

# CZA 系列可编程控制器

用户手册 [指令应用篇]

CZA 系列可编程控制器  
用户手册 [指令应用篇]

前言

---

程序编写方式概述 1

---

各种软元件的使用说明 2

---

基本顺控指令说明 3

---

应用指令说明 4

---

高数计数 5

---

脉冲输出 6

---

通讯功能 7

---

C语言功能块 8

---

常见问题及处理方法 9

---

附录

---

第二版

## 基本说明

- 感谢您购买了CZA 系列可编程逻辑控制器。
- 本手册主要介绍CZA 系列可编程逻辑控制器的指令应用等内容。
- 在使用产品之前，请仔细阅读本手册，并在充分理解手册内容的前提下进行接线。
- 软件及硬件方面的介绍，请查阅相关手册。
- 请将本手册交付给最终用户，感谢您的配合。

## 用户须知

- 只有具备一定的电气应用知识的操作人员才可以对产品进行接线等其他操作，如有使用不明的地方，请咨询本公司的技术部门。
- 手册等其他技术资料中所列举的示例仅供用户理解、参考用，不保证一定动作。
- 将该产品与其他产品组合使用的时候，请确认是否符合有关规格、原则等。
- 使用该产品时，请自行确认是否符合要求以及安全，对于本产品故障而可能引发机器故障或损失时，请自行设置后备及安全功能。

## 责任申明

- 本手册的内容虽然已经过仔细的核对，但我们不能保证完全一致。
- 我们会经常检查手册中的内容，并在后续版本中进行更正，欢迎提出宝贵意见。
- 手册中所介绍的内容，如有变动，请谅解不另行通知。

## 联系方式

如果您有任何关于本产品的使用问题，请与购买产品的代理商、办事处联系，也可以直接与我司联系。

- 电话：0577-62735555
- 传真：0577-62722963
- 地址：浙江省乐清市经济开发区纬十九路328号欣灵电气股份有限公司
- 邮编：325600

## 目 录

<b>CZA 系列可编程控制器</b> .....	<b>1</b>
<b>目 录</b> .....	<b>3</b>
本手册的内容构成.....	7
手册的适用范围.....	8
手册的获取途径.....	8
1 编程方式概述.....	9
1-1. 可编程控制器的特点.....	10
1-2. 编程语言.....	11
1-2-1. 种类.....	11
1-2-2. 互换性.....	11
1-3. 编程方式.....	12
2 软元件的作用和功能.....	13
2-1. 常数 (K、H) .....	14
2-2. 软元件一览表.....	16
2-2-1. 软元件一览.....	16
2-2-2. 停电保持区域.....	17
2-3. 输入输出继电器 (X、Y) .....	18
2-4. 辅助继电器 (M) .....	20
2-5. 状态继电器 (S) .....	21
2-6. 定时器 (T) .....	22
2-7. 计数器 (C) .....	25
2-8. 数据寄存器 (D) .....	28
2-8-1. 变址寄存器 (V、Z) .....	31
2-8-2. 变址应用举例.....	32
2-9. 指针 (P、I) 及编程原则 (中断、子程序、响应滞后、双线圈) .....	33
3 基本顺控指令说明.....	36
3-1. 基本指令一览表.....	38
3-2. [LD], [LDI], [OUT].....	41
3-3. [AND], [ANI].....	43
3-4. [OR], [ORI].....	44
3-5. [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF].....	45
3-6. [ORB].....	46
3-7. [ANB].....	47
3-8. [MCS], [MCR].....	48
3-9. 位取反[ALT].....	49
3-10. [PLS], [PLF].....	50
3-11. [SET], [RST].....	51
3-12. 对应计数器软元件的[OUT], [RST].....	52
3-13. [END].....	53
3-14. [GROUP], [GROUPE].....	54
3-15. 编程注意事项.....	55

4 应用指令说明.....	56
4-1. 应用指令一览表.....	57
4-2. 应用指令的阅读方法.....	59
4-3. 程序流程指令.....	61
4-3-1. 条件跳转[CJ].....	62
4-3-2. 子程序调用[CALL]/子程序返回[SRET].....	63
4-3-3. 中断[EI]、[DI]、[IRET].....	65
4-3-4. 流程[SET]、[STL]、[RET].....	69
4-3-5. 循环 [FOR]、[NEXT].....	71
4-3-6. 结束 [FEND]、[END].....	73
4-3-7. 监视定时器[WDT].....	74
4-4. 触点比较指令.....	75
4-4-1. 开始比较[LD□].....	76
4-4-2. 串联比较[AND□].....	78
4-4-3. 并联比较[OR□].....	80
4-5. 数据传送指令.....	82
4-5-1. 数据比较[CMR].....	83
4-5-2. 数据区间比较[ZCP].....	84
4-5-3. 传送[MOV].....	85
4-5-4. 成批传送[BMOV].....	86
4-5-5. 位移动[SMOV].....	88
4-5-6. 多点重复传送[FMOV].....	90
4-5-7. 批次复位[ZRST].....	92
4-5-8. 高低字节交换[SWAP].....	93
4-5-9. 交换[XCH].....	94
4-6. 数据运算指令.....	95
4-6-1. 加法运算[ADD].....	96
4-6-2. 减法运算[SUB].....	98
4-6-3. 乘法运算[MUL].....	100
4-6-4. 除法运算[DIV].....	102
4-6-5. 自加 1[INC]、自减 1[DEC].....	104
4-6-6. 求平均值[MEAN].....	105
4-6-7. 逻辑与[WAND]、逻辑或[WOR]、逻辑异或[WXOR].....	106
4-6-8. 逻辑取反[CML].....	108
4-6-9. 求补[NEG].....	109
4-7. 数据移位指令.....	110
4-7-1. 左回转[ROL]、右回转[ROR].....	111
4-7-2. 带进位左回转[RCL]、带进位右回转[RCR].....	112
4-7-3. 移位写入[SFWR]、移位读出[SFRD].....	113
4-7-4. 位左移[SFTL].....	115
4-7-5. 位右移[SFTR].....	116
4-7-6. 字左移[WSFL].....	117
4-7-7. 字右移[WSFR].....	118
4-8. 数据转换指令.....	119
4-8-1. 16 位整数转浮点数[FLT].....	120
4-8-2. 浮点转整数[INT].....	122

4-8-3. 译码[DECO].....	124
4-8-4. 高位编码[ENCO].....	126
4-9. 浮点运算指令.....	128
4-9-1. 浮点数比较[DECMP].....	129
4-9-2. 浮点数区间比较[DEZCP].....	131
4-9-3. 浮点数加法[DEADD].....	133
4-9-4. 浮点数减法[DESUB].....	134
4-9-5. 浮点数乘法[DEMUL].....	135
4-9-6. 浮点数除法[DEDIV].....	136
4-9-7. 浮点数开方[DESQR].....	137
4-9-8. 浮点 SIN 运算[DSIN].....	138
4-9-9. 浮点 COS 运算[DCOS].....	139
4-9-10. 浮点 TAN 运算[DTAN].....	140
4-10. 时钟指令.....	142
4-10-1. 时钟数据读取[TRD].....	143
4-10-2. 时钟数据写入[TWR].....	144
4-10-3. 时钟数据比较[TCMP].....	145
4-10-4. 时钟数据区间比较[TZCP].....	147
4-10-5. 时钟数据加法[TADD].....	149
4-10-6. 时钟数据减法[TSUB].....	151
<b>5 高速计数.....</b>	<b>153</b>
5-1. 功能概述.....	154
5-2. 高速计数模式.....	154
5-3. 高速计数值范围.....	155
5-4. 高速计数器输入端接线.....	155
5-5. 高速计数输入端口分配.....	155
5-6. 高速计数相关指令.....	156
5-6-1. 高速计数指令[OUT].....	156
5-6-2. 高速计数比较复位[HSCR].....	157
5-6-3. 高速计数比较置位[DHSCS].....	158
5-6-4. 高速计数复位模式[RST].....	159
5-7. 高速计数器与普通计数器的区别.....	159
5-8. 高速计数举例.....	160
5-9. 输入输出刷新[REF].....	161
<b>6 脉冲输出.....</b>	<b>162</b>
6-1. 功能概述及端口分布.....	164
6-2. 脉冲输出的种类与指令应用.....	165
6-2-1. 定频无加减速单向定量脉冲输出[PLSY].....	165
6-2-2. 带加减速脉冲输出 [PLSR].....	168
6-2-3. 脉宽调制[PWM].....	171
6-2-4. 原点回归[ZRN].....	172
6-2-5. 相对位置单段脉冲控制[DRVI].....	177
6-2-6. 绝对位置单段脉冲控制[DRVA].....	181
6-2-7. 带S型加减速的定向脉冲输出[PLSRS].....	185
6-2-8. 任意曲线脉冲输出[PLSRC].....	188

6-2-9. 带方向可变频率脉冲输出[PLSF].....	192
6-3. 输出端子接线.....	194
6-4. 注意事项.....	194
7 通讯功能.....	197
7-1. 概述.....	199
7-1-1. 通讯口.....	199
7-1-2. 通讯参数.....	201
7-2. MODBUS 通讯功能.....	202
7-2-1. 通讯功能.....	202
7-2-2. 通讯地址.....	203
7-2-3. Modbus 通讯数据格式.....	204
7-3. 通讯指令.....	209
7-3-1. 线圈读[COLR].....	210
7-3-2. 输入线圈读[INPR].....	211
7-3-3. 单个线圈写[COLW].....	212
7-3-4. 多个线圈写[MCLW].....	213
7-3-5. 寄存器读[REGR].....	214
7-3-6. 输入寄存器读[INRR].....	215
7-3-7. 单个寄存器读[REGW].....	216
7-3-8. 多个寄存器读[MRGW].....	217
7-3-9. 通讯样例及说明.....	218
7-4. 自由格式通讯.....	219
7-4-1. 通讯模式.....	219
7-4-2. RS串行通讯指令.....	220
8 C语言功能块.....	223
8-1. 概述.....	224
8-2. 指令形式.....	225
8-3. 操作步骤.....	226
8-4. 函数的导入、导出.....	230
8-5. 功能块的编辑.....	231
8-6. 程序举例.....	233
<b>附录1 特殊软元件一览表.....</b>	<b>235</b>
附录 1-1. 特殊辅助继电器一览.....	236
附录 1-2. 特殊数据寄存器一览.....	241
附录 1-3. 扩展模块地址一览.....	244
附录 1-4. 掉电保持区域设置一览.....	245
附录 1-5. 特殊寄存器总览.....	246
<b>附录2 应用指令一览.....</b>	<b>257</b>

## 本手册的内容构成

本手册主要介绍 CZA 系列。可编程控制器的指令应用，包含 CZA 系列可编程控制器的基本指令、应用指令等，同时涉及了编程中的要点、原则等，各章节内容概览如下：

### 1. 可编程逻辑控制器的概述

本章主要介绍 CZA 系列可编程控制器的性能特点、型号构成以及全系列产品构成、产品各部分说明等。

### 2. 各种软元件的使用说明

本章将就可编程逻辑控制器的数值以及内置的输入输出继电器、辅助继电器、状态继电器、计数器、数据寄存器等各种软元件的作用和功能加以说明，该章节是可编程逻辑控制器的编程基础。

### 3. 基本顺控指令说明

本章主要介绍 CZA 系列可编程控制器共用的基本顺控指令的种类及其功能。

### 4. 应用指令解说

本章主要介绍 CZA 系列可编程控制器的应用指令的种类及其功能。

### 5. 高速计数

本章主要介绍 CZA 系列可编程控制器的高速计数功能，内容包括了高速计数的模式、接线方法、高速计数值的读写、复位等。

### 6. 脉冲输出

本章主要介绍 CZA 系列可编程控制器的脉冲输出功能，内容包括脉冲输出指令用法及编程应用、输出端子的接线、注意事项以及相关线圈和寄存器等等。

### 7. 通讯功能

本章论述了 CZA 系列可编程控制器的通讯功能，内容主要包括通讯的基本概念、Modbus 通讯、232 通讯。

### 8. C语言功能块

本章主要重点介绍 C 语言编辑功能块的特点、编辑。指令调用、函数的存储、应用要点等，并带有常用功能块列表。

### 9. 常见问题及处理方法

本章主要就一些用户在使用 PLC 的过程中遇到的常见问题，提供快速而具体的分析和解决方法。

### 附录 1. 特殊软元件一览表

本章主要介绍 CZA 系列可编程控制器中功能性软元件、寄存器，以及扩展模块地址分配等。

### 附录 2. 应用指令一览

本章将列出应用指令及特殊功能指令的基本信息，以及与 CZA 系列 PLC 的适用关系。



## 手册的适用范围

本手册为 CZA 系列可编程控制器产品的指令手册，该手册适用于 CZA 系列所有产品：

1. CZA系列可编程控制器基本单元
  - CZA 系列
    - 1) 16点(两路高速脉冲，一路高数计数)
    - 2) 24 点（两路高速脉冲，两路高数计数）
    - 3) 30点（两路高速脉冲，两路高数计数）

## 手册的获取途径

本公司产品的用户手册用户一般可通过以下几种途径来获取：

1. 印刷版手册  
请向购买产品的供应商、代理商、办事处咨询索取。
2. 电子版手册
  - (1) 登陆我司官方网站下载中心。
  - (2) 向购买产品的供应商、代理商、办事处索取产品的电子版用户手册。

# 1 编程方式概述

CZA 系列 PLC 作为控制器，而程序则为为控制器的灵魂，谈到程序首先就需要编写程序。本章将以编程方式开篇，介绍 CZA 系列 PLC 的主要特点、所支持的两种编程语言，以及在软件中的编程方式。

1-1. 可编程控制器的特点

1-2. 编程语言

1-3. 编程方式

## 1-1. 可编程控制器的特点

<b>编程语言</b>	CZA 系列可编程控制器支持两种编程语言，指令表和梯形图，两种语言可以随意转换，实际编程时大部分程序员都以梯形图为主。
<b>程序安全性</b>	为了维护知识产权以及PLC运行时的可靠性，一般可根据需要对程序进行加密。加密后的程序在上传的时候，将以口令的形式进行验证，这样可以很好的维护用户的版权；同时还能限制下载，防止恶意修改 PLC 里面的程序。
<b>程序的注释</b>	当用户程序过于复杂冗长时，伴随着的就是大量的软元件被使用，常常需要对程序使用的软元件加以注释，一是方便编写者日后的修改，二是方便使用更好的学习和理解。
<b>变址功能</b>	在数据寄存器后加上变址后缀（如D0[Z0]，可实现间接寻址。如 Z0=9 时，D0[Z0]表示D9。
<b>充实的基本功能</b>	<ul style="list-style-type: none"><li>● CZA 系列可编程控制器囊括了三菱系列的大部分基本指令，可以满足基本的顺序控制、数据的传送和比较、四则运算、逻辑控制、数据的循环和移位等功能。</li><li>● CZA 系列可编程控制器还具有中断、模拟量、高速计数器专用比较指令、高速脉冲输出指令、频率的测量、精确定时、PID 控制、以及定位控制等指令。</li></ul>
<b>C 语言编辑功能块</b>	CZA 系列可编程控制器可实现利用 C 语言来编写功能块的功能，编辑好的功能块可以在程序中随意调用，保密性好，适用性强，也减小了编程的工作量，同时也可以根据需要自行定义编写功能块。
<b>上电运行停止功能</b>	在CZA 系列可编程控制器中具有一项比较重要的功能，即M8002上电运行触发停止，当PLC 在运行过程中出现了比较严重的错误，可能导致机器故障或损坏时，利用上电运行停止功能，可以马上停止所有输出。此外，通信参数错误设置也可以用此方法来连接上PLC，然后修改通信参数。
<b>通讯功能</b>	CZA 系列可编程控制器可支持多种通讯方式，如基本的 Modbus 通讯、自由格式通讯（232）；GPRS 网络（云端功能）；通过特定的扩展 BD 板，适应更加广泛的应用场合。

## 1-2. 编程语言

### 1-2-1. 种类

CZA 系列 PLC 支持以下两种编程语言。

#### 命令语

指令表编程是以“LD”、“AND”、“OUT”等顺控指令输入的方式。这种方式是编写顺控程序的基本输入形式，但不够直观，可读性较差。

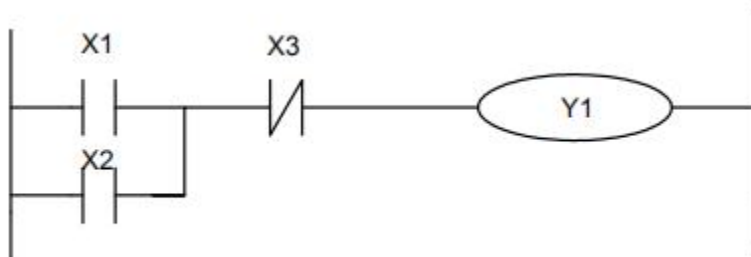
例： 步 指令 软元件号

0	LD	X1
1	OR	X2
2	ANI	X3
3	OUT	Y1

#### 梯形图

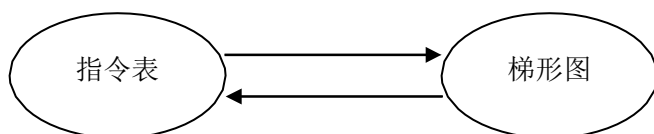
梯形图程序是采用顺控信号及软元件号，在图形画面上作出顺控电路图的方法。这种方法是用触点符号与线圈符号表示顺控回路，因而容易理解程序的内容。同时还可以直观地看出回路中软元件及线圈的运行状态。

例：



### 1-2-2. 互换性

以上这 2 种输入方法编制的程序表示及编辑都可相互转换。



## 1-3. 编程方式

## 直接输入

1-2 节介绍的两种编程语言，均可以在对应的窗口中直接输入，尤其梯形图窗口中还具有指令提示功能，使程序的编写效率有所提高。



## 面板配置

在 CZA 系列 PLC 支持的指令中，某些指令的用法比较复杂，或者用法较多，如脉冲输出指令、本体 PID 指令等，编程软件中还特别提供了这些特殊指令的配置方式。在相应的配置面板中，只要按照自己的要求，输入参数、地址等即可。



# 2 软元件的作用和功能

第一章简要介绍了 CZA 系列 PLC 的编程语言，而对于程序而言，其核心对象是控制器内部的继电器和寄存器等。本章将一一介绍这些继电器、寄存器的作用和功能，以及使用技巧。

2-1. 常数 (K、H)

2-2. 软元件编号一览表

2-3. 输入输出继电器 (X、Y)

2-4. 辅助继电器 (M)

2-5. 状态继电器 (S)

2-6. 定时器 (T)

2-7. 普通计数器

2-8. 数据寄存器 (D)

2-9. 指针 (P、I)

## 2-1. 常数 (K、H)

## 数据处理

CZA 系列可编程控制器根据不同的用途和目的，使用 5 种类型的数制。其作用和功能如下：

- 十进制数 (DEC: DECIMAL NUMBER)
  - 定时器和计数器的设定值 (K 常数)
  - 辅助继电器 (M)，定时器 (T)，计数器 (C)，状态 (S) 等的编号 (软元件编号)
  - 指定应用指令操作数中的数值与指令动作 (K 常数)
- 十六进制数 (HEX: HEXADECIMAL NUMBER)
  - 同十进制数一样，用于指定应用指令操作数中的数值与指令动作 (H 常数)
- 二进制数 (BIN: BINARY NUMBER)
  - 如前所述，以十进制数或是十六进制数对定时器、计数器或数据寄存器进行数值指定，但在可编程控制器内部，这些数字都用二进制数处理。而且，在外围设备上进行监控时，这些软元件将自动变换为十进制数 (也可切换为十六进制)。
- 八进制数 (OCT: OCTAL NUMBER)
  - CZA 系列可编程控制器的输入继电器、输出继电器的软元件编号以八进制数值进行分配，因此，可进行[0-7, 10-17, . . . 70-77, 100-107]的进位。
- BCD 码 (BCD: BINARY CODE DECIMAL)
  - BCD 是用 4 位二进制数来表示 1 位十进制数中的 0~9 这 10 个的方法。每个位的处理很容易，因此，BCD 码可用于以 BCD 形式输出的数字式开关或七段码的显示器控制等方面。
- 其他数值 (浮点数)
  - CZA 可编程控制器具有可进行高精度浮点运算的功能。  
用二进制浮点数进行浮点运算，同时用十进制浮点值实施监视。

## 常数 (K、H)

PLC 的程序进行数值处理时，必须使用常数 K、H。一般使用 K 指代十进制数，H 指代十六进制数，但 PLC 的输入、输出继电器使用八进制编号。

- 常数 K

---

K 是表示十进制整数的符号，如 K10，表示十进制数 10。其主要用于指定定时器、计数器的设定值，以及应用指令中的操作数等。

- 常数 H

H 是表示十六进制数的符号，如 HA，表示十六进制数 10。主要用于指定应用指令的操作数的数值。（**注意：作为指令操作数时，地址首位如果是字母，需要在前面加 0，如：HA 要写作 H0A。**）



## 2-2. 软元件一览表

## 2-2-1. 软元件一览

## CZA 系列

识别 记号	名称	范围			点数		
		16 点	24点	30点	16	24	30
I/O 点 数	输入点数	X0~X7	X0~X13	X0~X16	8	12	15
	输出点数	Y0~Y7	Y0~Y13	Y0~Y16	8	12	15
M	内部继电器	一般用M0~M499			500		
		保持用/M500-M1535			1036		
S	流程	一般用S0-S1023			1024		
T	定时器	T0~T199: 100ms 不累计			200		
		T200~T245: 10ms 不累计			46		
		T246~T249: 1ms 不累计			4		
		T250~T255: 100ms 不累计			6		
C	计数器	一般用/C0~C99:16位增量计数器			100		
		保持用/C100~C199:16位增量计数器			100		
		一般用/C200~C219: 32位加减计数器			20		
		保持用/C220~C234: 32位加减计数器			15		
		C235~C255:32位高速加减计数器			21		
D、V Z	数据寄存器	一般用/D0~D199			200		
		保持用/D200~D1999			1800		
		一般用/D2000~D7999			6000		
		特殊用/D8000~D8479			480		
		变址用(保持)/V0~V7、Z0~Z7			16		
P、I	嵌套指针	跳跃,子程序用/P0~P127			128		
		输入中断用/I00*~I01*,I04*~I05*			4		

## 2-2-2. 停电保持区域

CZA 系列 PLC 的断电保持区域设置如下图。

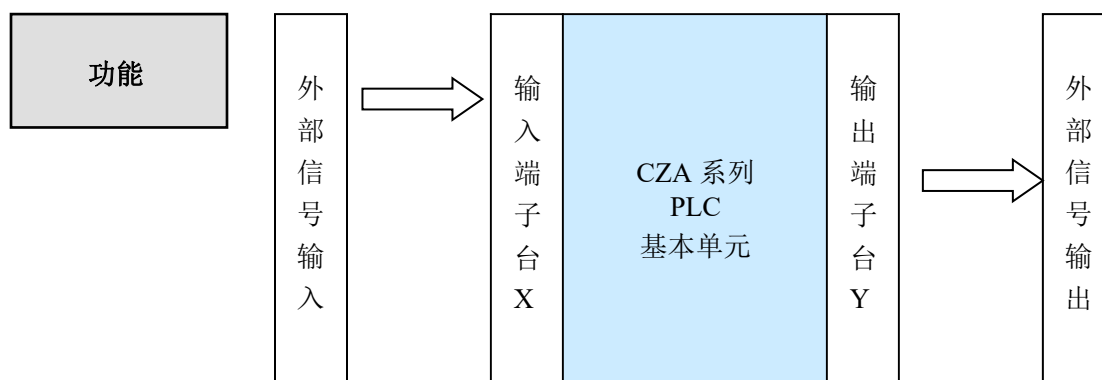
	软元件	功能	掉电记忆范围
CZA 系列	D	D 断电保存区域	D200~D1999
	M	M 断电保存区域	M500~M1535
	T	T 断电保存区域起	~
	C	C 断电保存区域	C100~C199/C220~C255
	S	S 断电保存区域	~

### 2-3. 输入输出继电器 (X、Y)

#### 编号一览

CZA 系列 PLC 的输入输出继电器全部以八进制来进行编址，各系列的编号请参见对应表格：

系列	名称	范围			点数		
		16点	24点	30点	16	24	30
CZA	X	X0~X7	X0~X13	X0~X16	8	12	15
	Y	Y0~Y7	Y0~Y13	Y0~Y16	8	12	15

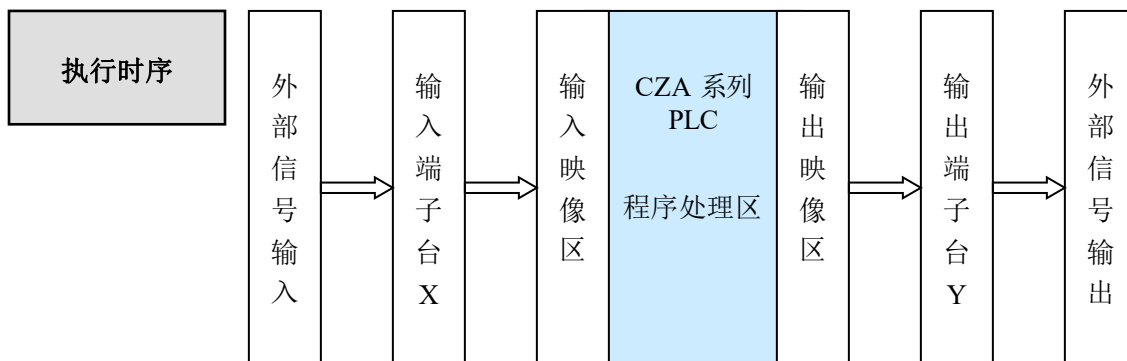


#### 输入继电器X

- PLC 的输入端子用于接收外部信号的输入，而输入继电器则是 PLC 内部与输入端子相连的一种光绝缘的电子继电器。
- 输入继电器具有无数的常开触点与常闭触点，它们可被随意使用。
- 没有与外设实连的输入继电器可作为快速内部继电器使用。

#### 输出继电器Y

- PLC 的输出端子用于向外部负载发送信号，在 PLC 内部，输出继电器的外部输出触点（包括继电器触点、晶体管触点）与输出端子相连。
- 输出继电器具有无数的常开触点与常闭触点，它们可被随意使用。
- 没有与外设实连的输出继电器可作为快速内部继电器使用。



- 输入处理
  - 外部信号从输入端子接入，PLC 在执行程序前，首先将输入端子的 ON/OFF 状态读取到输入映像区。
  - 程序执行的过程，也是不断进行扫描的过程，在本次扫描未结束前，即使输入端子状态发生变化，映像区中的内容也保持不变，直到下一个扫描周期来临，变化才被写入。
- 输出处理
  - 当所有指令执行完毕，输出 Y 的映像区中的 ON/OFF 状态将被传送到输出锁存存储区，即是 PLC 的实际输出状态。
  - PLC 内的外部输出用触点，按照输出软元件的响应滞后时间动

## 2-4. 辅助继电器 (M)

## 编号一览

CZA 系列 PLC 的辅助继电器 M 全部以十进制来进行编址，各系列的编号请参见对应表格：

系列	名称	范围		
		一般用	掉电保持用	特殊用
CZA	M	M0~M499 500点	M500~M1535 1036点	M8000~M8255 256点

在 PLC 内部，常常需要用到辅助继电器 M，该类继电器的线圈与输出继电器一样，由 PLC 内的各种软元件的触点驱动。

辅助继电器 M 有无数的常开、常闭触点，在 PLC 内部可随意使用，但该类触点不能直接驱动外部负载。

- 一般用
  - 此类辅助继电器只能作为普通的辅助继电器使用，即当 PLC 运行过程中停电，继电器将断开。
  - 一般用继电器不可用作停电保持，但可修改该段区域范围。
- 停电保持用
  - 停电保持用的辅助继电器，即使 PLC 断电后，也仍然保持断电前的 ON/OFF 状态。
  - 停电保持区域可以由用户自己修改。
  - 停电保持用继电器，通常用于需要记忆停电前的状态，上电后能够重现该状态的场合。
- 特殊用
  - 特殊用继电器指已经被系统赋予了特殊意义或功能的一部分继电器，通常从 M8000 开始。
  - 特殊继电器的用途有二，一是用于自动驱动线圈；二是用于特定的运行。如 M8002 为初始脉冲，仅在运行开始的瞬间接通；M8013 为 1 秒定时器。
  - 特殊用辅助继电器，不可作为普通继电器 M 使用。

## 2-5. 状态继电器 (S)

### 编号一览

CZA 系列 PLC 的状态继电器 S 全部以十进制来进行编址，各系列的编号请参见对应表格：

系列	名称	范围
		一般用
CZA	S	S0~S1023

### 功能

状态继电器 S 是对梯形图编程非常重要软元件，通常与指令 STL 配合使用，以流程的方式，可以使程序变得结构清晰易懂，并且易于修改。

- 一般用  
一般用的状态继电器 S 在 PLC 运行断电后，都将变为 OFF 状态。
- 状态继电器 S 也有着无数的常开、常闭触点，因此，可在程序中随意使用。

## 2-6. 定时器 (T)

### 编号一览

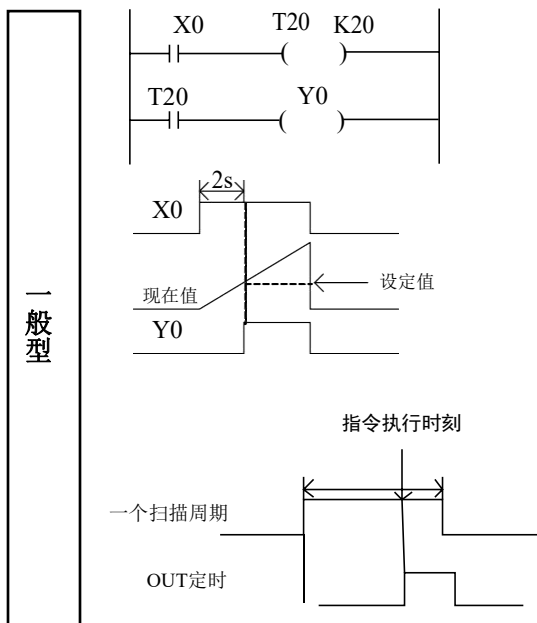
CZA 系列 PLC 的定时器 T 全部以十进制来进行编址，各系列的编号请参见对应表格：

系列	名称	范围	点数
CZA	T	T0~T199: 100ms 不累计	200
		T200-T245: 10ms 不累计	46
		T246-T249: 1ms 不累计	4
		T250-T255: 100ms 不累计	6

### 功能

定时器累计可编程控制器内的 1ms, 10ms, 10ms 等的时钟脉冲，当达到所定的设定值时输出触点动作。

普通定时器不设专用指令，使用 OUT 指令进行定时；采用程序存储器内的常数 (K) 作为设定值，也可用数据寄存器 (D) 的内容进行间接指定。

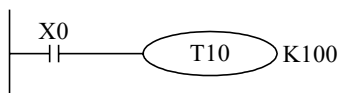


- 如果定时器线圈 T20 的驱动输入 X0 为 ON, T20用当前值计数器累计 100ms 的时钟脉冲。如果该值等于设定值 K20时，定时器的输出触点动作。也就是说输出触点在线圈驱动 2 秒后动作。驱动输入 X0 断开或停电，定时器复位，输出触点复位。

- OUT 的定时起始时间为 0。

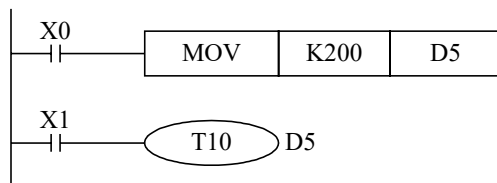
### 设定值的 指定方法

#### 《常数指定 (K)》



T10 是以 100ms 为单位的定时器。将 100 指定为常数，则  $0.1\text{s} \times 100 = 10\text{s}$  的定时器工作。

#### 《间接指定 (D)》

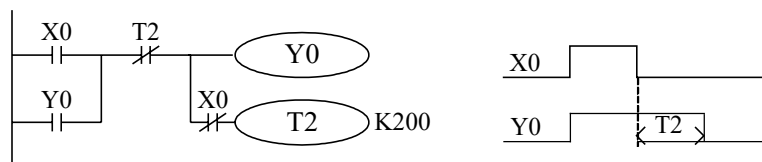


将间接指定数据寄存器的内容预先写入程序或通过数值开关输入。

在指定为停电保持用寄存器时，请注意电池电压不足会造成设定值不稳定的情况。

### 动作示例

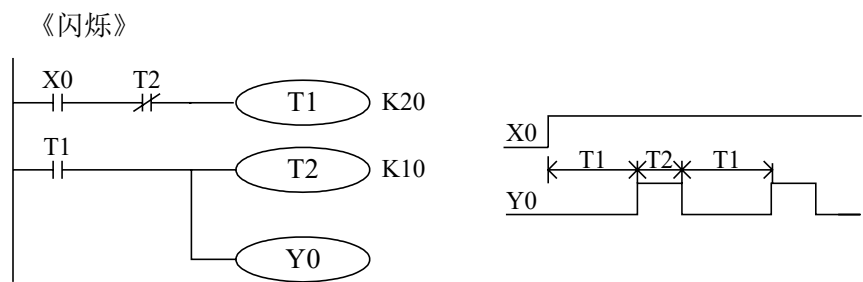
#### 《输出延时关断定时器》



X0 为 ON 时，输出 Y0；

当 X0 由 ON→OFF 时，将延时 T2（20 秒）时间，输出 Y0 才断开。





当 X0 闭合后，Y0 开始闪烁输出。

T1 控制 Y0 的断开时间，T2 控制 Y0 的闭合时间。

## 2-7. 计数器 (C)

## 编号一览

CZA系列 PLC 的计数器 C 全部以十进制来进行编址，各系列的编号请参见对应表格：

系列	名称	范围	点数
CZA	C	一般用/C0~C99: 16 位加计数器	100
		保持用/C100~C199:16位加计数器	100
		一般用C200~C219: 32 位加减计数器	20
		保持用C220~C234: 32位加减计数器	15

各种计数器的编号原则如下：

类型	说明
16 位加计数器	C0~C199
32 位加减计数器	C200~C234
高速计数器	C235~C255

1: 关于高速计数器的用法，请参阅第 5 章。

计数器  
特点

16 位计数器与 32 位计数器的特点如下表所示：

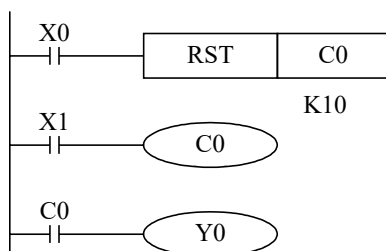
项目	16 位计数器	32 位计数器
计数方向	顺数	顺/倒数
设定值	1~32,767	-2,147,483,648~+2,147,483,647
指定的设定值	常数 K 或数据寄存器	同左，但是数据寄存器要一对
当前值的变化	顺数后变化	顺/倒数后变化（计到最大或最小值时，将保持）
输出接点	顺数后保持动作	顺数保持动作，倒数复位
复位动作	执行RST 命令时，计数器的当前值为零，当前状态复位	
当前值寄存器	16 位	32 位

## 功能

一般用计数器和停电保持用计数器的分配，可通过外围设备改变参数设定进行变更。

## 十六位计数器一般用停电保持用

16 位二进制增计数器，其有效设定值为 K1~K32,767(十进制常数)。设定值 K0 和 K1 具有相同的含义，即在第一次计数开始时输出触点就动作。

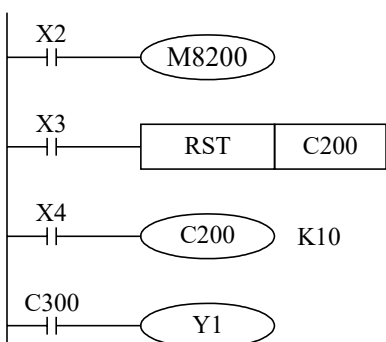


如果切断可编程控制器的电源，则一般用计数器的计数值被清除且状态值也会复位，而停电保持用的计数器则可储存停电前的计数值且状态值也会保持，因此计数器可按上次数值累计计数。

- 计数输入 X1 每驱动 C0 线圈一次，计数器的当前值就加 1，在执行第十次的线圈指令时，输出触点动作。以后计数器输入 X1 再动作，计数器的当前值将继续加 1。
- 如果复位输入 X0 为 ON，则执行 RST 指令，计数器的当前值为 0，输出状态复位。
- 计数器的设定值，除上述常数 K 设定外，还可由数据寄存器编号指定。例如，指定 D10，如果 D10 的内容为 123，则与设定 K123 时一样的，但 D10 的值可以任意更改，而 K 的值在更改后需要重新下载程序。
- 在以 MOV 等指令将设定值以上的数据写入当前值寄存器时，则在下次输入时，输出线圈接通，当前值寄存器变为设定值。

## 三十二位计数器一般用停电保持用

32 位二进制增/减计数器设定值有效范围为 K+2,147,483,648 ~K-2,147,483,647(十进制常数)。利用特殊的辅助继电器 M8200~M8234 指定所有 32 位增计数/减计数器 (C200~C234) 的方向。



- 如果 X2 驱动 M8200，则为减计数；不驱动时则为增计数。
- 根据常数 K 或数据寄存器 D 的内容，设定值为正。将连号的数据寄存器的内容视为一对，作为 32 位的数据处理。因此，在指定 D0 时，D1 和 D0 两项作为 32 位设定值处理。利用计数输入 X004 驱动 C200 线圈时，进行增/减计数。

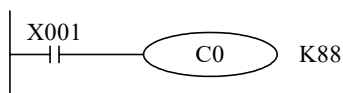
- 如果复位输入 X3 为 ON，则执行 RST 指令，计数器的当前值变为 0，输出状态也复位。
- 使用供停电保持用的计数器时，计数器的当前值、输出触点动作与复位状态停电保持。
- 32 位计数器也可作为 32 位数据寄存器使用。

### 设定值的 指定方法

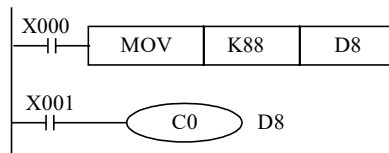
计数值的指定，分为 16 位数和 32 位数两种情况讨论。

#### ◆ 16 位计数器

《常数指定 (K)》



《间接指定 (D)》

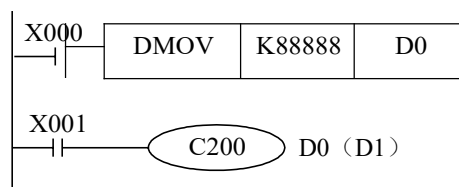


#### ◆ 32 位计数器

《常数指定 (K)》



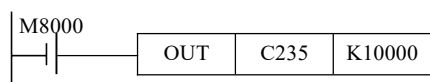
《间接指定 (D)》



### 计数值

计数器 C0~C199 的计数模式是 16 位线性递增模式 (0~K32,767)，当计数器的数值达到最大值 K32,767 会停止计时，计数器的状态保持不变。

计数器 C200~C234 的计数模式是 32 位线性增/减模式 (-2,147,483,648 ~+2,147,483,647)，当计数器的计数值递增达到最大值 K2,147,483,647 会变成 K-2,147,483,648，当计数器的计数值递减达到最小值 K-2,147,483,648 会变成 K2,147,483,647，649，计数器的 ON/OFF 状态也随计数值的变化而变化。



高速计数，C235 对应输入端子 X0，高速计数不受输入滤波器的响应滞后时间和循环扫描周期时间影响。因此，可以处理更高频率的输入脉冲。具体参考第 5 章高速计数部分。

## 2-8. 数据寄存器 (D)

### 编号一览

CZA 系列 PLC 的数据寄存器 D 全部以十进制来进行编址，各系列的编号请参见对应表格：

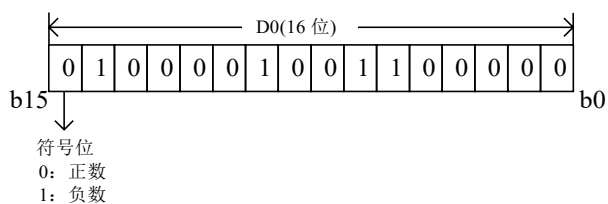
系列	名称	范围		
		一般用	停电保持用	特殊用
CZA	D	D0~D199	D200~D1999	D8000~D8369 370

### 结构

数据寄存器是用于存储数据的软元件，包括 16 位（最高位为符号位）、32 位（由两个数据寄存器组合，最高位为符号位）两种类型。

### 十六位

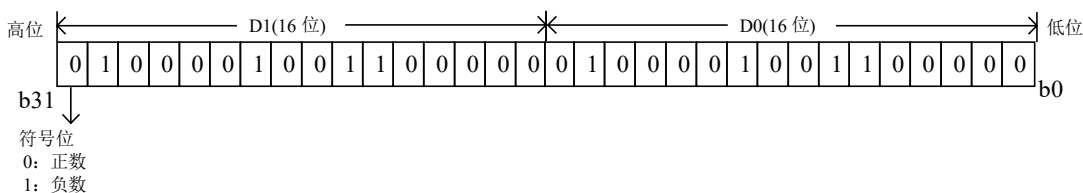
一个 16 位的数据寄存器，其处理的数值范围为  $K-32,768 \sim K+32,767$



数据寄存器的数值的读写一般采用应用指令。另外，也可通过其他设备，如人机界面向 PLC 写入或读取数值。

由两个相邻的数据寄存器组成的 32 位数据（高字在后，低字在前，如 D1D0 组成的双字，D0 为下位，D1 为上位）。

处理的数值范围为  $K-2,147,483,648 \sim K2,147,483,647$ 。



在指定 32 位寄存器时，如果指定了低位，如 D0，则默认其高位为后继的 D1。低位可用奇数或偶数的任意一种软元件来指定，但为方便起见，建议低位采用偶数软元件编号。

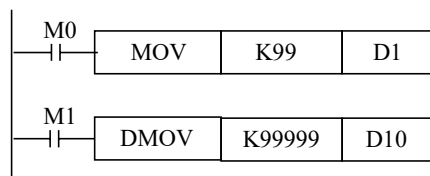
### 功能

- 一般用
  - 当向数据寄存器中成功写入数据后，只要不再重新写入，那么该寄存器中的数据将保持不变。
  - 当 PLC 由 RUN 转为 STOP 或由 STOP 转为 RUN 时，所有数据将被清零。
- 停电保持用
  - 停电保持区的数据寄存器在 PLC 由 RUN 转为 STOP 或停电后，仍然保持其中的数据不变。
  - 停电保持区域的范围，可以由用户自行设定。
- 特殊用
  - 特殊用寄存器用于写入特定目的的数据，或已由系统写入特定内容的数据。
  - 部分特殊寄存器中的数据，在 PLC 上电时，被初始化。
  - 特殊寄存器的编号和用途，请参阅附录部分。

动作示例

数据寄存器D 可以处理各种数据，通过数据寄存器可实现多种控制。

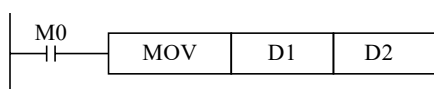
● 数据存储



M0 接通时，向D1 写入 16 位、十进制数 99。

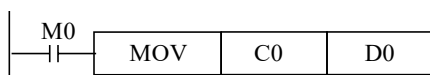
M1 接通时，向D11D10 写入 32 位、十进制数 99999。由于数值 99999 为 32 位数（超过 32767），因此存储数据双置，虽指定为D10，但 D11 也被自动占用。

● 数据传送



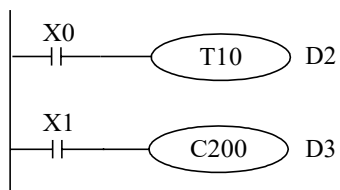
M0 接通时，将D1 中的数据传送给D2。

● 读取定时器或计数器



M0 接通时，将计数器C0 中的当前计数值读取到D0 中。

● 作为定时器或计数器的设定值



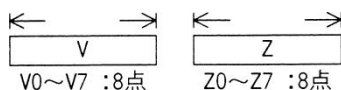
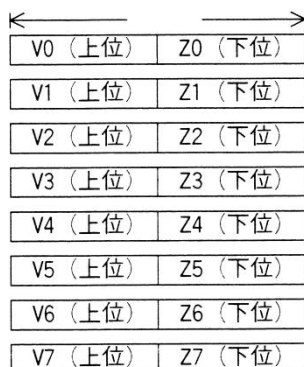
X0 接通时，T10 开始定时，定时时间由 D2 中的数值决定。

X1 每次接通时，C200 开始计数，计数值由 D3 决定。

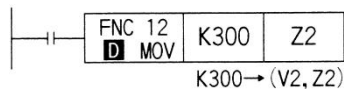
## 2-8-1. 变址寄存器 (V、Z)

## 功能与结构

- 变址寄存器同普通的数据寄存器一样，是进行数值数据的读入、写出的16位数据寄存器。V0~V7，Z0~Z7共有16个。
- 这种变址寄存器除了和普通寄存器有相同的使用方法外，在应用指令的操作数中，还可同其他的软元件编号或数值组合使用，可在程序中改变软元件编号或数值内容，是一种特殊的寄存器。
- 此外，请注意LD，AND，OUT等可编程控制器的基本顺控指令或步进梯形图指令的软元件编号不能同变址寄存器组合使用。
- 在处理32位应用指令中软元件或处理超过16位范围数值时，必须使用Z0~Z7，即V(高位)、Z(低位)，且Z0~Z7、V0~V7一一对应。即使在32位应用指令中使用的变址值没有超过16位数值使用范围，对于Z的数值写入也会出现如下图所示的情况。使用DMOV等32位指令，请同时改写V(高位)，Z(低位)32位变址寄存器写入举例。

16  
位32  
位

## 32位寄存器写入例



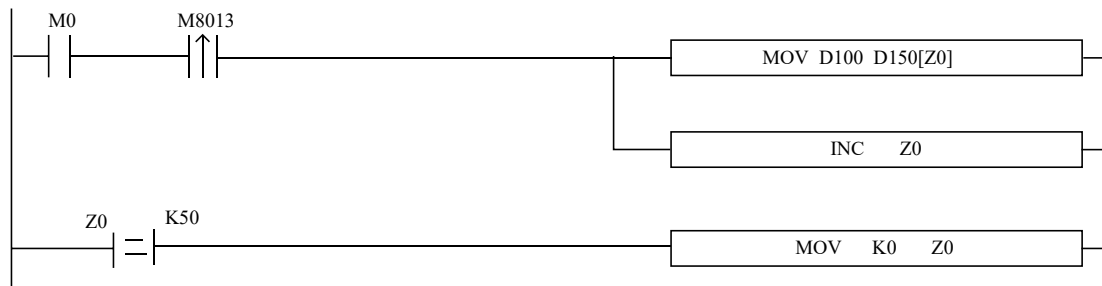
## 软元件的变址

- 只适用于软元件D，其变址内容如下所示。  
例如:V0=K5,执行D20[V0]时，被执行软元件编号为D(20+5)，即D25。



**2-8-2. 变址应用举例**

例：当 M0 为 ON 时，每 1S 钟对 D100 的数据进行一次保存，存放在 D150 开始的 50 个寄存器里。寄存器 Z0 作变址使用。



## 2-9. 指针 (P、I) 及编程原则 (中断、子程序、响应滞后、双线圈)

● 标记 P、I

标记 P 用于分支或子程序。

标记 I 用于中断 (外部中断, 定时中断, 高速计数中断, 精确定时中断等)。

分支或子程序用的标记 (P) 用于条件跳转或子程序的跳转目标。

中断用的标记 (I) 用于指定输入中断、定时中断等中断程序标志。

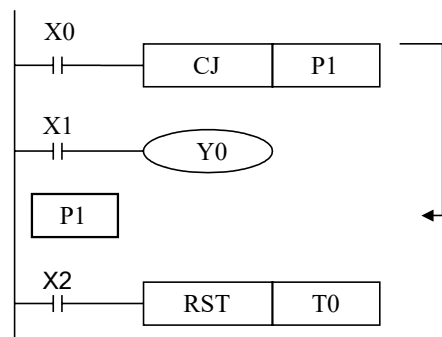
CZA 系列 PLC 的标记 P、I 全部以十进制来进行编址, 各系列的编号请参见对应表格

系列	名称	范围
CZA	P	P0~P127

系列	名称	范围			
		外部中断用			定时中断用
		输入端子	上升中断	下降中断	
CZA	I	X0	I0001	I0000	共有3路定时中断, 表示方法为: I06**, I07**, I08**。其中'**'表示定 时中断的时间10~99, 单位毫秒。
		X1	I0101	I0100	
		X4	I0401	I0400	
		X5	I0501	I0500	

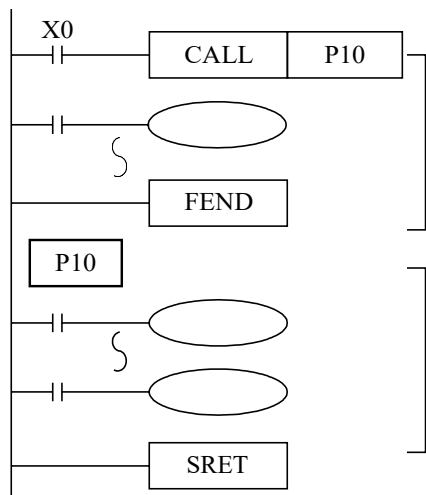
标记 P 通常用于流程中, 一般与 CJ (条件跳转)、CALL (子程序调用) 等指令配合使用。

● 条件跳转 CJ



当线圈 X0 接通时, 跳转到 P1 标记的后一步, 不执行中间部分程序。  
当线圈 X0 未接通时, 不执行跳转动作, 仍然按照原步骤执行。

● 子程序调用 CALL



当线圈 X0 接通时, 由主程序跳到子程序;  
当线圈 X0 未接通时, 仍然执行主程序。

当子程序执行完毕后, 返回主程序, 继续执行下面的程序。

子程序调用，写程序时必须以 Pn 作为一段子程序的开始，以 SRET 作为一段子程序的结束。用 CALL Pn 调用子程序。其中 n 可以为 0~9999 中的任意整数。

标记  
I

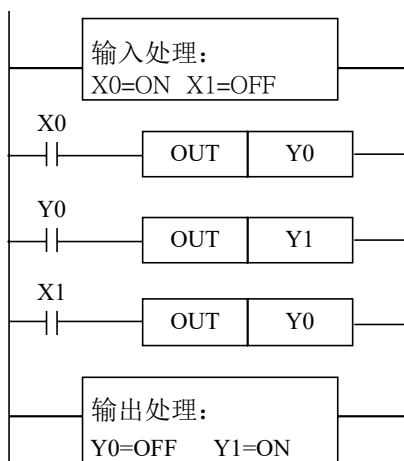
标记 I 一般用于中断功能，包括外部中断、定时中断等场合，通常与 IRET（中断返回）、EI（允许中断）、DI（禁止中断）等指令配合使用。

- 外部中断
  - 接收来自特定的输入端子的输入信号，不受扫描周期的影响。触发输入信号，执行中断子程序。
  - 通过外部中断可处理比扫描周期更短的信号，因而可在顺控过程中作为必要的优先处理或短时脉冲处理控制中使用。
- 定时中断
  - 在各指定的中断循环时间执行中断子程序。在需要有别于 PLC 的运算周期的循环中断处理控制中使用。

- 不接受宽度窄的输入脉冲信号

可编程控制器输入的 ON/OFF 的时间宽度应比可编程控制器的循环时间长。若考虑输入滤波器的响应滞后为 10ms，循环时间为 10ms，则 ON/OFF 的时间分别需要 20 ms。因此，不能处理  $1,000 / (20+20) = 25\text{Hz}$  以上的输入脉冲。但是，若采用可编程控制器的特殊功能与应用指令，（如高速计数功能、输入中断功能、输入滤波器值调整等。）可改进这方面的情况。

● 二重输出（双线圈）的动作



如左图所示，考虑在多处使用同一个线圈 Y0 的情况：

例如，取X0=ON，X1=OFF

最初的 Y0 由于 X0 为 ON，其映像存储区为 ON，输出 Y1 也为 ON。

但是，第二次的 Y0，由于 X1 为 OFF，因此，其映像存储区被修改为OFF。

因此，实际外部输出为Y0=OFF，Y1=ON。

据此可知，执行二重输出时（使用双线圈），后侧的优先动作。

# 3 基本顺控指令说明

---

本章主要介绍 CZA 系列可编程控制器共用的基本顺控指令的种类及其功能。

3-1. 基本指令一览表

3-2. [LD], [LDI], [OUT]

3-3. [AND], [ANI]

3-4. [OR], [ORI]

3-5. [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]

3-6. [LDD], [LDDI]

3-7. [ORB]

3-8. [ANB]

3-9. [MCS], [MCR]

3-10. [ALT]

3-11. [PLS], [PLF]

3-12. [SET], [RST]

3-13. 针对计数器软元件的[OUT], [RST]

3-14. [END]

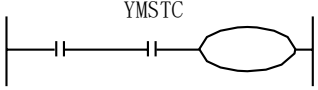
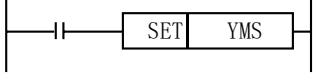
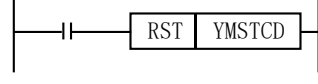
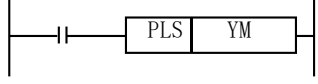
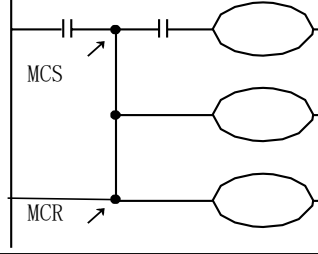

3-15. [GROUP], [GROUPE]

3-16. 编程注意事项



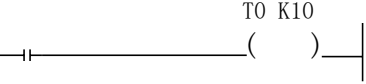
## 3-1. 基本指令一览表

CZA系列均支持基本指令。

助记符、名称	功能	回路表示和可用软元件
[LD]取	运算开始常开触点	XYMSTC 
[LDI]取反转	运算开始常闭触点	XYMSTC 
[LDP]取脉冲上升沿	上升沿检出运算开始	XYMSTC 
[LDF]取脉冲下降沿	下降沿检出运算开始	XYMSTC 
[AND]与	串联a触点	XYMSTC 
[ANI]与反转	串联b触点	XYMSTC 
[ANDP]与脉冲上升沿	上升沿检出串联连接	XYMSTC 
[ANDF]与脉冲下降沿	下降沿检出串联连接	XYMSTC 
[OR]或	并联a触点	XYMSTC 
[ORI]或反转	并联b触点	XYMSTC 
[ORP]或脉冲上升沿	上升沿检出并联连接	XYMSTC 
[ORF]或脉冲下降沿	下降沿检出并联连接	XYMSTC 
[ANB]回路块与	并联回路快的串联连接	
[ORB]回路快或	串联回路快的串联连接	

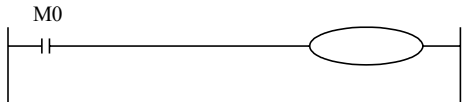
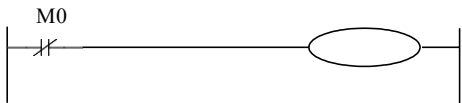
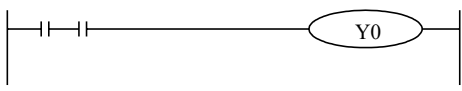
助记符、名称	功能	回路表示和可用软元件
[OUT] 输出	线圈驱动指令	
[SET] 置位	线圈接通保持指令	
[RST] 复位	线圈接通清除指令	
[PLS] 上升沿脉冲	上升沿检出指令	
[PLF] 下降沿脉冲	下降沿检出指令	
[MCS] 新母线开始	公共串联点的连接线圈指令	
[MCR] 母线复归	公共串联点的清除指令	
[ALT] 取反	运算结果的取反	
[NOP] 空操作	无动作	清除流程序或
[END] 结束	顺控程序结束	顺控顺序结束回到“0”



GROUP	指令块折叠开始无	
GROUPE	指令块折叠结束无	
TMR	定时	

## 3-2. [LD], [LDI], [OUT]

助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
LD 取	常开触点运算开始	 <p>操作元: X、Y、M、S、T、C</p>
LDI 取反	常闭触点运算开始	 <p>操作元: X、Y、M、S、T、C</p>
OUT 输出	线圈驱动	 <p>操作元: X、Y、M、S、T、C</p>

## 指令说明

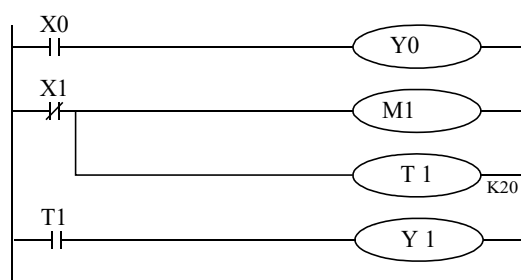
- LD, LDI 指令用于将触点连接到母线上。其他用法与后述的 ANB 指令组合, 在分支起点处也可使用。
- OUT 指令是对输出继电器、辅助继电器、状态、定时器、计数器的线圈驱动指令, 对输入继电器不能使用。

## 定时器、计数器的程序

- 对于定时器的计时线圈 T 或计数器的计数线圈 C, 使用 OUT 指令后, 必须设定常数 K 或寄存器 D。
- 常数 K 的设定范围、实际的定时器常数、相对于 OUT 指令的程序步数 (包括设定值)。如下表所示:

定时器, 计数器	K 的设定范围	实际的设定值
1ms 定时器	1~32,767	0.001~32.767 秒
10ms 定时器		0.01~327.67 秒
100ms 定时器		0.1~3276.7 秒
16 位计数器	1~32,767	同左
32 位计数器	1~2,147,483,647	同左



功能



LD	X0	
OUT	Y0	
LDI	X1	
OUT	M1	
OUT	T1	K20
LD	T1	
OUT	Y1	

## 3-3. [AND], [ANI]

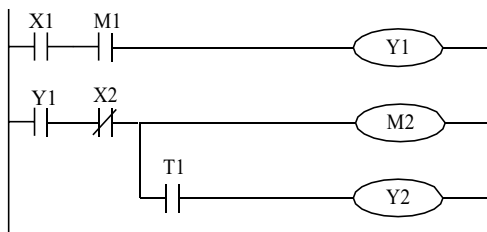
助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
AND 与	串联常开触点	 <p>操作元: X、Y、M、S、T、C</p>
ANI 与反	串联常闭触点	 <p>操作元: X、Y、M、S、T、C</p>

## 指令说明

- 用 AND、ANI 指令可串联连接一个触点。串联触点数量不受限制，该指令可多次使用。
- OUT 指令后，通过触点对其他线圈使用 OUT 指令，称之为纵接输出。（下图的 OUT M2 与 OUT Y3）。这样的纵接输出如果顺序不错，可重复多次。串联触点数量和纵接输出次数不受限制。

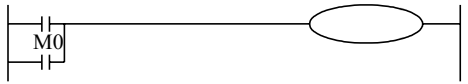
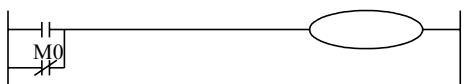
## 功能



LD	X1
AND	M1
OUT	Y1
LD	Y1
ANI	X2
OUT	M2
AND	T1
OUT	Y2

### 3-4. [OR], [ORI]

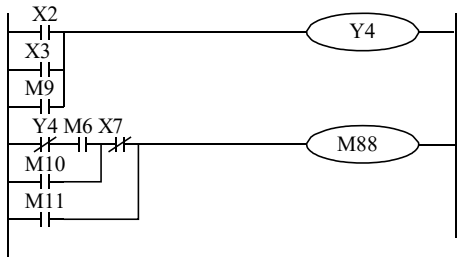
#### 助记符与功能

助记符、名称	功能	回路表示和可用软元件
OR 或	并联常开触点	 <p>操作元: X、Y、M、S、T、C</p>
ORI 或反转	并联常闭触点	 <p>操作元: X、Y、M、S、T、C</p>

#### 指令说明

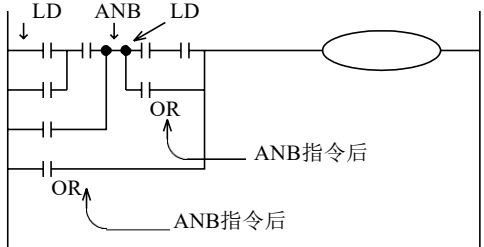
- OR, ORI 被用作一个触点的并联连接指令。如果有两个以上的触点串联连接, 并将这种串联回路块与其他回路并联连接时, 采用后述的 ORB 指令。
- OR, ORI 是指从该指令的步开始, 与前述的 LD、LDI 指令步, 进行并联连接。并联连接的次数不受限制。

#### 编程



LD	X2
OR	X3
OR	M9
OUT	Y4
LDI	Y4
AND	M6
OR	M10
ANI	X7
OR	M11
OUT	M88

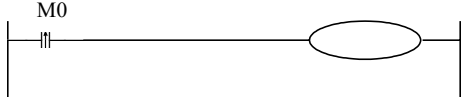
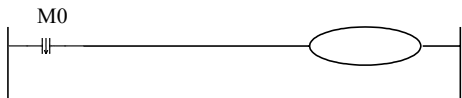
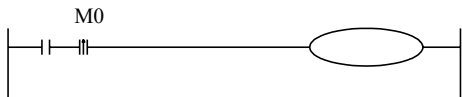
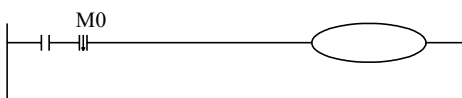

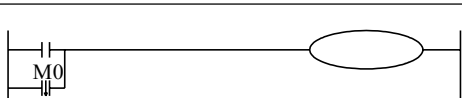
#### 与 ANB 的关系



使用 OR, ORI 指令所作的并联连接, 原则上是与前述的 LD, LDI 连接, 但在后述的 ANB 指令后, 则可在前面加一条 LD 或 LDI 指令连接。

## 3-5. [LDP], [LDF], [ANDP], [ANDF], [ORP], [ORF]

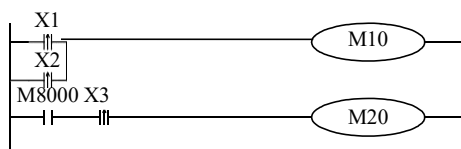
助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
LDP 取脉冲 上升沿	上升沿检出运算 开始	 <p>操作元: X、Y、M、S、T、C</p>
LDF 取脉冲 下降沿	下降沿检出运算 开始	 <p>操作元: X、Y、M、S、T、C</p>
ANDP 与脉冲 上升沿	上升沿检出串联 连接	 <p>操作元: X、Y、M、S、T、C</p>
ANDF 与脉冲 下降沿	下降沿检出串联 连接	 <p>操作元: X、Y、M、S、T、C</p>
ORP 或脉冲 上升沿	脉冲上升沿检出 并联连接	 <p>操作元: X、Y、M、S、T、C</p>
ORF 或脉冲 下降沿	脉冲下降沿检出 并联连接	 <p>操作元: X、Y、M、S、T、C、</p>

## 指令说明

- LDP、ANDP、ORP 指令是进行上升沿检出的触点指令，仅在指定位软元件的上升沿时（OFF→ON 变化时）接通一个扫描周期。
- LDF、ANDF、ORF 指令是进行下降沿检出的触点指令，仅在指定位软元件的下降沿时（ON→OFF 变化时）接通一个扫描周期。


## 编程



LDP	X1
ORP	X2
OUT	M10
LD	M8000
ANDP	X3
OUT	M20

**3-6. [ORB]**

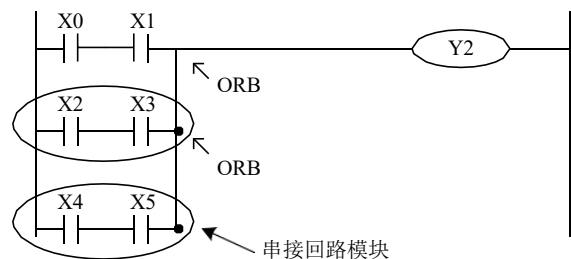
**助记符与功能**

助记符、名称	功能	回路表示和可用软元件
ORB 回路块 或	串联回路块的并联连接	 操作元：无

**指令说明**

- 由 2 个以上的触点串联连接的回路被称为串联回路块。将串联回路块并联连接时，分支开始用 LD, LDI 指令，分支结束用 ORB 指令。
- 如后述的 ANB 指令一样，ORB 指令是不带软元件编号的独立指令。
- 有多个并联回路时，如对每个回路块使用 ORB 指令，则并联回路没有限制。
- ORB指令也可成批使用，但由于LD、LDI指令重复的次数限制在8次以下，因此请务必注意。

**编程**



较佳的指令表程序写法：

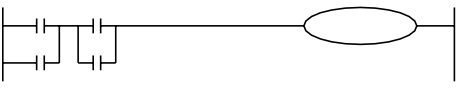
LD	X0
AND	X1
LD	X2
AND	X3
ORB	
LD	X4
AND	X5
ORB	
OUT	Y2

不佳的指令表程序写法：

LD	X0
AND	X1
LD	X2
AND	X3
LD	X4
AND	X5
ORB	
ORB	
OUT	Y2

**3-7. [ANB]**

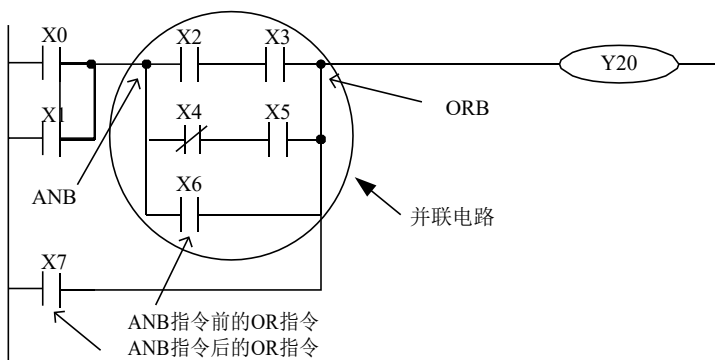
**助记符与功能**

助记符、名称	功能	回路表示和可用软元件
ANB 回路块与	并联回路块的串联连接	 <p>操作元：无</p>

**指令说明**

- 当分支回路（并联回路块）与前面的回路串联连接时，使用 ANB 指令。分支的起点用 LD、LDI 指令，并联回路块结束后，使用 ANB 指令与前面的回路串联连接。
- 若多个并联回路块按顺序和前面的回路串联时，ANB 指令的使用次数没有限制。也可成批的使用 ANB 指令，但在这种场合，与 ORB 指令一样，请务必注意 LD，LDI 指令使用次数限制在 8 次以下。

**编程**


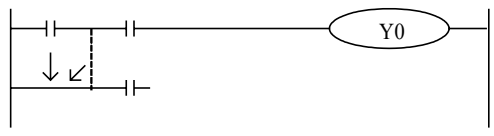


LD	X0	
OR	X1	
LD	X2	┌───┐ 分支起点
AND	X3	
LDI	X4	
AND	X5	
ORB		└───┘ 并联电路块结束
OR	X6	
ANB		
OR	X7	└───┘ 与前面的电路串联
OUT	Y20	



### 3-8. [MCS], [MCR]

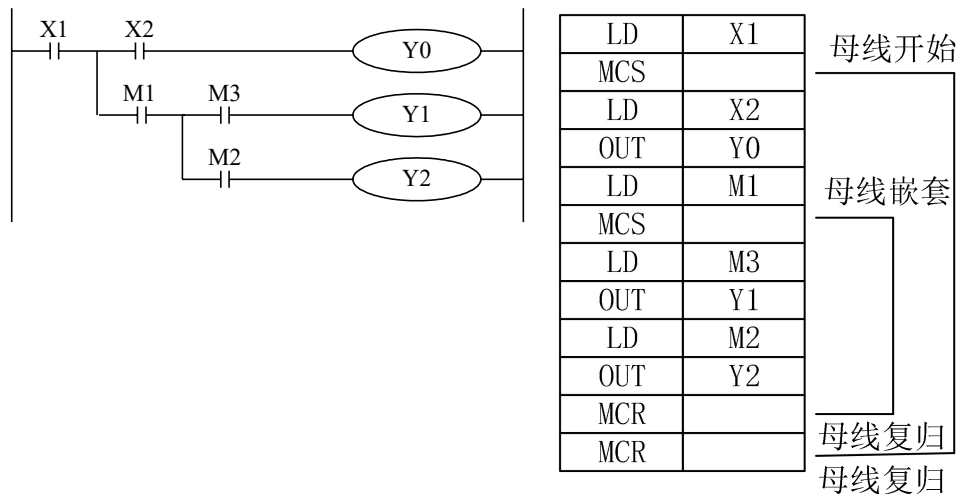
#### 助记符与功能

助记符、名称	功能	回路表示和可用软元件
MCS 主控	新母线开始	 <p>操作元: 无</p>
MCR 主控复位	母线复归	 <p>操作元: 无</p>

#### 指令说明

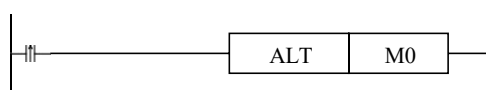
- 执行 MCS 指令后，母线（LD、LDI）向 MCS 接点后移动，将其返回到原母线的指令为 MCR。
- MCS、MCR 指令需配对使用。
- 母线可以嵌套使用，在配对的 MCS、MCR 指令之间使用配对的 MCS、MCR 指令，嵌套级随着 MCS 的使用逐个增加，嵌套级最大为10级。执行 MCR 指令时，返回到上一级母线。
- 在使用流程序时，母线管理只能用于同一个流程中；在结束某个流程时，必须返回到主母线。

#### 编程



## 3-9. 位取反[ALT]

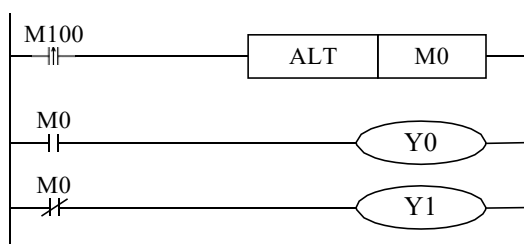
助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
ALT 取反	运算结果取反	 操作元： Y、M、S、T、C

## 指令说明

执行ALT后可以将线圈的状态取反。由原来的 ON 状态变成OFF 状态，或由原来的 OFF 状态变成 ON 状态, 每个扫描周期执行一次。


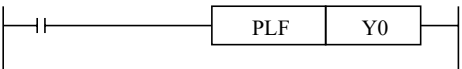
## 编程



LDP	M100
ALT	M0
LD	M0
OUT	Y0
LDI	M0
OUT	Y1

### 3-10. [PLS], [PLF]

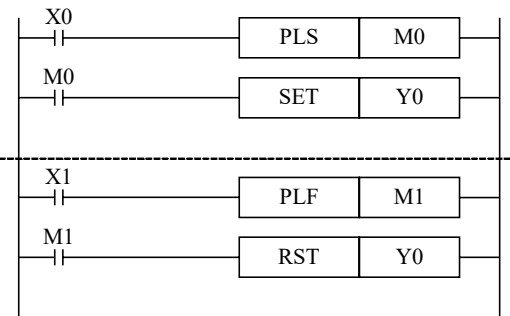
#### 助记符与功能

助记符、名称	功能	回路表示和可用软元件
PLS 上升沿脉冲	上升沿微分输出	 <p>操作元: Y、M、S、T、C</p>
PLF 下降沿脉冲	下降沿微分输出	 <p>操作元: Y、M、S、T、C</p>

#### 指令说明

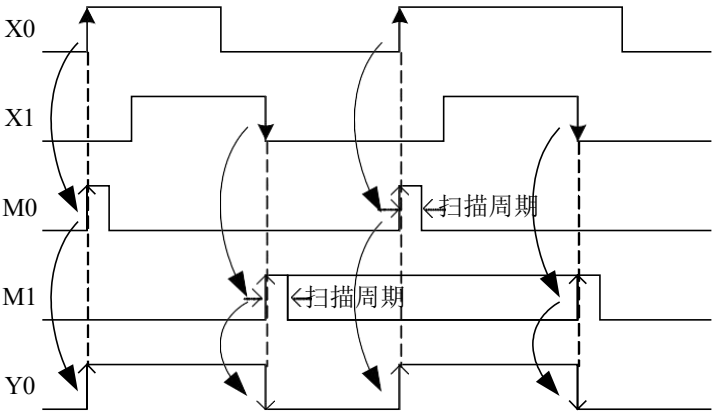
- 使用 PLS 指令时, 仅在驱动输入为 ON 后的一个扫描周期内, 软元件 Y, M 动作。
- 使用 PLF 指令时, 仅在驱动输入为 OFF 后的一个扫描周期内, 软元件 Y, M 动作。

#### 编程



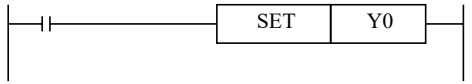
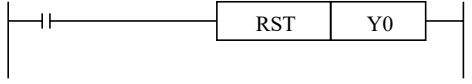
LD	X0
PLS	M0
LD	M0
SET	Y0

LD	X1
PLF	M1
LD	M1
RST	Y0



## 3-11. [SET], [RST]

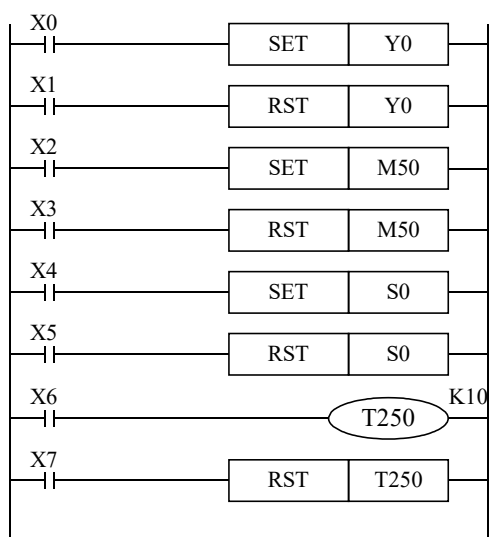
助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
SET 置位	动作保持	 操作元: Y、M、S
RST 复位	动作复位, 寄存器当前值清零	 操作元: Y、M、S、T、C、D、V、Z

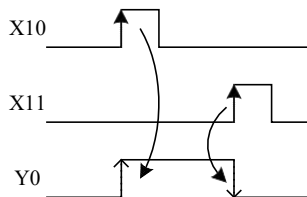
## 指令说明

- 在下述程序示例中, X10 一旦接通后, 即使它再断开, Y0 仍继续动作。X11 一旦接通时, 即使它断开, Y0 仍保持不被驱动。对于 M、S 也是一样的。
- 对于同一软元件, SET、RST 可多次使用, 顺序也可随意, 但最后执行者有效。
- 此外, 数据寄存器、计数器当前值的复位以及触点复位也可使用 RST 指令。
- 使用 SET、RST 指令时, 避免与 OUT 指令使用同一定义号。

## 编程

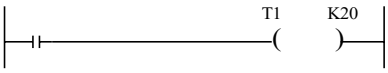
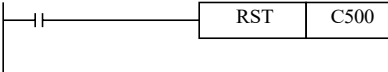


LD	X0	
SET	Y0	
LD	X1	
RST	Y0	
LD	X2	
SET	M50	
LD	X3	
RST	M50	
LD	X4	
SET	S0	
LD	X5	
RST	S0	
LD	X6	
OUT	T250	K10
LD	X7	
RST	T250	

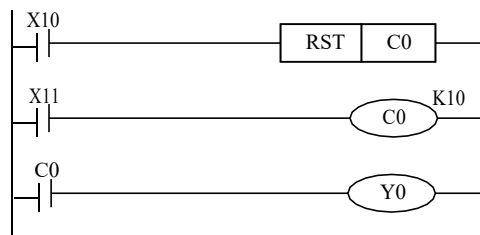


**3-12. 对应计数器软元件的[OUT], [RST]**

助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
OUT 输出	计数线圈的驱动	 操作元: K、D
RST 复位	输出触点的复位, 当前值清零	 操作元: C

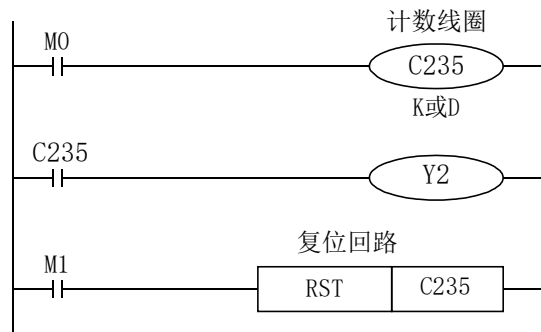
内部计数器  
的编程



C0 对 X11 的 OFF→ON 次数进行加计数, 当 C0 达到设定值 K10 时, 输出触点 C0 动作, 即 C0 的状态由 OFF 转变为 ON。此后, 当 X11 继续由 OFF→ON 变化时, 计数器的当前值会继续递加, 输出触点仍保持动作。为了将此清除, 令 X10 为接通状态, 使输出触点复位。有必要在 OUT C0 指令后面指定常数 K 或间接设定用数据寄存器的编号。

停电保持用计数器, 即使在停电时, 仍保持当前值以及输出触点的动作状态和复位状态。

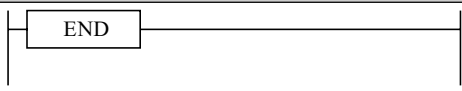
高速计数器的  
编程



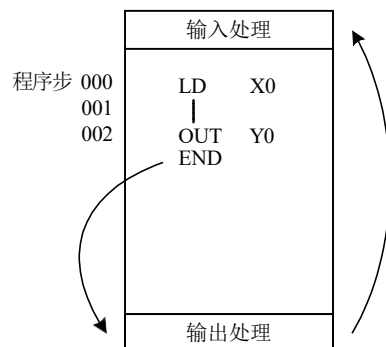
- 如上图例, 对 M0 的 OFF→ON 进行增计数。
- 计数器的当前值增加, 在达到设定值 (K 或 D 的内容) 时, 输出触点被置位。
- M1 为 ON 时, 计数器 C235 的输出触点复位, 计数器的当前值也变为 0。

## 3-13. [END]

助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
END 结束	输入输出处理以及返回到第 0 步	 操作元：无

## 指令说明



可编程控制器反复进行输入处理、程序执行和输出处理。若在程序的最后写入END指令，则END以后的其余程序步不再执行，而直接进行输出处理。在程序中没有END指令时，CZA可编程控制器一直处理到最终的程序步，然后从0步开始重复处理。在调试阶段，在各程序段插入END指令，可依次检出各程序段的动作。这时，在确认前面回路块动作正确无误后，依次删去END指令。

此外，RUN开始时的首次执行，从执行END指令开始。执行END指令时，也刷新监视定时器。

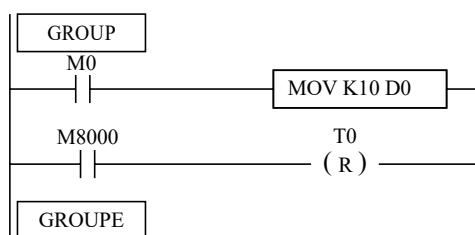
## 3-14. [GROUP], [GROUPE]

助记符  
与功能

助记符、名称	功能	回路表示和可用软元件
GROUP	指令块折叠开始	<div style="border: 1px solid black; display: inline-block; padding: 2px;">GROUP</div> 操作元：无
GROUPE	指令块折叠结束	<div style="border: 1px solid black; display: inline-block; padding: 2px;">GROUPE</div> 操作元：无

## 指令说明

- GROUP 和 GROUPE 指令必须成对使用。
- GROUP 和 GROUPE 指令并不具有实际意义，仅是对程序的一种结构优化，因此该组指令添加与否，并不影响程序的运行效果。
- GROUP 和 GROUPE 指令的使用方法与流程指令类似，在折叠语段的开始部分输入 GROUP 指令，在折叠语段的结束部分输入 GROUPE 指令。



GROUP 和 GROUPE 指令一般可根据指令段的功能的不同进行编组，同时，被编入的程序可以折叠或展开显示，对于程序冗长的工程，该组指令将特别适用。

### 3-15. 编程注意事项

#### 1、触点的结构与步数

即使在动作相同的顺控回路中，根据触点的构成方法也可简化程序与节省程序步数。一般编程的原则是：a) 将串联触点多的回路写在上方；b) 将并联触点多的回路写在左方。

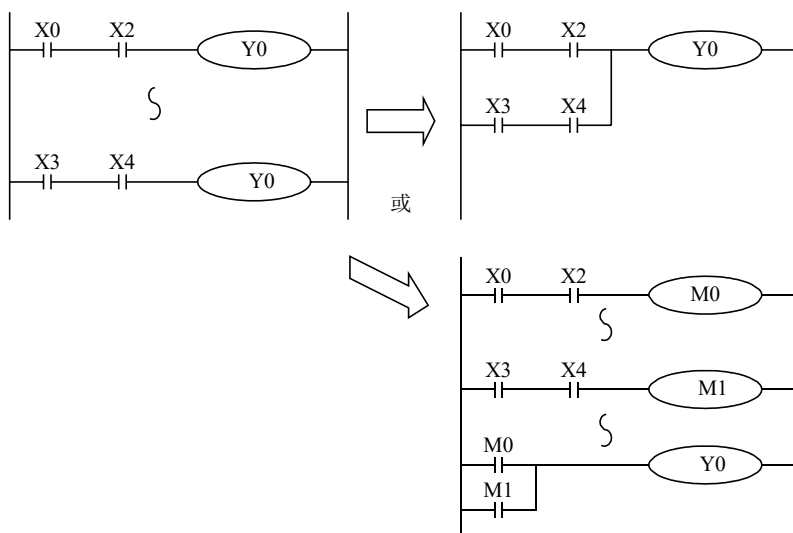
#### 2、程序的执行顺序

对顺控程序作【自上而下】和【自左向右】的处理。

顺控指令清单也沿着此流程编码。

#### 3、双重输出双线圈动作及其对策

- 若在顺控程序中进行线圈的双重输出（双线圈），则后面的动作优先执行。
- 双重输出（双线圈）在程序方面并不违反输入规则，但是由于上述的动作十分复杂，因此请按以下的示例改变程序。



- 综上编写程序时尽量避免线圈的双重输出。还有其他的方法，如采用跳转指令，或流程指令，不同流程控制同一输出线圈编程的方法。



# 4 应用指令说明

---

本章主要介绍 CZA 系列可编程控制器的应用指令的种类及其功能。

4-1. 应用指令一览表

4-2. 应用指令的表示方法

4-3. 程序流程指令

4-4. 触点比较指令

4-5. 数据传送指令

4-6. 数据运算指令

4-7. 数据移位指令

4-8. 数据转换指令

4-9. 浮点运算指令

4-10. 时钟指令

## 4-1. 应用指令一览表

指令助记符	功能	章节	指令助记符	功能	章节
<b>程序流程</b>			<b>数据比较</b>		
CJ	条件跳转	4-3-1	LD=	开始 (S1) = (S2) 时导通	4-4-1
CALL	子程序调用	4-3-2	LD>	开始 (S1) > (S2) 时导通	4-4-1
SRET	子程序返回	4-3-2	LD<	开始 (S1) < (S2) 时导通	4-4-1
EI	允许中断	4-3-3	LD<>	开始 (S1) <> (S2) 时导通	4-4-1
DI	禁止中断	4-3-3	LD<=	开始 (S1) <= (S2) 时导通	4-4-1
IRET	中断返回	4-3-3	LD>=	开始 (S1) >= (S2) 时导通	4-4-1
STL	流程开始	4-3-4	AND=	串联 (S1) = (S2) 时导通	4-4-2
SET	打开指定流程, 关闭所在流程	4-3-4	AND>	串联 (S1) > (S2) 时导通	4-4-2
RET	流程结束	4-3-4	AND<	串联 (S1) < (S2) 时导通	4-4-2
FOR	循环范围开始	4-3-5	AND<>	串联 (S1) <> (S2) 时导通	4-4-2
NEXT	循环范围结束	4-3-5	AND<=	串联 (S1) <= (S2) 时导通	4-4-2
FEND	主程序结束	4-3-6	AND>=	串联 (S1) >= (S2) 时导通	4-4-2
END	程序结束	4-3-6	OR=	并联 (S1) = (S2) 时导通	4-4-3
WDT	监控定时器	4-3-7	OR>	并联 (S1) > (S2) 时导通	4-4-3
<b>数据传送</b>			OR<	并联 (S1) < (S2) 时导通	4-4-3
CMP	数据的比较	4-5-1	OR<>	并联 (S1) <> (S2) 时导通	4-4-3
ZCP	数据的区间比较	4-5-2	OR<=	并联 (S1) <= (S2) 时导通	4-4-3
MOV	传送	4-5-3	OR>=	并联 (S1) >= (S2) 时导通	4-4-3
BMOV	数据块传送	4-5-4			
SMOV	移位传送	4-5-5	<b>数据移位</b>		
FMOV	多点重复传送	4-5-6	ROL	左回转	4-7-1
ZRST	批次复位	4-5-7	ROR	右回转	4-7-1
SWAP	高低字节交换	4-5-8	RCL	带进位循环左移	4-7-2
XCH	两个数据交换	4-5-9	RCR	带进位循环右移	4-7-2
<b>数据运算</b>			SFWR	移位写入	4-7-3
ADD	加法	4-6-1	SFRD	移位读出	4-7-3
SUB	减法	4-6-2	SFTL	位左移	4-7-4

#### 4 应用指令说明

MUL	乘法	4-6-3	SFTR	位右移	4-7-5
DIV	除法	4-6-4	WSFL	字左移	4-7-6
INC	自加1	4-6-5	WSFR	字右移	4-7-6
DEC	自减1	4-6-5	<b>浮点运算</b>		
MEAN	求平均值	4-6-6	ECMP	浮点数比较	4-9-1
WAND	逻辑与	4-6-7	EZCP	浮点数区间比较	4-9-2
WOR	逻辑或	4-6-7	EADD	浮点数加法	4-9-3
WXOR	逻辑异或	4-6-7	ESUB	浮点数减法	4-9-4
CML	取反	4-6-8	EMUL	浮点数乘法	4-9-5
NEG	求补	4-6-9	EDIV	浮点数除法	4-9-6
<b>数据转换</b>			ESQR	浮点数开方	4-9-7
FLT	16位整数转浮点	4-8-1	SIN	浮点数正弦运算	4-9-8
DFLT	32位整数转浮点	4-8-1	COS	浮点数余弦与运算	4-9-9
INT	浮点转整数	4-8-2	TAN	浮点数正切运算	4-9-10
DECO	译码	4-8-3			
ENCO	高位编码	4-8-4			
<b>时钟</b>					
TRD	时钟数据读取	4-10-1			
TWR	时钟数据写入	4-10-2			
TCMP	时钟数据比较	4-10-3			
TZCP	时钟区间比较	4-10-4			
TADD	时钟数据加法	4-10-5			
TSUB	时钟数据减法	4-10-5			

## 4-2. 应用指令的阅读方法

本手册中所记录的应用指令按以下形式进行说明。

### 1、指令概述

加法运算[ADD]			
16 位	ADD	32 位	DADD
执行条件	常开/闭、边沿触发	适用机型	CZA

### 2、操作数

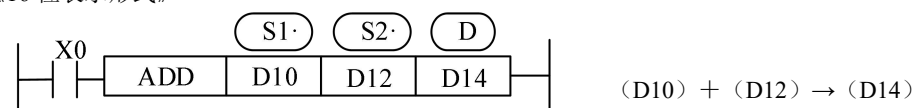
操作数	作用	数据类型
S1	指定进行加法运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行加法运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存加法结果的软元件地址编号	16 位/32 位, BIN

### 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√		
位软元件	操作数	系统								
		X	Y	M	S	T	C			

### 功能和动作

《16 位表示形式》



《32 位表示形式》



- 两个源数据进行二进制加法后传递到目标处。各数据的最高位是正（0）、负（1）符号位，这些数据以代数形式进行加法运算（5+（-8）=-3）。
- 运算结果为 0 时，0 标志会动作。如运算结果超过 32,767（16 位运算）或 2,147,483,647（32 位运算）时，进位标志会动作（参照下一页）。如运算结果超过-32,768（16 位运算）或-2,147,483,648（32 位运算）时，借位标志会动作（参照下一页）。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。

- 可以将源操作数和目标操作数指定为相同的软元件编号。
- 上例中驱动输入 X0 为 ON 时，每个扫描周期执行一次加法运算，请务必注意。

### 相关软元件

标志位的作用（动作及数值涵义）

软元件	名称	作用
M8020	零	ON: 运算结果为 0 时 OFF: 运算结果为 0 以外时
M8021	借位	ON: 运算结果超出-32,767（16 位运算）或是-2,147,483,647（32 位运算）时，借位标志位动作。 OFF: 运算结果不到-32,768（16 位运算）或是-2,147,483,648（32 位运算）时。
M8022	进位	ON: 运算结果超出 32,767（16 位运算）或是 2,147,483,647（32 位运算）时，进位标志位动作。 OFF: 运算结果不到 32,767（16 位运算）或是 2,147,483,647（32 位运算）时。

### 相关说明

- 数据的指定  
CZA 可编程控制器的数据寄存器为单字（16 位）数据寄存器，单字数据只占用一个数据寄存器，为单字对象指令指定的数据寄存器，处理范围为十进制-327,68~327,67 或十六进制 0000~FFFF。



双字（32 位）占用 2 个数据寄存器，由双字对象指令指定的数据寄存器及其下一个编号的数据寄存器组成，处理范围为十进制 -214,748,364,8~214,748,364,7 或十六进制 00000000~FFFFFFFF。



- 32 位指令的表示方法  
对于 16 位指令，其相应的 32 位指令的表示方法就是在该指令前面加“D”。  
例如：ADD D0 D2 D4 表示两个 16 位的数据相加；  
DADD D10 D12 D14 则表示两个 32 位的数据相加。

注：(S) 表示该操作数的内容不随指令的执行而变化，称之为源操作数。

(D) 表示该操作数的内容随指令的执行而变化，称之为目标操作数。

**4-3. 程序流程指令**

指令助记符	指令功能	章节
CJ	条件跳转	4-3-1
CALL	子程序调用	4-3-2
SRET	子程序返回	4-3-2
EI	允许中断	4-3-3
DI	禁止中断	4-3-3
IRET	中断返回	4-3-3
STL	流程开始	4-3-4
SET	打开指定流程，关闭所在流程（流程跳转）	4-3-4
RET	流程结束	4-3-4
FOR	循环范围开始	4-3-5
NEXT	循环范围结束	4-3-5
FEND	主程序结束	4-3-6
END	程序结束	4-3-6
WDT	监控定时器	4-3-7

### 4-3-1. 条件跳转[CJ]

#### 1、指令概述

CJ 作为执行序列一部分的指令，可以缩短运算周期及使用双线圈。

条件跳转[CJ]			
16 位指令	CJ(P)	32 位指令	-
执行条件	常开/闭线圈触发	适用机型	CZA

#### 2、操作数

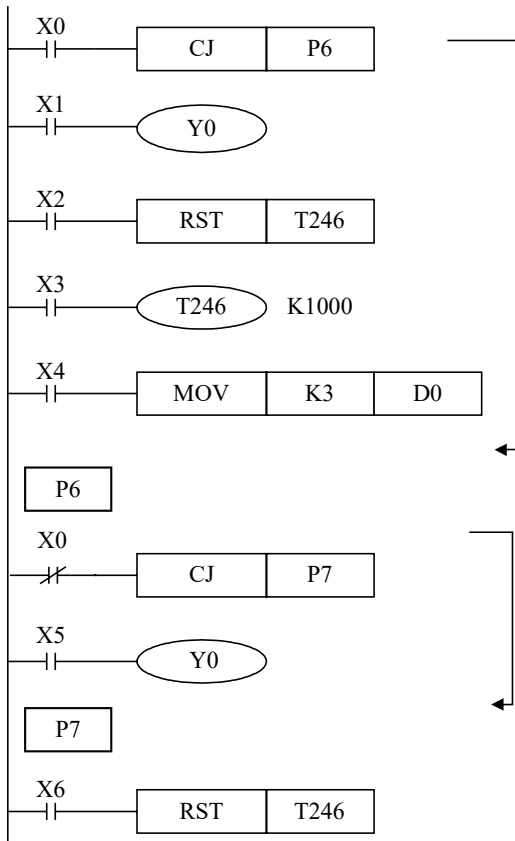
操作数	作用	类型
Pn	跳转到目标标记的指针编号P（P0~P9999）	指针编号

#### 3、适用软元件

其他	指针	
	P	I
	√	

#### 功能和动作

在下图的示例中，如果 X0 “ON”，则从第 1 步跳转到标记 P6 的后一步。X0 “OFF” 时，不执行跳转指令。



- 如左图，Y0 变成双线圈输出，但是，X0=OFF 时采用 X1 动作，即 LD X5 OUT Y0 不在扫描周期内。X0=ON 时采用 X5 动作，即 LD X1 OUT Y0 不在扫描周期内。
- CJ 不可以从一个 STL 跳转到另一个 STL。
- 程序定时器 T0~T255 及高速计数 C235、C237、C251、C253 如果在驱动后执行了 CJ 指令，则继续工作，输出接点也动作。
- 跳转指令 CJ 和 CJP 的主要区别：CJ 是连续执行指令，即在条件满足时，每个扫描周期执行一次该指令。CJP 是脉冲执行指令，仅在条件满足时执行一次该指令。
- 使用跳转指令时注意标号一定要匹配。没有执行跳转时，整体程序都会受到扫描周期影响，一旦执行跳转，跳转步与跳转标号其间程序不在扫描周期内。

## 4-3-2. 子程序调用[CALL]/子程序返回[SRET]

## 1、指令概述

调用要共同处理的程序，可减少程序的步数。

子程序调用[CALL]			
16 位	CALL(P)	32 位	-
执行条件	常开/闭、边沿触发	适用机型	CZA
子程序返回[SRET]			
16 位	SRET	32 位	-
执行条件	-	适用机型	CZA

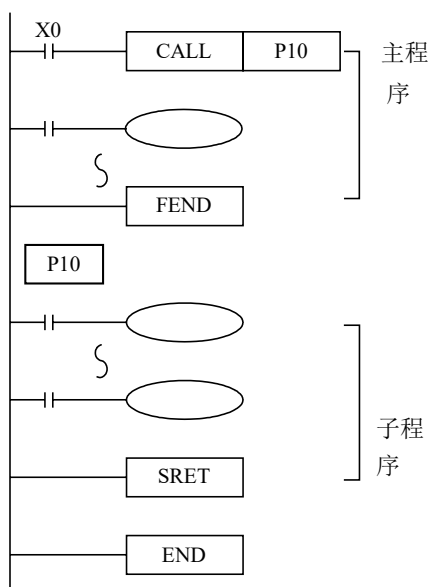
## 2、操作数

操作数	作用	类型
Pn	跳转到目标标记的指针编号P (P0~P127)	指针编号

## 3、适用软元件

其他	指针	
	P	I
	√	

## 功能和动作



- 如果 X0=“ON”，则执行调用指令，跳转到标记为 P10 的子程序步。在这里，执行子程序后，通过执行 SRET 指令后，返回到原来的主程序步，接着继续执行后续的主程序。
- 在后述的 FEND 指令后对标记编程。
- 在子程序内可以允许有 9 次调用指令，整体而言可做 10 层嵌套。
- 调用子程序时，主程序所属的 OUT、PLS、PLF、定时器等均保持。

- 子程序返回时，子程序所属的OUT、PLS、PLF 等均保持。

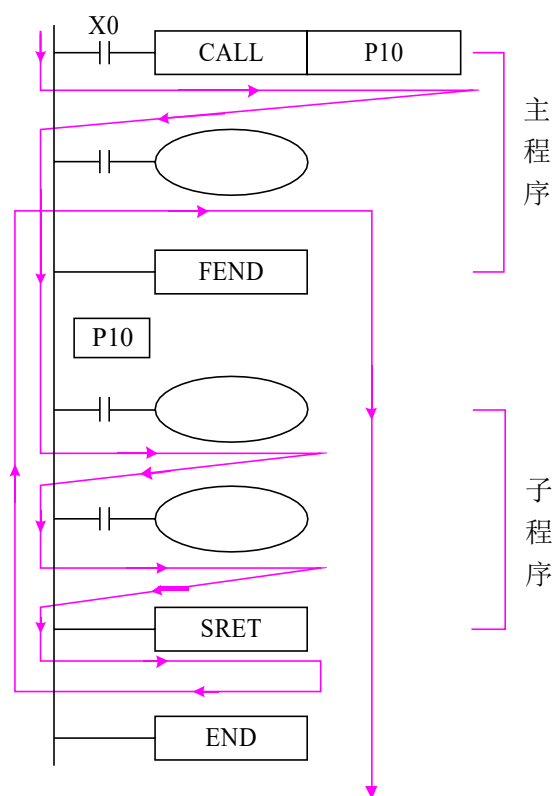
请勿将脉冲、定时、计数等一个扫描周期内无法完成的指令放在子程序中。

程序刚下完时只有主程序受扫描周期影响，执行调用时，跳转子程序受扫描周期影响，先后顺序是主程序先到调用指令，然后到调用子程序，最后又返回执行完主程序，只要执行调用会一直反复如此扫描。

- CALL和CALLP指令的区别：CALL前面为ON的情况下，就会一直执行子程序。CALLP只有在指令前面由OFF变成ON了，才执行一次子程序。



子程序执行图示说明：



如果 X0=“ON”，则按照图示箭头方向执行。

如果 X0=“OFF”，则不执行（CALL 指令）调用指令，只执行主程序部分。

子程序使用，写程序时注意格式：必须在 FEND 指令后对标记编程。Pn 作为一段子程序的开始，以 SRET 作为一段子程序的结束。用 CALL Pn 调用子程序。其中 n 可以为 0~127 中的任一值。

使用子程序调用，可以简化编程，可以将几个地方需要用的公共部分写在子程序中，再调用子程序即可实现。

## 4-3-3. 中断[EI]、[DI]、[IRET]

## 1、指令概述

用于执行允许中断，禁止中断和中断返回的指令。

允许中断[EI]	适用对象： CZA
不需要驱动接点的单独指令。	
对中断许可，无对象因素。	
禁止中断[DI]	
不需要驱动接点的单独指令。	
对禁止中断，无对象因素。不需要中断禁止区间时，不必对DI指令编程。	
中断返回[IRET]	
不需要驱动接点的单独指令。	
对中断返回许可，无对象因素。	

CZA系列 PLC 都具有中断功能，中断功能分两种：一、外部中断；二、定时中断。通过中断功能可以处理一些特定的程序，它不受可编程控制器的扫描周期的影响。

## 外部中断

输入端子X 可以作为外部中断的输入用，每一输入端对应于一个外部中断，输入的上升沿或者下降沿都可触发中断，中断子程序写在主程序之后（FEND 命令之后）。当产生中断后，主程序立即停止执行，转而执行相应的中断子程序，等中断子程序执行完成后，再立即返回继续执行主程序。当禁止中断指令线圈置位后，外部中断将不会执行。

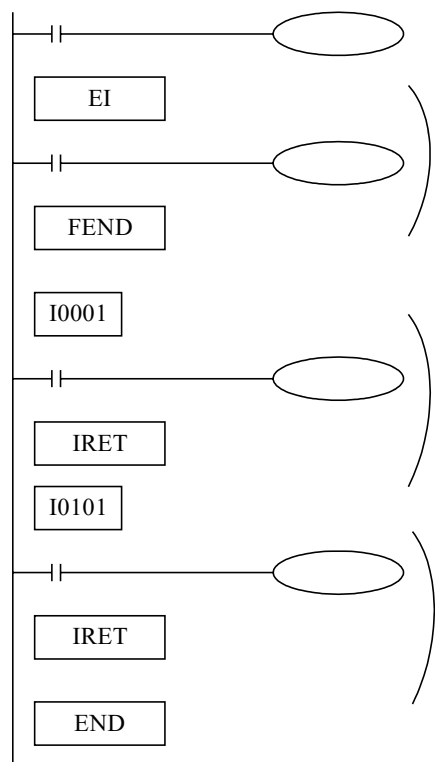
## 1) 外部中断端口定义

输入端子	指针编号		禁止中断指令
	上升中断	下降中断	
X0	I0001	I0000	M8050
X1	I0101	I0100	M8051
X4	I0401	I0400	M8054
X5	I0501	I0500	M8055

## 2) 中断指令

允许中断[EI]、禁止中断[DI]、中断返回[IRET]

注意点：①在外部中断时，在主程序没有进行复位处理的情况下，子程序若为LD M8000, OUT C0 K9999,则无论进多少次中断C0只会累计一次。若为LD M8000, OUT INC D200,则进一次中断D200就会累计一次。若为LD M8000, OUT Y5, 则进一次中断则Y5一直置ON。



中断许可范围

- 如果EI指令允许中断，则在扫描到中断输入触发上升沿时，执行中断程序①、②，结束后回到初始主程序。

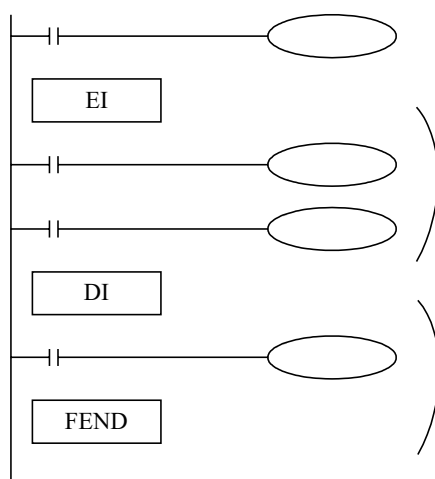
中断①

- 中断用指针（I\*\*\*），必须在FEND指令后作为标记编程。
- 可编程控制器平时呈允许中断状态。

中断②

- 在中断子程序里，只能写简单的置位、复位、传送、运算等可以在一个扫描周期内执行，其他如发脉冲、定时（精准定时除外）、通讯等需要持续执行的指令均不支持。

### 3) 中断范围的限制



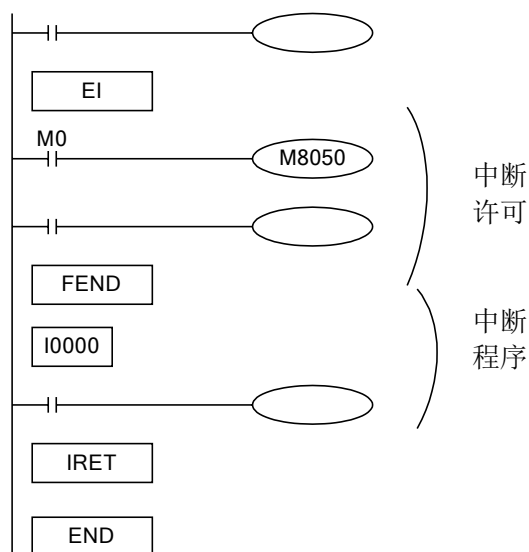
中断许可

- 通过对DI指令编程，可以设定中断禁止区间。
- 在EI~DI区间允许中断输入。

中断禁止

- 不需要中断禁止时，请仅对EI指令编程，无需对DI指令编程。

### 4) 中断禁止



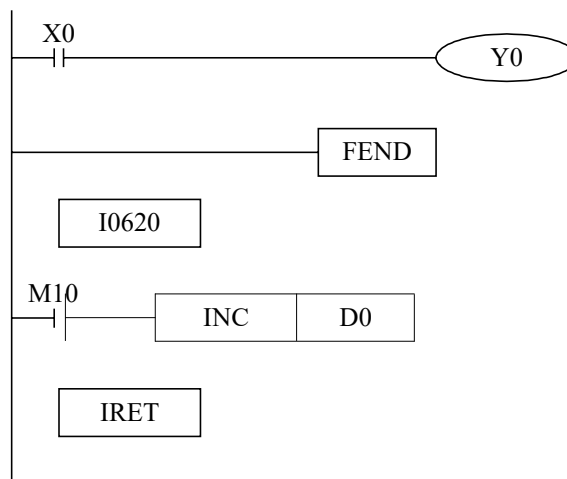
- 对于每个输入中断，分别配有禁止中断的特殊继电器（M8050~M8059）。

- 左图的程序示例中，如果用 M0 使 M8050 “ON”，则禁止 X0 中断输入。

### 定时中断

#### 1) 功能和动作

在主程序的执行周期很长的情况下，如果要处理特定的程序；或者在顺控扫描中，需要每隔一段时间执行特定的程序时，定时中断功能非常适用。它可以不受可编程控制器的扫描周期的影响，每隔 10ms~99ms 执行定时中断子程序。



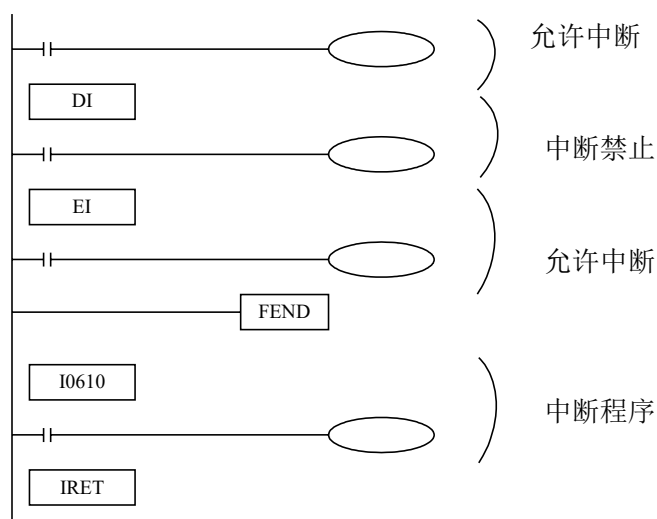
- 定时中断默认是打开状态，定时中断子程序类似其他中断子程序，必须写在主程序之后，以 I06xx 指令开始，结束于 IRET。
- 如上图，每隔 20ms 执行一次中断子程序。

2) 中断序号

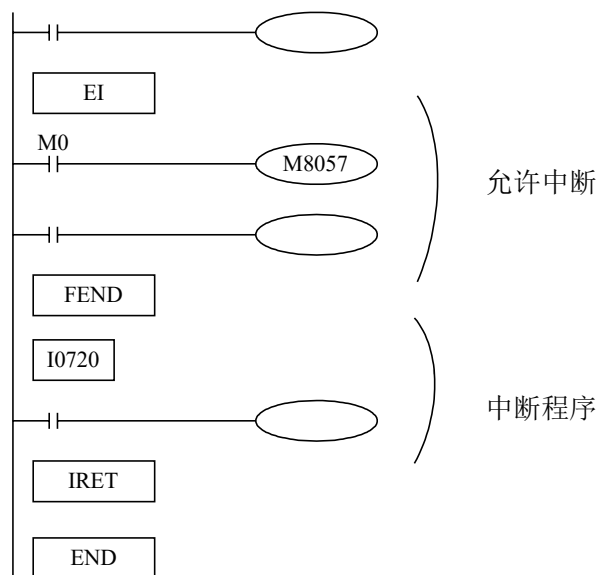
中断序号	中断禁止指令	说明
I06**	M8056	“**”表示定时中断的时间，范围10~99，单位“毫秒”
I07**	M8057	
I08**	M8058	

- 指针号为（I06、I07、I08）的指针不可重复使用。
- M8056~M8058在编程中接通时，与此对应的定时器中断被禁止。

3) 中断范围的限制



3) 中断禁止



如上图允许中断程序中M0驱动时，则中断程序则被禁止执行

## 4-3-4. 流程[SET]、[STL]、[RET]

## 1、指令概述

用于指定流程开始、结束、打开、关闭的指令。

打开指定流程，关闭所在流程[SET]			
16 位	SET	32 位	-
执行条件	常开/闭、边沿触发	适用机型	CZA
流程开始[STL]			
16 位	STL	32 位	-
执行条件	-	适用机型	CZA
流程结束[RET]			
16 位	RET	32 位	-
执行条件	-	适用机型	CZA

## 2、操作数

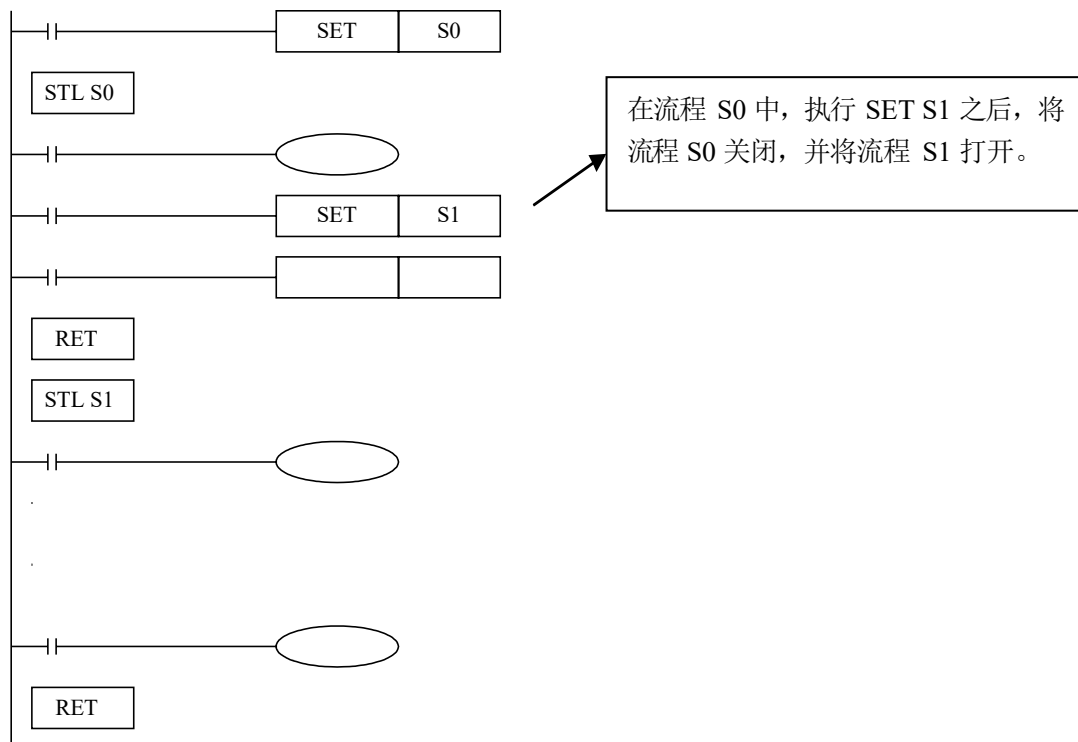
操作数	作用	类型
Sn	指定跳转到目标流程 S	流程编号

## 3、适用软元件

位软元件	操作数	系统					
		X	Y	M	S	T	C
	Sn				√		

## 功能和动作

- SET表示打开某一个流程，STL 表示一个流程的开始。
- 每一个流程书写都是独立的，写法上不能嵌套书写。在流程执行时，不一定要按 S0、S1、S2.....的顺序执行，流程执行的顺序在程序中可以按需求任意指定。可以先执行 S10 再执行 S5，再执行 S0。
- 执行 SET Sxxx 指令后，这些指令指定的流程为 ON。
- 执行 RST Sxxx 指令后，指定的流程为 OFF。
- 在流程 S0 中，SET S1 将所在的流程 S0 关闭，并将流程 S1 打开。
- 流程从 ON 变为 OFF 时，将流程所属的 OUT、PLS、PLF、不累计定时器等 OFF 或复位；SET、累计定时器等保持原有状态。
- 关闭流程前，需要先把流程里执行 SET 的线圈复位，且不允许在关闭流程后再对线圈进行复位。
- 在流程中执行 SET Sxxx 指令后，跳转到下一个流程，原流程中的脉冲指令也会关掉（包括单段、多段、相对绝对、原点回归）



## 4-3-5. 循环 [FOR]、[NEXT]

## 1、指令概述

以指定的次数对由 FOR 到 NEXT 之间的程序进行循环执行。

循环范围开始[FOR]			
16 位	FOR	32 位	-
执行条件	边沿触发	适用机型	CZA
循环范围结束[NEXT]			
16 位	NEXT	32 位	-
执行条件	常开/闭、边沿触发	适用机型	CZA

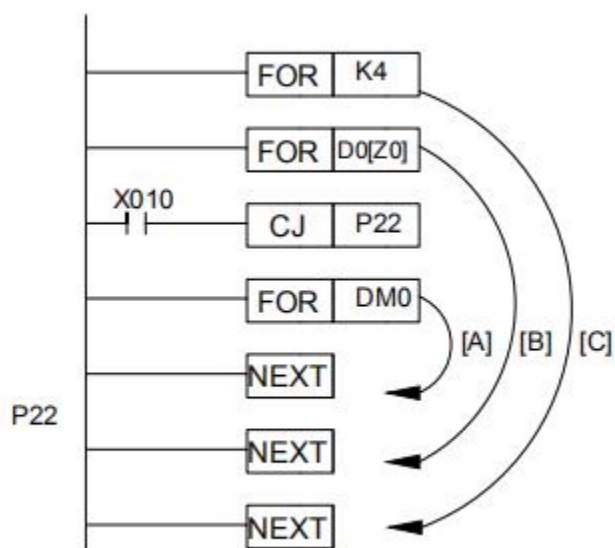
## 2、操作数

操作数	作用	数据类型
S	FOR~NEXT 之间程序循环执行的次数	16 位，二进制

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DS	DM	T	C	D	V、Z	K、H
S		√	√	√	√	√	√	√	√	√

## 功能和动作



- FOR、NEXT 必需配对使用，可以嵌套，嵌套层数最多为 8 层。
- 当 FOR~NEXT 指令之间的程序被执行指定次数（利用源数据指定的次数）后，才会处理 NEXT 指令后的程序。
- FOR/NEXT 指令 是不需要驱动触点的单一指令。
- [C] 的程序执行 4 次后向 NEXT 指令③以后的程序转移。



- 若在[C]的程序执行一次的过程中，数据寄存器D0[Z0]的内容为6，则[B]的程序执行6次。因此，[B]的程序合计一共被执行了24次。
- 若不想执行FOR~NEXT间的程序时，利用CJ指令，使之跳转。

(X10=0N)

当X10为OFF时，例如：DM0的内容为7，则在[B]的程序执行一次的过程中[A]被执行了7次，总计被执行了 $4*6*7=168$ 次。这样一共可以嵌套5层。

- 循环次数多时扫描周期会延长,有可能会出出现监视定时器错误，请务必注意。

## 4-3-6. 结束 [FEND]、[END]

## 1、指令概述

FEND 表示主程序结束，而 END 则表示程序结束。

主程序结束[FEND]			
指令形式	FEND		
执行条件	-	适用机型	CZA
程序结束[END]			
指令形式	END		
执行条件	-	适用机型	CZA

## 2、操作数

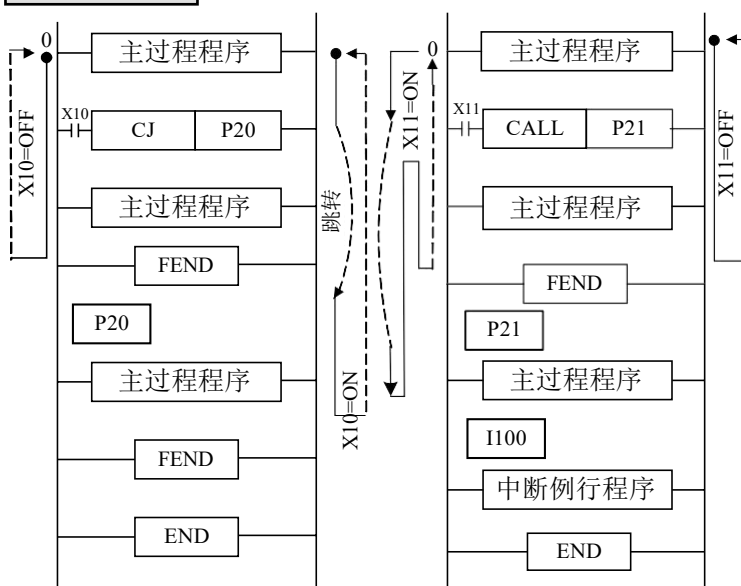
操作数	作用	数据类型
无	-	-

## 3、适用软元件

无
---

## 功能和动作

虽然[FEND]指令表示主程序的结束，但若执行此指令，则与 END 指令同样，执行输出处理、输入处理、监视定时器的刷新、向 0 步程序返回。



- CALL 指令的标签在 FEND 指令后编程，必须要有 SRET 指令。中断用指针也在 FEND 指令后编程，必须要有 IRET 指令。
- 在执行 CALL 指令后，SRET 指令执行前，如果执行了 FEND 指令；或者在 FOR 指令执行后，NEXT 指令执行前执行了 FEND 指令，则程序会出错。即不能在子程序中间或 FOR 和 NEXT 指令之间写 FEND 指令。

- 使用多个 FEND 指令的情况下，请在最后的 FEND 指令与 END 指令之间编写程序或中断子程序。

## 4-3-7. 监视定时器[WDT]

## 1、指令概述

WDT指令时执行监视用定时器刷新的指令。

监视定时器[WDT]			
指令形式	WDT		
执行条件	-	适用机型	CZA

## 2、操作数

操作数	作用	数据类型
无	-	-

## 3、适用软元件

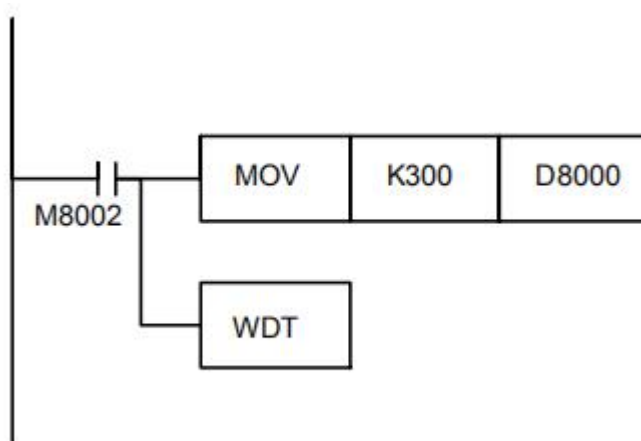
无
---

## 功能和动作

当可编程控制器的运行周期（0-END及FEND指令执行时间）超过200ms时，可编程控制器CPU的出错指示灯点亮同时停止工作，因此编程过程中插入使用该指令。

在CJ指令标签的步号比CJ指令小时，可在标签后编写WDT指令，或者在FOR~NEXT指令之间编程时使用该指令。

通过改写D8000（监视定时器时间）的内容，可改变监视定时器的检出时间。通过输入下述程序，在这之后的可编程控制器的程序将采用新的监视定时器时间执行监视，下述程序监视定时器时间为300ms，如果不编写监视定时器时间刷新指令（WDT），则在END处理时，D8000的值才有效。



## 4-4. 触点比较指令

助记符	指令功能	章节
LD=	开始 (S1) = (S2) 时导通	4-4-1
LD>	开始 (S1) > (S2) 时导通	4-4-1
LD<	开始 (S1) < (S2) 时导通	4-4-1
LD<>	开始 (S1) ≠ (S2) 时导通	4-4-1
LD≤	开始 (S1) ≤ (S2) 时导通	4-4-1
LD≥	开始 (S1) ≥ (S2) 时导通	4-4-1
AND=	串联 (S1) = (S2) 时导通	4-4-2
AND>	串联 (S1) > (S2) 时导通	4-4-2
AND<	串联 (S1) < (S2) 时导通	4-4-2
AND<>	串联 (S1) ≠ (S2) 时导通	4-4-2
AND≤	串联 (S1) ≤ (S2) 时导通	4-4-2
AND≥	串联 (S1) ≥ (S2) 时导通	4-4-2
OR=	并联 (S1) = (S2) 时导通	4-4-3
OR>	并联 (S1) > (S2) 时导通	4-4-3
OR<	并联 (S1) < (S2) 时导通	4-4-3
OR<>	并联 (S1) ≠ (S2) 时导通	4-4-3
OR≤	并联 (S1) ≤ (S2) 时导通	4-4-3
OR≥	并联 (S1) ≥ (S2) 时导通	4-4-3

## 4-4-1. 开始比较[LD□]

## 1、指令概述

LD□ 是连接母线的触点比较指令。

开始比较[LD□]			
16 位指令	下述	32 位指令	下述
执行条件	-	适用机型	CZA

## 2、操作数

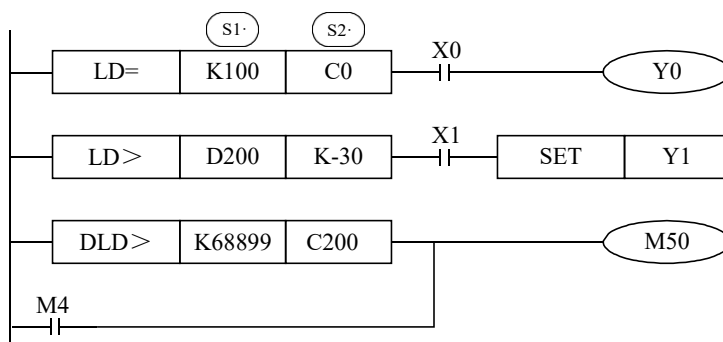
操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位, BIN
S2	指定比较数的数值或软元件地址编号	16/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S1		√	√	√	√	√	√	√	√	√
S2		√	√	√	√	√	√	√	√	√

## 功能和动作

16 位指令	32 位指令	导通条件	非导通条件
LD=	DLD=	(S1) = (S2)	(S1) ≠ (S2)
LD>	DLD>	(S1) > (S2)	(S1) ≤ (S2)
LD<	DLD<	(S1) < (S2)	(S1) ≥ (S2)
LD<>	DLD<>	(S1) ≠ (S2)	(S1) = (S2)
LD≤	DLD≤	(S1) ≤ (S2)	(S1) > (S2)
LD≥	DLD≥	(S1) ≥ (S2)	(S1) < (S2)



如上图，当计数器C0的当前值为100时且X0处于ON时驱动Y0。

当D200的内容为-29以上且X1处于ON时，置位Y1。

当计数器C200的内容小于68899或者M4处于ON时，驱动M50。

**注意事项**

- 当源数据的最高位（16 位指令：b15，32 位：b31）为 1 时，将该数值作为负数进行比较。
- 32 位计数器（C300~）的比较，必须以 32 位指令来进行。若指定 16 位指令时，会导致程序出错或运算错误。

## 4-4-2. 串联比较[AND□]

## 1、指令概述

AND□ 是与其它接点串联的比较指令。

串联比较[AND□]			
16 位指令	下述	32 位指令	下述
执行条件	常开/闭线圈触发	适用机型	CZA

## 2、操作数

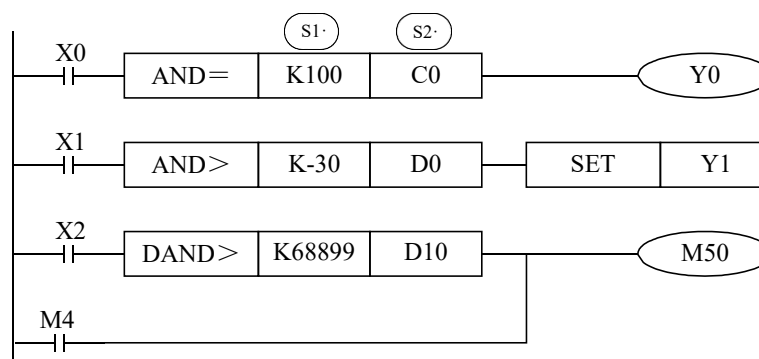
操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位, BIN
S2	指定比较数的数值或软元件地址编号	16/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S1		√	√	√	√	√	√	√	√	√
S2		√	√	√	√	√	√	√	√	√

## 功能和动作

16 位指令	32 位指令	导通条件	非导通条件
AND=	DAND=	(S1) = (S2)	(S1) ≠ (S2)
AND>	DAND>	(S1) > (S2)	(S1) ≤ (S2)
AND<	DAND<	(S1) < (S2)	(S1) ≥ (S2)
AND<>	DAND<>	(S1) ≠ (S2)	(S1) = (S2)
AND≤	DAND≤	(S1) ≤ (S2)	(S1) > (S2)
AND≥	DAND≥	(S1) ≥ (S2)	(S1) < (S2)



如上图，当X0处于ON，且计数器C0的当前值为100时，驱动Y0。

当X1处于ON，且数据寄存器D0的值小于-30时，置位Y1。

当X2处于ON，且数据寄存器D10的值小于68899时，或者M4处于ON时，驱动线圈M50。

**注意事项**

- 当源数据的最高位（16 位指令：b15，32 位：b31）为 1 时，将该数值作为负数进行比较。
- 32 位计数器（C200~）的比较，必须以 32 位指令来进行。若指定 16 位指令时，会导致程序出错或运算错误。



## 4-4-3. 并联比较[OR□]

## 1、指令概述

OR□ 是与其它接点并联的触点比较指令。

并联比较[OR□]			
16 位指令	下述	32 位指令	下述
执行条件	-	适用机型	CZA

## 2、操作数

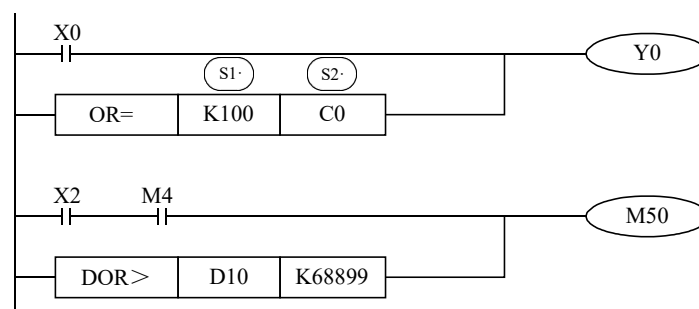
操作数	作用	类型
S1	指定被比较数的数值或软元件地址编号	16/32 位, BIN
S2	指定比较数的数值或软元件地址编号	16/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S1		√	√	√	√	√	√	√	√	√
S2		√	√	√	√	√	√	√	√	√

## 编程

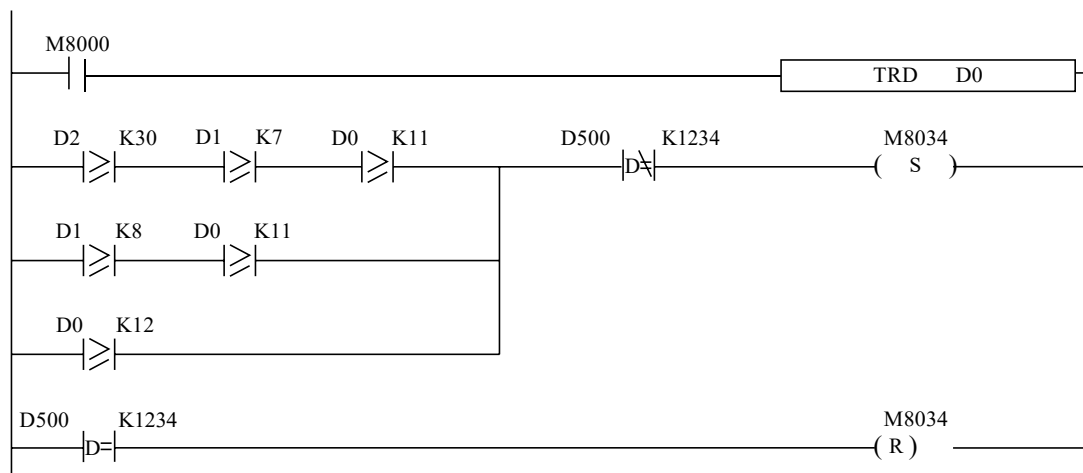
16 位指令	32 位指令	导通条件	非导通条件
OR=	DOR=	(S1) = (S2)	(S1) ≠ (S2)
OR>	DOR>	(S1) > (S2)	(S1) ≤ (S2)
OR<	DOR<	(S1) < (S2)	(S1) ≥ (S2)
OR<>	DOR<>	(S1) ≠ (S2)	(S1) = (S2)
OR≤	DOR≤	(S1) ≤ (S2)	(S1) > (S2)
OR≥	DOR≥	(S1) ≥ (S2)	(S1) < (S2)



## 注意事项

- 当源数据的最高位（16 位指令：b15，32 位：b31）为 1 时，将该数值作为负数进行比较。
- 32 位计数器（C200~）的比较，必须以 32 位指令来进行，不可指定 16 位指令形式。

**例 1:** 到达特定时间禁止输出。下例是当时间到 2011 年 7 月 30 日之后就禁止所有输出，1234 作为密码，用双字的 D500（D501）作为存放密码的寄存器，当密码正确时恢复所有输出。



LD M8000	//M8000 位常 ON 线圈
TRD D0	//读取时钟信息存入 D0~D6
LD >= D2 K30	//时钟日期大于或等于 30
AND >= D1 K7	//时钟月份大于或等于 7
AND >= D0 K11	//时钟年份大于或等于 11
LD >= D1 K8	//或者时钟月份大于或等于 8
AND >= D0 K11	//时钟年份大于或等于 11
ORB	//或者
OR >= D0 K12	//时钟年份大于或等于 12
DAND <> D500 K1234	//而且当密码不等于 K1234 时
SET M8034	//置位 M8034，所有输出禁止
DLD= D500 K1234	//当密码等于 K1234 时，则密码正确
RST M8034	//复位 M8034，恢复所有输出正常工作

**4-5. 数据传送指令**

指令助记符	指令功能	章节
CMP	数据比较	4-5-1
ZCP	数据区间比较	4-5-2
MOV	传送	4-5-3
BMOV	一并传送	4-5-4
SMOV	移位传送	4-5-5
FMOV	多点重复传送	4-5-6
ZRST	批次复位	4-5-7
SWAP	高低字节交换	4-5-8
XCH	两个数据交换	4-5-9

4-5-1. 数据比较[**CMP**]

## 1、指令概述

将指定的两个数据进行大小比较，并输出结果的指令。

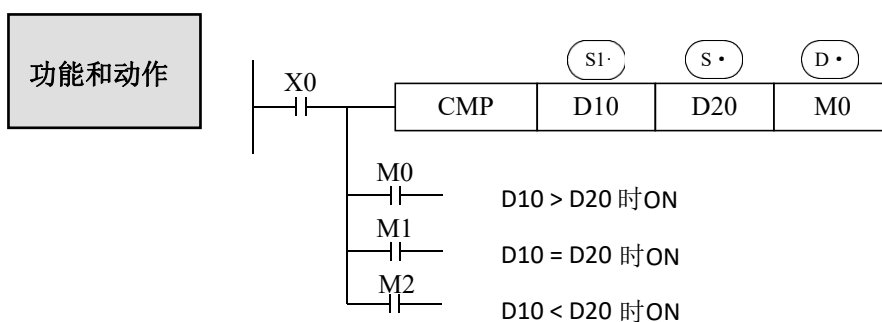
数据比较[ <b>CMP</b> ]			
16 位指令	<b>CMP</b>	32 位指令	<b>DCMP</b>
执行条件	常开/闭、边沿触发	适用机型	全系列

## 2、操作数

操作数	作用	类型
<b>S1</b>	指定被比较的数据或软元件地址编号	16 位, BIN
<b>S</b>	指定比较源的数据或软元件地址编号	16 位, BIN
<b>D</b>	指定输出比较结果的软元件地址编号	位

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	<b>S1</b>	√	√	√	√	√	√	√	√	√
	<b>S</b>	√	√	√	√	√	√	√	√	√
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	<b>D</b>		√	√	√					



即使使用  $\text{X0}=\text{OFF}$  停止执行 **CMP** 指令时， $\text{M0} \sim \text{M2}$  仍然保持  $\text{X0}$  变为  $\text{OFF}$  以前的状态。指令不执行时，如需清除比较结果可使用复位指令。

- 将数据  $\text{S1}$  与  $\text{S}$  相比较，根据大小一致输出以  $\text{D}$  起始的 3 点 ON/OFF 状态。
- $\text{D}$ ， $\text{D}+1$ ， $\text{D}+2$ ：根据比较结果位软元件 3 点 ON/OFF 输出。

## 4-5-2. 数据区间比较[ZCP]

## 1、指令概述

将 2 点指定数据与当前数据进行大小比较，并输出结果的指令。

数据区间比较[ZCP]			
16 位指令	ZCP	32 位指令	DZCP
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

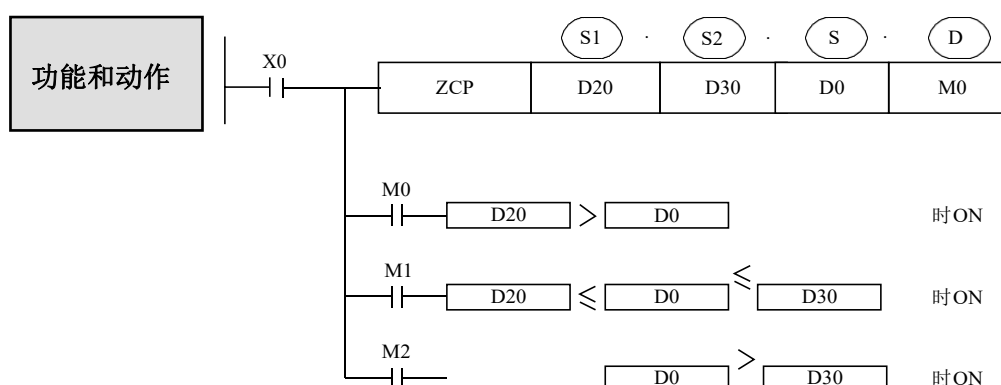
操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	16 位, BIN
S2	指定比较基准上限的数据或软元件地址编号	16 位, BIN
S	指定当前数据或软元件地址编号	16 位, BIN
D	指定比较结果的数据或软元件地址编号	位

## 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1	√	√	√	√	√	√	√	√	√	
	S2	√	√	√	√	√	√	√	√	√	
	S	√	√	√	√	√	√	√	√	√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
	D		√	√	√		



- 将 (S) 数据同上下两点的的数据比较范围相比较，(D) 根据区域大小输出起始的 3 点 ON/OFF 状态。
- (D), (D)+1, (D)+2 : 根据比较结果的区域位软元件 3 点 ON/OFF 输出。

## 4-5-3. 传送[MOV]

## 1、指令概述

使指定软元件的数据照原样传送到其他软元件中。

传送[MOV]			
16 位指令	MOV	32 位指令	DMOV
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

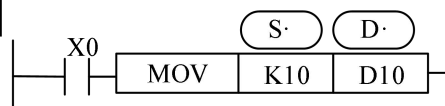
操作数	作用	类型
S	指定传送源的数据或保存数据的软元件编号	16 位/32 位, BIN
D	指定传送的目标软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S		√	√	√	√	√	√	√	√	√
D			√	√	√	√	√	√	√	

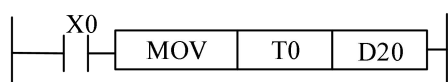
## 功能和动作

## 《16 位数据的传送》



- a.将源的内容向目标传送。
- b.X0 为 OFF 时，数据不变化。
- c.将常数K10 传送到D10。

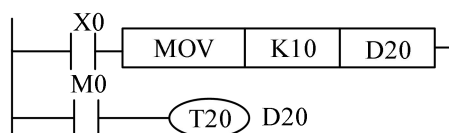
## 《定时器、计数器的当前值读出示例》



T0 当前值) → (D20)

关于计数器也一样。

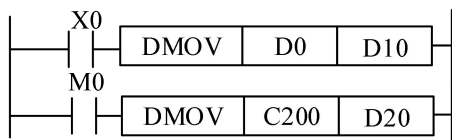
## 《定时器、计数器设定值的间接指定示例》



(K10) → (D20)

D20 = K10

## 《32 位数据的传送》



(D1, D0) → (D11, D10)

(C200 当前值) → (D21, D20)

运算结果以 32 位输出的应用指令 (MUL 等)、32 位数值或 32 位软元件的高速计数器当前值等数据的传送, 必须使用 DMOV 指令。

## 4-5-4. 成批传送[BMOV]

## 1、指令概述

使指定软元件的数据照原样传送到其他软元件中。

数据块传送[BMOV]			
16 位指令	BMOV(P)	32 位指令	-
执行条件	常开/闭线圈、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	指定传送源的数据或保存数据的软元件编号	16 位, BIN; 位
D	指定传送的目标软元件地址编号	16 位, BIN; 位
n	指定传送点数的数值	16 位, BIN

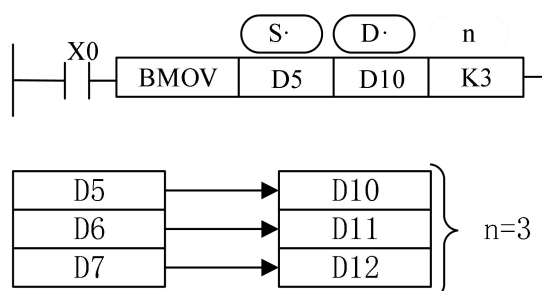
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S		√	√	√	√	√	√	√		
D			√	√	√	√	√	√		
n								√		√

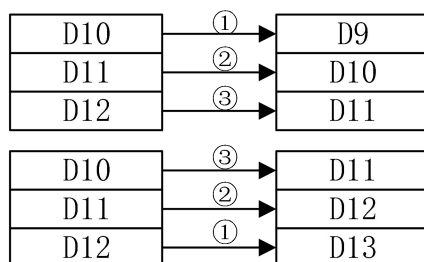
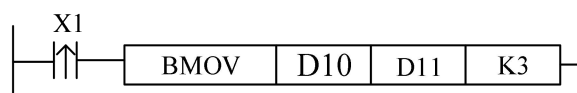
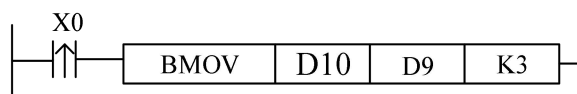
## 功能和动作

- 将以源指定的软元件为开头的 n 点数据向以目标指定的软元件为开头的 n 点软元件以数据块的形式传送。(在超过软元件编号范围时, 在可能的范围内传送)。

《字软元件》



- 如下图传送编号范围有重叠时，为了防止输送源数据没传送就改写，根据编号重叠的方法，该指令会按①~③的顺序进行自动传送。



注：BMOV指令只会执行一次。



## 4-5-5. 位移动[SMOV]

## 1、指令概述

进行数据分配，合成的指令。

位移动[SMOV]			
16 位指令	SMOV	32 位指令	-
执行条件	常开/闭线圈、边沿触发	适用机型	CZA

## 2、操作数

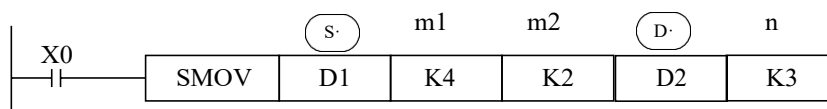
操作数	作用	类型
S	指定传送源的数据或保存数据的软元件编号	16 位, BIN; 位
D	指定传送的目标软元件地址编号	16 位, BIN; 位
n	指定传送点数的数值	16 位, BIN

## 3、适用软元件

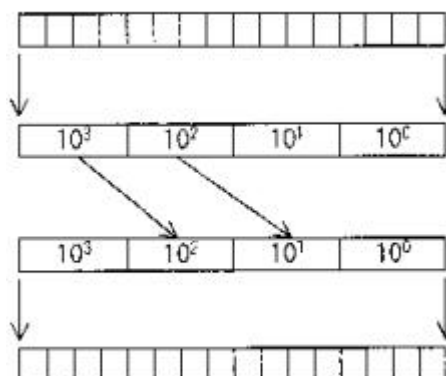
字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√	√	√	√	√	
	D		√	√	√	√	√	√	√	
	m1									√
	m2									√
	n									√

## 功能和动作

- m1,m2,n=1~4。
- D2的 $10^3$ 位及 $10^0$ 位在从D1传送时不受任何影响。将源数据（BIN）的BCD转换值从其第4位（m1=4）起的低2位部分（m2=2）向目标的第3位（n=3）开始传送，然后将其转换回BIN码。



D1(二进制16位)自动转换→D1(BCD 4 位)→位移动  
D2(BCD 4 位)自动转换→D2(二进制16位)  
BCD的值若超越 $0\sim 9999$ 范围则会出错。



例：当D1的值是251，D2的值为0时，执行上述指令的结果D2的值为20。

注解：D2的第4位及第一位在从D1传送时不受任何影响。将源数据（BIN）的BCD转换值从其第4位起的低2位部分向目标的第三位开始传送，然后将其转换成BIN码。

D1的值251转换成BCD码后表示为0251，而D2的值0转换成BCD码后表示为0000，执行指令后，D1的第4位起的低2位为02，送给D2的第三位起的低2位，D2值的BCD就变为0020，其值则为20。

同理：当D1的值是1251.D2的值为21时，执行上述指令的结果是D2的值为121。

## 4-5-6. 多点重复传送[FMOV]

## 1、指令概述

使指定软元件的数据照原样传送到其他软元件中。

多点传送[FMOV]			
16 位指令	FMOV	32 位指令	DFMOV
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

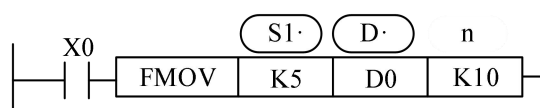
操作数	作用	类型
S	指定传送源的数据或保存数据的软元件编号	16 位/32 位, BIN
D	指定传送的目标软元件起始地址编号	16 位/32 位, BIN
n	指定传送点数的数值	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S		√	√	√	√	√	√	√	√	
D			√	√	√	√	√	√	√	
n										√

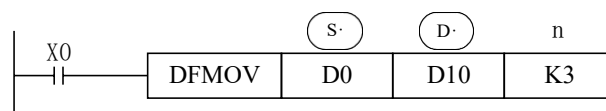
## 功能和动作

## 《16 位指令》



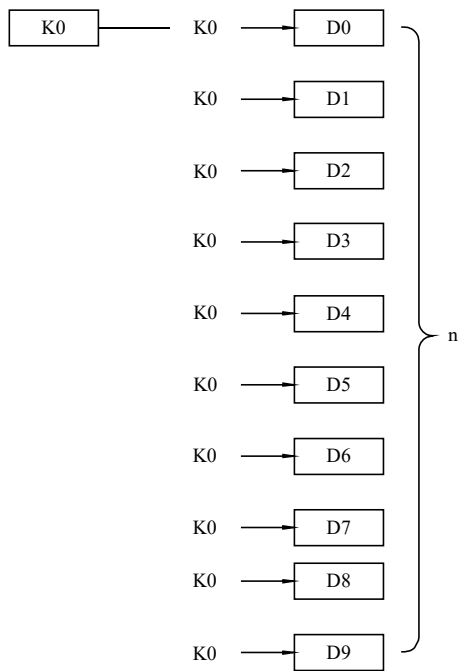
- $n \leq 12$ 。
- 将 K5 传送至 D0~D9，同一数据的多点传送指令。
- 将源指定的软元件的内容向以目标指定的软元件为开头的 n 点软元件进行传送，n 点软元件的内容都一样。
- 超过目标软元件号的范围时，向可能的范围传送。

## 《32 位指令》

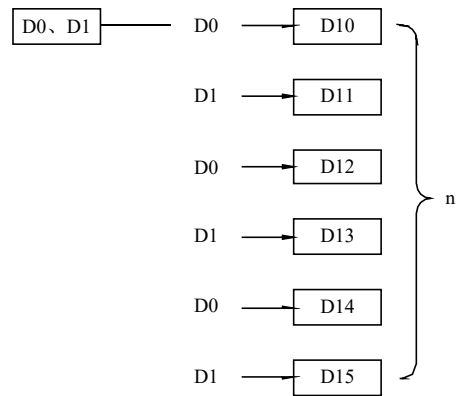


- 将 D0、D1 中的内容传送到 D10、D11；D12、D13；D14、D15。

## 《16 位数据传送》



## 《32 位数据传送》



### 4-5-7. 批次复位[ZRST]

#### 1、指令概述

将指定范围的位或字软元件进行复位或清零操作。

批次复位[ZRST]			
16 位指令	ZRST	32 位指令	-
执行条件	常开/闭线圈、边沿触发	适用机型	CZA

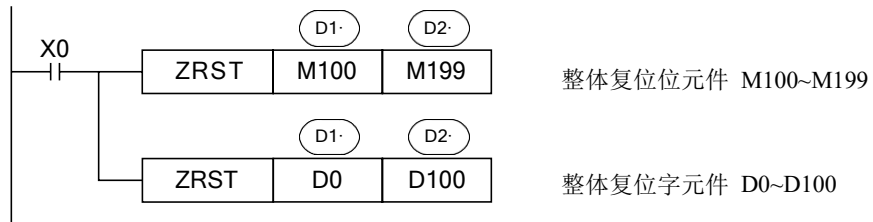
#### 2、操作数

操作数	作用	类型
D1	指定批次复位的起始软元件地址编号	位；16 位，BIN
D2	指定批次复位的结束软元件地址编号	位；16 位，BIN

#### 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	D1				√	√	√	√			
	D2					√	√	√	√		
位软元件	操作数	系统									
		X	Y	M	S	T	C				
	D1		√	√	√						
	D2		√	√	√						

#### 功能和动作



- (D1) (D2) 指定为同一种类的软元件，且 (D1) 编号 < (D2) 编号。
- 当 (D1) 编号 > (D2) 编号时，仅复位 (D1) 中指定的软元件。

#### 其他复位指令

- 作为软元件的单独复位指令，对于位元件 Y, M, S, T, C 和字元件 TD, CD, D, 可使用 RST 指令。
- 作为常数 K0 的成批写入指令 FMOV 指令，可以把 0 写入 DX, DY, DM, DS, T(TD), C(CD), D 的软元件中。

## 4-5-8. 高低字节交换[SWAP]

## 1、指令概述

将指定寄存器的高 8 位字节和低 8 位字节进行交换。

高低字节交换[SWAP]			
16 位指令	SWAP(P)	32 位指令	DSWAP(P)
执行条件	常开/闭、边沿触发	适用机型	全系列

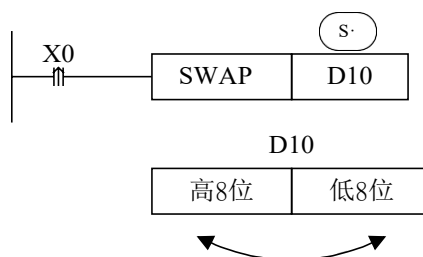
## 2、操作数

操作数	作用	类型
S	指定高低字节交换的软元件地址编号	16 位; BIN

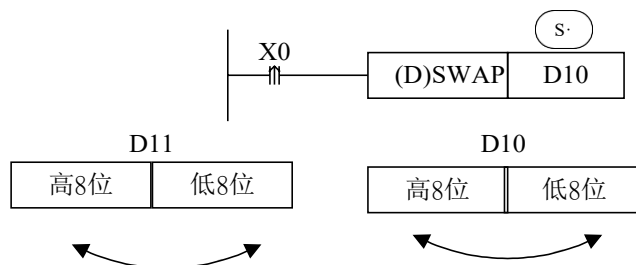
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S			√	√	√	√	√	√	√	

## 功能和动作



- SWAP指令的功能是将一个 16 位寄存器的低 8 位与高 8 位交换。
- 上例中如果将条件 X0 改为常开或常闭线圈触发，当输入 X0 为 ON 时，每个扫描周期都执行一次该指令，所以建议用上升沿或下降沿触发。



- DSWAP指令的功能是将一个 32 位的寄存器高 16 位的低 8 位与高 8 位以及低 16 位的低 8 位与高 8 位同时进行交换。



**4-6. 数据运算指令**

指令助记符	指令功能	章节
ADD	加法	4-6-1
SUB	减法	4-6-2
MUL	乘法	4-6-3
DIV	除法	4-6-4
INC	加 1	4-6-5
DEC	减 1	4-6-5
MEAN	求平均值	4-6-6
WAND	逻辑与	4-6-7
WOR	逻辑或	4-6-7
WXOR	逻辑异或	4-6-7
CML	取反	4-6-8
NEG	求补	4-6-9



## 4-6-1. 加法运算[ADD]

## 1、指令概述

将两个数据进行二进制加法运算，并对结果进行存储的指令。

加法运算[ADD]			
16 位指令	ADD (P)	32 位指令	DADD(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

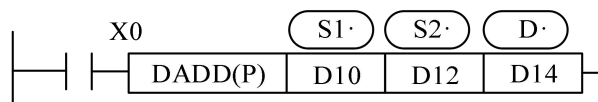
操作数	作用	类型
三个操作数时		
S1	指定进行加法运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行加法运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存加法结果的软元件地址编号	16 位/32 位, BIN
两个操作数时		
D	指定被加数及保存加法结果的软元件地址编号	16 位/32 位, BIN
S1	指定加数的数据或软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
三个操作数时										
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√	√	
两个操作数时										
	D		√	√	√	√	√	√	√	
	S1	√	√	√	√	√	√	√	√	√

## 功能和动作

## 《三个操作数时》



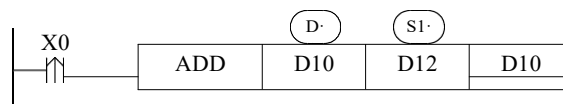
16位 (D10) + (D12) → (D14)

32位 (D10、D11) + (D12、D13) → (D14、D15)

- 两个源数据进行二进制加法后传递到目标处。各数据的最高位是正(0)、负(1)符号位，这些数据以代数形式进行加法运算(5+(-8)=-3)。
- 运算结果为0时，0标志会动作。如运算结果超过32,767(16位运算)或2,147,483,647(32位运算)时，进位标志会动作(参照下一页)。如运算结果超过-32,768(16位运算)或-2,147,483,648(32位运算)时，借位标志会动作(参照下一页)。

- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 可以将源操作数和目标操作数指定为相同的软元件编号。上例中驱动输入 X0 为 ON 时，每个扫描周期的都执行一次加法运算，请务必注意。

## 《两个操作数时》



16位 (D10) + (D12) → (D10)

32位 (D10、D11) + (D12、D13) → (D10、D11)

- 两个源数据进行二进制加法后传递到被加数地址处。各数据的最高位是正(0)、负(1)符号位，这些数据以代数形式进行加法运算(5+(-8)=-3)。
- 运算结果为 0 时，0 标志会动作。如运算结果超过 32,767 (16 位运算) 或 2,147,483,647 (32 位运算) 时，进位标志会动作(参照“相关软元件”)。如运算结果超过-32,768 (16 位运算) 或-2,147,483,648 (32 位运算) 时，借位标志会动作(参照“相关软元件”)。
- 进行 32 位运算时，字软元件的低 16 位侧的软元件被指定，紧接着上述软元件编号后的软元件将作为高位，为了防止编号重复，建议将软元件指定为偶数编号。
- 上例中如果 X0 为常开，当输入 X0 为 ON 时，每个扫描周期都执行一次该指令，所以建议用上升沿或下降沿触发，或者使用(D)ADDP指令。

## 标志位的动作及数值涵义

软元件	名称	作用
M8020	零	ON: 运算结果为 0 时。 OFF: 运算结果为 0 以外时。
M8021	借位	ON: 运算结果超出-32,768 (16 位运算) 或是-2,147,483,648 (32 位运算) 时，借位标志位动作。 OFF: 运算结果不到-32,768 (16 位运算) 或是-2,147,483,648 (32 位运算) 时。
M8022	进位	ON: 运算结果超出 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时，进位标志位动作。 OFF: 运算结果不到 32,767 (16 位运算) 或是 2,147,483,647 (32 位运算) 时。

## 4-6-2. 减法运算[SUB]

## 1、指令概述

将两个数据进行二进制减法运算，并对结果进行存储。

减法运算[SUB]			
16 位指令	SUB(P)	32 位指令	DSUB(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

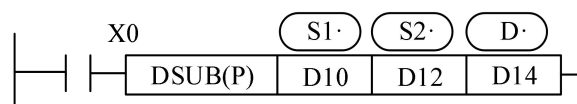
操作数	作用	类型
三个操作数时		
S1	指定进行减法运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行减法运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存减法结果的软元件地址编号	16 位/32 位, BIN
两个操作数时		
D	指定被减数及保存减法结果的软元件地址编号	16 位/32 位, BIN
S1	指定减数的数据或软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
三个操作数时											
	S1	√	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√	√		
两个操作数时											
	D		√	√	√	√	√	√	√		
	S1	√	√	√	√	√	√	√	√	√	√

## 功能和动作

## 《三个操作数时》

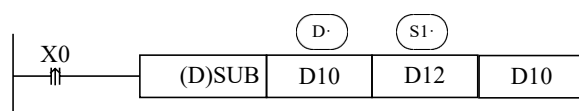


16位 (D10) - (D12) → (D14)

32位 (D10、D11) - (D12、D13) → (D14、D15)

- (S1) 指定的软元件的内容，以代数形式减去 (S2) 指定的软元件的内容，其结果被存入由 (D) 指定的软元件中 (5-(-8)=13)。
- 各种标志的动作、32 位运算软元件的指定方法等，均与上页的 ADD 指令相同。
- 要注意的是，上例中驱动输入 X0 为 ON 时，每个扫描周期都执行一次减法运算。
- 标志位的动作及作用参阅 4-6-1 相关内容。

## 《两个操作数时》

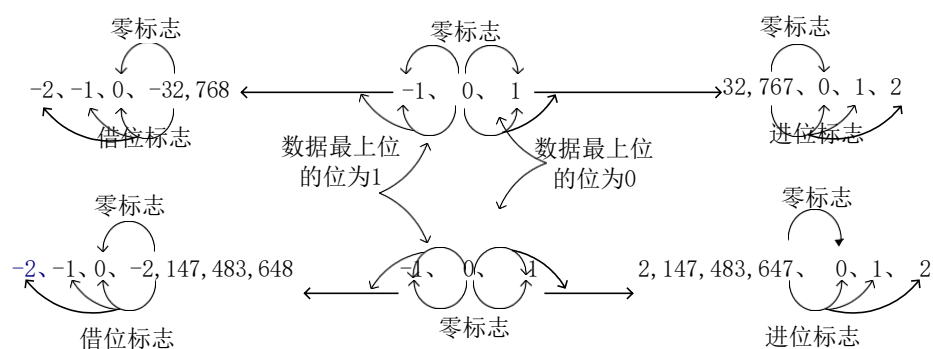


16位 (D) — (D12) → (D10)

32位 (D10、D11) — (D12、D13) → (D10、D11)

- (D) 指定的软元件的内容，以代数形式减去 (S1) 指定的软元件的内容，其结果被存入由 (D) 指定的软元件中 (5-(-8)=13)。
- 各种标志的动作、32位运算软元件的指定方法等，均与上页的 ADD 指令相同。
- 上例中如果 X0 为常开，当输入 X0 为 ON 时，每个扫描周期都执行一次该指令，所以建议用上升沿或下降沿触发，或者使用 (D) SUBP 指令。
- 标志位的动作及作用参阅 4-6-1 相关内容。

标志的动作与数值的正负关系如下所示：



## 4-6-3. 乘法运算[MUL]

## 1、指令概述

将两个数据进行二进制乘法运算，并对结果进行存储。

乘法运算[MUL]			
16 位指令	MUL(P)	32 位指令	DMUL(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

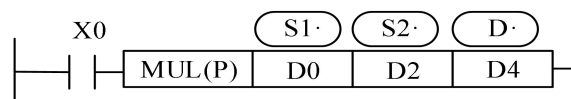
操作数	作用	类型
S1	指定进行乘法运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行乘法运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存乘法结果的软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√	√	

## 功能和动作

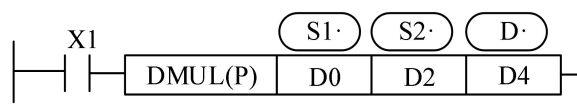
## 《16 位运算》



BIN      BIN      BIN  
 (D0) × (D2) → (D5, D4)  
 16 位      16 位 → 32 位

- 各源指定的软元件内容的乘积，以 32 位数据形式存入目标地址指定的软元件（低位）和紧接其后的软元件（高位）中。上图示例：(D0)=8、(D2)=9 时，(D5, D4)=72。
- 结果的最高位是正（0）、负（1）符号位。
- 要注意的是，上例中驱动输入 X0 为 ON 时，每个扫描周期都执行一次乘法运算。

## 《32 位运算》



BIN      BIN      BIN  
 (D1, D0) × (D3, D2) → (D7, D6, D5, D4)  
 32 位      32 位 → 64 位

- 在 32 位运算中，目标地址使用位软元件时，只能得到低 32 位的结果，不能得到高 32 位的结果。请向字元件传送一次后再进行运算。
- 在使用字元件时，也不能直接监视到 64 位数据的运算结果。

- 这种情况下建议最好使用浮点运算。
- 不能指定Z作为D-
- V、Z只限于于16位计算时，可指定。

## 4-6-4. 除法运算[DIV]

## 1、指令概述

将两个数据进行二进制除法运算，并对结果进行存储。

除法运算[DIV]			
16 位指令	DIV(P)	32 位指令	DDIV(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

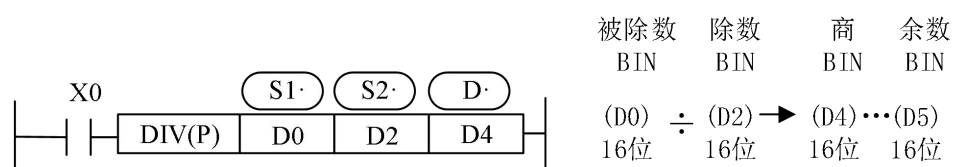
操作数	作用	类型
S1	指定进行除法运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行除法运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存除法结果的软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S1		√	√	√	√	√	√	√	√	√
S2		√	√	√	√	√	√	√	√	√
D			√	√	√	√	√	√	√	

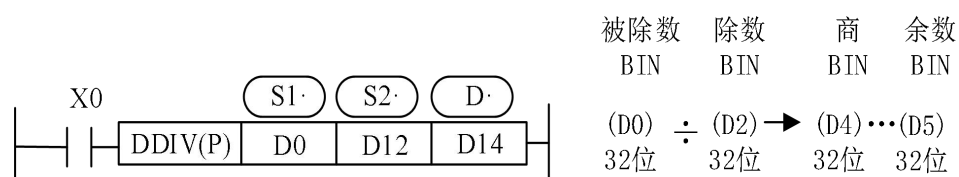
## 功能和动作

## 《16 位运算》



- (S1) 指定软元件的内容是被除数，(S2) 指定软元件的内容是除数，(D) 指定的软元件和其下一个编号的软元件将存入商和余数。
- 要注意的是，上例中驱动输入 X0 为 ON 时，每个扫描周期都执行一次除法运算，如果是 DIVP 指令就只会执行一次。

## 《32 位运算》



- 除数内容是由 (S1) 指定软元件和其下一个编号的软元件组合而成，除数内容是由 (S2) 指定的软元件和其下一个编号的软元件组合而成，其商和余数如上图所示，存入与 (D) 指定软元件相连接的 4 点软元件。
- 不能指定 Z 作为 D-，将位元件指定为 D- 时，无法得到余数。
- 除数为 0 时发生运算错误，不能执行指令。
- 商和余数的最高位为正 (0)、负 (1) 的符号位。当被除数或除数中的一方为负数时，商则为负，当被除数为负时余数则为负。
- V、Z 只限于于 16 位计算时，可指定。



## 4-6-5. 自加 1[INC]、自减 1[DEC]

## 1、指令概述

将指定软元件中的数据进行加 1/减 1 运算。

自加 1[INC]			
16 位指令	INC (P)	32 位指令	DINC(P)
执行条件	常开/闭、边沿触发	适用机型	CZA
自减 1[DEC]			
16 位指令	DEC(P)	32 位指令	DDEC(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

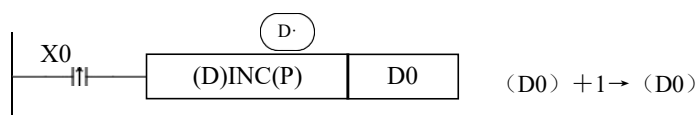
操作数	作用	类型
D	指定进行自加 1/减 1 运算的软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	D		√	√	√	√	√	√	√

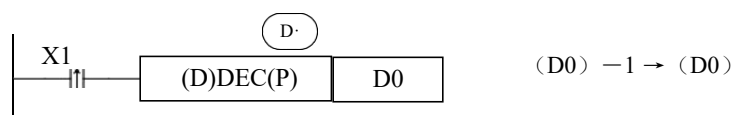
## 功能和动作

## 《自加 1 运算》



- X000 每置 ON 一次，(D) 指定的软元件的内容就加 1。
- 16 位运算时，如果+32,767 加 1 则变为-32,768，标志位动作；32 位运算时，如果+2,147,483,647 加 1 则变为-2,147,483,648，标志位动作。

## 《自减 1 运算》



- X001 每置 ON 一次，(D) 指定的软元件的内容就减 1。
- -32,768 或-2,147,483,648 减 1，则为+32,767 或+2,147,483,647，标志位动作。

## 4-6-6. 求平均值[MEAN]

## 1、指令概述

将指定数据或软元件进行求平均值运算。

求平均值[MEAN]			
16 位指令	MEAN (P)	32 位指令	DMEAN(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

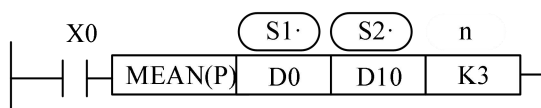
## 2、操作数

操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位, BIN
D	指定存储平均值结果的软元件地址编号	16 位, BIN
n	指定源数据个数的数值	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√	√	√	√		
	D		√	√	√	√	√	√	√	
	n									√

功能和动作



$$\frac{(D0) + (D1) + (D2)}{3} \longrightarrow (D10)$$

- 将 n 点的源数据的平均值（代数和被 n 除）存入目标地址中，余数舍去。
- 取 n 值时要注意，范围不要超过可用软元件编号，否则会发生运算错误。
- N=1~64。

## 4-6-7. 逻辑与[WAND]、逻辑或[WOR]、逻辑异或[WXOR]

## 1、指令概述

将指定数据或软元件的各位进行逻辑与/逻辑或/逻辑异或运算。

逻辑与[WAND]			
16 位指令	WAND	32 位指令	DWAND
执行条件	常开/闭、边沿触发	适用机型	CZA
逻辑或[WOR]			
16 位指令	WOR	32 位指令	DWOR
执行条件	常开/闭、边沿触发	适用机型	CZA
逻辑异或[WXOR]			
16 位指令	WXOR	32 位指令	DWXOR
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

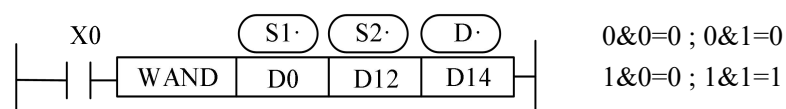
操作数	作用	类型
S1	指定进行运算的数据或软元件地址编号	16 位/32 位, BIN
S2	指定进行运算的数据或软元件地址编号	16 位/32 位, BIN
D	指定保存运算结果的软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√	√	

## 功能和动作

## 《逻辑与运算》



$0 \& 0 = 0$ ;  $0 \& 1 = 0$   
 $1 \& 0 = 0$ ;  $1 \& 1 = 1$

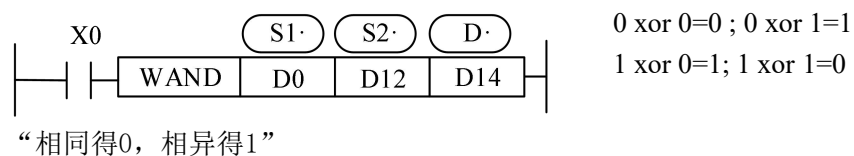
“有0得0，全1得1”



$0 \text{ or } 0 = 0$ ;  $0 \text{ or } 1 = 1$   
 $1 \text{ or } 0 = 1$ ;  $1 \text{ or } 1 = 1$

“有1得1，全0得0”

## 《逻辑异或运算》

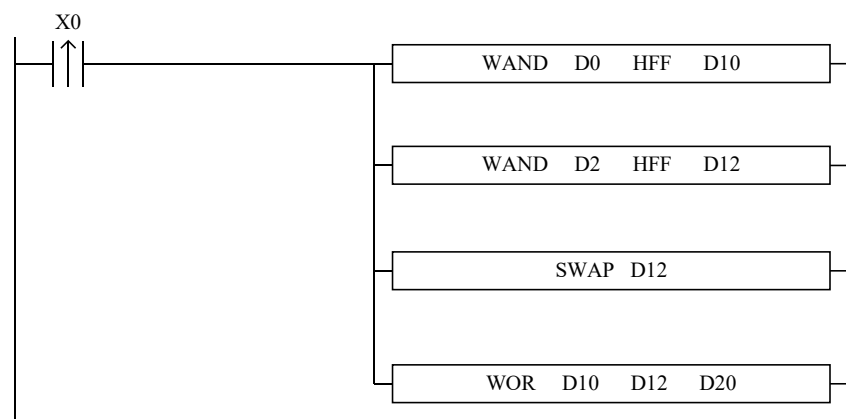


## 例 1:

H3FFF是十六进制数3FFF，用二进制数表示就是0011 1111 1111 1111。K4Y000表示是4个为元件组，起始位置为Y0，即Y0~Y7, Y11~Y17共16位。如下表示将十六进制常数3FFF同寄存器D110内的数值进行按位进行“与”操作，产生的结果放入起始位置为Y000的4个为元件组输出。



## 例 2: 将D0 的低 8 位和 D2 的低 8 位结合组成一个字。



```
LDP X0
```

```
WAND D0 HFF D10
```

```
WAND D2 HFF D12
```

```
SWAP D12
```

```
WOR D10 D12 D20
```

//输入 X0 的上升沿

//逻辑与，取 D0 的低 8 位数据，存放于 D10

//逻辑与，取 D2 的低 8 位数据，存放于 D12

//D12 的高 8 位和低 8 位数据交换

//D10 的低 8 位和 D12 的高 8 位组合成 16 位数据，存放于 D20。

**4-6-8. 逻辑取反[CML]**

1、指令概述

将指定数据或软元件中的数据进行反相传送的指令。

取反[CML]			
16 位指令	CML	32 位指令	DCML
执行条件	常开/闭、边沿触发	适用机型	CZA
硬件要求	-	软件要求	-

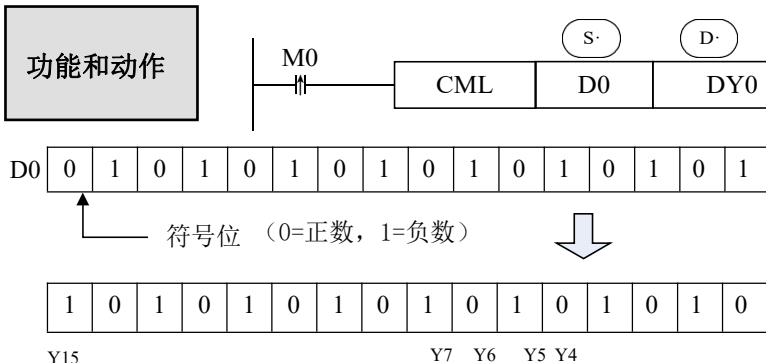
2、操作数

操作数	作用	类型
S	指定源数据值或软元件地址编号	16 位/32 位, BIN
D	指定保存结果的软元件地址编号	16 位/32 位, BIN

3、适用软元件

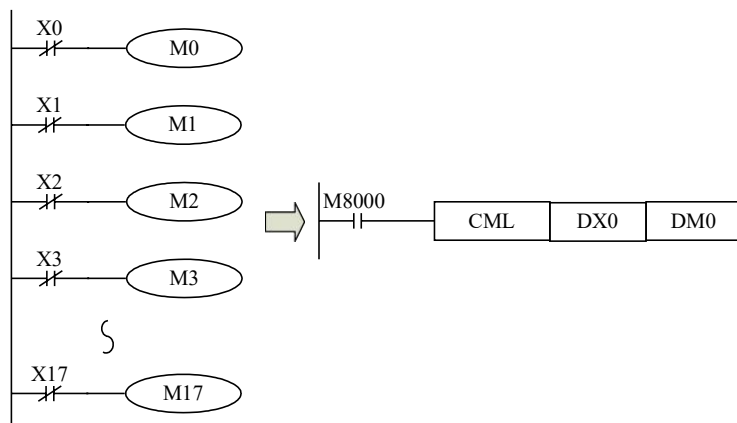
字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√	√	√	√	√	√
D		√	√	√	√	√	√	√		

**功能和动作**



- 将源数据的各位反相（1→0，0→1）后，传送到目标地址。在源数据中使用常数 K 的话，能自动地转换成二进制。
- 该指令适用于需要可编程控制器以逻辑反相输出的场合。

《反相输入的读取》



- 左边的顺控程序可以用下面的 CML 指令表示。

## 4-6-9. 求补[NEG]

## 1、指令概述

将指定软元件中的数据进行求负运算。

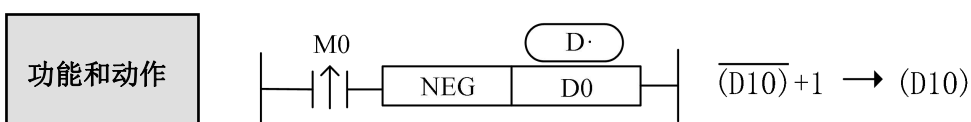
求负[NEG]			
16 位指令	NEG (P)	32 位指令	DNEG(P)
执行条件	边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
D	指定源数据的软元件地址编号	16 位/32 位, BIN

## 3、适用软元件

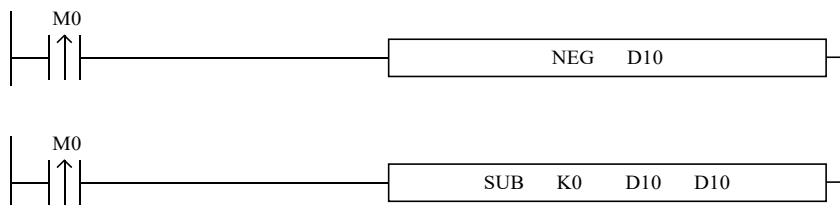
字软 元件	操作 数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	D		√	√	√	√	√	√	√



- 将  $\textcircled{D}$  指定软元件的内容中各位先取反 (1→0, 0→1), 然后再加 1, 将其结果再存入原先的软元件中。

上述动作, 假设 D10 起始数据为 20, M0 一次上升沿后, D10 的值转变为-20; 当 M0 再一次上升沿后, D10 的值变为+20。

下面的两条语句, 执行的效果是一样的。



**4-7. 数据移位指令**

指令助记符	指令功能	章节
ROL	左回转	4-7-1
ROR	右回转	4-7-1
RCL	带进位左回转	4-7-2
RCR	带进位右回转	4-7-2
SFWR	移位写入	4-7-3
SFRD	移位读出	4-7-3
SFTL	位左移	4-7-4
SFTR	位右移	4-7-5
WSFL	字左移	4-7-6
WSFR	字右移	4-7-7

## 4-7-1. 左回转[ROL]、右回转[ROR]

## 1、指令概述

使 16 位或 32 位数据的各位信息循环左移/循环右移的指令。

左回转[ROL]			
16 位指令	ROL (P)	32 位指令	DROL(P)
执行条件	常开/闭、边沿触发	适用机型	CZA
循环右移[ROR]			
16 位指令	ROR(P)	32 位指令	DROR(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

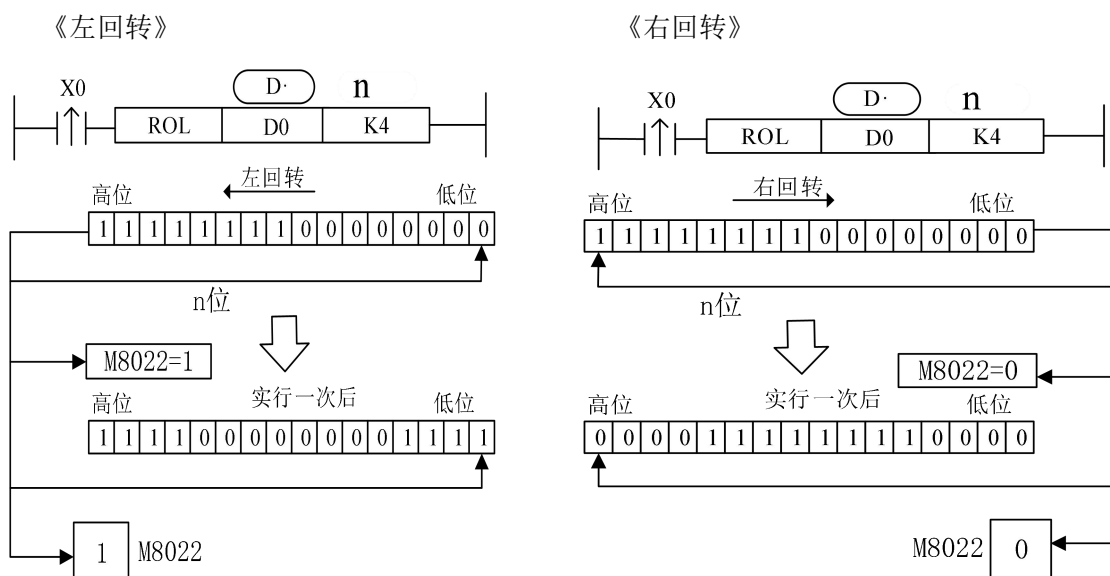
操作数	作用	类型
D	指定源数据的软元件地址编号	16 位/32 位, BIN
n	指定循环左移的位数	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
D			√	√	√	√	√	√	√	
n										√

## 功能和动作

- 回转量:  $n \leq 16$  (16 位指令),  $n \leq 32$  (32 位指令)。
- 每一次 X0 从 OFF→ON 变化一次时, 则进行 n 位循环左移或右移, 最终位被存入进位标志中。





**4-7-2. 带进位左回转[RCL]、带进位右回转[RCR]**

1、指令概述

将16位或32位数据的各位信息左右回转的指令。

带进位左回转[RCL]			
16 位指令	RCL(P)	32 位指令	DRCL(P)
执行条件	常开/闭、边沿触发	适用机型	CZA
带进位右回转[RCR]			
16 位指令	RCR(P)	32 位指令	DRCR(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

2、操作数

操作数	作用	类型
D	指定源数据的软元件地址编号	16 位/32 位, BIN
n	指定逻辑左移/逻辑右移的位数	16 位/32 位, BIN

3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
D			√	√	√	√	√	√	√	
n										√

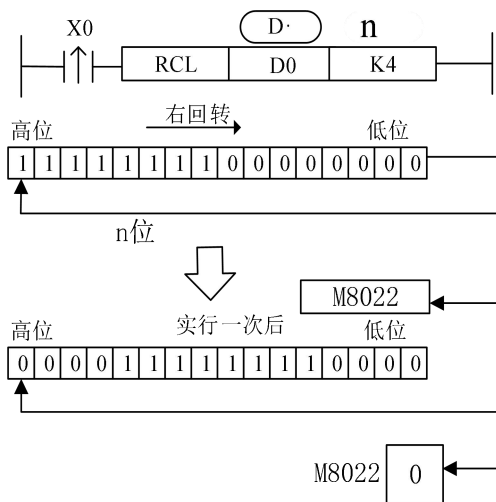
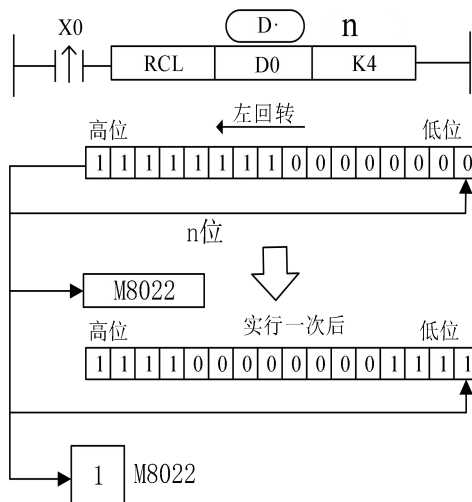
**功能和动作**

- 回转量：n≤16（16位指令），n≤32（32位指令）。
- 执行RCL 指令一次之后，就执行n位左回转。
- RCL 指令的意义和使用与ROL 相同。
- 执行RCR 指令一次之后，就执行n位右回转。

● RCR 与ROR 有所区别，前者在移位时，进位标志M8022的状态也参与移位；而后者在移位时，在左移进位时M8022会置1，右移时低位变高位时会将M8022置0。

《带进位左移》

《带进位右移》



## 4-7-3. 移位写入[SFWR]、移位读出[SFRD]

## 1、指令概述

为控制先进先出的数据写入指令和读出指令。

移位写入[SFWR]			
16 位指令	SFWR (P)	32 位指令	
执行条件	常开/闭、边沿触发	适用机型	CZA
移位读出[SFRD]			
16 位指令	SFRD (P)	32 位指令	
执行条件	常开/闭、边沿触发	适用机型	CZA

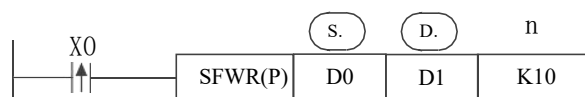
## 2、操作数

操作数	作用	类型
D	指定源数据的软元件地址编号	16 位, BIN
n	指定循环左移的位数	16 位, BIN

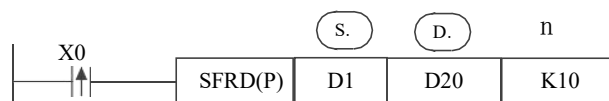
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√	√	√	√	√	√
	D		√	√	√	√	√	√		
n									√	

## 功能和动作



- n的范围： $0 \leq n \leq 512$ 。
- SFWR指令为控制先进先出的数据写入指令。
- 当X0从OFF→ON变化时，D0的内容被存入D2,(D1)的内容为1.当D0的内容变更以后，X0再次从OFF→ON时，这个D0的内容被存入D3,(D1)的内容变为2。（连续执行型指定中，每个扫描周期都执行依次保存）
- 和以下一样由右端开始在指针D1的内容中表示数据的存入点数，D1的内容如超过n-1,则不处理进位标志的M8022会工作。



- SFRD指令为控制先进先出法的数据读出指令。
- 当X001从OFF→ON变化时，D2的内容被传送到D20,与此同时，指针D1的内容减少，左侧的数据逐字向右侧移动。（连续执行型指令将在每个扫描周期执行移位）
- 数据的读出通常从D2开始。当指针的内容为0时，若X0再次从OFF→ON时，则不处理，同时零点标志M8020动作。本读出不能改变D10的内容。

## 4-7-4. 位左移[SFTL]

## 1、指令概述

将指定软元件中的数据进行位左移的指令。

位左移[SFTL]			
16 位指令	SFTL(P)	32 位指令	-
执行条件	边沿触发	适用机型	CZA

## 2、操作数

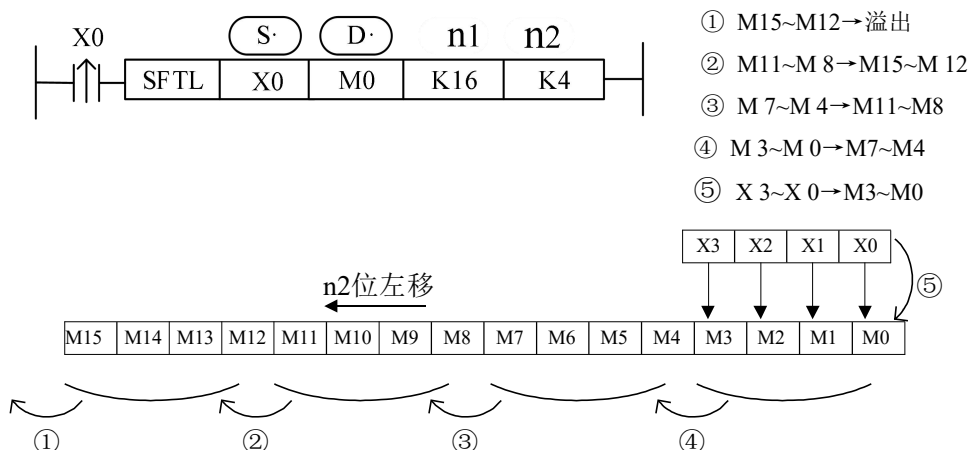
操作数	作用	类型
S	指定源数据的软元件首地址编号	位
D	指定目标软元件的首地址编号	位
n1	指定目标元件的个数（不超过 1024）	16 位/32 位, BIN
n2	指定位左移每次移动的位数（不超过 1024）	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	n1									√
	n2									√
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	S	√	√	√	√					
	D		√	√	√					

## 功能和动作

- 对于 n1 位（移动软元件的长度）的位元件进行 n2 的左移动的指令。（指令执行时执行 n2 位的移位）。
- 执行脉冲执行指令时，驱动输入 X0 由 OFF→ON 变化时，执行 n2 位移位。
- n2 为 K1 时，每执行一次移位指令，目标软元件左移 1 位。
- 位移指令是对[D·]所指定的n1个位元件连同[S·]所指定的n2个位元件的数据右移或左移n2位。



## 4-7-5. 位右移[SFTR]

## 1、指令概述

将指定软元件中的数据进行位右移的指令。

位左移[SFTR]			
16 位指令	SFTR(P)	32 位指令	-
执行条件	边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	指定源数据的软元件首地址编号	位
D	指定目标软元件的首地址编号	位
n1	指定目标元件的个数（不超过 1024）	16 位/32 位, BIN
n2	指定位右移每次移动的位数（不超过 1024）	16 位/32 位, BIN

## 3、适用软元件

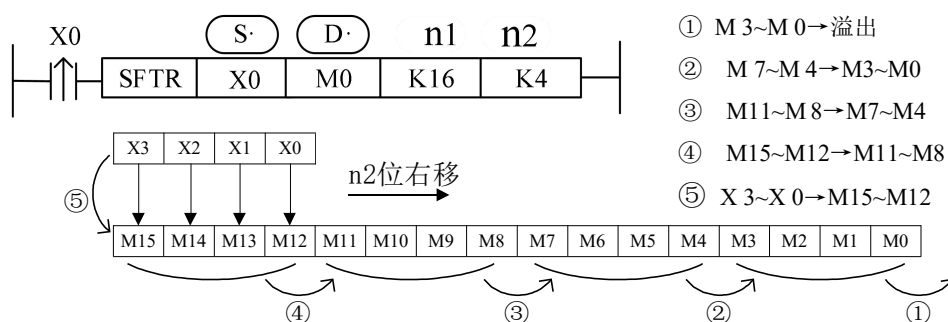
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	n1										√
	n2										√

位软元件	操作数	系统					
		X	Y	M	S	T	C
	S	√	√	√	√		
	D		√	√	√		

## 功能和动作

- 对于 n1 位（移动寄存器的长度）的位元件进行 n2 的右移动的指令。（指令执行时执行 n2 位的移位）。
- 执行脉冲指令时，驱动输入 X0 由 OFF→ON 变化时，执行 n2 位移位。
- n2 为 K1 时，每执行一次移位指令，目标软元件右移 1 位。
- 位移位指令是对[D·]所指定的n1个位元件连同[S·]所指定的n2个位元件的数据右移或左移n2位。



## 4-7-6. 字左移[WSFL]

## 1、指令概述

将指定软元件中的数据进行字左移的指令。

字左移[WSFL]			
16 位指令	WSFL (P)	32 位指令	-
执行条件	边沿触发	适用机型	CZA

## 2、操作数

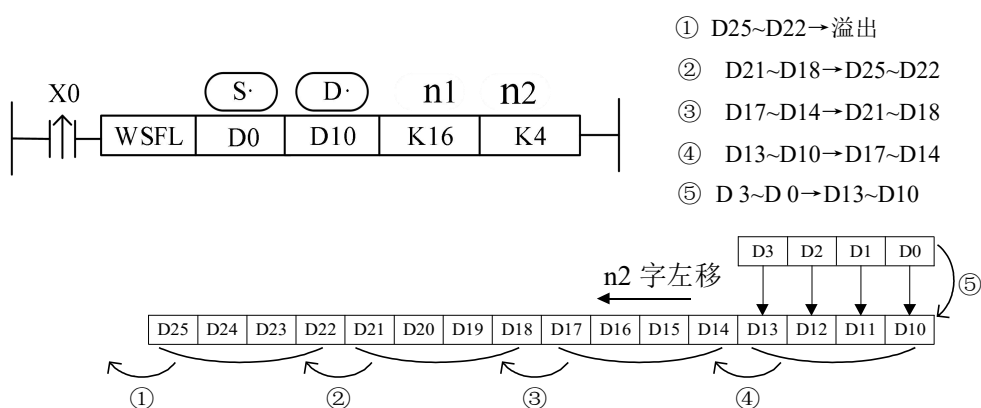
操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位, BIN
D	指定目标软元件的首地址编号	16 位, BIN
n1	指定目标软元件的个数 (不超过 512)	16 位, BIN
n2	指定每次左移的字个数 (不超过 512)	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√	√	√	√		
	D		√	√	√	√	√	√		
	n1									√
	n2									√

## 功能和动作

- 以字为单位, 对 n1 个字的字软元件进行 n2 个字的左移的指令。
- 执行脉冲指令时, 驱动输入 X0 从 OFF→ON 时就执行一次 n2 个字的移动。



- 上例中, D10~D25 (共 16 个寄存器) 用于接收从 D0~D3 传送过来的数值, 每次 X0 上升沿来时, 将 D0~D3 的数值传送到 D10~D13, 原 D0~D13 的数值左移到 D14~D17, 原 D14~D17 的数值左移到 D18~D21, ……依次类推, 原 D22~D25 的数值溢出。

## 4-7-7. 字右移[WSFR]

## 1、指令概述

将指定软元件中的数据进行字右移的指令。

字右移[WSFR]			
16 位指令	WSFR (P)	32 位指令	-
执行条件	边沿触发	适用机型	CZA

## 2、操作数

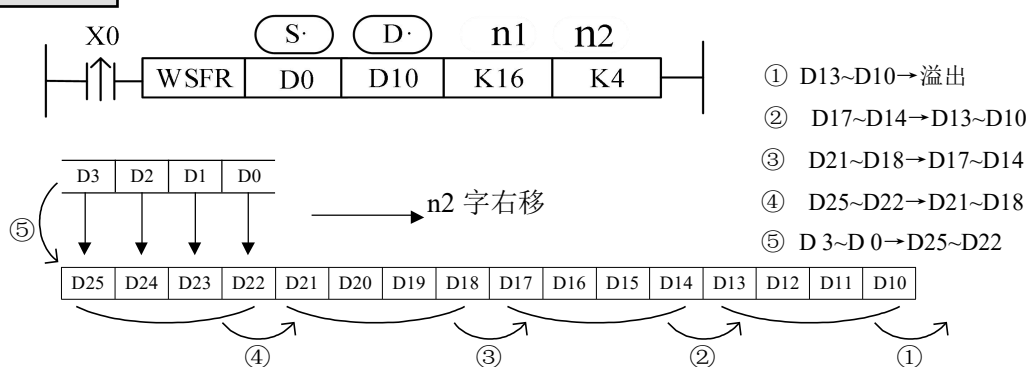
操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位, BIN
D	指定目标软元件的首地址编号	16 位, BIN
n1	指定目标软元件的个数 (不超过 512)	16 位, BIN
n2	指定每次右移的字个数 (不超过 512)	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S		√	√	√	√	√	√	√		
D			√	√	√	√	√	√		
n1										√
n2										√

## 功能和动作

- 以字为单位, 对 n1 个字的字软元件进行 n2 个字的右移的指令。
- 执行脉冲指令时, 驱动输入 X0 从 OFF→ON 时就执行一次 n2 个字的移动。



- 上例中, D10~D25 (共 16 个寄存器) 用于接收从 D0~D3 传送过来的数值, 每次 X0 上升沿来时, 将 D0~D3 的数值传送到 D22~D25, 原 D22~D25 的数值右移到 D18~D21, 原 D18~D21 的数值右移到 D14~D17, ……依次类推, 原 D13~D10 的数值溢出。

**4-8. 数据转换指令**

指令助记符	指令功能	章节
FLT	16 位整数转浮点	4-8-1
DFLT	32 位整数转浮点	4-8-1
INT	浮点转整数	4-8-2
DECO	译码	4-8-3
ENCO	高位编码	4-8-4



## 4-8-1. 16 位整数转浮点数[FLT]

## 1、指令概述

将指定数据或软元件中的整数转换为浮点数的指令。

16 位整数转浮点数[FLT]			
16 位指令	FLT	32 位指令	DFLT
执行条件	常开/闭、边沿触发		适用机型   CZA

## 2、操作数

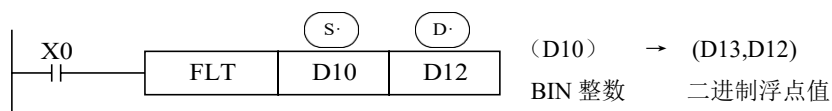
操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位/32 位/64 位, BIN
D	指定目标软元件的首地址编号	32 位/64 位, BIN

## 3、适用软元件

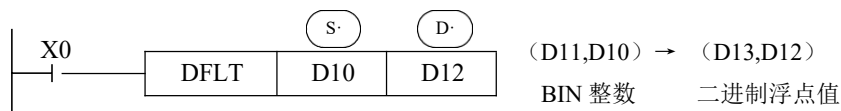
字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	S							√	
D							√		

## 功能和动作

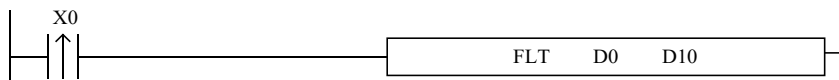
## 《16 位指令》



## 《32 位指令》



- 二进制整数与二进制浮点值间的转换指令。常数 K、H 在各浮点运算指令中不能自动转换，所以需先把常数 K、H 先转换成浮点数才可以进行浮点运算。
- 这个指令的逆变换指令是 INT。



初始设 D0 的值为整数 20，执行指令后，D10 的值为浮点数 20。在自由监控中添加 D10，选择浮点类型，可以正确的监控到 D10 的值。自由监控数据如下：

PLC1-自由监控			
<input checked="" type="button" value="监控"/> <input type="button" value="添加"/> <input type="button" value="修改"/> <input type="button" value="删除"/> <input type="button" value="上移"/> <input type="button" value="下移"/>			
寄存器	监控值	字长	进制
D0	20	单字	10进制
D10	20	浮点	10进制
D10	1101004800	双字	10进制

上图所示，D0 为整数 20，D10 为浮点数 20，当 D10 选择双字类型监控时，数据显示不是 20。这是因为整数和浮点数在底层存放格式不一样，所以监控浮点数时应该用自由监控，选择浮点类型监控，才能查看到正确的数据。

## 4-8-2. 浮点转整数[INT]

## 1、指令概述

将指定软元件中的浮点数转换为整数的指令。

浮点转整数[INT]			
16 位指令	INT	32 位指令	DINT
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

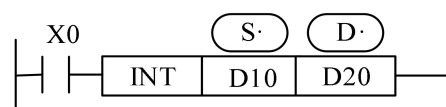
操作数	作用	类型
S	指定源数据的软元件首地址编号	16 位/32 位, BIN
D	指定目标软元件的首地址编号	16 位/32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S								√		
D								√		

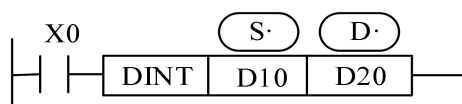
## 功能和动作

## 《16 位指令》



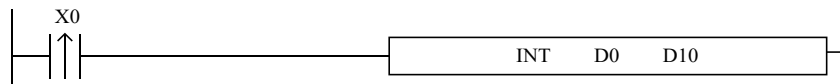
(D11,D10) → (D20)  
二进制浮点 → BIN 整数  
小数点以后的数舍去

## 《32 位指令》



(D11,D10) → (D20,D21)  
二进制浮点 → BIN 整数  
小数点以后的数舍去

- 将源数据地址内的二进制浮点值转换为 BIN 整数，存入目标地址中。此时，舍去小数点以后的值。
- 此指令为 FLT 指令的逆变换。
- 运算结果为 0 时，标志位为 ON。
- 运算结果超出以下范围而发生溢出时，进位标志位 ON。  
16 位运算时：-32,768~32,767  
32 位运算时：-2,147,483,648~2,147,483,647



假设D0中的浮点数为 130.2，执行 INT 指令后，得到整数 130 存放于 D10 中，如下图所示：

PLC1-自由监控			
监控 添加 修改 删除 上移 下移			
寄存器	监控值	字长	进制
D0	130.2	浮点	10进制
D10	130	单字	10进制

## 4-8-3. 译码[DECO]

## 1、指令概述

将任意一个数字数据转换为 1 点的 ON 位的指令。

译码[DECO]			
16 位指令	DECO(P)	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	指定要译码的字软元件地址编号	16 位, BIN
D	指定译码结果的字或位软元件的首地址编号	16 位, BIN
n	指定要译码的软元件的位的点数	16 位, BIN

## 3、适用软元件

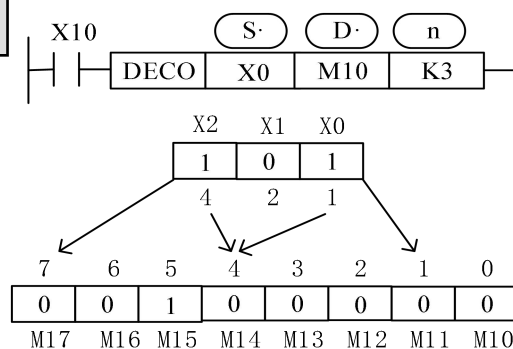
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
S						√	√	√	√	√	
D						√	√	√			
n										√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
S		√	√	√	√		
D			√	√	√		

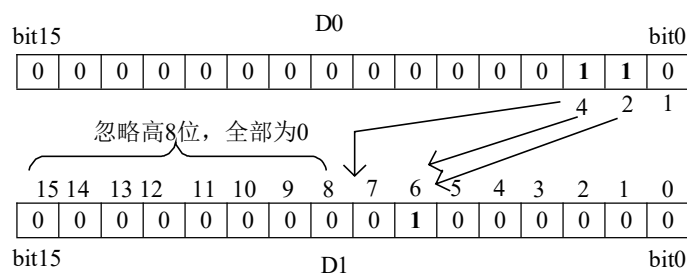
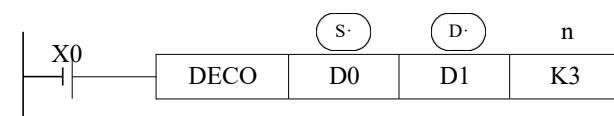
## 功能和动作

《 $\text{D}$ 》是位软元件时  $n \leq 8$



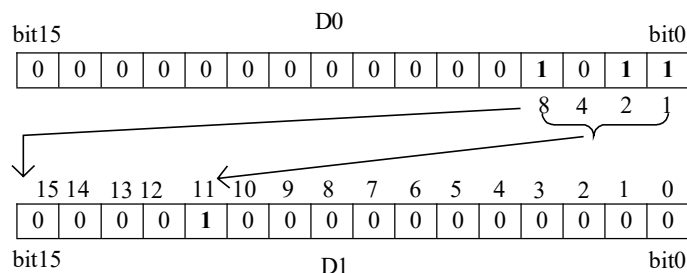
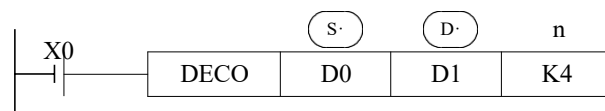
- $n=3$ , 所以译码对象为  $X0$  中的低 3 位, 即  $X2 \sim X0$ 。
- $n=3$ , 所以译码结果需要由  $2^3=8$  个位来表示, 即  $M17 \sim M10$ 。
- 当  $X2=1, X1=0, X0=1$ , 其所代表的数值是  $4+1=5$ , 因此从  $M10$  起第 5 位的  $M14$  变为 1; 当  $X2 \sim X0$  全部为 0 时, 数值也为 0, 所以  $M10$  为 1 ( $M10$  为第 0 位)。  
当  $X2 \sim X0$  全部为 1 时, 数值为 7, 所以  $M17$  为 1。  
当  $X2=0, X1=0, X0=1$ , 数值为 1, 所以  $M11$  为 1。
- $n=0$  时不处理,  $n=0 \sim 16$  以外的数值时会不执行指令。
- $n=16$  时, 如果译码命令  $\text{D}$  为位软元件时, 其点数是  $2^{16}=65536$ 。
- 驱动输入为 OFF 时, 指令不执行, 正在动作的译码输出保持动作。

《(D) 是字软件时》  $n \leq 4$



- 源地址的低  $n$  位 ( $n \leq 4$ ) 被解码至目标地址。  $n \leq 3$  时，目标的高 8 位都转为 0。
- $n=0$  时不处理， $n=0\sim 4$  以外时，不执行指令。
- $n=3$ ，所以 D0 中的译码对象为 bit2~bit0，其所表示的最大数值是  $4+2+1=7$ 。
- $n=3$ ，所以 D1 中需要  $2^3=8$  个位来表示译码结果，即 bit7~bit0。
- 当 bit2、bit1 均为 1，bit0 为 0，其所表示的数值是  $4+2=6$ ，因此 D1 中的 bit6 置 ON。

《(D) 是字软件时》  $n \leq 4$



- 源地址的低  $n$  位 ( $n \leq 4$ ) 被解码至目标地址。  $n \leq 3$  时，目标的高 8 位都转为 0。
- $n=0$  时不处理， $n=0\sim 4$  以外时，不执行指令。
- $n=4$ ，所以 D0 中的译码对象为 bit3~bit0，其所表示的最大数值是  $8+4+2+1=15$ 。
- $n=4$ ，所以 D1 中需要  $2^4=16$  个位来表示译码结果，即 bit15~bit0。
- 当 bit3、bit1、bit0 均为 1，bit2 为 0，其所表示的数值是  $8+2+1=11$ ，因此 D1 中的 bit11 置 ON。

## 4-8-4. 高位编码[ENCO]

## 1、指令概述

求出在数据中为 ON 的最高位的位置的指令。

高位编码[ENCO]			
16 位指令	ENCO	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	指定要编码的字或位软元件地址编号	16 位, BIN
D	指定编码结果的软元件的地址编号	16 位, BIN
n	指定编码结果的软元件的位的点数	16 位, BIN

## 3、适用软元件

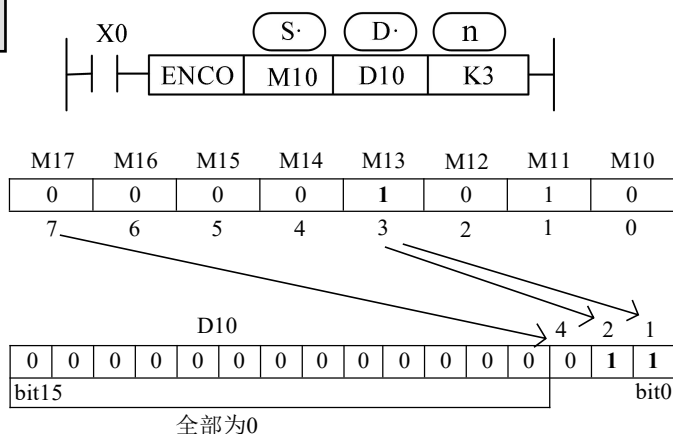
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
S						√	√	√	√		
D						√	√	√	√		
n										√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
S		√	√	√	√		

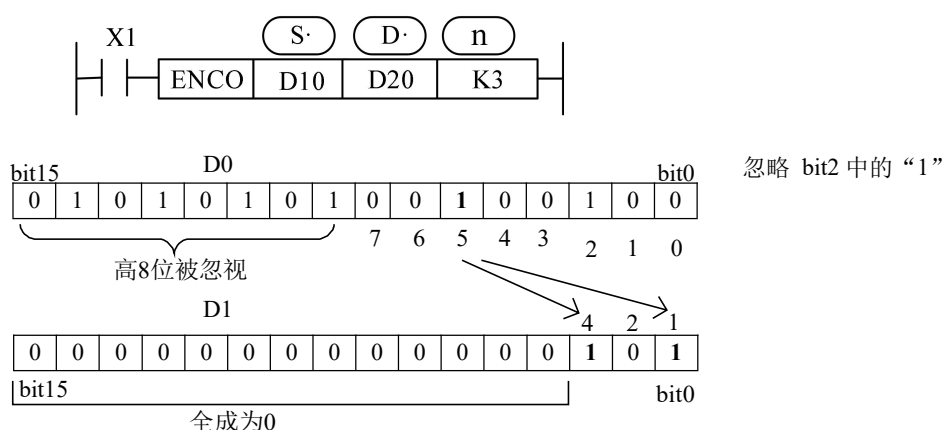
## 功能和动作

《(s) 是位软元件时》  $n \leq 16$



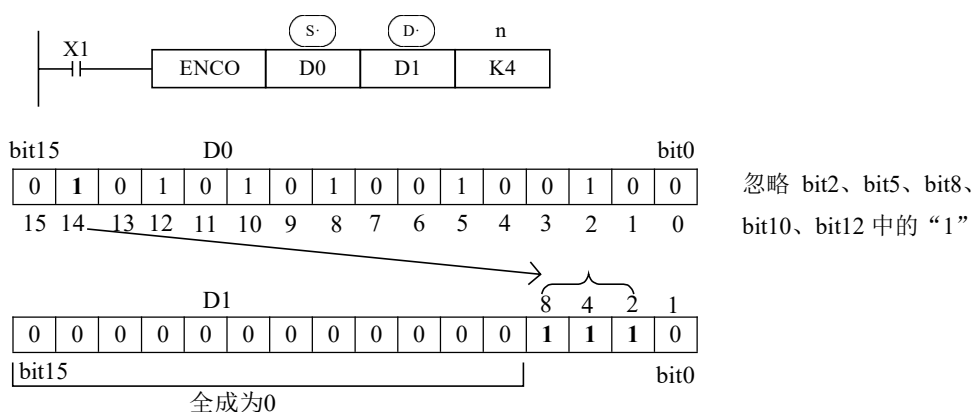
- 源地址内的多个位是 1 时，忽略低位侧，另外源地址都为 0 时会不执行指令。
- 驱动条件为 OFF 时，指令不被执行，编码输出不变化。
- $n=16$  时，编码指令的 (s) 如果是位元件，其点数为  $2^{16}=65536$ 。
- $n=3$ ，被编码对象有  $2^3=8$  位，即 M17~M10，编码结果存放在 D10 中的低 3 位，即 bit2~bit0。
- M13 和 M11 均为 1，忽略 M11，对 M13 编码，以 bit2~bit0 表示 3，则 bit0 和 bit1 为 1。

《(s) 是字软元件时》  $n \leq 4$



- 源地址内的多个位是 1 时，忽略低位侧，另外源地址都为 0 时会不执行指令。
- 驱动输入为 OFF 时，指令不被执行，编码输出不变化。
- $n \leq 3$  时，D0 中的高 8 位被忽视。
- $n=3$ ，被编码对象有  $2^3=8$  位，即 D0 中的 bit7~bit0，编码结果存放在 D1 中的低 3 位，即 bit2~bit0。
- D0 中的 bit5 和 bit2 均为 1 时，忽略 bit2，对 bit5 编码，以 bit2~bit0 表示 5，则 bit2 和 bit0 为 1。

《(s) 是字软元件时》  $n \leq 4$



- 源地址内的多个位是 1 时，忽略低位侧，另外源地址都为 0 时会不执行指令。
- 驱动输入为 OFF 时，指令不被执行，编码输出不变化。
- $n=4$ ，被编码对象有  $2^4=16$  位，即 D0 中的 bit15~bit0，编码结果存放在 D1 中的低 4 位，即 bit3~bit0。
- D0 中为 1 的最高位为 bit14，忽略所有低位的 1，对 bit14 编码，以 bit3~bit0 表示 14，则 bit3、bit2 和 bit1 为 1。



## 4-9. 浮点运算指令

指令助记符	指令功能	章节
DECMP	浮点数比较	4-9-1
DEZCP	浮点数区间比较	4-9-2
DEADD	浮点数加法	4-9-3
DESUB	浮点数减法	4-9-4
DEMUL	浮点数乘法	4-9-5
DEDIV	浮点数除法	4-9-6
DESQR	浮点数开方	4-9-7
DSIN	浮点数 SIN 运算	4-9-8
DCOS	浮点数 COS 运算	4-9-9
DTAN	浮点数 TAN 运算	4-9-10
DEMOV	浮点数传送指令	4-9-11

注意：

(1) 使用浮点运算指令时，若操作数是确定的浮点数，那么可以直接书写数值；若操作数为常数时，需先用数据寄存器存放这个常数，再把这个寄存器转换成浮点数；若操作数是寄存器类型的必须要先把寄存器转换成浮点型再执行浮点运算指令，否则程序不执行。

(2) 查看浮点运算结果时，请务必在自由监控中添加要监控对象，监控模式为浮点，这样就可以看到浮点结果。因为浮点数的存储方式和整数不同，所以梯形图中显示的数值不能直接当作浮点值。

## 4-9-1. 浮点数比较[DECMP]

## 1、指令概述

比较两个源数据内的二进制浮点数的指令。

浮点数比较[DECMP]			
16 位指令	-	32 位指令	DECMP(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

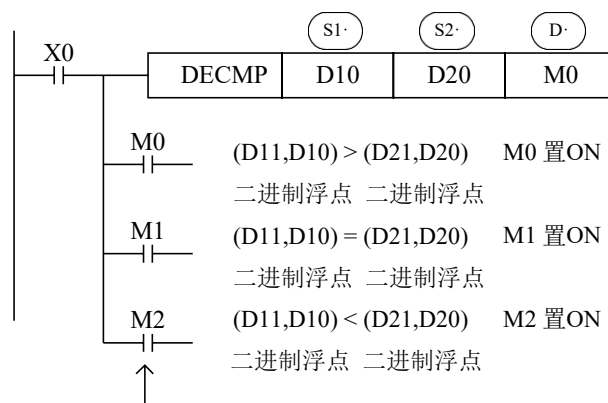
操作数	作用	类型
S1	指定进行比较的数据或软元件地址编号	32 位, BIN
S2	指定进行比较的数据或软元件地址编号	32 位, BIN
D	指定保存比较结果的软元件首地址编号	位

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		√
S2							√		√	
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	D		√	√	√					

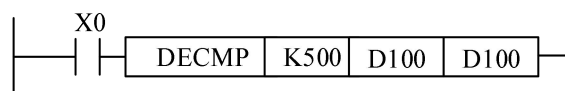
## 功能和动作

(D11,D10) : (D21,D20) → M0,M1,M2  
二进制浮点 二进制浮点



X0 为 OFF 时, 即使 DECMP 指令不执行, M0~M2 保持X0 为OFF 以前的状态。

- 比较两个源数据内的二进制浮点值，根据大小一致比较结果，对应输出 M0 开始的 3 点 ON/OFF 状态。
- 常数 K,H 被指定为源数据时，自动转换成二进制浮点值处理。



(K500) : (D101, D100) → M10, M11, M12

自动二进制 二进制浮点化

浮点化

## 4-9-2. 浮点数区间比较[DEZCP]

## 1、指令概述

将指定数据进行上下两点的范围比较的指令。

浮点数区间比较[DEZCP]			
16 位指令	-	32 位指令	DEZCP(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定比较基准下限的数据或软元件地址编号	32 位, BIN
S2	指定比较基准上限的数据或软元件地址编号	32 位, BIN
S3	指定当前比较数据或软元件地址编号	32 位, BIN
D	指定比较结果的软元件首地址编号	位

## 3、适用软元件

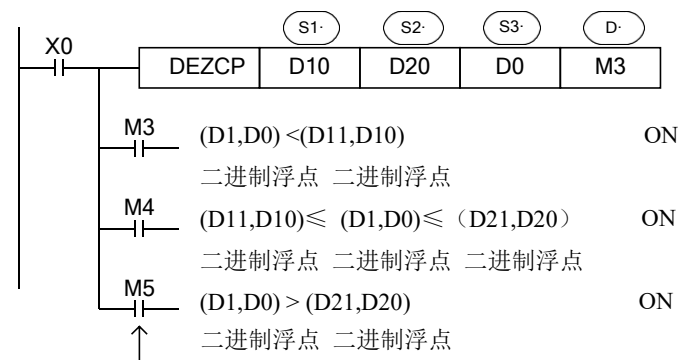
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1							√		√	
	S2							√		√	
	S3							√		√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
	D		√	√	√		

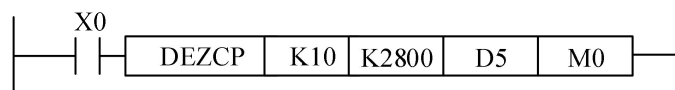
## 功能和动作

对2点的设定值的大小比较。



X0 为 OFF 时, 即使 EZCP 指令不执行, M3~M5 保持 X0 为 OFF 以前的状态。

- 将 [ (S), (S) + 1 ] 的内容与用二进制浮点值指定的上下 2 点的范围比较, 对应输出 (D) 开始的 3 点 ON/OFF 状态。
- 常数 K、H 被指定为源数据时, 自动转换成二进制浮点值处理。



(K10) : [D6,D5] : (K2800) →M0, M1, M2

自动二进 二进制 自动二进  
制浮点化 浮点数 制浮点化

- 请设置 (S1) ≤ (S2)，当 (S2) < (S1) 时，将 (S2) 的数值当作和 (S1) 相同进行比较。

## 4-9-3. 浮点数加法[DEADD]

## 1、指令概述

将两个数据进行浮点数相加运算的指令。

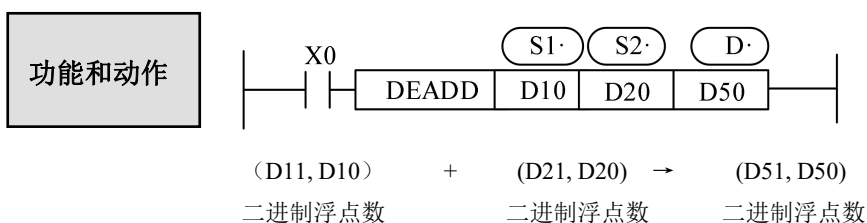
浮点数加法[DEADD]			
16 位指令	-	32 位指令	DEADD(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定进行相加的数据或软元件地址编号	32 位, BIN
S2	指定进行相加的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		√
	S2							√		√
	D							√		



- 两个数据源内的二进制浮点值相加后，作为二进制浮点值存入目的地址之中。
- 常数 K, H 被指定为源数据时，自动转换成二进制浮点值处理。



- 源数据和目的地址也可以指定同一元件号。请注意，当 X0 为 ON 时，在每个扫描周期均相加一次,或者使用EADDP指令只会执行一次。

## 4-9-4. 浮点数减法[DESUB]

## 1、指令概述

将两个数据进行浮点数相减运算的指令。

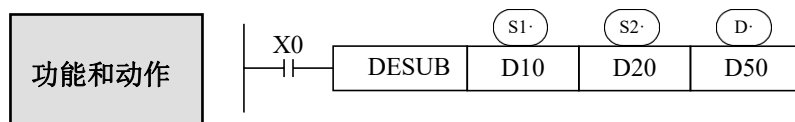
浮点数减法[DESUB]			
16 位指令	-	32 位指令	DESUB(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定进行相减的数据或软元件地址编号	32 位, BIN
S2	指定进行相减的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

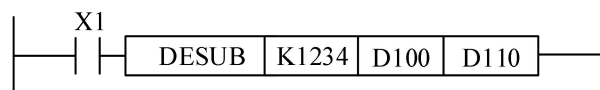
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		√
	S2							√		√
	D							√		



(D11, D10) - (D21, D20) → (D51, D50)  
二进制浮点      二进制浮点      二进制浮点

- (S1) 指定的元件内的二进制浮点值减去 (S2) 指定的元件内的二进制浮点值，并将其结果作为二进制浮点值存入目的地址之中。
- 常数 K, H 被指为源数据时，自动转换成二进制浮点值处理。



(K1234) - (D101, D100) → (D111, D110)  
自动二进      二进制浮点      二进制浮点  
制浮点化

- 源数据和目的地址也可以指定同一元件号。请注意，当 X0 为 ON 时，在每个扫描周期均相减一次或者使用ESUBP指令只会执行一次。

## 4-9-5. 浮点数乘法[DEMUL]

## 1、指令概述

将两个数据进行浮点数相乘运算的指令。

浮点数乘法[DEMUL]			
16 位指令	-	32 位指令	DEMUL(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

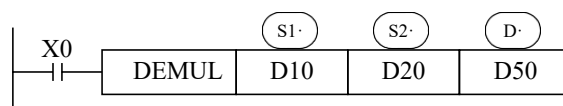
## 2、操作数

操作数	作用	类型
S1	指定进行相乘的数据或软元件地址编号	32 位, BIN
S2	指定进行相乘的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		√
	S2							√		√
	D							√		

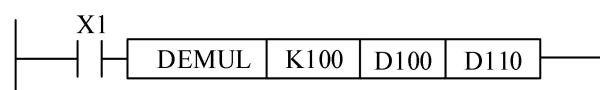
## 功能和动作



$$(D11, D10) \times (D21, D20) \rightarrow (D51, D50)$$

二进制浮点      二进制浮点      二进制浮点

- 将两个源数据内的二进制浮点值的积作为二进制浮点值存入目的地址中。
- 常数 K、H 被指定为源数据时，自动转换为二进制浮点值处理。
- 源数据和目的地址也可以指定同一元件号。请注意，当 X0 为 ON 时，在每个扫描周期均相减一次或者使用EMULP指令只会执行一次



$$(K100) \times (D101, D100) \rightarrow (D111, D110)$$

自动二进      二进制浮点      二进制浮点  
制浮点化



## 4-9-6. 浮点数除法[DEDIV]

## 1、指令概述

将两个数据进行浮点数相除运算的指令。

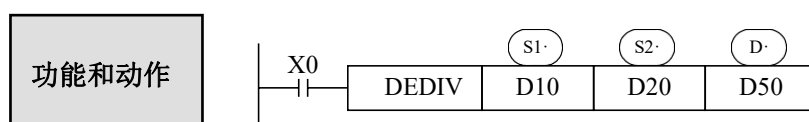
浮点数除法[DEDIV]			
16 位指令	-	32 位指令	DEDIV(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	进行相除的数据或软元件地址编号	32 位, BIN
S2	进行相除的数据或软元件地址编号	32 位, BIN
D	存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

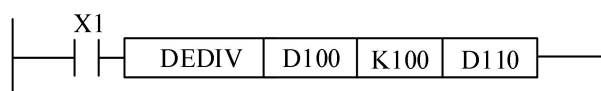
字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		√
	S2							√		√
	D							√		



$$(D11, D10) \div (D21, D20) \rightarrow (D51, D50)$$

二进制浮点      二进制浮点      二进制浮点

- (S1) 指定的元件内的二进制浮点值除以用 (S2) 指定的元件内的二进制浮点值，并将其结果作为二进制浮点值存入目的地址之中。
- 常数 K、H 被指为源数据时，自动转换成二进制浮点值处理。



$$(D101, D100) \div (K100) \rightarrow (D111, D110)$$

二进制浮点数      自动二进制浮点化      二进制浮点数

- 除数 (S2) 为 0 时，则运算错误，指令不能执行。

## 4-9-7. 浮点数开方[DESQR]

## 1、指令概述

对指定数据进行浮点数开方运算的指令。

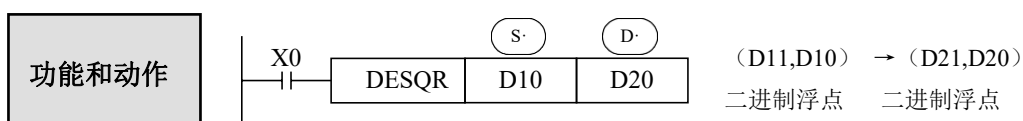
浮点数开方[DESQR]			
16 位指令	-	32 位指令	DESQR(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

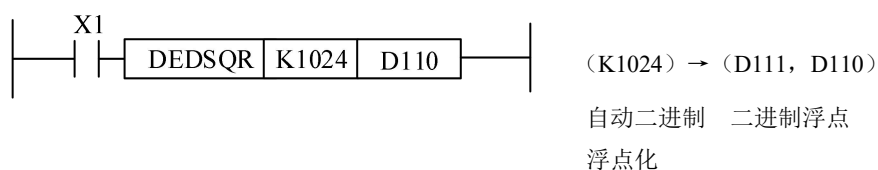
操作数	作用	类型
S	指定进行开方运算的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S								√		√
D								√		



- 进行用源数据指定的元件内二进制浮点值的平方根运算，作为二进制浮点数存入目的地址之中。
- 常数 K、H 被指为源数据时，自动转换成二进制浮点值处理。



- 运算结果为零时，零标志位动作。
- 源数据的内容只有正数时有效，负数时运算错误（M8067）动作，指令不能执行。

## 4-9-8. 浮点 SIN 运算[DSIN]

## 1、指令概述

对指定数据进行浮点数 SIN 运算的指令。

浮点 SIN 运算[DSIN]			
16 位指令	-	32 位指令	DSIN(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

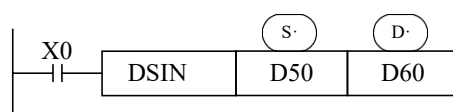
## 2、操作数

操作数	作用	类型
S	指定进行 SIN 运算的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S							√		
	D							√		

## 功能和动作

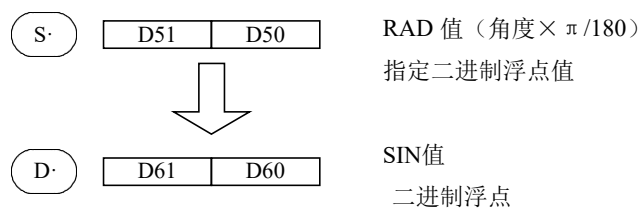


(D51, D50) → (D61, D60) SIN

二进制浮点

二进制浮点

- 此为求源数据指定的弧度 (RAD) 的 SIN 值, 并传送到目的地址中的指令。



### 4-9-9. 浮点 COS 运算[DCOS]

#### 1、指令概述

对指定数据进行浮点 COS 运算的指令。

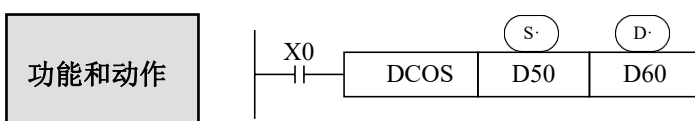
浮点COS 运算[DCOS]			
16 位指令	-	32 位指令	DCOS(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

#### 2、操作数

操作数	作用	类型
S	指定进行COS 运算的数据或软元件地址编号	32 位, BIN
D	指定存储运算结果的软元件地址编号	32 位, BIN

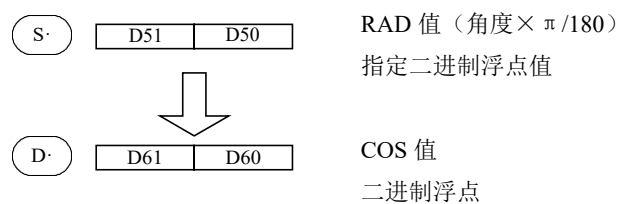
#### 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S								√		
D								√		



(D51, D50)RAD → (D61, D60) COS  
二进制浮点                  二进制浮点

- 此为求源数据指定的弧度 (RAD) 的 COS 值, 并传送到目的地址中的指令。



## 4-9-10. 浮点 TAN 运算[DTAN]

## 1、指令概述

对指定数据进行浮点 TAN 运算的指令。

浮点TAN 运算[DTAN]			
16 位指令	-	32 位指令	DTAN(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

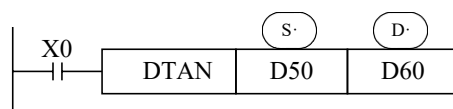
## 2、操作数

操作数	作用	类型
S	进行TAN 运算的数据或软元件地址编号	32 位, BIN
D	存储运算结果的软元件地址编号	32 位, BIN

## 3、适用软元件

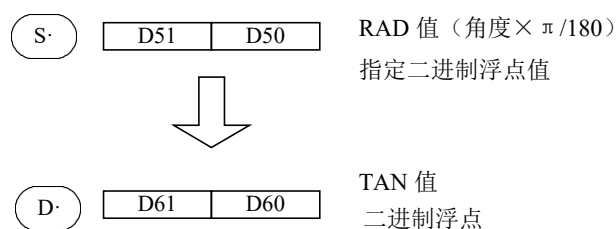
字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S								√		
D								√		

## 功能和动作



(D51, D50) RAD → (D61, D60) TAN  
二进制浮点                      二进制浮点

- 此为求源数据指定的弧度 (RAD) 的 TAN 值, 并传送到目的地址中的指令。



## 4-9-11. 浮点数传送[DEMOV]

## 1、指令概述

将指定软元件中的浮点数照原样传送到其他软元件中。

浮点数传送[EMOV]			
16 位指令	-	32 位指令	DEMOV(P)
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	指定传送源的数据或保存数据的软元件编号	32 位, BIN
D	指定传送的目标软元件起始地址编号	32 位, BIN

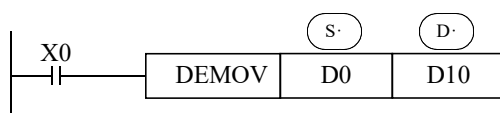
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S	√	√	√	√			√		√
	D		√	√	√			√		

## 功能和动作

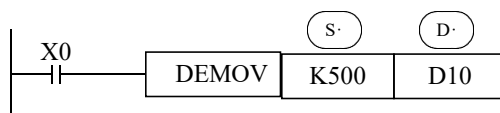
《32 位指令》

二进制浮点 → 二进制浮点



(D1,D0) → (D11,D10)

- X0 为 ON 时，将源的浮点数向目标传送。
- X0 为 OFF 时，D11、D10 不变化。



(K500) → (D11,D10)

- 常数 K, H 被指定为源数据时，自动转换成二进制浮点值处理。
- K500 自动二进制浮点化。

## 4-10. 时钟指令

指令助记符	指令名称	章节
TRD	时钟数据读取	4-10-1
TWR	时钟数据写入	4-10-2
TCMP	时钟数据比较	4-10-3
TZCP	时钟数据区间比较	4-10-4
TADD	时钟数据加法	4-10-5
TSUB	时钟数据减法	4-10-6

---

1: 不含时钟的机型不可以使用时钟指令。

2: CZA 系列PLC 的时钟存在一定误差, 大约为每月误差 $\pm 2$  分钟, 可通过PLC上位机在程序里定期进行校准。

---

## 4-10-1. 时钟数据读取[TRD]

## 1、指令概述

读取时钟数据的指令。

时钟数据读取[TRD]			
16 位指令	TRD	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

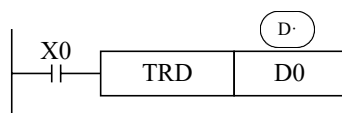
## 2、操作数

操作数	作用	类型
D	保存时钟数据的软元件首地址编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
D						√	√	√		

## 功能和动作



将可编程控制器的实时时钟的时钟数据读入7点数据寄存器中的指令。

- 按照下列格式读取可编程控制器的实时时钟数据。  
读取源为保存时钟数据的特殊数据寄存器（D8013~D8019）。

	元件	项目	时钟数据	→	元件	项目
特殊实时数据寄存器	D8018	年（公历）	2000~	→	D0	年（公历）
	D8017	月	1~12	→	D1	月
	D8016	日	1~31	→	D2	日
	D8015	时	0~23	→	D3	时
	D8014	分	0~59	→	D4	分
	D8013	秒	0~59	→	D5	秒
	D8019	星期	0(日)~6(六)	→	D6	星期

- 实时时钟数据在特殊数据寄存器（D8013~D8019）中是以 BCD 码形式存放，如需要数据监控时钟信息，请选择十六进制监控。
- 通过指令TRD 读取出来的时钟数据，是十进制形式的，监控时选用十进制即可。
- 该指令执行一次后，D0~D6 这 7 个寄存器均被占用，分别存放时钟信息中的年、月、日、时、分、秒、星期。



## 4-10-2. 时钟数据写入[TWR]

## 1、指令概述

写入时钟数据的指令。

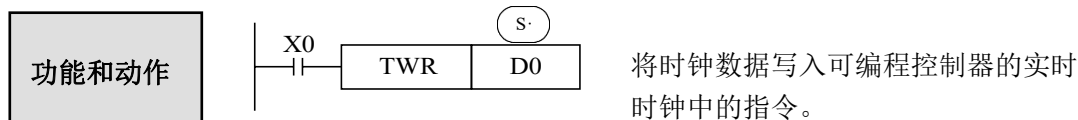
时钟数据读取[TWR]			
16 位指令	TWR	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S	写入时钟数据的软元件地址编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
S						√	√	√		



- 将设定时钟的数据写入可编程控制器的实时时钟内。

为了写入时钟数据，必须预先设定由 (S) 指定的元件地址号起始的 7 个数据寄存器。

时钟设定用数据	元件	项目	时钟数据	→	元件	项目	特殊实时数据寄存器
	D0	年 (公历)	2000~	→	D8018	年 (公历)	
	D1	月	1~12	→	D8017	月	
	D2	日	1~31	→	D8016	日	
	D3	时	0~23	→	D8015	时	
	D4	分	0~59	→	D8014	分	
	D5	秒	0~59	→	D8013	秒	
	D6	星期	0(日)~6(六)	→	D8019	星期	

执行 TWR 指令后，立即变更实时时钟的时钟数据，变为新时间。因此，请提前数分钟向源数据传送时钟数据，这样当到达正确时间时，请执行指令。

执行 TWR 指令后，立即变更实时时钟的时钟数据，变更为新时间。

因此，请提前数分钟向源数据传送时钟数据，这样当到达正确时间时，请执行指令。另外利用本指令校准时间时，无须控制特殊继电器 M8015 (时间停止和时间校准)。

## 4-10-3. 时钟数据比较[TCMP]

## 1、指令概述

将比较基准时间与时钟数据（时、分、秒）进行比较的指令。

时间比较[TCMP]			
16 位指令	TCMP	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

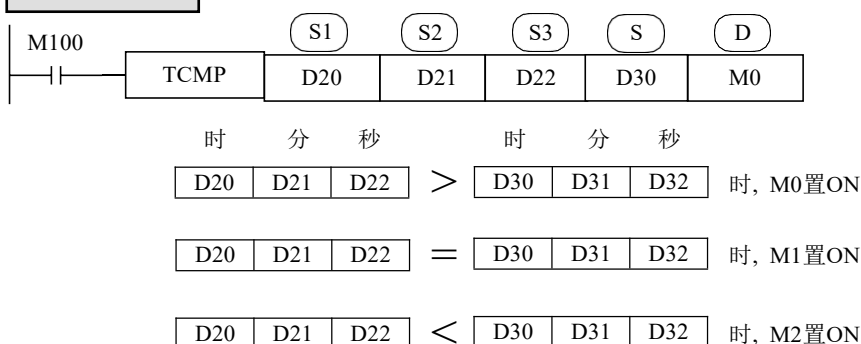
## 2、操作数

操作数	作用	类型
S1	指定比较基准时间“时”的软元件地址编号	16 位, BIN
S2	指定比较基准时间“分”的软元件地址编号	16 位, BIN
S3	指定比较基准时间“秒”的软元件地址编号	16 位, BIN
S	指定时钟数据（时、分、秒）的软元件首地址编号	16 位, BIN
D	指定比较结果的软元件首地址编号	位

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
	S3	√	√	√	√	√	√	√	√	√
S					√	√	√			
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	D		√	√	√					

## 功能和动作

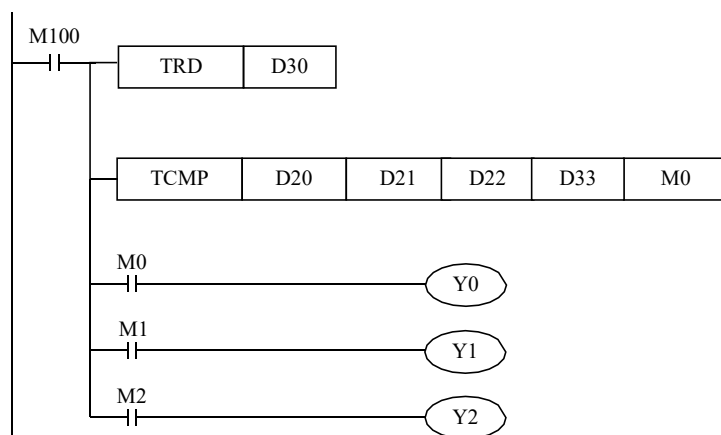


- M100 由 OFF 变为 ON 时, TCMP 指令执行, 以 D30 开始的三个寄存器（即时钟的

时、分、秒)和D20、D21、D22组成的时、分、秒进行比较,并按照比较结果置ON相应的线圈。当D20、D21、D22组成的时、分、秒大于D30、D31、D32,则M0置ON;当D20、D21、D22组成的时、分、秒等于D30、D31、D32,则M1置ON,当D20、D21、D22组成的时、分、秒小于D30、D31、D32,则M2置ON。

- M100为ON状态时,如果比较基准时间(D20、D21、D22)或时钟数据(D30、D31、D32)发生变化,则比较结果也会相应地发生变化。
- M100置OFF即停止执行TCMP指令时,M0~M2仍然保持M100置OFF前的状态。

#### 举例



例如上例中,读得当前的时钟是2014年7月30日15时32分49秒星期三,则D33=15, D34=32, D35=49,如果设定时钟是16时40分21秒,即D20=16, D21=40, D22=21,则Y0=ON;如果设定时钟是15时21分16秒,即D20=15, D21=21, D22=16,则Y2=ON;如果设定时钟是15时32分49秒,即D20=15, D21=32, D22=49,则Y1=ON。

## 4-10-4. 时钟数据区间比较[TZCP]

## 1、指令概述

将指定的两个基准时间（时、分、秒）与时钟数据（时、分、秒）进行比较的指令。

时间区间比较[TZCP]			
16 位指令	TZCP	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

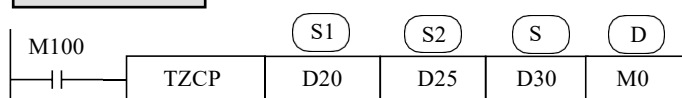
## 2、操作数

操作数	作用	类型
S1	指定第一基准时间值“时、分、秒”的软元件首地址编号	16 位, BIN
S2	指定第二基准时间值“时、分、秒”的软元件首地址编号	16 位, BIN
S	指定待比较时间值（时、分、秒）的软元件首地址编号	16 位, BIN
D	指定比较结果的软元件首地址编号	位

## 4、适用软元件

字软 元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		
	S2					√	√	√		
位软 元件	S3									
	操作数	系统								
		X	Y	M	S	T	C			
D		√	√	√						

## 功能和动作



时 分 秒

时 分 秒

时 分 秒

D20	D21	D22
-----	-----	-----

&gt;

D30	D31	D32
-----	-----	-----

M0 闭合

D20	D21	D22
-----	-----	-----

≤

D30	D31	D32
-----	-----	-----

≤

D25	D26	D27
-----	-----	-----

M1 闭合

D30	D31	D32
-----	-----	-----

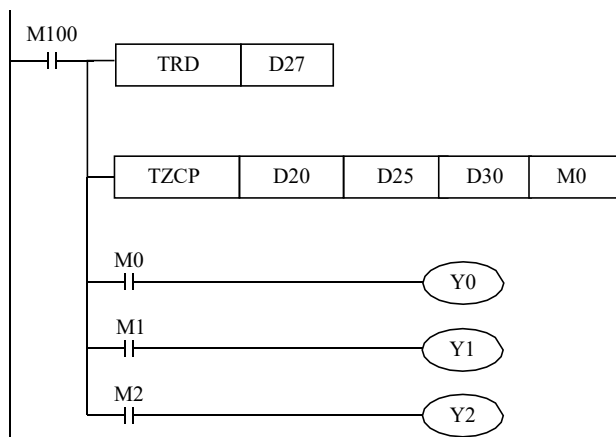
&gt;

D25	D26	D27
-----	-----	-----

M2 闭合

- M100 由 OFF 变为 ON 时，TZCP 指令执行，将“D20、D21、D22”、“D25、D26、D27”中的时间值与“D30、D31、D32”中的时间值进行比较，根据比较结果驱动 M0~M5,具体如下:
  - a:若“D30、D31、D32”中的时间值小于“D20、D21、D22”中的时间值，M0被驱动，M0常开触点闭合。
  - b:若“D30、D31、D32”中的时间值处于“D20、D21、D22”和“D25、D26、D27”时间值之间，M2被驱动，M2常开触点闭合。
  - c:若“D30、D31、D32”中的时间值大于“D25、D26、D27”中的时间值，M3被驱动，M3常开触点闭合。
- M100 置 OFF 即停止执行TZCP 指令时，M0~M2 仍然保持 M100 置 OFF 前的状态。

举例



例如上例中，读得当前的时钟是 2014 年7月30日15时32分49秒星期三，则 D30=15，D31=32，D32=49，如果设定第一基准时间值 16 时40分21秒，即 D20=16，D21=40，D22=21，则 Y0=ON；如果设定第一基准时间值15时21分16秒，即D20=15，D21=21，D22=16，第二基准时间值15时50分50秒，即D25=15，D26=50，D27=50，则 Y1=ON；如果设定第二基准时间值15时20分20秒，即D25=15，D26=20，D27=20,则Y2=ON。

## 4-10-5. 时钟数据加法[TADD]

### 1. 指令概述

两个时钟数据（时、分、秒）做加法运算的指令。

时钟数据加法运算[TADD]			
16 位指令	TADD	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

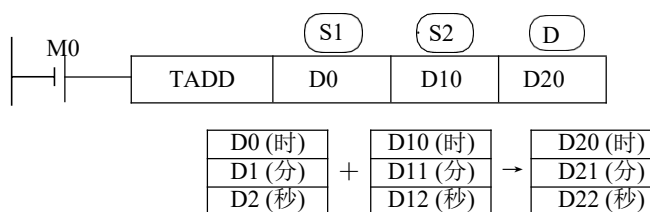
### 2. 操作数

操作数	作用	类型
S1	指定加法运算的时钟数据（时、分、秒）的软元件首地址编号	16 位, BIN
S2	指定加法运算的时钟数据（时、分、秒）的软元件首地址编号	16 位, BIN
D	指定存储加法运算结果的软元件地址编号	16 位, BIN

### 3. 适用软元件

操作数	系统									
	DX	DY	DM	DS	T	C	D	V、Z	K、H	
S1					√	√	√			
S2					√	√	√			
D					√	√	√			

### 功能和动作



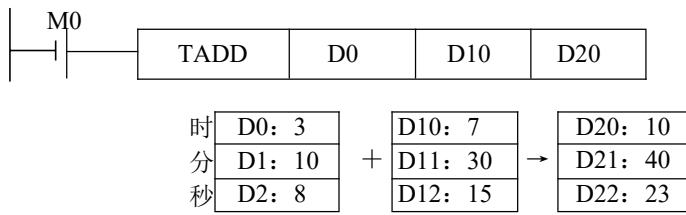
- M0 置 ON 时，执行时钟加法运算，将（D0、D1、D2）的时钟数据（时、分、秒）与（D10、D11、D12）的时钟数据（时、分、秒）进行相加，并将运算结果（时、分、秒）存储到（D20、D21、D22）中。

**注意：**寄存器的对应关系是固定的，即按照时、分、秒的顺序依次存放。

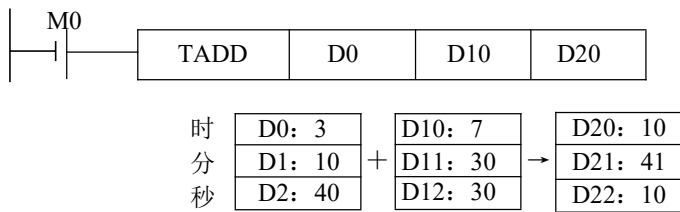
- 时的范围：0~23；分的范围：0~59，秒的范围：0~59。
- 若加法运算后的秒、分超过 59，将运算结果减去 60 后保存在秒、分寄存器中，同时分、小时的数值自动加 1；
- 若加法运算后的小时超过 23，将运算结果减去 24 后保存在时寄存器中。
- 操作数 S1、S2、D 各占用连续三个寄存器，请勿用作他用。

举例

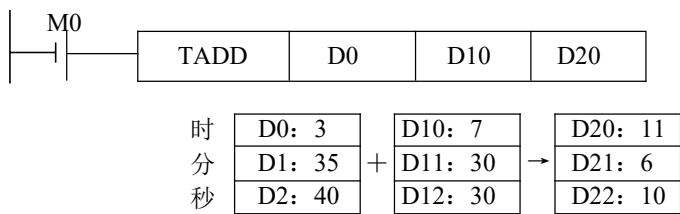
《一般情况》



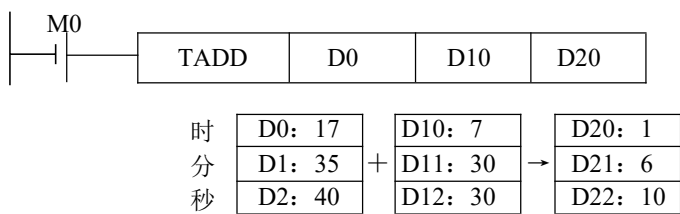
《秒超过 59 的情况》



《分超过 59 的情况》



《时超过 23 的情况》



## 4-10-6. 时钟数据减法[TSUB]

### 1. 指令概述

两个时钟数据（时、分、秒）做减法运算的指令。

时钟数据减法运算[TSUB]			
16 位指令	TSUB	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

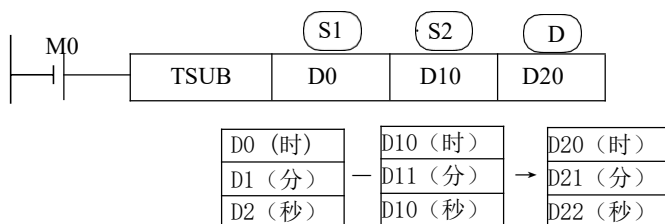
### 2. 操作数

操作数	作用	类型
S1	指定减法运算的时钟数据（时、分、秒）的软元件首地址编号	16 位, BIN
S2	指定减法运算的时钟数据（时、分、秒）的软元件首地址编号	16 位, BIN
D	指定存储减法运算结果的软元件地址编号	16 位, BIN

### 3. 适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		
	S2					√	√	√		
	D					√	√	√		

### 功能和动作



- M0 置ON 时，执行时钟减法运算，将（D0、D1、D2）的时钟数据（时、分、秒）与（D10、D11、D12）的时钟数据（时、分、秒）进行相减，并将运算结果（时、分、秒）存储到（D20、D21、D22）中。

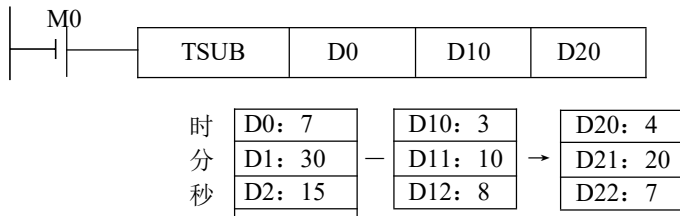
**注意：**寄存器的对应关系是固定的，即按照时、分、秒的顺序依次存放。

- 时的范围：0~23；分的范围：0~59，秒的范围：0~59。
- 若减法运算后的秒、分小于 0，将运算结果加上 60 后保存在秒、分寄存器中，同时分、小时的数值自动减 1；
- 操作数 S1、S2、D 各占用连续三个寄存器，请勿用作他用。

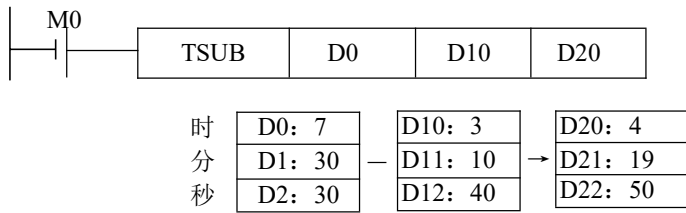


举例

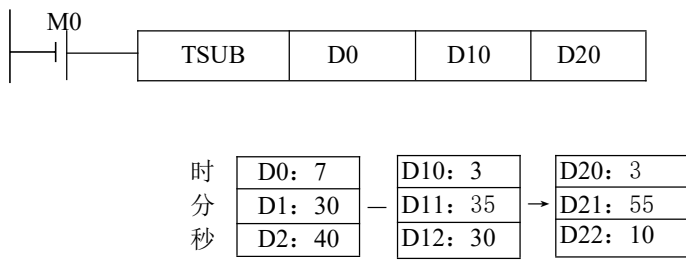
《一般情况》



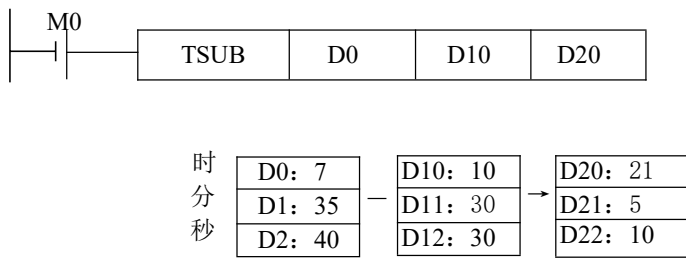
《秒小于 0 的情况》



《分小于 0 的情况》



《时小于 0 的情况》



---

# 5 高速计数

---

本章主要介绍 CZA 系列可编程控制器的高速计数功能，内容包括了高速计数的模式、接线方法、高速计数值的读写、复位等。

5-1. 功能概述

5-2. 高速计数模式

5-3. 高速计数值范围

5-4. 高速计数器输入端接线

5-5. 高速计数输入端口分配

5-6. 高速计数相关指令

5-7. 高速计数器与普通计数器的区别

5-8. 高速计数举例

## 5-1. 功能概述

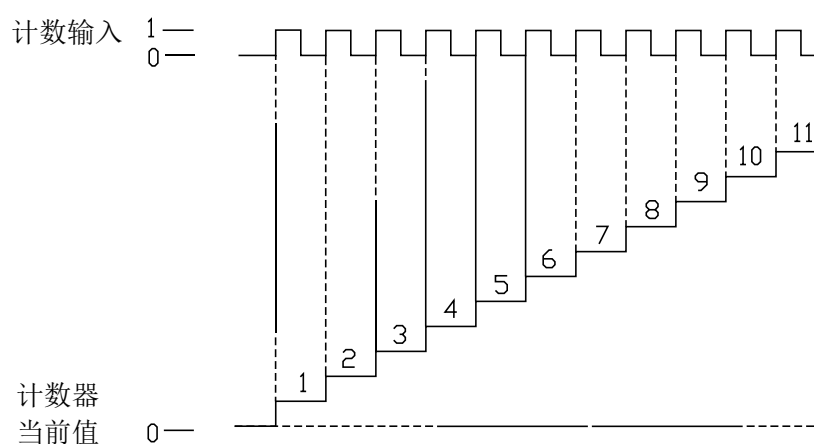
CZA 系列 PLC 具有与可编程控制器扫描周期无关的高速计数功能，通过选择不同计数器来实现针对测量传感器和旋转编码器等高速输入信号的测定，其最高测量频率可达 200 KHz。

## 5-2. 高速计数模式

CZA 系列高速计数功能暂时有两种计数模式，分别为单相单输入模式，双相双输入模式。

### 单相单输入模式

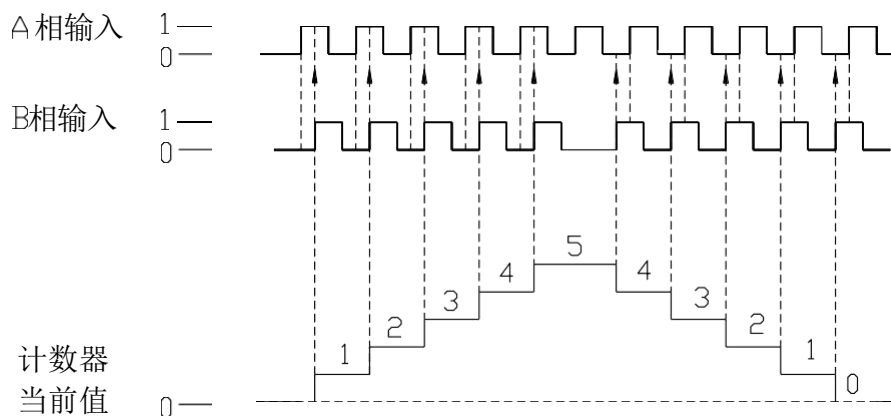
此模式下，计数输入脉冲信号，计数值随着每个脉冲信号的上升沿递增计数，在特定情况下同时也可进行减计数。



### 双相双输入模式

此模式下，高速计数值依照两种差分信号(A 相和 B 相)进行递增或递减计数，暂时只有一倍频模式。

- 一倍频模式



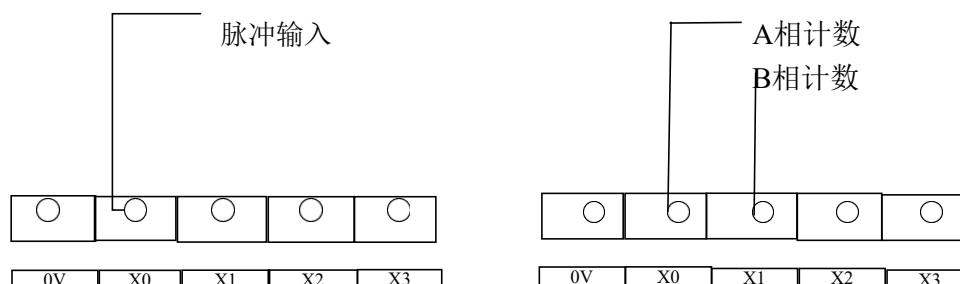
### 5-3. 高速计数值范围

高速计数器计数范围为： $K-2,147,483,648 \sim K+2,147,483,647$ 。当计数值超出此范围时，则产生上溢或下溢现象。

所谓产生上溢，就是计数值从  $K+2,147,483,647$  跳转为  $K-2,147,483,648$ ，并继续计数；而当产生下溢时，计数值从  $K-2,147,483,648$  跳转为  $K+2,147,483,647$ ，并继续计数。

### 5-4. 高速计数器输入端接线

对于计数脉冲输入端接线，依据可编程控制器型及计数器型号不同而稍加区别，其典型的几种输入端子接线方式。



### 5-5. 高速计数输入端口分配

各字母含义为：

U	U/D	A	B
计数脉冲输入	计数方向判断	A 相输入	B 相输入

在通常情况下，X0、X1、X2、X3 端子在单相和 AB 相模式下输入频率分别可达 200KHz；其它端子在单相和 AB 相模式下最高频率分别可达 60KHz。当 X 输入端不作为高速输入端口使用时，可作为普通输入端子使用。具体端口分配和功能如下表所示：

CZA				
高数计数器及使用端子	单相单输入模式		双相双输入模式	
	无启动/复位控制功能		C251	C253
	C235	C238		
X000	A(U/D)		A	
X001			B	
X002				B
X003		A(U/D)		A

注意：单相单输入模式U/D计数方向判断：

- 当使用端子X000时，若M8235为OFF时，C235为增计数；若M8235置ON时，C235位减计数；
- 当使用端子X003时，若M8238为OFF时，C238为增计数；若M8238为ON时，C238为减计数；

## 5-6. 高速计数相关指令

基本所有的高速计数器都支持高速计数输出指令[OUT]、比较置位指令[DHSCS]、比较复位指令[DHSCR]和复位指令[RST]。

### 5-6-1. 高速计数指令[OUT]

#### 1、指令概述

对某一个高速计数器进行高速计数的指令。

高速计数指令[OUT]			
16 位指令	-	32 位指令	OUT
执行条件	常开/闭、边沿触发	适用机型	CZA

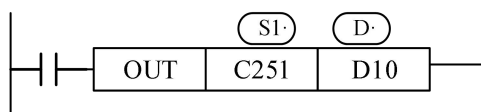
#### 2、操作数

操作数	作用	类型
S	指定高速计数器的编号	位, BIN
D	指定比较值的数值或软元件地址编号	32 位, 双字

#### 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	D					√	√	√		√
位软元件	操作数	系统								
		X	Y	M	S	C	T	C		
	S					√				

#### 功能和动作



- 当触发条件成立时，对高速计数器C235 进行高速计数。
- 比较值在D10（双字）中指定。
- 当计数值大于等于比较值时，C235 将被置 ON，但高速计数器仍将继续计数。
- 高速计数值可在 C235 中查看。
- 请勿将高速计数指令放在流程中使用。

### 5-6-2. 高速计数比较复位[HSCR]

#### 1、指令概述

将高速计数值到达设定值时执行的复位指令。

高速计数比较复位[DHSCR]			
16 位指令	-	32 位指令	DHSCR
执行条件	常开/闭、边沿触发	适用机型	CZA

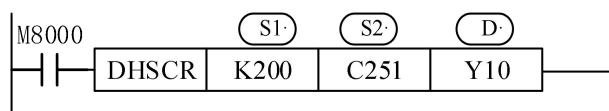
#### 2、操作数

操作数	作用	类型
S1	指定读取/写入的软元件地址编号	32 位, BIN
S2	指定高速计数器的编号	32 位, BIN
D	指定输出的位软元件	位

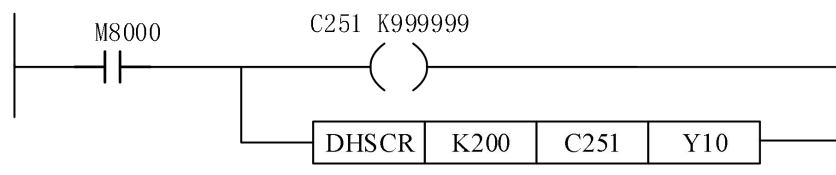
#### 3、适用软元件

字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	S1	√	√	√	√	√	√	√	√
S2						√			
位软元件	操作数	系统							
		X	Y	M	S	T	C		
	D		√	√	√				

功能和动作



- 由于比较，外部输出采用中断处理，C251的当前值变成199→200或201→200时，不受扫描周期影响，Y10立即复位。



### 5-6-3. 高速计数比较置位[DHSCS]

#### 1、指令概述

当高数计数值到达设定数值时执行置位的指令。

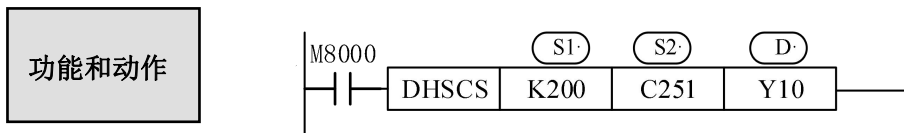
高速计数值写入[DHSCS]			
16 位指令	-	32 位指令	DHSCS
执行条件	常开/闭、边沿触发	适用机型	CZA

#### 2、操作数

操作数	作用	类型
S1	指定读取/写入的软元件地址编号	32 位, BIN
S2	指定高速计数器的编号	32 位, BIN
D	指定输出的位软元件	位

#### 3、适用软元件

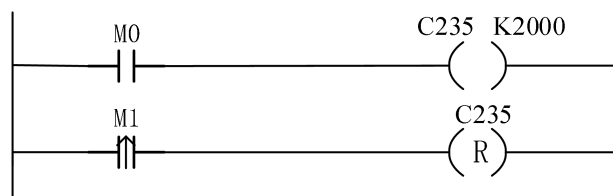
字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	S1	√	√	√	√	√	√	√	√
S2						√			
位软元件	操作数	系统							
		X	Y	M	S	T	C		
	D		√	√	√				



- 高数计数器是根据计数输入的OFF→ON的中断方式计数。计数器的当前值等于设定值时，计数器的输出节点立即工作。如下图上半部分程序那样，向外部输出与顺控有关，受扫描周期的影响。
- 如下图下半部分程序，能中断处理比较、外部输出，所以C255的当前值变为99→100或101→100时，Y10立即置位，不受扫描周期影响。

## 5-6-4. 高速计数复位模式[RST]

高速计数器的计数方式为软件复位方式。



如上图，当 M0 置 ON，C251 开始对 X0 端口的脉冲输入进行计数；当 M1 由 OFF 变为 ON 时，对 C251 的状态值进行复位，计数值清零。

## 5-7. 高速计数器与普通计数器的区别

高速计数器的驱动指令虽然与普通计数器的写法一样，均使用“OUT”，但其功能却大为不同：

普通计数器的导通条件“M0”，当 M0 由 OFF 变为 ON 一次，普通计数器的值加 1。而高速计数器计数时前面的导通条件必须处于常闭状态，此时相当于该高数计数器被启用，但是高数计数器的值并不改变，只有当相对应的外部信号输入端子接收到信号时，高数计数器才进行计数。如果外部信号输入端子有信号输入，而其触发条件没有闭合，则高数计数器也不会计数。

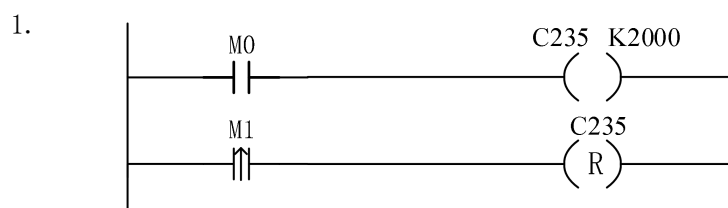
二者的区别如下表所示：

	指令格式	功能
普通计数器		对 M0 的 OFF→ON 的次数进行增计数，当计数值达到 2000 时，C0 置 ON
高速计数器		当 M0 导通时，对 X0 端口的脉冲输入进行增计数，当计数值达到 2000 时，C235 置 ON；且在高速计数过程中，M0 必须始终处于导通状态

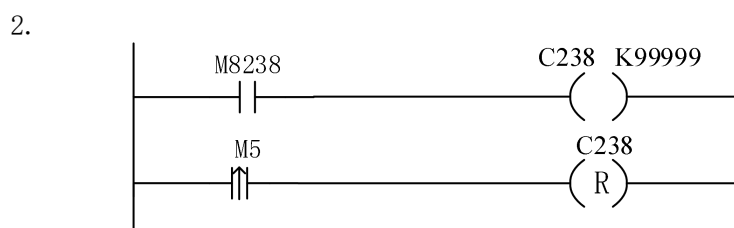


### 5-8. 高速计数举例

下面以 CZA 系列 32 点 PLC 为例介绍高速计数方式的编程方式及案例：



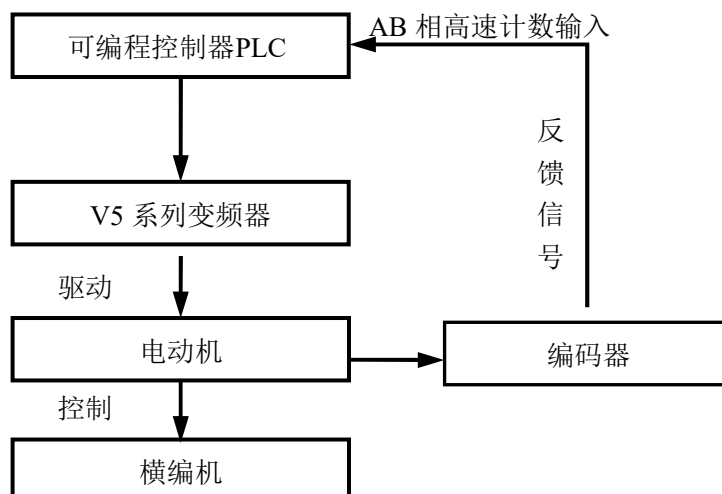
- C235 在 M0 为 ON 时，对输入 X0 的 OFF→ON 进行高速计数。
- 当 M1 上升沿来临时，将高速计数器 C235 复位。



- C238 在 M8238 为 ON 时，若输入 X3 的 OFF→ON 执行计数，则为减计数；C238 在 M8238 为 OFF 时，若输入 X3 的 OFF→ON 执行计数，则为增计数。
- 当 M5 上升沿来临时，将高速计数器 C238 复位。

#### 3. 横编机应用图例

系统原理如下图所示：通过可编程控制器 PLC 控制变频器相关端子，从而达到有效控制电动机的目的，同时经过编码器的反馈信号，对横编机进行有效的控制，即进行精确定位。



## 5-9. 输入输出刷新[REF]

## 1、指令概述

在运算过程中，若需要输入最新信息或者希望立即输出运算结果时，可以使用该指令。

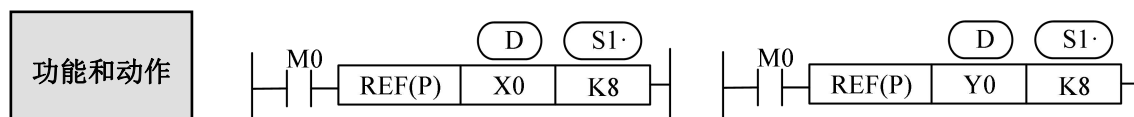
输入输出刷新[REF]			
16 位指令	REF(P)	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定输入输出刷新的点数	16 位, BIN
D	指定立即刷新的起始软元件地址编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1										√
位软元件	操作数	系统									
		X	Y	M	S	T	C				
	D	√	√								



## 1、输入刷新

在多个输入中，只刷新X0~X7的8点。如果在该指令执行前15ms(输入滤波应答时间)，置X0~X7为ON时，该指令执行时，输入映像存储区X0~X7为ON。

## 2、输出刷新

在多个输出中，Y0~Y7的8点被刷新，Y0~Y7中的任意一点为ON时，该指令执行时输出锁存存储区的该输出也为ON。

- 指定起始软元件号，如X0, X10..., Y0, Y10...这样，将最低位标号置为0。
- 刷新点数S1为8的倍数。
- 在FOR~NEXT指令之间，标号（新步号）~CJ（老步号）之间，一般需要REF指令。
- 在有输入输出动作的中断处理被执行中，在中断子程序中，进行输入输出刷新，获取最新的输入信息并且及时输出预算结果。
- 输出刷新中输出接点将在输出继电器应答时间后动作；  
继电器输出型的应答滞后时间小于15ms，晶体管输出型的输出应答滞后时间小于0.2ms。

# 6 脉冲输出

---

本章将主要介绍 CZA 系列可编程控制器的脉冲输出功能，内容包括脉冲输出指令用法及编程应用、输出端子的接线、注意事项以及相关线圈和寄存器等。

6-1. 功能概述

6-2. 脉冲输出的种类与指令应用

6-3. 输出端子接线

6-4. 注意事项

6-5. 示例说明

6-6. 脉冲输出相关线圈与寄存器

## 脉冲输出相关指令一览

指令助记符	功能	章节
脉冲输出		
PLSY	定频无加减速时间单向定量脉冲输出	6-2-1
PLSR	带加减速的定量脉冲输出	6-2-2
PWM	脉宽调制	6-2-3
ZRN	原点回归	6-2-4
DRVI	相对位置控制	6-2-5
DRVA	绝对位置控制	6-2-6
PLSRS	带S型加减速的定向脉冲输出	6-2-7
PLSRC	任意曲线定向脉冲输出	6-2-8
PLSF	带方向可变频脉冲输出	6-2-9

## 6-1. 功能概述及端口分布

CZA系列PLC 一般具有脉冲输出功能，且脉冲输出都为晶体管输出。通过使用不同的指令编程方式，可以进行无加速/减速的单向脉冲输出，也可以进行带S加速/减速的单向脉冲输出，还可以进行任意曲线、正反向输出等等，输出频率最高可达 250KHz。

CZA系列具有脉冲功能的PLC 脉冲分布情况如下表：

机型	I/O点	脉冲路数	输出点分布
CZA	16	2	Y0、Y1
~	24	2	Y0、Y1
~	30	2	Y0、Y1

PLC 可输出 200KHz~230KHz 的高速脉冲。

## 6-2. 脉冲输出的种类与指令应用

### 6-2-1. 定频无加减速单向定量脉冲输出[PLSY]

#### 1、指令概述

以指定的频率产生定量脉冲的指令。

无加减速时间变化的单向定量脉冲输出[PLSY]

16 位指令	PLSY	32 位指令	DPLSY
执行条件	常开/闭线圈触发	适用机型	CZA

#### 2、操作数

操作数	作用	类型
S1	指定频率的数值或软元件地址编号	16 位/32 位, BIN
S2	指定脉冲数目的数值或软元件的地址编号	16 位/32 位, BIN
D	指定脉冲输出端口编号	位

#### 3、适用软元件

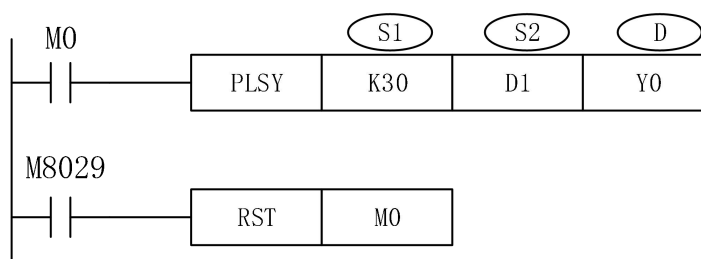
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
S1		√	√	√	√	√	√	√	√	√	
S2		√	√	√	√	√	√	√	√	√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
D			√				

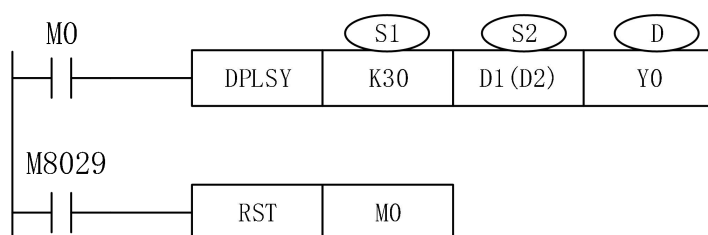
#### 功能和动作

《16 位指令形式》



- 频率范围：0~32767Hz；
- 产生脉冲量范围：K0~K32767；
- 脉冲输出：一般 2 轴为 Y0、Y1；
- 在 M0 为 ON 时，执行 PLSY 指令，Y0 输出频率为 30Hz 的脉冲，脉冲个数由 D1 指定。当输出脉冲个数达到设定值时，停止脉冲输出，此时线圈 M8029 置 ON 一次，并将 M0 复位，若脉冲发完时不复位 M0，则 M8029 将一直置 ON。

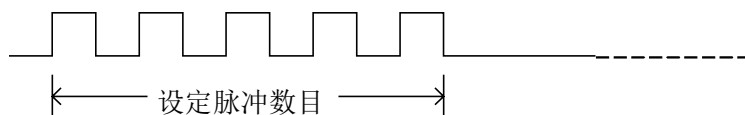
## 《32 位指令形式》



- 频率范围：0~250KHz；
- 产生脉冲量范围：K0~K2147483647；
- 脉冲输出：一般2轴为Y0、Y1；
- 在M0为ON时，执行DPLSY指令，Y0输出频率为30Hz的脉冲，脉冲个数由D2D1（双字）指定。当输出脉冲个数达到设定值时，停止脉冲输出，此时线圈M8029置ON一次，并将M0复位，若脉冲发完时不复位M0，则M8029将一直置ON。

输出模式

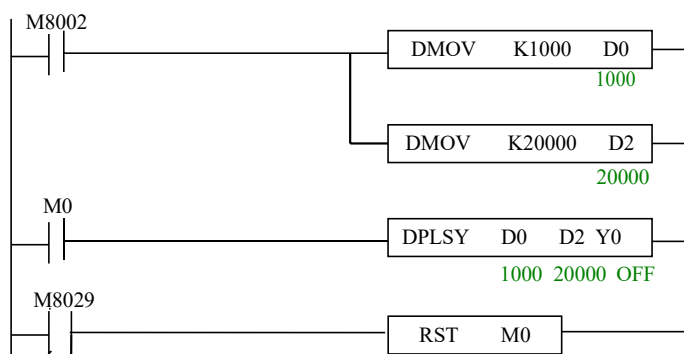
## 《连续的有限脉冲输出》



当输出完设定的脉冲数目之后，输出自动停止。

## 例

以指定的频率 1000Hz 无加减速时间变化的产生单向定量脉冲 20K。梯形图如下所示：



说明：D0 为脉冲频率，给 D0 赋值 1000；D2 为脉冲数目，给 D2 赋值 20000。

## 注意点

- 如果变频走速度，不需要走具体的位置时，只需把脉冲数赋值0，电机会一直以设定频率跑下去，而且频率随时可以更改。
- 脉冲输出口脉冲发完标志位：Y0对应M8029、Y1对应M8030。脉冲输出口脉冲累计计数寄存器：Y0对应D8340(低位)、D8341(高位)，Y1对应D8350(低位)、D8351(高位)，给个寄存器可以赋值0进行清零（脉冲发完或脉冲前置导通条件断开清零）。



## 6-2-2. 带加减速脉冲输出 [PLSR]

## 1、指令概述

以指定的频率和加减速时间产生定量脉冲的指令。

带加减速脉冲输出[PLSR]			
16 位指令	PLSR	32 位指令	DPLSR
执行条件	常开/闭线圈触发	适用机型	CZA

## 2、操作数

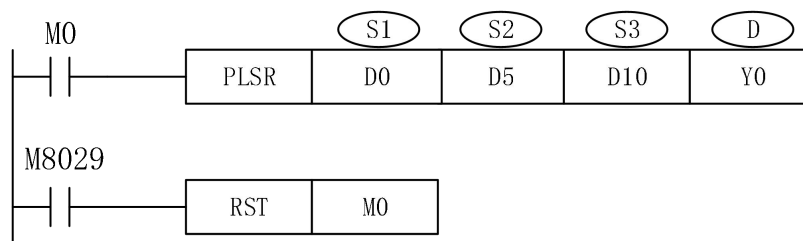
操作数	作用	类型
S1	指定分段脉冲参数起始软元件地址编号或脉冲频率	16 位/32 位, BIN
S2	指定加减速的数值或软元件的地址编号	16 位/32 位, BIN
S3	指定加减速时间数值或软元件地址编号	16 位, BIN
D	指定脉冲输出端口编号	位

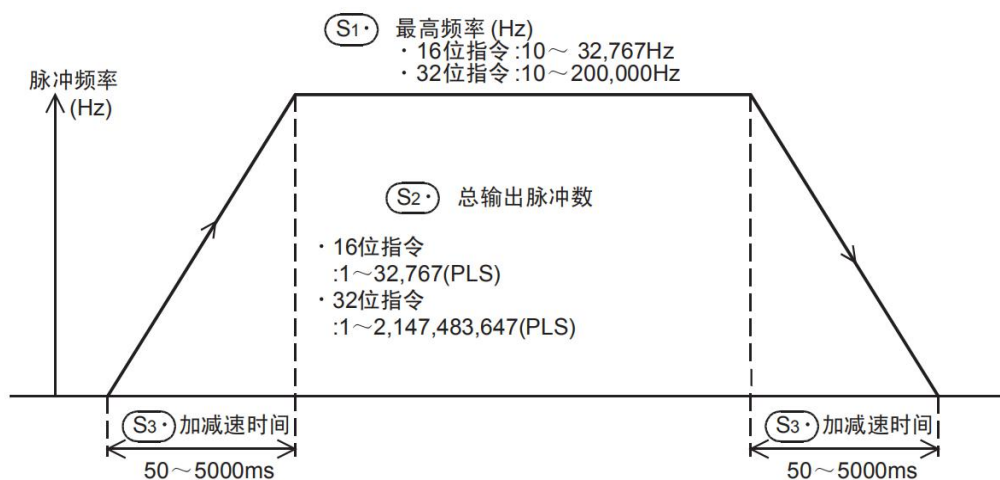
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1	√	√	√	√	√	√	√	√	√
	S2	√	√	√	√	√	√	√	√	√
S3	√	√	√	√	√	√	√	√	√	
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	D		√							

## 功能和动作

## 《16 位指令形式》





- 带加减速功能的定尺寸传送用的脉冲输出指令。
- 针对指定的最高频率，进行定加速，在即将达到所指定的输出脉冲数后，进行定减速。

- 各操作数的设定如下：

(S1) : 最高频率 (Hz)

- 可设定范围: 10-250KHz

-最高频率中指定值的1/10可作为减速时的一次变量。

(S2) : 脉冲数 (PLS)

-可设定范围: 16位运算, 100~32767 (PLS)

32位运算, 10~2147483647 (PLS)

-使用DPLSR指令时, 为此例中 (D5, D6) 作为32位设定值处理。

(S3) : 加减速时间 (ms)

-加速时间和减速时间以相同值动作, 设定要求如下:

①加速时间请在可编程控制器的扫描时间最大值 (D8012值以上) 的10倍以上, 指定不到10倍时, 加速时序不一定。

②作为加速时间可以设定的最小值公式如下:

$$(S3) \geq (90000 / (S1)) * 5$$

设定上述公式以下的值时, 加减速时间的误差增大, 此外, 设定不到9000/

(S1) 的值时, 对90000/(S1) 四舍五入运行。

④作为加减速时间可以设定的最大值的公式如下:

$$(S3) \leq ((S2) / (S1)) * 818$$

(D) : 脉冲输出口

- 一般指定为Y000或Y001。

- 输出一定是晶体输出, 除了作为脉冲输出也可作为普通IO口输出。

- 注意点：
1. 该指令的输出频率为10~250K Hz。最高速度、加减速时的变更速度超过此范围时，自动转换至范围内的数值后执行。
  2. 输出控制不受扫描周期影响进行中断处理。
  3. 在指令执行中即使改写操作数，运转也不反应，变更内容从下一次指令驱动开始有效。
  4. 设定的脉冲数输出完毕后，对应的发送完成特殊辅助寄存器置ON。如Y0输出出口的脉冲发完标志为M8029,Y1输出出口的脉冲发完标志为M8030。
  5. 自脉冲输出口每次发脉冲时脉冲数会存入特定的数据寄存器并进行脉冲计数，每次驱动时只会记录本次发完的脉冲数，下次再次驱动发脉冲时，脉冲数会清零重新计数。如Y0的脉冲累计特殊寄存器为D8340（低位）、D8341(高位)，Y1的脉冲累计特殊寄存器为D8350（低位）、D8351(高位)。
  6. 如在运行中需对累计寄存器清零，可以直接对特殊寄存器直接进行赋0值处理。如[(D)MOV K0 D8340]。
  7. 可编程控制器请使用晶体管输出。此外，为了进行高频脉冲输出，要按规定负载电流接入输出晶体管。
  8. 关于指令的使用限制
    - ①在编程过程中，同时使用2个PLSY指令或者2个PLSR指令能够在Y0和Y1输出端得到各自独立的脉冲输出。
    - ②在编程过程中，同时使用一个PLSY指令和1个PLSR指令能够在Y0和Y1输出端得到各自独立的脉冲输出。
    - ③在一个程序中，如果使用一个PLSY指令和1个PLSR指令同是输出的Y0，需用流程进行编写，否则程序运行会出错。

## 6-2-3. 脉宽调制[PWM]

## 1、指令概述

设定所需脉冲宽度和脉冲周期的输出指令。

脉宽调制[PWM]			
16 位指令	PWM	32 位指令	-
执行条件	边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定脉冲输出的宽度	16位字, BIN
S2	指定脉冲输出的周期	16位字, BIN
D	指定脉冲输出端子	位

## 3、适用软元件

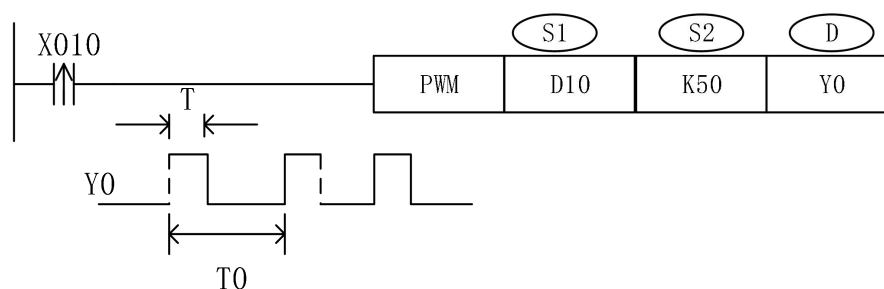
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
S1		√	√	√	√	√	√	√	√	√	
S2		√	√	√	√	√	√	√	√	√	

位软元件	操作数	系统					
		X	Y	M	S	T	C
D			√				

## 功能和动作

## 《16 位指令形式》



(S1): 指定脉冲宽度 $t=0\sim 32767\text{ms}$ 。

(S2): 指定周期 $T_0=1\sim 32767\text{ms}$ , 但 (S1)  $\leq$  (S2) 。

(D): 指定脉冲输出端口。

如上图, 当常开触点X10闭合时, PWM指令执行, 让Y000端子输出脉冲宽度为[S1]D10、周期为[S2]50的脉冲信号。

注意点: 该输出的ON/OFF可进行中断处理执行

### 6-2-4. 原点回归[ZRN]

#### 1、指令概述

使简单快速回到原点的指令。

原点回归[ZRN]			
16 位指令	ZRN	32 位指令	DZRN
执行条件	常开/闭线圈触发	适用机型	CZA

#### 2、操作数

操作数	作用	类型
S1	指定原点回归速度的数值或软元件地址编号	16/32 位, BIN
S2	指定爬行速度的数值或软元件地址编号	16/32 位, BIN
S3	指定近点信号的位软元件地址编号	位
D	指定脉冲输出端口的编号	位

#### 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1	√	√	√	√	√	√	√	√	√	
	S2	√	√	√	√	√	√	√	√	√	
位软元件	操作数	系统									
		X	Y	M	S	T	C				
	S3	√	√	√	√						
	D		√								

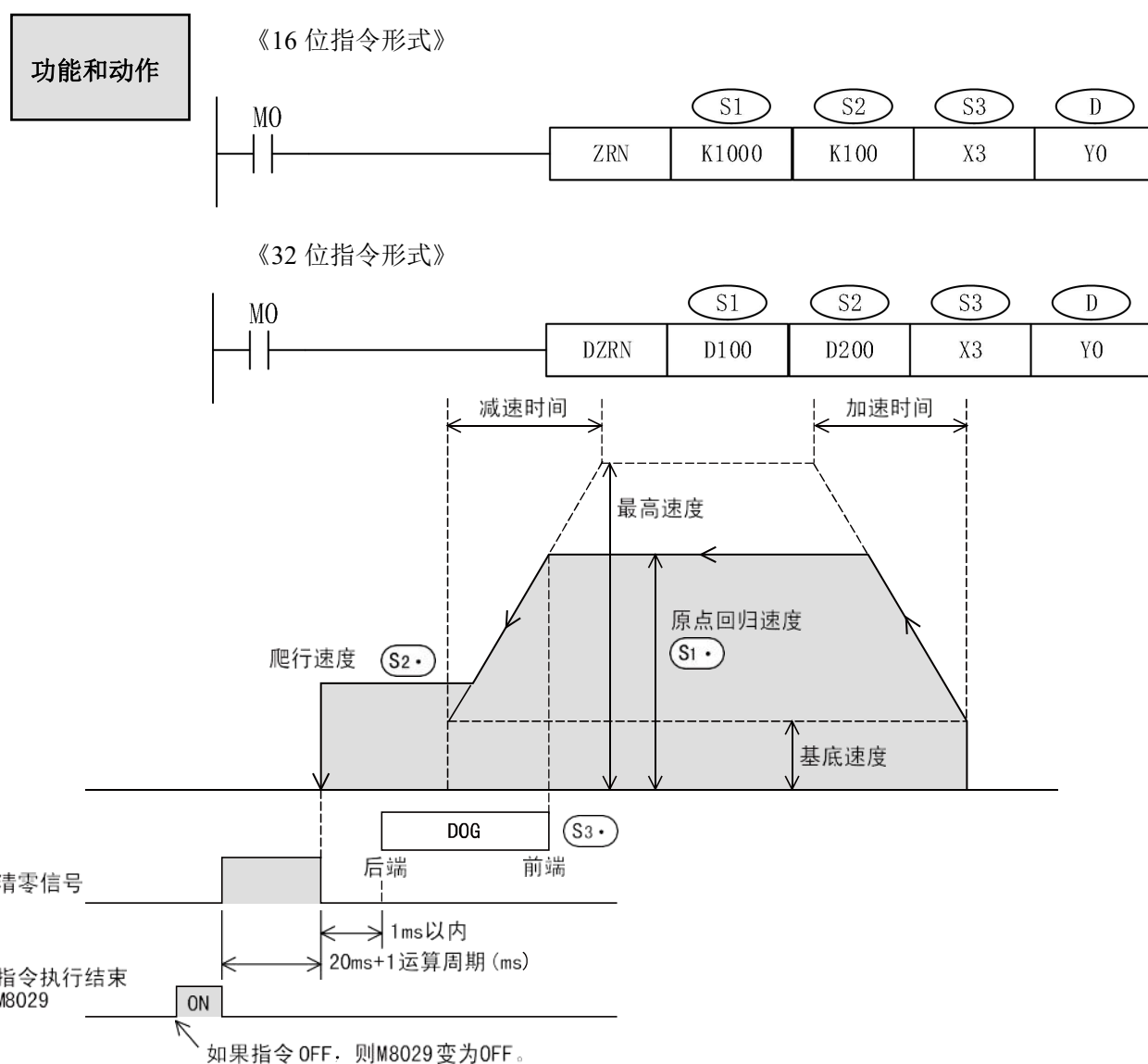
#### 相关软元件一览

##### 1. 特殊辅助继电器

软元件编号		名称	属性
Y0	Y1		
M8029	M8030	指令执行结束标志位	读出专用
M8340	MD8350	脉冲输出中监控 (BUSY/READY)	读出专用
M8348	M8358	定位指令驱动中	读出专用
M8349	M8359	脉冲停止指令	可驱动

## 2. 特殊数据寄存器

软元件编号				名称	数据长	初始值
Y0		Y1				
D8340	低位	D8350	低位	当前值寄存器[PLS]	32位	0
D8341	高位	D8351	高位			
D8342		D8352		基底速度[Hz]	16位	0
D8343	低位	D8353	低位	最高速度[Hz]	32位	100,000
D8344	高位	D8354	高位			
D8345		D8355		爬行速度[Hz]	16位	1000
D8346	低位	D8356	低位	原点回归速度 [Hz]	32位	50000
D8347	高位	D8357	高位			
D8348		D8358		加速时间[ms]	16位	100
D8349		D8359		减速时间 [ms]	16位	100



1) 在 (S1) 中指定原点回归速度。

如果原点回归速度设定得超过最高速度，则按照最高速度动作。

	设定范围
16位运算	10~32,767 (Hz)
32位运算	10~200,000 (Hz)

此外，下表中指定的原点回归速度不适用。

脉冲输出端软元件	原点回归速度
(D) =Y0	D8347, D8346
(D) =Y1	D8357, D8356

2) 在 (S2) 中指定爬行速度。

设定范围 10~32,767 (Hz)

3) 在 (S3) 中指定输入近点信号 (DOG) 的软元件编号。(无触点输入)

近点信号从OFF到ON时，开始减速到爬行速度；一旦从ON变为OFF，则结束原点回归。

①如果将基本单元的X000~X007指定为输入信号，那么通过中断处理停止。(输出清零信号时，请将清零信号输出有效标志位置ON。)

在下列条件时，受到输入常数以及顺控程序的扫描周期的影响，原点位置的偏差可能变大。

指定了X010以后的输入编号或者辅助继电器等其它的软元件时。

②指定X010以后的输入继电器为近点信号时，适用输入滤波器(10ms)。

4) 在 (S3) 中指定输出脉冲的输出编号Y000~Y001。

5) 原点回归方向

在该指令中，原点回归方向为反转方向。(在原点回归过程中，当前值寄存器的数值向减少的方向动作。)

此外，向正转方向进行原点回归时，请按照如下顺序，用程序对作为「旋转方向信号」接线的输出继电器(Y)进行控制。

a) 将Y□□□(旋转方向信号)置位(ON)。

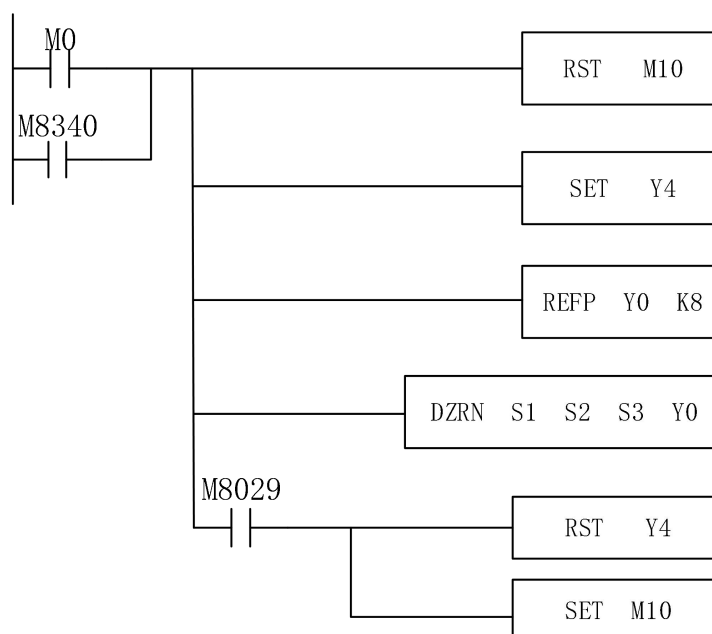
b) 用REF指令对Y□□□做输出刷新。

c) 执行原点回归(ZRN)指令。

d) 用原点回归(ZRN)指令的执行结束标志位(M8029)，复位(OFF)Y□□□(旋转方向信号)。

## 程序举例

程序举例为Y000用的旋转方向信号为Y004时的例子。



## 6) 清零信号的输出

该指令具有在零点位置停止后，输出清零信号的功能。

需要在原点回归动作中输出清零信号的时候，请将清零信号输出功能有效标志位（下表）置为ON 状态。

## a) 不使用清零信号软元件指定功能时

脉冲输出端软元件	清零信号输出有效标志位的状态
(D) = Y000	M8341=ON
(D) = Y001	M8351=ON

b) 使用清零信号软元件指定功能时, 如果清零信号软元件指定功能有效标志位为ON, 则可以用清零信号软元件指定用软元件来指定与软元件相对应的脉冲输出端软元件的清零信号(输出Y)。

脉冲输出端软元件	清零信号输出有效标志位的状态
(D) = Y000	M8341=ON
(D) = Y001	M8351=ON

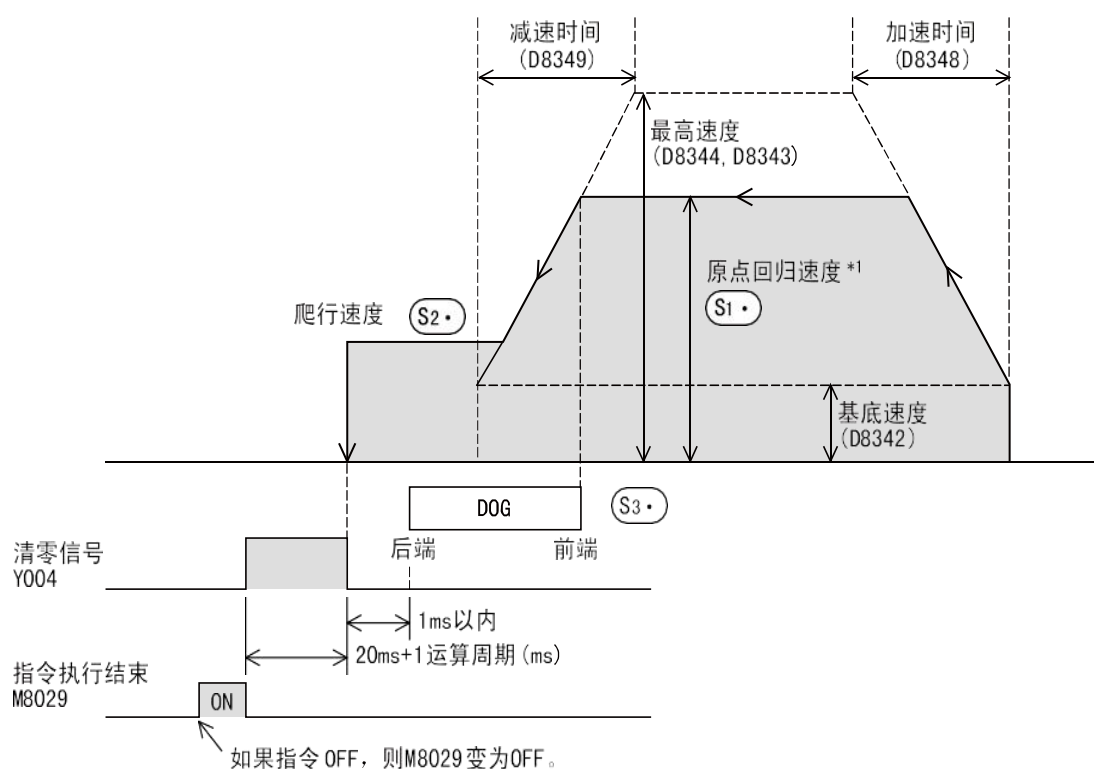
## 1. 原点回归动作

以脉冲输出端指定(D)为Y000为例，说明原点回归动作。如果使用Y000、Y001、Y002时，则请根据使用的输出编号，改读各相关标志位(特殊辅助继电器、特殊数据寄存器)

## 1) 执行原点回归用ZRN指令。



- 2) 以(S1)指定的原点回归速度运动。
- 3) 一旦(S3)指定的近点信号(DOG)为ON, 就开始减速, 知道减速到(S2)指定的爬行速度为止。
- 4) (S3)指定的近点信号(DOG)从ON到OFF后, 则立即停止脉冲的输出。
- 5) 清零信号输出功能(M8341)有效(ON)时, 在脉冲输出停止后1ms以内, 清零信号(Y004)在「20ms+1个运算周期(ms)」的时间内保持为ON。
- 6) 当前值寄存器(D8341, D8340)变为“0”(清零)。
- 7) 指令执行结束标志位为ON, 结束原点回归动作。



### 6-2-5. 相对位置单段脉冲控制[DRVI]

#### 1、指令概述

这是采用相对驱动的单速定位指令。用带正/负号的方式指定距离当前位置的移动距离，叫做增量(相对)驱动方式。

相对位置单段脉冲控制[DRVI]			
16 位指令	DRVI	32 位指令	DDRVI
执行条件	常开/闭线圈触发	适用机型	CZA

#### 2、操作数

操作数	作用	类型
S1	指定输出脉冲数的数值或软元件地址编号	16/32 位, BIN
S2	指定输出脉冲频率的数值或软元件地址编号	16/32 位, BIN
D1	指定脉冲输出端口的编号	位
D2	指定脉冲输出方向端口的编号	位

#### 3、适用软元件

字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	S1	√	√	√	√	√	√	√	√
S2	√	√	√	√	√	√	√	√	√
位软元件	操作数	系统							
		X	Y	M	S	T	C		
	D1		√						
	D2	√	√	√	√				

#### 相关软元件一览

##### 1. 特殊辅助继电器

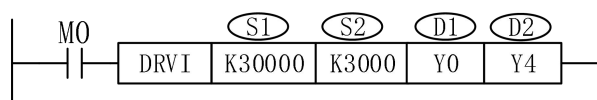
软元件编号		名称	属性
Y0	Y1		
M8029	M8030	指令执行结束标志位	读出专用
M8340	M8350	脉冲输出中监控 (BUSY/READY)	读出专用
M8348	M8358	定位指令驱动中	读出专用
M8349	M8359	脉冲停止指令	可驱动

## 2. 特殊数据寄存器

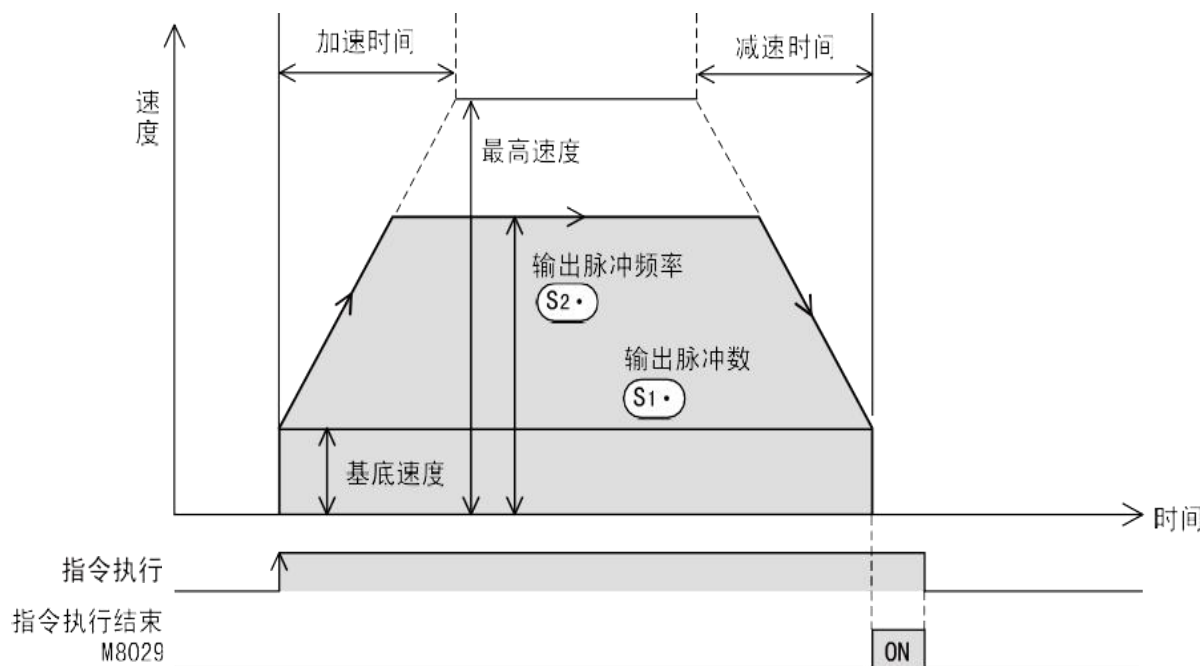
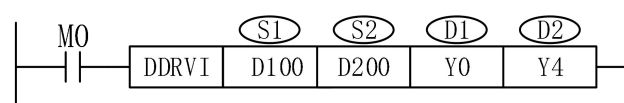
软元件编号				名称	数据长	初始值
Y0		Y1				
D8340	低位	DD835	低位	当前值寄存器[PLS]	32位	0
		0				
D8341	高位	D8351	高位			
D8342		D8352		基底速度[Hz]	16位	0
D8343	低位	D8353	低位	最高速度[Hz]	32位	100,000
D8344	高位	D8354	高位			
D8348		D8358		加速时间[ms]	16位	100
D8349		D8359		减速时间 [ms]	16位	100
D8464		D8465		清零信号软元件指定	16位	-

## 功能和动作

## 《16位指令形式》



## 《32位指令形式》



1) 在 (S1) 指定输出脉冲数 (相对地址值)

	设定范围
16位运算	-32,768~+32,767
32位运算	-999,999~+999,999

2) 在 (S2) 指定输出脉冲频率。

	设定范围
16位运算	10~32,767 (Hz)
32位运算	10~200,000 (Hz)

3) 在 (D1) 指定输出脉冲的输出编号Y0~Y1。

4) 在 (D2) 指定输出旋转方向信号的软元件编号。

使用内置的晶体管输出作为脉冲输出端时, 旋转方向信号请使用晶体管输出。

脉冲输出端软元件	旋转方向输出
(D1) =Y000	(D2) =Y004
(D1) =Y001	(D2) =Y005

旋转方向和指定软元件的ON/OFF状态如下表所示。

但是, 在该指令执行过程中, 请用户不要对 (D2) 指定的输出进行控制。

(D2) 中指定的软元件的ON/OFF状态	旋转方向 (当前值的增减)
ON	(S1) 中指定的脉冲输出值为正数时 正转 [(D1) 的脉冲输出使当前值增加]
OFF	(S1) 中指定的脉冲输出值为负数时 反转 [(D1) 的脉冲输出使当前值减少]

### 注意要点

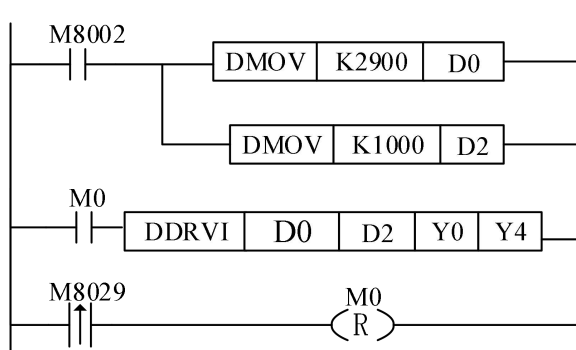
- 1) 在指令执行过程中, 即使改变操作数的内容, 也不反映到当前的运行中。在下次的指令驱动时才有效。
- 2) 在指令执行过程中, 如Y0驱动触点为OFF时, 立即停止。且此时指令执行结束标志位M8029不动作。
- 3) 脉冲输出中监控(BUSY / READY)为ON时, 使用该输出的定位用指令(包括PLSR、PLSY)不能执行。此外, 即使指令驱动触点为OFF, 在脉冲输出中监控(BUSY/READY)为ON期间, 也请不要执行指定了同一输出编号的定位指令(包括PLSR/PLSY)。

例

X 轴的当前坐标为 (100, 0)，现需要以 1000Hz 的速度移动到目标位置 (3000, 0)，脉冲输出端子为 Y0 方向控制端子为 Y4；由于目标位置相对于当前位置的距离是  $2900=3000-100$ ，所以通过相对位置单段脉冲控制指令 DRVI 执行示意图如下：



梯形图如下：



执行一次 M0，监控 D8340(双字)可看到，D8340（双字）的数值在原来的基础增加 K2900。

## 6-2-6. 绝对位置单段脉冲控制[DRVA]

### 1、指令概述

这是采用绝对驱动的单速定位指令。采用从原点(0点)开始的距离指定方式，也被称为绝对驱动方式。

绝对位置单段脉冲控制[DRVA]			
16 位指令	DRVA	32 位指令	DDRVA
执行条件	常开/闭线圈触发	适用机型	CZA

### 2、操作数

操作数	作用	类型
S1	指定输出脉冲数的数值或软元件地址编号	16/32 位, BIN
S2	指定输出脉冲频率的数值或软元件地址编号	16/32 位, BIN
D1	指定脉冲输出端口的编号	位
D2	指定脉冲输出方向端口的编号	位

### 3、适用软元件

字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、Z
	S1	√	√	√	√	√	√	√	√
S2	√	√	√	√	√	√	√	√	√
位软元件	操作数	系统							
		X	Y	M	S	T	C		
	D1		√						
	D2	√	√	√	√				

### 相关软元件一览

#### 1. 特殊辅助继电器

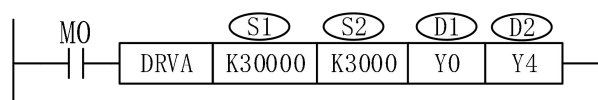
软元件编号		名称	属性
Y0	Y1		
M8029	M8030	指令执行结束标志位	读出专用
M8340	MD8350	脉冲输出中监控 (BUSY/READY)	读出专用
M8348	M8358	定位指令驱动中	读出专用
M8349	M8359	脉冲停止指令	可驱动

## 2. 特殊数据寄存器

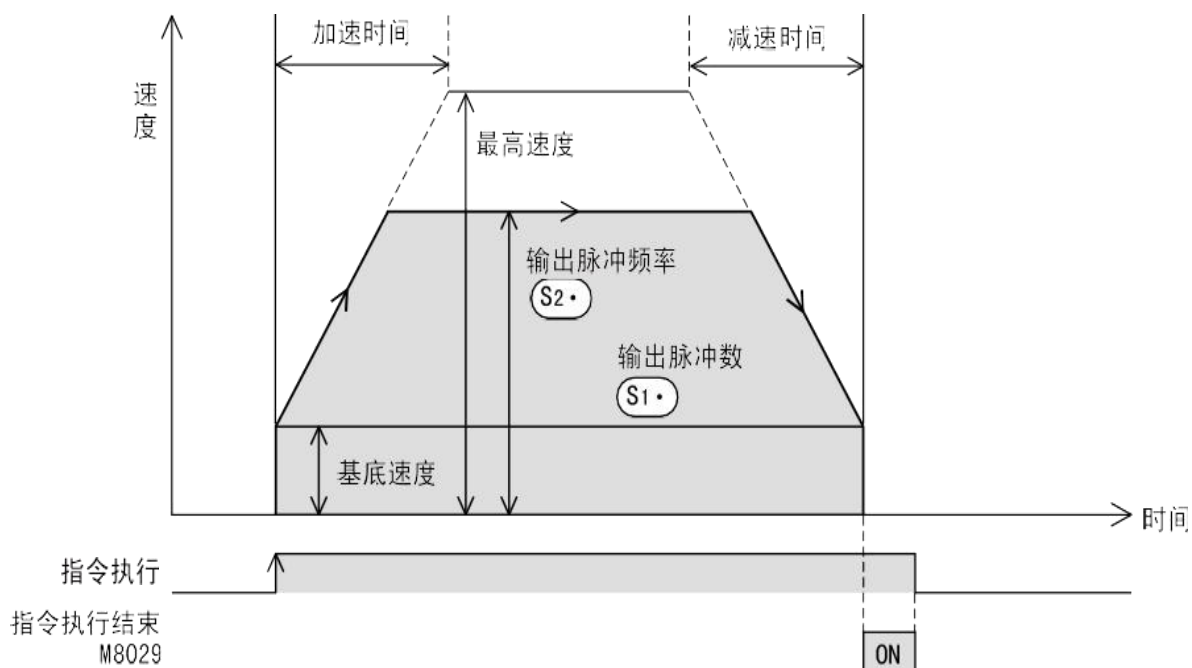
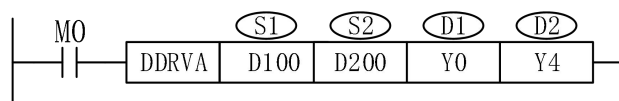
软元件编号				名称	数据长	初始值
Y0		Y1				
D8340	低位	DD835	低位	当前值寄存器[PLS]	32位	0
		0				
D8341	高位	D8351	高位			
D8342		D8352		基底速度[Hz]	16位	0
D8343	低位	D8353	低位	最高速度[Hz]	32位	100,000
D8344	高位	D8354	高位			
D8348		D8358		加速时间[ms]	16位	100
D8349		D8359		减速时间[ms]	16位	100
D8464		D8465		清零信号软元件指定	16位	-

## 功能和动作

## 《16位指令形式》



## 《32位指令形式》



1) 在 (S1) 指定输出脉冲数 (绝对地址值)

	设定范围
16位运算	-32,768~+32,767
32位运算	-999,999~+999,999

2) 在 (S2) 指定输出脉冲频率。

	设定范围
16位运算	10~32,767 (Hz)
32位运算	10~200,000 (Hz)

3) 在 (D1) 指定输出脉冲的输出编号Y0~Y1。

4) 在 (D2) 指定输出旋转方向信号的软元件编号。

使用内置的晶体管输出作为脉冲输出端时, 旋转方向信号请使用晶体管输出。

脉冲输出端软元件	旋转方向输出
(D1) =Y000	(D2) =Y004
(D1) =Y001	(D2) =Y005

旋转方向和指定软元件的ON/OFF状态如下表所示。

但是, 在该指令执行过程中, 请用户不要对 (D2) 指定的输出进行控制。

(D2) 中指定的软元件的ON/OFF状态	旋转方向 (当前值的增减)
ON	(S1) 中指定的脉冲输出值为正数时 正转 [(D1) 的脉冲输出使当前值增加]
OFF	(S1) 中指定的脉冲输出值为负数时 反转 [(D1) 的脉冲输出使当前值减少]

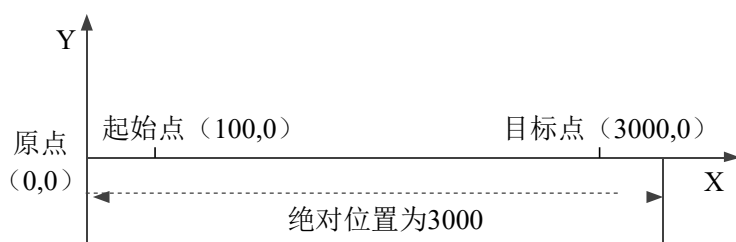
#### 注意要点

- 1) 在指令执行过程中, 即使改变操作数的内容, 也不反映到当前的运行中。在下次的指令驱动时才有效。
- 2) 在指令执行过程中, 如Y0驱动触点为OFF时, 立即停止。且此时指令执行结束标志位M8029不动作。
- 3) 脉冲输出中监控(BUSY / READY)为ON时, 使用该输出的定位用指令(包括PLSR、PLSY)不能执行。此外, 即使指令驱动触点为OFF, 在脉冲输出中监控(BUSY/READY)为ON期间, 也请不要执行指定了同一输出编号的定位指令(包括PLSR/PLSY)。

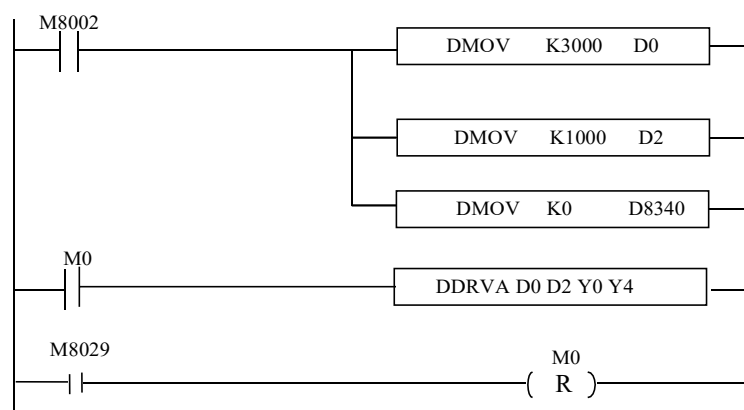


例

X 轴的当前坐标为 (100, 0)，现需要以 1000Hz 的速度移动到目标位置 (3000, 0)，脉冲输出端子为 Y0 方向控制端子为 Y4；由于目标位置相对于原点位置的距离是 3000，所以通过绝对位置单段脉冲控制指令 DRVA 执行示意图如下：



梯形图如下：



执行一次 M0，监控 D8340（双字）可看到，D8340（双字）的数值变为 K3000，所以在执行该指令之前，要先把 D8340（双字）清零，即找原点。

## 6-2-7. 带S型加减速的定向脉冲输出[PLSRS]

## 1、指令概述

以指定的脉冲数，最高频率、加减速时间以及S型加减速时间分段产生位置脉冲的指令。

绝对位置多段脉冲控制[PLSRS]			
16 位指令	PLSRS	32 位指令	DPLSRS
执行条件	常开/闭线圈触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定输出脉冲频率的数值或软元件地址编号	16/32 位, BIN
S2	指定输出脉冲数的数值或软元件地址编号	16/32 位, BIN
S3	指定加减速时间数值或软元件地址编号	16位, BIN
S4	指定S型加减速时间数值或软元件地址编号	16位, BIN
D	指定脉冲输出端口编号	位

## 3、适用软元件

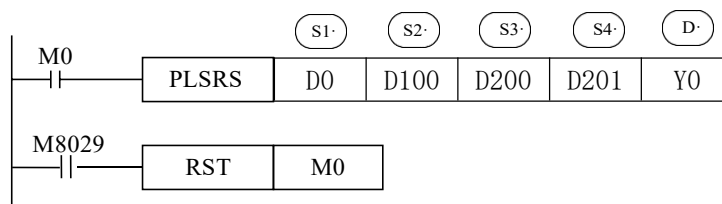
字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1	√	√	√	√	√	√	√	√	√	
	S2	√	√	√	√	√	√	√	√	√	
	S3	√	√	√	√	√	√	√	√	√	
	S4	√	√	√	√	√	√	√	√	√	

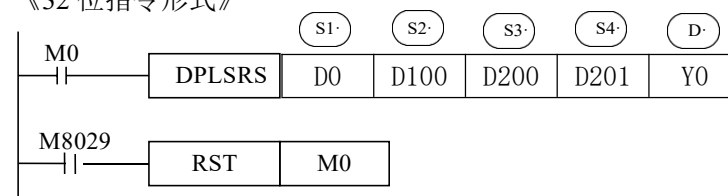
位软元件	操作数	系统					
		X	Y	M	S	T	C
	D1		√				

## 功能和动作

## 《16 位指令形式》



## 《32 位指令形式》



- 带加减速功能的定尺寸传送用的脉冲输出指令。
- 针对指定的最高频率，进行定加速，在即将达到所指定的输出脉冲数后，进行定减速。

- 各操作数的设定如下：

(S1)：最高频率 (Hz)

- 可设定范围：10-250KHz

-最高频率中指定值的1/10可作为减速时的一次变量。

(S2)：脉冲数 (PLS)

-可设定范围：16位运算，100~32767 (PLS)

32位运算，10~2147483647 (PLS)

-使用DPLSR指令时，为此例中 (D5, D6) 作为32位设定值处理。

(S3)：加减速时间 (ms)

-加速时间和减速时间以相同值动作，设定要求如下：

①加速时间请在可编程控制器的扫描时间最大值 (D8012值以上) 的10倍以上，指定不到10倍时，加速时序不一定。

②作为加速时间可以设定的最小值公式如下：

$$(S3) \geq (90000 / (S1)) * 5$$

设定上述公式以下的值时，加减速时间的误差增大，此外，设定不到9000/

(S1) 的值时，对90000/(S1) 四舍五入运行。

④作为加减速时间可以设定的最大值的公式如下：

$$(S3) \leq ((S2) / (S1)) * 818$$

(S4)：S型加减速时间 (ms)

-S型加减速时间是在加减速时间内，让运动设备在加减速可以更加平缓的参数。理论上S型加减速时间一般设定公式如下：

$$0 \leq (S4) \leq ((S3) / 2)$$

如 (S4) 设定时间为0ms, 则默认为没有S型加减速，如 (S4)  $\geq$  ((S3)/2)，则S型加减速时间默认为 ((S3)/2)。

(D)：脉冲输出口

- 一般指定为Y000或Y001。
- 输出一定是晶体输出，除了作为脉冲输出也可作为普通I/O口输出。

- 注意点：1. 该指令的输出频率为10~250KHz。最高速度、加减速时的变更速度超过此范围时，不能正常运行。
2. 输出控制不受扫描周期影响进行中断处理。
3. 在指令执行中即使改写操作数，运转也不反应，变更内容从下一次指令驱动开始有效。
4. 设定的脉冲数输出完毕后，对应的发送完成特殊辅助寄存器置ON。如Y0输出端的脉冲发完标志为M8029, Y1输出端的脉冲发完标志为M8030。
5. 自脉冲输出口每次发脉冲时脉冲数会存入特定的数据寄存器并进行脉冲计数，每次驱动时只会记录本次发完的脉冲数，下次再次驱动发脉冲时继上次一的脉冲继续计数，不会被清零。如Y0的脉冲累计特殊寄存器为D8340（低位）、D8341(高位)，Y1的脉冲累计特殊寄存器为D8350（低位）、D8351(高位)。
6. 如在运行中需对累计寄存器清零，可以直接对特殊寄存器直接进行赋0值处理。如[DMOV K0 D8340]。
7. 可编程控制器请使用晶体管输出。此外，为了进行高频脉冲输出，要按规定的负载电流接入输出晶体管。
8. 关于指令的使用限制
- ①在编程过程中，同时使用2个PLSY指令或者2个PLSR指令能够在Y0和Y1输出端得到各自独立的脉冲输出。
- ②在编程过程中，同时使用1个PLSY指令和1个PLSR指令能够在Y0和Y1输出端得到各自独立的脉冲输出。
- ③在一个程序中，如果使用1个PLSY指令和1个PLSR指令同是输出的Y0，需用流程进行编写，否则程序运行会出错。

## 6-2-8. 任意曲线脉冲输出[PLSRC]

## 1、指令概述

以指定若干段的脉冲数，最高频率分段产生位置脉冲的指令。

绝对位置多段脉冲控制[PLSRC]			
16 位指令	PLSRC	32 位指令	DPLSRC
执行条件	常开/闭线圈触发	适用机型	CZA

## 2、操作数

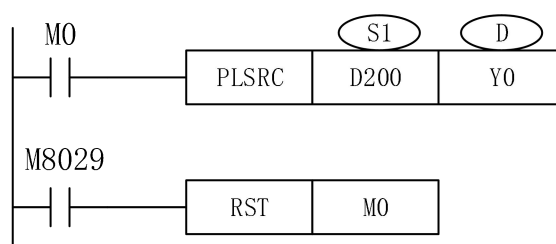
操作数	作用	类型
S1	指定的起始地址寄存器	16/32 位, BIN
D	指定脉冲输出端口编号	位

## 3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S						√				
位软元件	操作数	系统									
		X	Y	M	S	T	C				
	D		√								

## 功能和动作

## 《16 位指令形式》



- 参数地址是以Dn 为起始地址的一段区域。上例（16 位指令形式）：D200设定第 1段脉冲的最高频率、D202设定第 1 段脉冲的个数，D204 设定第 2 段脉冲的最高频率、D206 设定第 2 段脉冲的个数，..... 如果某一段脉冲频率或者脉冲数中存在一个O,则跳过该段执行下一段，最多可设定 88 段。注：占用384个寄存器+16个特殊用寄存器（暂不对用户开放）
- 对 32 位指令 DPLSRC，D200(D201)（双字）设定第 1 段脉冲的最高频率D202(D203)（双字）设定第 1 段脉冲的个数，D204(D205)（双字）设定第 2 段脉冲的最高频率、D206(D207)（双字）设定第 2 段脉冲的个数.....如果某一段脉冲频率或者脉冲数中存在

一个0则跳过该段执行下一段，最多可设定88段。

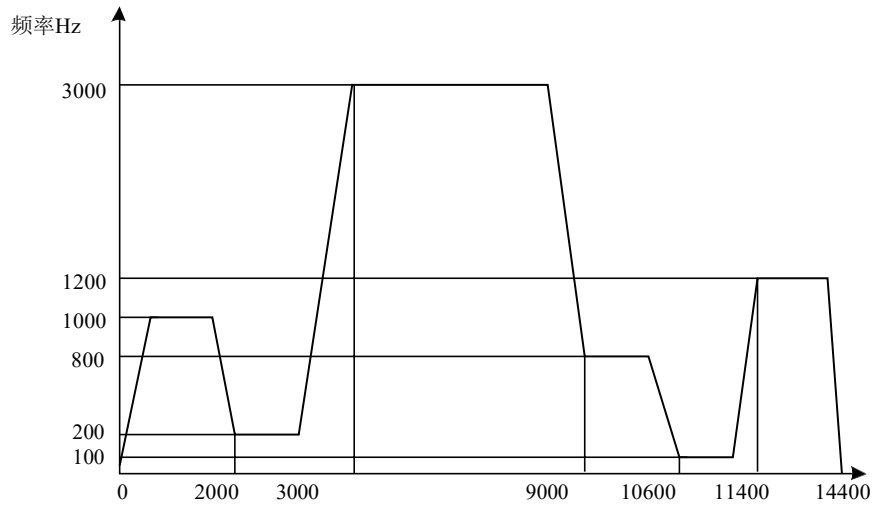
- 加减速时间是指从开始到第一段最高频率的加速时间，同时也定义了所有段的频率与时间的斜率，从而后面的加减速都按照这个斜率来加速/减速。
- 脉冲输出：一般 2 轴为 Y0、Y1；
- 频率范围：0~32767Hz（16 位指令）、0~100KHz（32 位指令）。
- 脉冲个数范围：K0~K32,767（16 位指令）、K0~K2,147,483,647（32 位指令）。
- 执行指令前请确认当前位置对应的寄存器里面的值是否正确（D8341、D8340[Y0]/D8350、D8351[Y1] ……）。

**注意：**参数首地址Dn由用户自行定义，如用户设置了a段曲线，则以Dn为首往后连续的4a寄存器则被自动占用，同时自被占用的最后一个地址再往后16个地址则也被同时占用，这16个地址关联着用户所设的这条曲线的详细说明，故用户在使用时尽量避免寄存器的重复使用。

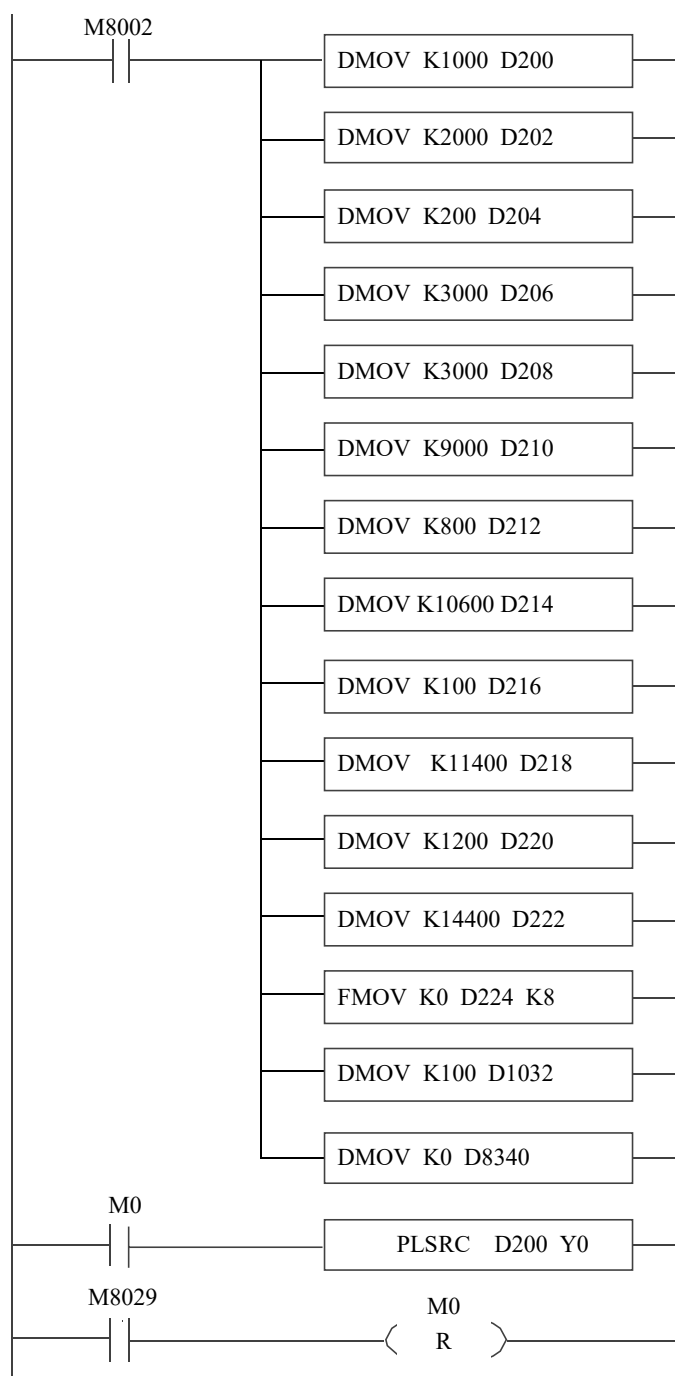
**例**

现需要通过任意曲线多段脉冲控制 DPLSC 发送 6 段脉冲，脉冲端子为Y0，每段的脉冲频率、相对于原点的位置与加减速时间如下表所示：

名称	频率设定值(Hz)	位置设定值
第 1 段脉冲	1000	2000
第 2 段脉冲	200	3000
第 3 段脉冲	3000	9000
第 4 段脉冲	800	10600
第 5 段脉冲	100	11400
第 6 段脉冲	1200	14400



梯形图如下所示：



执行一次 M0，监控 D8340（双字）可看到，D8340（双字）的数值最终变为K14400，所以在执行该指令之前，要先将 D8340（双字）清零，即找原点。



## 6-2-9. 带方向可变频率脉冲输出[PLSF]

## 1、指令概述

以可变频率的形式产生连续脉冲的指令。

带方向可变频率连续脉冲输出[PLSF]			
16 位指令	PLSF	32 位指令	DPLSF
执行条件	常开/闭线圈触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

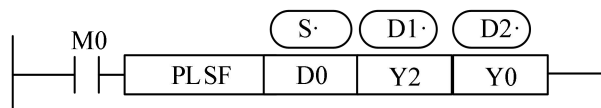
操作数	作用	类型
S	指定频率的数值或软元件地址编号	16 位/32 位, BIN
D1	指定方向输出端口编号	位
D2	指定脉冲输出端口编号	位

## 3、适用软元件

字软元件	操作数	系统							
		DX	DY	DM	DS	T	C	D	V、ZK、H
	S							√	√
位软元件	操作数	系统							
		X	Y	M	S	T	C		
	D1		√						
	D2		√						

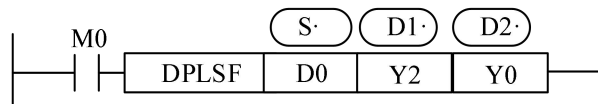
## 功能和动作

## 《16 位指令形式》

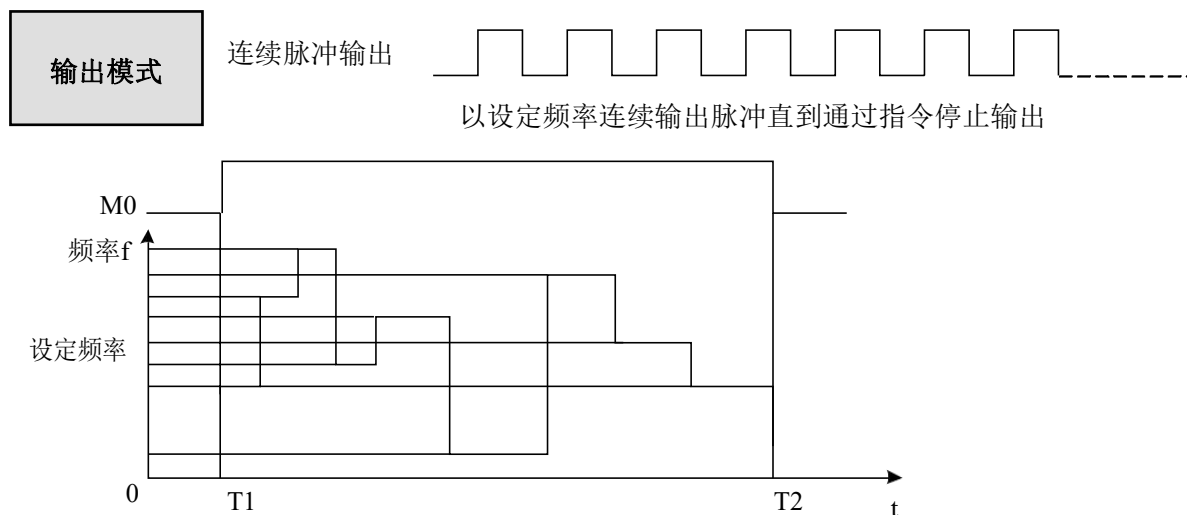


- 频率范围：频率可设正负，范围为-32767Hz~-5Hz,5Hz~32767Hz（当设定频率低于 5Hz 时，以 5Hz 的频率输出。）
- 脉冲输出：一般 2 轴为 Y0、Y1。
- 随着 D0 中设定频率的改变，从 Y0 输出的脉冲频率也跟着变化。
- 当脉冲频率为 0Hz 时，则结束脉冲输出。
- 当频率值为正值时，Y2为低电平，当频率值为负值时，Y2是高电平。
- 频率跳变时无加减速时间。
- 当脉冲指令的前提条件导通时，立即以设定的脉冲频率开始连续的输出脉冲，过程脉冲频率可调，直至脉冲指令的前提条件断开；可用于以可变脉冲频率连续输出脉冲。

## 《32 位指令形式》

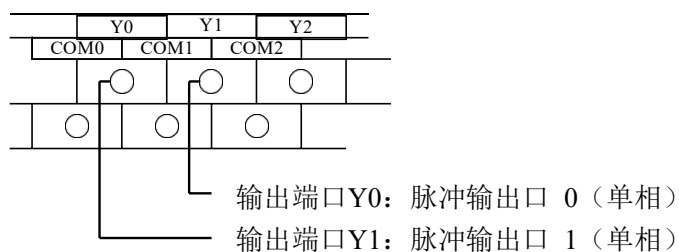


- 频率范围：频率可设正负，范围为-150KHz~5Hz,5Hz~100KHz（当设定频率低于 5Hz 时，以 5Hz 的频率输出。）
- 脉冲输出：一般 2 轴为 Y0、Y1。
- 随着 D0（双字）中设定频率的改变，从 Y0 输出的脉冲频率也跟着变化。
- 当频率值为正值时，Y2为低电平，当频率值为负值时，Y2是高电平。
- 频率跳变时无加减速时间。
- 当脉冲指令的前提条件导通时，立即以设定的脉冲频率开始连续的输出脉冲，过程脉冲频率可调，直至脉冲指令的前提条件断开；可用于以可变脉冲频率连续输出脉冲。



说明：T1 为脉冲启动时间，T2 为脉冲结束时间。

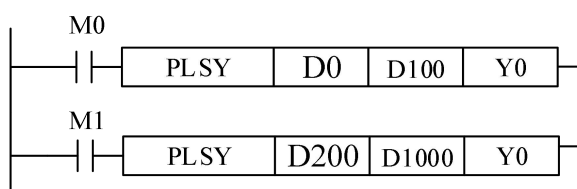
### 6-3. 输出端子接线



### 6-4. 注意事项

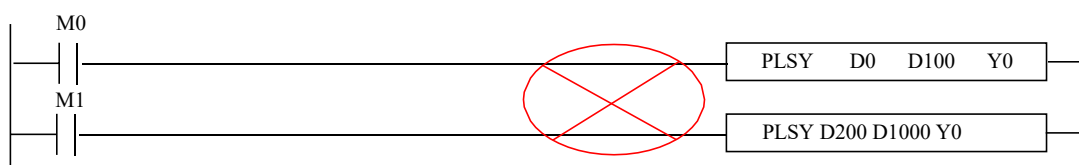
#### 1、脉冲输出不能进行双输出

- 无论是在同一个主程序中，还是在同一个流程或者子程序中，都不能对同一个输出口 Y 编写两条或两条以上的脉冲输出指令（相同与不同的脉冲指令都不可以）。
- 以下面的两条脉冲指令举例说明哪些是常见的脉冲双线圈输出：

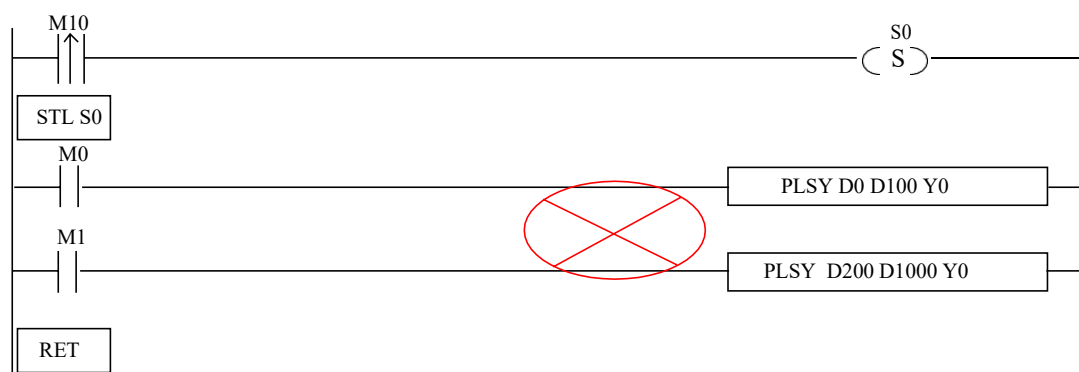


不能进行双输出的情况有以下几种：

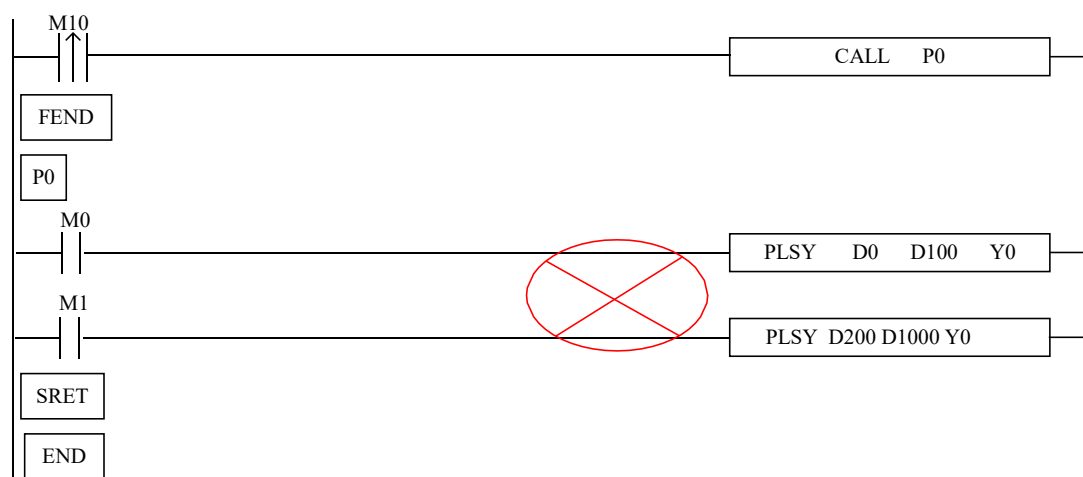
#### (1) 同时出现在主程序中



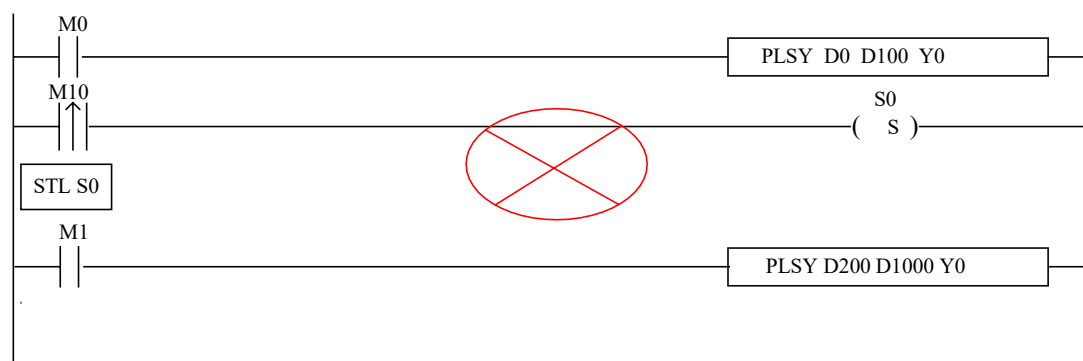
#### (2) 同时出现在流程（STL）中



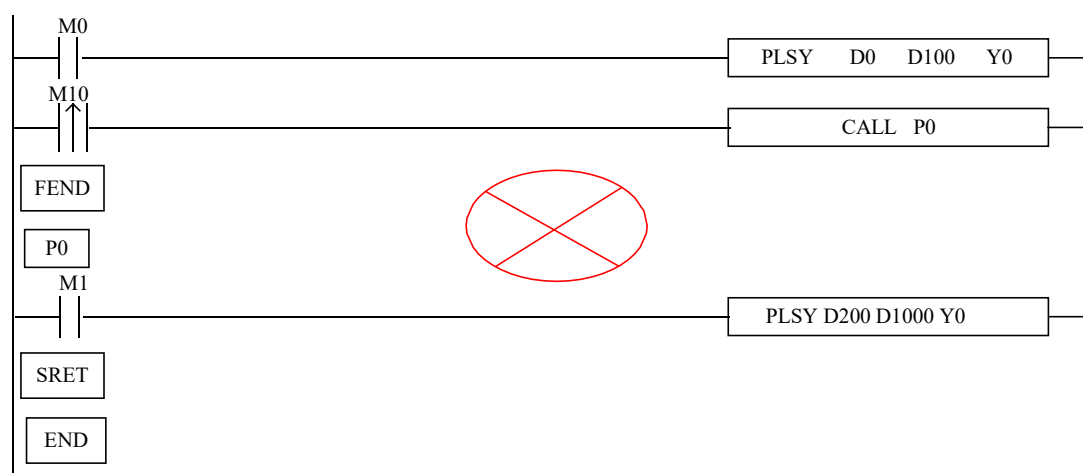
(3) 同时出现在子程序中



(4) 一条在主程序中，一条在流程（STL）中

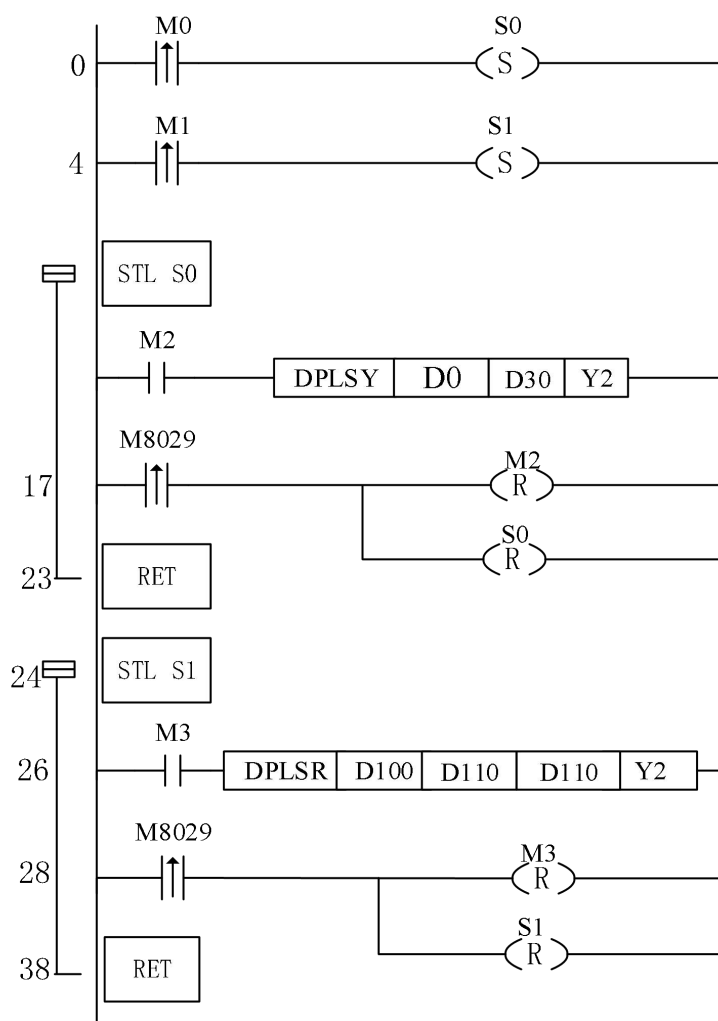


(5) 一条在主程序中，一条在子程序中等等。



当需要执行多条脉冲指令时，我们可以通过书写多个流程或者子程序，每个流程或者子程序里面都只有一条同一端口输出的脉冲指令。

例：



**注意：**两个流程（STL）不可以同时导通

# 7 通讯功能

---

本章论述了 CZA 系列可编程控制器的通讯功能，内容主要包括通讯的基本概念、Modbus 通讯、自由格式通讯。

7-1. 概述

7-2. Modbus 通讯功能

7-3. 自由格式通讯

## 通讯功能相关指令一览

指令助记符	功能	章节
MODBUS通讯		
COLR	线圈读	7-2-3
INPR	输入线圈读	7-2-3
COLW	单个线圈写	7-2-3
MCLW	多个线圈写	7-2-3
REGR	寄存器读	7-2-3
INRR	输入寄存器读	7-2-3
REGW	单个寄存器写	7-2-3
MRGW	多个寄存器写	7-2-3
自由格式通讯		
RS	串行数据通讯	7-3-2

## 7-1. 概述

CZA系列可编程控制器本体可以满足你的通讯和网络需求，它不仅支持比较简单的网络（Modbus 协议、自由通讯协议），还支持比较复杂的网络。

CZA系列可编程控制器都支持 Modbus 协议、自由协议通讯功能。

### 7-1-1. 通讯口

#### 通讯口

一. CZA 系列可编程控制器16点本体有1个通讯口。

此通讯口支持RS232\*2, RS485\*1。

DB九芯定义为：

- ① RS232(2、3、5) 为编程口，只支持MODBUS从站协议以及信捷XC协议（2RX/3TX/5GND）。
- ② RS232(1、4、5) 支持MODBUS的主从站协议、信捷XC协议和支持三菱通讯协议。(1RX/4TX/5GND)
- ③ RS485(8、9) 支持MODBUS主从站协议、信捷XC协议和三菱通讯协议。(8A/9B)

二. CZA 系列可编程控制器24点和30点本体有2 个通讯口（Port0、Port1）。

- ① Port1只支持RS232\*1，只支持MODBUS从站协议以及信捷XC协议（2RX/3TX/5GND），为编程口。
- ② Port0支持RS232\*1和RS485\*1，都支持MODBUS主从站协议、信捷XC协议和三菱通讯协议。（2RX/3TX/5GND）、（8A/9B）



### 1、RS232 通讯口

➤ **Port 1**引脚图如下:

2/1: RXD  
3/4: TXD  
5: GND

Mini Din 9 芯插座 (孔)

**Port0**引脚图如下:

2:RXD  
3:TXD 8: A  
5:GND 9: B

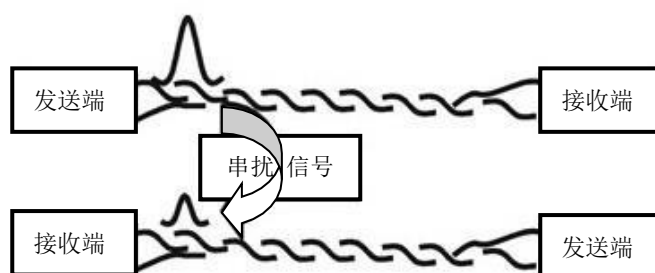
Mini Din 9芯插座 (孔)

注意: (1) **Port1** 只支持RS232(编程口)。

(2) **Port0** 既支持RS232 (串口2) 又支持RS485 (串口1)。

### 2、RS485 通讯口

RS485 通讯口引脚A 为 “+” 信号、B 为 “-” 信号。



## 7-1-2. 通讯参数

## 通讯参数

站号	Modbus 站号 1~254、255 (FF) 为自由格式通讯
波特率	300bps~115.2Kbps
数据位	7 个数据位 (无校验)、8 个数据位 (奇偶校验)
停止位	2 个停止位、1 位停止位
校验	偶校验、奇校验、无校验

编程口RS232默认参数：站号为 1、波特率 38400bps、7个数据位、1 个停止位、无校验。

通讯口RS485和RS232默认参数：MODBUS从站站号分别为 1和2、波特率 19200bps、7个数据位、1 个停止位、无校验。

CZA系列 PLC 可对通讯口进行通讯参数设置（见附录1）。

## 7-2. MODBUS 通讯功能

### 7-2-1. 通讯功能

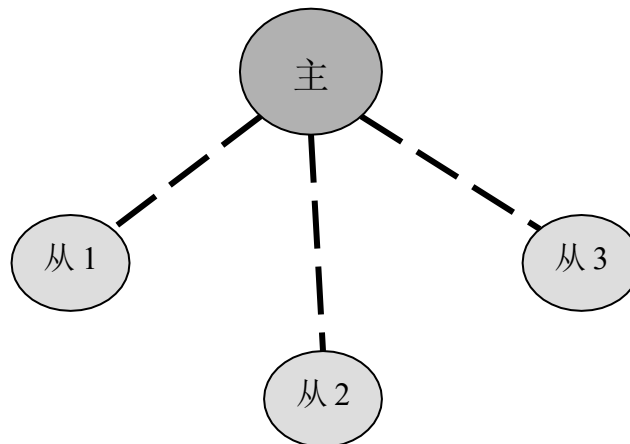
CZA 系列可编程控制器本体支持 Modbus 协议通讯主、从机形式。

**主站形式：**可编程控制器作为主站设备时，通过 Modbus 指令可与其它使用 MODBUS-RTU 协议的从机设备通讯；与其他设备进行数据交换（对于非 MODBUS-RTU 协议通讯，将在自由通讯章节做介绍）。

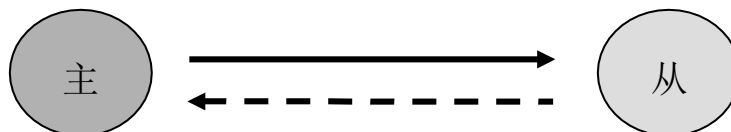
例：CZA系列PLC，可以通过通讯来控制变频器。

**从站形式：**可编程控制器作为从站设备时，只能对其它主站的要求作出响应。

**主从的概念：**在 RS485 网络中，某一时刻，可以有一主多从（如下图），其中主站可以对其中任意从站进行读写操作，从站之间不可直接进行数据交换，主站需编写通讯程序，对其中的某个从站进行读写，从站无需编写通讯程序，只需对主站的读写进行响应即可。（接线方式：所有的 485+连在一起，所有的 485-连在一起）



在 RS232 网络中（如下图），只能一对一通讯，某一时刻只有一主一从。



之所以图中有虚线箭头（包括 RS485 网络中），是因为理论上在两个网络中，只要各个 PLC 不发数据，网络中任意 PLC 都可以用来作为主站，其它 PLC 作为从站；但是由于多个 PLC 之间没有一个统一的时钟基准，容易出现在同一时刻有多个 PLC 发送数据，会导致通讯冲突失败，因此不建议这样使用。

**注意：**对于CZA系列PLC，RS232只支持半双工。

## 7-2-2. 通讯地址

可编程控制器内部软元件编号与对应的 Modbus 地址编号如下表示：

## 线圈空间：（Modbus 地址前缀为“0x”）

位元件地址	Modbus 地址 (十进制 K)	Modbus 地址 (十六进制 H)
M0~M7999	0~7999	0~4000
X0~X1037	16384~18432	4000~4800
Y0~Y1037	18432~20480	4800~5000
S0~S1023	20480~24576	5000~6000
T0~T618	25600~27648	6400~6C00
C0~C634	27648~29696	6C00~7400

## 寄存器空间：（Modbus 地址前缀为“4x”）

字元件地址	Modbus 地址 (十进制 K)	Modbus 地址 (十六进制 H)
D0~D7999	0~7999	0~3000
T0~T618	12288~14336	3000~3800
C0~C634	14336~16384	3800~4000
D8000~	16384~17408	4000~4400

- 以上表格中的地址在 PLC 作为下位机且使用 Modbus-RTU 协议通讯时使用，一般上位机为：组态/触摸屏/PLC等。
- 如果上位机为PLC 则按照 Modbus-RTU 协议编写程序。
- 输入输出点为八进制，请按照八进制计算对应的输入输出点 Modbus 地址，例如：Y0 对应的Modbus 地址是H4800，Y10 对应的Modbus 地址是H4808（并不是H4810），Y11 对应的 Modbus 地址是 H4809，Y12 对应的 Modbus 地址是 H480A，Y13 对应的 Modbus 地址是 H480B，Y17对应的Modbus 地址是 H480F，Y20对应的Modbus 地址是 H4810（并不是H4820）。

### 7-2-3. Modbus 通讯数据格式

Modbus 通讯传输模式包含两种传输模式，分别为 RTU 模式与 ASCII 模式；它定义了报文域的位内容在线路上串行的传送；它确定了信息如何打包为报文和解码；Modbus 串行链路上所有设备的传输模式（和串行口参数）必须相同。

Modbus 通讯数据结构：

#### 1、RTU 模式：

当设备使用 RTU（Remote Terminal Unit）模式在 Modbus 串行链路通信，报文中每个8位字节含有两个 4 位十六进制字符。这种模式的主要优点是较高的数据密度，在相同的波特率下比 ASCII 模式有更高的吞吐率。每个报文必须以连续的字符流传送。

RTU 模式帧检验域：循环冗余校验（CRC）。

RTU 模式帧描述：

Modbus 站号	功能代码	数据	CRC	
1 字节	1 字节	0~124 字节	2 字节	
			CRC 低	CRC 高

格式：

START	保持无输入信号大于等于10ms
Address	通讯地址：8-bit 二进制地址
Function	功能码：8-bit 二进制地址
DATA (n - 1)	资料内容： N*8-bit 资料，N<=8，最大8个字节
.....	
DATA 0	
CRC CHK Low	CRC校验码
CRC CHK High	16-bit CRC校验码由2个8-bit 二进制组合
END	保持无出入信号大于等于10ms

#### 2、通讯地址：

00H：所有CZA系列PLC广播（broadcast）——广播时候下位机不回复数据。

01H：对01地址PLC通讯。

0FH：对15地址PLC通讯。

10H：对16地址PLC通讯。以此类推 .....，最大可到254（FEH）。

## 3、功能码（Function）与资料内容（DATA）：

功能码	功能	对应的modbus指令	备注
01H	读线圈指令	COLR	
02H	读输入线圈指令	INPR	
03H	读出寄存器内容	REGR	
04H	读输入寄存器指令	INRR	
05H	写单个线圈指令	COLW	
06H	写单个寄存器指令	REGW	
10H	写多个寄存器指令	MRGW	
0FH	写多个线圈指令	MCLW	

1) 以 06 功能码（单个寄存器写）为例，介绍数据格式（其余功能码与此类似）：例如：上位机对PLC的H0002地址即D2写数据K5000（即H1388）。

RTU 模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	06H	功能码	06H
寄存器地址	00H	寄存器地址	00H
	02H		02H
数据内容	13H	数据内容	13H
	88H		88H
CRC CHECK Low	25H	CRC CHECK Low	25H
CRC CHECK High	5CH	CRC CHECK High	5CH

发送报文：01 06 00 02 13 88 25 5C(向从站PLC中D2寄存器写入内容K5000)

接收报文：01 06 00 02 13 88 25 5C

发送报文：01 06 00 03 0F A0 7C 42(向从站PLC中D3寄存器写入内容K4000)

接收报文：01 06 00 03 0F A0 7C 42

说明：

- 1) 地址即PLC的站号。
- 2) 功能码即 Modbus-RTU 协议中所定义的读写操作代码。
- 3) 寄存器地址即 7-2-2 章节表格中所列出的PLC modbus 通讯地址。
- 4) 数据内容即为往D2 寄存器中写的的数据。
- 5) CRC CHECK Low / CRC CHECK High 为 CRC 校验的低位和低位数据。

以下是 REGW 指令和 RTU 协议数据的对应关系（其余指令与此类似）：

REGW	功能码 06H
K1	站号地址
H0002	modbus 地址
K5000	数据内容即 1388H
K2	PLC 通讯串口号

完整的数据串是：01H 06H 00H 02H 13H 88H（系统自动进行CRC 校验）如果用串口调试工具监控串口 2，可得到数据如下：01 06 00 02 13 88 25 5C

**注意：**在程序中使用的数据不分二进制、十进制、十六进制，只要保证数值上相等即可，

例如：B10000、K16 和 H10 值相等，所以以下三条通讯指令功能相同：

REGW	K1	B111110100	D1	K2
REGW	K1	K500	D1	K2
REGW	K1	H1F4	D1	K2

## 2) 功能码 01H/02H：读线圈/读输入线圈

例如：读取线圈地址4800H为首的2个线圈的状态（Y0、Y1）。此时，Y0、Y1均为ON。

RTU模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	01H/02H	功能码	01H/02H
线圈地址	48H	字节数	01H
	00H		
线圈个数	00H	数据内容	03H
	02H		
CRC CHECK Low	AAH	CRC CHECK Low	11H
CRC CHECK High	6BH	CRC CHECK High	89H

由于 Y0 和 Y1 的状态均为 ON，所以数据内容为 03H（0000 0011）。

例：发送报文：01 02 48 00 00 02 AA 6B

回应报文：01 01 01 00 51 88（此时Y0、Y1的状态为OFF）

发送报文：01 01 48 00 00 02 AA 6B

回应报文：01 01 01 03 11 89（此时Y0、Y1的状态为ON）

发送报文：01 01 00 00 00 02 BD CB

回应报文：01 01 01 00 51 88（此时M0、M1为OFF）

发送报文：01 01 00 00 00 02 AA 6B

回应报文：01 01 01 03 11 89（此时M0、M1为ON）

**3) 功能码 03H: 读出寄存器内容**

例如：读出寄存器地址3E8H为首的2个寄存器的内容（D1000、D1001）。

RTU模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	03H	功能码	03H
寄存器地址	03H	字节数	04H
	E8H		
寄存器个数	00H	数据内容	12H
			2EH
	02H		04H
			E8H
CRC CHECK Low	44H	CRC CHECK Low	9DH
CRC CHECK High	7BH	CRC CHECK High	CCH

此时，读出的D1000 和 D1001 中的数值分别是：122EH（4654）和 04E8H（1256）。

例如：发送报文：01 03 03 E8 00 02 44 7B

回应报文：01 03 04 12 2E 04 E8 9D CC

发送报文：01 03 00 00 00 02 C4 0B

回应报文：01 03 04 00 0A 00 14 DA 3E（D0和D1中的为10和20）

**4) 功能码 05H: 写单个线圈**

例如：将线圈地址4800H（Y0）置ON。

RTU模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	05H	功能码	05H
线圈地址	48H	线圈地址	60H
	00H		00H
数据内容 (低在前高在后)	FFH	数据内容	FFH
	00H		00H
CRC CHECK Low	9BH	CRC CHECK Low	9BH
CRC CHECK High	9AH	CRC CHECK High	9AH

注意：单个线圈写时，ON 为 00FFH，OFF 为 0000H；且数据内容是低字节数据在前，高字节数据在后。

发送报文：01 05 48 00 FF 00 9B 9A（将PLC的Y0置ON）

回应报文：01 05 48 00 FF 00 9B 9A

发送报文：01 05 48 01 FF 00 CA 5A（将PLC的Y1置ON）

回应报文：01 05 48 01 FF 00 CA 5A



## 5) 功能码 0FH: 多个线圈写

例如：对PLC中的 4800H (Y0) 为首的 16 个线圈进行写入。

RTU 模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	0FH	功能码	0FH
线圈地址	48H	线圈地址	60H
	00H		00H
线圈个数	00H	线圈个数	00H
	10H		10H
字节数	02H	-	-
数据内容 (低在前高在后)	03H		
	01H		
CRC CHECK Low	EBH	CRC CHECK Low	43H
CRC CHECK High	14H	CRC CHECK High	A7H

数据内容为 0103H, 其二进制表示为 0000 0001 0000 0011, 写入对应 Y17~Y0, 所以 Y0、Y1、Y10 置 ON。

注意：在写数据内容时，低字节数据在前，高字节数据在后。

## 6) 功能码 10H: 多个寄存器写

例如：对PLC中的 0000H (D0) 为首的 3 个寄存器进行写入。

RTU 模式：

询问信息格式		回应信息格式	
地址	01H	地址	01H
功能码	10H	功能码	10H
寄存器地址	00H	寄存器地址	00H
	00H		00H
寄存器个数	00H	寄存器个数	00H
	03H		03H
字节数	06H	-	-
数据内容	00H		
	01H		
	00H		
	02H		
	00H		
	03H		
CRC CHECK Low	3AH	CRC CHECK Low	3AH
CRC CHECK High	81H	CRC CHECK High	81H

执行后，D0、D1、D2 中的数值分别为：1、2、3。

注意：字节数=寄存器个数\*2。

发送报文：01 10 00 00 00 03 06 00 03 00 04 00 05 23 42

接收报文：01 10 00 00 00 03 80 08 (D0、D1、D2的值为3、4、5)

### 7-3. 通讯指令

Modbus 指令，分为线圈读写、寄存器读写，下面具体介绍这些指令的用法。

通讯指令中各操作数定义说明：

1. 远端通讯局号：与PLC所连接下位机的串口站号。

例如：PLC 连接了三台变频器，要通过通讯来读写参数，此时将变频器的站号设置成 1.2.3，即变频器为下位机，PLC 为上位机且下位机的远端通讯局号分别为 1.2.3（下位机站号和上位机站号可设置成相同）

2. 远端线圈/寄存器首地址编号：

指定远端线圈/寄存器个数：PLC 对下位机读写操作时候的第一个线圈/寄存器地址，一般结合“指定线圈/寄存器个数”一起使用。

例如：PLC 要读一台伺服的实际转速（0X700）、输入速度（0X701）、内部转矩（0X702），则远端寄存器首地址为 0X700，指定线圈个数为 K3。

3. 本地接收/发送线圈/寄存器地址：PLC 中需要与下位机中进行数据交换的线圈/寄存器。

例如：写线圈 M0：将 M0 状态写到下位机中指定地址

写寄存器 D0：将 D0 值写到下位机指定地址

读线圈 M1：将下位机指定地址中的内容读到 M1

读寄存器 D1：将下位机指定寄存器内容读到 D1

## 7-3-1. 线圈读[COLR]

## 1、指令概述

将指定局号中指定线圈状态读到本机内指定线圈中的指令。

线圈读[COLR]			
16 位指令	COLR	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

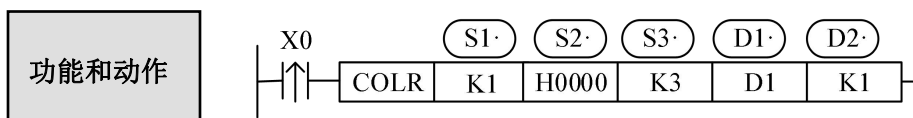
操作数	作用	类型
S1	指定远端通讯局号	16 位, BIN
S2	指定远端线圈首地址编号 (Modbus地址)	16 位, BIN
S3	指定线圈个数的数值	16 位, BIN
D1	指定本地接收线圈的首地址	位
D2	指定串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		√
	S2					√	√	√		√
	S3					√	√	√		√
	D2									K

位软元件	操作数	系统					
		X	Y	M	S	T	C
	D1	√	√	√	√	√	√



- 读线圈指令，Modbus 功能码为 01H。
- 串口号范围：K1~K2 (1:RS485, 2:RS232)。
- 操作数 S3：K1~K984，即读取线圈的最大个数为 984。

## 7-3-2. 输入线圈读[INPR]

## 1、指令概述

将指定局号中指定输入线圈状态读到本机内指定线圈中的指令。

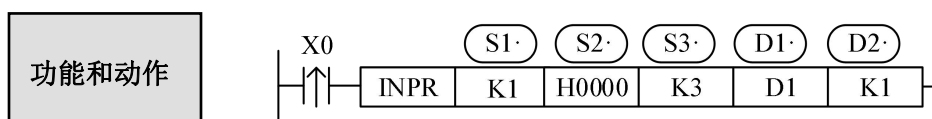
输入线圈读[INPR]			
16 位指令	INPR	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
S1	指定远端通讯局号	16 位, BIN
S2	指定远端线圈首地址编号 (Modbus地址)	16 位, BIN
S3	指定线圈个数的数值或寄存器地址	16 位, BIN
D1	指定本地接收线圈的首地址编号	位
D2	指定 PLC 串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		√
	S2					√	√	√		√
	S3					√	√	√		√
D2										K
位软元件	操作数	系统								
		X	Y	M	S	T	C			
	D1	√	√	√	√	√	√			



- 读输入线圈指令，Modbus 功能码为 02H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。
- 操作数 S3：K1~K984，读取输入线圈的最大个数为 984。

## 7-3-3. 单个线圈写[COLW]

## 1、指令概述

将本机内指定线圈状态写到指定局号中指定线圈的指令。

单个线圈写[COLW]			
16 位指令	COLW	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
D1	指定远端通讯局号	16 位, BIN
D2	指定远端线圈首地址编号 (Modbus地址)	16 位, BIN
S1	指定本地发送线圈首地址编号	位
S2	指定串口编号	16 位, BIN

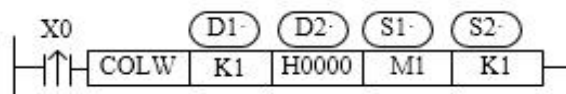
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	D1					√	√	√		√
	D2					√	√	√		√
	S2									K

位软元件	操作数	系统					
		X	Y	M	S	T	C
	S1	√	√	√	√	√	√

## 功能和动作



- 写单个线圈指令，Modbus 功能码为 05H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。

## 7-3-4. 多个线圈写[MCLW]

## 1、指令概述

将本机内指定的多个线圈的状态写到指定局号中指定线圈的指令。

多个线圈写[MCLW]			
16 位指令	MCLW	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
D1	指定远端通讯局号	16 位, BIN
D2	指定远端线圈首地址编号 (Modbus地址)	16 位, BIN
D3	指定线圈个数的数值	16 位, BIN
S1	指定本地发送线圈首地址编号	位
S2	指定串口编号	16 位, BIN

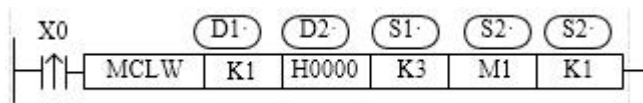
## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
D1						√	√	√		√
D2						√	√	√		√
D3						√	√	√		√
S2										K

位软元件	操作数	系统					
		X	Y	M	S	T	C
S1		√	√	√	√	√	√

功能和动作



- 写多个线圈指令，Modbus 功能码为 0FH。
- 串口号范围：K1~K2(1:RS485,2:RS232)。
- 操作数 D3 即写线圈的最大个数为 952。

## 7-3-5. 寄存器读[REGR]

## 1、指令概述

将指定局号指定寄存器读到本机内指定寄存器的指令。

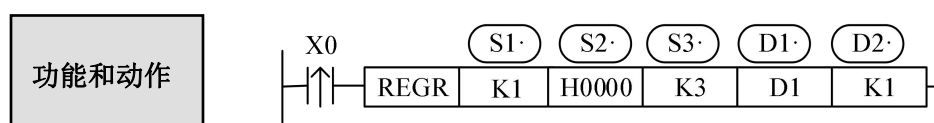
寄存器读[REGR]			
16 位指令	REGR	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
S1	指定远端通讯局号	16 位, BIN
S2	指定远端寄存器首地址编号 (Modbus地址)	16 位, BIN
S3	指定寄存器个数的数值	16 位, BIN
D1	指定本地接收寄存器首地址编号	16 位, BIN
D2	指定串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		√
	S2					√	√	√		√
	S3					√	√	√		√
	D1							√		
	D2									K



- 读寄存器指令，Modbus 功能码为 03H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。
- 操作数 S3 及读寄存器的最大个数为 61。

## 7-3-6. 输入寄存器读[INRR]

## 1、指令概述

将指定局号指定寄存器读到本机内指定寄存器的指令。

输入寄存器读[INRR]			
16 位指令	INRR	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

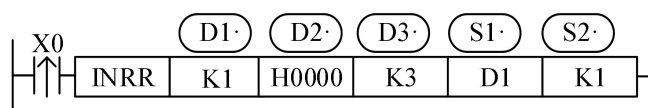
## 2、操作数

操作数	作用	类型
S1	指定远端通讯局号	16 位, BIN
S2	指定远端寄存器首地址编号 (Modbus地址)	16 位, BIN
S3	指定寄存器个数的数值	16 位, BIN
D1	指定本地接收寄存器首地址编号	16 位, BIN
D2	指定串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1					√	√	√		√
	S2					√	√	√		√
	S3					√	√	√		√
	D1							√		
	D2									K

## 功能和动作



- 读寄存器指令，Modbus 功能码为 04H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。
- 操作数 S3 及读寄存器的最大个数为 60。



## 7-3-7. 单个寄存器读[REGW]

## 1、指令概述

将本机内指定寄存器写到指定局号指定寄存器的指令。

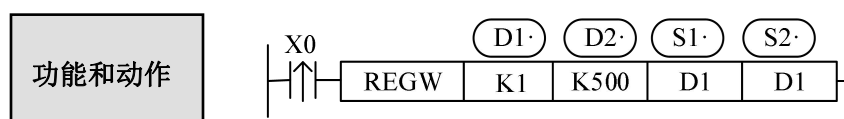
单个寄存器写[REGW]			
16 位指令	REGW	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
D1	指定远端通讯局号的数值	16 位, BIN
D2	指定远端寄存器首地址编号 (Modbus地址)	16 位, BIN
S1	指定本地发送寄存器首地址编号	16 位, BIN
S2	指定串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	D1					√	√	√		√
	D2					√	√	√		√
	S1							√		
	S2									K



- 写单个寄存器指令，Modbus 功能码为 06H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。

## 7-3-8. 多个寄存器读[MRGW]

## 1、指令概述

将本机内指定寄存器写到指定局号指定寄存器的指令。

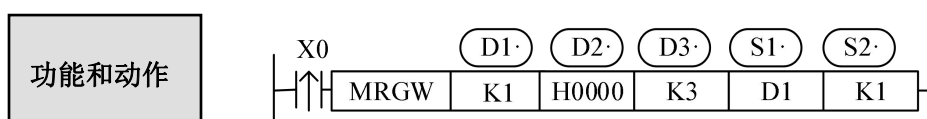
多个寄存器写[MRGW]			
16 位指令	MRGW	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

## 2、操作数

操作数	作用	类型
D1	指定远端通讯局号	16 位, BIN
D2	指定远端寄存器首地址编号	16 位, BIN
D3	指定寄存器个数的数值	16 位, BIN
S1	指定本地发送寄存器首地址编号	16 位, BIN
S2	指定串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	D1					√	√	√		√
	D2					√	√	√		√
	S1							√		
	S2									K



- 写多个寄存器指令，Modbus 功能码为 10H。
- 串口号范围：K1~K2(1:RS485,2:RS232)。
- 操作数 D3 即写寄存器的最大个数为 59。

**7-3-9. 通讯样例及说明**

## ● 接线方式

接线方式有如下两种：

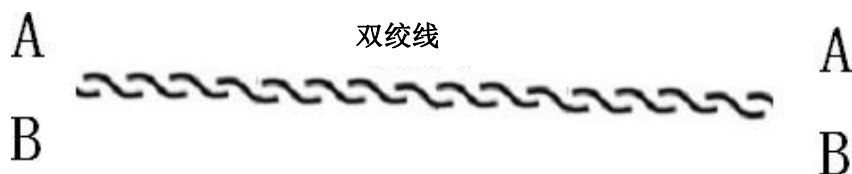
## A、232 连接方式

注意：（1）上图中以\*1 标记的通讯口 2 只标出了RS232 引脚，由于 RS485 端子已外置（A、B），因此这里不再列出。

（2）CZA 系列 PLC，RS232 通讯不支持全双工，因此只能单向通讯。

（3）RS232 通讯距离短（大约 13 米左右且和现场环境密切相关），如需远距离通讯请选用RS485 通讯方式。

## B、485 连接方式



A 接 A、B 接B，如果一个主站多个从站，只需将所有的A 相连，所有的B 相连。

A 为 485+、B 为 485-。

## 7-4. 自由格式通讯

### 7-4-1. 通讯模式

自由格式通讯是以数据块的形式进行数据传送，受 PLC 缓存的限制，每次发送数据量最大为 128 字节。

#### 1. 自由格式通讯模式说明

所谓自由格式，即自定义协议通讯，现在市场上很多智能设备都支持 RS232 或者 RS485 通讯，而各家产品所使用的协议不尽相同，例如：PLC 使用标准的 Modbus-RTU 协议，一些温度控制器厂家使用的自定义协议；如果用 PLC 和温度控制器通讯，读取温度控制器采集的当前温度，则需使用自由格式通讯，完全按照仪表厂家的协议来发送数据，这样即可通讯上。

自由格式通讯应具备的前提条件：

① Port0、Port1均可以进行自由格式通讯，但由于自由通讯时需要修改串口参数，因此不建议使用Port1。

② 波特率：300bps~115.2Kbps

③ 数据格式必须与下位机设置相同，有以下几种可选：

数据位：7bits、8bits

校验位：奇校验、偶校验、无校验

停止位：1 位、2 位

④ 通讯形式：8 位、16 位

选择 8 位缓冲形式进行通讯时，通讯过程中寄存器的高字节是无效的，PLC 只利用寄存器的低字节进行发送和接收数据。

选择 16 位缓冲形式进行通讯时，PLC 发送数据时，是先发送低字节数据，再发送高字节数据。

## 7-4-2. RS串行通讯指令

## 1、指令概述

该指令是一个通讯收发指令，并将指定寄存器区域的数据，自动向串口一次发送，将串口接受到的数据存放到指定区域，相当于用户程序直接访问通讯缓冲区，借助用户程序对通讯收发缓冲区的处理，实现自定义协议的通讯。

串行数据传送			
16 位指令	RS	32 位指令	-
执行条件	常开/闭、边沿触发	适用机型	CZA

## 2、操作数

操作数	作用	类型
S1	指定发送数据地址的首地址编号	16 位, BIN
m	指定发送数据的点数	16 位, BIN
D	指定接受数据地址首地址编号	16 位, BIN
n	指定接受数据的点数 (字节数)	16 位, BIN
S2	指定的串口编号	16 位, BIN

## 3、适用软元件

字软元件	操作数	系统								
		DX	DY	DM	DS	T	C	D	V、Z	K、H
	S1							√		
	m							√		√
	D							√		
	n							√		√
	S2									k



- m,n的取值范围为0~4096，但请指定 $m+n \leq 8000$ ， $n \leq m$ 。
- “S2”指定的串口编号，一般为K1或者K2。

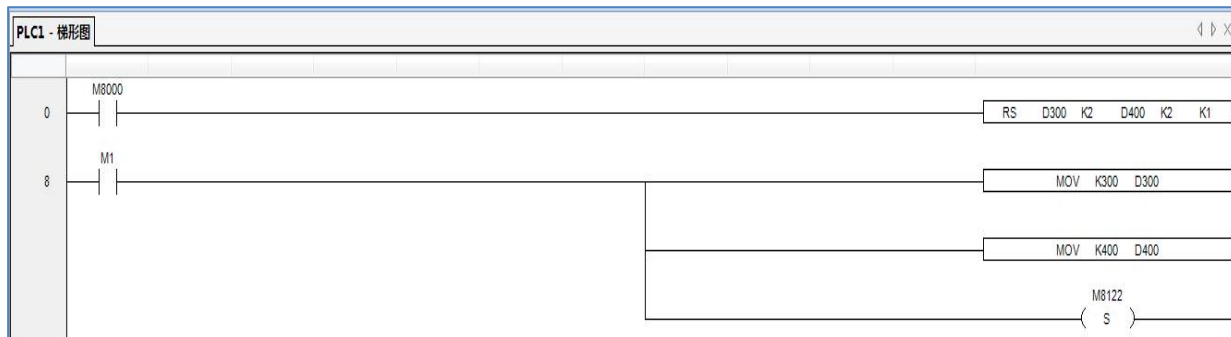
• •

### 发收信的 顺控程序

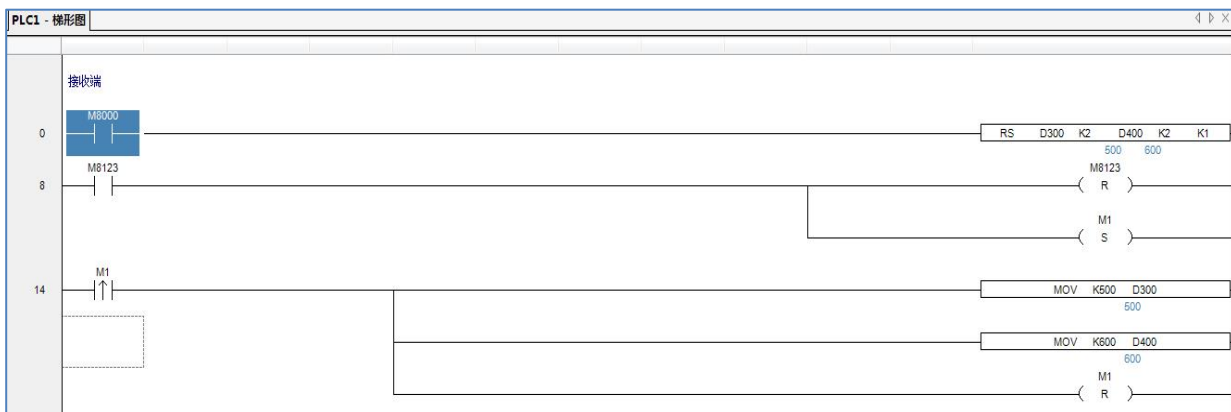
RS指令指定来自可编程控制器的发送数据的起始地址与数据点数以及接收数据存储用起始地址与可以接收的最大数据点数（字节数）。如下例：串口设置不分主从站  
串口设置：



发送端：



接收端：



发送请求

- 在接受等待状态或接受完成状态时，用脉冲指令置位M8122,就开始发送从D200开始的D0长度的数据，发送结束时M8122自动复位
- 另外，RS指令的驱动输入X10变成ON状态时，可编程控制器就进入接受等待状态。（发送标志：RS485对应M8122;RS232对应M8127）

接收完成

M8128)

- 接受完成标志M8123 ON后，请先把接受传送到其他存储地址后，再对M8123进行复位。（接收完成标志：RS485对应M8123;RS232对应M8128）
- 复位M8123后，则再次进入接受等待状态。M8123的复位由顺控程序执行。而且RS指令的驱动输入X10进入ON状态后，可编程控制器变为接受等待状态。
- 用(D1)=0执行RS指令时，则M8123(执行完成标志)将不动作，也不进入接受等待状态。从这个状态如需成为接受等待状态，请使(D1) ≥ 1之后，再对M8123进行ON→OFF操作。

# 8

## C语言功能块

本章重点介绍C语言编辑功能块的特点、编辑、指令调用、函数的存储、应用要点等，并附有常用函数列表。

8-1. 概述

8-2. 指令形式

8-3. 操作步骤

8-4. 函数的导入、导出

8-5. 功能块的编辑

8-6. 程序举例



CZA系列支持用户在PLC上位机编程软件中利用C语言编写功能块，在需要的地方进行调用，其最大的优点是支持几乎所有的C语言函数（支持全局变量），增强了程序的保密性，同时由于可进行多处调用和不同文件的调用，大大提高了编程人员的效率。

1、指令概述

将编辑好的C语言功能块在指定区域调用的指令。

调用C语言功能块[NAME_C]			
16 位指令	NAME_C	32 位指令	-
执行条件	常开/闭、上升沿触发	适用机型	CZA
硬件要求	-	软件要求	-

2、操作数

操作数	作用	类型
S1	C 语言函数功能块名称，由用户自定义	字符串
S2	对应C 语言函数内字M的起始地址	16 位，BIN
S3	对应C 语言函数内位D的起始地址	位，BIN

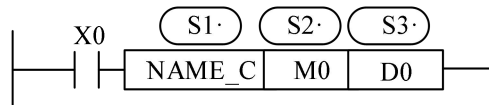
3、适用软元件

字软元件	操作数	系统									
		DX	DY	DM	DS	T	C	D	V、Z	K、H	
	S1										
	S3						√				

位软元件	操作数	系统					
		X	Y	M	S	T	C
	S2			√			

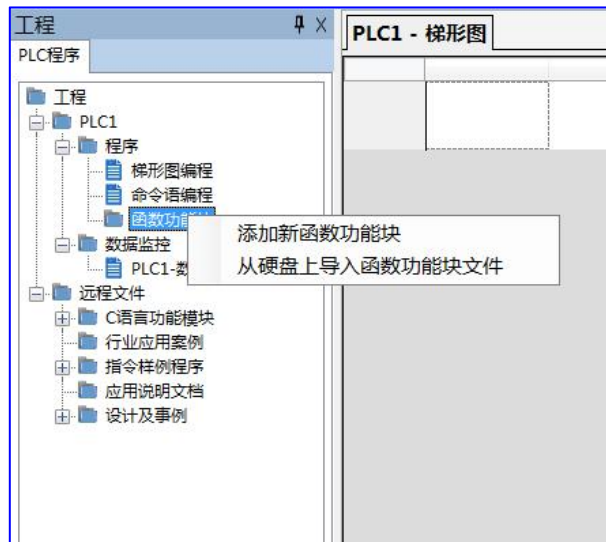
功能和动作



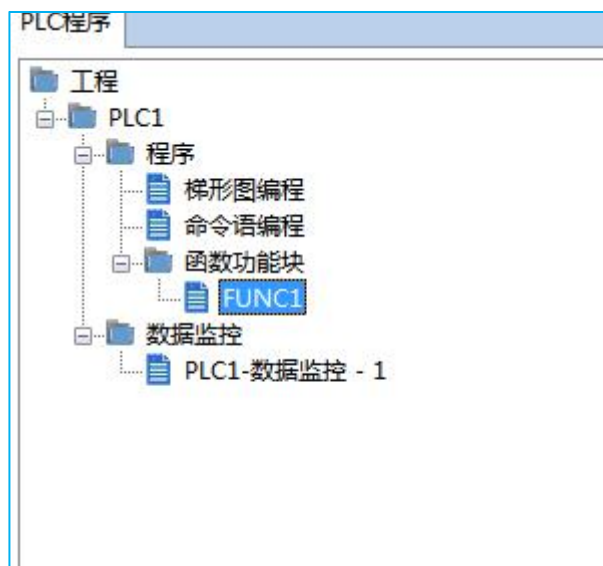
- S1为用户自定义函数名称，由数字，英文，下划线组成，首字符不能为数字，名称长度需<=9个字符。
- 函数名称不能与PLC内置指令名冲突，如LD、ADD、SUB、PLSR等。
- 函数名称不能与当前PLC已经存在的函数功能块同名。

### 8-3.操作步骤

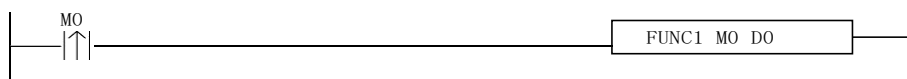
1、打开PLC 编辑软件，在左侧的“工程”工具栏内选择“函数功能块”，右击选择“添加新函数功能块”。



2、出现如下对话框，填写所要编辑函数的名称。



功能块名称即为梯形图中调用函数块时使用的名称，例如上图为：FUNC1 则在梯形图中调用时指令应写成如下格式：



2、在新建完成后，会出现下面的编辑画面。

```

1  /******
2  FunctionName: FUNC1
3  Description:
4
5  *****/
6  #include <include.h>
7  void FUNC1(unsigned int * M, unsigned short int * D)
8  {
9
10 }
11

```

参数传递方式：在梯形图调用时：

1、若数据格式为单字，则寄存器中的地址为梯形图指令中 D 的地址+C 语言中 WORD[] 的地址；继电器 M 的地址为梯形图中 M 的地址+C 语言中 M[] 的地址

2、若数据格式为双字，即寄存器中的地址为 D 的地址+DWORD[] 的地址\*2。

如上图，当指令FUNC1 DO MO中的为D0，M0开始，则WORD[0]为D0，WORD[10]为D10，M[0]为M0，M[10]为M10；如梯形图中使用的参数为D100，M100，则WORD[0]为D100，M[0]为M100；因此，字与位元件的首地址和是否使用掉电保持型数据寄存器和线圈由用户在PLC程序中设定。

注意：

在同一个C语言功能块里线圈和数据类型只能是一种。即要么整个C语言功能块里的所有线圈是掉电保持型，要么全是非掉电保持型；数据寄存器也是一样。

双字运算：在WORD前加个D，如DWORD[10]=100000，表示给WORD[10]WORD[11]合成的双字赋值。

浮点运算：支持在函数中定义浮点变量，以及进行浮点运算（例如：浮点数寄存器D0（双字）可表示为FLOATW[0]，FLOATW[0]=123.456）。

```

1  /*****
2  FunctionName: JJCC
3  Description:
4
5  加减乘除计算,若要赋值变量进行计算,最后计算结果要赋值到寄存器中
6  注意: WORD[],DWORD[],FLOATW[]必须大写
7
8  *****/
9  #include <include.h>
10 void JJCC(unsigned int * M, unsigned short int * D)
11 {
12     //单字计算,寄存器地址:DO=WORD[0]+D0;若指令中为D1;则D1=WORD[0]+D1,D3=WORD[2]+D1
13     WORD[0]=WORD[0]+WORD[1];
14     WORD[0]=WORD[0]-WORD[1];
15     WORD[0]=WORD[0]*WORD[1];
16     WORD[0]=WORD[0]/WORD[1];
17
18     //双字计算 (DWORD)
19     //寄存器地址:DO=2*DWORD[0]+D0;若指令中为D1,则D1=2*DWORD[0]+D1,D5=2*DWORD[2]+D1
20     //若指令中为D2,则D2=2*DWORD[0]+D2,D5=2*DWORD[2]+D2,D8=3*DWORD[2]+D2
21     //双字乘除计算结果占用连续4个寄存器地址;加减结果占用2个寄存器地址
22     DWORD[0]=DWORD[0]+DWORD[1];
23     DWORD[0]=DWORD[0]-DWORD[1];
24     DWORD[0]=DWORD[0]*DWORD[1];
25     DWORD[0]=DWORD[0]/DWORD[1];
26
27     //浮点运算 (FLOATW)
28     //寄存器地址:DO=2*FLOATW[0]+D0;若指令中为D1,则D1=2*FLOATW[0]+D1,D5=2*FLOATW[2]+D1
29     //若指令中为D2,则D2=2*FLOATW[0]+D2,D5=2*FLOATW[2]+D2,D8=3*FLOATW[2]+D2
30     //浮点乘除计算后,结果占用连续4个寄存器地址;加减结果占用2个寄存器地址
31     FLOATW[0]=FLOATW[0]+FLOATW[1];
32     FLOATW[0]=FLOATW[0]-FLOATW[1];
33     FLOATW[0]=FLOATW[0]*FLOATW[1];
34     FLOATW[0]=FLOATW[0]/FLOATW[1];
35 }

```

```

1  /*****
2  FunctionName: FUNC2
3  Description:
4  异或:相同为"0",相异为"1"
5  a,b的值不相等,则真,WORD[3]=4;相等,则为假,WORD[3]=5
6  *****/
7  #include <include.h>
8  void YIHUO(unsigned int * M, unsigned short int * D)
9  {
10     int a,b,c;
11     a=WORD[1];//将寄存器D1值赋值给a
12     b=WORD[2];
13     c=WORD[3];
14     if(a^b)
15     {
16         c=4;
17         WORD[3]=c;
18     }
19     else
20     {
21         c=5;
22         WORD[3]=c;
23     }
24 }

```

```

1  /*****
2  FunctionName: BIJIAO
3  Description:
4
5  *****/
6  #include <include.h>
7  void BIJIAO(unsigned int * M, unsigned short int * D)
8  {
9      int a,b,c;
10     if(a>1&&b>1)//逻辑与
11     {
12         c=1;
13     }
14     if(a>1||b<1)//逻辑或
15     {
16         c=2;
17     }
18     WORD[0]=c;
19 }

```

```

1  /*****
2  FunctionName: TONGHUO
3  Description:
4  Description:
5  同或:相同为"1",相异为"0"
6  a,b的值相等, 则真,WORD[3]=4; 不等, 则为假,WORD[3]=5
7  *****/
8  #include <include.h>
9  void TONGHUO(unsigned int * M, unsigned short int * D)
10 {
11     int a,b,c;
12     a=WORD[1];//将寄存器D1值赋值给a
13     b=WORD[2];
14     c=WORD[3];
15     if(!(a^b))
16     {
17         c=4;
18         WORD[3]=c;
19     }
20     else
21     {
22         c=5;
23         WORD[3]=c;
24     }
25 }

```

```

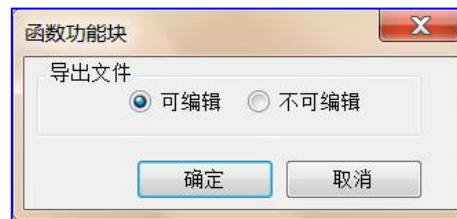
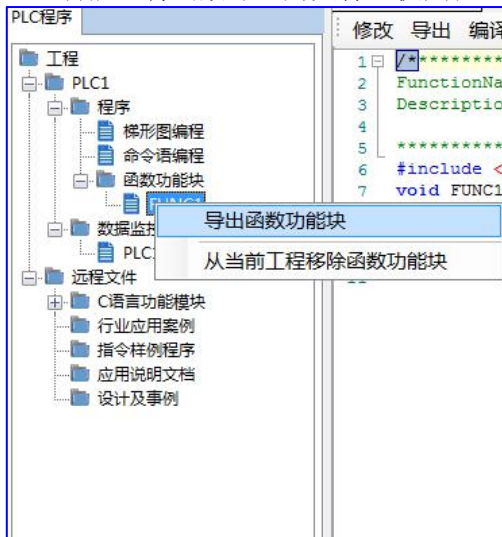
1  /*****
2  FunctionName: SANJIAO
3  Description:
4  三角函数运算采用弧度制计算
5  *****/
6  #include <include.h>
7  void SANJIAO(unsigned int * M, unsigned short int * D)
8  {
9      int a,b;
10     float c,d,f,g,h,i;
11     float PI=3.14;
12     a=WORD[10];
13     b=WORD[20];
14     c=sin(a*PI/180);
15     d=cos(a*PI/180);
16     f=tan(a*PI/180);
17     g=asin(a*PI/180);//反正弦
18     h=acos(a*PI/180);//反余弦
19     i=atan(a*PI/180);//反正切
20     //运算得出的结果, 要赋值到寄存器地址中
21     FLOATW[15]=c;//寄存器实际地址为D30
22     DWORD[20]=(int)c;//将c的格式强制转换成整型, 然后赋值给D40;
23
24 }

```

## 8-4.函数的导入、导出

### 1) 导出

(1) 功能：将函数导出为文件，供其他PLC程序导入用。

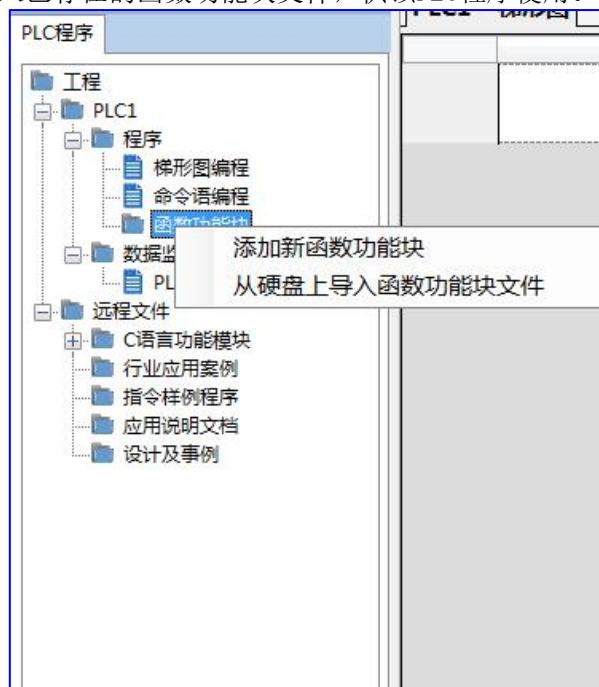


### (2) 导出方式

- a) 可编辑：将源代码也导出，并保存为文件。再次导入后，可再次编辑。
- b) 不可编辑：源代码不导出。再次导入后只可使用，无法编辑。

### 1) 导入

功能：导入已存在的函数功能块文件，供该PLC程序使用。



选中函数功能块，右键点击菜单“从硬盘导入函数功能块文件”，选择相应文件，按“确定”即可。

## 8-5. 功能块的编辑

例：将PLC寄存器中D0作为某圆的半径，数据格式为正数，算出的面积放在寄存器D10里，数据格式为浮点数。

- (1) 首先在“工程”工具栏里，新建一个函数功能块，在这里我们把它命名为 FUNC1，并且编辑C语言程序。
- (2) 编辑完之后，点击编译。

```
1  /*****
2  FunctionName: FUNC1
3  Description:
4
5  *****/
6  #include <include.h>
7  void FUNC1(unsigned int * M, unsigned short int * D)
8  {
9      float PI=3.14;
10     FLOATW[10]= WORD[0]*WORD[0]*PI
11     M[100]=1;
12 }
13
```

信息

错误列表 输出 查找结果

1.  
funcmp\FUNC1.c: In function 'FUNC1':  
funcmp\FUNC1.c:10:32: error: expected ';' before 'M' FLOATW[10]= WORD[0]\*WORD[0]\*PI ; M[100]=1; ~

根据编译信息列表内所显示的信息，我们可以查找修改 C 语言程序里的语法漏洞。在这里比较容易的发现程序中 `float PI=3.14;` 的后面缺少符号“;”。

当我们将程序修改后，再次进行编译。从列表信息里可以确认，在程序里面并没有语法错误。

```
1  /*****
2  FunctionName: FUNC1
3  Description:
4
5  *****/
6  #include <include.h>
7  void FUNC1(unsigned int * M, unsigned short int * D)
8  {
9      float PI=3.14;
10     FLOATW[10]= WORD[0]*WORD[0]*PI;
11     M[100]=1;
12 }
13
```

信息

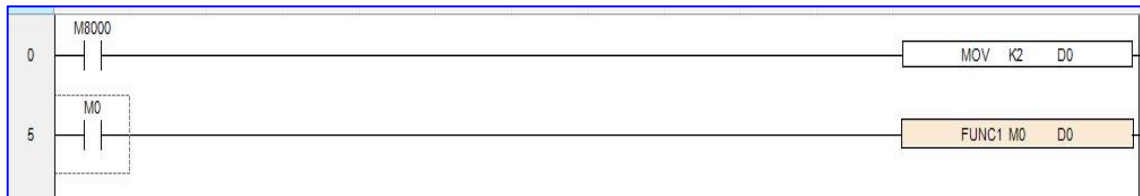
错误列表 输出 查找结果

1.  
funcmp\FUNC1.c: In function 'FUNC1':  
funcmp\FUNC1.c:10:32: error: expected ';' before 'M' FLOATW[10]= WORD[0]\*WORD[0]\*PI ; M[100]=1; ~

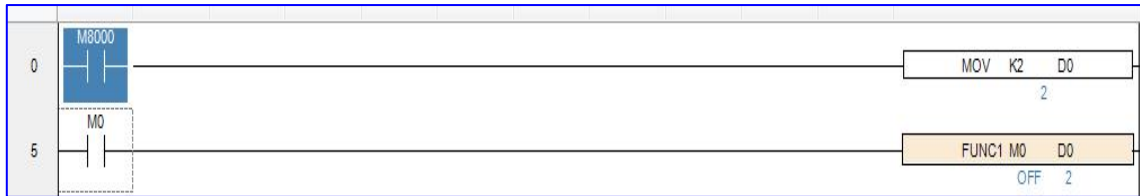
2. Compiled successfully!

- (3) 然后再编写 PLC 程序，赋值十进制数 2 到寄存器 D0 中，并调用函数功能块 FUNC1。如下所示：





(4) 然后将程序下载到PLC当中，运行PLC，并置位M0。



(5) 我们可以通过工具栏上的数据监控观察到 D20 的值变成了 12.56，说明赋值成功了。

D0	0.00		0.00	
D10	0.00		0.00	
D20	12.56		0.00	
D30	0.00		0.00	
D40	0.00		0.00	
D50	0.00		0.00	
D60	0.00		0.00	

## 8-6.程序举例

如果要进行一个“复杂运算”（包括加减运算，但是运算步骤很多），尤其是需要重复使用这个算法处理数据时候，使用C 函数功能块将非常方便。

例 1：用公式： $a= b/c+b*c+(c-3)*d$  完成运算。

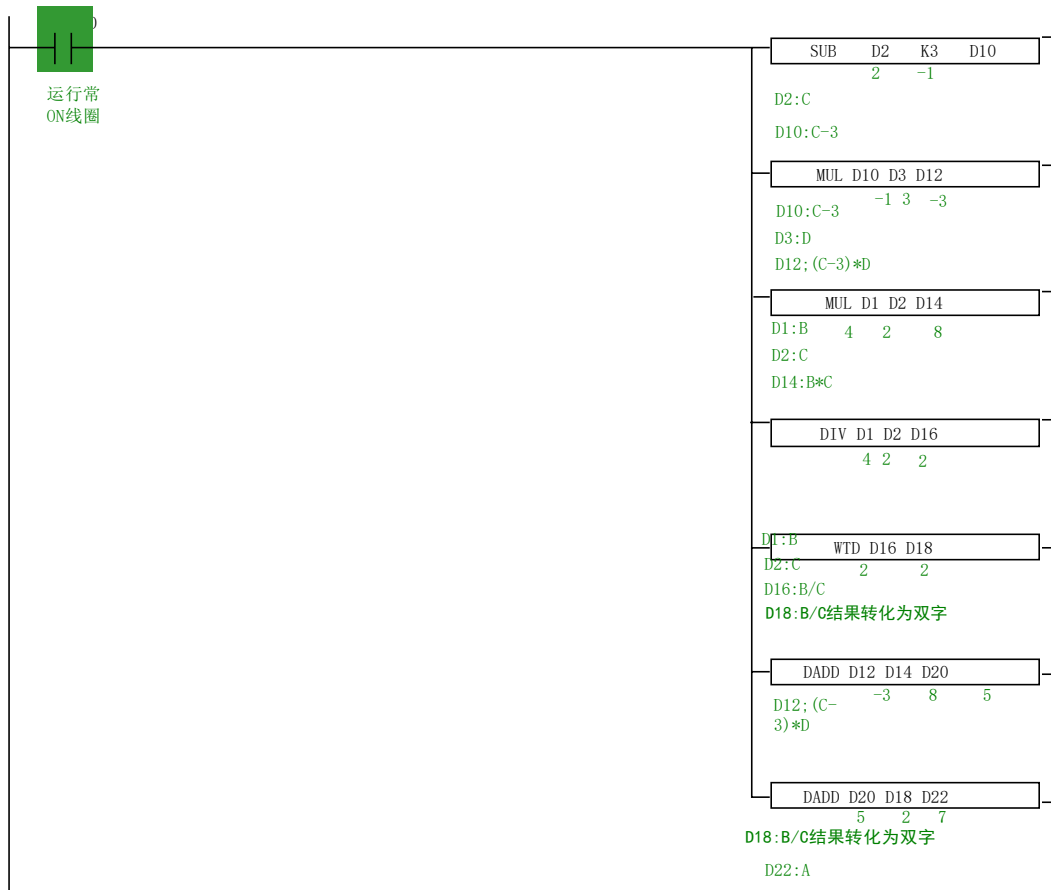
方法一：如果使用梯形图编写上述公式，处理步骤与程序如下：

首先求出  $c-3$

然后算出三个乘式的积

最后求和

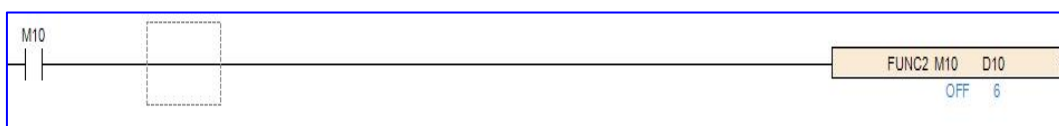
虽然只有以上三个步骤，但是梯形图只支持两个源操作数，所以必须分成多步求结果。



在上面梯形图运算中有几点要注意：

- (1) MUL 运算结果为双字，就是说 MUL D1 D2 D14[D15]，结果是存放在D14[D15]两个寄存器内。
- (2) DIV 运算结果分商和余数，即：DIV D1 D2 D16，商在 D16 中，余数在D17 中，所以如果运算有余数则精度就降低了，要得到精确的结果得用浮点数运算。
- (3) 在求和时，由于 D16 为商，是单字数据，所以加运算的时候得先统一数据类型，最终得到的结果存放在D22[D23]中。

方法二：使用C 函数写，梯形图程序如下：



首先，我们对上述 C 语言梯形图指令结构进行解析如下：

FUNC2	函数功能块的名称
D10	表示函数中 DWORD[10]为 D30, DWORD[12]为 D34 以此类推。如果 D10 位置寄存器为D20,则函数块中 DWORD[10]为 D40, D[12]为 D44 以此类推。
M10	表示函数中 M[0]为 M10, M[1]为 M11 以此类推。

C 语言部分内容如下:

```

1  /*****
2  FunctionName: FUNC2
3  Description:
4
5  *****/
6  #include <include.h>
7  void FUNC2(unsigned int * M, unsigned short int * D)
8  {
9      int a,b,c,d;
10     b=DWORD[10];
11     c=DWORD[12];
12     d=DWORD[14];
13     a=DWORD[16];
14     DWORD[16]=DWORD[10]/DWORD[12]+DWORD[10]*DWORD[12]+(DWORD[12]-3)*DWORD[14];
15
16 }
17

```

通过两种方法的对比可以看出, 通过 C 函数功能, 能够大大简化梯形图编程, 提高编程效率。上面的 C 函数运算和梯形图相似, 精度也不高, 如果要得到精确结果则使用浮点运算。

# 附录 1 特殊软元件一览表

附录 1 主要介绍 CZA 系列 PLC 中特殊用位软元件、字软元件功能用途，便于用户快速翻阅查找。

附录 1-1. 特殊辅助继电器一览

附录 1-2. 特殊数据寄存器一览

附录 1-3. 扩展模块地址一览

附录 1-4. 特殊内部寄存器一览

附录 1-5. 特殊寄存器总览

## 附录 1-1. 特殊辅助继电器一览

## PC 状态 (M8000-M8003)

地址号	功能	说明	
M8000	运行常 ON 线圈		PLC 运行时一直为ON
M8001	运行常 OFF 线圈		PLC 运行时一直为OFF
M8002	初始正向脉冲线圈		PLC 开始运行后第一个扫描周期为 ON
M8003	初始负向脉冲线圈		PLC 开始运行后第一个扫描周期为 OFF

## 时钟 (M8011-M8014)

地址号	功能	说明
M8011	以 10ms 的频率周期震荡	
M8012	以 100ms 的频率周期震荡	
M8013	以 1 秒钟的频率周期震荡	
M8014	以 1 分钟的频率周期震荡	

## 标志 (M8020-M8022)

地址号	功能	说明
M8020	零	加减运算结果为 0 时
M8021	借位	减法运算结果小于负的最大值
M8022	进位	加法运算结果发生进位时, 换位结果溢出发生时
M8029	Y0脉冲发完时动作	
M8030	Y1脉冲发完时动作	

### 中断 (M8050-M8059)

地址号	功能	说明
M8050 I000□	禁止输入中断X0	地址号中方框为中断输入引脚号 执行EI指令后，即使中断许可，但是当此 M 动作时，对应的输入中断将无法单独动作 例如：当 M8050 处于 ON 时，禁止中断 I000□
M8051 I010□	禁止输入中断X1	
M8054 I040□	禁止输入中断X4	
M8055 I050□	禁止输入中断X5	
M8059	禁止中断	禁止所有中断

**通讯 (M8120-M8131)**

地址号	名称
M8120	RS串口1, 485通讯(8位或16位 ASCII、HEX指令)
M8121	设置主从站1: 从站 0: 主站
M8211	RS串口1发送标志
M8123	RS串口1接受完成标志
M8124	
M8125	RS串口1超时判断
M8126	RS串口2, 232通讯 (8位或16位)
M8127	设置主从站1: 从站 0: 主站
M8128	RS串口2发送标志
M8129	RS串口2接受完成标志
M8130	
M8131	RS串口2超时判断



## 高速计数加减计数 (M8235-M8255)

高数计数号	加减标志位	说明
C235	M8235	驱动高速计数器时， 当加减标志位置ON 时，为减计数；当加 减标志位置OFF时， 为加计数。
C236	M8236	
C237	M8237	
C238	M8238	
C239	M8239	
C240	M8240	
C241	M8241	
C242	M8242	
C243	M8243	
C244	M8244	
C245	M8245	
C246	M8246	
C247	M8247	
C248	M8248	
C249	M8249	
C250	M8250	
C251	M8251	
C252	M8252	
C253	M8253	
C254	M8254	
C255	M8255	

## 附录 1-2. 特殊数据寄存器一览

## 时钟 (D8010-D8019)

地址号	功能	说明
D8010	当前扫描周期	0.1ms, ms 为单位
D8011	扫描时间的最小值	0.1ms, ms 为单位
D8012	扫描时间的最大值	0.1ms, ms 为单位
D8013	秒 (时钟)	0~59 (BCD 码形式)
D8014	分钟 (时钟)	0~59 (BCD 码形式)
D8015	小时 (时钟)	0~23 (BCD 码形式)
D8016	日 (时钟)	0~31 (BCD 码形式)
D8017	月 (时钟)	0~12 (BCD 码形式)
D8018	年 (时钟)	2000~2099 (BCD 码形式)
D8019	星期 (时钟)	0 (日) ~ 6 (六) (BCD 码形式)

D8013-D8019的时钟数据掉电保持。

## 脉冲输出 (D8340-D8351)

地址号	高频脉冲号	功能	说明
D8340	PULSE_1	累计脉冲个数低 16 位	作为Y0输出定位指令的当前值数据寄存器使用。对ZRN, DRVI, DRVA指令通用。
D8341		累计脉冲个数高 16 位	
D8350	PULSE_2	累计脉冲个数低 16 位	作为Y1输出定位指令的当前值数据寄存器使用。对ZRN, DRVI, DRVA指令通用。
D8351		累计脉冲个数高 16 位	

## 变址寄存器当前值 (D8180-D8195)

地址号	功能
D8180	Z0寄存器的内容
D8181	V0寄存器的内容
D8182	Z1寄存器的内容
D8183	V1寄存器的内容
D8184	Z2寄存器的内容
D8185	V2寄存器的内容
D8186	Z3寄存器的内容
D8187	V3寄存器的内容
D8188	Z4寄存器的内容
D8189	V4寄存器的内容
D8190	Z5寄存器的内容
D8191	V5寄存器的内容
D8192	Z6寄存器的内容
D8193	V6寄存器的内容
D8194	Z7寄存器的内容
D8195	V7寄存器的内容

**附录 1-3. 扩展模块地址一览**

以第一扩展模块为例说明（第 2~7 扩展模块地址号依次加 100）：

输入点	输出点
X100	Y100
X101	Y101
X102	Y102
X103	Y103
X104	Y104
X105	Y105
X106	Y106
X107	Y107

	<b>附录 1-4. 掉电保持区域设置一览</b>
--	---------------------------

## 1. 软元件掉电保持区域

	符号	进制	点数	起始	结束	锁存起始	结束
辅助继电器	M	10	3072	0	3071	500	1023
状态	S	10	1000	0	999	500	999
定时器	T	10	256	0	255		
计数器（16位）	C	10	200	0	199	100	199
计数器（32位）	C	10	56	200	255	220	255
数据寄存器	D	10	8000	0	7999	200	1999

## 附录 1-5. 特殊寄存器总览

CZA系列PLC特殊软元件一览表：

M元件	M元件的描述	D元件	D元件的描述
M8000	用户程序运行时置为ON状态	D8000	用户程序运行的监视定时器,初始值200ms
M8001	M8000状态取反	D8001	PLC型号(1~6)
M8002	上电初始脉冲	D8002	~
M8003	M8002状态取反	D8003	~
M8004	~	D8004	~
M8005	~	D8005	~
M8006	~	D8006	~
M8007	~	D8007	软件版本号
M8008	、	D8008	硬件版本号
M8009	~	D8009	错误代码
M8010	~	D8010	运行停止
M8011	10mS时钟周期的振荡时钟	D8011	拓展模块的个数
M8012	100mS时钟周期的振荡时钟	D8012	自更新版本号
M8013	1S时钟周期的振荡时钟	D8013	时钟秒(0~59)
M8014	1分钟时钟周期的振荡时钟	D8014	实时时钟分(0~59)
M8015	~	D8015	实时时钟小时(0~23)
M8016	~	D8016	实时时钟日(1~31)
M8017	~	D8017	实时时钟月(1~12)
M8018	~	D8018	实时时钟公历年(2000~2099)
M8019	~	D8019	实时时钟星期
M8020	运算零标志	D8020	X000~X007的输入滤波常数(REFF)0~60[默认10mS]
M8021	运算借位标志	D8021	数据上报时间间隔,单位:分钟
M8022	运算进位标志	D8022	数据上报起始地址

M8023	~	D8023	数据上报寄存器个数
M8024	M8024处于ON状态时, 传送方向取反	D8024	数据下行起始地址
M8025	~	D8025	数据下行最大存储个数
M8026	~	D8026	心跳时间间隔, 单位:分钟
M8027	~	D8027	当前扫描值
M8028	~	D8028	最小扫描时间
M8029	Y0设定脉冲发完后置位	D8029	最大扫描时间
M8030	Y1设定脉冲发完后置位	D8030	数据下行起始地址
M8031	~	D8031	~
M8032	~	D8032	~
M8033	~	D8033	~
M8034	~	D8034	~
M8035	~	D8035	~
M8036	~	D8036	~
M8037	强制停止命令	D8037	~
M8038	~	D8038	~
M8039	~	D8039	恒定扫描时间, 默认0, 以1mS为单位
M8040	~	D8040	~
M8041	~	D8041	~
M8042	~	D8042	~
M8043	~	D8043	~
M8044	~	D8044	~
M8045	~	D8045	~
M8046	~	D8046	~
M8047	~	D8047	~
M8048	ANS有报警器动作	D8048	~
M8049	ANS信号报警有效, [D8049有效]	D8049	最小报警器ANS
M8050	~	D8050	~
M8051	~	D8051	~
M8052	~	D8052	~



M8053	~	D8053	~
M8054	~	D8054	~
M8055	~	D8055	~
M8056	驱动 I06□中断禁止	D8056	~
M8057	驱动 I07□中断禁止	D8057	~
M8058	驱动 I08□中断禁止	D8058	~
M8059	驱动高数计数器中断禁止	D8059	~
M8060	~	D8060	~
M8061	PLC硬件错误	D8061	PLC硬件错误的错误代码序号
M8062	~	D8062	~
M8063	联机错误/232通讯错误	D8063	并行联机错误代码/RS232通信错误的代码序号
M8064	参数错误	D8064	参数错误代码序号
M8065	语法错误	D8065	语法错误的代码序号
M8066	回路错误	D8066	回路错误的代码序号
M8067	运算错误，为ON时表示出错；为OFF时表示正常。	D8067	运算错误的代码序号
M8068	运算错误锁存	D8068	锁存运算错误程序的步号
M8069	和D8069不一样，IO总线检测	D8069	M8065~M8067的错误发生的步号
M8070	~	D8070	~
M8071	~	D8071	~
M8072	~	D8072	~
M8073	~	D8073	~
M8074	~	D8074	~
M8075	~	D8075	~
M8076	~	D8076	~
M8077	~	D8077	~
M8078	~	D8078	~
M8079	~	D8079	~
M8080	~	D8080	~
M8081	~	D8081	~

M8082	~	D8082	~
M8083	~	D8083	~
M8084	~	D8084	~
M8085	~	D8085	~
M8086	~	D8086	~
M8087	~	D8087	~
M8088	~	D8088	~
M8089	~	D8089	~
M8090	~	D8090	~
M8091	~	D8091	~
M8092	~	D8092	~
M8093	~	D8093	~
M8094	~	D8094	~
M8095	~	D8095	~
M8096	~	D8096	~
M8097	~	D8097	~
M8098	~	D8098	~
M8099	~	D8099	~
M8100	~	D8100	~
M8101	~	D8101	~
M8102	~	D8102	~
M8103	~	D8103	~
M8104	~	D8104	~
M8105	~	D8105	~
M8106	~	D8106	~
M8107	~	D8107	~
M8108	~	D8108	~
M8109	~	D8109	~
M8110	~	D8110	~
M8111	~	D8111	~
M8112	~	D8112	~
M8113	~	D8113	~
M8114	~	D8114	~

M8115	~	D8115	~
M8116	~	D8116	~
M8117	~	D8117	~
M8118	~	D8118	~
M8119	~	D8119	~
M8120	RS串口1（8位或16位 ASCII,HEX指令）	D8120	~
M8121	~	D8121	~
M8122	RS串口1发送标志	D8122	~
M8123	RS串口1接收完成标志	D8123	~
M8124	~	D8124	~
M8125	RS串口1超时判断	D8125	~
M8126	RS串口2（8位或16位）	D8126	~
M8127	~	D8127	~
M8128	RS串口2发送标志	D8128	~
M8129	RS串口2接受完成标志	D8129	~
M8130	~	D8130	~
M8131	RS串口2超时判断	D8131	~
M8132	~	D8132	~
M8133	串口1三菱协议	D8133	~
M8134	串口2三菱协议	D8134	~
M8135	~	D8135	~
M8136	~	D8136	~
M8137	~	D8137	~
M8138	~	D8138	~
M8139	~	D8139	~
M8140	~	D8340	~
保留	~	D8341	~
M8142	~	D8350	~
M8143	~	D8351	~
M8144	~	D8144	~
M8145	~	D8145	~
M8146	~	D8146	~

M8147	~	D8147	~
M8148	~	D8148	~
M8149	~	D8149	~
M8150	~	D8150	~
M8151	~	D8151	~
M8152	~	D8152	~
M8153	~	D8153	~
M8154	~	D8154	~
M8155	~	D8155	~
M8156	~	D8156	~
M8157	~	D8157	~
M8158	~	D8158	~
M8159	~	D8159	~
M8160	~	D8160	串口1 参数配置
M8161	~	D8161	串口2 参数配置
M8162	~	D8162	串口1指令通讯间隔
M8163	~	D8163	串口2指令通讯间隔
M8164	~	D8164	串口1站号
M8165	~	D8165	串口2站号
M8166	~	D8166	~
M8167	~	D8167	~
M8168	~	D8168	~
M8169	~	D8169	~
M8170	~	D8170	~
M8171	~	D8171	~
M8172	~	D8172	~
M8173	~	D8173	~
M8174	~	D8174	~
M8175	~	D8175	~
M8176	~	D8176	~
M8177	~	D8177	~
M8178	~	D8178	~
M8179	~	D8179	~

M8180	~	D8180	Z0
M8181	~	D8181	V0
M8182	~	D8182	Z1
M8183	~	D8183	V1
M8184	~	D8184	Z2
M8185	~	D8185	V2
M8186	~	D8186	Z3
M8187	~	D8187	V3
M8188	~	D8188	Z4
M8189	~	D8189	V4
M8190	~	D8190	Z5
M8191	~	D8191	V5
M8192	~	D8192	Z6
M8193	~	D8193	V6
M8194	~	D8194	Z7
M8195	~	D8195	V7
M8196	~	D8196	~
M8197	~	D8197	~
M8198	~	D8198	~
M8199	~	D8199	~
M8200	当M8200被驱动后，对应的C200变成减型计数模式	D8200	~
M8201	当M8201被驱动后，对应的C201变成减型计数模式	D8201	拓展模块信息
M8202	当M8202被驱动后，对应的C202变成减型计数模式	D8202	拓展模块信息
M8203	当M8203被驱动后，对应的C203变成减型计数模式	D8203	拓展模块信息
M8204	当M8204被驱动后，对应的C204变成减型计数模式	D8204	拓展模块信息
M8205	当M8205被驱动后，对应的C205变成减型计数模式	D8205	拓展模块信息

	式		
M8206	当M8206被驱动后, 对应的C206变成减型计数模式	D8206	拓展模块信息
M8207	当M8207被驱动后, 对应的C207变成减型计数模式	D8207	拓展模块信息
M8208	当M8208被驱动后, 对应的C208变成减型计数模式	D8208	拓展模块信息
M8209	当M8209被驱动后, 对应的C209变成减型计数模式	D8209	拓展模块信息
M8210	当M8210被驱动后, 对应的C210变成减型计数模式	D8210	AD
M8211	当M8211被驱动后, 对应的C211变成减型计数模式	D8211	~
M8212	当M8212被驱动后, 对应的C212变成减型计数模式	D8212	AD
M8213	当M8213被驱动后, 对应的C213变成减型计数模式	D8213	~
M8214	当M8214被驱动后, 对应的C214变成减型计数模式	D8214	DA
M8215	当M8215被驱动后, 对应的C215变成减型计数模式	D8215	~
M8216	当M8216被驱动后, 对应的C216变成减型计数模式	D8216	DA
M8217	当M8217被驱动后, 对应的C217变成减型计数模式	D8217	~
M8218	当M8218被驱动后, 对应的C218变成减型计数模式	D8218	~
M8219	当M8219被驱动后, 对应的C219变成减型计数模式	D8219	~

	式		
M8220	当M8220被驱动后，对应的C220变成减型计数模式	D8220	~
M8221	当M8221被驱动后，对应的C221变成减型计数模式	D8221	~
M8222	当M8222被驱动后，对应的C222变成减型计数模式	D8222	~
M8223	当M8223被驱动后，对应的C223变成减型计数模式	D8223	~
M8224	当M8224被驱动后，对应的C224变成减型计数模式	D8224	~
M8225	当M8225被驱动后，对应的C225变成减型计数模式	D8225	~
M8226	当M8226被驱动后，对应的C226变成减型计数模式	D8226	~
M8227	当M8227被驱动后，对应的C227变成减型计数模式	D8227	~
M8228	当M8228被驱动后，对应的C228变成减型计数模式	D8228	~
M8229	当M8229被驱动后，对应的C229变成减型计数模式	D8229	~
M8230	当M8230被驱动后，对应的C230变成减型计数模式	D8230	~
M8231	当M8231被驱动后，对应的C231变成减型计数模式	D8231	~
M8232	当M8232被驱动后，对应的C232变成减型计数模式	D8232	~
M8233	当M8233被驱动后，对应的C233变成减型计数模式	D8233	~

	式		
M8234	当M8234被驱动后, 对应的C234变成减型计数模式	D8234	~
M8235	单相单输入增减计数标志位, 为ON时, C235减计数; 为OFF时, 为增计数	D8235	~
M8236	当M8236被驱动后, 对应的C236变成减型计数模式	D8236	~
M8237	当M8237被驱动后, 对应的C237变成减型计数模式	D8237	~
M8238	单相单输入增减计数标志位, 为ON时, C238减计数; 为OFF时, 为增计数	D8238	~
M8239	当M8239被驱动后, 对应的C239变成减型计数模式	D8239	~
M8240	当M8240被驱动后, 对应的C240变成减型计数模式	D8240	~
M8241	当M8241被驱动后, 对应的C241变成减型计数模式	D8241	~
M8242	当M8242被驱动后, 对应的C242变成减型计数模式	D8242	~
M8243	当M8243被驱动后, 对应的C243变成减型计数模式	D8243	~
M8244	当M8244被驱动后, 对应的C244变成减型计数模式	D8244	~
M8245	当M8245被驱动后, 对应的C245变成减型计数模式	D8245	~
M8246	~	D8246	~
M8247	~	D8247	~
M8248	~	D8248	~
M8249	~	D8249	~



M8250	~	D8250	~
M8251	~	D8251	~
M8252	~	D8252	Y0脉冲输出口PLRS指令 S型加减速时间, 单位: ms (低位)
M8253	~	D8253	Y0脉冲输出口PLRS指令 S型加减速时间, 单位: ms (高位)
M8254	~	D8254	Y1脉冲输出口PLRS指令 S型加减速时间, 单位: ms (低位)
M8255	~	D8255	Y1脉冲输出口PLRS指令 S型加减速时间, 单位: ms (高位)
M8340	[Y0]脉冲输出监控 (ON:BUSY/OFF:READ Y)	D8340	[Y0]定位指令脉冲数 (低 位)
M8341	[Y0]清除信号输出功能有 效	D8341	[Y0]定位指令脉冲数 (高 位)
M8342	[Y0]指定原点回归方向	D8342	[Y0]基速度
M8343	~	D8343	[Y0]最高速度 (低位)
M8344	~	D8344	[Y0]最高速度 (高位)
M8345	~	D8345	[Y0]爬行速度
M8346	~	D8346	[Y0]原点回归速度 (低 位)
M8347	~	D8347	[Y0]原点回归速度 (高 位)
M8348	[Y0]定位指令驱动中	D8348	[Y0]加速时间
M8349	[Y0]脉冲输出停止指令	D8349	[Y0]减速时间
MD8350	[Y1]脉冲输出监控 (ON:BUSY/OFF:READ Y)	DD8350	[Y1]定位指令脉冲数 (低 位)
M8351	[Y1]清除信号输出功能有 效	D8351	[Y1]定位指令脉冲数 (高 位)
M8352	[Y1]指定原点回归方向	D8352	[Y1]基速度
M8353	[Y1]正转限位	D8353	[Y1]最高速度 (低位)
M8354	[Y1]反转限位	D8354	[Y1]最高速度 (高位)

M8355	[Y1]近点DOG信号逻辑反转	D8355	[Y1]爬行速度
M8356	[Y1]零点信号逻辑反转	D8356	[Y1]原点回归速度（低位）
M8357	[Y1]中断信号逻辑反转	D8357	[Y1]原点回归速度（高位）
M8358	[Y1]定位指令驱动中	D8358	[Y1]加速时间
M8359	[Y1]脉冲输出停止指令	D8359	[Y1]减速时间

---

## 附录 2 应用指令一览

---

该部分将列出除基本指令以外的应用指令，以及这些指令与 CZA 系列 PLC 的适用关系。

指令 助记符	功能	章节
程序流程		
CJ	条件跳转	4-3-1
CALL	子程序调用	4-3-2
SRET	子程序返回	4-3-2
EI	允许中断	4-3-3
DI	禁止中断	4-3-3
IRET	中断返回	4-3-3

STL	流程开始	4-3-4
SET	打开指定流程，关闭所在流程	4-3-4
RET	流程结束	4-3-4
FOR	循环范围开始	4-3-5
NEXT	循环范围结束	4-3-5
FEND	主程序结束	4-3-6
END	程序结束	4-3-6
WDT	监控定时器	4-3-7
数据比较		
LD=	开始 (S1) = (S2) 时导通	4-4-1
LD>	开始 (S1) > (S2) 时导通	4-4-1
LD<	开始 (S1) < (S2) 时导通	4-4-1
LD<>	开始 (S1) ≠ (S2) 时导通	4-4-1
LD≤	开始 (S1) ≤ (S2) 时导通	4-4-1
LD≥	开始 (S1) ≥ (S2) 时导通	4-4-1
AND=	串联 (S1) = (S2) 时导通	4-4-2
AND>	串联 (S1) > (S2) 时导通	4-4-2
AND<	串联 (S1) < (S2) 时导通	4-4-2
AND<>	串联 (S1) ≠ (S2) 时导通	4-4-2
AND≤	串联 (S1) ≤ (S2) 时导通	4-4-2
AND≥	串联 (S1) ≥ (S2) 时导通	4-4-2
OR=	并联 (S1) = (S2) 时导通	4-4-3
OR>	并联 (S1) > (S2) 时导通	4-4-3
OR<	并联 (S1) < (S2) 时导通	4-4-3
OR<>	并联 (S1) ≠ (S2) 时导通	4-4-3
OR≤	并联 (S1) ≤ (S2) 时导通	4-4-3
OR≥	并联 (S1) ≥ (S2) 时导通	4-4-3
数据传送		
CMP	数据的比较	4-5-1
ZCP	数据的区间比较	4-5-2
MOV	传送	4-5-3
BMOV	数据块传送	4-5-4
SMOV	移位传送	4-5-5
FMOV	多点重复传送	4-5-6
ZRST	批次复位	4-5-7
SWAP	高低字节交换	4-5-8
XCH	两个数据交换	4-5-9
数据运算		
ADD	加法	4-6-1
SUB	减法	4-6-2
MUL	乘法	4-6-3
DIV	除法	4-6-4
INC	加 1	4-6-5
DEC	减 1	4-6-5

MEAN	求平均值	4-6-6
WAND	逻辑与	4-6-6
WOR	逻辑或	4-6-6
WXOR	逻辑异或	4-6-7
CML	取反	4-6-8
NEG	求负	4-6-9
数据移位		
ROL	循环左移	4-7-1
ROR	循环右移	4-7-1
RCL	带进位循环左移	4-7-2
PCR	带进位循环右移	4-7-2
SFWR	移位写入	4-7-3
SFRD	移位读出	4-7-3
SFTL	位左移	4-7-4
SFTR	位右移	4-7-5
WSFL	字左移	4-7-6
WSFR	字右移	4-7-7
数据转换		
FLT	16 位整数转浮点	4-8-1
DFLT	32 位整数转浮点	4-8-1
INT	浮点转整数	4-8-2

DECO	译码	4-8-7
ENCO	高位编码	4-8-8
浮点运算		
ECMP	浮点数比较	4-9-1
EZCP	浮点数区间比较	4-9-2
EADD	浮点数加法	4-9-3
ESUB	浮点数减法	4-9-4
EMUL	浮点数乘法	4-9-5
EDIV	浮点数除法	4-9-6
ESQR	浮点数开方	4-9-7
SIN	浮点数 SIN 运算	4-9-8
COS	浮点数 COS 运算	4-9-9
TAN	浮点数 TAN 运算	4-9-10
时钟		
TRD	时钟数据读取	4-10-1
TWR	时钟数据写入	4-10-2
TCMP	时钟数据比较	4-10-3
TZCP	时钟数据区间比较	4-10-4
TADD	时钟数据加法	4-10-5
TSUB	时钟数据减法	4-10-5
高速计数		
HSCR	高速计数比较复位	5-6-2
HSCS	高速计数比较置位	5-6-3
脉冲输出		
PLSY	无加减速时间变化的单向定量脉冲输出	6-2-1
PLSR	相对位置多段脉冲控制	6-2-2
PWM	脉宽调制	6-2-3
ZRN	原点回归	6-2-4
DRVI	相对位置单段脉冲控制	6-2-5
DRVA	绝对位置单段脉冲控制	6-2-6
PLSRS	带S型加减速的定向脉冲输出	6-2-7
PLSRC	任意曲线定向脉冲输出	6-2-8
PLSF	带方向可变频脉冲输出	6-2-9

MODBUS通讯		
COLR	线圈读	7-2-4
INPR	输入线圈读	7-2-4
COLW	单个线圈写	7-2-4
MCLW	多个线圈写	7-2-4
REGR	寄存器读	7-2-4
INRR	输入寄存器读	7-2-4
REGW	单个寄存器写	7-2-4
MRGW	多个寄存器写	7-2-4
自由格式		
RS	串行数据传送	7-3-2
其他		
FUNC*	C 语言函数调用	