

HNC-8 数控系统软件

PLC 编程说明书



V1.24

2016-02

武汉华中数控股份有限公司

目录

目录	I
前言	1
PLC 概述	2
PLC 规格	3
顺序程序概念	4
分配接口	5
顺序程序	6
顺序程序构成	9
地址	10
基本指令	11
LD.....	14
LDI.....	16
OUT	18
OOUT.....	19
SET	21
RST.....	22
AND	23
ANI.....	24
OR.....	25
ORI.....	26
LDP	27
LDF.....	29
ANDP	30
ANDF	31
ORP.....	32
ORF	33
ORB.....	34
ANB.....	36
MPS、MRD、MPP.....	38
基本元件	40
常开触点	41
常闭触点	42
常真触点	43
触点上升沿	44
触点下降沿	45
逻辑输出	46
逻辑反输出	47
置位输出	48
复位输出	49

基本功能模块	50
控制指令	51
M 指令获取 MGET	51
M 指令应答 MACK	52
T 指令获取 TGET	53
T 指令应答 TACK	54
手摇控制 RTOMPG	55
热误差补偿模块 TEMPSEN	56
数学运算	58
加法 ADD	58
减法 SUB	60
乘法 MUL	62
除法 DIV	64
加一 INC	66
减一 DEC	67
逻辑与 WAND	68
逻辑或 WOR	70
逻辑异或 WXOR	72
求补 NEG	74
计数器	75
加减计数器 CTR	75
计数器 CTRC	77
自定起始加减计数器 CTUD	78
定时器	80
延时导通定时器 TMRB	80
延时断开定时器 STMR	82
流程控制	84
初始化模块结束 IEND	84
PLC1 模块结束 1END	85
PLC2 模块结束 2END	86
跳转 JMP	87
标号 LBL	88
调用子程序 CALL	89
子程序开始 SP	90
子程序结束 SPE	91
子程序返回 RETN	92
循环 LOOP	93
下一次循环 NEXT	94
比较	95
比较 CMP	95
小于 LT	96
区域比较 ACOMP	97
一致性比较 COIN	99
数据操作	100

移动数据 MOV	100
相对移动数据 XMOV	101
成批移动 BMOV	103
多个移动 FMOV	105
数据交换 XCH	107
数据复位 ZRST	110
编码 ENCO	111
译码 DECO	113
代码转换 COD	115
数据查找 SER	117
寄存器合并 ASSEM	119
寄存器分解 DISAS	121
区域转换 ACVT	123
交替输出 ALT	125
取上升沿 PLS	126
取下降沿 PLF	127
点数转换 PTN	128
数点转换 NTP	130
计件 PARTCNT	132
计件清除 PARTCLR	133
温度采集模块 HEADSEN	134
状态字和控制字编程	136
状态字和控制字介绍	137
轴状态字	138
轴控制字	141
通道状态字	145
通道控制字	149
状态字和控制字编程实例	153
工作模式设置	153
工作模式获取	154
进给轴与主轴控制	155
回零	156
增量倍率修调	157
循环启动与进给保持	158
扩展功能模块	159
NC 功能	160
通道模式设置 MDST	160
通道模式获取 MDGT	161
模式 MDI	162
锁住通道 MST	163
循环启动开启 CYCLE	164
急停 STOP	165
复位 RESET	166

通道切换 CHANSW	167
进给保持开启 HOLD	168
循环启动指示灯 CYCLED.....	169
进给保持指示灯 HOLDLED	170
程序跳段 (G31) ESCBLK	171
快移修调设置 RPOVRD.....	172
进给修调设置 FEEDOVRD	173
主轴修调设置 SPDLOVRD	174
增量 (步进) 倍率设置 STEPMUL	175
空运行开关 DRYRUN.....	176
跳段开关 SKIP.....	177
用户输入 USERIN.....	178
用户输出 USEROUT.....	179
选择停开关 SELSTOP	180
矢量刀具方向设置 TOOLSET	181
矢量刀具方向清除 TOOLCLR	182
8 位数码管 NIXIE	183
刀具显示 TOOLUSE.....	184
轴功能单元	185
主轴手动设置 SPDLJOG	185
主轴控制【伺服主轴】SPDLBUS	186
带档位的主轴控制【伺服主轴】SPDLBUS1	187
主轴定向使能 SPDLORI.....	190
主轴定向完成 SPDLOROK	191
主轴控制【DA】SPDA	192
主轴零速检测 SPDLZERO.....	194
主轴速度到达 SPDLRCH.....	195
从轴回零允许 SUBAXEN	196
释放从轴 DESYN.....	197
轴点动控制 JOGSW.....	198
轴步进控制 STEPAXIS.....	199
轴点动速度选择 JOGVEL	201
轴回零启动 HOMRUN.....	202
轴回零启动 HOMERUN1.....	203
回零接近开关 HOMESW.....	204
轴回零完成 HOMLED	205
轴使能 AXEN.....	206
轴就绪指示【总线】AXRDY.....	207
轴锁住 AXISLOCK	208
PMC 轴相对移动 AXISMOVE.....	209
PMC 轴绝对移动 AXISMVTO.....	210
轴第 2 软极限 AXLMF2	211
正限位挡块开关 AXISPLMT	212
负限位挡块开关 AXISNLMT.....	213

手摇设置 MPGSET.....	214
伺服使能【总线】SVSW	215
轴工作模式 AXISMODE.....	216
轴参考点确认 REFPT.....	217
轴回零过程中 AXISHOM2.....	218
轴移动过程中 AXMOVING.....	219
系统功能	220
旋转控制 ROT.....	220
报警设置 ALARM.....	222
事件设置 EVENT.....	223
保存数据 SAVEDATA	224
复位设置输出 RSTCHK	225
复位清除 RSTCLR.....	226
梯形图运行监控与在线修改	227
梯图在线诊断	228
查找	230
修改	232
插入直线	233
插入竖线	233
删除竖线	234
删除元件	234
常开	235
常闭	236
逻辑输出	236
取反输出	237
功能模块	237
返回	238
命令	239
选择	240
删除	240
移动	241
复制	243
粘贴	244
插入行	244
增加行	245
返回	246
载入	247
放弃	248
保存	249
返回	250
PLC 开发环境使用说明	251
简介	252
开发环境的安装	253

开发环境界面	255
菜单	255
梯形图界面	257
语句表界面	259
符号表界面	260
开发环境操作	261
符号表操作	261
梯形图操作	264
语句表操作	269
附录 A	271

前言

读者对象 该资料面向机床用户。本系列说明书全面地介绍 HNC-8 数控系统的 PLC 指令及其使用、编程方法和示例等。

适用范围 该编程说明适用于控制系统：
HNC-8 数控系统 V1.24

注意事项 本说明书的更新和升级事宜，由武汉华中数控股份有限公司授权并组织实施。

未经本公司授权或书面许可，任何单位或个人无权对本说明书内容进行修改或更正，本公司概不负责由此而造成的客户损失。

本说明书中没有特别描述的事件均可视为“不可能”或“不允许”的事件。

本说明书版权归武汉华中数控股份有限公司，任何单位与个人进行出版或复印均属于非法行为，我公司将追究其法律责任。

技术支持 市场部：027-87180095，027-87180303

传 真：027-87180303

E-mail: market@huazhongcnc.com

邮 编：430223

地 址：武汉东湖高新庙山小区华中科技大学科技园

PLC 概述

本章主要内容:

1.1 PLC 规格

1.2 顺序程序概念

1.3 分配接口

1.4 顺序程序

1.5 顺序程序构成

1.6 地址

PLC 规格

规格 不同规格的 PLC，其程序容量，功能指令数，寄存器使用范围都有所不同。

规格	HNC8
编程语言	Ladder, STL
第一级程序执行周期	1ms
程序容量	
梯形图	5000 行
语句表	10000 行
符号名称	1000 条
指令 基本指令,功能指令	
单字节内部继电器 (R)	400 字节 (R0~~R399)
双字节内部寄存器 (W)	400 字节 (W0~~W199)
四字节内部寄存器 (D)	400 字节 (D0~~D99)
定时器 (T)	128 (T0~~T127)
计数器 (C)	128 (C0~~C127)
子程序 (S)	---
标号 (L)	---
用户自定义参数 (P)	200 (P0~~P199)
保持型存储区	
定时器 (T)	128 (T300~~T427)
计数器 (C)	128 (C300~~C427)
四字节寄存器 (B)	200 字节 (B0~~B49)
I/O 模块 (X)	X0~~X512
(Y)	Y0~~Y512

顺序程序概念

概念

在讲述编程操作前，对顺序程序的功能进行简要说明。所谓顺序程序是指对机床及其相关设备进行逻辑控制的程序。对于电气自动化控制工程人员而言，广泛采用的控制流程是以顺序控制为基础的。而顺序程序的编制正是在这一基础上产生的一种用于 PLC 控制的编程方式。

数控系统先将程序转换成某种格式，CPU 即可对其进行译码和运算处理。CPU 高速读出存储在存储器中的每条指令，通过算术运算来执行程序。顺序程序的编制由编制梯形图以及其它 PLC 标准语言开始的，所谓梯形图可理解为 CPU 中算术运算的执行顺序。

上述过程由 PLC 编程软件完成，PLC 编程软件的作用就是编制顺序程序。

分配接口

接口

PLC 是通过外部 I/O 与外部设备作交互的, 在确定了控制对象并计算出对应的输入/输出信号的点数后, 即可为设备分配相应的接口。

为了方便用户更加方便的进行数控 PLC 的调试, 现在已对 818 系列产品的面板接口输入输出点位作了规定, 其它的设备输入输出信号点定义具体请参考电气原理图。

配置

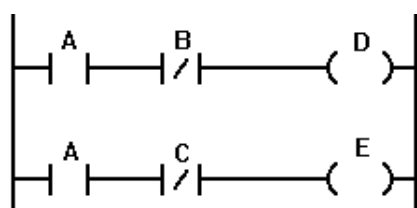
8 型系统标准 PLC 程序中已经对标准系统的面板点位做了配置处理, 无需用户再去修改定义。只是在编程时不在直接使用输入输出点寄存器进行编程, 而改用其它中间寄存器。为了使用户更好地了解该系列产品面板点位的分布情况, 将在附表 A 中对各种系统的接口进行描述, Y487、Y488 为面板上数码管的输出地址。X480 至 X491 为面板输入信号, Y480 至 Y486 为面板输出信号。

顺序程序

由于 PLC 顺序控制由软件来实现，所以和一般的继电器电路的工作原理不尽相同。因此在设计 PLC 顺序程序时应充分理解顺序控制的原理。

顺序程序 执行过程

在一般的继电器控制电路中，各继电器在时间上完全可以同时动作，在下图所举例中，当继电器 A 动作时，继电器 D 和 E 可同时动作（当触点 B 和 C 都闭合时）。在 PLC 顺序控制中，各个继电器依次动作。当继电器 A 动作时，继电器 D 首先动作，然后继电器 E 才动作（见图 2.1 (a)）。即各个继电器按梯形图中的顺序动作。



(A) 和 (B) 图显示了继电器电路和 PLC 程序动作之间的区别。

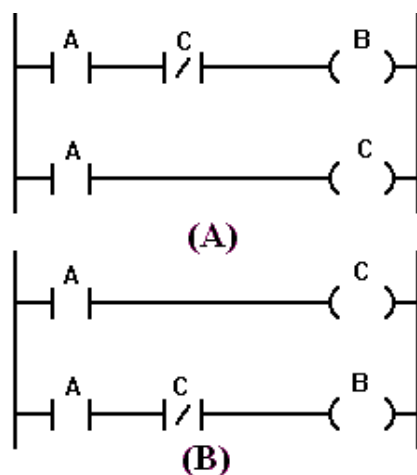


图 2.1 (b)

继电器 电路

图 2.1 (b) (A) 和 (B) 中的动作相同。接通 A (按钮开关) 后线圈 B 和 C 中有电流通过，B 和 C 接通。C 接通之后 B 断开。

PLC 程序

图 2.1 (b) (A) 中，同继电器电路一样，接通 A (按钮开关) 后，B 和 C 接通，经过 PLC 程序的一个循环周期后 B 关断。但在图 2.1 (b) (B) 中，接通 A (按钮开关) 后 C 接通，但 B 并不接通。

循环执行

顺序程序从梯形图的开头执行直至梯形图结束。在程序执行完后，再次从梯形图的开头执行，这被称作循环执行。

从梯形图的开头直至结束的执行时间称作循环处理周期。PLC2 的处理周期取决于控制的步数。处理周期越短，信号的响应速度越快。

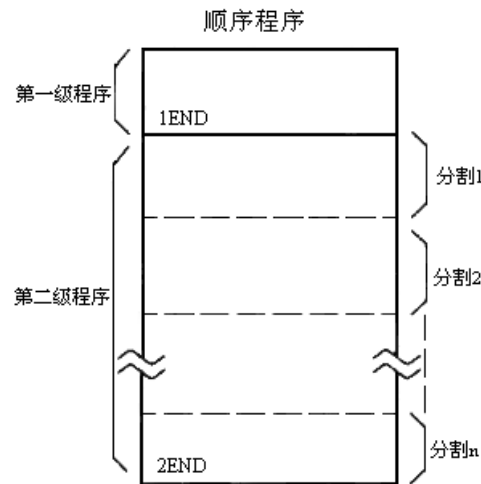
执行优先顺序

顺序程序由三部分组成：初始化程序部分、第一级程序部分和第二级程序部分。

初始化程序部分只在系统启动时执行一次。

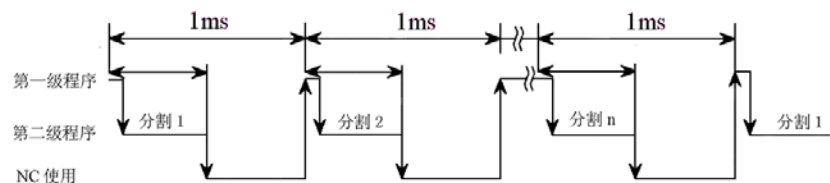
第一级程序部分每 1ms 执行一次。

如果第一级程序较长，那么总的执行时间就会延长。因此编制第一级程序时，应使其尽可能短。第二级程序每 n ms 执行一次。 n 为第二级程序的分割数。程序执行时，第二级程序将被自动分割。



第二级程序的分割

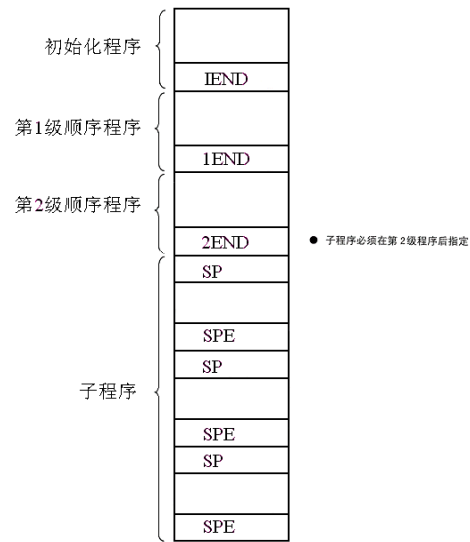
第二级程序的分割是为了执行第一级程序。当分割数为 n 时，程序的执行过程如图所示。



当最后（分割数为 n ）的第二级程序部分执行完后，程序又从头开始执行。这样当分割数为 n 时，一个循环的执行时间为 n ms ($1\text{ms} \times n$)。第一级程序每 1ms 执行一次，第二级程序每 $n \times 1\text{ms}$ 执行一次。如果第一级程序的步数增加，那么在 1ms 内第二级程序动作的步数就要相应减少，因此，分割数就要变多，整个程序处理时间变长。因此第一级程序应编得尽可能地短。

第一级程序仅处理短脉冲信号。这些信号包括急停，各轴超程等。

使用子程序时顺序程序的构成：



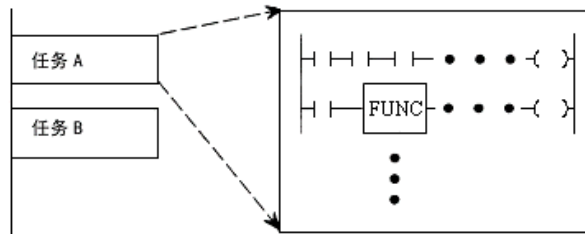
顺序程序构成

构成 在传统的 PLC 中，梯形图是只能按顺序编制的。而允许结构化编程的梯形图语言中，具有以下优点：

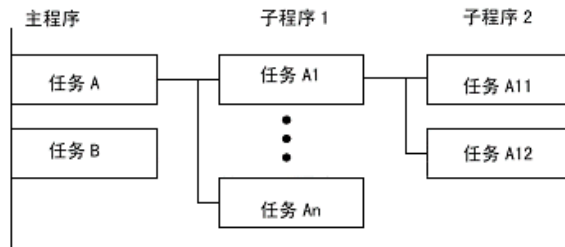
- ◆ 程序易于理解，便于编制；
- ◆ 更加方便找出编程错误；
- ◆ 出现运行错误时，易于找出原因。

主要的结构化编程方式有以下三种：

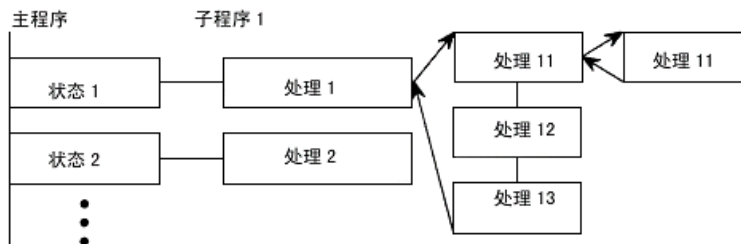
子程序 子程序以梯形图程序块为处理单元。



嵌套 由编制的子程序进行组合构成结构化程序。



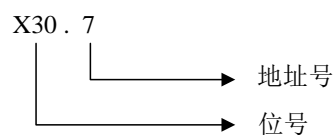
条件分支 主程序循环执行并检测条件是否满足。如果满足，执行相应的子程序。如果不满足，不执行相应的子程序。



地址

地址的定义 用来区分信号。不同的地址分别对应机床的输入、输出信号、CNC 的输入、输出信号、内部继电器、计数器等。每个地址由地址号和位号组成。

地址格式



在地址号的开头必须指定一个字母用来表示下表中所列的信号类型。

寄存器	信号类型	范围
X	来自机床的输入信号	X0~~X512
Y	由 PLC 输出到机床的信号	Y0~~Y512
F	来自 NC 的输入信号	F0~~F3119
G	由 PLC 输出到 NC 的信号	G0~~G3119
R	单字节内部继电器	R0~~R399
W	双字节内部继电器	W0~~W199
D	四字节内部继电器	D0~~D99
B	断电保持继电器	B0~~B49
P	用户自定义参数	P0~~P199
C	计数器 (C300 后为断电保持)	C0~~C127 C300~~C427
T	定时器 (T300 后为断电保持)	T0~~T127 T300~~T427
L	标记号	---
S	子程序号	---

基本指令

顺序程序主要由线圈、触点、符号和功能块等元素组成，梯形图中连接各个元素的线段构成了顺序程序的逻辑关系。可以使用梯形图或语句表语言来描述顺序程序。语句表语言需要使用助记符（LD，AND，OR 等）和寄存器地址来编写，梯形图则不必知道助记符的含义而使用继电器的线圈触点和功能块来编写。

对于 IEC61131-3 国际标准规定的梯形图语言和语句表语言而言，这两种语言在逻辑上是被规定可以互相转换的，并且可以通过某些编程手段可以避免出现翻译产生的歧义性。在 HNC_LADDER_WIN(V1.0)编辑软件中可以看到这两种语言是可以被互编译的。

为了更好的理解顺序程序的编制过程，更加清楚地了解其内部实现的过程，避免一些逻辑上或理解上的错误，以下对几个基本概念进行解释：

类型：

HNC-8 型 PLC 指令分为基本指令和功能指令两部分：

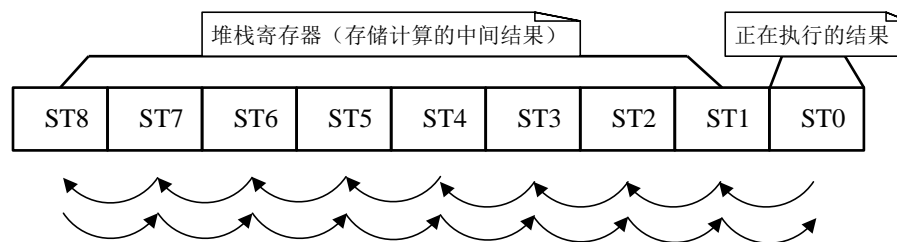
基本指令

组成顺序程序的最基本元素，也是最常用的元件，共 19 种。它们执行一位操作

功能指令

执行一些基本元素难以完成的功能，可以使用功能指令来简化编程。

逻辑结果存储器（ST）



逻辑结果存储器是类似堆栈的结构，当前指令的执行结果保存在 ST0 中，当出现 LD 或 LDI 等读取指令时，就需要对当前执行的结果进行进栈保存，当遇到 ANB 或 ORB 指令时，将 ST1 的结果出栈并与 ST0 中的结果做逻辑运算并保存在 ST0 中。所以在用语句表指令描述顺序程序时，ANB 和 ORB 必须与第一个输入指令以后的其他输入指令一一对应，否则将产生错误。

多输出逻辑结果存储器

该存储器的作用与逻辑结果存储器的作用类似，寄存当前节点的结果，常用在有条件判断的多个输出指令（MPS，MRD，MPP 用法见具体指令的介绍）。

区别在于可以允许读取该节点的结果而不需要将结果出栈。当需要嵌套使用多输出功能时才会对存储器进栈操作。同样 MPS 和 MPP 指令必须一一对应使用，否则会产生逻辑错误。

前置和后置

前置表示在该元件的前面可以连接其他的元件，后置则表示可以在后面连接其他元件。

下面是关于该说明文档内图形的约束规约：

图形	表示的含义
○	可以使用也可不使用
√	必须使用
×	不得使用
○——	可以使用也可不使用前置元件
┆——	必须使用前置元件
⋮——	不得使用前置元件
——○	可以使用也可以不使用后置元件
——┆	必须使用后置元件
——⋮	不得使用后置元件

基本指令的详述如下表：

序号	名称	功能描述
1	LD	读入指定的元件信号状态
2	LDI	读入指定的元件信号的取反状态
3	LDT	读入常真的元件信号状态
4	OUT	将逻辑运算结果输出到指定的地址
5	OOUT	将逻辑运算结果取反输出到指定的地址
6	SET	将行计算结果与指定地址中的信号逻辑或以后，将结果返回到该地址中
7	RST	将行计算结果取反与指定地址中的信号逻辑与以后，将结果返回到该地址中
8	AND	逻辑与
9	ANI	将指定信号取反后逻辑与
10	OR	逻辑或
11	ORI	将指定信号取反后逻辑或
12	LDP	读入信号的上升沿
13	LDF	读入信号的下降沿
14	ANDP	将指定信号取上升沿后逻辑与
15	ANDF	将指定信号取下降沿后逻辑与
16	ORP	将指定信号取上升沿后逻辑或
17	ORF	将指定信号取下降沿后逻辑或
18	ORB	块逻辑或
19	ANB	块逻辑与
20	MPS	节点结果入栈
21	MRD	读节点结果
22	MPP	节点结果出栈

LD

格式



功能描述

读出指定地址的状态信号（1 或 0），并保存到 ST0 中
用于从常开节点开始编程的场合。

参数说明

寄存器点位参数。

举例

梯形图									
	运行过程	序号	指令	地址	位号	说明	ST2	ST1	ST0
		1	LDI	X1	. 0	A			\overline{A}
		2	ANI	X2	. 0	B			$\overline{A.B}$
		3	ANI	R1	. 0	C			$\overline{A.B.C}$
		4	OUT	R10	. 0	输出 W1			$\overline{A.B.C}$
5	LD	X5	. 0	D			D		

	6	ORI	X5 . 1	E			$D + \bar{E}$
	7	OR	X5 . 3	F			$D + \bar{E} + F$
	8	AND	X5 . 4	G			$(D + \bar{E} + F)G$
	9	OUT	R10 . 0	输出 W2			$(D + \bar{E} + F)G$
描述							

LDI

格式



功能描述

出指定地址的状态信号（1 或 0），取反并保存到 ST0 中。
用于从常闭节点开始编程的场合。

参数说明

寄存器点位参数

举例

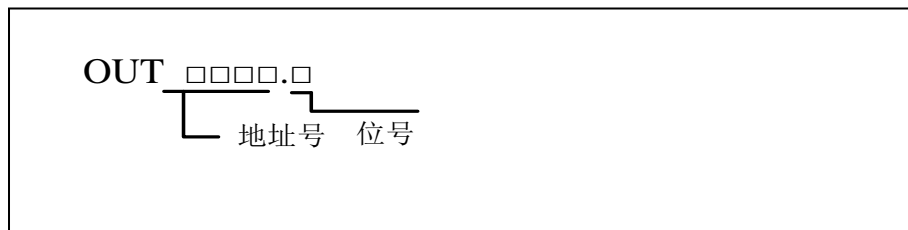
梯形图

运行过程	序号	指令	地址	位号	说明	ST2	ST1	ST0
	1	LDI	R1	. 0	A			\overline{A}
	2	ANI	R2	. 0	B			$\overline{A.B}$
	3	ANI	R1	. 1	C			$\overline{A.B.C}$
	4	OUT	R10	. 0	W1			$\overline{A.B.C}$
	5	LDI	X5	. 0	D			\overline{D}

	6	ORI	X5 . 1	E			$\overline{D} + \overline{E}$
	7	OR	X5 . 3	F			$\overline{D} + \overline{E} + F$
	8	AND	R5 . 4	G			$(\overline{D} + \overline{E} + F)G$
	9	OUT	R10 . 1	W2			$(\overline{D} + \overline{E} + F)G$
描述							

OUT

格式



功能描述

将逻辑计算结果。即 ST0 的状态输出到指定的地址中。
可以用于将结果输出到一个或一个以上地址中。

参数说明

寄存器点位参数

举例

梯形图								
	运行过程	序号	指令	地址	位号	说明	ST2	ST1
	1	LDI	R1	. 0	A			\bar{A}
	2	ORI	X5	. 0	C			$\bar{A} + \bar{C}$
	3	ANI	G1	. 1	B			$(\bar{A} + \bar{C})\bar{B}$
	4	OUT	R10	. 0	W1			$(\bar{A} + \bar{C})\bar{B}$
	5	OUT	R10	. 1	W2			$(\bar{A} + \bar{C})\bar{B}$
描述	串联电路与并联电路的例子。							

OOOUT

格式



功能描述

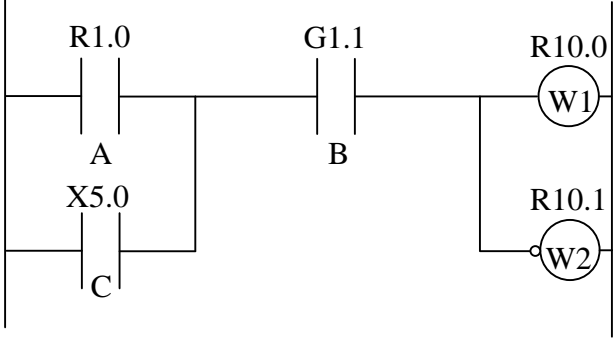
将逻辑计算结果。STO 的状态取反输出到指定的地址中。

可以用于将结果输出到一个或一个以上地址中。

参数说明

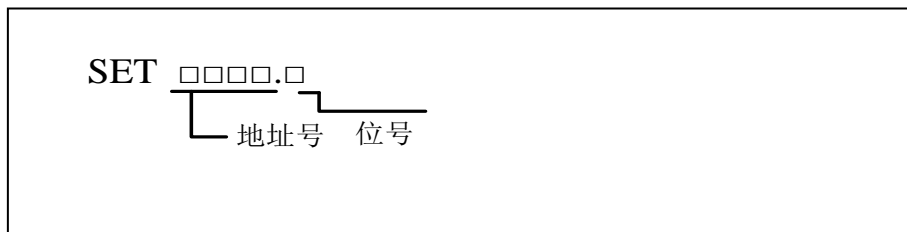
寄存器点位参数

举例

梯形图								
运行过程	序号	指令	地址	位号	说明	ST2	ST1	ST0
	1	LD	R1	. 0	A			A
	2	OR	X5	. 0	C			A + C
	3	AND	G1	. 1	B			(A + C).B
	4	OUT	R10	. 0	W1			(A + C).B
	5	OUT	R10	. 1	W2			(A + C).B
描述								

SET

格式



功能描述

将逻辑操作结果 ST0 与所指地址的内容进行逻辑或，并将结果输出至相同地址。

参数说明

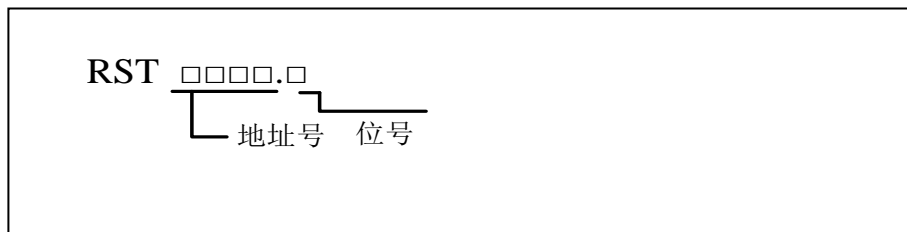
寄存器点位参数

举例

梯形图								
	序号	指令	地址	位号	说明	ST2	ST1	ST0
	1	LD	R1	. 0	A			A
	2	OR	X5	. 0	B			A + B
3	SET	R10	. 0	C			A + B	
运行过程								
描述								

RST

格式



功能描述

将逻辑操作结果 ST0 取反与所指地址的内容进行逻辑与，并将结果输出至相同地址

参数说明

寄存器点位参数

举例

梯形图								
	序号	指令	地址	位号	说明	ST2	ST1	ST0
	1	LD	R1	. 0	A			A
	2	OR	X5	. 0	B			A + B
运行过程	3	RST	R10	. 0	C			A + B
描述								

AND

格式

AND □□□□.□
└───┬───┘
地址号 位号

功能描述

逻辑与。

参数说明

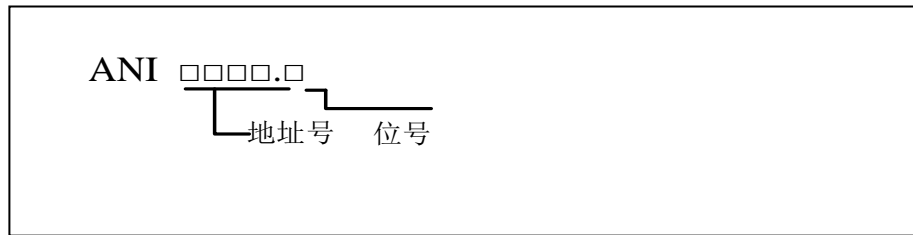
寄存器点位参数。

举例

见 LD 命令举例。

ANI

格式



功能描述

逻辑与非。

参数说明

寄存器点位参数。

举例

见 LD 命令举例。

OR

格式

OR □□□□.□
 └───┬───┘
 地址号 位号

功能描述

逻辑或。

参数说明

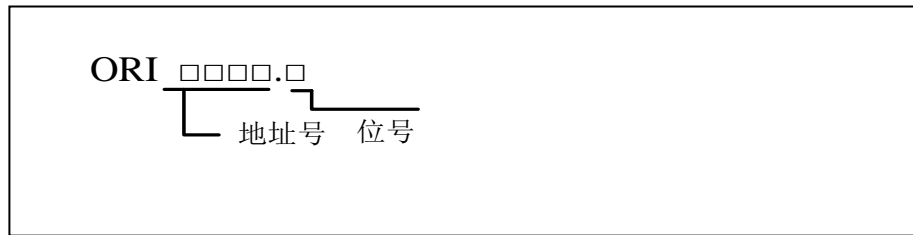
寄存器点位参数。

举例

见 LDI 命令举例。

ORI

格式



功能描述

逻辑或非。

参数说明

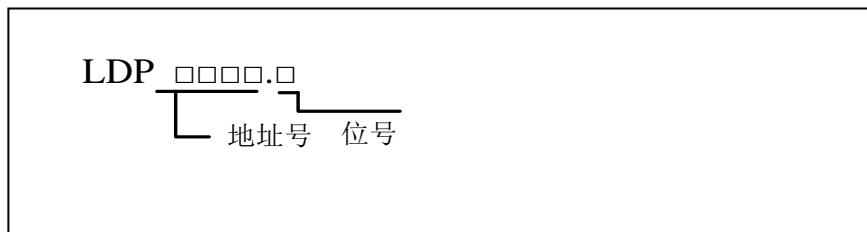
寄存器点位参数。

举例

见 LDI 命令举例。

LDP

格式



功能描述

获取上升沿触发元件信号，并将其保存到 ST0 中。

在输入信号上升沿的下一扫描周期中将输入信号设置为 1。

适合以上升沿元件开始编程的场合

参数说明

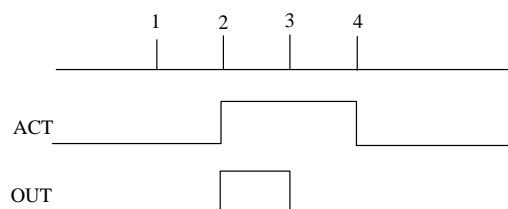
寄存器点位参数

控制条件

输入信号：在信号上升沿处（0->1）,将输出信号设置为 1

输出信号：在执行过程中，在一个 PLC 扫描周期以内输入信号保持为 1。

操作

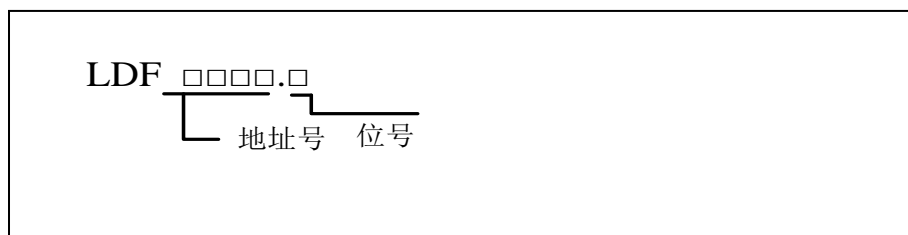


举例

梯形图					
运行过程	序号	指令	地址	位号	说明
	1	LDP	R1	. 0	取 A 上升沿
	2	ORF	X5	. 0	取 B 下降沿
	3	ANDP	R2	. 0	取 C 上升沿
	4	ANDF	R4	. 0	取 D 下降沿
	5	OUT	R10	. 1	输出 W1

LDF

格式



功能描述

获取下降沿触发元件信号，并将其保存到 **STO** 中。

在输入信号下降沿的扫描周期中将输入信号设置为 1。

适合以下降沿元件开始编程的场合。

参数说明

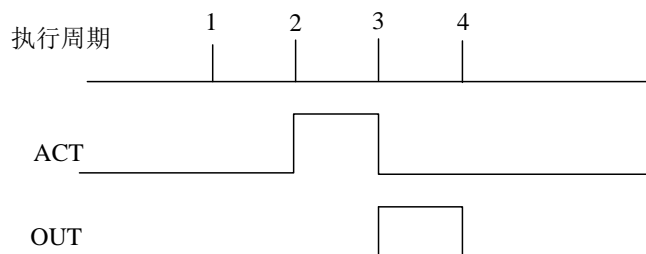
寄存器点位参数

控制条件

输入信号：在信号下降沿处（1->0），将输出信号设置为 1；

输出信号：在执行过程中，在一个 PLC 扫描周期以内输入信号保持为 1。

操作

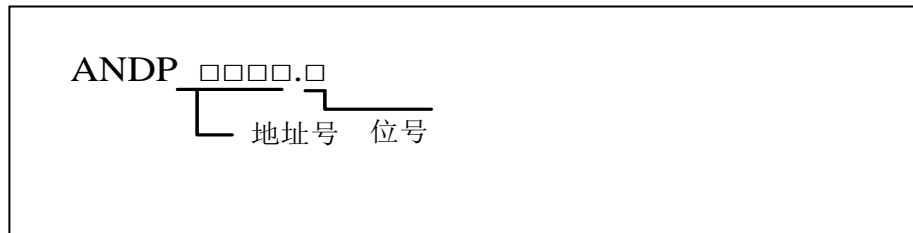


举例

如 LDP 例子。

ANDP

格式



功能描述

逻辑与上升沿。

参数说明

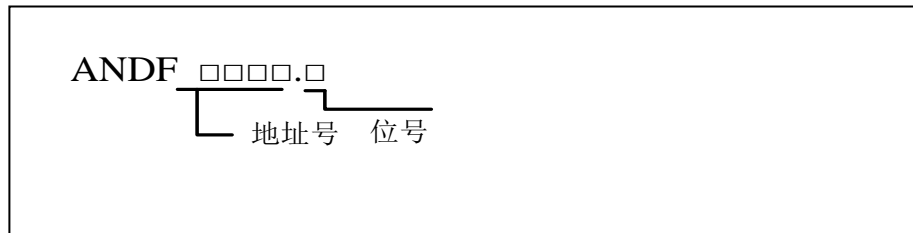
寄存器点位参数。

例子

见 LDP 例子。

ANDF

格式



功能描述

逻辑与下降沿。

参数说明

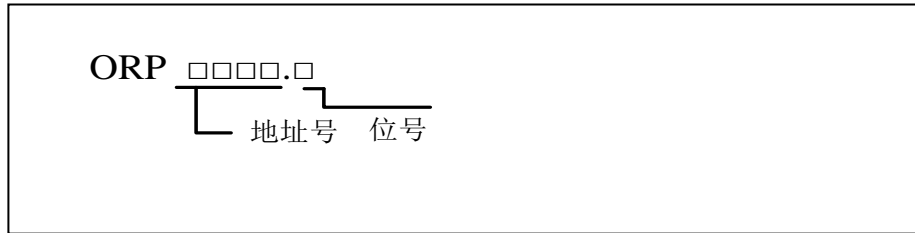
寄存器点位参数。

例子

见 LDP 例子。

ORP

格式



功能描述

逻辑或上升沿。

参数说明

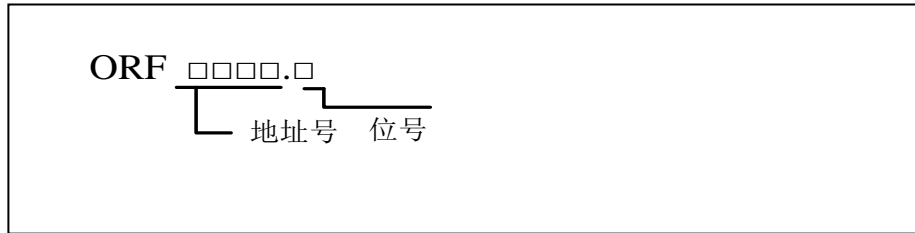
寄存器点位参数。

例子

见 LDP 例子。

ORF

格式



功能描述

逻辑或下降沿。

参数说明

寄存器点位参数。

例子

见 LDP 例子。

ORB

格式

ORB

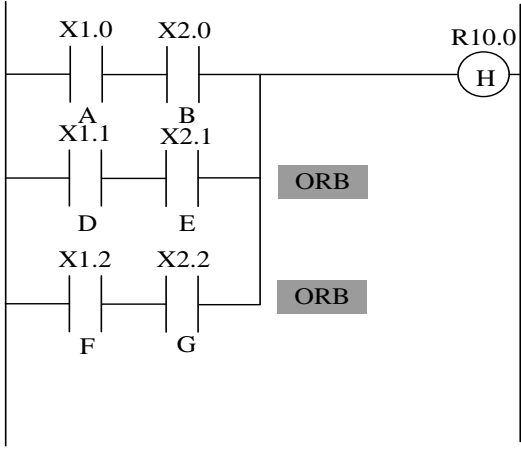
功能描述

- 1) ORB 是一个独立的结构，它不需要和其他元件或功能块相连。
- 2) ORB 是用于连接两个或两个以上的串行回路，串行回路是包含一个以上串联连接的块或包含有 ANB 串块的结构。
- 3) 以 LD 或 LDI 开始编程，通过 ORB 来完成所有串联块的并联合并。

参数说明

无参数。

举例

梯形图								
运行过程	序号	指令	地址	位号	说明	ST2	ST1	ST0
	1	LD	X1.	0	A			A
	2	AND	X2.	0	B			<i>A.B</i>
	3	LD	X1.	1	D		<i>A.B</i>	<i>D</i>
	4	AND	X2.	1	E		<i>A.B</i>	<i>D.E</i>
	5	ORB						<i>AB+D.E</i>
	6	LD	X1.	2	F		<i>AB+D.E</i>	<i>F</i>
	7	AND	X2.	2	G		<i>AB+D.E</i>	<i>F.G</i>
	8	ORB						<i>AB+D.E+F.G</i>
	9	OUT	R10.	1	H			<i>AB+D.E+F.G</i>
描述								

ANB

格式

ANB

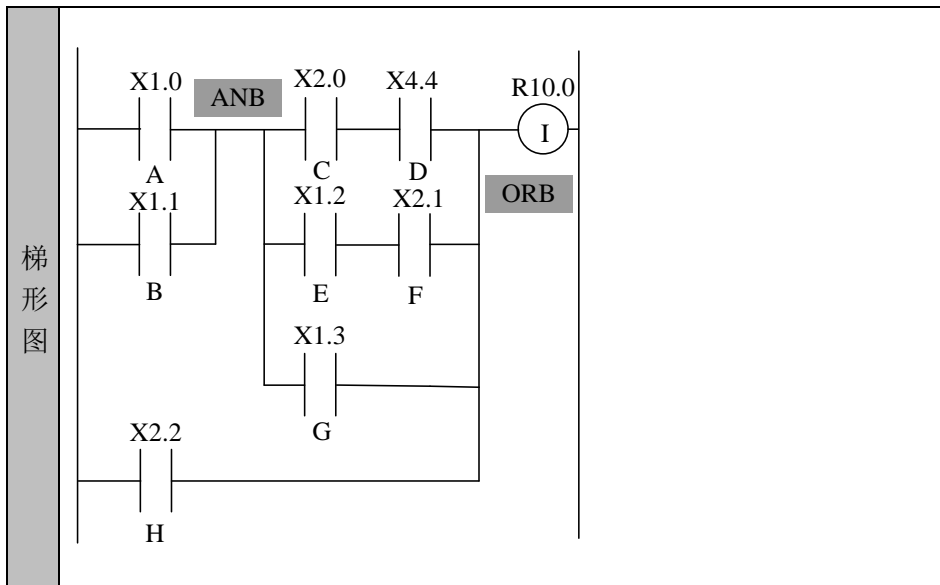
功能描述

- 1) ANB 是一个独立的结构，它不需要和其他元件或功能块相连。
- 2) ANB 是用于连接两个或两个以上的并行回路，并行回路是包含一个以上并行连接的块或包含有 ORB 并联块的结构。
- 3) 以 LD 或 LDI 开始编程，通过 ANB 来完成所有并联块的串联。

参数说明

无参数。

举例



运行过程

序号	指令	地址	位号	说明	ST2	ST1	ST0
1	LD	X1.	0	A			A.
2	OR	X1.	1	B			A + B
3	LD	X2.	0	C		A+B	C
4	AND	X4.	4	D		A+B	C.D
5	LD	X1.	2	E	A+	C.D	E
6	AND	X2.	1	F	A+	C.D	E.F
7	ORB					A+B	C.D + E.F
8	OR	X1.	3	G		A+B	C.D + E.F + G
9	ANB						(A+B)(C.D + E.F + G)
10	OR	X2.	2	H			(A+B)(C.D + E.F + G) + H
11	OUT	R10.	0	I			(A+B)(C.D + E.F + G) + H

描述

MPS、MRD、MPP

格式

MPS

MRD

MPP

功能描述

- 1) MPS 存储该点的信号状态，等待其他行的输出时被使用。
- 2) MRD 从上一个存储点中读取信号，连接到下一个节点且保持该点的信号状态不变。
- 3) MPP 从该存储点中弹出信号状态，它将首先连接到下一个节点然后移除该点的状态。
- 4) 每个 MPS 都必须配对有一个 MPP 作为结束。
- 5) 最后一个连接行必须使用 MPP 作为结束。

参数说明

无参数。

举例

梯形图	语句表	
	<pre>LD X1.0 MPS AND X1.6 ORB LD X1.1 ANB OR X1.2 OUT Y0.2 ANB MPP OUT Y1.0 AND X1.7 MRD OUT Y0.3 LD X1.3 LD X2.3 AND X1.4 OR X2.4 LD X1.5 ANB (转右表) OUT Y0.4</pre>	
	<pre>LD X1.0 MPS AND X1.1 AND X1.4 MPS MPP AND X1.2 AND X1.5 OUT Y1.0 OUT Y0.2 MPP MPP AND X1.3 AND X1.6 (转右表) OUT Y2.0</pre>	
	<pre>LD X1.0 MPS AND X1.1 AND X1.4 MPS MPP AND X1.2 AND X1.5 OUT Y1.0 OUT Y0.2 MPP MPP AND X1.3 AND X1.6 (转右表) OUT Y2.0</pre>	

基本元件

本章主要内容

3.1 常开触点

3.2 常闭触点

3.3 常真触点

3.4 触点上升沿

3.5 触点下降沿

3.6 逻辑输出

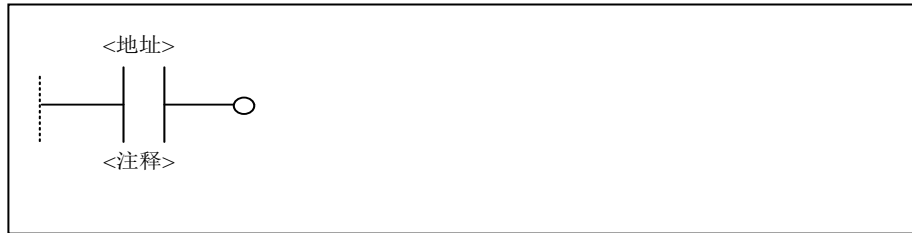
3.7 逻辑反输出

3.8 置位输出

3.9 复位输出

常开触点

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	X、Y、F、G、R、W、D、P、T、C、B	要检查的寄存器位	前置 ○ 后置 ✓

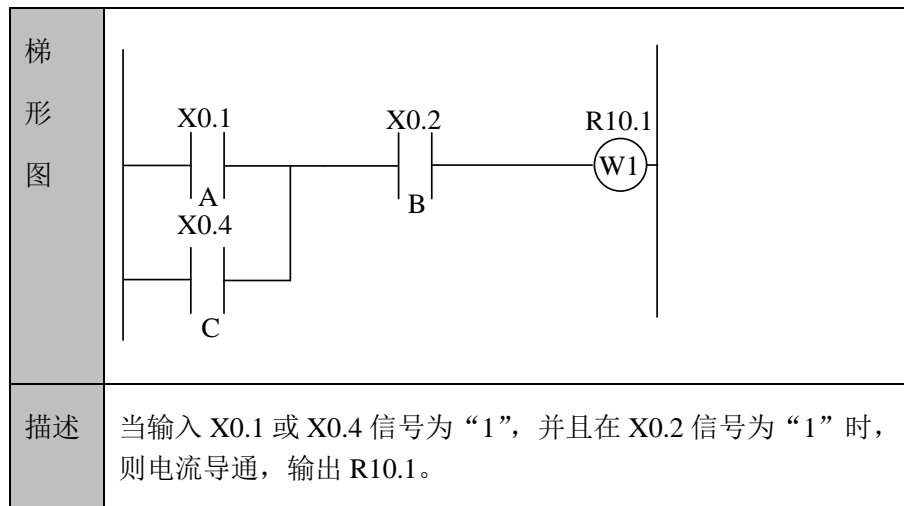
功能描述

存在指定地址中的位的值为“1”时，常开触点闭合。当触点闭合时，信号流将流过该接点。

参数说明

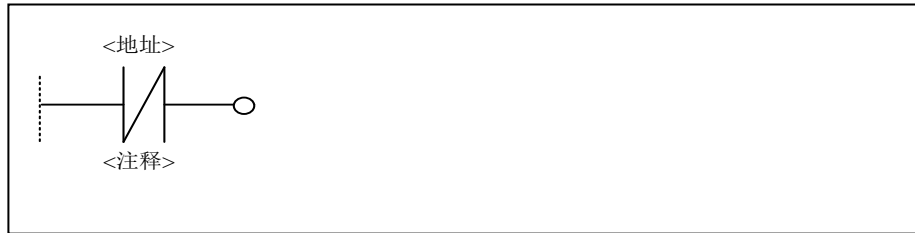
参数 1: 寄存器点位参数，形如 X0.1。

示例



常闭触点

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	X、Y、F、G、R、W、D、P、T、C、B	要检查的寄存器位	前置 ○
					后置 ✓

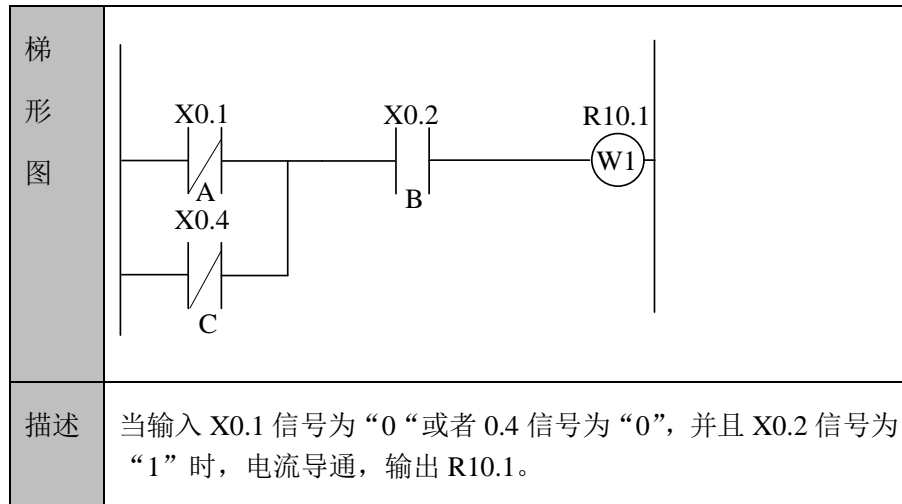
功能描述

当保存在指定地址中的位的值为“0”时，常闭触点开启。当触点开启时，信号流将流过该接点。

参数说明

参数 1: 寄存器点位参数，形如 X0.1。

示例



常真触点

符号



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无	无	前置 ○
					后置 ✓

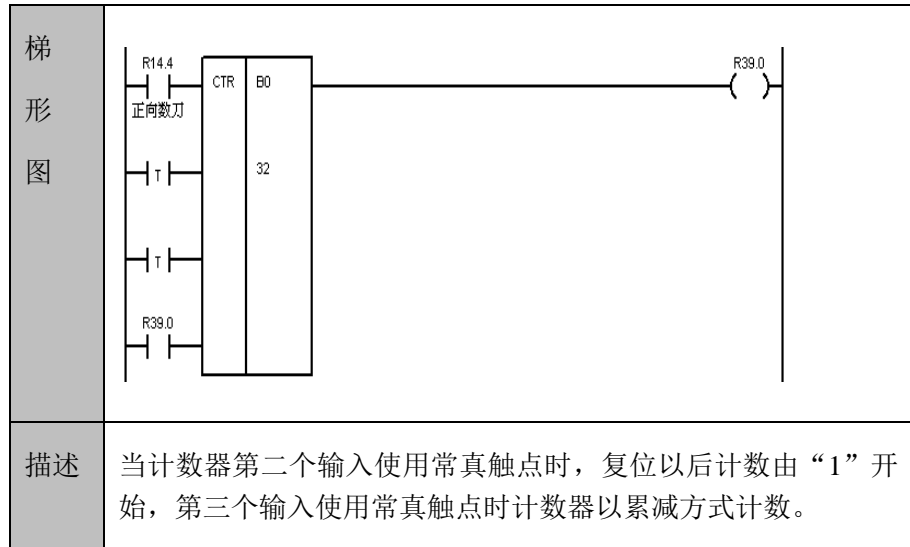
功能描述

当 PLC 上电以后，元件左边的信号始终可以通过元件到达元件右边。通常用作功能模块输入的开关设置和需要提供恒定有效输入的功能。

参数说明

无参数。

示例



触点上升沿

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	X、Y、F、 G、R、W、 D、P、T、 C、B	上升沿 检查触 点。	前置 ○ 后置 ✓

功能描述

当信号由“0”变为“1”时该触点导通。

参数说明

参数 1: 寄存器位。

示例

梯形图	见 LDP 指令举例。
描述	

触点下降沿

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	X、Y、F、G、R、W、D、P、T、C、B	下降沿检查触点。	前置 ○
					后置 ✓

功能描述 当信号由“1”变为“0”时该触点导通。

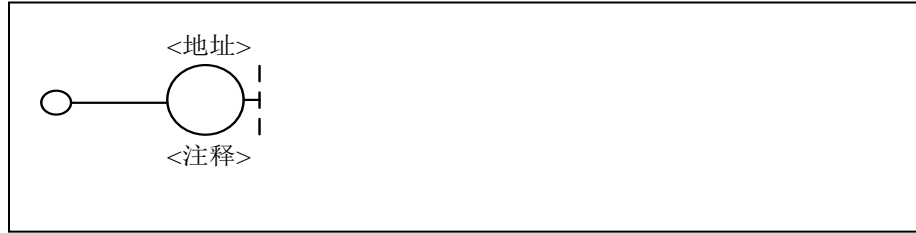
参数说明 参数 1: 寄存器位

示例

梯形图	见 LDF 指令举例。
描述	

逻辑输出

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	Y、G、R、 W、D、B	输出线圈	前置 ○
					后置 ×

功能描述 将逻辑运算结果输出到输出寄存器。

参数说明 参数 1: 寄存器位。

示例

梯形图	见 OUT 指令举例。
描述	

逻辑反输出

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	Y、G、R、 W、D、B	取反输出线圈	前置 ○ 后置 ×

功能描述

将逻辑运算结果取反输出到输出寄存器。

参数说明

参数 1：寄存器位。

示例

梯形图	见 OOUT 指令举例。
描述	

置位输出

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	Y、G、R、 W、D、B	置位输出线圈	前置 ○
					后置 ×

功能描述

如果逻辑运算结果为“1”时设置输出线圈为输出状态直到该线圈被其他功能复位。

参数说明

参数 1: 寄存器位。

示例

梯形图	见 SET 指令举例。
描述	

复位输出

符号



参数	参数格式	数据类型	存储区域	说明	属性
<地址>	□□□□.□	BOOL	Y、G、R、 W、D、B	复位输出线圈	前置 ○ 后置 ×

功能描述

如果逻辑运算结果为“1”时复位输出线圈，直到该线圈被其他功能对其置位。

参数说明

参数 1: 寄存器位。

示例

梯形图	见 RST 指令举例。
描述	

基本功能模块

本章主要内容

4.1 控制指令

4.2 数学运算

4.3 计数器

4.4 定时器

4.5 流程控制

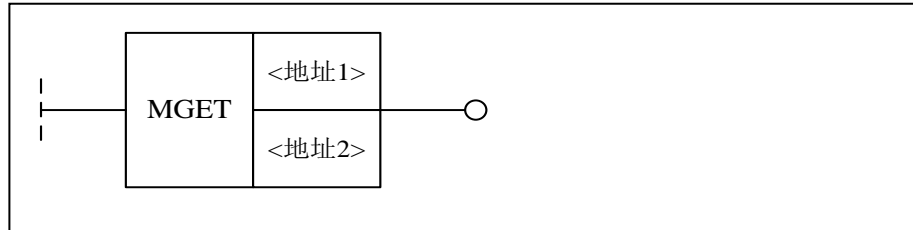
4.6 比较

4.7 数据操作

控制指令

M 指令获取 MGET

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数	M 代码号	后置 ✓

功能说明

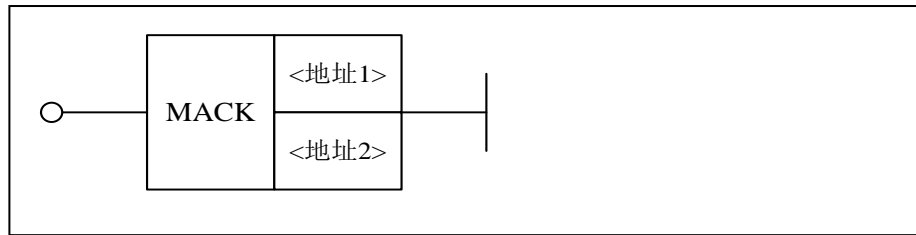
通过参数 1 选择的通道，参数 2 选择需要判断的 M 代码序号，当该通道获取到了该 M 代码，则输出为“1”，否则输出为“0”。

示例

梯形图	<p>The ladder logic diagram shows a normally open contact labeled X2.0 connected to an MGET instruction box. The MGET box has two parameters: 0 (channel) and 3 (M code). The output of the MGET instruction is connected to a set coil labeled R4.0.</p>
语句表	<pre>LD X2.0 MGET 0 3 WRT R4.0</pre>
描述	当 0 通道执行 M3 时，将 R4.0 置位

M 指令应答 MACK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数	M 代码号	后置 ×

功能说明

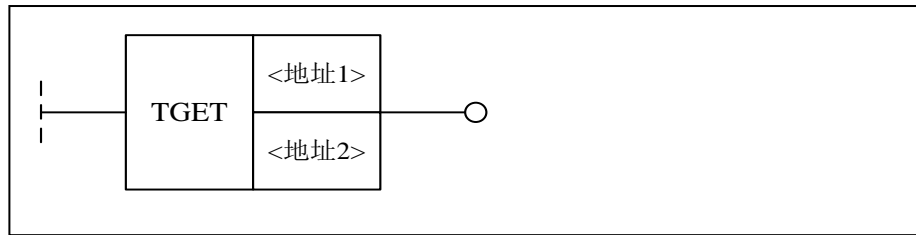
当该通道有 M 代码执行完毕需要对该 M 代码应答,当应答完成表示该 M 指令可以继续下面的指令。

示例

梯形图	
语句表	<pre>LD X3.6 MACK 0 3</pre>
描述	当 X3.6 有效时,对 0 通道的 M3 做出应答。

T 指令获取 TGET

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	常数、Y、G、R、W、D、B	T 代码号	

功能说明

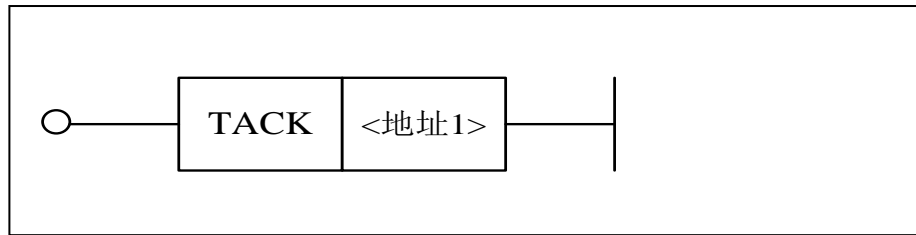
通过参数 1 选择的通道，参数 2 为获取 T 代码的存储位置，当该通道获取到 T 代码，则输出为“1”，否则输出为“0”

示例

梯形图	
语句表	<pre>LDT TGET 0 R5 OUT R6.1</pre>
描述	当 0 通道执行到 T 指令时，将 T 指令参数送到 R5 寄存器，并将 R6.1 置位。

T 指令应答 TACK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○ 后置 ×

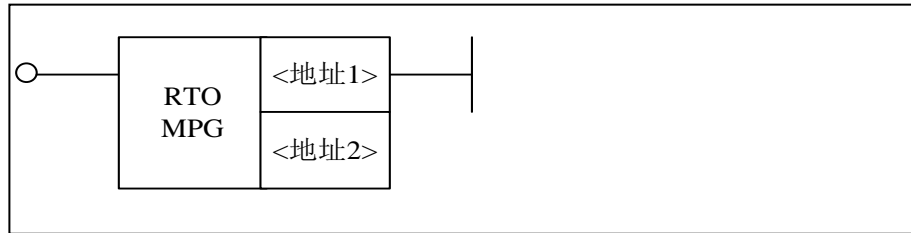
功能说明 通过参数 1 选择的通道，设置该通道 T 代码应答完成

示例

梯形图	
语句表	<pre>LD X3.4 TACK 0</pre>
描述	当 X3.4 导通时，应答该通道 T 指令。

手摇控制 RTOMPG

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ○ 后置 ×
<地址 2>	□□□□	INT	常数		

功能说明 手摇控制(8 型专用)。

参数说明 参数 1: 手摇脉冲增量输入的寄存器。(8 型手摇默认使用 X490 寄存器)。

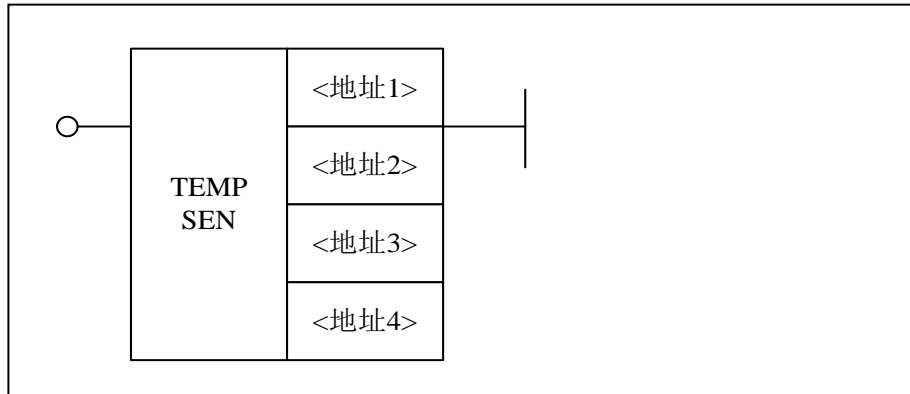
参数 2: MPG 编号 , 此参数用于描述手摇的编号, 当出现多个手摇时可用此参数区分。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 RTOMPG X40 0</pre>
描 述	<p>当 X3.4 导通时开启手摇控制。手摇脉冲控制寄存器 X40 设置到手摇 0 中。</p>

热误差补偿模块 TEMPSEN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○ 后置 ×
<地址 2>	□□□□. □	BOOL	X		
<地址 3>	□□□□	INT	常数		
<地址 4>	□□□□. □	BOOL	P		

功能说明

温度传感器模拟信号通过 IO 模块 AD 转换为数字信号并输入到 X 寄存器。数字信号输入到 X 寄存器的位置（组号）由 IO 模块设备参数确定。

参数说明

参数 1: 温度传感器编号（即温度寄存器序号），HNC-8 数控系统最多支 20 个温度采集信号输入，因此温度传感器编码取值范围 0~19。

参数 2: 温度采集数字信号对应 X 寄存器组号。

参数 3: 测温板类型（默认值 0；1: 通过“参数 4”指定的用户参数包括最小温度及最大温度[为 6.7V 对应的温度]，来建相应模型计算温度；2: 支持 PT100 型温度传感器，连接 HIO-1075 测温板；3: 支持 KTY84-300 型温度传感器，连接 HIO-1076 测温板；4: 测量的温度与通过所输入 D A 值换算出的电阻成线性关系，通过“参数 4”指定的用户 P 参数包括最小温度、最大温度及其对应的最小电阻、最大电阻[单位: 0.01 欧]来建立相应模型计算温度。（注：其中 2、3 为标准配置，可与相应总线式测温板直接连接，系统内部有相应温度模型，不需

要设置 P 参数内数值)。

参数 4: 通过用户参数 (P 参数) 配置温度传感器采集温度值范围, 如下图所示, P30 指定采集最小温度值, P31 指定采集最大温度值 (6.7V 电压对应的温度, 单位: 度), 当测温板类型为 2、3 时将不读取所设 P 参数值。

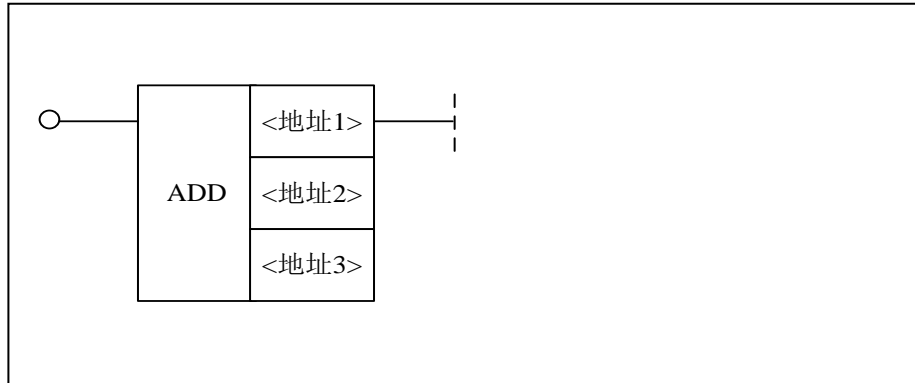
示例

梯 形 图	
语 句 表	TEMPSEN 0 X15 0 P30
描 述	0 号温度传感器采集后的温度放入 X15 寄存器中, P30 中指定最小采集温度值

数学运算

加法 ADD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、 W、D、B		

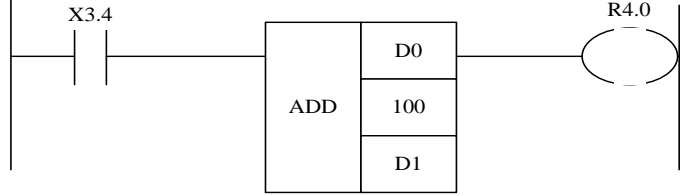
功能说明 进行加法运算。

参数说明 参数 1：被加数；

参数 2：加数；

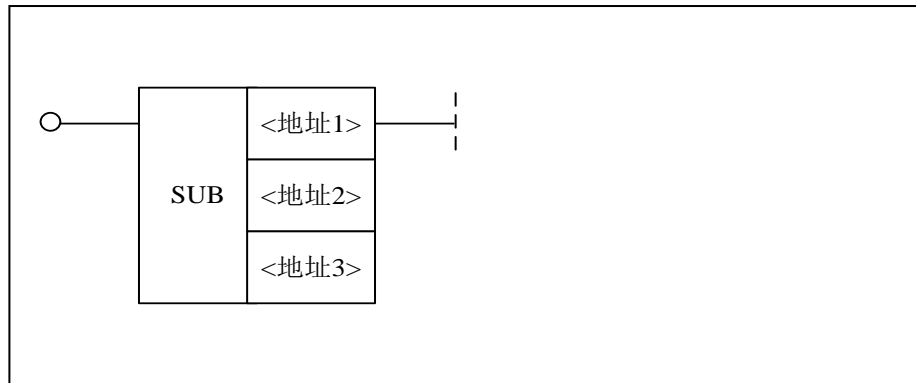
参数 3：运算结果输出地址。

示例

梯 形 图	 <p>The diagram shows a single ladder logic step. It begins with a normally open contact labeled X3.4. This contact is connected to the left side of a rectangular instruction block labeled 'ADD'. The 'ADD' block has three distinct input sections on its right side, labeled 'D0', '100', and 'D1' from top to bottom. A line connects the right side of the 'ADD' block to a coil symbol labeled R4.0, which is represented by a circle with a horizontal line through its center.</p>
语 句 表	<pre>LD X3.4 ADD D0 100 D1 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 $D1=D0+100$

减法 SUB

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、W、 D、B		

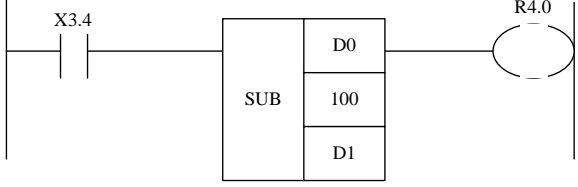
功能说明 进行减法运算。

参数说明 参数 1：被减数；

参数 2：减数；

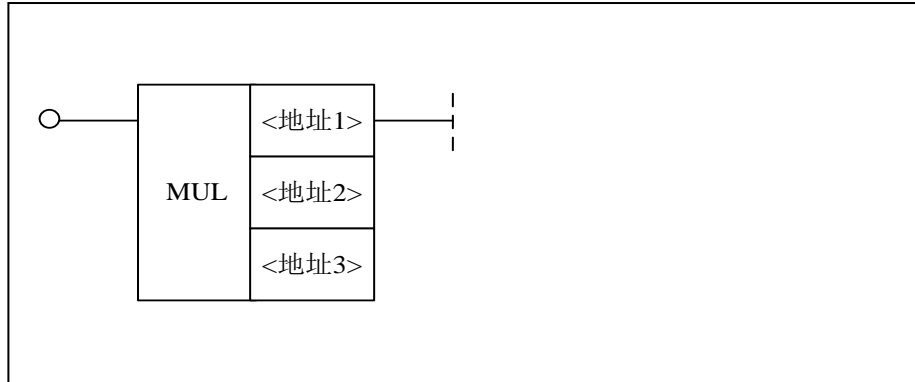
参数 3：运算结果输出地址。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 SUB D0 100 D1 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 $D1=D0-100$

乘法 MUL

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、W、 D、B		

功能说明 进行乘法运算。

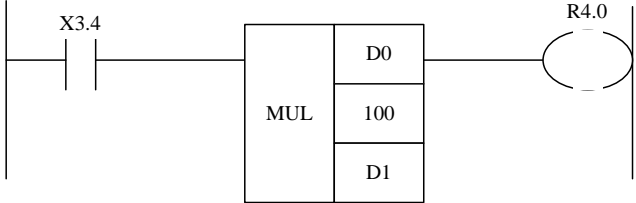
参数说明

参数 1: 被乘数;

参数 2: 乘数;

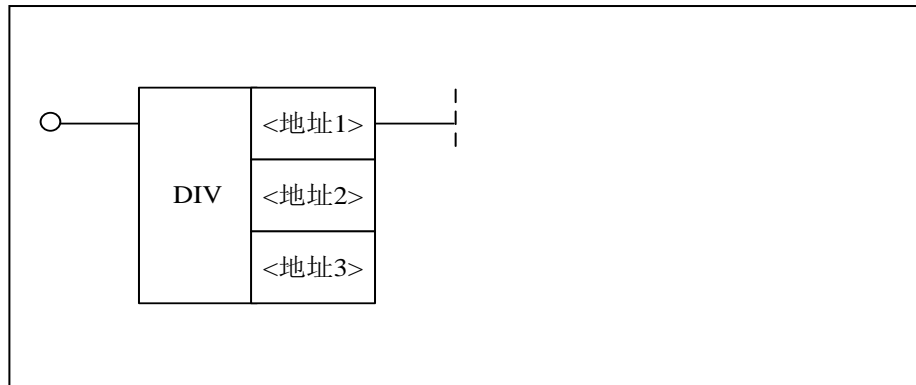
参数 3: 运算结果输出地址。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 MUL D0 100 D1 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 $D1=D0*100$

除法 DIV

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、W、 D、B		

功能说明 进行除法运算。

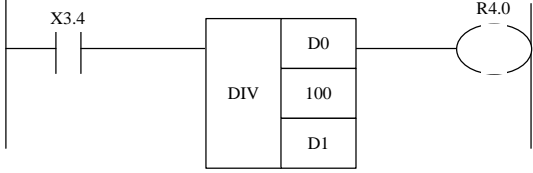
参数说明

参数 1: 被除数;

参数 2: 除数, 此数不得为 0;

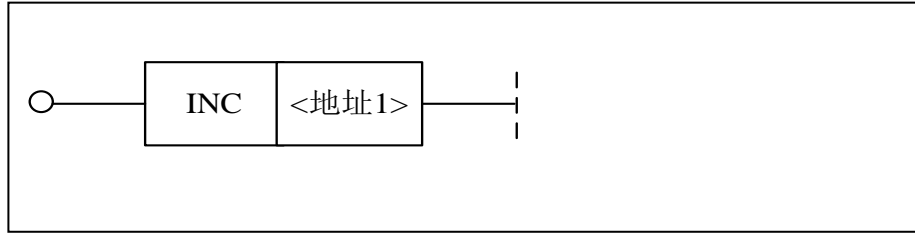
参数 3: 运算结果输出地址。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 DIV D0 100 D1 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 $D1=D0/100$

加一 INC

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	Y、G、R、 W、D、B		前置 √ 后置 ○

功能说明 进行自加 1 运算。

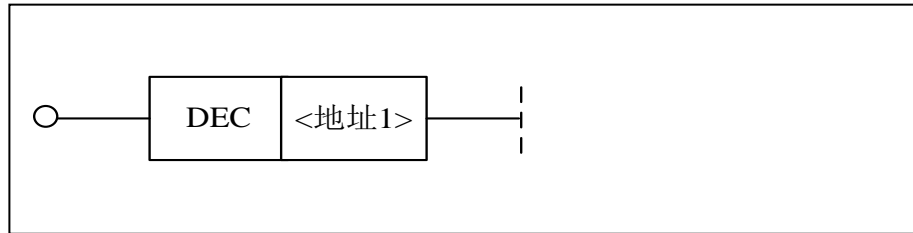
参数说明 参数 1：操作数。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 INC D0 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 D0=D0+1

減一 DEC

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	Y、G、R、 W、D、B		前置 √ 后置 ○

功能说明 进行自减 1 运算。

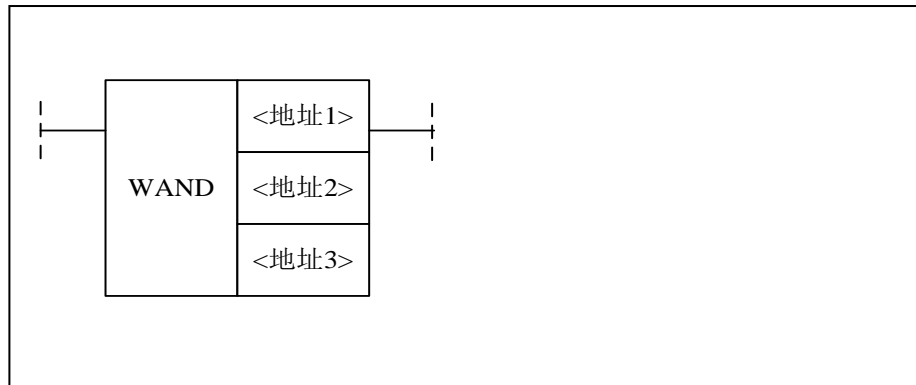
参数说明 参数 1：操作数。

示例

梯形图	
语句表	<pre>LD X3.4 DEC D0 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 D0=D0-1。

逻辑与 WAND

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、 W、D、B		

功能说明 进行逻辑与运算。

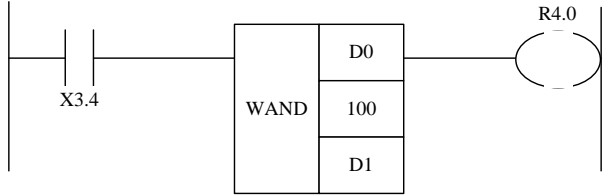
参数说明

参数 1: 被运算数;

参数 2: 运算数;

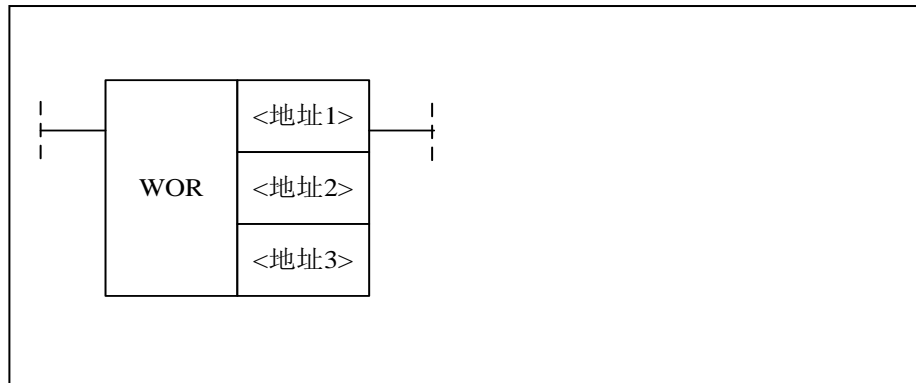
参数 3: 运算结果输出地址。

示例

梯形图	
语句表	<pre>LD X3.4 WAND D0 100 D1 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 $D1=D0\&100$ 。

逻辑或 WOR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 <input checked="" type="checkbox"/> 后置 <input type="checkbox"/>
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、 W、D、B		

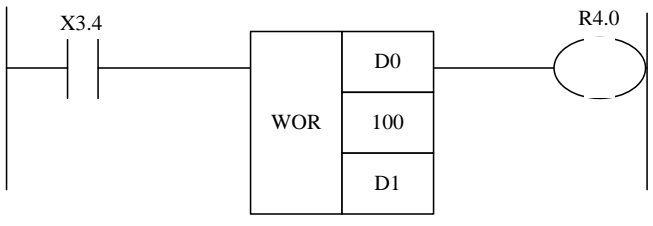
功能说明 进行逻辑或运算。

参数说明 参数 1：被运算数；

参数 2：运算数；

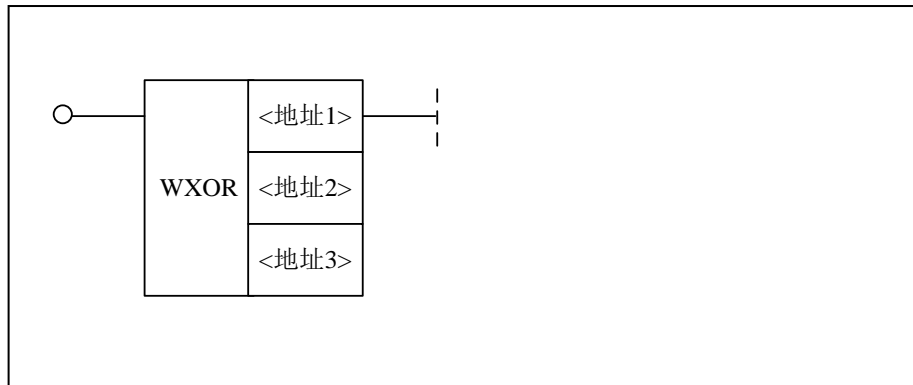
参数 3：运算结果输出地址。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 WOR D0 100 D1 OUT R4.0</pre>
描 述	<p>当 X3.4 导通时，执行 $D1=D0 100$。</p>

逻辑异或 WXOR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		前置 √ 后置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	Y、G、R、 W、D、B		

功能说明 进行异或运算。

参数说明

参数 1: 被运算数;

参数 2: 运算数;

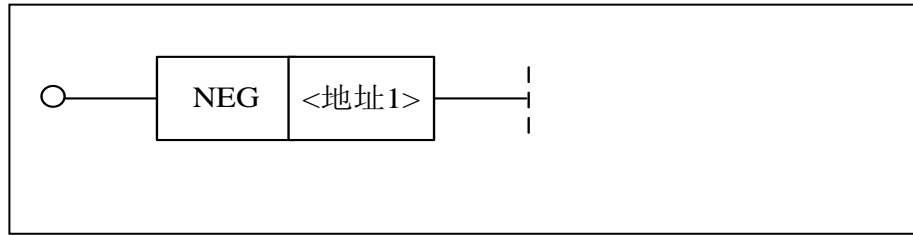
参数 3: 运算结果输出地址。

示例

梯 形 图	
语 句 表	<pre>LD X3.4 WXOR D0 100 D1 OUT R4.0</pre>
描 述	<p>当 X3.4 导通时，执行 $D1=D0\wedge 100$。</p>

求补 NEG

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	Y、G、R、 W、D、B		前置 ✓ 后置 ○

功能说明 进行求补运算。

参数说明 参数 1：操作数。

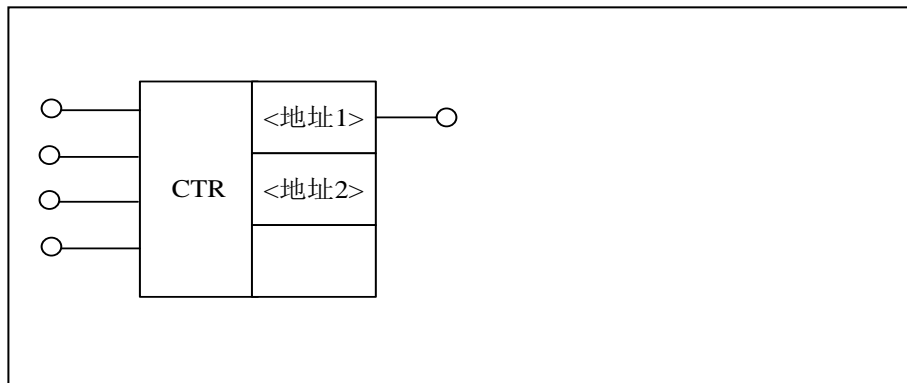
示例

梯形图	
语句表	<pre>LD X3.4 NEG D0 OUT R4.0</pre>
描述	当 X3.4 导通时，执行 D0=-D0。

计数器

加减计数器 CTR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□. □	BOOL	R、W、D、B		前置 ✓ 后置 ✓
<地址 2>	□□□□	INT	常数、R、W、D、B、P		

功能说明 一般加减计数器。

参数说明 参数 1: 计数器当前值, 用于获取当前的计数值;

参数 2: 计数器预置值。

输入说明 输入 1: 控制输入;

输入 2: 复位后的起始值, 导通为从 1 开始, 断开为从 0 开始;

输入 3: 加减输入, 导通为减计数, 断开为加计数;

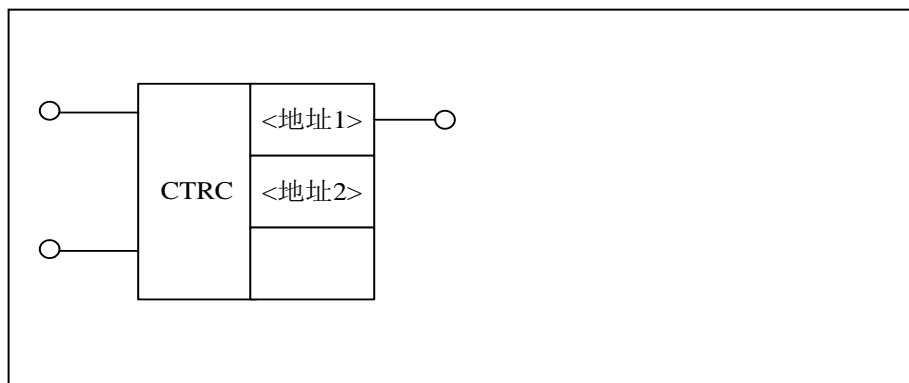
输入 4: 复位输入。

示例

<p>梯 形 图</p>	
<p>语 句 表</p>	<pre>LD X2.5 LD X4.0 LD X4.1 LD X4.2 CTR R0 16 OUT Y1.4</pre>
<p>描述</p>	<p>X4.0 导通，计数器从 1 开始计数，X4.1 导通为减计数，当 X2.5 导通 16 次输出至 Y1.4，X4.2 为复位计数器信号，清除计数器输出，当 X2.5 导通 5 次，R0 的值为 6。</p>

计数器 CTRC

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
<地址 2>	□□□□	INT	常数、R、W、 D、B、P		后置 ✓

功能说明 固定计数器。

参数说明 参数 1：计数器编号；
参数 2：计数器预置值。

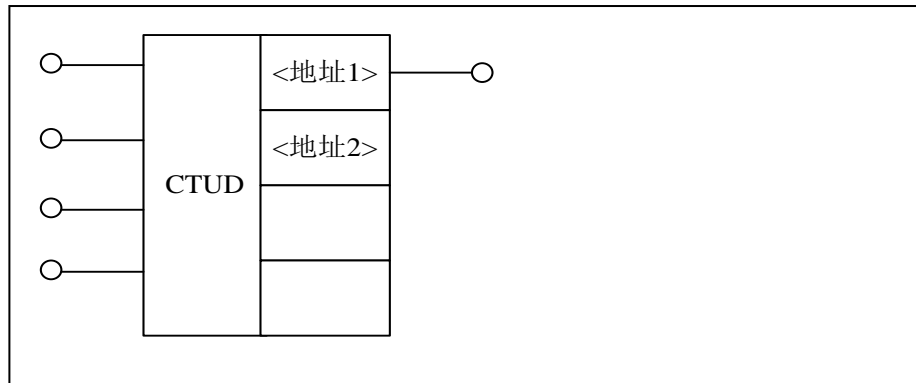
输入说明 输入 1：控制输入；
输入 2：复位输入。

示例

梯形图	
语句表	<pre>LD X2.5 LD X4.0 CTRC 0 100 OUT Y1.4</pre>
描述	当 X2.5 由通到断次数达到 100 次时，计数器导通。当 X4.0 导通时，计数器复位，输出到 Y1.4。

自定起始加减计数器 CTUD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
<地址 2>	□□□□	INT	常数、R、W、 D、B、P		后置 ✓

功能说明 自定义起始值的加减计数器。

参数说明 参数 1: 计数器编号;
参数 2: 计数器预置值。

输入说明 输入 1: 控制输入;
输入 2: 复位后的起始值, 导通为从 1 开始, 断开为从 0 开始。
输入 3: 加减输入, 导通为减计数, 断开为加计数。
输入 4: 复位输入。

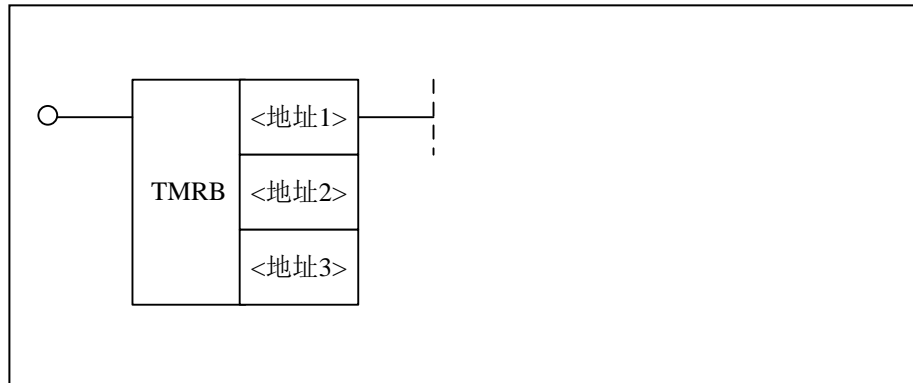
示例

梯 形 图	
语 句 表	<pre>LD X2.5 LD X2.6 LD X2.7 LD X2.8 CTUD 0 100 OUT Y1.4</pre>
描述	<p>0 号计数器当 X2.5 由通到断次数达到 100 次时，计数器导通输出到 Y1.4。当 X2.6 导通时，计数器复位后从 1 开始计数；否则从 0 开始计数。当 X2.7 断开时，计数器累加；否则累减。当 X2.8 导通时，计数器复位</p>

定时器

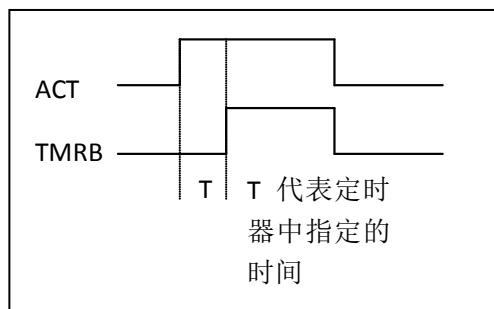
延时导通定时器 TMRB

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	常数、R、W、 D、P		

时序图



功能说明

延时导通定时器。

参数说明

参数 1: 定时器号;

参数 2: 时间单位, 以下为参数 2 的设置值解释;

(设置为 3 时, 时间单位为小时;

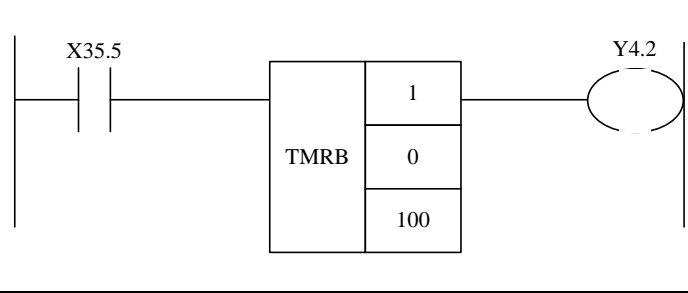
设置为 2 时, 时间单位为分钟;

设置为 1 时, 时间单位为秒;

设置为 0 时, 时间单位为毫秒。)

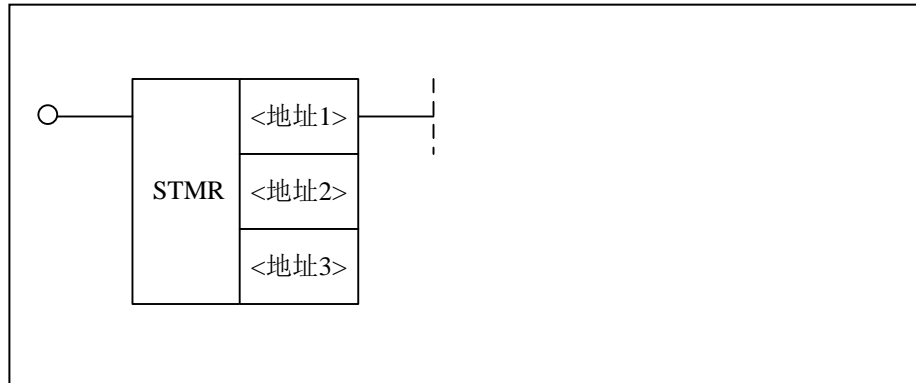
参数 3: 定时长度。

示例

梯形图	
语句表	<pre>LD X35.5 TMRB 1 0 100 OUT Y4.2</pre>
描述	当 X35.5 导通 100ms 后, 定时器 1 导通输出到 Y4.2

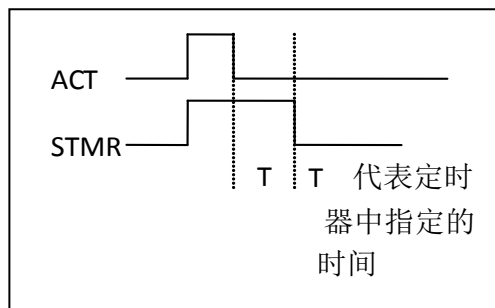
延时断开定时器 STMR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓ 后置 ○
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	常数、R、W、 D、P		

时序图



功能说明

延时断开定时器。

参数说明

参数 1: 定时器号;

参数 2: 时间单位, 以下为参数 2 的设置值解释;

(设置为 3 时, 时间单位为小时;

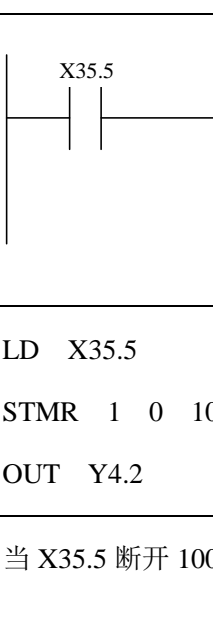
设置为 2 时, 时间单位为分钟;

设置为 1 时, 时间单位为秒;

设置为 0 时, 时间单位为毫秒。)

参数 3: 定时长度;

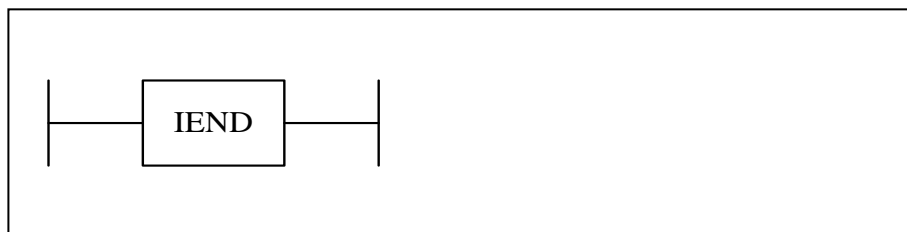
示例

梯形图	
语句表	<pre>LD X35.5 STMR 1 0 100 OUT Y4.2</pre>
描述	当 X35.5 断开 100ms 后, 定时器 1 断开, 切断 Y4.2 的输出。

流程控制

初始化模块结束 IEND

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 × 后置 ×

功能说明

定义初始化模块结束。一般将初始化模块放在程序的最前面，而且该模块只在系统上电以后执行一次。

示例

梯形图	
语句表	IEND
描述	初始化程序结束。

PLC1 模块结束 1END

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 × 后置 ×

功能说明

PLC1 模块结束。

示例

梯形图	
语句表	1END
描述	PLC1 程序结束。

PLC2 模块结束 2END

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 × 后置 ×

功能说明

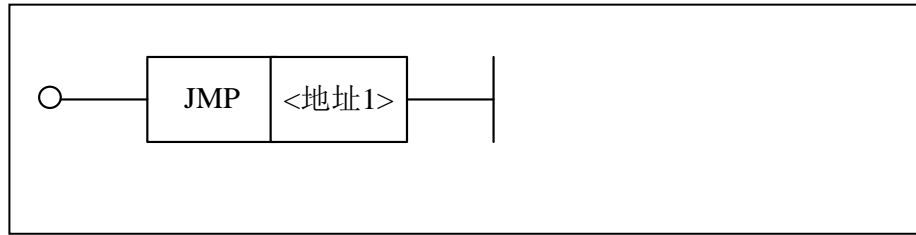
PLC2 执行模块结束。

示例

梯形图	
语句表	2END
描述	PLC2 程序结束。

跳转 JMP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	L		前置 √ 后置 ×

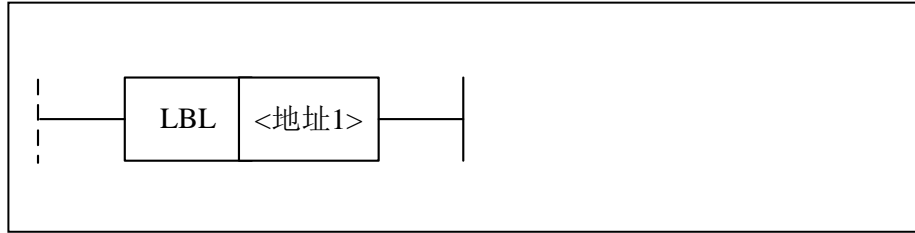
功能说明 按标记进行跳转。

示例

梯形图	
语句表	<pre>LD X35.5 JMP L1111</pre>
描述	如果 x35.5 导通，则跳转到标记为 L1111 的位置继续往下执行。

标号 LBL

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	L		前置 ○ 后置 ×

功能说明

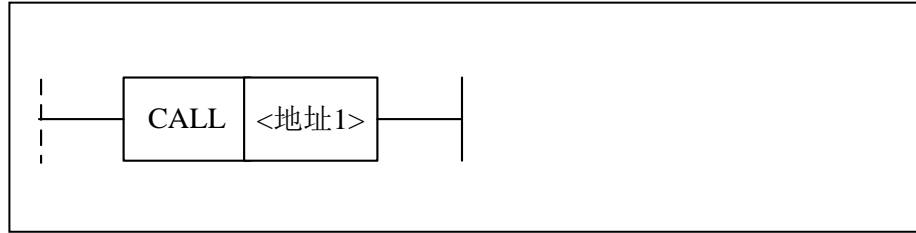
标号，按标号进行跳转，与 JMP 指令联合使用。

示例

梯形图	<p>The diagram shows a single horizontal line representing a ladder logic instruction. The instruction is a box divided into two parts: 'LBL' on the left and 'L1111' on the right. Vertical lines extend from the top and bottom of the box to the main horizontal line.</p>
语句表	LBL L1111
描述	设置标记 L1111。

调用子程序 CALL

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	S		前置 ○ 后置 ×

功能说明 调用子程序。

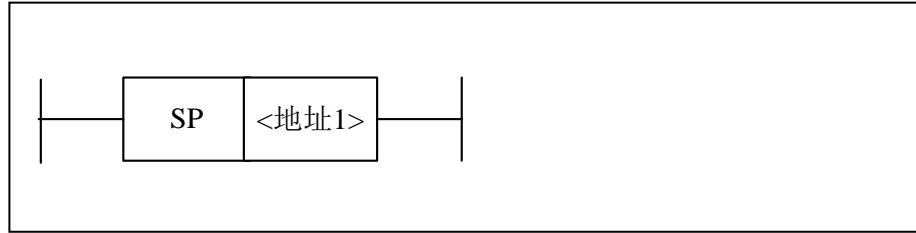
参数说明 子程序号。

示例

梯形图	<p>The diagram shows a vertical line on the left representing a power rail. A horizontal line connects this rail to a normally open contact labeled 'X12.2'. This contact is connected to a rectangular box labeled 'CALL' followed by 'S123'. The box is connected to another vertical line on the right, representing the output rail.</p>
语句表	<pre>LD X12.2 CALL S123</pre>
描述	当 X12.2 输入有效时，跳转到 S123 号子程序执行。

子程序开始 SP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	S		前置 × 后置 ×

功能说明 子程序开始。

参数说明 编号（最大支持 512 个子程序号）

示例

梯形图	
语句表	SP S111
描述	设置子程序号 S111。

子程序结束 SPE

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 × 后置 ×

功能说明 子程序结束。

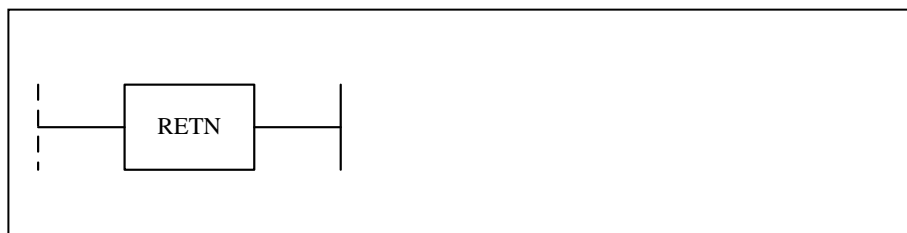
参数说明

示例

梯形图	
语句表	SPE
描述	子程序结束

子程序返回 RETN

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 ○ 后置 ×

功能说明

子程序返回。子程序中如果遇到该指令立即跳出子程序从而执行以下的内容。

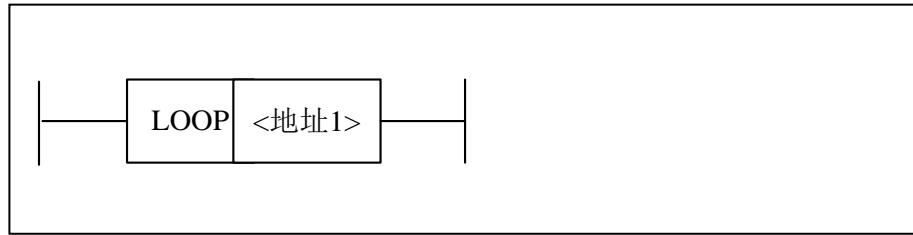
参数说明

示例

梯形图	
语句表	<pre>LDI R100.0 RETN</pre>
描述	R100.0 常闭点有效则子程序返回。

循环 LOOP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 × 后置 ×

功能说明

开始循环。将执行循环体内的语句，当达到循环次数后继续下面的语句。该指令必须与 NEXT 指令配合使用，在 LOOP 与 NEXT 之间的语句称为循环体。

参数说明

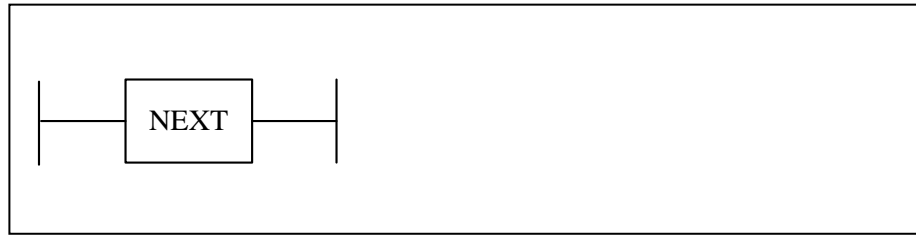
循环次数，可以使用常数和寄存器。

示例

梯形图	<p>The diagram shows a vertical line on the left and a vertical line on the right. A horizontal line connects them at the top, passing through a box labeled 'LOOP' with the number '5' to its right. Below this, there are three dots arranged vertically. At the bottom, another horizontal line connects the two vertical lines, passing through a box labeled 'NEXT'.</p>
语句表	<p>LOOP 5</p> <p>NEXT</p>
描述	<p>循环 5 次</p>

下一次循环 NEXT

格式



参数	参数格式	数据类型	存储区域	说明	属性
无	无	无	无		前置 × 后置 ×

功能说明 进入下一次循环。

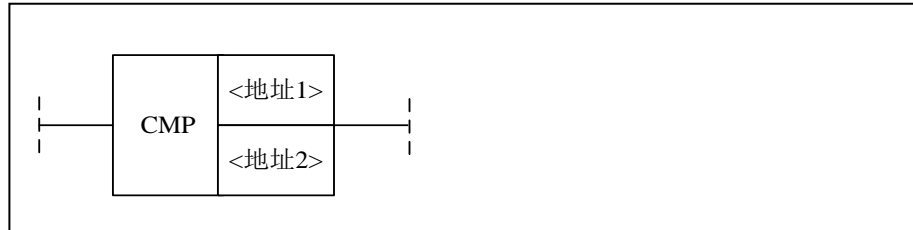
示例

梯形图	
语句表	NEXT
描述	进入下一次循环，与 LOOP 连用。

比较

比较 CMP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	大于时输出为 0，小于等于时输出为 1	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		

功能说明

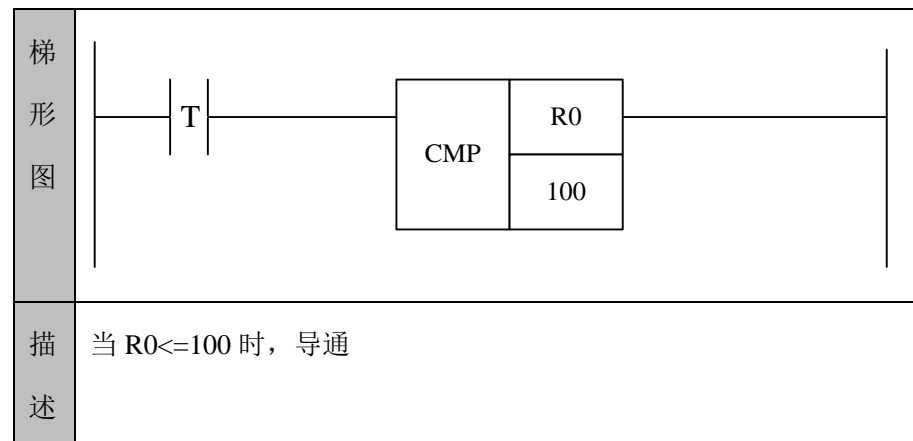
比较，大于时输出为 0，小于等于时输出为 1。

参数说明

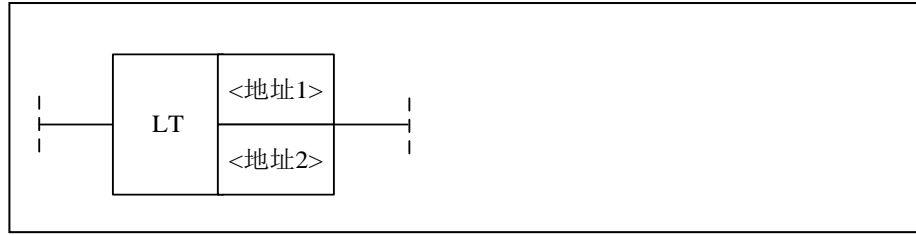
参数 1：比较数据，可以使用常数或寄存器；

参数 2：被比较数据，可以使用常数或寄存器

示例



小于 LT 格式

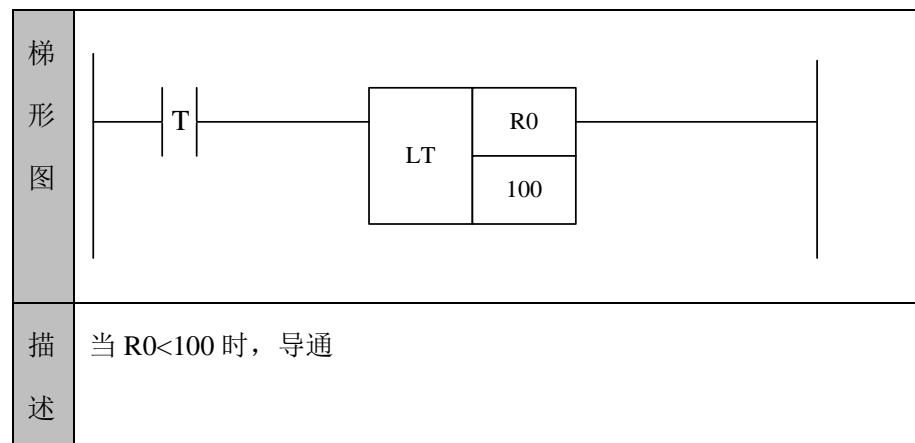


参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	大于等于时输出为 0，小于时输出为 1	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		

功能说明 比较，大于等于时输出为 0，小于时输出为 1。

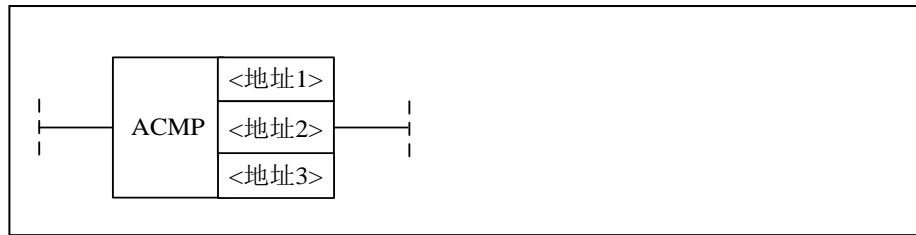
参数说明 参数 1：比较数据，可以使用常数或寄存器；
参数 2：被比较数据，可以使用常数或寄存器

示例



区域比较 ACMP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	地址 3 的数据大于地址 1, 小于地址 2 时, 输出为 1	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		
<地址 3>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		

功能说明

区域比较，地址 3 的数据大于地址 1，小于地址 2 时，输出为 1。

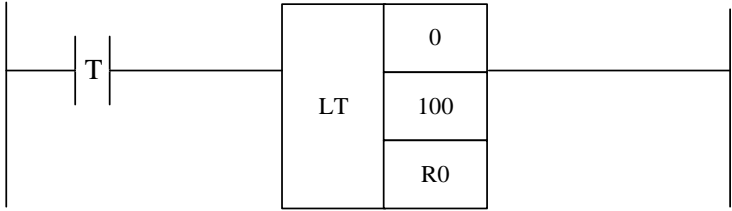
参数说明

参数 1: 比较范围下限，可以使用常数或寄存器；

参数 2: 比较范围上限，可以使用常数或寄存器

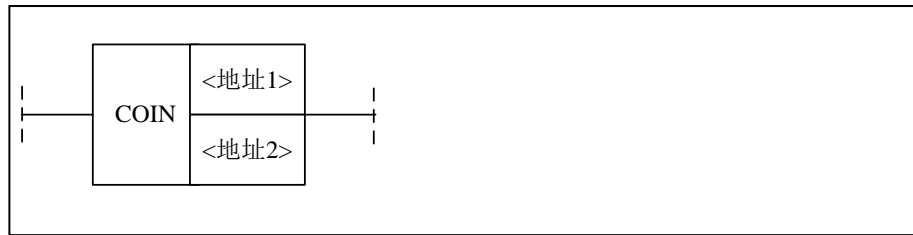
参数 2: 比较数据，可以使用常数或寄存器

示例

梯 形 图	 <p>The diagram shows a single ladder rung. On the left, there is a normally closed contact labeled 'T'. This contact is connected to a timer coil labeled 'LT'. The timer coil is represented by a rectangular box divided into three horizontal sections. The top section contains the value '0', the middle section contains '100', and the bottom section contains 'R0'. The rung ends with a vertical line on the right, representing the power rail.</p>
描 述	当 $R0 < 100$ 且 $R0 > 0$ 时，导通

一致性比较 COIN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	相等时输出为 1，不等时输出 0	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		

功能说明

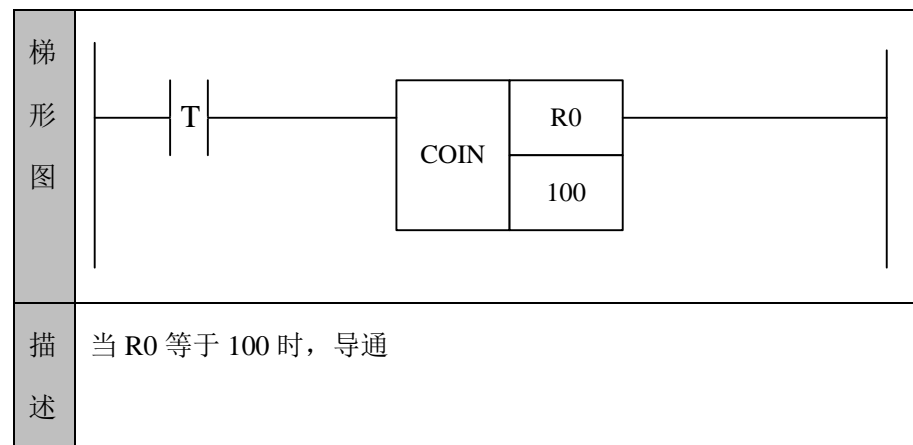
一致性比较，相等时输出为 1 不等时输出为 0

参数说明

参数 1：基准数据，可以使用常数或寄存器；

参数 2：比较数据，可以使用常数或寄存器；

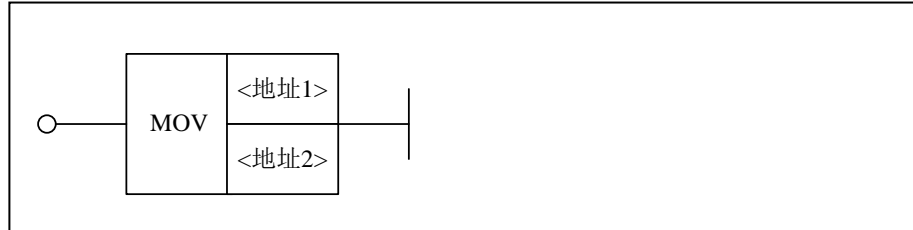
示例



数据操作

移动数据 MOV

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	移动数据	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	Y、G、R、W、D、B		

功能说明 移动数据，将源数据传送到目的地址中。

参数说明 参数 1：源数据，可以使用常数和寄存器。

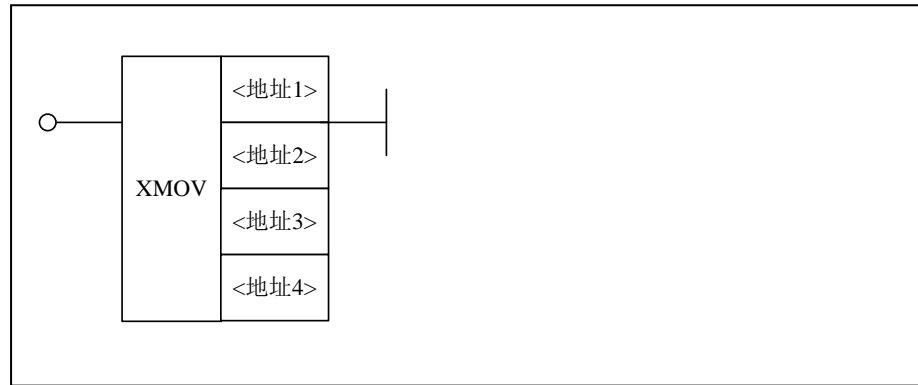
参数 2：目标地址，可以使用寄存器。

示例

梯形图	
描述	将 D0 中的数据赋给 D1

相对移动数据 XMOV

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	相对移动数据	前置 ○ 后置 ✓
<地址 2>	□□□□	INT	G、R、W、D、B		
<地址 3>	□□□□	INT	常数		
<地址 4>	□□□□	INT	G、R、W、D、B		

功能说明 移动数据，将源数据传送到目的地址中。

参数说明 参数 1：操作数 1 格式，0 代表寄存器，1 代表 B 寄存器，2 代表 P 寄存器；例如参数 1 为 0，参数 2 为 R10，代表的地址就是 R10；参数 1 为 1，参数 2 为 R10，代表的就是 B 寄存器，B 寄存器组号为 R10 存储的数据；参数 1 为 2，参数 2 为 R10，代表的就是 P 寄存器，P 寄存器组号为 R10 存储的数据；

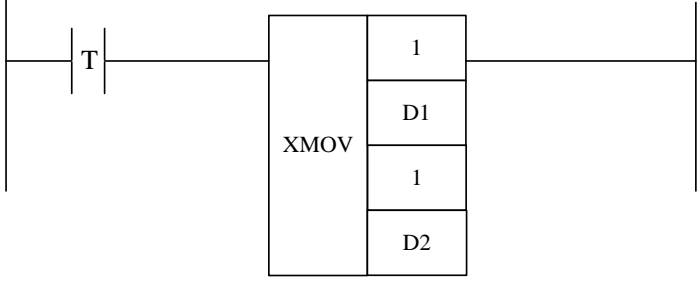
参数 2：操作数 1 地址；

参数 3：操作数 2 格式；

参数 4：操作数 2 地址。

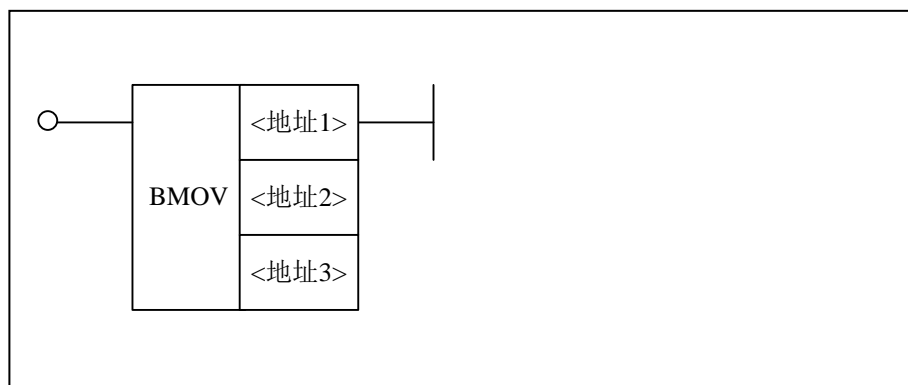
。

示例

梯形图	
描述	将 B 寄存器中偏移 D1 的数据赋给 B 寄存器中偏移 D2 的位置。

成批移动 BMOV

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B	成批移动数据	前置 ○
<地址 2>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		
<地址 3>	□□□□	INT	常数		后置 ✓

功能说明 批移动数据，将从源开始地址的多个数据传送到目的开始地址处。

参数说明 参数 1：源数据开始地址；

参数 2：目标开始地址；

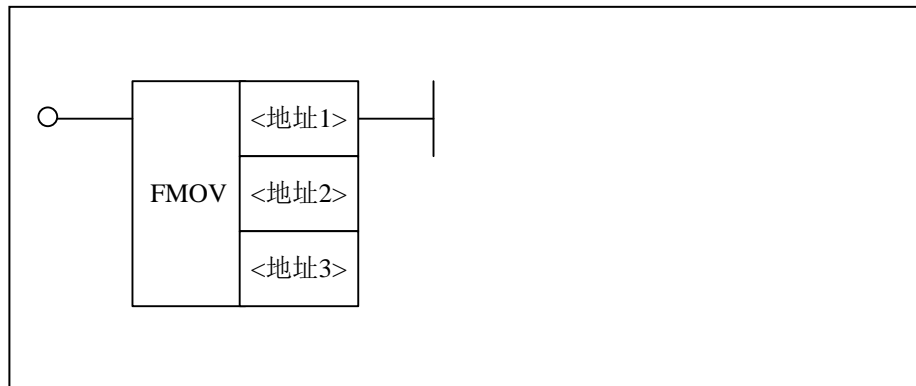
参数 3：移动个数，只能使用常数。

示例

梯 形 图	
描 述	<p>将从 D0 开始的 2 个数据赋值给 D2 开始的 2 个位置，即 D0 赋值给 D2，D1 赋值给 D3。</p>

多个移动 FMOV

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	Y、G、R、 W、D、B	多个移动 数据	前置 ○ 后置 ×
<地址 2>	□□□□	INT	Y、G、R、 W、D、B		
<地址 3>	□□□□	INT	常数、X、Y、 F、G、R、W、 D、P、B		

功能说明 多个移动数据，将源数据传送到从目的开始地址到目的结束地址这段空间。

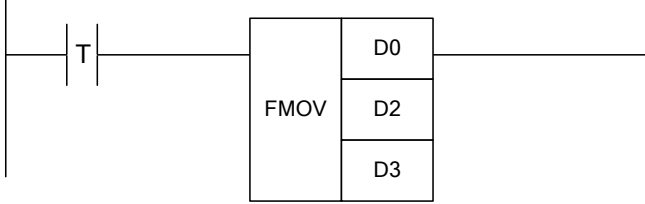
参数说明

参数 1：目标开始地

参数 2：目标结束地址；

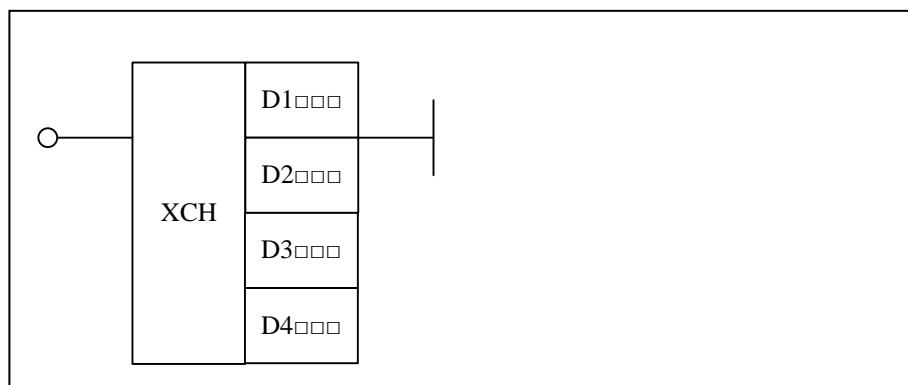
参数 3：源数据。

示例

梯 形 图	
描 述	将 D3 的数据赋值给从 D0 到 D2 的位置上。

数据交换 XCH

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	用于数据表的交换	前置 ○
<地址 2>	□□□□	INT	G、R、W、D、B		
<地址 3>	□□□□	INT	常数		后置 ×
<地址 4>	□□□□	INT	G、R、W、D、B		

功能说明

数据交换，将操作数 2 地址和操作数 4 地址的数据交换。操作数 2 的格式可以由地址 1 数值表示，0 表示地址 2 中所使用的默认寄存器类型，1 表示地址 2 中使用 B 寄存器，同理操作数 4 的格式由地址 3 的数值表示。

参数说明

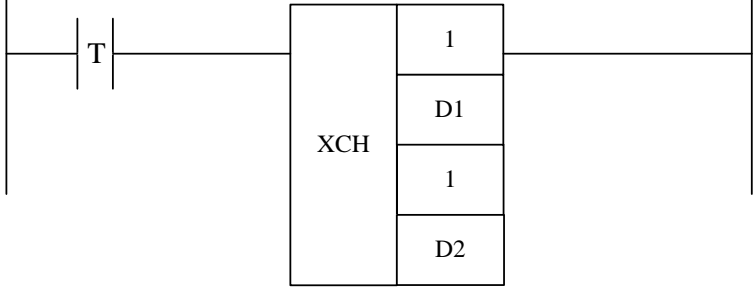
参数 1: 操作数 1 格式，0 代表寄存器，1 代表 B 寄存器，2 代表 P 寄存器；例如参数 1 为 0，参数 2 为 R10，代表的地址就是 R10；参数 1 为 1，参数 2 为 R10，代表的就是 B 寄存器，B 寄存器组号为 R10 存储的数据；参数 1 为 2，参数 2 为 R10，代表的就是 P 寄存器，P 寄存器组号为 R10 存储的数据；

参数 2: 操作数 1 地址；

参数 3: 操作数 2 格式；

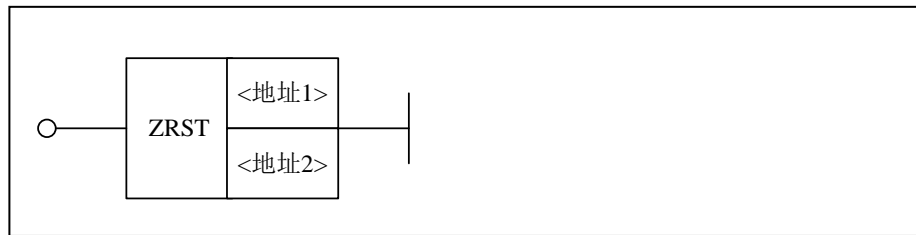
参数 4: 操作数 2 地址。

示例

梯形图	
描述	将 B 寄存器中偏移 D1 的数据与 B 寄存器中偏移 D2 的数据交换。

数据复位 ZRST

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□. □	BOOL	Y、G、R、W、 D、B	数据复位	前置 ○
<地址 2>	□□□. □	BOOL	Y、G、R、W、 D、B		后置 ✓

功能说明 数据复位，将从操作数起始地址到操作数结束地址的数据全部复位。

参数说明 参数 1：操作数起始地址；7

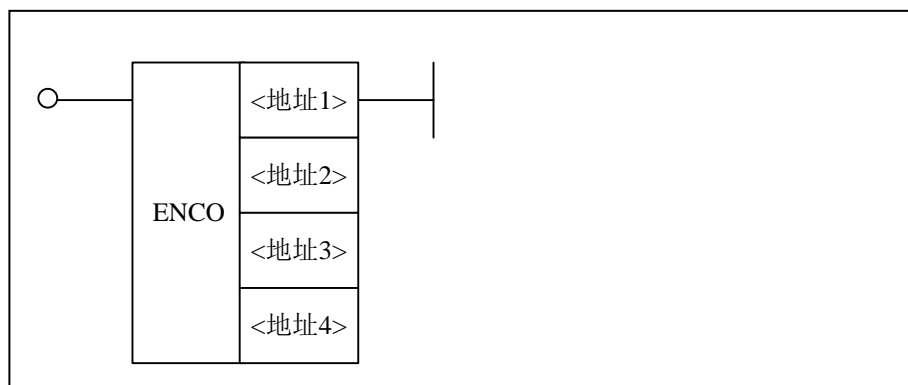
参数 2：操作数结束地址/。

示例

梯形图	
描述	将 0 赋值给 D0 到 D1 的位置。

编码 ENCO

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B	用于修 调值的 转换	前置 ○
<地址 2>	□□□□	INT	常数		后置 ×
<地址 3>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B		
<地址 4>	□□□□. □	BOOL	Y、G、R、W、D、 P、B		

功能说明

编码。当编码数据起始位置开始的 5 个数据位 3, 5, 7, 8, 9 时, 源数据为 3, 则输出 0000001B; 源数据为 5, 则输出 0000010B; 源数据为 7, 则输出 00000100B。

参数说明

参数 1: 编码数据的起始位置, 可以使用寄存器 D;

参数 2: 编码数据的个数, 可以使用常数;

参数 3: 源数据, 可以使用寄存器 R, D;

参数 4: 目标数据的输出地址, 可以使用寄存器 R, D。

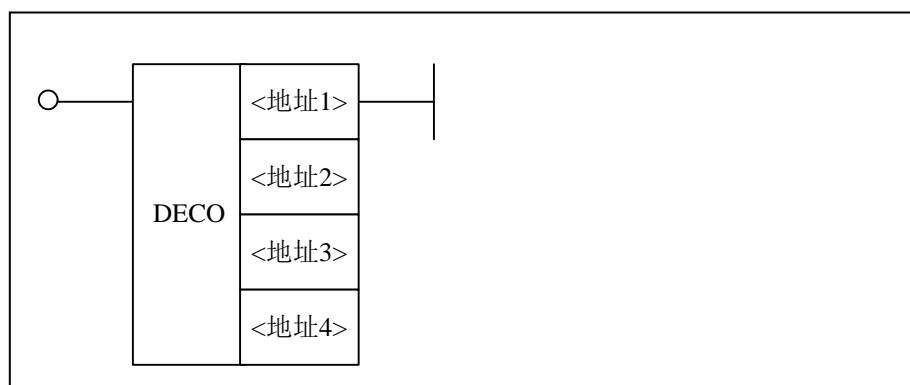
示例

梯 形 图	
描 述	将 D1 中的数据编码后输出到 D2 中。

译码 DECO

格式

参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B	用于修 调值的 转换	前置√
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B		
<地址 4>	□□□□. □	BOOL	Y、G、R、W、D、 P、B		后置×



功能说明 译码，是编码的反向操作。

参数说明

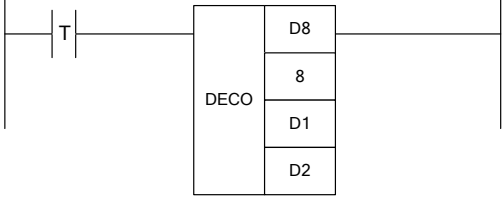
参数 1：译码数据的起始位置，可以使用寄存器 D。

参数 2：译码数据的个数，可以使用常数。

参数 3：源数据，可以使用寄存器 R，D。

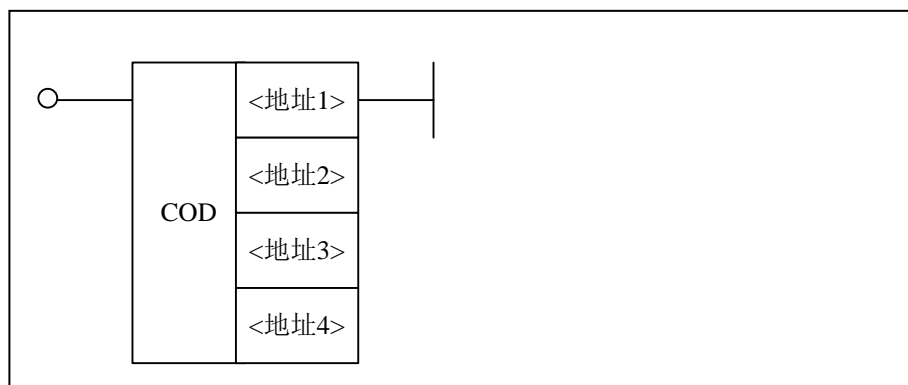
参数 4：目标数据的输出地址，可以使用寄存器 R，D。

示例

梯 形 图	
描 述	将 D1 中的数据译码后输出到 D2 中。

代码转换 COD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B	用于修 调值的 转换	前置 ✓
<地址 2>	□□□□	INT	常数		后置 ×
<地址 3>	□□□□. □	BOOL	X、Y、F、G、R、 W、D、P、B		
<地址 4>	□□□□. □	BOOL	Y、G、R、W、D、 P、B		

功能说明

代码转换，主要用于修调值的转换。以主轴修调为例，从 D0 开始的 8 个数据位 50、60、70、80、90、100、110、120，当源数据为 0，转换后的数据为 50；当源数据为 1，转换后的数据为 60；当源数据为 2，转换后的数据为 70。

参数说明

参数 1：转换数据的起始位置，可以使用寄存器 D；

参数 2：转换数据的个数，可以使用常数；

参数 3：源数据，可以使用寄存器 R，D；

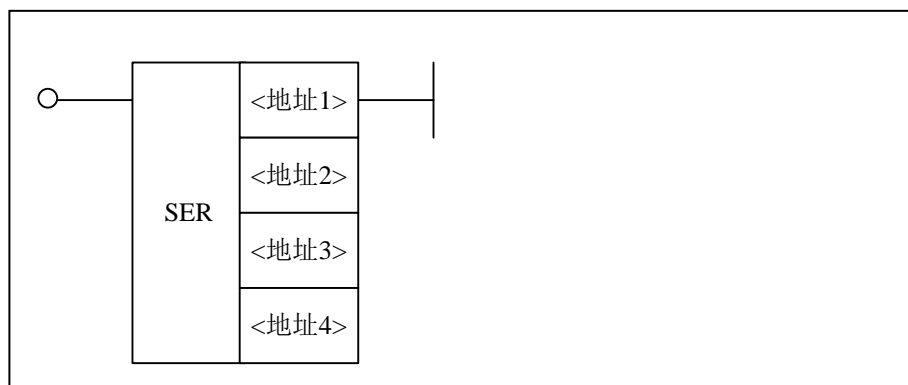
参数 4：目标数据的输出地址，可以使用寄存器 R，D。

示例

梯 形 图	
描 述	将 D1 中的数据进行代码转换，输出到 D2 中。

数据查找 SER

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	X、Y、F、G、R、 W、D、P、B	找到时输出 1,没找到输出 0	前置√
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	X、Y、F、G、R、 W、D、P、B		后置×
<地址 4>	□□□□	INT	Y、G、R、W、D、 P、B		

功能说明

进行数据查找。在数据表中的一段内查找某个数据，找到时输出 1，没找到输出 0。

参数说明

参数 1: 查找地址，只能使用 D 寄存器；

参数 2: 查找的范围，可以使用常数；

参数 3: 要查找的数据，常数和寄存器 X、Y、K、L、F、G、R、D；

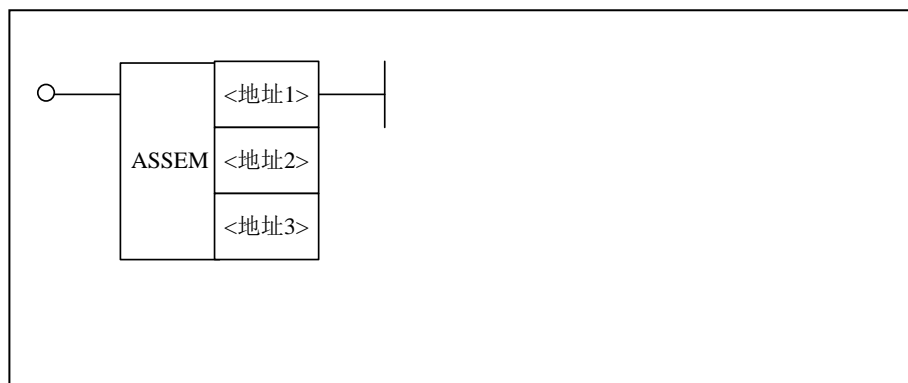
参数 4: 查找结果输出的地址，只能使用 D 寄存器。

示例

梯 形 图	
描 述	在 D0 开始的 4 个数据中查找 D4 中的数据, 将找到数据的位置输出到 D5 中。

寄存器合并 ASSEM

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	X、Y、F、G、R、 W、D、P、B	将几个寄存器的数据合并到一个寄存器中	前置√
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	G、W、D、B		后置×

功能说明 将几个寄存器的数据合并到一个寄存器中。

参数说明

参数 1: 源地址;

参数 2: 源寄存器数量, 只能使用常数;

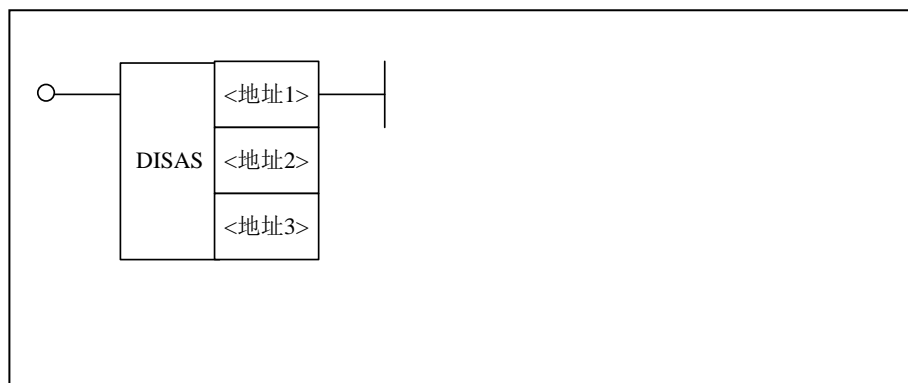
参数 3: 目标地址, 寄存器 G、W、D、B;

示例

梯 形 图	
描 述	<p>将从 X0 开始的 4 个数据合并成一个数据（即 4 个 8 位数据合并成一个 32 位数据），输出到 D4 中。</p>

寄存器分解 DISAS

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	F、G、W、D、P、B	将一个寄存器的数据分解到几个寄存器中	前置√
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	Y、G、R、W、		后置×

功能说明 将一个寄存器的数据分解到几个寄存器中。

参数说明

参数 1: 源地址;

参数 2: 源寄存器数量, 只能使用常数;

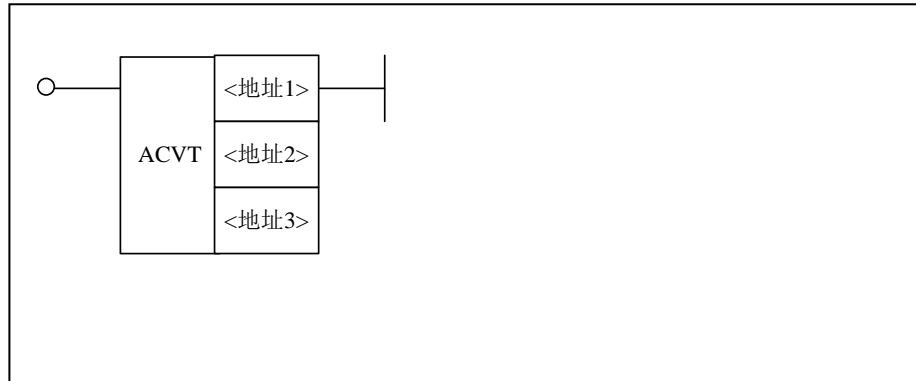
参数 3: 目标地址, 寄存器 Y、G、R、W;

示例

梯 形 图	
描 述	<p>将 D0 的数据分解到从 Y0 开始的 4 个数据中（即 1 个 32 位数据分解成 4 个 8 位数据）。</p>

区域转换 ACVT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	P	将源数据按一定比例关系转换到目标数据中	前置√
<地址 2>	□□□□	INT	X、Y、F、G、R、W、D、P、B		
<地址 3>	□□□□	INT	Y、G、R、W、D、B		后置×

功能说明

将源数据按一定比例关系转换到目标数据中。

参数说明

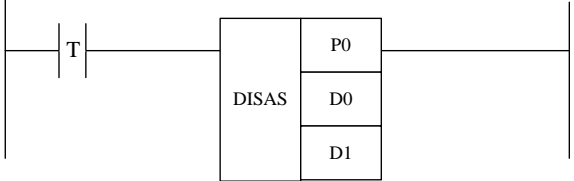
参数 1：比例关系地址；

0	源数据最小值
1	源数据最大值
2	目标数据最小值
3	目标数据最大值

参数 2：源寄存器数量；

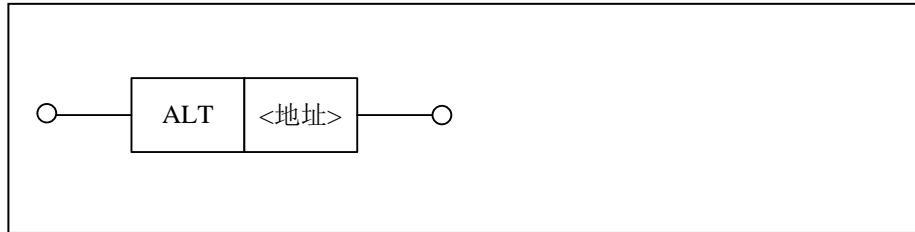
参数 3：目标地址，寄存器 Y、G、R、W、D、B；

示例

梯形图	
描述	将 D0 的数据按一定比例关系转换到 D1 中。 $D1 = (D0 - P0) * (P3 - P2) / (P1 - P0) + P0;$

交替输出 ALT

格式

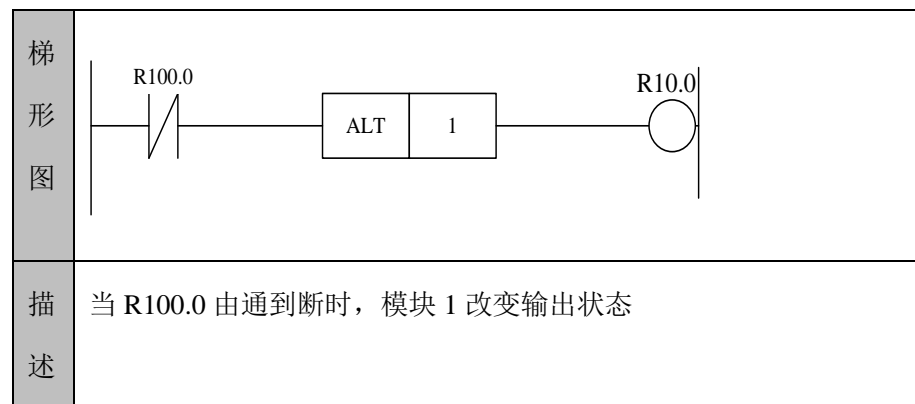


参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	编号	前置 ✓
					后置 ✓

功能说明

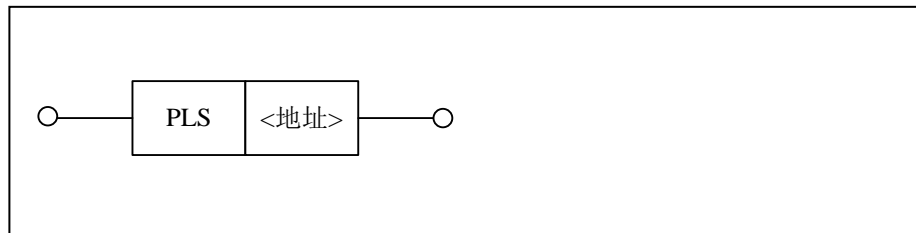
交替输出。此元件的输出状态将一直保持，直到遇到上升沿时，输出状态将改变（由 0 到 1，或由 1 到 0）。

示例



取上升沿 PLS

格式



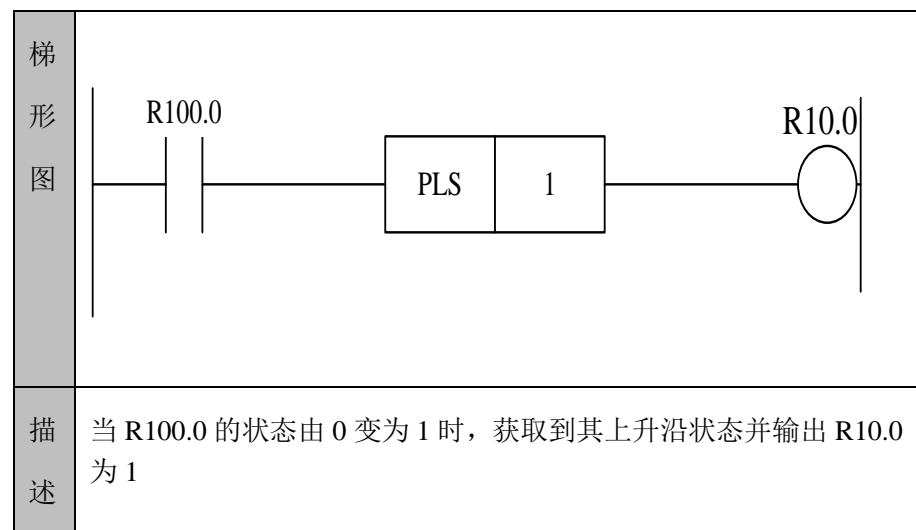
参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	上升沿模块号	前置 ○ 后置 ✓

功能说明

得到当前行当前位置的状态并获取其上升沿触发信号。

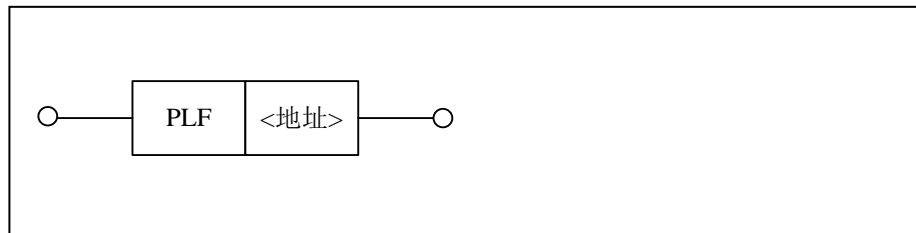
在信号上升沿的当前扫描周期中将输入信号设置为 1。(请注意此处功能与基本元件中上升沿触发元件的区别)适合任何需要检测上升沿状态的场合

示例



取下降沿 PLF

格式



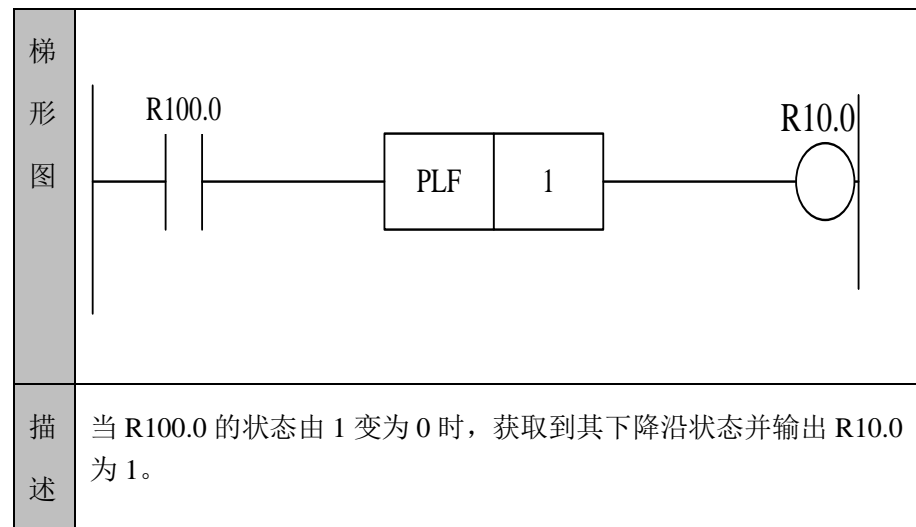
参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	下降沿模块号	前置 ○ 后置 ✓

功能说明

得到当前行当前位置的状态并获取其下降沿触发信号。

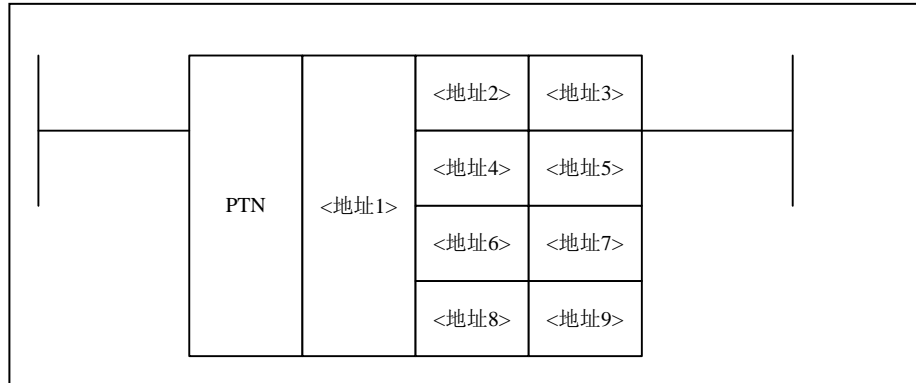
在信号下降沿的当前扫描周期中将输入信号设置为 1。（请注意此处功能与基本元件中下降沿触发元件的区别）适合任何需要检测下降沿状态的场合。

示例



点数转换 PTN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	BOOL	Y、G、R、 W、D、B、	点有效产生对应的数	前置 ○
<地址 2>	□□□□. □	BOOL	X、Y、F、G、 R、W、D、 P、T、C、B		
<地址 3>	□□□□	INT	常数		
<地址 4>	□□□□. □	BOOL	X、Y、F、G、 R、W、D、 P、T、C、B		
<地址 5>	□□□□	INT	常数		后置 ×
<地址 6>	□□□□. □	BOOL	X、Y、F、G、 R、W、D、 P、T、C、B		
<地址 7>	□□□□	INT	常数		
<地址 8>	□□□□. □	BOOL	X、Y、F、G、 R、W、D、 P、T、C、B		
<地址 9>	□□□□	INT	常数		

功能说明 建立点和数字的对应关系，当点有效时，产生对应的数。

参数说明

参数 1: 目的地址;

参数 2: 点 1;

参数 3: 数字 1;

参数 4: 点 2;

参数 5: 数字 2;

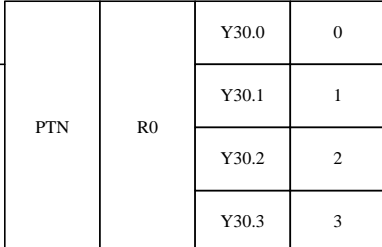
参数 6: 点 3;

参数 7: 数字 3;

参数 8: 点 4;

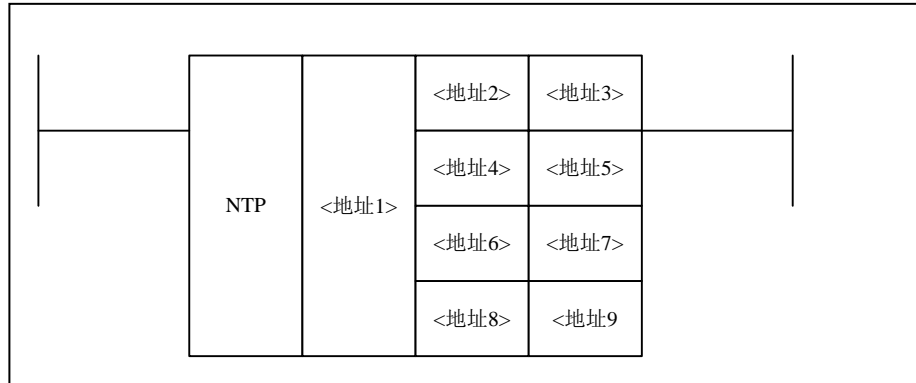
参数 9: 数字 4。

示例

梯 形 图				
	描	Y30.0 有效时，R0=0。		
	述	Y30.1 有效时，R0=1。		
		Y30.2 有效时，R0=2。		
		Y30.3 有效时，R0=3		

数点转换 NTP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	BOOL	Y、G、R、W、D、B		前置 ○
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□. □	BOOL	X、Y、F、G、R、W、D、P、T、C、B		
<地址 4>	□□□□	INT	常数		
<地址 5>	□□□□. □	BOOL	X、Y、F、G、R、W、D、P、T、C、B		后置 ×
<地址 6>	□□□□	INT	常数		
<地址 7>	□□□□. □	BOOL	X、Y、F、G、R、W、D、P、T、C、B		
<地址 8>	□□□□	INT	常数		
<地址 9>	□□□□. □	BOOL	X、Y、F、G、R、W、D、P、T、C、B		

功能说明

建立数字和点的对应关系，根据参数 1 中的数字产生对应的点信号。

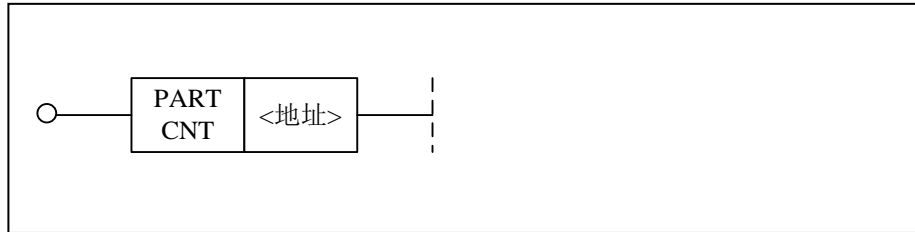
- 参数说明**
- 参数 1: 源数据地址;
 - 参数 2: 数字 1;
 - 参数 3: 点 1;
 - 参数 4: 数字 2;
 - 参数 5: 点 2;
 - 参数 6: 数字 3;
 - 参数 7: 点 3;
 - 参数 8: 数字 4;
 - 参数 9: 点 4。

示例

梯 形 图				
	描	R0=0 时, Y30.0 有效		
	述	R0=1 时, Y30.1 有效		
		R0=2 时, Y30.2 有效		
		R0=3 时, Y30.3 有效		

计件 PARTCNT

格式

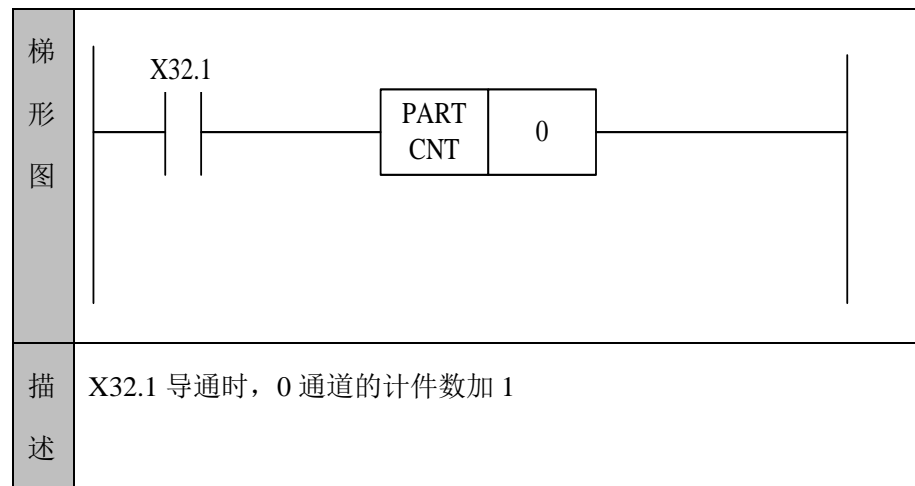


参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	导通时，<地址 1>通道的计件数加 1	前置 ○ 后置 ✓

功能说明 加工计件。

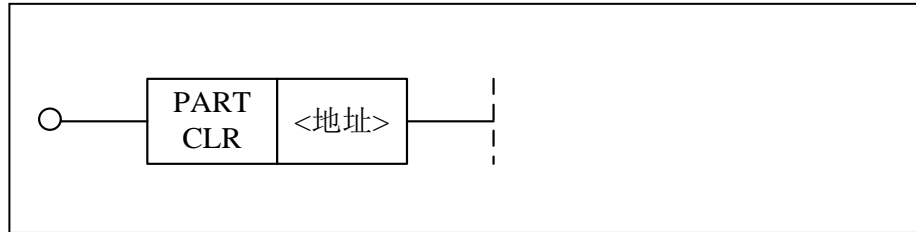
参数说明 参数 1：通道号。

示例



计件清除 PARTCLR

格式

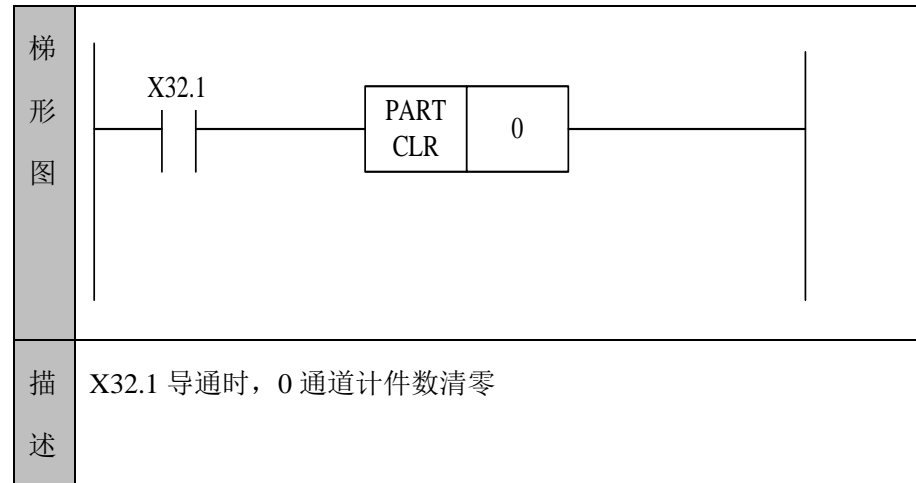


参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	导通时，<地址>通道计件数清零	前置 ○ 后置 ✓

功能说明 计件数清空。

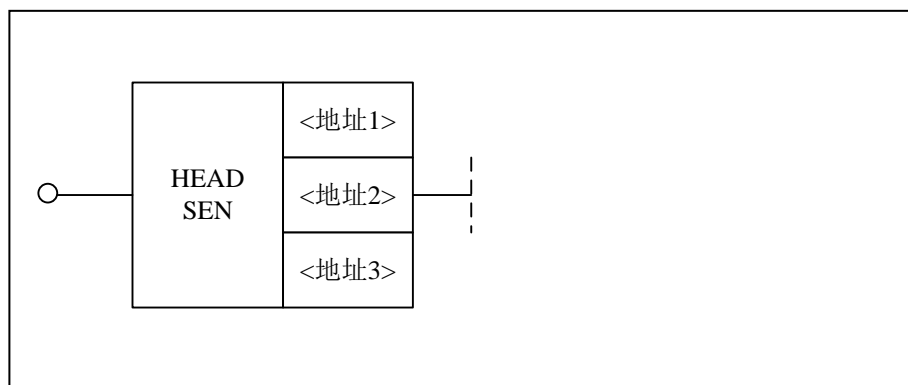
参数说明 参数 1：通道号。

示例



温度采集模块 HEADSEN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□. □	BOOL	X、Y、F、G、 R、W、D、 P、T、C、B	<地址 2> 为 0 时， 温度采集 模块开始 计数，从< 地址 3>中 给定的起 始位置存 储<地址 1>中温度 数据	前置 ○
<地址 2>	□□□□	INT	常数		后置 ×
<地址 3>	□□□□	INT	常数		

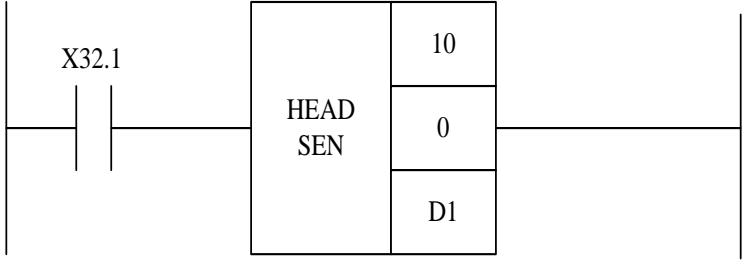
功能说明 温度采集模块。

参数说明 参数 1：温度采集的总数量，可以使用常数；

参数 2：温度采集模块使能开关，0 表示开始计数，其他值表示断开；

参数 3：温度采集数据存储的起始位置，可以使用寄存器 D。

示例

梯形图	 <p>The diagram shows a single step in a ladder logic program. It starts with a normally open contact labeled 'X32.1'. This contact is connected to a coil labeled 'HEAD SEN'. The coil has three parameters: '10' in the top section, '0' in the middle section, and 'D1' in the bottom section. The step ends with a vertical line on the right.</p>
描述	<p>X32.1 导通时，温度采集模块开始计数，从 D1 给定的起始位置存储 10 个温度数据。</p>

状态字和控制字编程

本章主要内容

5.1 状态字和控制字介绍

5.2 状态字和控制字编程实例

状态字和控制字介绍

概述




状态字和控制字是数控系统 CNC 与 PLC 之间交互的最直接的方式,通过获取系统的状态字可以得到系统运行中的状态数据,也可以通过写入控制字来改变系统的工作状态。在 HNC8 型系统中,状态字用 F 表示,其属性为只读,控制字用 G 表示,其属性为可读可写。

但为了限制某些关键系统功能的使用,某些控制字是被限制使用或对用户不可见的,请用户仔细阅读以下对状态字和控制字的限制条件。

状态字和

控制字的

使用限制

	表示可以使用
	表示保留使用供以后扩展
	禁止用户使用

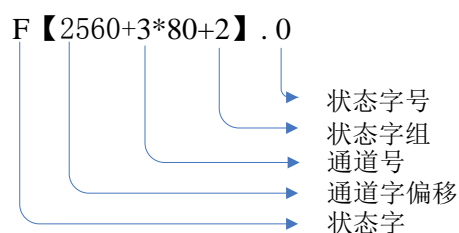
使用范围

状态字和控制字都可以按照在系统中起到的功能分为 3 类,根据系统型号的不同,它们都有各自的使用范围,具体可参见配置说明书:

- ◆ 轴状态字与控制字
- ◆ 通道状态字与控制字
- ◆ 系统状态字与控制字

符号格式

以通道类型字格式为例:



该示例表示的是通道 3, 状态字第 2 组, 0 号状态字的格式。其中 2560 是通道状态字的偏移量。其他类型的字格式与此类似。

轴状态字

概述 每轴配置状态字 80 个。每个字均有 16 位字节，第一行表示 0~7 位，第二行表示 8~15 位。使用轴状态字时，需要加上轴逻辑编号偏置。

轴状态字

D7 D6 D5 D4 D3 D2 D1 D0
D15 D14 D13 D12 D11 D10 D9 D8

F0	从轴跟随	从轴零点	从轴回零	回零完成	回零失败	回零第 2	回零第 1	轴移动
	轴复位	轴锁住	轴参数 ok	轴过载	第四参考	第三参考	第二参考	第一参考

F1	SPD 到达	主轴零速	定向完成	快移进给	保留	保留	主轴模式	PMC 使能
	分度轴锁	分度位置	分度轴解锁	保留	保留	保留	保留	保留

F2	伺服参数	零位捕获	保留	伺服回零	2Enc 零	保留	伺服准备好	捕获首 Z
	主轴零速	SPD 到达	增益切换	捕获 Z 脉	力矩控制	速度控制	位置控制	sv 准备好

F3	保留	保留	保留	保留	保留	伺服提示	伺服报警	伺服正常
	保留	保留	保留	保留	保留	保留	保留	定向完成

详细说明

- 【F0.0】轴移动中：轴在移动时为 1，轴未移动时为 0。
- 【F0.1】回零第一步：轴回零还未碰到回零挡块时为 1，否则为 0。
- 【F0.2】回零第二步：在找 Z 脉冲时为 1，否则为 0。
- 【F0.3】回零不成功：轴回零没有完成时，为 1，否则为 0。
- 【F0.4】回零成功：轴回零完成时，为 1，否则为 0。
- 【F0.5】从轴回零中。
- 【F0.6】从轴零点检查完成。
- 【F0.7】从轴跟随状态已经解除。
- 【F0.8】第一参考点确认：轴在第一参考点时为 1，否则为 0。
- 【F0.9】第二参考点确认：轴在第二参考点时为 1，否则为 0。
- 【F0.10】第三参考点确认：轴在第三参考点时为 1，否则为 0。
- 【F0.11】第四参考点确认：轴在第四参考点时为 1，否则为 0。
- 【F0.13】轴参数生效。
- 【F0.14】轴已经锁住。
- 【F0.15】轴已经重新定位。
- 【F1.0】PMC 控制使能，已使能时为 1，否则为 0。
- 【F1.1】进给主轴模式，1 为位置模式，0 为速度模式。
- 【F1.5】进给主轴定向完成。

- 【F1.6】进给主轴零速。
- 【F1.7】进给主轴速度到达。
- 【F1.13】分度轴解锁，1为系统通知plc分度轴解锁，上使能。
- 【F1.14】分度轴在分度位置。
- 【F1.15】分度轴锁住，1为系统通知plc分度轴锁住，断开使能。
- 【F2.0】捕获到回零一次Z脉冲信号，此时为1，否则为0。
- 【F2.1】伺服准备好标志为0时，伺服才能接受增量数据。
- 【F2.3】第二编码器Z脉冲捕获，主要用于距离码光栅尺回零。
- 【F2.4】伺服回零完成时为1。
- 【F2.6】零位捕获，用于主轴。主轴旋转碰到第一个Z脉冲时，设置为1。CS切换时，需要此位为1。
- 【F2.7】伺服参数切换状态，0：默认参数 1：已切换为第2套伺服参数。
- 【F2.8】总线伺服准备好时为1，否则为0。
- 【F2.9】伺服为位置控制模式时为1，否则为0。
- 【F2.10】伺服为位置速度模式时为1，否则为0。
- 【F2.11】伺服为力矩控制模式时为1，否则为0。
- 【F2.12】碰到Z脉冲为1，否则为0
- 【F2.13】
- 【F2.14】主轴速度到达时为1，否则为0。
- 【F2.15】主轴零速：当主轴停止时为1，否则为0。
- 【F3.0】伺服正常时为1。
- 【F3.1】伺服报警时为1。
- 【F3.2】伺服提示时为1。
- 【F3.8】主轴定向完成：当设置主轴定向后，主轴开始定向，完成后，伺服返回主轴定向完成信号，此时为1，否则为0。
- 【F4】轴所属通道号。（通道号为十进制存储）
- 【F5】引导的从轴个数。（从轴个数为十进制存储）
- 【F[6/7]】实时输出的指令增量，电机坐标。
- 【F[8/9/10/11]】实时输出的指令位置，电机坐标。（米度单位）
- 【F[12/13/14/15]】输出指令脉冲位置，脉冲单位。
- 【F[16/17]】每个周期的指令脉冲。每个周期向伺服发送的指令脉冲数
- 【F[18/19]】输出指令力矩。
- 【F[20/21/22/23]】1号编码器反馈实际位置，米度单位。
- 【F[24/25/26/27]】2号编码器反馈实际位置，米度单位。
- 【F[28/29/30/31]】机床指令位置，米度单位。
- 【F[32/33/34/35]】机床实际位置，米度单位。
- 【F[36/37]】轴报警。
- 【F36.2】压正限位挡块。
- 【F36.3】压负限位挡块。
- 【F36.4】实际速度超速。
- 【F36.6】超速。
- 【F36.7】超加速。
- 【F36.8】找不到Z脉冲。
- 【F36.9】失去联接。

- 【F36.10】未回参考点。
- 【F36.11】同步位置超差。
- 【F36.12】从轴零点检查失败。
- 【F36.13】同步速度超差
- 【F37.0】已超出行程正限位。
- 【F37.2】已超出行程负限位。
- 【F37.2】加速度和最高速不匹配。
- 【F[38/39]】轴提示。
- 【F38.0】最大补偿率超出。
- 【F38.1】最大补偿值超出。
- 【F38.2】零点偏置参数过小。
- 【F38.4】软限位值太大。
- 【F38.5】第2软限位值太大。
- 【F38.6】绝对值编码器循环位数不合法。
- 【F38.7】位置溢出。
- 【F38.8】目标点在正限位外。
- 【F38.9】目标点在负限位外。
- 【F38.10】需要调整Z脉掩码角度。
- 【F38.11】需要调整参考点位置。
- 【F38.12】跟踪误差过大。
- 【F[70]】轴当前模式。

轴控制字

概述

每轴配置控制字 80 个。每个字均有 16 位字节，第一行表示 0~7 位，第二行表示 8~15 位。使用轴控制字时，需要加上轴逻辑编号偏置。

轴控制字

D7	D6	D5	D4	D3	D2	D1	D0
D15	D14	D13	D12	D11	D10	D9	D8

G0

轴使能	轴锁住	回零挡块	回零开始	负向禁止	正向禁止	负限位	正限位
轴复位	补偿扩展	保留	保留	从轴跟随	保留	保留	保留

G1

*SP 反转	*SP 正转	*SP 定向	*SP 点动	扩展软限	第 2 软限	pmc 相移	pmc 绝移
锁住应答	解锁应答	保留	CS 应答	保留	保留	保留	保留

G2

伺服参数	保留	保留	保留	获 2 码盘 Z	保留	保留	捕获 Zpuls
主轴限流	定向换挡	保留	主轴定向	力矩控制	速度控制	位置控制	伺服增益

G3

保留	保留	保留	保留	保留	保留	保留	伺服使能
保留	保留	保留	保留	保留	保留	保留	保留

详细说明

- 【G0.0】轴的正限位开关。
- 【G0.1】轴的负限位开关。
- 【G0.2】轴正向移动禁止。
- 【G0.3】轴负向移动禁止。
- 【G0.4】设置回零开始。
- 【G0.5】设置回零挡块。
- 【G0.6】设置轴锁住。
- 【G0.7】设置轴使能。
- 【G0.11】设置解除从轴跟随功能。
- 【G0.14】补偿扩展
- 【G0.15】单轴复位。
- 【G1.0】PMC 轴绝对移动使能。
- 【G1.1】PMC 轴相对移动使能。
- 【G1.2】第二软限位使能。

- 【G1.3】扩展软限位使能。
- 【G1.4】进给主轴点动
- 【G1.5】进给主轴定向。
- 【G1.6】进给主轴正转。
- 【G1.7】进给主轴反转。
- 【G1.12】PLC 对主轴 C/S 切换的应答标记。
- 【G1.14】PLC 对分度轴解锁信号的应答标记。
- 【G1.15】PLC 对分度轴锁住信号的应答标记。
- 【G2.0】Z 脉冲标志。(当电机在 Z 脉冲位置时, 此标记为 1)。
- 【G2.1】等待零脉冲。
- 【G2.2】关闭寻找零脉冲功能。
- 【G2.3】捕获第二编码器的零脉冲。
- 【G2.7】伺服参数切换 0: 默认参数 1: 切换为第 2 套参数
- 【G2.8】伺服增益切换
- 【G2.9】切换到位置控制模式
- 【G2.10】切换到速度控制模式。
- 【G2.11】切换到力矩控制模式。
- 【G2.12】主轴定向开始。
- 【G2.14】主轴定向换挡
- 【G2.15】主轴限流
- 【G3.0】伺服使能开关。
- 【G4】轴点动标记, 轴手动, 回零, 主轴旋转时, 此标记有效。
- 【G5】轴增量标记, 轴增量移动时, 此标记有效。
- 【G[6/7]】点动速度, 0,停止;1,参数中的手动速度;2 参数中的快移动速度;>2 自定义的速度。
- 【G8】增量倍率。
- 【G9】手摇倍率。
- 【G[10/11]】手摇脉冲
- 【G[12/13/14/15]】轴反馈位置, 脉冲单位。
- 【G[16/17/18/19]】轴反馈位置 2, 脉冲单位。

【G[20/21]】轴的实际速度，脉冲单位。轴的实际速度，为轴实际反馈位置（G12 - G15）每个周期的增量值。

【G[22/23]】轴的实际速度 2

【G[24/25]】轴的实际力矩

【G[26/27]】跟踪误差。（轴的跟踪误差，为轴的实际反馈位置（G12-G15）和轴的指令位置（F12-F15）之间的差值。）

【G[28/29/30/31]】编码器 1 的计数器值。

【G[32/33/34/35]】编码器 2 的计数器值。

【G[36/37]】实时补偿值。

【G[38/39]】采样时间戳。

【G[40/41/42/43]】锁存位置 1(当第一编码器有 Z 脉冲时，锁存的当前位置，用于 G31 或距离码回零。)

【G[44/45/46/47]】锁存位置 2(当第二编码器有 Z 脉冲时，锁存的当前位置，用于 G31 或距离码回零。)

【G[48/49/50/51]】PMC 轴绝对移动目标位置。

【G[52/53/54/55]】PMC 轴增量移动量。

【G[56/57]】伺服报警号。

【G[58/59]】伺服提示号。

【G60】轴控制模式切换。(2 为手轮中断，103 为 PMC 模式)

【G61】PMC 轴修调值。

【G62.0】PMC 轴停止移动

【G62.1】手轮中断量清零

【G62.2】开启切线跟随功能

【G62.4】分度轴开关

【G62.5】从轴耦合恢复时，同步同步轴位置

【G62.8】主轴控制，将实际转速写到指令中

【G62.9】启动主轴换挡转速

【G64】轴当前档位

【G66/67】主轴换挡使用

【G68/69】Z 脉冲位置

【G70/71】Z 脉冲间距 1

【G72/73】 Z 脉冲间距 2

【G74】 主轴换挡使用

【G78/79】 伺服采样数据

通道状态字

概述 每通道配置状态字 80 个。每个字均有 16 位字节，第一行表示 0~7 位，第二行表示 8~15 位。使用通道状态字时，需要加上通道编号偏置

轴状态字

D7 D6 D5 D4 D3 D2 D1 D0
D15 D14 D13 D12 D11 D10 D9 D8

F2560	用户干预	非自动时运动	循环启动	进给保持	模式#3	模式#2	模式#1	模式#0
	找 Z 脉冲	复位中	暂停请求	复位标志	校验状态	保留	车螺纹	正在切削

F2561	保留	保留	等待完成	中断跳过	中断完成	程序完成	程序启动	程序选中
	保留	保留	保留	保留	保留	保留	非空完成	非空指令

F2562	保留	保留	保留	保留	保留	保留	保留	保留
	4S 指令	3S 指令	2S 指令	1S 指令	主轴恒线	分度指令	刀偏标记	保留

详细说明

【F2560.0 ~F2560.3】获取模式。

- 0: 复位模式 1: 自动模式 2: 手动模式
- 3: 增量模式 4: 手摇模式 5: 回零模式
- 6: PMC 模式 7: 单段模式 8: MDI 模式

【F2560.4】进给保持：通道处于进给保持状态。

【F2560.5】循环启动：通道处于循环启动状态。

【F2560.6】非自动时有运动

【F2560.7】有用户运动干预中。

【F2560.8】正在切削

【F2560.9】螺纹切削：通道处于螺纹切屑状态，不允许进给保持。

【F2560.11】校验状态

【F2560.12】通道复位：当通道复位或按下面板上复位按键时，通道复位有效，直到设置通道复位应答。

【F2560.13】暂停请求

【F2560.14】通道复位中。

【F2560.15】通道内有轴回零找 Z 脉冲，禁止切换模式。

- 【F2561.0】程序选中，由解释器置位。
- 【F2561.1】程序启动，由通道控制置位。
- 【F2561.2】程序完成，由通道控制置位。
- 【F2561.3】G28/G31 等中断指令完成。
- 【F2561.4】中断指令跳过。
- 【F2561.5】等待指令完成。
- 【F2561.8】通道有非空指令标志
- 【F2561.9】通道完成非空指令标志
- 【F2562.9】刀偏标记[T 中含刀偏号]。
- 【F2562.10】PLC 分度指令标记。
- 【F2562.11】主轴恒线速。
- 【F2562.12】第 1 个 S 指令
- 【F2562.13】第 2 个 S 指令
- 【F2562.14】第 3 个 S 指令
- 【F2562.15】第 4 个 S 指令
- 【F2569】刀偏号。T 指令中含有的刀偏号
- 【F[2570/2571]】第 1 个 S 指令，单位 0.001 转/分。
- 【F[2572/2573]】第 2 个 S 指令，单位 0.001 转/分。
- 【F[2574/2575]】第 3 个 S 指令，单位 0.001 转/分。
- 【F[2576/2577]】第 4 个 S 指令，单位 0.001 转/分。
- 【F2578/79】当前等待信号的 G31 编号。
- 【F2580】当前运行的坐标系
- 【F[2581/2589]】通道中 9 个轴的轴号。
- 【F[2590/2593]】通道中 4 个主轴的轴号。
- 【F[2594/2595]】语法错报警号。
- 【F[2596/2599]】通道报警号。
- 【F[2600/2603]】通道提示号。
- 【F[2604/2607]】用户输出。
- 【F[2608/2615]】通道中运行的 M 代码，最多 8 个。

通道中 T 指令。通道中在执行 T 代码时，寄存器中是 T 代码的值；否则为 -1。

【F2617】通道中 B 指令。镗床 B 轴 PLC 执行,另外分度用 B 指令

【F2632】刀具到达最大寿命报警的刀具号。

【F2636.0】通道复位中

【F2632.1】程序准停到位

【F2632.2】斜轴标志

【F2632.3】通道运行插补指令

【F2632.4】主轴同步标志

【 F2616 】

【F2632.5】手轮进给方向

【F2637.0】子程序流程开始

【F2637.1】子程序等待进给保持，保存断点

【F2637.2】断点标志

【F2637.3】开始加载子程序

【F2637.4】加载完成

【F2637.5】开始运行

【F2637.6】运行完成

【F2637.7】断点恢复完成

【F2637.8】流程结束

【F2637.9】流程错误

【F2637.10】流程复位

【F2637.11】流程等待解释器复位完成

【F2638.0】刀具寿命换刀累加标记

通道控制字

概述 每通道配置状态字 80 个。每个字均有 16 位字节，第一行表示 0~7 位，第二行表示 8~15 位。使用通道状态字时，需要加上通道编号偏置。

轴控制字

D7 D6 D5 D4 D3 D2 D1 D0
D15 D14 D13 D12 D11 D10 D9 D8

G2560	测量中断	空运行	循环启动	进给保持	工作模式	工作模式	工作模式	工作模式
	数据保存	数据恢复	复位	清缓冲	急停	面板复位	复位应答	校验

G2561	数据恢复	任意行	重运行	解释复位	选择停	跳段标志	重运行 2	解释器启
	保留	程序修改	保留	手轮中断	外部中断	用户运动	保留	解释保存

G2562	保留	保留	保留	保留	保留	保留	保留	保留
	保留	保留	转速到达	无主轴	MST 锁	速度检查	保留	保留

G2620	面板使能	PMC	手摇	回零	增量	手动	单段	自动
	保留	保留	保留	保留	保留	快移	增量倍率	

G2621	手摇 1				手摇 0			
	保留	保留	保留	摇 1 使能	摇 1 倍率		摇 0 倍率	

G2622	轴 7+	轴 6+	轴 5+	轴 4+	轴 3+	轴 2+	轴 1+	轴 0+
	保留	保留	保留	保留	保留	保留	保留	轴 8+

G2623	轴 7-	轴 6-	轴 5-	轴 4-	轴 3-	轴 2-	轴 1-	轴 0-
	保留	保留	保留	保留	保留	保留	保留	轴 8-

详细说明 【G2560.0/1/2/3】工作模式控制字。0:复位模式 1:自动模式 2:手动模式 3:增量模式 4:手摇模式 5:回零模式 6:PMC 模式 7:单段模式 8:MDI 模式

【G2560.4】进给保持：设置通道进给保持。

【G2560.5】循环启动：设置通道循环启动。

【G2560.6】空运行：设置通道为空运行状态。

【G2560.7】测量中断标志。设置此标志为 1 时，系统中断 G31 运行的指令，与 G2582 配合使用。

- 【G2560.8】校验
- 【G2560.9】PLC 复位应答：PLC 复位完成时，设置此标志为 1。
- 【G2560.10】面板复位标志。PLC 通过检测此标志来判断系统是否在复位
- 【G2560.11】急停标志。机床急停时，设置此标志。
- 【G2560.12】清通道缓冲标志。
- 【G2560.13】机床复位时，设置此标记。
- 【G2560.14】通道数据恢复标志
- 【G2560.15】通道数据保存
- 【G2561.0】解释器启动标志
- 【G2561.1】程序重新运行第 2 步标志
- 【G2561.2】跳段标志。设置此标志为 1 时，系统跳段。
- 【G2561.3】选择停标志。设置此标志为 1 时，系统选择停。
- 【G2561.4】解释器复位标志
- 【G2561.5】程序重新运行标志
- 【G2561.6】MDI 复位到程序头标志
- 【G2561.7】解释器数据恢复标志
- 【G2561.8】解释器数据保存标志
- 【G2561.9】准停检查
- 【G2561.10】用户运动控制标志
- 【G2561.11】外部中断标志
- 【G2561.12】开启手轮中断
- 【G2561.13】快移修调为 0 时，用进给修调控制 G00，最高 25%
- 【G2561.14】程序修改标志
- 【G2561.15】工件或工具坐标变化，请求重新解释
- 【G2562.0】1 号主轴 S 指令应答字
- 【G2562.1】2 号主轴 S 指令应答字
- 【G2562.2】3 号主轴 S 指令应答字
- 【G2562.3】4 号主轴 S 指令应答字
- 【G2562.8】手轮试切进给方向。0 为前进，1 为后退
- 【G2562.10】主轴速度检查

【G2560.10】面板复位标志。PLC 通过检测此标志来判断系统是否在复位

【G2560.11】急停：设置通道急停。

【G2562.11】MST 锁

【G2562.12】主轴未启动

【G2562.13】主轴速度未到达

【G2562.14】跟随开始开启

【G2562.15】手轮试切，使用手轮倍率

【G2563】T 指令

【G2564】进给修调

【G2565】快移修调

【G2566/67/68/69】主轴修调。通道中 4 个主轴的修调

【G2570/71/72/73/74/75/76/77】主轴输出指令。通道中 4 个主轴的输出指令，PLC 获取主轴转速（F2570-F2577）后，计算主轴修调等，输出主轴指令。在伺服主轴中输出为主轴转速；在变频器主轴中，输出为 DA 值。

【G2578】F2578.1 虚轴控制

【G2579】加工计件

【G2580/81】保护区掩码

【G2582】G31 的编号。执行 G31 中断时，中断的 G31 编号。

【G2584/85/86/87】用户位输入

【G2588~2607】用户数值输入

【G2608~2615】通道 M 代码响应。PLC 没有执行 M 代码时，设置为-1；
PLC 正在执行 M 代码时，设置为-2；
PLC 执行 M 代码完成时，设置为当前执行完的 M 代码

【G2616】通道 T 代码响应。PLC 执行 T 代码完成时，设置为当前执行完的 T 代码；

否则，设置为-1。

【G2617】刀具切线跟随

【G2636.0】通道复位（PLC 置寄存器，通知 HMI 进行通道复位）

【G2636.3】IRQ 控制

【G2636.4】通道禁止复位【Reset 按键无效】

【G2636.5】寿命计时/计数暂停

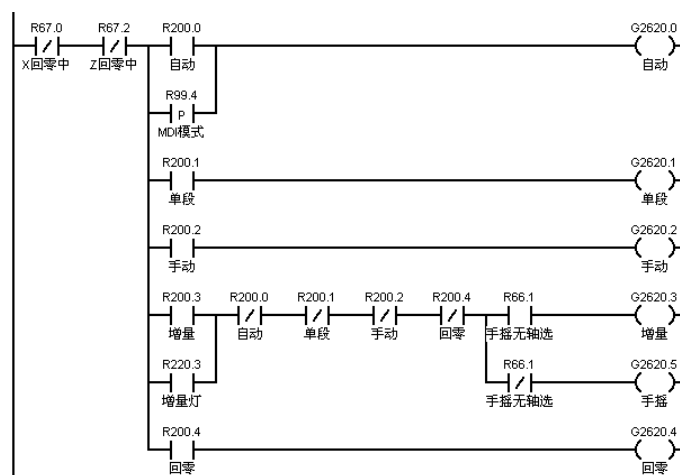
- 【G2560.15】通道数据保存。
- 【G2561.0】解释器启
- 【G2637】子程序调用启动
- 【G2638】换刀计数次数
- 【G2970】系统活动通道标志
- 【G2978】系统运动控制通道的控制字
- 【G2980~2989】手摇的控制字【上一个轴选】
- 【G2990~2999】手摇的显示输出
- 【G3010~3025】PLC 外部报警(PLC 外部报警，同时可有 $8*32=256$ 种 PLC 外部报警)
- 【G3040~3055】PLC 外部事件(PLC 外部事件，同时可有 $8*32=256$ 种 PLC 外部事件)
- 【G3056~3070】PLC 外部提示(PLC 外部事件，同时可有 $8*32=256$ 种 PLC 外部事件)
- 【G3080~3099】温度传感器值

状态字和控制字编程实例

工作模式设置

示例

梯形图



功能说明

在通道工作模式中设置状态，轴在位置控制模式下，设置当前通道中的工作模式为自动，单段，手动，增量，手摇或回零。

工作模式获取

示例

梯形图



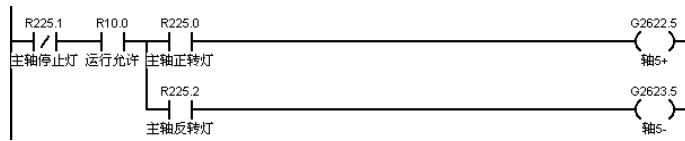
功能说明

在通道工作模式中获取状态，获取当前通道中的工作模式为自动，单段，手动，增量，手摇或回零状态。

进给轴与主轴控制

示例

梯形图

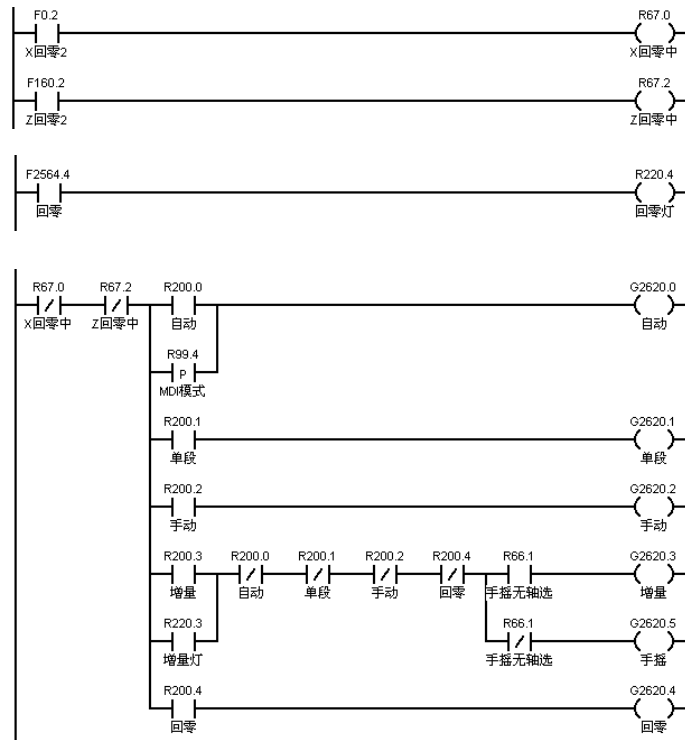


功能说明

在控制进给轴的移动，主轴的正反转时，需要设置当前通道模式为手动模式下，当选择了轴选并且按下了轴的正向或负向移动键时，设置当前轴的移动状态，从而使轴开始移动，而主轴则是按下了主轴正反转的按键时，设置主轴的转动方向。

回零

示例 梯形图



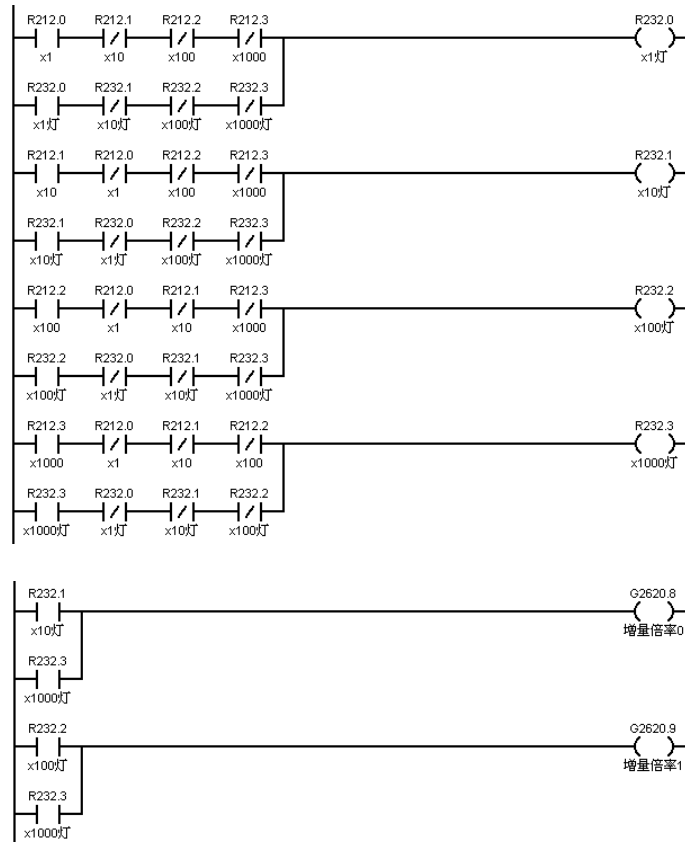
功能说明

通过状态寄存器获取当前通道是否处于回零过程中,如果在碰到回零挡块的过程中即回零第一过程,切换其他运行状态是被允许的,而当在轴寻找 Z 脉冲的过程中即回零第二过程,是不允许切换状态的。

增量倍率修调

示例

梯形图



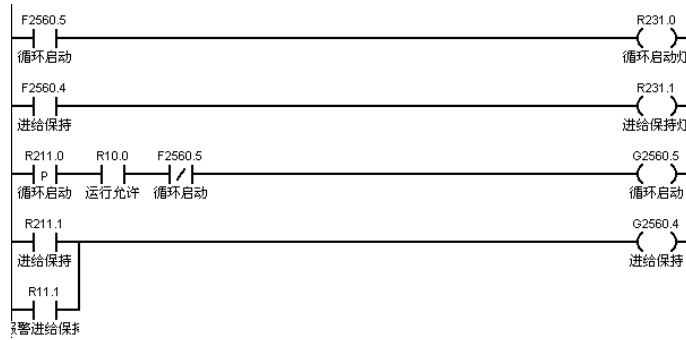
功能说明

增量倍率占用 2 位。00 代表 x1；01 代表 x10；10 代表 x100；11 代表 x1000。并且通过以上轴移动寄存器的设置来控制轴的移动。

循环启动与进给保持

示例

梯形图



功能说明

在通道工作模式在自动或单段下，并且不在循环启动状态时，设置循环启动。在循环启动状态，可以设置进给保持。当设置成功置进给保持状态。

扩展功能模块

本章主要介绍以下内容

6.1 NC 功能

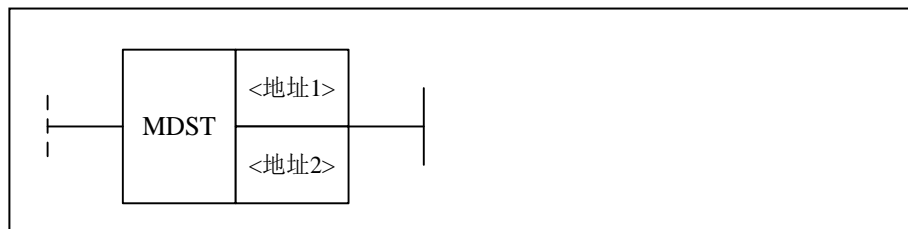
6.2 轴功能单元

6.3 系统功能

NC 功能

通道模式设置 MDST

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数、F、G、R、W、D、P、B	工作模式值	后置 ×

功能说明

设定当前通道的工作模式（自动，单段，手动，增量，回零，手摇，PMC）

参数说明

工作方式 参数	工作方式						
	自动	单段	手动	增量	回零	手摇	PMC
D2□□□□	1	2	4	8	16	32	64

补充说明

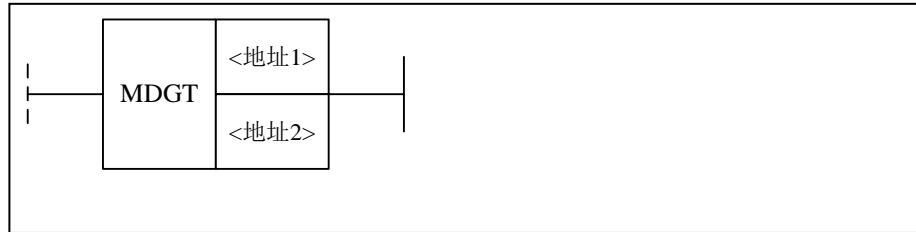
通道状态中有轴回零时，不允许切换模式。

示例

梯形图	
语句表	MDST 0 R0
描述	根据 R0 的值设置通道 0 的工作模式

通道模式获取 MDGT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数、F、G、R、W、D、P、B	工作模式值	后置 ×

功能说明

获取当前通道的工作模式值。

参数说明

参数	工作方式						
	自动	单段	手动	增量	回零	手摇	PMC
D2□□□	1	2	4	8	16	32	64

补充说明

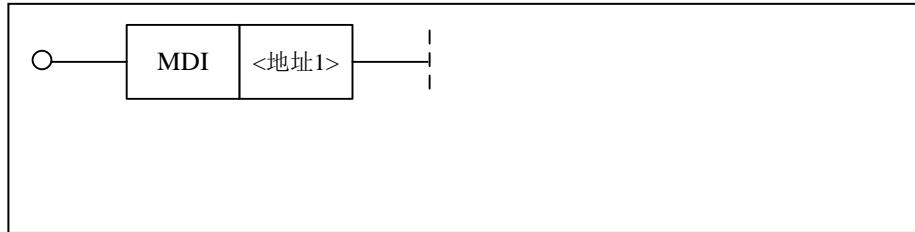
取通道状态时，有轴回零时，不允许切换模式。

示例

梯形图	
语句表	MDGT 0 R1
描述	取通道 0 的工作模式，放到 R1 寄存器中

模式 MDI

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○ 后置 ×

功能说明 获取通道 MDI 模式。

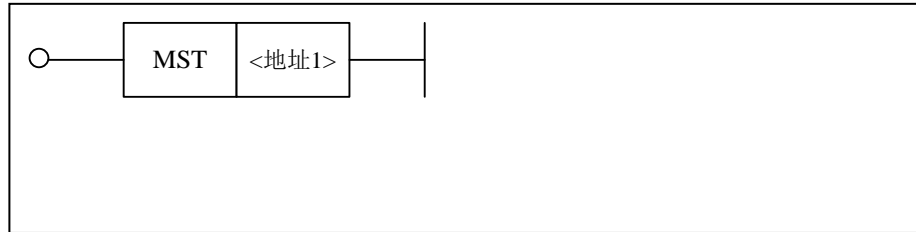
参数说明 参数 1：通道号。

示例

梯形图	
语句表	MDI 0
描述	X36.4 通时，通道 0 为 MDI 模式

锁住通道 MST

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓ 后置 ×

功能说明

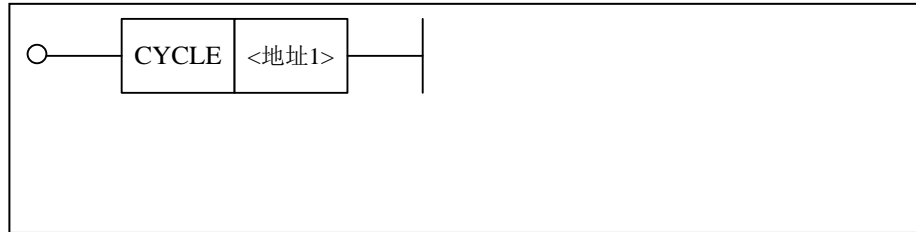
锁住通道 MST，当该功能块导通该通道的所有 MST 指令无法使用，将直接跳过。

示例

梯形图	
语句表	MST 0
描述	X36.4 通时，通道 0 被锁

循环启动开启 CYCLE

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓ 后置 ×

功能说明

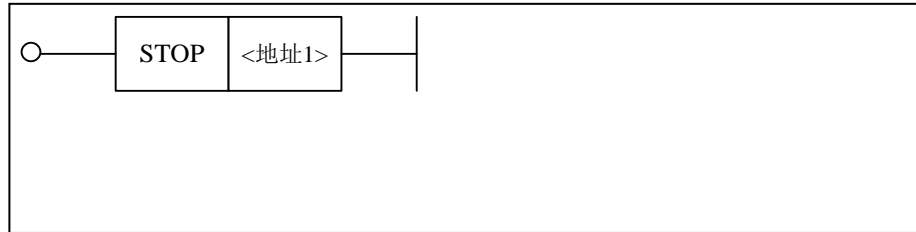
通过参数设定需要启动循环启动的通道, 并通过 ACT 信号来启动循环启动。

示例

梯形图	
语句表	CYCLE 0
描述	X36.4 导通时, 将通道 0 设为循环启动

急停 STOP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓ 后置 ×

功能说明

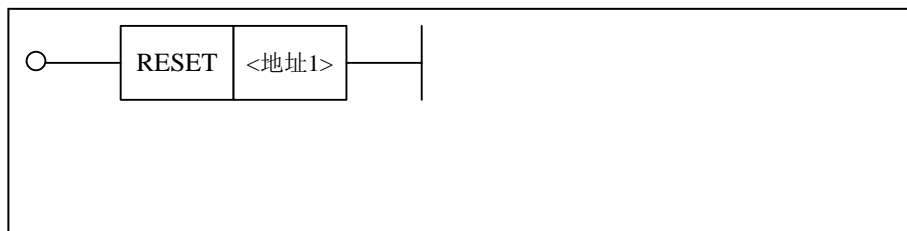
通过参数设定需要急停的通道，并通过 ACT 信号来启动急停。

示例

梯形图	
语句表	<pre>LD X1.2 STOP 0</pre>
描述	X1.2 导通时，将通道 0 设为急停

复位 RESET

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓ 后置 ×

功能说明

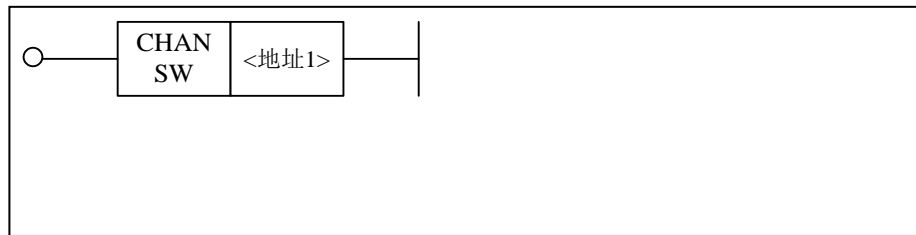
通过参数设定需要复位的通道，并通过 ACT 信号来启动复位。

示例

梯形图	
语句表	<pre>LD X2.4 RESET 0</pre>
描述	X2.4 导通时，将通道 0 设为复位

通道切换 CHANSW

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	置进给保持的通道	前置 ✓ 后置 ×

功能说明

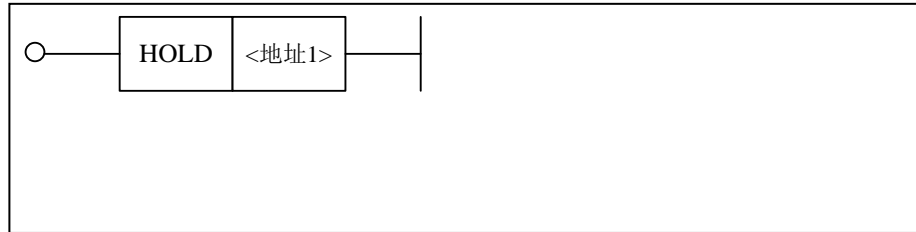
通过参数设定需要切换的通道，并通过 ACT 信号来启动切换。

示例

梯形图	
语句表	<pre>LD X36.4 CHANSW 0</pre>
描述	X36.4 导通时，将通道 0 设为活动通道

进给保持开启 HOLD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	置进给保持的通道	前置 ✓ 后置 ×

功能说明

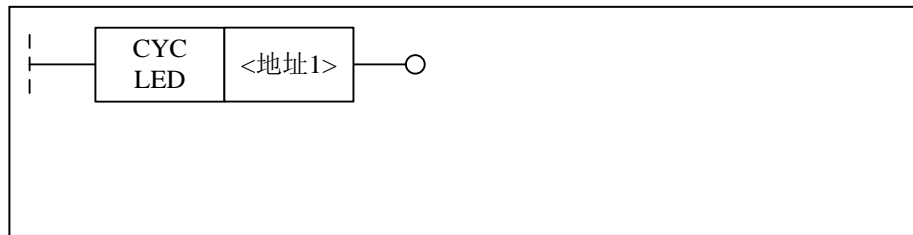
通过参数设定需要进给保持的通道，并通过 ACT 信号来进给保持。

示例

梯形图		
语句表	<pre>LD X36.4 HOLD 0</pre>	
描述	<p>X36.4 导通时，将通道 0 设为进给保持</p>	

循环启动指示灯 CYCLED

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	获取循环启动状态的通道	前置 ○ 后置 ×

功能说明

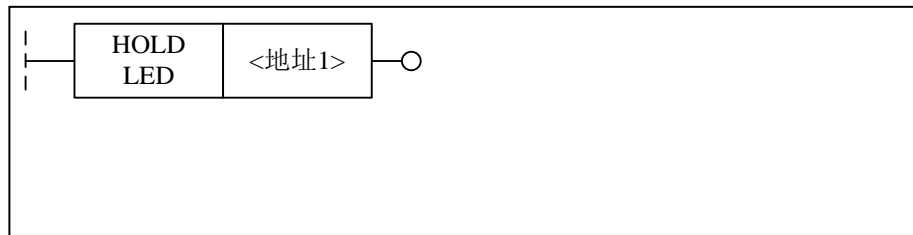
通过参数设定需要获取循环启动状态的通道，循环启动成功则输出点亮循环启动灯。

示例

梯形图	
语句表	<pre>LDT CYCLED 0 OUT Y36.4</pre>
描述	取通道 0 的循环启动状态，点亮循环启动灯

进给保持指示灯 HOLDLED

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	获取进给保持状态的通道	前置 ○ 后置 ✓

功能说明

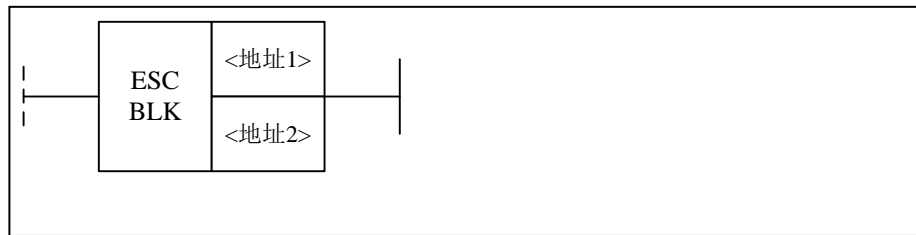
通过参数设定需要点亮进给保持等的通道，并通过 ACT 信号来点亮进给保持灯。

示例

梯形图	
语句表	<pre>LDT HOLDLED 0 OUT Y36.5</pre>
描述	根据通道 0 的进给保持状态，控制进给保持灯

程序跳段（G31）ESCBLK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	需要激活跳段功能的通道	前置 ○
<地址 2>	□□□□	INT	常数	G31 的序号	后置 ×

功能说明

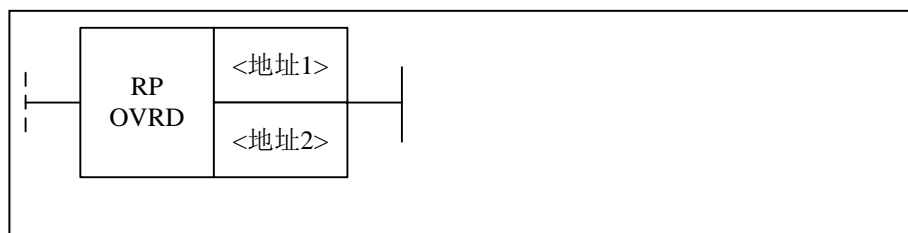
通过参数设定需要激活跳段的通道，并通过 ACT 信号使能该功能。

示例

梯形图	
语句表	<pre>LDP X31.4 ESCBLK 0 1</pre>
描述	当 X31.4 上升沿导通时，0 号通道的第一个 G31 语句激活（G31.1）

快移修调设置 RPOVRD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	修调值	后置 ×

功能说明 通过参数 1 选择的通道，参数 2 通过寄存器来传递修调值，通过 ACT 来使能快速修调功能。

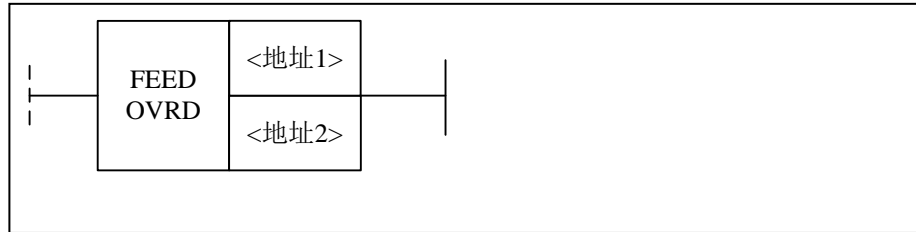
补充说明 螺纹切屑时不允许改变修调值。

示例

梯形图	
语句表	<pre>LDT RPOVRD 0 R7</pre>
描述	用 R7 的值设置 0 通道的快移修调

进给修调设置 FEEDOVRD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	修调值	后置 ×

功能说明 通过参数 1 选择的通道，参数 2 通过寄存器来传递修调值，通过 ACT 来使能进给修调功能。

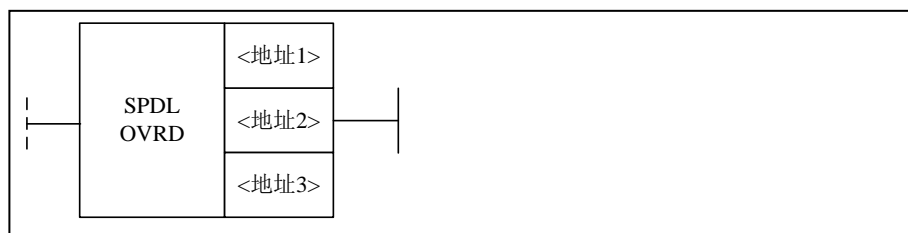
补充说明 螺纹切屑时不允许改变修调值。

示例

梯形图	
语句表	<pre>LDT FEEDOVRD 0 R7</pre>
描述	用 R7 的值设置 0 通道的进给修调

主轴修调设置 SPDLOVRD

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○ 后置 ×
<地址 2>	□□□□	INT	常数	主轴号	
<地址 3>	□□□□	INT	常数、X、Y、F、 G、R、W、D、P、 B	修调值	

功能说明

通过参数 1 选择的通道，参数 2 选择主轴号，参数 3 通过寄存器来传递修调值，通过 ACT 来使能进给修调功能。

补充说明

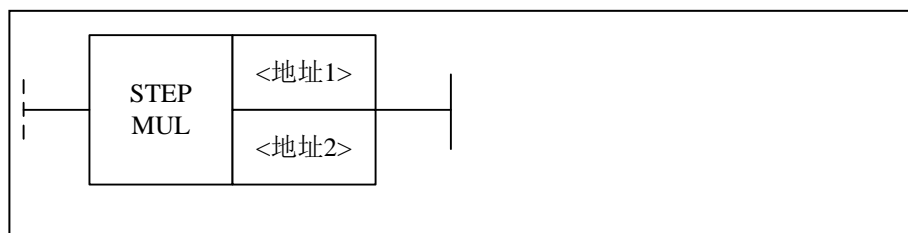
螺纹切屑时不允许改变修调值。

示例

梯形图	
语句表	<pre>LDT SPDLOVRD 0 0 R7</pre>
描述	用 R7 的值设置 0 通道的 0 号主轴修调值

增量（步进）倍率设置 STEPMUL

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	轴号	前置 ○
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	倍率值	后置 ×

功能说明

通过参数 1 选择的轴号，参数 2 通过寄存器来传递倍率值，通过 ACT 来使能进给修调功能。

补充说明

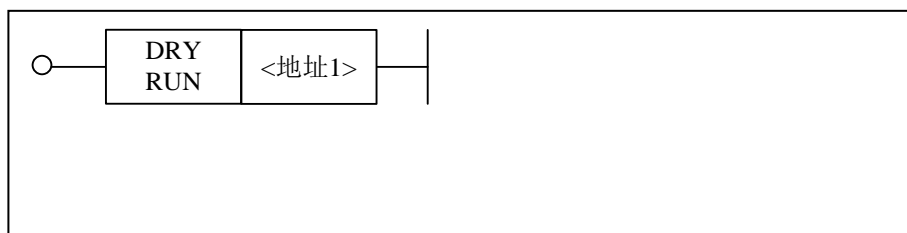
该功能只能在切换到增量（步进）状态时才能使用。

示例

梯形图	
语句表	<pre>LDT STEPMUL 0 R7</pre>
描述	用 R7 的值设置 0 通道的增量倍率

空运行开关 DRYRUN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 √ 后置 ×

功能说明

在自动方式下按一下机床控制面板上的空运行按键指示灯亮 CNC 处于空运行状态程序中编制的进给速率被忽略坐标轴以最大快移速度移动。

空运行不做实际切削目的在于确认切削路径及程序。

在实际切削时应关闭此功能否则可能会造成危险。

此功能对螺纹切削无效。

参数说明

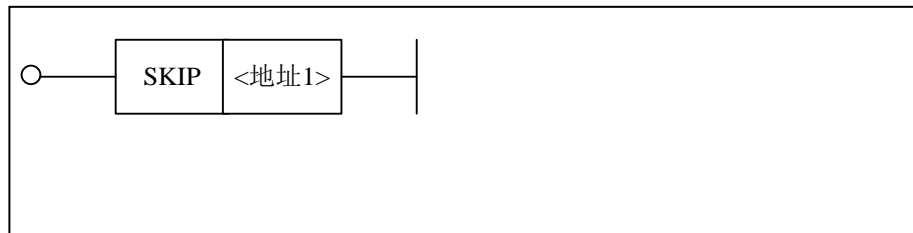
参数 1: 通道号。

示例

梯形图	
语句表	DRYRUN 2
描述	当 Y32.2 导通时，通道 2 空运行

跳段开关 SKIP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明

自动加工时，系统可跳过某些指定的程序段。如在某程序段首加上“/”，且面板上按下该开关，则在自动加工时，该程序段被跳过不执行；而当释放此开关时，“/”不起作用，该段程序被执行。

参数说明

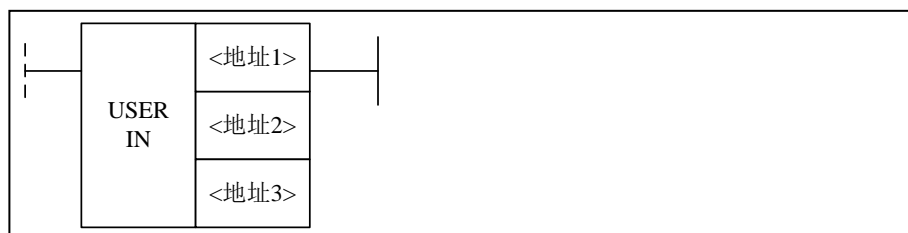
参数 1: 通道号。

示例

梯形图	
语句表	SKIP 2
描述	当 Y32.2 导通时，通道 2 跳段有效

用户输入 USERIN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
<地址 2>	□□□□	INT	常数		后置 ×
<地址 3>	□□□□	INT	常数		

功能说明

设置用户输入，当 ACT 有效，设置通道中用户自定义的组和位为 1，此时系统中的宏变量将随之变化。

参数说明

参数 1: 通道号

参数 2: 暂不使用

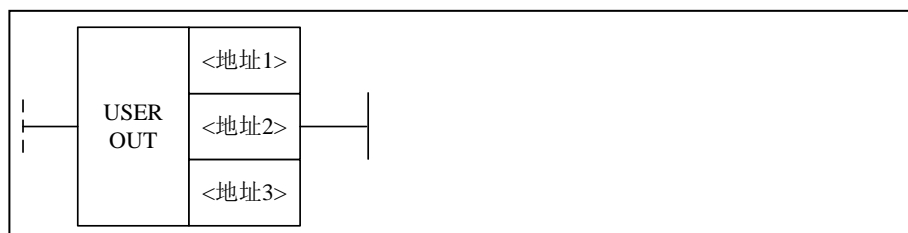
参数 3: 2 的幂，如填 17，那么#1190 的值就是 $2^{17}=131072$

示例

梯形图	
语句表	USERIN 0 1 1
描述	当 X31.4 导通时，0 号通道对应的用户输入组宏变量#1190 值为 2

用户输出 USEROUT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	无	无	无		后置 ×
<地址 3>	□□□□	INT	Y、R		

功能说明

设置用户输出，在程序中设定宏变量#1191 的值，由此值来确定需要输出的用户自定义输出的组号和位置号。共定义 32 位输出，得到的 4 组 8 位输出，输出的开始地址由参数 3 定义，往后延续 4 组。

参数说明

参数 1：通道号；

参数 2：暂时不生效；

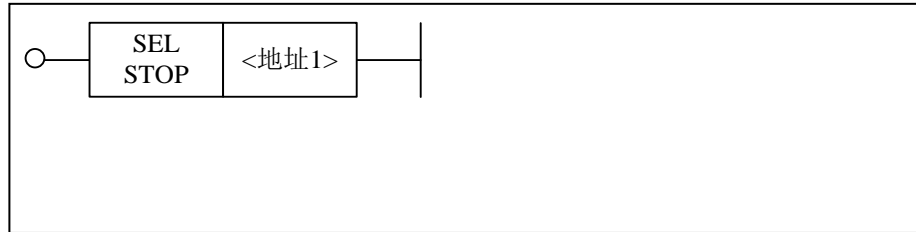
参数 3：输出寄存器开始地址，输出值是 32 位的，所以对于 8 位的 y 寄存器，将占用连续 4 个 y 寄存器；

示例

梯形图	
语句表	USEROUT 0 1 Y1
描述	当 X36.4 有信号时，如果#1191 = 12，则输出 Y1.2 和 Y1.3，同时 Y1~Y4 的其他位都是 0

选择停开关 SELSTOP

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 √
					后置 ×

功能说明 “选择停”按键有效时（指示灯亮），自动方式下，遇有 M01 程序停止。

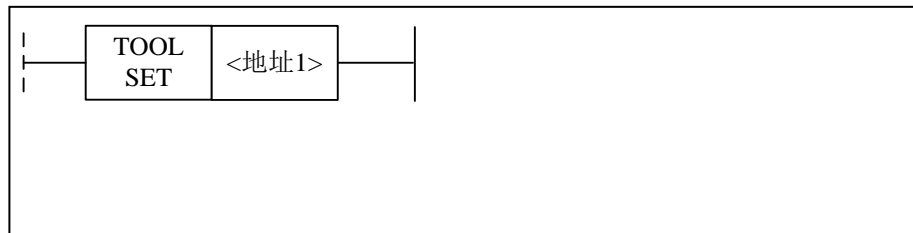
参数说明 参数 1：通道号。

示例

梯形图	
语句表	SELSTOP 0
描述	当 Y32.2 导通时，通道 0 选择停有效

矢量刀具方向设置 TOOLSET

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ×
					后置 ✓

功能说明

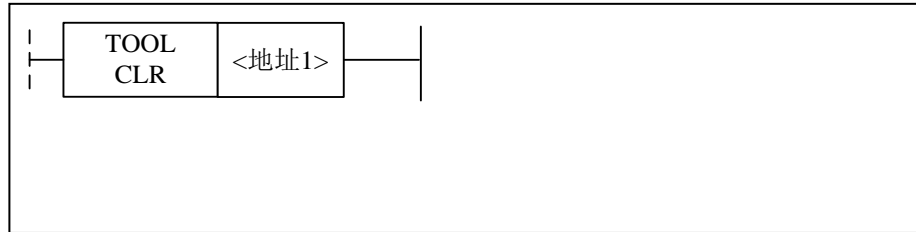
该功能一般在 5 轴加工过程中使用,在该通道设置以当前刀具的矢量方向为 Z 方向。通过使能该功能手动将刀具沿着刀具矢量的方向进退刀。

示例

梯形图	<p>The diagram shows a single rungs of a ladder logic circuit. On the left, there is a normally open contact labeled 'X36.4'. This contact is connected to a coil (output) labeled 'TOOL SET' with the value '0' inside it. The circuit ends with a vertical line on the right.</p>
语句表	TOOLSET 0
描述	当 Y32.2 导通时, 通道 0 刀具方向有效

矢量刀具方向清除 TOOLCLR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ×
					后置 ✓

功能说明

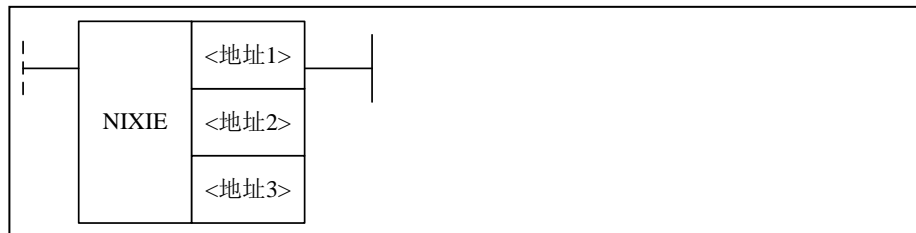
该功能一般在 5 轴加工过程中使用,在该通道取消以当前刀具的矢量方向为 Z 方向。该功能与 TOOLSET 功能配合使用。

示例

梯形图	
语句表	TOOLCLR 0
描述	设置通道 0 的刀具方向无效

8 位数码管 NIXIE

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B	要在数码管上显示的数字	前置 ✓
<地址 2>	□□□□	INT	常数	“0”表示单字节，“1”表示双字节	后置 ✓
<地址 3>	□□□□	BOOL	Y、R、W、D、B	设置面板上的 8 位数码管	

功能说明

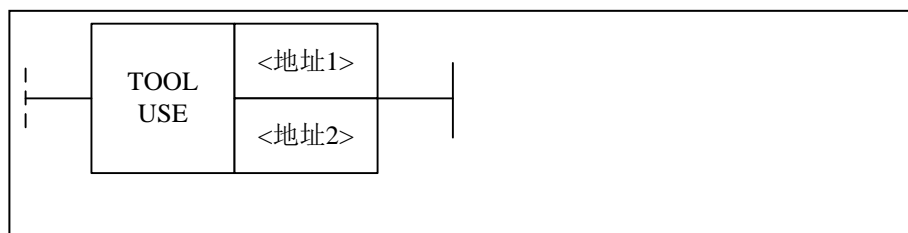
在面板上通过 8 位数码管显示当前刀的刀号

示例

梯形图	<p>The diagram shows a normally open contact labeled 'T' connected to the input of a rectangular box labeled 'NIXIE'. To the right of the NIXIE box, there are three stacked rectangular boxes representing outputs, labeled 'R23', '0', and 'Y37' from top to bottom. A vertical line on the right side of the NIXIE box indicates a power or ground connection.</p>
语句表	NIXIE R23 0 Y37
描述	将 R23 中的刀具号显示到数码管上

刀具显示 TOOLUSE

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ○
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		后置 ×

功能说明

将当前执行的 T 代码中的刀具号显示到数控系统界面上。

参数说明

参数 1: 通道号;

参数 2: 刀号;

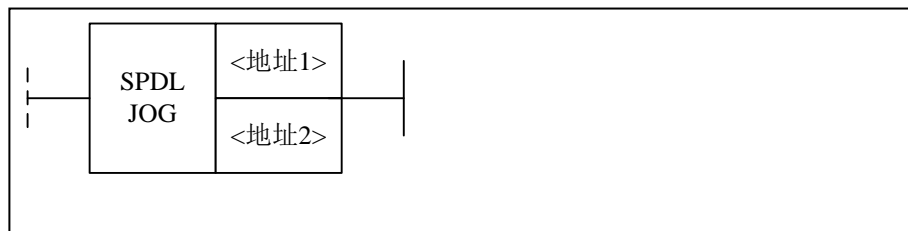
示例

梯形图	
语句表	TOOLUSE 0 R23
描述	将 0 号通道的刀具号显示到界面上

轴功能单元

主轴手动设置 SPDLJOG

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	BOOL	X、Y、F、G、R、 W、D、P、B		后置 ×

功能说明 手动控制主轴。

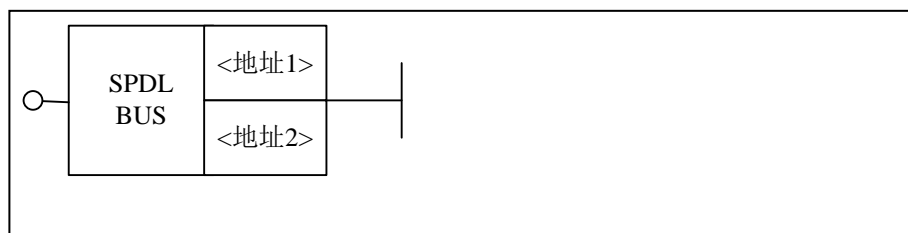
参数说明 参数 1：主轴号；
参数 2：正转点；

示例

梯形图	<p>The diagram shows a single rung of a ladder logic circuit. It starts with a normally open contact labeled 'X31.4'. This is followed by an SPDLJOG instruction box. The box is divided into two parts: the top part contains the number '0' and the bottom part contains 'X32.4'. The circuit ends with a vertical line representing the power rail.</p>
语句表	SPDLJOG 0 X32.4
描述	当 X31.4 有效时，如果 X32.4 有效，0 号轴以默认转速正转

主轴控制【伺服主轴】SPDLBUS

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	常数		后置 ×

功能说明

设置某通道的某个主轴有效,将通道参数中设置的与主轴号相关联的设备设置为主轴。例如当前 0 通道的 0 号主轴的逻辑轴号为 5, (假定 5 号轴已经伺服使能成功) 则将逻辑轴 5 作为当前通道的第一个主轴, 通过该功能模块使该主轴有效。

参数说明

参数 1: 通道号;

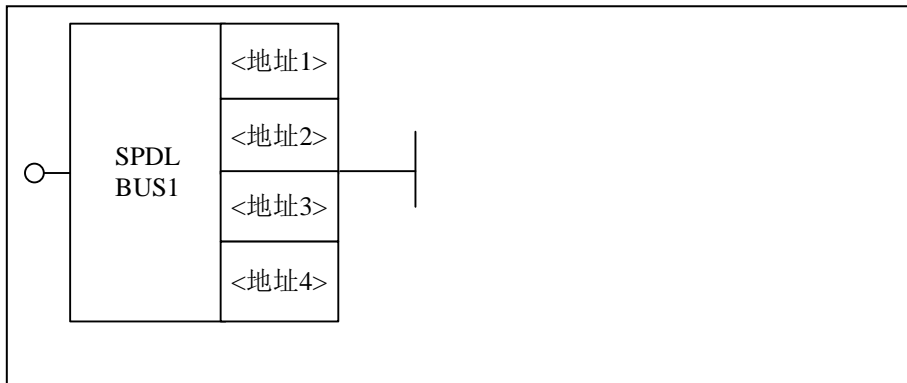
参数 2: 主轴号;

示例

梯形图	
语句表	SPDLBUS 0 1
描述	当 X31.4 有效时, 控制 0 通道 1 号主轴

带档位的主轴控制【伺服主轴】SPDLBUS1

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	常数		
<地址 3>	□□□□	INT	常数、Y、G、R、 W、D、B		
<地址 4>	□□□□	INT	P		

功能说明 带档位的总线式主轴控制。

参数说明 参数 1: 通道号;

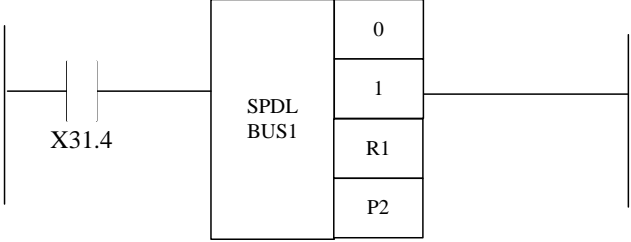
参数 2: 主轴号;

参数 3: 档位寄存器, 从 1 开始;

参数 4: 控制参数, 指定的参数存放着主轴电机的最大转速、初始转速等数据; 参数 4 主轴控制值参考包括:

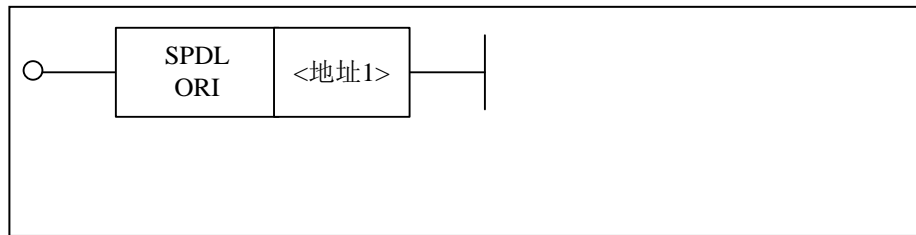
0	电机最大转速
1	实测最小转速
2	实测最大转速
3	当前传动比分子
4	当前传动比分母

示例

<p>梯形图</p>	
<p>语句表</p>	
<p>描述</p>	<p>当 X31.4 有效时，0 通道 1 号主轴的主轴修调当前档位在 R1 寄存器中，控制参数在中从 P2 开始的用户参数中。P 参数的填写根据机床的实际情况参见参数说明书。</p>

主轴定向使能 SPDLORI

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	轴号	前置 ○ 后置 ×

功能说明

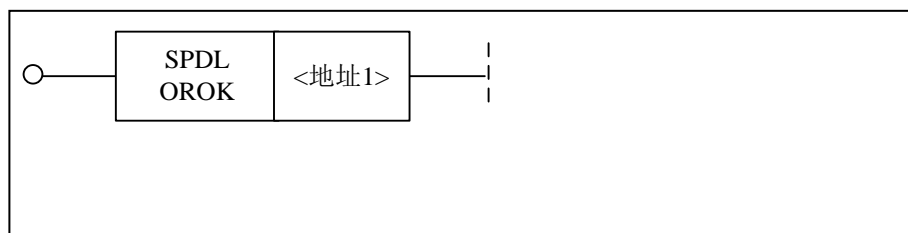
主轴定向使能。在换刀或刚性攻丝开始时需要将主轴定向到一个规定的角度，通过该功能开始主轴定向。主轴的定向角度在伺服驱动器中由参数设定。

示例

梯形图	
语句表	SPDLORI 0
描述	轴号为 0 的主轴定向

主轴定向完成 SPDLOROK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○ 后置 ✓

功能说明

主轴定向完成，表明主轴已经转到了指定的定向角度。

参数说明

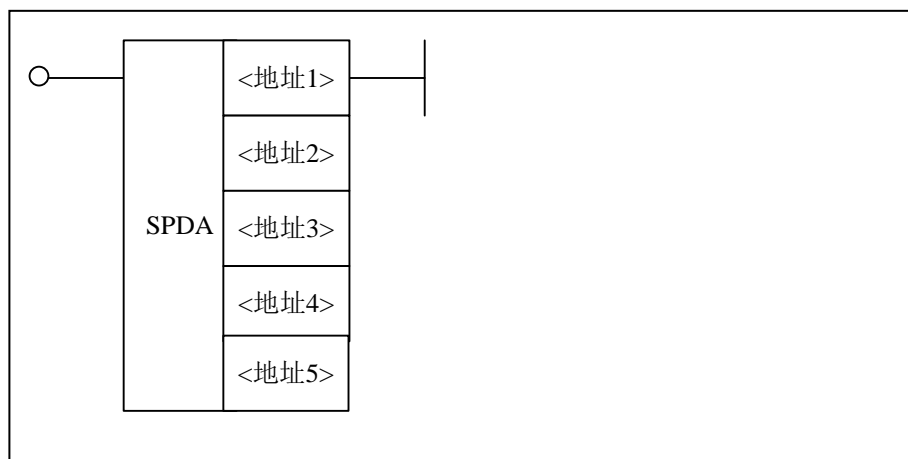
参数 1：轴号。

示例

梯形图	
语句表	SPDLOROK 0
描述	轴号为 0 的主轴定向使能完成时，设置 R10.1 有效

主轴控制【DA】SPDA

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	常数		后置 ×
<地址 3>	□□□□	INT	常数、Y、G、R、 W、D、B		
<地址 4>	□□□□	BOOL	Y、G、R、W、D、 B		
<地址 5>	□□□□	BOOL	P		

功能说明 主轴 DA 控制，用于模拟量主轴的控制。

参数说明 参数 1：通道号

参数 2：主轴号；

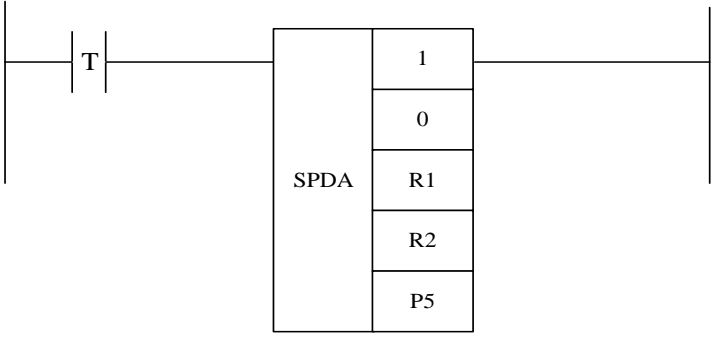
参数 3：档位寄存器（档位从 1 开始）

参数 4：无效

参数 5：主轴控制值参考包括

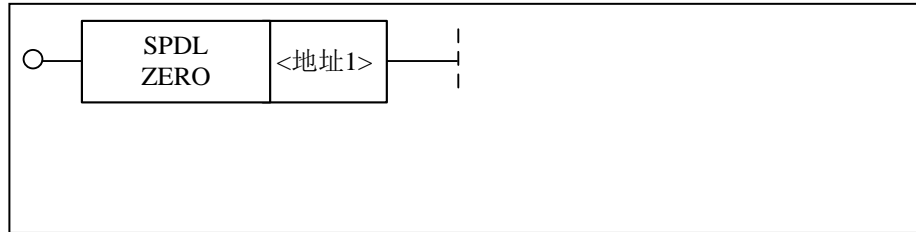
0	电机最大转速
1	实测最小转速
2	实测最大转速
3	当前传动比分子
4	当前传动比分母

示例

<p>梯形图</p>	
<p>语句表</p>	<p>SPDA 1 0 R1 R2 P5</p>
<p>描述</p>	<p>1 通道，在 1 通道中 0 号主轴，当前档位在 R1 寄存器中，主轴控制参考值在 R2 中，控制参数在 P5 中</p>

主轴零速检测 SPDLZERO

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
					后置 ×

功能说明 主轴零速检测。

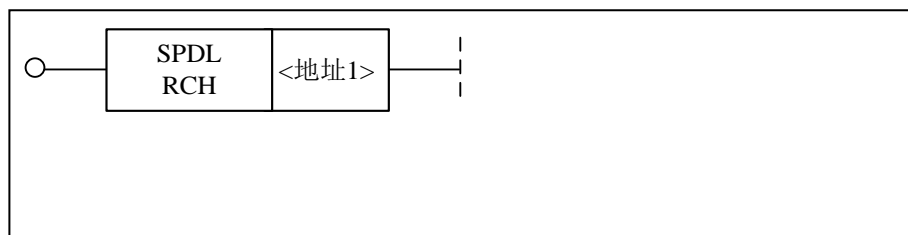
参数说明 参数 1：轴号。

示例

梯形图	
语句表	SPDLZERO 1
描述	1 号主轴零速检测

主轴速度到达 SPDLRCH

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ○ 后置 ×

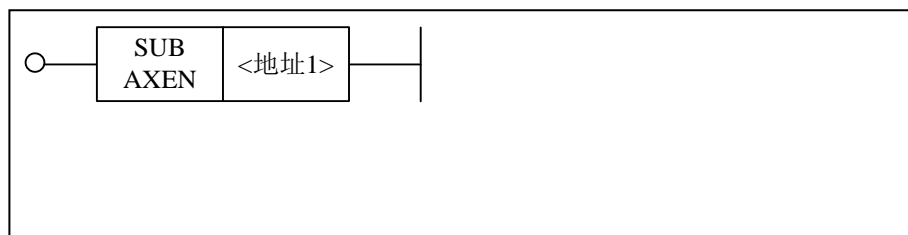
功能说明 检测主轴速度是否到达指令转速。

示例

梯形图	
语句表	SPDLRCH 1
描述	1 号主轴速度到达

从轴回零允许 SUBAXEN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓
					后置 ×

功能说明 设置从轴回零使能，开启该功能将首先由从动轴回零寻找 Z 脉冲。当找到 Z 脉冲后表示从动轴回零完成，然后继续主动轴回零。

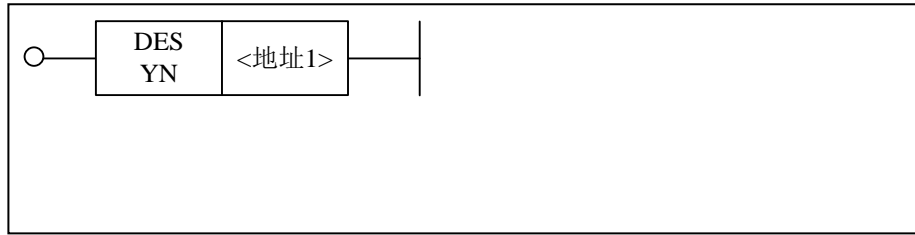
参数说明 参数 1：从轴号

示例

梯形图	
语句表	SUBAXEN 0
描述	当 X32.6 有效时，0 号从轴回零允许

释放从轴 DESYN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓ 后置 ×

功能说明

释放从轴。由参数将一个轴设定为另外一个轴的从动轴，那么在给主动轴发送指定的同时从轴也将得到同样的指令。当开启从轴释放功能时，从轴将与主动轴分离，从而不再接受主动轴的指令脉冲。

参数说明

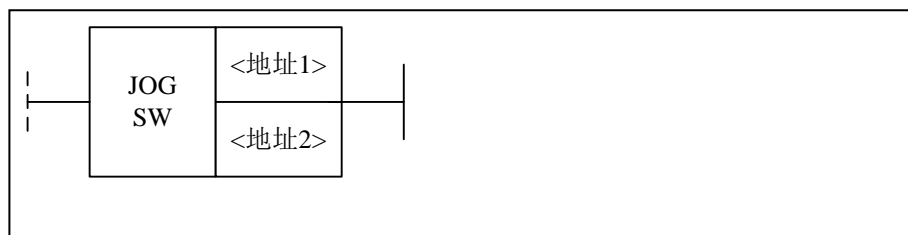
参数 1：从轴的轴号；

示例

梯形图	
语句表	DESYN 0
描述	当 X32.6 有效时，释放 0 号从轴

轴点动控制 JOGSW

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ○
<地址 2>	□□□□	BOOL	X、Y、F、G、R、W、D、P、B		后置 ×

功能说明 手动控制轴点动使能。

参数说明 参数 1：轴号；

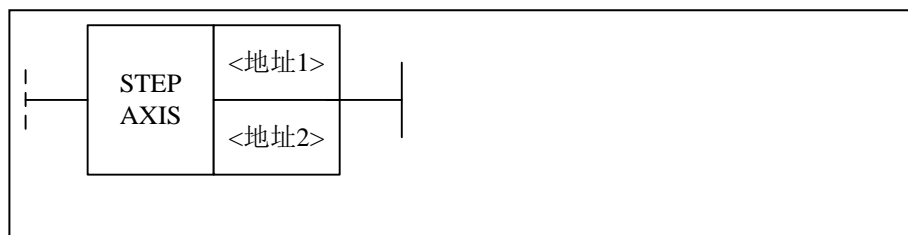
参数 2：轴正点动方向，设置值为“1”表示正方向，“0”表示负方向

示例

梯形图	
语句表	JOGSW 0 X32.3
描述	X32.3 接通表示轴 0 正向手动使能未接通表示轴 0 负向手动使能

轴步进控制 STEPAXIS

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ○
<地址 2>	□□□□	BOOL	X、Y、F、G、R、W、D、P、B		后置 ×

功能说明

步进控制轴使能。

参数说明

参数 1：轴号；

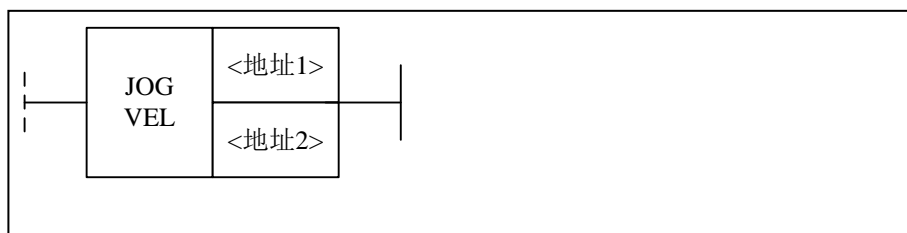
参数 2：轴步进方向，“0”表示正方向点动，“1”表示负方向点动。

示例

梯形图	
语句表	STEPAXIS 0 X32.3
描述	X32.3 接通表示轴 0 负向点动使能，未接通表示轴 0 正向点动使能

轴点动速度选择 JOGVEL

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		后置 ×

功能说明 手动控制轴点动速度。

参数说明 参数 1：轴号；

参数 2：轴速度，取值如下：

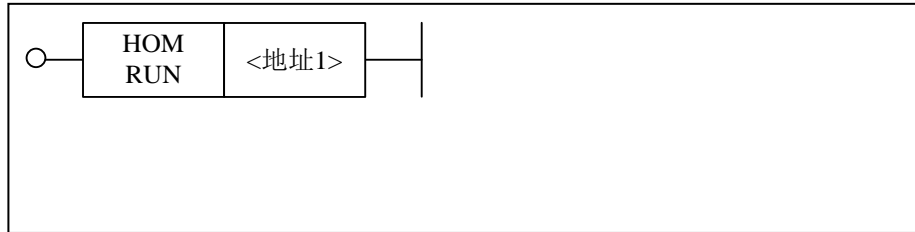
- 1：参数点动速度
- 2：参数快移速度
- >2：速度（脉冲/周期）

示例

梯形图	
语句表	JOGVEL 0 R0
描述	当 X33.2 导通时，使轴 0 以 R0 中给定的速度运转

轴回零启动 HOMRUN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 回零开始。

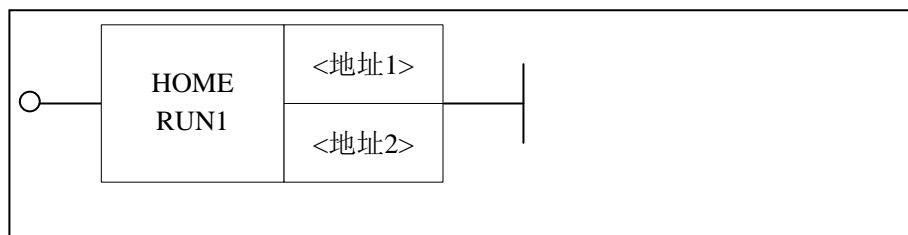
参数说明 参数 1：轴号；

示例

梯形图	
语句表	HOMRUN 1
描述	X1.1 导通，轴 1 回零开始

轴回零启动 HOMERUN1

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
<地址 2>	□□□□	BOOL	X、Y、F、G、R、 W、D、P、B		后置 ×

功能说明 回零开始。

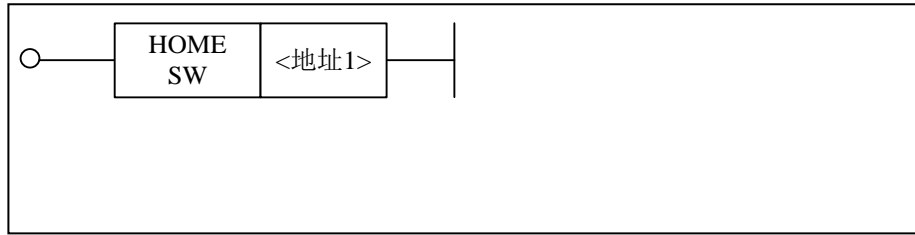
参数说明 参数 1：轴号；
参数 2：轴回零启动方向；

示例

梯形图	
语句表	HOMERUN1
描述	X1.1 导通, 轴 1 回零开始, X23.3 接通表示轴 0 正向点动使能, 未接通表示轴 0 负向点动使能

回零接近开关 HOMESW

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 轴遇到回零挡块。

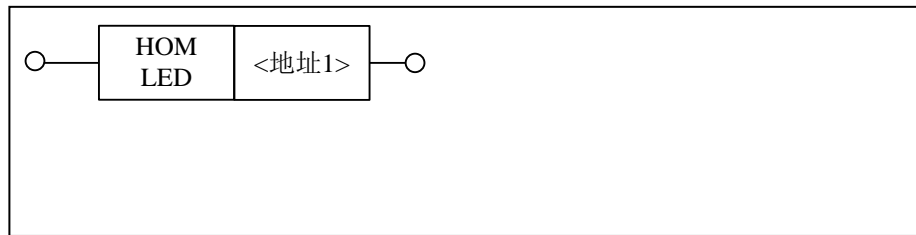
参数说明 参数 1: 轴号;

示例

梯形图	
语句表	HOMESW 1
描述	X1.4 导通表明 1 号轴遇到回零挡块

轴回零完成 HOMLED

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ✓

功能说明 回零完成。

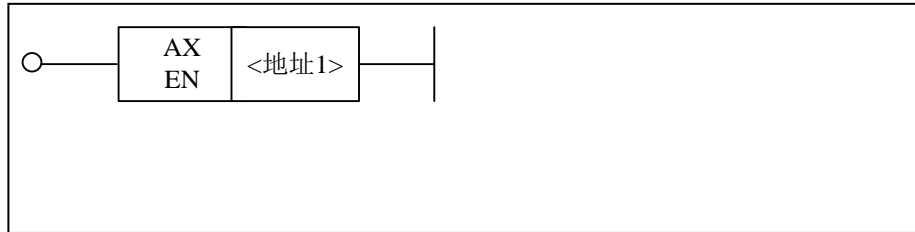
参数说明 参数 1：轴号；

示例

梯形图	
语句表	HOMLED 1
描述	1 号轴回零完成，点亮 1 号轴灯

轴使能 AXEN

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 轴使能。

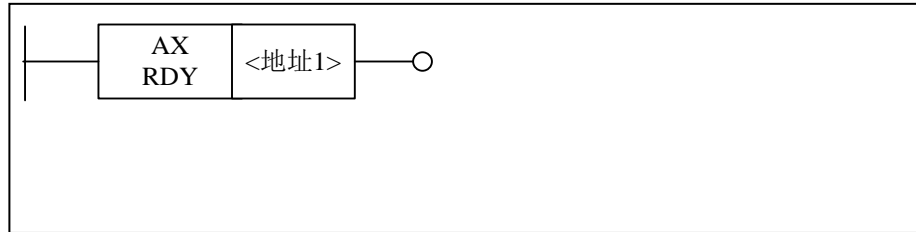
参数说明 参数 1：轴号，可以是常数或者寄存器。

示例

梯形图	
语句表	AXEN 1
描述	X0.1 导通时，1 号轴使能

轴就绪指示【总线】AXRDY

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 × 后置 ✓

功能说明 轴准备好。

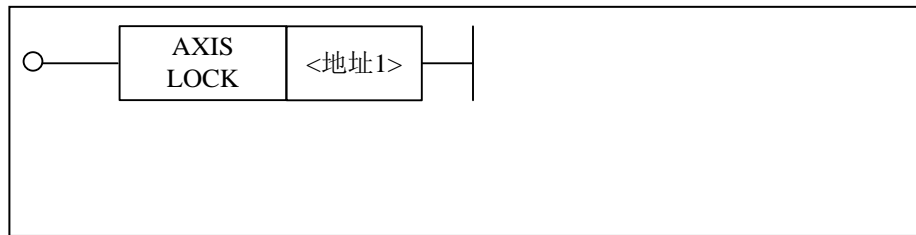
参数说明 参数 1：轴号

示例

梯形图	
语句表	AXRDY 1
描述	轴 1 准备好置 R10.1 为 1

轴锁住 AXISLOCK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明

轴锁住。

参数说明

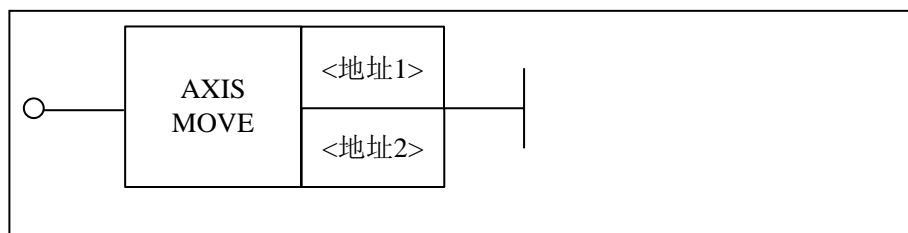
参数 1：轴号。

示例

梯形图	
语句表	AXISLOCK 2
描述	X2.0 导通时，锁住 2 号轴

PMC 轴相对移动 AXISMOVE

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		后置 ×

功能说明

PMC 轴是特殊的移动轴，该轴是不能被指令控制移动的，且不参与插补。它只能通过 PLC 程序来控制移动。该指令是用于指定该 PMC 轴移动的相对距离，并使其移动。

参数说明

参数 1：轴号；

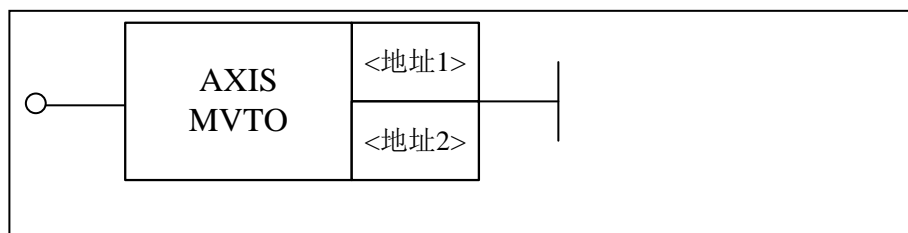
参数 2：轴移动量（单位 1/1000 毫米或 1/1000 度）；

示例

梯形图	
语句表	AXISMOVE 2 2
描述	X2.0 导通时，2 号轴相对移动 2 个单位的距离

PMC 轴绝对移动 AXISMVTO

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		后置 ×

功能说明 该指令用于控制 PMC 轴移动到一个绝对的位置。

参数说明 参数 1：轴号；

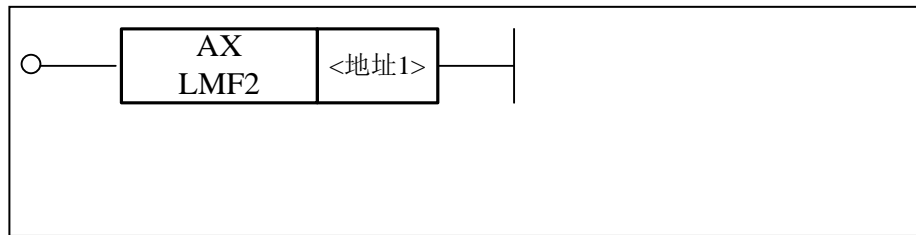
参数 2：轴移动位置（单位 1/1000 毫米或 1/1000 度）；

示例

梯形图	
语句表	AXISMVTO 2 2
描述	X2.0 导通时，2 号轴移动到 2 的位置

轴第 2 软极限 AXLMF2

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		前置 ✓
					后置 ×

功能说明 轴的第 2 软限位

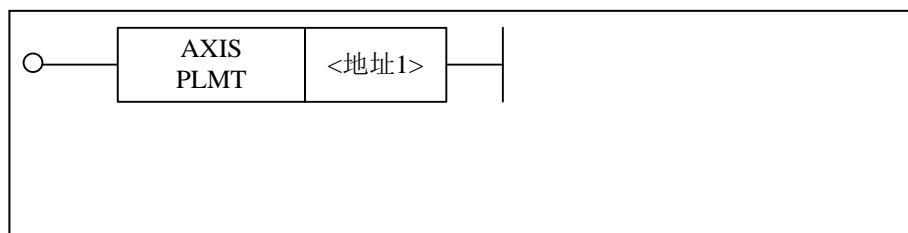
参数说明 参数 1：轴号；

示例

梯形图	
语句表	AXLMF2 2
描述	X2.0 导通时，2 号轴第 2 软限位有效

正限位挡块开关 AXISPLMT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 轴的正向限位。

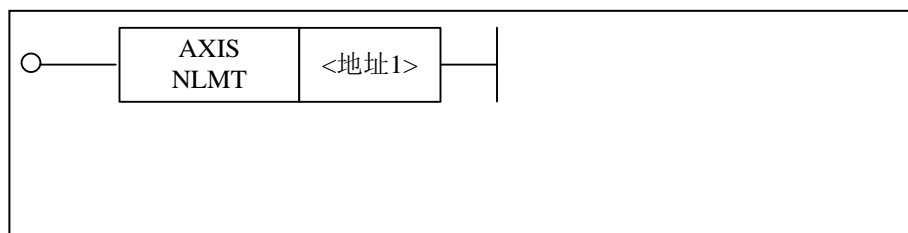
参数说明 参数 1：轴号。

示例

梯形图	
语句表	AXISPLMT 1
描述	X1.1 有效时表示轴 1 正限位

负限位挡块开关 AXISNLMT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 轴的负向限位。

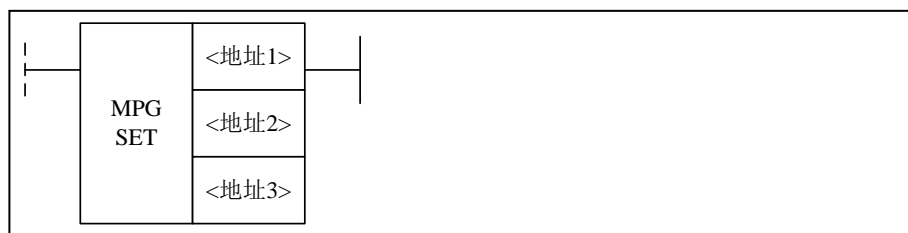
参数说明 参数 1：轴号。

示例

梯形图	
语句表	AXISNLMT 1
描述	X1.2 有效表示轴 1 碰到负限位点

手摇设置 MPGSET

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
<地址 2>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		后置 ×
<地址 3>	□□□□	INT	常数、X、Y、F、G、R、W、D、P、B		

功能说明

设置手摇。

参数说明

参数 1: 手摇号;

参数 2: 轴号;

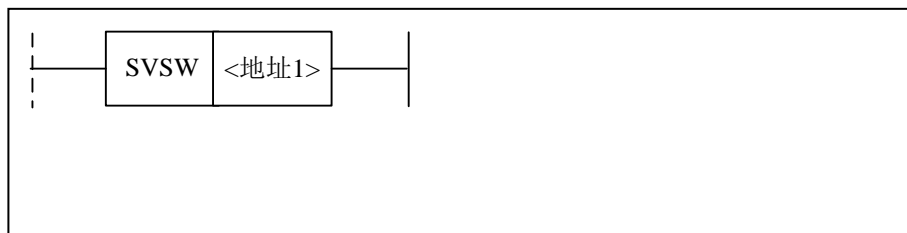
参数 3: 修调值;

示例

梯形图	
语句表	MPGSET 1 R6 R7
描述	1 号手摇获得其增量值，R6 中存放 1 号手摇选中轴号，R7 存放 1 号手摇倍率修调值

伺服使能【总线】SVSW

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
					后置 ×

功能说明 伺服使能。

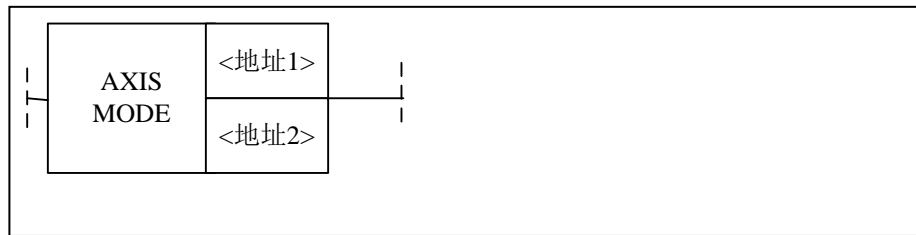
参数说明 参数 1：轴号。

示例

梯形图	
语句表	SVSW 1
描述	1 号轴伺服使能

轴工作模式 AXISMODE

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	常数		后置 ×

功能说明 轴工作模式选择。

参数说明 参数 1: 轴号;

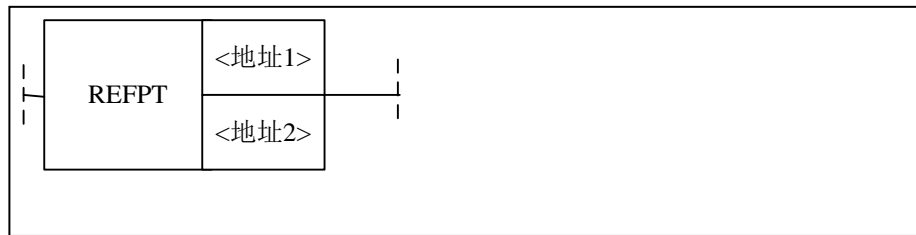
参数 2: “0” 为位置, “1” 为速度, “2” 为力矩;

示例

梯形图	
语句表	AXISMODE 1 1
描述	轴 1 工作模式选择为速度

轴参考点确认 REFPT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	常数		后置 ✓

功能说明 轴参考点确认。

参数说明 参数 1：轴号；

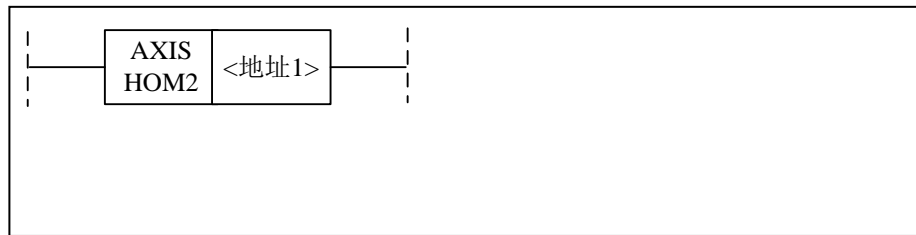
参数 2：“2”为第二参考点有效，“3”为第三参考点有效，“4”为第四参考点有效，“5”为第五参考点有效；

示例

梯形图	
语句表	REFPT 1 2
描述	轴 1 的第二参考点有效，输出 R10.1

轴回零过程中 AXISHOM2

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	轴号	前置 ○ 后置 ○

功能说明

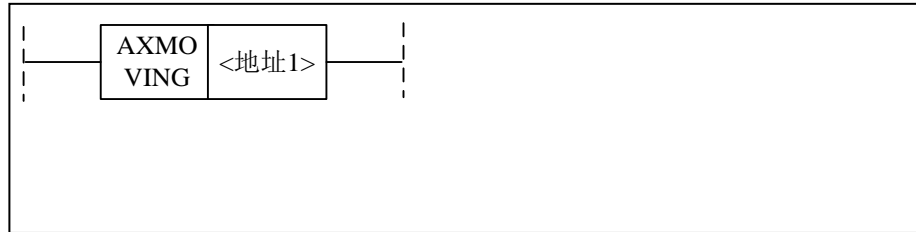
获取轴回零过程中的回零状态。在轴回零的过程中，不允许进行某些操作，此时需要判断该状态。对应 F 状态字 F0.2。

示例

梯形图	
语句表	AXISHOM2 0
描述	轴 0 的回零过程标志有效，轴正在回零过程中，输出 R1.1，此时不允许其他手动操作

轴移动过程中 AXMOVING

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	轴号	前置 ○ 后置 ○

功能说明

获取轴移动过程中的状态。在轴移动过程中，不允许进行某些操作，此时需要判断该状态。对应 F 状态字 F0.0。

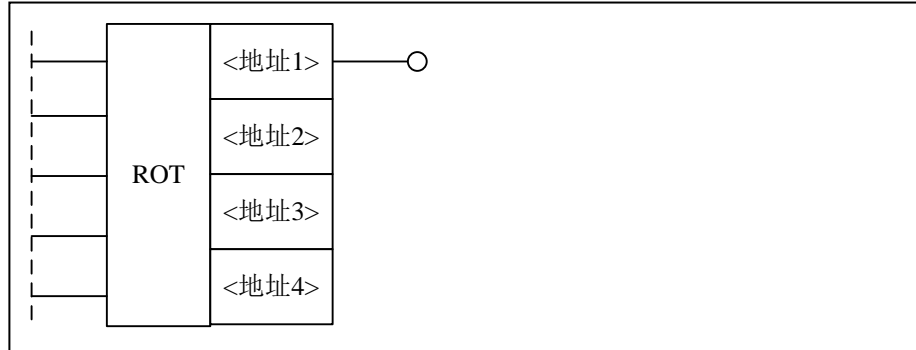
示例

梯形图	
语句表	AXMOVING 0
描述	轴 0 的移动标志有效，轴正在移动过程中，输出 R1.1

系统功能

旋转控制 ROT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
<地址 2>	□□□□	INT	X、Y、F、G、R、 W、D、P、B		后置 ✓
<地址 3>	□□□□	INT	X、Y、F、G、R、 W、D、P、B		
<地址 4>	□□□□	INT	X、Y、F、G、R、 W、D、P、B		

功能说明 旋转控制，用于刀架等。正向旋转输出 0，反向旋转输出 1。

参数说明 输入 1：使能通断。

输入 2：起始编号，为 0 时旋转设备的位置号由 0 开始；为 1 时旋转设备的位置号由 1 开始。

输入 3：是否就近选刀，为 0 时顺时针反向选刀，为 1 时就近选刀。

输入 4：目标位置类型，为 0 时计数目标当前位置，为 1 时计数目标前一位。

输入 5：计数结果类型，为 0 时计数位置数，为 1 时计数步数。

参数 1：刀架最大数量；

参数 2：当前位置地址；

参数 3：目标位置地址；

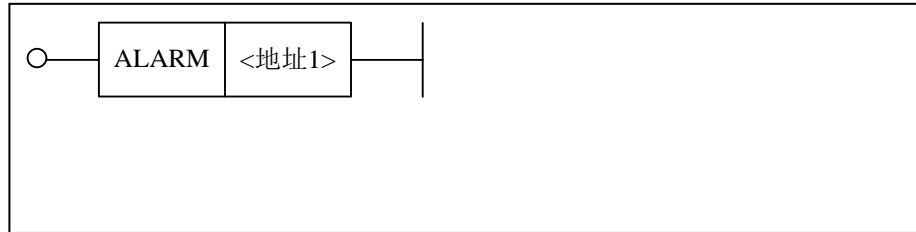
参数 4：计数结果地址，计数结果含义由输入 4 和输入 5 联合决定

示例

<p>梯形图</p>	
<p>语句表</p>	<p>ROT 16 B0 R27 R59</p>
<p>描述</p>	<p>R66.0 为常真触点，此时 ROT 模块的设置状态为，起始编号由 1 开始计数，使用短路径选刀功能，计数当前目标位置，输出计数步数，将计数的数值输出到 R59 中。B0 中保存当前刀位置，R27 中保存目标刀位置。</p>

报警设置 ALARM

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ○
					后置 ×

功能说明 产生报警。

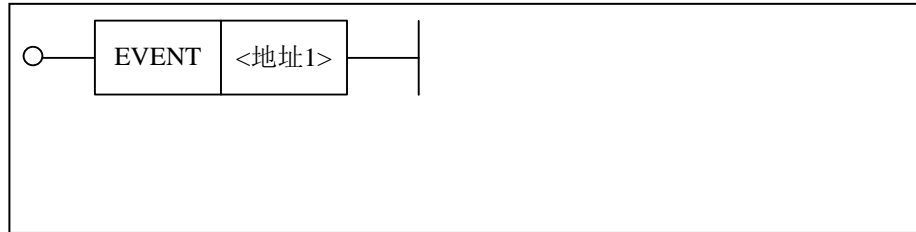
参数说明 参数 1: 报警号。PLC 报警号从 1~256, PLC 提示号从 501 到 884

示例

梯形图	
语句表	ALARM 3
描述	X3.4 导通时, 产生 3 号报警

事件设置 EVENT

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数		前置 ✓
					后置 ×

功能说明 产生事件对象。

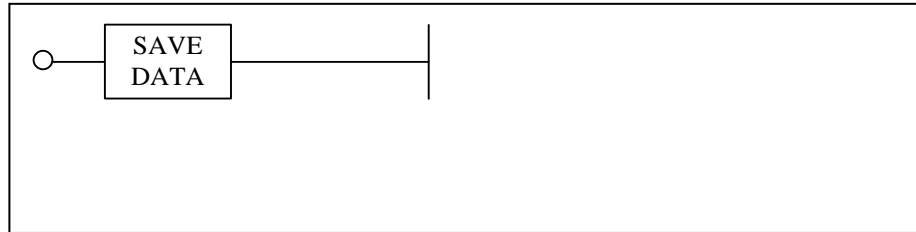
参数说明 参数 1：事件号。

示例

梯形图	
语句表	EVENT 122
描述	X30.4 导通时，产生 122 号事件

保存数据 SAVEDATA

格式



参数	参数格式	数据类型	存储区域	说明	属性
无					前置 ✓
					后置 ×

功能说明 保存所有数据。

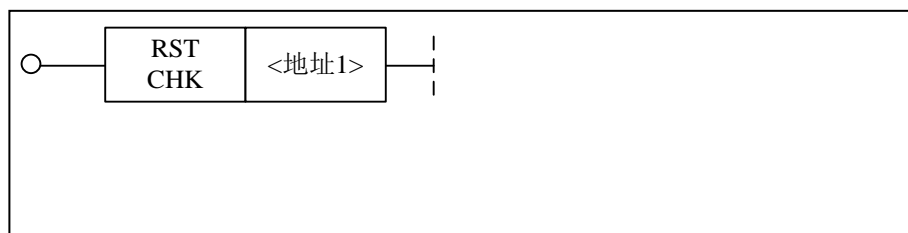
参数说明 无

示例

梯形图	
语句表	SAVEDATA
描述	X30.4 导通时，保存断电数据

复位设置输出 RSTCHK

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	通道号	前置 ✓ 后置 ✓

功能说明

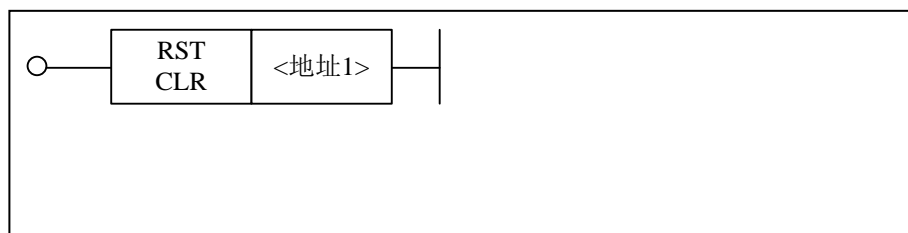
获取面板复位信号（必须和 RSTCLR 同时使用），当需要在面板复位键按下以后对 PLC 中执行一些复位动作时，通过该功能模块来获取。此时在数控系统界面上显示“复位中...”

示例

梯形图	
语句表	RSTCHK 0
描述	复位设置输出为 1。

复位清除 RSTCLR

格式



参数	参数格式	数据类型	存储区域	说明	属性
<地址 1>	□□□□	INT	常数	必须和 RSTCHK 同时使用	前置 ✓ 后置 ×

功能说明

当 PLC 中的复位动作全部完成以后必须复位清除（必须和 RSTCHK 同时使用），并将复位完成信号传递给数控系统，界面上将显示“复位完成”。

参数说明

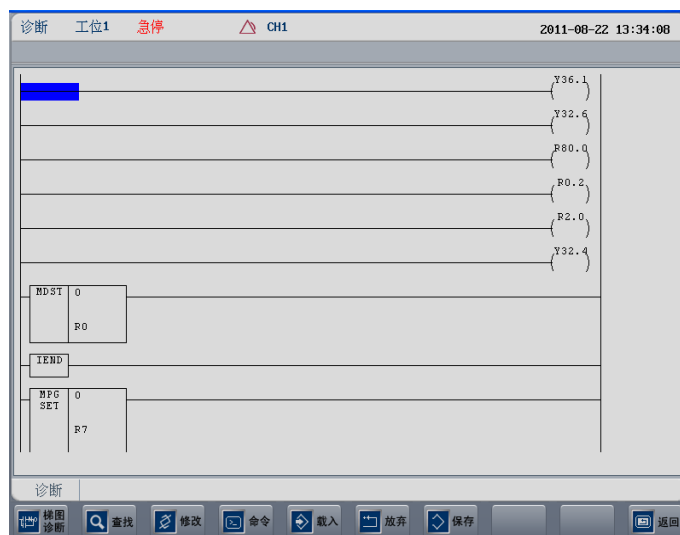
通道号

示例

梯形图	
语句表	RSTCLR 0
描述	X30.5 导通时，复位清除，输出为 0。

梯形图运行监控与在线修改

梯形图运行监控与在线修改功能是在数控系统的 PLC 编辑功能中提供的，它将实时监控着梯形图中每个元件的状态的改变以及可以通过强制的修改某个元件的状态来达到调试的目的。



按诊断操作界面中的“梯图监控”，即进入梯图监控操作界面，如上图所示。梯图监控操作界面上的按键包括梯图诊断、查找、修改、命令、载入、放弃、保存和返回。

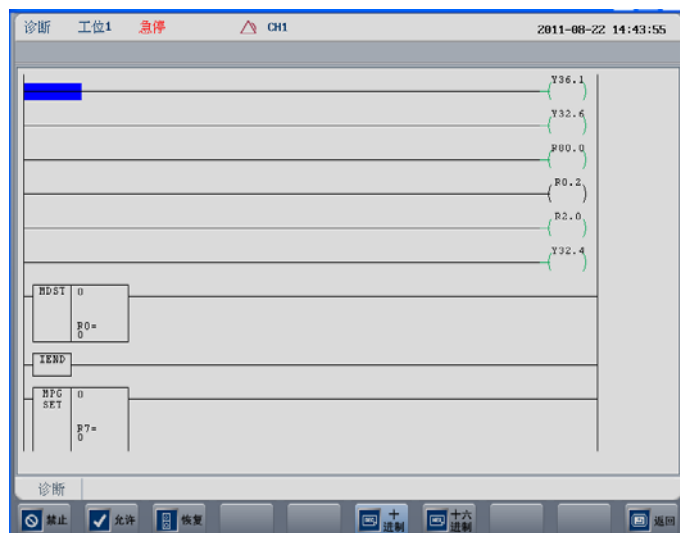
	梯图诊断：可查看每个变量的值，可对元件进行干预执行操作。
	查找：输入元件名，即可查找元件。
	修改：可对元件进行修改操作。
	命令：可对梯形图进行编辑。
	载入：载入当前梯形图信息。
	放弃：撤消对梯形图的编辑操作。
	保存：保存对梯形图的编辑操作。
	返回：返回到前一操作。

下面各节将对每个功能按键操作进行详细说明。

梯图在线诊断



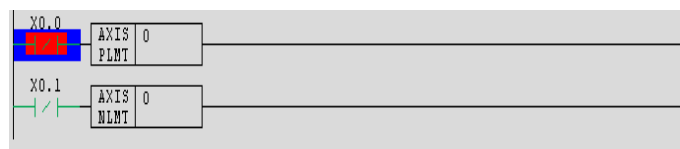
按梯图诊断功能键，即进入梯图诊断操作界面，如下图所示。梯图诊断操作界面包括禁止、允许、恢复、十进制、十六进制和返回 6 个按键。



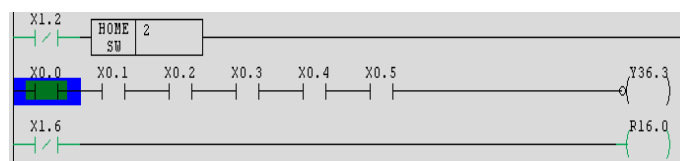
按“诊断→梯图监控→梯图诊断”，即可查看每个变量的值。用户可以上下移动光标查看每个变量的情况。如上图中，元件变为绿色代表该元件接通或者有效。用户可以元件进行禁止、允许、恢复等操作。



禁止功能键，将光标移到元件上，按下禁止功能键，即可以屏蔽该元件。如下图所示，光标移到元件上，按下禁止功能键后该元件变成红色，表示被屏蔽。如果要恢复元件功能，按下恢复键，该功能会在后面讲到。

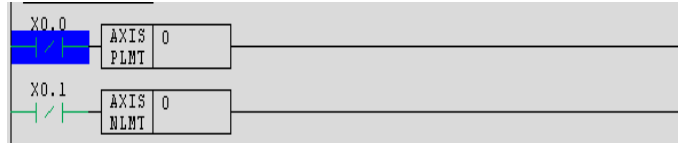


允许功能键，将光标移到元件上，按下允许功能键，即可以激活该元件。如下图所示，光标移到元件上，按下禁止功能键后该元件变成绿色，表示被激活。图中 X0.0 为常开，光标移到 X0.1 上后，按下允许功能键后，该元件变成绿色，由开变成闭。



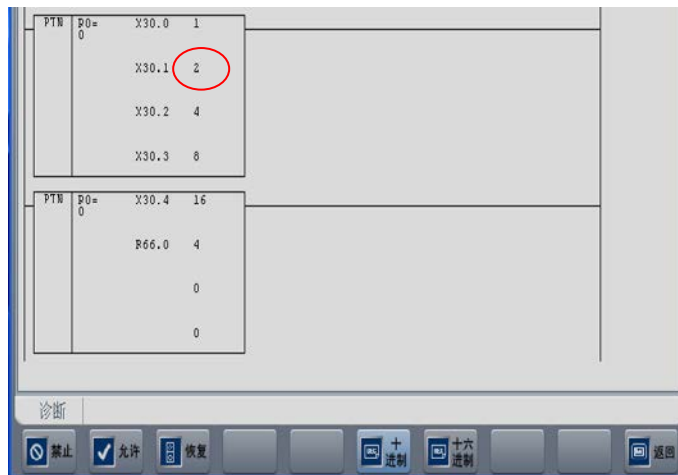


恢复功能键，将光标移到元件上，按下恢复功能键，即可以撤消上述屏蔽或激活元件的操作。禁止功能后按下恢复键，元件红色显示消失，表示恢复元件功能，如下图所示。

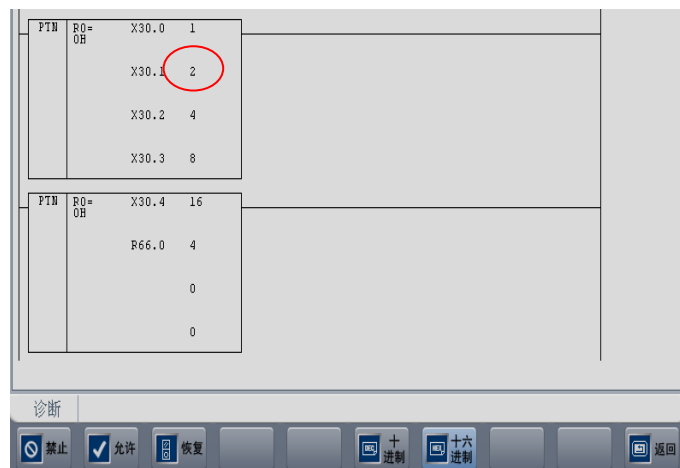


系统在默认情况下，系统显示的值以“十进制”表示，用户可以按“十六进制”对应的功能键，系统显示的值将以“十六进制”表示。如下图所示：

➤ 十进制显示



➤ 十六进制显示

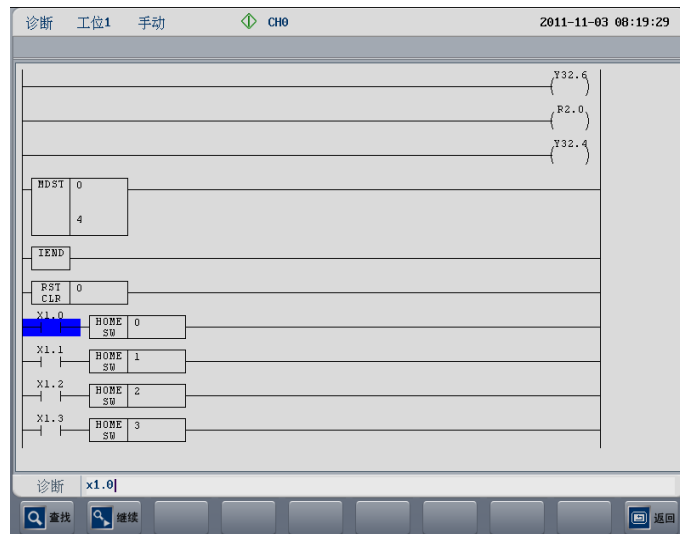


返回功能键，按下该功能键即返回到梯图监控操作界面，进行其他操作。

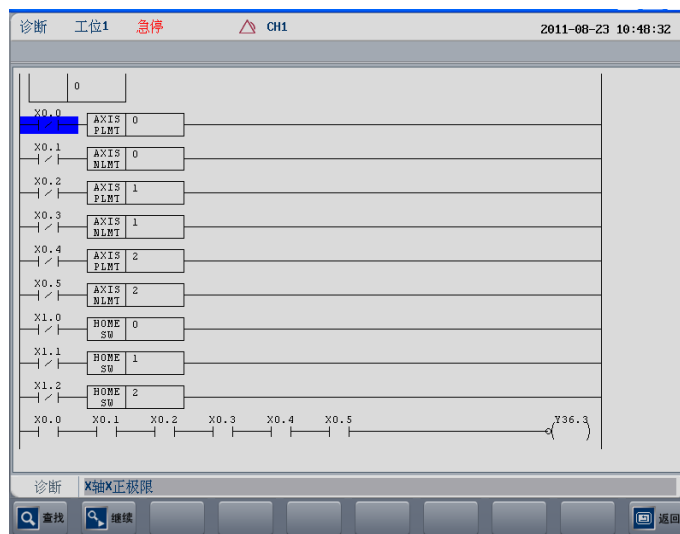
查找



按下查找功能键后，出现如下图所示的操作界面，可对元件进行查找。

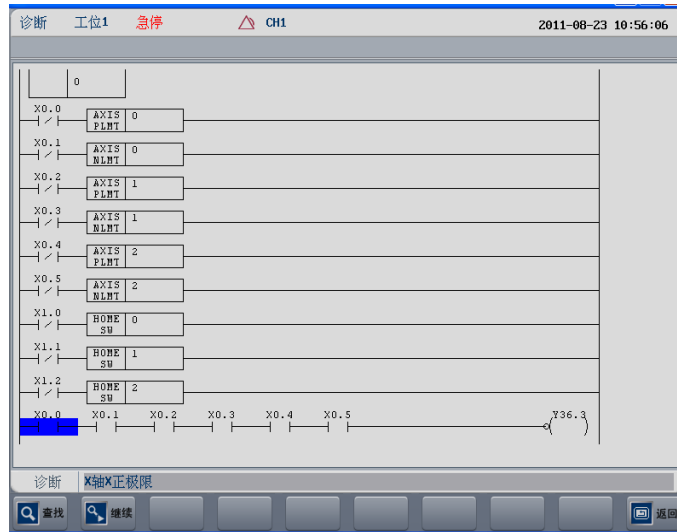


例如，输入 X0.0，按 Enter 键，就可查到光标行下程序中首个的 X0.0。如下图所示：





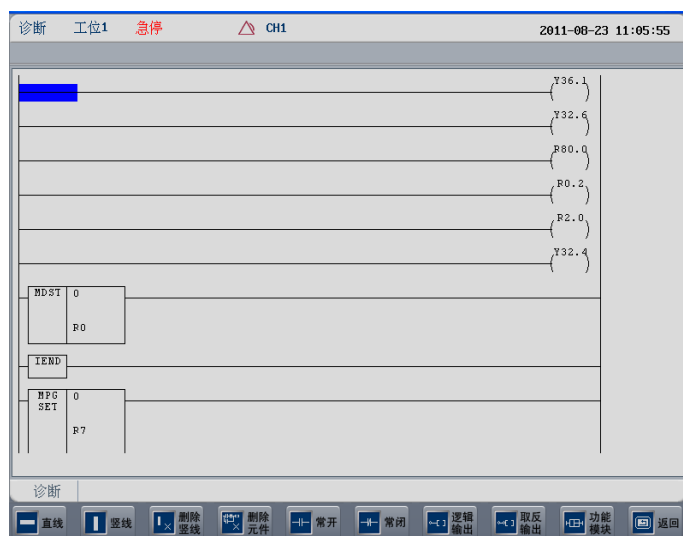
查找到的元件上有蓝色光标聚焦，如需继续查找，按继续功能键，系统即可查找到下一个同名的元件，如下图所示。



返回功能键，按下该功能键即返回到梯图监控操作界面，进行其他操作。

修改

用户可以按“修改”菜单对应的功能键，进行新建元件的操作：

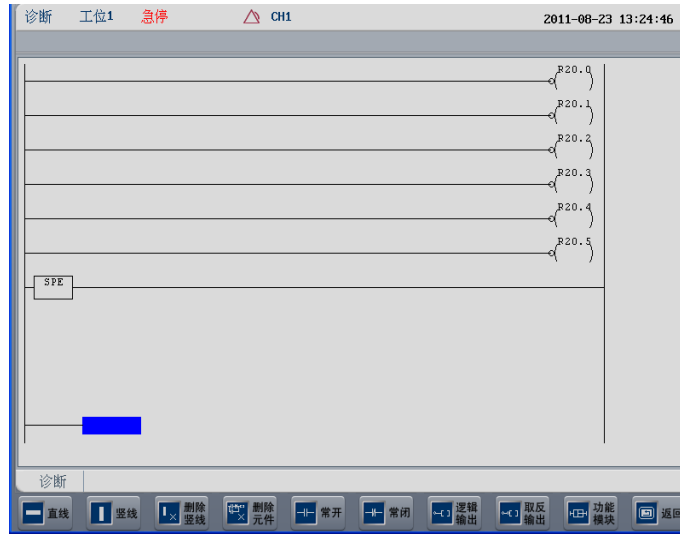


	直线：插入直线。
	竖线：插入竖线。
	删除竖线：删除竖线。
	删除元件：删除元件。
	常开：插入常开触点。
	常闭：插入常闭触点。
	逻辑输出：插入输出。
	取反输出：插入反输出。
	功能模块：插入功能(用户可以按元件的首写字母,可直接选择元件)。

插入直线



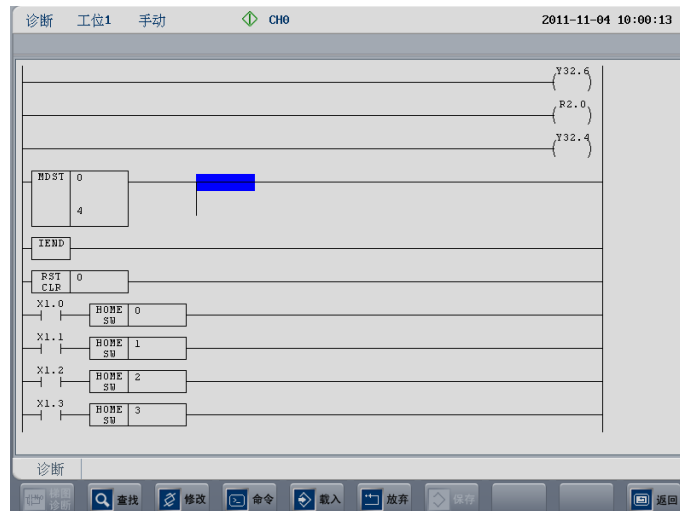
按直线功能键，可以在梯形图中插入了一条直线，如下图所示：



插入竖线



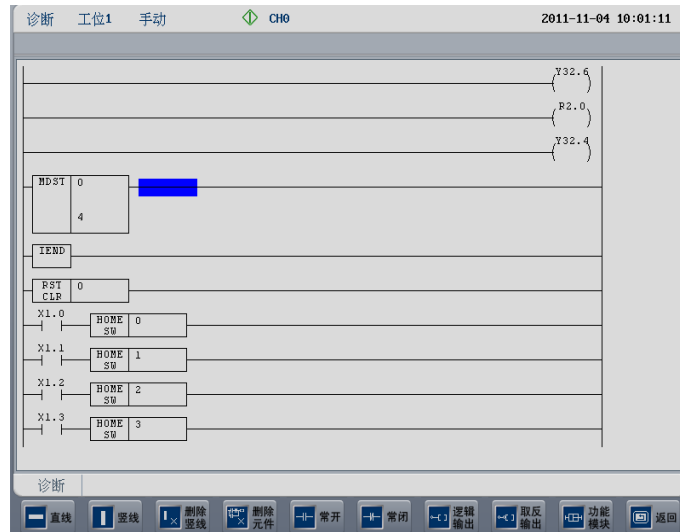
按竖线功能键，可以在光标后插入了一条竖线，如下图所示：



删除竖线



按删除竖线功能键，可以删除光标后的竖线，如下图所示。

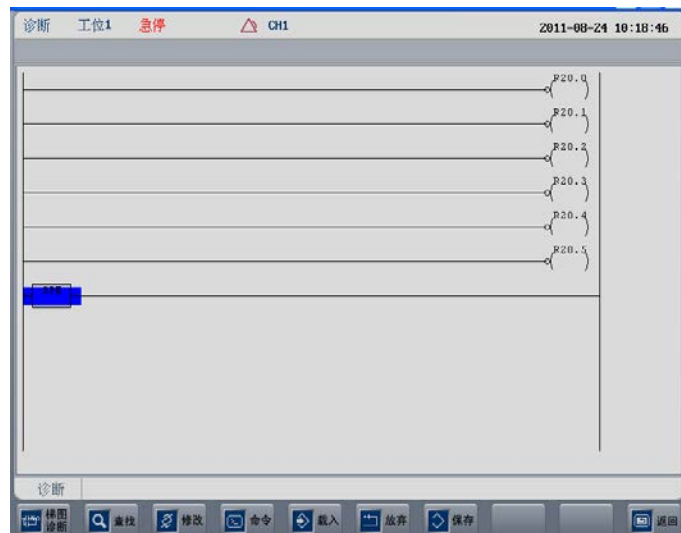


删除元件

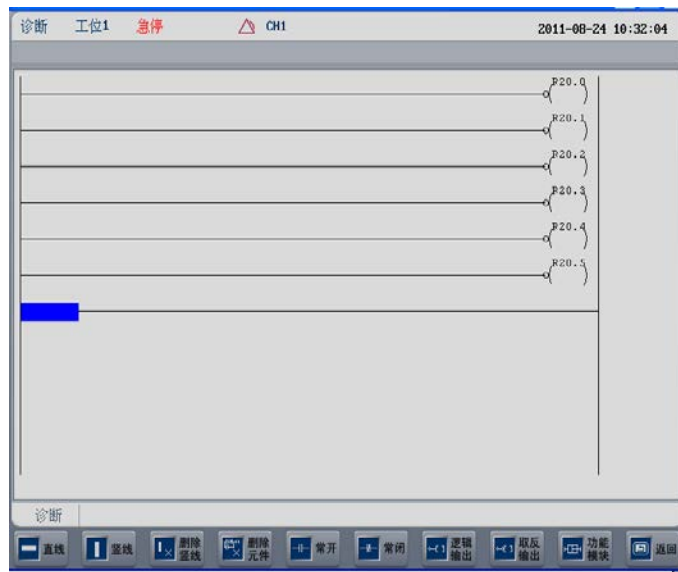


将光标移动到需要删除的元件上，按删除元件功能键，可以删除梯形图中的元件。

删除前



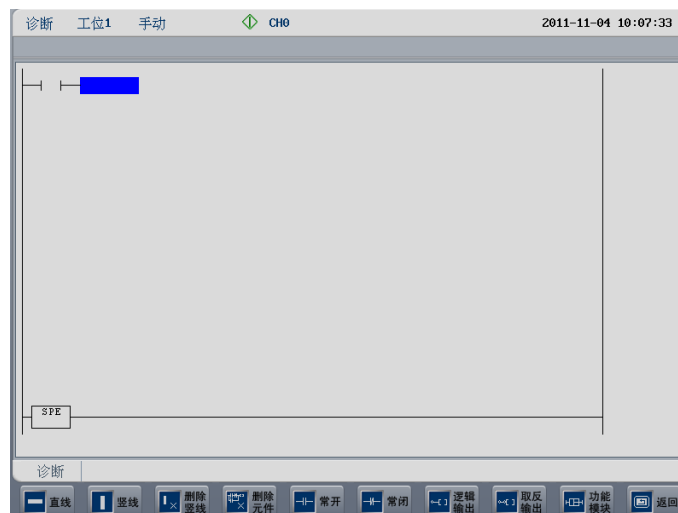
➤ 删除后



常开



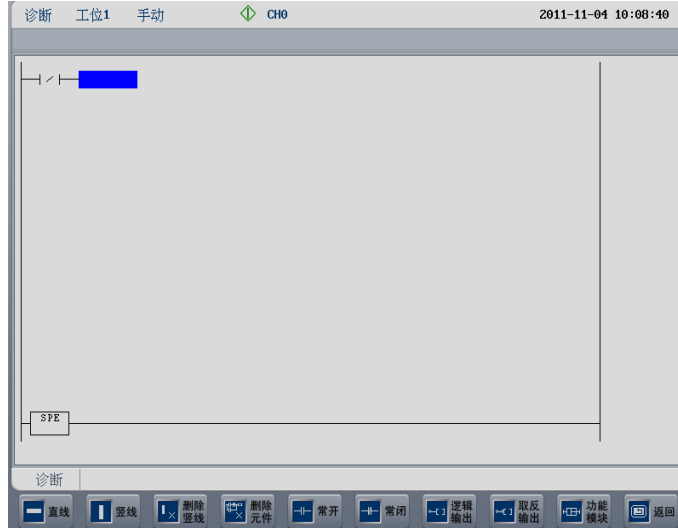
将光标移动到需要插入常开的位置处，按常开功能键，可以在梯形图中指定的位置插入常开，如下图所示：



常闭



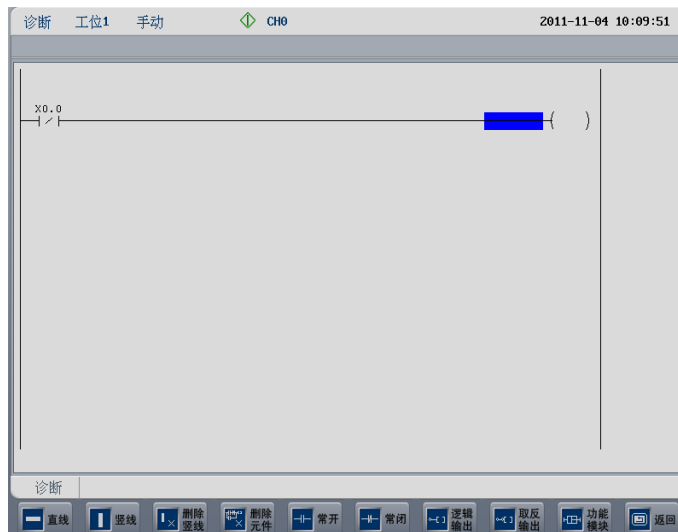
将光标移动到需要插入常开的位置处，按常闭功能键，可以在梯形图中指定的位置插入常闭，如下图所示。



逻辑输出



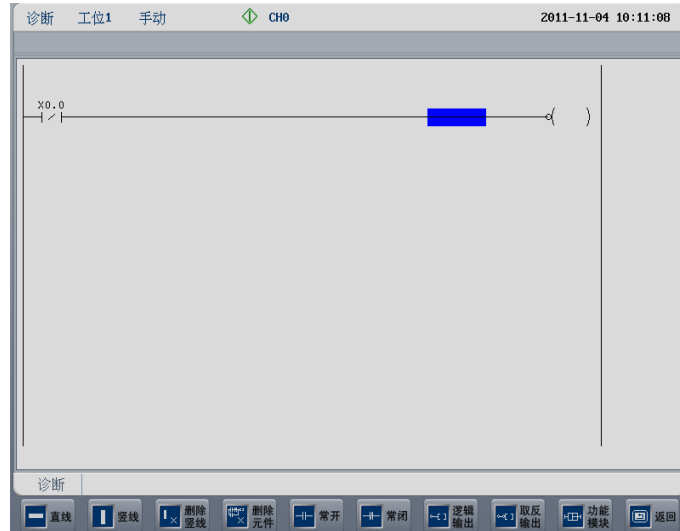
将光标移动到需要插入逻辑输出的位置处，按逻辑输出功能键，可以在梯形图中指定的位置插入逻辑输出，如下图所示。需要说明的是逻辑输出可以加前置，不可以加后置，详细说明请参看编程说明部分。



取反输出



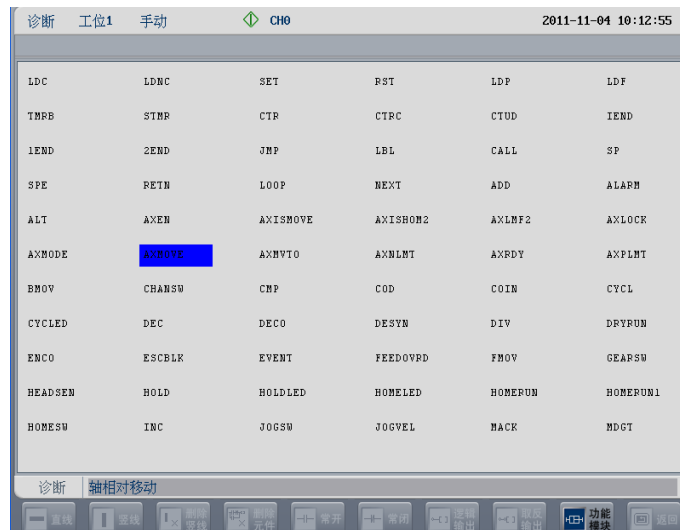
将光标移动到需要插入取反输出的位置处，按取反输出功能键，可以在梯形图中指定的位置插入取反输出，如下图所示。需要说明的是逻辑输出可以加前置，不可以加后置。



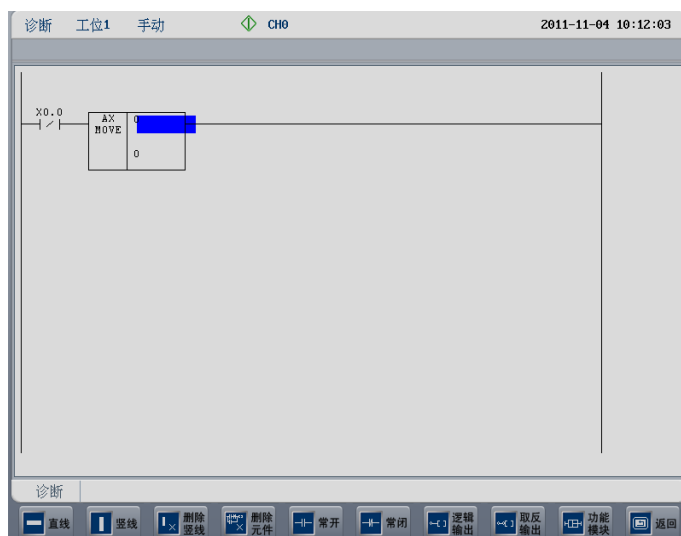
功能模块



按功能模块键进入如下图所示的操作界面，选择需要的功能模块：



然后按 **Enter** 键即可在梯形图中输入所选功能模块，用户可以按元件的首写字母，可直接选择元件，如下图所示。



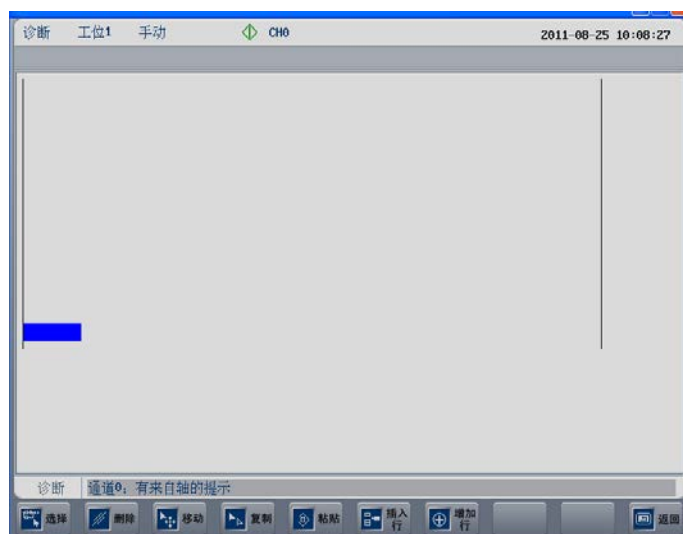
再次按下功能模块键时即可返回到修改操作界面。








返回

按返回功能键，即返回上级操作界面，进行其它操作。

命令

用户可以通过按以下按键，进行编辑梯形图；

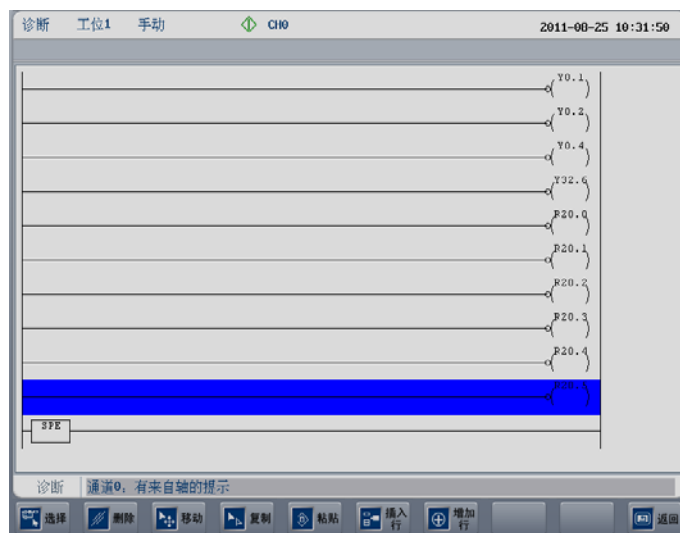


 选择	选择：选择光标所在行
 删除	删除：删除光标所在行
 移动	移动：移动用户所选的元件
 复制	复制：复制用户所选的元件
 粘贴	粘贴：粘贴用户所选的元件
 插入行	插入行：在光标所在行之前插入一行
 增加行	增加行：在光标所在行之后插入一行

选择



将光标移到想要选择的行，然后按选择功能键，所选择的行变为蓝色，再次按下选择键，将选择当前行的下一行。如下图所示。选择所选的行可以进行删除等后续操作。

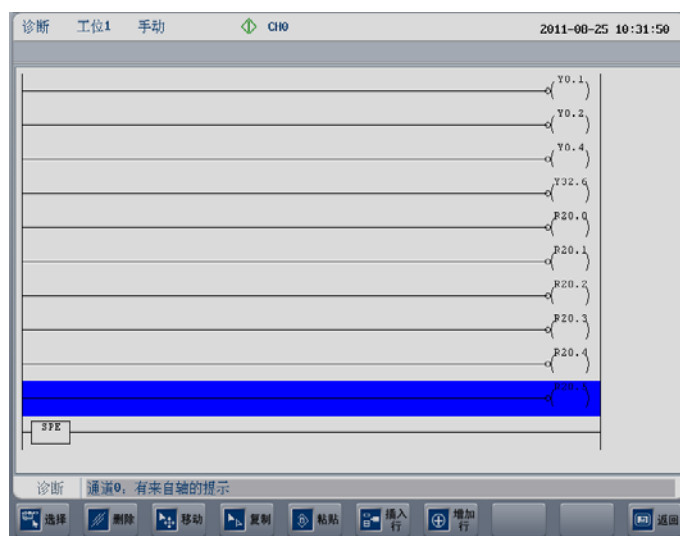


删除

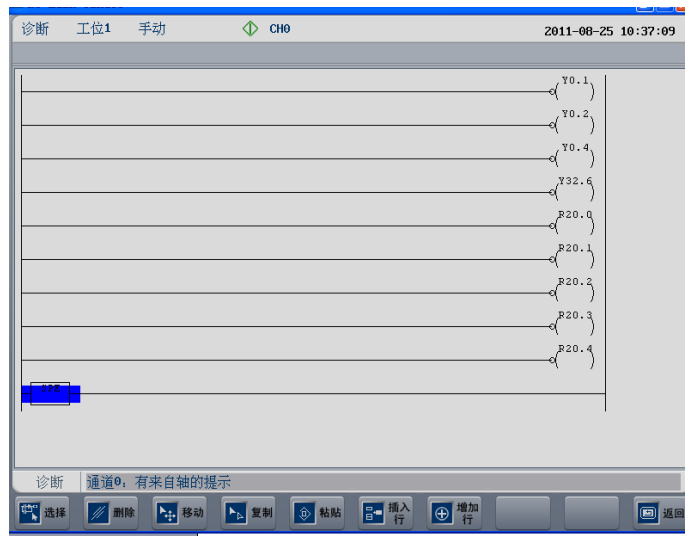


用选择功能键选择需要删除的行，将光标移向需要删除的行，选择该行，颜色变成蓝色，然后按删除功能键即可删除所选行，如下图所示。

➤ 删除前



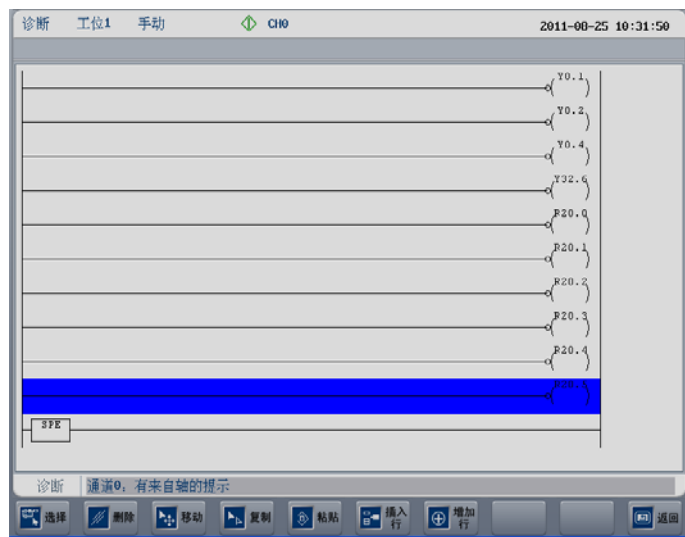
➤ 删除后



移动

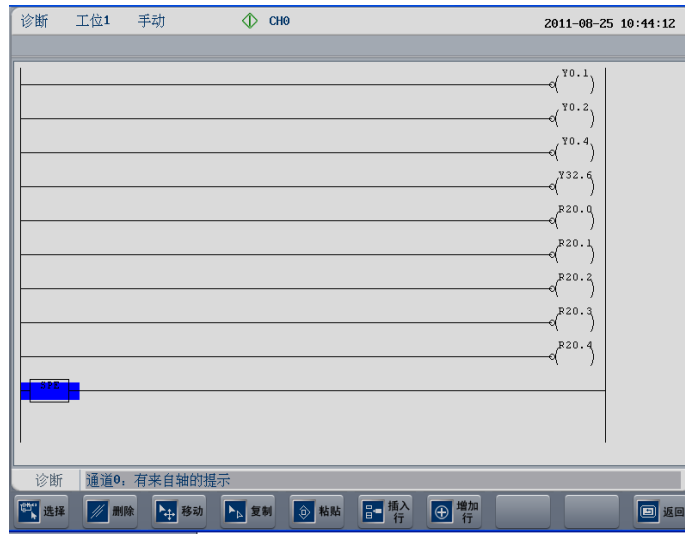


需要移动行，首先将光标移到需要移动的行，按选择功能键，该行变为蓝色，如下图所示。

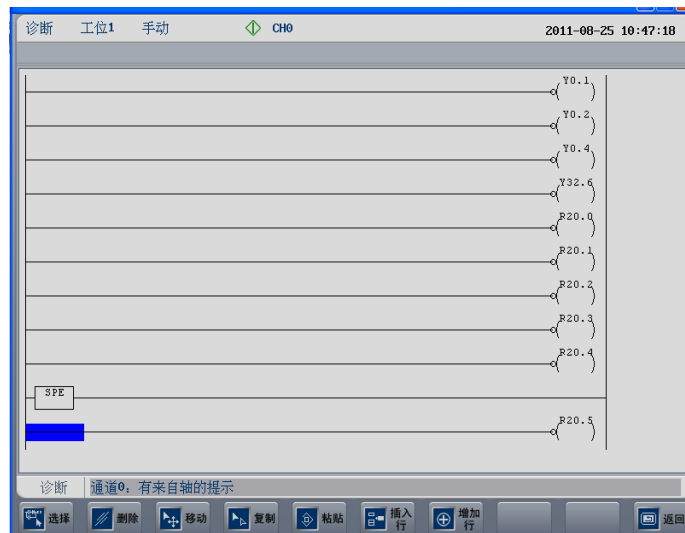




然后按移动功能键，此时界面如下图所示，所选的行消失



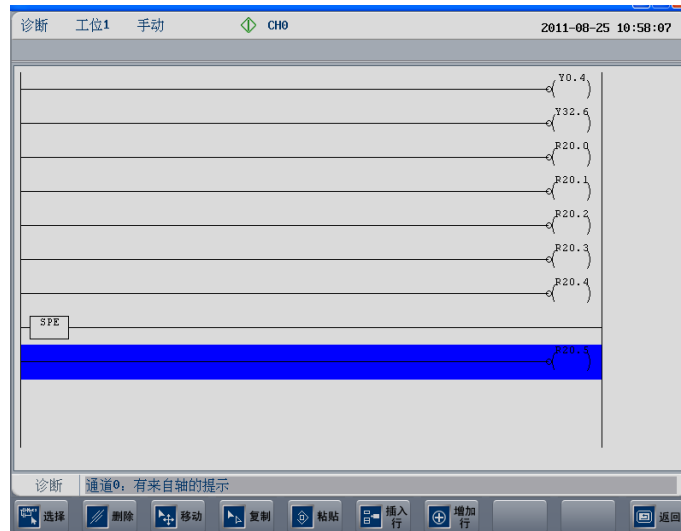
将光标移到目标行，按粘贴功能键，即可以移动到目标行，如下图所示：



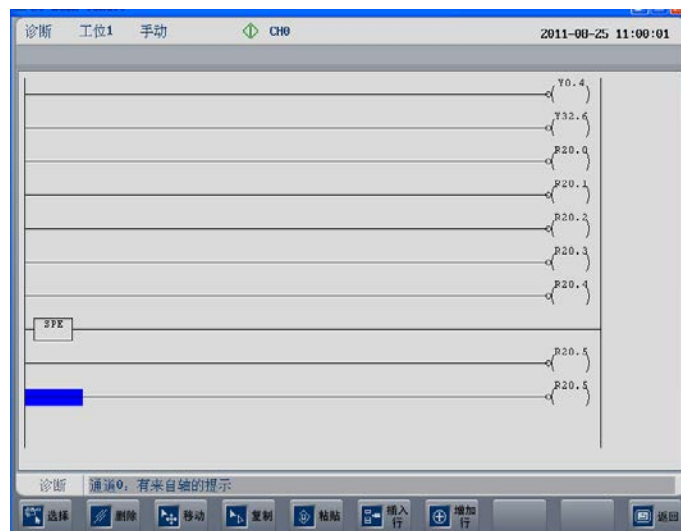
复制



将光标移到需要复制的行所在位置，按选择功能键后，再按复制功能键，如下图所示，



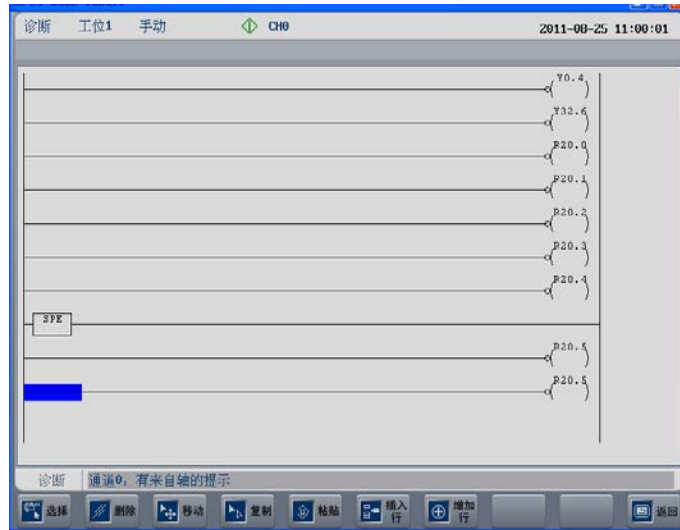
然后将光标移到目标行，按粘贴功能键即完成复制功能：



粘贴



粘贴功能键已经在 7.4.3 和 7.4.4 节中用所运用，详细操作请参考以上两节。

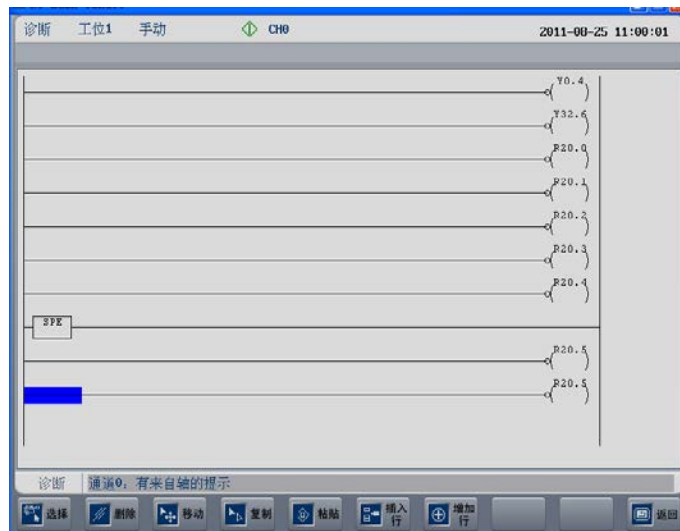


插入行

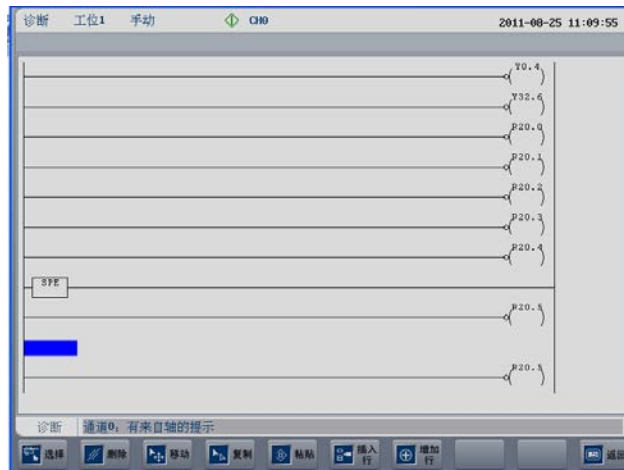


将光标移到需要插入行的下一行，按插入行功能键，即可以插入行，操作如下图所示。需要注意的是，插入行一般是插入光标所在行的上方。

➤ 插入前



➤ 插入后

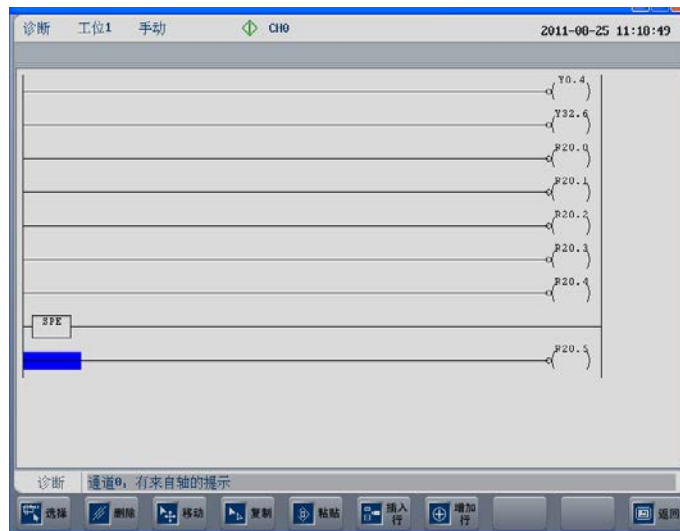


增加行

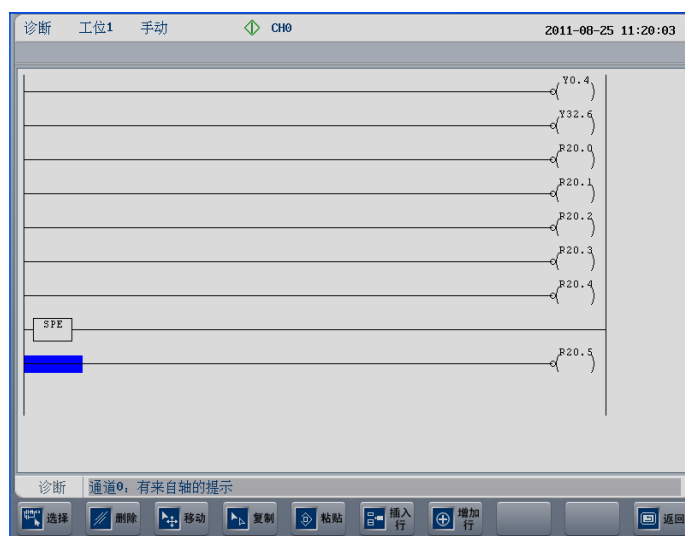


增加行与插入行相反的是增加的行是加在光标所在位置的下方，如下图所示，按增加行功能键后，光标所在的行下方增加了一行。

➤ 增加前



➤ 增加后



返回

按下返回功能键后返回上级操作界面，进行其它操作。

载入



梯形图编写完成后，经过核对无误后，按下载入功能键后，系统即载入当前梯形图。

放弃



编辑梯形图后如果需要重新编辑时，可以按放弃功能键，就可以撤消对梯形图的编辑操作。

保存



保存梯形图后，按保存功能键，可保存对梯形图的编辑操作。

返回

返回功能键，即返回到诊断操作界面。

PLC 开发环境使用说明

本章主要内容:

7.1 简介

7.2 开发环境的安装

7.3 开发环境界面

7.4 开发环境操作

简介

HNC-LADDER-WIN 数控梯形图编程软件是为华中数控 8 型系统最新研发的 PLC 程序开发环境软件。该软件在 WindowsXP 操作系统中运行，通过可视化图形编程方便地进行梯形图开发。兼容多种标准 PLC 语言，并符合 IEC61131-3 国际标准，是一种简捷、高效、可靠的 PLC 开发工具。

开发环境的安装

以 WindowsXP 中文版下用光盘完全安装梯形图开发环境为例说明梯形图开发环境的安装。

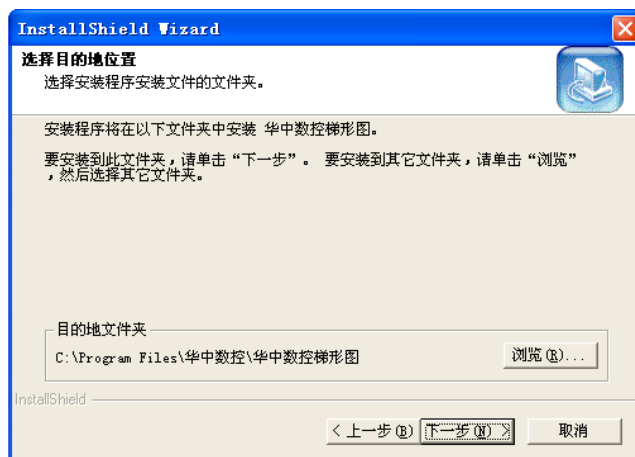
- (1) 进入 WindowsXP 中文版的操作环境。
- (2) 在光盘驱动器中插入含梯形图开发环境软件的光盘。
- (3) 双击光盘华中数控梯形图目录下的 Setup.exe 文件, 安装程序将自动运行, 稍后会出现一个安装向导。



- (4) 安装向导界面后, 将出现一个欢迎信息窗口。



- (5) 单击“下一步 (N) >”按钮后, 屏幕上出现目录选择对话框



- (6) 在目录选择对话框中，可对安装的路径进行必要的修改，然后点击“下一步 (N)”按钮。



- (7) 屏幕中出现上图所示画面，开始正式安装；



- (8) 安装完毕后，屏幕上出现安装完成提示框。

开发环境界面

梯形图开发环境被划分为四个部分：菜单、梯形图、语句表和符号表。

菜单

在开发环境顶部的一水平的长条称为菜单，在菜单里列出了梯形图界面的各个下拉菜单项。用鼠标单击某个菜单项，便会显示出下拉菜单中的各个命令选项，用鼠标单击某个命令，就可完成相应的操作。

开发环境菜单分为文件、编辑、查看、工具、窗口和帮助六项。下面分别介绍这些菜单项的具体内容。

文件

在“文件”菜单中包含用于对文件进行操作的命令选项。主要为用户提供对梯形图文件的各种操作。

新建	该选项用于创建新的工程。
打开	该选项用于打开已有的 dft 文件
保存	该选项用于把当前窗口的文件内容保存为 dft 文件
另存为	该选项的功能与“保存梯图”选项类似，也是保存打开的文件，不过该选项是将打开的文件用新的文件名加以保存
关闭	该选项用于关闭当前梯形图界面
载入 dit 文件	该选项用于打开已有的 dit 文件
打印	该选项用于打印当前窗口中的内容
打印预览	该选项用于查看将要打印出来的效果
打印设置	该选项用于设置打印的参数
退出	选择该选项将退出程序

编辑

编辑包含了复制粘贴等快速操作方式，编辑中的各种功能是为了提高用户编写梯形图的效率，主要用于用户编写梯形图时。

剪切	剪切字符串或元件
复制	复制字符串或元件
粘贴	粘贴字符串或元件
插入行	在光标所在位置插入行
删除行	删除光标所在行

查看

查看项用于控制在主窗口中显示的子窗口。

梯形图	打开（关闭）梯形图视图
语句表	打开（关闭）语句表视图
符号表	打开（关闭）符号表视图
图元树	打开（关闭）左边的图元树
消息框	打开（关闭）下边的消息框
工具栏	打开（关闭）工具栏
状态栏	打开（关闭）状态栏

工具

查找替换功能。

搜索	该选项用于查找指定的字符串
搜索下一个	继续查找指定的字符串
替换	该选项用于替换指定的字符串

窗口

选项中的内容是用于打开各个窗口。

层叠	以层叠方式显示所有子窗口
平铺	以平铺方式显示所有子窗口
REG	显示符号表窗口
STL	显示语句表窗口
LADDER	显示梯形图窗口

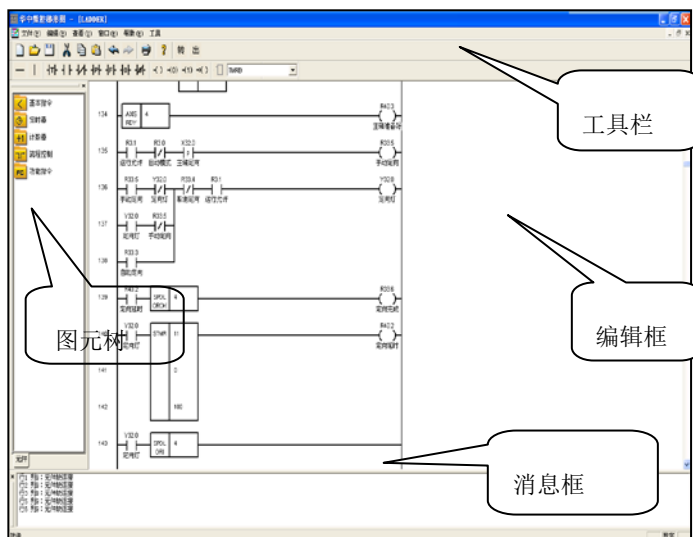
帮助

关于 NEWPLC：显示软件版本。

梯形图界面

在梯形图界面中包括工具栏、图元树、编辑窗口和消息框部分。

工具栏和图元树都可以随意停靠。也就是说它们可以放置在主窗口的4个侧边的任意一个上。也可以使工具栏“浮”在桌面上的任何位置。



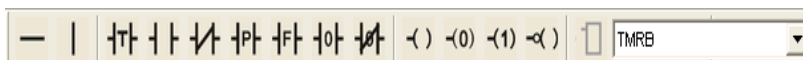
工具栏

在梯形图界面中种包括操作和元件两个工具栏：

- (1) 操作工具栏用于快捷的操作新建文件，放大缩小，撤销恢复等菜单中的操作。



- (2) 元件工具栏用于快捷加入基本输入输出单元和选择功能模块



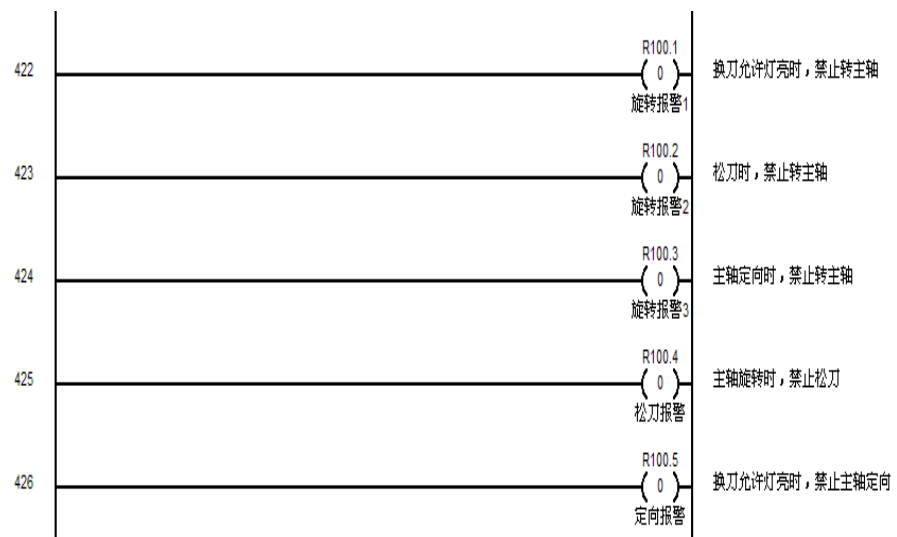
图元树

图元树用于选择功能模块。通过双击图标来展开或收起指令树，然后从指令树中选取需要使用的指令图标。



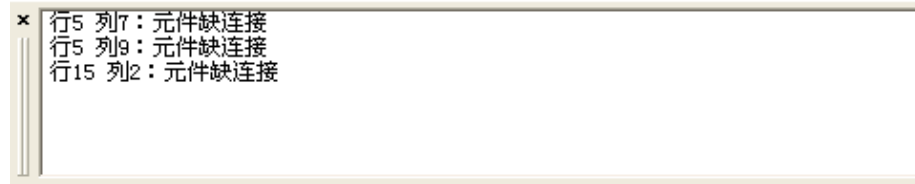
编辑窗口

编辑窗口用于显示和编辑梯形图。左右母线之间的区域是梯形图的编辑区，左母线左侧显示当前编辑的行号，右母线的右侧显示的是当前行输出状态含义的注释。



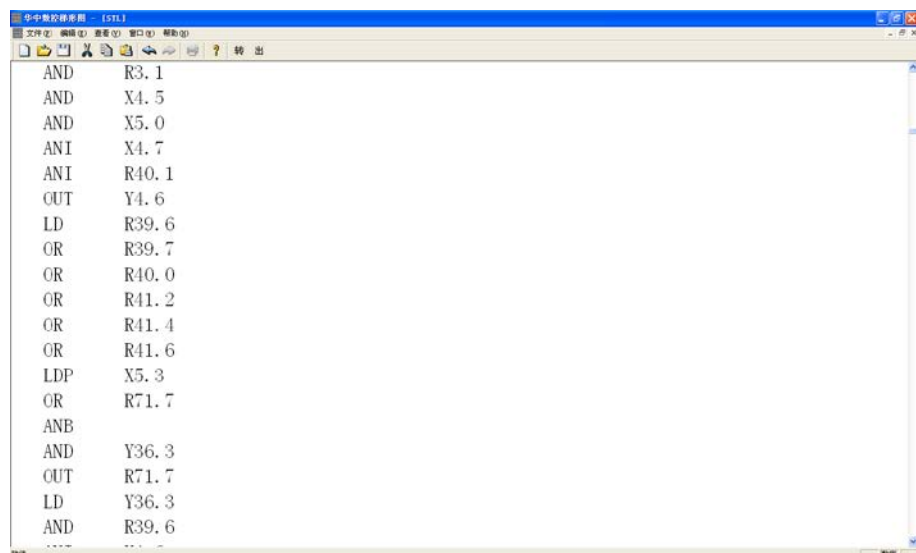
消息框

在编译转换梯形图的时候,如果在梯形图中存在着语句错误或者可以识别的语法错误,这时就需要消息框是用来显示转换、输出时出现的错误。



语句表界面

在语句表界面中包括工具栏和编辑窗口部分。



工具栏

在语句表界面中种包括一个操作工具栏。



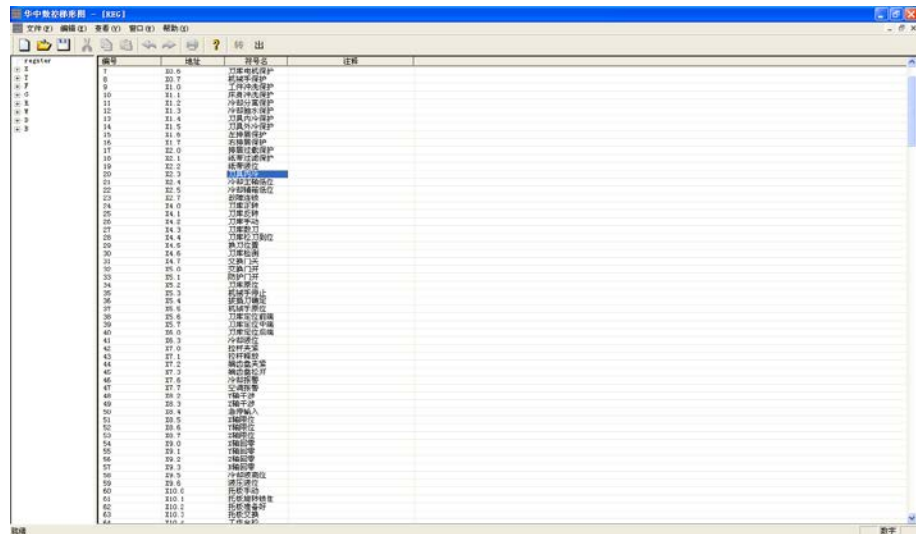
编辑窗口

编辑窗口用于显示和编辑语句表,并且可以在编辑语句表的时候为当前行判断错误。

× LDT x3

符号表界面

符号表界面用于定义相应地址的符号名和注释。



符号表编辑窗口左边为寄存器选择框，右边为寄存器编辑框。

在寄存器编辑框中包括编号、地址、符号名和注释四部分。

- 编号：显示当前符号名在所有符号名中的编号，自动生成。
- 地址：指定的地址。
- 符号名：指定地址所对应的符号名。
- 注释：指定地址所对应的注释

开发环境操作

在开始编辑 PLC 前，首先要在符号表界面中对要用到的地址定义符号名，并对其进行注释。然后再使用梯形图或语句表方式进行编辑 PLC。

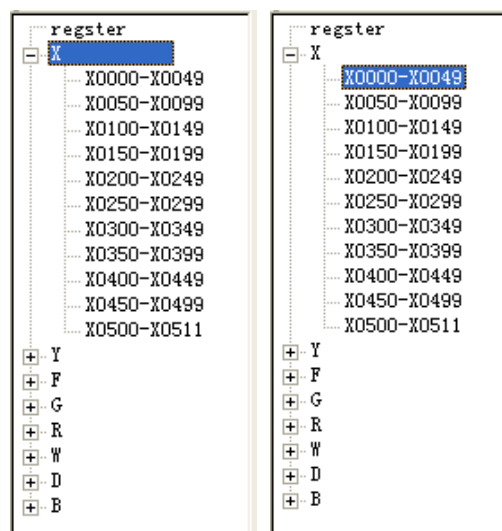
符号表操作

符号表用于对指定地址定义符号名和注释。

增加符号表

下面以 X10.0（X 轴正限位）为例。

X10.0 在 X 寄存器中，首先在寄存器选择框中选中 X 寄存器。X10.0 在 X000-X0049 中，选择分栏项。



在寄存器编辑框中将会显示 X0000 到 X0049 的所有寄存器位和寄存器点。

编号	地址	符号名	注释
	X0		
	X0.0		
	X0.1		
	X0.2		
	X0.3		
	X0.4		
	X0.5		
	X0.6		
	X0.7		
	X1		
	X1.0		
	X1.1		
	X1.2		
	X1.3		
	X1.4		
	X1.5		
	X1.6		
	X1.7		
	X2		
	X2.0		
	X2.1		
	X2.2		
	X2.3		
	X2.4		
	X2.5		
	X2.6		
	X2.7		
	X3		
	X3.0		
	X3.1		
	X3.2		
	X3.3		
	X3.4		

找到 X10.0 的地址，在符号名项上点击 3 次，将弹出编辑框。

编号	地址	符号名	注释
	X9.2		
	X9.3		
	X9.4		
	X9.5		
	X9.6		
	X9.7		
	X10		
	X10.0	<input type="text"/>	
	X10.1		
	X10.2		
	X10.3		
	X10.4		
	X10.5		
	X10.6		
	X10.7		
	X11		
	X11.0		
	X11.1		
	X11.2		
	X11.3		
	X11.4		
	X11.5		
	X11.6		
	X11.7		
	X12		
	X12.0		
	X12.1		
	X12.2		

在编辑框中输入“X 正限位”，然后点击回车键。

输入符号名后，再对此地址进行注释。在注释项上点击 3 次，将弹出编辑框。

编号	地址	符号名	注释
	X9.2		
	X9.3		
	X9.4		
	X9.5		
	X9.6		
	X9.7		
	X10		
0	X10.0	X正限位	<input type="text"/>
	X10.1		
	X10.2		
	X10.3		
	X10.4		
	X10.5		
	X10.6		
	X10.7		
	X11		
	X11.0		
	X11.1		
	X11.2		

在编辑框中输入“X 正限位，高电平有效”，然后点击回车键。

编号	地址	符号名	注释
	X9.2		
	X9.3		
	X9.4		
	X9.5		
	X9.6		
	X9.7		
	X10		
0	X10.0	X正限位	X正限位，高电平有效
	X10.1		
	X10.2		
	X10.3		
	X10.4		
	X10.5		
	X10.6		
	X10.7		
	X11		
	X11.0		
	X11.1		
	X11.2		
	X11.3		

对 X10.0 定义符号名和注释就完成了。

删除符号表

当不需要 X10.0 的符号名和注释时，就需要删除它。

首先在地址项上选中 X10.0，按下 Delete 键，就可删除此项了。

编号	地址	符号名	注释
	X9.2		
	X9.3		
	X9.4		
	X9.5		
	X9.6		
	X9.7		
	X10		
0	X10.0	X正限位	X正限位，高电平有效
	X10.1		
	X10.2		
	X10.3		
	X10.4		
	X10.5		
	X10.6		
	X10.7		
	X11		
	X11.0		
	X11.1		
	X11.2		
	X11.3		

梯形图操作

梯形图是由行组成，每行最多有 10 个单元。

插入元件

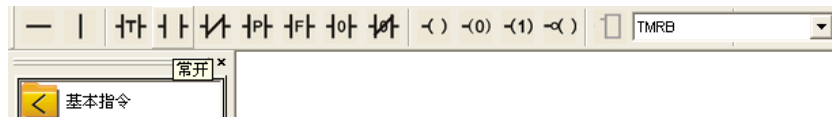
插入元件分为两种方式，一种为插入基本元件，另一种为插入功能元件。

➤ 插入基本元件

(1) 插入基本元件时，首先在梯形图上选中位置；



(2) 在工具栏上单击要加入的基本元件；

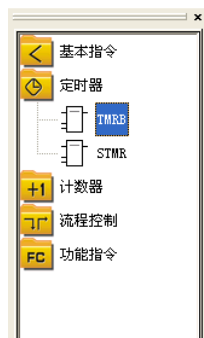


(3) 基本元件被加入到梯形图中；

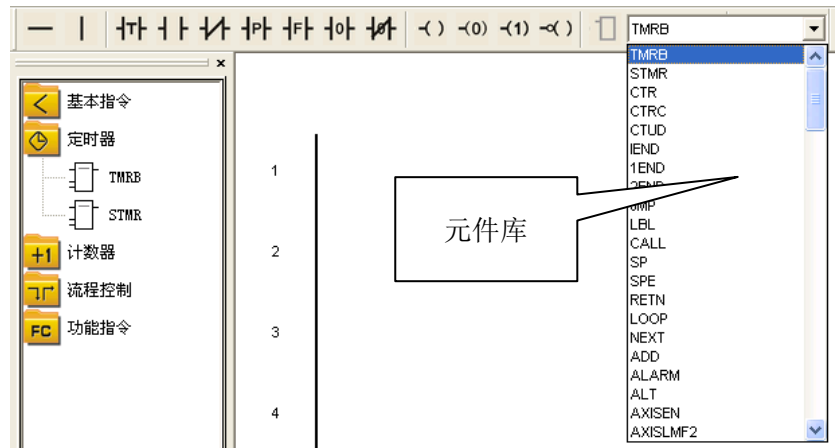


➤ 插入功能元件

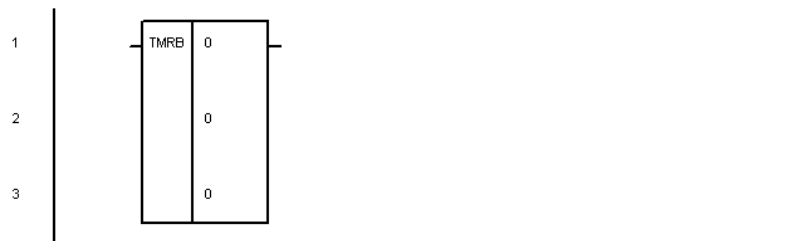
(1) 插入功能元件时，首先要选中需要加入的功能元件，选择需要加入的功能元件，可以在图元树中选择；



也可以在工具栏的功能元件，选择框中选择；

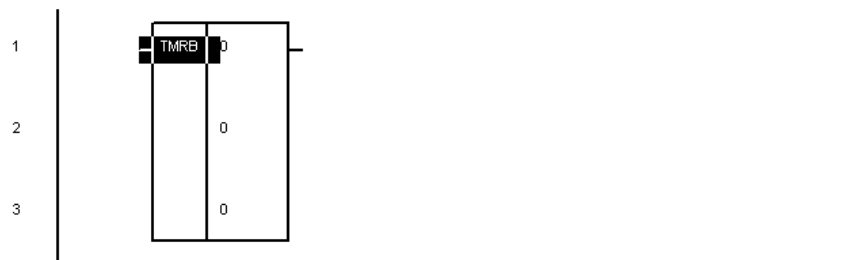


(2) 在梯形图中鼠标左键双击，就可以加入功能元件；



删除元件

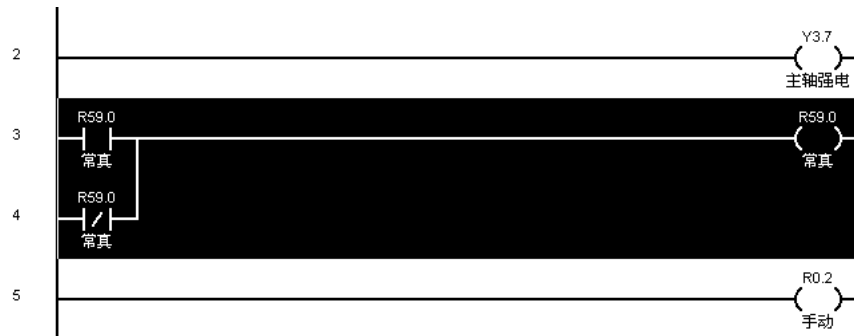
删除元件先要在梯形图中选中要删除的元件。



按下 Delete 键就可以删除选中的元件。

删除多行

删除多行先要选中需要删除的行。（使用鼠标拖动选择区域）



按下 Delete 键就可以删除选中的区域。

剪切、复制和粘贴元件

剪切、复制元件时，首先在梯形图中选择一个元件。



然后再在“编辑”菜单中选择剪切、或复制项。也可以在要剪切或复制的元件上单击鼠标右键，选择剪切或复制项。

➤ 方式 1



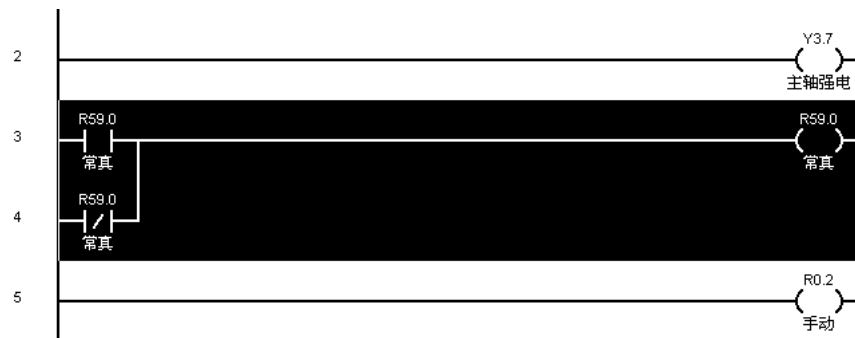
➤ 方式 2



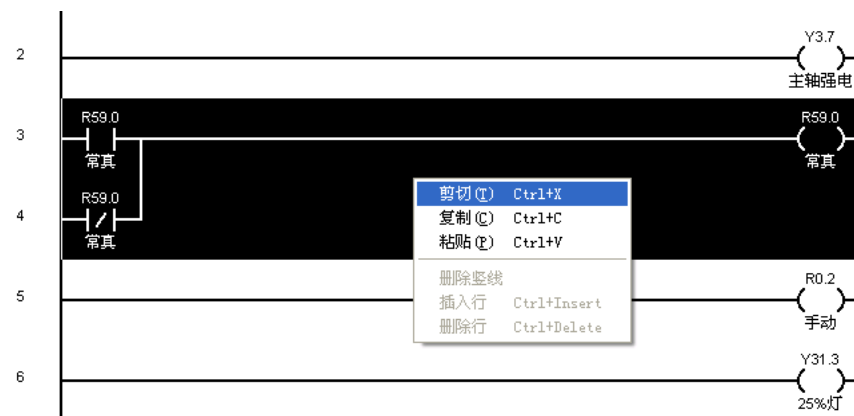
最后在此元件粘贴在其他位置。

剪切、复制和粘贴多行

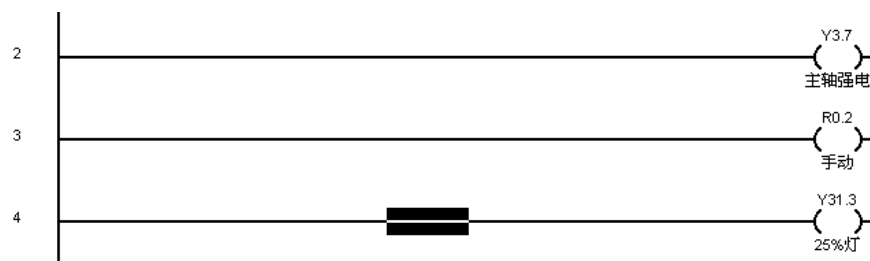
第一步：剪切、复制多行先要选中需要操作的行使用鼠标拖动选择的区域；



第二步：在菜单中选择剪切、或复制项，也可以在要剪切或复制的元件上单击鼠标右键，选择剪切或复制项。

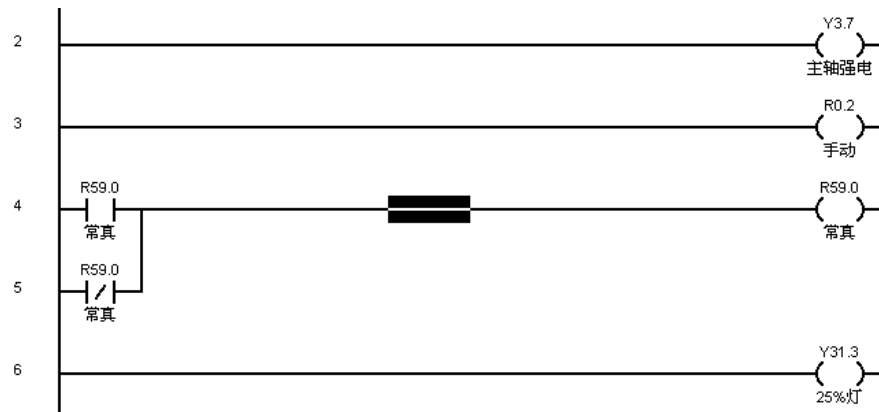


第三步：在梯形图中选择某个位置；



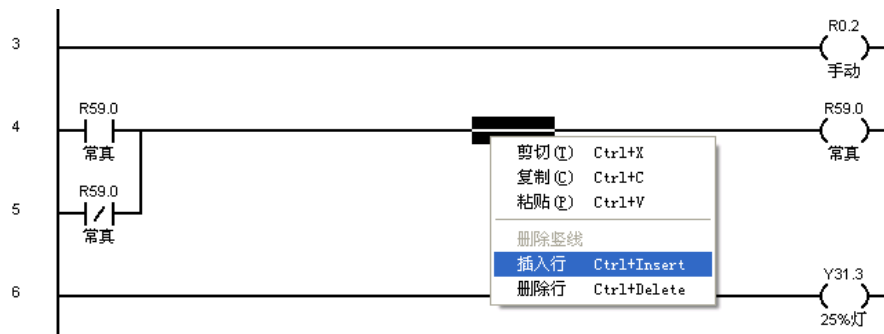
择粘贴项

第四步：选



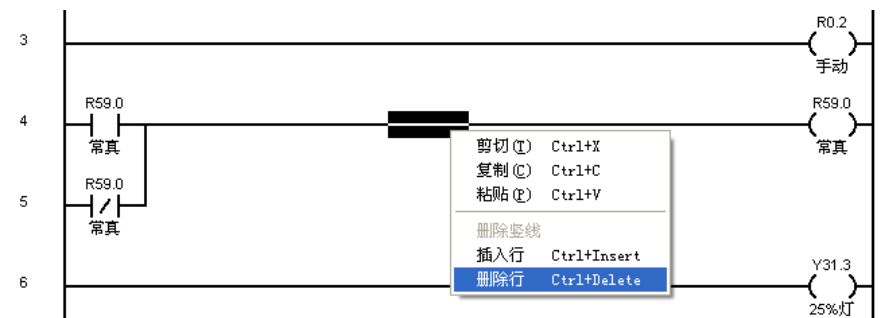
插入行

在梯形图中选中某个位置后，可以在此位置前插入一行。



删除行

在梯形图中选中某个位置后，可以删除此行。



撤消



在工具栏上选择此键，就可以撤消以前的操作。

恢复



在工具栏上选择此键，就可以恢复以前撤销的操作。

转换

转

在工具栏上选择此键，将把当前的梯形图转换成对应的语句表。如果梯形图中存在错误，将弹出消息框，显示错误信息。

输出

出

在工具栏上选择键，将把当前的梯形图转换成对应的语句表，并且输出 plc.dit 文件（梯形图执行文件）。如果梯形图中存在错误，将弹出消息框，显示错误信息。

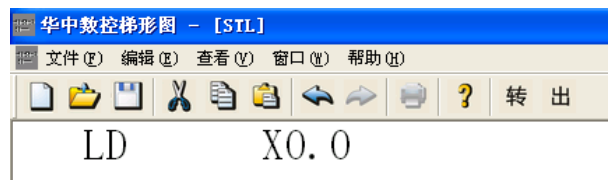
语句表操作

编辑

在语句表中，可以直接输入字符，进行语句表的编辑。



当一行语句输入完成后，光标移开时，将对此行进行检查。并对此行进行排列。

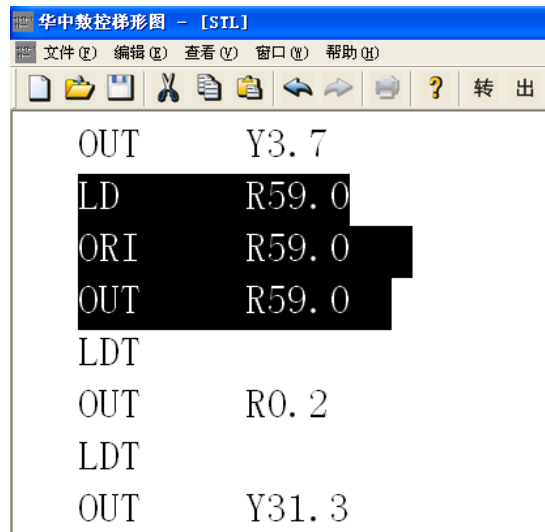


如果此行存在错误，语句表将会对此行进行错误注释。



剪切、复制和粘贴

在语句表中，使用鼠标拖动，可以选择一段语句表。



然后使用菜单中的剪切、复制和粘贴键，可以执行相应的操作。

转换

转

在工具栏上选择此键，将把当前的语句表转换成对应的梯形图。

输出

出

在工具栏上选择键，将把当前的梯形图转换成对应的语句表，并且输出 plc.dit 文件（梯形图执行文件）。

附录 A

➤ 818A 车床面板

	0	1	2	3	4	5	6	7
X480	自动	单段	手动	增量	回参 考点	卡盘 松紧	内卡 外卡	空运行
X481	程序 跳段	选择停	MST 锁住	机床 锁住	尾台 松紧	液压 启动	进给 保持 II	手动 换刀
X482		—X		x1	x10	x100	x1000	工作灯
X483	防护门	—Z	快进	+Z	主轴点 动	冷却	润滑	主轴 升档
X484	排屑 正转	排屑 反转		+X		主轴 正转	主轴 停止	主轴 反转
X485	主轴 降档		超程 解除					
X486	快移修调				循环启 动	进给保持		
X487	主轴修调							
X488	手摇急停、手摇轴选和手摇倍率							
X489	进给修调							
X490	手摇每周周期增量脉冲							
X491								

➤ 818A 铣床面板

	0	1	2	3	4	5	6	7
X480	自动	单段	手动	增量	回参 考点	换刀 允许	刀具 松紧	空运行
X481	程序 跳段	选择 停	Z 轴 锁住	机床 锁住	防护 门	机床 照明	进给 保持 II	手 动 换 刀
X482	+4	+Z	—Y	x1	x10	x100	x1000	F1
X483	F2	+X	快进	—X	主轴定向	主轴 点动	主轴 制动	冷却
X484	F3	F4	+Y	—Z	—4	主轴 正转	主轴 停止	主轴 反转
X485	润滑		超程 解除					
X486	快移修调				循环 启动	进给 保持		
X487	主轴修调							
X488	手摇急停、手摇轴选和手摇倍率							
X489	进给修调							
X490	手摇每周期增量脉冲							
X491								

➤ 818B 车床面板

	0	1	2	3	4	5	6	7
X480	自动	单段	手动	增量	回参 考点	卡盘 松紧	尾台 松紧	空运 行
X481	程序 跳段	选择 停	MST 锁住	机床 锁住	中心 架	尾台连接	进给 保持 II	手动 换刀
X482				0%	25%	主轴正转	主轴 停止	主轴 反转
X483	工作灯	+C	-Y		50%	100%	主轴 点动	主轴 升档
X484	主轴 降档	防护 门	-X	快进	+X	F1	F2	冷却
X485	润滑	液压 启动	自动 断电		+Y	-C	F3	F4
X486	排屑 正转	排屑 停止	排屑 反转	超程 解除	循环 启动	进给 保持		
X487	主轴修调							
X488	手摇急停、手摇轴选和手摇倍率							
X489	进给修调							
X490	手摇每周期增量脉冲							
X491								

➤ 818B 铣床面板

	0	1	2	3	4	5	6	7
X480	自动	单段	手动	增量	回参 考点	换刀 允许	刀具 松紧	空运行
X481	程序 跳段	选择停	Z 轴 锁住	机床 锁住			刀库 正转	刀库 反转
X482	X	Y	Z	0%	25%	主轴正转	主轴 停止	主轴 反转
X483	工作灯	A	B	C	50%	100%	主轴 定向	主轴 点动
X484	主轴 制动	防护 门	7	8	9	F1	F2	冷却
X485	润滑	吹屑	自动 断电	—	快进	+	F3	F4
X486	排屑 正转	排屑 停止	排屑 反转	超程 解除	循环 启动	进给 保持		
X487	主轴修调							
X488	手摇急停、手摇轴选和手摇倍率							
X489	进给修调							
X490	手摇每周期增量脉冲							
X491								

