CZK2/CZK3 系列

可编程控制器（PLC）编程手册

版本号	2.0.19
修订日期	2019 年 9 月

欣灵电气股份有限公司

注意事项

基本说明

- 感谢您选购了欣灵 CZK2/CZK3 系列可编程控制器。
- 本手册主要介绍 CZK2/CZK3 系列可编程控制器的硬件特性等内容。
- 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。
- 软件及编程方面的介绍，兼容三菱 GX Developer/GX WORKS2，请查阅相关手册。
- 请将本手册交付最终用户。

用户须知

- 只有具备一定的电气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。
- 将该产品与其它产品组合使用的时候，请确认是否符合有关规格、原则以及技术要求等。
- 使用该产品时，请自行确认是否符合要求以及安全，对于本产品的故障而可能引发机器故障或者损失时，请自行设置后备及安全功能。

责任申明

- 手册中的内容虽然已经过仔细的核对，但差错难免，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，请与欣灵电气股份有限公司联系。

- 电话/传真：0577-62735555
- 地址：乐清经济开发区纬十九路 328 号
- 网址：http://www.c-lin.cn

前言

以下将介绍本手册的内容构成、手册的适用范围、手册的约定俗成、相关手册介绍、获取手册的途径以及修订的内容。

手册的内容构成

本手册涉及 CZK2/CZK3 系列可编程控制器的指令的应用，主要介绍 CZK2/CZK3 系列可编程控制器的基本指令、应用指令、特殊指令等，同时记载了编程中的要点、原则等。

手册的适用范围

本手册为 CZK2/CZK3 系列 PLC 用户编程手册，该手册涉及以下产品信息：

➢主机

0806、1410、1814、2416的所有机型。➢

混合主机

0810、1810的所有机型。

➢输入输出扩展以及单机

1、扩展：CZM-E16X、CZM-E16YR、CZM-E16YT、CZM-E8X8YR、CZM-E8X8YT
2、单机：CZM-S16X、CZM-S16YR、CZM-S16YT、CZM-S8X8YR、CZM-S8X8YT

➢模拟量扩展以及单机

1、2 通道模拟输入/输出扩展以及单机
扩展：CZM-E2WT　单机：CZM-S2WT
2、4 通道模拟输入/输出扩展以及单机
1）扩展：CZM-E4AD、CZM-E4P(T)、CZM-E4N(R)、CZM-E4KT、CZM-E4DA
2）单机：CZM-S4AD、CZM-S4P(T)、CZM-S4N(R)、CZM-S4KT、CZM-S4DA
3、8 通道模拟输入/输出扩展模块
CZM-E8AD、CZM-E8P(T)、CZM-E8N(R)、CZM-E8DA

➢混合扩展以及单机

1、4 通道模拟输入/输出混合扩展以及单机（电压、电流、温度、电阻尺）：
1）扩展：CZM-E2AD2P(T)、CZM-E2AD2N(R)、CZM-E2AD2KT
CZM-E2DA2AD、CZM-E2DA2P(T)、CZM-E2DA2N(R)、CZM-E2DA2KT
2）单机：CZM-S2AD2P(T)、CZM-S2AD2N(R)、CZM-S2AD2KT
CZM-S2DA2AD、CZM-S2DA2P(T)、CZM-S2DA2N(R)、CZM-S2DA2KT

2、8 通道模拟输入/输出混合扩展模块（电压、电流、温度）：
CZM-E4AD4P(T)、CZM-E4AD4N(R)、CZM-E4DA4AD、CZM-E4DA4P(T)
CZM-E4DA4N(R)、CZM-E4DA2AD2P(T)、CZM-E4DA2AD2N(R)、CZM-E2DA6AD
CZM-E2DA6P(T)、CZM-E2DA6N(R)、CZM-E2DA2AD4P(T)

手册的约定俗成

限于篇幅，手册中可能使用一定的简称来代替原有的名称，现将这些可能涉及到的名称列于下表，以便对照。

简称	说明
CZK2/CZK3 系列 PLC	CZK2/CZK3 系列可编程控制器的总称
基本单元或本体	CZK2/CZK3 系列可编程控制器的基本单元的简称
扩展模块	CZK2/CZK3 系列可编程控制器的全部扩展模块的总称
输入输出扩展或 I/O 扩展	CZK2/CZK3 系列可编程控制器的全部输入输出扩展模块的简称
模拟量扩展	CZK2/CZK3 系列可编程控制器的全部模拟量扩展模块的简称
外围设备	编程软件、人机界面、网络模块的总称
编程软件	兼容三菱 GX Developer/GX WORKS2

相关手册

本手册只涉及 CZK2/CZK3 系列 PLC 的指令方面的情况，编程软件兼容三菱 GX Developer/GX WORKS2，其他方面的应用，如安装、扩展设备以及选型等，请查阅相关手册资料。

《CZK2/CZK3 系列可编程控制器安装使用手册》、《CZK2/CZK3 系列可编程控制器扩展模块用户手册》

《CZK2/CZK3 系列可编程控制器(PLC)硬件手册》、《CZK2/CZK3 系列 PLC 用户手册-高精度称重篇》

《CZK3 系列 PLC 用户手册-MODBUS 通讯篇》以及《CZK2/CZK3 系列可编程控制器选型手册》

获取手册的途径

1、获取手册印刷版可向购买本系列产品供应商索取。

2、获取手册电子文档（PDF 文件），可从欣灵电气股份有限公司网站（www.c-lin.cn）查询下载。

目录

注意事项 2

基本说明 2
用户须知 2
责任申明 2
联系方式 2

前言 3

手册的内容构成 3
手册的适用范围 3
手册的约定俗成 4
相关手册 4
获取手册的途径 4

第一章 编程方式概述 13

1-1 可编程控制器的编程语言 13
1-2 CZK2/CZK3 系列指令说明 14

第二章 软元件的作用和功能 23

2-1 软元件编号一览 23
2-2 输入输出继电器【X,Y】 25
2-3 辅助继电器【M】 25
2-4 状态【S】 26
2-5 定时器【T】 28
2-6 计数器【C】 31
2-8 数据寄存器、文件寄存器【D】 47
2-9 文件寄存【R】、扩展文件寄存器【ER】 52
2-10 变址寄存器【V，Z】 57
2-11 指针【P】,【I】 58

第三章指令的软元件・常数的指定方法 63

3-1 可编程控制器处理的数据(8 进制数/10 进制数/16 进制数/实数) 63
3-2 常数 K, H，E (10 进制数/16 进制数/实数)的指定 66
3-3 字符串 66
3-4 位的位数指定(Kn□***) 68
3-5 字软元件的位指定(D□.b) 69
3-6 缓冲存储器的直接指定(U□\G□) 70
3-6 变址修正 71

第四章编程前须知 76

4-1 指令说明的阅读方法 76
4-2 编程方面的基本注意事项 78
4-3 输入输出处理，响应延迟 81
4-5 应用指令的一般通则 82
1 应用指令的表示和执行形式 82
2 一般标志位的使用 84
3 运算出错标志位的使用 85
4 扩展功能用标志位的使用 86
5 指令使用次数的限制 86

第五章 基本指令 88

5-1 LD 系列连接到母线的指令 88
LD 取指令 88
LDI 取反指令 88
LDP 取脉冲上升沿指令 89
LDF 取脉冲下降沿指令 89
5-3 AND 系列串联连接的指令 90
AND 与指令 90
ANI 与反转指令 90
ANDP 与脉冲上升沿指令 91
ANDF 与脉冲下降沿指令 91
5-4 OR 系列并联连接的指令 92
OR 或指令 92
ORI 或反转指令 92
ORP 或脉冲上升沿指令 93
ORF 或脉冲下降沿指令 93
5-5 回路块指令 94
ORB 回路块或指令 94
ANB 回路块与指令 94
5-6 线圈控制指令 95
OUT 输出指令 95
SET 置位指令 95
RST 复位指令 96
5-7 脉冲输出指令 96
PLS 上升沿脉冲输出指令 96
PLF 下降沿脉冲输出指令 97
5-8 主控指令 97
MC 主控指令 97
MCR 主控复位指令 98
5-9 堆栈指令 99
MPS 进栈指令 99
MRD 读栈指令 99
MPP 出栈指令 99

5-10 其他基本指令 100
INV 取反指令 100
NOP 空操作指令 100
END 结束指令 100

第六章 梯形图步进指令 101

6-1 步进梯形图指令 101
STL 步进梯形图指令 101
RET 返回指令 102

第七章 应用指令 103

7-1 程序流程指令 103
FNC 00 - CJ 条件跳转指令 103
FNC01 - CALL 子程序调用指令 104
FNC02 - SRET 子程序返回指令 104
FNC03 - IRET：中断返回 105
FNC04 - EI：允许中断 106
FNC05 - DI：禁止中断 106
FNC06 - FEND 主程序结束指令 106
FNC07- WDT：看门狗定时器 107
FNC08 - FOR 循环范围开始指令 108
FNC09 - NEXT 循环范围结束指令 108
7-2 传送与比较指令 109
FNC10 - CMP 比较指令 109
FNC11 - ZCP 区域比较指令 109
FNC12 - MOV 传送指令 110
FNC13 - SMOV 位移动指令 110
FNC14 - CML 反相传送指令 111
FNC15 - BMOV 成批传送指令 112
FNC16 - FMOV 多点传送指令 112
FNC17 - XCH 交换指令 113
FNC18 - BCD：BCD 转换 113
FNC19 - BIN：BIN 转换 114
7-3 四则逻辑运算指令 115
FNC20 - ADD：BIN 加法运算 115
FNC21 - SUB：BIN 减法运算 116
FNC22 - MUL：BIN 乘法运算 117
FNC23 - DIV：BIN 除法运算 117
FNC24 - INC：BIN 加一 118
FNC25 - DEC：BIN 减一 119
FNC26 - WAND：逻辑与 119
FNC27 - WOR：逻辑或 120

FNC28 - WXOR：逻辑异或 120
FNC29 - NEG：补码 121
7-4 循环移位 - FNC30~FNC39 122
FNC30 - ROR：循环右移 122
FNC31 - ROL：循环左移 123
FNC32 - RCR：带进位循环右移 123
FNC33 - RCL：带进位循环左移 124
FNC34 - SFTR：位右移 124
FNC35 - SFTL：位左移 125
FNC36 - WSFR 字右移 125
FNC37 - WSFL：字左移 126
FNC38 - SFWR：移位写入【先入先出/先入后出控制用】 126
FNC39 - SFRD：移位读出【先入先出控制用】 127
7-5 数据处理指令 127
FNC40 - ZRST: 成批复位 127
FNC41 - DECO：译码 128
FNC42 - ENCO：编码 128
FNC43 - SUM：ON 位数 129
FNC44 - BON：ON 位的判断 130
FNC45 - MEAN：平均值 130
FNC46 - ANS：信号报警器置位 131
FNC47 - ANR：信号报警器复位 131
FNC48 - SQR：BIN 开方运算 132
FNC49 - FLT：BIN 整数转 2 进制浮点数 132
7-6 高速处理 - FNC50~FNC59 133
FNC50 - REF：输入输出刷新 133
FNC51 - REFF：输入刷新【带滤波设定】 133
FNC52 - MTR：矩阵输入 133
FNC53 - HSCS：比较置位【高速计数器用】 135
FNC54 - HSCR：比较复位【高速计数器用】 135
FNC55 - HSZ：区间比较【高速计数器用】 136
FNC56 - SPD：脉冲密度 137
FNC57 - PLSY：脉冲输出指令 139
FNC58 - PWM：脉宽调制 140
FNC59 - PLSR：带加减速的脉冲输出 140
7-7 方便指令 - FNC60~FNC69 141
FNC60 - IST：初始化状态 141
FNC61 - SER：数据检索 141
FNC62 - ABSD：凸轮控制绝对方式 144
FNC63 - INCD：凸轮控制相对方式 147
FNC64 - TTMR：示教定时器 149
FNC65 - STMR：特殊定时器 151

FNC66 - ALT：交替输出 153
FNC67 - RAMP：斜坡信号 155
FNC68 - ROTC：旋转工作台控制 156
FNC69 - SORT：数据排序 158
7-8 外围设备 I/O - FNC70~FNC79 160
FNC70 - KEY：数字键输入 160
FNC71 - HEY：16 进制输入 160
FNC72 - DSW：数字开关 160
FNC73 - SEGD：七段译码 160
FNC74 - SEGL：七段码时分显示 161
FNC75 - ARWS：箭头开关 161
FNC76 - ASC：ASCII 数据输入 161
FNC77 - PR：ASCII 码打印 164
FNC78 - FROM：BFM 读出 165
FNC79 - TO：BFM 写入 166
7-9 外围设备 SER - FNC80~FNC89 167
FNC80 - RS: 串行数据传送 167
FNC81 - PRUN：8 进制位传送 168
FNC82 - ASCI：HEX 转 ASCII 168
FNC83 - HEX：ASCII 转 HEX 169
FNC84 - CCD：校验码 170
FNC87 - RS2：串行数据传送 2 171
FNC88 - PID：PID 运算 173
7-10 数据传送 2 - FNC100~FNC109 174
FNC102 - ZPUSH 变址寄存器的成批保存 174
FNC103 - ZPOP 变址寄存器的恢复 174
7-11 浮点数指令 - FNC110~FNC139 174
FNC110 - ECMP：2 进制浮点比数 174
FNC111 - EZCP：2 进制浮点数区间比较 175
FNC112 - EMOV：2 进制浮点数据传送 175
FNC116 - ESTR：2 进制浮点数转字符串 176
FNC117 - EVAL：字符串转 2 进制浮点数 182
FNC118 - EBCD： 2 进制浮点转十进制浮点数 185
FNC119 - EBIN：10 进制浮点数转 2 进制浮点数 186
FNC120 - EADD：2 进制浮点数加法运算 186
FNC121 - ESUB：2 进制浮点数减法运算 187
FNC122 - EMUL：2 进制浮点数乘法运算 187
FNC123 - EDIV：2 进制浮点数除法运算 188
FNC124 - EXP：2 进制浮点数指数运算 188
FNC125 - LOGE：2 进制浮点数自然对数运算 190
FNC126 - LOGE10：2 进制浮点数常用对数运算 191
FNC127 - ESQR：2 进制浮点数开方运算 192

FNC128 - ENEG：2 进制浮点数符号翻转 192
FNC129 - INT：2 进制浮点数转 BIN 整数 193
FNC130 - SIN：2 进制浮点数数 SIN 运算 193
FNC131 - COS: 2 进制浮点数数 COS 运算 194
FNC132 - TAN：2 进制浮点数 TAN 运算 194
FNC133 - ASIN：2 进制浮点数 SIN-1 运算 195
FNC134 - ACOS：2 进制浮点数 COS-1 运算 196
FNC135 - ATAN：2 进制浮点数 TAN-1 运算 197
FNC136 - RAD：2 进制浮点数角度转弧度 198
FNC137 - DEG：2 进制浮点数弧度转角度 200
7-12 数据处理 2 - FNC140~FNC149 201
FNC140 - WSUM：算出数据合计值 201
FNC141 - WTOB：字节单位的数据分离 202
FNC142 - BTOW：字节单位的数据结合 204
FNC143 - UNI：16 位数据的 4 位结合 206
FNC144 - DIS：16 位数据的 4 位分离 207
FNC147 - SWAP 高低字节互换 208
FNC149 - SORT2：数据排序 2 208
7-13 定位控制指令 - FNC150~FNC159 213
FNC150 - DSZR：带 DOG 搜索的原点回归 213
FNC151 - DVIT：中断定位 214
FNC152 - TBL：表格设定定位 214
FNC155 - ABS：读出 ABS 当前值 215
FNC156 - ZRN：原点回归 215
FNC157 - PLSV：可变速脉冲输出 216
FNC158 - DRVI：相对定位 216
FNC159 - DRVA：绝对定位 217
7-14 时钟连算指令 - FNC160~FNC169 218
FNC160 - TCMP：时钟数据比较 218
FNC161 - TZCP：时钟数据区间比较 218
FNC162 - TADD：时钟数据加法运算 219
FNC163 - TSUB：时钟数据减法运算 220
FNC164 - HTOS：时、分及秒数据的秒转换 220
FNC165 - STOH：秒数据的【时、分及秒】转换 222
FNC166 - TRD：读出时钟数据 224
FNC167 - TWR：写入时钟数据 224
FNC169 - HOUR：计时表 224
7-15 外部设备 - FNC170~FNC179 225
FNC170 - GRY：格雷码的转换 225
FNC171 - GBIN：格雷码的逆转换 225
FNC176 - RD3A：模拟量模块的读出 225
FNC177 - WR3A：模拟量模块的写入 225

7-16 其他指令 - FNC181~FNC189 225
FNC182 - COMRD：读出软元件的注释数据 225
FNC184 - RND：产生随机数 225
FNC186 - DUTY：产生定时脉冲 225
FNC188 - CRC：CRC 运算 228
FNC189 - HCMOV：高速计算器传送 231
7-17 数据块处理 - FNC190~FNC199 234
FNC192 - BK+：数据块加法运算 234
FNC193 - BK-：数据块减法运算 236
FNC194 - BKCMP=：数据块的比较 S1. = S2. 239
FNC195 - BKCMP>：数据块加法运算 S1. > S2. 239
FNC196 - BKCMP<：数据块加法运算 S1. < S2. 239
FNC197 - BKCMP<>：数据块加法运算 S1. <> S2. 239
FNC198 - BKCMP<=：数据块加法运算 S1. <= S2. 239
FNC199 - BKCMP>=：数据块加法运算 S1. >= S2. 239
7-18 数据块处理 - FNC200~FNC209 243
FNC200 - STR：BIN 转字符串 243
FNC201 - VAL：字符串转 BIN 247
FNC202 - $+：字符串的结合 251
FNC203 - LEN：检测出字符串的长度 253
FNC204 - RIGHT：从符串的右侧开始取出 254
FNC205 - LEFT：从符串的左侧开始取出 254
FNC206 - MIDR：从字符串中任意取出 254
FNC207 - MIDW：字符串中的任意替换 254
FNC208 - INSTR：字符串的检索 255
FNC209 - $MOV：字符串的传送 255
7-19 数据块处理 3 - FNC210~FNC219 255
FNC210 - FDEL：数据表的数据删除 255
FNC211 - FINS：数据表的数据插入 255
FNC212 - POP：读取后入的数据【先入后出控制器用】 255
FNC213 - SFR：16 位数据 n 位右移（带进位） 255
FNC214 - SFL：16 位数据 n 位左移（带进位） 255
7-20 触点比较指令 - FNC220~FNC249 255
FNC224~FNC230 - LD=,>,<,<>,<=,>= ：母线连接的触点比较 255
FNC232~FNC238 - AND=,>,<,<>,<=,>= ：串联连接的触点比较指令 256
FNC240~FNC246 - OR=,>,<,<>,<=,>= ：并联连接的触点比较指令 257
7-21 数据表处理指令 - FNC250~FNC269 259
FNC256 - LIMIT：上下限限位控制 259
FNC257 - BAND：死区控制 261
FNC258 - ZONE：区域控制 263
FNC259 - SCL：定坐标（不同点坐标数据） 266
FNC260 - DABBIN：10 进制 ASCII 转 BIN 270
FNC261 - BINDA：BIN 转 10 进制 ASCII 的转换 270

7-22 外部设备通信（变频器）- FNC270~FNC274 270
FNC270 - IVCK：变频器的运行监控 270
FNC271 - IVDR：变频器的运行控制 270
FNC272 - IVRD：变频器的参数读取 270
FNC273 - IVWR：变频器的参数写入 270
FNC274 - IVBWR：变频器的参数成批写入 270
7-23 数据传送 3 - FNC278~FNC279 270
FNC278 - RBFM：BFM 分割读取 270
FNC279 - WBFM：BFM 分割写入 270
7-24 高速处理 2 - FNC280~FNC289 270
FNC280 - HSCT：高速计数器表比较 270
7-25 扩展文件寄存器控制 - FNC290~FNC299 271
FNC290 - LOADR：读出扩展文件寄存器 271
FNC291 - SAVER：成批写入扩展文件寄存器 271
FNC292 - INITR：扩展寄存器的初始化 271
FNC293 - LOGR：登录到扩展寄存器 271
FNC294 - RWER：扩展文件寄存器的删除・写入 271
FNC295 - INITER：扩展文件寄存器的初始化 271

第一章 编程方式概述

在本章中，说明了关于 CZK2/CZK3 系列可编程控制器（PLC）的基本事项。

1-1 可编程控制器的编程语言

与 CZK2/CZK3 系列可编程控制器（PLC）编程的有关特点有以下几点。

➢编程语言的种类

CZK2/CZK3 系列编程控制器支持下面 3 种编程语言。

1、指令表编程，形成程序基础的指令表编程方式

1）特点：指令表编程方式，就是通过“LD”,“AND”,“OUT”等指令语言输入顺控指令的方式。该方式是顺控程序中基本的输入形态。

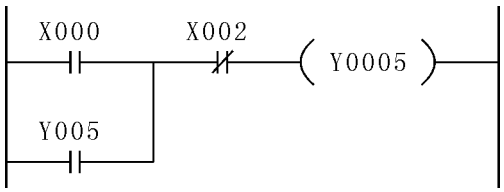
2）列表显示实例

步	指令	软元件编号
0000	LD	X000
0001	OR	Y001
0002	ANI	X002
0003	OUT	Y001
...	...	...

2、梯形图编程，在图示的画面上梯形图符号的梯形图编程方式

1）特点：梯形图编程方式，就是使用顺控符号和软元件编号在图示的画面上画顺控梯形图的方式。由于顺控回路是通过触点符号和线圈符号来表现的，所以程序的内容更加容易理解。即使在梯形图显示的状态下也可以执行可编程控制器的运行监控。

2）梯形图显示实例



3、梯 SFC【STL 步进梯形图】编程，可以根据机械的动作流程进行顺控设计的输入方式

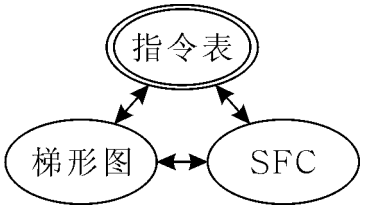
1）特点：SFC（顺序功能图）程序就是根据机械的动作流程设计顺控的方式。

2）SFC 程序和其他程序形式的互换性

可以相互转换的指令表程序及梯形图程序，若依照一定的规则编制，就可以倒过来转换成 SFC 图。

➢程序的互换性

采用上述的 3 种方法制作的顺控程序，都通过指令(指令表编程时的内容)保存到可编程控制器的程序内存中。使用如下图所示的各种输入方式编制的程序都可以相互转换后进行显示、编辑。



1-2 CZK2/CZK3 系列指令说明

➢基本指令说明

指令助记符	名称	功能	CZK2 系列	CZK3 系列	页码
LD	取	a 触点的逻辑运算开始	●	●	
LDI	取反转	a 触点的逻辑运算开始	●	●	
LDP	取脉冲上升沿	检出上升沿逻辑运算开始	●	●	
LDF	取脉冲下降沿	检出下降沿逻辑运算开始	●	●	
AND	与	串联 a 触点	●	●	
ANI	与反转	串联 b 触点	●	●	
ANDP	与脉冲上升沿	上升沿检出串联连接	●	●	
ANDF	与脉冲下降沿	下降沿检出串联连接	●	●	
OR	或	并联 a 触点	●	●	
ORI	或反转	并联 b 触点	●	●	
ORP	或脉冲上升沿	上升沿检出并联连接	●	●	
ORF	或脉冲下降沿	下降沿检出并联连接	●	●	
ORB	回路块或	串联回路块的并联连接	●	●	
ANB	回路块与	并联回路块的串联连接	●	●	
OUT	输出	线圈驱动	●	●	
SET	置位	线圈驱动保持	●	●	
RST	复位	线圈驱动清除	●	●	
PLS	上升沿脉冲	上升沿检出输出	●	●	
PLF	下降沿脉冲	下降沿检出输出	●	●	
MC	主控	公共串联点的连接线圈指令	●	●	
MCR	主控复位	公共串联点的清除指令	●	●	
MPS	进栈	压入堆栈，运算储存	●	●	
MRD	读栈	读取堆栈，储存读出	●	●	
MPP	出栈	弹出堆栈，储存读出与复位	●	●	
INV	反转	运算结果取反	●	●	
NOP	空操作	无动作	●	●	
END	结束	顺序控制程序结束	●	●	

➢梯形图步进指令说明

指令助记符	名称	功能	CZK2 系列	CZK3 系列	页码
STL	步进梯形图	步进梯形图开始	●	●	
RET	返回	步进梯形图结束	●	●	

➢应用指令说明

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
程序流程						
0	CJ	条件跳转	CJ Pn	●	●	
1	CALL	子程序调用	CALL Pn	●	●	
2	SRET	子程序返回	SRET	●	●	
3	IRET	中断返回	IRET	●	●	
4	EI	终端许可	EI	●	●	
5	DI	中断禁止	DI	●	●	
6	FEND	主程序结束	FEND	●	●	
7	WDT	监控定时器	WDT	●	●	
8	FOR	循环范围开始	FOR S	●	●	
9	NEXT	循环范围终了	NEXT	●	●	
传送与比较						
10	CMP	比较	CMP S1 S2 D	●	●	
11	ZCP	区域比较	ZCP S1 S2 S D	●	●	
12	MOV	传送	MOV S D	●	●	
13	SMOV	移位传送	SMOV S m1 m2 D n	●	●	
14	CML	倒转传送	CML S D	●	●	
15	BMOV	一并传送	BMOV S D n	●	●	
16	FMOV	多点传送	FMOV S D n	●	●	
17	XCH	交换	XCH D1 D2	●	●	
18	BCD	BCD 转换	BCD S D	●	●	
19	BIN	BIN 转换	BIN S D	●	●	
四则逻辑运算						
20	ADD	BIN 加法	ADD S1 S2 D	●	●	
21	SUB	BIN 减法	SUB S1 S2 D	●	●	
22	MUL	BIN 乘法	MUL S1 S2 D	●	●	
23	DIV	BIN 除法	DIV S1 S2 D	●	●	
24	INC	BIN 加一	INC D	●	●	
25	DEC	BIN 减一	DEC D	●	●	
26	WAND	逻辑字与	WAND S1 S2 D	●	●	
27	WOR	逻辑字或	WOR S1 S2 D	●	●	
28	WXOR	逻辑字异或	WXOR S1 S2 D	●	●	
29	NEG	求补码	NEG D	●	●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
			循环移位			
30	ROR	循环右移	ROR D n	●	●	
31	ROL	循环左移	ROL D n	●	●	
32	RCR	带进位循环右移	RCR D n	●	●	
33	RCL	带进位循环左移	RCL D n	●	●	
34	SFTR	位右移	SFTR S D n1 n2	●	●	
35	SFTL	位左移	SFTL S D n1 n2	●	●	
36	WSFR	字右移	WSFR S D n1 n2	●	●	
37	WSFL	字左移	WSFL S D n1 n2	●	●	
38	SFWR	移位写入	SFWR S D n	●	●	
39	SFRD	移位读出	SFRD S D n	●	●	
			数据处理			
40	ZRST	批次复位	ZRST D1 D2	●	●	
41	DECO	译码	DECO S D n	●	●	
42	ENCO	编码	ENCO S D n	●	●	
43	SUM	ON 位数	SUM S D	●	●	
44	BON	ON 位数判定	BON S D n	●	●	
45	MEAN	平均值	MEAN S D n	●	●	
46	ANS	信号报警器置位	ANS S m D	●	●	
47	ANR	信号报警器复位	ANR	●	●	
48	SQR	BIN 开方	SQR S D	●	●	
49	FLT	BIN 整数-2 进制浮点数转换	FLT S D	●	●	
			高速处理			
50	REF	输入输出刷新	REF D n	●	●	
51	REFF	滤波器调整	REFF n	●	●	
52	MTR	矩阵输入	MTR S D1 D2 n	●	●	
53	HSCS	比较置位(高速计数器)	HSCS S1 S2 D	●	●	
54	HSCR	比较复位(高速计数器)	HSCR S1 S2 D	●	●	
55	HSZ	区间比较(高速计数器)	HSZ S1 S2 S D	●	●	
56	SPD	脉冲密度	SPD S1 S2 D	●	●	
57	PLSY	脉冲输出	PLSY S1 S2 D	●	●	
58	PWM	脉冲调制	PWM S1 S2 D	●	●	
59	PLSR	带加减速的脉冲输出	PLSR S1 S2 S3 D	●	●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
方便指令						
60	IST	初始化状态	IST S D1 D2	●	●	
61	SER	数据查找	SER S1 S2 D n	●	●	
62	ABSD	凸轮控制(绝对方式)	ABSD S1 S2 D n	●	●	
63	INCD	凸轮控制(增量方式)	INCD S1 S2 D n	●	●	
64	TTMR	示教定时器	TTMR D n	●	●	
65	STMR	特殊定时器	STMR S m D	●	●	
66	ALT	交替输出	ALT D	●	●	
67	RAMP	斜坡信号	RAMP S1 S2 D n	●	●	
68	ROTC	旋转工作台控制	ROTC S m1 m2 D	●	●	
69	SORT	数据排列	SORT S m1 m2 D n	●	●	
外围设备 I/O						
70	TKY	数字键输入	TKY S D1 D2	●	●	
71	HKY	16 键输入	HKY S D1 D2 D3	●	●	
72	DSW	数字式开关	DSW S D1 D2 n	●	●	
73	SEGD	7 段详码	SEGD S D	●	●	
74	SEGL	7 段码按时间分割显示	SEGL S D n	●	●	
75	ARWS	箭头开关	ARWS S D1 D2 n	●	●	
76	ASC	ASCII 码变换	ASC S D	●	●	
77	PR	ADCII 码打印输出	PR S D	●	●	
78	FROM	BFM 读出	FROM m1 m2 D n	●	●	
79	TO	BFM 写入	TO m1 m2 S n	●	●	
外围设备 SER						
80	RS	串行数据传输	RS S m D n	●	●	
81	PRUN	8 进制位传送	PRUN S D	●	●	
82	ASCI	HEX-ASCII 转换	ASCI S D n	●	●	
83	HEX	ASCII-HEX 转换	HEX S D n	●	●	
84	CCD	校验码	CCD S D n	●	●	
85	VRRD	电位器读出	VRRD S D	●	●	
86	VRSC	电位器刻度	VRSC S D	●	●	
87	RS2	串行数据传送 2	RS2 S m D n n1		●	
88	PID	PIC 运算	PID S1 S2 S3 D	●	●	
数据传送 2						
102	ZPUSH	变址寄存器的批次躲避	ZPUSH D		●	
103	ZPOP	变址寄存器的恢复	ZPOP D		●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
浮点数						
110	ECMP	2 进制浮点数比较	ECMP S1 S2 D	●	●	
111	EZCP	2 进制浮点数区间比较	EZCP S1 S2 S D	●	●	
112	EMOV	2 进制浮点数数据传送	EMOV S D		●	
116	ESTR	2 进制浮点数转字符串的转换	ESTR S1 S2 D		●	
117	EVAL	字符串转 2 进制浮点数的转换	EVAL S D		●	
118	EBCD	2 进制浮点数-10 进制浮点数转换	EBCD S D	●	●	
119	EBIN	10 进制浮点数-2 进制浮点数转换	EBIN S D	●	●	
120	EADD	2 进制浮点数加法	EADD S1 S2 D	●	●	
121	ESUB	2 进制浮点数减法	ESUB S1 S2 D	●	●	
122	EMUL	2 进制浮点数乘法	EMUL S1 S2 D	●	●	
123	EDIV	2 进制浮点数除法	EDIV S1 S2 D	●	●	
124	EXP	2 进制浮点数指数运算	EXP S D		●	
125	LOGE	2 进制浮点数自然对数运算	LOGE S D		●	
126	LOGE10	2 进制浮点数常用对数运算	LOGE10 S D		●	
127	ESQR	2 进制浮点数开方	ESQR S D	●	●	
128	ENEG	2 进制浮点数符号翻转	ENEG D		●	
129	INT	2 进制浮点数-BIN 整数转换	INT	●	●	
130	SIN	2 进制浮点数 SIN 运算	SIN	●	●	
131	COS	2 进制浮点数 COS 运算	COS	●	●	
132	TAN	2 进制浮点数 TAN 运算	TAN	●	●	
133	ASIN	2 进制浮点数 SIN-1 运算	ASIN S D		●	
134	ACOS	2 进制浮点数 COS-1 运算	ACOS S D		●	
135	ATAN	2 进制浮点数 TAN-1 运算	ATAN S D		●	
136	RAD	2 进制浮点数角度转弧度	RAD S D		●	
137	DEG	2 进制浮点数弧度转角度	DEG S D		●	
140	WSUM	算出数据合计值	WSUM S D n		●	
141	WTOB	字节单位的数据分离	WTOB S D n		●	
142	BTOW	字节单位的数据结合	BTOW S D n		●	
143	UNI	16 位数据的 4 位结合	UNI S D n		●	
144	DIS	16 位数据的 4 位分离	DIS S D n		●	
147	SWAP	上下字节转换	SWAP S	●	●	
149	SORT2	数据排列 2	SORT2 S m1 m2 D n		●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
定位						
150	DSZR	带 DOG 搜索的原点回归	DSZR S1 S2 D1 D2		●	
151	DVIT	中断定位	DVIT S1 S2 D1 D2		●	
152	TBL	表格	TBL D n		●	
155	ABS	读出 ABS 当前值	ABS S D1 D2	●	●	
156	ZRN	原点返回	ZRN S1 S2 S3 D	●	●	
157	PLSV	可变速脉冲输出	PLSV S D1 D2	●	●	
158	DRVI	相对定位#	DRVI S1 S2 D1 D2	●	●	
159	DRVA	绝对定位#	DRVA S1 S2 D1 D2	●	●	
时钟运算						
160	TCMP	时钟数据比较	TCMP S1 S2 S3 S D	●	●	
161	TZCP	时钟数据区间比较	TZCP S1 S2 S D	●	●	
162	TADD	时钟数据加法	TADD S1 S2 D	●	●	
163	TSUB	时钟数据减法	TSUB S1 S2 D	●	●	
164	HTOS	小时、分、秒数据的秒转换	HTOS S D		●	
165	STOH	秒数据的【小时、分、秒】转换	STOH S D	●	●	
166	TRD	时钟数据读出	TRD D	●	●	
167	TWR	时钟数据写入	TWR S	●	●	
169	HOUR	计时	HOUR S D1 D2	●	●	
外围设备						
170	GRY	格雷码变换	GRY S D	●	●	
171	GBIN	格雷码逆变换	GBIN S D	●	●	
176	RD3A	模拟量模块的读出	RD3A m1 m2 D	●	●	
177	WR3A	模拟量模块的写入	WR3A m1 m2 S	●	●	
扩展功能						
180	EXTR	扩展 ROM 功能（CZK2/CZK2C）	EXTR S SD1 SD2 SD3		●	
其他指令						
182	COMRD	读出软元件的注释数据	COMRD S D		●	
184	RND	产生随机数	RND D		●	
186	DUTY	出现定时脉冲	DUTY n1 n2 D		●	
188	CRC	CRC 运算	CRC S D n		●	
189	HCMOV	高速计数器传送	HCMOV S D n		●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
数据块处理						
192	BK+	数据块加法运算	BK+ S1 S2 D n		●	
193	BK-	数据块减法运算	BK- S1 S2 D n		●	
194	BKCMP=	数据块的比较 S1 = S2	BKCMP= S1 S2 D n		●	
195	BKCMP>	数据块的比较 S1 > S2	BKCMP> S1 S2 D n		●	
196	BKCMP<	数据块的比较 S1 < S2	BKCMP< S1 S2 D n		●	
197	BKCMP<>	数据块的比较 S1 ≠ S2	BKCMP<> S1 S2 D n		●	
198	BKCMP<=	数据块的比较 S1 ≤ S2	BKCMP<= S1 S2 D n		●	
199	BKCMP>=	数据块的比较 S1 ≥ S2	BKCMP>= S1 S2 D n		●	
字符串的控制						
200	STR	BIN 转字符串	STR S1 S2 D		●	
201	VAL	字符串转 BIN	VAL S D1 D2		●	
202	$+	字符串的合并	$+ S1 S2 D		●	
203	LEN	检测出字符串的长度	LEN S D		●	
204	RIGHT	从字符串的右侧开始取出	RIGHT S D n		●	
205	LEFT	从字符串的左侧开始取出	LEFT S D S2		●	
206	MIDR	从字符串中任意取出	MIDR S1 D S2		●	
207	MIDW	字符串的检索	MIDW S1 D S2		●	
208	INSTR	字符串的检索	INSTR S1 S2 D n		●	
209	$MOV	字符串的传送	$MOV S D		●	
数据处理 3						
210	FDEL	数据表的数据删除	FDEL		●	
211	FINS	数据表的数据插入	FINS		●	
212	POP	后入的数据读取，后人先出控制用	POP		●	
213	SFR	16 位数据 n 位右移（带进位）	SFR		●	
214	SFL	16 位数据 n 位左移（带进位）	SFL		●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
触点比较						
224	LD=	(S1)＝(S2)	LD= S1 S2	●	●	
225	LD>	(S1)＞(S2)	LD> S1 S2	●	●	
226	LD<	(S1)＜(S2)	LD< S1 S2	●	●	
228	LD<>	(S1)≠(S2)	LD<> S1 S2	●	●	
229	LD<=	(S1)≤(S2)	LD<= S1 S2	●	●	
230	LD>=	(S1)≥(S2)	LD>= S1 S2	●	●	
232	AND=	(S1)＝(S2)	AND= S1 S2	●	●	
233	AND>	(S1)＞(S2)	AND> S1 S2	●	●	
234	AND<	(S1)＜(S2)	AND< S1 S2	●	●	
236	AND<>	(S1)≠(S2)	AND<> S1 S2	●	●	
237	AND<=	(S1)≤(S2)	AND<= S1 S2	●	●	
238	AND>=	(S1)≥(S2)	AND>= S1 S2	●	●	
240	OR=	(S1)＝(S2)	OR= S1 S2	●	●	
241	OR>	(S1)＞(S2)	OR> S1 S2	●	●	
242	OR<	(S1)＜(S2)	OR< S1 S2	●	●	
244	OR<>	(S1)≠(S2)	OR<> S1 S2	●	●	
245	0R<=	(S1)≤(S2)	0R<= S1 S2	●	●	
246	OR>=	(S1)≥(S2)	OR>= S1 S2	●	●	
数据表的处理						
256	LIMIT	上下限限位控制	LIMIT S1 S2 S3 D		●	
257	BAND	死区控制	BAND S1 S2 S3 D		●	
258	ZONE	区域控制	ZONE S1 S2 S3 D		●	
259	SCL	定标（不同点坐标数据）	SCL S1 S2 D		●	
260	DABIN	10 进制 ASCII 转 BIN	DABIN S D		●	
261	BINDA	BIN 转 10 进制 ASCII	BINDA S D		●	
269	SCL2	定标 2（X/Y 坐标数据）	SCL2 S1 S2 D		●	
外部设备通信（变频器通信）						
270	IVCK	变频器的运行监控	IVCK S1 S2 D n		●	
271	IVDR	变频器的运行控制	IVDR S1 S2 S3 n		●	
272	IVRD	变频器的参数读取	IVRD S1 S2 D n		●	
273	IVWR	变频器的参数写入	IVWR S1 S2 S3 n		●	
274	IVBWR	变频器的参数成批写入	IVBWR S1 S2 S3 n		●	
数据传送 3						
278	RBFM	BFM 分割读出	RBFM m1 m2 D n1 n2		●	
279	WBFM	BFM 分割写入	WBFM m1 m2 S n1 n2		●	
高速处理 2						
280	HSCT	高速计数器表比较	HSCT S1 m S2 D n		●	

FNC NO.	指令助记符	功能	符号	CZK2 系列	CZK3 系列	页码
扩展文件寄存器的控制						
290	LOADR	读出扩展文件寄存器			●	
291	SAVER	扩展文件寄存器的一并写入			●	
292	INITR	扩展寄存器的初始化			●	
293	LOGR	记入扩展寄存器			●	
294	RWER	扩展文件寄存器的删除・写入			●	
295	INITER	扩展文件寄存器的初始化			●	

第二章 软元件的作用和功能

在本章中，对可编程控制器中使用的数值和内置的输入输出继电器、辅助继电器、状态、计数器、数据寄存器等各种软元件的作用和功能进行了说明。这些内容是使用可编程控制器时的基础知识。

2-1 软元件编号一览

软元件编号如下进行分配。此外，在基础单元上连接了输入输出扩展设备和特殊扩展设备时，输入继电器和输出继电器请参考使用的可编程控制器硬件手册后进行确认。

软元件名称	内容			CZK 2 系列	CZK 3 系列	页码
输入输出继电器						
输入继电器	X000~X367	248 点	通过软元件的编号为 8 进制编号，输入输出合计为 256 点			
输出继电器	Y000~Y367	248 点				
辅助继电器						
一般用（16 位）【可变】	M0~M499	500 点	通过参数可以更改保持/非保持的设定			
保持用（16 位）【可变】	M500~M1023	524 点				
保持用（16 位）【固定】	M1024~M7679	6656 点	--			
特殊用（16 位）	M8000~M8511	512 点	--			
状态						
初始化状态（一般用）【可变】	S0~S9	10 点	通过参数可以更改保持/非保持的设定			
一般用【可变】	S10~S499	490 点				
保持用【可变】	S500~S899	400 点				
信号报警器用（保持用）【可变】	S900~S999	100 点				
保持用【固定】	S1000~S4095	3096 点	--			
定时器（ON 延迟定时器）						
100ms	T0~T191	192 点	0.1~3276.7 秒			
100ms【子程序、中断子程序用】	T192~T199	8 点	0.1~3276.7 秒			
10ms	T200~T245	46 点	0.01~327.67 秒			
1ms 累计型	T246~T249	4 点	0.001~32.767 秒			
100ms 累计型	T250~T255	6 点	0.1~3276.7 秒			
1ms	T256~T511	256 点	0.001~32.767 秒		●	
计数器						
一般用增计数（16 位）【可变】	C0~C99	100 点	0~32767 的计数器通过参数可以更改保持/非保持的设定			
保持用增计数（16 位）【可变】	C100~C199	100 点				
一般用双方向（32 位）【可变】	C200~C219	20 点	-2147483648 ~ +2147483647 的计数器，通过参数可以更改保持/非保持的设定			
保持用双方向（32 位）【可变】	C220~C234	15 点				

软元件名称	内容			CZK 2 系列	CZK 3 系列	页码
高速计数器						
单相单计数的输入双方向(32 位)	C235~C245	C235~C255 中最多使用 8 点（保持用）通过参数可以更改保持/非保持的设定-2147483648～+2147483647 的计数器； ● **硬件计数器** 单相：100KHz *6 点，10KHz *2 点 双相：50KHz(1 倍)，50KHz(4 倍)				
单相双计数的输入双方向(32 位)	C246~C250					
双相双计数的输入双方向(32 位)	C251~C255					
一般用（16 位）【可变】	D0~D199	200 点	通过参数可以更改保持/非保持的设定			
保持用（16 位）【可变】	D200~D511	312 点				
保持用（16 位）【固定】 <文件寄存器>	D512~D7999 <D512~D7999>	7488 点 <7000 点>	通过参数可以将寄存器 7488 点中 D1000 以后的软元件以每 500 点为单位设定为文件寄存器			
特殊用（16 位）	D8000~8511	512 点	--			
变址用（16 位）	V0~V7,Z0~Z7	16 点	--			
文件寄存器・扩展文件寄存器						
文件寄存器（16 位）	R0~R32767	32768 点	通过电池进行停电保持			
扩展文件寄存器（16 位）	ER0~ER32767	32768 点	仅在安装存储器盒时可用			
指针						
JUMP,CALL 分支用	P0~P4095	4096 点	CJ 指令，CALL 指令用			
输入中断 输入延迟中断	I0□□~I5□□	6 点				
定时器中断	I6□□~I8□□	3 点				
计数器中断	I010~I060	6 点	HSCS 指令用			
嵌套						
主控用	N0~N7	8 点	MC 指令用			
常数						
10 进制数（K）	16 位	-32768~+32767				
	32 位	-2147483648～+2147483647				
16 进制数（H）	16 位	0x0~0xFFFF				
	32 位	0x0~0xFFFFFFFF				
实数(E)	32 位	可以用小数点和指数形式表示				
字符串（“”）	字符串	用“”框起来的字符进行指定。指令上的常数中，最多可以使用到半角的 32 个字符				

2-2 输入输出继电器【X,Y】

输入继电器、输出继电器的编号是由基本单元持有的固定编号，和针对扩展设备连接顺序分配的编号组成的。由于这些编号使用 8 进制数，所以不存在 8 和 9 的数值。

➢输入输出继电器的编号

输入继电器（X）、输出继电器（Y）的编号如下表所示，（编号以 8 进制数分配）。

CZK2/CZK3 系列可编程控制器	型号	CZK3-32MT	扩展时	合计 256 点
	输入	X000~X021 共 18 点	X000~X367 共 240 点	
	输出	Y000~Y014 共 14 点	Y000~Y367 共 240 点	

CZK2/CZK3 系列主机其他型号详见《CZK2/CZK3 系列可编程控制器硬件手册》中 1.2 节。

➢功能和作用

PLC 中，输入端子是从外部的开关接收信号的窗口。PLC 的输入端子上连接的输入继电器（X）为光耦隔离的电子式继电器，因此具有无数的常开触点（a 触点）和常闭触点（b 触点）。在 PLC 中可以随意地使用这些触点。这个输入继电器不能通过程序来驱动。

PLC 中，输出端子是向外部的负载发出信号的窗口。PLC 的输出继电器的外部输出用触点（继电器触点、晶闸管、晶体管等输出元器件），是与这个输出端子相连接的。输出继电器拥有无数个电子式的常开触点、常闭触点，可以在 PLC 中随意地使用。

2-3 辅助继电器【M】

PLC 中有多个辅助继电器。这些辅助继电器的线圈与输出继电器相同，是通过 PLC 中的各种软元件的触点驱动。辅助继电器有无数的电子常开触点和常闭触点，可在 PLC 中随意地使用。但是，不能通过这个触点直接驱动外部负载，外部负载必须通过输出继电器进行驱动。

➢辅助继电器的编号

辅助继电器(M)的编号如下表所示，编号以 10 进制数分配。

一般用	停电保持用	停电保持用	特殊用	CZK2 系列	CZK3 系列
M0~M499 共 500 点	M500~M1023 共 524 点	M1024~M7679 共 6656 点	M1024~M7679 共 6656 点		●

➢功能和动作实例

1、一般用

当 PLC 的电源断开后一般用的辅助继电器都变为 OFF。希望根据停电之前的状态进行控制时，就是用停电保持用辅助继电器。

2、停电保持用

如在 PLC 的运行过程中断开电源，输出继电器和一般的辅助继电器全部都变为 OFF。当再次上电时，除去输入条件为 ON 的以外，都为 OFF。但是，根据控制对象不同，也可能出现停电之前的状态被记住，在再次运行时重新再现的情况。这样的情况下，使用停电保持用辅助继电器。

2-4 状态【S】

状态 S 是对工序步进形式的控制进行简易编程所需要的重要软元件，需要与步进梯形图指令 STL 组合使用。而且在使用 SFC 图的编程方式中也可以使用状态。

➢状态的编号

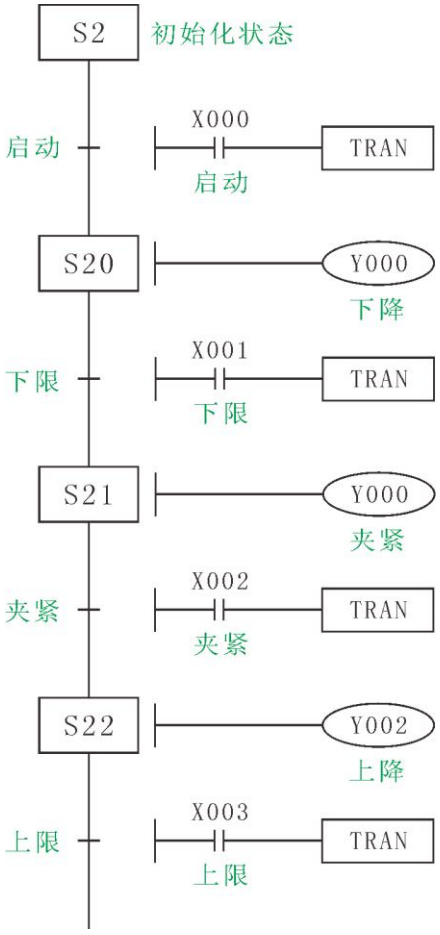
状态(S)的编号如下表所示。（编号以 10 进制数分配）

一般用	停电保持用	固定停电保持专用	信号报警器用	CZK2 系列	CZK3 系列
S0~S499 共 500 点※1， S0~S9 作为初始化用	S500~S899 共 400 点※2	S1000~S4096 共 3096 点※3	S900~S999 共 100 点※2		●

※1 非停电保持区域。根据设定的参数，可以更改为停电保持区域。
※2 停电保持区域。根据设定的参数，可以更改为非停电保持区域。
※3 不能通过参数进行改变停电保持的特性。

➢功能和动作实例

1、一般用



如左图所示的工序步进控制中，启动信号X000为ON后，状态S20被置位（ON），下降用电磁阀Y000工作。

其结果是，如果下限限位开关X001为ON时，状态S21就被置位（ON），夹紧用的电磁阀Y001工作。如确认夹紧的限位开关X002为ON，状态S22就会置位（ON）。

随着动作的转移，转态也会被自动复位（OFF）成移动前状态。

当可编程控制器的电源断开后，一般用状态都变成OFF。

如果想要从停电前的状态开始运行时，请使用停电保持用状态。

状态与辅助继电器相同，有无数常开触点，常闭触点，可以在顺控程序中随意使用。而且不用于步进梯形图时，状态 S 也和辅助继电器 M 相同，可以在一般的顺控中使用。

2、停电保持用

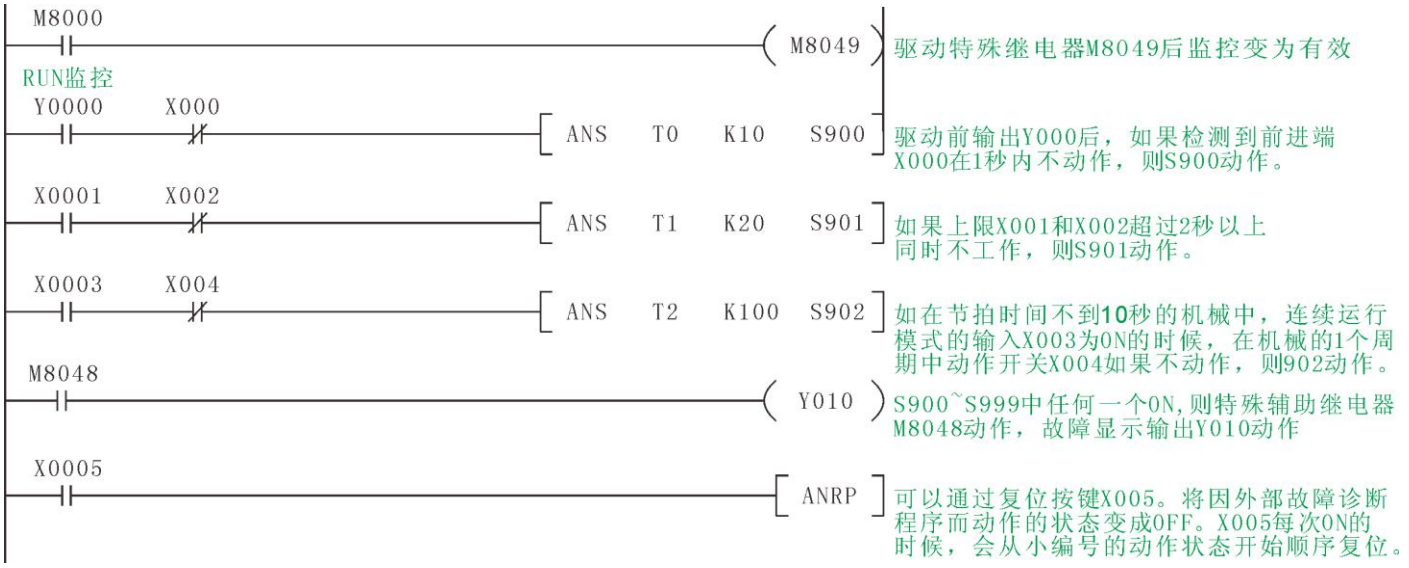
停电保持用就是，即使在 PLC 的运行过程中断开电源，也能记住停电之前的 ON/OFF 状态，并且在再次运行的时候可以从中途的工序开始重新运行。

将停电保持用状态作为一般用状态使用时，请在程序的开头附近设置如图所示的复位梯形图。

3、信号报警用

信号报警器用状态，也可以作为外部故障用的输出使用。

例如，制作如下图所示的外部故障诊断回路，对特殊数据寄存器 D8049 的内容进行监控后，会显示出 S900~S999 中的动作状态的最小编号。发生多个故障时，消除最小故障后即可以知道下一个故障编号。



不驱动特殊辅助继电器 M8049 时，停电保持用状态与一般状态一样，可以在顺控程序内使用。

2-5 定时器【T】

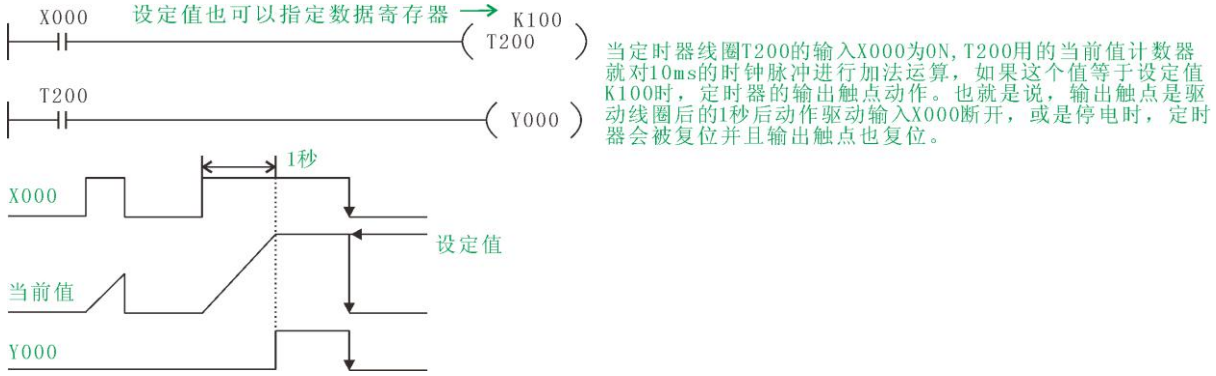
➢定时器的编号

定时器（T）的编号如下表所示，编号以 10 进制数分配。不作为定时器使用的定时器编号，也可以作为存储数值用的数据寄存器使用。

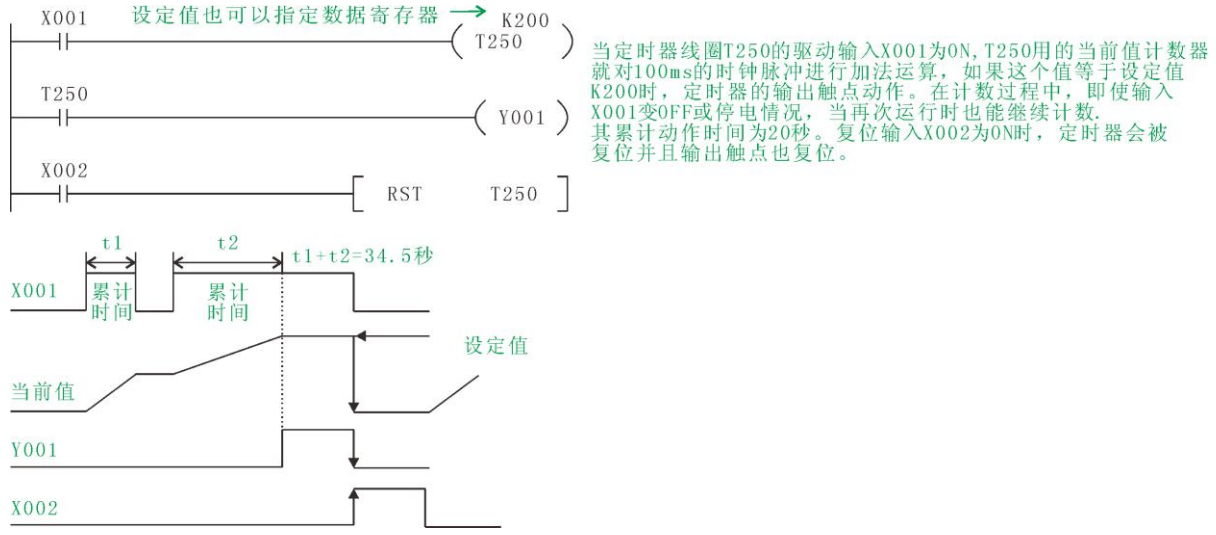
100ms 型 0.1~3276.7 秒	10ms 型 0.01~327.67 秒	1ms 累计型 0.001~32.767 秒	100ms 累计型 0.1~3276.7 秒	1ms 累计型 0.001~32.767 秒	CZK 3 系列
T0~T199 共 200 点， 子程序用 T192~T199	T200~T245 共 46 点	T246~T249 共 4 点，执行中断保持用	T250~T255 共 6 点，保持用	T256~T511 共 256 点	●

➢功能和动作实例

1、一般用

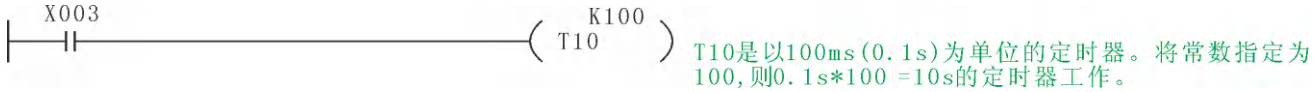


2、累计型

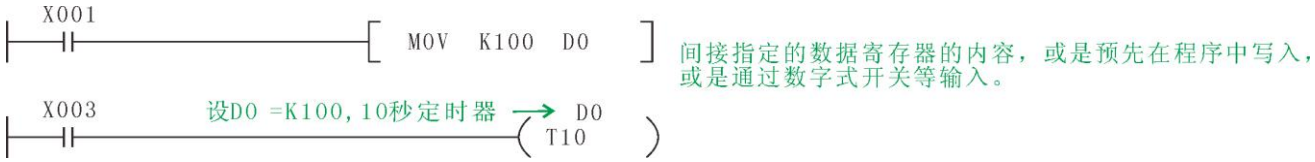


➢设定值的指定方法

1、指定常数（K）



2、间接指定



➢子程序内的注意事项

1、在子程序和中断子程序中，请使用 T192~T199 的定时器。这种定时器在执行线圈指令的时候，或是执行 END 指令的时候进行计时。

如果达到设定值，则在执行线圈指令，或是执行 END 指令的时候输出触点动作。由于一般的定时器，仅在执行线圈指令的时候进行计时，参考下面的【定时器动作的详细内容和定时器精度】，所以只有在某种条件下，才执行线圈指令的子程序和中断子程序，如果使用该定时器计时就不能执行，不能正常动作。

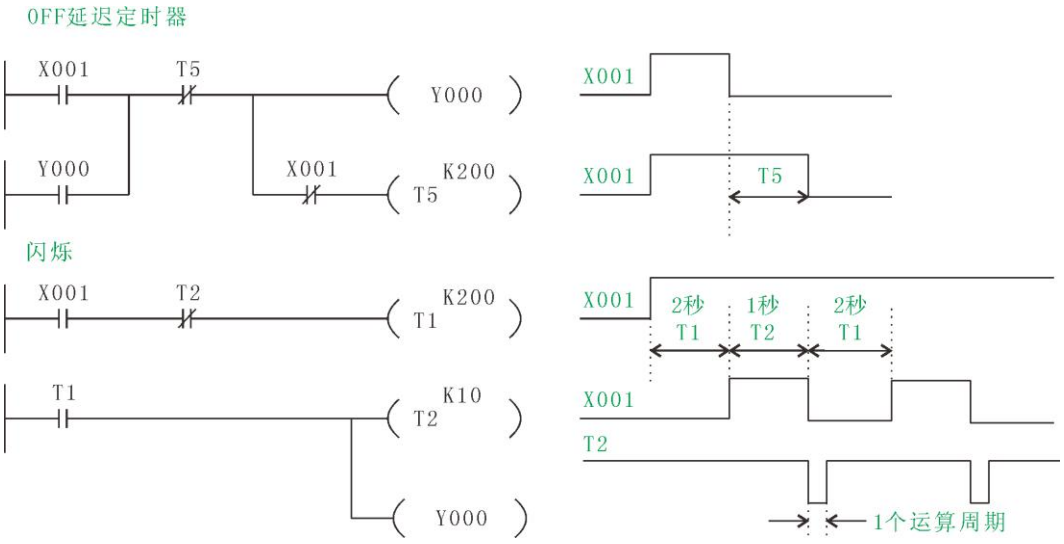
2、在子程序和中断子程序中，如果使用了 1ms 累计定时器，当它达到设定值以后，在最执行的线圈指令处触点会动作。请务必注意。

➢定时器动作的详细内容和定时器的精度

除中断执行型定时器以外，线圈被驱动后开始计时，到时间以后，在最初执行的线圈指令处触点动作。

如果编程的时候，触点在定时器线圈前面的话，最大误差情况为+2T0。此外，定时器的设定值为 0 时，在下一个循环中，线圈指令执行时，输出触点动作。此外，中断执行型的 1ms 定时器是在线圈指令执行后，以中断方式对 1ms 的时钟脉冲进行计数。

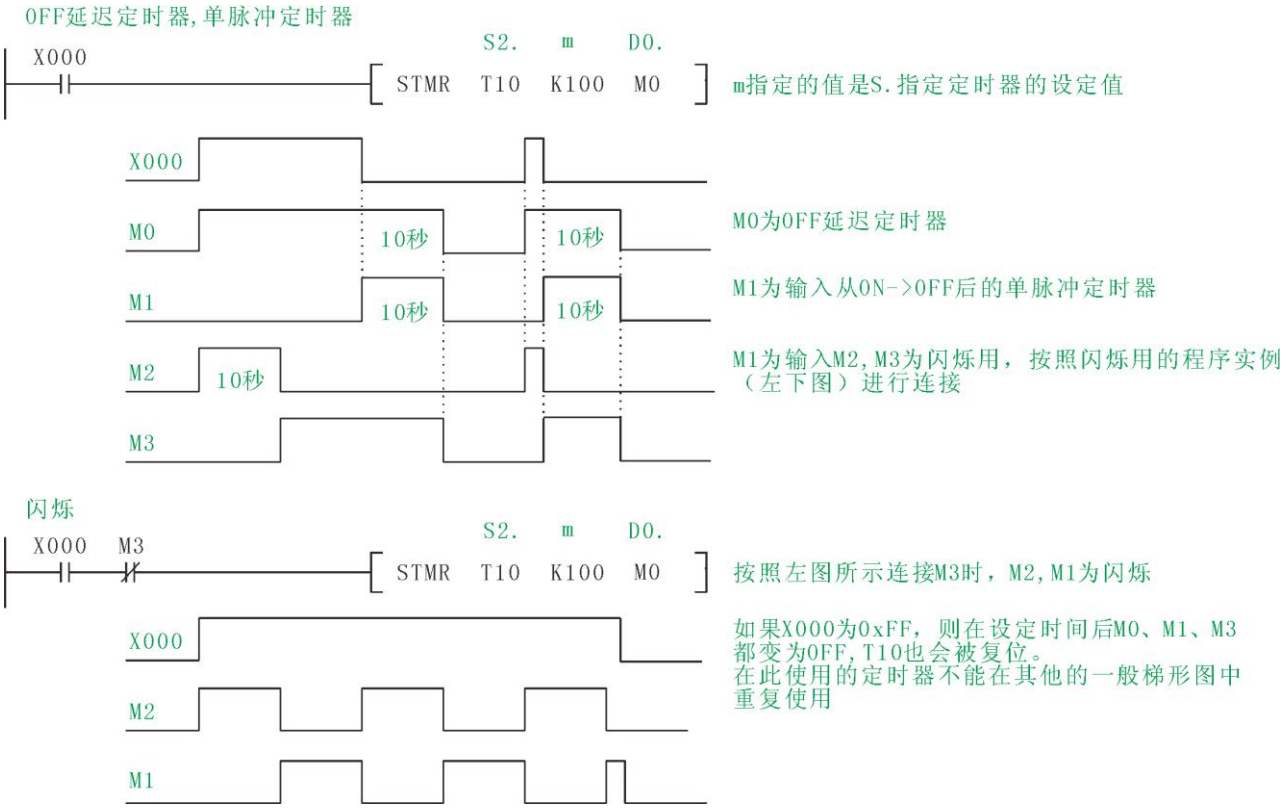
➢程序实例【OFF 延迟定时器，闪烁】



使用 ALT 指令也可以实现闪烁的动作。

1、使用了 FNC65 - STMR 指令的多重定时器

使用这个指令，可以简易制作 OFF 延迟定时器、单脉冲定器，闪烁定时器。

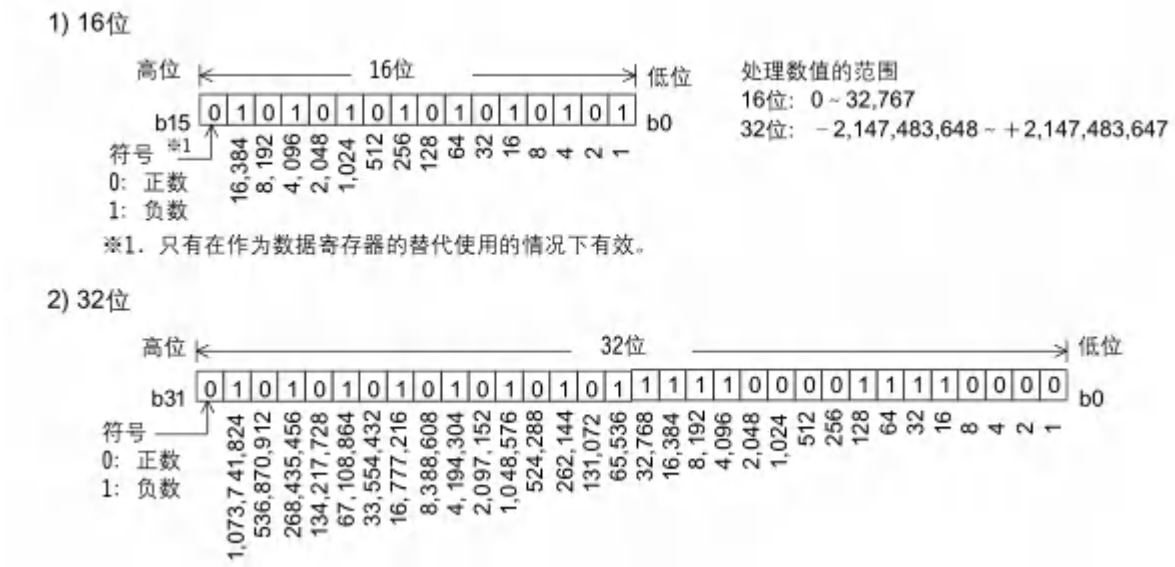


此外,使用 TTMR(FNC64)示教定时器指令时，还可以根据开关的输入时间来设定定时器的时间

➢作为数据软元件的处理

定时器分为：根据设定值而动作的输出触点被使用的情况，以及在控制中将当前值作为数值数据使用的情况。定时器的当前值寄存器的结构如下所示，当对应指令中的操作数指定了定时器编号后，就被作为与数据寄存器相同的 16 位或 32 位的保存数据的软元件来处理。

1、定时器的当前值寄存器的结构



2、在应用指令中使用的实例

想要将定时器作为数值软元件灵活使用时，请阅读后述的应用指令说明内容。

2-6 计数器【C】

➢高速计数器的种类和软元件的编号

1、高速计数器的种类

基本单元中，内置了 32 位增减计数器的高速计数器(单相单计数、单相双计数以及双相双计数)。在这个高速计数器中，根据计数的方法不同可以分为硬件计数器和软件计数器两种。而且，在高速计数器中，提供了可以选择外部复位输入端子和外部启动输入端子(开始计数)的功能。

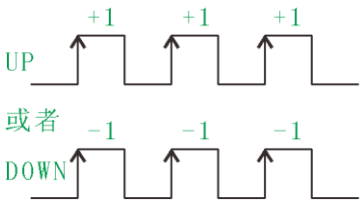
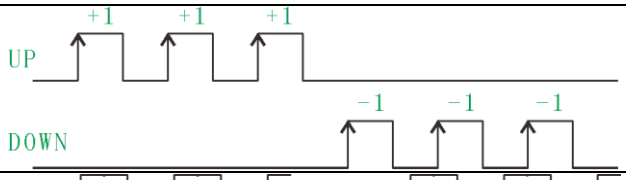
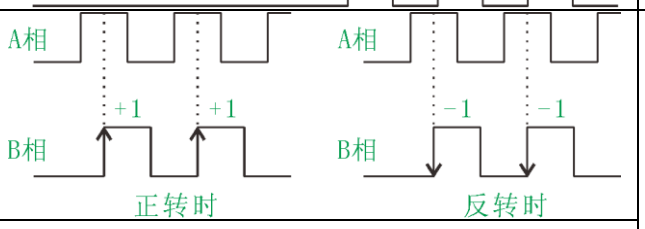
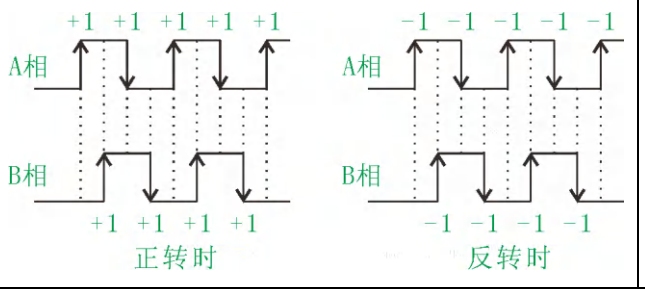
2、根据高速计数器的计数不同的区分

硬件计数器:这种计数器就是通过硬件进行计数。但是，根据使用条件，也可以切换成软件计数器。

软件计数器:这种计数器就是通过 CPU 的中断处理进行计数。每个计数器需要在最大响应频率和综合频率的两个限制条件下使用。

3、高速计数器的种类和输入信号的形式

有关高速计数器的种类(单相单计数、单相双计数以及双相双计数)和输入信号(波形)如下所示。

		输入信号形式	计数方向
单相单计数的输入			通过 M8235-M8245 的 ONOFF 来指定增计数或是减计数。 ON:减计数；OFF:增计数
单相双计数的输入			如左图所示，进行增计数或是减计数。其计数方向可以通过 M8246-M8250 进行设置。 ON:减计数；OFF 增计数
双相双计数的输入	1 倍		如左图所示，进行增计数或是减计数。其计数方向可以通过 M8251-M8255 进行设置。 ON:减计数；OFF 增计数
	4 倍		

4、与高速计数器输入相连的设备的注意事项

高速计数器的输入，使用通用输入 X000- X007，电压输出型或绝对型编码器，不可以连接到高速计数器输入上。有关接线的内容请参考可编程控制器硬件手册。

5、高速计数器的软元件一览

	区分	计数器编号	1 倍/4 倍	数据长度	外部复位的输入端子	外部开始的输入端子
单相单计数器输入	硬件计数器※1	C235※2 C236※2 C237※2 C238※2 C239※2 C240※2	-	32 位 增减计数器	无	无
		C244(OP)※3 C245(OP)※3	-			
	软件计数器	C241、C242、C243	-		有※5	-
		C244※3、C245※3	-		有※5	有
单相单计数输入	硬件计数器※1	C246※2 C248(OP)※2※3	-	32 位 增减计数器	无	无
	软件计数器	C247、C248※3	-		有※5	无
		C249、C250	-		有※5	有
双相双计数输入	硬件计数器※1	C251※2	1 倍※4	32 位 增减计数器	无	无
			1 倍※4			
		C251※3	1 倍※4		有※5	
			1 倍※4			
	软件计数器	C252	1 倍※4		有※5	无
			1 倍※4			
		C253(OP)※3	1 倍※4		无	
			1 倍※4			
		C254	1 倍※4		有※5	有
		C255	1 倍※4			

※1 根据使用条件可以作为软件计数器处理。被作为软件计数器处理时，受到最大响应频率和综合频率两者的限制。

※2 在这个高速计数器中，接线上有需要注意的事项。有关接线的内容，

※3 C244、C245、C248 通常是作为软件计数器使用的，但是和特殊辅助继电器(M8388、M8390~M8392)，一起使用后，也可以作为硬件计数器 C244(OP)、C245(OP)、C248(OP)使用。

※4 双相双计数的输入计数器，通常是 1 倍的计数器，但是如果和特殊辅助继电器(M8388、M8198、M8199)，一起使用后，可以作为 4 倍的计数器使用。

※5 外部复位输入，通常在 ON 的时候复位，但是如果和特殊辅助继电器(M8388、M8392)起使用时， 可以更改为在 OFF 时复位。

※6 C253 通常是作为硬件计数器使用的，但是如果和特殊辅助继电器(M8388、M8392) 一起使用的话，就可以作为不带复位输入的计数器 C253 (OP)使用但是，此时 C253 (OP)作为软件计数器使用。

有关高速计数器的软元件的记载

CZK2/CZK3 系列可编程控制器的高速计数器中，通过与特殊辅助继电器组合使用、可以改变输入端子的分配情况。在本节中，将这些高速计数器的软元件如下表所示地进行了区别，编程的时候，请注意不可以输入(OP)。

普通的软元件编号	切换后的软玩件编号
C244	C244(OP)
C245	C245(OP)
C248	C248(OP)
C253	C253(OP)

➢高速计数器的输入分配

对应各个高速计数器的编号，输入 X000 一 X007 如下表所示进行分配。

使用高速计数器时，对应的基本单元输入编号的滤波器常数会自动变化。但是，不作为高速计数器使用的输入端子，可以作为一般的输入使用。有关基本单元的输入规格，请参考要使用的可编程控制器硬件手册。

	计数器编号	区分	输入端子的分配							
			X000	X001	X002	X003	X004	X005	X006	X007
单相单计数器输入	C235※1	H/W※2	U/D							
	C236※1	H/W※2								
	C237※1	H/W※2								
	C238※1	H/W※2				U/D				
	C239※1	H/W※2								
	C240※1	H/W※2								
	C241	S/W	U/D	R						
	C242	S/W				U/D	R			
	C243	S/W								
	C244	S/W	U/D	R					S	
	C245	S/W				U/D	R			S
单相双计数输入	C246※1	H/W※2	D	U						
	C247	S/W	D	U	R					
	C248	S/W				D	U	R		
	C249	S/W	D	U	R				S	
	C250	S/W				D	U	R		S
双相双计数输入	C251※1	H/W※2	A	B						
	C252	S/W	A	B	R					
	C253※1	H/W※2				A	B	R		
	C254	S/W	A	B	R				S	
	C255	S/W				A	B	R		S

H/W:硬件计数器；S/W:软件计数器；U:增计数输入；D:成计数输入；A :A 相输入；B: B 相输入；R:外部复位输入；S:外部启动输入。

※1 在这个速计数器中，接线上有需要注意的项。有关接线的内容，

※2 与高速计数器用的比较置位复位指令(DHSCS、DHSCR、DHSZ、DHSCT)组合使用时，硬件计数器(HW)变为软件(SW)计数器。而且，执行外部复位输入的逻辑反转以后，C253 会 变成软件计数器。

※3 通过用程序驱动特殊辅助继电器可以切换使用的输入端子及功能。

※4 双相双计数的计数器通常为 1 倍计数。但是，如果和特殊辅助继电器组合使用时，可以变成 4 倍计数。

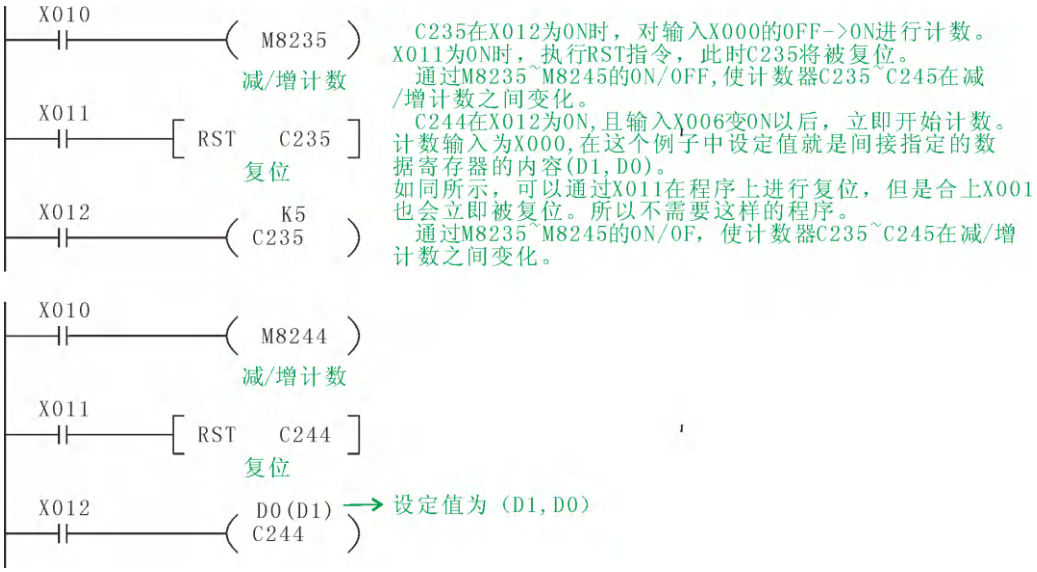
有关禁止重复使用输入端子

输入 X000- X007，可用于高速计数器、输入中断、脉冲捕捉以及 SPD、ZRN、DSZR、DVIT 指令和通用输入。因此，请勿重复使用输入端子。

例如，使用 C251 的时候 X000、X001 被占用了，所以 C235、C236、C241、C244、C246、C247、C249、C252、C254；输入中断指针 1000、1011；脉冲捕捉用触点 M8170、 M8171；以及使用相应输入的 SPD、ZRN、DSZR、DVIT 指令都不可以使用。

➢高速计数器的使用

1、单相单计数的输入



动作实例

上述的计数器 C235 的动作如下图所示。

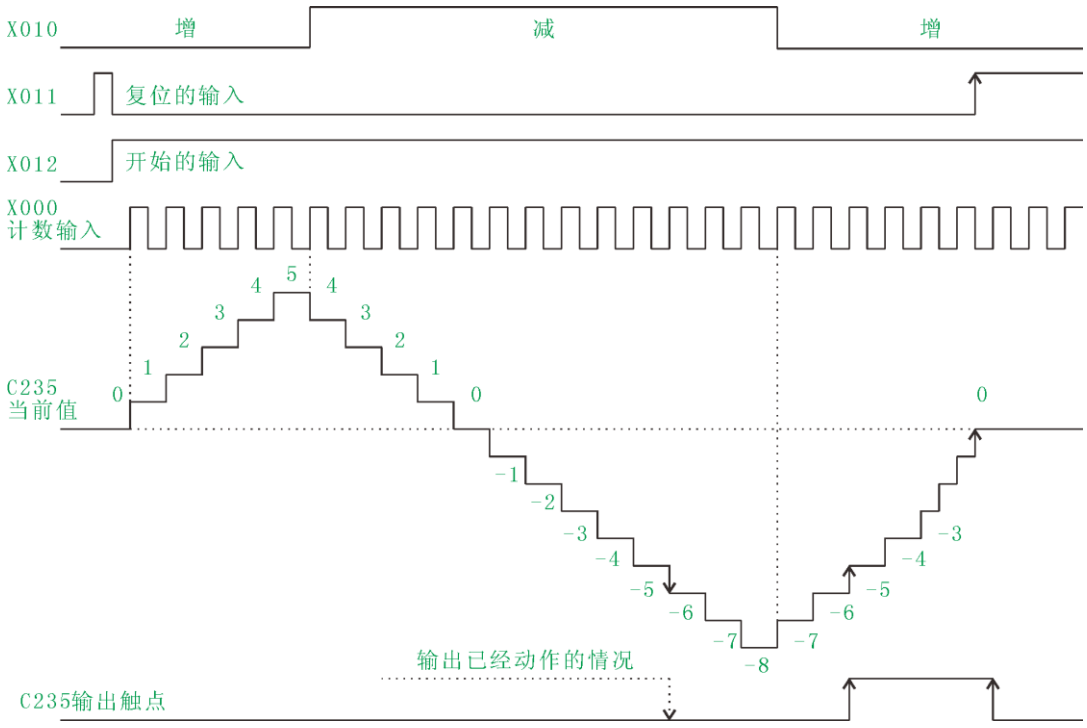
根据计数输入 X000, C235 通过中断进行增或是减的计数。

1）当前值从”-6 增加到"-5”的时候输出触点被置位，当前值从”-5"减少到”-6”的时候输出触点被复位。

2）当前值的增减与输出触点的动作与无关，如果从 2147483647 开始增计数，则变成-2147483648。同样地，如果从-2147483648 开始减计数，则变成 2147483647。(这样的动作被称为环形计数)

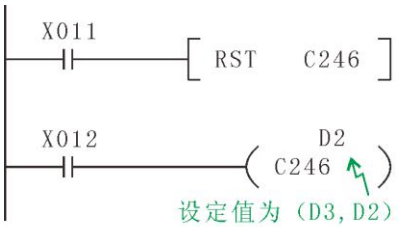
3）复位输入 X011 为 ON，执行 RST 指令，此时，计数器的当前值变为 0,输出触点也复位。

4）在停电保持用的高速计数器中，即使电源断开，计数器的当前值和输出触点的动作、复位状态都会被保持。

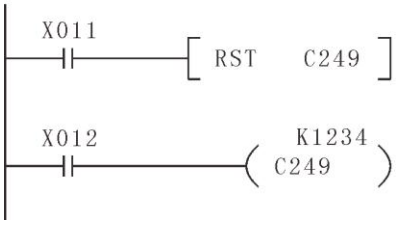


2、单相双计数的输入

就是 32 位增/减的二进制计数器，对应于当前值的输出触点的动作与上述的单相单计数输入的高速计数器相同。



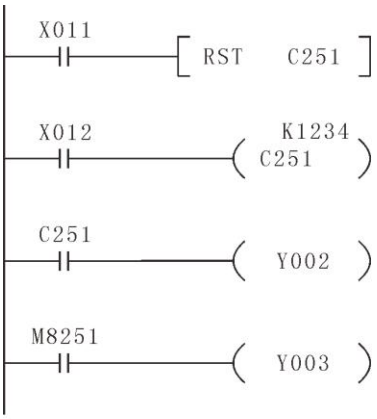
C246在X012为ON时，对输入X000的OFF->ON时为增计数，如果过X001由OFF->ON时就为减计数。
C246~C250的减/增计数动作可以通过M8246~M8250的ON/OFF动作进行监控。ON减计数，OFF增计数



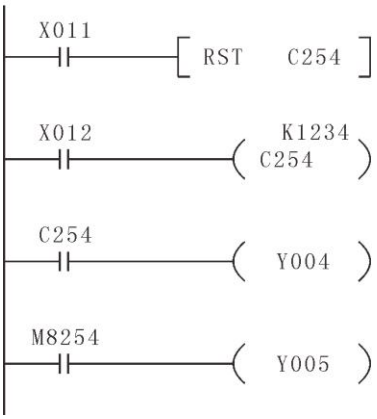
C249在X012为ON时，如果输入X006为ON以后就立即开始计数。增计数输入为X000。减计数输入为X001。
如左图所示，可以通过X011在程序上进行复位，但是X002合上时就会立即被复位。所以不需要这样的程序。
C246~C250的减/增计数动作可以通过M8246~M8250的ON/OFF动作进行监控。ON为减计数,OFF为增计数。

3、双相双计数的输入

就是32位增/减的二进制计数器,对应于当前值的输出触点的动作与上述的单相高速计数器相同。



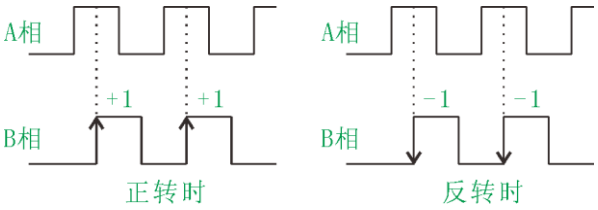
X012为ON时，C251通过中断，对输入X000(A相),X001(B相)进行计数。
X011为ON,执行RST指令,此时C251将被复位。
当前值超出设定值的话Y002为ON,在设定值以下范围内变化时为OFF。
Y003根据计数方向而ON(减)，OFF(增)。



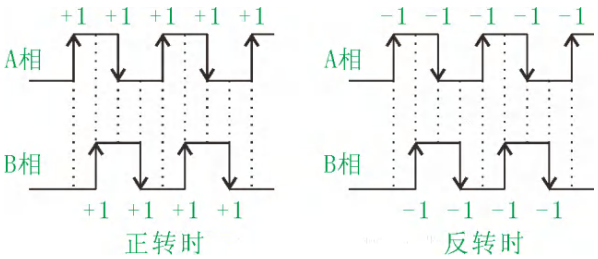
X012为ON时，如果X006为ON后就立即开始C254的计数。该计数器的输入为X000(A相)、X001(B相)。
除了使用X011在程序上进行复位以外，X002为ON时也可以立即将C254复位。
当前值超出设定值（D1,D0）的时Y004动作，在设定值以下的范围内变化时为OFF。
Y005是根据计数方向而ON(减)，OFF(增)。

1）双相编码器输出有 90 度相位差的 A 相和 B 相。据此，高速计数器如下图所示自动地执行增/减的计数。

以 1 倍动作的时



以 4 倍动作的时



2）C251~C255 的减/增计数状态，可以通过 M8251~M8255 的 ON/OFF 动作进行监控。

ON：减计数；OFF：增计数。

当前值更新时序及当前值的比较

1、当前值的更新时间

在高速计数器用的输入端子中输入脉冲后会执行增计数或是减计数，但是软元件的当前值是按照下表所示的时序进行更新的。因此，当硬件计数器通过使用常 用的 MOV 指令、CMP 指令和触点比较指令等应用指令，将高速计数器的当前值原样不动地进行处理时，使用的是已经按照下表中的时序更新了的当前值，所以会受到扫描的影响。

	当前值的更新时序
硬件计数器	当执行计数器的 OUT 指令，或 HCMOV 指令
软件计数器	当计数输入时

2、当前值的比较

比较高速计数器的当前值后输出时，有以下的 2 种方法。

1）使用用比较指令(CMP)、区间比较指令(ZCP)和比较触点指令

善计数器计数时不需要比较结果的情况下，在比较指令(CMP 指令/ZCP 指令)或比较触点指令的前面使用 HCMOV 指令时，在主程序内就可以更加适时※1 地进行比较。

※1 要在高速计数器的当前值已经变化的时序中执行比较，改变输出触点(Y)时，请使用高速计数器用的比较指令(HSCS/HSCR/HSZ/HSCT 指令)

2）使高速计数器用的比较指令(HSCS/HSCR/HSZ/HSCT 指令)，就是在作为对象的高速计数器进行计数时，执行比较并且输出比较的结果。这些指令如下表所示，在使用次数上有限制。对比较结果指定了输出继电器(Y)时，不等到 END 指令的输出刷新，就直接反映到输出的 ON/OFF 状态中。

如果是继电器输出型的可编程控制器，由于存在机械性的动作延迟(约 10ms)所以请使用晶体管输出型的产品。

指令	指令使用次数的限制
HSCS	HSCS 包括 HSCT 指令在内可以使用 32 次
HSCR	
HSZ※1	
HSCT※1	只能使用一次

※1 使用 HSZ 指令或 HSCT 指令时，所有的软件计数器的最大响应频率和综合频率都受到限制。

➢相关软元件

1、单相单计数输入计数器的增/减计数的切换使用

种类	计数器编号	指定用的软元件	增计数	减计数
单相单计数输入	C235	M8235	OFF	ON
	C236	M8236		
	C237	M8237		
	C238	M8238		
	C239	M8239		
	C240	M8240		
	C241	M8241		
	C242	M8242		
	C243	M8243		
	C244	M8244		
	C245	M8245		

2、单相双计数和双相双计数输入计数器的增/减计数单相的监控用

种类	计数器编号	指定用的软元件	增计数	减计数
单相双计数输入	C246	M8246	OFF	ON
	C247	M8247		
	C248	M8248		
	C249	M8249		
	C250	M8250		
双相双计数输入	C251	M8251		
	C252	M8252		
	C253	M8253		
	C254	M8254		
	C255	M8255		

3、高速计数器的功能切换使用

软元件编号	计数器编号	内容
M8388	高速计数器的功能变更用触点	高速计数器的功能变更用触点
M8389	功能切换的软元件	外部复位输入的逻辑切换
M8390		C244 用功能切换软元件
M8391		C245 用功能切换软元件
M8392		C248，C253 用功能切换软元件
M8198		C251，C252，C254 用的 1 倍/4 倍的切换软元件
M8199		C253，C255，C253（OP）用的 1 倍/4 倍的切换软元件

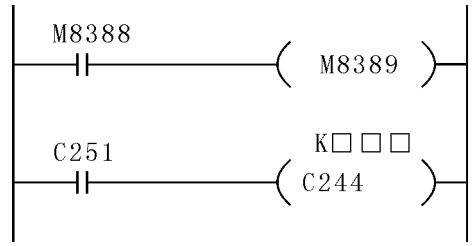
4、硬件计数器/软件计数器的动作状态

软元件编号	名称	内容	ON	OFF
M8380※1	动作状态	C235、C241、C244、C246、C247、C249、C251、C252、C254 的动作状态	软件计数器	硬件计数器
M8381※1		C236 的动作状态		
M8382※1		C237、C242、C245 的动作状态		
M8383※1		C238、C248、C248(OP)、C250、C253、C255 的动作状态		
M8384※1		C239、C243 的动作状态		
M8385※1		C240 的动作状态		
M8386※1		C244(OP)的动作状态		
M8387※1		C245(OP)的动作状态		

※1 当 PLC 由 STOP 变成 RUN 时会被清除

➢关于外部复位输入信号的额逻辑变更

计数器的 C241~C245，C247~C250 和 C252~C255 的外邢复位输入，通常在 ON 的时候复位。可以通过编写下面的程序，使逻辑反转，也就是可以改为当输入 OFF 的时候复位。

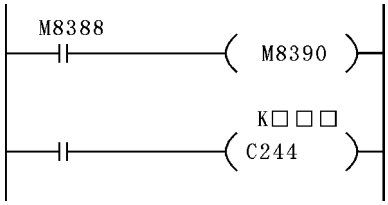
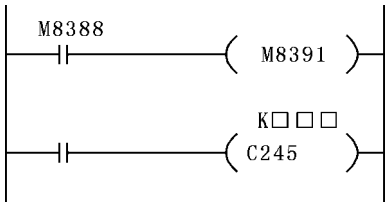
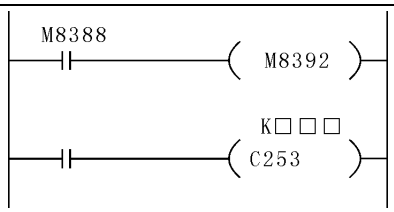
计数器编号	外部复位纳入信号的逻辑反转	变化的内容
C241~C245 C247~C250 C253~C255		将外部复位输入的逻辑反转,也就是在 OFF 的时候复位。 (对象的计数器编号所有的逻辑都反转。)

注意事项：

外部复位输入信号的逻辑反转以后，C253 会变为软件计数器。

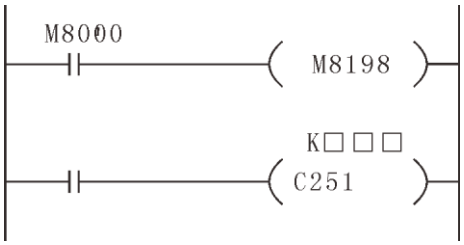
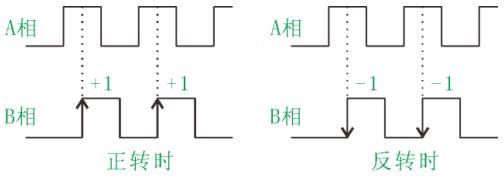
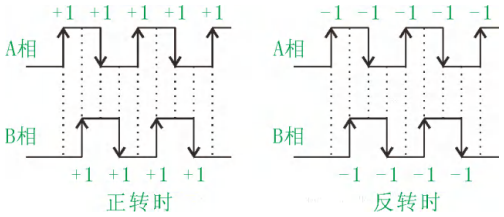
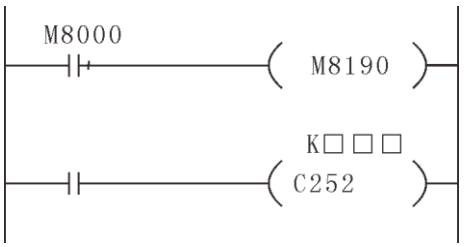
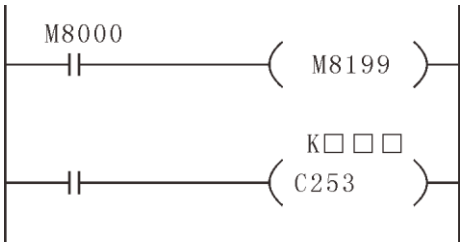
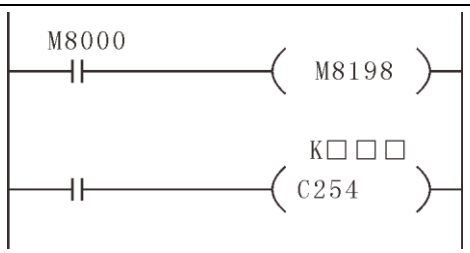
➢关于计数器的输入端子分配和功能的切换

软件计数器 C244、C245、C248 和 C253 可以通过和以下的特殊辅助继电器组合使用，使输入端子的分配和功能产生如下所示变化。此外，请在编程的时候将特殊辅助继电器写在计数器前面。

计数器编号	外部复位纳入信号的逻辑反转	变化的内容
C244(OP)		1）计数输入从 X000 变成 X006； 2）没有复位输入； 3）作为软件计数器动作。没有复位输入； 4）作为软件计数器动作。没有复位输入； 5）没有启动输入。
C245(OP)		1）计数输入从 X001 变成 X007； 2）没有复位输入； 3）没有启动输入； 4）作为硬件计数器动作。
C248(OP)		1）没有复位输入； 2）作为硬件计数器动作。
C253(OP)		1）没有复位输入 2）作为软件计数器动作

➢以 4 倍频使用双相双计数的计数器 C251~C255 的方法

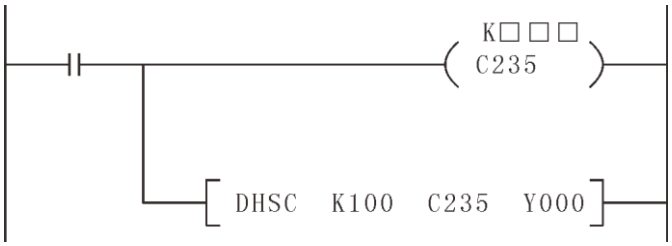
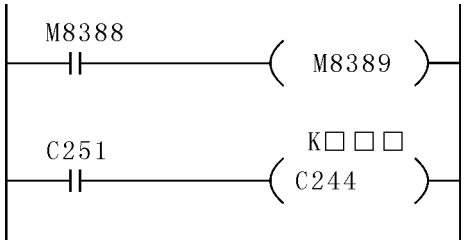
双相双计数输入计数器 C251~C255，通常是 1 倍的，但是如果按照下表所示编程的话，可以以 4 倍动作。

计数器编号	作为 4 倍增的双相输入计数器使用的场合	变化的内容
C251		1 倍（变更前）  4 倍（变更后） 
C252		
C253		
C253(OP)		
C254		
C255		

➢将硬件计数器作为软件计数器使用的条件

高速计数器中包括硬件计数器和软件计数器。但是根据使用方法，硬件计数器也可以和软件计数器执行相同的处理。在这种情况下，请在软件计数器的最大响应频率和综合频率范围内使用。

作为软件计数器使用的条件

计数器编号	内作为软件计数器使用的条件
C235 C236 C237 C238 C239 C244(OP) C245(OP) C246 C248(OP) C251 C253(OP)	硬件计数器，就是用硬件进行计数的计数器，所以能与综合频率无关进行计数。但是，在下述条件下使用时，就与软件计数器的处理方法相同。在这种情况下，最大响应频率和综合频率，都和其他的软件计数酒相同。高速计数器中，可以通过 M8380- MB387 来确认是以硬件计数景，还是软件计数器动作。 1）对硬件计数据编号，使用 DHSCS(FNC 53)指令，DHSCR(FNC 54)指令，DHSZ(FNC 55)指令，DHSCT(FNC 280)指令时，等同于软件计数器的处理。例如: C235  这种情况下,C235 为软元件计数器 2）对于 DHSCS(FNC 53)指令，DHSCR(FNC 54)指令，DHSZ(FNC 55)指令，DHSCT(FNC 280)指令中指定的计数器编号，使用变址寄存器时，所有的硬件计数器都和软件计数善的处理方法相同。例如: C235Z0  3）通过外部复位输入信号逻辑委更功能。执行建辑反转以后 C253 (硬件计数器)安得和软件计数器的处理方法相同。例如:反转 C253 的外部复位情号的逻辑。 

➢高速计数器的响应频率

1、硬件计数器的响应频率

硬件计数器的最大响应频率如下表所示。但是，根据使用条件，有时候硬件计数器也会和软件计数器一样达到最大响应频率，从而受到综合频率的限制。

		计数器编号	基本单元
单相单计数输入		C235、C236、C237、C238、C239、C240	100KHz
		C244(OP)、C245(OP)	10KHz
单相双计数输入		C246、C248(OP)	100KHz
双相双计数输入	1 倍	C251、C253	50KHz
	4 倍		50KHz

2、软件计数器的响应频率和综合频率

软件计数器的最大响应频率和综合频率如下表所示。

在程序中使用了 HSZ 指令或 HSCT 指令的情况下，与指令的操作数无关，所有的软件计数器的最大响应频率和综合频率都有限制。

在讨论系统配置，或者编程的时候，要考虑到该限制内容，在符合最大响应频率和综合频率的范围内使用。

1)不使用模拟量特殊适配器或 CZK3. CZK3C 系列的特殊功能模块/单元时

计数器的种类				计算综合频率用的倍率	根据使用指令的条件而定的响应频率和综合频率（KHz）							
	软件计数器		下面的计数器中 HSCS、HSCR、HSZ、HSCT 指令并用的软件计数器 ※1		无 HSZ,HSCT 指令		仅有 HSCT 指令		仅有 HSZ 指令		HSZ,HSCT 指令都有	
					最大响应频率	综合频率	最大响应频率	综合频率	最大响应频率	综合频率	最大响应频率	综合频率
单相单计数输入	C241、C242、C243、C244、C245		C235、C236、C237、C238、C239、C240	X1	40	80	30	60	40-指令使用次数※2	80-1.5x 指令使用次数	30-指令使用次数※2	60-1.5x 指令使用次数
	-		C244(OP)、C245(OP)	X1	10		10					
单相双计数输入	C247、C248、C249、C250		C246、C248(OP)	X1	40		30					
双相双计数输入	1 倍	C252、C253(OP、C254、C255	C251、C253	X1	40		30					
	4 倍				10		7.5		40-指令使用次数/4		30-指令使用次数/4	

※1 在 HSCS、HSCR、HSZ、HSCT 指令指定的计数器编号上附加变址寄存器时，所有的硬件计数器都切换成软件计数器。

※2 高速计数器 C244(OP)和 C245(OP),不能进行 10kHz 以上的计数。

2)使用了模拟量特殊适配器和 CZK2/CZK3 系列的特殊功能模块/单元时

计数器的种类				计算综合频率用的倍率	根据使用指令的条件而定的响应频率和综合频率（KHz）							
		软件计数器	下面的计数器中 HSCS、HSCR、HSZ、HSCT 指令并用的软件计数器※1		无 HSZ、HSCT 指令		仅有 HSCT 指令		仅有 HSZ 指令		HSZ、HSCT 指令都有	
					最大响应频率	综合频率	最大响应频率	综合频率	最大响应频率	综合频率	最大响应频率	综合频率
		C241,C242, C243,C244, C245	C235,C236,C237, C238，C239，C240	X1	40	80	30	60	41-（指令使用次数）※2	80-1.5x(指令使用次数)	30-（指令使用次数）※2	60-1.5x(指令使用次数)
		-	C244(OP), C245(OP)	X1	10		10					
单相双计数输入		C247,C248, C249,C250	C246，C248(OP)	X1	40		30					
双相双计数输入	1倍	C252, C253(OP) C254, C255	C251，C253，	X1	40		30					
	4倍				10		7.5		(40-指令使用次数）/4		(30-指令使用次数）/4	

※1 在 HSCS、HSCR、HSZ、HSCT 指令指定的计数器编号上附加变址寄存器时，所有的硬件计数器都切换成软件计数器。

※2 高速计数器 C244(OP)和 C245(OP),不能进行 10kHz 以上的计数。

- 有关综合频率的计算

综合频率≥ [高速计数器的响应频率 X 综合频率计算用倍率]的合计，有关具体计算的实例，请参考下一页。

- 计算实例

在程序中仅仅使用了 6 次 HSZ 指令的情况下,根据上表的仅有 HSZ 指令中的项目进行如下计算。这个计算实例中，是没有使用模拟量特殊适配器和 CZK2/CZK3 系列的特殊功能模块/单元的系统配置。

使用的高速计数器编号		输入频率	最大响应频率的计算	计算综合频率用的倍率	使用指令
C237	作为软件计数器动作	30kHz	40-6（次）=34kHz	X1	HSZ 指令 6 次
C241	软件计数器	20kHz	40-6（次）=34kHz	X1	
C253(OP)[4 倍]		4kHz	40-6（次）/4=8.5kHz	X4	

1) 使用的指令是 HSZ 指令使用了 6 次，所有按照下面的公司计算出综合频率。其值为 80-1.5*6 =71KHz

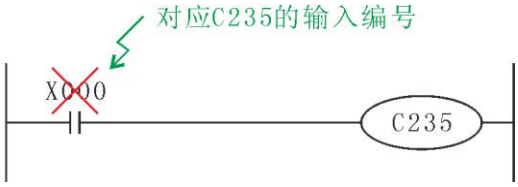
2) 使用的高速计数器的响应频率。其值为 30KHz * 1【C237】+ 20KHz * 1【C241】+ 4 * 4【C253(OP)】=66KHz < 71KHz

使用上的注意事项

高速计数器的线圈驱动用触点，在高速计数时，请使用为 ON 的那种触点。

例：M8000(RUN 监控)

计数器请使用一直为ON的触点进行编程。



指定了计数用输入继电器的编号后，
高速计数器不能正确进行计数。

1、如果用 拟开关等有触点的设备执行 速计数器的动作时，由于开关的振动，计数器可能出现计数误差，请注意。

2、速计数器中使用的基本单元输入端子的输入滤波器会被自动设定为 5u s(X000~X005), 或是 50us (X006, X007)。

因此，不需要使用 REFF 指令和特殊数据寄存器 D8020(输入滤波器的调节)。

此外，不作为高速计数器输入使用的输入继电器的输入滤波器维持 10ms(初始值)。

3、输入 X000~ X007 可以用于高速计数器、输入中断、脉冲捕捉以及 SPD, DSZR, DVIT, ZRN 指令和通用输入。因此，请勿重复使用输入端子。

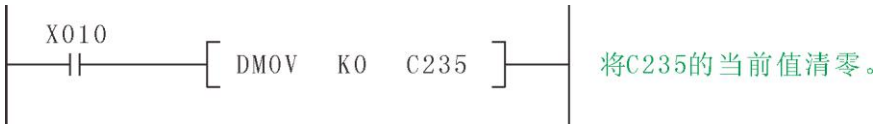
例如使用了 C251 时，由于 X0OO, X001 都被占用了，所以「C235, C236, C241, C244, C246, C247, C249,C252,C254」「输入中断指针 100*, I10* 」「脉冲捕捉用触点 M8170, M8171」和「适用输入的 SPD 指令」都不能使用。

4、所有的高速计数器，例如，在当前值=设定值的状态下，即使执行指令，只要不能给出计数输入脉冲，输出触点都不会动作。

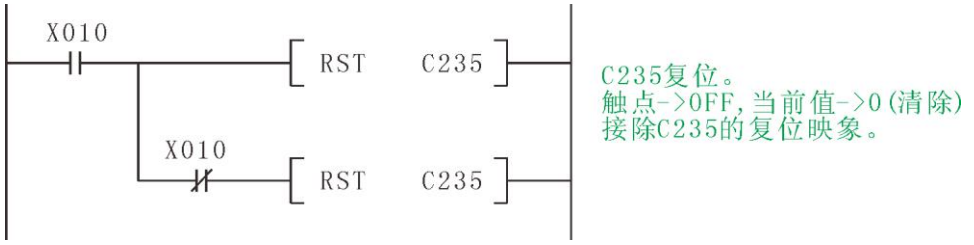
5、通过使高速计数器的输出线圈(OUT C***)ON/OFF.可以使计数开始/停止，但是请在主 程序中使用这种输出线圈进行编程。如果在步进梯形图(SFC)内和子程序、中断子程序内用这种线圈编程时，到执行这些步进梯形图和子程序以前，都不可以执行计数和停止。

6、输入到高速计数器中的信号，不能超过上述的响应频率。如果输入了超出这个频率的信号时，可能会使 WDT 出错，且并联链接不能正常运行。请务必注意。

7、使用 RST 指令对高速计数器进行复位时，换行到 RST 指令的驱动 OFF 之前，高速计数器都不能进行计数。希望执行「允许仅清除当前值的情况」和触点的 OFF 和当前值的复位!时，请按照下面的内容操作。允许仅清除当前值的情况



触点的 OFF+清除当前值的情况



2-8 数据寄存器、文件寄存器【D】

数据寄存器就是保存数值数据用的软元件，文件寄存器是处理这种数据寄存器的初始值的软元件。全都是 16 位数据(最高位为正负符号)，将 2 个数据寄存器、文件寄存器组合后可以保存 32 位(最高位为正负符号)的数值数据。

➢数据寄存器、文件寄存器的编号

数据寄存器、文件寄存器(D)的编号如下表所示。（编号以 10 进制数分配）

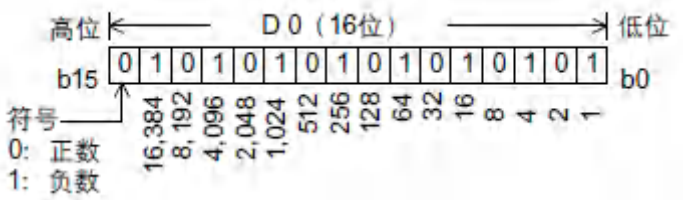
	数据寄存器				文件寄存器（保持）
	一般用	停电保持用（电池保持）	停电保持用（电池保持）	特殊用	
CZK3・CZK3C 可编程控制器	D0~D199 200 点※1	D200~D 312 点※2	D512~D7999 7488 点※3※4	D8000~D8511 512 点	D1000※4 以后最大 7000 点

※1 非停电保持区域。通过设定参数可以更改为停电保持(保持)区域。
※2 停电保持区域(保持通过设定参数，可以更改为非原电保持区域。
※3 关于停电保持的特性不能通过参数进行变更。
※4 通过设定常数，可以将 D1000 以后的数据寄存器以 500 点为单位作为文件寄存器。
使用简易 PC 间链接和并联链接的时候一部分的数据寄存器被占用为经接用。参考通信控制手册。

➢数据寄存器、文件寄存器的构造

1)16 位

1 个（16 位）数据寄存器、文件寄存器可以处理-32768~+32767 的数值。



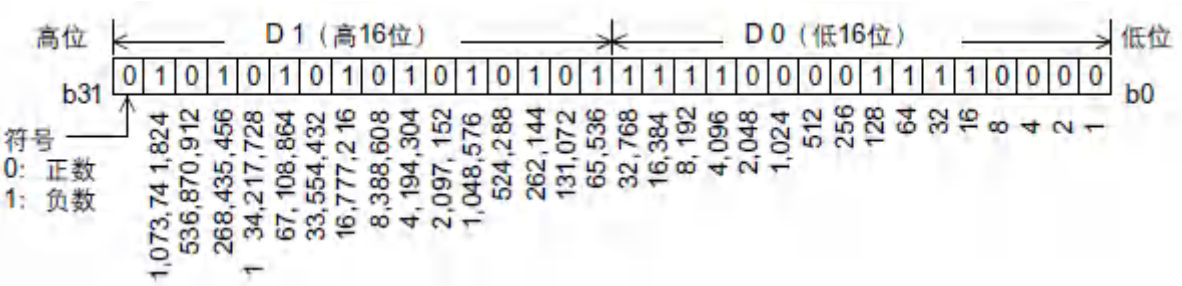
一般情况下，使用应用指令对数据寄存器的数值进行读出/写入。
此外，也可以通过人机界面、显示模块、编程工具直接进行读出/写入。

2）32 位

使用 2 个相邻的数据寄存器、文件寄存器，显示 32 位数据。

数据寄存器的高位编号大，地位编号小。变址寄存器的 V 为高位，Z 为地位。

据此，可以处理-2,147,483,648~+2,147,483,647 的数值。



指定 32 位时，如指定了低位侧(例如: DO)，高位侧就自动占有紧接的号码(例如: D1)。

低位侧既可指定奇数，也可指定偶数的软元件编号，但是考虑到人机界面、显示模块、编程工具的监控功能等，建议低位侧取偶数的软元件编号。

➢ 数据寄存器的功能和动作实例

数据寄存器就是保存数值数据用的软元件。

该软元件为 16 位数据(最高位为正负符号)，但是组合 2 个软元件后可以保存 32 位(最高位为正负符号)的数值数据。

● 一般用/停电保持用

1）数据寄存器中的数据旦被写入，在其他数据未被写入之前都不变化。

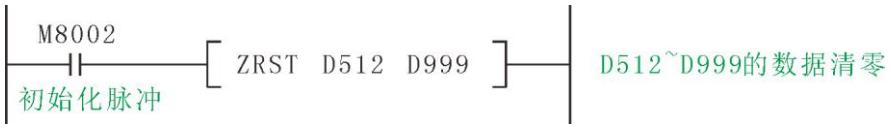
在 RUN→ STOP 时以及停电时，一般用数据寄存器的所有数据都被清除为 0。

但是，如果驱动特殊辅功继电器 M8033,即使 RUN 一 STOP 时也能保持。

2）停电保持(保持) 用数据寄存器，在 RUN/STOP 以 及停电时都保持其内容。

3）数据寄存器的停电保持是通过可编程控制器内置的后备用电池执行的。

4）将停电保持专用的数据寄存器作为一般用使用时，请使用 RST.或是 ZRST 指令在程序的开头步中设置如下所示的复位梯形图。



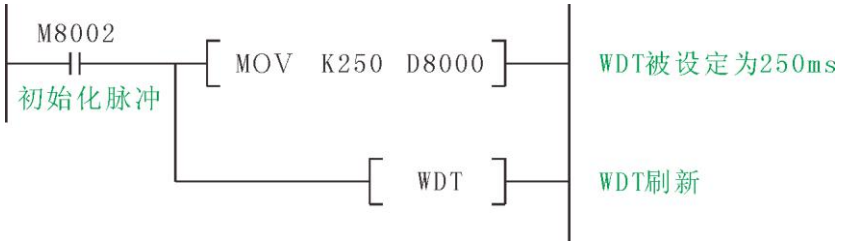
● 特殊用

1）写入特定目的的数据，预先写入特定的内容的数据寄存器。

该内容在每次上电时会被设置为初始值。

(一般被清零，带初始值的通过系统 ROM 被写入。)

2）例如，系统 ROM 对 D8000 中的 WDT 时间进行初始设定，但如果要更改，使用传送指令 MOV (FNC12)可以向 D8000 中写入目的时间。



● 动作实例

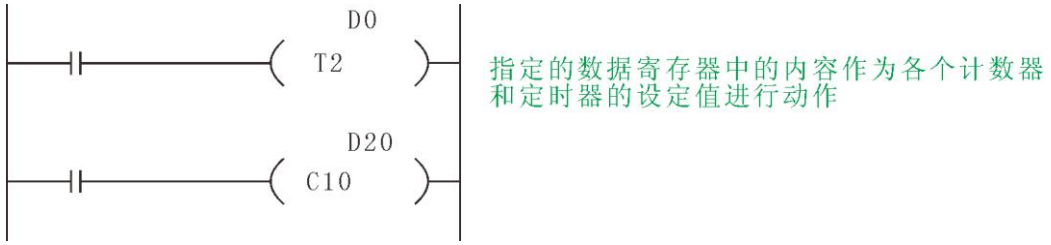
数据寄存器可以处理数值数据，用于各种控制。

在本项中，从这些用途中选取了基本指令和应用指令的代表例说明动作。

此外，为了能够更有效使用数据寄存器，请阅读后面的应用指令说明。

1)基本指令中的数据寄存器

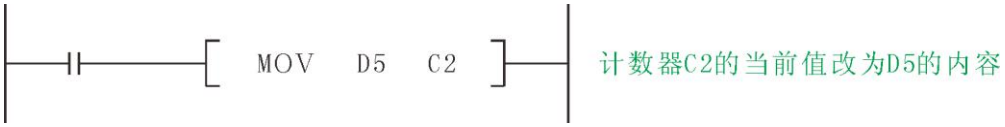
指定为定时器和计数器的设定值。



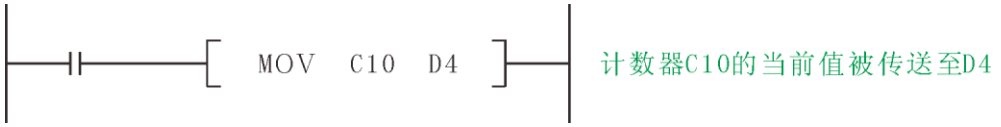
2）应用指令中的数据寄存器

FNC 12(MOV)指令的动作实例

2.1）更改计数器的当前值。



2.2）将定时器和计数器的当前值读出到数据寄存器中。



2.3）数值保存在数据寄存器中。

16位数值

32位数值

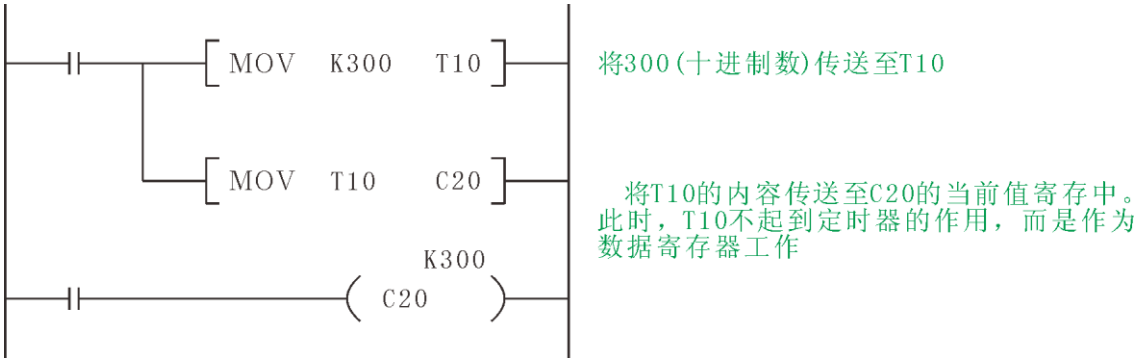
将80000（十进制数）传送至D10(D11)。
由于超出32767的数值是32位的，所以使用
32位运算。数据寄存器指定了低位侧（D10）后，
高位侧（D11）会自动被占用

2.4）将数据寄存器的内容传送至其他数据寄存器中。

3)将未使用的定时器及计数据作为数据存器使用。

FNC12 - MOV 指令的动作实例

程序中不使用的定时器和计数器可作为 16 位或是 32 位的数值保存软元件(数据寄存器)使用。



作为 32 位使用时，与数据寄存器相同、用 2 个 16 位软元件(例如: C1、C0 等)，显示 32 位数值。此外，1 个 32 位计数器(例如: C200 等) 能够处理 32 位的数值。

➢文件寄存器的功能和动作实例

文件寄存器，是对相同软元件编号的数据寄存器设定初始值的软元件。

这个软元件也和数据寄存器相同，是 16 位数据(最高位为正负符号)，但是组合 2 个软元件后可以保存 32 位(最高位为正负符号)的数值数据。

通过设定参数，可以将数据寄存器 D1000 以后的停电保持专用的数据寄存器设定为文件寄存器。最多可设定 7000 点。

1）参数的设定，可以指定 1- 14 个块(每个块相当于 500 点的文件寄存器)，但是这样每个块就减少了 500 步的程序内存区域。

2）希望将 D1000 以后的一部分设定为文件寄存器时，剩余的寄存器可以作为停电保持专用的数据寄存器使用。

1、文件寄存器的作用

1）当可编程控制器上电时和 STOP -> RUN 时，在内置存储器、或是存储器选件中设定的文件寄存器区域([A]部)会被一并传送至系统 RAM 的数据内存区域[B]部中。

因此，数据寄存器区域[B]部为停电保持软元件，如通过参数设定为文件寄存器，当可编程控制器上电时或 STOP -> RUN 时，程序内存中的文件寄存器区域[A]部会被传送。因此，执行电源复位或者 STOP -> RUN 的操作后，在数据内存中更改的内容会被初始化。

如果需要通过顺控程序，在数据内存中保存更改的数据时，请利用后述的 BMOV(FNC 15)指令的同编号寄存器更新模式，将文件寄存器区域[A]部更新成更改后的值。

2）FNC15 - BMOV 指令和其他指令的区别

针对文件寄存器(D1000 以后)的 FNC15 - BMOV 指令和其他指令的区别如下表所示。

指令	传送内容	备注
BMOV 指令	以对程序内存中的文件寄存器区域[A]部进行读出/写入。	-
BMOV 指令以外的应用指令等	针对内存映像区中的数据寄存器区域[B]部，采用与一般的数据寄存器相同的处理，可以进行读出/写入。	数据寄存器区域[B]部是设置在可编程控制器的系统 RAM 中的，因此可以不受选件内存的型号眼制，随意更改内容。

设定为文件寄存器的数据寄存器，在上电时会自动地将数据从文件寄存器区域[A]部复制到数据寄存器区域[B]部中。

通过外围设备对文件寄存器进行监控时，读出数据内存中的数据寄存器区域[B]部。

此外，在外围设备上执行文件寄存器软元件的[更改当前值]。[强制复位或是 IPC 存储器的全部清除]的时候，先对程序内存中的文件寄存器区域[A]部进行更改，然后自动传送给数据寄存器区域[B]部。

因此，执行文件寄存器软元件改写时，程序内存需要在内置存储器(RAM)]或是[存储器盒(闪存)]的"写保护开关 OFF 的状态。(存储器盒(闪存)的写保护开关如果为 ON 就不能从外围设备上进行更改。

2、文件寄存器←→数据寄存器<使用 BMOV(FNC 15)指令更新相同编号>

数据寄存器可以处理数值数据如 BMOV(FNV 15)指令的 S•, D•都是定为相同的文件寄存器，该指令就会变成同编号寄存器更新模式。

1）更新相同编号的文件寄存器的时候，必须将文件寄存器的编号设定为 S.=D.。

此外，设定的时候以 n 指定的传送点数不能超出文件寄存器区域。如超出文件寄存器区域，会出运算错误，而不能执行指令。

2）对 S., D.采用变址修饰时，实际的软元件编号要在文件寄存器区域内，与此同时，只有当传送点数在文件寄存器区域范围内，才能执行指令。

3、文件寄存器←→数据寄存器(使用 FNC15 - BMOV 指令写入)

对 FNC15 - BMOV 指令的目标操作数指定了文件寄存器（D1000 以后）时，可以直接写入程序内存的文件寄存器区域[A]部。

1）X001 为 ON 后，将数据传送至数据寄存器区域[B]部和文件寄存器区域[A]部中。此外，在[A]部中存储器盒(闪存)的写保护开关为 ON 而不能写入的情况下，只写入[B]部。使用一般应用指令，在 D D 中指定文件寄存器软元件的时候，只将数据传送到数据寄存器区域[B]部中。

2）也可以在 SD 中指定文件寄存器，但是如果指定了和 D D 相同的编号时，就变成相同编号更新模式。

3）通过控制 BMOV(FNV 15)的 BMOV 反向传送 M8024，可以用一个程序向两个反向传送。

读出的注意事项

即使对 BMOV(FNC 15)指令的源操作数指定文件寄存器(D1000 以后)，如在目标操作数中不指定相同编号的文件寄存器(同编号寄存器更新模式以外)、不会读出程序内存中的文件寄存器区域[A]部的内容。

1）在源操作数中指定文件寄存器，目标操作数中指定数据寄存器的情况。

2）源操作数和目标操作数指定了不同的软元件编号的文件寄存器的情况。

➢使用文件寄存器的注意事项

1、使用存储器的注意事项

更改存储器盒闪存中的文件寄存器内容时，请按照以下条件执行。

1）存储器盒的写保护开关请于 OFF 一侧。

2）闪存的允许写入次数在 1 万次以下。

通过程序进行写入时，如使用连续执行的指令，在可编程控制器的每个扫描周期中都会对闪存写入。如要避免这种情况，必须使用脉冲执行型(BMOVP)指令。

3）闪存的写入，一个连续的块(500 点)需要中 66 132ms。在此之间的程序执行被中断。由于此时的 WD 被刷新，所以需要采取在用户程序中插入 WDT 指令等的措施。

2、用 BMOV(FNC 15)指令的相同编号更新模式使用文件寄存器时的注意事项

1）更新相同编号的文件寄存器时，必须将文件寄存器的编号设定为 S・=D・。

2）n 指定的传送点数请勿超出文件寄存器的范围。

3）超出文件寄存器范围的情况下，会出运算错误(M8067) 不执行指令。

4）变址修饰的情况对 S・, D・进行变址修饰时，实际的软元件编号要在文件寄存器区域内，同时，只有传送点数在文件寄存器范围内，才能执行指令。

2-9 文件寄存【R】、扩展文件寄存器【ER】

文件寄存器(R)是扩展数据寄存器(D)用的软元件。停电保持。

此外，使用存储器盒时，文件寄存器(R)的内容也可以保存在扩展文件寄存器(ER)中。但是，只有在使用了存储器盒的情况下才可以使用这种扩展文件寄存器。

➢文件寄存器（R）、扩展文件寄存器（ER）的编号

寄存器 (R)和扩展文件寄存器 (ER)的编号如下表所示。 (编号按 10 进制数分配)。

	文件寄存器（R）【保持用】	扩展文件寄存器（ER）【文件用】
CZK2/CZK3 系列可控编程控制器	R0~R32767 共 32768 点	ER0~ER32767 共 32768 点※1

※1 仅在使用存储器盒的时候可以使用(保存在存储器盒的闪存中)。

➢数据的存储地点和访问方法

由于文件寄存器(R)和扩展文件春行器保存数据用的内存不同， 因此访问的方法也如下表所示不同。

数据存储地点

软元件	数据存储地点
文件寄存器	内置 RAM
扩展文件寄存器	存储器

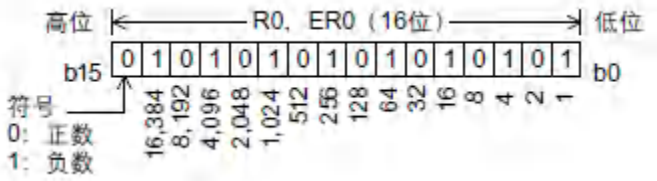
访问方法的差异

访问方法		文件寄存器	扩展文件寄存器
程序中读出		○	△仅专业指令可以
程序中写入		○	△仅专业指令可以
显示模块		○	○
数据的变更方法	GX Developer 的在线测试操作	○	×
	使用 GX Developer 进行成批写入	○	○
	计算机链接功能	○	×

文件寄存器、扩展文件寄存器的构造

文件寄存器由 1 点 16 位构成。这种软元件和数据寄存器相同，可以在应用指令等中用 16 位/32 位运算指令进行处理。

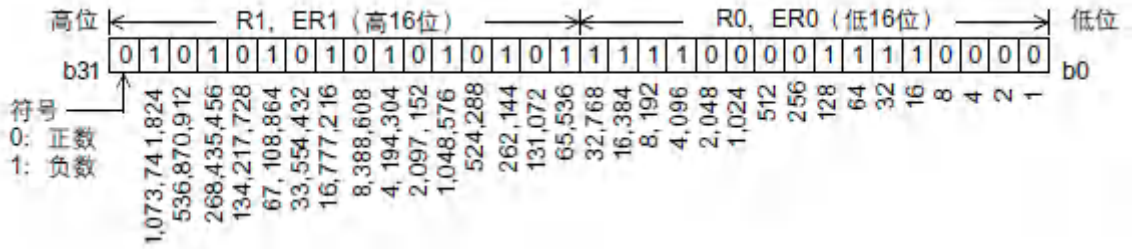
1)16 位，1 个文件寄存器(16 位)中，可以处理-32768~ +32768 的数值。



一般使用应用指令对文件寄存器进行数值的读出写入。

此外，也可以用人机界面、显示模块，编程工具直接进行读出写入。

2)32 位，使用相邻的 2 个文件寄存器，显示 32 位数据。(高位编号大，低位编号小)。因此，可以处理 2147483648~ +2147483647 的数值。



指定 32 位时，指定低位侧(例如: RO)后高位侧会被紧接(例如: R1)的号码自动占用。

低位侧中可以指定奇数、或偶数的软元件编号，但是考虑到人机界面，显示模块、编程工具的监控功能等，建议在低位侧使用偶数的软元件编号。

➢文件寄存器，扩展文件寄存器的初始化

即使执行了【电源 OFF】和【STOP→RUN 的操作】，文件寄存器的内容也通过内置电池被保持。

对文件寄存器的内容初始化时，请通过使用顺控程序或是 GX Developer 执行数据清除的操作。

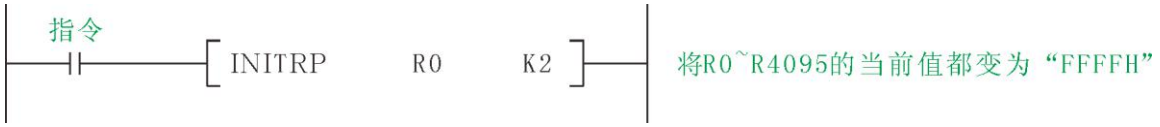
1、在程序中执行的方法

1）一部分的文件寄存器(R)的初始化

例如)将 R0~ R199 初始化(清除)的时候

2）以段为单位初始化文件寄存器以及扩展文件寄存器

例如)初始化 R0~ R4095 和 ER0~ ER4095 (初始化 R0 和 ER0 的起始的 2 个段) 时



2、在 GX Developer 中执行的方法

在 GX Developer 中，选择[Online]→ [Clear PLC memory]后，清除[Data device]。

但是，执行这个操作的时候，定时器、计数器、数据寄存器、文件寄存器以及文件寄存器的内容都被初始化。

➢文件寄存器的功能和动作实例

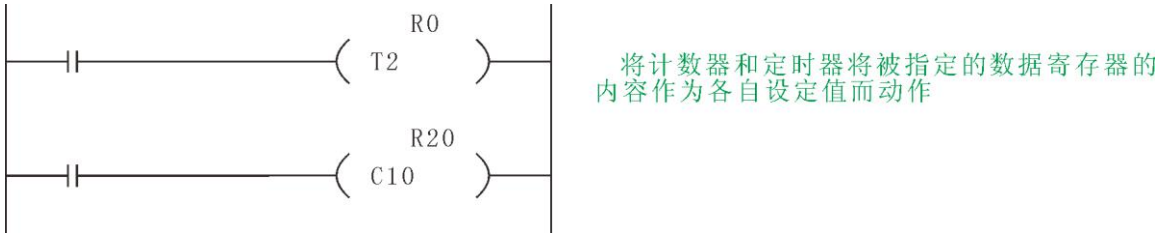
文件寄存器和数据寄存器相同，都可以用于处理数值数据的各种控制。

在本项中，从这些用途中选取基本指令和应用指令的代表对动作进行说明。

此外，为了能有效使用文件寄存器，请阅读后述的应用指令说明。

1、基本指令中的文件寄存器

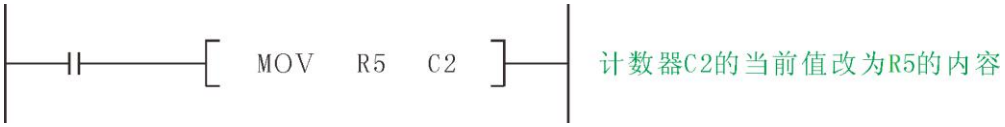
1）指定为定时器和计数器的设定值



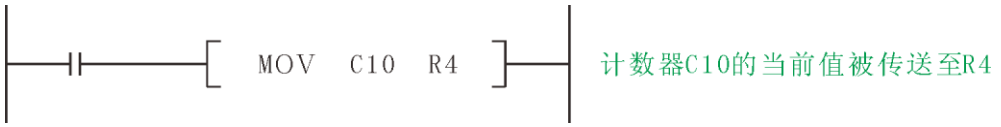
2、应用指令中的文件寄存器

FNC12 - MOV 指令的动作实例

1）更改计数器的当前值。

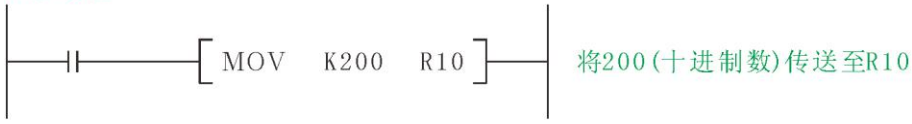


2）将定时器和计数器的当前值读入文件寄存器中。

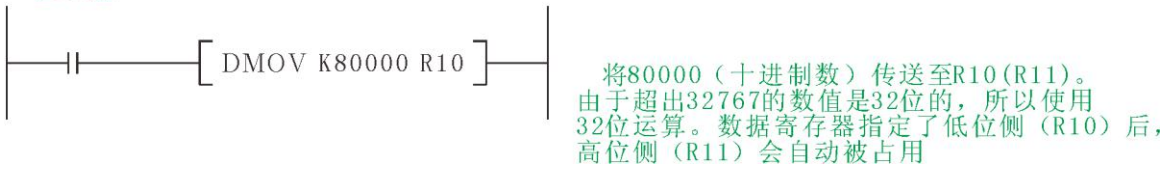


3）将数值保存到文件寄存器中。

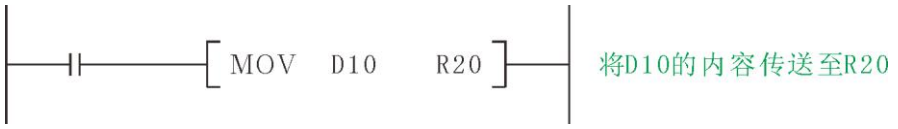
16位数值



32位数值



4）将数据寄存器的内容传送到其他文件寄存器中.



➢扩展文件寄存器的功能和动作实例

扩展文件寄存器(ER)，通常可以作为记录数据的保存位置和设定数据的保存位使用。

只有通过下表中的专用指令才可使用这种软元件。如果通过其他的指令使用数据内容时，请在将内容读出到相同软元件编号的文件寄存器中后，再使用文件寄存器一侧的软元件。

但是，只有当使用了存储器盒时方可使用这种软元件。

指令	内容
FNC290 - LOADR	将扩展文件寄存器(ER)的数据读出到文件寄存器(R)中的指令
FNC291 - SANER	将文件寄存器(R)的数据以 2048 点(1 段)为单位写入(传送)到扩展文件寄存器(ER)中的指令。用于将新制作的 1 段(2048 点)的数据保存到扩展文件寄存器(ER)※1 的情况.
FNC292 - INITR	文件寄存器(R)以及扩展文件寄存器(ER)※1 中以 2048 点(1 段)为单位进行初始化的指令。使用 LOGR 指令开始记录数据前，对文件寄存器(R)以及扩展文件寄存器(ER)※1 进行初始化时，使用该指令。
FNC293 - LOGR	记录指定数据，写入到文件寄存器(R)※1 和扩展文件寄存器(ER)中的指令。
FNC294 - RWER	将指定的文件寄存器(R)写入(传送)到扩展文件寄存器(ER)※1 中的指令。CZK2/CZK3 系列支持。将任意的文件寄存器(R)的内容保存到扩展文件寄存器(ER)※1 中时，使用该指令。
FNC2905 - INITER	以 2048 点(1 段)为单位对扩展文件寄存器(ER)※1 进行初始化的指令。CZK2/CZK3 系列的执行 SAVER 指令前对扩展文件寄存器(ER)※1 进行初始化时，使用该指令。

※1 只有使用存储器盒的时候才可访问扩展文件寄存器。

1、扩展文件寄存器和文件寄存器的关系

在可编程控制器中,扩展文件寄存器 R 和文件寄存器 ER 可通过 LOGR、INITER、SAVER、RWER、LOADR 指令相互关联。

2、有关文件寄存器和扩展文件寄存器的段

在数据结构上，文件寄存器和扩展文件寄存器中都有段。每 1 段是由 2048 点软元件构成的，各段的起始软元件如下表所示。

段编号	起始软元件编号	软元件范围
段 0	R0	ER0~ER2047，R0~R2047
段 1	R2048	ER2048~ER4095，R2048~R4095
段 2	R4096	ER4096~ER6143，R4096~R6143
段 3	R6144	ER6144~ER8191，R6144~R8191
段 4	R8192	ER8192~ER10239，R8192~R10239
段 5	R10240	ER10240~ER12287，R10240~R12287
段 6	R12288	ER12288~ER14335，R12288~R14335
段 7	R14336	ER14336~ER16338，R14336~R16338
段 8	R16384	ER16384~ER18431，R16384~R18431
段 9	R18432	ER18432~ER20479，R18432~R20479
段 10	R20480	ER20480~E22527，R20480~R22527
段 11	R22528	ER22528~ER24575，R22528~R24575
段 12	R24576	ER24578~ER26623， R24576~R26623
段 13	R26624	ER26824~ER28671，R26624~R28671
段 14	R28672	ER28672~ER30719，R28672~R30719
段 15	R30720	ER30720~ER32767，.R30720~R32767

➢使用扩展文件寄存器的注意事项

1、对扩展文件寄存器写入数据时的注意事项

由于扩展文件寄存器是保存在存储器盒内的闪存中，因此请注意以下一些要点

1）使用 SAVER 指令将数据写入扩展文件寄存器中时

执行该指令前，请先将作为写入对象的段进行初始化。而且，请在初始化后，将写入数据保存到文件寄存器中。

CZK2/CZK3 系列使用 RWER 指令后无需对写入对象的段进行初始化。

2）使用 LOGR 指令，将数据写入扩展文件寄存器中时数据开始记录之前，请先对写入对象的段进行初始化。

3）使用 INITR 指令时

该指令是对被指定的段的文件寄存器以及扩展文件寄存器的内容进行初始化。如果仅仅使用该指令对扩展文件寄存器初始化时，必须在执行指令前，将文件寄存器的内容暂时保存到不使用的文件寄存器文件寄存器或不使用的数据寄存器中。

CZK2/CZK3 系列 PLC 仅仅对扩展文件寄存器初始化时，请使用 INITER 指令。

2、有关扩展文件寄存器的初始化

由于扩展文件寄存器的内容是保存在存储器盒的闪存中的。因此可以通过顺控程序或梯形图编程软件 GX Works2 或 GX Developer 中的数据清除操作来执行初始化。

1)在程序中执行的方法

1.1）以段为单位仅初始化扩展文件寄存器 ER

例如：FR0~ ER4095 (ER0 的起始 2 段的初始化）

1.2）以段为单位初始化文件寄存器 R 以及扩展文件寄存器 ER

例如：R0~R4095 和 ER0~ER4095 的初始化(R0 和 ER0 的起始 2 段的初始化)

2)在 GX Developer 中执行的方法

在 GX Developer 中,选择[Online]- [Clear PLC memory]后,清除[Data device]。但是,执行这个操作的时候,定时器、计数器、数据寄存器、文件寄存器以及扩展文件寄存器的内容都被初始化。

2-10 变址寄存器【V，Z】

变址寄存器是除了可与数据寄存器的使用方法相同以外，还可以通过在应用指令的操作数中组合使用其他的软元件编号和数值，从而在程序中更改软元件的编号和数值内容的特殊寄存器。

➢变址寄存器的编号

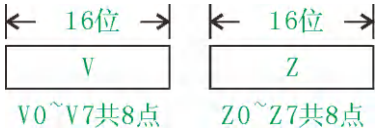
变址寄存器[V, Z]的编号如下表所示。(编号以 10 进制数分配)仅仅指定变址寄存器 V 或是 Z 的时候，分别作为 V0, Z0 处理。

	变址用
CZK2/CZK3 系列可编程控制器	VO(V)-V7, Z0(Z)~Z7 共 16 点※1

※1 有关停电保持的特性可以通过参数进行更改。

➢功能和构造

1、16 位，变址寄存器具有和数据寄存器相同的结构。



2、32 位，修饰 32 位的应用指令中的软元件时，或者及处理超出 16 位范围的数值时必须使用 Z0~ Z7。

如左图所示的V、Z组合，由于PLC将Z侧作为32位寄存器的低位侧动作，所以即使指定了高位侧的V0~V7也不会执行修饰。此外，作为32位指定时，会同时参考V(高位)，Z(低位)，因此一旦V(高位)侧中留存有别的用途中的数据时，会变成相当大的数值，从而出现运算错误。

即使32位应用指令中使用的变址值没有超出16位数据范围，也请按照左图所示在对Z进行数值的写入时，使用DMOV指令等的32位运算指令，同时改写V(高位)，Z(低位)。

➢软元件的修饰

可以被修饰的软元件，及其修饰的内容如下所示。

10 进制数软元件、数值: M、S、T、C、D、R、KnM、KnS、P、K。

例如，V0=K5,执行 D20V0 时， 对软元件编号为 D25 (D20+5)的执行指令。

此外，还可以修饰常数，指定 K30V0 时，被执行指令的是作为 10 进制的数值 K35 (30+5)。8 进制数软元件: X、Y、KnX、KnY。

例如，Z1=K8， 执行 X0Z1 时，对软元件编号为 X10 (X0+8:8 进制数加法)的执行指令。

对软元件编号为 8 进制数的软元件进行变址修饰时，V、Z 的内容也会被换算成 8 进制数后进行加法运算。因此，假定 Z1=K10, X0Z1 被指定为 X12，请务必注意此时不是 X10。16 进制数值: H

例如，V5=K30,指定常数 H30V5 时，被视为 H4E(30H+K30)。此外，V5=H30,指定常数 H30V5 时，被视为 H60(30H+30H)。

2-11 指针【P】，【I】

➢指针的编号

指针(P)、(1)的编号如下表所示。(编号以 10 进制数分配)

此外，使用输入中断用指针时，分配给指针的输入编号，不能和使用相同输入范围的[高速计数器]以及[脉冲密度(FNC56)]等一起使用。

系列	分支用		输入中断输入延迟中断用	定 时 器中断用	计数器中断用
		End 跳转用			
CZK2/CZK3 系列编程控制器	P0~P62 P64~P4095 4095 点	P63 1 点	I00□(X00) 130□X(003) I10□(X001) 140□(X004) I20□(X002) I50□(X005) 共 6 点	I6□□ I7□□ I8□□ 共 3 点	I010 I040 I020 I050 I030 I060 共 6 点

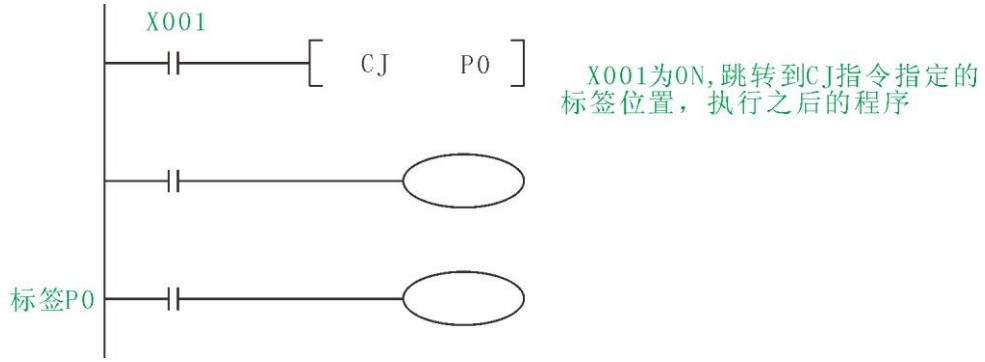
➢分支用指针的功能和动作实例

分支用指针的作用和动作如下所示。

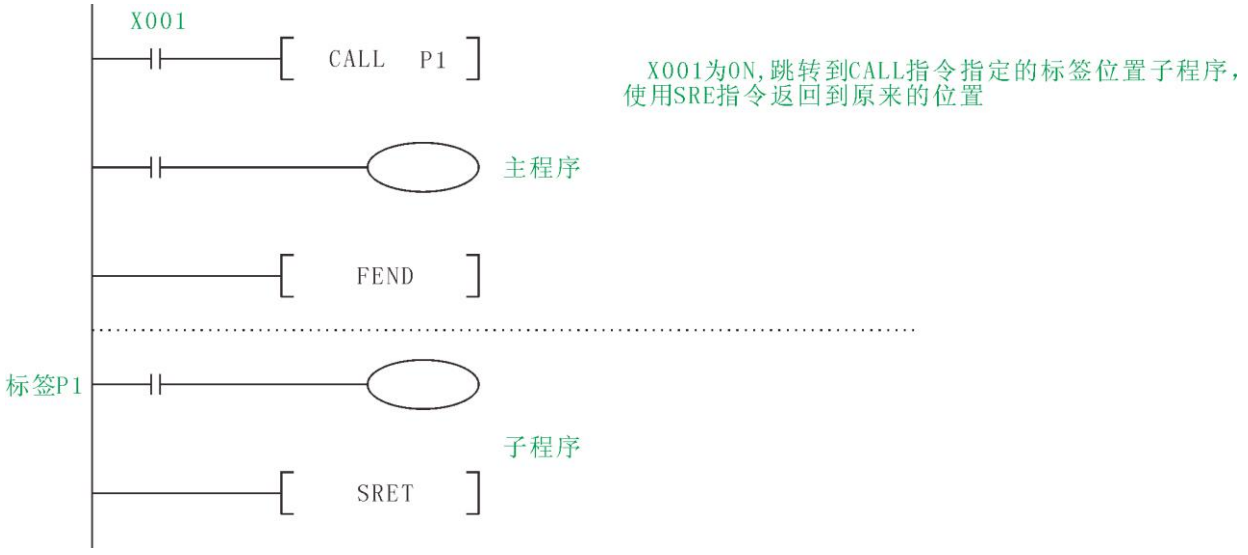
此外，这些指针都与应用指令组合使用，所以有关详细的使用方法请参考各指令的详细说明。

1、使用分支用指针(P)的应用指令

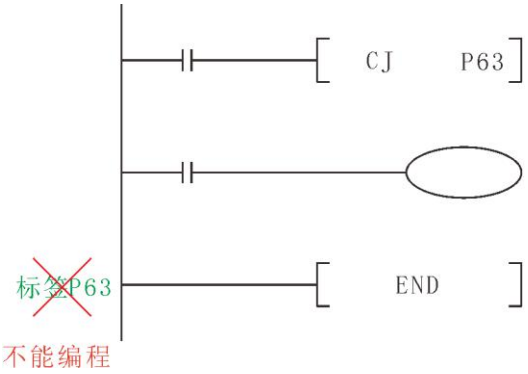
1）CJ(FNC 00)条件跳转



2）CALL(FNC 01)子程序调用



3）END 跳转用指针 P63 的作用



P63是表示使用CJ指令时跳跃到END步的特殊指针。
因此，将P63作为标签编程时，程序会出错，请注意。

➢**中断用指针的功能和动作实例**

中断用指针包括以下 3 种，与应用指令 IRET(FNC 03)中断返回、El(FNC 04)允许中断、和 DI(FNC 05)禁止中断一起使用。

1、输入中断(延迟中断)用：6 点

变址寄存器具有和数据寄存器相同的结构。

可以在不受可编程控制器扫描周期的影响下，接收来自特定的输入编号的输入信号。触发该输入信号，执行中断子程序。由于输入中断可以处理比扫描周期更短的信号，因此可在顺控过程中作为需要优先处理或者短时间脉冲处理控制时使用。

输入	输入中断用指令		禁止中断标志位	输入信号的 ON 脉宽或是 OFF 脉宽
	上升沿中断	下降沿中断		
X000	I001	I000	M8050※1	5μS 以上
X001	I101	I100	M8051※1	
X002	I201	I200	M8052※1	
X003	I301	I300	M8053※1	
X004	I401	I400	M8054※1	
X005	I501	I500	M8055※1	

※1 RUN→STOP 时清除

注意输入端子的重复使用(禁止)

输入 X000~ X007,用于高速计数器、输入中断、脉冲捕捉和 SPD、DSZR、DVIT、ZRN 指令及通用输入。因此请勿重复使用输入端子。

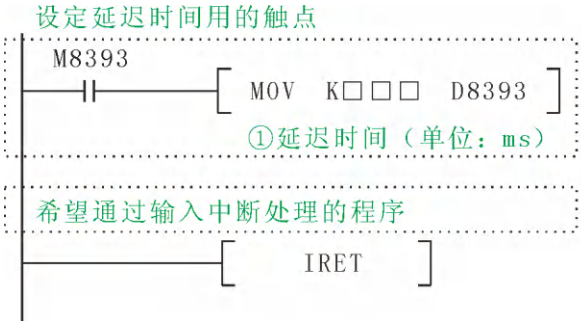
例如，使用输入中断指针【1001】时，由于 X000 被占用，所以不能使用【C235、C241、C244、C246、C247、C249、C251、C252、C254】,【输入中断指针 000】,【脉冲捕捉用触点 M8170】和【该输入的 SPD 指令】。

输入中断的延迟功能

在输入中断中，有以 1ms 为单位延迟执行中断子程序的功能。

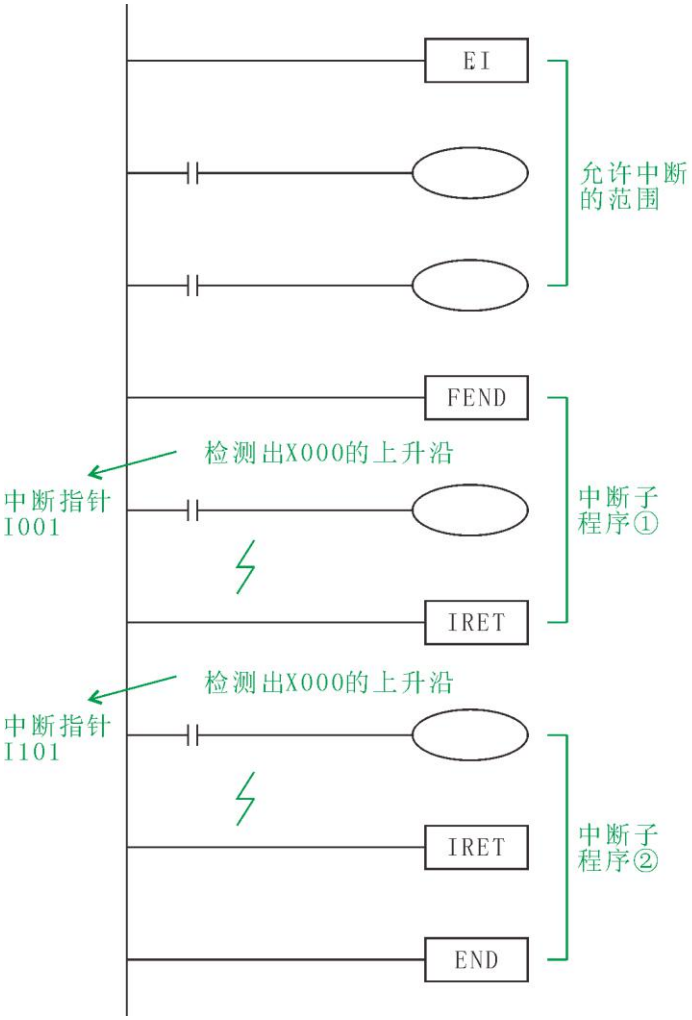
可以使用下面的模板程序来指定延迟时间。

使用了这种延迟功能后，在调节输入中断中使用的传感器的安装位置时，可以无需挪动实际的位置而进行电气上的调节。



指定延迟时间的程序
左边的指定延迟时间的程序，必须要写在中断子程序的开头处。
这个程序是模板程序，使用时只需要修改延迟时间①。
此外，这个时间的指定，只能使用常数K或是数据寄存器D。

动作



可编程控制器通常为禁止中断的状态.使用EI指令允许中断后，在烧写程序过程中，X000或X001为ON,执行中断子程序①和②，然后通过IRET指令返回到主程序

中断用指针（I□□□），在编程时请务必作为标签放在FEND指令后

2、定时器中断用：3 点

每隔指定的中断循环时间(10ms - 99ms)，执行中断子程序。

用于与可编程控制器的扫描周期不同的需要循环中断处理的控制中。

定时器中断用	中断周期（ms）	禁止中断标志位
I6□□	在指针名称的□□中，填写 10~99 的整数	M8056※1
I7□□	例如：I610=10ms 的定时中断	M8057※1
I8□□		M8058※1

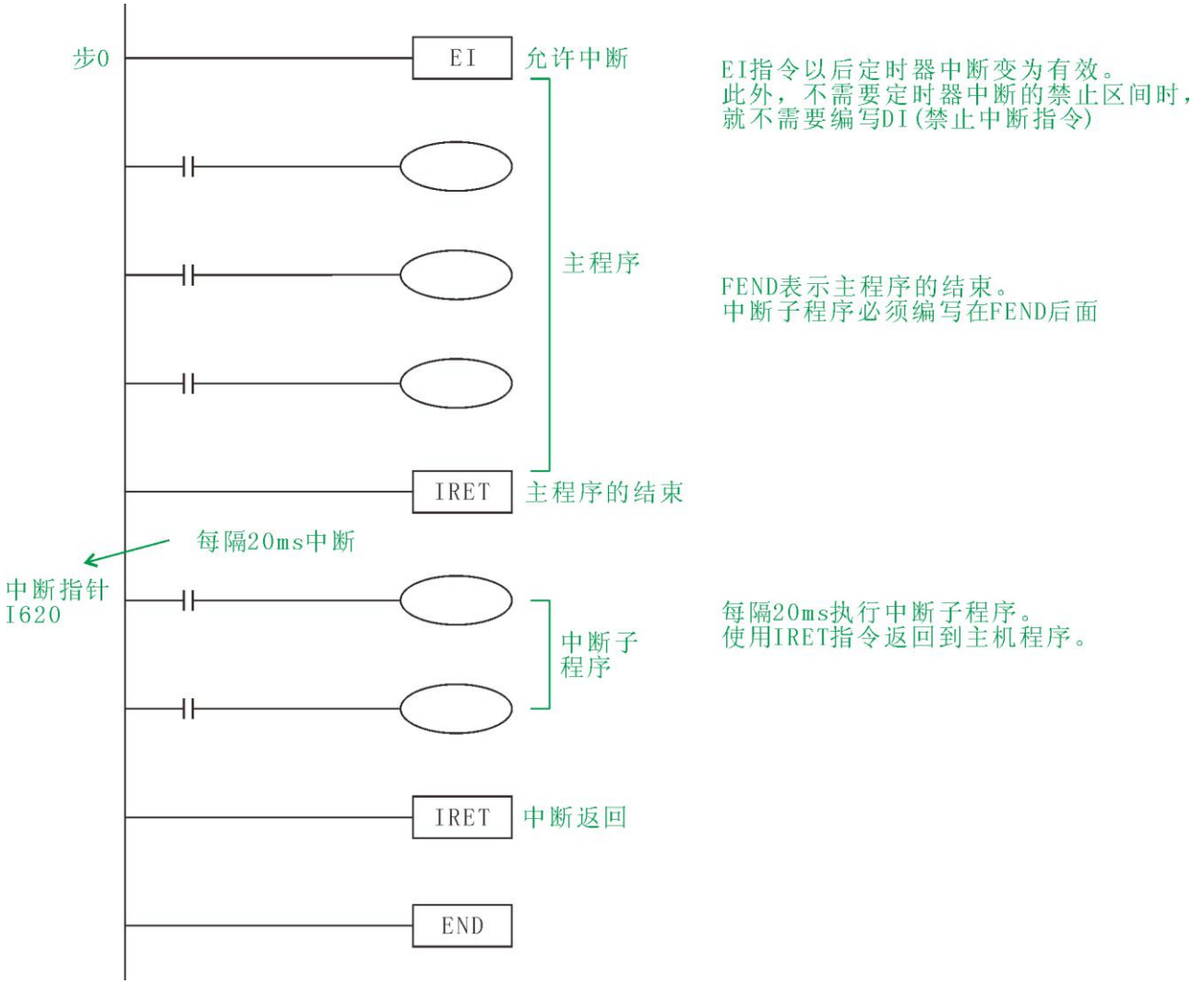
※1 RUN→STOP 时清除

注意事项

定时器中断时间设定在 9ms 以下时，在以下情况下有可能出现不能按照正确的周期处理定时器中断，所以建议在 10ms 以上使用。

1）中断程序的处理时间较长的情况

2）主程序内使用了处理时间较长的应用指令的情况



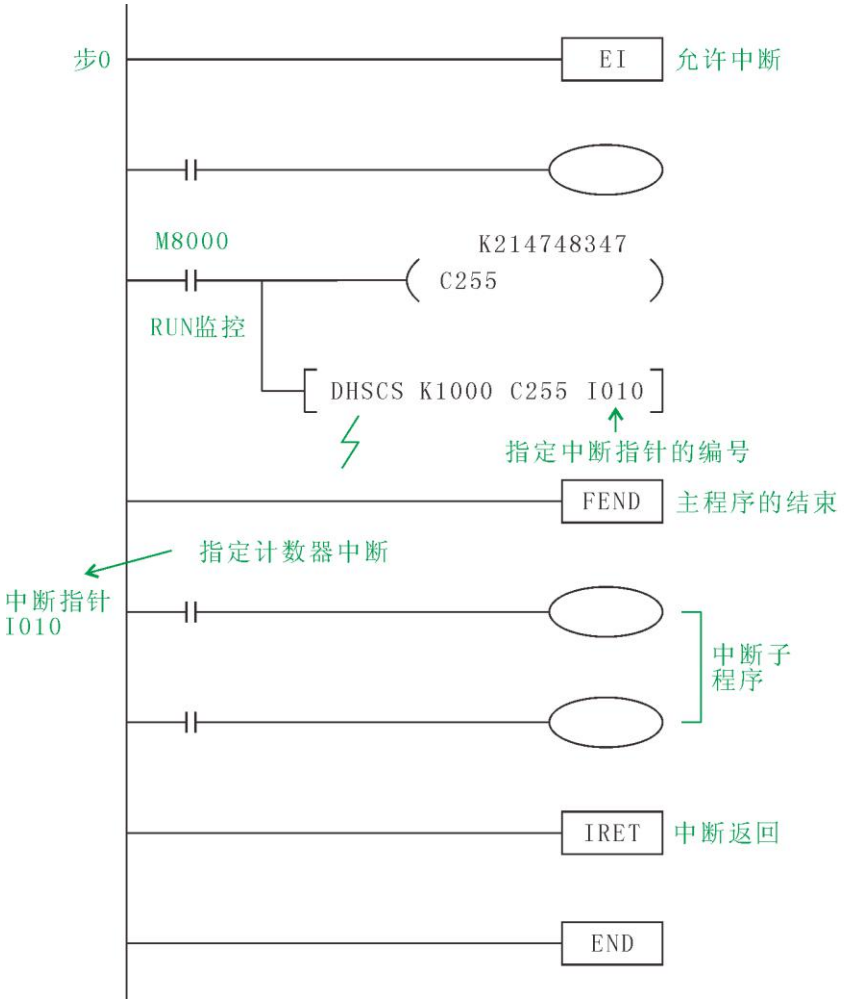
3、计数器中断用：6 点

根据高速计数器用比较置位指令(DHSCS 指令)的比较结果，执行中断子程序。用于使用高速计数器优先处理计数结果的控制。

指针编号	禁止中断标志位
I010	M8059※1
I020	
I030	
I040	M8059※1
I050	
I060	

※1 RUN→STOP 时清除

功能



EI指令以后允许中断,编写主程序。

驱动高速计数器的线圈，在DHSCS指令中指定中断指针。

C255的当前值在999->1000或1001->1000中变化的时候，执行中断子程序。使用中断程序的实例，请参考上述的输入中断。

第三章 指令的软元件・常数的指定方法

在本章中，说明了作为可编程控制器指令使用基础的顺控指令的源操作数和目标操作数的指定方法。

1、10 进制数，16 进制数和实数的常数指定。
2、位软元件的位数指定
3、数据寄存器的位位置指定
4、特殊 CZK2/CZK3 系列功能模块/单元的 BFM(缓冲存储区)的直接指定
5、附加了变址寄存器的变址修饰

3-1 可编程控制器处理的数据(8 进制数/10 进制数/16 进制数/实数)

在可编程控制器中，根据各自的用途和目的不同，有 5 种数值可供使用。这些数值的作用和功能如下所示。

➢数值的种类

1、10 进制数(DEC)

1）定时器和计数器的设定值(K 常数)
2）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)
3）应用指令的操作数中的数值指定和指令动作的指定(K 常数)

2、16 进制数(HEX)

应用指令的操作数中的数值指定和指令动作的指定(H 常数)

3、2 进制数(BIN)

对定时器、计数器或是数据寄存器的数值指定，是按照上述的 10 进制数和 16 进制数执行的，但是在可编程控制器的内部，这些数值都以 2 进制数进行处理。

此外，在外围设备上监控这些软元件的时候，会如下图所示，自动转换成 10 进制数后显示。 (也可以切换成 16 进制)。

3.1）负数的处理

在可编程控制器内部，负数是以 2 的补码来表现的。详细内容请参考 FNC29 - NEG 指令的说明。

4、8 进制数(OCT)

CZK2/CZK3 系列可编程控制器中，输入继电器、输出继电器的软元件编号都是以 8 进制数分配的。

由于在 8 进制数中，不存在[8,9]， 所以按(0~ 7, 10~17，...70~77，100~ 107)上升排列。

5、BCD

BCD 就是将构成 10 进制数的各位，上 0 - 9 的数值以 4 位的 BIN 来表现的方式。

由于各位便于使用，所以适用于 BCD 输出型的数字式开关和 7 段码显示器控制等用途中。

6、实数(浮点数)

CZK2/CZK3 系列可编程控制器中，具有能够执行高精度运算的浮点数运算功能。

采用 2 进制浮点数(实数)进行浮点运算，并采用了 10 进制浮点数(实数)进行监控。

➢数值的转换

CZK2/CZK3 系列可编程控制器中处理的数据，可以按照下表的内容进行转换。

10 进制数（DEC）	8 进制数（OCT）	16 进制数（HEX）	2 进制数（BIN）		BCD	
0	0	00	0000	0000	0000	0000
1	1	01	0000	0001	0000	0001
2	2	02	0000	0010	0000	0010
3	3	03	0000	0011	0000	0011
4	4	04	0000	0100	0000	0100
5	5	05	0000	0101	0000	0101
6	6	06	0000	0110	0000	0110
7	7	07	0000	0111	0000	0111
8	10	08	0000	1000	0000	1000
9	11	09	0000	1001	0000	1001
10	12	0A	0000	1010	0001	0000
11	13	0B	0000	1011	0001	0001
12	14	0C	0000	1100	0001	0010
13	15	0D	0000	1101	0001	0011
14	16	0E	0000	1110	0001	0100
15	17	0F	0000	1111	0001	0101
16	20	10	0001	0000	0001	0110
⋮	⋮	⋮	⋮	⋮	⋮	⋮
99	143	63	0110	0110	1001	1001
⋮	⋮	⋮	⋮	⋮	⋮	⋮

主要用途

10 进制数（DEC）	8 进制数（OCT）	16 进制数（HEX）	2 进制数（BIN）	BCD
常数 K 以及输入输出继电器以外的内部软元件编号	输入继电器、输出继电器的软元件编号	常数 H 等	可编程控制器内部的处理	BCD 数字式开关，7 段码显示器

➢浮点运算中数值的使用

浮点运算中的数值的处理

在可编程控制器内部使用 BIN 的整数值。

在整数的除法运算中，会得出例如 40+3=13 余 1 的答案。

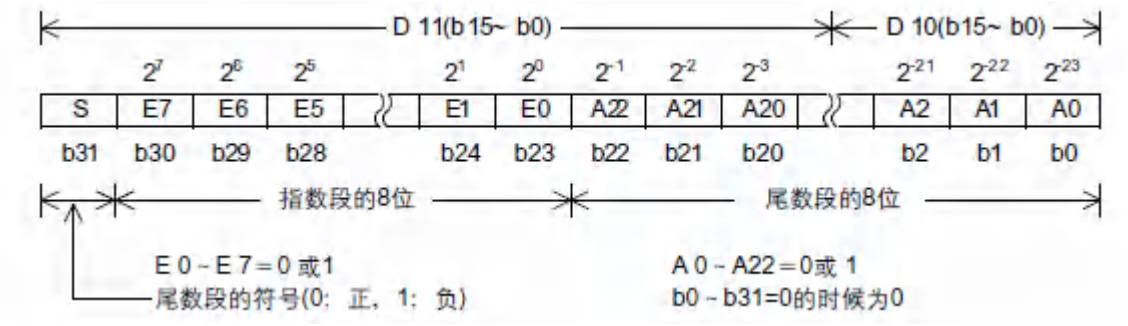
整数的开方运算中小数点也被舍去。

在 CZK2/CZK3 系列可编程控制器中，为了能够更加高精度地执行这些运算，可以执行浮点运算。

1、2 进制浮点数（实数）

在数据寄存器中处理 2 进制浮点数(实数)的时候，使用编号连续的一对数据寄存器。

例如(D11,D10)的时候，如下所示。



2 进制浮点数(实数) =±(2^0 +A22 × 2-1+A21x 2-2+ ..+A0× 2-23)
×2(E7×27+E6×26+... +E0× 20)/2^127

(例) A22= 1,A21=0, A20= 1,A19~ A0=0,E7=1,E6~E1=0, E0=1

2 进制浮点数(实数) =±(2^0 +1×2^-1+0×2^-2+1×2^-3+ ..+0×2^-23)
×2(1×2^7+0×2^6+... +1×2^0)/2 127
=±1.625×2129/ 2127=+1.625×22

正负由 b31 的符号决定，但不是补码的处理。

零(M8020)，借位(M8021)，进位(M8022)的处理。

在浮点扫描中的各种标志位的动作如下所示。

零标志:结果真为 0 时为 1;

借位标志:结果未达到最小单位，但不是 0 时为 1;

进位标志:结果的绝对值超出可以处理的数值时为 1。

2 进制浮点数(实数)的监控：

在 GX Developer 之类的支持了浮点数显示的编程软件中，可以直接监控 2 进制浮点数(实数)。

此外，在不支持浮点数显示的编程工具中，可以将 2 进制浮点数(实数)可以转换成 10 进制浮点数(实数)后再进行监控。

2、10 进制浮点数（实数）

由于对用户而言，2 进制浮点数(实数)是不易理解的数值，所以也可以将其转换成 10 进制浮点数(实数)。但是，内部的扫描仍然是采用 2 进制浮点数(实数)。

在数据寄存器中处理 10 进制浮点数(实数)的时候，使用编号连续的一对数据寄存器，但是与 2 进制浮点数(实数)不同，编号小的为尾数段，编号大的为指数段。

例如，使用数据寄存器(D1,D0)的时候如下所示，采用 MOV 指令对 D0 和 D1 进行写入。

10 进制浮点数(实数) =[尾数 D0]x 10[指数 D1]
尾数 D0=±(1000~9999)或 0
指数 D1=-41~ +35

总之，D0,D1 的最高位为正负符号位，都作为 2 的补码处理。

此外，在尾数 D0 中，例如不存在 100。在为 100 的时候，就变成 1000x10-1

10 进制浮点数(实数)在下面的指令中有效。

2 进制浮点数(实数)→10 进制浮点数(实数)的转换: FNC118 - 【D】EBCD

10 进制浮点数(实数)→2 进制浮点数(实数)的转换: FNC119 - 【D】EBIN

3-2 常数 K， H，E (10 进制数/16 进制数/实数)的指定

顺控程序中处理常数时，使用常数 K (10 进制数)、常数 H (16 进制数)或 E (浮点数)。

在编程用的外围设备中，有关指令上的数值操作中，10 进制数的数值中附加 K, 16 进制数的数值中附加 H，浮点数(实数)的数值中附加 E 后输入。(例如: 10 进制数 K100, 16 进制数 H64，实数 E1.23 或是 E1.23+10)。

➢常数 K (10 进制数)

【K】是表示 10 进制整数的符号。主要用于指定定时器和计数器的设定值，或是应用指令的操作数中的数值。(例如: K1234)

10 进制常数的指定范围如下所示。

1）使用字数据(16 位)时 K-32768～K32767

2）使用 2 个字数据(32 位)时 K-2147483648～K2147483647

➢常数 H (16 进制数)

【H】是表示 16 进制数的符号。主要用于指定应用指令的操作数的数值。(例如: H1234)

而且，各位数在 0 ～ 9 的范围内使用的时候，各位的状态(1 或 0)和 BCD 代码相同，因此可以指定 BCD 数据。(例如: H1234 以 BCD 指定数据时，请在 0～9 的范围内指定 16 进制数的各位数)16 进制常数的设定范围如下所示。

1）使用字数据(16 位)时.....H0~HFFFF (BCD 数据的时候为 H0～H9999)

2）使用 2 个字数据(32 位)时...H0 ~HFFFFFFFF (BCD 数据的时候为 H0～9999999)

➢常数 E (实数)

【E】是表示实数(浮点数)的符号。主要用于指定应用指令的操作数的数值。(例如: E1.234 或是 E1.234+3)

实数的指定范围为，-1.0x2^128~ -1.0x2^-126, 0, 1.0x2^-126~1.0x2^128。在顺控程序中，实数可以指定"普通表示"和”指数表示”两种。

1）普通表示，就将设定的数值指定。

例如，10.2345 就以 E10.2345 指定。

2）指数表示，设定的数值以(数值) x 10n 指定。

例如，1234 以 E1.234 + 3 指定。【E1.234+3】的【+3】表示 10 的 n 次方(+3 为 10^3)。

3-3 字符串

字符串中，包括在应用指令的操作数中直接指定字符串的字符串常数和字符串数据。

➢字符串常数("ABC")

字符串是顺控程序中直接指定字符串的软元件。

以" "框起来的半角字符(例如: "ABCD1234") 指定。

字符串中可以使用 JIS8 代码。但是，字符串最多可以指定 32 个字符。

➢字符串数据

字符串的数据，从指定软元件开始，到 NUL 代码(00H)为止以字节为单位被视为一个字符串。

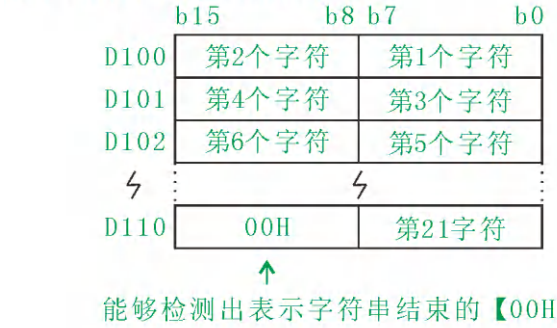
但是，在指定位数的位软元件中体现(认识)字符串数据的时候，由于指令为 16 位长度，所以包含指示字符串数据结束的 NUL 代码(00H)的数据也需要是 16 位。(参考下面 2 中的例 2)总之，如下情况下，应用指令中会出现扫描出错。(出错代码: K6706)

1、在应用指令的源程序中指定的软元件编号以后，相应软元件范围内未设定「00H」的情况。

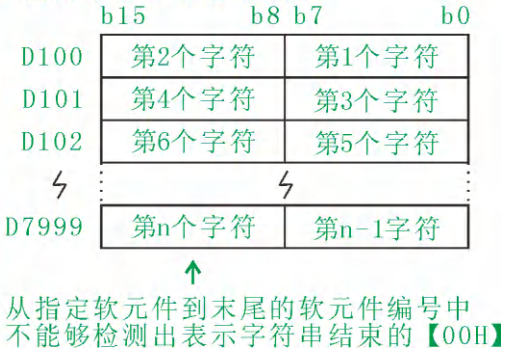
2、在应用指令的嵌套中指定的软元件中，保存字符串数据(包含了表示字符串数据的末尾的「00H」或「000H」)用的软元件数不够的情况。

1)字软元件中保存的字符串数据

能够识别为字符串数据的例子

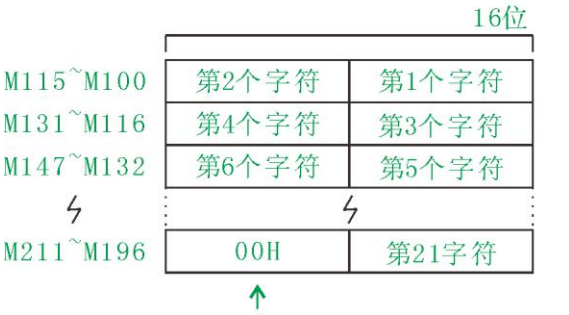


不能够识别为字符串数据的例子



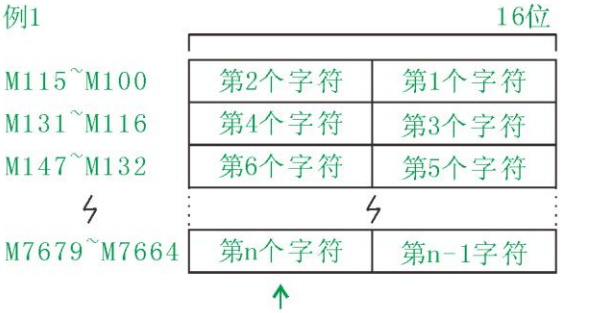
2)位数指定的位软元件中保存的字符串数据

能够识别为字符串数据的例子

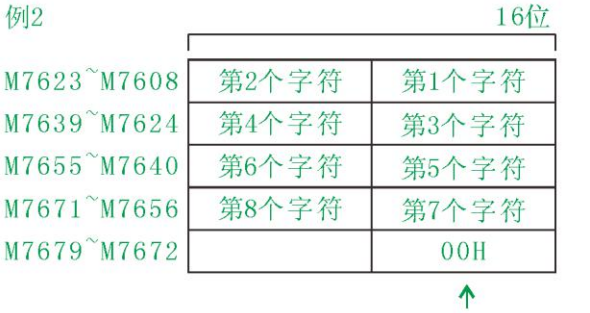


能够识别为字符串数据的例子

例1



例2



3-4 位的位数指定(Kn□***)

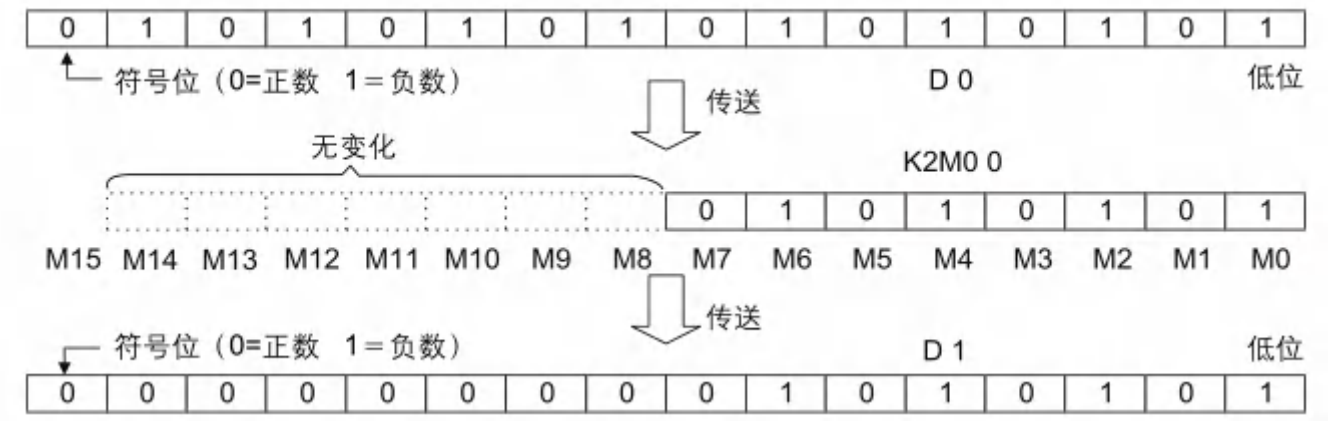
1、位软元件的处理

诸如 X,Y,M,S, 仅处理 ON/OFF 信息的软元件被称为位软元件。

与此相对，T,C,D,R 等处理数值的软元件被称为字软元件。

即使是位软元件，通过组合使用后也可以处理数值。这种情况下，以位数 Kn 和起始软元件的编号的组合来显示。位数为 4 位单位的 K1 ~K4 (16 位数据)、K1-K8 (32 位数据)。

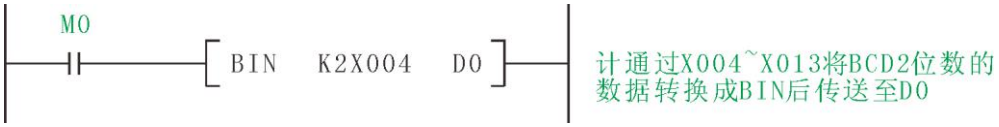
例如，K2M0



向 K1M0~K3M06 传送 16 位数据后，数据长度不足的高位部分不被传送。

32 位数据的情况相同。

在 16 位(或 32 位)的运算过程中，对位软元件指定 K1~K3 (或 K1~K7)的位数时，不足的高位被一直视为 0。因此，一直处理正数。



只要没有特别的限制，被指定的位软件的编号可以是任意的，但是建议在 X, Y 的场合，尽量将最低位的编号设置为 0。(指定 X000,X010,X020…Y000,Y010,Y020…等)

在 M，S 的场合，最理想是 8 的倍数，但是为了避免混乱，建议设定为 M0,M10,M20..等。

2、连续字的指定

所谓以 D1 开头的一连串的数据寄存器，就是 D1，D2，D3，D4.....

在字的场合，通过指定位数也可以将其作为一连串的字进行处理。如下图所示。

K1X000，K1X004，K1X010，K1X014…

K2Y000，K2Y020，K2X030…

K2M0，K3M12，K3M24，K3M36…

K4S16，K4S32，K4S48…

也就是说，在不跳过软元件的前提下，以各位数的单位如上述所示地使用软元件。

但是，32 位运算中使用了 K4Y000 的时候，高 16 位被视为 0。需要 32 位数据的时候，要使用 K8Y000。

3-5 字软元件的位指定(D□.b)

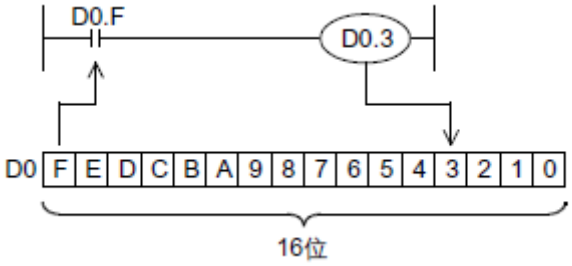
指定字软元件的位，可以将其作为位数据使用。

指定字软元件的位时，请使用字软元件编号和位编号(16 进制数)进行设定。(例如: D0.0 表示数据寄存器 D 的 0 位编号)

在软元件编号，位编号中不能执行变址修正。

对象的字软元件:数据寄存器或特殊数据寄存器

位编号: 0~F (16 进制)



3-6 缓冲存储器的直接指定(U□\G□)

可以直接指定特殊功能模块和特殊功能单元的 BFM(缓冲存储器)。BFM 为 16 位或 32 位的字数据，主要用于应用指令的操作数。

BFM 是接着特殊功能模块或特殊功能单元的模块号(U)和 BFM 编号(\G) 后指定的。

(例如: U0\G0…表示模块号为 0 的特殊功能模块或特殊功能单元的 BFM#0 号。)此外，在 BFM 编号中可以进行变址修正。指定范围如下所示。

模块号(U)： 0~7，BFM 编号(\G)： 0~32767。

MOV指令的例子

修正BFM编号的例子

3-6 变址修正

有关变址寄存器的功能和结构，已在[4.10 变址寄存器[V,Z]」中详细描述了，所以请事先阅读。

➢基本指令的变址修正

1、位软元件的情况

LD,LDI,AND,ANI,OR,ORI,OUT,SET,RST,PLS,PLF 指令中使用的位软元件[X,Y,M(特殊辅助继电器除外), T,C(0~199)都可以进行变址修正。

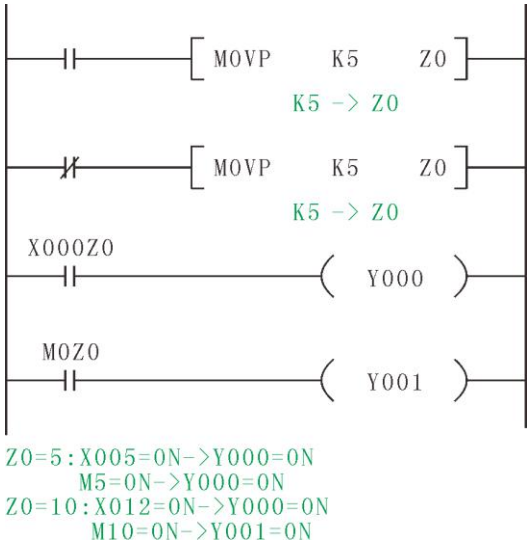
下面例举在变址寄存器 Z (0)中，对 LD 指令的 X000 和 M0

进行修正的例子来说明动作。如下图所示：

将 K5 或 K10 事先传送到变址寄存器 Z (0)中。

当 Z(0)=5 时，如果「M(0+5)=X005」后 X005 为 ON,则输出(ON)Y000、如果「M(0+5)=M5」后 M5 为 ON, 则输出(ON) Y001。

此外，当 Z(0)=10 时，如果「X(0+10)=X012※1」后 X012※1 为 ON,则输出(ON) Y000,如果「M(0+10)=M10」后 M10 为 ON,则输出(ON) Y001。



Z0=5:X005=0N->Y000=0N
M5=0N->Y000=0N
Z0=10:X012=0N->Y000=0N
M10=0N->Y001=0N

※1 请参考注意事项

1）修正的变址寄存器中，可以使用 Z0~Z7， V0~V7。

2）对于定时器，计数器的 OUT 指令，可以修正定时器编号，计数器编号和设定值中指定的软元件。

注意事项

1) 32 位计数器和特殊辅助继电器不能进行变址修正。

2) 16 位计数器进行变址修正后，不能作为 32 位的计数器处理。

3)变址修正 X，Y 的 8 进制数软元件编号的时候，对软元件编号进行变址修正的内容以 8 进制数换算进行加法运算。例如，在输入 X000 上附加的变址修正值星 K0, K8, K16 变化的情况下，输入 X000 也会按照「X(000+0)=X000」「(X(000+8)=X10」,「X(000+16)=X20」和 8 进制数换算后，对软元件编号进行加法运算后改变。

2、字软元件、常数的情况

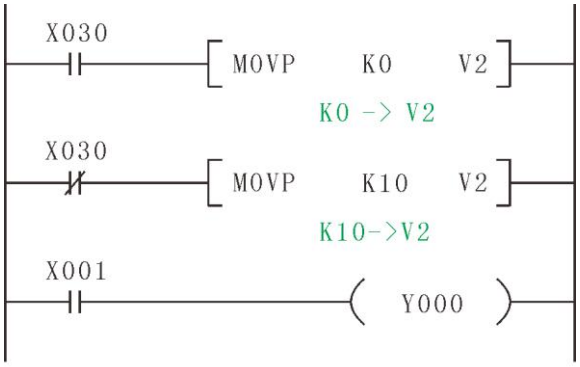
可以变址修正在 OUT 指令中使用的位软元件[T,C]的设定值。

下面例举在变址寄存器 V2 中， 修正 OUT 指令 TO 的设定值 D0 的例子进行说明。如下图所示：

将 K0 或 K10 事先传送到变址寄存器 V2 中。X001 为 ON, V2=0 的时候，如果「D(0+0)=D0」,设定值为 D0,则 T0 动作。此外，V2=10 的时候，如果「 D(0+10)=D10」，设定值为 D10，则 TO 动作。

注意事项

1）OUT 指令为 32 位计数器的时候，不能变址修正设定值。

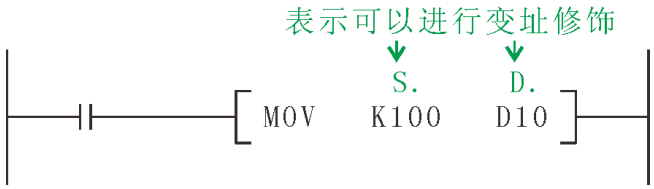


V2=0:T0的设定值为D0的当前值
V2=10:T0的设定值为D10的当前值

➢应用指令的变址修饰

1、可以变址修饰应用指令的标明方法

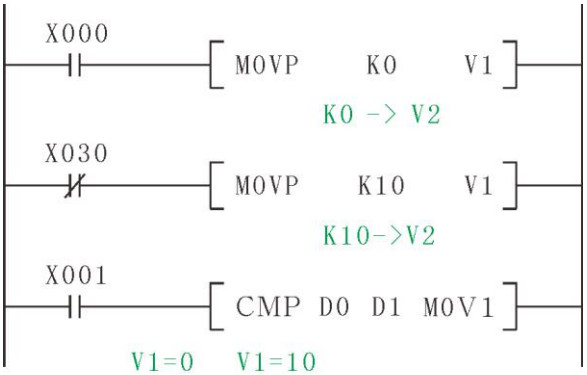
在应用指令说明中，可以进行变址修饰的操作数的表示方法如下图所示，通过在源操作数 S.或是目标操作数符号 D.中加上「.」，以此和不带修饰功能的操作数进行区别。



2、位软元件的情况

用变址寄存器 V1 对 CMP(FNC 10)的 比较结果进行修饰，以此为例说明。

将 K0 或 K10 事先传送到变址寄存器 V1 中。当 X001 为 ON, V1=0 的时候，则「M(0+0)=M0」，比较结果输出到 M0～M2 中。此外，V1=10 时，则「M(0+10)=M10」 ，比较结果输出到 M10~ M12 中。在进行修饰的 变址寄存器中，可以使用 Z0~Z7, V0~V7。

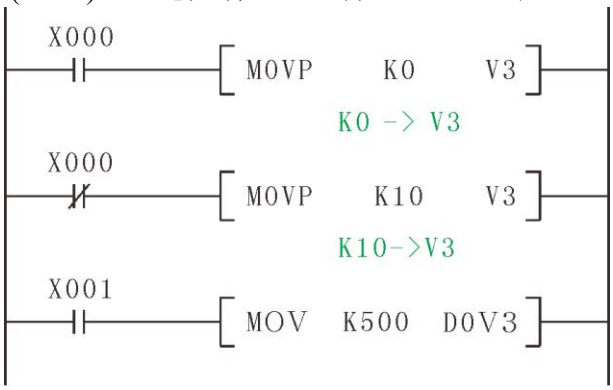


D0>D1->M0=ON; M10=ON
D0=D1->M1=ON; M11=ON
D0<D1->M2=ON; M12=ON

3、字软元件的情况

1）16 位指令的操作数修饰

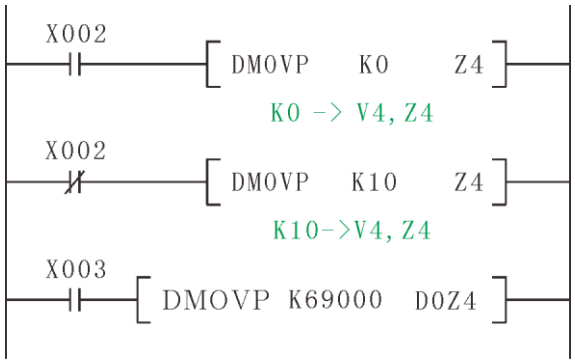
用变址寄存器 V3 中对 MOV 指令的传送目标 D0 进行修饰，以此为例说明如下图。将 K0 或 K10 事先传送到变址寄存器 V3 中。X001 为 ON, V3=0 时，则「D(0+0)=D0」 ，将 K500 传送至 D0 中。此外，当 V3=10 时， 则「D(0+10)=D10」，将 K500 传送至 D10 中。



V3=0:K500->D0(D0+0)
V3=10:K500->D10(D10+10)

2）32 位指令的操作数修饰

32 位指令的场合，指令中使用的变址寄存器也需要以 32 位进行指定。在 32 位指令中指定变址寄存器为 Z 侧(Z0~Z7)后，即包含了与 Z 侧成组的 V 侧(V0~V7)，一起作为 32 位寄存器运行。用变址寄存器[V4,Z4]修饰 DMOV 指令的传送目标[D1,D0]以此为例说明如下图。将 K0 或 K10 预先传送到变址寄存器[V4,Z4]中。X003 为 ON, [V4,Z4]=0 时，则[D(1+0)，D(0+0)]=[D1,D0]，将 K69000 传送至[D1,D0]中。此外，[V4,Z4]=10 时， 则[D(1+10), D(0+10)]=[D11,D10],将 K69000 传送至[D11,D10]中。



V4, Z4=0:K69000->D1, D0(D0+0)
V4, Z4=10:K69000->D11, D10(D10+10)

注意事项

1)即使写入变址寄存器的数值没有超出 16 位的数值范围(0～32767),也必须使用 32 位指令对 V, Z 都进行改写。如仅仅改写了 Z 侧，V 侧中会存有其他的数值，从而变成相当大的数值，导致出现运算出错。

2）对 16 位计数器变址修饰后，不可以作为 32 位的计数器使用。如果变址修饰的结果需要是 32 位计数器的情况下，请对计数器 C200 以后的计数器加 Z0~Z7。

3）变址寄存器不能对 V, Z 本身进行变址修饰。

4）特殊功能模块/单元的缓冲存储区的直接指定

缓冲存储区的直接指定 U[ ]\GL]其缓冲存储区的编号可以被变址修饰。

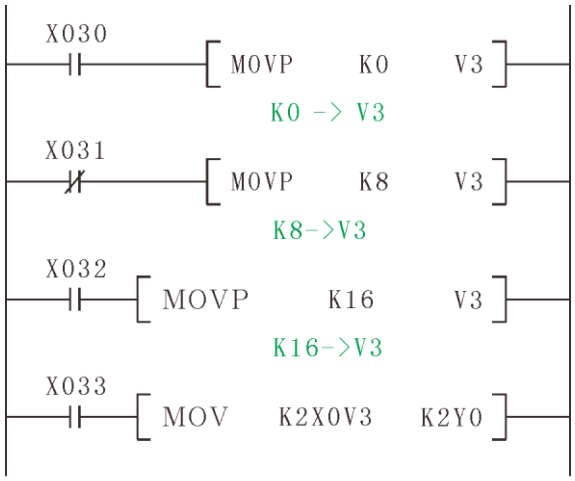
模块号不能被变址修饰。(U0\G0Z0 有效，U0Z0\G0 不可)位数指定的变址修饰

5）位数指定的变址修饰

指定位数用的 Kn 的"n"不能进行变址修饰。(K4M0Z0 有效，K0Z0M0 不可)

6）输入输出继电器(8 进制软元件编号)的变址修饰对 X,Y,KnX,KnY 的 8 进制软元件编号进行变址修饰时，对软元件编号进行变址修饰的变址寄存器内容会被换算成 8 进制数后再进行加法运算。

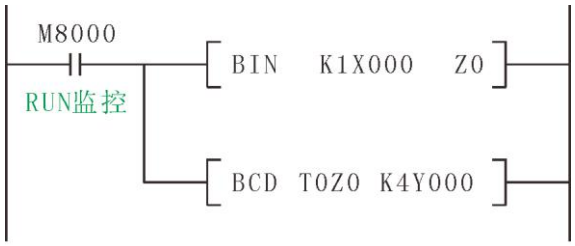
在下图的例子中，使用 MOV 指令输出 Y007 -Y000， 通过变址修饰后可以将该输入切换成 X000 ~X007, X017~X010, X027~X020。通过将变址值改写成 K0,K8,K16,则「X000+0=X000」,「X000+8=X10」「X000+16=X20」即在 8 进制数换算后，再加在软元件编号上，使作为源操作数的输入端子改变。



V3=0:X7~X0->Y7~Y0
V3=8:X17~X10->Y7~Y0
V3=16:X27~X20->Y7~Y0

定时器当前值的显示示例

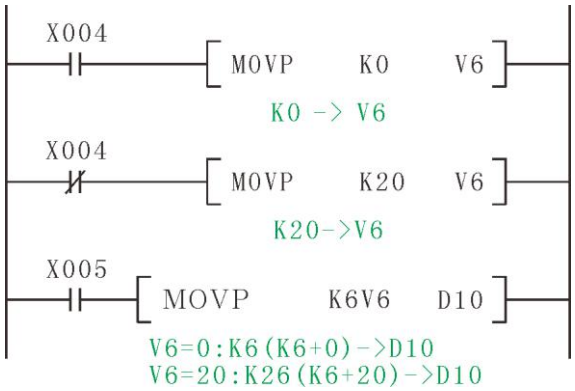
可以使用变址寄存器编写显示定时器 T0~T9 当前值用的程序。



对应Z0=0~9，则T0Z0=T0~T9

常数的情况

用变址寄存器 V6 修饰 MOV 指令的发送源，以此为例说明,如下图所示。将 K0 或 K20 事先传送到变址寄存器 V6 中。X005 为 ON，V6=0 时，则「K(6+0)=K6」，将 K6 传送至 D10 中。此外，V6=20 时，则「K(6+20)=K26」，将 K26 传送至 D10 中。



V6=0:K6(K6+0)->D10
V6=20:K26(K6+20)->D10

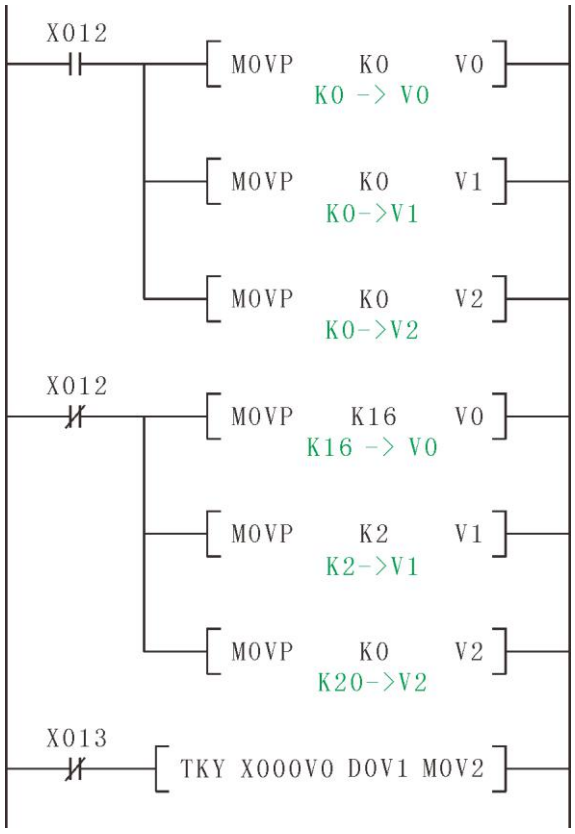
➢使用次数受限制的指令的变址修饰实例

如果用变址寄存器 V、Z 修饰对象软元件的编号，那么可通过程序改变对象软元件的编号。如果对有使用次数限制的指令使用这一功能的话，则可以得到与多次使用该指令编程相同的效果。

使用 TKY (FNC 70)指令的修饰实例将 2 组 0~9 为止的键(数字键)输入的输入数据保存到 D0,D1 中。

TKY 指令(FNC 70)是在程序中只能使用一次的指令，但是通过对输入数据的起始软元件编号、

输入数据的保存软元件编号、使按键信息为 ON 的起始软元件编号进行修饰，可输入 2 组 0~ 9 的键(数字键)输入。此外，在该指令执行过程中，即使改变 V，切换也无效。如果要想使该变更有效，请使指令的驱动 OFF 一次。



第四章 编程前须知

本章中，说明了在编写顺控程序中，输入输出处理和指令相互的关系，及编程的方法等。

4-1 指令说明的阅读方法

本手册中的应用指令，都按照如下的格式说明。

有关应用指令的表示方法和基本规则，请事先阅读后面的【6.5 应用指令的一般通则】

➢概要

1、指令格式

1)表示应用指令编号(FNC No.)和指令符合。简略表示的意思如下所示

与 16 位 32 位指令无关的单独指令。WDT(FNV 07)

【D】表示只可以使用 32 位指令。

【P】表示的脉冲执行型指令。

2、设定数据

表示可以指定为指令操作数的软元件的内容及可以设定的数据类型。

1)内容

描述了各指令中使用的操作数的内容。

2)源操作数/目标操作数的修饰

像 S・,S1・这样的添加了【・】符号的操作数，表示可以进行变址修饰。不执行变址修饰的操作数，会像 S,S1 这样表示。

3)数据类型

位:位软元件

BIN 16 位 ：16 位的二进制代码

BIN 32 位：32 位的二进制代码

BIN 64 位：64 位的二进制代码

BIN 16/32 位：16 位或 32 位的二 进制代码

BIN 32/64 位：32 位或 64 位的二进制代码

BCD4 位：4 位数(16 位)的 BCD 码

BCD8 位：8 位数(32 位)的 BCD 码

BCD4/8 位：4 位数(16 位)或 8 位数(32 位)的 BCD 码

字符串：ASCII 码，Shift JIS 代码等的字符代码

字符串(仅 ASCI)：ASCII 码

实数(2 进制)：2 进制的浮点数

实数(10 进制)：10 进制的浮点数

➢对象软元件

表示可以指定为指令操作数的软元件。

支持该指令的时候，用「●」符号表示。

1、位软元件

1）X :输入继电器(X)

2）Y:输出继电器(Y)

3）M:辅助继电器(M)

4）S :状态(S)

2、字软元件

1）K：10 进制整数
2）H：16 进制整数
3）KnX ：输入继电器(X)的位数指定※1
4）KnY ：输出继电器(Y)的位数指定※1
5）KnM：辅助继电器(M)的位数指定※1
6）KnS：状态(S)的位数指定※1
7）T：定时器(T)的当前值
8）C：计数器(C)的当前值
9）D：数据寄存器(文件寄存器)
10）V、 Z：变址寄存器
11）修饰：可否使用变址寄存器进行修饰

※1.未指定的 Kn, 16 位时为 K1 一 K4,32 位时为 K1～K8

➢**动作说明**

说明各指令的功能。

➢**注意要点**

记载了使用各指令时的注意事项。

➢**出错**

记载了使用各指令时可以考虑到的主要的出错。

➢**举例**

记载了使用了各指令的具体例子。

4-2 编程方面的基本注意事项

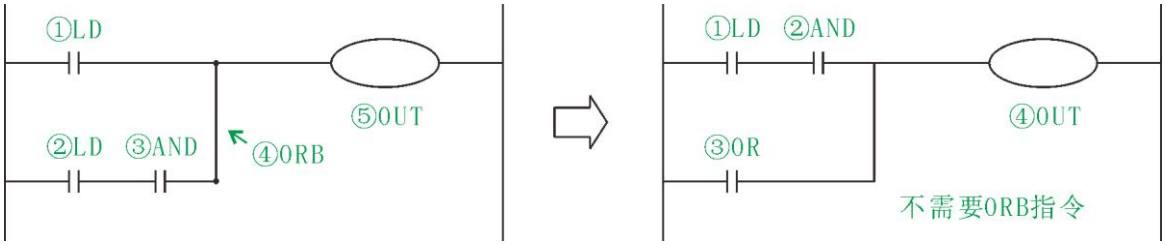
说明了在编程是需要注意的内容。

➢程序的步骤及执行顺序

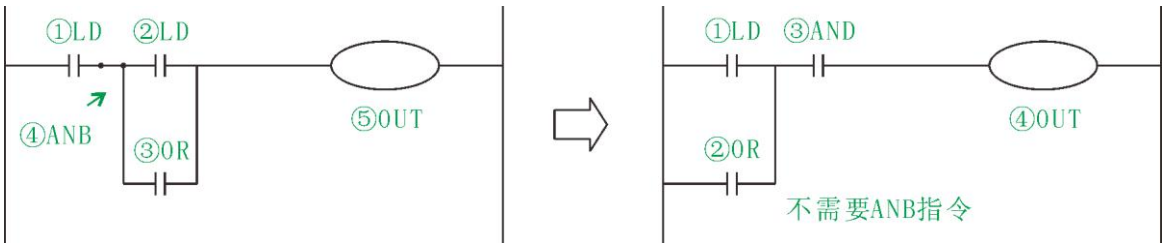
1、触点的构成和步

说明了在编程时需要注意的内容。

1)即使是执行相同的动作的顺控梯形图，触点的构成方法不同，也能简化程序和节约步数。



2)并联触点较多的梯形图写在左方比较好



2、程序的执行及编程顺序

顺控程序是按照【从上至下】到【从左至右】的顺序进行处理的。

顺控指令表也请按照这个顺序编码。

➢双重输出（双线圈）的动作及对策

1、双重输出的动作

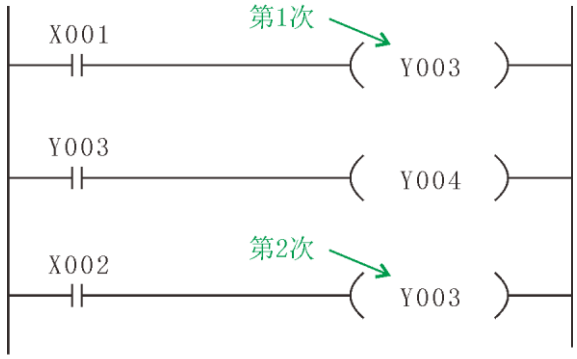
如果顺控程序中执行线圈的双重输出（双线圈），则后侧的线圈优先动作。

如右图所示，请考虑一下同一线圈 Y003 被在多个位置使用的情况。作为一种例子，X001=ON,X002=OFF

最初的 Y003 由于 X001 为 ON,所以其在映象存储区内为 ON,输出 Y004 也为 ON。

但是，第 2 次的 Y003 由于输入 X002 为 OFF，所以其在映象存储区内被改写为 OFF。

因此、实际的外部输出 Y003=OFF，Y004=ON。

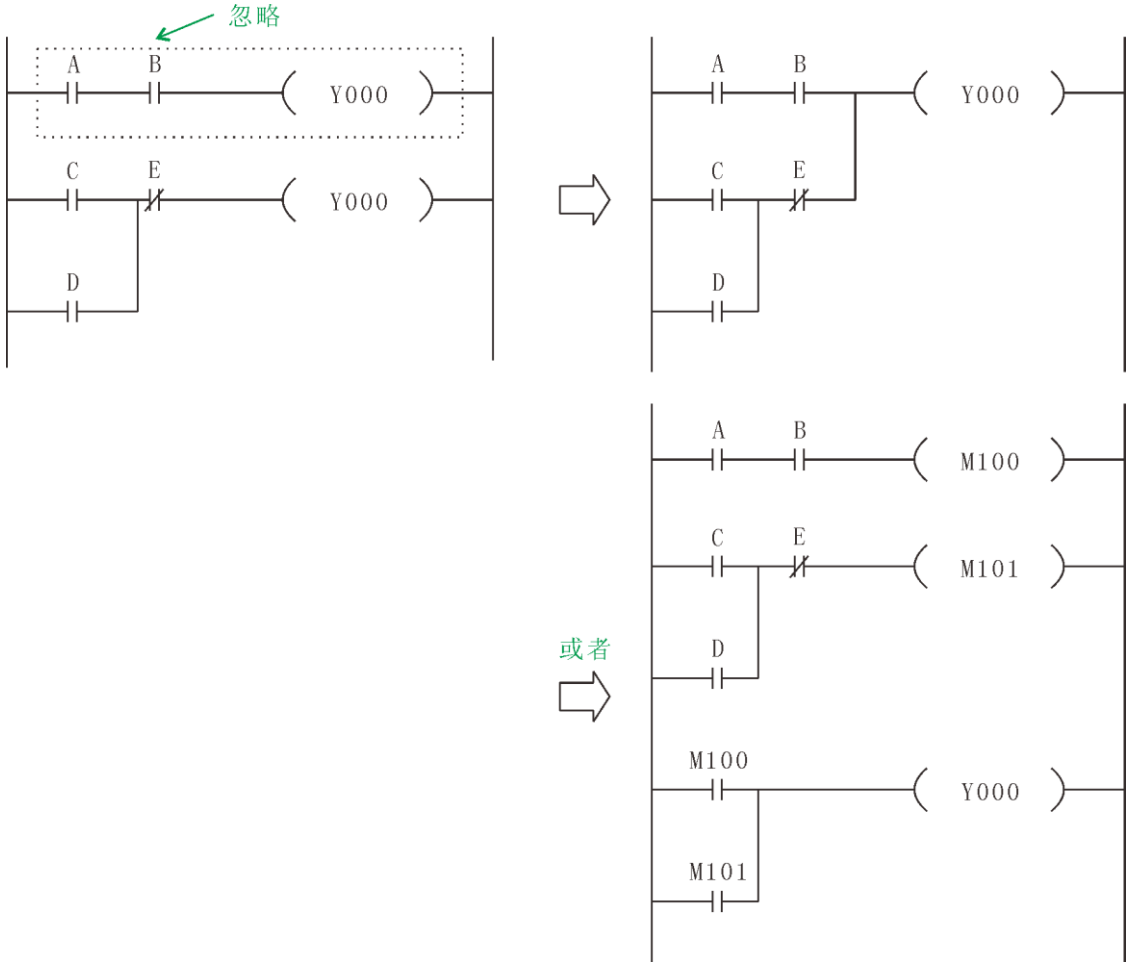


输入处理:X001=ON, X002=OFF

输出处理:Y003=ON, Y004=ON

2、双重输出的对策

双重输出(双线圈)，并非违背了程序中的输入(程序出错)，但是由于会使上述的动作变得复杂，因此请学习下面的例子后更改程序。



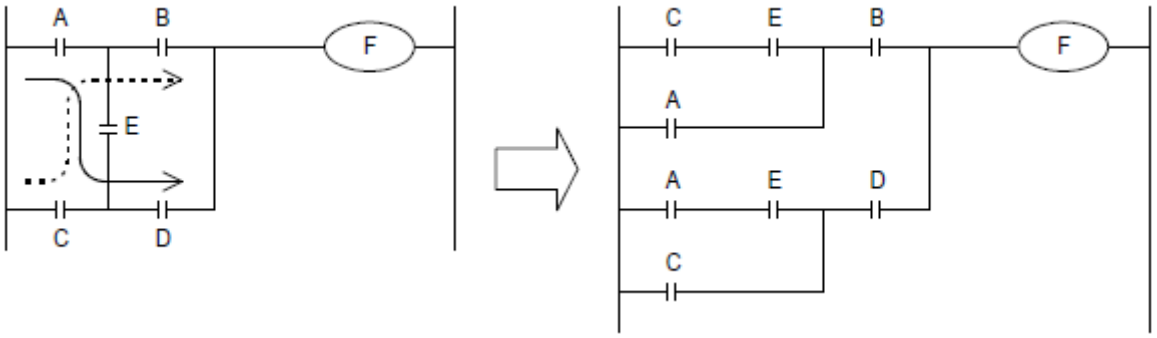
此外，还有其它编程方法，如使用 SET, RST 指令或跳转指令，以及使用步进梯形图指令， 在各状态中对同一个输出线圈编程。

此外，使用步进梯形图指令的时候请注意:如果主程序中存在的输出线圈，在状态中也被编程时，会被视为双重线圈。

➢不能编程的回路图及对策

1、桥式电路

请按照下图所示，更改两个方向都有电流流过的回路。(将没有 D 时的回路和没有 B 时的回路进行并联的结果。)

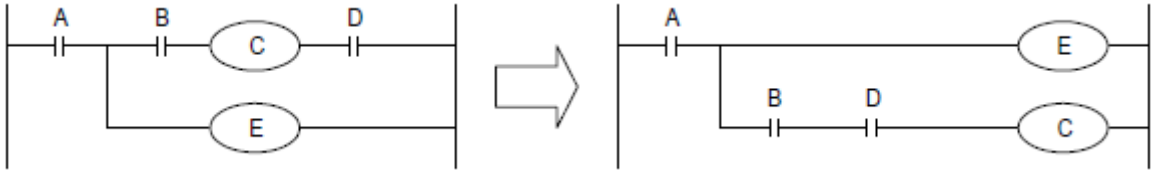


2、线圈连接的位置

1）线圈的右侧请勿写触点。

2）建议触点之间的线圈放在前面编程。

如触点 A 和 B 之间的线圈(E)放在程序前面，可以节省步数。



4-3 输入输出处理，响应延迟

1、输入输出继电器的动作时序和响应延迟

CZK2/CZK3 系列可编程控制器中，重复执行①~③进行输入输出处理。

因此，在可编程控制器的控制中，除了输入滤波器和输出元器件的驱动时间以外，根据扫描周期有时会出现响应延迟。

以最新信息获取最新输入输出，在上述的扫描周期中，想要获取输入的最新信息时或者要将运算结果立即输出时，可以使用【输入输出刷新指令】。

2、不能获取宽度窄的输入脉冲

可编程控制器的输入 ON 时间或 OFF 时间，必须比可编程控制器的循环扫描时间+输入滤波器的时间更长。考虑输入滤波器 10ms 的响应延迟，10ms 的循环扫描时间的话，ON 时间、OFF 时间各需要 20ms。

因此，不可以处理 1000/(20+20)=25Hz 以 上的输入脉冲。但是，使用可编程控制器的特殊功能和应用指令时，可以改善这个情况。

➢**改善用的便利功能**

使用下面的功能，可以获取比扫描周期更短的脉冲。

1）高速计数器功能；

2）输入中断功能；

3）脉冲捕捉功能；

4）输入滤波器值的调节功能。

4-5 应用指令的一般通则

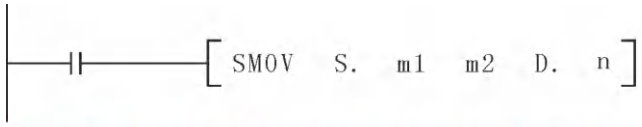
1 应用指令的表示和执行形式

1、指令和操作数

1）该可编程控制器的应用指令被分配了 FNC 00~- FNC□□□的功能编号，各指令中被授予了表示其内容的符号(助记符)。

例如，在 FNC 13 中被授予了称为 SMOV (位移动)的符号。

2）应用指令中仅有指令部分就能工作的，但更多的是在指令名后紧接操作数的组合构成。



S. :不会因通过执行指令，而使内容变化的操作数称为源操作数，以该符号表示。
可以使用变址寄存器修饰软元件编号的时候，以添加了【.】符号的S.表示，
有多个源操作数的时候以S1.,S2.表示。

D. :通过执行指令，其内容发生变化的操作数称为目标操作数，以该符号表示。
可以使用变址寄存器修饰目标操作数有多个的时候，以D1.,D2.等表示。
有多个源操作数的时候以S1.,S2.表示。

m,n :不符合源操作数也不符合目标操作数的操作数以m和n表示。
可以进行变址修饰的操作数有多个的时候，以m1.,m2.,n1.,n2.等表示。

3）应用指令的指令部分的程序步通常为 1 步，但是各操作数根据是 1 作位指令或 32 位指令，会变成 2 或 4 步。

2、操作数的对象软元件

1）有时会处理 X,Y,M,S 等的位软元件。

2）有时组合这些位软元件，显示为 KnX, KnY, KnM, KnS,将此可以作为数值数据处理。

3）有时候使用数据寄存器 D 或定时器 T、计数器 C 的当前值寄存器。

4）数据寄存器 D 为 16 位，处理 32 位数据的时候，会组合连续 2 点的数据寄存器。

例如，作为 32 位指令的操作数，指定了数据寄存器 D0 的时候，就处理(D1,D0)的 32 位数据。(D1 为 高 16 位，D0 为低 16 位)，将 T、C 的当前值寄存器作为一般的数据寄存器使用时，处理相同。

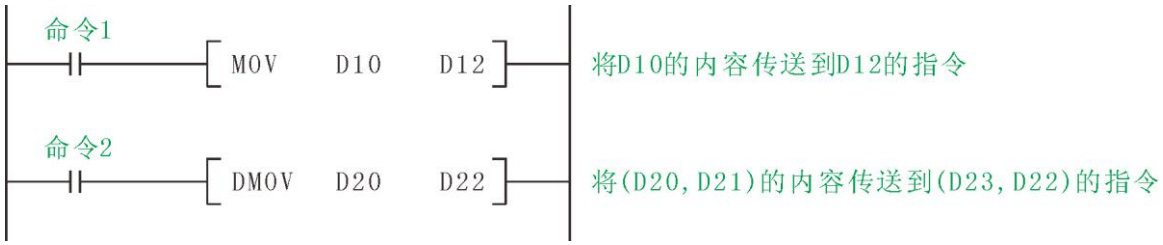
但是，C200~ C255 的 32 位计数器处理 1 点的 32 位的数据，不能指定为 16 位指令的操作数。

3、指令形式和执行形式

根据应用指令处理的数值的大小，可以分成「16 位指令」和「32 位指令」两种。而且，根据该指令的各执行形式不同，分为「连续执行型」和「脉冲执行型」两种类型。根据应用指令不同，分为具备这些所有的组合和不具备两种。

1、16 位/32 位指令

1）处理数值的应用指令中，根据数值数据的位长分为 16 位和 32 位情况。



2）32 位指令的时候，在 DMOV 中添加了[D]的符号来表示。

3）指定软元件可以使用偶数或是奇数，该号码与紧接其后的软元件组合使用。 (T,C,D 等的字软元件的情况)为了避免混乱，建议在 32 位指令的操作数中指定低位侧软元件使用偶数号码。

4）32 位计数器(C200 C255)该软元件 1 个即为 32 位，不可以作为 16 位指令的操作数使用。

2、脉冲执行/连续执行型

2.1）脉冲执行型

如下图所示，X000 从 OFF 变成 ON 的时候，只执行一次指令，除此以外的情况都不执行。因此，不需要一直执行的情况下，建议使用脉冲执行型指令。

P 的符号表示脉冲执行型的指令。DMOVP 也相同。

2.2）连续执行型

下图中为连续执行型的指令，X001 为 ON 的时候，每个扫描周期都会执行。

FNC 24(INC),FNC 25(DEC)等指令，如使用连续执行型指令时，每个扫描周期目标操作数的内容都会改变。

任何一种情况下，驱动输入 X000 和 X001 为 OFF 的时候不执行指令，除非特别记载的指令，目标操作数也不会改变。

2 一般标志位的使用

根据应用指令的种类不同，标志位的动作如下所示。

(例) M8020:零标志　　M8021:借位标志　　M8022:进位标志

M8029:指令执行结束标志　　M8090:块比较标志　　M8328:指令不执行标志

M8329:指令执行异常结束标志　M8304:零标志位　　M8306:进位标志位

各种指令每次 ON 执行时，这些标志位为 ON 或 OFF 动作，但是 OFF 执行的时候和出错的时候不改变。

由于标志位在多数的指令中会变化，所以每次执行这些指令的时候呈 ON/OFF 变化。

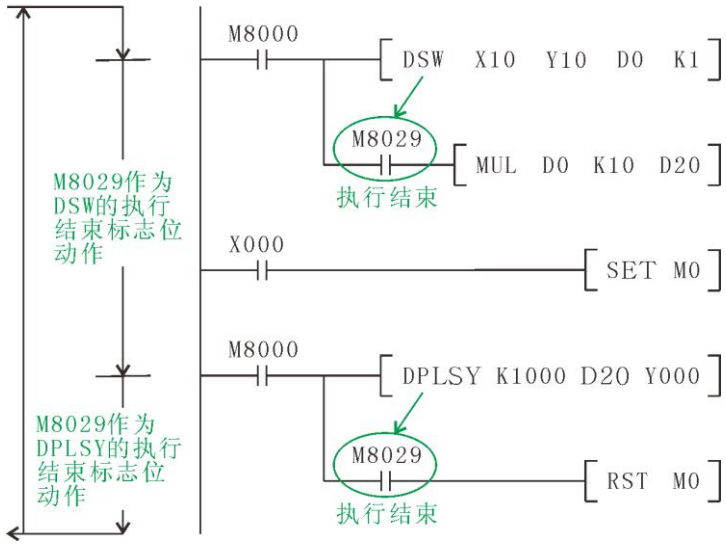
参考下面的例子，请在各指令的正下方编写标志位触点。

1、多个标志位的程序(指令执行结束标志位 M8029 的例子)

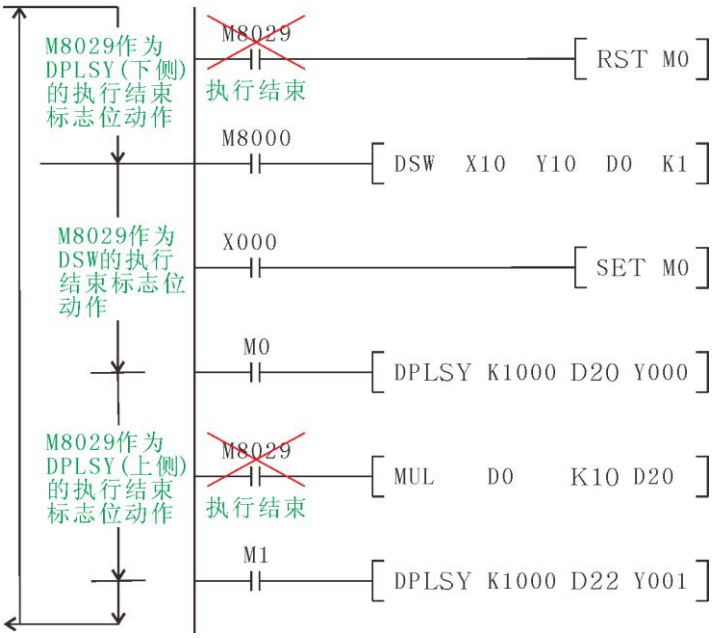
对使用相同标志位动作的应用指令而言，将指令执行结束标志位 M8029 集中在一起编程时，除了难于判断哪个指令的执行内容导致标志位控制执行，此外也可能不能正常读取各个指令相对应的标志位。

在应用指令的正下方以外的位置中使用时，请参考下一页的内容。

正确例子



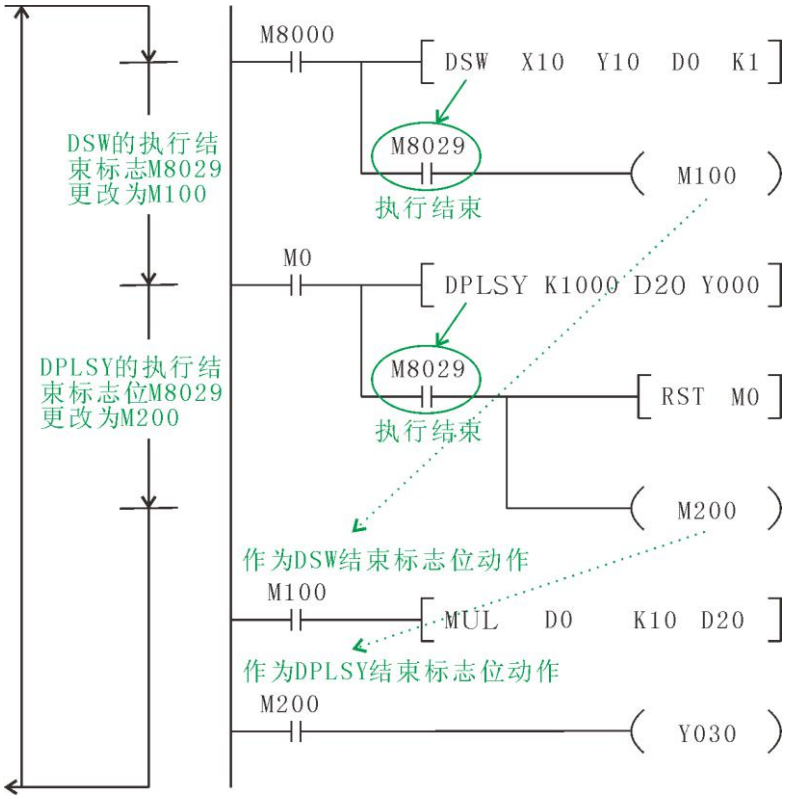
错误例子



2、在应用指令的正下方以外的位置中任用的方法介绍

对多个应用指令编程后，一般标志位会根据各自的应用指令的 ON 执行进行变化。

因此，想要在该指令的正下方以外的位置中使用时，先在应用指令的正下方用一般标志位，ON/OFF 其他的位软元件，然后将该触点作为指令触点使用。



3 运算出错标志位的使用

应用指令的结构和对象软元件及其编号范围等出现错误，在运算执行过程中出错的时候，下面的标志位动作，并且出错的信息被记忆下来。

1、运算出错

出错标志位	保存出错代码的软元件	保存出错步的软元件（32 位）	保存出错步的软元件※1
M8067	D8067	D8315,D8314	D8069

1）运算出错后，M8067 被置位，运算出错代码被保存到 D8067 中，出错步被保存到 D8315,D8314 (32 位)中。

2）发生出错的步在 32767 步以前的情况下，用 D8069 (16 位)也能够确认出错步。

3）其他的步中出现新的错误时，这个指令的出错代码和步编号依次被更新。(出错解除时为 OFF。)

4）可编程控制器从 STOP -RUN 时瞬间被清除，出错仍存在的情况下会再次置 ON。

2、运算出错锁存

出错标志位	保存出错代码的软元件	保存出错步的软元件（32 位）	保存出错步的软元件※1
M8068	-	D8313,D8312	D8068

1）运算出错后，M8068 被置位，出错的步编号会被保存到 D8313,D8312 中。

2）即使其他的指令中出现新的错误，内容也不被更新，到执行强制复位或断电为止动作都被保持。

3）发生出错的步在 32767 步以前的情况下，用 D8068 (16 位)也能够确认出错步。

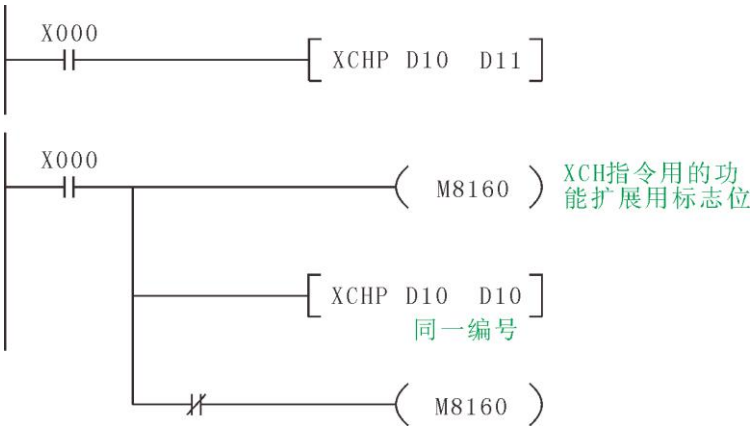
4 扩展功能用标志位的使用

一部分指令中，可以通过与该应用指令决定的特有的辅助继电器一起使用，
实现功能的扩展，以下举例说明。

1）X000 为 ON 时，互换 D10 和 D11 的内容指令。

2）在 XCH 指令前先驱动 M8160,将 XCH 指令的源操作数和目标操作数指定为相同的软元件，就可进行高 8 位和低 8 位的互换。

3）为了返回到普通的 XCH 指令，需要先断开 M8160。



此外，在中断程序中使用需要功能扩展标志位的指令时，驱动功能扩展标志位之前，请在 DI 指令(禁止中断)、功能扩展标志位的 OFF 后编写 EI 指令(允许中断)。

5 指令使用次数的限制

1、指令的使用次数和同时驱动的限制

在应用指令中，有只能编写指定次数，禁止重复使用的指令。

指令名称	使用次数	备注
FNC 52(MTR)	1	-
FNC 56(SPD)	8 (1 个指令/1 个输入 以下)	请注意不能和 DVIT 指令的中断输入、ZRN 指令的 DOG 输入、DSZR 指令的零点信号、输入中断和高速计数器的各输入编号重复。
FNC 60(IST)	1	-
FNC 69(SORT)	1	-
FNC 70(TKY)	1	-
FNC 71(HKY)	1	-
FNC 75(ARWS)	1	-
FNC 77(PR)	2	-
FNC 149(SORT2)	2	-
FNC 186(DUTY)	5 (1 个指令/1 个输入 以下)	-
FNC 280(HSCT)	1	-

要多次使用上述指令的时候

有关操作数中可以变址修饰的指令,使用变址寄存器可以更改指令内软元件的编号和数值。因此，不需要多个同时起动时，也可以得到与实际多次使用的控制相同的效果。

2、指令的同时驱动限制

在应用指令中，有这样的指令即使指令本身可以多次编程，但是同时起动点数有规定。

即使是下列指令以外的指令，如果同时起动多个针对同一输入输出编号的指令时，请注意会形成双重输出。而且，由于指令的组合会使动作变得复杂，有时候会出现不能执行指令的情况。详细内容请参考各指令的说明中的注意事项。

此外，有关各指令的组合请参考【程序流程控制指令的相互关系】。

1）定位指令

请勿对相同的输出编号同时使用 FNC57(PLSY), FNC58 (PWM), FNC59(PLSR), FNC150(DSZR), FNC151(DVIT)，FNC156(ZRN)，FNC157(PLSV)，FNC158(DRV)，FNC159(DRVA)指令。

2）高速处理指令

FNC53(HSCS)，FNC54(HSCR)，FNC55(HSZ)，包含 FNC280(HSCT)指令在内同时驱动的合计点数不能超出 32 点。[FNC280(HSCT)指令的使用次数仅为 1 次]

但是,「FNC280(HSCT)指令」 "FNC55(HSZ)指令的表高速比较模式”和「FNC55(HSZ)指令的频率控制模式」的使用次数为各 1 次。

3）外部设备通信指令

请不要对相同的通信口同时使用多个 FNC80(RS),FNC87(RS2)指令。

FNC80(RS)，FNC87(RS2)，FNC270(IVCK)，FNC271(IVRD)，FNC273(IVWR)，FNC274(IVBWR)指令针对相同的通信口时，不能组合使用。

可以对相同的通信口同时使用多个 FNC270(IVCK)，FNC271(IVRD)，FNC273(IVWR)，FNC274(IVBWR)指令。

第五章 基本指令

5-1 LD 系列连接到母线的指令

LD 取指令

➢功能说明

将触点连接到母线上，检出触点导通逻辑开始运算。

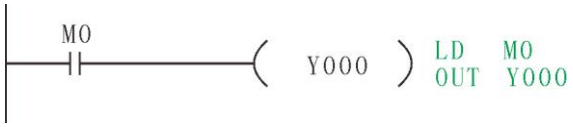
➢指令格式

LD S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

当（M0）=0 时，（Y0）=0；当（M0）=1 时，（Y0）=1；



LDI 取反指令

➢功能说明

将触点连接到母线上，检出触点不导通逻辑开始运算。

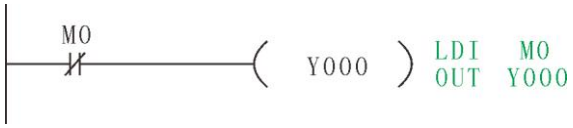
➢指令格式

LDI S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

当（M0）=0 时，（Y0）=1；当（M0）=1 时，（Y0）=0；



LDP 取脉冲上升沿指令

➢功能说明

将触点连接到母线上，上升沿检出触点导通逻辑开始运算。

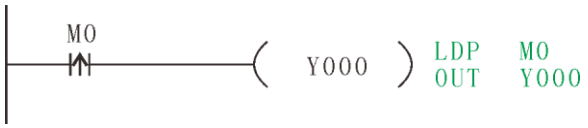
➢指令格式

LDP　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例:

梯形图运行第一个周期：（M0）=0，（Y0）=0；
梯形图运行第二个周期：（M0）=1，（Y0）=1；
梯形图运行第三个周期：（M0）=1，（Y0）=0；
梯形图运行第四个周期：（M0）=0，（Y0）=0；
梯形图运行第五个周期：（M0）=0，（Y0）=0；



LDF 取脉冲下降沿指令

➢功能说明

将触点连接到母线上，下降沿检出触点导通逻辑开始运算。

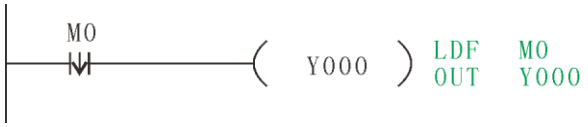
➢指令格式

LDF　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例:

梯形图运行第一个周期：（M0）=0，（Y0）=0；
梯形图运行第二个周期：（M0）=1，（Y0）=0；
梯形图运行第三个周期：（M0）=1，（Y0）=0；
梯形图运行第四个周期：（M0）=0，（Y0）=1；
梯形图运行第五个周期：（M0）=0，（Y0）=0；



5-3 AND 系列串联连接的指令

AND 与指令

➢功能说明

串联连接一个触点，检出触点导通逻辑开始运算。

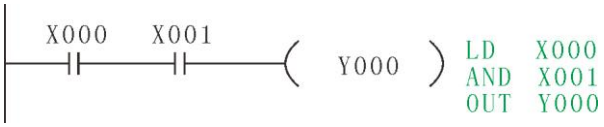
➢指令格式

AND　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

当（X0）=0，（X1）=0 时，（Y0）=0；
当（X0）=0，（X1）=1 时，（Y0）=0；
当（X0）=1，（X1）=0 时，（Y0）=0；
当（X0）=1，（X1）=1 时，（Y0）=1；



ANI 与反转指令

➢功能说明

串联连接一个触点，检出触点不导通逻辑开始运算。

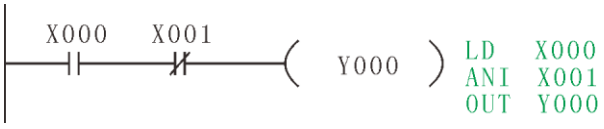
➢指令格式

ANDI　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

当（X0）=0，（X1）=0 时，（Y0）=0；
当（X0）=0，（X1）=1 时，（Y0）=0；
当（X0）=1，（X1）=0 时，（Y0）=1；
当（X0）=1，（X1）=1 时，（Y0）=0；



ANDP 与脉冲上升沿指令

➢功能说明

串联连接一个触点，上升沿检出触点导通逻辑开始运算。

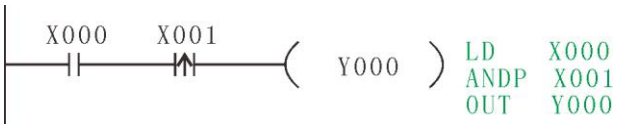
➢指令格式

ANDP　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

梯形图运行第一个周期：（X0）=1,（X1）=0，（Y0）=0;
梯形图运行第二个周期：（X0）=1,（X1）=1，（Y0）=1;
梯形图运行第三个周期：（X0）=1,（X1）=1，（Y0）=0;
梯形图运行第四个周期：（X0）=1,（X1）=0，（Y0）=0;
梯形图运行第五个周期：（X0）=1,（X1）=0，（Y0）=0;
梯形图运行第六个周期：（X0）=0,（X1）=0，（Y0）=0;
梯形图运行第七个周期：（X0）=0,（X1）=1，（Y0）=0;
梯形图运行第八个周期：（X0）=0,（X1）=0，（Y0）=0;



ANDF 与脉冲下降沿指令

➢功能说明

串联连接一个触点，下降沿检出触点导通逻辑开始运算。

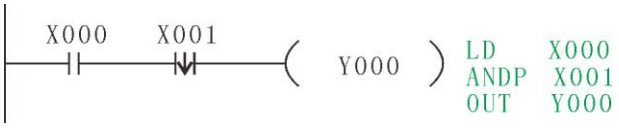
➢指令格式

ANDF　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢举例

梯形图运行第一个周期：（X0）=1,（X1）=0，（Y0）=0;
梯形图运行第二个周期：（X0）=1,（X1）=1，（Y0）=0;
梯形图运行第三个周期：（X0）=1,（X1）=1，（Y0）=0;
梯形图运行第四个周期：（X0）=1,（X1）=0，（Y0）=1;
梯形图运行第五个周期：（X0）=1,（X1）=0，（Y0）=0;
梯形图运行第六个周期：（X0）=0,（X1）=0，（Y0）=0;
梯形图运行第七个周期：（X0）=0,（X1）=1，（Y0）=0;
梯形图运行第八个周期：（X0）=0,（X1）=0，（Y0）=0;



5-4 OR 系列并联连接的指令

OR 或指令

➢功能说明

并联连接一个触点，检出触点导通逻辑开始运算。

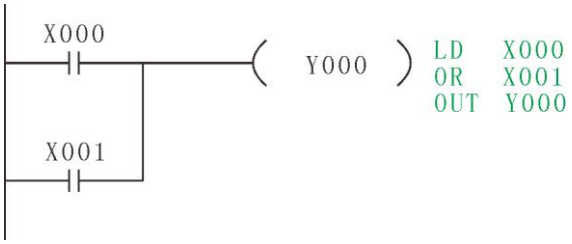
➢指令格式

OR　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

➢ 举例

当（X0）=0，（X1）=0 时，（Y0）=0；
当（X0）=0，（X1）=1 时，（Y0）=1；
当（X0）=1，（X1）=0 时，（Y0）=1；
当（X0）=1，（X1）=1 时，（Y0）=1；



ORI 或反转指令

➢功能说明

并联连接一个触点，检出触点不导通逻辑开始运算。

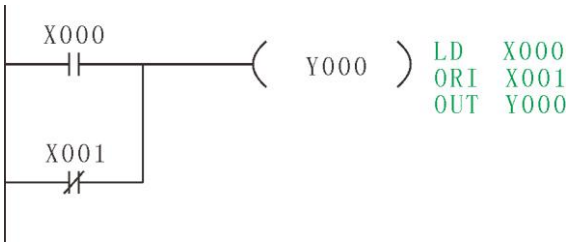
➢指令格式

ORI　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	X、Y、M、T、C、S	--	--	位

举例

当（X0）=0，（X1）=0 时，（Y0）=1；
当（X0）=0，（X1）=1 时，（Y0）=0；
当（X0）=1，（X1）=0 时，（Y0）=1；
当（X0）=1，（X1）=1 时，（Y0）=1；



ORP 或脉冲上升沿指令

➢功能说明

并联连接一个触点，上升沿检出触点导通逻辑开始运算。

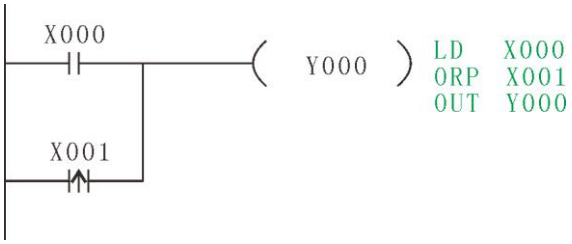
➢指令格式

ORP　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	X、Y、M、T、C、S	--	--	位

➢举例

梯形图运行第一个周期：（X0）=0,（X1）=0，（Y0）=0；
梯形图运行第二个周期：（X0）=0,（X1）=1，（Y0）=1；
梯形图运行第三个周期：（X0）=0,（X1）=1，（Y0）=0；
梯形图运行第四个周期：（X0）=0,（X1）=0，（Y0）=0；
梯形图运行第五个周期：（X0）=1,（X1）=0，（Y0）=1；



ORF 或脉冲下降沿指令

➢功能说明

并联连接一个触点，下降沿检出触点导通逻辑开始运算。

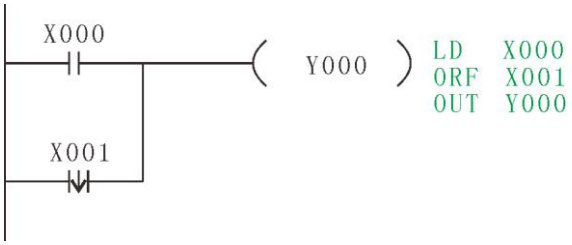
➢指令格式

ORF　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	X、Y、M、T、C、S	--	--	位

➢举例

梯形图运行第一个周期：（X0）=0,（X1）=0，（Y0）=0；
梯形图运行第二个周期：（X0）=0,（X1）=1，（Y0）=0；
梯形图运行第三个周期：（X0）=0,（X1）=1，（Y0）=0；
梯形图运行第四个周期：（X0）=0,（X1）=0，（Y0）=1；
梯形图运行第五个周期：（X0）=0,（X1）=0，（Y0）=0；
梯形图运行第六个周期：（X0）=1,（X1）=0，（Y0）=1；



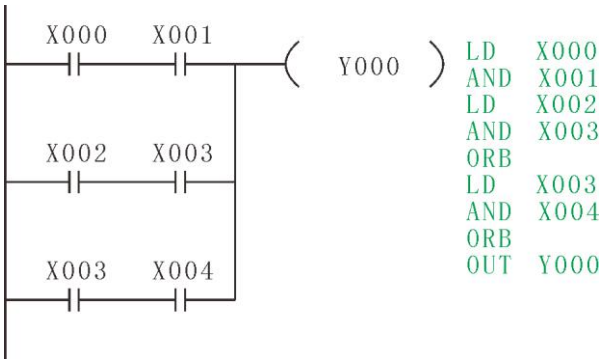
5-5 回路块指令

ORB 回路块或指令

➢功能说明

当串联电路（串联回路块）与前面的回路并联连接时，使用 ORB 回路块结束分支。

➢举例

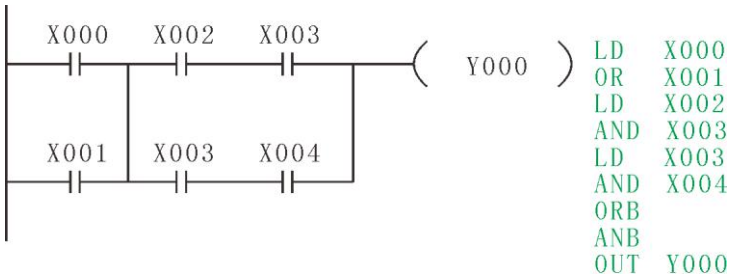


ANB 回路块与指令

➢功能说明

当并联电路（并联回路块）与前面的回路串联连接时，使用 ANB 回路块与指令与前面连接。

➢举例



5-6 线圈控制指令

OUT 输出指令

➢功能说明

线圈驱动输出。

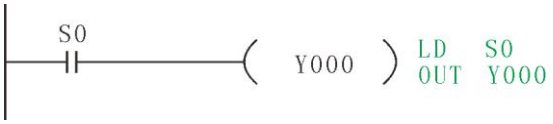
➢指令格式

OUT　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	Y、M、T、C、S	--	--	位

➢举例

当（S0）=0 时，（Y0）=0；
当（S0）=1 时，（Y0）=1；



SET 置位指令

➢功能说明

线圈置位输出。

➢指令格式

SET　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	Y、M、S	--	--	位

举例

梯形图运行第一个周期：（M0）=0，（Y0）=0；
梯形图运行第二个周期：（M0）=1，（Y0）=1；
梯形图运行第三个周期：（M0）=0，（Y0）=1；

RST 复位指令

➢功能说明

线圈复位不输出。

➢指令格式

SET　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	Y、M、S	--	--	位

➢举例

梯形图运行第一个周期：（M1）=0，（Y0）=1；
梯形图运行第二个周期：（M1）=1，（Y0）=0；
梯形图运行第三个周期：（M1）=0，（Y0）=0；

5-7 脉冲输出指令

PLS 上升沿脉冲输出指令

➢功能说明

仅在上升沿检出触点导通后的一个扫描周期，源操作数动作。

➢指令格式

PLS　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	Y、M	--	--	位

➢举例

梯形图运行第一个周期：（X1）=0，（Y0）=0；
梯形图运行第二个周期：（X1）=1，（Y0）=1；
梯形图运行第三个周期：（X1）=1，（Y0）=0；
梯形图运行第四个周期：（X1）=0，（Y0）=0；

PLF 下降沿脉冲输出指令

➢功能说明

仅在下降沿检出触点导通后的一个扫描周期，源操作数动作。

➢指令格式

PLF　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	源操作数	Y、M	--	--	位

➢举例

梯形图运行第一个周期：（X1）=1，（Y0）=0；
梯形图运行第二个周期：（X1）=0，（Y0）=1；
梯形图运行第三个周期：（X1）=0，（Y0）=0；
梯形图运行第四个周期：（X1）=1，（Y0）=0；

5-8 主控指令

MC 主控指令

➢功能说明

触点导通，母线移动到 MC 触电之后。则执行源操作数 1 对应指针 N 一致的相距最近的，MC 与 MCR 指令之间的指令。与 MCR 指令联合使用。

➢指令格式

MC　S1.　S2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	--	--	--	指针 N	BIN16 位
S2.	--	Y、M	--	--	BIN16 位

备注：

1）0<=S1.<= 7；

2）可嵌套使用，非嵌套使用指针编号重复使用，多次使用时需注意 S2.不能重复；

3）MC 指令指令至相应 MCR 指令之间的程序，由执行变为不执行时，部分指令对应的软元件会复位。具体包括非累积定时器，OUT 指令驱动的 Y、M、S、T；

➢举例

详见 MCR 主控复位指令举例。

MCR 主控复位指令

➢功能说明

返回源操作数一致的 MC 指令的母线位置。

➢指令格式

MCR S.

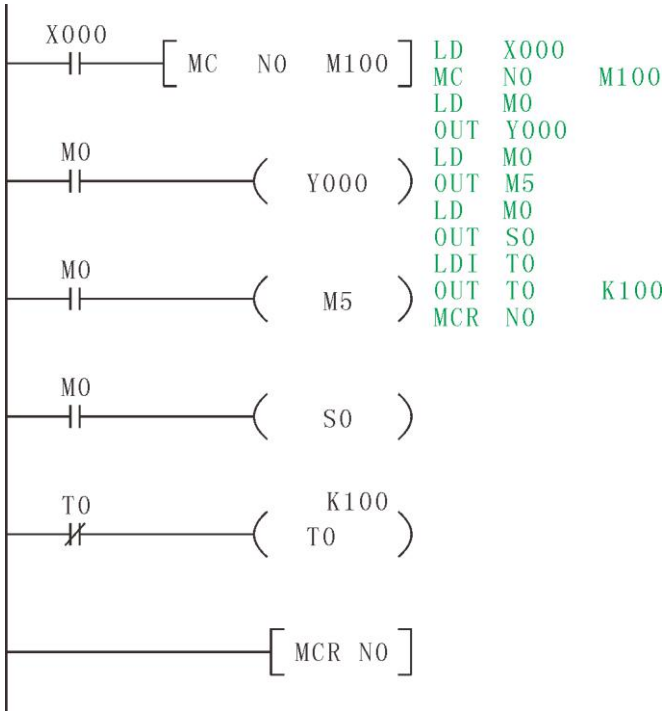
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.		--	--	指针 N	BIN16 位

备注：

1）0<=S.<= 7；

➢举例

梯形图运行第一个周期：（X0）=0，（M0）=0，（Y0）=0，（M5）=0，（S0）=0，（T0）=0；
梯形图运行第二个周期：（X0）=1，（M0）=0，（Y0）=0，（M5）=0，（S0）=0，（T0）开始计时；
梯形图运行第三个周期：（X0）=1，（M0）=1，（Y0）=1，（M5）=1，（S0）=1，（T0）开始计时；
梯形图运行第四个周期：（X0）=0，（M0）=1，（Y0）=0，（M5）=0，（S0）=0，（T0）=0；



5-9 堆栈指令

MPS 进栈指令

➢ 功能说明

并联回路块的串联连接，将指令运算中间结果压入堆栈。

➢ 举例

详见 MPP 出栈指令举例。

MRD 读栈指令

➢ 功能说明

并联回路块，将指令运算中间结果读出堆栈。

➢ 举例

详见 MPP 出栈指令举例。

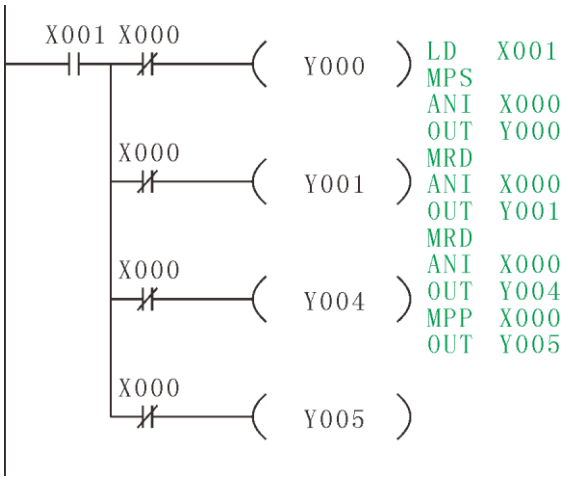
MPP 出栈指令

➢ 功能说明

并联回路块的串联连接，将指令运算中间结果读出堆栈并将其复位。

➢ 举例

梯形图运行第一个周期：（X1）=0，（Y0）=0，（Y1）=0，（Y4）=0，（Y5）=0；
梯形图运行第二个周期：（X1）=1，（Y0）=1，（Y1）=1，（Y4）=1，（Y5）=1；
梯形图运行第三个周期：（X1）=0，（Y0）=0，（Y1）=0，（Y4）=0，（Y5）=0；



5-10 其他基本指令

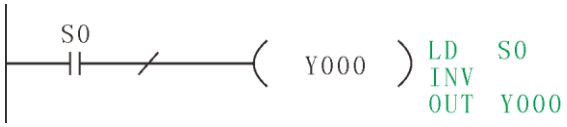
INV 取反指令

➢功能说明

运算结果反转。

➢举例

当（S0）=0 时，（Y0）=1；
当（S0）=1 时，（Y0）=0；



NOP 空操作指令

➢功能说明

没有任何操作。

END 结束指令

➢功能说明

输入及输出数据处理，并返回梯形图 0 步。

➢举例

返回梯形图 0 步；

第六章 梯形图步进指令

6-1 步进梯形图指令

STL 步进梯形图指令

➢功能说明

利用内软元件状态（S 软元件），在顺序控制上面进行工序步进形控制的指令，与 RET 指令联合使用。

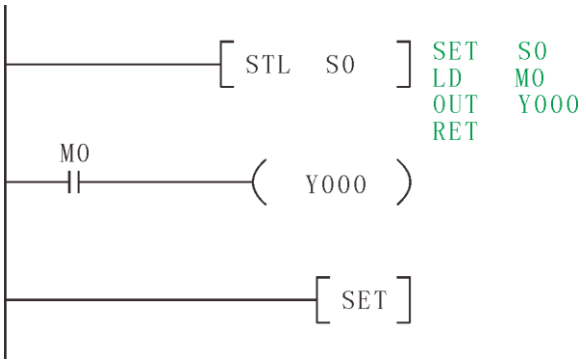
➢指令格式

STL S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	--	S	--	--	位

➢举例

S0 为 1 时允许执行 STL S0 至 RET 中间的这一段程序：
当（S0）=0，（M0）=0 时，（Y0）=0；
当（S0）=0，（M0）=1 时，（Y0）=0；
当（S0）=1，（M0）=0 时，（Y0）=0；
当（S0）=1，（M0）=1 时，（Y0）=1；



RET 返回指令

➢ **功能说明**

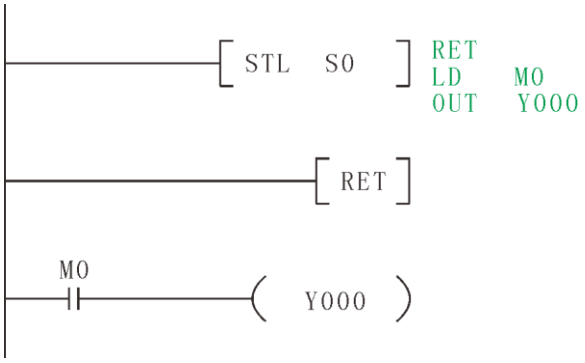
返回表示状态流程（STL 指令）的结束。

➢ **指令格式**

RET

➢ **举例**

RET 指令之下的程序不受 STL 指令控制：
当（S0）=0，（M0）=0 时，（Y0）=0；
当（S0）=0，（M0）=1 时，（Y0）=1；
当（S0）=1，（M0）=0 时，（Y0）=0；
当（S0）=1，（M0）=1 时，（Y0）=1；



第七章 应用指令

7-1 程序流程指令

FNC 00 - CJ 条件跳转指令

➢功能说明

使用 CJ、CJP 指令开始到指针（P）为止的顺控程序不执行的指令。可以缩短循环时间（扫描周期）和执行使用双线圈的程序。

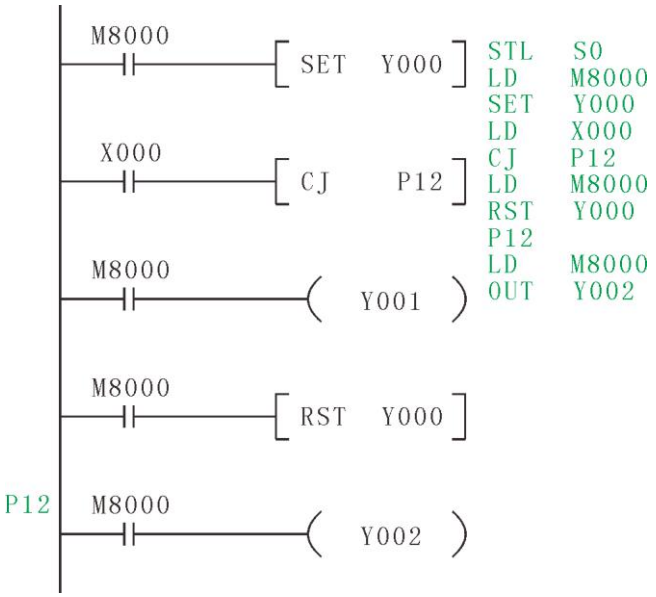
➢指令格式

CJ【P】 Pn.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	Pn.跳转目标标记编号的指针编号，0<=n<=127，指针 P63 固定指向梯形图结束 END	--	--	指针 P	BIN16 位

➢举例

梯形图运行第一个周期：（X0）=0，（Y0）=1，（Y1）=0，（Y2）=1；
梯形图运行第二个周期：（X0）=1，（Y0）=0，（Y1）=1，（Y2）=1；
梯形图运行第三个周期：（X0）=1，（Y0）=1，（Y1）=1，（Y2）=1；



FNC01 - CALL 子程序调用指令

➢功能说明

跳转至目标指针子程序处，开始执行。与 SRET 指令联合使用。

➢指令格式

CALL【P】 Pn.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	Pn.跳转目标标记编号的指针编号，0<=n<=127，指针 P63 固定指向梯形图结束 END	--	--	指针 P	BIN16 位

➢举例

见 SRET 子程序返回指令举例。

FNC02 - SRET 子程序返回指令

➢功能说明

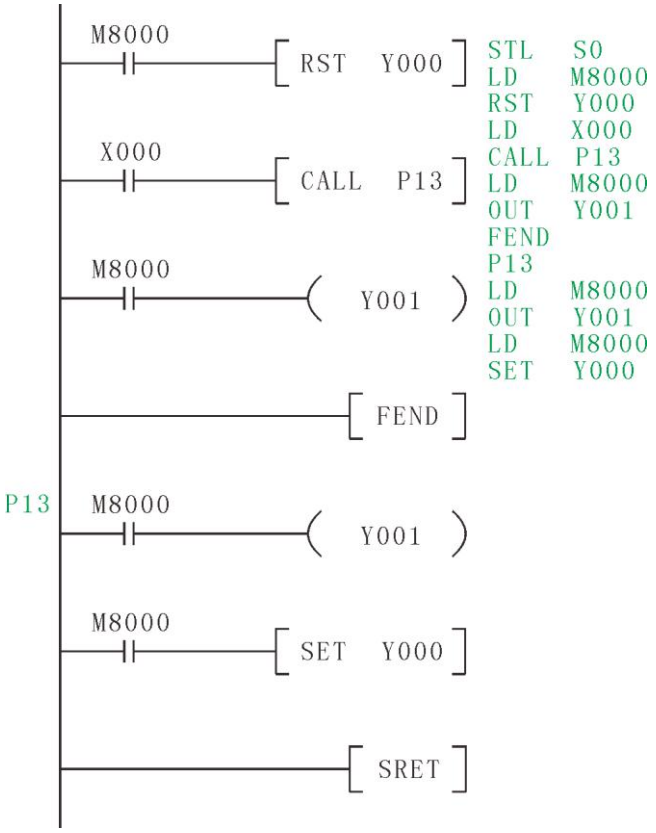
跳转出子程序，进入主程序，开始执行。与 CALL 指令联合使用。SRET 为单独指令。

➢指令格式

SRET

➢举例

梯形图运行第一个周期：（X0）=0，（Y0）=0，（Y1）=1，（Y2）=0；
梯形图运行第二个周期：（X0）=1，（Y0）=1，（Y1）=1，（Y2）=1；
梯形图运行第三个周期：（X0）=0，（Y0）=0，（Y1）=1，（Y2）=1；



FNC03 - IRET：中断返回

➢功能说明

从中断子程序返回到主程序的指令。不需要驱动触点的独立指令。

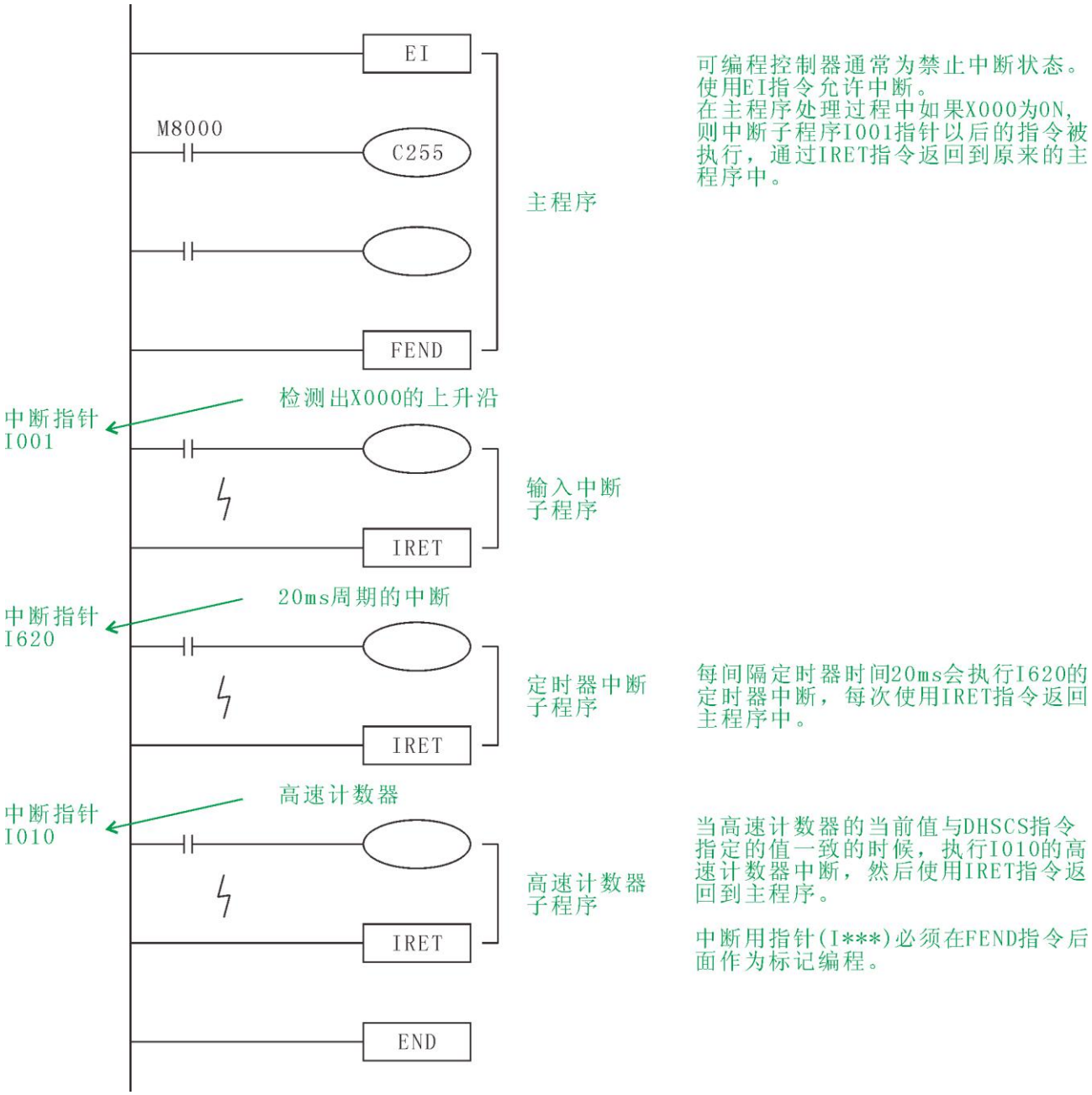
在处理主程序过程中如果产生中断（输入、定时器、计数器），则跳转到中断（I）程序，然后使用 IRET 指令返回主程序。跳转到中断程序的方法包括下表中的 3 种。

功能	中断编号	内容	参考
输入中断	I00*~I50*	通过输入（X）信号的 ON/OFF 执行中断处理	
定时器中断	I6**~I8**	每隔指定的时间间隔（固定周期）执行中断处理	
计数器中断	I010~I060	高速计数器增计数时执行中断处理	

➢指令格式

IRET

➢举例



FNC04 - EI：允许中断

➢功能说明

PLC 通常为禁止中断状态。使用这个指令，可以使 PLC 变为允许中断的状态。
使用输入中断和定时器中断，计数器中断功能的时候，请使用该指令。
EI 指令是不需要指令（驱动）触点的独立指令。

➢指令格式

EI

FNC05 - DI：禁止中断

➢功能说明

在改为允许中断后，使用 DI 指令可以再次更改为禁止中断。
DI 指令是不需要指令（驱动）触点的独立指令。
DI 指令以后产生的中断（要求），在执行了 EI 指令后方可处理。

➢指令格式

DI

FNC06 - FEND 主程序结束指令

➢功能说明

主程序结束的指令。

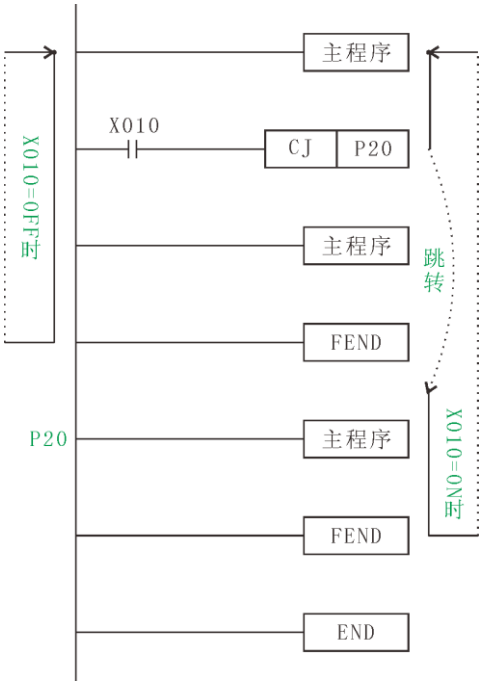
➢指令格式

FEND

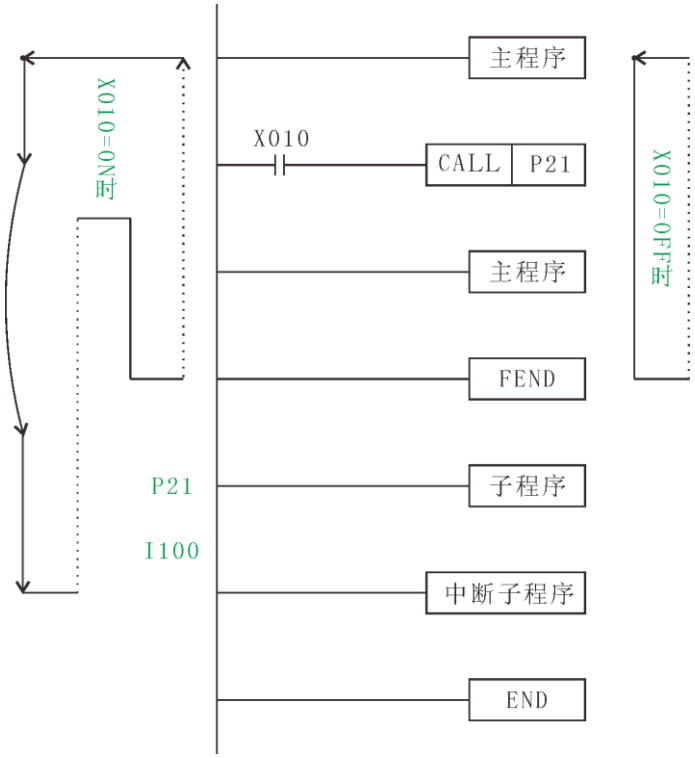
➢动作说明

执行 FEND 指令后，会执行与 END 指令相同的输出处理、输入处理、看门狗定时器的刷新，然后返回到 0 步的程序。在编写子程序和中断程序时需要使用这个指令。

1、CJ 指令的情况



2、CALL 指令的情况



➢**注意要点**

1）多次编写 FEND 指令时，请在最后的 FEND 指令和 END 指令之间编写子程序和中断程序。
2）CALL、CALLP 指令时，在 FEND 指令后编写标签，必须使用 SRET 指令。
3）CALL、CALLP 指令时后，到执行 SRET 指令之前，如执行了 FEND 指令，会出错。
4）FOR 指令时，执行 FOR 指令后，到执行 NEXT 指令之前，如执行了 FEND 指令，会出错。

FNC07- WDT：看门狗定时器

FNC08 - FOR 循环范围开始指令

➢功能说明

从 FOR 指令开始到 NEXT 指令之间的程序按指定次数（S.）重复执行。
与 NEXT 指令成对编程。

➢指令格式

FOR　S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	FOR~NEXT 指令之间的重复次数，1≤S.≤32767	--	KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位

备注：

1）S.为 FOR~NEXT 指令之间的重复次数，1<=S.<=32767，当 S.<1 时按循环 1 次处理）；

2）FOR 指令最多嵌套 5 层。

➢举例

见 NEXT 循环范围结束指令举例。

FNC09 - NEXT 循环范围结束指令

➢功能说明

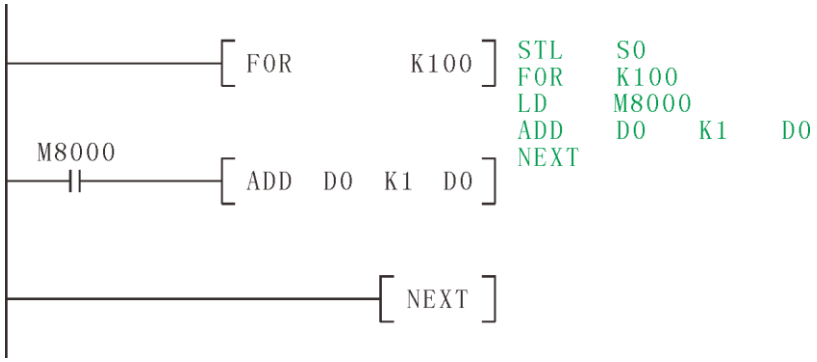
FOR 指令循环范围结束，判断决定跳转至 FOR 指令继续循环或结束循环继续执行程序。NEXT 为单独指令。

➢指令格式

NEXT

➢举例

操作前：（D0）=0；



操作后：（D0）=100；

7-2 传送与比较指令

FNC10 - CMP 比较指令

➢功能说明

将 S1.与 S2.的内容进行比较，并将比较结果送到 D.及其后两个软元件上。

➢指令格式

【D】CMP【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	成为比较值的数据或软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H、E	BIN16/32 位
S2.	成为比较源的数据或软元件编号	--	KnX、KnY、KnM、KnS、C、D、R、V、Z	K、H、E	BIN16/32 位
D.	输出比较结果的起始软元件编号	Y、M、S	--	K、H	位

备注：

1）所有源操作数都被看成二进制值处理；

2）D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用。

➢举例

CMP K100 C10 M2

操作后：当 K100>（C10）时，（M2）=1，（M3）=0，（M4）=0；

当 K100=（C10）时，（M2）=0，（M3）=1，（M4）=0；

当 K100<（C10）时，（M2）=0，（M3）=0，（M4）=1；

FNC11 - ZCP 区域比较指令

➢功能说明

将 S.与 S1.和 S2.的内容进行比较，并将比较结果送到目的操作数及其后两个软元件上。

➢指令格式

【D】ZCP【P】 S1. S2. S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	下侧的比较值的数据或软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H、E	BIN16/32 位
S2.	上侧的比较值的数据或软元件编号	--	KnX、KnY、KnM、KnS、C、D、R、V、Z	K、H、E	BIN16/32 位
S.	比较源的数据或软元件编号	--	KnX、KnY、KnM、KnS、C、D、R、V、Z	K、H、E	BIN16/32 位
D.	输出比较结果的起始软元件编号	Y、M、S	--	K、H	位

备注：

1）所有源操作数都被看成二进制值处理)；

2）D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用)；

3）源操作数 1 不得大于源操作数 2，当不满足该条件时，默认两个比较值都为源操作数 1)；

➢举例

ZCP K100 K120 C10 M2
操作后：当 K100>（C10）时，（M2）=1，（M3）=0，（M4）=0；
当 K100<=（C10）<=K120 时，（M2）=0，（M3）=1，（M4）=0；
当 K120<（C10）时，（M2）=0，（M3）=0，（M4）=1；

FNC12 - MOV 传送指令

➢功能说明

将 S.传送到 D.。

➢指令格式

【D】MOV【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	传送源的数据或保存数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	传送目标的软元件编号	--	KnY、KnM、KnS、C、D、R、V、Z	--	BIN16/32 位

备注：源操作数为常数时，将被自动转换成 BIN 码。

➢举例

操作前：（D0）=0x1234,（D11）=0;
MOV D0 D11
操作后：（D0）=0x1234,（D11）=0x1234;
操作前：（D1,D0）=0x12345678,（D12,D11）=0;
DMOV D0 D11
操作后：（D1,D0）=0x12345678,（D12,D11）=0x12345678;

FNC13 - SMOV 位移动指令

➢功能说明

将 S.的 BCD 码转换值从其第 m1 位起的低 m2 位部分向 D.（BCD 码转换值）开始传送，然后将其转换回 BIN 码。当驱动 M8168 执行时不进行 BCD 码转换，直接传送。

➢指令格式

SMOV【P】 S. m1 m2 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存有要进行位移动的数据软元件的编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16 位
m1	要移动的起始位的位置	--	--	K、H	BIN16 位
m2	要移动的位的个数	--		K、H	BIN16 位
D.	保存已经进行移动的数据的软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16 位
n	指定移动目标的起始位的位置	--	--	K、H	BIN16 位

备注：

1）m1、m2、n=1~4，并且 n>=m2，S.<9999，D.<9999；

2）当驱动 M8168 时，S.与 D.中的数据不转换为 BCD 码，按十六进制的格式移动；

➢ **举例**

操作前：（D10）=1234，（D2）=5678，（M8168）=0；
SMOV D10 K4 K2 D2 K3
操作后：（D10）=1234，（D2）=5128，（M8168）=0；

操作前:（D10）=0x6789，（D2）=0x1234，（M8168）=1；
SMOV D10 K4 K2 D2 K2
操作后：（D10）=0x6789，（D2）=0x1267，（M8168）=1；

FNC14 - CML 反相传送指令

➢**功能说明**

将 S.的各位反相（0 变 1， 1 变 0）后，传送到 D.。

➢**指令格式**

【D】CML【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	要执行反转的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	保存要执行反转后的数据的目标字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：源操作数为常数时，将被自动转换成 BIN 码；

➢**举例**

操作前：（D0）=0x6969，（D10）=0；
CML D0 D10
操作后：（D0）=0x6969，（D10）=0x9696；

操作前：（D1，D0）=0xA5A59696，（D11，D10）=0；
DCML D0 D10
操作后：（D1，D0）=0xA5A59696，（D11，D10）=0x5A5A6969；

FNC15 - BMOV 成批传送指令

➢功能说明

将以 S.指定的软元件为开头的 n 个软元件，向以 D.指定的软元件开头的 n 个软元件成批传送。扩展指令，当驱动 M8024 时执行，反向传输。

➢指令格式

BMOV【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	传送源的数据，或是保存数据的软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
D.	传送目标的软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
n	传送点数（包括文件寄存器），n≤512	--	D	K、H	BIN16 位

备注：

1）n <= 512，在超过软元件编号范围时，在可能的范围内传送；

2）当 n<0 时会置位运算错误标志位 M8067；

3）当驱动 M8024 时，执行指令传送方向反转；

4）带有位指定的软元件（如：K1M0）时，源操作数和目标操作数要采用相同的位数；

➢举例

操作前：（D7）=0x1234，（D8）=0x5678，（D20）=0，（D21）=0,（M8024）=0；

BMOV D7 D20 K2

操作后：（D7）=0x1234，（D8）=0x5678，（D20）=0x1234，（D21）=0x5678,（M8024）=0；

操作前: (D5）=0，(D6）=0，（D10）=0x6789，（D11）=0xABCD,（M8024）=1；

BMOV D5 D10 K2

操作后：(D5）=0x6789，(D6）=0xABCD，（D10）=0x6789，（D11）=0xABCD,（M8024）=1；

FNC16 - FMOV 多点传送指令

➢功能说明

将 S.的内容向以 D.为开头的 n 个软元件进行重复传送。操作后从 D.开始的 n 个软元件的内容都一样。

➢指令格式

【D】FMOV【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	传送源的数据，或是保存数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	传送目标的起始字软元件编号（传送源的同一数据被成批传送）。	--	KnY、KnM、KnS、T、C、D、R	--	BIN16/32 位
n	传送点数，1≤n≤512	--	D	K、H	BIN16 位

备注：

1）当 n<0 时会置位运算错误标志位 M8067；

2）超过目标软元件号的范围，向可能的范围传送。

➢举例

操作前：（D20）=0，（D21）=0，（D22）=0；
FMOV　K12　D20　K3
操作后：（D20）=12，（D21）=12，（D22）=12；
操作前：（D11，D10）=0，（D13，D12）=0，（D15，D14）=0；
DFMOV　K12　D10　K3
操作后：（D11，D10）=12，（D13，D12）=12，（D15，D14）=12；

FNC17 - XCH 交换指令

➢功能说明

将 S.与 D.数据交换。扩展指令，当驱动 M8160 且 S.与 D.是统一软元件时，该操作数低八位与高八位交换。

➢指令格式

【D】XCH【P】　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D1.	保存交换数据的软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
D2.	保存交换数据的软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：

1）对于 C200 及其他高级计数器不能用 XCH 操作，视为出错；

2）当驱动 M8160 时，并且 S.与 D.是同一软元件时，低 8 位与高 8 位可进行交换。非同一软元件时会置位运算错误标志位 M8067（扩展功能与 SWAP 指令功能相同）；

3）当驱动 M8160 时，S.与 D.的软元件编号不同时，出错标志 M8067 变为 ON 状态，该指令无法执行。

➢举例

操作前：（D10）=20，（D12）=30，（M8160）=0；
XCH　D10　D12
操作后：（D10）=30，（D12）=20，（M8160）=0；
操作前：（D11，D10）=0x12345678，（D13，D12）=0xABCDEF21，（M8160）=0；
XCH　D10　D12
操作后：（D11，D10）=0xABCDEF21，（D13，D12）=0x12345678，（M8160）=0；
操作前：（D10）=0x695A，（M8160）=1；
XCH　D10　D10
操作后：（D10）=0x5A69，（M8160）=1；
操作前：（D11,D10）=0x1234695A，（M8160）=1；
DXCH　D10　D10
操作后：（D11,D10）=0x34125A69，（M8160）=1；

FNC18 - BCD：BCD 转换

➢功能说明

将 S.中的 2 进制数转换成 BCD 码（10 进制数）送到 D.中。

➢指令格式

【D】BCD【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存转换源（2 进制）数据的子软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
D.	转换目标（10 进制）的软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：当 BIN 指令源操作数超出 0~9999 范围，或当 DBIN 指令源操作数超出 0~99999999 范围，会置位运算错误标志位 M8067。

➢举例

操作前：（D10）=18，（D12）=0；
BCD D10 D12
操作后：（D10）=18，（D12）=0x18；

操作前：（D11，D10）=43981，（D13，D12）=0；
DBCD D10 D12
操作后：（D11，D10）=43981，（D13，D12）=0x43981；

FNC19 - BIN：BIN 转换

➢功能说明

将 S.中的 BCD 码装换成二进制数送到 D.中。

➢指令格式

【D】BIN【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存转换源（10 进制）数据的子软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
D.	转换目标（2 进制）的软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：

1）当 BIN 指令目标操作数超出 0~9999 范围，或当 DBIN 指令目标操作数超出 0~99999999 范围，会置位运算错误标志位 M8067 与预算错误锁存标志位 M8068；

2）当源操作数的数据不是 BCD 码时，会发生 M8067（运算错误），M8068（运算错误锁存）将不工作。

➢举例

操作前：（D0）=0x1234，（D12）=0；
BIN D10 D12
操作后：（D0）=0x1234，（D12）=1234；

操作前：（D1，D0）=0x12345678，（D13，D12）=0；
DBIN D10 D12
操作后：（D1，D0）=0x12345678，（D13，D12）=12345678；

7-3 四则逻辑运算指令

FNC20 - ADD：BIN 加法运算

➢功能说明

将 S1.与 S2.数据进行二进制加法后传递到 D.中。

➢指令格式

【D】ADD【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	加法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R	K、H	BIN16/32 位
S2.	加法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	保存加法运算结果的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：

1）运算结果为 0 时，0 标志 M8020 会动作；

2）运算结果超过 32767（16 位运算）或 2147483647（32 位运算）时，进位标志 M8022 会动作；

3）运算结果不满-32768（16 位运算）或-2147483648（32 位运算）时，借位标志 M8021 会动作。

➢举例

操作前：（D0）=2，（D10）=0；

ADD K2 D0 D10

操作后：（D0）=2，（D10）=4；

操作前：（D1，D0）=99999，（D11，D10）=0；

DADD K99999 D0 D10

操作后：（D1，D0）=99999，（D11，D10）=199998；

FNC21 - SUB：BIN 减法运算

➢功能说明

将 S1.的数据减去 S2.的数据后传递到 D.中。

➢指令格式

【D】SUB【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	减法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R	K、H	BIN16/32 位
S2.	减法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	保存减法运算结果的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

备注：

1）运算结果为 0 时，0 标志 M8020 会动作；

2）运算结果超过 32767（16 位运算）或 2147483647（32 位运算）时，进位标志 M8022 会动作；

3）运算结果不满-32768（16 位运算）或-2147483648（32 位运算）时，借位标志 M8021 会动作。

➢举例

操作前：（D0）=555，（D10）=55，（D20）=0；

SUB D0 D10 D20

操作后：（D0）=555，（D10）=55，（D20）=500；

操作前：（D1，D0）=777777，（D11，D10）=34567，（D21，D20）=0；

DSUB D0 D10 D20

操作后：（D1，D0）=777777，（D11，D10）=34567，（D21，D20）=743210；

FNC22 - MUL：BIN 乘法运算

➢功能说明

将 S1.和 S2.的数据相乘后传递到 D.中。

➢指令格式

【D】MUL【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	乘法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、Z	K、H	BIN16/32 位
S2.	乘法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、Z	K、H	BIN16/32 位
D.	保存乘法运算结果的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、Z	--	BIN16/32 位

备注：

1）运算结果为 0 时，0 标志 M8020 会动作；

2）D.是位元件时，可以进行 K1~K8 的位指定，指定 K4 时，智能求得乘积运算的低 16 位；

3）32 位指令且目标操作数使用位软元件时，只能得到低 32 位地址；

4）不能指定 Z 作为 D.。

➢举例

操作前：（D0）=11111，（D10）=11112，（D21，D20）=0；

MUL D0 D10 D20

操作后：（D0）=11111，（D10）=11112，（D21，D20）=123465432；

操作前：（D1，D0）=777777，（D11，D10）=222222，（D23，D22，D21，D20）=0；

DMUL D0 D10 D20

操作后：（D1，D0）=777777，（D11，D10）=222222，（D23，D22，D21，D20）=172839160494；

FNC23 - DIV：BIN 除法运算

➢功能说明

将 S1.中内容整除 S2.中的内容，并将商和余数分别放入 D.和 D.的下一个编号的软元件中。

DIV－16 位指令，源操作数与目标操作数为 16 位；

DDIV－32 位指令，源操作数与目标操作数为 32 位。

➢指令格式

【D】DIV【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	除法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、Z	K、H	BIN16/32 位
S2.	除法运算的数据，或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、Z	K、H	BIN16/32 位
D.	保存除法运算结果的字软元件编号（商、余数）	--	KnY、KnM、KnS、T、C、D、R、Z	--	BIN16/32 位

备注：

1）除数为 0 时置位运算错误标志位 M8067，不能执行指令；

2）将位软元件指定为 D.时，无法得到余数）；

3）不能指定 Z 作为 D.。

➢举例

操作前：（D0）=25，（D10）=10，（D20）=0，（D21）=0；

DIV D0 D10 D20

操作后：（D0）=25，（D10）=10，（D20）=2，（D21）=5；

操作前：（D1，D0）=777777，（D11，D10）=222222，（D21，D20）=0，（D23，D22）=0；

DDIV D0 D10 D20

操作后：（D1，D0）=777777，（D11，D10）=222222，（D21，D20）=3，（D23，D22）=111111；

FNC24 - INC：BIN 加一

➢功能说明

D.的内容加一。

➢指令格式

【D】INC【P】 D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存被加一数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、Z	--	BIN16 位

➢举例

操作前：（D0）=300；

INC D0

操作后：（D0）=301；

操作前：（D1，D0）=800000；

DINC D0

操作后：（D1，D0）=800001；

FNC25 - DEC：BIN 减一

➢功能说明

D.的内容自减一。

➢指令格式

【D】DEC【P】 D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存被减一数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、Z	--	BIN16 位

➢举例

操作前：（D0）=300；
INC D0
操作后：（D0）=299；

操作前：（D1，D0）=800000；
DINC D0
操作后：（D1，D0）=799999；

FNC26 - WAND：逻辑与

➢功能说明

将 S1.的数据与 S2.的数据按位与，并将结果传递到 D.中。

➢指令格式

【D】WAND【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	逻辑与数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	逻辑与数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	逻辑与结果的字元件编号	--	KnY、KnM、KnS、C、D、R、V、Z	--	BIN16/32 位

➢举例

操作前：（D0）=0xF，（D10）=0x5，（D20）=0；
WAND D0 D10 D20
操作后：（D0）=0xF，（D10）=0x5，（D20）=0x5；
操作前：（D1，D0）=0xFFFFF000，（D11，D10）=0xFFFF，（D21，D20）=0；
DAND D0 D10 D20
操作后：（D1，D0）=0xFFFFF000，（D11，D10）=0xFFFF，（D21，D20）=0xF000；

FNC27 - WOR：逻辑或

➢功能说明

将 S1.的数据与 S2.的数据按位或，并将结果传递到 D.中。

➢指令格式

【D】WOR【P】　S1.　S2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	逻辑或数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	逻辑或数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	逻辑或结果的字元件编号	--	KnY、KnM、KnS、C、D、R、V、Z	--	BIN16/32 位

➢举例

操作前：（D0）=0xF，（D10）=0x5，（D20）=0；
WOR　D0　D10　D20
操作后：（D0）=0xF，（D10）=0x5，（D20）=0xF；
操作前：（D1，D0）=0x1234FF00，（D11，D10）=0xFFFF，（D21，D20）=0；
DOR　D0　D10　D20
操作后：（D1，D0）=0x1234FF00，（D11，D10）=0xFFFF，（D21，D20）=0x1234FFFF；

FNC28 - WXOR：逻辑异或

➢功能说明

将 S1.的数据与 S2.的数据按位异或,并将结果传递到 D.中。

➢指令格式

【D】WXOR【P】　S1.　S2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	逻辑异或数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	逻辑异或数据或保存数据的字元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
D.	逻辑异或结果的字元件编号	--	KnY、KnM、KnS、C、D、R、V、Z	--	BIN16/32 位

➢举例

操作前：（D0）=15，（D10）=5，（D20）=0；
WXOR　D0　D10　D20
操作后：（D0）=15，（D10）=5，（D20）=10；
操作前：（D1，D0）=0x1234FF00，（D11，D10）=0xFFFF，（D21，D20）=0；
DXOR　D0　D10　D20
操作后：（D1，D0）=0x1234FF00，（D11，D10）=0xFFFF，（D21，D20）=0x123400FF；

FNC29 - NEG：补码

➢功能说明

D.的内容中各位先取反然后在加一，将其结果在存入原先的软元件中。

➢指令格式

【D】NEG【P】 D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存欲求补码的数据的字软元件编号，以及保存目标软元件编号（运算结果被保存在同一字软元件编号中）	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位

➢举例

操作前：（D0）=111；
NEG D0
操作后：（D0）=-111；

操作前：（D1，D0）=800000；
DNEG D0
操作后：（D1，D0）=-800000；

7-4 循环移位 - FNC30~FNC39

FNC30 - ROR：循环右移

➢功能说明

D.的内容右移 n 位，超出寄存器范围的 n 位补入低位部分。

➢指令格式

【D】ROR【P】 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存循环右移数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
n	旋转移动的位数，16 位指令时 n≤16；32 位指令时 n≤32	--	D、R	K、H	BIN16/32 位

备注：

1）16 位指令时，n<=16，32 位指令时，n<=32；

2）补入的最后一位放入进位标志位 M8022；

3）在位指定软元件情况下，只有 K4（16 位指令）和 K8（32 位指令）是有效的

➢举例

操作前：（D0）=0xFF08；

ROR D0 K4

操作后：（D0）=0x8FF0；

操作前：（D1，D0）=0x12345678；

DROR D0 K4

操作后：（D1，D0）=0x81234567；

FNC31 - ROL：循环左移

➢功能说明

D.的内容左移 n 位，超出寄存器范围的 n 位补入高位部分。

➢指令格式

【D】ROL【P】 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存循环左移数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
n	旋转移动的位数，16 位指令时 n≤16；32 位指令时 n≤32	--	D、R	K、H	BIN16/32 位

备注：

1）16 位指令时，n<=16，32 位指令时，n<=32；

2）补入的最后一位放入进位标志位 M8022；

3）在位指定软元件情况下，只有 K4（16 位指令）和 K8（32 位指令）是有效的。

➢举例

操作前：（D0）=0x1234；
ROL D0 K4
操作后：（D0）=0x2341；
操作前：（D1，D0）=0x12345678；
DROL D0 K20
操作后：（D1，D0）=0x67812345；

FNC32 - RCR：带进位循环右移

➢功能说明

D. 中的数据连带进位标志位 M8022,一起向右移动 n 个二进制位,移出的低位连带标志位 M8022 的数据循环进入 D.的高位，最后移出的位值移入标志位 M8022。

➢指令格式

【D】RCR【P】 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存循环右移数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
n	旋转移动的位数，16 位指令时 n≤16；32 位指令时 n≤32	--	D、R	K、H	BIN1632 位

➢举例

操作前：（D0）=0xF0C5，（M8022）=1；
RCR D0 K4
操作后：（D0）=0xBF0C，（M8022）=0；
操作前：（D1，D0）=0xF00000C5，（M8022）=1；
DRCR D0 K4
操作后：（D1，D0）=0xBF00000C，（M8022）=0；

FNC33 - RCL：带进位循环左移

➢功能说明

D.中的数据连带进位标志位 M8022，一起向左移动 n 个二进制位，移出的高位连带标志位 M8022 的数据循环进入 D.的低位，最后移出的位值移入标志位 M8022。

➢指令格式

【D】RCL【P】 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存循环左移数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16 位
n	旋转移动的位数，16 位指令时 n≤16；32 位指令时 n≤32	--	D、R	K、H	BIN16 位

备注：

1）16 位指令时，n<=16，32 位指令时，n<=32；

2）进位标志位 M8022，会参与到指令操作中；

3）在位指定软元件情况下，只有 K4（16 位指令）和 K8（32 位指令）是有效的。

➢举例

操作前：（D0）=0xB30F，（M8022）=0；

RCL D0 K4

操作后：（D0）=0x30F5，（M8022）=1；

操作前：（D1，D0）=0xB300000F，（M8022）=0；

DRCL D0 K4

操作后：（D1，D0）=0x300000F5，（M8022）=1；

FNC34 - SFTR：位右移

➢功能说明

D.开始的 n1 个位软元件，右移 n2 位。高位补 n2 位的 S.。

➢指令格式

SFTR【P】 S. D. n1 n2

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	右移后在移位数据中保存的起始位软元件编号	X、Y、M、S	--	--	位
D.	保存的起始位软元件编号	Y、M、S	--	--	BIN16 位
n1	移位数据的位数据长度，n2<=n1<=1024	--	--	K、H	BIN16 位
n2	右移的位点数，n2<=n1<=1024	--	D、R	K、H	

➢举例

操作前：（X1）=0，（X0）=1，（M3）=0，（M2）=1，（M1）=1，（M0）=1；

SFTR X0 M0 K4 K2

操作后：（X1）=0，（X0）=1，（M3）=0，（M2）=1，（M1）=0，（M0）=1；

FNC35 - SFTL：位左移

➢功能说明

D.开始的 n1 个位软元件，左移 n2 位。低位补 n2 位的 S.。

➢指令格式

SFTL【P】 S. D. n1 n2

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	左移后在移位数据中保存的起始位软元件编号	X、Y、M、S	--	--	位
D.	保存的起始位软元件编号	Y、M、S	--	--	BIN16 位
n1	移位数据的位数据长度，n2<=n1<=1024	--	--	K、H	BIN16 位
n2	左移的位点数，n2<=n1<=1024	--	D、R	K、H	--

➢举例

操作前：（X1）=1，（X0）=1，（M4）=1，（M3）=1，（M2）=0，（M1）=1，（M0）=0；
SFTR X0 M0 K5 k2
操作后：（X1）=1，（X0）=1，（M4）=0，（M3）=1，（M2）=0，（M1）=1，（M0）=1；

FNC36 - WSFR 字右移

➢功能说明

将以 D.为首地址字元件组合向右移动 n2 为，其高位由 n2 位字元件组合 S.移入，移出的 n2 个低位被舍弃，而字元件组合 S.保持原值不变。

➢指令格式

WSFR【P】 S. D. n1 n2

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	右移后在移位数据中保存的起始字软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	位
D.	保存右移数据的起始字软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
n1	移位数据的字数据长度，n2<=n1<=512	--	--	K、H	BIN16 位
n2	右的字点数，n2<=n1<=512	--	D、R	K、H	--

➢举例

操作前：（D20）=25，（D21）=15，（D0）=20，（D1）=30，（D2）=40，（D3）=50，（D4）=60，（D5）=70，（D6）=80，（D7）=90；
WSFR D20 D0 K8 K2
操作后：（D20）=25，（D21）=15，（D0）=40，（D1）=50，（D2）=60，（D3）=70，（D4）=80，（D5）=90，（D6）=25，（D7）=15；

FNC37 - WSFL：字左移

➢功能说明

将以 D.为首地址字元件组合向左移动 n2 为，其低位由 n2 位字元件组合 S.移入，移出的 n2 个高位被舍弃，而字元件组合 S.保持原值不变。

➢指令格式

WSFL【P】 S. D. n1 n2

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	左移后在移位数据中保存的起始字软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
D.	保存左移数据的起始字软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
n1	移位数据的字数据长度，n2<=n1<=512	--	--	K、H	BIN16 位
n2	左移的字点数，n2<=n1<=512	--	D、R	K、H	BIN16 位

➢举例

操作前：（D0）=1，（D10）=25，（D11）=20，（D12）=15，（D13）=10，（D14）=5；
WSFL D0 D10 K5 k1
操作后：（D0）=1，（D10）=1，（D11）=25，（D12）=20，（D13）=15，（D14）=10；

FNC38 - SFWR：移位写入【先入先出/先入后出控制用】

➢功能说明

为控制先进先出的数据写入指令。执行时 D.先复位为零后作计数用。按执行次数，S.的内容从目标操作数之后的一个开始依次往后放入寄存器中。直到计数到达 n-1 个后不再填充寄存器只计数。

➢指令格式

SFWR【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据的起始字软元件编号（最前端为指针，数据是从 S.+1 开始的）	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
D.	保存先出的数据的字软元件编号 2≤n≤512	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
n	请指定被保存的数据点数+1 的值，2≤n≤512	--	--	K、H	BIN16 位

备注：

1）2<n<=512；

2）执行次数超过 n-1 次进位标志 M8022 置位。

➢举例

操作前：（D0）=0x5，（D10）=0，（D11）=0，（D12）=0，（D13）=0x123，（M8022）=0；
SFWR D0 D10 K3
操作一次后：（D0）=0x5，（D10）=1，（D11）=0x5，（D12）=0，（D13）=0x123，（M8022）=0；
改变 D0 的值再次重复操作：（D0）=0x7；
操作二次后：（D0）=0x7，（D10）=2，（D11）=0x5，（D12）=0x7，（D13）=0x123，（M8022）=0；
改变 D0 的值再次重复操作：（D0）=0x9；
操作三次后：（D0）=0x9，（D10）=2，（D11）=0x5，（D12）=0x7，（D13）=0x123，（M8022）=1；

FNC39 - SFRD：移位读出【先入先出控制用】

➢功能说明

为控制先进先出的数据读出指令。S.每次执行减一，直至减至零。每次执行时，总执行次数 X，S.之后的第 X 个寄存器的内容，放入从 D.开始的第 X 个寄存器。直到 S.减 n-1 时不再减少也不再填充寄存器。

➢指令格式

SFRD【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据的起始字软元件编号（最前端为指针，数据是从 S.+1 开始的）	--	KnY、KnM、KnS、T、C、D、R	--	BIN16 位
D.	保存先出的数据的字软元件编号，2≤n≤512	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16 位
n	请指定被保存的数据点数+1 的值，2≤n≤512	--	--	K、H	BIN16 位

备注：
1）2<n<=512；
2）源操作数减至零后，再次执行零点标志 M8020 置位。

➢举例

操作前：（D10）=2，（D20）=0，（D11）=0x2，（D12）=0x3，（D13）=123，（M8020）=0；
SFWR D10 D20 K3
操作一次后：（D10）=1，（D20）=0x2，（D11）=0x2，（D12）=0x3，（D13）=123，（M8020）=0；
操作二次后：（D10）=0，（D20）=0x3，（D11）=0x3，（D12）=0x3，（D13）=123，（M8020）=0；
操作三次后：（D10）=0，（D20）=0x3，（D11）=0x3，（D12）=0x3，（D13）=123，（M8020）=1；

7-5 数据处理指令

FNC40 - ZRST: 成批复位

➢功能说明

复位 S1.至 S2.的全部寄存器。

➢指令格式

ZRST【P】 D1. D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D1.	成批复位的最前端的位/字软元件编号	Y、M、S	T、C、D、R	--	BIN16/BIN32 位
D2.	成批复位的末尾的位/字软元件编号	Y、M、S	T、C、D、R	--	BIN16/BIN32 位

备注：

1）D1.D2.需为同一种软元件且 D1.<=D2.，如果 D1.>D2.则只复位 D1.中的软元件；

2）该指令为 16 位指令，但 D1.D2.也适用 32 位计数器。但是不能 16 位 32 位混合，如 D1.为 16 位计数器，D2.位 32 位计数器。

➢举例

操作前：（D0）=0x1234, （D1）=0x2345, （D2）=0x3456, （D3）=0x4567, （D4）=0x5678;

ZRST D0 D4

操作后：（D0）=0, （D1）=0, （D2）=0, （D3）=0, （D4）=0;

FNC41 - DECO：译码

➢功能说明

由源址 S.所表示的二进制值 m 使终址 D.中编号为 m 的位元件或字元件中 bm 位置 ON。S.的位数指定为 2n。

➢指令格式

DECO【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要译码的数据，或是数据的字软元件编号	X、Y、M、S	T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	保存译码结果数据的位/字软元件编号	Y、M、S	T、C、D、R	--	BIN16/BIN32 位
n	保存译码结果的软元件的位点数（0≤ n ≤8，n=0 时不处理）	--	--	K、H	--

➢举例

操作前：（D0）=2，（Y0）=0，（Y1）=0，（Y2）=0，（Y3）=0；

DECO D0 Y0 K3

操作后：（D0）=2，（Y0）=0，（Y1）=0，（Y2）=1，（Y3）=0；

FNC42 - ENCO：编码

➢功能说明

把 S.中置 ON 的位元件或字元件中置 ON 的位的位置值转换成二进制整数传送到 D.。S 的位数指定为 2n 位；

➢指令格式

ENCO【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要编码的数据，或是数据的字软元件编号	X、Y、M、S	T、C、D、R、V、Z	--	BIN16 位
D.	保存结果数据的字软元件编号	--	T、C、D、R、V、Z	--	BIN16 位
n	保存编码结果的软元件的位点数（0≤ n ≤8，n=0 时不处理）	--	--	K、H	BIN16 位

备注：

1）1<=n<=8，n=0 时不处理；n=8 时，编码指令的 S.如果是位软件，其点数是 28=256 点；

2）S.是位元件时 n<=8,是字软原件时 n<=4；

3）S.内多个位是 1 时，忽略低侧，另外全部位都为时出现运算出错。

➢举例

操作前：（M10）=0，（M11）=0，（M12）=0，（M13）=1，（M14）=0，（M15）=0，（M16）=0，（M17）=0,（D0）=0；

ENCO M10 D0 K3

操作后：（M10）=0，（M11）=0，（M12）=0，（M13）=0，（M14）=0，（M15）=0，（M16）=0，（M17）=0,（D0）=3；

FNC43 - SUM：ON 位数

➢功能说明

S.在二进制下 1 的位数存入 D.中；

➢指令格式

【D】SUM【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	保存结果数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/BIN32 位

➢举例

操作前：（D0）=0x7F，（D1）=0；

SUM D0 D1

操作后：（D0）=0x7F，（D1）=7；

操作前：（D1，D0）=0xFFFFFFFF，（D3，D2）=0；

DSUM D0 D2

操作后：（D1，D0）=0xFFFFFFFF，（D3，D2）=32;

FNC44 - BON：ON 位的判断

➢功能说明

判断 S.第 n 位是否为零；

➢指令格式

【D】BON【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	驱动的位软元件编号	Y、M、S	--	--	BIN16/BIN32 位
n	要判定的位位置【16 指令：0≤ n≤15；32 指令：0≤ n≤31】	--	D、R	K、H	BIN16/BIN32 位

备注：

1）16 指令：0≤ n≤15；32 指令：0≤ n≤31；

2）n 超出允许范围时，会置位运算错误标志位 M8067。

➢举例

操作前：（D0）=4，（M20）=0；

B0N D0 M20 K2

操作后：（D0）=4，（M20）=1；

FNC45 - MEAN：平均值

➢功能说明

S.起 n 个软元件求平均值，放入 D.中；

➢指令格式

【D】MEAN【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存想要的平均值数据的起始软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R	--	BIN16/BIN32 位
D.	保存取得的平均值数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/BIN32 位
n	平均数据数（1<=n<=64）	--	D、R	K、H	BIN16/BIN32 位

备注：1<=n<=64，n 超出允许范围时，会置位运算错误标志位 M8067；

➢举例

操作前：（D0）=3，（D1）=5，（D2）=7，（D3）=0；

MEAN D0 D3 K3

操作后：（D0）=3，（D1）=5，（D2）=7，（D3）=5；

操作前：（D1，D0）=0x111111，（D3，D2）=0x222222，（D5，D4）=0x333333，（D7，D6）=0；

DMEAN D0 D6 K3

操作后：(D1，D0)=0x111111，(D3，D2)=0x222222，(D5，D4)=0x333333，(D7，D6)=0x222222；

FNC46 - ANS：信号报警器置位

➢功能说明

用于驱动信号报警器的方便指令；

➢指令格式

ANS S. m D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	判断时间的计时定时器编号	--	T	--	BIN16 位
m	判断时间的数据【m=1~32767(100ms 单位)】	--	D、R	K、H	BIN16 位
D.	设置的信号报警器软元件	S	--	--	BIN16 位

备注：

1）m 它的范围在 0 到 32767 之间超出范围会报错 M8067；

2）D.它的范围是 S900 到 S999 之间超出范围会报错。

➢举例

操作前：（D901）=0;

ANS T0 K10 S901

操作后：（T0=0 ；S901=0） → （计数到 T0=10；S901=0 ）→ （T0=0；S901=1）

定时 1 秒钟，导通一秒以后 S901 被置位，置位以后，不管 ANS 语句是否导通，S901 被一直动作。

M8049 动作时，将会把 S900 到 S999 中导通的，最小一个编号值给存到 D8049 中。

M8049 动作时，S900 到 S999 中任意一个为 ON 时，M8048，将动作报警（为 ON）

FNC47 - ANR：信号报警器复位

➢功能说明

S900 到 S999 指令 ANS 将报警给解除；

➢指令格式

ANR【P】

➢举例

操作前：

ANR

操作后：当 ANR 导通时，会从 S900 到 S999（ON）状态最小编号，开始依次复位。

ANRP

操作后：当 ANRP 导通一次，会把 S900 到 S999（ON）状态最小编号，复位。

FNC48 - SQR：BIN 开方运算

➢功能说明

进行平方根开方运算指令；

➢ 指令格式

【D】SQR【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要被开平方根运算数据的字软元件编号	--	D、R	K、H	BIN16/BIN32 位
D.	保存被执行了开平方根运算数据的数据寄存器编号	--	D、R	--	BIN16/BIN32 位

备注：

1）进行平方根运算的指令；

2）仅在 S.是正数才有效，如果是负数时运算错误标志位 M8067 会工作，指令不被执行；

3）运算结果舍去小数为整数。舍去时，借位标志位会动作；

4）结果为零时，令标志位 M8020 会动作。

➢举例

操作前：（D11,D10）=4，（D13，D12）=0；

DSQR D10 D12

操作前：（D11,D10）=4，（D13，D12）=2.000；

FNC49 - FLT：BIN 整数转 2 进制浮点数

➢功能说明

S.内的 BIN 整数值，转换为 2 进制浮点值放入 D.及其后一软元件；

➢指令格式

【D】FLT【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 BIN 整数值的数据寄存器编号	--	D、R	--	BIN16/BIN32 位
D.	保存 2 进制浮点数（实数）的数据寄存器编号	--	D、R	--	BIN16/BIN32 位

备注：常数 K、H 在各浮点运算指令中被自动转换，因此在本 FLT 指令中不能使用。

➢举例

操作前：（D0）=1，（D13，D12）=0；

FLT D0 D12

操作后：（D0）=1，float（D13，D12）=0x3F80 即二进制浮点数 1.000；

操作前：（D1，D0）=1000000，（D13，D12）=0；

DFLT D0 D12

操作后：（D1，D0）=1000000，float（D13，D12）=0x49742400 二进制浮点值 1.000e+006;

7-6 高速处理 - FNC50~FNC59

脉冲输出类指令使用的特殊寄存器：

软元件	内容含义
M8029	指令执行完成标志位
M8144	Y2 停止脉冲输出的指令
M8145	Y0 停止脉冲输出的指令
M8146	Y1 停止脉冲输出的指令
M8147	Y0 脉冲输出中的监视（BUSY/READY）
M8148	Y1 脉冲输出中的监视（BUSY/READY）
M8149	Y2 脉冲输出中的监视（BUSY/READY）

编号	位数	出厂值	内容含义
D8140（低位）	32	0	Y0 输出位置当前值，应用脉冲指令 PLSY，PLSR 时，对脉冲输出值进行累加当前值
D8141（高位）			
D8142（低位）	32	0	Y1 输出位置当前值，应用脉冲指令 PLSY，PLSR 时，对脉冲输出值进行累加当前值
D8143（高位）			
D8160（低位）	32	0	Y2 输出位置当前值，应用脉冲指令 PLSY，PLSR 时，对脉冲输出值进行累加当前值
D8161（高位）			
D8136（低位）	32	0	Y0，Y1，Y2 输出脉冲合计数的累计值
D8137（高位）			

FNC50 - REF：输入输出刷新

FNC51 - REFF：输入刷新【带滤波设定】

FNC52 - MTR：矩阵输入

➢功能说明

以 8 点输入和输出（晶体管）的时间分割方式读取 8 点 Xn 列的输入信号（开关）的指令；

对 8 点的 S.输入和 n 点的 D1.晶体管输出进行时间分割控制，以便依次读取 8 点 n 列的输入信号，然后输出到 D2.中。

➢指令格式

MTR S. D1. D2. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	矩阵的行信号输入的起始软元件（X）编号 X000、X010、X020...最终的输入 X 编号为止（最低位的位数编号只能为 0）	X	--	--	位
D1.	矩阵的列信号输出的起始软元件（Y）编号 Y000、Y010、Y020...最终的输出 Y 编号为止（最低位的位数编号只能为 0）	Y	--	--	位
D2.	ON 输出目标地址的起始软元件（Y、M、S）编号 Y000、Y010、Y020，M000、M010、M020...，S000、S010、S020...最终的 Y、M、S 编号为止。	Y、M、S	--	--	位
n	设定矩阵输入的列数（K2~K8/H2~H8）	--	--	K、H	BIN 16 位

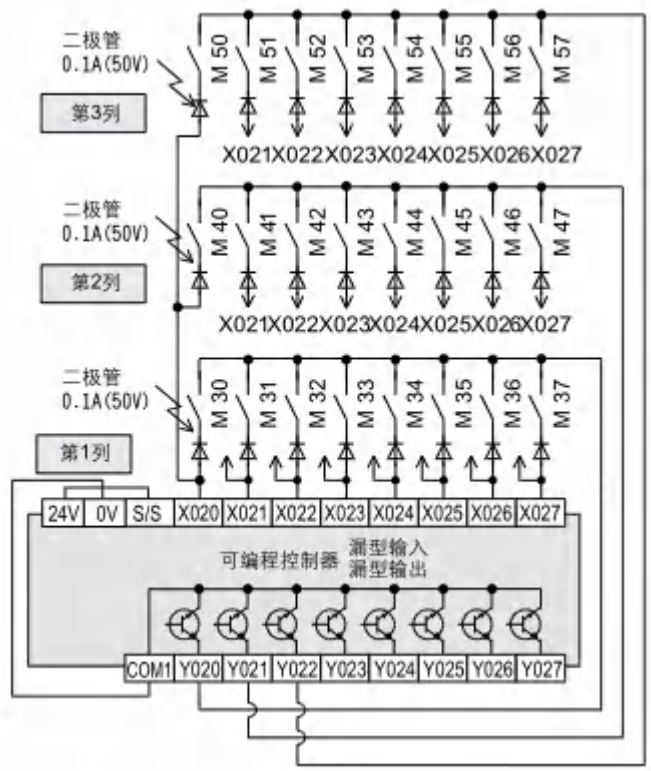
➢举例

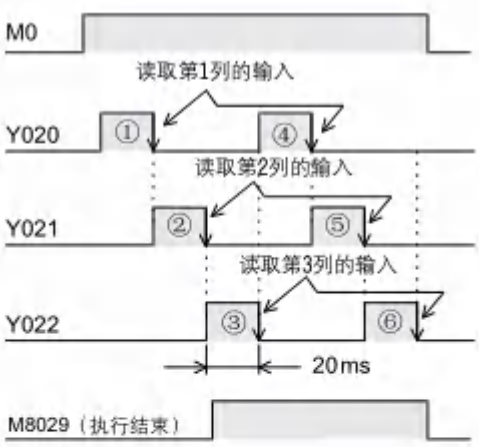
n=3 点的输出（Y20、Y21、Y22）依次重复置 ON。

每次依次反复，获得第 1 列，第 2 列，第 3 列的输入 8 点，并保存到 M30~M37，M40~M47，M50~M57 中。

下面的例子中，此程序是是以 CZK2/CZK3 系列的基本单元为例。

有关接线的情况，请参考使用的可编程控制器的下述手册。





FNC53 - HSCS：比较置位【高速计数器用】

➢功能说明

每次计数时，都将高速计数器的计数值和指定值做比较，然后立即置位外部输出（Y）的指令。

当 S2.中指定的高速计数器（C235~C255）的当前值，变成比较值【S1.+1，S1.】时比较值 K200 时为 199->200 或 201->200，位阮元件 D.被置位（ON）,与扫描周期无关。这个指令是接着高速计数器的计数处理之后执行比较的指令。

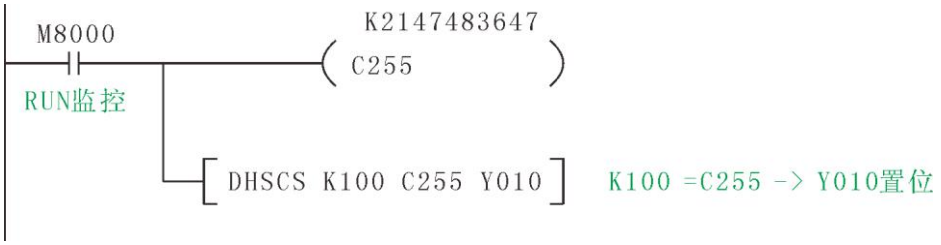
➢指令格式：

DHSCS S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	与高速计数器的当前值比较的数据，或是保存比较数据的字软元件编号	X	KnX、KnY、KnM、KnS、T、C、D、R、Z	K、H	BIN16/BIN32 位
S2.	高速计数器的软元件编号【C235~C255】	--	C	--	BIN16/BIN32 位
D.	一致后进行置位（ON）的位软元件编号	Y、M、S	C	--	位

➢举例

高速计数器 C255 的当前值从 99 变化为 100 或者从 101 变化为 100（计数）时，Y010 被置位（输出刷新）。



相关指令

和高速计数器一起使用的指令如下所示：

指令	说明
FNC53 - DHSCS	比较置位（高速计数器）
FNC54 - DHSCR	比较复位（高速计数器）
FNC55 - DHSZ	区间比较（高速计数器）
FNC189 - DHCMOV	高速计数器的传送
FNC280 - DHSCT	高速计数器数据表的比较

FNC54 - HSCR：比较复位【高速计数器用】

➢功能说明

每次计数时，都将告高速计数器的计数值和指定值做比较，然后立即复位外部输出（Y）的指令。

当 S2.中指定的高速计数器（C235~C255）的当前值，变成比较值【S1.+1，S1.】时比较值 K200 时为 199->200 或 201->200，位阮元件 D.被复位（ON）,与扫描周期无关。这个指令是接着高速计数器的计数处理之后执行比较的指令。

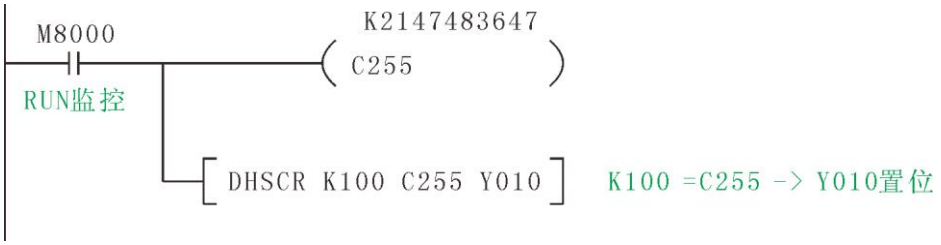
➢指令格式：

DHSCR S1. S2. S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	与高速计数器的当前值比较的数据，或是保存比较数据的字软元件编号	X	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	高速计数器的软元件编【C235~C255】	--	C	--	BIN16/BIN32 位
D.	一致后进行复位（OFF）的位软元件编号	Y、M、S	C	--	位

➢举例

高速计数器 C255 的当前值从 99 变化为 100 或者从 101 变化为 100（计数）时，Y010 被置位（输出刷新）。



1. 自我复位的梯形图实例

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

相关指令

和高速计数器一起使用的指令如下所示：

指令	说明
FNC53 - DHSCS	比较置位（高速计数器）
FNC54 - DHSCR	比较复位（高速计数器）
FNC55 - DHSZ	区间比较（高速计数器）
FNC189 - DHCMOV	高速计数器的传送
FNC280 - DHSCT	高速计数器数据表的比较

FNC55 - HSZ：区间比较【高速计数器用】

FNC56 - SPD：脉冲密度

➢功能说明

采用中断输入方式对指定时间内的输入脉冲计数的指令。

根据版本不同，该指令的功能可能会不同。

➢指令格式：

【D】SPD　S1.　S2.　D.

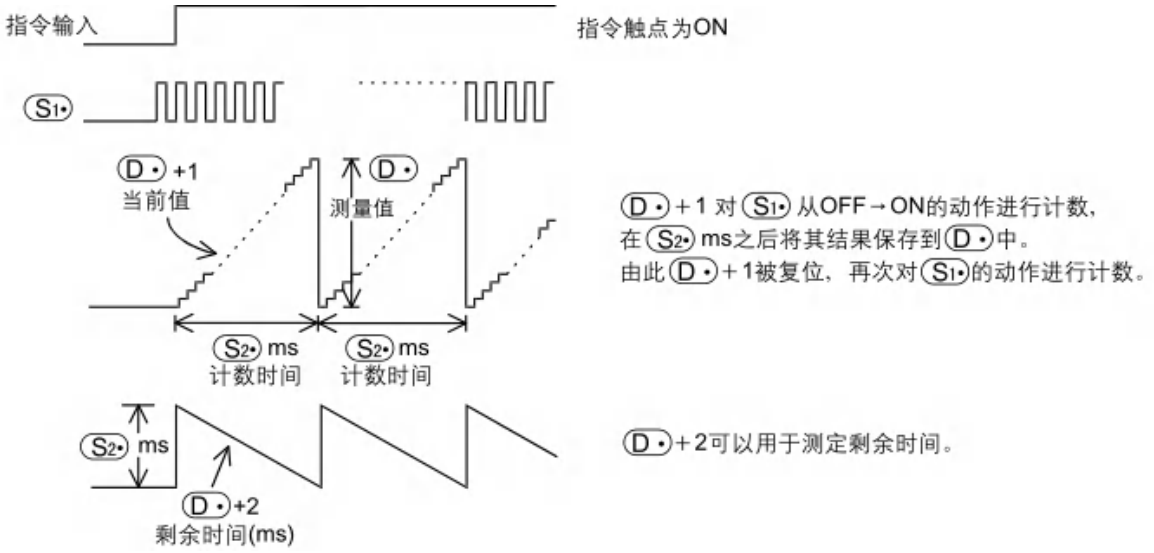
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	输入（X）脉冲的软元件编号	X	--	--	位
S2.	时间（ms）数据或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	保存脉冲密度数据的起始	--	T、C、D、R、V、Z	--	BIN16/BIN32 位

➢动作说明

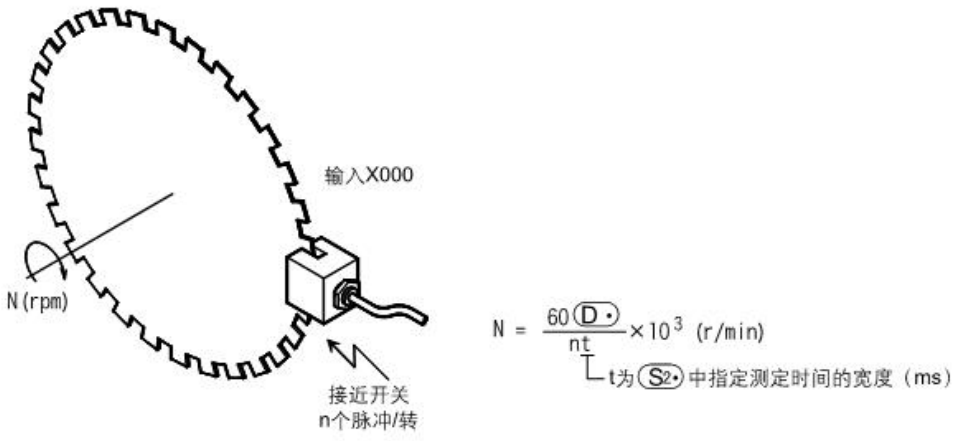
1、16 位运算(SPD)

只在 S2. * 1 ms 时间内对输入 S1 的脉冲进行计数，测定值保存到 D.，当前值保存到 D.+1,剩余时间保存到 D.+2 ms 中

1）时序图



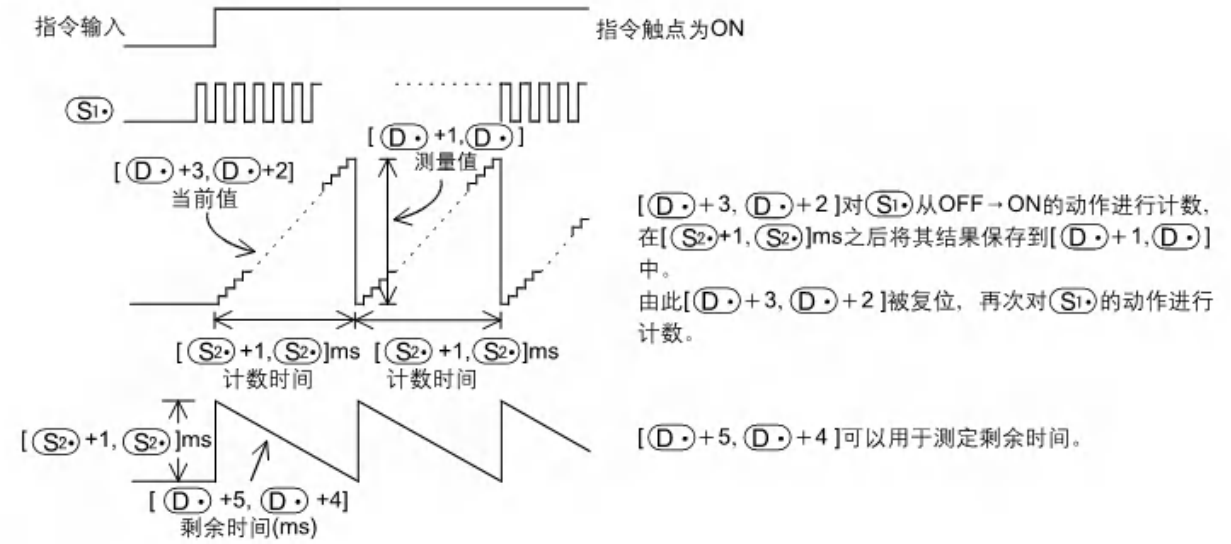
2）测量值 D.的值，如下所示与转数成比例



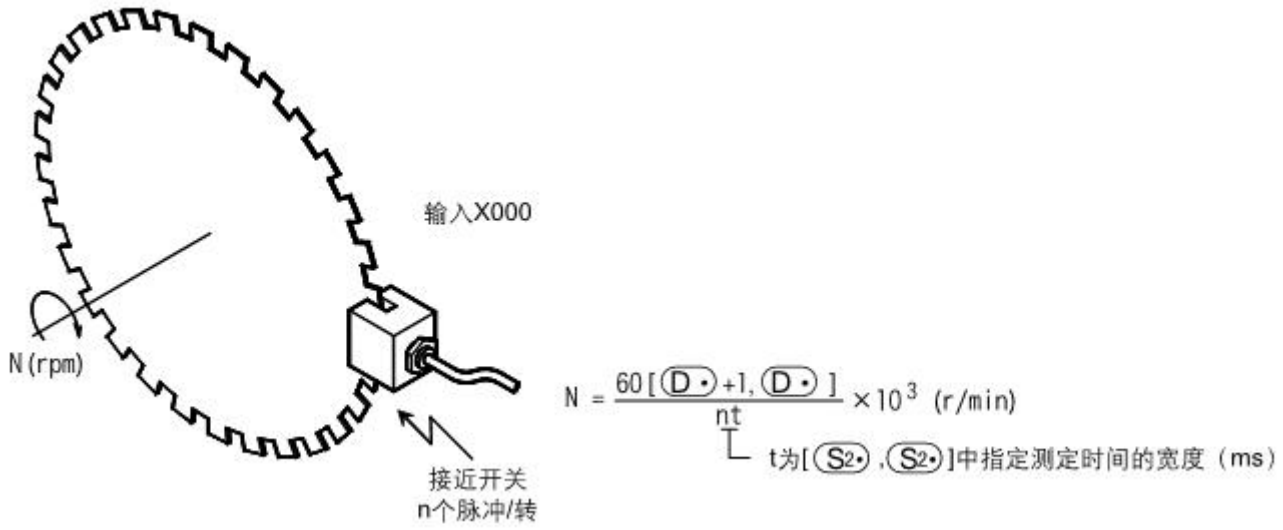
1、32 位运算（DSPD）

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

1）时序图



2）测量值【D.+1，D.】的值，如下所示与转数成比例



➢注意要点

1、S1.输入的输入规格

1）指定的 S1.输入的 X000~X007 不能与下面的用途重复使用。

1.1）高速计数器

1.2）输入中断

1.3）脉冲捕捉

1.4）DSZR 指令

1.5）DVIT 指令

1.6）ZRN 指令

2）这个指令，每 1 点输入为 1 个指令一下。

3）输入 X000~X007 的 ON/OFF 的最大频率，如下表所示。

使用的输入编号	最大输入频率
	基本单元
X000~X005	100KHz
X0006，X0007	10KHz

2、占用软元件

2.1）16 位运算的情况，以 D.为起始占用 3 点。

2.2）32 位运算的情况，以 D.为起始占用 6 点。

FNC57 - PLSY：脉冲输出指令

➢功能说明

发出脉冲信号用的指令。

在 D.指定的输出口，输出一个脉冲串。脉冲串频率在源操作数 S1.内，数量在源操作数 S2.内，占空比为 50%；

➢指令格式

【D】PLSY　S1.　　S2.　　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	频率值（Hz）或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	脉冲个数或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	输出脉冲的位软元件（Y）编号	Y	--	--	位

备注：

1）16 位指令：1<=S1<=32767；0<=S2<=32767。

2）32 位指令：1<=S1<=100000；0<=S2<=2147483647。

3）目标操作数必须为支持脉冲输出的 Y 软元件。

➢举例

PLSY　K1000　K100　Y0

操作后：Y0 通道输出一个频率为 1000Hz，数量为 100 个，占空比为 50%的脉冲串。

FNC58 - PWM：脉宽调制

➢功能说明

在 D.指定的输出口，输出脉冲。脉冲的脉宽在源操作数 S1.内，周期在源操作数 S2.内；

➢指令格式

PWM　S1.　　S2.　　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	脉宽（ms）数据或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	周期（ms）数据或是保存数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	输出脉冲的软元件（Y）编号	Y	--	--	位

备注：

1）0<=S1<=32767；1<=S2<=32767；

2）目标操作数必须为支持脉冲输出的 Y 软元件。

➢举例

PWM　K1　K2　Y1

操作后：Y1 通道输出一个周期为 2mS，脉宽为 1mS 的脉冲。

FNC59 - PLSR：带加减速的脉冲输出

➢功能说明

在 D.指定的输出口，输出一个开始时十级加速至最高频，结束时十级减速至最低频的脉冲串。脉冲串最高频率在源操作数 S1.内，总数量在源操作数 S2.内，加减速时间在 S3.内，占空比为 50%；

➢指令格式

【D】PLSR　S1.　　S2.　　S3.　　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存最高频率（Hz）数据，或是数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	保存总的脉冲数（PLS）数据，或是数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S3.	保存加减时间（ms）数据，或是数据的字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
D.	输出脉冲的软元件（Y）编号	Y	--	--	位

备注：

1）16 位指令：1<=S1<=32767，32 位指令：1<=S1<=100000；

2）16 位指令：110<=S2<=32767，32 位指令：110<=S2<=2147483647；

3）1<=S3<=5000；

4）目标操作数必须为支持脉冲输出的 Y 软元件；

5）参数设置时需要注意是否合理，例如：最高频很低加速时间内无法达到最高频，这些情况仍会有脉冲输出，但不会满足所有设置条件。

➢举例

PLSR K10000 K10000 K50 Y2

操作后：Y2 通道输出一个总数量为 10000 个，占空比为 50%，频率开始时十级加速至 10000Hz，结束时十级减速的脉冲串。

7-7 方便指令 - FNC60~FNC69

FNC60 - IST：初始化状态

FNC61 - SER：数据检索

➢功能说明

从数据表中检索相同数据、以及最大值、最小值的指令。

➢指令格式

【D】SER【P】 S1. S2. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	检索相同数据、最大值、最小值的起始软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R	--	BIN16/BIN32 位
S2.	检索相同数据、最大值、最小值的参考值或是保存目标软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
D.	检索相同数据、最大值、最小值后，保存这些个数的起始软元件编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16/BIN32 位
n	检索相同数据、最大值、最小值的个数（16 位指令时：1~256，16 位指令时：1~128，）	--	D、R	K、H	BIN16/BIN32 位

➢动作说明

1、16 位运算（SER，SERP）

对以 S1.开始的 n 个数据进行检索，检索与 S2.的数据相同的数据，并将结果保存在 D.~D+4 中。

1.1 被检索的数据的内容及结果

1）存在相同数据时

在以 D.开始的 5 个软元件中，保存相同数据的个数、初次/最终的位置以及最大值、最小值的位置。

2）不存在相同数据时

在以 D.开始的 5 个软元件中，保存相同数据的个数、初次/最终的位置以及最大值、最小值的位置。但是，在以 D.开始的 3 个软元件（相同数据的个数、初次/最终的位置）中，0 被保存。

1.2 动作例子

1）检索结果表格的结构及数据示例

被检索软元件 S1.	被检索数据 S1 的值	比较数据 S2 的值	数据的位置	检索结果		
				最大值 D.+4	一致 D.	最小值 D.+3
S1.	K100	K100	0		●（初次）	
S1.+1	K111		1			
S1.+2	K100		2			
S1.+3	K98		3			
S1.+4	K123		4			
S1.+5	K66		5			●
S1.+6	K100		6		●（最终）	
S1.+7	K95		7			
S1.+8	K210		8	●		
S1.+9	K88		9			

2）检索结果表格

操作数种类	内容	检索结果项目
D.	3	相同数据的个数
D.+1	0	相同数据的位置（初次）
D.+2	6	相同数据的位置（最终）
D.+3	5	最小值的最终位置
D.+4	8	最大值的最终位置

2、32 位运算（DSER，DSERP）

对以 S1.+1，S1.开始的 n 个数据进行检索，检索与 S2.+1，S2.的数据相同的数据，并将结果保存在【D.+1，D.】~【D.+9，D.+8】

1.1 被检索的数据的内容及结果

1）存在相同数据时

在以 D.+1，D.开始的 5 个 32 位软元件中，保存相同数据的个数、初次/最终的位置以及最大值、最小值的位置。

2）不存在相同数据时

在以 D.+1，D.开始的 5 个 32 位软元件中，保存相同数据的个数、初次/最终的位置以及最大值、最小值的位置。但是，在以 D.开始的 3 个 32 位数据的（相同数据的个数、初次/最终的位置）中，0 被保存。

1.2 动作例子

1）检索结果表格的结构及数据示例

被检索软元件 S1.	被检索数据 S1 的值	比较数据 S2 的值	数据的位置	检索结果		
				最大值 D.+4	一致 D.	最小值 D.+3
S1.+1，S1.	K100000	K100000	0		●（初次）	
S1.+3，S1.+2	K110100		1			
S1.+5，S1.+4	K100000		2			
S1.+7，S1.+6	K98000		3			
S1.+9，S1.+8	K123000		4			
S1.+11，S1.+10	K66000		5			●
S1.+13，S1.+12	K100000		6		●（最终）	
S1.+15，S1.+14	K95000		7			
S1.+17，S1.+16	K910000		8	●		
S1.+19，S1.+18	K910000		9			

2）检索结果表格

操作数种类	内容	检索结果项目
D.+1，D.	3	相同数据的个数
D.+3，D.+2	0	相同数据的位置（初次）
D.+5，D.+4	6	相同数据的位置（最终）
D.+7，D.+6	5	最小值的最终位置
D.+9，D.+8	9	最大值的最终位置

➢注意要点

1、大小的比较

以代数方式执行。

2、存在多个最小值、最大值时数据中存在多个最小值、最大值时，分别保存各个后侧的位置。

3、软元件的占用点数

驱动指令后，检索结果 D.会占用如下的软元件点数。

请注意不要与机器其他控制器中使用的软元件重复。

1）16 位运算时，占用【D.，D.+1，D.+2，D+3，D.+4】5 点

1）32 位运算时，占用【D.+1，D.】,【D.+3，D.+2】,【D.+5，D.+4】,【D.+7，D.+6】,【D.+9，D.+8】10 点

FNC62 - ABSD：凸轮控制绝对方式

➢功能说明

对应计数器的当前值，产生多个输出模式的指令。

注意要点：

1、高速计数器（C235~C255）的指定。

DABSD 指令中也可以指定高速计数器。

但是此时，对于计数器的当前值，在输出模式中会由于扫描周期而造成响应延迟。

需要响应性时，请使用 HSZ 指令进行表格的高速比较功能，或是使用 HSCT 指令。

2、在 S1.中指定位软元件的位数时

1）软元件编号

请指定 16 的倍数（0、16、32、64）。

2）位数

ABSD(16 位运算)时仅为 K4；DABSD(32 位运算)时仅为 K8。

3、其他的注意事项

由 n（1 ≤ n ≤64）的值决定对象的输出点数。即使指令输入为 OFF，输出也不改变。

➢指令格式：

【D】ABSD S1. S2. D. n

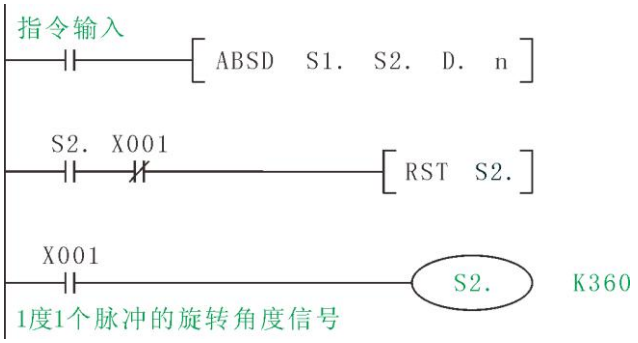
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存表数据（上升沿、下降沿）的起始软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	与表格数据比较的当前值监控用计数器编号	--	C	K、H	BIN16/BIN32 位
D.	输出的起始位软元件编号	Y、M、S	--		位
n	表格的行数以及输出的位软元件的点数（1≤ n ≤64）	--	--	K、H	BIN16 位

➢动作说明

1、16 位运算（ABSD）

平台旋转 1 次（0~360 度）期间，使输出 ON/OFF,以此为例进行说明。（1 度 1 个脉冲的旋转角度信号）。

将 S1.开始的 n 行表格数据（占用 n 行*2 点），与计算器的当前值 S2.做比较，在旋转 1 次期间，对 D.开始的连续 n 点输出进行 ON/OFF 控制。



1）先使用传送指令，在 S1.~S1+2n+1 中写入如下所示的数据。

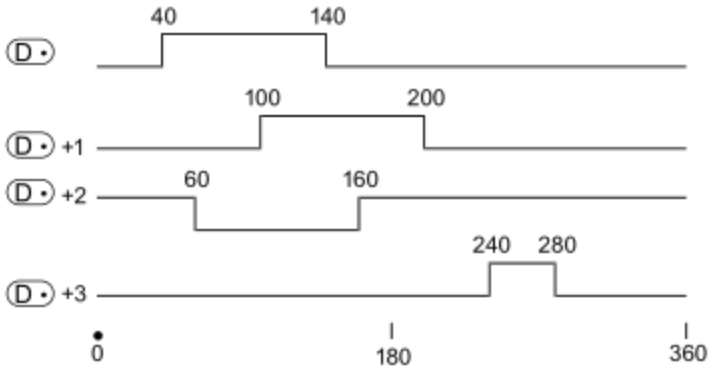
上升点		下降点		对象输出
	数据值（例）		数据值（例）	
(S1·)	40	(S1·)+1	140	(D·)
(S1·)+2	100	(S1·)+3	200	(D·)+1
(S1·)+4	160	(S1·)+5	60	(D·)+2
(S1·)+6	240	(S1·)+7	280	(D·)+3
⋮	–	⋮	–	⋮
(S1·)+2n		(S1·)+2n+1		(D·)+n-1

例如，上升点数据在偶数编号的软元件，下降点数据在奇数编号的软元件中，以16位数据进行保存。

2）输出模式

指令输入为 ON 后，以 D.开始的 n 点也如下进行变化。

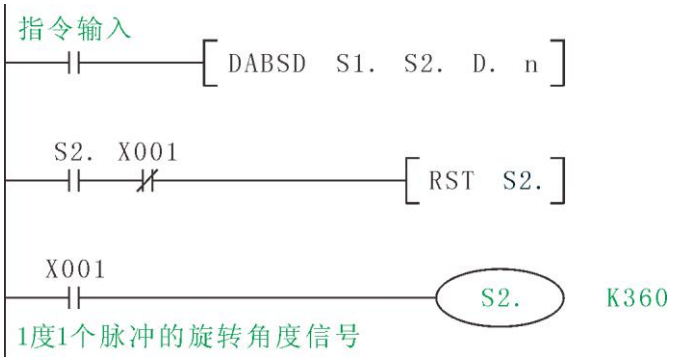
各上升点/下降点，可通过改写 S1~S1+n*2 的数据以个别进行变更。



2、32 位运算（DABSD）

平台旋转 1 次（0~360 度）期间，使输出 ON/OFF,以此为例进行说明。（1 度 1 个脉冲的旋转角度信号）。

将【S1.+1，S1.】开始的 n 行表格数据（占用 n 行*4 点），与计算器的当前值 S2.做比较，在旋转 1 次期间，对 D.开始的连续 n 点输出进行 ON/OFF 控制。



1）先使用传送指令，在【S1.，S1.+1】~【S1.，S1.+1】+2n+1 中写入如下所示的数据。

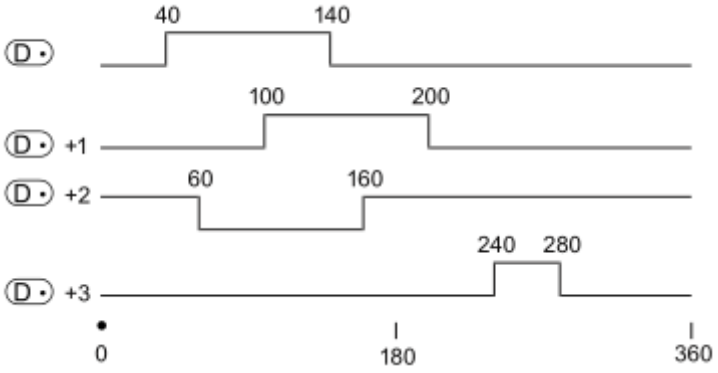
上升点	数据值（例）	下降点	数据值（例）	对象输出
[(S1·)+1, (S1·)]	40	[(S1·)+3, (S1·)+2]	140	(D·)
[(S1·)+5, (S1·)+4]	100	[(S1·)+7, (S1·)+6]	200	(D·)+1
[(S1·)+9, (S1·)+8]	160	[(S1·)+11, (S1·)+10]	60	(D·)+2
[(S1·)+13, (S1·)+12]	240	[(S1·)+15, (S1·)+14]	280	(D·)+3
⋮	–	⋮	–	⋮
[(S1·)+4n+1, (S1·)+4n]		[(S1·)+4n+3, (S1·)+4n+2]		(D·)+n-1

例如，上升点数据在偶数编号的软元件，下降点数据在奇数编号的软元件中，以32位数据进行保存。

2）输出模式

指令输入为 ON 后，以 D.开始的 n 点也如下进行变化。

各上升点/下降点，可通过改写【S1.+1，S1】~【S1.+（n*2）+3，S1.+（n*2）+2】，的数据以个别进行变更。



FNC63 - INCD：凸轮控制相对方式

➢功能说明

使用一对计数器，产生多个输出模式的指令。

➢指令格式：

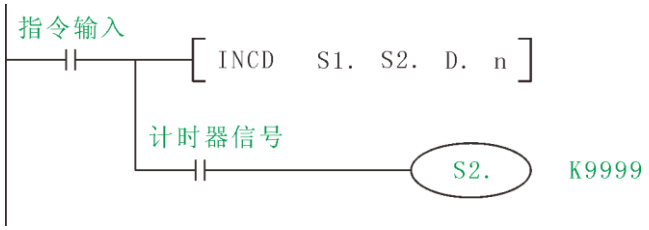
INCD S1. S2. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存设定值的起始字软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/BIN32 位
S2.	监控当前值用的计数器的起始编号	--	C	K、H	BIN16/BIN32 位
D.	输出的起始位软元件编号	Y、M、S	--		位
n	输出的位软元件的点数（1≤ n ≤64）	--	--	K、H	BIN16 位

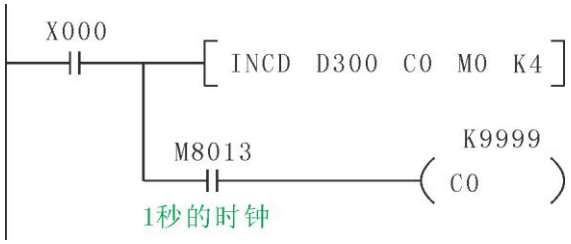
➢动作说明

1、16 位运算（INCD）

将 S1.开始的 n 行表格数据（占用 n 行*1 点），与计数器的当前值 S2.进行比较，如果一致，则复位，并依次对输出进行 ON/OFF 控制。



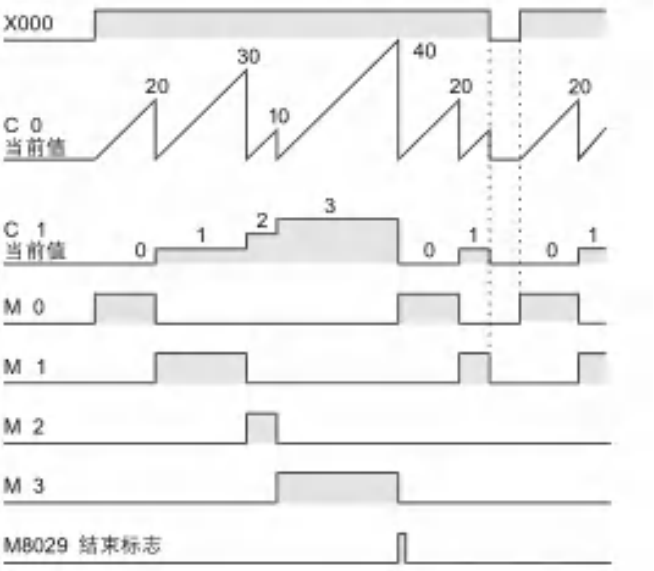
➢举例



1）时序图

使用传送指令，写入下表中的数据。

保存软元件		输出	
	数据值（例）		例如
(S·)	D300=20	(D·)	M0
(S·)+1	D301=30	(D·)+1	M1
(S·)+2	D302=10	(D·)+2	M2
(S·)+3	D303=40	(D·)+3	M3
⋮	⋮	⋮	⋮
(S·)+n-1	—	(D·)+n-1	—



2）指令触点如果为 ON，则 M0 输出也为 ON。

3）当 C0 的当前值达到比较值 D300 时，输出（M0）被复位，工序计数器 C1 的计数值被+1，计数器 C0 的当前值也被复位。

4）接下来的输出 M1 为 ON。

5）C0 当前值与比较值 D301 比较，如果达到比较值，则输出 M1 被复位，工序计数值被+1,计数器 C0 的当前值也被复位。

6）同样地方法一直比较到 n(K4，1≤n≤64)指定的点数为止。

7）在 n 指定的最后工序结束后，执行结束标志 M8029 保持 1 个运算周期为 ON。

8）由于 M8029 是对个指令的指令使用执行结束的标志位，所以直接在指令的后面作为触点使用，请务必作为该指令专用的结束标志。

9）回到最初，重复输出。

FNC64 - TTMR：示教定时器

➢功能说明

测量 TTMR 指令为 ON 的时间的指令。

采用按钮来调节定时器的设定时间的情况下，可使用。

➢指令格式

TTMR　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存示教数据的软元件编号	--	D、R	--	BIN16 位
n	示教数据乘以的倍率数（K0~K2/H0~H2）	--	D、R	K、H	BIN16 位

➢动作说明

1、16 位运算（TIMER）

以秒为单位，对指令输入（按键）的按下时间进行测量，然后乘以倍率（10 的 n 次方）后传送到 D.中

被传到 D.中的时间是这样的，按下时间为 τ 0(1 秒单位)时，根据 n 的倍率得到实际的 D.值，如下所示。

n	倍率	D.
K0	τ 0	D.*1
K1	10 τ 0	D.*10
K2	100 τ 0	D.*100

还有以下方便指令。

指令	内容
HOUR	以 1 小时为单位，累计计算 HOUR 指令为 ON 时间，在指定的时间输出的指令

➢注意要点

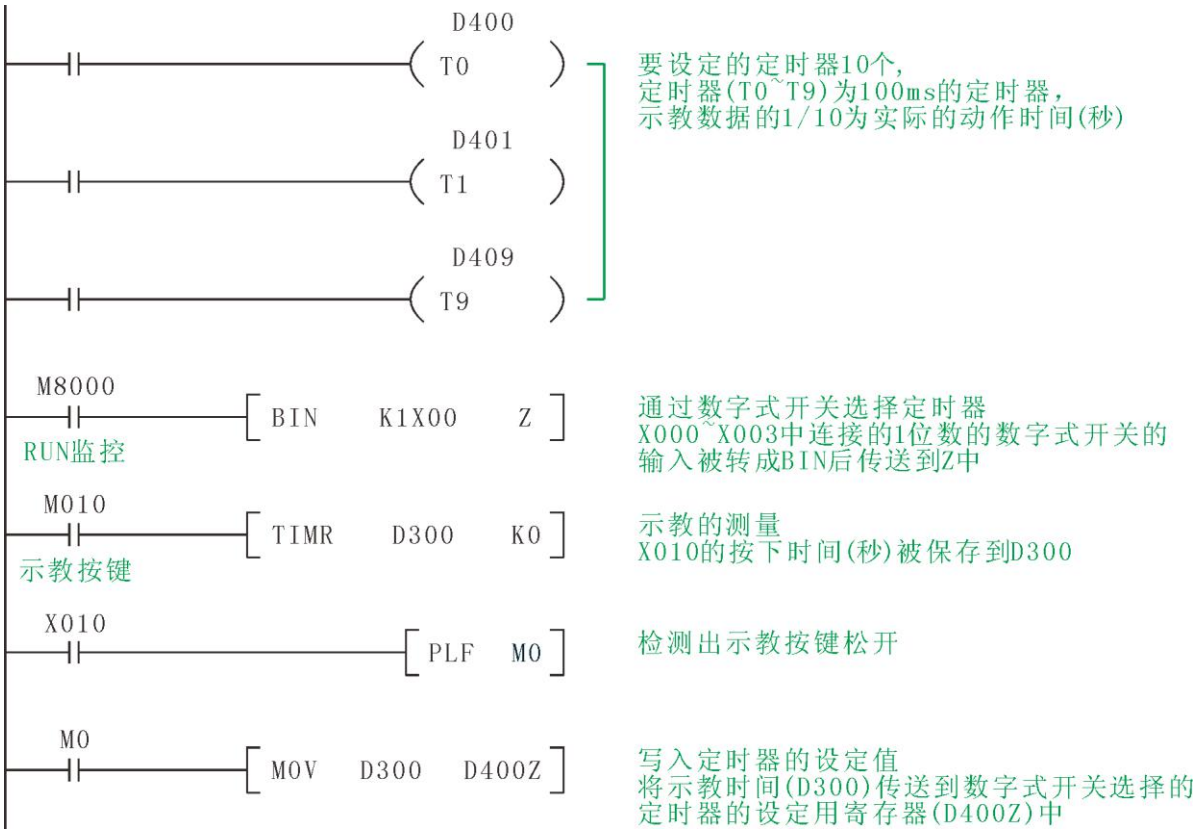
1、当指令触点为 OFF 时，按下时间的当前值 D.+1 被复位，示教时间 D.不改变

2、软元件的占用点数

以示教时间 D.中指定的软元件为起始，占用 2 点。请注意不要与机械控制中使用的软元件重复。D.示教时间；D.+1 按下时间的当前值。

➢**举例**

在 10 种数据寄存器中，写入示教时间。



FNC65 - STMR：特殊定时器

➢功能说明

用于轻松制作断开延迟定时器、单脉冲定时器、闪烁定时器的指令。

➢指令格式

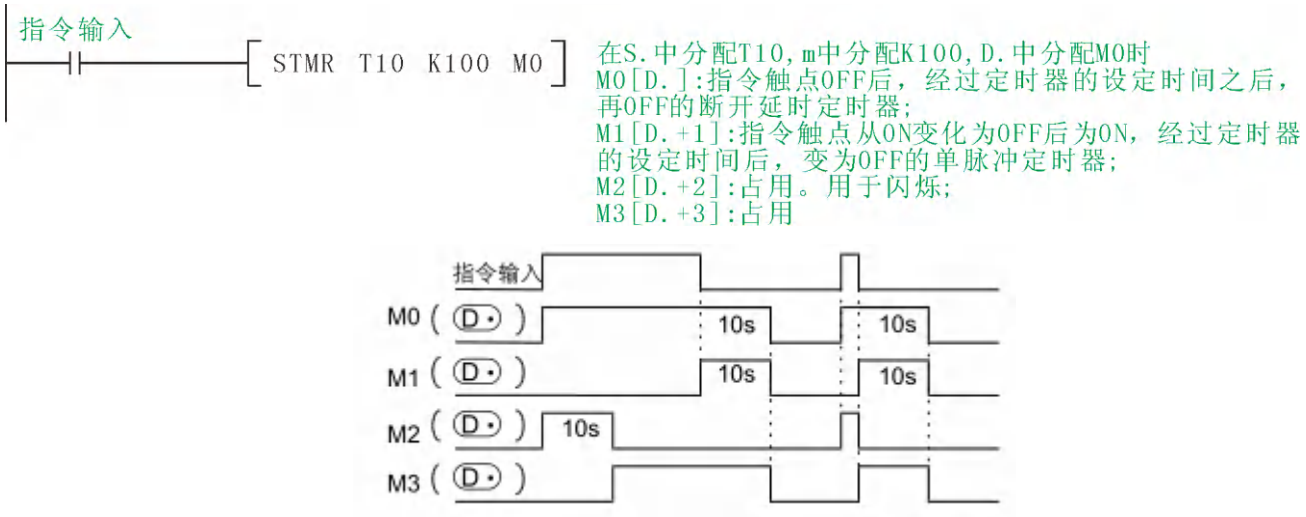
TTMR　S.　m　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	使用的定时器编号，T0~T199（100ms 定时器）		T		BIN16 位
m	定时器的设定值（1~32767）		D、R	K、H	BIN16 位
D.	被输出的起始编号	Y、M、S			位

➢动作说明

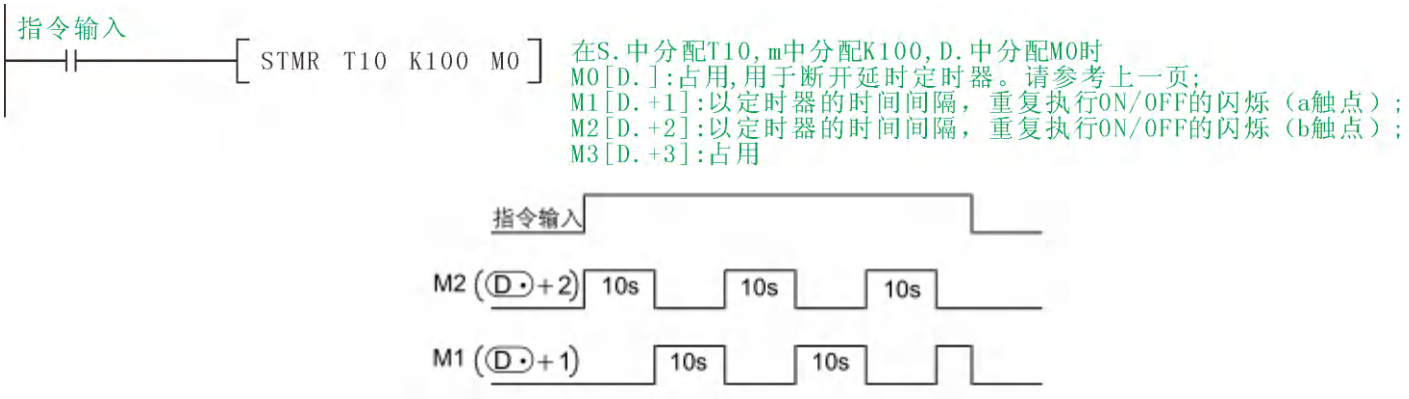
1、16 位运算(STMR)

将 m 指定的值作为 S.中指定的定时器的设定值，从 D.开始输出 4 点。



➢举例

用在 D.+3 的 b 触点断开该指令，通过编写这样的如下所示的程序，在 D.+1，D.+2 中输出闪烁。占用 D.，D.+3。



➢**注意要点**

1、指定定时器的使用

这个指令中所指定的定时器编号，不能在其他普通回路中（OUT 指令等）重复使用。

如果重复使用，该定时器将不能正常工作。

2、软元件的专用点数

以 D.中指定的软元件为起始占用 4 点。请注意不要与机器其他控制中使用的软元件重复。

软元件	断开延时定时器单脉冲定时器	功能
D.	断开延时定时器	占用
D.+1	单脉冲定时器	闪烁（a 触点）
D.+2	占用	闪烁（b 触点）
D.+3	占用	闪烁（c 触点）

3、指令触点 OFF 时

D.，D.+1，D.+3 在经过设定时间后变为 OFF。D.+2 和定时器 D.被即时复位。

FNC66 - ALT：交替输出

➢功能说明

输入为 ON 时，使位软元件反转（ON 至 OFF）用的指令。

➢指令格式

ATL【P】 D.

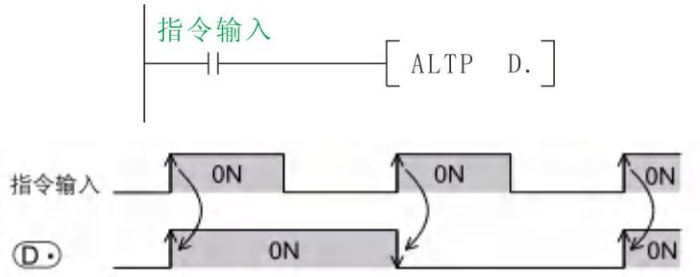
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	交替输出的位软元件编号	Y、M、S	--	K、H	BIN16 位

➢动作说明

1、16 位运算（ALT，ALTP）

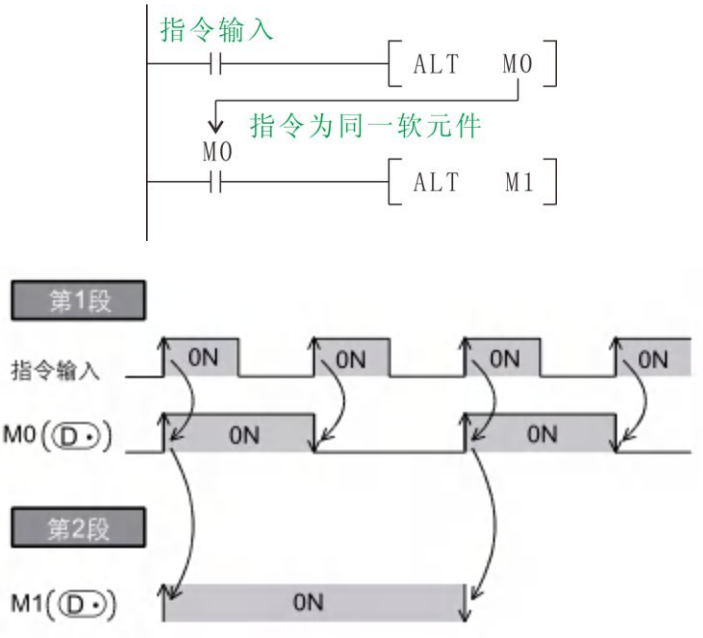
交替输出（1 段）

指令输入每次从 OFF 变为 ON 时，D.中指定的位软元件就执行 ON <->OFF 的反转。



分频输出【采用交替输出 2 段】

组合多个 ALTP 指令后，可以实现使用多个段的分频输出。



➢注意要点

1、使用 ALT 指令（连续执行型）时交替输出（1 段）

使用 ALT 指令编程时，每个扫描周期都执行反转动作。

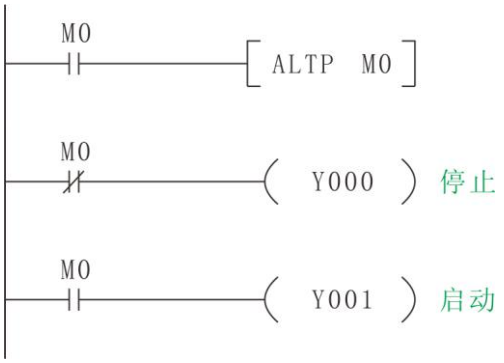
希望通过指令的 ON/OFF 使其反转动作时，请使用 ALTP 指令（脉冲执行型），或是 LDP 指令（脉冲执行型）触点等

➢举例

1、通过 1 个输入启动/停止

1）按下按键 X000 后，启动输出 Y001 的动作。

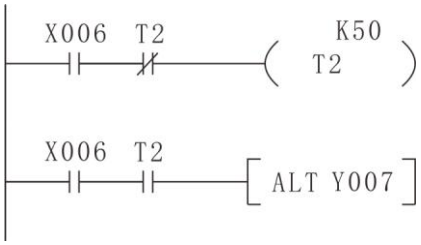
2）再次按下按键 X000 后，停止输出 Y000 的动作。

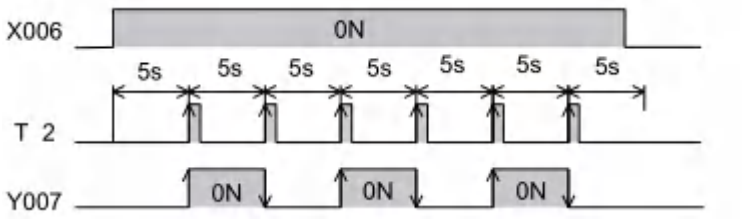


2、闪烁动作

1）按下 X006 为 ON 时，定时器 T2 的触点每隔 5 秒瞬间动作一次。

2）T2 的触点，每次 ON 时都使输出 Y007 交替 ON/OFF。



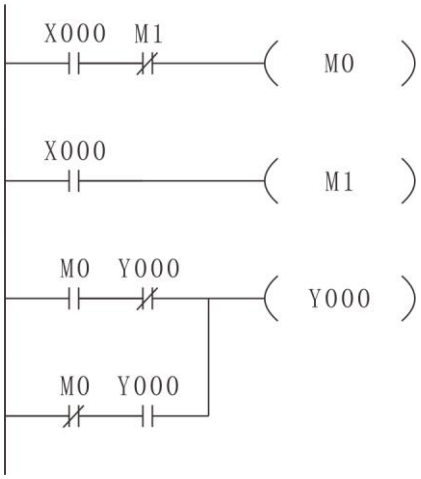


3、使用了辅助继电器(M)的交替输出动作（和 ALT 指令相同的动作）

在下面的梯形图中，是使用了基本指令以及辅助继电器(M)，实现与 ALT 指令相同交替动作的例子。

1）X001 为 ON 时，M0 仅维持 1 个扫描周期为 ON。

2）当 M0 初次为 ON 时，Y000 自保持，当第 2 次为 ON 时，解除自保持。



FNC67 - RAMP：斜坡信号

➢功能说明

在开始（初始值）和结束的 2 个值之间，按照指定的 n 次得到变化的数据指令。

➢指令格式

RAMP S1. S2. D. n

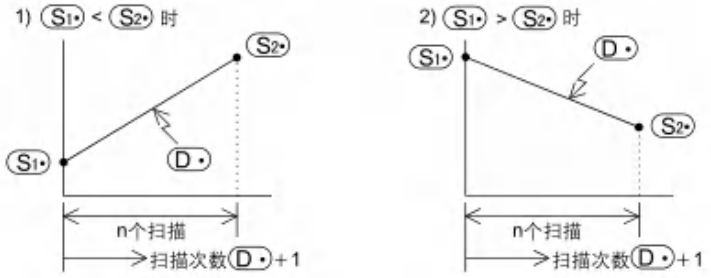
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存设定的斜坡初始值的软元件编号	--	D、R	--	BIN16 位
S2.	保存设定的斜坡目标值的软元件编号	--	D、R	--	BIN16 位
D.	保存设定的斜坡当前值的软元件编号	--	D、R	--	BIN16 位
n	斜坡的转移时间（扫描）	--	D、R	K、H	BIN16 位

➢动作说明

1、16 位运算（RAMP）

先指定开始的值 S1.和要结束的值 S2，当指令输入为 ON 后，在每个扫描周期中，将按照 n 中指定的次数进行等分的值依次加到 S1.中，然后将加得得值保存到 D.中。

将该指令和模拟量输出相结合，可以输出缓冲启动/停止指令。

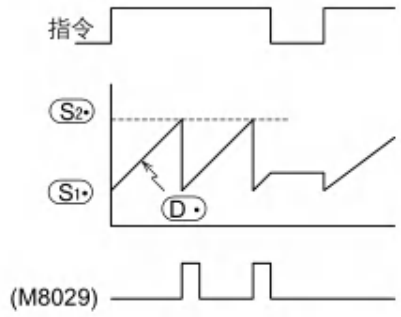


1）D.+1 中保存扫描次数（0->n 次）。

2）从开始到结束所需要时间。为运算周期*n 个扫描。

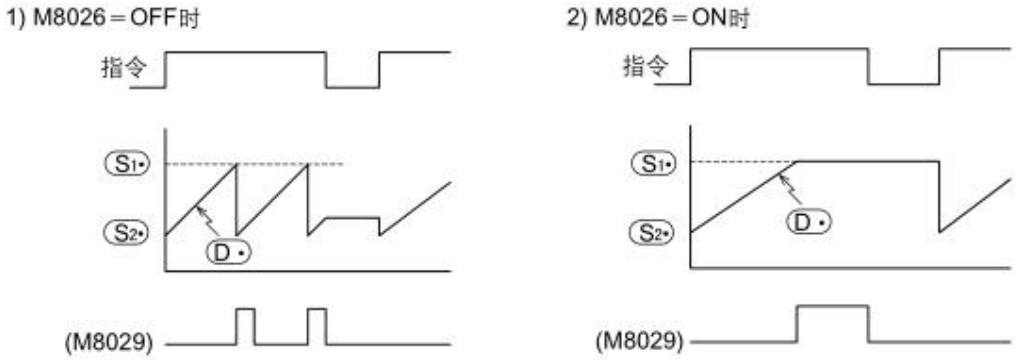
3）如果在动作过程中断开指令输入，则变为执行中断的状态。D.当前值保持，D.+1 扫描次数清除，再次置 ON 后，D.被清除，从 S1 开始重新动作。

4）转移结束后，指令执行结束标志位 M8029 动作，D.的值返回到 S1.的值。



2、模式标志位（M8026）的动作

根据模式标志位 M8026 的 ON/OFF 状态，D.+1 的内容也作如下变更：



➢相关软元件

操作数种类	名称	内容	备注
M8029	指令执行结束	n 个扫描周期后，D.＝S2.时置 ON	RUN —> STOP 时清零
M8026	RAMP 模式	请参考上面的模式标志位 M8026 的操作	

➢注意要点

1、D.中指定停电保持软元件时指令输入按原样置 ON，PLC 设置到 RUN 时，请先把 D.清零。

FNC68 - ROTC：旋转工作台控制

➢功能说明

为了取放旋转工作台上的物品，针对要求取放的窗口，就近旋转工作台，适用于此种用途的指令。

➢指令格式

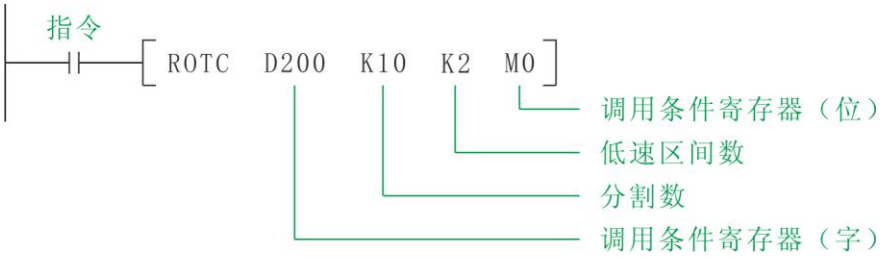
ROTC　S.　m1　m2　D.

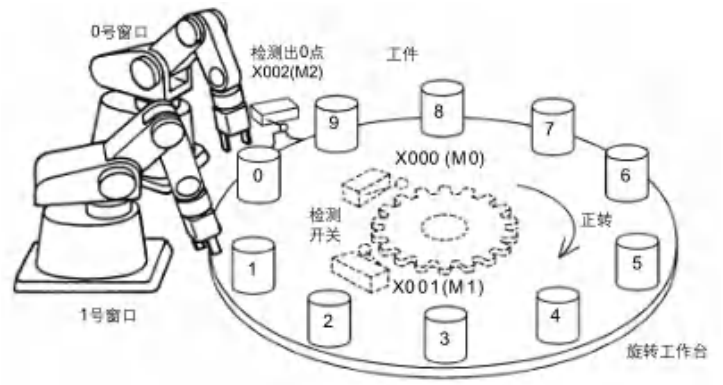
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存设定的斜坡初始值的软元件编号	--	D、R	--	BIN16 位
S2.	保存设定的斜坡目标值的软元件编号	--	D、R	--	BIN16 位
D.	保存设定的斜坡当前值的软元件编号	--	D、R	--	BIN16 位
n	斜坡的转移时间（扫描）	--	D、R	K、H	BIN16 位

➢动作说明

1、16 位运算（ROTC）

如下图所示，为了取放分割为 m1(=10)个的旋转工作台上的工件，按照 m2 和 S.，D.的条件就近旋转工作台。





1）调用条件的指定寄存器 S.

S.	计数器用寄存器	使用传送指令事先进行设定。
S.+1	调用窗口编号的设定	
S.+2	调用工件编号的设定	

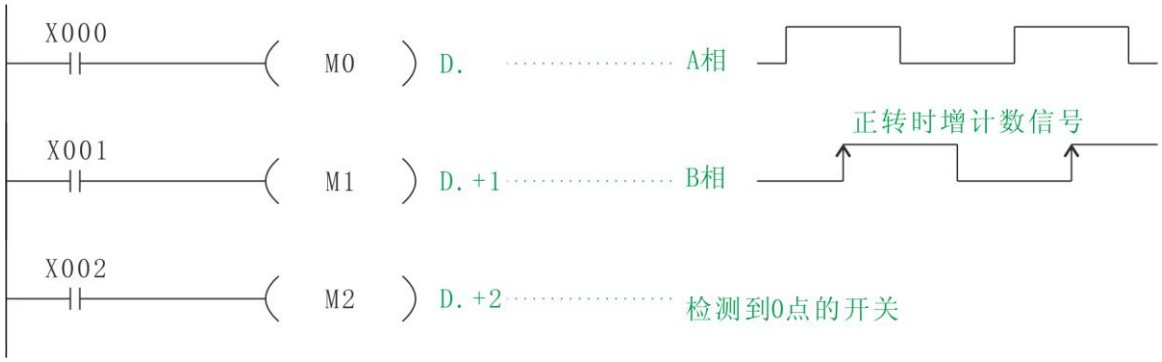
2）调用条件的指定位 D.

D.	A 相信号	预先构建重新由输入信号（X）驱动的内部触点回路。
D.+1	B 相信号	
D.+2	检测出 0 点的信号	
D.+3	高速正转	
D.+4	低速正转	
D.+5	停止	
D.+6	低速反转	
D.+7	高速反转	

动作条件

作为使用这个指令的必要条件，如下面的例子所示。

1）旋转检测信号：X—>D.。请设置用于检测工作台的正转/反转用的 2 相开关（X000，X001），以及 0 号窗口时会动作的开关 X002。



用 X000~X002 替换 D.~D.+2 的内部触点。

X 和 D.中指定的起始软元件编号为任意点。

2) 计数用寄存器的指定 S.。S.是用于对 0 号窗口中几号工件的到来进行计数的计数器。

3) 调用条件的指定寄存器 S.+1，S.+2。在 S.+1 中设定要调用的窗口编号；在 S.+2 中设定调用的工件编号。

4) 分割数 m1 和低速时间 m2，指定工作台的分割数 m1 和低速运转区域 m2。

指定以上的条件后，在指令的起始软元件 D.中指定的 D.+3~D.+7 输出中，可以获得正转/反转，高速/低速/停止等输出。

➢注意要点

1、根据指令输入的 ON/OFF 而动作

1）指令输入为 ON 后，驱动这个指令，会自动求出 D.+3~D.+7 的结果。

2）指令输入为 OFF 时，D.+3~D.+7 也变为 OFF。

2、旋转检测信号 D.~D.+2 的工作 1 分割区域内动作 10 次时，请将分割数设定，调用窗口的编号设定、工件编号的设定都设定为 10 倍的值。由此，低速区域的设定值可以设定为分割量的中间值等。

3、关于 0 点检测信号 D.。指令输入为 ON，0 点检测信号（M2）为 ON 时，计数用的寄存器 S.的内容被清零。需要预先执行这个清除操作后，再开始运行。

FNC69 - SORT：数据排序

➢功能说明

该指令是用于将数据（行）和群数据（列）构成的数据表格，以指定的群数据（列）为标准，按照行单位将数据表格重新升序排列。在这个指令中，群数据（列）被保存在连续的软元件中。此外，数据（行方向）被保存在连续的软元件中。还有便于增加数据（行），支持升序/降序排列 FNC149 - SORT2 指令。

➢指令格式

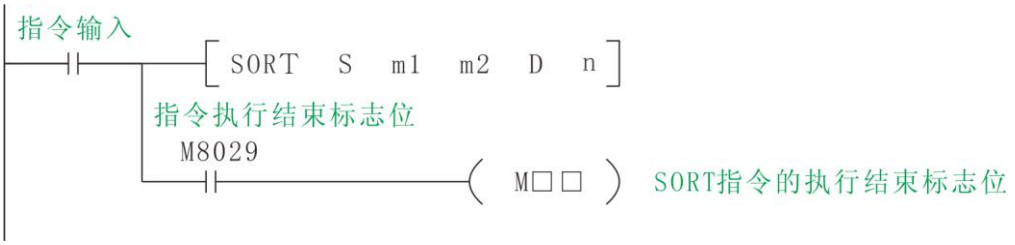
SORT S. m1 m2 D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据表格的软元件的起始编号【占用 m1*m2 点】	--	D、R		BIN16 位
m1	数据行数【1~32】	--		K、H	BIN16 位
m2	群数据列数【1~6】	--		K、H	BIN16 位
D	保存运算结果的软元件的起始编号【占用 m1*m2 点】	--	D、R	K、H	BIN16 位
n	作为排列标准的群数据（列）的列编号【m1~m2】		D、R	K、H	BIN16 位

➢动作说明

1、16 位运算（SORT）

在 S.开始的 m1*m2 点的数据表格（排序前）中，以 n 列的群数据为标准，按照升序重新排列数据行，然后保存在从 D.开始的 m1*m2 点的数据表格（排序后）中。



1）下面以排序前 m1=3、m2 =4 的情况为例来说明数据表格的结构。如是排序后的数据表格，请将 S.改读成 D.。

		群数 m2 个（m2=4 时）			
		1 管理编号	2 身高	3 体重	4 年龄
数据数 m1=3 的情况	1	S.	S.+3	S.+6	S.+9
	2	S.+1	S.+4	S.+7	S.+10
	3	S.+2	S.+5	S.+8	S.+11

2）下面以排序前 m1=3、m2 =4 的情况为例来说明数据表格的结构。如是排序后的数据表格，请将 S.改读成 D.。

		群数 m2 个（m2=4 时）			
		1 管理编号	2 身高	3 体重	4 年龄
数据数 m1=3 的情况	1	S.	S.+3	S.+6	S.+9
	2	S.+1	S.+4	S.+7	S.+10
	3	S.+2	S.+5	S.+8	S.+11

指令输入为 ON 时开始数据的排序，m1 个扫描后数据排序结束，指令执行结束标志位 M8029 为 ON。

➢注意要点

1、动作过程中，请勿改变操作数和数据的内容。

2、再次执行时，请将指令输入 OFF 一次。

3、指令的使用次数的限制，程序中仅可以使用 1 次。

4、S.和 D.中指定了同一软元件时，原来的数据被修改成排序后的数据顺序。到执行结束为止，请特别注意不要改变 S 的内容。

7-8 外围设备 I/O - FNC70~FNC79

FNC70 - KEY：数字键输入

FNC71 - HEY：16 进制输入

FNC72 - DSW：数字开关

FNC73 - SEGD：七段译码

➢功能说明

数据译码后，点亮 7 段数码管（1 位数）的指令。

➢指令格式

SEGD【P】 S. D.

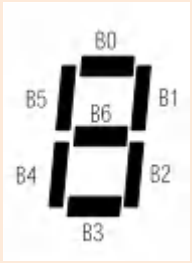
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	译码的起始字软元件	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
D.	保存7段码显示用数据的字软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位

➢动作说明

1、16 位运算（SEGD、SEGDP）

将 S.的低 4 位（1 位数）的 0~F(16 进制数)译码成 7 段码显示用的数据，并保存到 D.的低 8 位中。

2、7 段译码表

S.					7 段码的构成	D.								显示数据
16 进制数	b3	b2	b1	b0		B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0		0	0	1	1	1	1	1	1	
1	0	0	0	1		0	0	0	0	0	1	1	0	
2	0	0	1	0		0	1	0	1	1	0	1	1	
3	0	0	1	1		0	1	0	0	1	1	1	1	
4	0	1	0	0		0	1	1	0	0	1	1	0	
5	0	1	0	1		0	1	1	0	1	1	0	1	
6	0	1	1	0		0	1	1	1	1	1	0	1	
7	0	1	1	1		0	0	1	0	0	1	1	1	
8	1	0	0	0		0	1	1	1	1	1	1	1	
9	1	0	0	1		0	1	1	0	1	1	1	1	
A	1	0	1	0		0	1	1	1	0	1	1	1	
B	1	0	1	1		0	1	1	1	1	1	0	0	
C	1	1	0	0		0	0	1	1	1	0	0	1	
D	1	1	0	1		0	1	0	1	1	1	1	0	
E	1	1	1	0		0	1	1	1	1	0	0	1	
F	1	1	1	1		0	1	1	1	0	0	0	1	

➢注意要点

1、软元件的占用点数，软元件 D.的输出开始的低 8 位被占用，高 8 位不变化。

2、位软元件的起始，或是字软元件的最低位位 B0。

FNC74 - SEGL：七段码时分显示

FNC75 - ARWS：箭头开关

FNC76 - ASC：ASCII 数据输入

➢功能说明

将半角/英文数字字符串转换成 ASCII 的指令。

用于在外部显示器中选择显示多个消息。

➢指令格式

ASC　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	从计算机输入的 8 个字符的半角英文数字	--	--	字符串	字符串（仅 ASCII 码）
D.	保存 ASCII 数据的起始字软元件编号	--	T、C、D、R、V、Z	--	BIN16/32 位

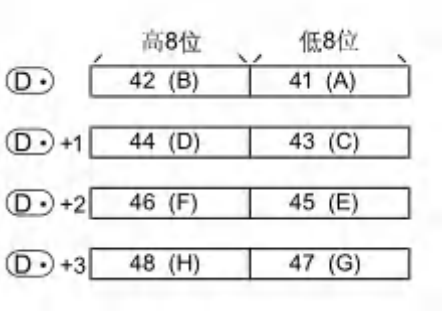
➢动作说明

1、16 位运算（ASC）

将 S.中指定的半角、英文、数字字符串转换成 ASCII 码后，依次传送到 D.中。

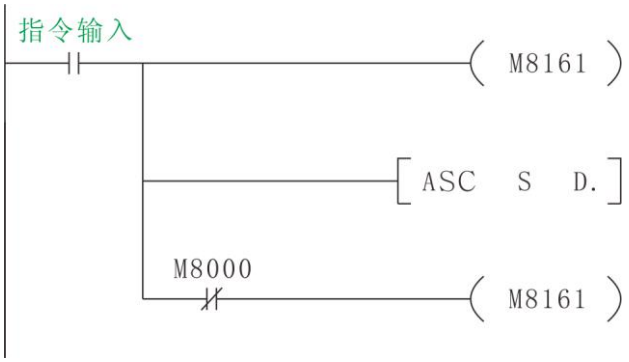
1）在 S.中处理 A~Z、0~9、符号的半角字符。（不处理全角字符串）。在用编程工具编程时，输入字符串。

2）转换后的 ASCII 码按照低 8 位、高 8 位的顺序，每 2 个字符/1 字节保存在 D.中。



扩展功能

M8161 置 ON 后，扩展功能变为有效，此时将 S.中指定的半角/英文数字字符串转换成 ASCI 码，然后将其依次传送到 D.的低 8 位（1 个字节）中，高 8 位为 H00。



	D.		S.
	高 8 位	低 8 位	字符串
D.	00	41	A
D.+1	00	42	B
D.+2	00	43	C
D.+3	00	44	D
D.+4	00	45	E
D.+5	00	46	F
D.+6	00	47	G
D.+7	00	48	H

➢相关软元件

软元件	名称	内容
M8161	扩展功能标志位	FNC76 - ASC，FNC80 - RS，FNC82 - ASCI，FNC83 - HEX，FNC84 - CCD 指令的 8 位处理模式。OFF 时每 2 个字符被按照低 8 位、高 8 位的顺序加以保存；ON 时每 1 个字符被保存在低 8 位中（1 个字符/1 个字）

➢注意要点

1、软元件的占用点数

1）扩展功能 OFF 时，D.占用字符串除以 2 点。（不能整除时，则将小数点进位）

2）扩展功能 ON 时，D.占用的点数和字符串占用的数目相同。

2、使用 FNC80 - RS，FNC82 - ASCI，FNC83 - HEX，FNC84 - CCD 等时扩展功能标志位 M8161 是与其他指令通用的标志位。

使用上述指令和 ASC 指令时，请注意，在 ASC 指令的前面编写 M8161 ON 或 OFF 的程序，不造成影响。

FNC77 - PR：ASCII 码打印

➢功能说明

该指令是将 ASCII 码的数据并行输出到 Y。

➢指令格式

PR　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 ASCII 码数据的软元件的起始编号	--	T、C、D、R	--	BIN16 位
D.	输出 ASCII 码数据的起始 Y 编号	Y	--	--	位

➢动作说明

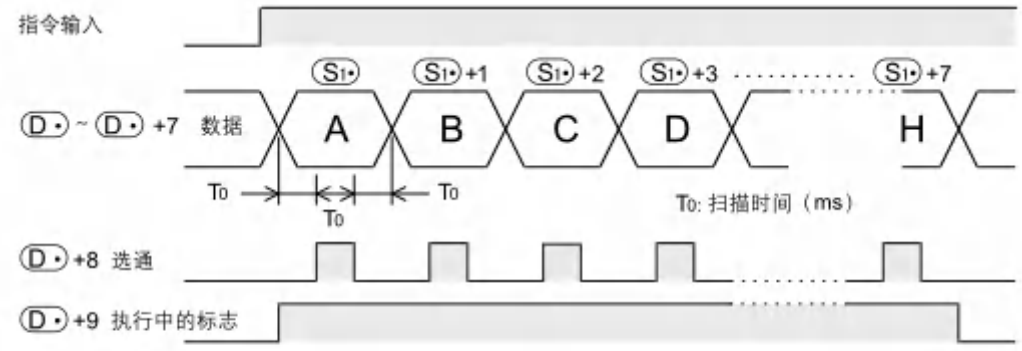
1、16 位运算（PR）

将 S.~S.+7 的低 8 位（1 个字节）中保存的 ASCII 码，按时分方式逐个字符输出到 D.~D.+7 中。

S. 中保存的 ASCII 码如下所示，下述时序图即以此为例进行说明。发送的顺序是从 S. ="A"开始，到 S.+7 = "H"为止结束，发送 8 个字节。

S.	S.+1	S.+2	S.+3	S.+4	S.+5	S.+6	S.+7
A（H41）	B（H42）	C（H43）	D（H44）	E（H45）	F（H46）	G（H47）	H（H48）

2、时序图



输出信号的种类

- D· ~ D· +7：发送输出　　D· 为低位侧，　　D· +7为高位侧。
- D· +8　　：选通信号
- D· +9　　：执行中的标志　按照上述的时序图动作。

➢相关软元件

软元件	名称	内容	RUN—>STOP 时清零
M8027	PR 模式	OFF 时 8 个字符串行输出（固定为 8 个字符串）；ON 时 16 个字符串输出（1~16 个字符）	

➢注意要点

1、指令输入以及指令的动作。指令输入等于 ON 时，即使是执行连续为 ON 的指令或执行脉冲指令，只要循坏一次的输出结束，则执行就结束。仅当 M8027 等于 ON 时 M8029 动作；指令输入等于 OFF 时，输出全部为 OFF。

2、与扫描时间（运算时间）的关系，该指令与扫描时间同步执行。扫描时间较短时，可以使用恒定扫描模式驱动；较长时可以使用定时器中断驱动。

3、关于 PLC 的输出，请使用晶体输出型的 PLC。

4、数据中存在 00H（NULL）时 M8027 等于 ON 时，指令执行结束，不输出剩余的数据。此外，M8029 维持 1 个扫描周期为 ON。

FNC78 - FROM：BFM 读出

➢功能说明

从第 m1 号从机模块缓冲储存器（BFM）的 m2 起始地址的储存区，开始读出 n 个数据放入从本机 D.开始的软元件中。

➢指令格式

【D】FROM　m1　　m2　　D.　　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
m1	特殊功能单元/模块的单元号（从基本的右侧开始依次为 K0~K7）	--	D、R	K、H	BIN16/32 位
m2	传送源缓冲存储区（BFM）编号	---	D、R	K、H	BIN16/32 位
D.	传送目标软元件编号	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
n	传送点数	--	D、R	K、H	BIN16/32 位

备注：

1) -1<=m1<=7；0<=m2<=1000；1<=n<=128；

2）m1 为-1 时代表对本机操作。

➢举例

FROM K2 K20 D5 K2

操作后：2 号从机的 BFM 储存器第 20、21 个数据，放入本机的 D5、D6 软元件。

FNC79 - TO：BFM 写入

➢功能说明

从本机 S.开始的软元件，读出 n 个数据。放入第 m1 号从机模块缓冲储存器（BFM）以 m2 起始地址开始的储存区。

➢指令格式

【D】TO　m1　　m2　　S.　　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
m1	特殊功能单元/模块的单元号（从基本的右侧开始依次为 K0~K7）	--	D、R	K、H	BIN16/32 位
m2	传送源缓冲存储区（BFM）编号	---	D、R	K、H	BIN16/32 位
D.	保存传送源数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
n	传送点数	--	D、R	K、H	BIN16/32 位

备注：

1）-1<=m1<=7；0<=m2<=1000；1<=n<=128；

2）m1 为-1 时代表对本机操作。

➢举例

TO　K2　K20　D5　K2

操作后：本机的 D5、D6 软元件数据,放入 2 号从机的 BFM 储存器第 20、21 个储存区。

7-9 外围设备 SER - FNC80~FNC89

FNC80 - RS: 串行数据传送

➢功能说明

通过安装在基本单元上的 RS-485 进行无协议通信，从而执行数据的发送和接收的指令。

➢指令格式

RS S. m D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存发送数据的数据寄存器的起始软元件。	--	D、R、修饰		BIN16 位/字符串
	发送数据的字节数，0≤m≤4096		D、R	K、H	BIN16 位
D.	数据接收结束时，保存接收数据的数据寄存器的起始软元件	---	D、R、修饰	--	BIN16 位/字符串
n	接收数据的字节数，0≤n≤4096	--	D、R	K、H	BIN16 位

➢动作说明

1、16 位运算（RS）

该指令是用于通过安装在基本单元上的 RS-485 串行通讯口进行无协议通信，从而执行数据的发送和接收的指令。

➢相关软元件

软元件	名称	软元件	名称
D8120※2	通信格式的设定	M8063	串行通信出错
D8122※3	发送数据的剩余点数	M8121※1	发送等待标志
D8123※3	接收点数的监控	M8122※1	发送请求
D8124	报头	M8123※1	接收结束标志
D8125	报尾	M8124	载波的检测标志
D8129※2	超时时间的设定	M8129	超时的判断标志
D8063	串行通信出错 1 的错误代码编号	M8161※3	8 位处理模式
D8405	通信参数的显示		
D8419	运行模式的显示		

※1、左边的情况清除，RUN —> STOP 时；RS 指令 OFF 时。
※2、停电保持
※3、RUN —> STOP 时清除

➢关于 FNC80 - RS 指令和 FNC87 - RS2 指令区别

项目	RS 指令	RS2 指令	备注
报头点数	最大 1 个字符（字节）	1~4 个字符（字节）	RS2 指令中，报头或报尾中最多可以指定 4 个字符（字节）。
报尾点数	最大 1 个字符（字节）	1~4 个字符（字节）	
附加和校验	请用用户程序对应	可以自动附加	RS2 指令中，可以在收发的数据上自动附加和校验。但是请务必在收发的通信帧中使用报尾。

➢注意要点

1）FNC80 - RS 指令，可以用于 RS485。
2）请勿同时驱动针对同一个通信口的多个 FNC80 - RS 指令或者 FNC87 - RS2 指令。
3）不可以对同一个通信口使用 FNC80 - RS 指令和 FNC87 - RS2 指令。

FNC81 - PRUN：8 进制位传送

FNC82 - ASCI：HEX 转 ASCII

➢功能说明

S.按照 16 进制，从低位开始共 n 位转化成 ASCII 码，转化后的 ASCII 码依次放入 D.。

➢指令格式

ASCI【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换的 HEX 的软元件的起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
D.	保存转换后的 ASCII 码的软元件的起始编号	---	KnY、KnM、KnS、T、C、D、R	--	字符串（仅 ASCII 码）
n	要转换的 HEX 码的字符数（位数）设定范围 1~256	--	D、R	K、H	BIN16 位

备注：

1）1<=n<=256；
2） M8161=0 时，为 16 位转化模式。M8161=1 时，为 8 位转化模式。

➢举例

操作前：（D0）=0x270F，（D5）=0，（D6）=0；
ASCI D0 D5 K4
操作后：（D0）=0x270F，（D5）=0x3732，（D6）=0x4630；

FNC83 - HEX：ASCII 转 HEX

➢功能说明

这个指令是将 ASCII 码转换成 HEX 的指令。

此外，也有将 ASCII 码转换成 BIN 数据的 FNC260 - DABIN 指令和 ASCII 码转成 2 进制浮点数的 FNC117 - EVAL 指令。

➢指令格式

HEX【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换的 ASCII 码的软元件的起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R	K、H	字符串（仅 ASCII 码）
D.	保存转换后的 HEX 的软元件的起始编号	---	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16/32 位
n	要转换的 ASCII 码的字符数（字节数）设定范围 1~256	--	D、R	K、H	BIN16 位

备注：请仅将 ASCII 码设定为“0”~“9”,“A”~“F”。

➢动作说明

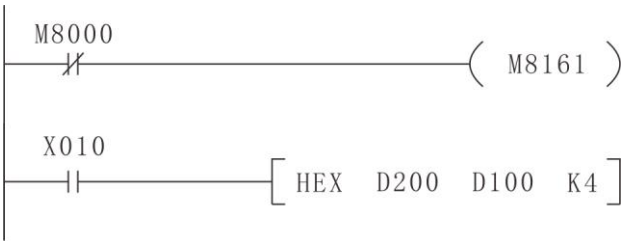
1、16 位运算（HEX/HEXP）

将 S.开始的软元件中保存的 ASCII 码的 n 个字符串转成 HEX 代码，然后保存到 D.开始的软元件中。在这个指令中，作为转换时使用的模式有 16 位模式和 8 位模式。有关其各自的动作，请参考下一页以后的内容。

2、M8161 等于 OFF 即 16 位转换模式时，M8161 为 RS、ASCI、CCD、CRC 指令通用。

将保存在 S.的高低各 8 位（字节）中的 ASCII 字符码转换成 HEX 数据，然后按照每 4 位数的方式传送到 D.中。用 n 指定要转换的字符数。M8161 在 RUN —>STOP 时被清零。

➢举例



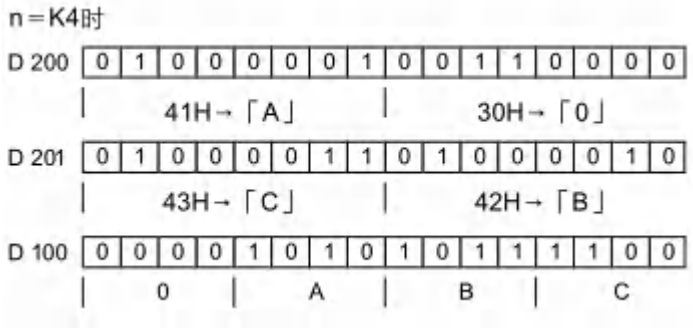
转换源数据

S.	ASCII 码	HEX 转换
D200 低	30H	0
D200 高	41H	A
D201 低	42H	B
D201 高	43H	C
D202 低	31H	1
D202 高	32H	2
D203 低	33H	3
D203 高	34H	4

D204 低	35H	5

指定字符数及转换结果【.】为 0。

S.	D102	D101	D100
n			
1	不变化	不变化	0H
2	不变化	不变化	0AH
3	不变化	不变化	0ABH
4	不变化	不变化	0ABCH
5	不变化	0H	ABC1H
6	不变化	0AH	BC12H
7	不变化	0ABH	C123H
8	不变化	0ABCH	1234H
9	0H	ABC1H	2345H



1）输入数据为 BCD 时，需要在执行该指令后，进行 BCD->BIN 转换。

2）在 HEX 指令中，被保存在 S.中的数据如不是 ASCII 码，则会出现运算出错，不能进行 HEX 转换。尤其是当 M8181 为 OFF 时，在 S.的高 8 位中也需要先保存 ASCII 码，请务必注意。

3、M8181 等于 ON 即 8 位转换模式时，M8161 为 RS、ASCI、CCD、CRC 指令通用。

并保存在 S.的低 8 位（字节）中的 ASCII 字符码转换成 HEX 数据，然后按照每 4 位数的方式传送到 D.中。用 n 指定要转换的字符数。M8161 在 RUN—>STOP 时被清零。

FNC84 - CCD：校验码

➢功能说明

M8161 为 0 时，以 S.指定的元件为起始的 n 点 8 位数据（16 位寄存器先低位后高位），将其总和与水平检验数据存储于 D.软元件与之后的一个软元件中。M8161 为 1 时只取低 8 位数据，其他操作与上述一致。

➢指令格式

CCD【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	对象软元件的起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R	--	BIN16/32 位
D.	保存计算出的数据的软件的起始编号	--	KnY、KnM、KnS、T、C、D、R	--	BIN16/32 位
n	传送点数	--	D、R	K、H	BIN16/32 位

备注：

1）1<=n<=256；

2）M8161=0 时，为 16 位转化模式。M8161=1 时，为 8 位转化模式。

➢举例

操作前：（M8161）=1，（D0）=37，（D1）=166，（D2）=77，（D10）=0，（D11）=206；
二进制每一位 1 的个数，为奇数时水平校验为 1，为偶数时水平校验为 0；
（D0） =37 -00100101
（D1） =166 -10100110
（D2） =77 -01001101
（D11）=206 -11001110
（D10）=（D0）+（D1）+（D2）
CCD D0 D10 K3
操作后：（M8161）=1，（D0）=37，（D1）=166，（D2）=77，（D10）=280，（D11）=206；

FNC87 - RS2：串行数据传送 2

➢功能说明

通过安装在基本单元上的 RS232 或者 RS-485 进行无协议通信，从而执行数据的发送和接收的指令。

➢指令格式

RS2 S. m D. n n1

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存发送数据的数据寄存器的起始软元件。	--	D、R、修饰		BIN16 位/字符串
	发送数据的字节数，0≤m≤4096		D、R	K、H	BIN16 位
D.	数据接收结束时，保存接收数据的数据寄存器的起始软元件	---	D、R、修饰	--	BIN16 位/字符串
n	接收数据的字节数，0≤n≤4096	--	D、R	K、H	BIN16 位
n1	使用通道号（K0 时 RS485，K1 时 RS232）	--	--	K、H	BIN16 位

➢动作说明

1、16 位运算（RS）

该指令是用于通过安装在基本单元上的 RS-485 串行通讯口进行无协议通信，从而执行数据的发送和接收的指令。

➢相关软元件

软元件		说明	软元件		说明
RS485	**RS232**		**RS485**	**RS232**	
D8400	D8420	通信格式的设定	M8401	M8421	发送等待标志※1
D8402	D8422	发送数据的剩余点数	M8402	M8422	发送请求※1
D8403	D8423	接收点数的监控※1	M8403	M8423	接收结束标志※1
D8405	D8425	通信参数的设定	M8404	M8424	载波的检测标志
D8409	D8429	超时时间的设定	M8405	M8425	数据设定准备就绪（DSR）标志位
D8410	D8430	报头 1,2	M8409	M8429	超时的判断标志
D8411	D8431	报头 3,4	M8063	M84238	串行通信出错
D8412	D8432	报尾 1,2			
D8413	D8433	报尾 3,4			
D8414	D8434	接收和校验（接收数据）			
D8415	D8435	接收和校验（计算结果）			
D8416	D846	发送和校验			
D8419	D849	运行模式的显示			
D8063	D8438	串行通信出错的错误代码编号			

※1、RUN —> STOP 时清除

➢关于 **FNC80 - RS 指令和 FNC87 - RS2 指令区别**

项目	RS 指令	RS2 指令	备注
报头点数	最大 1 个字符（字节）	1~4 个字符（字节）	RS2 指令中，报头或报尾中最多可以指定 4 个字符（字节）。
报尾点数	最大 1 个字符（字节）	1~4 个字符（字节）	
附加和校验	请用用户程序对应	可以自动附加	RS2 指令中，可以在收发的数据上自动附加和校验。但是请务必在收发的通信帧中使用报尾。

➢注意要点

1）FNC87 - RS 指令，可以用于 RS485 和 RS232。
2）请勿同时驱动针对同一个通信口的多个 FNC80 - RS 指令或者 FNC87 - RS2 指令。
3）不可以对同一个通信口使用 FNC80 - RS 指令和 FNC87 - RS2 指令。

FNC88 - PID：PID 运算

➢功能说明

用于进行 PID 控制的 PID 运算程序。达到采样时间的 PID 指令在其后的扫描时进行 PID 运算。参数中 S1.为设定目标值 SV，S2.为测量值 PV，S3.~S3.+24 控制参数，D.为输出值 MV。

➢指令格式

PID S1. S2. S3. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存目标值（SV）的数据寄存器编号	--	D、R	--	BIN16 位
S2.	保存测量值（PV）的数据寄存器编号	--	D、R	--	BIN16 位
S3.	保存参数的数据寄存器编号		D、R		BIN16 位
D.	保存参数的数据寄存器编号		D、R		BIN16 位
n	保存输出值（MV）的数据寄存器编号	--	D、R	K、H	BIN16 位

寄存器	功能	取值范围	
S3.	采样时间（Ts）	1~32767(ms)(但比运算周期短的时间无法执行)	
S3.+1	动作方向（ACT）	Bit0 0：正动作 1：逆动作 Bit1 0:关闭输入变化量报警，1：打开 Bit2 0：关闭输出变化量报警，1：打开 Bit3 不可用 Bit4 0：自整定不动作，1：执行自整定 Bit5 0：关闭输出上下限，1：打开 Bit6~bit15 不可用	
S3.+2	输入滤波常数（α）	0~99[%]，0 时没有输入滤波	
S3.+3	比例增益（Kp）	1~32767[%]	
S3.+4	积分时间（TI）	0~32767(*100ms) 0 时作为∞处理	
S3.+5	微分增益（KD）	0~100[%] 0 时无积分增益	
S3.+6	微分时间（TD）	0~32767(*10ms) 0 时无微分处理	
S3.+7	PID 运算的内部处理占用		
…			
S3.+19			
S3.+20	输入变化量（增侧）报警设定值	0~32767（S3+1 的 bit1=1 时有效）	
S3.+21	输入变化量（减侧）报警设定值	0~32767（S3+1 的 bit1=1 时有效）	
S3.+22	输出变化量（增侧）报警设定值。另外，输出上限设定值	0~32767（S3+1 的 bit2=1 bit5=0 时有效） -32767~32767(S3+1 的 bit2=0 bit5=1 时有效)	
S3.+23	输出变化量（减侧）报警设定值。另外，输出下限设定值	0~32767（S3+1 的 bit2=1 bit5=0 时有效） -32767~32767(S3+1 的 bit2=0 bit5=1 时有效)	
S3.+24	报警输出	Bit0 输入变化量上溢出 Bit1 输入变化量下溢出 Bit2 输出变化量上溢出 Bit3 输出变化量下溢出	S3+1 的 bit1=1 或 bit2=1 时有效

备注：

1）需要占有自 S3.起始的 25 个数据寄存器；
2）若指定电池保持寄存器时，在可编程控制器 RUN 时，务必清除保持内容；

➢举例

操作前：目标值 SV(D0)，测定值 PV（D1），参数（D100），输出值 MV（D150）；
PID D0 D1 D100 D150

7-10 数据传送 2 - FNC100~FNC109

FNC102 - ZPUSH 变址寄存器的成批保存

FNC103 - ZPOP 变址寄存器的恢复

7-11 浮点数指令 - FNC110~FNC139

FNC110 - ECMP：2 进制浮点比数

➢功能说明

将 S1.与 S2.内的二进制浮点值进行比较，并将比较结果送到 D.及其后两个软元件上。

➢指令格式

DECMP【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存要比较的 2 进制浮点数数据的软元件编号	--	D、R	K、H	实数（2 进制）
S2.	保存要比较的 2 进制浮点数数据的软元件编号	--	D、R	K、H	
D.	输出结果的起始位软元件编号（占用 3 点）	Y、M、S	D、R	--	位

备注：
1）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；
2）D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用。

➢举例

DECMP D0 D7 M3
操作后：当 float (D1，D0)> float（D8，D7）时，（M3）=1，（M4）=0，（M5）=0；
当 float (D1，D0)= float（D8，D7）时，（M3）=0，（M4）=1，（M5）=0；
当 float (D1，D0)< float（D8，D7）时，（M3）=0，（M4）=0，（M5）=1；

FNC111 - EZCP：2 进制浮点数区间比较

➢功能说明

将 S.与 S1.和 S2.的二进制浮点值比较，并将比较结果送到 D.及其后两个软元件上。

➢指令格式

DEZCP【P】 S1. S2. S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存要比较的 2 进制浮点数数据的软元件编号	--	D、R	K、H、Z	实数（2 进制）
S2.	保存要比较的 2 进制浮点数数据的软元件编号	--	D、R	K、H、Z	
S.	保存要比较的 2 进制浮点数数据的软元件编号		D、R	K、H、Z	
D.	输出结果的起始位软元件编号（占用 3 点）	Y、M、S	--	--	位

备注：

1）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；

2）源 S1.的内容不得大于源 S2.的内容，当源 S1.的内容大于源 S2.的内容时，系统默认把源 S2.的内容当成源 S1.的内容进行计算；

3）D.占有 3 点位软元件；

4）S1.<=S2.；

➢举例

DEZCP K100 K120 D7 M2

操作后：当 K100> float（D8，D7）时，（M2）=1，（M3）=0，（M4）=0；

当 K100<= float（D8，D7）<=K120 时，（M2）=0，（M3）=1，（M4）=0；

当 K120< float（D8，D7）时，（M2）=0，（M3）=0，（M4）=1；

FNC112 - EMOV：2 进制浮点数据传送

➢功能说明

传送 2 进制浮点数数据的指令。

➢指令格式

DEMOV【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	传送源的 2 进制浮点数数据，或是保存数据的软元件编号	--	D、R	实数 E	实数（2 进制）
D.	保存 2 进制浮点数数据的软元件编号	--	D、R	--	

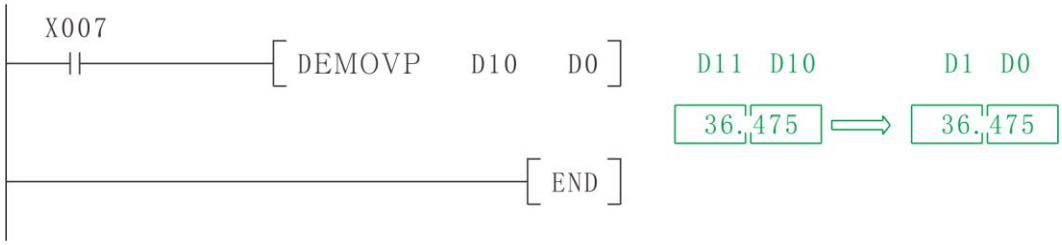
➢动作说明

1、32 位运算（DEMOV/DEMOVP）

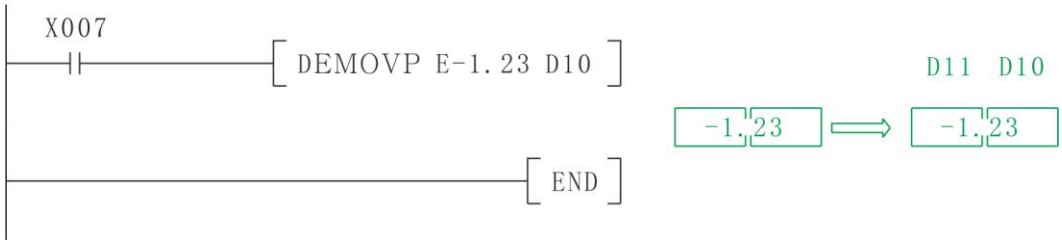
将传送源 S.+1，S.的内容（2 进制浮点数据）传送到 D.+1，D.中。此外，还可以在 S.中直接指定实数 E。

➢举例

1、X007 为 ON 时，将 D11、D10 的实数保存到 D1、D0 中的程序。



2、X007 为 ON 时，将实数-1.23 保存到 D1、D0 中的程序。



FNC116 - ESTR：2 进制浮点数转字符串

➢功能说明

该指令是用于将 2 进制浮点数数据转换成定位数的字符串（ASCII 码）。
还有可以将 BIN 数据转换成字符串（ASCII 码）的 FNC200 - STR 指令。

➢指令格式

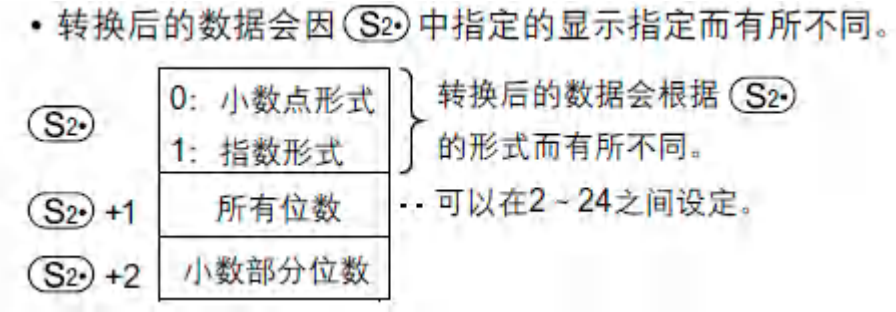
DESTR【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	要转换的 2 进制浮点数数据，或是保存数据的软元件的起始编号	--	D、R	实数 E	实数（2 进制）
S2.	保存要转换数值的显示指定的软元件起始编号		KnX、KnY、KnM、KnS、T、C、D、R		BIN16 位
D.	保存转换后字符串的软元件起始编号	--	KnY、KnM、KnS、T、C、D、R	--	字符串

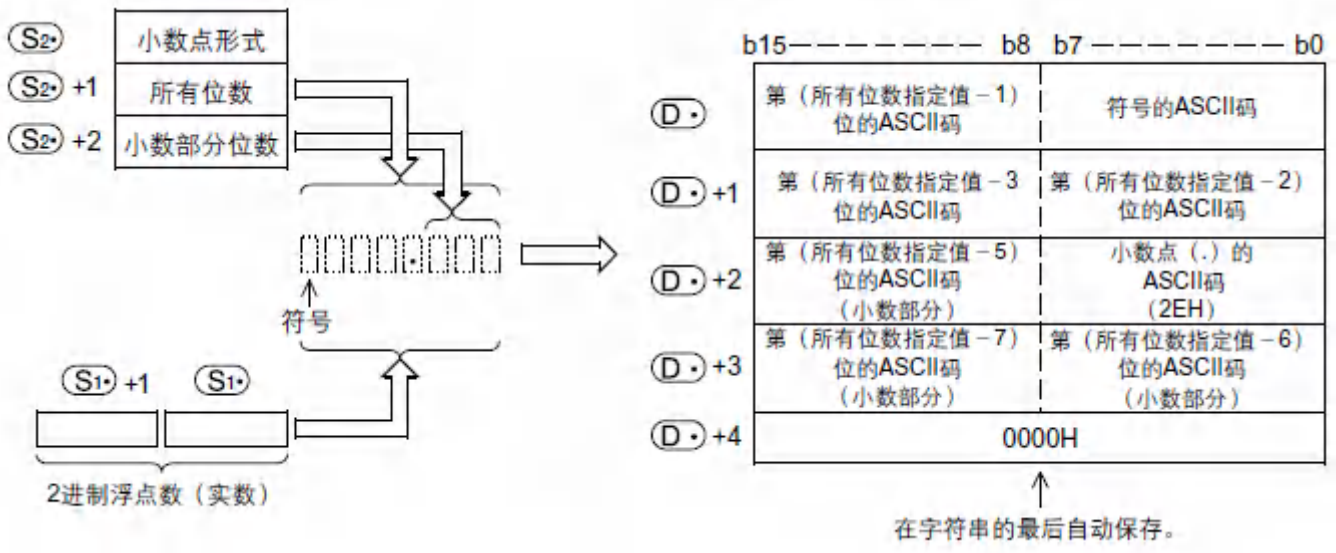
➢动作说明

1、32 位运算（DESTR/DESTRP）

根据 S2.,S2.+1,S2.+2 中指定的内容，将[S1.+1,S1.]的内容（2 进制浮点数数据）转换成字符串，并保存至 D.开始的软元件中。此外，还可以在 S1.中直接指定实数（E）。



2、小数点形式的情况



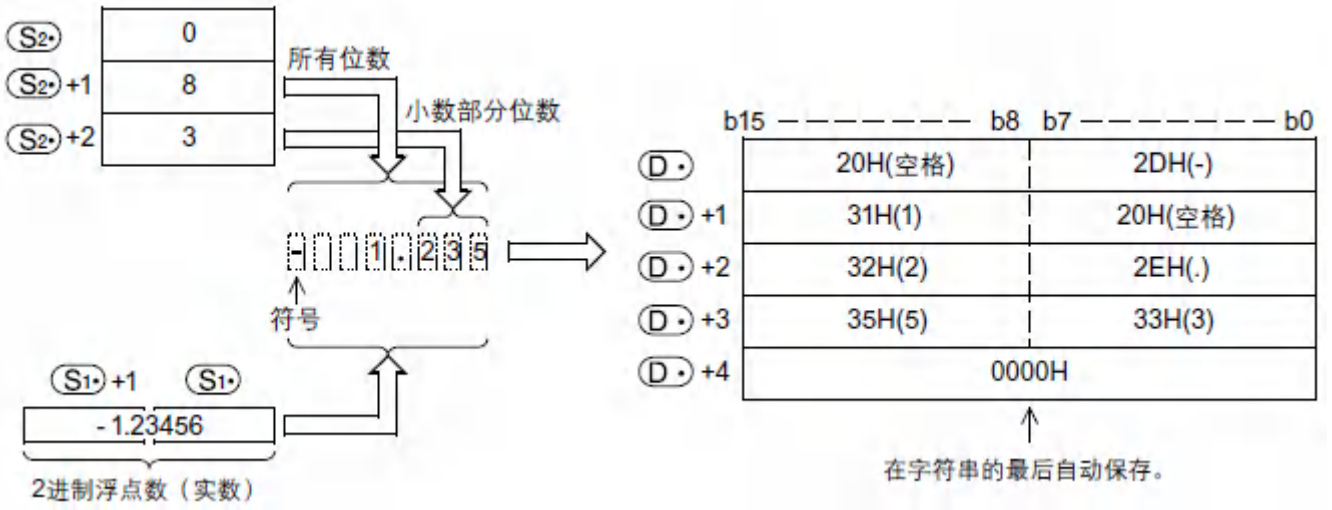
1）在[S2.+1]中可以指定的所有位数如下。（最大：24 位数）

小数部分的位数为“0”时所有位数≥2

小数部分的位数为“0”以外的数字时所有位数≥（小数部分位数+3）

2）在[S2.+2]中可以指定的小数部分位数为 0~7 位数。但是，请设定为小数部分位数≤（所有位数-3）。

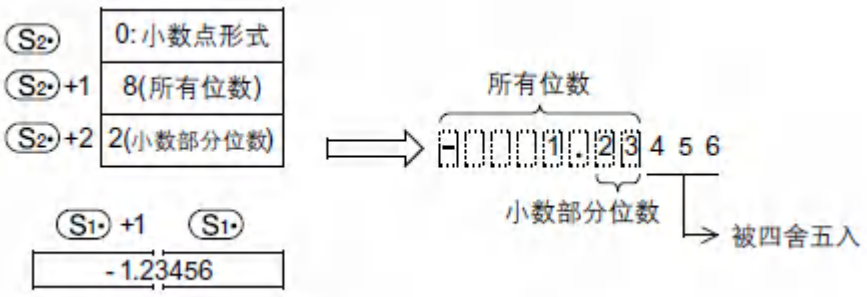
例如，所有位数为 8，小数部分位数为 3，指定-1.23456 时，[D.]开始的软元件中会如下背保存。



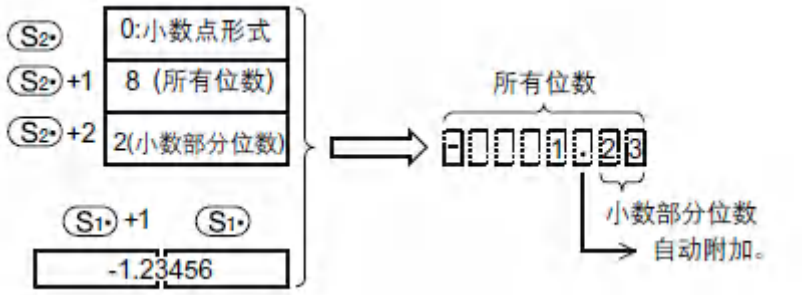
1）转换后字符串数据，如下所示被保存在[D.]以后的软元件中。

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

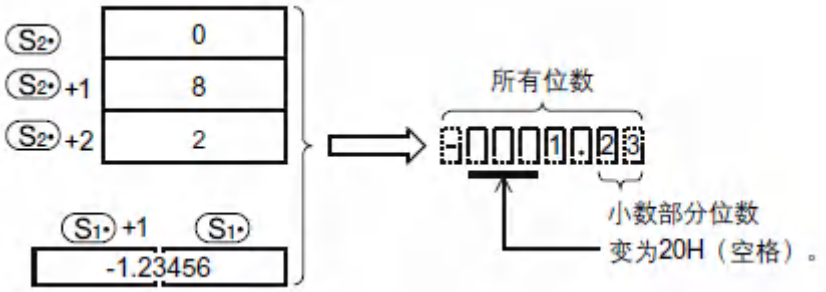
小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五入。



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

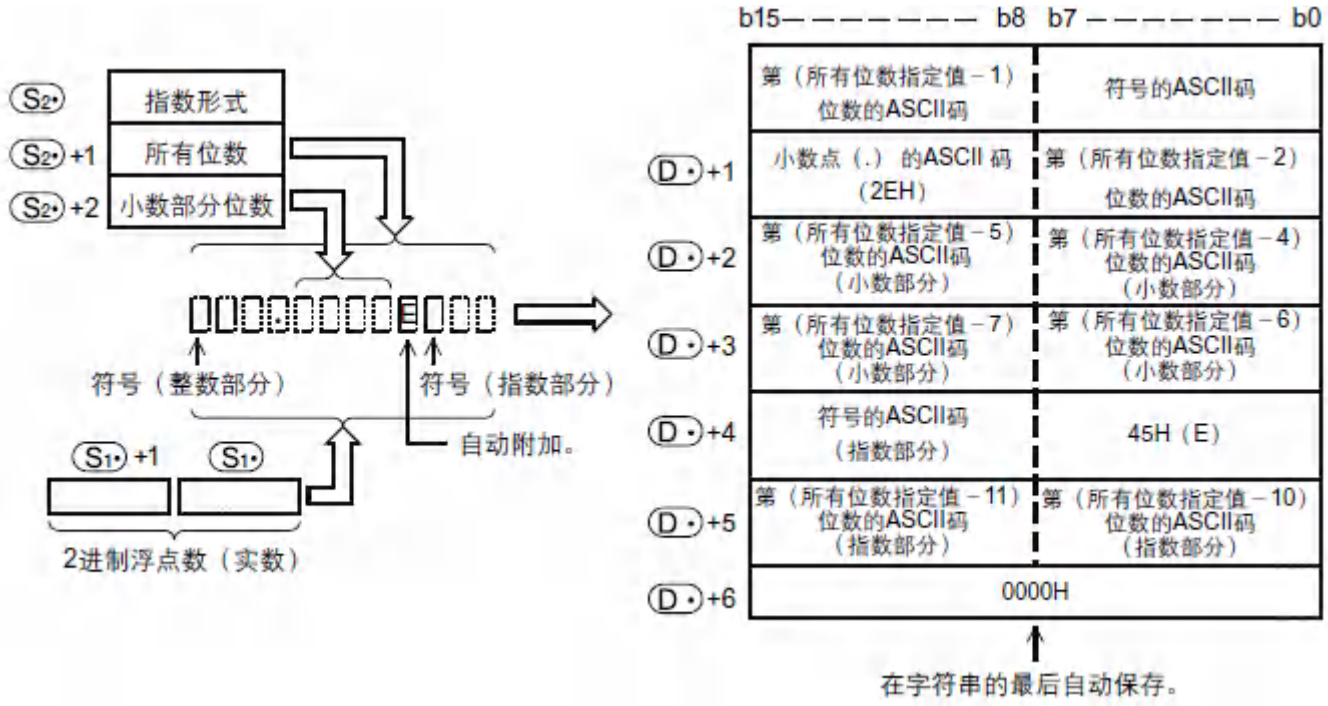


当从所有位数中去除符号、小数点、小数部分的为位数后，其位数比 2 进制浮点数数据的整数部分大的时候，在符号和整数部分之间保存“20H”（空格）。



在转换后的字符串最后，自动保存“00H”

3、指数形式的情况



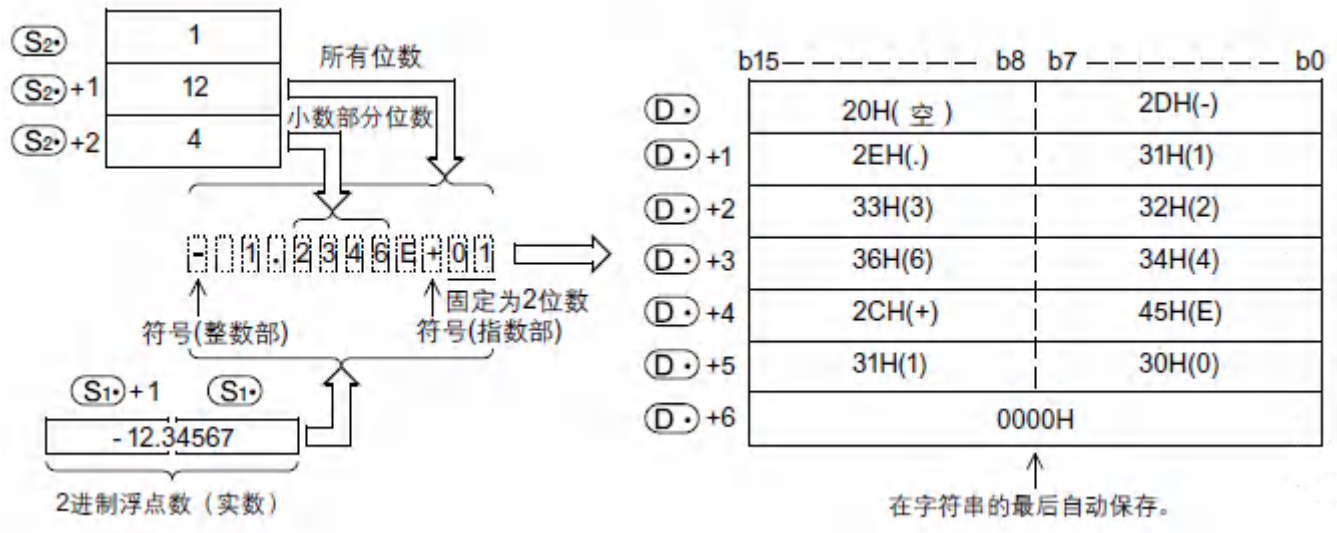
1）[S2.+1]中可以指定的所有位数如下。（最大：24 位数）

小数部分的位数为“0”时位数≥6

小数部分的位数为“0”以外数字时位数≥（小数部分位数+7）

2）[S2.+2]中可以指定的小数部分为 0~7 位数。但是请设定为小数部分位数≤（所有位数-7）

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

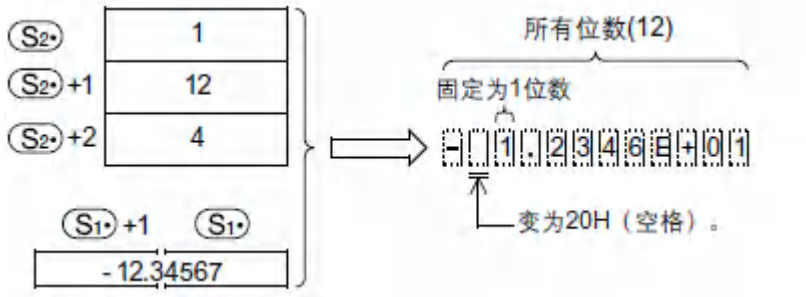


3）换后的字符串数据，如下所示被保存在[D.]开始的软元件中。

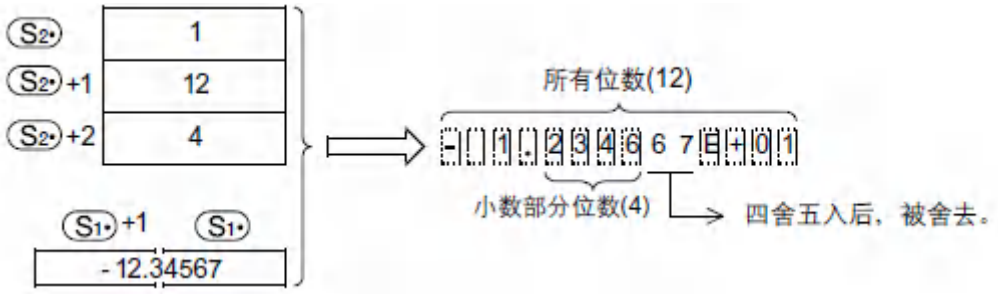
在整数部分的符号中，2 进制浮点数数据为正时保存“20H”（空格），为负时保存“2DH”(-)。

整数部分固定为 1 位数

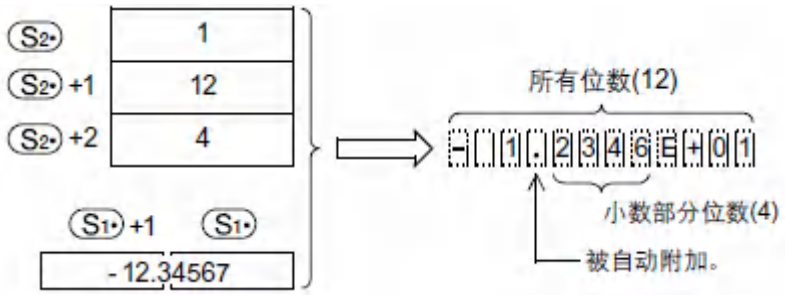
在整数部分和符号之间保存“20H”（空格）



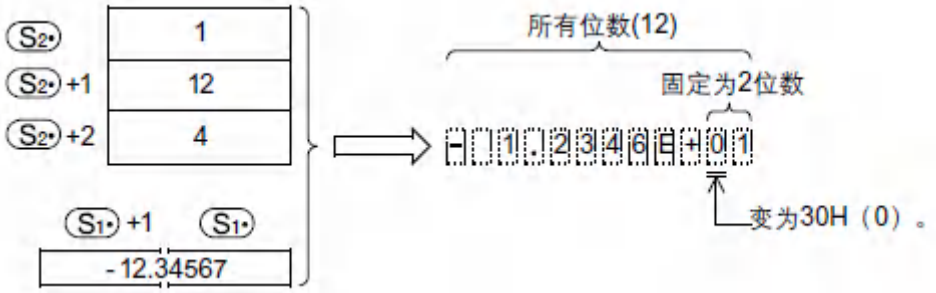
在小数部分位数的范围中不能容纳 2 进制浮点数数据的小数部分时，小数低位部分被四舍五如入。



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



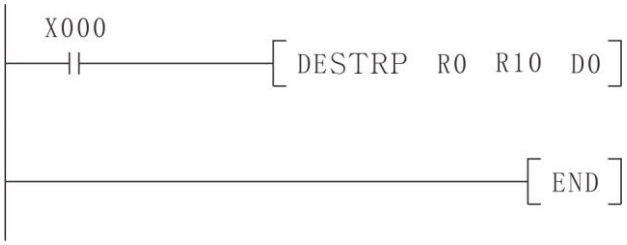
在指数部分的符号中，指数为正时保存“2BH”(+)，指数为负时保存“2DH”(-)。
指数部分固定为 2 位数。
当指数部分为 1 位数时，在其与指数部分的符号之间中保存“30H”(0)。

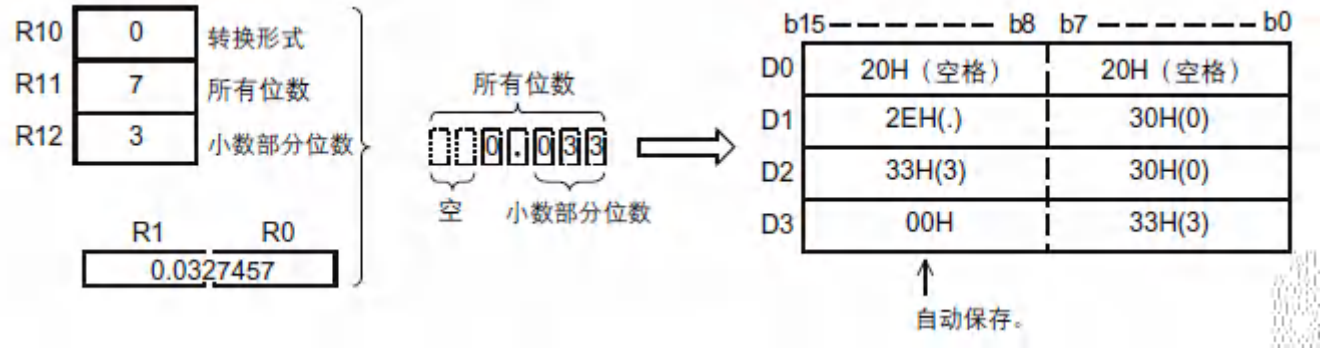


- 在转换的字符串最后，自动保存“00H”。

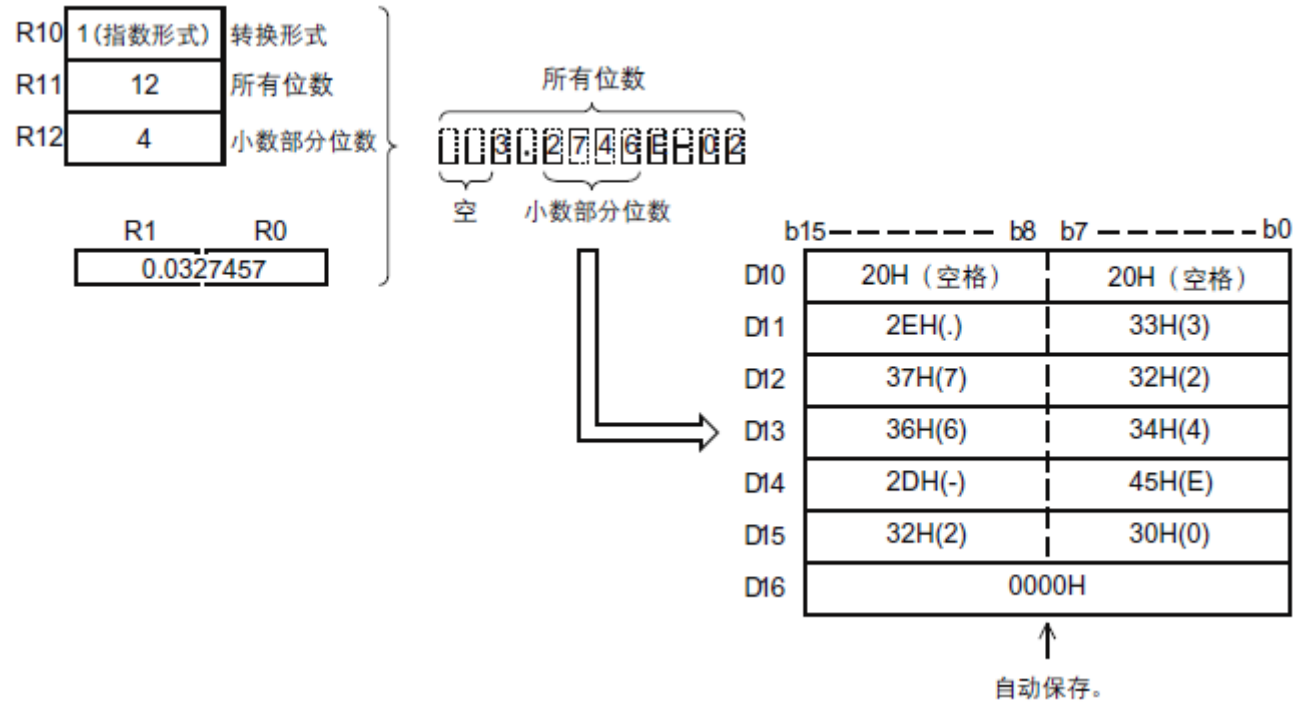
➢举例

1、X000 为 ON 时，根据 R10~R12 中指定的内容，将 R0，R1 的内容（2 进制浮点数数据）做转换，并保存在 D0 以后的软元件中的程序。





2、X000 为 ON 时，根据 R10~R12 中指定的内容，将 R0，R1 的内容（2 进制浮点数数据）做转换，并保存在 D10 以后的软元件中的程序。



FNC117 - EVAL：字符串转 2 进制浮点数

➢功能说明

将[S.]为起始软元件的字符串（ASCII 码）转换成 2 进制浮点数数据存储到[D.]中

➢指令格式

DEVAL【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S2.	保存要转换成 2 进制浮点数数据的字符串数据的软元件的起始编号		KnX、KnY、KnM、KnS、T、C、D、R	--	BIN16 位
D.	保存已转换的 2 进制浮点数数据的软元件的起始编号	--	D、R	--	字符串

备注：

1）整数部分、小数部分中不能存在“30H”（0）~“39H”（9）；

2）[D.]指定的字符串中不能存在 2 个以上的“2EH”（.）；

3）指数木方不能存在“45H”（E），“2CH”（+），“2DH”（-）以外的字符；

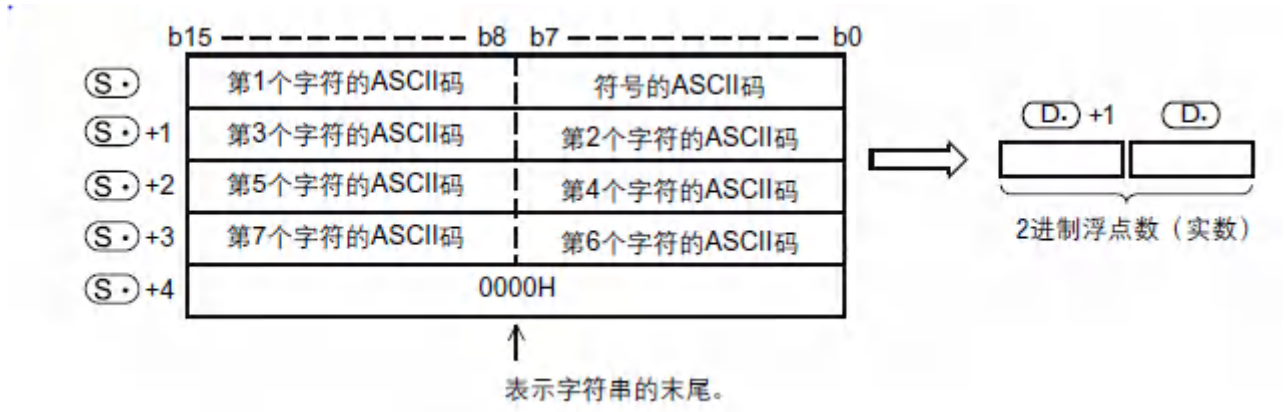
4）[S.]开始的相应软元件范围内应有“00H”作结束符；

5）[S.]以后的字符数不能为零，不能超出 24 个字符；

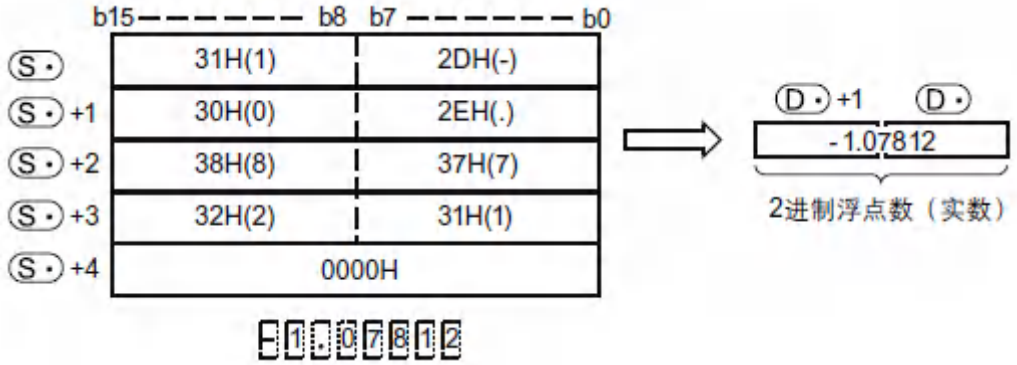
➢动作说明

1. 32 位运算（DEVAL/DEVALP）

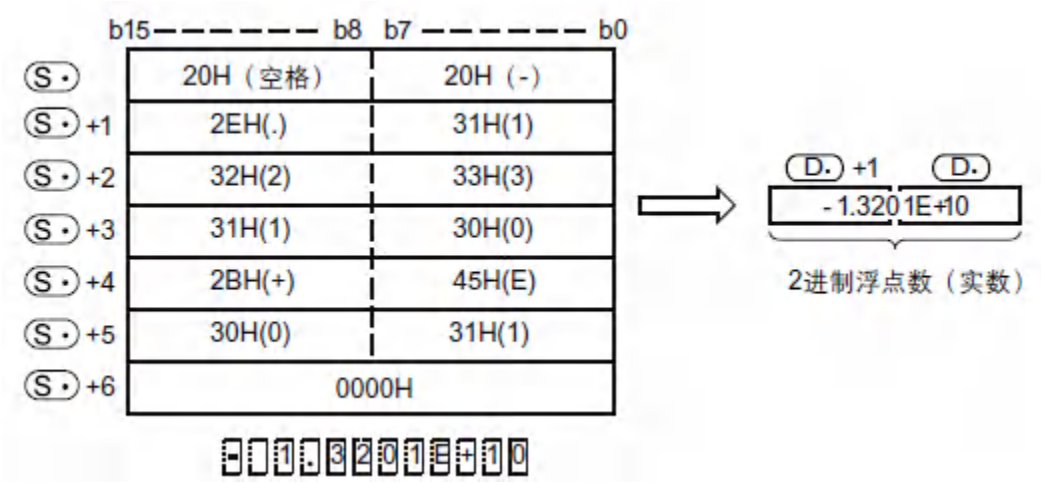
将[S.]开始的软元件中保存的字符串转换成 2 进制浮点数后，保存到[D.+1，D.]中。



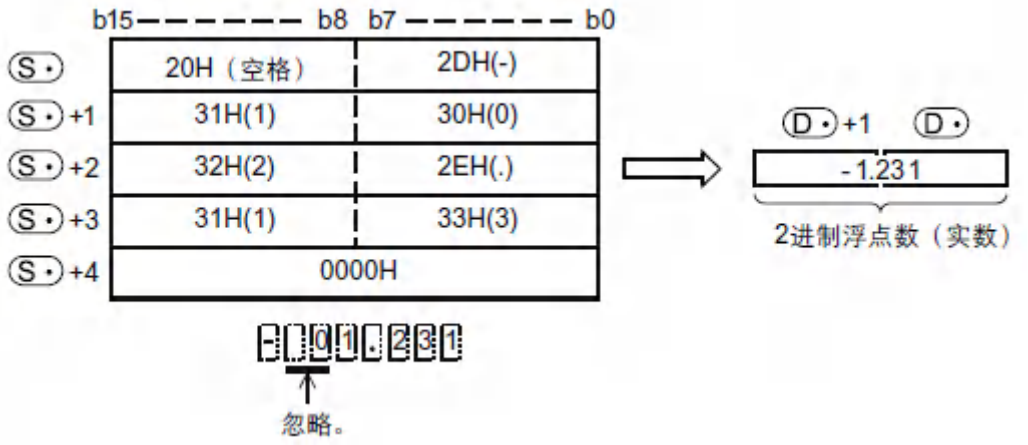
1）小数点形式时



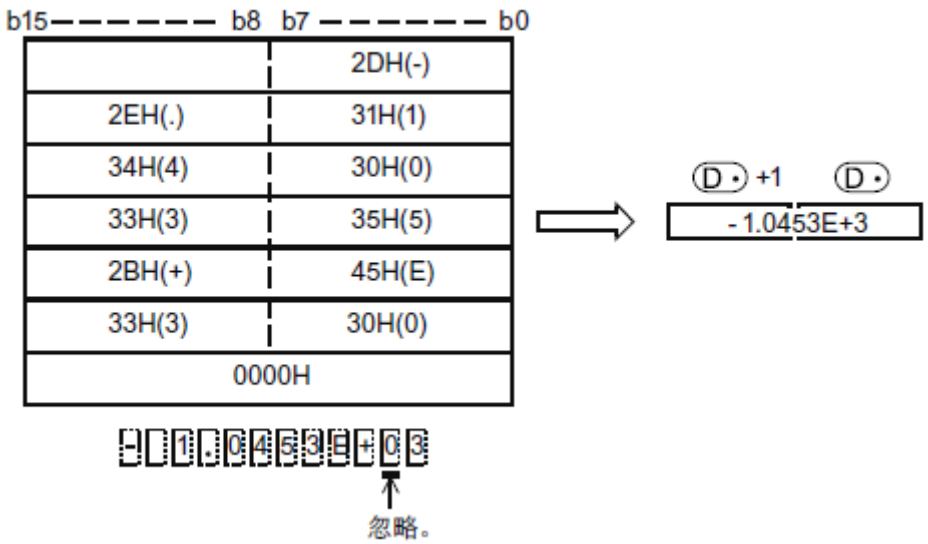
2）指数形式时



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



在指数形式的字符串中，“E”和数值之间如果存在“20H”（空格）或是“30H”（0），则忽略“30H”而进行转换。会忽略“20H”，“30H”而进行转换。字符串最大可以设定到 24 个字符。

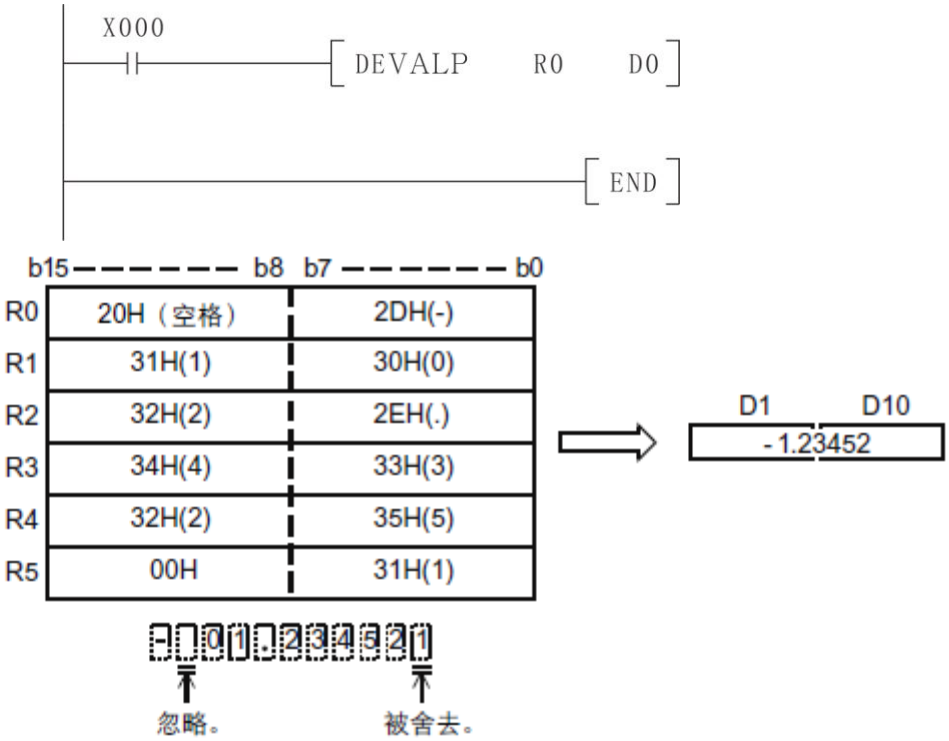


相关特殊继电器

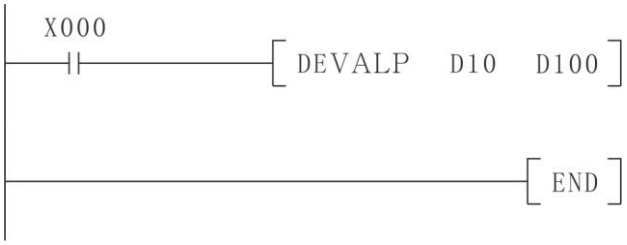
位元件	名称	内容	
		条件	动作
M8020	零位	转换结果真的为零（位数部分为“0”时）	零位标志位（M8020）为 ON
M8021	借位	转换结果的绝对值 $<2^{-126}$	[D.]的值小于 32 位实数的最小值（2^{-126}）部分被舍去，借位标志位（M8021）为 ON
M8022	进位	转换结果的绝对值 $\geq 2^{128}$	[D.]的值大于 32 位实数的最小值（2^{128}）部分被舍去，进位标志位（M8022）为 ON

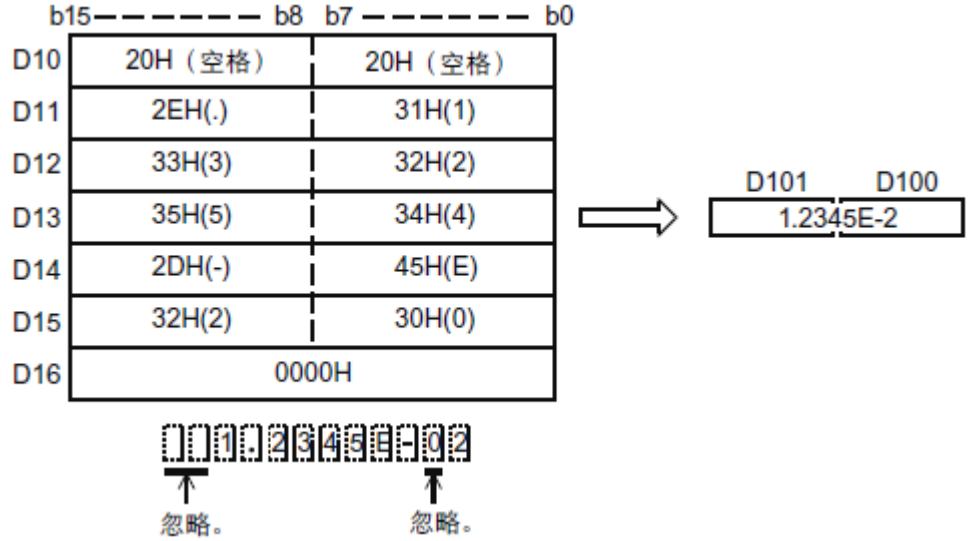
➢**举例**

1）当 X000 为 NO 时，将 R0 开始的软元件中保存的字符串转换成 2 进制浮点数，并保存到 D0，D1 中的程序。



2）当 X000 为 NO 时，将 D10 开始的软元件中保存的字符串转换成 2 进制浮点数，并保存到 D100，D101 中的程序





上溢出，下溢出，零时的动作

条件	动作
转换结果真的为零（位数部分为“0”时）	零位标志位（M8020）为 ON
转换结果的绝对值 $<2^{-126}$	[D.]的值小于 32 位实数的最小值（2^{-126}）部分被舍去，借位标志位（M8021）为 ON
转换结果的绝对值 $\geq 2^{128}$	[D.]的值大于 32 位实数的最小值（2^{128}）部分被舍去，进位标志位（M8022）为 ON

FNC118 - EBCD： 2 进制浮点转十进制浮点数

➢功能说明

将 S.指定的软元件的二进制浮点值转化成十进制浮点值，存入 D.中及其后一个软元件中。

➢指令格式

DEBCD【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 2 进制浮点数数据的数据寄存器编号	--	D、R	--	实数（2 进制）
D.	保存被转换的 10 进制浮点数数据的数据寄存器编号	--	D、R	--	实数（10 进制）

备注：二进制浮点数值为不易判别的数值，可以通过将其转化成十进制浮点值得外部设备很容易的监控。

➢举例

操作前：bin float（D1，D0）=1120403456，（D6）= 0，（D7） = 0；
DEBCD D0 D6
操作后：bin float（D1，D0）=1120403456，（D6）= 1000，（D7） = -1；
(D7，D6)表示的是 1000*10^(-1),值也为 100.

FNC119 - EBIN：10 进制浮点数转 2 进制浮点数

➢功能说明

将 S. 及其后一个软元件指定的软元件的十进制浮点值转化成二进制浮点值，存入 D.中。

➢指令格式

DEBIN【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 10 进制浮点数数据的数据寄存器编号	--	D、R	--	实数（10 进制）
D.	保存被转换的 2 进制浮点数数据的数据寄存器编号	--	D、R	--	实数（2 进制）

备注：十进制浮点数 S=D0;（D1，D0）=D0 * 10^(D1);

➢举例

操作前：（D0）=1，（D1）=2，(D7，D6) =0，(D1，D0)表示的是 1*10^(2),值也为 100.
DEBIN D0 D6
操作后：（D0）=1，（D1）=2，bin float (D7，D6) =1120403456；

FNC120 - EADD：2 进制浮点数加法运算

➢功能说明

将 S1.与 S2.内的二进制浮点值相加后，作为二进制浮点值存入 D.中。

➢指令格式

DEADD【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存进行加法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	实数
S2.	保存进行加法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	
D.	保存进行加法运算后的 2 进制浮点数数据	--	D、R	--	

备注：

1）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；

2）源数据和目的地址也可以指定同一元件号，此时，如用连续执行指令，就会在每个运算周期均相加，因此，请注意。

➢举例

操作前：float（D1，D0）=5.250，float（D8，D7）=2.500，（D11，D10）=0；
DEADD D0 D7 D10
操作后：float（D1，D0）=5.250，float（D8，D7）=2.500，float（D11，D10）=7.750；

FNC121 - ESUB：2 进制浮点数减法运算

➢功能说明

将 S1.指定的软元件的二进制浮点值减去 S2.指定的软元件的二进制浮点值，并将结果作为二进制浮点值存入 D.中。

➢指令格式

DESUB【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存进行减法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	实数（2 进制）
S2.	保存进行减法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	
D.	保存进行减法运算后的 2 进制浮点数数据	--	D、R	--	

备注：

1）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；

2）源数据和目的地址也可以指定同一元件号，此时，如用连续执行指令，就会在每个运算周期均运行该指令，因此，请注意。

➢举例

操作前：float（D1，D0）=5.750，float（D8，D7）=2.500，（D11，D10）=0；

DESUB D0 D7 D10

操作后：float（D1，D0）=5.750，float（D8，D7）=2.500，float（D11，D10）=3.250；

FNC122 - EMUL：2 进制浮点数乘法运算

➢功能说明

将 S1.与 S2.内的二进制浮点值的积作为二进制浮点值存入 D.中。

➢指令格式

DEMUL【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存进行乘法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	实数（2 进制）
S2.	保存进行乘法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	
D.	保存运算结果的数据寄存器编号	--	D、R	--	

备注：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）；

➢举例

操作前：float（D1，D0）=2.500，float（D8，D7）=2.500，（D11，D10）=0；

DEMUL D0 D7 D10

操作后：float（D1，D0）=2.500，float（D8，D7）=2.500，float（D11，D10）=6.250；

FNC123 - EDIV：2 进制浮点数除法运算

➢功能说明

将 S1.指定的软元件的二进制浮点值除以 S2.指定的软元件的二进制浮点值，并将结果作为二进制浮点值存入 D.中。

➢指令格式

DEDIV【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存进行除法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	实数（2 进制）
S2.	保存进行除法运算的 2 进制浮点数数据的字软元件编号	--	D、R	K、H、E	
D.	保存运算结果的数据寄存器编号	--	D、R	--	

备注：

1）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；

2）除数 S2.为 0 时，则置位运算错误标志位 M8067，指令不能执行。

➢举例

操作前：float（D1，D0）=6.250，float（D8，D7）=2.500，（D11，D10）=0；

DEDIV D0 D7 D10

操作后：float（D1，D0）=6.250，float（D8，D7）=2.500，float（D11，D10）=2.50；

FNC124 - EXP：2 进制浮点数指数运算

➢功能说明

将[S.]为起始软元件作为指数，以 e（2.71828）为底的指数运算指令，将运算结果 2 进制浮点数数据存储到[D.]中

➢指令格式

DEXP【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存执行指数运算的 2 进制浮点数数据的软元件起始编号		D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件起始编号	--	D、R	--	

➢动作说明

1、32 位运算（DLOGE/DLOGEP）

以[S.+1，S.+2]为指数做运算，将运算结果保存到[D.+1，D.]中。此外，也可以在[S.]中直接指定实数。

在指数运算中，将底（e）作为“2.71828”进行运算。

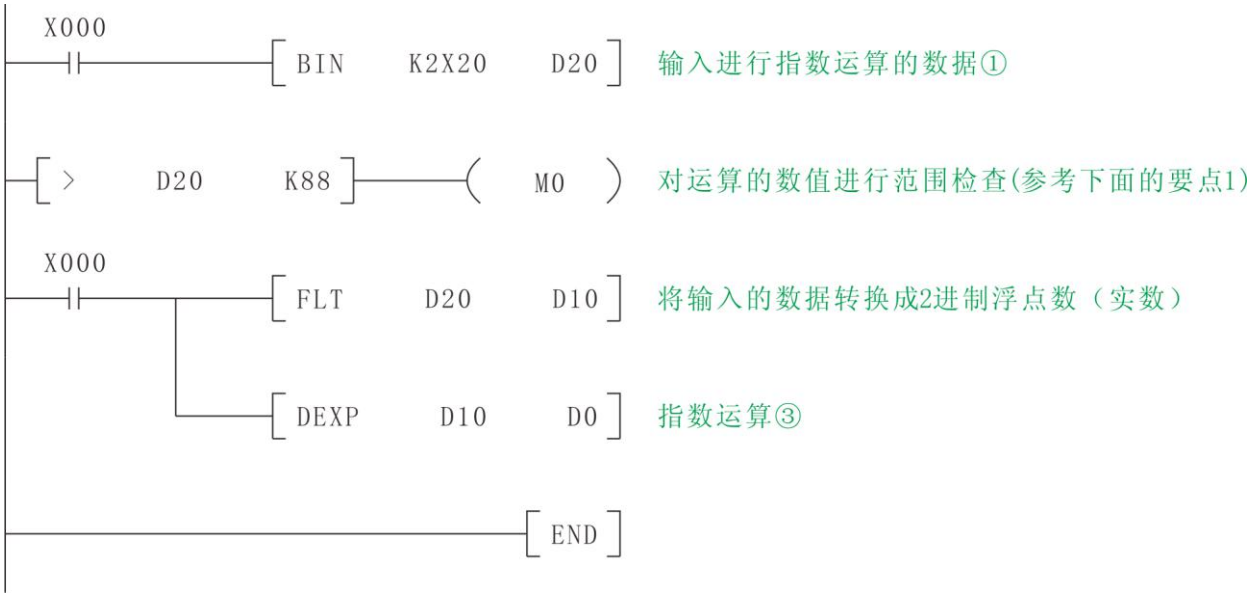
➢出错

下面的情况下会发生运算出错，出错标志位（M8067）为 ON，在 D8067 中保存错误代号。运算结果不在下面的范围内时。（错误代码：K6706）

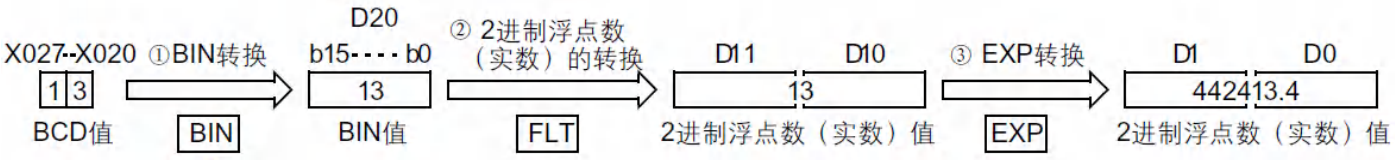
$$2^{-126} \leq |\text{运算结果}| < 2^{128}$$

➢举例

X000 为 ON 后，对 X20~X027 中以 BCD2 位数形式设定的数值进行指数运算，并且保存在 2 进制浮点数 D0，D1 中的程序。



在 X020~X027 中指定 l13 四的动作



要点

1）由于 $\log_e 2^{128} = 88.7$，因此当 X020~X027 的 BCD 值为 88 以下时，此时运算结果不满 2^{128}。设定了 89 以上的数值时，会发生运算出错，因此设定 89 以上的数值时，M0 置 ON，从而不执行运算

2）从自然对数向常用对数的转换

在 CPU 中，执行自然对数的运算。

要求出常用对数值时，请在[S.+1，S.]中指定用 0.4342945 分割常用对数的值。

$$10^{X} = e^{\frac{X}{0.4342945}}$$

FNC125 - LOGE：2 进制浮点数自然对数运算

➢功能说明

以 e（2.71828）为底的[S.+1，S.]的自然对数运算指令，将运算结果数据存储到[D.]中

➢指令格式

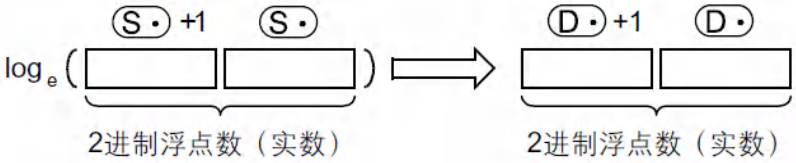
DLOGE【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存进行自然对数运算的 2 进制浮点数数据的软元件起始编号		D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件起始编号	--	D、R	--	

➢动作说明

1、32 位运算（DESTR/DESTRP）

执行[S.+1，S.+2]的自然对数[以 e（2.17828）为底时的对数]运算，并将运算结果保存到[D.+1,D.]中。此外，可以在[S.]中直接指定实数



在[S.+1,S.]中指定的值，只可以设定正值。（负值不能运算）

出错

下面的情况下会发生运算出错，出错标志位（M8067）为 ON，在 D8067 中保存错误代号。

S.中指定的值为负时，错误代码：K6706；

S.中指定的值为 0 时，错误代码：K6706；

➢举例

X000 为 ON 后，求出 D50 中设定的“10”的自然对数，并保存到 D30，D31 中的程序。

FNC126 - LOGE10：2 进制浮点数常用对数运算

➢功能说明

以 10 为底的[S.+1，S.]的常用对数运算指令，将运算结果数据存储到[D.]中

➢指令格式

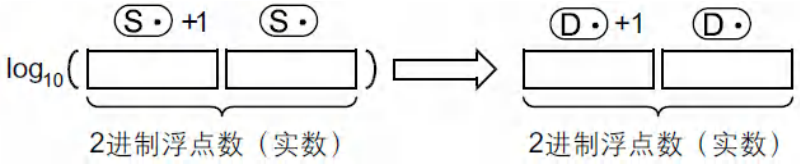
DLOG10【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存进行常用对数运算的 2 进制浮点数数据的软元件起始编号		D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件起始编号	--	D、R	--	

➢功能和动作说明

1、32 位运算（DLOG10/DLOG10P）

执行[S.+1，S.+2]的常用对数[以 10 为底时的对数]运算，并将运算结果保存到 [D.+1,D.]中。此外，可以在[S.]中直接指定实数。



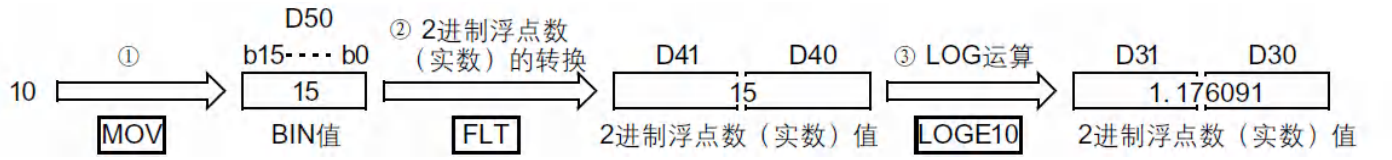
在[S.]中指定的值，只可以设定正值。（负值不能运算）

出错

S.中指定的值为负时，错误代码：K6706；S.中指定的值为 0 时，错误代码：K6706；

➢举例

X000 为 ON 后，求出 D50 中设定的“15”的自然对数，并保存到 D30，D31 中的程序。



FNC127 - ESQR：2 进制浮点数开方运算

➢功能说明

将 S.及其后的一个软元件表示的二进制浮点值进行开方根运算，存入 D. 及其后的一个软元件中。

➢指令格式

DESQR【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存进行常用对数运算的 2 进制浮点数数据的软元件起始编号		D、R	K、H、E	实数（2 进制）
D.	保存运算结果的软元件起始编号	--	D、R	--	

备注：

1）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；

2）运算结果为零时，零标志号动作 M8020 ；

3）源数据的内容只有正数有效，负数时运算错误标志位 M8067，指令不能执行。

➢举例

操作前：float （D1，D0）=9.000，float （D7，D6）=0;

DESQR　D0　D6

操作后：float （D1，D0）=9.000，float （D7，D6）=3.000;

FNC128 - ENEG：2 进制浮点数符号翻转

➢功能说明

将[D.+1，D.]的 2 进制浮点数数据符号翻转运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

DENEG【P】　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	保存要执行符号翻转的 2 进制浮点数数据软元件起始编号	--	D、R	实数 E	实数（2 进制）

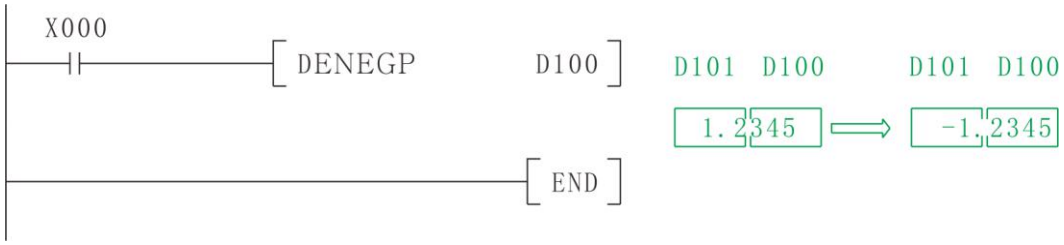
➢动作说明

1、32 位运算（DENEG/DENEGP）

将[D.+1，D.]的 2 进制浮点数数据的符号翻转，并将运算结果保存到[D.+1，D.]中。

➢举例

X000 为 ON 时，将 D100、D101 的 2 进制浮点数数据的符号翻转，并保存到 D100、D101 中的程序。



FNC129 - INT：2 进制浮点数转 BIN 整数

➢功能说明

将 S. 及其后的一个软元件表示的软元件的十进制浮点值转成 BIN 整数，存入 D. 及其后的一个软元件中。

➢指令格式

【D】INT【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换成 BIN 整数的 2 进制数的数据寄存器编号	--	D、R	--	实数（2 进制）
D.	保存转换后的 BIN 整数的数据寄存器编号	--	D、R	K、H	BIN16/32 位

备注：

1）运算结果为零时，零标志号动作 M8020 为 NO；

2）转化时不满 1 而舍去，借位志号动作 M8021 为 NO；

3）运算结果超出以下范围而发生溢出时，进位标志位为 NO。

➢举例

操作前：（D0，D1）=123.6,(D7，D6) =0；

DINT　D0　D6

操作后：(D0，D1）=123.6 ,(D7，D6) =123；

FNC130 - SIN：2 进制浮点数数 SIN 运算

➢功能说明

S.指定的角度（RAD）的 SIN 值，并传送到 D.中的指令。

➢指令格式

DSIN【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 2 进制浮点数的 RAD（角度）的软元件编号	--	D、R	实数 E	实数（2 进制）
D.	保存 2 进制浮点数的 SIN 值的软元件编号	--	D、R	--	

备注：

1）源操作数保存的角度是弧度；

2）计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度；

超出范围的产生报错 M8067 为 NO。

➢举例

操作前：(D3, D2) =0.785（弧度），(D7，D6) =0；

DSIN　D2　D6

操作后：(D3, D2) =0.785 ，(D7，D6) =0.707；

FNC131 - COS: 2 进制浮点数数 COS 运算

➢功能说明

源操作数指定的角度（RAD）的 COS 值，并传送到目的地址中的指令。

➢指令格式

DCOS【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 2 进制浮点数的 RAD（角度）的软元件编号	--	D、R	实数 E	实数（2 进制）
D.	保存 2 进制浮点数的 COS 值的软元件编号	--	D、R	--	

备注：

1）源操作数保存的角度是弧度；

2)计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度超出范围的产生报错 M8067 为 NO。

➢举例

操作前： （D3，D2）= 45 * 31415926/1800000000=0.785（45 度的角度）

DCOS D2 D6

操作后：(D7，D6) =0.707；

FNC132 - TAN：2 进制浮点数 TAN 运算

➢功能说明

源操作数指定的角度（RAD）的 TAN 值，并传送到目的地址中的指令。

DTAN－32 位指令，源操作数与目标操作数为 32 位。

➢指令格式

DTAN【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存 2 进制浮点数的 RAD（角度）的软元件编号	--	D、R	实数 E	实数（2 进制）
D.	保存 2 进制浮点数的 TAN 值的软元件编号	--	D、R	--	

备注：

1）源操作数保存的角度是弧度；

2）计算的角度范围是 0 度到 360 度，也就是也就是 0 到 6.28318 弧度，当为 90 度和 270 度以及超出范围的会产生报错 M8067 为 NO。

➢举例

操作前：(D3，D2)=0.785（弧度），(D7，D6) =0；

DTAN D2 D6

操作后：(D3, D2)=0.785（弧度），(D7，D6) =1.000；

FNC133 - ASIN：2 进制浮点数 SIN-1 运算

➢功能说明

[S.+1，S.]的 SIN 值求出角度的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

DASIN【P】　S.　D.

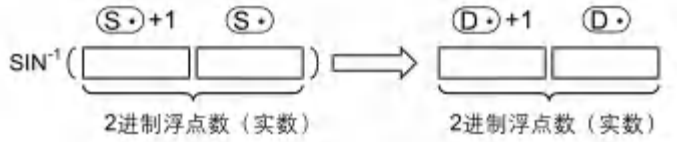
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存执行 SIN^{-1}（反正弦）运算的 SIN 值的软元件起始编号	--	D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件编号	--	D、R	--	

➢动作说明

1、32 位运算（DASIN/DASINP）

从[S.+1，S.]的 SIN 值求出角度，并将运算结果保存到　[D.+1,D.]中。

此外，在[S.]中可以直接指定实数。



[S.+1，S.]的 SIN 值，可以在-1.0~1.0 的范围内设定。

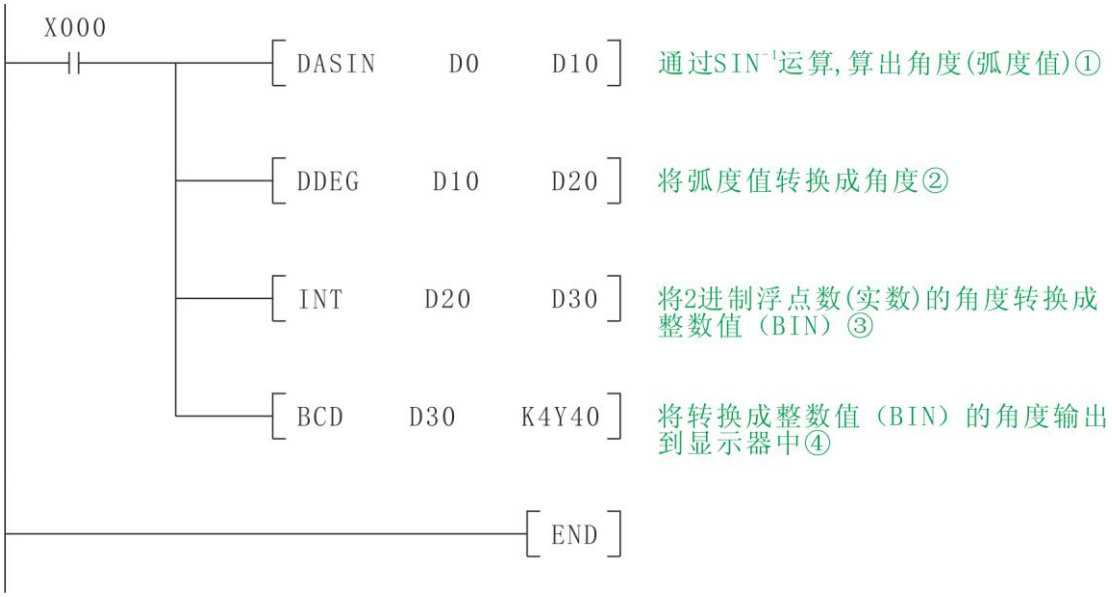
[D.+1，D.]中保存的角度（运算结果）是保存弧度单位的（-π/2）~（π/2）的值。

➢出错

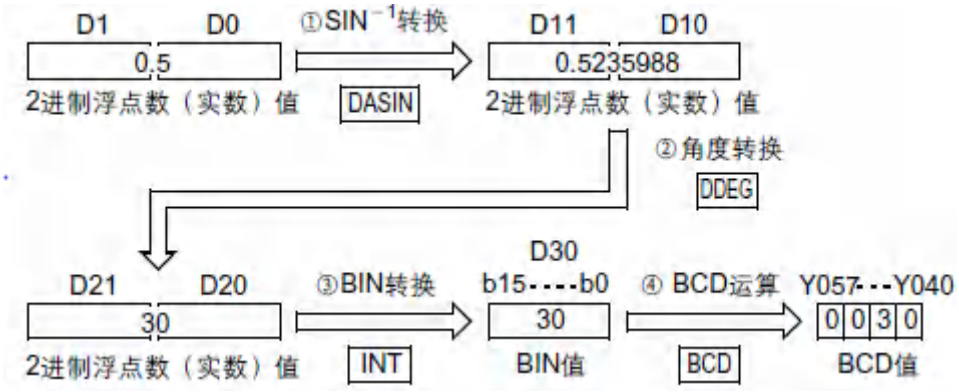
下面的情况运算出错，出错标志位（M8067）为 ON，在 D8067 中保存错误代码。S.中指定的值不在-1.0~1.0 的范围内时，错误代码 K6707.

➢举例

X000 为 ON 时，将 D0、D1 的 2 进制浮点数的 SIN^{-1}，然后将其角度以 BCD4 位数形式输出到 Y040~Y057 中的程序。



D0，D1 的值为 0.5 时的动作



FNC134 - ACOS：2 进制浮点数 COS-1 运算

➢功能说明

[S.+1，S.]的 COS 值求出角度的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

DACOS　S.　D.

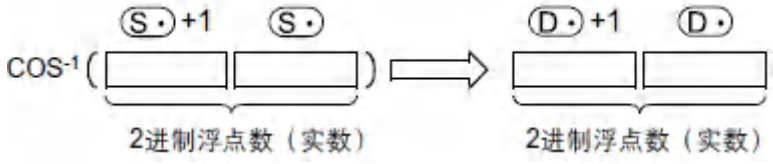
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存执行 COS^{-1}（反余弦）运算的 cos 值的软元件起始编号	--	D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件编号	--	D、R	--	

➢动作说明

1、32 位运算（DACOS/DACOSP）

从[S.+1，S.]的 COS 值求出角度，并将运算结果保存到　[D.+1,D.]中。

此外，在[S.]中可以直接指定实数。

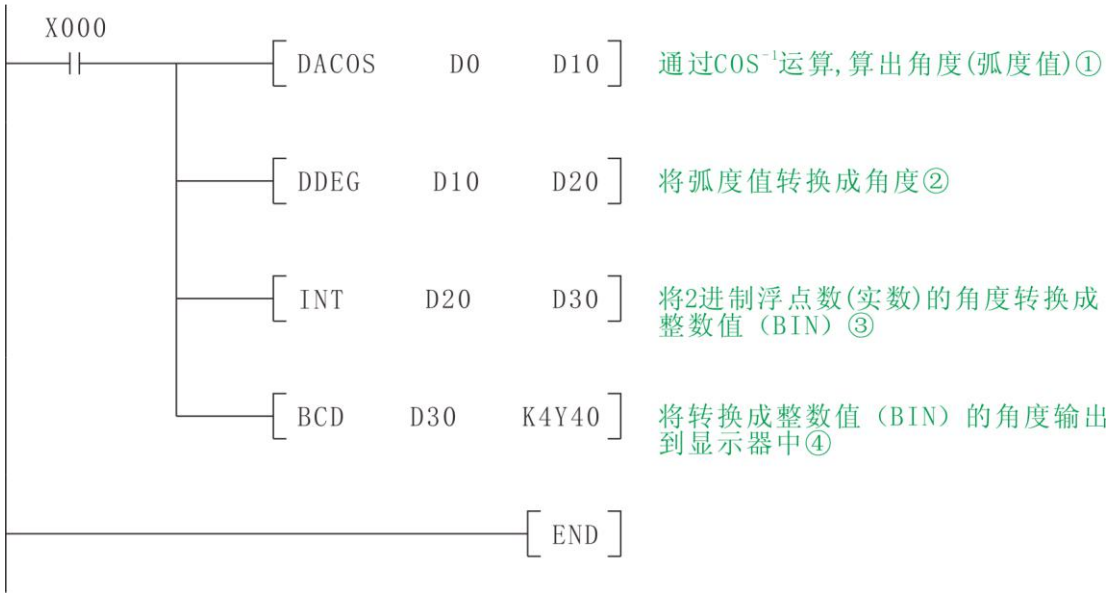


[S.+1，S.]的 COS 值，可以在-1.0~1.0 的范围内设定。

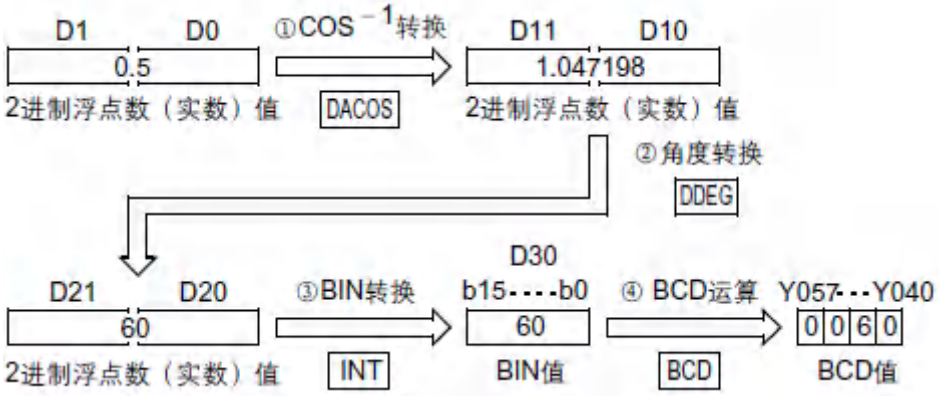
[D.+1，D.]中保存的角度（运算结果）是保存弧度单位的 0~π 的值。

➢举例

X000 为 ON 时，将 D0、D1 的 2 进制浮点数的 COS^{-1}，然后将其角度以 BCD4 位数形式输出到 Y040~Y057 中的程序。



D0，D1 的值为 0.5 时的动作



FNC135 - ATAN：2 进制浮点数 TAN-1 运算

➢功能说明

[S.+1，S.]的 TAN 值求出角度的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

DATAN【P】 S. D.

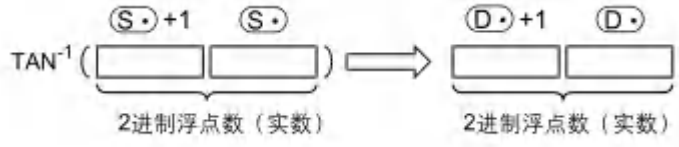
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存执行 TAN^{-1}（反正弦）运算的 tan 值的软元件起始编号	--	D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件编号	--	D、R	--	

➢动作说明

1、32 位运算（DATAN/DATANP）

从[S.+1，S.]的 TAN 值求出角度，并将运算结果保存到 [D.+1,D.]中。

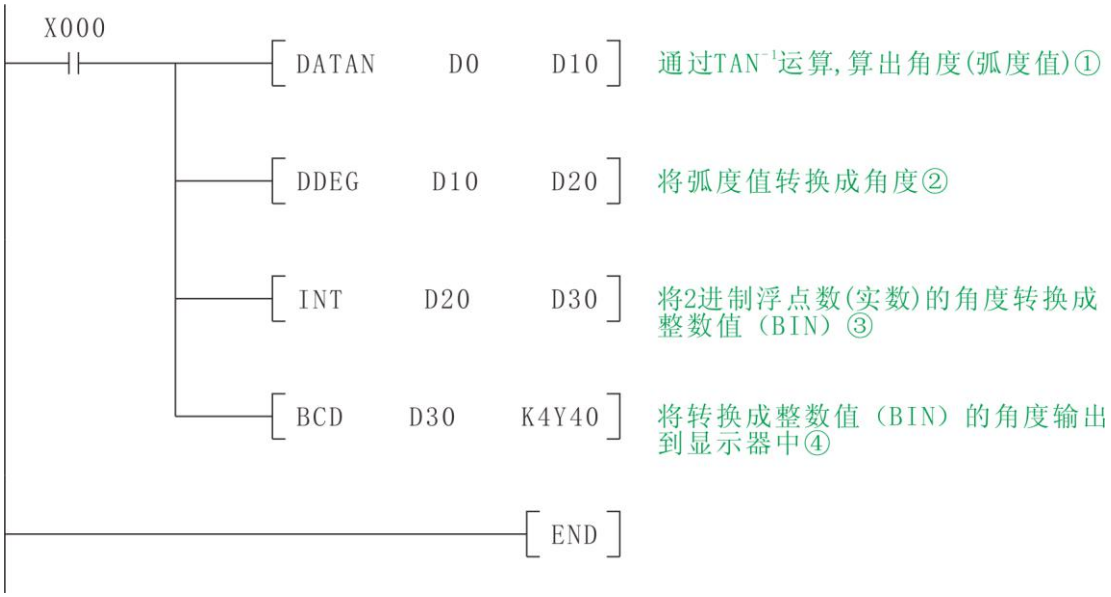
此外，在[S.]中可以直接指定实数。



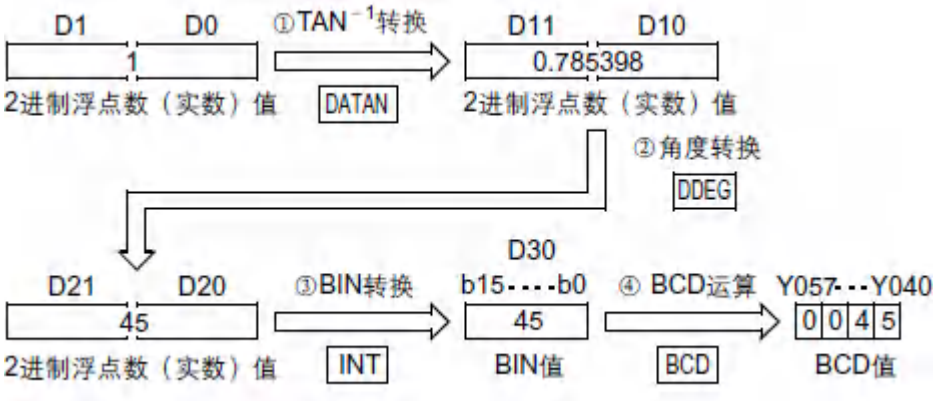
[D. +1，D.]中保存的角度（运算结果），保存以弧度为单位的比（-π/2）大，比（π/2）小的值。

➢举例

X000 为 ON 时，将 D0、D1 的 2 进制浮点数的 TAN^{-1}，然后将其角度以 BCD4 位数形式输出到 Y040~Y057 中的程序。



D0，D1 的值为 1 时的动作



FNC136 - RAD：2 进制浮点数角度转弧度

➢功能说明

将[S.+1，S.]的角度单位值转换成弧度单位的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

DRAD【P】　S.　D.

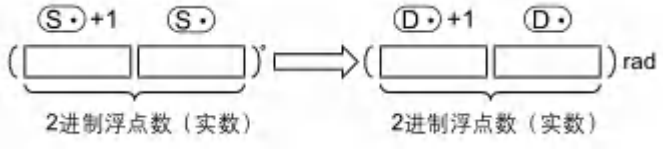
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换成弧度单位的角度的软元件起始编号	--	D、R	实数 E	实数（2 进制）
D.	保存运算结果的软元件编号	--	D、R	--	

➢动作说明

1、32 位运算（DRAD/DRADP）

从[S.+1，S.]的单位从角度单位转换成弧度单位，并将运算结果保存到 [D.+1，D.]中。

此外，在[S.]中可以直接指定实数。



角度单位→弧度单位的转换如下所示执行。

弧度单位=角度单位×π/180

➢举例

X000 为 ON 时，将 X020~X037 中以 BCD4 位数形式设定的角度转换成弧度，以 2 进制浮点数行式保存在 D20，D21 中的程序。

在 X020~X037 中指定了 120 时的动作

FNC137 - DEG：2 进制浮点数弧度转角度

➢功能说明

将[S.+1，S.]的单位从弧度单位准话成角度的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

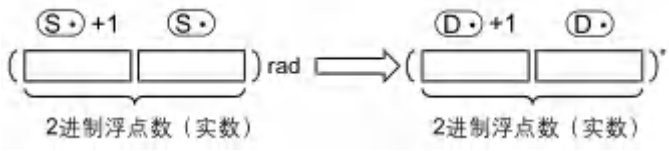
DDEG【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换成角度单位的角度的软元件起始编号	--	D、R	实数 E	实数（2 进制）
D.	保存已转换角度单位的值的软元件编号	--	D、R	--	

➢动作说明

1、32 位运算（DDEG/DDEGP）

单位从角 将[S.+1，S.]的单位从弧度单位转换成角度单位，并将运算结果保存到 [D.+1 ， D.]中。

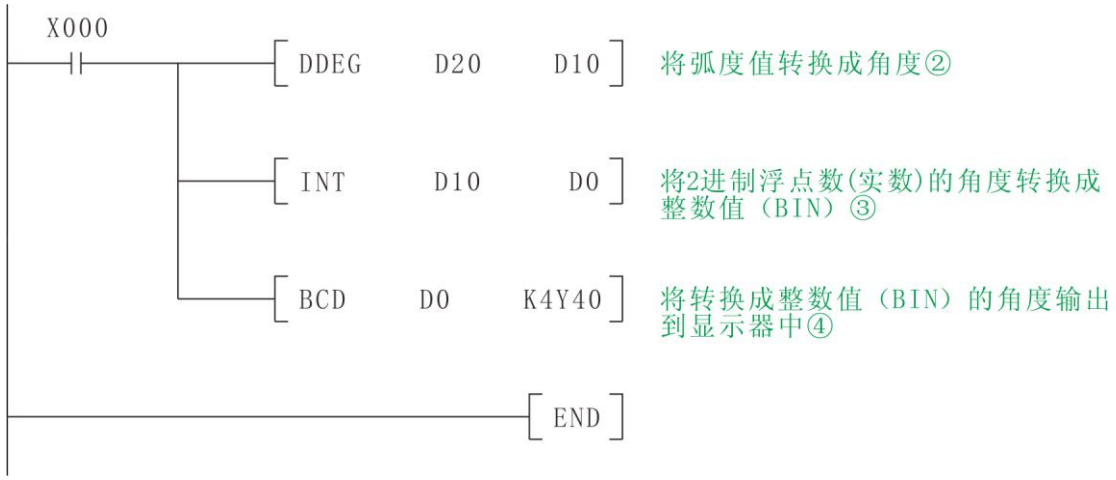


弧度单位→角度单位的转换如下所示执行。

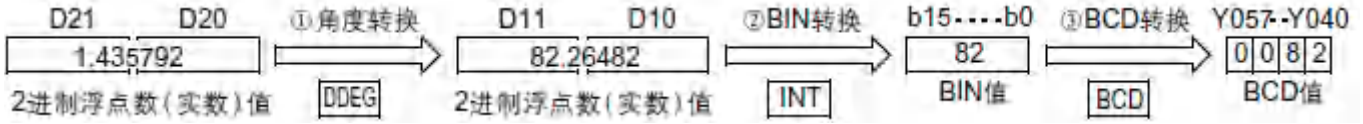
角度单位=弧度单位×180/π

➢举例

X000 为 ON 时，将 D20、D21 中以 2 进制浮点数形式设定的弧度值转换成角度后，以 BCD 值形式输出到 Y040~Y057 中的程序。



D20、D21 的值为 1.435792 时的动作



7-12 数据处理 2 - FNC140~FNC149

FNC140 - WSUM：算出数据合计值

➢功能说明

将[S.]开始的 n 点 16 位数据或者 32 位数据的合计值的运算指令，将运算结果数据存储到[D.+1，D.]中

➢指令格式

WSUM【P】　S.　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要算出合计值的软元件编号	--	T、C、D、R	--	BIN16/32 位
D.	保存合计值的软元件编号	--	T、C、D、R	--	BIN16/64 位
n	数据个数（0<n）	--	D、R	K、H	BIN16/32 位

备注：

1）以[S.]开始的 n 点结束软元件范围不能超出使用的该软元件的范围（出错代码：K6706）

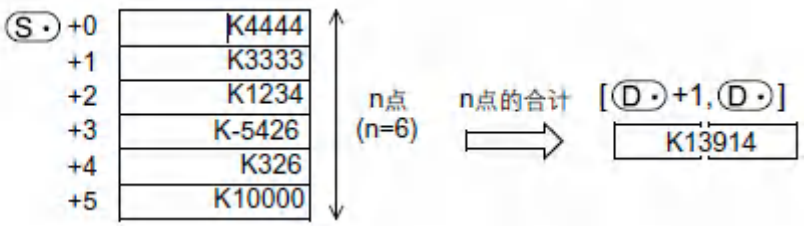
2）n>0 出错代码：（K6706）

3）[D.]不能超出该软元件的使用范围 （出错代码：K6706）

➢动作说明

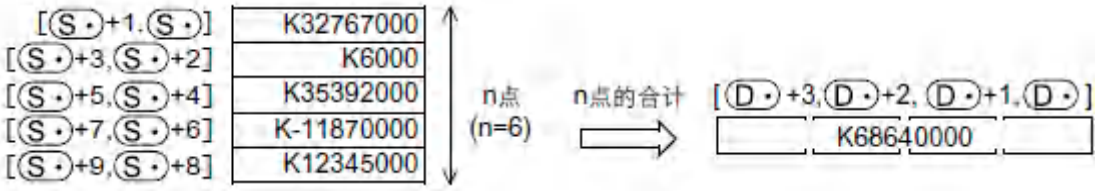
1、16 位运算（WSUM/WSUMP）

将[S.]开始的 n 点 16 位数据的合计值，以 32 位数据形式保存到 [D.+1，D.]中。



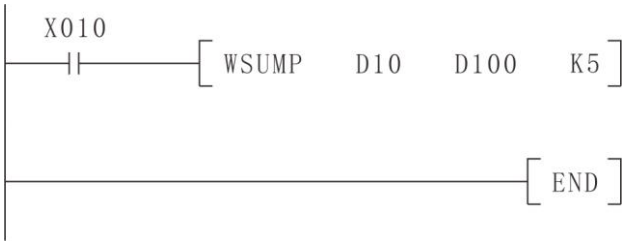
2、32 位运算（DWSUM/DWSUMP）

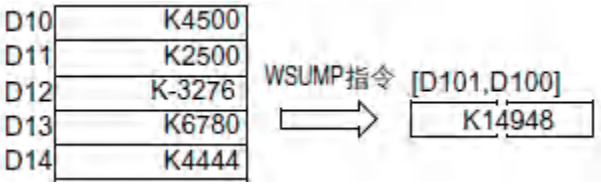
将[S.+1，S.]开始的 n 点 32 位数据的合计值，以 64 位数据形式保存在 [D.+3，D.+2，D.+1，D.]中。



➢举例

X010 为 ON 时，将 D10~D14 的 16 位数据的合计值保存到[D101，D100]中的程序。





FNC141 - WTOB：字节单位的数据分离

➢功能说明

将[S.]开始的 n/2 个软元件中的 16 位数据分离成 n 个字节的运算指令，将运算结果数据存储到[D.]开始的 n 个软元件中

➢指令格式

WTOB【P】　S.　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要按照字节单位进行分离的数据的软元件编号	--	T、C、D、R	--	BIN16 位
D.	保存已经按照字节单位分离的结果的软元件编号	--	T、C、D、R	--	BIN16 位
n	要分离的字节数据个数（0≤n）	--	D、R	K、H	BIN16 位

备注：

1）分离源软元件[S.]~[S.+n/2-1]使用范围不能超出该软元件的范围（出错代码：K6706）

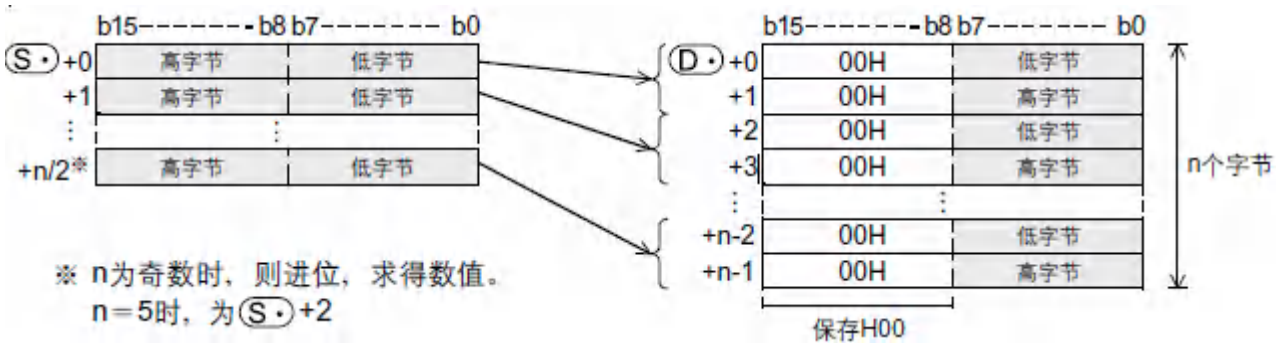
2）保存已分离的数据的软元件[D.]~[D.+n-1]范围不能超出该软元件的范围（K6706）

3）保存分离源的软元件和保存已分离的数据的软元件可以重复使用，但 n 为奇数时，分离前的最终数据的高字节（8 位）的数据被覆盖后有可能会丢失。

➢动作说明

1、16 位运算（WTOBP/WTOBP）

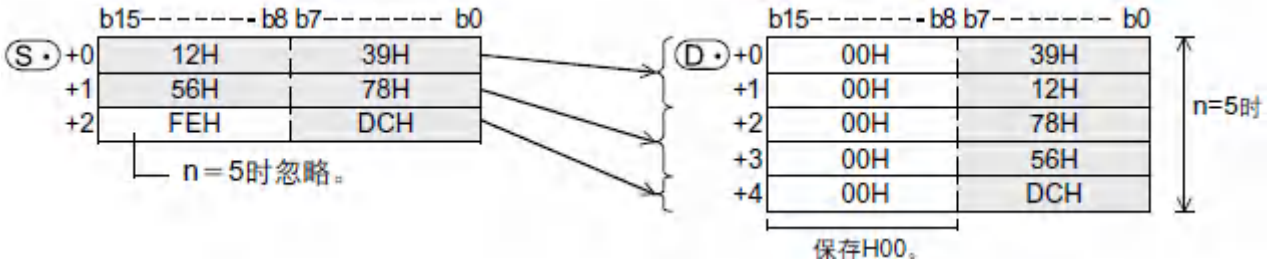
1）将[S.]开始的 n/2 个软元件中保存的 16 位数据分离成 n 个字节，如下保存到以[D.]开始的 n 点软元件中。



2）在保存分离后字节数据的软元件（[D.]以后）的高字节（8 位）中，保存 H00。

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

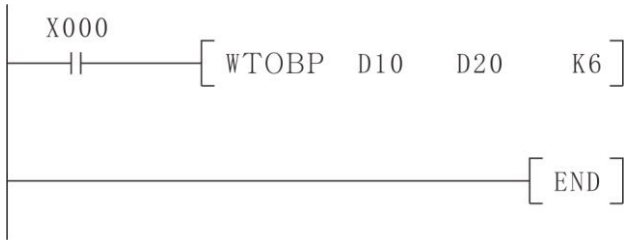
例如，n=5 时，[S.]~[S.+2]的低字节（8 位）的数据被保存在[D.]~[D.+4]中。

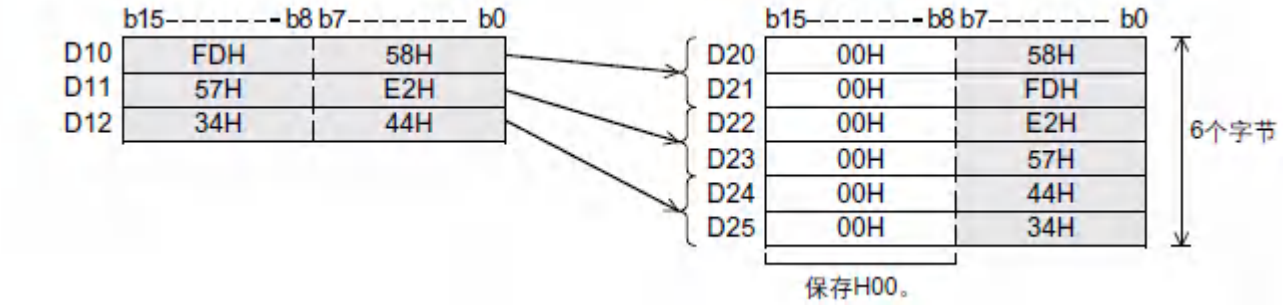


4）n=0 时，指令不执行。

➢举例

X000 为 ON 时，将 D10~D12 的数据按照字节单位分离，然后保存到 D20~D25 中的程序。





FNC142 - BTOW：字节单位的数据结合

➢功能说明

将[S.]开始的 n 个软元件中的 16 位数据的低字节（8 位）结合在一起的运算指令，将运算结果数据存储到[D.] 开始的 n/2 个软元件中

➢指令格式

BTOW【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要按照字节单位结合的数据的软元件编号	--	T、C、D、R	--	BIN16 位
D.	保存已经按照字节单位结合的结果的软元件编号	--	T、C、D、R	--	BIN16 位
n	要结合的字节数据个数（0≤n）	--	D、R	K、H	BIN16 位

备注：

1）结合源软元件[S.]~[S.+n-1]使用范围不能超出该软元件的范围（出错代码：K6706）

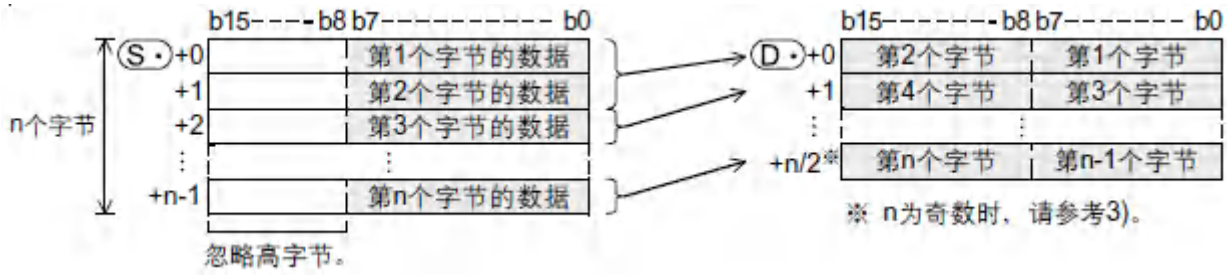
2）保存已结合的数据的软元件[D.]~[D.+n/2-1]范围不能超出该软元件的范围（K6706）

3）保存结合源的软元件和保存已结合的数据的软元件可以重复使用，但请注意重复使用的软元件中，保存的结合源数据的高字节（8 位）的数据会被结合后的数据覆盖后丢失。

➢动作说明

1、16 位运算（BTOW/BTOWP）

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

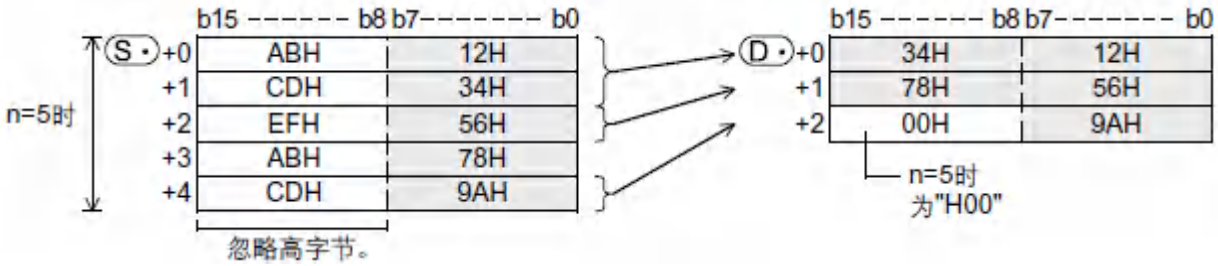


2） 结合源的 16 位数据（[S.]以后）的高字节（8 位）被忽略。

3） n 为奇数时，如下图所示，最终结合后的数据的高字节（8 位）为 H00。

例如，n=5 时，[S.]~[S.+4]的低字节（8 位）的数据被保存在[D.]~[D.+2]中。

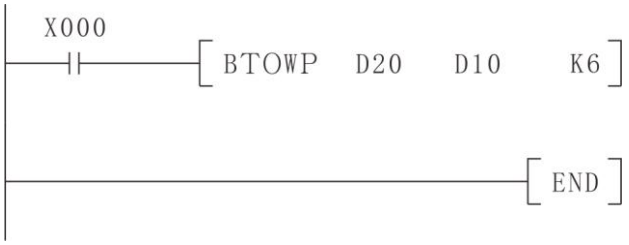
[D.+2]的高字节（8 位）为 H00.

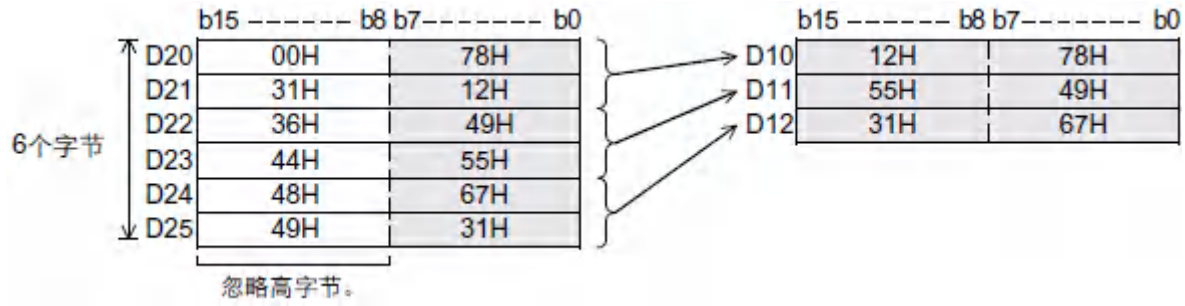


4） n=0 时，指令不执行。

➢举例

X000 为 ON 时，将 D20~D25 的低字节（8 位）数据被结合后，然后保存到 D10~D12 中的程序。





FNC143 - UNI：16 位数据的 4 位结合

➢功能说明

将[S.]开始的 n 个软元件中的 16 位数据的低 4 位结合在一起的运算指令，将运算结果数据存储到[D.]软元件中

➢指令格式

UNI【P】　S.　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要结合的数据的软元件编号	--	T、C、D、R	--	BIN16 位
D.	保存已经结合的结果的软元件编号	--	T、C、D、R	--	BIN16 位
n	结合数（0≤n≤4，n=0 时不处理）	--	D、R	K、H	BIN16 位

备注：

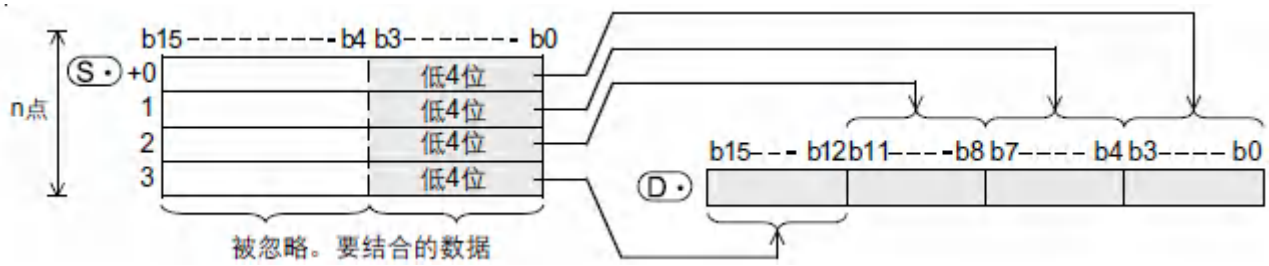
1）[S.]~[S.+n]中指定的软元件使用范围不能超出该软元件的范围，出错代码 K6706；

2） n 只能指定 0~4 的数字，出错代码 K6706。

➢功能和动作说明

1、16 位运算（UNI/UNIP）

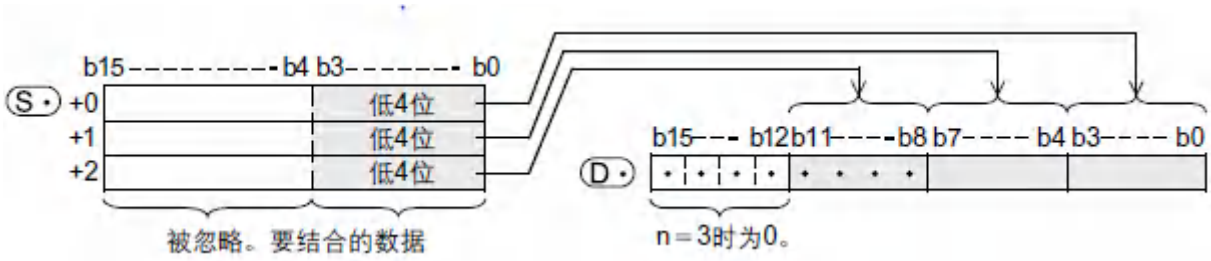
1）将[S.]开始的 n 点 16 位数据的低 4 位结合后的 16 位数据，如下保存到 [D.]软元件中。



2）在 n 中指定 1~4。n=0 时，不执行指令

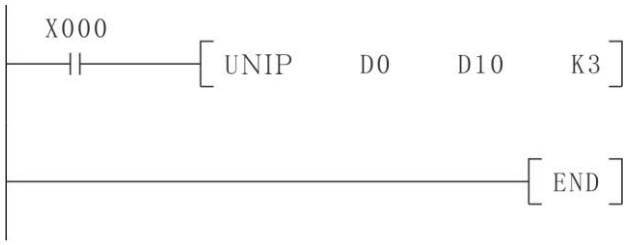
3）1≤n≤3 时，[D.]高位｛4X（4-n）｝个位为 0

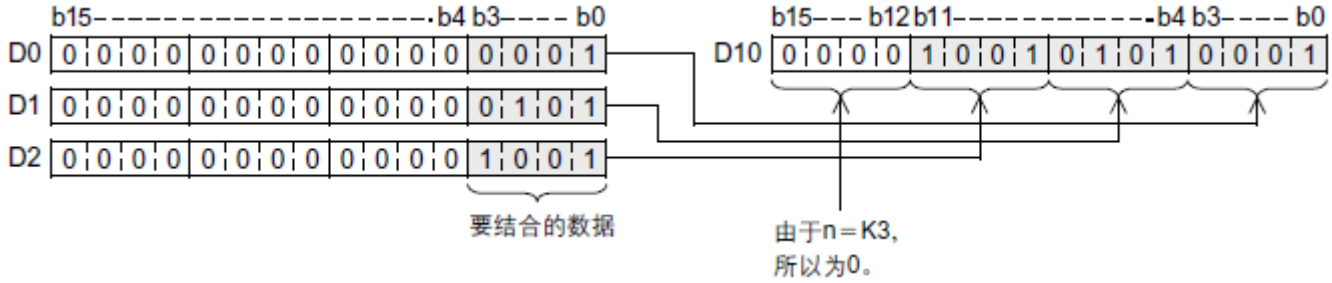
例如，n=3 时，[S.]~[S.+2]的低 4 位被保存到[D.]的 b0~b11 中，[D.]的高 4 位为 0



➢举例

X000 为 ON 时，将 D0~D2 的低 4 位结合后，然后保存到 D10 中的程序。





FNC144 - DIS：16 位数据的 4 位分离

➢功能说明

将[S.]的 16 位数据以低 4 位结合为单位分离的运算指令，将运算结果数据存储到[D.] 软元件中

➢指令格式

DIS【P】 S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要分离的数据的软元件编号	--	T、C、D、R	--	BIN16 位
D.	保存已经分离的结果的软元件编号	--	T、C、D、R	--	BIN16 位
n	分离数（0≤n≤4，n=0 时不处理）	--	D、R	K、H	BIN16 位

备注：

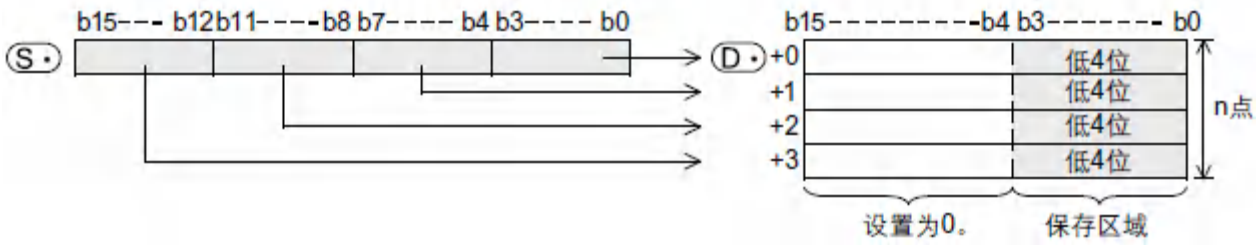
1）[D.]开始的 n 点的软元件使用范围不能超出该软元件的范围，出错代码：K6706；

2） n 只能指定 0~4 的数字，出错代码：K6706。

➢动作说明

1、16 位运算（DIS/DISP）

1）将[S.]的 16 位数据以低 4 位为单位分离后，如下保存到 [D.]软元件中。

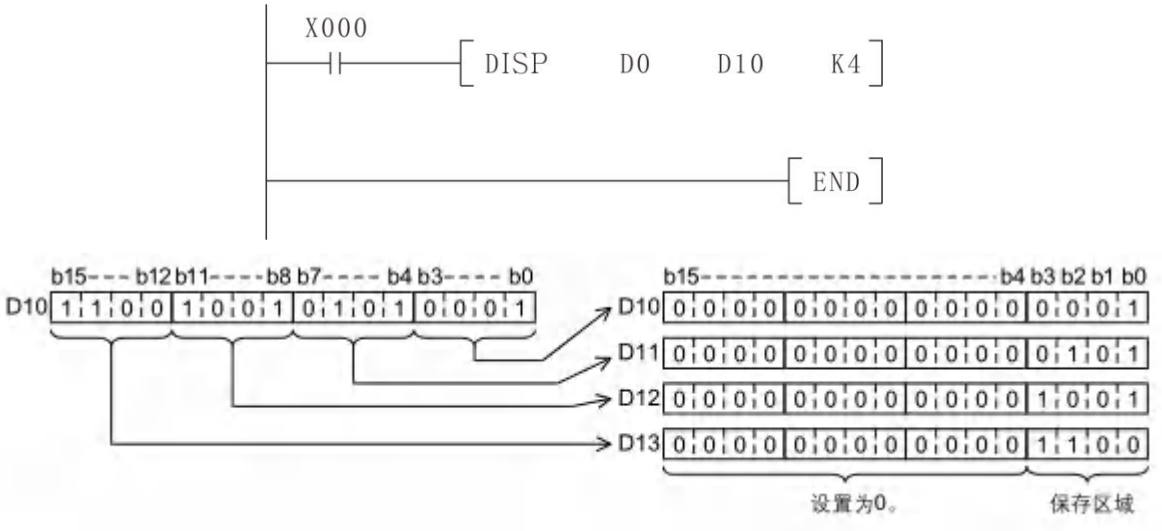


2）在 n 中指定 1~4。n=0 时，不执行指令。

3）[D.]开始的 n 点软元件中的高 12 为设置为 0。

➢举例

当 X000 为 ON 后，将 D0 每隔 4 个位分离后，保存到 D10~D13 中的程序。



FNC147 - SWAP 高低字节互换

➢功能说明

S.的各个低八位与高八位相互交换。

➢指令格式

【D】SWAP【P】 S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	高低字节互换的字软元件	--	KnY、KnM、KnS、T、C、D、R、V、Z	--	BIN16 位

➢举例

操作前：（D0）=0x1234；
SWAP D0
操作后：（D0）=0x3412；

操作前：（D0）=0x12345678；
DSWAP D0
操作后：（D0）=0x34127856；

FNC149 - SORT2：数据排序 2

➢ 功能说明

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

➢ 指令格式

【D】SORT2　S.　m1　m2　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存数据表格的软元件起始编号（占用 m1*m2 点）	--	D、R	--	BIN16/32 位
m1	数据（行）数（1≤m1≤32）	--	D、R	K、H	
m2	群数据（列）数（1≤m1≤6）	--	--	K、H	
D.	保存运算结果的软元件起始编号（占用 m1*m2 点）	--	D、R	--	
n	作为排序标准的群数据（列）的列编号（1~m2）	--	D、R	K、H	

动作说明

1、16 位运算（SORT2）

针对[S.]开始的（m1×m2）点的数据表格（排列前），以 n 列的群数据为基准，将数据行进行升序或是降序排列，然后保存到从[D.]开始的（m1×m2）点的数据表格（排序后）中

下面列举排序前 m1=3，m2=4 的例子说明数据表格中，请把[S.]改读成[D.]。

		群数 m2 个（m2=4）			
		1 管理编号	2 身高	3 体重	4 年龄
数据 m1=3	1	[S.]	[S.+1]	[S.+2]	[S.+3]

的情况	2	[S.+4]	[S.+5]	[S.+6]	[S.+7]
	3	[S.+8]	[S.+9]	[S.+10]	[S.+11]

通过 M8165 的 ON/OFF 状态来设定排序

软元件	设定排序的顺序
M8165=ON	降序排序
M8165=OFF	升序排序

指令输入为 ON 时开始数据排列，m1 个扫描后数据排列结束，指令执行标志位为 ON。

2、32 位运算（DSORT2）

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

下面列举排序前 m1=3，m2=4 的例子说明数据表格中，请把[S.]改读成[D.]。

		群数 m2 个（m2=4）			
		1 管理编号	2 身高	3 体重	4 年龄
数据 m1=3 的情况	1	[S.+1，S.]	[S.+3，S.+2]	[S.+5，S.+4]	[S.+7，S.+6]
	2	[S.+9，S.+8]	[S.+11，S.+10]	[S.+13，S.+12]	[S.+15，S.+14]
	3	[S.+17，S.+16]	[S.+19，S.+18]	[S.+21，S.+20]	[S.+23，S.+22]

通过 M8165 的 ON/OFF 状态来设定排序

软元件	设定排序的顺序
M8165=ON	降序排序
M8165=OFF	升序排序

指令输入为 ON 时开始数据排列，m1 个扫描后数据排列结束，指令执行标志位为 ON

在 m1 中使用数据寄存器 D 或文件寄存器 R 时，为 32 位长度数据

例如，在 D0 中指定 m1 时，m1 为[D1，D0]的 32 位数据

3、动作举例

在“n=2（列号 2）”和“n=3（列号 3）”的情况下，对如下所示的排列前的数据进行排列，动作如下所示。

下面例举了 16 位运算的动作例子。执行 32 为运算时，请使用 BIN32 位构成数据表格。

此外，如果先在第 1 列中输入管理编号等连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前的数据

		群数 m2 个（m2=4）			
		1 管理编号	2 身高	3 体重	4 年龄
数据 m1=5 的情况	1	[S.]	[S.+1]	[S.+2]	[S.+3]
		1	150	45	20
	2	[S.+4]	[S.+5]	[S.+6]	[S.+7]
		2	180	50	40
	3	[S.+8]	[S.+9]	[S.+10]	[S.+11]
		3	160	70	30

	4	[S.+12]	[S.+13]	[S.+14]	[S.+15]
		4	100	20	8
	5	[S.+16]	[S.+17]	[S.+18]	[S.+19]
		5	150	50	45

1）以 n=K2（列号 2）为基准执行指令时的排序结果（升序的情况）

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

2）以 n=K3（列号 3）为基准执行指令时的排序结果（降序的情况）

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

相关特殊继电器

软元件	名称	内容
M8029	指令执行结束	数据排序结束时为 ON
M8165	降序排序	M8165=ON 时，降序排序 M8165=OFF 时，升序排序

➢相关指令

操作数种类	内容
FNC69 - SORT	数据排序 以指定的群数据（列）为基准，以行为单位，将由数据（行）和群数据（列） 构成的数据表进行升序排列的指令。这个指令中，在连续的软元件中保存群数据（列方向）

➢注意要点

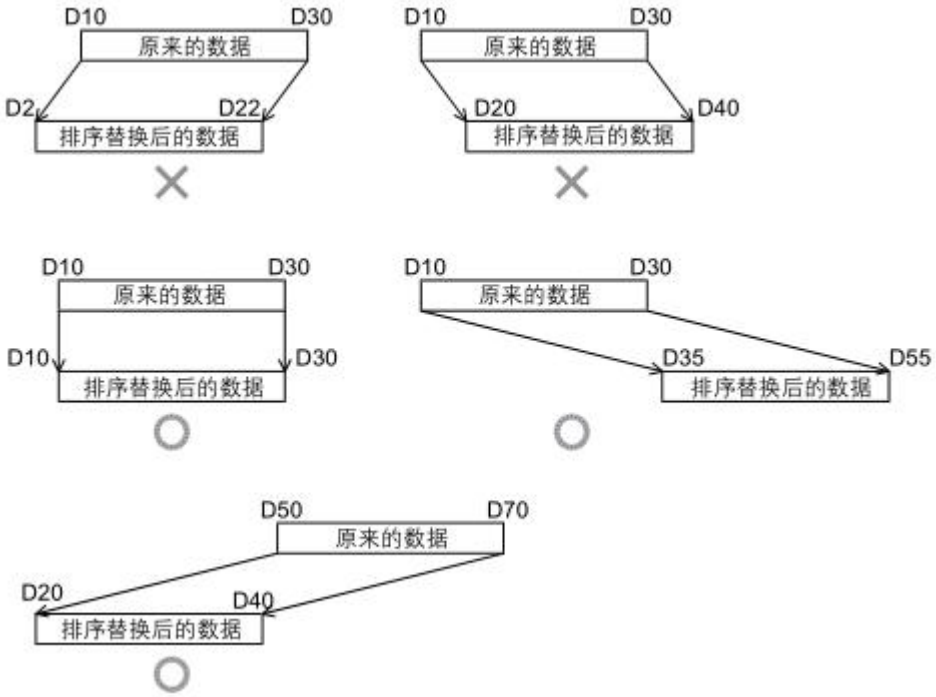
1）动作过程中，请勿使操作数和数据的内容变化；

2）再次执行时，请将指令输入 OFF 一次；

3）指令的使用次数的限制，在程序的使用次数最多可同时驱动 2 次。

4）包含该指令的回路块不能在 RUN 中写入。原来的数据按照排序后的数据顺序被改写。到指令执行结束之前，尤其请注意不要改变 S.的内容。

5）原来的数据和排序替换后的数据，请错开，不要重叠。



7-13 定位控制指令 - FNC150~FNC159

FNC150 - DSZR：带 DOG 搜索的原点回归

➢ 功能说明

执行原点回归，使机械位置与可编程控制器内的当前值寄存器一致的指令。

此外，FNC156 - ZRN 指令不支持一下几种情况，但本指令可以支持。

1）DOG 搜索功能的对应

2）允许使用近点 DOG 和零点信号的原点回归，但是不可以对零信号计数后决定原点。

➢ 指令格式

DSZR　S1.　S2.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定输入近点信号（DOG）的软元件编号	X、Y、M、T、D□.b	--	--	BIN16/32 位
S2.	指定输入零点信号的输入编号	X	--	K、H	
D1.	指定输出脉冲的输出编号	Y	--	K、H	
D2.	指定旋转方向信号的输出对象编号	Y、M、T、D□.b	--	--	

备注：

1）D□.b 不能变址修饰（V、Z）。

2）请指定 X000~X007。

3）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ 关于 RUN 中写入的注意事项

在执行 FNC150 - DSZR 指令过程中（脉冲输出过程中），请避免执行 RUN 中写入。

万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

FNC151 - DVIT：中断定位

➢ 功能说明

执行单速中断定长进给的指令。

➢ 指令格式

【D】DVIT　S1.　S2.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定中断后的输出脉冲数（相对地址）	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	指定输出脉冲频率	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
D1.	指定输出脉冲的输出编号	Y	修饰	K、H	
D2.	指定旋转方向信号的输出对象编号	Y、M、S、D□.b	修饰	--	位

备注：

1）16 位运算时为-32768≤S1.≤+32767（0 除外）；32 位运算时为-999999≤S1.≤+999999（0 除外）；

2）16 位运算时为 10≤S2.≤+32767（Hz）；32 位运算时为 10≤S2.≤100000（Hz）；

3）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ 关于 RUN 中写入的注意事项

在执行 FNC151 - DVIT 指令过程中（脉冲输出过程中），请避免执行 RUN 中写入。

万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

FNC152 - TBL：表格设定定位

➢ 功能说明

预先将数据表格中被设定的指令的动作，变为指定的 1 个表格的动作。

指令	内容
FNC151 - DVIT	中断定位
FNC157 - PLSV	可变速脉冲输出

FNC158 - DRVI	相对定位
FNC159 - DRVA	绝对定位

➢ 指令格式

DTBL D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定输出脉冲的输出编号	Y	--	--	位
n	执行的表格编号，1≤n≤100	Y、M、S、D□.b	--	K、H	BIN32 位

备注：

1）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ 关于 **RUN** 中写入的注意事项

不能对包含 FNC152 - TBL 指令的回路块执行 RUN 中写入。

FNC155 - ABS：读出 ABS 当前值

FNC156 - ZRN：原点回归

➢ 功能说明

执行原点回归使机械位置与可编程控制器内的当前值寄存器一致的指令。

需要 DOG 搜索功能时，请使用 FNC150 - DSZR 指令。

➢ 指令格式

ZRN S1. S2. S3. D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定开始原点回归时的速度，10≤S2≤100000(Hz)	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	指定爬行速度，10≤S2≤32767(Hz)	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
S3.	指定要输入近点信号（DOG）的输入编号的软元件编号	X、Y、M、S、D□.b	修饰	--	
D2.	指定要输出脉冲的输出编号	Y	修饰	--	位

备注：

1）D□.b 不能变址修饰（V、Z）。

2）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ 关于 **RUN** 中写入的注意事项

在 FNC156 - ZRN 指令执行过程中（脉冲输出过程中），请避免执行 RUN 中写入。

万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

FNC157 - PLSV：可变速脉冲输出

➢ 功能说明

执行原点回归使机械位置与可编程控制器内的当前值寄存器一致的指令。

需要 DOG 搜索功能时，请使用 FNC150 - DSZR 指令。

➢ 指令格式

【D】PLSV　S1.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定输出脉冲频率的软元件编号，-32768≤S2≤+32768 Hz（0 除外）	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
D1.	指定要输出脉冲的输出编号。	Y	修饰	--	位
D2.	指定旋转方向信号的输出对象编号	Y、M、S、D□.b	修饰	--	

备注：

1）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ 关于 **RUN** 中写入的注意事项

在执行 FNC158 - DRVI 指令执行过程中（脉冲输出过程中），请避免执行 RUN 中写入。

万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

FNC158 - DRVI：相对定位

➢ 功能说明

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

➢ 指令格式

【D】DRVI　S1.　S2.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定输出脉冲数	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	指定输出脉冲频率	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
D1.	指定输出脉冲的输出编号	Y	修饰	--	位
D2.	指定旋转方向信号的输出对象编号	Y、M、S、D□.b	修饰	--	

备注：

1）16 位运算时为-32768≤S1.≤+32767（0 除外）；32 位运算时为-999999≤S1.≤+999999（0 除外）；

2）16 位运算时为 10≤S2.≤+32767（Hz）；32 位运算时为 10≤S2.≤100000（Hz）；

3）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ **关于 RUN 中写入的注意事项**

在执行 FNC157 - PLSV 指令执行过程中（脉冲输出过程中），请避免执行 RUN 中写入。

万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

FNC159 - DRVA：绝对定位

➢ **功能说明**

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

➢ **指令格式**

【D】DRVA　S1.　S2.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定输出脉冲数(绝对地址)	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	指定输出脉冲频率	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
D1.	指定输出脉冲的输出编号	Y	修饰	--	位
D2.	指定旋转方向信号的输出对象编号	Y、M、S、D□.b	修饰	--	

备注：

1）16 位运算时为-32768≤S1.≤+32767（0 除外）；32 位运算时为-999999≤S1.≤+999999（0 除外）；

2）16 位运算时为 10≤S2.≤+32767（Hz）；32 位运算时为 10≤S2.≤100000（Hz）；

3）请指定基本单元的晶体管输出 Y000、Y001、Y002。

➢ **关于 RUN 中写入的注意事项**

在执行 FNC159 - DRVA 指令执行过程中（脉冲输出过程中），请避免执行 RUN 中写入。万一在脉冲输出过程中，对包含该指令的回路块执行了 RUN 中写入，请注意脉冲输出会减速停止。

7-14 时钟连算指令 - FNC160~FNC169

FNC160 - TCMP：时钟数据比较

➢功能说明

S1.作时，S2.作分，S3.作秒。与将 S.起始的三点内的时钟数据进行比较，并将比较结果送到 D.及其后两个软元件上。

➢指令格式

TCMP【P】 S1. S2. S3. S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定比较基准时间的“时”，设定范围：0~23	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
S2.	指定比较基准时间的“分”，设定范围：0~59	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
S3.	指定比较基准时间的“秒”，设定范围：0~59	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16 位
S.	指定时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
D.	根据比较结果 ON/OFF 位软元件。（占用 3 点）	Y、M、S	--	--	位

备注：

1）所有源操作数都被看成二进制值处理；

2）D.自动占三点，如 D.为 M0 时，M0、M1、M2 自动被占用；

➢举例

TCMP K10 K30 K50 D7 M0

操作后：当 10 时 30 分 50 秒>（（D7）时（D8）分（D9）秒）时，（M0）=1，（M1）=0，（M2）=0；

当 10 时 30 分 50 秒=（（D7）时（D8）分（D9）秒）时，（M2）=0，（M3）=1，（M4）=0；

当 10 时 30 分 50 秒<（（D7）时（D8）分（D9）秒）时，（M2）=0，（M3）=0，（M4）=1；

FNC161 - TZCP：时钟数据区间比较

➢功能说明

将 S.起始的三点内的时钟数据,与将 S1.起始的三点内的时钟数据和将 S2.起始的三点内的时钟数据进行比较，并将比较结果送到 D.及其后两个软元件上。

➢指令格式

TZCP S1. S2. S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定比较下限时间（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S2.	指定比较上限时间（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S.	指定时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
D.	根据比较结果 ON/OFF 位软元件。（占用 3 点）	Y、M、S	--	--	位

备注：

1）所有源操作数都被看成二进制值处理；

2）D.自动占三点，如 D.为 M2 时，M2、M3、M4 自动被占用；

3）源操作数 1 起始的三点内的时钟数据不得大于源操作数 2 起始的三点内的时钟数据，当不满足该条件时，默认两个比较值都为源操作数 1 起始的三点内的时钟数。

➢举例

TZCP D10 D20 D0 M2

操作后：当（（D10）时（D11）分（D12）秒）>（（D0）时（D1）分（D2）秒），（M2）=1，（M3）=0，（M4）=0；

当（（D10）时（D11）分（D12）秒）<=（（D0）时（D1）分（D2）秒）<=（（D20）时（D21）分（D22）秒），（M2）=0，（M3）=1，（M4）=0；

当（（D0）时（D1）分（D2）秒）<（（D20）时（D21）分（D22）秒），（M2）=0，（M3）=0，（M4）=1；

FNC162 - TADD：时钟数据加法运算

➢功能说明

保存于 S1.起始的三点内的时钟数据，加上源 S2.起始的三点内的时钟数据，并将其结果保存于以 D.起始的三点软元件内。

➢指令格式

TADD【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定进行加法运算的时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S2.	指定进行加法运算的时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S.	保存 2 个时间数据（时、分、秒）加法运算的结果	--	T、C、D、R、V、Z	--	BIN16 位

备注：

1）当运算结果为 0 时（0 时 0 分 0 秒），零标志位 M8020 置位；

2）当运算结果超过 24 小时时，进位标志位 M8022 置位，将运算的结果减去 24 小时后，将该值作为运算结果保存；

➢举例

操作前：（D0）=10，（D1）=30，（D2）=10，（D10）=3，（D11）=10，（D12）=5，（D20）=0，（D21）=0，（D22）=0，（M8020）=0，（M8022）=0；

TADD D0 D10 D20

操作后：（D0）=10，（D1）=30，（D2）=10，（D10）=3，（D11）=10，（D12）=5，（D20）=13，（D21）=40，（D22）=15，（M8020）=0，（M8022）=0；

FNC163 - TSUB：时钟数据减法运算

➢功能说明

保存于 S1.起始的三点内的时钟数据，减去 S2.起始的三点内的时钟数据，并将其结果保存于以 D.起始的三点软元件内。

TSUB－16 位指令，源操作数与目标操作数为 16 位。

➢指令格式

TSUB【P】　S1.　S2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	指定进行减法运算的时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S2.	指定进行减法运算的时间数据（时、分、秒）的“时”。（占用 3 点）	--	T、C、D、R、V、Z	--	BIN16 位
S.	保存 2 个时间数据（时、分、秒）减法运算的结果	--	T、C、D、R、V、Z	--	BIN16 位

备注：

1）当运算结果为 0 时（0 时 0 分 0 秒），零标志位 M8020 置位；

2）当运算结果超过 0 小时时，借位标志位 M8021 置位，将运算的结果加上 24 小时后，将该值作为运算结果保存）。

➢举例

操作前：（D0）=5，（D1）=20，（D2）=40，（D10）=18，（D11）=10，（D12）=5，（D20）=0，（D21）=0，（D22）=0，（M8020）=0，（M8021）=0；

TSUB　D0　D10　D20

操作后：（D0）=5，（D1）=20，（D2）=40，（D10）=18，（D11）=10，（D12）=5，（D20）=11，（D21）=10，（D22）=35，（M8020）=0，（M8021）=1；

FNC164 - HTOS：时、分及秒数据的秒转换

➢功能说明

将【时、分、秒】单位的时间（时刻）数据转换成秒单位的数据的指令。

➢指令格式

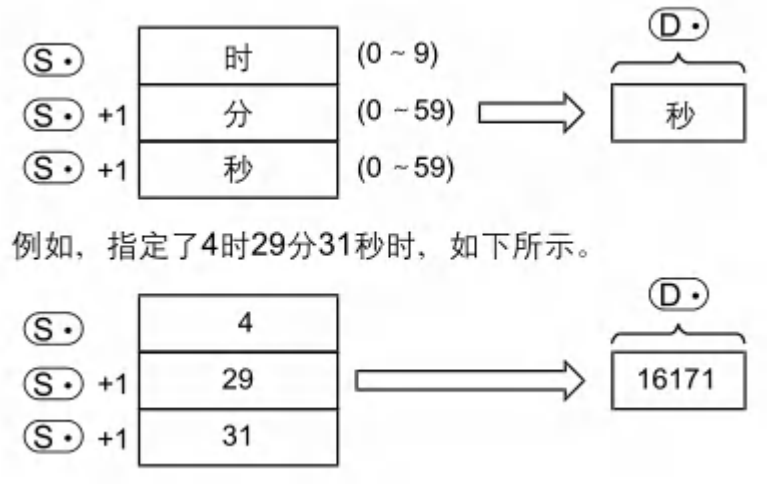
【D】HTOS【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存转换前的时间（时刻）数据（时、分、秒）的软元件的起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	BIN16 位
D.	保存转换后的时间（时刻）数据（秒）的软元件编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	BIN32/16 位

➢动作说明

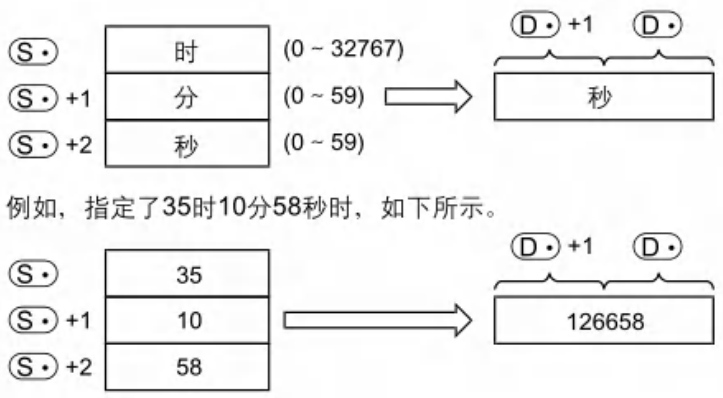
1、16 位运算（HTOS/HTOSP）

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



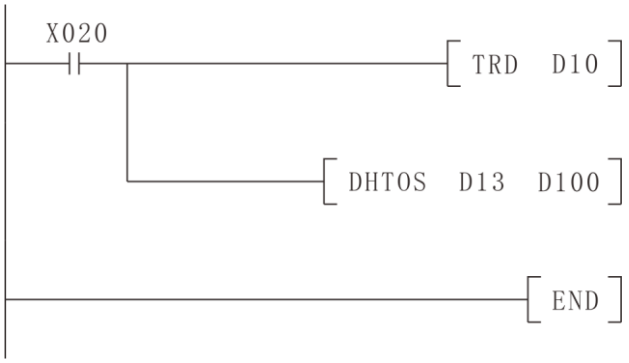
2、32 位运算（DHTOS/DHTOSP）

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



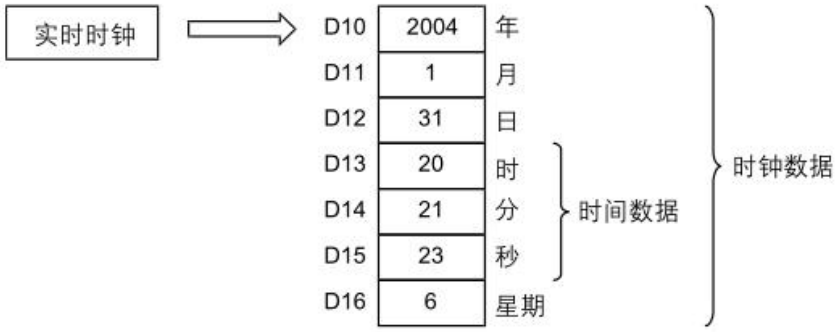
➢**举例**

当 X020 为 ON 时，从 PLC 内置的实时时钟中读出时间数据，换算成秒，然后保存到 D100、D101 中的。

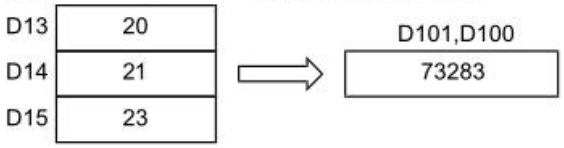


动作

• 使用TRD(FNC 166)指令读出时间数据的动作



• 使用DHTOS(FNC 164)指令转换成秒的动作



FNC165 - STOH：秒数据的【时、分及秒】转换

➢功能说明

将秒单位的时间（时刻）数据转换成【时、分、秒】单位的数据的指令。

➢指令格式

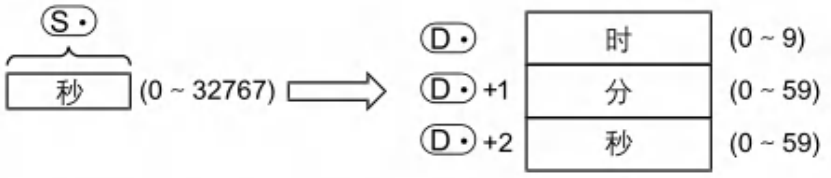
【D】STOH【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存转换后的时间（时刻）数据（秒）的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	BIN16 位
D.	保存转换前的时间（时刻）数据（时、分、秒）的软元件的起始编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	BIN32/16 位

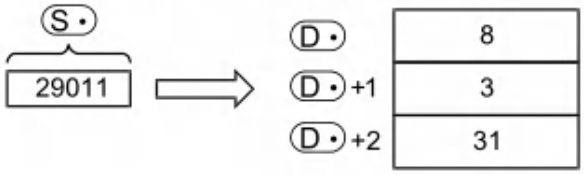
➢动作说明

1、16 位运算（STOH/STOHP）

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

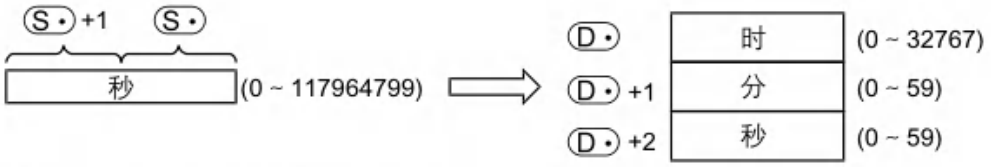


例如，指定了29011秒时，如下所示。

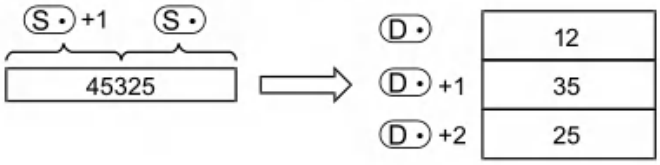


2、32 位运算（DSTOH/DSTOHP）

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



例如，指定了45325秒时，如下所示。

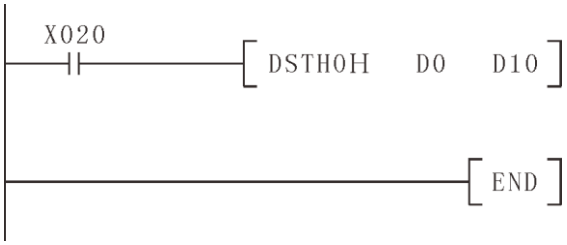


➢出错

以下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。S.数据超出范围时，错误代码 K6706。

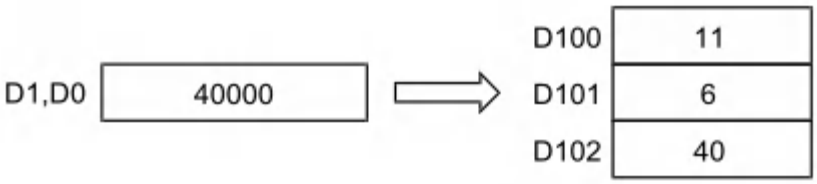
➢举例

当 X020 为 ON 时，将 D0、D1 中保存的秒数据换算成时、分、秒后，其结果保存到【D100、D101、D102】中的程序。



动作

• 使用STOHP指令转换成时、分、秒（在D1、D0中指定40000秒时）



FNC166 - TRD：读出时钟数据

➢功能说明

将可编程控制器的实时时钟的时钟数据，按照年、月、日、时、分、秒、星期读入，以 D.为起始的七点数据寄存器中。

➢指令格式

TRD D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	指定保存读出时间数据的起始软元件编号。（占用 7 点）	--	T、C、D、R	--	BIN16 位

➢举例

操作前：（D0）=0，（D1）=0，（D2）=0，（D3）=0，（D4）=0，（D5）=0，（D6）=0；

TRD D0

操作后：（D0）=2015，（D1）=12，（D2）==26，（D3）=11，（D4）=11，（D5）=9，（D6）=6；

FNC167 - TWR：写入时钟数据

➢功能说明

将时钟数据即以 D.为起始的七点数据寄存器，写入可编程控制器的实时时钟中。

➢指令格式

TWR S.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
D.	指定写入时间数据的起始软元件编号。（占用 7 点）	--	T、C、D、R	--	BIN16 位

➢举例

操作前：（D0）=2015，（D1）=12，（D2）==26，（D3）=11，（D4）=11，（D5）=9，（D6）=6；

TWR D0

操作后： 可编程控制器的时间变为，2015 年 12 月 26 日 11 时 11 分 9 秒星期六；

FNC169 - HOUR：计时表

7-15 外部设备 - FNC170~FNC179

FNC170 - GRY：格雷码的转换

FNC171 - GBIN：格雷码的逆转换

FNC176 - RD3A：模拟量模块的读出

FNC177 - WR3A：模拟量模块的写入

7-16 其他指令 - FNC181~FNC189

FNC182 - COMRD：读出软元件的注释数据

FNC184 - RND：产生随机数

FNC186 - DUTY：产生定时脉冲

➢功能说明

将指定次数的运算周期作为 1 个周期，产生这样定时信号的指令。

➢指令格式

【D】DUTY【P】　n1.　n2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
n1.	ON 的扫描次数（运算周期），n1＞0	--	T、C、D、R	K、H	BIN16 位
n2.	OFF 的扫描次数（运算周期），n2＞0	--	T、C、D、R	K、H	
D.	定时时钟输出的目标地址	M（请指定 M8330~M8334）	修饰		位

➢动作说明

1、16 位运算（DUTY）

1）定时时钟输出 D.是按照 n1 个扫描 ON，n2 个扫描 OFF 的方式进行 ON/OFF。

2）请在定时器时钟的输出的目标地址中，指定 M8330~M8334。

3）与定时时钟输出的目标地址 D.对应的扫描数的计数值被保存到 D8330~D8334 中。扫描数的计数值 D8330~D8334，在计数值变为 n1+n2 时，或者在指令输入（指令）变为 ON 时被复位。

定时时钟输出的目标地址 D.	对扫描数计数用的软元件
M8330	D8330
M8331	D8331
M8332	D8332
M8333	D8333
M8334	D8334

4）在指令输入的上升沿开始动作，在 END 指令处，ON/OFF 定时时钟输出。此外，指令输入即使被切断动作也不停止。STOP 时，通过中断，或是断电时停止。

5）将 n1，n2 设定为 0 时，如下表所示。

n1，n2 的状态	D.的 ON/OFF 状态
n1＝0，n2 ≥ 0	D.固定为 OFF
n1＞0，n2＝0	D.固定为 ON

➢相关软元件

软元件	名称	内容
M8330	定时时钟输出 1	FNC186 - DUTY 指令的定时时钟输出
M8331	定时时钟输出 2	
M8332	定时时钟输出 3	
M8333	定时时钟输出 4	
M8334	定时时钟输出 5	
D8330	定时时钟输出 1 用的扫描次数的计数	FNC186 - DUTY 指令的定时时钟输出 1 用的扫描次数的计数值

D8331	定时时钟输出 2 用的扫描次数的计数	FNC186 - DUTY 指令的定时时钟输出 2 用的扫描次数的计数值
D8332	定时时钟输出 3 用的扫描次数的计数	FNC186 - DUTY 指令的定时时钟输出 3 用的扫描次数的计数值
D8333	定时时钟输出 4 用的扫描次数的计数	FNC186 - DUTY 指令的定时时钟输出 4 用的扫描次数的计数值
D8334	定时时钟输出 5 用的扫描次数的计数	FNC186 - DUTY 指令的定时时钟输出 5 用的扫描次数的计数值

➢**注意要点**

1、该指令能使用 5 次（点）。但是在多个 FUC186 - DUTY 指令中不能使用相同的定时时钟输出目标地址 D.

➢**出错**

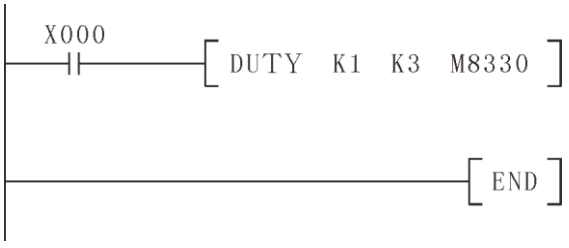
以下一些情况下会出现运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

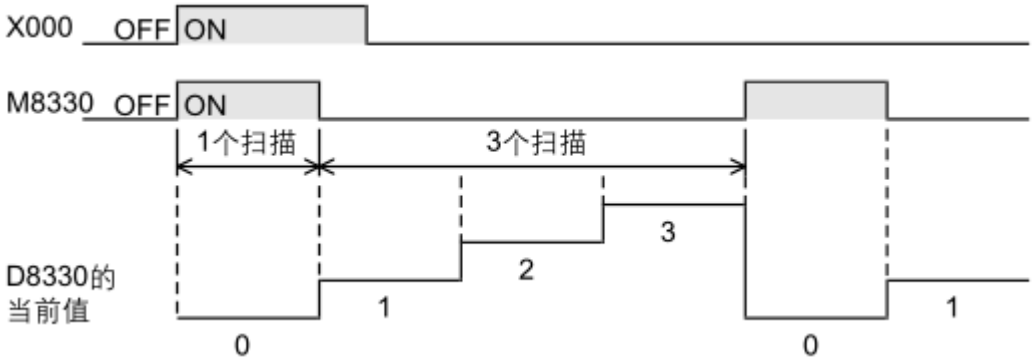
1）n1，n2 不满时。错误代码 K6706。

2）D.在 m8330~M8334 以外范围时。错误代码 K6705。

➢**程序举例**

X000 为 ON 后，M8330 1 个扫描为 ON，3 个扫描为 OFF 的程序。





FNC188 - CRC：CRC 运算

➢功能说明

在通信等中被使用的出错校验方法之一为 CRC（循环冗余校验），用 CRC 指令计算出该 CRC 值。在出错校验的方法中，除了 CRC 以外还有奇偶校验以及和校验（校验和），在求水平校验值时，可以使用 FNC84 - CCD 指令。

➢指令格式

CRC【P】　S.　D.　n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存作为 CRC 值生成对象的数据的软元件起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	BIN16 位
D.	保存被生成的 CRC 值的软元件编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	
n	要计算 CRC 值的 8 位数据（字节）数，或是保存数据数的软元件编号	--	D、R	K、H	

➢动作说明

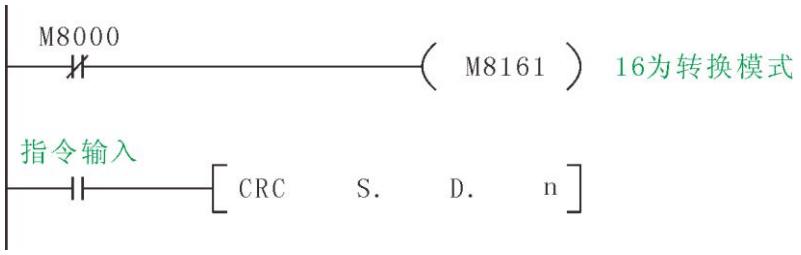
1、16 位运算

以 S.中指定的软元件位起始的 n 点 8 位数据（字节单位），对其生成 CRC 值后保存到 D.中。在这个指令中有 8 位和 16 位的转换模式，根据 M8161 的 ON/OFF 来切换转换模式。

有关各自的动作请参考下一页以后的内容。

1）16 位转换模式【M8161 = OFF】

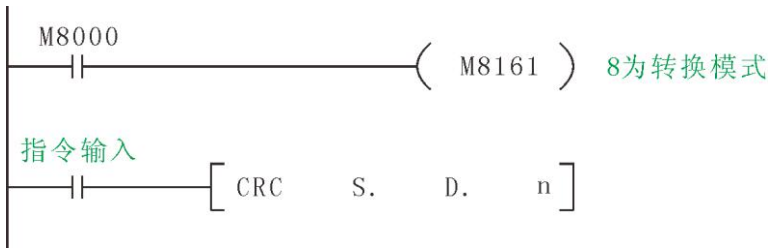
在 16 位模式下，对 S.软元件的高 8 位（字节）和低 8 位（字节）进行运算。在 D.指定的 1 点软元件的 16 位中保存运算结果。



			例如：S.＝D100，D.＝D0，n＝6		
			软元件	对象数据的内容	
				8 位	16 位
保存生产 CRC 值的对象数据的地址	S.	低字节	D100 低字节	01H	0301H
		高字节	D100 高字节	03H	
	S.+1	低字节	D101 低字节	03H	0203H
		高字节	D101 高字节	02H	
	S.+1	低字节	D102 低字节	00H	1400H
		高字节	D102 高字节	14H	
	...				
	S.+n/2+1	低字节	...		
		高字节			
保存 CRC 值的地址	D.	低字节	D0 低字节	E4H	41E4H
		高字节	D0 高字节	41H	

2）8 位转换模式【M8161＝ON】

在 8 位模式下，对 S.软元件的低 8 位（字节）进行运算。计算结果使用 D.指定的软元件开始的 2 点，在 D.中保存低 8 位（字节），在 D.+1 中保存高 8 位（字节）。



			例如：S.＝D100，D.＝D0，n＝6	
			软元件	对象数据的内容
保存生产 CRC 值的对象数据的地址	S.	低字节	D100 低字节	01H
		高字节	D100 高字节	03H

	S.+1	低字节	D101 低字节	03H
		高字节	D101 高字节	02H
	S.+1	低字节	D102 低字节	00H
		高字节	D102 高字节	14H
	...			
	S.+n/2+1	低字节	...	
		高字节		
保存 CRC 值的地址	D.	低字节	D0 低字节	E4H
		高字节	D0 高字节	41H

➢相关软元件

软元件	内容		备注
M8161	ON	CRC 指令在 8 位模式下动作	RUN —> STOP 时清除
	OFF	CRC 指令在 16 位模式下动作	

➢相关软元件

以下一些情况下会出现运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

1）S.，D.中使用的位软元件的位数指定，指定了 4 位数以外的值时，错误代码 K6706。

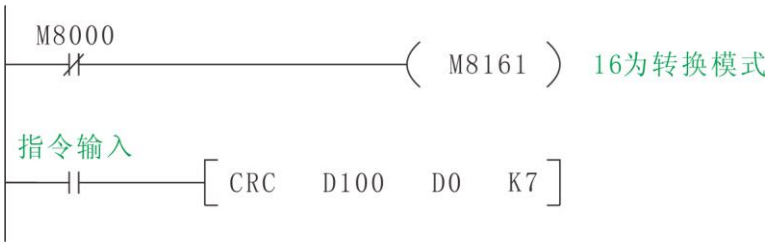
2）N 在指定范围（1~256）以外时，错误代码 K6706。

3）S.+n-1，D.+1 超出软元件范围时，错误代码 K6706。

➢程序举例

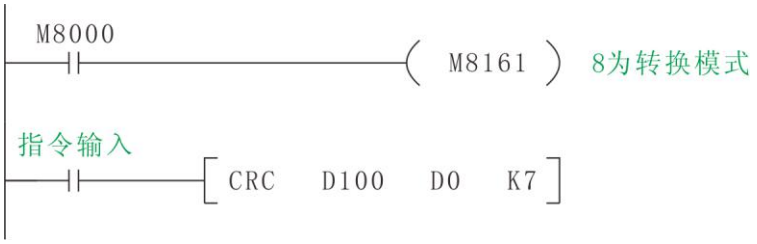
M0 为 ON 时，生成 D100~D106 中保存的 ASCII 码【0123456】的 CRC 值后，保存到 D0 中的程序。

1、16 位模式下



软元件	内容		对象数据	
保存生成 CRC 值的对象数据的地址	D100	3130H	低字节	30H
			高字节	31H
	D101	3332H	低字节	32H
			高字节	33H
	D102	3534H	低字节	34H
			高字节	35H
	D103	3736H	低字节	36H
			--	--
保存 CRC 值的地址	D10	2ACFH	低字节	CFH
			高字节	2AH

2、8 位模式下



软元件	内容	对象数据	
保存生成 CRC 值的对象数据的地址	D100	低字节	30H
	D101	高字节	31H
	D102	低字节	32H
	D103	高字节	33H
	D104	低字节	34H
	D105	高字节	35H
	D106	低字节	36H
保存 CRC 值的地址	D0	低字节	CFH
	D1	高字节	2AH

FNC189 - HCMOV：高速计算器传送

➢功能说明

传送指定的高速计数器、或是环形计数器的当前值的指令。

➢指令格式

DHCMOV S. D. n

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	作为传送源的高速计数器，或是环形计数器的软元件编号	--	C、D	--	BIN32 位
D.	传送目标的软元件编号	--	D、R	--	
n	传送后，作为清除传送源的高速计数器，或是环形计数器的当前值的指示【K1 时清除，K0 时不处理】	--	--	K、H	BIN16 位

➢动作说明

1、32 位运算（DHCMOV）

1）将 S 的高速计数器、或环形计数器的当前值传送到【D.+1，D.】中。

S.软元件		执行指令后的【D.+1，D】
高速计数器	C235~C255	高速计数器 S 的当前值【D.+1，D.】
环形计数器	D8099	D8099 —> D，在 D.+1 中保存“0”
	D8398	【D8398，D8398】的当前值【D.+1，D.】

2）传送后，根据 n 的设定值，对高速计数器、或是环形计数器的当前值执行如下所示的处理。

n 的设定	动作
K0（H0）	不清除当前值，不处理
K1（H1）	将当前值清除为 0

3、高速计数器的当前值更新时序以及 HCMOV 指令的处理

1）高速计数器的当前值更新时间

向高速计数器（C235~255）用的输入端子输入脉冲后，执行递增或是递减计数，此时，按照下表中的时序更新软元件的当前值。因此，使用一般的 MOV 指令等应用处理高速计数器的当前值时，由于使用了按照下表中的时序更新的当前值，所以会受扫描的影响。

	当前值的更新时序
硬件计数器	当执行计数器的 OUT 指令时
软件计数器	每次脉冲输入时

使用这个 DHCMOV 指令，可以按照执行指令的时序对当前值进行更新并进行传送。

2）HCMOV 指令的效果

2.1）同时使用输入中断和 DHCMOV 指令时，可以根据外部输入上升沿或下降沿的时序（接收输入中断时），对高速计数器的当前值进行读取。

2.2）在比较指令（CMP 指令/ZCP 指令/触点比较指令）的前面使用 DHCMOV 指令时，可以比较高速计数器的最新值。此外和高速计数器用的比较指令相比，还有如下的效果。

a）如果不是使用高速计数器用的比较指令比较硬件计数器的当前值，而是使用 CMP/ZCP/触点比较指令比较，则软件计数器中没有变化。

b）如果能够减少对软件计数器使用高速计数器用的比较指令的次数，则综合频率的限制会有所缓和。

c）根据高速计数器的当前值变化时时序进行比较，使输出触点（Y）改变时，请使用高速计数器用比较指令（HSCS/HSCR/HSZ 指令）。

d）HCMOV 指令的使用次数没有限制。

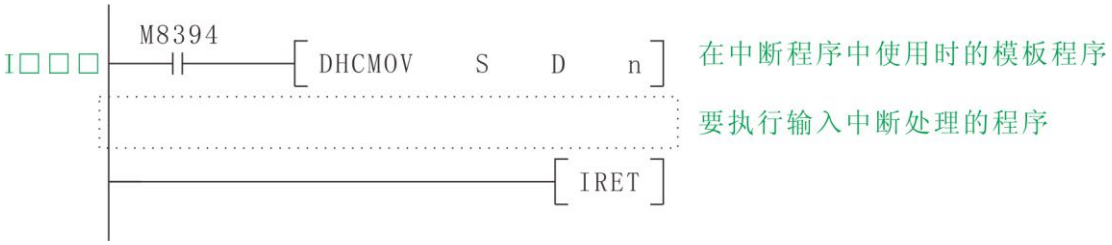
➢注意要点

在输入中断程序中编程时，需要注意一下几点。

关于输入中断用指针和输入的分配，请参考下面的 5）表格。

1、请在主程序中编写 FNC04 - EI 指令和 FNC06 - FEDN 指令。执行输入中断程序时需要。

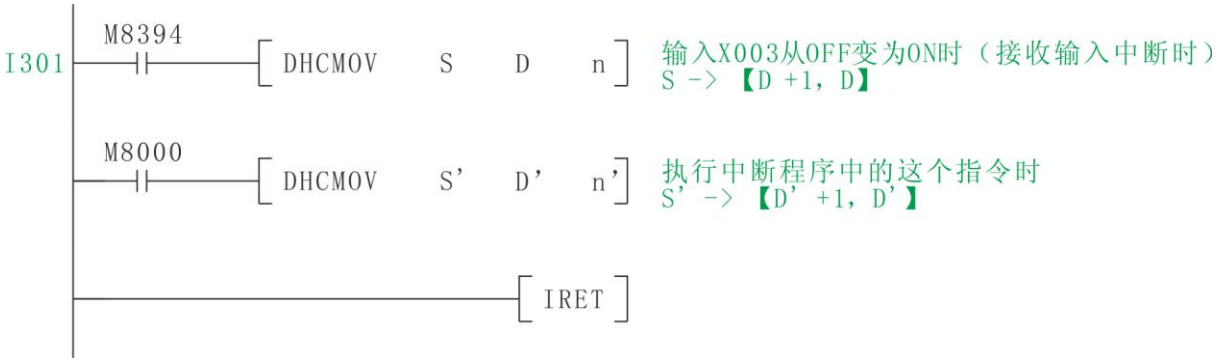
2、在输入中断程序的第 1 行中编程时，请务必编写如下的模板程序。



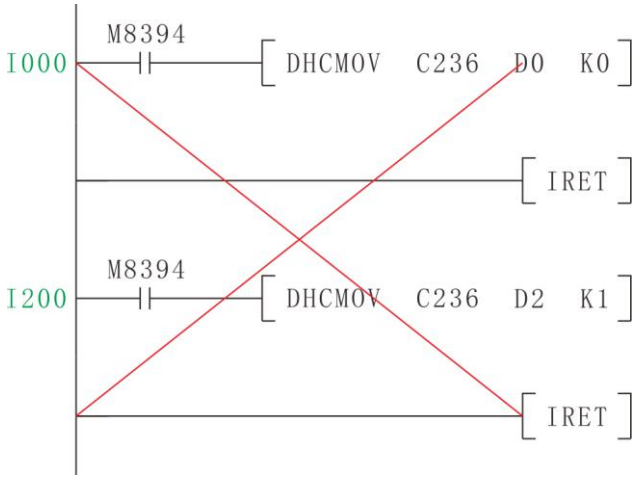
3、在 1 个输入中断程序中多次使用 DHCMOV 指令时，产生中断事件，则只有中断指针后的第 1 个指令被执行，然后处理中断程序。

第 2 个以后的 DHCMOV 指令，与通常的中断处理相同，在执行指令时处理。

第 2 个以后的指令触点中，请勿使用 M8394。



4、在多个输入中断程序中，不可以对同一个计数器使用 DHCMOV 指令。



5、根据中断禁止标志位（下表），输入中断处于禁止状态下，其输入中断程序不被处理，因此 DHCMOV 指令，不执行。

中断禁止标志位	对应的中断指针	与中断指针相对应的输入编号	备注
M8050	I000，I001	X000	RUN —> STOP 时清除
M8051	I100，I101	X001	
M8052	I200，I201	X002	
M8053	I300，I301	X003	
M8054	I400，I401	X004	
M8055	I500，I501	X005	

4）通过使用中断禁止标志位 M8050~M8055 以外（执行 DI 指令后，执行 EI 指令前）使处于输入中断禁止的状态下，此时如产生了输入中断，则立即执行 DHCMOV 指令，但是中断程序的执行被保留。执行中断程序。

➢出错

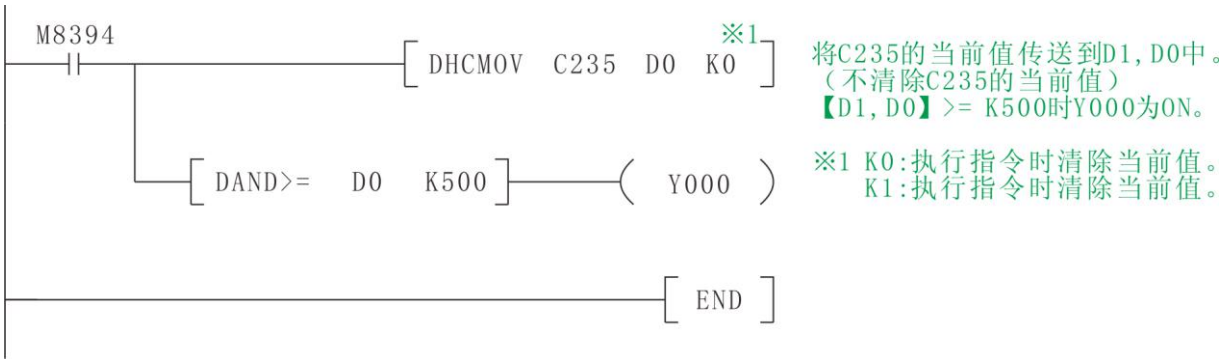
以下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

S.，【D.+1，D.】中指定的软元件为对象以外。错误代码 K6705。

➢举例

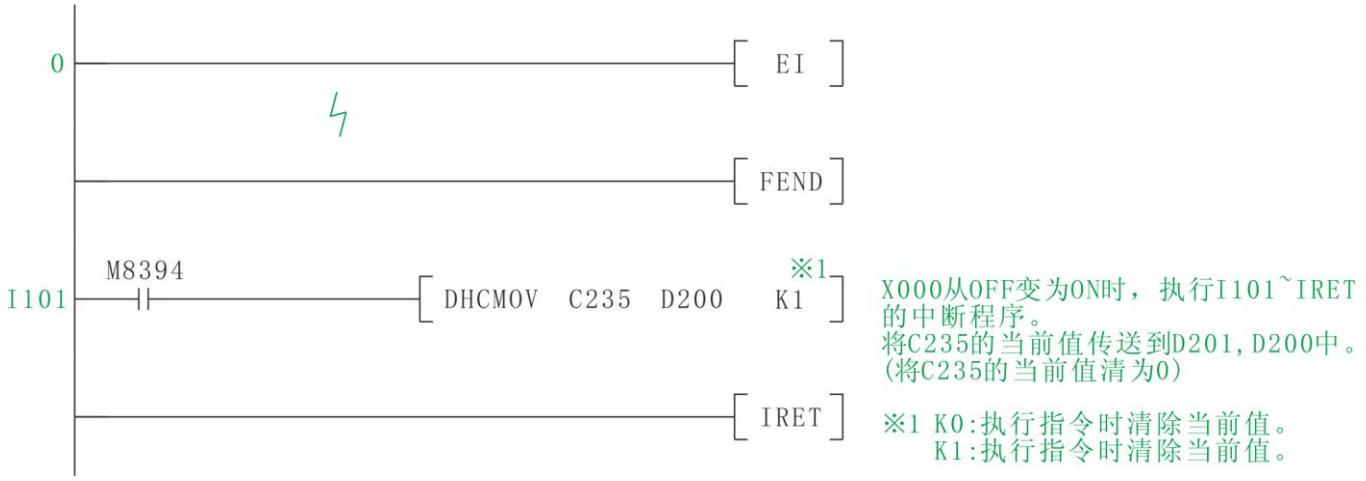
1、程序举例 1

每个运算周期比较高速计数器 C235 的当前值，当为 K500 以上时，使输出 Y000 动作的程序。（不清除 C235 的当前值的情况下）



2、程序举例 2

当 X001 从 OFF 变为 ON 时，将 C235 的当前值传送到 D201，D200 中，并清除当前值的程序。



7-17 数据块处理 - FNC190~FNC199

FNC192 - BK+: 数据块加法运算

➢功能说明

数据块的 BIN 加法运算的指令。

➢指令格式

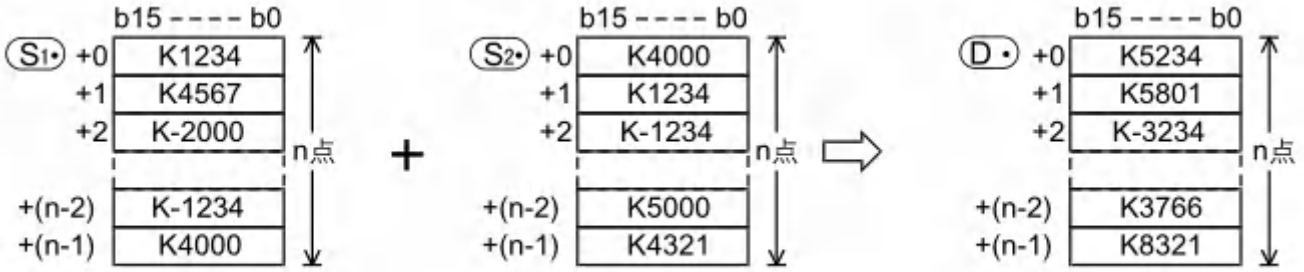
【D】BK+【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存执行加法运算的数据的软元件编号	--	T、C、D、R、修饰	--	BIN32/16 位
S2.	执行加法运算的常数，或是保存执行加法运算的数据的软元件起始编号	--	T、C、D、R、修饰	K、H	
D.	保存运算结果的软元件起始编号	--	T、C、D、R、修饰	--	
n	数据的个数	--	D、R	K、H	

➢动作说明

1、16 位运算（BK+/BKP+）

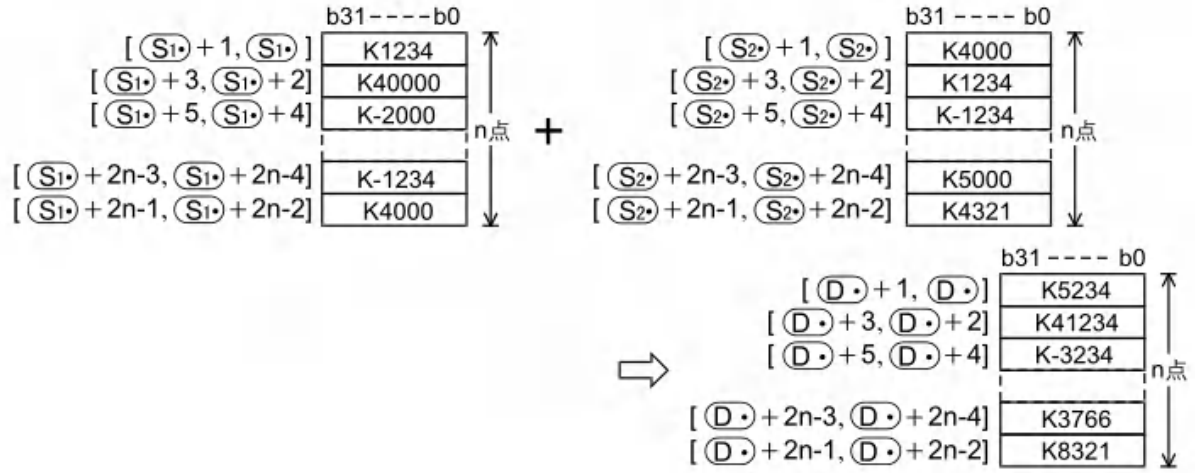
1）将 S1.开始的 n 点 16 位数据和 S2.开始的 n 点 16 位数据（BIN）进行加法运算后，将运算结果保存到 D.开始 n 点中。



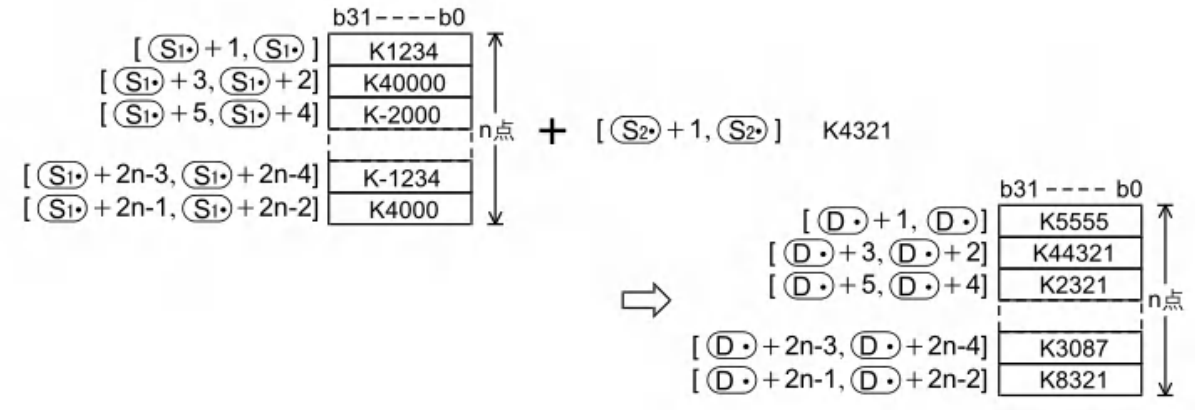
2) 可以在S2•中直接指定－32,768～32,767（16位）的常数。

2、32 位运算（DBK+/DBKP+）

1）将【S1.+1，S1.】开始的 2n 点 32 位数据和【S2.+1，S2.】开始的 2n 点 32 位数据（BIN）进行加法运算后，将运算结果保存到【D.+1，D.】开始 2n 点中。



2）可以在【S2.+1，S2.】中直接指定-2147483648~2147483647（32 位）的常数。



➢相关指令

FNC193 - BK-，数据块的减法运算指令。

➢注意要点

1、运算结果中产生了下溢出，上溢出时，如下所示。
此时，进位标志位不置 ON。

16位运算时

K32767(H7FFF) + K2(H0002) → K-32767(H8001)
K-32767(H8000) + K-2(HFFFE) → K32766(H7FFE)

32位运算时

K2,147,483,647(H7FFFFFFF) + K2(H00000002) → K-2,147,483,647(H80000001)
K-2,147,483,648(H80000000) + K-2(HFFFFFFFE) → K2,147,483,646(H7FFFFFFE)

➢出错

一下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

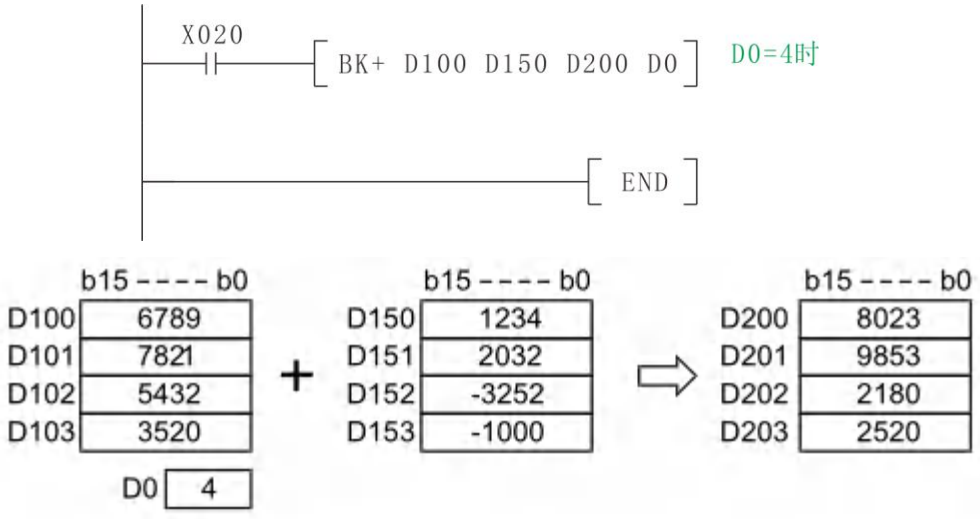
1）S1.、S2.、D.开始的 n 点（32 位运算时为 2n 点）软元件超出相应的软元件范围时，错误代码 K6706；

2）S1.开始的 n 点软元件和 D.开始的 n 点软元件重复时（32 位运算时为 2n 点），错误代码 K6706；

3）S2.开始的 n 点软元件和 D.开始的 n 点软元件重复时（32 位运算时为 2n 点），错误代码 K6706；

➢举例

当 X020 为 ON 时，将从 D100 开始的软元件数据（软元件点数为 D0 中所保存的数值），和从 D150 开始的软元件数据（软元件点数为 D0 中所保存的数值）进行加法运算，并将其结果保存到 D200 以后中的程序。



FNC193 - BK-：数据块减法运算

➢功能说明

数据块的 BIN 减法运算的指令。

➢指令格式

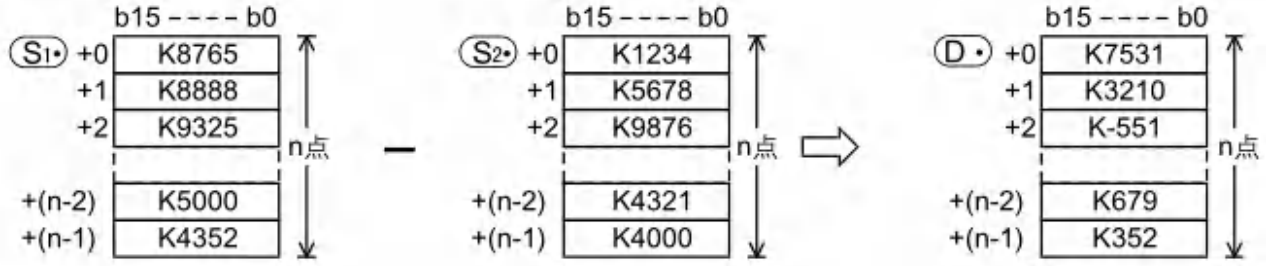
【D】BK-【P】 S. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存执行减法运算的数据的软元件编号	--	T、C、D、R、修饰	--	BIN32/16 位
S2.	执行减法运算的常数，或是保存执行加法运算的数据的软元件起始编号	--	T、C、D、R、修饰	K、H	
D.	保存运算结果的软元件起始编号	--	T、C、D、R、修饰	--	
n	数据的个数	--	D、R	K、H	

➢动作说明

1、16 位运算（BK-/BKP-）

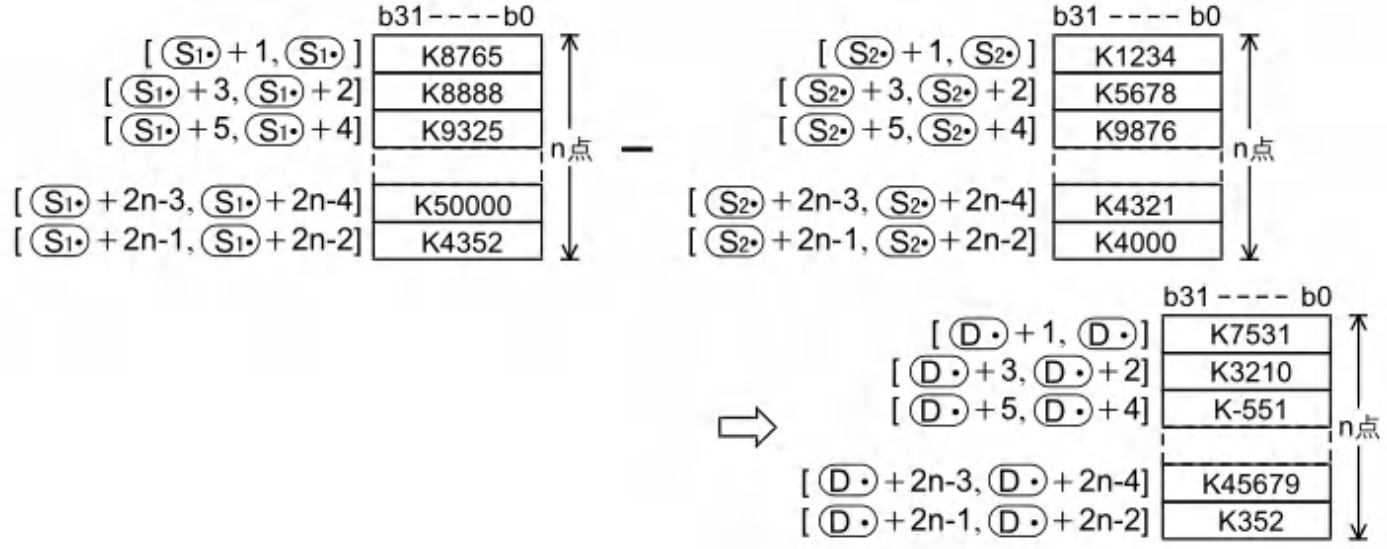
1）将 S1.开始的 n 点 16 位数据和 S2.开始的 n 点 16 位数据（BIN）进行减法运算后，将运算结果保存到 D.开始 n 点中。



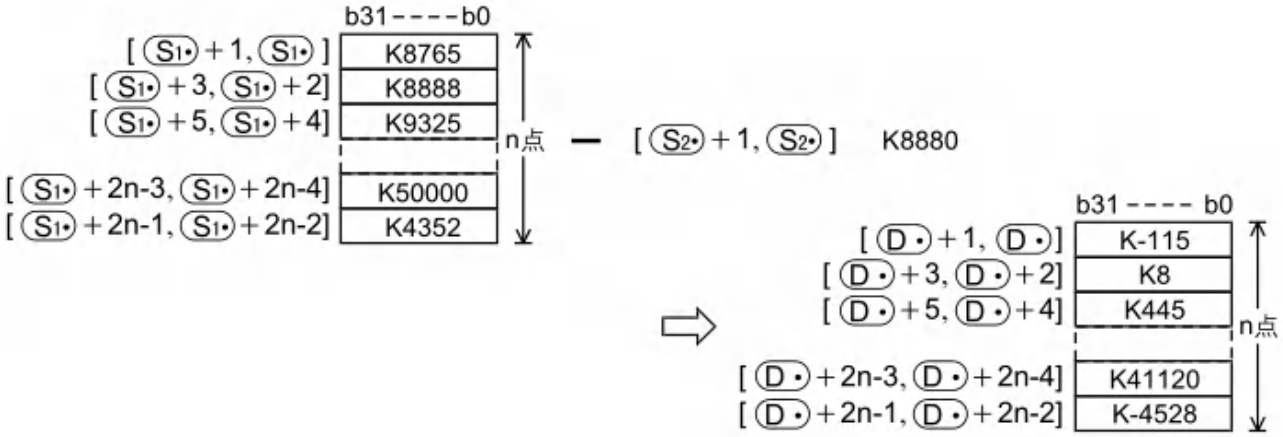
2）可以在在 S2.中直接指定-32768~32767（16 位）的常数。

2、32 位运算（DBK-/DBKP-）

1）将【S1.+1，S1.】开始的 2n 点 32 位数据和【S2.+1，S2.】开始的 2n 点 32 位数据（BIN）进行减法运算后，将运算结果保存到【D.+1，D.】开始 2n 点中。



2）可以在【S2.+1，S2.】中直接指定-2147483648~2147483647（32 位）的常数。



➢相关指令

FNC192 - BK+，数据块的加法运算指令。

➢注意要点

2、运算结果中产生了下溢出，上溢出时，如下所示。

此时，进位标志位不置 ON。

16位运算时

K32767(H7FFF) + K2(H0002) → K-32767(H8001)

K-32767(H8000) + K-2(HFFFE) → K32766(H7FFE)

32位运算时

K2,147,483,647(H7FFFFFFF) + K2(H00000002) → K-2,147,483,647(H80000001)

K-2,147,483,648(H80000000) + K-2(HFFFFFFFE) → K2,147,483,646(H7FFFFFFE)

➢出错

一下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

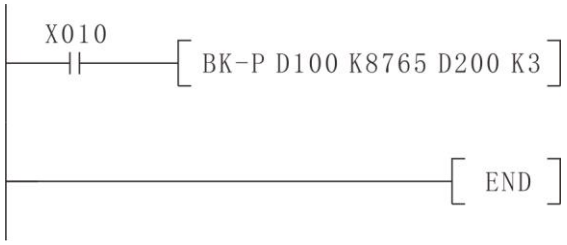
1）S1.、S2.、D.开始的 n 点（32 位运算时为 2n 点）软元件超出相应的软元件范围时，错误代码 K6706；

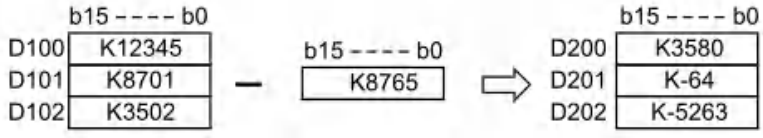
2）S1.开始的 n 点软元件和 D.开始的 n 点软元件重复时（32 位运算时为 2n 点），错误代码 K6706；

3）S2.开始的 n 点软元件和 D.开始的 n 点软元件重复时（32 位运算时为 2n 点），错误代码 K6706；

➢举例

当 X010 为 ON 时，将从 D100 开始的 3 点数据和常数 8765 相减后，并将其结果保存到 D200 以后中的程序。





FNC194 - BKCMP=：数据块的比较 S1. = S2.

FNC195 - BKCMP>：数据块加法运算 S1. > S2.

FNC196 - BKCMP<：数据块加法运算 S1. < S2.

FNC197 - BKCMP<>：数据块加法运算 S1. <> S2.

FNC198 - BKCMP<=：数据块加法运算 S1. <= S2.

FNC199 - BKCMP>=：数据块加法运算 S1. >= S2.

➢功能说明

这些指令是按照各个指令的比较条件来比较数据块。

➢指令格式（FNC194~FNC199 通用）

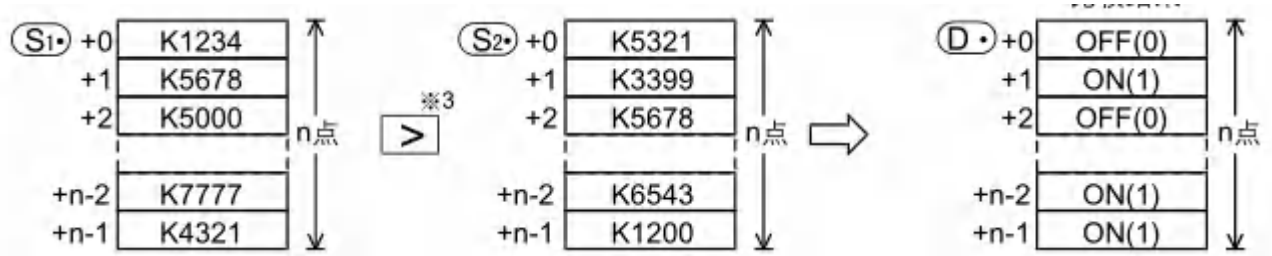
【D】BKCMP□【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	比较值或是保存比较值的软元件编号	--	T、C、D、R、修饰	K、H	BIN32/16 位
S2.	保存比较源数据的软元件起始编号	--	T、C、D、R、修饰	--	
D.	保存比较结果的软元件起始编号	Y、M、S、D□.b	修饰	--	位
n	要比较的数据数	--	D、R	K、H	BIN16/32 位

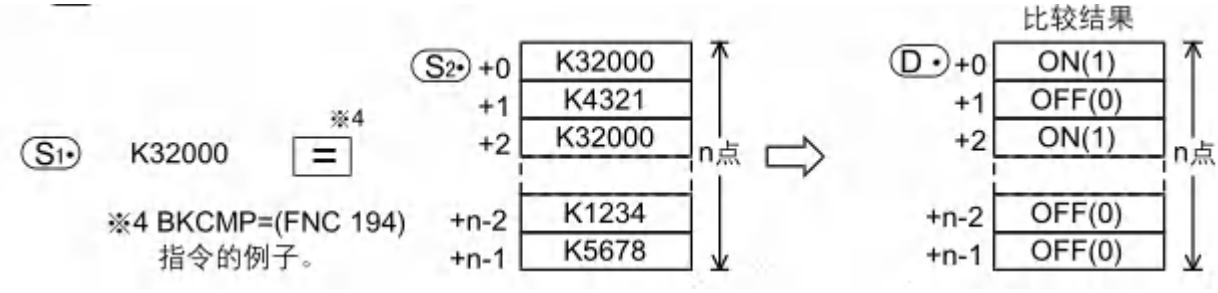
➢动作说明

1、16 位运算（BKCMP=、>、<、<>、<=、>=/BKCMPP=、>、<、<>、<=、>=）

1）将 S1.开始的 n 点 16 位数据和 S2.开始的 n 点 16 位数据（BIN）比较后，将运算结果保存到 D.开始 n 点中。



2）可以在 S1 中直接指定常数。



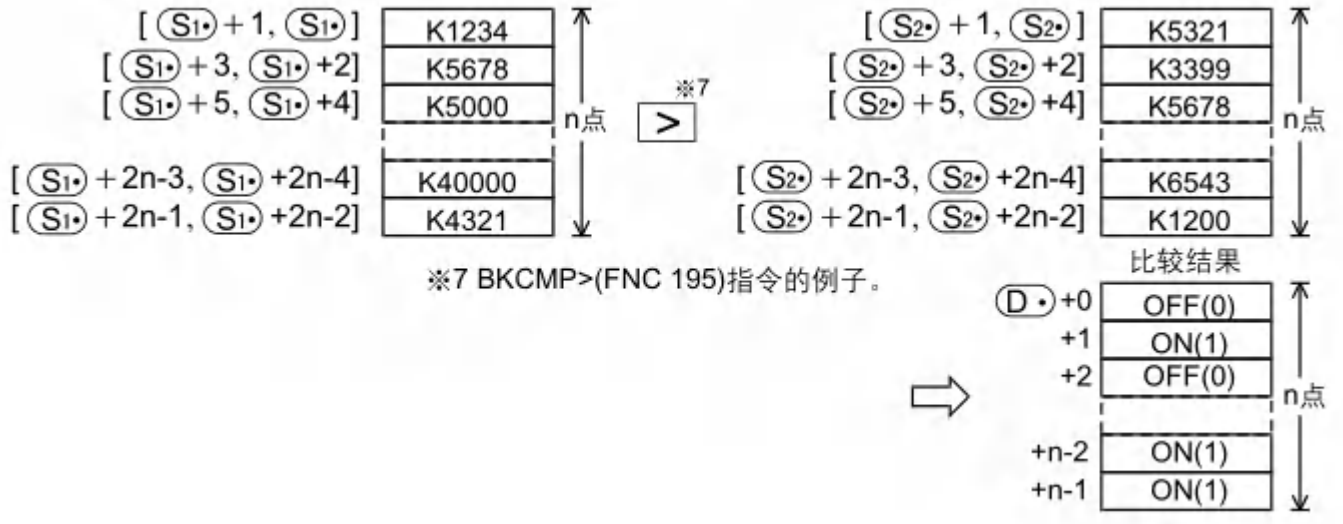
3）各个指令的比较结果如下所示。

指令	比较结果 ON 的条件	比较结果 ON 的条件
FNC194 - BKCMP=	S1. = S2.	S1. ≠ S2.
FNC194 - BKCMP=	S1. > S2.	S1. ≤ S2.
FNC194 - BKCMP=	S1. < S2.	S1. ≥ S2.
FNC194 - BKCMP=	S1. ≠ S2.	S1. = S2.
FNC194 - BKCMP=	S1. ≤ S2.	S1. > S2.
FNC194 - BKCMP=	S1. ≥ S2.	S1. < S2.

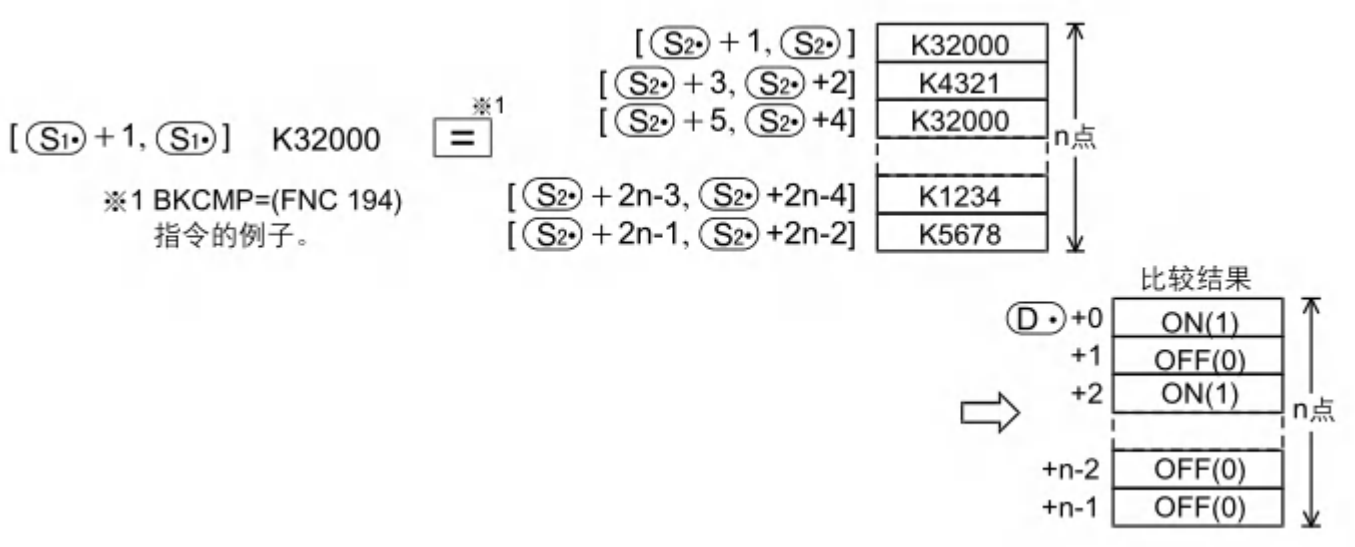
4）D.开始的 n 点的比较结果都为 ON 时，M8090(块比较信号)为 ON。

2、32 位运算（DBKCMP=、>、<、<>、<=、>=/DBKCMPP=、>、<、<>、<=、>=）

1）将【S1.+1，S1.】开始的 2n 点 32 位数据和【S2.+1，S2.】开始的 2n 点 32 位数据（BIN）比较后，将比较结果保存到【D.+1，D.】开始 2n 点中。



2）可以在【S2.+1，S2.】中直接指定常数。



3）各个指令的比较结果如下所示。

指令	比较结果 ON 的条件	比较结果 ON 的条件
FNC194 - BKCMP=	【S1.+1，S1.】=【S2.+1，S2.】	【S1.+1，S1.】≠【S2.+1，S2.】
FNC194 - BKCMP=	【S1.+1，S1.】>【S2.+1，S2.】	【S1.+1，S1.】≤【S2.+1，S2.】
FNC194 - BKCMP=	【S1.+1，S1.】<【S2.+1，S2.】	【S1.+1，S1.】≥【S2.+1，S2.】
FNC194 - BKCMP=	【S1.+1，S1.】≠【S2.+1，S2.】	【S1.+1，S1.】=【S2.+1，S2.】
FNC194 - BKCMP=	【S1.+1，S1.】≤【S2.+1，S2.】	【S1.+1，S1.】>【S2.+1，S2.】
FNC194 - BKCMP=	【S1.+1，S1.】≥【S2.+1，S2.】	【S1.+1，S1.】<【S2.+1，S2.】

4）D.开始的 n 点的比较结果都为 ON 时，M8090(块比较信号)为 ON。

➢相关软元件

软元件	名称	内容
M8090	块比较信号	数据块指令的比较结果都为 ON 时变为 ON。 FNC194 - BKCMP=、FNC194 - BKCMP>、FNC194 - BKCMP<、 FNC194 - BKCMP<>、FNC194 - BKCMP<=、FNC194 - BKCMP>=

➢注意要点

使用 32 位高速计数器时。32 位高速计数器（C200~C255）的比较，必须在 32 位运算（DBKCMP=、DBKCMP>、DBKCMP<、DBKCMP<>、DBKCMP<=、DBKCMP>=）下进行比较。

在 16 位运算（BKCMP=、BKCMP>、BKCMP<、BKCMP<>、BKCMP<=、BKCMP>=）下指定，则发生运算出错，错误代码 K6705。

➢出错

一下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

1）S1.、S2.开始的 n 点（32 位运算时为 2n 点）软元件超出相应的软元件范围时，错误代码 K6706；

2）D.开始的 n 点软元件超出了相应软元件的范围时，错误代码 K6706；

3）D.指定为 D□.b 时，D.的数据寄存器和 S1.开始的 n 点软元件重复时（32 位运算时为 2n 点）软元件范围重复时，错误代码 K6706；

4）D.指定为 D□.b 时，D.的数据寄存器和 S2.开始的 n 点软元件重复时（32 位运算时为 2n 点）软元件范围重复时，错误代码 K6706；

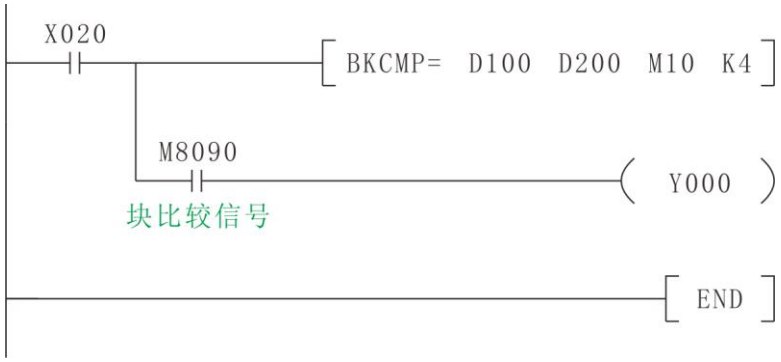
5）16 位运算中，在 S1.、S2.中指定了 32 位计数器（C200~C255）时，错误代码 K6705

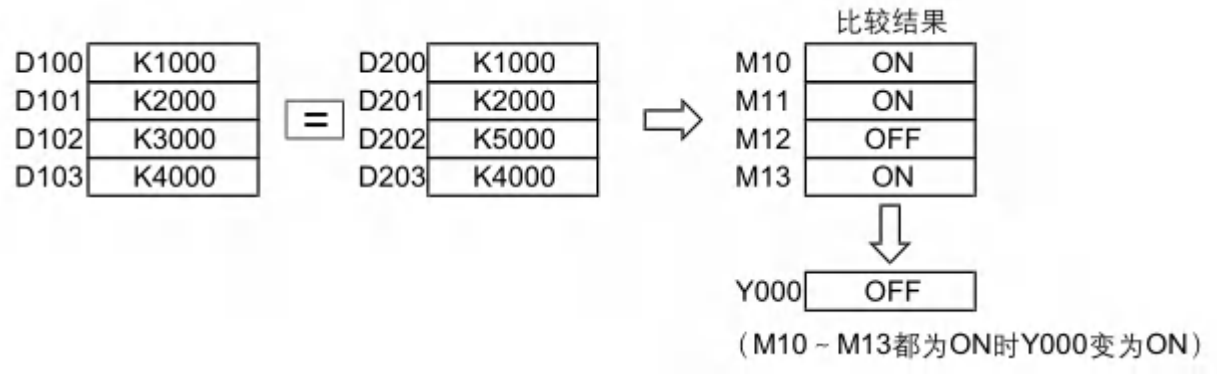
32 位计数器请用 32 运算（DBKCMP=、DBKCMP>、DBKCMP<、DBKCMP<>、DBKCMP<=、DBKCMP>=）指令进行比较。

➢**举例**

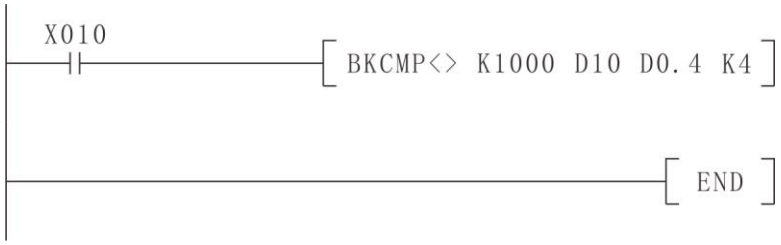
1）当 X020 为 ON 时，使用 FNC194 - BKCMP=指令对 D100 开始的 4 点 16 位数据（BIN），D200 开始的 4 点 16 位数据（BIN）进行比较，并将其结果保存到 M10 开始的 4 点软元件中的程序。

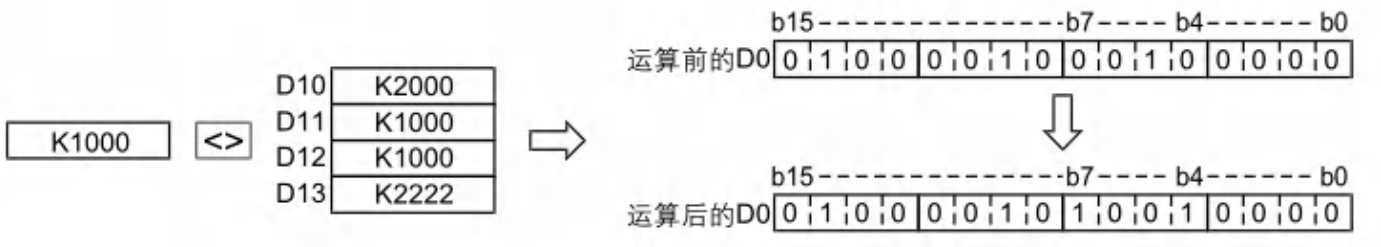
此外，比较结果（M10 开始的 4 点）全部位 ON 时，Y000 置 ON。





3）当 X010 为 ON 时，将常数 K1000 和 D10 开始的 4 点数据进行比较，然后将其结果保存为 D0 的 b4~b7 中的程序。





7-18 数据块处理 - FNC200~FNC209

FNC200 - STR：BIN 转字符串

➢ 功能说明

将 BIN 数据转换成字符串（ASCII 码）的指令。
还有将浮点数数据转换成字符串的 FNC116 - ESTR 指令。

➢ 指令格式

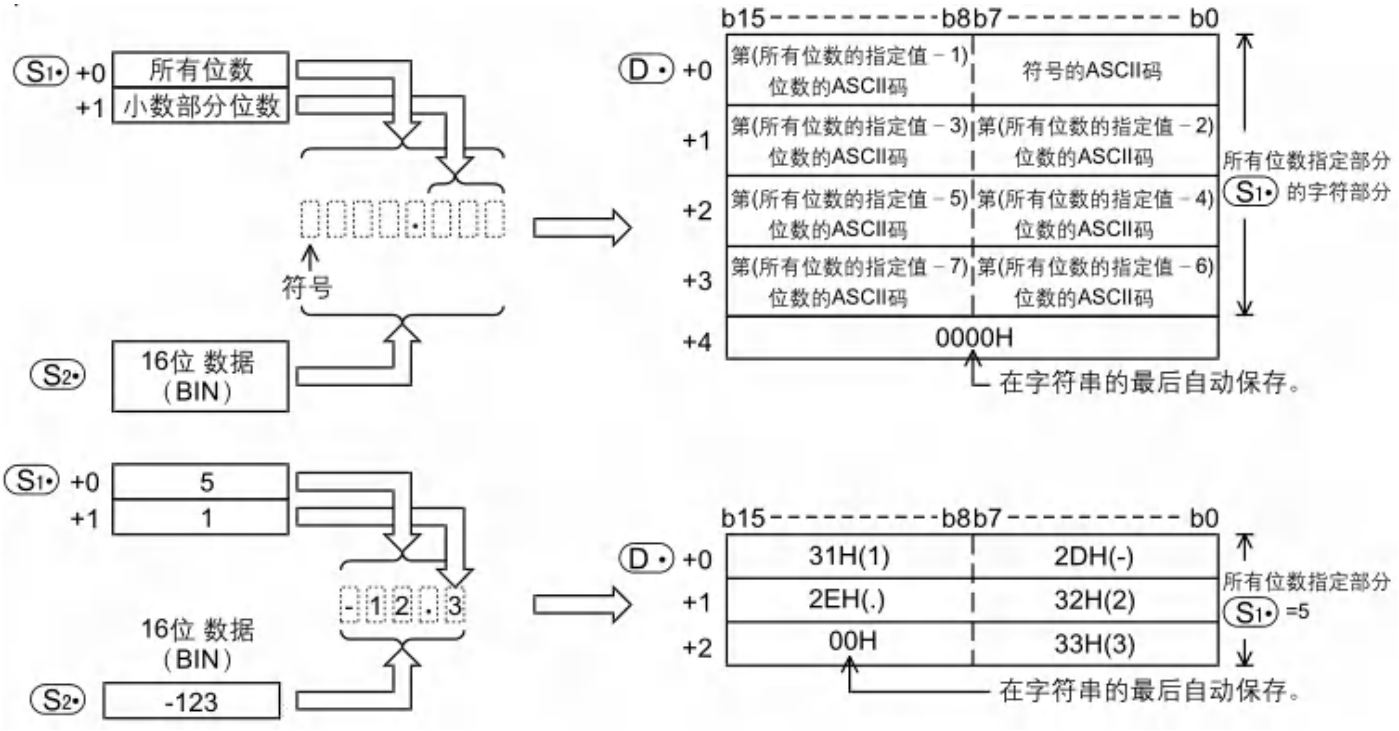
【D】STR【P】　S1.　S2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存要转换数值的位数的软元件起始编号	--	T、C、D、R、修饰	--	BIN16 位
S2.	保存要转换的 BIN 数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
D.	保存已转换的字符串的软元件起始编号	--	T、C、D、R、修饰	--	字符串

➢ 动作说明

1、16 位运算（STR/STRP）

1.1、将 S2.的 16 位数据（BIN）,在所有位数 S1.、小数点部分位数 S1.+1 指定的位置中加上小数点后，转换成字符串，保存到 D.开始的软元件中。



1.2、在 2~8 位数的范围内设定所有位数 S1.。

1.3、在 0~5 位数的范围内设定小数部分位数 S1.+1。但是，请设定为小数部分位数≤（所有位数-3）。

1.4、要转换的 16 位数据（BIN）S2.的值在-32768~32767 的范围内。

1.5、转换后的字符串数据会如下所示的保存到 D.开始的软元件编号中。

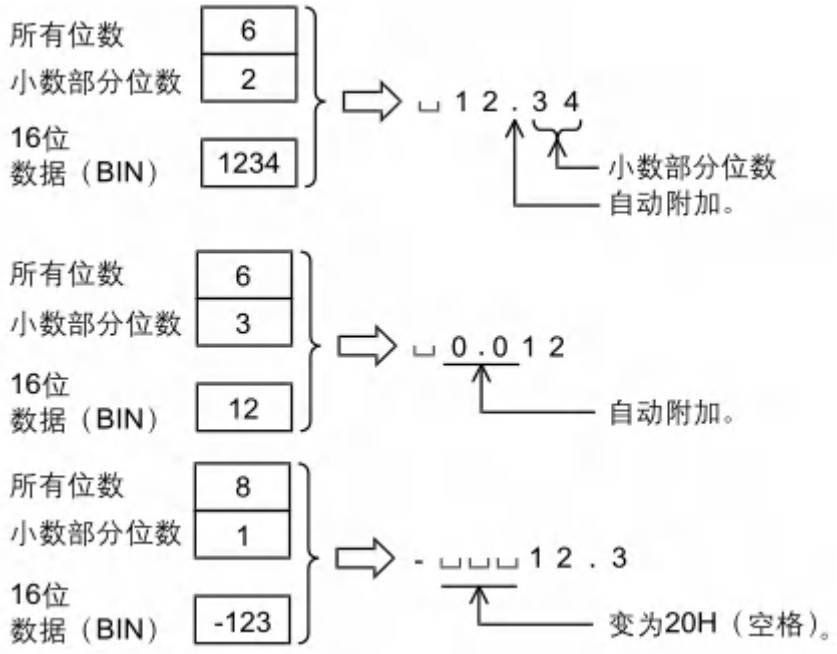
1）在符号中，16 位数据（BIN）S2.为正时保存“空格（20H）”,为负时保存“-（2DH）”。

2）在小数部分位数 S1.+1 中设定为“0”以外的数字时，会自动在小数部分位数+1 位数的位置上加上小数点“.(2EH)”。小数部分位数 S1+1 为“0”时，不附加小数点。

3）与 S2 的 16 位数据（BIN）的位数相比，S1.+1 的小数部分位数较多时，会自动向右对齐，在左边附加“0（30H）”后进行转换。

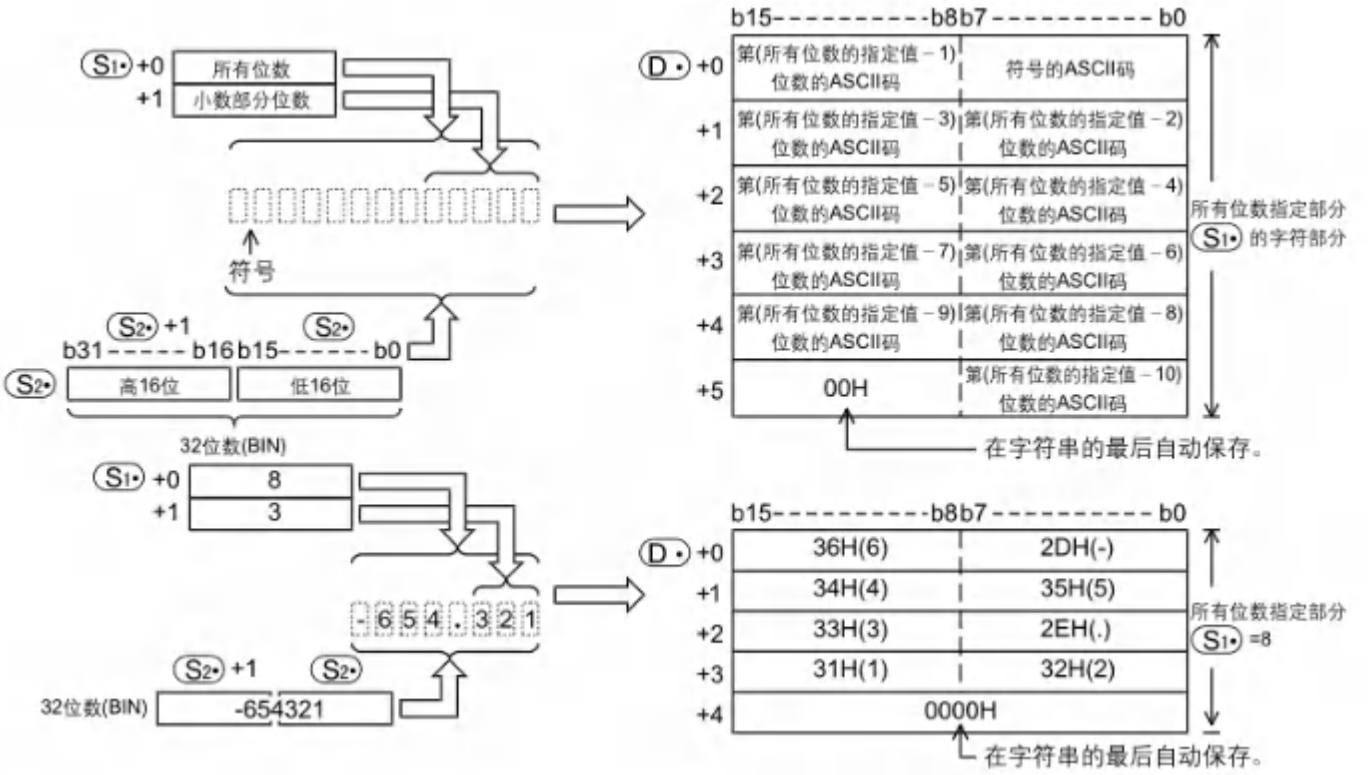
4）如果除去小数点和符号以外，所有位数 S1.的位数多于 S2.的 16 位数据（BIN）的位数时，在符号和数值之间保存“空格（20H）”。此外 S2 的 16 位数据（BIN）的位数较多时，会出错。

5）在已转换的字符串的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。总位数为偶数时，在保存末尾字符软元件的后一个软元件中保存“00H”。此外当为奇数位数时，在保存末尾字符的软元件的高字节（8 位）中保存“00H”。



2、32 位运算（DSTR/DSTRP）

2.1、将【S2.+1，S2.】的 32 位数据（BIN）,在所有位数 S1.、小数点部分位数 S1.+1 指定的位置中加上小数点后，转换成字符串，保存到 D.开始的软元件中。



2.2、在 2~13 位数的范围内设定所有位数 S1.。

2.3、在 0~10 位数的范围内设定小数部分位数 S1.+1。但是，请设定为小数部分位数≤（所有位数-3）。

2.4、要转换的 32 位数据（BIN）S2.的值在-2147483648~2147483647 的范围内。

2.5、转换后的字符串数据会如下所示的保存到 D.开始的软元件编号中。

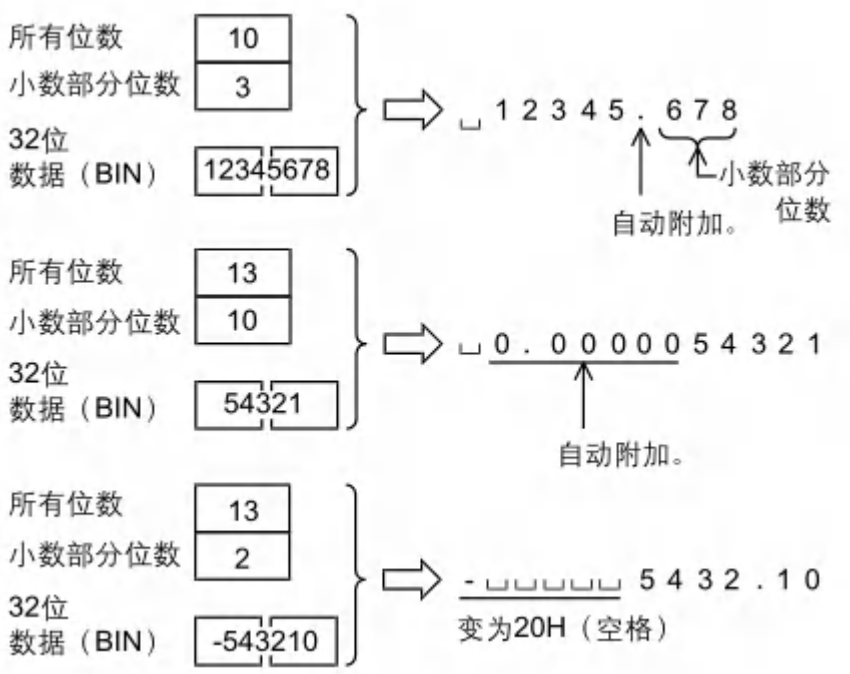
1）在符号中，32 位数据（BIN）S2.为正时保存“空格（20H）”,为负时保存“-（2DH）”。

2）在小数部分位数 S1.+1 中设定为“0”以外的数字时，会自动在小数部分位数+1 位数的位置上加上小数点“.(2EH)”。小数部分位数 S1+1 为“0”时，不附加小数点。

3）与 S2 的 32 位数据（BIN）的位数相比，S1.+1 的小数部分位数较多时，会自动向右对齐，在左边附加“0（30H）”后进行转换。

4）如果除去小数点和符号以外，所有位数 S1.的位数多于 S2.的 32 位数据（BIN）的位数时，在符号和数值之间保存“空格（20H）”。此外 S2 的 32 位数据（BIN）的位数较多时，会出错。

5）在已转换的字符串的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。总位数为偶数时，在保存末尾字符软元件的后一个软元件中保存“00H”。此外当为奇数位数时，在保存末尾字符的软元件的高字节（8 位）中保存“00H”。



➢ 相关指令

名称	内容
FNC116 - ESTR	将 2 进制浮点数数据转换成指定位数的字符串（ASCII 码）的指令
FNC117 - EVAL	将字符串（ASCII 码）数据转换 2 进制浮点数数据的指令
FNC201 - VAL	将字符串（ASCII 码）数据转换成 BIN 数据的指令

➢ 出错

一下一些情况会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

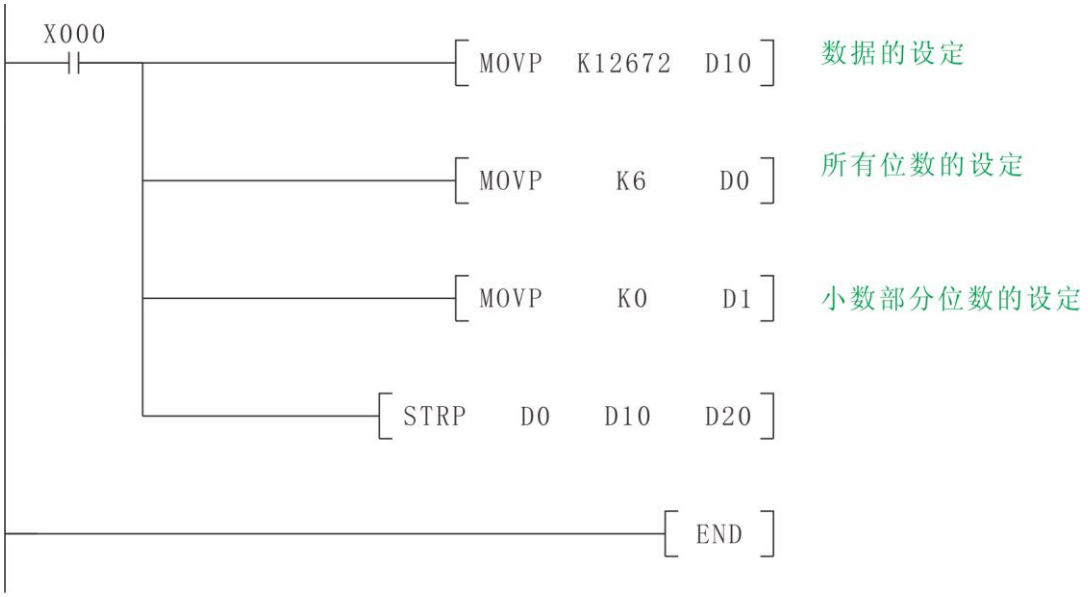
1）所有位数 S1.为如下所示的范围以外时。错误代码 K6706。16 位预算时：2≤S1.≤8
32 位预算时：2≤S1.≤13；

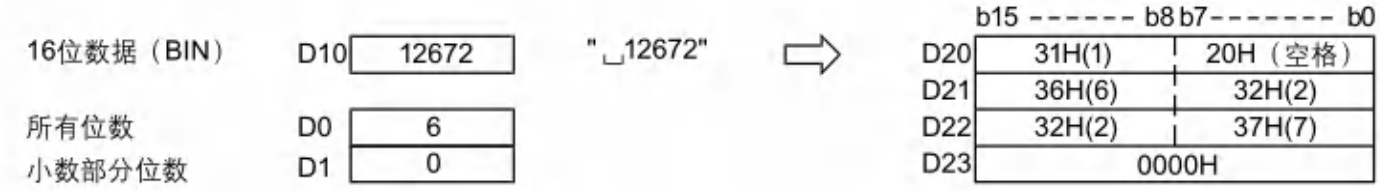
2）小数部分位数 S1+1.为如下所示的范围以外时。错误代码 K6706。16 位预算时：2≤S1.≤5
32 位预算时：2≤S1.≤10；

3）保存字符串的 D.以后的软元件超出了相应的软元件范围时，错误代码 K6706。

➢ **程序举例**

当 X000 为 ON 时，根据 D0、D1 的位数指定，将 D10 中保存的 BIN 数据（16 位）转换成字符串，然后保存到 D20~D23 中的程序。





FNC201 - VAL：字符串转 BIN

➢ **功能说明**

将字符串（ASCII 码）转成 BIN 数据的指令。

还有将字符串（ASCII 码）转换成浮点数数据的 FNC117 - EVAL 指令。

➢ **指令格式**

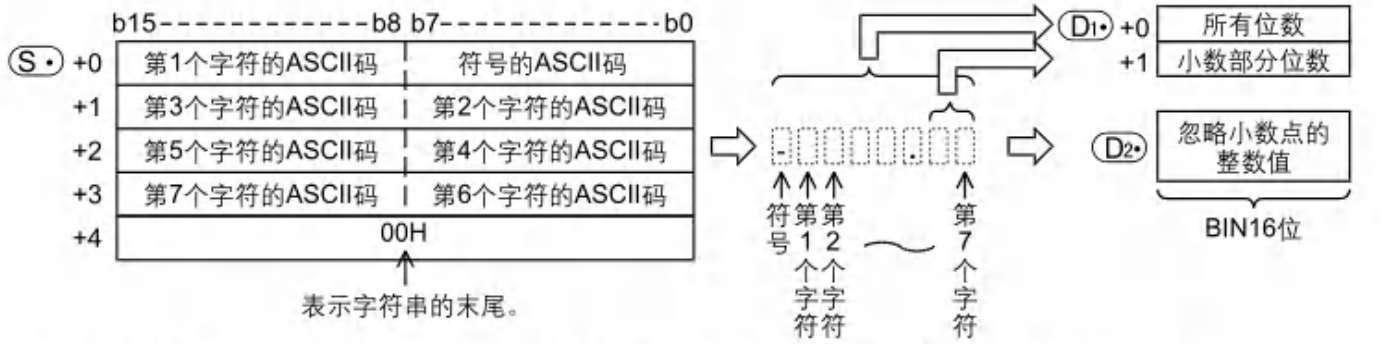
【D】VAL【P】　S.　D1.　D2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要转换BIN数值的字符串的软元件起始编号	--	T、C、D、R、修饰	--	字符串
D1.	保存已经转换的BIN数据位数的软元件编号	--	T、C、D、R、修饰	--	BIN16/32 位
D2.	保存已转换的BIN数据的软元件起始编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	BIN16/32 位

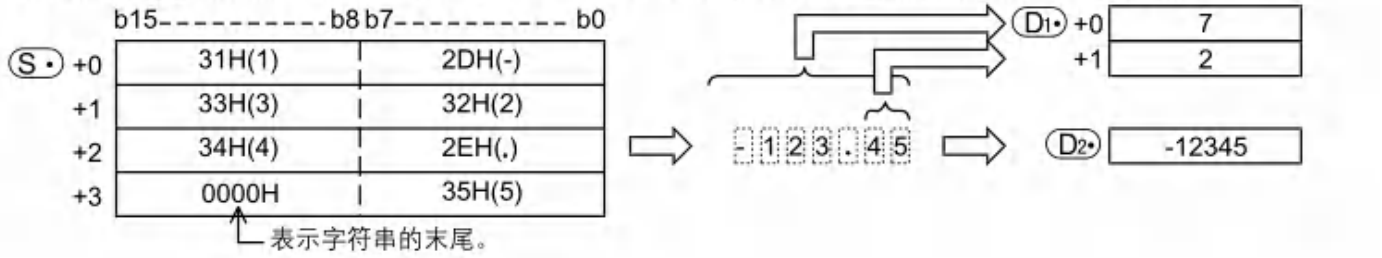
➢ 动作说明

1、16 位运算（VAL/VALP）

1.1、将 S.开始的软元件中保存的字符串转换成 16 位数据（BIN）,然后将所有位数保存到 D1.中，将小数点部分 D.+1 中，将 BIN 数据保存到 D2.中。从字符串转换成 BIN 时，以字节为单位将 S.开始到保存“00H”的软元件编号为止的数据，作为字符串进行处理。



例如，在 S· 开始的软元件中指定了“－123.45”的字符串时，D1·、D2· 中如下所示保存。



1.2、要转换的字符串数据

1）字符串的字符数，忽略小数点时的数值范围

	内容
所有字符（位）数	2~8 个字符
小数部分的字符（位）数	0~5 个字符，但是【所有位数-3】以下
忽略小数点时的数值范围	-32768~32767 例如：“123.45” —> “12345”

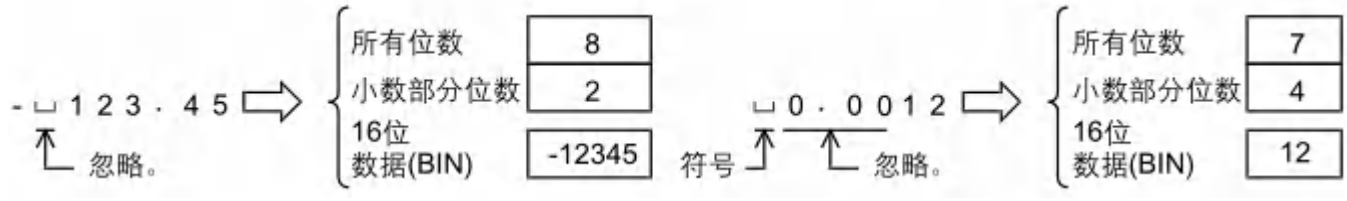
2）在要转换的字符中使用的字符种类

		字符的种类
符号	正的数值	“空格（20H）”
	负的数值	“-（20H）”
小数点		“。（2EH）”
数字		“0（30H）” ~ “9（39H）”

1.3、D1.中保存所有位数。所有位数，就是所有的字符数（包括数字、符号、小数点）。

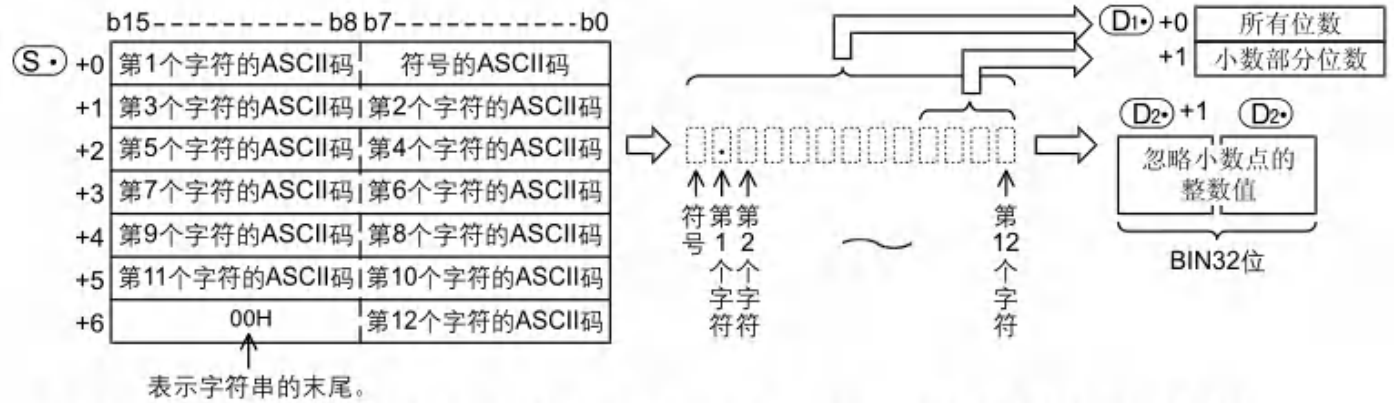
1.4、D1+1 中保存小数部分的位数。小数部分的位数为小数点“.2EH”以后的字符数。

1.5、在 D2 中，无视小数点，将字符串转换成 16 位的数据（BIN）。但是在 S.开始的字符串中，符号和最初的“0”以外的数字之间的“空格（20H）”或是“0（30H）”被忽略，而转换成 16 位数据（BIN）。

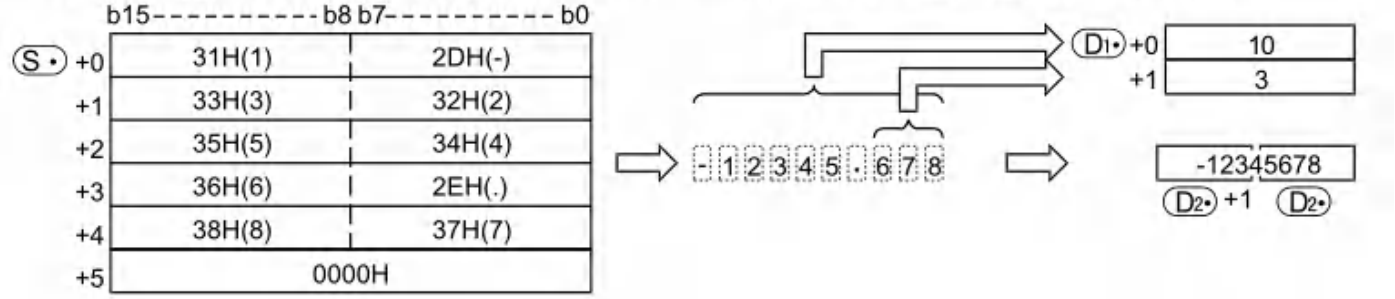


2、32 位运算（DSTR/DSTRP）

2.1、将 S.开始的软元件中保存的字符串转换成 32 位数据（BIN）,然后将所有位数保存到 D1.中，将小数点部分 D.+1 中，将 BIN 数据保存到【D2.+1，D2.】中。从字符串转换成 BIN 时，以字节为单位将 S.开始到保存“00H”的软元件编号为止的数据，作为字符串进行处理。



例如，在 S· 开始的软元件中指定了“－12345.678”的字符串时，D1·、D2·中如下所示保存。



2.2、要转换的字符串数据

1）字符串的字符数，忽略小数点时的数值范围

	内容
所有字符（位）数	2~13 个字符
小数部分的字符（位）数	0~10 个字符，但是【所有位数-3】以下
忽略小数点时的数值范围	-2147483648～2147483647 例如：“12345.678” —> “12345678”

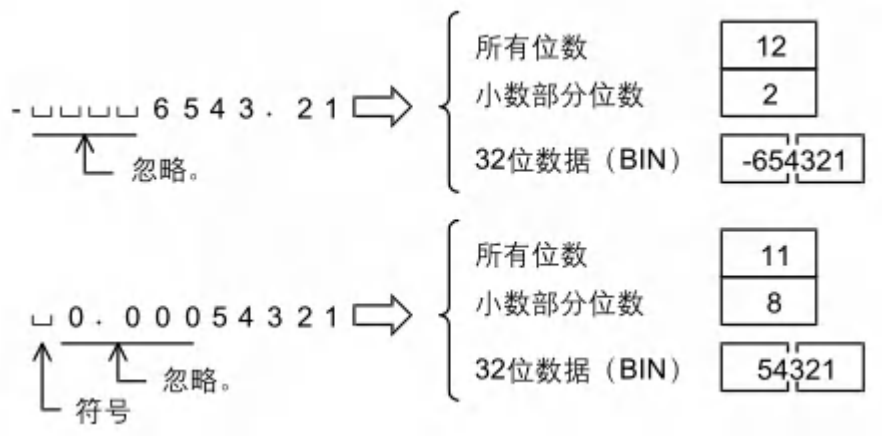
2）在要转换的字符中使用的字符种类

		字符的种类
符号	正的数值	“空格（20H）”
	负的数值	“-（20H）”
小数点		“。（2EH）”
数字		“0（30H）” ~ “9（39H）”

1.3、D1.中保存所有位数。所有位数，就是所有的字符数（包括数字、符号、小数点）。

1.4、D1+1 中保存小数部分的位数。小数部分的位数为小数点“.2EH”以后的字符数。

1.5、在 D2 中，无视小数点，将字符串转换成 16 位的数据（BIN）。但是在 S.开始的字符串中，符号和最初的“0”以外的数字之间的“空格（20H）”或是“0（30H）”被忽略，而转换成 32 位数据（BIN）。



➢ 相关指令

名称	内容
FNC116 - ESTR	将 2 进制浮点数数据转换成指定位数的字符串（ASCII 码）的指令
FNC117 - EVAL	将字符串（ASCII 码）数据转换 2 进制浮点数数据的指令
FNC200 - STR	将 BIN 数据转换成字符串（ASCII 码）数据的指令

➢ 出错

一下一些情况会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

1）所有位数 S1.为如下所示的范围以外时。错误代码 K6706。16 位预算时：2≤S1.≤8
32 位预算时：2≤S1.≤13；

2）小数部分位数 S1+1.为如下所示的范围以外时。错误代码 K6706。16 位预算时：2≤S1.≤5
32 位预算时：2≤S1.≤10；

4）要转换的字符串（S.以后）的所有字符数，和小数部分的字符数之间的关系非如下所示的范围时，错误代码 K6706。所有字符数-3 ≥ 小数部分字符数

5）在符号中设定了“空格（20H）”,“-（2DH）”以外的 ASCII 码时，错误代码 K6706。

6）各数字的位数中设定了“0（30H）”～ “9（39H）”,以及小数点“.(2EH)”以外的 ASCII 码时，错误代码 K6706。

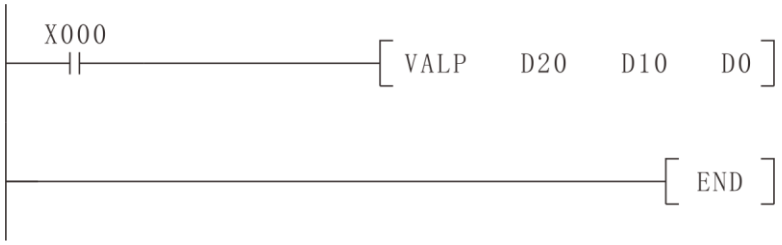
7）要转换的字符串（S.以后）中设定了多个小数点“.(2EH)”时，错误代码 K6706。

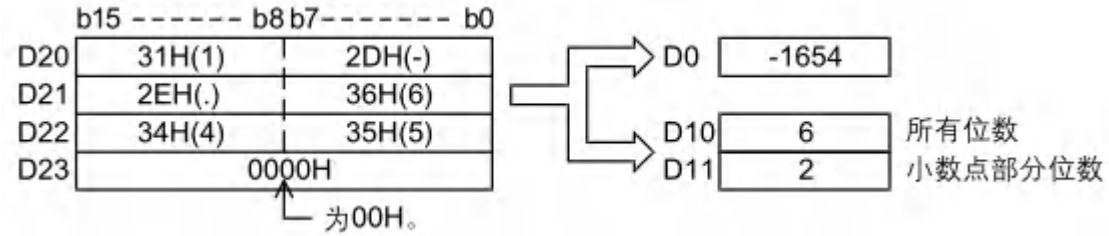
8）转换后的 BIN 数据超出了如下所示的范围时，错误代码 K6706。16 位运算时：-32768~32767；
32 位运算时：-2147483648~2147483647。

9）从 S.开始到相应软元件的最后软元件编号之间不存在“00H”时，错误代码 K6706。

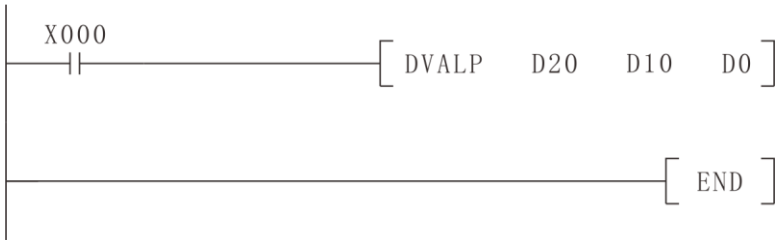
➢ 程序举例

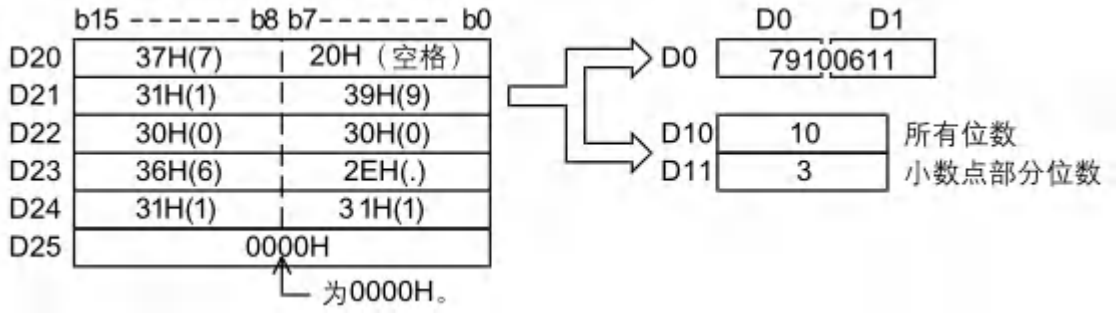
1）当 X000 为 ON 时，根据 D20~D22 中保存的字符串数据视为整数值转换成 BIN 值，然后保存到 D0 中的程序。





1）当 X000 为 ON 时，根据 D20~D24 中保存的字符串数据视为整数值转换成 BIN 值，然后保存到 D0 中的程序。





FNC202 - $+：字符串的结合

➢ 功能说明

连接字符串与字符串的指令。

➢ 指令格式

$+【P】 S1. S2. D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存连接源数据（字符串）的软元件起始编号，或是被直接指定的字符串	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	字符串“□”	字符串
S2.	保存要连接的数据（字符串）的软元件起始编号，或是被直接指定的字符串	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	字符串“□”	
D.	保存连接后的数据（字符串）的软元件起	--	KnY、KnM、KnS、T、C、	--	

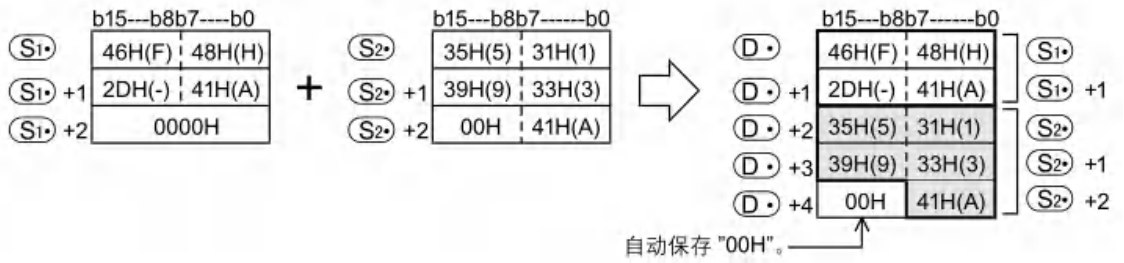
	始编号		D、R、修饰		

➢ 动作说明

1、16 位运算（$+/$P+）

在 S1.开始的字符串数据后面连接 S2.开始的字符串数据，然后保存到 D.以后的软元件中。

S1.和 S2.的字符串，是指以字节为单位从被指定的软元件开始到检测第一个【00H】的位置为止的数据。



1）字符串的结合，就是指忽略 S1 中表示指定字符串末尾的“00H”,接着 S1 的最后字符连接 S2.中指定的字符串。此外，执行字符串的结合后会自动将“00H”附加在最后。

连接后的字符数为奇数时，在保护最后字符的软元件的高字节中保存“00H”。

连接后的字符数为偶数时，在保护最后字符的软元件的下一个软元件中保存“0000H”。

➢ 注意要点

1）直接指定字符串时，可以指定（输入）的字符数最多为 32 个字符。但是 S1 和 S2 中被指定了字软元件时，字符数没有限制。

2）S1.，S2.中任何一个的值都是从“00H”开始时（字符数为 0 时），在 D.中保存“0000H”。

➢ 出错

一下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

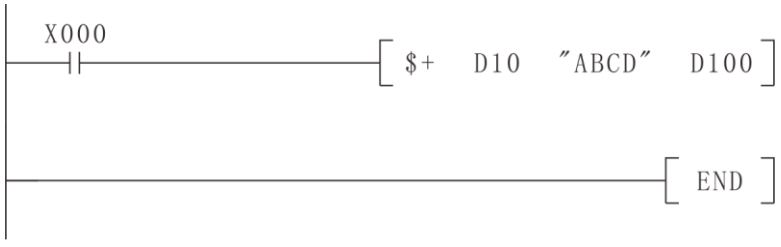
1）D.中指定的软元件编号开始的软元件数，比保存所有已经结合的字符串所需要的软元件数量更少时。（在所有的字符串和最终字符后面不能保存【00H】）,错误代码 K6706。

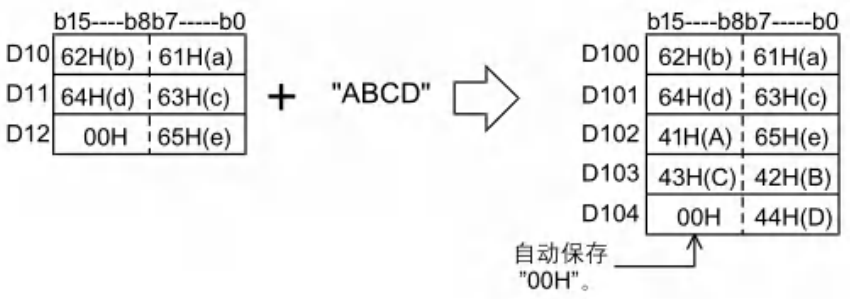
2）当 S1.和 S2.中指定的保存字符串的软元件和 D.中指定的保存字符串的软元件编号重复时。错误代码 K6706。

3）当 S1.和 S2.中指定的软元件开始的相应软元件范围中没有设定【00H】时，错误代码 K6706.

➢ 举例

当 X000 为 ON 时，将 D10~D12 中保存的字符串（abcde）和字符串“ABCD”结合，然后保存到 D100 中的程序。





FNC203 - LEN：检测出字符串的长度

➢ 功能说明

检测出指定字符串的字符数（字符数）的指令。

➢ 指令格式

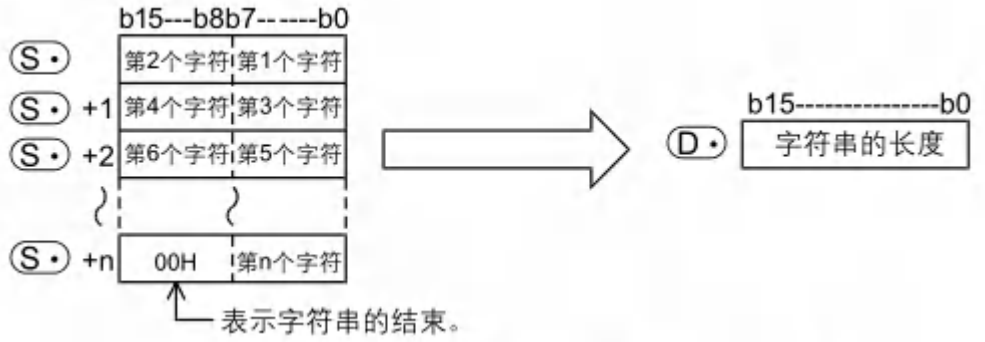
LEN【P】　S.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S.	保存要检测出字符数的字符串的软元件起始编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	字符串“□”	字符串
D.	保存已检测出的字符串长度（字节数）的软元件编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	BIN16 位

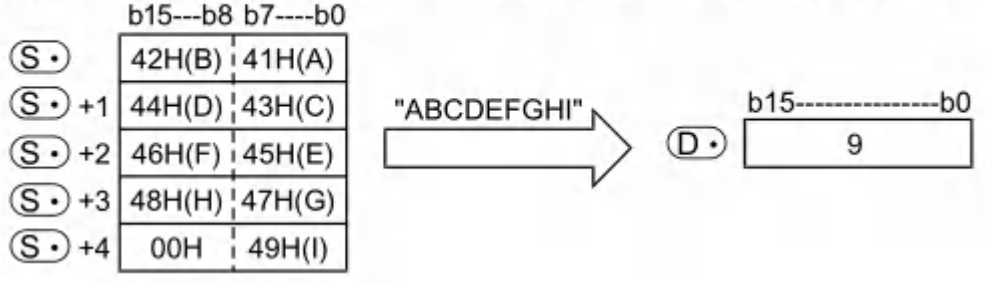
➢ 动作说明

1、16 位运算（LEN/LENP）

检测出以 S.开头的字符串的长度，将字符串的长度保存到 D.中，以字节为单位，将从 S.开始到第 1 个保存有【00H】的软元件编号为止的数据，作为字符串进行处理。



例如，如下所示的在S·中保存“ABCDEFGHI”时，D·中保存K9。



➢ **注意要点**

1）在这个指令中，也可以指定使用 ASCII 码以外的字符代码，但是字符串的长度，只能以字节为单位（8 位）。

➢ **出错**

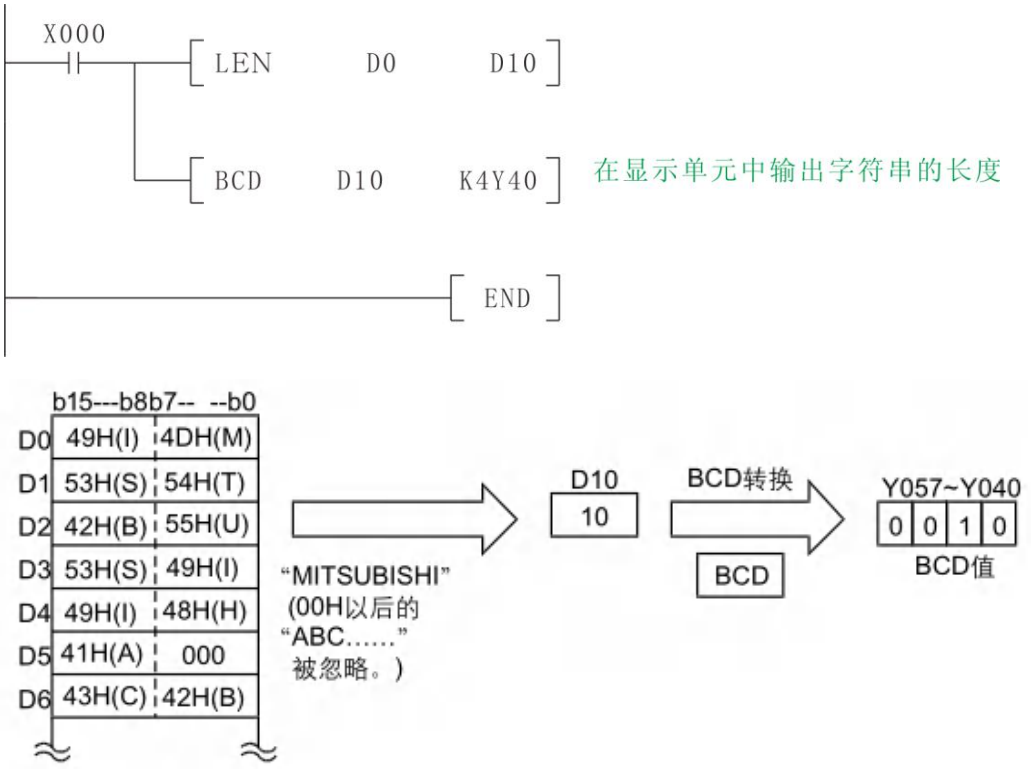
一下一些情况下会发生运算出错，出错标志位 M8067 置 ON，D8067 中保存错误代码。

1）当 S.指定的软元件编号开始的相应软元件范围内没有设定【00H】时，错误代码 K6706.

2）检测出字符数超过 32768 个时。错误代码 K6706。

➢ **举例**

当 X000 为 ON 时，将 D0 开始的字符串的长度，以 BCD 4 位数形式输出到 Y040 ~Y057 中的程序。



FNC204 - RIGHT：从符串的右侧开始取出

FNC205 - LEFT：从符串的左侧开始取出

FNC206 - MIDR：从字符串中任意取出

FNC207 - MIDW：字符串中的任意替换

FNC208 - INSTR：字符串的检索

FNC209 - $MOV：字符串的传送

7-19 数据块处理 3 - FNC210~FNC219

FNC210 - FDEL：数据表的数据删除

FNC211 - FINS：数据表的数据插入

FNC212 - POP：读取后入的数据【先入后出控制器用】

FNC213 - SFR：16 位数据 n 位右移（带进位）

FNC214 - SFL：16 位数据 n 位左移（带进位）

7-20 触点比较指令 - FNC220~FNC249

FNC224~FNC230 - LD=,>,<,<>,<=,>= ： 母线连接的触点比较

➢功能说明

对 S1.与 S2.内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，连接母线的触点比较指令。

➢指令格式：（S-源操作数）

LD【D】xx S1. S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
224	LD＝	LDD＝	S1.＝S2.	S1.≠S2.
225	LD＞	LDD＞	S1.＞S2.	S1.≤S2.
226	LD＜	LDD＜	S1.＜S2.	S1.≥S2.
228	LD＜＞	LDD＜＞	S1.≠S2.	S1.＝S2.
229	LD＜＝即≤	LDD＜＝即≤	S1.≤S2.	S1.＞＝S2.
230	LD＞＝即≥	LDD＞＝即≥	S1.≥S2.	S1.＜S2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	

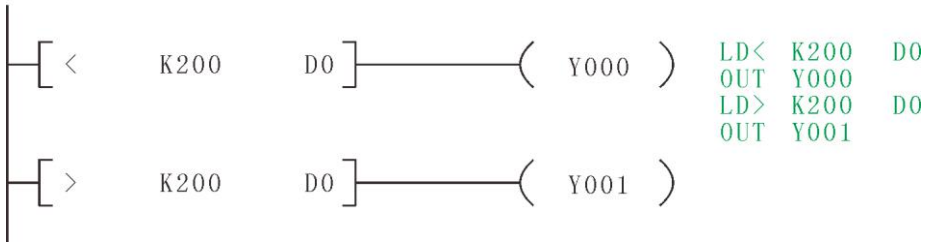
➢举例

1、16 位

操作前：（D0）=100,（Y0）=0，（Y1）=0；

操作后：（D0）=100,（Y0）=0，（Y1）=1；

16位

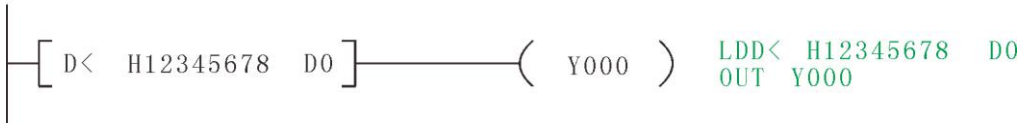


2、32 位

操作前：（D0）=0X12345678,（Y0）=0；

操作后：（D0）=100,（Y0）=0；

32位



FNC232~FNC238 - AND=,>,<,<>,<=,>= ： 串联连接的触点比较指令

➢功能说明

对 S1.与 S2.内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，与其它接点串联连接的触点比较指令。

➢指令格式

AND【D】xx S1. S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
232	AND=	ANDD=	S1.=S2.	S1.≠S2.
233	AND>	ANDD>	S1.>S2.	S1.≤S2.
234	AND<	ANDD<	S1.<S2.	S1.≥S2.
236	AND<>	ANDD<>	S1.≠S2.	S1.=S2.
237	AND<=即≤	ANDD<=即≤	S1.≤S2.	S1.>S2.
238	AND>=即≥	ANDD>=即≥	S1.≥S2.	S1.<S2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	

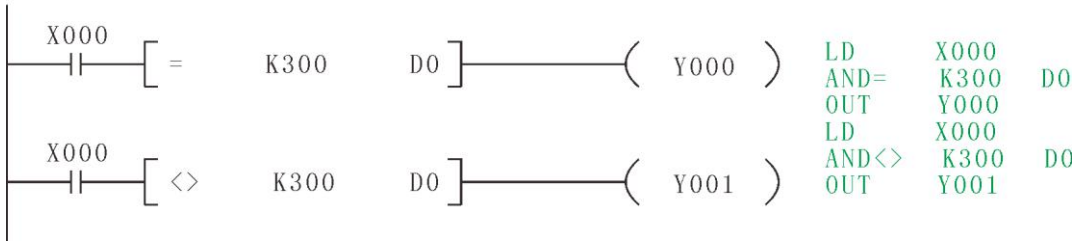
➢举例

1、16 位

操作前：（X0）=1，（D0）=300,（Y0）=0，（Y1）=0；

操作后：（X0）=1，（D0）=300,（Y0）=1，（Y1）=0；

16位

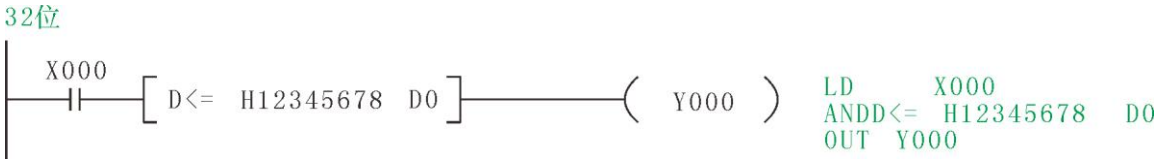


2、32 位

操作前：（X0）=1，（D0）=0X12345678,（Y0）=0；

操作后：（X0）=1，（D0）=0x12345678,（Y0）=1；

32位



FNC240~FNC246 - OR=,>,<,<>,<=,>= ： 并联连接的触点比较指令

➢功能说明

对 S1.与 S2.内容进行 BIN 比较，对应其结果执行后段运算。该类指令是，与其它接点并联连接的触点比较指令。

➢指令格式

OR【D】xx　S1.　S2.

功能号	16 位指令	32 位指令	导通条件	非导通条件
240	OR＝	ORD＝	S1.＝S2.	S1.≠S2.
241	OR＞	ORD＞	S1.＞S2.	S1.≤S2.
242	OR＜	ORD＜	S1.＜S2.	S1.≥S2.
244	OR＜＞	ORD＜＞	S1.≠S2.	S1.＝S2.
245	OR＜＝即≤	ORD＜＝即≤	S1.≤S2.	S1.＞S2.
246	OR＞＝即≥	ORD＞＝即≥	S1.≥S2.	S1.＜S2.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	BIN16/32 位
S2.	保存比较数据的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、V、Z	K、H	

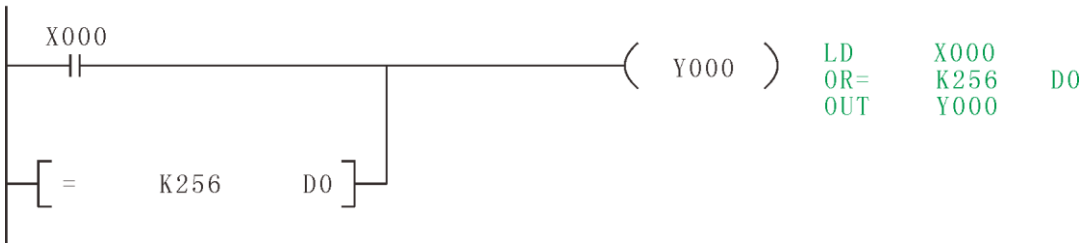
➢ **举例**

1、16 位

操作前：（X0）=0，（D0）=256,（Y1）=0；

操作后：（X0）=0，（D0）=256,（Y1）=1；

16位

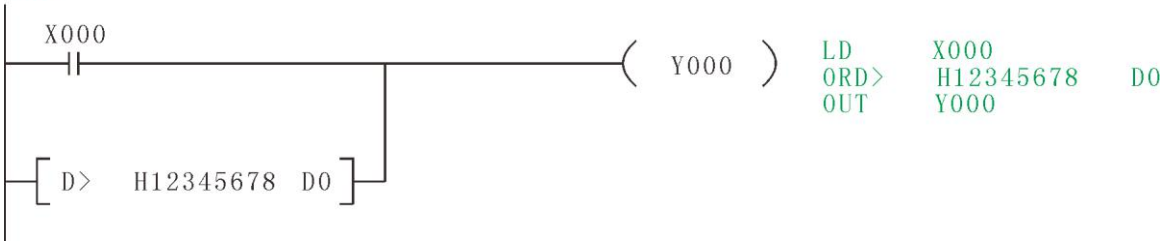


2、32 位

操作前：（X0）=0，（D0）=0X12345678,（Y0）=0；

操作后：（X0）=0，（D0）=0x12345678,（Y0）=0；

32位



7-21 数据表处理指令 - FNC250~FNC269

FNC256 - LIMIT：上下限限位控制

➢功能说明

取出指定的字符串中任意位置上的字符串的指令。

➢指令格式

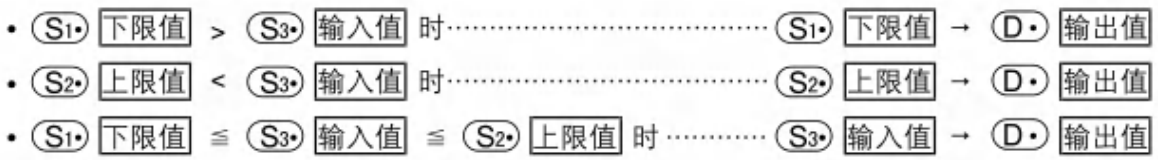
【D】LIMIT【P】 S1. S2. S3 D.

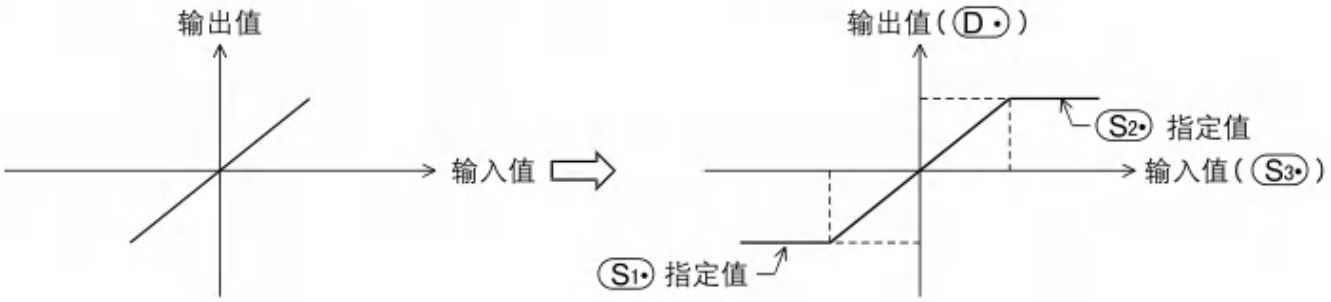
操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	下限限位值（最小输出界限值）	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN32/16 位
S2.	上限限位值（最小输出界限值）	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
S3.	需要通过上下限限位控制的输入值	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	
D.	保存已经上下限位控制的输出值的软元件起始编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	

➢动作说明

1、16 位运算（LIMT/LIMITP）

通过判断 S3.中指定的输入值（BIN16 位值），是否在 S1.、S2.指定的上下限制的范围内，以此控制保存在 D.



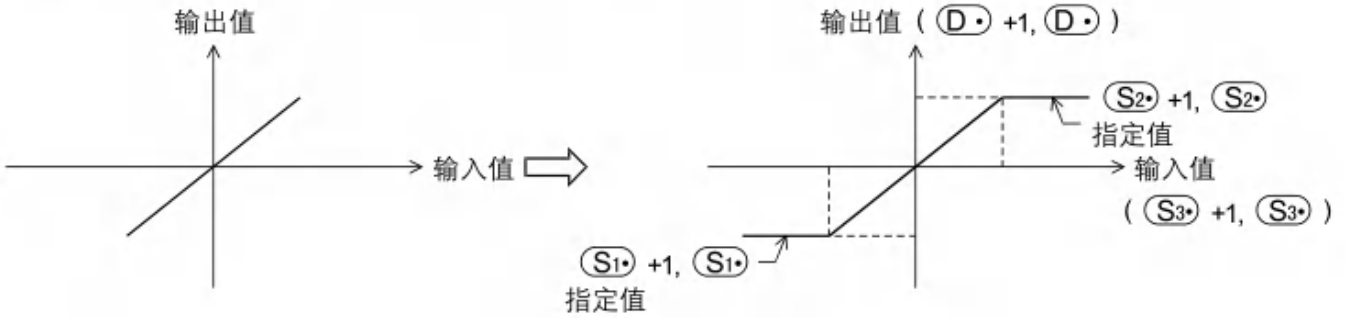


1）仅通过上限限位值进行控制时，在 S1.指定的下限限位值中设定“-32768”。

2）仅通过下限限位值进行控制时，在 S2.指定的下限限位值中设定“32768”。

2、32 位运算（DLIMT/DLIMITP）

通过判断【S3.+1，S3.】中指定的输入值（BIN32 位值），是否在【S1.+1，S1.】、【S2.+1，S2.】指定的上下限制的范围内，以此控制保存在【D.+1，D.】指定的软元件中的输出值。



1）仅通过上限限位值进行控制时，在【S1.+1，S1.】指定的下限限位值中设定“-2147483648”。

2）仅通过下限限位值进行控制时，在【S2.+1，S2.】指定的下限限位值中设定“2147483647”。

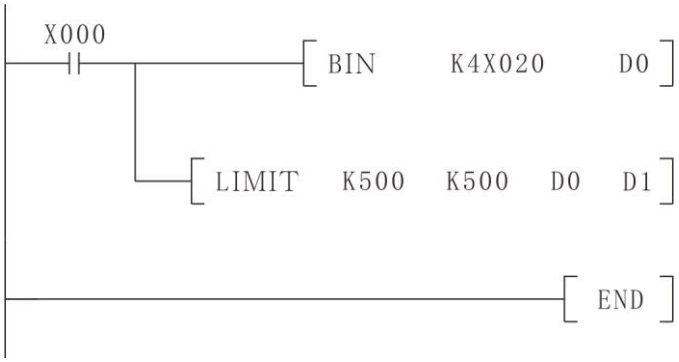
➢出错

在下述设定状态下执行指令后，会出现运算出错，出错标志位（M8067）为 ON，错误代码 K6706 保存在 D8067 中。16 位运算时，S1. ≤ S2.；32 位运算时，【S1.+1，S1.】 ≤ 【S2+1，S2.】。

➢举例

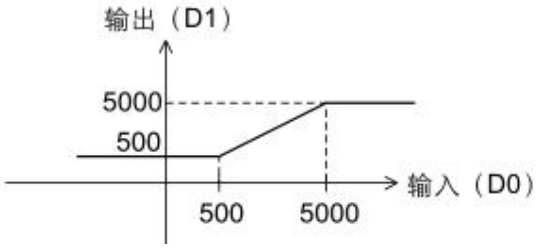
1、程序示例 1

当 X000 为 ON 时，对 X020~X037 中以 BCD 值设定的数据执行 500~5000 的限位值控制，并保存到 D1 中的程序。



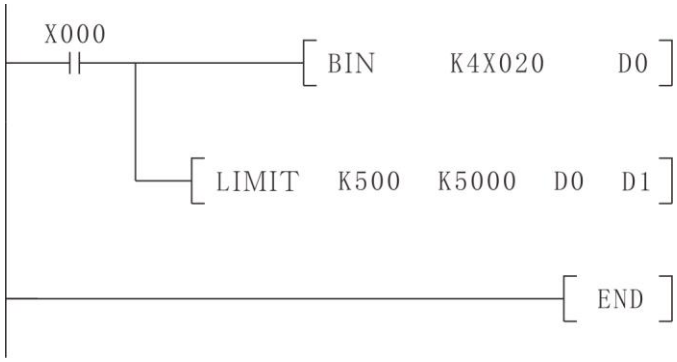
动作

- D0 < 500时，D1为500。
- 500 ≦ D0 ≦ 5000时，D1为D0的值。
- 5000 < D0时，D1为5000。



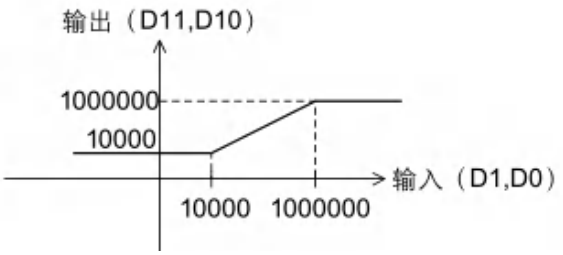
2、程序示例 2

当 X000 为 ON 时，对 X020~X057 中以 BCD 值设定的数据执行 10000~1000000 的限位值控制，并保存到 D11，D10 中的程序。



动作

- (D1,D0) < 10000时，(D11,D10)为10000。
- 10000 ≦ (D1,D0) ≦ 1000000时，(D11,D10) 为(D1,D0)的值。
- 1000000 < (D1,D0)时，(D11,D10)为1000000。



FNC257 - BAND：死区控制

➢功能说明

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令。

➢指令格式

【D】BAND【P】　S1.　S2.　S3　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	死区（无输出区域）	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	死区（无输出区域）的上限值	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	
S3.	要通过死区控制的输入值	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	
D.	保存经过死区控制的输出值的软元件编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	

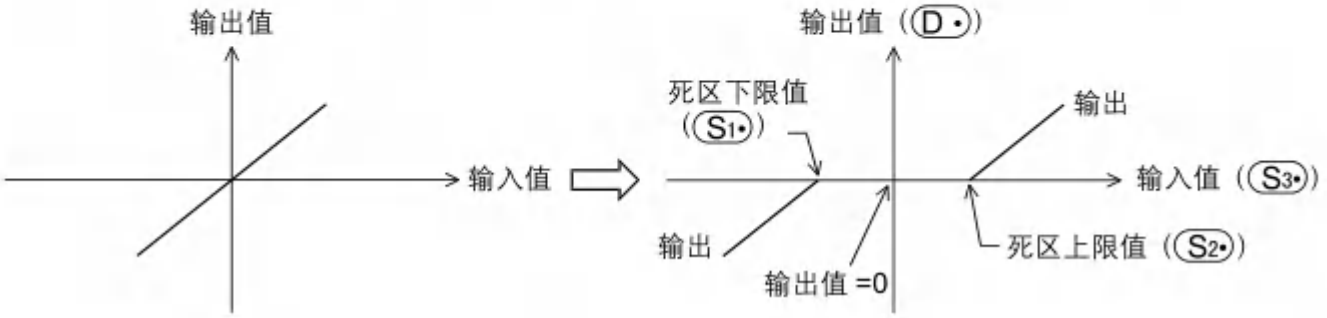
动作说明

1、16 位运算（BAND/BANDP）

通过判断 S3.指定的输入值（BIN16 位值）是否在【S1.、S2.】中指定的死区范围内，以此来控制保存在 D.指定的软元件中的输出值。

输出值如下所示被控制

- (S1·) 下限值 > (S3·) 输入值 时 …… (S3·) 输入值 − (S1·) 下限值 → (D·) 输出值
- (S2·) 上限值 < (S3·) 输入值 时 …… (S3·) 输入值 − (S2·) 上限值 → (D·) 输出值
- (S1·) 下限值 ≦ (S3·) 输入值 ≦ (S2·) 上限值 时 …… 0 → (D·) 输出值

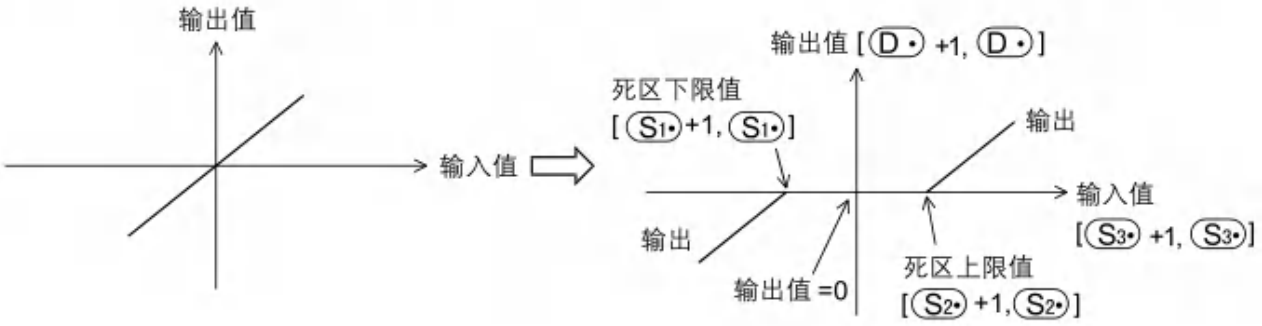


2、32 位运算（DBAND/DBANDP）

通过判断【S3.，S3+1】指定的输入值（BIN32 位值）是否在【S1.，S1.+1】、【S2.，S2+1】中指定的死区范围内，以此来控制保存在 D.指定的软元件中的输出值。

输出值如下所示被控

- (S1·)+1, (S1·) 下限值 > (S3)+1, (S3) 输入值 时 …… (S3)+1, (S3) 输入值 − (S1·)+1, (S1·) 下限值 → (D·)+1, (D·) 输出值
- (S2·)+1, (S2·) 上限值 < (S3)+1, (S3) 输入值 时 …… (S3)+1, (S3) 输入值 − (S2·)+1, (S2·) 上限值 → (D·)+1, (D·) 输出值
- (S1·)+1, (S1·) 下限值 ≦ (S3)+1, (S3) 输入值 ≦ (S2·)+1, (S2·) 上限值 时 …… 0 → (D·)+1, (D·) 输出值



➢注意要点

1、输出值增益时，如下所示。

1）16 位运算时，输出值为带符号的 16 位 BIN 值，因此，运算结果超出-32768~32767。

2）32 位运算时，输出值为带符号的 16 位 BIN 值，因此，运算结果超出-2147483648~2147483647。

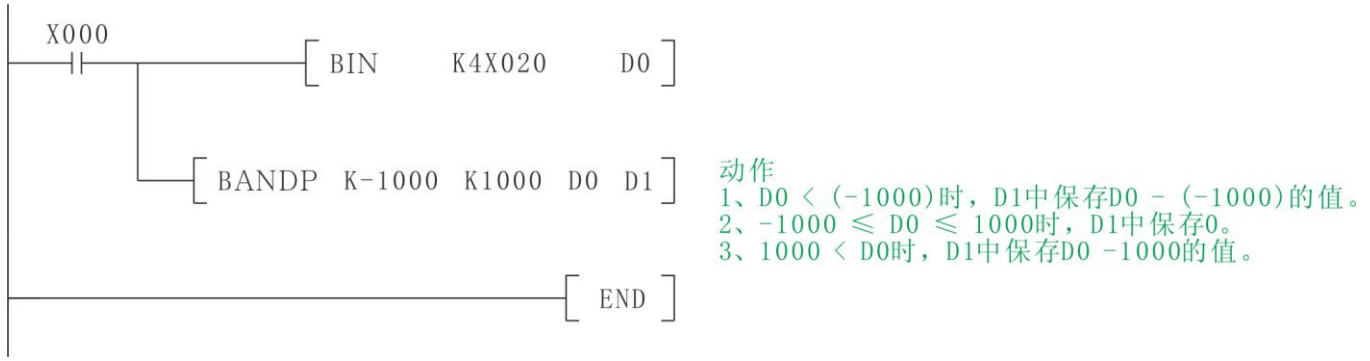
➢出错

在下述设定状态下执行指令时，会出现运算出错，出错标志位（M8067）为 ON，错误代码 K6706，保存在 D8067 中。16 位运算时：S1.＞ S2.；32 位运算时：【S1.+1，S1.】＞【S2.+1，S2.】。

➢举例

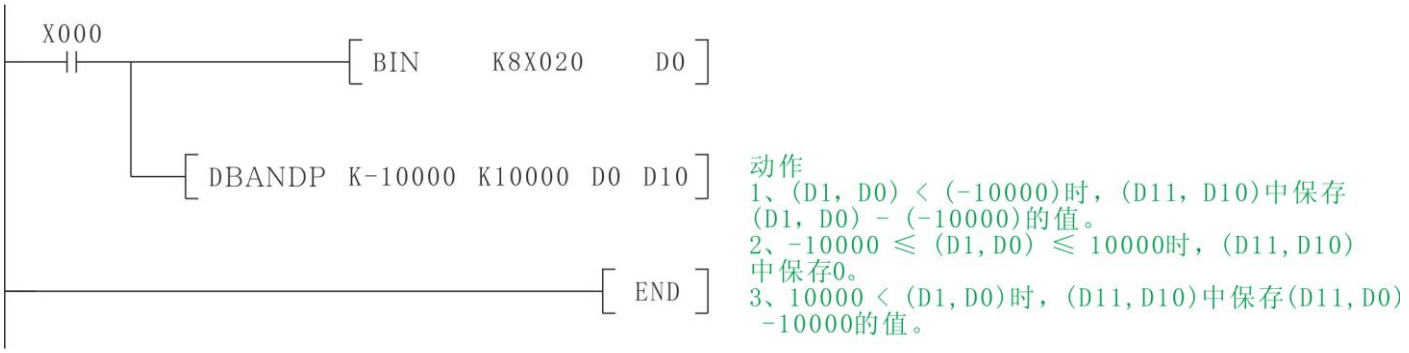
1、程序示例 1

当 X000 为 ON 时，对 X020~X037 中以 BCD 值设定的数据执行-1000 ~ 1000 的死区控制，并保存到 D1 中的程序。



2、程序示例 2

当 X000 为 ON 时，对 X020~X057 中以 BCD 值设定的数据执行-1000 ~ 1000 的死区控制，并保存到 D10，D11 中的程序。



FNC258 - ZONE：区域控制

➢功能说明

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令。

➢指令格式

【D】ZONE【P】 S1. S2. S3 D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	加在输入值上的负偏差值	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN32/16 位
S2.	加在输入值上的正偏差值	--	KnX、KnY、KnM、KnS、	K、H	

			T、C、D、R、修饰		
S3.	要通过区域控制的输入值	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	--	
D.	保存已通过区域控制的输出值的软元件起始编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	

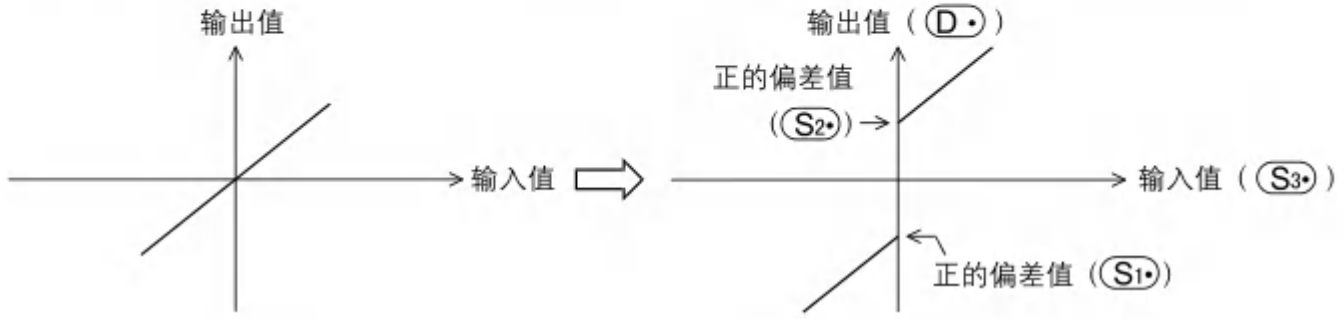
动作说明

1、16 位运算（ZONE/ZONEP）

在 S3.指定的输入值上加上 S1.或 S2.指定的偏差值，然后保存到 D.指定的软元件编号中。

偏差值，如下所示执行。

- (S3·) 输入值 < 0时 ····· (S3·) 输入值　(S1·) 负的偏差值 → (D·) 输出值
- (S3·) 输入值 = 0时 ····· 0 → (D·) 输出值
- (S3·) 输入值 > 0时 ····· (S3·) 输入值　(S2·) 正的偏差值 → (D·) 输出值

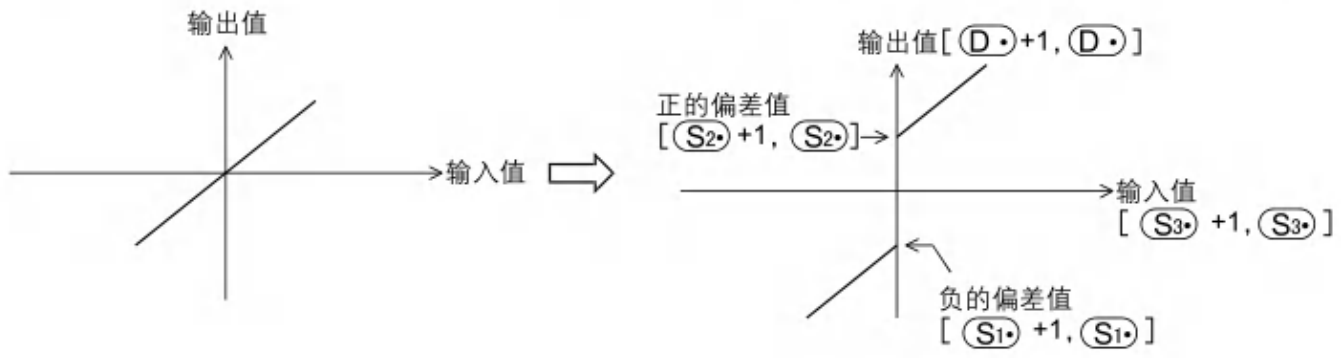


2、32 位运算（ZONE/ZONEP）

在【S3.+1，S3.】指定的输入值上加上【S1.+1，S1.】或【S2.+1，S2.】指定的偏差值，然后保存到【D.，D.+1】指定的软元件编号中。

偏差值，如下所示执行。

- (S3·)+1, (S3·) 输入值 < 0时 ····· (S3·)+1, (S3·) 输入值　(S1·)+1, (S1·) 负的偏差值 → (D·)+1, (D·) 输出值
- (S3·)+1, (S3·) 输入值 = 0时 ····· 0 → (D·)+1, (D·) 输出值
- (S3·)+1, (S3·) 输入值 > 0时 ····· (S3·)+1, (S3·) 输入值　(S2·)+1, (S2·) 正的偏差值 → (D·)+1, (D·) 输出值



➢**注意要点**

1、输出值增益时，如下所示。

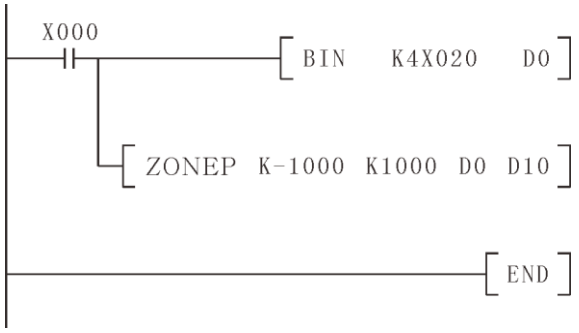
1）16 位运算时，输出值为带符号的 16 位 BIN 值，因此，运算结果超出-32768~32767。

2）32 位运算时，输出值为带符号的 16 位 BIN 值，因此，运算结果超出-2147483648~2147483647。

➢**举例**

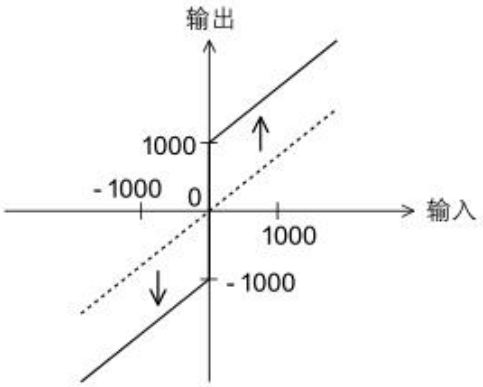
1、程序示例 1

当 X1000 为 ON 时，对 X020~X037 中以 BCB 值设定的数据执行-1000~1000 的区域控制，并保存到 D1 中的程序。



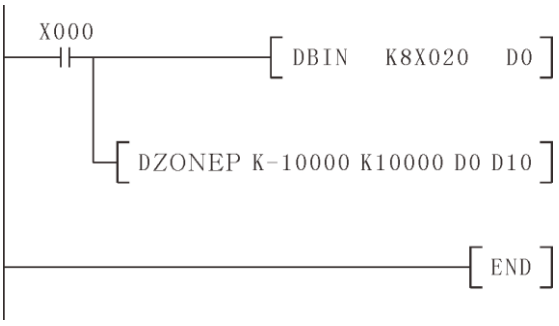
动作

- D0 < 0时，D1中保存(D0)＋(-1000)的值。
- D0 ＝0时，D1中保存0。
- 0< D0时，D1中保存(D0)＋1000的值。



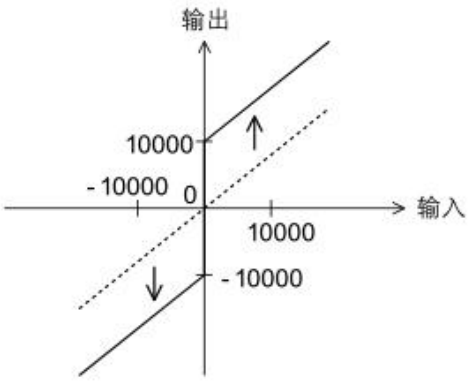
2、程序示例 2

当 X000 为 ON 时，对 X020~X057 中以 BCB 值设定的数据执行-10000~10000 的区域控制，并保存到 D10，D11 中的程序。



动作

- (D1,D0) < 0时，(D11,D10)中保存(D1,D0)＋(-10000)的值。
- (D1,D0) ＝ 0时，(D11,D10)中保存0。
- 0 < (D1,D0)时，(D11,D10)中保存(D1,D0)＋10000的值。



FNC259 - SCL：定坐标（不同点坐标数据）

➢功能说明

根据指定的数据表格，对输入值执行定坐标后输出的指令。此外，还有数据表格结构不同的 SCL2 指令。

➢指令格式

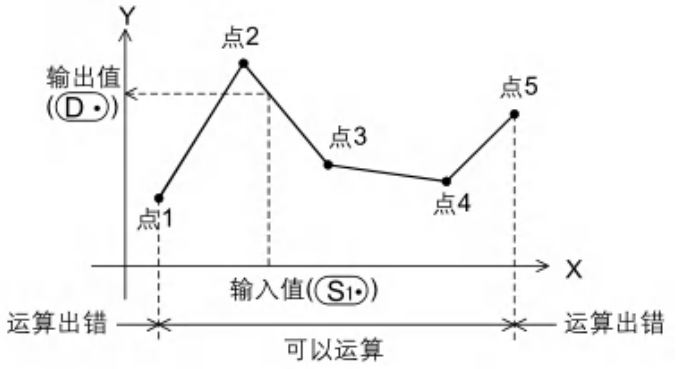
【D】SCL【P】　S1.　S2.　D.

操作数种类	内容	适用软元件			数据类型
		位软元件	字软元件	其他	
S1.	执行定坐标的输入值或保存输入值的软元件编号	--	KnX、KnY、KnM、KnS、T、C、D、R、修饰	K、H	BIN16/32 位
S2.	定坐标用的转换表格软元件的起始编号	--	D、R、修饰	--	
D.	保存被定坐标控制的输出值的软元件编号	--	KnY、KnM、KnS、T、C、D、R、修饰	--	

➢动作说明

1、16 位运算（SCL/SCLP）

根据指定的转换特性，对 S1.指定的输入值执行定坐标，然后保存到 D.指定的软元件编号中。定坐标用的转换，是依据保存在 S2.指定的软元件开始的数据表格执行的。但是输出数据不是整数数值时，小数第 1 位四舍五入后输出。

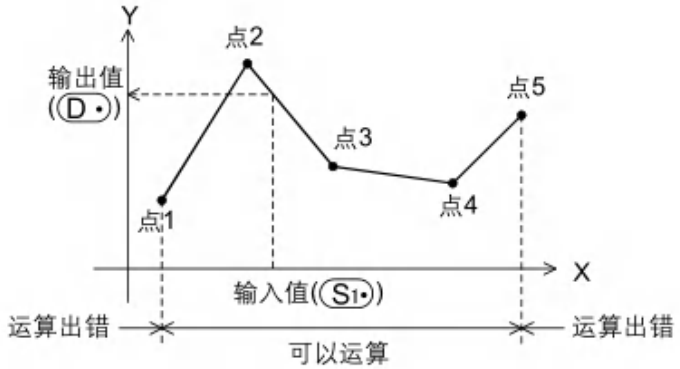


设定项目		设定数据表格的软元件分配
坐标点数（当位上图时变为“5”）		S2.
X 坐标	点 1	S2.+1

	点 2	S2.+2
	点 3	S2.+3
	点 4	S2.+4
	点 5	S2.+5
Y 坐标	点 1	S2.+6
	点 2	S2.+7
	点 3	S2.+8
	点 4	S2.+9
	点 5	S2.+10

2、32 位运算（DSCL/DSCLP）

根据指定的转换特性，对【S1.+1，S1.】指定的输入值执行定坐标，然后保存到【D.+1，D.】指定的软元件编号中。定坐标用的转换，是依据保存在【S2.+1，S2.】指定的软元件开始的数据表格执行的。但是输出数据不是整数数值时，小数第 1 位四舍五入后输出。



设定项目		设定数据表格的软元件分配
坐标点数（当位上图时变为“5”）		【S2.+1，S2.】
点 1	X 坐标	【S2.+3，S2.+2】
	Y 坐标	【S2.+5，S2.+4】
点 2	X 坐标	【S2.+7，S2.+6】
	Y 坐标	【S2.+9，S2.+8】
点 3	X 坐标	【S2.+11，S2.+10】
	Y 坐标	【S2.+13，S2.+12】
点 4	X 坐标	【S2.+15，S2.+14】
	Y 坐标	【S2.+17，S2.+16】
点 5	X 坐标	【S2.+19，S2.+18】
	Y 坐标	【S2.+21，S2.+20】

3、定坐标用转换表格的设定

定坐标用转换表标，是依据保存在【S2.+1,S2.】指定的软元件开始的数据表格执行的。数据表格的结构如下所示。

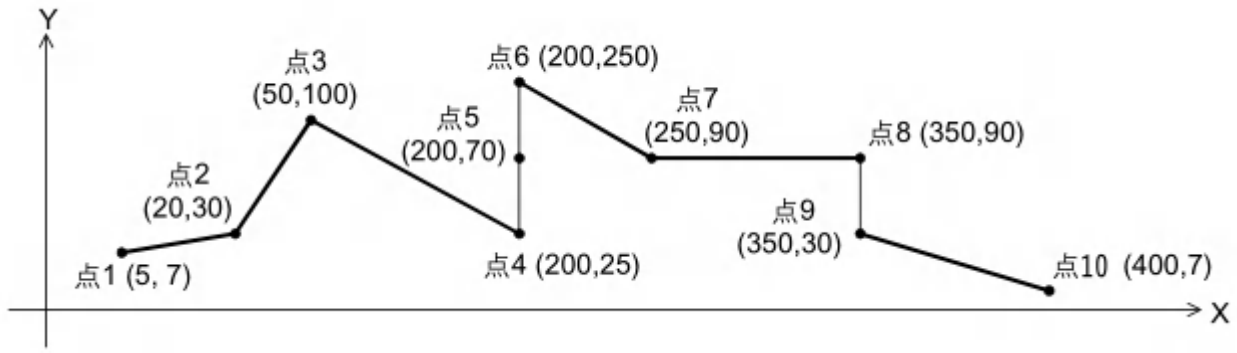
设定项目		设定数据表格的软元件分配	
		16 位运算	32 位运算
坐标点数		S2.	【S2.+1，S2.】
点 1	X 坐标	S2.+1	【S2.+3，S2.+2】

	Y 坐标	S2.+2	【S2.+5，S2.+4】
点 2	X 坐标	S2.+3	【S2.+7，S2.+6】
	Y 坐标	S2.+4	【S2.+9，S2.+8】
...			
点 n(最后)	X 坐标	S2.+2n-1	【S2.+4n-1，S2.+4n-2】
	Y 坐标	S2.+2n	【S2.+4n+1，S2.+4n】

定坐标用转换表格的设定例子

定坐标用转换表格的设定例子中列举了 16 位运算时的例子。执行 32 位运算时，请用 BIN32 位数据设定各设定项目中的数据。

当为下图所示的定坐标用转换特性时，请设定成如下所示的数据表格。



定坐标用转换设定数据表格的设定

设定项目		设定软元件以及设定内容			备注
		S2.中指定了 R0 时		设定内容	
坐标点数		S2.	R0	K10	
点 1	X 坐标	S2.+1	R1	K5	
	Y 坐标	S2.+2	R2	K7	
点 2	X 坐标	S2.+3	R3	K20	
	Y 坐标	S2.+4	R4	K30	
点 3	X 坐标	S2.+5	R5	K50	
	Y 坐标	S2.+6	R6	K100	
点 4	X 坐标	S2.+7	R7	K200	如果想这样指定 3 点的坐标，则输出值为中间值。这个例子中，将点 5 的 Y 坐标指定位输出值（中间值）。此外，即使 3 点以上的 X 坐标相同时，都输出第 2 点的数值。
	Y 坐标	S2.+8	R8	K25	
点 5	X 坐标	S2.+9	R9	K200	
	Y 坐标	S2.+10	R10	K70	
点 6	X 坐标	S2.+11	R11	K200	
	Y 坐标	S2.+12	R12	K250	
点 7	X 坐标	S2.+13	R13	K250	
	Y 坐标	S2.+14	R14	K90	
点 8	X 坐标	S2.+15	R15	K350	如果这样指定 2 点的坐标，则输出值取后一个点的 Y 坐标值。这个例子中，将点 9 的 Y 坐标指定为输出值。
	Y 坐标	S2.+16	R16	K90	
点 9	X 坐标	S2.+17	R17	K350	

	Y 坐标	S2.+18	R18	K30	
点 10	X 坐标	S2.+19	R19	K400	
	Y 坐标	S2.+20	R20	K7	

➢出错

下面的情况下会发生运算出错，出错标志位（M8067）为 ON，错误代码保存在 D8067 中。

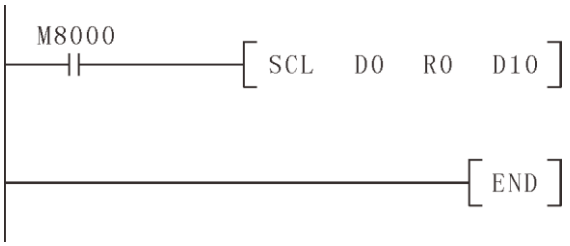
1）数据表格的 Xn 数据没有按照升序排序时，错误代码 K6706。但是由于运算是从数据表格的软元件编号的低位侧开始检索的，所以即使数据表格的一部分没有按照升序排列，但到这个部分为止的运算不会出现运算错误，指令会被执行。

2）S1.在数据表设定的范围以外时。错误代码 K6707。

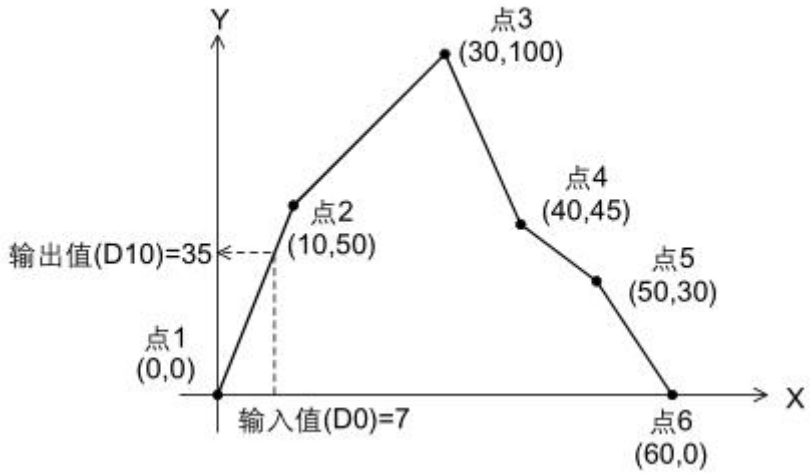
3）运算过程中的数值超出了 32 位数据的范围时，错误代码 K6707。此时请确认各点之间的距离没有超出 65535 以上。如果超出 65535 时，请缩短各点之间的距离。

➢举例

根据 R0 开始的软元件设定的定坐标用转换表格，对 D0 输入的值执行定坐标，然后输出到 D10 中的程序。



动作



定坐标用转换设定数据表格

设定项目		软元件	设定内容
坐标点数		R0	K6
点 1	X 坐标	R1	K0
	Y 坐标	R2	K0
点 2	X 坐标	R3	K10
	Y 坐标	R4	K50
点 3	X 坐标	R5	K30
	Y 坐标	R6	K100
点 4	X 坐标	R7	K40
	Y 坐标	R8	K45

点 5	X 坐标	R9	K50
	Y 坐标	R10	K30
点 6	X 坐标	R11	K60
	Y 坐标	R12	K0

FNC260 - DABBIN：10 进制 ASCII 转 BIN

FNC261 - BINDA：BIN 转 10 进制 ASCII 的转换

7-22 外部设备通信（变频器）- FNC270~FNC274

FNC270 - IVCK：变频器的运行监控

FNC271 - IVDR：变频器的运行控制

FNC272 - IVRD：变频器的参数读取

FNC273 - IVWR：变频器的参数写入

FNC274 - IVBWR：变频器的参数成批写入

7-23 数据传送 3 - FNC278~FNC279

FNC278 - RBFM：BFM 分割读取

FNC279 - WBFM：BFM 分割写入

7-24 高速处理 2 - FNC280~FNC289

FNC280 - HSCT：高速计数器表比较

7-25 扩展文件寄存器控制 - FNC290~FNC299

FNC290 - LOADR：读出扩展文件寄存器

FNC291 - SAVER：成批写入扩展文件寄存器

FNC292 - INITR：扩展寄存器的初始化

FNC293 - LOGR：登录到扩展寄存器

FNC294 - RWER：扩展文件寄存器的删除・写入

FNC295 - INITER：扩展文件寄存器的初始化