

SL1S 使用说明书

感谢您使用德天奥科技生产的 SL 系列可编程控制器。

安全注意事项：

- 1: 在进行微型可编程控制器的安装，保养，检修之前，请务必熟读此使用手册和其它相关书籍，并正确使用。
- 2: 可编程控制器的安装位置尽量远离高电压，强电流，高频率等对周围有较强干扰的设备。
- 3: 请一定在可程式控制器外部组成紧急停止电路，制止正反转同时运行的连锁电路，上下限定位连锁电路。
- 4: 对于变更运行中的程序，强制输出，**RUN**，**STOP** 等操作，必需熟读使用手册，充分确认其安全之后进行。
- 5: 关于 32MR 和 32MT，编程时有使用 32 点时，请选择 PLC 类型 **FX1N**，下载程序时变更为 **FX1S**。
- 6: 在配线安装时，不要将铁屑，电线屑落入可编程控制器内部，否则有火灾及损坏可编程控制器的危险。
- 7: 请不要将可编程控制器安装在阳光直射或雨水溅到的地方。
- 8: 本手册为随机发送的附件,请交付给最终用户,以便需要时查阅。

性能规格:

项目		性能
运算控制方式		存储程序反复运算方式
编程语言		继电器符号+步进梯形图方式
程序内存		内置 2KEEPROM (含文件寄存器注释最大 2K 步)
指令种类		基本指令 27 条 步进梯形图指令 2 条 应用指令 27 条
运算处理速度		0、55-0、10US(基本指令) 10-50US(应用指令)
输入输出点数	SL1S-16MR	X000—X007 8 点(8 进制) Y000—Y007 8 点(8 进制) (可扩展输入输出模块)
	SL1S-20MR	X000—X013 12 点(8 进制) Y000—Y007 8 点(8 进制) (可扩展输入输出模块)
	SL1S-32MR SL1S-32MT	X000—X017 16 点(8 进制) Y000—Y017 16 点(8 进制) (可扩展输入输出模块)
辅助继电器 (M)		M0—M383(一般用) M384—M511(保持用)
状态继电器 (S)		S0—S127(全点停电保持)
定时器 (T)		T0—T63(延时 NO)
计数器 (C)		C0—C31(0—32767 秒)
数据寄存器 (D)		D0--D127(一般用) D128--D255(保持用)

基本逻辑指令一览表

助记符、名称	功能	可用软元件	程序步
LD 取	常开触点逻辑运算开始	X,Y,M,S,T,C	1
LDI 取反	常闭触点逻辑运算开始	X,Y,M,S,T,C	1

LDP 取脉冲上升沿	上升沿检出运算开始	X,Y,M,S,T,C	2
LDF 取脉冲下降沿	下降沿检出运算开始	X,Y,M,S,T,C	2
AND 与	常开触点串联连接	X,Y,M,S,T,C	1
ANI 与非	常闭触点串联连接	X,Y,M,S,T,C	1
ANDP 与脉冲上升沿	上升沿检出串联连接	X,Y,M,S,T,C	2
ANDF 与脉冲下降沿	下降沿检出串联连接	X,Y,M,S,T,C	2
OR 或	常开触点并联连接	X,Y,M,S,T,C	1
ORI 或非	常闭触点并联连接	X,Y,M,S,T,C	1
ORP 或脉冲上升沿	上升沿检出并联连接	X,Y,M,S,T,C	2
ORF 或脉冲下降沿	下降沿检出并联连接	X,Y,M,S,T,C	2
ANB 块与	并联回路块的串联连接		1
ORB 块或	串联回路块的并联连接		1
OUT 输出	线圈驱动	Y,M,S,T,C	注 1
SET 置位	动作保持	Y,M,S	注 2
RST 复位	清除动作保持, 寄存器清零	Y,M,S,T,C,D,V,Z	
PLS 上升沿脉冲	上升沿输出	Y,M (特殊 M 除外)	1
PLF 下降沿脉冲	下降沿输出	Y,M (特殊 M 除外)	1
MC 主控	公共串联点的连接线圈指令	Y,M (特殊 M 除外)	3
MCR 主控复位	公共串联点的消除指令		2
MPS 压栈	运算存储		1
MRD 读栈	存储读出		1
MPP 出栈	存储读出与复位		1
INV 取反	运算结果的反转		1
NOP 空操作	无动作		1
END 结束	输入输出及返回到开始		1

- 软元件为 Y 和一般 M 的程序步为 1, S 和特殊辅助继电器 M 的程序步为 2, 定时器 T 的程序步为 3, 计数器 C 的程序步为 3-5。
- 软元件为 Y 和一般 M 的程序步为 1, S 和特殊辅助继电器 M、定时器 T、计数器 C 的程序步为 2, 数据寄存器 D 为 3。

[LD],[LDI],[LDP],[LDF],[OUT] 指令

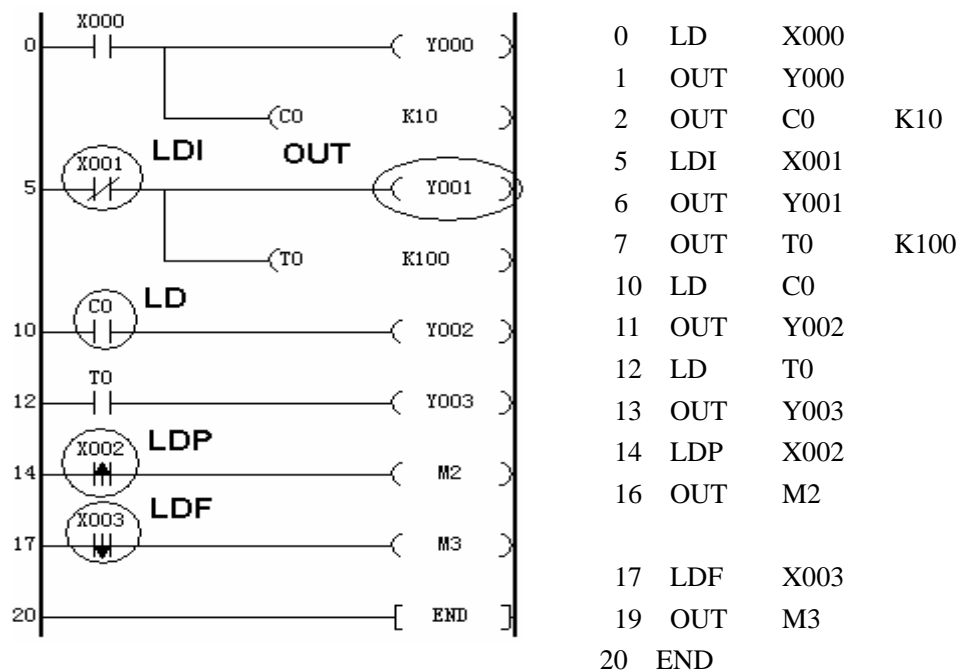
SL1S

助记符、名称	功能	可用软元件	程序步
LD 取	常开触点逻辑运算开始	X,Y,M,S,T,C	1
LDI 取反	常闭触点逻辑运算开始	X,Y,M,S,T,C	1

LDP 取脉冲上升沿	上升沿检出运算开始	X,Y,M,S,T,C	2
LDF 取脉冲下降沿	下降沿检出运算开始	X,Y,M,S,T,C	2
OUT 输出	线圈驱动	Y,M,S,T,C	见说明

- LD,LDI,LDP,LDF 指令将触点连接到母线上。多个分支用 ANB,ORB 时也使用。
- LDP 指令在上升沿（软元件由 OFF 到 ON 变化时）接通一个周期；LDF 指令在下降沿（软元件由 ON 到 OFF 变化时）接通一个周期。
- LD,LDI,LDP,LDF 指令的重复使用次数在 8 次以下。即与后面的 ANB,ORB 指令使用时串并连使用的最大次数为 8 个。
- 软元件为 Y 和一般 M 的程序步为 1，S 和特殊辅助继电器 M 的程序步为 2，定时器 T 的程序步为 3，计数器 C 的程序步为 3—5。
- OUT 指令各种软元件的线圈驱动，但对输入继电器不能使用。并列的 OUT 可多次连续使用。
- OUT 指令驱动计数器时，当前面的线圈从 ON 变成 OFF，或者是从 OFF 变成 ON 时，计数器才加一。

SL1S 编程示例



- 用 LD,LDI,LDP,LDF 指令与母线连接。输出使用 OUT 指令驱动线圈。
- 使用 OUT 指令驱动定时器的计时线圈或者计数器的计数线圈时，必须设定定时和计数的时间和计数的值，可以是常数 K，或者由数据寄存器间接指定数值。

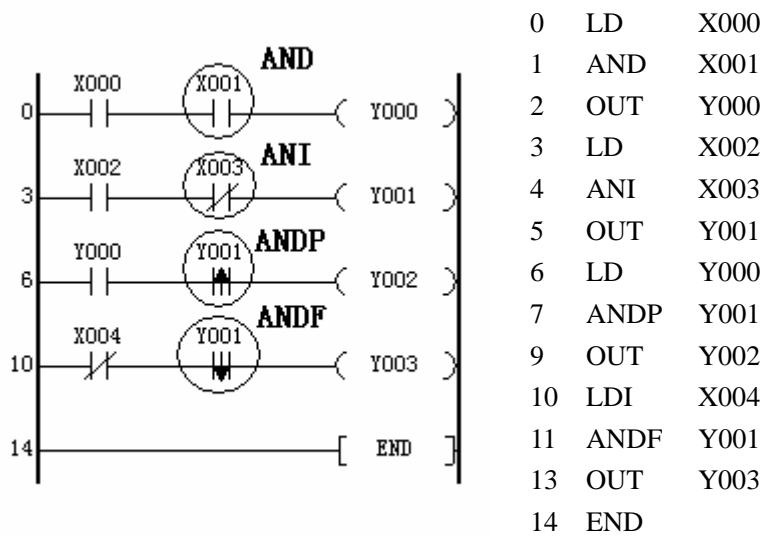
[AND],[ANI],[ANDP],[ANDF] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
AND 与	常开触点串联连接	X,Y,M,S,T,C	1
ANI 与非	常闭触点串联连接	X,Y,M,S,T,C	1
ANDP 与脉冲上升沿	上升沿检出串联连接	X,Y,M,S,T,C	2
ANDF 与脉冲下降沿	下降沿检出串联连接	X,Y,M,S,T,C	2

- AND,ANI,ANDP,ANDF 指令只能串接一个触点，两个以上的并联回路串联时使用后面的 ANB 指令。串联次数不受限制。
- ANDP,ANDF 指令在上升沿（即软元件由 ON 到 OFF 变化时）和下降沿即（软元件由 OFF 到 ON 变化时）接通一个周期。

SL1S 编程示例



- 实例中 X001, X003, Y001 作为串联触点与前面的触点相连。

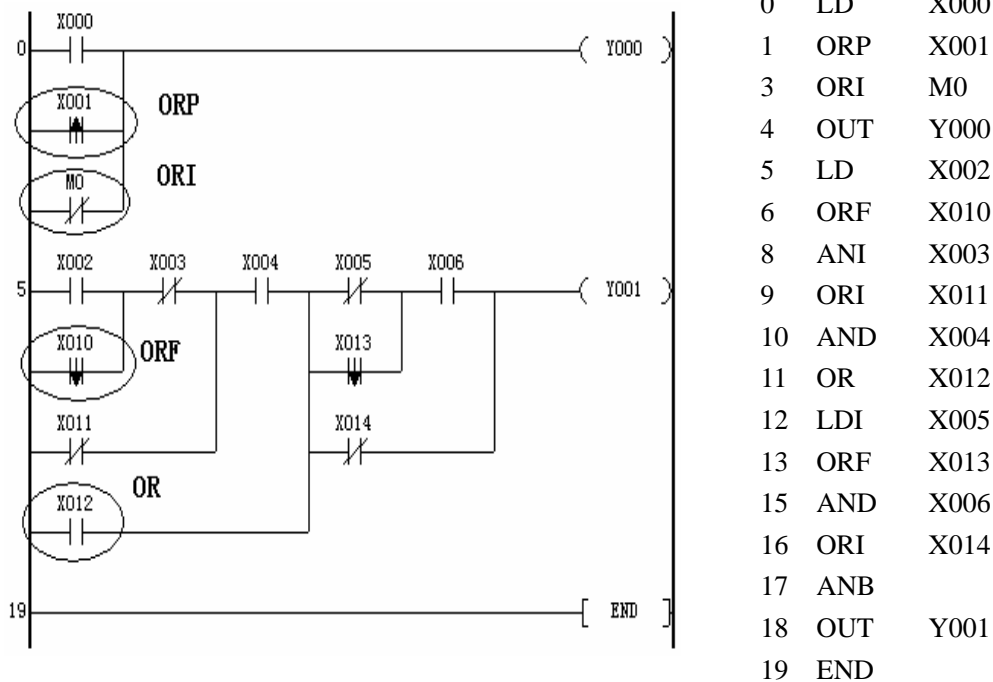
[OR],[ORI],[ORP],[ORF] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
OR 或	常开触点并联连接	X,Y,M,S,T,C	1
ORI 或非	常闭触点并联连接	X,Y,M,S,T,C	1
ORP 或脉冲上升沿	上升沿检出并联连接	X,Y,M,S,T,C	2
ORF 或脉冲下降沿	下降沿检出并联连接	X,Y,M,S,T,C	2

- OR,ORI,ORP,ORF 指令只能并联一个触点，两个以上的串联回路并联时使用后面的 ORB 指令。
- ORP,ORF 指令在上升沿（即软元件由 OFF 到 ON 变化时）和下降沿（即软元件由 ON 到 OFF 变化时）接通一个周期。
- OR,ORI,ORP,ORF 指令和前面的 LD,LDI,LDPLDF 指令一起使用，并联次数不受限制。

SL1S 编程示例



- 使用 OR,ORI,ORP,ORF 与前面的 LD,LDI,LDPLDF 并联连接,在程序步 12 到 16 中,由于是两个并联回路块的串联,所以使用 ANB 指令,关于 ANB 指令详见后面的说明。

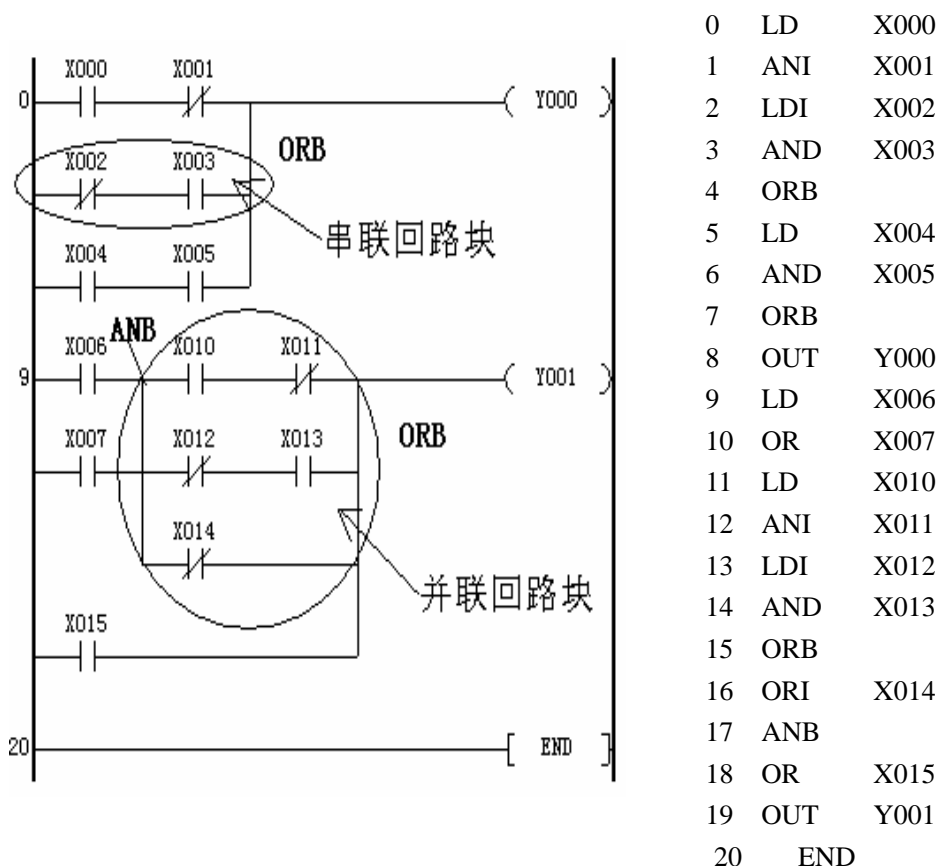
[ANB],[ORB] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
ANB 块与	并联回路块的串联连接		1
ORB 块或	串联回路块的并联连接		1

- 当多分支回路与前面的回路串联连接时，使用 ANB 指令。分支以 LD,LDI,LDP,LDF 指令作为起点，使用 ANB 指令与前面以 LD,LDI,LDP,LDF 指令作为起点的分支串联连接。
- 当 2 个以上的触点串接的串联回路块并联连接时，每个分支使用 LD,LDI 指令开始，ORB 指令结束。
- ANB,ORB 指令都是不带软元件的指令。
- ANB,ORB 使用的并串联回路的个数不受限制，但是当成批使用时，必须考虑 LD,LDI 的使用次数在 8 次以下。

SL1S 编程示例



- 在每个分支的最后使用 ORB 指令，不要在所有的分支后面使用 ORB 指令，如程序步 4 和 7 所示。

- ORB 和 ANB 指令只是对块的连接，如果不是块就不能使用，如程序步 16 和 18 不是块就不能使用。如图所示，串联回路块和并联回路块的示例。

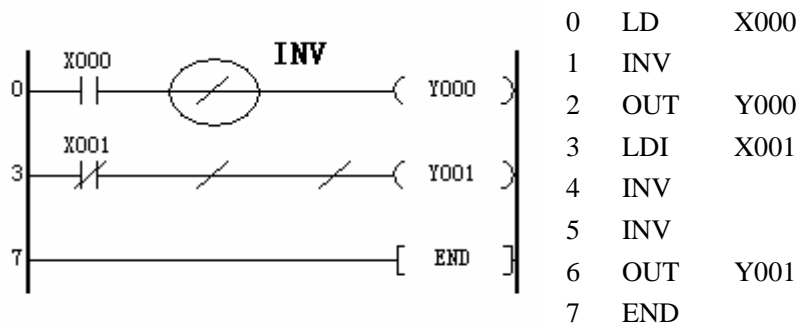
[INV] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
INV 取反	运算结果的反转		1

- INV 指令是将 INV 指令之前，LD,LDI,LDP,LDF 指令之后的运算结果取反的指令，没有软元件。

SL1S 编程示例



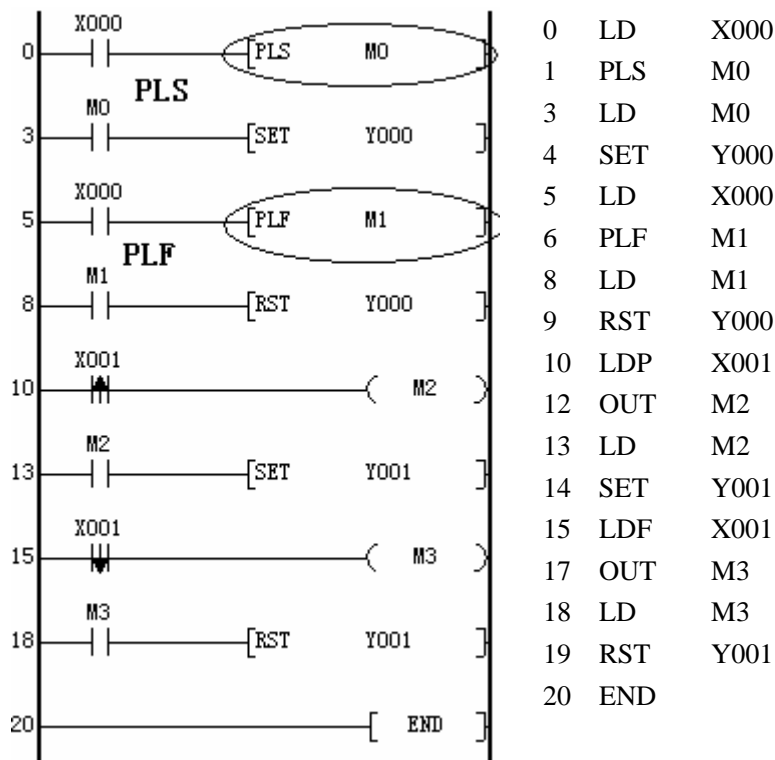
[PLS],[PLF] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
PLS 上升沿脉冲	上升沿输出	Y,M (特殊 M 除外)	1
PLF 下降沿脉冲	下降沿输出	Y,M (特殊 M 除外)	1

- 使用 PLS 指令时，只在线圈由 OFF 变成 ON 的一个扫描周期内，驱动软元件。
- 使用 PLF 指令时，只在线圈由 ON 变成 OFF 的一个扫描周期内，驱动软元件。
- 对具有停电保持功能的软元件，它只在第一次运行时产生脉冲动作。

SL1S 编程示例



- 程序段 0—2 和 10—12 的动作相同，都是在线圈闭合的上升沿，驱动一个扫描周期的输出。同样，程序段 5—7 和 15—17 的动作相同，都是在在线圈闭合的下降沿，驱动一个扫描周期的输出。
- 关于 SET,RST 指令的作用详见后面的说明。

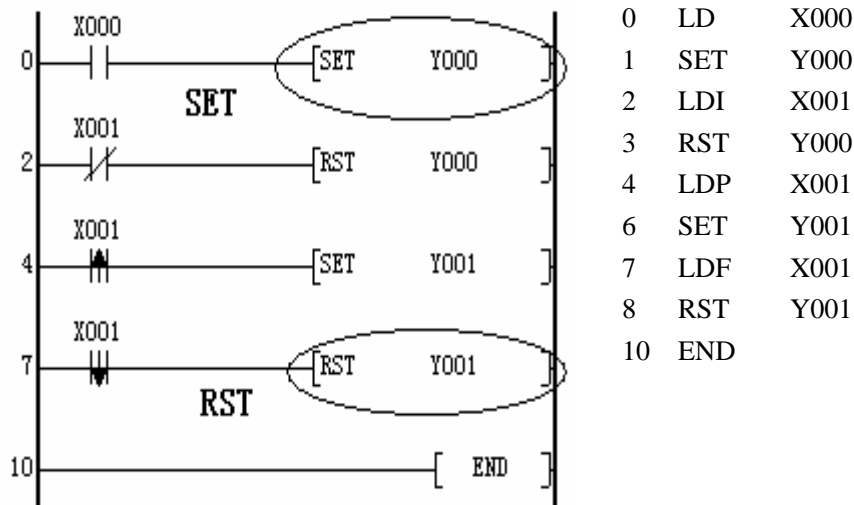
[SET],[RST] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
SET 置位	动作保持	Y,M,S	见说明
RST 复位	清除动作保持，寄存器清零	Y,M,S,T,C,D	

- 软元件为 Y 和一般 M 的程序步为 1，S 和特殊辅助继电器 M、定时器 T、计数器 C 的程序步为 2，数据寄存器 D 的程序步为 3。
- SET 指令在线圈接通的时候就对软元件进行置位，只要置位了，除非用 RST 指令复位，否则将保持为 1 的状态。同样，对 RST 指令只要对软元件复位，将保持为 0 的状态，除非用 SET 指令置位。
- 对同一软元件，SET,RST 指令可以多次使用，顺序随意，但是程序最后的指令有效。
- RST 指令可以对数据寄存器(D)，定时器(T)和计数器(C)，辅助继电器 (M)，不论是保持还是非保持的都可以复位置零。

SL1S 编程示例



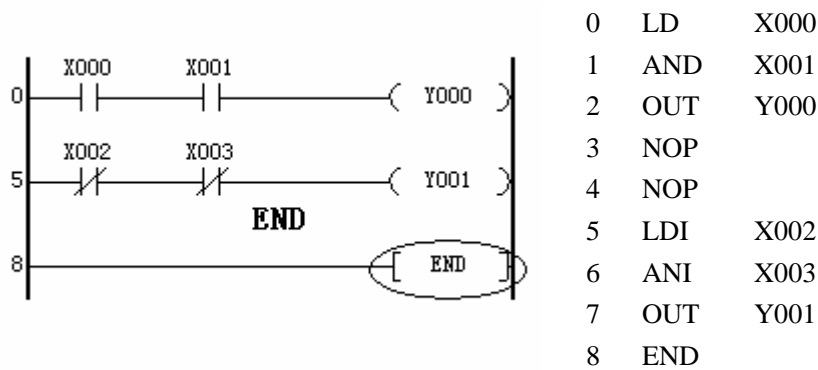
[NOP],[END] 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
NOP 空操作	无动作		1
END 结束	输入输出处理以及返回到开始		1

- 程序清除时指令变为 NOP 指令，指令之间加入 NOP 指令，程序对他不做任何事情，继续向下执行，只是增加了程序的步数。
- 每个程序必须有一个且只有一个 END 指令，表示程序的结束。PLC 不断反复进行如下操作：输入处理，从程序的 0 步开始执行直到 END 指令，程序处理结束，接着进行输出刷新。然后开始循环操作。

SL1S 编程示例



[MPS],[MRD],[MPP] 指令

SL1S 指令解说

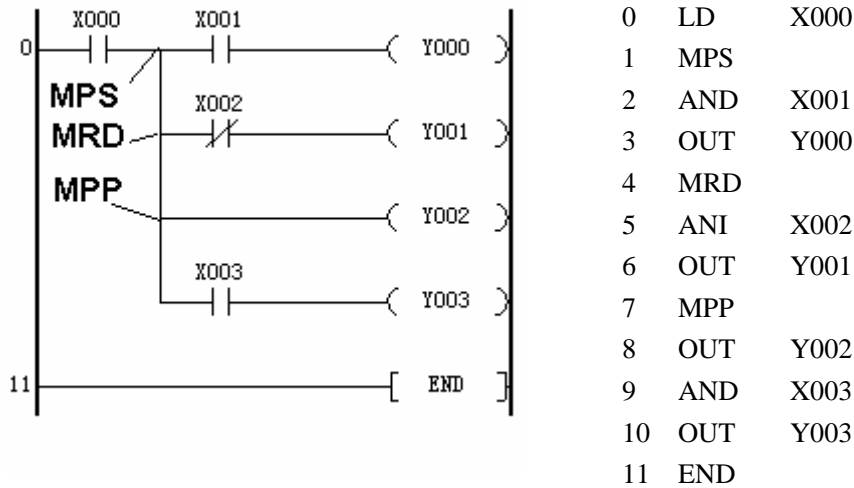
助记符、名称	功能	可用软元件	程序步
MPS 压栈	运算存储		1
MRD 读栈	存储读出		1
MPP 出栈	存储读出与复位		1

- 嵌入式 PLC 中有 11 个栈空间，也就是说可以压栈的最大深度为 11 级。每使用一次 MPS 将当前结果压入第一段存储，以前压入的结果依次移入下一段。MPP 指令将第一段读出，并且删除它，同时以下的单元依次向前移。MRD 指令读出第一段，但并不删除它。其他单元保持不变。使用这三条指令可以方便多分支的编程。
- 在进行多分支编程时，MPS 保存前面的计算结果，以后的分支可以利用 MRD,MPP 从栈中读出前面的计算结果，再进行后面的计算。最后一个分支必须用 MPP，保证 MPS,MPP 使用的次数相同。注意，使用 MPP 以后，就不能再使用 MRD 读出运算结果，也就是 MPP 必须放在最后的分支使用。

- MRD 指令可以使用多次，没有限制。MPS 连续使用的最多次数为 11，但是可以多次使用。每个 MPS 指令都有一个 MPP 指令对应，MPP 的个数不能多于 MPS 的个数。

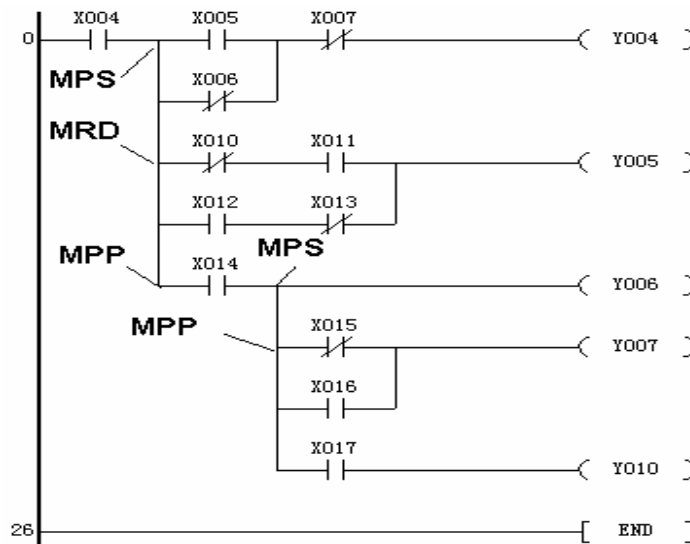
SL1S 编程示例

实例 1:



- 该实例只使用一级堆栈，使用一个 MPS 指令压栈，一个 MRD 指令读栈，一个 MPP 指令出栈。

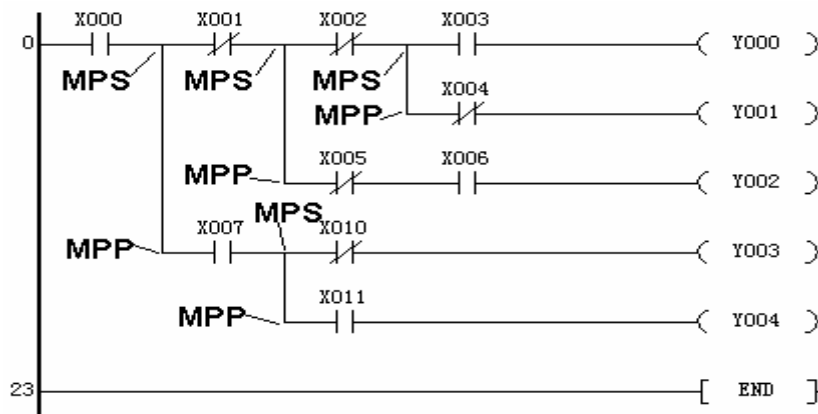
实例 2:



0	LD	X004	14	OUT	Y05
1	MPS		15	MPP	
2	LD	X005	16	AND	X014
3	ORI	X006	17	OUT	Y06
4	ANB		18	MPS	
5	ANI	X007	19	LDI	X015
6	OUT	Y004	20	OR	X016
7	MRD		21	ANB	
8	LDI	X010	22	OUT	Y007
9	AND	X011	23	MPP	
10	LD	X012	24	AND	X017
11	ANI	X013	25	OUT	Y010
12	ORB		26	END	
13	ANB				

- 该实例使用一级两段堆栈，并且跟 OR,ORB,ANB 指令混合使用。

实例 3



0	LD	X000	12	ANI	X005
1	MPS		13	AND	X006
2	ANI	X001	14	OUT	Y002
3	MPS		15	MPP	
4	ANI	X002	16	AND	X007
5	MPS		17	MPS	
6	AND	X003	18	ANI	X010
7	OUT	Y000	19	OUT	Y003
8	MPP		20	MPP	
9	ANI	X004	21	AND	X011
10	OUT	Y001	22	OUT	Y004
11	MPP		23	END	

该实例使用三级堆栈，即堆栈嵌套三级。

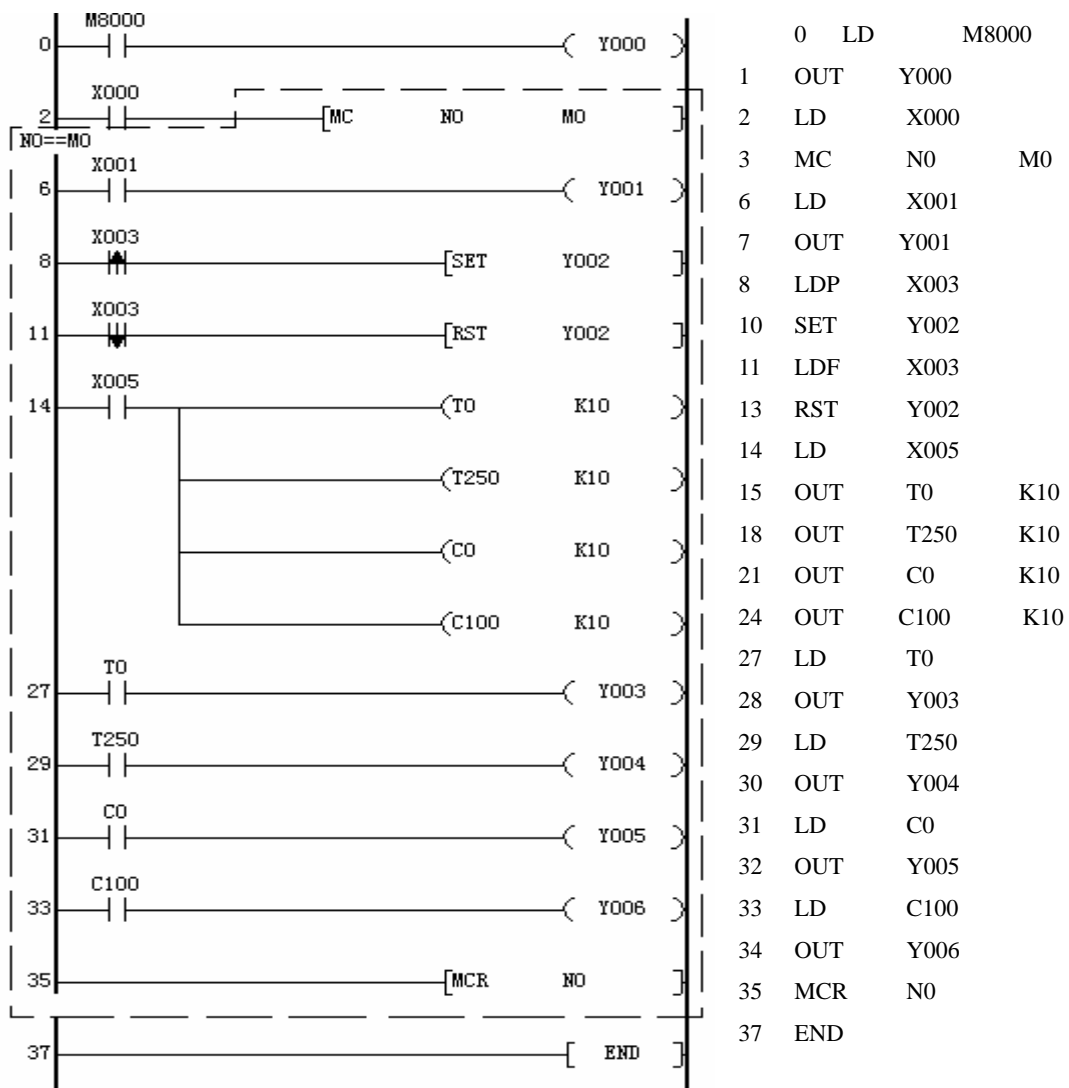
[MC],[MCR] 指令 指令解说

助记符、名称	功能	可用软元件	程序步
MC 主控	公共串联点的连接线圈指令	Y,M (特殊 M 除外)	3
MCR 主控复位	公共串联点的消除指令		2

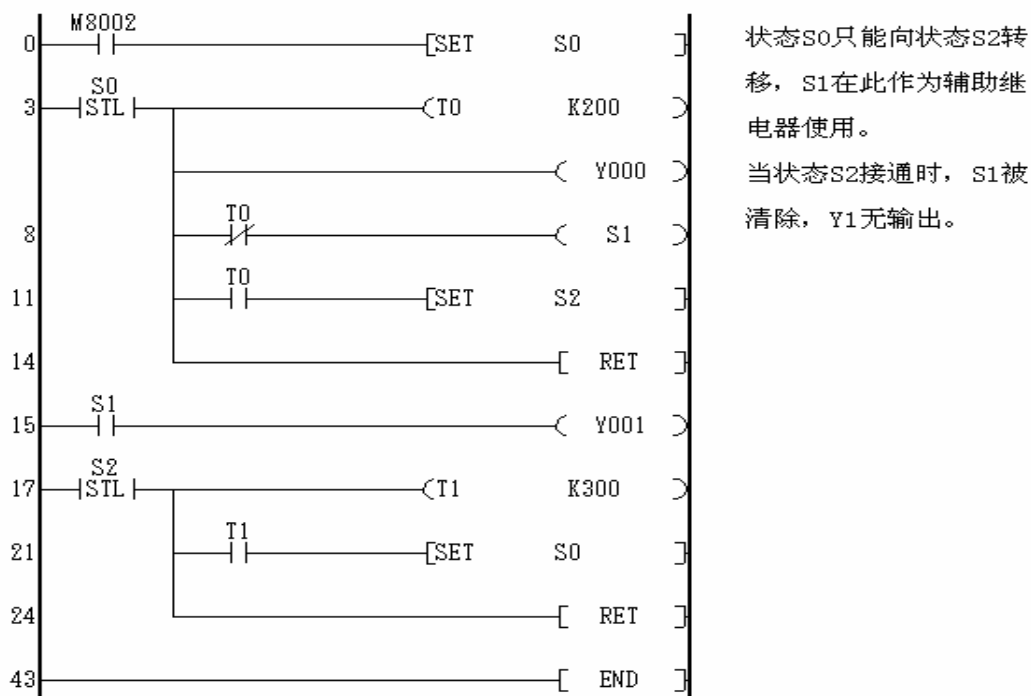
- 当前面的触点接通时，就执行 MC 到 MCR 的指令。执行 MC 指令时，母线向 MC 触点后移动，执行 MCR 指令返回母线。
- 使用 MC 指令时，嵌套级 N 的编号按顺序依次增大，也就是说只有使用 N0，才能嵌套 N1。相反使用 MCR 指令时，必须从大往小返回母线。最大嵌套级数为 7 级 (N6)。
- 通过不同的软元件 Y,M，可以多次使用 MC 指令，如果使用相同的软元件，将同 OUT 指令一样，会出现双线圈输出。

SL1S 编程示例

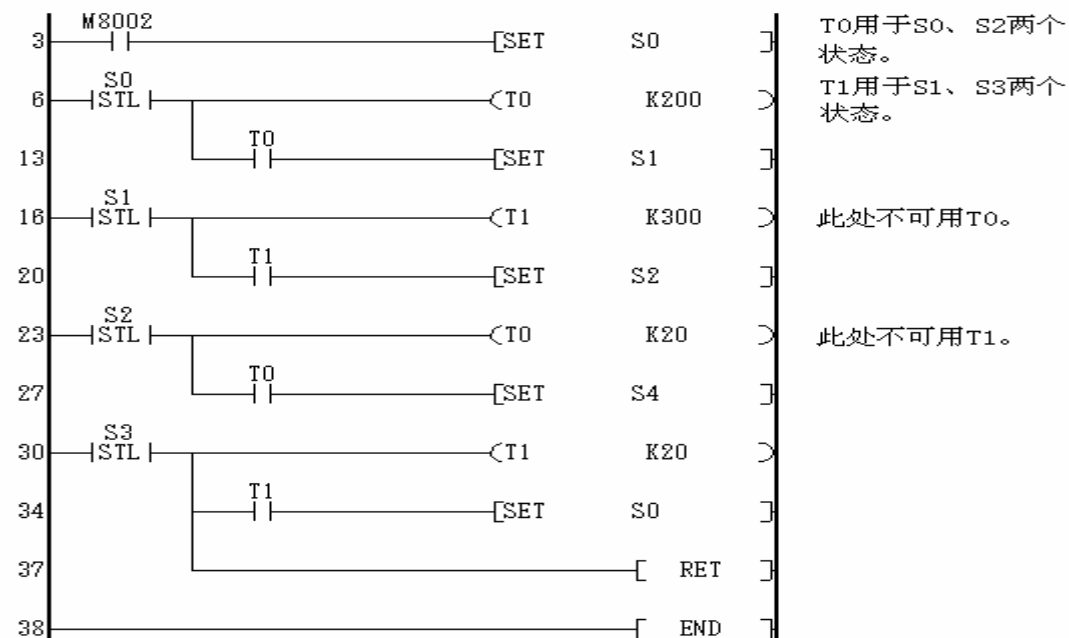
- 该实例只使用一个 MC,MCR 指令，嵌套级数也是 1，可以进行 7 级嵌套。
- 该实例中当 X000 接通时，执行 MC,MCR 之间的指令，当 X000 断开时，成为如下两种形式。
现状保持：累积定时器的值，计数器的值，用 SET/RST 指令驱动的软元件。
变为断开的元件：非累积定时器的值，用 OUT 指令驱动的软元件。



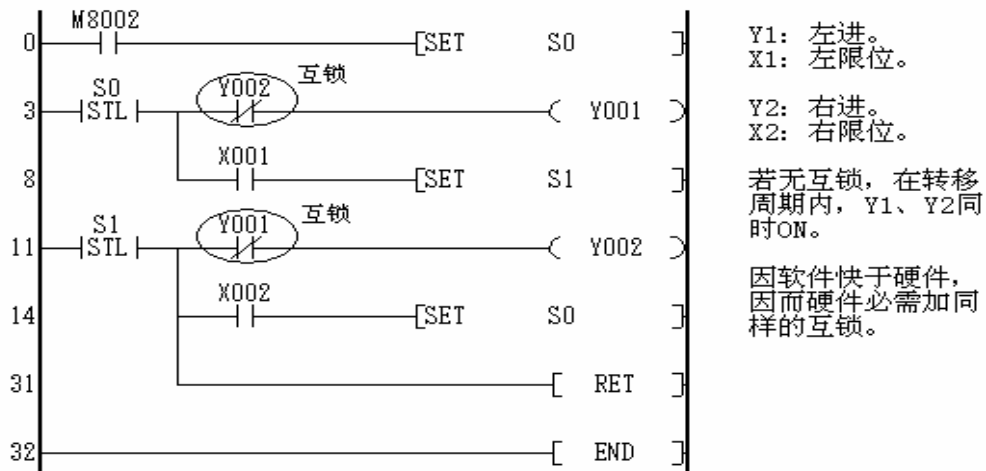
- 状态转移只能用 SET 指令，不能用 OUT 指令。
- 使用 OUT S 时，S 作为辅助继电器使用，而不是状态寄存器。



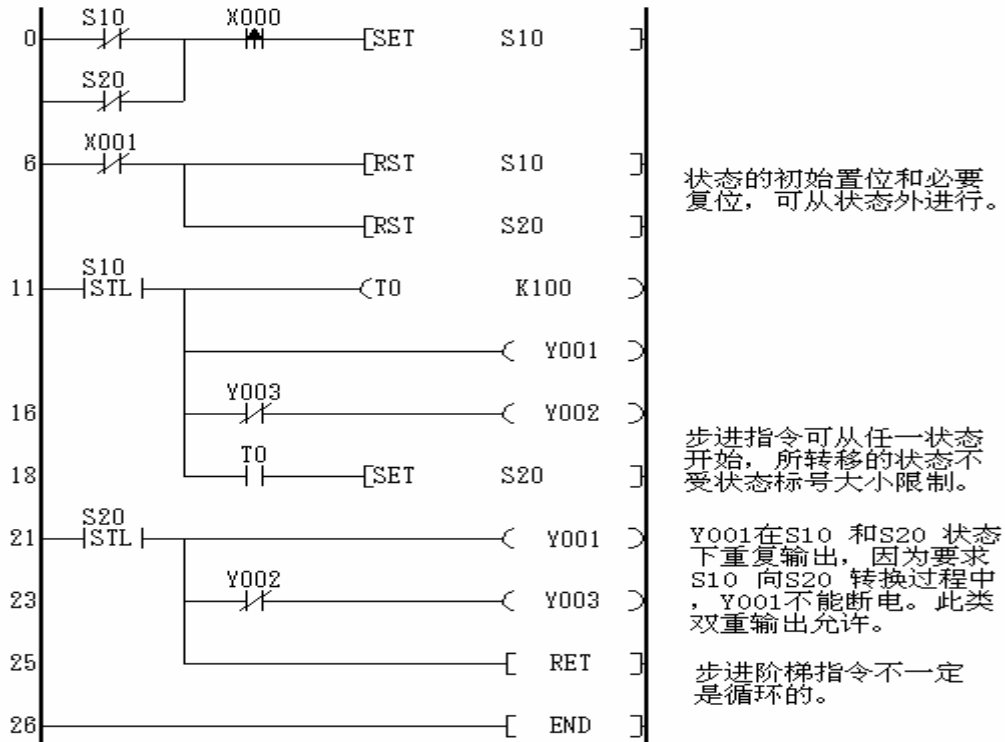
- 时间继电器 T 可重复使用，但相邻两个状态不能重复使用同一时间继电器。



- 两个矛盾继电器输出时，必需加软件互锁。考虑软件快于硬件，相矛盾的硬件输出也必需互锁。



- 允许同一继电器在不同状态下输出，其实际输出视状态转移的位置确定。



:

功能指令一览表

传 送 与 比 较	CMP	比较	
	MOV	传送	
	ZRST	批复位	
四 则 逻 辑 运 算	ADD	BIN 加法	
	SUB	BIN 减法	
	MUL	BIN 乘法	
	DIV	BIN 除法	
	INC	BIN 加 1	
	DEC	BIN 减 1	

接 点 比 较	LD=	(S1) = (S2)	
	LD>	(S1) > (S2)	
	LD<	(S1) < (S2)	
	LD<>	(S1) <> (S2)	
	LD≡	(S1) ≡ (S2)	
	LD≧	(S1) ≧ (S2)	
	AND=	(S1) = (S2)	
	AND>	(S1) > (S2)	
	AND<	(S1) < (S2)	
	AND<>	(S1) <> (S2)	
	AND≡	(S1) ≡ (S2)	
	AND≧	(S1) ≧ (S2)	
	OR=	(S1) = (S2)	
	OR>	(S1) > (S2)	
	OR<	(S1) < (S2)	
	OR<>	(S1) <> (S2)	
	OR≡	(S1) ≡ (S2)	
OR≧	(S1) ≧ (S2)		

SL1S 比较指令 [CMP]

	CMP	
D		P

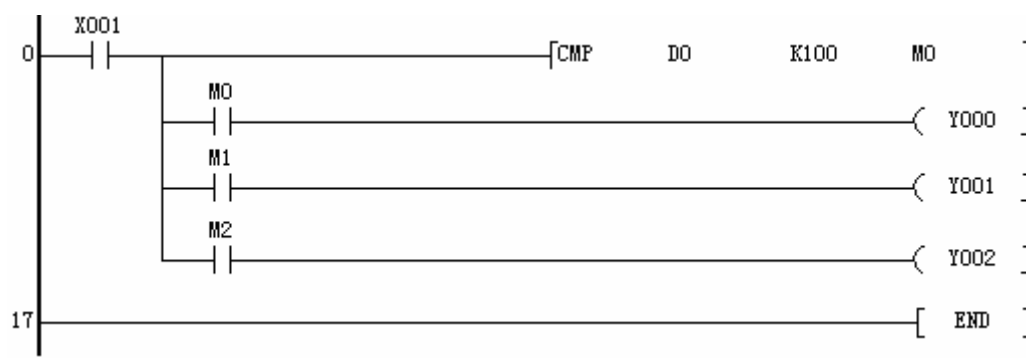
16 位指令 CMP (连续执行型)
7 步 CMPP (脉冲执行型)

32 位指令 DCMP (连续执行型)
13 步 DCMP (脉冲执行型)

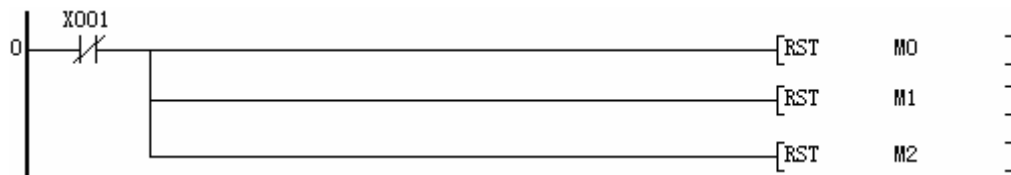
适用
软
元
件

- 字软元件 (S1 • 、S2 •) K、H、T、C、D、 Y、M、S
- 位软元件 (D •)

功能和动作



- 上图示例是 D0 的内容与常数 100 进行比较，大小比较是按代数形式进行的 ($-8 < 0$)；
 - 所有数据都以 2 进制值处理；
 - 当 $D0 > 100$ ，M0 “ON”，当 $D0 = 100$ ，M1 “ON”，当 $D0 < 100$ ，M2 “ON”；
 - 目标地址指定 M0，则 M1、M2 被自动占用；
 - 当 X001 “OFF”时，M0、M1、M2 仍保持以前状态。如当 D0 的内容为 50，则 $50 < 100$ ，M2 “ON”，M0、M1 都 “OFF”，X001 “OFF”时，M2 仍 “ON”。
- 指令不执行时，想要清除比较结果，可使用复位指令。



SL1S 传送指令 [MOV]

	MOV	
D		P

16 位指令 MOV (连续执行型)
7 步 MOV (脉冲执行型)

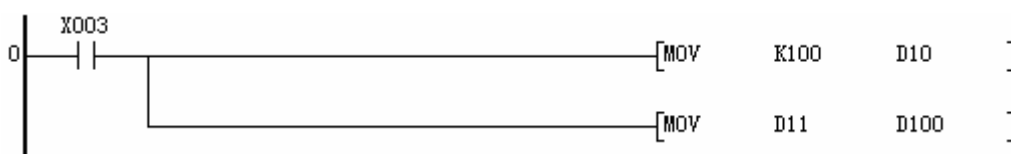
32 位指令 DMOV (连续执行型)
13 步 DMOV (脉冲执行型)

适用软元件

- 字软元件 (S) T、C、D、
- 字软元件 (D) T、C、D、

功能和动作

使数据原样传送的指令。



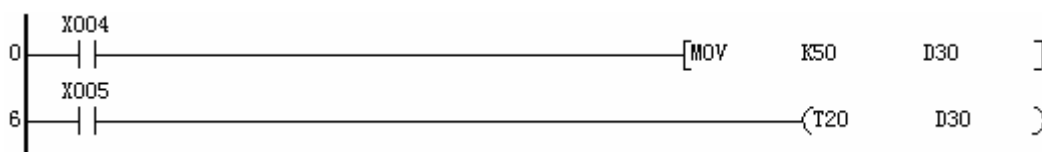
- 将源 (D11) 的内容向目标 (D100) 传送, X003 “OFF” 时, 目标 (D100) 的内容不变化;
- 常数 K100 被自动转换成 BIN 码。

《定时、计数器的当前值读出示例》



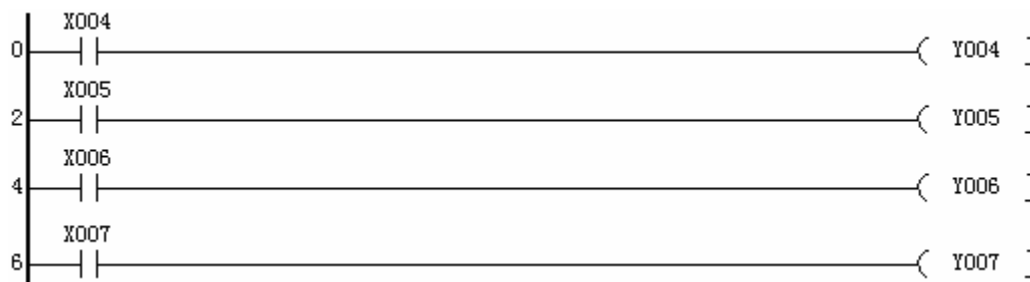
将 T0 当前值传送给 D20。

《定时、计数器设定值的间接指定示例》

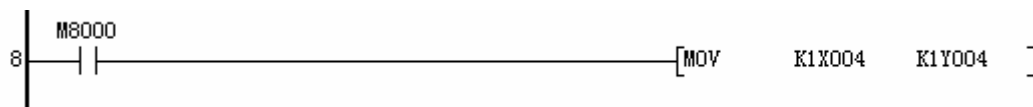


- T20 定时时间为 5 秒。

《位软元件的传送》

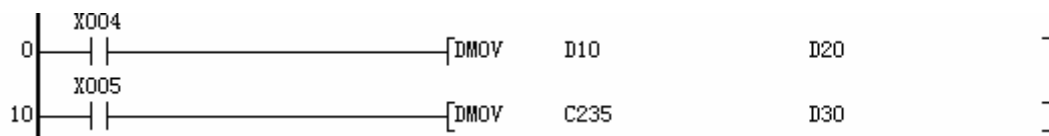


- 上图示例可用下面的 MOV 指令来实现,



《32 位数据的传送》

运算结果是 32 位的应用指令（**MUL** 等）、32 位数值、32 位软元件或 32 位计数器等 32 位数据的传送，必须使用 **DMOV** 指令。



- 上例将（D11、D10）的内容传送给（D21、D20），（C235 的当前值）传送给（D31、D30）。

则逻辑运算

SL1S BIN 加法运算 [ADD]

	ADD	
D		P

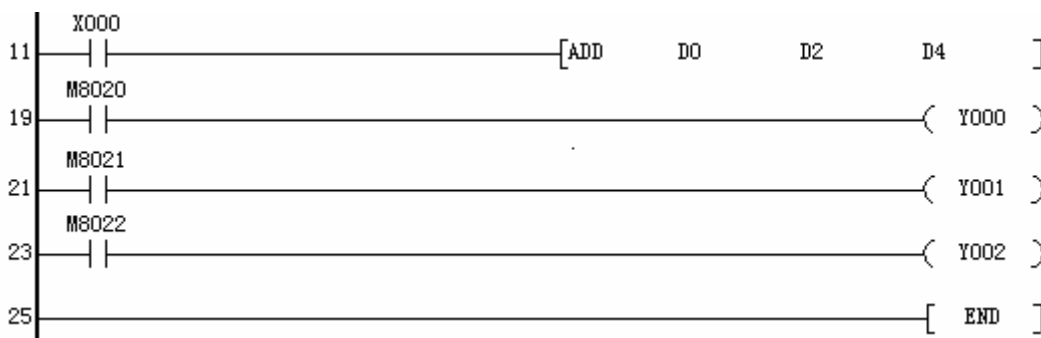
16 位指令 ADD (连续执行型)
7 步 ADDP (脉冲执行型)

32 位指令 DADD (连续执行型)
13 步 DADDP (脉冲执行型)

适用软元件

- 字软元件 (S1 • 、S2 •) K、H、T、C、D
- 字软元件 (D •) T、C、D

功能和动作



- 两个源数据进行加法后传送到目标处，各数据的最高位是符号位（正数为 0，负数为 1），数据以代数形式进行加法运算（ $8 + (-8) = 0$ ）。
- 运算结果为 0 时，0 标志位 M8020 动作；运算结果超出 32767（16 位运算）或 2147483647（32 位运算）时，进位标志位 M8022 动作；运算结果小于 -32768（16 位运算）或 -2147483648（32 位运算）时，借位标志位 M8021 动作；
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 对于脉冲型指令，每出现一次 OFF 到 ON 的变化，操作数做一次运算。
- 可以将源 (S •) 和目标 (D •) 指定为相同的软元件编号。这种情况下，如使用连续执行型指令 (ADD、DADD)，则每个扫描周期加一次，请务必注意。

SL1S BIN 减法运算 [SUB]

	SUB	
D		P

16 位指令 SUB (连续执行型)
7 步 SUBP (脉冲执行型)

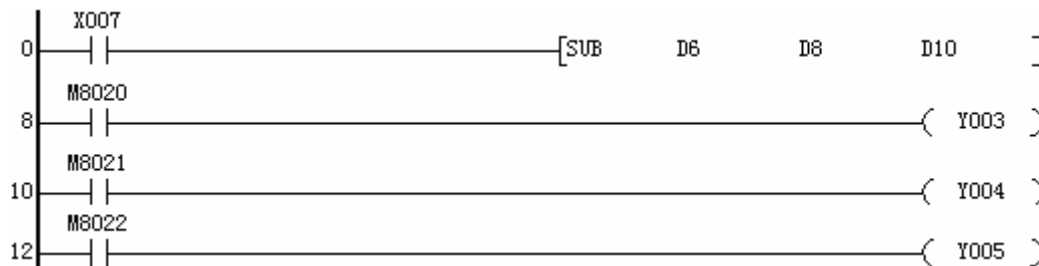
32 位指令 ISUB (连续执行型)
13 步 DSUBP (脉冲执行型)

适用

- 字软元件 (S1 • 、S2 •) K、H、T、C、D、
- 字软元件 (D •) T、C、D、

软 元 件	
-------------	--

功能和动作



- (S1 •) 指定的内容和 (S2 •) 指定的内容相减，结果存入 (D •) 指定的软元件中。(8-(-8)=16)。
- 各种标志位的动作，32 位运算软元件的指定方法，连续型和脉冲型的差异等都跟 ADD 指令相同。

SL1S BIN 乘法运算 [MUL]

	MUL	
D		P

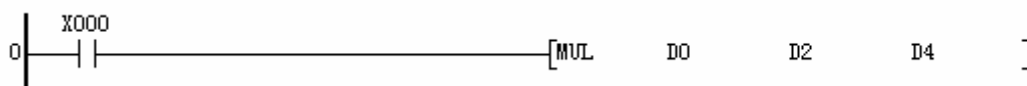
16 位指令 MUL (连续执行型)
7 步 MULP (脉冲执行型)

32 位指令 DMUL (连续执行型)
13 步 DMULP (脉冲执行型)

适 用 软 元 件	<ul style="list-style-type: none"> • 字软元件 (S1 •、S2 •) K、H、T、C、D (仅限 16 位计算) • 字软元件 (D •) T、C、D
-----------------------	--

功能和动作

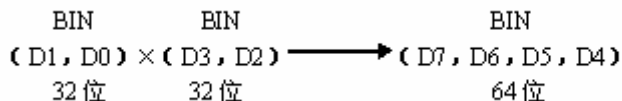
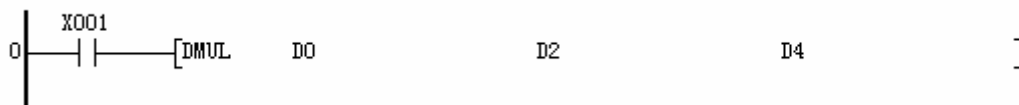
《16 位运算》



$$\begin{array}{ccc}
 \text{BIN} & \text{BIN} & \text{BIN} \\
 (D0) \times (D2) & \longrightarrow & (D5, D4) \\
 16 \text{ 位} & & 32 \text{ 位}
 \end{array}$$

- 各源指定的软元件内容的乘积，以 32 位数据形式存入目标地址指定的软元件 (低位) 和紧接其后的软元件 (高位) 中，如 (D0)=125, (D2)=8, 则 (D5, D4)=1000;
- 结果的最高位是符号位，0 为正，1 为负;

《32 位运算》



- 在 32 位运算中，目标地址使用位软元件，只能得到低 32 位的结果，最好先向字元件传送一次后再进行运算；
- 即使使用字元件，也不能一下子监视 64 位数据的运算结果，此种情况下建议进行浮点数运算；

SL1S BIN 除法运算 [DIV]

D	DIV	P
---	-----	---

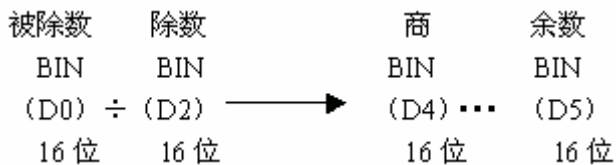
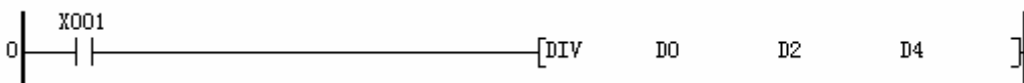
16 位指令 DIV (连续执行型)
7 步 DIVP (脉冲执行型)

32 位指令 DDIV (连续执行型)
13 步 DDIVP (脉冲执行型)

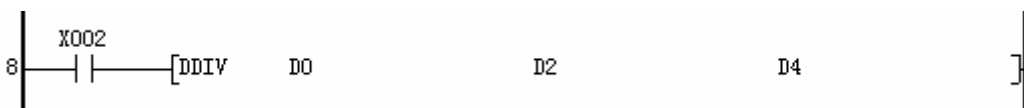
适用软元件	<ul style="list-style-type: none"> • 字软元件 (S1 • 、S2 •) K、H、T、C、D、(仅限 16 位计算) • 字软元件 (D •) T、C、D、
-------	--

功能和动作

《16 位运算》



《32 位运算》



被除数	除数	商	余数
BIN	BIN	BIN	BIN
(D1, D0) ÷	(D3, D2)	→ (D5, D4) ...	(D7, D6)
32 位	32 位	32 位	32 位

- 32 位运算不能指定 Z 作为 (D)；
- 除数为 0 时，如果被除数为正数，商为 32767 (16 位) 或 2147483647 (32 位)；如果被除数为 0，商为 0；如果被除数为负数，商为 -32768 (16 位) 或 -2147483648 (32 位)；
- 商和余数的最高位为符号位，0 为正，1 为负，当被除数或除数中的一方为负数时，商为负，当被除数为负时，余数则为负。

SL1S BIN 增 1 [INC]

	INC	
D		P

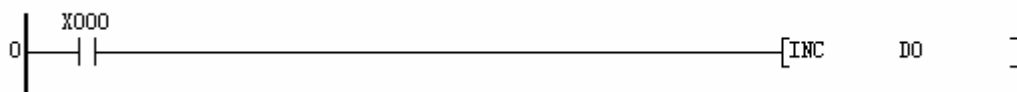
16 位指令 INC (连续执行型)
3 步 INCP (脉冲执行型)

32 位指令 DINC (连续执行型)
5 步 DINCP (脉冲执行型)

适用软元件

• 字软元件 (D) T、C、D

功能和动作



(D0) + 1 → (D0)

- X000 每置“ON”一次，D0 的内容增 1，在连续执行指令中，每个扫描周期执行加 1 运算，所以务必引起注意；
- 16 位运算时，如果 32767 加 1 变为 -32768，标志位不动作；32 位运算时，如果 2147483647 加 1 变为 -2147483648，标志位不动作；

SL1S BIN 减 1 [DEC]

	DEC	
D		P

16 位指令 DEC (连续执行型)
3 步 DECP (脉冲执行型)

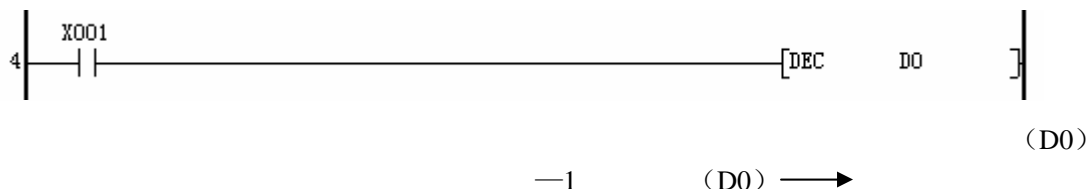
32 位指令 DDEC (连续执行型)
5 步 DDECP (脉冲执行型)

适

• 字软元件 (D) T、C、D

用 软 元 件	
------------------	--

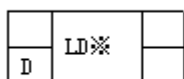
功能和动作



- X001 每置“ON”一次，D0 的内容减 1，在连续执行指令中，每个扫描周期执行减 1 运算，所以务必引起注意；
- -32768 或-2147483648 减 1 变为 32767 或 2147483647，标志位不动作

SL1S 触点比较指令

SL1S 接点比较指令 [LD※]



16 位指令 LD※（连续执行型）
5 步

32 位指令 LDD※（连续执行型）
9 步

※ 表示：=、>、<、◇、≡、≥。

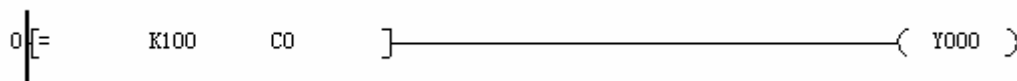
适 用 软 元 件	<ul style="list-style-type: none"> • 字软元件 (S1 • 、S2 •) K、H、T、C、D
-----------------------	---

指令形式
与功能

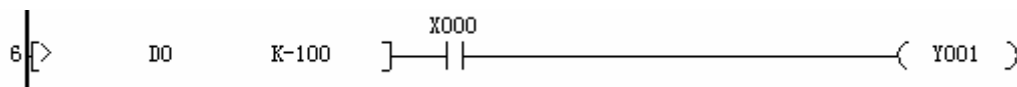
对源数据进行 BIN 比较，对应其结果执行后段的运算

16 指令	32 位指令	导通条件	非导通条件
LD=	LDD=	(S1 •) = (S2 •)	(S1 •) ≠ (S2 •)
LD>	LDD>	(S1 •) > (S2 •)	(S1 •) ≤ (S2 •)

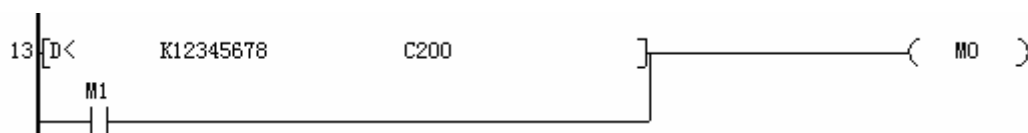
LD<	LDD<	(S1 •) < (S2 •)	(S1 •) ≧ (S2 •)
LD<>	LDD<>	(S1 •) ≠ (S2 •)	(S1 •) = (S2 •)
LD≧	LDD≧	(S1 •) ≧ (S2 •)	(S1 •) > (S2 •)
LD≦	LDD≦	(S1 •) ≦ (S2 •)	(S1 •) < (S2 •)



- 当计数器 C0 的当前值为 100 时，驱动。



- 当 D0 的内容大于-100，且 X000 处于“ON”时，驱动 Y1。



- 当计数器 C200 的内容大于 12345678，或者 M1 处于“ON”时，驱动 M0。

注意事项

- 当源数据的最高位（16 位指令：b15，32 位指令：b31）为 1 时，将该数值作为负数进行比较；

SL1S 接点比较指令 [AND※]

D	AND※	
---	------	--

16 位指令 AND※（连续执行型）
5 步

32 位指令 ANDD※（连续执行型）
9 步

※表示：=、>、<、<>、≧、≦。

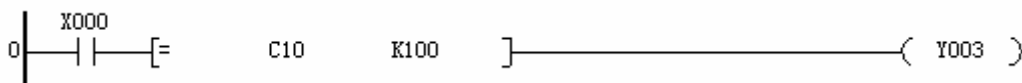
适用软元件

- 字软元件 (S1 •、S2 •) K、H、T、C、D

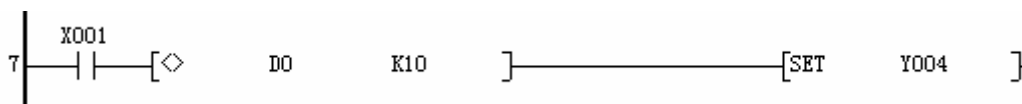
指令形式与功能

对源数据进行 BIN 比较，对应其结果执行后段的运算。

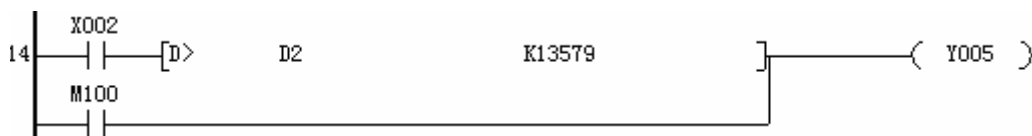
16 指令	32 位指令	导通条件	非导通条件
AND=	ANDD=	$(S1 \bullet) = (S2 \bullet)$	$(S1 \bullet) \neq (S2 \bullet)$
AND>	ANDD>	$(S1 \bullet) > (S2 \bullet)$	$(S1 \bullet) \leq (S2 \bullet)$
AND<	ANDD<	$(S1 \bullet) < (S2 \bullet)$	$(S1 \bullet) \geq (S2 \bullet)$
AND<>	ANDD<>	$(S1 \bullet) \neq (S2 \bullet)$	$(S1 \bullet) = (S2 \bullet)$
AND \leq	ANDD \leq	$(S1 \bullet) \leq (S2 \bullet)$	$(S1 \bullet) > (S2 \bullet)$
AND \geq	ANDD \geq	$(S1 \bullet) \geq (S2 \bullet)$	$(S1 \bullet) < (S2 \bullet)$



- 当 X000 处于“ON”时，且计数器 C10 的当前值等于 100 时，驱动 Y3。



- 当 X001 处于“ON”时，且 D0 的内容不等于 10 时，置位 Y4。

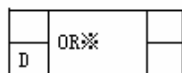


- 当 X002 处于“ON”，且 (D3, D2) 的内容大于 13579 时，或者 M100 处于“ON”时，驱动 Y5。

注意事项

- 当源数据的最高位（16 位指令：b15，32 位指令：b31）为 1 时，将该数值作为负数进行比较；

SL1S 接点比较指令 [OR※]



16 位指令 OR※（连续执行型）
5 步

32 位指令 ORD※（连续执行型）
9 步

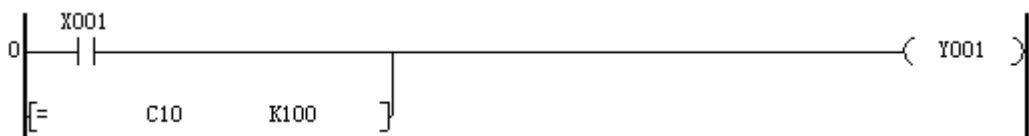
※表示：=、>、<、<>、 \leq 、 \geq 。

适用 软元 件	· 字软元件 (S1 • 、S2 •) K、H、T、C、D
---------------	-------------------------------

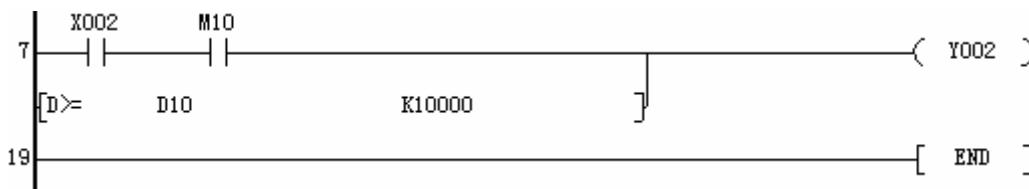
指令形式
与功能

对源数据进行 BIN 比较，对应其结果执行后段的运算。

16 指令	32 位指令	导通条件	非导通条件
OR=	ORD=	$(S1 \bullet) = (S2 \bullet)$	$(S1 \bullet) \neq (S2 \bullet)$
OR>	ORD>	$(S1 \bullet) > (S2 \bullet)$	$(S1 \bullet) \leq (S2 \bullet)$
OR<	ORD<	$(S1 \bullet) < (S2 \bullet)$	$(S1 \bullet) \geq (S2 \bullet)$
OR<>	ORD<>	$(S1 \bullet) \neq (S2 \bullet)$	$(S1 \bullet) = (S2 \bullet)$
OR \leq	ORD \leq	$(S1 \bullet) \leq (S2 \bullet)$	$(S1 \bullet) > (S2 \bullet)$
OR \geq	ORD \geq	$(S1 \bullet) \geq (S2 \bullet)$	$(S1 \bullet) < (S2 \bullet)$



- 当 X001 处于“ON”，或计数器 C10 的当前值等于 100 时，驱动 Y1



- 当 X002 和 M10 处于“ON”时，或者 (D11, D10) 的内容大于等于 10000 时，驱动 Y2。

注意事项

- 当源数据的最高位（16 位指令：b15，32 位指令：b31）为 1 时，将该数值作为负数进行比较；

功能指令的基本规则

本节叙述可编程控制器功能指令的表示方法与基本规则。在使用功能指令编程时，需要大致了解指令中有关软元件的使用及其执行形式。

SL1S 功能指令的表示与执行形式

1、指令与操作数

- ①、功能指令用助记符表示。
- ②、有些功能指令仅有指令段（助记符），但更多的有操作数。
- ③、指令中的操作数符号表示方法及解释。

\boxed{s} ：表示数据源。内容不随指令执行而变化的操作数称为源。

在可变址修改软元件编号的情况下，加上“•”符号的 $\boxed{s \bullet}$ 表示。

$\boxed{S_1 \bullet}$ $\boxed{S_2 \bullet}$

源的数量多时，以 \square 、 \square 等表示。

\square D：表示目标操作数。内容随指令执行而改变的操作数被称作目标。

可作变址修饰时，加上“·”符号的 \square D·表示。

在目标数量多时，以 \square D₁·、 \square D₂·等表示。

\square n· \square m·：以 \square m· 或 \square n· 表示既不做源，也不做目标的操作数。

这样的操作数数量很多时，以 \square m1·、 \square m2·、 \square n1·、 \square n2· 等表示。

2、可用作操作数的软元件

1、X, Y, M, S 等位元件。

2 数据寄存器 D、定时器 T 的当前值寄存器、计数器 C 的当前值寄存器。

- 数据寄存器 D 为 16 位，在处理 32 位数据时使用一对数据寄存器的组合。

例如，将数据寄存器 D0 指定为 32 位指令的操作数时，处理(D1, D0)32 位数据(D1 为高 16 位，D0 为低 16 位)。

- T、C 的当前值寄存器也可作为一般寄存器处理

3、指令的形态与执行形式

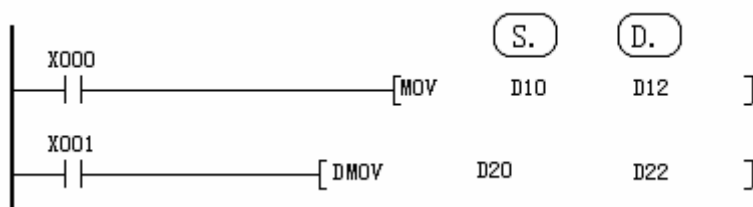
根据功能指令处理数值的大小，分为《16 位指令》和《32 位指令》。

根据功能指令的执行形式，分为《连续执行型》与《脉冲执行型》。

功能指令可将这些形式组合使用或单独使用。

①、16 位指令和 32 位指令

- 在数值处理的功能指令中，根据数值数据的位长分为 16 位与 32 位。

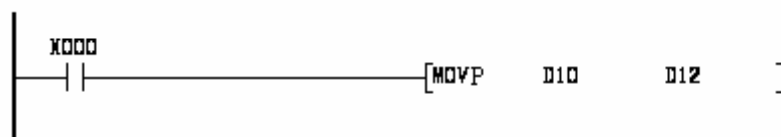


16 位指令：通过 MOV 将 D10 的内容传送到 D12 中的指令。

32 位指令：通过 DMOV 将(D21, D20)的内容传送到(D23, D22)中。

②、脉冲执行形式和连续执行形式

- 脉冲执行型



如图所示，在 X000 从 OFF→ON 变化时，指令执行一次。

指令在不执行时的处理时间快，建议尽量采用脉冲执行型指令。
符号 **P** 表示脉冲执行型命令。

- 连续执行型



图为连续执行型指令，X001 接通时，每个扫描周期都执行。

- INC、DEC 等指令根据程序要求的内容而采取不同的执行型式。
如果采用连续执行型指令，则每个扫描周期，其操作数的内容都发生变化。这种指令采用连续形式指令时，必须注意。
- 在功能指令解说时，使用下图符号以示区别。



←使用连续执行命令时，每一扫描周期“源”的内容都发生变化。

功能指令内的数值处理

1、位元件的处理

- ①、位元件：象 X, Y, M, S 等只处理 ON/OFF 信息的软元件

字元件：象 T, C, D 等处理数值的软元件。

- 若向 K1M0-K3M0 传送 16 位数据，则数据长度不足的高位部分不被传送。32 位数据亦同样。
- 在 16 位(或 32 位)运算中，对应位元件的位指定是 K1-K3(或 K1-K7)时，长度不足的高位通常被视为 0。因此，通常将其作为正数处理。
- 被指定的位元件编号，一般可自由指定。建议：
在 X, Y 的场合，最低位的编号尽可能设定为 0，如 X000, X010, X020...Y000, Y010, Y020...
在 M, S 的场合，最低位的编号设定为 8 的倍数，或设定为 0，如 M0, M10 等。

- ②、连续字的指定

- 所谓以 D1 为开头的一系列数据寄存器就是 D1, D2, D3, D4...等。

- 通过位指定，在字的场合，也可将其作为一系列的字处理。如下所示。

K1X000 K1X004 K1X010 K1X014...

K2Y010 K2Y020 Y2X030...

K3M0 K3M12 K3M24 K3M36...

K4S16 K4S32 K4S48...

也就是说，按照各位的进制且不跳过硬元件，连续使用软元件。

- 在 32 位运算中采用 K4Y000 时，则将高 16 位看作 0。在需要 32 位数据时，要用 K8Y000。

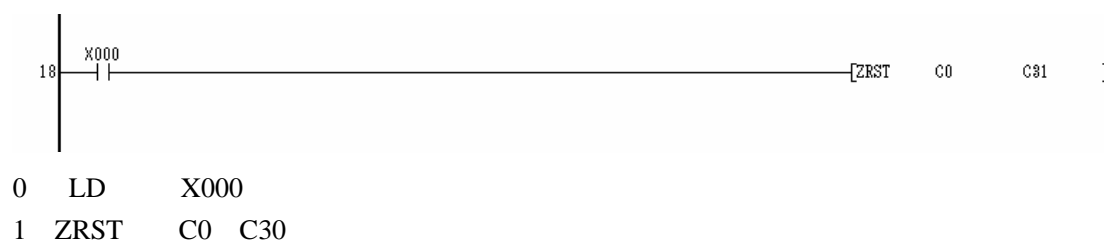
ZRST 指令

SL1S 指令解说

助记符、名称	功能	可用软元件	程序步
ZRST	批复位，寄存器清零	Y,M,S,C	4

- ZRST 指令一次只能使用同一个软元件，并且顺序是由小到大。

SL1S 编程示例



- 实例中 X001 NO 时，C0 至 C31 中的数值全部清零。