Y7 对应 H0~H7，Y10~Y17 对应 H10~H17.....Y370~Y377 对应 H370~H377；通过设定的内容 0/1 控制对应表格中的输出 Y 的 ON/OFF。

一般通过数据传送指令来构建表格，比如

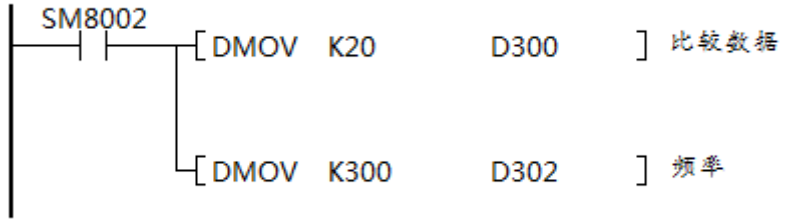


（3）频率控制模式

当参数 D 为特殊辅助继电器 SM8132 时，即指定频率控制模式，根据表格频率切换来控制 PLSY 指令进行高速输出的频率，此模式下，注意操作数的意义。

- S1 表示比较表格的起始地址，只能是数据寄存器 D；
- S2 表示表格的行数（仅 K、H）限制为 K1~K128/H1~H80；
- S 必须为 C235~C255；
- D 为 SM8132，指定频率控制模式。

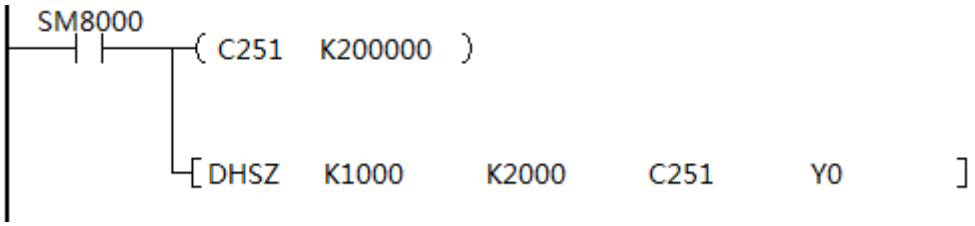
一般通过数据传送指令来构建频率表格，比如



3、程序举例

（1）普通模式

高速计数器 C251 的当前值如下变化(计数)时，在 Y000、Y001、Y002 的任意一个中输出比较结果。

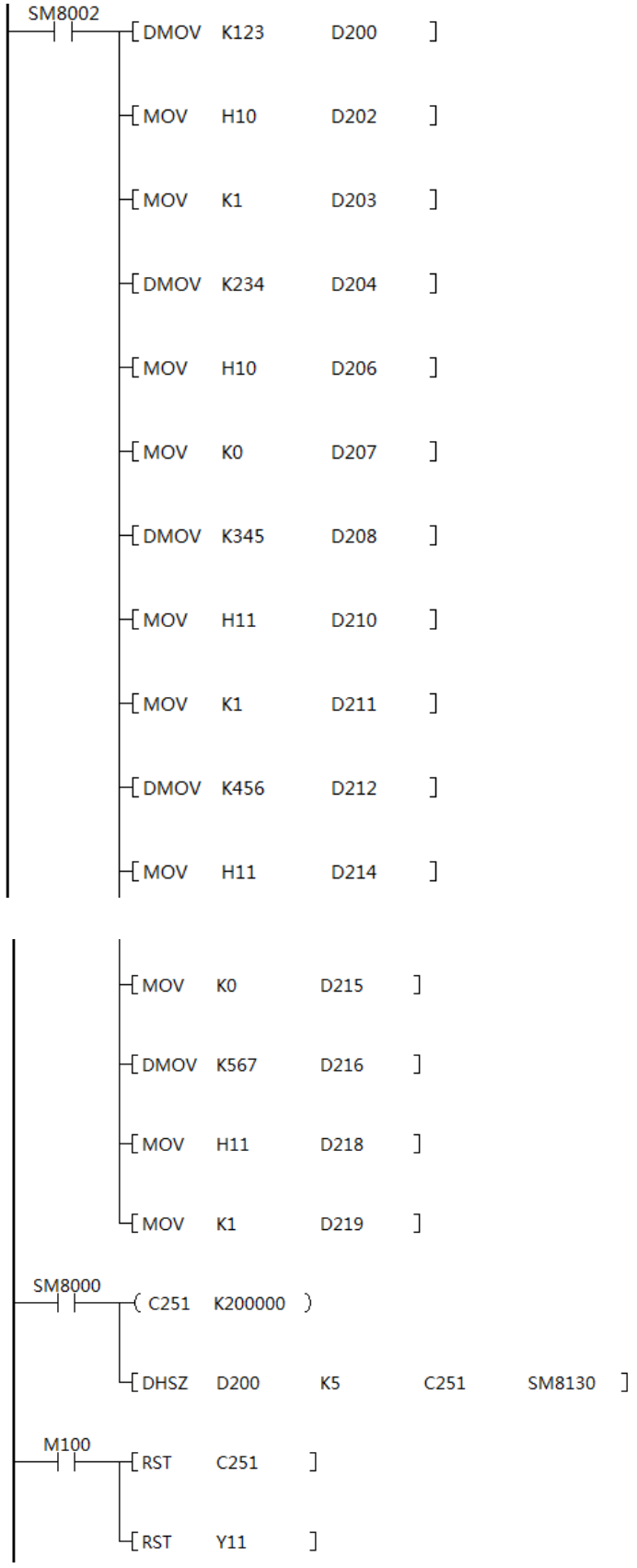


当 K1000 > C251 时，置位 Y0；

当 K1000 ≤ C251 < K2000 时，置位 Y1；

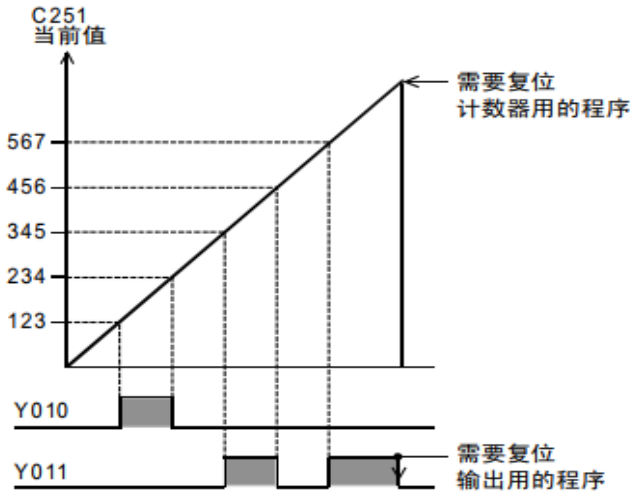
当 C251 ≥ K2000 时，置位 Y2；

（2）高速表格比较模式


SM8002
DMOV K123 D200
MOV H10 D202
MOV K1 D203
DMOV K234 D204
MOV H10 D206
MOV K0 D207
DMOV K345 D208
MOV H11 D210
MOV K1 D211
DMOV K456 D212
MOV H11 D214
MOV K0 D215
DMOV K567 D216
MOV H11 D218
MOV K1 D219
SM8000
C251 K200000
DHSZ D200 K5 C251 SM8130
M100
RST C251
RST Y11

梯形图转换成比较表格：

比较数据（32 位）	输出 Y 编号	SET/RST	表格计数器（SD8130）
D201,D200 K123	D202 H10	D203 K1	0 ↓
D205,D204 K234	D206 H1	D207 K0	1 ↓
D209,D208 K345	D210 H11	D211 K1	2 ↓
D213,D212 K456	D214 H11	D215 K0	3 ↓
D217，D216 K567	D218 H11	D219 K1	4 ↓ 从 0 开始重复



①执行该指令后，表格数据的最上一行的表格被设置成比较对象数据。

②高速计数器 C251 的当前值和比较对象的表格数据一致时，比较对象的表格数据中指定的输出 Y 编号被执行 SET/RST。并且，这个输出 Y 不等到执行 END 指令的输出刷新，直接执行输出处理。

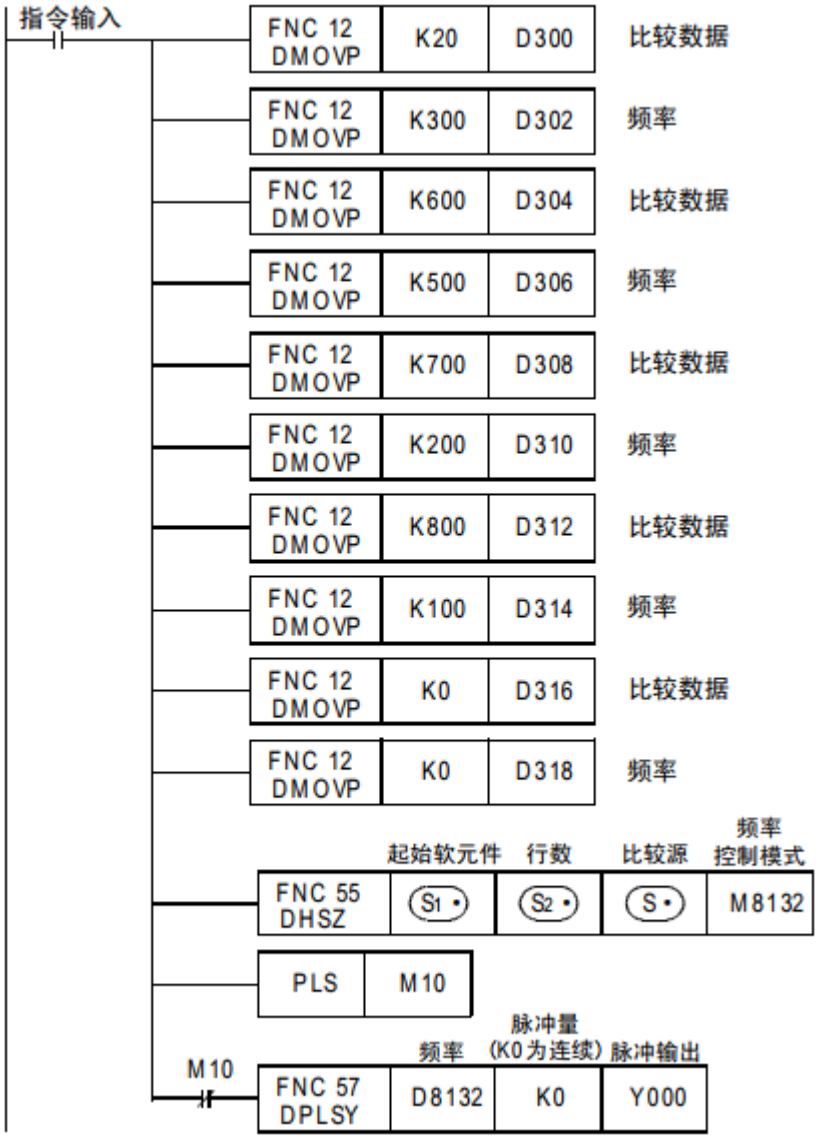
③表格计数器 SD8130 的当前值+1。

④比较对象的表格数据移到下一行的表格。

⑤在表格计数器 SD8130 的当前值变为 4 之前，重复②～③的动作。当前值变为 4 时，返回到动作①，表格计数器复位(D8130=0)。此时，结束标标志位 SM8131 为 ON。

⑥指令触点断开(指令执行 OFF)后，中止执行指令，表格计数器 SD8130 复位(D8130=0)。

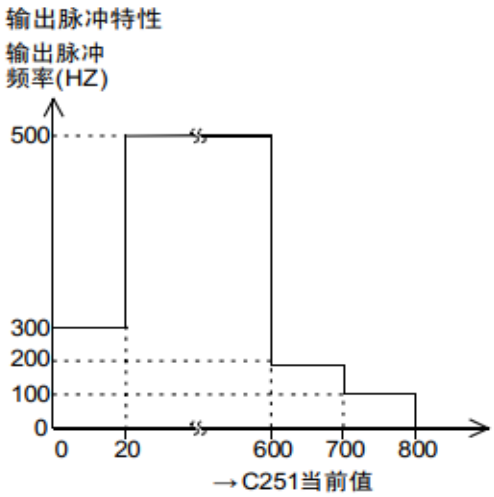
（3）频率控制模式



梯形图转换为频率表格：

比较数据（32 位）	输出 Y 编号	SET/RST	表格计数器（SD8130）
D301,D300 K20	D302，D303 K300	D203 K1	0 ↓
D305,D304 K600	D306，D307 K500	D207 K0	1 ↓
D309,D308 K700	D310，D311 K200	D211 K1	2 ↓
D313,D312 K800	D314，D315 K100	D215 K0	3 ↓

D317，D316 K0	D318，D319 K0	D219 K1	4 ↓ 从 0 开始重复



①在构成表格的数据寄存器中，像本程序举例中一样，预先写入定下的数据。
②一起到 S 中指定的高速计数器的当前值与(D301，D300)的值相等为止，PLSY 指令的输出频率都为(D303，D302)的值。(D302 为低 16 位。D303 为高 16 位，一般一直为 0)
③此后，进入第 2 行的动作，依次执行各行的动作。
④执行完最后一行的动作后，结束标志位 M8133 动作，返回到第 1 行重复动作。
⑤希望在最后一行中停止动作时，将最后的表格的频率设定为 K0。
⑥指令输入 OFF 后，脉冲输出变为 OFF，表格计数器 SD8131 也被复位。
⑦在该指令初次执行后的 END 指令处完成表格的制作，此后该指令有效。
⑧因此,PLSY 指令的执行,是从指令输入为 ON 后的第 2 次扫描开始执行的,因此使用 PLS M10 的触点。

4、注意事项

（1）HSCS、HSCR、HSZ 与普通指令一样可以多次使用，但这些指令同时驱动的个数限制在每一路高速输入在 6 条指令之内，4 路高速输入最大 24 条。但是 HSZ 指令特殊模式（高速表格比较模式、频率控制模式）在梯形图里仅能同时驱动 1 个指令，也不影响其他路的 6 条。

（2）由于在执行 END 时，表格才会制作完成，所以特殊模式下的动作是在第二个周期开始动作。高速表格模式下，第二个扫描周期才开始表格比较置位或复位输出；频率控制模式下，第二个扫描周期 PLSY 才根据频率表格控制输出端子输出对应的频率。

（3）频率控制模式下，编程中使用 PLSY 指令以及 PLSR 指令无法同时得到 2 路脉冲输出。

（4）普通模式下，即使高速计数器没有打开也会进行区间比较置位对应软元件，特殊模式下不会进行比较置位。

（5）相关的特殊寄存器

SM8130	HSZ 高速表格比较模式
SM8131	高速表格比较执行结束标志
SM8132	HSZ 频率控制模式
SM8133	频率表格输出执行结束标志
SD8130	HSZ 高速表格比较模式下的表格计数器
SD8131	HSZ 频率控制模式下的表格计数器
SD8132	HSZ、PLSY 频率控制模式下频率低位
SD8133	HSZ、PLSY 频率控制模式下频率高位
SD8134	HSZ、PLSY 频率控制模式下目标脉冲数低位
SD8135	HSZ、PLSY 频率控制模式下目标脉冲数高位

2.7.6 SPD/脉冲密度指令

采用中断输入方式对指定时间内的输入脉冲进行计数的指令。（有些型号不支持）

1、指令格式

FNC56	指令所占程序步	指令分类	指令格式
16 位	7	SPD	(D)SPD S1 S2 D
		\	
32 位	13	DSPD	
		\	

操作数说明

操作数	内容	类型
S1	输入(X)脉冲的软元件编号	位
S2	时间(ms)数据或是保存数据的字软元件编号	BIN16/32 位
D	保存脉冲密度数据的起始字软元件编号	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1	*																						
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*	*	*	*					

2、功能和动作说明

（1）16 位指令

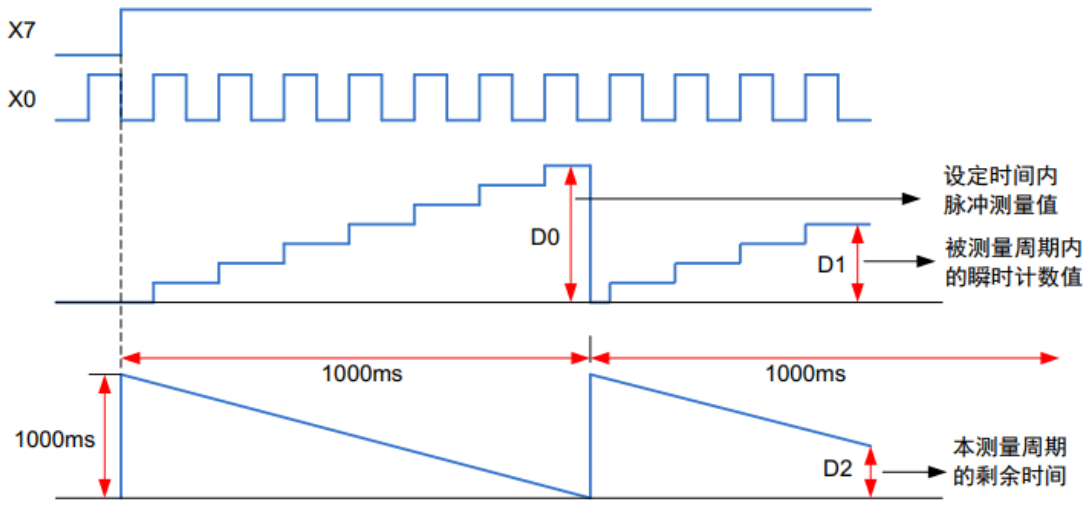
只在 S2×1ms 时间内对输入 S1 的脉冲进行计数，测定值保存到 D，当前值保存到 D+1，剩余时间保存到 D+2 中。重复这个操作，可以在测量值 D 中，得到脉冲密度(也就是与转速成比例的值)。

（2）32 位指令

只在（S2+1，S2）×1ms 时间内对输入 S1 的脉冲进行计数，测定值保存到（D+1，D），当前值保存到（D+3，D+2），剩余时间保存到（D+5，D+4）中。重复这个操作，可以在测量值（D+1，D）中，得到脉冲密度(也就是与转速成比例的值)。

3、程序举例





当 x7 置 ON 时，D1 对 x0 的 OFF->ON 动作计数，1000ms 后将其结果存入 D0，随之 D1 复位，再次对 X0 的动作计数。D2 用于测定剩余时间。，因此根据 D0 和 S2 的设定值可以求得脉冲频率。若脉冲信号取自旋转编码器，可求得转速。

4、注意事项

（1）指定的 S1 输入的 X0~X3 不能同时使用于高速计数器和输入中断。

（2）每个输入点只能使用一条该指令。

2.7.7 PLSY/脉冲输出指令

发出脉冲信号用的指令。

1、指令格式

FNC57	指令所占程序步	指令分类	指令格式
16 位	7	PLSY	-[(D)PLSY S1 S2 D]
		\	
32 位	13	DPLSY	
		\	

操作数说明

操作数	内容	类型
S1	率数据(Hz)或是保存数据的字软元件编号	BIN16/32 位
S2	脉冲量数据或是保存数据的字软元件编号	BIN16/32 位
D	输出脉冲的位软元件(Y)编号	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*																					

2、功能和动作说明

（1）16 位指令

从输出 Y 中输出 S2 个频率为 S1 的脉冲串。

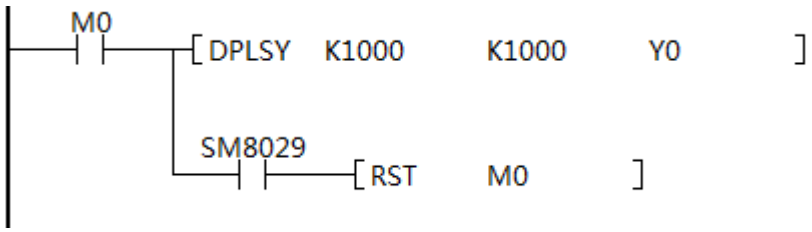
- S1：频率，范围 1~32767（HZ）。
- S2：脉冲量，范围 1~32767(PLS)；当 S2 设定为 0 时，可按 S1 的频率无限输出脉冲。
- D ：脉冲输出的 Y 编号，范围 Y0~Y3。

（2）32 位指令

从输出 Y 中输出（S2+1，S2）个频率为（S1+1，S1）的脉冲串。

- （S1+1，S1）：频率，范围 1~60000（HZ）。（不同型号 PLC 的最大频率不同）
- （S2+1，S2）：脉冲量，范围 1~2147483647(PLS)；当 S2 设定为 0 时，可按 S1 的频率无限输出脉冲。
- D ：脉冲输出的 Y 编号，范围 Y0~Y3。

3、程序举例



输出 1000 个频率为 1000HZ 的脉冲串，输出完毕后 M8029 指令结束标志动作。

4、注意事项

（1）指令执行结束标志位和异常结束标志位

使用了标志位变化的其他指令和多个 PLSY/PLSR 指令，请务必在监视指令的正下方使用，即 SM8029 或 SM8329 要与指令并联。

（2）输出脉冲数当前值监控

软元件		内容	指令名称
高位	低位		
SD8141	SD8140	Y000 输出脉冲计数	PLSY 和 PLSR 指令从 Y000 输出的脉冲数计数
SD8143	SD8142	Y001 输出脉冲计数	PLSY 和 PLSR 指令从 Y001 输出的脉冲数计数

（3）脉冲输出状态监控

SM8340	Y0 脉冲输出中监控标志位（输出中该标志位置位）
SM8350	Y1 脉冲输出中监控标志位（输出中该标志位置位）

（4）停止脉冲输出

SM8349	停止 Y000 脉冲输出（即刻停止）
SM8359	停止 Y001 脉冲输出（即刻停止）

（5）PLSY 指令执行过程中：

S1 的数据被更改时，输出频率随之改变；

S2 被更改时，从指令下一次被驱动开始变更内容生效。

（6）指定了同一输出端口的脉冲输出指令和定位指令不能被同时驱动，否则报 6764 错误。

2.7.8 PWM/脉宽调制指令

指定了脉冲的周期和 ON 时间的脉冲输出的指令。

1、指令格式

FNC58	指令所占程序步	指令分类	指令格式
16 位	7	PWM \	[PWM S1 S2 D]

操作数说明

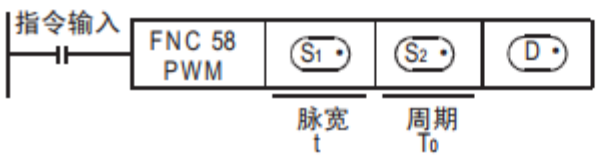
操作数	内容	类型
S1	脉宽(ms)数据或是保存数据的字软元件编号	BIN16 位
S2	周期(ms)数据或是保存数据的字软元件编号	BIN16 位
D	输出脉冲的位软元件(Y)编号	位

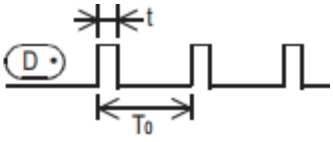
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*																					

2、功能和动作说明

以周期 S2(ms)单位输出 ON 脉冲宽度 S1(ms)的脉冲。





- S1：脉宽，范围 0~32767ms。
- S2：周期，1~32767ms。
- D：输出脉冲编号 Y0~Y3。

3、程序举例

略。

4、注意事项

设定脉宽和周期时，S1<=S2。

2.7.9 PLSR/带加减速的脉冲输出指令

指定了脉冲的周期和 ON 时间的脉冲输出的指令。

1、指令格式

FNC59	指令所占程序步	指令分类	指令格式
16 位	9	PLSR	-[(D)PLSR S1 S2 S3 D]
		\	
32 位	17	DPLSR	
		\	

操作数说明

操作数	内容	类型
S1	保存最高频率(Hz)数据，或是数据的字软元件编号	BIN16/32 位
S2	保存总的脉冲数(PLS)数据，或是数据的字软元件编号	BIN16/32 位
S3	保存加减速时间(ms)数据，或是数据的字软元件编号	BIN16/32 位
D	输出脉冲的位软元件(Y)编号	位

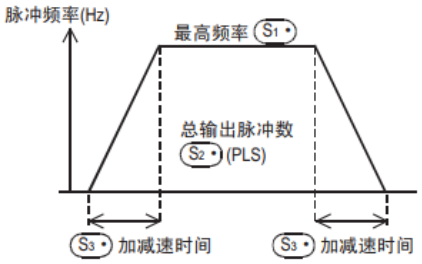
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S3								*	*	*	*	*	*	*	*	*	*	*	*	*			
D		*																					

2、功能和动作说明

（1）16 位指令

从输出 Y 输出脉冲，其最高频率 S1，执行 S3(ms)时间的加减速，输出脉冲数仅为 S2。



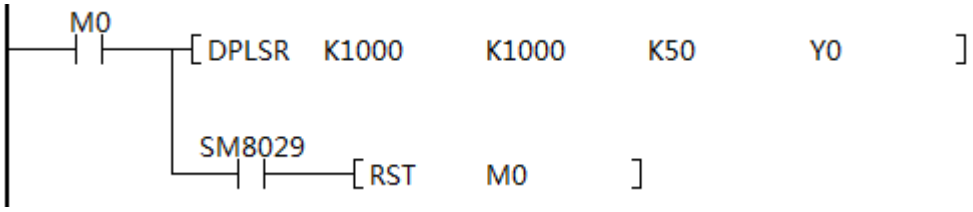
- S1：最高频率，范围 16~32767（HZ）。
- S2：总输出脉冲数，范围 1~32767（PLS）。
- S3：加减速时间，范围 50~5000（ms）。
- D：输出脉冲编号 Y0~Y3。

（2）32 位指令

从输出 Y 输出脉冲，其最高频率（S1+1，S1），执行（S3+1，S3）(ms)时间的加减速，输出脉冲数仅为（S2+1，S2）。

- （S1+1，S1）：最高频率，范围 16~60000（HZ）。（不同型号 PLC 的最大频率不同）
- （S2+1，S2）：总输出脉冲数，范围 1~2147483647（PLS）。
- （S3+1，S3）：加减速时间，范围 50~5000（ms）。
- D：输出脉冲编号 Y0~Y3。

3、程序举例



加减速时间为 50ms，输出 1000 个频率为 1000HZ 的脉冲串，输出完毕后 SM8029 指令结束标志动作。

4、注意事项

（1）指令执行结束标志位和异常结束标志位

使用了标志位变化的其他指令和多个 PLSY/PLSR 指令，请务必在监视指令的正下方使用，即 SM8029 或 SM8329 要与指令并联。

（2）输出脉冲数当前值监控

软元件		内容	指令名称
高位	低位		
SD8141	SD8140	Y000 输出脉冲计数	PLSY 和 PLSR 指令从 Y000 输出的脉冲数计数
SD8143	SD8142	Y001 输出脉冲计数	PLSY 和 PLSR 指令从 Y001 输出的脉冲数计数

（3）脉冲输出状态监控

SM8340	Y0 脉冲输出中监控标志位（输出中该标志位置位）
SM8350	Y1 脉冲输出中监控标志位（输出中该标志位置位）

（4）脉冲输出停止标志

SM8349	停止 Y000 脉冲输出（即刻停止）
SM8359	停止 Y001 脉冲输出（即刻停止）

（5）PLSR 指令执行过程中：S1、S2、S3 的值在脉冲输出期间被改变时，只有在下一次指令被驱动才会生效。

（6）指定了同一输出端口的脉冲输出指令和定位指令不能被同时驱动，否则报 6764 错误。

2.8 通信指令

两条通信指令用于方便组建特殊的通信网络，分别是 RS2（串行数据通信指令）、ADPRW（MODBUS 主站通信用指令）

2.8.1 RS2/串行数据传送 2 指令

参考 3.4.2 章节。

2.8.2 ADPRW/MODBUS 通信主站指令

参考 3.4.3 章节。

2.9 浮点运算指令

用于对精度要求比较高的浮点数数据分析计算，包含基本的浮点数四则运算、浮点数转换、三角函数等。

2.9.1 ECMP/2 进制浮点数比较指令

比较 2 个浮点数数据值，将结果(大于、等于或小于)输出到位软元件(3 点)中的指令。

1、指令格式

FNC110	指令所占程序步	指令分类	指令格式
32 位	13	DECMP	┤[DECMP(P) S1 S2 D]├
		DECMPP	

操作数说明

操作数	内容	类型
S1	比较值 1	实数(2 进制)
S2	比较值 2	实数(2 进制)
D	比较结果	位

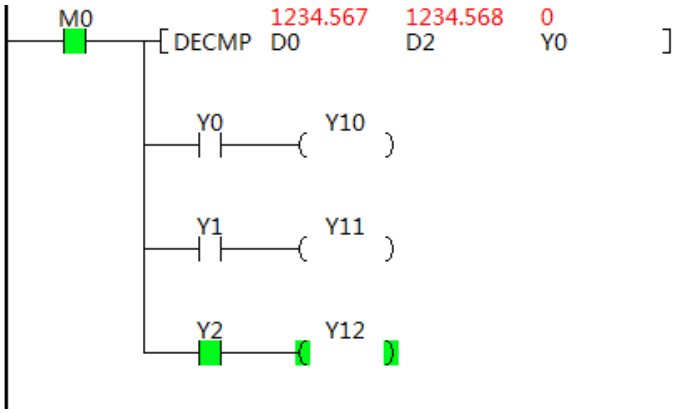
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
D		*	*			*	*											*					

2、功能和动作说明

32 位运算指令，将比较值 S1 和比较源 S2 作为浮点数数据进行比较，然后根据比较的结果将 D、D+1、D+2 中的任意一位置 ON。S1、S2 中指定了常数(K、H)时，会自动将数值从 BIN 转换成 2 进制浮点数后再处理。

3、程序举例



指令执行时，D0 代表的 32 位浮点数小于 D2，因此 Y2 置位，输出 Y12；

如果 D0 = D2，Y1 置位，输出 Y11；

如果 D0 > D2，Y0 置位，输出 Y10。

4、注意事项

（1）D 占用 3 点，（D，D+1，D+2），注意不要与其他用途的软元件重复。

（2）即使指令输入 OFF，不执行 DECMP 指令，D～D+2 也能保持指令输入 OFF 之前的状态。

2.9.2 EZCP/2 进制浮点数区间比较指令

将 2 个区间的浮点数和浮点数比较源进行比较，根据其结果输出到位软元件(3 点)中的指令。

1、指令格式

FNC111	指令所占程序步	指令分类	指令格式
32 位	17	DEZCP DEZCPP	DEZCP(P) S1 S2 S D

操作数说明

操作数	内容	类型
S1	左区间浮点数	实数(2 进制)
S2	右区间浮点数	实数(2 进制)
S	浮点数比较源	实数(2 进制)
D	输出结果的起始位软元件编号(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
S														*	*			*	*	*	*		
D		*	*			*	*											*					

2、功能和动作说明

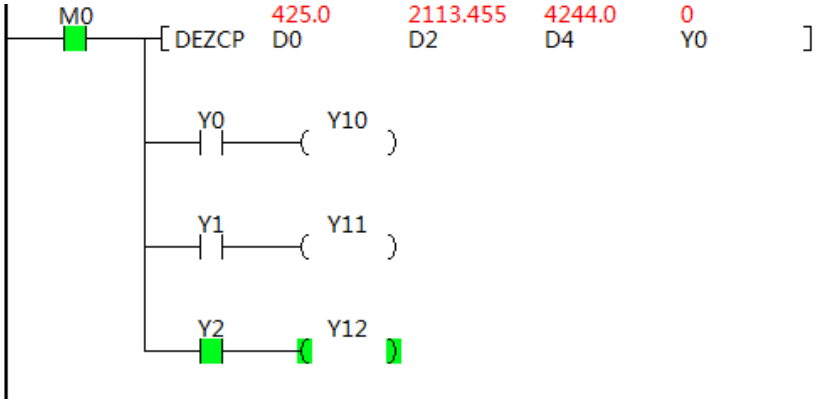
32 位运算指令，将比较值 S1、S2 和比较源 S 作为浮点数数据进行比较。

当 S1 ＞ S 时 D 置位；

当 S1 ≤ S ≤ S2 时 D+1 置位；

当 S2 ＜ S 时 D+2 置位。

3、程序举例



指令执行时，比较源 D4 大于右区间浮点数，因此 Y2 置位，输出 Y12；

如果 D0 ＞ D4，Y0 置位，输出 Y10；

如果 D0 ≤ D4 ≤ D2 ，Y1 置位，输出 Y11。

4、注意事项

（1）S1 ＞ S2 时，S2 的数值视为与 S1 相同，从而进行比较。

（2）即使指令输入 OFF，不执行 DEZCP 指令，D～D+2 也能保持指令输入 OFF 之前的状态。

2.9.3 EMOV/2 进制浮点数传送指令

传送 2 进制浮点数数据的指令。

1、指令格式

FNC112	指令所占程序步	指令分类	指令格式
32 位	9	DEMOV	[DEMOV(P) S D]
		DEMOVP	

操作数说明

操作数	内容	类型
S	传送源	实数(2 进制)
D	目标数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将传送源 S 的内容(2 进制浮点数数据)传送到 D 中。此外还可以在 S 中直接指定实数(E)。

3、程序举例

M0 为 ON 时，将实数-123.456 保存到 D0 代表的 32 位浮点数中，（D0，D1） = 1.23456



4、注意事项

注意与普通的 MOV 指令的区分。

2.9.4 ESTR/2 进制浮点数→字符串的转换指令

该指令用于将 2 进制浮点数数据转换成指定位数的字符串(ASCII 码)。

1、指令格式

FNC116	指令所占程序步	指令分类	指令格式
32 位	13	DESTR DESTRP	[DESTR(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	传送源（2 进制浮点数）	实数(2 进制)
S2	指定转换形式的软元件起始编号	BIN16 位
D	目标数据（存储转换后的字符串）	字符串

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*			*		
S2								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

32 位运算指令，根据 S2,S2+1,S2+2 中指定的内容，将（S1+1，S1）的内容(2 进制浮点数数据）转换成字符串，并保存至 D 开始的软元件中。

其中 S2 表示转换形式，转换后的数据结果与 S2 的指定显示有关。S2 为 0 时，表示小数点形式转换，为 1 时表示指数形式转换；

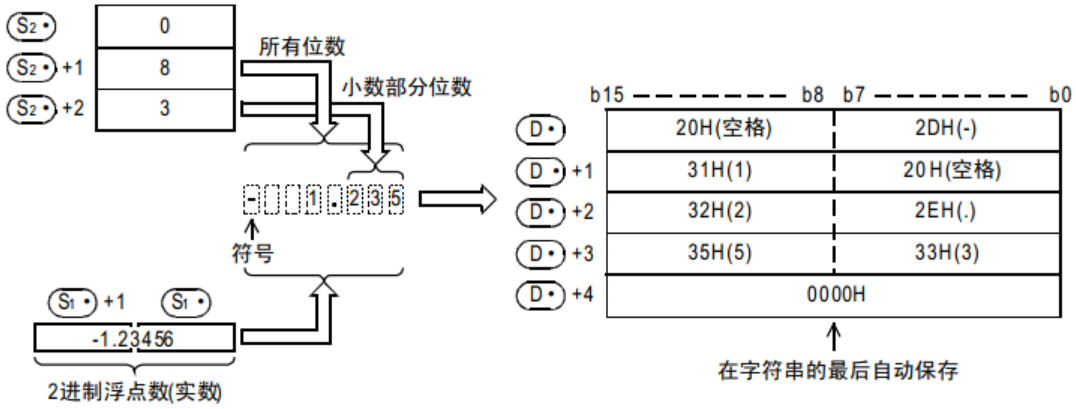
S2+1 表示所有位数；

S2+2 表示小数部分位数。

3、程序举例

（1）小数点形式

例如，所有位数为 8，小数部分位数为 3，指定-1.23456 时，D 开始的软元件中会如下所示被保存。



• 字符中，2 进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。

• 小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五入。

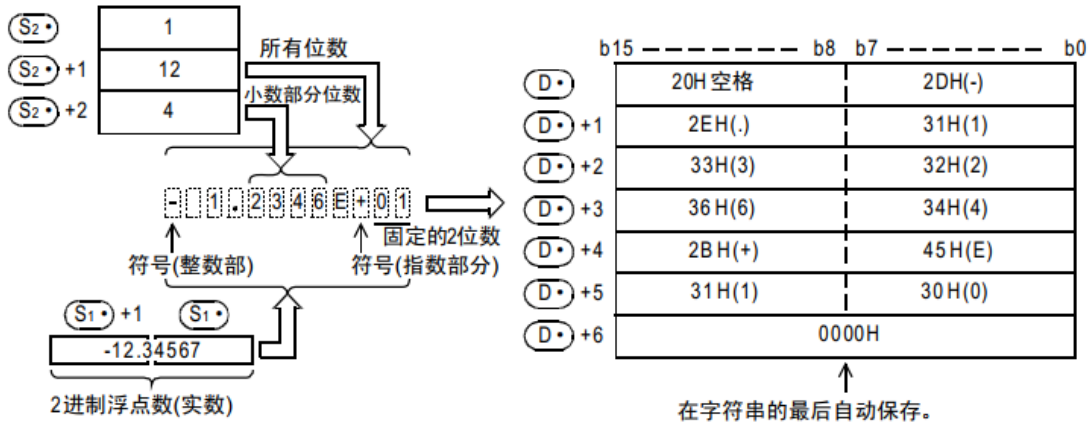
• 小数部分位数设定为“0”以外数字时，会自动将“2EH”(.)保存到指定的小数部分位数+1 的位中。但是，小数部分位数为“0”时，不保存“2EH”(.)。

• 当从所有位数中去除符号、小数点、小数部分的位数后，其位数比 2 进制浮点数数据的整数部分大的时候，在符号和整数部分之间保存“20H”(空格)。

• 在转换的字符串最后，自动保存“00H”或“0000H”。

（2）指数形式

例如，所有位数为 12，小数部分位数为 4，指定-12.34567 时，在 D 开始的软元件中会如下所示被保存。



• 在整数部分的符号中，2 进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。

• 整数部分固定为 1 位数。在整数部分和符号之间保存“20H”(空格)。

• 在小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五入。

• 小数部分位数设定为“0”以外数字时，会自动将“2EH”(.)保存到指定的小数部分位数+1 的位中。但是，小数部分位数为“0”时，不保存“2EH”(.)。

• 在指数部分的符号中，指数为正时保存“2BH”(+)，指数为负时保存“2DH”(-)。

• 指数部分固定为 2 位数。当指数部分为 1 位数时，在其与指数部分的符号之间中保存“30H”(0)。

• 在转换的字符串最后，自动保存“00H”或“0000H”。

4、注意事项

以下情况会出现运算错误：

（1）S1 不在范围 0，$\pm 2^{-126} \leqslant S1 < \pm 2^{128}$ 中时。

（2）S2 中指定的形式为 0、1 以外时。

（3）S2+1 中指定的所有位数指定不符合下面的范围时

小数点形式：

小数部分的位数为“0”时，所有位数≧2；

小数部分的位数为“0”以外的数字时，所有位数≧(小数部分位数+3)。

指数形式：

小数部分的位数为“0”时，所有位数≧6；

小数部分的位数为“0”以外的数字时，所有位数≧(小数部分位数+7)。

（4）S2+2 中指定的小数部分的位数指定不符合下面范围时

小数点形式时：小数点位数≦(所有位数-3)。

指数形式时：小数点位数≦(所有位数-7)。

（5）当保存指定字符串的软元件的范围，超出相应软元件的范围时。

（6）转换结果超出已指定的所有位数时。

2.9.5 EVAL/字符串→2 进制浮点数的转换指令

用于将字符串(ASCII 码)转换成 2 进制浮点数数据的指令。

1、指令格式

FNC117	指令所占程序步	指令分类	指令格式
32 位	9	DEVAL	-[DEVAL(P) S D]
		DEVALP	

操作数说明

操作数	内容	类型
S	传送源（字符串数据）	字符串
D	目标数据（2 进制浮点数数据）	实数(2 进制)

操作数可用元件类型

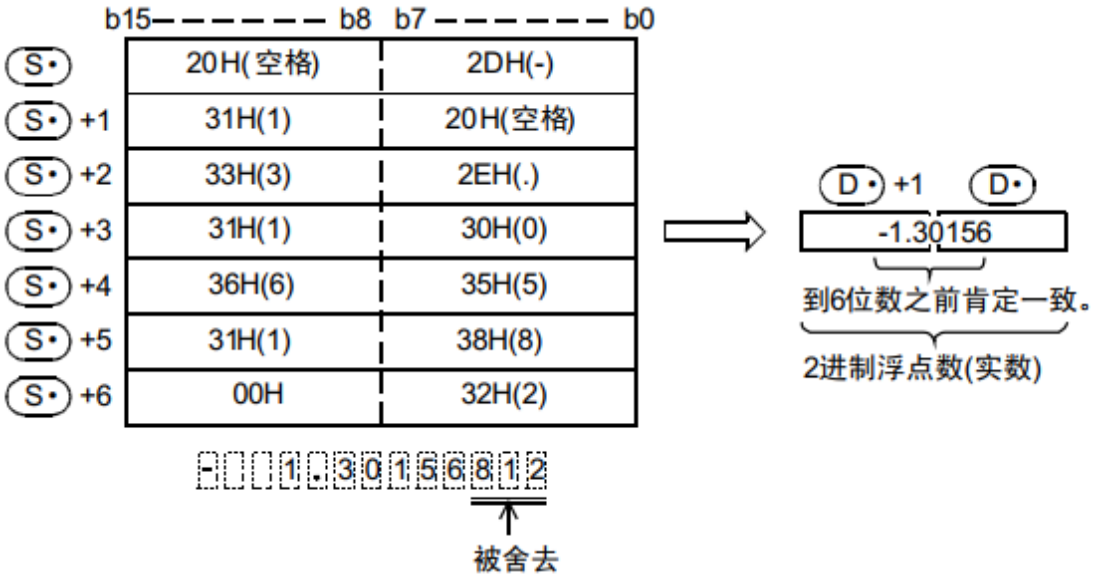
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D														*	*			*					

2、功能和动作说明

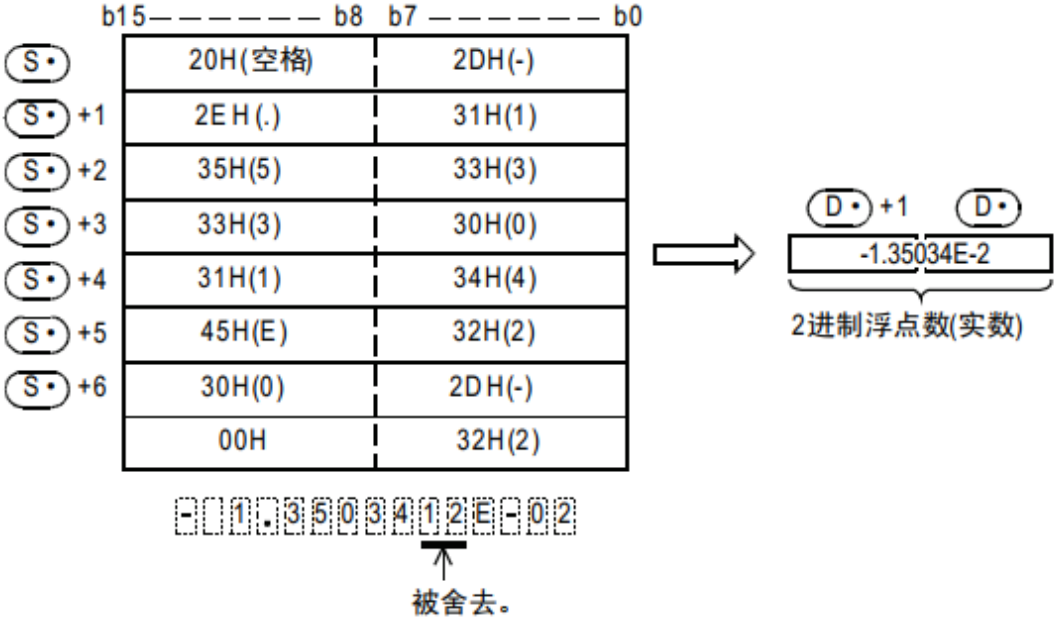
32 位运算指令，将 S 开始的软元件中保存的字符串转换成 2 进制浮点数数据后，保存到(D+1，D）中。指定的字符串无论是小数点形式，还是指数形式，都可以转换成 2 进制浮点数数据 。

3、程序举例

（1）小数点形式



（2）指数形式



• S 中指定的，要转换成 2 进制浮点数的字符串，去除了符号、小数点、指数部分仍然有 7 位数以上时，舍去第 7 位数以后的数。

• 在小数点形式中，将符号指定为“2BH”(+)或省略符号，则作为正值转换。此外，将符号指定为"2DH"(-)则作为负的值转换。

• 用指数形式在指数部分的符号内指定"2BH"(+)，省略符号后作为正的指数转换。将指数部分的符号指定为"2DH"(-)则作为负的指数转换。

• 指定的字符串中，在最初的“0”以外的数值之间如果存在“20H”(空格)或是“30H”(0)时，会忽略“20H”、“30H”而进行转换。

• 在指数形式的字符串中，“E”和数值之间如果存在“30H”(0)，则忽略“30H”而进行转换。

•字符串最大可以设定到 24 个字符。字符串中的“20H”(空格)、“30H”(0)也作为一个字符来计算。

4、注意事项

（1）相关软元件

软元件	名称	内容	
		条件	动作
M8020	零位	转换结果真的为零 (尾数部分为“0”时)	零位标志位(M8020)为 ON。
M8021	借位	转换结果的绝对值＜2^-126	D 的值小于 32 位实数的最小值(2^-126) 部分被舍去，借位 标志位(M8021)为 ON。
M8022	进位	转换结果的绝对值≧2^128	的值大于 32 位实数的最大值(2^128)部 分被舍去，进位 标志位(M8022)为 ON。

（2）以下情况会出现运算错误，报错 6706

- 整数部分、小数部分中存在“30H”(0)～“39H”(9)以外的字符时。
- S 指定的字符串中存在 2 个以上的“2EH”(.)时。
- 指数部分中有“45H”(E)、“2BH”(+)、“2DH”(-)以外的字符或是有多个指数部分时。
- S 开始的相应软元件范围内没有“00H”时。
- S 以后的字符数为 0 或是超出了 24 个字符时。

2.9.6 EBCD/2 进制浮点数→10 进制浮点数的转换指令

将软元件中的 2 进制浮点数转换成→10 进制浮点数的指令。

1、指令格式

FNC118	指令所占程序步	指令分类	指令格式
32 位	9	DEBCD	-[DEBCD(P) S D]
		DEBCDP	

操作数说明

操作数	内容	类型

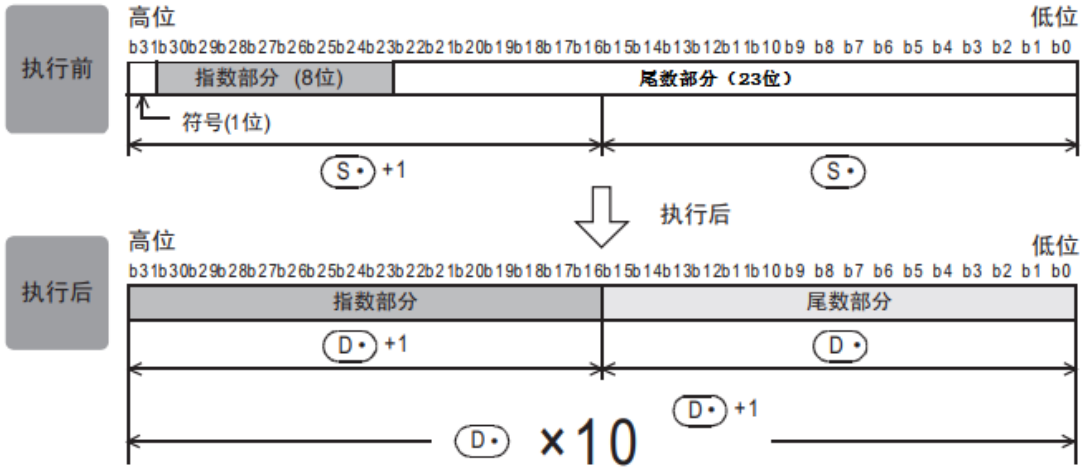
S	源数据（2 进制数据）	实数(2 进制)
D	目标数据（10 进制浮点数数据）	实数(10 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 2 进制浮点数转换成 10 进制浮点数后，保存到（D+1，D）中。



3、程序举例

略。

4、注意事项

在浮点数运算中，都以 2 进制浮点数执行。但是，由于 2 进制浮点数本身是不易理解的数值(专用的监控方法)，所以转换成 10 进制浮点数运算后，可以便于在外围设备上进行监控等。

2.9.7 EBIN/10 进制浮点数→2 进制浮点数的转换指令

将软元件中的 10 进制浮点数转换成 2 进制浮点数的指令。

1、指令格式

FNC119	指令所占程序步	指令分类	指令格式
32 位	9	DEBIN	-[DEBIN(P) S D]
		DEBINP	

操作数说明

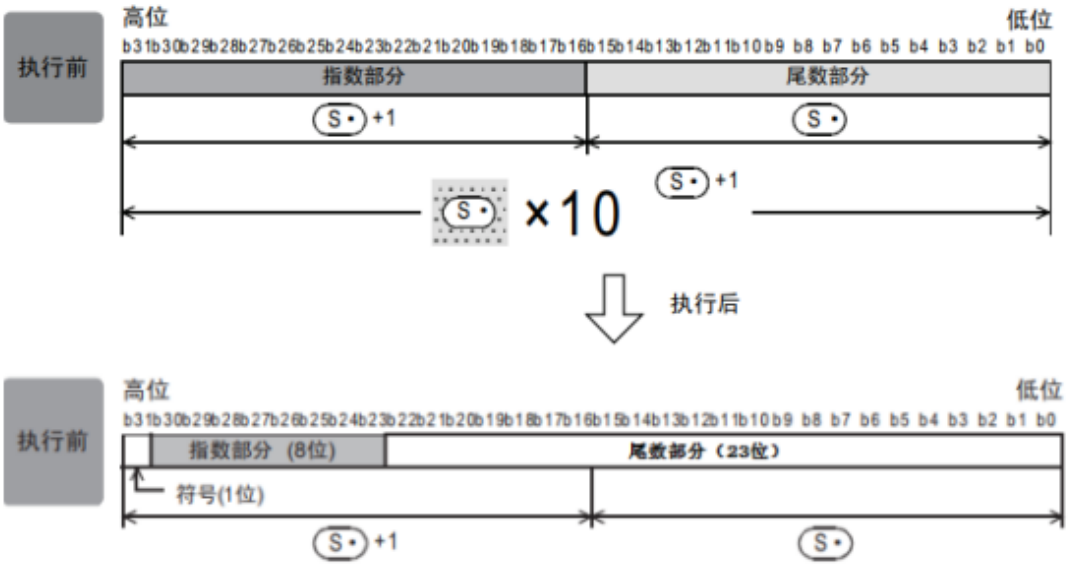
操作数	内容	类型
S	保存 10 进制浮点数数据的数据寄存器编号	实数(10 进制)
D	保存被转换的 2 进制浮点数数据的数据寄存器编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 10 进制浮点数转换成 2 进制浮点数后，保存到（D+1，D）中。



3、程序举例

略。

4、注意事项

注意 2 进制浮点数的存储方式与 10 进制的区别。

2.9.8 EADD/2 进制浮点数加法运算指令

2 个 2 进制浮点数加法运算的指令。

1、指令格式

FNC120	指令所占程序步	指令分类	指令格式
32 位	\	DEADD DEADDP	[DEADD(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	保存进行加法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行加法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存加法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将（S1+1，S1）和（S2+1，S2)的 2 进制浮点数数据进行加法运算，并将其运算结果以 2 进制浮点数形式传送到(D+1，D)中。在(S1+1，S1)和(S2+1，S2)中指定了常数(K、H)时，数值会自动被转换成 2 进制浮点数。

3、程序举例

略。

4、注意事项

无。

2.9.9 ESUB/2 进制浮点数减法运算指令

2 个 2 进制浮点数减法运算的指令。

1、指令格式

FNC121	指令所占程序步	指令分类	指令格式
32 位	13	DESUB DESUBP	-[DESUB(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	保存进行减法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行减法的 2 进制浮点数数据的字软元件编	实数(2 进制)

	号	
D	保存减法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将（S1+1，S1）减去（S2+1，S2）的 2 进制浮点数数据，并将其运算结果以 2 进制浮点数形式传送到（D+1，D）中。在（S1+1，S1）和（S2+1，S2）中指定了常数(K、H)时，数值会自动被转换成 2 进制浮点数。

3、程序举例

略。

4、注意事项

无。

2.9.10 EMUL/2 进制浮点数乘法运算指令

2 个 2 进制浮点数乘法运算的指令。

1、指令格式

FNC122	指令所占程序步	指令分类	指令格式
32 位	13	DEMUL	-[DEMUL(P) S1 S2 D]
		DEMULP	

操作数说明

操作数	内容	类型
S1	保存进行乘法运算的2进制浮点数数据的字软元	实数(2 进制)

	件编号	
S2	保存进行乘法的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存乘法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将（S1+1，S1）和（S2+1，S2）的 2 进制浮点数数据相乘，并将其运算结果以 2 进制浮点数形式传送到（D+1，D）中。在（S1+1，S1）和（S2+1，S2）中指定了常数(K、H)时，数值会自动被转换成 2 进制浮点数。

3、程序举例

略。

4、注意事项

运算结果超过单精度浮点数范围时，对应的标志位置位

软元件	名称	内容	
		条件	动作
SM8020	零位	转换结果真的为零 (尾数部分为“0”时)	零位标志位(SM8020)为 ON。
SM8021	借位	转换结果的绝对值＜2^-126	D 的值小于 32 位实数的最小值(2^-126) 部分被舍去，借位 标志位(SM8021)为 ON。
SM8022	进位	转换结果的绝对值≧2^128	的值大于 32 位实数的最大值(2^128)部 分被舍去，进位 标志位(SM8022)为 ON。

2.9.11 EDIV/2 进制浮点数除法运算指令

2 个 2 进制浮点数除法运算的指令。

1、指令格式

FNC123	指令所占程序步	指令分类	指令格式
32 位	9	DEDIV	[DEDIV(P) S1 S2 D]
		DEDIVP	

操作数说明

操作数	内容	类型
S1	保存进行除法运算的 2 进制浮点数数据的字软元件编号	实数(2 进制)
S2	保存进行除法的 2 进制浮点数数据的字软元件编号	实数(2 进制)
D	保存除法运算后的 2 进制浮点数数据	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1														*	*			*	*	*	*		
S2														*	*			*	*	*	*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，将（S1+1，S1）和（S2+1，S2）的 2 进制浮点数数据相除，并将其运算结果以 2 进制浮点数形式传送到（D+1，D）中。在（S1+1，S1）和（S2+1，S2）中指定了常数(K、H)时，数值会自动被转换成 2 进制浮点数。

3、程序举例

略。

4、注意事项

运算结果超过单精度浮点数范围时，对应的标志位 SM8020，SM8021，SM8022 的动作。

软元件	名称	内容	
		条件	动作
SM8020	零位	转换结果真的为零 (尾数部分为“0”时)	零位标志位(SM8020)为 ON。
SM8021	借位	转换结果的绝对值＜2^-126	D 的值小于 32 位实数的最小值(2^-126) 部分被舍去，借位 标志位(SM8021)为 ON。
SM8022	进位	转换结果的绝对值≧2^128	的值大于 32 位实数的最大值(2^128)部 分被舍去，进位 标志位(SM8022)为 ON。

2.9.12 EXP/2 进制浮点数指数运算指令

该指令是以 e(2.71828)为底的指数运算指令。

1、指令格式

FNC124	指令所占程序步	指令分类	指令格式
32 位	9	DEXP	-[DEXP(P) S D]
		DEXPP	

操作数说明

操作数	内容	类型
S	保存执行指数运算的 2 进制浮点数数据的软元件 起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，以（S+1，S1）为指数做运算，并将其运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

$e^{(S+1,S)}$ = (D+1,D)

3、程序举例

略。

4、注意事项

结果不在 $2^{-126} \leqq |$ 运算结果 $| < 2^{128}$ 范围内，报运算错误 6706。

2.9.13 LOGE/2 进制浮点数自然对数运算指令

该指令执行自然对数运算。

1、指令格式

FNC125	指令所占程序步	指令分类	指令格式
32 位	9	DLOGE	-[DLOGE(P) S D]
		DLOGEP	

操作数说明

操作数	内容	类型
S	保存进行自然对数运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行（S+1，S）的自然对数（以 e(2.17828) 为底时的对数） 运算,并将其运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

3、程序举例

略。

4、注意事项

（S+1，S）中指定的值只可以设定正数，指定为负数或 0 时报错 6706。

2.9.14 LOG10/2 进制浮点数常用对数运算指令

该指令执行常用对数运算。

1、指令格式

FNC126	指令所占程序步	指令分类	指令格式
32 位	9	DLOG10	-[DLOG10(P) S D]
		DLOG10P	

操作数说明

操作数	内容	类型
S	保存进行常用对数运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行（S+1，S）的常用对数(10 为底时的对数)运算,并将其运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

3、程序举例

略。

4、注意事项

（S+1，S）中指定的值只可以设定正数，指定为负数或 0 时报错 6706。

2.9.15 ESQR/2 进制浮点数开方运算指令

2 进制浮点数开方(开根号)运算的指令。

1、指令格式见

FNC127	指令所占程序步	指令分类	指令格式
32 位	9	DESQR	-[DESQR(P) S D]
		DESQRP	

操作数说明

操作数	内容	类型
S	保存执行开方运算的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，执行（S+1，S）进行开方(开根号)运算(2 进制浮点数运算)后,将其运算结果保存到（D+1，D）中。

3、程序举例

略。

4、注意事项

（1）（S+1，S）值为正数有效，为负数时报错 6706。

（2）运算结果为 0 时 SM8020 置位。

软元件	名称	内容
SM8020	零位	运算结果真为 0 时 ON

2.9.16 ENEG/2 进制浮点数符号翻转指令

使 2 进制浮点数(实数)数据的符号翻转的指令。

1、指令格式

FNC128	指令所占程序步	指令分类	指令格式
32 位	5	DENEG	[DENEG(P) D]
		DENEGP	

操作数说明

操作数	内容	类型
D	保存要执行符号翻转的 2 进制浮点数数据的软元件的起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D														*	*			*					

2、功能和动作说明

32 位运算指令，（D+1，D）的 2 进制浮点数数据的符号翻转，保存在（D+1，D）中。

3、程序举例

略。

4、注意事项

无。

2.9.17 INT/2 进制浮点数→BIN 整数的转换指令

将 2 进制浮点数，转换成可编程控制器中的一般数据形式 BIN 整数的指令(2 进制浮点数→BIN 整数)。

1、指令格式

FNC129	指令所占程序步	指令分类	指令格式
16 位	5	INT	-[(D)INT(P) S D]
		INTP	
32 位	9	DINT	
		DINTP	

操作数说明

操作数	内容	类型
S	保存要转换成 BIN 整数的 2 进制浮点数数据的数据寄存器编号	实数(2 进制)
D	保存转换后的 BIN 整数的数据寄存器编号	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
D														*	*			*					

2、功能和动作说明

S 的 2 进制浮点数转换成 BIN 整数后，传送到（D+1，D）中。

3、程序举例

略。

4、注意事项

（1）相关标志位的动作

软元件	名称	内容
SM8020	零位	运算结果真为 0 时 ON。
SM8021	借位	位转换时，有因不满 1 而被舍去的情况发生时，置 ON。
SM8022	进位	运算结果超出-32,768～32,767(16 位运算时)，或是-2,147,483,583～2,147,483,583(32 位运算时)的范围而产生溢出时为 ON。(运算结果不反映。)

（2）小数点以后的值被舍去

2.9.18 SIN/2 进制浮点数 SIN 运算指令

求角度(RAD)的 SIN 值的指令。

1、指令格式

FNC130	指令所占程序步	指令分类	指令格式
32 位	9	DSIN	-[DSIN(P) S D]
		DSINP	

操作数说明

操作数	内容	类型
S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 SIN 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,将（S+1，S）中指定的角度值(2 进制浮点数)转换成 SIN 值后，传送到（D+1，D）中。

3、程序举例

略。

4、注意事项

运算结果为 0 时，SM8020 置位。

RAD 值 = 角度 *π/180。

2.9.19 COS/2 进制浮点数 COS 运算指令

求角度(RAD)的 COS 值的指令。

1、指令格式

FNC131	指令所占程序步	指令分类	指令格式
32 位	9	DCOS	-[DCOS(P) S D]
		DCOSP	

操作数说明

操作数	内容	类型

S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 COS 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,将（S+1，S）中指定的角度值(2 进制浮点数)转换成 COS 值后，传送到（D+1，D）中。

3、程序举例

略。

4、注意事项

RAD 值 = 角度 *π/180。

2.9.20 TAN/2 进制浮点数 TAN 运算指令

求角度(RAD)的 TAN 值的指令。

1、指令格式

FNC132	指令所占程序步	指令分类	指令格式
32 位	9	DTAN	-[DTAN(P) S D]
		DTANP	

操作数说明

操作数	内容	类型
S	保存 2 进制浮点数的 RAD(角度)的软元件编号	实数(2 进制)
D	保存 2 进制浮点数的 TAN 值的软元件编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令,将（S+1，S）中指定的角度值(2 进制浮点数)转换成 TAN 值后，传送到（D+1，D）中。

3、程序举例

略。

4、注意事项

运算结果为 0 时，M8020 置位。

RAD 值 = 角度 *π/180。

2.9.21 ASIN/2 进制浮点数 SIN-1 运算指令

执行 SIN^{-1} 运算的指令。

1、指令格式

FNC133	指令所占程序步	指令分类	指令格式
32 位	9	DASIN	[DASIN(P) S D]
		DASINP	

操作数说明

操作数	内容	类型
S	保存执行 SIN^{-1}(反正弦)运算的 SIN 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 SIN 值求出角度，将运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

3、程序举例

略。

4、注意事项

（1）（S+1，S）的 SIN 值，设定在-1.0~1.0 范围，不在此范围报错 6706。

（2）（D+1，D）中保存的角度是弧度值（-π/2）~（π/2）。

2.9.22 ACOS/2 进制浮点数 COS-1 运算指令

执行 COS^{-1} 运算的指令。

1、指令格式

FNC134	指令所占程序步	指令分类	指令格式
32 位	9	DACOS	[DACOS(P) S D]
		DACOSP	

操作数说明

操作数	内容	类型
S	保存执行 COS^{-1}(反余弦)运算的 COS 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 COS 值求出角度，将运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

3、程序举例

略。

4、注意事项

（1）（S+1，S）的 COS 值，设定在-1.0~1.0 范围，不在此范围报错 6706。

（2）（D+1，D）中保存的角度是弧度值 0~π。

2.9.23 ATAN/2 进制浮点数 TAN-1 运算指令

执行 TAN^{-1} 运算的指令。

1、指令格式

FNC135	指令所占程序步	指令分类	指令格式
32 位	9	DATAN	-[DATAN(P) S D]
		DATANP	

操作数说明

操作数	内容	类型
S	保存执行 TAN^{-1}(反正切)运算的 TAN 值的软元件的起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的 TAN 值求出角度，将运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。

3、程序举例

略。

4、注意事项

（D+1，D）中保存的角度是弧度值>(-π/2)或<(π/2)

2.9.24 RAD/2 进制浮点角度→弧度的转换指令

将角度单位的值转换成弧度单位的指令。

1、指令格式

FNC136	指令所占程序步	指令分类	指令格式
32 位	9	DRAD	[DRAD(P) S D]
		DRADP	

操作数说明

操作数	内容	类型
S	保存要转换成弧度单位的角度的软元件起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的单位从角度单位转换成弧度单位 ，将运算结果保存到（D+1，D）中。此外，可以在 S 中直接指定实数。弧度单位 = 角度单位*π/180。

3、程序举例

略。

4、注意事项

无。

2.9.25 DEG/2 进制浮点弧度→角度的转换指令

将弧度单位的值转换成角度(DEG)单位的指令。

1、指令格式

FNC137	指令所占程序步	指令分类	指令格式
32 位	9	DDEG	-[DDEG(P) S D]
		DDEGP	

操作数说明

操作数	内容	类型
S	保存要转换成角度单位的弧度的软元件起始编号	实数(2 进制)
D	保存运算结果的软元件起始编号	实数(2 进制)

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*			*		
D														*	*			*					

2、功能和动作说明

32 位运算指令，（S+1，S）的单位从弧度单位转换成角度单位 ，将运算结果保存到（D+1，D）中。

3、程序举例

略。

4、注意事项

无。

2.10 数据处理 2 指令

2.10.1 WSUM/算出数据合计值指令

该指令可计算出连续的 16 位或是 32 位数据的合计值。

1、指令格式

FNC140	指令所占程序步	指令分类	指令格式
16 位	7	WSUM	(D)WSUM(P) S D n
		WSUMP	
32 位	13	DWSUM	
		DWSUMP	

操作数说明

操作数	内容	类型
S	保存要算出合计值的数据的软元件起始编号	BIN16/32 位

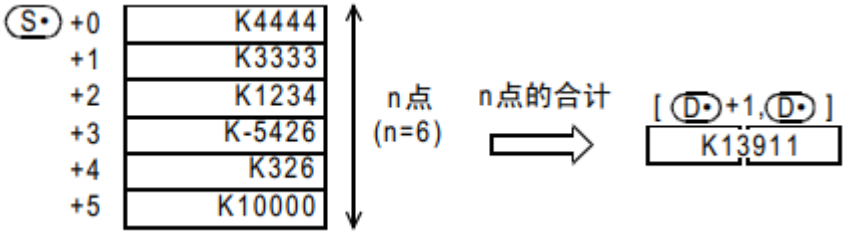
D	保存合计值的软元件起始编号	BIN32/64 位
n	数据个数(0＜n)	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

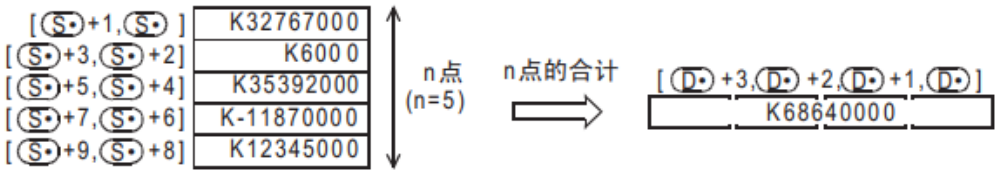
（1）16 位指令



将 S 开始的 n 点 16 位数据的合计值，以 32 位数据形式保存在（D+1，D）中。

（2）32 位指令

将（S+1，S）开始的 n 点 32 位数据的合计值，以 64 位数据形式保存在（D+3，D+2，D+1，D）中。



3、程序举例

略。

4、注意事项

（1）数据个数设定≤0 时，报错 6706。

（2）注意运算结果，16 位指令时，超过 16 位数的情况下，查看 D+1，D 分别是运算结果的高 16 位、低 16 位，未超过查看 D 即可；32 位指令时，超过的情况下，查看（D+3，D+2），（D+1，D）分别是运算结果的高 32 位、低 32 位，未超过时查看（D+1,D）即可。

2.10.2 WTOB/字节单位的数据分离指令

将连续的 16 位数据按照字节(8 位)单位进行分离的指令。

1、指令格式

FNC141	指令所占程序步	指令分类	指令格式
16 位	7	WTOB	-[WTOB(P) S D n]
		WTOBP	

操作数说明

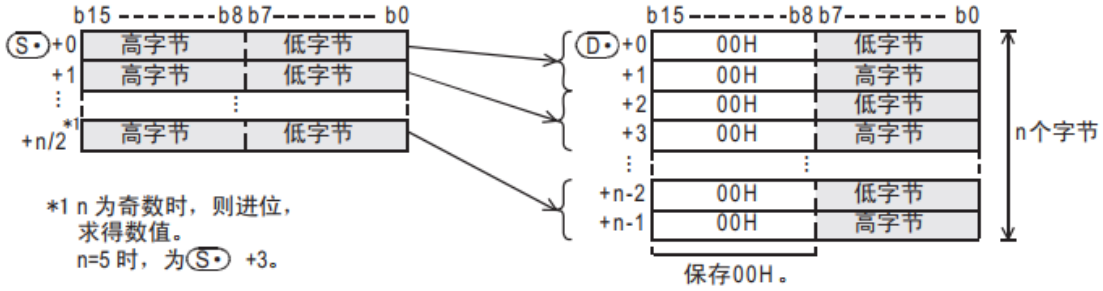
操作数	内容	类型
S	保存要按照字节单位进行分离的数据的软元件起始编号	BIN16 位
D	保存已经按照字节单位分离的结果的软元件起始编号	BIN16 位
n	要分离的字节数据个数(0≦n)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

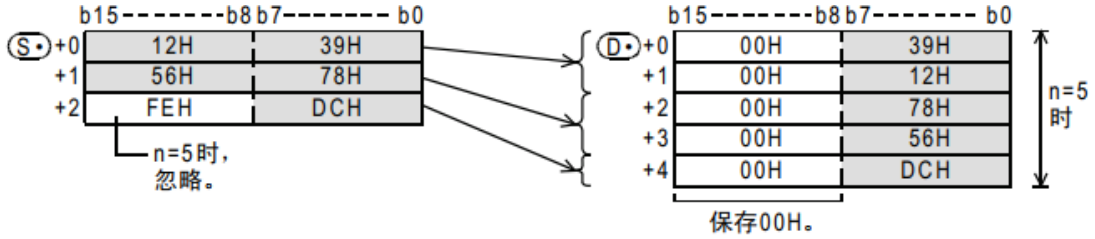
2、功能和动作说明

（1）16 位指令，将 S 开始的 n/2 个软元件中保存的 16 位数据分离成 n 个字节，如下所示保存到以 D 开始的 n 点软元件中。



（2）在保存分离后字节数据的软元件(D 以后)的高字节(8 位)中，保存 00H。

（3）n 为奇数时，如下图所示，在分离源的最终数据中，只有低字节(8 位)为对象数据。



3、程序举例

略。

4、注意事项

（1）当分离源软元件的 S～(S+n/2)超出了指定软元件的软元件范围时。 n 为奇数时，需要占用进位后数值的个数部分的软元件，报错 6706。

（2）当保存已分离的数据的软元件 D～(D+n-1)超出了指定软元件的软元件范围时，报错 6706。

（3）n=0 时，不执行指令。

2.10.3 BTOW/字节单位的数据结合指令

将连续的 16 位数据的低 8 位(低字节)结合在一起的指令。

1、指令格式

FNC142	指令所占程序步	指令分类	指令格式
16 位	7	BTOW	-[BTOW(P) S D n]
		BTOWP	

操作数说明

操作数	内容	类型

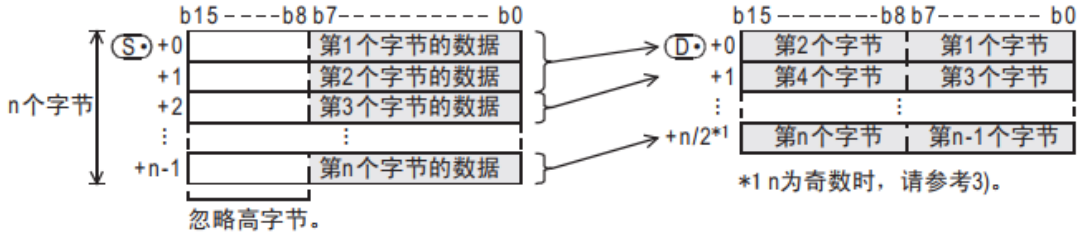
S	保存要按照字节单位结合的数据的软元件起始编号	BIN16 位
D	保存已经按照字节单位结合的结果的软元件起始编号	BIN16 位
n	要结合的字节数据个数(0≦n)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

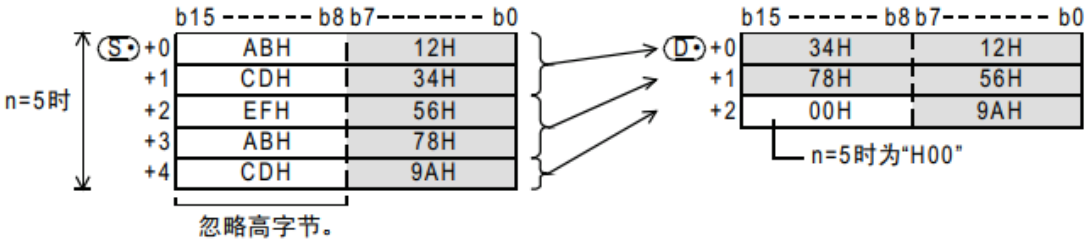
2、功能和动作说明

（1）16 位指令，将 S 开始的 n 点 16 位数据的低字节(8 位)结合在一起后的 16 位数据，如下所示保存到以 D 开始的 n/2 点软元件中。



（2）结合源的 16 位数据(S 以后)的高字节(8 位)被忽略。

（3）n 为奇数时，如下图所示，最终结合后的数据的高字节(8 位)为 00H。



3、程序举例

略。

4、注意事项

（1）结合源的软元件的 S～(S+n-1)中指定的软元件超出了该软元件范围时，报错 6706。

（2）当保存已结合数据的软元件 D～(D +n/2) 超出了指定软元件的软元件范围时。当 n 为奇数时，需要占用进位后数值的个数部分的软元件，报错 6706。

（3）n=0 时，不执行指令。

2.10.4 UNI/16 位数据的 4 位结合指令

将连续的 16 位数据的低 4 位结合在一起的指令。

1、指令格式

FNC143	指令所占程序步	指令分类	指令格式
16 位	7	UNI	[UNI(P) S D n]
		UNIP	

操作数说明

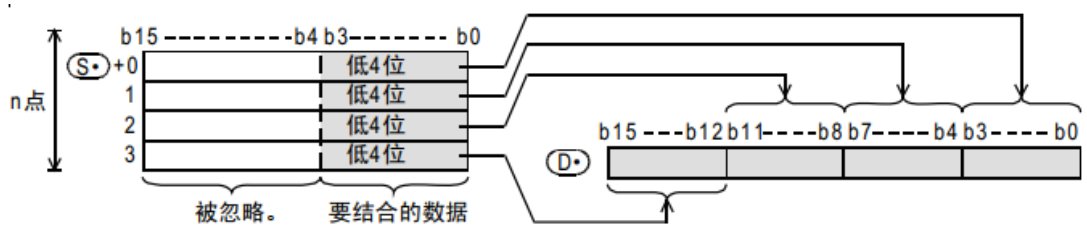
操作数	内容	类型
S	保存要结合的数据的软元件起始编号	BIN16 位
D	保存已结合的数据的软元件编号	BIN16 位
n	结合数(0～4，n=0 时不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

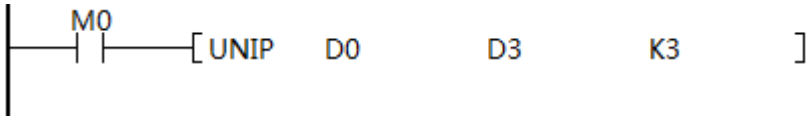
2、功能和动作说明

16 位指令，将 S 开始的 n 点 16 位数据的低 4 位结合后的 16 位数据，如下所示保存到 D 中。



3、程序举例

M0 为 ON 后，将 D0~D2 的低 4 位结合后，保存到 D3 中。



软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
D1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5
D2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	9
D3	0	0	0	0	1	0	0	1	0	1	0	1	0	0	0	1	2385

4、注意事项

注意连续执行指令时会一直执行结合。

2.10.5 DIS/16 位数据的 4 位分离指令

将连续的 16 位数据的低 4 位结合在一起的指令。

1、指令格式

FNC144	指令所占程序步	指令分类	指令格式
16 位	7	DIS	DIS(P) S D n
		DISP	

操作数说明

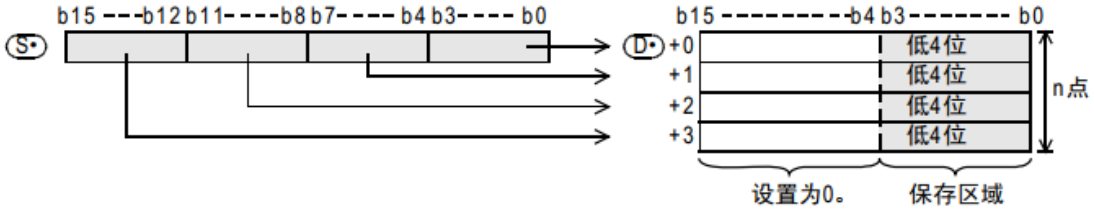
操作数	内容	类型
S	保存要分离的数据的软元件起始编号	BIN16 位
D	保存已分离的数据的软元件编号	BIN16 位
n	分离数(0～4，n=0 时不处理)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令，将 S 开始的 16 位数据以 4 位结合后为单位分离，如下所示保存到 D 中。



（2）n 范围(0～4，n=0 时不处理)

（3）D 开始的 n 点软元件的高 12 位设置为 0。

3、程序举例

略。

4、注意事项

（1）当 D 开始的 n 点软元件超出了软元件的范围时，报错 6706。

（2）n 指定的 0~4 意外数字报错 6706。

2.10.6 SWAP/高低字节互换指令

互换字数据的高 8 位和低 8 位的指令。

1、指令格式

FNC147	指令所占程序步	指令分类	指令格式
16 位	3	SWAP	-[(D)SWAP(P) S]
		SWAPP	
32 位	5	DSWAP	
		DSWAPP	

操作数说明

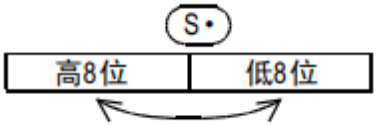
操作数	内容	类型
S	高低字节互换的字软元件	BIN16 位/32 位

操作数可用元件类型

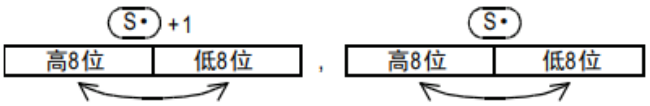
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

（1）16 位指令



（2）32 位指令



3、程序举例

略。

4、注意事项

即使是 32 位指令，也是执行各自的低 8 位和高 8 位的互换。与 XCH 指令的扩展功能相同。

2.10.7 SORT2/数据排序 2 指令

以指定的群数据(列)为基准，以行为单位，将由数据(行)和群数据(列)构成的数据表进行升序/降序重新排序的指令。在这个指令中，由于是在连续的软元件中保存数据(行方向)，所以便于增加数据(行)。

1、指令格式

FNC149	指令所占程序步	指令分类	指令格式
16 位	11	SORT2	-[(D)SORT2 S m1 m2 D n]
		\	
32 位	21	DSORT2	
		\	

操作数说明

操作数	内容	类型
S	保存数据表格的软元件起始编号[占用 m1×m2 点]	BIN16 位/32 位
m1	数据(行)数[1～32]	BIN16 位/32 位
m2	群数据(列)数[1～6]	BIN16 位/32 位
D	保存运算结果的软元件起始编号[占用 m1×m2 点]	BIN16 位/32 位
n	作为排序标准的群数据(列)的列编号[1～m2]	BIN16 位/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*								
m1														*	*				*	*			
m2																			*	*			
D														*	*								
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

针对 开始的(m1×m2)点的数据表格(排序前)，以 n 列的群数据为基准，将数据行进行升序或是降序的排列，然后保存到从 开始的(m1×m2)点的数据表格(排序后)中。

（2）32 位指令

针对(S+1,S)开始的(m1×m2)点的数据表格(排序前)，以 n 列的群数据为基准，将数据行进行升序或是降序的排列，然后保存到从(D+1,D)开始的(m1×m2)点的数据表格(排序后)中。

3、程序举例

在“n=K2(列号 2)”和“n=K3(列号 3)”的情况下，对如下所示的排序前的数据进行排序，动作如下所示。下面例举了 16 位运算的动作例子。执行 32 位运算时，请使用 BIN32 位构成数据表格。此外，如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据：

列号 / 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	S	S+1	S+2	S+3
		1	150	45	20
	2	S+4	S+5	S+6	S+7
		2	180	50	40
	3	S+8	S+9	S+10	S+11

		3	160	70	30
	4	S+12	S+13	S+14	S+15
		4	100	20	8
	5	S+16	S+17	S+18	S+19
		5	150	50	45

（1）以 n=K2(列号 2)为基准执行指令时的排序结果(升序的情况)

列号 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+1	D+2	D+3
		4	100	20	8
	2	D+4	D+5	D+6	D+7
		1	150	45	20
	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		3	160	70	30
	5	D+16	D+17	D+18	D+19
		2	180	50	40

（2）以 n=K3(列号 3)为基准执行指令时的排序结果(降序的情况)

列号 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+1	D+2	D+3
		3	160	70	30
	2	D+4	D+5	D+6	D+7
		2	180	50	40

	3	D+8	D+9	D+10	D+11
		5	150	50	45
	4	D+12	D+13	D+14	D+15
		1	150	45	20
	5	D+16	D+17	D+18	D+19
		4	100	20	8

4、注意事项

（1）相关软元件的动作

SM8029：指令结束标志，排序结束时置位；

SM8165：置 ON 时降序排列，OFF 时升序排列。

（2）以下操作会引起错误排序，请注意

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入 OFF 一次。
- 指令的使用次数的限制在程序中最多可同时驱动 2 次。
- 原来的数据和排序替换后的数据，请错开，不要重叠。

2.11 定位指令

X3 PLC 支持的定位指令有 PLSV、DRVI、DRVA、DVIT、ZRN、DSZR，32 点 PLC 支持 4 路 Y000~Y003，60 点支持 6 路 Y000~Y005。定位指令输出的脉冲和方向，接伺服控制器驱动电机工作，控制电机转动的定长和方向，将轴台上的光感接到 PLC 输入端子，输入信号有正反转极限、中断输入、近点信号、零点信号。

1. I/O 点分配

（1）输入点分配

用途	输入编号	说明
正转限位	所有本体输入点	请在任意输入端上接线 如果接了线的输入为 ON，则驱动正转极限标志位
反转极限	所有本体输入点	请在任意输入端上接线 如果接了线的输入为 ON，则驱动正转极限标志位
中断输入	X000~X005	通过中断输入指定功能，可以用中断输入指定寄存器 SD8336 指定脉冲输出端相对应的 X000~X005 输入编号，

		或者指定用户中断指令标志位
近点信号	所有本体输入点	请在 ZRN 和 DSZR 指令指定为近点信号的任意输入端上接线
零点信号	X000~X005	请在 DSZR 指定为零点信号的 X000~X005 输入上接线

（2）输出点分配

用途	输入编号	说明
脉冲信号	Y000~Y005	请对定位指令设定为脉冲输出端 Y000~Y005 接线
方向信号	所有本体输出点	支持本体任意输出点
清零信号	所有本体输出点	如果使用清零信号指定功能有效，可以用清零信号软元件指定寄存器来指定与脉冲输出端对应的任意输出

2.定位指令相关软元件

（1）特殊辅助继电器

软元件编号				名称	属性	对象指令
Y000	Y001					
SM8029				指令执行结束标志位	只读	PLSY/PLSR/DSZR/DVIT/ZRN/DRVI/DRVA
SM8329				指令执行异常结束标志位	只读	PLSY/PLSR/DSZR/DVIT/ZRN/PLSV/DRVI/DRVA
SM8338				加减速动作	读/写	PLSV
SM8340	SM8350			脉冲输出监控	只读	PLSY/PWM/PLSR/DSZR/DVIT/ZRN/PLSV/DRVI/DRVA
SM8341	SM8351			清零信号输出功能有效	读/写	DSZR/ZRN
						PLSY/PLSR/DSZR

SM8343	SM8353			正转极限	读/写	/DVIT/ZRN/PLSV /DRVI/DRVA
SM8344	SM8354			反转极限	读/写	PLSY/PLSR/DSZR /DVIT/ZRN/PLSV /DRVI/DRVA
SM8345	SM8355			近点信号逻辑反转	读/写	DSZR
SM8346	SM8356			零点信号逻辑反转	读/写	DSZR
SM8347	SM8357			中断信号逻辑反转	读/写	DVIT
SM8348	SM8358			定位指令驱动中	只读	PLSY/PWM/PLSR/ DSZR/DVIT/ZRN/ PLSV/DRVI/DRVA
SM8349	SM8359			脉冲停止指令	读/写	PLSY/PLSR/DSZR /DVIT/ZRN/PLSV /DRVI/DRVA
SM8460	SM8461			用户中断输入指令	读/写	DVIT

（2）特殊数据寄存器

软元件编号				名称	数据长（位）	初始值	对象指令
Y000	Y001						
SD8336（32 点）				中断输入指定	16	—	DVIT
SD8336,SD8337（60 点）					32		
SD8340 (低位)	SD8350 (低位)			当前寄存器值（PLS）	32	0	DSZR/DVIT/ZRN/ PLSV/DRVI/DRVA
SD8341 (高位)	SD8351 (高位)						
SD8342	SD8352			基底速度（HZ）	16	800	DSZR/DVIT/ZRN/ PLSV/DRVI/DRVA
SD8343 (低位)	SD8353 (低位)			最高速度（HZ）	32	200000	DSZR/DVIT/ZRN/ PLSV/DRVI/DRVA
SD8344 (高位)	SD8354 (高位)						

SD8345	SD8355			爬行速度（HZ）	16	1000	DSZR
SD8346（低位）	SD8356（低位）			原点回归速度（HZ）	32	50000	DSZR
SD8347（高位）	SD8357（高位）						
SD8348	SD8358			加速时间（ms）	16	1000	DSZR/DVIT/ZRN/PLSV/DRVI/DRVA
SD8349	SD8359			减速时间（ms）	16	1000	DSZR/DVIT/ZRN/PLSV/DRVI/DRVA
SD8464	SD8465			清零信号软元件指定	16	—	DSZR

3.特别说明：

◆DRVI，DRVA，DVIT 这三个指令相类似，DRVI 和 DVIT 是相对位置，DRVA 是绝对位置，如果在指令闭合前输入中断信号的话，DVIT 相当于 DRVI，但是这种情况只支持用户输入中断方式，不支持 X 输入中断方式。

◆需要注意的是指定相同输出端口的定位指令不能同时被驱动，包括高速指令 PLSY、PLSR、PWM，若同时驱动报 6764 错误。

◆指定了同一输入端口不能同时指定输入中断、高速计数器、DVIT、脉冲捕捉、ZRN、DSZR。考虑到输出的脉冲驱动电机转动的角度可能太小，所有定位指令脉冲个数不做限制，以尽可能走最长距离。

◆基底速度（可自行调整）。最小为 16Hz.

◆所有定位指令的脉冲和方向输出都不支持变址

◆两轴共用了同一个原点信号或者 DOG 信号，不影响，不报错。

◆DOG 信号和中断信号共用一个端子，不报错。

◆为安全考虑，除了回原点指令，其他定位指令都改为在左右极限位置，立即停止。

◆DOG 块的长度不允许同时覆盖原点和 DOG 开关。

特别注意：上表中数据长为 32 位的，例如 SD8340 和 SD8341 结对（当前值寄存器），在上位机上监控表里监控 SD8340，请选择 DWORD，否则只能看到 16 位 WORD 的数值。

2.11.1 DSZR/带 DOG 搜索的原点回归指令

执行原点回归，使机械位置与可编程控制器内的当前值寄存器一致的指令，带 DOG 搜索的原点回归。

1、指令格式

FNC150	指令所占程序步	指令分类	指令格式
16 位	9	DSZR \	[DSZR S1 S2 D1 D2]

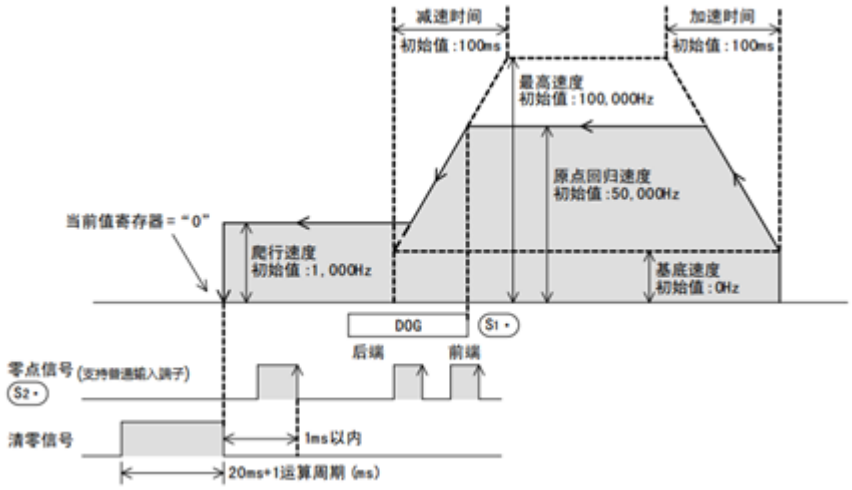
操作数说明

操作数	内容	类型
S1	输入近点信号的软元件编号	BIN16 位/32 位
S2	输入零点信号的输入编号，支持普通输入端子	BIN16 位/32 位
D1	输出脉冲的输出编号	位
D2	输出脉冲方向的对象编号	位

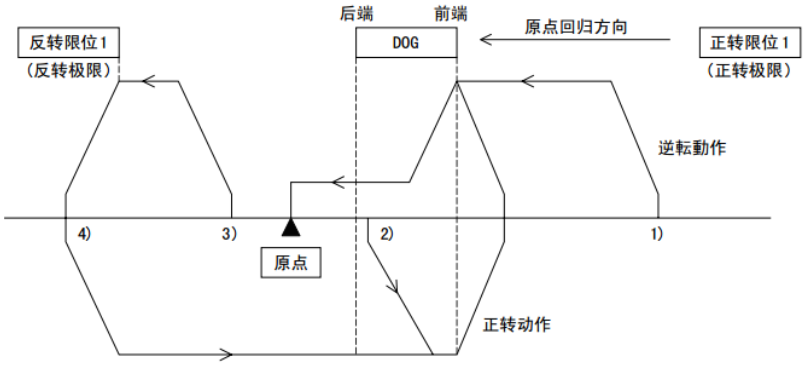
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1	*	*	*			*	*																
S2	*																						
D1		*																					
D2		*	*			*	*																

2、功能和动作说明



DOG 搜索功能



（1）开始位置在通过 DOG 前的时候

a）通过执行原点回归用指令， 开始原点回归动作。

b）以原点回归速度， 向原点回归方向开始移动。

c）一旦检测出 DOG 的前端， 就开始减速到爬行速度。

d）检测出 DOG 的后端后， 在检测出第一个零点信号时停止。

（2）开始位置在通过 DOG 内的时候

a）通过执行原点回归用指令， 开始原点回归动作。

b）以原点回归速度， 向与原点回归方向相反的方向开始移动。

c）检测出 DOG 的前端后减速停止。（离开 DOG。）

d）以原点回归速度， 向原点回归方向开始移动。（再次进入 DOG。）

e）一旦检测出 DOG 的前端， 就开始减速到爬行速度。

f）检测出 DOG 的后端后， 在检测出第一个零点信号时停止。

（3）开始位置在近点信号 OFF(通过 DOG 后)的时候

a）通过执行原点回归用指令， 开始原点回归动作。

b）以原点回归速度， 向原点回归方向开始移动。

c) 检测出反转限位 1(反转极限)时减速停止。

d) 以原点回归速度， 向与原点回归方向相反的方向开始移动。

e) 检测出 DOG 的前端后减速停止。 (检出(离开)DOG。)

f) 以原点回归速度， 向原点回归方向开始移动。 (再次进入 DOG。)

g) 一旦检测出 DOG 的前端， 就开始减速到爬行速度。

h) 检测出 DOG 的后端后， 在检测出第一个零点信号时停止。

（4）原点回归方向的限位开关(正转限位 1 或者反转限位 1)为 ON 的时候

a) 通过执行原点回归用指令， 开始原点回归动作。

b) 以原点回归速度， 向与原点回归方向相反的方向开始移动。

c) 检测出 DOG 的前端后减速停止。 (检出(离开)DOG。)

d) 以原点回归速度， 向原点回归方向开始移动。 (再次进入 DOG。)

e) 一旦检测出 DOG 的前端， 就开始减速到爬行速度。

f) 检测出 DOG 的后端后， 在检测出第一个零点信号时停止。

如果将近点信号与零点信号指定为相同的输入,此时变为和 ZRN 指令一样,不使用零点信号,而是根据近点信号（DOG）的前端和后端执行动作。

3、程序举例

略。

4、注意事项

（1）不需要指定原点回归方向，因为 DSZR 带搜索的功能。

(2)清零信号的软元件只能为 Y,并且统一放到 SD8XXX 里面去。Y000~Y005 由 SD8464~SD8469 的值来指定清零信号软元件编号，其初始值如下表，6/7/8/9 表示清零信号软元件为 Y006/Y007/Y010/Y011。若用户要指定其他编号，就自己修改 SD8464~SD8469 的值（从 0 开始的 10 进制数如下表，注意 Y 软元件的下标需要换算一下）。

	SD8464	SD8465	SD8466	SD8467	SD8468	SD8469
16 点	6(Y6)	7(Y7)	/	/	/	/
32 点	6(Y6)	7(Y7)	/	/	/	/
60 点	6(Y6)	7(Y7)	/	/	/	/

2.11.2 DVIT/中断定位指令

执行单速中断定长进给的指令。

1、指令格式

FNC151	指令所占程序步	指令分类	指令格式
16 位	9	DVIT	-[(D)DVIT S1 S2 D1 D2]
		\	
32 位	17	DDVIT	
		\	

操作数说明

操作数	内容	类型
S1	指定中断后的输出脉冲数(相对地址)	BIN16 位/32 位
S2	指定输出脉冲频率	BIN16 位/32 位
D1	指定输出脉冲的输出编号	位
D2	指定旋转方向信号的输出对象编号	位

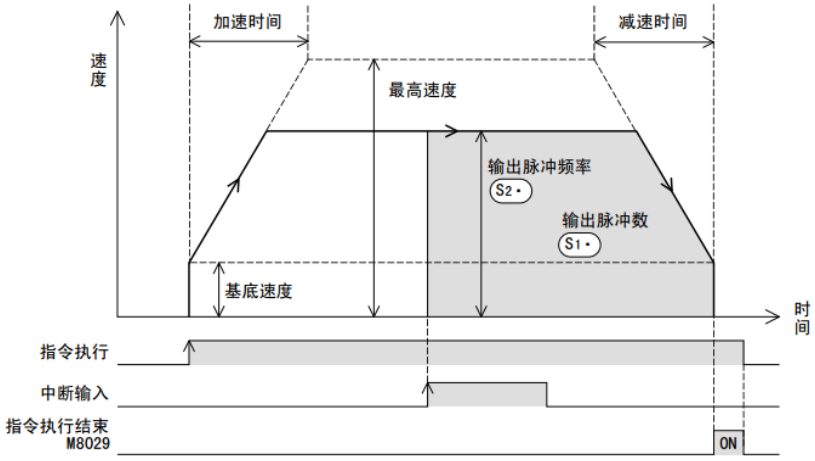
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*				*	*			
S2								*	*	*	*	*	*	*	*				*	*			
D1		*																					
D2		*	*			*	*																

2、功能和动作说明

S1 设定范围：16 位时-32768~32767（0 除外）；32 位时-999999~999999（0 除外）

S2 设定范围：16 位 16~32767（HZ）；32 位时 16~60000（HZ）。（不同型号 PLC 的最大频率不同）

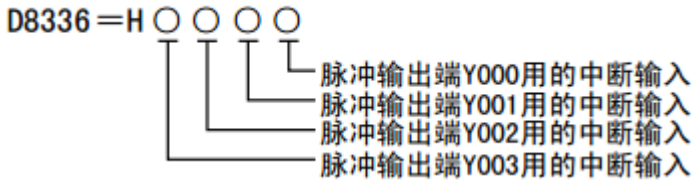


3、程序举例

略。

4、注意事项

（1）针对 D1 的输出，SD8336 直接指定作为中断输入为输入软元件编号（X000~X007）、或指定用户中断指令软元件。32 点 PLC 通过 SD8336 指定，60 点 PLC 通过 SD8336（低位）、SD8337（高位）指定，SD8337 所表示的 HXXXX 后面两个值表示的由低到高表示 Y004、Y005。



设定值	内容
0	将 X000 指定为中断输入信号
1	将 X001 指定为中断输入信号
…	…
7	将 X007 指定为中断输入信号

<table>
<tr><td>8</td><td>用户中断指令软元件指定为中断输入<table><tr><td>脉冲输出端软元件</td><td>用户中断指令软元件</td></tr><tr><td>Y000</td><td>M8460</td></tr><tr><td>Y001</td><td>M8461</td></tr><tr><td>Y002</td><td>M8462</td></tr><tr><td>Y003</td><td>M8463</td></tr><tr><td>Y004（60 点）</td><td>M8464</td></tr><tr><td>Y005（60 点）</td><td>M8465</td></tr></table></td></tr>
<tr><td>F</td><td>将 DVIT 指令中没有使用到的脉冲输出端软元件指定为 F</td></tr>
</table>

（2）如果在指令闭合前输入中断信号的话，DVIT 相当于按 DRVI 来动作，但是这种情况只支持用户输入中断方式，不支持外部输入中断方式（外部输入是边沿触发）。

2.11.3 ZRN/原点回归指令

执行原点回归，使机械位置与可编程控制器内的当前值寄存器一致的指令。需要 DOG 搜索功能时，请使用 DSZR(FNC 150)指令。

1、指令格式

<table>
<tr><td>FNC156</td><td>指令所占程序步</td><td>指令分类</td><td>指令格式</td></tr>
<tr><td rowspan="2">16 位</td><td rowspan="2">7</td><td>ZRN</td><td rowspan="4">[(D)ZRN S1 D1 D2]</td></tr>
<tr><td>\</td></tr>
<tr><td rowspan="2">32 位</td><td rowspan="2">13</td><td>DZRN</td></tr>
<tr><td>\</td></tr>
</table>

操作数说明

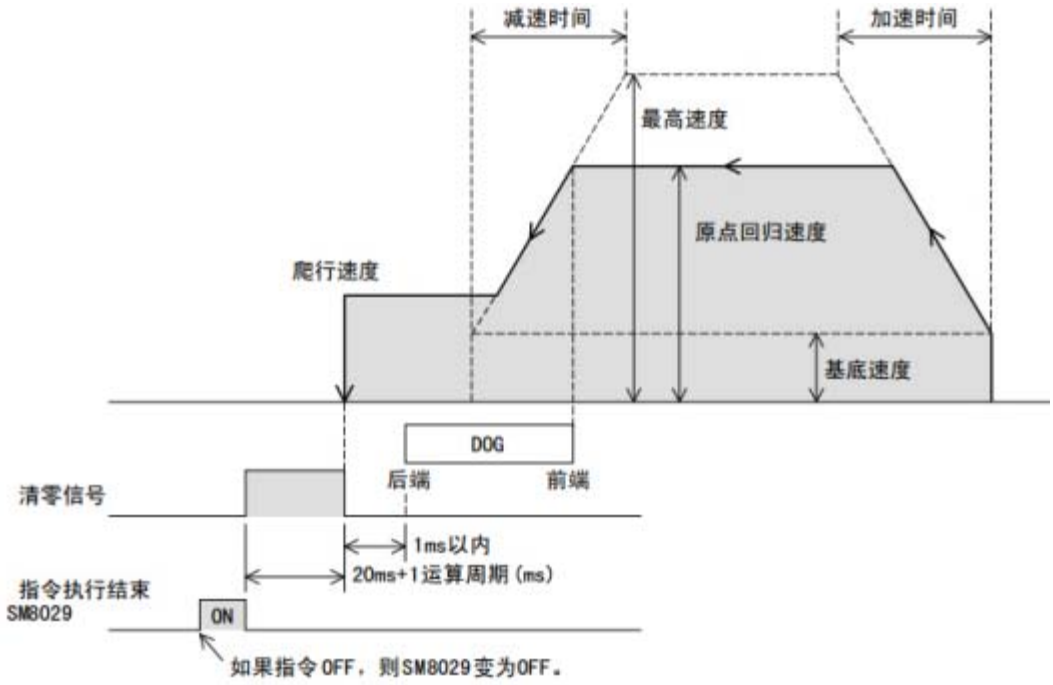
操作数	内容	类型
S1	指定输入近点信号(DOG)的软元件编号	位
D1	指定要输出脉冲的端子	位
D2	指定要输出方向的端子	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1	*																						
D1		*																					
D2		*																					

2、功能和动作说明

动作：



3、程序举例

略。

4、注意事项

基底速度≦原点回归速度≦最高速度，如果设定的原点回归速度超过最高速度，则以最高速度运行。

清零信号端子 Y 设置在特殊辅助寄存器。详情见 DSZR 指令。

2.11.4 PLSV/可变速脉冲输出指令

输出带旋转方向的可变速脉冲的指令。

1、指令格式

FNC157	指令所占程序步	指令分类	指令格式
16 位	9	PLSV	-[(D)PLSV S1 D1 D2]
		\	
32 位	17	DPLSV	
		\	

操作数说明

操作数	内容	类型
S1	指定输出脉冲频率的软元件编号	BIN16 位/32 位
D1	指定要输出脉冲的输出编号	位
D2	指定旋转方向信号的输出对象编号	位

操作数可用元件类型

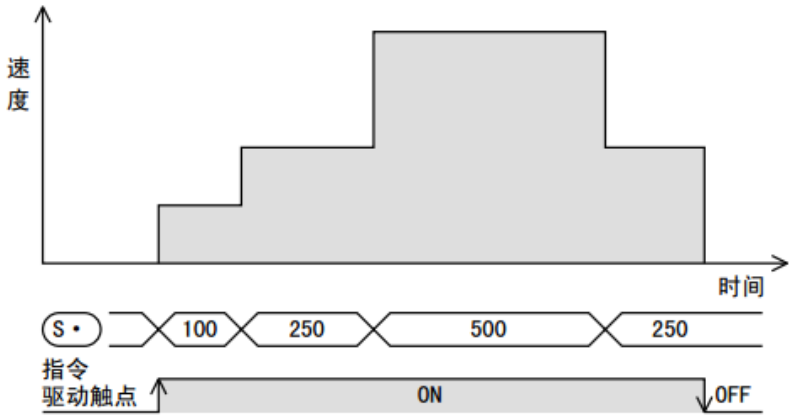
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*				*	*			
D1		*																					
D2		*	*			*	*																

2、功能和动作说明

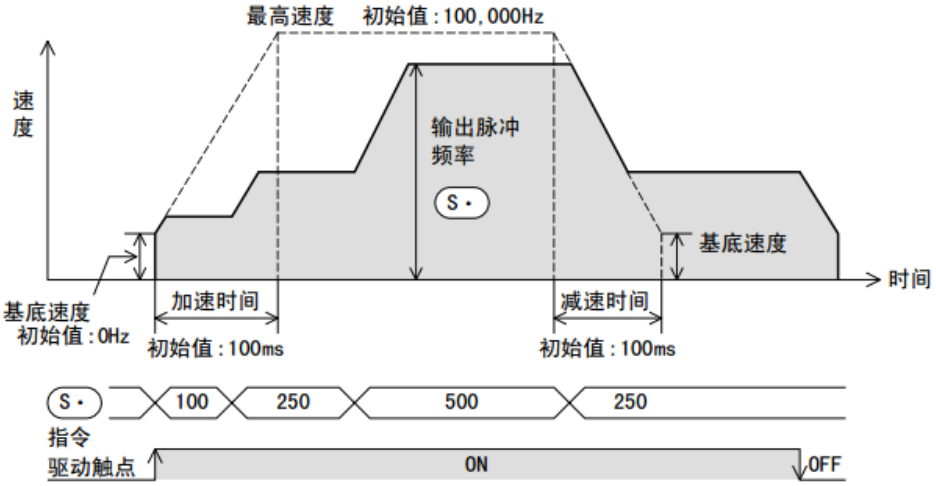
S1 设定范围：16 位时-32768~32767（HZ）；32 位时-60000~60000（HZ）。（不同型号 PLC 的最大频率不同）最低频率 16HZ.

动作：

（1）无加减速时（SM8338=OFF），这种模式在 X3PLC 上没有。



（2）有加减速时（SM8338=ON）



3、程序举例

略。

4、注意事项

（1）PLSV 指令即使在脉冲输出过程中，也能随意更改输出脉冲频率 S1。

（2）将 S1 变为 K0 时，脉冲输出在带加减速时减速停止，不带加减速时直接停止。

2.11.5 DRVI/相对定位指令

以相对驱动方式执行单速定位的指令。用带正/负的符号指定从当前位置开始的移动距离的方式，也称为增量(相对)驱动方式。

1、指令格式

FNC158	指令所占程序步	指令分类	指令格式
16 位	9	DRVI	-[(D)DRVI S1 S2 D1 D2]
		\	
32 位	17	DDRVI	
		\	

操作数说明

操作数	内容	类型
S1	指定输出脉冲数	BIN16 位/32 位
S2	指定输出脉冲频率	BIN16 位/32 位
D1	指定要输出脉冲的输出编号	位
D2	指定旋转方向信号的输出对象编号	位

操作数可用元件类型

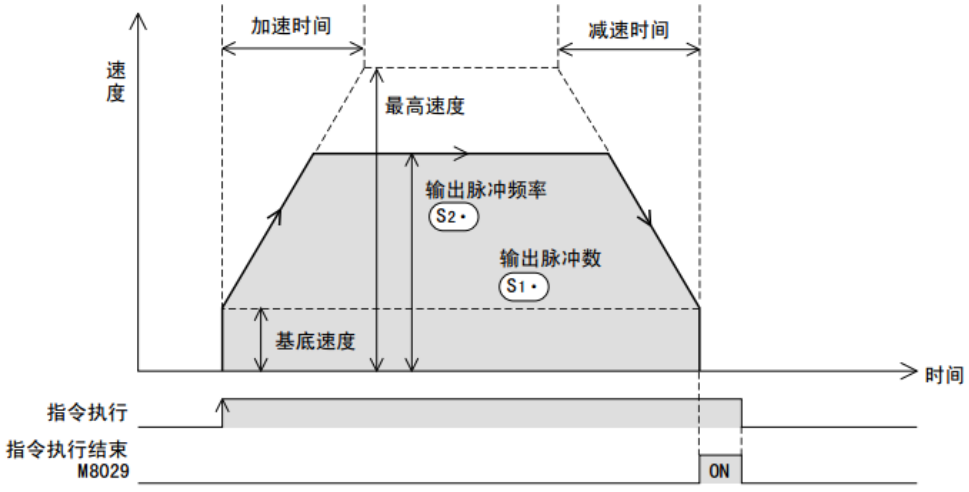
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*				*	*			
S2								*	*	*	*	*	*	*	*				*	*			
D1		*																					
D2		*	*			*	*																

2、功能和动作说明

S1 设定范围：16 位时-32768~32767（0 除外）；32 位时-999999~999999（0 除外）

S2 设定范围：16 位 16~32767（HZ）；32 位时 16~60000（HZ）。（不同型号 PLC 的最大频率不同）

动作：



3、程序举例

略。

4、注意事项

（1）在指令执行过程中，即使改变操作数的内容，也不反映到当前的运行中。在下次的指令驱动时才有效。

（2）在指令执行过程中，驱动触点为 OFF 时，减速停止。且此时指令执行结束标志位 M8029 不动作。

（3）动作方向的极限标志位(正转限位标志位或者反转限位标志位)动作时，减速停止。此时，指令执行异常结束标志位 M8329 置 ON，结束指令的执行。

2.11.6 DRVA/绝对定位指令

以相对驱动方式执行单速定位的指令。用带正/负的符号指定从当前位置开始的移动距离的方式，也称为增量(相对)驱动方式。

1、指令格式

FNC159	指令所占程序步	指令分类	指令格式
16 位	9	DRVA	[(D)DRVA S1 S2 D1 D2]
		\	
32 位	17	DDRVA	
		\	

操作数说明

操作数	内容	类型
S1	指定输出脉冲数	BIN16 位/32 位
S2	指定输出脉冲频率	BIN16 位/32 位
D1	指定要输出脉冲的输出编号	位
D2	指定旋转方向信号的输出对象编号	位

操作数可用元件类型

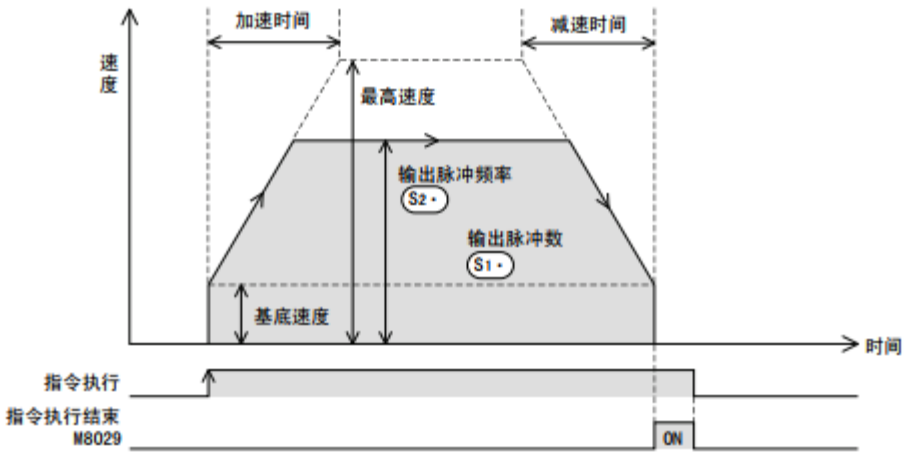
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*				*	*			
S2								*	*	*	*	*	*	*	*				*	*			
D1		*																					
D2		*	*			*	*																

2、功能和动作说明

S1 设定范围：16 位时-32768~32767；32 位时-999999~999999

S2 设定范围：16 位 16~32767（HZ）；32 位时 16~60000（HZ）。（不同型号 PLC 的最大频率不同）

动作：



3、程序举例

略。

4、注意事项

（1）在指令执行过程中，即使改变操作数的内容，也不反映到当前的运行中。在下次的指令驱动时才有效。

（2）在指令执行过程中，驱动触点为 OFF 时，减速停止。且此时指令执行结束标志位 M8029 不动作。

（3）动作方向的极限标志位(正转限位标志位或者反转限位标志位)动作时，减速停止。此时，指令执行异常结束标志位 M8329 置 ON，结束指令的执行。

2.12 时钟运算指令

针对时钟数据进行运算、比较的指令，还可以修改内置时钟时间。

2.12.1 TCMP/时钟数据比较指令

将比较基准时间和时间数据进行大小比较，根据比较的结果控制位软元件 ON/OFF。。

1、指令格式

FNC160	指令所占程序步	指令分类	指令格式
16 位	11	TCMP	[TCMP(P) S1 S2 S3 S D]
		TCMPP	

操作数说明

操作数	内容	类型
S1	指定比较基准时间的“时”。[设定范围: 0～23]	BIN16 位
S2	指定比较基准时间的“分”。[设定范围: 0～59]	BIN16 位
S3	指定比较基准时间的“秒”。[设定范围: 0～59]	BIN16 位
S	指定时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	根据比较结果 ON/OFF 位软元件。(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S3								*	*	*	*	*	*	*	*	*	*	*	*	*			
D1												*	*	*	*			*					
D2		*	*			*	*											*					

2、功能和动作说明

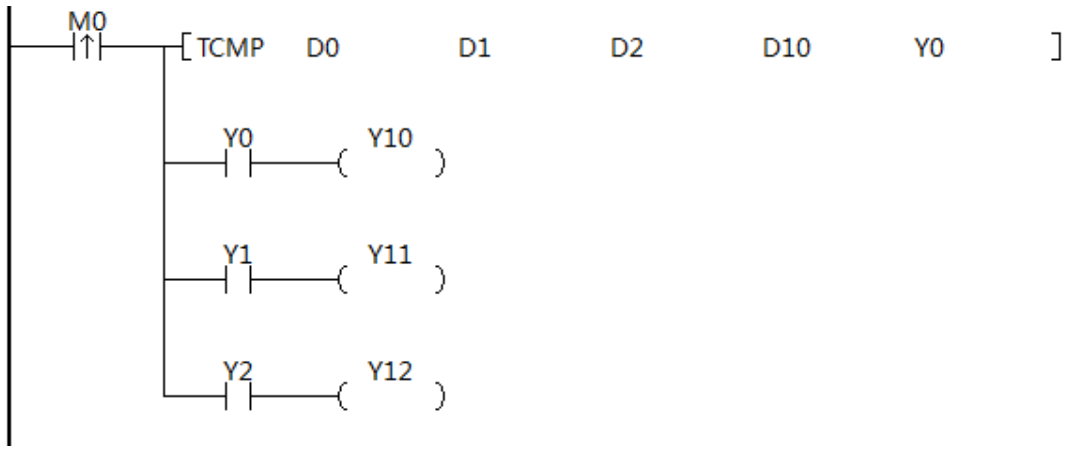
将比较基准时间(时、分、秒)（S1，S2 ，S3 ） 的时间与时间数据(时、分、秒)（S ， S+1，S+2）进行大小比较，根据其大小一致的结果 ON/OFF D 开始的 3 点。

当（S1，S2，S3）的时间 >（S，S+1，S+2）的时间，D 置位；

当（S1，S2，S3）的时间 =（S，S+1，S+2）的时间，D+1 置位；

当（S1，S2，S3）的时间 <（S，S+1，S+2）的时间，D+2 置位；

3、程序举例



当 D0、D1、D2 对应的时间比 D10~D12 组成的时间大时，Y0 置位，输出 Y10；

当 D0、D1、D2 对应的时间比 D10~D12 组成的时间相等时，Y1 置位，输出 Y11；

当 D0、D1、D2 对应的时间比 D10~D12 组成的时间小时，Y2 置位，输出 Y12。

4、注意事项

（1）即使是指令输入为 OFF,TCMP 指令不执行时，D～D+2 也会保持当指令输入从 ON 变为 OFF 之前的状态。

（2)使用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时，请使用 TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

2.12.2 TZCP/时钟数据区间比较指令

将上下 2 点的比较基准时间和时间数据进行大小比较，根据比较的结果控制指定位软元件的 ON/OFF。

1、指令格式

FNC161	指令所占程序步	指令分类	指令格式
16 位	9	TZCP	TZCP(P) S1 S2 S3 D
		TZCPP	

操作数说明

操作数	内容	类型
S1	指定比较下限时间(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定比较上限时间(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S3	指定时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	根据比较结果 ON/OFF 位软元件。(占用 3 点)	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
S3								*	*	*	*	*	*	*	*	*	*	*	*	*			
D2		*	*			*	*											*					

2、功能和动作说明

将上下 2 点比较基准时间(时、分、秒)与以 S 开头的 3 点时间数据(时、分、秒)进行比较，根据比较的结果 ON/OFF 从 D 开始的 3 点软元件。

当（S1，S1+1，S1+2）的时间 >（S，S+1，S+2）的时间，D 置位；

当（S1，S1+1，S1+2）的时间 ≤（S，S+1，S+2） ≤ （S2，S2+1，S2+2）的时间，D+1 置位；

当（S1，S1+1，S1+2）的时间 >（S，S+1，S+2）的时间，D+2 置位；

3、程序举例

略。

4、注意事项

用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时，请使用 TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

2.12.3 TADD/时钟数据加法运算指令

将 2 个时间数据进行加法运算后，保存在字软元件中。

1、指令格式

FNC162	指令所占程序步	指令分类	指令格式
16 位	7	TADD TADDP	-[TADD(P) S1 S2 D]

操作数说明

操作数	内容	类型
S1	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定进行加法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	保存两个时间数据(时、分、秒)加法运算的结果。(占用 3 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*					
D												*	*	*	*			*					

2、功能和动作说明

将（S1，S1+1，S1+2） 的时间数据(时、分、秒) 与[S2，S2+1，S2+2） 的时间数据(时、分、秒)进行加法运算，其结果保存到（D，D+1，D+2）(时、分、秒)中。

“时”范围：0~23

“分”范围：0~59

“秒”范围：0~59

3、程序举例

略。

4、注意事项

主要相关标志位的动作：当运算结果超出 24 小时时，进位标志位变为 ON，从单纯的加法运算值中减去 24 个小时后将该时间作为运算。运算结果为 0(0 时 0 分 0 秒)时，零位标志位变为 ON。

2.12.4 TSUB/时钟数据减法运算指令

将 2 个时间数据进行减法运算后，保存在字软元件中。

1、指令格式

FNC163	指令所占程序步	指令分类	指令格式
16 位	7	TSUB	-[TSUB(P) S1 S2 D]
		TSUBP	

操作数说明

操作数	内容	类型

S1	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
S2	指定进行减法运算的时间数据(时、分、秒)的“时”。(占用 3 点)	BIN16 位
D	保存两个时间数据(时、分、秒)减法运算的结果。(占用 3 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*					
D																		*					

2、功能和动作说明

从（S1，S1+1，S1+2） 的时间数据(时、分、秒) 中减去（S2，S2+1，S2+2） 的时间数据(时、分、秒)，其结果保存到[D，D+1，D+2）(时、分、秒)中。

“时”范围：0~23

“分”范围：0~59

“秒”范围：0~59

3、程序举例

略。

4、注意事项

相关标志位的动作：当运算结果小于 0 时，借位标志位变为 ON，从单纯的减法运算值中加上 24 个小时后，将该时间作为运算结果被保存。运算结果为 0(0 时 0 分 0 秒)时，零位标志位变为 ON。

2.12.5 HTOS/时、分、秒数据的秒转换指令

将[时、分、秒）单位的时间(时刻)数据转换成秒单位的数据的指令。

1、指令格式

FNC164	指令所占程序步	指令分类	指令格式
16 位	5	HTOS	-[(D)HTOS(P) S D]
		HTOSP	
32 位	9	DHTOS	
		DHTOSP	

操作数说明

操作数	内容	类型
S	保存转换前的时间(时刻)数据(时、分、秒)的软元件的起始编号	BIN16 位
D	保存转换后的时间(时刻)数据(秒)的软元件编号。	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

8. 功能和动作说明

（1）16 位指令

将（S，S+1，S+2）的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到 D 中。

S 范围：0~9

S+1 范围：0~59

S+2 范围：0~59

（2）32 位指令

将（S，S+1，S+2）的时间(时刻)数据(时、分、秒)换算成秒后，将结果保存到（D+1，D）中。

S 范围：0~32767

S+1 范围：0~59

S+2 范围：0~59

9. 程序举例

略。

4、注意事项

S、S+1、S+2 的数据超出与指令对应的可设定范围时，报错 6706。

2.12.6 STOH/秒数据的（时、分、秒）转换指令

将秒单位的时间(时刻)数据转换成[时、分、秒）单位的数据的指令。

1、指令格式

FNC165	指令所占程序步	指令分类	指令格式
16 位	5	STOH	[(D)STOH(P) S D]
		STOHP	
32 位	9	DSTOH	
		DSTOHP	

操作数说明

操作数	内容	类型
S	保存转换前的时间(时刻)数据(秒)的软元件编号	BIN16/32 位
D	保存转换后的时间(时刻)数据(时、分、秒)的软元件起始编号	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

（1）16 位指令

将 S 的秒数据换算成时、分、秒，其结果保存到（D，D+1，D+2）(时、分、秒)中。

S 范围：0~32767

（2）32 位指令

将（S+1，S）的秒数据换算成时、分、秒，其结果保存到（D，D+1，D+2）(时、分、秒)中。

（S+1，S）范围：0~117964799

3、程序举例

略。

4、注意事项

要转换的数据超出与指令对应的可设定范围时，报错 6706。

2.12.7 TRD/读出时钟数据指令

将可编程控制器内置实时时钟的时钟数据(SD8013~SD8019)按照下面的格式读出 D~D+6 中。

1、指令格式

FNC166	指令所占程序步	指令分类	指令格式
16 位	3	TRD	[TRD(P) D]
		TRDP	

操作数说明

操作数	内容	类型
D	指定保存读出时间数据的起始软元件编号。(占用 7 点)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D												*	*	*	*			*					

2、功能和动作说明

16 位指令，将可编程控制器内置实时时钟的时钟数据(SD8013～SD8019)按照下面的格式读出到 D~D+6 中。

特殊数据寄存器	软元件	项目	时钟数据		软元件	项目
	SD8018	年（公历）	0~99（公历后 2 位数）	→	D	年（公历）
	SD8017	月	1~12	→	D+1	月
	SD8016	日	1~31	→	D+2	日
	SD8015	时	0~23	→	D+3	时
	SD8014	分	0~59	→	D+4	分
	SD8013	秒	0~59	→	D+5	秒
	SD8019	星期	0（日）~6（六）	→	D+6	星期

3、程序举例

略。

4、注意事项

D 占用 7 点软元件，请注意不要与机器其他控制中使用的软元件重复。

2.12.8 TWR/写入时钟数据指令

向可编程控制器内置实时时钟写入时钟数据的指令。

1、指令格式

FNC167	指令所占程序步	指令分类	指令格式
16 位	3	TWR	[TWR(P) S]
		TWRP	

操作数说明

操作数	内容	类型
S	指定写入时间数据的源地址的起始软元件编号。（占用 7 点）	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					

2、功能和动作说明

将设定的时钟数据 S~S+6 写入可编程控制器内置实时时钟的时钟数据(SD8013~SD8019)中。

设定时间用的数据	软元件	项目	时钟数据		软元件	项目	特殊数据寄存器
	S	年（公历）	0~99（公历后 2 位数）	→	SD8018	年（公历）	
	S+1	月	1~12	→	SD8017	月	
	S+2	日	1~31	→	SD8016	日	
	S+3	时	0~23	→	SD8015	时	
	S+4	分	0~59	→	SD8014	分	
	S+5	秒	0~59	→	SD8013	秒	
	S+6	星期	0（日）~6（六）	→	SD8019	星期	

3、程序举例

略。

4、注意事项

（1）执行 TWR(FNC 167)指令后，实时时钟的时钟数据即刻被更改。因此，请先将快几分钟的时钟数据传送到 S~S+6 中，等到变成正确的时间时才执行指令。

（2）使用这个指令设定时钟数据(时间校准)时，不需要控制特殊辅助继电器 SM8015(时间停止以及时间校准)。

（3）设定了不可能显示的日期时间数值时，不执行时钟数据的变更。此时，请设定正确的时钟数据后再次写入。

（4）星期(SD8019)与写入数值无关,是根据日期内容自动补正。

（5）占用 S 开始的连续 7 点软元件，请注意不要与机器其他控制中使用的软元件重复。

（6）设定时间时，设定快 1 分钟的时间，等到达正确时间时使 TWR 指令前面的触发开关置位，以使时钟数据更新。

2.12.9 HOUR/计时表指令

以 1 个小时为单位，对输入触点持续 ON 的时间进行累加检测的指令。

1、指令格式

FNC169	指令所占程序步	指令分类	指令格式
16 位	7	HOUR	[(D)HOUR S D1 D2]
		\	
32 位	13	DHOUR	
		\	

操作数说明

操作数	内容	类型
S	使 D2 为 ON 的时间(以 1 个小时为单位设定)	BIN16/32 位
D1	以 1 个小时为单位的当前值(指定停电保持用数据寄存器)	BIN16/32 位
D2	报警输出的起始编号	位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D1														*	*			*					
D2		*	*			*	*											*					

2、功能和动作说明

当指令输入的累计 ON 时间超出了 S 时，D2 变为 ON。D1+1 中不满 1 个小时的当前值，以 1 秒单位被保存。

S：D2 变 ON 为止的时间以 1 个小时为单位指定。
D1：以 1 个小时为单位的当前值
D1+1 ：不满 1 个小时的当前值(1 秒单位)
D2：报警输出目标地址编号，当前值 D1 在 S 的指定时间以上时置 ON。

3、程序举例

略。

4、注意事项

（1）由于即使断开可编程控制器的电源后，也可以继续使用当前值数据，所以请在 D1 中指定停电保持用的数据寄存器。使用一般的数据寄存器时，由于可编程控制器的电源 OFF 和 STOP→RUN 的操作，当前值会被清除。

（2）报警输出 D2 为 ON 以后，测量仍能继续。再次开始进行测量的情况下，请清除 D1～D1+1(32 位指令为 D1～D1+2)的当前值。清除当前值时后，报警输出将变为 OFF。

（3）当前值 D1 达到最大值时停止测量。要继续测量时，请清除 D1～D1+1(32 位指令为 D1～D1+2)的当前值。

（4）D1 占用 2 个(16 位运算)或者 3 个(32 位运算)软元件。请注意不要与机器其他控制中使用的软元件重复。

2.13 格雷码指令

主要用于格雷码数据的转换操作。

2.13.1 GRY/格雷码的转换指令

将 BIN 值转换成格雷码后进行传送的指令。

1、指令格式

FNC170	指令所占程序步	指令分类	指令格式
16 位	5	GRY	-[(D)GRY(P) S D]
		GRYP	
32 位	9	DGRY	
		DGRYP	

操作数说明

操作数	内容	类型
S	转换源数据，或是保存转换源数据的字软元件	BIN16/32 位
D	保存转换后数据的字软元件	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

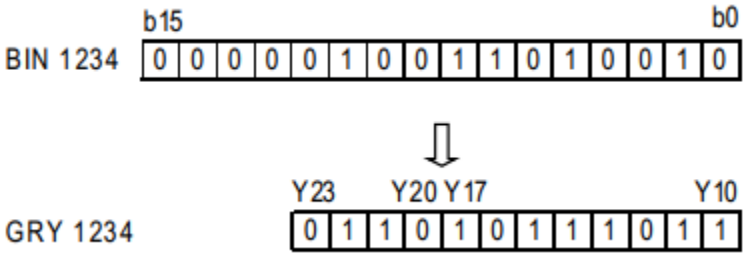
2、功能和动作说明

源数据(BIN)→目标数据(GRY)的转换传送指令。格雷码的值只需要在原来的二进制的基础上右移一位再异或原来的二进制值即可得到。

3、程序举例

S 为 K3456，D 为 K3Y0 时





4、注意事项

（1）数据的转换速度取决于可编程控制器的扫描时间。

（2）16 位指令时，S 范围为 0~32767；32 位指令时，S 范围为 0~2147483647。

2.13.2 GBIN/格雷码的逆转换指令

将格雷码转换成 BIN 值后进行传送的指令。

1、指令格式

FNC171	指令所占程序步	指令分类	指令格式
16 位	5	GBIN	-[(D)GBIN(P) S D]
		GBINP	
32 位	9	DGBIN	
		DGBINP	

操作数说明

操作数	内容	类型
S	转换源数据，或是保存转换源数据的字软元件	BIN16/32 位
D	保存转换后数据的字软元件	BIN16/32 位

操作数可用元件类型

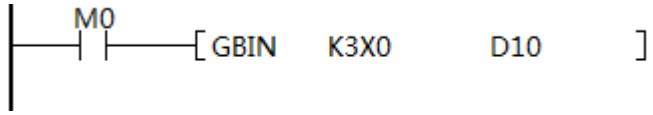
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D									*	*	*	*	*	*	*	*	*	*					

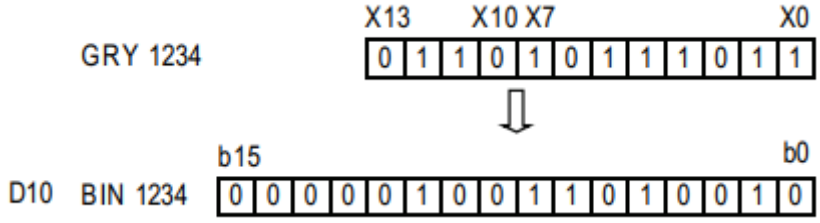
2、功能和动作说明

源数据(GRY)→目标数据(BIN)的转换传送指令。二进制格雷码转换成自然二进制码,其法则是保留格雷码的最高位作为自然二进制码的最高位,而次高位自然二进制码为高位自然二进制码与次高位格雷码相异或,而自然二进制码的其余各位与次高位自然二进制码的求法相类似。

3、程序举例

S 为 K3X000, D 为 D10 时





4、注意事项

16 位指令时，S 范围为 0~32767；32 位指令时，S 范围为 0~2147483647。

2.14 数据块指令

对批量数据同时进行加法、减法、比较运算操作的指令。

2.14.1 BK+/数据块的加法运算指令

数据块的 BIN 加法运算的指令。

1、指令格式

FNC192	指令所占程序步	指令分类	指令格式
16 位	9	BK+	(D)BK+(P) S1 S2 D n
		BK+P	
32 位	17	DBK+	
		DBK+P	

操作数说明

操作数	内容	类型
S1	保存执行加法运算的数据的软元件起始编号	BIN16/32 位
S2	执行加法运算的常数，或是保存执行加法运算的数据的软元件起始编号	BIN16/32 位
D	保存运算结果的软元件起始编号	BIN16/32 位
n	数据的个数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*	*	*			
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

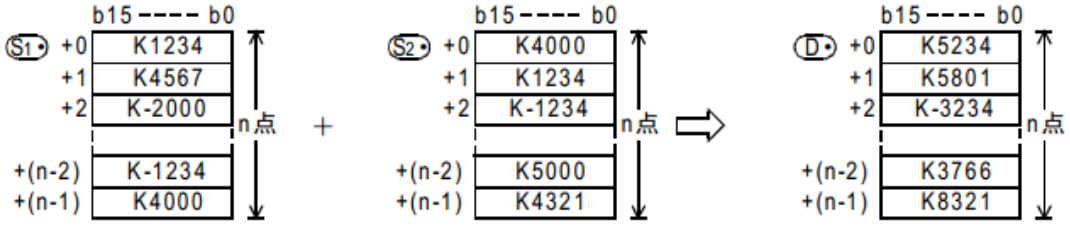
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行加法运算后，将运算结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行加法运算后，将运算结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令运算为例。



4、注意事项

（1）运算结果中产生数据上溢出、下溢出时，进位标志不置位。

（2）出现以下情况会报错 6706。

•S1、S2、D 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。
•S1 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。
•S2 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。

2.14.2 BK-/数据块的减法运算指令

数据块的 BIN 减法运算的指令。

1、指令格式

FNC193	指令所占程序步	指令分类	指令格式
16 位	9	BK-	[(D)BK-(P) S1 S2 D n]
		BK-P	
32 位	17	DBK-	
		DBK-P	

操作数说明

操作数	内容	类型
S1	保存执行减法运算的数据的软元件起始编号	BIN16/32 位
S2	执行减法运算的常数，或是保存执行减法运算的数	BIN16/32 位

	据的软元件起始编号	
D	保存运算结果的软元件起始编号	BIN16/32 位
n	数据的个数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2												*	*	*	*			*	*	*			
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

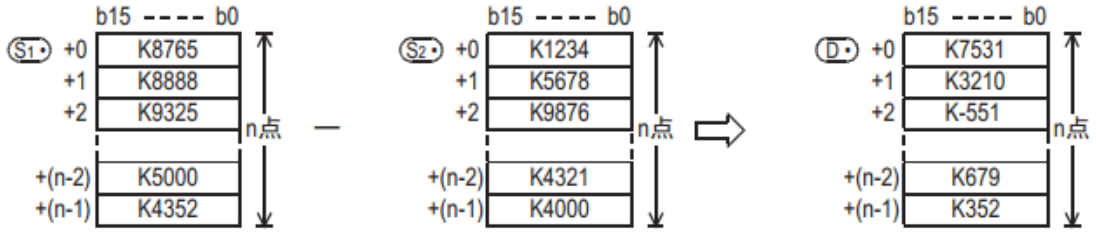
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行减法运算后，将运算结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行减法运算后，将运算结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令运算为例。



4、注意事项

（1）运算结果中产生数据上溢出、下溢出时，进位标志不置位。

（2）出现以下情况会报错 6706。

•S1、S2、D 开始的 n 点(32 位运算时为 2n 点)软元件超出相应的软元件范围时。
•S1 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。
•S2 开始的 n 点软元件和 D 开始的 n 点软元件重复时(32 位运算时为 2n 点)。

2.14.3 BKCMP=、>、<、<=、>=/数据块的比较指令

这些指令按照各个指令的比较条件来比较数据块。

1、指令格式

FNC NO	指令位数	指令所占程序步	指令分类	指令格式
194	16 位	9	BKCMP=	┤[(D)BKCMP=(P) S1 S2 D n]
			BKCMPP=	
	32 位	17	DBKCMP=	
			DBKCMPP=	
195	16 位	9	BKCMP>	┤[(D)BKCMP>(P) S1 S2 D n]
			BKCMPP>	
	32 位	17	DBKCMP>	
			DBKCMPP>	
196	16 位	9	BKCMP<	┤[(D)BKCMP<(P) S1 S2 D n]
			BKCMPP<	
	32 位	17	DBKCMP<	
			DBKCMPP<	
197	16 位	9	BKCMP<>	┤[(D)BKCMP<>(P) S1 S2 D n]
			BKCMPP<>	
	32 位	17	DBKCMP<>	
			DBKCMPP<>	

198	16 位	9	BKCMP<=	┤[(D)BKCMP<=(P) S1 S2 D n]
			BKCMPP<=	
	32 位	17	DBKCMP<=	
			DBKCMPP<=	
199	16 位	9	BKCMP>=	┤[(D)BKCMP>=(P) S1 S2 D n]
			BKCMPP>=	
	32 位	17	DBKCMP>=	
			DBKCMPP>=	

操作数说明

操作数	内容	类型
S1	比较值或是保存比较值的软元件编号	BIN16/32 位
S2	保存比较源数据的软元件起始编号	BIN16/32 位
D	保存比较结果的软元件起始编号	BIN16/32 位
n	要比较的数据数	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*	*	*			
S2												*	*	*	*			*					
D		*	*			*	*											*					
n														*	*				*	*			

2、功能和动作说明

（1）16 位指令

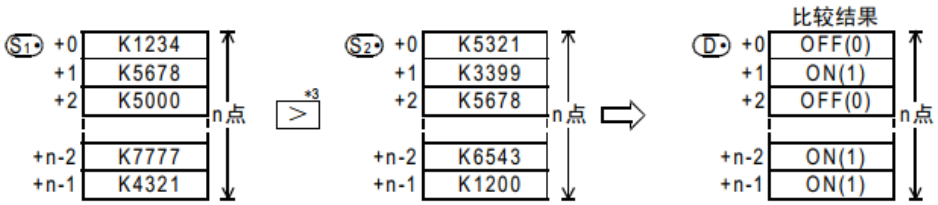
将 S1 开始 n 点 16 位数据和 S2 开始的 n 点 16 位数据(BIN)进行比较后，将比较结果保存到 D 开始的 n 点中。

（2）32 位指令

将（S1+1，S1）开始 2n 点 32 位数据和（S2+1，S2）开始的 2n 点 32 位数据(BIN)进行比较后，将比较结果保存到（D+1，D）开始的 2 n 点中。

3、程序举例

以 16 位指令 BKCMP>指令为例。



4、注意事项

（1）指令比较结果如下

FNC No	16 位指令	32 位指令	比较结果 ON 的条件	比较结果 OFF 的条件
194	BKCMP=	DBKCMP=	S1=S2	S1≠S2
195	BKCMP>	DBKCMP>	S1>S2	S1≤S2
196	BKCMP<	DBKCMP<	S1<S2	S1≥S2
197	BKCMP<>	DBKCMP<>	S1≠S2	S1=S2
198	BKCMP<=	DBKCMP<=	S1≤S2	S1>S2
199	BKCMP>=	DBKCMP>=	S1≥S2	S1<S2

（2）SM8090 块比较信号的动作：数据块指令的比较结果都为 ON(1)时变为 ON。

2.15 字符串控制指令

针对字符串进行控制的相关操作指令，包含对字符串的转换、替换、检索、取出等。

2.15.1 STR/BIN→字符串的转换指令

将 BIN 数据转换成字符串(ASCII 码)的指令。

1、指令格式

FNC200	指令所占程序步	指令分类	指令格式
16 位	7	STR	-[(D)STR(P) S1 S2 D]
		STRP	
32 位	13	DSTR	
		DSTRP	

操作数说明

操作数	内容	类型
S1	保存要转换数值的位数的软元件起始编号	BIN16/32 位
S2	保存要转换的 BIN 数据的软元件编号	BIN16/32 位
D	保存已转换的字符串的软元件起始编号	字符串

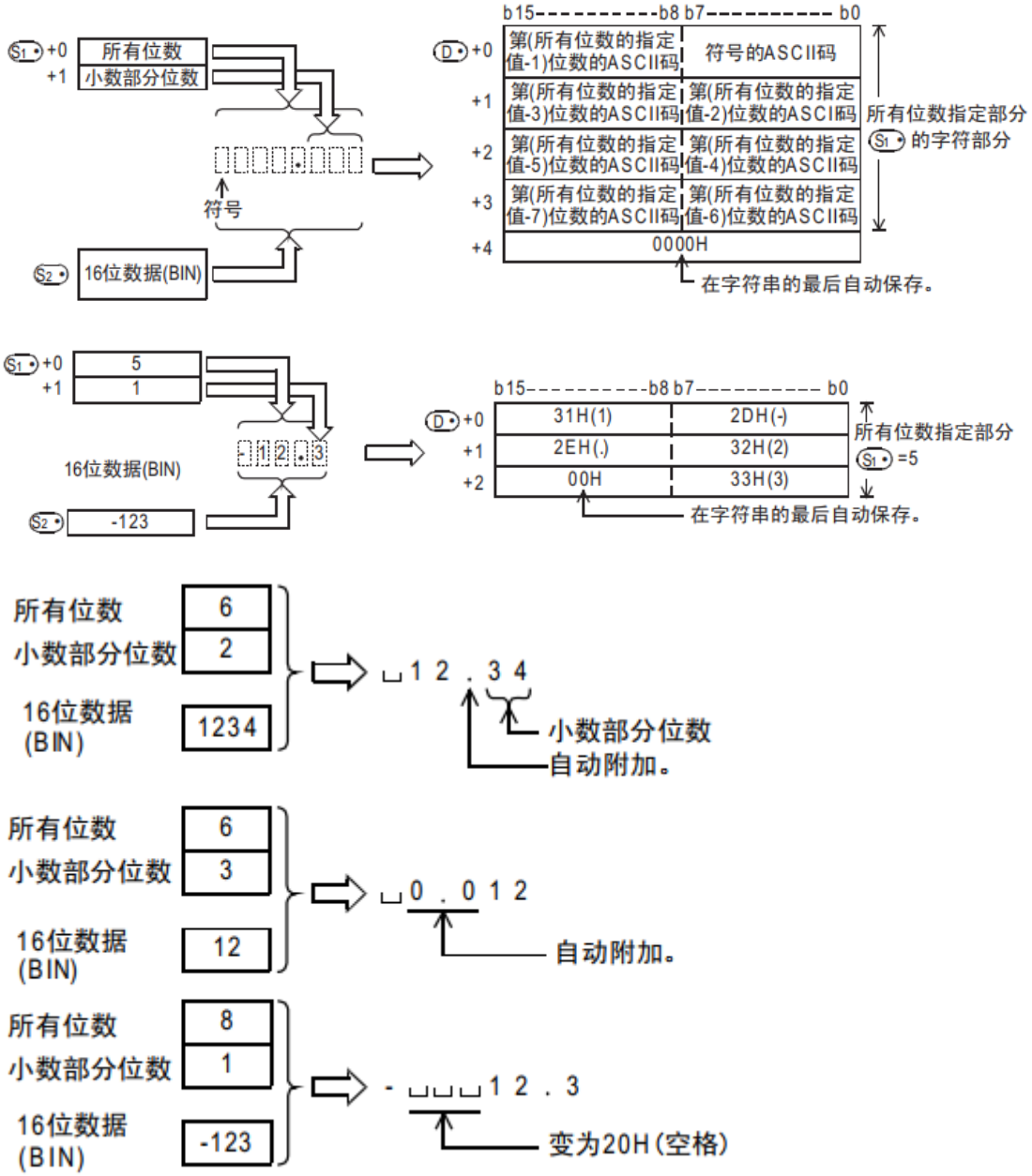
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*					
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*			*					

2、功能和动作说明

将 S2 的 16 位或 32 位数据(BIN)，在所有位数(S1)、小数点部分位数(S1 +1) 指定的位置中加上小数点后，转换成字符串，保存到 D 开始的软元件中。

以 16 位指令数据说明指令的动作：



转换后的字符串数据会如下所示地保存到 D 开始的软元件编号中：

• 在符号中，数据(BIN)S2 为正时保存“空格(20H)”，为负时保存“-(2DH)”。

• 在小数部分位数 S1+1 中设定为“0”以外的数字时，会自动在小数部分位数+1 位数的位置上加上小数点“. (2EH)”。小数部分位数 +1 为“0”时，不附加小数点。

• 与 S2 的 16 位或 32 位数据(BIN)的位数相比，S1+1 的小数部分位数较多时，会自动向右对齐，在左边附加“0(30H)”后进行转换。

• 如果除去小数点和符号以外，所有位数 S1 的位数多于 S2 的数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。此外，S2 的数据(BIN)的位数较多时，会错误。

• 在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8 位)中保存“00H”。

3、程序举例

略。

4、注意事项

以下情况会报错 6706：

• 所有位数 S1 为如下所示的范围以外时。

16 位运算：2~8；32 位运算 2~13。

• 所有位数 S1+1 为如下所示的范围以外时。

16 位运算：0~5；32 位运算 0~10。

• 所有位数 S1 和小数部分位数 S1+1 的关系非如下所示的范围时。

(所有位数-3)≧小数部分位数

• 所有位数 S1+符号、小数点的位数比 S2 的 BIN 数据位数少时。

• 保存字符串的 D 以后的软元件超出了相应的软元件范围时。

2.15.2 VAL/字符串→BIN 的转换指令

将字符串(ASCII 码)转换成 BIN 数据的指令。

1、指令格式

FNC201	指令所占程序步	指令分类	指令格式
16 位	7	VAL	[(D)VAL(P) S D1 D2]
		VALP	
32 位	13	DVAL	
		DVALP	

操作数说明

操作数	内容	类型
S	保存要转换成BIN数据的字符串的软元件起始编号	字符串
D1	保存已经转换的 BIN 数据位数的软元件起始编号	BIN16 位
D2	保存已经转换的 BIN 数据的软元件起始编号	BIN16/32 位

操作数可用元件类型

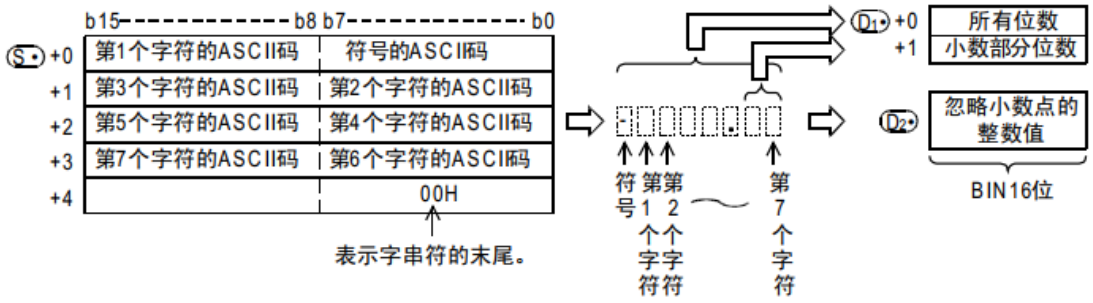
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D1												*	*	*	*			*					
D2									*	*	*	*	*	*	*			*					

2、功能和动作说明

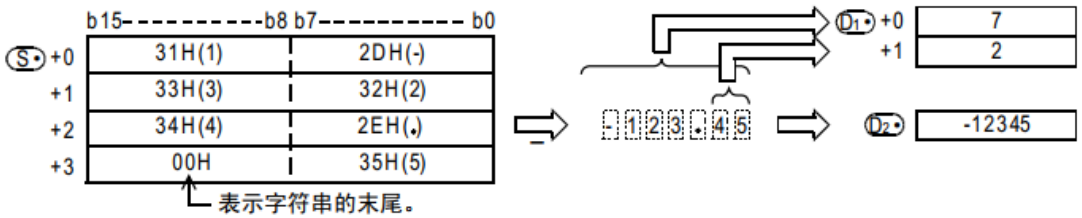
（1）16 位指令

将 S 开始的软元件中保存的字符串转换成 16 位数据（BIN），然后将所有位数保存到 D1 中、将小数部分位数保存到 D1+1 中、将 BIN 数据保存到 D2 中。
从字符串转换成 BIN 时，以字节为单位将 S 开始到保存“00H” 的软元件编号为止的数据，作为字符串进行处理。

- D1 中保存所有位数。所有位数，就是所有的字符数(包括数字、符号、小数点)。
- D1 +1 中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。
- 在 D2 中，无视小数点，将字符串转换成 16 位或 32 位的数据(BIN)。但是，在 S 开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略。



例如，在 S 开始的软元件中指定了“-123.45”的字符串时，D1、D2 中如下所示保存。



- 要转换的字符串数据范围限制：

	内容
所有字符位数	16 位指令：2~8 个字符 32 位指令：2~13 个字符

小数部分的字符位数	16 位指令：0~5 个字符，但是位数在（所有位数-3）以下 32 位指令：0~10 个字符，但是位数在（所有位数-3）以下
忽略小数点的数值范围	16 位指令：-32768~32767（例“123.45”-> “12345”） 32 位指令：-2147483648~2147483647（例“12345.678”-> “12345678”）

• 在要转换的字符中使用的字符限制：

		字符的种类
符号	正的数值	“空格（20H）”
	负的数值	“-（2DH）”
小数点		“.(2EH)”
数字		“0(30H)”～“9（39H）”

3、程序举例

略。

4、注意事项

要转换的字符串的数据范围限制和字符使用限制，不在限制范围内会报错 6706 运算错误。

2.15.3 $+/字符串的结合指令

连接字符串与字符串的指令。

1、指令格式

FNC202	指令所占程序步	指令分类	指令格式
16 位	7	$+	[$+(P) S1 S2 D]
		$+P	

操作数说明

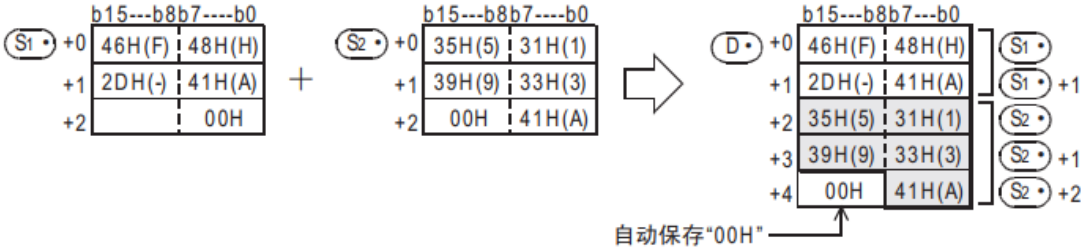
操作数	内容	类型
S1	字符串 1	字符串
S2	字符串 2	字符串
D	连接后的字符串	字符串

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*			*		
D1								*	*	*	*	*	*	*	*			*			*		
D2									*	*	*	*	*	*	*			*					

2、功能和动作说明

在 S1 开始的字符串数据后面连接 S2 开始的字符串数据，然后保存到 D 以后的软元件中。S1 和 S2 的字符串，是指以字节为单位从被指定的软元件开始到检测到第一个[00H]的位置为止的数据。



字符串的结合，就是指忽略 S1 中表示指定字符串末尾的“00H”，接着 S1 的最后字符连接 S2 中指定的字符串。此外，执行字符串的结合后会自动将“00H”附加在最后。

- 连接后的字符数为奇数时，在保存最后字符的软元件的高字节中保存“0000H”。
- 连接后的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”。

3、程序举例

略。

4、注意事项

（1）直接指定字符串时，可以指定(输入)的字符数最多为 32 个字符。 但是 S1 和 S2 中被指定了字软元件时，字符数没有限制。

（2）S1、S2 中任何一个的值从 00H 开始时，在 D 中保存“0000H”。

（3）以下情况报错 6706：

- D 中指定的软元件编号开始的软元件数，比保存所有已经结合的字符串所需的软元件数量更少时。(在所有的字符串和最终字符后面不能保存[00H])
- 当 S1 和 S2 中指定的保存字符串的软元件和 D 中指定的保存字符串的软元件编号重复时。
- 当 S1 和 S2 指定的软元件开始的相应软元件范围中[00H]未被设定时。

2.15.4 LEN/检测出字符串的长度指令

检测出指定字符串的字符数(字节数)的指令。

1、指令格式

FNC203	指令所占程序步	指令分类	指令格式
16 位	5	LEN	-[LEN(P) S D]
		LENP	

操作数说明

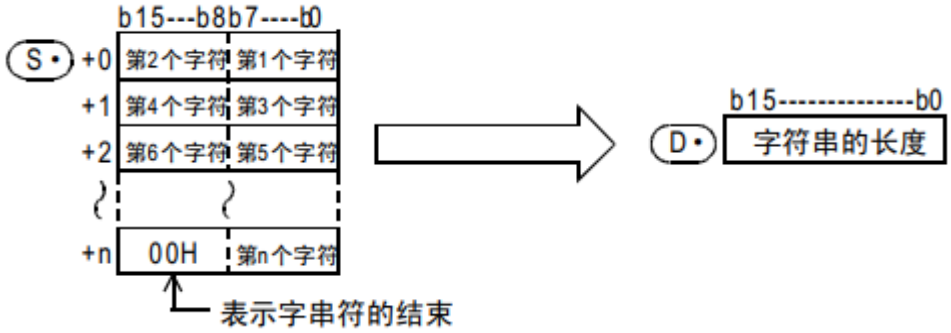
操作数	内容	类型
S	保存要检测出字符数的字符串的软元件起始编号	字符串
D	保存已检测出的字符串的长度(字节数)的软元件编号	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

检测出以 S 开头的字符串的长度，将字符串的长度保存到 D 中。S 以字节为单位，将从开始到第 1 个保存有[00H]的软元件编号为止的数据，作为字符串进行处理。



3、程序举例

略。

4、注意事项

以下情况会报错 6706：

- 在 S 指定的软元件编号开始的相应软元件范围内没有设定[00H]时。
- 检测出字符数超过 32768 个时。

2.15.5 RIGHT/从字符串的右侧开始取出指令

从指定的字符串的右侧取出指定字符数的字符的指令。

1、指令格式

FNC204	指令所占程序步	指令分类	指令格式
16 位	7	RIGHT RIGHTP	[RIGHT(P) S D n]

操作数说明

操作数	内容	类型
S	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
n	要取出的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*	*				*	*			

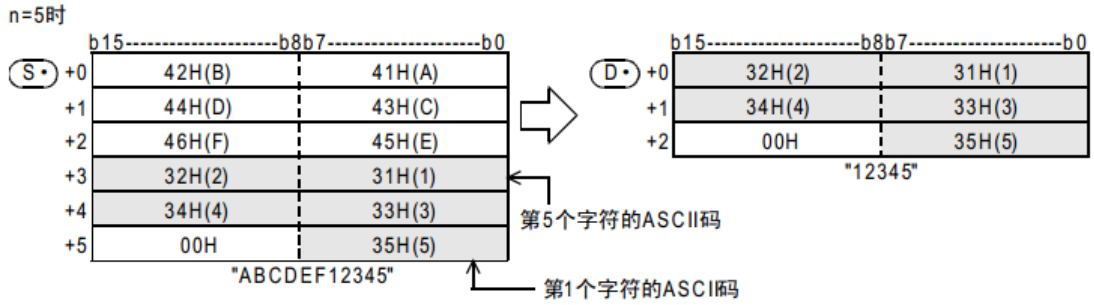
2、功能和动作说明

从 S 开始的软元件中保存的字符串数据的右侧(字符串的末尾)取出 n 个字符的数据，保存到 D 开始的软元件中。但是，n 中指定的字符数为“0”时，D 中保存 NULL 代码(0000H)。
此外，取出字符串时，会在最后自动附加“00H”。

- 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时， 在保存最后字符的软元件的下一个软元件中保“0000H”。

3、程序举例

将 S 为起始的字符串数据“ABCDEF12345”的右 5 位字符存到 D 中：



以 S 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第 1 个[00H]的位置为止的数据。

4、注意事项

以下情况会报错 6706：

- S 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- n 超出了 S 中指定的字符数时。
- D 中指定的软元件编号开始的软元件数，比保存已经取出的字符串(n 个字符)所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])
- n 为负值时。

2.15.6 LEFT/从字符串的左侧开始取出指令

从指定的字符串的左侧取出指定字符数的字符的指令。

1、指令格式

FNC205	指令所占程序步	指令分类	指令格式
16 位	7	LEFT	-[LEFT(P) S D n]
		LEFTP	

操作数说明

操作数	内容	类型
S	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
n	要取出的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

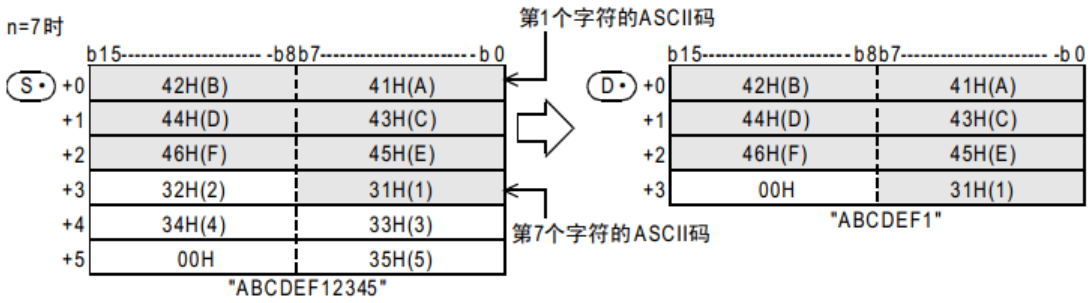
从 S 开始的软元件中保存的字符串数据的左侧(字符串的开头)取出 n 个字符的数据，保存到 D 开始的软元件中。但是，n 中指定的字符数为“0”时，D 中保存 NULL 代码(0000H)。

此外，取出字符串时，会在最后自动附加“00H”。

- 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。
- 要取出的字符数为偶数时， 在保存最后字符的软元件的下一个软元件中保“0000H”。

3、程序举例

将 S 为起始的字符串数据“ABCDEF12345”的左 7 位字符存到 D 中：



以 S 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第 1 个[00H]的位置为止的数据。

4、注意事项

以下情况会报错 6706：

- S 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- n 超出了 S 中指定的字符数时。
- D 中指定的软元件编号开始的软元件数，比保存已经取出的字符串(n 个字符)所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])
- n 为负值时。

2.15.7 MIDR/从字符串的任意取出指令

取出指定的字符串中任意位置上的字符串的指令。

1、指令格式

FNC206	指令所占程序步	指令分类	指令格式
16 位	7	MIDR	[MIDR(P) S1 D S2]
		MIDRP	

操作数说明

操作数	内容	类型
S1	保存字符串的软元件起始编号	字符串
D	保存被取出的字符串的软元件起始编号	字符串
S2	指定要取出的字符的起始位置以及字符数的软元件起始编号 S2：起始字符位置 S2+1：字符数	BIN16 位

操作数可用元件类型

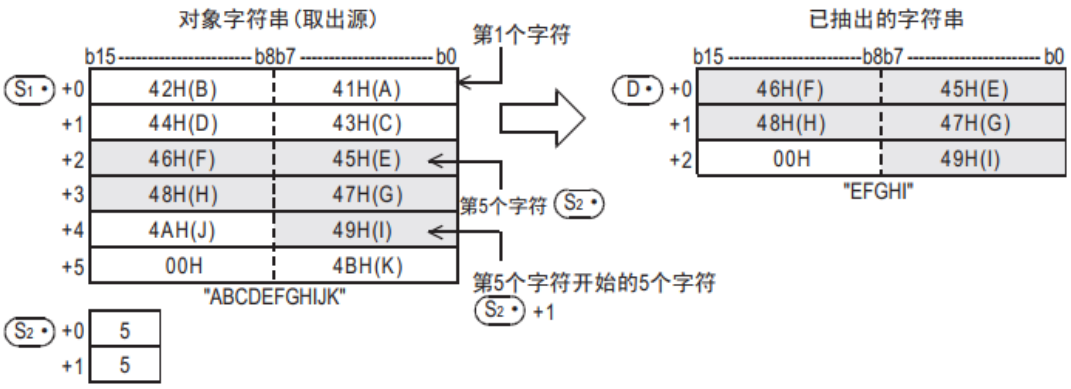
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
S2								*	*	*	*	*	*	*	*			*					

2、功能和动作说明

从 S1 开始的软元件中保存的字符串数据的左侧(字符串的开头)起第 S2 个字符开始，取出（S2+1）个字符的数据，保存到 D 开始的软元件中。此外，取出字符串时，会在最后自动附加“00H”。

• 要取出的字符数（S2+1）为奇数时，在保存最后字符的软元件的高字节中保存“00H”。

• 要取出的字符数(S2+1)为偶数时， 在保存最后字符的软元件的下一个软元件中保存“0000H”。



• S1 中指定的字符串(数据)，就是指从被指定的软元件开始到检测到第 1 个[00H]为止的数据。

• S2+1 中指定的要取出的字符数为“0”时不执行处理。

• S2+1 中指定的要取出的字符数为“-1”时，到 S1 指定的字符串的最终字符数据为止的内容被保存在 D 中。

3、程序举例

略。

4、注意事项

以下情况会报错 6706：

• S1 中指定的软元件开始的相应软元件范围内没有设定[00H]时。

• S2 的值超出了 S1 中指定的字符串的字符数时。

• D 开始的（S2+1）的字符数超出了 D 的软元件范围时。

• D 中指定的软元件编号开始的软元件数，比保存已取出的字符串 （S2+1）个字符所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])

• S2 为负值时。

• S2+1 为-2 以下的值时。

• S2+1 超出了 S1 的字符数时。

2.15.8 MIDW/从字符串的任意替换指令

用指定的字符串中任意位置上的字符串去替换指定的字符串的指令。

1、指令格式

FNC207	指令所占程序步	指令分类	指令格式
16 位	7	MIDW MIDWP	-[MIDW(P) S1 D S2]

操作数说明

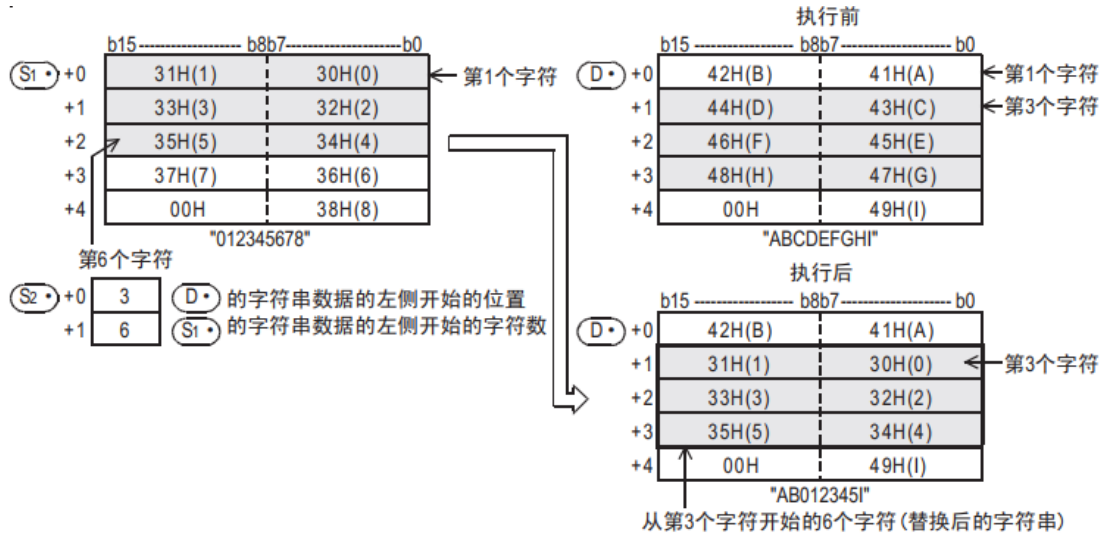
操作数	内容	类型
S1	保存字符串的软元件起始编号	字符串
D	保存替换后的字符串的软元件起始编号	字符串
S2	指定要替换的字符的起始位置以及字符数的软元件起始编号 S2：被替换的字符串的起始字符位置 S2+1：要替换的字符数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					
S2								*	*	*	*	*	*	*	*			*					

2、功能和动作说明

针对 S1 起始的软元件中保存的字符串数据，将其左侧(字符串的开头) 开始的字符数据，(字符数据数为 S2+1 中指定的字符数)保存到 D 起始软元件所保存的字符串中，保存位置从字符串数据左起数 S2 指定的位置开始。

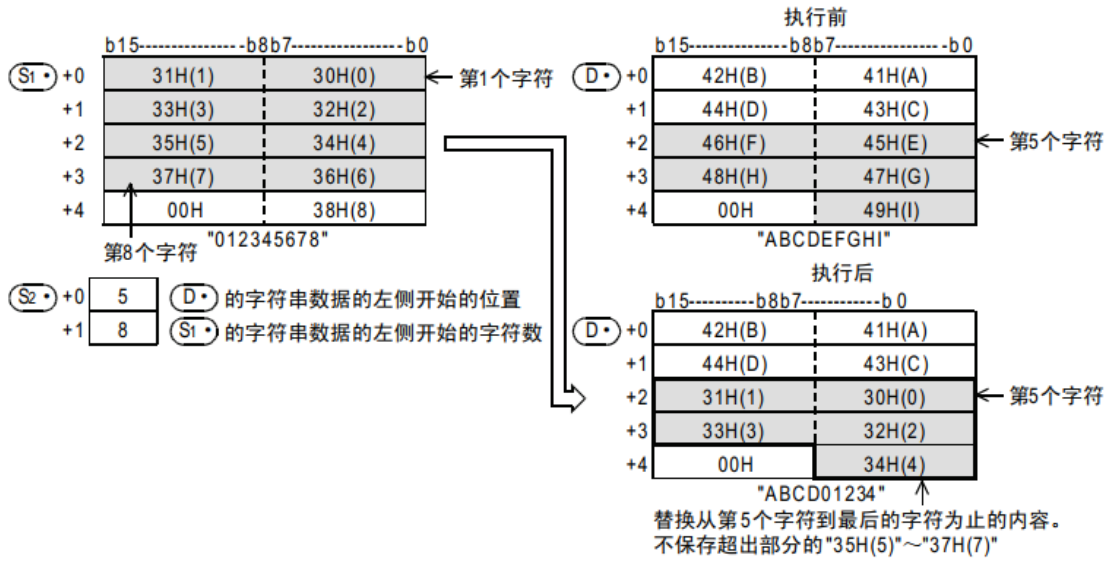


• S1 中 D 被指定的字符串(数据)，就是指从指定的软元件开始到检测开头到第 1 个[00H]为止的数据。

• S2+1 中指定的要替换的字符数为“00”时不执行处理。

• S2+1 中指定的字符数为“-1”时，到 S1 指定的最终字符数据为止的内容都被保存到 D 指定的软元件以后中。

• S2+1 中指定的要替换的字符数，一旦超出了 D 开始的字符串数据的最后字符时，则保存到最后字符为止的数据。



2、程序举例

略。

3、注意事项

以下情况报错 6706：

- S1 和 D 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
- S2 的值超出了 D 的字符数时。
- S2 为负值时。
- S2+1 为-2 以下的值时。
- S2+1 超出了 S1 的字符数时。

2.15.9 INSTR/字符串的检索指令

从指定的字符串中检索指定字符串的指令。

1、指令格式

FNC208	指令所占程序步	指令分类	指令格式
16 位	9	INSTR	-[INSTR(P) S1 S2 D n]
		INSTRP	

操作数说明

操作数	内容	类型

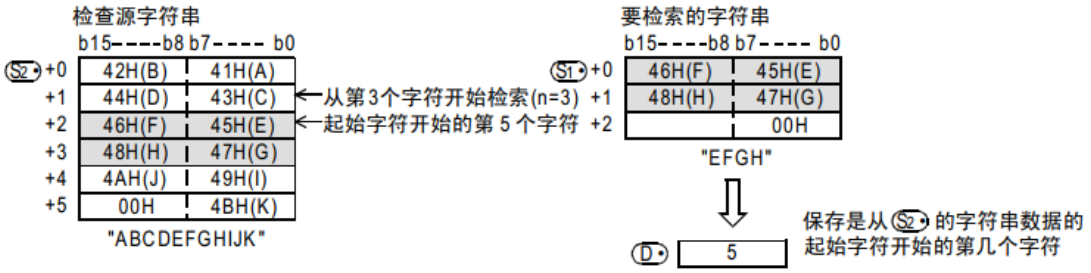
S1	保存要检索的字符串的软元件起始编号	字符串
S2	保存检索源字符串的软元件起始编号	字符串
D	保存检索结果的软元件起始编号	BIN16 位
n	开始检索的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1												*	*	*	*			*				*	
S2												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

（1）从 S2 起始的软元件中保存的检索源字符串的左起(起始字符)第 n 个字符开始，检索与保存在 S1 起始的软元件中的字符串相同的字符串，然后将检索结果的字符串位置信息保存到 D 中。检索的结果，就是检索源字符串左起(起始字符)的第几个字符的字符位置信息。



（2）不存在一致的字符串时，在 D 中保存“0”。

（3）开始检索的位置 n 为负数或是“0”不执行处理。

（4）字符串 S1 可以直接指定字符串类型数据

3、程序举例

略。

4、注意事项

以下情况报错 6706：

- 开始检索的位置 n 超过了 S2 的字符数时。
- S1 开始的相应软元件的软元件范围内没有 00H(NULL)时。
- S2 开始的相应软元件的软元件范围内没有 00H(NULL)时。

2.15.10 $MOV/字符串的传送指令

传送字符串数据的指令。

1、指令格式

FNC209	指令所占程序步	指令分类	指令格式
16 位	5	$MOV	[$MOV(P) S D]
		$MOVP	

操作数说明

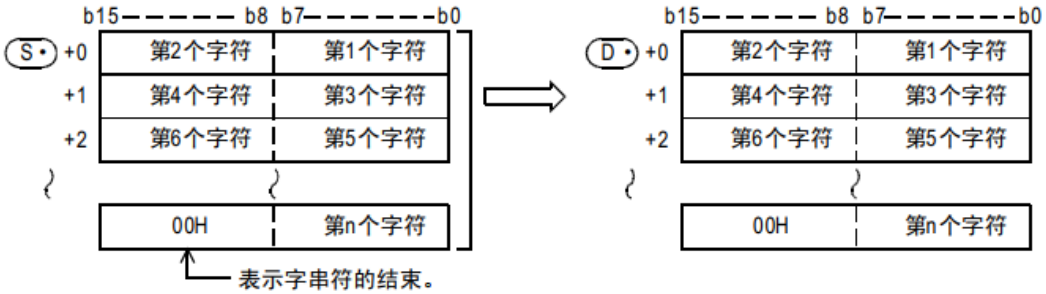
操作数	内容	类型
S	源字符串(最大 32 个字符)，	字符串
D	目的字符串	字符串

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*			*				*	
D									*	*	*	*	*	*	*			*					

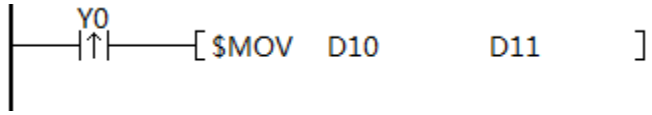
2、功能和动作说明

将 S 指定软元件中保存的字符串数据，传送到 D 起始的软元件中。在传送字符串过程中，从 S 指定的软元件编号开始，一直到之后的软元件中，其高字节或是低字节中包含“00H”的软元件为止，都一次进行传送。



3、程序举例

将 D10～D13 中保存的字符串 123456 传送到 D11～D14 中。



指令执行后，可看到传送后 D10 内的字符串没有变动。

显示格式： 16I 32I 32E ASC 10 16

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D10	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12
D11	0	0	1	1	0	0	1	0	0	0	1	1	0	0	0	1	12
D12	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	34
D13	0	0	1	1	0	1	1	0	0	0	1	1	0	1	0	1	56

保存要传送的字符串数据的软元件范围（S～S+n），以及保存被传送的字符串数据的软元件范围（D~D+n/2）使用重复时，也能执行传送。

4、注意事项

（1）S+n 的低字节中保存了“00H”时，D+n 的高字节、低字节中都保存“00H”。

（2）以下情况报错 6706：

• S 中指定的软元件编号～指定软元件的最终编号之间不存在“00H”时。
• 在 D 中指定的软元件编号～指定软元件的最终编号之间的点数中，不能保存所有指定的字符串时。

2.16 数据处理 3 指令

用于读取后入的数据、控制带进位的左右移指令。

2.16.1 FDEL/数据表的数据删除指令

删除数据表格中任意数据的指令。

1、指令格式

FNC210	指令所占程序步	指令分类	指令格式
16 位	7	FDEL	-[FDEL(P) S D n]
		FDELP	

操作数说明

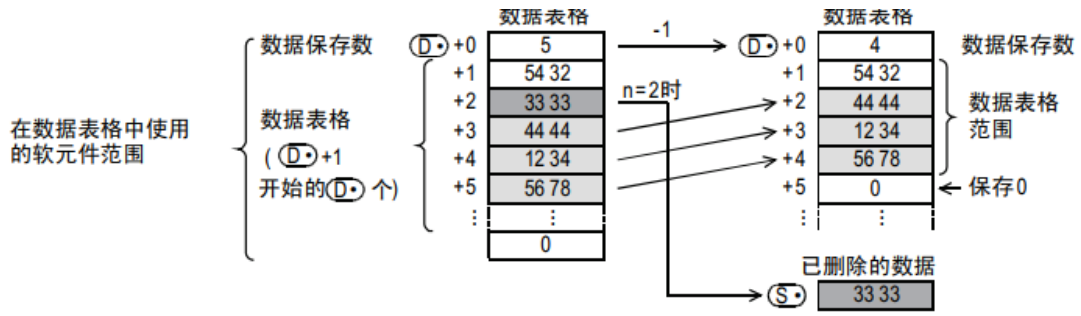
操作数	内容	类型
S	保存被删除的数据	BIN16 位
D	数据表格起始软元件	BIN16 位
n	要删除的数据在表格的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

删除数据表格(D 起始)的第 n 个数据，将删除的数据保存到 S 中。数据表格的第 n+1 个开始的数据逐个前移，数据保存数减 1。



3、程序举例

略。

4、注意事项

（1）数据表格的范围，为数据保存数 D 的下一个软元件(D+1)开始的 D 个。

（2）以下情况报错 6706：

- 要删除的数据的表格位置 n 比数据保存数更大时。
- n 的值超出了数据表格 D 的软元件范围时。
- 在 n≦0 的情况执行了指令时。
- 数据保存数 D 的值为 0 时。
- 数据表格的范围超出了相应的软元件范围时。

2.16.2 FINS/数据表的数据插入指令

在数据表格中的任意位置处插入数据的指令。

1、指令格式

FNC211	指令所占程序步	指令分类	指令格式
16 位	7	FINS	[FINS(P) S D n]
		FINSP	

操作数说明

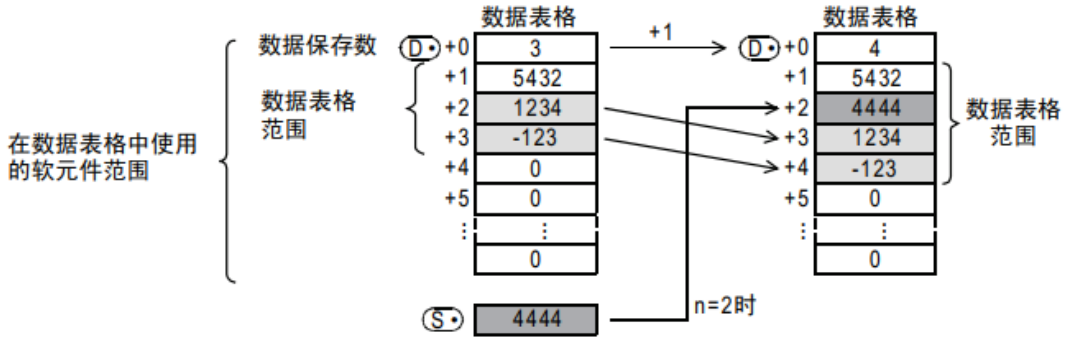
操作数	内容	类型
S	保存插入数据的软元件编号	BIN16 位
D	数据表格的起始软元件编号	BIN16 位
n	插入数据在表格的位置	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D												*	*	*	*			*					
n														*	*				*	*			

2、功能和动作说明

将 16 位数据 S 插入到数据表格(D 以后)的第 n 号中。数据表格的第 n 号以后的数据逐个后移，数据保存数加 1。



3、程序举例

略。

4、注意事项

（1）数据表格的范围，为数据保存数 D 的下一个软元件(D+1)开始的 D 个。

（2）以下情况报错 6706：

- 要插入的数据的表格位置 n 比数据保存数+1 更大时。
- n 的值超出了数据表格 D 的软元件范围时。
- 在 n≦0 的情况执行了指令时。
- 数据表格的范围超出了相应的软元件范围时。

2.16.3 POP/读取后入的数据指令

先入后出形式的数据读出。

1、指令格式

FNC212	指令所占程序步	指令分类	指令格式
16 位	7	POP	-[POP(P) S D n]
		POPP	

操作数说明

操作数	内容	类型

S	要读出的数据的起始字元件 (S 是数据点数，数据是从 S+1 开始的)	BIN16 位
D	读出的数据	BIN16 位
n	保存的数据点数+1（2≦n≦512）	BIN16 位

操作数可用元件类型

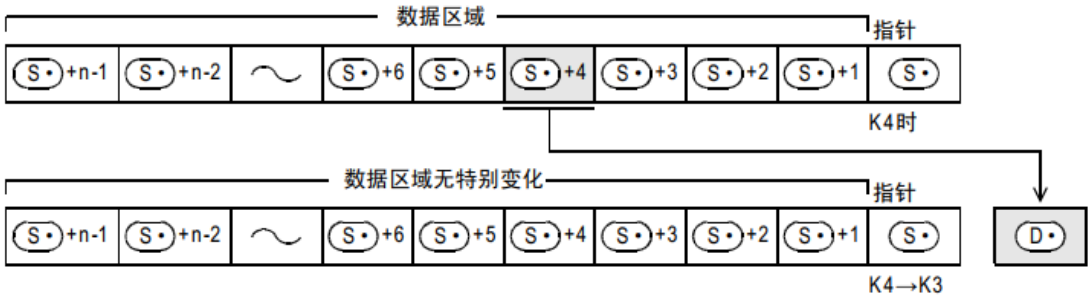
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S									*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					
n																			*	*			

2、功能和动作说明

先入后出控制用数据

值	内容
S	指针数据（被保存的数据个数）
S+1	数据区域（使用唯一写入指令 SFWR 被先入的数据）
S+2	
...	
S+n-2	
S+n-1	

对于（S～S+n-1）的字软元件，每次执行指令时，读出（S +指针数据 S）的软元件保存到 D 中（使用先入先出控制用的移位写入指令 (SFWR)写入的最后的数据读出到 D 中)，执行一次指针数据 S 的值减 1。



3、程序举例

略。

4、注意事项

（1）M8020 的动作，当指针 S=0 时，执行指令后变为 ON，且不处理指令。

（2）指针的当前值为 1 时，S 中被写入 0，M8020 置 ON。

（3）以下情况报错 6706：

- S ＞ n-1 时。
- S ＜ 0 时。

2.16.4 SFR/16 位数据 n 位右移（带进位）指令

使字软元件中的 16 位向右移动 n 位的指令。

1、指令格式

FNC213	指令所占程序步	指令分类	指令格式
16 位	5	SFR	[SFR(P) D n]
		SFRP	

操作数说明

操作数	内容	类型
D	保存要移动的数据的起始字软元件编号	BIN16 位
n	移动的次数 0≦n≦15	BIN16 位

操作数可用元件类型

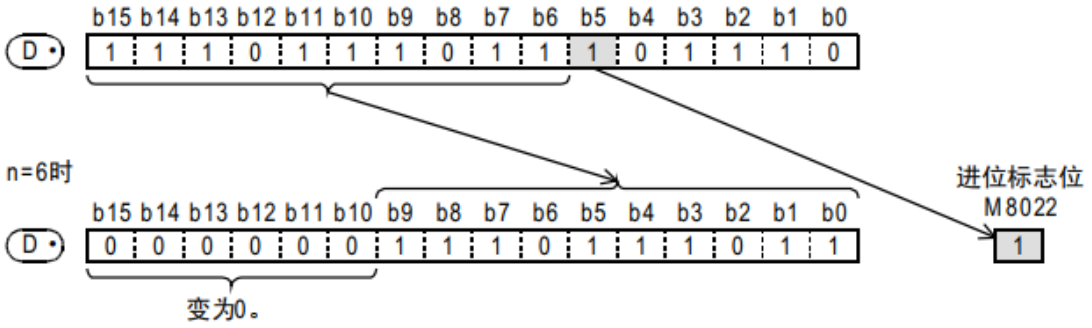
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n								*	*	*	*	*	*	*	*	*	*		*	*			

2、功能和动作说明

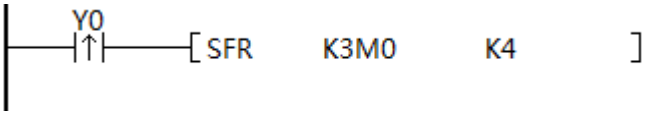
（1）字软元件 D 中的 16 位右移 n 位。n 指定 0～15 的数字。当 n 中指定了 16 以上的数值时，根据 n/16 的余数移动。例如，如 n=18 时，18/16=1 余 2，所以右移 2 位。

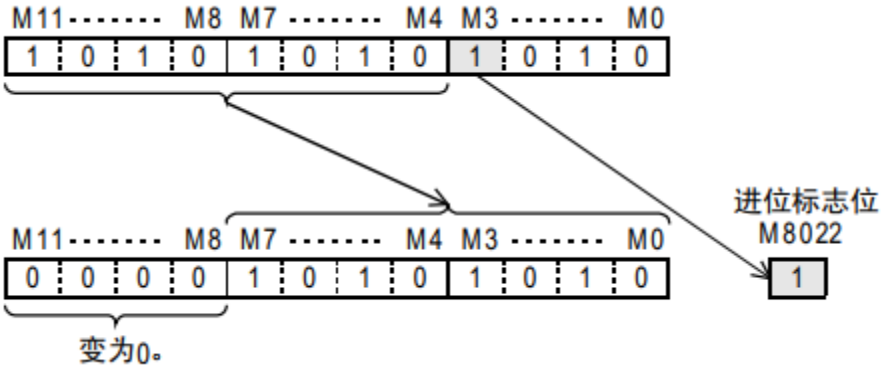
（2） 将字软元件 D 中的第 n 位(n-1 位)的 ON(1)/OFF(0)状态转移到进位标志 M8022 中。

（3） 最高位开始的 n 位变为 0。



（4）通过位数指定来指定位软元件时，将指定位数（4*K 的系数）的数据进行移位。





3、程序举例

略。

4、注意事项

（1）M8022 进位标志的动作，根据移动 n-1 的状态 ON/OFF。

（2）n 为负值时报错 6706。

2.16.5 SFL/16 位数据 n 位左移（带进位）指令

使字软元件中的 16 位向左移动 n 位的指令

1、指令格式

FNC214	指令所占程序步	指令分类	指令格式
16 位	5	SFL	-[SFL(P) D n]
		SFLP	

操作数说明

操作数	内容	类型
D	保存要移动的数据的起始字软元件编号	BIN16 位
n	移动的次数 0≦n≦15	BIN16 位

操作数可用元件类型

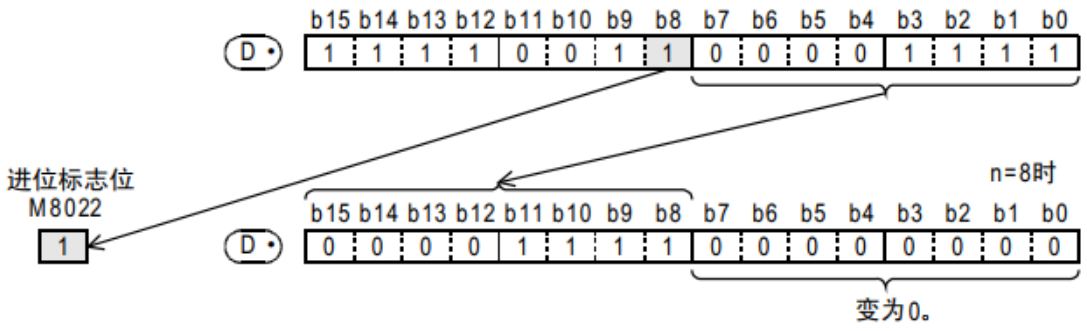
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D									*	*	*	*	*	*	*	*	*	*					
n								*	*	*	*	*	*	*	*	*	*		*	*			

2、功能和动作说明

（1）字软元件 D 中的 16 位左移 n 位。n 指定 0～15 的数字。当 n 中指定了 16 以上的数值时，根据 n/16 的余数移动。例如，如 n=18 时，18/16=1 余 2，所以右移 2 位。

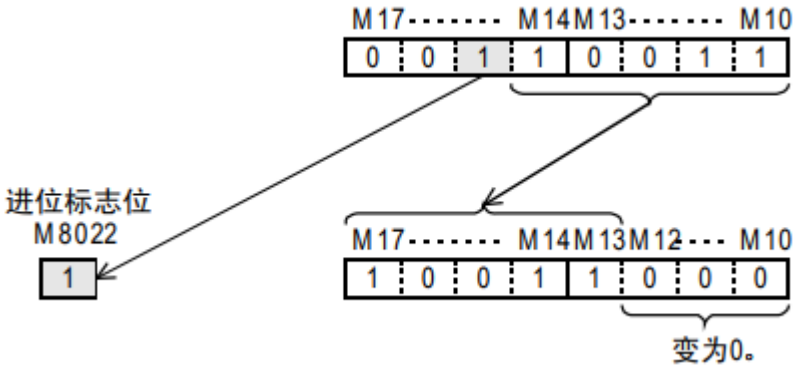
（2） 将字软元件 D 中的第 n+1 位(n 位)的 ON(1)/OFF(0)状态转移到进位标志 M8022 中。

（3） 最低位开始的 n 位变为 0。



（4）通过位数指定来指定位软元件时，将指定位数(4×K□)的数据进行移位。





3、程序举例

略。

4、注意事项

（1）M8022 进位标志的动作，根据移动 n 的状态 ON/OFF。

（2）n 为负值时报错 6706。

2.17 触点比较指令

提供使用触点符号进行数据比较的指令，在 UNISYS 中显示出来的是触点比较符号。

2.17.1 LD=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

FNC NO	指令位数	指令所占程序步	指令分类	指令格式
224	16 位	5	LD=	[LD(D)= S1 S2]
	32 位	9	LDD=	
225	16 位	5	LD>	[LD(D)> S1 S2]
	32 位	9	LDD>	

226	16 位	5	LD<	-[LD(D)< S1 S2]
	32 位	9	LDD<	
228	16 位	5	LD<>	-[LD(D)<> S1 S2]
	32 位	9	LDD<>	
229	16 位	5	LD<=	-[LD(D)<= S1 S2]
	32 位	9	LDD<=	
230	16 位	5	LDD>	-[LD(D)>= S1 S2]
	32 位	9	LDD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

2、功能和动作说明

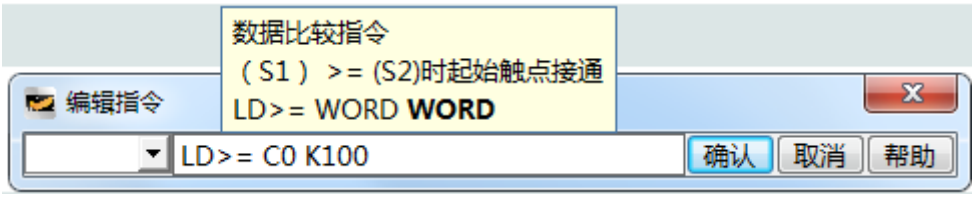
连接在母线上的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

FNC No	16 位指令	32 位指令	导通条件	不导通条件

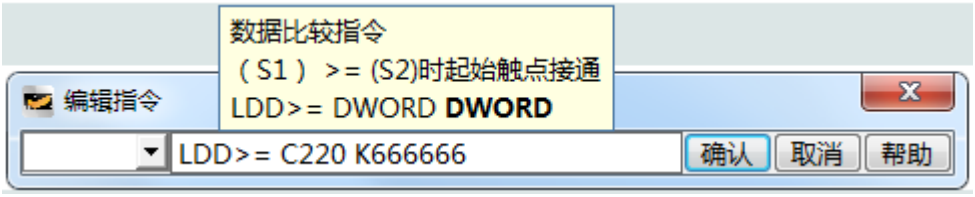
224	LD=	LDD=	S1=S2	S1≠S2
225	LD>	LDD>	S1>S2	S1≤S2
226	LD<	LDD<	S1<S2	S1≥S2
228	LD<>	LDD<>	S1≠S2	S1=S2
229	LD<=	LDD<=	S1≤S2	S1>S2
230	LD>=	LDD>=	S1≥S2	S1<S2

3、程序举例

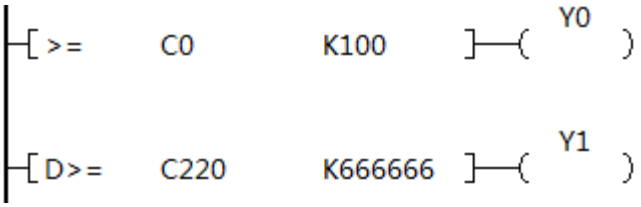
（1）16 位运算输入



（2）32 位运算输入



（3）实例



C0 值>=100 时，Y0 被驱动，C220 值>=66666 时 Y1 被驱动。

4、注意事项

（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

（2）注意指令的书写方式。

2.17.2 AND=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

FNC NO	指令位数	指令所占程序步	指令分类	指令格式
232	16 位	5	AND=	-[AND(D)= S1 S2]
	32 位	9	ANDD=	
233	16 位	5	AND>	-[AND(D)> S1 S2]
	32 位	9	ANDD>	
234	16 位	5	AND<	-[AND(D)< S1 S2]
	32 位	9	ANDD<	
236	16 位	5	AND<>	-[AND(D)<> S1 S2]
	32 位	9	ANDD<>	
237	16 位	5	AND<=	-[AND(D)<= S1 S2]
	32 位	9	ANDD<=	
238	16 位	5	AND>	-[AND(D)>= S1 S2]
	32 位	9	ANDD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

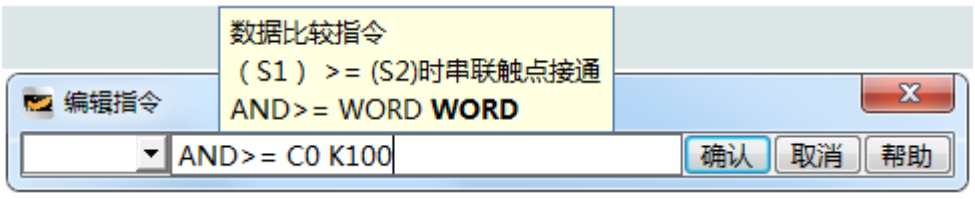
2、功能和动作说明

与其他触点串联的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

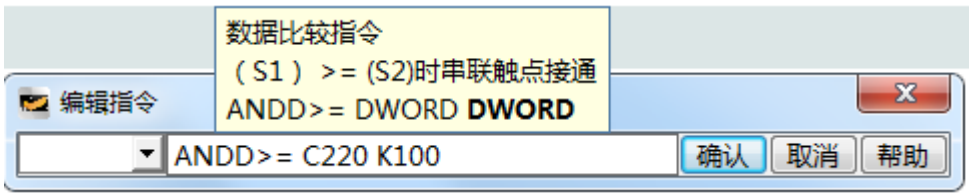
FNC No	16 位指令	32 位指令	导通条件	不导通条件
232	AND=	ANDD=	S1=S2	S1≠S2
233	AND>	ANDD>	S1>S2	S1≤S2
234	AND<	ANDD<	S1<S2	S1≥S2
236	AND<>	ANDD<>	S1≠S2	S1=S2
237	AND<=	ANDD<=	S1≤S2	S1>S2
238	AND>=	ANDD>=	S1≥S2	S1<S2

3、程序举例

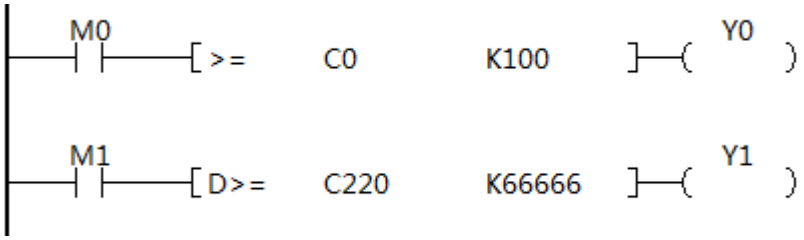
（1）16 位运算输入



（2）32 位运算输入



（3）实例



M0 为 ON 且 C0 值>=100 时，Y0 被驱动；M1 为 ON 且 C220 值>=66666 时 Y1 被驱动。

4、注意事项

（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

（2）注意指令的书写方式。

2.17.3 OR=、>、<、<>、<=、>=/触点比较指令

执行数值的比较，当条件满足时使触点置 ON 的触点比较运算开始的指令。

1、指令格式

FNC NO	指令位数	指令所占程序步	指令分类	指令格式
240	16 位	5	OR=	OR(D)= S1 S2
	32 位	9	ORD=	
241	16 位	5	ORD>	OR(D)> S1 S2
	32 位	9	ORD>	
242	16 位	5	OR<	OR(D)< S1 S2
	32 位	9	ORD<	
244	16 位	5	OR<>	OR(D)<> S1 S2
	32 位	9	ORD<>	
245	16 位	5	OR<=	OR(D)<= S1 S2
	32 位	9	ORD<=	

246	16 位	5	OR>	-[OR(D)>= S1 S2]
	32 位	9	ORD>=	

操作数说明

操作数	内容	类型
S1	比较数据 1	BIN16/32 位
S2	比较数据 2	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*	*	*	*	*	*			
S2								*	*	*	*	*	*	*	*	*	*	*	*	*			

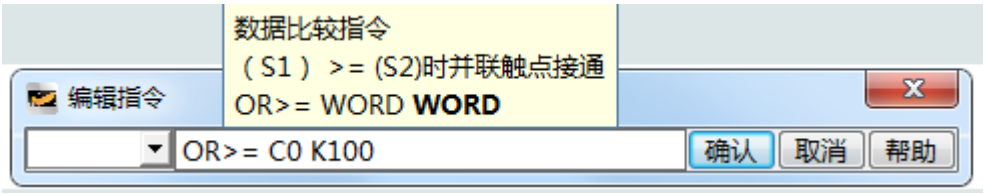
2、功能和动作说明

与其他触点并联的触点比较指令。对 S1、S2 的内容进行 BIN 比较，根据其结果来控制触点的导通或是不导通。

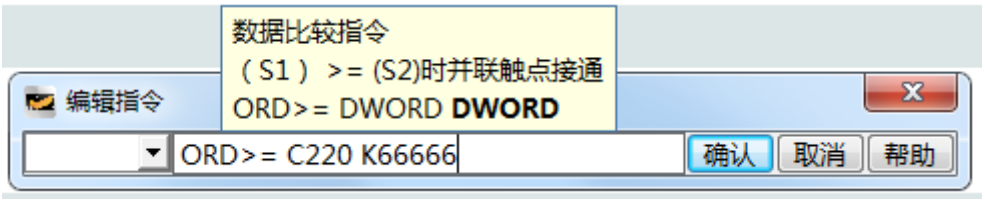
FNC No	16 位指令	32 位指令	导通条件	不导通条件
240	OR=	ORD=	S1=S2	S1≠S2
241	OR>	ORD>	S1>S2	S1≤S2
242	OR<	ORD<	S1<S2	S1≥S2
244	OR<>	ORD<>	S1≠S2	S1=S2
245	OR<=	ORD<=	S1≤S2	S1>S2
246	OR>=	ORD>=	S1≥S2	S1<S2

3、程序举例

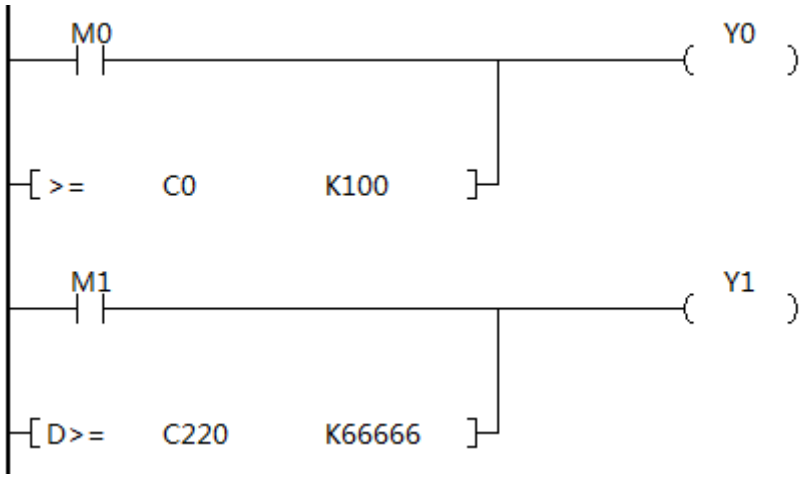
（1）16 位运算输入



（2）32 位运算输入



（3）实例



M0 为 ON 或 C0 值>=100 时，Y0 被驱动；M1 为 ON 或 C220 值>=66666 时 Y1 被驱动。

4、注意事项

（1）注意当比较源是 32 位计数器（C200~C255）的时候选择 32 位指令。

（2）注意指令的书写方式。

2.18 数据表处理指令

主要用于输入数据通过群数据时对输出数据的控制。

2.18.1 LIMIT/上下限位控制指令

设置输入数值的上限值/下限值然后输出的指令。

1、指令格式

FNC256	指令所占程序步	指令分类	指令格式
16 位	9	LIMIT	[(D)LIMIT(P) S1 S2 S3 D]
		LIMITP	
32 位	17	DLIMIT	
		DLIMITP	

操作数说明

操作数	内容	类型
S1	下限限位值	BIN16/32 位
S2	上限限位值	BIN16/32 位
S3	要通过上下限限位控制的输入值	BIN16/32 位
D	经过上下限限位控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

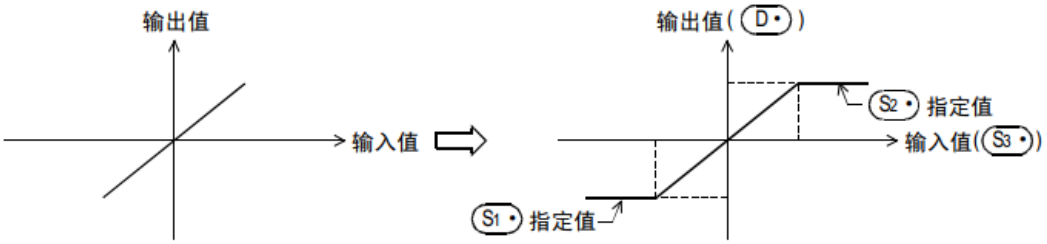
2、功能和动作说明

通过判断 S3 中指定的输入值，是否在 S1、S2 指定的上下限值的范围内，以此控制保存在 D 指定的软元件中的输出值。

下限值 > 输入值时，输出值 = 下限值；

上限值 < 输入值时，输出值 = 上限值；

下限值 ≤ 输入值 ≤ 上限值时，输出值 = 输入值。



3、程序举例

略。

4、注意事项

执行指令后，若设定的下限值 > 上限值时报错 6706。

2.18.2 BAND/死区控制指令

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令。

1、指令格式

FNC257	指令所占程序步	指令分类	指令格式
16 位	9	BAND	[(D)BAND(P) S1 S2 S3 D]
		BANDP	
32 位	17	DBAND	
		DBANDP	

操作数说明

操作数	内容	类型
S1	死区(无输出区域)的下限值	BIN16/32 位
S2	死区(无输出区域)的上限值	BIN16/32 位
S3	要通过死区控制的输入值	BIN16/32 位
D	经过死区控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

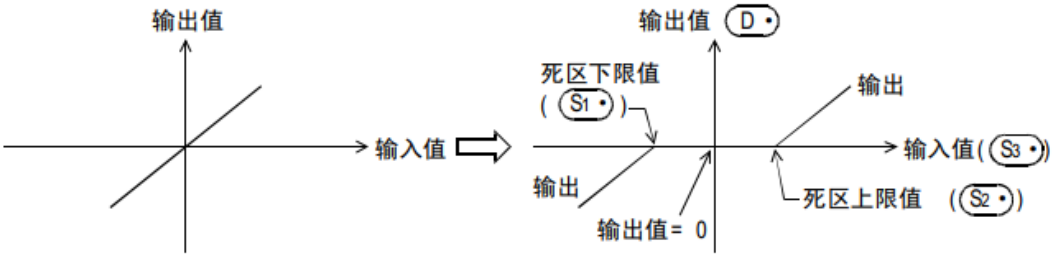
2、功能和动作说明

通过判断 S3 指定的输入值是否在 S1、S2 指定的死区范围内，以此来控制保存在 D 指定的软元件中的输出值。

下限值 > 输入值时，输出值 = 输入值 - 下限值；

上限值 < 输入值时，输出值 = 输入值 -上限值；

下限值 ≤ 输入值 ≤ 上限值时，输出值 = 0。



3、程序举例

略。

4、注意事项

（1）输出值溢出时，输出值 = 输入值 - 死区下限值

16 位运算：运算结果超出-32768~32767 时

例：S1 = 10，S3 = -32768，输出 D = -32768 - 10 = 8000H - AH = 7FF6H = 32758

32 位运算：运算结果超出-2147483648～2147483647 时

例：（S1+1,S1）= 1000，（S3+1，S3） = -2147483648，输出（D+1，D）= -2147483648 - 1000 = 80000000H - 000003E8H = 7FFFFC18H = 2147482648

（2）指令运行时，如果死区下限值 > 死区上限值时报错 6706

2.18.3 ZONE/区域控制指令

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令。

1、指令格式

FNC258	指令所占程序步	指令分类	指令格式
16 位	9	ZONE	-[(D)ZONE(P) S1 S2 S3 D]
		ZONEP	
32 位	17	DZONE	
		DZONEP	

操作数说明

操作数	内容	类型
S1	负偏差值	BIN16/32 位
S2	正偏差值	BIN16/32 位
S3	要通过区域控制的输入值	BIN16/32 位
D	通过区域控制的输出值	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2								*	*	*	*	*	*	*	*			*	*	*			
S3								*	*	*	*	*	*	*	*			*					
D									*	*	*	*	*	*	*			*					

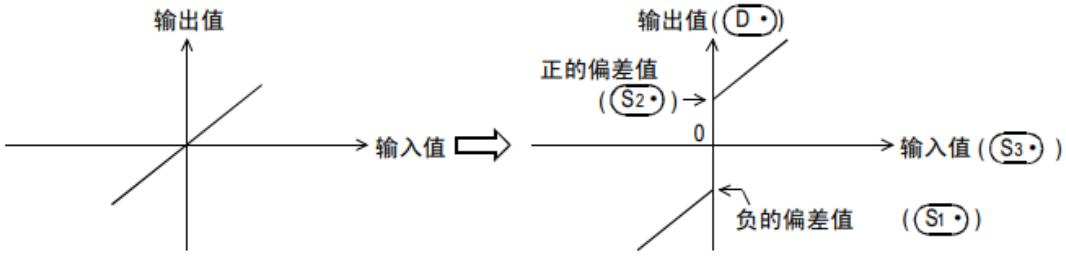
2、功能和动作说明

在 S3 指定的输入值上加上 S1 或 S2 指定的偏差值，然后保存到 D 指定的软元件编号中。

输入值 < 0 时，输出值 = 输入值 + 负的偏差值；

输入值 = 0 时，输出值 = 0；

输入值 > 0 时，输出值 = 输入值 + 正的偏差值。



3、程序举例

略。

4、注意事项

（1）输出值溢出时，输出值 = 输入值 + 负的偏差值

16 位运算：运算结果超出-32768~32767 时

例：S1 = -100，S3 = -32768，输出 D = -32768 + （-100） = 8000H + FF9CH = 7F9CH = 32668

32 位运算：运算结果超出-2147483648～2147483647 时

例：（S1+1,S1）= -1000，（S3+1，S3） = -2147483648，输出（D+1，D）= -2147483648 + （-1000） = 80000000H - FFFFFC18H = 7FFFFC18H = 2147482648

（2）指令运行时，如果负的偏差值 > 正的偏差值时报错 6706

2.18.4 SCL/定坐标（不同点坐标数据）指令

根据指定的数据表格组成的坐标系，对输入的 X 值执行定坐标指令后输出 Y 值。

1、指令格式

FNC259	指令所占程序步	指令分类	指令格式
16 位	7	SCL	[(D)SCL(P) S1 S2 D]
		SCLP	
32 位	13	DSCL	
		DSCLP	

操作数说明

操作数	内容	类型
S1	X 坐标点	BIN16/32 位
S2	组成坐标系的数据表格	BIN16/32 位
D	输出 Y 坐标点	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2														*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

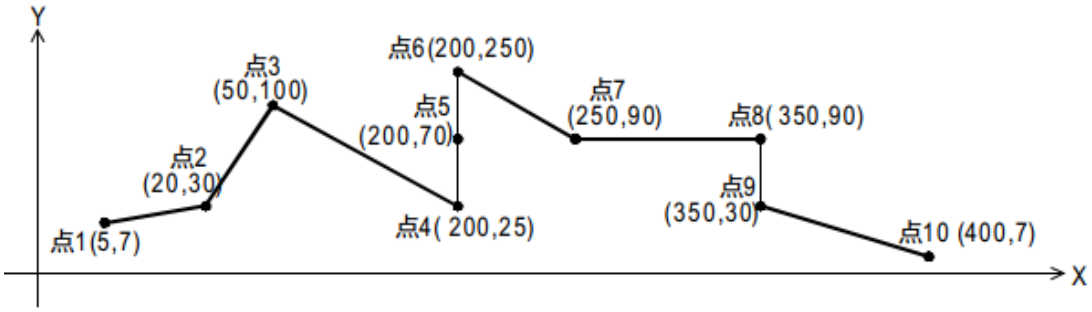
根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

3、程序举例

（1）定坐标用转换表格的设定

设定项目		设定数据表格的软元件分配	
		16 位指令运算	32 位指令运算
坐标点数		S2	（S2+1，S2）
点 1	X 坐标	S2+1	（S2+3，S2+2）
	Y 坐标	S2+2	（S2+5，S2+4）
点 2	X 坐标	S2+3	（S2+7，S2+6）
	Y 坐标	S2+4	（S2+9，S2+8）
···	···	···	···
点 n	X 坐标	S2+2n-1	（S2+4n-1，S2+4n-2）
	Y 坐标	S2+2n	（S2+4n+1，S2+4n）

（2）以 16 位运算为例，执行 32 位运算时按 32 位数据格式设定表格数据



设定项目		设定软元件及设定内容			备注
		S2 中指定了 R0 时		设定内容	
坐标点数		S2	R0	K10	
点 1	X 坐标	S2+1	R1	K5	
	Y 坐标	S2+2	R2	K7	
点 2	X 坐标	S2+3	R3	K20	
	Y 坐标	S2+4	R4	K30	
点 3	X 坐标	S2+5	R5	K50	
	Y 坐标	S2+6	R6	K100	
点 4	X 坐标	S2+7	R7	K200	如果象这样指定 3 点的坐标，则输出值为中间值。 这个例子中，将点 5 的 y 坐标指定为输出值(中间值)。 此外，3 点以上的 X 坐标相同时，也输出第 2 点的数值 数值
	Y 坐标	S2+8	R8	K25	
点 5	X 坐标	S2+9	R9	K200	
	Y 坐标	S2+10	R10	K70	
点 6	X 坐标	S2+11	R11	K200	
	Y 坐标	S2+12	R12	K250	
点 7	X 坐标	S2+13	R13	K250	如果象这样指定 2 点的坐标，则输出值取后一个点的 y 坐标值。 这个例子中，将点 9 的 y 坐标指定为输出值
	Y 坐标	S2+14	R14	K90	
点 8	X 坐标	S2+15	R15	K350	
	Y 坐标	S2+16	R16	K90	
点 9	X 坐标	S2+17	R17	K350	
	Y 坐标	S2+18	R18	K30	
点 10	X 坐标	S2+19	R19	K400	
	Y 坐标	S2+20	R20	K7	

4、注意事项

（1）当 X 坐标或 Y 坐标相同时，按上图表格中所写方式输出结果。

（2）以下情况报错 6706：

• 数据表格的 Xn 数据没有按照升序排列时。但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

• S1 在数据表格设定的范围以外时。

• 运算过程中的数值超出了 32 位数据的范围时。此时，请确认各点之间的距离没有超出 65535 以上。如果超出 65535 时，请缩短各点之间的距离。

2.18.5 DABIN/10 进制 ASCII→BIN 的转换指令

将以 10 进制数字的 ASCII 码(30H～39H)形式现实的数据转换成 BIN 数据的指令。

1、指令格式

FNC260	指令所占程序步	指令分类	指令格式
16 位	5	DABIN	[(D)DABIN(P) S D]
		DABINP	
32 位	9	DDABIN	
		DDABINP	

操作数说明

操作数	内容	类型
S	保存要转换成 BIN 值的数据(ASCII 码)的软元件起始编号	字符串
D	保存转换结果的软元件编号	BIN16/32 位

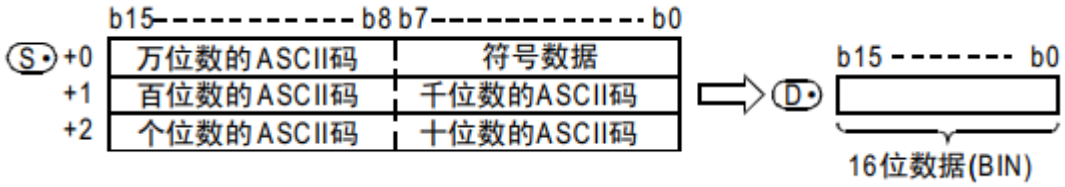
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*	*			*					
D									*	*	*	*	*	*	*	*	*	*					

2、功能和动作说明

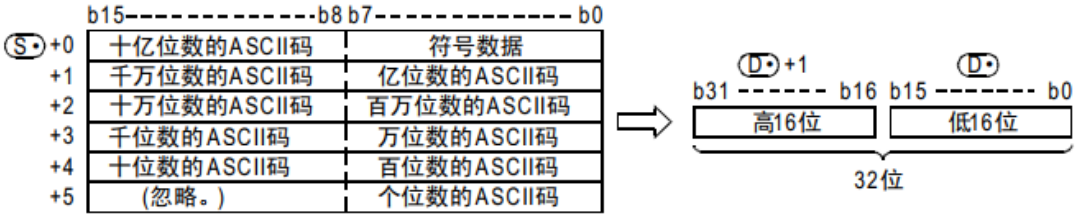
（1）16 位指令

将以 10 进制数字的 ASCII 码 (30H ～ 39H) 显示的 D~D+2 的数据转换成 16 位数据 (BIN) 后，保存到 D 中。



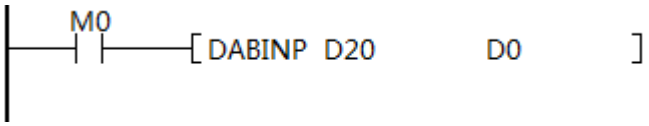
（2）32 位指令

将以 10 进制数字的 ASCII 码（30H ～ 39H）显示的D~D+5的数据转换成32位数据（BIN）后，保存到（D+1，D）中。



3、程序举例

当 M0 为 ON 时，将 D20～D22 中设定的符号以及 5 位 10 进制数字的 ASCII 码数据转换成 BIN 值后，保存到 D0 中的程序。



显示格式： 16I 32I 32E ASC 10 16

软元件	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0	值
D20	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	-
D21	0	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	2
D22	0	0	1	1	0	1	0	0	0	0	1	1	0	0	1	1	34

M0 置位后，D0 = -234

4、注意事项

（1）各位数的 ASCII 码位“20H”、“00H（NULL）”时，作为“30H”处理。

（2）以下情况报错 6706：

• 符号数据(S 的低位字节)中，要转换的数据为正时设定“20H(空格)”，为负时设定"2DH(-)"，设定其它以外值报错。

• S～S+2(5)的各位数的 ASCII 码为“30H”～“39H”、“20H(空格)”“00H(NULL)”以外的值时。

• S～S+2(5)的数值范围为下述的范围以外时：

16 位运算：-32768~32767；32 位运算：-2147483648~2147483647。

• S～S+2(5)超出软元件范围时。

2.18.6 BINDA/BIN→10 进制 ASCII 的转换指令

将 BIN 数据转换成 ASCII 码(30H～39H)的指令。

1、指令格式

FNC261	指令所占程序步	指令分类	指令格式
16 位	5	BINDA	-[(D)BINDA(P) S D]
		BINDAP	
32 位	9	DBINDA	
		DBINDAP	

操作数说明

操作数	内容	类型
S	保存要转换成 ASCII 码的 BIN 数据的软元件编号	BIN16/32 位
D	保存转换结果的软元件起始编号	字符串

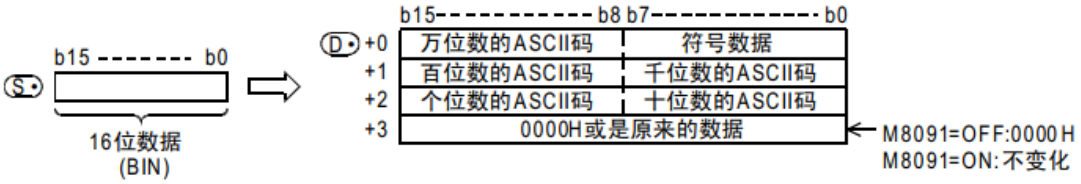
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S								*	*	*	*	*	*	*	*	*	*	*	*	*			
D												*	*	*	*			*					

2、功能和动作说明

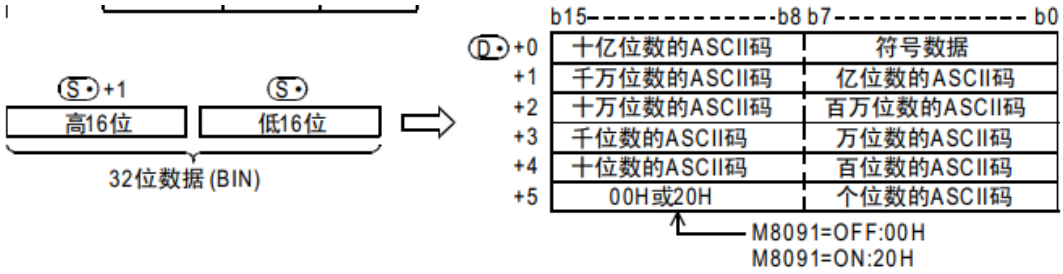
（1）16 位指令

将 S 的 16 位数据(BIN)按照 10 进制的各个位数转换成 ASCII 码(30H～39H)，然后保存到 D 中。



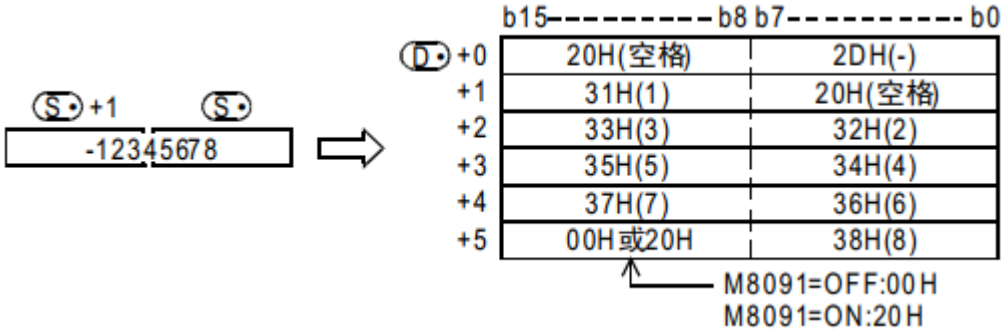
（2）32 位指令

将（S+1，S）的 32 位数据(BIN)按照 10 进制的各个位数转换成 ASCII 码(30H～39H)，然后保存到 D 开始的软元件中。



3、程序举例

例如，（S+1，S）为-12,345,678 时，D 开始的软元件中如下所示地进行保存。



4、注意事项

（1）相关软元件的动作：

软元件	名称	内容	
		16 位运算	32 位运算
M8091	输出字符数切换信号	M8091=OFF 时，D+3 为 0000H（NULL）。 M8091=ON 时，D+3 不变化。	M8091=OFF 时，D+5 高字节为 00H（NULL）。 M8091=ON 时，D+5 高字节为 20H（空格）。

要注意 16 位运算时 M8091 为 OFF 的情况，要占用 4 个字软元件。

（2）符号数据(D 的低位字节)中，要转换的数据为正时保存“20H(空格)”,为负时保存"2DH(-)"。

（3）当 S 中数据的总位数低于 D 能存储的总位数时，有效位数的左侧保存“20H（空格）”，如上面的例子。

（4）D 的占用点数不能超出改软元件的范围，超出报错 6706。

2.18.7 SCL2/定坐标 2（X/Y 坐标数据）指令

根据指定的数据表格组成的坐标系，对输入的 X 值执行定坐标指令后输出 Y 值。

1、指令格式

FNC269	指令所占程序步	指令分类	指令格式
16 位	7	SCL2	[(D)SCL2(P) S1 S2 D]
		SCL2P	
32 位	13	DSCL2	
		DSCL2P	

操作数说明

操作数	内容	类型
S1	X 坐标点	BIN16/32 位
S2	组成坐标系的数据表格	BIN16/32 位
D	输出 Y 坐标点	BIN16/32 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*	*	*			
S2														*	*			*					
D									*	*	*	*	*	*	*			*					

2、功能和动作说明

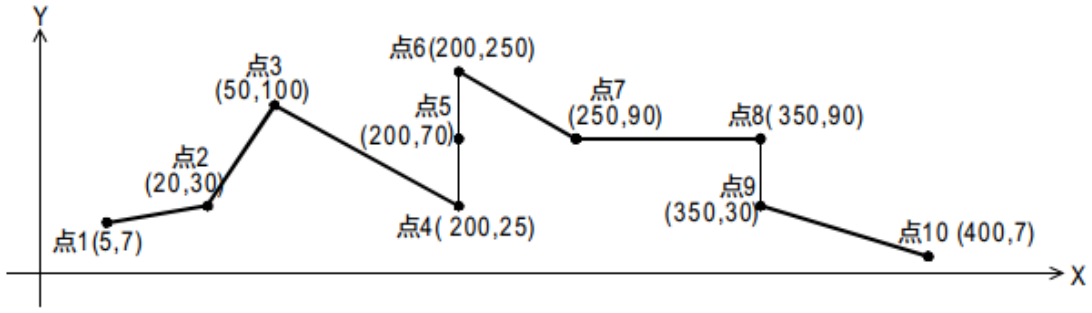
根据指定的转换特性，对 S1 指定的输入值执行定坐标，然后保存到 D 指定的软元件编号中。定坐标用的转换，是依据保存在 S2 指定的软元件开始的数据表格执行的。但是，输出数据不是整数值时，小数第 1 位四舍五入后输出。

3、程序举例

（1）定坐标用转换表格的设定

设定项目		设定数据表格的软元件分配	
		16 位指令运算	32 位指令运算
坐标点数		S2	（S2+1，S2）
X 坐标	点 1	S2+1	（S2+3，S2+2）
	点 2	S2+2	（S2+5，S2+4）
	···	···	···
	点 n（最后）	S2+n	（S2+2n+1，S2+2n）
Y 坐标	点 1	S2+n+1	（S2+2n+3，S2+2n+2）
	点 2	S2+n+2	（S2+2n+5，S2+2n+4）
	···	···	···
	点 n（最后）	S2+2n	（S2+4n+1，S2+4n）

（2）以 16 位运算为例，执行 32 位运算时按 32 位数据格式设定表格数据



设定项目		设定软元件及设定内容			备注
		S2 中指定了 R0 时		设定内容	
坐标点数		S2	R0	K10	
X 坐标	点 1	S2+1	R1	K5	
	点 2	S2+2	R2	K20	
	点 3	S2+3	R3	K50	
	点 4	S2+4	R4	K200	*1
	点 5	S2+5	R5	K200	
	点 6	S2+6	R6	K200	
	点 7	S2+7	R7	K250	
	点 8	S2+8	R8	K350	*2
	点 9	S2+9	R9	K350	
	点 10	S2+10	R10	K400	
Y 坐标	点 1	S2+11	R11	K7	
	点 2	S2+12	R12	K30	
	点 3	S2+13	R13	K100	
	点 4	S2+14	R14	K25	*1
	点 5	S2+15	R15	K70	
	点 6	S2+16	R16	K250	
	点 7	S2+17	R17	K90	
	点 8	S2+18	R18	K90	*2
	点 9	S2+19	R19	K30	
	点 10	S2+20	R20	K7	

*1. 如果像点 4、5、6 那样指定 3 点的坐标时则输出值为中间值。在这个例子中，将点 5 的 y 坐标指定为输出值(中间值)。此外，即使 3 点以上的 X 坐标相同时，输出第 2 点的数值。
*2. 如果像点 8、9 那样指定 2 点的坐标，则输出值取后一个点的 y 坐标的值。在这个例子中，将点 9 的 y 坐标指定为输出值。

4、注意事项

（1）当 X 坐标或 Y 坐标相同时，按例程中所写方式输出结果。

（2）以下情况报错 6706：

• 数据表格的 Xn 数据没有按照升序排列时。但是，由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。
• S1 在数据表格设定的范围以外时。
• 运算过程中的数值超出了 32 位数据的范围时。此时，请确认各点之间的距离没有超出 65535 以上。如果超出 65535 时，请缩短各点之间的距离。

2.19 扩展文件寄存器指令

ER 寄存器存储在外部 FLASH，对于一些想长期保存并且不想占用 PLC 的软元件资源的数据，可以使用 ER 寄存器来保存。

段编号与起始软元件编号对照表：

段编号	起始软元件编号	写入软元件范围	段编号	起始软元件编号	写入软元件范围
段 0	R0	ER0~ER2047	段 8	R16384	ER16384~ER18431
段 1	R2048	ER2048~ER4095	段 9	R18432	ER18432~ER20479
段 2	R4096	ER4096~ER6143	段 10	R20480	ER20480~ER22527
段 3	R6144	ER6144~ER8191	段 11	R22528	ER22528~ER24575
段 4	R8192	ER8192~ER10239	段 12	R24576	ER24576~ER26623
段 5	R10240	ER10240~ER12287	段 13	R26624	ER26624~ER28671
段 6	R12288	ER12288~ER14335	段 14	R28672	ER28672~ER30719
段 7	R14336	ER14336~ER16383	段 15	R30720	ER30720~ER32767

2.19.1 LOADR/读出扩展文件寄存器指令

将扩展文件寄存器 ER 的当前值读到 R 寄存器中。

1、指令格式

FNC290	指令所占程序步	指令分类	指令格式
16 位	5	LOADR	-[LOADR(P) S n]
		LOADRP	

操作数说明

操作数	内容	类型
S	保存数据的扩展寄存器的软元件编号(作为数据传送源的扩展文件寄存器与 S 的编号相同)	BIN16 位
n	读出点数（0≦n≦32767)	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n														*					*	*			

2、功能和动作说明

将存储在外部 FLASH 的 ER 寄存器读出到对应的 R 寄存器，以点为单位进行读出，最多可以读出 32768 点。

3、程序举例

略。

4、注意事项

当 n=0 时，按 n=32768 执行指令。

2.19.2 SAVER/成批写入扩展文件寄存器指令

将任意点数的扩展寄存器(R)的当前值 1 段(2048 点)写入扩展文件寄存器(ER)中时使用。

1、指令格式

FNC291	指令所占程序步	指令分类	指令格式
16 位	7	SAVER \	-[SAVER S n D]

操作数说明

操作数	内容	类型
S	保存数据的扩展寄存器的软元件编号，但是，只可以指定扩展寄存器的段的起始软元件编号	BIN16 位
n	每个周期写入点数（0≤n≤2048）	BIN16 位
D	记录已经写入的点数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			
D														*				*					

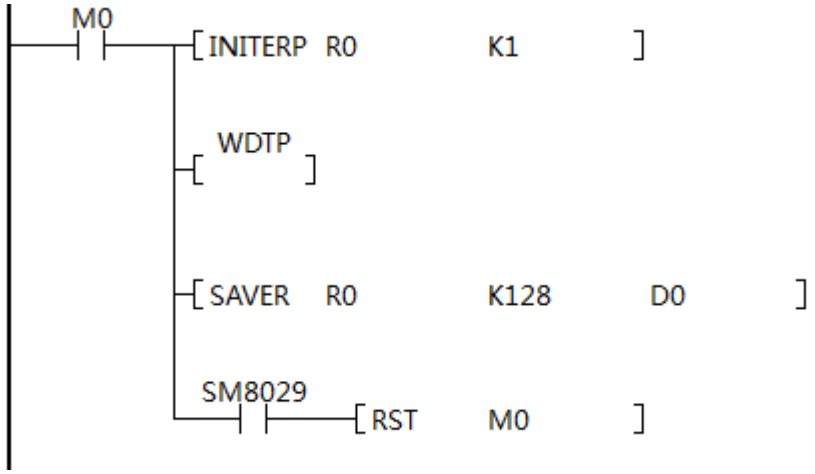
2、功能和动作说明

通过 2048/n 个运算周期(如有余数时，为+1 个扫描)将 S～S+2047 的扩展寄存器（R）的内容(当前值)写入具有相同编号的扩展文件寄存器(ER)中。指令执行过程中，在 D 中保存已经写入的点数。

- 每个运算周期写入 n 点。
- 以段为单位(2048 点)执行写入。各段的起始软元件编号对应 2.19 章节的表格。
- 在 n 中指定了 0 的值时，作为 n=2048 执行指令。
- 当 2048 点的写入(传送)结束时，指令执行结束，指令执行结束标志位 SM8029 变为 ON。

3、程序举例

当 M0 为 ON 时，通过每个运算周期写入 128 点，将设定数据用的扩展寄存器 R0～R2047(0 段)的变更内容，反映到扩展文件寄存器(ER)中，执行结束后 M0 会被复位。



①先执行 INITER 指令初始化扩展文件寄存器（ER）

②使用 SAVER 指令每个周期写入 128 点，将 R0~R2047 的值都会写入 ER0~ER2047。

4、注意事项

（1）SAVER 指令执行之前必须先执行 INITER（FNC 295）指令或者 INITR（FNC 292）指令进行 ER 数据的初始化，否则报错 6770。

（2）为防止 SAVER 指令执行写入时间过长造成看门狗超时，在此之前可刷新看门狗时间或更改比较大的时间保证能完成写入。

2.19.3 INITR/扩展寄存器的初始化指令

对扩展寄存器(R)和外部 FLASH 中的扩展文件寄存器(ER)进行初始化(HFFFF<K-1>)时使用本指令。

1、指令格式

FNC292	指令所占程序步	指令分类	指令格式
16 位	5	INITR	INITR(P) S n
		INITRP	

操作数说明

操作数	内容	类型
S	要初始化的扩展寄存器和扩展文件寄存器的软元	BIN16 位

	件编号，但是，仅可以指定扩展寄存器的段的起始软元件编号	
n	要初始化的扩展寄存器和扩展文件寄存器的段数	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			

2、功能和动作说明

对与 S 指定的扩展寄存器编号相同的扩展文件寄存器开头的 n 个段，执行初始化(写入初始值 HFFFF(K-1))，按段执行初始化。

3、程序举例

略。

4、注意事项

（1）INITR 指令同时初始化相同段中的 2048 个 R 寄存器和 ER 寄存器。

（2）多个段初始化时注意看门狗时间，防止超时。

（3）n 的范围为 1~16，为其他数字时报错 6706。

（4）要初始化的软元件编号超出 32767 时，到 R32767（ER32767）为止的软元件执行初始化，并报错 6706。

2.19.4 LOGR/登录到扩展文件寄存器指令

这个指令用于执行指定软元件的登录，并可以将已经登录的数据保存到扩展寄存器(R)以及扩展文件寄存器(ER)中。

1、指令格式

FNC293	指令所占程序步	指令分类	指令格式
16 位	11	LOGR	-[LOGR(P) S m D1 n D2]
		LOGRP	

操作数说明

操作数	内容	类型
S	执行登录的对象软元件起始编号	BIN16 位
m	登录点数 1≦m≦8000	BIN16 位
D1	登录中使用的软元件起始编号	BIN16 位
n	登录段数 1≦n≦16	BIN16 位
D2	已登录的数据数	BIN16 位

操作数可用元件类型

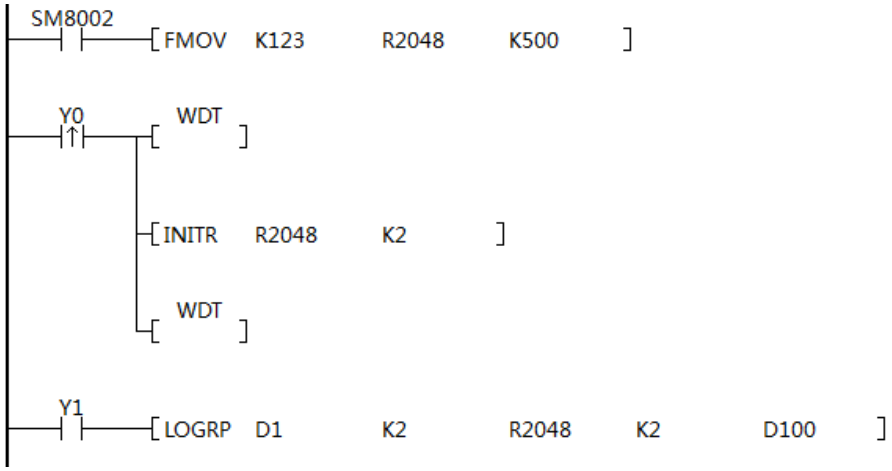
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S												*	*	*				*					
m														*					*	*			
D1															*								
n																			*	*			
D2														*									

2、功能和动作说明

在指令被驱动期间，到 D1 开头的扩展寄存器(R)以及扩展文件寄存器(ER)的 n 段的区域被填满为止，一直登录 S 开始的 m 点的软元件。此外，将保存的登录数据的数据数存放在 D2 中。

3、程序举例

每次 Y001 为 ON 时，在 R2048～R6143、ER2048～ER6143 的区域中登录 D1~D2 数据的程序。



第 1 次登陆后：R2048 = D1，R2049 = D2，ER2048 = D1，ER2049 = D2，D100 = 2；

第 2 次登陆后：R2050 = D1，R2051 = D2，ER2050 = D1，ER2051 = D2，D100 = 4；

...

第 2048 次登陆后：R6142 = D1，R6143 = D2，ER6142 = D1，ER6143 = D2，D100 = 4096。

至此，R2048～R6143 这两段区域以及 ER2048～ER6143 这 2 段区域的数据登录完毕。

4、注意事项

（1）登录数据之前，先使用 INITR 指令，以段为单位进行区域初始化，未初始化就执行该指令，报错 6770。INITR 中的段区域和段数要与 LOGR 指令的对应。

（2）LOGR 指令连续执行时，在每个运算中都执行登录，通过 1 次输入执行 1 次登录时，请使用脉冲执行型。

（3）当登录的点数超过可登录的最大点数时，登录到 R32767（ER32767）为止，报错 6706。

2.19.5 RWER/扩展文件寄存器的写入指令

将扩展寄存器(R)的任意点数的当前值写入扩展文件寄存器(ER)中时使用。

1、指令格式

FNC294	指令所占程序步	指令分类	指令格式
16 位	5	RWER	RWER(P) S n
		RWERP	

操作数说明

操作数	内容	类型

S	扩展寄存器段起始地址（只能是段起始地址）	BIN16 位
n	写入点数（0 或 2048 整数倍）	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n														*					*	*			

2、功能和动作说明

S 开始的 n 点扩展寄存器(R)的当前值写入相同编号的扩展文件寄存器(ER)中。ER 寄存器内容存储在外部 FLASH，因此无法监控，只能通过 LOADR 或 LOGR 指令读出，其中 R 的偏移只能是起始段位置，n 只能是 2048 的整数倍或 0。

3、程序举例

略。

4、注意事项

（1）传送点数较多时，注意看门狗超时。

（2）n=0 时，n=32768

（3）要传送的软元件的末尾软元件编号超出了 32767 时,对到末尾软元件编号的 R32767 为止的软元件执行写入。

2.19.6 INITER/扩展文件寄存器的初始化指令

执行 SAVER 指令前，对扩展文件寄存器(ER)进行初始化的情况时使用。

1、指令格式

FNC295	指令所占程序步	指令分类	指令格式
16 位	5	INITER	-[INITER(P) S n]
		INITERP	

操作数说明

操作数	内容	类型
S	与要执行初始化的扩展文件寄存器的软元件编号相同的扩展寄存器的软元件编号，但是，仅可以指定扩展寄存器的段的起始软元件编号。	BIN16 位
n	要初始化的扩展文件寄存器的段数（1≤n≤16）	BIN16 位

操作数可用元件类型

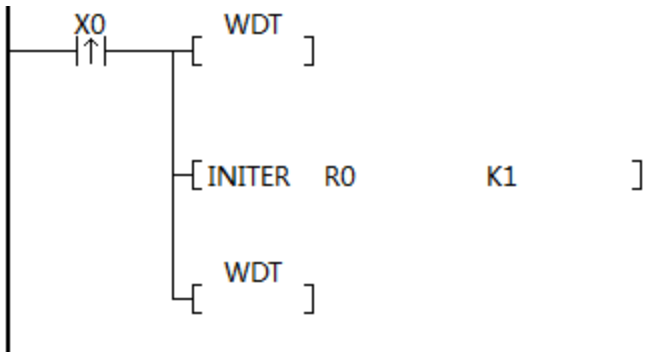
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S															*			*					
n																			*	*			

2、功能和动作说明

对与 S 编号相同的扩展文件寄存器(ER) 开头的 n 个段，执行初始化(写入初始值 HFFFF(K-1))，按段执行初始化。

3、程序举例

对段 0 中的扩展文件寄存器 ER0～ER2047 进行初始化的程序。



扩展文件寄存器(ER) 内容的变化如下，当然这个寄存器的值在上位机上是观察不到的，因为存储在外部 FLASH，只能通过对应的读写文件寄存器的命令读出来观察。

软元件编号	当前值	
	执行前	执行后
ER0	H1234	HFFFF

ER1	H5678	HFFFF
...		HFFFF
ER2047	HCDEF	HFFFF

4、注意事项

（1）对多段数据进行初始化时，注意看门狗超时。

（2）当 S 设定了 ER 段的起始软元件编号以外的数字时报错 6706。

（3）要初始化的软元件的软元件编号超出了 32767 时,初始化到 32767 为止。

2.20 方便指令

用最少的顺控程序实现复杂的控制的方便指令。

2.20.1 ABSD/凸轮控制绝对方式指令

根据计数器的当前值，产生多个输出模式的指令。

1、指令格式

FNC62	指令所占程序步	指令分类	指令格式
16 位	9	ABSD	[(D)ABSD(P) S1 S2 D n]
		ABSDP	
32 位	17	DABSD	
		DABSDP	

操作数说明

操作数	内容	类型
S1	数据表起始软元件	BIN16
S2	计数器	BIN16
D	输出起始位元件	位
n	输出的位元件点数/数据表数据组数（1≤n≤64）	BIN16

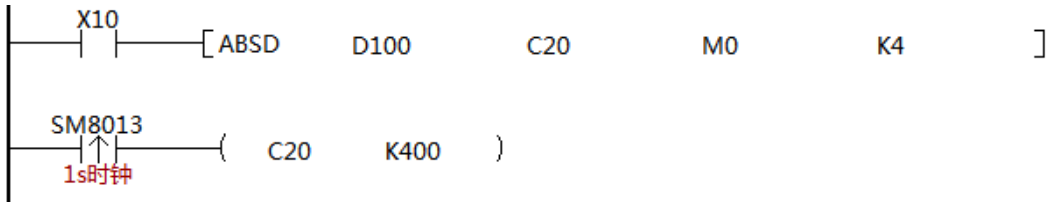
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
S2													*					*					
D		*	*			*	*											*					
n																			*	*			

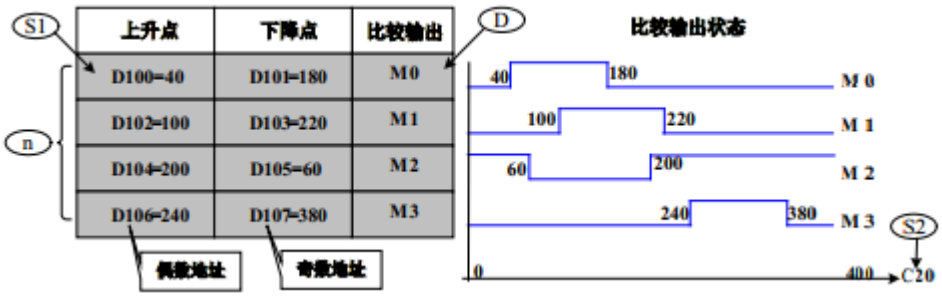
2、功能和动作说明

将 S1 为起始地址存储的数据表格数据与计数器 S2 中的当前数据进行比较，根据比较结果对 D 为起始地址的位元件进行 ON/OFF 控制。该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按绝对方式设置。

3、程序举例



当 X10 为 ON 时，执行结果如下：



4、注意事项

（1）该指令是主程序中扫描执行，比较结果受扫描时间的滞后影响。

（2）当 DABSD 指令中采用了高速计数器，比较输出 D 也受用户程序扫描时间的滞后影响，对于需要及时响应的应用，可以采用 HSZ 高速比较指令。

2.20.2 INCD/凸轮控制相对方式指令

使用连续的两个计数器，产生多个输出模式的指令。

1、指令格式

FNC63	指令所占程序步	指令分类	指令格式
16 位	9	INCD INCDP	-[INCD(P) S1 S2 D n]

操作数说明

操作数	内容	类型
S1	数据表起始软元件	BIN16
S2	计数器，占用 2 个，相邻的 S2+1 对应的计数器用于计算比较匹配后计数器复位的次数	BIN16
D	输出起始位元件	位
n	输出的位元件点数/数据表数据组数（1≤n≤64）	BIN16

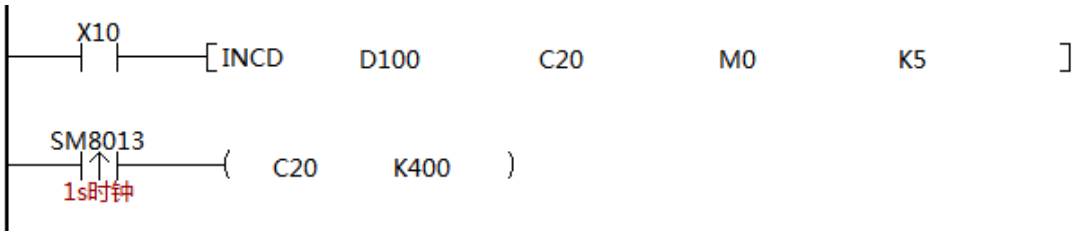
操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S1								*	*	*	*	*	*	*	*			*					
S2													*					*					
D		*	*			*	*											*					
n																			*	*			

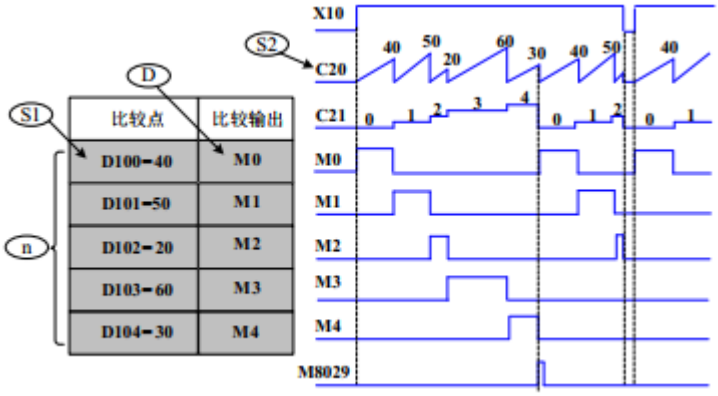
2、功能和动作说明

将 S1 为起始地址存储的数据表格数据与计数器 S2 中的当前数据进行比较，根据比较结果对 D 为起始地址的位元件进行 ON/OFF 控制。该指令完成的操作是多区段比较，用于实现凸轮控制，比较用的表格、计数器等均按增量方式设置。

3、程序举例



当 X10 为 ON 时，执行结果如下：



4、注意事项

（1）该指令是主程序中扫描执行，比较结果受扫描时间的滞后影响，对于需要及时响应的应用，可采用 HSZ 高速比较指令。

（2）n 的组数比较完毕后，指令执行完毕标志 SM8029 会置位。

2.20.3 ALT/交替输出指令

反转位软元件状态用的指令。

1、指令格式

FNC66	指令所占程序步	指令分类	指令格式
16 位	3	ALT	ALT(P) D
		ALTP	

操作数说明

操作数	内容	类型
D	交替输出的位软元件	位

操作数可用元件类型

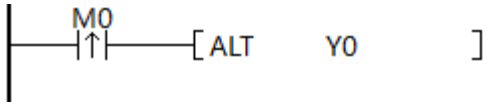
操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
D		*	*			*												*					

2、功能和动作说明

指令输入每次从 OFF 变为 ON 时，D 中指定的位软元件就执行状态的反转。

3、程序举例

每次 M0 的上升沿触发，Y0 的状态都会反转



4、注意事项

使用 ALT 指令编程时，每个运算周期都执行反转动作，使用触点指令来控制输出的状态反转。

2.20.4 SORT/数据排序指令

该指令是用于将数据行和数据列构成的数据表格，以指定的列为标准，按照行单位将数据表格重新升序排列。

1、指令格式

FNC69	指令所占程序步	指令分类	指令格式
16 位	11	SORT \	SORT S m1 m2 D n

操作数说明

操作数	内容	类型
S	源数据表格[占用 m1×m2 点]	BIN16 位
m1	行数[1～32]	
m2	列数[1～6]	

D	保存排序后的表[占用 m1×m2 点]	
n	作为排序标准的列编号[1～m2]	

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*								
m1														*	*				*	*			
m2														*	*				*	*			
D																							
n														*	*				*	*			

2、功能和动作说明

在 S 开始的(m1×m2)点的数据表格(排序前)中，以 n 列的群数据为标准，按照升序重新排列数据行,然后保存在从 D 开始的(m1×m2)点的数据表格(排序后)中。指令执行结束后 SM8029 会置位，可以通过此标志复位 SORT 前面的输入执行开关。

3、程序举例

在“n=K2(列号 2)”和“n=K3(列号 3)”的情况下，对如下所示的排序前的数据进行排序，动作如下所示。如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号。

（1）排序前数据：

列号 / 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	S	S+5	S+10	S+15
		1	150	45	20
	2	S+1	S+6	S+11	S+16
		2	180	50	40
	3	S+2	S+7	S+12	S+17

		3	160	70	30
	4	S+3	S+8	S+13	S+18
		4	100	20	8
	5	S+4	S+9	S+14	S+19
		5	150	50	45

（2）以 n=K2(列号 2)为基准执行指令时的排序结果

列号 / 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+5	D+10	D+15
		4	100	20	8
	2	D+1	D+6	D+11	D+16
		1	150	45	20
	3	D+2	D+7	D+12	D+17
		5	150	50	45
	4	D+3	D+8	D+13	D+18
		3	160	70	30
	5	D+4	D+9	D+14	D+19
		2	180	50	40

（3）以 n=K3(列号 3)为基准执行指令时的排序结果

列号 / 行号		群数 m2 个（m2=K4 时）			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=K5 的情况	1	D	D+5	D+10	D+15
		4	100	20	8
	2	D+1	D+6	D+11	D+16
		1	150	45	20

	3	D+2	D+7	D+12	D+17
		2	180	50	40
	4	D+3	D+8	D+13	D+18
		5	150	50	45
	5	D+4	D+9	D+14	D+19
		3	160	70	30

4、注意事项

（1）相关软元件的动作

SM8029：指令结束标志，排序结束时置位；

（2）以下操作会引起错误排序，请注意

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入 OFF 一次。
- 程序中仅可使用 1 次。
- 原来的数据和排序替换后的数据存放的位置，请错开，不要重叠。

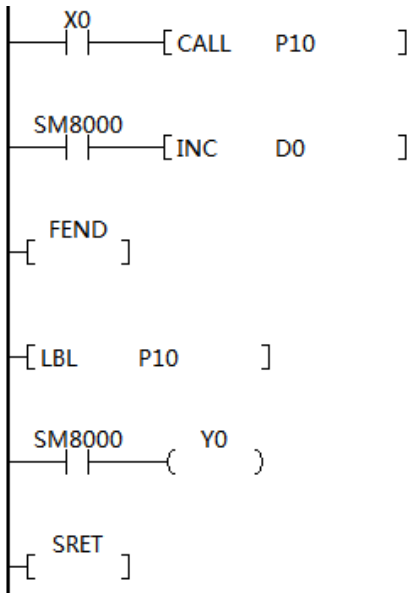
三、应用篇

3.1 子程序

动作说明：当指令输入为 ON 时，执行 CALL 指令，向标记 P 的步跳转。接着，执行标记 P 的子程序，执行 SRET 后，返回到 CALL 指令的下一步。子程序的调用返回程序有两种编写方式，第一种是指定 P 标签形式，第二种是直接插入子程序方式，举例说明如下。

子程序编程的两种方式：

（1）指定 P 标签



①CALL 后面写入要跳转的标签 P 编号

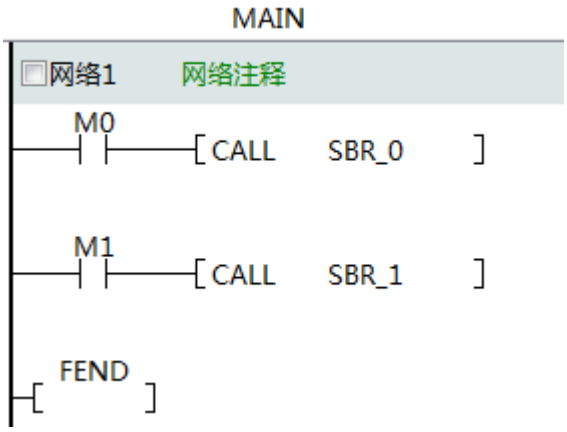
②用 LBL P 来引导 CALL 的跳转

（2）直接插入子程序

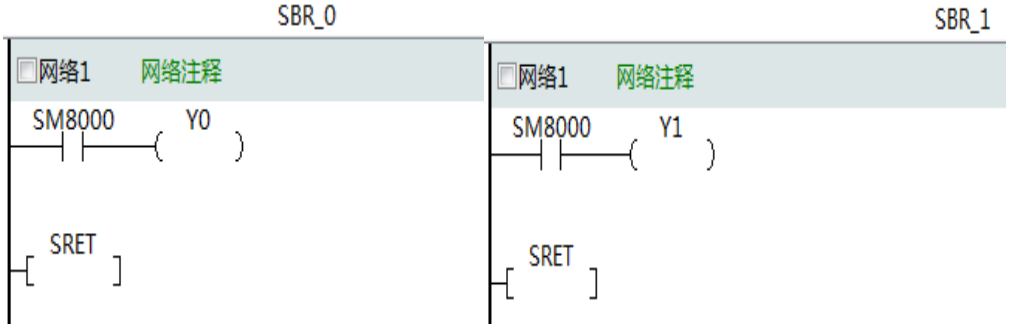
①右键程序块插入子程序



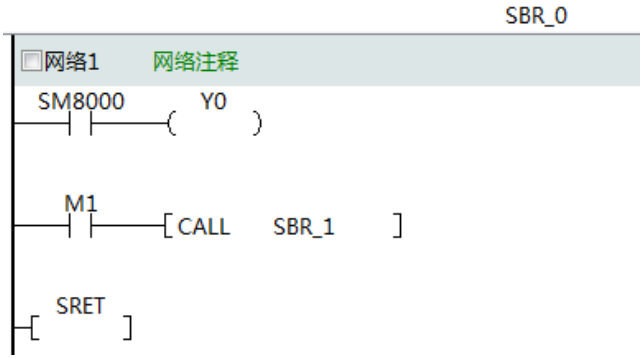
②CALL 后面写入子程序名称，其中,SBR 后面的标号与 P 标签对应



③子程序，带 SRET 子程序返回



④扩展：子程序嵌套，在子程序中添加其它子程序调用即可



注意事项：

（1）在主程序的最后用 FEND 指令编程；

（2）标签形式写子程序时，CALL 指令用的标记 P，必须在 FEND 指令后编程；

（3）最多允许 6 层嵌套，超出报 6634 停机错误；

（4）子程序没有 SRET 指令，XPLC 报 6637 停止运行。

关于流程类的梯形图 XPLC 在自检的时候就会把错误找出来并停机错误。

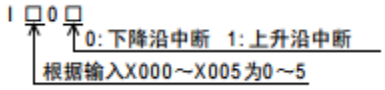
3.2 中断程序

动作说明：EI 是中断开放指令，DI 是中断禁止指令。中断源触发中断后，PLC 就跳转到其中断用指针编号指定的子程序执行。IRET 是中断返回指令，它是中断处理程序的结束指令，其作用是使 PLC 返回到被中断时的下一条指令继续执行。 若没有中断处理子程序，虽然触发了中断，但没有处理任何事情。

1、中断事件分类

（1）外部输入中断：6 点

外部中断指针标号如下：



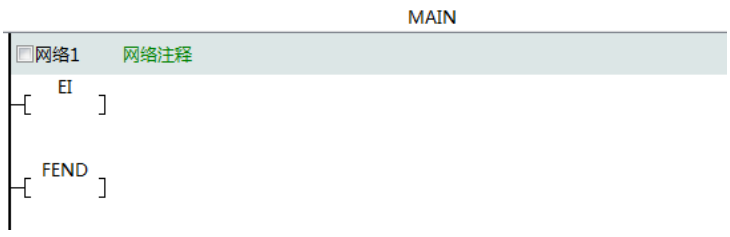
在不受可编程控制器运算周期的影响下，接收来自特定的输入编号的输入信号。触发该输入信号，执行中断子程序。由于输入中断可以处理比运算周期更短的信号，因此可在顺控过程中作为需要优先处理或者短时间脉冲处理控制时使用。

输入	输入中断用指针		禁止中断标志位
	上升沿中断	下降沿中断	
X00	I001	I000	SM8050
X01	I101	I100	SM8051
X02	I201	I200	SM8052
X03	I301	I300	SM8053
X04	I401	I400	SM8054
X05	I501	I500	SM8055

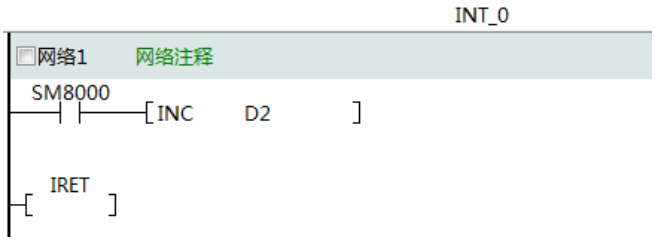
同一个梯形图中同一个输入的上升沿和下降沿都可以触发对应的外部中断。

程序举例:

主程序：

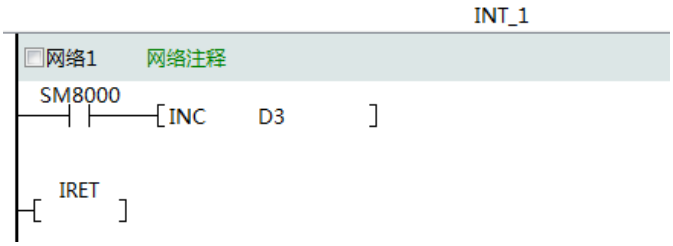


I000 中断子程序：



中断说明：当 X0 发生下降沿中断时，D2 计数立即加 1。

I001 中断子程序：



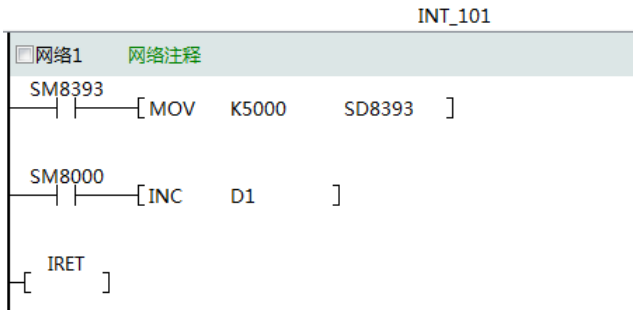
当 X0 发生上升沿中断时，D3 计数加立即 1。

I100 中断子程序：



当 X1 发生下降沿中断时，由于 SM8499 外部中断延时的作用，会有 SD8499 的延迟，这里是 5S，然后 D0 计数加 1。

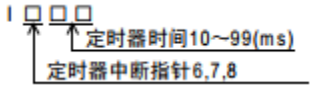
I101 中断子程序：



当 X1 发生上升沿中断时，由于 SM8499 外部中断延时的作用，会有 SD8499 的延迟，这里是 5S，5S 后 D1 计数加 1。

（2）定时器中断：3 点

定时器中断指针标号如下：



每隔指定的中断循环时间(10ms～99ms)，执行中断子程序。在可编程控制器的运算周期以外，需要循环中断处理的控制中使用。

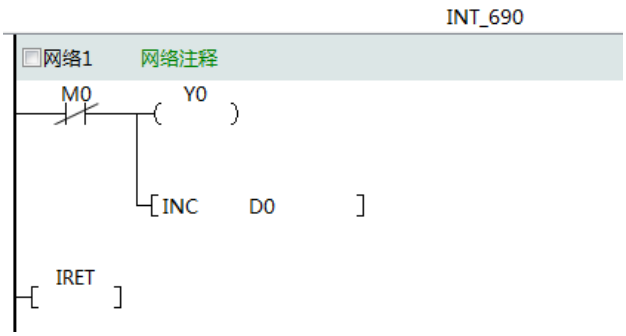
输入编号	中断周期（ms）	中断禁止标志位
I6XX	在指针名的 XX 中，输入 10~99 的整数。 例如:I610=每 10ms 的定时器中断	SM8056
I7XX		SM8057
I8XX		SM8058

程序举例：

主程序：

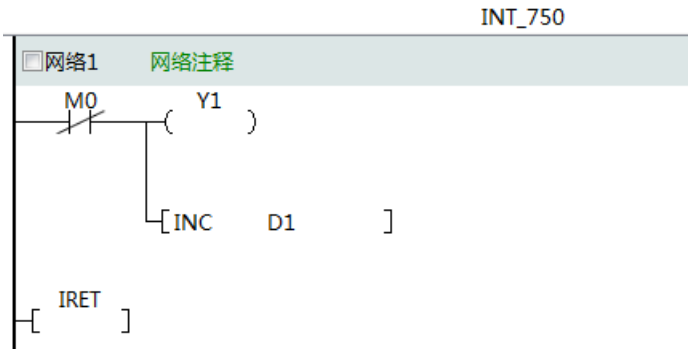


I690 中断子程序：



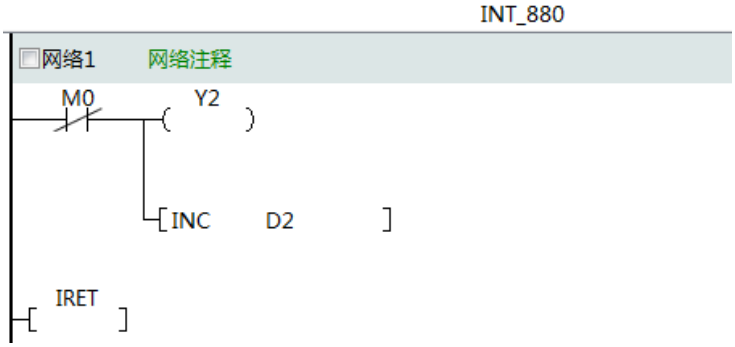
中断说明：定时器 1 计时 90ms 后，发生定时器中断，D0 计数立即加 1。

I750 中断子程序：



中断说明：定时器 2 计时 50ms 后，发生定时器中断，D1 计数立即加 1。

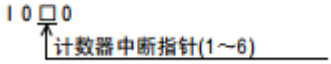
I880 中断子程序：



中断说明：定时器 3 计时 80ms 后，发生定时器中断，D2 计数立即加 1。

（3）计数器中断：6 点

计数器中断指针标号如下：

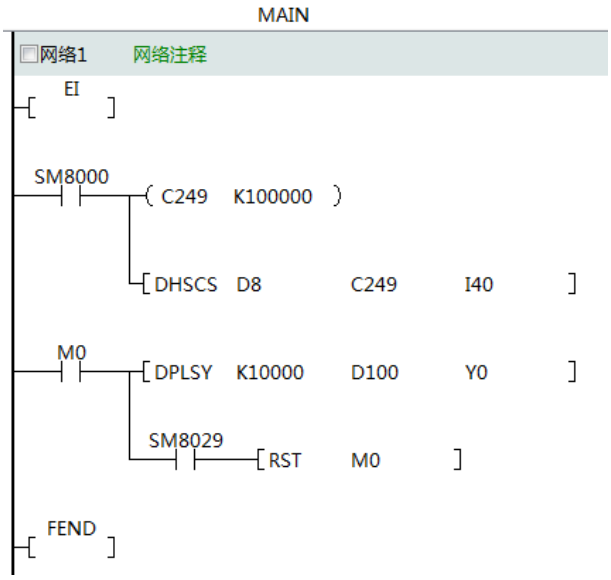


根据高速计数器用比较置位指令(DHSCS 指令)的比较结果，执行中断子程序。用于使用高速计数器优先处理计数结果的控制。

指针编号	中断禁止标志位
I010	SM8059
I020	
I030	
I040	
I050	
I060	

程序举例：

主程序：



I40 中断子程序：

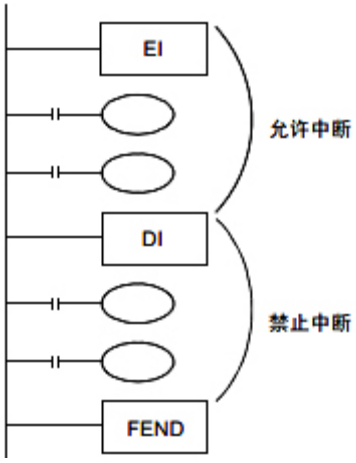


中断说明：当高速计数器 C249 计数值与 D8 相等时，会立即触发 I400 这个计数器中断，然后 D1000 值加 1。

（4）中断范围限制

通过 DI 指令，可设定中断禁止的区域，当在 DI~EI 之间发生中断时，不会马上执行而是中断被存储下来，在 EI 执行之后才执行

程序举例：

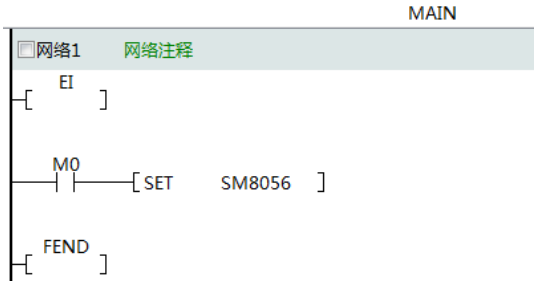


（5）禁止中断

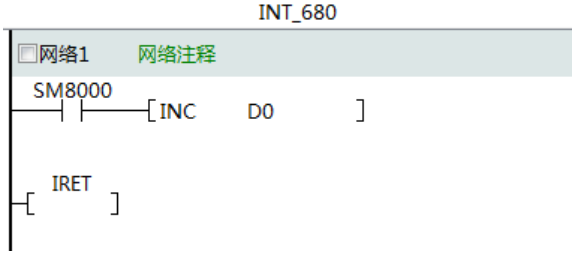
每个中断子程序都有自己的禁止中断，当置位对应的禁止中断后，即使发生了中断，中断也不会执行。

程序举例：

主程序：



I680 中断子程序：



正常情况下，每过 50ms，会产生定时器中断，D0 值加 1，当置位 SM8056 后，D0 值将不再增加，复位 SM8056 后又会每 50ms 产生中断使 D0 加 1。

2、中断优先级

外部输入中断 > 定时器中断 > 计数器中断

I000 > I100 > I200 > I300 > I400 > I500 > I001 > I101 > I201 > I301 > I401 > I501 > I6xx > I7xx > I8xx > I010 > I020 > I030 > I040 > I050 > I060

3、注意事项

（1）如果对应的中断禁止标志置位，即使产生了中断也不会执行。

（2）注意 SM8393 寄存器的动作，中断程序后面紧接一个 LD SM8393，表示中断产生后不立即动作，而是延迟 SD8393 的延时时间才动作。

3.3 高速计数

根据不同点数 PLC，有不同的适用范围。

点数	规格
16 点	2 个单相 60K 或 1 个 AB 相 30K（最多选择同时使用 2 个计数器）
24 点	2 个单相 60K 或 1 个 AB 相 30K（最多选择同时使用 2 个计数器）
32 点	2 个单相 60K 或 1 个 AB 相 30K（最多选择同时使用 2 个计数器）
40 点	2 个单相 60K 或 1 个 AB 相 30K（最多选择同时使用 2 个计数器）
60 点	2 个单相 60K 或 1 个 AB 相 30K（最多选择同时使用 2 个计数器）

根据每个点数的端子，再从下表选择端子对应模式的计数器。

(1) 支持使用的高速计数器分类：

U：增计数输入；D：减计数输入；R 复位输入；S：启动输入；

A：A 相脉冲输入；B：B 相脉冲输入

	C235	C236	C237	C243	C244	C251	C252
X0	U/D			U/D		A	A
X1		U/D		R		B	B
X2			U/D		U/D	R	
X3					R		

注意：占用 X0，X1 口的两个单相计数器不能同时使用。

（2）控制特殊寄存器：

单相单计数增/减计数控制特殊 SM 元件：

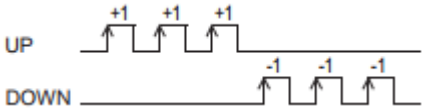
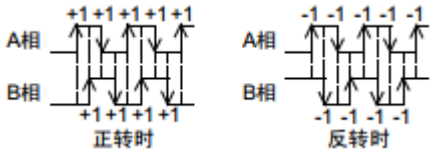
	C235	C236	C237	C243	C244	C251	C252

增减计数控制	SM8235	SM8236	SM8237	SM8243	SM8244	SM8251	SM8252

倍频控制特殊 SM 元件：

4 倍频使能	C251	C252
	SM8195	SM8196

（3）计数模式说明

计数模式		输入信号形式	计数方向
单相单计数			通过 SM8235～SM8245 的 ON/OFF 来指定增计数或是减计数。 ON：减计数 OFF：增计数
单相双计数			通过 SM8246～SM8250 的 ON/OFF 来查看增计数或是减计数状态。 ON：减计数状态 OFF：增计数状态
双相双计数	1 倍频		通过 SM8251～SM8255 的 ON/OFF 来查看增计数或是减计数状态。 ON：减计数状态 OFF：增计数状态 通过置位 SM8195～SM8199 可使 C251～C255 转换为 4 倍频计数模式
	4 倍频		

（4）高速指令的使用，请参考 2.7 章节。

注意：使用 RST 指令复位高速计数器时，RST 指令前面不要用常闭开关，如 SM8000。否则，会导致计数器处于复位状态而不能正常计数。

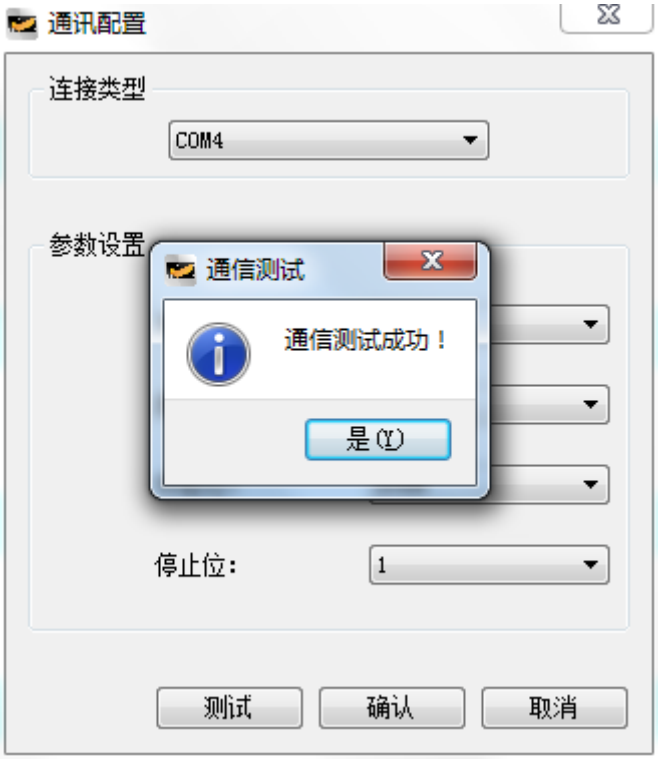
3.4 串口通讯部分

X 系列 PLC 包含三路可用串口，232 口、485 口，还有一路 BD 板口，当 BD 板时 485BD 板时，其功能与另外两个串口功能相同,都是以梯形图形式来实现 232 口和 485 口的 RS2 或 MODBUS 通信。COM0、COM1、COM2 接 485BD 板时,都可以进行监控下载、RS2 通信、MODBUS 通信。COM0、COM1、COM2 是相对于 PLC 板子的三个串口通道的顺序，上位机显示的则是实际电脑检测出来的 COM 标号，不能混淆。

3.4.1 监控下载

其中 RS485 接口和 RS232 接口支持下载监控协议。

通过 USB-485 通信转换器或 USB-232 通信转换器，将 PLC 对应端口与 PC 相连接，在 PC 上打开 UniSYS 软件，即可通过 UniSYS 软件对 PLC 进行下载梯形图和监控操作。值得说明的是，在通信开始之前，需要在 UniSYS 软件上对 PLC 进行通信测试，依次点击菜单栏工具->通信配置，选择对应端口，设置波特率，点击测试，若弹框显示通信测试成功，则说明 UniSYS 软件可与 PLC 进行正常的通信操作，若显示其他弹框，则说明通信存在故障，UniSYS 软件无法与 PLC 进行正常通信，此时需查找故障原因（查看接线是否正确，PLC 是否正常）。下图为正常通信状态显示弹框。



下载监控协议是一种主从协议，其中 UniSYS 软件作为主机，PLC 作为从机。从机一般不主动发送通信数据，在主机有通信请求时，从机会向主机发送通信数据作为应答。

下载监控协议是一类协议命令的统称，此协议包含 UniSYS 软件对 PLC 的大部分操作，包括下载，上载，校验，监控，强制等功能。

下载通信协议可以设置多种波特率，最常见的有 9600baud，19.2kbaud 和 115.2kbaud。

USB-485 或 USB-232 通信转换器是一种即插即用设备，可支持 USB V1.1 及 USB2.0 的 PC，此设备提供 PC 端 USB 端口的数据向 RS485 或 RS232 类型数据的转换，以便 PLC 端对应的端口可以收到正常的通信数据。必要时，使用此设备需要在 PLC 端安装对应的驱动程序，可使用驱动安装软件进行自动检测，并一键安装相应的驱动。

3.4.2 RS2 通信

通过 X3-PLC 本体上的 COM0/COM1/COM2(接 485BD 板)口进行自由口通信，从而执行数据的发送和接收的指令。

1、RS2 指令格式

FNC87	指令所占程序步	指令分类	指令格式
16 位	11	RS2 \	[RS2 S m D n n1]

操作数说明

操作数	内容	类型
S	保存发送数据的数据寄存器的起始软元件	BIN16 位
m	发送数据的字节数[设定范围: 1～512]	BIN16 位
D	数据接收结束时，保存接收数据的数据寄存器的起始软元件	BIN16 位
n	接收数据的字节数[设定范围: 1～512]	BIN16 位
n1	使用通道编号[设定内容：K0：COM0； K1：COM1；K2：COM2]	BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
m														*	*				*	*			
D														*	*			*					
n														*	*				*	*			
n1																			*	*			

2、功能和动作说明

该指令用于基本单元上的 RS-232C 或 RS-485 串行通信口的无协议通信，从而执行数据的发送和接收。使用该指令首先得使用梯形图配置通信相关的寄存器。

（1）通信格式寄存器配置表

通信格式配置	Bit15~13	Bit12~11	Bit10~7	Bit6~4	Bit3	Bit2~1	Bit0
协议	协议类型	硬件类型	RS2 的帧格式	波特率	停止位	校验位	数据位
RS2	010：RS2	00:232口 01:485口	0001~1100	001:4800 010:9600 011:19200 100:38400	0:1bit 停止位 1:2bit 停止位	00:无校验 01:奇校验 10:偶校验	0:7bit 数据位 1:8bit 数据位

通过配置 RS2 帧格式，选择要在通信中使用的报文帧：

0001	数据
0010	数据+CR+LF
0011	数据+报尾
0100	数据+报尾+CR+LF
0101	数据+报尾+和校验
0110	数据+报尾+和校验+CR+LF
0111	报头+数据

1000	报头+数据+CR+LF
1001	报头+数据+报尾
1010	报头+数据+报尾+CR+LF
1011	报头+数据+报尾+和校验
1100	报头+数据+报尾+和校验+CR+LF

（2）COM0（232 口）使用 RS2 通信时的相关软元件

位软元件	名称
SM8062	串行通信错误
SM8120	发送等待标志位
SM8121	发送请求
SM8122	接收结束标志位
SM8129	超时的判断标志位

字软元件	名称
SD8062	串行通信错误代码
SD8120	设定通信格式
SD8122	发送数据的剩余点数
SD8123	接收点数的监控
SD8129	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms
SD8124	报头 1、2
SD8125	报头 3、4
SD8126	报尾 1、2
SD8127	报尾 3、4
SD8405	接收和校验(接收数据)
SD8418	接收和校验(计算结果)
SD8419	发送和校验

（3）COM1（485 口）使用 RS2 通信时的相关软元件

位软元件	名称
SM8063	串行通信错误
SM8401	发送等待标志位

SM8402	发送请求
SM8403	接收结束标志位
SM8409	超时的判断标志位
字软元件	名称
SD8063	串行通信错误代码
SD8400	设定通信格式
SD8402	发送数据的剩余点数
SD8403	接收点数的监控
SD8409	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms
SD8410	报头 1、2
SD8411	报头 3、4
SD8412	报尾 1、2
SD8413	报尾 3、4
SD8414	接收和校验(接收数据)
SD8415	接收和校验(计算结果)
SD8416	发送和校验

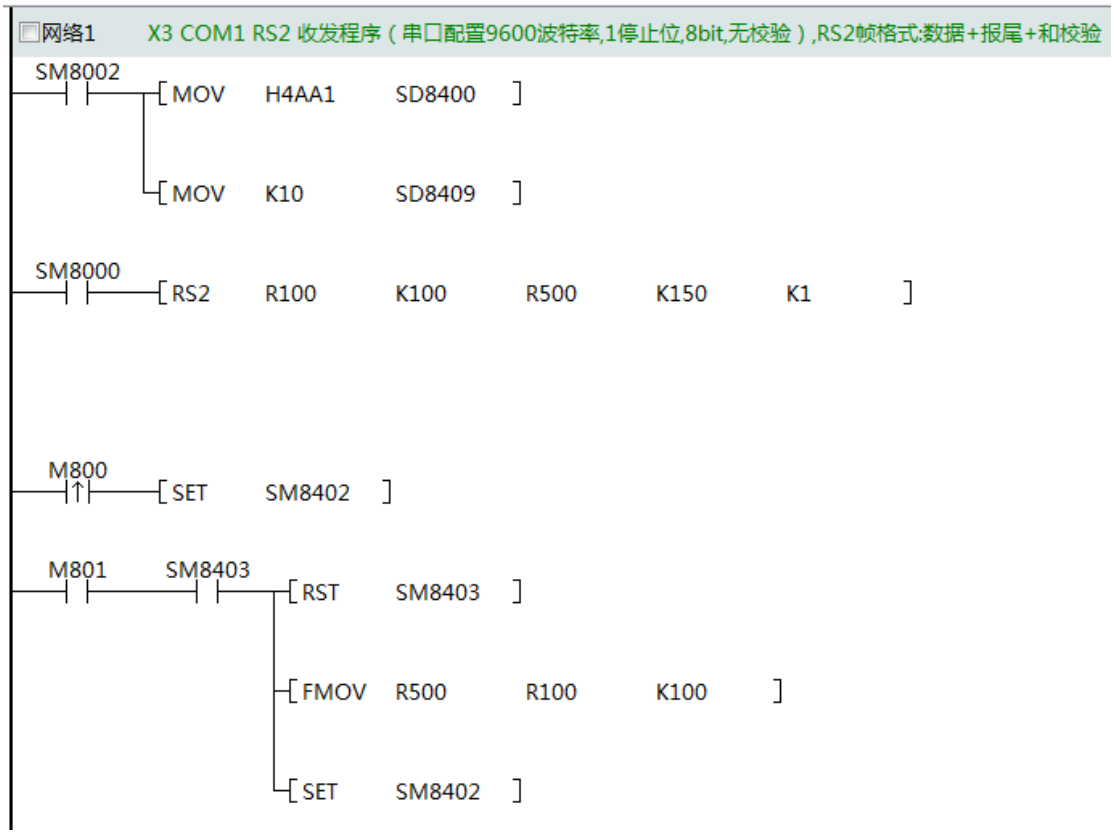
（4）COM2（接 485BD 板）使用 RS2 通信时的相关软元件

位软元件	名称
SM8438	串行通信错误
SM8421	发送等待标志位
SM8422	发送请求
SM8423	接收结束标志位
SM8429	超时的判断标志位

字软元件	名称
SD8438	串行通信错误代码
SD8420	设定通信格式
SD8422	发送数据的剩余点数
SD8423	接收点数的监控
SD8429	设定响应超时时间（单位 10ms），设定为 0 时，默认 100ms

SD8430	报头 1、2
SD8431	报头 3、4
SD8432	报尾 1、2
SD8433	报尾 3、4
SD8434	接收和校验(接收数据)
SD8435	接收和校验(计算结果)
SD8436	发送和校验

3、程序举例



当与其通信的设备串口配置与此一致时，梯形图中置位 SM8402（发送请求），PLC 将发送 R100~R149 中的 100 个字节的数据，发送完毕后会自动复位 SM8402。当 SM8403（接收结束标志）置位时，监控可以发现 R500 以后的 150 个字节保存的是通信设备发送过来的数据，此时再次置位 M801，PLC 会将接收的数据中的前 100 个字节发送给通信设备。

4、注意事项

（1）报头：最多 4 个报头，放于对应的特殊寄存器，00H 之前的数据有效，按低位、高位、低位、高位的顺序排布。比如 COM1 的报头设定为 SD8410 = 1234H，SD8411 = 5600H，则实际报头为 H34（SD8410 低位） H12（SD8410 高位）。

（2）报尾：最多四个报尾，使用方法类似报头。

（3）和校验：选择和校验时必须设定有报尾。发送数据时，计算数据+报尾的和，附加在发送的数据中；接收数据时，检查接收到的和，是否与 PLC 中计算的相同。

（4）CR+LF：通信格式设定中选择了 CR+LF，会在发送的数据后加上 CR+LF 的字符代码，接收数据时如果连续接收到 CR+LF，则表示接收结束。

（5）接收数据的字节点数 n 应该大于等于实际接收的数据长度加上 8。

（6）一般情况下，接收完毕最好不要立即置位发送请求标志，给一点延时时间等待串口状态转换。

3.4.3 串口 MODBUS RTU 通信

1、梯形图中配置通信需要的参数说明

（1）通信格式配置表

通信格式配置	Bit15~13	Bit12~11	Bit10~7	Bit6~4	Bit3	Bit2~1	Bit0
协议	协议类型	硬件类型	主从站	波特率	停止位	校验位	数据位
MODBUS	011：MODBUS	00:232 口 01:485 口	0000：从站 0001：主站	001:4800 010:9600 011:19200 100:38400 101:57600 110:115200	0:1bit 停止位 1:2bit 停止位	00:无校验 01:奇校验 10:偶校验	0:7bit 数据位 1:8bit 数据位

RTU 模式数据传输方式：

编码系统：8 位二进制，十六进制 0-9，A-F

每个字节的位：1 个起始位；8 个数据位；带校验时 1 位停止位；无校验时 2 位停止位

错误校验区：循环冗余校验（CRC）

（2）通信协议配置表

位	名称	内容	
		0（bit=OFF）	1（bit=ON）
b0	选择协议	其他通信协议	MODBUS 协议
b1~b3	不可使用		
b4	主从站设定	MODBUS 主站	MODBUS 从站
b5~b7	不可使用		
b8	RTU/ASCII 模式	RTU	ASCII（不支持）

b9~b15	不可使用		

（3）SD8120/SD8400/SD8420 允许配置的通信格式

主站			从站			通信
数据位	校验位	停止位	数据位	校验位	停止位	
8bit	无	1bit	8bit	无	1bit	是
8bit	无	2bit	8bit	无	2bit	是
8bit	奇	1bit	8bit	奇	1bit	是
8bit	奇	2bit	8bit	奇	2bit	是
8bit	偶	1bit	8bit	偶	1bit	是
8bit	偶	2bit	8bit	偶	2bit	是

2、相关特殊寄存器及错误码说明

（1）MODBUS 相关特殊寄存器

其中：COM0 是 232 口，COM1 是 485 口，COM2 是 485BD 板口，下一次通信是指下一条 ADPRW 指令执行或轮询下一行 MODBUS 配置表格，正常情况下 PLC STOP 时所有相关寄存器都会清除。

特殊数据寄存器			有效站	详细内容
COM0	COM1	COM2		
SM8029	SM8029	SM8029	主站	指令结束标志（MODBUS 中用于 ADPRW 指令）
SM8121	SM8401	SM8421	主站/从站	MODBUS 通信中
SM8122	SM8402	SM8422	主站/从站	MODBUS 通信发生错误，下一次通信时清除
SM8123	SM8403	SM8423	主站/从站	MODBUS 通信出错锁存
SM8062	SM8063	SM8438	主站/从站	串口通信出错锁存
SM8128	SM8408	SM8428	主站	主站发生重发事件的判断标志，下一次通信时清除
SM8129	SM8409	SM8429	主站	超时的判断标志，当到达重发次数时，超时标志才会置位，下一次通信时清除
SD8120	SD8400	SD8420	主站/从站	通信格式设定
SD8121	SD8401	SD8421	主站/从站	协议
SD8122	SD8402	SD8422	主站/从站	通信错误寄存器，显示当前最新错误
SD8123	SD8403	SD8423	主站/从站	通信错误锁存寄存器，锁存第一次的错误
SD8062	SD8063	SD8438	主站/从站	串口错误锁存寄存器，MODBUS 通信出错时，里面写入 6321
SD8128	SD8408	SD8428	主站	显示主站当前的重发次数
SD8129	SD8409	SD8429	主站	设定超时时间（单位 ms），填 0 时默认 3s
SD8093	SD8411	SD8431	主站	请求间延迟（帧间延迟，单位 ms），波特率大于 19200，延迟时间默认 4ms，其他为

				10ms，从站中不可更改，当主站连接的从站数比较多时，建议设定>=10ms
SD8094	SD8412	SD8432	主站	设定进行重发的次数（0~20），超过 20 为 20
SD8095	SD8414	SD8434	从站	从站本站号（范围 1~247）

（2）MODBUS 错误码说明

错误码	出错内容	有效站	相关元件动作	解决方式
202	MODBUS 参数设定错误	主站/从站	（1）使用 COM0 时：SM8122 、 SM8123 、SM8062 会置 ON，当前的错误码存在 SD8122 中，SD8123 会锁存第一次的错误码，SD8062 存入 6321 （2）使用 COM1 时：SM8402 、 SM8403 、SM8063 会置 ON，当前的错误码存在 SD8402 中，SD8403 会锁存第一次的错误码，SD8063 存入 6321 （3）使用 COM2 时：SM8422 、 SM8423 、SM8438 会置 ON，当前的错误码存在 SD8422 中，SD8423 会锁存第一次的错误码，SD8438 存入 6321	检查 MODBUS 相关通信配置梯形图或配置参数表是否正确
205	CRC/LCR 校验码错误	主站/从站		检查数据协议帧的校验是否正确
207	响应文本长度不正确	主站/从站		确定接收的响应文本的长度是否正确
210	非法软元件地址读写	主站/从站		确认主站是否访问了有效的软元件范围（地址是否是允许范围内）； 确认从站的 MODBUS 地址分配是否正确或者包含了读写的地址范围
212	异常文本	主站/从站		确定主站读取的地址+点数是否超出规定范围； 检测 ADPRW 指令参数设定是否正确
214	无此功能码	主站/从站		确认 MODBUS 是否支持该功能码
217	ADPRW 指令不正确使用	主站/从站		确认主站中的 MODBUS 通信否配置成功； 确认从站中是否使用了 ADPRW 指令，如果有则删除
211	从站响应超时	主站	除上述动作外，对应的串口超时标志位（分别是 SM8129、SM8409、SM8429）会置位	确定从站站号和通信参数是否正确； 确定通信线是否连接稳定； 确定从站是否正常启动

3、MODBUS 标准功能支持一览表

功能代码	子功能代码	功能名	文本可访问点数	该功能是否允许
0x01		线圈读出	1~2000 点	√
0x02		输入读出	1~2000 点	√
0x03		保持寄存器读出	1~125 点	√
0x04		输入寄存器读出	1~125 点	√
0x05		1 线圈写入	0 固定	√
0x06		1 寄存器写入	0 固定	√
0x0F		批量线圈写入	1~1968 点	√
0x10		批量寄存器写入	1~123 点	√
0x17		批量寄存器读出/写入	读出：1~125 点 写入：1~121 点	√

4、主站与从站通信专用指令 ADPRW

用于和 MODBUS 主站所对应从站进行通信(数据的读出/写入)的指令，指令对应的参数可参照上表进行配置。指令格式见下表

FNC276	指令所占程序步	指令分类	指令格式
16 位	11	ADPRW \	[ADPRW S S1 S2 S3 S4/D]

操作数说明

操作数	内容	类型
S	指定站号和通信串口号，从站本站号(1~247)，S+1 为串口号（范围 0~2），在程序运行的第一个周期务必先将串口号填写完毕	BIN16 位
S1	功能代码	BIN16 位
S2	MODBUS 地址（OH~FFFFH）	BIN16 位
S3	读写 MODBUS 地址点数或子功能数据	BIN16 位
S4/D	读/写数据缓存区	位/BIN16 位

操作数可用元件类型

操作数	位元件							字元件								变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	Dx.y	KnX	KnY	KnM	KnS	T	C	D	R	V	Z	修饰	K	H	E	“”	P
S														*	*			*					
S1														*	*			*	*	*			
S2														*	*			*	*	*			
S3														*	*			*	*	*			
S4/D	*	*	*			*								*	*			*	*	*			

5、MODBUS 软元件地址分配

（1）位软元件

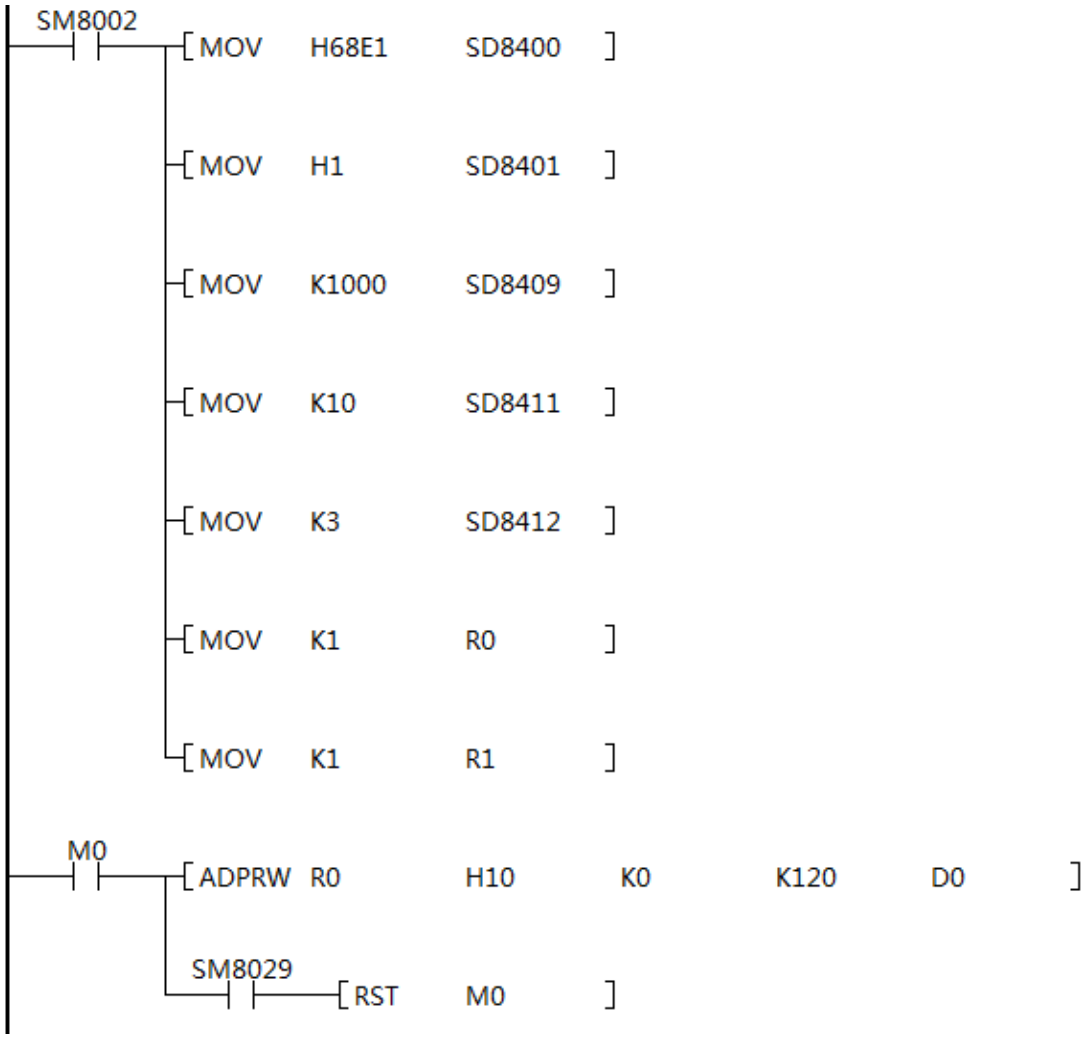
MODBUS 软元件		X 系列软元件
输入（读出专用）	线圈（读出/写入）	
0x0000～0x1DFF	0x0000～0x1DFF	M0～M7679
0x1E00～0x1FFF	0x1E00～0x1FFF	SM8000～SM8511
0x2000～0x2FFF	0x2000～0x2FFF	S0～S4095
0x3000～0x31FF	0x3000～0x31FF	TS0～TS511
0x3200～0x32FF	0x3200～0x32FF	CS0～CS255
0x3300～0x33FF	0x3300～0x33FF	Y0～Y377
0x3400～0x34FF	-	X0～X377

（2）字软元件

MODBUS 软元件		X 系列软元件
输入寄存器（读出专用）	保持寄存器（读出/写入）	
0x0000～0x1F3F	0x0000～0x1F3F	D0～D7999
0x1F40～0x213F	0x1F40～0x213F	SD8000～SD8511
0x2140～0xA13F	0x2140～0xA13F	R0～R32767
0xA140～0xA33F	0xA140～0xA33F	T0～T511
0xA340～0xA407	0xA340～0xA407	C0～C199
0xA408～0xA477	0xA408～0xA477	C200～C255
0xA478～0xA657	0xA478～0xA657	M0～M7679
0xA658～0xA677	0xA658～0xA677	SM8000～SM8511
0xA678～0xA777	0xA678～0xA777	S0～S4095
0xA778～0xA797	0xA778～0xA797	TS0～TS511
0xA798～0xA7A7	0xA798～0xA7A7	CS0～CS255
0xA7A8～0xA7B7	0xA7A8～0xA7B7	Y0～Y377
0xA7B8～0xA7C7	-	X0～X377

6、梯形图配置 MODBUS 通信举例

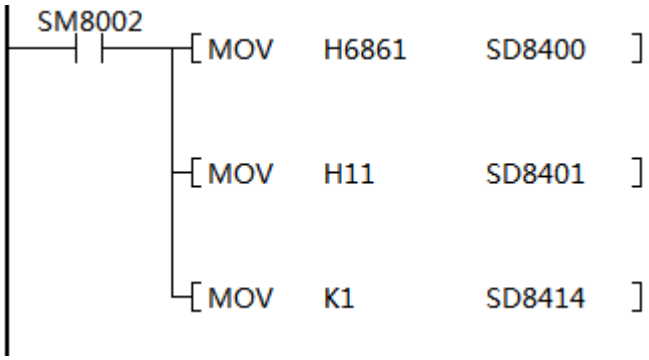
（1）COM1 主站配置举例



其中：COM1 的主站通信配置为波特率 115200，8 位数据位，无检验，1 位停止位，超时时间 1s，帧间延迟 10ms，重试次数 3 次。当 M0 置位时，执行批量寄存器写入功能，将 D0~D199 的 120 寄存器数据写到从站。

注意：①ADPRW 参数的实际意义，S1，S1+1 参数要在第一个周期进行初始化；②一般的，主站根据串口号对应的寄存器配置 5 个寄存器即可完成通信配置；③SM8029 的作用周期，一条 ADPRW 指令结束置位，下一条于 ADPRW 指令开始时复位。可通过此标志进行 MODBUS 的轮询操作。

（2）COM1 从站配置举例



其中：COM1 的从站通信配置为波特率 115200，8 位数据位，无检验，1 位停止位，站号为 1。
注意：一般的，从站根据串口号对应的寄存器配置 3 个寄存器即可完成通信配置。

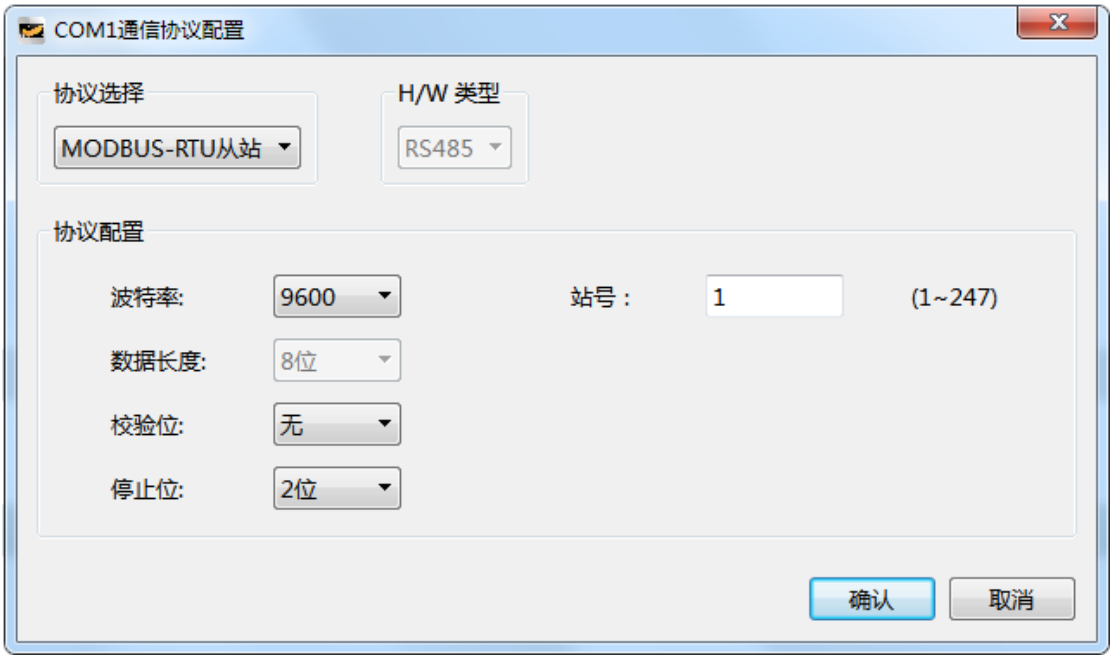
7、配置表法配置 MODBUS 通信

（1）协议配置

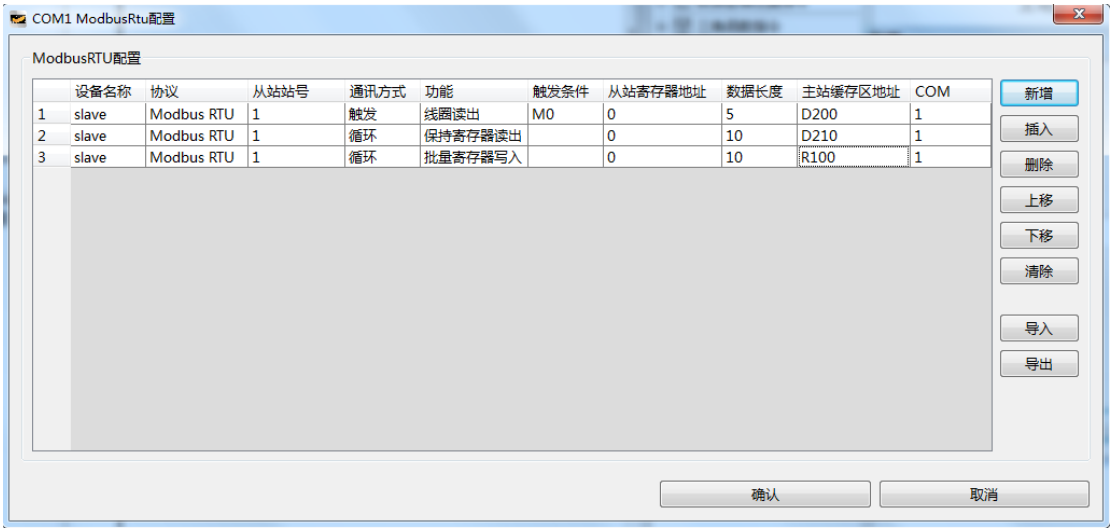
主站配置：



从站配置：



（2）通信配置



（3）使用说明

①配置 MODBUS 通信时，可以直接采用配置表方式，将配置表和程序一起下载到 PLC，即可执行 MODBUS 通信，不需要再去编写梯形图来配置通信协议，配置表法配置通信的优先级高于梯形图配置。

②MODBUS RTU 配置表，主要用于主站通信，分为触发通讯方式和循环通讯方式。

③循环方式和触发方式的使用建议如下

• 循环通信:需要反复、尽快读写从站内变化比较快的数据，例如读取变频器的运行频率、运行状态、输入端口状态;在过程控制应用系统中，实时通信改写变频器的运行频率、输出端口的状态，等等，可选择“循环”通信方式。PLC 在执行用户程序时，会反复扫描执行通信配置表中的所有“循环” 配置项。

• 触发通信:需要定时读或写从站的刷新速度较慢的数据，例如读取变频器的输出电流、输出功率、当前故障告警信息，等等，可选择“触发”通信方式，在用户程序中,每置位触发

条件 1 次，即会引起通信配置表中对应的通信项的 1 次通信操作;用户程序中定时置位该触发条件，即可实现所需频度的通信读写操作。

• 通信方式设定的建议:根据所需要交互参数刷新的特性要求进行合理选择，可使得通信表现大为改善，不要为了编程简单，将所有通信项都设置为“循环”项，这样可能因循环项太多，而使得数据交互及时性降低，影响系统的控制效果。将一些不重要的数据访问，安排为“触发”项，根据轻重缓急来触发通信，可大为改善通信及时性效果。

④配置表法中可用功能只有 4 个，线圈读出、批量线圈写入、保持寄存器读出、批量寄存器写入。

8、特别说明

（1）X3 的通信格式配置务必参考此文档进行配置。

（2）MODBUS 通信只有在 PLC 为 RUN 的状态下才通信，STOP 状态 PLC 恢复为监控协。

（3）X3 目前支持 RTU 模式传输数据，不支持 ASCII 模式。

（4）站号可设定范围为 1-247，为保证通信稳定最大允许连接 32 台从站，当主站连接的从站数比较多时，建议设定>=10ms。

（5）支持同时使用多个 COM 口进行 MODBUS 通信，MODBUS 通信是开始于第二个周期的，故 PLC 运行的第一个周期请在 ADPRW 指令之前初始化通信配置参数以及 ADPRW 指令使用到的串口参数。

（6）不允许中途更改通信参数配置和 ADPRW 指令中的串口参数，如果更改了，只有下一次 PLC 由停止变为运行时才会生效。

（7）主站不支持播放功能。

（8）MODBUS RTU 配置表的使用优先级高于梯形图配置的寄存器内容和 ADPRW 通信指令。

3.5 密码功能

分为初级加密和高级加密。初级加密主要是通过密码对程序的读写进行限制操作，防止使用者随意上载或下载或查看程序。高级加密主要是通过秘钥来限制该程序的使用权限，只有给 PLC 注入了秘钥，才能使用与之匹配的扩展程序，扩展程序被加密后始终具有不可查看和不可更改属性，可以很好的保护自己的知识产权。

3.5.1 PLC 初级加密功能

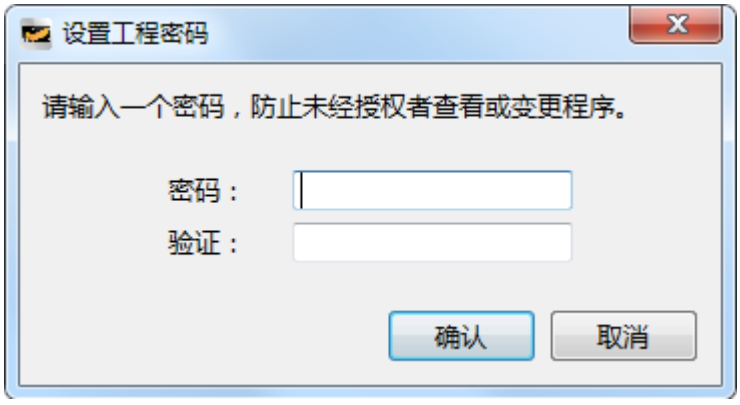
上位机通过 PLC 密码设置可以设置 PLC 程序的读写（修改）权限，初级加密分为工程加密、程序加密两大块，程序加密又分为禁止上载、禁止下载、屏蔽主程序、屏蔽子程序。

1、工程加密

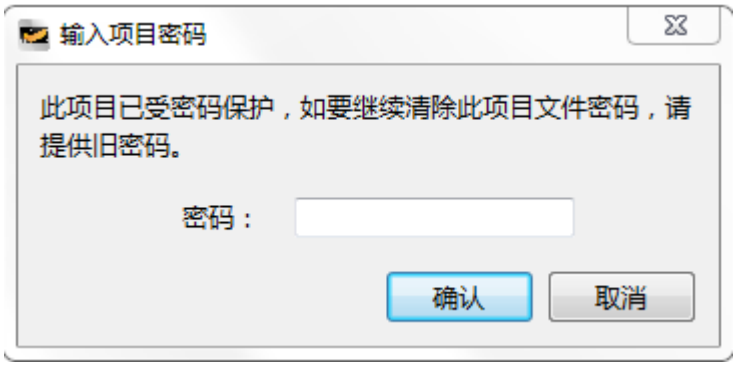
防止未授权者查看和变更程序，只允许下载，下载后，PLC 中存储此密码，此程序将不再允许上载，只有在 PLC 的密码设置中清除密码才能上载；当加密的工程下载至 PLC 后，可以允许其它程序的下载，但是下载后其它程序也会变更为禁止上载状态，只有进行存储器清除或在 PLC 的密码设置中清除密码才能正常上载，这个功能主要用于保护客户原程序不被上载。

操作方法及效果如下：

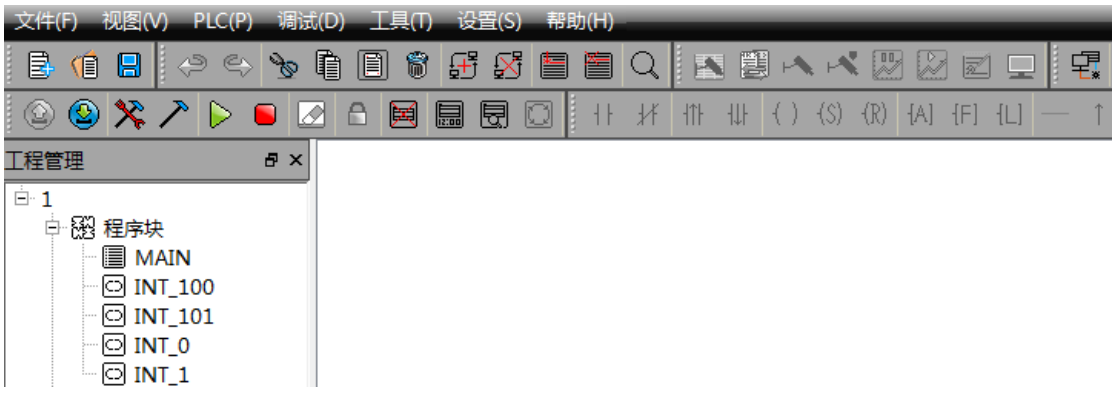
①点击 UNISYS 上位机中的文件->设置工程密码，密码设定后，再次打开这个工程，这个工程会被加密。



②打开加密的工程文件时，如果输入正确的工程密码，则可查看和修改程序内容。



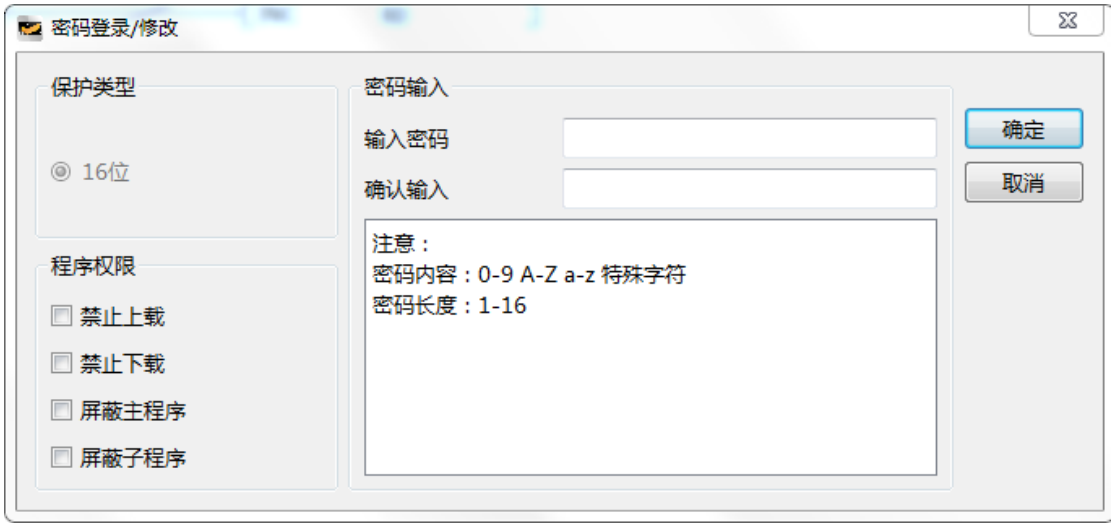
③跳过上面的密码验证过程，或者不进行文件->清除工程密码，则程序块部分中的程序全部被屏蔽，不可查看和更改，效果如下。



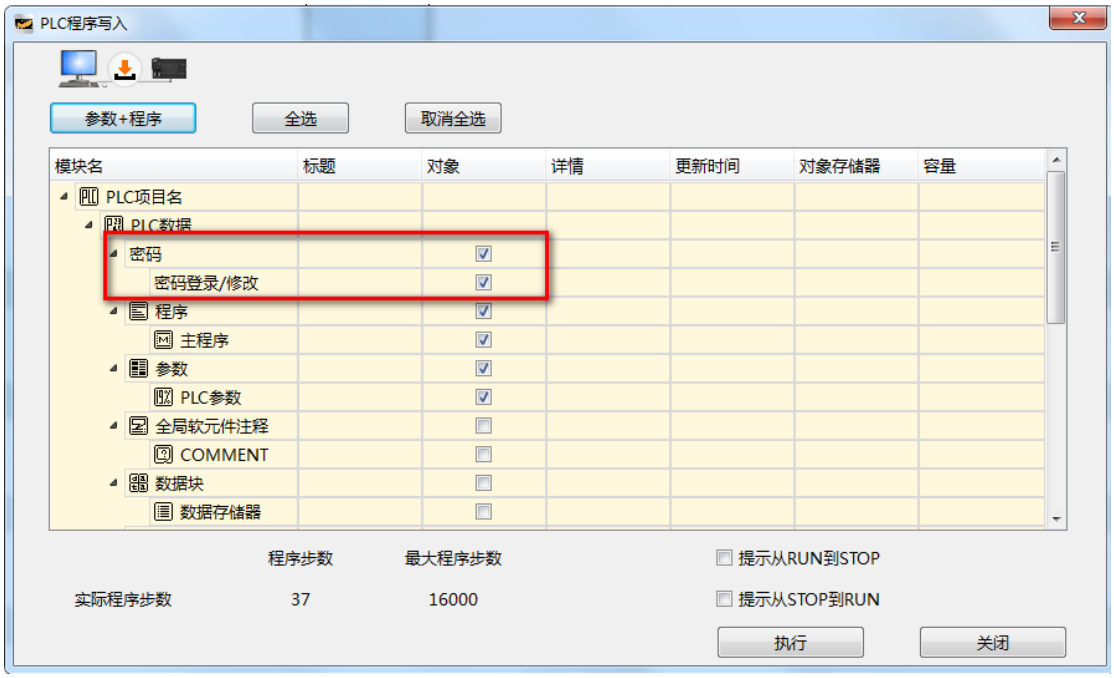
④项目工程未解密或者清除密码的情况下，将此工程下载至 PLC，PLC 会将此项目密码保存起来，并变更程序的权限为禁止上载。

2、程序加密

对用户程序进行相应权限的加密，可防止重要部分的梯形图程序被查看或变更，分为禁止上载、禁止下载、屏蔽主程序、屏蔽子程序。

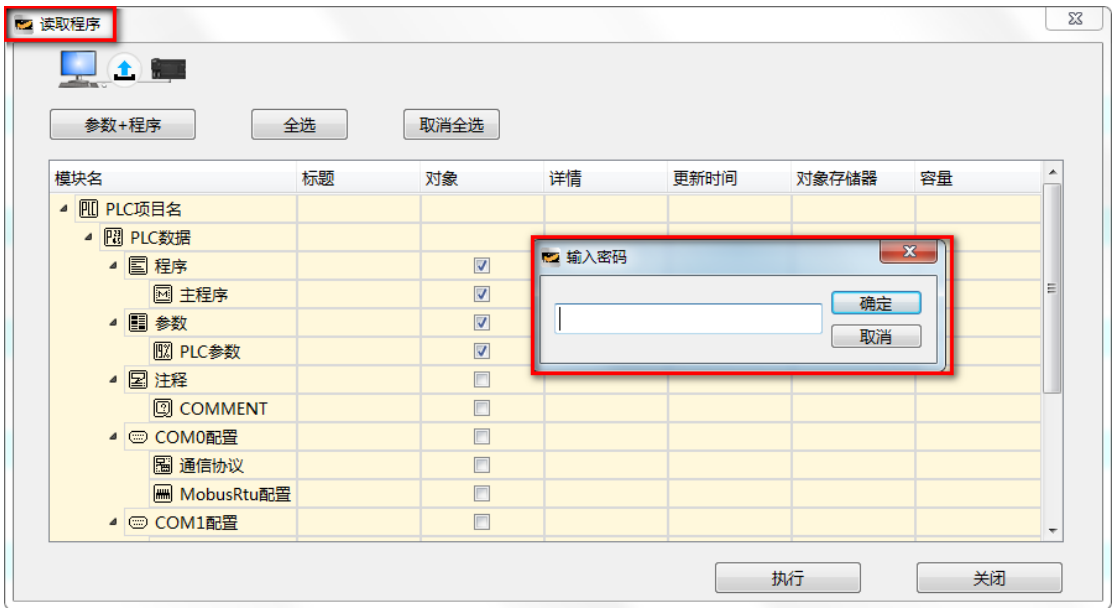


勾选对应的权限以限制程序的可查看或可变更权限，在下载时跟随程序、配置等一起将密码下载到 PLC，如果 PLC 有密码的情况下则需要先输入原密码，然后 PLC 才会变更为当前下载的密码。

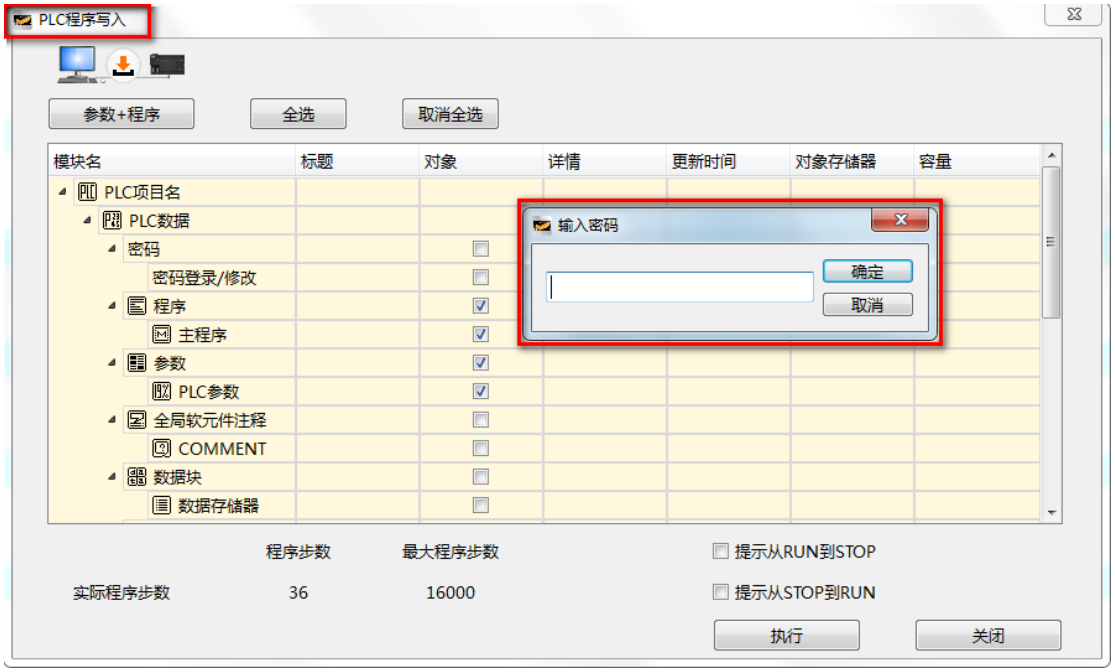


效果如下（前提是 PLC 已经进行了相应的程序权限加密）：

（1）禁止上载：上载时



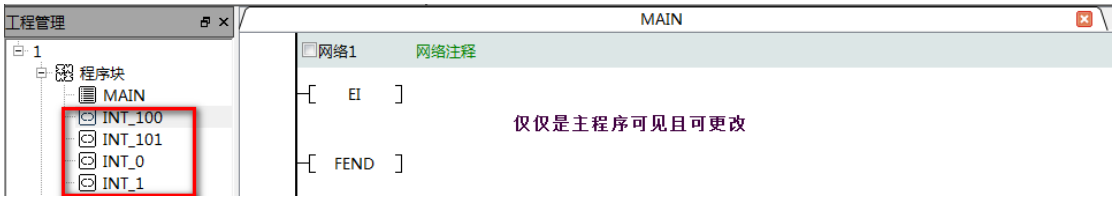
（2）禁止下载：下载时需要输入密码才可下载程序



（3）屏蔽主程序：主程序部分不可见、不可更改，子程序部分可以更改

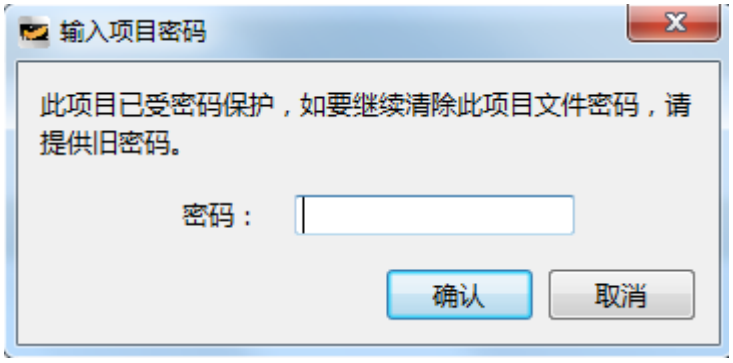


（4）屏蔽子程序：

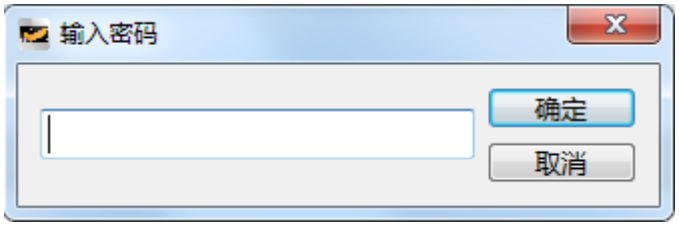


3、密码清除

（1）清除工程密码：点击 UNISYS 中文件->清除工程密码，清除密码时需要输入正确密码，清除后，项目中所有程序变为可见和可变更状态。



（2）清除 PLC 的密码：点击 UNISYS 中 PLC->PLC 密码设置->清除，清除密码时需要输入正确密码，密码清除后则存储在 PLC 中的程序权限将消失。另外 PLC 存储器清除时会清除 PLC 中存储的密码。



4、注意事项

（1）禁止上载权限不能同时与屏蔽主程序权限或者屏蔽子程序权限共同存在，其它的权限可以同时存在。

（2）密码功能主要是限制程序的读写权限，当忘记密码时，可使用存储器清除功能清除掉密码，但此时 PLC 存储的程序也会被清除。

3.5.2 PLC 高级加密功能

应用场景 1

A 和 B 合作开发一个梯形图，并各自保护自己的知识产权，A 将加密了的扩展程序工程给 B 使用（前提是 A 将秘钥下载到 PLC），但 B 无法查看和变更此部分的梯形图。同时 B 可以在 A 的工程的基础上编辑自己的梯形图且也可做相应的加密操作（将 B 的扩展程序的秘钥下载到 PLC 并加密 B 的扩展程序），这样 A 和 B 均可保护自己的知识产权。

应用场景 2

A 事先将秘钥下载到 PLC，将 PLC 给 B 使用，然后 B 就可以使用 A 提供的加密了的扩展程序（B 中带有这个扩展程序对应的秘钥），但是 B 却不能将此工程给 C 的 PLC 使用，因为 C 的 PLC 没有此秘钥。

以上场景的应用都可通过生成扩展程序块和 32 位秘钥来对梯形图进行保护。上位机通过右键程序块可以添加扩展程序块，预使该扩展程序生效，必须将对应的 32 位秘钥通过上位机->PLC 下的秘钥注入操作注入到 PLC 中，然后该扩展程序下载至 PLC 后才会生效，可以通过这个 32 位秘钥有效保护扩展程序。扩展程序块可以编写三块，附加功能有扩展程序加密、密码清除、秘钥生成导出、秘钥导入、资源使用表。

1、扩展程序加密

如果使用高级加密功能并且允许上位机随机生成秘钥，加密前需要先执行秘钥生成导出操作，然后再加密。如果使用自己用字符组合成的秘钥，加密前先执行秘钥导入。主要用于保护扩展程序块，使其不可更改和查看。

2、密码清除

主要用于开发者查看和更改扩展程序块，密码清除后扩展程序变为可见和可更改状态。

3、秘钥生成导出

主要用于生成与扩展程序块匹配的 32 位秘钥，再将这个秘钥注入到 PLC，该扩展程序块下载到 PLC 才能被识别使用。

4、秘钥导入

主要用于开发者自己编写自己的秘钥。

5、资源使用表

可编辑可保存，主要用于开发者提供资源使用情况，方便后面的开发者在有扩展程序的工程中继续编程，当然也可以不提供，不让其他客户知道初次开发者使用了哪些资源。

6、具体使用方法

扩展程序中生成导出（或导入）秘钥后，该扩展程序只能被带有此秘钥的 PLC 使用，通过下载此秘钥到 PLC，使 PLC 获得该扩展程序的使用权限。

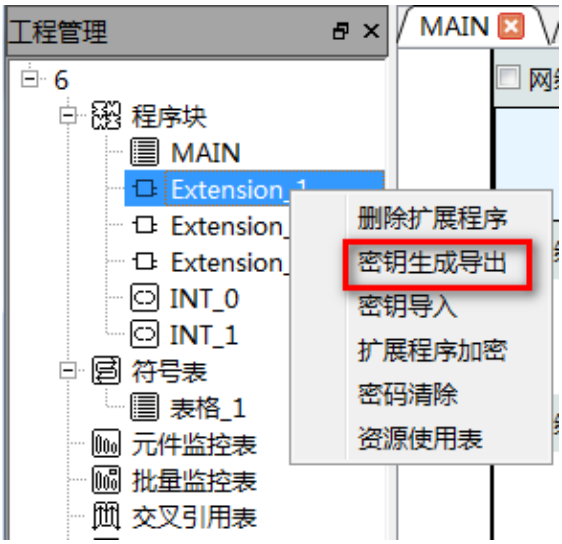
初次开发者按如下流程编写扩展程序块，并将 32 位秘钥注入到 PLC，如此该 PLC 才能使对应的扩展程序块生效。

（1）编写扩展程序

打开 UNISYS 中程序块，右键程序块插入扩展程序，使用几块就插入几块，最多三块，右键扩展程序块可插入子程序或中断子程序，所有程序的总容量大小是与系统参数设定相关的，编程规则与 MAIN 程序一致。

（2）秘钥生成导出

用上位机给扩展程序匹配随机的 32 位秘钥，并导出到文本文件*.txt，方便开发者查看具体秘钥信息。

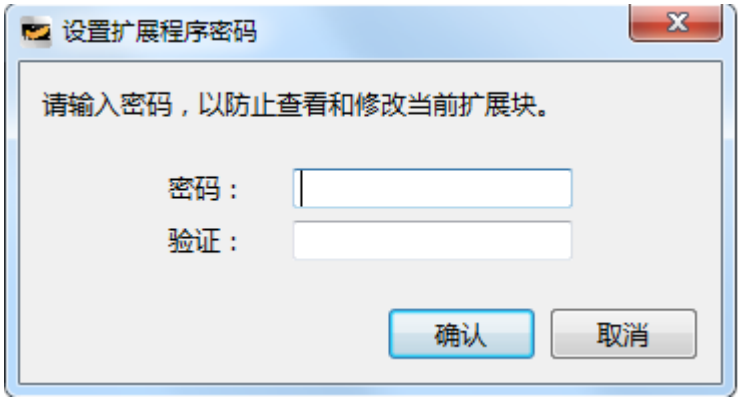


（3）秘钥导入

采用自己组合的 32 位秘钥，则跳过（2）步骤，开发者创建一个.txt 文件，将 32 位秘钥信息写入（有效字符 0~9，a~z，A~Z），可以在扩展程序中使用自己的秘钥。

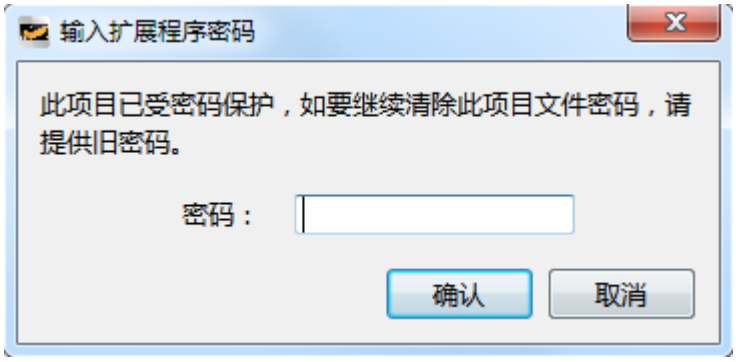
（4）加密扩展程序块

扩展程序块加密后才能被下载到 PLC 中，否则上位机不会下载扩展程序，加密后，扩展程序的屏蔽功能会生效，将不让查看和修改。



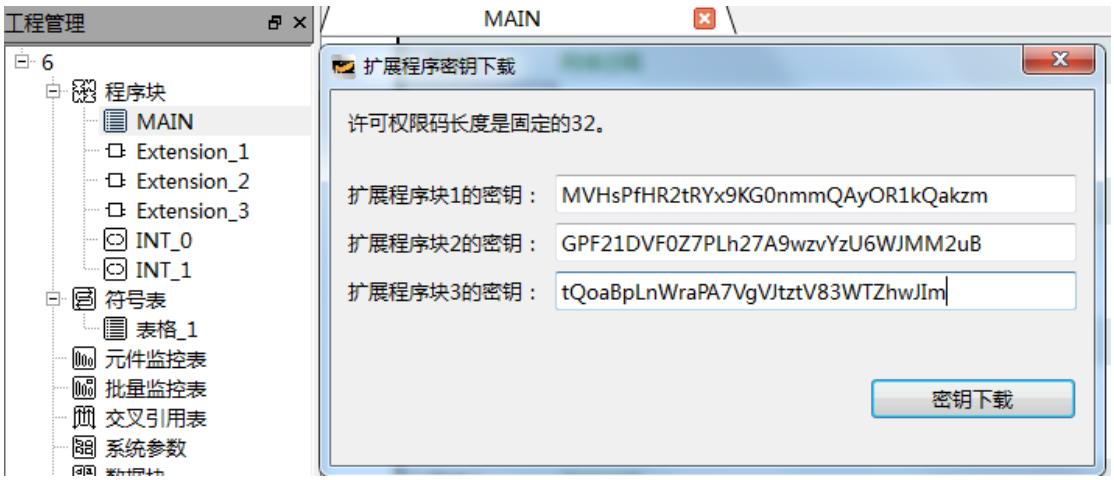
（5）密码清除

方便开发者查看和更改扩展程序。



（6）PLC 秘钥注入

使用扩展程序块之前，通过上位机->PLC 下的扩展程序秘钥下载操作先将对应的秘钥注入到 PLC，这个秘钥是开发者提前下载到 PLC 的与扩展程序匹配的秘钥。



（8）勾选程序下载，会将扩展程序块对应的工程也下载到 PLC 中

7、注意事项

（1）扩展程序块是从属于程序块的，编写梯形图是编写整个程序块，这里只是把要加密的内容放到了扩展程序块，但编程原则还是和单独编程 MAIN 程序是一致的，只是相当于把以前的梯形图程序划分为多块而已，MAIN 程序是前部分的梯形图，扩展程序则是后部分的梯形图，当然可以选择只编写扩展程序，这样就相当于加密整个梯形图数据。

（2）扩展程序编写完毕一定要先对其进行秘钥生成导出或秘钥导入后再加密，以便获得该扩展程序的使用秘钥，这个秘钥是用来下载到 PLC，使 PLC 对该扩展程序块生效用的。

（3）当 PLC 有扩展程序块时，对应扩展程序块标号的新的秘钥注入不会生效，即不允许有扩展程序块时更改秘钥，需进行存储器清除清除或程序清除时才能下载新的秘钥。

（4）当扩展程序里面的秘钥与 PLC 中的秘钥不匹配时，扩展程序会下载失败，PLC 中将只生效 MAIN 程序和下载成功的扩展程序。

（5）删除扩展程序块时需要输入密码验证，正确才能删除，同时删除里面的程序、秘钥、原密码。

3.6 以太网功能

1、以太网模式下的监控下载

选择上位机的通信配置，连接类型更改为以太网，显示一个默认的 IP 地址，点击搜索可搜索出同网段连接了以太网的 PLC IP 地址及其对应的 MAC 地址，更改网段可搜索其他网段的 PLC IP 地址。然后选择要连接的 PLC IP，点击测试，测试成功后即可进行监控下载了。支持多个安装了 UNISYS 软件的电脑监控下载或一台电脑上使用多个 UNISYS 监控下载,但是个数有限制，具体看下面的连接协议分配介绍。

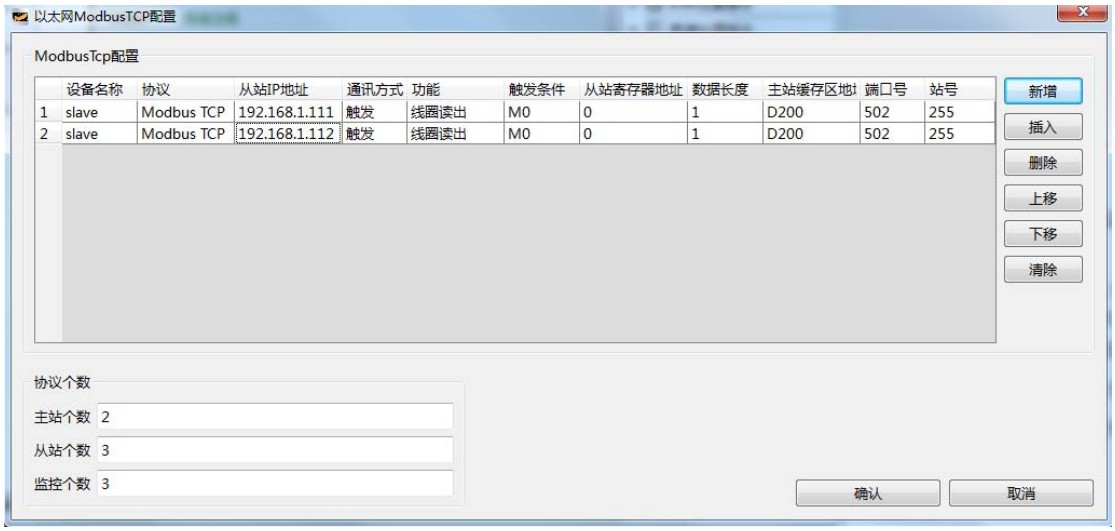


2、上下载 PLC 的 IP 的地址

上位机可以重新配置 PLC 的 IP 地址参数，PLC 默认 IP 是 192.168.1.254



3、以太网通信协议配置



（1）协议个数

根据配置表来分配协议个数，主站个数与 IP 有关，从站和监控个数可自行分配，总的连接个数不能超过八个,即最多允许 8 个不同 IP。

① 主站个数：根据配置参数中的不同 IP 来分配，有几个 IP 就分配几个主站个数，因为 PLC 满足八个功能码，所以允许同一个 IP 配置参数可以有八行配置表格，且在输入表格时可以任意选择配置参数，但同一 IP 的总配置表格行数不能超过 8 行

② 从站个数：表示允许多少个主站（IP 不同）来连接当前的 PLC

③ 监控个数：表示允许多少个上位机软件对该 PLC 进行监控下载

（2）MODBUS TCP 通信配置

① 序列号：对应配置表所在行数

② 协议：MODBUS TCP

③ 设备名称：用于备注 PLC 连接的从站设备名称

④ 从站 IP 地址：即 PLC 连接的服务器的 IP 地址

⑤ 通讯方式：触发表示主站必须将触发条件置位才能进行数据发送来访问从站；循环表示主站一直发送数据访问从站

⑥ 功能：包括线圈读出,输入读出,保持寄存器读出,输入寄存器读出,单个线圈写入,单个（保持）寄存器写入,批量线圈写入,批量（保持）寄存器写入

⑦ 触发条件：在通讯模式为触发的情况下，需要将对应的条件标志位置位才能通信，通信超时或者通信完毕后会复位触发条件

⑧ 从站寄存器地址：主站要访问的从站对应的软元件地址，地址分配与串口 MODBUS 软元件地址分配一致

⑨ 数据长度：根据选择的功能来限制数据长度。线圈读出、输入读出：1~2000；保持寄存器读出、输入寄存器读出：1~125；单个线圈写入、单个（保持）寄存器写入：固定 0；批量线圈写入：1~1968；批量（保持）寄存器写入：1~123

⑩ 主站缓冲区地址：存储从站回传的数据信息，只能是 D 或 R 寄存器

11 端口号：MODBUS TCP 通信用，固定 502

（3）特别说明

① PLC 掉电时会保存上次的 MODBUS TCP 参数，下次会根据配置进行通信；

② 当分配的监控个数为 0 且将该分配协议个数情况下载至下位机时，上位机的以太网模式下将不能进行监控下载功能；

③ MODBUS TCP 参数配置表最大可配置 64 行，只允许 8 个不同 IP，且每个 IP 配置表不能超过 8 行，主站个数要与配置表对应，否则不允许这样的配置

④ 不同从站（IP 不同）的触发条件必须不同，上位机会进行限制

⑤ 使用 PLC 存储器清除功能时会清除 PLC 中存储的 MODBUS TCP 参数配置表

4、以太网相关特殊寄存器

元件	功能
SM8500（可读写）	从站不在线标志 0：从站都在线，此时会复位 SM8500 1：有从站不在线，任意从站不在线时置位
SM8501（可读写）	PLC 网线未连接或断开 0：网线连接正常 1：网线未连接或断开
SM8502（可读写）	以太网功能控制位 0：开启以太网功能 1：禁用以太网功能
SM8503（可读写）	MODBUS TCP 功能控制位 0：开启 MODBUS TCP 功能 1：禁用 MODBUS TCP 功能
SM8080~SM8087（可读写）	发生超时的从站 IP，依次对应 MODBUS TCP 配置表格里的 IP1~IP8 的超时，置位后表示对应 IP 超时，需要多次判断超时时可进行手动复位

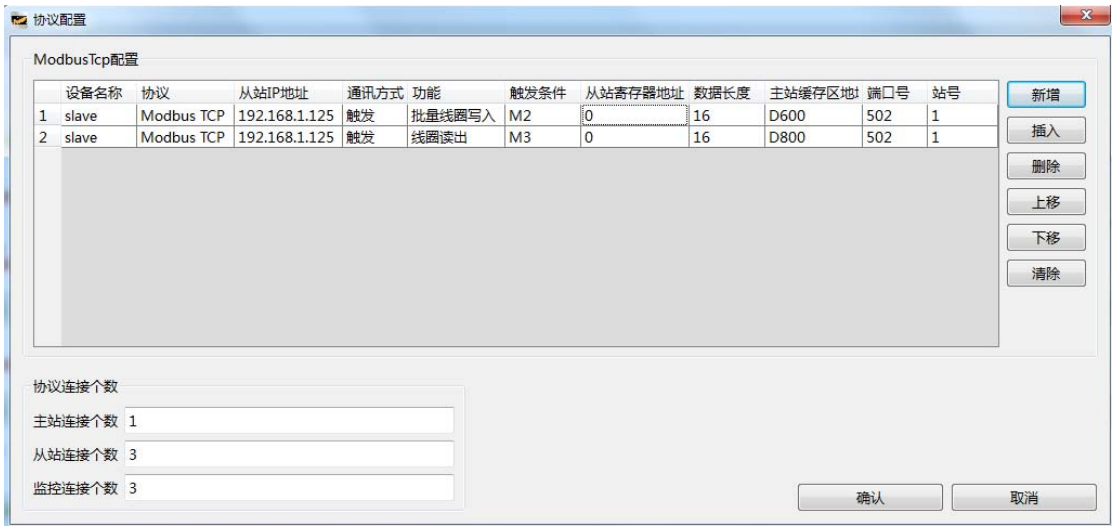
SD8500（只读）	显示 PLC 当前分配协议个数 0：表示有 3 个监控，4 个从站，0 个主站 16 进制数：11-8bit，7-4bit，3-0bit 组成的四位数分别代表监控、从站、主站个数 （SD8500 = H340 表示监控 3 个，从站 4 个，主站 0 个）
SD8501（可读写）	不在线从站对应的 IP（第四段），当 SM8500 置位时查看，此时需要重新打开服务器（从设备）才能再次建立通信，通信建立成功后会清除该寄存器，如果值不为 0，可能是还有其他的设备未正常连接或者该设备重启失败，默认值 0
SD8502（可读写）	显示 MODBUS TCP 错误码 0：无错误 -1：SD8500 值设置不正确，此时会使用默认连接配置 PLC 作从站时的报错（错误码组成：高八位为站号，低八位为出错类型）：此情况表示主站发送给 PLC 的数据帧错误 出错类型： 0x01 表示非法功能 0x02 表示非法数据地址，从地址设置错误 0x03 表示非法数据值，数据长度设置错误或者主站缓冲区数据设置错误 PLC 作主站时的报错（错误码组成：高八位为 IP，低八位为出错类型）：此情况表示从站返回给 PLC 的数据帧错误 出错类型： 0x11 表示非法功能 0x12 表示非法数据地址 0x13 表示非法数据值
SD8503（可读写）	设定超时时间（ms 单位），默认 0 时代表设定超时时间为 1000ms
SD8504（只读）	显示当前 TCP 连接数量（主+从+监控的总个数）

SD8505（不允许使用）	烧录 mac 地址专用寄存器，为-1 时，表示需要烧写 mac 地址，
SD8506（可读写）	显示最新的接收超时的从站 IP 及该 IP 对应的超时功能码所在表格位置（高八位为超时 IP，低八位为功能码位置，例：0x4901 表示超时 IP 为 73，超时的功能码为该 IP 的第一个读或写功能）
SD8507（可读写）	尝试重连时间间隔（从站掉线后每次主站尝试重连从站的时间间隔，ms 单位），为 0 时，默认为 100ms
SD8508	本机 IP 第一段
SD8509	本机 IP 第二段
SD8510	本机 IP 第三段
SD8511	本机 IP 第四段

5、MODBUS TCP 触发方式协议配置举例

用触发轮询方式对 IP 为 192.168.1.125 的从站进行线圈读写操作。

（1）下载协议配置



（2）编写梯形图操作触发条件，使其可以按照程序流程进行通信





（3）启动 M20，可看到主站 PLC 上的输出 LED 开启跑马灯模式。

（4）通信流程解析

①PLC 会根据配置表中的配置功能一行行顺次执行，执行完整个表后就再次从第一行配置顺次执行。

②哪个触发条件使能了就执行哪一条配置行的功能，也是遵循①中的执行顺序。

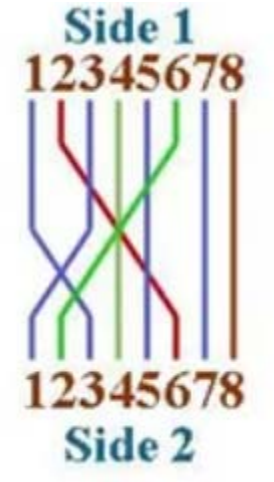
③触发功能执行完了或者发生了超时现象后会复位触发位，可根据此触发位的复位来判断当前配置行执行结束，然后用沿跳变方式来执行下一个功能配置行，按此方式可完成整个表的触发轮询。

④如果配置表中选择循环功能时，PLC 会一直按配置表顺次执行，一直进行通信功能，触发方式则可以根据用户的流程需求来选择什么时候进行通信。

6、以太网功能使用注意事项

（1）当 PLC 之间不能正常 MODBUS TCP 通信时，首先检查 PLC 的 IP 问题，是否有 IP 冲突或 IP 配置不正确，再检查网线是否正常、网络是否稳定，再检查配置表是否配置出错

（2）PLC 之间的网线直连通信，中间通过交换机连接或者采用交叉网线方式，交叉网线接线方式如下



（3）循环方式和触发方式的使用建议如下

• 循环通信:需要反复、尽快读写从站内变化比较快的数据，例如读取变频器的运行频率、运行状态、输入端口状态;在过程控制应用系统中，实时通信改写变频器的运行频率、输出端口的状态，等等，可选择“循环”通信方式。PLC 在执行用户程序时，会反复扫描执行通信配置表中的所有“循环” 配置项。

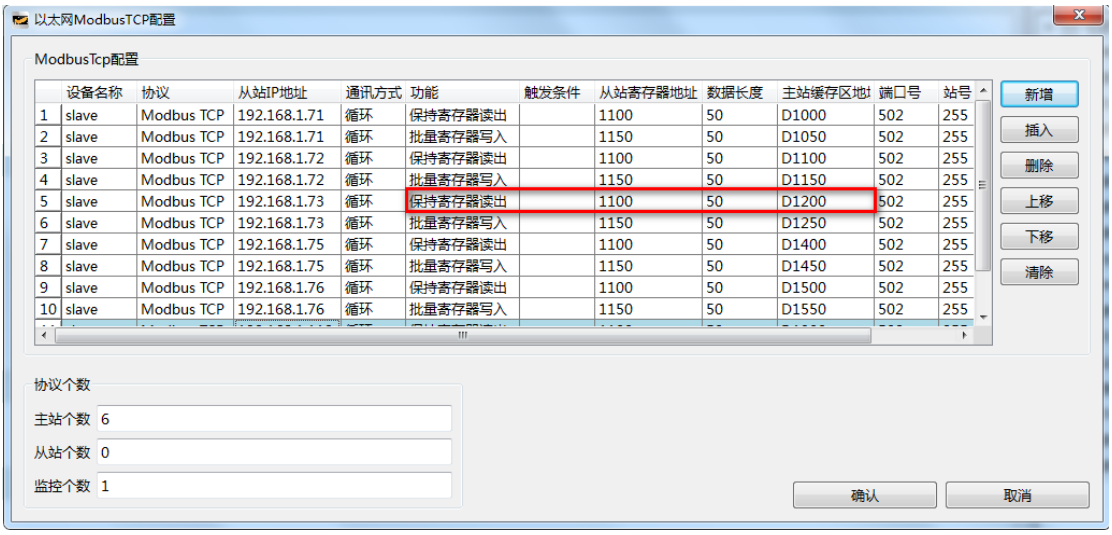
• 触发通信:需要定时读或写从站的刷新速度较慢的数据，例如读取变频器的输出电流、输出功率、当前故障告警信息，等等，可选择“触发”通信方式，在用户程序中,每置位触发条件 1 次，即会引起通信配置表中对应的通信项的 1 次通信操作;用户程序中定时置位该触发条件，即可实现所需频度的通信读写操作。

• 通信方式设定的建议:根据所需要交互参数刷新的特性要求进行合理选择，可使得通信表现大为改善，不要为了编程简单，将所有通信项都设置为“循环”项，这样可能因循环项太多，而使得数据交互及时性降低，影响系统的控制效果。将一些不重要的数据访问，安排为“触发”项，根据轻重缓急来触发通信，可大为改善通信及时性效果。

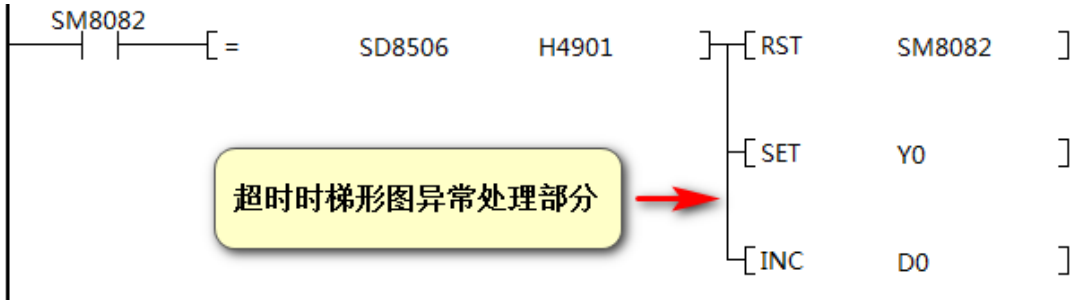
（4）通信超时时的异常处理

是通信就会出错导致超时,如果使用者使用以太网通信会发生的错误会对设备运行产生重大事故，举例说明规避方法：

IP73 的第一个功能保持寄存器读出功能读出的数据是特别重要的数据，不允许超时导致数据不刷新而造成设备运行故障，则处理方法如下。



IP73 对应的是第三个 IP，则超时标志是 SM8082，超时时 SD8506 高八位显示 IP，低八位显示超时的功能数据所在该 IP 下的位置，异常处理梯形图如下



3.7 BD 板模块的使用

3.7.1 BD 板的 ID 及版本说明

1、可用 BD 板模块及对应 ID 如下。

特殊寄存器	数值	说明
SD8219	H1240	数字量，2DI4DO
	H1600	数字量，6DI
	H1060	数字量，6DO
	H2401	模拟量（电流），4AI
	H2202	热电阻，2RTD
	H2210	模拟量（电压），2AI1AO
	H230A	（电流+RTD），1AI1RTD
	H2020	模拟量 2AO

2、BD 板版本寄存器

SD8220	100	10 进制 100 表示当前 BD 板软件的版本号为 V1.00

3.7.2 数字量 BD 板使用说明

数字量的 DI0-DI7 点对应 PLC 的 X370-X377，DO0-DO7 点对应 PLC 的 Y370-Y377。上电的时候根据 ID 可以得到各个槽对应的 X/Y 的编号。

DI0-DI7	DO0-DO7
X370-X377	Y370-Y371

3.7.3 模拟量 BD 板使用说明

1、BD 板配置命令说明

命令形式：0xDCBA		说明
15~12bit	0-1	0000：配置第2通道为两线制/四线制 0001：配置第2通道三线制接法
11~8bit	0-1	0000：配置第1通道为两线制/四线制 0001：配置第1通道三线制接法
7~4bit	0-3	0000：配置第2通道为PT100类型 0001：配置第2通道为PT200四种类型

		0010：配置第2通道为PT1000四种类型 0011：配置第2通道为Ni1000四种类型
3~0bit	0-3	0000：配置第1通道为PT100类型 0001：配置第1通道为PT200四种类型 0010：配置第1通道为PT1000四种类型 0011：配置第1通道为Ni1000四种类型

举例：0x0011 表示配置第 1、2 通道为 PT200 输入，注意写配置后需要置位配置使能标志 SM8179

2、BD 板状态寄存器 SD8221 说明

地址	分别表示两个BD板的运行状态
bit0	1: 通信错误　0:正常
bit1	1：配置错误　0:正常
bit2	1: 命令错误　0:正常
bit3	1: 未校准　　0:正常
Bit4~15	预留

3.7.4 模拟量 BD 板地址对照表

1、4AI BD 板的地址对应表

地址	说明
SD8219	H2401，表示 4AI BD 板
SD8220	100，表示版本号 V1.00
SD8221	BD 板状态
SD8222	显示输入第 1 通道数值
SD8223	显示输入第 2 通道数值
SD8224	显示输入第 3 通道数值
SD8225	显示输入第 4 通道数值

2、2RTD BD 板的地址对应表

地址	说明
SD8219	H2202，表示 2RTD BD 板
SD8220	100，表示版本号 V1.00
SD8221	BD 板状态
SD8222	显示 RTD 输入第 1 通道数值
SD8223	显示 RTD 输入第 2 通道数值
SD8224	配置寄存器
SM8179	配置命令使能

3、2AI1AO BD 板的地址对应表

地址	说明
SD8219	H2210，表示 2AI1AO BD 板
SD8220	100，表示版本号 V1.00
SD8221	BD 板状态
SD8222	显示输入第 1 通道数值
SD8223	显示输入第 2 通道数值
SD8224	显示输出第 1 通道数值

4、1AI1RTD BD 板的地址对应表

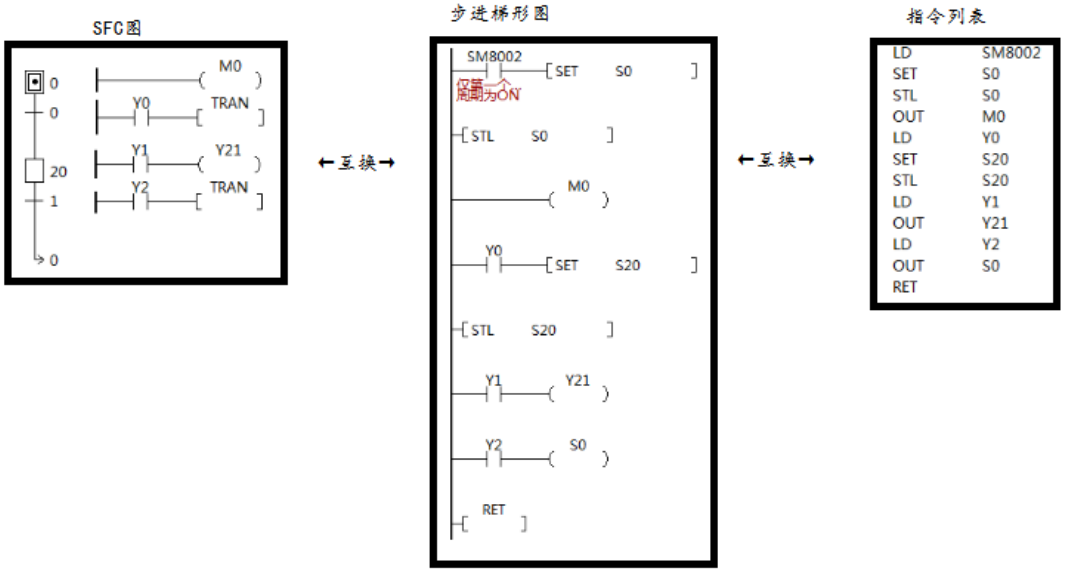
地址	说明
SD8219	H230A，表示 1AI1RTD BD 板
SD8220	100，表示版本号 V1.00
SD8221	BD 板状态
SD8222	显示输入第 1 通道数值
SD8223	显示 RTD 输入第 1 通道数值
SD8224	配置寄存器
SM8179	配置命令使能

本章主要说明使用 UNISYS 进行“SFC”和“步进梯形图”编程的要领以及顺控动作。

3.8 SFC 程序和步进梯形图

3.8.1 概要

在 X 系列可编程控制器中，可以使用 SFC 图(顺序功能图)或步进梯形图实现顺控，把状态 S 当作 1 个控制工序，在其中编写输入条件和输出控制的顺控程序。由于工序推进时，前工序就转为不动作，所以可以按各工序的简单的顺序来控制机械，使顺控的设计变得简单。此外，SFC 程序和步进梯形图指令，都是按照既定的规则进行编程的，所以可以相互转换，因此，实质内容完全一致，也可以作为熟悉的继电器梯形图使用。SFC 图、步进梯形图、指令表转换举例如下：



3.8.2 相关指令动作说明

1、STL 指令：状态 ON 后，通过 STL 触点，使与其连接的梯形图（内部梯形图）动作。

2、TRAN 指令：用于状态转移，转移到下一个状态。

3、RET 指令：放在步进顺控程序最后面。

4、OUT 指令：工序（状态步）中对 Y、M 进行输出置位后，在下一次状态转移时会被复位，但在下一个工序中仍然可置位，然后再下一次状态转移复位；OUT 后接的是 T 元件，则状态切换后，T0~T245、T256~T511 的数据、触点、线圈全部复位，T246~T255 保持转移前的数据值和触点状态；OUT 后接的是 C 元件，状态切换后，保持转移前的数据值和触点状态；步进梯形图中，OUT S 表示跳转到对应的状态。

5、SET 指令：工序（状态步）中对 Y、M 进行 SET 置位后，在下一次状态转移时仍然保持原来的 ON 输出；步进梯形图中 SET S 表示转移到对应的状态。

6、RST 指令：工序（状态步）中对 Y、M 进行 RST 复位后，在后面的工序中仍可置位；复位 T、C，数据值、触点、线圈都会被复位，但会保持 T246~T255、C200~C255 的复位状态；步进梯形图中 RST S 表示复位对应的状态。

3.8.3 相关特殊辅助寄存器

寄存器	说明	内容
SM8046	STL 状态动作（在执行 END 指令时处理）	当 SM8047 接通时，S0 ~S899、S1000~S4095 中任意一个为 ON 则接通
SM8047	STL 监控有效（在执行 END 指令时处理）	SM8047 接通，SD8040~SD8047 有效
SM8048	信号报警器动作	当 SM8049 接通时，S900~S999 中任意一个为 ON 则接通
SM8409	信号报警器有效	SM8049 接通后，SD8049 的动作有效
SD8040	ON 状态编号 1	状态 S0~S899,S1000~S4095 中正在动作的状态的最小编号保存到 SD8040 中。其次为 ON 的状态保存到紧接着的寄存器，最大八点，默认-1
SD8041	ON 状态编号 2	
SD8042	ON 状态编号 3	
SD8043	ON 状态编号 4	
SD8044	ON 状态编号 5	
SD8045	ON 状态编号 6	
SD8046	ON 状态编号 7	
SD8047	ON 状态编号 8	
SD8049	ON 状态最小编号（SM8049 有效时）	SM8049 为 ON 时，保存信号报警继电器 S900~S999 中为 ON 的状态的最小编号 ，默认-1

3.8.4 SFC 编程说明

1、SFC 编程相关符号

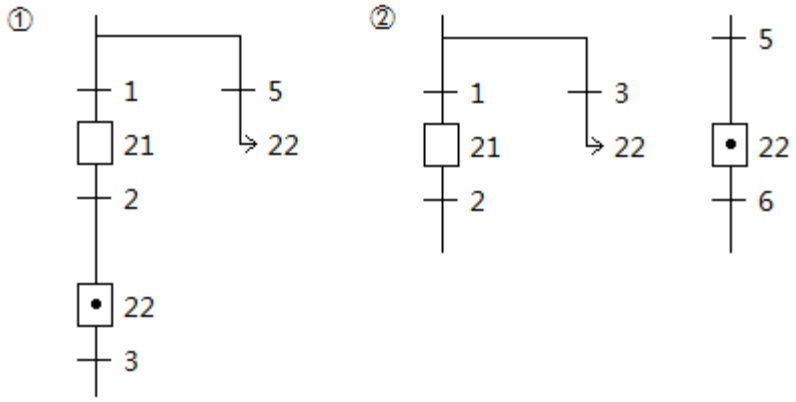
符号	说明	备注
	初始工序	初始工序里的状态只能是 S0~S9

	工序	非初始工序里的状态只能是 S10~S4095
	转移符	满足条件，跳转到下一个工序（状态步）
	选择分支	每个分支限制在 8 个回路以下
	选择汇合	
	并行分支	每个分支限制在 8 个回路以下
	并行汇合	
	跳转符	跳转到对应的状态步
	复位符	复位对应的状态位

2、SFC 流程的形态

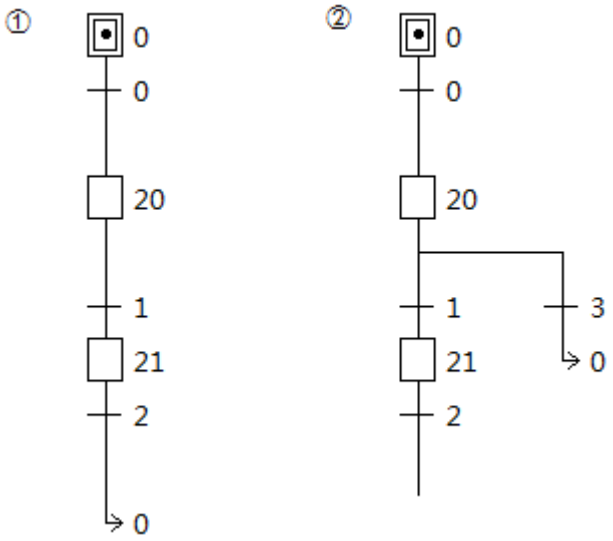
（1）流程跳转

跳转到下方的状态，或转移到流程外的状态，用跳转符表示要转移的目标状态。



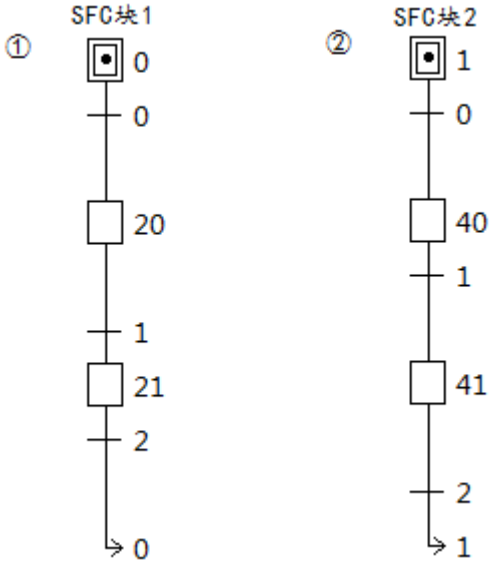
（2）流程重复

转移到上方，重新执行此流程称为重复，用跳转符表示要转移的目标状态。



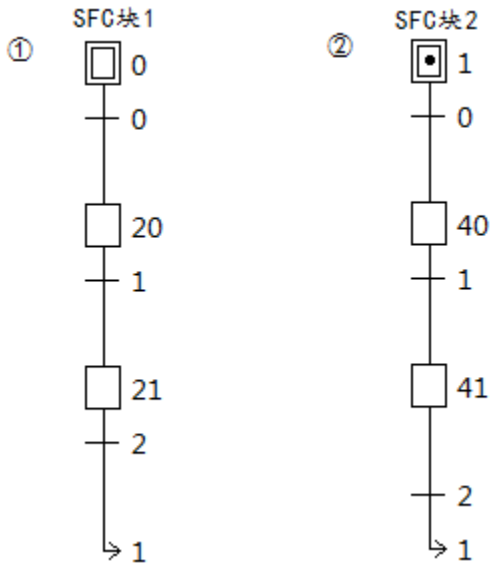
（3）流程分离

具有多个初始化状态的 SFC 程序，分成多个程序块编程。



（4）跳到流程外

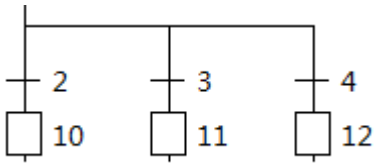
在分离程序块后制作的 SFC 程序之间也可以进行状态转移。



3、SFC 的分支、合并状态

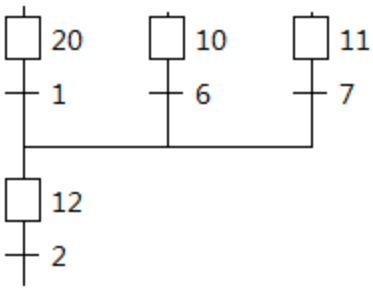
（1）选择分支

在分支后，编写转移条件。



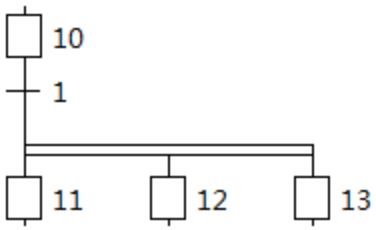
（2）选择汇合

先编写转移条件，再进行汇合。



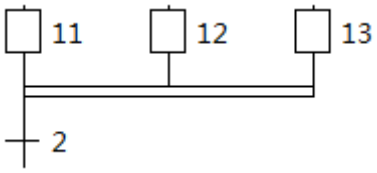
（3）并行分支

先编写转移条件，再进行分支。



（4）并行汇合

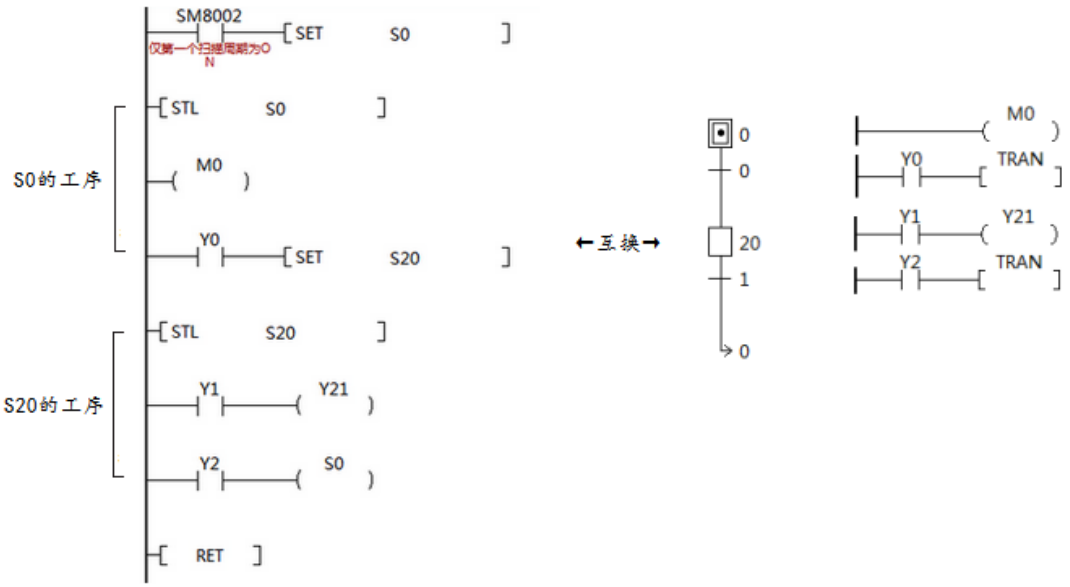
在汇合后，编写转移条件。



3.8.5 步进梯形图编程说明

1、与 SFC 图的转换关系

步进梯形图的实质与 SFC 是相同的，只是换成了继电器梯形图风格的表现形式。转换举例如下：



其中：

SET S 表示转移到对应的状态 S；

STL S 表示执行对应 S 的工序（内部梯形图）；

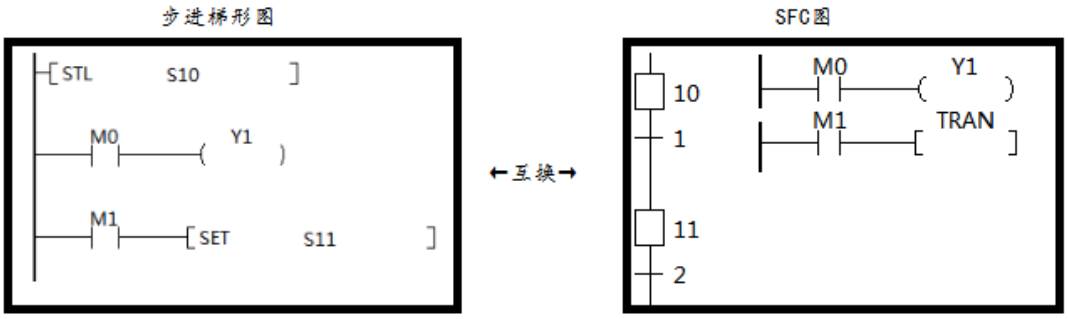
OUT S 表示跳转到对应的状态 S；

RET 放在步进梯形图的末尾，主要用于区别普通梯形图。

2、步进梯形图的流程形态

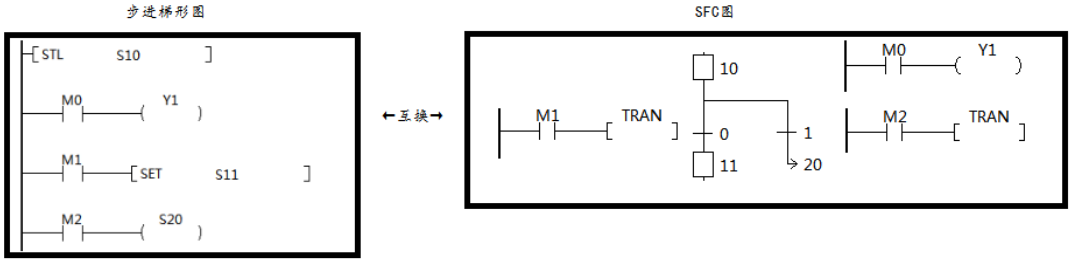
（1）转移目标

当 SET S 前面的条件成立时，就会转移到下一个对应的状态。



（2）跳转/重复

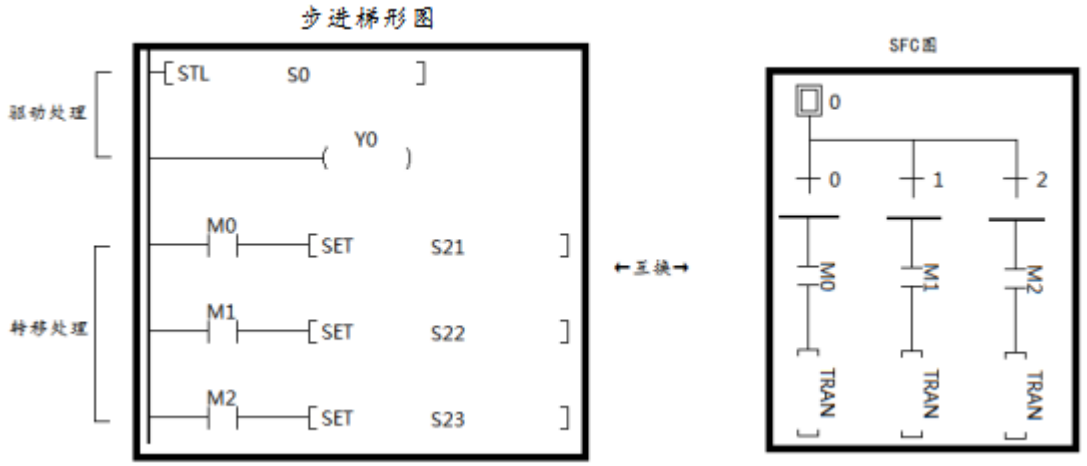
用 OUT S 表示要跳转/重复的状态。



3、步进梯形图的分支、合并状态

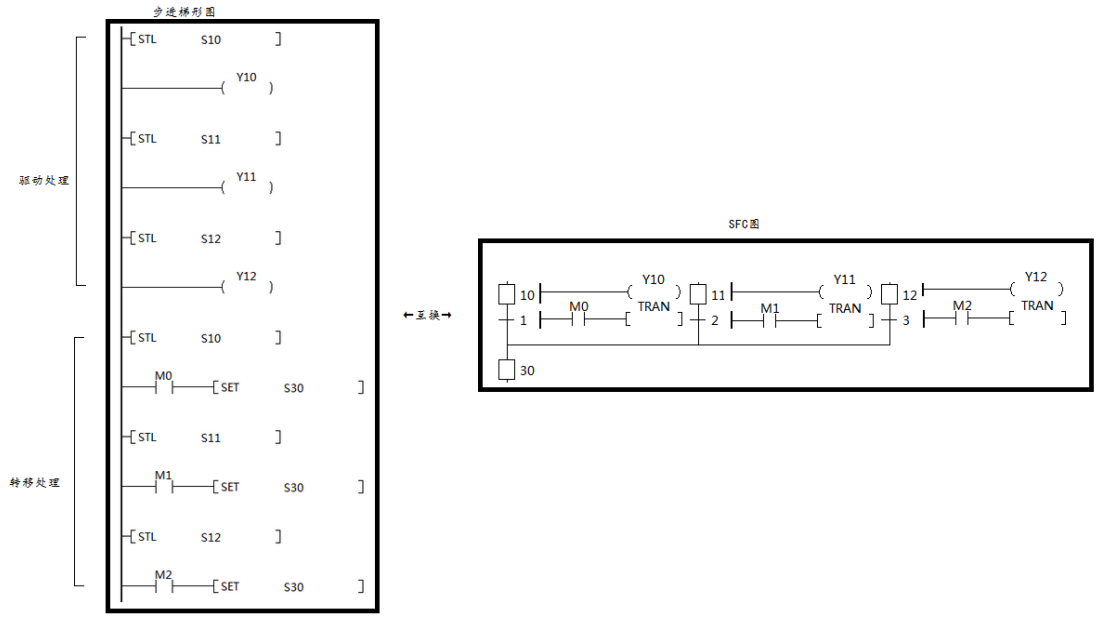
（1）选择分支

先执行驱动处理，接着执行转移处理，依次执行所有的转移处理。



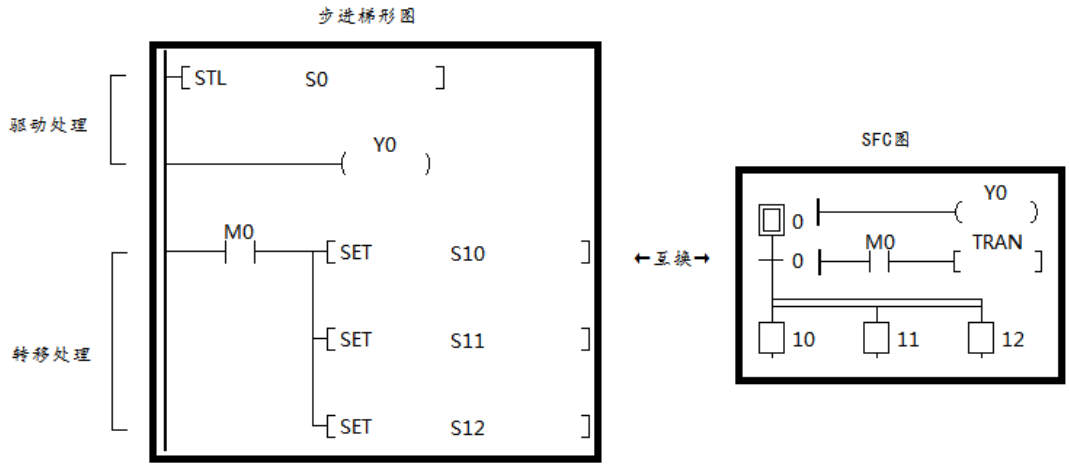
（2）选择汇合

先执行汇合前状态的驱动程序，再依次执行汇合状态的转移处理。



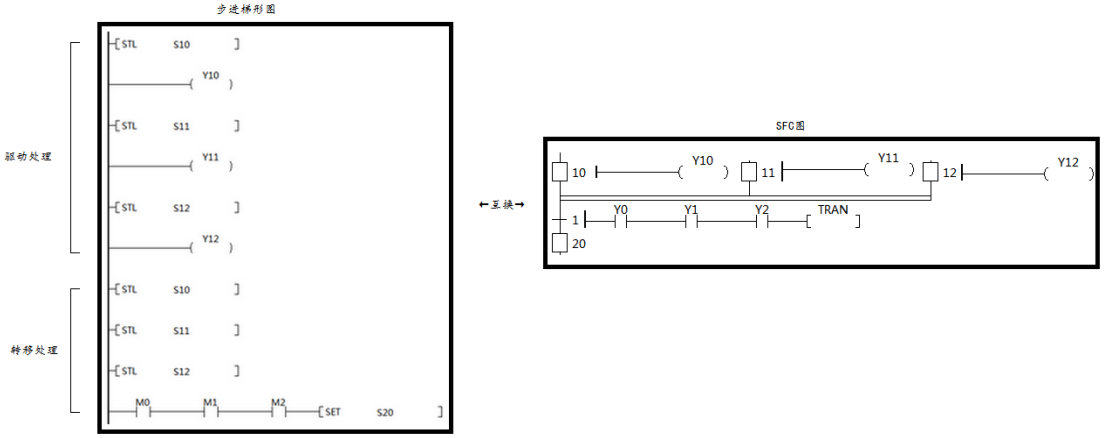
（3）并行分支

先执行驱动处理，接着执行转移处理，依次执行所有的转移处理。



（4）并行汇合

先执行汇合前状态的驱动程序，再依次执行汇合状态的转移处理。



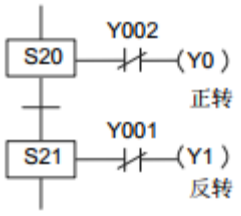
3.8.6 SFC 或步进梯形图编程注意事项

（1）初始工序状态编号只能是 S0~S9。

（2）状态 S 编号不可重复使用。

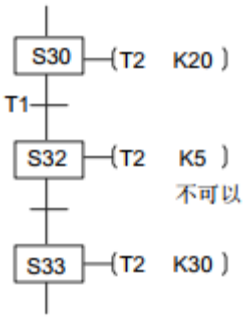
（3）输出的互锁

在状态转移过程中，只有一瞬间(1 个运算周期)两个状态会同时为 ON。因此，在不可以同时接通的一对输出之间，为了避免同时为 ON，在可编程控制器的外部设置互锁。此外，请同时在程序中执行如图所示的相互的互锁。



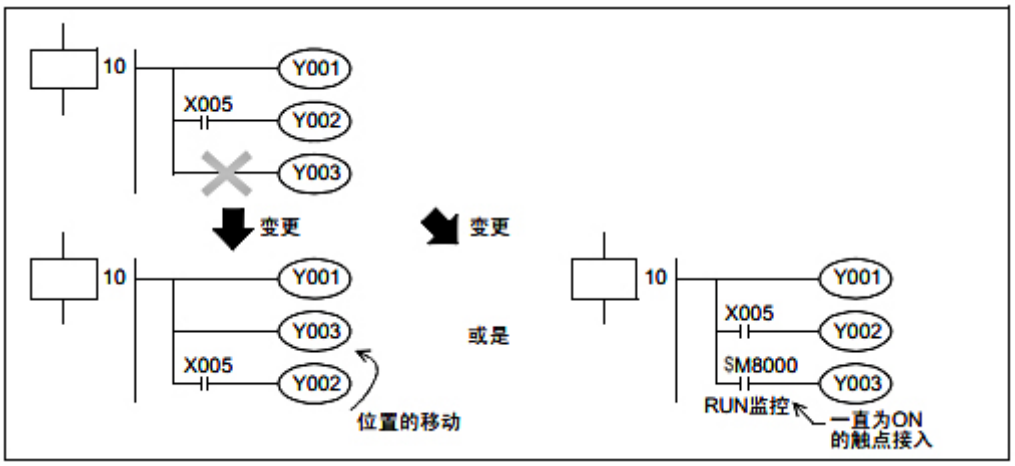
（4）定时器的重复使用

定时器线圈也与输出线圈相同，可以在不同的状态中对同一个软元件进行编程，但是在相邻的状态中不能编程。如果在相邻状态中编程，则工序转移时，定时器线圈不断开，当前值不会被复位。



（5）输出驱动

如下图所示，从状态内的母线开始，一旦写入 LD 或是 LDI 指令后，就不能再编写不需要触点的指令，请按照下图所示方法进行更改。



（6）可在状态内处理的顺控指令

状态 \ 命令		LD/LDI/LDP/LDF，AND/ANI/ ANDPIANDF，OR/ORIORF，INV, OUT，SET/RST，PLS/PLF	ANB/ORB MPS/MRD/MPP	MC/MCR
初始状态/一般状态		可使用	可使用	不可使用
分支，汇合状态	输出处理	可使用	可使用	不可使用
	转移处理	可使用	不可使用	不可使用

（7）在中断程序与子程序内，不能使用 STL 指令。

（8）在 STL 指令内不禁止使用跳转指令，但其动作复杂，不建议使用。

四、附录

4.1 错误码内容

错误类型	错误类型编码	错误代码编号	错误描述	错误级别	诊断信息
PLC 硬件错误	8061	6101	RAM 错误	Error	可能是上电时外部 flash 读写失败,或损坏
		6105	看门狗定时超时错误		采样(运算时间)超出了 SD8000 的值。请确认程序。
		6116	栈错误		程序中 P 标签的存储出现问题
参数错误	8064	6402	内存容量的设定错误		检查系统参数区设定与当前 PLC 支持的参数范围是否一致
		6403	保持区域的设定错误		
		6404	注释区域的设定错误		
		6409	RUN 端子设定错误		
语法错误	8065	6501	指令-软元件符号-软元件的编号组合错误		出现此错误是指令的使用方法不正确，请检查指令的使用方法，也有可能是程序下载过程中突然断电导致程序 不完整
		6504	标签编号重复		
		6505	软元件编号超出范围		
		6506	使用了未定义的指令		
		6508	中断输入（I）的定义错误		
		6509	程序错误		
		6510	MC 的嵌套编号的大小关系错误		
回路/流程错误	8066	6610	LD、LDI 连续使用次数超过 9 次		作为回路块整体的指令组合方法有不正确的地方时，或者成对的指令的关系不正确时，会发生这样的错误，检查指令的相互关系
		6617	应该从母线开始的指令没有连接在母线上		
		6618	只能在主程序中使用的指令在主程序以外		
		6619	FOR-NEXT 之间有不能使用		

			的指令：STL、RET、MC、MCR、I、IRET		
		6620	FOR-NEXT 嵌套超出（最大 6 层）		
		6621	FOR NEXT 数量关系错误		
		6623	无 MC 指令		
		6624	无 MCR 指令		
		6625	STL 的连续使用次数超出 9 次		
		6626	STL-RET 之间有不能使用的指令：MC、MCR、I、SRET、IRET		
		6627	无 STL 指令		
		6629	无 P、I		
		6634	CALL 的嵌套超出 6 个		
		6637	分支错误		
运算错误	8067	6701	①没有 CJ、CALL 的跳转目标地址②变址修饰的结果越界或不正确	Warning	这些是在运算执行过程中产生的错误，请修改程序或是检查应用指令的操作数的内容。 即使没有发生语法、回路错误，例如，软元件变址超过其最大范围
		6705	应用指令中的操作数在规定之外的软元件		
		6706	应用指令的操作数的软元件编号范围或数据的值超出		
		6710	参数之间的不匹配		在移位指令等中，存在源操作数和目标操作数重复的情况
		6730	采样时间(Ts)为对象范围以外(Ts≦0)		在控制参数的设定值或 PID 运算中出现数据错误，请检查参数的内容。
		6732	输入滤波器常数(α)为对象范围以外(α<0 或 100≦α)		
		6733	比例增益(KP)为对象范围		

			以外(KP<0)		
		6734	积分时间(TI)为对象范围以外(TI<0)		
		6735	微分增益(KD)为对象范围以外(KD<0 或 201≦KD)		
		6736	微分时间(TD)为对象范围以外(TD<0)		
		6742	测量值变化量超出(ΔPV<-32768 或是 32767<ΔPV)		各参数在最大值或最小值下继续运行
		6743	偏差超出(EV<-32768 或是 32767<EV)		
		6747	PID 运算结果超出(-32768～32767 以外)		
		6748	PID 输出上限设定值<输出下限设定值		请确认对象的设定内容是否正确
		6749	PID 输入变化量报警设定值、输出变化量报警设定值异常(设定值<0)		
		6764	输入或输入端子被其他功能占用		可能是其他计数器或脉冲输出指令占用了端子
		6770	文件寄存器指令写入错误		使用文件寄存器写入指令之前没有对其进行初始化导致，先使用相关指令初始化
		6774	对应的高速计数器未使能		
		6775	一个高速计数器最多可以使用 6 条比较指令		
		6776	每种表格比较指令只能使用一个		
串行通信错误	/	6201	串行通信奇偶校验、溢出、帧错误		
		6203	串行通信数据校验错误		

		6204	串行通信数据格式错误		