HCFA
禾川股份

指令控制手册

浙江禾川科技股份有限公司
Zhejiang Hechuan Technology Co.,Ltd.

A1/A2/A2C/A8/A8C 系列微型可编程控制器
编程手册[基本·应用指令说明书]

通告

此次承蒙购买A1/A2/A2C/A8/A8C系列可编程控制器产品，诚表谢意。
本手册描述了与A1/A2/A2C/A8/A8C系列可编程控制器编程有关的基本·应用指令以及步进梯形图指令。
在使用之前，请阅读本书以及相关产品的手册，希望在充分理解其规格的前提下正确使用产品。
此外，希望本手册能够送达至最终用户处。

根据本书的内容，并非对工业所有权其他的权利的实施予以保证，或是承诺实施权。此外，关于因使用本书中记载的内容而引起的工业所有权方面的各种问题，本公司不承担任何责任。

使用时的请求

- 该产品是以一般的工业为对象制作的通用产品，因此不是以用于关系到人身安全之类的情况下使用的机器或是系统为目的而设计、制造的产品。
- 考虑将该产品用于原子能、电力、宇航、医疗、乘用移动物体用的机器或是系统等特殊用途的时候，请与本公司的营业窗口查询。
- 虽然该产品是在严格的质量体系下生产的，但是用于那些因该产品故障而可能导致重大故障或是产生损失的设备的时候，请在系统上设置后备和安全功能。
- 该产品和其他产品组合使用的情况下，请用户确保应该符合的规格、法规或是规则。此外，关于用户使用的系统、机械、设备中该产品的适用性和安全性，请用户自行确认。

预先通知

- 使用产品时如有疑问，请向具有电气知识（电气施工人员或是同等以上的知识）的专业电气技术人员咨询。
  关于该产品的操作和使用方法有疑问时，请向技术咨询窗口咨询。
- 本书、技术资料、样本等中记载的事例是作为参考用的，不保证动作。选用的时候，请用户自行对机器、设备的功能和安全性进行确认以后使用。
- 关于本书的内容，有时候为了改进可能会有不事先预告就更改规格的情况，还望见谅。
- 关于本书的内容期望能做到完美，可是万一有疑问或是发现有错误，烦请联系本书封底记载的本公司或办事处。
  此时，请将前页中记载的手册编号一并告知。

关于商标

- Microsoft®、Windows®、Excel®是美国Microsoft Corporation在美国以及其他国家中的注册商标或者商标。
- 其他的公司名称、产品名称是其各公司的商标或注册商标。

目录

关于手册中使用的总称·简称的记载…… 13

1. 前言 15

1.1 可编程控制器的编程语言…… 15
1.1.1 编程语言的种类…… 15
1.1.2 程序的互换性…… 16

2. 概要(顺控程序) 17

2.1 便利功能的介绍…… 17
2.1.1 输入处理用的便利功能…… 17
2.1.2 输出处理用的便利功能…… 18
2.1.3 支持顺控的功能…… 18
2.2 应用指令的介绍…… 20
2.2.1 主要应用指令…… 20
2.3 模拟量·定位特殊控制…… 21
2.4 链接·通信…… 22
2.5 构成可编程控制器的软元件的介绍…… 23
2.5.1 各软元件的关系…… 23
2.5.2 软元件的动作及一览…… 24
2.6 程序内存和软元件…… 26
2.6.1 存储器结构…… 26
2.6.2 存储器的动作和停电保持(电源ON/OFF、RUN/STOP)…… 27
2.6.3 有关停电保持方式的种类…… 29
2.6.4 一般用软元件和停电保持软元件的变更…… 30
2.6.5 停电保持软元件的初始化方法…… 30
2.7 参数的种类和设定…… 31
2.7.1 参数一览…… 31
2.7.2 参数的初始值及可以变更的工具…… 32
2.7.3 存储器容量设定范围…… 33
2.7.4 存储器盒适合的机型…… 33
2.7.5 关键字…… 34
2.7.6 特殊功能模块的初始值设定【HCP works】…… 36
2.7.7 定位设定【HCP works】…… 36
2.7.8 使用HCP works设定参数…… 36

3. 指令一览 47

3.1 基本指令…… 47
3.2 步进梯形图指令…… 48
3.3 应用指令 - 按FNC.No顺序…… 49

4. 软元件的作用和功能 62

4.1 软元件编号一览…… 62
4.2 输入输出继电器[X、Y]…… 65
4.2.1 输入输出继电器的编号…… 65
4.2.2 功能和作用…… 65
4.2.3 输入继电器的动作时序…… 66

4.3 辅助继电器[M]…… 67
4.3.1 辅助继电器的编号…… 67
4.3.2 功能和动作实例…… 67
4.4 状态[S]…… 69
4.4.1 状态的编号…… 69
4.4.2 功能和动作实例…… 69
4.5 定时器[T]…… 71
4.5.1 定时器的编号…… 71
4.5.2 功能和动作实例…… 71
4.5.3 设定值的指定方法…… 73
4.5.4 子程序内的注意事项…… 73
4.5.5 定时器动作的详细内容和定时器的精度…… 73
4.5.6 程序举例【断开延时定时器，闪烁】…… 74
4.5.7 作为数据软元件的处理…… 76
4.6 计数器[C]…… 77
4.6.1 计数器的编号…… 77
4.6.2 计数器的特征…… 77
4.6.3 相关软元件(增/减的指定)【32位计数器】…… 77
4.6.4 功能和动作实例…… 78
4.6.5 设定值的指定方法…… 79
4.6.6 计数器的响应速度…… 79
4.6.7 作为数据软元件的使用…… 80
4.7 高速计数器[C](A8/A8C可编程控制器)…… 81
4.7.1 高速计数器的种类和软元件的编号…… 81
4.7.2 高速计数器的输入分配…… 83
4.7.3 高速计数器的使用…… 84
4.7.4 当前值更新时序及当前值的比较…… 86
4.7.5 相关软元件…… 87
4.7.6 关于外部复位输入信号的逻辑变更…… 88
4.7.7 关于计数器的输入端子分配和功能的切换…… 88
4.7.8 以4倍频使用双相双计数的计数器C251～C255的方法…… 89
4.7.9 将硬件计数器作为软件计数器使用的条件…… 90
4.7.10 高速计数器的响应频率…… 91
4.7.11 使用上的注意事项…… 93
4.8 数据寄存器、文件寄存器[D]…… 95
4.8.1 数据寄存器、文件寄存器的编号…… 95
4.8.2 数据寄存器、文件寄存器的构造…… 95
4.8.3 数据寄存器的功能和动作实例…… 96
4.8.4 文件寄存器的功能和动作实例…… 98
4.8.5 使用文件寄存器的注意事项…… 101
4.9 扩展寄存器[R]、扩展文件寄存器[ER]…… 102
4.9.1 扩展寄存器、扩展文件寄存器的编号…… 102
4.9.2 数据的存储地点和访问方法…… 102
4.9.3 扩展寄存器，扩展文件寄存器的构造…… 103
4.9.4 扩展寄存器、扩展文件寄存器的初始化…… 103
4.9.5 扩展寄存器的功能和动作实例…… 104
4.9.6 扩展文件寄存器的功能和动作实例…… 105
4.9.7 使用扩展文件寄存器的注意事项…… 107
4.9.8 关于扩展寄存器及扩展文件寄存器的数据登录…… 108
4.10 变址寄存器[V、Z]…… 136
4.10.1 变址寄存器的编号…… 136
4.10.2 功能和构造…… 136
4.10.3 软元件的修饰…… 136
4.11 指针[P]、[I]…… 137
4.11.1 指针的编号…… 137
4.11.2 分支用指针的功能和动作实例…… 137
4.11.3 中断用指针的功能和动作实例…… 138

5. 指令的软元件·常数的指定方法 117

5.1 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)…… 117
5.1.1 数值的种类…… 117
5.1.2 数值的转换…… 118
5.1.3 浮点运算中数值的处理…… 119
5.2 常数K、H、E(10进制数/16进制数/实数)的指定…… 121
5.2.1 常数K(10进制数)…… 121
5.2.2 常数H(16进制数)…… 121
5.2.3 常数E(实数)…… 121
5.3 字符串…… 122
5.3.1 字符串常数(“ABC”)…… 122
5.3.2 字符串数据…… 122
5.4 位的位数指定(K n□***)…… 123
5.5 字软元件的位指定(D□.b)…… 124
5.6 缓冲寄存器的直接指定(U□\G□)…… 124
5.7 变址修正…… 125
5.7.1 基本指令的变址修正…… 125
5.7.2 应用指令的变址修饰…… 126
5.7.3 使用次数受限制的指令的变址修饰实例…… 128

6. 编程前须知 129

6.1 指令说明的阅读方法…… 129
6.2 编程方面的基本注意事项…… 131
6.2.1 程序的步骤及执行顺序…… 131
6.2.2 双重输出(双线圈)的动作及对策…… 132
6.2.3 不能编程的回路图及对策…… 133
6.3 输入输出处理，响应延迟…… 134
6.4 程序流程控制指令之间的相互关系…… 135
6.5 应用指令的一般通则…… 137
6.5.1 应用指令的表示和执行形式…… 137
6.5.2 一般标志位的使用…… 139
6.5.3 运算错误标志位的使用…… 141
6.5.4 扩展功能用标志位的使用…… 141
6.5.5 指令使用次数的限制…… 142

7. 基本指令 143

7.1 LD、LDI指令…… 145
7.2 OUT指令…… 147
7.3 AND、ANI指令…… 150
7.4 OR、ORI指令…… 153
7.5 LDP、LDF、ANDP、ANDF、ORP、ORF指令…… 155
7.6 ORB指令…… 162
7.7 ANB指令…… 163
7.8 MPS、MRD、MPP指令…… 164
7.9 MC、MCR指令…… 169
7.10 INV指令…… 172
7.11 MEP、MEF指令…… 173
7.12 PLS、PLF指令…… 175
7.13 SET、RST指令…… 177
7.14 NOP指令…… 180
7.15 END指令…… 181
7.16 指令步数和指定软元件…… 182

8. 程序流程 - FNC 00 ~ FNC 09 183

8.1 FNC 00 - CJ/条件跳转 184
8.1.1 CJ指令和触点线圈的动作 188
8.1.2 主控和跳转指令的关系 190
8.2 FNC 01 - CALL/子程序调用 191
8.2.1 子程序和中断子程序内的注意事项 193
8.3 FNC 02 - SRET/子程序返回 195
8.4 FNC 03 - IRET/中断返回 196
8.5 FNC 04 - EI/允许中断 198
8.6 FNC 05 - DI/禁止中断 199
8.7 FNC 06 - FEND/主程序结束 200
8.8 FNC 07 - WDT/看门狗定时器 202
8.9 FNC 08 - FOR/循环范围的开始 205
8.10 FNC 09 - NEXT/循环范围的结束 206

9. 传送·比较 - FNC 10 ~ FNC 19 209

9.1 FNC 10 - CMP/比较 210
9.2 FNC 11 - ZCP/区间比较 212
9.3 FNC 12 - MOV/传送 214
9.4 FNC 13 - SMOV/位移动 217
9.5 FNC 14 - CML/反转传送 219
9.6 FNC 15 - BMOV/成批传送 221
9.6.1 文件寄存←→数据寄存器之间的传送功能 223
9.7 FNC 16 - FMOV/多点传送 225
9.8 FNC 17 - XCH/交换 227
9.9 FNC 18 - BCD/BCD转换 228
9.10 FNC 19 - BIN/BIN转换 231

10. 四则逻辑运算 - FNC 20 ~ FNC 29 234

10.1 FNC 20 - ADD/BIN加法运算 235
10.2 FNC 21 - SUB/BIN减法运算 238
10.3 FNC 22 - MUL/BIN乘法运算 241
10.4 FNC 23 - DIV/BIN除法运算 244
10.5 FNC 24 - INC/BIN加一 247
10.6 FNC 25 - DEC/BIN减一 249
10.7 FNC 26 - WAND/逻辑与 250
10.8 FNC 27 - WOR/逻辑或 252
10.9 FNC 28 - WXOR/逻辑异或 254
10.10 FNC 29 - NEG/补码 256

11. 循环·移位 - FNC 30～FNC 39 258

11.1 FNC 30 – ROR/循环右移 259
11.2 FNC 31 – ROL/循环左移 261
11.3 FNC 32 – RCR/带进位循环右移 263
11.4 FNC 33 – RCL/带进位循环左移 265
11.5 FNC 34 – SFTR/位右移 267
11.6 FNC 35 – SFTL/位左移 269
11.6.1 F1·F2可编程控制器的SFT(移位)指令替换 271
11.7 FNC 36 – WSFR/字右移 272

11.8 FNC 37 – WSFL/字左移 274
11.9 FNC 38 – SFWR/移位写入[先入先出/先入后出控制用] 276
11.10 FNC 39 – SFRD/移位读出[先入先出控制用] 279

12. 数据处理 - FNC 40 ~ FNC 49 281

12.1 FNC 40 - ZRST/成批复位 282
12.2 FNC 41 - DECO/译码 285
12.3 FNC 42 - ENCO/编码 288
12.4 FNC 43 - SUM/ON位数 290
12.5 FNC 44 - BON/ON位的判定 293
12.6 FNC 45 - MEAN/平均值 295
12.7 FNC 46 - ANS/信号报警器置位 297
12.8 FNC 47 - ANR/信号报警器复位 299
12.9 FNC 48 - SQR/BIN开方运算 300
12.10 FNC 49 - FLT/BIN整数→2进制浮点数转换 301

13. 高速处理 - FNC 50 ~ FNC 59 303

13.1 FNC 50 - REF / 输入输出刷新 304
13.1.1 使用REF指令前须知 306
13.2 FNC 51 - REFF/输入刷新(带滤波器设定) 307
13.2.1 使用REFF指令前须知 309
13.3 FNC 52 - MTR/矩阵输入 310
13.3.1 MTR指令的使用方法和注意要点 313
13.4 FNC 53 - HSCS/比较置位(高速计数器用) 314
13.4.1 使用高速计数器用指令时的通用注意事项 317
13.5 FNC 54 - HSCR/比较复位(高速计数器用) 320
13.6 FNC 55 - HSZ/区间比较(高速计数器用) 322
13.6.1 电源ON时比较结果置ON的程序[ZCP(FNC 11)指令] 325
13.6.2 表格高速比较模式(M8130) 327
13.6.3 频率控制模式(HSZ，PLSY指令)(M8132) 330
13.7 FNC 56 - SPD/脉冲密度 332
13.8 FNC 57 - PLSY/脉冲输出 336
13.9 FNC 58 - PWM/脉宽调制 341
13.10 FNC 59 - PLSR/带加减速的脉冲输出 344

14. 方便指令 - FNC 60 ~ FNC 69 349

14.1 FNC 60 - IST/初始化状态 350
14.1.1 IST指令的等价梯形图 352
14.1.2 导入IST指令的实例(工件传送设备举例) 353
14.2 FNC 61 - SER/数据检索 359
14.3 FNC 62 - ABSD/凸轮控制绝对方式 362
14.4 FNC 63 - INCD/凸轮控制相对方式 365
14.5 FNC 64 - TTMR/示教定时器 367
14.6 FNC 65 – STMR/特殊定时器 369
14.7 FNC 66 – ALT/交替输出 371
14.8 FNC 67 – RAMP/斜坡信号 373
14.9 FNC 68 – ROTC/旋转工作台控制 375
14.10 FNC 69 – SORT/数据排序 378

15. 外部设备I/O – FNC 70～FNC 79 381

15.1 FNC 70 – TKY/数字键输入 382
15.2 FNC 71 – HKY/16进制数字键输入 385
15.3 FNC 72 – DSW/数字开关 388
15.4 FNC 73 – SEGD/7段解码器 391
15.5 FNC 74 – SEGL/7SEG码时分显示 393
15.5.1 7段数码管的选用要领 396
15.5.2 根据7段数码管显示的规格选择参数n的要领 396
15.6 FNC 75 – ARWS/箭头开关 398
15.7 FNC 76 - ASC/ASCII数据输入 402
15.8 FNC 77 - PR/ASCII 码打印 404
15.9 FNC 78 - FROM/BFM 的读出 407
15.9.1 FROM/TO指令的共通事项(详细) 409
15.10 FNC 79 - TO/BFM的写入 411

16. 外部设备SER(选件设备) - FNC 80～FNC 89 413

16.1 FNC 80 - RS/串行数据传送 414
16.2 FNC 81 - PRUN/8进制位传送 416
16.3 FNC 82 - ASCI/HEX→ASCII的转换 418
16.4 FNC 83 - HEX/ASCII→HEX的转换 421
16.5 FNC 84 - CCD/校验码 424
16.6 FNC 87 - RS2/串行数据传送2 427
16.7 FNC 88 - PID/PID运算 430

17. 数据传送2 - FNC 100～FNC 109 433

17.1 FNC 102 - ZPUSH/变址寄存器的成批保存 434
17.2 FNC 103 - ZPOP/变址寄存器的恢复 436

18. 浮点数运算 - FNC 110～FNC 139 437

18.1 FNC 110 - ECMP/2进制浮点数比较 439
18.2 FNC 111 - EZCP/2进制浮点数区间比较 440
18.3 FNC 112 - EMOV/2进制浮点数数据传送 442
18.4 FNC 116 - ESTR/2进制浮点数→字符串的转换 443
18.5 FNC 117 - EVAL/字符串→2进制浮点数的转换 449
18.6 FNC 118 - EBCD/2进制浮点数→10进制浮点数的转换 454
18.7 FNC 119 - EBIN/10进制浮点数→2进制浮点数的转换 455
18.8 FNC 120 - EADD/2进制浮点数加法运算 457
18.9 FNC 121 - ESUB/2进制浮点数减法运算 458
18.10 FNC 122 - EMUL/2进制浮点数乘法运算 459
18.11 FNC 123 - EDIV/2进制浮点数除法运算 460
18.12 FNC 124 - EXP/2进制浮点数指数运算 461
18.13 FNC 125 - LOGE/2进制浮点数自然对数运算 463
18.14 FNC 126 - LOG10/2进制浮点数常用对数运算 465
18.15 FNC 127 - ESQR/2进制浮点数开方运算 467
18.16 FNC 128 – ENEG/2进制浮点数符号翻转 468
18.17 FNC 129 – INT/2进制浮点数→BIN整数的转换 469
18.18 FNC 130 – SIN/2进制浮点数SIN运算 471

18.19 FNC 131 – COS/2进制浮点数COS运算 472
18.20 FNC 132 – TAN/2进制浮点数TAN运算 473
18.21 FNC 133 – ASIN/2进制浮点数SIN-1运算 474
18.22 FNC 134 – ACOS/2进制浮点数COS-1运算 476
18.23 FNC 135 – ATAN/2进制浮点数TAN-1运算 478
18.24 FNC 136 – RAD/2进制浮点数角度→弧度的转换 480
18.25 FNC 137 – DEG/2进制浮点数弧度→角度的转换 482

19. 数据处理2 – FNC 140~FNC 149 483

19.1 FNC 140 – WSUM/算出数据合计值 484
19.2 FNC 141 – WTOB/字节单位的数据分离 486
19.3 FNC 142 – BTOW/字节单位的数据结合 488
19.4 FNC 143 – UNI/16位数据的4位结合 490
19.5 FNC 144 – DIS/16位数据的4位分离 492
19.6 FNC 147 – SWAP/高低字节互换 494
19.7 FNC 149 – SORT2/数据排序2 495

20. 定位控制 - FNC 150～FNC 159 499

20.1 FNC 150 – DSZR/带DOG搜索的原点回归 500
20.2 FNC 151 – DVIT/中断定位 502
20.3 FNC 152 – TBL/表格设定定位 504
20.4 FNC 155 – ABS/读出ABS当前值 505
20.5 FNC 156 – ZRN/原点回归 506
20.6 FNC 157 – PLSV/可变速脉冲输出 508
20.7 FNC 158 – DRVI/相对定位 510
20.8 FNC 159 – DRVA/绝对定位 512

21. 时钟运算 - FNC 160～FNC 169 514

21.1 FNC 160 – TCMP/时钟数据比较 515
21.2 FNC 161 – TZCP/时钟数据区间比较 517
21.3 FNC 162 – TADD/时钟数据加法运算 519
21.4 FNC 163 – TSUB/时钟数据减法运算 521
21.5 FNC 164 – HTOS/时、分、秒数据的秒转换 523
21.6 FNC 165 – STOH/秒数据的[時、分、秒]转换 525
21.7 FNC 166 – TRD/读出时钟数据 527
21.8 FNC 167 – TWR/写入时钟数据 528
21.9 FNC 169 – HOUR/计时表 530

22. 外部设备 - FNC 170～FNC 179 532

22.1 FNC 170 – GRY/格雷码的转换 533
22.2 FNC 171 – GBIN/格雷码的逆转换 534
22.3 FNC 176 – RD3A/模拟量模块的读出 535
22.4 FNC 177 – WR3A/模拟量模块的写入 536

23. 替换指令的介绍 - FNC 180 537

23.1 指令替换对照表 537

24. 其他指令FNC 181～FNC 189 538

24.1 FNC 182 – COMRD/读出软元件的注释数据 539
24.2 FNC 184 – RND/产生随机数 541
24.3 FNC 186-DUTY/产生定时脉冲 542
24.4 FNC 188 – CRC/CRC运算 544
24.5 FNC 189 – HCMOV/高速计数器传送 548

25. 数据块处理 - FNC 190～FNC 199 552

25.1 FNC 192 – BK+/数据块的加法运算 553
25.2 FNC 193 – BK-/数据块的减法运算 556
25.3 FNC 194～199-BKCMP=、>、<、<>、<=、>=/数据块比较 559

26. 字符串控制 - FNC 200～FNC 209 563

26.1 FNC 200 – STR/BIN→字符串的转换 564
26.2 FNC 201 – VAL/字符串→BIN的转换 568
26.3 FNC 202 – $+/字符串的结合 572
26.4 FNC 203 – LEN/检测出字符串的长度 574
26.5 FNC 204 – RIGHT/从字符串的右侧开始取出 576
26.6 FNC 205 – LEFT/从字符串的左侧开始取出 578
26.7 FNC 206 – MIDR/从字符串中的任意取出 580
26.8 FNC 207 – MIDW/字符串中的任意替换 582
26.9 FNC 208 – INSTR/字符串的检索 585
26.10 FNC 209 – $MOV/字符串的传送 587

27. 数据处理3-FNC 210～FNC 219 589

27.1 FNC 210 - FDEL/数据表的数据删除 590
27.2 FNC 211 - FINS/数据表的数据插入 592
27.3 FNC 212 - POP/读取后入的数据[先入后出控制用] 594
27.4 FNC 213 - SFR/16位数据n位右移(带进位) 597
27.5 FNC 214 - SFL/16位数据 n 位左移(带进位) 599

28. 触点比较指令 - FNC 220～FNC 249 601

28.1 FNC 224～230 - LD=、>、<、<>、<=、>=/触点比较 603
28.2 FNC 232～238 - AND=、>、<、<>、<=、>=/触点比较 606
28.3 FNC 240～246 - OR=、>、<、<>、<=、>=/触点比较 609

29. 数据表处理 - FNC 250～FNC 269 612

29.1 FNC256 - LIMIT/上下限限位控制 613
29.2 FNC 257 - BAND/死区控制 616
29.3 FNC 258 - ZONE/区域控制 619
29.4 FNC 259 - SCL/定坐标(不同点坐标数据) 622
29.5 FNC 260-DABIN/10进制ASCII→BIN的转换 626
29.6 FNC 261 - BINDA/BIN→10进制ASCII的转换 629
29.7 FNC 269 - SCL2/定坐标2(X/Y坐标数据) 632

30. 外部设备通信(变频器通信) - FNC 270～FNC 274 636

30.1 FNC 270 - IVCK/变频器的运行监视 637
30.2 FNC 271 - IVDR/变频器的运行控制 639

30.3 FNC 272 – IVRD/读取变频器的参数 641
30.4 FNC 273 – IVWR/写入变频器的参数 643
30.5 FNC 274 – IVBWR/成批写入变频器的参数 645

31. 数据传送3 – FNC 275～FNC 279 647

31.1 FNC 278-RBFM/BFM分割读出 648
31.1.1 RBFM(FNC 278)/WBFM(FNC 279)命令的通用事项 649
31.2 FNC 279-WBFM/BFM分割写入 652

32. 高速处理2-FNC 280～FNC 289 654

32.1 FNC 280 – HSCT/高速计数器表比较 655

33. 扩展文件寄存器控制 - FNC 290～FNC 299 660

33.1 FNC 290 – LOADR/读出扩展文件寄存器 661
33.2 FNC 291 – SAVER/成批写入扩展文件寄存器 663
33.3 FNC 292 – INITR/扩展寄存器的初始化 671
33.4 FNC 293 – LOGR/登录到扩展寄存器 674
33.5 FNC 294 – RWER/扩展文件寄存器的删除·写入 678
33.6 FNC295 – INITER/扩展文件寄存器的初始化 682

34. SFC程序和步进梯形图 685

34.1 SFC程序 685
34.1.1 概要 685
34.1.2 功能和动作说明 685
34.1.3 SFC程序的创建步骤 686
34.1.4 初始状态的使用和作用 689
34.1.5 停电保持(保持用)状态 690
34.1.6 RET指令的作用 690
34.1.7 编制SFC程序所需的预备知识 690
34.1.8 SFC流程的形态 696
34.1.9 分支·合并状态的程序 699
34.1.10 制作分支回路方面的规则 700
34.1.11 程序举例 705
34.2 步进梯形图 713
34.2.1 概要 713
34.2.2 功能和动作说明 713
34.2.3 步进梯形图的形式 714
34.2.4 步进梯形图的编写(SFC程序→STL程序) 715
34.2.5 编制步进梯形图所需的预备知识 716
34.2.6 分支·合并状态的程序 719
34.2.7 程序举例 723

35. 中断功能和脉冲捕捉功能 729

35.1 概要 729
35.2 通用事项 730
35.2.1 中断功能及脉冲捕捉功能的禁止方法 730
35.2.2 相关事项 731
35.2.3 使用上的注意事项(通用) 732
35.3 输入中断(通过外部信号中断)[不使用延迟功能] 734
35.3.1 输入中断(通过外部信号中断)[不使用延迟功能] 734
35.3.2 实用程序举例(短时间脉宽的测量程序) 738

35.4 输入中断(通过外部信号中断)[使用延迟功能]…… 740
35.5 定时器中断(一定周期的中断)…… 741
35.5.1 定时器中断(一定周期的中断)…… 741
35.5.2 实用程序举例(应用指令的定时器中断程序)…… 742
35.6 计数器中断 - 根据高速计数器的计数到位来中断…… 744
35.7 脉冲捕捉功能[M8170～M8177]…… 745

36. 特殊软元件的动作(M8000～,D8000～) 746

36.1 特殊软元件一览(M8000～,D8000～)…… 746
36.1.1 特殊辅助继电器(M8000～M8511)…… 746
36.1.2 特殊数据寄存器(D8000～D8511)…… 760
36.2 特殊软元件(M8000～，D8000～)的补充…… 775
36.2.1 RUN监控，初始脉冲的使用[M8000～M8003]…… 775
36.2.2 看门狗定时器时间[D8000]…… 776
36.2.3 检测到电池电压过低[M8005，M8006]…… 777
36.2.4 停电检测时间[D8008，M8008，M8007]…… 777
36.2.5 运算时间(监控)[D8010～D8012]…… 777
36.2.6 内部时钟[M8011～M8014]…… 778
36.2.7 实时时钟[M8015～M8019，D8013～D8019]…… 778
36.2.8 输入滤波器的调节[D8020]…… 780
36.2.9 电池[BATT(BAT)] LED,[ALM] LED 灭灯指令[M8030]…… 781
36.2.10 清除指令[M8031，M8032]…… 782
36.2.11 存储器保持停止[M8033](STOP中的输出保持)…… 782
36.2.12 禁止所有输出的指令[M8034]…… 783
36.2.13 RUN/STOP输入的独立操作[M8035～M8037]…… 783
36.2.14 恒定扫描模式[M8039，D8039](运算处理时间的固定化)…… 784
36.2.15 使用STL指令、SFC图实现程序的状态控制[M8040]…… 784
36.2.16 模拟量特殊适配器[M8260～M8299,D8260～D8299]…… 785

37. 故障的查看方法和错误代码一览 791

37.1 显示可编程控制器运行情况的LED及显示颜色…… 791
37.1.1 POWER(POW) LED [灯亮/闪烁/灯灭][A8/A8C]…… 791
37.1.2 RUN LED [灯亮/灯灭][A8/A8C]…… 792
37.1.3 BATT(BAT) LED [灯亮/灯灭][A8/A8C]…… 792
37.1.4 ERROR(ERR) LED [灯亮/闪烁/灯灭][A8/A8C]…… 792
37.1.5 L RUN LED [仅A8C-32MT-LT(-2)]…… 793
37.1.6 L ERR. LED [仅A8C-32MT-LT(-2)]…… 793
37.2 错误代码的查看方法及显示内容…… 794
37.2.1 使用HCP works的操作及查看方法…… 794
37.3 检测错误用的软元件的补充说明…… 795
37.3.1 错误的检测(M8060～/D8060～)…… 795
37.3.2 检测错误用的特殊软元件的动作关系…… 796
37.3.3 错误的检测时序…… 796
37.4 错误代码一览及解决方法…… 797

关于手册中使用的总称・简称的记载

简称・总称	名称
可编程控制器	
A8系列	A8系列可编程控制器的总称
A8可编程控制器 或是基本单元	A8系列可编程控制器基本单元的总称
A8C系列	A8C系列可编程控制器的总称
A8C可编程控制器 或是基本单元	A8C系列可编程控制器基本单元的总称
A2系列	A2系列可编程控制器的总称
A2C系列	A2C系列可编程控制器的总称 该产品只有日语的手册。
A1系列	A1系列可编程控制器的总称
功能扩展板	
功能扩展板	功能扩展板(下列型号)的总称
特殊适配器	
特殊适配器	高速输入输出特殊适配器、通信特殊适配器、模拟量特殊适配器的总称 但是，根据所使用的基本单元不同可以连接的设备也不同。有关可以连接的设备，请参考所使用的基本单元的用户手册[硬件篇]进行确认。
通信特殊适配器	通信特殊适配器(下列型号)的总称 HCA8C-C24-ADP
模拟量特殊适配器	模拟量特殊适配器(下列型号)的总称 A8-4AD-ADP、A8-4DA-ADP、A8-3A-ADP、A8-4AD-PT-ADP、 A8-4AD-PTW-ADP、A8-4AD-PNK-ADP、A8-4AD-TC-ADP
扩展设备	
扩展设备	输入输出扩展设备、特殊扩展设备的总称 但是，根据所使用的基本单元不同可以连接的设备也不同。有关可以连接的设备请参考所使用的基本单元的用户手册[硬件篇]进行确认。

简称・总称	名称
特殊功能单元/模块 或是特殊扩展设备	特殊功能单元、特殊功能模块的总称 但是，根据所使用的基本单元不同可以连接的设备也不同。有关可以连接的设备，请参考所使用的基本单元的用户手册[硬件篇]进行确认。
特殊功能单元	特殊功能单元的总称 但是，根据所使用的基本单元不同可以连接的设备也不同。有关可以连接的设备，请参考所使用的基本单元的用户手册[硬件篇]进行确认。
特殊功能模块	特殊功能模块的总称 但是，根据所使用的基本单元不同可以连接的设备也不同。有关可以连接的设备，请参考所使用的基本单元的用户手册[硬件篇]进行确认。
A8C系列 特殊功能模块	A8C-4AD等等
开放的现场总线CC-Link、CC-Link/LT	
HC-Link设备	HC-Link远程I/O站、专用电源的总称
编程工具	
编程工具	编程软件、手持式编程器的总称
编程 软件	编程软件的总称
HCP-works	型号为工业自动化（PLC，HMI，INV，SERVO）一体化编程软件
手册	
A8硬件篇手册	A8系列用户手册[硬件篇]
A8C硬件篇手册	A8C系列用户手册[硬件篇]
编程手册	A8/A8C系列编程手册[基本・应用指令说明书]
模拟量控制手册	A8/A8C系列用户手册[模拟量控制篇]
定位控制手册	A8/A8C系列用户手册[定位控制篇]

1. 前 言

在本章中，说明了关于A8/A8C可编程控制器编程的基本事项。

1.1 可编程控制器的编程语言

A8/A8C可编程控制器编程的有关特点有以下几点。

1.1.1 编程语言的种类

A8/A8C可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

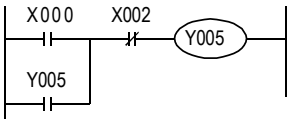
步	指令	软元件编号
0000	LD	X000
0001	OR	Y005
0002	ANI	X002
0003	OUT	Y005
⋮	⋮	⋮

2. 梯形图编辑

在图示的画面上画梯形图符号的梯形图编程方式

1) 特点
梯形图编程方式，就是使用顺序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



上述的指令表程序以梯形图表现

3. 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

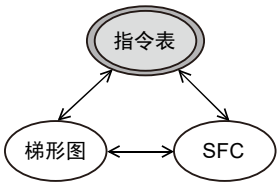
1) 特点
SFC(顺序功能图)程序就是根据机械的动作流程设计顺控的方式。

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，如果依照一定的规则编制，就可以倒过来转换成SFC图。

1.1.2 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。



2. 概要(顺控程序)

在本章中，说明了A8/A8C可编程控制器的基本功能。
在基本功能中，包括了可编程控制器的特点以及为了能让用户有效地使用该可编程控制器的功能所需的典型的功能介绍、参数、存储器运行等的项目，所以请在设计程序之前阅读。

2.1 便利功能的介绍

在A8/A8C可编程控制器中，有以下的指令功能。

2.1.1 输入处理用的便利功能

1. 高速输入的计数中单相或是双相的[高速计数器]功能

由于单相高速计数器，是采用中断的方式对从特定的输入继电器出来的高速脉冲进行处理，所以无论运算时间多少，在A8/A8C可编程控制器的情况下，最高可以执行100kHz的计数。
双相高速计数器，在A8/A8C可编程控制器的情况下，最高可以达到50kHz。
计数的结果或是使用高速计数器专用的比较指令立即输出，或是对高速计数器的计数值用高速计数器中断方式执行特定的程序处理。

*1. 仅A8可编程控制器可以连接。

高速计数器数量不足时连接特殊功能模块(高速计数模块)

如增设高速计数模块*2的硬件计数器，可以读取最高50kHz(2、4倍增的情况除外)的高速脉冲。

*2. 仅A8/A8C可编程控制器可以连接。

2. 读取最新输入信息中的[输入输出刷新]功能

在成批刷新方式下运行的可编程控制器的输入端子信息，可以在0步的运算之前被成批读入输入映像区。在执行END指令时成批输出输出信息。
输入输出刷新指令，就是可以在程序运算的过程中获得输入的最新信息，立即输出运算结果。

3. 变更输入继电器的时间常数中的[调节输入滤波器]功能

在可编程控制器的输入继电器中，为解决输入信号的波动和抗干扰，设置了约10ms的C-R滤波器。由于输入继电器中的X000～X017*1使用了数字式滤波器，所以可以通过顺控程序改变滤波器的值。

4. [脉冲捕捉]功能

脉冲捕捉功能是读取短时间的脉冲信号的一种方法。
在脉冲捕捉中监视特定的输入继电器发出的信号，在输入的同时通过中断处理来置位特殊辅助继电器。
脉冲捕捉可以简单地读取窄脉宽的输入，因此能应用广泛。
此外，想要通过特定的触发信号以中断的方式优先处理复杂的运算时，可以使用下面的[中断]功能。

5. 短时间脉冲的读取及优先处理中的3种[中断]功能

1) 输入中断
监视由特定的输入继电器发出的信号，最优先处理在输入的上升沿、或是下降沿指定的中断子程序。

2) 定时器中断
每隔一定时间，最优先处理指定的中断子程序。

3) 计数器中断*1
最优先处理用高速计数器的当前值指定的中断子程序。

*1.仅对应A8/A8C可编程控制器。

6. 脉宽/周期测量功能*1
能以10μs为单位测量由基本单元的输入端子(X000、X001、X003、X004)输入的脉冲的脉宽或周期。
在脉宽/周期测量方面，由于通过特殊辅助继电器可以方便地测量输入脉宽或者周期，因此可以用于多种用途。此外，使用多个输入端子，可以测量输入信号的延迟时间。

2.1.2 输出处理用的便利功能

1. 最新输入信息的输出中的[输入输出刷新]功能
在成批刷新方式下运行的可编程控制器的输入端子信息，可以在0步的运算之前被成批读入输入映像区。在执行END指令时成批输出输出信息。
输入输出刷新指令，就是可以在程序运算的过程中获得输入的最新信息，立即输出运算结果。

2. 脉冲串输出控制的[脉冲输出]功能

3. 定位控制的[定位]功能

2.1.3 支持顺控的功能

1. 将可编程控制器的运算周期固定化的[恒定扫描]模式
循环运算扫描方式的可编程控制器的扫描周期根据程序的执行内容不同而变化。
使用恒定扫描模式(M8039、D8039)，就可以固定运算周期，所以在运算中同步执行的指令也可以在固定的周期内被处理。

2. 将输出信号成批OFF的[禁止所有输出]模式
驱动特殊辅助继电器M8034，输出锁存内存会被清除。
因此，可编程控制器在继续运行的状态时将输出继电器(Y)都设置为OFF。
但是，各元件映像区内的输出Y的状态不会被清除。因此，如在编程工具等中进行监控，软元件为ON的情况也可能存在。

3. 在STOP时也能保持RUN中的输出状态的[存储器保持停止]功能
通过驱动特殊辅助继电器M8033，可以在原封不动保持RUN中的输出状态下STOP可编程控制器。

4. 保护程序的[关键字]的登录

作为防止编制的顺控程序被误写入・读出，可以登录关键字。
用HCP Works手持式编程器等的在线操作，通过关键字的指定方法还可以设定程序的保护级别。
此时，还可以执行称为[禁止变更程序，但是允许监控和变更当前值]的指定。

5. 对顺控程序附加[注释]

根据设定的参数，可以在程序内存中确保软元件注释(假名/英文数字/汉字)区域。

6. RUN中的程序写入

具备了可以在可编程控制器运行中更改程序的功能。
因此，无需停止机械装置，就可以更加高效率地调整和更改程序。

2.2 应用指令的介绍

1. 充实的基本性能
准备了以数据的传送和比较、四则运算、逻辑运算、数据的循环和移位等的基本应用指令为首的，输入输出刷新、中断、高速计数器专用比较指令、高速脉冲输出等的高速处理指令、在SFC控制上将机械控制的常规动作程序封装化的初始化状态指令等，追求“基本功能、高速处理、使用简单” 的各种规格。

2. 高级控制变得简单
由于备有多个将复杂的顺序控制封装化的便捷指令，因此可以减轻编制顺控程序的工作量，并且可以节约输入输出点数。
此外，为了适应更加高级的控制还备有浮点运算和PID运算。

2.2.1 主要应用指令

介绍从应用指令中选取的有代表性的指令。

1. 程序流程
- 条件跳转(CJ/FNC 00)
- 子程序调用(CALL/FNC 01)
- 允许中断(EI/FNC 04)
- 禁止中断(DI/FNC 05)
- 循环范围的起始(FOR/FNC 08)等

2. 传送・比较
- 比较(CMP/FNC 10)
- 触点比较(FNC 224～246)
- 浮点数比较
(ECMP/FNC 110、EZCP/FNC 111)
- 区间比较(ZCP/FNC 11)
- 高速计数器比较(FNC 53～55)
- 高速计数器表比较(HSCT/FNC 280)
- 数据传送(MOV/FNC 12)
- 浮点数数据传送(EMOV/FNC 112)
- 高速计数器传送(HCMOV/FNC 189)
- BCD转换(BCD/FNC 18)
- BIN转换(BIN/FNC 19)
- 格雷码的转换(FNC 170、171)等

3. 四则・逻辑运算
- BIN加法运算(ADD/FNC 20)
- BIN减法运算(SUB/FNC 21)
- BIN乘法运算(MUL/FNC 22)
- BIN除法运算(DIV/FNC 23)
- BIN加一(INC/FNC 24)
- BIN开方运算(SQR/FNC 48)
- 三角函数(FNC 130～135)
- 浮点数转换(FNC 49、118、119、129)
- 浮点数四则运算(FNC 120～123)
- 浮点数开方运算(ESQR/FNC 127)等

4. 循环・移位
- 循环右移(ROR/FNC 30)
- 循环左移(ROL/FNC 31)
- 带进位循环右移(RCR/FNC 32)
- 带进位循环左移(RCL/FNC 33)
- 位右移(SFTR/FNC 34)
- 位左移(SFTL/FNC 35)
- 字右移(WSFR/FNC 36)
- 字左移(WSFL/FNC 37)等

5. 数据处理
- 成批复位(ZRST/FNC 40)
- 译码(DECO/FNC 41)
- 编码(ENCO/FNC 42)
- ON位数(SUM/FNC 43)
- 平均值(MEAN/FNC 45)
- 字节单位的数据分离・结合(FNC 141、142)
- 16位数据的4位的结合・分离(FNC 143、144)
- 上下限限位控制(LIMIT/FNC 256)
- 死区控制(BAND/FNC 257)
- 区域控制(ZONE/FNC 258)
- 数据块处理(FNC 192～199)
- 字符串处理(FNC 200～209)等

6. 高速处理
- 输入刷新(REF/FNC 50)
- 输入刷新(带滤波器设定)(REFF/FNC 51)
- 脉冲密度(SPD/FNC 56)
- 脉冲输出(PLSY/FNC 57)
- 带加减速的脉冲输出(PLSR/FNC 59)等

7. 便捷指令・外围设备用的指令
   - 初始化状态(IST/FNC 60)
   - 示教定时器(TTMR/FNC 64)
   - 交替输出(ALT/FNC 66)
   - 斜坡指令(RAMP/FNC 67)
   - 旋转工作台控制(ROTC/FNC 68)
   - 数字键输入(TKY/FNC 70)
   - 数字开关(DSW/FNC 72)
   - 7段解码器(SEGD/FNC 73)
   - 7SEG时分显示(SEGL/FNC 74)
   - ASCII 数据输入(ASC/FNC 76)
   - BFM的读出、BFM的写入 (FNC 78、79、278、279)
   - 串行数据传送(FNC 80、87)
   - 模拟量旋钮(FNC 85、86)
   - 变频器通信(FNC 270～274)
   - HEX→ASCII转换(ASCI/FNC 82)
   - ASCII→HEX转换(HEX/FNC 83)
   - CRC运算(CRC/FNC 188)
   - 产生随机数(RND/FNC 184)
   - 时钟数据处理(FNC 160～167)
   - 计时表(HOUR/FNC 169)
   - 发出定时脉冲(DUTY/FNC 186)
   - 登录到扩展寄存器(LOGR/FNC 293)等

8. 复杂的控制
   - 数据检索(SER/FNC 61)
   - 数据排列(FNC 69、149)
   - PID运算(PID/FNC 88)等

9. 定位控制
   - 带DOG搜索的原点回归(DSZR/FNC 150)
   - 中断定位(DVIT/FNC 151)
   - 使用成批设定方式定位(TBL/FNC 152)
   - 读出ABS的当前值(ABS/FNC 155)
   - 原点回归(ZRN/FNC 156)
   - 可变速的脉冲输出(PLSV/FNC 157)
   - 相对定位(DRVI/FNC 158)
   - 绝对定位(DRVA/FNC 159)

2.3 模拟量·定位特殊控制

详细内容请参阅各产品的手册。

1. 模拟量输入输出控制
   - 模拟量输入
   - 模拟量输出
   - Pt100温度传感器输入
   - 热电偶温度传感器输入
   - 温度控制模块

2. 高速计数器
   - 高速计数器(硬件计数器，带倍增功能)

2.4 链接·通信

A8/A8C可编程控制器支持下列的通信功能。

1. HC-Link
   A8/A8C可编程控制器作为HC-Link的主站，可以构建出HC-Link系统。

2. HC-Link/LT
   A8/A8C可编程控制器作为HC-Link/LT的主站，可以构建出HC-Link/LT系统。
   远程I/O模块，在一般的X(输入)、Y(输出)软元件被分配给可以编写与通用输入输出相同的程序以控制其运行。

3. AS-I系统
   A8/A8C可编程控制器作为AS-i系统的主站，可以构建执行传感器层的网络系统。

4. 简易的PC间链接
   最多可连接8台可编程控制器，在这些产品之间自动地进行数据交换。

5. 并联链接
   连接2台可编程控制器，在这些产品之间自动地进行数据交换。

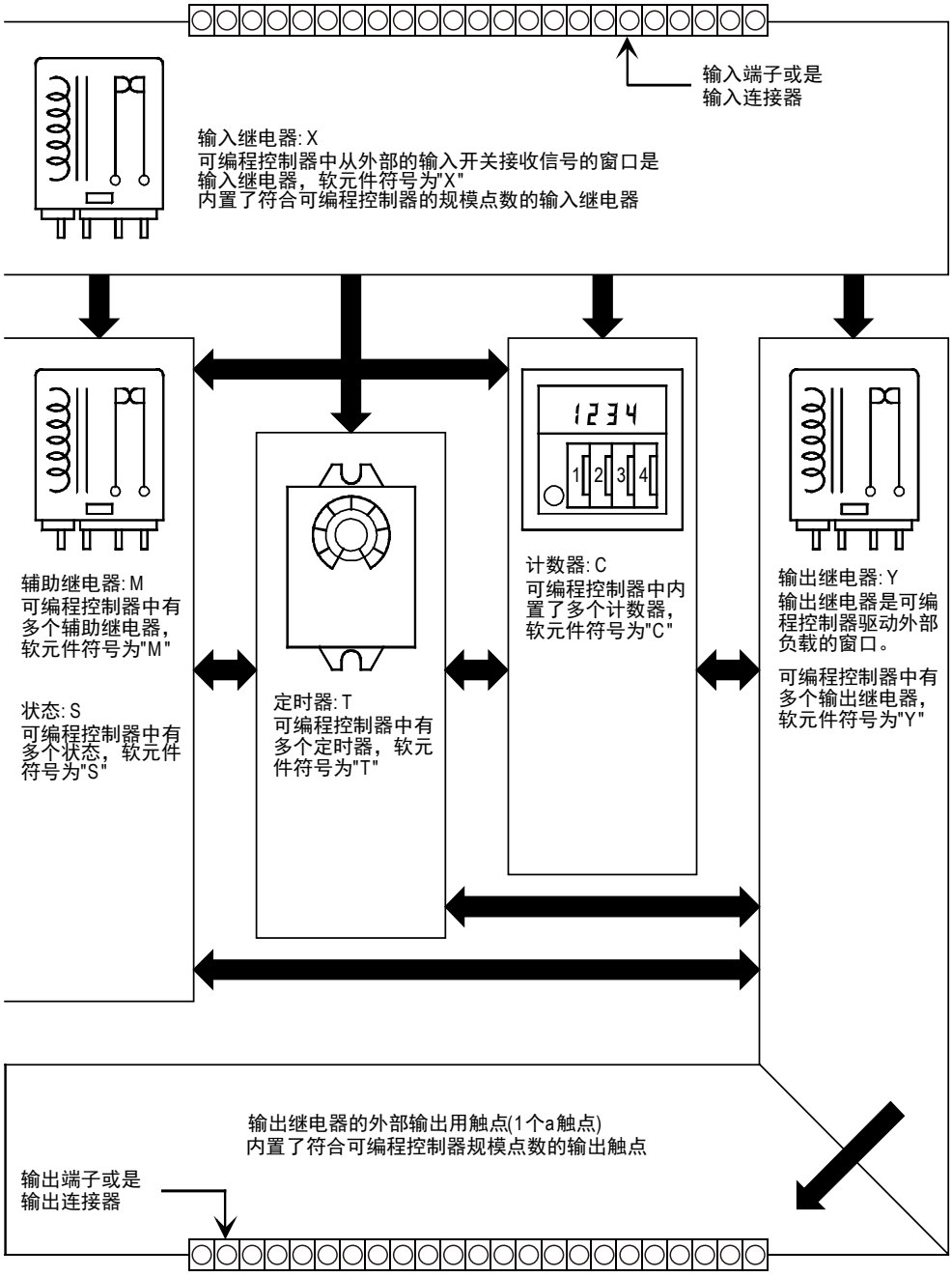
6. 无协议通信
   与条形码阅读器、打印机、计算机、测量仪器等的RS-232C/RS-485接口设备之间，可以实现无协议的串行通信。

2.5 构成可编程控制器的软元件的介绍

在A1/A2/A2C/A8/A8C可编程控制器中内置了多个继电器、定时器和计数器，无论哪个都有无数的a触点(常开触点)和b触点(常闭触点)。
连接这些触点和线圈，构成顺控回路。
而且，在可编程控制器中，备有作为保存数值数据用的记忆软元件的数据寄存器(D)、扩展寄存器(R)等。

2.5.1 各软元件的关系

箭头表示信号的接收。



2.5.2 软元件的动作及一览

1. 输入(X)・输出(Y)继电器

- 各基本单元中，都按照X000～X007、X010～X017…、Y000～Y007、Y010～Y017…分配了8进制的输入继电器、输出继电器的编号。
  扩展单元和扩展模块的编号，也是从基本单元开始按连接顺序分别取X、Y各自的8进制的连续编号。
- 在特定的输入继电器的输入滤波器中，使用了数字式滤波器，通过程序可以更改滤波器的值。因此，用于以高速读取为目的的用途中时，会分配这个输入继电器编号。(请参考滤波器调节、输入中断、高速计数器、各种应用指令等的说明。)

2. 辅助继电器(M)

- 可编程控制器内部带有的继电器就是辅助继电器。与输入输出继电器不同，是既不能读取外部的输入，也不能直接驱动外部负载的程序用的继电器。
- 还有即使断开可编程控制器的电源，也能够记忆ON/OFF状态的保持用的继电器。

3. 状态(S)

- 作为步进梯形图或是 SFC 表示工程编号使用的继电器。
- 不作为工程编号使用的时候，和辅助继电器相同，可以作为一般的触点/线圈来编程。
- 可以作为信号报警器用于诊断外部故障。

4. 定时器(T)

- 定时器就是，用加法计算可编程控制器中的1ms、10ms、100ms等的时钟脉冲，当加法计算的结果达到所指定的设定值时，输出触点就动作的软元件。
  根据基础的时钟脉冲，定时器可以测量0.001～3276.7秒。
- T192 ～ T199 是子程序和中断子程序专用的定时器。
  T250 ～T255 是100ms 的基准时钟的定时器，但是其当前值是累计方式的，所以定时器线圈的驱动输入即使OFF 也能保持当前值，进行累计。

5. 计数器(C)

计数器有以下几种。根据目的和用途不同可以分开使用。

1) 计数器(保持)用

计数器是可编程控制器的内部信号用的，其响应速度为通常数10kHz以下。

- 16位计数器: 增计数用，计数范围 1～32,767
- 32位计数器: 增计数/减计数用，计数范围 -2,147,483,648～+2,147,483,647

2) 高速计数器用[停电保持(保持)]

高速计数器与可编程控制器的运算无关，都执行几kHz的计数。

- 32位计数器: 上升/下降用，计数范围 -2,147,483,648～+2,147,483,647(单相单计数、单相双计数、双相双计数)在特定的输入继电器中分配。

6. 数据寄存器(D)

数据寄存器就是保存数据的软元件。
HCA系列可编程控制器的数据寄存器都是16位(最高位是正负位)，组合2个寄存器后可以处理32位(最高位是正负位)的数值。(有关数值范围请参考上述的“计数器”)
和其他的软元件相同，数据寄存器也有一般用的和停电保持(保持)用的。

7. 关于扩展寄存器(R)、扩展文件寄存器(ER)

扩展寄存器(R)是数据寄存器(D)的扩展软元件。A8/A8C可编程控制器的情况下，通过电池 进行停电保持。

A8/A8C可编程控制器也可以将扩展寄存器(R)的内容保存在扩展文件寄存器(ER)中。但是，A8/A8C可编程控制器，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。

8. 变址寄存器(V、Z)

在寄存器中，被称为变址(修饰)用的有V、Z两种寄存器。
V、Z是如下所示地附加在其他软元件上的。
[V0、Z0=5时]
D100V0=D105、C20Z0=C25←软元件编号+V□或是+Z□的值
数据寄存器和变址寄存器，可用于间接指定定时器和计数器的设定值，以及用于应用指令中。

9. 指针(P)(I)

在指针中，有分支用和中断用两种。

- 分支用指针(P)是用于指定CJ(FNC 00)条件转移和CALL(FNC 01)子程序调用的对象目的地。
- 中断用指针(I)是用于指定输入中断、定时器中断或是计数器中断的中断子程序。

10.常数(K)(H)(E)

可编程控制器中使用的各种数值中，K表示10进制数，H表示16进制数，E表示实数(浮点数)。
这些都用做定时器和计数器的设定值及当前值，或是应用指令的操作数。

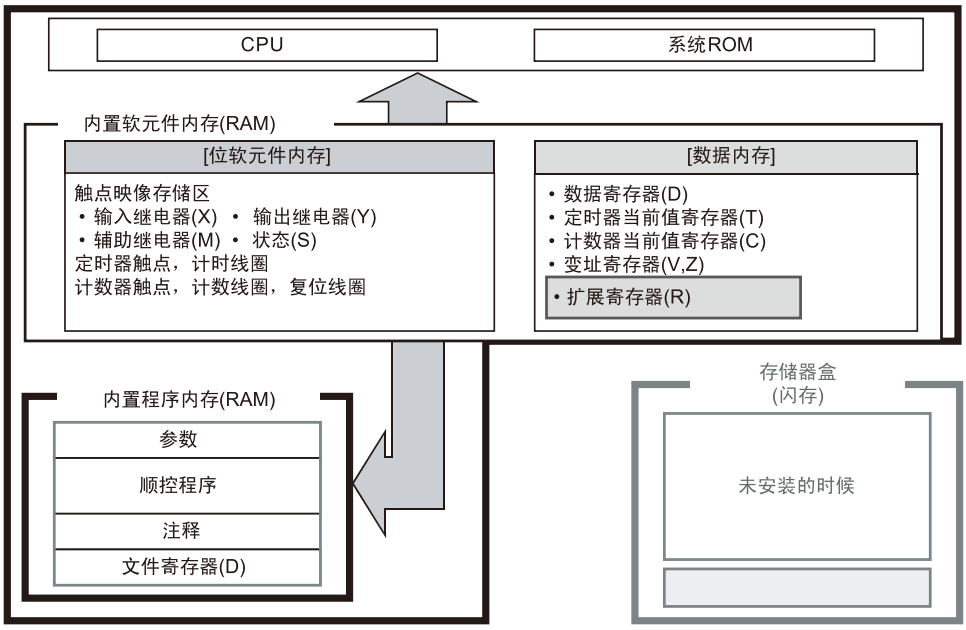
2.6 程序内存和软元件

2.6.1 存储器结构

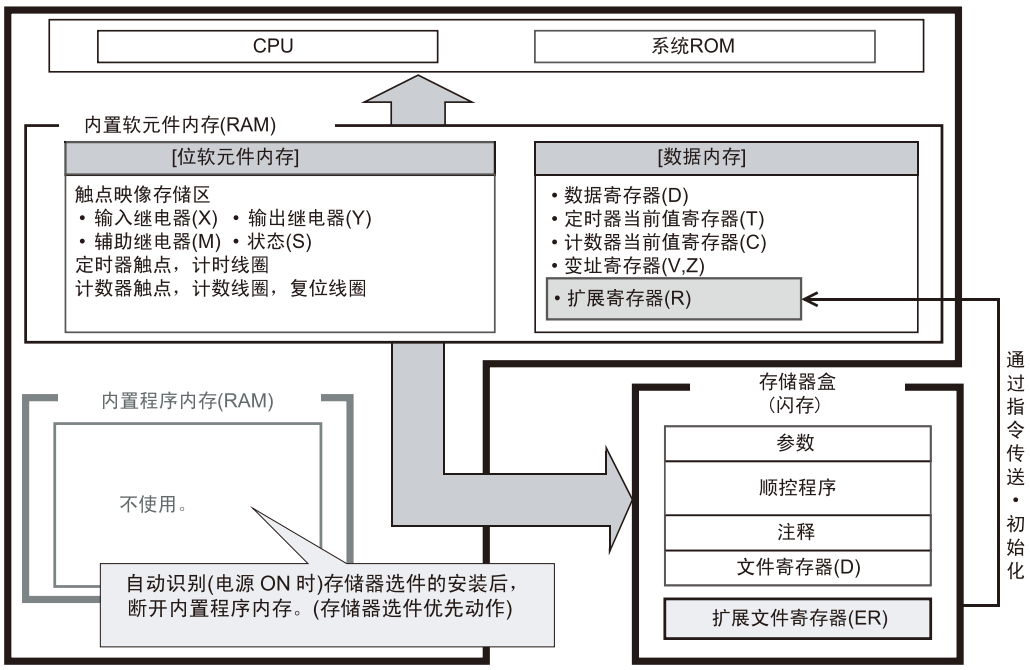
1. A8/A8C可编程控制器的情况下

在A8/A8C可编程控制器中标准配备了RAM存储器。通过安装存储器盒就可以更改存储器的种类。

1) 内置存储器的情况(未安装存储器盒)



2) 安装了存储器盒的情况(未使用内置程序内存)



2.6.2 存储器的动作和停电保持(电源ON/OFF、RUN/STOP)

1. 备份动作

A8/A8C可编程控制器的数据内存、位软元件内存以及程序内存的动作如下区分。

1) 程序内存的种类

项目		电源OFF	电源OFF→ON	STOP→RUN	RUN→STOP
参数		不变化*1			
顺控程序		不变化*1			
注释	根据参数的设定可确保	不变化*1			
文件寄存器		不变化*1			

2) 字软元件内存的种类

a) A8/A8C可编程控制器

项目		电源OFF	电源OFF→ON	STOP→RUN	RUN→STOP
数据寄存器(D)	一般用	清除		不变化	清除
				M8033=ON的时候不变化	
	停电保持用	不变化*2			
	文件用	不变化*1			
	特殊用	清除	初始值设定*3	不变化*3	
扩展寄存器(R)	停电保持用	不变化*2			
扩展文件寄存器(ER)*4	文件用	不变化			
变址寄存器(V、Z)	V、Z	清除		不变化	
定时器当前值寄存器(T)	100ms用	清除		不变化	清除
				M8033=ON的时候不变化	
	10ms用	清除		不变化	清除
				M8033=ON的时候不变化	
	累计100ms用	不变化*2			
	累计1ms用	不变化*2			
计数器当前值寄存器(C)	一般用	清除		不变化	清除
				M8033=ON的时候不变化	
	停电保持用	不变化*2			
	高速用	不变化*2			
时钟数据	当前值	不变化*2			

*1. A8/A8C可编程控制器中，不使用存储器盒时，如果电池电压比保持电压低，将不能正确保持程序内存的内容和软元件的状态。

*2. 如果电池电压比保持电压低时，将不能正确保持软元件的状态。

*3. 一部分的软元件在STOP→RUN的时候被清除。

*4. 需要选件存储器盒。

• 注意

电池的电压由于寿命等原因而降低时，不能正确保持程序(未使用存储器盒时)、停电保持(电池保持)、软元件和时钟的状态。这种情况下，执行停电保持软元件的清除，
再次传送程序(未使用存储器盒时)，而且，需要的情况下，请进行初始值和时钟的设定。

3) 位软元件内存的种类

a) A8/A8C可编程控制器

项目		电源OFF	电源OFF→ON	STOP→RUN	RUN→STOP
触点映像存储器 (X、Y、M、S)	输入继电器(X)	清除		不变化	清除
				M8033=ON的时候不变化	
	输出继电器(Y)	清除		不变化	清除
				M8033=ON的时候不变化	
	一般辅助用继电器(M)	清除		不变化	清除
				M8033=ON的时候不变化	
	停电保持用辅助继电器(M)	不变化			
	特殊辅助继电器(M)	清除	初始值设定*1	不变化*1	
	一般用状态(S)	不变化			
	停电保持用状态(S)	不变化			
	信号报警器(S)	不变化			
定时器触点 计时线圈(T)	100ms用	清除		不变化	清除
				M8033=ON的时候不变化	
	10ms用	清除		不变化	清除
				M8033=ON的时候不变化	
	累计100ms用	不变化			
	累计1ms用	不变化			
计数器触点 计数线圈 复位线圈(C)	一般用	清除		不变化	清除
				M8033=ON的时候不变化	
	停电保持用	清除		不变化	清除
				M8033=ON的时候不变化	
	高速用	清除		不变化	清除
				M8033=ON的时候不变化	

*1. 一部分的软元件在STOP→RUN的时候被清除。

2.6.3 有关停电保持方式的种类

可编程控制器的程序内存和内置软元件的停电保持有以下几种。

1. 电池后备方式

A8/A8C可编程控制器

项目	内容
保持内容	使用电池，可以保持可编程控制器内置的RAM存储器、停电保持软元件、以及时钟数据。
维护	寿命基准为5年(周围温度:25℃)。 有关更换基准，请参考各可编程控制器的用户手册[硬件篇]。
注意事项	1) 电池电压过低时，顺控程序和其他的保持内容都会消失。 2) 安装存储器盒选件(闪存)的时候，无需对顺控程序进行电池后备。

2. 存储器盒后备方式

A8/A8C可编程控制器

项目	内容
保持内容	1) 使用存储器盒中内置的闪存，可以保持顺控程序。 2) 对停电保持软元件、时钟数据进行保持时需要同时使用电池后备。
维护	无需维护。
注意事项	改写的次数有限制。 (参考使用的可编程控制器主机的硬件篇手册)

2.6.4 一般用软元件和停电保持软元件的变更

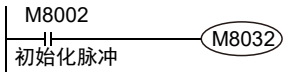
1. 将停电保持软元件作为非停电保持软元件使用的时候
   A8/A8C可编程控制器，根据参数设定情况，可将部分停电保持软元件的一部分更改成非停电保持软元件。
   停电保持专用的软元件不可以更改成非停电保持软元件。
   这种情况下，在程序中使用初始化脉冲(M8002)清除保持软元件，这样就可以将其作为非停电保持软元件使用。

2. 将非停电保持软元件作为停电保持软元件使用时
   A8/A8C可编程控制器，根据参数设定情况，可将非停电保持软元件更改成停电保持软元件。

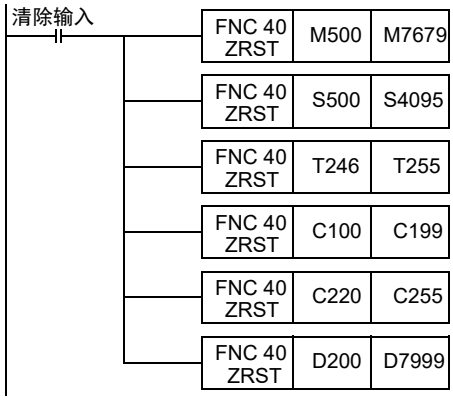
2.6.5 停电保持软元件的初始化方法

在停电保持软元件的初始化方法中，有利用外围设备的PLC存储器全部清除、通过特殊辅助继电器(M8032)保持存储区的全部清除、ZRST指令等。在此主要对两种方法加以说明。

1. M8032(保持存储区的全部清除)
   M8032置ON时，执行停电保持软元件*1(包括定时器、计数器的复位线圈)的清除。虽然M8032可以对外围设备强制执行ON/OFF操作或在顺控程序中ON/OFF，但请注意当M8032置ON时，不能使停电保持软元件ON。
   另外，请注意，程序中M8032为ON时，清除停电保持软元件的时序会在M8032置ON后变为END指令。
   程序举例: 清除所有停电保持软元件的程序举例。



2. ZRST(FNC 40)指令(成批复位)
   根据ZRST指令，执行各个软元件的清除。(根据ZRST指令清除时，因为可以指定软元件的范围，所有一部分保持软元件也能够被清除。)
   程序举例：在A8/A8C可编程控制器中，清除停电保持(电池保持)软元件的程序举例。软元件范围见如下表所示。



	电池保持的软元件范围
辅助继电器	M500～M7679
状态	S500～S4095
Timer	T246～T255
计数器	C100～C199, C220～C255
数据寄存器	D200～D7999

2.7 参数的种类和设定

设定参数就是为了可编程控制器能够运行而进行的环境设定。
在HCA可编程控制器中，几乎大都可以在出厂时的初始值的状态下使用，但是需要增加存储器盒或设定注释的容量等有关存储器的内容，以及需要进行串行通信口的通信设定的情况下，请使用计算机等的编程工具更改设定。

2.7.1 参数一览

在参数中，包括下列的设定内容。

区分	项目	内容
内存容量	存储器	可以输入的顺控程序步数的最大值。 1) 由内置存储器，或是存储器盒的大小来决定上限。 2) 程序内存、文件寄存器、注释领域、其他的特殊设定容量包含在该存储器容量中。
	注释区域	在程序内存中包含注释。 1) 由于可编程控制器中留有注释，维护的时候就便于理解内容。 2) 每指定1个块，在可编程控制器中就加入50点的注释，但是由于使用了500步存储器容量，所以减少了程序内存容量。
	文件寄存器	在程序内存中包含数据寄存器。 1) 归纳顺控程序和加工设定值等的控制数据，可以便于使用。 2) 每指定1个块，就可以制作500点的文件寄存器，但是由于使用了500步存储器容量，所以减少了程序内存容量。
	其他 特殊设定容量	1) 设定使用/不使用特殊模块/单元的初始值设定功能。*1 使用的情况下，由于占用4000步(8个块)存储器容量，所以减少了程序内存容量。 2) 设定使用/不使用TBL(FNC 152)指令进行定位设定(常数、设定表)。使用的情况下，由于占用9000步(18个块)存储器容量，所以减少了程序内存容量。 3) 设定使用/不使用内置HC-Link/LT的功能。 使用的情况下，由于占用500步(1个块)存储器容量，所以减少了程序内存容量。
软元件设定	锁存范围设定	可以更改可编程控制器内部软元件的停电保持区域和非停电保持区域的范围。
设定I/O分配	设定I/O分配	不能写入到可编程控制器中。 根据系统配置设定范围的话，可以使用HCP Works的程序检查进行检查。
	特殊单元的设定	对特殊模块/单元的编号设定缓冲存储器(BFM)的初始值。 但是，需要设定内存的容量。
	内置HC-Link/LT设定	设定传送速度、点数模式、站信息。但是，设定站信息时需要设定内存容量。
PLC系统的设定(1) [PLC模式]	无电池模式*1	为了拆下可编程控制器的电池也能够运行所需的设定。 通过设定无电池模式，就可以自动停止检测电池电压下降错误，以及自动对停电保持软元件进行初始化。
	电池模式	为了在可编程控制器上安装电池并运行所需的设定。 通过选择使用电池，可以停电保持一般软元件。可以更改为停电保持(电池保持)。
	调制解调器初始化	将指定的AT指令作为初始化指令自动传送给连接在串行通信口上的调制解调器。
	RUN端子的输入设定	设定是否将可编程控制器的一个输入端子作为RUN输入使用。
	RUN端子的输入编号	指定上述的RUN输入的输入编号。 X000～X017 (A8-16M□、A8C-16M□)
PLC系统的设定(2) [串行通信]	串行通信口的动作设定	通过指定计算机画面上的各内容来对应下面的设定。 设定通信格式(D8120、D8400、D8420) 设定站号(D8121、D8421) 设定超时判定(D8129、D8409、D8429)

区分	项目	内容
定位设定	常数的设定	设定最高速度、偏离速度、爬行速度、原点回归速度、加速时间、减速时间、DVIT指令用*1 中断输入设定。但是，需要设定存储器容量。
	详细设定	设定动作表。但是，需要设定存储器容量。
其他	关键字	为了防止顺控程序被误写入・读出而设置的保护。 关键字(8个字符)、第2关键字(8个字符) 可以指定A～F、0～9的英文数字。
	程序标题	可以设定作为程序标题的字符串。

*1. 只对应A8/A8C可编程控制器。

2.7.2 参数的初始值及可以变更的工具

1) A8/A8C可编程控制器

项目		HCP works 的初始值	变更范围
存储器容量(步)	程序容量	16000	请参考2.7.3项
	注释容量	0	
	文件寄存器	0	
	特殊单元的初始值设定	不使用	
	定位的设定	不使用	
	内置HC-Link/LT设定	不使用	
锁存范围(停电保持区域)	辅助继电器[M]	500~1023	0~1023
	状态[S]	500~999	0~999
	计数器[C](16位)	100~199	0~199
	计数器[C](32位)	220~255	200~255
	数据寄存器[D]	200~511	0~511
程序标题		未登录	请参考2.7.1项
关键字的登录		未登录	
无电池模式		无	
调制解调器的初始化指定		无设定	
RUN端子输入		无	
串行通信口的动作设定		无设定	

2.7.3 存储器容量设定范围

◎: 内置存储器容量○: 通过更改常数可以设定

设定存储器容量		注释容量设定 单位: 块	文件寄存器 容量设定 单位: 块	特殊模块的 初始值设定*1 单位: 块	定位的设定*2 单位: 块	内置CC-Link/LT 设定 单位: 块
步数	A8/A8C					
2000	○	0~3	0~3	—	—	1
4000	○	0~7	0~7	—	—	1
8000	○	0~15	0~14	8	—	1
16000	○	0~31	0~14	8	18	1
32000	○	0~63	0~14	8	18	1
64000	◎	0~127	0~14	8	18	1

设定存储器容量时的注意事项

各容量的设定，为每1个块减少500步存储器容量。
各设定需要满足以下公式。

存储器容量的设定范围 ≧ 注释容量设定 + 文件寄存器容量设定 + 特殊功能模块的初始值设定*1 + 定位的设定*2 + 内置HC-Link/LT设定

1) 注释的容量设定为每1个块中有50点的软元件注释。
2) 文件寄存器的容量设定为每1个块中有500点的文件寄存器(16位)。
3) 特殊功能模块的初始值设定*1使用8个块(4000步)。
4) 定位的设定*2使用18个块(9000步)。
5) 内置HC-Link/LT设定*3使用1块(500步)。

*1. 仅A8/A8C可编程控制器对应
HCP Works以上可以设定。

*2. A8/A8C可编程控制器HCP Works以上。

注意要点

- 更改了存储器容量的设定时，务必将程序和参数一起写入可编程控制器。
仅将参数写入到可编程控制器中的话，在可编程控制器一侧有时候会出现程序错误(参数错误、回路错误、语法错误等)。

例) 基本指令LD的指令执行时间
标准模式: 0.21 μs 扩展模式: 0.42 μs

2.7.4 存储器盒适合的机型

1. A8/A8C可编程控制器

型号	最大步数	存储器类型	允许写入次数	备注
A8-FLROM-64	64000	闪存	1万次	有保护开关
A8-FLROM-16	16000	闪存	1万次	有保护开关
A8-FLROM-64L	64000	闪存	1万次	有保护开关， 带程序传送功能

2.7.5 关键字

如在可编程控制器中登录了关键字，就可以限制(访问限制)编程工具、显示模块、人机界面更改可编程控制器的程序、监控、变更当前值的功能。

1. 关键字的种类决定的不同动作

根据登录的关键字种类的不同，关键字的更改、删除、解除的动作也不同。

登录的 关键字	输入 关键字	更改	删除	解除	概要
不能解除保护	—	×	×	×	不能更改、删除、解除的保护。
关键字	关键字	○	○	○	可以更改、删除、解除的关键字。
关键字+ 第2关键字	关键字+ 第2关键字	○	○	○	可以更改、删除、解除的关键字+第2关键字。
关键字+ 第2关键字+ 客户关键字	关键字+ 第2关键字	○	○	○	可以更改、删除、解除的关键字+第2关键字。
	客户 关键字	×	×	○	可以解除的关键字

3. 关键字的设定和访问的限制

<table>
<tr><th>设定个数</th><th>外围设备和访问限制</th></tr>
<tr><td>选择了不可解除的保护时</td><td>根据所选择的登录条件有以下的动作。
此外，一旦加以限制，限制将会变为不可解除状态。
想解除限制时，或者可编程控制器中再次写入程序时，请全部清除可编程控制器存储器。
<table>
<tr><th rowspan="2">登录条件</th><th colspan="2">程序</th><th rowspan="2">监控</th><th rowspan="2">更改当前值</th></tr>
<tr><th>读出</th><th>写入</th></tr>
<tr><td>禁止写入</td><td>○</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止读出/写入</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止所有的在线操作</td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
</table></td></tr>
<tr><td>设定了客户关键字时</td><td>根据所选择的登录条件有以下的动作。
客户关键字中的关键字不能删除。
<table>
<tr><th rowspan="2">登录条件</th><th colspan="2">程序</th><th rowspan="2">监控</th><th rowspan="2">更改当前值</th></tr>
<tr><th>读出</th><th>写入</th></tr>
<tr><td>禁止写入</td><td>○</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止读出/写入</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止所有的在线操作</td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
</table></td></tr>
<tr><td>设定了关键字和
第2关键字的场合</td><td>根据所选择的登录条件有以下的动作。
<table>
<tr><th rowspan="2">登录条件</th><th colspan="2">程序</th><th rowspan="2">监控</th><th rowspan="2">更改当前值</th></tr>
<tr><th>读出</th><th>写入</th></tr>
<tr><td>禁止写入</td><td>○</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止读出/写入</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止所有的在线操作</td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
</table></td></tr>
<tr><td>仅仅设定了关键字时</td><td>根据关键字(8个字符)的开头字符有以下的动作。
<table>
<tr><th rowspan="2"></th><th rowspan="2">关键字的
开头文字</th><th colspan="2">程序</th><th rowspan="2">监控</th><th rowspan="2">更改当前值</th></tr>
<tr><th>读出</th><th>写入</th></tr>
<tr><td>禁止所有操作</td><td>A,D～F,0 ～9</td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>禁止
误写入•读出</td><td>B</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>禁止误写入</td><td>C</td><td>○</td><td>×</td><td>○</td><td>○</td></tr>
</table></td></tr>
<tr><td>未设定关键字的场合</td><td>允许所有操作</td></tr>
</table>

选择了不可解除的保护时的注意事项

- 一旦加以限制，限制将变为不可解除状态。
  想解除限制时，或者可编程控制器中再次写入程序时，请将可编程控制器存储器全部清除。

登录关键字时的注意事项

- 关键字的作用就是针对用户编制的程序，限制从外围设备进行访问。
  请务必注意保管好关键字。
  根据编程工具、登录关键字的内容，如果遗忘关键字的话，用编程工具不能对可编程控制器进行在线操作。

2.7.6 特殊功能模块的初始值设定【HCP works】

A8 A8C

在特殊单元的初始值设定中，可以用HCP works将A8/A8C可编程控制器上连接的特殊功能模块/单元的缓冲存储器(BFM)的初始值作为参数进行设定。
如果使用这个参数，则对于需要初始(初始化)设定的特殊功能模块/单元而言，无需在用户程序中进行初始(化)设定。使用4000步(8块)存储器容量。

2.7.7 定位设定【HCP works】

A8 A8C

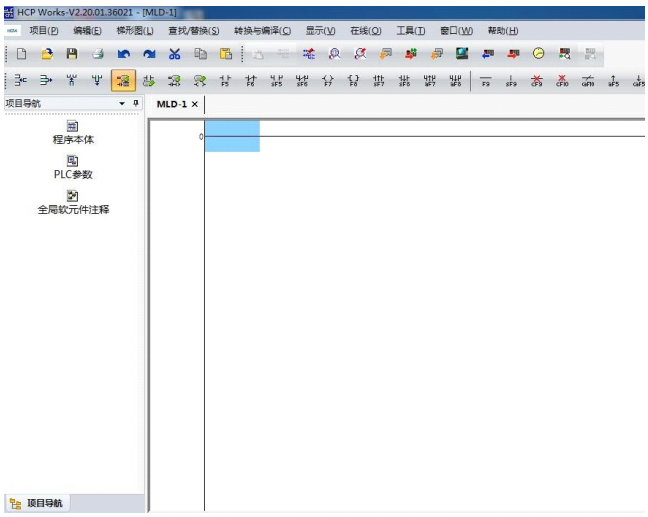
定位设定是在A8可编程控制器或A8/A8C可编程控制器中，对增加的TBL(FNC 152)指令的表、参数进行设定。
使用TBL(FNC 152)指令的场合，必须设定这个参数。
使用9000步(18块)存储器容量。

2.7.8 使用HCP works设定参数

说明了使用HCP works设定参数的步骤。

1 打开参数的设定。

双击画面左边的工程目录中的[Parameter]→[PLC parameter]。
没有显示出工程目录时，选择菜单栏[View]→[Project data list]。



2 设定存储器容量。



画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
Memory capacity	设定程序内存的容量。初始值: 16000*1	参考2.7.3项
Comments capacity	设定保存在可编程控制器中的注释容量。初始值: 0 软元件注释50点/1块(500步)	
File register capacity	设定文件寄存器的容量。初始值: 0 文件寄存器500点/1块(500步)	
Program capacity	显示在顺控程序中可以使用的步数。	
Special Funtion Memory capacity	设定特殊功能模块的初始值设定及定位设定有效还是无效。	—
Special Funtion Block Settings*2	如选择，特殊功能模块/单元的初始值设定功能变成有效。(变成有效后，在“I/O分配的设定”选项卡中会显示特殊功能模块的设定。)	—
Positioning Instruction Settings	如选择，TBL(FNC 152)指令的设定功能变成有效。 (变成有效后会显示“定位的设定”选项卡)	—
Built-in HC-Link/LT Setup*3	如选择，HC-Link/LT的站信息设定功能变成有效。	

*1. HCP works，可编程控制器时的初始值为8000步。

*2. 只对应A8/A8C可编程控制器。

*3. 只对应A8C。

注意要点

例) 基本指令LD的指令执行时间
标准模式: 0.21 μs 扩展模式: 0.42 μs

3 软元件的设定。

1. 点击“Device”选项卡后设定软元件。

只能设定A8/A8C可编程控制器。

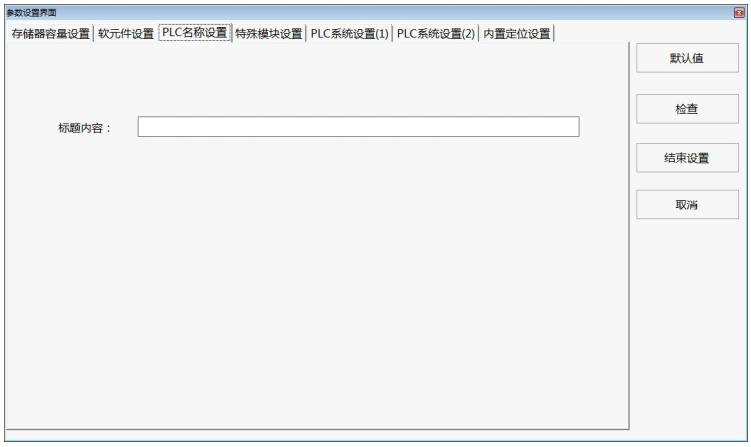


画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
Supplemental relay	设定辅助继电器的停电保持范围。初始值: 500～1023	0~1023
State	设定状态的停电保持范围。初始值: 500～999	0~999
Timer	不能更改设定。	—
Counter(16 bit)	设定计数器(16位)的停电保持范围。初始值: 100～199	0~199
Counter(32 bit)	设定计数器(32位)的停电保持范围。初始值: 220～255	220~255
Data register	设定数据寄存器的停电保持范围。初始值: 200～511	0~511
Extended register	固定为停电保持用的软元件。	—

PLC名称的设定。

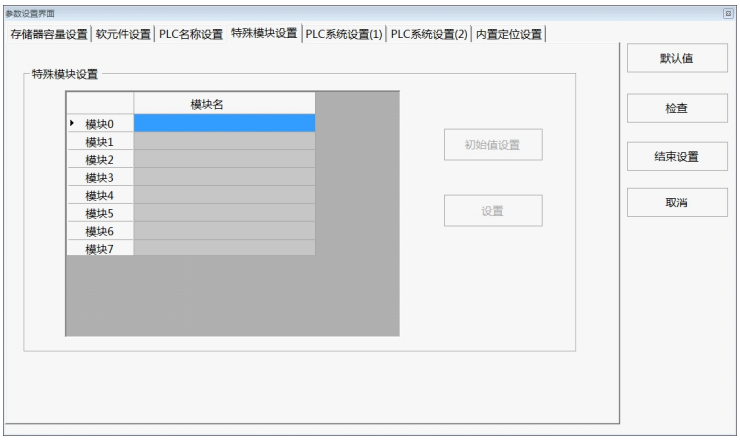
1. 点击“PLC Name”选项卡后，输入程序标题。



设定项目	设定内容	设定范围
Title	以半角32个字符(全角16个字符)设定程序标题。	半角32个字符 (全角16个字符)

5 I/O分配、特殊模块的初始值设定、内置HC-Link设定。

1. 点击“ I/O assignment ”选项卡后，设定I/O分配、特殊功能模块/单元的单元名称、内置HC-Link设定。
执行“Special Unit”时，需要在“Memory capacity”选项卡中选择“Special Function Block Settings”。
“内置HC-Link设定”执行站信息的设定时，需要在“存储器容量设定”选项卡中选择“内置HC-Link/LT设定”。



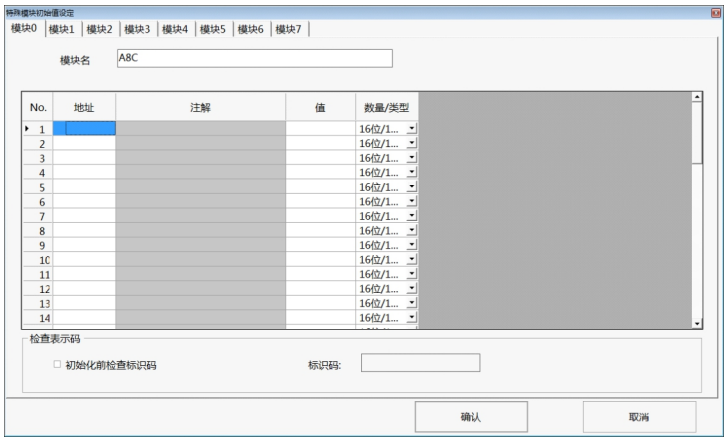
画面是A8/A8C可编程控制器的例子。

设定项目		设定内容	设定范围
I/O	Input relay	根据系统配置事先设定了范围的话，使用HCP works可以在输入时可以进行检查。	A8/A8C：0~377
	Output relay		
Special Unit*1 (需要设定存储器容量。)	Unit No.	特殊功能模块/单元的单元号。	—
	Unit Name	对设定初始值的特殊功能模块/单元的名称进行设定。	半角32个字符 (全角16个字符)
	Up	选中的行移动到上一行。(和上一行替换。)	—
	Down	选中的行移动到下一行。(和下一行替换。)	—
	Set...	显示选中的单元号的“特殊模块的初始值设定”对话框。 →参照后页	—
	Delete	对选中的模块的设定内容进行删除。	—
	Delete All	对已设定的“特殊功能模块的设定”进行全部清除。	—
	Read PLC data	从连接的可编程控制器中读出“特殊功能模块的设定”。	—
Built-in CC-Link/LT Setup*2	Set	显示“内置设定”对话框。	—

*1. 只有A8/A8C可编程控制器可以设定。

*2. 只有A8C可以设定。

2. 在“Special Function Block Settings”对话框中设定特殊功能模块/单元的初始值。
只能设定A8/A8C可编程控制器。



画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
“Unit No.”选项卡	选择要设定的特殊功能模块/单元的模块号。	—
Unit Name	对设定初始值的特殊功能模块/单元的名称进行设定。 (在“I/O分配的设定”选项卡中显示已经设定的内容)	半角32个字符 (全角16个字符)
No.	在选中的单元号中进行初始值设定的顺序。 可以在No.1～98之间进行设定。	—
Address	以10进制数值对要设定初始值的缓冲存储区地址(BFM号)进行设定。	*1
Comment	登录了软元件注释后显示。 上述画面中，在U0\G0(单元号为0、BFM#0)的软元件注释中已经设定了“指定通道1～通道4的输入模式”时的显示。	—
Value	在要设定初始值的缓冲存储区地址(BFM号)中设定设定值。设定值的数据长度、类型都在“大小/类型”中指定。	*2
Size/Type	在缓冲存储区中要设定的值的大小、类型如下所示选择。 16位/10进制 32位/10进制 16位/16进制 32位/16进制	—
Insert	在选择的位置中插入行。	—
Delete	删除选中的行。	—
Up	选中的行移动到上一行。(和上一行替换。)	—
Down	选中的行移动到下一行。(和下一行替换。)	—
Check the identification Code before initializing	如选中，就对初始化前的特殊功能模块/单元的机型代码进行检查。	—
Identification code	对要设定的特殊功能模块/单元的机型代码进行设定。	*3

*1. 请将缓冲存储区地址(BFM号)设置成连接的特殊功能模块/单元持有的编号。

*2. 缓冲存储区地址(BFM)中设定的值，务必要遵从连接的特殊功能模块/单元的设定范围。

*3. 请参考连接的特殊功能模块/单元的手册。

3. 点击了[OK]键后设定结束，关闭“Special Function Block Settings”对话框。

4. 在“内置HC-Link/LT设定”对话框中进行内置HC-Link/LT的设定。
仅可设定A8C可编程控制器。

设定项目	设定内容	设定范围
Transmission speed	内置HC-Link/LT的Transmission speed如下所示选择。 2.5Mbps 625kbps 156kbps	—
Point Mode	内置HC-Link/LT的点数模式如下所示选择。 16点模式 4点模式	—
No.	内置HC-Link/LT的站号。 站号可以在1～64范围内设定。	—
Station type	内置HC-Link/LT的站类别如下所示选择。 远程I/O站输入型 远程I/O站输出型 远程I/O站输入输出型 远程设备站*1	—
I/O pts	远程I/O站和远程设备站的点数如下所示选择。 1～16,32,48,64*2	1～16,32,48,64*2
Reserve station select	设定内置CC-Link/LT的预约站/无设定。	—
Up	选中的行移动到上一行。(和上一行替换。)	—
Down	选中的行移动到下一行。(和下一行替换。)	—
Insert	在选择的位置中插入行。	—
Delete	删除选中的行。	—
Read Buffer memory	读出内置HC-Link/LT的Transmission speed、点数模式和站信息。	—

*1. 使用远程设备站时，请将点数模式设定为16点模式。4点模式的情况下，不能选择远程设备站。
此外，站号只能设定在49～64之间。

*2. 32、48、64在站类别中选择远程设备站时有效。

5. 点击了[OK]键后设定结束，关闭“内置HC-Link/LT设定”对话框。

6 设定PLC的系统设定(1)。

点击“PLC system(1)”选项卡，设定“Battery less mode”、“Modem initialized”、“RUN terminal input”。

A8/A8C可编程控制器



设定项目	设定内容	设定范围
Battery less mode*1	拆下电池后无电池运行时，选择该项。会自动执行电池报错灯的灭灯以及停电保持区域的软元件清除的操作。	—
Battery mode	安装电池并运行时，选择该项。	—
MODEM initialized	在可编程控制器上电时，对调制解调器进行自动初始化的时候设定。	—
Run terminal input	要将输入端子(X)中的1点作为RUN输入的情况下。	无 X000～X017*2

*1. 仅可设定A8/A8C可编程控制器。

*2. A8/A8C可编程控制器 (14点型)是X000～X007。

7 设定PLC的系统设定(2)。

1. 点击“PLC system(2)”选项卡。
2. 只有在对扩展的可编程控制器串行通信口进行通信设定时，选择要设定的通道，选中“Operate Communication Setting”。
   不设定的情况下，请勿选择“Operate Communication Setting”。

请勿选择



画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
通道的选择	选择设定为串行通信口的通道。	CH1、CH2
Operate communication setting	选中的串行通信口用于“计算机链接”、“无协议通信”、“变频器通信”时做选择。 在使用HCP-works传送・监控顺控程序，或是在简易PC间链接、并联链接下使用时，请勿做选择。	—
Protocol	需要根据具体用途进行设定。 ➞有关设定的详细内容，请参考通信控制手册。	
Data length		
Parity		
Stop bit		
Transmission speed		
Header		
Terminator		
Control line		
H/W type		
Control mode		
Sum check		
Transmission control procedure		
Station number setting		
Time out judge time		

8 定位的设定。

定位设定可以在A8/A8C可编程控制器中使用。

1. 点击“Positioning”选项卡。
   “Positioning”选项卡，是在“Memory capacity”选项卡中选择“Positioning”后显示的。
2. 使用TBL(FNC 152)指令对要使用的定位参数进行设定。



画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
Bias speed [Hz]	对输出脉冲的各输出编号设定起始速度。 初始值: 0	最高速度的1/10以下
Max. speed [Hz]	对输出脉冲的各输出编号设定最高速度。初始值: 100000	*1
Creep speed [Hz]	对输出脉冲的各输出编号设定DSZR(FNC 150)指令的爬行速度。 初始值: 1000	10～32767*2
Zero return speed [Hz]	对输出脉冲的各输出编号设定DSZR(FNC 150)指令的原点回归速度。 初始值: 50000	*1
Acceleration time [ms]	对输出脉冲的各输出编号设定的加速时间。初始值: 100	50～5000
Decelertion time [ms]	对输出脉冲的各输出编号设定的减速时间。初始值: 100	50～5000
Interruption input of DVIT instruction*4	输出脉冲的各输出编号设定DVIT(FNC 151)指令动作用的中断输入*3。不使用DVIT指令的情况下，请在脉冲输出对象软元件中指定用户中断指令软元件(M)。 初始设定: 脉冲输出对象Y000: X000　设定范围: X000～X007、M8460 脉冲输出对象Y001: X001　X000～X007、M8461 脉冲输出对象Y002: X002　X000～X007、M8462 脉冲输出对象Y003*6: X003　X000～X007、M8463	左侧
Y0	脉冲输出对象为Y000的设定项目。	—
Y1	脉冲输出对象为Y001的设定项目。	—
Y2	脉冲输出对象为Y002的设定项目。	—
Y3*5	脉冲输出对象为Y003的设定项目。	—
Individual setting	显示TBL(FNC 152)指令的表设定用的“详细设定”对话框。 →有关设定操作的内容请参考下一个项目	—

*1. A8/A8C可编程控制器的情况下，设定范围为10～100000Hz。
A8的可编程控制器中脉冲输出对象为A8的情况下，设定范围为10～200000Hz。

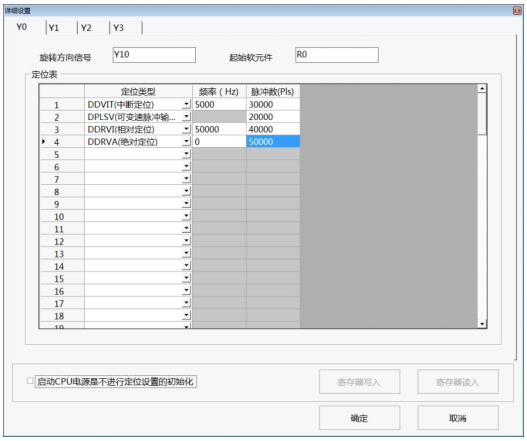
*2. 爬行速度必须要满足[偏离速度≤爬行速度≤最高速度]的关系。

*3. 设定的中断输入，不可以与高速计数器、输入中断、脉冲捕捉输入、SPD(FNC 67)指令的输入、其他的DVIT(FNC 151)指令的中断输入等并用。

*4. 仅可设定A8/A8C可编程控制器。

*5. 如果A8可编程控制器连接2台A8-2HSY-ADP，则可以设定。

3. 点击[Individual setting]键后，会显示“Positioning instruction settings”对话框。
设定各脉冲输出对象的定位表。



画面是A8/A8C可编程控制器的例子。

设定项目	设定内容	设定范围
Y0	设定脉冲输出对象为Y000的定位表。	—
Y1	设定脉冲输出对象为Y001的定位表。	—
Y2	设定脉冲输出对象为Y002的定位表。	—
Y3*1	设定脉冲输出对象为Y003的定位表。	—
Rotation direction signal	设定旋转方向输出的输出信号。 初始设定: 脉冲输出对象Y000:Y010 脉冲输出对象Y001:Y011 脉冲输出对象Y002:Y012 脉冲输出对象Y003:Y013	A8/A8C Y000～Y357 M0～M7679 S0～S4095
Head Address	保存脉冲数、频率的设定数据的软元件起始编号 和已经设定的软元件在开头使用的轴数无关，占用1600点(A8/A8C)。 初始设定:R0	A8/A8C D0～D6400 R0～R31168
No.	表编号。 可以在1～100范围内设定。	—
Positioning Instruction	定位的种类可以从下面几种中选择。 DDVIT(中断定位)*2、DPLSV(可变速脉冲输出)、 DDRVI(相对定位)、DDRVA(绝对定位)	—
Pulse(Pls)	在不同的定位种类已设定的动作(指令)中设定要输出的脉冲数。	定位控制 请参考手册
Frequency(Hz)	在不同的定位种类已设定的动作(指令)中设定要输出的速度(脉冲频率)。	定位控制 请参考手册
Up	选中的行移动到上一行。(和上一行替换。)	—
Down	选中的行移动到下一行。(和下一行替换。)	—
Insert	在选择的位置中插入行。	—
Delete	删除选中的行。	—
Delete all lines	将选中的脉冲输出对象的定位表的设定内容全部删除。	—

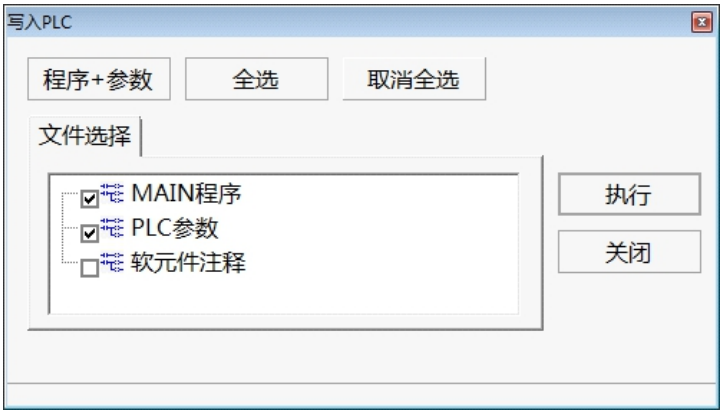
设定项目		设定内容	设定范围
Positioning table settings will not be initialized when the PLC is powered on		如选择该项，可编程控制器上电的时候定位设定的内容不会被传送。 从显示器等中更改定位设定的内容，并且在两次上电后依旧使用该内容的情况下，要做选择。而且，此时请将“起始软元件”设定为停电保持用的软元件。	—
	Write	将已设定的定位表的内容写入到可编程控制器的“起始软元件”起的1600点(A8/A8C)。	—
	Read	被设定的定位表的内容可以从可编程控制器的“起始软元件”起的1600点(A8/A8C)所有的脉冲输出对象中读出。 但是，不读出未设定定位种类的编号。	—

*1. 如果A8可编程控制器连接2台HCA8-2HSY-ADP，则可以设定。

*2. 仅可设定A8/A8C可编程控制器。

9 将参数(+顺控程序)传送到可编程控制器中。

1. 选择工具栏菜单的[Online]→[Write to PLC]后，显示“Write to PLC”对话框。



画面是A8/A8C可编程控制器的例子。

*1. 根据HCP works，PLC机型变为A8C。

2. 选中参数后，点击[Execute]键。

选中的内容会传送到可编程控制器中。
完成传送的参数，在可编程控制器STOP→RUN后变为有效。
此外，在步7的“PLC系统设定(2)”中更改了通信设定时，需要将可编程控制器的电源断开后再上电(ON)。

注意要点

更改了存储器容量的设定时，务必将程序和参数一起写入可编程控制器。
仅将参数写入到可编程控制器中的话，在可编程控制器一侧有时候会出现程序错误(参数错误、回路错误、语法错误等)。

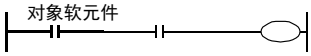
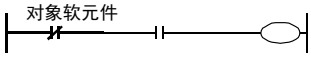
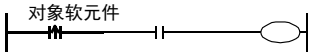
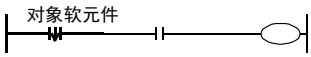
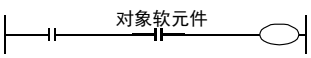
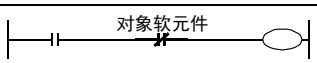
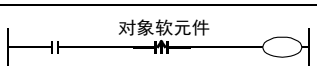
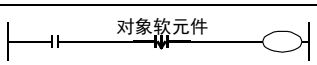
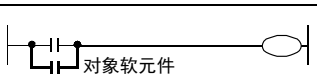
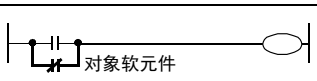
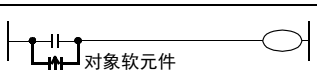
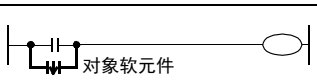
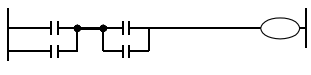
3. 指令一览

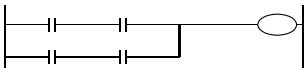
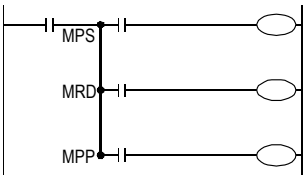
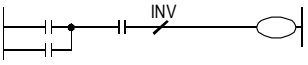
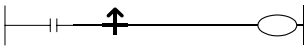
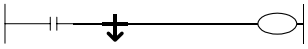
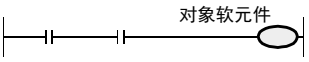
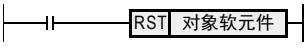
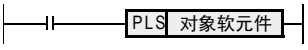
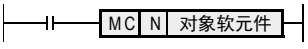
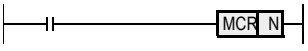
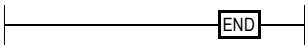
在本章中，介绍了编程中可以使用的指令的一览。

3.1 基本指令

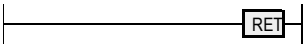
基本指令在下面的系列中对应，但是对象软元件如下表所示不同。

对应的可编程控制器	A8	A8C	A1	A2	A2C
MEP、MEF以外的基本指令	○	○	○	○	○
MEP、MEF指令	×	×	×	×	×
有/无对象软元件(D□.b)	○	○	×	×	×
有/无对象软元件(R)	○	○	×	×	×

记号	称呼	符号	功能	对象软元件	参考节
触点指令					
LD	取		a触点的逻辑运算开始	X,Y,M,S,D □.b,T,C	7.1 节
LDI	取反		b触电的逻辑运算开始	X,Y,M,S,D □.b,T,C	7.1 节
LDP	取脉冲上升沿		检测上升沿的运算开始	X,Y,M,S,D □.b,T,C	7.5 节
LDF	取脉冲下降沿		检测下降沿的运算开始	X,Y,M,S,D □.b,T,C	7.5 节
AND	与		串联a触点	X,Y,M,S,D □.b,T,C	7.3 节
ANI	与反转		串联b触点	X,Y,M,S,D □.b,T,C	7.3 节
ANDP	与脉冲上升沿		检测上升沿的串联连接	X,Y,M,S,D □.b,T,C	7.5 节
ANDF	与脉冲下降沿		检测下降沿的串联连接	X,Y,M,S,D □.b,T,C	7.5 节
OR	或		并联a触点	X,Y,M,S,D □.b,T,C	7.4 节
ORI	或反转		并联b触点	X,Y,M,S,D □.b,T,C	7.4 节
ORP	或脉冲上升沿		检测上升沿的并联连接	X,Y,M,S,D □.b,T,C	7.5 节
ORF	或脉冲下降沿		检测下降沿的并联连接	X,Y,M,S,D □.b,T,C	7.5 节
结合指令					
ANB	回路块与		回路块的串联连接	—	7.7节

记号	称呼	符号	功能	对象软元件	参考节
ORB	回路块或		回路块的并联连接	—	7.6节
MPS	存储器进栈		压入堆栈	—	7.8节
MRD	存储读栈		读取堆栈		7.8节
MPP	存储出栈		弹出堆栈		7.8节
INV	反转		运算结果的反转	—	7.10节
MEP	M・E・P		上升沿时导通	—	7.11节
MEF	M・E・F		下降沿时导通	—	7.11节
输出指令					
OUT	输出		线圈驱动	Y,M,S,D □.b,T,C	7.2 节
SET	置位		动作保持	Y,M,S,D □.b	7.13节
RST	复位		解除保持的动作，清除当前值及寄存器	Y,M,S,D □.b,T,C,D,R,V,Z	7.13节
PLS	脉冲		上升沿微分输出	Y,M	7.12节
PLF	下降沿脉冲		下降沿微分输出	Y,M	7.12节
主控指令					
MC	主控		连接到公共触点	Y,M	7.9节
MCR	主控复位		解除连接到公共触点	—	7.9节
其他指令					
NOP	空操作		无处理	—	7.14节
结束指令					
END	结束		程序结束以及 输入输出处理和返回0步	—	7.15节

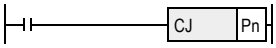
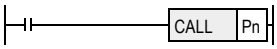
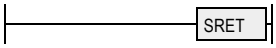
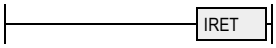
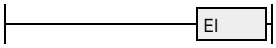
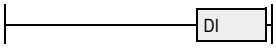
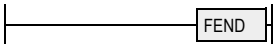
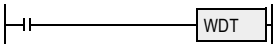
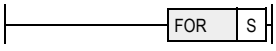
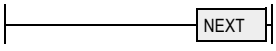
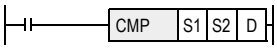
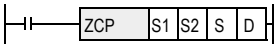
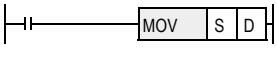
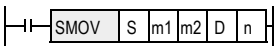
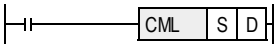
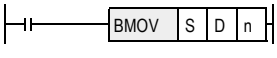
3.2 步进梯形图指令

记号	称呼	符号	功能	对象软元件	参考章
STL	步进梯形图		步进梯形图的开始	S	34章
RET	返回		步进梯形图的结束	—	34章

3.3 应用指令 - 按FNC.No顺序

应用指令主要在执行四则运算和旋转位移、便捷指令等，尤其是处理数值数据时使用。

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
程序流程									
00	CJ		条件跳转	○	○	○	○	○	8.1节
01	CALL		子程序调用	○	○	○	○	○	8.2节
02	SRET		子程序返回	○	○	○	○	○	8.3节
03	IRET		中断返回	○	○	○	○	○	8.4节
04	EI		允许中断	○	○	○	○	○	8.5节
05	DI		禁止中断	○	○	○	○	○	8.6节
06	FEND		主程序结束	○	○	○	○	○	8.7节
07	WDT		看门狗定时器	○	○	○	○	○	8.8节
08	FOR		循环范围的开始	○	○	○	○	○	8.9节
09	NEXT		循环范围的结束	○	○	○	○	○	8.10节
传送・比较									
10	CMP		比较	○	○	○	○	○	9.1节
11	ZCP		区间比较	○	○	○	○	○	9.2节
12	MOV		传送	○	○	○	○	○	9.3节
13	SMOV		位移动	○	○	—	—	—	9.4节
14	CML		反转传送	○	○	—	—	—	9.5节
15	BMOV		成批传送	○	○	○	○	○	9.6节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
传送・比较									
16	FMOV	FMOV S D n	多点传送	○	○	—	—	—	9.7节
17	XCH	XCH D1 D2	交换	○	○	—	—	—	9.8节
18	BCD	BCD S D	BCD转换	○	○	○	○	○	9.9节
19	BIN	BIN S D	BIN转换	○	○	○	○	○	9.10节
四则・逻辑运算									
20	ADD	ADD S1 S2 D	BIN加法运算	○	○	○	○	○	10.1节
21	SUB	SUB S1 S2 D	BIN减法运算	○	○	○	○	○	10.2节
22	MUL	MUL S1 S2 D	BIN乘法运算	○	○	○	○	○	10.3节
23	DIV	DIV S1 S2 D	BIN除法运算	○	○	○	○	○	10.4节
24	INC	INC D	BIN加一	○	○	○	○	○	10.5节
25	DEC	DEC D	BIN减一	○	○	○	○	○	10.6节
26	WAND	WAND S1 S2 D	逻辑与	○	○	○	○	○	10.7节
27	WOR	WOR S1 S2 D	逻辑或	○	○	○	○	○	10.8节
28	WXOR	WXOR S1 S2 D	逻辑异或	○	○	○	○	○	10.9节
29	NEG	NEG D	补码	○	○	—	—	—	10.10节
循环・移位									
30	ROR	ROR D n	循环右移	○	○	—	—	—	11.1节
31	ROL	ROL D n	循环左移	○	○	—	—	—	11.2节
32	RCR	RCR D n	带进位循环右移	○	○	—	—	—	11.3节
33	RCL	RCL D n	带进位循环左移	○	○	—	—	—	11.4节
34	SFTR	SFTR S D n1 n2	位右移	○	○	○	○	○	11.5节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
循环・移位									
35	SFTL	SFTL S D n1 n2	位左移	○	○	○	○	○	11.6节
36	WSFR	WSFR S D n1 n2	字右移	○	○	—	—	—	11.7节
37	WSFL	WSFL S D n1 n2	字左移	○	○	—	—	—	11.8节
38	SFWR	SFWR S D n	移位写入 [先入先出/先入后出控制用]	○	○	○	○	○	11.9节
39	SFRD	SFRD S D n	移位读出[先入先出控制用]	○	○	○	○	○	11.10节
数据处理									
40	ZRST	ZRST D1 D2	成批复位	○	○	○	○	○	12.1节
41	DECO	DECO S D n	译码	○	○	○	○	○	12.2节
42	ENCO	ENCO S D n	编码	○	○	○	○	○	12.3节
43	SUM	SUM S D	ON位数	○	○	—	—	—	12.4节
44	BON	BON S D n	ON位的判定	○	○	—	—	—	12.5节
45	MEAN	MEAN S D n	平均值	○	○	—	—	—	12.6节
46	ANS	ANS S m D	信号报警器置位	○	○	—	—	—	12.7节
47	ANR	ANR	信号报警器复位	○	○	—	—	—	12.8节
48	SQR	SQR S D	BIN开方运算	○	○	—	—	—	12.9节
49	FLT	FLT S D	BIN整数→2进制浮点数转换	○	○	—	—	—	12.10节
高速处理									
50	REF	REF D n	输入输出刷新	○	○	○	○	○	13.1节
51	REFF	REFF n	输入刷新(带滤波器设定)	○	○	—	—	—	13.2节
52	MTR	MTR S D1 D2 n	矩阵输入	○	○	○	○	○	13.3节
53	HSCS	HSCS S1 S2 D	比较置位(高速计数器用)	○	○	○	○	○	13.4节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
高速处理									
54	HSCR	HSCR S1 S2 D	比较复位(高速计数器用)	○	○	○	○	○	13.5节
55	HSZ	HSZ S1 S2 S D	区间比较(高速计数器用)	○	○	—	—	—	13.6节
56	SPD	SPD S1 S2 D	脉冲密度	○	○	○	○	○	13.7节
57	PLSY	PLSY S1 S2 D	脉冲输出	○	○	○	○	○	13.8节
58	PWM	PWM S1 S2 D	脉宽调制	○	○	○	○	○	13.9节
59	PLSR	PLSR S1 S2 S3 D	带加减速的脉冲输出	○	○	○	○	○	13.10节
方便指令									
60	IST	IST S D1 D2	初始化状态	○	○	○	○	○	14.1节
61	SER	SER S1 S2 D n	数据检索	○	○	—	—	—	14.2节
62	ABSD	ABSD S1 S2 D n	凸轮顺控绝对方式	○	○	○	○	○	14.3节
63	INCD	INCD S1 S2 D n	凸轮顺控相对方式	○	○	○	○	○	14.4节
64	TTMR	TTMR D n	示教定时器	○	○	—	—	—	14.5节
65	STMR	STMR S m D	特殊定时器	○	○	—	—	—	14.6节
66	ALT	ALT D	交替输出	○	○	○	○	○	14.7节
67	RAMP	RAMP S1 S2 D n	斜坡信号	○	○	○	○	○	14.8节
68	ROTC	ROTC S m1 m2 D	旋转工作台控制	○	○	—	—	—	14.9节
69	SORT	SORT S m1 m2 D n	数据排序	○	○	—	—	—	14.10节
外部设备I/O									
70	TKY	TKY S D1 D2	数字键输入	○	○	—	—	—	15.1节
71	HKY	HKY S D1 D2 D3	16进制数字键输入	○	○	—	—	—	15.2节
72	DSW	DSW S D1 D2 n	数字开关	○	○	○	○	○	15.3节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
外部设备I/O									
73	SEGD	SEGD S D	7段解码器	○	○	—	—	—	15.4节
74	SEGL	SEGL S D n	7SEG时分显示	○	○	○	○	○	15.5节
75	ARWS	ARWS S D1 D2 n	箭头开关	○	○	—	—	—	15.6节
76	ASC	ASC S D	ASCII数据输入	○	○	—	—	—	15.7节
77	PR	PR S D	ASCII码打印	○	○	—	—	—	15.8节
78	FROM	FROM m1 m2 D n	BFM的读出 —	○	○	—	○	○	15.9节
79	TO	TO m1 m2 S n	BFM的写入	○	○	—	○	○	15.10节
外部设备(选件设备)									
80	RS	RS S m D n	串行数据传送	○	○	○	○	○	16.1节
81	PRUN	PRUN S D	8进制位传送	○	○	○	○	○	16.2节
82	ASCI	ASCI S D n	HEX→ASCII的转换	○	○	○	○	○	16.3节
83	HEX	HEX S D n	ASCII→HEX的转换	○	○	○	○	○	16.4节
84	CCD	CCD S D n	校验码	○	○	○	○	○	16.5节
85	VRRD	VRRD S D	电位器读出	—	—	○	○	○	16.6节
86	VRSC	VRSC S D	电位器刻度	—	—	○	○	○	16.7节
87	RS2	RS2 S m D n n1	串行数据传送2	○	○	—	—	—	16.8节
88	PID	PID S1 S2 S3 D	PID运算	○	○	○	○	○	16.9节
89~99	—								—
数据传送2									
100, 101	—								—
102	ZPUSH	ZPUSH D	变址寄存器的成批保存	○	*5	—	—	—	17.1节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
数据传送2									
103	ZPOP	ZPOP D	变址寄存器的恢复	○	*5	—	—	—	17.2节
104~109	—								—
浮点数运算									
110	ECMP	ECMP S1 S2 D	2进制浮点数比较	○	○	—	—	—	18.1节
111	EZCP	EZCP S1 S2 S D	2进制浮点数区间比较	○	○	—	—	—	18.2节
112	EMOV	EMOV S D	2进制浮点数数据传送	○	○	—	—	—	18.3节
113~115	—								—
116	ESTR	ESTR S1 S2 D	2进制浮点数→字符串的转换	○	○	—	—	—	18.4节
117	EVAL	EVAL S D	字符串→2进制浮点数的转换	○	○	—	—	—	18.5节
118	EBCD	EBCD S D	2进制浮点数→10进制浮点数的转换	○	○	—	—	—	18.6节
119	EBIN	EBIN S D	10进制浮点数→2进制浮点数的转换	○	○	—	—	—	18.7节
120	EADD	EADD S1 S2 D	2进制浮点数加法运算	○	○	—	—	—	18.8节
121	ESUB	ESUB S1 S2 D	2进制浮点数减法运算	○	○	—	—	—	18.9节
122	EMUL	EMUL S1 S2 D	2进制浮点数乘法运算	○	○	—	—	—	18.10节
123	EDIV	EDIV S1 S2 D	2进制浮点数除法运算	○	○	—	—	—	18.11节
124	EXP	EXP S D	2进制浮点数指数运算	○	○	—	—	—	18.12节
125	LOGE	LOGE S D	2进制浮点数自然对数运算	○	○	—	—	—	18.13节
126	LOG10	LOG10 S D	2进制浮点数常用对数运算	○	○	—	—	—	18.14节
127	ESQR	ESQR S D	2进制浮点数开方运算	○	○	—	—	—	18.15节
128	ENEG	ENEG D	2进制浮点数符号翻转	○	○	—	—	—	18.16节
129	INT	INT S D	2进制浮点数→BIN整数的转换	○	○	—	—	—	18.17节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
浮点数运算									
130	SIN	SIN S D	2进制浮点数SIN运算	○	○	—	—	—	18.18节
131	COS	COS S D	2进制浮点数COS运算	○	○	—	—	—	18.19节
132	TAN	TAN S D	2进制浮点数TAN运算	○	○	—	—	—	18.20节
133	ASIN	ASIN S D	2进制浮点数SIN-1运算	○	○	—	—	—	18.21节
134	ACOS	ACOS S D	2进制浮点数COS-1运算	○	○	—	—	—	18.22节
135	ATAN	ATAN S D	2进制浮点数TAN-1运算	○	○	—	—	—	18.23节
136	RAD	RAD S D	2进制浮点数角度→弧度的转换	○	○	—	—	—	18.24节
137	DEG	DEG S D	2进制浮点数弧度→角度的转换	○	○	—	—	—	18.25节
138, 139	—								—
数据处理2									
140	WSUM	WSUM S D n	算出数据合计值	○	*5	—	—	—	19.1节
141	WTOB	WTOB S D n	字节单位的数据分离	○	*5	—	—	—	19.2节
142	BTOW	BTOW S D n	字节单位的数据结合	○	*5	—	—	—	19.3节
143	UNI	UNI S D n	16数据位的4位结合	○	*5	—	—	—	19.4节
144	DIS	DIS S D n	16数据位的4位分离	○	*5	—	—	—	19.5节
145, 146	—								—
147	SWAP	SWAP S	高低字节互换	○	○	—	—	—	19.6节
148	—								—
149	SORT2	SORT2 S m1 m2 D n	数据排序2	○	*5	—	—	—	19.7节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
定位									
150	DSZR	DSZR S1 S2 D1 D2	带DOG搜索的原点回归	○	*4	—	—	—	20.1节
151	DVIT	DVIT S1 S2 D1 D2	中断定位	○	*2 *4	—	—	—	20.2节
152	TBL	TBL D n	表格设定定位	○	*5	—	—	—	20.3节
153, 154	—								—
155	ABS	ABS S D1 D2	读出ABS当前值	○	○	○	○	○	20.4节
156	ZRN	ZRN S1 S2 S3 D	原点回归	○	*4	○	○	○	20.5节
157	PLSV	PLSV S D1 D2	可变速脉冲输出	○	○	○	○	○	20.6节
158	DRVI	DRVI S1 S2 D1 D2	相对定位	○	○	○	○	○	20.7节
159	DRVA	DRVA S1 S2 D1 D2	绝对定位	○	○	○	○	○	20.8节
时钟运算									
160	TCMP	TCMP S1 S2 S3 S D	时钟数据比较	○	○	○	○	○	21.1节
161	TZCP	TZCP S1 S2 S D	时钟数据区间比较	○	○	○	○	○	21.2节
162	TADD	TADD S1 S2 D	时钟数据加法运算	○	○	○	○	○	21.3节
163	TSUB	TSUB S1 S2 D	时钟数据减法运算	○	○	○	○	○	21.4节
164	HTOS	HTOS S D	时、分、秒数据的秒转换	○	○	—	—	—	21.5节
165	STOH	STOH S D	秒数据的[时、分、秒]转换	○	○	—	—	—	21.6节
166	TRD	TRD D	读出时钟数据	○	○	○	○	○	21.7节
167	TWR	TWR S	写入时钟数据	○	○	○	○	○	21.8节
168	—								—
169	HOUR	HOUR S D1 D2	计时表	○	○	○	○	○	21.9节

*1: A8C产品中可以更改功能

*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
外部设备									
170	GRY	GRY S D	格雷码的转换	○	○	—	—	—	22.1节
171	GBIN	GBIN S D	格雷码的逆转换	○	○	—	—	—	22.2节
172~175	—								—
176	RD3A	RD3A m1 m2 D	模拟量模块的读出	○	○	—	○	○	22.3节
177	WR3A	WR3A m1 m2 S	模拟量模块的写入	○	○	—	○	○	22.4节
178, 179	—								—
扩展功能									
180	EXTR	EXTR S SD1 SD2 SD3	扩展ROM功能(A2/A2C用)	—	—	—	—	—	—
其他指令									
181	—								—
182	COMRD	COMRD S D	读出软元件的注释数据	○	*5	—	—	—	24.1节
183	—								—
184	RND	RND D	产生随机数	○	○	—	—	—	24.2节
185	—								—
186	DUTY	DUTY n1 n2 D	产生定时脉冲	○	*5	—	—	—	24.3节
187	—								—
188	CRC	CRC S D n	CRC运算	○	○	—	—	—	24.4节
189	HCMOV	HCMOV S D n	高速计数器传送	○	*4	—	—	—	24.5节
数据块处理									
190, 191	—								—
192	BK+	BK+ S1 S2 D n	数据块的加法运算	○	*5	—	—	—	25.1节
193	BK-	BK- S1 S2 D n	数据块的减法运算	○	*5	—	—	—	25.2节

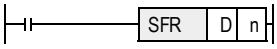
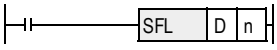
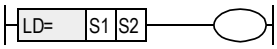
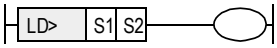
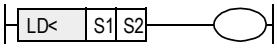
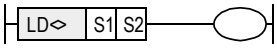
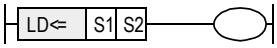
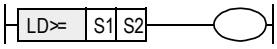
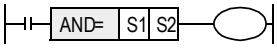
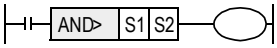
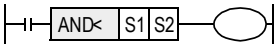
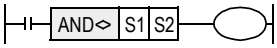
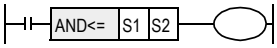
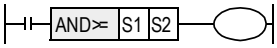
*1: A8C产品中可以更改功能

*2: A8C产品中对应

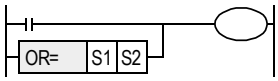
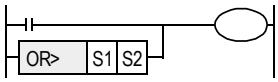
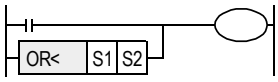
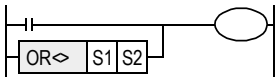
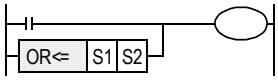
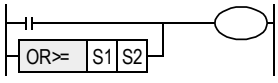
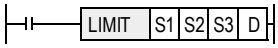
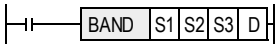
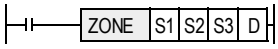
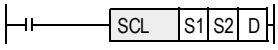
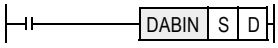
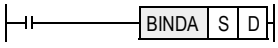
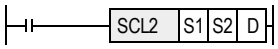
FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
数据块处理									
194	BKCMP=	BKCMP= S1 S2 D n	数据块比较 (S1) = (S2)	○	*5	—	—	—	25.3节
195	BKCMP>	BKCMP> S1 S2 D n	数据块比较 (S1) > (S2)	○	*5	—	—	—	25.3节
196	BKCMP<	BKCMP< S1 S2 D n	数据块比较 (S1) < (S2)	○	*5	—	—	—	25.3节
197	BKCMP<>	BKCMP<> S1 S2 D n	数据块比较 (S1) ≠ (S2)	○	*5	—	—	—	25.3节
198	BKCMP<=	BKCMP<= S1 S2 D n	数据块比较 (S1) ≦ (S2)	○	*5	—	—	—	25.3节
199	BKCMP>=	BKCMP>= S1 S2 D n	数据块比较 (S1) ≧ (S2)	○	*5	—	—	—	25.3节
字符串控制									
200	STR	STR S1 S2 D	BIN→字符串的转换	○	*5	—	—	—	26.1节
201	VAL	VAL S D1 D2	字符串→BIN的转换	○	*5	—	—	—	26.2节
202	$+	$+ S1 S2 D	字符串的结合	○	○	—	—	—	26.3节
203	LEN	LEN S D	检测出字符串的长度	○	○	—	—	—	26.4节
204	RIGHT	RIGHT S D n	从字符串的右侧开始取出	○	○	—	—	—	26.5节
205	LEFT	LEFT S D n	从字符串的左侧开始取出	○	○	—	—	—	26.6节
206	MIDR	MIDR S1 D S2	从字符串中的任意取出	○	○	—	—	—	26.7节
207	MIDW	MIDW S1 D S2	字符串中的任意替换	○	○	—	—	—	26.8节
208	INSTR	INSTR S1 S2 D n	字符串的检索	○	*5	—	—	—	26.9节
209	$MOV	$MOV S D	字符串的传送	○	○	—	—	—	26.10节
数据处理3									
210	FDEL	FDEL S D n	数据表的数据删除	○	*5	—	—	—	27.1节
211	FINS	FINS S D n	数据表的数据插入	○	*5	—	—	—	27.2节
212	POP	POP S D n	读取后入的数据 [先入后出控制用]	○	○	—	—	—	27.3节

*1: A8C产品中可以更改功能

*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
数据处理3									
213	SFR		16位数据n位右移(带进位)	○	○	—	—	—	27.4节
214	SFL		16位数据n位左移(带进位)	○	○	—	—	—	27.5节
215~219	—								
触点比较指令									
220~223	—								
224	LD=		触点比较LD (S1) = (S2)	○	○	○	○	○	28.1节
225	LD>		触点比较LD (S1) > (S2)	○	○	○	○	○	28.1节
226	LD<		触点比较LD (S1) < (S2)	○	○	○	○	○	28.1节
227	—								
228	LD<>		触点比较LD (S1) ≠ (S2)	○	○	○	○	○	28.1节
229	LD<=		触点比较LD (S1) ≦ (S2)	○	○	○	○	○	28.1节
230	LD>=		触点比较LD (S1) ≧ (S2)	○	○	○	○	○	28.1节
231	—								
232	AND=		触点比较AND (S1) = (S2)	○	○	○	○	○	28.2节
233	AND>		触点比较AND (S1) > (S2)	○	○	○	○	○	28.2节
234	AND<		触点比较AND (S1) < (S2)	○	○	○	○	○	28.2节
235	—								
236	AND<>		触点比较AND (S1) ≠ (S2)	○	○	○	○	○	28.2节
237	AND<=		触点比较AND (S1) ≦ (S2)	○	○	○	○	○	28.2节
238	AND>=		触点比较AND (S1) ≧ (S2)	○	○	○	○	○	28.2节
239	—								—

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
触点比较指令									
240	OR=		触点比较OR (S1) = (S2)	○	○	○	○	○	28.3节
241	OR>		触点比较OR (S1) > (S2)	○	○	○	○	○	28.3节
242	OR<		触点比较OR (S1) < (S2)	○	○	○	○	○	28.3节
243	—								
244	OR<>		触点比较OR (S1) ≠ (S2)	○	○	○	○	○	28.3节
245	OR<=		触点比较OR (S1) ≦ (S2)	○	○	○	○	○	28.3节
246	OR>=		触点比较OR (S1) ≧ (S2)	○	○	○	○	○	28.3节
247~249	—								
数据表处理									
250~255	—								
256	LIMIT		上下限限位控制	○	○	—	—	—	29.1节
257	BAND		死区控制	○	○	—	—	—	29.2节
258	ZONE		区域控制	○	○	—	—	—	29.3节
259	SCL		定坐标 (不同点坐标数据)	○	○	—	—	—	29.4节
260	DABIN		10进制ASCII→BIN的转换	○	*5	—	—	—	29.5节
261	BINDA		BIN→10进制ASCII的转换	○	*5	—	—	—	29.6节
262~268	—								
269	SCL2		定坐标2 (X/Y坐标数据)	○	*3	—	—	—	29.7节

*1: A8C产品中可以更改功能
*2: A8C产品中对应

FNC No.	指令记号	符号	功能	A8	A8C	对应的可编程控制器			参考节
						A1	A2	A2C	
外部设备通信(变频器通信)									
270	IVCK	IVCK S1 S2 D n	变换器的运转监视	○	○	—	—	—	30.1节
271	IVDR	IVDR S1 S2 S3 n	变频器的运行控制	○	○	—	—	—	30.2节
272	IVRD	IVRD S1 S2 D n	读取变频器的参数	○	○	—	—	—	30.3节
273	IVWR	IVWR S1 S2 S3 n	写入变频器的参数	○	○	—	—	—	30.4节
274	IVBWR	IVBWR S1 S2 S3 n	成批写入变频器的参数	○	○	—	—	—	30.5节
275~277	—								
数据传送3									
278	RBFM	RBFM m1 m2 D n1 n2	BFM分割读出	○	*5	—	—	—	31.1节
279	WBFM	WBFM m1 m2 S n1 n2	BFM分割写入	○	*5	—	—	—	31.2节
高速处理2									
280	HSCT	HSCT S1 m S2 D n	高速计数器表比较	○	○	—	—	—	32.1节
281~289	—								
扩展文件寄存器控制									
290	LOADR	LOADR S n	读出扩展文件寄存器	○	○	—	—	—	33.1节
291	SAVER	SAVER S m D	成批写入扩展文件寄存器	○	○	—	—	—	33.2节
292	INITR	INITR S m	扩展寄存器的初始化	○	○	—	—	—	33.3节
293	LOGR	LOGR S m D1 n D2	登录到扩展寄存器	○	○	—	—	—	33.4节
294	RWER	RWER S n	扩展文件寄存器的删除・写入	○	*3	—	—	—	33.5节
295	INITER	INITER S n	扩展文件寄存器的初始化	○	*3	—	—	—	33.6节
296~299	—								—

4. 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。

4.1 软元件编号一览

软元件编号如下进行分配。
此外，在基本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的编号请参考使用的可编程控制器主机的硬件篇手册后进行确认。

1) A8/A8C可编程控制器

软元件名	内容			参考
输入输出继电器				
输入继电器	X000～X367*1	248点	软元件的编号为8进制编号 输入输出合计为256点	4.2节
输出继电器	Y000～Y367*1	248点		
辅助继电器				
一般用[可变]	M0～M499	500点	通过参数可以更改保持/非保持的设定	4.3节
保持用[可变]	M500～M1023	524点		
保持用[固定]	M1024～M7679	6656点		
特殊用*2	M8000～M8511	512点		36章
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	4.4节
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	0.1～3,276.7秒	4.5节
100ms [子程序、中断子程序用]	T192～T199	8点	0.1～3,276.7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	0.1～3,276.7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	0～32,767的计数器 通过参数可以更改保持/非保持的设定	4.6节
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	-2,147,483,648～+2,147,483,647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		

*1. 根据可编程控制器型号而不同。详细内容请参考4.2节。

*2. 对应功能请参考36章。
有关停电保持区域的使用，请参考2.6节。

软元件名	内容			参考
高速计数器				
单相单计数的输入 双方向(32位)	C235～C245	C235~C255中最多可以使用8点[保持用] 通过参数可以更改保持/非保持的设定 -2,147,483,648～+2,147,483,647的计数器 硬件计数器*1 单相: 100kHz×6点，10kHz×2点 双相: 50kHz(1倍)、50kHz(4倍) 软件计数器 单相: 40kHz 双相: 40kHz(1倍)、10kHz(4倍)		4.8节
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			
数据寄存器(成对使用时32位)				
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	4.9节
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 <D1000～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*2	D8000～D8511	512点		36章
变址用(16位)	V0~V7,Z0~Z7	16点		4.11节
扩展寄存器・扩展文件寄存器				
扩展寄存器(16位)	R0～R32767	32768点	通过电池进行停电保持	4.10节
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	4.12节
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	-32,768～ +32,767		5章
	32位	-2,147,483,648～ +2,147,483,647		
16进制数(H)	16位	0～FFFF		
	32位	0～FFFFFFFF		
实数(E)　位	32	-1.0×2^{128}～ -1.0×2^{-126}, 0, 1.0×2^{-126}～ 1.0×2^{128} 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		

*1. A8可编程控制器中，如使用HCA8-4HSX-ADP，最大输入频率如下所示。
单相: 200kHz
双相: 100kHz(1倍)、100kHz(4倍)

*2. 对应功能请参考36章。
有关停电保持区域的使用，请参考2.6节。

软元件名	内容			参考
高速计数器				
单相单计数的输入 双方向(32位) (EEPROM保持)	C235～C245	-2,147,483,648～2,147,483,647的计数器 软件计数器 单相: 60kHz×4点，10kHz×2点 双相: 30kHz×2点，5kHz×1点		4.7节
单相双计数的输入 双方向(32位) (EEPROM保持)	C246～C250			
双相双计数的输入 双方向(32位) (EEPROM保持)	C251～C255			
数据寄存器(成对使用时32位)				
一般用(16位)	D0～D127	128点		4.9节
EEPROM保持用(16位)	D128～D1099	972点		
一般用(16位)*1	D1100～D7999	6900点		
EEPROM保持用(16位) <文件寄存器>	D1000～D7999	7000点	通过参数可以将EEPROM保持用数据寄存器7000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)*2	D8000～D8511	512点		36章
变址用(16位)	V0~V7,Z0~Z7	16点		4.11节
扩展寄存器・扩展文件寄存器				
扩展寄存器(16位)*1	R0～R23999	24000点		4.10节
扩展文件寄存器(16位) (EEPROM保持)	ER0～ER23999	24000点		
指针				
JUMP、CALL分支用	P0～P2047	2048点	CJ指令、CALL指令用	4.12节
输入中断	I0□□～I5□□	6点		
定时器中断	I6□□～I8□□	3点		
嵌套				
主控用	N0～N7	8点	MC指令用	
常数				
10进制数(K)	16位	-32,768～+32,767		5章
	32位	-2,147,483,648～+2,147,483,647		
16进制数(H)	16位	0～FFFF		
	32位	0～FFFFFFFF		
实数(E)*3	32位	-1.0×2^{128}～ -1.0×2^{-126}, 0, 1.0×2^{-126}～ 1.0×2^{128} 可以用小数点和指数形式表示		

*1. 使用选件的电池时，通过参数可以更改为保持用软元件(电池后备)。
但是，不能设定停电保持范围。

*2. 对应功能请参考36章。
有关停电保持区域的使用，请参考2.6节。

4.2 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”、“9”的数值。

4.2.1 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)

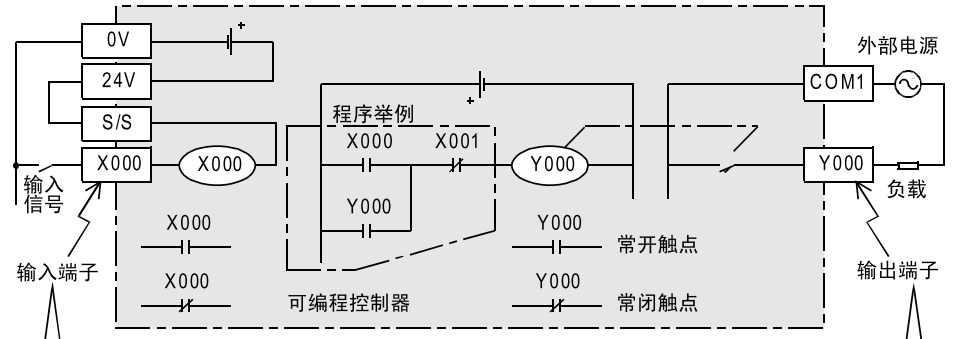
	型号	A8C-16M	A8C-32M	A8C-64M	A8C-96M	扩展时	
A8C(D、DSS) 可编程控制器	输入	X000～X007 8点	X000～X017 16点	X000～X037 32点	X000～X057 48点	X000～X367 248点	合计 256点
	输出	Y000～Y007 8点	Y000～Y017 16点	Y000～Y037 32点	Y000～Y057 48点	Y000～Y367 248点	

	型号	A8C-32MT-LT(-2)	扩展时	
A8C-32MT-LT(-2) 可编程控制器	输入	X000～X017 16点	X000～X357 240点	合计 256点
	输出	Y000～Y017 16点	Y000～Y357 240点	

	型号	A8-16M	A8-32M	A8-48M	A8-64M	A8-80M	A8-128M	扩展时	
A8 可编程控制器	输入	X000～X007 8点	X000～X017 16点	X000～X027 24点	X000～X037 32点	X000～X047 40点	X000～X077 64点	X000～X367 248点	合计 256点
	输出	Y000～Y007 8点	Y000～Y017 16点	Y000～Y027 24点	Y000～Y037 32点	Y000～Y047 40点	Y000～Y077 64点	Y000～Y367 248点	

4.2.2 功能和作用

端子名称及接线(输入为漏型输入时)以A8系列可编程控制器为例。

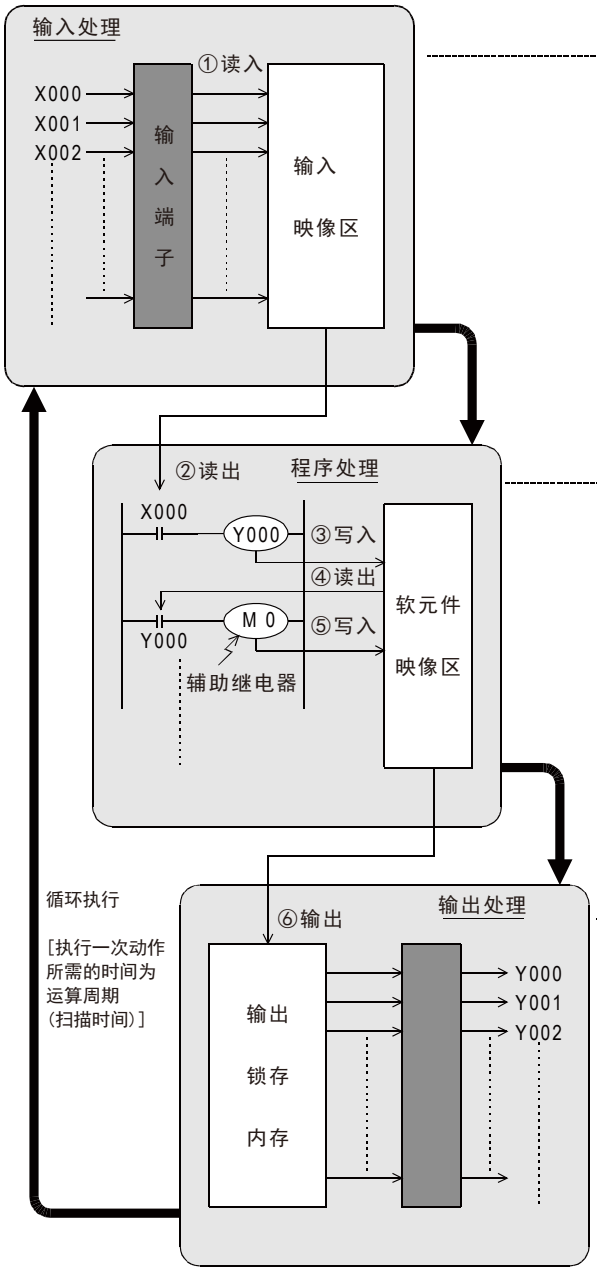


可编程控制器中，输入端子是从外部的开关接收信号的窗口。
可编程控制器中，可编程控制器的输入端子上连接的输入继电器(X)为光耦隔离的电子式继电器，因此具有无数的常开触点(a触点)和常闭触点(b触点)。在可编程控制器中可以随意地使用这些触点。
这个输入继电器不能通过程序来驱动。

可编程控制器中，输出端子是向外部的负载发出信号的窗口。
在可编程控制器中，输出继电器的外部输出用触点(继电器触点、晶闸管、晶体管等的输出元器件)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，可以在可编程控制器中随意地使用。
此外，外部输出触点(输出元器件)和内部触点的动作之间的差别如后页所示。

4.2.3 输入继电器的动作时序

可编程控制器是通过循环执行如下所示的处理顺序而实现顺序控制的。像这样的成批输入输出方式中，除了输入滤波器和输出元器件的驱动时间以外，还会因运算周期而出现响应延迟的现象。
(参照6.3节)



①②③…⑥表示处理顺序。

输入处理

可编程控制器在执行程序前，将可编程控制器的所有输入端子的ON/OFF状态读入输入映像区。
程序执行过程中即使输入发生变化，输入映像区的内容也不会变化，在执行下一个循环的输入处理时读取该变化。
此处，输入触点即使从ON→OFF、OFF→ON变化，到判断其ON/OFF之前，根据输入滤波器的情况有可能出现响应延迟（约10ms）。
（使用数字式滤波器的输入端子的输入滤波器，可以通过顺控程序对其数值进行修改。）

程序处理

可编程控制器根据程序内存中的指令内容，从输入映像区和其他软元件的映像区中读出各软元件的ON/OFF状态，然后从0步依次开始运算，并将每次得出的结果写入到映像区中。
因此，各软元件的映像区随着程序的执行逐步变其内容。
此外，输出继电器的内部触点根据输出映像区的内容而执行动作。

输出处理

所有指令执行结束后，输出Y映像区中的ON/OFF状态会传送至输出锁存内存，这个就作为可编程控制器的实际输出。
可编程控制器内的外部输出用触点，经过输出用元器件的响应延迟时间后动作。

4.3 辅助继电器[M]

可编程控制器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，是通过可编程控制器中的各种软元件的触点来驱动。
辅助继电器有无数的电子常开触点和常闭触点，可在可编程控制器中随意地使用。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。

4.3.1 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)

A8/A8C可编程控制器

一般用	停电保持用 (电池保持)	停电保持专用 (电池保持)	特殊用
M0～M499 500点*1	M500～M1023 524点*2	M1024～M7679 6656点*3	M8000～M8511 512点

*1. 非停电保持区域。根据设定的参数，可以更改为停电保持(保持)区域。

*2. 停电保持区域(保持)。根据设定的参数，可以更改为非停电保持。

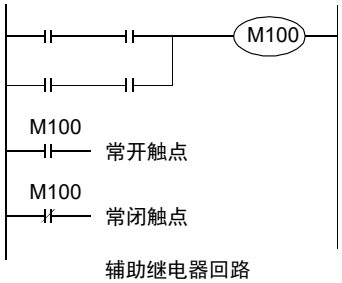
*3. 关于停电保持的特性可以通过参数进行变更。

使用简易PC间链接和并联链接的情况下，一部分的辅助继电器被占用为链接使用。

→ 参考通信控制手册

4.3.2 功能和动作实例

1. 一般用



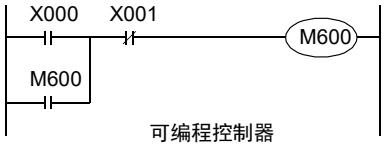
辅助继电器回路

当可编程控制器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，就使用停电保持(保持)用辅助继电器。

2. 停电保持用

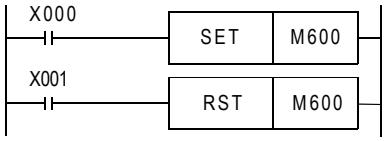
如在可编程控制器的运行过程中断开电源，输出继电器和一般的辅助继电器全部都变为OFF。
当再次上电时，除去输入条件为ON的以外，都为OFF。但是，根据控制对象不同，也可能出现停电之前的状态被记住，在再次运行时重新再现的情况。这样的情况下，使用停电保持用辅助继电器(又名保持继电器)。
A8/A8C可编程控制器的情况下，可以通过可编程控制器内置的电池执行软元件的停电保持。

→ 有关停电保持方法的详细内容，请参考2.6节



可编程控制器
停电保持(自保持回路)

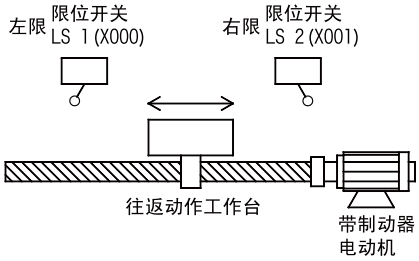
左图中是使用M600(停电保持用软元件)自保持动作的梯形图实例。
在该梯形图中，X000为ON，M600动作的话，X000即使开路，M600也能对动作进行自我保持。
由于M600是停电保持用的软元件，即使由于停电导致X000开路，当再次运行的时候，M600会继续之前的动作。但是，再次运行的时候，如X001的常闭触点开路，M600就不会动作。



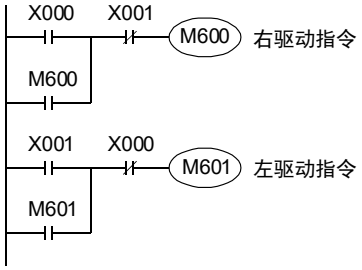
可编程控制器
停电保持(置位/复位回路)

使用了置位、复位指令时，为左图所示的梯形图。

1) 停电保持用的用途实例

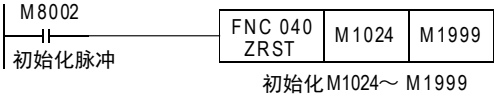


希望再次起动时，前进方向与停电前的前进方向相同。



X000=ON(左限)→M600=ON→向右驱动→停电→平台中途停止→再次起动(M600=ON)→X001=ON(右限)→M600=OFF、M601=ON→向左驱动

2) 将停电保持专用继电器作为一般用继电器使用的方法
将停电保持专用的辅助继电器作为一般用的辅助继电器使用时，请在程序的开头步附近设置如下图所示的复位梯形图。
A8/A8C可编程控制器的举例



初始化M1024～ M1999

4.4 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(SequentialFunctionChart)的编程方式中也可以使用状态。

→ 有关步进梯形图指令，以及SFC方式的编程，请参考34章

4.4.1 状态的编号

状态(S)的编号如下所示。(编号以10进制数分配)

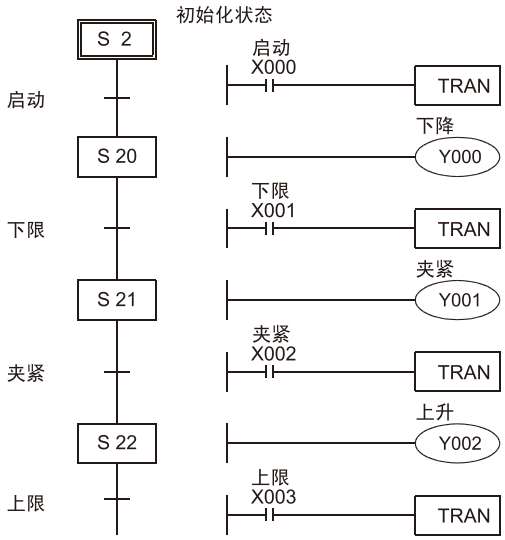
A8/A8C可编程控制器

初始状态用	一般用	停电保持用 (电池保持)	停电保持专用 (电池保持)	信号报警器用
S0～S9 10点*1	S0～S499 500点*1	S500～S899 400点*2	S1000～S4095 3096点*3	S900～S999 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(保持)区域。
*2. 停电保持区域(保持)。根据设定的参数，可以更改为非停电保持区域。
*3. 关于停电保持的特性可以通过参数进行变更。

4.4.2 功能和动作实例

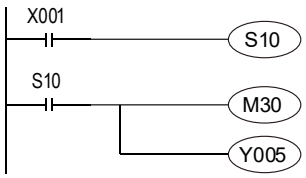
1. 一般用



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的话，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，状态也会被自动地复位(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(保持)用状态。

状态与辅助继电器相同，有无数个常开触点、常闭触点，可以在顺控程序中随意使用。
而且，不用于步进梯形图指令的时候,状态(S)也和辅助继电器(M)相同，可以在一般的顺控中使用。(右图)



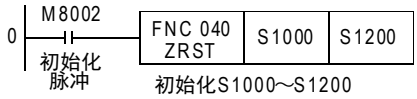
2. 停电保持用

- 停电保持用状态就是，即使在可编程控制器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。
  A8/A8C可编程控制器的情况下，可以通过可编程控制器内置的电池执行软元件的停电保持。

→ 有关停电保持的详细内容，请参考2.6节

- 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右图所示的复位梯形图。

A8/A8C可编程控制器的举例

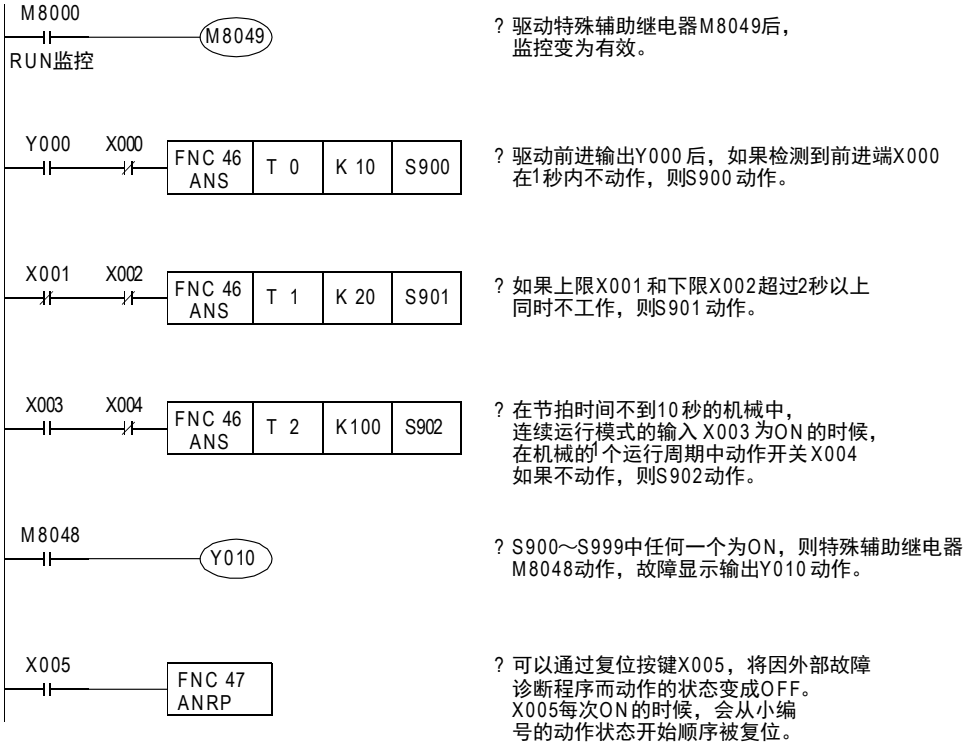


初始化S1000～S1200

3. 信号报警器用

信号报警器用状态，也可以作为诊断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，消除最小编号的故障后即可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(保持)用状态与一般状态一样，可以在顺控程序内使用。
但是，在HCP-works编程模式下，不能将S900～S999作为SFC图的工序进行编程。

4.5 定时器[T]

定时器就是，用加法计算可编程控制器中的1ms、10ms、100ms等的时钟脉冲，当加法计算的结果达到所指定的设定值时，输出触点就动作的软元件。
作为设定值，可使用程序内存中的常数(K)、以及通过数据寄存器(D)的内容间接指定。

4.5.1 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)

A8/A8C可编程控制器

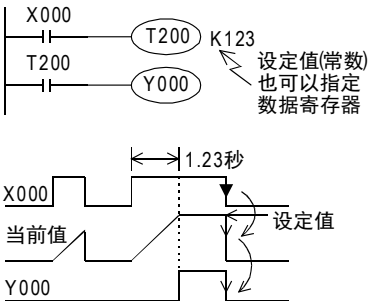
100ms型 0.1～3276.7秒	10ms型 0.01～327.67秒	1ms累计型*1 0.001～32.767秒	100ms累计型*1 0.1～3276.7秒	1ms型 0.001～32.767秒
T0～T199 200点 ---------- 子程序 程序用 T192～T199	T200～T245 46点	T246～T249 4点 执行中断 保持用*1	T250～T255 6点 保持用*1	T256～T511 256点

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。

*1. A8/A8C可编程控制器的累计型定时器是通过电池进行停电保持的。

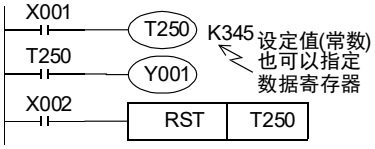
4.5.2 功能和动作实例

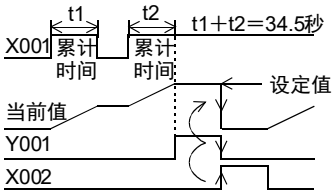
1. 一般用



当定时器线圈T200的驱动输入X000为ON，T200用的当前值计数器就对10ms的时钟脉冲进行加法运算，如果这个值等于设定值K123时，定时器的输出触点动作。
也就是说，输出触点是在驱动线圈后的1.23秒后动作。
驱动输入X000断开，或是停电时，定时器会被复位并且输出触点也复位。

2. 累计型





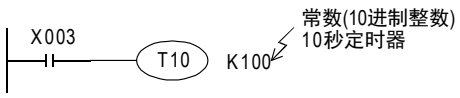
当定时器线圈T250的驱动输入X001为ON，T250用的当前值计数器就对100ms的时钟脉冲进行加法运算，如果这个值等于设定值K345时，定时器的输出触点动作。

在计数过程中，即使出现输入X001变OFF或停电的情况，当再次运行时也能继续计数。其累计动作时间为34.5秒。

复位输入X002为ON时，定时器会被复位并且输出触点也复位。

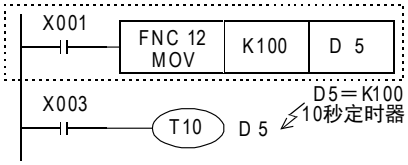
4.5.3 设定值的指定方法

1. 指定常数(K)



T10是以100ms(0.1s)为单位的定时器。
将常数指定为100，则0.1s×100=10s的定时器工作。

2. 间接指定



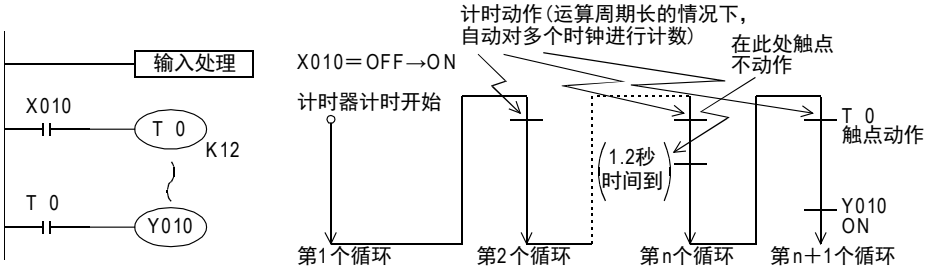
间接指定的数据寄存器的内容，或是预先在程序中写入，或是通过数字式开关等输入。
指定了停电保持(电池保持)用寄存器的时候，如果电池电压下降，设定值有可能会变得不稳定，需要注意。

4.5.4 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累计型定时器，当它达到设定值以后，在最初执行的线圈指令处输出触点会动作。请务必注意。

4.5.5 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输出触点动作。



从上面的动作图可以得知，从驱动线圈开始到触点动作为止的定时器触点的动作精度大致可以用下面的公式表示。

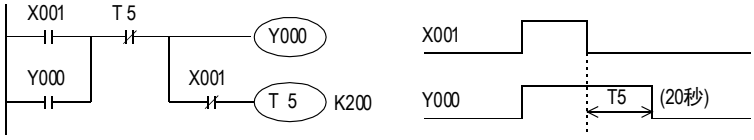
$$T^{+T_0}_{-\alpha}$$

α ：根据1ms, 10ms, 100ms定时器分别为0.001、0.01、0.1(秒)
T ：定时器设定时间(秒)
T_0 ：运算周期(秒)

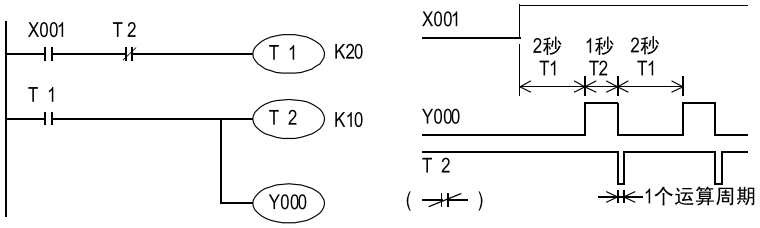
如果编程的时候，触点在定时器线圈前面的话，最大误差情况为$+2T_0$。
此外，定时器的设定值为0的时候，在下一个循环中，线圈指令执行时，输出触点动作。此外，中断执行型的1ms定时器是在线圈指令执行后，以中断方式对1ms的时钟脉冲进行计数。

4.5.6 程序举例【断开延时定时器，闪烁】

断开延时定时器



闪烁



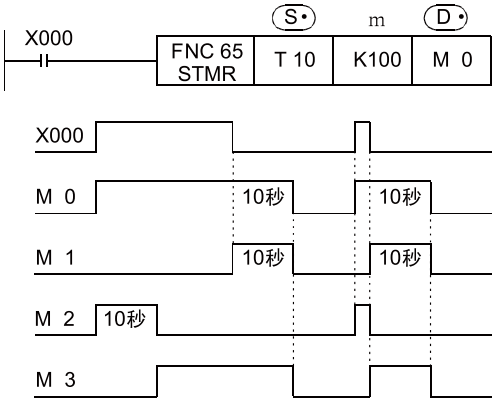
除此以外，使用ALT(FNC 66)指令也可以实现闪烁的动作。

使用了STMR(FNC 65)指令的多重定时器<A8/A8C可编程控制器>

使用这个指令，可以简易地制作断开延时定时器、单脉冲定时器，闪烁定时器。

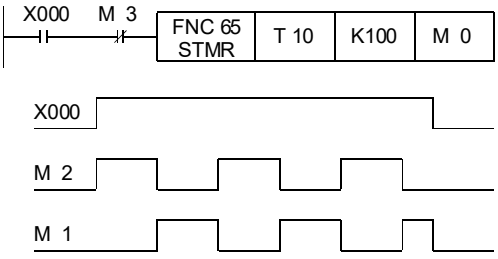
→ 详细内容，参考14.6节

断开延时定时器・单脉冲定时器



- m指定的值是 (S•) 指定定时器的设定值。在这个例子中，为10秒。
- M0为断开延时定时器。
- M1为输入从ON→OFF后的单脉冲定时器。
- M2,M3为闪烁用，按照闪烁用的程序举例(左下图)进行连接。

闪烁



- 按照左图所示连接M3的时候，M2,M1为闪烁输出。
- 如果X000为OFF，则在设定时间后M0，M1，M3都变为OFF，T10也会被复位。
- 在此处使用的定时器不能在其他的一般梯形图中重复使用。

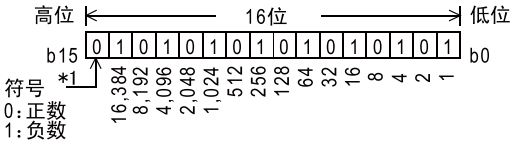
此外，使用TTMR(FNC 64)示教定时器指令时，还可以根据开关的输入时间来设定定时器的时间。

4.5.7 作为数据软元件的处理

定时器按使用方法分为: 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，当对应用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位或是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

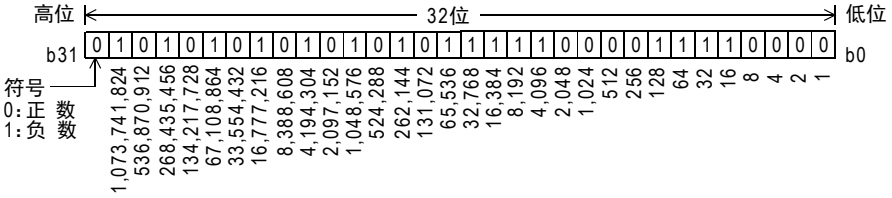
1) 16位



处理数值的范围
16位：0～32,767
32位：-2,147,483,648～+2,147,483,647

＊1 .只有在作为数据寄存器的替代使用的情况下有效。

2) 32位



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。

4.6 计数器[C]

4.6.1 计数器的编号

计数器的编号(C)如下表所示。(编号以10进制数分配)

→ 有关高速计数器的内容，请参考4.7节

A8/A8C可编程控制器

16位增计数器 0～32767计数		32位增/减计数器 -2,147,483,648～+2,147,483,647	
一般用	停电保持用 (电池保持)	一般用	停电保持用 (电池保持)
C0～C99 100点*1	C100～C199 100点*2	C200～C219 20点*1	C220～C234 15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(保持)区域。

*2. 停电保持区域(保持)。根据设定的参数，可以更改为非停电保持区域。

不作为计数器使用的计数器编号，可以作为保存数值用的数据寄存器使用。

4.6.2 计数器的特征

16位计数器和32位计数器的特点如下所示。可以按照计数方向的切换，以及计数范围等的使用条件不同而分开使用。

项目	16位计数器	32位计数器
计数方向	增计数	增/减计数可切换使用(参考4.6.3项)
设定值	1～32767	-2,147,483,648～2,147,483,647
设定值的指定	常数K或是数据寄存器	同左，但是数据寄存器需要成对(2个)
当前值的变化	计数值到后不变化	计数值到后，仍然变化(环形计数)
输出触点	计数值到后保持动作	增计数时保持，减计数时复位
复位动作	执行RST指令时计数器的当前值为0，输出触点也复位	
当前值寄存器	16位	32位

4.6.3 相关软元件(增/减的指定)【32位计数器】

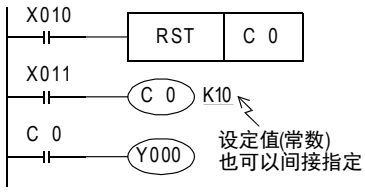
增减计数切换用的辅助继电器如果ON时为减计数器，OFF时为增计数器。

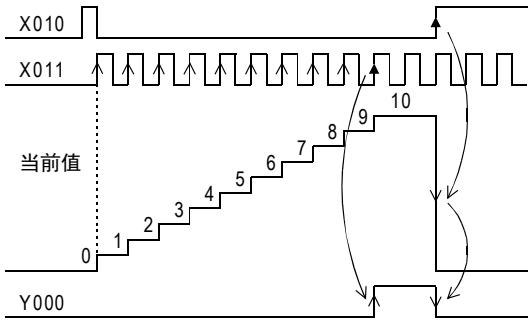
计数器号	切换方向	计数器号	切换方向	计数器号	切换方向	计数器号	切换方向
C200	M8200	C209	M8209	C218	M8218	C227	M8227
C201	M8201	C210	M8210	C219	M8219	C228	M8228
C202	M8202	C211	M8211	C220	M8220	C229	M8229
C203	M8203	C212	M8212	C221	M8221	C230	M8230
C204	M8204	C213	M8213	C222	M8222	C231	M8231
C205	M8205	C214	M8214	C223	M8223	C232	M8232
C206	M8206	C215	M8215	C224	M8224	C233	M8233
C207	M8207	C216	M8216	C225	M8225	C234	M8234
C208	M8208	C217	M8217	C226	M8226		

4.6.4 功能和动作实例

1. 16位计数器一般用/停电保持用

- 16位的2进制增计数器的设定值在K1～K32767(10进制常数)范围内有效。
  K0的动作和K1相同，在初次计数时输出触点动作。
- 一般用计数器的情况下，如果可编程控制器的电源断开，则计数值会被清除，但是停电保持用计数器的情况下，会记住停电之前的计数值，所以能够继续在上一次的值上进行累计计数。
- 通过计数输入X011，每驱动一次C0线圈，计数器的当前值就会增加，在第10次执行线圈指令的时候输出触点动作。
  此后，即使计数输入X011动作，但是计数器的当前值不会变化。
  如果输入复位X010为ON，在执行RST指令的时候，计数器的当前值变0，输出触点也复位。





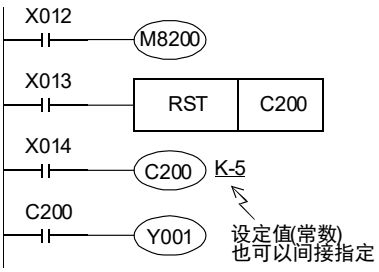
- 作为计数器的当前值，除了可以通过上述的常数K进行设定以外，还可以通过数据寄存器编号进行指定。例如，指定D10后，D10的内容如果是123时，就等同于K123的设定。
- 使用MOV 指令等对当前值寄存器写入超过设定值的数据时，当有下一个计数输入的时候，OUT线圈为ON，当前值寄存器为设定值。
- 停电保持用的情况下，计数器的当前值和输出触点的动作、复位状态都会被停电保持。
  A8/A8C可编程控制器的情况下，可以通过可编程控制器内置的电池执行计数器的停电保持。

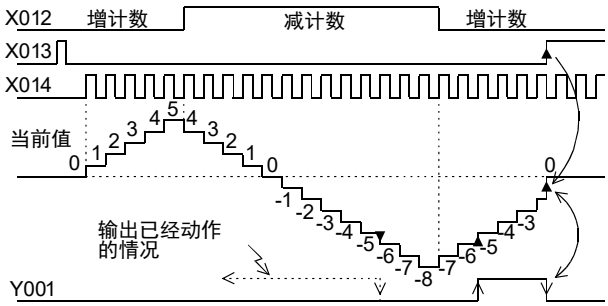
→ 有关停电保持方法的详细内容，请参考2.6节

2. 32位增/减计数器 一般用/停电保持用

32位的2进制增/减计数器的设定值在 -2,147,483,648～+2,147,483,647(10进制常数)的范围内有效。可以使用辅助继电器M8200～M8234指定增计数/减计数的方向。

- 对于C△△△，驱动M8△△△后为减计数器，不驱动的时候为增计数器。(参考上一页)
- 根据常数K或是数据寄存器D的内容，设定值可以使用正负的值。
  使用数据寄存器的情况下，将编号连续的软元件视为一对，将32位数据作为设定值。
  例如，指定D0的情况下，D1、D0这2个就是32位的设定值。
- 使用计数输入X014驱动C200线圈的时候，可增计数也可减计数。
  在计数器的当前值由“-6”增加到“-5”的时候，输出触点被置位，在由“-5”减少到“-6”的时候被复位。





- 当前值的增减与输出触点的动作无关，如果从 2,147,483,647 开始增计数的话则变成 -2,147,483,648。同样地，如果从-2,147,483,648开始减计数，就变成2,147,483,647。(像这样的动作称为环形计数。)
- 如果复位输入X013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。
- 停电保持用的情况下，计数器的当前值和输出触点的动作、复位状态都会被停电保持。
  A8/A8C可编程控制器的情况下，可以通过可编程控制器内置的电池执行计数器的停电保持。

→ 有关停电保持方法的详细内容，请参考2.6节

- 32位的计数器也可以作为32位的数据寄存器使用。而且，32位的计数器不能成为16位应用指令中的对象软元件。
- 使用 DMOV 指令等对当前值寄存器写入超过设定值的数据的情况下，当有下一个计数输入时，计数器就会继续计数，触点也不会变化。

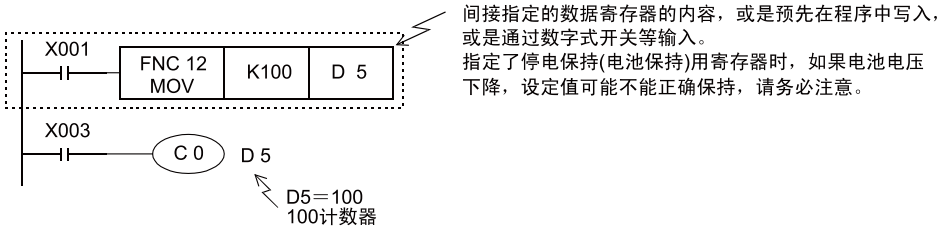
4.6.5 设定值的指定方法

1. 16位计数器

1) 指定常数(K)

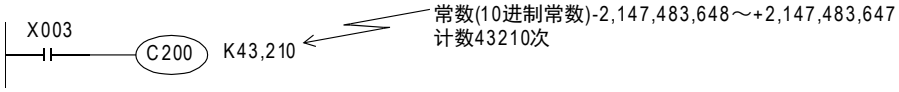


2) 间接指定(D)

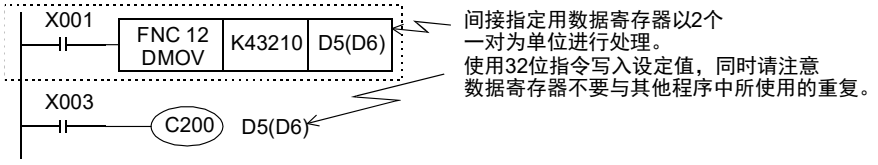


2. 32位计数器

1) 指定常数(K)



2) 间接指定(D)



4.6.6 计数器的响应速度

计数器就是在对可编程控制器的内部信号X、Y、M、S、C等触点的动作执行循环运算的同时进行计数。例如，X011作为计数输入时，它的ON和OFF的持续时间必须要比可编程控制器的扫描时间还要长。(通常是几10Hz以下)
对于这个问题，后面将要提及的高速计数器，就是用中断处理对特定的输入计数，与扫描时间无关，执行几kHz的计数

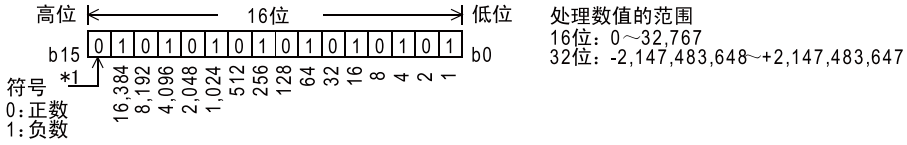
→ 有关高速计数器，参考4.7节

4.6.7 作为数据软元件的使用

计数器按使用方法分为: 根据设定值动作的输出触点被使用的情况，以及在控制中将计数器数值(当前值)作为数值数据使用的情况。
计数器的当前值寄存器的结构如下所示，如果在应用指令的操作数中指定计数器编号时，与数据寄存器相同，可以作为16位或是32位的数据保存软元件处理。
但是，32位的计数器，是作为1点32位数据处理。

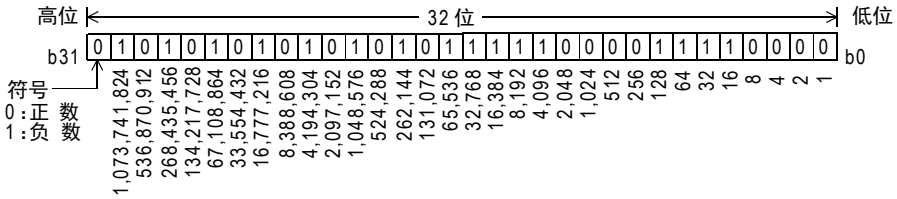
1. 计数器的当前值寄存器的结构

1) 16位



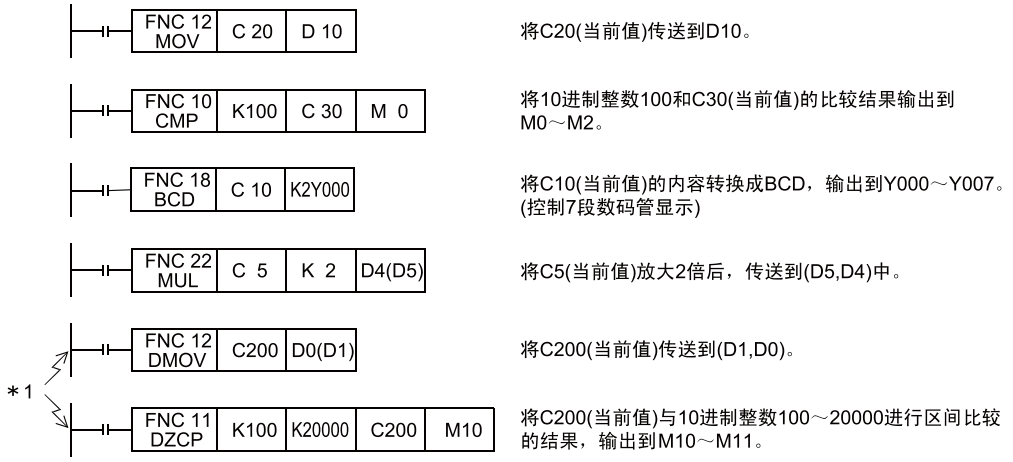
＊1 .只有在作为数据寄存器的替代使用的情况下有效。

2) 32位



2. 在应用指令中使用的实例

为了更好的将计数器作为数值软元件使用，请阅读后述的应用指令说明的内容。



＊1.对32位计数器，请务必使用32位的运算指令

4.7 高速计数器[C](A8/A8C可编程控制器)

4.7.1 高速计数器的种类和软元件的编号

1. 高速计数器的种类

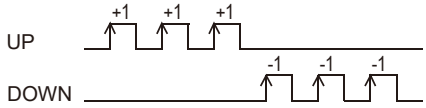
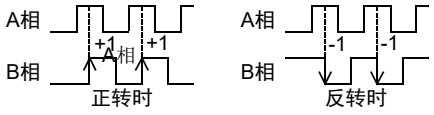
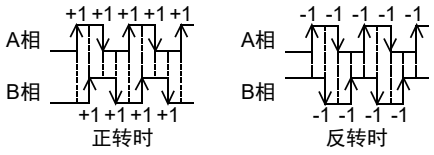
基本单元中，内置了32位增减计数器的高速计数器(单相单计数、单相双计数以及双相双计数)。在这个高速计数器中，根据计数的方法不同可以分为硬件计数器和软件计数器两种。
而且，在高速计数器中，提供了可以选择外部复位输入端子和外部启动输入端子(开始计数)的功能。

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。
  → 有关作为软件计数器使用的条件，参考4.7.9项
- 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。
  → 有关综合频率对响应频率的限制内容，参考4.7.10项

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ON/OFF来指定增计数或是减计数。 ON: 减计数 OFF: 增计数
单相双计数的输入			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，根据 /B相的输入状态变化，自动的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接的设备的注意事项

高速计数器的输入，使用通用输入X000～X007。根据所连接的端子,可以连接下表中的输出形式的编码器*1。
此外，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主机的硬件篇手册

可以直接连接到基本单元的输入端子的输出方式	开路集电极型的晶体管输出方式中对应24V
可以直接连接到HCA8-4HSX-ADP的输入端子的输出方式	差动输出方式(输出电压为DC5V以下)

*1. 按照电气上的相性，有时候即使使用上表的输出方式中的旋转编码器也可能不能正常运行，所以还望事先能够确认规格。

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
单相单计数的输入	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	无
		C244*3 C245*3	–		有*5	有
单相双计数的输入	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
双相双计数的输入	硬件计数器*1	C251*2	1倍*4	32位 增减计数器	无	无
			4倍*4			
		C253*2	1倍*4		有*5	
			4倍*4			
	软件计数器	C252	1倍*4		有*5	无
			4倍*4			
		C253(OP)*6	1倍*4		无	
			4倍*4			
		C254 C255	1倍*4		有*5	有
			4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。
→ 有关作为软件计数器使用的条件，参考4.7.9项
→ 有关综合频率，参考4.7.10项

*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，请参考所使用的可编程控制器主机的硬件篇手册

*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。
→ 有关切换计数器功能的方法，参考4.7.7项

*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。
→ 有关作为4倍的计数器使用的方法，参考4.7.8项

*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) 一起使用时，可以更改为在OFF时复位。
→ 有关更改外部复位输入逻辑的方法，参考4.7.6项

*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

A8/A8C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	切换后的软元件编号
C244	C244(OP)
C245	C245(OP)

普通的软元件编号	切换后的软元件编号
C248	C248(OP)
C253	C253(OP)

4.7.2 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～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(X000～X005: 5μs，X006、X007: 50μs)。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。
使用A8-4HSX-ADP时，下表中粗框内表示的是分配给A8可编程控制器连接的第1台A8-4HSX-ADP的输入编号。除此以外的地方，是分配给第2台的输入编号。

→ 有关A8-4HSX-ADP的输入规格，请参考A8硬件篇的手册
→ 有关基本单元的输入规格，请参考要使用的可编程控制器主机的硬件篇手册

	计数器编号	区分	输入端子的分配							
			X000	X001	X002	X003	X004	X005	X006	X007
单相单计数的输入	C235*1	H/W*2	U/D							
	C236*1	H/W*2		U/D						
	C237*1	H/W*2			U/D					
	C238*1	H/W*2				U/D				
	C239*1	H/W*2					U/D			
	C240*1	H/W*2						U/D		
	C241	S/W	U/D	R						
	C242	S/W			U/D	R				
	C243	S/W					U/D	R		
	C244	S/W	U/D	R					S	
	C244(OP)*3	H/W*2							U/D	
	C245	S/W			U/D	R				S
	C245(OP)*3	H/W*2								U/D
单相双计数的输入	C246*1	H/W*2	U	D						
	C247	S/W	U	D	R					
	C248	S/W				U	D	R		
	C248(OP)*1*3	H/W*2				U	D			
	C249	S/W	U	D	R				S	
	C250	S/W				U	D	R		S
双相双计数输入*4	C251*1	H/W*2	A	B						
	C252	S/W	A	B	R					
	C253*1	H/W*2				A	B	R		
	C253(OP)*3	S/W				A	B			
	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: 减计数输入
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. 在这个高速计数器中，接线上有需要注意的事项。
→ 有关接线，请参考所使用的可编程控制器主机的硬件篇手册

*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(H/W)变为软件(S/W)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。
→ 有关作为软件计数器处理的方法，参考4.7.9项

*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。
→ 有关作为硬件计数器处理的方法，参考4.7.7项

*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。
→ 有关可以作为4倍的计数器使用的方法，参考4.7.8项

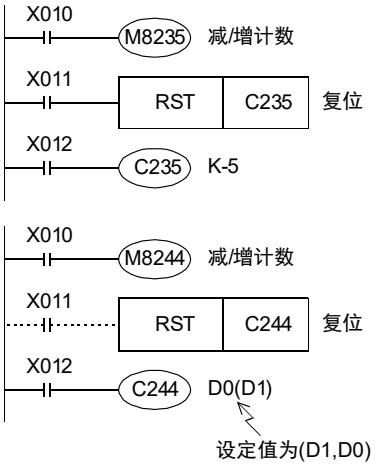
有关禁止重复使用输入端子

- 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I000、I101]、[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。
- A8-4HSX-ADP一侧的输入端子和A8可编程控制器的基本单元一侧的输入端子，分配相同的编号，但请务必只使用其中一侧的输入端子。如果两者的输入端子都使用时，A8-4HSX-ADP一侧和基本单元输入一侧的输入就呈OR的关系运行，这样可能不能实现想要的运行。

一侧和基本单元输入一侧的输入就呈

4.7.3 高速计数器的使用

1. 单相单计数的输入

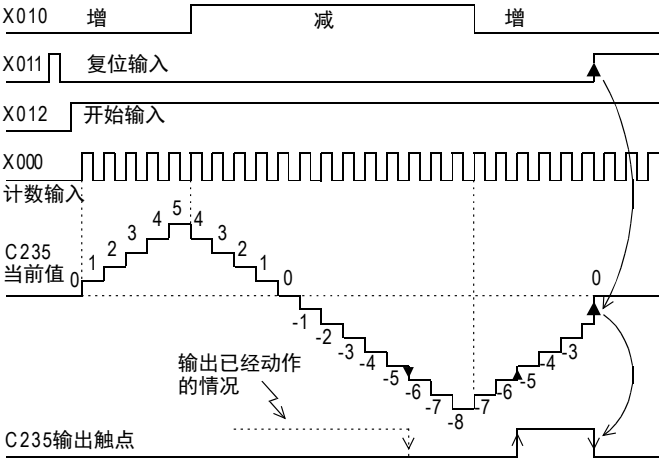


- C235在X012为ON时，对输入X000的OFF→ON进行计数。
- X011为ON时，执行RST指令，此时C235将被复位。
- 通过M8235～M8245的ON/OFF，使计数器C235～C245在减/增计数之间变化。

- C244在X012为ON，且输入X006变ON以后，立即开始计数。计数输入为X000，在这个例子中设定值就是间接指定的数据寄存器的内容(D1,D0)。
- 如图所示，可以通过 X011 在程序上进行复位，但是合上X001也会立即被复位。所以不需要这样的程序。
- 通过M8235～M8245的ON/OFF，使计数器C235～C245在减/增计数之间变化。

动作例子

上述的计数器C235的动作如下所示。

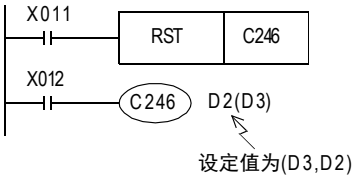


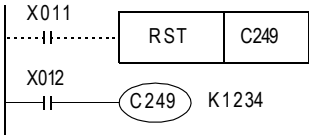
根据计数输入X000，C235通过中断进行增或是减的计数。

- 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。
- 当前值的增减与输出触点的动作无关，如果从2,147,483,647开始增计数的话则变成-2,147,483,648。同样地，如果从－2,147,483,648开始减计数，就变成2,147,483,647。(像这样的动作称为环形计数。)
- 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。
- 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。

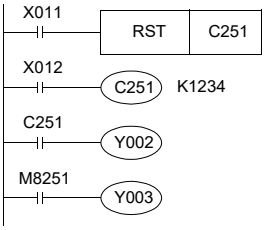




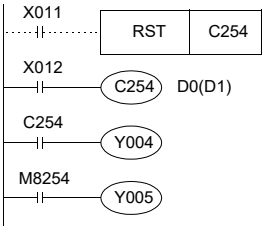
- C246在X012为ON的时候，如果输入X000由OFF→ON的话就为增计数，如果输入X001由OFF→ON时就为减计数。
- C246～C250的减/增计数动作可以通过 M8246～M8250的ON/OFF动作进行监控。
  ON: 减计数
  OFF: 增计数
- C249 在 X012 为 ON 时，如果输入 X006 为 ON 以后就立即开始计数。
  增计数输入为X000，减计数输入为X001。
- 如左图所示，可以通过X011在程序上进行复位，但X002合上时就会立即被复位。所以不需要这样的程序。
- C246～C250的减/增计数动作可以通过 M8246～M8250的ON/OFF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



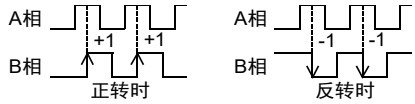
- X012为ON的时候，C251通过中断对输入X000(A相)、X001(B相)的动作进行计数。
  X011为ON，执行RST指令，此时C251将被复位。
- 当前值超出设定值的话Y002为ON，在设定值以下范围内变化时为OFF。
- Y003根据计数方向而ON(减)、OFF(增)。



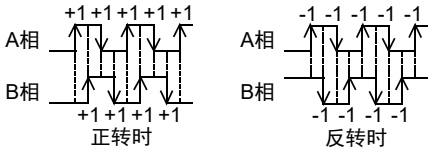
- X012为ON时，如果X006为ON后就立即开始C254的计数。该计数的输入为X000(A相)、X001(B相)。
- 除了使用X011在程序上进行复位以外，X002为ON时也可以立即将C254复位。
- 当前值超出设定值(D1,D0)的时候Y004动作，在设定值以下的范围内变化时为OFF。
- Y005根据计数方向而ON(减)、OFF(增)。

- 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倍动作的时候



  — 以4倍动作的时候



- 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

4.7.4 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2种方法。

1) 使用比较指令(CMP)、区间比较指令(ZCP)和比较触点指令
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) 使用高速计数器用的比较指令(HSCS/HSCR/HSZ/HSCT指令)
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	指令的使用次数的限制
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。

→ 有关软件计数器的最大响应频率和综合频率，参考4.7.10项

4.7.5 相关软元件

1. 单相单计数输入计数器的增/减计数的切换用

种类	计数器编号	指定用软元件	增计数	减计数
单相单计数的输入	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/减计数方向的监控用

种类	计数器编号	监控用软元件	OFF	ON
单相双计数的输入	C246	M8246	增计数	减计数
	C247	M8247		
	C248	M8248		
	C249	M8249		
	C250	M8250		
双相双计数的输入	C251	M8251		
	C252	M8252		
	C253	M8253		
	C254	M8254		
	C255	M8255		

3. 高速计数器的功能切换用

软元件编号	名称	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	功能切换软元件	外部复位输入的逻辑切换(详细参见4.7.6项)
M8390		C244用功能切换软元件(详细参见4.7.7项)
M8391		C245用功能切换软元件(详细参见4.7.7项)
M8392		C248、C253用功能切换软元件(详细参见4.7.7项)
M8198		C251、C252、C254用的1倍/4倍的切换软元件(详细参见4.7.8项)
M8199		C253、C255、C253(OP)用的1倍/4倍的切换软元件(详细参见4.7.8项)

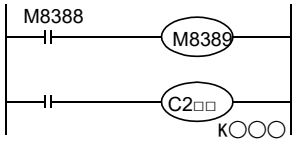
4. 硬件计数器/软件计数器的动作状态

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. 从STOP→RUN时清除

4.7.6 关于外部复位输入信号的逻辑变更

计数器的C241～C245、C247～C250和C252～C255的外部复位输入，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。

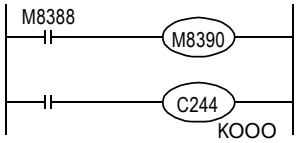
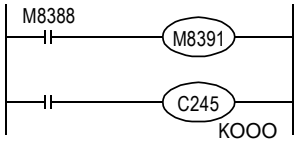
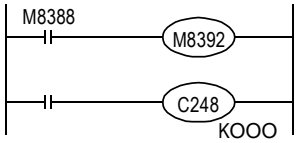
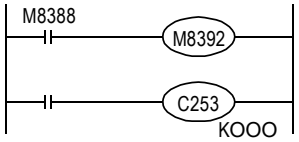
计数器编号	外部复位输入信号的逻辑反转	变化的内容
C241～C245 C247～C250 C252～C255		将外部复位输入的逻辑反转，也就是在OFF的时候复位。 (对象的计数器编号所有的逻辑都反转。)

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。

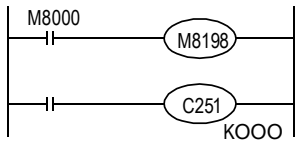
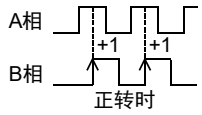
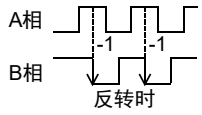
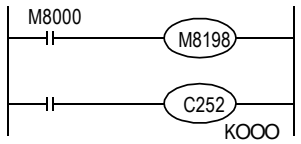
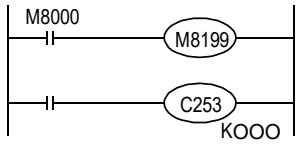
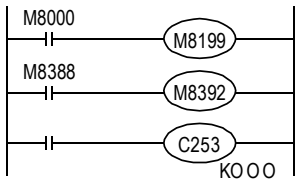
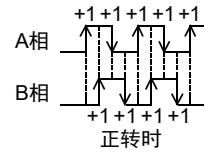
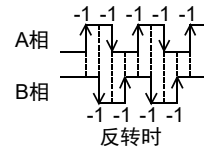
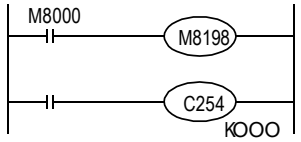
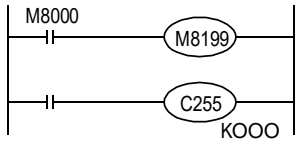
4.7.7 关于计数器的输入端子分配和功能的切换

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	作为硬件计数器使用的场合	变化的内容
C244(OP)		• 计数输入从X000变为X006。 • 没有复位输入。 • 没有启动输入。 • 作为硬件计数器动作。
C245(OP)		• 计数输入从X002变为X007。 • 没有复位输入。 • 没有启动输入。 • 作为硬件计数器动作。
C248(OP)		• 没有复位输入。 • 作为硬件计数器动作。
C253(OP)		• 没有复位输入。 • 作为软件计数器动作。

4.7.8 以4倍频使用双相双计数的计数器C251～C255的方法

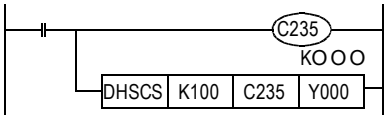
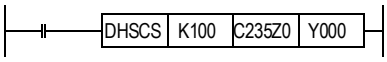
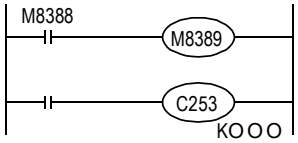
双相双计数输入计数器C251～C255，通常是1倍的，但是如果按照下表所示编程的话，可以以4倍动作。

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

4.7.9 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	作为软件计数器使用的条件
C235 C236 C237 C238 C239 C240 C244(OP) C245(OP) C246 C248(OP) C251 C253	硬件计数器，就是用A8/A8C的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器相同。 高速计数器中，可以通过M8380～M8387来确认是以硬件计数器，还是软件计数器动作。 • 对硬件计数器编号,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为软件计数器。 • 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，所有的硬件计数器和软件计数器的处理方法相同。 例如: C235Z0  • 通过外部复位输入信号逻辑变更功能，执行逻辑反转以后，C253(硬件计数器) 变得和软件计数器的处理方法相同。 例如: 反转C253的外部复位信号的逻辑。 →关于外部复位输入信号的逻辑反转，请参考4.7.6项 

4.7.10 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。

→ 关于可以作为软件计数器使用的条件请参考上一页

		计数器编号	最大响应频率
			基本单元
单相单计数的输入		C235,C236,C237,C238,C239,C240	100kHz
		C244(OP),C245(OP)	10kHz
单相双计数的输入		C246,C248(OP)	100kHz
双相双计数的输入	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。

→ 关于可以作为软件计数器使用的条件请参考上一页

1) 不使用模拟量特殊适配器或A8/A8C系列的特殊功能模块/单元时

计数器的种类				综合频率计算用的倍率	根据使用指令的条件而定的响应频率和综合频率							
		软件计数器	下面的计数器中和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指令 HSCT指令两者	
					最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)
单相单计数输入		C241,C242,C243,C244,C245	C235,C236,C237,C238,C239,C240	×1	40	80	30	60	40-(指令使用次数)*2	80-1.5×(指令使用次数)	30-(指令使用次数)*2	60-1.5×(指令使用次数)
		—	C244(OP),C245(OP)	×1	10		10					
单相双计数输入		C247,C248,C249,C250	C246,C248(OP)	×1	40		30					
双相双计数输入	1倍	C252,C253(OP),C254,C255	C251,C253	×1	40		30					
	4倍			×4	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A8/A8C系列的特殊功能模块/单元时

计数器的种类				综合频率计算用的倍率	根据使用指令的条件而定的响应频率和综合频率							
		软件计数器	下面的计数器中和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指令 HSCT指令两者	
					最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)	最大响应频率(kHz)	综合频率(kHz)
单相单计数输入		C241,C242, C243,C244, C245	C235,C236, C237,C238, C239,C240	×1	30	60	25	50	30-(指令使用次数)*2	50-1.5×(指令使用次数)	25-(指令使用次数)*2	50-1.5×(指令使用次数)
		—	C244(OP), C245(OP)	×1	10		10					
单相双计数输入		C247,C248, C249,C250	C246,C248(OP)	×1	30		25					
双相双计数输入	1倍	C252, C253(OP), C254, C255	C251,C253	×1	30		25					
	4倍			×4	7.5		6.2		(30-指令使用次数)÷4		(25-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。

3. 有关综合频率的计算

综合频率≧[高速计数器的响应频率×综合频率计算用倍率]的合计

4. 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A8/A8C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	使用的指令
C237	作为软件计数器动作	30kHz	40-6(次)=34kHz	×1	HSZ指令6次
C241	软件计数器	20kHz	40-6(次)=34kHz	×1	
C253(OP)[4倍]		4kHz	{40-6(次)}÷4=8.5kHz	×4	

1)使用的指令是HSZ指令使用了6次，所有按照下面的公式计算出综合频率。

综合频率 = 80 - 1.5×6 = 71kHz

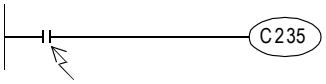
2)使用的高速计数器的响应频率的合计如下所示。

『30kHz ×1[C237]』+『20kHz ×1[C241]』+『4kHz ×4[C253(OP)]』= 66kHz ≦ 71kHz

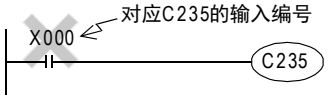
4.7.11 使用上的注意事项

- 高速计数器的线圈驱动用触点，在高速计数时，请使用一直为ON的那种触点。

例：M8000(RUN监控)



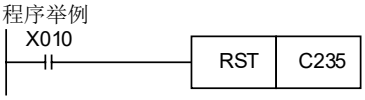
计数时请使用一直为ON的触点
进行编程



指定了计数用输入继电器的编号以后，
高速计数器不能正确进行计数。

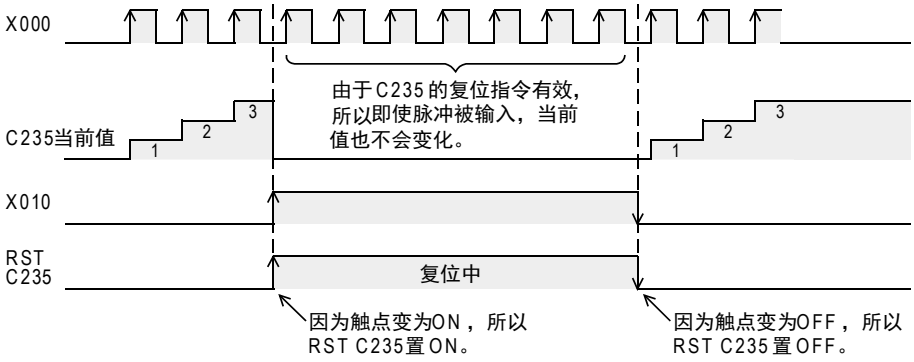
- 如果用模拟开关等有触点的设备执行高速计数器的动作时，由于开关的振动，计数器可能出现计数误差，请注意。
  高速计数器中使用的基本单元输入端子的输入滤波器会被自动设定为5μs(X000～X005)，
  或是50μs(X006,X007)。
- 因此，不需要使用REFF指令和特殊数据寄存器D8020(输入滤波器的调节)。
  此外，不作为高速计数器输入使用的输入继电器的输入滤波器维持10ms(初始值)。
- 输入 X000 ～ X007，用于高速计数器、输入中断、脉冲捕捉以及 SPD、DSZR、DVIT、ZRN 指令和通用输入。因此，请勿重复使用输入端子。
  例如使用了C251时，由于X000、X001都被占用了，所以[C235、C236、C241、C244、C246、C247、C249、C252、C254]、[输入中断指针I00*、I10*]、[脉冲捕捉用触点M8170、M8171]和[适用输入的SPD指令]都不能使用。
- 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。
- 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。
- 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错误，且并联链接不能正常运行。请务必注意。
- 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。

1) 程序举例



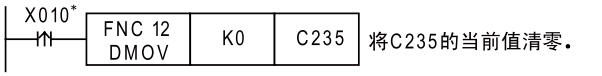
X010为ON期间，C235复位。

2) 时序图



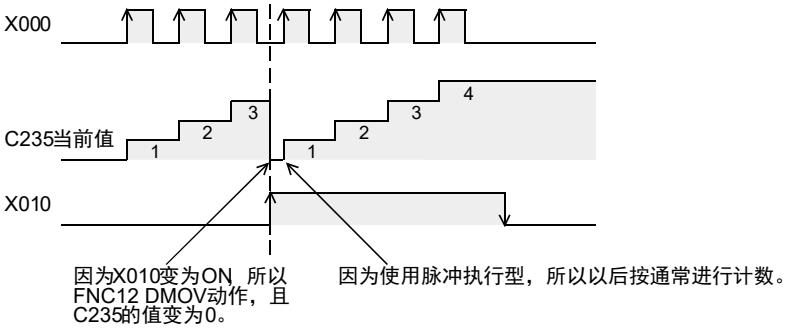
- 当高速计数器，[允许仅清除当前值(触点不为OFF)]时，请如下所示设置。

1) 程序举例



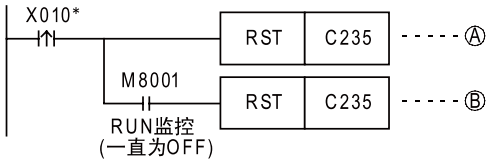
* 驱动触点为连续执行的情况下，X010变为ON时，每执行一个扫描，计数器的当前值被清除为0。

2) 时序图



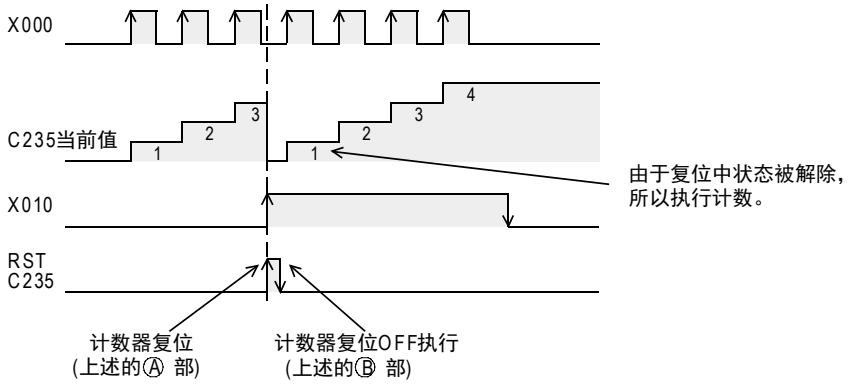
- 希望执行高速计数器[触点的OFF和当前值的复位]时，请按照下面的内容操作。

1) 程序举例



* 驱动触点为连续执行型时，X010变为ON时，每执行一个扫描，计数器复位和计数器复位解除就会被执行。

2)时序图



4.8 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。

→ 有关文件寄存器的功能及动作，请参考4.9.4项

4.8.1 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)

A8/A8C可编程控制器

数据寄存器				文件寄存器 (保持)
一般用	停电保持用 电池保持	停电保持专用 (电池保持)	特殊用	
D0～D199 200点*1	D200～D511 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。根据设定的参数，可以更改为停电保持(保持)区域。

*2. 停电保持区域(保持)。根据设定的参数，可以更改为非停电保持区域。

*3. 关于停电保持的特性不能通过参数进行变更。

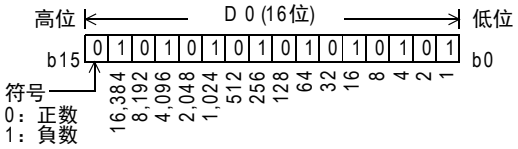
*4. 根据设定的参数，可以将D1000以后的数据寄存器以500点为单位作为文件寄存器。

→ 参考通信控制手册

4.8.2 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。

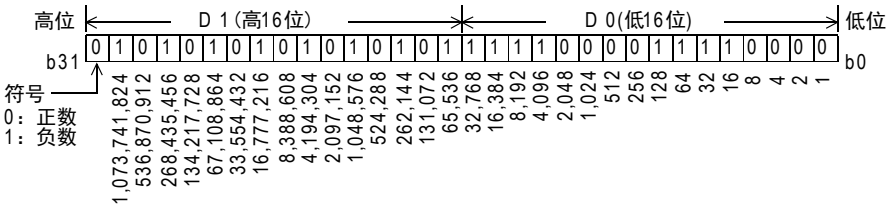
2) 32位

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，低位编号小。

－变址寄存器的V为高位，Z为低位。

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:D0)，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。

4.8.3 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。

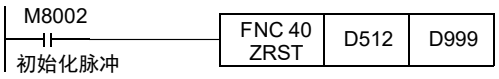
1. 一般用/停电保持用

- 数据寄存器中的数据一旦被写入，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，如果驱动特殊辅助继电器M8033，即使RUN→STOP时也能保持。
- 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。
  A8/A8C可编程控制器时，通过可编程控制器内置的电池执行数据寄存器的停电保持。

→ 有关各个停电保持方法的详细内容，请参考2.6节

- 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。

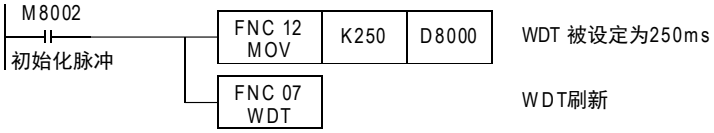
  A8/A8C可编程控制器的举例



→ 有关文件寄存器，参考4.9.4项

2. 特殊用

- 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。)
- 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(FNC 12)可以向D8000中写入目的时间。

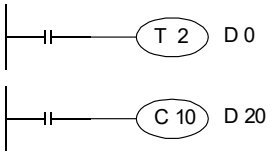


→ 有关特殊数据寄存器的停电保持特性，请参考2.6节及36章
→ 有关特殊数据寄存器的种类和功能，请参考36章

3. 动作举例

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。

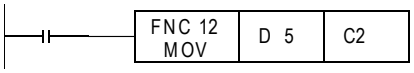
1) 基本指令中的数据寄存器
指定为定时器和计数器的设定值。



指定的数据寄存器中的内容作为各计数和定时器的设定值进行动作。

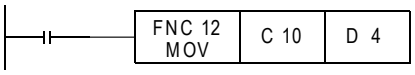
2) 应用指令中的数据寄存器
MOV(FNC 12)指令的动作实例

a) 更改计数器的当前值。



计数器(C2)的当前值改成D5的内容。

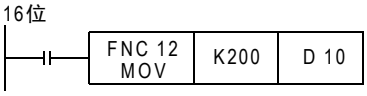
b) 将定时器和计数器的当前值读出到数据寄存器中。



计数器(C10)的当前值被传送至D4。

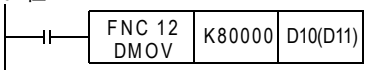
c) 数值保存在数据寄存器中。

16位



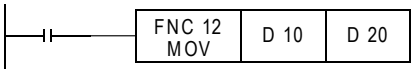
将200(10进制数)传送至D10。

32位



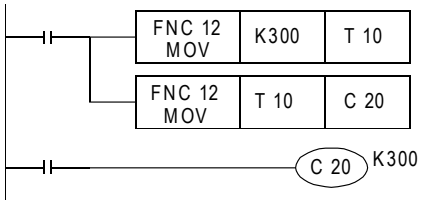
80,000(10进制数)传送至D10(D11)。
由于超出32,767的数值是32位的，所以使用32位运算。数据寄存器指定了低位侧(D10)后，高位侧(D11)会自动被占用。

d) 将数据寄存器的内容传送至其他数据寄存器中。



将D10的内容传送至D20。

3) 将未使用的定时器及计数器作为数据寄存器使用
MOV(FNC 12)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。



将300(10进制数)传送至T10。

将T10的内容传送至C20的当前值寄存器中。
此时，T10不起到定时器的作用，而是作为数据寄存器工作。

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。

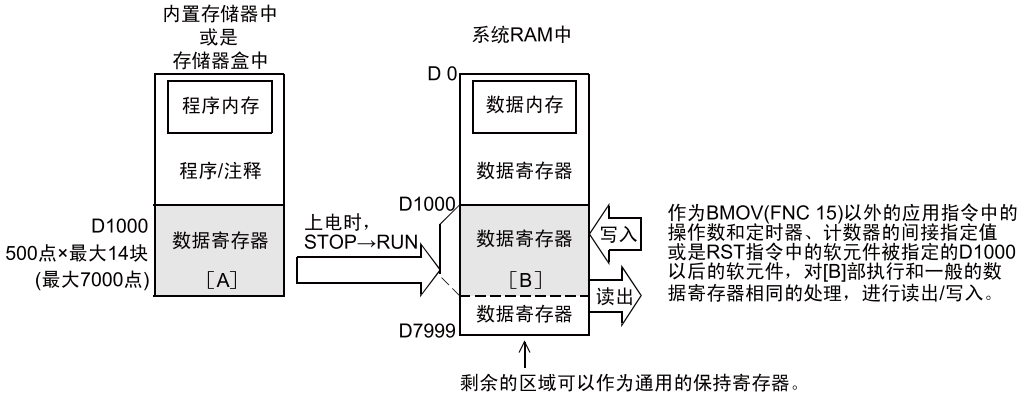
4.8.4 文件寄存器的功能和动作实例

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。
根据设定的参数，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。

- 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。
- 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. 文件寄存器的动作

- 当可编程控制器上电时和STOP→RUN时，在内置存储器、或是存储器盒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。
  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。



- BMOV(FNC 15)指令和其他指令的区别
  针对文件寄存器(D1000以后)的BMOV(FNC 15)指令和其他指令的区别如下表所示。

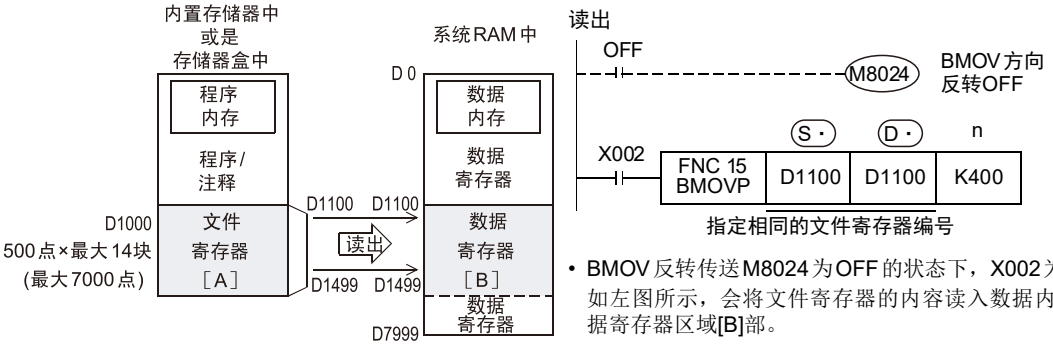
指令	传送内容	备注
BMOV命令	可以对程序内存中的文件寄存器区域[A]部进行读出/写入。	—
BMOV指令以外的应用指令等	对于映象存储区中的数据寄存器区域[B]部，可以和普通的数据寄存器一样地进行读出/写入的处理。	由于数据寄存器区域[B]部，是设在可编程控制器内的系统RAM中，因此不受存储器盒形式的限制，可以随意地更改内容。

被设定为文件寄存器的数据寄存器，在上电时数据会自动地从文件寄存器区域[A]部复制到数据寄存器区域[B]部。

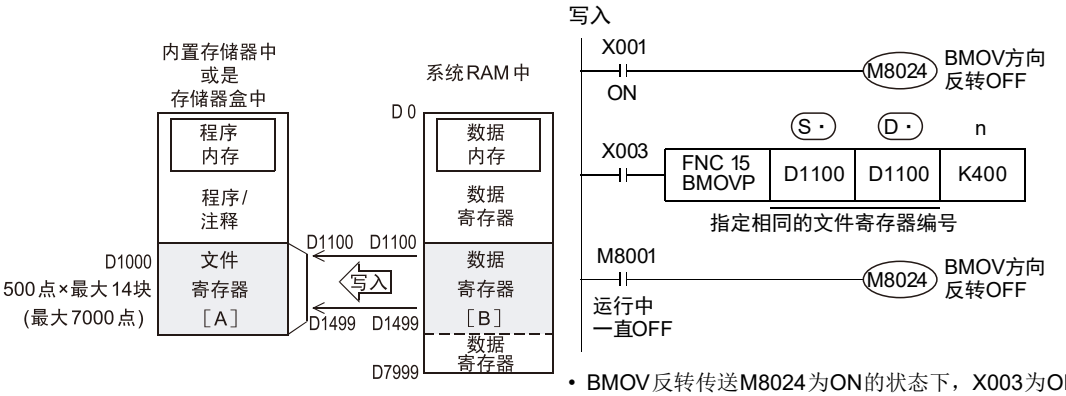
- 通过外围设备对文件寄存器进行监控时，读出数据内存中的数据寄存器区域[B]部。
  此外，在外围设备上执行文件寄存器软元件的[更改当前值]、[强制复位]或是[PLC存储器的全部清除]的时候，先对程序内存中的文件寄存器区域[A]部进行更改，然后自动传送给数据寄存器区域[B]部。
  因此，执行文件寄存器软元件改写时，程序内存需要在[内置存储器]、或是[存储器盒]的“写保护开关OFF”的状态。(存储器盒的写保护开关如果为ON，就不能从外围设备上进行更改。)

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

BMOV(FNC 15)指令的 (S·) 、(D·) 都指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，会执行以下的动作。



- BMOV反转传送M8024为OFF的状态下，X002为ON时，如左图所示，会将文件寄存器的内容读入数据内存中的数据寄存器区域[B]部。

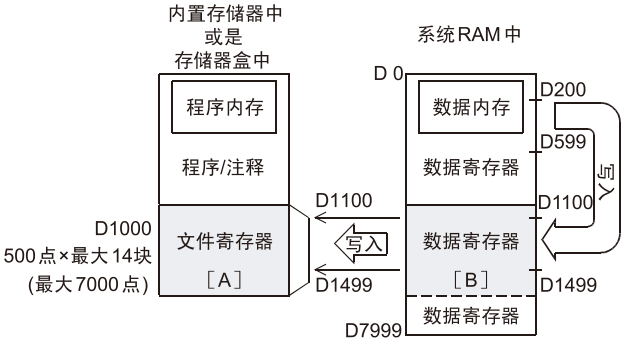


- BMOV反转传送M8024为ON的状态下，X003为ON时，如左图所示，会将数据内存中的数据寄存器的内容自动写入到程序内存的文件寄存器区域中。

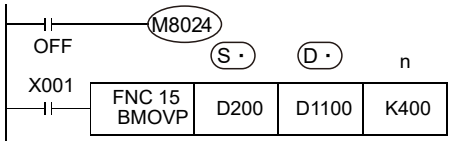
- 更新相同编号的文件寄存器的时候，必须将文件寄存器的编号设定为 (S·)=(D·) 。此外，设定的时候以n指定的传送点数不能超出文件寄存器区域。如超出文件寄存器区域，会出运算错误，而不能执行指令。
- 对 (S·)，(D·) 采用变址修饰时，实际的软元件编号要在文件寄存器区域内，与此同时，只有当传送点数在文件寄存器区域范围内，才能执行指令。

3. 数据寄存器→文件寄存器<使用BMOV(FNC 15)指令写入>

对BMOV(FNC 15)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。



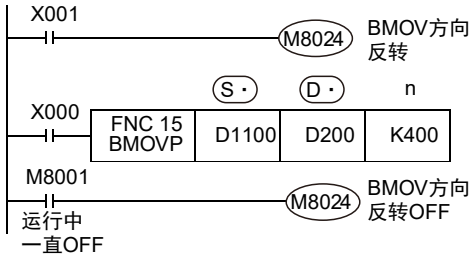
写入



- X001为ON后，如左图所示，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，当[A]部由于存储器盒的写保护开关为ON而不能写入的时候，只能对[B]部进行写入。
  使用一般应用指令，在 (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。
- 也可以在 (S・) 中指定文件寄存器，但是如果指定了和 (D・) 相同的编号则变成相同编号更新模式。

→ 有关相同编号的更新，参考上一页

- 通过控制BMOV(FNC 15)的BMOV反方向传送M8024，可以在1个程序中实现双向传送。(下图)



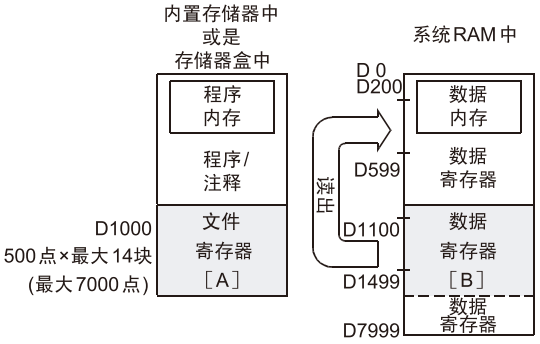
(S・) → (D・) 数据寄存器的成批传送
M8024(OFF):D1100 D200

(S・) ← (D・) 数据寄存器的成批传送
以及文件寄存器的写入
M8024(ON) :D1100 D200

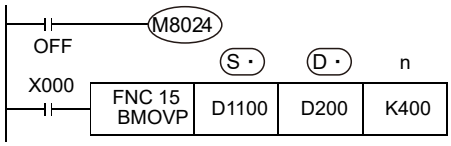
读出的注意事项

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



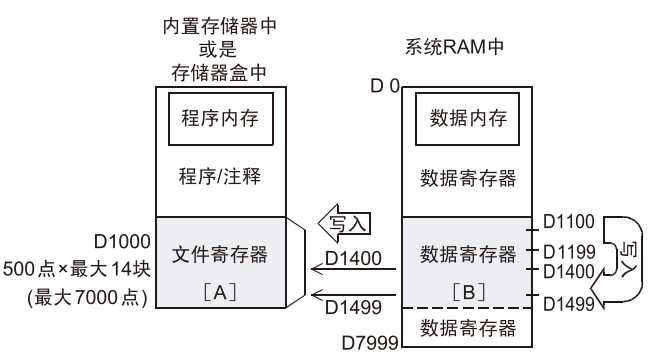
数据寄存器的成批传送



- X000为ON后，如左图所示读出数据寄存器区域[B]部的内容。
- 也可以在 (D・) 中指定文件寄存器，但是如果指定了和 (S・) 相同的编号则变成同编号更新模式。

→ 有关相同编号的更新，参考上一页

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



• X001为ON后，如左图所示，数据寄存器区域[B]部的数据会被传送到数据寄存器区域[B]部和文件寄存器区域[A]部中。此外，当[A]部由于存储器盒的写保护开关为ON而不能写入的时候，只能对[B]部进行写入。

4.8.5 使用文件寄存器的注意事项

1. 使用存储器盒的注意事项

更改存储器盒中的文件寄存器内容时，请按照以下条件执行。

- 存储器盒的写保护开关请置于[OFF]一侧。
- 写入到存储器盒(闪存)时，连续的1个块 (500点)需要66～132ms。写入到存储器盒(EEPROM)时，连续的1个块(500点)需要80ms。
  在此之间的程序执行被中断。此时，看门狗定时器不会被刷新，因此需要在用户程序中采取插入WDT指令等的对应措施。*1

*1. 只对应A8/A8C可编程控制器。

2. 关于存储器的允许写入次数

- 存储器盒允许写入次数在1万次以下。
  通过程序进行写入时，如使用连续执行型指令，在可编程控制器的每个运算周期中都会对存储器写入。
  如要避免这种情况，必须使用脉冲执行型(BMOVP)指令。

3. 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

- 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·)。
- n指定的传送点数请勿超出文件寄存器的范围。
- 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。
- 变址修饰时
  (S·)、(D·)的变址修饰其实际的软元件编号在文件寄存器区域中，与此同时，传送点数也在文件寄存器范围内，只有这样才执行指令。

4.9 扩展寄存器[R]、扩展文件寄存器[ER]

扩展寄存器(R)是扩展数据寄存器(D)用的软元件。
此外，扩展寄存器(R)的内容也可以保存在扩展文件寄存器(ER)中。但是，这个扩展文件寄存器为A8/A8C可编程控制器时，只有使用存储器盒时可以利用。

4.9.1 扩展寄存器、扩展文件寄存器的编号

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

A8/A8C可编程控制器

扩展寄存器(R) (电池保持)	扩展文件寄存器(ER) (文件用)
R0～R32767 32768点	ER0～ER32767 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)。

4.9.2 数据的存储地点和访问方法

由于扩展寄存器(R)和扩展文件寄存器，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	可编程控制器	数据存储地点
扩展寄存器	A8/A8C	内置RAM(电池后备的区域)
扩展文件寄存器	A8/A8C	存储器盒(闪存)

访问方法的差异

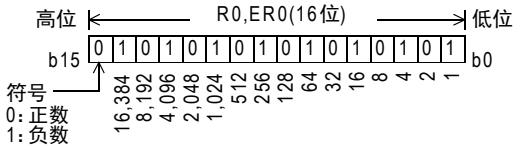
访问方法		扩展寄存器	扩展文件寄存器
程序中读出		○	△仅专用指令可以
程序中写入		○	△仅专用指令可以
显示模块		○	○
数据的变更方法	HCP-works的在线测试操作	○	×
	使用HCP-works进行成批写入	○	○
	计算机链接功能	○	×

4.9.3 扩展寄存器，扩展文件寄存器的构造

扩展寄存器由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个扩展寄存器(16位)中，可以处理-32768～+32768的数值。



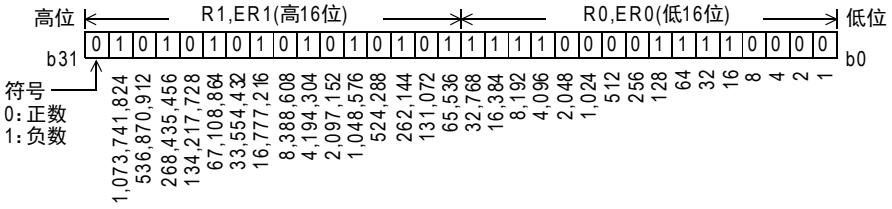
一般使用应用指令对扩展寄存器进行数值的读出/写入。

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个扩展寄存器，显示32位数据。(高位编号大，低位编号小)

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



- 指定32位时，指定低位侧(例如:R0)后高位侧会被紧接(例如:R1)的号码自动占用。
  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，建议低位侧使用偶数的软元件编号。

4.9.4 扩展寄存器、扩展文件寄存器的初始化

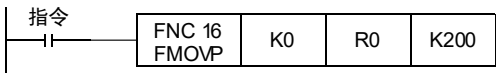
A8/A8C可编程控制器，使用选件的电池设定更改为保持用时，即使执行了[电源OFF]和[STOP→RUN的操作]，扩展寄存器的内容也通过内置电池被保持。

对扩展寄存器的内容初始化时，请通过使用顺控程序或是HCP-works执行数据清除的操作。

1. 在程序中执行的方法

- 一部分的扩展寄存器(R)的初始化

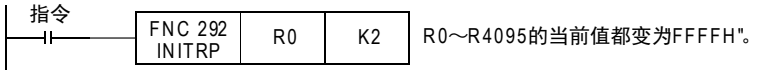
  例如)将R0～R199初始化(清除)的时候



- 以段为单位初始化扩展寄存器以及扩展文件寄存器

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

  A8/A8C可编程控制器的举例



2. 在HCP-works中执行的方法

在HCP-works中，选择[Online]→[Clear PLC memory]后，清除[Data device]。

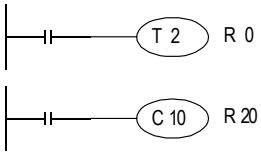
但是，执行这个操作的时候，定时器、计数器、数据寄存器、文件寄存器以及扩展寄存器的内容都被初始化。

4.9.5 扩展寄存器的功能和动作实例

扩展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，从这些用途中选取了基本指令和应用指令的代表例说明动作。
此外，为了能有效使用扩展寄存器，请阅读后述的应用指令说明。

1. 基本指令中的扩展寄存器

- 指定为定时器和计数器的设定值。

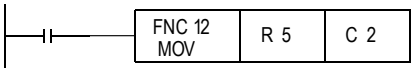


计数器和定时器将被指定的扩展寄存器的内容作为各自的设定值而动作。

2. 应用指令中的扩展寄存器

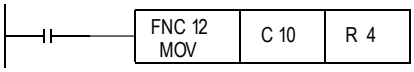
MOV(FNC 12)指令的动作实例

- 更改计数器的当前值。



计数器(C2)的当前值更改为R5的内容。

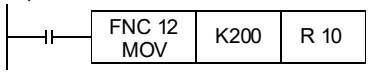
- 将计数器的当前值读入扩展寄存器中。



将计数器(C10)的当前值传送给R4。

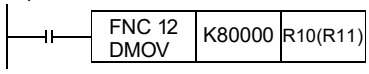
- 将数值保存到扩展寄存器中。

16位



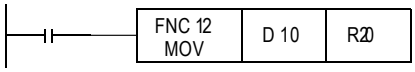
200(10进制数)传送到R10。

32位



80,000(10进制数)传送到R10(R11)。
由于数据超出32,767，所以使用32位双重D指令(32位运算)。指定了扩展寄存器的低位侧(R10)后，高位侧(R11)会被自动占用。

- 将数据寄存器的内容传送至扩展寄存器中。



D10的内容传送给R20。

4.9.6 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置使用。
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扩展寄存器中后，再使用扩展寄存器一侧的软元件。
但是，A8/A8C可编程控制器的情况下，只有当使用了存储器盒时方可使用这种软元件。

• A8/A8C可编程控制器

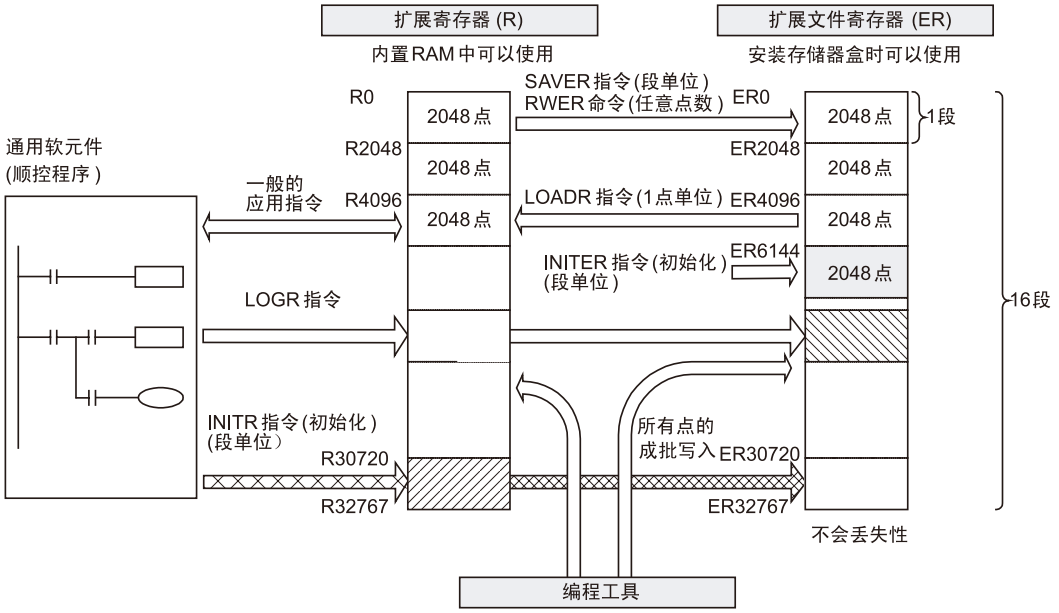
指令	内容
LOADR(FNC 290)	将扩展文件寄存器(ER)*1的数据读出到扩展寄存器(R)中的指令
SAVER(FNC 291)	将扩展寄存器(R)的数据以2048点(1段)为单位写入(传送)到扩展文件寄存器(ER)*1中的指令。 用于将新制作的1段(2048点)的数据保存到扩展文件寄存器(ER)*1的情况。
INITR(FNC 292)	扩展寄存器(R)以及扩展文件寄存器(ER)*1以2048点(1段)为单位进行初始化的指令。 LOGR 指令开始记录数据前，对扩展寄存器(R)以及扩展文件寄存器(ER)*1进行初始化时，使用该指令。
LOGR(FNC 293)	记录指定数据，写入到扩展寄存器(R)和扩展文件寄存器(ER)*1中的指令。
RWER(FNC 294)	将指定的扩展寄存器(R)写入(传送)到扩展文件寄存器(ER)*1中的指令。 A8C HCP works支持。 将任意的扩展寄存器(R)的内容保存到扩展文件寄存器(ER)*1中时，使用该指令。
INITER(FNC 295)	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。 A8C HCP works支持。 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. 扩展文件寄存器和扩展寄存器的关系

在可编程控制器中，扩展文件寄存器和扩展寄存器的位置关系如下图所示。

a) A8/A8C可编程控制器



HCP-works

2. 有关扩展寄存器和扩展文件寄存器的段

在数据结构上，A8/A8C可编程控制器的扩展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～ER16383,R14336～R16383

段编号	起始软元件编号	软元件范围
段8	R16384	ER16384～ER18431,R16384～R18431
段9	R18432	ER18432～ER20479,R18432～R20479
段10	R20480	ER20480～ER22527,R20480～R22527
段11	R22528	ER22528～ER24575,R22528～R24575
段12	R24576	ER24576～ER26623,R24576～R26623
段13	R26624	ER26624～ER28671,R26624～R28671
段14	R28672	ER28672～ER30719,R28672～R30719
段15	R30720	ER30720～ER32767,R30720～R32767

4.9.7 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(A8/A8C可编程控制器)

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

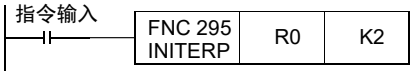
- 使用SAVER指令将数据写入扩展文件寄存器中时
  执行该指令前，请先将作为写入对象的段进行初始化。而且，请在初始化后，将写入数据保存到扩展寄存器中。
  A8C系列HCP-works的时候，使用REER指令后无需对写入对象的段进行初始化。
- 使用LOGR指令，将数据写入扩展文件寄存器中时
  数据开始记录之前，请先对写入对象的段进行初始化。
- 使用INITR指令时
  该指令是对被指定的段的扩展寄存器以及扩展文件寄存器的内容进行初始化。如果仅仅使用该指令对扩展文件寄存器初始化时，必须在执行指令前，将扩展寄存器的内容暂时保存到不使用的扩展寄存器或不使用的数据寄存器中。
  A8C系列HCP-works以上产品中，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或内置EEPROM中的，因此可以通过顺控程序或HCP-works中的数据清除操作来执行初始化。
A8/A8C可编程控制器中，将数据写入扩展文件寄存器中时，需要先将写入对象区域进行初始化。

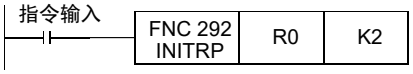
1) 在程序中执行的方法(仅适用A8/A8C可编程控制器)

a) 以段为单位仅仅初始化扩展文件寄存器。[Ver.1.30以上]
例如)ER0～ER4095的初始化(ER0的起始2段的初始化)



ER0～ER4095的当前值都变为"FFFFH"。

b) 以段为单位初始化扩展寄存器以及扩展文件寄存器
例如)R0～R4095和ER0～ER4095的初始化(R0和ER0的起始2段的初始化)



R0～R4095和ER0～ER4095的当前值都变为"FFFFH"。

2) 在HCP-works中执行的方法
在HCP-works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寄存器的内容都被初始化。

3. 关于存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点。

- A8/A8C可编程控制器的情况下
  存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。

  即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，也不被计入存储器的写入次数。
  但是，执行SAVER(FNC 291)、LOGR(FNC 293) 指令之前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令进行初始化时，
  因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，所以请注意存储器的写入次数。

4.9.8 关于扩展寄存器及扩展文件寄存器的数据登录

HCP-works操作。

→ 有关HCP-works操作的详细内容，参考HCP-works的手册

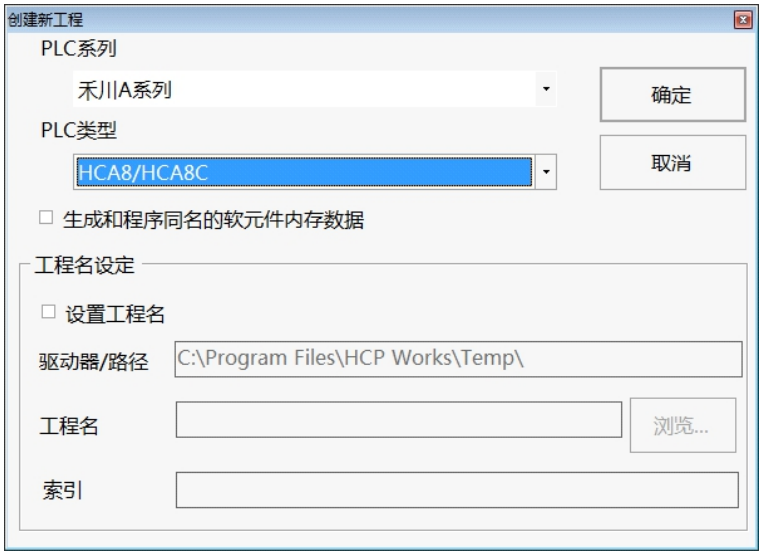
1 启动HCP-works。

点击Windows→的[开始]－[程序]－[HCP-works]。

2 设定可编程控制器的机型。

按照下表内容设定“PLC Series”和“PLC Type”。

必须设定项目	设定内容
PLC Series	HCA系列CPU
PLC Type	A8(C)*1



*1. 在HCP-works中，请将PLC型号选择为A8C。

3 数据的设定。

1. 右键点击Project datda list中的[Device memory]，打开子菜单。
2. 点击[New...]后显示 “New” 对话框。
3. 点击[OK]后，显示确认用的对话框。

4. 在 “Device Label” 中输入要设定的软元件编号，然后点击[Display]按钮。
5. 在 “Display” 按钮右边的2个的选择框中，选择要设定的数据类型。
6. 根据要设定的数据的内容，将数据输入到各个软元件中，或输入到字符串中。

4 写入(传送)到可编程控制器中。

1. 选择[Online]→[Write to PLC]后，打开“Write to PLC”对话框。
2. 选择 “软元件内存”的 “MAIN(制作的软元件内存的名称)”。
3. 点击[Execute]按键后，执行写入(传送)。

4.10 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。

4.10.1 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。

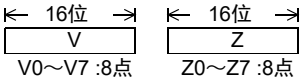
变址用
V0(V)～V7，Z0(Z)～Z7 16点*1

*1. 关于停电保持的特性可以通过参数进行变更。

4.10.2 功能和构造

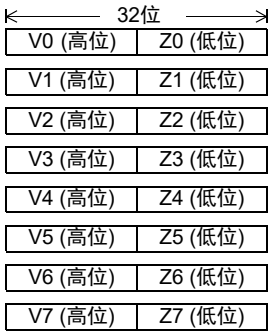
1. 16位

变址寄存器具有和数据寄存器相同的结构。



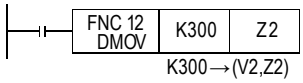
2. 32位

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HCA可编程控制器将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值时，会变成相当大的数值，从而出现运算错误。

32位变址寄存器的写入实例



即使32位应用指令中使用的变址值没有超出16位数值范围，也请按照上图所示在对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)。

4.10.3 软元件的修饰

可以被修饰的软元件，极其修饰的内容如下所示。

→ 有关变址修饰的使用方法、注意事项，请参考5.7节

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)。

4.11 指针[P]、[I]

4.11.1 指针的编号

指针(P)、(I)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(FNC 56)]等一起使用。

A8/A8C可编程控制器

分支用	END跳转用	输入中断 输入延迟中断用	定时器中断用	计数器中断用
P0～P62 P64～P4095 4095点	P63 1点	I00□(X000) I30□(X003) I10□(X001) I40□(X004) I20□(X002) I50□(X005) 6点	I6□□ I7□□ I8□□ 3点	I010 I040 I020 I050 I030 I060 6点

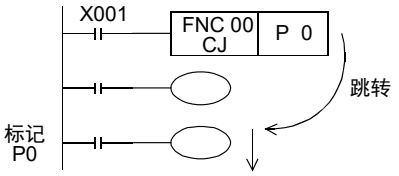
4.11.2 分支用指针的功能和动作实例

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。

→ 中断功能的详细内容，请参考35章

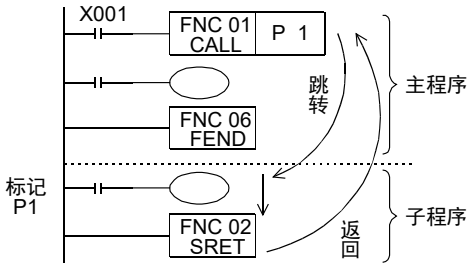
1. 使用分支用指针(P)的应用指令

- CJ(FNC 00)条件跳转



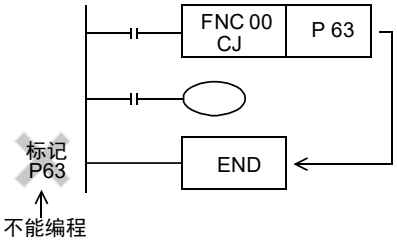
X001为ON，跳转到CJ(FNC 00)指令指定的标记位置，执行之后的程序。

- CALL(FNC 01)子程序调用



X001为ON，执行CALL(FNC 01)指令指定的标签位置的子程序，使用SRET(FNC 02)指令返回到原来的位置。

• END跳转用指针P63的作用



P63是表示使用CJ(FNC 00)指令时跳跃到END步的特殊指针。
因此，将P63作为标签编程时，程序会错误，请注意。
→ 请参考8.1节5.不需要输入指针P63的标记

4.11.3 中断用指针的功能和动作实例

→ 中断功能的详细内容，参考35章

中断用指针包括以下3种，与应用指令IRET(FNC 03)中断返回、EI(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

→ 输入中断功能的详细内容，参考35.3节和35.4节

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

输入	输入中断用指针		禁止中断标志位	输入信号的ON脉宽或是OFF脉宽
	上升沿中断	下降沿中断		A8/A8C
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	

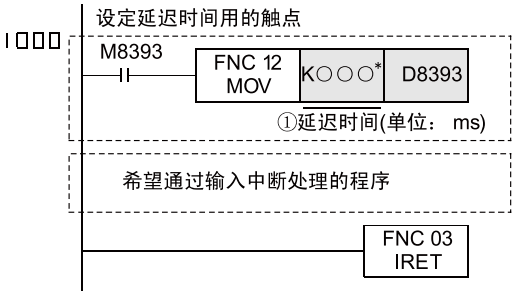
*1. 从RUN→STOP时清除

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，使用输入中断指针[I001]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I000]、[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*2

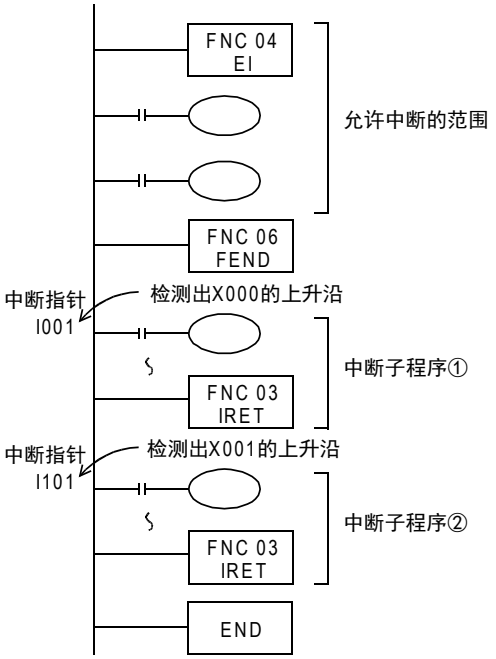
在输入中断中，有以1ms为单位延迟执行中断子程序的功能。
通过执行以下格式的程序来指定要延迟的时间。
使用了这种延迟功能后，在调节输入中断中使用的传感器的安装位置时，可以无需挪动实际的位置而进行电气上的调节。



• 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。
由于这个程序是模板程序，所以届时只需要更改延迟时间（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)。
• 中断程序结束

*2. 仅对应A8/A8C可编程控制器。

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，在扫描程序过程中，X000或X001为ON，执行中断子程序①或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记放在FEND指令后。

2. 定时器中断用: 3点

→ 定时器中断功能的详细内容，参考35.5节

每隔指定的中断循环时间(10ms～99ms)，执行中断子程序。
在可编程控制器的运算周期以外，需要循环中断处理的控制中使用。

输入编号	中断周期(ms)	中断禁止标志位
I6□□	在指针名的□□中，输入10～99的整数。 例如: I610=每10ms的定时器中断	M8056*1
I7□□		M8057*1
I8□□		M8058*1

*1. 从RUN→STOP时清除

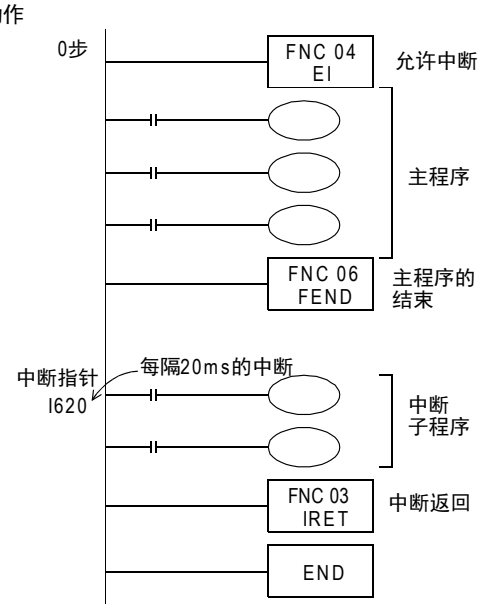
注意

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。

中断程序的处理时间较长时

主程序中使用了处理时间较长的指令时

动作



- EI指令以后定时器中断变为有效。
  此外，不需要定时器中断的禁止区间时，
  就不需要编写DI(禁止中断指令)。
- FEND表示主程序的结束。
  中断子程序必须编写在FEND后。
- 每隔20ms执行中断子程序。
  使用IRET指令返回到主程序。

3. 计数器中断用:6点*1

→ 计数器中断功能的详细内容，参考35.6节

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

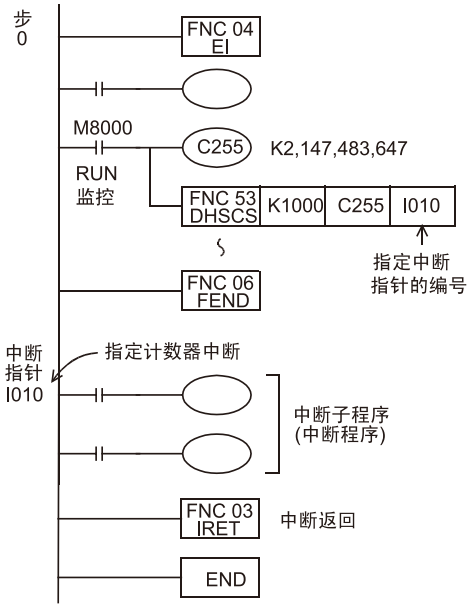
指针编号	中断禁止标志位
I010	M8059*2
I020	
I030	

指针编号	中断禁止标志位
I040	M8059*2
I050	
I060	

*1. 仅对应A8/A8C可编程控制器。

*2. 从RUN→STOP时清除

动作



- 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(FNC 53)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。

5. 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。

- 10进制数，16进制数和实数的常数指定
- 位软元件的位数指定
- 数据寄存器的位位置指定
- 特殊模块/单元的BFM(缓冲存储区)的直接指定
- 附加了变址寄存器的变址修饰

5.1 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HC系列可编程控制器中，根据各自的用途和目的不同，有5种数值可供使用。
其作用和功能如下所示。

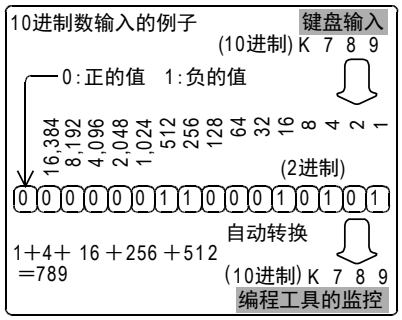
5.1.1 数值的种类

1. 10进制数(DEC: DECIMAL NUMBER)
   - 定时器和计数器的设定值(K常数)
   - 辅助继电器(M)、定时器(T)、计数器(C)、状态(S)等的软元件编号(软元件编号)
   - 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: HEXADECIMAL NUMBER)
   - 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: BINARY NUMBER)

   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，会如右图所示，自动转换成10进制数后显示。(也可以切换成16进制)

   - 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NEG(FNC 29)指令的说明。

4. 8进制数(OCT: OCTAL NUMBER)

   HC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: BINARY CODE DECIMAL)

   BCD就是将构成10进制数的各位上0～9的数值以四位的BIN来表现的形式。
   由于各位便于使用，所以使用于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(浮点数数据)

   A8/A8C可编程控制器，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。

5.1.2 数值的转换

HC可编程控制器中处理的数值，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	0011	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数字开关、7段码显示器

5.1.3 浮点运算中数值的处理

A8 A8C

浮点运算中数值的处理

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。

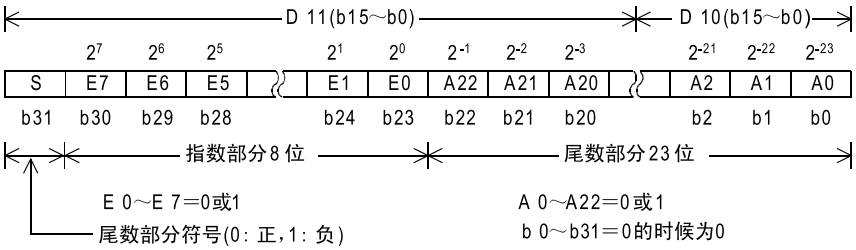
整数的开方运算中小数点也被舍去。

A8/A8C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。

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)时，如下图所示。



2进制浮点数(实数) = $\pm(2^0+A22\times2^{-1}+A21\times2^{-2}+\cdots+A0\times2^{-23})\times2^{(E7_{\times2^7}+E6\times2^6+\cdots+E0\times2^0)}/2^{127}$

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

2进制浮点数(实数) = $\pm(2^0+1\times2^{-1}+0\times2^{-2}+1\times2^{-3}+\cdots+0\times2^{-23})\times2^{(1\times2^7+0\times2^6+\cdots+1\times2^0)}/2^{127}$

= $\pm1.625\times2^{129}/2^{127}=\pm1.625\times2^2$

正负由b31的符号位决定，但不是补码处理。

有效位数

2进制浮点数的有效位数如用10进制数表示，大约为7位数。2进制浮点数的处理范围如下所示。

- 最小绝对值1175494×10^{-44}
- 最大绝对值3402823×10^{32}

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。

- 零标志位　：结果真为0时为1
- 借位标志位：结果未达到最小单位，但不是0时为1
- 进位标志位：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

HCP-works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。

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，内部的运算仍然是采用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，编号小的为尾数部分，编号大的为指数部分。
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]×10[指数D1]
尾数D0=±(1,000～9,999)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，假如不存在100。在为100的时候，就变成1000×10^{-1}。
10进制浮点数(实数)的处理范围如下所示。

－ 最小绝对值1175×10^{-41}
－ 最大绝对值3402×10^{35}

有效位数
10进制浮点数的有效位数如用10进制数表示大约为4位数。10进制浮点数的使用范围如上所示。

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: FNC 118([D]EBCD)
－ 10进制浮点数(实数) →2进制浮点数(实数)的转换: FNC 119([D]EBIN)

5.2 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，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)。
其作用和功能如下所示。

5.2.1 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。

－ 使用字数据(16位)时 ・・・ K-32768～K32767
－ 使用2个字数据(32位)时 ・・・ K-2,147,483,648～K2,147,483,647

5.2.2 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。

－ 使用字数据(16位)时・・・H0～HFFFF　　(BCD数据的时候为H0～H9999)
－ 使用2个字数据(32位)时・・・H0～HFFFFFFFF　(BCD数据的时候为H0～H99999999)

5.2.3 常数E(实数)

A8 A8C

[E]是表示实数(浮点数数据)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，-1.0×2^{128}～-1.0×2^{-126}、0、1.0×2^{-126}～1.0×2^{128}。
在顺控程序中，实数可以指定“普通表示”和“指数表示”两种。

－ 普通表示　就将设定的数值指定。
　例如，10.2345就以E10.2345指定。
－指数表示　设定的数值以(数值)×10^n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)。

5.3 字符串

A8 A8C

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。

5.3.1 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。

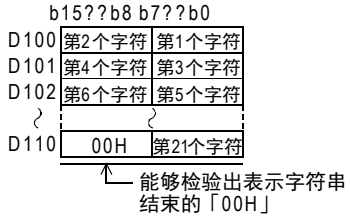
5.3.2 字符串数据

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(OOH)的数据也需要是16位。参考(下面2)中的例2
总之，如下情况时，应用指令中会出现运算错误。(错误代码: K6706)

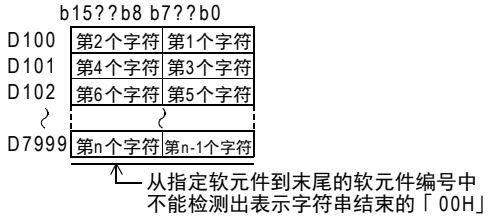
- 在应用指令的源程序中指定的软元件编号以后，相应软元件范围内未设定[00H]的情况。
- 在应用指令的嵌套中指定的软元件中，保存字符串数据(包含了表示字符串数据的末尾的 [00H]或[0000H])用的软元件数不够的情况。

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

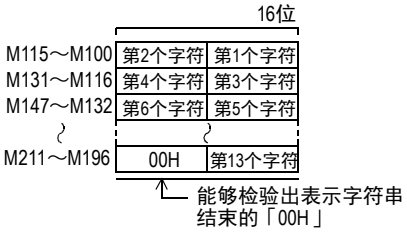


• 不能识别为字符串数据的例子

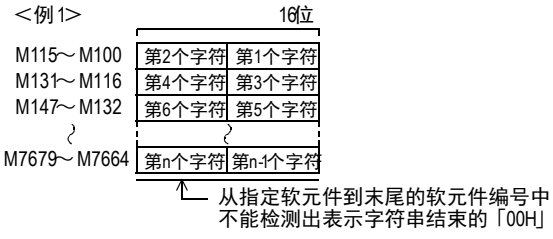


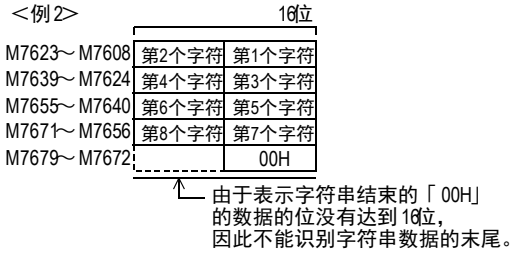
2) 位数指定的位软元件中保存的字符串数据

• 能够识别为字符串数据的例子



• 不能识别为字符串数据的例子

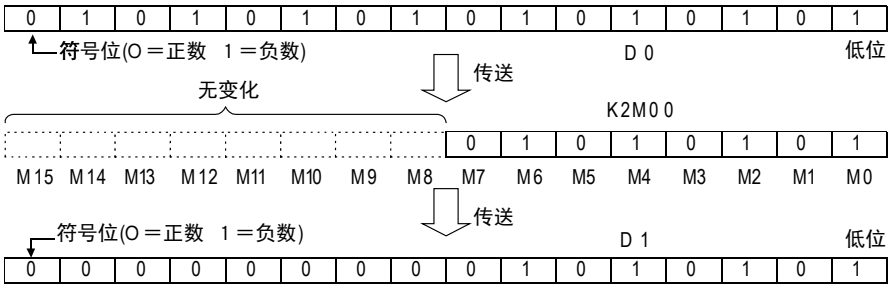




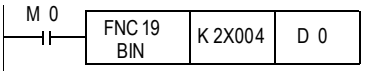
5.4 位的位数指定(K n □***)

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表示。
位数为4位单位的K1～K4(16位数据)、K1～K8(32位数据)。
例如，K2M0，由于是M0～M7，所以是2位数的数据。



向K1M0～K3M0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。



通过X004～X013将BCD2位数的数据转换成BIN后传送至DO

只要没有特别的限制，被指定的位软元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，最理想的是8的倍数，但是为了避免混乱，建议设定为M0、M10、M20…等。

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……
K2Y010、	K2Y020、	Y2X030……	
K3M0、	K3M12、	K3M24、	K3M36……
K4S16、	K4S32、	K4S48……	

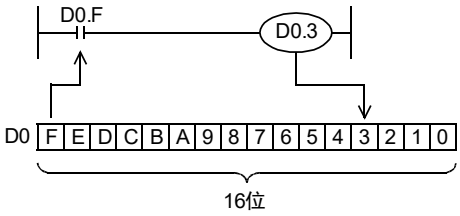
也就是说，在不跳过软元件的前提下，以各位数的单位如上述所示地使用软元件。
但是，32位运算中使用了K4Y000的时候，高16位被视为0。
需要32位数据的时候，要使用K8Y000。

5.5 字软元件的位指定(D□.b)

A8 A8C

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饰。

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)



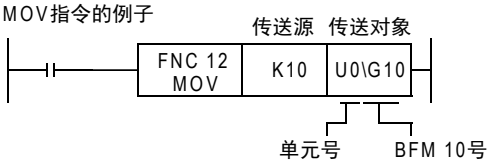
5.6 缓冲寄存器的直接指定(U□\G□)

A8 A8C

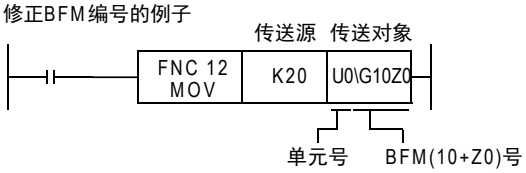
可以直接指定特殊功能模块和特殊功能单元的BFM(缓冲存储器)。
BFM为16位或32位的字数据，主要用于应用指令的操作数。
BFM是接着特殊功能模块或特殊功能单元的模块号(U)和BFM编号(\G)后指定的。
(例如: U0\G0・・・表示模块号为0的特殊功能模块或特殊功能单元的BFM #0号)
此外，在BFM编号中可以进行变址修正。
指定范围如下所示。

单元号(U)・・・・0～7
BFM编号(\G) ・・0～32766

MOV指令的例子



修正BFM编号的例子



5.7 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。

5.7.1 基本指令的变址修正

A8 A8C

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)]都可以进行变址修饰。

下面举在变址寄存器Z(0)中，对LD指令的X000和M0进行修正的例子来说明动作。(右图)

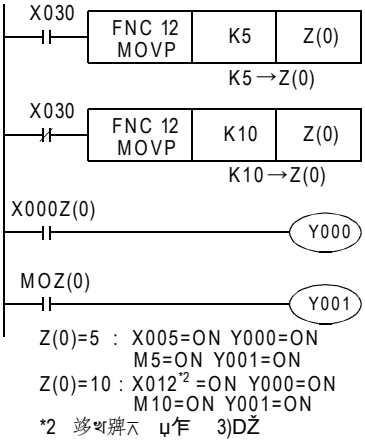
将K5或K10事先传送到变址寄存器Z(0)中。

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。

*1. 请参考注意事项3)。

- 在进行修饰的变址寄存器中，可以使用Z0～Z7、V0～V7。
- 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。



*2 ¾ÍÔÚÖ¸Áî ÊÂÏî 3)DŽ

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饰后，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呈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。

字软元件、常数的情况

可以变址修饰在OUT指令中使用的字软元件[T,C(0～199)]的设定值。

下面例举在变址寄存器V2中，修饰OUT指令TO的设定值D0的例子进行说明。(右图)

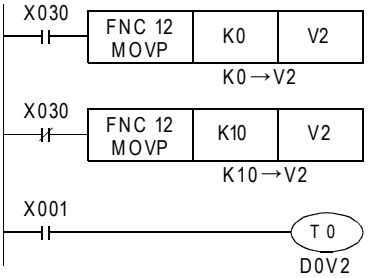
将K0或K10事先传送到变址寄存器V2中。

X001为ON，V2=0的时候，如果[D(0+0)=D0]，设定值为D0，则TO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

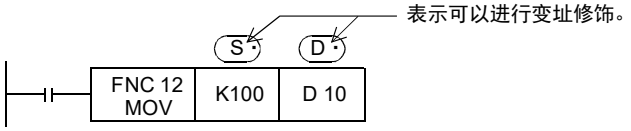
1) OUT指令为32位计数器的时候，不能变址修正设定值。



5.7.2 应用指令的变址修饰

可以变址修饰应用指令的标明方法

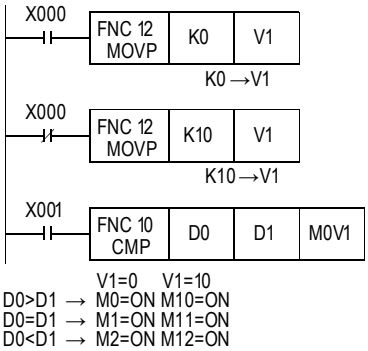
在应用指令说明中，可以进行变址修饰的操作数的表示方法如下图所示，通过在源操作数 (S) 或是目标操作数符号 (D) 中加上以此和不带修饰功能的操作数进行区别。



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。(右图)
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。

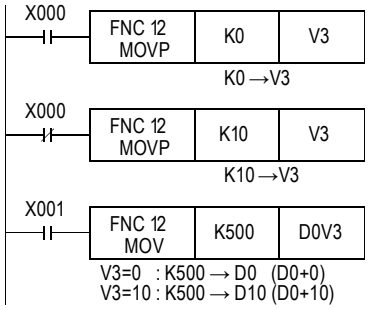
- 在进行修饰的变址寄存器中，可以使用Z0～Z7、V0～V7。



字软元件的情况

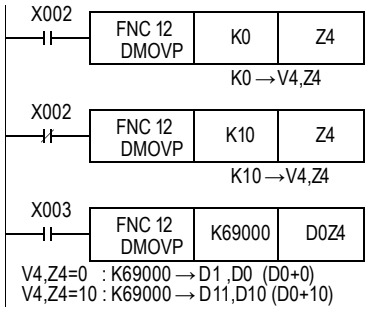
1. 16位指令的操作数修饰

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。(右图)
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。



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。(右图)
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。



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，出现运算错误。

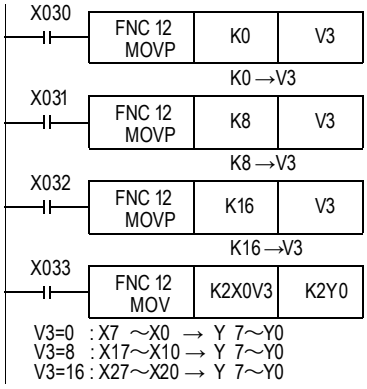
2) 对16位计数器变址修饰后，不可以作为32位的计数器使用。
如果变址修饰的结果需要是32位计数器的情况下，请对计数器C200以后的计数器加Z0～Z7。

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□\G□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)

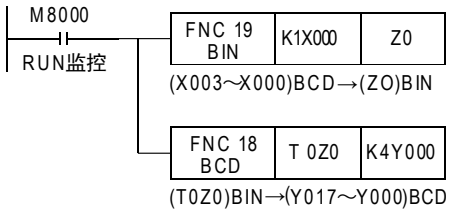
5) 位数指定的变址修饰
指定位数用的中Kn的“n”不能进行变址修饰。(K4M0Z0有效、K0Z0M0不可)

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。
在右图的例子中，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X007～X000、X017～X010、X027～X020。
通过将变址值改写成K0、K8、K16，[X000+0=X000]、[X000+8=X10]、[X000+16=X20]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。

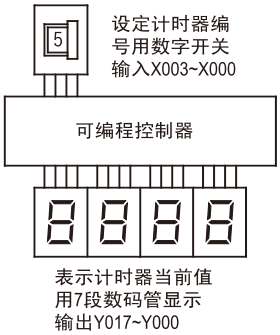


定时器当前值的显示示例

可以使用变址寄存器编写显示定时器T0～T9当前值用的顺控程序。

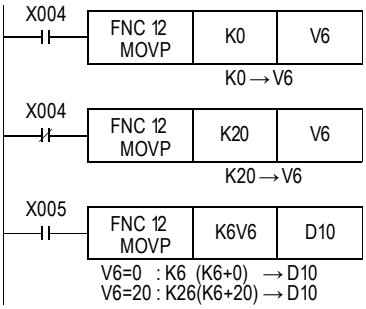


对应Z0=0～9，则T0Z0=T0～T9。



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。(右图)
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。



5.7.3 使用次数受限制的指令的变址修饰实例

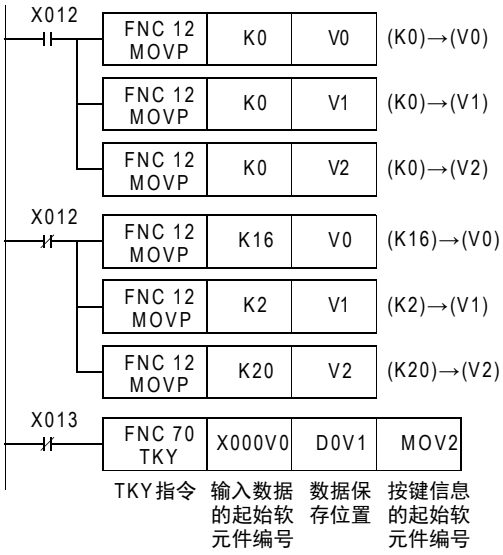
如果用变址寄存器V、Z修饰对象软元件的编号，那么可通过程序改变对象软元件的编号。如果对有使用次数限制的指令使用这一功能的话，则可以得到与多次使用该指令编程相同的效果。

使用TKY指令(FNC 70)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、D2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。



6. 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。

6.1 指令说明的阅读方法

概要

1. 指令格式

1) 表示应用指令编号(FNC No.)和指令符号。简略表示的意思如下所示。

符号	内容	相应指令(例)
	左侧上下的虚线为与16位32位指令无关的单独指令。	WDT(FNC 07)
	左侧上半段的实线表示可以使用16位指令，下半段的[D]表示可以使用32位指令。	MOV(FNC 12)
	无32位指令的指令在左侧下半段为虚线，上半段的实线表示只可以使用16位指令。	CJ(FNC 00)
	无16位指令的指令在左侧上半段为虚线，下半段的[D]表示只可以使用32位指令。	HSCS(FNC 53)
	右侧上半段的实线表示可以使用连续执行指令，下半段的[P]表示可以使用脉冲执行型的指令。	CMP(FNC 10)
	无脉冲执行方式的指令在右侧下半段为虚线，上半段的实线表示只可以使用连续执行型。	MTR(FNC 52)
	右侧上半段 表示使用了连续执行型指令后，每个运算周期目标操作数的内容都会变化的指令。 在驱动指令时只执行一次运算的情况下，使用右侧下半段中以[P]表示的脉冲执行型指令。	INC(FNC 24)

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
- BIN16位 : 16位的二进制代码
- BIN32位 : 32位的二进制代码
- BIN64位 : 64位的二进制代码
- BIN16/32位 : 16位或32位的二进制代码
- BIN32/64位 : 32位或64位的二进制代码
- BCD4位 : 4位数(16位)的BCD代码
- BCD8位 : 8位数(32位)的BCD代码
- BCD4/8位 : 4位数(16位)或8位数(32位)的BCD代码
- 字符串 : ASCII码、Shift JIS代码等的字符代码
- 字符串(仅ASCII) : 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

- X ：输入继电器(X)
- Y ：输出继电器(Y)
- M ：辅助继电器(M)
- S ：状态(S)

等等

2) 字软元件

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

等等

*1. 未指定的Kn，16位时为K1～K4，32位时为K1～K8

功能和动作说明

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。

错误

记载了使用各指令时可以考虑到的主要的错误。
有关错误的详细内容，请参考[37故障的查看方法和错误代码一览]。

程序举例

记载了使用了各指令的具体例子。

6.2 编程方面的基本注意事项

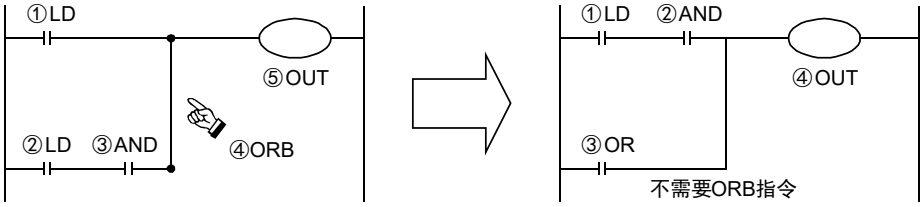
说明了在编程时需要注意的内容。

6.2.1 程序的步骤及执行顺序

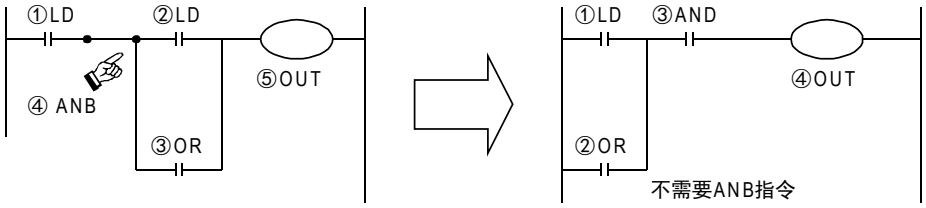
1. 触点的构成和步

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

1) 串联触点较多的梯形图写在上方比较好



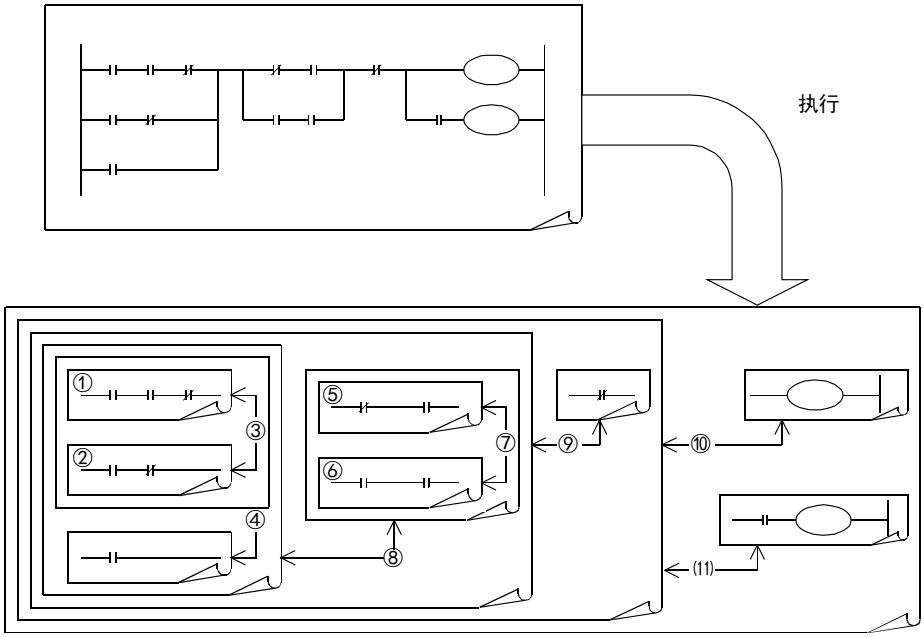
2) 并联触点较多的梯形图写在左方比较好



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。



6.2.2 双重输出(双线圈)的动作及对策

1. 双重输出的动作

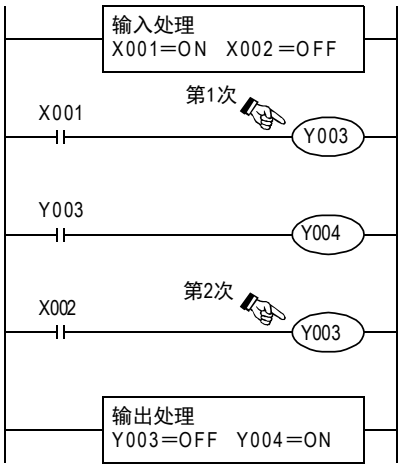
如果顺控程序中执行线圈的双重输出(双线圈)，则后侧的线圈优先动作。

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。

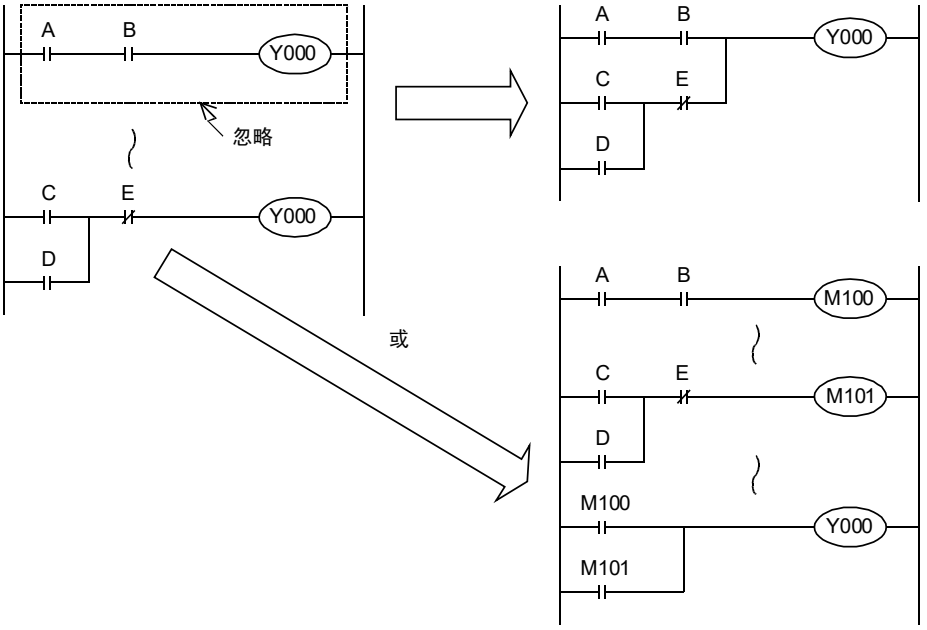
但是，第2次的Y003由于输入X002为OFF，所以其在映象存储区内被改写为OFF。

因此，实际的外部输出Y003=OFF，Y004=ON。



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(程序错误)，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。



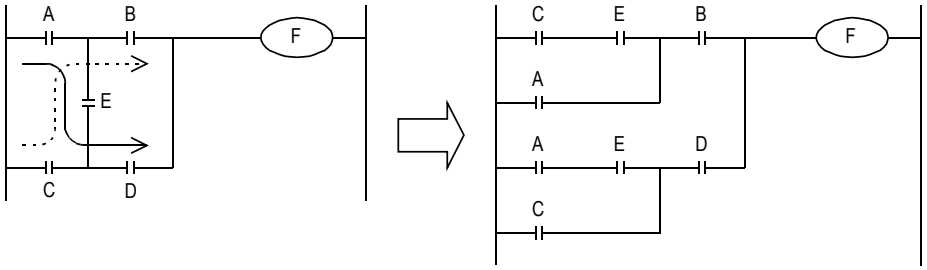
此外，还有其它编程方法，如使用SET、RST指令或跳转指令，以及使用步进梯形图指令，在各状态中对同一个输出线圈编程。

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，在状态中也存在时，会被视为双重线圈。

6.2.3 不能编程的回路图及对策

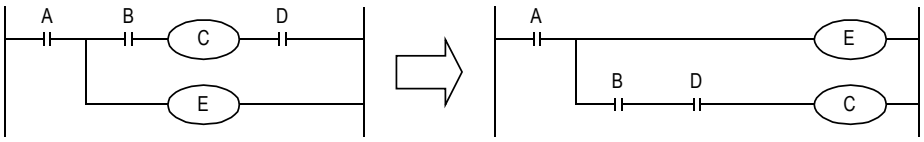
1. 桥式电路

请按照右图所示，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

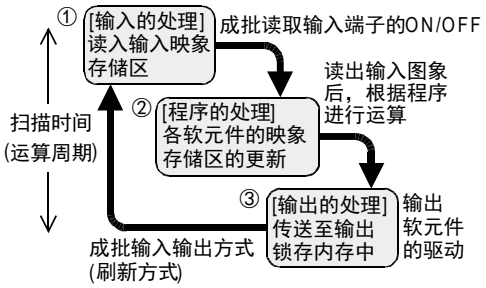
- 线圈的右侧请勿写触点。
- 建议触点之间的线圈放在前面编程。
  如触点A和B之间的线圈(E)放在程序前面，可以节省步数。



6.3 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

HC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，根据运算周期有时会出现响应延迟。



以最新信息获取最新输入输出

在上述的运算周期中，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]。

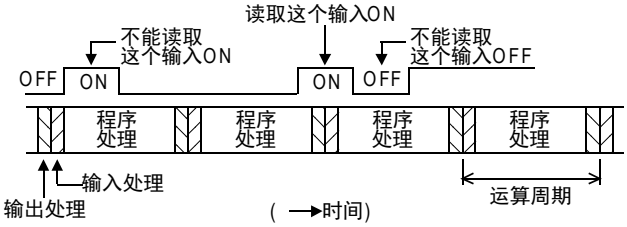
2. 不能获取宽度窄的输入脉冲

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

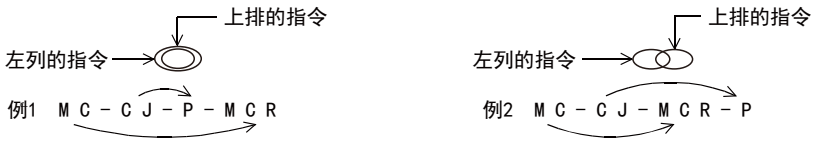
使用下面的功能，可以获取比运算周期更短的脉冲。

- 高速计数器功能
- 输入中断功能
- 脉冲捕捉功能
- 输入滤波器值的调节功能



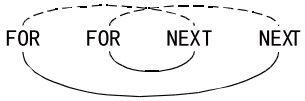
6.4 程序流程控制指令之间的相互关系

以下表示了各种程序流程控制指令的相互关联。
下表中的 ◯ 表示有包含关系的情况，◯◯ 中表示区间有前后重复的情况。



左列＼上列	MC-MCR	CJ-P	EI-DI	FOR-NEXT	STL-RET
MC-MCR	◯◯ ○ 8层	◯◯ ○ 例1	◯◯ ○	◯◯ ○	◯◯ ○
	◯◯ ×	◯◯ △ 例2	◯◯ ○	◯◯ ×	◯◯ ×
CJ-P	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○
	◯◯ △	◯◯ △	◯◯ ○	◯◯ △	◯◯ △
EI-DI	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○
	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○
FOR-NEXT	◯◯ ×	◯◯ ○	◯◯ ○	◯◯ ○ 5层	◯◯ ×
	◯◯ ×	◯◯ △	◯◯ ○	◯◯ * 2	◯◯ ×
STL-RET	◯◯ ×	◯◯ △	◯◯ ○	◯◯ ○（1个STL中）	◯◯ ×
	◯◯ ×	◯◯ △	◯◯ ○	◯◯ ×	◯◯ ×
P-SRET	◯◯ ×	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ×
	◯◯ ×	◯◯ △	◯◯ ○	◯◯ ×	◯◯ ×
I-IRET	◯◯ ×	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ×
	◯◯ ×	◯◯ △	◯◯ ○	◯◯ ×	◯◯ ×
FEND-END	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ △
	◯◯ ×	◯◯ ×	◯◯ * 1	◯◯ ×	◯◯ ×
O-FEND	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○
	◯◯ ×	◯◯ ○	◯◯ ○	◯◯ ×	◯◯ ×
O-END（无FEND）	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○	◯◯ ○
	◯◯ ×	◯◯ ×	◯◯ * 1	◯◯ ×	◯◯ ×

○：没有问题，可以使用的组合。
×：禁止使用的组合，会发生运算错误。
△：并非禁止使用，但是会使动作变得复杂，所以建议尽量不要使用的组合。

P-SRE T	I-IRET	FEND-EN D	备　注
◎ ×	◎ ×	◎ ×	＊1　含变为遗忘DI的状态。 并非错误。
◎◎×	◎◎×	◎◎×	
◎ △	◎ △	◎ ×	＊2　  按照实线所示动作。
◎◎△	◎◎△	◎◎○	
◎ ○	◎ ○	◎ ＊1	
◎◎○	◎◎○	◎◎○	＊3　最初的FEND和END有效， 并非当初想要的程序。 并非错误。
◎ ×	◎ ×	◎ ×	
◎◎×	◎◎×	◎◎×	
◎ ×	◎ ×	◎ ×	
◎◎×	◎◎×	◎◎×	
◎ ×	◎ ×	◎ ×	除去一部分的例外，可以组合有包含关系的指令。 但是，请注意以下的例外情况。
◎◎×	◎◎×	◎◎×	①　MC-MCR不能在FOR-NEXT,STL-RET,P-SRET, I-IRET中使用。
◎ ×	◎ ×	◎ ×	②　STL-RET不能在FOR-NEXT, P-SRET,I-IRET 中使用。
◎◎×	◎◎×	◎◎×	③　MC-MCR,FOR-NEXT,P-SRET, I-IRET指令之间不被I,IRET,SRET, FEND,END 等指令遮挡。
◎ ○	◎ ○	◎ ＊3	
◎◎×	◎◎×	◎ ＊3	
◎ ×	◎ ×	◎ ＊3	
◎◎×	◎◎×	◎ ＊3	
◎ ×	◎ ×	◎ ＊3	
◎◎×	◎◎×	◎ ＊3	

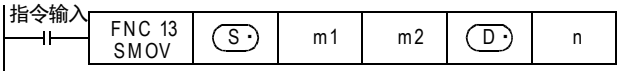
6.5 应用指令的一般通则

6.5.1 应用指令的表示和执行形式

指令和操作数

－ 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。

－ 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。
同样地，可以进行变址修饰的目标操作数有多个的时候，以(D1·)、(D2·)等表示。

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。
同样地，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。

－ 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16位指令或32位指令，会变成2或4步。

操作数的对象软元件

－ 有时会处理X、Y、M、S等的位软元件。

－ 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。

→ 请参考5.4节

－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。

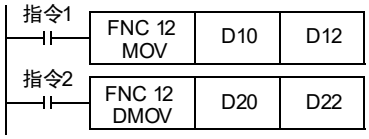
－数据寄存器D为16位，处理32位数据的时候，会组合连续2点的数据寄存器。
例如，作为32位指令的操作数，指定了数据寄存器D0的时候，就处理(D1,D0)的32位数据。
(D1为高16位，D0为低16位)
将T、C的当前值寄存器作为一般的数据寄存器使用时，处理相同。
但是，C200～C255的32位计数器处理1点的32位的数据，不能指定为16位指令的操作数。

指令形式和执行形式

根据应用指令处理的数值的大小，可以分为[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

－ 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。



将D10的内容传送到D12的指令。

将(D21,D20)的内容传送到(D23,D22)的指令。

－ 32位指令的时候，在DMOV中添加了[D]的符号来表示。

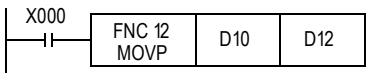
－ 指定软元件可以使用偶数或是奇数，该号码与紧接其后的软元件组合使用。(T、C、D等的字软元件的情况)
为了避免混乱，建议在32位指令的操作数中指定低位侧软元件使用偶数号码。

－ 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

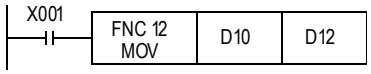
脉冲执行型

如右图所示，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。



连续执行型

右图中为连续执行型的指令，X001为ON的时候，每个运算周期都会执行。



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

关于在使用连续执行型指令时需要注意的指令，在下图所示的应用指令说明的标题部分中，加上了 ◥ 符号以示区别。



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。

6.5.2 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。

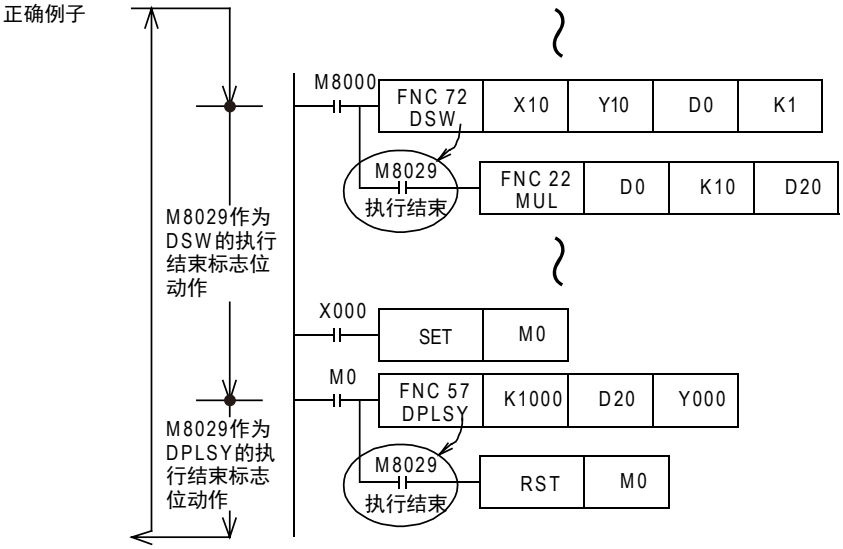
(例) M8020: 零标志位　　M8021: 借位标志位　　M8022: 进位标志位
M8029: 指令执行结束标志位　　M8090: 块比较标志位　　M8328: 指令不执行标志位
M8329: 指令执行异常结束标志位　　M8304: 零标志位　　M8306: 进位标志位

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

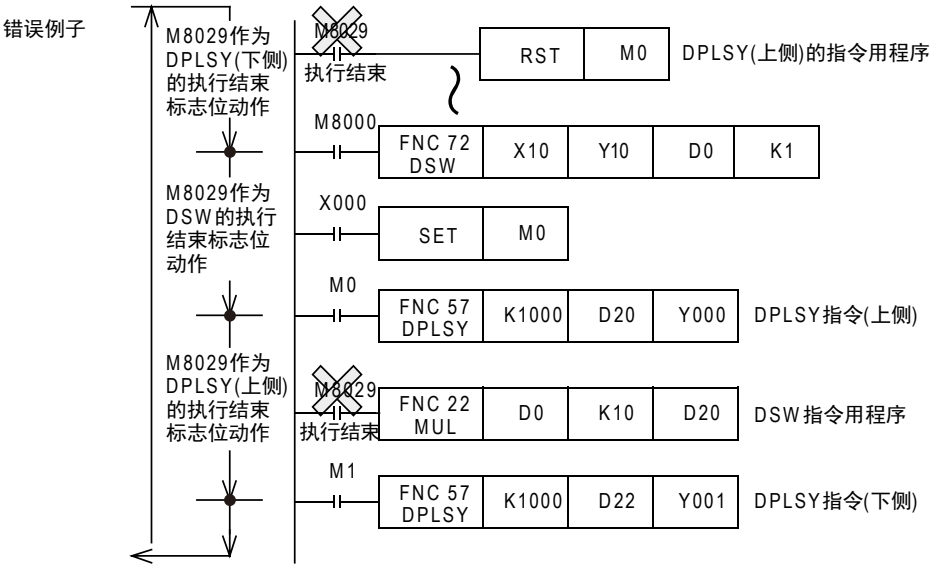
1. 多个标志位的程序(指令执行结束标志位M8029的例子)

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



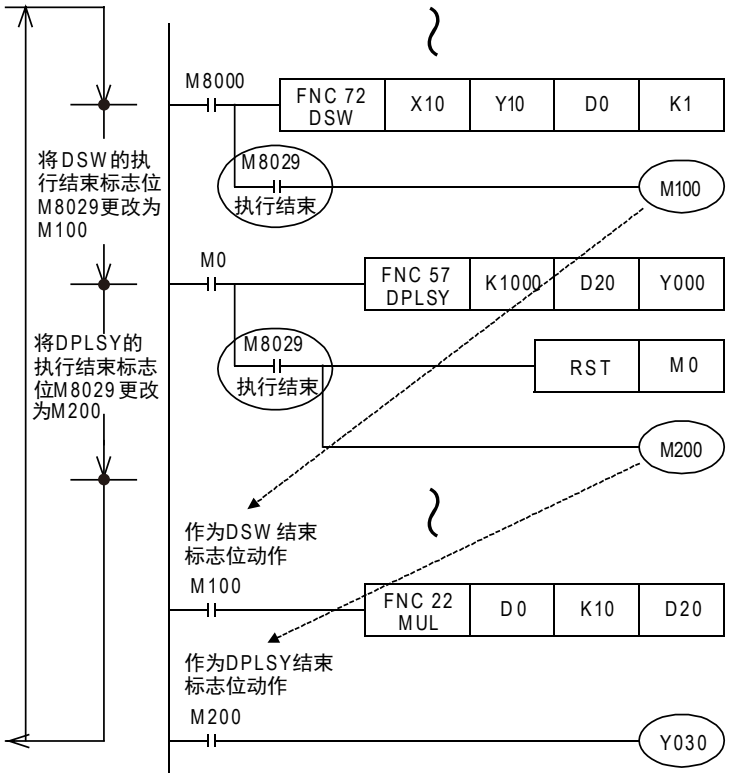
错误例子



2. 在应用指令的正下方以外的位置中使用的方法介绍

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。



6.5.3 运算错误标志位的使用

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

1. 运算错误

错误标志位	保存错误代码的软元件	保存错误步的软元件
		A8/A8C可编程控制器
M8067	D8067	D8315、D8314、D8069

- 运算错误后，M8067被置位，运算错误代码被保存到D8067中。
- A8/A8C可编程控制器的情况下，错误步被保存到 D8315、D8314(32位)中。
  发生错误的步在32767步以前的情况下，用D8069(16位)也能够确认错误步。
- 其他的步中出现新的错误时，这个指令的错误代码和步编号依次被更新。(错误解除时为OFF。)
- 可编程控制器从STOP→RUN时瞬间被清除，错误仍存在的情况下会再次置ON。

2. 运算错误锁存

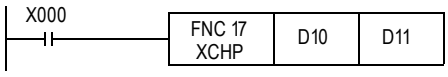
错误标志位	保存错误代码的软元件	保存错误步的软元件
		A8/A8C可编程控制器
M8068	—	D8313、D8312、D8068

- 运算错误后，M8068被置位。
- A8/A8C可编程控制器的情况下，错误步被保存到D8313、D8312(32位)中。发生错误的步在32767步以前的情况下，用D8068(16位)也能够确认错误步。
- 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。
- 发生错误的步在32767步以前的情况下，用D8068(16位)也能够确认错误步。

6.5.4 扩展功能用标志位的使用

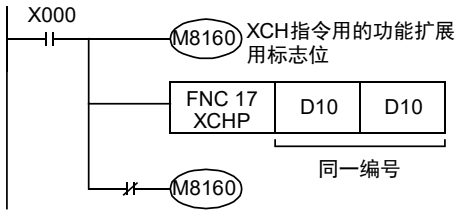
一部分指令中，可以通过与该应用指令决定的特有的辅助继电器一起使用，实现功能的扩展，以下举例说明。

－ X000为ON时，互换D10和D11的内容指令。



－ 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。

－ 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)。

6.5.5 指令使用次数的限制

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 53(HSCS)	6*1	—
FNC 54(HSCR)		
FNC 55(HSZ)		
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个输出以下)	—
FNC 280(HSCT)	1	—

*1. FNC 53(HSCS)、FNC 54(HSCR)、FNC 55(HSZ)指令的合计使用次数。

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。

→ 参考5.7.3项 使用次数受限制的指令的变址修饰实例

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[6.4 程序流程控制指令之间的相互关系]。

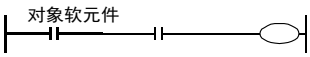
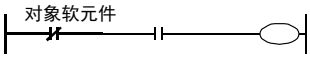
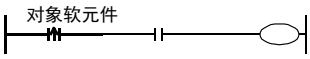
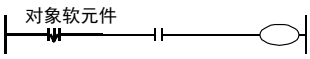
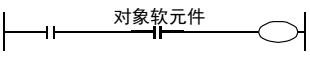
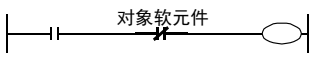
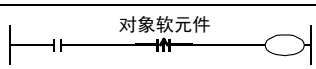
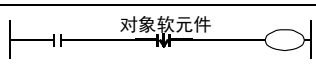
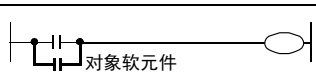
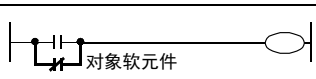
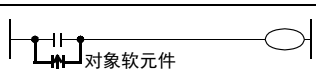
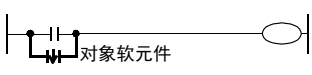
1. 定位指令
   请勿对相同的输出编号同时使用FNC 57(PLSY)、FNC 58(PWM)、FNC 59(PLSR)、FNC 150(DSZR)、FNC 151(DVIT)、FNC 156(ZRN)、FNC 157(PLSV)、FNC 158(DRVI)、FNC 159(DRVA)指令。

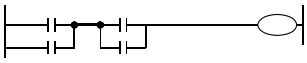
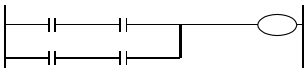
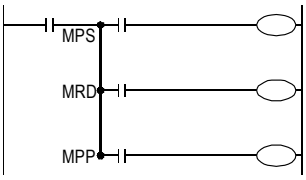
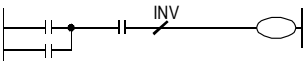
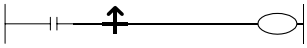
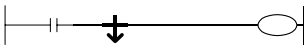
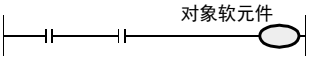
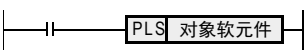
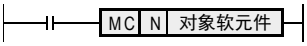
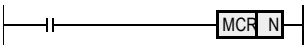
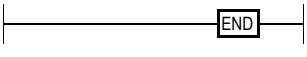
2. 高速处理指令
   - A8/A8C可编程控制器
     FNC 53(HSCS)、FNC 54(HSCR)、FNC 55(HSZ)、包含FNC 280(HSCT)指令在内同时驱动的合计点数不能超出32点。[FNC 280(HSCT)指令的使用次数仅为1次]
     但是，[FNC 280(HSCT)指令]、[FNC 55(HSZ)指令的表高速比较模式]和 [FNC 55(HSZ)指令的频率控制模式]的使用次数为各1次。

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FNC 80(RS)、FNC 87(RS2)指令。
   - FNC 80(RS)、FNC 87(RS2)、FNC 270(IVCK)、FNC 271(IVDR)、FNC 272(IVRD)、FNC 273(IVWR)、FNC 274(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FNC 270(IVCK)、FNC 271(IVDR)、FNC 272(IVRD)、FNC 273(IVWR)、FNC 274(IVBWR)指令。

7. 基本指令

在本章中，描述了基本顺控指令的种类及其功能。
对于顺序控制的初学者而言，准备了学习用教材[入门篇]、[继电器梯形图篇]以供参考。另外，准备了学习可编程控制器用的软件(初学者篇)。

记号	称呼	符号	功能	对象软元件	参考
触点指令					
LD	取		a触点的逻辑运算开始	X,Y,M,S,D □.b,T,C	7.1 节
LDI	取反		b触点的逻辑运算开始	X,Y,M,S,D □.b,T,C	7.1 节
LDP	取脉冲上升沿		检测到上升沿运算开始	X,Y,M,S,D □.b,T,C	7.5 节
LDF	取脉冲下降沿		检测到下降沿运算开始	X,Y,M,S,D □.b,T,C	7.5 节
AND	与		串联a触点	X,Y,M,S,D □.b,T,C	7.3 节
ANI	与反转		串联b触点	X,Y,M,S,D □.b,T,C	7.3 节
ANDP	与脉冲上升沿		上升沿检出的串联连接	X,Y,M,S,D □.b,T,C	7.5 节
ANDF	与脉冲下降沿		下降沿检出的串联连接	X,Y,M,S,D □.b,T,C	7.5 节
OR	或		并联a触点	X,Y,M,S,D □.b,T,C	7.4 节
ORI	或反转		并联b触点	X,Y,M,S,D □.b,T,C	7.4 节
ORP	或脉冲上升沿		上升沿检出的并联连接	X,Y,M,S,D □.b,T,C	7.5 节
ORF	或脉冲下降沿		下降沿检出的并联连接	X,Y,M,S,D □.b,T,C	7.5 节

记号	称呼	符号	功能	对象软元件	参考
结合指令					
ANB	回路块与		回路块的串联连接	—	7.7节
ORB	回路块或		回路块的并联连接	—	7.6节
MPS	存储进栈		压入堆栈	—	7.8节
MRD	存储读栈		读取堆栈		7.8节
MPP	存储出栈		弹出堆栈		7.8节
INV	反转		运算结果的反转	—	7.10节
MEP	M・E・P		上升沿时导通	—	7.11节
MEF	M・E・F		下降沿时导通	—	7.11节
输出指令					
OUT	输出		线圈驱动	Y,M,S,D □.b,T,C	7.2 节
SET	置位		动作保持	Y,M,S,D □.b	7.13节
RST	复位		解除保持的动作，清除当前值及寄存器	Y,M,S,D □.b,T,C,D,R,V,Z	7.13节
PLS	脉冲		上升沿微分输出	Y,M	7.12节
PLF	下降沿脉冲		下降沿微分输出	Y,M	7.12节
主控指令					
MC	主控		连接到公共触点	Y,M	7.9节
MCR	主控复位		解除连接到公共触点	—	7.9节
其他指令					
NOP	空操作		无处理	—	7.14节
结束指令					
END	结束		程序结束以及输入输出处理和返回0步	—	7.15节

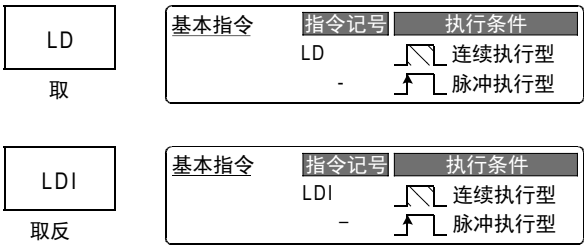
7.1 LD、LDI 指令

A8 A8C

概要

LD、LDI指令是连接在母线上的触点。
和后述的ANB指令组合后，也可用在分支起点处。

1. 指令格式



→ 有关指令步数，参考7.16节

2. 对象软元件

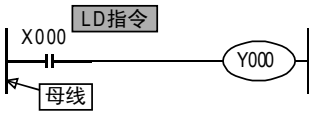
指令	位软元件							字软元件												其他					
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针	
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P	
LD	●	●	▲1	●	▲1	▲2	▲3												▲4						
LDI	●	●	▲1	●	▲1	▲2	▲3												▲4						

▲1: 对特殊辅助继电器(M)、32位计数器(C)不能进行变址修饰(V、Z)
▲2: 状态(S)不能变址修饰(V、Z)
▲3: D□.b只对应A8/A8C可编程控制器。但是，不能进行变址修饰(V、Z)。
▲4: 只对应A8/A8C可编程控制器。

功能和动作说明

1. LD指令(a触点的逻辑运算开始)

梯形图程序



指令表程序

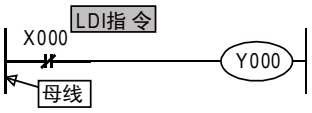
```
0000 LD   X000 ←——与母线连接
0001 OUT Y000
```

时序图



2. DI指令(b触点的逻辑运算开始)

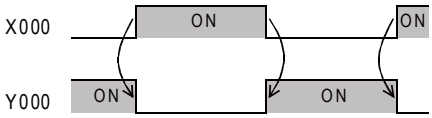
梯形图程序



指令表程序

```
0000 LDI  X000 ←——与母线连接
0001 OUT  Y000
```

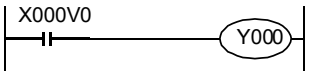
时序图



3. 变址修饰*1

LD和LDI指令中使用的软元件，可以用变址寄存器(V、Z)进行修饰。
(状态(S)、特殊辅助继电器(M)、32位计数器(C)、D□.b不能修饰。)

梯形图程序



指令表程序

```
0000  LD   X000V0
0003  OUT  Y000
```

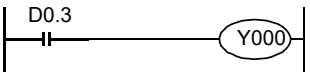
变址修饰中可以使用V0～V7、Z0～Z7。
使用的软元件为输入(X)、输出(Y)的时候，变址寄存器(V、Z)的值换算成8进制数后进行加法运算。
例: V0的值为10的时候，由X012决定LD触点ON(导通)/OFF(不导通)。

*1. 仅对应A8/A8C可编程控制器。

4. 数据寄存器(D)的位指定*1

LD和LDI指令使用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

```
0000 LD   D0.3
0003 OUT Y000
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“.”，然后接着输入位编号(0～F)。
可以使用的数据寄存器仅16位的有效。
请从低位开始按照0,1,2,…9,A,B,…F的顺序指定位编号。
例: 左边的例子中，D0的第3位决定 LD触点ON(导通)/OFF(不导通)。

*1. 仅对应A8/A8C可编程控制器。

错误

- LD或LDI指令中使用I/O编号，由于变址修饰变成实际上不存在的I/O编号的时候，M8316(I/O非实际安装的指定错误)为ON。
- 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码: 6706)。

7.2 OUT 指令

A8 A8C

概要

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

1. 指令格式





→ 有关指令步数，参考7.16节

2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
OUT		●	▲1	●	▲1	▲2	▲3												▲4					
设定值														●	●				▲4	●				

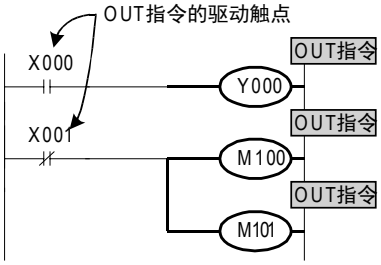
▲1： 对特殊辅助继电器(M)、32位计数器(C)不能进行变址修饰(V、Z)
▲2： 状态(S)不能变址修饰(V、Z)
▲3： D□.b仅对应A8/A8C可编程控制器。但是，不能进行变址修饰(V、Z)。
▲4： 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 使用位元软件时

用OUT指令编写的软元件，根据驱动触点的状态执行ON/OFF。
并联的OUT命令能够多次连续使用。
下面的程序举例中，接着OUT M100的OUT M101就是这个意思。
但是，对同一软元件编号，使用多个OUT指令时，会变成双重输出(双线圈)，请注意。

梯形图程序

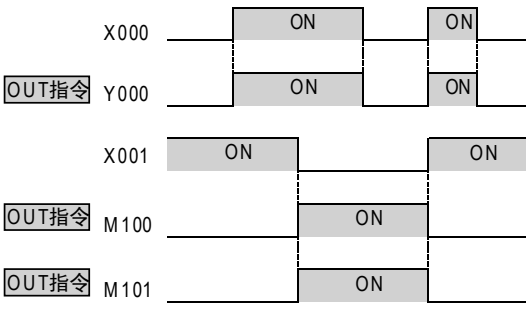


指令表程序

```
0000 LD   X000
0001 OUT Y000
0002 LDI  X001
0003 OUT M100
0004 OUT M101
```

会自动管理程序步号。

时序图

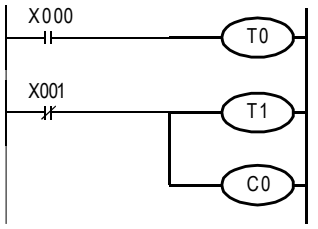


2. 使用定时器和计数器时

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

1) 直接指定

梯形图程序



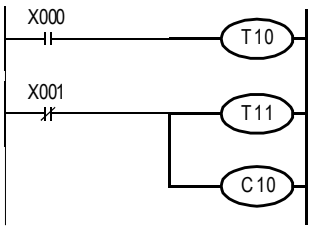
指令表程序

```
0000 LD   X000
0001 OUT T0
     (SP) K30
0004 LDI  X001
0005 OUT T1
     (SP) K30
0008 OUT C0
     (SP) K50
```

以10进制数(K)设定定时器、计数器的设定值。

2) 间接指定

梯形图程序



指令表程序

```
0000 LD   X000
0001 OUT T10
     (SP) D10
0004 LDI  X001
0005 OUT T11
     (SP) R15
0008 OUT C10
     (SP) D20
```

可以在数据寄存器(D)和扩展寄存器(R)中设定定时器和计数器的设定值。此时，数据寄存器(D)和扩展寄存器(R)的当前值即为定时器的设定值。
在驱动定时器和计数器之前，需要事先通过MOV指令或DSW指令、人机界面等将设定值写入作为设定值使用的数据寄存器(D)和扩展寄存器(R)中。

3) 定时器、计数器的设定范围

定时器和计数器的设定值的设定范围以及实际的定时器常数、OUT指令的程序步数(包含设定值)都参照下表所示。

定时器、计数器	设定范围(K的值或D、R的当前值)	实际的设定值	步数
1ms定时器	1～32,767	0.001～32.767秒	3
10ms定时器	1～32,767	0.01～327.67秒	3
100ms定时器		0.1～3,276.7秒	
16位计数器	1～32,767	同左	3
32位计数器	-2,147,483,648～+2,147,483,647	同左	5

3. 变址修饰*1

OUT指令中使用的软元件，可以用变址寄存器(V、Z)进行修饰。
(状态(S)、特殊辅助继电器(M)、32位计数器(C)、D□.b不能修饰。)

梯形图程序



指令表程序

```
0000  LD   X000
0001  OUT Y000Z0
```

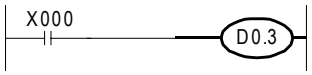
变址修饰中可以使用V0～V7、Z0～Z7。
使用的软元件为输入(X)、输出(Y)的时候，变址寄存器(V、Z)的值换算成8进制数后进行加法运算。
例如: Z0的值为20的时候，Y024 ON/OFF。

*1. 仅对应A8/A8C可编程控制器。

4. 数据寄存器(D)的位指定*1

OUT指令使用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

```
0000 LD   X000
0001 OUT D0.3
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“．”，然后紧接着输入位编号(0～F)。

可以使用的数据寄存器仅16位的有效。请从低位开始按照0,1,2,…9,A,B,…F的顺序指定位编号。

例: 左边的例子中，通过X000的ON/OFF控制D0的第3位的ON/OFF。

*1. 仅对应A8/A8C可编程控制器。

注意要点

- 使用特殊内部继电器 (M)、定时器、计数器时，程序步按照上述的“定时器、计数器的设定范围”中记载的内容增加。
- 32计数器的设定值中，请勿使用数据寄存器(D)和扩展寄存器(R)的末尾编号。

错误

- OUT指令中使用输出Y编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316 (I/O非实际安装的指定错误)为ON。
- 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码: 6706)。

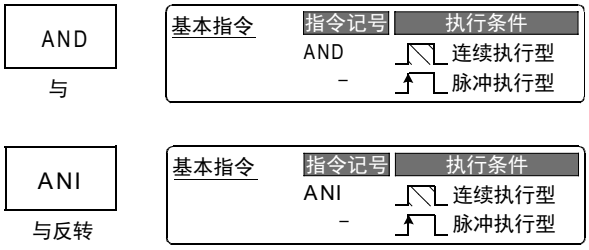
7.3 AND、ANI 指令

A8 A8C

概要

AND、ANI指令是执行串联连接1个触点。
串联触点的数量没有限制，该指令可以连续多次使用。
OUT指令后，通过触点对其他的线圈使用OUT指令，称为纵接输出。
只要顺序不错，这样的纵接输出可以重复使用多次。

1. 指令格式



→ 有关指令步数，参考7.16节

2. 对象软元件

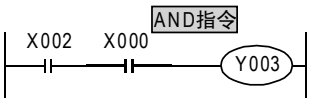
指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
AND	●	●	▲1	●	▲1	▲2	▲3												▲4					
ANI	●	●	▲1	●	▲1	▲2	▲3												▲4					

▲1: 对特殊辅助继电器(M)、32位计数器(C)不能进行变址修饰(V、Z)
▲2: 状态(S)不能变址修饰(V、Z)
▲3: D□.b仅对应A8/A8C可编程控制器。但是，不能进行变址修饰(V、Z)。
▲4: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. AND指令(串联a触点)

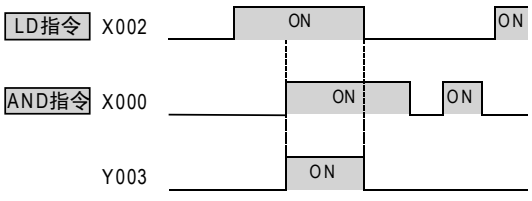
梯形图程序



指令表程序

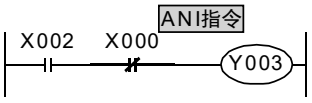
```
0000 LD   X002
0001 AND X000   ←串联触点
0002 OUT Y003
```

时序图



2. ANI指令(串联b触点)

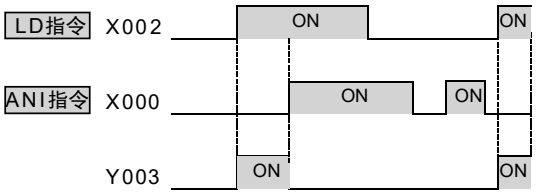
梯形图程序



指令表程序

```
0000 LD   X002
0001 ANI  X000   ←串联触点
0002 OUT  Y003
```

时序图

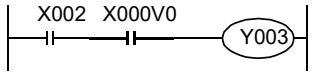


3. 变址修饰*1

AND和ANI指令中使用的软元件，可以用变址寄存器(V、Z)进行修饰。

[状态(S)、特殊辅助继电器(M)、32位计数器(C)、D□.b不能修饰。]

梯形图程序



指令表程序

```
0000 LD   X002
0001 AND  X000V0
0004 OUT  Y003
```

变址修饰中可以使用V0～V7、Z0～Z7。

使用的软元件为输入(X)、输出(Y)的时候，变址寄存器(V、Z)的值换算成8进制数后进行加法运算。

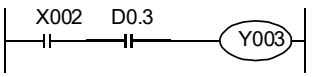
例: V0 的值为10的时候，通过X012使AND触点ON/OFF。

*1. 仅对应A8/A8C可编程控制器。

4. 数据寄存器(D)的位指定*1

AND和ANI指令使用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

```
0000 LD   X002
0001 AND  D0.3
0004 OUT  Y003
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“.”，然后紧接着输入位编号(0～F)。

可以使用的数据寄存器仅16位的有效。

请从低位开始按照0,1,2,…9,A,B,…F 的顺序指定位编号。

例: 左边的例子中，D0 的第3位为 ON 的时候，AND触点为ON(导通)。

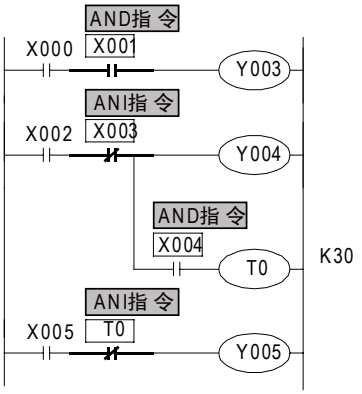
*1. 仅对应A8/A8C可编程控制器。

错误

- AND或ANI指令中使用I/O编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316(I/O非实际安装的指定错误)为ON。
- 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码: 6706)。

程序举例

梯形图程序



指令表程序

```
0000 LD   X000
0001 AND X001 ←— 串联触点
0002 OUT Y003
0003 LD   X002
0004 ANI  X003 ←— 串联触点
0005 OUT Y004
0006 AND X004 ←— 串联触点
0007 OUT T0
     (SP) K30
0010 LD   X005
0011 ANI  T0   ←— 串联触点
0012 OUT Y005
```

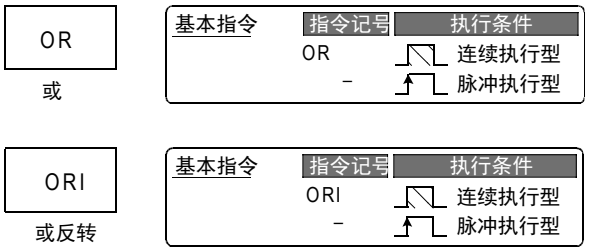
7.4 OR、ORI 指令

A8 A8C

概要

OR、ORI指令可以作为并联连接1个触点的指令使用。
串联连接了2个以上的触点时，要将这样的串联回路块与其他回路并联的时候，采用后述的ORB指令。
OR、ORI是从这个指令的步开始，与前面的LD、LDI指令的步进行并联连接。并联连接的次数不受限制。

1. 指令格式



→ 有关指令步数，参考7.16节

2. 对象软元件

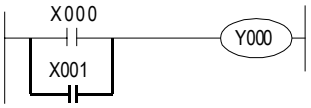
指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
OR	●	●	▲1	●	▲1	▲2	▲3												▲4					
ORI	●	●	▲1	●	▲1	▲2	▲3												▲4					

▲1: 对特殊辅助继电器(M)、32位计数器(C)不能进行变址修饰(V、Z)
▲2: 状态(S)不能变址修饰(V、Z)
▲3: D□.b仅对应A8/A8C可编程控制器。但是，不能进行变址修饰(V、Z)。
▲4: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. OR指令(并联a触点)

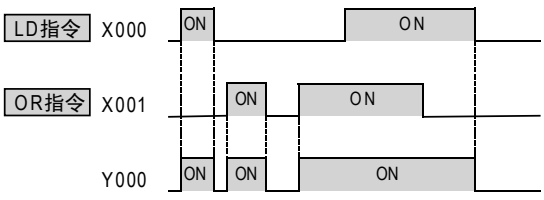
梯形图程序



指令表程序

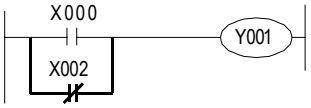
```
0000 LD   X000
0001 OR   X001
0002 OUT Y000
```

时序图



2. ORI指令(并联b触点)

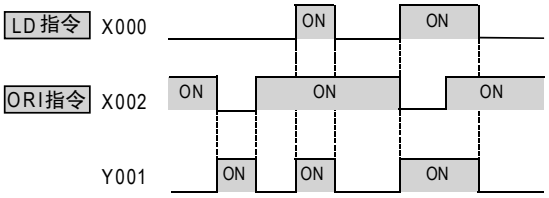
梯形图程序



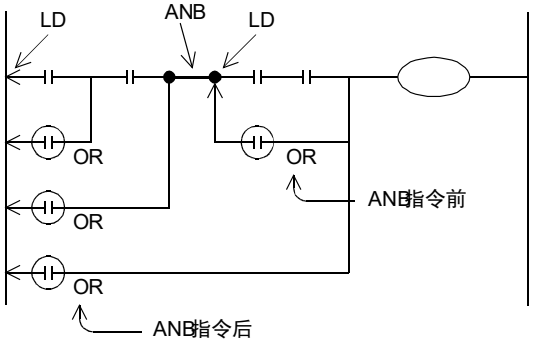
指令表程序

```
0000 LD   X000
0001 ORI  X002
0002 OUT  Y001
```

时序图



3. ANB指令的关系

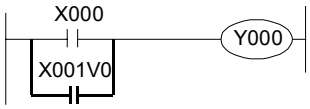


用OR、ORI指令并联连接，原则上是与前面的LD、LDI点连接，但是在后述的ANB指令后，则是与前一个的LD、LDI点连接。

4. 变址修饰*1

OR和ORI指令中使用的软元件，可以用变址寄存器(V、Z)进行修饰。
(状态(S)、特殊辅助继电器(M)、32位计数器(C)、D□.b不能修饰。)

梯形图程序



指令表程序

```
0000  LD   X000
0001  OR   X001V0
0004  OUT  Y000
```

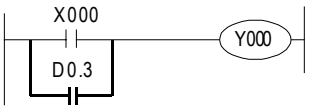
变址修饰中可以使用V0～V7、Z0～Z7。
使用的软元件为输入(X)、输出(Y)的时候，变址寄存器(V、Z)的值换算成8进制数后进行加法运算。
例: V0的值为10的时候，由X013决定OR触点ON(导通)/OFF(不导通)。

*1. 仅对应A8/A8C可编程控制器。

5. 数据寄存器(D)的位指定*1

OR和ORI指令使用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

```
0000 LD   X000
0001 OR_  D0.3
0004 OUT  Y000
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“.”，然后输入位编号(0～F)。

可以使用的数据寄存器仅16位的有效。

请从低位开始按照0,1,2,…9,A,B,…F的顺序指定位编号。

例: 左边的例子中，由D0的第3位决定OR触点的ON(导通)/OFF(不导通)。

*1. 仅对应A8/A8C可编程控制器。

错误

- OR或ORI指令中使用I/O编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316(I/O非实际安装的指定错误)为ON。
- 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码: 6706)。

7.5 LDP、LDF、ANDP、ANDF、ORP、ORF 指令

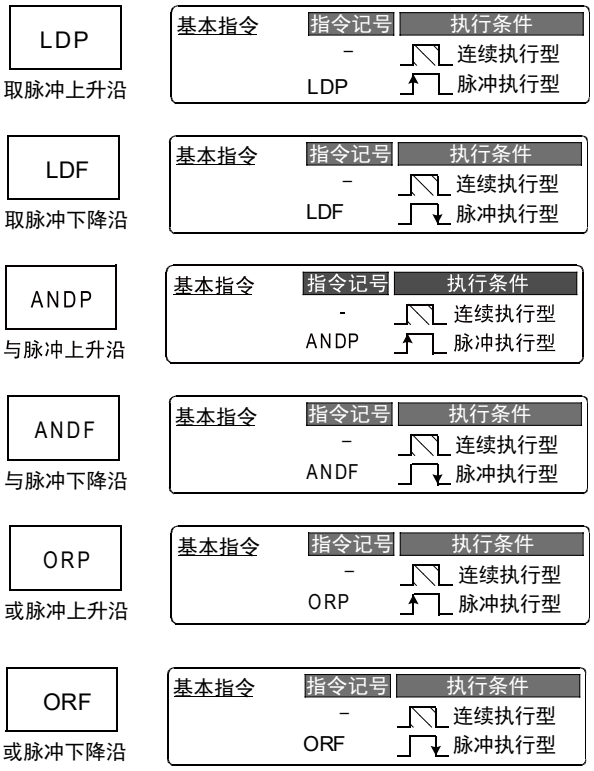
A8 A8C

概要

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

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

1. 指令格式



→ 有关指令步数，参考7.16节

2. 对象软元件

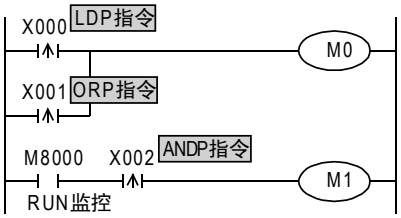
指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
LDP	●	●	●	●	●	●	▲																	
LDF	●	●	●	●	●	●	▲																	
ANDP	●	●	●	●	●	●	▲																	
ANDF	●	●	●	●	●	●	▲																	
ORP	●	●	●	●	●	●	▲																	
ORF	●	●	●	●	●	●	▲																	

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

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

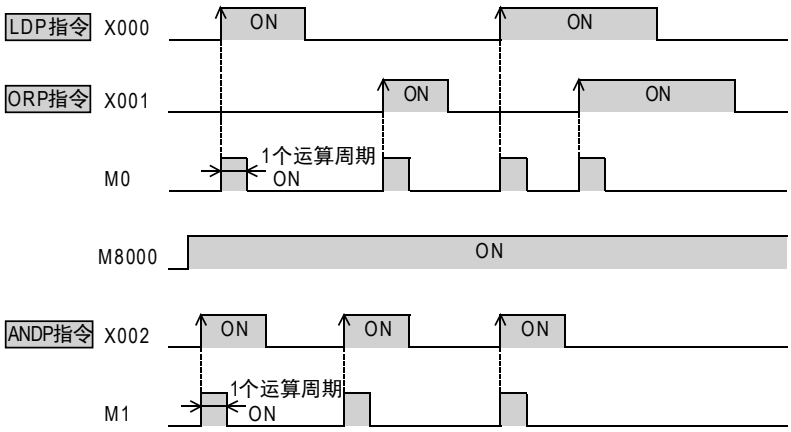
梯形图程序



指令表程序

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

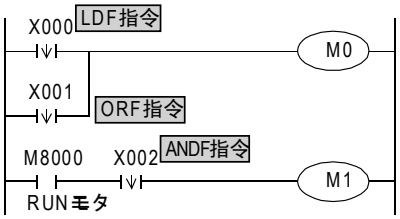
时序图



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

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

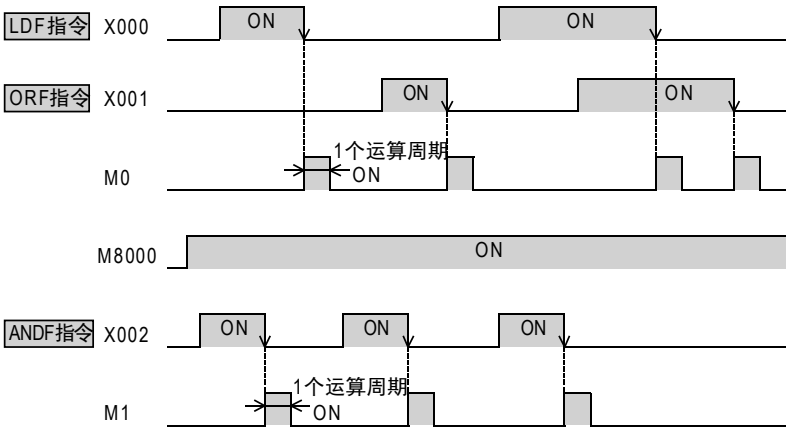
梯形图程序



指令表程序

```
0000 LDF   X000
0002 ORF   X001
0004 OUT   M0
0005 LD    M8000
0006 ANDF  X002
0008 OUT   M1
```

时序图

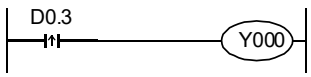


在上图中，X000～X002从ON变成OFF时，M0或M1只维持1个运算周期为ON。

3. 数据寄存器(D)的位指定*1

LDP、LDF、ANDP、ANDF、ORP、ORF指令用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

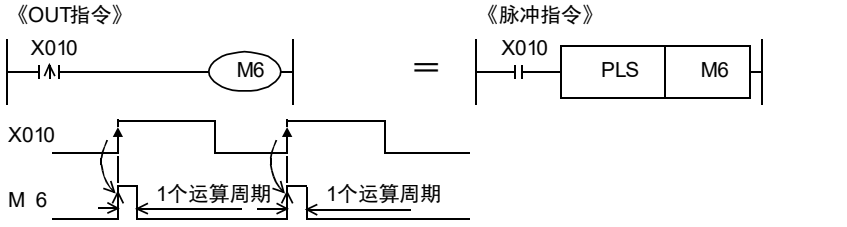
```
0000 LDP  D0.3
0003 OUT  Y000
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“.”，然后输入位编号(0～F)。
可以使用的数据寄存器仅16位的有效。
请从低位开始按照0,1,2,…9,A,B,…F的顺序指定位编号。
例: 左边的例子中，D0的第3位决定从 OFF 变 ON时，LDP触点ON(导通)/OFF(不导通)。

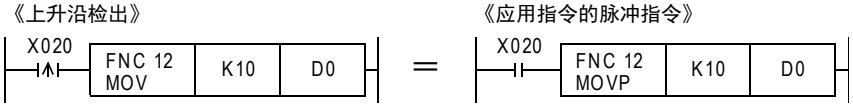
*1. 仅对应A8/A8C可编程控制器。

4. 输出驱动实例

与下面的梯形图动作相同。



无论何种情况，X010从OFF变为ON的时候，M6只维持1个运算周期为ON。

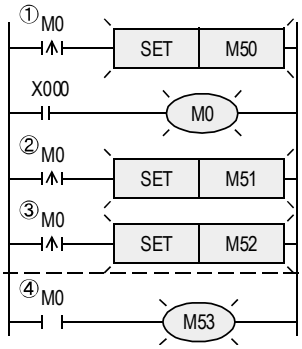


无论何种情况，X020从OFF变为ON时，只执行1次MOV指令。

5. 辅助继电器(M)编号决定的不同动作

将辅助继电器(M)指定为LDP、LDF、ANDP、ANDF、ORP、ORF指令的软元件时，根据这个软元件的编号范围，会如下图所示，出现不同的动作。

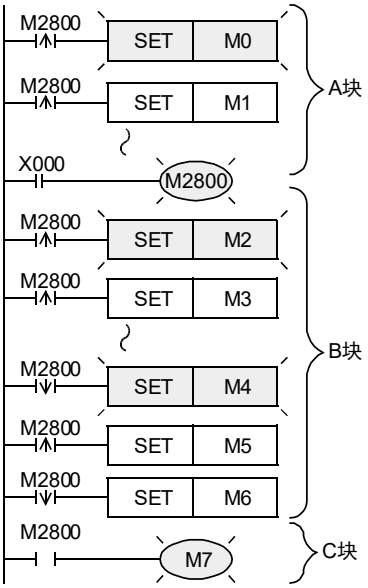
《M0～M2799,M3072～M7679》



由X000驱动M0后，M0对应的①～④的所有触点都会动作。

- ①～③，执行M0的上升沿检测。
- ④中由于有LD指令，所以M0为ON时导通。

《M2800～M3071》



以X000驱动的M2800为中心，分为上下AB块。AB各块内的上升沿或下降沿检测触点，只有最前面1个会动作。
C块的触点中由于有LD指令，所以在M2800为ON时导通。
使用这个特性，能够更加有效地对步进梯形图回路中的「采用同一信号进行状态转移」进行编程。

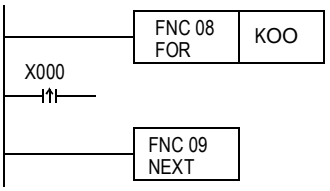
注意要点

1. 同一步中编写的LDP、LDF、ANDP、ANDF、ORP、ORF指令，在1个运算周期中被多次执行的情况下的注意事项

同一步中编写的LDP、LDF、ANDP、ANDF、ORP、ORF指令，在1个运算周期中被多次执行的时候，如下所示。

被执行多次的程序

- FOR-NEXT指令之间的程序
- 在1个运算周期中，多个CALL指令调用执行相同的子程序
- 使用CJ指令，跳转到步编号小的标签(P)的程序



动作

1) 软元件从OFF变化到ON的情况
第1次 : LDP、ANDP、ORP指令置ON。
第2次以后 : 和执行上一次指令时的软元件状态相同时，指令置OFF。

2) 软元件从ON变化到OFF的情况
第1次 : LDF、ANDF、ORF指令置ON。
第2次以后 : 和执行上一次指令时的软元件状态相同时，指令置OFF。

2. RUN中写入时的注意事项

1) 下降沿指令
包含下降沿指令(LDF指令/ANDF指令/ORF指令)的回路的RUN中写入结束时，无论下降沿指令的对象软元件是在ON/OFF的哪个状态下，都不执行指令。
此外，在下降沿指令(PLF指令)的情况下，动作条件的软元件在ON/OFF的任一状态下都不执行。
对象软元件、动作条件的软元件再次从ON到OFF的时候执行下降沿指令。

2) 上升沿指令
包含上升沿指令的梯形图在RUN中写入结束时，如果上升沿指令的对象软元件、动作条件的软元件为ON状态时则执行。
作为对象的上升沿指令: LDP、ANDP、ORP、脉冲执行型应用指令(MOVP等)

触点的ON/OFF状态 (RUN中写入时的导通状态)	上升沿指令	下降沿指令
OFF	不执行	不执行
ON	执行*1	不执行

*1. 不执行PLS指令。

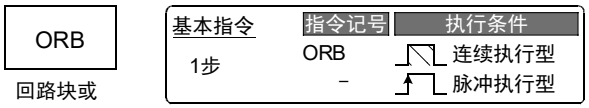
7.6 ORB 指令

A8 A8C

概要

由2个以上的触点串联连接的回路称为串联回路块。

1. 指令格式



2. 对象软元件

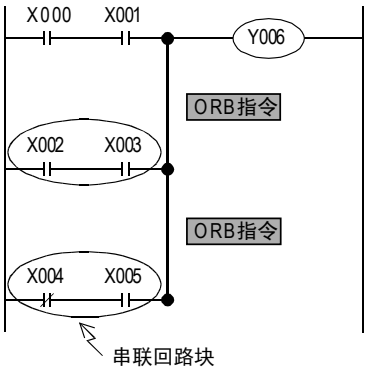
指令	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
ORB	无对象软元件																							

功能和动作说明

1. ORB指令(回路块的并联连接)

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

梯形图程序



指令表程序

理想的程序

```
0000 LD   X000
0001 AND X001
0002 LD   X002
0003 AND X003
0004 ORB
0005 LDI  X004
0006 AND X005
0007 ORB
0008 OUT Y006
```

不理想的程序

```
0000 LD   X000
0001 AND X001
0002 LD   X002
0003 AND X003
0004 LDI  X004
0005 AND X005
0006 ORB
0007 ORB
0008 OUT Y006
```

注意要点

使用ORB指令连接的并联回路数量没有限制。(上述理想程序的场合)
此外，虽然成批使用ORB指令也无妨，但是由于LD、LDI指令的重复使用次数限制在8次以下，因此请务必注意。(上述不理想程序的场合)。

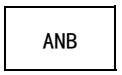
7.7 ANB 指令

A8 A8C

概要

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

1. 指令格式



回路块与

基本指令	指令记号	执行条件
1步	ANB	连续执行型
	–	脉冲执行型

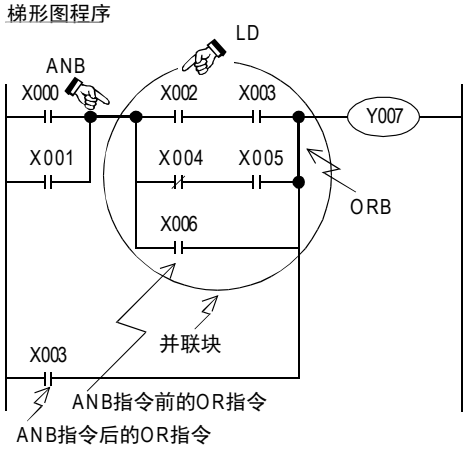
2. 对象软元件

指令	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
ANB	无对象软元件																							

功能和动作说明

1. ANB指令(回路块的串联连接)

梯形图程序



指令表程序

```
0000 LD  X000
0001 OR  X001
0002 LD  X002    <- 分支起点
0003 AND X003
0004 LDI X004    <-
0005 AND X005
0006 ORB         <- 并联块的结束
0007 OR  X006    <-
0008 ANB         <- 与前面的回路串联连接
0009 OR  X003
0010 OUT Y007
```

注意要点

ANB指令的使用次数没有限制。
此外，允许成批使用ANB指令，但是请注意和ORB指令相同，LD、LDI指令的使用次数有限制(8次以下)。

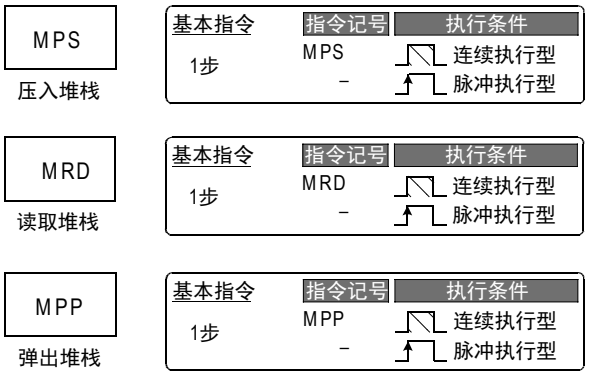
7.8 MPS、MRD、MPP 指令

A8 A8C

概要

在可编程控制器中，有11个被称为堆栈的内存，用于记忆运算的中间结果(ON或OFF)。

1. 指令格式



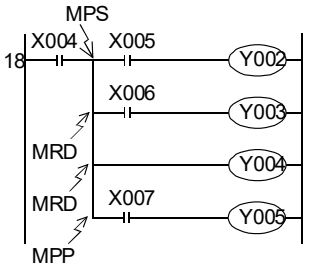
2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
MPS	无对象软元件																							
MRD	无对象软元件																							
MPP	无对象软元件																							

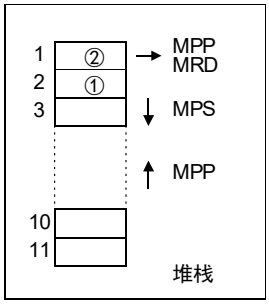
功能和动作说明

该指令是用于编写多重分支输出回路的便捷指令。

1. MPS、MRD、MPP指令(压入堆栈、读取堆栈、弹出堆栈)




```
0018 LD   X004
0019 MPS
0020 AND  X005
0021 OUT  Y002
0022 MRD
0023 AND  X006
0024 OUT  Y003
0025 MRD
0026 OUT  Y004
0027 MPP
0028 AND  X007
0029 OUT  Y005
0030 END
```



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

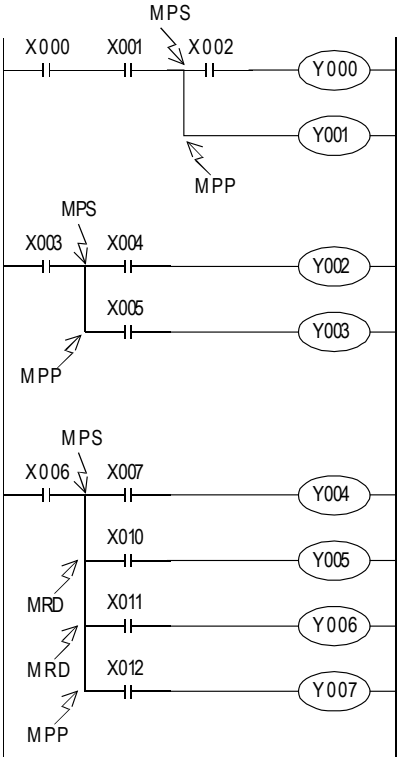
错误

MPS指令也可以重复使用，但是MPS指令和MPP指令的数量差小于11，最终两者的指令数量需要一致。

程序举例

1) 程序举例1(1段堆栈)
在这个实例中，只使用了一段堆栈。

梯形图程序

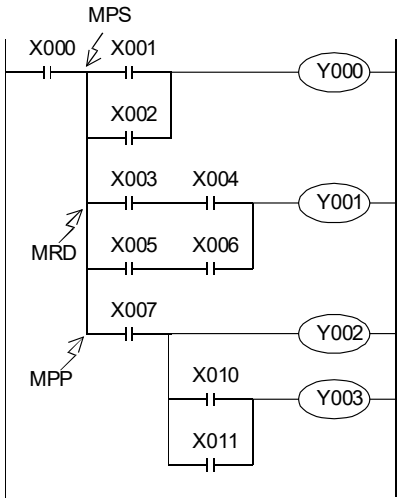


指令表程序

```
0000 LD   X000
0001 AND  X001
0002 MPS
0003 AND  X002
0004 OUT  Y000
0005 MPP
0006 OUT  Y001
0007 LD   X003
0008 MPS
0009 AND  X004
0010 OUT  Y002
0011 MPP
0012 AND  X005
0013 OUT  Y003
0014 LD   X006
0015 MPS
0016 AND  X007
0017 OUT  Y004
0018 MRD
0019 AND  X010
0020 OUT  Y005
0021 MRD
0022 AND  X011
0023 OUT  Y006
0024 MPP
0025 AND  X012
0026 OUT  Y007
```

2) 程序举例2(1段堆栈ANB、ORB指令并用)

梯形图程序

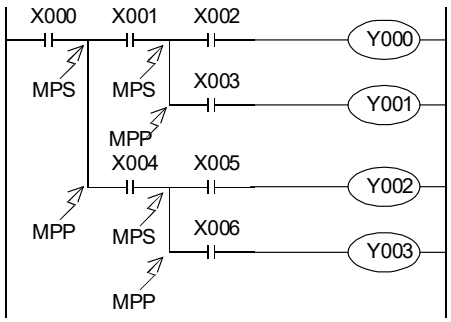


指令表程序

```
0000 LD   X000
0001 MPS
0002 LD   X001
0003 OR   X002
0004 ANB
0005 OUT  Y000
0006 MRD
0007 LD   X003
0008 AND  X004
0009 LD   X005
0010 AND  X006
0011 ORB
0012 ANB
0013 OUT  Y001
0014 MPP
0015 AND  X007
0016 OUT  Y002
0017 LD   X010
0018 OR   X011
0019 ANB
0020 OUT  Y003
```

3) 编程示例3(2段堆栈)

梯形图程序

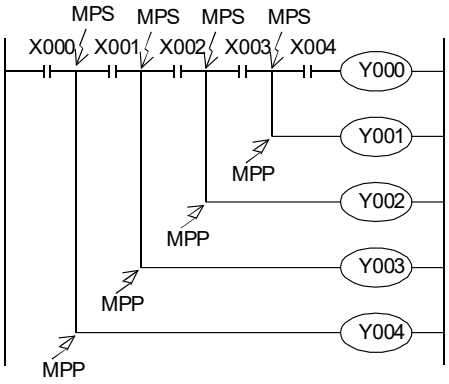


指令表程序

```
0000 LD    X000
0001 MPS
0002 AND   X001
0003 MPS
0004 AND   X002
0005 OUT   Y000
0006 MPP
0007 AND   X003
0008 OUT   Y001
0009 MPP
0010 AND   X004
0011 MPS
0012 AND   X005
0013 OUT   Y002
0014 MPP
0015 AND   X006
0016 OUT   Y003
```

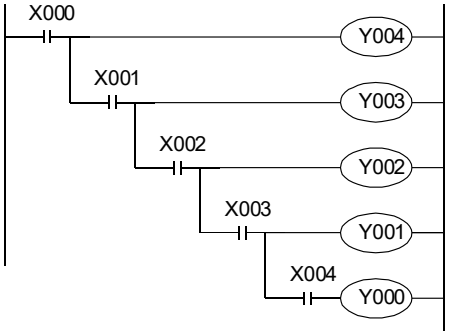
4) 编程示例4(4段堆栈)

梯形图程序



指令表程序

```
0000 LD   X000
0001 MPS
0002 AND  X001
0003 MPS
0004 AND  X002
0005 MPS
0006 AND  X003
0007 MPS
0008 AND  X004
0009 OUT  Y000
0010 MPP
0011 OUT  Y001
0012 MPP
0013 OUT  Y002
0014 MPP
0015 OUT  Y003
0016 MPP
0017 OUT  Y004
```




```
0000 LD   X000
0001 OUT  Y004
0002 AND  X001
0003 OUT  Y003
0004 AND  X002
0005 OUT  Y002
0006 AND  X003
0007 OUT  Y001
0008 AND  X004
0009 OUT  Y000
```

在图上方的梯形图中，需要使用三重的MPS指令编程。
但是，如果改用图中下方的回路，不使用MPS指令也能轻松编程。

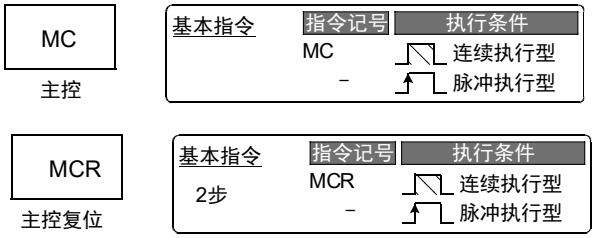
7.9 MC、MCR 指令

A8 A8C

概要

执行MC指令后，母线(LD、LDI点)移动到MC触点之后。
使用MCR指令，可以将其返回原来的母线位置。
通过更改软元件编号Y、M，可以多次使用MC指令。
但使用同一软元件编号时，和OUT指令相同，会出现双线圈输出。

1. 指令格式



→ 有关MC指令的指令步数，参考7.16节

2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
MC		●	▲																					
MCR	无对象软元件																							

▲: 特殊辅助继电器(M)除外

功能和动作说明

1. MC、MCR指令(连接到公共触点、解除连接到公共触点)

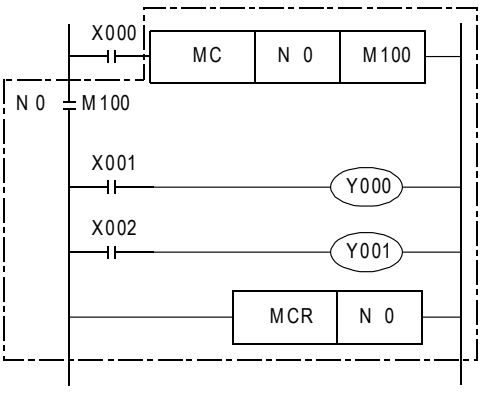
执行MC指令后，母线移动到MC触点之后。
MC触点后的母线上连接的驱动指令，只在MC指令执行时才执行各个动作，不执行MC指令时为OFF执行(与触点OFF时的动作相同)。
在如下所示的程序举例中，当输入X000为ON时，则执行从MC到MCR的指令，
但是当X000为OFF时，各个驱动软元件的动作如下。

变为OFF的软元件: 定时器(累计定时器除外)，用OUT指令驱动的软元件

保持状态的软元件: 累计定时器、计数器、用SET/RST指令驱动的软元件

此外，梯形图程序的动作说明用HCP–works的回路(读出)和回路(监控)显示。

梯形图程序



指令表程序

```
0000  LD   X000
0001  MC   N 0      ) 3步指令
      SP   M100     )
0004  LD   X001
0005  OUT  Y000
0006  LD   X002
0007  OUT  Y001
0008  MCR  N 0 ←2 步指令
```

← 请写入MCR N0指令

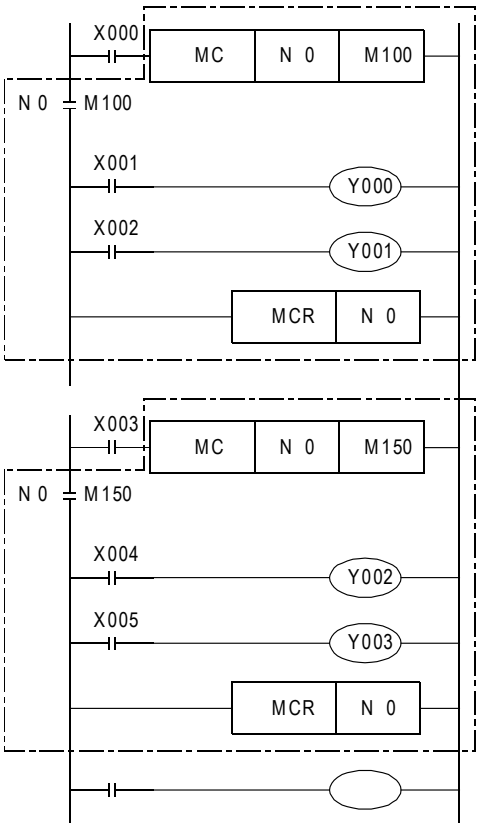
注意要点

- 在MC指令后如果没有紧跟着连接母线的指令(LD,LDI等)，会出现回路错误(错误代码：6611)。
- RUN中写入时的注意事项
  RUN中写入中，在MC～N□之间增加了指令时，无法检测错误。
  可编程控制器从STOP变为RUN后，才会报错。

编程实例

1) 无嵌套的场合

梯形图程序



指令表程序

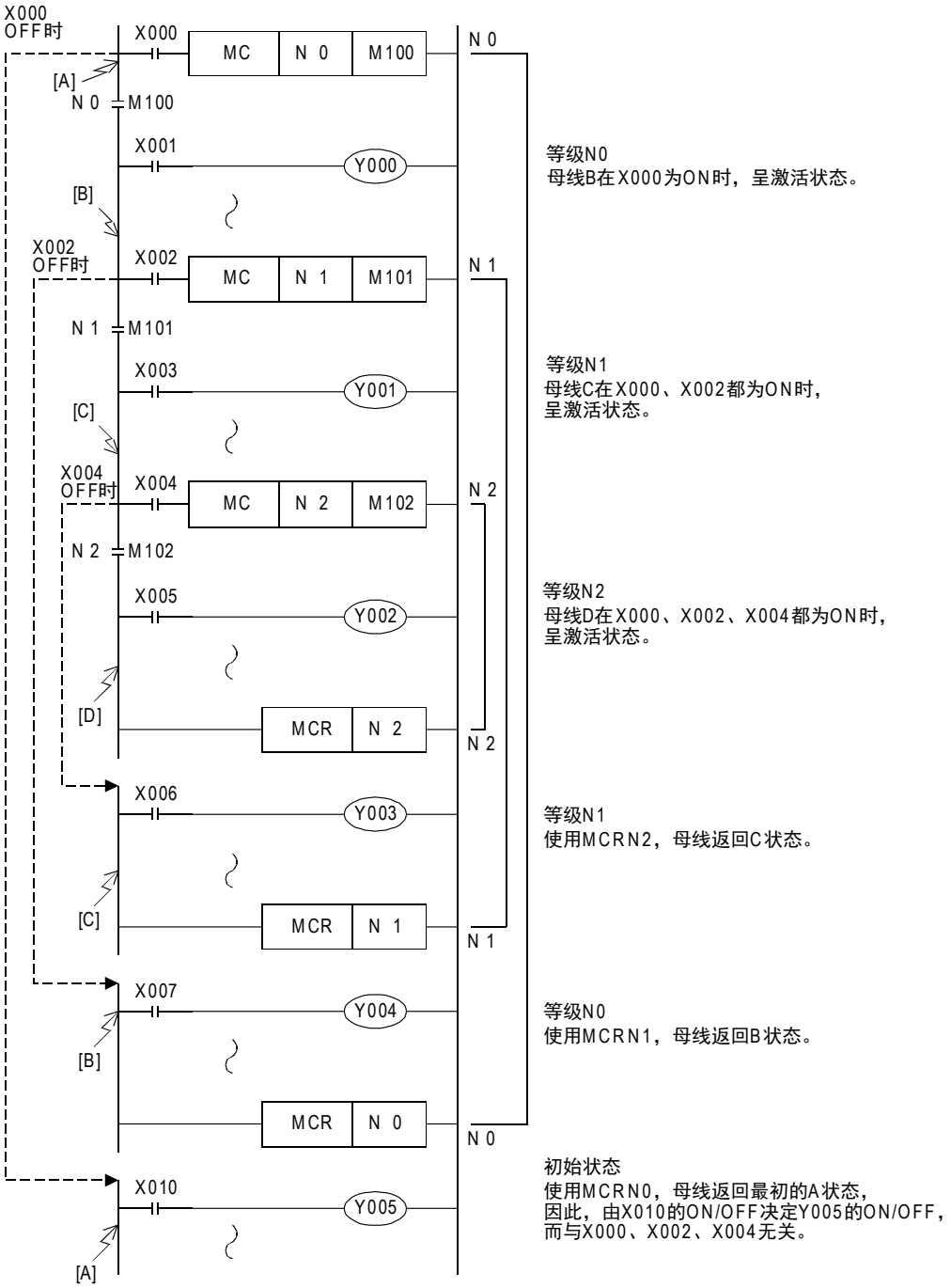
```
0000  LD   X000
0001  MC   N 0      ) 3步指令
      SP   M100     )
0004  LD   X001
0005  OUT  Y000
0006  LD   X002
0007  OUT  Y001
0008  MCR  N 0   ←2 步指令
```

←母线返回(N0为嵌套等级)

←没有嵌套结构的时候，
可以再次使用N0编程。
N0的使用次数没有限制。
在有嵌套结构时，
如一页中的例2所示，
嵌套等级N的编号从N0→N1…N6→N7，
依次增大。

2) 有嵌套的场合
使用MC指令时，嵌套等级N的编号依次增大。
(N0→N1→N2→N3→N4→N5→N6→N7)
返回时，采用MCR指令，从大的嵌套等级开始解除。
(N7→N6→N5→N4→N3→N2→N1→N0)
例如，不对MCR N6、MCR N7编程时，如对MCR N5编程，则嵌套等级会一下子返回到5。
嵌套等级最大可以编写8级(N7)。

梯形图程序



7.10 INV 指令

A8 A8C

概要

1. 指令格式

INV指令，是将INV指令执行前的运算结果反转的指令，无需指定软元件编号。



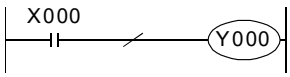
2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
INV	无对象软元件																							

功能和动作说明

1. INV指令(运算结果的反转)

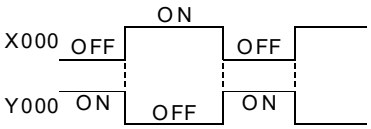
梯形图程序



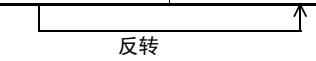
指令表程序

```
0  LD   X000
1  INV
2  OUT  Y000
```

时序图



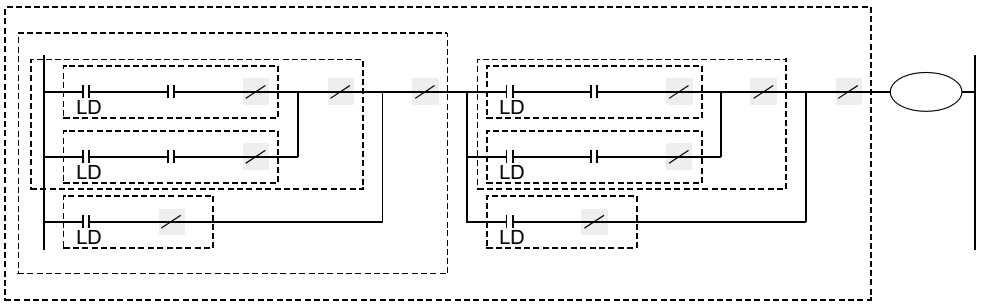
INV指令执行前的运算结果	INV指令执行后的运算结果
OFF	ON
ON	OFF



上图中，X000 为OFF 时，Y000 为ON，如果X000 为ON 时，则Y000 为OFF。
INV指令可以在与串联触点指令(AND、ANI、ANDP、ANDF指令)相同的位置处编程。
不能像指令表上的LD、LDI、LDP、LDF 那样与母线连接，也不能像OR、ORI、ORP、ORF 指令那样独立地与触点指令并联使用。

2. INV指令的动作范围

在包含ORB指令、ANB指令的复杂的回路中编写INV指令时，INV指令的动作范围如下图所示。



INV指令的功能，是将INV指令执行前存在的LD、LDI、LDP、LDF指令以后的运算结果反转。
因此，如上图所示，在ORB 指令、ANB 指令中编程时，从各自的INV 指令的位置上见到的LD、LDI、LDP、LDF以后的块作为INV运算的对象。

7.11 MEP、MEF 指令

A8 A8C

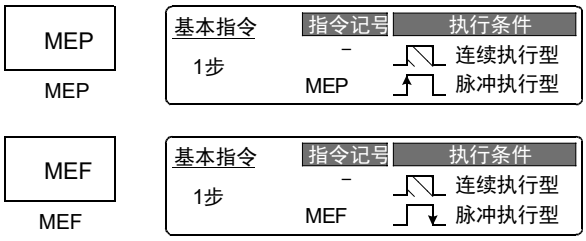
概要

MEP、MEF指令是使运算结果脉冲化的指令，不需要指定软元件编号。

1) MEP
在到MEP指令为止的运算结果，从OFF→ON时变为导通状态。
如果使用MEP指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理。

2) MEF
在到MEF指令为止的运算结果，从ON→OFF时变为导通状态。
如果使用MEF指令，那么在串联了多个触点的情况下，非常容易实现脉冲化处理。

1. 指令格式



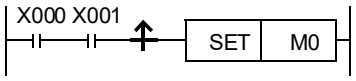
2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
MEP	无对象软元件																							
MEF	无对象软元件																							

功能和动作说明

1. MEP指令(运算结果的上升沿时为ON)

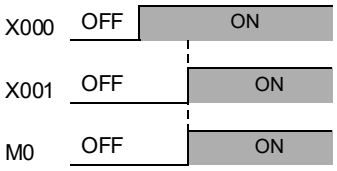
梯形图程序



指令表程序

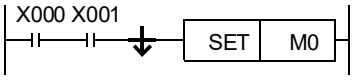
```
0  LD   X000
1  AND  X001
2  MEP
3  SET  M0
```

时序图



2. MEF指令(运算结果的下降沿时为ON)

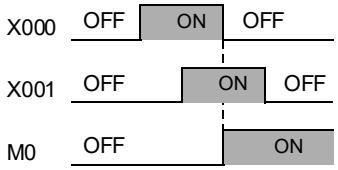
梯形图程序



指令表程序

```
0  LD   X000
1  AND  X001
2  MEF
3  SET  M0
```

时序图



注意要点

1. 在子程序以及FOR～NEXT指令等中，用MEP、MEF指令对用变址修饰的触点进行脉冲化的话，可能无法正常动作。
2. MEP、MEF指令是根据到MEP/MEF 指令正前面为止的运算结果而动作的，所以请在与AND指令相同的位置上使用。
   MEP、MEF指令不能用于LD、OR的位置。
3. RUN中写入时的注意事项
   1) 运算结果上升沿脉冲化指令(MEP指令)
      对包含MEP指令的回路的RUN中写入结束时，到MEP指令为止的运算结果为ON时，MEP指令的执行结果变为ON(导通状态)。
   2) 运算结果下降沿脉冲化指令(MEF指令)
      对包含MEF指令的回路的RUN中写入结束时，与到MEF指令为止的运算结果(ON/OFF)无关，MEF指令的执行结果变为OFF(非导通状态)。到 MEF 指令的运算结果再次从ON变为OFF时，MEF 指令的执行结果变为ON(导通状态)。

到MEP/MEF指令为止的运算结果 (RUN中写入时的导通状态)	MEP指令	MEF指令
OFF	OFF(非导通)	OFF(非导通)
ON	ON(导通)	OFF(非导通)

错误

- MEP、MEF指令不会出现运算错误。

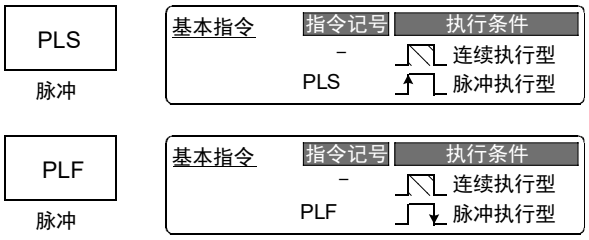
7.12 PLS、PLF 指令

A8 A8C

概要

使用PLS指令后，仅在驱动输入ON以后的1个运算周期内，对象软元件动作。
使用PLF指令后，仅在驱动输入OFF以后的1个运算周期内，对象软元件动作。
例如，驱动输入保持为ON时，可编程控制器从RUN→STOP→RUN时，PLS M0指令动作，
但是PLS M600(电池后备)不动作。(后侧的RUN时)。
这是因为STOP过程中M600仍然保持了动作状态。

1. 指令格式



→ 有关指令步数，参考7.16节

2. 对象软元件

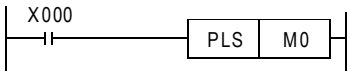
指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
PLS		●	▲1																▲2					
PLF		●	▲1																▲2					

▲1: 特殊辅助继电器(M)除外
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. PLS指令(上升沿的微分输出)

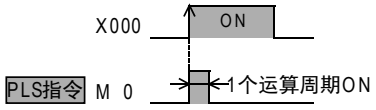
梯形图程序



指令表程序

```
0000  LD   X000
0001  PLS  M 0
```

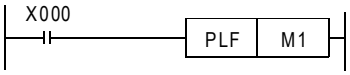
时序图



在上图中，X000从OFF变为ON时，只有一个运算周期的M1为ON。

2. PLF指令(下降沿的微分输出)

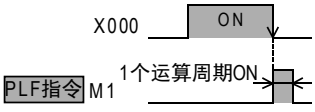
梯形图程序



指令表程序

```
0000  LD   X000
0001  PLF  M 1
```

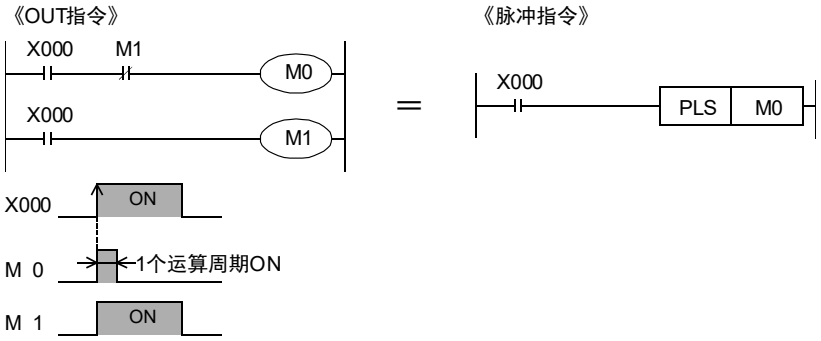
时序图



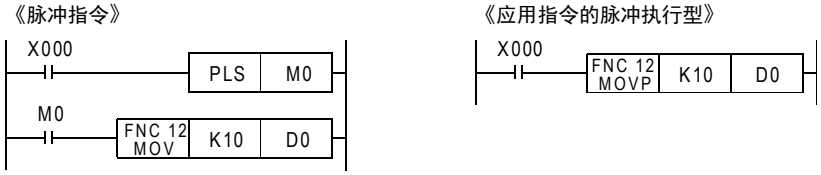
在上图中，X000从ON变为OFF时，只有一个运算周期的M1为ON。

3. 输出驱动实例

与下面的回路的动作相同。



无论哪种情况，当X000从OFF变为ON时，都只有一个运算周期的M0为ON。



无论哪种情况，当X000从OFF变为ON时，MOV指令只执行一次。

注意要点

1. RUN中写入时的注意事项

1) 下降沿指令

包含下降沿指令(LDF指令/ANDF指令/ORF指令)的回路的RUN中写入结束时，下降沿指令的对象软元件在ON/OFF任一状态下都不执行。

此外，在下降沿指令(PLF指令)的情况下，动作条件的软元件在ON/OFF的任一状态下都不执行。

对象软元件、动作条件的软元件再次从ON到OFF的时候执行下降沿指令。

2) 上升沿指令

包含上升沿指令的梯形图在RUN中写入结束时，如果上升沿指令的对象软元件、动作条件的软元件为ON状态时则执行。

作为对象的上升沿指令: LDP、ANDP、ORP、脉冲执行型应用指令(MOVP等)

触点的ON/OFF状态 (RUN中写入时的导通状态)	上升沿指令	下降沿指令
OFF	不执行	不执行
ON	执行*1	不执行

*1.不执行PLS指令。

7.13 SET、RST 指令

概要

A8 A8C

1) 位软元件的置位(SET指令[动作保持])
SET指令是当指令输入为ON时，对输出继电器(Y)、辅助继电器(M)、状态(S)以及字软元件的位指定(D□.b)ON的指令。
此外，即使指令输入为OFF，通过SET指令置ON的软元件也可以保持ON动作。

2) 位软元件的复位(RST指令[解除动作保持])
RST指令是对输出继电器(Y)、辅助继电器(M)、状态(S)、定时器(T)、计数器(C)以及字软元件的位指定(D□.b)进行复位的指令。
可以对用SET指令置ON的软元件进行复位(OFF 处理)。

3) 字软元件的当前值清除(RST指令[当前值及寄存器的清除])
RST指令是清除(T)、计数器(C)、数据寄存器(D)、扩展寄存器(R)和变址寄存器(V)、(Z)的当前值数据的指令。
此外，要将数据寄存器(D)和变址寄存器(V)、(Z)的内容清零时，也可使用RST指令。(使用常数为K0的MOV传送指令也可以得到相同效果。)
另外，使用RST指令也可以对累计定时器T246~T255的当前值和触点复位。
可以对于同一软元件，多次使用SET、RST指令，而且顺序也可随意。

1. 指令格式

SET
置位

基本指令	指令记号	执行条件
	SET	连续执行型
	–	脉冲执行型

RST
复位

基本指令	指令记号	执行条件
	RST	连续执行型
	–	脉冲执行型

→ 有关指令步数，参考7.16节

2. 对象软元件

指令	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址				常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰		K	H	E	"□"	P
SET		●	▲1			▲2	▲3												▲4						
RST		●	▲1	●	▲1	▲2	▲3					▲2	▲2	▲2	▲2		▲2	▲2	▲4						

▲1 ：不可以对特殊辅助继电器(M)、32位计数器(C)进行变址修饰(V、Z)
▲2 ：状态(S)不能变址修饰(V、Z)
▲3 ：D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲4 ：仅对应A8/A8C可编程控制器。

功能和动作说明

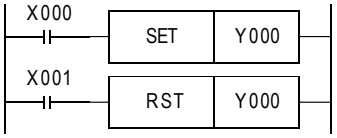
SET指令是对输出继电器(Y)、辅助继电器(M)、状态(S)以及数据寄存器(D)的指定位进行线圈驱动的指令。

1. 使用位元软件时

并联的SET指令，可以连续使用多次。

下面的程序举例中，SET Y000后接着RST Y000的程序就是这个情况。

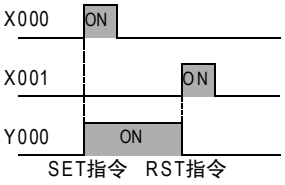
梯形图程序



指令表程序

```
0  LD   X000
1  SET  Y000
2  LD   X001
3  RST  Y000
```

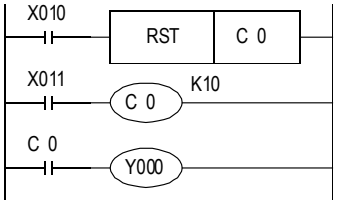
时序图



2. 使用字软元件(定时器、计数器)时

使用RST指令复位计数器和累计型定时器。

1) 内部计数器的编程



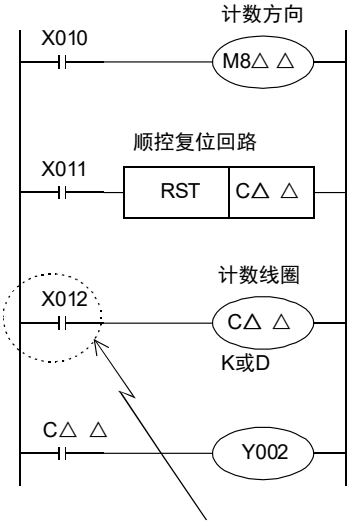
由C0对X011的OFF→ON的次数增计数，计数结果达到设定值K10的时候，输出触点C0动作。此后，即使X011从OFF变为ON，计数器的当前值也不改变，输出触点也保持动作。

为了清除这些，恢复输出触点，要使X010为ON。

OUT C指令后面，需要指定常数K或间接设定用的数据寄存器编号。

停电保持(保持)用计数器的场合，即使停电也能保持当前值以及输出触点的动作状态和复位状态。

2) 高速计数器的编程



C235～C245的单相单输入计数器中，使用特殊辅助继电器M8235～M8245来指定计数方向。

X010: ON时递减

X010: OFF时递增

X011为ON时，计数器C△△△的输出触点恢复，计数器的当前值也变为0。

对于带复位输入的计数器(C241、C242、…)，其相应的复位输入为ON时，通过中断动作可实现同上相同的运行，所以不需要为此编程。

X012为ON时，对由计数器编号决定的计数输入X000～X005的ON/OFF次数进行计数。

对于带开始输入的计数器(C244、C245、…)，其相应的开始输入若不为ON，则无法开始进行计数。

计数器的当前值增加，经过设定值(K或D的内容)的时候，输出触点被置位，在减少的方向中经过的时候被复位。

对于驱动高速计数器计数线圈用的触点，请使用高速计数执行时一直为ON的触点编程。

如果使用了分配为高速计数器用输入编号的输入继电器(X000～X005)将其作为计数线圈驱动时，不能正确计数。

用于跳转程序/子程序/中断程序时的注意事项

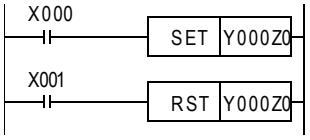
在跳转程序、子程序以及中断程序中对定时器、计数器执行RST指令时，定时器和计数器保持被复位的状态，并且定时器和计数器有可能出现不工作的情况。
有关详细内容，请参考下面内容。

→ 有关跳转的程序，参考8.1.1项
→ 有关子程序，参考8.2.1项
→ 有关中断程序，参考35.2.3项

3. 变址修饰*1

SET指令和RST指令中使用的软元件，可以采用变址寄存器(V、Z)进行修饰。
(状态(S)、特殊辅助继电器(M)、32位计数器(C)、D□.b、字软元件不能修饰。)

梯形图程序



指令表程序

```
0000 LD   X000
0001 SET Y000Z0
0004 LD   X001
0005 RST Y000Z0
```

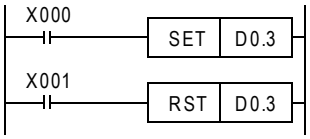
变址修饰中可以使用V0～V7、Z0～Z7。
使用的软元件为输入(X)、输出(Y)的时候，变址寄存器(V、Z)的值换算成8进制数后进行加法运算。
例: Z0的值为20的时候，Y024 ON/OFF。

*1. 仅对应A8/A8C可编程控制器。

4. 数据寄存器(D)的位指定*1

在SET指令和RST指令使用的软元件中，可以指定数据寄存器(D)的位。

梯形图程序



指令表程序

```
0000 LD   X000
0001 SET D0.3
0004 LD   X001
0005 RST D0.3
```

执行数据寄存器的位指定时，请在数据寄存器(D)的编号后输入“.”，然后输入位编号(0～F)。
可以使用的数据寄存器仅16位的有效。
请从低位开始按照0,1,2,…9,A,B,…F 的顺序指定位编号。
例: 左侧的例子中，X000 为 ON 后，D0 的第 3 位为 ON，X001为ON后，D0的第3位变OFF。

*1. 仅对应A8/A8C可编程控制器。

注意要点

在同一运算周期内，对输出继电器(Y)执行SET和RST指令时，会输出距END指令(程序的结束)近的那条指令的结果。

错误

- SET或RST指令中使用输出编号时，由于变址修饰变成实际上不存在的I/O编号时，M8316(I/O非实际安装的指定错误)为ON。
- 使用I/O编号以外的软元件(M、T、C)时，由于变址修饰变成实际上不存在的软元件编号时，出现运算错误(错误代码: 6706)。

7.14 NOP 指令

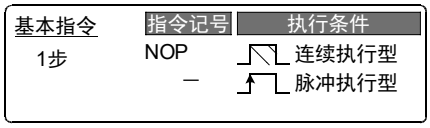
A8 A8C

概要

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

1. 指令格式





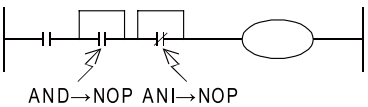
2. 对象软元件

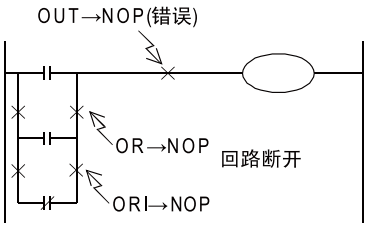
指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
NOP	无对象软元件																							

功能和动作说明

1. NOP指令(空操作)　　指令为空操作的指令。

NOP
写在程序中时，可编程控制器会无视其存在而继续运行。
更改现有的程序，改写成NOP指令时，等同于执行删除指令的操作。





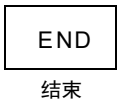
7.15 END 指令

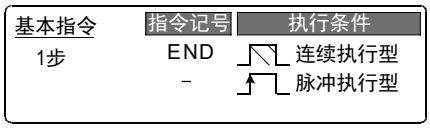
A8 A8C

概要

END指令为表示程序结束的指令。
(请勿在程序中间写入END指令。)

1. 指令格式





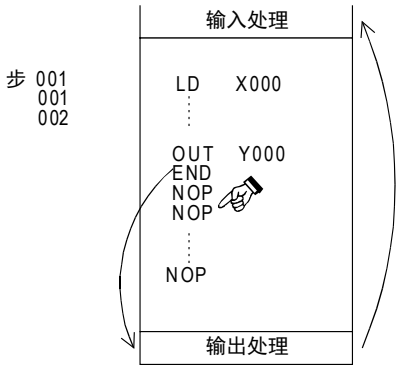
2. 对象软元件

指令	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
END	无对象软元件																							

功能和动作说明

1. END指令(程序结束以及输入输出处理和返回0步)

可编程控制器重复执行[输入处理]→[执行程序]→[输出处理]，若在程序的最后写入END指令，则不执行此后的剩余的程序步，而直接进行输出处理。
在程序的最后没有写END指令的时候，HC可编程控制器会执行到程序的最后一步，然后才执行输出处理。
此外，第一次执行开始RUN时，是从END指令开始执行的。执行END指令时，也刷新看门狗定时器(检查运算周期是否过长的定时器)。



注意要点

程序中间请勿写入END指令。
通过编程工具传送时，END指令以后都成为NOP指令(空操作)。

7.16 指令步数和指定软元件

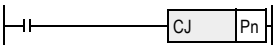
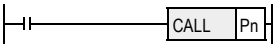
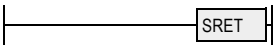
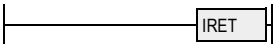
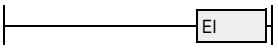
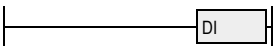
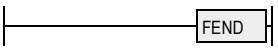
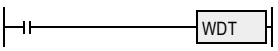
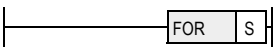
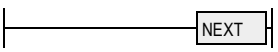
基本指令的指令步数如下所示。可用的软元件和软元件范围根据可编程控制器型号而有所不同。有关软元件的详细信息，请参考第4章。

关于ORB、ANB、MPS、MRD、MPP、MCR、INV、MEP、MEF、NOP、END指令，请参考各指令的说明页。

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

8. 程序流程—FNC 00～FNC 09

FNC 00～FNC 09中，提供了程序的条件执行及优先处理等主要与顺控程序的控制流程相关的指令。

FNC №	指令记号	符号	功能	参考
00	CJ		条件跳转	8.1节
01	CALL		子程序调用	8.2节
02	SRET		子程序返回	8.3节
03	IRET		中断返回	8.4节
04	EI		允许中断	8.5节
05	DI		禁止中断	8.6节
06	FEND		主程序结束	8.7节
07	WDT		看门狗定时器	8.8节
08	FOR		循环范围的开始	8.9节
09	NEXT		循环范围的结束	8.10节

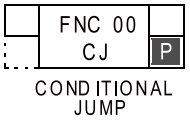
8.1 FNC 00 - CJ/条件跳转

A8 A8C

概要

使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(运算周期)和执行使用双线圈的程序。

1. 指令格式



16位指令	指令记号	执行条件
3步	CJ	连续执行型
	CJP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(Pn・)	跳转目标标记的指针编号(P) A8/A8C : n=0～4095	指针编号

3. 对象软元件

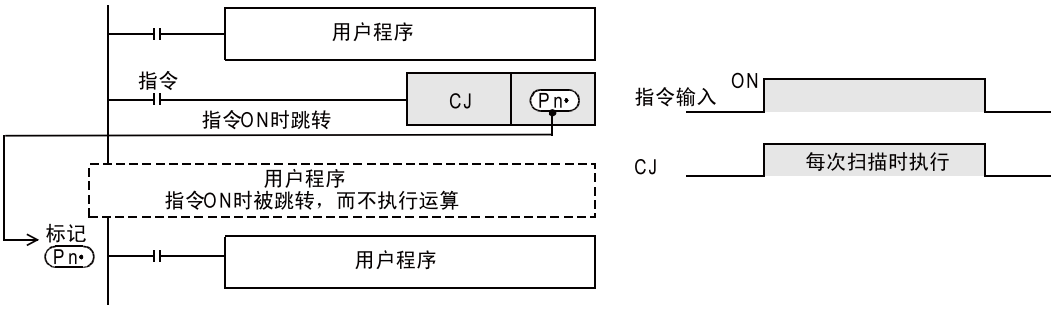
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(Pn・)																			●					●

功能和动作说明

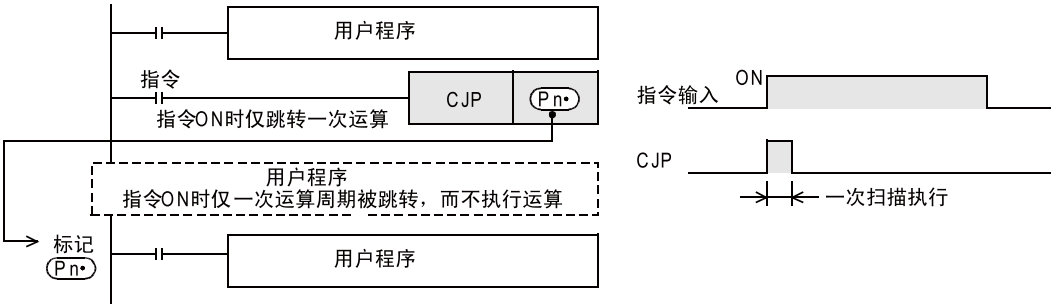
1. 16位运算(CJ、CJP)

当指令输入为ON时，执行指定标记(指针编号)的程序。

1) CJ指令时



2) CJP指令时

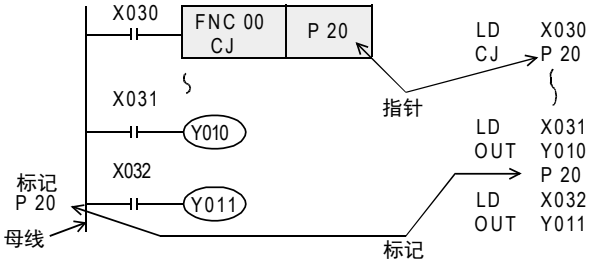


注意要点

1. 标记的输入位置与指令表程序的关系

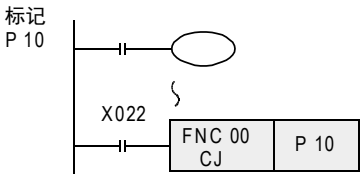
标记的编程如下所示。

编写梯形图程序时，将光标移动到梯形图的母线左侧，在回路块起始位置处输入标记(P)。



2. 对比CJ指令，步号小的步编写标记

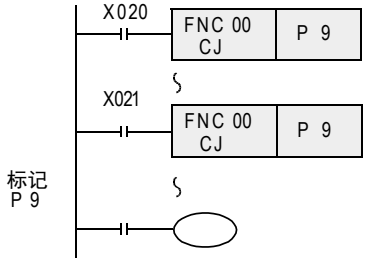
可以在比CJ指令步号小的位置中编写标记，但是扫描时间超出200ms(缺省设定)时，会发生看门狗定时器错误，请务必注意。



3. 从多个CJ指令向1个标记的跳转

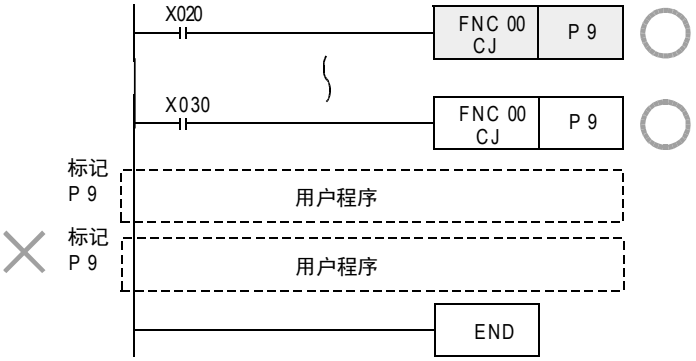
操作数中的指针编号为同一编号，标记为1个时，动作如下所示。

X020为ON时，从此处向标记P9跳转。X020为OFF，X021为ON时，从X021的CJ指令向标记P9跳转。



4. 标记(P)的重复使用

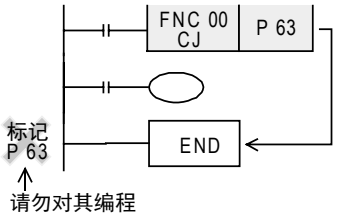
标记编号包括后述的CALL指令用的标记，如果使用了重复编号时会错误。



5. 不需要输入指针P63的标记

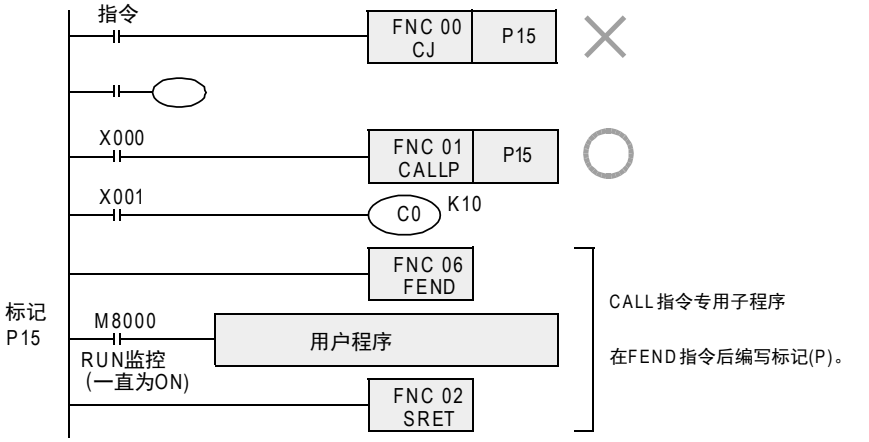
指针P63表示向END步跳转。此时，请勿对P63编程。

对标记P63编程时，可编程控制器中会显示错误代码6507(标记定义错误)并停止运行。



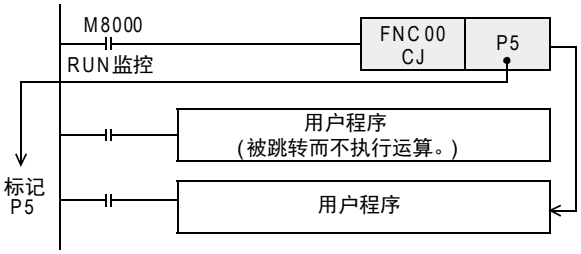
6. 向子程序的指针跳转的情况

CALL指令使用的标记和以CJ指令使用的标记不能共用。



7. 指令触点常ON时为无条件跳转

在可编程控制器运行中M8000是处于常ON的状态，所以如下所示的使用方法为无条件跳转。

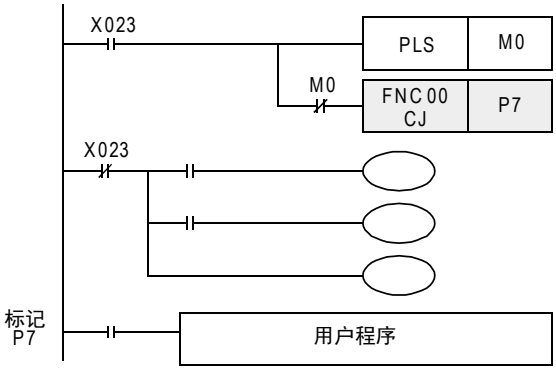


程序举例

1. OFF处理后要跳转的情况

在X023从OFF变为ON的1个运算周期后，CJ P7指令有效。

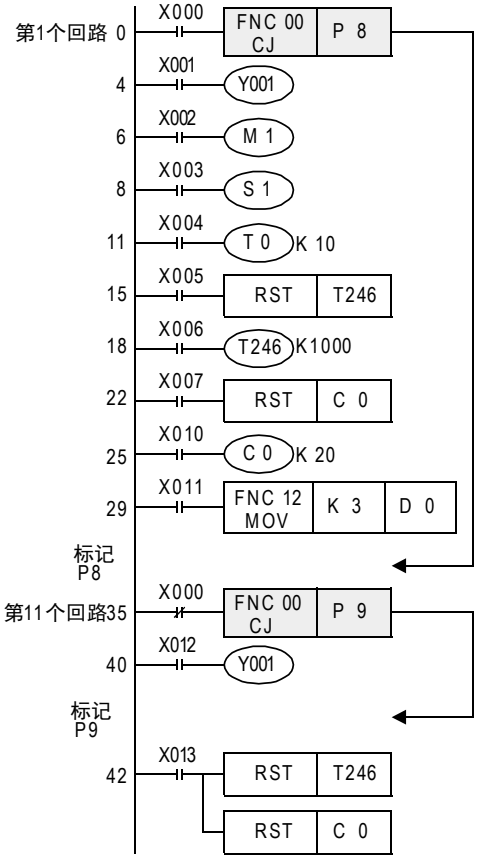
采用这个方法，可以将CJ P7指令～标记P7之间的输出全部OFF后才进行跳转。



8.1.1 CJ指令和触点线圈的动作

下面的程序举例中，当X000为ON时，从第1个回路的 CJ指令向标记P8跳转。当X000为OFF时不进行跳转，但是从1步开始按顺序执行程序，在第11个回路的CJ指令向标记P9跳转。
被跳转的指令不执行。

1. 说明动作用的梯形图实例1

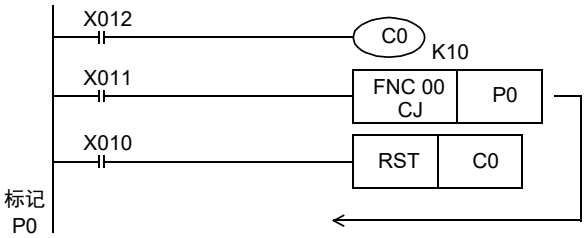


- Y001输出的双线圈动作
  X000=OFF时通过X001动作
  X000=ON时通过X012动作
  即使是因条件跳转而区分开的程序，在跳转内或是跳转外对同一线圈(Y000)编程2次以上时，则视为一般的双重线圈处理。
- 累计定时器(T246)的复位(RST)指令在跳转外时计数线圈(OUT T246) 即使被跳转，复位(触点的恢复和当前值的清除)也有效。
- 计数器(C0)的复位指令在跳转外时
  计数线圈即使被跳转，复位(触点的恢复和当前值的清除)也有效。
- 子程序定时器(T192～T199)的动作
  线圈驱动后，即使跳转，动作也持续，输出触点也动作。
- 高速计数器(C235～C255)的动作
  线圈驱动后，即使跳转，动作也持续，输出触点也动作。

上述的程序中，如跳转中各个输入变化，则各线圈的动作如下表所示。

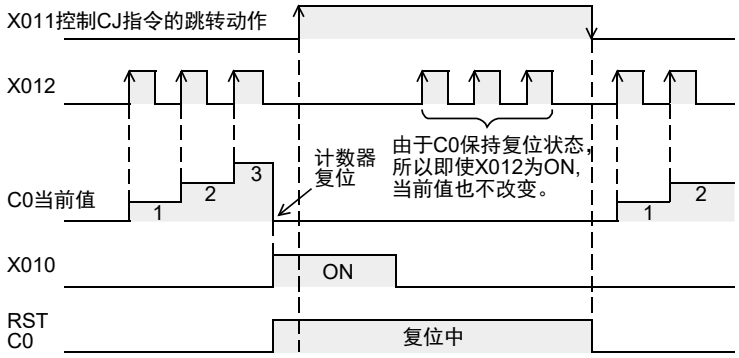
区分	跳转前的触点状态	跳转中的线圈动作
Y,M,S (Y001,M1,S1)	X001、X002、X003 OFF	Y001、M1、S1 OFF
	X001、X002、X003 ON	Y001、M1、S1 ON
10ms、100ms定时器 (T0)	X004 OFF	定时器不工作
	X004 ON	计时中断(X000 OFF后继续)
1ms定时器 (T246)	X005 OFF X006 OFF	定时器不工作 但是，X013为ON时被复位
	X005 OFF X006 ON	计时继续(X000 OFF后触点动作)
计数器 (C0)	X007 OFF X010 OFF	计数中断 但是，X013为ON时被复位
	X007 OFF X010 ON	计数中断(X000 OFF后继续)
应用命令 (MOV)	X011 OFF	跳转中不执行FNC指令。 但是，FNC 52～58的动作继续。
	X011 ON	

2. 说明动作用的梯形图实例2(仅针对定时器・计数器的RST指令被跳转的情况)

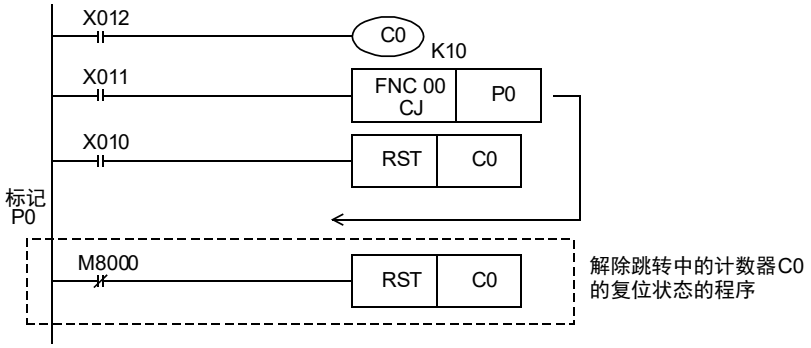


当针对计数器C0的RST指令在运行的状态下(X010为ON)时，如果 X011为ON，使用CJ(FNC 00)指令，会在已经执行RST指令的状态下进行跳转。
在这个跳转的状态中，计数器C0一直呈复位状态，因此，即使X012为ON，当前值也会保持0的状态。
想要解除该复位状态时，需要在OFF下执行针对计数器C0的RST指令。(参考下面的程序)

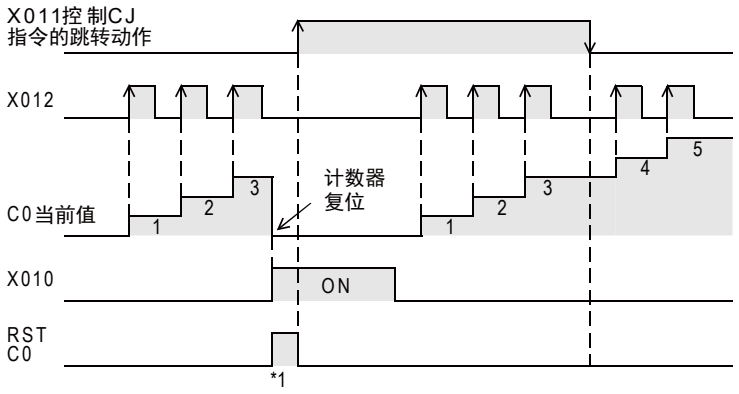
时序图



在跳转中使用定时器・计数器动作的程序举例



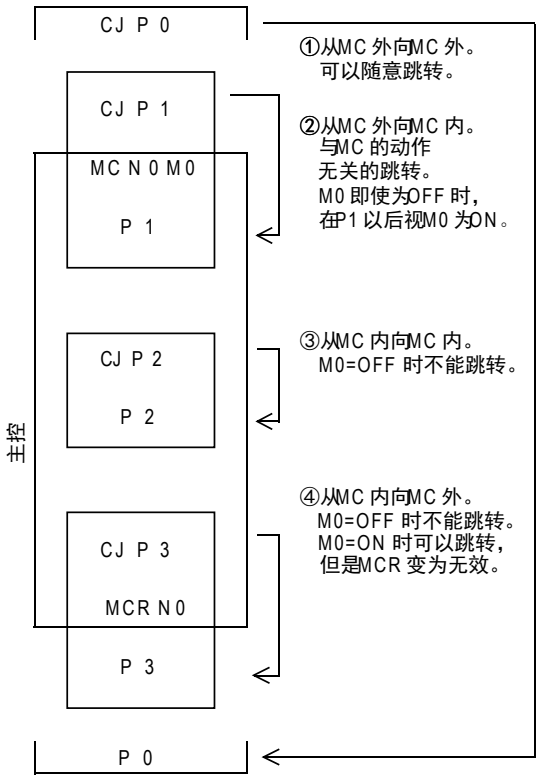
时序图

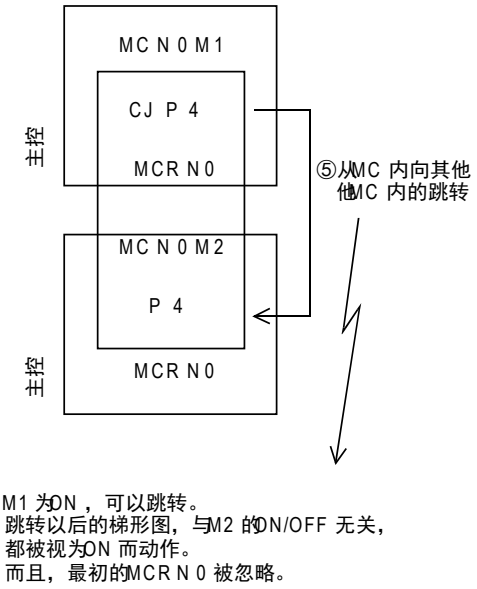


*1 在复位相同的运算周期中，计数器C0的复位状态也被解除

8.1.2 主控和跳转指令的关系

主控指令和跳转指令的关系及动作内容如下所示。
但是，由于②、④、⑤的运行会变得复杂，所以请避免使用。





M1 为ON，可以跳转。
跳转以后的梯形图，与M2 的ON/OFF 无关，
都被视为ON 而动作。
而且，最初的MCR N 0 被忽略。

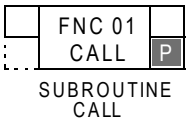
8.2 FNC 01 - CALL/子程序调用

A8 A8C

概要

在顺控程序中，对想要共同处理的程序进行调用的指令。
可以减少程序的步数，更加有效地设计程序。
此外，编写子程序时，还需要使用FEND(FNC 06)指令和SRET(FNC 02)指令。

1. 指令格式



16位指令	指令记号	执行条件
3步	CALL	连续执行型
	CALLP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(Pn•)	跳转目标标记的指针编号(P) A8/A8C : n=0～62、64～4095	指针编号

子程序调用的指针(Pn•)是A8/A8C可编程控制器时可指定P0～P62、P64～P4095的编号。
此外，由于P63为CJ(FNC 00)专用(END跳转)，所以不可以作为CALL(FNC 01)指令的指针使用。

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(Pn•)																			●					●

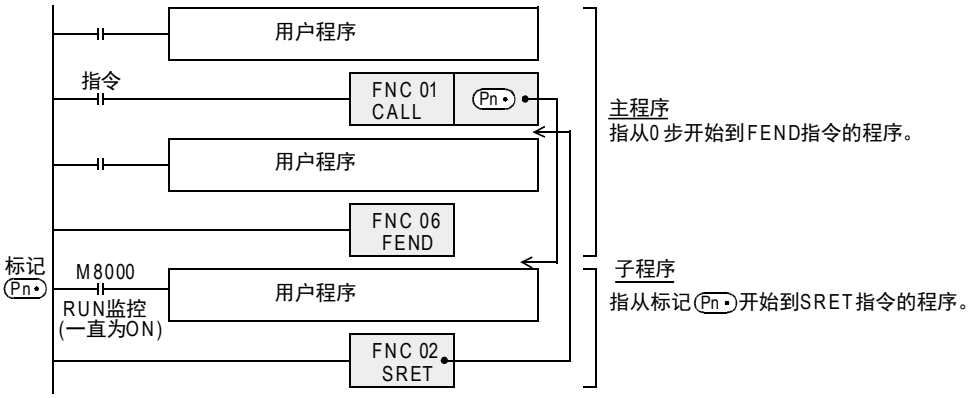
功能和动作说明

1. 16位运算

当指令输入为ON时，执行CALL指令，向标记(Pn•)的步跳转。
接着，执行标记(Pn•)的子程序。
执行SERT(FNC 02)后，返回到CALL指令的下一步。

在主程序的最后用FEND指令编程。

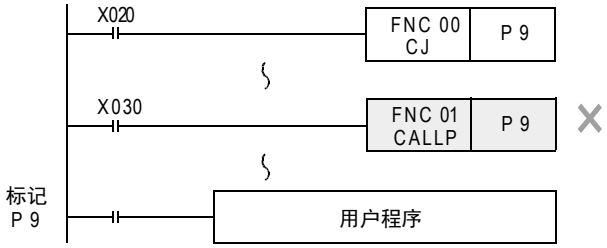
CALL指令用的标记(P)，在FEND指令后编程。



注意要点

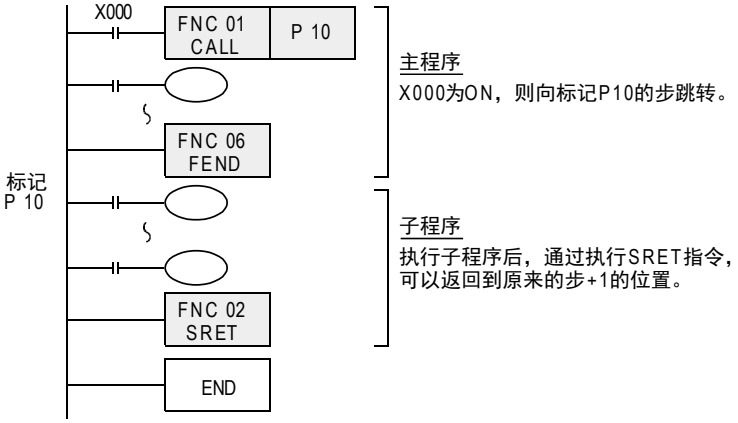
1. 标记(P)编号的重复

CALL指令中，操作数(P)的编号重复也无妨。
但是，请勿与CALL指令以外(CJ指令)的指令中使用标记(P)的编号重复。



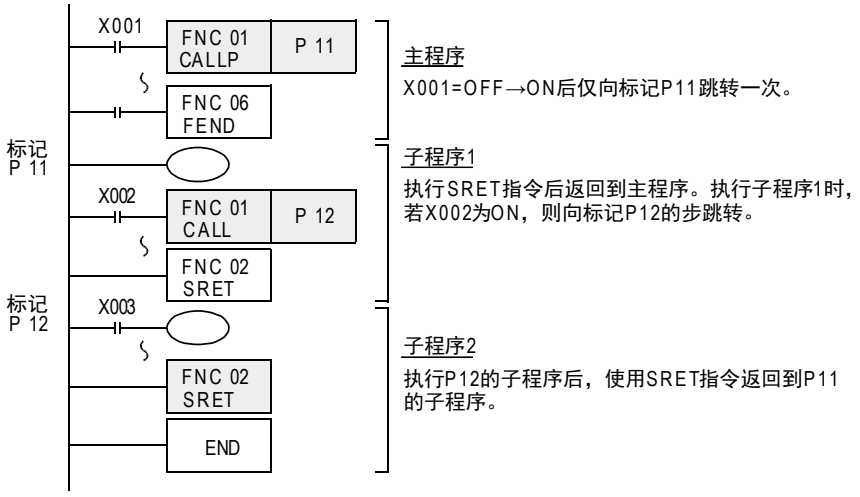
程序举例

1. 基本使用例子(1层)



2. 子程序内的多重CALL例子(多层嵌套)

子程序内的CALL指令最多允许使用4次，整体而言最多允许5层嵌套。



8.2.1 子程序和中断子程序内的注意事项

在此说明子程序和中断子程序内编程上的注意事项。
都依照在子程序内的情况加以说明，中断子程序内的情况相同。

1. 在子程序内使用定时器的情况(中断子程序内相同)

请在子程序内使用T192～T199的累计定时器。
该定时器在执行线圈指令时，或是执行END指令时进行计时。
如果达到定时器设定值，在执行线圈指令时，或是执行END指令时输出触点动作。
由于一般的定时器只在执行线圈指令时进行计时，因此，如果用于仅在某种条件下才执行线圈指令的子程序内时，则不能进行计时。

2. 使用1ms累计定时器时的注意事项(中断子程序内相同)

在子程序内使用1ms累计定时器时，当达到设定值后，输出触点会在最初执行线圈指令时(执行子程序时)动作，请务必注意。

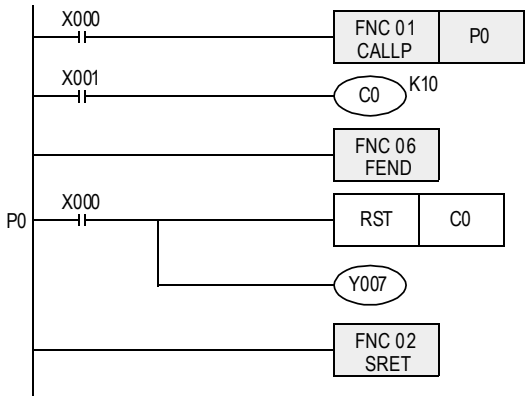
3. 子程序内使用的软元件的保持对策(中断子程序内相同)

在子程序内被置ON的软元件，在程序结束后也被保持。(参考下面程序)
此外，对定时器和计数器执行RST指令后，定时器和计数器的复位状态也被保持。
因此，对这些软元件编程时，或是编写成在程序结束后的主程序中进行复位，或是编写成在子程序中执行复位和执行OFF的程序。(参考下页中的程序)

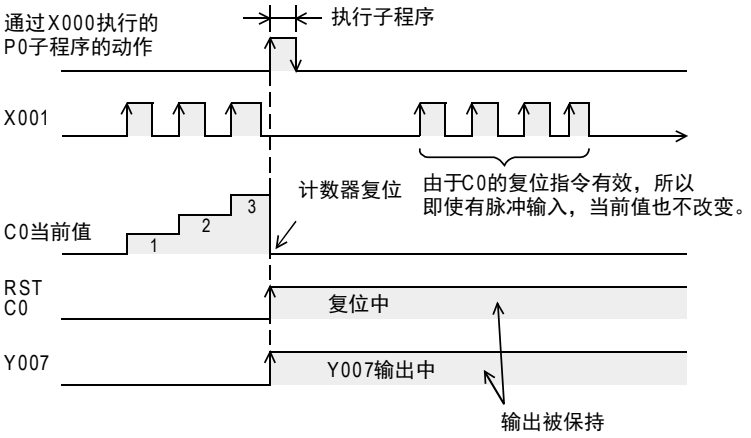
输出被保持的例子

有C0对X001计数，输入X000后子程序P0仅执行1个扫描，计数器复位及输出Y007的程序。

1) 程序举例

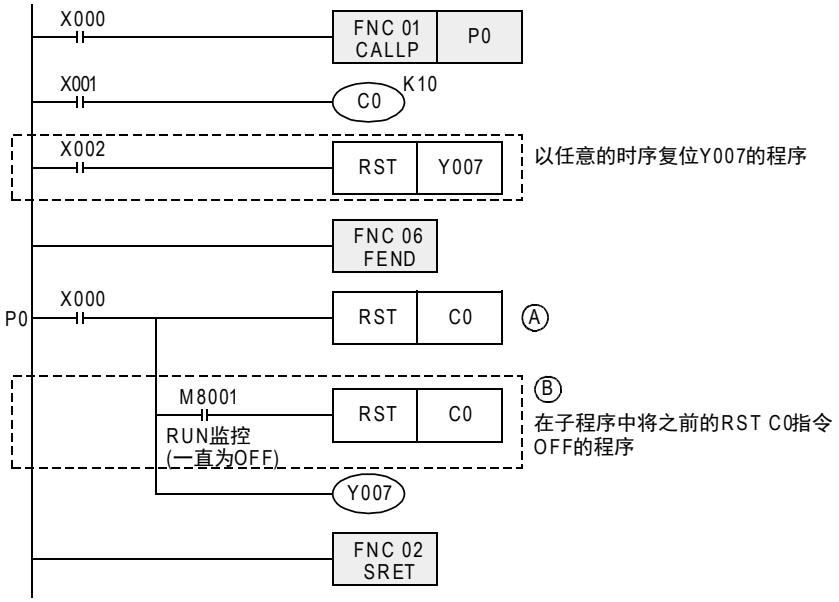


2) 时序图

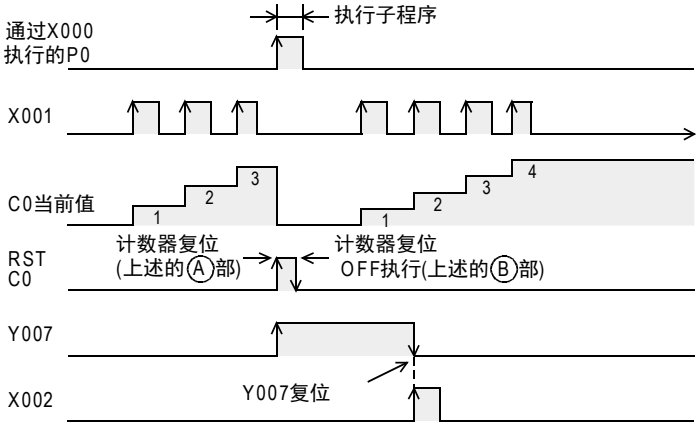


复位被保持的输出的例子(对策)

1) 程序举例



2) 时序图



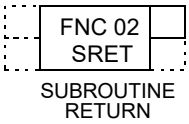
8.3 FNC 02 - SRET/子程序返回

A8 A8C

概要

从子程序返回到主程序的指令。

1. 指令格式



独立指令	指令记号	执行条件	
1步	SRET	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

执行了主程序中的CALL指令后，跳转到子程序，然后使用SRET指令返回到主程序。

→ 参考上述的8.2节

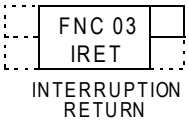
8.4 FNC 03 - IRET/中断返回

A8 A8C

概要

从中断子程序返回到主程序的指令。

1. 指令格式



独立指令	指令记号	执行条件	
1步	IRET	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

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

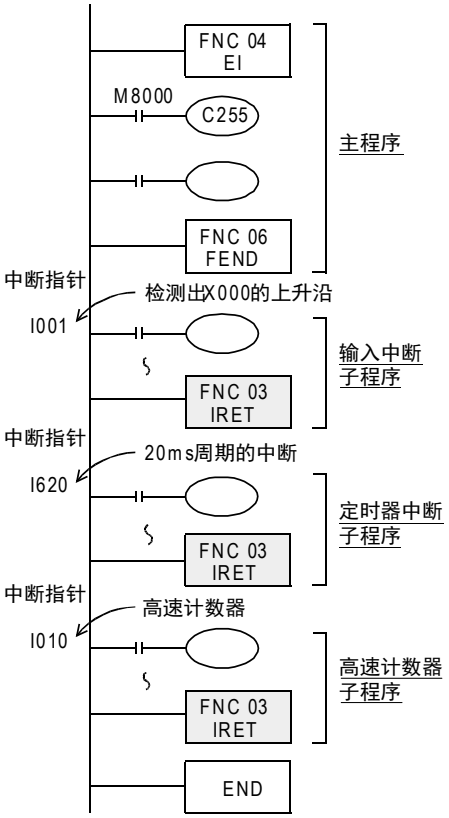
1. 中断功能的种类

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。	35.3节，35.4节
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。	35.5节
计数器中断*1	I010～I060	高速计数器增计数时执行中断处理。	35.6节

*1. 仅对应A8/A8C可编程控制器。

→ 有关中断功能，参考35章

程序举例



可编程控制器通常为禁止中断状态。
使用EI指令允许中断。
在主程序处理过程中如果X000为ON，则中断子程序I001指针以后的指令被执行，通过IRET指令返回到原来的主程序中。

每间隔定时器时间20ms会执行I620的定时器中断，每次使用IRET指令返回到主程序中。

当高速计数器的当前值与DHSCS(FNC 53)指令指定的值一致的时候，执行I010的高速计数器中断，然后使用IRET指令返回到主程序。

中断用指针(I***)必须在FEND指令后面作为标记编程。

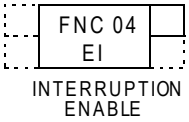
8.5 FNC 04 - EI/允许中断

A1 A2 A2C A8 A8C

概要

可编程控制器通常为禁止中断状态。使用这个指令，可以使可编程控制器变为允许中断的状态。
使用输入中断和定时器中断，计数器中断功能的时候，请使用该指令。

1. 指令格式



独立指令	指令记号	执行条件	
1步	EI	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

EI指令是不需要指令(驱动)触点的独立指令。

→ 有关中断功能，参考35章

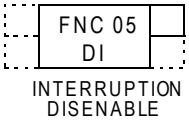
8.6 FNC 05 - DI/禁止中断

A1 A2 A2C A8 A8C

概要

在改为允许中断后，使用EI(FNC 04)指令可以再次更改为禁止中断。

1. 指令格式



独立指令	指令记号	执行条件	
1步	DI	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

DI指令是不需要指令(驱动)触点的独立指令。

→ 有关中断功能，参考35章

注意要点

DI指令以后产生的中断(要求)，在执行了EI(FNC 04)指令后方可处理。

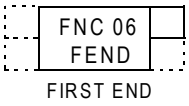
8.7 FNC 06 - FEND/主程序结束

A1 A2 A2C A8 A8C

概要

表示主程序结束的指令。

1. 指令格式



独立指令	指令记号	执行条件	
1步	FEND	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

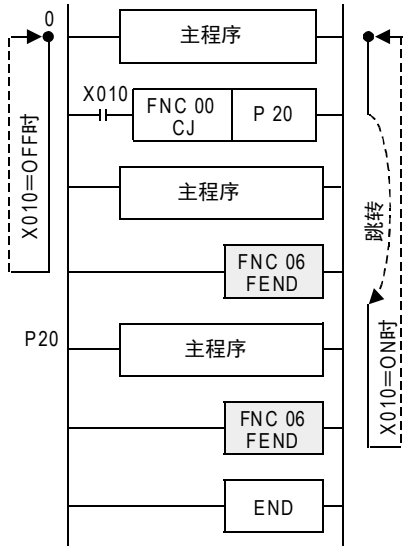
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

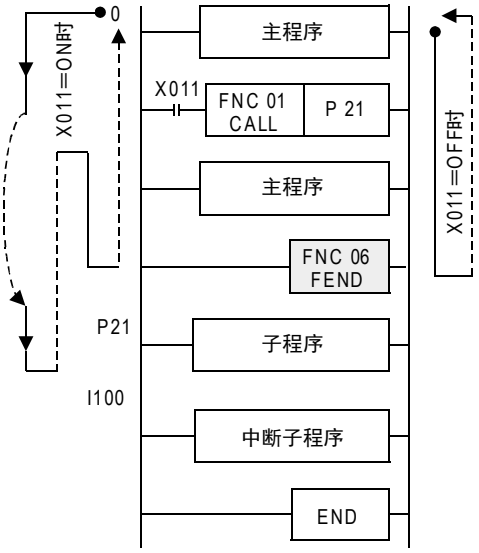
功能和动作说明

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子程序。

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
   执行CALL、CALLP指令后，到执行SRET指令之前，如执行了FEND指令，会错误。

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，会错误。

5. 使用中断功能(I)时
   必须在FEND指令后编写中断用标签(指针)，而且需要使用IRET指令。

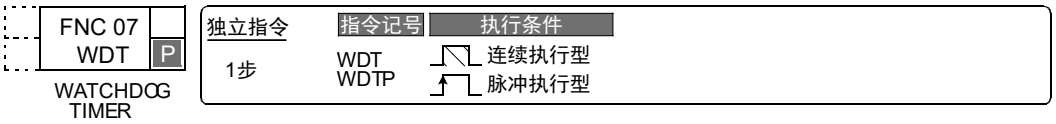
8.8 FNC 07 - WDT/看门狗定时器

A1 A2 A2C A8 A8C

概要

通过顺控程序对看门狗定时器进行刷新。

1. 指令格式



2. 设定数据

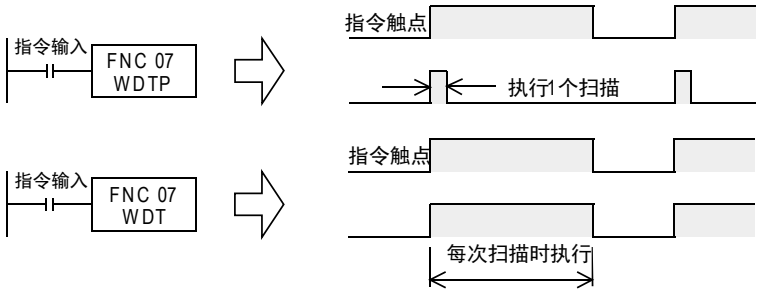
操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

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



相关软元件

软元件	名称	内容
D8000	看门狗 定时器时间	最大可以设定到32,767ms。(初始值: 200) 单位为ms

注意要点

1. 看门狗定时器错误的情况

如下所示的情况下，会产生看门狗定时器错误，所以在起始步的附近请输入下面的程序，以延长看门狗定时器时间，或是错开FROM/TO指令的执行时序。

1) 连接较多特殊扩展设备时的注意事项
在连接了较多数量的特殊扩展设备(定位、凸轮开关、模拟量、链接等)的系统中，可编程控制器运行时被执行的缓冲存储区的初始化时间会变长，运算时间会延长，因此有时会出现看门狗定时器错误。

2) 较多的FROM/TO指令同时驱动时的注意事项
执行多个FROM/TO 指令,传送多个缓冲存储区的时候，运算时间会延长，因此有时会出现看门狗定时器错误。

3) 高速计数器(软件计数器)较多时的注意事项
编写多个高速计数器同时对高频率脉冲进行计数时，运算时间会延长，因此有时会出现看门狗定时器错误。

2. 有更改看门狗定时器时间的方法。

→ 有关[看门狗定时器时间的变更]的详细内容，参考36.2.2项

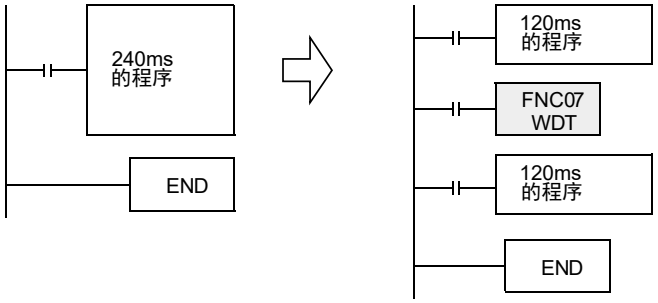
通过改写D8000(看门狗定时器时间)的内容，可以更改看门狗定时器的检测时间。(初始值为200ms)
输入下面的程序后，此后的顺控程序就会按照新的看门狗定时器时间进行监视。



程序举例

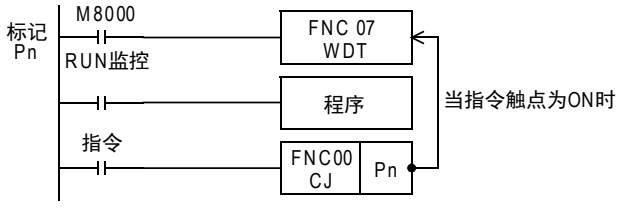
1. 出现运算周期长的故障时

例如将240ms的程序一分为二，在其中间编写WDT指令后，前半部分和后半部分都变成200ms以下。



2. CJ指令的标签(P)比CJ指令的步编号小的情况

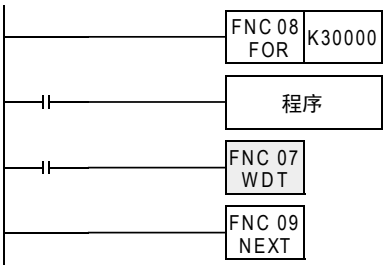
在标签(P)之后编程。



在指令触点中使用输入继电器(X)后，因为不能执行输入刷新，所以不能从P-CJ中脱出来。
在指令触点中，请使用可以在跳转的程序中被OFF的软元件。

3. FOR～NEXT指令的重复次数较多的情况

在FOR～NEXT之间编程。



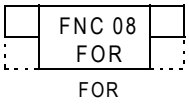
8.9 FNC 08 - FOR/循环范围的开始

A1 A2 A2C A8 A8C

概要

从FOR指令开始到NEXT(FNC 09)指令之间的程序按指定次数重复运行。

1. 指令格式



16位指令	指令记号	执行条件
3步	FOR	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	FOR～NEXT指令之间的重复次数 [(S·)=K1～K32767(-32768～0作为1处理。)]	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

→ 详细内容，参考NEXT(FNC 09)指令

相关指令

与NEXT(FNC 09)指令成对编程。

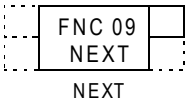
8.10 FNC 09 - NEXT/循环范围的结束

A1 A2 A2C A8 A8C

概要

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

1. 指令格式



独立指令	指令记号	执行条件	
1步	NEXT	连续执行型	不需要驱动触点的独立指令。

2. 设定数据

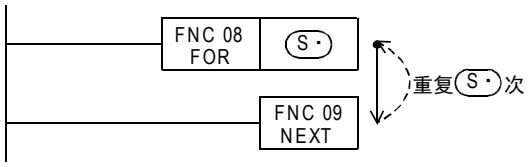
操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

FOR～NEXT指令的处理重复n次(源数据中指定的次数)
重复了指定次数后，执行NEXT指令后的步处理。



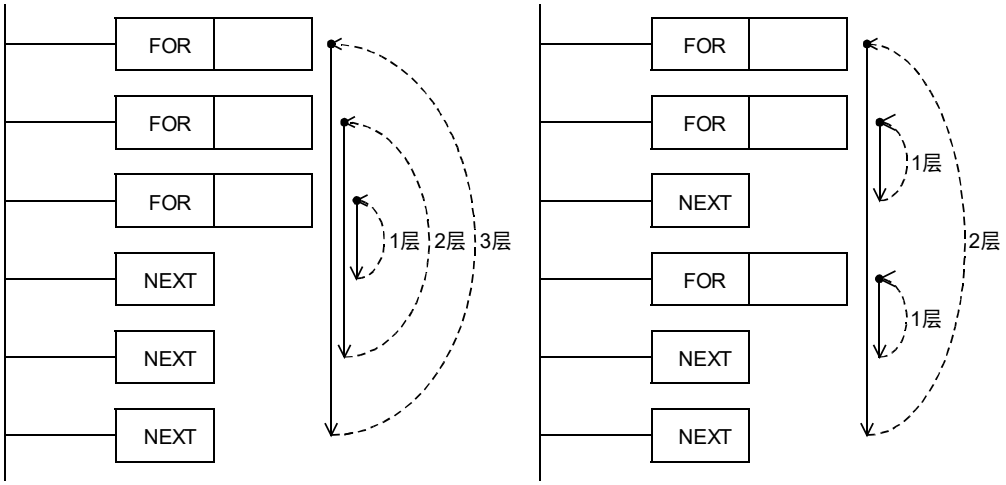
相关指令

与FOR(FNC 08)指令成对编程。

注意要点

1. 多层次数限制

在FOR－NEXT指令之间，将FOR－NEXT指令嵌套编程时，最多允许5层。



错误

1. 看门狗定时器错误

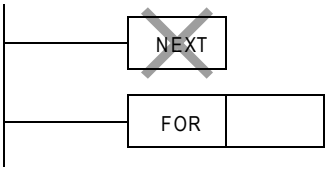
当重复次数设定较大，则循环时间(运算周期)(D8010)会变大，发生看门狗定时器错误的时候，需要更改看门狗定时器的时间，或是执行看门狗定时器的复位。

→ 有关[看门狗定时器时间的变更]的详细内容，参考36.2.2项
→ 有关看门狗定时器时间的复位情况，参考WDT(FNC 07)指令

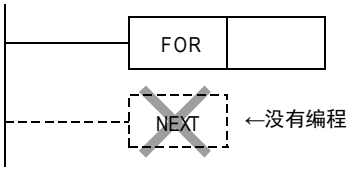
2. 容易错误的程序举例

下面的程序会错误。

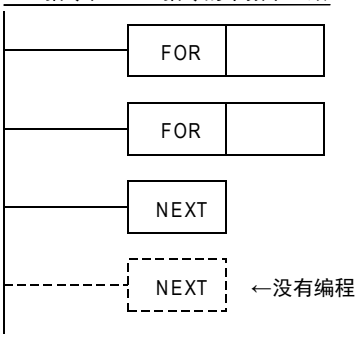
NEXT指令在FOR指令的前面。



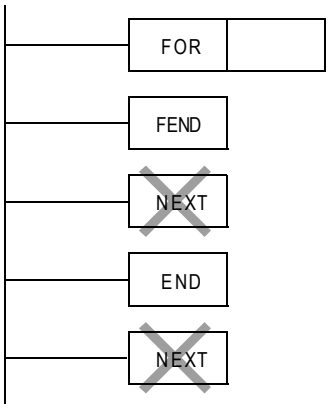
无NEXT 指令。



FOR 指令和NEXT指令的个数不一致。

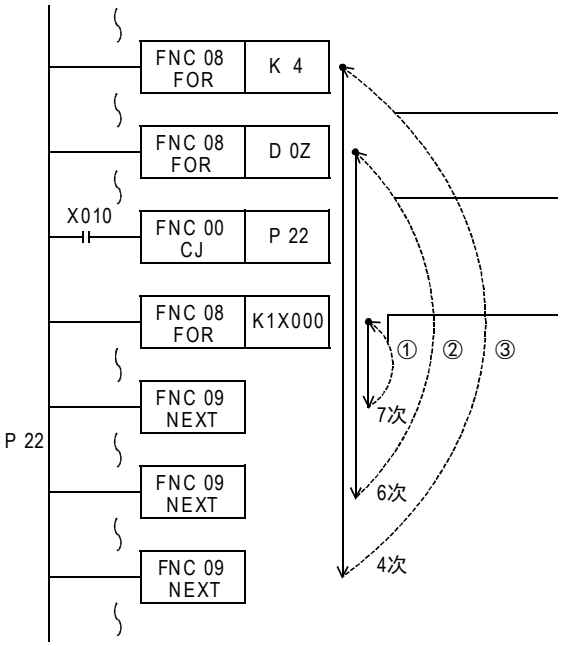


FEND指令或END指令之后有NEXT 指令。



程序举例

1. FOR～NEXT指令3个环路程序举例



③之间重复4次。

D0Z(Z为4的时候为D4)的数据值(当前值)为6时，②之间重复6次。

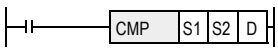
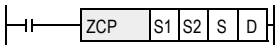
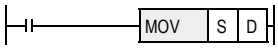
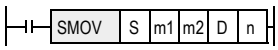
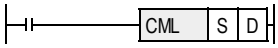
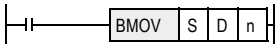
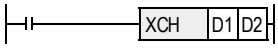
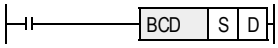
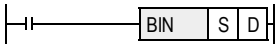
・X010=OFF时
K1X000=7时，①之间重复7次。
・X010=ON时
跳转到指针P22，不执行①。

关于①,②,③的重复次数

	X010=OFF	X010=ON
①	7×6×4=168次	0次
②	6×4=24次	24次
③	4次	4次

9. 传送•比较—FNC 10～FNC 19

FNC 10～FNC 19中，准备了使用应用指令时最为重要的数据传送和比较等基本的数据操作指令。

FNC №	指令记号	符号	功能	参考
10	CMP		比较	9.1节
11	ZCP		区间比较	9.2节
12	MOV		传送	9.3节
13	SMOV		位移动	9.4节
14	CML		反转传送	9.5节
15	BMOV		成批传送	9.6节
16	FMOV		多点传送	9.7节
17	XCH		交换	9.8节
18	BCD		BCD转换	9.9节
19	BIN		BIN转换	9.10节

9.1 FNC 10 - CMP/比较

A1 A2 A2C A8 A8C

概要

比较2个值，将其结果(大、一致、小)输出到位软元件中(3点)。

→ 有关触点比较指令，参考28復
→ 有关浮点数比较，参考18.1节

1. 指令格式

D	FNC 10 CMP	P

COMPARE

16位指令	指令记号	执行条件
7步	CMP	连续执行型
	CMPP	脉冲执行型

32位指令	指令记号	执行条件
13步	DCMP	连续执行型
	DCMPP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	成为比较值的数据或软元件编号	BIN16/32位
(S2・)	成为比较源的数据或软元件编号	BIN16/32位
(D・)	输出比较结果的起始位软元件编号	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D・)		●	●			●	▲1												●					

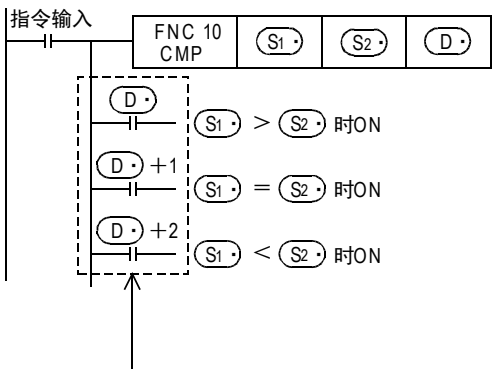
▲1 :D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲2 :仅对应A8/A8C可编程控制器。

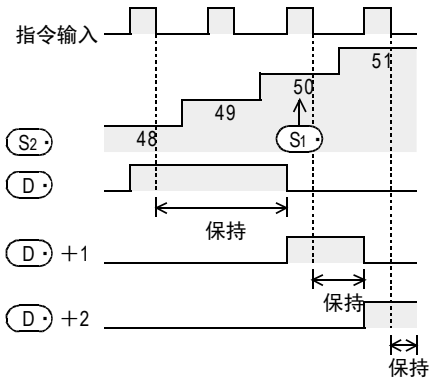
功能和动作说明

1. 16位运算(CMP、CMPP)

对比较值(S1・)和比较源(S2・)的内容进行比较，根据其结果(小、一致、大)，使(D・)、(D・)+1、(D・)+2其中一个为ON。

- 源数据(S1・)、(S2・)，作为BIN(二进制)的值进行处理。
- 按代数形式进行大小的比较。例如: -10＜2



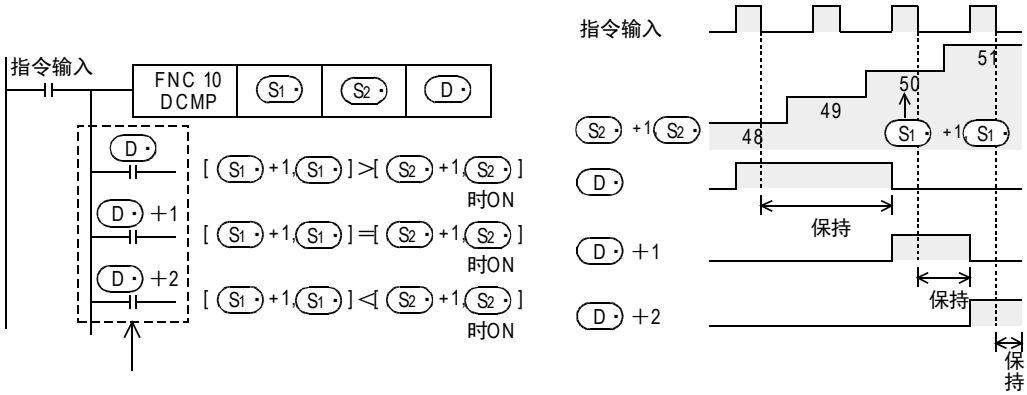


即使是指令输入为OFF,CMP指令不执行时，(D・) ～ (D・)+2也会保持当指令输入从ON变为OFF之前的状态。

2. 32位运算(DCMP、DCMPP)

对比较值[(S1·)+1,(S1·)]和比较源[(S2·)+1,(S2·)]的内容进行比较，根据其结果(小，一致，大)，使(D·)、(D·)+1、(D·)+2其中一个为ON。

- 源数据[(S1·)+1，(S1·)]、[(S2·)+1，(S2·)]作为BIN(二进制)的值进行处理。
- 按代数形式进行大小的比较。例如: -125400＜22466



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

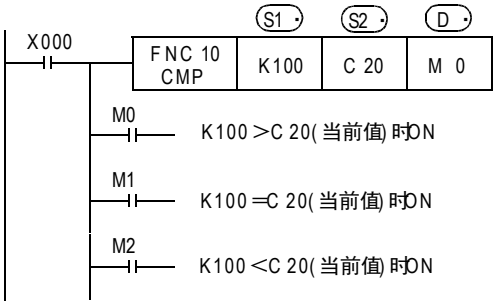
注意要点

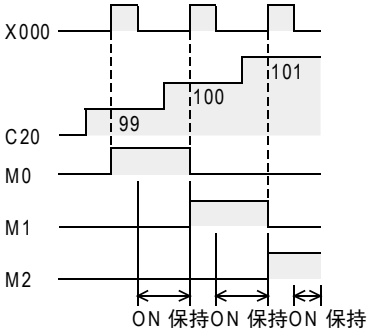
1. 软元件的占用点数

以(D·)中指定的软元件为起始点占用注意不要与其他控制中使用的软元件重复。

程序举例

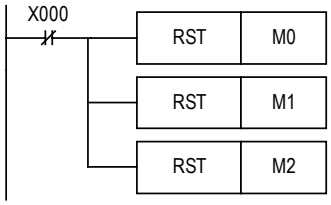
1. 比较计数器的当前值



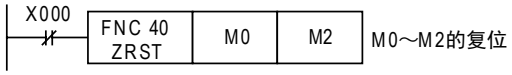


想在不执行指令时清除比较结果，请在上述的程序下方补充。

1) RST指令



2) ZRST指令



9.2 FNC 11 - ZCP/区间比较

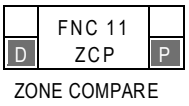
A1 A2 A2C A8 A8C

概要

针对2个值(区间)，将与比较源的值比较得出的结果(小于、等于（区域内）、大于)输出到位软元件(3点)中。

→ 有关触点比较指令，参考28復
→ 有关浮点数比较，参考18.2节

1. 指令格式



16位指令	指令记号	执行条件
9步	ZCP	连续执行型
	ZCPP	脉冲执行型

32位指令	指令记号	执行条件
17步	DZCP	连续执行型
	DZCPP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	下侧的比较值的数据或软元件编号	BIN16/32位
(S2・)	上侧的比较值的数据或软元件编号	BIN16/32位
(S・)	成为比较源的数据或软元件编号	BIN16/32位
(D・)	输出比较结果的起始位软元件编号	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D・)		●	●			●	▲1												●					

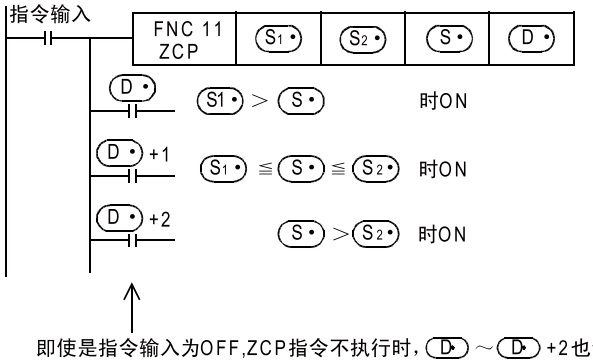
▲1 ：D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲2 ：仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(ZCP、ZCPP)

将比较源(S1・)的内容与下比较值(S2・)和上比较值(S・)进行比较，根据其结果(小、区域内、大)，使(D・)、(D・)+1、(D・)+2其中一个为ON。

• 按代数形式进行大小的比较。例如: -10＜2＜10

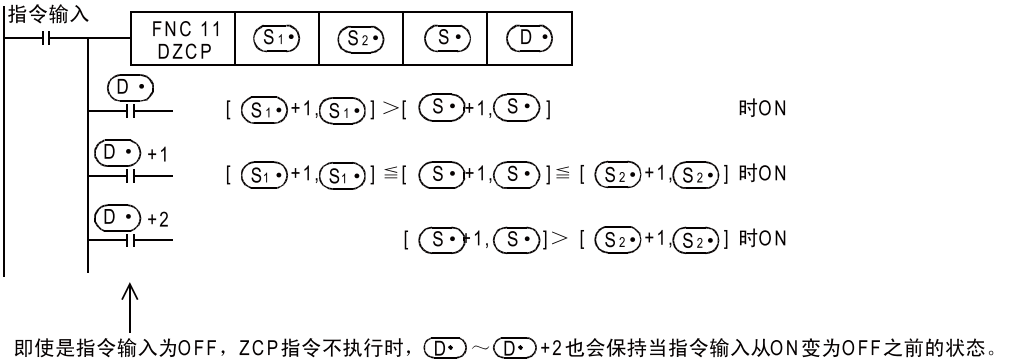


即使是指令输入为OFF,ZCP指令不执行时，(D・)～(D・)+2也会保持当指令输入从ON变为OFF之前的状态。

2. 32位运算(DZCP、DZCPP)

将比较源[(S·)+1，(S·)]的内容与下比較值[(S1·)+1，(S1·)]和上比較值[(S2·)+1，(S2·)]进行比较，根据其结果(小、区域内、大)，使(D·)、(D·)+1、(D·)+2其中一个为ON。

- 按代数形式进行大小的比较。例如: -125400＜22466＜1015444



注意要点

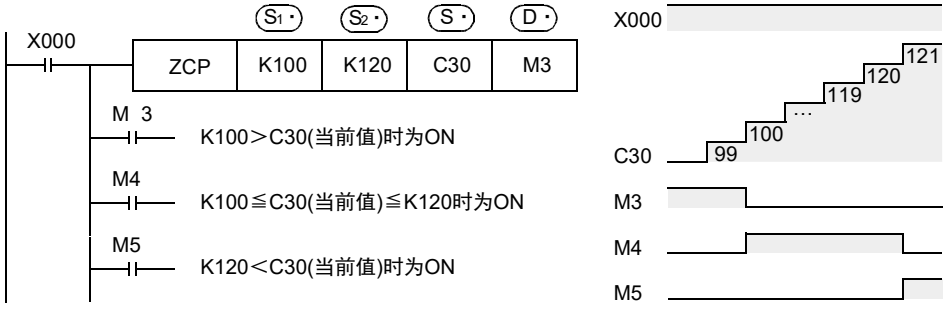
1. 软元件的占用点数

以(D·)中指定的软元件为起始点，占用注意不要与其他控制中使用的软元件重复。

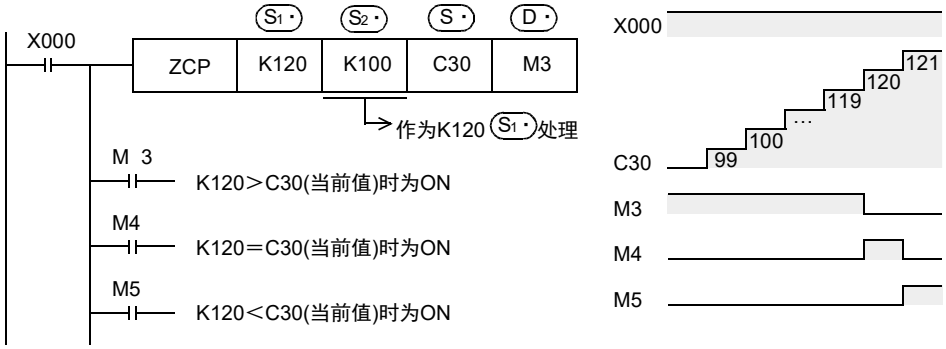
2. 根据上下比较值的大小的注意事项

下比较值(S1·)的值需要比上比较值(S2·)小。

1) 下比较值(S1·)＜上比较值(S2·)时



2) 下比较值(S1·)＞上比较值(S2·)时

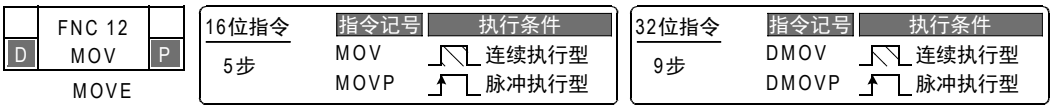


9.3 FNC 12 - MOV/传送

概要

将软元件的内容传送(复制)到其他的软元件中的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	传送源的数据，或是保存数据的软元件编号	BIN16/32位
(D・)	传送目标的软元件编号	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

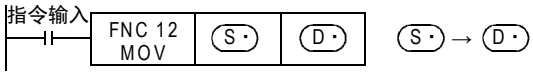
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(MOV、MOVP)

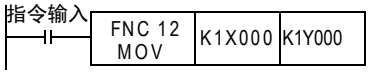
将传送源(S・)的内容1点传送给传送目标。(D・)

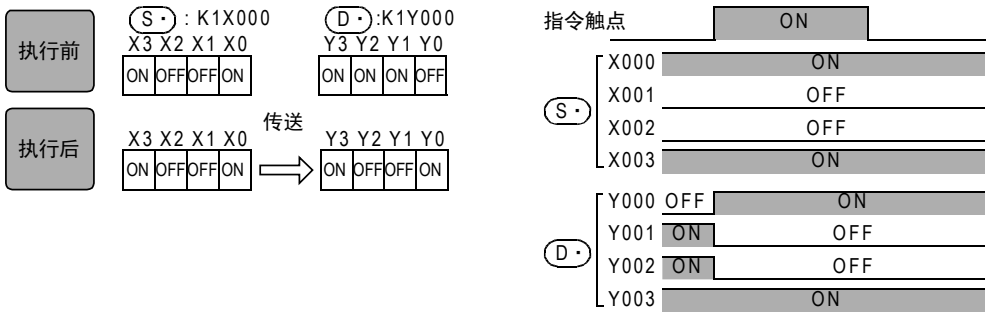
- 指令输入为OFF时，传送目标(D・)不变化。
- 传送源(S・)中指定了常数(K)时，会自动执行BIN转换。



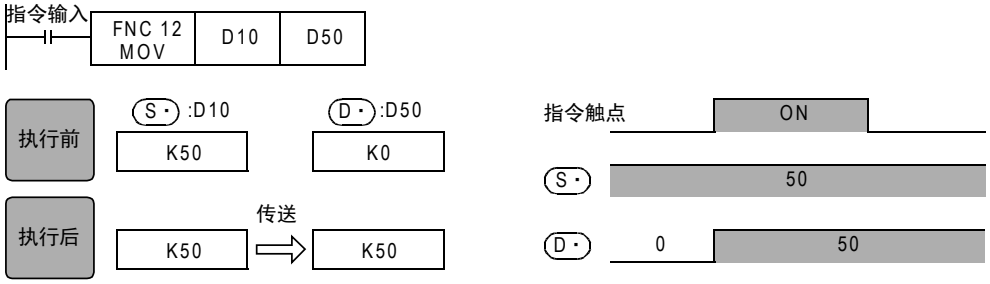
指定位软元件的位数(K1X000→K1Y000)的情况

最多传送16个(4的倍数)位软元件





字软元件的情况
传送一个字软元件。

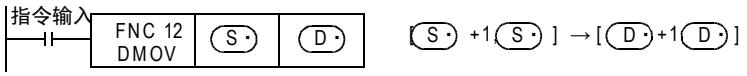


2. 32位运算(DMOV、DMOVP)

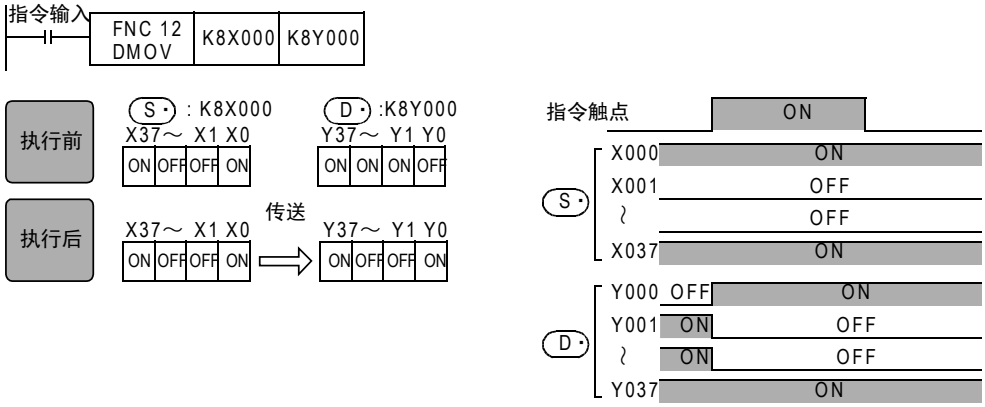
将传送源[(S·)+1，(S·)]的内容1点传送到传送目标[(D·)+1，(D·)]中。(字软元件为2点的传送)

指令输入为OFF时，传送目标(D·)不变化。

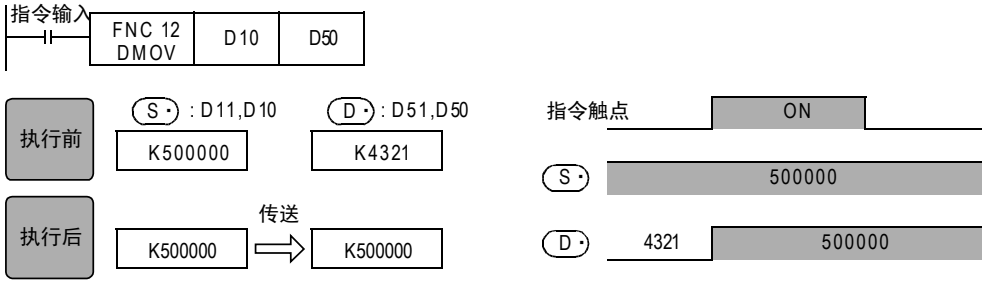
传送源[(S·)+1，(S·)]中指定了常数(K)时，会自动执行BIN转换。



指定位软元件的位数(K8X000→K8Y000)的情况
最多传送32个(4的倍数)位软元件

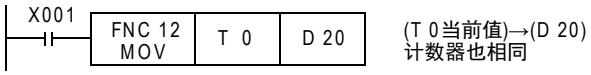


字软元件的情况
传送两个字软元件。



程序举例

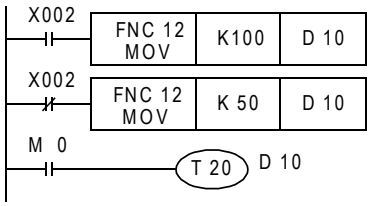
1. 读出定时器、计数器当前值的例子



(T 0当前值)→(D 20)
计数器也相同

2. 间接指定定时器、计数器设定值的例子

通过开关(X002)的ON/OFF可以对定时器(T20)设定2个设定值。
2个以上时，需要使用多个开关。



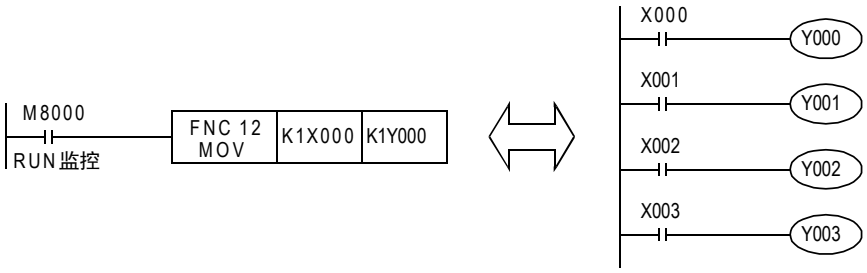
(K100)→(D 10)

(K50)→(D 10)

X002＝ON时　D 10＝K100(10秒定时器)
X002＝OFF时　D 10＝K50(5秒定时器)

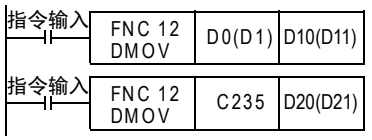
3. 位软元件的传送

如右图所示采用基本指令的梯形图，也可以用下面的MOV指令来表现。



4. 32位数据的传送

运算结果作为32位被输出的应用指令(MUL等)或者用 32位的数值、或是32位的位软元件传送高速计数器当前值(C235～C255)时，必须使用DMOV指令。



(D 1,D 0)→(D 11,D 10)

(C235当前值)→(D 21,D 20)

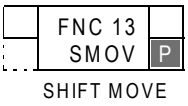
9.4 FNC 13 - SMOV/位移动

A8 A8C

概要

以位数为单位(4位)进行数据分配合成的指令。

1. 指令格式



16位指令	指令记号	执行条件
11步	SMOV	连续执行型
	SMOVP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存有要进行位移动的数据软元件的编号	BIN16位
m1	要移动的起始位的位置	BIN16位
m2	要移动的位的个数	BIN16位
(D·)	保存已经进行位移动的数据的软元件编号	BIN16位
n	移动目标的起始位的位置	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●					
m1																				●	●			
m2																				●	●			
(D·)									●	●	●	●	●	●	●	▲	●	●	●					
n																				●	●			

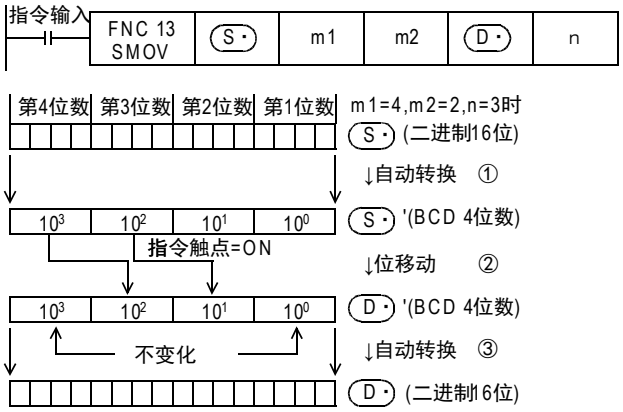
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(SMOV、SMOVP)

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

- 指令输入为OFF时，传送目标 (D・) 不变化。
- 指令输入为ON时，传送源 (S・) 的数据以及传送目标 (D・) 中的指定传送以外的位数不变化。



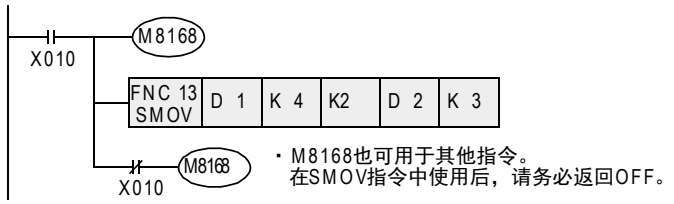
① (S・) 从BIN转换为BCD。

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

③ 合成的数据(BCD)转换成BIN后，保存到 (D・) 中。

2. 扩展功能

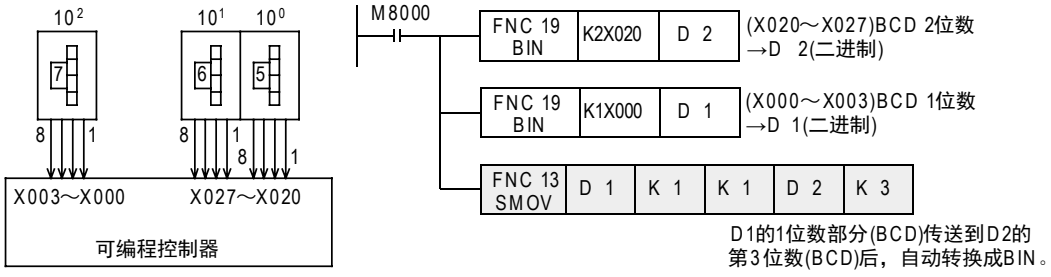
将M8168置ON后，执行SMOV指令时，则不能进行BIN→BCD转换。
位移动以4位为单位执行。



・M8168也可用于其他指令。
在SMOV指令中使用后，请务必返回OFF。

程序举例

合成3位数的数字式开关的数据后，以二进制保存到D2中。



将非连续的输入端子中连接的3个数字式开关的数据进行合成。

D1的1位数部分(BCD)传送到D2的第3位数(BCD)后，自动转换成BIN。

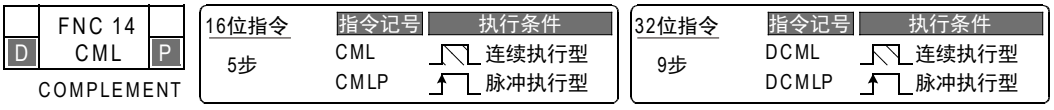
9.5 FNC 14 - CML/反转传送

A8 A8C

概要

以位为单位反转数据后进行传送(复制)的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	要执行反转的数据，或是保存数据的字软元件编号	BIN16/32位
(D·)	保存要执行反转后的数据的目标字软元件编号	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

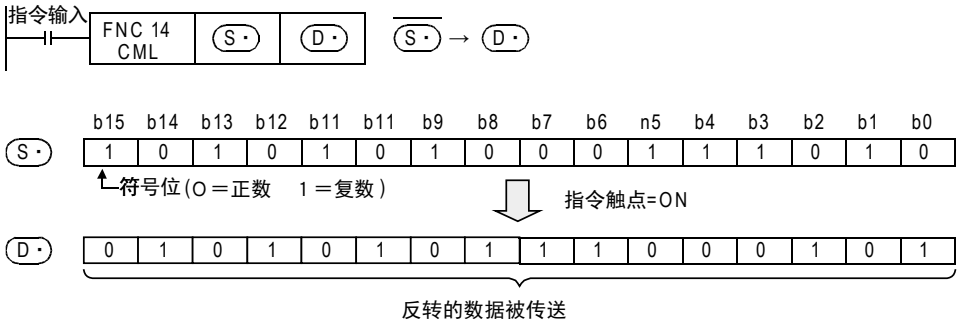
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(CML、CMLP)

将(S·)中指定的软元件的各位反转(0→1,1→0)后，传送至(D·)。

- 在(S·)中指定常数(K)时，会自动转换为BIN。
- 希望将可编程控制器的输出以逻辑反转输出时可以使用。

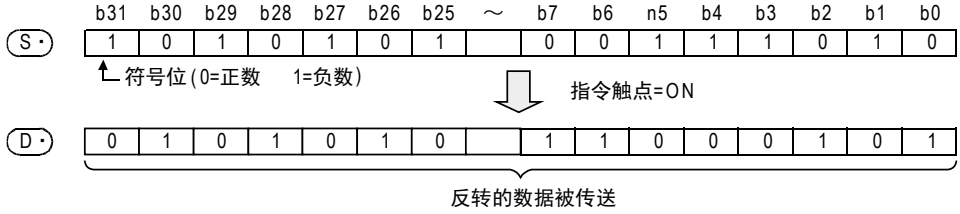


2. 32位运算(DCML、DCMLP)

将[(S・)+1，(S・)]中指定的软元件的各位反转(0→1, 1→0)后，传送至[(D・)+1，(D・)]。

- [在(S・)+1，(S・)]中指定常数(K)时，会自动转换为BIN。
- 希望将可编程控制器的输出以逻辑反转输出时可以使用。

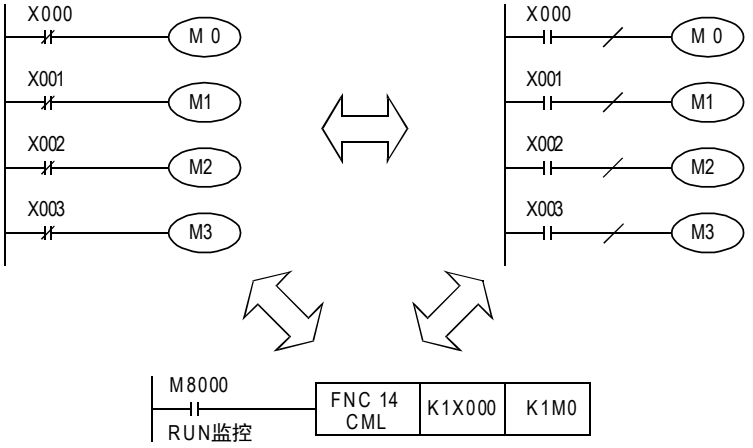




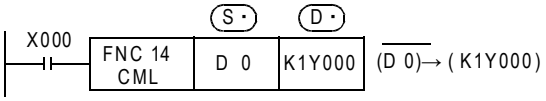
程序举例

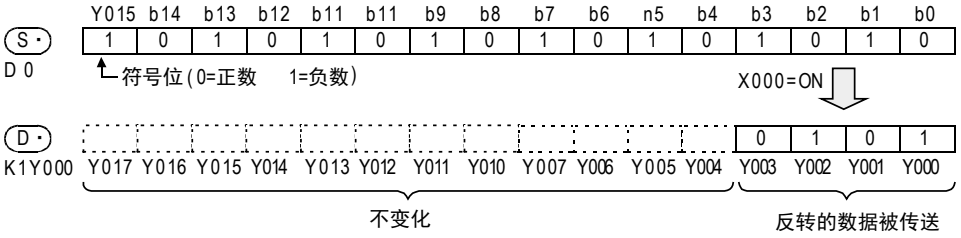
1. 反转输入的获取

可以使用CALL指令编写下述的顺控程序。



2. 指定位数的软元件的位数为4点的情况





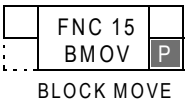
9.6 FNC 15 - BMOV/成批传送

A8 A8C

概要

对指定点数的多个数据进行成批传送(复制)。

1. 指令格式



16位指令	指令记号	执行条件
7步	BMOV	连续执行型
	BMOVP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	传送源的数据，或是保存数据的软元件编号	BIN16位
(D·)	传送目标的软元件编号	BIN16位
n	传送点数)[n≦512]	BIN16位

(包括文件寄存器

3. 对象软元件

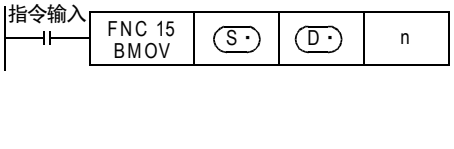
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲			●					
(D·)									●	●	●	●	●	●	●	▲			●					
n														●						●	●			

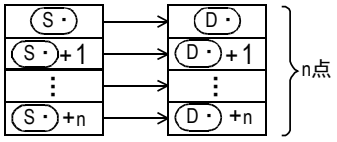
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

将 (S·) 开始的n点的数据成批传送到 (D·) 开始的n点中。

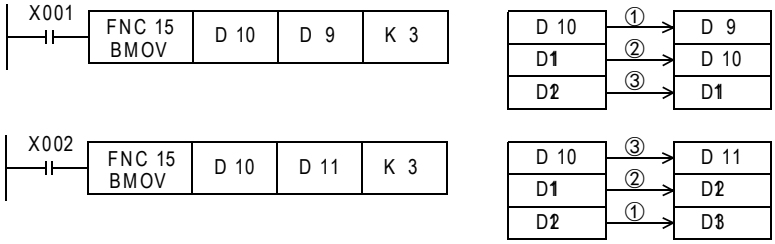
- 超出软元件编号范围时，在可能的范围内传送。





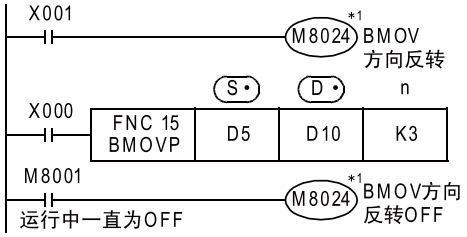
即使传送编号范围重迭也能传送

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



扩展功能(双向传送功能)

通过控制BMOV(FNC 15)指令的方向反转标志位M8024*1，可以在1个程序中实现双向传送。

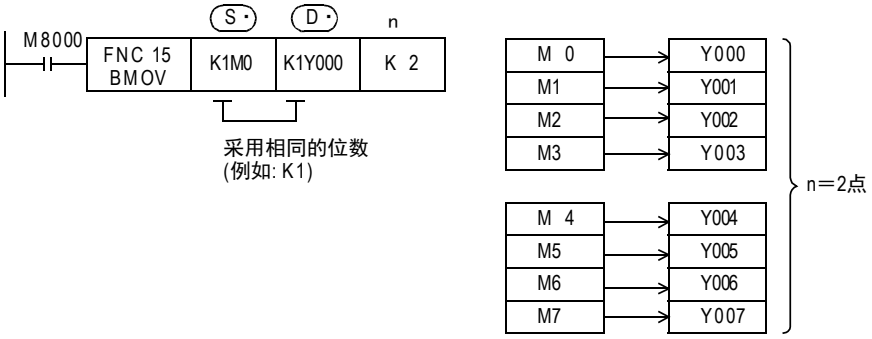


BMOV方向反转标志位	传送方向	(S・), (D・)
M8024*1: OFF	(S・) → (D・)	D5→D10 D6→D11 D7→D12
M8024*1: ON	(S・) ← (D・)	D5→D10 D6←D11 D7←D12

*1. M8024在RUN→STOP时被清除。

注意要点

带有位数指定的位软元件的情况下，(S・)和(D・)要采用相同的位数。



9.6.1 文件寄存器 ←→ 数据寄存器之间的传送功能

BMOV(FNC 15)指令，对文件寄存器(D1000以后)具备特别的功能

→ 文件寄存器的详细内容，参考4.9节

1. 文件寄存器

通过参数设定，可以将D1000～D7999作为文件寄存器写入到程序内存区域中，或从中读取。

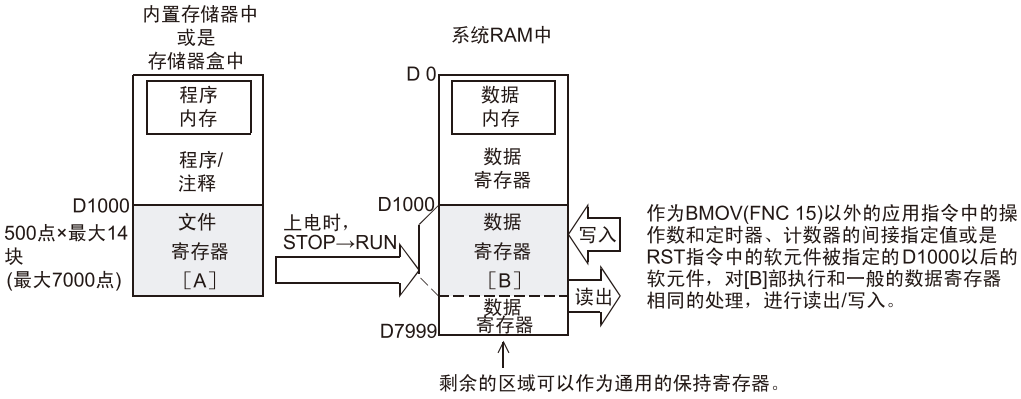
1) 设定概要
文件寄存器(D1000～D7999)在初始状态下不存在，只有通过编程工具进行参数设定，确保了文件寄存器点数后才变为有效。

2) 文件寄存器点数
在参数设定中，以文件寄存器500点为1块进行设定。
可以设定的点数，是可以指定1～14块(每1块500点)。
每1块会使程序内存区减少500步。

3) BMOV(FNC 15)指令和其他指令的区别
对文件寄存器(D1000以后)而言，BMOV(FNC 15)和其他指令的区别如下表所示。

指令	传送内容	备注
BMOV指令	可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	对于映象存储区中的数据寄存器区域[B]部，可以和普通的数据寄存器一样地进行读出/写入的处理。	由于数据寄存器区域[B]部，是设在可编程控制器内的系统RAM中，因此不受存储器盒形式的限制，可以随意地更改内容。

被设定为文件寄存器的数据寄存器，在上电时数据会自动地从文件寄存器区域[A]部复制到数据寄存器区域[B]部。



2. 使用上的注意事项

1) 更新同编号的文件寄存器内容(同编号更新模式)时，文件寄存器编号必须设定成 (S・) = (D・)。

2) 使用同编号更新模式时，请注意以n指定的传送点数不能超出文件寄存器的区域。

3) 使用同编号更新模式时，若超出了文件寄存器区域，则发生运算错误(M8067)，指令不执行。

4) 执行变址修饰(同编号更新模式)时
(S・)，(D・)的变址修饰其实际的软元件编号在文件寄存器区域中，与此同时，传送点数也在文件寄存器范围内，只有这样才执行指令。

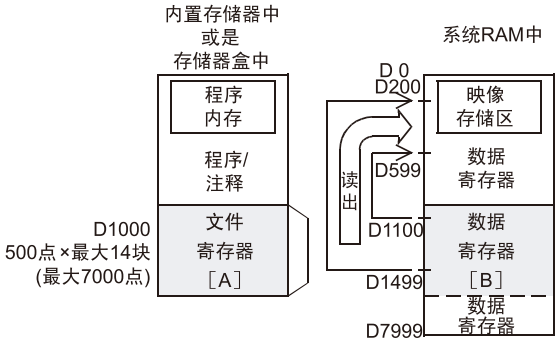
5) 存储器盒的处理
更改存储器盒中的文件寄存器内容时，请按照以下条件执行。
— 存储器盒的写保护开关请置于OFF一侧。
— 写入到存储器盒(快闪存储器)时，连续的1个块 (500点)需要66～132ms。写入到存储器盒(EEPROM)时，连续的1个块(500点)需要66～80ms。
其间程序的执行被中断。此时，看门狗定时器不会被刷新，因此需要在用户程序中采取插入WDT指令等的对应措施。

6) 关于存储器允许写入次数
存储器盒允许写入次数在1万次以下。
通过程序进行写入时，如使用连续执行型指令，在可编程控制器的每个运算周期中都会写入存储器盒。
如要避免这种情况，必须使用脉冲执行型(BMOVP)指令。

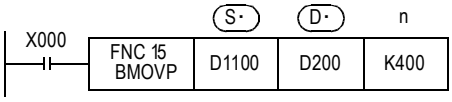
7) 有关文件寄存器的动作
在内置存储器或是存储器盒中可保存文件寄存器。
对文件寄存器的读出/写入，与普通的数据寄存器不同，只有用外围设备，或是BMOV(FNC 15)指令才可以直接处理文件寄存器。

8) BMOV(FNC 15)指令的目标操作数中未指定文件寄存器时，不访问文件寄存器。

a) 存储器的动作概要



b) 程序举例
X000为ON后，读出数据寄存器区域[B]部。



也可以在(D・)中指定文件寄存器，但是若指定了与(S・)相同的编号，则变为同编号寄存器更新模式。但是，即使在(S・)，(D・)中指定了不同编号的文件寄存器，也不能从文件寄存器区域向文件寄存器区域传送数据。类似这样的情况下，需要先在同编号寄存器更新模式中，将(S・)中指定的文件寄存器内容读出到数据寄存器区域[B]部以后，执行写入。

→ 有关文件寄存器的同编号寄存器更新模式，参考4.9.2项

9.7 FNC 16 - FMOV/多点传送

A8 A8C

概要

将同一数据传送到指定点数的软元件中的进行多点传送指令。

1. 指令格式

D	FNC 16 FMOV	P

FILL MOVE

16位指令	指令记号	执行条件
7步	FMOV	连续执行型
	FMOVP	脉冲执行型

32位指令	指令记号	执行条件
13步	DFMOV	连续执行型
	DFMOVP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	传送源的数据、或是保存数据的软元件编号	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。)	BIN16/32位
n	传送点数[K1≤n≤K512，H1≤n≤H1FF]	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统·用户							位数指定				系统·用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲			●					
n																				●	●			

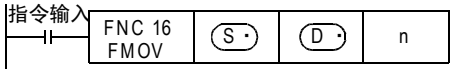
▲: 仅对应A8/A8C可编程控制器。

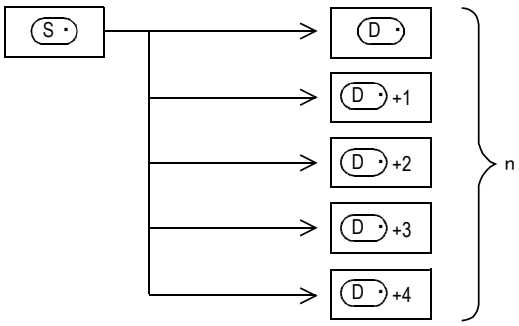
功能和动作说明

1. 16位运算(FMOV、FMOVP)

将(S·)的内容传送到以(D·)起始的n点的软元件中。

- n点的软元件内容都相同。
- 以n指定的个数超出了软元件编号范围时，在可能的范围内传送。
- 指令输入为OFF时，传送目标(D·)不变化。
- 指令输入为ON时，传送源(S·)的数据不变化。
- 传送源(S·)中指定了常数(K)时，会自动执行BIN转换。

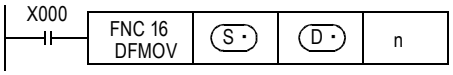


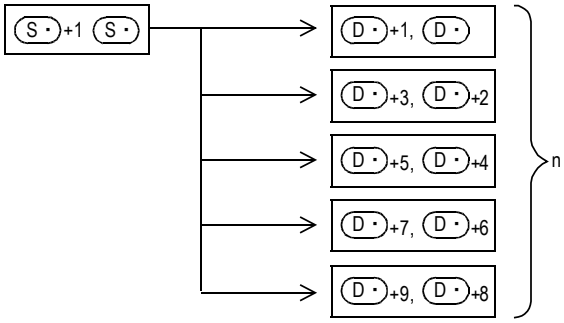


2. 32位运算(DFMOV、DFMOVP)

将[(S・)+1，(S・)]的内容传送到以[(D・)+1，(D・)]起始的n点的32位软元件中。

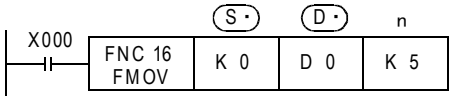
- n点的32位软元件内容都相同。
- 以n指定的个数超出了软元件编号范围时，在可能的范围内传送。
- 指令输入为OFF时，传送目标[(D・)+1，(D・)]不变化。
- 指令输入为ON时，传送源[(S・)+1，(S・)]的数据不变化。
- 传送源[(S・)+1，(S・)]中指定了常数(K)时，会自动执行BIN转换。

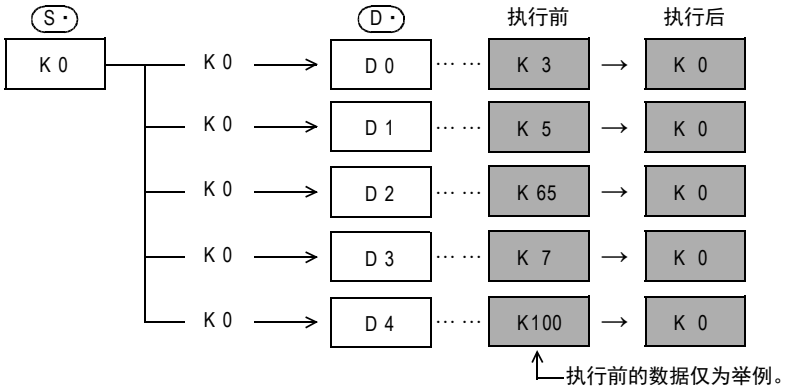




程序举例

1. 指定数据多次写入





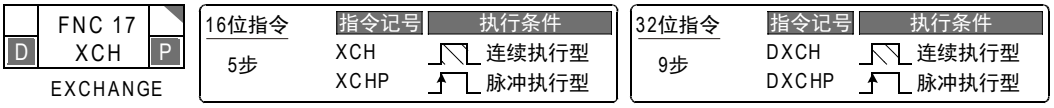
9.8 FNC 17 - XCH/交换

A8 A8C

概要

在2个软元件之间进行数据交换。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D1·)	保存交换数据的软元件编号	BIN16/32位
(D2·)	保存交换数据的软元件编号	BIN16/32位

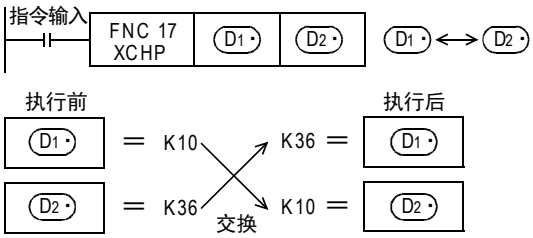
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D1·)									●	●	●	●	●	●	●	●	●	●	●					
(D2·)									●	●	●	●	●	●	●	●	●	●	●					

功能和动作说明

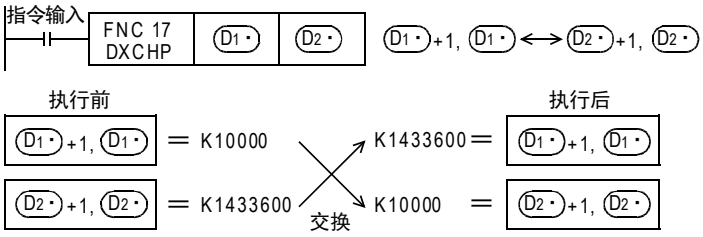
1. 16位运算(XCH、XCHP)

(D1·) 和 (D2·) 相互之间进行数据交换。



2. 32位运算(DXCH、DXCHP)

[(D1·)+1，(D1·)]和[(D2·)+1，(D2·)]相互之间进行数据交换。



错误

下面的情况下会运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- M8160为ON时，(D1·) 和 (D2·) 的软元件编号不一致的时候。

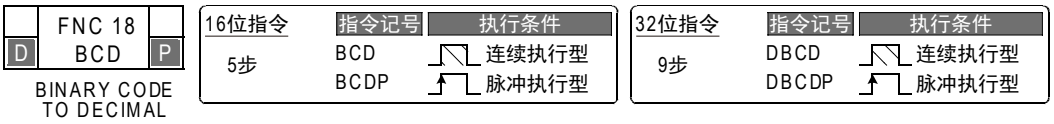
9.9 FNC 18 - BCD/BCD转换

A8 A8C

概要

将BIN(2进制数)转换成BCD(10进制数)后传送的指令。
可编程控制器的运算按照BIN(2进制数)数据进行处理，在带BCD译码的7段码显示器中显示数值时,可使用本指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	保存转换源(2进制数)数据的字软元件编号	BIN16/32位
(D・)	转换目标(10进制数)的字软元件编号	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●					
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

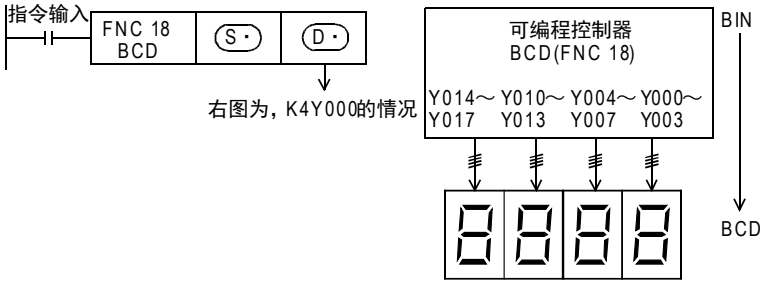
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(BCD、BCDP)

将(S・)的BIN(2进制数)数据转换成BCD(10进制数)数据后传送到(D・)中。

- (S・)的数据可以转换成K0～K9999的BCD(10进制数)。
- (S・)和(D・)指定位数的时候，参考下表。

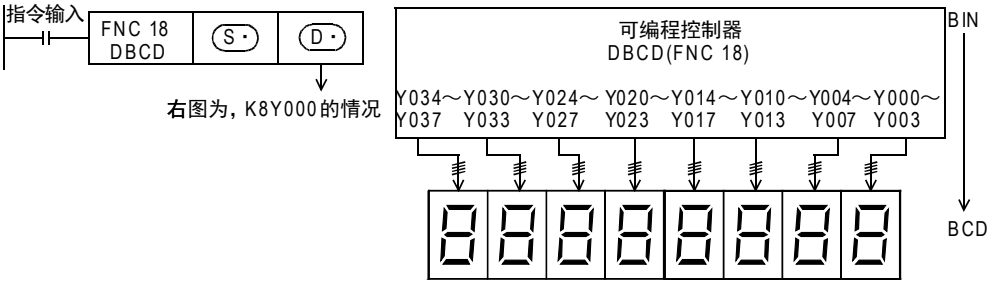


(D・)	位数	数据范围
K1Y000	1位数	0～9
K2Y000	2位数	00～99
K3Y000	3位数	000～999
K4Y000	4位数	0000～9999

2. 32位运算(DBCD、DBCDP)

将[(S·)+1，(S·)]的BIN(2进制数)数据转换成BCD(10进制数)数据后传送到[(D·)+1，(D·)]中。

- [(S·)+1，(S·)]的数据，可以转换成K0～K99999999的BCD(10进制数)。
- [(S·)+1，(S·)]和[(D·)+1，(D·)]指定位数的时候，参考下表。



[(D·)+1，(D·)]	位数	数据范围
K1Y000	1位数	0～9
K2Y000	2位数	00～99
K3Y000	3位数	000～999
K4Y000	4位数	0000～9999
K5Y000	5位数	00000～99999
K6Y000	6位数	000000～999999
K7Y000	7位数	0000000～9999999
K8Y000	8位数	00000000～99999999

相关指令

指令	功能
BIN(FNC 19)	10进制数(BCD)→2进制数(BIN)的转换

注意要点

1. 使用SEGL(FNC 74)、ARWS(FNC 75)时

由于SEGL(FNC 74)、ARWS(FNC 75)能够自动执行BCD和BIN之间的转换，因此不需要使用该指令。

2. 有关BCD的输入输出处理

四则运算(+−×÷)和加一，减一指令等可编程控制器内的运算都以BIN(2进制数)执行。

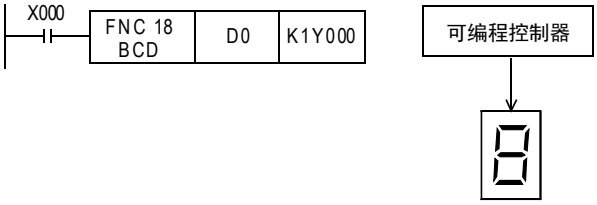
- 在将BCD(10进制数)的数字式开关信息读入可编程控制器时，使用BIN(FNC 19)的BCD→BIN转换传送指令。
- 在向BCD(10进制数)的7段码显示器进行输出时，使用BCD(FNC 18)的BIN→BCD的转换传送指令。

错误

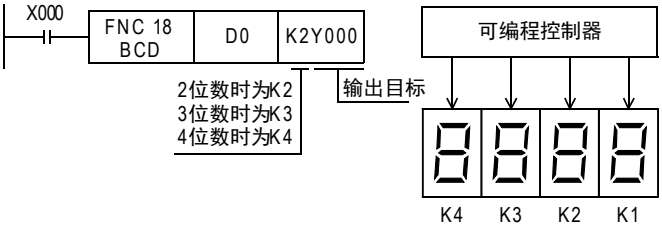
BCD、BCDP(16位指令)的情况下，(S·)的值若在0～9,999以外范围时出现运算错误。
DBCD、DBCDP(32位指令)的情况下，(S·)的值若在0～99,999,999以外范围时出现运算错误。

程序举例

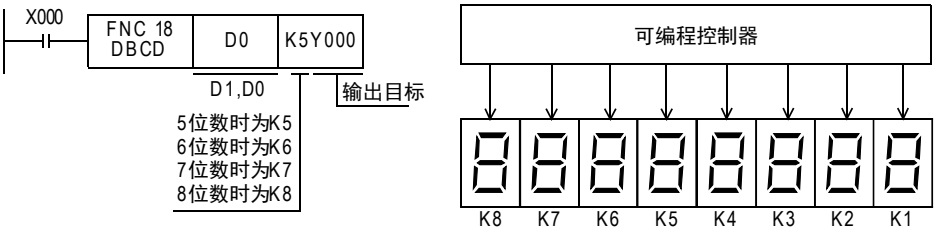
1. 7段数码管显示1位数的情况



2. 7段数码管显示2位数以上、4位数以下的情况



3. 7段数码管显示5位数以上、8位数以下的情况



9.10 FNC 19 - BIN/BIN转换

A8 A8C

概要

将10进制数(BCD)转换成2进制数(BIN)的指令。
在将数字式开关之类，以BCD(10进制数)设定的数值转换成可编程控制器的运算中可以处理的BIN(2进制数)数据后读取的情况下，可以使用本指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	BIN	连续执行型
	BINP	脉冲执行型

32位指令	指令记号	执行条件
9步	DBIN	连续执行型
	DBINP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存转换源(10进制数)数据的字软元件编号	BIN16/32位
(D·)	转换目标(2进制数)的字软元件编号	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●					
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

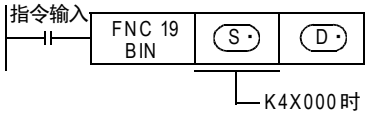
▲: 仅对应A8/A8C可编程控制器。

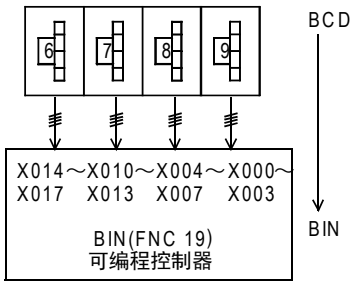
功能和动作说明

1. 16位运算(BIN、BINP)

将(S·)的BCD(10进制数)数据转换成BIN(2进制数)数据后传送到(D·)中。

- (S·)的数据可以在0～9999(BCD)的范围内转换。
- (S·)和(D·)指定位数时，参考下表。



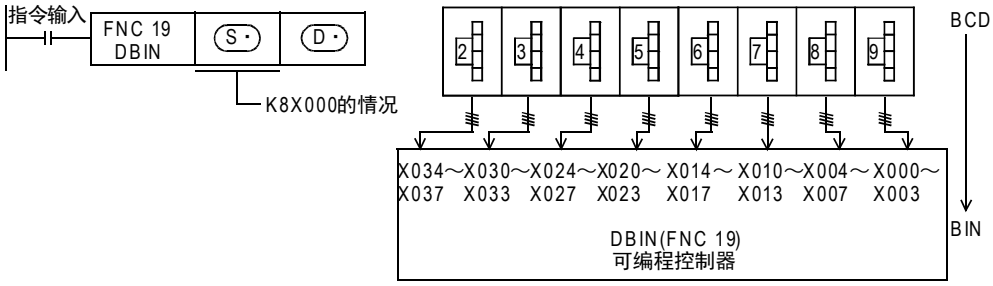


(S·)	位数	数据范围
K1X000	1位数	0～9
K2X000	2位数	00～99
K3X000	3位数	000～999
K4X000	4位数	0000～9999

2. 32位运算(DBIN、DBINP)

将[(S・)+1，(S・)]的BCD(10进制数)数据转换成BIN(2进制数)数据后传送到[(D・)+1，(D・)]中。

- [(S・)+1，(S・)]的数据，可以在0～99999999(BCD)的范围内转换。
- [(S・)+1，(S・)]和[(D・)+1，(D・)]指定位数时，参考下表。



(S・)+1，(S・)	位数	数据范围
K1X000	1位数	0～9
K2X000	2位数	00～99
K3X000	3位数	000～999
K4X000	4位数	0000～9999
K5X000	5位数	00000～99999
K6X000	6位数	000000～999999
K7X000	7位数	0000000～9999999
K8X000	8位数	00000000～99999999

相关指令

指令	功能
BCD(FNC 18)	2进制数(BIN)转换成10进制数(BCD)

注意要点

1. 使用DSW(FNC 72)时

由于DSW(FNC 72)指令能够自动执行BCD和BIN之间的转换，因此不需要使用该指令。

2. 有关BCD的输入输出处理

四则运算(+−×÷)和加一，减一指令等可编程控制器内的运算都以BIN(2进制数)执行。

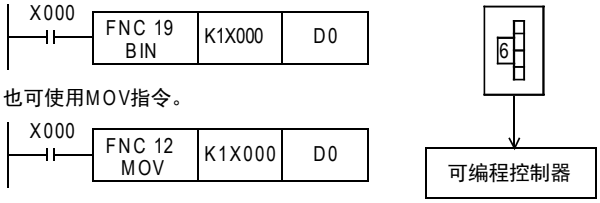
- 在将BCD(10进制数)的数字式开关信息读入可编程控制器中时，使用BIN(FNC 19)的BCD→BIN转换传送指令。
- 在向BCD(10进制数)的7段数码管显示进行输出时，使用BCD(FNC 18)的BIN→BCD的转换传送指令。

错误

非BCD(10进制数)时，M8067(运算错误)为ON。

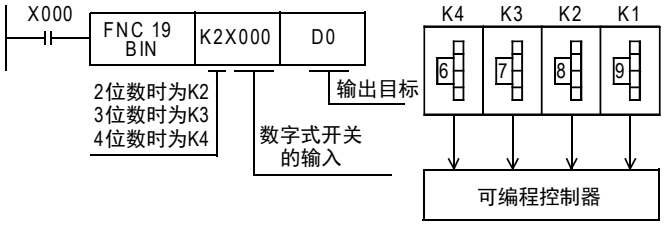
程序举例

1. 数字式开关1位数的情况

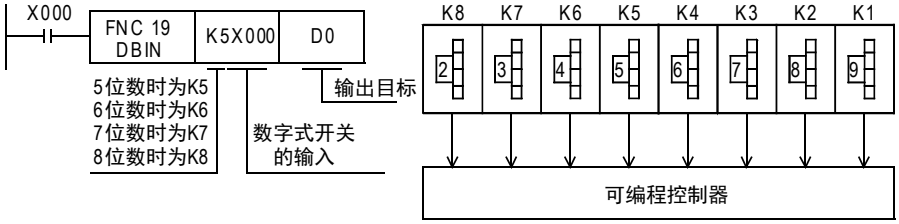


也可使用MOV指令。

2. 数字式开关2位数以上、4位数以下的情况

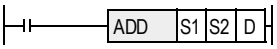
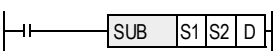
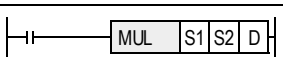
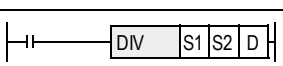
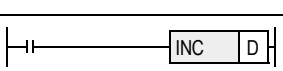
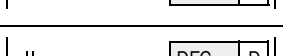
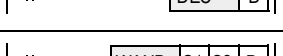
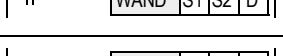
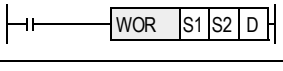
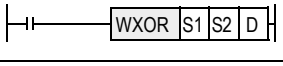


3. 数字式开关5位数以上、8位数以下的情况



10. 四则逻辑运算—FNC 20～FNC 29

FNC 20～FNC 29中，准备了针对数值数据执行四则运算及逻辑运算的指令。

FNC No.	指令记号	符号	功能	参考
20	ADD		BIN加法运算	10.1节
21	SUB		BIN减法运算	10.2节
22	MUL		BIN乘法运算	10.3节
23	DIV		BIN除法运算	10.4节
24	INC		BIN加一	10.5节
25	DEC		BIN减一	10.6节
26	WAND		逻辑与	10.7节
27	WOR		逻辑或	10.8节
28	WXOR		逻辑异	10.9节
29	NEG		补码	10.10节

有关浮点数运算指令

A8/A8C可编程控制器中，除了BIN形式的四则运算指令以外，还准备了浮点数形式的四则运算指令。

FNC No.	指令记号	处理内容
120	[D]EADD	2进制浮点数加法运算
121	[D]ESUB	2进制浮点数减法运算
122	[D]EMUL	2进制浮点数乘法运算
123	[D]EDIV	2进制浮点数除法运算

有关详细内容，请参阅各指令的解说。

→ 有关浮点数运算，参考18章

10.1 FNC 20 - ADD/BIN加法运算

A1 A2 A2C A8 A8C

概要

2个值进行加法运算(A+B=C)后得出结果的指令。

→ 有关浮点数加法运算[EADD(FNC 120)指令]，参考18.8节

1. 指令格式

D	FNC 20 ADD	P

ADDITION

16位指令	指令记号	执行条件
7步	ADD	连续执行型
	ADDP	脉冲执行型

32位指令	指令记号	执行条件
13步	DADD	连续执行型
	DADDP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	加法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(S2·)	加法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(D·)	保存加法运算结果的字软元件编号	BIN16/32位

3. 对象软元件

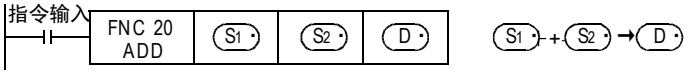
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(ADD、ADDP)

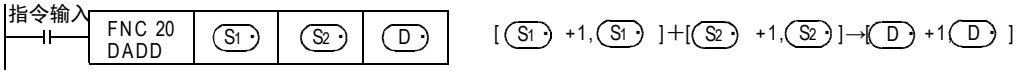
将(S1·)和(S2·)的内容进行二进制加法运算后传送到(D·)中。



- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行加法运算。
  5+(-8)=-3
- (S1·)和(S2·)中指定常数(K)时，会自动进行BIN转换。

2. 32位运算(DADD,DADDP)

将[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]的内容进行二进制加法运算后传送到[(D·)+1，(D·)]中。



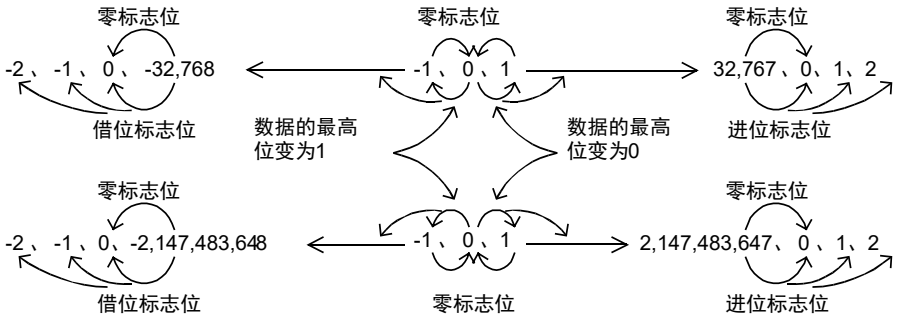
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行加法运算。
  5,500+(-8,540)=-3,040
- [(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定常数(K)时，会自动进行BIN转换。

相关软元件

1. 标志位的动作及数值的正负的关系

→ 有关标志位的动作，参考6.5.2项

软元件	名称	内容
M8020	零位	ON: 运算结果为0时 OFF: 运算结果为0以外时
M8021	借位	ON: 运算结果小于-32,768(16位运算)或是-2,147,483,648(32位运算)时，借位标志位动作。 OFF: 运算结果不小于-32,768(16位运算)或是-2,147,483,648(32位运算)时
M8022	进位	ON: 运算结果大于 32,767(16 位运算)或者 2,147,483,647(32 位运算)时，进位标志位动作。 OFF: 运算结果不大于32,767(16位运算)或者2,147,483,647(32位运算)时



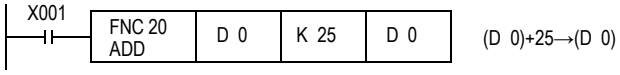
注意要点

1. 使用32位运算(DADD、DADDP)指令时

在字软元件的指定中，是指定低16位一侧的软元件，其后连续编号的软元件则成为高位侧。
为了编号不重复，建议指定软元件为偶数编号。

2. 源操作数和目标操作数中指定为同一软元件

源操作数和目标操作数也可以指定为同一个软元件的编号。
这种情况下，如使用连续执行型的指令(ADD、DADD)，则每个运算周期加法运算的结果都会变化，请注意。



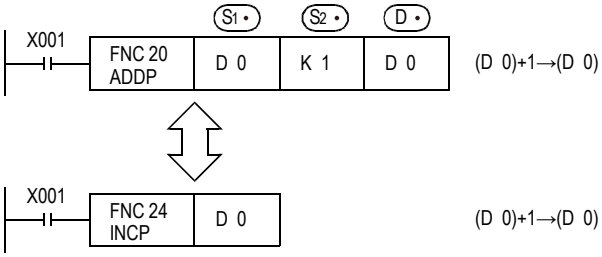
程序举例

1. 采用+1加法运算程序的ADD指令和INC指令的区别

ADD[P]，就是每次X001从OFF变为ON时，D0的内容上加一运算。

这个指令虽然与后述的INCP指令很类似，但是有下面一些内容上的不同。

			ADD/ADDP/DADD/DADDP指令	INC/INCP/DINC/DINCP指令
标志位(零、借位、进位)			动作	不动作
运算结果	16位运算	(S·)+(+1)=(D·)	+32,767→0→+1→+2→	+32,767→-32,768→-32,767
		(S·)+(-1)=(D·)	←-2←-1←0←-32,768	—
	32位运算	(S·)+(+1)=(D·)	+2,147,483,647→0→+1→+2→	+2,147,483,647→-2,147,483,648→-2,147,483,647
		(S·)+(-1)=(D·)	←-2←-1←0←-2,147,483,648	—



10.2 FNC 21 - SUB/BIN减法运算

A1 A2 A2C A8 A8C

概要

2个值进行减法运算(A－B=C)后得出结果的指令。

→ 有关浮点数减法运算[ESUB(FNC 121)指令]，参考18.9节

1. 指令格式

	FNC 21 SUB	
D		P

SUBTRACTION

16位指令	指令记号	执行条件
7步	SUB	连续执行型
	SUBP	脉冲执行型

32位指令	指令记号	执行条件
13步	DSUB	连续执行型
	DSUBP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	减法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(S2・)	减法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(D・)	保存减法运算结果的字软元件编号	BIN16/32位

3. 对象软元件

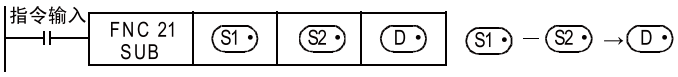
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(SUB、SUBP)

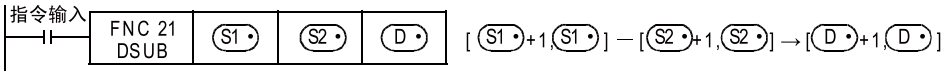
将(S1・)和(S2・)的内容进行二进制减法运算后传送到(D・)中。



- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行减法运算。
  (5-(-8)=13)
- (S1・)和(S2・)中指定常数(K)时，会自动进行BIN转换。

2. 32位运算(DSUB,DSUBP)

将[(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]的内容进行二进制减法运算后传送到[(D・)+1，(D・)]中。



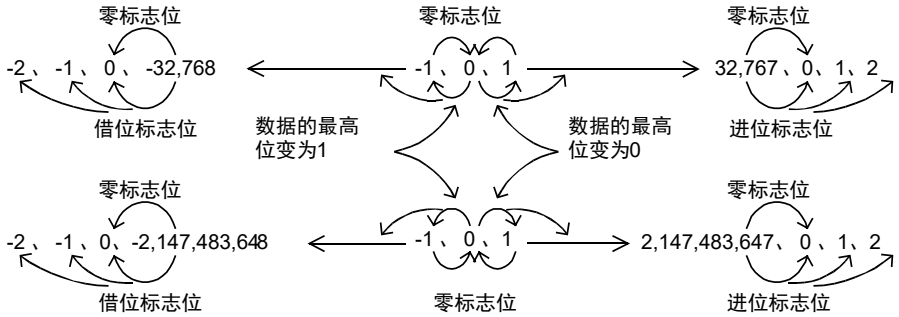
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行减法运算。
  (5,500-(-8,540)=14,040)
- [(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]中指定常数(K)时，会自动进行BIN转换。

相关软元件

1. 标志位的动作及数值的正负的关系

→ 有关标志位的动作，参考6.5.2项

软元件	名称	内容
M8020	零位	ON：运算结果为0时 OFF: 运算结果为0以外时
M8021	借位	ON：运算结果小于-32,768(16位运算)或是 -2,147,483,648(32位运算)时，借位标志位动作。 OFF: 运算结果不小于-32,768(16位运算)或是-2,147,483,648(32位运算)时
M8022	进位	ON： 运算结果大于32,767(16位运算)或者 2,147,483,647(32位运算)时，进位标志位动作。 OFF：运算结果不大于32,767(16位运算)或者2,147,483,647(32位运算)时



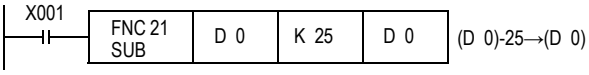
注意要点

1. 使用32位运算(DSUB,DSUBP)指令时

在字软元件的指定中，是指定低16位一侧的软元件，其后连续编号的软元件则成为高位侧。
为了编号不重复，建议指定软元件为偶数编号。

2. 源操作数和目标操作数中指定为同一软元件

源操作数和目标操作数也可以指定为同一个软元件的编号。
这种情况下，使用连续执行型的指令(SUB、DSUB)后，每个运算周期减法运算的结果都会变化，请注意。



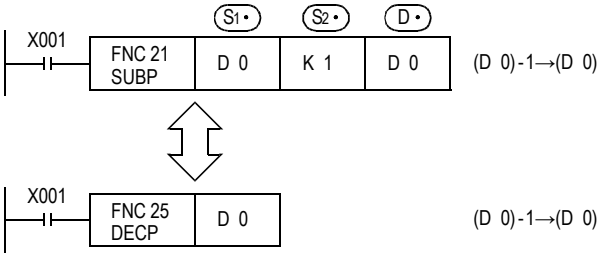
程序举例

1. 采用-1减法运算程序的SUB指令和DEC指令的区别

每次X001从OFF变为ON时，D0的内容上减一运算。

这个指令虽然与后述的DECP指令很类似，但是有下面一些内容上的不同。

			SUB/SUBP/DSUB/DSUBP指令	DEC/DECP/DDEC/DDECP指令
标志位(零、借位、进位)			动作	不动作
运算结果	16 位运算	(S・)-(+1)=(D・)	←-2←-1←0←-32,768	-32,768→+32,767→+32,766
		(S・)-(-1)=(D・)	+32,767→0→+1→+2→	—
	32 位运算	(S・)-(+1)=(D・)	←-2←-1←0←-2,147,483,648	-2,147,483,648→+2,147,483,647→ +2,147,483,646
		(S・)-(-1)=(D・)	+2,147,483,647→0→+1→+2→	—



10.3 FNC 22 - MUL/BIN乘法运算

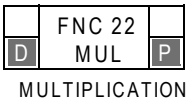
A1 A2 A2C A8 A8C

概要

2个值进行乘法运算(A×B=C)后得出结果的指令。

→ 有关浮点数乘法[EMUL(FNC 122)指令]，参考18.10节

1. 指令格式



16位指令	指令记号	执行条件
7步	MUL	连续执行型
	MULP	脉冲执行型

32位指令	指令记号	执行条件
13步	DMUL	连续执行型
	DMULP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	乘法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(S2·)	乘法运算的数据，或是保存数据的字软元件编号	BIN16/32位
(D·)	保存乘法运算结果的起始字软元件编号	BIN32/64位

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	▲1		●	●	●	●			
(S2·)								●	●	●	●	●	●	●	●	▲1		●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲1		▲2	●					

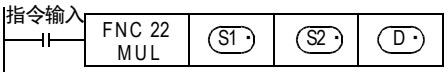
▲1 ：仅对应A8/A8C可编程控制器。

▲2 ：仅16位运算时可以/32位运算时不可以

功能和动作说明

1. 16位运算(MUL,MULP)

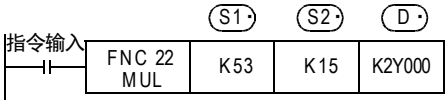
将(S1·)和(S2·)的内容进行二进制乘法运算后传送到[(D·)+1，(D·)]的32位(双字)中。

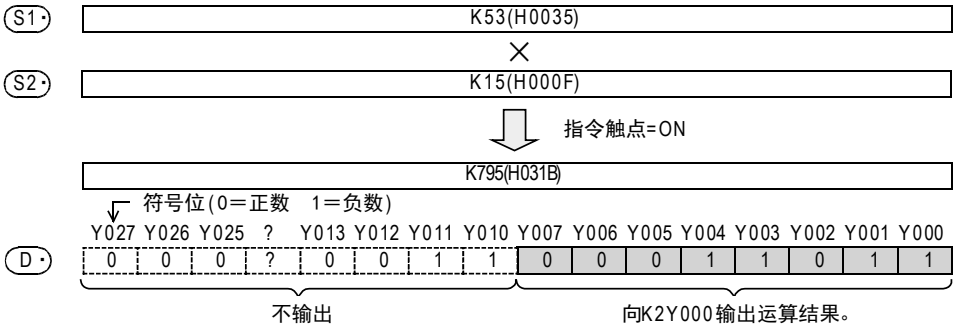


BIN (S1·) 16位 × BIN (S2·) 16位 → BIN (D·)+1,(D·) 32位

- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行乘法运算。
  5× (-8)=-40
- (S1·)和(S2·)中指定常数(K)时，会自动进行BIN转换。

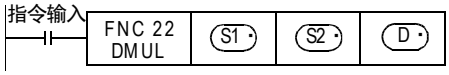
• [(D・)+1，(D・)]指定位数(K1～8)的情况
可以指定K1～K8的位数。
例如，指定K2时，只能得到乘积(32位)中的低8位。

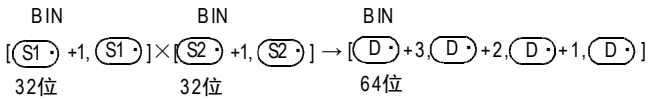




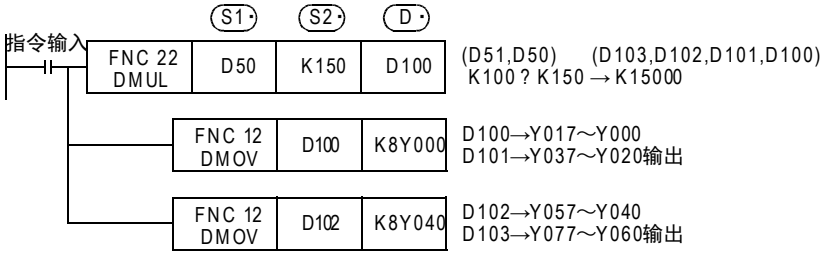
2. 32位运算(DMUL,DMULP)

将 [(S1・)+1，(S1・)] 和 [(S2・)+1，(S2・)] 的内容进行二进制乘法运算后传送到 [(D・)+3，(D・)+2，(D・)+1，(D・)]的64位(字软元件×4)中。





• 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行乘法运算。
5,500×(-8,540)=-46,970,000
• [(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]中指定常数(K)时，会自动进行BIN转换。
• [(D・)+3，(D・)+2，(D・)+1，(D・)]中指定位数(K1～8)时，
只能获得低32位的结果，得不到高32位的结果。
请先传送一次到字软元件中后，再执行运算。



相关软元件

1. 标志位的动作和数值的关系

软元件	名称	内容
M8304	零位	ON: 运算结果为0时 OFF: 运算结果为0以外时

注意要点

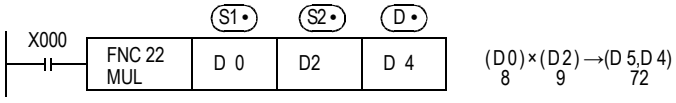
1. (D·) 的指定软元件
   - 使用32位运算(DMUL、DMULP)时，不能指定Z。

2. 通过编程工具监控运算结果时
   即使使用字软元件时，也不可以对作为运算结果的64位数据进行成批监控。
   这种情况下，建议执行浮点数运算。

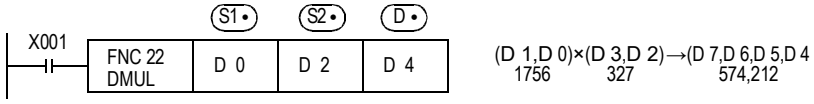
→ 有关浮点数运算，参考18章

程序举例

1. 16位运算



2. 32位运算



根据版本不同的功能变更

对应版本		项目	功能概要
A8	A8C		
以上	以上	零标志位	MUL指令的运算结果为0时，特殊软元件M8304为ON。

10.4 FNC 23 - DIV/BIN除法运算

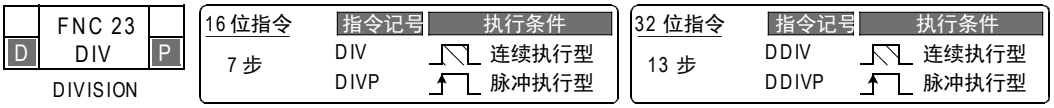
A1 A2 A2C A8 A8C

概要

2个值进行除法运算[A÷B=C…(余数)]后得出结果的指令。

→ 有关浮点数除法运算[EDIV(FNC 123)指令]，参考18.11节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1・)	除法运算的数据，或是保存数据的字软元件编号(被除数)	BIN16/32位
(S2・)	除法运算的数据，或是保存数据的字软元件编号(除数)	BIN16/32位
(D・)	保存除法运算结果的起始字软元件编号(商、余数)	BIN32/64位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲1		●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲1		●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲1		▲2	●					

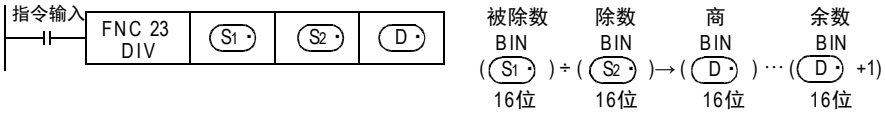
▲1 ：仅对应A8/A8C可编程控制器。

▲2 ：仅16位运算时可以/32位运算时不可以

功能和动作说明

1. 16位运算(DIV、DIVP)

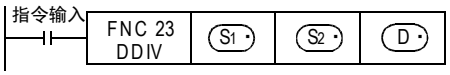
(S1・)的内容作为被除数，(S2・)的内容作为除数，商传到(D・)中，余数传到(D・)+1中。

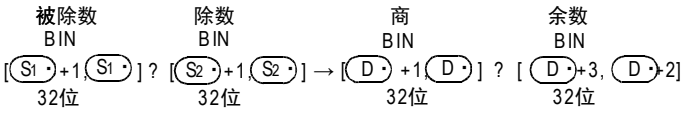


- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行除法运算。(36÷(-5)= - 7(商)，1(余数))]
- 运算结果(商，余数)，会占用指定(D・)开始合计2点的软元件，所以请注意不能与其他控制重复。
- (S1・)和(S2・)中指定常数(K)时，会自动进行BIN转换。

2. 32位运算(DDIV、DDIVP)

[(S1·)+1,(S1·)]的内容作为被除数，[(S2·)+1，(S2·)])的内容作为除数，商传到[(D·)+1，(D·)]，余数传到[(D·)+3，(D·)+2]中。





- 运算结果(商，余数)，会占用指定(D·)开始的软元件，所以请注意不能与其他控制重复。
- 各数据的最高位为正(0)、负(1)的符号位，这些数据以代数方式进行除法运算。
  (5,500÷(-540)= - 10(商)，100(余数))]。
- [(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定常数(K)时，会自动进行BIN转换。

相关软元件

软元件	名称	内容
M8304*1	零位	ON ：运算结果为0时 OFF：运算结果为0以外时
M8306*1	进位	ON：运算结果超过32,767(16位运算)或者2,147,483,647(32位运算)时，进位标志位动作。 OFF: 运算结果为32,767(16位运算)或者2,147,483,647(32位运算)以下时

*1. A8/A8C可编程控制器需要Ver.2.30 以上的版本才能对应。

注意要点

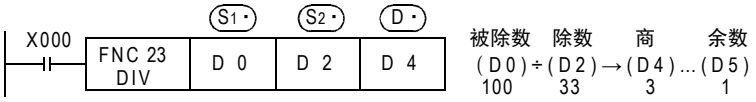
1. 有关运算结果
   - 商和余数的最高位显示位正(0)、负(1)的符号。
   - 当被除数或除数其一为负时，商为负。
     当被除数为负时，余数为负。
2. (D·)的指定软元件
   - 通过指定位数来指定位软元件时，不能得出余数。
   - 使用32位运算(DDIV、DDIVP)时，不能指定Z。

错误

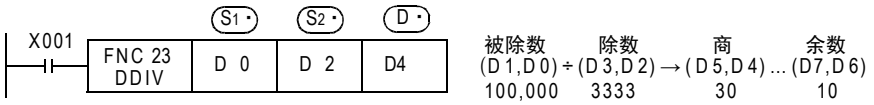
- 除数(S2·)为会发生运算错误，并且不能执行指令。
- 运算结果超过32,767(16位运算)或者2,147,483,647(32位运算)时，出现运行错误。(进位标志位也为ON)

程序举例

1. 16位运算



2. 32位运算



根据版本不同的功能变更

对应版本		项目	功能概要
A8	A8C		
Ver.2.30 以上	Ver.2.30 以上	零标志位	DIV指令的运算结果为0时，M8304为ON。
		进位标志位	DIV指令的运算结果溢出时，M8306为ON。 16位运算时: 最大的负数(-32,768)除以-1时 32位运算时: 最大的负数(-2,147,483,648)除以-1时

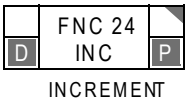
10.5 FNC 24 - INC/BIN加一

A1 A2 A2C A8 A8C

概要

指定的软元件数据中加“1”(+1加法)的指令。

1. 指令格式



16位指令	指令记号	执行条件
3步	INC	连续执行型
	INCP	脉冲执行型

32位指令	指令记号	执行条件
5步	DINC	连续执行型
	DINCP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(D·)	保存被加一数据的字软元件编号	BIN16/32位

3. 对象软元件

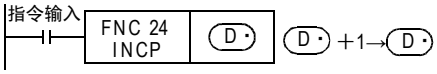
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

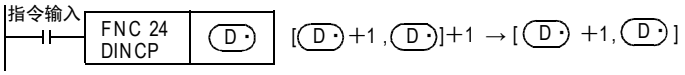
1. 16位运算(INC、INCP)

(D·)的内容加一运算后，传送到(D·)中。



2. 32位运算(DINC、DINCP)

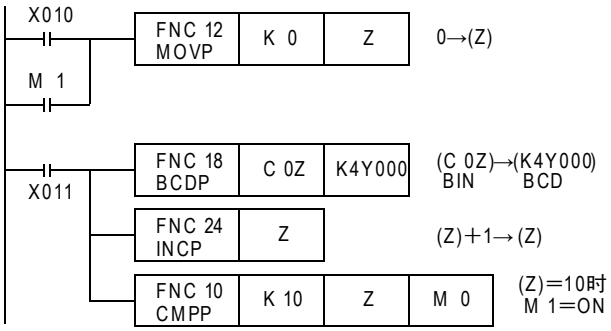
[(D·)+1，(D·)]的内容加一运算后，传送到[(D·)+1，(D·)]中。



注意要点

1. 连续执行型指令中，每个运算周期都执行加一运算，请务必注意。
2. 有关标志位的动作
   1) 16位运算
   +32,767上加1后，变为-32,768，但是标志位(零、借位、进位)不动作。
   2) 32位运算
   +2,147,483,647上加1后，变为-2,147,483,648，但是标志位(零、借位、进位)不动作。

程序举例



通过复位输入X010清除Z

计数器C0～C9的当前值经过BCD转换后输出到K4Y000中。

每当X011为ON时，依次输出C0,C1...C9的当前值

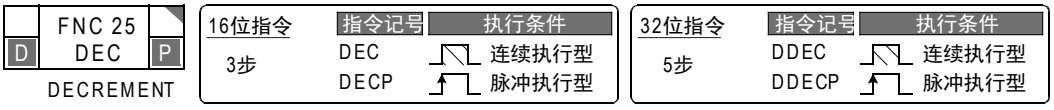
10.6 FNC 25 - DEC/BIN减一

A1 A2 A2C A8 A8C

概要

指定的软元件数据中减“1”(-1加法)的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D·)	保存被减一数据的字软元件编号	BIN16/32位

3. 对象软元件

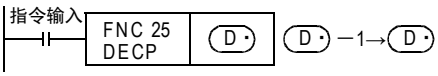
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

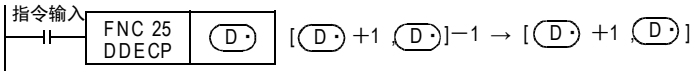
1. 16位运算(DEC、DECP)

(D·)的内容减一运算后，传送到(D·)中。



2. 32位运算(DDEC、DDECP)

[(D·)+1，(D·)]的内容减一运算后，传送到[(D·)+1，(D·)]中。



注意要点

1. 有关标志位的动作

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

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

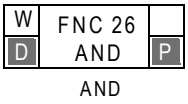
10.7 FNC 26 - WAND/逻辑与

A1 A2 A2C A8 A8C

概要

2个数值进行逻辑与运算的(AND)指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	WAND	连续执行型
	WANDP	脉冲执行型

32位指令	指令记号	执行条件
13步	DAND	连续执行型
	DANDP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	逻辑与数据或保存数据的字软元件编号	BIN16/32位
(S2・)	逻辑与数据或保存数据的字软元件编号	BIN16/32位
(D・)	保存逻辑与结果的字软元件编号	BIN16/32位

3. 对象软元件

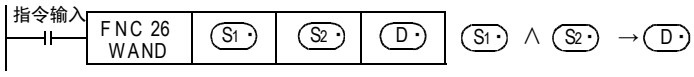
操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(WAND，WANDP)

(S1・)和(S2・)的内容以各位为单位进行逻辑与(AND)运算后，传送到(D・)中。



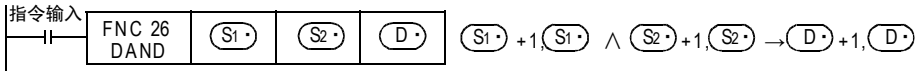
- 指令输入为OFF时，传送目标(D・)的数据不变化。
- 指令输入为ON时，传送源((S1・)，(S2・))的数据不变化。
- 传送源(S1・)和(S2・)中指定常数(K)时，会自动进行BIN转换。
- 逻辑与运算以位为单位，如下表中(1∧1=1　0∧1=0　1∧0=0　0∧0=0)所示变化。

表中: 1=ON，0=OFF

	(S1・)	(S2・)	(D・)
			WAND(FNC 26)指令
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	1	1	1

2. 32位运算(DAND、DANDP)

[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]的内容以各位为单位进行逻辑与(AND)运算后，传送到[(D·)+1，(D·)]中。



- 指令输入为OFF时，传送目标[(D·)+1，(D·)]的数据不变化。
- 指令输入为ON时，传送源[(S1·)+1，(S1·)]、[(S2·)+1，(S2·)]的数据不变化。
- 传送源[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定常数(K)时，会自动进行BIN转换。
- 逻辑与运算以位为单位，如下表中(1∧1=1　0∧1=0　1∧0=0　0∧0=0)所示变化。

表中: 1=ON，0=OFF

	(S1·)+1, (S1·)	(S2·)+1, (S2·)	(D·)+1, (D·)
			DAND(FNC 26)指令
位单位的逻辑运算	0	0	0
	1	0	0
	0	1	0
	1	1	1

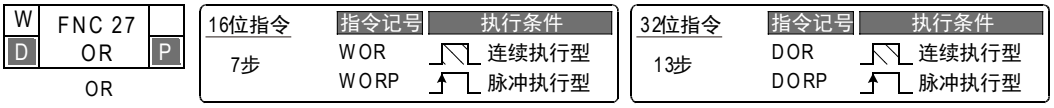
10.8 FNC 27 - WOR/逻辑或

A1 A2 A2C A8 A8C

概要

2个数值进行逻辑或运算的(OR)指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1・)	逻辑或数据或保存数据的字软元件编号	BIN16/32位
(S2・)	逻辑或数据或保存数据的字软元件编号	BIN16/32位
(D・)	保存逻辑或结果的字软元件编号	BIN16/32位

3. 对象软元件

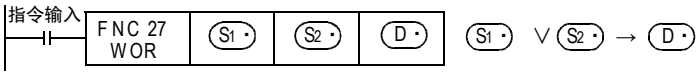
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(WOR、WORP)

(S1・)和(S2・)的内容以各位为单位进行逻辑或(OR)运算后，传送到(D・)中。



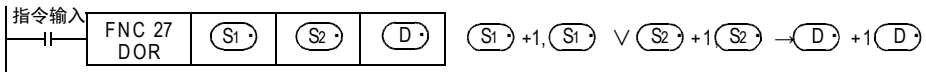
- 指令输入为OFF时，传送目标(D・)的数据不变化。
- 指令输入为ON时，传送源(S1・)、(S2・)的数据不变化。
- 传送源(S1・)和(S2・)中指定常数(K)时，会自动进行BIN转换。
- 逻辑或运算以位为单位，如下表中(1∨1=1　0∨1=1　0∨0=0　1∨0=1)所示变化。

表中: 1=ON，0=OFF

	(S1・)	(S2・)	(D・)
			WOR(FNC 27)指令
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

2. 32位运算(DOR、DORP)

[(S1•)+1，(S1•)]和[(S2•)+1，(S2•)]的内容针对各位进行逻辑或(OR)运算后，传送到[(D•)+1，(D•)]中。



- 指令输入为OFF时，传送目标[(D•)+1，(D•)]的数据不变化。
- 指令输入为ON时，传送源[(S1•)+1，(S1•)]，[(S2•)+1，(S2•)]的数据不变化。
- 传送源[(S1•)+1，(S1•)]和[(S2•)+1，(S2•)]中指定常数(K)时，会自动进行BIN转换。
- 逻辑或运算以位为单位，如下表中(1∨1=1　0∨1=1　0∨0=0　1∨0=1)所示变化。

表中: 1=ON，0=OFF

	(S1•)+1, (S1•)	(S2•)+1, (S2•)	(D•)+1, (D•) DOR(FNC 27)指令
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	1

10.9 FNC 28 - WXOR/逻辑异或

A1 A2 A2C A8 A8C

概要

2个数值进行逻辑异或运算的(XOR)指令。

1. 指令格式

W/D	FNC 28 XOR	P

EXCLUSIVE OR

16位指令	指令记号	执行条件
7步	WXOR	连续执行型
	WXORP	脉冲执行型

32位指令	指令记号	执行条件
13步	DXOR	连续执行型
	DXORP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	要进行逻辑异或的数据，或保存数据的字软元件编号	BIN16/32位
(S2・)	要进行逻辑异或的数据，或保存数据的字软元件编号	BIN16/32位
(D・)	保存逻辑异或结果的字软元件编号	BIN16/32位

3. 对象软元件

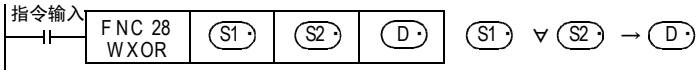
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(WXOR、WXORP)

(S1・) 和 (S2・) 的内容以各位为单位，进行逻辑或(XOR)运算后，传送到 (D・) 中。



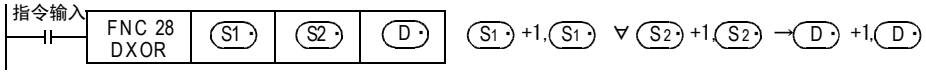
- 指令输入为OFF时，传送目标 (D・) 的数据不变化。
- 指令输入为ON时，传送源((S1・)， (S2・))的数据不变化。
- 传送源 (S1・) 和 (S2・) 中指定常数(K)时，会自动进行BIN转换。
- 逻辑异或运算以位为单位，如下表中(1∀1=0　0∀0=0　1∀0=1　0∀1=1)所示变化。

表中: 1=ON，0=OFF

	(S1・)	(S2・)	(D・)
			WXOR(FNC 28)指令
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	0

2. 32位运算(DXOR、DXORP)

[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]的内容以各位为单位进行逻辑异或(XOR)运算后，传送到[(D·)+1，(D·)]中。



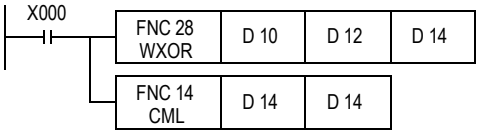
- 指令输入为OFF时，传送目标[(D·)+1，(D·)]的数据不变化。
- 指令输入为ON时，传送源[(S1·)+1，(S1·)]、[(S2·)+1，(S2·)]的数据不变化。
- 传送源[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定常数(K)时，会自动进行BIN转换。
- 逻辑异或运算以位为单位，如下表中(1∀1=0　0∀0=0　1∀0=1　0∀1=1)所示变化。

表中: 1=ON，0=OFF

	(S1·)+1，(S1·)	(S2·)+1，(S2·)	(D·)+1，(D·) DXOR(FNC 28)指令
位单位的逻辑运算	0	0	0
	1	0	1
	0	1	1
	1	1	0

程序举例

WXOR指令和CML(FNC 14)指令组合使用时，还可以执行逻辑异或否(XORNOT)的运算。



10.10 FNC 29 - NEG/补码

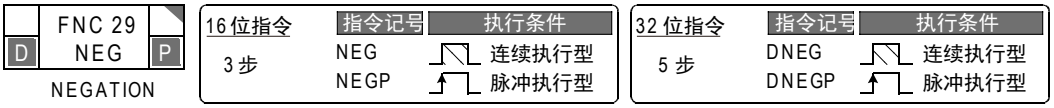
A8 A8C

概要

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

→ 有关浮点数符号反转指令[ENEG(FNC 128)指令]，参考18.16节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(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。)	BIN16/32位

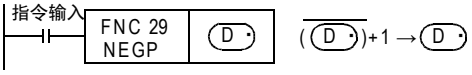
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D・)									●	●	●	●	●	●	●	●	●	●	●					

功能和动作说明

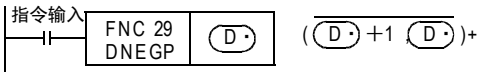
1. 16位运算(NEG、NEGP)

将(D・)内容中的各位反转(0→1、1→0)后加一的结果保存到原先的软元件中。



2. 32位运算(DNEG、DNEGP)

将[(D・)+1，(D・)]内容中的各位反转(0→1、1→0)后加一的结果保存到原先的软元件中。



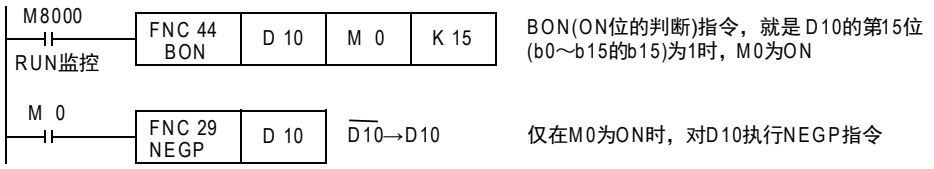
注意要点

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

程序举例

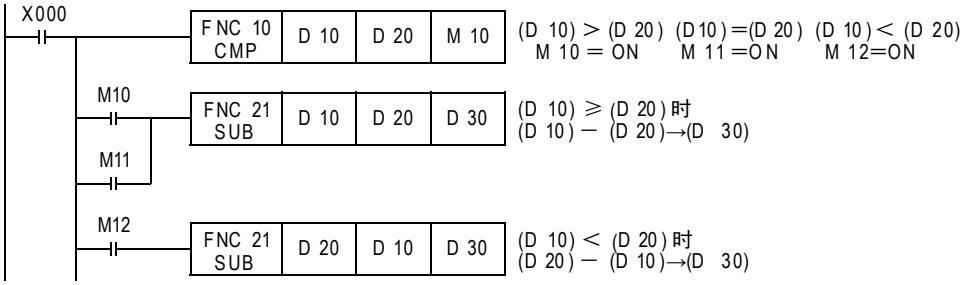
下面介绍针对负的BIN值，求出其绝对值的例子。

1. 使用NEG指令使负数绝对值化



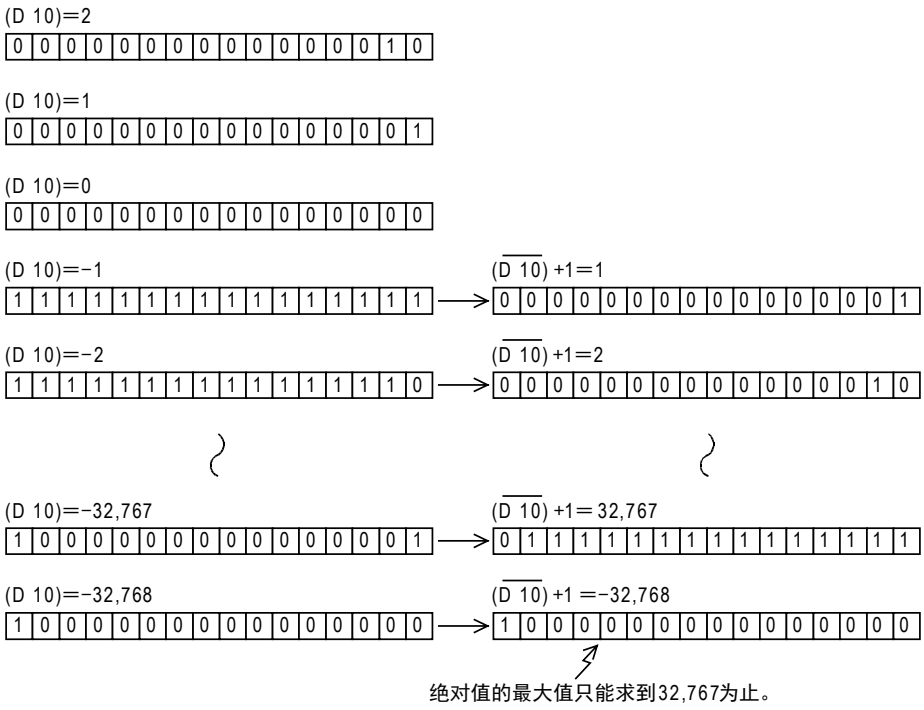
2. 使用SUB(减法运算)指令的绝对值处理

即使不使用NEG指令(补码运算)，(D30)中也会一直保存差的绝对值。



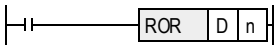
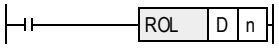
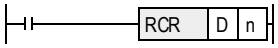
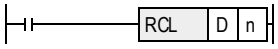
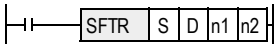
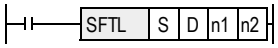
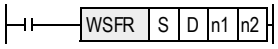
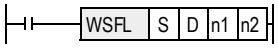
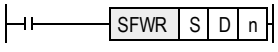
负数的显示及绝对值(参考)

在可编程控制器中，用2的补码显示负数。
最高位为1时为负数，使用补码指令可以求出其绝对值。



11. 循环•移位—FNC 30～FNC 39

FNC 30～FNC 39中，准备了可以使位数据和字数据按指定方向，循环并移位的指令。

FNC No.	指令记号	符号	功能	参考
30	ROR		循环右移	11.1节
31	ROL		循环左移	11.2节
32	RCR		带进位循环右移	11.3节
33	RCL		带进位循环左移	11.4节
34	SFTR		位右移	11.5节
35	SFTL		位左移	11.6节
36	WSFR		字右移	11.7节
37	WSFL		字左移	11.8节
38	SFWR		移位写入[先入先出/先入后出控制用]	11.9节
39	SFRD		移位读出[先入先出控制用]	11.10节

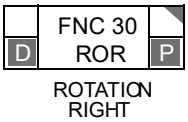
11.1 FNC 30 - ROR/循环右移

A8 A8C

概要

使不包括进位标志在内的指定位数部分的位信息右移、循环的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	ROR	连续执行型
	RORP	脉冲执行型

32位指令	指令记号	执行条件
9步	DROR	连续执行型
	DRORP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(D·)	保存循环右移数据的字软元件编号	BIN16/32位
n	循环移动的位数[n≦16(16位指令)，n≦32(32位指令)][*1]	BIN16/32位

*1. 旋转的位数请不要设定成负值。

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									▲1	▲1	▲1	●	●	●	●	▲2	●	●	●					
n														●	●					●	●			

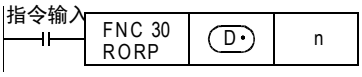
▲1 ：16位运算中，K4Y○○○、K4M○○○、K4S○○○有效。
32位运算中，K8Y○○○、K8M○○○、K8S○○○有效。

▲2 ：仅对应A8/A8C可编程控制器。

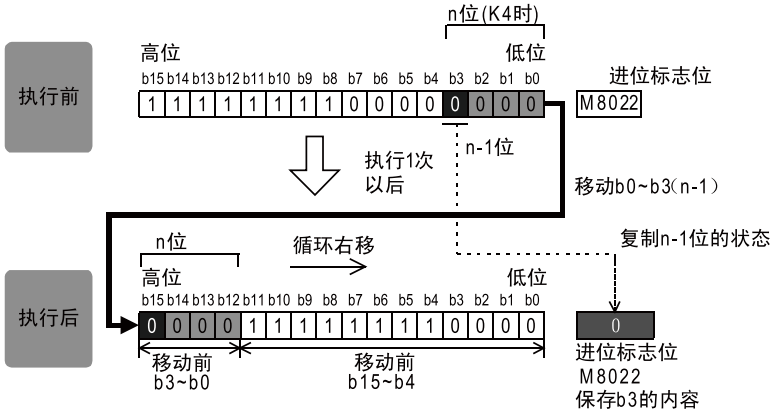
功能和动作说明

1. 16位运算(ROR、RORP)

(D·) 的16位中的n位循环右移。

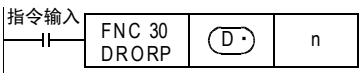


- 最后的位保存在进位标志位(M8022)中。
- 位数指定软元件的情况下，K4(16位指令)有效。

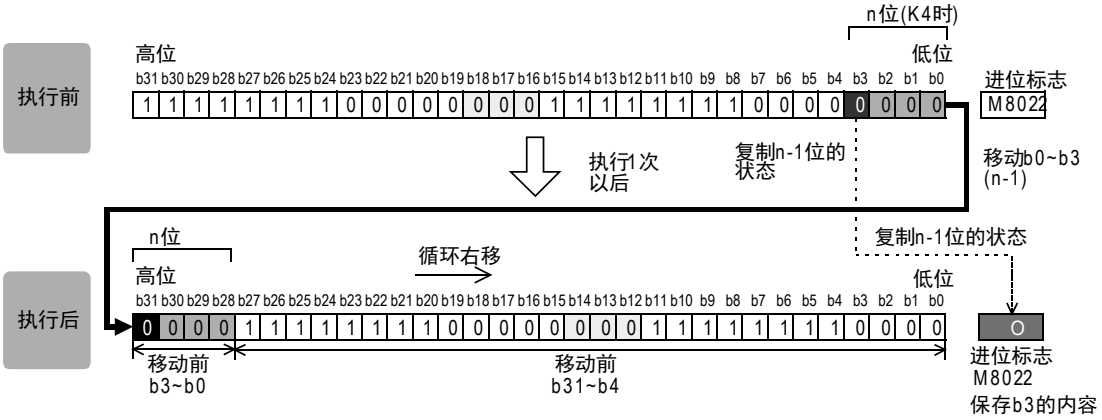


2. 32位运算(DROR、DRORP)

[(D・)+1， (D・)]的32位中的n位循环右移。



- 最后的位保存在进位标志位(M8022)中。
- 位数指定软元件的情况下，K8(32位指令)有效。



相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	最后从最低位移出的位为1时为ON。

注意要点

- 连续执行型(ROR、DROR)指令的场合
  请注意每个扫描周期(运算周期)都会执行循环移位。
- 在 (D・) 中指定位数指定软元件时
  仅K4(16位指令)或K8(32位指令)有效。(例如K4Y010、K8M0)
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DROR D100 R0时，则n=[R1, R0] 。

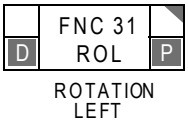
11.2 FNC 31 - ROL/循环左移

A8 A8C

概要

使不包括进位标志位在内的指定位数部分的位信息左移、循环的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	ROL	连续执行型
	ROLP	脉冲执行型

32位指令	指令记号	执行条件
9步	DROL	连续执行型
	DROLP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(D·)	保存循环左移数据的字软元件编号	BIN16/32位
n	循环移动的位数[n≦16(16位指令)，n≦32(32位指令)][*1]	BIN16/32位

*1. 旋转的位数请不要设定成负值。

3. 对象软元件

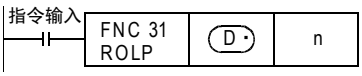
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									▲1	▲1	▲1	●	●	●	●	▲2	●	●	●					
n														●	●					●	●			

▲1 ：16位运算中，K4Y○○○、K4M○○○、K4S○○○有效。
32位运算中，K8Y○○○、K8M○○○、K8S○○○有效。
▲2 ：仅对应A8/A8C可编程控制器。

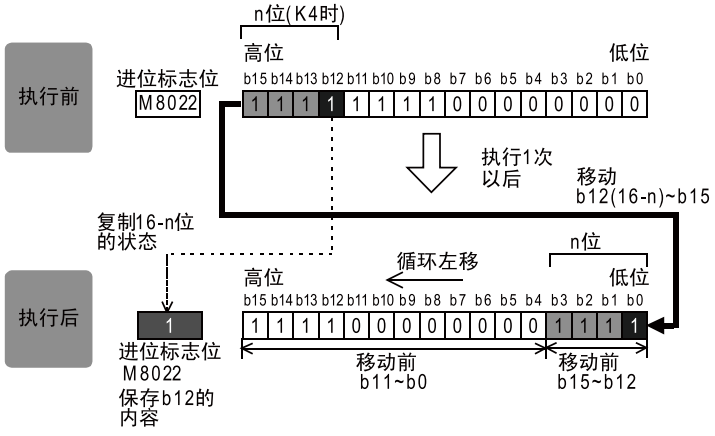
功能和动作说明

1. 16位运算(ROL、ROLP)

(D·)的16位中的n位循环左移。

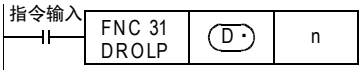


- 最后的位保存在进位标志位(M8022)中。
- 位数指定软元件的情况下，K4(16位指令)有效。

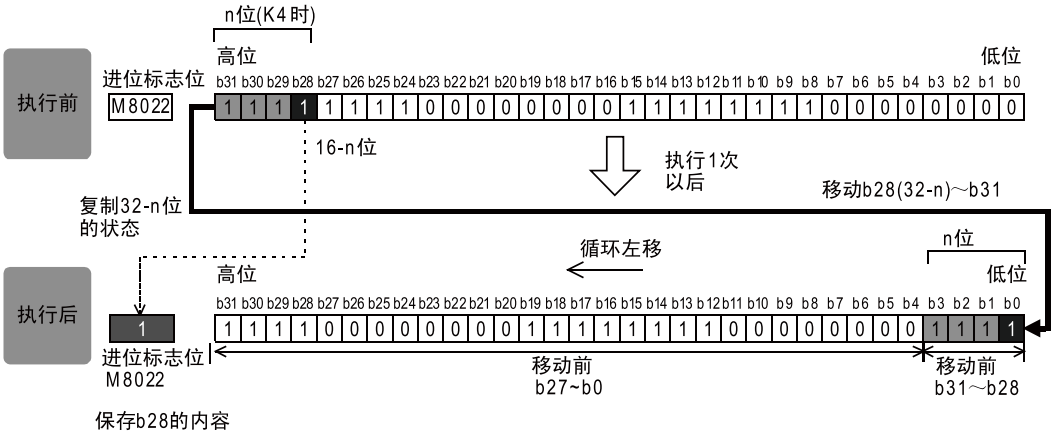


2. 32位运算(DROL、DROLP)

[(D・)+1，(D・)]的32位中的n位循环左移。



- 最后的位保存在进位标志位(M8022)中。
- 位数指定软元件的情况下，K8(32位指令)有效。



相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	最后从最高位移出的位为1时置ON。

注意要点

- 连续执行型(ROL、DROL)指令的场合
  请注意每个扫描周期(运算周期)都会执行循环移位。
- 在 (D・) 中指定位数指定软元件的场合
  仅K4(16位指令)或K8(32位指令)有效。(例如K4Y010、K8M0)
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DROL D100 R0时，则n=[R1, R0] 。

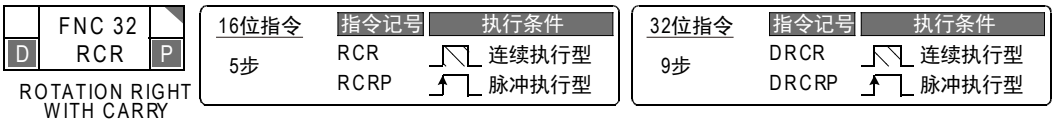
11.3 FNC 32 - RCR/带进位循环右移

A8 A8C

概要

使包括进位标志位在内的指定位数部分的位信息右移、循环的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D·)	保存循环右移数据的字软元件编号	BIN16/32位
n	循环移动的位数[n≦16(16位指令)，n≦32(32位指令)][*1]	BIN16/32位

*1. 旋转的位数请不要设定成负值。

3. 对象软元件

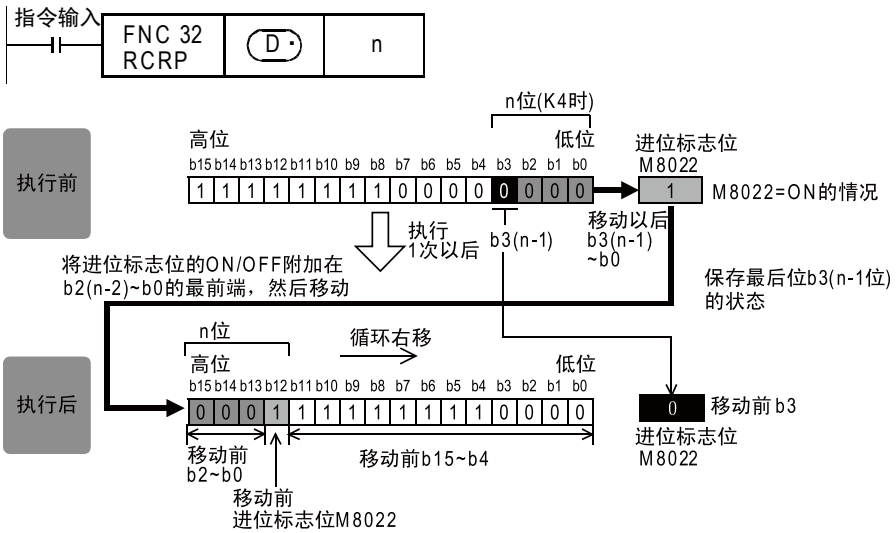
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									▲	▲	▲	●	●	●	●	●	●	●	●					
n														●	●					●	●			

▲: 16位运算中，K4Y○○○、K4M○○○、K4S○○○有效。
32位运算中，K8Y○○○、K8M○○○、K8S○○○有效。

功能和动作说明

1. 16位运算(RCR、RCRP)

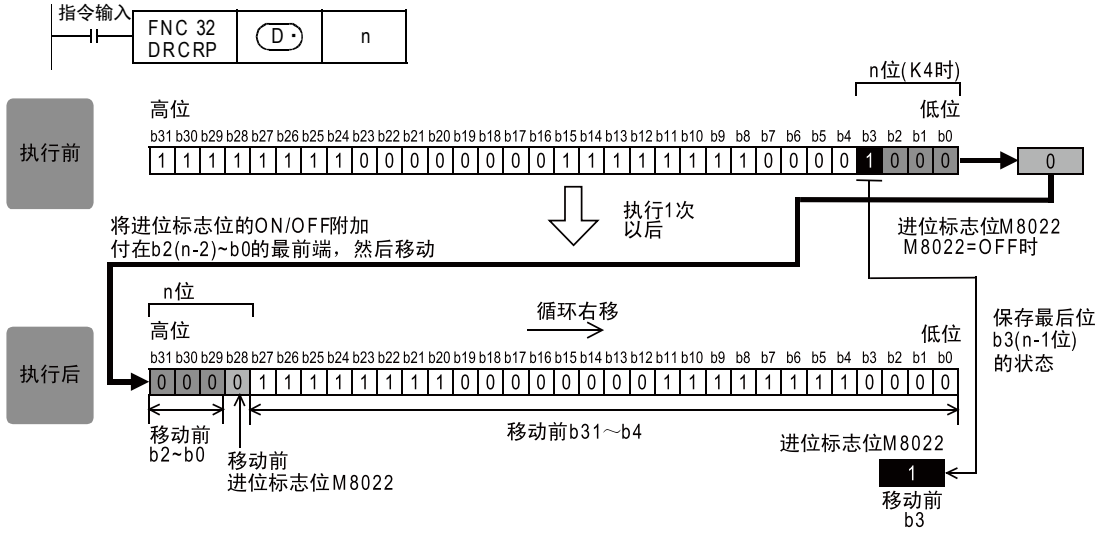
(D·) 的16位+1位(进位标志位M8022)向右移动n位。



因为循环回路中有进位标志位，所以如果执行循环移位指令之前M8022就先ON或OFF，则会被送入目标操作数中。

2. 32位运算(DRCR、DRCRP)

(D・)+1，(D・)的32位+1位(进位标志位M8022)向右移动n位。



相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	最后从最低位移出的位为1时置ON。

注意要点

- 连续执行型(RCR、DRCR)指令的场合
  请注意每个扫描周期(运算周期)都会执行循环移位。
- 在 (D・) 中指定位数指定软元件的场合
  仅K4(16位指令)或K8(32位指令)有效。(例如K4Y010、K8M0)
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DRCR D100 R0时，则n=[R1, R0] 。

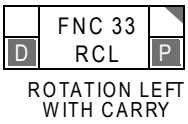
11.4 FNC 33 - RCL/带进位循环左移

A8 A8C

概要

使包括进位标志位在内的指定位数部分的位信息左移、循环的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	RCL	连续执行型
	RCLP	脉冲执行型

32位指令	指令记号	执行条件
9步	DRCL	连续执行型
	DRCLP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(D·)	保存循环左移数据的字软元件编号	BIN16/32位
n	循环移动的位数[n≦16(16位指令)，n≦32(32位指令)][*1]	BIN16/32位

*1. 旋转的位数请不要设定成负值。

3. 对象软元件

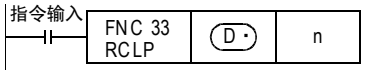
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									▲	▲	▲	●	●	●	●	●	●	●	●					
n														●	●					●	●			

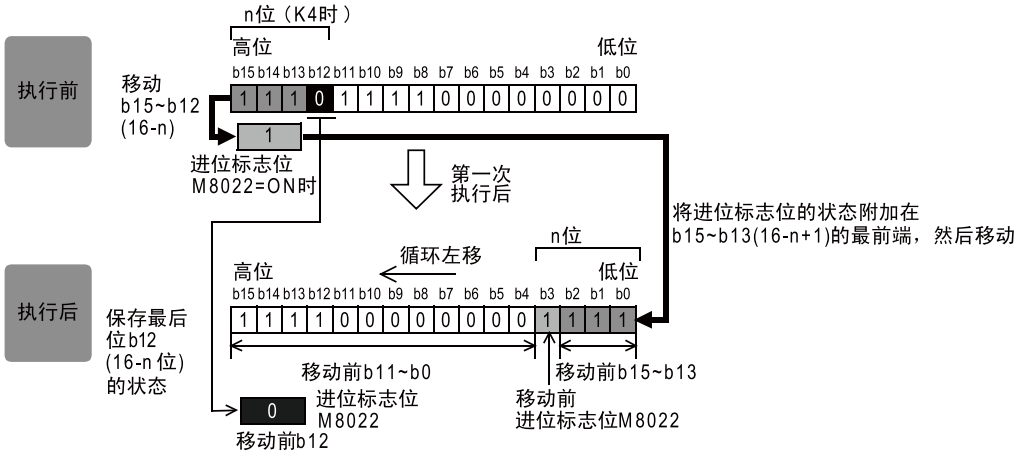
▲: 16位运算中，K4Y○○○、K4M○○○、K4S○○○有效。
32位运算中，K8Y○○○、K8M○○○、K8S○○○有效。

功能和动作说明

1. 16位运算(RCL、RCLP)

(D·)的16位+1位(进位标志位M8022)向左移动n位。

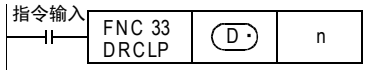


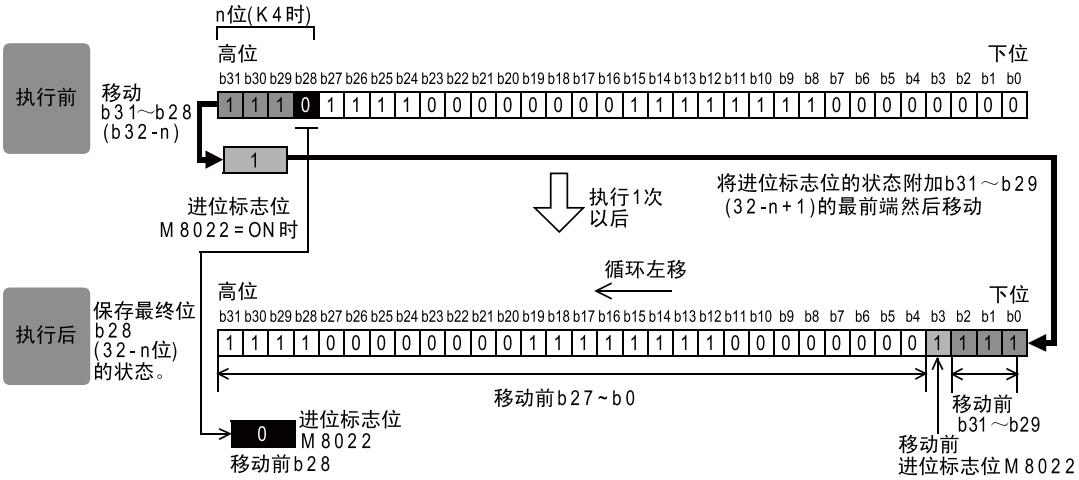


因为循环回路中有进位标志位，所以如果执行循环移位指令之前M8022就先ON或OFF，则会被送入目标操作数中。

2. 32位运算(DRCL、DRCLP)

(D・)+1、(D・)的32位+1位(进位标志位M8022)向左移n位。





相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	最后从最高位移出的位为1时置ON。

注意要点

- 连续执行型(RCL、DRCL)指令的场合
  请注意每个扫描周期(运算周期)都会执行循环移位。
- 在(D・)中指定位数指定软元件的场合
  仅K4(16位指令)或K8(32位指令)有效。(例如K4Y010、K8M0)
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DRCL D100 R0时，则n=[R1, R0] 。

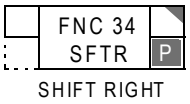
11.5 FNC 34 - SFTR/位右移

A8 A8C

概要

使指定位长度的位软元件每次右移指定的位长度的指令。
移动后，从最高位开始传送n2点长度的(S·)位软元件。

1. 指令格式



16位指令	指令记号	执行条件
9步	SFTR	连续执行型
	SFTRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	右移后在移位数据中保存的起始位软元件编号	位
(D·)	右移的起始位软元件编号	位
n1	移位数据的位数据长度n2≦n1≦1024	BIN16位
n2	右移的位点数n2≦n1≦1024*1	BIN16位

*1. 右移的位点数请不要设定成负值。

3. 对象软元件

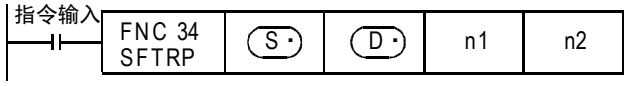
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●	●	●			●	▲												●					
(D·)		●	●			●													●					
n1																				●	●			
n2														●	●					●	●			

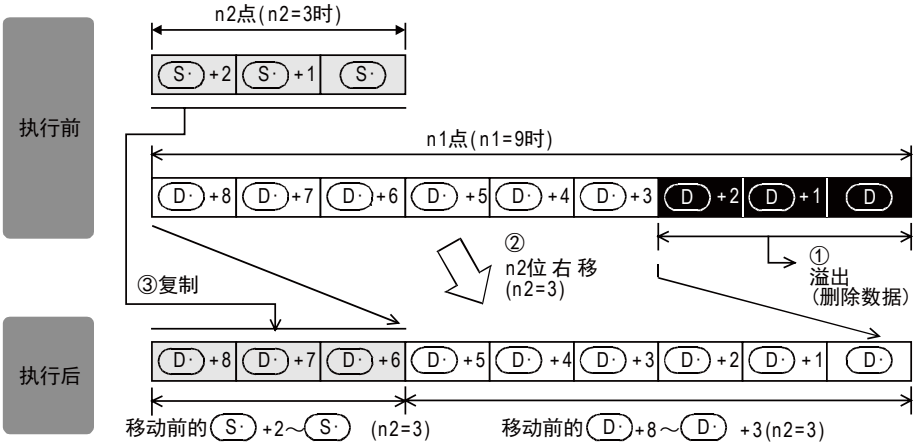
▲:D□.b仅对应A8/A8C可编程控制器。但是不能变址修饰

功能和动作说明

1. 16位运算(SFTR、SFTRP)

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





注意要点

SFTRP指令中，每次当指令输入从OFF变为ON时，执行n2位移位，但是请注意SFTR指令中，每个扫描周期(运算周期)都执行移位。

错误

A8/A8C可编程控制器的情况下，传送源 (S・) 和移位软元件 (D・) 重复时，发生运算错误。
(错误代码: K6710)

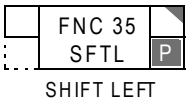
11.6 FNC 35 - SFTL/位左移

A1 A2 A2C A8 A8C

概要

使指定位长度的位软元件每次左移指定的位长度的指令。
移动后，从最低位开始传送n2点长度的(S·)位软元件。

1. 指令格式



16位指令	指令记号	执行条件
9步	SFTL	连续执行型
	SFTLP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	左移后在移位数据中保存的起始位软元件编号	位
(D·)	左移的起始位软元件编号	位
n1	移位数据的位数据长度n2≦n1≦1024	BIN16位
n2	左移的位点数n2≦n1≦1024*1	BIN16位

*1. 左移的位点数请不要设定成负值。

3. 对象软元件

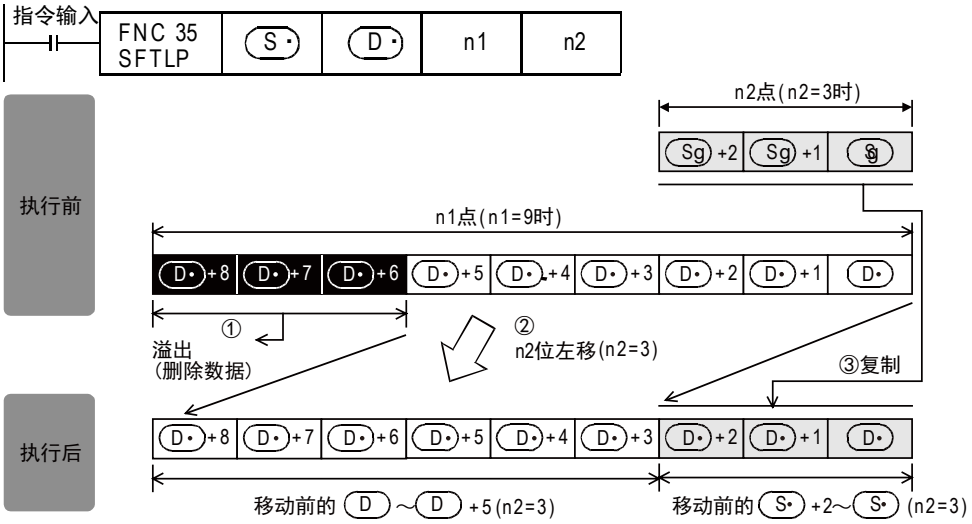
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●	●	●			●	▲												●					
(D·)		●	●			●													●					
n1																				●	●			
n2														●	●					●	●			

▲: D□.b仅对应A8/A8C可编程控制器。但是不能变址修饰(V、Z)

功能和动作说明

1. 16位运算(SFTL、SFTLP)

对于以(D·)起始的n1位(移位寄存器的长度)数据，左移n2位(下记的①、②)。
移位后，将(S·)开始的n2位数据传送到(下面的③)从(D·)开始的n2位中。



注意要点

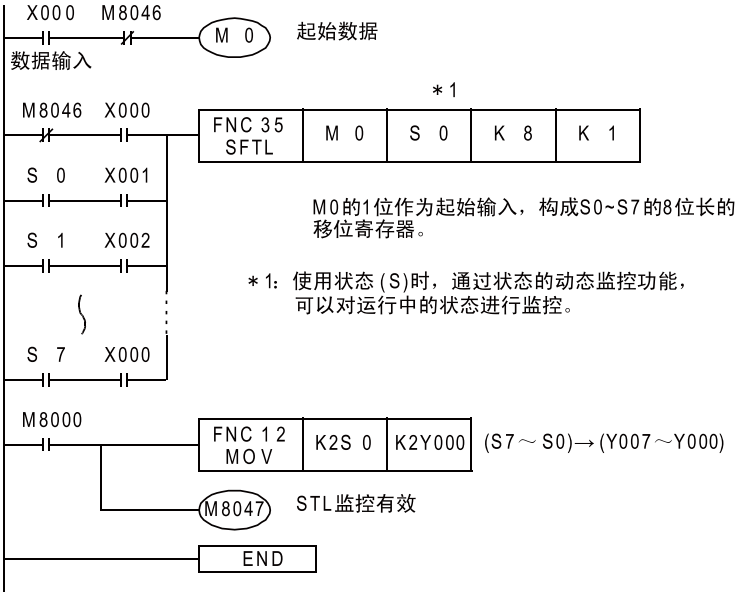
SFTLP指令中，每次当指令输入从OFF变为ON时，执行n2位移位，但是请注意SFTL指令中，每个扫描周期(运算周期)都执行移位。

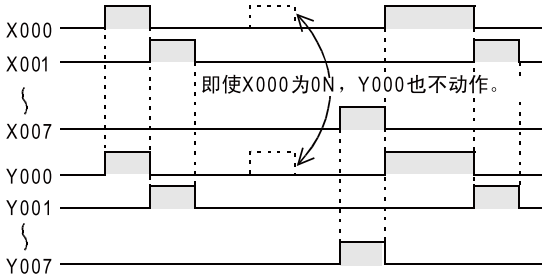
错误

A8/A8C可编程控制器的情况下，传送源 (S・) 和移位软元件 (D・) 重复时，发生运算错误。
(错误代码: K6710)

程序举例(带条件一位数据的步进)

使X000～X007依次置ON，则Y000～Y007也依次动作。
一旦顺序错误，将不会动作。





11.6.1 F1·F2可编程控制器的SFT(移位)指令替换

F1・F2可编程控制器中的SFT指令和A8/A8C可编程控制器中的SFTL(FNC 35)指令的对应关系如下所示。

1. F1・F2可编程控制器

M100: 输入数据

M101～M117(8进制编号): 移位寄存器15段

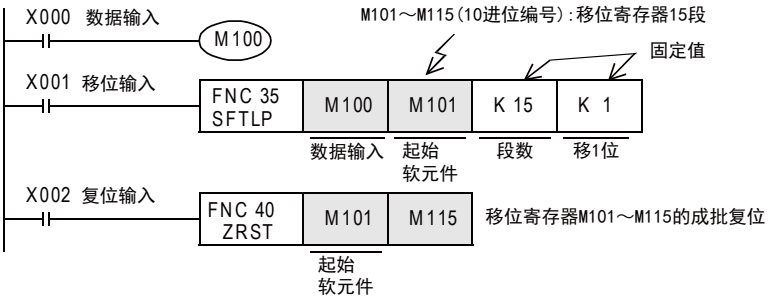
梯形图程序



指令表程序

```
0000 LD   X000
0001 OUT  M100
0002 LD   X001
0003 SFT  M100
0004 LD   X002
0005 RST  M100
```

2. A8/A8C可编程控制器



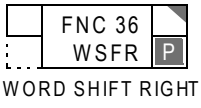
11.7 FNC 36 - WSFR/字右移

A8 A8C

概要

将n1个字长的字软元件右移n2个字的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	WSFR	连续执行型
	WSFRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	右移后在移位数据中保存的起始位软元件编号	BIN16位
(D・)	保存右移数据的起始字软元件编号	BIN16位
n1	移位数据的字数据长度n2≦n1≦512	BIN16位
n2	右移的字点数n2≦n1≦512*1	BIN16位

*1. 右移的字点数请不要设定成负值。

3. 对象软元件

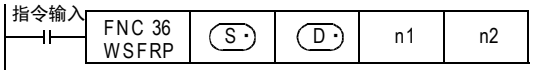
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲			●					
(D・)									●	●	●	●	●	●	●	▲			●					
n1																				●	●			
n2														●	●					●	●			

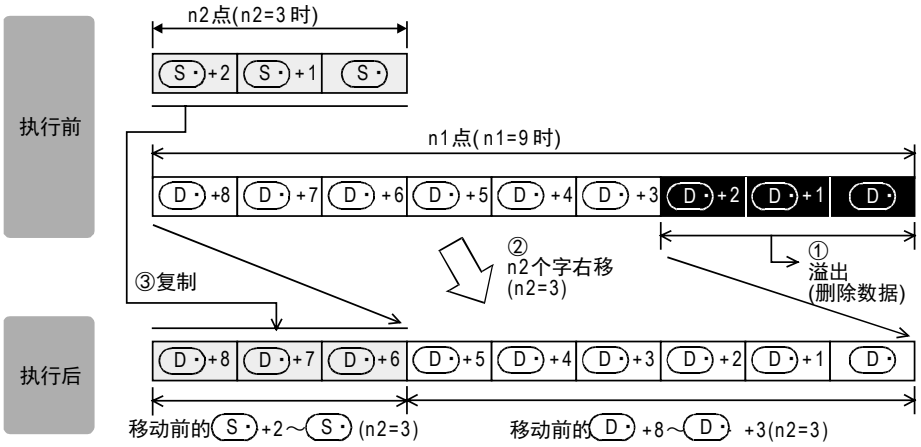
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(WSFR、WSFRP)

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





注意要点

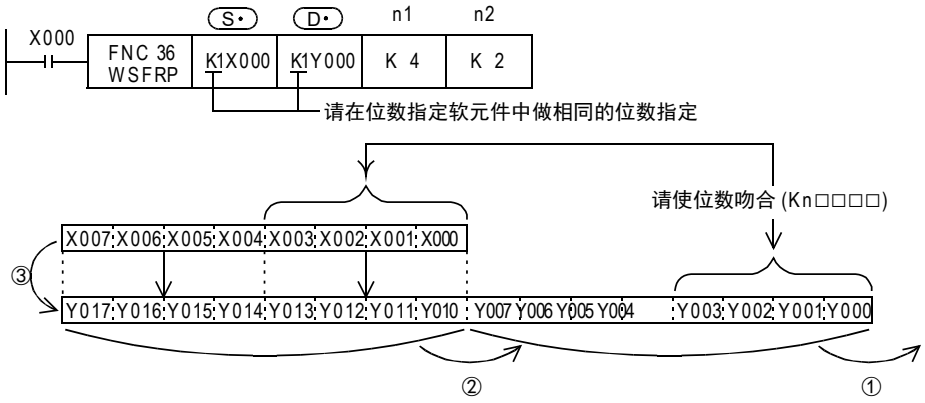
WSFRP指令中驱动输入ON后，移动n2个字，但是在WSFR指令中每个运算周期都会执行移动，请注意。

错误

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

程序举例

1. 位数指定软元件的移位



11.8 FNC 37 - WSFL/字左移

A8 A8C

概要

将字数据信息左移指定字个数的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	WSFL	连续执行型
	WSFLP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	左移后在移位数据中保存的起始软元件编号	BIN16位
(D・)	保存左移数据的起始字软元件编号	BIN16位
n1	移位数据的字数据长度n2≦n1≦512	BIN16位
n2	左移的字点数n2≦n1≦512*1	BIN16位

*1. 左移的字点数请不要设定成负值。

3. 对象软元件

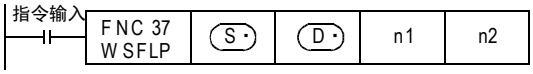
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲			●					
(D・)									●	●	●	●	●	●	●	▲			●					
n1																				●	●			
n2														●	●					●	●			

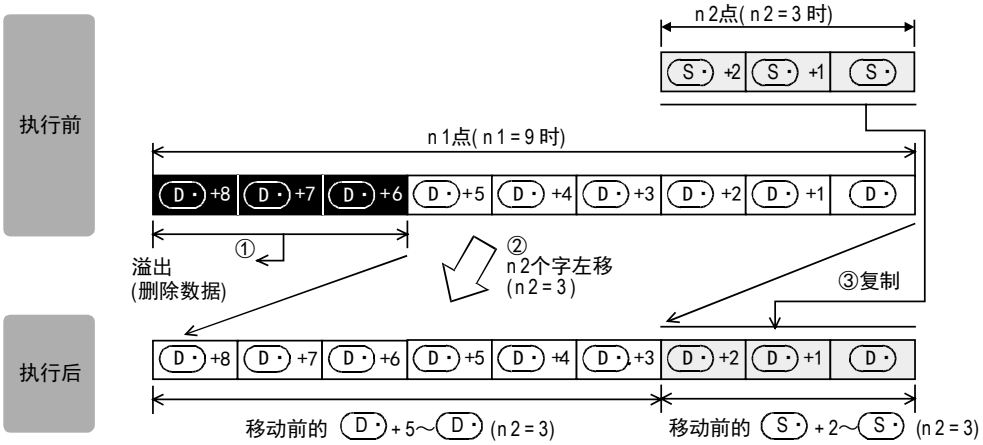
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(WSFL、WSFLP)

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





注意要点

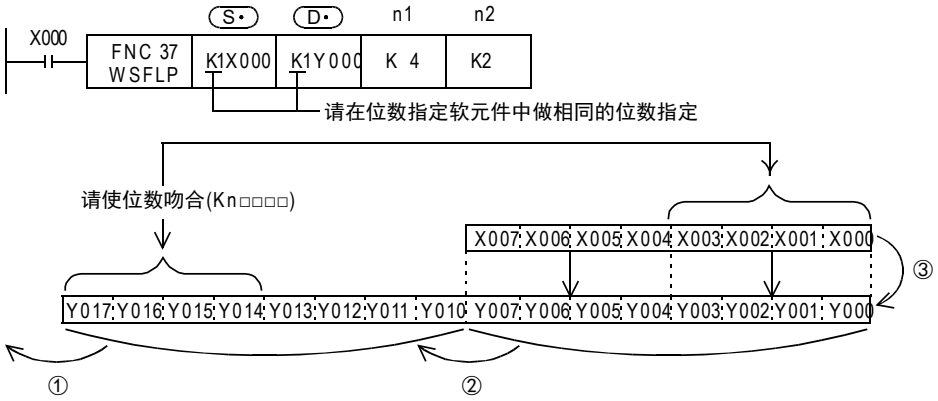
WSFLP指令中，每次当指令输入从OFF变为ON，就执行n2个字的移位，但是请注意WSFL指令中，每个运算周期都执行移位。

错误

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

程序举例

1. 位数指定软元件的移位



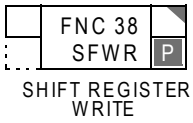
11.9 FNC 38 - SFWR/移位写入[先入先出/先入后出控制用]

A1 A2 A2C A8 A8C

概要

为先入先出和先入后出控制准备的数据写入指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	SFWR	连续执行型
	SFWRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存想先入的数据的字软元件编号	BIN16位
(D・)	保存数据并移位的起始字软元件编号 (最前端为指针，数据是从(D・)+1开始的)	BIN16位
n	请指定被保存的数据的点数+1*1的值 2≦n≦512	BIN16位

*1. +1为指针的部分。

3. 对象软元件

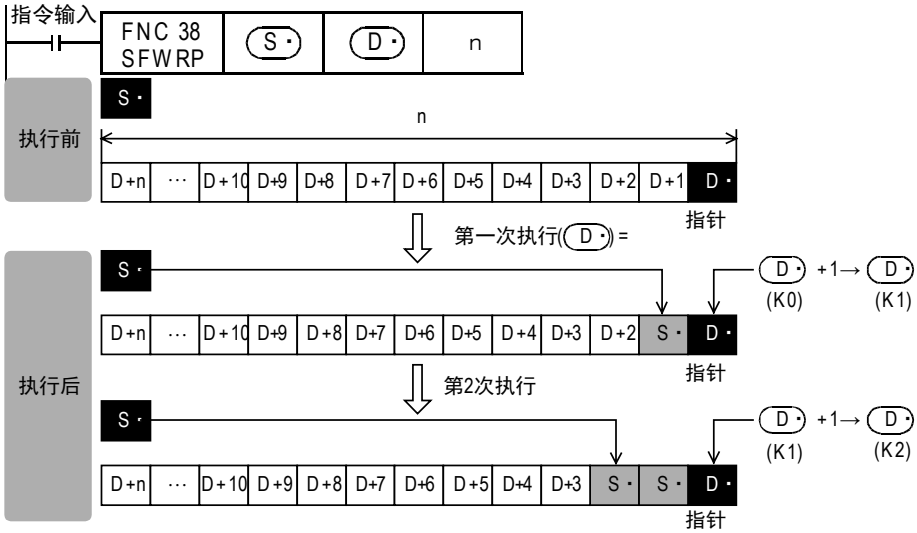
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲			●					
n																				●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(SFWR、SFWRP)

在 (D·) +1开始的n－1点中依次写入 (S·) 的内容，并对 (D·) 中保存的数据数+1。
例如， (D·) =0时，写入 (D·) +1， (D·) =1时，写入 (D·) +2。



1) X000从OFF变为ON时， (S·) 的内容被保存到 (D·) +1中， (D·) +1的内容变为 (S·) 的值。
2) (S·) 的内容变化后再次执行输入从OFF变为ON后， (S·) 的内容被保存到 (D·) +2中，
(D·) +2的内容变为 (S·) 。(由于用连续执行型指令SFWR，每个运算周期都依次被保存，因此请用脉冲执行型指令SFWRP编程。)
3) 以下过程相同，从右端依次执行，在指针 (D·) 的内容中表示数据的保存点数。

相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	指针 (D·) 的内容n超1时，变为无处理(不写入)，且进位标志位 M8022置ON。

相关指令

指令	内容
SFRD(FNC 39)	移位读出[先入先出控制用]
POP(FNC 212)	读取后入的数据[先入后出控制用]

注意要点

- 连续执行型(SFWR)指令的场合
请注意每个扫描周期(运算周期)都会依次保存(改写)。

程序举例

1. 先入先出控制的例子

→ 有关先入后出控制的程序举例，参考27.3节

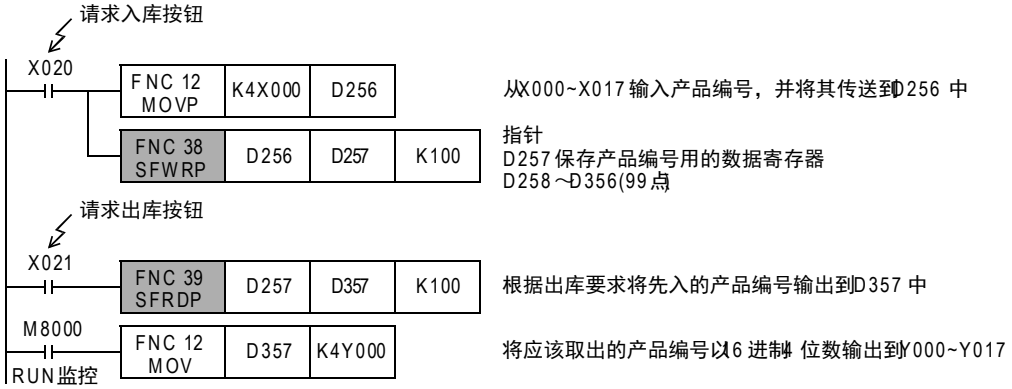
下面例举使用了移位写入(SFWR)、移位读出(SFRD)指令的事例加以说明。

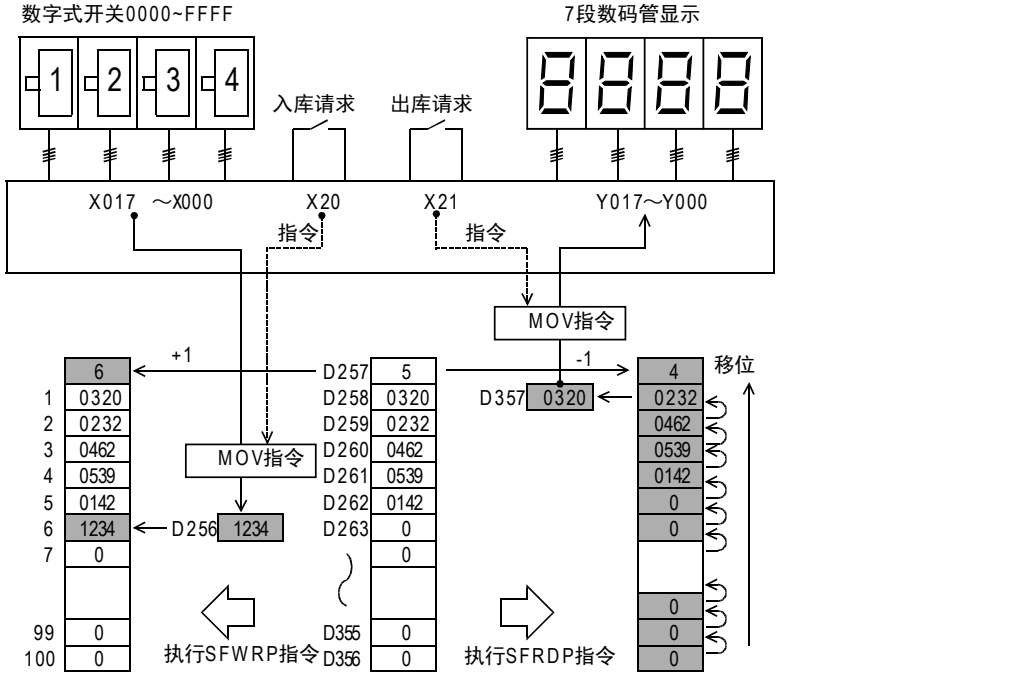
1) 动作内容

一 登记产品编号的同时，为了能实现先入库的物品先出库这样的先入先出原则，以下介绍了输出当前应该取的产品编号的梯形图实例。

一产品编号为4位数以下的16进制数，最大库存量在99点以下。

2) 程序





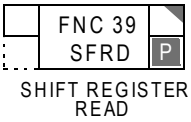
11.10 FNC 39 - SFRD/移位读出[先入先出控制用]

A1 A2 A2C A8 A8C

概要

为先入先出控制准备的数据读出指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	SFRD	连续执行型
	SFRDP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(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)	BIN16位
(D·)	保存先出的数据的字软元件编号	BIN16位
n	请指定被保存的数据的点数+1*1的值。2≦n≦512	BIN16位

*1. +1为指针的部分。

3. 对象软元件

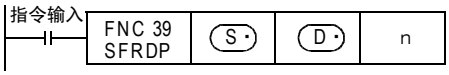
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)									●	●	●	●	●	●	●	▲			●					
(D·)									●	●	●	●	●	●	●	▲	●	●	●					
n																				●	●			

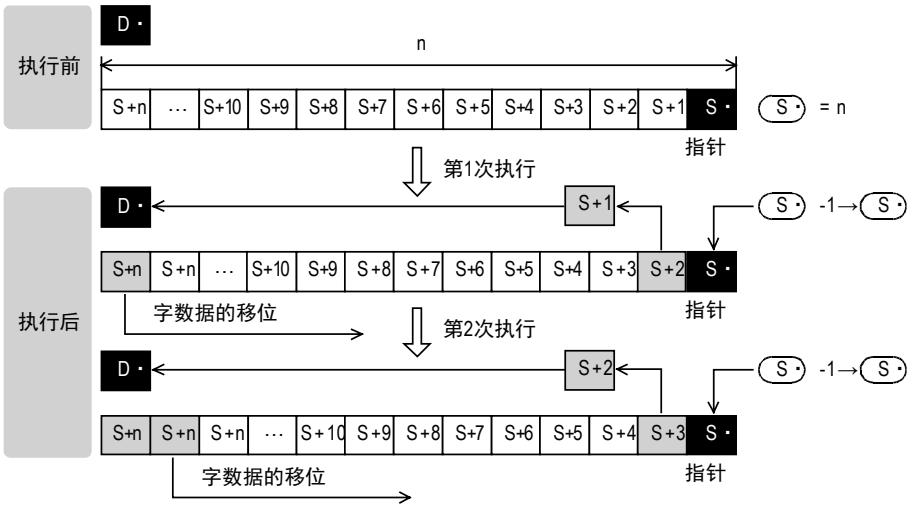
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(SFRD、SFRDP)

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





1) 指令触点为ON时，(S・)+1的内容传送(读出)到(D・)中。

2) 与此同时，指针(S・)的内容减少，左侧的数据逐字右移。
(由于用连续执行型指令SFRD，每个运算周期都移位，因此请用脉冲执行型指令SFRDP编程。)

相关软元件

→ 有关零标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8020	零位	数据的读出，通常从(S・)+1开始执行，但是指针(S・)的内容为0时，零标志位M8020动作。

相关指令

指令	内容
SFWR(FNC 38)	移位写入[先入先出/先入后出控制用]
POP(FNC 212)	读取后入的数据[先入后出控制用]

注意要点

1. 执行读出后的数据
(S・)+n的内容不会因为读出而变化。

2. 连续执行型(SFRD)指令时
每个扫描周期(运算周期)都会执行依次读出，但(S・)+n的内容不会变化。

3. 指针(S・)为0时
为无处理，且(D・)的内容不会变化。

程序举例

请参考SFWR(FNC 38)的程序举例。

→ 有关程序举例，参考11.9节

12. 数据处理—FNC 40～FNC 49

相对于FNC 10～FNC 39的基本应用指令，FNC 40～FNC 49指令能够进行更加复杂的处理，或作为满足特殊用途的指令使用。

FNC No.	指令记号	符号	功能	参考
40	ZRST	ZRST D1 D2	成批复位	12.1节
41	DECO	DECO S D n	译码	12.2节
42	ENCO	ENCO S D n	编码	12.3节
43	SUM	SUM S D	ON位数	12.4节
44	BON	BON S D n	ON位的判定	12.5节
45	MEAN	MEAN S D n	平均值	12.6节
46	ANS	ANS S m D	信号报警器置位	12.7节
47	ANR	ANR	信号报警器复位	12.8节
48	SQR	SQR S D	BIN开方运算	12.9节
49	FLT	FLT S D	BIN整数→2进制浮点数转换	12.10节

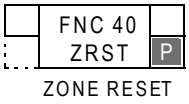
12.1FNC 40 - ZRST/成批复位

A1 A2 A2C A8 A8C

概要

2个指定的软元件之间执行成批复位的指令。
用于在中断运行后从初期开始运行时，以及对控制数据进行复位时。

1. 指令格式



16位指令	指令记号	执行条件
5步	ZRST	连续执行型
	ZRSTP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容		数据类型
(D1・)	成批复位的最前端的位/字软元件编号	(D1・) ≦ (D2・) 指定同一种类的要素	BIN16位
(D2・)	成批复位的末尾的位/字软元件编号		BIN16位

3. 对象软元件

操作数种类	位软元件						字软元件												其他				
	系统・用户						位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D1・)		●	●			●						●	●	●	●	▲			●					
(D2・)		●	●			●						●	●	●	●	▲			●					

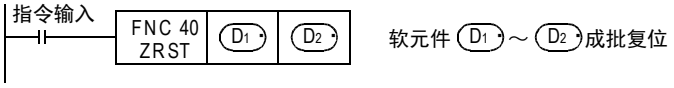
▲: 仅对应A8/A8C可编程控制器。

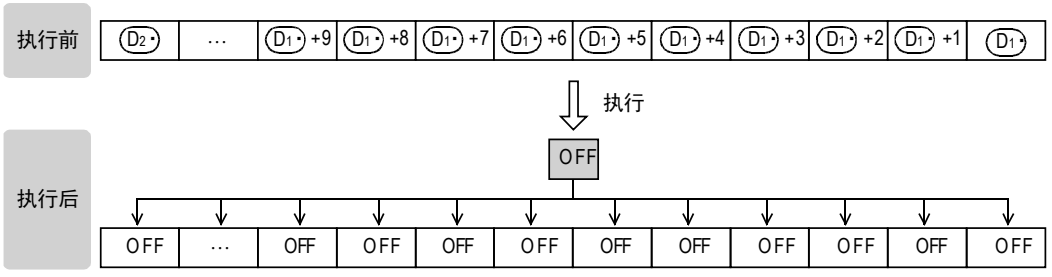
功能和动作说明

1. 16位运算(ZRST、ZRSTP)

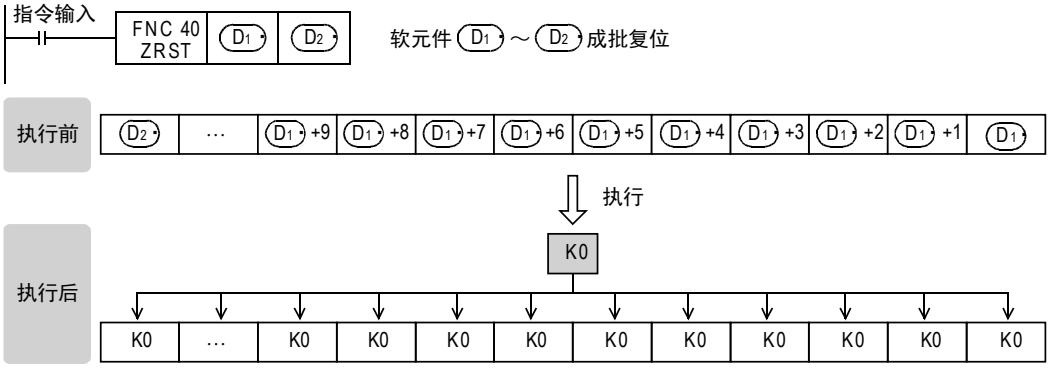
将同一种类的 (D1・) ～ (D2・) 全部复位。

(D1・)，(D2・) 为位软元件时
(D1・) ～ (D2・) 的软元件范围全部被写入OFF(复位)。





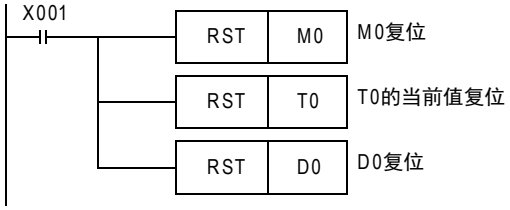
(D1•)，(D2•)为字软元件时
(D1•)～(D2•)的软元件范围全部被写入K0。



相关指令

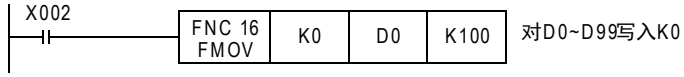
1. RST指令

作为软元件的单独复位指令，可以对位软元件(Y、M、S)和字软元件(T、C、D、R)使用RST指令。



2. FMOV(FNC 16)指令

作为常数(例:K0)的成批写入指令有FMOV(FNC 16)指令，可以在字软元件(KnY、KnM、KnS、T、C、D、R)中写入0。



注意要点

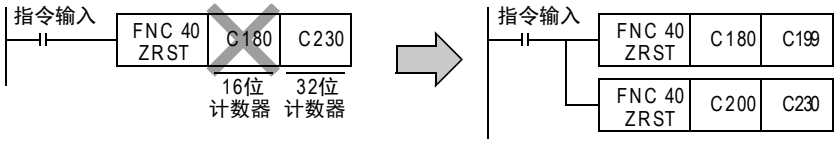
1. 指定软元件时的注意事项

(D1•)，(D2•)指定为同一种类的软元件，且(D1•)编号≦(D2•)编号。
(D1•)编号＞(D2•)编号时，(D1•)中指定的软元件仅仅复位

2. 关于高速计数器(C235～C255)的指定

ZRST指令作为16位处理的指令，但也可以在(D1•)，(D2•)中指定32位计数器。
但是，指定时不允许出现类似(D1•)中指定16位计数器，(D2•)中指定32位计数器的混在的情况。

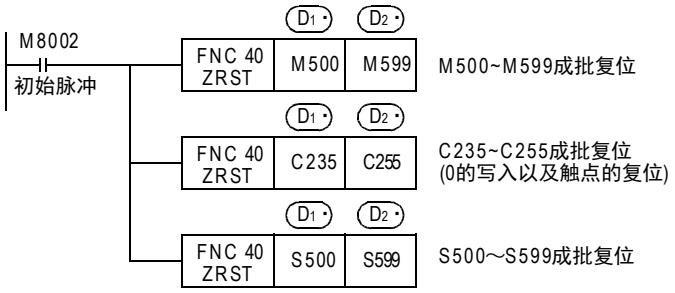
例如



程序举例

1. 将保持区域的软元件作为非保持使用的情况

当可编程控制器的电源为ON和RUN时，对位软元件和字软元件的指定范围执行复位。



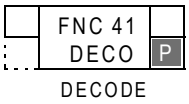
12.2 FNC 41 - DECO/译码

A1 A2 A2C A8 A8C

概要

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

1. 指令格式



16位指令	指令记号	执行条件
7步	DECO	连续执行型
	DECOP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存要译码的数据，或是数据的字软元件编号	BIN16位
(D·)	保存译码结果的位/字软元件编号	BIN16位
n	保存译码结果的软元件的位点数(n=1～8) (n=0时为不处理)	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●	●	●			●						●	●	●	●	▲	●	●	●	●	●			
(D·)		●	●			●						●	●	●	●	▲			●					
n																				●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(DECO、DECOP)

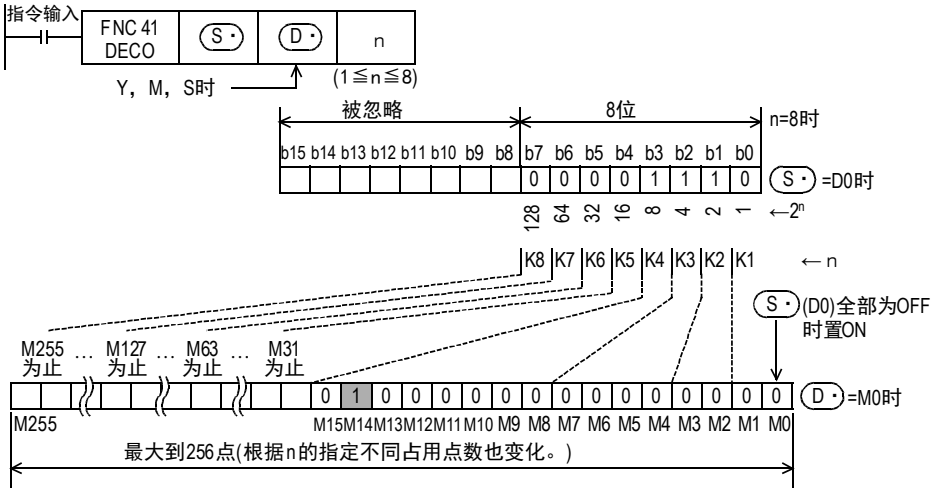
(S・) 的值相对应的 (D・) ～ (D・) +2^n-1中的1个置ON。

1) (D・) 为位软元件时(1≦n≦8)

(S・) 中指定的软元件的n位数(1≦n≦8)，在 (D・) 中被译码。

— (S・) 都为0时， 位软元件 (D・) 为ON。

— n=8时，最大2^8 =256点。

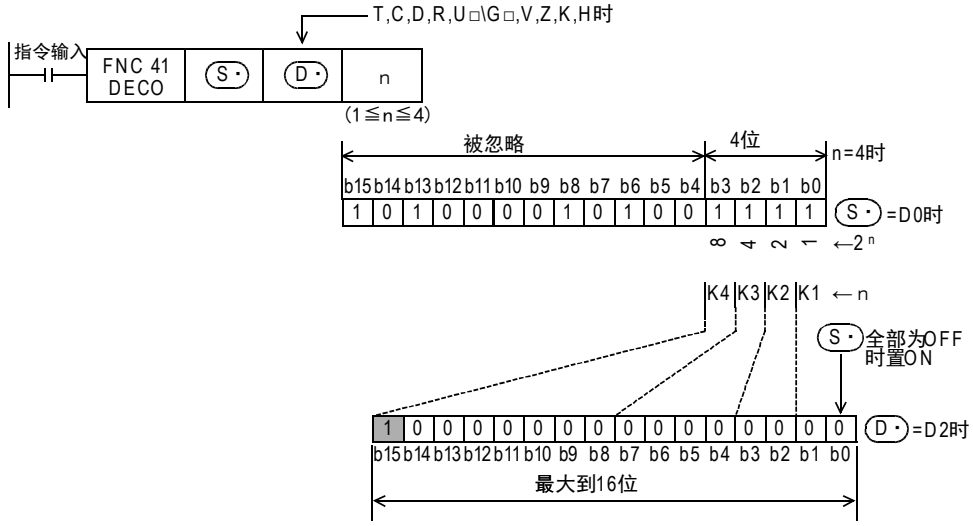


2) (D・) 为字软元件时(1≦n≦4)

(S・) 的低n位，在 (D・) 中被译码。

— (S・) 都为0时， 字软元件 (D・) 的b0为ON。

— n≦3时 (D・) 的高字节都为0(OFF)。



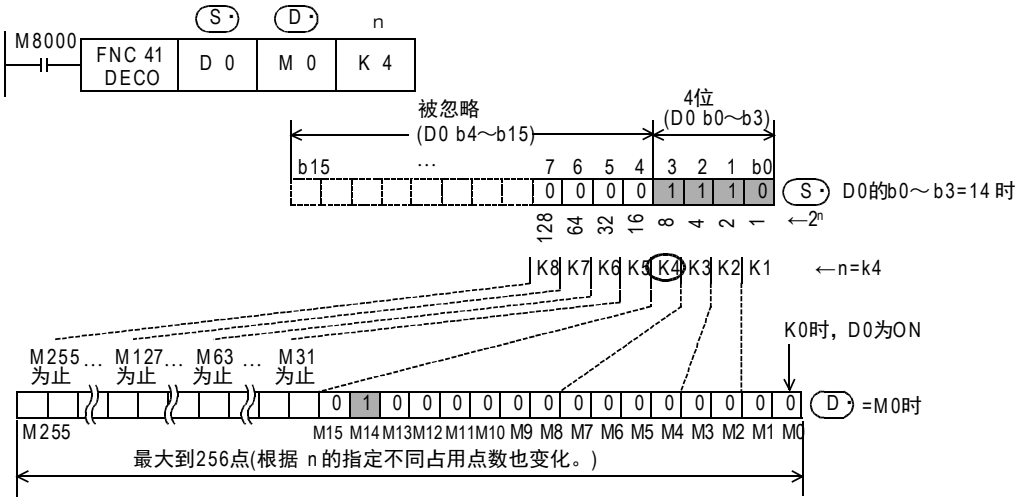
注意要点

- 指令输入为OFF时，不执行指令，但是已经在运行的译码输出会保持之前的ON/OFF状态。
- n=0时的指令为不处理。

程序举例

1. 根据数据寄存器的数值，使位软元件置ON的情况

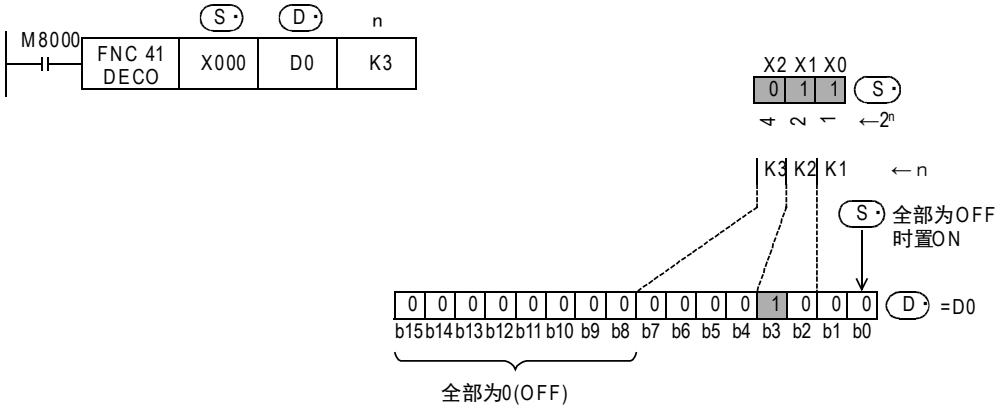
D0的值(当前值取14)在M0～M15中译码。



- D0的b0～b3的值为14(0+2+4+8)时，M0开始的第15号的M14为1(ON)。
- D0=0时，M0为1(ON)。
- 使n=K4，根据D0(0～15)的数值，M0～M15中任意一个为1点(ON)。
- 若使n在K1～K8之间变化，D0就可以对应0～255的数值。
  但是，这样的话作为译码所需的 D· 的软元件范围就被占用了，所以请注意不能与其他控制重复。

2. 根据位软元件的内容，使字软元件中的位置ON的情况

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



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

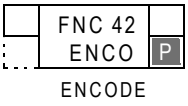
12.3 FNC 42 - ENCO/编码

A1 A2 A2C A8 A8C

概要

求出在数据中ON位的位置的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	ENCO	连续执行型
	ENCOP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要编码的数据，或是数据的字软元件编号	BIN16位
(D・)	保存编码结果的字软元件编号	BIN16位
n	保存编码结果的软元件的位点数(n=1～8) (n=0为不处理)	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)	●	●	●			●						●	●	●	●	▲	●	●	●					
(D・)												●	●	●	●	▲	●	●	●					
n																				●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(ENCO、ENCOP)

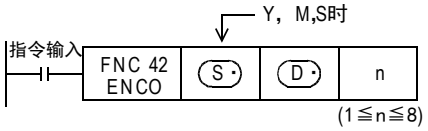
在 (D·) 中保存 (S·) 的2n位编码后的值。
编码就是将ON位的位置转换成BIN数据。

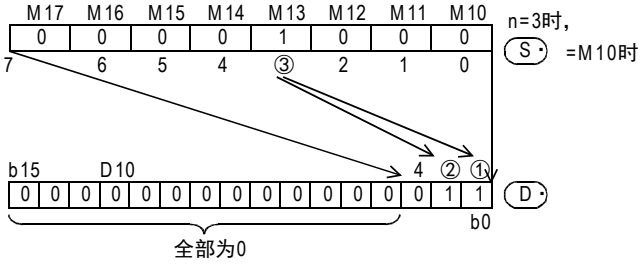
1) (S·) 为位软元件时(1≦n≦8)

(S·) 开始2^n个(1≦n≦8)的ON位位置，在 (D·) 中编码。

— n=8时，(S·) 为最大2^8=256点。

— (D·) 的编码结果从高位到低位n位，全部为0(OFF)。

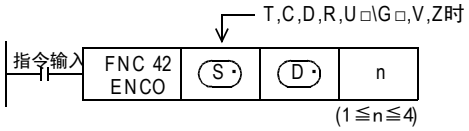


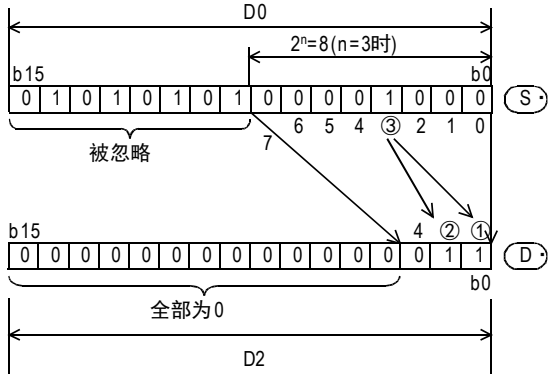


2) (S·) 为字软元件时(1≦n≦4)

到 (S·) 中指定的软元件的位2^n个(1≦n≦4)为止的ON位位置，都在 (D·) 中编码。

— (D·) 的编码结果从高位到低位n位，全部为0(OFF)。





注意要点

1. (S·) 的数据中多个位为ON的情况

忽略低位侧，对高位侧的ON位置进行编码。

2. 指令输入为OFF时

不执行指令，但是已经在运行的编码输出会保持之前的ON/OFF状态。

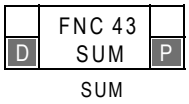
12.4 FNC 43 - SUM/ON位数

A8 A8C

概要

计算在指定的软元件的数据中有多少个为“1”(ON)的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	SUM	连续执行型
	SUMP	脉冲执行型

32位指令	指令记号	执行条件
9步	DSUM	连续执行型
	DSUMP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存数据的字软元件编号	BIN16/32位
(D・)	保存结果数据的字软元件编号	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

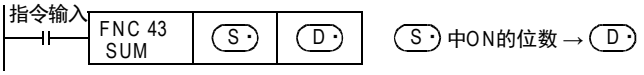
▲: 仅对应A8/A8C可编程控制器。

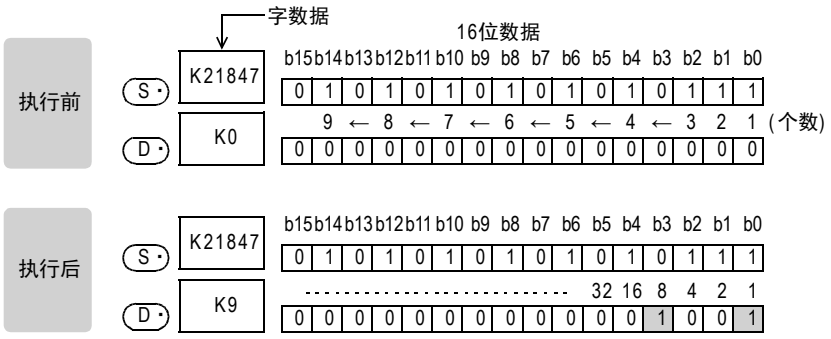
功能和动作说明

1. 16位运算(SUM、SUMP)

对(S・)中为ON的位进行计数后保存到(D・)中。

- (S・)都为0(OFF)时，零位标志位M8020为ON。

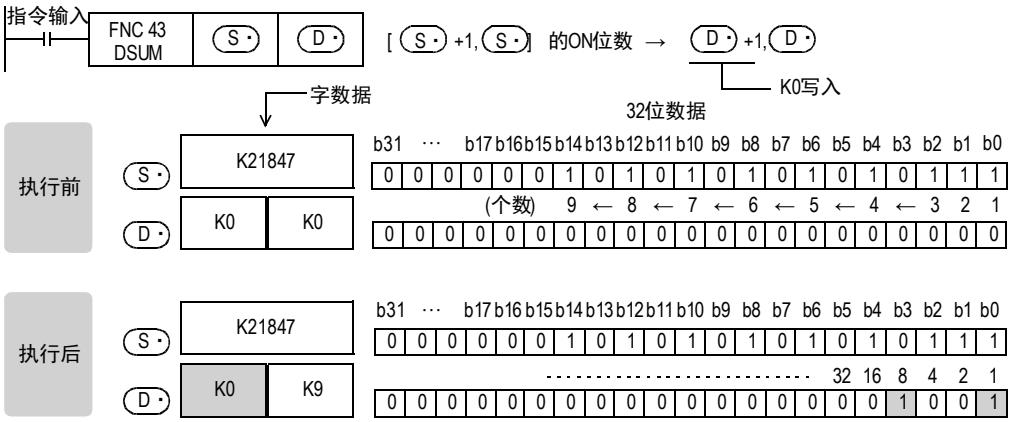




2. 32位运算(DSUM、DSUMP)

对[(S·)+1，(S·)]中为ON的位进行计数后保存到(D·)中。

- (D·)中保存ON位的点数，(D·)+1中保存K0。
- [(S·)+1，(S·)]都为0(OFF)时，零位标志位M8020为ON。



3. 依据 (S·) 的值的 (D·) 的运算结果(16位运算的情况)

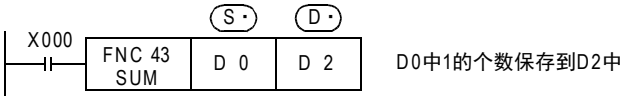
(S·) 位软元件																(S·) 字软元件		(D·)	M8020 零标志位
b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	10进制数	16进制数		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	0	ON
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0001	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	0002	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	3	0003	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4	0004	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	5	0005	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	6	0006	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	7	0007	3	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	8	0008	1	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	9	0009	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	10	000A	2	OFF
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	11	000B	3	OFF
⋮																⋮	⋮	⋮	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	-5	FFFB	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	-4	FFFC	14	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	-3	FFFD	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	-2	FFFE	15	OFF
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	-1	FFFF	16	OFF

注意要点

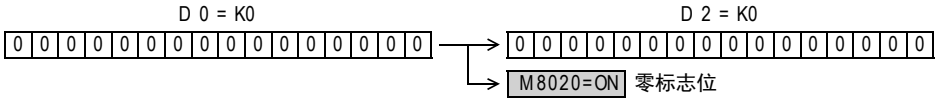
指令输入为OFF时不执行指令，但是已经在动作的ON位数的输出会保持之前的ON/OFF状态。

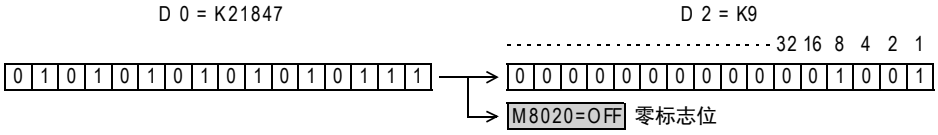
程序举例

X000为ON时，对D0中ON的位数计数后保存到D2中。



D0中1的个数保存到D2中





12.5 FNC 44 - BON/ON位的判定

A8 A8C

概要

检查软元件中指定位的位置为ON还是OFF的指令。

1. 指令格式

D	FNC 44 BON	P

BIT ON CHECK

16位指令	指令记号	执行条件
7步	BON	连续执行型
	BONP	脉冲执行型

32位指令	指令记号	执行条件
13步	DBON	连续执行型
	DBONP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存数据的字软元件编号	BIN16/32位
(D·)	驱动的位软元件编号	BIN16/32位
n	要判定的位位置[n: 0～15(16位指令)，n: 0～31(32位指令)]	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D·)		●	●			●	▲1												●					
n														●	●					●	●			

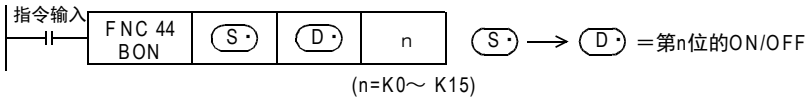
▲1 ：D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲2 ：仅对应A8/A8C可编程控制器。

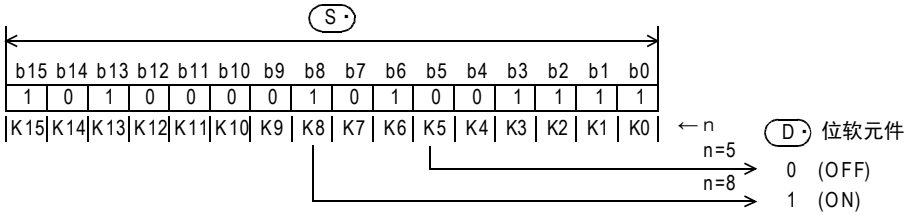
功能和动作说明

1. 16位运算(BON、BONP)

(S·)的n位的状态(ON或OFF)输出到(D·)。[ON→(D·)=ON，OFF→(D·)=OFF]

- 传送源(S·)中指定了常数(K)时，会自动执行BIN转换。

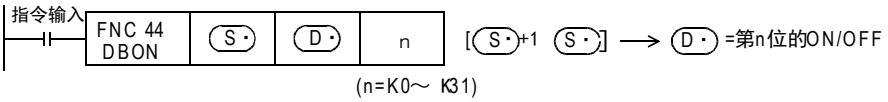


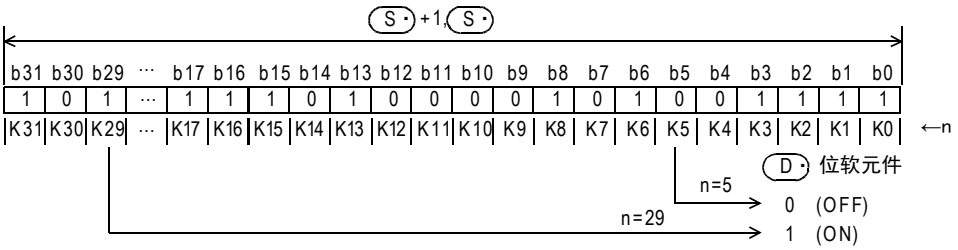


2. 32位运算(DBON、DBONP)

将[(S・)+1，(S・)]中的n位的状态(ON或OFF)输出到(D・)。[ON→(D・)=ON，OFF→(D・)=OFF]

- 传送源[(S・)+1，(S・)]中指定了常数(K)时，会自动执行BIN转换。



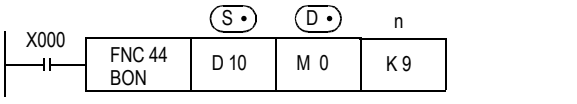


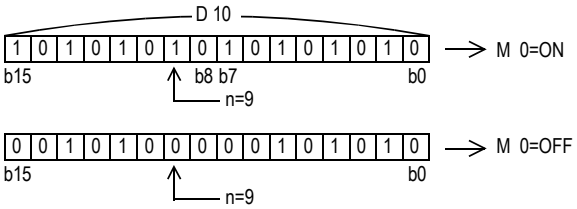
注意要点

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

程序举例

D10中的n=第9位为1(ON)时，M0为1(ON)。





12.6 FNC 45 - MEAN/平均值

A8 A8C

概要

求数据的平均值的指令。

1. 指令格式

D	FNC 45 MEAN	P

MEAN

16位指令	指令记号	执行条件
7步	MEAN	连续执行型
	MEANP	脉冲执行型

32位指令	指令记号	执行条件
13步	DMEAN	连续执行型
	DMEANP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存想要的平均值数据的起始字软元件编号	BIN16/32位
(D·)	保存取得的平均值数据的字软元件编号	BIN16/32位
n	平均数据数(n=1～64)	BIN16/32位

n=1～64

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲			●					
(D·)									●	●	●	●	●	●	●	▲	●	●	●					
n														●	●					●	●			

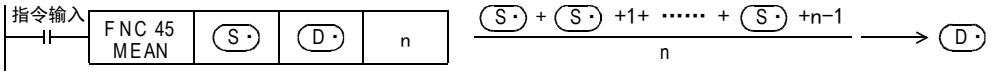
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(MEAN、MEANP)

将(S·)开始的n个16位数据的平均值保存到(D·)中。

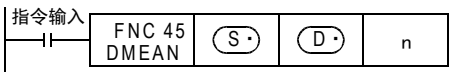
- 合计是求出代数和后被n除。
- 余数舍去。



2. 32位运算(DMEAN、DMEANP)

将[(S·)+1，(S·)]开始的n个32位数据的平均值保存到[(D·)+1，(D·)]中。

- 合计是求出代数和后被n除。
- 余数舍去。



$$\frac{[(S\cdot)+1,(S\cdot)]+[(S\cdot)+3,(S\cdot)+2]+\cdots\cdots+[\{(S\cdot)+n\times 2-1\},\{(S\cdot)+n\times 2-2\}]}{n}\rightarrow[(D\cdot)+1,(D\cdot)]$$

注意要点

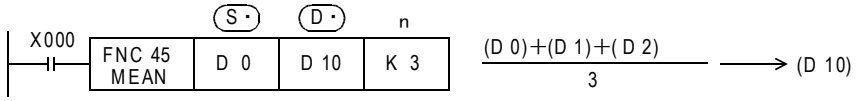
- 软元件编号溢出时，在可能的范围内将n作为较小的值处理。
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DMEAN D0 D100 R0时，则n=[R1, R0]。

错误

- 当n为1～64以外时，会发生运算错误(M8067)。

程序举例

将D0、D1、D2的数据相加，除以3后求得的值保存到D10中。



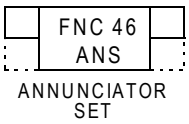
12.7 FNC 46 - ANS/信号报警器置位

A8 A8C

概要

对信号报警器用的状态(S900～S999)进行置位用的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	ANS	连续执行型

32位指令	指令记号	执行条件
—		

2. 设定数据

操作数种类	内容	数据类型
(S·)	判断时间的计时定时器编号	BIN16位
m	判断时间的数据[m=1～32,767(100ms单位)]	BIN16位
(D·)	设置的信号报警器软元件	BIN16位

3. 对象软元件

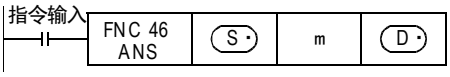
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)												▲1							●					
m														●	●					●	●			
(D·)						▲2													●					

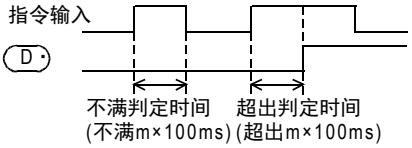
▲1:T0 ～T199
▲2: S900～S999

功能和动作说明

1. 16位运算

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





相关软元件

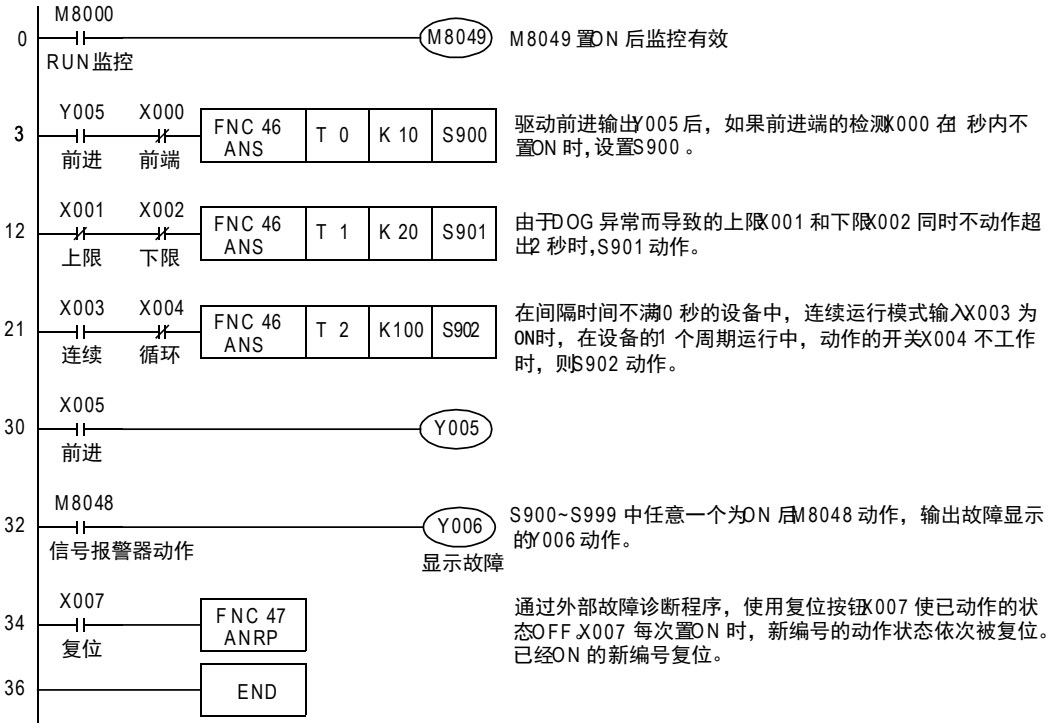
软元件	名称	内容
M8049	信号报警器有效	M8049置ON后，下面的M8048和D8049工作。
M8048	信号报警器动作	M8049为ON，状态S900～S999中任一动作的时候，M8048置ON。
D8049	ON状态最小编号	保存S900～S999中动作的最小编号。

程序举例

1. 通过信号报警器显示故障编号

如下所示，编写诊断外部故障用的程序，如监控D8049(ON状态最小编号)的内容时，会显示S900～S999中为ON的状态的最小编号。

同时发生多个故障时，排除了最小编号的故障后可以得知下一个故障编号。



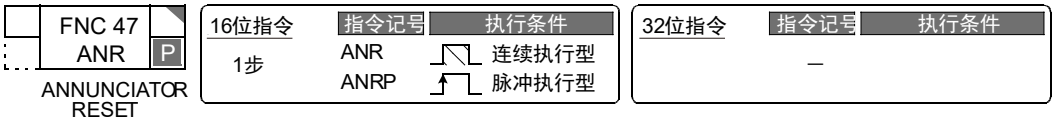
12.8 FNC 47 - ANR/信号报警器复位

A8 A8C

概要

对信号报警器(S900～S999)中已经置ON的小编号进行复位。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
—	无设定数据。	—

3. 对象软元件

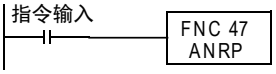
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
—	无对象软元件。																							

功能和动作说明

1. 16位运算(ANR、ANRP)

指令输入为ON后，将信号报警器S900～S999中运行中的状态复位。

- 如有多个状态动作时，复位编号最小的一个状态。
  再次使指令输入为ON后，在动作的信号报警器用状态(S900～S999)中，下一个最小的编号被复位。



相关软元件

软元件	名称	内容
M8049	信号报警器有效	M8049置ON后，下面的M8048和D8049工作。
M8048	信号报警器动作	M8049为ON，状态S900～S999中任一动作的时候，M8048置ON。
D8049	ON状态最小编号	保存S900～S999中动作的最小编号。

注意要点

1. 每个运算周期的执行

- 使用ANR指令时，每个运算周期都依次被复位。
- 使用ANRP指令时，仅执行1个运算周期(1次)。

程序举例

请参考ANS(FNC 46)指令。

→ 有关程序举例，参考12.7节

12.9 FNC 48 - SQR/BIN开方运算

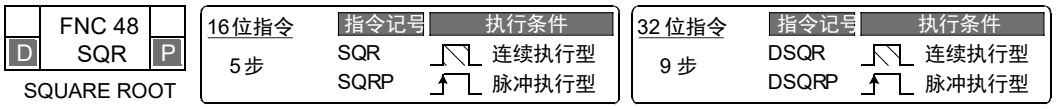
A1 A2 A2C A8 A8C

概要

求平方根(开根号)的指令。
也有浮点数运算用的ESQR(FNC 127)指令

→ ESQR(FNC 127)指令，请参考18.15节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要被开平方根运算数据的字软元件编号	BIN16/32位
(D・)	保存被执行了开平方根运算数据的数据寄存器编号	BIN16/32位

(S・): 16位运算: K0～K32,767/32位运算: K0～K2,147,483,647

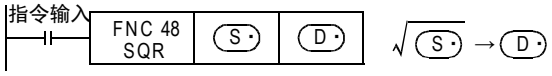
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●	●	●			
(D・)														●	●	●			●					

功能和动作说明

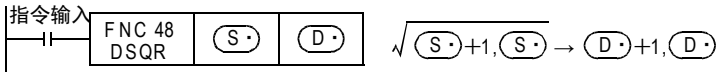
1. 16位运算(SQR、SQRP)

计算出(S・)的数据的平方根后，保存到(D・)中。



2. 32位运算(DSQR、DSQRP)

计算出[(S・)+1，(S・)]的数据的平方根后，保存到[(D・)+1，(D・)]中。



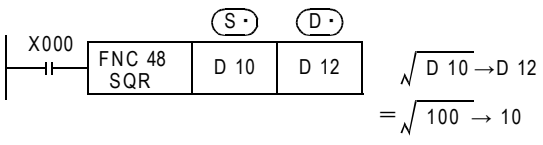
注意要点

1. 有关运算结果

1) 舍去小数点取整数。
舍去后生成时，M8021(借位标志位)置ON。
2) 真的0时，M8020(零位标志位)置ON。

程序举例

D10的平方根保存到D12中。
D10的值为100。



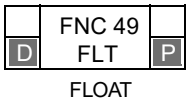
12.10 FNC 49 - FLT/BIN整数→2进制浮点数转换

A8 A8C

概要

将BIN整数值转换成2进制浮点数(实数)的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	FLT	连续执行型
	FLTP	脉冲执行型

32位指令	指令记号	执行条件
9步	DFLT	连续执行型
	DFLTP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存BIN整数值的数据寄存器编号	BIN16/32位
(D·)	保存2进制浮点数(实数)的数据寄存器编号	实数(2进制)

3. 对象软元件

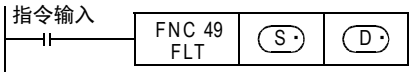
操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	▲			●					
(D·)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(FLT、FLTP)

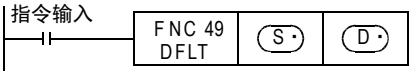
将(S·)的BIN整数值数据转换成2进制浮点数(实数)值后，保存在[(D·)+1，(D·)]中。



(S·) → ((D·)+1(D·))
BIN整数值 2进制浮点数(实数)值

2. 32位运算(DFLT、DFLTP)

将[(S·)+1，(S·)]的BIN整数值数据转换成2进制浮点数(实数)值后，保存在[(D·)+1，(D·)]中。



((S·) 1,(S·))→((D·) 1,(D·))
BIN整数值 2进制浮点数(实数)值

相关指令

指令	内容
INT(FNC 129)	FLT指令的逆转换动作指令

注意要点

1. 不需要常数(K、H)的浮点数转换

由于在各2进制浮点数(实数)运算指令中，指定的K、H的值会自动转换成2进制浮点数(实数)，所以不需要使用FLT指令进行转换。

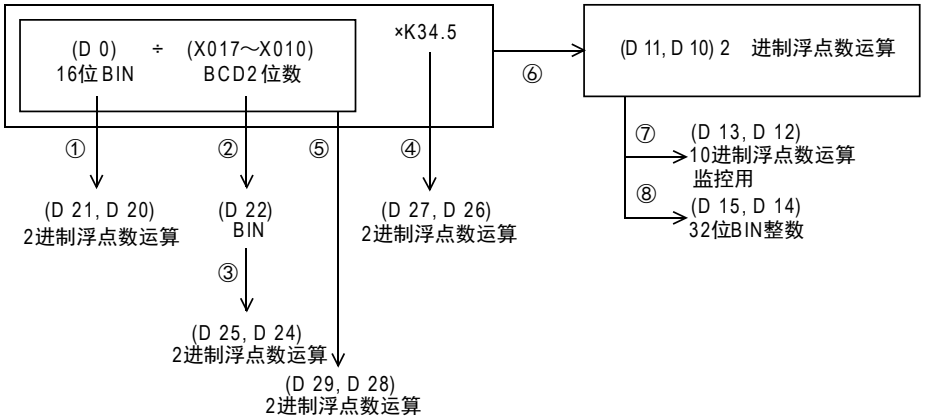
(RAD指令、DEG指令、EXP指令、LOGE指令等都不能指定K、H。)

程序举例

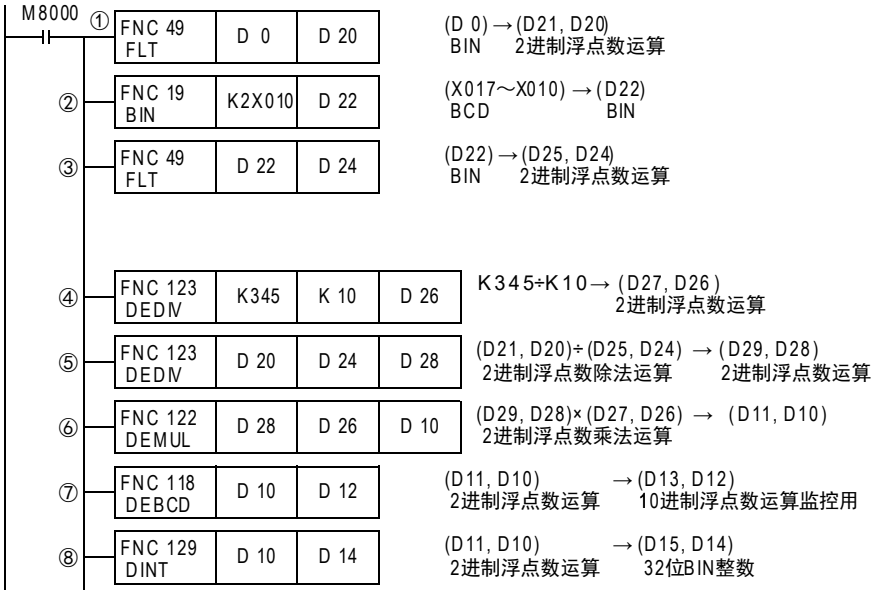
1. 采用2进制浮点数运算的四则运算

说明下面例题中的顺控程序。

1) 计算例子

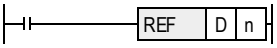
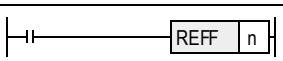
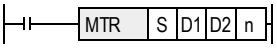
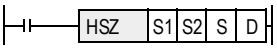
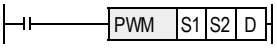
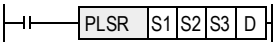


2) 顺控程序



13. 高速处理—FNC 50～FNC 59

在FNC 50～FNC 59中，提供了用最新的输入输出信息进行顺控，以及利用可编程控制器的高速处理能力的中断处理型的高速处理指令。

FNC No.	指令记号	符号	功能	参考
50	REF		输入输出刷新	13.1节
51	REFF		输入刷新(带滤波器设定)	13.2节
52	MTR		矩阵输入	13.3节
53	HSCS		比较置位(高速计数器用)	13.4节
54	HSCR		比较复位(高速计数器用)	13.5节
55	HSZ		区间比较(高速计数器用)	13.6节
56	SPD		脉冲密度	13.7节
57	PLSY		脉冲输出	13.8节
58	PWM		脉宽调制	13.9节
59	PLSR		带加减速的脉冲输出	13.10节

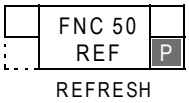
13.1 FNC 50 - REF/输入输出刷新

A1 A2 A2C A8 A8C

概要

在顺控程序扫描过程中，想要获得最新的输入(X)信息时，以及将输出(Y)扫描结果立即输出的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	REF	连续执行型
	REFP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(D)	刷新的位软元件(X、Y)编号	位
n	刷新的位软元件点数(8～256中8的倍数) A8/A8C:8～256中8的倍数，	BIN16位

3. 对象软元件

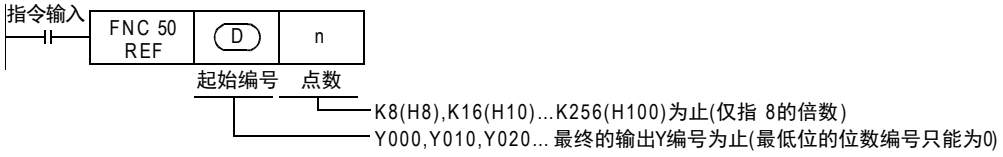
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D)	▲1	▲2																						
n																				▲3	▲3			

▲1 ：X000、X010、X020.....最终输入编号为止(最低位位数编号仅为0)
▲2 ：Y000、Y010、Y020 .....最终输出编号为止(最低位位数编号仅为0)
▲3 ：A8/A8C可编程控制器时，从K8(H8)、K16(H10).....到K256(H100)为止(仅限8的倍数)

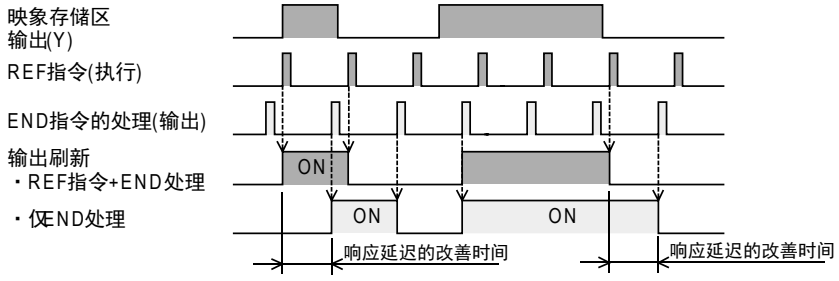
功能和动作说明

1. 16位运算(REF、REFP)

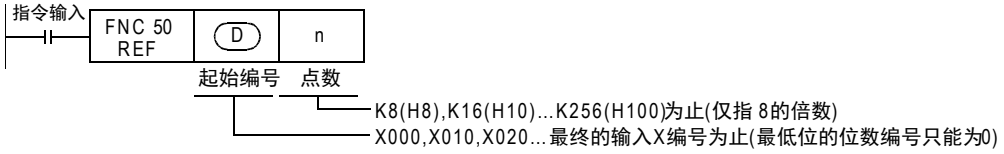
1) 刷新输出(Y)时
刷新输出(D)开始的n点(8点单位)。



— 执行该指令时，将指定范围内的输出状态刷新到输出锁存存储区。



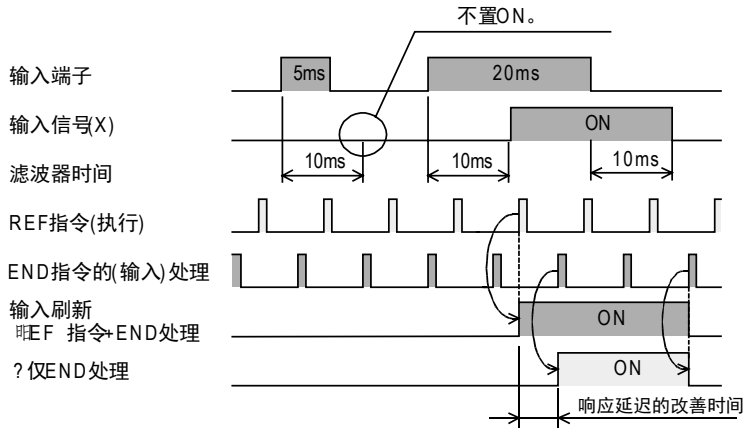
2) 刷新输入(X)时
刷新输入(D)开始的n点(8点单位)。



－ 如果在指令执行之前约10ms(输入滤波器的响应延迟时间)，输入信息就已经置ON，
则执行该指令时，输入映像存储区为ON。

－ X000～X017[*1]的输入滤波器的响应延迟时间可以改变。

→ 详细内容参考后述的[13.1.1 使用REF指令前须知]



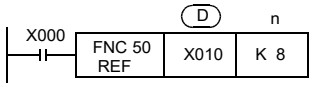
注意要点

1. 刷新点数[n]的设定
请设置为类似K8(H8)、K16(H10)…K256(H100)的8的倍数。如果是除此以外的数值，会错误。

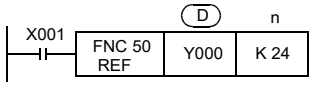
2. 指定起始软元件号(D)的设定
请像X000、X010、X020…和Y000、Y010、Y020…这样，将最低位的位数编号设置为0。

程序举例

1. 输入刷新的情况
仅X010～X017的8点被刷新。



2. 输出刷新的情况
仅Y000～Y007、Y010～Y017、Y020～Y027的24点被刷新。



13.1.1 使用REF指令前须知

1. 关于输入滤波器的变更

输入滤波器的值，是根据D8020的设计内容(初期值: 10ms)而变化的。
要缩短输入滤波器的值时，请使用MOV指令等更改D8020的内容。

对应范围: X000～X017(X020以后固定为10ms，不能更改)
上述范围仅针对基本单元，根据内置输入点数有时候会减少。

2. 关于输出响应时间

执行REF指令后，输出(Y)在下述的响应时间以后接通输出信号。

→ 详细内容，参考使用的可编程控制器硬件手册中的规格

对应范围:Y000 ～连接的最终输出编号为止

1) 继电器输出型的情况
在输出继电器的响应时间后输出触点动作。
－Y000以后: 约10ms

2) 晶体管输出型的情况
a) A8/A8C(D、DSS)可编程控制器
－Y000、Y001、Y002: 5μs以下(负载电流10mA以上 DC5～24V)
－Y003以后: 0.2ms以下(负载电流100mA以上DC24V)
b) A8C-32MT-LT(-2)可编程控制器
－Y000、Y001、Y002、Y003: 5μs以下(负载电流10mA以上DC5～24V)
－Y004以后: 0.2ms以下(负载电流100mA DC24V)

3. 在FOR～NEXT指令之间及标签(新步号)～CJ指令(老步号)之间使用的情况

如果在控制过程中，即使是在子程序扫描期间，也需要输入信息及即时输出时，可以在子程序内进行刷新。

4. 使用输入中断功能(I)的情况

在执行有输入输出动作的中断处理时，在中断子程序内执行输入输出刷新，读取最新的输入(X)信息以及即时输出(Y)的扫描结果，从而改善因扫描时间引起的偏差。

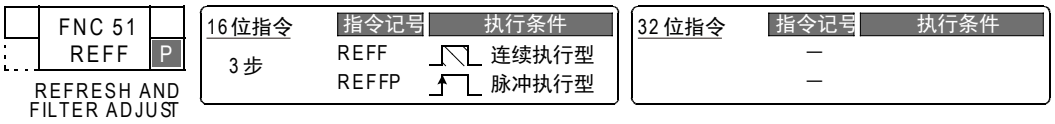
13.2 FNC 51 - REFF/输入刷新(带滤波器设定)

A8 A8C

概要

输入X000～X017[*1]的输入滤波器为数字式滤波器，采用本指令和D8020可以更改滤波器的时间。
使用本指令，可以在程序的任意步中，根据指定的输入滤波器时间获取输入X000～X017[*1]的信息，并且传送到映象存储区中。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
n	数字式滤波器的时间数据[K0～K60(H0～H3C)×1ms]	BIN16位

3. 对象软元件

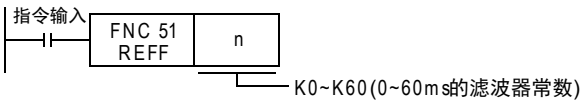
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
n														●	●					▲	▲			

▲: K0(H0)～K60(H3C)

功能和动作说明

1. 16位运算(REFF、REFFP)

根据输入的数字式滤波器时间[n×1ms]刷新输入X000～X017[*1]的16点映像存储区。



- 如果在指令执行前的n× 1ms前输入已经为ON时，则输入映象存储区为ON，如果输入已经为OFF时，则输入映象存储区为OFF。
- 指令输入为ON时，REFF指令在每个运算周期都被执行。
- 指令输入为OFF时，不执行该指令，X000～X017[*1]的输入滤波器为D8020的设定值。(变为输入处理时的值)

*1. A8-16M□、A8C-16M□时，为X000～X007。

注意要点

1. 滤波器时间[n]的设定
   请在K0(H0)～K60(H3C)[0～60ms]的范围内设定n。

2. 输入滤波器的运行
   X000～X017*1内置数字式滤波器，可以采用应用指令在0～ 60ms范围内以1ms为单位更改滤波器的时间。但是，设定为0时，变为下表中的值。

输入编号	输入滤波常数设定为0时的值
X000～X005	5μs*2
X006,X007	50μs
X010～X017*3	200μs*3

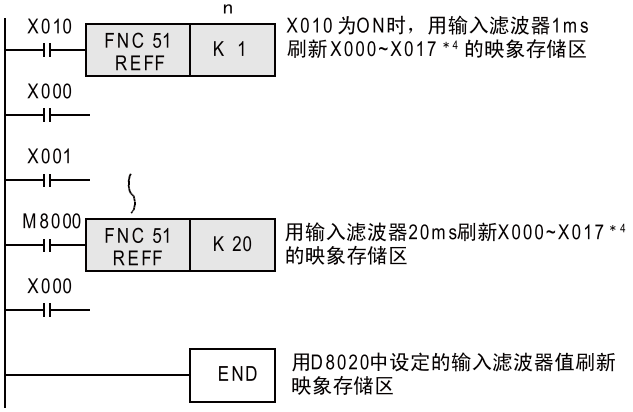
*1. A8-16M□和A8C-16M□为X000～X007。

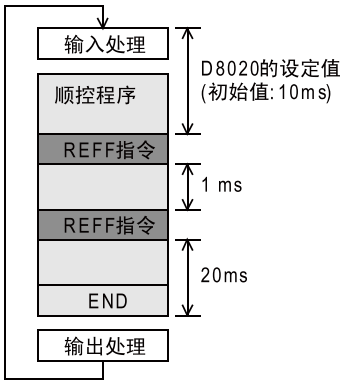
*2. 使用5μs输入滤波器时，请如下所示设置。
- 接线长度请勿超出5m。
- 在输入端子中连接1.5kΩ(1W以上)的漏电阻，与主机的输入电流相符合，对方一侧机器的开集电极型晶体管输出的负载电流需要在20mA以上。

*3. 使用A8-16M□、A8C-16M□时，X010～X017的滤波器时间固定为10ms。

程序举例

1. 程序和滤波器时间的关系





*4. A8-16M□、A8C-16M□时，为X000～X007。

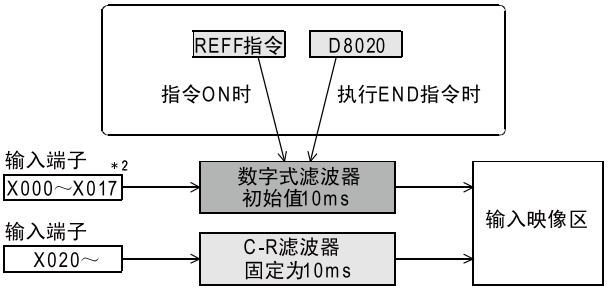
13.2.1 使用REFF指令前须知

通常，为了防止输入触点的振荡和噪音会在可编程控制器的输入中设置约10ms的C-R滤波器。
在这样的可编程控制器中，输入X000～X017*1采用了数字式滤波器，因此通过指令可以在0～ 60ms之间更改其值。

*1. A8-16M□和A8C-16M□为X000～X007。

1. 数字式滤波器的变更方法(执行END指令)

在特殊数据寄存器D8020中设定X000～X017*2的输入滤波器的初始值(10ms)。
因此，如采用MOV指令等改写该值时，可以更改在执行END指令时被执行的X000～X017*2的输入滤波器值。



*2. A8-16M□和A8C-16M□为X000～X007。

2. 有关自动更改数字式滤波器的指令

与采用REFF指令更改滤波器的时间无关，下述的功能和指令，都是在输入滤波器自动变化(X000～X005: 5μs、X006 X007: 50 μs)后执行。
但是，在下述的功能和指令以外的用途中，用于普通程序中时，由于会变成D8020中设定的滤波器时间，所以如果相应的输入信号的ON宽度或是OFF宽度不能超出输入滤波器时间，则不能正确处理程序。

- 输入中断功能中指定的中断指针的输入
- 高速计数器中使用的输入
- SPD(FNC 56)指令中使用的输入

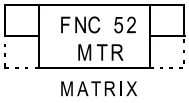
13.3 FNC 52 - MTR/矩阵输入

A1 A2 A2C A8 A8C

概要

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

1. 指令格式



16位指令	指令记号	执行条件
9步	MTR	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S)	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)	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)	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 (最低位的位数编号只能为0)	位
n	设定矩阵输入的列数(K2～K8/H2～H8)	BIN16位

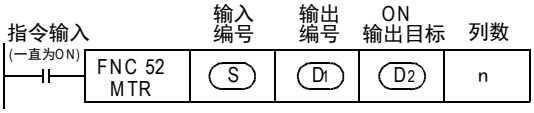
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S)	●																							
(D1)		●																						
(D2)		●	●			●																		
n																				●	●			

功能和动作说明

1. 16位运算(MTR)

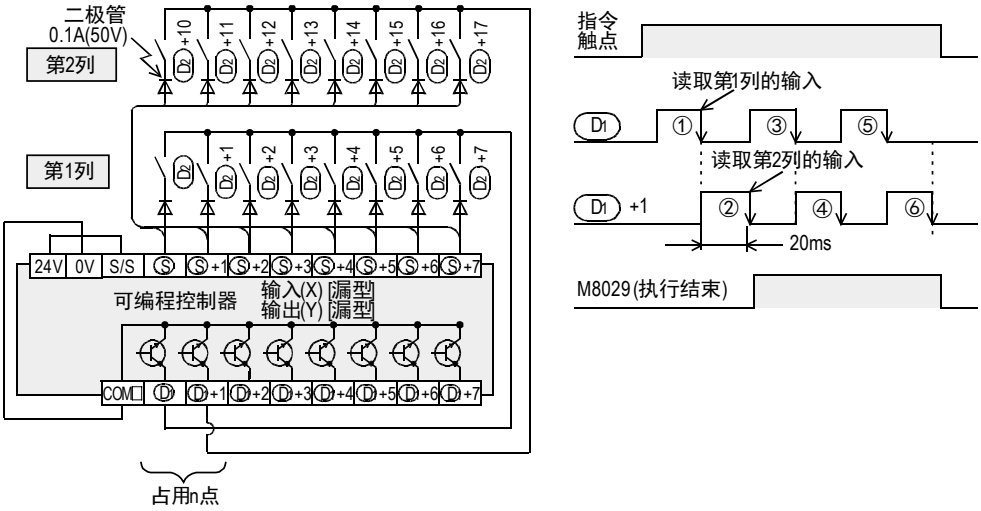
对8点的 (S) 输入和n点的 (D1) 晶体管输出进行时间分割控制，以便依次读取8点n列的输入信号，然后输出到 (D2) 中。



- 考虑到输入滤波器响应延迟10ms，各输出按每20ms顺序依次进行中断，执行即时的输入输出处理。

下图是以A8系列的基本单元(漏型输入/漏型输出)为例。有关接线，请参考所使用的可编程控制器的下面一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册



相关软元件

软元件	名称	内容
M8029	指令执行结束	初次的循环动作后置ON。

注意要点

1. 软元件的占用点数
   1) 以 (S) 中指定的输入作为起始，占用8点的输入。
   2) 以 (D1) 中指定的输出作为起始，占用n点的输出。
   (D2) 中指定输出时，请注意不要与 (D1) 中指定的输出编号(占用n点)重复。
2. 有关接线
   对1个开关，必须要1个0.1A/50V的二极管。
3. 有关输出形式
   请使用晶体管输出。

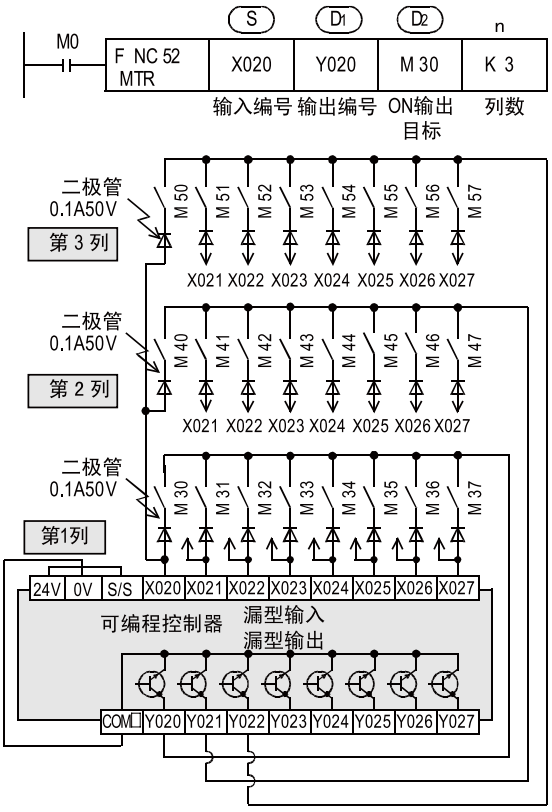
程序举例

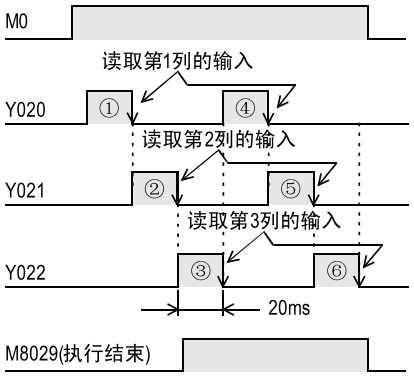
n=3点的输出(Y020、Y021、Y022)依次重复置ON。

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，此程序是以A8系列的基本单元(漏型输入/漏型输出)为例。有关接线，请参考所使用的可编程控制器的下面一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册

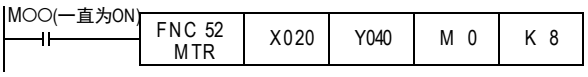




13.3.1 MTR指令的使用方法和注意要点

1. 有关指令输入

1) 指令输入一直为ON时
MTR指令，通常使用一直为ON的信号作为指令输入。



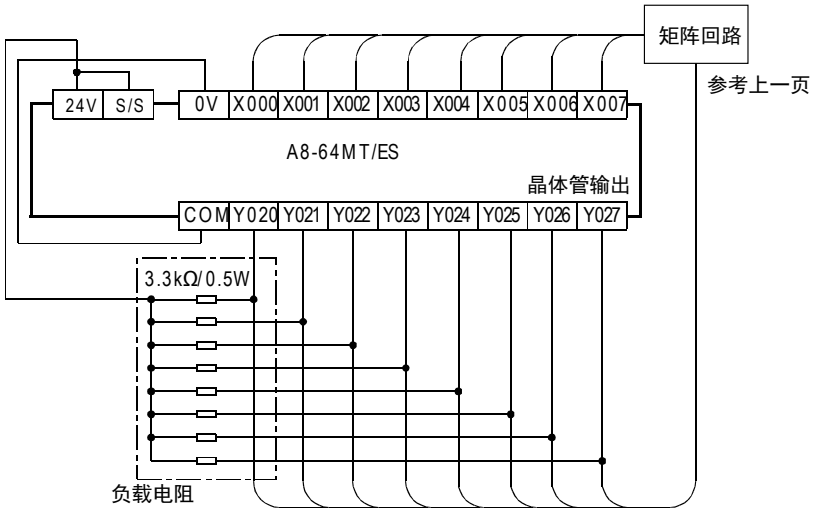
2. 有关MTR指令使用的输入编号

1) MTR指令使用的输入
通常请使用输入X020以后的。

2) 使用输入X000～X017时
读取速度会变快，但是由于输出晶体管的还原时间长并且输入灵敏度高，因此会产生误输入的情况。
因此，请在MTR指令使用的晶体管输出上连接负载电阻(3.3kΩ/0.5W)。
有关负载中使用的电源，请使用下表中的电源。

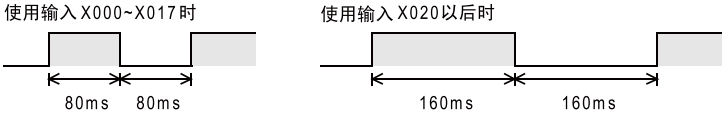
	负载中使用的电源
AC电源型的可编程控制器	工作电源
DC电源型的可编程控制器	驱动可编程控制器的电源

下图是以A8系列的基本单元(漏型输入/漏型输出)为例。



3. 有关输入信号的ON/OFF时间

由于对8行×8列=64点的输入，按照80ms、或是160ms的读取周期进行动作，所以各输入信号的ON/OFF宽度需要达到下面的值以上。



13.4 FNC 53 - HSCS/比较置位(高速计数器用)

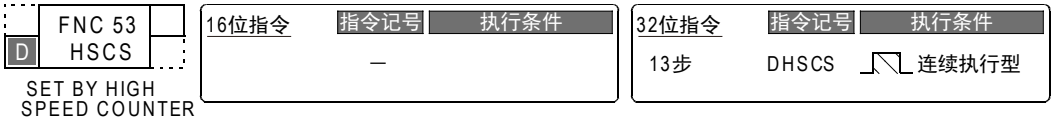
A1 A2 A2C A8 A8C

概要

每次执行时，都将高速计数器的计数值和指定值做比较，如果两个值相等，立即置位外部输出(Y)的指令。

→ 有关本指令使用的计数器中断，请参考35.6节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(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	BIN32位
(S2・)	高速计数器的软元件编号[C235～C255]	BIN32位
(D・)	一致后进行置位(ON)的位软元件编号	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2		●	●	●	●			
(S2・)													●						●					
(D・)		●	●			●	▲1												●					▲3

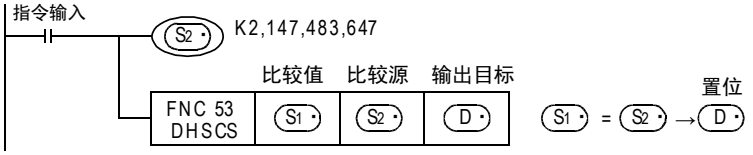
▲1: D□.b仅对应A8/A8C可编程控制器。但是不能变址修饰(V、Z)。
▲2: 仅对应A8/A8C可编程控制器。
▲3: A8/A8C可编程控制器中使用计数器中断时，指定中断指针。

→ 有关本指令使用的计数器中断，请参考35.6节

功能和动作说明

1. 32位运算(DHSCS)

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)，位软元件(D・)被置位(ON)，与运算周期无关。这个指令是接着高速计数器的计数处理之后执行比较处理的指令。



动作

高速计数器C255的当前值从99变为100或者从101变为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	FNC No.	指令名称
DHSCS	FNC 53	比较置位(高速计数器)
DHSCR	FNC 54	比较复位(高速计数器)
DHSZ	FNC 55	区间比较(高速计数器)
DHCMOV	FNC 189	高速计数器的传送
DHSCT	FNC 280	高速计数器的表格比较

注意要点

1. 计数比较方法的选定

1) A8/A8C可编程控制器

在A8/A8C可编程控制器中使用该指令时，硬件计数器(C235、C236、C237、C238、C239、C240、C244(OP)、C245(OP)、C246、C248(OP)、C251、C253)会自动切换成软件计数器，并影响各计数器的最高频率以及综合频率。

请参考下面的计数动作后，根据具体控制内容来选择是使用该指令，还是使用通用的比较指令。

a) 使用该指令时的要点

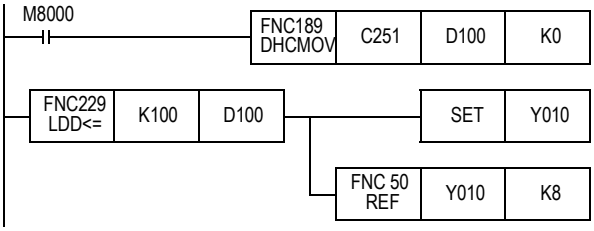
－不受可编程控制器扫描时间的影响， 想要计数一致时输出的时候使用

b) 使用通用的比较指令时的要点

－频率超出软件计数器的计数能力时(硬件计数器的计数能力以下)

－重视计数， 但是使用了计数结果的动作可以忽略扫描时间的影响

－超出指令的使用次数(32次)时



2. 软元件的指定范围

(S・)中可以指定的软元件，仅高速计数器(C235～C255)有效。

3. 只可以使用32位运算指令

由于高速计数器用的指令是32位专用指令，所以请输入“DHSCS”(FNC 53)。

4. 关于针对相同的高速计数器执行HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55)指令的优先动作顺序

→ 详细内容，请参考13.4.1项6

5. 通过外部端子执行复位动作

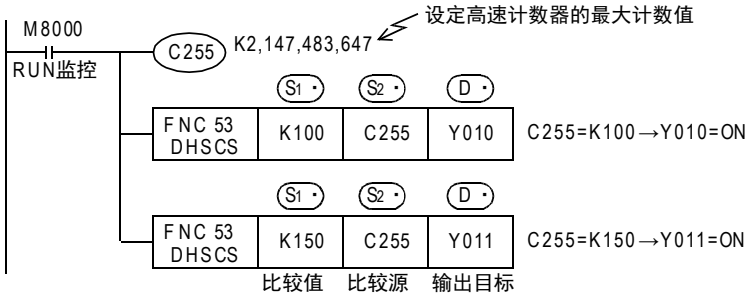
→ 详细内容情参考13.4.1项5

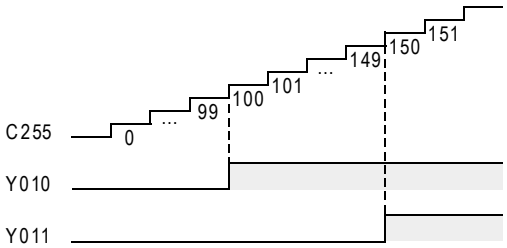
6. 其他的使用上的注意事项，请参考后述内容

→ 详细内容，请参考下一页

程序举例

针对计数器的当前值，用任意的2个值分别作比较，以分别使各输出(Y)置位(ON)。





13.4.1 使用高速计数器用指令时的通用注意事项

高速计数器用的指令包括DHSCS(FNC 53),DHSCR(FNC 54),DHSZ(FNC 55),DHSCT(FNC 280)指令，下面说明通用的注意事项。

1. 程序中的使用次数的限制

1) A8/A8C可编程控制器

DHSCS、DHSCR、DHSZ和普通的指令相同可以无数次编程。但是，可以同时驱动这些指令的数量是有限制的。而且，DHSCT在程序中只可以使用1次。

指令	指令同时驱动点数的限制
DHSCS	包括DHSCT指令在内最多可以同时驱动32条。
DHSCR	
DHSZ	
DHSCT	只可以驱动1条。(指令只可以使用1次)

2. 高速计数器的响应频率

对A8/A8C可编程控制器,使用DHSZ指令或DHSCT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。

→ 有关A8/A8C可编程控制器的最大响应频率和综合频率的详细信息，请参考4.7.10项

3. 有关输出Y编号的指令

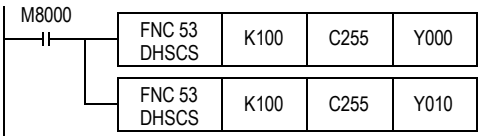
使用多个相同的高速计数器用指令时，以及同时驱动其他的高速计数器用指令时，对象输出(Y)的高2位数请使用同一编号(8点单位内)的软元件。

1) 使用同一编号(8点单位内)的软元件的情况

例如: 使用Y000时为Y000～Y007、使用Y010时为Y010～Y017等

2) 多次使用高速计数器用指令后，输出(Y)编号错开的情况

以下面的例子进行说明。



当C255达到K100，以中断方式执行Y000的输出驱动，而Y010在END处理时被驱动。

希望采取中断驱动时，请用高2位数的编号相同的Y001～Y007。

4. 通过更改当前值执行计数器动作时的注意事项

高速计数器用指令，在高速计数器用输入(X)中有脉冲输入时，比较结果动作。
但是，如果使用下面的方法更改高速计数器的当前值时不动作。

1) 更改方法(例)

a) 使用DMOV指令等改写作为比较对象的字软元件的内容。

b) 在程序中复位高速计数器的当前值。

2) 动作

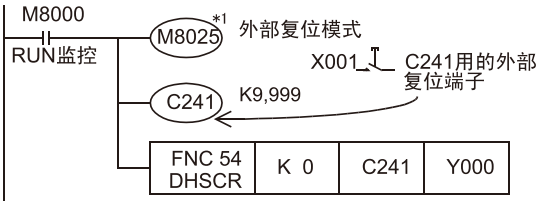
即使比较结果是使输出置ON或置OFF的先决条件，仅仅驱动指令时比较结果不变化。

5. 通过外部端子复位的动作[M8025: HSC(外部复位)模式]

关于C241等备有外部复位端子(R)的高速计数器，都是依据复位输入信号的上升沿执行指令后，输出比较结果。

1) 程序

在驱动了M8025的状态下使用高速计数器用指令时，在通过外部复位端子清除高速计数器(C241)的当前值时再次执行指令，无需计数输入，就可以输出比较结果。



*1. M8025在RUN→STOP时被清除。如果是

2) 动作

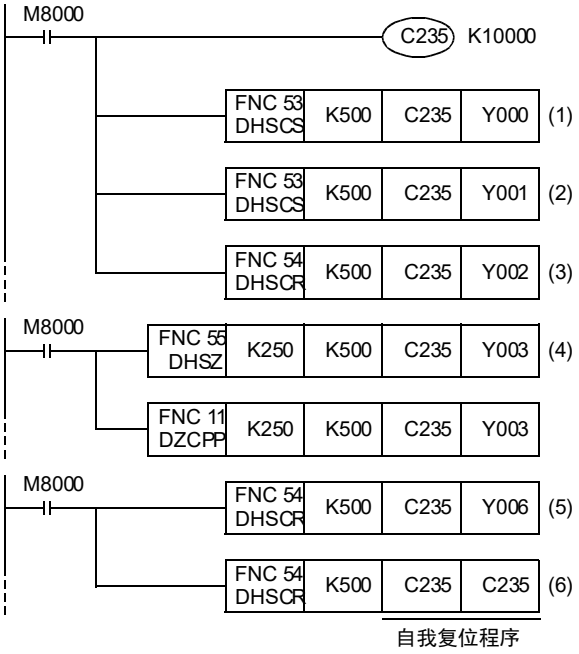
假设C241的当前值为100，外部复位输入X001置ON后，则C241的当前值变为0，即使没有计数输入，此时Y000也会被复位。

6. 关于针对相同的高速计数器使用相同比较值的HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55)指令的优先动作顺序

1) A8/A8C可编程控制器

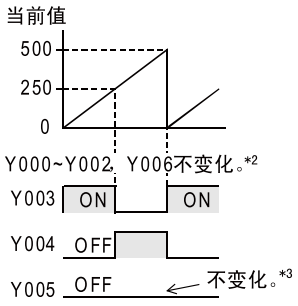
对相同高速计数器使用相同的比较值的情况下，HSCS、HSCR、HSZ指令的动作中，最优先执行HSCR的比较对照高速计数器的复位(自我复位)动作。(下表)

此时，与HSCR指令的自我复位使用相同比较值编程的HSCS、HSCR、HSZ指令的比较结果都不变化。为了驱动比较结果，请将比较值设定在“K0”。



程序的顺序	处理的顺序	
	A8/A8C	A1/A2/A2C
(1)的DHSCS	(6)的DHSCR (自我复位)	(1)的DHSCS
(2)的DHSCS	(4)的DHSZ	(2)的DHSCS
(3)的DHSCR	(1)的DHSCS	(3)的DHSCR
(4)的DHSZ	(2)的DHSCS	—
(5)的DHSCR	(3)的DHSCR	(5)的DHSCR
(6)的DHSCR (自我复位)	(5)的DHSCR	(6)的DHSCR (自我复位)

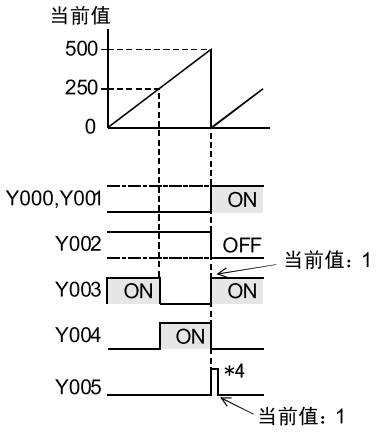
A8/A8C可编程控制器的动作



*2. 为了使(1)~(3)、(5)的指令的比较结果变化，请将(1)~(3)、(5)的比较值“K500”设定为“K0”。

*3. 希望使(4)的HSZ的Y005为ON时，比较值“K500”设定为较小的值。
但是，由于输出中会出现响应延迟，所以当前值从设定的值变化为K500(K0)的时间内，输出有时候会不动作。

A2/A2C/A1可编程控制器的动作



*4. 在输出中，由于存在响应延迟，所以当前值从0变为1的时间内，输出有时候会不动作。

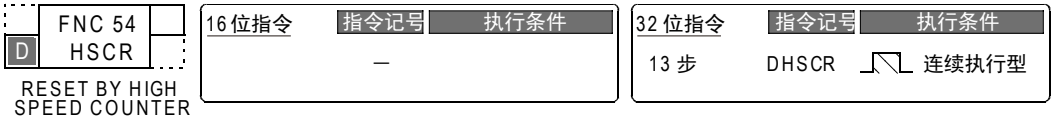
13.5 FNC 54 - HSCR/比较复位(高速计数器用)

A1 A2 A2C A8 A8C

概要

每次执行时，将高速计数器的计数值和指定值做比较，如果两个值相等，立即复位外部输出(Y)的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(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	BIN32位
(S2・)	高速计数器的软元件编号[C235～C255]	BIN32位
(D・)	一致后进行复位(OFF)的位软元件编号	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲3		●	●	●	●			
(S2・)													●						●					
(D・)		●	●			●	▲1						▲2						●					

▲1: D□.b仅对应A8/A8C可编程控制器。但是不能变址修饰(V、Z)。
▲2: 也可指定与 (S2・) 相同的计数器(参考程序举例)
▲3: 仅对应A8/A8C可编程控制器。

功能和动作说明

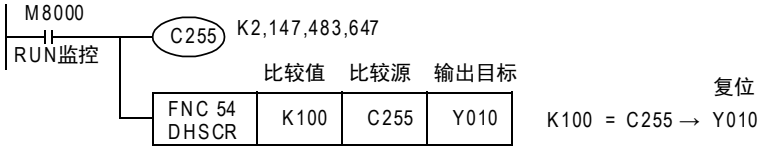
1. 32位运算(DHSCR)

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)，无论运算周期多少，位软元件(D・)被复位(OFF)，与运算周期无关。这个指令是接着高速计数器的计数处理之后执行比较处理的指令。



动作

高速计数器C255的当前值从99变为100或者从101变为100(计数)时，Y010被复位(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	FNC No.	指令名称
DHSCS	FNC 53	比较置位(高速计数器)
DHSCR	FNC 54	比较复位(高速计数器)
DHSZ	FNC 55	区间比较(高速计数器)
DHCMOV	FNC 189	高速计数器的传送
DHSCT	FNC 280	高速计数器的表格比较

注意要点

1. 计数比较方法的选定

1) A8/A8C可编程控制器

在A8/A8C可编程控制器中使用该指令时，硬件计数器(C235、C236、C237、C238、C239、C240、C244(OP)、C245(OP)、C246、C248(OP)、C251、C253)会自动切换成软件计数器，并影响各计数器的最高频率以及综合频率。

请参考下面的计数动作后，根据具体控制内容来选择是使用该指令，还是使用通用的比较指令。

a) 使用该指令时的要点

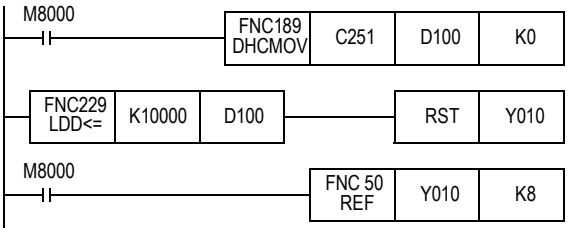
－不受可编程控制器扫描时间的影响， 想要计数一致时输出的时候使用

b) 使用通用的比较指令时的要点

－频率超出软件计数器的计数能力时(硬件计数器的计数能力以下)

－重视计数， 但是使用了计数结果的动作可以忽略扫描时间的影响

－超出指令的使用次数(32次)时



2. 只可以使用32位运算指令

由于高速计数器用的指令是32位专用指令，所以请务必输入“DHSCR”(FNC 54)。

3. 关于针对相同的高速计数器执行HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55)指令的优先动作顺序

→ 详细内容，请参考13.4.1项6

4. 通过外部端子执行复位动作

→ 详细内容，请参考13.4.1项5

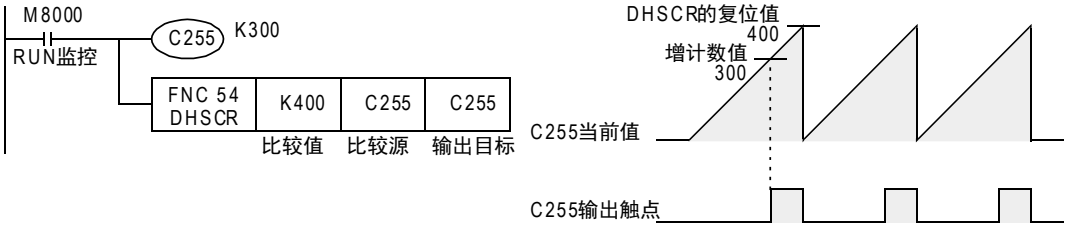
5. 其他的使用上的注意事项

→ 详细内容，请参考13.4.1项

程序举例

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。



13.6 FNC 55 - HSZ/区间比较(高速计数器用)

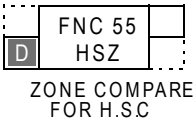
A8 A8C

概要

将高速计数器的当前值和2个值(区间)进行比较，并将比较结果输出(刷新)位软元件(3点)中。

→ 有关表格高速比较模式，请参考13.6.2项
→ 关于频率控制模式，请参考13.6.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
17步	DHSZ	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	与高速计数器的当前值进行比较的数据，或是保存比较数据的字软元件编号(比较值1)	BIN32位
(S2・)	与高速计数器的当前值进行比较的数据，或是保存比较数据的字软元件编号(比较值2)	BIN32位
(S・)	高速计数器的软元件编号[C235～C255]	BIN32位
(D・)	输出与比较上限值和比较下限值比较的结果的起始位软元件编号	位

3. 对象软元件

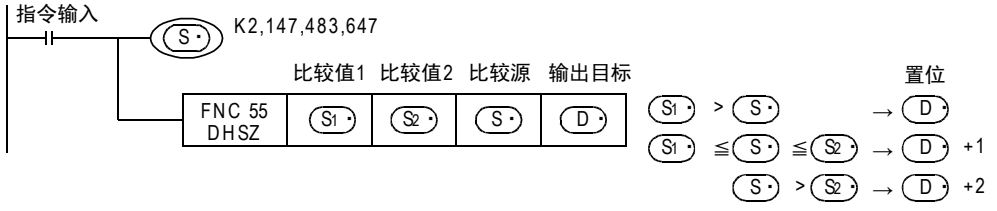
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2		●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲2		●	●	●	●			
(S・)													●						●					
(D・)		●	●			●	▲1												●					

▲1: D□.b仅对应A8/A8C可编程控制器。但是不能变址修饰(V、Z)。
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 32位运算(DHSZ)

当(S・)中指定的高速计数器(C235～C255)的当前值和2个比较点(比较值1、比较值 2)进行区间比较，与运算周期无关，将比较得出的小、区间内、大的结果的(D・)、(D・)+1、(D・)+2中任意一个置ON。
这个指令是接着高速计数器的计数处理之后执行比较处理的指令。



比较点

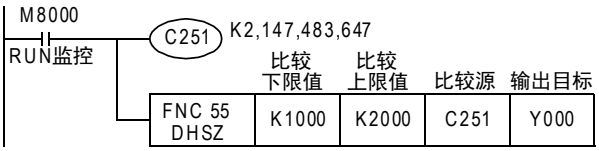
比较值1和比较值2的关系如下所示。

[(S1•)+1、(S1•)]≦[(S2•)+1、(S2•)]

比较点	内容[32位]
比较值1	(S1•)+1, (S1•)
比较值2	(S2•)+1, (S2•)

动作

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



ON

K1000 > C251 → Y000

K1000 ≦ C251 ≦ K2000 → Y001

C251 > K200 0 → Y002

比较模式	C251的当前值	输出触点(Y)的变化		
		Y000	Y001	Y002
(S1•) > (S•)	1000> (S•)	ON	OFF	OFF
	999→1000	ON→OFF	OFF→ON	OFF
	999←1000	OFF→ON	ON→OFF	OFF
(S1•) ≦ (S•) ≦ (S2•)	999→1000	ON→OFF	OFF→ON	OFF
	999←1000	OFF→ON	ON→OFF	OFF
	1000≦ (S•) ≦2000	OFF	ON	OFF
	2000→2001	OFF	ON→OFF	OFF→ON
	2000←2001	OFF	OFF→ON	ON→OFF
(S•) < (S2•)	2000→2001	OFF	ON→OFF	OFF→ON
	2000←2001	OFF	OFF→ON	ON→OFF
	(S•) >2000	OFF	OFF	ON

相关指令

和高速计数器一起使用的指令如下所示。

指令	FNC No.	指令名称
DHSCS	FNC 53	比较置位(高速计数器)
DHSCR	FNC 54	比较复位(高速计数器)
DHSZ	FNC 55	区间比较(高速计数器)
DHCMOV	FNC 189	高速计数器的传送
DHSCT	FNC 280	高速计数器的表格比较

注意要点

可编程控制器上使用该指令时，硬件计数器

1. 计数比较方法的选定

1) A8/A8C可编程控制器

在A8/A8C可编程控制器上使用该指令时，硬件计数器(C235、C236、C237、C238、C239、C240、C244(OP)、C245(OP)、C246、C248(OP)、C251、C253)会自动切换成软件计数器，并影响各计数器的最高频率以及综合频率。

请参考下面的计数动作后，根据具体控制内容来选择是使用该指令，还是使用通用的比较指令。

a) 使用该指令时的要点

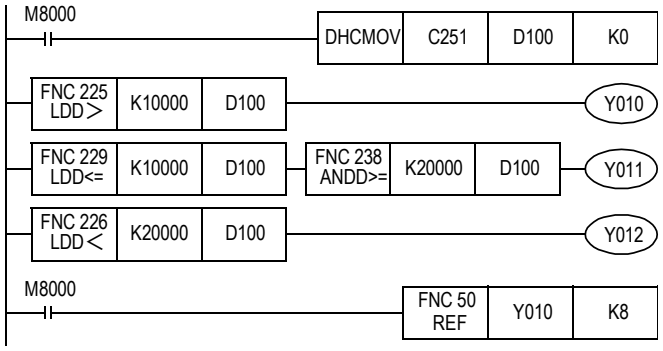
－不受可编程控制器扫描时间的影响， 想要计数一致时输出的时候使用

b) 使用通用的比较指令时的要点

－频率超出软件计数器的计数能力时(硬件计数器的计数能力以下)

－认为计数重要， 但是使用了计数结果的动作可以忽略扫描时间的影响

－超出指令的使用次数(32次)时



2. 软元件的指定范围
(S·)中可以指定的软元件，仅高速计数器(C235～C255)有效。

3. 只可以使用32位运算指令
由于高速计数器用的指令是32位专用指令，所以请务必输入“DHSZ”(FNC 55)。

4. 比较值1 (S1·) 和比较值2 (S2·) 的设定数据值的注意事项
请设定成(S1·) ≦ (S2·)。

5. 比较时序和输出结果之间的关系
1) DHSZ指令仅在高速计数器的计数脉冲被输入时执行比较，然后使结果输出置ON。
((S1·)为1000，(S2·)为1999时，当C235的当前值从999变为1000或从1999变为2000时，输出(D·)立即被置ON。)
2) 电源ON时或STOP→RUN时，由于得不到比较结果，因此即使符合比较条件结果也不输出。
→ 有关详细内容，请参考[13.6.1 电源ON时比较结果置ON的程序[ZCP(FNC 11)指令]]

6. 关于针对相同的高速计数器执行HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55)指令的优先动作顺序
→ 详细内容，请参考13.4.1项6

7. 通过外部端子执行复位动作
→ 详细内容，请参考13.4.1项5

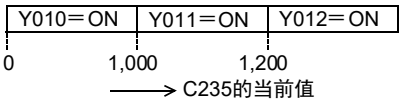
8. 软元件的占用点数
1) 比较值占用(S1·)、(S2·)起始各 点。
2) 输出占用(D·)起始的 点。

13.6.1 电源ON时比较结果置ON的程序[ZCP(FNC 11)指令]

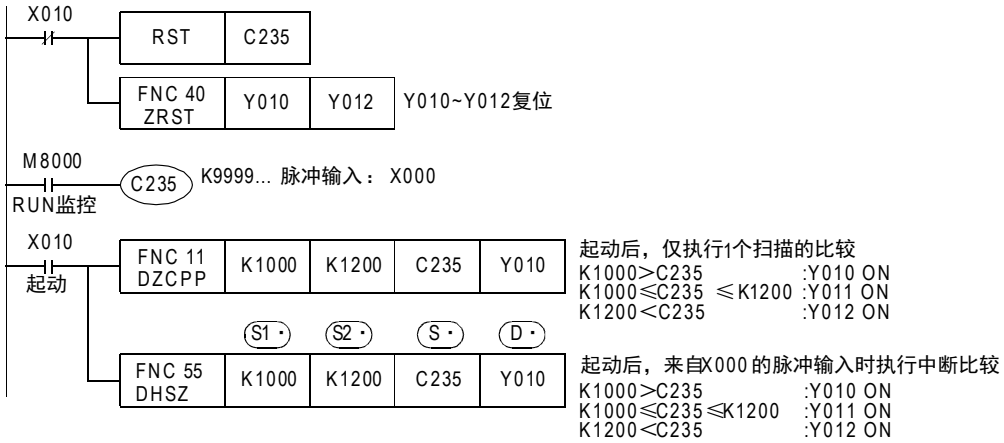
DHSZ指令仅在计数脉冲被输入时，输出比较结果。C235的当前值即使为0，起动时Y010也会保持之前的OFF的状态。
为了执行Y010的初始化动作，使用普通的区间比较指令DZCPP，通过仅在起动时的脉冲执行，比较C235的当前值和K1000、K1200后，先驱动Y010。
例举下面的程序举例进行说明。

动作说明

输出Y010～Y012如下所示。



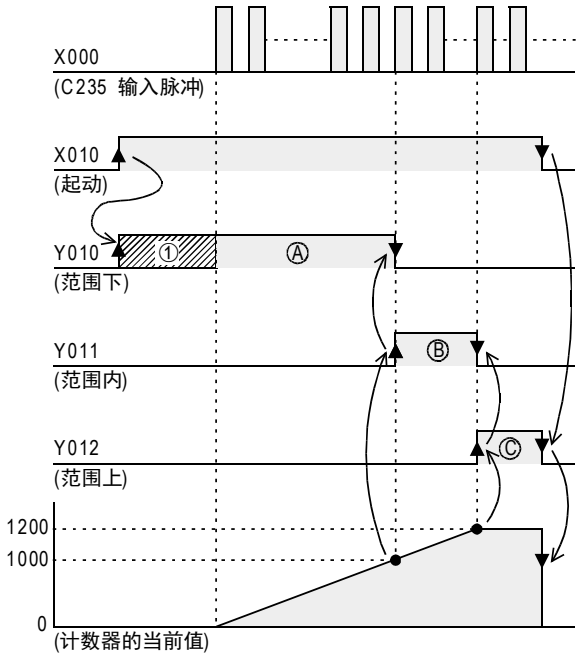
程序举例



时序图

在时序图中的①中，电源ON时高速计数器(例如: C235)的当前值为0时，Y010仍然保持OFF状态。

1) 为了执行 Y010 的初始化动作，使用普通的区间比较指令 DZCPP，通过仅在 RUN 时的脉冲执行，比较C235的当前值和K1000，K1200后，先驱动Y010。
2) 一直到输入脉冲输入，采用DHSZ指令的比较输出被驱动之前，Y010的比较结果都被保持。
3) 根据计数器的当前值，DHSZ指令驱动(A)、(B)或(C)的输出。



13.6.2 表格高速比较模式(M8130)

A1 A2 A2C A8 A8C

说明了关于DHSZ指令的表格高速比较模式(高速模型输出)。
想要一次执行多个输出动作时，请使用最大可以使16点输出变化的HSCT指令。

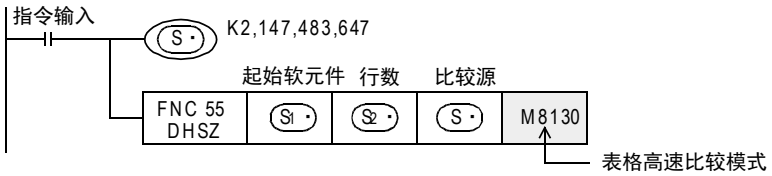
1. 设定数据

操作数种类	内容	数据类型
(S1•)	保存表格数据的起始字软元件编号(仅指数据寄存器D)	BIN32位
(S2•)	表格的行数(仅K、H)K1～K128/H1～H80	BIN32位
(S•)	高速计数器的软元件编号[C235～C255]	BIN32位
(D•)	M8130(表格高速比较模式声明用的特殊辅助继电器)	位

功能和动作说明

1. 32位运算(DHSZ)

可以通过指定DHSZ指令的(D•)为表格高速比较模式声明用的特殊辅助继电器M8130，实现以下的特殊功能。



比较表格

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

1) 在(S1•)中指定比较表格用软元件的起始编号。
由于比较表格用的数据每1行占用4个软元件，因此占用以(S1•)起始的(S2•)×4点。

2) (S2•)中指定比较表格的行数。
制作的表格从(S1•)中指定的起始寄存器开始，行数为(S2•)中指定的行数。

3) 比较数据
比较数据请使用32位数据。

4) 输出Y编号
输出Y编号，请用16进制数指定。
例如) 指定Y010时，为H10。
指定Y020时，为H20。

5) 置位/复位的指定
该置位、复位通过中断直接控制。

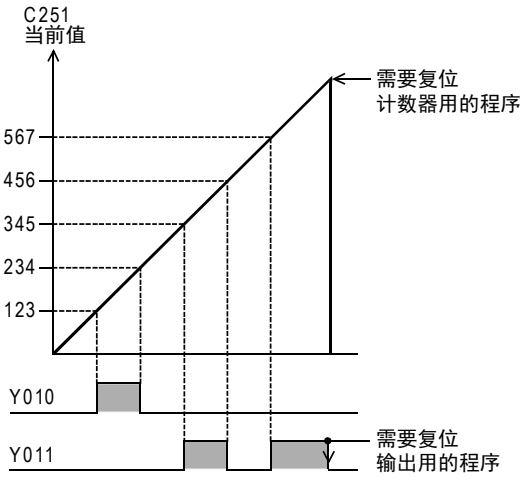
	设定内容
置位(ON)	K1/H1
复位(OFF)	K0/H0

2. 动作


M8002
FNC 12 DMOV K123 D200 比较数据
FNC 12 MOV H10 D202 输出Y编号
FNC 12 MOV K1 D203 输出的SET/RST K1：置位。
FNC 12 DMOV K234 D204 比较数据
FNC 12 MOV H10 D206 输出Y编号
FNC 12 MOV K0 D207 输出的SET/RST K0：复位。
FNC 12 DMOV K345 D208 比较数据
FNC 12 MOV H11 D210 输出Y编号
FNC 12 MOV K1 D211 输出的SET/RST K1：置位。
FNC 12 DMOV K456 D212 比较数据
FNC 12 MOV H11 D214 输出Y编号
FNC 12 MOV K0 D215 输出的SET/RST K0：复位。
FNC 12 DMOV K567 D216 比较数据
FNC 12 MOV H11 D218 输出Y编号
FNC 12 MOV K1 D219 输出的SET/RST K1：置位。
指令输入
C251
K2,147,483,647
起始软元件 行数 比较源
FNC 55 DHSZ D200 K5 C251 M8130
表格高速比较模式
复位指令输入
RST C251
RST Y011

比较表格

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



1) 执行该指令后，表格数据的最上一行的表格被设置成比较对象数据。
2) 高速计数器C251的当前值和比较对象的表格数据一致时，比较对象的表格数据中指定的输出Y编号被执行SET/RST。
   并且，这个输出Y不等到执行END指令的输出刷新，直接执行输出处理。
3) 表格计数器(D8130)的当前值被[+1]。
4) 比较对象的表格数据移到下一行的表格。
5) 在表格计数器(D8130)的当前值变为4之前，重复2)～3)的动作。
   当前值变为4时，返回到动作1)，表格计数器复位(D8130=0)。
   此时，结束标志位M8131为ON。
6) 指令触点断开(指令执行OFF)后，中止执行指令，表格计数器(D8130)复位(D8130=0)。

注意要点

1. 指令的使用次数的限制

这个指令，在程序中只能编程1次。
此外，与用于其他用途的DHSCS,DHSCR,DHSZ,DHSCT指令配合，可以同时驱动的指令限定在32个以下。

2. 中途断开指令输入的情况

中断指令的执行，表格计数器D8130被复位(K0)。
但是，在此之前，被置位/复位的输出会保持其状态。

3. 有关输出的开始日期

在该指令首次执行后的END指令处完成表格的制作，此后就为有效。
因此，输出从第2次扫描开始动作。

4. 有关高速计数器的当前值

作为动作对象的高速计数器的当前值小于比较表格的第1行的值，请务必从此处开始执行该指令。

13.6.3 频率控制模式(HSZ，PLSY指令)(M8132)

A1 A2 A2C A8 A8C

将频率控制模式申明用辅助继电器M8132指定为DHSZ指令的 (D・)，通过与DPLSY指令组合使用，具有以下的特殊功能。

此时， (S1・) 只对应数据寄存器D，(S2・) 只对应K、H，被限制为1≤K，H≤128。

此外， (S・) 可以指定为高速计数器C235～C255。

频率控制模式(非区间比较。)

PLSY指令，如下页所示，仅脉冲输出可在用户侧变更。

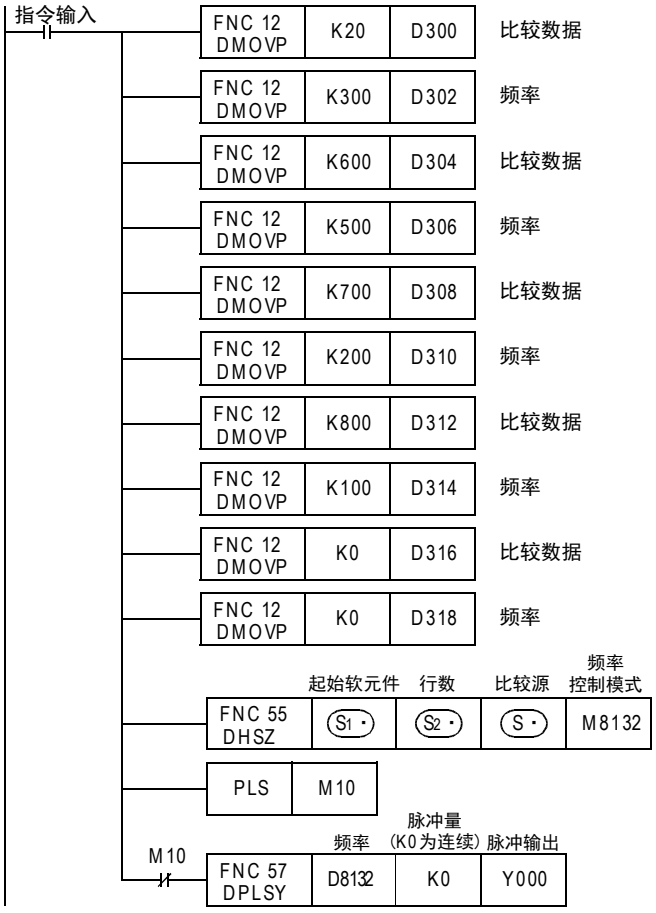
1. 控制例子

表格的构成及数据设定实例

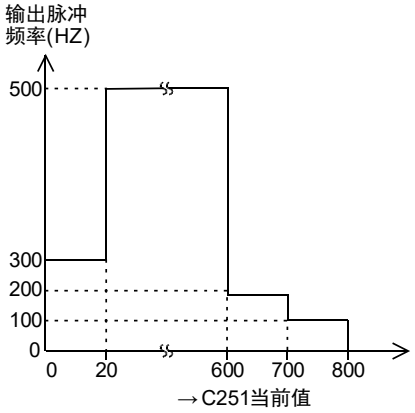
比较数据	频率	表格计数器(D8131)
D301、D300 K20	D302、D303 K300	0 ↓
D305、D304 K600	D306、D307 K500	1 ↓
D309、D308 K700	D310、D311 K200	2 ↓
D313、D312 K800	D314、D315 K100	3 ↓
D317、D316 K0	D318、D319 K0	4 ↓

(S1・) 中指定的起始软元件(32位)

(S2・) 中指定的行数



输出脉冲特性



1) 在构成表格的数据寄存器中，像本程序举例中一样，预先写入定下的数据。
2) 一起到(S·)中指定的高速计数器的当前值与(D301，D300)的值相等为止，PLSY指令的输出频率都为(D303，D302)的值。(D302为低16位。D303为高16位，一般一直为0。)
3) 此后，进入第2行的动作，依次执行各行的动作。
4) 执行完最后一行的动作后，结束标志位M8133动作，返回到第1行重复动作。
5) 希望在最后一行中停止动作时，将最后的表格的频率设定为K0。
6) 指令输入OFF后，脉冲输出变为OFF，表格计数器D8131也被复位。
7) 在该指令初次执行后的END指令处完成表格的制作，此后该指令有效。
8) 因此，PLSY指令的执行，是从指令输入为ON后的第2次扫描开始执行的，因此使用PLS M10的触点。

可以如本例所示通过程序向表格中写入数据，也可以通过外围设备的键盘操作直接写入。

1) M8132: 频率控制模式申明用的特殊辅助继电器。
2) D8132: 频率控制模式时，针对表格计数器D8131的递增计数，依次将表格内的设定频率读入D8132中。
3) D8134(低位)、D8135(高位)
   在频率控制模式时，针对表格计数器的递增计数，依次读取表格内的比较数据。

注意要点

1) 该指令只能使用1次。
2) 加上用于其他用途的DHSCS(FNC 53)、DHSCR(FNC 54)、DHSZ(FNC 55)、DHSCT(FNC 280)指令，可以同时驱动的指令限定在32个以下。
3) 由于在执行END指令时才制作表格，所以到制作完成之前需要延迟PLSY(FNC 57)指令的执行。
4) 在HSZ指令驱动的过程中，请勿更改表格的数据。
5) 使用频率控制模式时，不能对Y000、Y001同时输出。

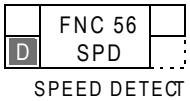
13.7 FNC 56 - SPD/脉冲密度

A1 A2 A2C A8 A8C

概要

采用中断输入方式对指定时间内的输入脉冲进行计数的指令。
根据版本不同，这个指令的功能也不同。

1. 指令格式



16位指令	指令记号	执行条件
7步	SPD	连续执行型

32位指令	指令记号	执行条件
13步	DSPD	连续执行型

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)	▲1																		●					
(S2•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D•)												●	●	●	●		●	●	●					

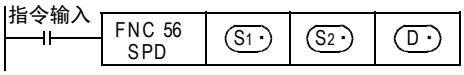
▲1: 可以指定X000～X007。
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

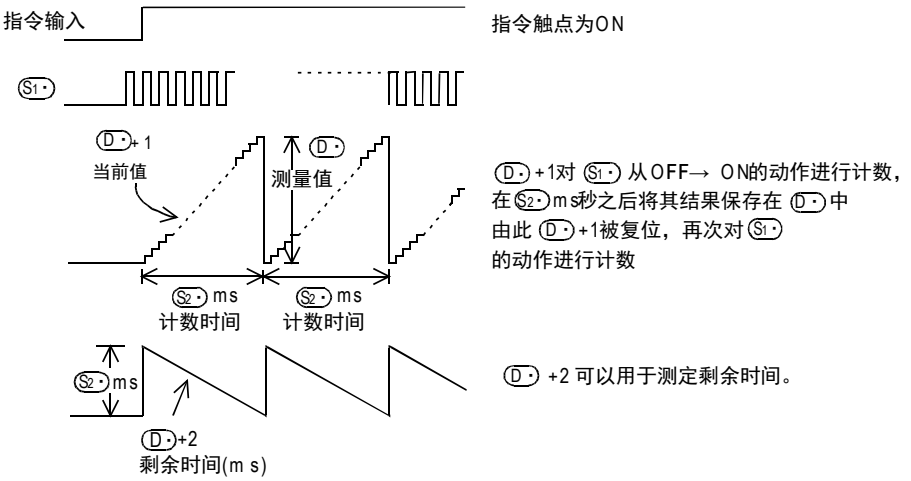
1. 16位运算(SPD)

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

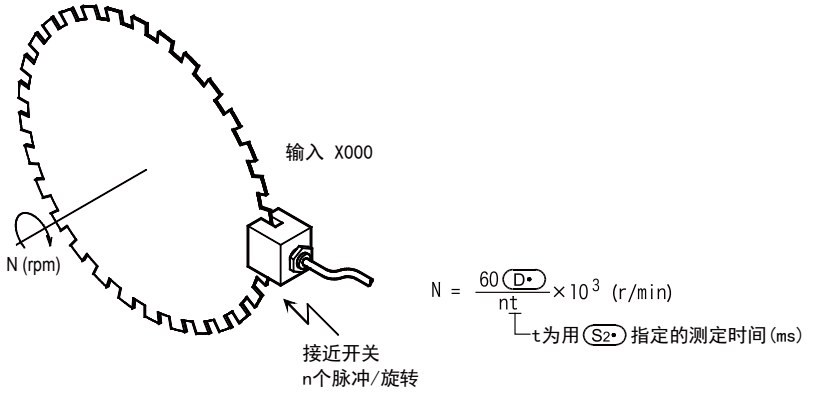
重复这个操作，可以在测量值 (D·) 中，得到脉冲密度(也就是与转速成比例的值)。



1) 时序图



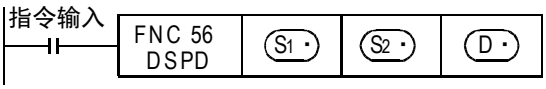
2) 测量值 (D·) 的值，如下所示与转数成比例。



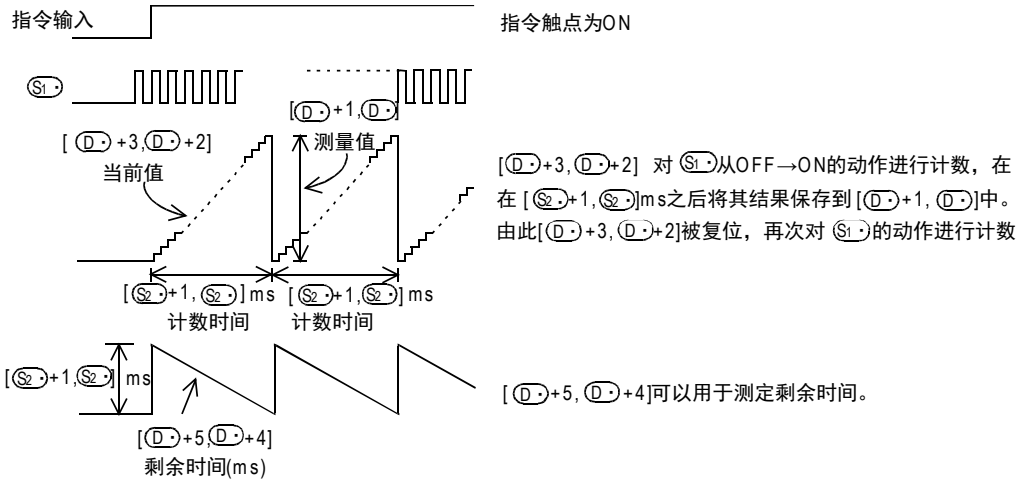
2. 32位运算(DSPD)

A1 A2 A2C A8 A8C

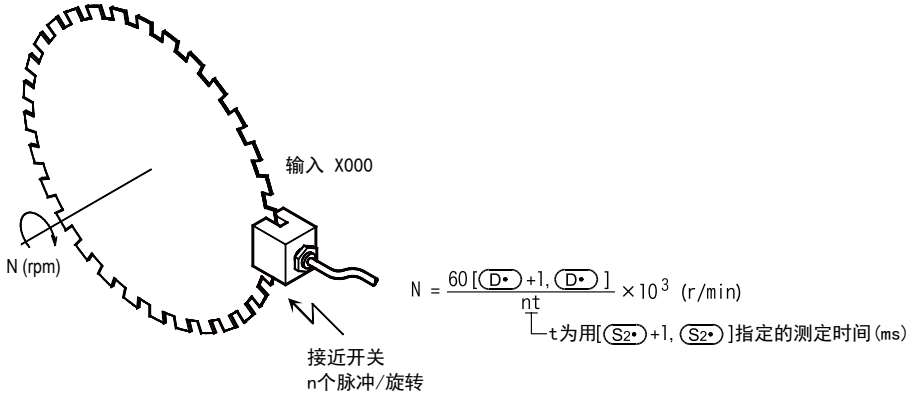
只在[(S2・)+1、(S2・)]× 1ms时间内对输入(S1・)的脉冲进行计数，测量值保存到[(D・)+1、(D・)]，当前值保存到[(D・)+3、(D・)+2]，剩余时间保存到[(D・)+5、(D・)+4](ms)中。
重复这个操作，可以在测量值[(D・)+1，(D・)]中，得到脉冲密度(也就是与转速成比例的值)。



1) 时序图



2) [(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
   - 指定的(S1•)输入的X000～X007不能与下面的用途重复使用。
     - 高速计数器
     - 输入中断
     - 脉冲捕捉
     - DSZR指令
     - DVIT指令
     - ZRN指令
   - 每个输入点只能使用一条该指令
   - 输入X000～X007的ON/OFF的最大频率，如下表所示。
     - A8/A8C可编程控制器

使用的输入编号	最大输入频率		
	A8C可编程控制器	A8可编程控制器	
		基本单元	A8-4HSX-ADP
X000～X005	100kHz*1	100kHz*1	200kHz
X006,X007	10kHz	10kHz	

*1. 希望读取相当于50k～100kHz的响应频率的脉冲时，请务必设置如下。
   - 接线长度请勿超出5m。
   - 输入端子中连接1.5KΩ(1W以上)的漏电阻，对方设备一侧的开集电极型晶体管的输出负载电流要在20mA以上。

2. 占用软元件
   1) 16位运算时
      以(D•)为起始点占用
   2) 32位运算时
      以(D•)为起始点占用

3. (S2•)中指定字软元件的情况
   如果在指令执行过程中更改字软元件的值时，按照测量周期反映到操作中。

根据版本不同的功能变更

根据版本不同，这个指令的功能如下表变更。

对应版本		项目	功能概要
A8	A8C		
Ver.2.20 以上	Ver.2.20 以上	增加32位指令	允许32位运算(DSPD)。

13.8 FNC 57 - PLSY/脉冲输出

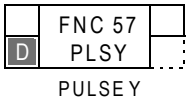
A1 A2 A2C A8 A8C

概要

发出脉冲信号用的指令。

→ 关于频率控制模式，请参考13.6.3项

1. 指令格式



16位指令	指令记号	执行条件
7步	PLSY	连续执行型

32位指令	指令记号	执行条件
13步	DPLSY	连续执行型

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D・)		▲1																	●					

▲1: 请对基本单元的晶体管输出，或是高速输出特殊适配器*1的Y000、Y001做指定。

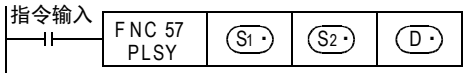
▲2: 仅对应A8/A8C可编程控制器。

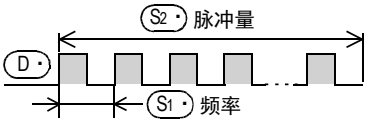
*1. 高速输出特殊适配器只能连接到A8可编程控制器。

功能和动作说明

1. 16位运算(PLSY)

从输出Y(D・)中输出(S1・)个频率为(S2・)的脉冲串。

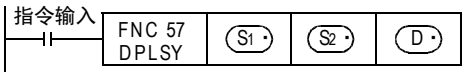


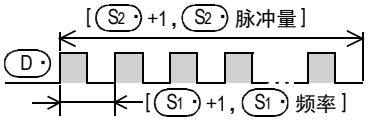


- (S1・)中指定频率。 允许设定范围: 1～32767(Hz)
- (S2・)中指定发出的脉冲量。 允许设定范围: 1～32767(PLS)
- (D・)在中指定有脉冲输出的Y编号。 允许设定范围:Y000 、Y001

2. 32位运算(DPLSY)

输出Y (D·) 中输出[(S1·) +1、 (S1·)]个频率为[(S2·) +1、 (S2·)]的脉冲串。





- 在[(S1·) +1、 (S1·)]中指定频率。
  - －使用高速输出特殊适配器时
    允许设定范围: 1～200,000(Hz)
  - －在使用A8/A8C可编程控制器基本单元时
    允许设定范围: 1～100,000(Hz)
- 在[(S2·) +1、 (S2·)]中指定发出的脉冲量。 允许设定范围: 1～2,147,483,647(PLS)
- (D·) 在中指定有脉冲输出的Y编号。 允许设定范围:Y000 、Y001

→ 有关无限制输出脉冲的方法，参考程序举例

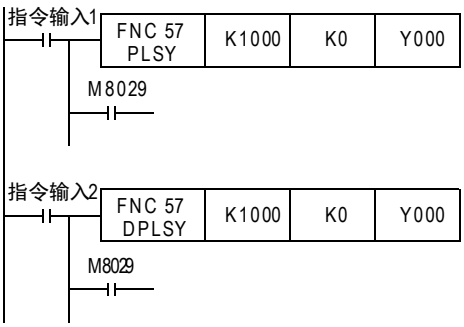
相关软元件

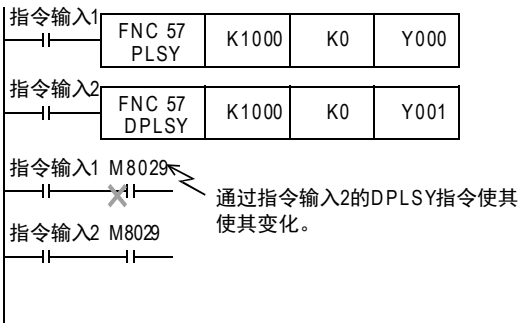
1. 指令执行结束标志位

指令执行结束的标志位，在其他指令中也使用相同的标志位(M8029)。使用了使标志位变化的其他指令和多个PLSY(FNC 57)指令时，请务必在要监视的指令的正下方使用。

→ 有关指令执行结束的标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	ON ：指定的脉冲数的发生结束 OFF : 不到指定脉冲量时的中断以及停止发出连续脉冲时



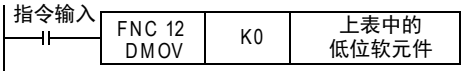


2. 输出脉冲数的当前值监控

从Y000、Y001输出的脉冲数会被保存在下面的特殊数据寄存器中。

软元件		内容	指令名称
高位	低位		
D8141	D8140	Y000的输出脉冲数累计	使用PLSY指令，PLSR指令从Y000输出的脉冲数的累计
D8143	D8142	Y001的输出脉冲数累计	使用PLSY指令，PLSR指令从Y001输出的脉冲数的累计
D8137	D8136	Y000、Y001输出脉冲数的合计累计数	使用PLSY指令，PLSR指令从Y000和Y001输出的脉冲数的合计累计数

可以用下面的程序清除各数据寄存器的内容。



3. 停止脉冲输出
   - 指令输入OFF后，会即刻停止输出，再次置ON后，从最初开始运行。
   - 下面的特殊辅助继电器(M)置ON后输出会停止。

软元件 A8/A8C	内容
M8349	停止Y000脉冲输出(即刻停止)
M8359	停止Y001脉冲输出(即刻停止)

再次输出脉冲时，如果与输出信号相对应的软元件
A8/A8C: M8349、M8359)OFF后，请对脉冲输出指令执行OFF→ON操作后再次驱动。

注意要点

1. (S1・) 、 (S2・) 中指定字软元件的情况
   如果在指令执行过程中更改字软元件的值时，指令的动作变为如下所示的情况。
   - (S1・) 中的数据被更改时，输出频率也随之相应改变。
   - (S2・) 被更改时，从指令下一次被驱动开始变更内容生效。

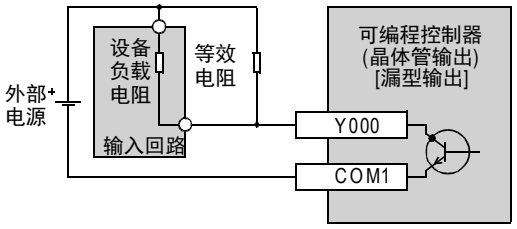
2. 频率 (S1・)
   使用基本单元的晶体管输出时，请将输出频率 (S1・) 设定在100,000Hz以下。
   用超出100,000Hz频率的脉冲使负载运行时，可编程控制器有时候会出现故障。

3. 有关脉冲输出
   - (D・) 中可以指定的软元件，只有基本单元的晶体管输出，以及高速输出特殊适配器*1的Y000、Y001有效。
     在继电器输出型的A8可编程控制器中使用PLSY(FNC 57)指令时，需要使用高速输出特殊适配器。
     *1. 高速输出特殊适配器只能连接到A8可编程控制器。
   - 脉冲的ON/OFF时间占空比为50%。(即ON=50%、OFF=50%)
   - 采用不受顺控程序(运算周期)影响的专用硬件来执行输出控制。
   - 发出连续脉冲时，指令输入若为OFF，则 (D・) 输出为OFF。

4. A8/A8C系列基本单元的脉冲输出端子的使用
   输出Y000和Y001为高速响应输出。
   使用脉冲输出指令和定位相关指令时，请将开集电极型晶体管输出的负载电流调节在10～100mA (DC5～24V)。

项目	内容
使用电压范围	DC5～24V
使用电流范围	10～100mA
输出频率	100kHz以下

   负载较轻的场合，如下图中的回路所示，请在使用的输出端子(Y000或Y001)的外部并联等效电阻，以确保输出晶体管中流过的电流符合上述的规定电流。



5. 使用高速输出特殊适配器时的注意事项

1) 高速输出特殊适配器的输出为差动线性驱动。

2) 高速输出特殊适配器的脉冲输出方式的设定开关，请设为[脉冲串+方向](PLS・DIR)一侧。
如设定为[正转脉冲串，反转脉冲串](FP・RP)一侧，则在下表所示的输出状态下，输出目标地址会改变，因此不能正常动作。

脉冲输出对象	对操作有影响的输出	动作
(D•) =Y000	Y004	Y004为ON时，从高速输出适配器的Y000输出脉冲。 Y004为OFF时，从高速输出适配器的Y004输出脉冲。
(D•) =Y001	Y005	Y005为ON时，从高速输出适配器的Y001输出脉冲。 Y005为OFF时，从高速输出适配器的Y005输出脉冲。

3) 请在可编程控制器STOP时或是电源OFF时，操作脉冲输出方式的设定开关。
在脉冲发出过程中，请勿操作脉冲输出方式的设定开关。

4) 连接了高速输出特殊适配器时，按照下表内容所示分配与基本单元相同的输出编号。
输出端子仅使用两者之一，不使用的输出端子上请勿接线。
高速输出特殊适配器和基本单元的输出，如下所示动作。

高速输出特殊适配器的输出编号分配

输出方式设定开关的状态	信号名称	各定位指令的设定名称	输出编号			
			第1台		第2台	
			第1轴	第2轴	第3轴	第4轴
FP RP一侧	正转脉冲串(FP)	脉冲输出对象	Y000	Y001	Y002	Y003
	反转脉冲串(RP)	旋转方向的信号	Y004	Y005	Y006	Y007
PLS・DIR一侧	脉冲串	脉冲输出对象	Y000	Y001	Y002	Y003
	方向	旋转方向的信号	Y004	Y005	Y006	Y007

输出的动作

	输出的动作
继电器输出型的基本单元	指令动作中，相应的输出也变为ON状态。(LED也为ON状态) 请使用高速输出特殊适配器。
高速输出特殊适配器	执行ON/OFF操作。 输出频率不能超出200kHz。
晶体管输出型的基本单元	执行ON/OFF操作。 输出频率不能超出100kHz。

6. 其他

1) 脉冲输出指令和定位指令的种类及对象输出编号

区分	指令	指令名称	对象输出编号
脉冲输出	PLSY(FNC 57)	脉冲输出	Y000、Y001
	PLSR(FNC 59)	带加减速的脉冲输出	Y000、Y001
定位	DSZR(FNC 150)	带DOG搜索的原点回归	Y000、Y001、Y002、Y003*1
	DVIT(FNC 151)*2	中断定位	Y000、Y001、Y002、Y003*1
	ZRN(FNC 156)	原点回归	Y000、Y001、Y002、Y003*1
	PLSV(FNC 157)	可变速脉冲输出	Y000、Y001、Y002、Y003*1
	DRVI(FNC 158)	相对定位	Y000、Y001、Y002、Y003*1
	DRVA(FNC 159)	绝对定位	Y000、Y001、Y002、Y003*1
高速处理	PWM(FNC 58)	脉宽调制	Y000、Y001、Y002、Y003*1

*1. 只有在A8可编程控制器中使用2台高速输出适配器时，才可以指定脉冲输出对象Y003。

*2. 仅对应A8/A8C可编程控制器。

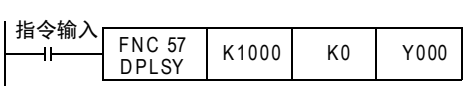
2) 多个指令下使用同一输出(Y000、Y001)的情况
脉冲输出过程中监控(BUSY/READY)的标志位置ON时，不能执行使用了相同输出的脉冲输出指令和定位指令。
因此，即使指令驱动触点为OFF，只要是脉冲输出过程中监控的标志位为ON，就请勿执行指定了同一输出编号的定位指令和脉冲输出指令。
被驱动时，当脉冲输出监控的标志位为OFF后，请经过一次运算周期以上后再次执行指令。

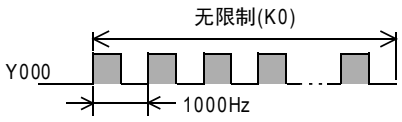
脉冲输出对象软元件	脉冲输出过程中监控的标志位
Y000	M8340
Y001	M8350

3) 组合了DHSZ(FNC 55)指令和PLSY(FNC 57)指令的「频率控制模式」，在程序中只可以使用1次。

程序举例(希望脉冲的输出数量没有限制时)

(S2・)设定为K0时，可以无限制发出脉冲。





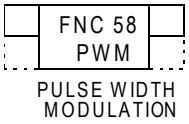
13.9 FNC 58 - PWM/脉宽调制

A1 A2 A2C A8 A8C

概要

指定了脉冲的周期和ON时间的脉冲输出的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	PWM	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D•)		▲1																	●					

▲1: 请指定基本单元的晶体管输出 Y000、Y001、Y002 或是高速输出特殊适配器 *1 的 Y000、Y001、Y002、Y003。

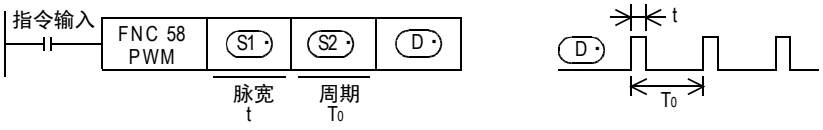
▲2: 仅对应A8/A8C可编程控制器。

*1. 高速输出特殊适配器只能连接到A8可编程控制器。

功能和动作说明

1. 16位运算(PWM)

以周期[(S2•) ms]单位输出ON脉冲宽度为(S1•) ms的脉冲。



- 在(S1•) 中指定脉宽t。
  设定范围: 0～32,767ms
- 在(S2•) 中指定周期T0。
  设定范围: 1～32,767ms
- 在(D•) 中指定输出脉冲的Y编号。
  设定范围:Y000 、Y001、Y002、Y003

注意要点

1. 设定脉宽及周期的时间
   脉宽 (S1・) 和周期 (S2・) 的值，请设定为 (S1・) ≦ (S2・)。

2. 有关脉冲输出
   - 可以在 (D・) 中指定的软元件，根据系统配置情况仅有下列的输出有效。
     －使用高速输出特殊适配器*1时 Y000、Y001、Y002*2、Y003*2
     －使用基本单元的晶体管输出时
     :Y000 、Y001、Y002*3

     *1. 高速输出特殊适配器只能连接到A8可编程控制器。
     在继电器输出型的A8可编程控制器中使用PWM(FNC 58)指令时，需要高速输出特殊适配器。

     *2. 指定高速输出特殊适配器的Y002、Y003时，需要第2个高速输出特殊适配器。

   - 采用不受顺控程序(运算周期)的影响的中断处理方式来执行输出控制。
   - 指令输入为OFF时，则 (D・) 输出也为OFF。
   - 脉冲输出过程中监控 (BUSY/READY) 的标志位置 ON 时，不能执行使用了相同输出的脉冲输出指令和定位指令。
     因此，即使指令驱动触点为OFF，只要是脉冲输出过程中监控的标志位为ON，就请勿执行指定了同一输出编号的定位指令和脉冲输出指令。
     被驱动时，当脉冲输出监控的标志位为OFF后，请经过1个扫描周期以上后再次执行指令。

脉冲输出对象软元件	脉冲输出过程中监控的标志位
Y000	M8340
Y001	M8350
Y002	M8360
Y003	M8370

3. 使用高速输出特殊适配器时的注意事项
   1) 高速输出特殊适配器的输出为差动线性驱动。
   2) 高速输出特殊适配器的脉冲输出方式的设定开关，请设为[脉冲串+方向](PLS・DIR)一侧。
   [正转脉冲串、反转脉冲串](FP・RP)一侧，则在下表所示的输出状态下，输出目标地址会改变，因此不能正常动作。

脉冲输出对象	对动作有影响的输出	动作
(D・) =Y000	Y004	Y004为ON时，从高速输出适配器的Y000输出脉冲。 Y004为OFF时，从高速输出适配器的Y004输出脉冲。
(D・) =Y001	Y005	Y005为ON时，从高速输出适配器的Y001输出脉冲。 Y005为OFF时，从高速输出适配器的Y005输出脉冲。
(D・) =Y002	Y006	Y006为ON时，从高速输出适配器的Y002输出脉冲。 Y006为OFF时，从高速输出适配器的Y006输出脉冲。
(D・) =Y003	Y007	Y007为ON时，从高速输出适配器的Y003输出脉冲。 Y007为OFF时，从高速输出适配器的Y007输出脉冲。

   3) 请在可编程控制器STOP时或是电源OFF时，操作脉冲输出方式的设定开关。
   在脉冲发出过程中，请勿操作脉冲输出方式的设定开关。

4) 连接了高速输出特殊适配器时，按照下表内容所示分配与基本单元相同的输出编号。
输出端子仅使用两者之一，不使用的输出端子上请勿接线。
高速输出特殊适配器和基本单元的输出，如下所示动作。

高速输出特殊适配器的输出编号分配

输出方式设定开关的状态	信号名称	各定位指令的设定名称	输出编号			
			第1台		第2台	
			第1轴	第2轴	第3轴	第4轴
FP ・RP一侧	正转脉冲串(FP)	脉冲输出对象	Y000	Y001	Y002	Y003
	反转脉冲串(RP)	旋转方向的信号	Y004	Y005	Y006	Y007
PLS・DIR一侧	脉冲串	脉冲输出对象	Y000	Y001	Y002	Y003
	方向	旋转方向的信号	Y004	Y005	Y006	Y007

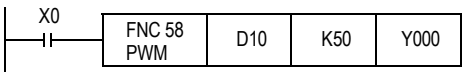
输出的动作

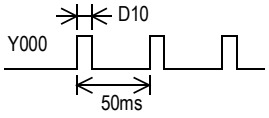
	输出的动作
继电器输出型的基本单元	执行ON/OFF动作，但是由于PWM(FNC 58)指令不适用于继电器输出，所以请勿使用。(请注意，由于输出的响应延迟，触点的振荡，触点寿命等的理由可以导致不能正常动作，因此请勿使用))
高速输出特殊适配器	执行ON/OFF操作。 输出频率不能超出200kHz。 请使用基本单元的晶体管输出型。
晶体管输出型的基本单元	执行ON/OFF操作。 输出频率不能超出100kHz。

程序举例

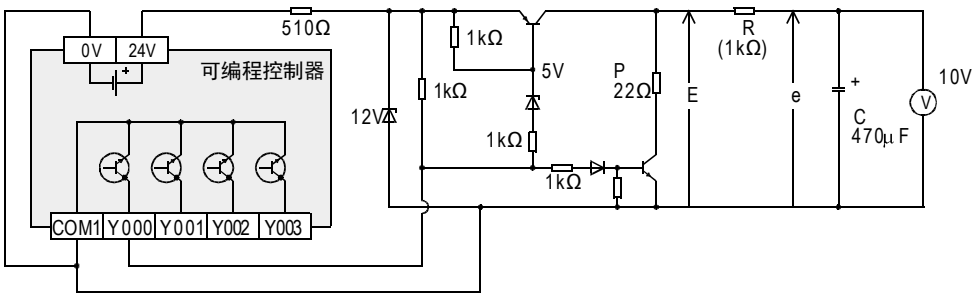
在下面例举的程序举例中，使D10的内容在0～50间变化时，Y000的平均输出为0～100%。
(D10)的内容超出50时会错误。
在下面的例子中，使用了A8系列的基本单元(漏型输出)。请根据所使用的可编程控制器的下述手册进行连接。

→ A8硬件篇手册
→A8C硬件篇手册





平滑回路的例子



R>>P
T=R(kΩ)×C(μF)=470ms>>To
滤波器的时间常数τ要远大于脉冲周期T0
平均输出电流e中的波动值Δe 大概为 $\frac{\Delta e}{e} \leqq \frac{To}{\tau}$

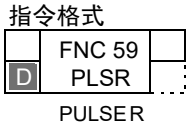
13.10 FNC 59 - PLSR/带加减速的脉冲输出

A1 A2 A2C A8 A8C

概要

带加减速功能的脉冲输出指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	PLSR	连续执行型

32位指令	指令记号	执行条件
17步	DPLSR	连续执行型

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S3・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D・)		▲1																	●					

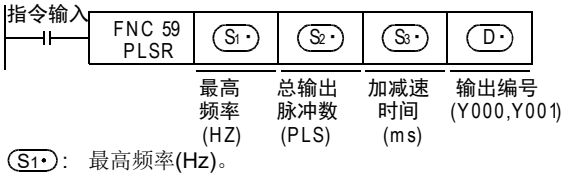
▲1: 请对基本单元的晶体管输出，或是高速输出特殊适配器*1的Y000、Y001做指定。
▲2: 仅对应A8/A8C可编程控制器。

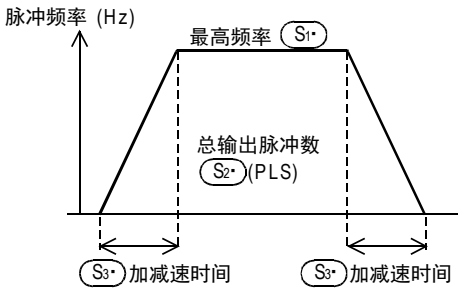
*1. 高速输出特殊适配器只能连接到A8可编程控制器。

功能和动作说明

1. 16位运算(PLSR)

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





(S1・): 最高频率(Hz)。
允许设定范围 : 10～32,767(Hz)

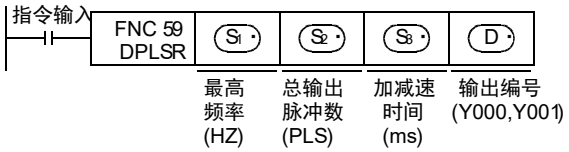
(S2・): 总输出脉冲数(PLS)。
允许设定范围
A8/A8C : 1～32,767(PLS)

(S3・): 加减速时间(ms)。
允许设定范围 : 50～5000(ms)

(D・): 脉冲输出信号
允许设定范围 : Y000 、Y001

2. 32位运算(DPLSR)

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



[(S1•) +1、 (S1•)]: 最高频率(Hz)。

－ 使用高速输出特殊适配器时

允许设定范围: 10～200,000(Hz)

－在使用A8/A8C可编程控制器基本单元时，

允许设定范围: 10～100,000(Hz)

[(S2•) +1、 (S2•)]: 总输出脉冲数(PLS)。

允许设定范围:

A8/A8C : 1～2,147,483,647(PLS)

[(S3•) +1、 (S3•)]: 加减速时间(ms)

允许设定范围: 50～5000(ms)

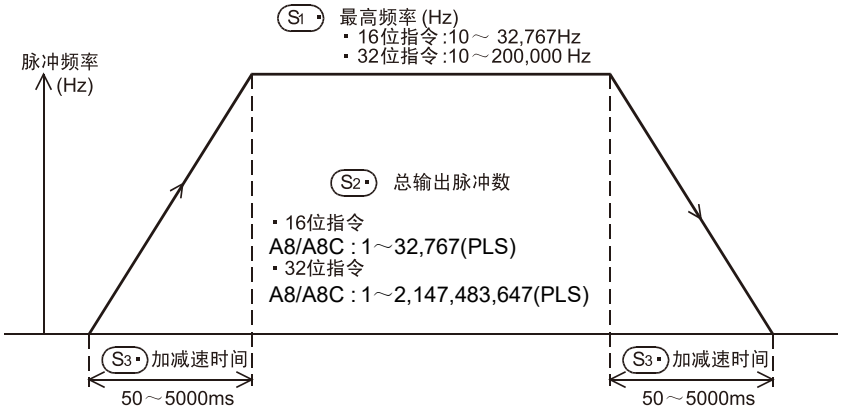
(D•): 脉冲输出信号

允许设定范围:Y000 、Y001

3. 有关脉冲输出的规格

- 简易的定位(带加减速功能)

运行型式如下图所示。



- 有关输出处理

采用不受运算周期影响的专用硬件来执行输出控制。

- 关于指令运行中的数据更改

即使替换了操作数，在运行中也不反映。在下一次的指令驱动时， 更改内容有效。

相关软元件

1. 指令执行结束标志位

→ 关于指令执行结束标志位的使用方法，请参考6.5.2项

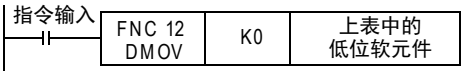
软元件	名称	内容
M8029	指令执行结束	OFF: 指令输入为OFF或脉冲输出过程中(在输出脉冲中途中断时，不置ON。) ON: (S2•) 中设定的脉冲数的输出结束

2. 输出脉冲数的监控

从Y000、Y001输出的脉冲数会被保存在下面的特殊数据寄存器中。

软元件		内容	指令名称
高位	低位		
D8141	D8140	Y000的输出脉冲数累计	使用PLSY指令，PLSR指令从Y000输出的脉冲数的累计
D8143	D8142	Y001的输出脉冲数累计	使用PLSY指令，PLSR指令从Y001输出的脉冲数的累计
D8137	D8136	Y000、Y001输出脉冲数的合计累计数	使用PLSY指令，PLSR指令从Y000和Y001输出的脉冲数的合计累计数

可以用下面的程序清除各数据寄存器的内容。



3. 停止脉冲输出

- 指令输入OFF后，会即刻停止输出，再次置ON后，从最初开始运行。
- 下面的特殊辅助继电器(M)置ON后输出会停止。

软元件	内容
A8/A8C	
M8349	停止Y000脉冲输出(即刻停止)
M8359	停止Y001脉冲输出(即刻停止)

再次输出脉冲时，如果与输出信号相对应的软元件
A8/A8C : M8349、M8359)OFF后，请对脉冲输出指令执行OFF→ON操作后再次驱动。

注意要点

1. 最高频率 (S1•)

使用基本单元的晶体管输出时，最高频率 (S1•) 请设定在100,000Hz以下。
用超出100,000Hz频率的脉冲使负载运行时，可编程控制器有时候会出现故障。

2. 有关脉冲输出

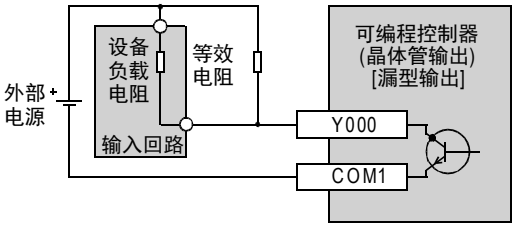
- (D•) 中可以指定的软元件，只有基本单元的晶体管输出，以及高速输出特殊适配器*1的Y000、Y001有效。
  *1. 高速输出特殊适配器只能连接到A8可编程控制器。
  在继电器输出型的A8可编程控制器中使用PLSY(FNC 59)指令时，需要高速输出特殊适配器。
- 脉冲的ON/OFF时间占空比为50%。(即ON=50%、OFF=50%)
- 采用不受顺控程序(运算周期)影响的专用硬件来执行输出控制。
- 发出连续脉冲时，指令输入若为OFF，则 (D•) 输出为OFF。

3. A8/A8C系列基本单元的脉冲输出端子的使用

输出Y000和Y001为高速响应输出。
使用脉冲输出指令和定位相关指令时，请将开集电极型晶体管输出的负载电流调节在10～100mA(DC5～24V)。

项目	内容
使用电压范围	DC5～24V
使用电流范围	10～100mA
输出频率	100kHz以下

负载较轻的场合，如下图中的回路所示，请在使用的输出端子(Y000或Y001)的外部并联等效电阻，以确保输出晶体管中流过的电流符合上述的规定电流。



4. 使用高速输出特殊适配器时的注意事项

1) 高速输出特殊适配器的输出为差动线性驱动。

2) 高速输出特殊适配器的脉冲输出方式的设定开关，请设为[脉冲串+方向](PLS・DIR)一侧。
如设定为[正转脉冲串、反转脉冲串](FP・RP)一侧，则在下表所示的输出状态下，输出目标地址会改变，因此不能正常动作。

脉冲输出对象	对操作有影响的输出	动作
(D•) =Y000	Y004	Y004为ON时，从高速输出适配器的Y000输出脉冲。 Y004为OFF时，从高速输出适配器的Y004输出脉冲。
(D•) =Y001	Y005	Y005为ON时，从高速输出适配器的Y001输出脉冲。 Y005为OFF时，从高速输出适配器的Y005输出脉冲。

3) 请在可编程控制器STOP时或是电源OFF时，操作脉冲输出方式的设定开关。
在脉冲发出过程中，请勿操作脉冲输出方式的设定开关。

4) 连接了高速输出特殊适配器时，按照下表内容所示分配与基本单元相同的输出编号。
输出端子仅使用两者之一，不使用的输出端子上请勿接线。
高速输出特殊适配器和基本单元的输出，如下所示动作。

高速输出特殊适配器的输出编号分配

输出方式设定开关的状态	信号名称	各定位指令的设定名称	输出编号			
			第1台		第2台	
			第1轴	第2轴	第3轴	第4轴
FP・RP一侧	正转脉冲串(FP)	脉冲输出对象	Y000	Y001	Y002	Y003
	反转脉冲串(RP)	旋转方向的信号	Y004	Y005	Y006	Y007
PLS・DIR一侧	脉冲串	脉冲输出对象	Y000	Y001	Y002	Y003
	方向	旋转方向的信号	Y004	Y005	Y006	Y007

输出的动作

	输出的动作
继电器输出型的基本单元	指令动作中，相应的输出也变为ON状态。(LED也为ON状态) 请使用高速输出特殊适配器。
高速输出特殊适配器	执行ON/OFF操作。 输出频率不能超出200kHz。
晶体管输出型的基本单元	执行ON/OFF操作。 输出频率不能超出100kHz。

5. 其他

1) 脉冲输出指令和定位指令的种类及对象输出编号

区分	指令	指令名称	对象输出编号
脉冲输出	PLSY(FNC 57)	脉冲输出	Y000、Y001
	PLSR(FNC 59)	带加减速的脉冲输出	Y000、Y001
定位	DSZR(FNC 150)	带DOG搜索的原点回归	Y000、Y001、Y002、Y003*1
	DVIT(FNC 151)*2	中断定位	Y000、Y001、Y002、Y003*1
	ZRN(FNC 156)	原点回归	Y000、Y001、Y002、Y003*1
	PLSV(FNC 157)	可变速脉冲输出	Y000、Y001、Y002、Y003*1
	DRVI(FNC 158)	相对定位	Y000、Y001、Y002、Y003*1
	DRVA(FNC 159)	绝对定位	Y000、Y001、Y002、Y003*1
高速处理	PWM(FNC 58)	脉宽调制	Y000、Y001、Y002、Y003*1

*1. 只有在A8可编程控制器中使用2台高速输出适配器时，才可以指定脉冲输出对象Y003。

*2. 仅对应A8/A8C可编程控制器。

2) 多个命令下使用同一输出继电器(Y000，Y001)的情况

脉冲输出过程中监控(BUSY/READY)的标志位置ON时，不能执行使用了相同输出的脉冲输出指令和定位指令。

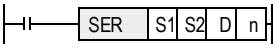
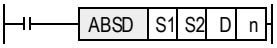
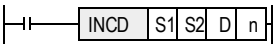
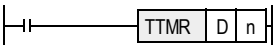
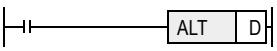
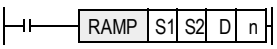
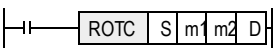
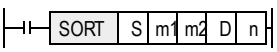
因此，即使指令驱动触点为OFF，只要是脉冲输出过程中监控的标志位为ON，就请勿执行指定了同一输出编号的定位指令和脉冲输出指令。

被驱动时，当脉冲输出监控的标志位为OFF后，请经过一次运算周期以上后再次执行指令。

脉冲输出对象软元件	脉冲输出过程中监控的标志位
Y000	M8340
Y001	M8350

14. 方便指令—FNC 60～FNC 69

在FNC 60～FNC 69中，提供了可以用最少的顺控程序实现复杂的控制的方便指令。

FNC No.	指令记号	符号	功能	参考
60	IST		初始化状态	14.1节
61	SER		数据检索	14.2节
62	ABSD		凸轮顺控绝对方式	14.3节
63	INCD		凸轮顺控相对方式	14.4节
64	TTMR		示教定时器	14.5节
65	STMR		特殊定时器	14.6节
66	ALT		交替输出	14.7节
67	RAMP		斜坡信号	14.8节
68	ROTC		旋转工作台控制	14.9节
69	SORT		数据排序	14.10节

14.1 FNC 60 - IST/初始化状态

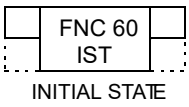
A1 A2 A2C A8 A8C

概要

在采用步进梯形图的程序中，对初始化状态以及特殊辅助继电器进行自动控制的指令。

→ 有关SFC编程及步进梯形图，请参考34章

1. 指令格式



16位指令	指令记号	执行条件
7步	IST	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	运行模式的切换开关的起始软元件编号	位
(D1・)	自动模式下实用状态的最小状态编号[(D1・) < (D2・)]	位
(D2・)	自动模式下实用状态的最大状态编号[(D1・) < (D2・)]	位

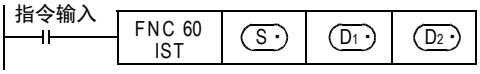
3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址				常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰		K	H	E	"□"	P
(S・)	●	●	●				▲1												●						
(D1・)						▲2													●						
(D2・)						▲2													●						

▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)

▲2: S20～S899、S1000～S4095

功能和动作说明



- 在(S・)中指定运行模式的起始输入。
  选择运行模式用的开关，占用从起始软元件(S・)开始的8点，分别分配下面的开关功能。
  如下表所示，分配X020时，为了防止X020～X024同时为ON，必须使用旋转开关。
  不使用的开关，无需接线，但是由于被IST指令占用了，所以不能用于其他用途。

源地址	软元件编号(例)	开关功能
(S・)	X020	各个操作
(S・)+1	X021	原点回归
(S・)+2	X022	步进
(S・)+3	X023	循环运行一次

源地址	软元件编号(例)	开关功能
(S・)+4	X024	连续运行
(S・)+5	X025	原点回归开始
(S・)+6	X026	自动开始
(S・)+7	X027	停止

- (D1・)中指定实用状态的最小编号。(自动模式用)
- (D2・)中指定实用状态的最大编号。(自动模式用)

1. 通过开关操作控制软元件(占用软元件)

指令输入为ON时，下面的软元件会被控制自动切换。指令输入变为OFF时，则不变化。

软元件编号	动作功能
M8040	禁止转移
M8041*1	转移开始
M8042	启动脉冲
M8043*1	原点回归结束
M8045	禁止所有输出复位
M8047*2	STL监控有效

软元件编号	动作功能
S0	各个操作的初始化状态
S1	原点回归的初始化状态
S2	自动运行的初始化状态

*1. 从RUN→STOP时清除

*2. 在执行END指令时处理

请勿将以下的状态作为普通的状态编程。

软元件编号	动作功能
S0～S9	作为初始化状态被占用 • S0～S2，作为如上所述的各个操作用、原点回归用、自动运行用被使用。 • S3～S9可以随意使用。
S10～S19	作为原点回归用被占用

原点回归完成(M8043)不置ON时，如果在各个操作(X020)、原点回归(X021)、自动(X022、X023、X024)与各开关功能之间进行切换，则所有输出变为OFF。
在原点回归完成以后，才可以再次驱动自动运行。

→ 关于该指令的导入，请参考[14.1.2 导入IST指令的实例(工件传送设备举例)]。

注意要点

1. 关于 (S•) 软元件和使用开关

选择模式用的开关，不需要全部使用。
请将不使用的开关，设置为空号(不能用于其他用途)。

2. 关于IST指令和STL指令的编程顺序

- 需要在状态S0～S2等一连串的STL回路前先编写IST指令的程序。

3. 关于原点回归动作中使用的状态

原点回归动作用的状态请使用S10～S19。
原点回归动作的最终状态中，请在M8043置位后，执行自我复位。

4. 关于指令的使用次数的限制

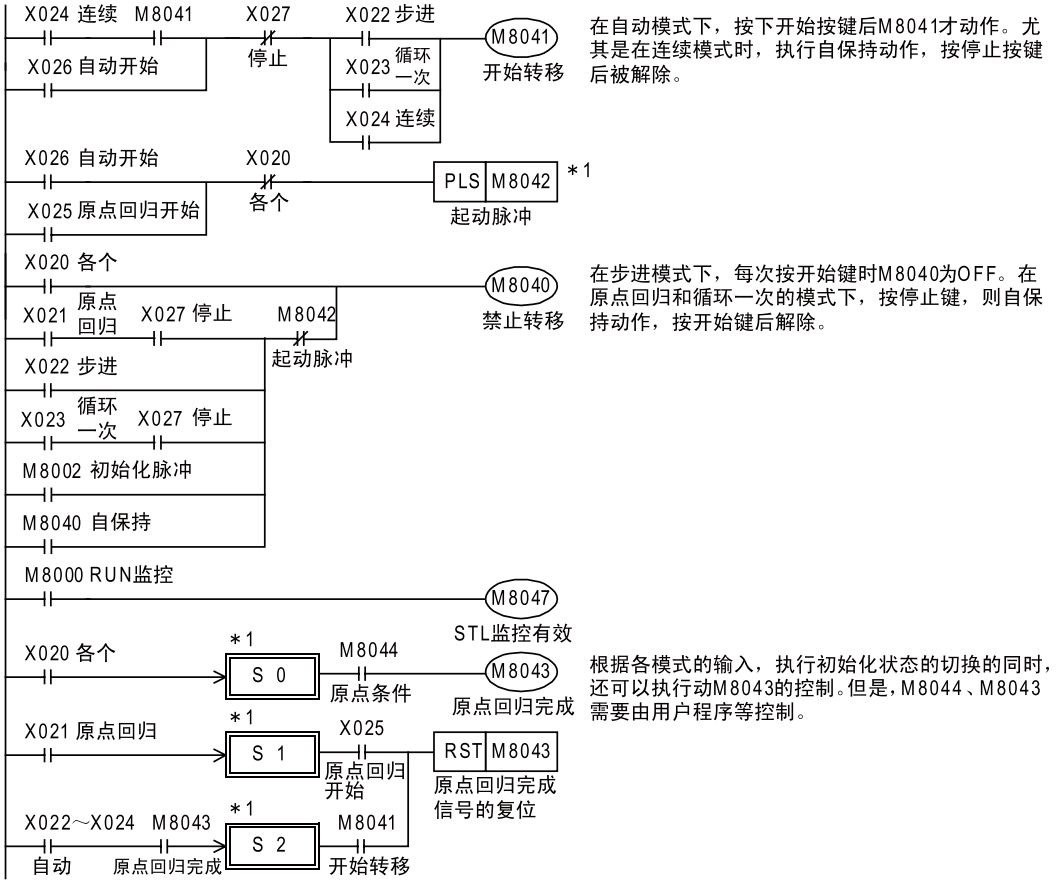
IST指令在程序中只能编写1个。

14.1.1 IST指令的等价梯形图

有关使用IST指令被自动控制的特殊辅助继电器(M)和初始化状态(S0～ S9)的详细内容，如下面的等价梯形图所示。(作为参考知识，请阅读一遍)

此外，该等价梯形图不能编写为程序。

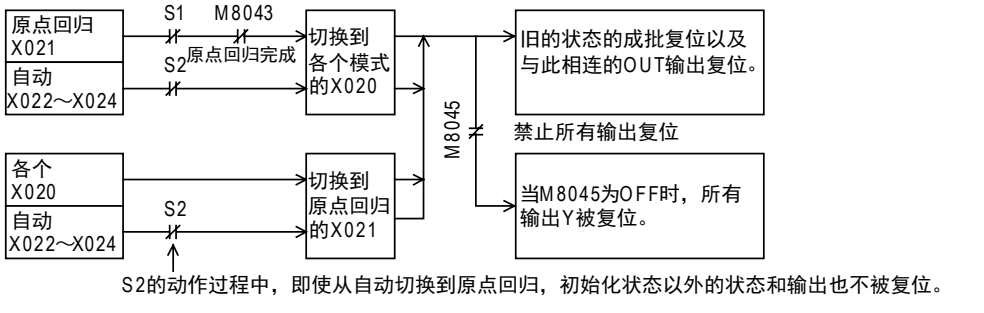
1. 等价梯形图



*1. 等价梯形图仅用于说明，实际上不能编写这样的程序。

2. 运行模式的切换

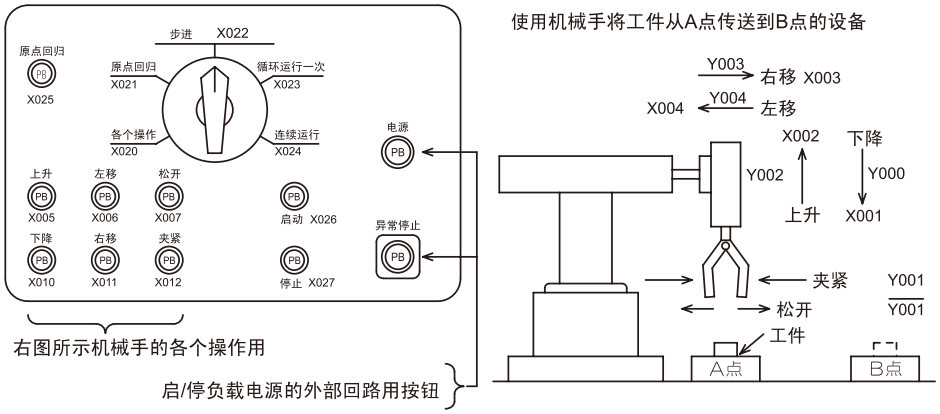
执行各个、原点回归、自动模式之间的切换后，如机械处于非原点位置时，对所有输出以及旧的状态进行成批复位。(驱动M8045时不对所有输出*2进行复位。)



*2. 所有输出: 未被状态 (S) 驱动的输出(Y)、以及被状态 (S) 通过OUT、SET指令驱动的输出(Y)

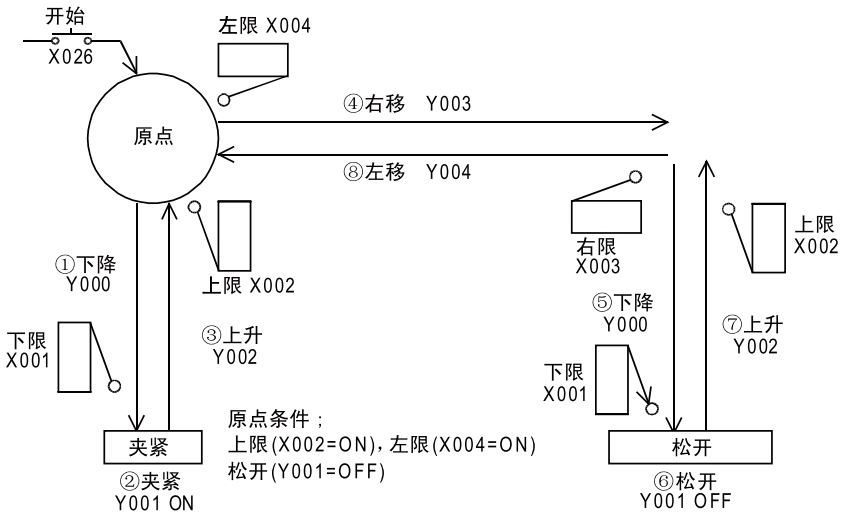
14.1.2 导入IST指令的实例(工件传送设备举例)

1. 运行模式



操作模式		动作内容
手动	各个操作:	使用各按钮对各负载启停操作的模式。
	原点回归:	按下原点回归用按键，使机械自动返回到原点的模式。
自动	步进:	每次按下开始按键就前进1个工序。
	循环运行一次:	在原点位置按下开始按键后，执行1个循环的自动运行后，在原点位置停止。 如果在中途按下停止按键，就在该工序停止，再次按开始按键后会从刚才的位置开始继续运行，然后在原点处自动停止。
	连续运行:	在原点的位置按下开始按键后，开始连续的重复运行。 按下停止按健后，会运行到原点位置后停下。

2. 传送机械



以左上为原点，按照下降，夹紧，上升，右移，下降，松开，上升，左移的顺序将工件从左移到右。
下降/上升、左移/右移时使用双电磁阀(驱动/非驱动的2个输入)，夹紧时使用单电磁阀(仅通电时动作)。

3. 模式选择输入的分配

为了使用IST指令，需要按照下面所示模式输入，分配连续编号的输入。

非连续编号时或者省略了一部分的模式时，如下图所示使用辅助继电器更改排列，将其作为指定模式的起始输入使用。

－X020: 各个操作

－X021: 原点回归

－X022: 步进

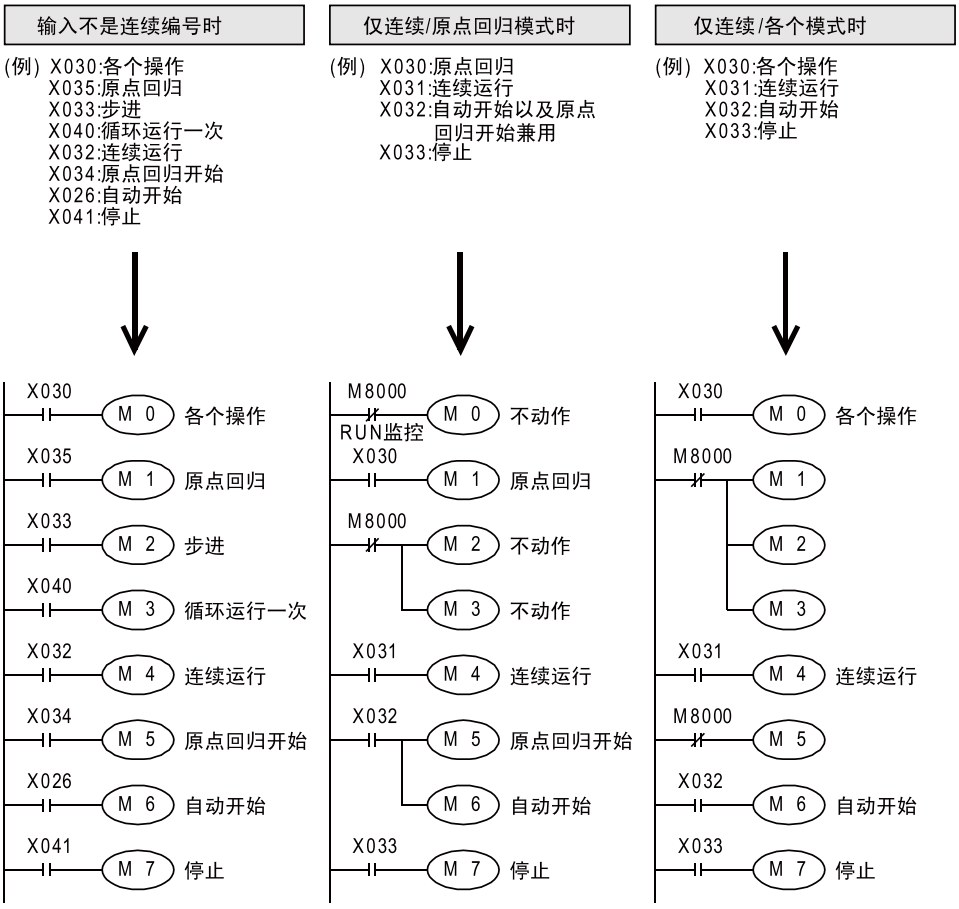
－X023: 循环运行一次

－X024: 连续运行

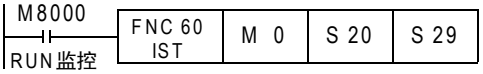
－X025: 原点回归开始

－X026: 自动开始

－X027: 停止



在这个例子中，将M0作为模式指定起始输入使用



4. IST指令用的特殊辅助继电器(M)

在IST指令中使用的辅助继电器(M)，区分为: 根据指令本身具体情况自动进行控制，以及根据运行的准备和控制目的，需要用程序进行控制。

1) IST指令自动进行控制

a) M8040: 禁止转移

一旦该辅助继电器工作，则所有的状态转移被禁止。

各个 : M8040一直动作。

原点回归、循环运行一次 : 按下停止键后，到按开始键为止，动作被保持。

步进 : M8040一直动作。但是仅当按下开始键时不动作，并执行转移。

其他 : 可编程控制器从STOP切换到RUN时动作也保持，按下开始键后被解除。
即使在禁止转移状态时，状态内的输出仍继续原来的动作。

b) M8041: 转移开始

作为从初始化状态S2开始转移到下一个状态的转移条件的辅助继电器。

各个、原点回归 : 不动作。

步进、循环运行一次 : 仅当按下开始键时动作。

连续 : 按下开始键后动作被保持，按下停止键后被解除。

c) M8042: 启动脉冲

仅在按下开始键时的一瞬间动作。

d) M8047: STL监控有效

使用IST指令后，M8047置ON。

通过M8047置ON，使STL监控变为有效，动作中的状态编号(S0～S899)按照从小到大的顺序被保存在特殊辅助继电器D8040～D8047中。

因此，最多可以监控8个动作状态编号。

此外，如果这些状态中任意一个动作，特殊辅助继电器M8046也动作。

2) 用顺控程序驱动

→ 有关这些控制的详细内容，请参考下一页

a) M8043: 原点回归结束

在原点回归模式下，当机械回归到原点时，请使用用户程序使这个特殊辅助继电器(M)动作。

b) M8044: 原点条件

请在检测出机械的原点条件后，驱动这个特殊辅助继电器。所有模式下都为有效信号。

c) M8045: 禁止所有输出复位

在各个、原点回归、自动模式之间执行切换后，当机械不在原点位置时，所有输出和动作状态被复位。
但是如先驱动M8045，则仅动作状态被复位。

5. 程序举例

1) 梯形图

下面的顺控回路中，阴影部分以外的为常规回路。

请务必将结合控制的内容编写阴影部分的梯形图。

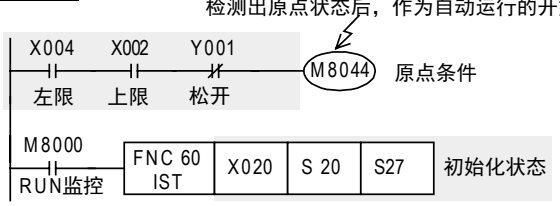
a) 初始回路

机械运行过程中，可以随意在“自动运行”模式内进行运行的切换。(步进/循环一次/连续)

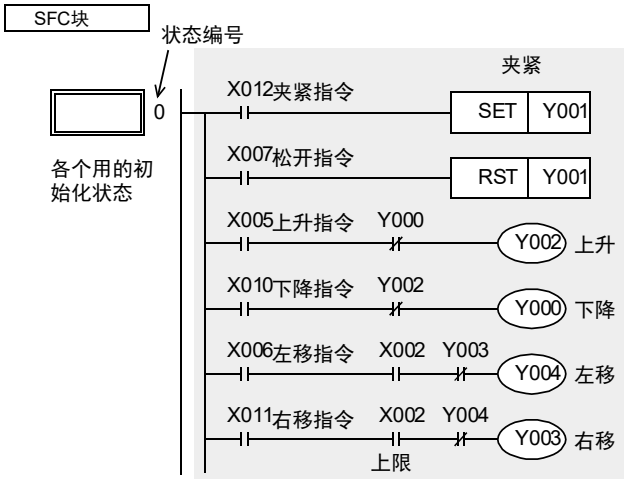
机械运行过程中，在“各个操作”/“原点回归”/“自动运行”间执行切换时，为了安全起见，在对所有输出复位一次后，切换后的模式设置才变为有效。

(禁止所有输出复位M8045为ON时，不能被复位)

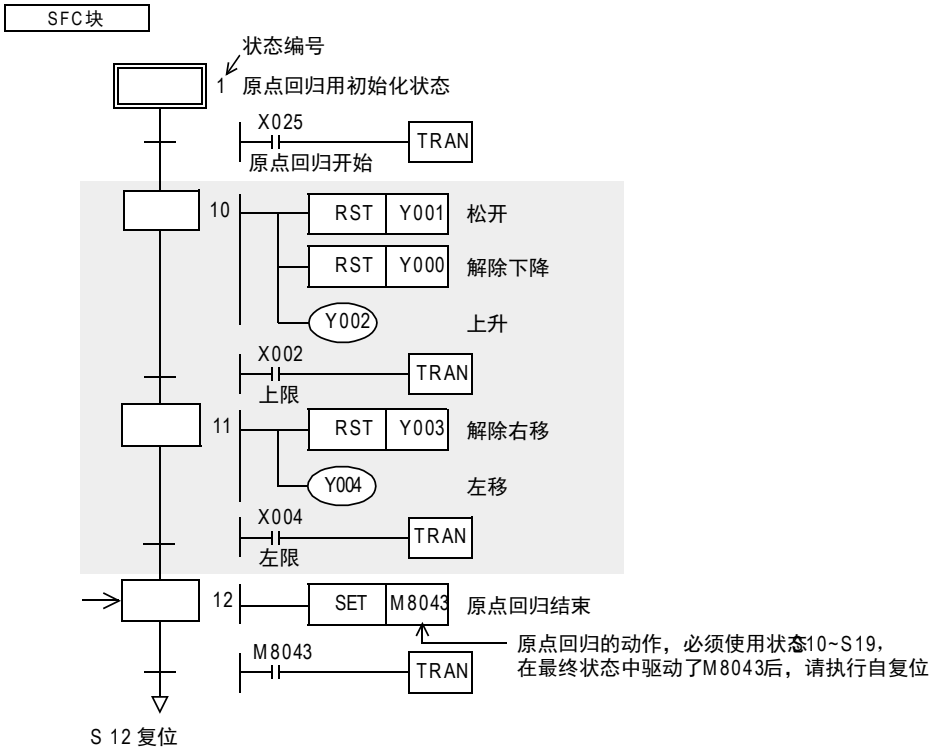
梯形图块



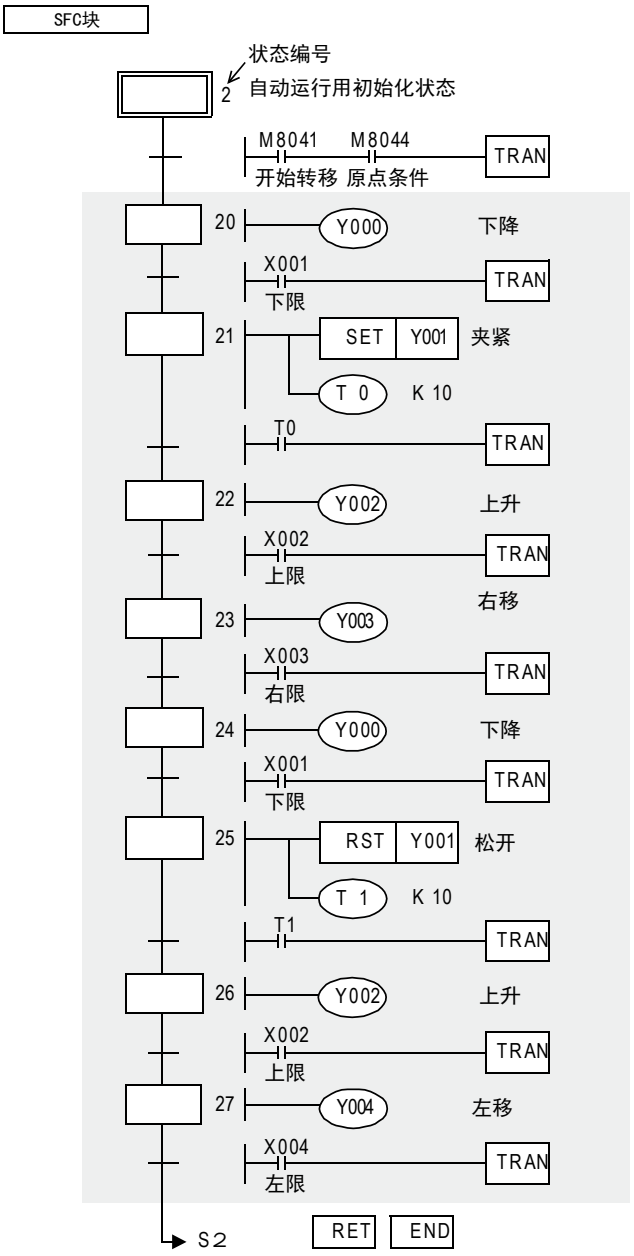
b) 各个操作
无各个模式时不需要编程。



c) 原点回归
无原点回归模式时不需要编程。但是在自动运行之前，需要先将原点回归结束M8043置位一次。



d) 自动运行(步进/循环一次/连续)



6. 指令表程序

上一页的梯形图中的指令表程序如下所示。

初始回路

```
0  LD   X 004
1  AND  X 002
2  ANI  Y 001
3  OUT  M8044
5  LD   M8000
6  IST      60
   (SP) X 020
   (SP) S  20
   (SP) S  27
13 STL  S   0
```

各个操作

```
14 LD   X 012
15 SET  Y 001
16 LD   X 007
17 RST  Y 001
18 LD   X 005
19 ANI  Y 000
20 OUT  Y 002
21 LD   X 010
22 ANI  Y 002
23 OUT  Y 000
24 LD   X 006
25 AND  X 002
26 ANI  Y 003
27 OUT  Y 004
28 LD   X 011
29 AND  X 002
30 ANI  Y 004
31 OUT  Y 003
   (RET)
```

不需要编程。

原点回归

```
32 STL  S   1
33 LD   X 025
34 SET  S  10
36 STL  S  10
37 RST  Y 001
38 RST  Y 000
39 OUT  Y 002
40 LD   X 002
41 SET  S  11
43 STL  S  11
44 RST  Y 003
45 OUT  Y 004
46 LD   X 004
47 SET  S  12
49 STL  S  12
50 SET  M8043
52 LD   M8043
53 RST  S  12
   (RET)
55 STL  S   2
56 LD   M8041
57 AND  M8044
58 SET  S  20
60 STL  S  20
61 OUT  Y 000
62 LD   X 001
63 SET  S  21
65 STL  S  21
66 SET  Y 001
67 OUT  T   0
   (SP) K  10
```

自动运行

```
70  LD   T   0
71  SET  S  22
73  STL  S  22
74  OUT  Y 002
75  LD   X 002
76  SET  S  23
78  STL  S  23
79  OUT  Y 003
80  LD   X 003
81  SET  S  24
83  STL  S  24
84  OUT  Y 000
85  LD   X 001
86  SET  S  25
88  STL  S  25
89  RST  Y 001
90  OUT  T   1
    (SP) K  10
93  LD   T   1
94  SET  S  26
96  STL  S  26
97  OUT  Y 002
98  LD   X 002
99  SET  S  27
101 STL  S  27
102 OUT  Y 004
103 LD   X004
104 OUT  S   2
106 RET
107 END
```

14.2 FNC 61 - SER/数据检索

A8 A8C

概要

从数据表中检索相同数据、以及最大值、最小值的指令。

1. 指令格式

	FNC 61	
D	SER	P
DATA SEARCH		

16位指令	指令记号	执行条件
9步	SER	连续执行型
	SERP	脉冲执行型

32位指令	指令记号	执行条件
17步	DSER	连续执行型
	DSERP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	检索相同数据、最大值、最小值的起始软元件编号	BIN16/32位
(S2·)	检索相同数据、最大值、最小值的参考值或是其保存目标软元件编号	BIN16/32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	BIN16/32位
n	检索相同数据、最大值、最小值的个数 [16位指令时:1～256，32位指令时:1～128]	BIN16/32位

3. 对象软元件

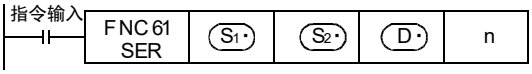
操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	▲			●					
(S2·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲			●					
n														●	●					●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

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) 被检索的数据的内容及结果

a) 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。

b) 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。

2) 动作例子

a) 检索结果表格的结构及数据示例

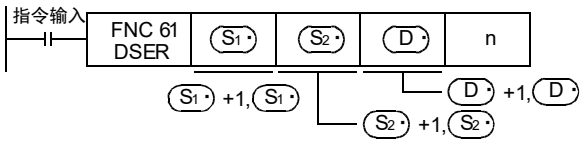
被检索软元件 (S1・)	被检索数据 (S1・) 的值(例)	比较数据 (S2・) 的值(例)	数据的位置	检索结果		
				最大值 (D・) +4	一致 (D・)	最小值 (D・) +3
(S1・)	K100	K100	0		○(初次)	
(S1・) +1	K111		1			
(S1・) +2	K100		2		○	
(S1・) +3	K98		3			
(S1・) +4	K123		4			
(S1・) +5	K66		5			○
(S1・) +6	K100		6		○(最终)	
(S1・) +7	K95		7			
(S1・) +8	K210		8	○		
(S1・) +9	K88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

b) 检索结果表格

软元件编号	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]中。



1) 被检索的数据的内容及结果

a) 存在相同数据时

在以[(D・) +1，(D・)]开始的5个32位数据的中，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。

b) 不存在相同数据时

[在以(D・) +1，(D・)]开始的5个32位数据的中，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・) +1，(D・)]开头的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。

2) 动作例子

a) 检索结果表格的结构及数据示例

被检索软元件 (S1•)+1	被检索数据 (S1•)的值(例)	比较数据 (S2•)	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]	K 95000		7			
[(S1•)+17, (S1•)+16]	K910000		8	○		
[(S1•)+19, (S1•)+18]	K910000		9	○		

b) 检索结果表格

软元件编号	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

- 大小的比较
  以代数方式执行。(-10<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，分别保存最后一个的位置。
- 软元件的占用点数
  驱动本指令后，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。

2) 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。

- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DSER D0 D100 D200 R0时，则n=[R1, R0]。

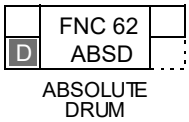
14.3 FNC 62 - ABSD/凸轮控制绝对方式

A1 A2 A2C A8 A8C

概要

对应计数器的当前值，产生多个输出模式的指令。

1. 指令格式



16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
9步	ABSD	连续执行型	17步	DABSD	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存数据表格(上升沿、下降沿)的起始软元件编号	BIN16/32位
(S2・)	与数据表格比较的当前值监控用计数器编号	BIN16/32位
(D・)	输出的起始位软元件编号位	
n	表格的行数以及输出的位软元件的点数[1≤n≤64]	BIN16位

3. 对象软元件

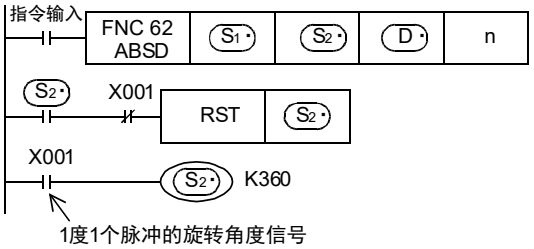
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲2			●					
(S2・)													●						●					
(D・)		●	●			●	▲1												●					
n																				●	●			

▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

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)与计数器的当前值 (S2・) 做比较，在旋转1次期间，对从 (D・) 开始的连续n点输出进行ON/OFF控制。

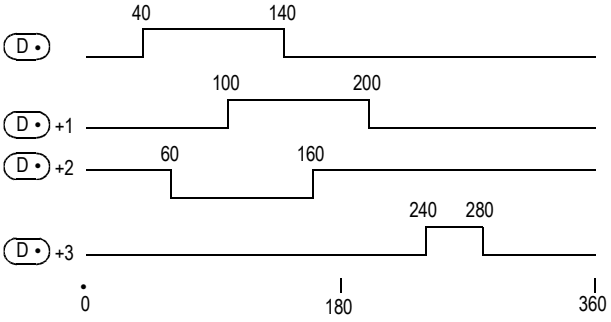


1) 先使用传送指令，在 (S1·) ～ (S1·) +2n+1中写入如下所示的数据。

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

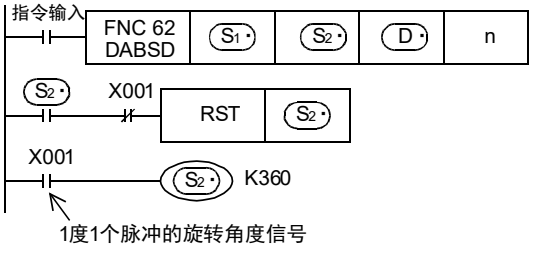
例如，上升点数据在偶数编号的软元件，下降点数据在奇数编号的软元件中，以16位数据进行保存。

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)，与计数器的当前值 (S2·) 做比较，在旋转1次期间，对从 (D·) 开始的连续n点输出进行ON/OFF控制。

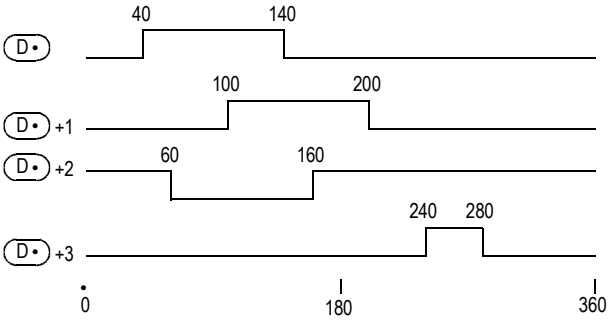


1) 先使用传送指令，在[(S1・)，(S1・)+1]～[(S1・)+4n+2，(S1・)+4n+3]中写入如下所示的数据。

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。

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。



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(S2・)。
但是此时，对于计数器的当前值，在输出模式中会由于运算周期而造成响应延迟。
A8/A8C使用可编程控制器时，使用HSZ指令进行表格的高速比较功能，或使用HSCT指令做出响应性。

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

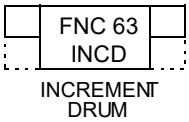
14.4 FNC 63 - INCD/凸轮控制相对方式

A1 A2 A2C A8 A8C

概要

使用一对计数器，产生多个输出模式的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	INCD	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1•)	保存设定值的起始字软元件编号	BIN16位
(S2•)	监控当前值用的计数器的起始编号	BIN16位
(D•)	输出的起始位软元件编号	位
n	输出的位软元件的点数[1≤n≤64]	BIN16位

3. 对象软元件

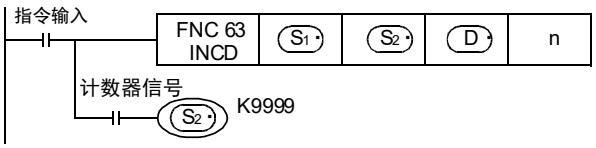
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲2			●					
(S2•)													●						●					
(D•)		●	●			●	▲1												●					
n																				●	●			

▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)
▲2: 仅对应A8/A8C可编程控制器。

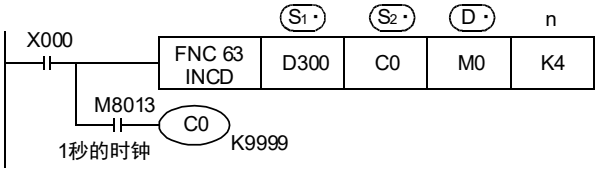
功能和动作说明

1. 16 位运算(INCD)

对 (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

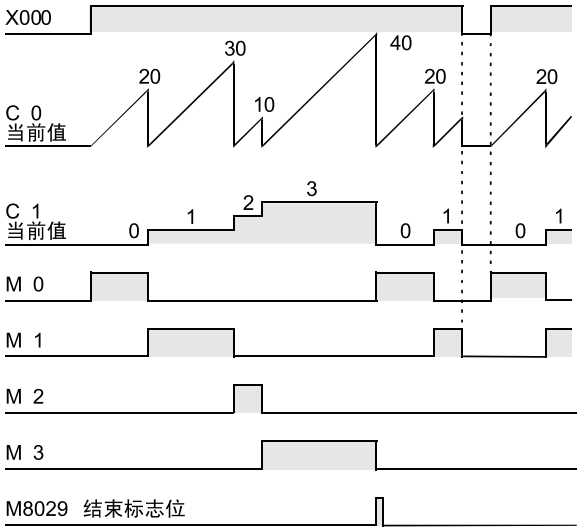


动作



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
(S1・)	D300=20	(D・)	M0
(S1・)+1	D301=30	(D・)+1	M1
(S1・)+2	D302=10	(D・)+2	M2
(S1・)+3	D303=40	(D・)+3	M3
⋮	⋮	⋮	⋮
(S1・)+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，工序计数器C1的计数值被+1，计数器C0的当前值也被复位。
6) 同样地方式一直要比较到n(K4)指定的点数为止。(1≤n≤64)
7) 在n指定的最后工序结束后，执行结束标志位M8029保持1个运算周期为ON。
由于M8029是多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位。
8) 回到最初，重复输出。

注意要点

1. 在(S1・)中指定位软元件的位数时
软元件编号中，请指定16的倍数(0、16、32、64…)。

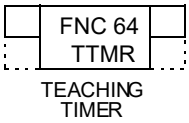
14.5 FNC 64 - TTMR/示教定时器

A8 A8C

概要

测量TTMR指令为ON的时间用的指令。
采用按钮来调节定时器的设定时间的情况下，可使用。

1. 指令格式



16位指令	指令记号	执行条件
5步	TTMR	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(D·)	保存示教数据的软元件编号	BIN16位
n	示教数据乘以的倍率数[K0～K2/H0～H2]	BIN16位

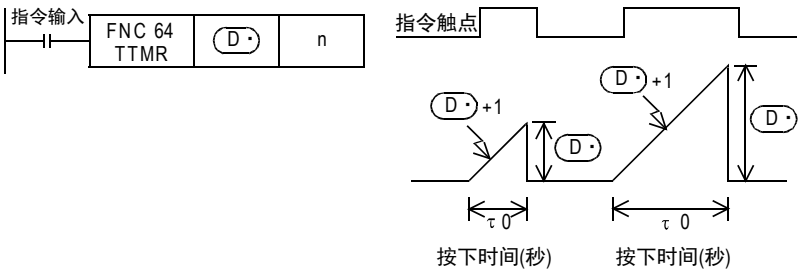
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)														●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(TTMR)

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^n)后传送到(D·)中。



被传倒(D·)中的时间是这样的，按下时间为τ0(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

相关指令

还有以下的方便指令。

指令	内容
HOUR(FNC 169)	以1小时为单位累计计算HOUR指令为ON的时间，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

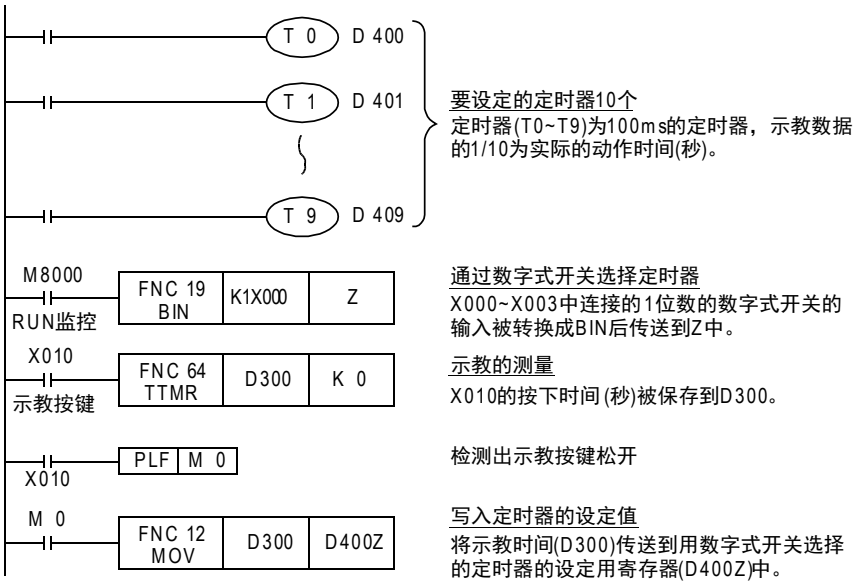
   以示教时间(D・)中指定的软元件为起始，占用2点。
   请注意不要与机器其他控制中使用的软元件重复。

   - (D・)：示教时间
   - (D・)+1：按下时间的当前值

程序举例

1. 在10种数据寄存器中，写入示教时间

   在D400～D409中先写入设定值。



要设定的定时器10个
定时器(T0~T9)为100ms的定时器，示教数据的1/10为实际的动作时间(秒)。

通过数字式开关选择定时器
X000~X003中连接的1位数的数字式开关的输入被转换成BIN后传送到Z中。

示教的测量
X010的按下时间(秒)被保存到D300。

检测出示教按键松开

写入定时器的设定值
将示教时间(D300)传送到用数字式开关选择的定时器的设定用寄存器(D400Z)中。

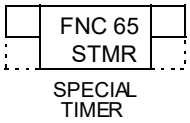
14.6 FNC 65 - STMR/特殊定时器

A8 A8C

概要

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。

1. 指令格式



16位指令	指令记号	执行条件
7步	STMR	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	使用的定时器编号[T0～T199(100ms定时器)]	BIN16位
m	定时器的设定值[1～32, 767]	BIN16位
(D·)	被输出的起始位编号(占用4点)	位

3. 对象软元件

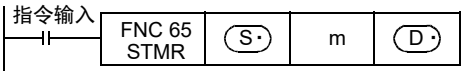
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)												●							●					
m														●	●					●	●			
(D·)		●	●			●	▲												●					

▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

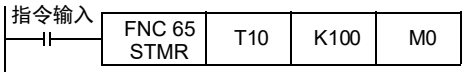
1. 16位运算(STMR)

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。
请根据使用用途，参考以下内容编写程序。

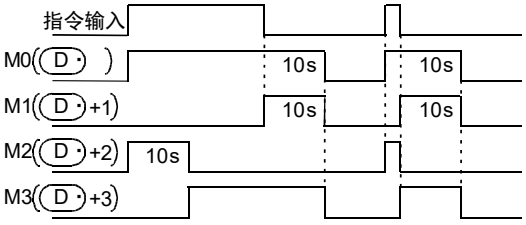


断开延时定时器・单脉冲定时器

在(S·)、m、(D·)中分别分配T10、K100、M0时

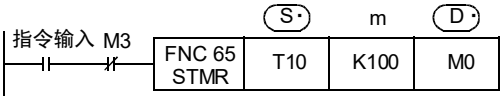


- M0[(D·)] ：指令触点OFF后，经过定时器的设定时间之后，再OFF的断开延时定时器。
- M1[(D·)+1]：指令触点从ON变为OFF后为ON，经过定时器设定时间后变为OFF的单脉冲定时器。
- M2[(D·)+2]：占用。用于闪烁。
- M3[(D·)+3]：占用。

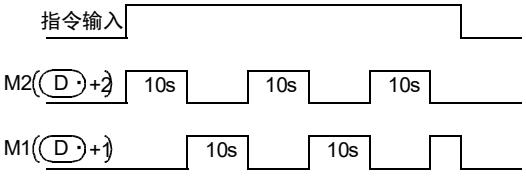


闪烁

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。



- M0[(D・)] ：占用。(用于断开延时定时器。请参考上一页)
- M1[(D・)+1]：以定时器的时间间隔，重复执行ON/OFF的闪烁(a触点)。
- M2[(D・)+2]：以定时器的时间间隔，重复执行ON/OFF的闪烁(b触点)。
- M3[(D・)+3]：占用。



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. 软元件的占用点数
   以 (D・) 中指定的软元件为起始点占用
   请注意不要与机器其他控制中使用的软元件重复。

软元件	功能	
	断开延时定时器 单脉冲定时器	闪烁
(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(b触点)

3. 当指令触点为OFF时
   (D・)、(D・)+1、(D・)+3在经过设定时间后变为OFF。(D・)+2和定时器(S・)被即时复位。

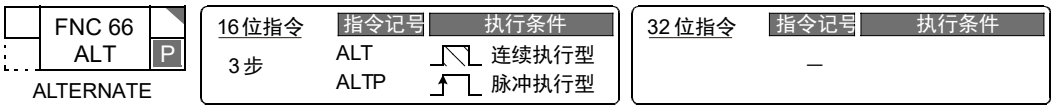
14.7 FNC 66 - ALT/交替输出

A1 A2 A2C A8 A8C

概要

输入为ON时，使位软元件反转(ON↻OFF)用的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D·)	交替输出的位软元件编号	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)		●	●			●	▲												●					

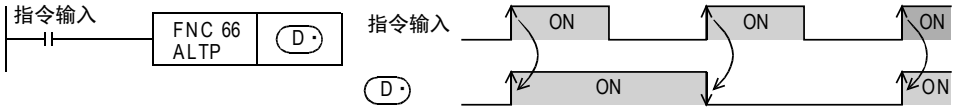
▲: D□.b仅对应A8A/8C可编程控制器。但是，不能变址修饰(V、Z)

功能和动作说明

1. 16位运算(ALT、ALTP)

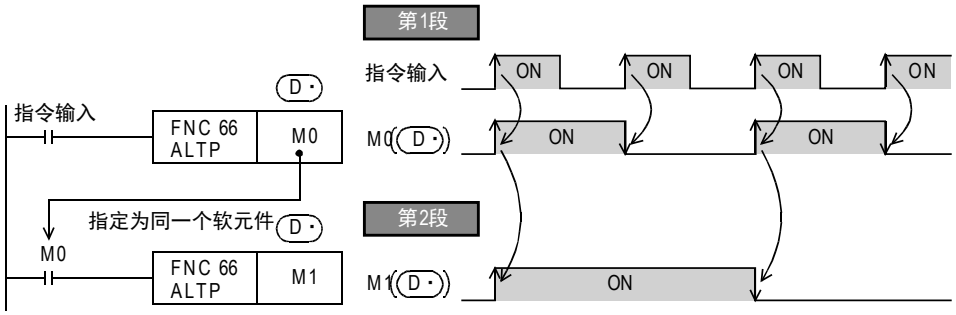
交替输出(1段)

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

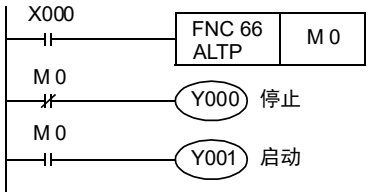
1. 使用ALT指令(连续执行型)时交替输出(1段)
   - 使用ALT指令编程时，每个运算周期都执行反转动作。
     希望通过指令的ON/OFF使其反转动作时，请使用ALTP指令(脉冲执行型)、或是LDP指令触点等(脉冲执行型)。

程序举例

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。

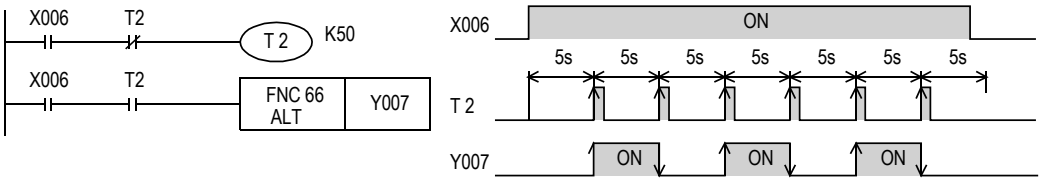
2) 再次按下按键X000后，停止输出Y000的动作。



2. 闪烁动作

1) 输入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。

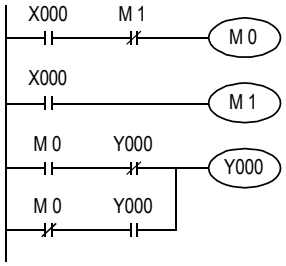


3. 使用了辅助继电器(M)的交替输出动作(和ALT指令相同的动作)

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

1) X000为ON时，M0仅维持一次运算周期为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。



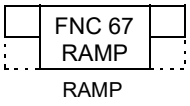
14.8 FNC 67 - RAMP/斜坡信号

A1 A2 A2C A8 A8C

概要

在开始(初始值)和结束(目标值)的 2 个值之间，按照指定的n次得到变化的数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	RAMP	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1·)	保存设定的斜坡初始值的软元件编号	BIN16位
(S2·)	保存设定的斜坡目标值的软元件编号	BIN16位
(D·)	保存斜坡的当前值数据的软元件编号	BIN16位
n	斜坡的转移时间(扫描周期)[1～32, 767]	BIN16位

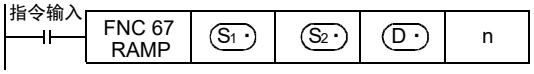
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●				●					
(S2·)														●	●				●					
(D·)														●	●				●					
n														●	●					●	●			

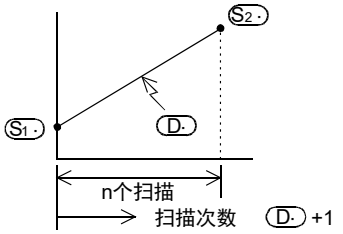
功能和动作说明

1. 16位运算(RAMP)

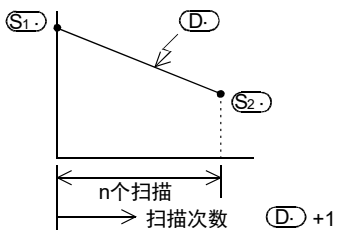
先指定开始的值(S1·)和要结束的值(S2·)，当指令输入为ON后，在每个运算周期中，将按照n中指定的次数进行等分的值依次加到(S1·)中，然后将加得的值保存到(D·)中。
将该指令和模拟量输出相结合，可以输出缓冲启动/停止指令。



1) (S1·) < (S2·) 时

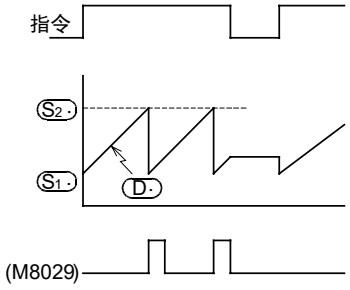


2) (S1·) > (S2·) 时



- (D·)+1中保存扫描次数(0→n次)。
- 从开始到结束所需时间，为运算周期×n个扫描。
- 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·)当前值数据，保持；(D·)+1扫描次数被清除)，再次置ON后，(D·)被清除，从(S1·)开始重新动作。

- 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。

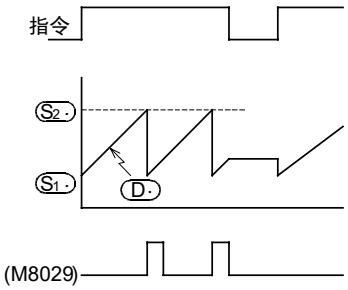


- 固定时间间隔下求出运算结果时(恒定扫描模式)
  将既定的扫描时间(比实际的扫描时间稍微长一点的值。)写入 D8039，当M8039为ON时，可编程控制器就处于恒定扫描模式。
  例如，这个值指定了20ms，n=100次的时候，就表示在20秒内(D・)的值从(S1・)变化到(S2・)。

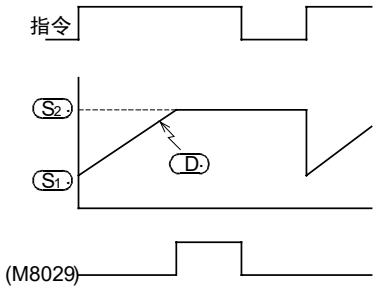
2. 模式标志位(M8026)的动作

A8/A8C可编程控制器中，根据模式标志位M8026的ON/OFF状态，(D・)+1的内容也作如下变更。

1) M8026=OFF时



2) M8026=ON时



相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	n个运算周期后，(D・)=(S2・)时置ON。
M8026*1	RAMP模式	请参考上面的模式标志位(M8026)的动作。

*1. 从RUN→STOP时清除。仅对应A8/A8C可编程控制器。

注意要点

1. (D・)中指定停电保持软元件(保持区域)时

指令输入按原样置ON，可编程控制器设置到RUN(开始)时，请先清除(D・)。

14.9 FNC 68 - ROTC/旋转工作台控制

A8 A8C

概要

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。

1. 指令格式

FNC 68 ROTC
ROTARY TABLE CONTROL

16位指令	指令记号	执行条件
9步	ROTC	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	计数用的数据寄存器	BIN16位
m1	分割数	BIN16位
m2	低速区间数	BIN16位
(D·)	驱动的起始软元件编号	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●				●					
m1																				●	●			
m2																				●	●			
(D·)		●	●			●	▲												●					

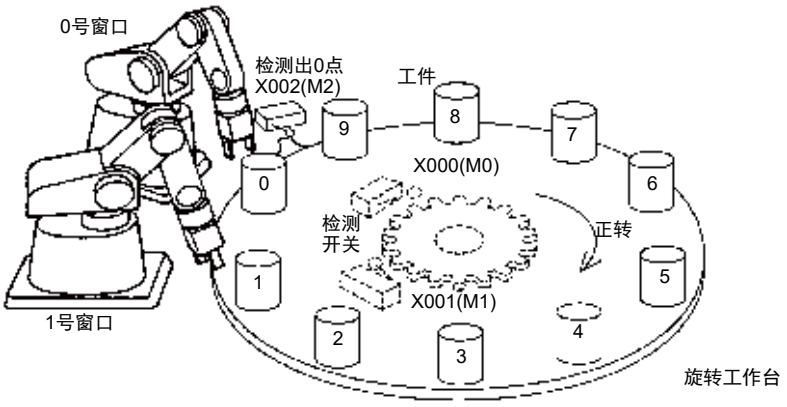
▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

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，针对要求取放的窗口，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・)

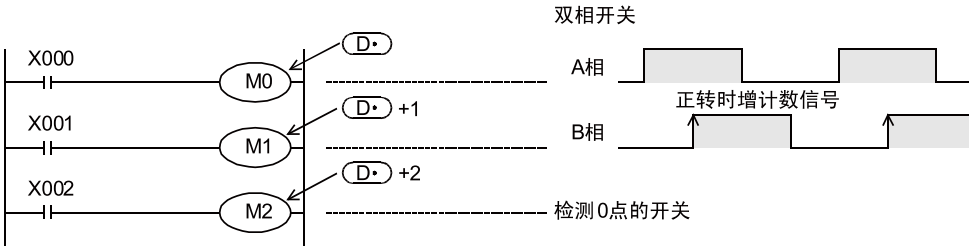
(S・)	计数器用寄存器	使用传送指令事先进行设定。
(S・)+1	调用窗口编号的设定	
(S・)+2	调用工件编号的设定	

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/反转用的双相开关(X000、X001)，以及0号工件到0号窗口时会动作的开关X002。
   －请编制如下例所示的顺控程序



用X000～X002替换 (D・) ～ (D・)+2 的内部触点。
X和 (D・) 指定的起始软元件编号为任意的。

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
   a) 在 (S・)+1中设定要调用的窗口编号。
   b) 在 (S・)+2中设定要调用的工件编号。
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
   - 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。
   - 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)的工件1分割区域内的多次动作

   例如，旋转检测信号((D·) ～ (D·) +2)在工件1分割区域内动作10次时，请将分割数的设定，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·)

   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。

14.10 FNC 69 - SORT/数据排序

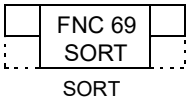
A8 A8C

概要

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/降序排列的SORT2(FNC 149)指令。

→ 有关SORT2(FNC 149)指令，请参考19.7节

1. 指令格式



16位指令	指令记号	执行条件
11步	SORT	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

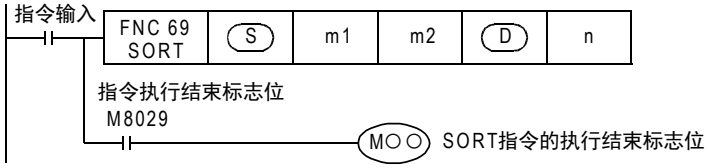
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S)														●	●									
m1																				●	●			
m2																				●	●			
(D)														●	●									
n														●	●					●	●			

功能和动作说明

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。

→ 有关动作实例，请参考下一页



- 下面例举排序前 m1=K3、m2=K4 的例子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)。

列号 / 行号		群数m2个(m2=K4时)			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数m1=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时开始数据排列，m1个扫描后数据排列结束，指令执行结束标志位M8029为ON。

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

2. 动作举例

在“n=K2(列号2)”和“n=K3(列号3)”中执行下述的排序前的数据后，会如下所示动作。

此外，如果先在第1列中输入管理编号等的连续编号，则可以根据其内容判断出原来所在的行号，因此非常方便。

排序前数据

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

1) 以n=K2(列号2)执行指令时的排序结果

列号 / 行号	1	2	3	4
	管理编号	身高	体重	年龄
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)执行指令时的排序结果

列号 / 行号	1	2	3	4
	管理编号	身高	体重	年龄
1	(D)	(D)+5	(D)+10	(D)+15
	4	100	20	8
2	(D)+1	(D)+6	(D)+11	(D)+16
	1	150	45	20
3	(D)+2	(D)+7	(D)+12	(D)+17
	2	180	50	40
4	(D)+3	(D)+8	(D)+13	(D)+18
	5	150	50	45
5	(D)+4	(D)+9	(D)+14	(D)+19
	3	160	70	30

相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

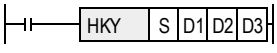
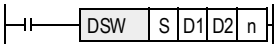
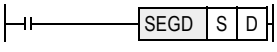
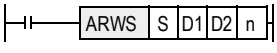
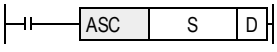
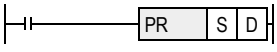
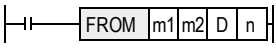
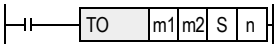
软元件	名称	内容
M8029	指令执行结束	数据排列结束时位ON。

注意要点

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入OFF一次。
- 指令的使用次数的限制
  程序中仅可以使用1次。
- (S)和(D)中指定了同一个软元件时
  原来的数据按照排序后的数据顺序被改写。
  到执行结束为止，请特别注意不要改变(S)的内容。

15. 外部设备I/O—FNC 70～FNC 79

FNC 70～FNC 79中，主要准备了使用可编程控制器的输入输出与外部设备之间进行数据交换的指令。
由于使用这些指令，可以通过最小的顺控程序和外部接线简便地实现复杂的控制，因此具有与前面所述的方便指令相似的特征。
而且，控制特殊单元和特殊模块所必需的FROM、TO指令也包含在其中。(A8/A8C可编程控制器中，也可以使用MOV指令进行传送。)

FNC No.	指令记号	符号	功能	参考
70	TKY		数字键输入	15.1节
71	HKY		16进制数字键输入	15.2节
72	DSW		数字开关	15.3节
73	SEGD		7段解码器	15.4节
74	SEGL		7SEG时分显示	15.5节
75	ARWS		箭头开关	15.6节
76	ASC		ASCII数据输入	15.7节
77	PR		ASCII码打印	15.8节
78	FROM		BFM的读出	15.9节
79	TO		BFM的写入	15.10节

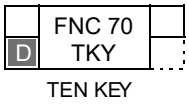
15.1 FNC 70 - TKY/数字键输入

A8 A8C

概要

通过0～9的键盘(数字键)输入，对定时器和计数器等设定数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	TKY	连续执行型

32位指令	指令记号	执行条件
13步	DTKY	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	输入数字键的起始位软元件[占用10点]	位
(D1・)	保存数据的字软元件编号	BIN16/32位
(D2・)	按键信息为ON的起始位软元件编号[占用11点]	位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)	●	●	●			●	▲												●					
(D1・)									●	●	●	●	●	●	●	●	●	●	●					
(D2・)		●	●			●	▲												●					

▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

1. 16位运算(TKY)

对连接的数字键输入[(S・)～(S・)+9]按动键盘，将输入的数值保存在(D1・)中，在(D2・)～(D2・)+10中输出键盘输入信息以及检测到的键盘输出。

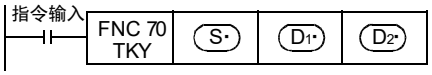
1) 有关输入的数值(D1・)

- 若为9,999以上时，从高位数溢出。
- 输入的数值，以BIN(2进制数)保存。
- 在下页的图中，按照①、②、③、④的顺序按数字键后，在(D1・)中保存为2130。

2) 有关按键信息的[(D2•)～(D2•)+10]

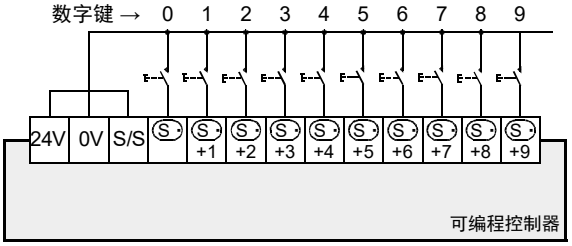
— (D2•)～(D2•)+9的按键信息，根据所按下的键ON/OFF。

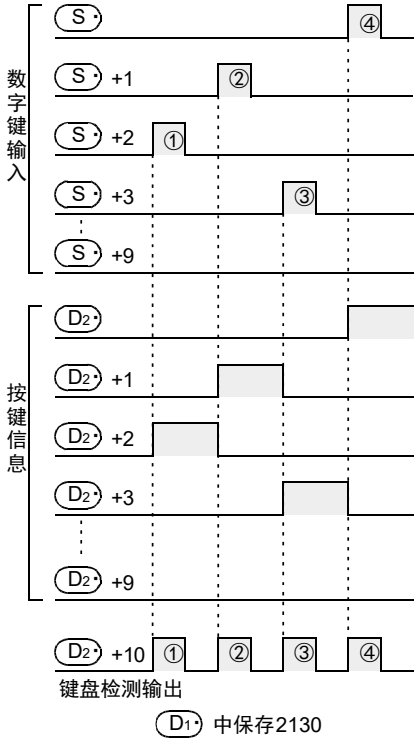
— 当0～9中任意一个键被按下时，(D2•)+10的键盘检测输出为ON。



下图中例举了A8可编程控制器(漏型输入)的例子。
有关接线，请参考所使用的可编程控制器下述的一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册





2. 32位运算(DTKY)

对连接的数字键输入[(S•)～(S•)+9]按动键盘，将输入的数值保存在[(D1•)+1、(D1•)]中，在(D2•)～(D2•)+10输出按键信息以及检测到的键盘输出。

1) 有关输入的数值(D1•)

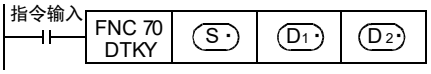
— 若为99,999,999以上时，从高位数溢出。

— 输入的数值，以BIN(2进制数)保存。

2) 有关按键信息的[(D2•)～(D2•)+10]

— (D2•)～(D2•)+9的按键信息，根据所按下的键ON/OFF。

— (D2•)+10的键盘检测输出为ON，当0～9中任意一个键被按下时。



有关数字键的连接实例以及按键信息，请参考上述的16位运算(TKY)的内容。

注意要点

1. 同时按下键盘时

同时按下多个键时，只有先按下的键有效。

2. 指令触点为OFF时

即使为OFF，(D1•)的内容也不改变，但是(D2•)～(D2•)+10为止都变为OFF。

3. 软元件的占用点数

1) 连接数字键的输入，占用从 (S・) 开始的10点。
即使不连接数字键(不使用)时，由于已被占用，所以不能用作其他用途。

2) 占用从按键信息输出用的起始软元件 (D2・) 开始的11点。
请注意不要与机器其他控制中使用的软元件重复。
－ (D2・) ～ (D2・) +9: 根据数字键0～9的输入置ON。
－ (D2・) +10: 按下0～9之间的任意一个键时为ON。(键盘检测输出)

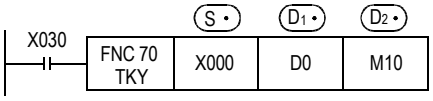
4. 指令的使用次数的限制

TKY指令或DTKY指令，在程序中只可以使用其中1个。
要多次使用时，请使用变址修饰(V、Z)功能进行编程。

程序举例

例举以输入X000作为开始，连接了0～9的数字键的例子进行说明。

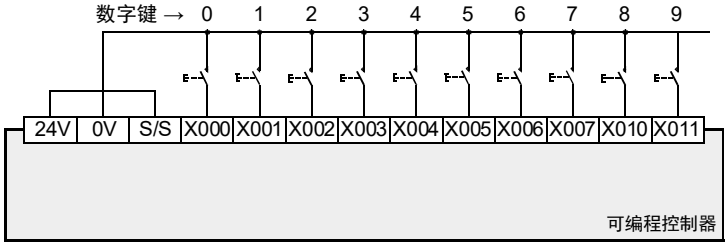
1. 程序



2. 接线图

这个接线图，是以A8可编程控制器(漏型输入)为例的。有关接线，请参考所使用的可编程控制器的下面一些手册。

→A8硬件篇手册
→A8C硬件篇手册

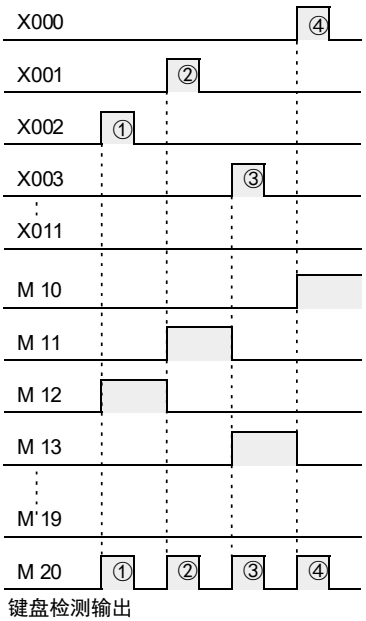


3. 时序图

1) 按照①、②、③、④的顺序按数字键后，D0的内容变为2130。
9,999以上的数值时，会从高位数依次溢出。
(实际D0的内容为BIN数据。)

2) 按下X002后，直到按动其他键盘为止，M12都置位(ON)。
其他的键盘也相同。
如此这般，根据输入X000～X011的动作，M10～M19动作。

3) 任何一个键被按动后，仅在按下的时间内键盘检测输出M20为ON。



15.2 FNC 71 - HKY/16进制数字键输入

A8 A8C

概要

通过0～F的键盘(16键)输入，设定数值(0～9)及运行条件(A～F功能键)等的输入数据用的指令。
当扩展功能为ON时，可以使用0～F键的16进制数进行键盘的输入。

1. 指令格式



16位指令	指令记号	执行条件
9步	HKY	连续执行型

32位指令	指令记号	执行条件
17步	DHKY	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	输入16键的起始位软元件(X)编号(占用4点)	位
(D1·)	输出的起始软元件(Y)编号(占用4点)	位
(D2·)	保存从16键输入的数值的软元件编号	BIN16/32位
(D3·)	按键信息为ON的起始位软元件编号(占用8点)	位

3. 对象软元件

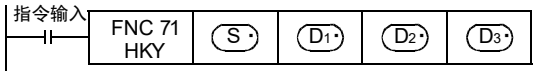
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●																		●					
(D1·)		●																	●					
(D2·)												●	●	●	●	●	●	●	●					
(D3·)		●	●			●	▲												●					

▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

1. 16位运算(HKY)

扫描连接16键(0～F)的输入[(S·)～(S·)+3]和列输出[(D1·)～(D1·)+3]的信号，按下0～9按键，其数值保存在(D2·)中，键盘检测输出到(D3·)+7中。
此外，按下A～F键后，与键盘对应的按键信息[(D3·)～(D3·)+5]为ON，键盘检测输出到(D3·)+6。



1) 关于使用0～9的键输入数值(D2·)、(D3·)+7
- 若为9,999以上时，从高位数溢出。
- 输入的数值，以BIN(2进制数)值保存在(D2·)中。
- 当0～9中任意一个键被按下时，键盘检测输出(D3·)+7为ON。

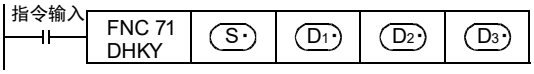
2) 关于A～F键的按键信息(D3·)～(D3·)+6
- 与A～F键相对应(D3·)的6点为ON。
- 当A～F中任意一个键被按下时，键盘检测输出(D3·)+6为ON。

键盘	按键信息
A	(D3·)
B	(D3·)+1
C	(D3·)+2

键盘	按键信息
D	(D3·)+3
E	(D3·)+4
F	(D3·)+5

2. 32位运算(DHKY)

扫描连接16键(0～F)的输入[(S・) ～ (S・)+3]和列输出[(D1・) ～ (D1・)+3]的信号，按下0～9按键，其数值保存在[(D2・)+1， (D2・)]中，键盘检测输出到 (D3・)+7中。
此外，按下A～F键后，与键盘对应的按键信息[(D3・) ～ (D3・)+5]为ON，键盘检测输出到 (D3・)+6。



1) 关于使用0～9的键输入数值[(D2・)+1、 (D2・)]、 (D3・)+7
- 若为99,999,999以上时，从高位数溢出。
- 输入的数值，以BIN(2进制数)值保存在[(D2・)+1、 (D2・)]中。
- 当0～9中任意一个键被按下时，键盘检测输出 (D3・)+7为ON。

2) 关于A～F键的按键信息 (D3・) ～ (D3・)+6
关于键盘按动信息，请参考上一页中16位运算(HKY)

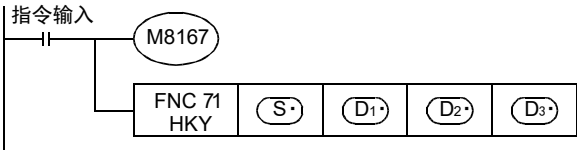
扩展功能

M8167为ON，扩展功能变为有效后，以BIN方式保存0～F的16进制按键的数据。
除下面以外，都与上述的[功能及动作说明]相同。

1. 16位运算(HKY)

使用0～F键输入的16进制数据原封不动地写入到 (D2・) 中。

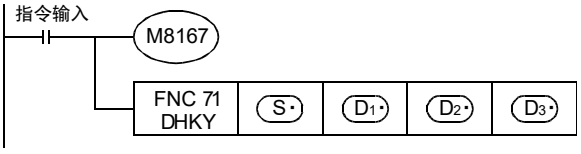
1) 关于使用0～F键的数值输入 (D2・)
- 当为FFFF以上时，从高位数溢出。
- 例如
  输入[1]→[2]→[3]→[B]→[F]时，以BIN方式将“23BF”保存到 (D2・) 中。
  当输入[F]时，[1]溢出。



2. 32位运算(DHKY)

使用0～F键输入的16 进制数据原封不动地写入到[(D2・)+1、 (D2・)]中。

1) 关于使用0～F键的数值输入[(D2・)+1、 (D2・)]
- 当为FFFFFFFF以上时，从高位数溢出。
- 例如
  输入[9]→[2]→[3]→[B]→[F]→[A]→[F]时，以BIN方式将“923BFAF”保存到[(D2・)+1、 (D2・)]中。



相关软元件

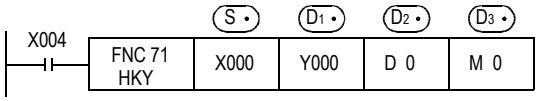
→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8167	功能扩展标志位	HKY(FNC 71)指令的HEX数据处理功能 OFF: 数字键+功能键 ON：16进制键
M8029	指令执行结束	OFF: (D1・) ～ (D1・)+3的扫描中，或指令未执行 ON：在循环输出 (D1・) ～ (D1・)+3一次的动作(0～F键盘扫描)后置ON

注意要点

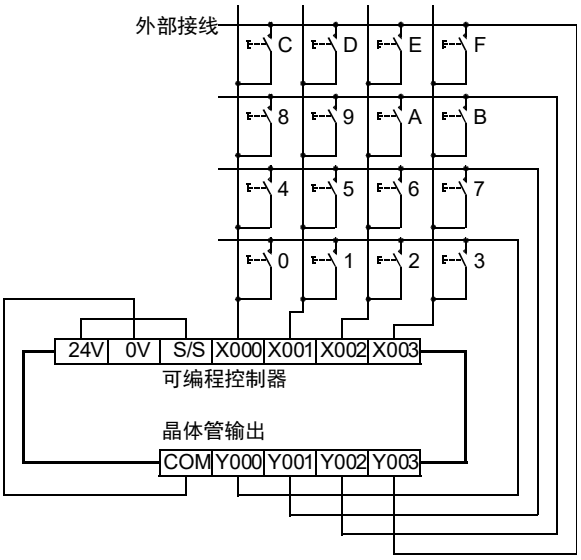
1. 指令的使用次数的限制
   HKY指令或DHKY指令，在程序中只可以使用其中1个。
   要使用多个时，请使用变址修饰(V、Z)功能编程。
2. 同时按下键盘时
   同时按下多个键时，先按下的键有效。
3. 指令触点为OFF时
   即使为OFF，(D2•)的内容也不改变，但是(D3•)～(D3•)+7为止都变为OFF。
4. 软元件的占用点数
   1) 连接16键时，占用从输入(X)的起始软元件(S•)开始的4点。
   2) 连接16键时，占用从输出(Y)的起始软元件(D1•)开始的4点。
   3) 占用从按键信息输出用的起始软元件(D3•)开始的8点。
      请注意不要与机器其他控制中使用的软元件重复。
      － (D3•)～(D3•)+5 : A～F键的按键信息
      － (D3•)+6 : A～F键的键盘检测输出
      － (D3•)+7 : 0～9键的键盘检测输出
5. 关于键盘输入的读取时序
   HKY、DHKY指令与可编程控制器的运算周期同步执行。
   完成一系列的键盘扫描，需要8个运算周期的时间。
   为了防止因键盘输入的滤波器延迟而导致的读取遗漏，请灵活使用[恒定扫描模式]和[定时器中断]功能。
6. 输出形式
   请使用晶体管输出型的可编程控制器。

程序举例



下图中的接线图，是以A8系列的基本单元(漏型输入/漏型输出)为例的。
有关接线，请参考所使用的可编程控制器下述的一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册



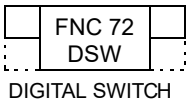
15.3 FNC 72 - DSW/数字开关

A1 A2 A2C A8 A8C

概要

读取数字开关设定值的指令。
可以读取1组4位数(n=K1)或是2组4位数(n=K2)的数据。

1. 指令格式



16位指令	指令记号	执行条件
9步	DSW	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	连接数字开关的起始软元件(X)编号(占用4点)	位
(D1・)	输出选通信号的起始软元件(Y)编号(占用4点)	位
(D2・)	保存数字开关的数值的软元件编号(占用n点)	BIN16位
n	数字开关的组数(4位数/1组)[n=1或是2]	BIN16位

3. 对象软元件

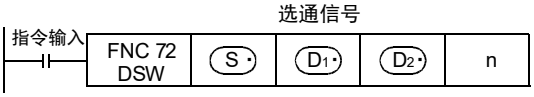
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)	●																		●					
(D1・)		●																	●					
(D2・)												●	●	●	●	▲	●	●	●					
n																				●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(DSW)

对(S・)中连接的数字开关的值执行时分处理(通过100ms间隔的输出信号，从第1位数开始依次输入)，并保存在(D2・)中。



1) 有关数据(D1・)
- 可以读取0～9,999的4位数。
- 以BIN(2进制数)值保存数据。
- 第1组保存到(D2・)中，第2组保存到(D2・)+1中。

2) 有关组数n的指定
- 使用4位数/1组×1时[n=K1]通过选通信号
  (S・)～[(S・)+3]，依次读取(D1・)～[(D1・)+3]中连接的BCD4位数的数字开关，并且将其值作为BIN值保存到(D2・)中。
- 使用4位数/1组×2时[n=K2]通过选通信号
  (S・)～[(S・)+3]，依次读取(D1・)～[(D1・)+3]中连接的BCD4位数的数字开关，并且将其值作为BIN值保存到(D2・)中。
  通过选通信号(D1・)～[(D1・)+3]，依次读取(S・)+4～[(S・)+7]中连接的BCD4位数的数字开关，并且将其值作为BIN值保存到(D2・)+1中。

相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	OFF：(D1•)～(D1•)+3的扫描中，或指令未执行 ON ：(D1•)～(D1•)+3循环输出一次的动作(1～4位数的扫描)后置ON

注意要点

1. 指令触点为OFF时

   即使为OFF，(D2•)的内容也不改变，但是(D1•)～(D1•)+3为止都变为OFF。

2. 软元件的占用点数

   1) 使用4位数2组(n=K2)时，占用从(D2•)开始的2点。

   2) 当为4位数1组时(S•)被占用4点，当为4位数2组时被占用8点。

3. 连接不满4位数的数字开关时

   对于没有使用的位数，选通信号<指定位数用的输出>(D1•)无需接线，但是即使有未使用的位数，其输出也已经被这个指令占用了，所以也不能用于其他用途。请务必将不使用的输出空出。

4. 推荐使用晶体管输出型

   为了能够连续地读取数字开关的值，请务必使用晶体管输出型的可编程控制器。

   → 有关继电器输出型，请参考后述的[继电器输出型中的使用方法]

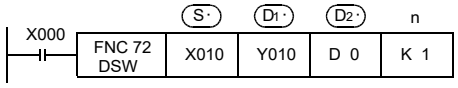
5. 有关数字开关

   请使用BCD输出型的数字开关。

程序举例

以输入X010开始，指定位数的输出以Y010开始，以它们连接的数字开关为例进行说明。

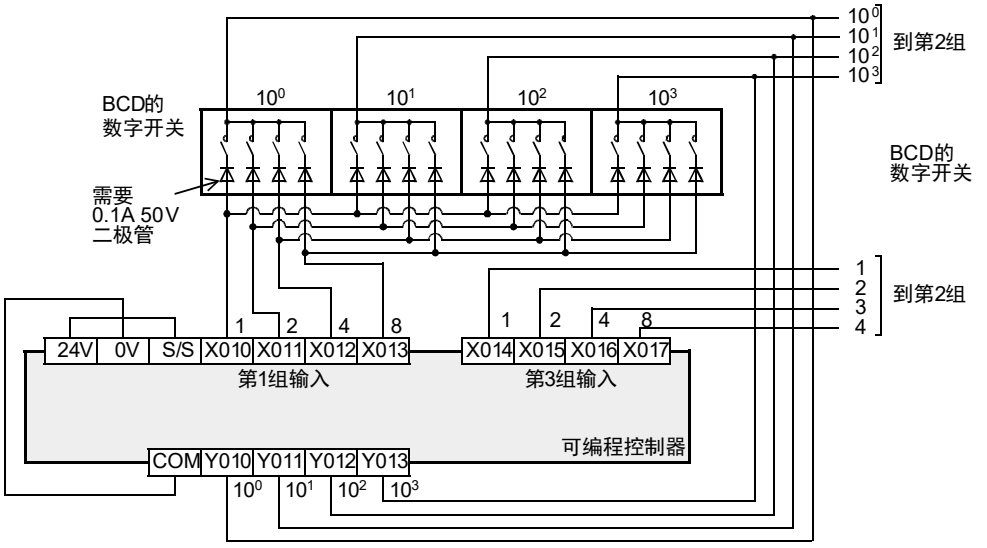
1. 程序



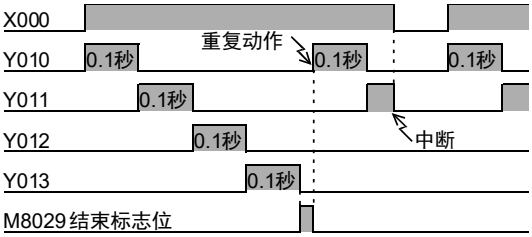
2. 接线图

   下图中的接线图，是以A8系列的基本单元(漏型输入/漏型输出)为例的。
   有关接线，请参考所使用的可编程控制器下述的一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册



3. 时序图

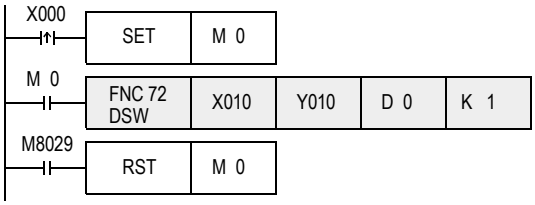


在X000为ON期间，Y010～Y013每隔100ms就依次置ON，循环动作1次后，执行结束标志位M8029动作。

4. 继电器输出型中的使用方法

通过设置[数字开关读取输入]，也可以使用继电器输出型的可编程控制器。

如果X000为按键输入，则仅当按键按下时DSW(FNC 72)才执行一系列的动作。因此，在使用该指令的情况下，Y010～Y013即使为继电器输出，也无需担心寿命的问题。



1) M0(数字开关读取输入)为ON期间，DSW(FNC 72)动作。

2) DSW(FNC 72)在其结束1个循环的动作，指令执行结束标志位(M8029)变为ON之前，一直动作。

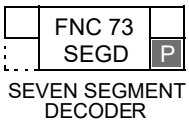
15.4 FNC 73 - SEGD/7段解码器

A8 A8C

概要

数码译码后，点亮 7 段数码管(1 位数)的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	SEGD	连续执行型
	SEGDP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	译码的起始字软元件	BIN16位
(D·)	保存7段码显示用数据的字软元件编号	BIN16位

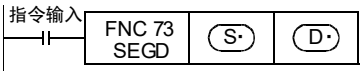
3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	●	●	●	●					

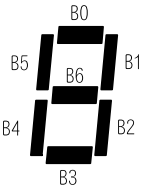
功能和动作说明

1. 16位运算(SEGD,SEGDP)

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



2. 7段码译码表

(S・)					7段码的构成	(D・)											显示数据
16进制数	b3	b2	b1	b0		B15	…	B8	B7	B6	B5	B4	B3	B2	B1	B0	
0	0	0	0	0		—		—	0	0	1	1	1	1	1	1	0
1	0	0	0	1		—		—	0	0	0	0	0	1	1	0	1
2	0	0	1	0		—		—	0	1	0	1	1	0	1	1	2
3	0	0	1	1		—		—	0	1	0	0	1	1	1	1	3
4	0	1	0	0		—		—	0	1	1	0	0	1	1	0	4
5	0	1	0	1		—		—	0	1	1	0	1	1	0	1	5
6	0	1	1	0		—		—	0	1	1	1	1	1	0	1	6
7	0	1	1	1		—		—	0	0	1	0	0	1	1	1	7
8	1	0	0	0		—		—	0	1	1	1	1	1	1	1	8
9	1	0	0	1		—		—	0	1	1	0	1	1	1	1	9
A	1	0	1	0		—		—	0	1	1	1	0	1	1	1	A
B	1	0	1	1		—		—	0	1	1	1	1	1	0	0	b
C	1	1	0	0		—		—	0	0	1	1	1	0	0	1	C
D	1	1	0	1		—		—	0	1	0	1	1	1	1	0	d
E	1	1	1	0		—		—	0	1	1	1	1	0	0	1	E
F	1	1	1	1		—		—	0	1	1	1	0	0	0	1	F

↑
位软元件的起始，或是字软元件的最低位为B0。

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。

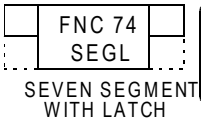
15.5 FNC 74 - SEGL/7SEG码时分显示

A1 A2 A2C A8 A8C

概要

控制1组或是2组4位数带锁存的7段数码管显示的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	SEGL	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	BCD转换的起始字软元件	BIN16位
(D·)	被输出的起始Y编号	位
n	参数编号[设定范围 :K0(H0)～K7(H7)]	BIN16位

3. 对象软元件

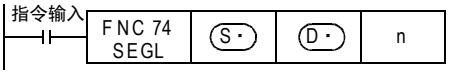
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)		●																	●					
n																				●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(SEGL)

(S·)的4位数值转换成BCD数据，采用时分方式，依次将每1位数输出到带BCD译码的7段数码管中。



使用4位数1组时(n=K0～K3)

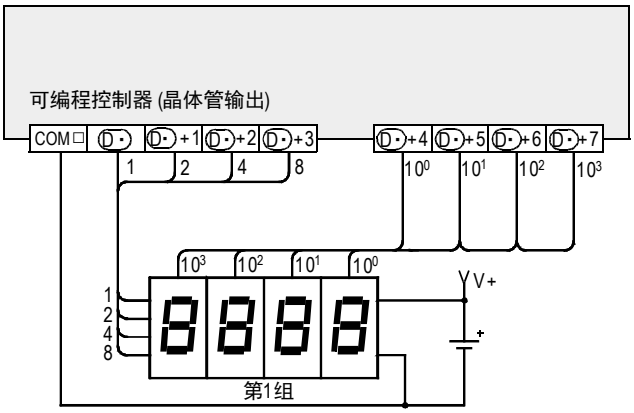
→ n的选择，请参考15.5.2项

1) 数据和选通信号
将(S·)的4位数值从BIN转换为BCD后，采用时分方式，从(D·)～[(D·)+3]依次对每一位数做输出。
此外，选通信号输出[(D·)+4]～[(D·)+7]也依次以时分方式输出，锁定为4位数第1组的7段码显示。

2) (S·)为0～9,999范围的BIN数据时有效。

3) 连接7段数码管的例子
下图，是以A8系列的基本单元(漏型输出)为例。
有关接线，请参考所使用的可编程控制器的下面一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册



使用4位数2组时(n=K4～K7)

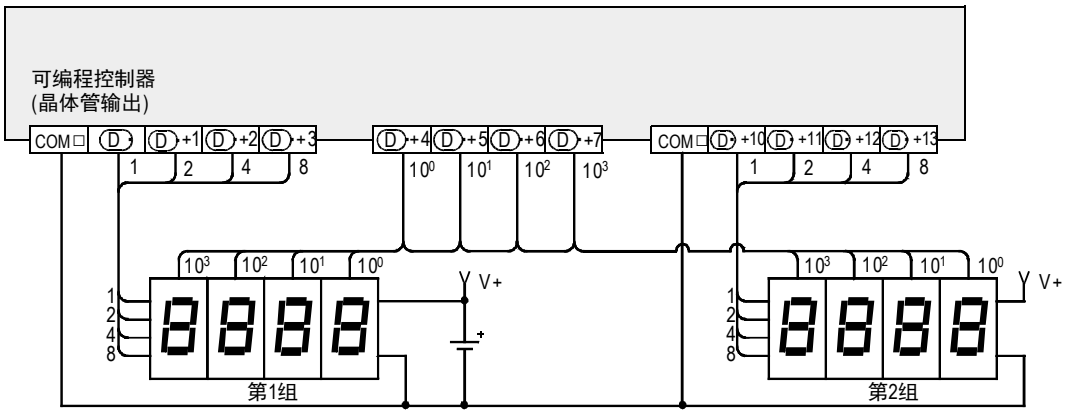
→ n的选择，请参考15.5.2项

1) 数据和选通信号
a) 4位数第1组
将 S・ 的4位数值从BIN转换为BCD后，采用时分方式，从 D・ ～ D・ +3依次对每一位数做输出。
选通信号输出 D・ +4～ D・ +7也依次以时分方式输出，锁定为4位数第1组的7段码显示。
b) 4位数第2组
将 S・ +1的4位数值从 BIN转换为BCD后，采用时分方式，从 D・ +10～ D・ +13依次对每一位数做输出。
选通信号输出 D・ +4～ D・ +7也依次以时分方式输出，锁定为4位数第2组的7段码显示。(选通信号输出 D・ +4～ D・ +7对各组都通用。)

2) S・ 和 S・ +1为0～9,999范围的BIN数据时有效。

3) 连接7段数码管的例子
下图，是以A8系列的基本单元(漏型输出)为例。
有关接线，请参考所使用的可编程控制器的下面一些手册。

→ A8硬件篇手册
→ A8C硬件篇手册



相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	4位数的输出结束后置ON

注意要点

1. 关于更新7段码4位数的显示所需的时间
   更新4位数(1组或2组)的显示所需的时间为扫描时间(运算时间)的12倍。

2. 指令输入为OFF时的动作
   当指令输入为ON时，重复执行动作，但是如在一动作过程中，指令触点变为OFF，则动作会中断，再次为ON时从最初动作开始。

3. 软元件的占用点数
   使用4位数1组时: 占用 (S·) 中指定的起始软元件开始的1点。
   占用 (D·) 中指定的起始软元件开始的8点。即使位数少时，占用的点也不能用于其他用途。
   使用4位数2组时: 占用 (S·) 中指定的起始软元件开始的2点。
   占用 (D·) 中指定的起始软元件开始的 12点。即使位数少时，占用的点也不能用于其他用途。

4. 关于扫描时间(运算周期)和显示时序
   SEGL指令与可编程控制器的扫描时间(运算周期)同步执行。
   为了执行一连串的显示，可编程控制器的扫描时间需要超出10ms。
   不满10ms时，请使用恒定扫描模式，在10ms以上的扫描时间下运行。

5. 关于可编程控制器的输出形式
   请使用晶体管输出型的可编程控制器。

15.5.1 7段数码管的选用要领

按下按键依据7段数码管的电气特性进行选择时，请参考下面内容。

→ 有关接线，请参考所使用的可编程控制器主机的硬件篇手册

1. 依据7段码的规格而定的确认要点

1) 数据输入和选通信号的输入电压・电流特性是否满足可编程控制器的输出规格
   － 输入信号电压(Lo)是否在大约1.5V以下？
   － 输入电压是否在DC5V～DC30V范围内？

2) 是否带BCD译码以及锁存功能？

15.5.2 根据7段数码管显示的规格选择参数n的要领

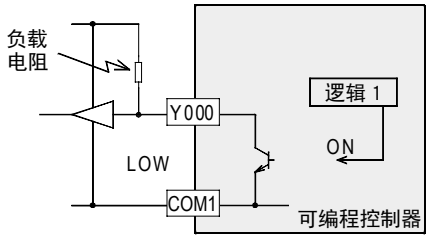
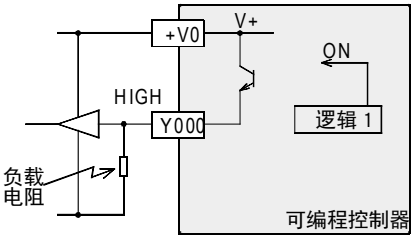
参数n中设定的值，根据7段数码管显示的信号逻辑不同而变化。
请按照下面的要领选择。
在表格的最后一行中备有确认用的栏目。请在相应的正负逻辑栏中做选择，以便在选择参数的设定时，能灵活使用。

1. 关于参数n的作用

根据7段码的数据输入的逻辑(正/负)、选通信号的逻辑(正/负)、以及是4位数1组的控制还是2组的控制，来选择参数n的编号。

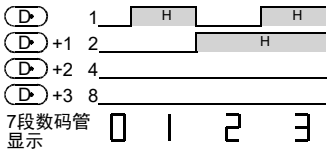
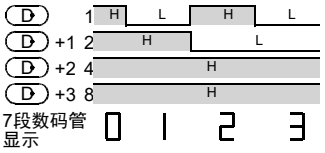
2. 确认可编程控制器的输出逻辑。

可编程控制器的晶体管输出分为漏型输出和源型输出2种，其各自的规格如下所示。

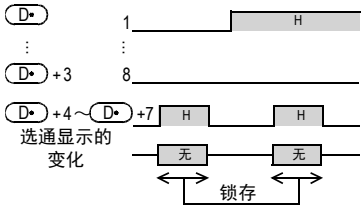
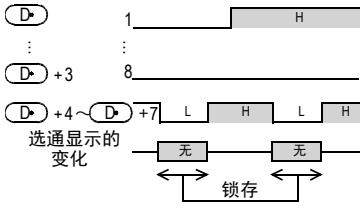
逻辑	负逻辑	正逻辑
输出形式	漏型输出[-公共端]	源型输出[+公共端]
输出回路		
说明	由于是晶体管输出(漏型)，所以内部逻辑为1(ON输出)时，输出为低电平(0V)。 这个称为“负逻辑”。	由于是晶体管输出(源型)，所以内部逻辑为1(ON输出)时，输出为高电平(V+)。 这个称为“正逻辑”。
确认用选项		

3. 确认7段数码管的逻辑。

1) 数据输入

逻辑	负逻辑	正逻辑
时序图		
说明	以低电平决定BCD数据	以高电平决定BCD数据
确认用选项		

2) 选通信号

逻辑	负逻辑	正逻辑
时序图		
说明	低电平保持下被锁存的数据	高电平保持下被锁存的数据
确认用选项		

4. 参数n的选择

根据可编程控制器一侧的正负逻辑以及7段码一侧的正负逻辑，按照下表进行选择。

可编程控制器输出逻辑	数据输入	选通信号	参数n	
			4位数1组时	4位数2组时
负逻辑	负逻辑(一致)	负逻辑(一致)	0	4
		正逻辑(不一致)	1	5
	正逻辑(不一致)	负逻辑(一致)	2	6
		正逻辑(不一致)	3	7
正逻辑	正逻辑(一致)	正逻辑(一致)	0	4
		负逻辑(不一致)	1	5
	负逻辑(不一致)	正逻辑(一致)	2	6
		负逻辑(不一致)	3	7

5. 通过例子说明参数n的选择方法

连接下面的7段码显示时，如为4位数1组时n=1，如为4位数2组时n=5。

1) 可编程控制器的晶体管输出
   - 漏型输出=负逻辑
   - 源型输出=正逻辑

2) 7段码显示
   - 数据输入=负逻辑
   - 选通信号=正逻辑

可编程控制器输出逻辑	数据输入	选通信号	参数n	
			4位数1组时	4位数2组时
负逻辑	负逻辑(一致)	负逻辑(一致)	0	4
		正逻辑(不一致)	1	5
	正逻辑(不一致)	负逻辑(一致)	2	6
		正逻辑(不一致)	3	7

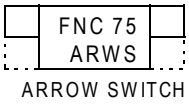
15.6 FNC 75 - ARWS/箭头开关

A8 A8C

概要

通过使用位数移动和增减各位数值用的箭头开关，输入数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	ARWS	连续执行型

32位指令	指令记号	执行条件
—		

2. 设定数据

操作数种类	内容	数据类型
(S・)	输入的起始位软元件编号	BIN16位
(D1・)	保存BCD换算数据的字软元件编号	BIN16位
(D2・)	连接7段数码管显示的起始位软元件(Y)	BIN16位
n	7段数码管显示的位数指定[设定范围: K0～K3]	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)	●	●	●			●	▲												●					
(D1・)												●	●	●	●	●	●	●	●					
(D2・)		●																	●					
n																				●	●			

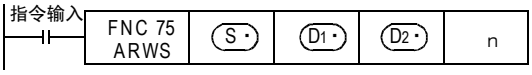
▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

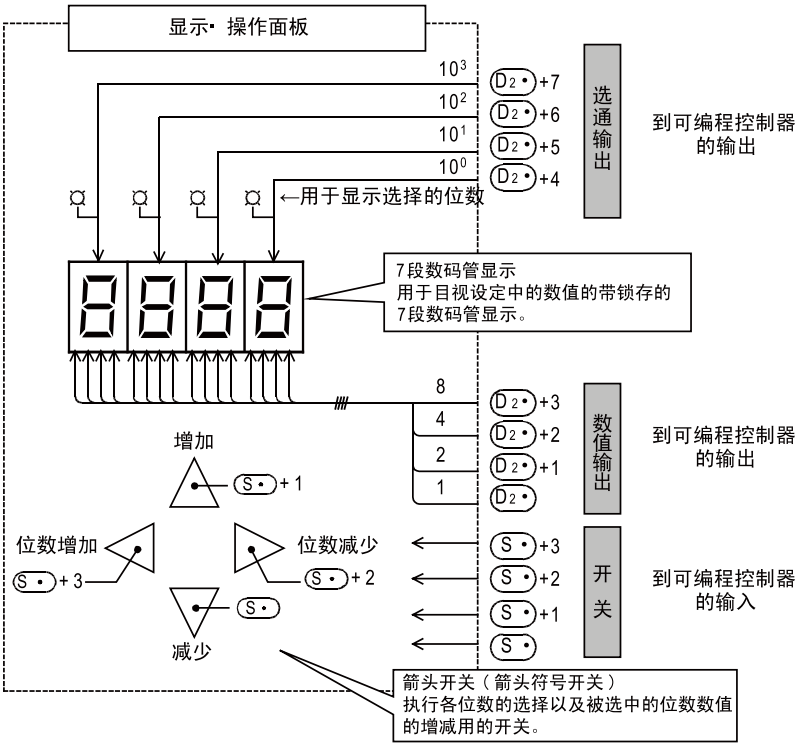
在输入 (S·) ～ (S·)+3上连接4个箭头开关，在输出 (D2·)～ (D2·)+7上连接带BCD译码的7段数码管显示，将数值输入到 (D1·) 中。

1. 16位运算(ARWS)

在 (D1·) 中保存0～9,999的16位BIN值，为了方便起见，以下的说明中以BCD转换显示。
指令输入为ON时，ARWS指令会如下所示动作。



显示及操作部分的内容



1) 带BCD译码的7段数码管显示的位数指定n
在下面的动作说明中，以4位数(10^3位数)为例加以说明。

2) 位数选择开关的动作((S·)+2、 (S·)+3)
－ 位数减少的输入 (S·)+2为ON时的动作
每次按下开关时，位数指定按照10^3→10^2→10^1→10^0→10^3改变。
－ 位数增加的输入 (S·)+3为ON时的动作
每次按下开关时，位数指定按照10^3→10^0→10^1→10^2→10^3改变。

3) 显示选择位数用的LED的动作((D2·)+4～ (D2·)+7)
可以通过选通信号 (D2·)+4～ (D2·)+7，用LED显示指定的位数。

4) 以位数为单位的数据变更开关的动作((S·)、 (S·)+1)
针对上述的“位数选择开关”指定的位数，使其数据发生变化。
－ 增加输入为ON时的动作
每次按下开关时， (D1·) 的内容按照0→1→2→…→8→9→0→1改变。
－ 减少输入为ON时的动作
每次按下开关时， (D1·) 的内容按照0→9→8→7…1→0→9改变。
这些内容可以在7段数码管显示中显示。
如上所示，通过一连串的操作，在查看7段数码管显示的同时，可以将目标数值写入 (D1·) 中。

注意要点

1. 参数n的设定
   请参考SEGL(FNC 74)指令的参数设定。但是，设定范围为0～3。

→ 有关参数的设定，请参考15.5.2项

2. 关于可编程控制器的输出形式
   请使用晶体管输出型的可编程控制器。

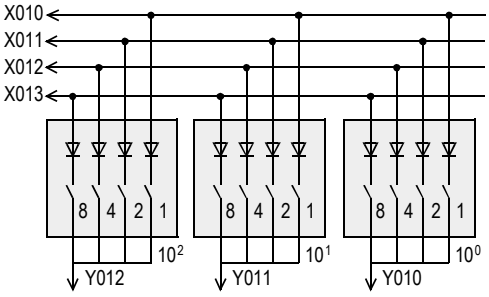
3. 关于扫描时间(运算周期)和显示时序
   ARWS指令与可编程控制器的扫描时间(运算周期)同步执行。
   为了执行一连串的显示，可编程控制器的扫描时间需要超出10ms。
   不满10ms时，请使用恒定扫描模式，在10ms以上的扫描时间下运行。

4. 软元件的占用点数
   1) 软元件 (S・) 的输入占用4点。
   2) 软元件 (D2・) 的输出占用8点。

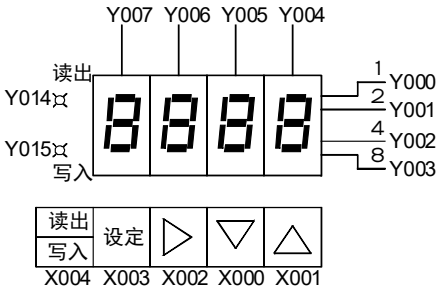
5. 指令的使用次数的限制
   ARWS指令，在程序只可以使用1个。
   要使用多个时，请使用变址修饰(V、Z)功能编程。

程序举例

1. 更改定时器的设定以及显示其当前值时
   1) 使用3位数的数字开关指定定时器编号



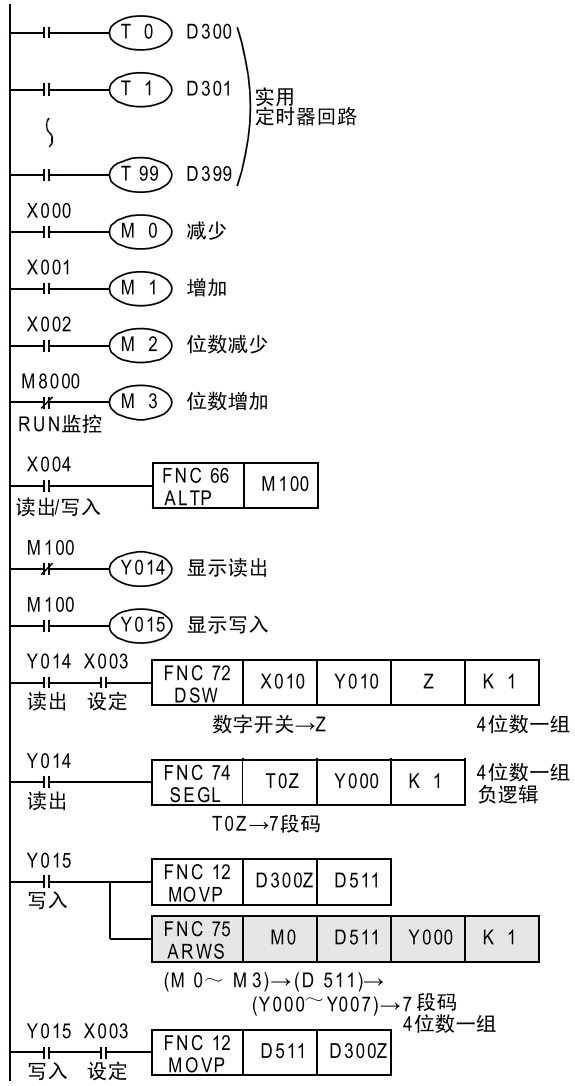
   2) 使用箭头开关设定定时器的常数



操作说明
每次按下读出/写入键时，读出、写入的LED就会交替亮灯。

- 读出时
  使用数值开关指定定时器编号后，按设定开关(X003)。
- 写入时
  使用箭头开关，查看7段码的同时设定数值，按X003。

程序


T 0 D300
T 1 D301
实用
定时器回路
T 99 D399
X000
M 0 减少
X001
M 1 增加
X002
M 2 位数减少
M8000
M 3 位数增加
RUN监控
X004
读出/写入
FNC 66 ALTP M100
M100
Y014 显示读出
M100
Y015 显示写入
Y014 X003
读出 设定
FNC 72 DSW X010 Y010 Z K 1
数字开关→Z
4位数一组
Y014
读出
FNC 74 SEGL T0Z Y000 K 1
4位数一组
负逻辑
T0Z→7段码
Y015
写入
FNC 12 MOVP D300Z D511
FNC 75 ARWS M0 D511 Y000 K 1
(M 0～ M 3)→(D 511)→
(Y000～Y007)→7 段码
4位数一组
Y015 X003
写入 设定
FNC 12 MOVP D511 D300Z

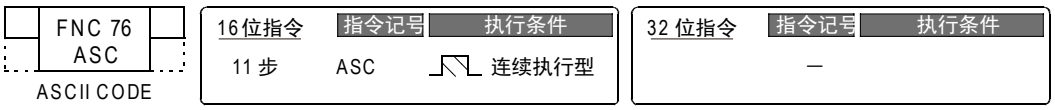
15.7 FNC 76 - ASC/ASCII数据输入

A8 A8C

概要

将半角/英文数字字符串转换成ASCII码的指令。
用于在外部显示器中选择显示多个消息。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S)	从计算机输入的8个字符的半角英文数字 (仅	字符串 ASCI码)
(D・)	保存ASCII数据的起始字软元件编号	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S)																							●*1	
(D・)												●	●	●	●	●			●					

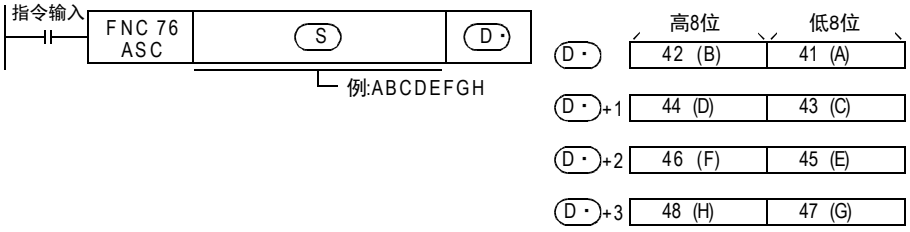
*1. 在ASC指令中，不需要给 (S) 字符串添加["]。

功能和动作说明

1. 16位运算(ASC)

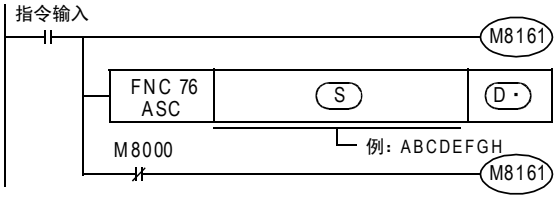
将 (S) 中指定的半角、英文、数字字符串转换成ASCII码后，依次传送到 (D・) 中。

- 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。
- 转换后的ASCII码按照低8位、高8位的顺序，每2个字符/1字节地保存在 (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ASCII码，然后将其依次传送到 (D·) 的低 (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	ASC(FNC 76)、RS(FNC 80)、ASCI(FNC 82)、HEX(FNC 83)、CCD(FNC 84)指令的8位处理模式 OFF：每2个字符被按照低8位、高8位的顺序，加以保存(2个字符/两个字) ON ：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·) 占用字符数÷2点(不能整除时，则将小数点进位。)
   2) 扩展功能ON时
      — (D·) 占用的点数和字符数占用的数目相同。
2. 使用RS(FNC 80)、ASCI(FNC 82)、HEX(FNC 83)、CCD(FNC 84)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，以不造成影响。

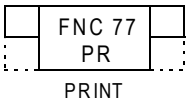
15.8 FNC 77 - PR/ASCII码打印

A8 A8C

概要

该指令是将ASCII码的数据并行输出到输出(Y)。

1. 指令格式



16位指令	指令记号	执行条件
5步	PR	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存ASCII码数据的软元件的起始编号	字符串(仅ASCII码)
(D・)	输出ASCII码数据的起始Y编号	位

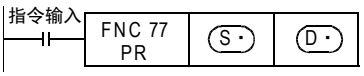
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)		●																	●					

功能和动作说明

1. 16位运算(PR)

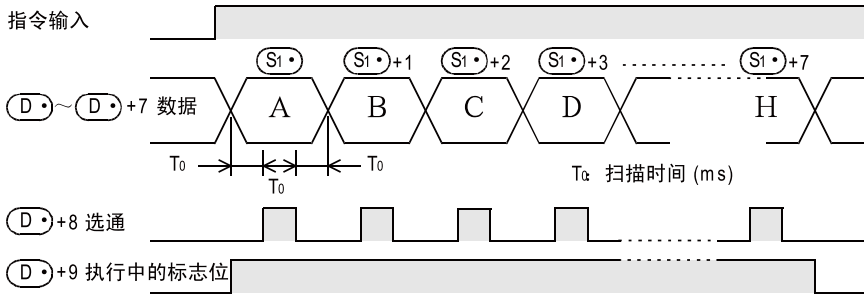
将(S・)～(S・)+7的低8位(1个字节)中保存的ASCII码，按时分方式逐个字符输出到(D・)～(D・)+7中。



(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，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

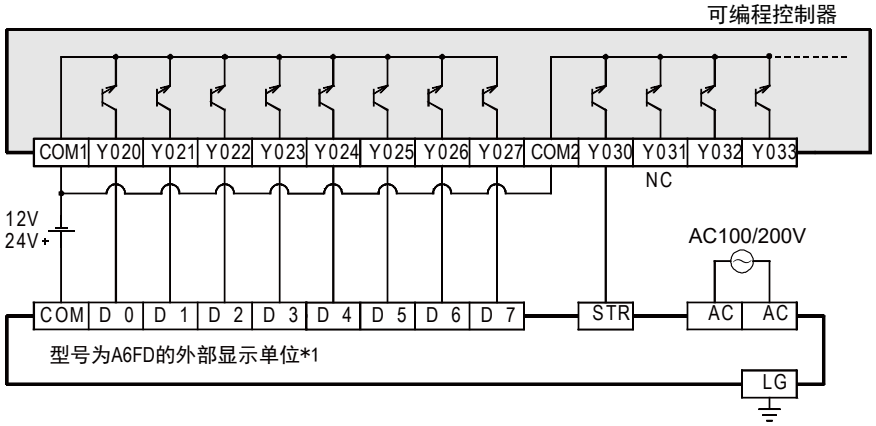
- (D・)～(D・)+7 ：发送输出　　(D・)为低位侧，(D・)+7为高位侧。
- (D・)+8 ：选通信号
- (D・)+9 ：执行中的标志位　按照上述的时序图动作。

扩展功能

1. 16个字节的串行输出

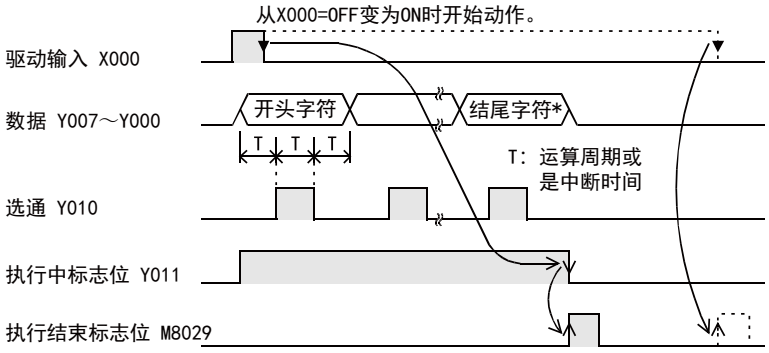
通过特殊辅助继电器M8027的ON/OFF，指令2次驱动中附带的输出字符数会变化。
M8027=OFF时为8个字节的串行输出(固定为8个字符)，M8027=ON时为16个字节的串行输出(1～16个字符)。
下面列举了在显示设备(例如: 型号为A6FD的外部显示单元*1)等中，如下所示显示16个字符(1个字符/1个字节)的例子进行说明。
显示的数据是例如D300～D307中以16进制代码形式保存的数据。

1) 型号为A6FD的外部显示单元*1的连接例子
下图中的可编程控制器是以A8-32M□连接为例的。



*1. 型号为A6FD的外部显示单元于2002年11月终止生产。

2) 时序图(M8027=ON时)



*数据中（16个字符中）有H00（NUL代码）时，H00（NUL代码）的前1个字符为最后的字符。

相关软元件

软元件	名称	内容
M8027*1	PR模式	OFF : 8个字节串行输出(固定为8个字符) ON : 16个字节串行输出(1～16个字符)

*1. 从RUN→STOP时清除

注意要点

1. 指令输入及指令的动作

   指令输入=ON ：即使是执行连续为ON的指令或者执行脉冲指令，只要循环一次的输出结束，则执行就结束。
   仅当M8027=ON时M8029动作。

   指令输入=OFF ：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. 关于可编程控制器的输出

   请使用晶体管输出型的可编程控制器。

4. 数据中存在00H(NUL)时(M8027=ON时)

   指令执行结束，不输出剩余的数据。
   此外，M8029维持一次运算周期为ON。

15.9 FNC 78 - FROM/BFM的读出

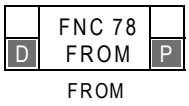
A2 A2C A8 A8C

概要

将特殊功能单元/模块的缓冲存储区(BFM)中的内容读入可编程控制器的指令。
用该指令一次读取多个缓冲存储区(BFM)的数据时，有可能发生看门狗定时器错误。即使分割读取数据也不会对控制产生影响时，请使用RBFM(FNC 278)指令。

→ 关于RBFM(FNC 278)指令，请参考31.1节

1. 指令格式



16位指令	指令记号	执行条件
9步	FROM	连续执行型
	FROMP	脉冲执行型

32位指令	指令记号	执行条件
17步	DFROM	连续执行型
	DFROMP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
m1	特殊功能单元/模块的单元号 (从基本单元的右侧开始依次为K0～K7)	BIN16/32位
m2	传送源缓冲存储区(BFM)编号	BIN16/32位
(D•)	传送目标的软元件编号	BIN16/32位
n	传送点数	BIN16/32位

3. 对象软元件

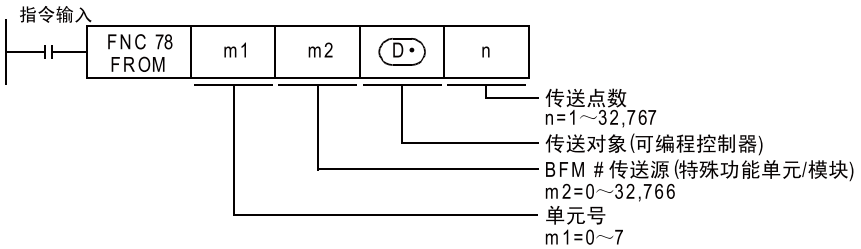
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
m1														●	●					●	●			
m2														●	●					●	●			
(D•)									●	●	●	●	●	●	●		●	●	●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(FROM、FROMP)

→ 关于FROM/TO指令的通用事项，请参考15.9.1项

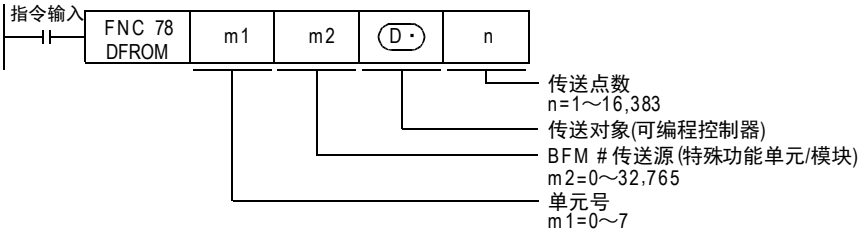
特殊扩展(BFM)→可编程控制器(字软元件)
将单元号为m1的特殊功能单元/模块中的缓冲存储区 (BFM)m2开始的n点 16位数据传送到(读出)可编程控制器内以 (D•) 开始的n点中。



2. 32位运算(DFROM、DFROMP)

特殊扩展(BFM)→可编程控制器(字软元件)

将单元号为m1的特殊功能单元/模块中的缓冲存储区 (BFM)[m2+1、m2]开始的n点 32位数据传送到(读出) 可编程控制器内以[(D・) +1、 (D・)]开始的n点中。



相关软元件

软元件	名称	内容
M8028	允许中断标志位	在FROM/TO指令执行过程中禁止/允许中断 →有关详细内容，请参考下一页以后的[关于在FROM/TO执行过程中的中断接受(M8028)] OFF：禁止中断(处理FROM/TO指令后执行中断) ON ：允许中断

注意要点

- 关于 (D・) 位软元件的位数指定
  当为16位运算指令时设置为K1～K4，32位运算时设置为K1～K8。
- 将D、R指定为32位指令的m1、m2、n时，[m1+1,m1]、[m2+1,m2][n+1,n]的32位值便生效，敬请注意。
  DFROM D0 D2 D100 R0时，则m1=[D1, D0]、m2=[D3, D2]、n=[R1, R0]。

程序举例

使用FROM指令、缓冲存储区的直接指定*1，将特殊功能单元/模块的缓冲存储区 (BFM)的内容读出到(传送给)指定的数据寄存器(D)、扩展寄存器(R)以及辅助继电器(M)的指定位数等。

*1. 仅适用于A8/A8C可编程控制器。

例) 将A8C-32MT-LT(-2)内置的CC-Link主站(单元号固定为0)的BFM#4(异常站信息)的内容读出到D0中的程序。

－ FROM指令时

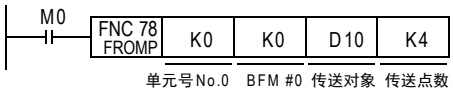


－ MOV指令时

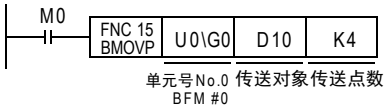


将A8C-32MT-LT(-2)内置的CC-Link/LT主站(单元号固定为0)的BFM #0～#3(远程站的连接信息)的内容读出到D10～D13中的程序。

－ FROM指令时



－ BMOV指令时



15.9.1 FROM/TO指令的共通事项(详细)

有关操作数的指定内容

1. 特殊功能单元/模块的单元号[m1]

单元号是用于指定FROM/TO指令是针对哪个设备进行动作的。

设定范围: K0～K7

单元No. 0 内置 CC-Link		单元No. 1	单元No. 2		单元No. 3
A8C-32MT-LT(-2)型基本单元	输入输出扩展模块	特殊功能模块	特殊功能模块	输入输出扩展模块	特殊功能模块

对于可编程控制器而言，其连接的特殊功能单元/模块的单元号会自动被分配。

单元号是从离基本单元最近的模块开始依次为No.0→No.1→No.2…。

A8C-32MT-LT(-2)可编程控制器的情况下，由于内置CC-Link/LT主站，单元号是从离基本单元最近的模块开始依次为No.1→No.2→No.3…。

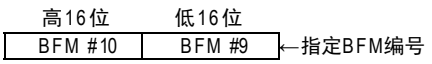
2. 缓冲存储区(BFM)编号[m2]

在特殊功能单元/模块中，最多内置了32,767点16位的RAM内存，这个就称为缓冲存储区。

缓冲存储区的编号为0～32766，其内容由各设备的控制目的而决定。

设定范围: K0～K32766

- 在32位指令中处理BFM时，指定的BFM为低16位、编号紧接的BFM为高16位。

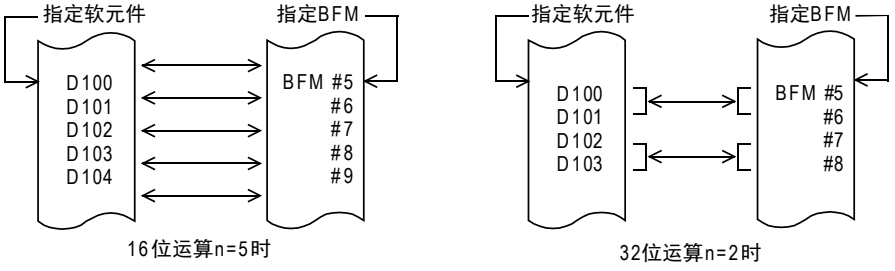


3. 传送点数[n]

设定范围: K1～K32767

用n指定传送的字点数。

16位指令中的n=2和32位指令中的n=1是相同的意思。



关于在FROM/TO执行过程中的中断接受(M8028)

1. M8028=OFF时
   FROM/TO指令执行过程中，自动变为禁止中断状态，不执行输入中断和定时器中断。
   在此期间产生的中断会在执行完FROM/TO指令之后立即被执行。
   此外，FROM/TO指令也可以在中断程序中使用。

2. M8028=ON时
   在FROM/TO指令执行过程中如产生中断，则中断执行，然后执行中断程序。
   但是，不能在中断程序中使用FROM/TO指令。

发生看门狗定时器错误时的对应措施

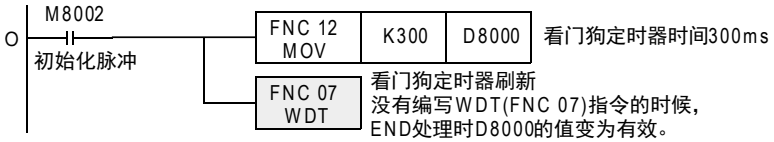
1. 产生看门狗定时器错误的原因
   下面的情况下，会发生看门狗定时器错误。
   1) 连接较多特殊扩展设备时
      在连接了较多特殊扩展设备(定位、凸轮开关、链接、模拟量等)的系统配置中，可编程控制器 RUN时执行的缓冲存储区的初始化时间会变长，因而运算时间会延长，会产生看门狗定时器错误。
   2) 同时驱动多个FROM/TO指令时
      当执行了多条FROM/TO 指令对多个缓冲存储区进行传送操作时，运算时间会延长，有可能会发生看门狗定时器错误的情况。

2. 对应方法
   1) 使用RBFM(FNC 278)、WBFM(FNC 279)指令的方法[Ver.2.20以上]

A1 A2 A2C A8 A8C

→ 有关BFM分割读出[RBFM(FNC 278)]，请参考31.1节
→ 有关BFM分割写入[WBFM(FNC 279)]，请参考31.2节

   2) 更改看门狗定时器时间的方法
      通过改写D8000(看门狗定时器时间)的内容，可以更改看门狗定时器的检测时间。
      输入下面的程序后，此后的顺控程序就会按照新的看门狗定时器时间进行监视。



   3) FROM/TO指令的执行时序的变更
      请错开FROM/TO指令的执行时序，缩短运算时间。

关于特殊功能单元/模块的使用

关于特殊功能单元/模块的连接方法、允许连接数量以及输入输出编号的使用等，请参考可编程控制器主机的手册以及各特殊功能单元/模块的手册。

15.10 FNC 79 - TO/BFM的写入

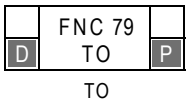
A2 A2C A8 A8C

概要

将数据从可编程控制器中写入到特殊功能单元/模块的缓冲存储区(BFM)中的指令。
用该指令一次向多个缓冲存储区(BFM)写入数据时，有可能发生看门狗定时器错误。即使分割写入数据也不会对控制产生影响时，请使用WBFM(FNC 279)指令。

→ 有关WBFM(FNC 279)指令，请参考31.2节

1. 指令格式



16位指令	指令记号	执行条件
9步	TO	连续执行型
	TOP	脉冲执行型

32位指令	指令记号	执行条件
17步	DTO	连续执行型
	DTOP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
m1	特殊功能单元/模块的单元号 (从基本单元的右侧开始依次为K0～K7)	BIN16/32位
m2	传送目标缓冲存储区(BFM)编号	BIN16/32位
(S·)	保存传送源数据的软元件编号	BIN16/32位
n	传送点数	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
m1														●	●					●	●			
m2														●	●					●	●			
(S·)								●	●	●	●	●	●	●	●		●	●	●	●	●			
n														●	●					●	●			

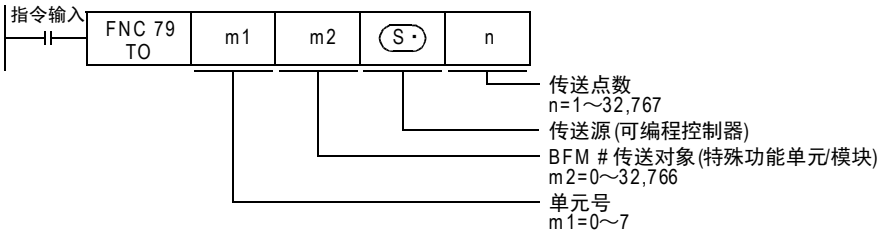
功能和动作说明

1. 16位运算(TO、TOP)

→ 关于FROM/TO指令的通用事项，请参考15.9.1项

可编程控制器(字软元件)→特殊扩展(BFM)

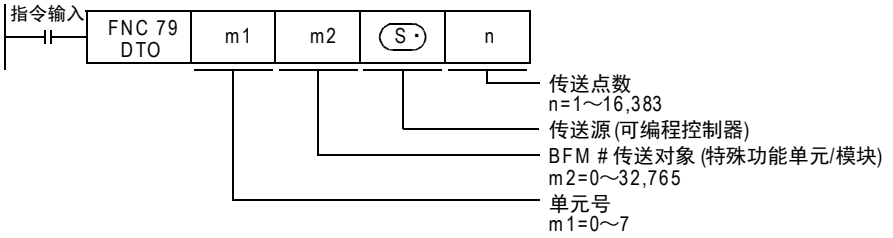
将可编程控制器中(S·)起始的n点16位数据传送到(写入)单元号为m1的特殊功能单元/模块中的缓冲存储区(BFM)m2开始的n点中。



2. 32位运算(DTO、DTOP)

可编程控制器(字软元件)→特殊扩展(BFM)

将可编程控制器中以[(S・) ， (S・)+1]开始的n点 32位数据传送到(写入)单元号为 m1的特殊功能单元/模块中的缓冲存储区(BFM)[m2+1，m2]开始的n点中。



相关软元件

软元件	名称	内容
M8028	允许中断标志位	在FROM/TO指令执行过程中禁止/允许中断 →有关详细内容，请参考15.9.1项中的[关于在FROM/TO执行过程中的中断接受(M8028)] OFF：禁止中断(处理FROM/TO指令后执行中断) ON ：允许中断

注意要点

- 关于 (S・) 位软元件的位数指定
  当为16位运算指令时设置为K1～K4，32位运算时设置为K1～K8。
- 将D、R指定为32位指令的m1、m2、n时，[m1+1,m1]、[m2+1,m2][n+1,n]的32位值便生效，敬请注意。
  DTO D0 D2 D100 R0时，则m1=[D1, D0]、m2=[D3, D2]、n=[R1, R0]。

程序举例

使用TO指令、缓冲存储区的直接指定*1，将特殊功能单元/模块的缓冲存储区(BFM)的内容写入(传送到)指定的数据寄存器(D)、扩展寄存器(R)以及辅助继电器(M)的指定位数(K，H)。

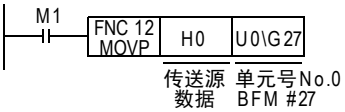
*1. 仅适用于A8/A8C可编程控制器。

例) 将[H0]写入在A8C-32MT-LT(-2)内置的CC-Link/LT主站(单元号固定为0)的BFM #27(命令)中的程序

－ TO指令时

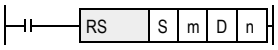
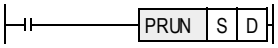
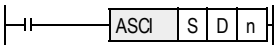
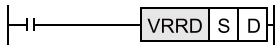
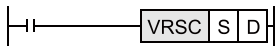
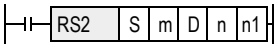
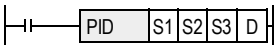


－ MOV指令时



16. 外部设备SER（选件设备）—FNC 80～FNC 89

FNC 80～FNC 89中，提供了对连接在串行通信口上的特殊适配器进行控制的指令。
此外，还包括了PID运算指令。

FNC No.	指令记号	符号	功能	参考
80	RS		串行数据的传送	16.1节
81	PRUN		8进制位传送	16.2节
82	ASCI		HEX→ASCII的转换	16.3节
83	HEX		ASCII→HEX的转换	16.4节
84	CCD		校验码	16.5节
85	VRRD		电位器读出	16.6节
86	VRSC		电位器刻度	16.7节
87	RS2		串行数据传送2	16.8节
88	PID		PID运算	16.9节
89	—			—

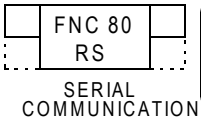
16.1 FNC 80 - RS/串行数据传送

A1 A2 A2C A8 A8C

概要

通过安装在基本单元上的RS-232C或RS-485串行通信口(仅通道1)进行无协议通信，从而执行数据的发送和接收的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	RS	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存发送数据的数据寄存器的起始软元件	BIN16位/字符串
m	发送数据的字节数[设定范围: 0～4096]	BIN16位
(D・)	数据接收结束时，保存接收数据的数据寄存器的起始软元件	BIN16位/字符串
n	接收数据的字节数[设定范围: 0～4096]	BIN16位

3. 对象软元件

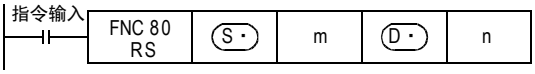
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●				●					
m														●	●					●	●			
(D・)														●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(RS)

该指令是用于通过安装在基本单元上的RS-232C或RS-485串行通信口进行无协议通信，从而执行数据的发送和接收的指令。

→ 关于详细说明，请参考通信控制手册



相关软元件

→ 有关详细说明，请参考通信控制手册

软元件	名称
M8063*1	串行通信错误1
M8121*2	发送等待标志位
M8122*2	发送请求
M8123*2	接收结束标志位
M8124	载波的检测标志位
M8129	超时的判断标志位
M8161*4	8位处理模式

软元件	名称
D8120*3	设定通信格式
D8122*4	发送数据的剩余点数
D8123*4	接收点数的监控
D8124	报头
D8125	报尾
D8129*3	设定超时时间
D8063*1	串行通信错误1的错误代码编号
D8405	通信参数的显示
D8419	动作模式的显示

*1. A8/A8C可编程控制器时，电源从OFF变为ON时清除
A8/A8C以外的可编程控制器时，从STOP→RUN时清除

*2. 下面的情况下清除
- RUN→STOP时
- RS指令OFF时

*3. 停电保持

*4. 从RUN→STOP时清除

系统配置

为了使用这个指令，需要在基本单元中安装下表中的任一产品。

→ 有关系统配置，请参考所使用的可编程控制器主机的硬件篇手册

→ 有关详细说明，请参考通信控制手册

可编程控制器	通信的种类	选件
A8, A8C-32MT-LT(-2)	RS-232C通信	A8-232-BD或A8-232ADP(-MB)(需要A8-CNV-BD)
	RS-485通信	A8-485-BD或A8-485ADP(-MB)(需要A8-CNV-BD)
A8C(D,DSS)	RS-232C通信	A8-232ADP(-MB)
	RS-485通信	A8-485ADP(-MB)

关于RS(FNC 80)指令和RS2(FNC 87)指令的区别

项目	RS2指令	RS指令	备注
报头点数	1～4个字符 (字节)	最大1个字符 (字节)	用RS2指令，报头或报尾中最多可以指定4个字符(字节)。
报尾点数	1～4个字符 (字节)	最大1个字符 (字节)	
附加和校验	可以自动附加	请用用户程序 对应。	用RS2指令，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

→ 关于其他的注意事项，请参考通信控制手册

- RS(FNC 80)指令可以用于通道1。(不能用于通道2。)
- RS(FNC 80)和RS2(FNC 87)指令，请勿同时驱动针对同一个通信口的多个这种指令。
- 不可以对同一个通信口使用RS(FNC 80)、RS2(FNC 87)指令和IVCK(FNC 270)、IVDR(FNC 271)、IVRD(FNC 272)、IVWR(FNC 273)、IVBWR(FNC 274)指令。

16.2 FNC 81 - PRUN/8进制位传送

A1 A2 A2C A8 A8C

概要

这个指令是将被指定了位数的 (S・) 和 (D・) 的软元件8进制数处理，并传送数据。

1. 指令格式

FNC 81 PRUN	16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
D P	5步	PRUN	连续执行型	9步	DPRUN	连续执行型
PARALLEL RUNNING		PRUNP	脉冲执行型		DPRUNP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	位数指定*1	BIN16/32位
(D・)	传送目标软元件编号*1	BIN16/32位

*1. 指定要素编号的最低位数请设置为0。

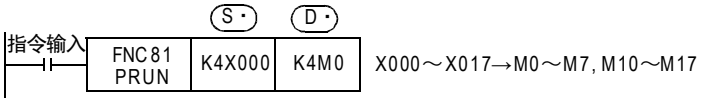
3. 对象软元件

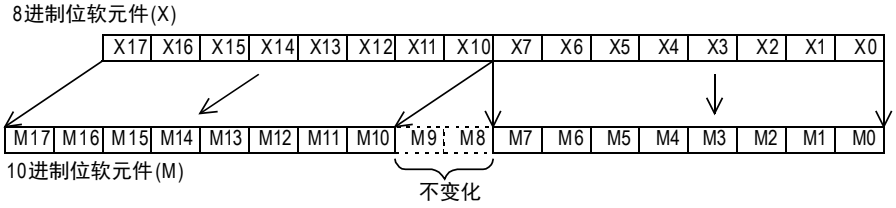
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●		●									●					
(D・)									●	●									●					

功能和动作说明

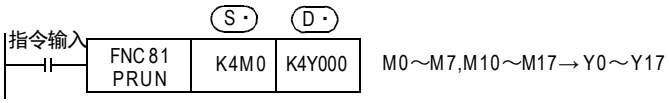
1. 16位运算(PRUN、PRUNP)

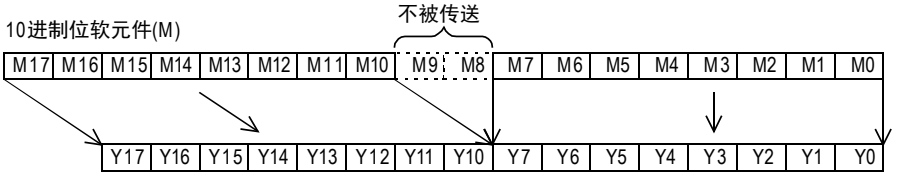
8进制位软元件→10进制位软元件





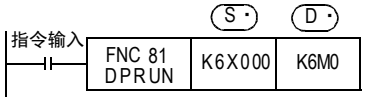
10进制位软元件→8进制位软元件





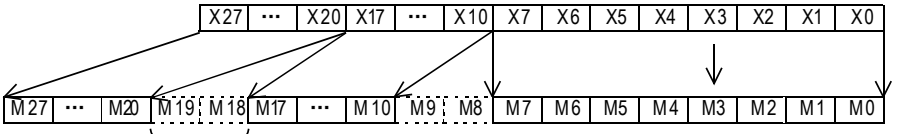
2. 32位运算(DPRUN、DPRUNP)

8进制位软元件➞10进制位软元件



X000～X027 →M0～M7, M10～M17, M20～M27

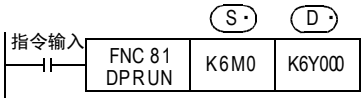
8进制位软元件(X)



不变化

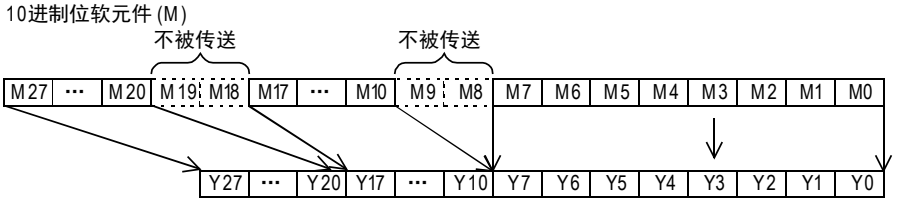
10进制位软元件(M)

10进制位软元件➞8进制位软元件



M0～M7,M10～M17,M20～M27 → Y000～Y027

10进制位软元件 (M)



8进制位软元件(Y)

16.3 FNC 82 - ASCI/HEX→ASCII的转换

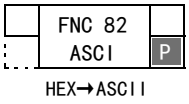
A1 A2 A2C A8 A8C

概要

这个指令是将HEX转换成ASCII码的指令。
此外，也有将BIN数据转换成ASCII码的BINDA(FNC 261)指令和将2进制浮点数数据转换成ASCII码的ESTR(FNC 116)指令。

→ 有关BINDA(FNC 261)指令，请参考29.6节
→ 有关ESTR(FNC 116)指令，请参考18.4节

1. 指令格式



16位指令	指令记号	执行条件
7步	ASCI	连续执行型
	ASCIP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要转换的HEX的软元件的起始编号	BIN16位
(D・)	保存转换后的ASCII码的软元件的起始编号 (仅	字符串 ASCII码)
n	要转换的HEX的字符数(位数)[设定范围: 1～256]	BIN16位

3. 对象软元件

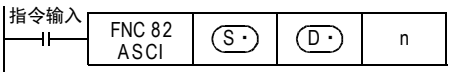
操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲			●					
n														●	●					●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作

1. 16位运算(ASCI/ASCIP)

将(S・)开始的软元件中保存的HEX的n个字符(位数)转换成ASCII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。



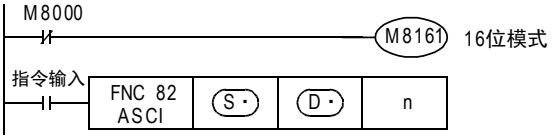
2. 《16位转换模式》M8161=OFF时(M8161为RS、HEX、CCD、CRC指令通用)

将 (S·) 开始的软元件中保存的各位HEX数据转换成ASCII码，然后传送到 (D·) 开始的各软元件的高低8位(字节)中的指令。用n指定转换的位数(字符数)。

(D·) 分低8位，高8位来保存ASCII数据。

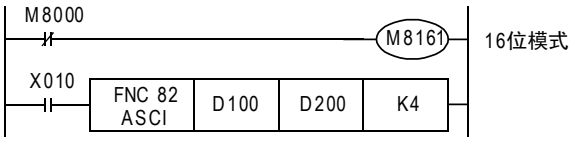
此外，M8161为RS、HEX、CCD、CRC指令通用。使用16位转换模式时，请将M8161一直置为OFF。

M8161在RUN→STOP时被清除。



动作

当为下述程序时，如下所示执行转换。

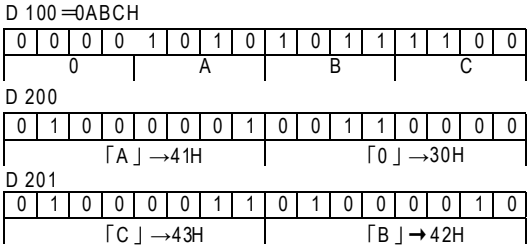


(S·) 开始的软元件
(D100)=0ABCH
(D101)=1234H
(D102)=5678H

指定位数(字符数)及转换结果

n / (D·)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200低	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D200高		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D201低			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D201高				[C]	[B]	[A]	[0]	[4]	[3]
D202低					[C]	[B]	[A]	[0]	[4]
D202高						[C]	[B]	[A]	[0]
D203低			不变化				[C]	[B]	[A]
D203高								[C]	[B]
D204低									[C]

n=K4时的位结构



ASCII码
「0」=30H「1」=31H「5」=35H
「A」=41H「2」=32H「6」=36H
「B」=42H「3」=33H「7」=37H
「C」=43H「4」=34H「8」=38H

- 使用打印机等输出BCD数据时，需要在执行该指令之前，先进行BIN→BCD的转换。

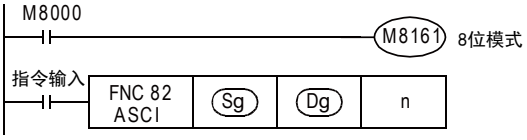
3. 《8位转换模式》M8161=ON时(M8161为RS、HEX、CCD、CRC指令通用)

将 (S・) 开始的软元件中保存的各位HEX数据转换成ASCII码，然后传送到 (D・) 开始的各软元件的低8位(字节)中的指令。用n指定转换的位数(字符数)。

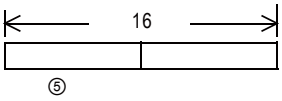
(D・) 的高8位为0。

此外，M8161为RS、HEX、CCD、CRC指令通用。使用8位转换模式时，请将M8161一直置为ON。

M8161在RUN→STOP时被清除。

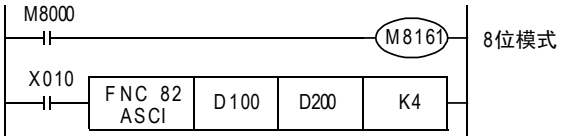


M8161=ON时为8位模式、
如下所示执行转换。



动作

当为下述程序时，如下所示执行转换。



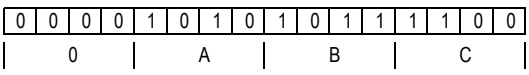
(S・) 开始的软元件
(D100)=0ABCH
(D101)=1234H
(D102)=5678H

指定位数(字符数)及转换结果

n / (D・)	K1	K2	K3	K4	K5	K6	K7	K8	K9
D200	[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]	[8]
D201		[C]	[B]	[A]	[0]	[4]	[3]	[2]	[1]
D202			[C]	[B]	[A]	[0]	[4]	[3]	[2]
D203				[C]	[B]	[A]	[0]	[4]	[3]
D204					[C]	[B]	[A]	[0]	[4]
D205						[C]	[B]	[A]	[0]
D206		不变化					[C]	[B]	[A]
D207								[C]	[B]
D208									[C]

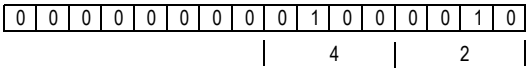
n=K2时的位结构

D 100 = OABCH

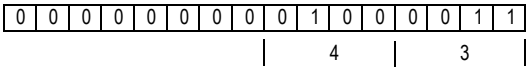


ASCII码
「0」=30H「1」=31H「5」=35H
「A」=41H「2」=32H「6」=36H
「B」=42H「3」=33H「7」=37H
「C」=43H「4」=34H「8」=38H

D 200＝B的ASCII码＝42H



D 201＝C的ASCII码＝43H



- 使用打印机等输出BCD数据时，需要在执行该指令之前，先进行BIN→BCD的转换。

16.4 FNC 83 - HEX/ASCII→HEX的转换

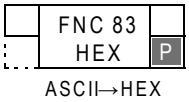
A1 A2 A2C A8 A8C

概要

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DABIN(FNC 260)指令和将ASCII码转换成2进制浮点数的EVAL(FNC 117)指令。

→ 有关DABIN(FNC 260)指令，请参考29.5节
→ 有关EVAL(FNC 117)指令，请参考18.5节

1. 指令格式



16位指令	指令记号	执行条件
7步	HEX	连续执行型
	HEXP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存要转换的ASCII码的软元件的起始编号	字符串(仅ASCII码)*1
(D·)	保存转换后的HEX的软元件的起始编号	BIN16/32位
n	要转换的ASCII码的字符数(字节数)[设定范围: 1～256]	BIN16位

*1. 请仅将ASCII码设定为“0”～“9”、“A”～“F”。

3. 对象软元件

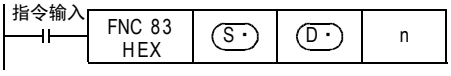
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲			●	●	●			
(D·)									●	●	●	●	●	●	●	▲	●	●	●					
n														●	●					●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(HEX/HEXP)

将(S·)开始的软元件中保存的ASCII码的n个字符转换成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。

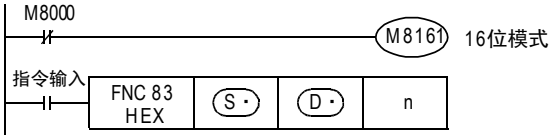


2. 《16位转换模式》M8161=OFF时(M8161为RS、ASCI、CCD、CRC指令通用)

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。

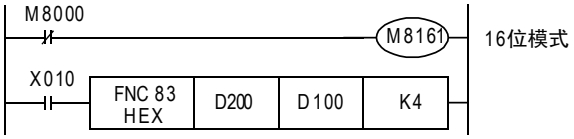
M8161为RS、ASCI、CCD、CRC指令通用。使用16位转换模式时，请将M8161一直置为OFF。

M8161在RUN→STOP时被清除。



动作

当为下述程序时，如下所示执行转换。



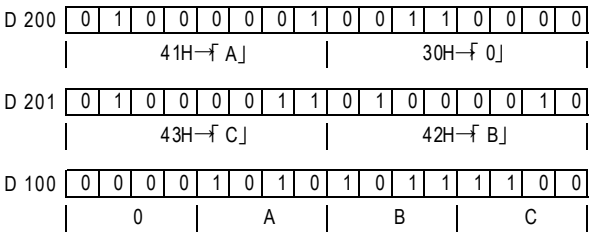
转换源数据

(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

(D・) / n	D102	D101	D100
1	不变化		・・・0H
2			・・0AH
3			・0ABH
4			0ABCH
5		・・・0H	ABC1H
6		・・0AH	BC12H
7		・0ABH	C123H
8		0ABCH	1234H
9	・・・0H	ABC1H	2345H

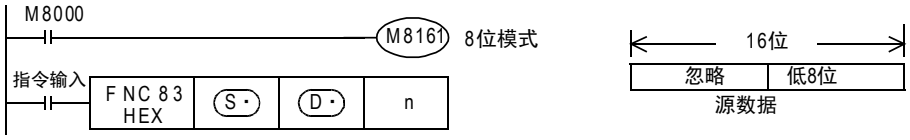
n＝K4时



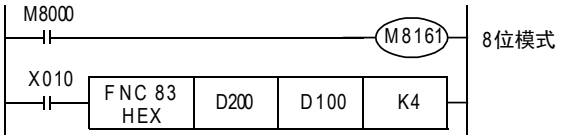
- 输入数据为BCD时，需要在执行该指令后，进行BCD→BIN转换。
- 在HEX指令中，被保存在(S・)中的数据如不是ASCII码，则会出现运算错误，不能进行HEX转换。尤其是当M8161为OFF时，在(S・)的高8位中也需要事先保存ASCII码，请务必注意。

3. 《8位转换模式》M8161=ON时(M8161为RS、ASCI、CCD、CRC指令通用)

将保存在 (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为RS、ASCI、CCD、CRC指令通用。使用8位转换模式时，请将M8161一直置为ON。
M8161在RUN→STOP时被清除。



动作
当为下述程序时，如下所示执行转换。



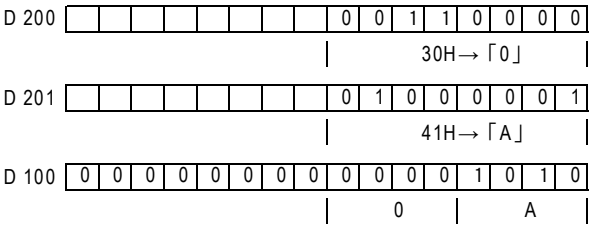
转换源数据

(S·)	ASCII码	HEX转换
D200	30H	0
D201	41H	A
D202	42H	B
D203	43H	C
D204	31H	1
D205	32H	2
D206	33H	3
D207	34H	4
D208	35H	5

指定字符数及转换结果 [·]为0。

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

n＝K2时



- 输入数据为BCD时，需要在执行该指令后，进行BCD→BIN转换。

16.5 FNC 84 - CCD/校验码

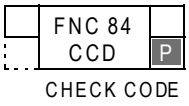
A1 A2 A2C A8 A8C

概要

在通信等中使用的错误校验方法有水平校验以及和校验，该指令是用于计算校验值的。在错误校验的方法中，除这些以外还有CRC(Cyclic Redundancy Check: 循环冗余校验)。求CRC值时，请使用CRC指令。

→ 有关CRC指令，请参考24.4节
→ 有关[NEG(FNC 29)指令]请参考10.10节

1. 指令格式



16位指令	指令记号	执行条件
7步	CCD	连续执行型
	CCDP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	对象软元件的起始编号	BIN16位/字符串
(D・)	保存计算出的数据的软元件的起始编号	BIN16位/字符串
n	数据数[设定范围: 1～256]	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲			●					
(D・)									●	●	●	●	●	●	●				●					
n														●	●					●	●			

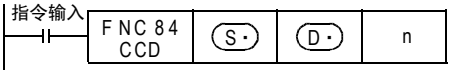
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(CCD/CCDP)

计算出(S・)～(S・)+n-1中保存的数据的总和及水平校验，将总和数据保存在(D・)中，将水平校验保存在(D・)+1中。

在这个指令中，作为计算时使用的模式有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。

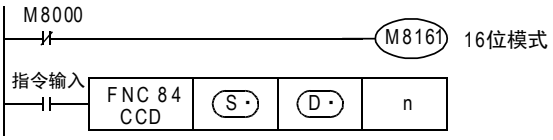


2. 《16位转换模式》M8161=OFF时(M8161为RS、ASCI、HEX、CRC指令通用)

关于以(S·)开始的n点数据，将其高低各8位数据的总和以及水平校验分别保存到(D·)和(D·)+1的软元件中。

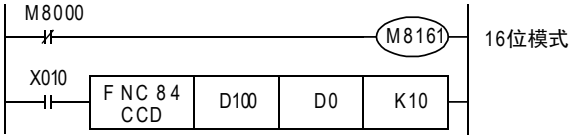
M8161为RS、ASCI、HEX、CRC指令通用。使用16位转换模式时，请将M8161一直置为OFF。

M8161在RUN→STOP时被清除。

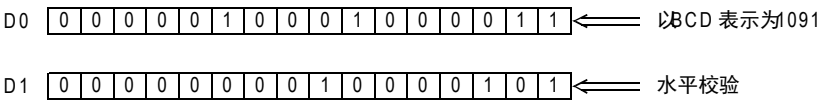


16位转换的例子

当为下述程序时，如下所示执行转换。

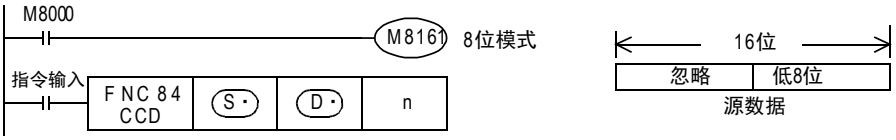


(S·)	数据内容的例子	
D100低	K100 = 01100100	
D100高	K111 = 0110111①	←
D101低	K100 = 01100100	
D101高	K98 = 01100010	
D102低	K123 = 0111101①	←
D102高	K66 = 01000010	
D103低	K100 = 01100100	
D103高	K95 = 0101111①	←
D104低	K210 = 11010010	
D104高	K88 = 01011000	
总和	K1091	
水平校验	1000010①	← 1的个数为奇数时水平校验为1 1的个数为偶数时水平校验为0



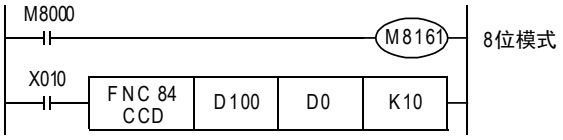
3. 《8位转换模式》M8161=ON时(M8161为RS、ASCI、HEX、CRC指令通用)

关于以(S・)开始的n点数据(低位仅为8位)，将其总和以及水平校验分别保存到(D・)和(D・)+1的软元件中。
M8161为RS、ASCI、HEX、CRC指令通用。使用8位转换模式时，请将M8161一直置为ON。
M8161在RUN→STOP时被清除。

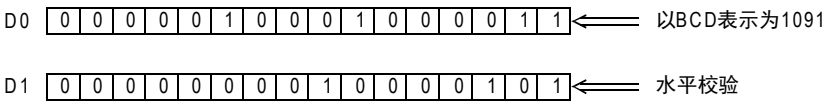


8位转换的例子

当为下述程序时，如下所示执行转换。



(S・)	数据内容的例子	
D100	K100 = 01100100	
D101	K111 = 0110111①	←
D102	K100 = 01100100	
D103	K98 = 01100010	
D104	K123 = 0111101①	←
D105	K66 = 01000010	
D106	K100 = 01100100	
D107	K95 = 0101111①	←
D108	K210 = 11010010	
D109	K88 = 01011000	
总和	K1091	
水平校验	1000010①	← 1的个数为奇数时水平校验为1 1的个数为偶数时水平校验为0



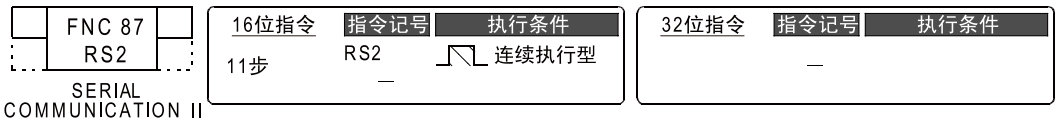
16.6 FNC 87 - RS2/串行数据传送2

A8 A8C

概要

该指令是用于通过安装在基本单元上的RS-232C或RS-485串行通信口进行无协议通信，从而执行数据的发送和接收的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	保存发送数据的数据寄存器的起始软元件	BIN16位/字符串
m	发送数据的字节数[设定范围: 0～4096]	BIN16位
(D·)	数据接收结束时，保存接收数据的数据寄存器的起始软元件	BIN16位/字符串
n	接收数据的字节数[设定范围: 0～4096]	BIN16位
n1	使用通道编号[設定内容: K0: 通道0、K1: 通道1、K2: 通道2]	BIN16位

3. 对象软元件

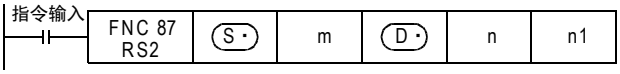
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●				●					
m														●	●					●	●			
(D·)														●	●				●					
n														●	●					●	●			
n1																				●	●			

功能和动作说明

1. 16位运算(RS2)

该指令是用于通过安装在基本单元上的RS-232C或RS-485串行通信口进行无协议通信，从而执行数据的发送和接收的指令。

→ 关于详细说明，请参考通信控制手册



相关软元件

→ 有关详细说明，请参考通信控制手册

软元件			名称
通道0	通道1	通道2	
M8371	M8401	M8421	发送等待标志位*1
M8372	M8402	M8422	发送请求*1
M8373	M8403	M8423	接收结束标志位*1
—	M8404	M8424	载波的检测标志位
—	M8405	M8425	数据设定准备就绪(DSR)标志位*2
—	—	—	—
M8379	M8409	M8429	超时的判断标志位
—	—	—	—
M8062	M8063	M8438	串行通信错误*3

软元件			名称
通道0	通道1	通道2	
M8370	D8400	D8420	设定通信格式
—	—	—	—
M8372	D8402	D8422	发送数据的剩余点数*1
M8373	D8403	D8423	接收点数的监控*1
M8375	D8405	D8425	通信参数的显示
M8379	D8409	D8429	设定超时时间
M8380	D8410	D8430	报头1、2
M8381	D8411	D8431	报头3、4
M8382	D8412	D8432	报尾1、2
M8383	D8413	D8433	报尾3、4
M8384	D8414	D8434	接收和校验(接收数据)
M8385	D8415	D8435	接收和校验(计算结果)
M8386	D8416	D8436	发送和校验
M8389	D8419	D8439	动作模式的显示
M8062	D8063	D8438	串行通信错误的错误代码编号*3

*1. 从RUN→STOP时清除

*2. A8/A8C可编程控制器需要Ver2.30以上的版本才能对应。

*3. 电源从OFF变为ON时清除

系统配置

为了使用这个指令，需要在基本单元中安装下表中的任一产品。

→ 有关系统配置，请参考所使用的可编程控制器主机的硬件篇手册

→ 有关详细说明，请参考通信控制手册

可编程控制器	通信的种类	选件
A8, A8C-32MT-LT(-2)	RS-232C通信	A8-232-BD或A8-485ADP(-MB)
	RS-485通信	A8-485-BD或A8-485ADP(-MB)
A8C(D,DSS)	RS-232C通信	A8-232ADP(-MB)
	RS-485通信	A8-485ADP(-MB)

关于RS(FNC 80)指令和RS2(FNC 87)指令的区别

项目	RS2指令	RS指令	备注
报头点数	1～4个字符(字节)	最大1个字符(字节)	用RS2指令，报头或报尾中最多可以指定4个字符(字节)。
报尾点数	1～4个字符(字节)	最大1个字符(字节)	
和校验的附加	可以自动附加	请用用户程序对应。	用RS2指令，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

→ 关于其他的注意事项，请参考通信控制手册

- RS(FNC 80)和RS2(FNC 87)指令，请勿同时驱动针对同一个通信口的多个这种指令。
- 不可以对同一个通信口使用RS(FNC 80)、RS2(FNC 87)指令和IVCK(FNC 270)、IVDR(FNC 271)、IVRD(FNC 272)、IVWR(FNC 273)、IVBWR(FNC 274)指令。
- 使用报头、报尾时，RS2指令驱动之前请在相应的特殊D中设定报头、报尾数据。另外，驱动RS2命令时请不要更改报头、报尾的值。

根据版本不同的功能变更

对应版本		项目	功能概要
A8	A8C		
Ver.2.30 以上	Ver.2.30 以上	通道1 数据设定准备就绪(DSR)标志位	通道 1 的 DR(DSR) 信号为 ON 时，特殊软元件 M8405置ON
		通道2 数据设定准备就绪(DSR)标志位	通道 2 的 DR(DSR) 信号为 ON 时，特殊软元件 M8425置ON。

16.7 FNC 88 - PID/PID运算

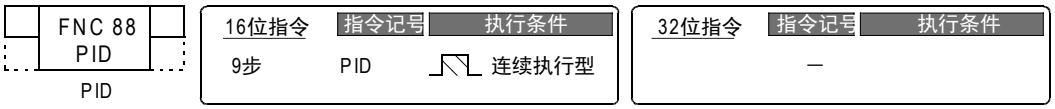
A1 A2 A2C A8 A8C

概要

该指令用于执行根据输入的变化量而改变输出值的PID控制。

→ 详细内容，请参考模拟量控制手册

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1)	保存目标值(SV)的数据寄存器编号	BIN16位
(S2)	保存测量值(PV)的数据寄存器编号	BIN16位
(S3)	保存参数的数据寄存器编号	BIN16位
(D)	保存输出值(MV)的数据寄存器编号	BIN16位

3. 对象软元件

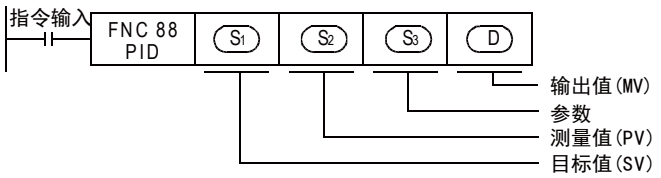
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1)														●	●	▲								
(S2)														●	●	▲								
(S3)														●	●									
(D)														●	●	▲								

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(PID)

执行对目标值(S1)、测量值(S2)、参数(S3)～(S3)+6进行设定的程序后，每隔采样时间(S3)将运算结果(MV)保存到输出值(D)中。



2. 设定项目

设定项目		内容	占用点数
(S1)	目标值(SV)	• 设定目标值(SV)。 • PID指令不更改设定内容。 • 使用自整定(极限循环法)时的注意事项 当自整定用的目标值与执行PID控制时的目标值不同时，需要设定加上了偏差值后的数值，在自整定标志位变为OFF时，保存实际的目标值。	1点
(S2)	测量值(PV)	PID运算的输入值。	1点
(S3)	参数*1	1) 自整定: 极限循环法的情况下 占用从 (S3) 中指定的起始软元件开始的29点软元件。 2) 自整定: 阶跃响应法的情况下 a) 动作设定(ACT)的设定: bit1、bit2、bit5全部为“0”以外数字时占用从 (S3) 中指定的起始软元件开始的25点软元件。 b) 动作设定(ACT)的设定: bit1、bit2、bit5全部为“0”时占用从 (S3) 中指定的起始软元件开始的20点软元件。	29点 25点 20点
(D)	输出值(MV)	1) PID控制时(普通处理的时候) 指令驱动之前在用户一侧设置初始输出值。 此后的运算结果将被保存。 2) 自整定: 极限循环法的情况下 在自整定过程中，自动地输出ULV值或LLV值，当自整定结束后，既定的MV值会被设定。 3) 自整定: 阶跃响应法的情况下 指令驱动之前请在用户一侧设置步输出值。 在自整定过程中，不能在PID指令一侧更改MV输出。	1点

*1. 不使用自整定时，与使用步响应法时占用相同的点数。

3. 参数 (S3) ～ (S3) +28的一览

设定项目			设定内容	备注
(S3)	采样时间(TS)		1～32767〔ms〕	比运算周期短的值无法执行。
(S3) +1	动作设定(ACT)	bit0	0: 正动作 1: 逆动作	动作方向
		bit1	0: 无输入变化量报警 1: 输入变化量报警有效	
		bit2	0: 无输出变化量报警 1: 输出变化量报警有效	bit2和bit5请勿同时置ON。
		bit3	不可以使用	
		bit4	0: 自整定不动作 1: 执行自整定	
		bit5	0: 无输出值上下限设定 1: 输出值上下限设定有效	bit2和bit5请勿同时置ON。
		bit6	0: 阶跃响应法 1: 极限循环法	选择自整定的模式
		bit7～bit15	不可以使用	
(S3) +2	输入滤波常数(α)		0～99〔%〕	0时表示无输入滤波
(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(无积分)
(S3) +5	微分增益(KD)		0～100〔%〕	0时无微分增益
(S3) +6	微分时间(TD)		0～32767〔×10ms〕	0时无积分
(S3) +7 : (S3) +19	被PID运算的内部处理占用。请不要更改数据。			
(S3) +20*2	输入变化量(增加侧)报警设定值		0～32767	动作方向(ACT): (S3) +1 bit1=1时有效
(S3) +21*1	输入变化量(减少侧)报警设定值		0～32767	动作方向(ACT): (S3) +1 bit1=1时有效

*1. 当 (S3) +1动作设定(ACT)的bit1=1、bit2=1或是bit5=1时， (S3) +20～+24被占用。

设定项目			设定内容	备注
(S3)+22*1	输出变化量(增加侧)报警设定值		0～32767	动作方向(ACT): (S3)+1,bit2=1,bit5=0 时有效
	输出上限的设定值		-32768～32767	动作方向(ACT): (S3)+1,bit2=0,bit5=1 时有效
(S3)+23*1	输出变化量(减少侧)报警设定值		0～32767	动作方向(ACT): (S3)+1,bit2=1,bit5=0 时有效
	输出下限的设定值		-32768～32767	动作方向(ACT): (S3)+1,bit2=0,bit5=1 时有效
(S3)+24*1	报警输出	bit0	0: 输入变化量(增加侧)未溢出 1: 输入变化量(增加侧)溢出	动作方向(ACT): (S3)+1 bit1=1 或是bit2=1时有效
		bit1	0: 输入变化量(减少侧)未溢出 1: 输入变化量(减少侧)溢出	
		bit2	0: 输出变化量(增加侧)未溢出 1: 输出变化量(增加侧)溢出	
		bit3	0: 输出变化量(减少侧)未溢出 1: 输出变化量(减少侧)溢出	
使用极限循环法是需要以下的设定(动作方向(ACT)b6: ON 时)				
(S3)+25	PV值临界值 (滞后)宽度(SHPV)		根据测量值(PV)的波动而设定	动作设定(ACT)b6: 选择极限循环法(ON)时占用
(S3)+26	输出值上限(ULV)		输出值(MV)的最大输出值 (ULV)设定	
(S3)+27	输出值下限(LLV)		输出值(MV)的最小输出值 (LLV)设定	
(S3)+28	从自整定循环结束到PID控制开始为止的等待设定参数(Kw)		-50～32717%	

*1. 当(S3)+1动作设定(ACT)的bit1=1、bit2=1或是bit5=1时，(S3)+20～+24被占用。

注意要点

1. 使用多个指令时

可以同时多次执行(环路数没有限制)，但是请注意运算中使用的(S3)和(D)的软元件编号不能重复。

2. 参数 (S3) 的占用点数

1) 极限循环法的情况

— 占用从(S3)中指定的起始软元件开始的29点软元件。

2) 阶跃响应法的情况

— 动作设定(ACT)的设定: bit1、bit2、bit5全部为“0”以外数字时占用从(S3)中指定的起始软元件开始的25点软元件。

— 动作设定(ACT)的设定: bit1、bit2、bit5全部为“0”时占用从(S3)中指定的起始软元件开始的20点软元件。

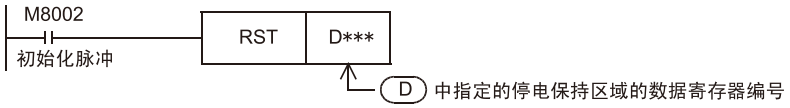
3. 指定停电保持区域的软元件时

请对PID指令的输出值(MV)，指定停电保持区域除外的数据寄存器(D)。

程序举例

指定了停电保持区域的数据寄存器时，请务必采用下面的程序在可编程控制器运行时，清除备份的内容。

程序实例



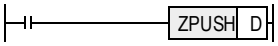
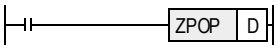
错误

发生运算错误后，特殊辅助继电器M8067为ON，错误代码被保存在特殊数据寄存器D8067中。

→ 错误代码参考37.4节

17. 数据传送2—FNC 100～FNC 109

FNC 100～FNC 109中，提供了以特殊处理为目的，执行比基本应用指令处理更加复杂的指令。

FNC No.	指令记号	符号	功能	参考
100	—			—
101	—			—
102	ZPUSH		变址寄存器的成批保存	17.1节
103	ZPOP		变址寄存器的恢复	17.2节
104	—			—
105	—			—
106	—			—
107	—			—
108	—			—
109	—			—

17.1 FNC 102 - ZPUSH/变址寄存器的成批保存

A8 A8C

概要

该指令是暂时保存变址寄存器V0～V7、Z0～Z7的当前值的指令。
要使暂时保存的当前值返回时，使用ZPOP(FNC 103)指令。

→ 有关ZPOP(FNC 103)指令，请参考17.2节

1. 指令格式

FNC 102 ZPUSH P
INDEX REGISTER PUSH

16位指令	指令记号	执行条件
3步	ZPUSH	连续执行型
	ZPUSHP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

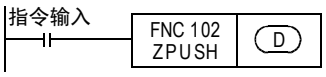
操作数种类	内容	数据类型
(D)	暂时保存变址寄存器V0～V7、Z0～Z7的当前值的软元件起始编号 (D) ：成批保存次数 (D)+1～(D)+16×成批保存次数: 成批保存的数据保存的位置	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D)														●	●									

功能和动作说明

1. 16位运算(ZPUSH/ZPUSHP)

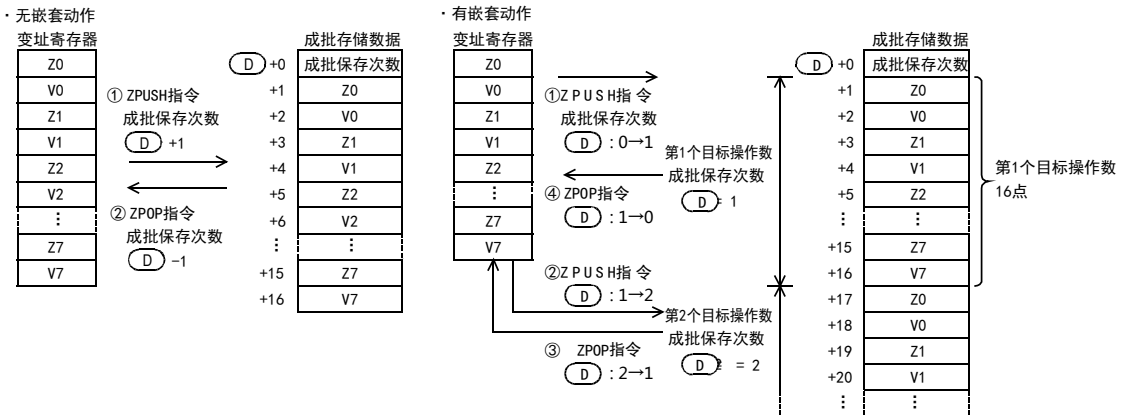


1) 将变址寄存器 Z0 ～ Z7、V0 ～ V7 的内容成批保存到(D)开始的软元件中。成批保存了变址寄存器的内容后，成批保存次数(D)就被+1。

2) 使用ZPOP(FNC 103)指令使数据返回。
成对使用ZPUSH(FNC 102)、ZPOP(FNC 103)指令。

3) 通过对(D)指定相同的软元件，可以嵌套使用ZPUSH(FNC 102)～ZPOP(FNC 103)指令。
此时，每次执行ZPUSH(FNC 102)指令时，(D)开始使用的区域会每次增加16点。因此，请事先确保嵌套中使用的次数的区域。

4) (D)以后被成批保存的数据的结构如下所示。



相关指令

指令	内容
ZPOP(FNC 103)	使因ZPUSH(FNC 102)指令而暂时成批保存的变址寄存器V0～V7、Z0～Z7恢复的指令。

注意要点

- 无嵌套动作时，执行ZPUSH(FNC 102)指令前请清除成批保存次数 (D) 。
- 有嵌套动作时，请在第一次执行前清除成批保存次数 (D) 。

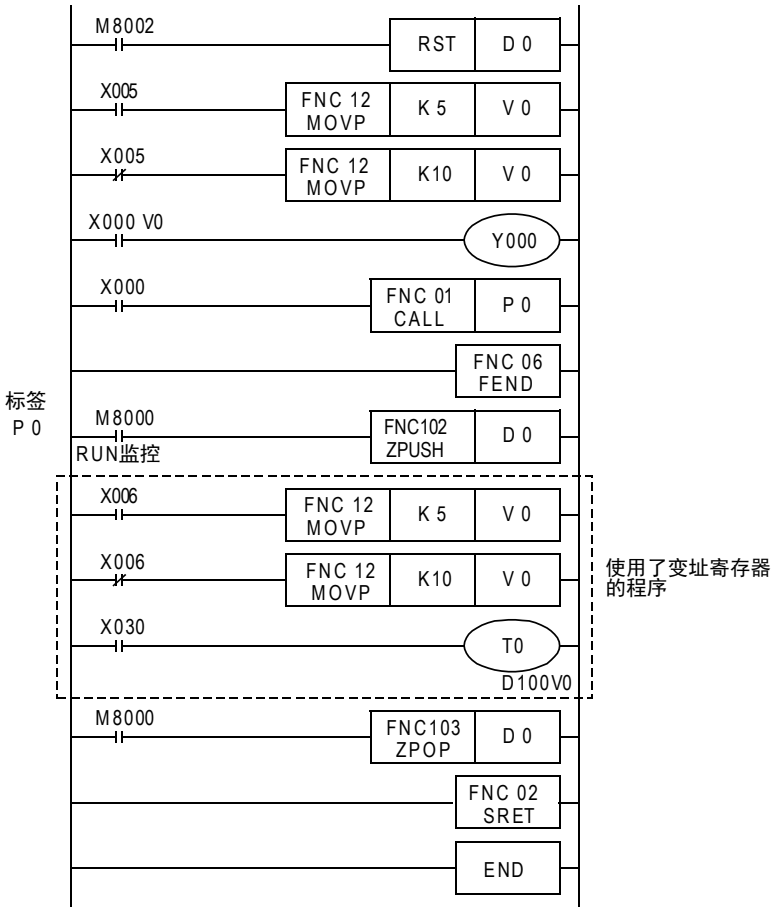
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- ZPUSH(FNC 102)指令中，当 (D) 开始使用的点数范围超出相应的软元件的范围时。
  (错误代码: K6706)
- 执行ZPUSH(FNC 102)指令时， (D) (成批保存次数)为负时(错误代码: K6707)

程序举例

下面例举了在指针P0以后的子程序中使用了变址寄存器时，在执行子程序之前，先将变址寄存器Z0～Z7、V0～V7的内容成批保存到D0以后的程序。



17.2 FNC 103 - ZPOP/变址寄存器的恢复

A8 A8C

概要

该指令是将使用ZPUSH(FNC 102)指令进行暂时成批保存的变址寄存器V0～V7、Z0～Z7中的内容恢复回去的指令。

→ 有关ZPUSH(FNC 102)指令，请参考17.1节

1. 指令格式



16位指令	指令记号	执行条件
3步	ZPOP	连续执行型
	ZPOPP	脉冲执行形

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(D)	暂时成批保存变址寄存器V0～V7、Z0～Z7内容的软元件的起始编号 (D) : 成批保存次数 (D)+1～(D)+16×成批保存次数: 成批保存的数据保存位置	BIN16位

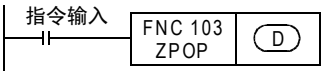
3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址				常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰		K	H	E	"□"	P
(D)														●	●										

功能和动作说明

1. 16位运算(ZPOP/ZPOPP)

→ 有关功能及动作也请参考17.1节



1) 将使用ZPUSH(FNC 102)指令被暂时成批保存到(D)开始的软元件中的变址寄存器V0～V7、Z0～Z7的内容恢复到原来的变址寄存器中。变址寄存器的内容恢复后，成批保存次数(D)被-1。

2) 使用ZPUSH(FNC 102)指令，暂时成批保存数据。
ZPUSH(FNC 102)、ZPOP(FNC 103)指令都成对使用。

相关指令

指令	内容
ZPUSH(FNC 102)	暂时成批保存变址寄存器V0～V7、Z0～Z7的当前值的指令。

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 执行ZPOP(FNC 103)指令时，成批保存次数(D)的内容为0或是负数时。(错误代码: K6706)

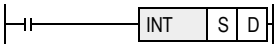
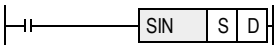
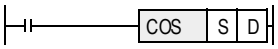
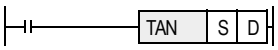
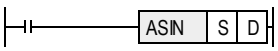
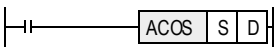
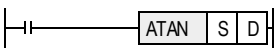
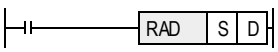
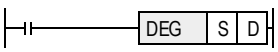
程序举例

→ 有关程序举例，请参考17.1节

18. 浮点数运算—FNC 110～FNC 139

FNC 110～FNC 119、FNC 120～FNC 129、FNC 130～FNC 139中，提供了用于浮点数的转换、比较、四则运算、开方运算、三角函数等指令。

FNC No.	指令记号	符号	功能	参考
110	ECMP	ECMP S1 S2 D	2进制浮点数比较	18.1节
111	EZCP	EZCP S1 S2 S D	2进制浮点数区间比较	18.2节
112	EMOV	EMOV S D	2进制浮点数数据传送	18.3节
113	—			—
114	—			—
115	—			—
116	ESTR	ESTR S1 S2 D	2进制浮点数→字符串的转换	18.4节
117	EVAL	EVAL S D	字符串→2进制浮点数的转换	18.5节
118	EBCD	EBCD S D	2进制浮点数→10进制浮点数的转换	18.6节
119	EBIN	EBIN S D	10进制浮点数→2进制浮点数的转换	18.7节
120	EADD	EADD S1 S2 D	2进制浮点数加法运算	18.8节
121	ESUB	ESUB S1 S2 D	2进制浮点数减法运算	18.9节
122	EMUL	EMUL S1 S2 D	2进制浮点数乘法运算	18.10节
123	EDIV	EDIV S1 S2 D	2进制浮点数除法运算	18.11节
124	EXP	EXP S D	2进制浮点数指数运算	18.12节
125	LOGE	LOGE S D	2进制浮点数自然对数运算	18.13节
126	LOG10	LOG10 S D	2进制浮点数常用对数运算	18.14节
127	ESQR	ESQR S D	2进制浮点数开方运算	18.15节
128	ENEG	ENEG D	2进制浮点数符号翻转	18.16节

FNC No.	指令记号	符号	功能	参考
129	INT		2进制浮点数→BIN整数的转换	18.17节
130	SIN		2进制浮点数SIN运算	18.18节
131	COS		2进制浮点数COS运算	18.19节
132	TAN		2进制浮点数TAN运算	18.20节
133	ASIN		2进制浮点数SIN^{-1}运算	18.21节
134	ACOS		2进制浮点数COS^{-1}运算	18.22节
135	ATAN		2进制浮点数TAN^{-1}运算	18.23节
136	RAD		2进制浮点数角度→弧度的转换	18.24节
137	DEG		2进制浮点数弧度→角度的转换	18.25节
138	—			—
139	—			—

18.1 FNC 110 - ECMP/2进制浮点数比较

A8 A8C

概要

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

→ 浮点数的使用，请参考5.1.3项

1. 指令格式

FNC 110 ECMP (D: P)
EXTENDED COMPARE

16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
13步	DECMP	连续执行型
	DECMPP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	保存要比较的2进制浮点数数据的软元件编号	实数(2进制)*1
(S2·)	保存要比较的2进制浮点数数据的软元件编号	
(D·)	输出结果的起始位软元件编号(占用3点)	位

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲2			●	●	●	●		
(S2·)														●	●	▲2			●	●	●	●		
(D·)		●	●			●	▲1												●					

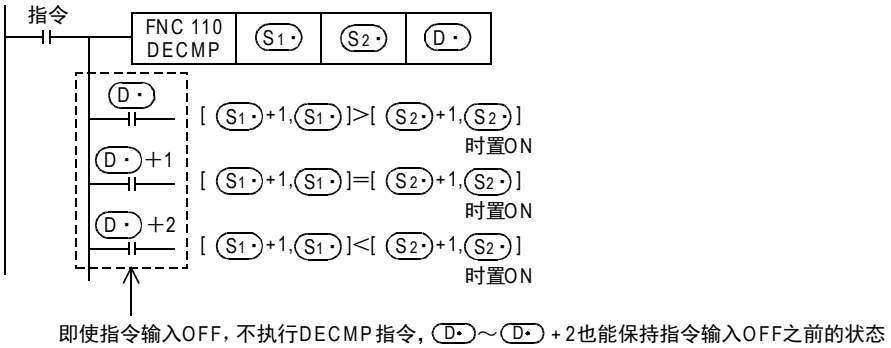
▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能进行变址修饰(V、Z)
▲2: 只对应A8/A8C可编程控制器。

功能和动作说明

1. 32位运算(DECMP、DECMPP)

将比较值[(S1·)+1，(S1·)]和比较源[(S2·)+1，(S2·)]作为浮点数数据进行比较，然后根据比较的结果(小于、等于、大于)将(D·)、(D·)+1、(D·)+2中的任意一位置ON。

- [(S1·)+1，(S1·)]、[(S2·)+1，(S2·)]中指定了常数(K、H)时，会自动将数值从BIN转换成2进制浮点数后再处理。



注意要点

1. 软元件的占用点数

(D·)占用3点。[(D·)，(D·)+1，(D·)+2]
请注意不要与用于其他用途的软元件重复。

18.2 FNC 111 - EZCP/2进制浮点数区间比较

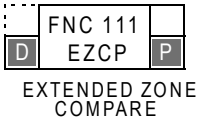
A8 A8C

概要

将上下2点的比较范围和数据(2进制浮点数)进行比较，根据其结果输出到位软元件(3点)中的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
17步	DEZCP	连续执行型
	DEZCPP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存要比较的2进制浮点数数据的软元件编号	实数(2进制)*1
(S2・)	保存要比较的2进制浮点数数据的软元件编号	
(S・)	保存要比较的2进制浮点数数据的软元件编号	
(D・)	输出结果的起始位软元件编号(占用3点)	位

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)														●	●	●			●	●	●	●		
(S2・)														●	●	●			●	●	●	●		
(S・)														●	●	●			●	●	●	●		
(D・)		●	●			●	▲												●					

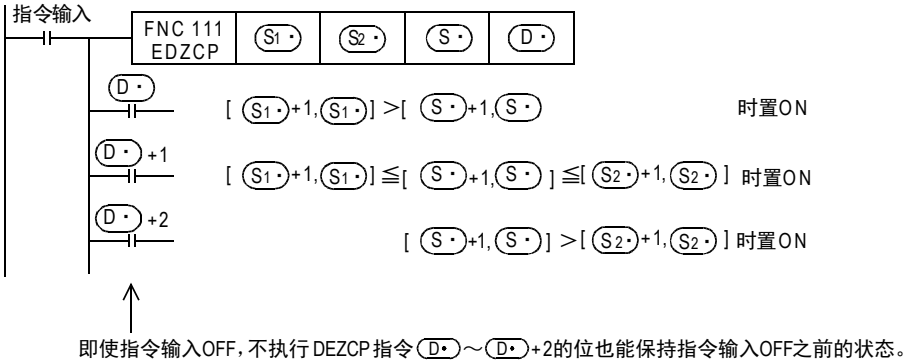
▲: D□.b不能变址修饰(V、Z)。

功能和动作说明

1. 32位运算(DEZCP、DEZCPP)

将比较值[(S1・)+1，(S1・)]、[(S2・)+1，(S2・)]和比较源[(S・)+1，(S・)]作为浮点数数据进行比较，然后根据结果(小于、等于或大于)将(D・)、(D・)+1、(D・)+2中的任意一位置ON。

- 在[(S1・)+1，(S1・)]、[(S2・)+1，(S2・)]、[(S・)+1，(S・)]中指定了常数 (K、H)时，会自动将数值转换成2进制浮点数后再处理。



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

注意要点

1. 软元件的占用点数
   (D•)的软元件占用3点。
   请注意不要与用于其他用途的软元件重复。

2. 关于 (S1•) 和 (S2•) 的比较数据
   比较数据的大小关系请设置为[(S1•)+1，(S1•)]≦[(S2•)+1，(S2•)]。
   [(S1•)+1，(S1•)]＞[(S2•)+1，(S2•)]的情况时，[(S2•)+1，(S2•)]的数值视为与[(S1•)+1，(S1•)]相同，从而进行比较。

18.3 FNC 112 - EMOV/2进制浮点数数据传送

A8 A8C

概要

传送2进制浮点数数据的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DEMOV	连续执行型
	DEMOVP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(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	实数(2进制)
(D・)	保存2进制浮点数数据的软元件编号	

3. 对象软元件

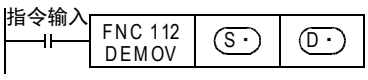
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	▲			●			●		
(D・)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

功能和动作说明

1. 32位运算(DEMOV/DEMOVP)

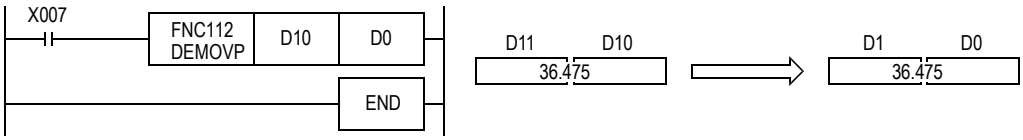
将传送源[(S・)+1，(S・)]的内容(2进制浮点数数据)传送到[(D・)+1，(D・)]中。此外，还可以在(S・)中直接指定实数(E)。



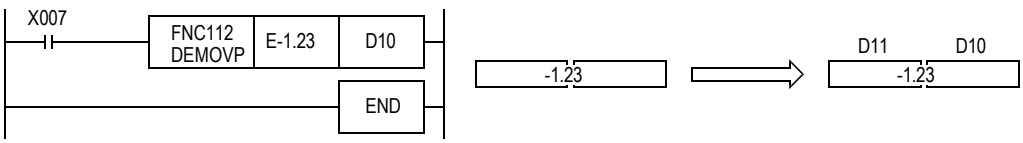


程序举例

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D11、D10中的程序。



18.4 FNC 116 - ESTR/2进制浮点数→字符串的转换

A8 A8C

概要

该指令用于将2进制浮点数数据转换成指定位数的字符串(ASCII码)。
还有可以将BIN数据转换成字符串(ASCII码)的STR(FNC 200)指令。

→ 关于字符串，请参考5.3节
→ 浮点数的使用，请参考5.1.3项
→ 有关STR(FNC 200)指令，请参考26.1项

1. 指令格式

FNC 116 ESTR (D / P)
EXTENDED STRING

16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
13步	DESTR	连续执行型
	DESTRP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(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	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	BIN16位
(D•)	保存已转换的字符串的软元件起始编号	字符串

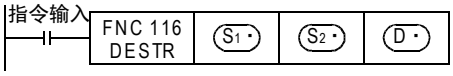
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)														●	●	●			●			●		
(S2•)								●	●	●	●	●	●	●	●	●			●					
(D•)									●	●	●	●	●	●	●	●			●					

功能和动作说明

1. 32位运算(DESTR/DESTRP)

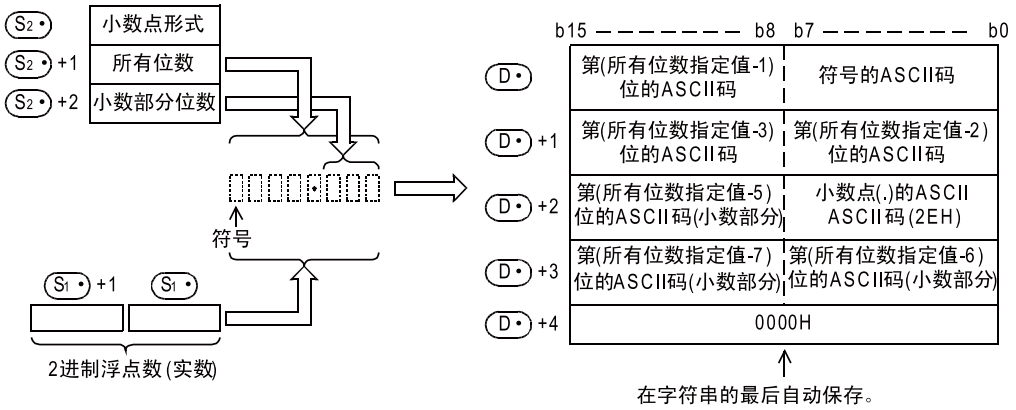
根据(S2•)，(S2•)+1，(S2•)+2中指定的内容，将[(S1•)+1，(S1•)]的内容(2进制浮点数数据)转换成字符串，并保存至(D•)开始的软元件中。此外，还可以在(S1•)中直接指定实数。



- (S2•)指定的显示指定会导致转换后的数据有所不同。

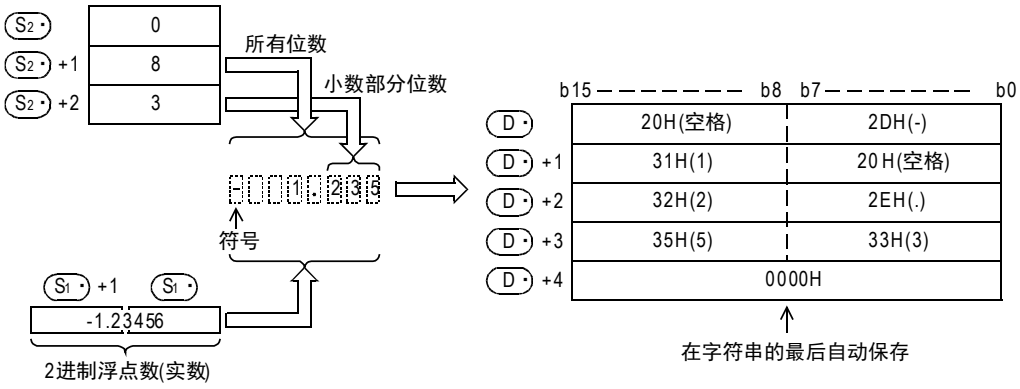
(S2・)	0: 小数点形式 1: 指数形式	(S2・)的形式不同相应的转换后的数据也不同
(S2・)+1	所有位数	可以在2～24之间设定。
(S2・)+2	小数部分位数	

2. 小数点形式时

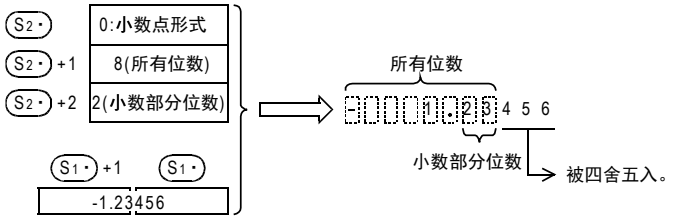


- 在 (S2・)+1中可以指定的所有位数如下。(最大: 24 位数)
  小数部分的位数为“0”时・・・・・・・・・・所有位数≧2
  小数部分的位数为“0”以外的数字时・・・・所有位数≧(小数部分位数+3)
- (S2・)+2中可以指定的小数部分位数为0～7位数。
  但是，请设定为小数部分位数≦(所有位数-3)。

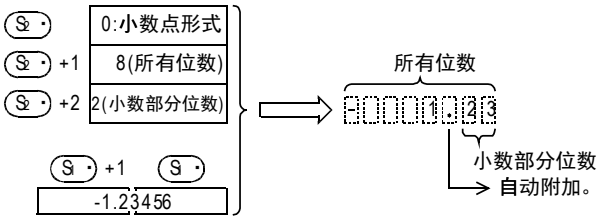
例如，所有位数为8，小数部分位数为3，指定 -1.23456时，(D・)开始的软元件中会如下所示被保存。



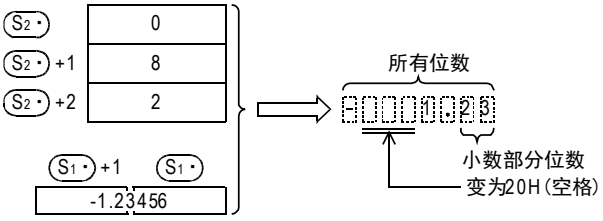
- 转换后的字符串数据，如下所示被保存在 (D・)开始的软元件中。
  －字符中，2进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。
  －小数部分位数的范围中不能容纳2进制浮点数数据的小数部分时，小数低位部分被四舍五入。



－ 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"(.)。

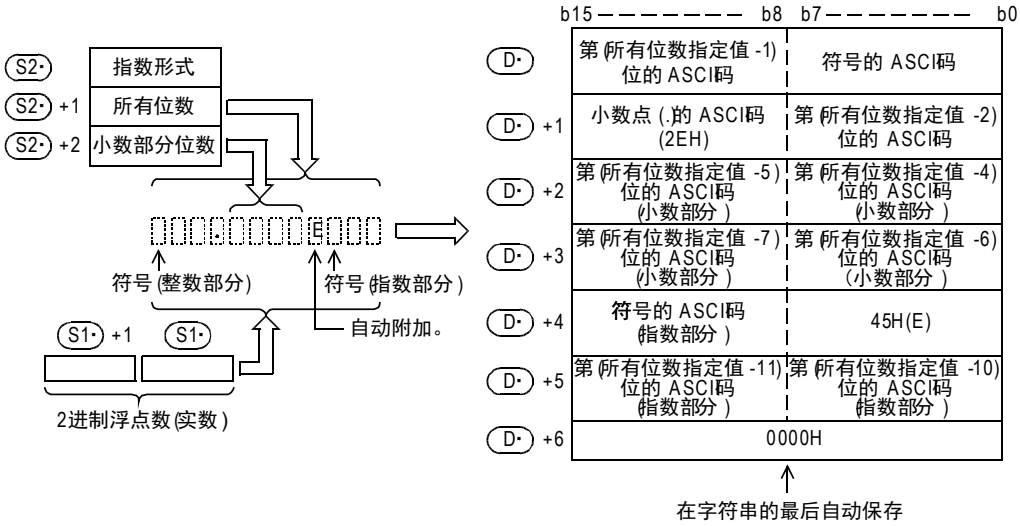


－ 当从所有位数中去除符号、小数点、小数部分的位数后，其位数比 2进制浮点数数据的整数部分大的时候，在符号和整数部分之间保存“ 20H”(空格)。



－ 在转换的字符串最后，自动保存“00H”或“0000H”。

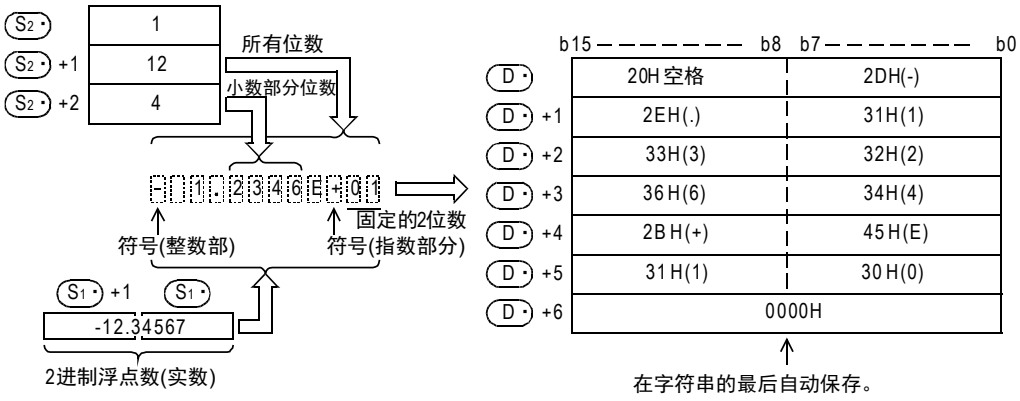
3. 指数形式时



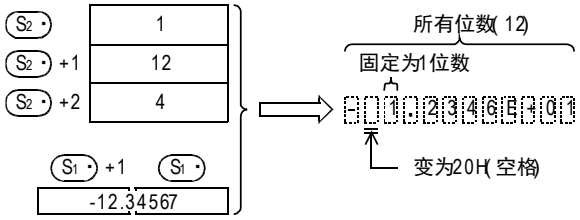
- 在(S2・)+1中可以指定的所有位数如下。(最大:24位数)
小数部分的位数为“0”时・・・・・・・・・位数≧6
小数部分的位数为“0”以外的数字时・・・・位数≧(小数部分位数+7)

- (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-7)。

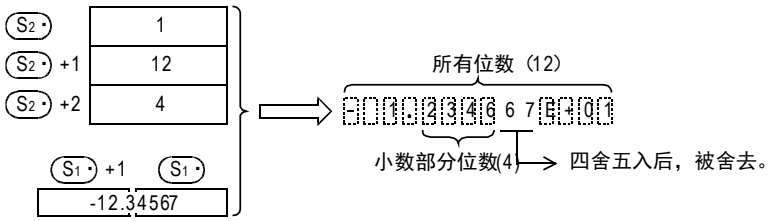
例如，所有位数为12，小数部分位数为4，指定-12.34567时，在(D・)开始的软元件中会如下所示被保存。



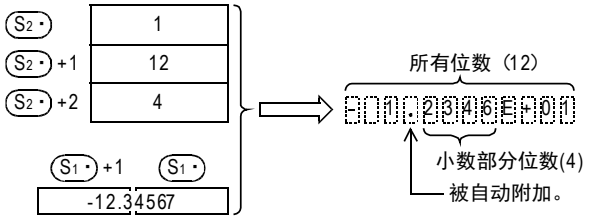
- 换后的字符串数据，如下所示被保存在(D・)开始的软元件中。
  － 在整数部分的符号中，2进制浮点数数据为正时保存“20H”(空格)，为负时保存“2DH”(-)。
  － 整数部分固定为1位数。
  在整数部分和符号之间保存“20H”(空格)。



  － 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入。

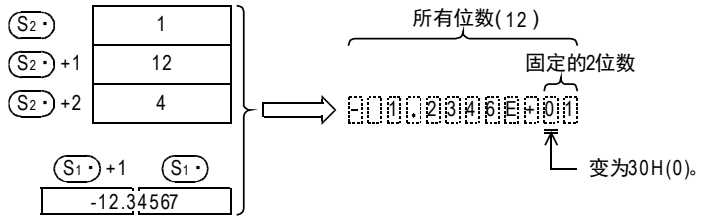


  － 小数部分位数设定为“0”以外数字时，会自动将“2EH”(.)保存到指定的小数部分位数+1的位中。
  但是，小数部分位数为“0”时，不保存“2EH”(.)。



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

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



－ 在转换的字符串最后，自动保存“00H”或“0000H”。

相关指令

指令	内容
EVAL(FNC 117)	将字符串(ASCII码)数据转换成2进制浮点数数据的指令。
STR(FNC 200)	将BIN数据转换成字符串(ASCII码)的指令。
VAL(FNC 201)	将字符串(ASCII码)数据转换成BIN数据的指令。

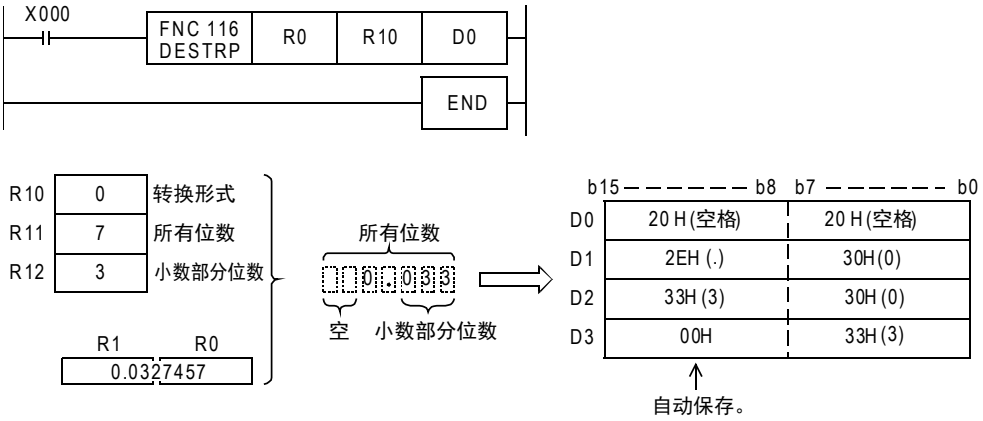
错误

以下一些情况下会出现运算错误，错误标志位(M8067)置ON，错误代码保存在D8067中。

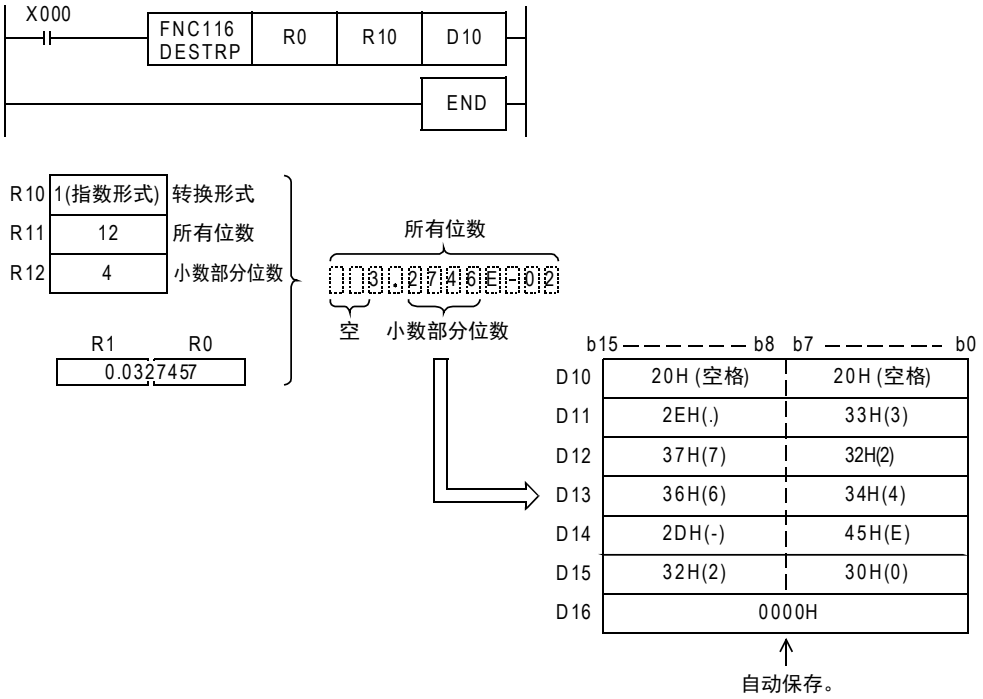
- (S1・)不在下列范围中时。(错误代码: K6706)
  0, $\pm 2^{-126} \leqq$ (S1・) $< \pm 2^{128}$
- (S2・)中指定的形式为0、1以外时。(错误代码: K6706)
- (S2・)+1中指定的所有位数指定不符合下面的范围时。(错误代码: K6706)
  小数点形式
  小数部分的位数为“0”时・・・・・・・・・所有位数≧2
  小数部分的位数为“0”以外的数字时・・・・所有位数≧(小数部分位数+3)
  指数形式
  小数部分的位数为“0”时・・・・・・・・・所有位数≧6
  小数部分的位数为“0”以外的数字时・・・・所有位数≧(小数部分位数+7)
- (S2・)+2中指定的小数部分的位数指定不符合下面范围时。(错误代码: K6706)
  小数点形式时: 小数点位数≦(所有位数-3)
  指数形式时: 小数点位数≦(所有位数-7)
- 当保存指定字符串的软元件(D・)的范围，超出相应软元件的范围时。
  (错误代码: K6706)
- 转换结果超出已指定的所有位数时。(错误代码: K6706)

程序举例

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，并且保存在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，并且保存在D10以后的软元件中的程序。



18.5 FNC 117 - EVAL/字符串→2进制浮点数的转换

A8 A8C

概要

用于将字符串(ASCII码)转换成2进制浮点数数据的指令。
还有可以将字符串(ASCII码)转换成BIN数据的VAL(FNC 201)指令。

→ 关于字符串，请参考5.3节
→ 浮点数的使用，请参考5.1.3项
→ 有关VAL(FNC 201)指令，请参考26.2节

1. 指令格式

FNC 117 EVAL (D: P)
EXTENDED VALUE

16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DEVAL	连续执行型
	DEVALP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	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	字符串
(D·)	保存已转换的2进制浮点数数据的软元件的起始编号	实数(2进制)

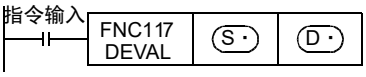
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●			●					
(D·)														●	●	●			●					

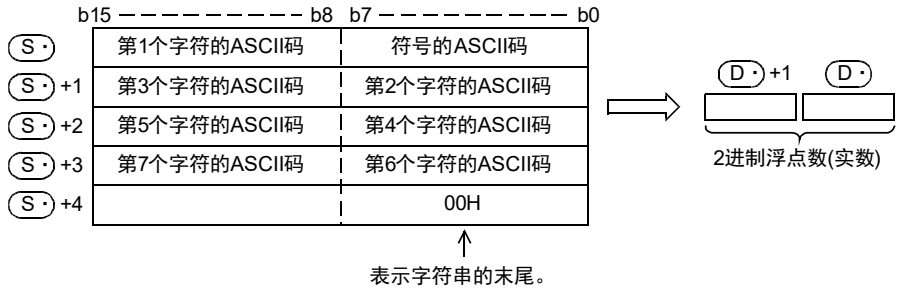
功能和动作说明

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进制浮点数数据后，保存到[(D·)+1，(D·)]中。

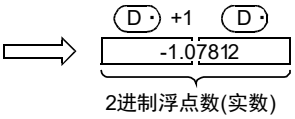


指定的字符串无论是小数点形式，还是指数形式，都可以转换成2进制浮点数数据。



a) 小数点形式时

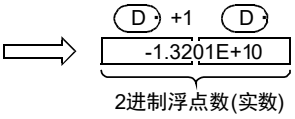
	b15 ------ b8	b7 ------ b0
S・	31H(1)	2DH(-)
S・+1	30H(0)	2EH(.)
S・+2	38H(8)	37H(7)
S・+3	32H(2)	31H(1)
S・+4		00H



-1.07812

b) 指数形式时

	b15 ------ b8	b7 ------ b0
S・	20H(空格)	2DH(-)
S・+1	2EH(.)	31H(1)
S・+2	32H(2)	33H(3)
S・+3	31H(1)	30H(0)
S・+4	2BH(+)	45H(E)
S・+5	30H(0)	31H(1)
S・+6		00H

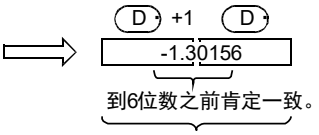


-1.3201E+10

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

a) 小数点形式时

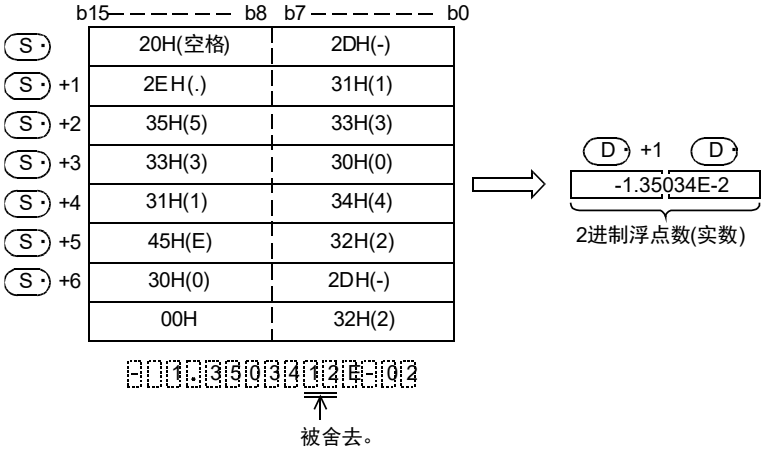
	b15 ------ b8	b7 ------ b0
S・	20H(空格)	2DH(-)
S・+1	31H(1)	20H(空格)
S・+2	33H(3)	2EH(.)
S・+3	31H(1)	30H(0)
S・+4	36H(6)	35H(5)
S・+5	31H(1)	38H(8)
S・+6	00H	32H(2)



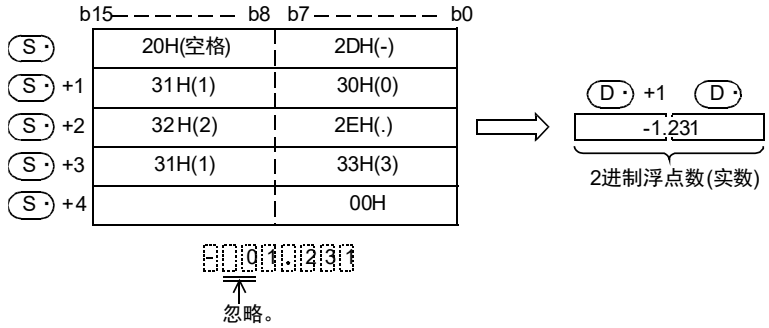
-1.301568 12

被舍去

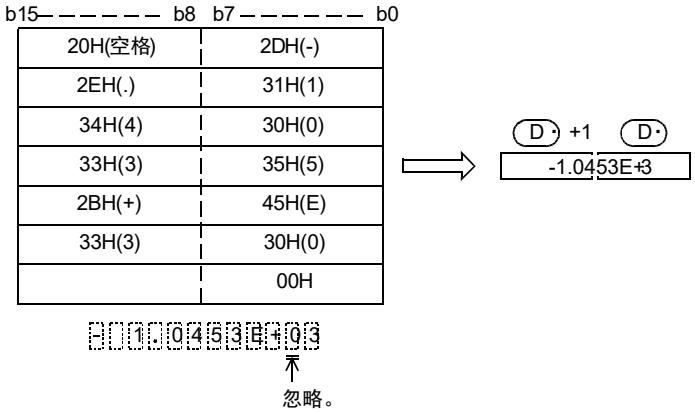
b) 指数形式时



- 在小数点形式中，将符号指定为“2BH”(+)或省略符号，则作为正值转换。
  此外，将符号指定为"2DH"(-)则作为负的值转换。
- 用指数形式在指数部分的符号内指定"2BH"(+)，省略符号后作为正的指数转换。
  将指数部分的符号指定为"2DH"(-)则作为负的指数转换。
- (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“30H”(0)时，会忽略“20H”、“30H”而进行转换。



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



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

相关软元件

→ 关于零位、借位、进位标志位的使用方法，请参考6.5.2项

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

相关指令

指令	内容
ESTR(FNC 116)	将2进制浮点数数据转换成字符串(ASCII码)的指令。
STR(FNC 200)	将BIN数据转换成字符串(ASCII码)的指令。
VAL(FNC 201)	将字符串(ASCII码)数据转换成BIN数据的指令。

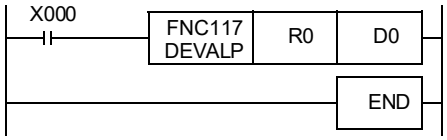
错误

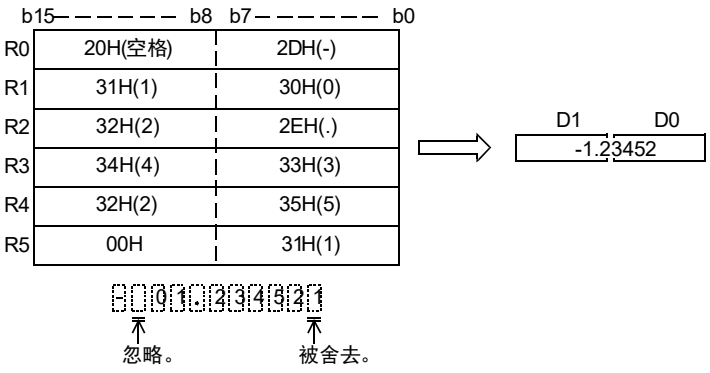
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 整数部分、小数部分中存在“30H”(0)～“39H”(9)以外的字符时。(错误代码: K6706)
- (S・)指定的字符串中存在2个以上的“2EH”(.)时。(错误代码: K6706)
- 指数部分中有“45H”(E)、“2BH”(+)、“2DH”(-)以外的字符或是有多个指数部分时。(错误代码: K6706)
- (S・)开始的相应软元件范围内没有“00H”时。(错误代码: K6706)
- (S・)以后的字符数为0或是超出了24个字符时。(错误代码: K6706)

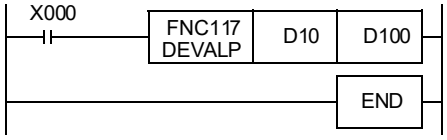
程序举例

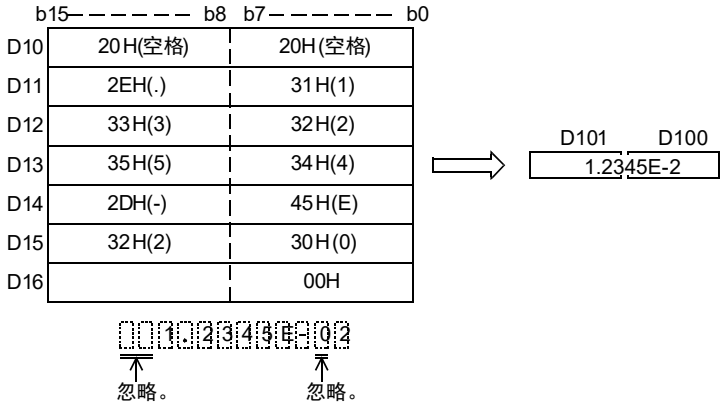
1) 当X000为ON时，将R0开始的软元件中保存的字符串转换成2进制浮点数，并保存到D0、D1中的程序。





2) 当X000为ON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≧2^{128}	(D•)的值大于32位实数的最大值(2^{128})部分被舍去，进位标志位(M8022)为ON。
转换结果真的为零 (尾数部分为“0”时)	零位标志位(M8020)为ON。

18.6 FNC 118 - EBCD/2进制浮点数→10进制浮点数的转换

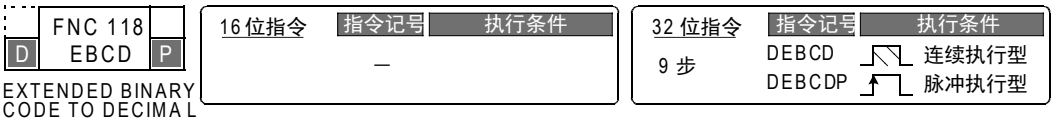
A8 A8C

概要

将软元件中的2进制浮点数转换成→10进制浮点数的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	保存2进制浮点数数据的数据寄存器编号	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	实数(10进制)

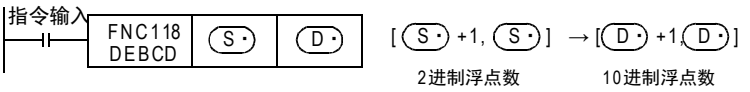
3. 对象软元件

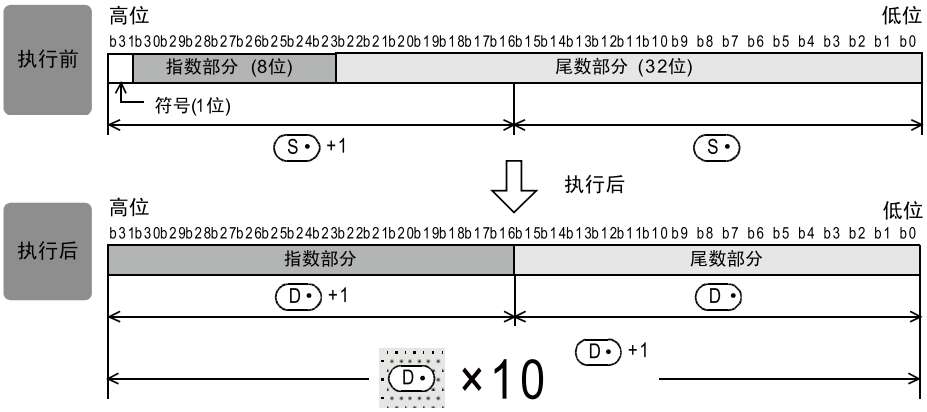
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●					
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DEBCD,DEBCDP)

将[(S・)+1、(S・)]的2进制浮点数转换成10进制浮点数后，传送到[(D・)+1，(D・)]中。





注意要点

1. 浮点数运算的处理

在浮点数运算中，都以2进制浮点数执行。

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

此外，在HCP-works和GOT中，具备直接监控和显示2进制浮点数的功能。

18.7 FNC 119 - EBIN/10进制浮点数→2进制浮点数的转换

A8 A8C

概要

将软元件中的10进制浮点数转换成2进制浮点数的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



2. 设定数据

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

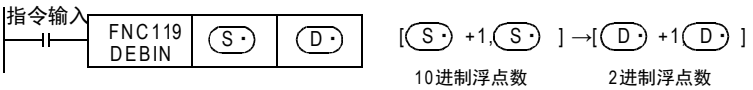
3. 对象软元件

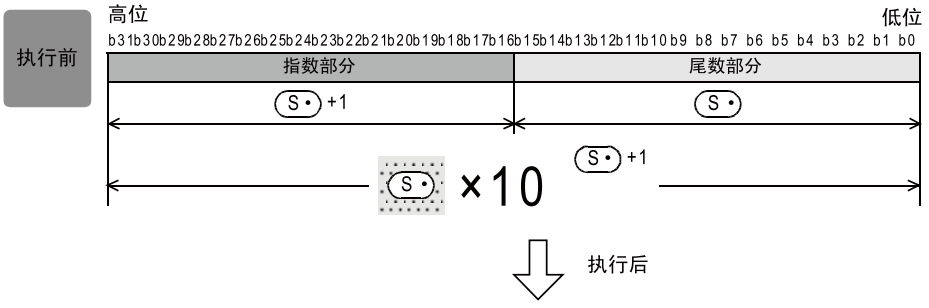
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●					
(D·)														●	●	●			●					

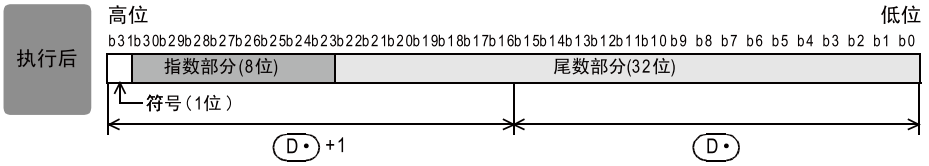
功能和动作说明

1. 32位运算(DEBIN、DEBINP)

将[(S·)+1，(S·)]的10进制浮点数转换成2进制浮点数后，传送到[(D·)+1，(D·)]中。





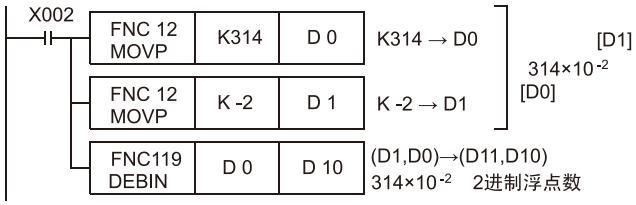


程序举例

使用DEBIN指令后，包含了小数点的数值等可以被直接转换为2进制浮点数。

例: 3.14的2进制浮点数转换

$3.14=314\times10^{-2}$(10进制浮点数)



→ 浮点数运算的程序举例，参考12.10节

18.8 FNC 120 - EADD/2进制浮点数加法运算

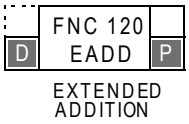
A8 A8C

概要

2个2进制浮点数加法运算的指令。

→ 浮点数运算的程序举例，参考12.10节
→ 浮点数的使用，请参考5.1.3项
→ 标志位的动作，请参考6.5.2项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
13步	DEADD	连续执行型
	DEADDP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	保存进行加法运算的2进制浮点数数据的字软元件编号	实数(2进制)*1
(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·)	保存加法运算后的2进制浮点数数据的数据寄存器编号	

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

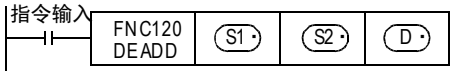
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●	●		
(S2·)														●	●	▲			●	●	●	●		
(D·)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

功能和动作说明

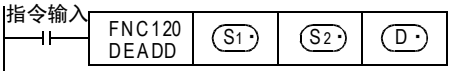
1. 32位运算(DEADD、DEADDP)

将[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]的2进制浮点数数据进行加法运算，并将其运算结果以2进制浮点数形式传送到[(D·)+1，(D·)]中。



[(S1·)+1(S1·)]+[(S2·)+1(S2·)]→[(D·)+1(D·)]
2进制浮点数(实数) 2进制浮点数(实数) 2进制浮点数(实数)

在[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定了常数(K、H)时，数值会自动被转换成2进制浮点数。



[(S1·)+1(S1·)]+[K2346] → [(D·)+1(D·)]
2进制浮点数　自动转化为2进制浮点数　2进制浮点数

注意要点

1. 指定了同一软元件时

[(S1·)+1，(S1·)]以及[(S2·)+1，(S2·)]和[(D·)+1，(D·)]中也可以指定同一个软元件编号。
此时，如使用连续执行型的指令(DEADD)，则每一个运算周期其加法运算的结果都会改变，请务必注意。

18.9 FNC 121 - ESUB/2进制浮点数减法运算

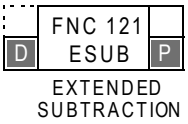
A8 A8C

概要

2个2进制浮点数减法运算的指令。

→ 浮点数运算的程序举例，参考12.10节
→ 浮点数的使用，请参考5.1.3项
→ 标志位的动作，请参考6.5.2项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
13步	DESUB	连续执行型
	DESUBP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存执行减法运算的2进制浮点数数据的字软元件编号	实数(2进制)*1
(S2・)	保存执行减法运算的2进制浮点数数据的字软元件编号	
(D・)	保存减法运算后的2进制浮点数数据	

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

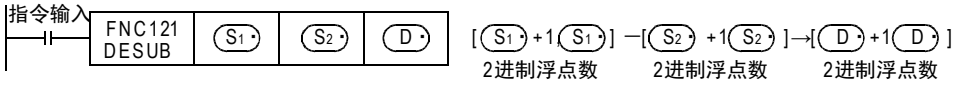
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)														●	●	▲			●	●	●	●		
(S2・)														●	●	▲			●	●	●	●		
(D・)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

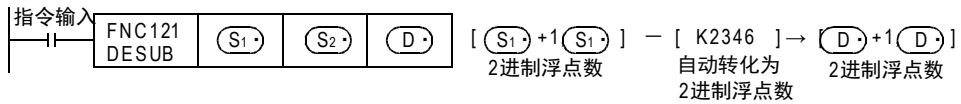
功能和动作说明

1. 32位运算(DESUB、DESUBP)

从[(S1・)+1，(S1・)]中减去[(S2・)+1，(S2・)]的2进制浮点数数据，并将其运算结果以2进制浮点数形式传送到[(D・)+1，(D・)]中。



在[(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]中指定了常数(K、H)时，数值会自动被转换成2进制浮点数。



注意要点

1. 指定了同一软元件时

[(S1・)+1，(S1・)]以及[(S2・)+1，(S2・)]和[(D・)+1，(D・)]中也可以指定同一个软元件编号。
此时，如使用连续执行型的指令(DESUB)，则每一个运算周期其减法运算的结果都会改变，请务必注意。

18.10 FNC 122 - EMUL/2进制浮点数乘法运算

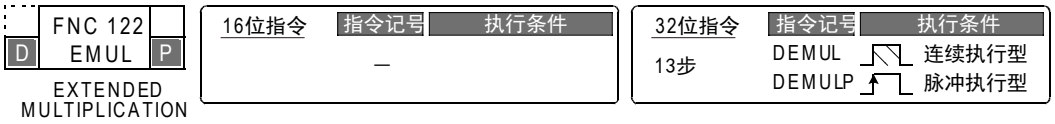
A8 A8C

概要

2个2进制浮点数乘法运算的指令。

→ 浮点数运算的程序举例，参考12.10节
→ 浮点数的使用，请参考5.1.3项

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1·)	保存执行乘法运算的2进制浮点数数据的字软元件编号	实数(2进制)*1
(S2·)	保存执行乘法运算的2进制浮点数数据的字软元件编号	
(D·)	保存乘法运算后的2进制浮点数数据的数据寄存器编号	

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

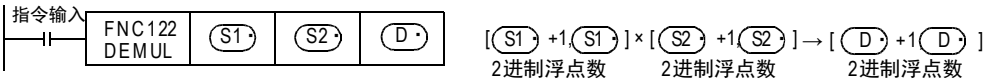
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●	●		
(S2·)														●	●	▲			●	●	●	●		
(D·)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

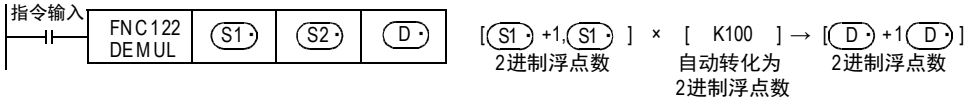
功能和动作说明

1. 32位运算(DEMUL、DEMULP)

将[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]的2进制浮点数数据相乘，并将其运算结果以2进制浮点数形式传送到[(D·)+1，(D·)]中。



在[(S1·)+1，(S1·)]和[(S2·)+1，(S2·)]中指定了常数(K、H)时，数值会自动被转换成2进制浮点数。



18.11 FNC 123 - EDIV/2进制浮点数除法运算

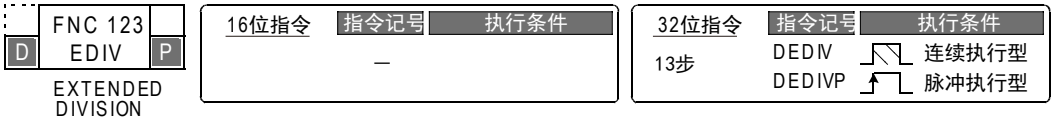
A8 A8C

概要

2个2进制浮点数除法运算的指令。

→ 浮点数运算的程序举例，参考12.10节
→ 浮点数的使用，请参考5.1.3项
→ 标志位的动作，请参考6.5.2项

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存执行除法运算的2进制浮点数数据的字软元件编号	实数(2进制)*1
(S2・)	保存执行除法运算的2进制浮点数数据的字软元件编号	
(D・)	保存除法运算后的2进制浮点数数据的数据寄存器编号	

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

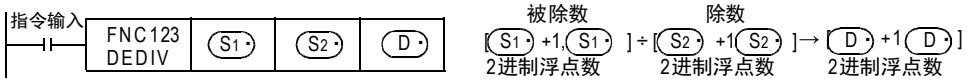
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)														●	●	▲			●	●	●	●		
(S2・)														●	●	▲			●	●	●	●		
(D・)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

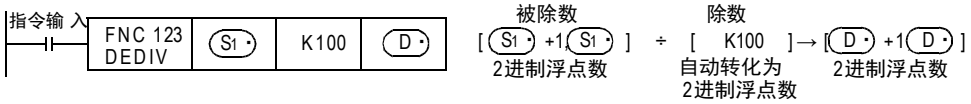
功能和动作说明

1. 32位运算(DEDIV、DEDIVP)

将[(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]的2进制浮点数数据相除，并将其运算结果以2进制浮点数形式传送到[(D・)+1，(D・)]中。



在[(S1・)+1，(S1・)]和[(S2・)+1，(S2・)]中指定了常数(K、H)时，数值会自动被转换成2进制浮点数。



18.12 FNC 124 - EXP/2进制浮点数指数运算

A8 A8C

概要

该指令是以e(2.71828)为底的指数运算指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



2. 设定数据

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

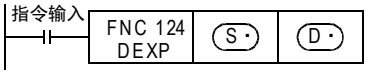
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●			●		
(D·)														●	●	●			●					

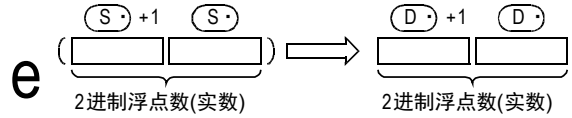
功能和动作说明

1. 32位运算(DEXP/DEXPP)

以[(S·)+1，(S·)]为指数做运算，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。



- 在指数运算中，将底(e)作为“2.71828”进行运算。



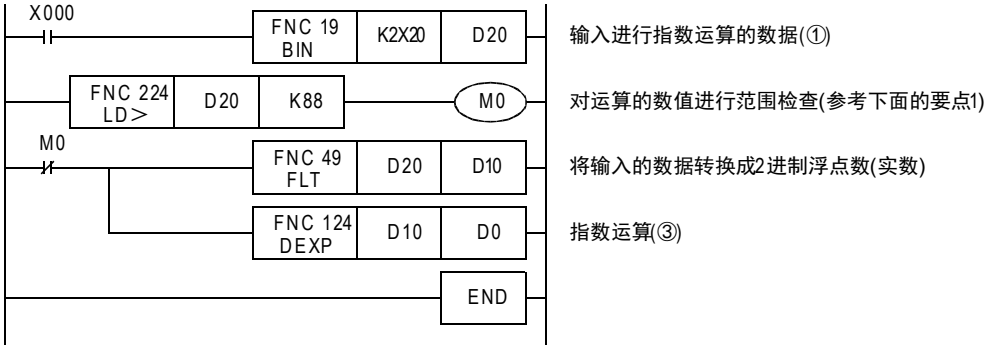
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

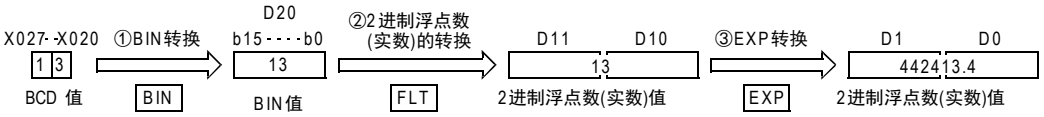
- 运算结果不在下面的范围时。(错误代码: K6706)
  $2^{-126} \leqq |$运算结果$| < 2^{128}$

程序举例

X000为ON后，对X020～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中指定了13时的动作



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，会发生运算错误，因此设定了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。

$$10^{X}=e^{\frac{X}{0.4342945}}$$

18.13 FNC 125 - LOGE/2进制浮点数自然对数运算

A8 A8C

概要

该指令执行自然对数运算。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DLOGE	连续执行型
	DLOGEP	脉冲执行型

2. 设定数据

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

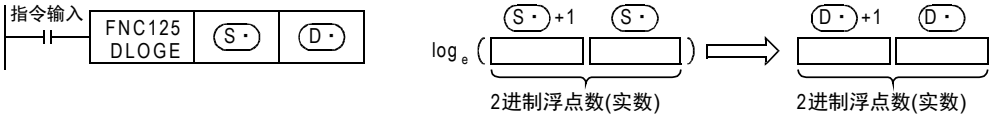
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●			●		
(D·)														●	●	●			●					

功能和动作说明

1. 32位运算(DLOGE/DLOGEP)

执行[(S·)+1，(S·)]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。



- [(S·)+1，(S·)]中指定的值，只可以设定正数。(负数不能运算。)

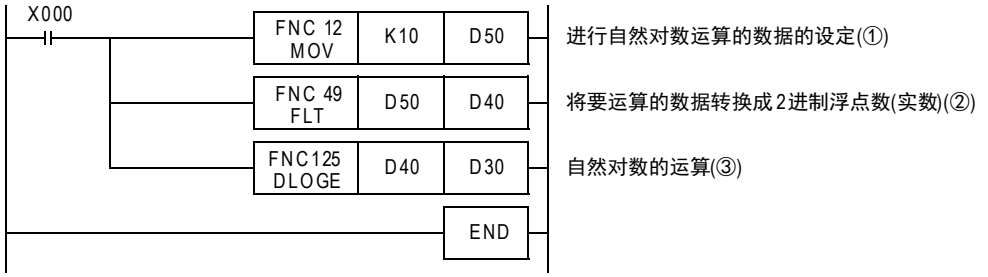
错误

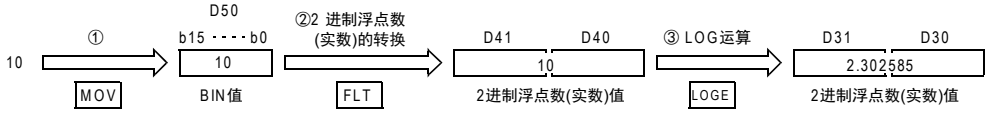
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- (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)

程序举例

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。





18.14 FNC 126 - LOG10/2进制浮点数常用对数运算

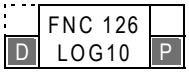
A8 A8C

概要

该指令执行常用对数运算。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DLOG10	连续执行型
	DLOG10P	脉冲执行型

2. 设定数据

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

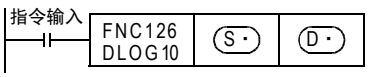
3. 对象软元件

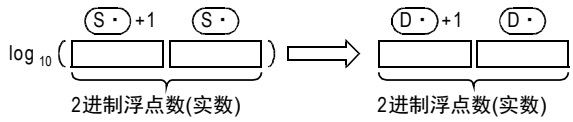
操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●			●		
(D·)														●	●	●			●					

功能和动作说明

1. 32位运算(DLOG10/DLOG10P)

执行[(S·)+1，(S·)]的常用对数(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，只可以设定正数。(负数不能运算。)

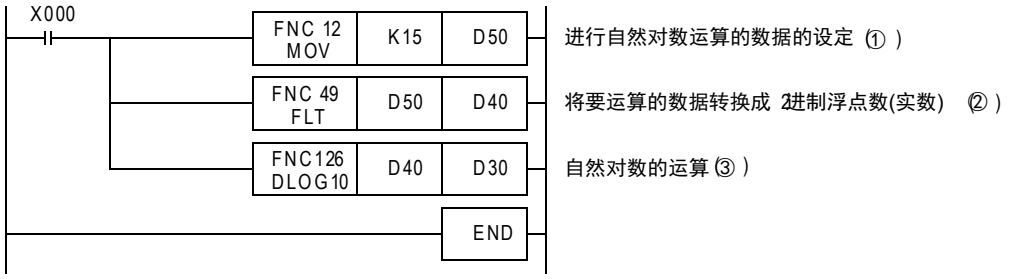
错误

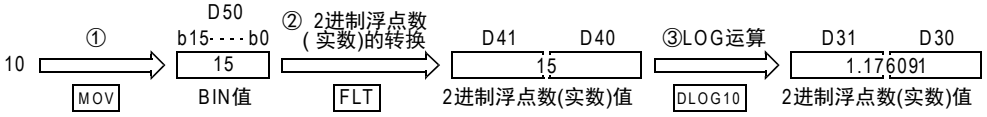
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- (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)

程序举例

X000为ON后，求出D50中设定的“15”的常用对数，并保存到D30、D31中的程序。





18.15 FNC 127 - ESQR/2进制浮点数开方运算

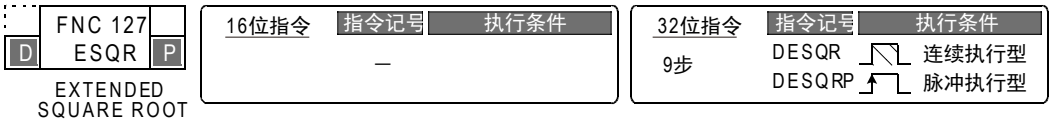
A8 A8C

概要

2进制浮点数开方(开根号)运算的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	保存执行开方运算的2进制浮点数数据的软元件的起始编号	实数(2进制)*1
(D·)	保存开方运算后的2进制浮点数数据的数据寄存器编号	

*1. 指定了常数(K、H)时，会自动将数值从BIN转换为2进制浮点数(实数)，再执行指令。

3. 对象软元件

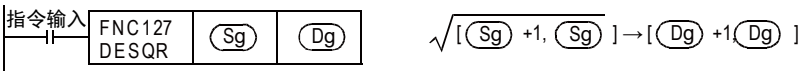
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	▲			●	●	●	●		
(D·)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

功能和动作说明

1. 32位运算(DESQR、DESQRP)

将[(S·)+1，(S·)]进行开方(开根号)运算(2进制浮点数运算)后，将其结果传送到[(D·)+1，(D·)]中。



相关软元件

→ 有关零标志的使用方法，参考6.5.2项

软元件	名称	内容
M8020	零位	运算结果真的为0时ON。

错误

[(S1·)+1，(S1·)]的内容仅正数有效，若为负数时，运算错误(M8067)动作，不执行指令。

18.16 FNC 128 - ENEG/2进制浮点数符号翻转

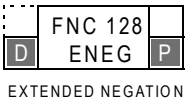
A8 A8C

概要

使2进制浮点数(实数)数据的符号翻转的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
5步	DENEG	连续执行型
	DENEGP	脉冲执行型

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D・)														●	●	●			●					

功能和动作说明

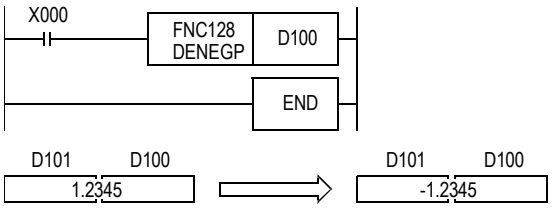
1. 32位运算(DENEG/DENEGP)

[(D・)+1，(D・)]的2进制浮点数数据的符号翻转，保存在[(D・)+1，(D・)]中。



程序举例

X000为ON时，将D100、D101的2进制浮点数数据的符号翻转，并且保存到D100、D101中的程序。



18.17 FNC 129 - INT/2进制浮点数→BIN整数的转换

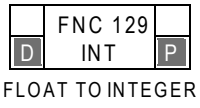
A8 A8C

概要

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

→ 浮点数运算的程序举例，参考12.10节
→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
5步	INT	连续执行型
	INTP	脉冲执行型

32位指令	指令记号	执行条件
9步	DINT	连续执行型
	DINTP	脉冲执行型

2. 设定数据

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

3. 对象软元件

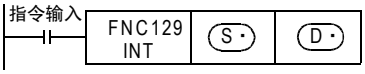
操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	▲			●					
(D·)														●	●	▲			●					

▲: 只对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(INT、INTP)

[(S·)+1，(S·)]的2进制浮点数转换成BIN整数后，传送到(D·)中。



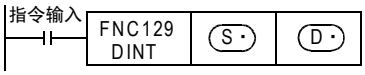
(S·)+1,(S·) → (D·)
2进制浮点数　16位BIN整数
小数点以后舍去

逆换动作的指令
这个指令运行的逆转换动作为FLT(FNC 49)。

→ 有关FLT(FNC 49)指令，参考12.10节

2. 32位运算(DINT、DINTP)

[(S·)+1，(S·)]的2进制浮点数转换成BIN整数后，传送到[(D·)+1，(D·)]中。



(S·)+1,(S·) → (D·)+1,(D·)
2进制浮点数　32位BIN整数
小数点以后舍去

逆换动作的指令
这个指令的逆转换为指令DFLT(FNC 49)。

→ 有关FLT(FNC 49)指令，参考12.10节

相关软元件

→ 关于零位、借位、进位标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8020	零位	运算结果为0时置ON。
M8021	借位	借位转换时，有因不满1而被舍去的情况发生时，置ON。
M8022	进位	运算结果超出-32,768～32,767(16位运算时)，或是-2,147,483,648～2,147,483,647(32位运算时)的范围而产生溢出时为ON。(运算结果不反映。)

注意要点

1. 运算时的注意事项
   - 小数点以后的值被舍去。

18.18 FNC 130 - SIN/2进制浮点数SIN运算

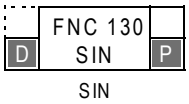
A8 A8C

概要

求角度(RAD)的SIN值的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
9步	DSIN	连续执行型
	DSINP	脉冲执行型

2. 设定数据

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

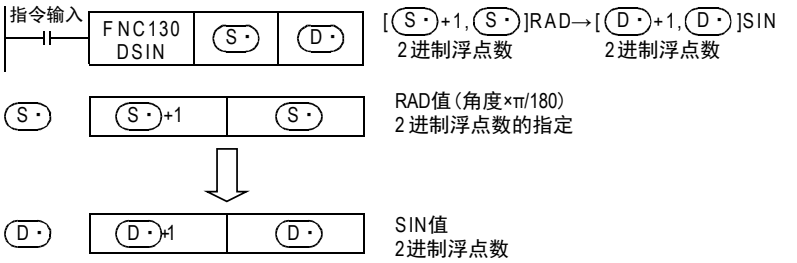
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●			●		
(D·)														●	●	●			●					

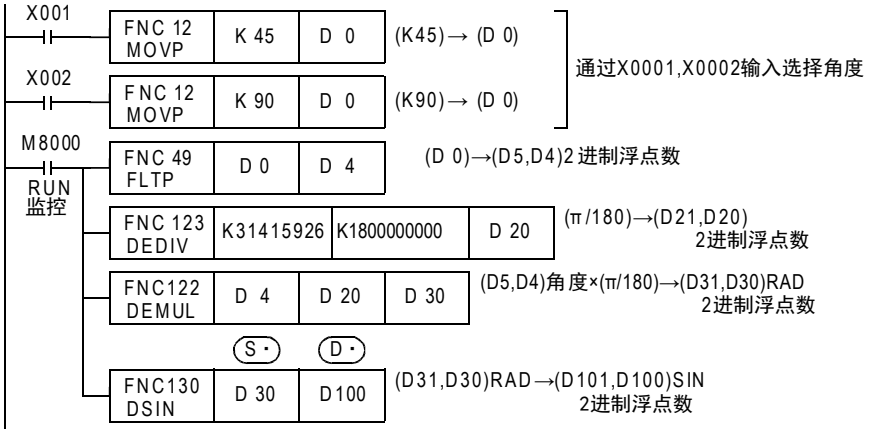
功能和动作说明

1. 32位运算(DSIN、DSINP)

将[(S·)+1，(S·)]中指定的角度值(2进制浮点数)转换成SIN值后，传送到[(D·)+1，(D·)]中。



程序举例



18.19 FNC 131 - COS/2进制浮点数COS运算

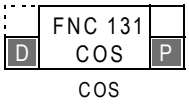
A8 A8C

概要

求角度(RAD)的COS值的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
9步	DCOS	连续执行型
	DCOSP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存2进制浮点数的RAD(角度)的软元件编号	实数(2进制)
(D・)	保存2进制浮点数的COS值的软元件编号	

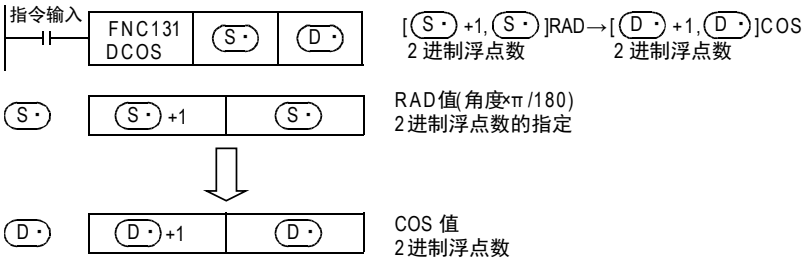
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DCOS、DCOSP)

将[(S・)+1，(S・)]中指定的角度值(2进制浮点数)转换成COS值后，传送到[(D・)+1，(D・)]中。



18.20 FNC 132 - TAN/2进制浮点数TAN运算

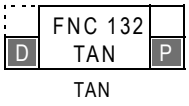
A8 A8C

概要

求角度(RAD)的TAN值的指令。

→ 浮点数的使用，请参考5.2.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	

32位指令	指令记号	执行条件
9步	DTAN	连续执行型
	DTANP	脉冲执行型

2. 设定数据

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

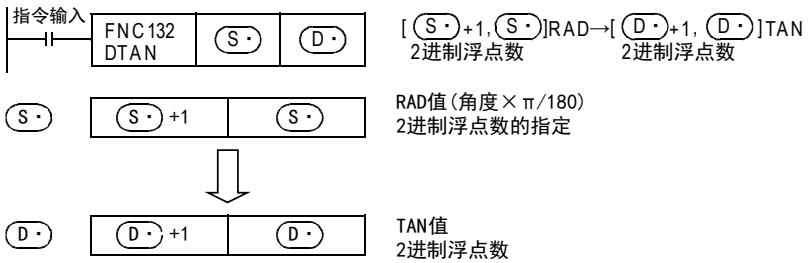
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)														●	●	●			●			●		
(D·)														●	●	●			●					

功能和动作说明

1. 32位运算(DTAN、DTANP)

将[(S·)+1，(S·)]中指定的角度值(2进制浮点数)转换成TAN值后，传送到[(D·)+1，(D·)]中。



18.21 FNC 133 - ASIN/2进制浮点数SIN^{-1}运算

A8 A8C

概要

执行SIN^{-1}运算的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DASIN	连续执行型
	DASINP	脉冲执行型

2. 设定数据

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

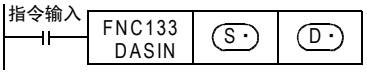
3. 对象软元件

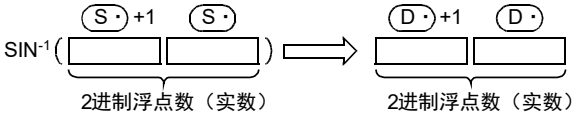
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DASIN/DASINP)

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





- [(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(-π/2)～(π/2)的值。
  关于弧度与角度之间的转换，请参考RAD(FNC 136)命令、DEG(FNC 137)指令。

→ 有关RAD(FNC 136)指令，请参考18.24节
→ 有关DEG(FNC 137)指令，请参考18.25节

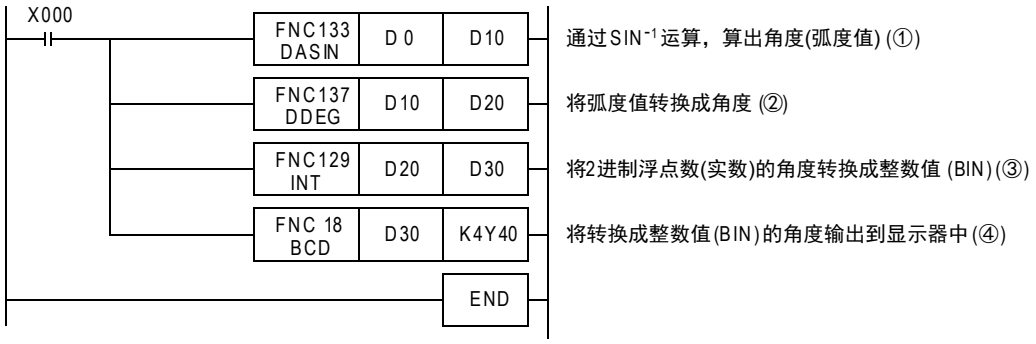
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

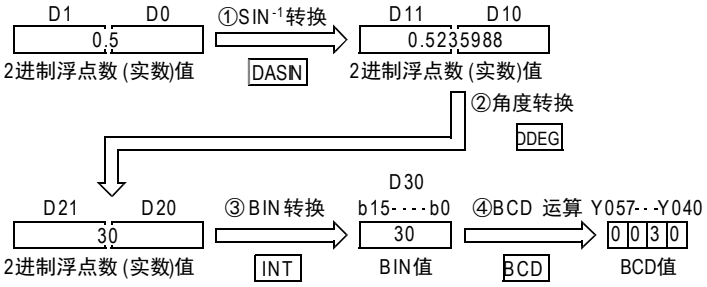
- (S・)中指定的值不在-1.0～1.0的范围内时。(错误代码: K6706)

程序举例

X000为ON时，求出D0、D1(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



18.22 FNC 134 - ACOS/2进制浮点数COS^{-1}运算

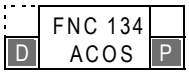
A8 A8C

概要

执行COS^{-1}运算的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DACOS	连续执行型
	DACOSP	脉冲执行型

2. 设定数据

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

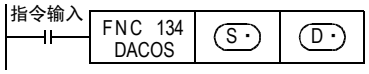
3. 对象软元件

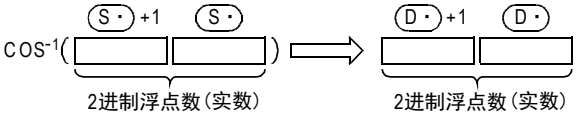
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DACOS/DACOSP)

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





- [(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0～π的值。
  关于弧度与角度之间的转换，请参考RAD(FNC 136)指令、DEG(FNC 137)指令。

→ 有关RAD(FNC 136)指令，请参考18.24节
→ 有关DEG(FNC 137)指令，请参考18.25节

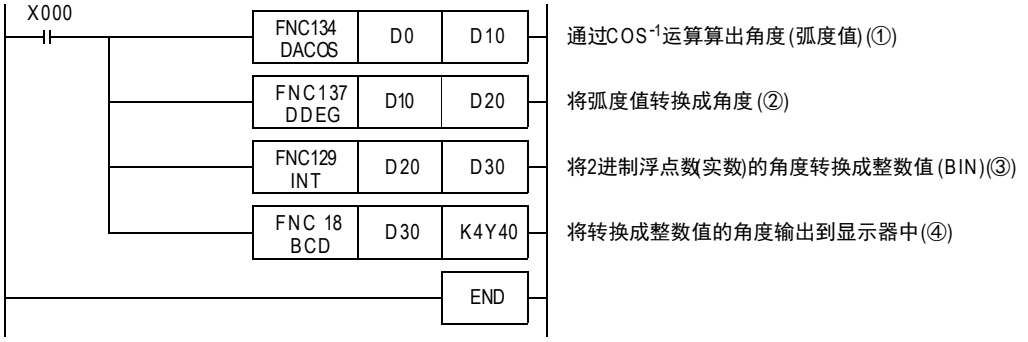
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

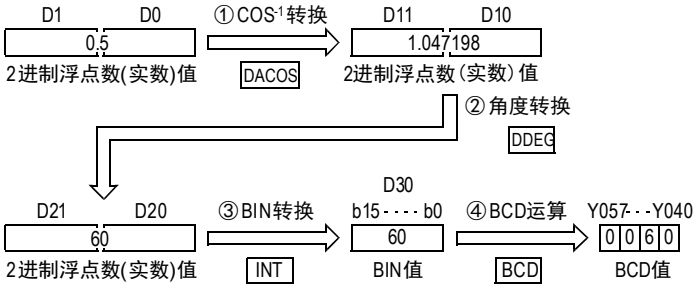
- (S・)中指定的值不在-1.0～1.0的范围内时。(错误代码: K6706)

程序举例

X000为ON时，求出D0、D1(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



18.23 FNC 135 - ATAN/2进制浮点数TAN^{-1}运算

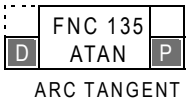
A8 A8C

概要

执行TAN^{-1}运算的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DATAN	连续执行型
	DATANP	脉冲执行型

2. 设定数据

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

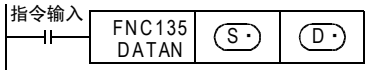
3. 对象软元件

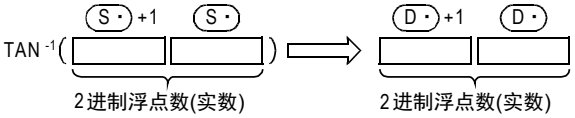
操作数种类	位软元件							字软元件											其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DATAN/DATANP)

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



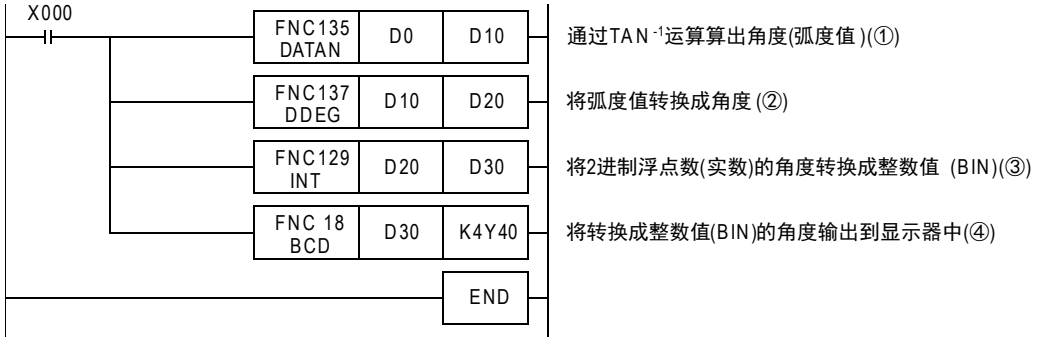


- [(D・)+1，(D・)]中的角度(运算结果)，保存以弧度为单位的比(-π/2)大、比(π/2)小的数值。
  关于弧度与角度之间的转换，请参考RAD(FNC 136)指令、DEG(FNC 137)指令。

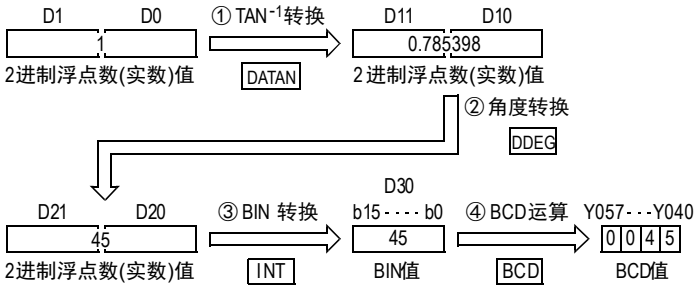
→ 有关RAD(FNC 136)指令，请参考18.24节
→ 有关DEG(FNC 137)指令，请参考18.25节

程序举例

X000为ON时，求出D0、D1(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



18.24 FNC 136 - RAD/2进制浮点数角度→弧度的转换

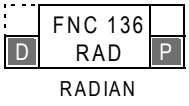
A8 A8C

概要

将角度单位的值转换成弧度单位的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DRAD	连续执行型
	DRADP	脉冲执行型

2. 设定数据

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

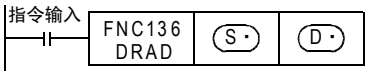
3. 对象软元件

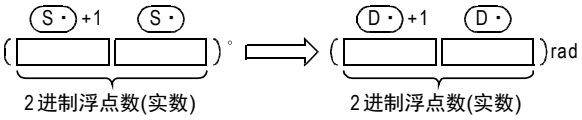
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DRAD/DRADP)

[(S・)+1，(S・)]的单位从角度单位转换成弧度单位，并保存到[(D・)+1，(D・)]中。
此外，可以在(S・)中直接指定实数。



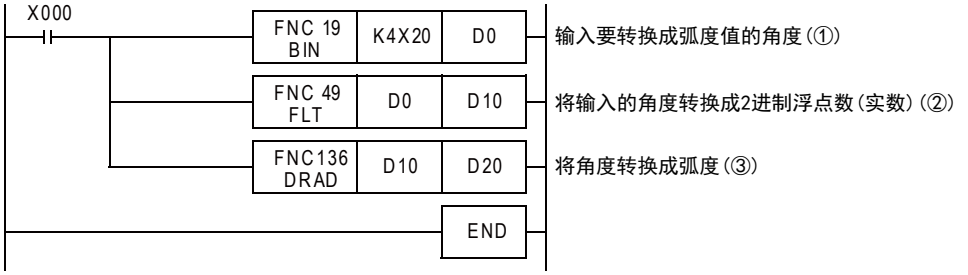


- 角度单位→弧度单位的转换如下所示执行。

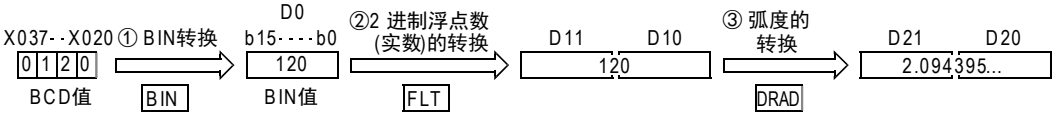
$$\text{弧度单位} = \text{角度单位} \times \frac{\pi}{180}$$

程序举例

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进制浮点数形式保存在D20、D21中的程序。



在X020～X037中指定了120时的动作



18.25 FNC 137 - DEG/2进制浮点数弧度→角度的转换

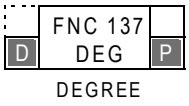
A8 A8C

概要

将弧度单位的值转换成角度(DEG)单位的指令。

→ 浮点数的使用，请参考5.1.3项

1. 指令格式



16位指令	指令记号	执行条件
	—	
	—	

32位指令	指令记号	执行条件
9步	DDEG	连续执行型
	DDEGP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要转换成角度单位的弧度的软元件起始编号	实数(2进制)
(D・)	保存已转换成角度单位的值的软元件起始编号	

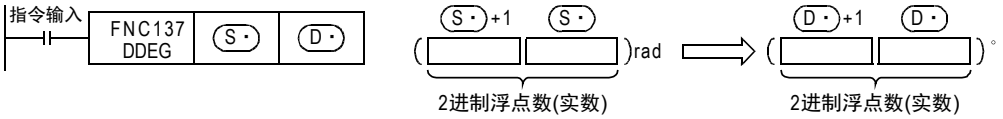
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)														●	●	●			●			●		
(D・)														●	●	●			●					

功能和动作说明

1. 32位运算(DDEG/DDEGP)

[(S・)+1，(S・)]的单位从弧度单位转换成角度单位，并保存到[(D・)+1，(D・)]中。

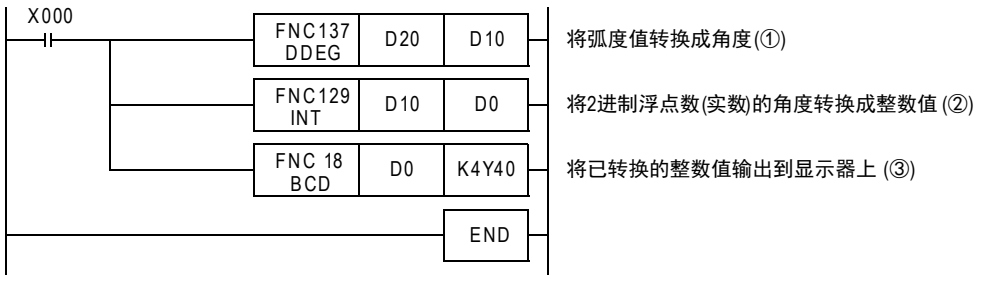


- 弧度单位→角度单位的转换如下所示执行。

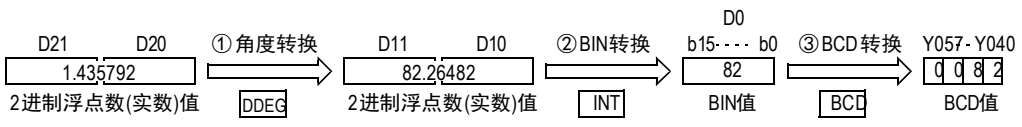
$$角度单位 = 弧度单位 \times \frac{180}{\pi}$$

程序举例

当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值的形式输出到Y040～Y057中的程序。

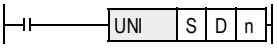
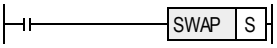
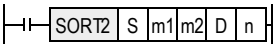


D20、D21的值为1.435792时的动作



19. 数据处理2—FNC 140～FNC 149

提供了与基本的应用指令相比，能执行更加复杂的处理，或是用于满足特殊用途的指令。

FNC No.	指令记号	符号	功能	参考
140	WSUM		算出数据合计值	19.1节
141	WTOB		字节单位的数据分离	19.2节
142	BTOW		字节单位的数据结合	19.3节
143	UNI		16位数据的4位结合	19.4节
144	DIS		16位数据的4位分离	19.5节
145	—			
146	—			
147	SWAP		高低字节互换	19.6节
148	—			
149	SORT2		数据排序2	19.7节

19.1 FNC 140 - WSUM/算出数据合计值

A8 A8C

概要

该指令可计算出连续的16位或是32位数据的合计值。
计算以字节(8位)为单位的加法运算数据(合计值)时，请使用CCD(FNC 84)指令。

→ 关于CCD(FNC 84)指令，请参考16.5节

1. 指令格式

FNC 140		
D	WSUM	P
WORD SUM		

16位指令	指令记号	执行条件
7步	WSUM	连续执行型
	WSUMP	脉冲执行型

32位指令	指令记号	执行条件
13步	DWSUM	连续执行型
	DWSUMP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要算出合计值的数据的软元件起始编号	BIN16/32位
(D・)	保存合计值的软元件起始编号	BIN32/64位
n	数据个数(0＜n)	BIN16/32位

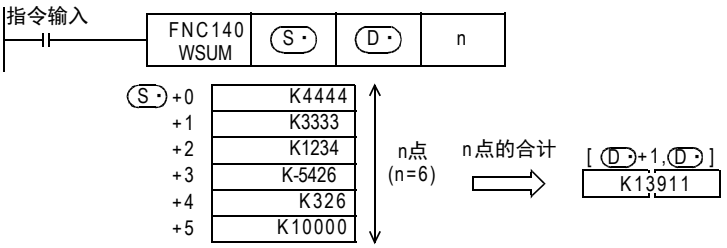
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●	●			●					
(D・)												●	●	●	●	●			●					
n														●	●					●	●			

功能和动作说明

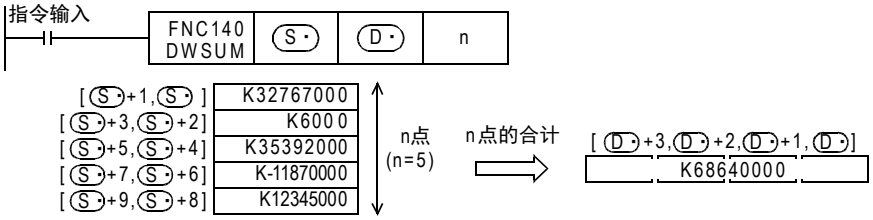
1. 16位运算(WSUM/WSUMP)

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



2. 32位运算(DWSUM/DWSUMP)

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



相关指令

指令	内容
CCD(FNC 84)	校验码 以字节(8位)为单位，计算出16位数据的合计值以及水平检验数据的指令。

注意要点

- 32位运算中，合计值为64位的数据。A8/A8C可编程控制器中，不能处理64位数据。但是，当合计值在32位数据的数值范围(K-2,147,483,648～K2,147,483,647)内时，可以忽略高位的32位数据，而只将低位的32位数据作为合计值处理。
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DWSUM D0 D100 R0时，则n=[R1, R0]。

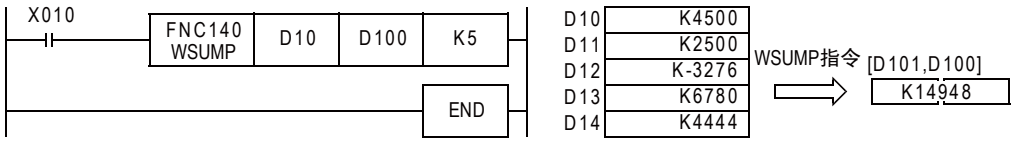
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 以(S·)开始的n点软元件超过了指定软元件的软元件范围时。(错误代码: K6706)
- n≦0时(错误代码: K6706)
- (D·)超出了软元件的范围。(错误代码: K6706)

程序举例

当X010为ON时，将D10～D14的16位数据合计值保存到[D101，D100]中的程序。



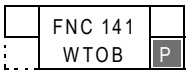
19.2 FNC 141 - WTOB/字节单位的数据分离

A8 A8C

概要

将连续的16位数据按照字节(8位)单位进行分离的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	WTOB	连续执行型
	WTOBP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

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

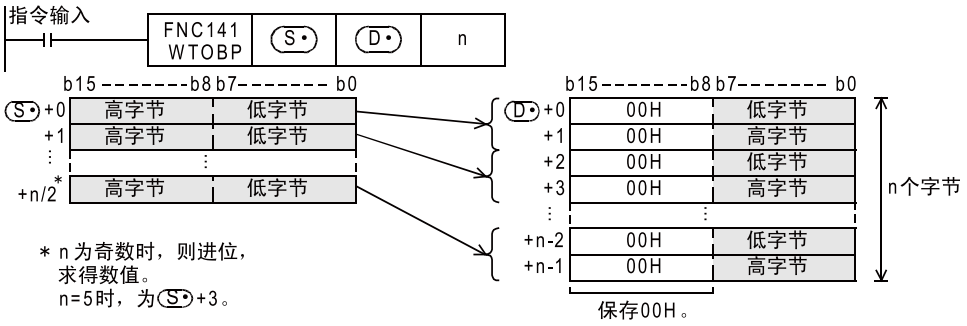
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

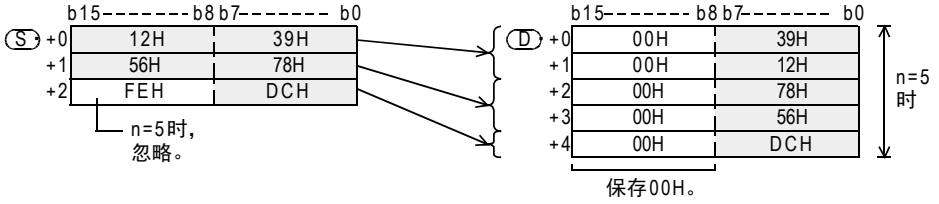
1. 16位运算(WTOB/WTOBP)

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，如下所示保存到以(D・)开始的n点软元件中。



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

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



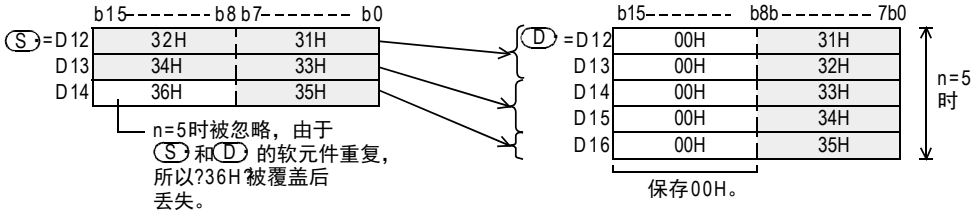
4) n=0时，不执行指令。

相关指令

指令	内容
BTOW(FNC 142)	将连续的16位数据的低8位(低字节)结合起来的指令。

注意要点

保存分离源数据的软元件和保存已经分离的数据的软元件可以重复使用。
但是，请注意当n为奇数时，如下面的例子所示，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。



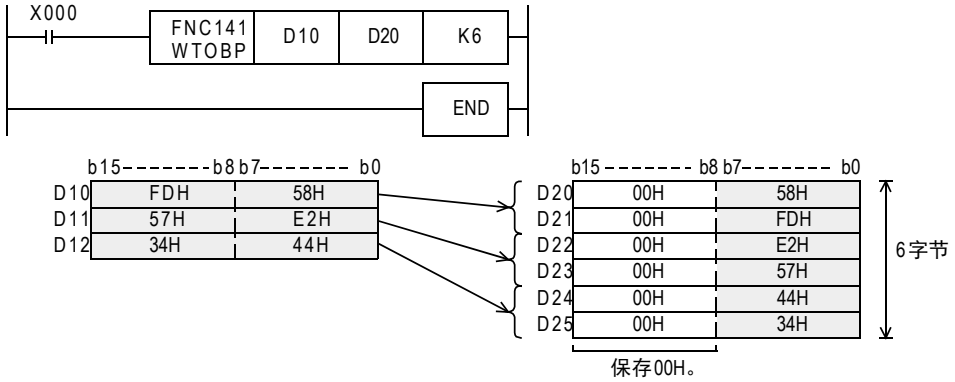
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 当分离源软元件的(S·)～((S·)+n/2)超出了指定软元件的软元件范围时。
  n为奇数时，需要占用进位后数值的个数部分的软元件。(错误代码: K6706)
- 当保存已分离的数据的软元件(D·)～((D·)+n-1)超出了指定软元件的软元件范围时。(错误代码: K6706)

程序举例

当X000为ON后，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。



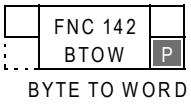
19.3 FNC 142 - BTOW/字节单位的数据结合

A8 A8C

概要

将连续的16位数据的低8位(低字节)结合在一起的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	BTOW	连续执行型
	BTOWP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

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

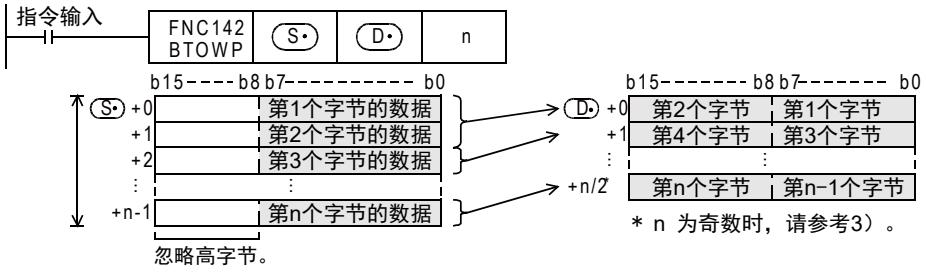
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

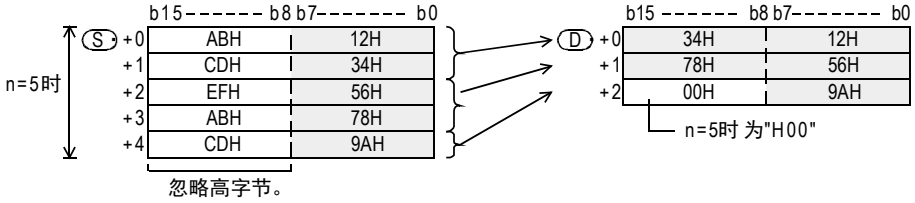
1. 16位运算(BTOW/BTOWP)

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，如下所示保存到以(D・)开始的n/2点软元件中。



2) 结合源的16位数据((S・)以后)的高字节(8位)被忽略。

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



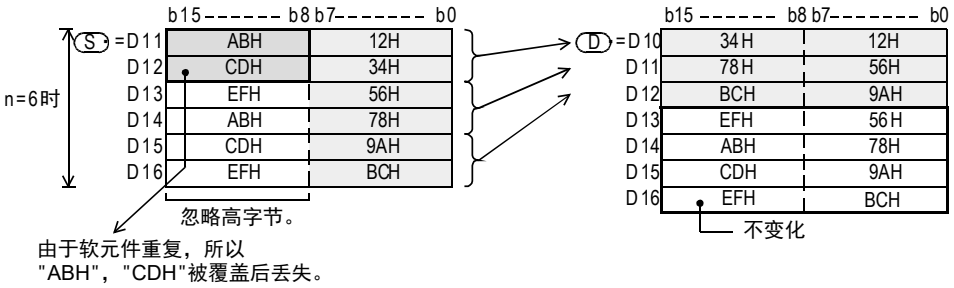
4) n=0时，不执行指令。

相关指令

指令	内容
WTOB(FNC 141)	按照字节单位(8位)分离连续的16位数据的指令。

注意要点

保存结合源数据的软元件和保存已经结合的数据的软元件可以重复使用。
但是，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)，如下面的例子所示，高字节(8位)的数据会在被结合后的数据覆盖后丢失。



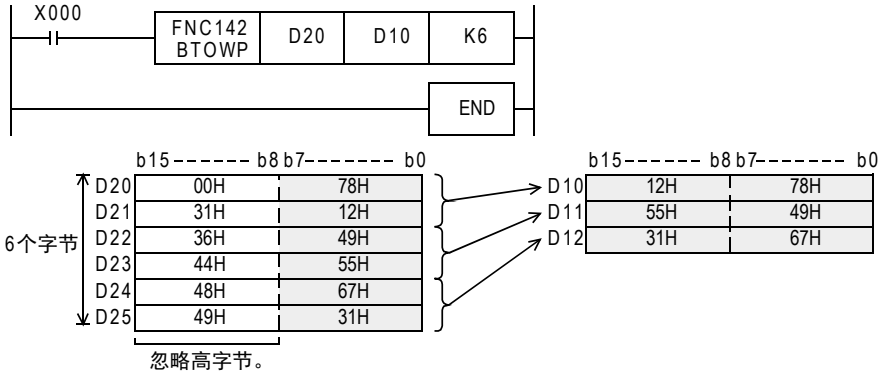
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 结合源的软元件的(S·)～((S·)+n-1)中指定的软元件超出了该软元件范围时。
  (错误代码: K6706)
- 当保存已结合数据的软元件(D·)～((D·)+n/2)超出了指定软元件的软元件范围时。当n为奇数时，需要占用进位后数值的个数部分的软元件。(错误代码: K6706)

程序举例

当X000为ON后，D20～D25的低字节(8位)数据被结合后，保存到D10～D12中的程序。



19.4 FNC 143 - UNI/16位数据的4位结合

A8 A8C

概要

将连续的16位数据的低4位结合在一起的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	UNI	连续执行型
	UNIP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

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

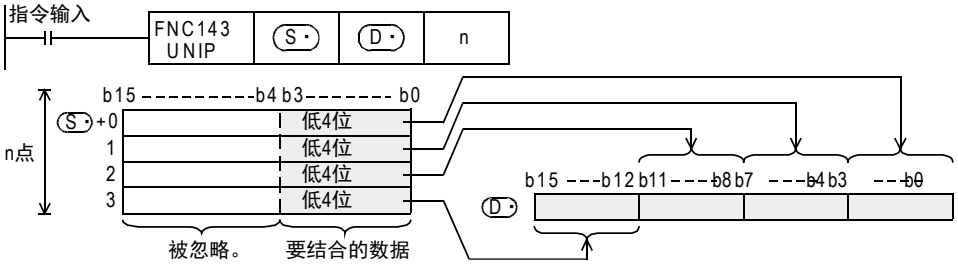
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

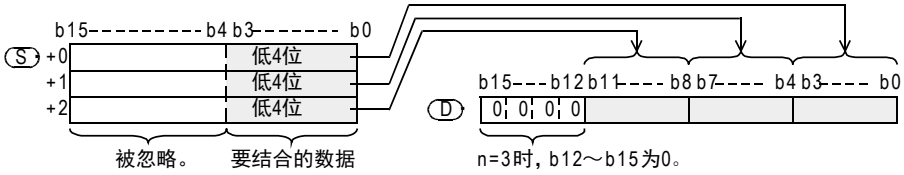
1. 16位运算(UNI/UNIP)

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



2) 在 n 中指定1～4。
n=0时，不执行指令。

3) 1≦n≦3时， (D・) 的高位{4×(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位变为0。



相关指令

指令	内容
DIS(FNC 144)	按照4位的单位分离16位数据的指令。

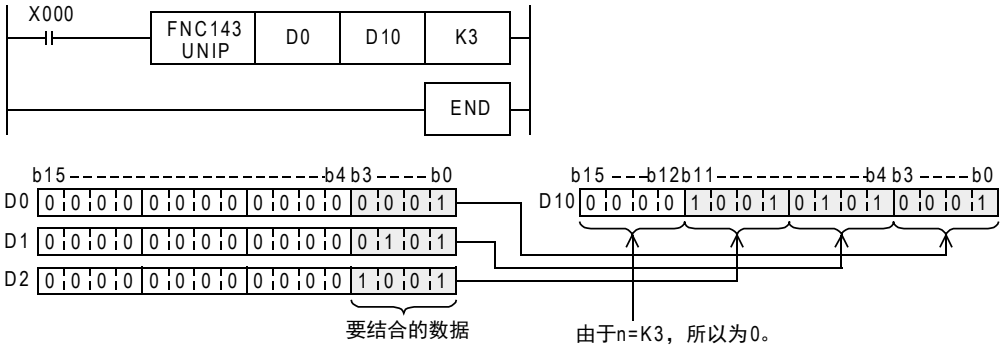
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (S·)～(S·)+n)中指定的软元件超出了该软元件范围时。(错误代码: K6706)
- n指定了0～4以外的数字时。(错误代码: K6706)

程序举例

当X000为ON后，将D0～D2的低4位结合后，保存到D10中的程序。



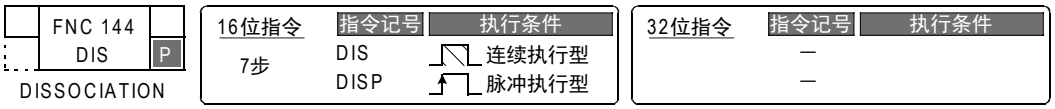
19.5 FNC 144 - DIS/16位数据的4位分离

A8 A8C

概要

将16位数据以4位为单位分离的指令。

1. 指令格式



2. 设定数据

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

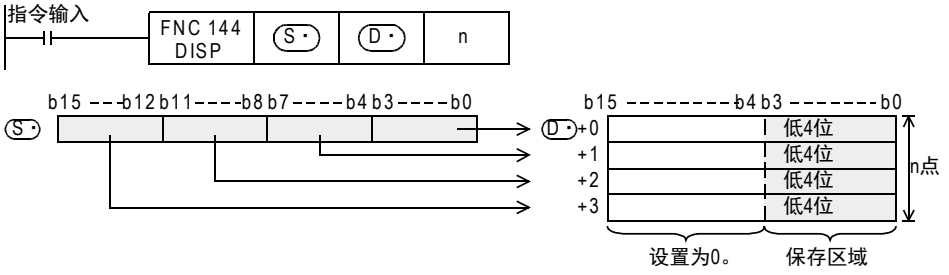
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(DIS/DISP)

1) 将(S・)的16位数据以4位结合为单位分离后，如下所示保存在(D・)中。



2) 在n中指定1～4。
n=0时，不执行指令的处理。

3) (D・)开始的n点软元件中的高12位设置为0。

相关指令

指令	内容
UNI(FNC 143)	16位数据的低4位结合的指令。

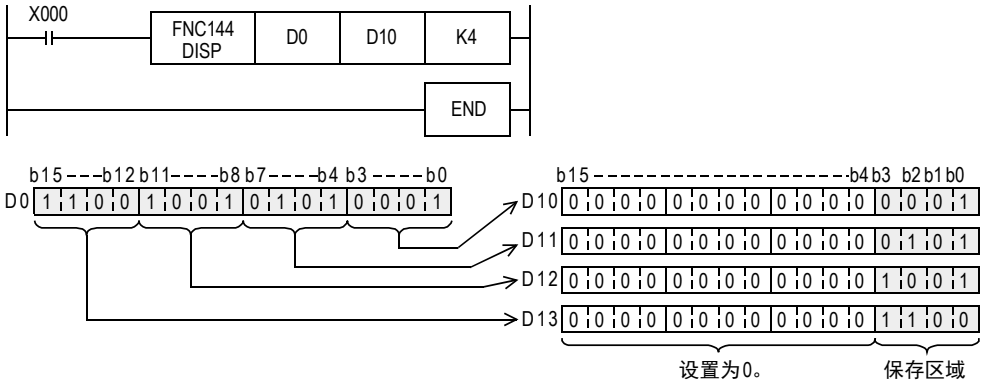
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 当(D·)开始的n点软元件超出了指定软元件的软元件范围时。(错误代码: K6706)
- n指定了0～4以外的数字时。(错误代码: K6706)

程序举例

当X000为ON后，将D0每隔4个位分离后，保存到D10～D13中的程序。



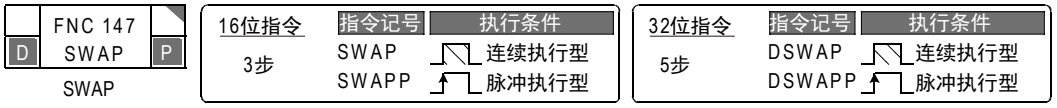
19.6 FNC 147 - SWAP/高低字节互换

A8 A8C

概要

互换字数据的高8位和低8位的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	高低字节互换的字软元件	BIN16/32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)									●	●	●	●	●	●	●	●	●	●	●					

功能和动作说明

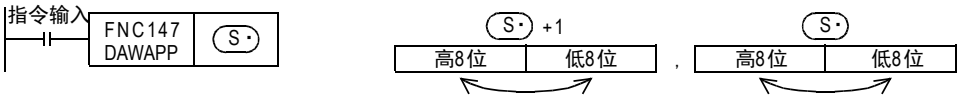
1. 16位运算(SWAP/SWAPP)

执行低8位和高8位的互换。



2. 32位运算(DSWAP/DSWAPP)

即使是32位指令，也是执行各自的低8位和高8位的互换。



注意要点

- 使用连续执行型指令时，请注意每个运算周期都会执行互换。
  与XCH(FNC 17)指令的扩展功能动作相同。

19.7 FNC 149 - SORT2/数据排序2

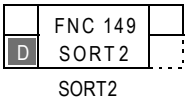
A8 A8C

概要

以指定的群数据(列)为基准，以行为单位，将由数据(行)和群数据(列)构成的数据表进行升序/降序重新排序的指令。在这个指令中，由于是在连续的软元件中保存数据(行方向)，所以便于增加数据(行)。
此外，还有仅支持升序排列，数据结构(以列方向中连续的软元件构成数据)不同的SORT(FNC 69)指令。

→ 有关SORT(FNC 69)指令，参考14.10节

1. 指令格式



16位指令	指令记号	执行条件
11步	SORT2	连续执行型

32位指令	指令记号	执行条件
21步	DSORT2	连续执行型

2. 设定数据

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

3. 对象软元件

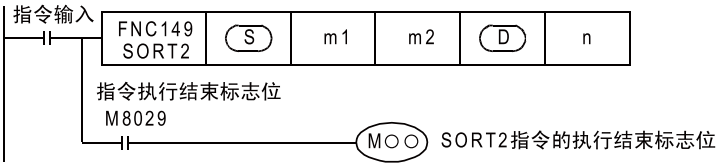
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S)														●	●									
m1														●	●					●	●			
m2																				●	●			
(D)														●	●									
n														●	●					●	●			

功能和动作说明

1. 16位运算(SORT2)

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

→ 有关动作举例，请参考531页



下面例举排序前m1=K3、 m2=K4的例子说明数据表格的结构。 在排序后的数据表格中， 请将(S)改读成(D)。

行号 \ 列号		群数m2个(m2=K4时)			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数m1=3的情况	1	(S)	(S)+1	(S)+2	(S)+3
	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。

	设定排序的顺序
M8165=ON	降序
M8165=OFF	升序

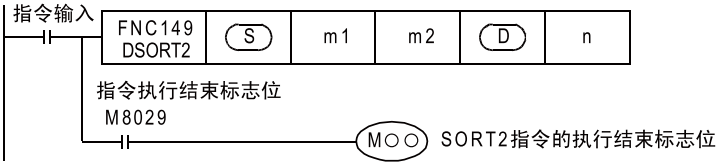
- 指令输入为ON时开始数据排列，m1个扫描后数据排列结束，指令执行结束标志位M8029为ON。

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

2. 32位运算(DSORT2)

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

→ 有关动作举例，请参考531页



下面例举排序前m1=K3、m2=K4的例子说明数据表格的结构。在排序后的数据表格中，请将 (S) 改读成 (D) 。

列号 / 行号		群数m2个(m2=K4时)			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数m1=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。

	设定排序的顺序
M8165=ON	降序
M8165=OFF	升序

- 在m1中使用数据寄存器D或扩展寄存器R时，为32位长度的数据。
  例如，在D0中指定m1时，m1为[D1，D0]的32位数据。
- 指令输入为ON时开始数据排列，m1个扫描后数据排列结束，指令执行结束标志位M8029为ON。

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

3. 动作举例

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

排序前数据

行号 \ 列号		群数m2个(m2=K4时)			
		1	2	3	4
		管理编号	身高	体重	年龄
数据数 m1=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)

行号 \ 列号	1	2	3	4
	管理编号	身高	体重	年龄
1	(D)	(D)+1	(D)+2	(D)+3
	4	100	20	8
2	(D)+4	(D)+5	(D)+6	(D)+7
	1	150	45	20
3	(D)+8	(D)+9	(D)+10	(D)+11
	5	150	50	45
4	(D)+12	(D)+13	(D)+14	(D)+15
	3	160	70	30
5	(D)+16	(D)+17	(D)+18	(D)+19
	2	180	50	40

2) 以n=K3(列号3)为基准执行指令时的排序结果(降序的情况)

行号 \ 列号	1	2	3	4
	管理编号	身高	体重	年龄
1	(D)	(D)+1	(D)+2	(D)+3
	3	160	70	30
2	(D)+4	(D)+5	(D)+6	(D)+7
	2	180	50	40
3	(D)+8	(D)+9	(D)+10	(D)+11
	5	150	50	45
4	(D)+12	(D)+13	(D)+14	(D)+15
	1	150	45	20
5	(D)+16	(D)+17	(D)+18	(D)+19
	4	100	20	8

相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

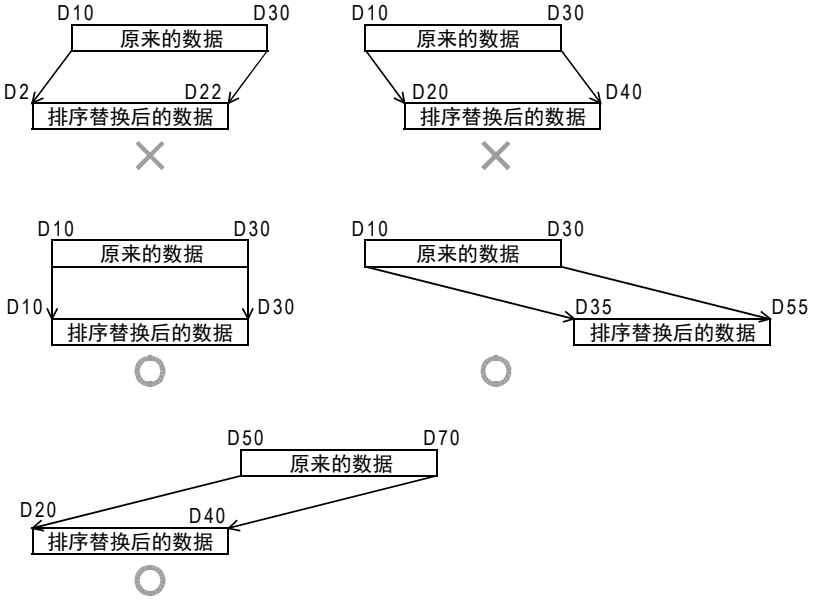
软元件	名称	内容
M8029	指令执行结束	数据排列结束时位ON。
M8165	降序排序	M8165=ON时，降序排列。 M8165=OFF时，升序排列。

相关指令

指令	内容
SORT(FNC 69)	数据排列 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

- 动作过程中，请勿使操作数和数据的内容变化。
- 再次执行时，请将指令输入OFF一次。
- 指令的使用次数的限制
  在程序中最多可同时驱动2次。
- 包含该指令的回路块不能在RUN中写入。
- (S)和(D)中指定了同一个软元件时
  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。
- 原来的数据和排序替换后的数据，请错开，不要重叠。



- 将D、R指定为32位指令的m1、n时，[m1+1,m1]、[n+1,n]的32位值便生效，敬请注意。
  DSORT2 D0 D50 K4 D100 R0时，则m1=[D51, D50]、n=[R1, R0]。

20. 定位控制—FNC 150～FNC 159

在FNC 150～FNC 159中，提供了使用可编程控制器内置的脉冲输出功能进行定位控制的指令。

→ 详细内容，请参考定位控制手册

FNC No.	指令记号	符号	功能	参考
150	DSZR	DSZR S1 S2 D1 D2	带DOG搜索的原点回归	20.1节
151	DVIT	DVIT S1 S2 D1 D2	中断定位	20.2节
152	TBL	TBL D n	表格设定定位	20.3节
153	—			—
154	—			—
155	ABS	ABS S D1 D2	读出ABS当前值	20.4节
156	ZRN	ZRN S1 S2 S3 D	原点回归	20.5节
157	PLSV	PLSV S D1 D2	可变速脉冲输出	20.6节
158	DRVI	DRVI S1 S2 D1 D2	相对定位	20.7节
159	DRVA	DRVA S1 S2 D1 D2	绝对定位	20.8节

关于RUN中写入的注意事项

在执行定位控制指令(FNC 150～151、156～159)的过程中(脉冲输出过程中)，请避免RUN中写入。
如果在脉冲输出过程中，对包含如下指令的梯形图程序执行了RUN中写入时，请注意会如下所示动作。

对象指令		指令动行中执行RUN中写入时的动作
DSZR(FNC 150)		脉冲输出减速停止。
DVIT(FNC 151)		
TBL(FNC 152)		不能执行RUN中写入。
ZRN(FNC 156)		脉冲输出减速停止。
PLSV(FNC 157)	带加减速动作时	脉冲输出减速停止。
	无加减速动作时	即时停止脉冲输出。
DRVI(FNC 158)		脉冲输出减速停止。
DRVA(FNC 159)		

20.1 FNC 150 - DSZR/带DOG搜索的原点回归

A8 / A8C

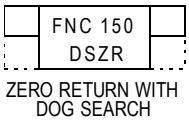
概要

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，ZRN(FNC 156)指令不支持以下几种情况，但本指令可以支持。

- DOG搜索功能的对应
- 允许使用近点DOG和零点信号的原点回归
  但是，不可以对零点信号计数后决定原点。

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	DSZR	连续执行型
	—	

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1•)	指定输入近点信号(DOG)的软元件编号。	位
(S2•)	指定输入零点信号的输入编号。	
(D1•)	指定输出脉冲的输出编号。	
(D2•)	指定旋转方向信号的输出对象编号。	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)	●	●	●			●	▲1												●					
(S2•)	▲2																		●					
(D1•)		▲3																	●					
(D2•)		▲4	●			●	▲1												●					

▲1: D□.b仅适用于A8/A8C可编程控制器。但是，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的Y000、Y001、Y002或是高速输出特殊适配器*1的Y000、Y001、Y002*2、Y003*2。

*1. 高速输出特殊适配器仅可连接A8可编程控制器。

*2. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

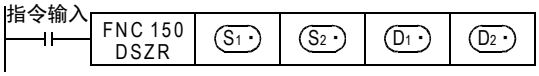
要点

- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

▲4: 使用A8可编程控制器的脉冲输出对象地址中高速输出特殊适配器时，旋转方向信号
请使用下表中的输出
使用A8/A8C可编程控制器的脉冲输出对象地址中内置的晶体管输出时，旋转方向信号请使用晶体管输出。

高速输出特殊适配器的连接位置	脉冲输出	旋转方向的输出
第1台	(D1•) =Y000	(D2•) =Y004
	(D1•) =Y001	(D2•) =Y005
第2台	(D1•) =Y002	(D2•) =Y006
	(D1•) =Y003	(D2•) =Y007

功能和动作说明



关于RUN中写入的注意事项

在执行DSZR(FNC 150)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。

根据版本不同的功能变更

根据版本不同，这个指令的功能如下表变更。

→ 有关指令的说明以及功能变更的内容，请参考定位控制手册

对应版本		项目	功能概要
A8	A8C		
Ver.2.20 以上	Ver.2.20 以上	指定清除信号输出对象的功能	(D1•) 对应的特殊辅助继电器置ON后，将清除信号的输出对象更改成 (D•) 对应的特殊数据寄存器中指定的输出编号。

20.2 FNC 151 - DVIT/中断定位

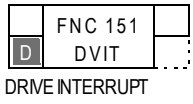
A8 A8C

概要

执行单速中断定长进给的指令。

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	DVIT	连续执行型
	—	

32位指令	指令记号	执行条件
17步	DDVIT	连续执行型
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1•)	指定中断后的输出脉冲数(相对地址)。*1	BIN16/32位
(S2•)	指定输出脉冲频率。*2	
(D1•)	指定输出脉冲的输出编号。	位
(D2•)	指定旋转方向信号的输出对象编号。	

*1. 设定范围: 16位运算时为-32,768～+32,767(0除外)
32位运算时为-999,999～+999,999(0除外)

*2. 设定范围: 16为运算时为10～32,767(Hz)
32位运算时如下所示。

脉冲输出对象		设定范围
A8可编程控制器	高速输出特殊适配器	10～200,000(Hz)
A8/A8C可编程控制器	基本单元(晶体管输出)	10～100,000(Hz)

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2•)								●	●	●	●	●	●	●	●	●			●	●	●			
(D1•)		▲1																	●					
(D2•)		▲2	●			●	▲3												●					

▲1: 请指定基本单元的晶体管输出Y000、Y001、Y002，或是高速输出特殊适配器*3的Y000、Y001、Y002*4、Y003*4。

*3. 高速输出特殊适配器只可连接A8。

*4. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

要点

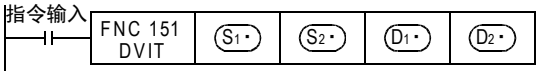
- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

▲2: 使用A8可编程控制器的脉冲输出对象地址中的高速输出特殊适配器时，旋转方向信号请使用下表中的输出。
使用A8/A8C可编程控制器的脉冲输出对象地址中内置的晶体管输出时，旋转方向信号请使用晶体管输出。

高速输出特殊适配器的连接位置	脉冲输出	旋转方向的输出
第1台	(D1•) =Y000	(D2•) =Y004
	(D1•) =Y001	(D2•) =Y005
第2台	(D1•) =Y002	(D2•) =Y006
	(D1•) =Y003	(D2•) =Y007

▲3: D□.b不能变址修饰(V、Z)

功能和动作说明



关于RUN中写入的注意事项

在执行DVIT(FNC 151)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。

根据版本不同的功能变更

根据版本不同，这个指令的功能如下表变更。

→ 有关指令的说明以及功能变更的内容，请参考定位控制手册

对应版本		项目	功能概要
A8	A8C		
Ver.2.20 以上	Ver.1.30 以上	中断输入信号指定功能	M8336为ON后，将Y000～Y003对应的中断输入信号更改成D8336中指定的输入编号(X000～X007)。 但是，使用基本单元的晶体管输出时，不可以指定Y003。
Ver.2.20 以上	Ver.2.20 以上	用户中断模式	在D8336中将Y000～Y003对应的中断输入信号指定为8，M8336为ON后，中断输入信号就变为特殊辅助继电器。用输入中断程序使这个已变更的特殊辅助继电器从OFF变为ON时，开始执行中断动作。 此外，使用了该功能时，不能翻转中断输入的逻辑。 此外，使用基本单元的晶体管输出时，不可以指定Y003。

20.3 FNC 152 - TBL/表格设定定位

A8 A8C

概要

该指令是用HCP-works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。

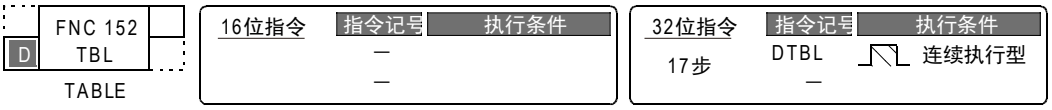
→ 关于指令的说明，请参考定位控制手册

→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

指令	内容
DVIT(FNC 151)*1	中断定位
PLSV(FNC 157)	可变速脉冲输出
DRVI(FNC 158)	相对定位
DRVA(FNC 159)	绝对定位

*1. 仅对应A8/A8C可编程控制器。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D)	指定输出脉冲的输出编号。	位
n	执行的表格编号[1～100]	BIN32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D)		▲1																						
n																				●	●			

▲1: 请指定基本单元的晶体管输出Y000、Y001、Y002，或是高速输出特殊适配器*2Y000、Y001、Y002*3、Y003*3。

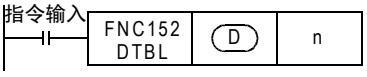
*2. 仅可连接高速输出特殊适配器A8。

*3. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器

要点

- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

功能和动作说明



关于RUN中写入的注意事项

不能对包含TBL(FNC 152)指令的回路块执行RUN中写入。

20.4 FNC 155 - ABS/读出ABS当前值

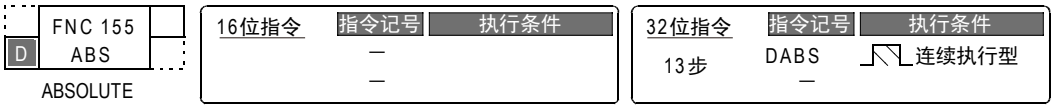
A1 A2 A2C A8 A8C

概要

与本公司的型号为MR-H、MR-J2(S)或MR-J3的伺服放大器(带绝对位置检测功能)连接后，读出绝对位置(ABS)数据的指令。数据以脉冲换算值形式被读出。

→ 关于指令的说明，请参考定位控制手册

1. 指令格式



2. 设定数据

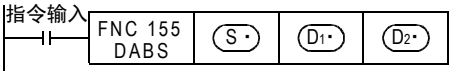
操作数种类	内容	数据类型
(S·)	指定将来自伺服放大器的绝对值(ABS)数据用的输出信号输入进来的软元件起始编号。占用以(S·)开头的3点。	位
(D1·)	指定要将自伺服放大器绝对值(ABS)数据用的输出信号输出的软元件起始编号。占用以(D1·)开头的3点。	
(D2·)	指定保存绝对值(ABS)数据(32位值)的软元件编号。	BIN32位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●	●	●			●	▲2												●					
(D1·)		▲1	●			●	▲2												●					
(D2·)									●	●	●	●	●	●	●	▲3	●	●	●					

▲1: 请指定晶体管输出。
▲2: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲3: 仅对应A8/A8C可编程控制器。

功能和动作说明



20.5 FNC 156 - ZRN/原点回归

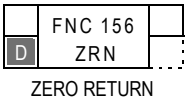
A1 A2 A2C A8 A8C

概要

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

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	ZRN	连续执行型
	—	

32位指令	指令记号	执行条件
17步	DZRN	连续执行型
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1・)	指定开始原点回归时的速度。*1	BIN16/32位
(S2・)	指定爬行速度。[10～32,767(Hz)]	
(S3・)	指定要输入近点信号(DOG)的输入编号的软元件编号。	位
(D・)	指定要输出脉冲的输出编号。	

*1. 设定范围: 16为运算时为10～32,767(Hz)
32位运算时如下所示。

脉冲输出对象		设定范围
A8可编程控制器	高速输出特殊适配器	10～200,000(Hz)
A8/A8C可编程控制器	基本单元(晶体管输出)	10～100,000(Hz)

3. 对象软元件

操作数种类	位软元件							字软元件														其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	▲3	●	●	●	●	●			
(S2・)								●	●	●	●	●	●	●	●	▲3	●	●	●	●	●			
(S3・)	●	●	●			●	▲1												●					
(D・)		▲2																	●					

▲1: D□.b仅对应A8/A8C　　可编程控制器。但是，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，或是高速输出特殊适配器*2 Y000、Y001、Y002*3、Y003*3。

*2. 高速输出特殊适配器仅可连接A8可编程控制器。

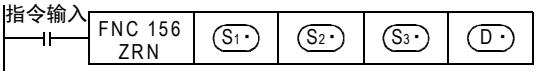
*3. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

要点

- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

▲3: 仅对应A8/A8C可编程控制器。

功能和动作说明



关于RUN中写入的注意事项

在ZRN(FNC 156)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。

根据版本不同的功能变更

根据版本不同，这个指令的功能如下表变更。

→ 有关指令的说明以及功能变更的内容，请参考定位控制手册

对应版本		项目	功能概要
A8	A8C		
Ver.2.20 以上	Ver.2.20 以上	指定清除信号输出对象的功能	(D•) 对应的特殊辅助继电器置ON后，将清除信号的输出对象更改成 (D•) 对应的特殊数据寄存器中指定的输出编号。

20.6 FNC 157 - PLSV/可变速脉冲输出

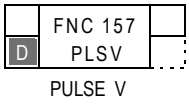
A1 A2 A2C A8 A8C

概要

输出带旋转方向的可变速脉冲的指令。

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	PLSV	连续执行型
	—	

32位指令	指令记号	执行条件
17步	DPLSV	连续执行型
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1•)	指定输出脉冲频率的软元件编号*1	BIN16/32位
(D1•)	指定输出脉冲的输出编号。	位
(D2•)	指定旋转方向信号的输出对象编号。	

*1. 设定范围: 16位运算时，-32,768～-1,+1～32,767(Hz)
32位运算时如下所示。

脉冲输出对象		设定范围
A8可编程控制器	高速输出特殊适配器	-200,000～-1，+1～200,000(Hz)
A8/A8C可编程控制器	基本单元(晶体管输出)	-100,000～-1，+1～100,000(Hz)

3. 对象软元件

操作数种类	位软元件							字软元件											其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲4	●	●	●	●	●			
(D1•)		▲1																	●					
(D2•)		▲2	●			●	▲3												●					

▲1: 请指定基本单元的晶体管输出Y000、Y001、Y002，或是高速输出特殊适配器*2Y000、Y001、Y002*3、Y003*3。

*2. 高速输出特殊适配器只可连接可编程控制器A8。

*3. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

要点

- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

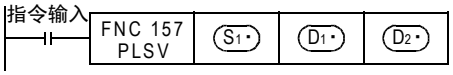
▲2: 使用A8可编程控制器的脉冲输出对象地址中的高速输出特殊适配器时，旋转方向信号请使用下表中的输出。
在使用A8/A8C可编程控制器的脉冲输出对象地址中内置的晶体管输出时，旋转方向信号请使用晶体管输出。

高速输出特殊适配器的连接位置	脉冲输出	旋转方向的输出
第1台	(D1•) =Y000	(D2•) =Y004
	(D1•) =Y001	(D2•) =Y005
第2台	(D1•) =Y002	(D2•) =Y006
	(D1•) =Y003	(D2•) =Y007

▲3: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。

▲4: 仅对应A8/A8C可编程控制器。

功能和动作说明



关于RUN中写入的注意事项

在执行PLSV(FNC 157)指令的过程中(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，请注意脉冲输出会如下所示变化。

	指令动行中执行RUN中写入时的动作
带加减速动作时	脉冲输出减速停止。
无加减速动作时	即时停止脉冲输出。

根据版本不同的功能变更

根据版本不同，这个指令的功能如下表变更。

→ 有关指令的说明以及功能变更的内容，请参考定位控制手册

对应版本		追加功能	功能概要
A8	A8C		
Ver.2.20 以上	Ver.2.20 以上	加减速动作功能	当M8338为ON后， (S1·) 发生变化时，根据 (D1·) 对应的加速时间、减速时间的设定，会加速或减速到 (S1·) 。

20.7 FNC 158 - DRVI/相对定位

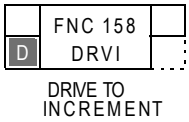
A1 A2 A2C A8 A8C

概要

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

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
9步	DRVI	连续执行型	17步	DDRVI	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S1•)	指定输出脉冲数(相对地址)。*1	BIN16/32位
(S2•)	指定输出脉冲频率。*2	
(D1•)	指定输出脉冲的输出编号。	位
(D2•)	指定旋转方向信号的输出对象编号。	

*1. 设定范围: 16位运算时为-32,768～+32,767(0除外)
32位运算时为-999,999～+999,999(0除外)

*2. 设定范围: 16为运算时为10～32,767(Hz)
32位运算时如下所示。

脉冲输出对象		设定范围
A8可编程控制器	高速输出特殊适配器	10～200,000(Hz)
A8/A8C可编程控制器	基本单元(晶体管输出)	10～100,000(Hz)

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲4	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲4	●	●	●	●	●			
(D1•)		▲1																	●					
(D2•)		▲2	●			●	▲3												●					

▲1: 请指定基本单元的晶体管输出Y000、Y001、Y002，或是高速输出特殊适配器*3Y000、Y001、Y002*4、Y003*4。

*3. 高速输出特殊适配器只可连接A8可编程控制器。

*4. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

要点

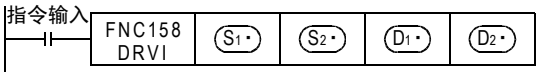
- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

▲2: 使用A8可编程控制器的脉冲输出对象地址中的高速输出特殊适配器时，旋转方向信号请使用下表中的输出。
在使用A8/A8C可编程控制器的脉冲输出对象标地址中内置的晶体管输出时，旋转方向信号请使用晶体管输出。

高速输出特殊适配器的连接位置	脉冲输出	旋转方向的输出
第1台	(D1•) =Y000	(D2•) =Y004
	(D1•) =Y001	(D2•) =Y005
第2台	(D1•) =Y002	(D2•) =Y006
	(D1•) =Y003	(D2•) =Y007

▲3: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲4: 仅对应A8/A8C可编程控制器。

功能和动作说明



关于RUN中写入的注意事项

在执行DRVI(FNC 58)指令的过程中(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。

20.8 FNC 159 - DRVA/绝对定位

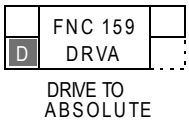
A1 A2 A2C A8 A8C

概要

以绝对驱动方式执行单速定位的指令。用指定从原点(零点)开始的移动距离的方式，也称为绝对驱动方式。

→ 关于指令的说明，请参考定位控制手册
→ 关于使用高速输出特殊适配器的注意事项，请参考定位控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	DRVA	连续执行型

32位指令	指令记号	执行条件
17步	DDRVA	连续执行型

2. 设定数据

操作数种类	内容	数据类型
(S1•)	指定输出脉冲数(绝对地址)。*1	BIN16/32位
(S2•)	指定输出脉冲频率。*2	
(D1•)	指定输出脉冲的输出编号。	位
(D2•)	指定旋转方向信号的输出对象编号。	

*1. 设定范围: 16位运算时为-32,768～+32,767
32位演算时为-999,999～+999,999

*2. 设定范围: 16为运算时为10～32,767(Hz)
32位运算时如下所示。

脉冲输出对象		设定范围
A8可编程控制器	高速输出特殊适配器	10～200,000(Hz)
A8/A8C可编程控制器	基本单元(晶体管输出)	10～100,000(Hz)

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲4	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲4	●	●	●	●	●			
(D1•)		▲1																	●					
(D2•)		▲2	●			●	▲3												●					

▲1: 请指定基本单元的晶体管输出Y000、Y001、Y002，或是高速输出特殊适配器*3Y000、Y001、Y002*4、Y003*4。

*3. 高速输出特殊适配器仅可连接A8可编程控制器。

*4. 使用高速输出特殊适配器的Y002、Y003时，需要使用第2台的高速输出特殊适配器。

要点

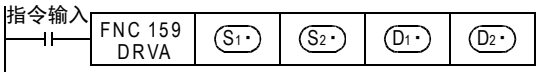
- 使用继电器输出型的A8可编程控制器时，需要使用高速输出特殊适配器。
- 高速输出特殊适配器的输出为差动线性驱动。

▲2: 使用A8可编程控制器的脉冲输出对象地址中的高速输出特殊适配器时，请指定下表中的输出。
在使用A8/A8C可编程控制器的脉冲输出对象地址中内置的晶体管输出时，旋转方向信号请使用晶体管输出。

高速输出特殊适配器的连接位置	脉冲输出	旋转方向的输出
第1台	(D1•) =Y000	(D2•) =Y004
	(D1•) =Y001	(D2•) =Y005
第2台	(D1•) =Y002	(D2•) =Y006
	(D1•) =Y003	(D2•) =Y007

▲3: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)。
▲4: 仅对应A8/A8C可编程控制器。

功能和动作说明

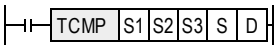
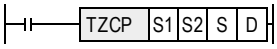
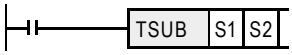
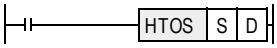
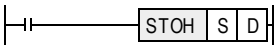
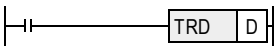
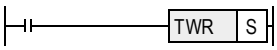


关于RUN中写入的注意事项

在执行DRVA(FNC 59)指令的过程中(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。

21. 时钟运算—FNC 160～FNC 169

在FNC 160～FNC 169中，针对时钟数据进行运算、比较的指令。
此外，还可以执行可编程控制器内置实时时钟的时间校准以及时间数据的格式转换。

FNC No.	指令记号	符号	功能	参考
160	TCMP		时钟数据比较	21.1节
161	TZCP		时钟数据区间比较	21.2节
162	TADD		时钟数据加法运算	21.3节
163	TSUB		时钟数据减法运算	21.4节
164	HTOS		时、分、秒数据的秒转换	21.5节
165	STOH		秒数据的[时、分、秒]转换	21.6节
166	TRD		读出时钟数据	21.7节
167	TWR		写入时钟数据	21.8节
168	—			—
169	HOUR		计时表	21.9节

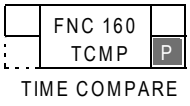
21.1 FNC 160 - TCMP/时钟数据比较

A1 A2 A2C A8 A8C

概要

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

1. 指令格式



16位指令	指令记号	执行条件
11步	TCMP	连续执行型
	TCMPP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

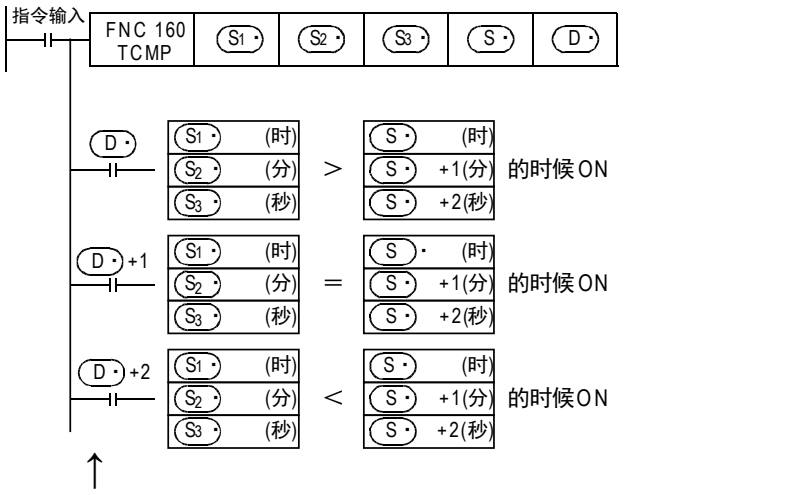
操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S3•)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(S•)												●	●	●	●	▲2			●					
(D•)		●	●			●	▲1												●					

▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(TCMP)

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



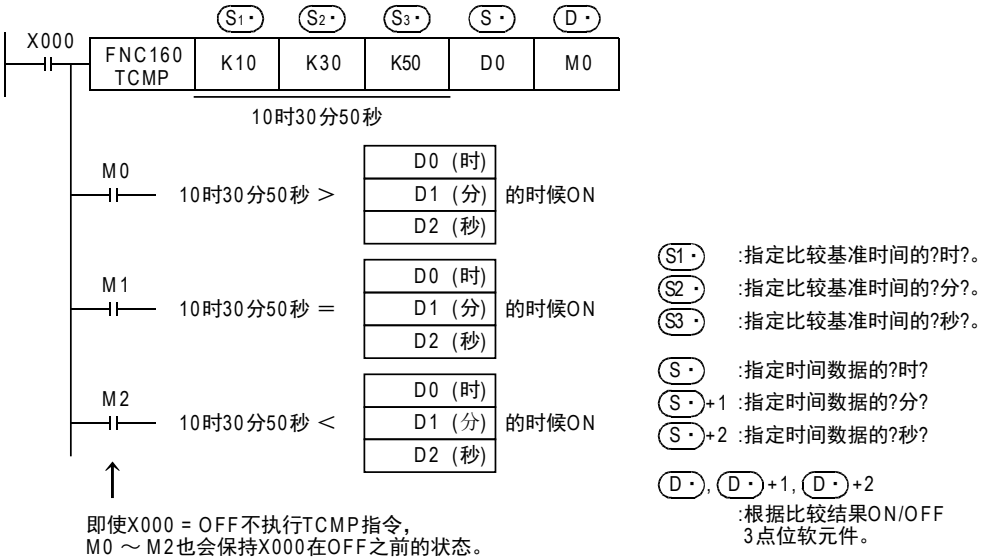
由于指令触点由ON变为OFF，TCMP指令不被执行，即使如此，(D・)，(D・)+1，(D・)+2也会保持指令触点OFF之前的状态。

注意要点

1) 软元件的占用点数
(S・)、(D・)各占用3点软元件。
请注意不要与机器其他控制中使用的软元件重复。

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

程序举例



即使X000 = OFF不执行TCMP指令，
M0 ～ M2也会保持X000在OFF之前的状态。

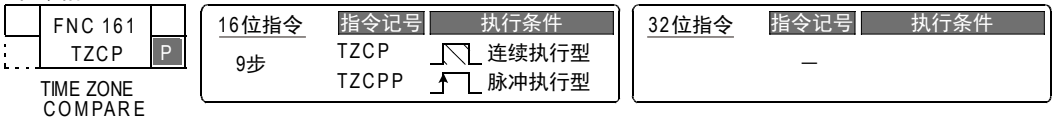
21.2 FNC 161 - TZCP/时钟数据区间比较

A1 A2 A2C A8 A8C

概要

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

1. 指令格式



2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)												●	●	●	●	▲2			●					
(S2·)												●	●	●	●	▲2			●					
(S·)												●	●	●	●	▲2			●					
(D·)		●	●			●	▲1												●					

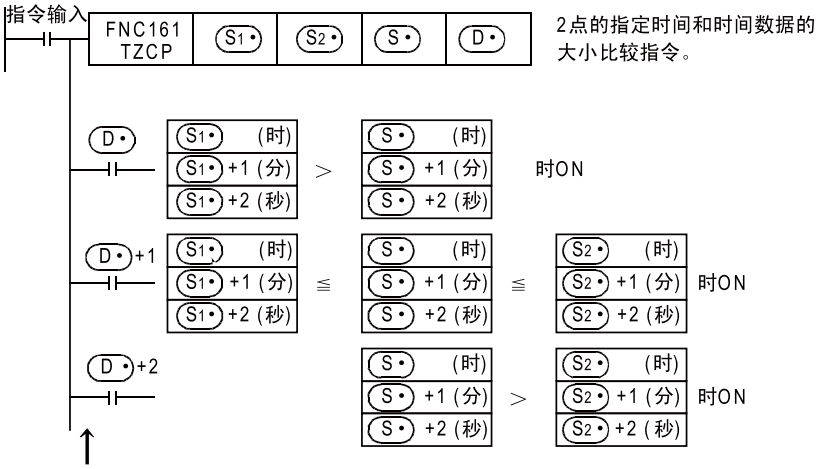
▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)

▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(TZCP)

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



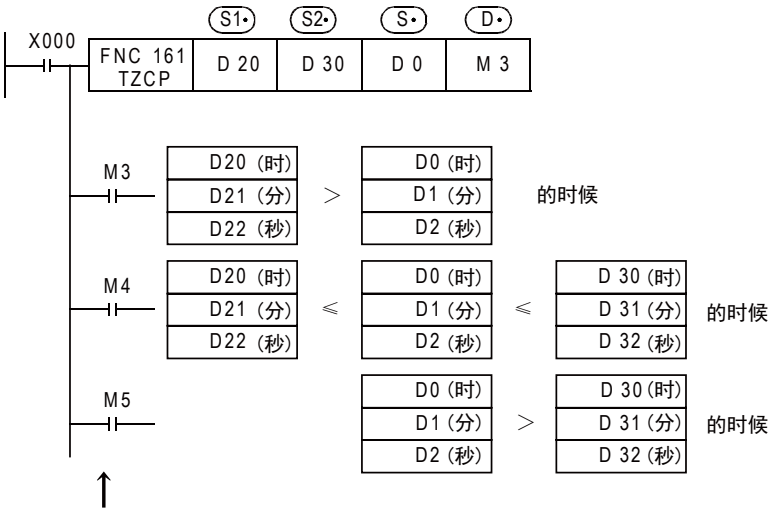
由于指令触点由ON变为OFF，TCMP指令不被执行，即使如此，(D·)，(D·)+1，(D·)+2 也会保持指令触点 OFF 之前的状态。

注意要点

1) 软元件的占用点数
(S1・)、(S2・)、(S3・)、(D・)各占用3点软元件。
请注意不要与机器其他控制中使用的软元件重复。

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

程序举例



即使由于X000而不执行TCMP指令，
M3～M5也会保持X000在OFF之前的状态。

(S1・)，(S1・)+1,(S1・)+2：以“时”、“分”、“秒”指定比较基准时间的下线。

(S2・)，(S2・)+1,(S2・)+2：以“时”、“分”、“秒”指定比较基准时间的上线。

(S・)，(S・)+1,(S・)+2：以“时”、“分”、“秒”指定时间数据。

(D・)，(D・)+1,(D・)+2：根据比较结果，控制这3点位软元件的ON/OFF。

“时”的范围为「0～23」
“分”的范围为「0～59」
“秒”的范围为「0～59」

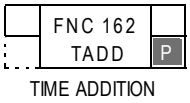
21.3 FNC 162 - TADD/时钟数据加法运算

A1 A2 A2C A8 A8C

概要

将2个时间数据进行加法运算后，保存在字软元件中。

1. 指令格式



16位指令	指令记号	执行条件
7步	TADD	连续执行型
	TADDP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

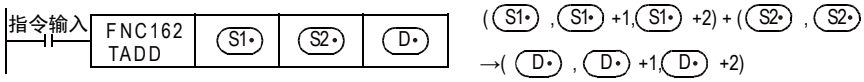
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)												●	●	●	●	▲			●					
(S2·)												●	●	●	●	▲			●					
(D·)												●	●	●	●	▲			●					

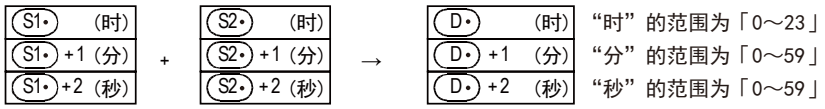
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(TADD)

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





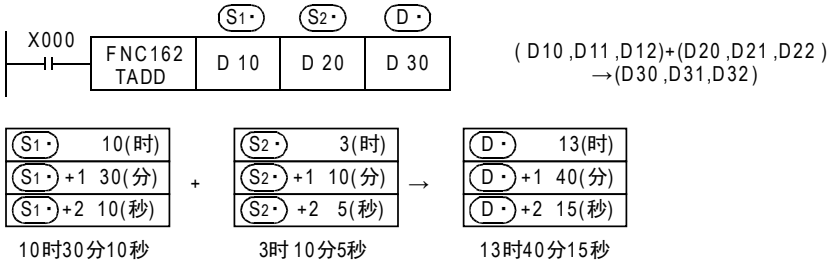
- 当运算结果超出24小时时，进位标志位变为ON，从单纯的加法运算值中减去24个小时后将该时间作为运算结果被保存。
- 运算结果为0(0时0分0秒)时，零位标志位变为ON。

注意要点

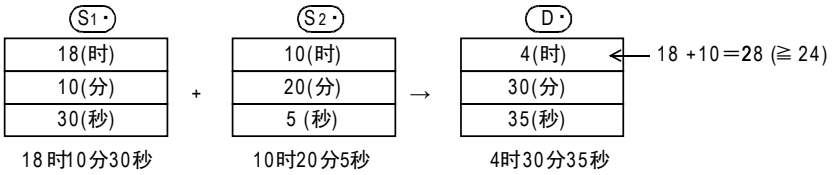
1) 软元件的占用点数
(S1·)、(S2·)、(D·)各占用3点软元件。
请注意不要与机器其他控制中使用的软元件重复。

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

程序举例



当运算结果超出24小时时



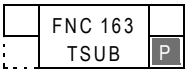
21.4 FNC 163 - TSUB/时钟数据减法运算

A1 A2 A2C A8 A8C

概要

将2个时间数据进行减法运算后，保存在字软元件中。

1. 指令格式



16位指令	指令记号	执行条件
7步	TSUB	连续执行型
	TSUBP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

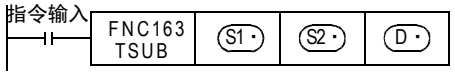
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)												●	●	●	●	▲			●					
(S2·)												●	●	●	●	▲			●					
(D·)												●	●	●	●	▲			●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(TSUB)

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



(S1·), (S1·)+1 (S1·) +2)-((S2·), (S2·)+1,(S2·)+2)
→ ((D·), (D·)+1,(D·)+2)

(S1·)	(时)	-	(S2·)	(时)	→	(D·) (时)
(S1·)+1	(分)		(S2·)+1	(分)		(D·)+1 (分)
(S1·)+2	(秒)		(S2·)+2	(秒)		(D·)+2 (秒)

“时”的范围为「0～23」
“分”的范围为「0～59」
“秒”的范围为「0～59」

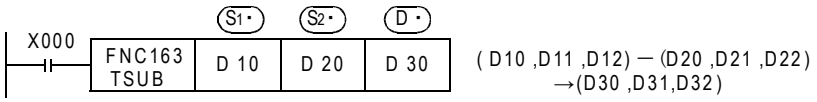
当运算结果小于0时，借位标志位变为ON，从单纯的减法运算值中加上24个小时后，将该时间作为运算结果被保存。运算结果为0(0时0分0秒)时，零位标志位变为ON。

注意要点

1) 软元件的占用点数
(S1·)、(S2·)、(D·)各占用3点软元件。
请注意不要与机器其他控制中使用的软元件重复。

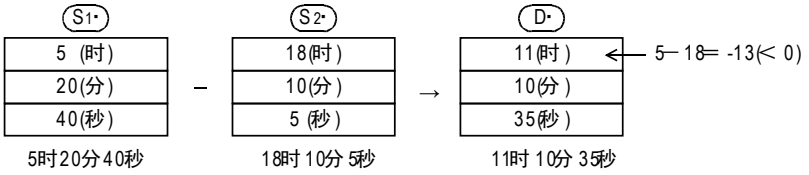
2) 使用可编程控制器内置实时时钟的时钟数据的时间(时、分、秒)时
请使用TRD(FNC 166)指令，读出特殊数据寄存器的值以后，在各个操作数中指定其字软元件。

程序举例





当运算结果小于0小时时



21.5 FNC 164 - HTOS/时、分、秒数据的秒转换

A8 A8C

概要

将[时、分、秒]单位的时间(时刻)数据转换成秒单位的数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	HTOS	连续执行型
	HTOSP	脉冲执行型

32位指令	指令记号	执行条件
9步	DHTOS	连续执行型
	DHTOSP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存转换前的时间(时刻)数据(时、分、秒)的软元件的起始编号	BIN16位
(D·)	保存转换后的时间(时刻)数据(秒)的软元件编号。	BIN32/16位

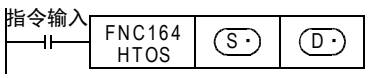
3. 对象软元件

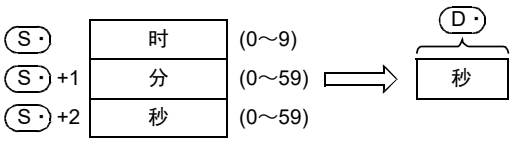
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●			●					
(D·)									●	●	●	●	●	●	●	●			●					

功能和动作说明

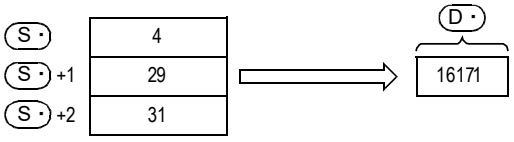
1. 16位运算(HTOS/HTOSP)

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。



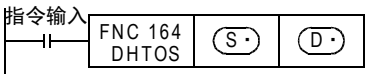


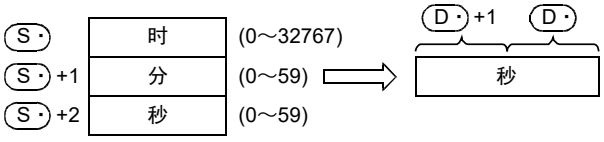
例如，指定了4时29分31秒时，如下所示。



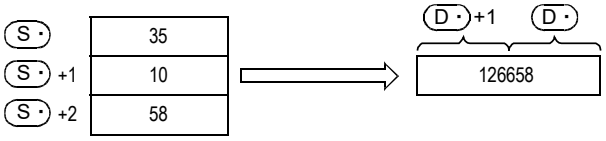
2. 32位运算(DHTOS/DHTOSP)

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。





例如，指定了35时10分58秒时，如下所示。



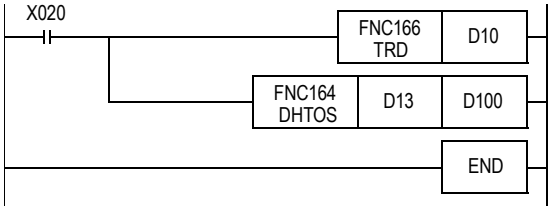
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- (S・)、(S・)+1、(S・)+2的数据超出范围时。(错误代码:K6706)

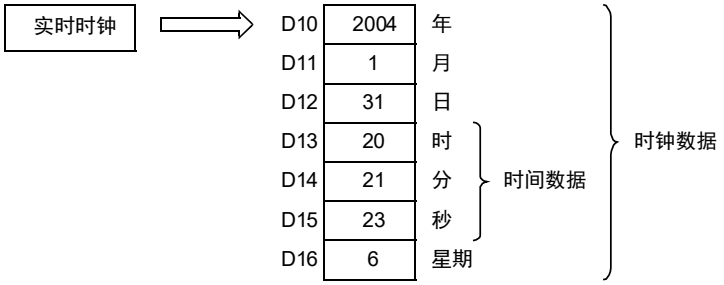
程序举例

当X020为ON时，从可编程控制器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中的程序。

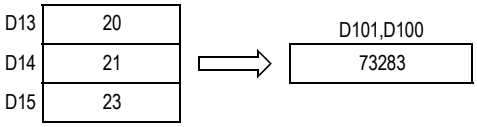


动作

- 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



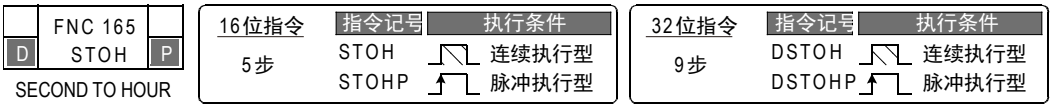
21.6 FNC 165 - STOH/秒数据的[時、分、秒]转换

A8 A8C

概要

将秒单位的时间(时刻)数据转换成[时、分、秒]单位的数据的指令。

1. 指令格式



2. 设定数据

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

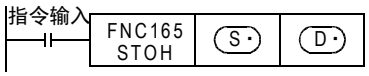
3. 对象软元件

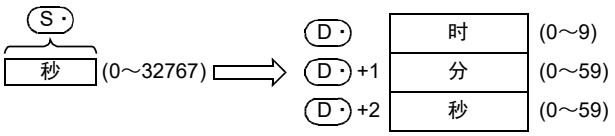
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●			●					
(D·)									●	●	●	●	●	●	●	●			●					

功能和动作说明

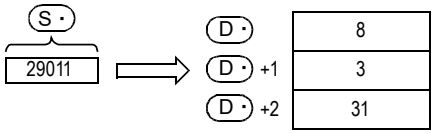
1. 16位运算

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。



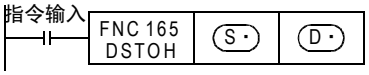


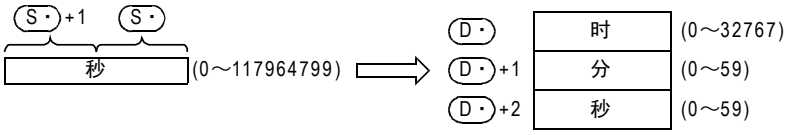
例如，指定了29011秒时，如下所示。



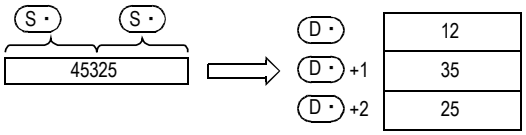
2. 32位运算

将[(S・)+1，(S・)]中的秒数据换算成时、分、秒，其结果保存到以[(D・)，(D・)+1，(D・)+2]开头的3点(时、分、秒)中。





例如，指定了45325秒时，如下所示。



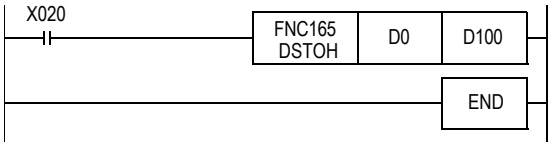
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- (S・)数据超出范围时。(错误代码: K6706)

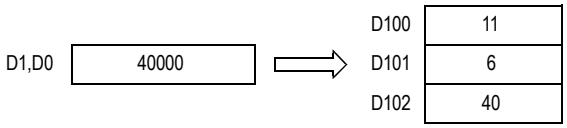
程序举例

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)



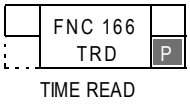
21.7 FNC 166 - TRD/读出时钟数据

A1 A2 A2C A8 A8C

概要

读出可编程控制器内置实时时钟的时钟数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
3步	TRD	连续执行型
	TRDP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

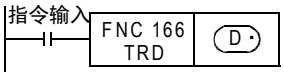
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)												●	●	●	●	▲			●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算(TRD)

将可编程控制器内置实时时钟的时钟数据(D8013～D8019)按照下面的格式读出到(D·)～(D·)+6中。



将可编程控制器的实时时钟数据读出到7点数据寄存器中的指令。

	软元件	项目	时钟数据		软元件	项目
特殊数据寄存器	D8018	年(公历)	0～99(公历后 2 位数)	→	D0	年(公历)
	D8017	月	1～12	→	D1	月
	D8016	日	1～31	→	D2	日
	D8015	时	0～23	→	D3	时
	D8014	分	0～59	→	D4	分
	D8013	秒	0～59	→	D5	秒
	D8019	星期	0(日)～6(六)	→	D6	星期

注意要点

1. 软元件的占用点数

(D·) 占用7点软元件。

请注意不要与机器其他控制中使用的软元件重复。

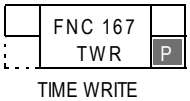
21.8 FNC 167 - TWR/写入时钟数据

A1 A2 A2C A8 A8C

概要

向可编程控制器内置实时时钟写入时钟数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
3步	TWR	连续执行型
	TWRP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

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

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●	▲			●					

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

将设定的时钟数据(S・)～(S・)+6写入可编程控制器内置实时时钟的时钟数据(D8013～D8019)中。



－ D8018(年数据)，也可以切换到4位数模式。(参考程序举例)

	软元件	项目	时钟数据		软元件	项目	
设定时间用的数据	D10	年(公历)	0～99 (公历后 2 位数)	→	D8018	年(公历)	特殊数据寄存器
	D11	月	1～12	→	D8017	月	
	D12	日	1～31	→	D8016	日	
	D13	时	0～23	→	D8015	时	
	D14	分	0～59	→	D8014	分	
	D15	秒	0～59	→	D8013	秒	
	D16	星期	0(日)～6(六)	→	D8019	星期	

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

－使用这个指令设定时钟数据(时间校准)时，不需要控制特殊辅助继电器M8015(时间停止以及时间校准)。

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

注意要点

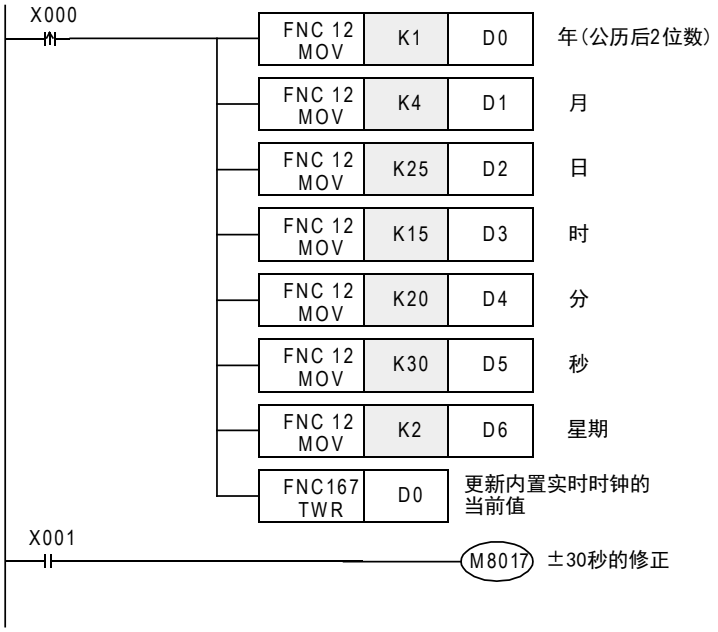
1. 软元件的占用点数

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

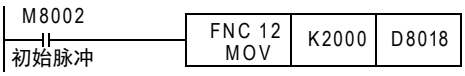
程序举例

1. 时钟数据(时刻)的设定实例

设定实时时钟。《2001年4月25日(星期二)15时20分30秒的情况下》



- 阴影部分为各个项目的设定值。
- 设定时间的时候，设定快几分钟的时间，等到达到正确时间时使X000置ON，在实时时钟中写入已设定的时间，时钟数据被更新。
- X001每次为ON时，可以执行±30秒的修正。
- 将年作为公历4位数处理时，请增加如下所示的程序。
  D8018在可编程控制器RUN后第2个运算周期以后，作为公历4位数动作。



－可编程控制器通常在公历后2位数的模式下运行。在可编程控制器 RUN后，执行上述指令时，仅在1个运算周期内，向D8018(年)传送[K2000(固定值)]，据此切换成4位数模式。

－ 请在每次运行可编程控制器时执行这个程序。此外，即使传送K2000也只是显示切换成4位数，而对当前的日期时间没有影响。

－ 公历4位数模式的情况下，设定值[80～99]相当于[1980～1999年]，[00～79]相当于[2000～2079年]。
例如: 80=1980年，99=1999年，00=2000年，79=2079年

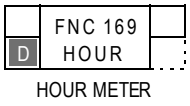
21.9 FNC 169 - HOUR/计时表

A1 A2 A2C A8 A8C

概要

以1个小时为单位，对输入触点持续ON的时间进行累加检测的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	HOUR	连续执行型

32位指令	指令记号	执行条件
13步	DHOUR	连续执行型

2. 设定数据

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

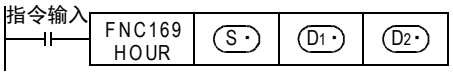
3. 对象软元件

操作数种类	位软元件							字软元件														其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲2	●	●	●	●	●			
(D1・)														●	●				●					
(D2・)		●	●			●	▲1												●					

▲1: D□.b仅对应A8/A8C可编程控制器。但是，不能变址修饰(V、Z)
▲2: 仅对应A8/A8C可编程控制器。

功能和动作说明

1. 16位运算

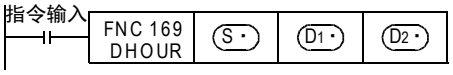


当指令输入的累计ON时间超出了(S・)时，(D2・)变为ON。
(D1・)+1中不满1个小时的当前值，以1秒单位被保存。

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

- 由于即使断开可编程控制器的电源后，也可以继续使用当前值数据，所以请在(D1・)中指定停电保持用的数据寄存器。
  使用一般的数据寄存器时，由于可编程控制器的电源OFF和STOP→RUN的操作，当前值会被清除。
- 报警输出(D2・)为ON以后，测量仍能继续。
- 当前值(D1・)达到16位的最大值时停止测量。
  要继续测量时，请清除(D1・)～(D1・)+1的当前值。

2. 32位运算



[(S·)+1，(S·)]：(D2·)变为ON为止的时间指定用
(S1·)+1(高位)，(S1·)(低位)指定。
[(D1·)+1，(D1·)]：以1个小时为单位的当前值保存在
(D1·)+1(高位)，(D1·)(低位)中。
(D1·)+2 ：不满1个小时的当前值(1秒单位)
(D2·) ：报警输出的指定
当前值(D1·)，(D1·)+1超出(S·)的指定时间时，置ON。

- 由于即使断开可编程控制器的电源后，也可以继续使用当前值数据，所以请在(D1·)中指定停电保持用的数据寄存器。
  使用一般的数据寄存器时，由于可编程控制器的电源OFF和STOP→RUN的操作，当前值会被清除。
- 报警输出(D2·)为ON以后，测量仍能继续。
- 当前值[(D1·)+1，(D1·)]达到32位的最大值时停止测量。
  要继续测量时，请清除(D1·)～(D1·)+2的当前值。

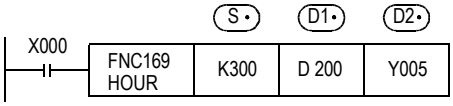
注意要点

软元件的占用点数

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

程序举例

在X000的ON累计时间超出300个小时的时候，Y005为ON。
在D201中以1秒为单位，保存不满1个小时的当前值。



(S·) ：(D2·)变ON的时间指定
以1个小时为单位指定。
(D1·) ：以1个小时为单位的当前值
(D1·)+1 ：不满1个小时的当前值(1秒单位)
(D2·) ：报警输出的指定
当前值(D1·)超出(S·)的指定时间时，置ON。
(在这个例子中300个小时+1秒时为ON)

22. 外部设备—FNC 170～FNC 179

在FNC 170～FNC 179中，提供了在绝对型(绝对位置)的旋转编码器中使用的格雷码转换指令以及模拟量模块的读写专用指令。

FNC No.	指令记号	符号	功能	参考
170	GRY	GRY S D	格雷码的转换	22.1节
171	GBIN	GBIN S D	格雷码的逆转换	22.2节
172	—			—
173	—			—
174	—			—
175	—			—
176	RD3A	RD3A m1 m2 D	模拟量模块的读出	22.3节
177	WR3A	WR3A m1 m2 S	模拟量模块的写入	22.4节
178	—			—
179	—			—

22.1 FNC 170 - GRY/格雷码的转换

A8 A8C

概要

将BIN值转换成格雷码后进行传送的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	GRY	连续执行型
	GRYP	脉冲执行型

32位指令	指令记号	执行条件
9步	DGRY	连续执行型
	DGRYP	脉冲执行型

2. 设定数据

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

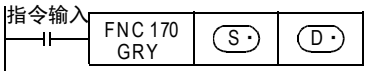
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D·)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

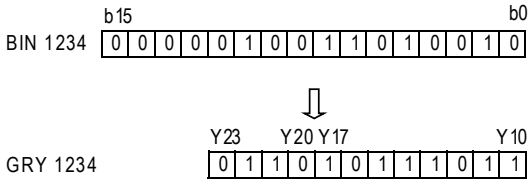
功能和动作说明

1. 16位运算(GRY、GRYP)



源数据(BIN)→目标数据(GRY)的转换传送指令。

(S·)为K1234, (D·)为K3Y10时



— 对(S·)而言，下面的数值范围为有效的。
0～32,767

2. 32位运算(DGRY、DGRYP)

— 可以执行最大32位的格雷码转换。

— (S·)为0～2,147,483,647时有效。

注意要点

数据的转换速度取决于可编程控制器的扫描时间。

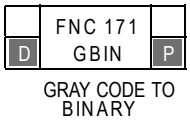
22.2 FNC 171 - GBIN/格雷码的逆转换

A8 A8C

概要

将格雷码转换成BIN值后进行传送的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	GBIN	连续执行型
	GBINP	脉冲执行型

32位指令	指令记号	执行条件
9步	DGBIN	连续执行型
	DGBINP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存转换源数据的字软元件	BIN16/32位
(D・)	保存转换后数据的字软元件	BIN16/32位

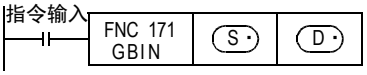
3. 对象软元件

操作数种类	位软元件							字软元件														其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(D・)									●	●	●	●	●	●	●	▲	●	●	●					

▲: 仅对应A8/A8C可编程控制器。

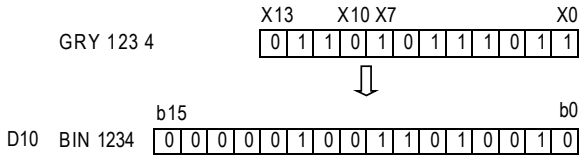
功能和动作说明

1. 16位运算(GBIN、GBINP)



源数据(GRY)→目标数据(BIN)的转换传送指令。

(S・)为K3X000, (D・)为10时



— 在使用格雷码方式的编码器检测绝对位置等情况下，可以使用。

— 对(S・)而言，下面的数值范围为有效的。

0～32,767

2. 32位运算(DGBIN、DGBINP)

— 可以执行最大32位的BIN转换。

— (S・)为0～2,147,483,647时有效。

注意要点

(S・)中指定输入继电器(X)时的响应延迟，为[可编程控制器的扫描时间+输入滤波常数]。

通过执行REFF(FNC 51)指令或D8020(滤波器的调节)，转换X000～X017的输入滤波值，可以去除滤波器常数部分的延迟。

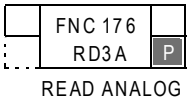
22.3 FNC 176 - RD3A/模拟量模块的读出

A8 A8C

概要

读取TX0N-3A以及TX2N-2AD模拟量模块的模拟量输入值的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	RD3A	连续执行型
	RD3A P	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

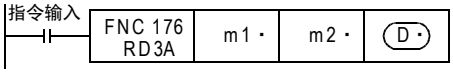
操作数种类	内容	数据类型
m1・	特殊模块编号 • A8/A8C(D、DSS)系列: K0～K7 • A8C-32MT-LT(-2): K1～K7	BIN16位
m2・	模拟量输入通道编号	BIN16位
(D・)	保存读出的数据的字软元件	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他					
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针	
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P	
m1・								●	●	●	●	●	●	●	●		●	●	●	●	●				
m2・								●	●	●	●	●	●	●	●		●	●	●	●	●				
(D・)									●	●	●	●	●	●	●		●	●	●						

功能和动作说明

1. 16位运算(RD3A)



m1・ : 特殊模块编号
A8/A8C(D、DSS)系列: K0～K7
A8C-32MT-LT(-2): K1～K7(K0为内置HC-Link/LT主站。)

m2・ : 模拟量输入通道编号
TX0N-3A*1 : K1(通道1)，K2(通道2)
TX2N-2AD : K21(通道1)，K22(通道2)

(D・) : 读出数据
保存从模拟量模块中读出的数值。
TX0N-3A*1 : 0～255(8位)
TX2N-2AD : 0～4095(12位)

*1. TX0N-3A仅对应A8/A8C可编程控制器。

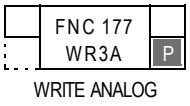
22.4 FNC 177 - WR3A/模拟量模块的写入

A8 A8C

概要

向TX0N-3A以及TX2N-2DA模拟量模块写入数字值的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	WR3A	连续执行型
	WR3AP	脉冲执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

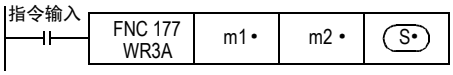
操作数种类	内容	数据类型
m1・	特殊模块编号 • A8/A8C(D、DSS)系列: K0～K7 • A8C-32MT-LT(-2): K1～K7	BIN16位
m2・	模拟量输入通道编号	BIN16位
(S・)	写入的数据，或是保存写入数据的字软元件	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
m1・								●	●	●	●	●	●	●	●		●	●	●	●	●			
m2・								●	●	●	●	●	●	●	●		●	●	●	●	●			
(S・)									●	●	●	●	●	●	●		●	●	●					

功能和动作说明

1. 16位运算(WR3A)



m1・ ： 特殊模块编号
A8/A8C(D、DSS)系列: K0～K7
A8C-32MT-LT(-2): K1～K7(K0为内置HC-Link/LT主站。)

m2・ ： 模拟量输出通道编号
TX0N-3A*1 : K1(通道1)
TX2N-2DA : K21(通道1)，K22(通道2)

(S・) ： 读入数据
指定输出到模拟量模块的数值。
TX0N-3A*1 : 0～255(8位)
TX2N-2DA : 0～4095(12位)

*1. TX0N-3A仅对应A8/A8C可编程控制器。

23. 替换指令的介绍—FNC 180

23.1 指令替换对照表

概要

EXTR指令是TX2N、TX2NC可编程控制器使用的指令。
A8/A8C可编程控制器内置了变频器通信功能，提供了以下的专用指令。
(不提供EXTR指令。)

指令替换对照表

A8A/8C可编程控制器		A8/A8C可编程控制器		内容
EXTR K10	→	FNC 270	IVCK	变换器的运转监视
EXTR K11	→	FNC 271	IVDR	变频器的运行控制
EXTR K12	→	FNC 272	IVRD	变频器的参数读取
EXTR K13	→	FNC 273	IVWR	变频器的参数写入
—		FNC 274	IVBWR*1	变频器的参数成批写入

*1. 仅对应A8/A8C可编程控制器。

→ 详细内容，请参考通信控制手册

24. 其他指令—FNC 181～FNC 189

在FNC 181～FNC 189中，提供了用于产生随机数、CRC数据运算、高速计数器运算的数据处理的指令。

FNC No.	指令记号	符号	妹撽	嵩徠
181	—			—
182	COMRD	COMRD S D	读出软元件的注释数据	24.1节
183	—			—
184	RND	RND D	产生随机数	24.2节
185	—			—
186	DUTY	DUTY n1 n2 D	产生定时脉冲	24.3节
187	—			—
188	CRC	CRC S D n	CRC运算	24.4节
189	HCMOV	HCMOV S D n	高速计数器的传送	24.5节

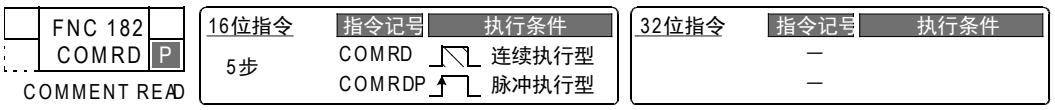
24.1 FNC 182 - COMRD/读出软元件的注释数据

A8 A8C

概要

该指令是将用HCP-works等编程软件登录(写入)的软元件的注释数据读出来。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	登录了要读出注释的软元件编号	软元件名
(D·)	保存已经读出注释的软元件起始编号	字符串

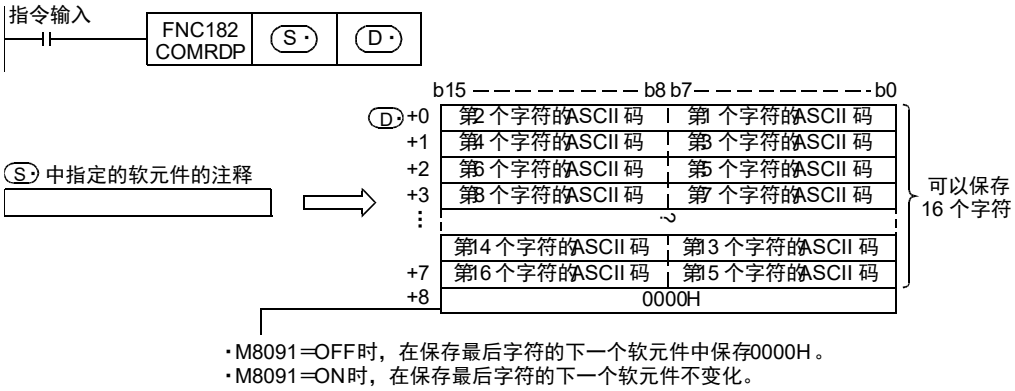
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)	●	●	●			●						●	●	●	●				●					
(D·)												●	●	●	●				●					

功能和动作说明

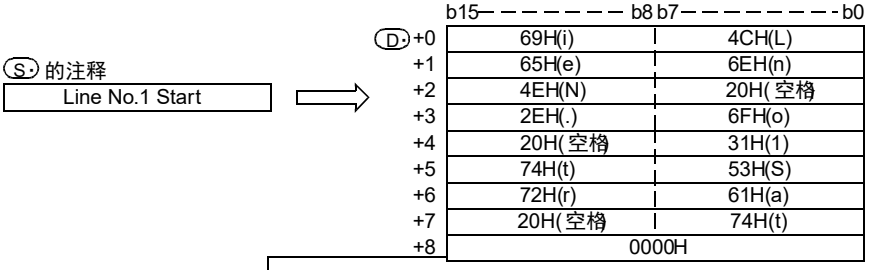
1. 16位运算(COMRD/COMRDP)

1) 读出软元件(S·)的注释后，以ASCII码保存在(D·)开始的软元件中。



·M8091=OFF时，在保存最后字符的下一个软元件中保存0000H。
·M8091=ON时，在保存最后字符的下一个软元件不变化。

例如，软元件(S·)的注释为“Line NO.1 Start”时，从(D·)开始的软元件中保存如下所示的内容。



·M8091=OFF时，在保存最后字符的下一个软元件中保存0000H。
·M8091=ON时，在保存最后字符的下一个软元件不变化。

2) 根据M8091的ON/OFF状态，(D・) +8的数据如下所示。

ON/OFF状态	处理内容
M8091=OFF	在保存最后注释字符的软元件的下一个软元件中保存0000H。
M8091=ON	保存最后注释字符的软元件的下一个软元件不变化。

相关软元件

软元件	名称	内容
M8091	输出字符数切换信号	参考上述的说明

注意要点

- 请在(S・)的软元件中，指定在可编程控制器中登录(写入)了注释的软元件编号。
  在软元件(S・)中未登录(写入)注释时，注释的字符数(半角16个字符)全部都以“20H”(空格)保存到(D・)开始的软元件中。

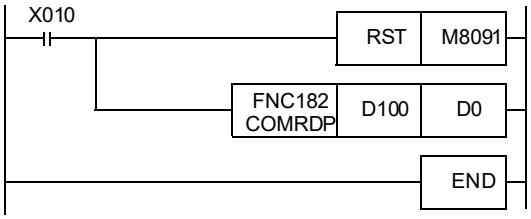
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 软元件(S・)中未登录(写入)注释时。(错误代码:K6706)
- 当(D・)开始的相当于注释长度的软元件点数超出了相应的软元件范围时。但是，写入一部分注释。(错误代码:K6706)

程序举例

X010为ON后，D100中设定的注释“Target Line A”会以ASCII码保存到从D0开始的软元件中的程序。
(M8091＝OFF时)



D100的注释
Target Line A

	b15 ---- b8	b7 ---- b0
D0	61H(a)	54H(T)
D1	67H(g)	72H(r)
D2	74H(t)	65H(e)
D3	4CH(L)	20H(空格)
D4	6EH(n)	69H(i)
D5	20H(空格)	65H(e)
D6	20H(空格)	41H(A)
D7	20H(空格)	20H(空格)
D8	0000H	

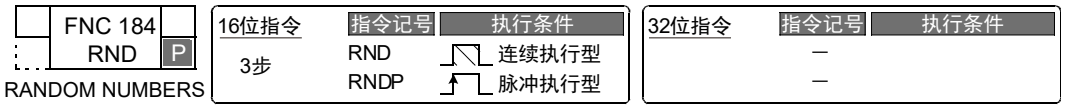
24.2 FNC 184 - RND/产生随机数

A8 A8C

概要

产生随机数的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(D·)	保存随机数的软元件起始编号	BIN16位

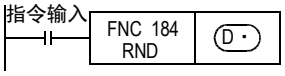
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(D·)									●	●	●	●	●	●	●	●			●					

功能和动作说明

1. 16位运算(RND/RNDP)

这个指令产生0～32767的伪随机数，将其数值作为随机数保存到(D·)中。
在伪随机数系列中，每次计算出随机数的原始值，然后使用这随机数的原始值计算出伪随机数。



伪随机数的计算公式:

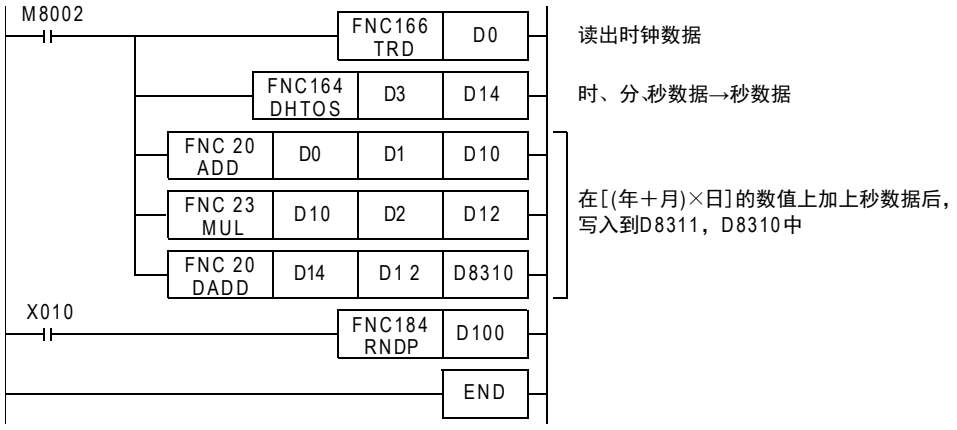
(D8311，D8310)＝(D8311，D8310)*1×1103515245＋12345…①

(D·)=([D8311，D8310]>>16)&<逻辑与>00007FFFh

*1. 在这个 (D8311，D8310) 中，请在 STOP → RUN 时仅仅写入一次非负的数值 (0 ～ 2,147,483,647)。
[(D8311，D8310)作为初始值，在上电时写入K1。]

程序举例

X010每次为ON时，在D100中保存随机数的程序。
从STOP→RUN时，将时间数据进行秒转换后的结果，与[(年+月)×日]的值相加，得到的数值写入到(D8311，D8310)中。



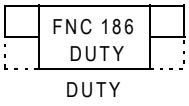
24.3 FNC 186 - DUTY/产生定时脉冲

A8 A8C

概要

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。

1. 指令格式



16位指令	指令记号	执行条件
7步	DUTY	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
n1	ON的扫描次数(运算周期) [n1>0]	BIN16位
n2	OFF的扫描次数(运算周期)[n2>0]	
(D・)	定时时钟输出的目标地址	位

3. 对象软元件

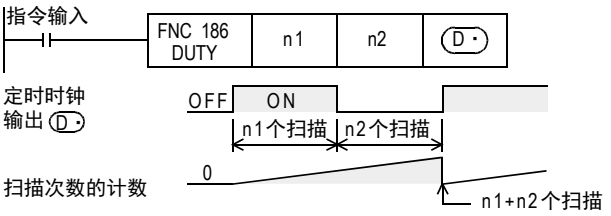
操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
n1												●	●	●	●					●	●			
n2												●	●	●	●					●	●			
(D・)			▲																●					

▲: 请指定M8330～M8334。

功能和动作说明

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) 请在定时时钟的输出的目标地址 (D・) 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， (D・) 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•) ON/OFF的状态
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	DUTY(FNC 186)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 扫描次数的计数	DUTY(FNC 186)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 扫描次数的计数	DUTY(FNC 186)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 扫描次数的计数	DUTY(FNC 186)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 扫描次数的计数	DUTY(FNC 186)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 扫描次数的计数	DUTY(FNC 186)指令的定时时钟输出5用的扫描次数的计数值

注意要点

- 该指令能使用5次(点)。
  但是，在多个DUTY(FNC 186)指令中不能使用相同的定时时钟输出目标地址 (D•) 。

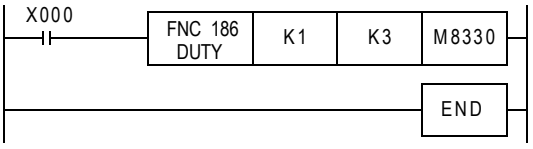
错误

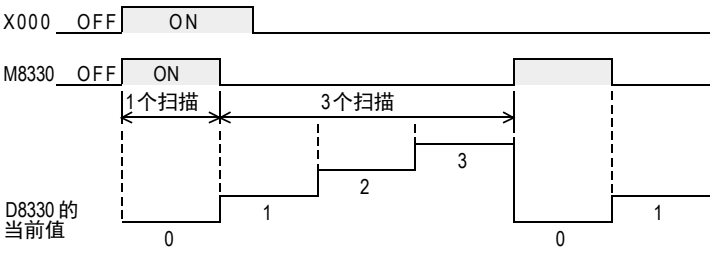
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- n1、n2不满时。(错误代码: K6706)
- (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。





24.4 FNC 188 - CRC/CRC运算

A8 A8C

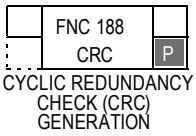
概要

在通信等中被使用的错误校验方法之一为CRC(Cyclic Redundancy Check:循环冗余校验)，用CRC指令计算出该CRC值。在错误校验的方法中，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CCD指令(FNC 84)。

此外，在这个指令中，使用了生成CRC值(CRC-16)用的[$X^{16}+X^{15}+X^2+1$]的生成多项式。

→ 有关CCD指令(校验码)，请参考16.5节

1. 指令格式



16位指令	指令记号	执行条件
7步	CRC	连续执行型
	CRCP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存作为CRC值生成对象的数据的软元件起始编号	BIN16位
(D・)	保存被生成的CRC值的软元件编号	
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	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								▲	▲	▲	▲	●	●	●	●	●			●					
(D・)									▲	▲	▲	●	●	●	●	●			●					
n														●	●					●	●			

▲:位软元件的位数指定，请务必指定4位数(K4□○○○)。

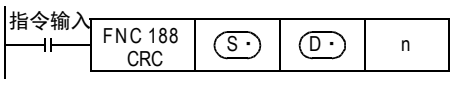
功能和动作说明

1. 16位运算

以(S・)中指定的软元件为起始的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。

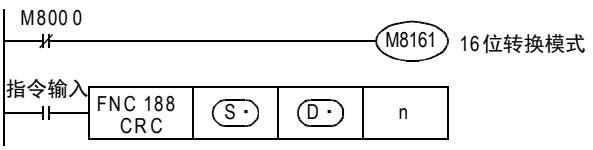
此外，为了生成CRC值(CRC-16)，使用了[$X^{16}+X^{15}+X^2+1$]的生成多项式。



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。

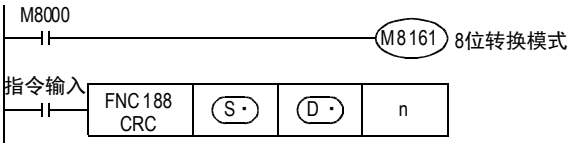


			例如) (S·)=D100 (D·)=D0 n= 6		
			软元件	对象数据的内容	
				8位	16位
保存生成CRC值的对象数据的地址	(S·)	低字节	D100低字节	01H	0301H
		高字节	D100高字节	03H	
	(S·)+1	低字节	D101低字节	03H	0203H
		高字节	D101高字节	02H	
	(S·)+2	低字节	D102低字节	00H	1400H
		高字节	D102高字节	14H	
	≀	≀	—		
	(S·)+n/2-1	低字节	-		
		高字节			
保存CRC值的地址	(D·)	低字节	D0低字节	E4H	41E4H
		高字节	D0高字节	41H	

8位转换模式[M8161＝ON]

在8位转换模式下，仅对 (S·) 软元件的低8位(字节)执行运算。

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
保存生成CRC值的对象数据的地址	(S·)	低字节	D100低字节	01H
	(S·)+1	低字节	D101低字节	03H
	(S·)+2	低字节	D102低字节	03H
	(S·)+3	低字节	D103低字节	02H
	(S·)+4	低字节	D104低字节	00H
	(S·)+5	低字节	D105低字节	14H
	≀		—	
	(S·)+n-1	低字节	—	
保存CRC值的地址	(D·)	低字节	D0低字节	E4H
	(D·)+1	低字节	D1低字节	41H

2. 相关软元件

相关软元件	内容	
M8161*1	ON	CRC指令在8位模式下动作
	OFF	CRC指令在16位模式下动作

*1. 从RUN→STOP时清除

注意要点

在这个指令中，使用CRC值(CRC-16)的生成多项式[$X^{16}+X^{15}+X^2+1$]，此外，针对在CRC值，还有各种标准化的生成多项式。请注意，如果使用了不同的生成多项式，会产生完全不同的CRC值。

参考: 主要的CRC值生成多项式

名称	生成多项式
CRC-12	$X^{12}+x^{11}+X^3+X^2+X+1$
CRC-16	$X^{16}+X^{15}+X^2+1$
CRC-32	$X^{32}+X^{26}+X^{23}+X^{22}+X^{16}+X^{12}+X^{11}+X^{10}+X^8+X^7+X^5+X^4+X^2+X+1$
CRC-CCITT	$X^{16}+X^{12}+X^5+1$

错误

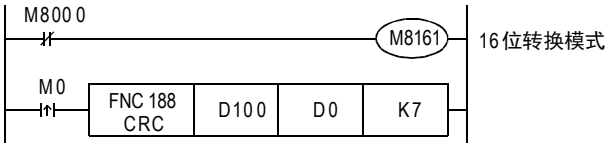
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (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)
- n在指定范围(1～256)以外时。(错误代码: K6706)
- (S・)+n-1、(D・)+1超出软元件范围时。(错误代码: K6706)

程序举例

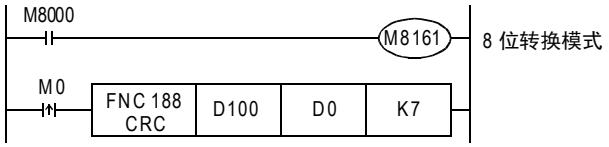
M0为ON时，生成D100～D106中保存的ASCII码[0123456]的CRC值后，保存到D0中的程序。

1. 16位模式下



	数据的内容			
			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	D0	2ACFH	低字节	CFH
			高字节	2AH

2. 8位模式下



	对象数据的内容		
保存生成CRC值的对象数据的地址	D100	低字节	30H
	D101	低字节	31H
	D102	低字节	32H
	D103	低字节	33H
	D104	低字节	34H
	D105	低字节	35H
	D106	低字节	36H
保存CRC值的地址	D0	低字节	CFH
	D1	低字节	2AH

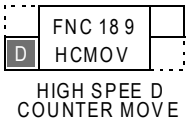
24.5 FNC 189 - HCMOV/高速计数器传送

A8 A8C

概要

传送指定的高速计数器、或是环形计数器的当前值的指令。
根据版本不同，这个指令的功能也不同。

1. 指令格式



16 位指令	指令记号	执行条件
	—	
	—	

32 位指令	指令记号	执行条件
13 步	DHCMOV	连续执行型
	—	

2. 设定数据

操作数种类	内容	数据类型
(S)	作为传送源的高速计数器，或是环形计数器*1的软元件编号	BIN32位
(D)	传送目标的软元件编号	
n	传送后，作为清除传送源的高速计数器，或是环形计数器*1的当前值的指示[清除(K1)，不处理(K0)]	BIN16位

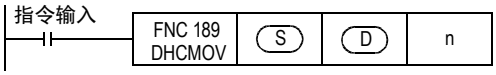
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统用户							位数指定				系统用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S)													▲	▲										
(D)														●	●									
n																				●	●			

▲: 仅可以指定高速计数器(C235～C255)、环形计数器(D8099，D8398)*1。

功能和动作说明

1. 32位运算(DHCMOV)



- 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) 的软元件		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	[D8399，D8398]的当前值→[(D)+1，(D)]

- 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。

2. 高速计数器的当前值更新时序以及DHCMOV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～C255)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指令等应用指令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) DHCMOV指令的效果

- 同时使用输入中断和DHCMOV指令时，可以根据外部输入的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。

→ 请参考程序举例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。
  - 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。

→ 关于将硬件计数器作为软件计数器使用的条件，请参考4.7.9项

  - 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。

→ 关于因软件计数器的综合频率而受到的限制，请参考4.7.10项

  - 根据高速计数器的当前值变化时序执行比较，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)。
  - DHCMOV指令的使用次数没有限制。

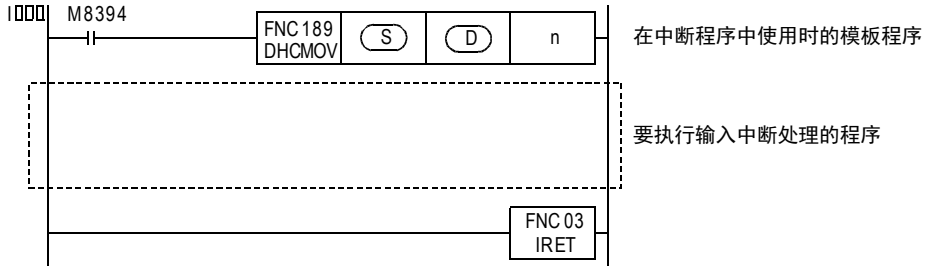
注意要点

在输入中断程序中编程时，需要注意以下几点。
关于输入中断用指针和输入的分配，请参考下面的5)表格。

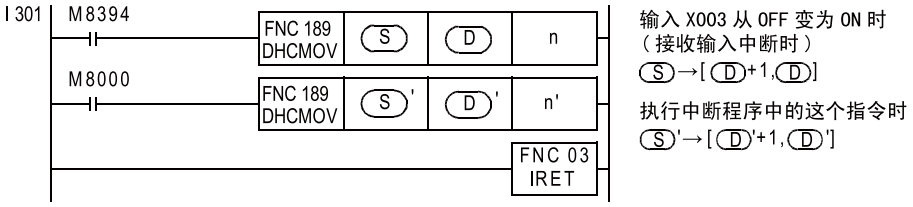
1) 请在主程序中编写EI指令(FNC 04)和FEND指令(FNC 06)指令。
执行输入中断程序时需要。

→ 关于EI指令(FNC 04)和FEND(FNC 06)指令，请参考8.5节和8.6节

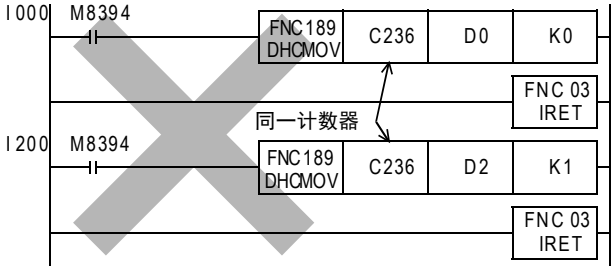
2) 在输入中断程序的第1行中编程时，请务必编写如下的模板程序。
指令触点请务必使用M8394。



3) 在1个输入中断程序中多次使用 DHCMOV指令时，当产生中断事件，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DHCMOC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
M8050*1	I000、I001	X000
M8051*1	I100、I101	X001
M8052*1	I200、I201	X002
M8053*1	I300、I301	X003
M8054*1	I400、I401	X004
M8055*1	I500、I501	X005

*1. 从RUN→STOP时清除

6) 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EI指令后，等达到允许中断的状态才执行中断程序。

错误

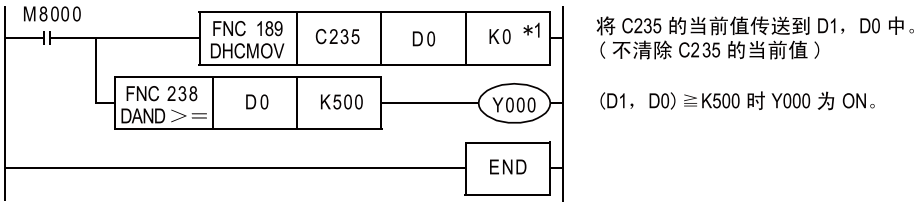
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (S)、[(D)+1，(D)]中指定的软元件为对象以外。(错误代码:K6705)

程序举例

1. 程序举例1

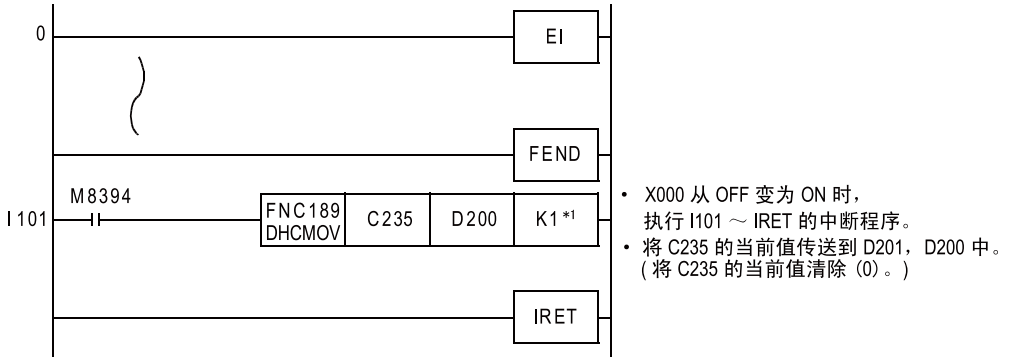
每个运算周期比较高速计数器C235的当前值，当为K500以上时，使输出Y000动作的程序。
(不清除C235的当前值的情况下)



*1. K0：执行指令时不清除当前值。
K1：执行指令时清除当前值。

2. 程序举例2

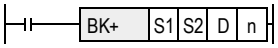
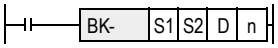
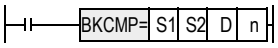
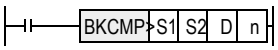
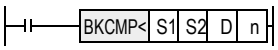
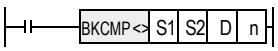
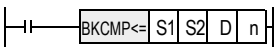
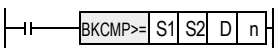
当X001从OFF变为ON时，将C235的当前值传送到D201、D200中，并清除当前值的程序。



*1. K0：执行指令时不清除当前值。
K1：执行指令时清除当前值。

25. 数据块处理—FNC 190～FNC 199

在FNC 190～FNC 199中，提供了执行数据块的加法运算、减法运算、比较的指令。

FNC No.	指令记号	符号	功能	参考
190	—			—
191	—			—
192	BK+		数据块的加法运算	25.1节
193	BK-		数据块的减法运算	25.2节
194	BKCMP=		数据块的比较 (S1) = (S2)	25.3节
195	BKCMP>		数据块的比较 (S1) > (S2)	25.3节
196	BKCMP<		数据块的比较 (S1) < (S2)	25.3节
197	BKCMP<>		数据块的比较 (S1) ≠ (S2)	25.3节
198	BKCMP<=		数据块的比较 (S1) ≦ (S2)	25.3节
199	BKCMP>=		数据块的比较 (S1) ≧ (S2)	25.3节

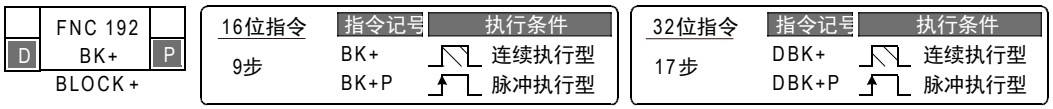
25.1 FNC 192 - BK+/数据块的加法运算

A8 A8C

概要

数据块的BIN加法运算的指令。

1. 指令格式



2. 设定数据

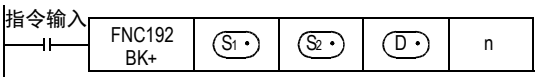
操作数种类	内容	数据类型
(S1·)	保存执行加法运算的数据的软元件起始编号	BIN16/32位
(S2·)	执行加法运算的常数，或是保存执行加法运算的数据的软元件起始编号	
(D·)	保存运算结果的软元件起始编号	
n	数据的个数	

3. 对象软元件

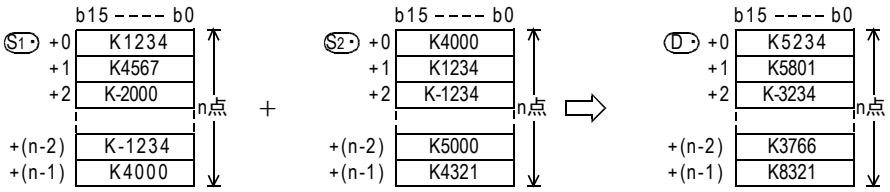
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)												●	●	●	●				●					
(S2·)												●	●	●	●				●	●	●			
(D·)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(BK+/BK+P)



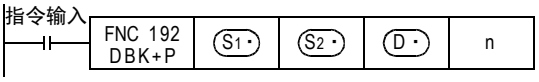
1) 将(S1·)开始n点16位数据和(S2·)开始的n点16位数据(BIN)进行加法运算后，将运算结果保存到(D·)开始的 n 点中。



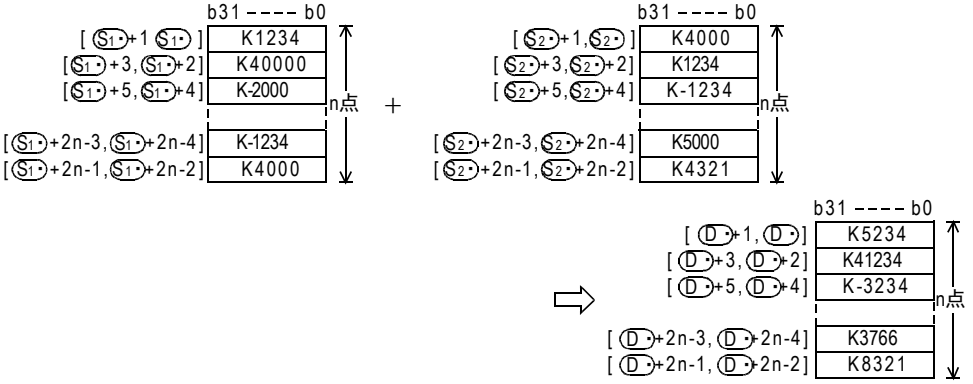
2) 可以在(S2·)中直接指定-32768～32767(16位)的常数。



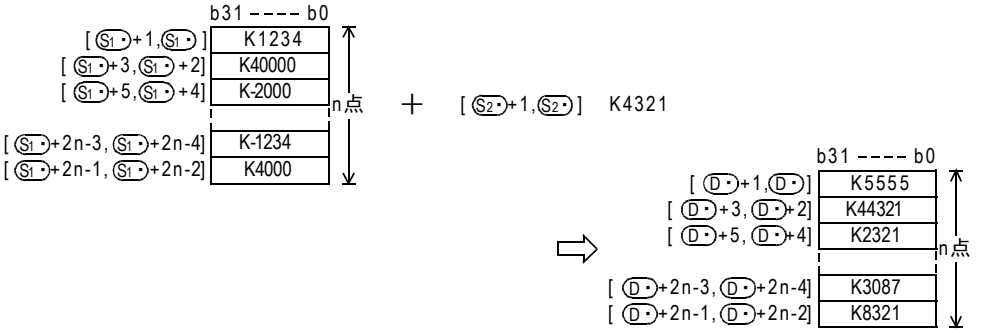
2. 32位运算(DBK+/DBK+P)



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・]开始的2n点中。



2) 可以在[S2・+1，S2・]中直接指定-2,147,483,648～2,147,483,647(32位)的常数。



相关指令

指令	内容
BK-(FNC 193)	数据块的减法运算指令

注意要点

- 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-32768(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)

- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DBK+ D0 D100 D200 R0时，则n=[R1, R0]。

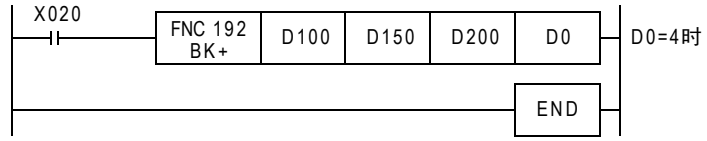
错误

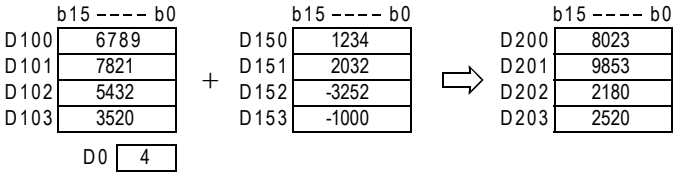
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (S1·)、(S2·)、(D·)开始的 (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)
- (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)
- (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)

程序举例

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。





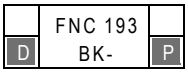
25.2 FNC 193 - BK-/数据块的减法运算

A8 A8C

概要

数据块的BIN减法运算的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	BK-	连续执行型
	BK-P	脉冲执行型

32位指令	指令记号	执行条件
17步	DBK-	连续执行型
	DBK-P	脉冲执行型

2. 设定数据

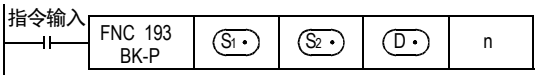
操作数种类	内容	数据类型
(S1・)	保存减法运算数据的软元件起始编号	BIN16/32位
(S2・)	要进行减法运算的常数，或是保存进行减法运算的数据的软元件起始编号	
(D・)	保存运算结果的软元件起始编号	
n	数据的个数	

3. 对象软元件

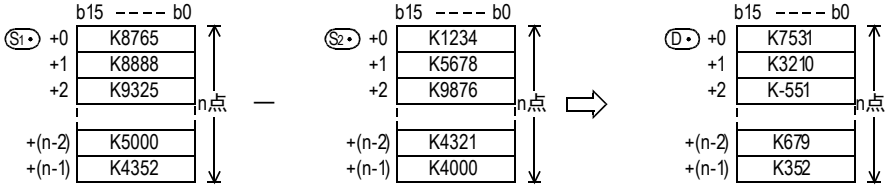
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)												●	●	●	●				●					
(S2・)												●	●	●	●				●	●	●			
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

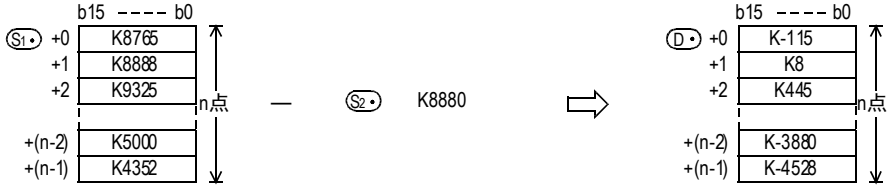
1. 16位运算(BK-/BK-P)



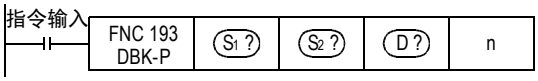
1) 将(S1・)开始n点16位数据和(S2・)开始的n点16位数据(BIN)相减后，将运算结果保存到(D・)开始的n点中。



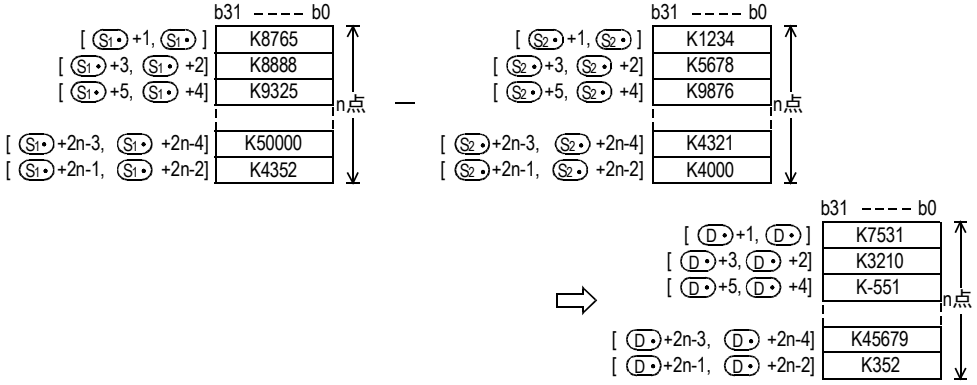
2) 可以在(S2・)中直接指定-32768～32767(16位)的常数。



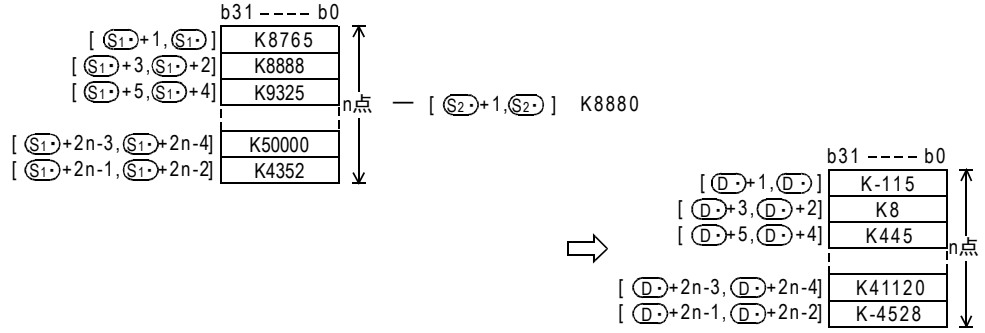
2. 32位运算(DBK-/DBK-P))



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)相减后，将运算结果保存到[(D•)+1,(D•)]开始的2n点中。



2) 可以在[(S2•)+1，(S2•)]中直接指定-2,147,483,648～2,147,483,647(32位)的常数。



相关指令

指令	内容
BK+(FNC 192)	数据块的BIN加法运算指令

注意要点

- 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

  K-32768(H8000) - K2(H0002) → K32766(H7FFE)
  K32767(H7FFF) - K-2(HFFFE) → K-32767(H8001)

  — 32位运算时

  K-2,147,483,648(H80000000) - K2(H00000002) → K2,147,483,646(H7FFFFFFE)
  K2,147,483,647(H7FFFFFFF) - K-2(HFFFFFFFE) → K-2,147,483,647(H80000001)

- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DBK- D0 D100 D200 R0时，则n=[R1, R0]。

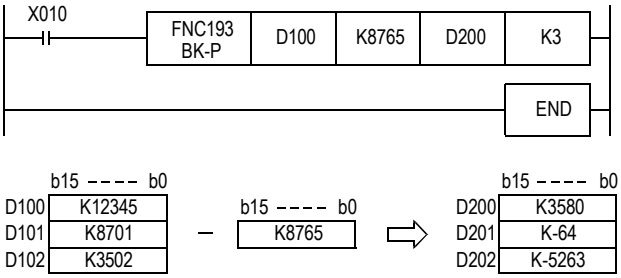
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (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)
- (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)
- (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)

程序举例

当X010为ON时，从D100开始的3点数据和常数8765相减后，将其结果保存到D200以后的软元件中的程序。



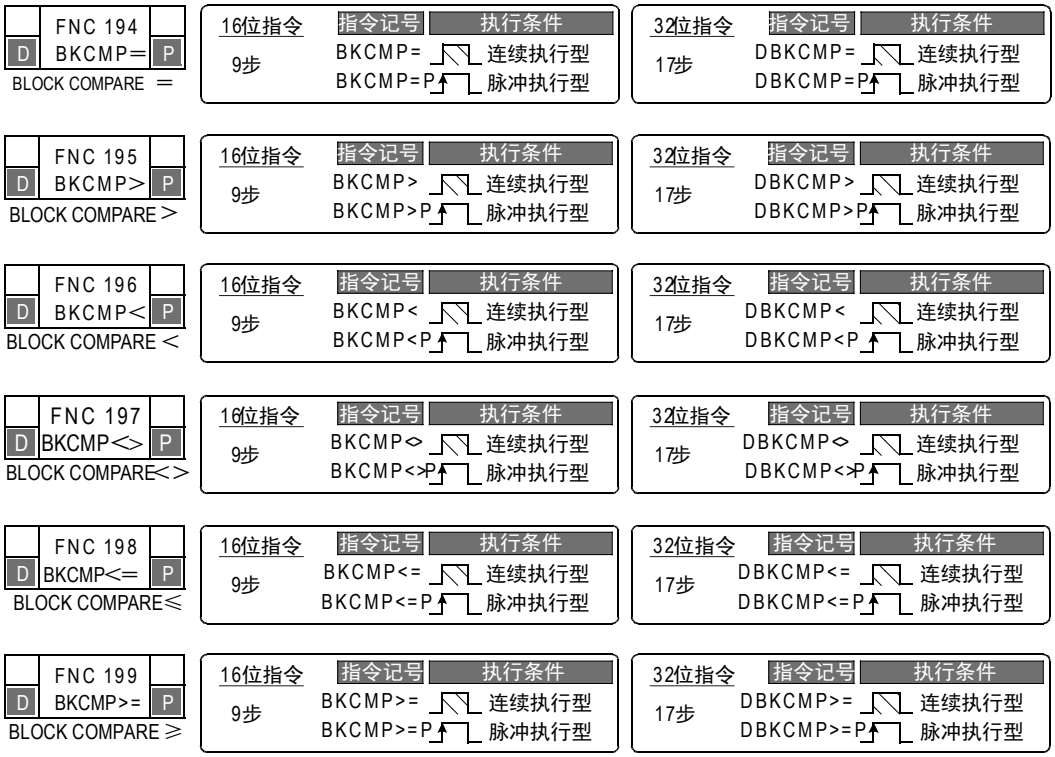
25.3 FNC 194～199-BKCMP=、>、<、<>、<=、>=/数据块比较

A8 A8C

概要

这些指令是按照各个指令的比较条件来比较数据块。

1. 指令格式



2. 设定数据(FNC 194～FNC 199通用)

操作数种类	内容	数据类型
(S1·)	比较值或是保存比较值的软元件编号	BIN16/32位
(S2·)	保存比较源数据的软元件起始编号	
(D·)	保存比较结果的软元件起始编号	位
n	要比较的数据数	BIN16/32位

3. 设定数据(FNC 194～FNC 199通用)

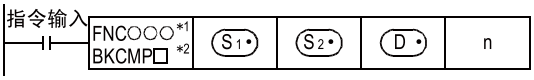
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)												●	●	●	●				●	●	●			
(S2·)												●	●	●	●				●					
(D·)		●	●			●	▲												●					
n														●	●					●	●			

▲: D□.b不能变址修饰(V、Z)。

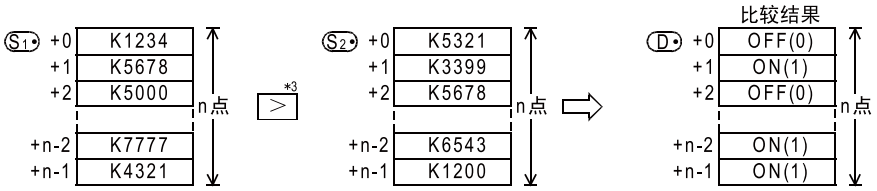
功能和动作说明

1. 16位运算(BKCMP=、>、<、<>、<=、>=/BKCMP=P、>P、<P、<>P、<=P、>=P)

1) 将 (S1・) 开始的n点16位数据(BIN)和 (S2・) 开始的n点16位数据(BIN)比较后，将比较结果保存到 (D・) 开始的n点中。

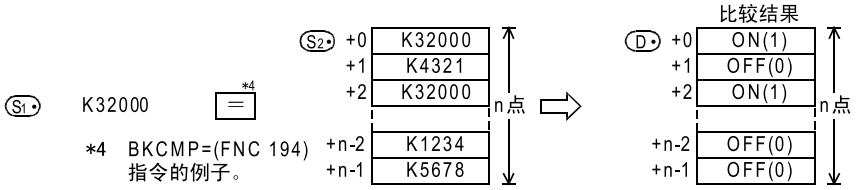


*1 ○○○中输入194～199。
*2 在□中，输入FNC No.相对应的=,>,<,<>,<=,>=。



*3 BKCMP>(FNC195) 指令的例子。

2) (S1・) 中可直接指定常数。



*4 BKCMP=(FNC 194) 指令的例子。

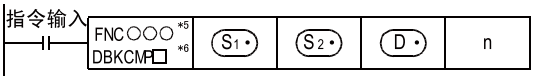
3) 各个指令的比较结果如下所示。

指令	比较结果ON(1)的条件	比较结果OFF(0)的条件
BKCMP=(FNC 194)	(S1・) = (S2・)	(S1・) ≠ (S2・)
BKCMP>(FNC 195)	(S1・) > (S2・)	(S1・) ≦ (S2・)
BKCMP<(FNC 196)	(S1・) < (S2・)	(S1・) ≧ (S2・)
BKCMP<>(FNC 197)	(S1・) ≠ (S2・)	(S1・) = (S2・)
BKCMP<=(FNC 198)	(S1・) ≦ (S2・)	(S1・) > (S2・)
BKCMP>=(FNC 199)	(S1・) ≧ (S2・)	(S1・) < (S2・)

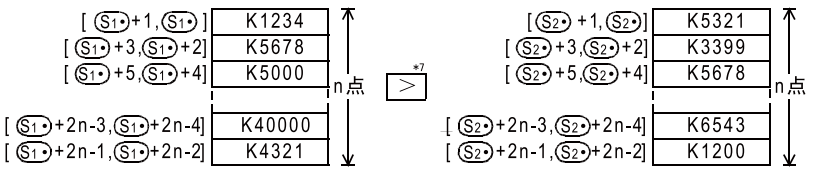
4) (D・) 开始的n点的比较结果都为ON(1)时，M8090(块比较信号)为ON。

2. 32位运算(DBKCMP=、>、<、<>、<=、>=/DBKCMP=P、>P、<P、<>P、<=P、>=P)

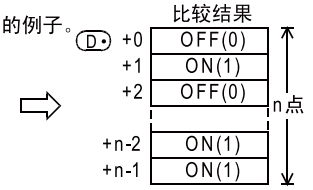
1) 将[(S1・)+1，(S1・)]开始的n点32位数据(BIN)和[(S2・)+1，(S2・)]开始的n点32位数据(BIN)比较后，将比较结果保存到 (D・) 开始的n点中。



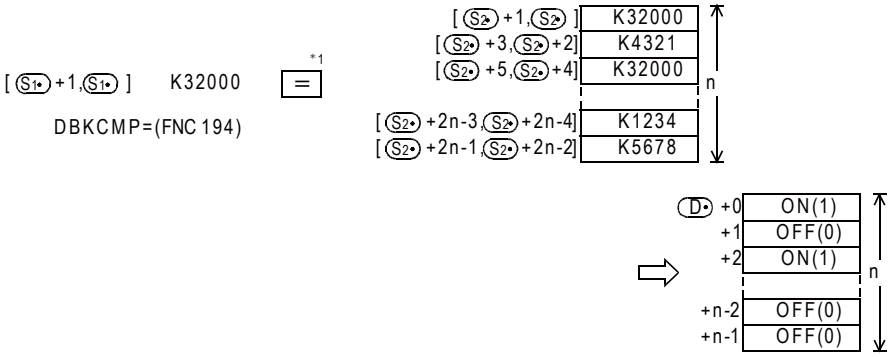
*5 ○○○中输入194～199。
*6 在□中，输入FNC No.相对应的=,>,<,<>,<=,>=。



*7 DBKCMP>(FNC195)指令的例子。



2) 可以在[(S1·)+1，(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(1)的条件	比较结果OFF(0)的条件
DBKCMP=(FNC 194)	[(S1·)+1, (S1·)]=[(S2)+1, (S2)]	[(S1·)+1, (S1·)]≠[(S2)+1, (S2)]
DBKCMP>(FNC 195)	[(S1·)+1, (S1·)]> (S2)+1, (S2)]	[(S1·)+1, (S1·)]≦ (S2)+1, (S2)]
DBKCMP<(FNC 196)	[(S1·)+1, (S1·)] [(S2)+1, (S2)]	[(S1·)+1, (S1·)]≧ (S2)+1, (S2)]
DBKCMP<>(FNC 197)	[(S1·)+1, (S1·)] [(S2)+1, (S2)]	[(S1·)+1, (S1·)]= (S2)+1, (S2)]
DBKCMP<=(FNC 198)	[(S1·)+1, (S1·)]≦ (S2·)+1, (S2·)]	[(S1·)+1, (S1·)]> (S2·)+1, (S2·)]
DBKCMP>=(FNC 199)	[(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(1)时，M8090(块比较信号)为ON。

相关软元件

→ 关于块比较信号的使用方法，请参考6.5.2项

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(1)时变为ON。 DBKCMP=(FNC 194)、DBKCMP>(FNC 195)、DBKCMP<(FNC 196)、 DBKCMP<>(FNC 197)、DBKCMP<=(FNC 198)、DBKCMP>=(FNC 199)

注意要点

- 使用32位计数器(包含高速计数器)时
  32位计数器(C200～C255)的比较，必须在32位运算(DBKCMP=、DBKCMP>、DBKCMP<等)下进行比较。如在16位运算(BKCMP=、BKCMP>、BKCMP<等)下指定，则发生运算错误。
  (错误代码: K6705)
- 将D、R指定为32位指令的n时，[n+1,n]的32位值便生效，敬请注意。
  DBKCMP= D0 D2 D100 R0时，则n=[R1, R0]。

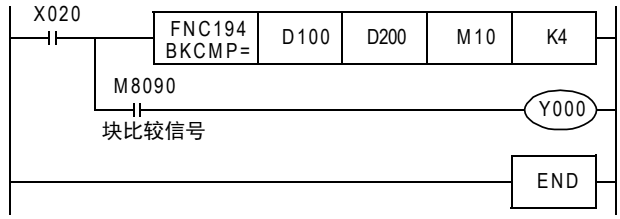
错误

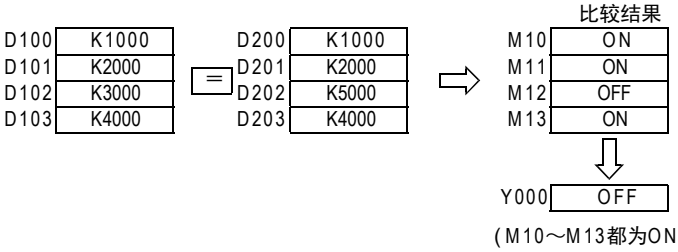
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (S1・)、(S2・)开始的 (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)
- (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)
- (D・)指定为“ D□ .b”时，(D・)的数据寄存器和(S1・)开始的n点 (32位运算时为 2n 点)软元件范围重复时。(错误代码: K6706)
- (D・)指定为“ D□ .b”时，(D・)的数据寄存器和(S2・)开始的n点 (32位运算时为 2n 点)软元件范围重复时。(错误代码: K6706)
- 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位运算(DBKCMP=、DBKCMP>、DBKCMP<等)指令进行比较。

程序举例

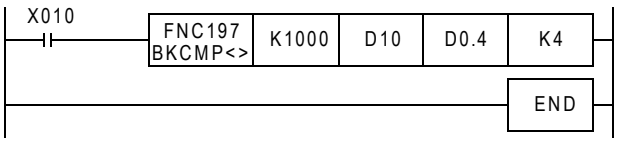
1) 当X020为ON时，使用BKCMP=(FNC 194)指令对D100开始的4点16位数据(BIN)和D200开始的4点16位数据(BIN)进行比较，并将其结果保存到M10开始的4点软元件中的程序。
此外，比较结果(M10开始的4点)全部为ON(1)时，Y000置ON。

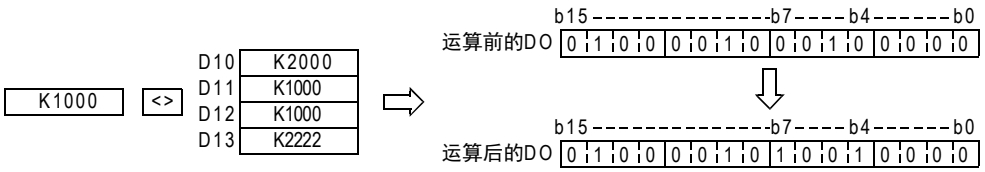




(M10～M13都为ON时Y000变为ON)

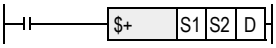
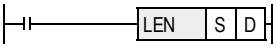
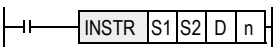
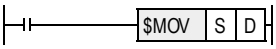
2) 当X010为ON时，将常数K1000和D10开始的4点数据进行比较，然后将其结果保存到D0的b4～b7中的程序。





26. 字符串控制—FNC 200～FNC 209

提供了结合字符串数据、替换部分字符以及从左右取出部分字符等的针对字符串进行控制的指令。

FNC No.	指令记号	符号	功能	参考
200	STR		BIN→字符串的转换	26.1节
201	VAL		字符串→BIN的转换	26.2节
202	$+		字符串的结合	26.3节
203	LEN		检测出字符串的长度	26.4节
204	RIGHT		从字符串的右侧开始取出	26.5节
205	LEFT		从字符串的左侧开始取出	26.6节
206	MIDR		从字符串中的任意取出	26.7节
207	MIDW		字符串中的任意替换	26.8节
208	INSTR		字符串的检索	26.9节
209	$MOV		字符串的传送	26.10节

26.1 FNC 200 - STR/BIN→字符串的转换

A8 A8C

概要

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ESTR(FNC 116)指令。

→ 关于字符串，请参考5.3节
→ 有关ESTR(FNC 116)指令，请参考18.4节

1. 指令格式

FNC 200 STR (D / P)
STRING

16位指令	指令记号	执行条件
7步	STR	连续执行型
	STRP	脉冲执行型

32位指令	指令记号	执行条件
13步	DSTR	连续执行型
	DSTRP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存要转换数值的位数的软元件起始编号	BIN16位
(S2・)	保存要转换的BIN数据的软元件编号	BIN16/32位
(D・)	保存已转换的字符串的软元件起始编号	字符串

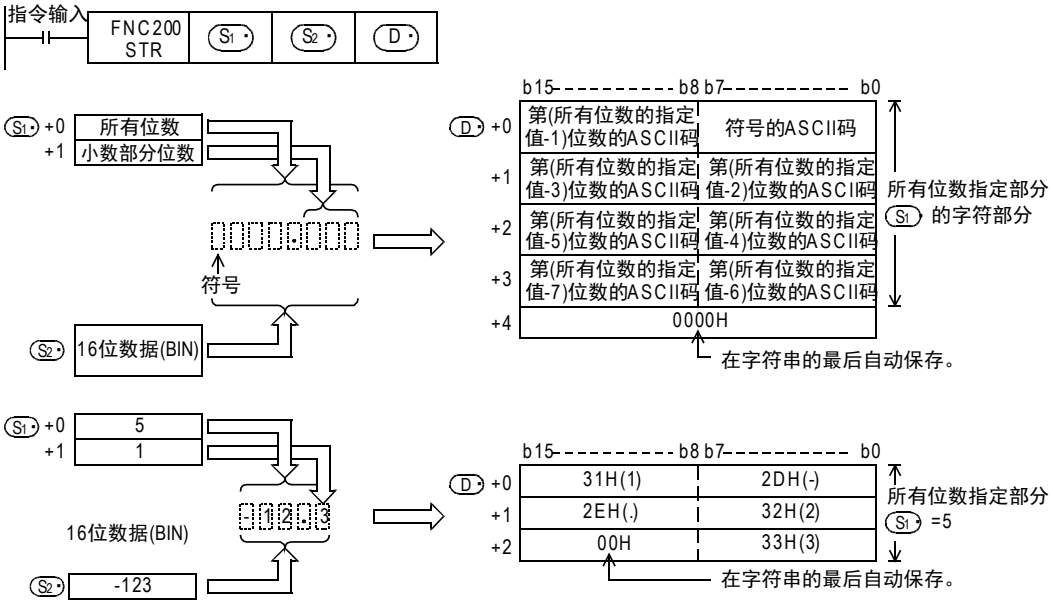
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)												●	●	●	●				●					
(S2・)								●	●	●	●	●	●	●	●	●	●	●	●	●	●			
(D・)												●	●	●	●				●					

功能和动作说明

1. 16位运算(STR/STRP)

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。



2) 在2～8位数的范围内设定所有位数 (S1·) 。

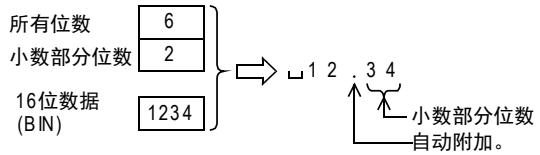
3) 在0～5位数的范围内设定小数部分位数 (S1·) +1。
但是，请设定为小数部分位数≦(所有位数-3)。

4) 要转换的16位数据(BIN) (S2·) 的值在-32768～32767的范围内。

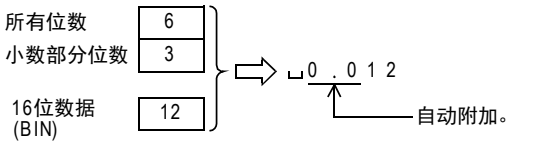
5) 转换后的字符串数据会如下所示地保存到 (D·) 开始的软元件编号中。

— 在符号中，16位数据(BIN) (S2·) 为正时保存“空格(20H)”，为负时保存“-(2DH)”。

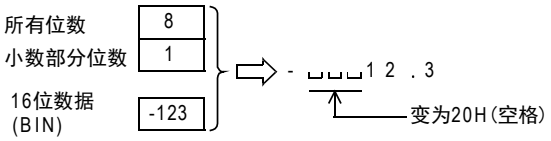
— 在小数部分位数 (S1·) +1中设定为“0”以外的数字时，会自动在小数部分位数+1位数的位置上加上小数点“.(2EH)”。
小数部分位数 (S1·) +1为“0”时，不附加小数点。



— 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。



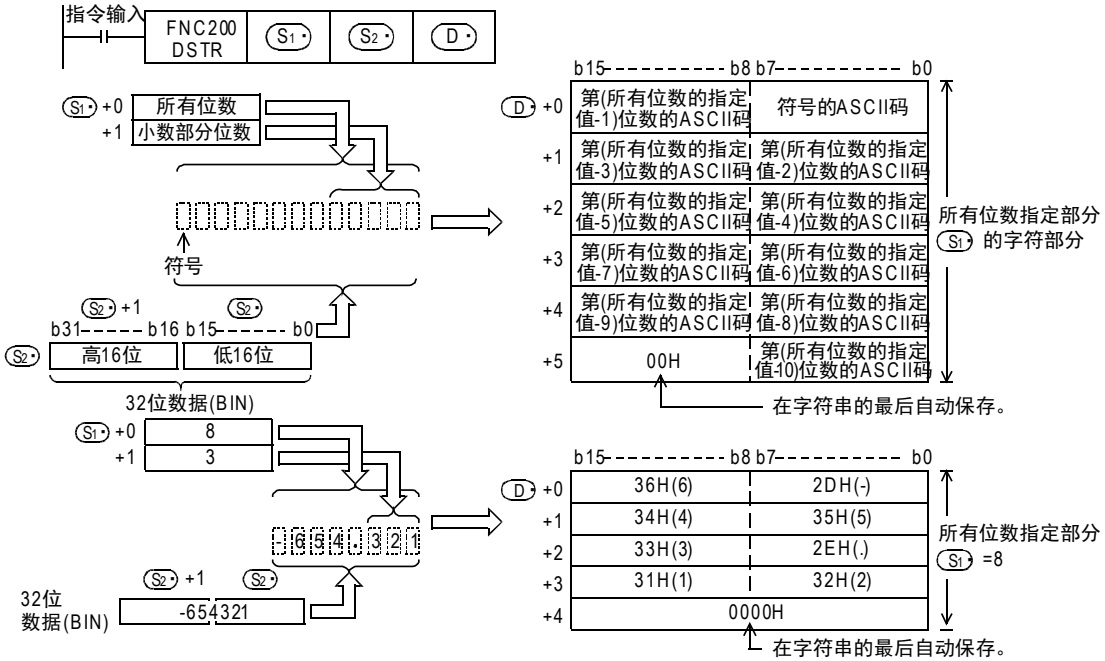
— 如果除去小数点和符号以外，所有位数 (S1·) 的位数多于 (S2·) 的16位数据(BIN)的位数时，在符号和数值之间保存“空格(20H)”。
此外， (S2·) 的16位数据(BIN)的位数较多时，会错误。



— 在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。
总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8位)中保存“00H”。

2. 32位运算(DSTR/DSTRP)

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) 在2～13位数的范围内设定所有位数 (S1・)。

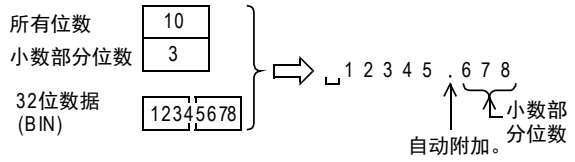
3) 在0～10位数的范围内设定小数部分位数 (S1・)+1。
但是，请设定为小数部分位数≦(所有位数-3)。

4) 要转换的32位数据(BIN) (S2・) 的值在-2,147,483,648～2,147,483,647的范围内。

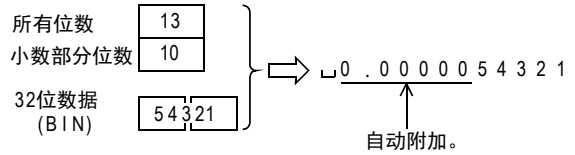
5) 转换后的字符串数据会如下所示地保存到 (D・) 开始的软元件编号中。

— 在符号中，32位数据(BIN) (S2・) 为正时保存“空格(20H)”，为负时保存“-(2DH)”。

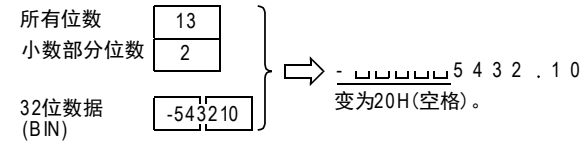
— 在小数部分位数 (S1・)+1中设定为“ 0”以外的数字时，会自动在小数部分位数+1位数的位置上加上小数点“. (2EH)”。
小数部分位数 (S1・)+1为“0”时，不附加小数点。



— 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。



— 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，会错误。



— 在已转换的字符串的末尾处，会自动保存表示字符串末尾含义的“00H”。
总位数为偶数位数时，在保存末尾字符软元件的后一个软元件中保存“0000H”。
此外，当为奇数位数时，在保存末尾字符的软元件的高字节(8位)中保存“00H”。

相关指令

指令	内容
ESTR(FNC 116)	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。
EVAL(FNC 117)	将字符串(ASCII码)数据转换成2进制浮点数数据的指令。
VAL(FNC 201)	将字符串(ASCII码)数据转换成BIN数据的指令。

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 所有位数 (S1・) 为如下所示的范围以外时。(错误代码 : K6706)

	设定范围
16位运算	2～8
32位运算	2～13

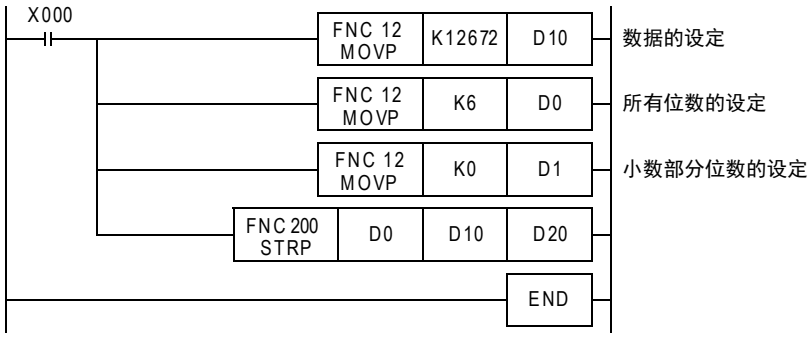
- 所有位数 (S1・)+1为如下所示的范围以外时。(错误代码: K6706)

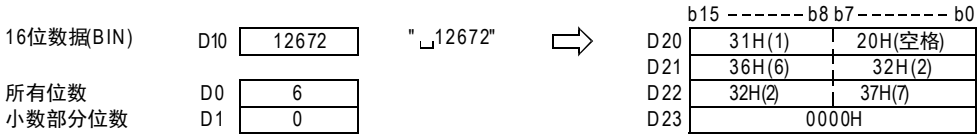
	设定范围
16位运算	0～5
32位运算	0～10

- 所有位数 (S1・) 和小数部分位数 (S1・)+1的关系非如下所示的范围时。(错误代码: K6706)
(所有位数-3)≧小数部分位数
- 所有位数 (S1・)+符号、小数点的位数比 (S2・) 的BIN数据位数少时。
(错误代码: K6706)
- 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。





26.2 FNC 201 - VAL/字符串→BIN的转换

A8 A8C

概要

将字符串(ASCII码)转换成BIN数据的指令。
还有将字符串(ASCII码)转换成浮点数数据的EVAL(FNC 117)指令。

→ 关于字符串，请参考5.3节
→ 有关EVAL(FNC 117)指令，请参考18.5节

1. 指令格式

FNC 201 VAL	D	P
VALUE		

16位指令	指令记号	执行条件
7步	VAL	连续执行型
	VALP	脉冲执行型

32位指令	指令记号	执行条件
13步	DVAL	连续执行型
	DVALP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要转换成BIN数据的字符串的软元件起始编号	字符串
(D1・)	保存已经转换的BIN数据位数的软元件起始编号	BIN16位
(D2・)	保存已经转换的BIN数据的软元件起始编号	BIN16/32位

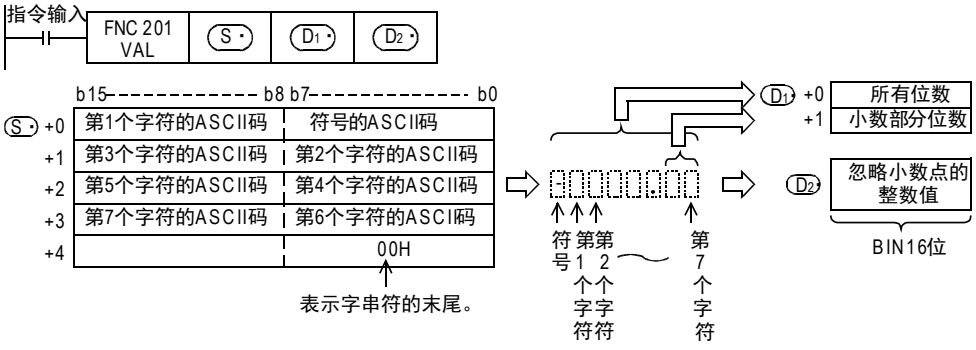
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D1・)												●	●	●	●				●					
(D2・)									●	●	●	●	●	●	●	●			●					

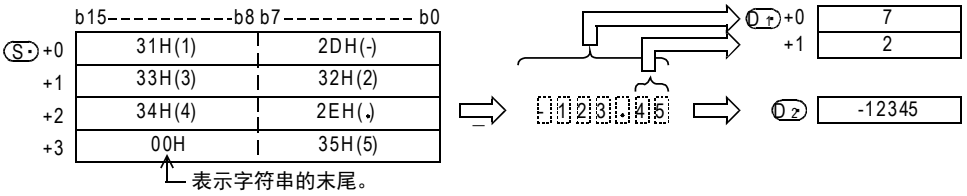
功能和动作说明

1. 16位运算(VAL/VALP)

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・)小数部分位数保存到(D1・)+1中、将BIN数据保存到(D2・)中。
从字符串转换成BIN时，以字节为单位将(S・)开始到保存“00H”的软元件编号为止的数据，作为字符串进行处理。



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。



2) 要转换的字符串数据

a) 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”

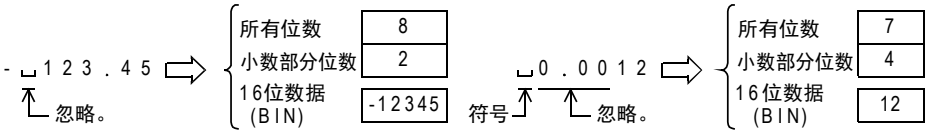
b) 在要转换的字符中使用的字符种类

		字符的种类
符号	正的数值	“空格(20H)”
	负的数值	“-(2DH)”
小数点		“. (2EH)”
数字		“0(30H)”～“9(39H)”

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)。

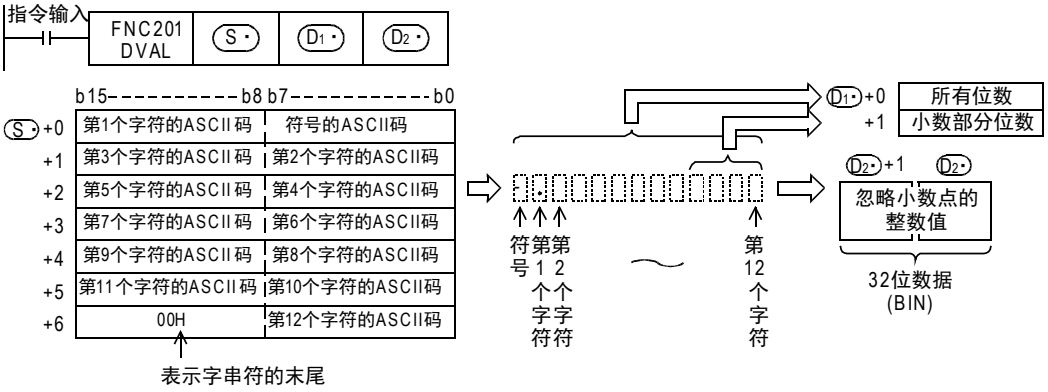
4) (D1•)+1中保存小数部分的位数。小数部分的位数为小数点“. (2EH)”以后的字符数。

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)。

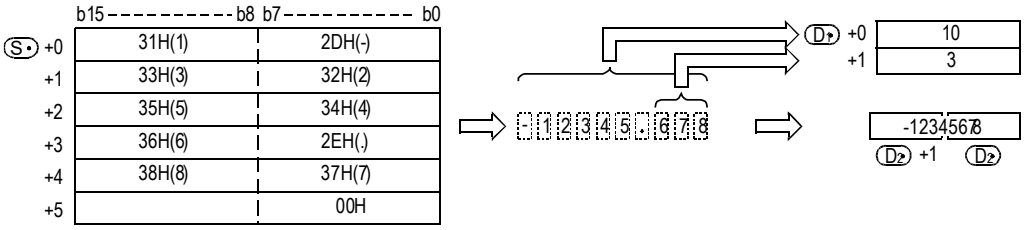


2. 32位运算(DVAL/DVALP)

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•)，小数部分位数保存到(D1•)+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) 要转换的字符串数据

a) 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	-2,147,483,648～2,147,483,647 例如)“12345.678”→“12345678”

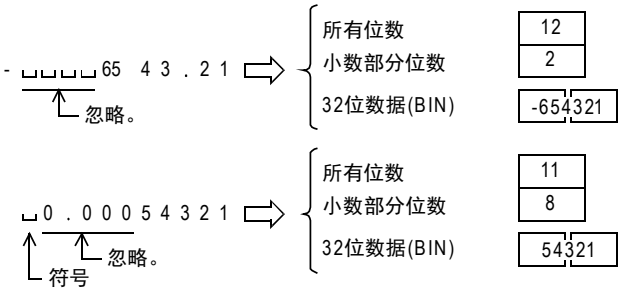
b) 在要转换的字符中使用的字符种类

		字符的种类
符号	正的数值	“空格(20H)”
	负的数值	“-(2DH)”
小数点		“. (2EH)”
数字		“0(30H)”～“9(39H)”

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)。

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。

5) 在[(D2・)+1，(D2・)]中，无视小数点，将字符串转换成32位的数据(BIN)。
但是，在(S・)开始的字符串中，符号和最初的“0”以外的数字之间的“空格(20H)”或是“0(30H)”被忽略，而转换成32位数据(BIN)。



相关指令

指令	内容
ESTR(FNC 116)	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。
EVAL(FNC 117)	将字符串(ASCII码)数据转换成2进制浮点数数据的指令。
STR(FNC 200)	将BIN数据转换成字符串(ASCII码)数据的指令。

注意要点

符号数据“空格(20H)”或是“-(2DH)”，必须保存在第1个字节(在(S・)中设定的起始软元件的低8位)里。
此外，从(S・)的第 个字节开始到字符串末尾00H的ASCII码数据，只能保存“0(30H)”～“9(39H)”、“空格(20H)”以及小数点“. (2EH)”。
在第2个字节后保存“-(2DH)”，会发生运算错误(错误代码:K6706)。

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 要转换的字符串((S・)以后)的字符数为如下所示的范围以外时。(错误代码:K6706)

	设定范围
16位运算	2～8
32位运算	2～13

- 要转换的字符串(S·以后)的小数部分的字符数为如下所示的范围以外时。
  (错误代码: K6706)

	设定范围
16位运算	0～5
32位运算	0～10

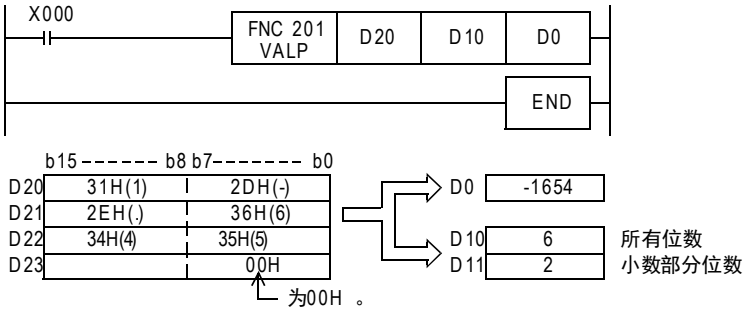
- 要转换的字符串(S·以后)的所有字符数，和小数部分的字符数之间的关系非如下所示的范围时。
  (错误代码: K6706)
  (所有字符数-3)≧小数部分字符数
- 在符号中设定了“空格(20H)”、“-(2DH)”以外的ASCII码时。
  (错误代码: K6706)
- 各数字的位数中设定了“0(30H)”～“9(39H)”，以及小数点“. (2EH)”以外的ASCII码时。
  (错误代码: K6706)
- 要转换的字符串(S·以后)中设定了多个小数点“. (2EH)”时。(错误代码 K6706)
- 转换后的BIN数据超出了如下所示的范围时。(错误代码: K6706)

	设定范围
16位运算	-32768～32767
32位运算	-2,147,483,648～2,147,483,647

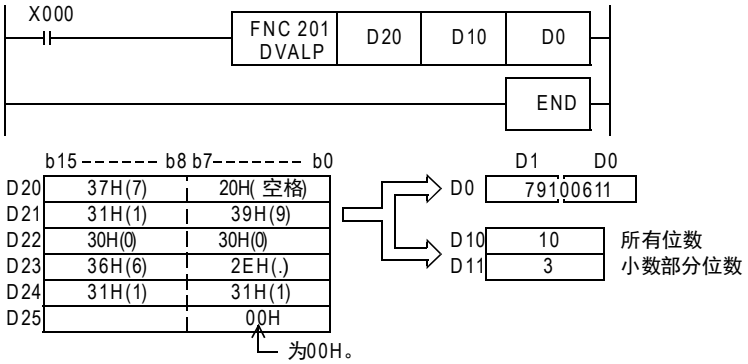
- (S·)从开始到相应软元件的最后软元件编号之间不存在“00H”时。(错误代码: K6706)

程序举例

1) 当X000为ON后，将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。



2) 当X000为ON时，将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。



26.3 FNC 202 - $+/字符串的结合

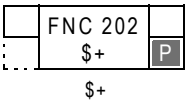
A8 A8C

概要

连接字符串与字符串的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
7步	$+	连续执行型
	$+P	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(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	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	
(D・)	保存连接后的数据(字符串)的软元件起始编号	

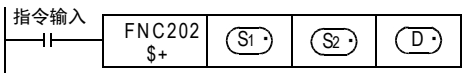
3. 对象软元件

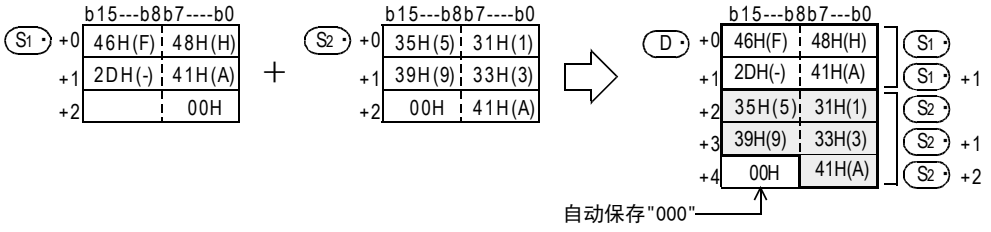
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●				●	
(S2・)								●	●	●	●	●	●	●	●	●			●				●	
(D・)									●	●	●	●	●	●	●	●			●					

功能和动作说明

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到第一个[00H]的位置为止的数据。





- 字符串的结合，就是指忽略 (S1•) 中表示指定字符串末尾的“00H”，接着 (S1•) 的最后字符连接 (S2•) 中指定的字符串。
  此外，执行字符串的结合后会自动将“00H”附加在最后。
  － 连接后的字符数为奇数时，在保存最后字符的软元件的高字节中保存“0000H”。
  － 连接后的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”。

注意要点

- 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。
- (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”。

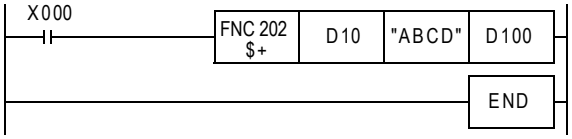
错误

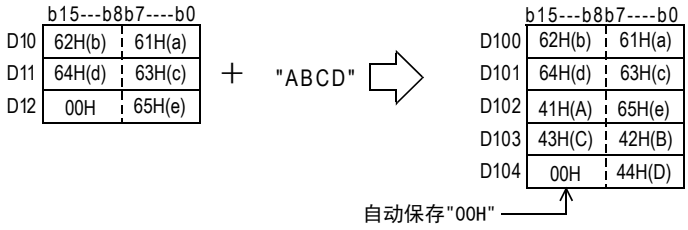
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (D•) 中指定的软元件编号开始的软元件数，比保存所有已经结合的字符串所需的软元件数量更少时。
  (在所有的字符串和最终字符后面不能保存[00H])
  (错误代码: K6706)
- 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)
- 当 (S1•) 和 (S2•) 指定的软元件开始的相应软元件范围中[00H]未被设定时。
  (错误代码: K6706)

程序举例

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。





26.4 FNC 203 - LEN/检测出字符串的长度

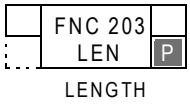
A8 A8C

概要

检测出指定字符串的字符数(字节数)的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
5步	LEN	连续执行型
	LENP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要检测出字符数的字符串的软元件起始编号	字符串
(D・)	保存已检测出的字符串的长度(字节数)的软元件编号	BIN16位

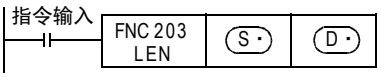
3. 对象软元件

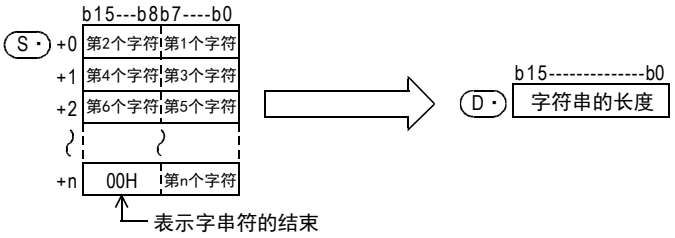
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					

功能和动作说明

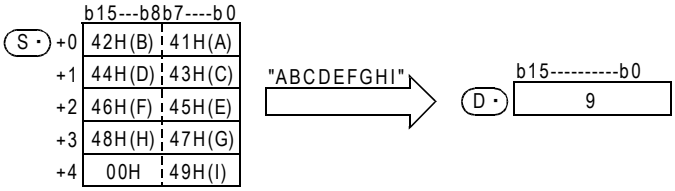
1. 16位运算(LEN/LENP)

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。(S・)以字节为单位，将从开始到第1个保存有[00H]的软元件编号为止的数据，作为字符串进行处理。





例如，如下所示的在(S・)中保存“ABCDEFGHI”时，(D・)中保存K9。



注意要点

- 在这个指令中，也可以使用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)。因此，例如使用SHIFT JIS代码，以2个字节代表1个字符的字符代码的情况下，1个字符的字符串的长度为“2”。

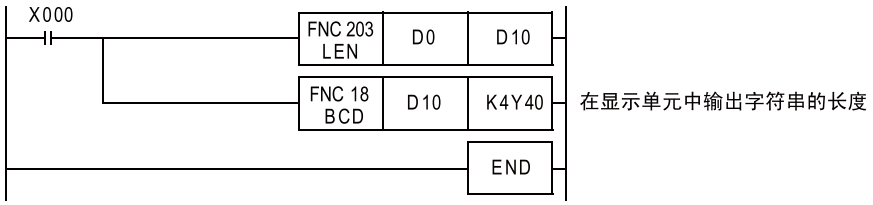
错误

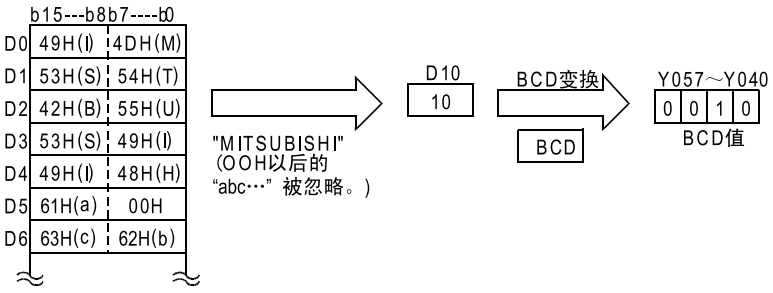
以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- 在 (S·) 指定的软元件编号开始的相应软元件范围内没有设定[00H]时。
  (错误代码: K6706)
- 检测出字符数超过32768个时。(错误代码: K6706)

程序举例

当X000为ON时，将D0开始的字符串的长度，以BCD4位数形式输出到Y040～Y057中的程序。





26.5 FNC 204 - RIGHT/从字符串的右侧开始取出

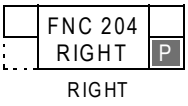
A8 A8C

概要

从指定的字符串的右侧取出指定字符数的字符的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
7步	RIGHT	连续执行型
	RIGHTP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存字符串的软元件起始编号	字符串
(D・)	保存被取出的字符串的软元件起始编号	
n	要取出的字符数	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(RIGHT/RIGHTP)

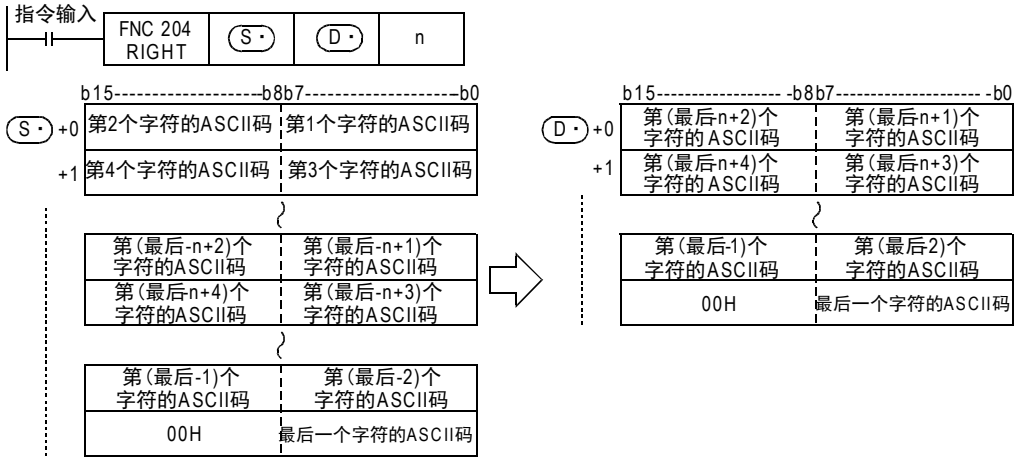
从(S・)开始的软元件中保存的字符串数据的右侧(字符串的末尾)取出n个字符的数据，保存到(D・)开始的软元件中。

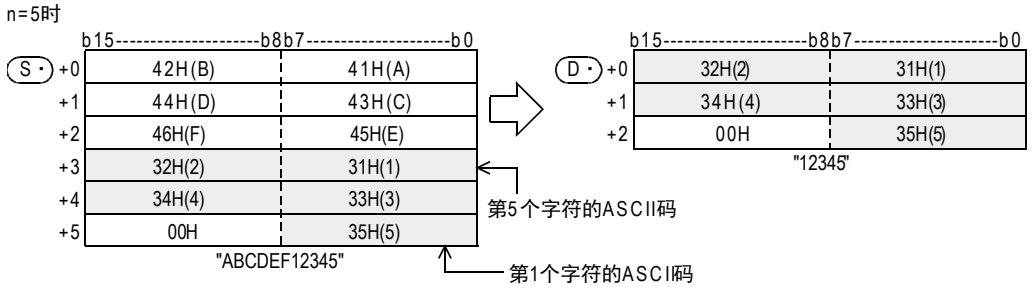
但是，但是，n中指定的字符数为“0”时，(D・)中保存NULL代码(0000H)。

此外，取出字符串时，会在最后自动附加“00H”。

－ 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。

－ 要取出的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”。





- 以 (S·) 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第1个[00H]的位置为止的数据。

注意要点

使用ASCII码以外的字符代码时，请注意以下几点。

- 字符数使用字节单位(8位)。因此，例如使用 SHIFT JIS代码，以2个字节代表1个字符的字符代码的情况下，1个字符的字符串的长度为“2”。
- 从包含了类似SHIFT JIS代码的以2个字节代表1个字符的字符代码的字符串中取出字符串时，请事先考虑一下，以1个字符的字符代码为单位取出的字符数。
  如果从2个字节的字符代码中只取出1个字节时，不能得到期望的字符代码，请务必注意。

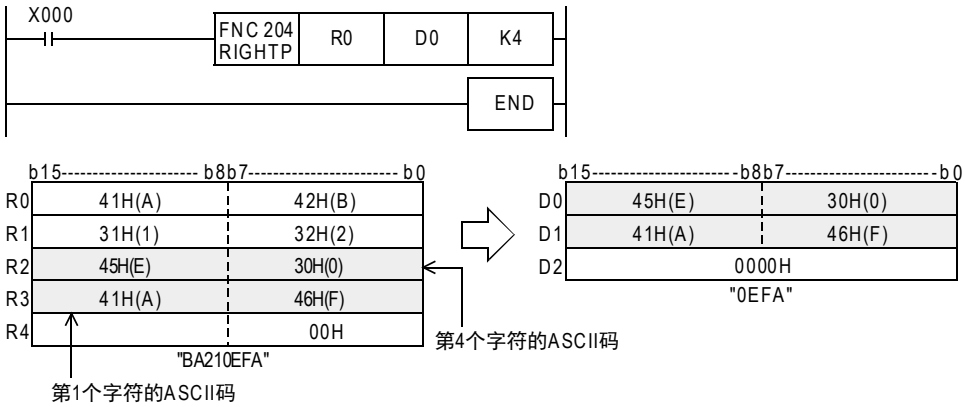
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- (S·) 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
  (错误代码: K6706)
- n超出了 (S·) 中指定的字符数时。(错误代码: K6706)
- (D·) 中指定的软元件编号开始的软元件数，比保存已经取出的字符串(n个字符)所需的软元件数更少时。
  (所有的字符串和最终字符后面不能保存[00H])
  (错误代码: K6706)
- n为负值时。(错误代码: K6706)

程序举例

当X000为ON时，将R0开始的软元件中被保存的字符串数据的右侧起的4个字符数据，保存到D0开始的软元件中的程序。



26.6 FNC 205 - LEFT/从字符串的左侧开始取出

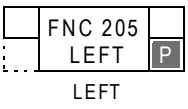
A8 A8C

概要

从指定的字符串的左侧取出指定字符数的字符的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
7步	LEFT	连续执行型
	LEFTP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存字符串的软元件起始编号	字符串
(D・)	保存被取出的字符串的软元件起始编号	
n	要取出的字符数	BIN16位

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(LEFT/LEFTP)

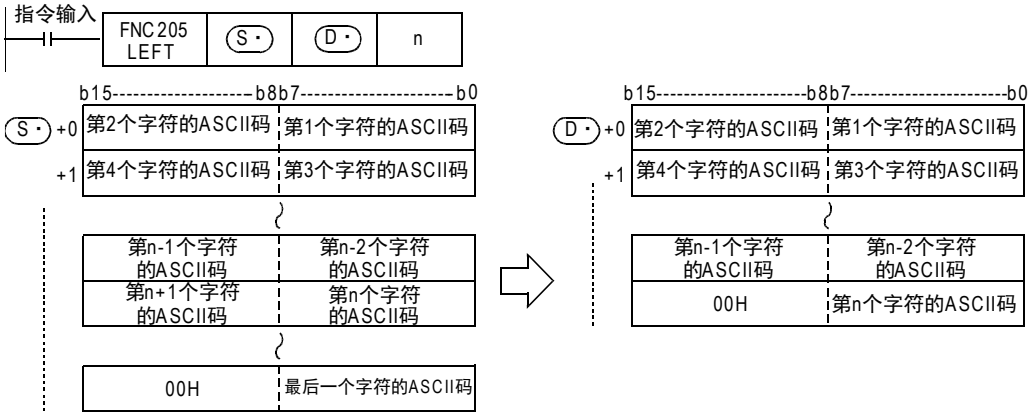
从(S・)开始的软元件中保存的字符串数据的左侧(字符串的开头)取出n个字符的数据，保存到(D・)开始的软元件中。

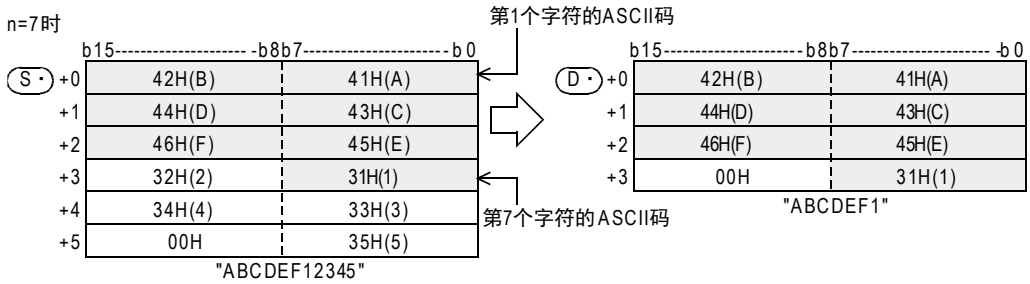
但是，n中指定的字符数为“0”时，(D・)中保存NULL代码(0000H)。

此外，取出字符串时，会在最后自动附加“00H”。

－ 要取出的字符数为奇数时，在保存最后字符的软元件的高字节中保存“00H”。

－ 要取出的字符数为偶数时，在保存最后字符的软元件的下一个软元件中保存“0000H”。





- 以 (S·) 开头的字符串，就是指从被指定的软元件开始，以字节为单位，到检测出第1个[00H]的位置为止的数据。

注意要点

使用ASCII码以外的字符代码时，请注意以下几点。

- 字符数使用字节单位(8位)。因此，例如使用 SHIFT JIS代码，以2个字节代表1个字符的字符代码的情况下，1个字符的字符串的长度为“2”。
- 从包含了类似SHIFT JIS代码的以2个字节代表1个字符的字符代码的字符串中取出字符串时，请事先考虑一下，以1个字符的字符代码为单位取出的字符数。
  如果从2个字节的字符代码中只取出1个字节时，不能得到期望的字符代码，请务必注意。

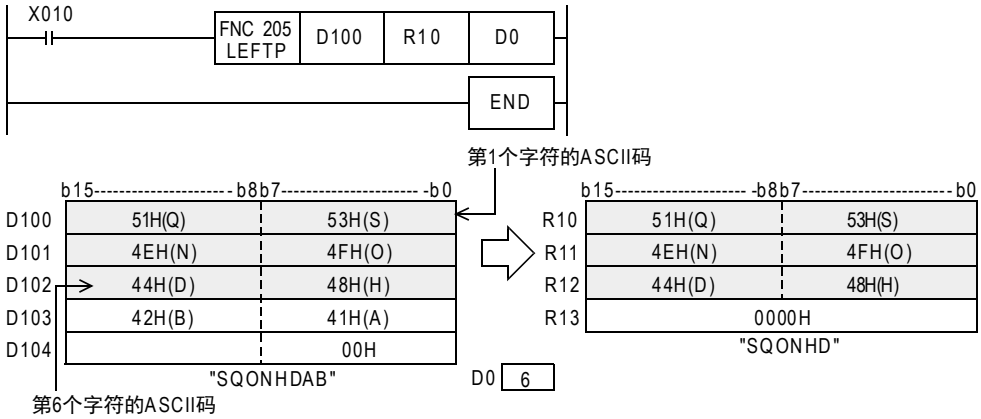
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- (S·) 中指定的软元件开始的相应软元件范围内没有设定[00H]时。
  (错误代码: K6706)
- n超出了 (S·) 中指定的字符数时。(错误代码: K6706)
- (D·) 中指定的软元件编号开始的软元件数，比保存已经取出的字符串 (n 个字符)所需的软元件数更少时。
  (所有的字符串和最终字符后面不能保存[00H])(错误代码: K6706)
- n为负值时。(错误代码: K6706)

程序举例

当X010为ON时，从D100开始的软元件中保存的字符串数据的左侧取出字符数据，字符数据个数为保存在D0中的值，将取出的字符数据保存到R10开始的软元件中的程序。



26.7 FNC 206 - MIDR/从字符串中的任意取出

A8 A8C

概要

取出指定的字符串中任意位置上的字符串的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
7步	MIDR	连续执行型
	MIDRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存字符串的软元件起始编号	字符串
(D・)	保存被取出的字符串的软元件起始编号	
(S2・)	指定要取出的字符的起始位置以及字符数的软元件起始编号 (S2・)　：起始字符位置 (S2・)+1：字符数	BIN16位

3. 对象软元件

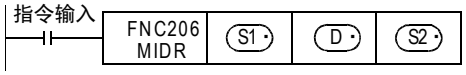
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					
(S2・)								●	●	●	●	●	●	●	●	●			●					

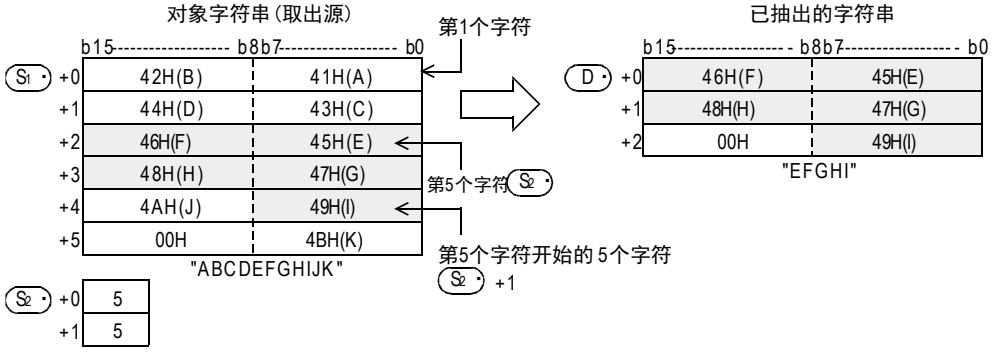
功能和动作说明

1. 16位运算(MIDR/MIDRP)

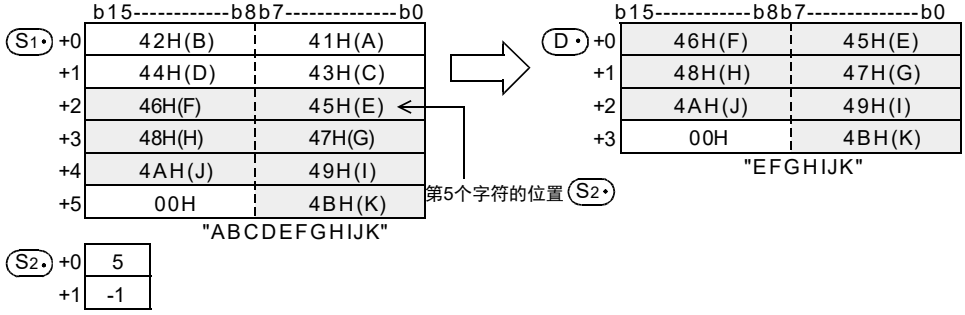
从(S1・)开始的软元件中保存的字符串数据的左侧(字符串的开头)起第[(S2・)]个字符开始，取出[(S2・)+1]个字符数据，保存到(D・)开始的软元件中。
此外，取出字符串时，会在最后自动附加“00H”。

- 取出的字符数[(S2・)+1]为奇数时，在保存最终字符的软元件的高字节中保存“00H”。
- 取出的字符数[(S2・)+1]为偶数时，在保存最终字符的软元件的下一个软元件中保存“0000H”。





- (S1•)中指定的字符串(数据)，就是指从被指定的软元件开始到检测到第1个[00H]为止的数据。
- (S2•)+1中指定的要取出的字符数为“0”时不执行处理。
- (S2•)+1中指定的要取出的字符数为“-1”时，到(S1•)指定的字符串的最终字符数据为止的内容被保存在(D•)中。



注意要点

使用ASCII码以外的字符代码时，请注意以下几点。

- 字符数使用字节单位(8位)。因此,例如使用 SHIFT JIS代码，以2个字节代表1个字符的字符代码的情况下，长度为“2”。
- 从包含了类似SHIFT JIS代码的以2个字节代表1个字符的字符代码的字符串中取出字符串时，请事先考虑一下，以1个字符的字符代码为单位取出的字符数。
  如果从2个字节的字符代码中只取出1个字节时，不能得到期望的字符代码，请务必注意。

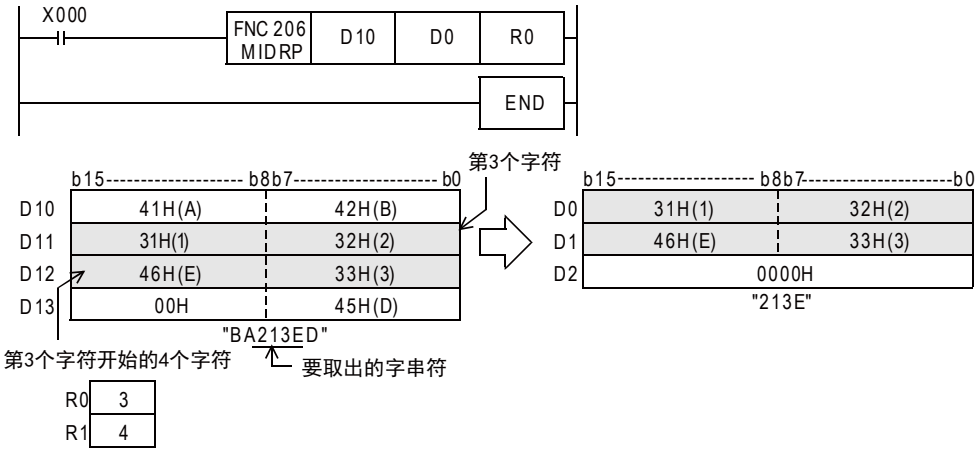
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- (S1•)中指定的软元件开始的相应软元件范围内没有设定[00H]时。
  (错误代码: K6706)
- (S2•)的值超出了(S1•)中指定的字符串的字符数时。(错误代码: K6706)
- (D•)开始的[(S2•)+1]的字符数超出了(D•)的软元件范围时。(错误代码: K6706)
- (D•)中指定的软元件编号开始的软元件数，比保存已取出的字符串 [(S2•)+1] 个字符所需的软元件数更少时。(所有的字符串和最终字符后面不能保存[00H])
  (错误代码: K6706)
- (S2•)n为负值时。(错误代码: K6706)
- (S2•)+1为-2以下的值时。(错误代码: K6706)
- (S2•)+1超出了(S1•)的字符数时。(错误代码: K6706)

程序举例

当X000为ON时，从D10开始的软元件中保存的字符串数据的左侧开始数起，将第3个字符开始的4个字符的数据保存到D0开始的软元件中的程序。



26.8 FNC 207 - MIDW/字符串中的任意替换

A8 A8C

概要

用指定的字符串中任意位置上的字符串去替换指定的字符串的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式

FNC 207 MIDW P
MIDDLE WRITE

16位指令	指令记号	执行条件
7步	MIDW	连续执行型
	MIDWP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1・)	保存要替换的字符串的软元件起始编号	字符串
(D・)	保存替换后的字符串的软元件起始编号	
(S2・)	指定要替换的字符的起始位置以及字符数的软元件起始编号 (S2・) ：被替换的字符串的起始字符位置 (S2・)+1：要替换的字符数	BIN16位

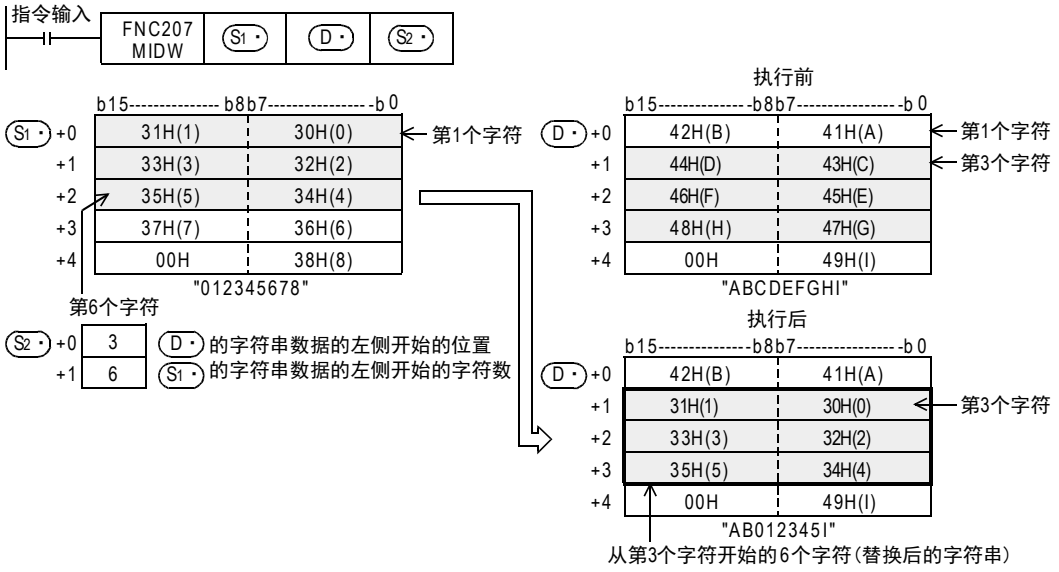
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					
(S2・)								●	●	●	●	●	●	●	●	●			●					

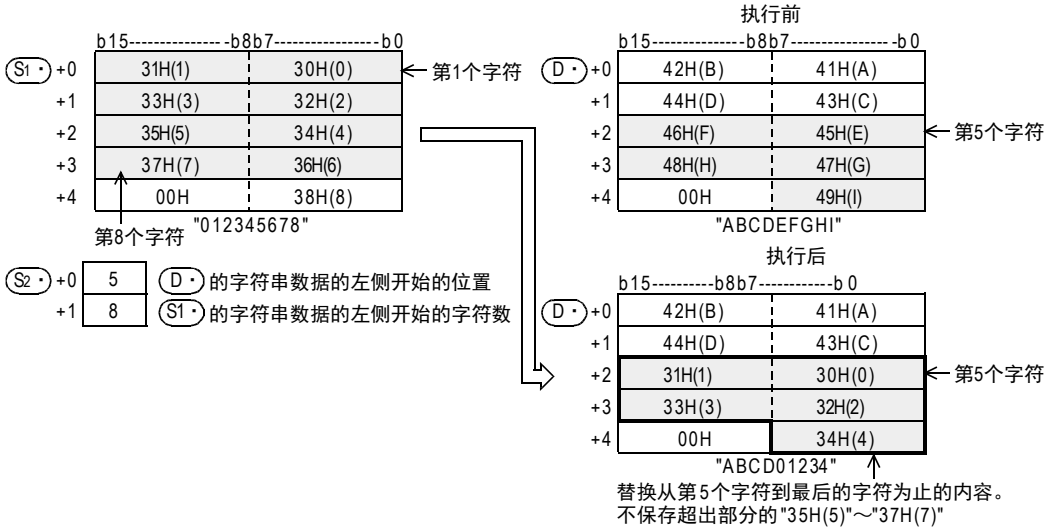
功能和动作说明

1. 16位运算(MIDW/MIDWP)

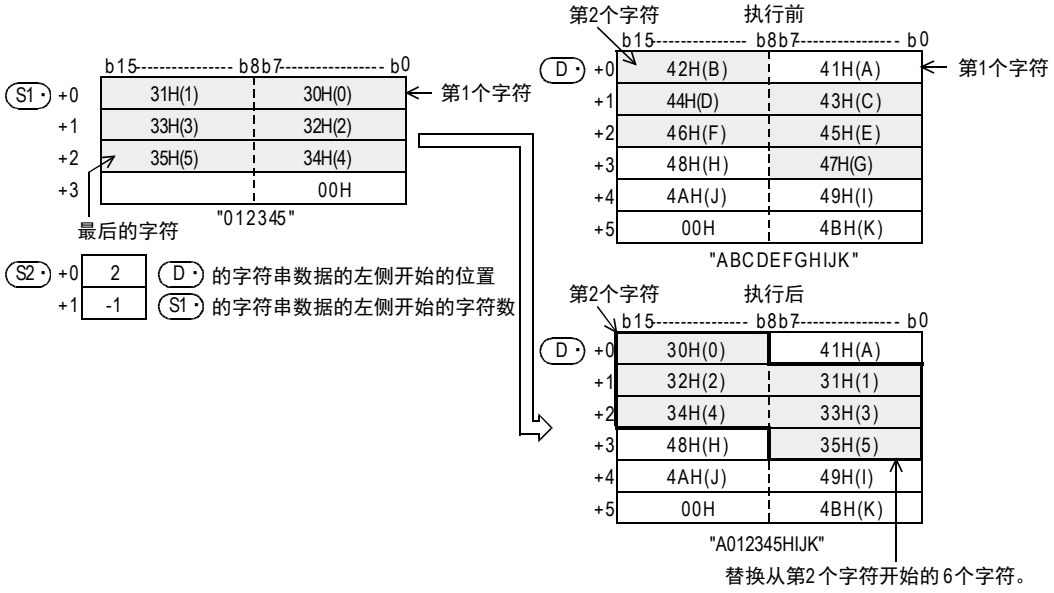
针对(S1・)起始的软元件中保存的字符串数据，将其左侧(字符串的开头)开始的字符数据，(字符数据数为(S2・)+1中指定的字符数)保存到(D・)起始软元件所保存的字符串中，保存位置从字符串数据左起数(S2・)指定的位置开始。



- (S1·)中(D·)被指定的字符串(数据)，就是指从指定的软元件开始到检测开头到第1个[00H]为止的数据。
- (S2·)+1中指定的要替换的字符数为"0"时不执行处理。
- (S2·)+1 中指定的要替换的字符数，一旦超出了(D·)开始的字符串数据的最后字符时，则保存到最后字符为止的数据。



- (S2·)+1 中指定的字符数为“-1”时，到(S1·)指定的最终字符数据为止的内容都被保存到(D·)指定的软元件以后中



注意要点

使用ASCII码以外的字符代码时，请注意以下几点。

- 字符数使用字节单位(8位)。因此,例如使用 SHIFT JIS代码，以2个字节代表1个字符的字符代码的情况下，长度为“2”。
- 从包含了类似SHIFT JIS代码的以2个字节代表1个字符的字符代码的字符串中取出字符串时，请事先考虑一下，以 1个字符的字符代码为单位取出的字符数。
  如果从2个字节的字符代码中只取出1个字节时，不能得到期望的字符代码，请务必注意。

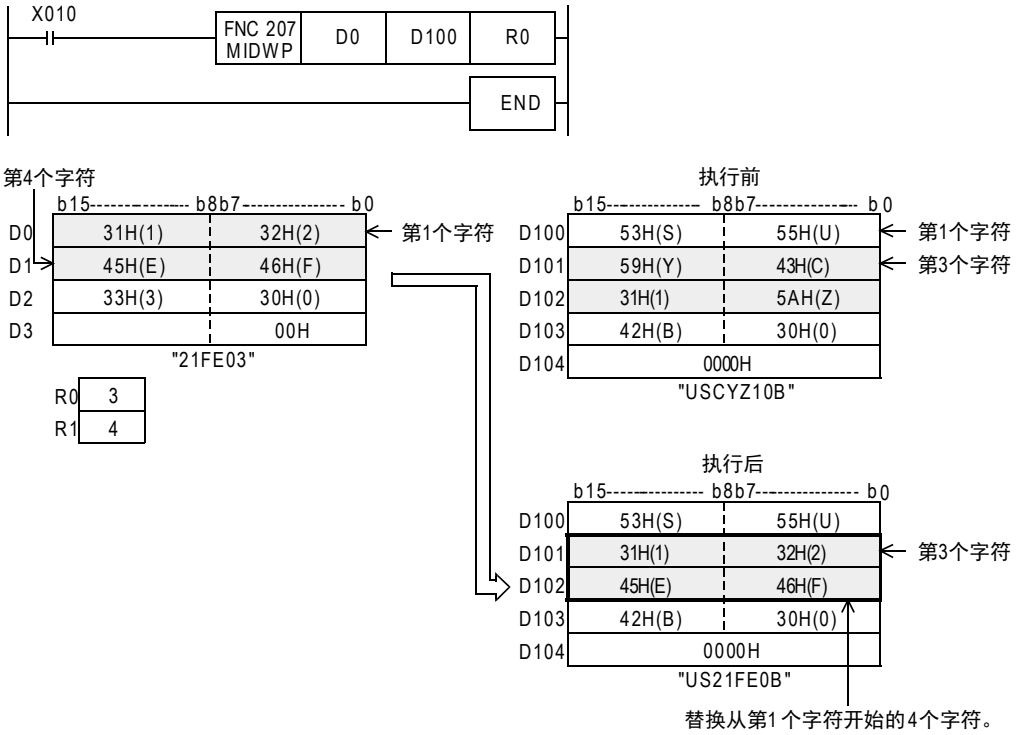
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- (S1・)和(D・)和中指定的软元件开始的相应软元件范围内没有设定[00H]时。
  (错误代码:K6706)
- (S2・)的值超出了(D・)的字符数时。(错误代码:K6706)
- (S2・)+1的值超出了(S1・)的字符数时。(错误代码:K6706)
- (S2・)n为负值时。(错误代码:K6706)
- (S2・)+1为-2以下的值时。(错误代码:K6706)

程序举例

程序为: 当X010为ON时，将D0起始的软元件中保存的字符串数据中的4个字符，保存到D100起始的字符串数据中，保存位置从左起的第3个字符开始。



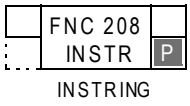
26.9 FNC 208 - INSTR/字符串的检索

A8 A8C

概要

从指定的字符串中检索指定字符串的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	INSTR	连续执行型
	INSTRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1•)	保存要检索的字符串的软元件起始编号	字符串
(S2•)	保存检索源字符串的软元件起始编号	字符串
(D•)	保存检索结果的软元件起始编号	BIN16位
n	开始检索的位置	BIN16位

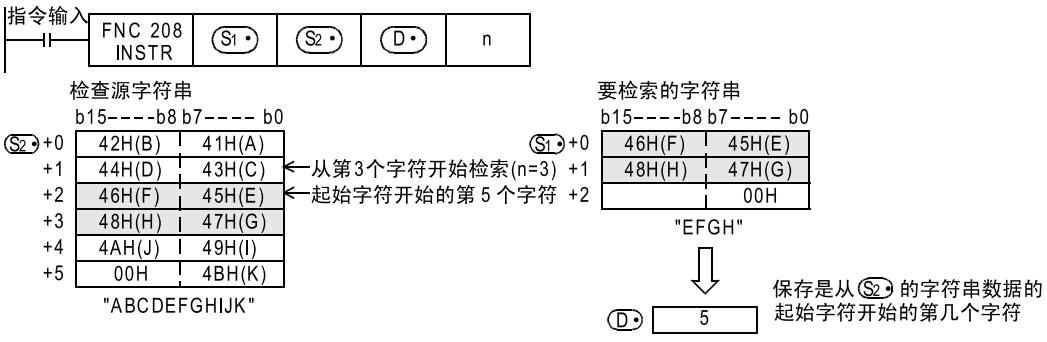
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)												●	●	●	●				●				●	
(S2•)												●	●	●	●				●					
(D•)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(INSTR/INSTRP)

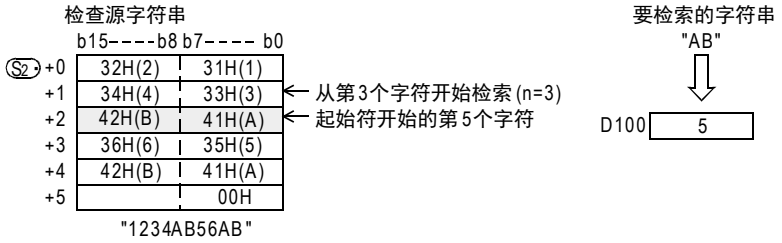
1) 从 (S2•) 起始的软元件中保存的检索源字符串的左起(起始字符)第n个字符开始，检索与保存在 (S1•) 起始的软元件中的字符串相同的字符串，然后将检索结果的字符串位置信息保存到 (D•) 中。
检索的结果，就是检索源字符串左起(起始字符)的第几个字符的字符位置信息。



2) 不存在一致的字符串时，在 (D•) 中保存“0”。

3) 开始检索的位置n为负数或是“0”不执行处理。

4) 在要检索的字符串 (S1・) 中，可以直接指定字符串。



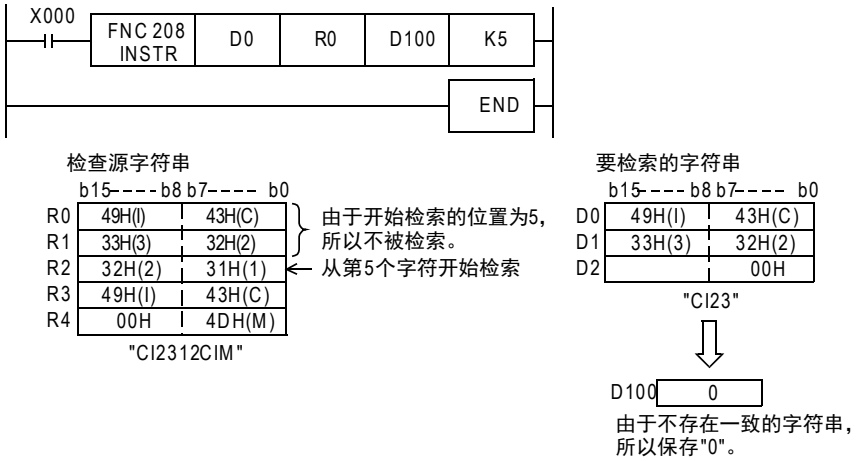
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 开始检索的位置n超过了 (S2・) 的字符数时。(错误代码: K6706)
- (S1・) 开始的相应软元件的软元件范围内没有00H(NULL)时。(错误代码: K6706)
- (S2・) 开始的相应软元件的软元件范围内没有00H(NULL)时。(错误代码: K6706)

程序举例

1) 当X000为ON时，从检索源字符串 “CI2312CIM”(R0以后)的左起(起始字符)第5个字符开始检索要检索的字符串 “CI23”(D0以后)，并将检索结果保存到D100中的程序。



26.10 FNC 209 - $MOV/字符串的传送

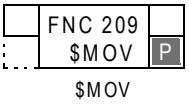
A8 A8C

概要

传送字符串数据的指令。

→ 关于字符串的处理，请参考5.3节

1. 指令格式



16位指令	指令记号	执行条件
5步	$MOV	连续执行型
	$MOVP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	传送源中被直接指定的字符串(最大32个字符)， 或是保存字符串的软元件起始编号	字符串
(D·)	保存传送字符串的软元件起始编号	

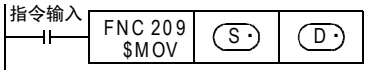
3. 对象软元件

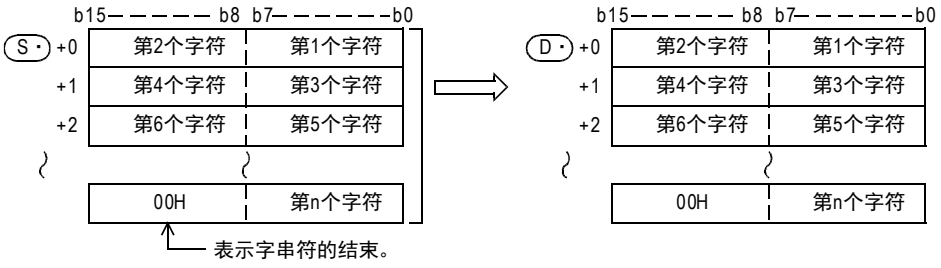
操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●			●				●	
(D·)									●	●	●	●	●	●	●	●			●					

功能和动作说明

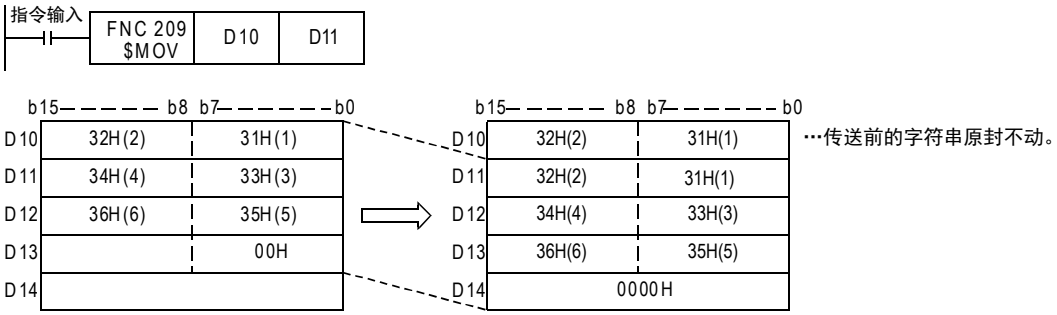
1. 16位运算($MOV/$MOVP)

将(S·)指定的软元件编号起始的软元件中保存的字符串数据，传送到(D·)的软元件编号起始的软元件中。
在传送字符串过程中，从(S·)指定的软元件编号开始，一直到之后的软元件中，其高字节或是低字节中包含“00H”的软元件为止，都一次进行传送。



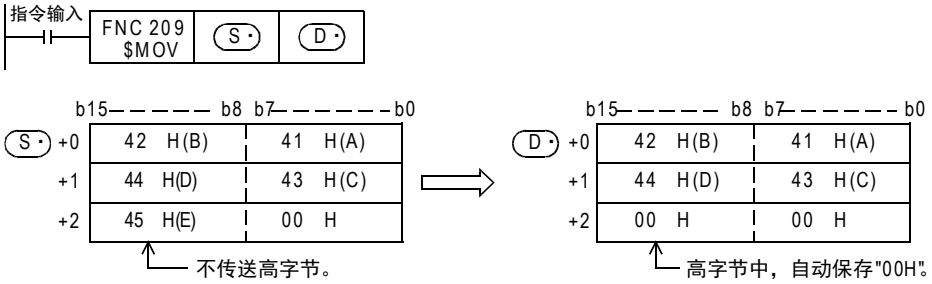


保存要传送的字符串数据的软元件范围[(S・)～(S・)+n]，以及保存被传送的字符串数据的软元件范围[(D・)～(D・)+n/2]使重复时，也能执行传送。
例如，将D10～D13中保存的字符串传送到D11～D14中时，如下所示。



注意要点

(S・)+n的低字节中保存了“00H”时，(D・)+n的高字节、低字节中都保存“00H”。



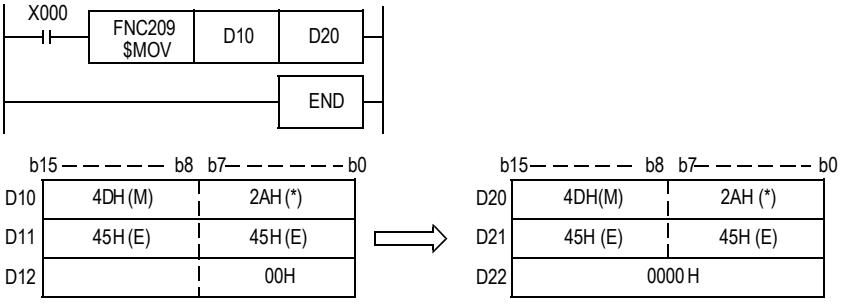
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

- (S・)中指定的软元件编号～指定软元件的最终编号之间不存在“00H”时。
  (错误代码:K6706)
- 在(D・)中指定的软元件编号～指定软元件的最终编号之间的点数中，不能保存所有指定的字符串时。
  (错误代码:K6706)

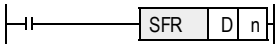
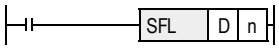
程序举例

当X000为ON时，将保存在D10～D12中的字符串数据传送到D20～D22中的程序。



27. 数据处理3—FNC 210～FNC 219

提供了读取后入的数据，控制带进位的左右移位的指令。

FNC No.	指令记号	符号	功能	参考
210	FDEL		数据表的数据删除	27.1节
211	FINS		数据表的数据插入	27.2节
212	POP		读取后入的数据[先入后出控制用]	27.3节
213	SFR		16位数据n位右移(带进位)	27.4节
214	SFL		16位数据n位左移(带进位)	27.5节
215	—			—
216	—			—
217	—			—
218	—			—
219	—			—

27.1 FNC 210 - FDEL/数据表的数据删除

A8 A8C

概要

删除数据表格中任意数据的指令。

1. 指令格式

FNC 210 FDEL P
DATE TABLE DELETE

16位指令	指令记号	执行条件
7步	FDEL	连续执行型
	FDELP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存被删除的数据的软元件编号	BIN16位
(D・)	数据表格的起始软元件编号	
n	要删除的数据的表格位置	

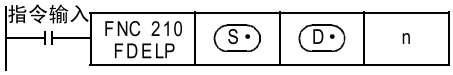
3. 对象软元件

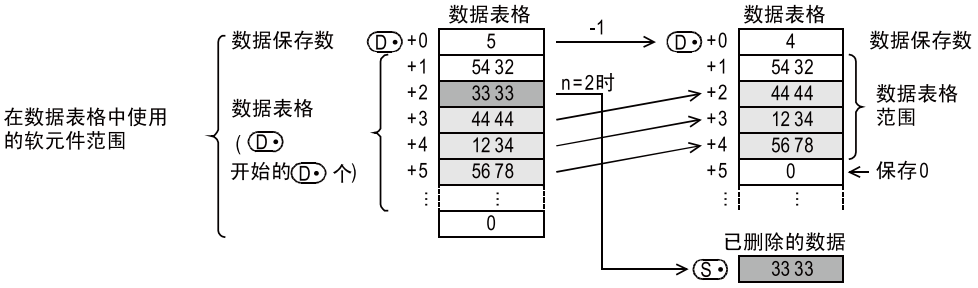
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(FDEL/FDELP)

删除数据表格((D・) 起始) 的第n个数据，将删除的数据保存到 (S・) 中。数据表格的第 n+1 个开始的数据逐个向前靠拢，数据保存数减1。





注意要点

- 请用户自行管理数据表格中使用的软元件范围。
  数据表格的范围，为数据保存数 (D・) 的下一个软元件((D・) +1)开始的 (D・) 个。

→ 参考程序举例

相关指令

指令	内容
FINS(FNC 211)	在数据表格的任意位置中插入数据的指令。

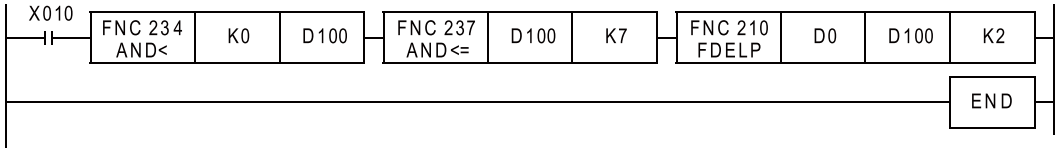
错误

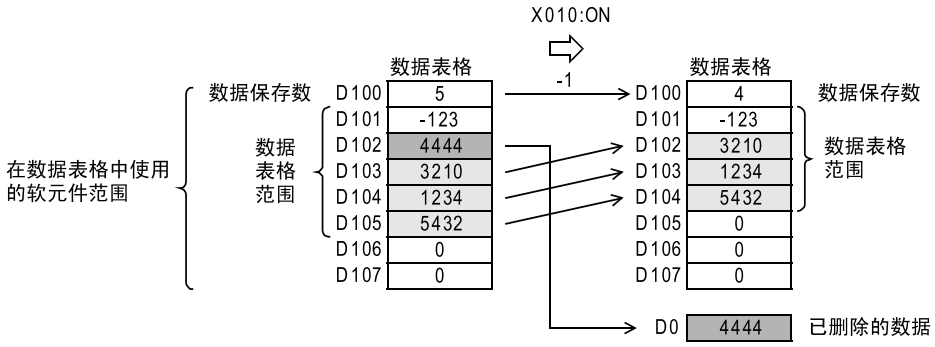
以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 要删除的数据的表格位置 n 比数据保存数更大时。(错误代码: K6706)
- n的值超出了数据表格 (D•) 的软元件范围时。(错误代码 K6706)
- 在n≦0的情况执行了指令时。(错误代码 : K6706)
- 数据保存数 (D•) 的值为0时。(错误代码 : K6706)
- 数据表格的范围超出了相应的软元件范围时。(错误代码: K6706)

程序举例

当X010为ON时，删除D100～D105的数据表格中的第2个数据，将删除的数据保存到D0中的程序。
但是，当数据保存数为0时，请勿执行FDEL(FNC 210)指令。(数据表格中使用的软元件范围为D100～D107。)





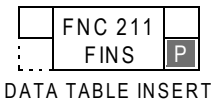
27.2 FNC 211 - FINS/数据表的数据插入

A8 A8C

概要

在数据表格中的任意位置处插入数据的指令。

1. 指令格式



16位指令	指令记号	执行条件
7步	FINS	连续执行型
	FINSP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存插入数据的软元件编号	BIN16位
(D・)	数据表格的起始软元件编号	
n	插入数据的表格位置	

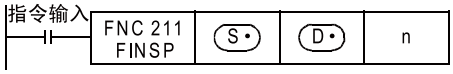
3. 对象软元件

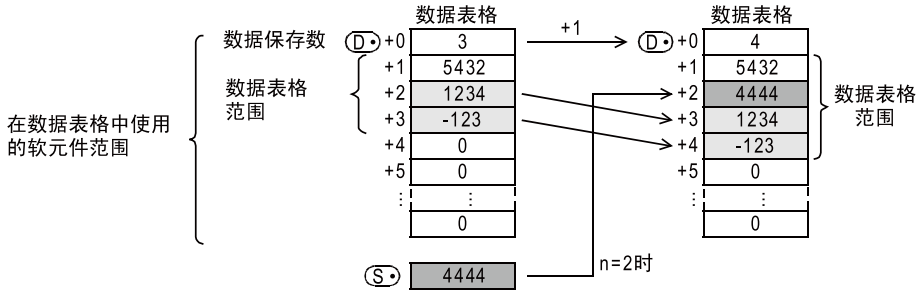
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	E	P
(S・)												●	●	●	●				●	●	●			
(D・)												●	●	●	●				●					
n														●	●					●	●			

功能和动作说明

1. 16位运算(FINS/FINSP)

1) 将16位数据 (S・) 插入到数据表格((D・) 以后)的第n号中。数据表格的第n号以后的数据逐个后移，数据保存数加1。





注意要点

- 请用户自行管理数据表格中使用的软元件范围。
  数据表格的范围，为数据保存数 (D・) 的下一个软元件((D・) +1)开始的 (D・) 个。

→ 参考程序举例

相关指令

指令	内容
FDEL(FNC 210)	删除数据表格的任意数据的指令。

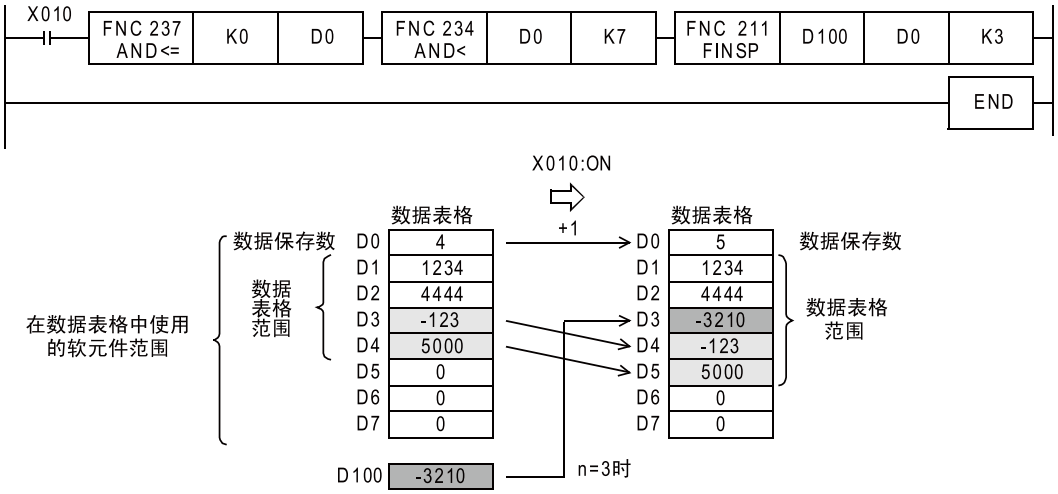
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 要插入数据的表格位置 n 比数据保存数+1更大时。(错误代码: K6706)
- n的值超出了数据表格 (D•) 的软元件范围时。(错误代码 : K6706)
- 在n≦0的情况执行了指令时。(错误代码: K6706)
- 数据表格的范围超出了相应的软元件范围时。(错误代码: K6706)

程序举例

当X010为ON时，在D0～D4的数据表格的第3号中插入D100的数据的程序。
但是，当数据保存数超过7时，不执行FINS(FNC 211)指令。
(数据表格中使用的软元件范围为D0～D7。)



27.3 FNC 212 - POP/读取后入的数据[先入后出控制用]

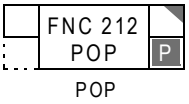
A8 A8C

概要

该指令用于读出使用先入后出控制用的移位写入指令(SFWR)，写入最后的数据。

→ 关于SFWR(FNC 38)指令，请参考11.9节

1. 指令格式



16位指令	指令记号	执行条件
7步	POP	连续执行型
	POPP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

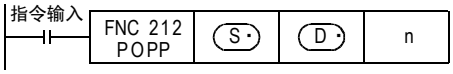
操作数种类	内容	数据类型
(S・)	保存先入数据(包含指针数据)的起始软元件编号 (保存数据的起始字软元件编号)	BIN16位
(D・)	保存后出的数据的软元件编号	
n	被保存的数据的点数 (由于包含了指针数据，所以请设置为+1后的值。) 2≦n≦512	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)									●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●	●	●	●					
n																				●	●			

功能和动作说明

1. 16位运算(POP/POPP)

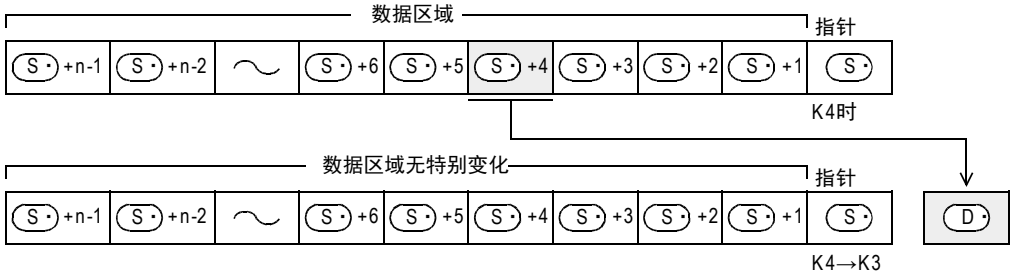


先入后出控制用数据

	内容
(S・)	指针数据(被保存的数据个数)
(S・)+1	数据区域 (使用移位写入指令(SFWR)被先入的数据)
(S・)+2	
(S・)+3	
≀	
(S・)+n-3	
(S・)+n-2	
(S・)+n-1	

- 对于 [(S・)～(S・)+n-1] 的字软元件，每次执行指令时，读出 [(S・)+指针数据(S・)] 的软元件保存到(D・)中(使用先入先出控制用的移位写入指令(SFWR)写入的最后的数据读出到(D・)中)。n可以指定2～512。

- 指针数据 (S·) 的值减



相关软元件

→ 有关零标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8020	零位	当指针 (S·) =0时，执行指令后变为ON。

相关指令

指令	内容
SFWR(FNC 38)	移位写入[先入先出/先入后出控制用]
SFRD(FNC 39)	移位读出[先入先出控制用]

注意要点

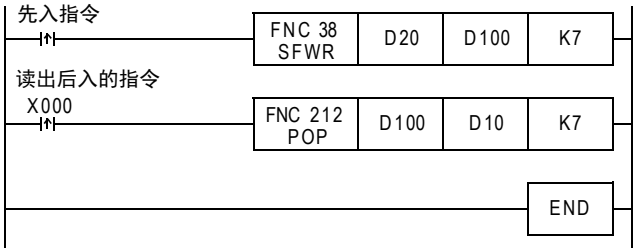
- 用连续执行型编写该指令时，每个运算周期都会处理指令，所以请注意有时可能会出现意外的动作。
  通常，编程时使用[脉冲执行型]或是通过[脉冲化的指令触点]执行。
- 指针 (S·) 的当前值为0时，零位标志位M8020为ON，不处理指令。
  此时，请先使用比较指令确认 (S·) 的当前值是否 ≦ (S·) ≦(n-1)，然后执行这个指令。
- 指针 (S·) 的当前值为1时，(S·) 中被写入0，零位标志位M8020为ON。

错误

- 以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。
  - (S·) ＞n-1时(错误代码: K6706)
  - (S·) ＜0时(错误代码: K6706)

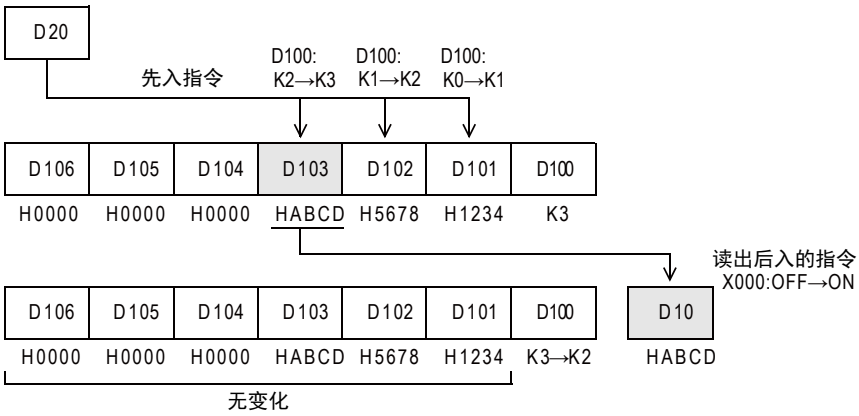
程序举例

X000每次为ON时，对于D101～D106中被先输入的D20的值中，最后被保存的值都会保存到D10中，然后将数据保存数(指针D100)减1的程序。



先输入的数据为下表中的内容时

指针	D100	K3
数据	D101	H1234
	D102	H5678
	D103	HABCD
	D104	H0000
	D105	H0000
	D106	H0000



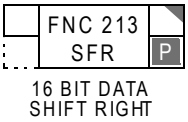
27.4 FNC 213 - SFR/16位数据n位右移(带进位)

A8 A8C

概要

使字软元件中的16位向右移动n位的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	SFR	连续执行型
	SFRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

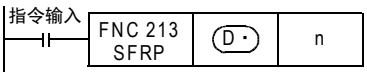
操作数种类	内容	数据类型
(D·)	保存要移动的数据的软元件编号	BIN16位
n	移动的次数0≦n≦15	

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址				常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰		K	H	E	"□"	P
(D·)									●	●	●	●	●	●	●	●	●	●	●						
n								●	●	●	●	●	●	●	●	●	●	●			●	●			

功能和动作说明

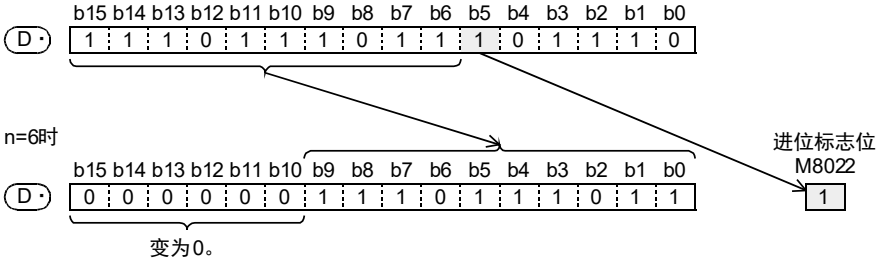
1. 16位运算(SFR/SFRP)



1) 字软元件 (D·) 中的16位右移n位。
n指定0～15的数字。
当n中指定了16以上的数值时，根据n/16的余数移动。例如，如n=18时，18/16=1余2，所以右移2位。

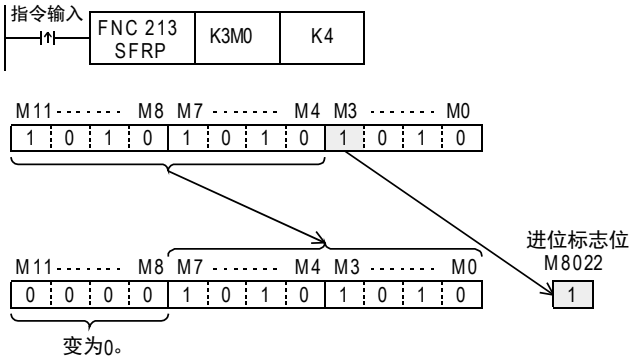
2) 将字软元件 (D·) 中的第n位(n-1位)的ON(1)/OFF(0)状态转移到进位标志位M8022中。

3) 最高位开始的n位变为0。



通过位数指定来指定位软元件时

将指定位数(4×K□)的数据进行移位。



相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	移动n-1的状态(ON/OFF)。

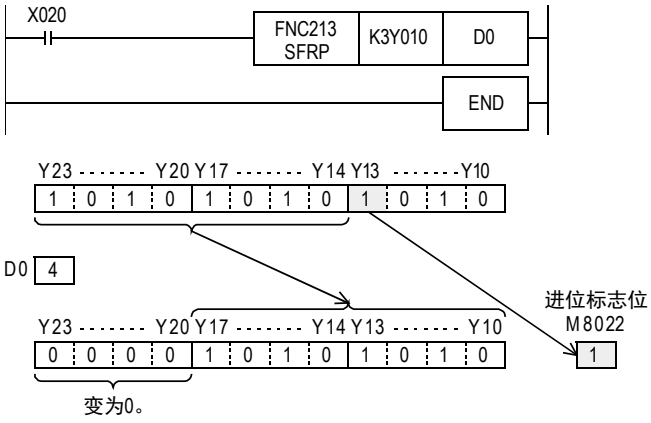
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- n指定了负值时。(错误代码: K6706)

程序举例

当X020为ON时，将Y010～Y023的内容按照D0中指定的位数右移的程序。



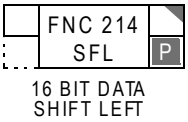
27.5 FNC 214 - SFL/16位数据 n 位左移(带进位)

A8 A8C

概要

使字软元件中的16位向左移动n位的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	SFL	连续执行型
	SFLP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

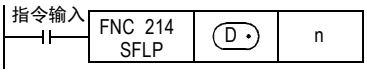
操作数种类	内容	数据类型
(D·)	保存要移动的数据的软元件编号	BIN16位
n	移动的次数0≦n≦15	

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址				常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰		K	H	E	"□"	P
(D·)									●	●	●	●	●	●	●	●	●	●	●						
n								●	●	●	●	●	●	●	●	●	●	●			●	●			

功能和动作说明

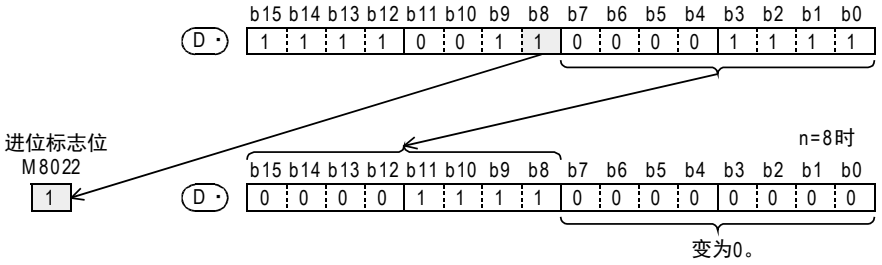
1. 16位运算(SFL/SFLP)



1) 字软元件 (D·) 中的16位左移n位。
n指定0～15的数字。
当n中指定了16以上的数值时，根据n/16的余数移动。例如，如n=18时，18/16=1余2，所以左移2位。

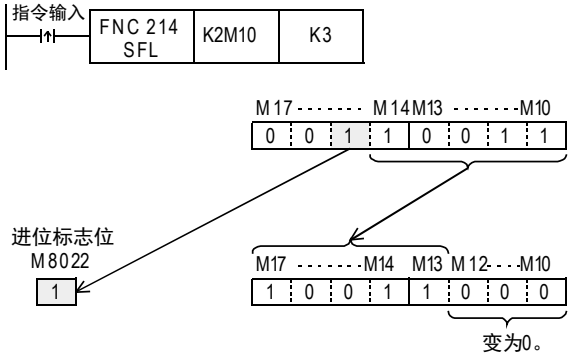
2) 将字软元件 (D·) 中的第n+1位(n位)的ON(1)/OFF(0)状态转移到进位标志位M8022中。

3) 最低位开始的n位变为0。



通过位数指定来指定位软元件时

将指定位数(4×K□)的数据进行移位。



相关软元件

→ 有关进位标志位的使用方法，参考6.5.2项

软元件	名称	内容
M8022	进位	移动n的状态(ON/OFF)。

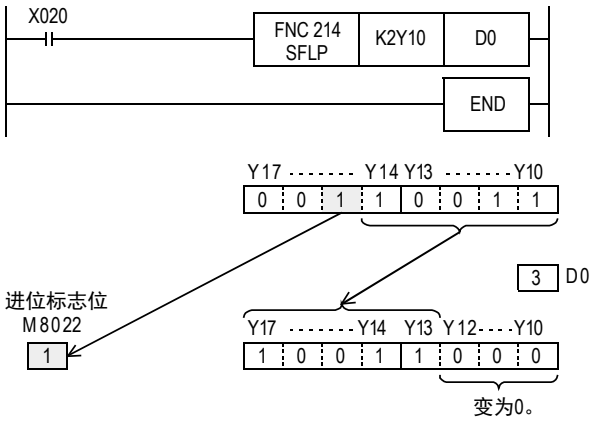
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- n指定了负值时。(错误代码: 6706)

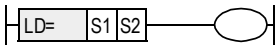
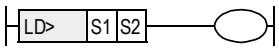
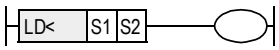
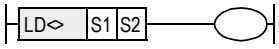
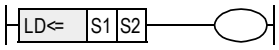
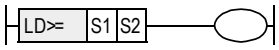
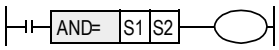
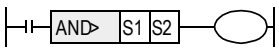
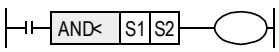
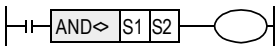
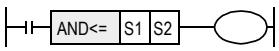
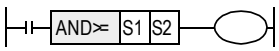
程序举例

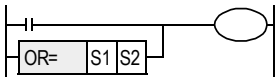
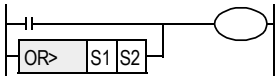
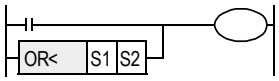
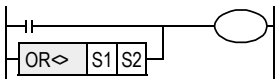
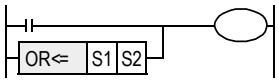
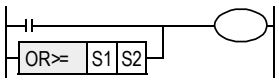
当X020为ON时，将Y010～Y017的内容按照D0中指定的位数左移的程序。



28. 触点比较指令—FNC 220～FNC 249

FNC 220～FNC 249中提供了使用RLD、AND、OR触点符号进行数据比较的指令。

FNC No.	指令记号	符号	功能	参考
220	—			—
221	—			—
222	—			—
223	—			—
224	LD=		触点比较LD (S 1) = (S 2)	28.1节
225	LD＞		触点比较LD (S 1) ＞ (S 2)	28.1节
226	LD＜		触点比较LD (S 1) ＜ (S 2)	28.1节
227	—			—
228	LD＜＞		触点比较LD (S 1) ≠ (S 2)	28.1节
229	LD＜=		触点比较LD (S 1) ≦ (S 2)	28.1节
230	LD＞=		触点比较LD (S 1) ≧ (S 2)	28.1节
231	—			—
232	AND=		触点比较AND (S 1) = (S 2)	28.2节
233	AND＞		触点比较AND (S 1) ＞ (S 2)	28.2节
234	AND＜		触点比较AND (S 1) ＜ (S 2)	28.2节
235	—			—
236	AND＜＞		触点比较AND (S 1) ≠ (S 2)	28.2节
237	AND＜=		触点比较AND (S 1) ≦ (S 2)	28.2节
238	AND＞=		触点比较AND (S 1) ≧ (S 2)	28.2节

FNC No.	指令记号	符号	功能	参考
239	—			—
240	OR=		触点比较OR (S1) = (S2)	28.3节
241	OR＞		触点比较OR (S1) ＞ (S2)	28.3节
242	OR＜		触点比较OR (S1) ＜ (S2)	28.3节
243	—			—
244	OR＜＞		触点比较OR (S1) ≠ (S2)	28.3节
245	OR＜=		触点比较OR (S1) ≦ (S2)	28.3节
246	OR＞=		触点比较OR (S1) ≧ (S2)	28.3节
247	—			—
248	—			—
249	—			—

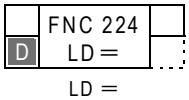
28.1 FNC 224 ～230 — LD=、＞、＜、＜＞、＜＝、＞＝/触点比较

A1 A2 A2C A8 A8C

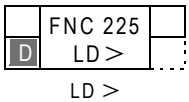
概要

执行数值的比较，当条件满足时使触点置ON的触点比较运算开始的指令。

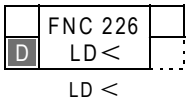
1. 指令格式



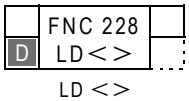
16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＝	连续执行型	9步	LDD＝	连续执行型



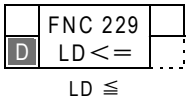
16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＞	连续执行型	9步	LDD＞	连续执行型



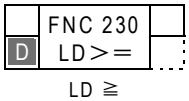
16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＜	连续执行型	9步	LDD＜	连续执行型



16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＜＞	连续执行型	9步	LDD＜＞	连续执行型



16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＜＝	连续执行型	9步	LDD＜＝	连续执行型



16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	LD＞＝	连续执行型	9步	LDD＞＝	连续执行型

2. 设定数据(FNC 224～FNC 230通用)

操作数种类	内容	数据类型
(S1•)	保存比较数据的软元件编号	BIN16/32位
(S2•)	保存比较数据的软元件编号	BIN16/32位

3. 对象软元件(FNC 224～FNC 230通用)

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

连接在母线上的触点比较指令。

对 (S1・)、(S2・) 的内容进行BIN比较，根据其结果来控制触点的导通或是不导通。

FNC №	16位指令	32位指令	导通条件	不导通条件
224	LD=	LDD=	(S1) = (S2)	(S1) ≠ (S2)
225	LD＞	LDD＞	(S1) ＞ (S2)	(S1) ≦ (S2)
226	LD＜	LDD＜	(S1) ＜ (S2)	(S1) ≧ (S2)
228	LD＜＞	LDD＜＞	(S1) ≠ (S2)	(S1) = (S2)
229	LD＜=	LDD＜=	(S1) ≦ (S2)	(S1) ＞ (S2)
230	LD＞=	LDD＞=	(S1) ≧ (S2)	(S1) ＜ (S2)

注意要点

1. 关于负数

   (S1・)、(S2・) 的数据最高位为1时，将其值当作负数进行比较。

   - 16位运算时: b15
   - 32位运算时: b31

2. 使用32位计数器(包含高速计数器)时

   32位计数器(C200～C255)的比较必须用32位(LDD=、LDD＞、LDD＜等)运算执行。

   如指定了16位运算(LD=、LD＞、LD＜等)时，会发生程序错误或是运算错误。

3. 触点比较指令的编程

   使用HCP-works编程时，不能输入≦和≧符号。

   ≦是分成＜和=，≧是分成＞和=进行输入的。

   输入操作如下所示。

   操作步骤

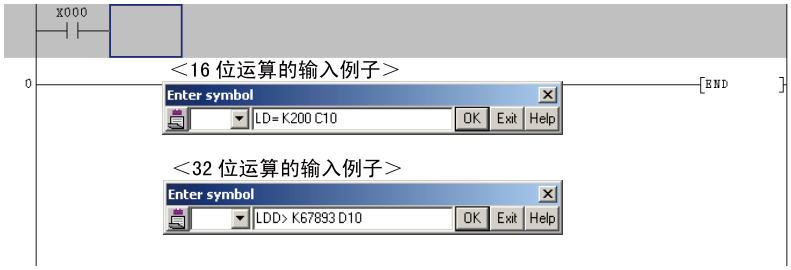
   a) 在[显示梯形图程序的编辑画面]中，将光标停在要使用触点比较指令的位置。

   b) 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子，请参考下面的[HCP-works的指令输入画面]。

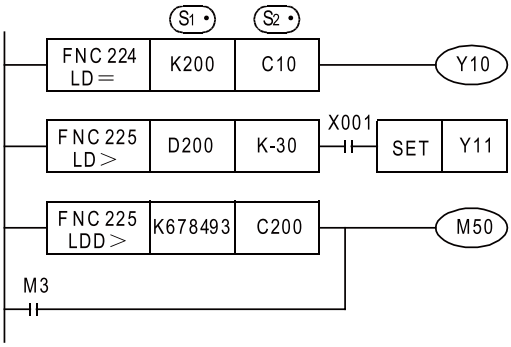
   c) 点击[OK]键。

   d) 接着，请继续输入其他的触点以及线圈部分。

   HCP-works的指令输入画面



程序举例



计数器C10的当前值为200时，Y010被驱动。

D200的内容为-29以上，且X001为ON时Y011被置位。

计数器C200的内容比K678,493小时，或是M3为ON时，M50被驱动。

28.2 FNC 232 ~238 — AND=、>、<、<>、<=、>=/触点比较

A1 A2 A2C A8 A8C

概要

执行数值的比较，当条件满足时使触点置ON的触点比较运算指令。

1. 指令格式

	16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
FNC 232 D AND= AND =	5步	AND=	连续执行型	9步	ANDD=	连续执行型
FNC 233 D AND> AND >	5步	AND>	连续执行型	9步	ANDD>	连续执行型
FNC 234 D AND< AND <	5步	AND<	连续执行型	9步	ANDD<	连续执行型
FNC 236 D AND<> AND <>	5步	AND<>	连续执行型	9步	ANDD<>	连续执行型
FNC 237 D AND<= AND ≦	5步	AND<=	连续执行型	9步	ANDD<=	连续执行型
FNC 238 D AND>= AND ≧	5步	AND>=	连续执行型	9步	ANDD>=	连续执行型

2. 设定数据(FNC 232～FNC 238通用)

操作数种类	内容	数据类型
(S1•)	保存比较数据的软元件编号	BIN16/32位
(S2•)	保存比较数据的软元件编号	BIN16/32位

3. 对象软元件(FNC 232～FNC 238通用)

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1•)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2•)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

与其他触点串联的触点比较指令。

对 (S1·)、(S2·) 的内容进行BIN比较，根据其结果控制触点的导通或是不导通。

FNC №	16位指令	32位指令	导通条件	不导通条件
232	AND=	ANDD=	(S1·) = (S2·)	(S1·) ≠ (S2·)
233	AND＞	ANDD＞	(S1·) ＞ (S2·)	(S1·) ≦ (S2·)
234	AND＜	ANDD＜	(S1·) ＜ (S2·)	(S1·) ≧ (S2·)
236	AND＜＞	ANDD＜＞	(S1·) ≠ (S2·)	(S1·) = (S2·)
237	AND＜＝	ANDD＜＝	(S1·) ≦ (S2·)	(S1·) ＞ (S2·)
238	AND＞＝	ANDD＞＝	(S1·) ≧ (S2·)	(S1·) ＜ (S2·)

注意要点

1. 关于负数

   (S1·)、(S2·) 的数据最高位为1时，将其值当作负数进行比较。

   - 16位运算时:b15
   - 32位运算时:b31

2. 使用32位计数器(包含高速计数器)时

   32位计数器(C200～255)的比较必须用32位(ANDD=、ANDD＞、ANDD＜等)运算执行。如指定了 16位运算(AND=、AND＞、AND＜等)时，会发生程序错误或是运算错误。

3. 触点比较指令的编程

   使用HCP-works编程时，不能输入≦和≧符号。

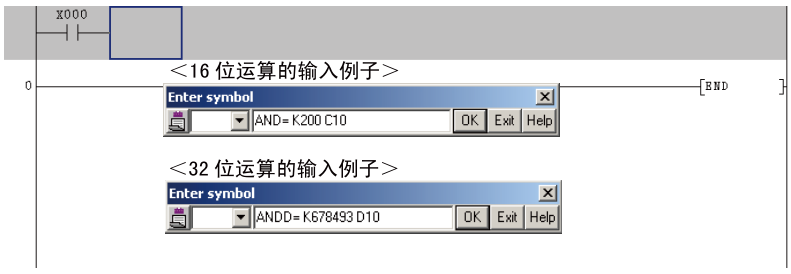
   ≦是分成＜和=，≧是分成＞和=进行输入的。

   输入操作如下所示。

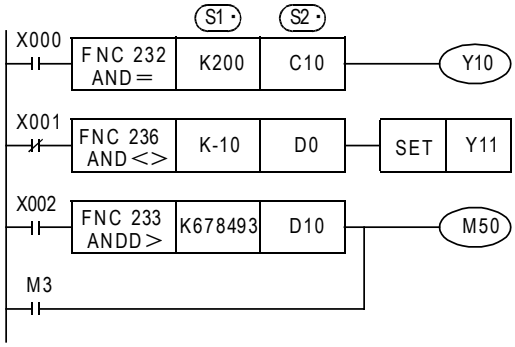
   操作步骤

   a) 在[显示梯形图程序的编辑画面]中，将光标停在要使用触点比较指令的位置。
   b) 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子，请参考下面的[HCP-works的指令输入画面]。
   c) 点击[OK]键。
   d) 接着，请继续输入其他的触点以及线圈部分。

   HCP-works的指令输入画面



程序举例



X000为ON，且计数器C10的当前值为200时，Y010被驱动。

X001为OFF,且数据计数器DO的内容不为-10时，Y011被置位。

X002为ON,且数据计数器D11、D10的内容比K678,493小时，或是M3为ON时，M50被驱动。

28.3 FNC 240 ～246 — OR=、＞、＜、＜＞、＜＝、＞＝/触点比较

A1 A2 A2C A8 A8C

概要

执行数值的比较，当条件满足时使触点置ON的触点比较运算指令。

1. 指令格式

FNC 240 OR＝ (D)
OR ＝

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＝	连续执行型	9步	ORD＝	连续执行型

FNC 241 OR＞ (D)
OR ＞

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＞	连续执行型	9步	ORD＞	连续执行型

FNC 242 OR＜ (D)
OR ＜

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＜	连续执行型	9步	ORD＜	连续执行型

FNC 244 OR＜＞ (D)
OR ＜＞

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＜＞	连续执行型	9步	ORD＜＞	连续执行型

FNC 245 OR＜＝ (D)
OR ≦

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＜＝	连续执行型	9步	ORD＜＝	连续执行型

FNC 246 OR＞＝ (D)
OR ≧

16位指令	指令记号	执行条件	32位指令	指令记号	执行条件
5步	OR＞＝	连续执行型	9步	ORD＞＝	连续执行型

2. 设定数据(FNC 240～FNC 246通用)

操作数种类	内容	数据类型
(S1·)	保存比较数据的软元件编号	BIN16/32位
(S2·)	保存比较数据的软元件编号	BIN16/32位

3. 对象软元件(FNC 240～FNC 246通用)

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			
(S2·)								●	●	●	●	●	●	●	●	▲	●	●	●	●	●			

▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

与其他触点并联的触点比较指令。

对 (S1・)、(S2・) 的内容进行BIN比较，根据其结果来控制触点的导通或是不导通。

FNC №	16位指令	32位指令	导通条件	不导通条件
240	OR=	ORD=	(S1・)=(S2・)	(S1・)≠(S2・)
241	OR＞	ORD＞	(S1・)＞(S2・)	(S1・)≦(S2・)
242	OR＜	ORD＜	(S1・)＜(S2・)	(S1・)≧(S2・)
244	OR＜＞	ORD＜＞	(S1・)≠(S2・)	(S1・)=(S2・)
245	OR＜=	ORD＜=	(S1・)≦(S2・)	(S1・)＞(S2・)
246	OR＞=	ORD＞=	(S1・)≧(S2・)	(S1・)＜(S2・)

注意要点

1. 关于负数

   (S1・)、(S2・) 的数据最高位为1时，将其值作为负数进行比较。

   - 16位运算时:b15
   - 32位运算时:b31

2. 使用32位计数器(包含高速计数器)时

   32位计数器(C200～C255)的比较必须用32位(ORD=、ORD＞、ORD＜等)运算执行。如指定了16位运算(ORD=、OR＞、OR＜等)时会发生程序错误或是运算错误。

3. 触点比较指令的编程

   使用HCP-works编程时，不能输入≦和≧符号。

   ≦是分成＜和=，≧是分成＞和=进行输入的。

   输入操作如下所示。

   操作步骤

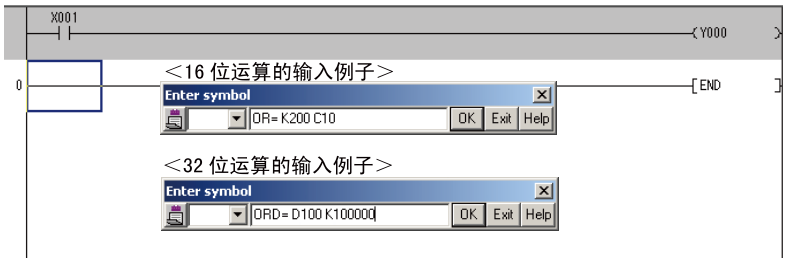
   a) 在[显示梯形图程序的编辑画面]中，将光标停在要使用触点比较指令的位置。

   b) 输入[指令]→空格→[比较值或是软元件]→空格→[比较值或是软元件]。关于输入的例子，请参考下面的[HCP-works的指令输入画面]。

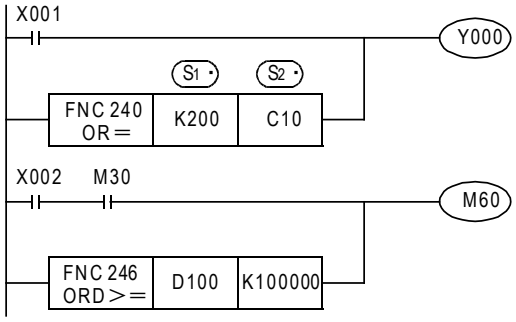
   c) 点击[OK]键。

   d) 接着，请继续输入其他的触点以及线圈部分。

   HCP-works的指令输入画面



程序举例



X001为ON时，或是计数器C10的当前值为200时，Y000被驱动。

X002以及M30为ON时，或是数据寄存器D101，D100 的内容为K100,000以上时，M60被驱动。

29. 数据表处理—FNC 250～FNC 269

FNC No.	指令记号	符号	功能	参考
250	—			—
251	—			—
252	—			—
253	—			—
254	—			—
255	—			—
256	LIMIT	LIMIT S1 S2 S3 D	上下限限位控制	29.1节
257	BAND	BAND S1 S2 S3 D	死区控制	29.2节
258	ZONE	ZONE S1 S2 S3 D	区域控制	29.3节
259	SCL	SCL S1 S2 D	定坐标(不同点坐标数据)	29.4节
260	DABIN	DABIN S D	10进制ASCII→BIN的转换	29.5节
261	BINDA	BINDA S D	BIN→10进制ASCII的转换	29.6节
262	—			—
263	—			—
264	—			—
265	—			—
266	—			—
267	—			—
268	—			—
269	SCL2	SCL2 S1 S2 D	定坐标2(X/Y坐标数据)	29.7节

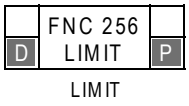
29.1 FNC256 - LIMIT/上下限限位控制

A8 A8C

概要

设置输入数值的上限值/下限值然后输出的指令。

1. 指令格式



16位指令	指令记号	执行条件
9步	LIMIT	连续执行型
	LIMITP	脉冲执行型

32位指令	指令记号	执行条件
17步	DLIMIT	连续执行型
	DLIMITP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	下限限位值(最小输出界限值)	BIN16/32位
(S2·)	上限限位值(最大输出界限值)	
(S3·)	需要通过上下限限位控制的输入值	
(D·)	保存已经过上下限限位控制的输出值的软元件起始编号	

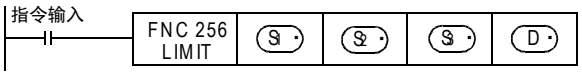
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2·)								●	●	●	●	●	●	●	●	●			●	●	●			
(S3·)								●	●	●	●	●	●	●	●	●			●					
(D·)									●	●	●	●	●	●	●	●			●					

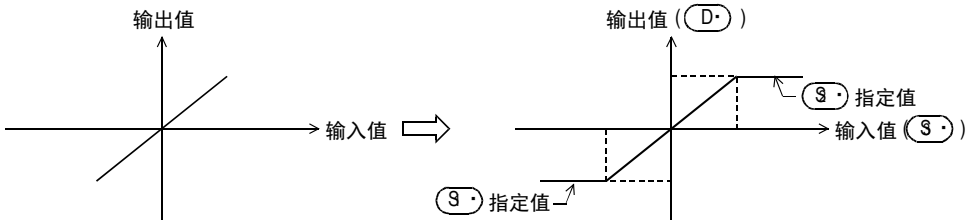
功能和动作说明

1. 16位运算(LIMIT/LIMITP)

通过判断(S3·)中指定的输入值(BIN16位值)，是否在(S1·)、(S2·)指定的上下限值的范围内，以此控制保存在(D·)指定的软元件中的输出值。
输出值如下所示被控制。



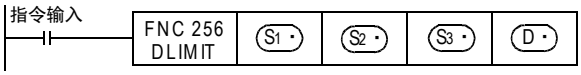
- (S1·) 下限值 > (S3·) 输入值 时……(S3·) 下限值 → (D·) 输出值
- (S2·) 上限值 < (S3·) 输入值 时……(S3·) 上限值 → (D·) 输出值
- (S1·) 下限值 ≦ (S3·) 输入值 ≦ (S2·) 上限值 时……(S3·) 输入值 → (D·) 输出值

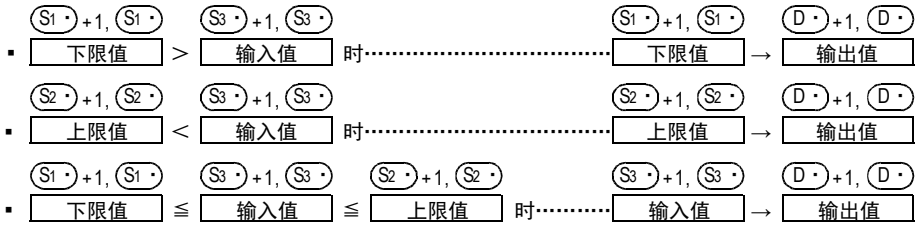


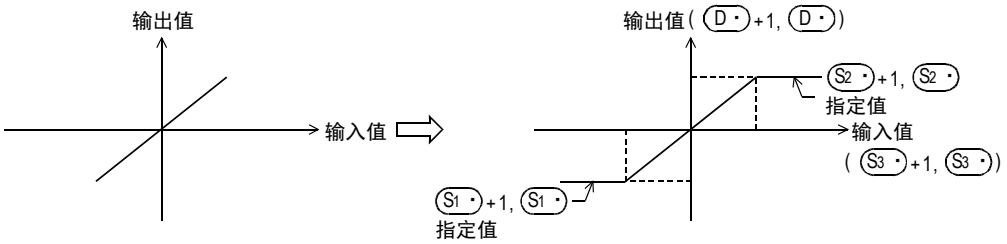
- 仅通过上限限位值进行控制时，在(S1·)指定的下限限位值中设定“-32768”。
- 仅通过下限限位值进行控制时，在(S2·)指定的上限限位值中设定“32767”。

2. 32位运算(DLIMIT/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・)]指定的上下限值的范围内，以此控制保存在[(D・)+1,(D・)]指定的软元件中的输出值。







- 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“-2,147,483,648”。
- 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“2,147,483,647”。

错误

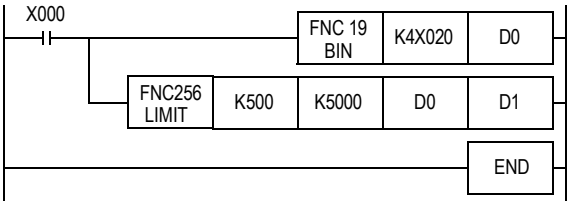
在下述设定状态下执行指令后，会出现运算错误，错误标志位(M8067)为ON，错误代码(K6706)保存在D8067中。

	大小关系
16位运算	(S1・)>(S2・)
32位运算	[(S1・)+1,(S1・)]>[(S2・)+1,(S2・)]

程序举例

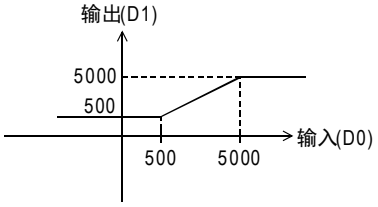
1. 程序举例1

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。



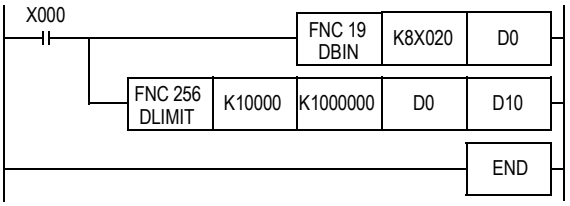
动作

- D0 ＜ 500时，D1为500。
- 500 ≦ D0 ≦ 5000时，D1为D0的值。
- 5000 ＜ D0时，D1为5000。



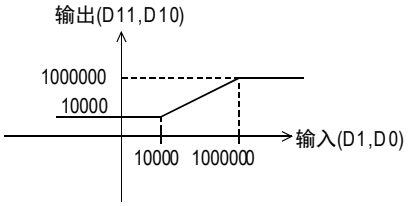
2. 程序举例2

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。



29.2 FNC 257 - BAND/死区控制

A8 A8C

概要

通过判断输入值是否在指定的死区的上下限范围内，从而来控制输出值的指令。

1. 指令格式

D	FNC 257 BAND	P

BAND

16位指令	指令记号	执行条件
9步	BAND	连续执行型
	BANDP	脉冲执行型

32位指令	指令记号	执行条件
17步	DBAND	连续执行型
	DBANDP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	死区(无输出区域)的下限值	BIN16/32位
(S2・)	死区(无输出区域)的上限值	
(S3・)	要通过死区控制的输入值	
(D・)	保存经过死区控制的输出值的软元件编号	

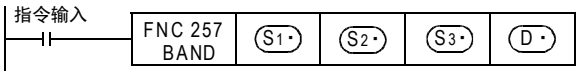
3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2・)								●	●	●	●	●	●	●	●	●			●	●	●			
(S3・)								●	●	●	●	●	●	●	●	●			●					
(D・)									●	●	●	●	●	●	●	●			●					

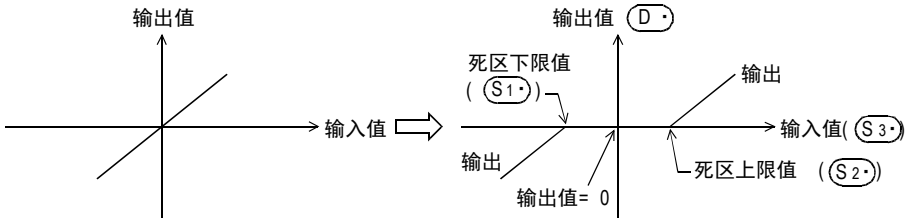
功能和动作说明

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・)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。

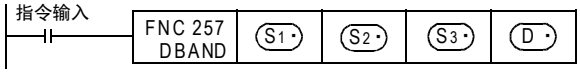


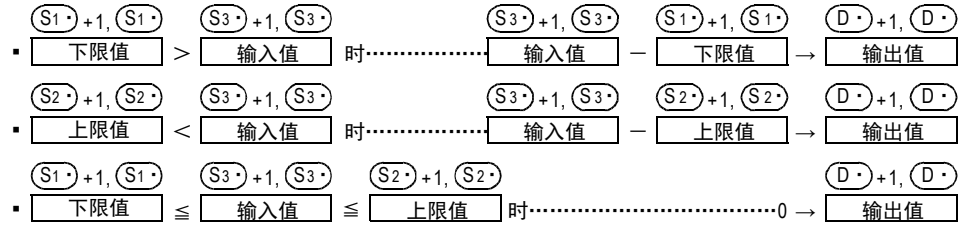
- (S1・) 下限值 ＞ (S3・) 输入值 时……(S3・) 输入值 － (S1・) 下限值 → (D・) 输出值
- (S2・) 上限值 ＜ (S3・) 输入值 时……(S3・) 输入值 － (S2・) 上限值 → (D・) 输出值
- (S1・) 下限值 ≦ (S3・) 输入值 ≦ (S2・) 上限值 时……0 → (D・) 输出值

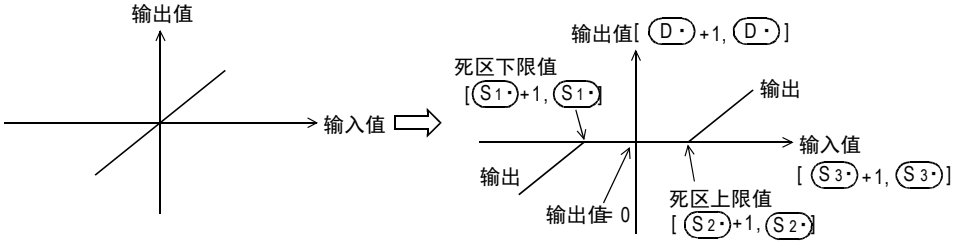


2. 32位运算(DBAND/DBANDP)

通过判断[(S3・)+1、(S3・)]指定的输入值(BIN32位值)是否在[(S1・)+1、(S1・)]、[(S2・)+1、(S2・)]指定的死区范围内，以此来控制保存在[(D・)+1、(D・)]指定的软元件中的输出值。
输出值如下所示被控制。







注意要点

- 输出值溢出时，如下所示。

－ 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时如下所示。

死区下限值 (S1・) =10
输入值 (S3・) = -32, 768

输出值= -32, 768 - 10
= 8000H - AH
= 7FF6H
= 32, 758

－ 32位运算时

输出值为带符号的32位BIN值。因此，运算结果超出-2147483648～2147483647时如下所示。

死区下限值 [(S1・) +1, (S1・)]=1000
输入值 [(S3・) +1, (S3・)]=-2,147,483,648

输出值 =-2,147,483,648 - 1000
= 80000000H - 000003E8H
= 7FFFFC18H
= 2,147,482,648

错误

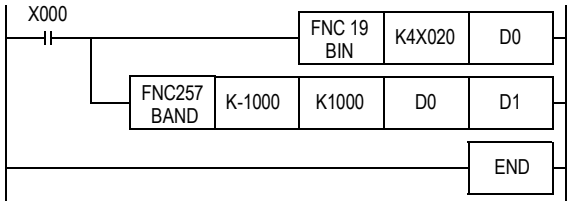
在下述设定状态下执行指令后，会出现运算错误，错误标志位(M8067)为ON，错误代码(K6706)保存在D8067中。

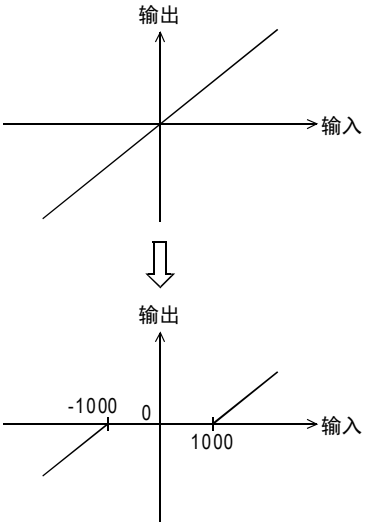
16位运算	(S1・) > (S2・)
32位运算	[(S1・)+1, (S1・)]>[(S2・)+1, (S2・)]

程序举例

1. 程序举例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。



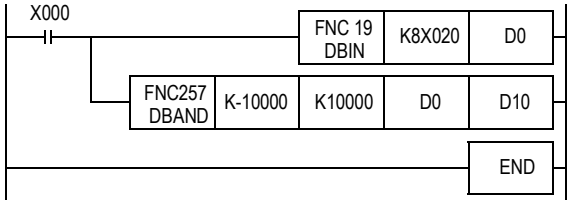


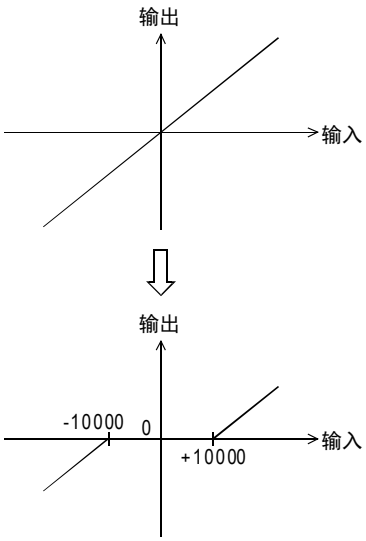
动作

- D0＜(-1000)时，D1中保存(D0)-(-1000)的值。
- -1000≦D0≦1000时，D1中保存0。
- 1000＜D0时，D1中保存(D0)-1000的值。

2. 程序举例2

当X000为ON时，对X020～X057中以BCD值设定的数据执行-10000～10000的死区控制，并保存到D10、D11中的程序。





动作

- (D1、D0)＜(-10000)时，(D11、D10)中保存(D1、D0)-(-10000)的值。
- -10000≦(D1、D0)≦10000时，(D11、D10)中保存0。
- 10000＜(D1、D0)时，(D11、D10)中保存(D1、D0)-10000的值。

29.3 FNC 258 - ZONE/区域控制

A8 A8C

概要

根据输入值是正数还是负数，用指定的偏差值来控制输出值的指令。

1. 指令格式

FNC 258 ZONE (D, P)

ZONE

16位指令	指令记号	执行条件
9步	ZONE	连续执行型
	ZONEP	脉冲执行型

32位指令	指令记号	执行条件
17步	DZONE	连续执行型
	DZONEP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1·)	加在输入值上的负偏差值	BIN16/32位
(S2·)	加在输入值上的正偏差值	
(S3·)	要通过区域控制的输入值	
(D·)	保存已通过区域控制的输出值的软元件起始编号	

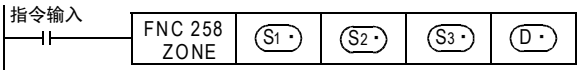
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2·)								●	●	●	●	●	●	●	●	●			●	●	●			
(S3·)								●	●	●	●	●	●	●	●	●			●					
(D·)									●	●	●	●	●	●	●	●			●					

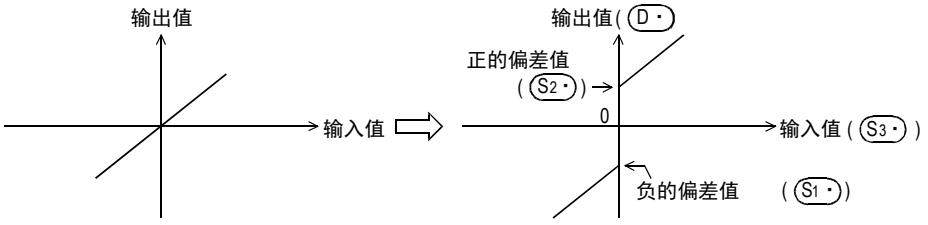
功能和动作说明

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。
偏差值的附加，如下所示执行。



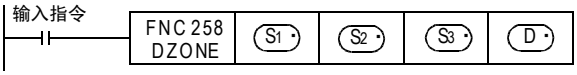
- (S3·) 输入值 ＜ 0 时 ····· (S3·) 输入值 ＋ (S1·) 负的偏差值 → (D·) 输出值
- (S3·) 输入值 ＝ 0 时 ····· 0 → (D·) 输出值
- (S3·) 输入值 ＞ 0 时 ····· (S3·) 输入值 ＋ (S2·) 正的偏差值 → (D·) 输出值

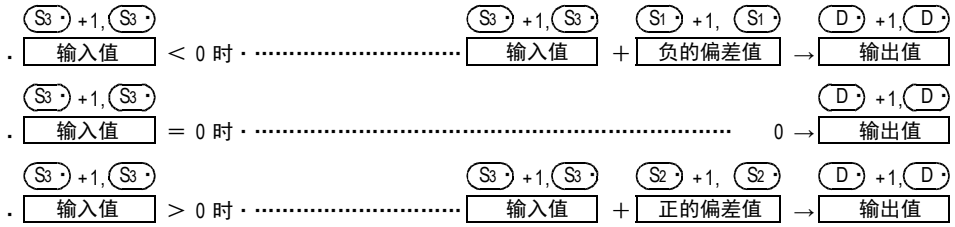


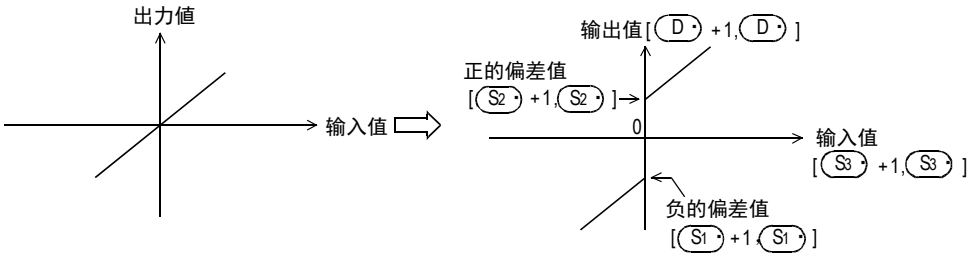
2. 32位运算(DZONE/DZONEP)

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。

偏差值的附加，如下所示执行。







注意要点

- 输出值溢出时，如下所示。

－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时如下所示。

负的偏差值 (S1・) ＝ -100
输入值 (S3・) ＝ -32, 768

⇨ 输出值 ＝ -32, 768+(-100)
＝ 8000H+FF9CH
＝ 7F9CH
＝ 32, 668

－32位运算时

输出值为带符号的32位BIN值。因此，运算结果超出-2147483648～2147483647时如下所示。

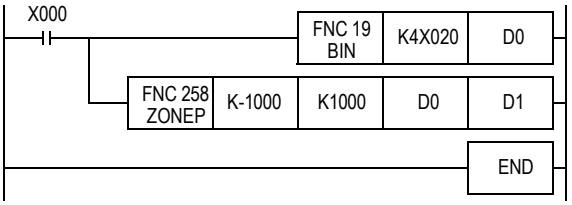
负的偏差值 [(S1・)+1, (S1・)]=-1000
输入值[(S3・)+1, (S3・)]=-2,147,483,648

⇨ 输出值 ＝ -2,147,483,648+(-1000)
＝ 80000000H+FFFFFC18H
＝ 7FFFFC18
＝ 2,147,482,648

程序举例

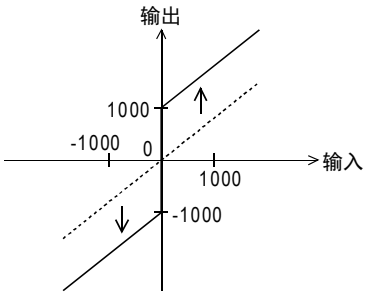
1. 程序举例1

当X000为ON时，对X020～X37中以BCD值设定的数据执行-1000～1000的区域控制，并保存到D1中的程序。



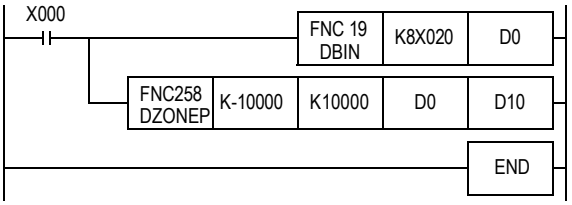
动作

- 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。



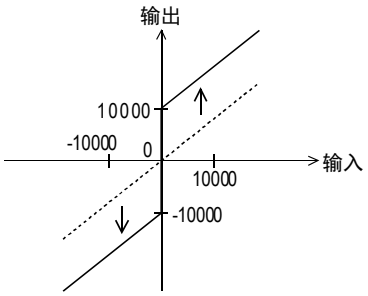
2. 程序举例2

当X000为ON时，对X020～X057中以BCD值设定的数据执行-10000～10000的区域控制，并保存到D0、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。



29.4 FNC 259 - SCL/定坐标(不同点坐标数据)

A8 A8C

概要

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。

→ 关于SCL(FNC 269)指令，请参考29.7节

1. 指令格式

D	FNC 259 SCL	P

SCALING

16位指令	指令记号	执行条件
7步	SCL	连续执行型
	SCLP	脉冲执行型

32位指令	指令记号	执行条件
13步	DSCL	连续执行型
	DSCLP	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	执行定坐标的输入值或是保存输入值的软元件编号	BIN16/32位
(S2・)	定坐标用的转换表格软元件的起始编号	
(D・)	保存被定坐标控制的输出值的软元件编号	

3. 对象软元件

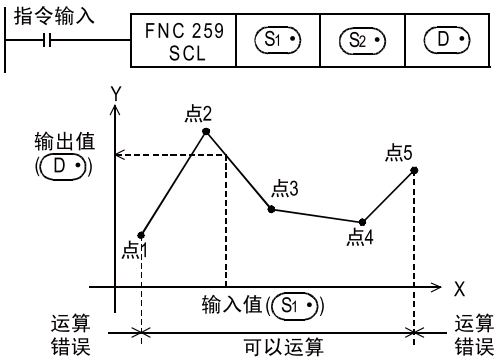
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2・)														●	●				●					
(D・)									●	●	●	●	●	●	●	●			●					

功能和动作说明

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，输出数据不是整数值时，小数第1位四舍五入后输出。

→ 关于定坐标用转换表格的设定方法，请参考下一页

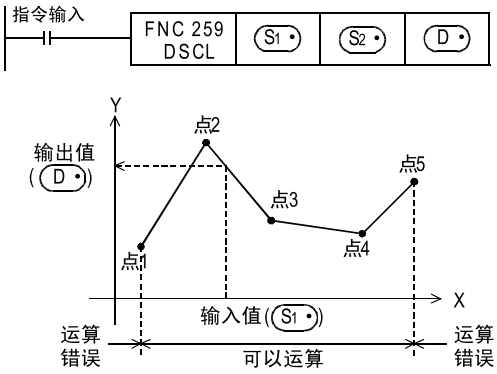


定坐标用转换设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数 (当为左图时变成“5”)		(S2)
点1	x坐标	(S2)+1
	y坐标	(S2)+2
点2	x坐标	(S2)+3
	y坐标	(S2)+4
点3	x坐标	(S2)+5
	y坐标	(S2)+6
点4	x坐标	(S2)+7
	y坐标	(S2)+8
点5	x坐标	(S2)+9
	y坐标	(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，输出数据不是整数值时，小数第1位四舍五入后输出。



定坐标用转换设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数 (当为左图时变成"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	乳(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

定坐标用的转换表格，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。

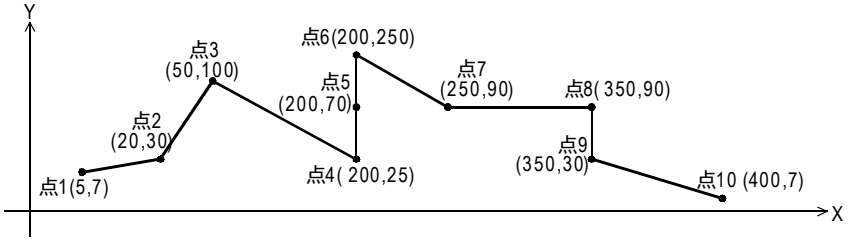
→ 关于设定的例子，请参考下一页

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例举了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		设定软元件及设定内容			备注
		(S2)中指定了R0时		设定内容	
坐标点数		(S2)	R0	K10	
点1	x坐标	(S2)+1	R1	K5	
	y坐标	(S2)+2	R2	K7	
点2	x坐标	(S2)+3	R3	K20	
	y坐标	(S2)+4	R4	K30	
点3	x坐标	(S2)+5	R5	K50	
	y坐标	(S2)+6	R6	K100	
点4	x坐标	(S2)+7	R7	K200	如果象这样指定3点的坐标，则输出值为中间值。 这个例子中，将点5的y坐标指定为输出值(中间值)。 此外，3点以上的X坐标相同时，也输出第2点的数值。
	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	如果象这样指定2点的坐标，则输出值取后一个点的y坐标值。 这个例子中，将点9的y坐标指定为输出值。
	y坐标	(S2)+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	

错误

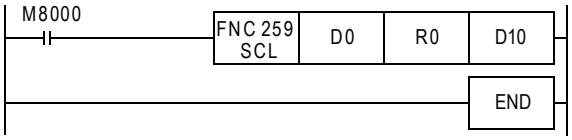
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 数据表格的Xn数据没有按照升序排列时。(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。
- (S1・)在数据表格设定的范围以外时。(错误代码K6706)
- 运算过程中的数值超出了32位数据的范围时。(错误代码 K6706)
  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。

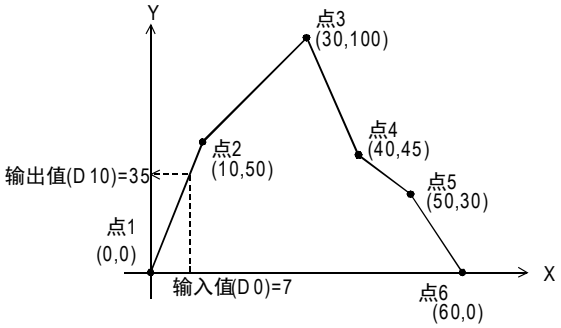
程序举例

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。

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

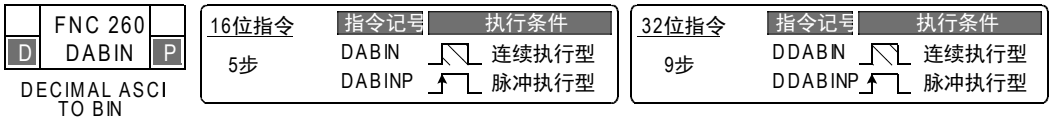
29.5 FNC 260 - DABIN/10进制ASCII→BIN的转换

A8 A8C

概要

将以10进制数字的ASCII码(30H～39H)形式现实的数据转换成BIN数据的指令。

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S・)	保存要转换成BIN值的数据(ASCII码)的软元件起始编号	字符串
(D・)	保存转换结果的软元件编号	BIN16/32位

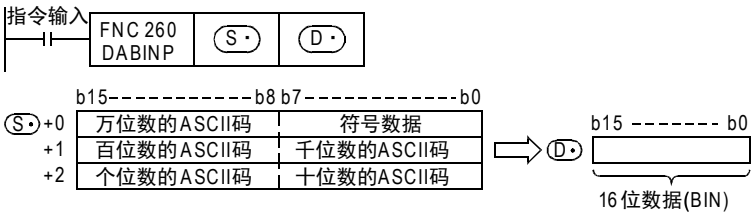
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●	●				●					
(D・)									●	●	●	●	●	●	●	●	●	●	●					

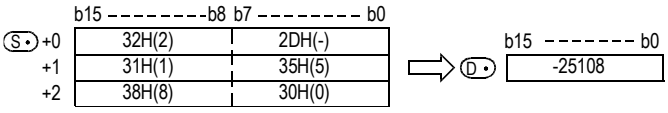
功能和动作说明

1. 16位运算(DABIN/DABINP)

1) 将以10进制数字的ASCII码(30H～39H)显示的(S・)～(S・)+2的数据转换成16位数据(BIN)后，保存到(D・)中。



例如，(S・)～(S・)+2为-25108的ASCII码时，在(D・)中如下所示地保存16位数据(BIN)。



2) (S・)～(S・)+2的数值范围为-32768～32767。

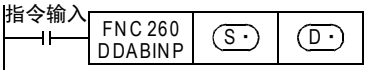
3) “符号数据”(S・)的低位字节)中，要转换的数据为正时设定“20H(空格)”为负时设定“2DH(-)”。

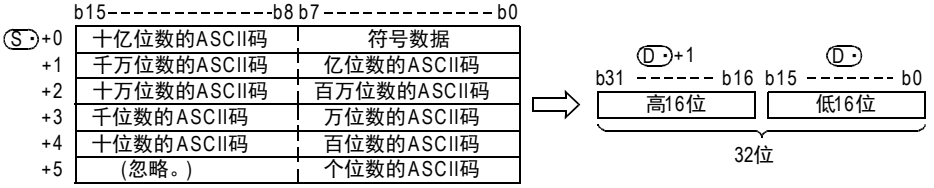
4) 各位数的ASCII码为“30H”～“39H”的范围。

5) 各位数的ASCII码为“20H(空格)”、“00H(NULL)”时，作为“30H”处理。

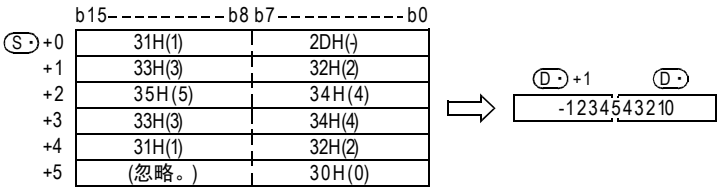
2. 32位运算(DDABIN/DDABINP)

1) 将以10进制数字的ASCII码(30H～39H)形式显示的 (S·) ～ (S·) +5的数据转换成32位数据(BIN)后，保存到[(D·) +1、 (D·)]中。





例如， (S·) ～ (S·) +5为-1,234,543,210的ASCII码时，在[(D·) +1、 (D·)]中如下所示地保存32位数据(BIN)。



2) (S·) ～ (S·) +5的数值范围为-2,147、483,648～2、147、483、647。
而且，忽略 (S·) +5的高字节。

3) “符号数据” (S·) 的低位字节)中，要转换的数据为正时设定“20H(空格)”为负时设定“2DH(-)”。

4) 各位数的ASCII码为“30H”～“39H”的范围。

5) 各位数的ASCII码为“20H(空格)”、“00H(NULL)”时，作为“30H”处理。

相关指令

指令	内容
ASCI(FNC 82)	将HEX代码转换成ASCII码的指令。
HEX(FNC 83)	将ASCII码转换成HEX代码的指令。
STR(FNC 200)	将BIN数据转换成字符串(ASCII码)的指令。
VAL(FNC 201)	将字符串(ASCII码)数据转换成BIN数据的指令。
BINDA(FNC 261)	将BIN数据转换成10进制的ASCII码(30H～39H)的指令。

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

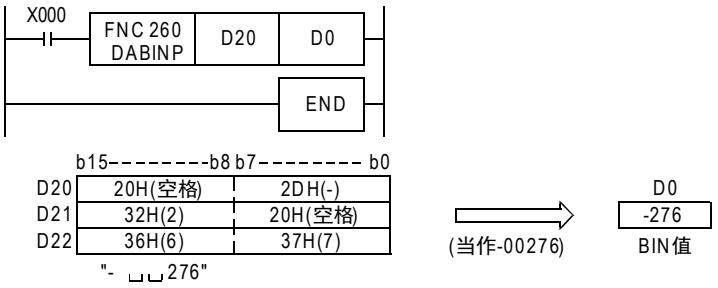
- 符号数据((S·) 的低位字节)为“20H(空格)”,"2DH(-)"以外的值时。
-(错误代码: K6706)
- (S·) ～ (S·) +2(5) 的各位数的 ASCII 码为“30H”～“39H”、“20H(空格)”“00H(NULL)”以外的值时(错误代码: K6706)
- (S·) ～ (S·) +2(5)的数值范围为下述的范围以外时。(错误代码: K6706)

	设定范围
16位运算	-32768～32767
32位运算	-2,147,483,648～2,147,483,647

- (S·) ～ (S·) +2(5)超出软元件范围时。(错误代码: K6706)

程序举例

当X000为ON时，将D20～D22中设定的符号以及5位10进制数字的ASCII码数据转换成BIN值后，保存到DO中的程序。



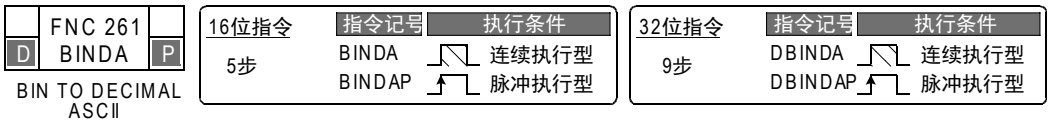
29.6 FNC 261 - BINDA/BIN→10进制ASCII的转换

A8 A8C

概要

将BIN数据转换成ASCII码(30H～39H)的指令

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	保存要转换成ASCII码的BIN值的软元件编号	BIN16/32位
(D·)	保存转换结果的软元件起始编号	字符串

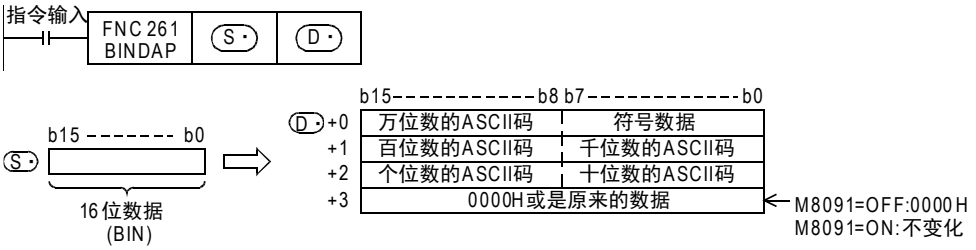
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)								●	●	●	●	●	●	●	●	●	●	●	●	●	●			
(D·)												●	●	●	●				●					

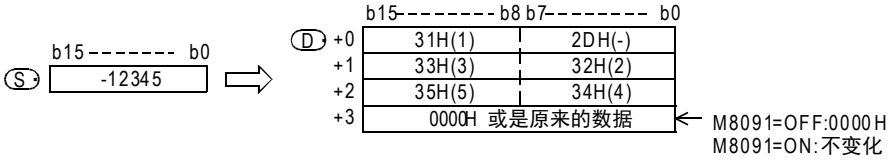
功能和动作说明

1. 16位运算(BINDA/BINDAP)

1) 将(S·)的16位数据(BIN)按照10进制的各个位数转换成ASCII码(30H～39H)，然后保存到(D·)中。



例如，(S·)为-12345时，在(D·)以后如下所示的进行保存。

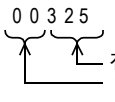


2) 16位数据(BIN)(S·)的数值范围为-32768～32767。

3) 运算结果(D·)如下所示。

a) “符号数据”((D·)的低位字节)中，16 位数据 (BIN)(S·)为正时，保存“20H(空格)”；为负时保存“2DH(-)”。

b) 有效位数的左侧的"0"中保存 “20H(空格)” (补0。)

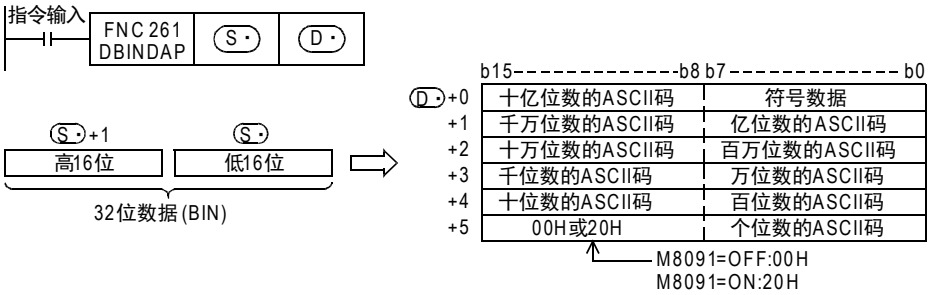


c) 根据M8091的ON/OFF状态，(D・)+3如下所示。

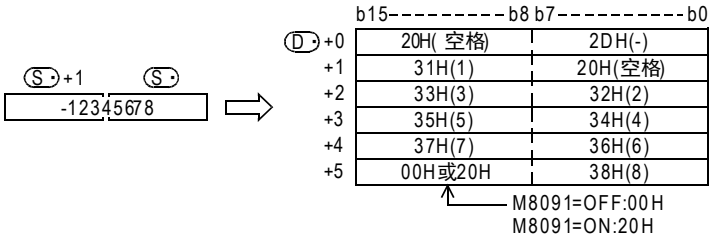
ON/OFF状态	处理内容
M8091=OFF	(D・)+3为0000H(NULL)。
M8091=ON	(D・)+3不变化。

2. 32位运算(DBINDA/DBINDAP)

1) 将[(S・)+1,(S・)]的32位数据(BIN)按照10进制的各个位数转换成ASCII码(30H～39H)，然后保存到(D・)开始的软元件中。



例如，[(S・)+1、(S・)]为-12,345,678时，(D・)开始的软元件中如下所示地进行保存。

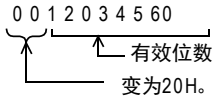


2) 32位数据(BIN)[(S・)+1、(S・)]的数值范围为-2,147,483,648～2,147,483,647。

3) (D・)中保存的运算结果，如下所示被保存。

a) 符号数据”(D・)的中，32位数据(BIN)[(S・)+1、(S・)]为正时保存“20H(空格)”；为负时保存“2DH(-)”。

b) 有效位数的左侧的“0”中保存“20H(空格)”(补0。)



c) 根据M8091的ON/OFF状态，(D・)+5的高字节如下所示。

ON/OFF状态	处理内容
M8091=OFF	(D・)+5的高字节为00H(NULL)。
M8091=ON	(D・)+5的高字节为20H(空格)。

相关软元件

软元件	名称	内容
M8091	输出字符数切换信号	• 16位运算时 — M8091=OFF时，(D・)+3为0000H(NULL)。 — M8091=ON时，(D・)+3不变化。 • 32位运算时 — M8091=OFF时，(D・)+5的高字节为00H(NULL)。 — M8091=ON时，(D・)+5的高字节为20H(空格)。

相关指令

指令	内容
ASCI(FNC 82)	将HEX代码转换成ASCII码的指令。
HEX(FNC 83)	将ASCII码转换成HEX代码的指令。
STR(FNC 200)	将BIN数据转换成字符串(ASCII码)的指令。
VAL(FNC 201)	将字符串(ASCII码)数据转换成BIN数据的指令。
DABIN(FNC 260)	将以10进制数字的ASCII码(30H～39H)形式显示的数值数据转换成BIN数据的指令。

注意要点

1. 软元件的占用点数

在16位运算(BINDA/BINDAP)时M8091的ON/OFF，以及32位运算(DBINDA/DBINDAP)中，(D·)的占用点数如下表所示。

		(D·)的占用点数
16位运算	M8091=ON	3
	M8091=OFF	4
32位运算		6

错误

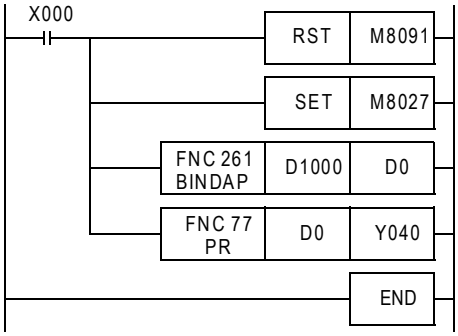
以下一些情况下会发生运算错误，错误标志位M8067置ON，D8067中保存错误代码。

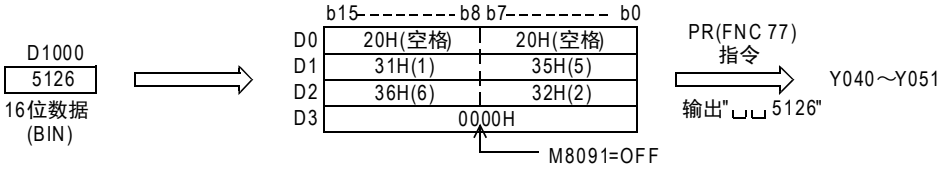
- ASCII码的字符串保存位置(D·)的占用点数，超出该软元件的范围时。
  (错误代码: K6706)

程序举例

当X000为ON后，将16位数据(BIN)D1000的值转换成10进制数字的ASCII码，然后使用PR(FNC 77)指令将已经转换的ASCII码逐个字符依次时分输出到Y040～Y051中的程序。
输出字符的切换信号M8091为OFF，通过将PR模式标志位M8027置ON，使ASCII码一直输出到00H为止。

→ 关于PR模式标志位PR(FNC 77)，请参考15.8节





29.7 FNC 269 - SCL2/定坐标2(X/Y坐标数据)

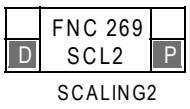
A8 A8C

概要

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(FNC 259)指令。

→ 关于SCL(FNC 259)指令(定坐标)，请参考29.4节

1. 指令格式



16位指令	指令记号	执行条件
7步	SCL2	连续执行型
	SCL2P	脉冲执行型

32位指令	指令记号	执行条件
13步	DSCL2	连续执行型
	DSCL2P	脉冲执行型

2. 设定数据

操作数种类	内容	数据类型
(S1・)	执行定坐标的输入值或是保存输入值的软元件编号	BIN16/32位
(S2・)	定坐标用的转换表格软元件的起始编号	
(D・)	保存被定坐标控制的输出值的软元件编号	

3. 对象软元件

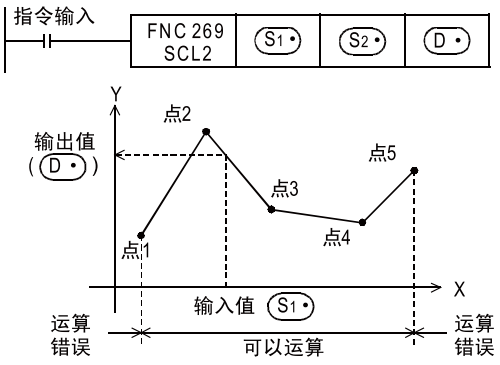
操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1・)								●	●	●	●	●	●	●	●	●			●	●	●			
(S2・)														●	●				●					
(D・)									●	●	●	●	●	●	●	●			●					

功能和动作说明

1. 16位运算(SCL2/SCL2P)

根据指定的转换特性，对(S1・)指定的输入值执行定坐标，然后保存到(D・)指定的软元件编号中。定坐标用的转换，是依据保存在(S2・)指定的软元件开始的数据表格执行的。
但是，输出数据不是整数值时，小数第1位四舍五入后输出。

→ 关于定坐标用转换表格的设定方法，请参考下一页

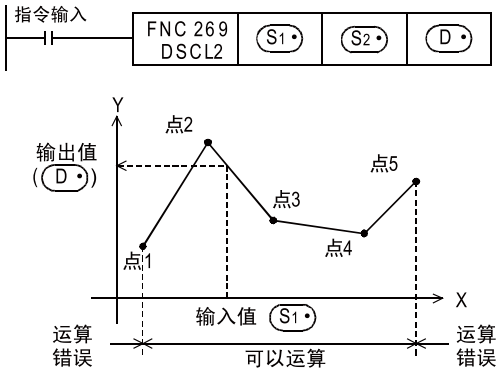


定坐标用转换设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数 (当为左图时变成“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(DSCL2/DSCL2P)

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，输出数据不是整数值时，小数第1位四舍五入后输出。



定坐标用转换设定数据表格

设定项目		设定数据表格的软元件分配
坐标点数 (当为左图时变成“5”)		[(S2) +1, (S2)]
x坐标	点1	[(S2) +3, (S2) +2]
	点2	[(S2) +5, (S2) +4]
	点3	[(S2) +7, (S2) +6]
	点4	[(S2) +9, (S2) +8]
	点5	[(S2) +11, (S2) +10]
y坐标	点1	[(S2) +13, (S2) +12]
	点2	[(S2) +15, (S2) +14]
	点3	[(S2) +17, (S2) +16]
	点4	[(S2) +19, (S2) +18]
	点5	[(S2) +21, (S2) +20]

3. 定坐标用转换表格的设定

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。数据表格的结构如下所示。

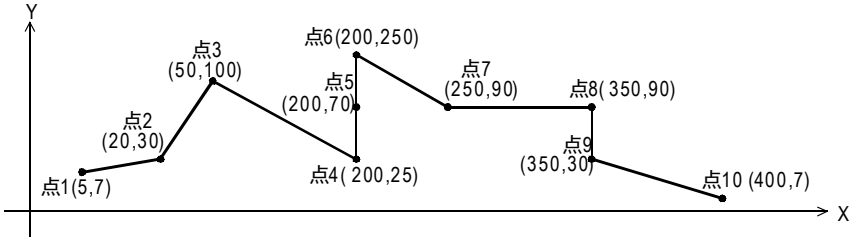
→ 关于设定的例子，请参考下一页

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)]
x坐标	点1	(S2) +1	[(S2) +3, (S2) +2]
	点2	(S2) +2	[(S2) +5, (S2) +4]
	≀	≀	≀
	点n(最后)	(S2) +n	[(S2) +2n+1、 (S2) +2n]
y坐标	点1	(S2) +n+1	[(S2) +2n+3、 (S2) +2n+2]
	点2	(S2) +n+2	[(S2) +2n+5、 (S2) +2n+4]
	≀	≀	≀
	点n(最后)	(S2) +2n	[(S2) +4n+1、 (S2) +4n]

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例举了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		设定软元件及设定内容			备注
		(S2・)中指定了R0时		设定内容	
坐标点数		(S2・)	R0	K10	
x坐标	点1	(S2・)+1	R1	K5	
	点2	(S2・)+2	R2	K20	
	点3	(S2・)+3	R3	K50	
	点4	(S2・)+4	R4	K200	参考*1
	点5	(S2・)+5	R5	K200	
	点6	(S2・)+6	R6	K200	
	点7	(S2・)+7	R7	K250	
	点8	(S2・)+8	R8	K350	参考*2
	点9	(S2・)+9	R9	K350	
	点10	(S2・)+10	R10	K400	
y坐标	点1	(S2・)+11	R11	K7	
	点2	(S2・)+12	R12	K30	
	点3	(S2・)+13	R13	K100	
	点4	(S2・)+14	R14	K25	参考*1
	点5	(S2・)+15	R15	K70	
	点6	(S2・)+16	R16	K250	
	点7	(S2・)+17	R17	K90	
	点8	(S2・)+18	R18	K90	参考*2
	点9	(S2・)+19	R19	K30	
	点10	(S2・)+20	R20	K7	

*1. 如果像点4、5、6那样指定3点的坐标时则输出值为中间值。
在这个例子中，将点5的y坐标指定为输出值(中间值)。
此外，即使3点以上的X坐标相同时，输出第2点的数值。

*2. 如果像点8、9那样指定2点的坐标，则输出值取后一个点的y坐标的值。
在这个例子中，将点9的y坐标指定为输出值。

错误

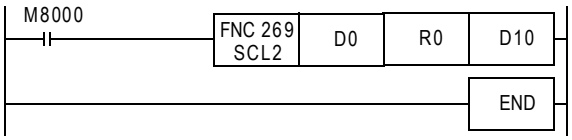
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 数据表格的Xn数据没有按照升序排列时。(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。
- (S1・)在数据表格设定的范围以外时。(错误代码K6706)
- 运算过程中的数值超出了32位数据的范围时。(错误代码K6706)
  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。

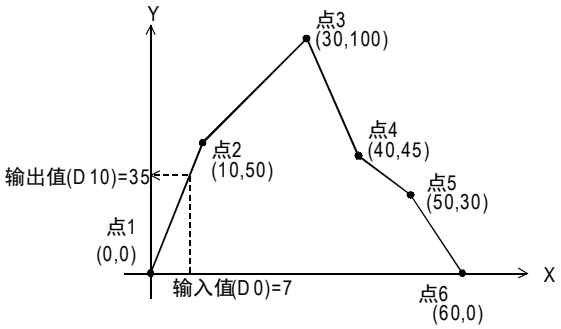
程序举例

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。

程序



动作

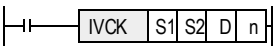
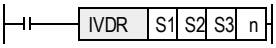
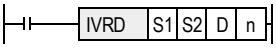
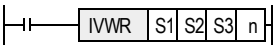
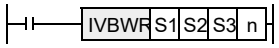


定坐标用转换设定数据表格

设定项目		软元件	设定内容
坐标点数		R0	K6
x坐标	点1	R1	K0
	点2	R2	K10
	点3	R3	K30
	点4	R4	K40
	点5	R5	K50
	点6	R6	K60
y坐标	点1	R7	K0
	点2	R8	K50
	点3	R9	K100
	点4	R10	K45
	点5	R11	K30
	点6	R12	K0

30. 外部设备通信（变频器通信）—FNC 270～FNC 274

连接多台FREQROL系列变频器，执行运行控制，以及参数的读出/写入用的指令。

FNC No.	指令记号	符号	功能	参考
270	IVCK		变换器的运转监视	30.1节
271	IVDR		变频器的运行控制	30.2节
272	IVRD		变频器的参数读取	30.3节
273	IVWR		变频器的参数写入	30.4节
274	IVBWR		变频器的参数成批写入	30.5节

30.1 FNC 270 - IVCK/变频器的运行监视

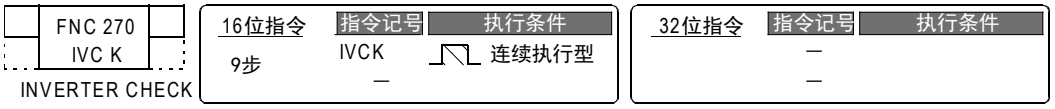
A8 A8C

概要

使用变频器一侧的计算机链接运行功能，在可编程控制器中读出变频器运行状态。根据版本不同，该指令适用的变频器也不同。

→ 关于指令的详细说明，请参考通信控制手册

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1·)	变频器的站号(K0～K31)	BIN16位
(S2·)	变频器的指令代码(下一页)	
(D·)	保存读出值的软元件编号	
n	使用的通道(K1:通道1，K2:通道2)	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●			
(S2·)														●	●	▲			●	●	●			
(D·)									●	●	●			●	●	▲			●					
n																				●	●			

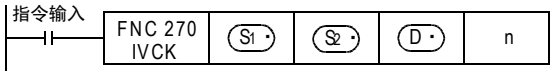
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

→ 关于指令的详细说明，请参考通信控制手册

1. 16位运算(IVCK)

针对通信口n上连接的变频器的站号(S1·)，根据(S2·)的[指令代码][*1]，将相应的变频器运行状态读出到(D·)中。



*1. 请参考下一页的指令代码一览。
此外，还请参考变频器使用手册中关于计算机链接的详细说明。

2. 变频器的指令代码

(S2•)中指定的变频器的指令代码以及其功能如下表所示。
有关指令代码，请参考变频器使用手册中关于计算机链接的详细说明。

(S2•)中指定的变频器的指令代码	读出的内容	适用的变频器								
		F700	A700	E700	D700	V500	F500	A500	E500	S500
H7B	运行模式	○	○	○	○	○	○	○	○	○
H6F	输出频率[速度]	○	○	○	○	○	○	○	○	○
H70	输出电流	○	○	○	○	○	○	○	○	○
H71	输出电压	○	○	○	○	○	○	○	○	—
H72	特殊监控	○	○	○	○	○	○	○	—	—
H73	特殊监控选择号	○	○	○	○	○	○	○	—	—
H74	故障内容	○	○	○	○	○	○	○	○	○
H75	故障内容	○	○	○	○	○	○	○	○	○
H76	故障内容	○	○	○	○	○	○	○	○	—
H77	故障内容	○	○	○	○	○	○	○	○	—
H79	变频器状态监控(扩展)	○	○	○	○	—	—	—	—	—
H7A	变频器状态监控	○	○	○	○	○	○	○	○	○
H6E	读取设定频率(E2PROM)	○	○	○	○	○	○	○	○	○
H6D	读取设定频率(RAM)	○	○	○	○	○	○	○	○	○

3. 相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

编号		内容
通道1	通道2	
M8029		指令执行结束
M8063	M8438	串行通信错误
M8151	M8156	变频器通信中*1
M8152	M8157	变频器通信错误*1
M8153	M8158	变频器通信错误锁定*1
M8154	M8159	IVBWR指令错误*1

编号		内容
通道1	通道2	
D8063	D8438	串行通信错误的错误代码
D8150	D8155	变频器通信响应等待时间
D8151	D8156	变频器通信中的步编号*2
D8152	D8157	变频器通信错误代码*1
D8153	D8158	变频器通信错误步的锁定*2
D8154	D8159	IVBWR指令错误的参数编号*2

*1. 从STOP→RUN时清除

*2. 初始值: -1

注意要点

→ 关于其他的注意事项，请参考通信控制手册

- 不能对同一个端口使用RS(FNC 80)、RS2(FNC 87)指令和变频器通信指令(FNC 270～FNC 274)。
- 可以对同一个端口同时驱动多台变频器通信指令(FNC 270～FNC 274)。

30.2 FNC 271 - IVDR/变频器的运行控制

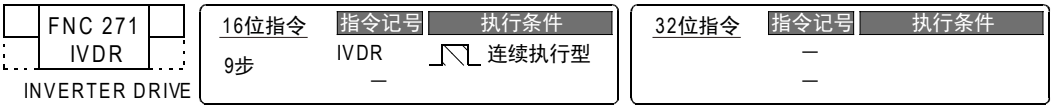
A8 A8C

概要

该指令dd是使用变频器一侧的计算机链接运行功能，通过可编程控制器中写入变频器运行所需的控制值。相

→ 关于指令的详细说明，请参考通信控制手册

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1·)	变频器的站号(K0～K31)	BIN16位
(S2·)	变频器的指令代码(下一页)	
(S3·)	写入到变频器的参数中的设定值，或是保存设定数据的软元件编号	
n	使用的通道(K1:通道1，K2: 通道2)	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●			
(S2·)														●	●	▲			●	●	●			
(S3·)								●	●	●	●			●	●	▲			●	●	●			
n																				●	●			

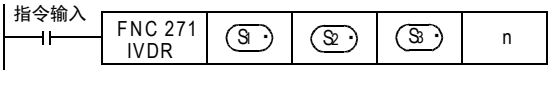
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

→ 关于指令的详细说明，请参考通信控制手册

1. 16位运算(IVDR)

针对连接在通信口n上的站号 (S1·) 的变频器，根据 (S2·) 的[指令代码]*1写入 (S3·) 的控制值。



*1. 请参考下一页中的指令代码一览。
此外，还请参考变频器使用手册中关于计算机链接的详细说明。

2. 变频器的指令代码

(S2•) 中指定的变频器的指令代码以及其功能如下表所示。
有关指令代码，请参考变频器使用手册中关于计算机链接的详细说明。

(S2•) 中指定的变频器的指令代码16进制数	写入的内容	适用的变频器								
		F700	A700	E700	D700	V500	F500	A500	E500	S500
HFB	运行模式	○	○	○	○	○	○	○	○	○
HF3	特殊监示的选择号	○	○	○	○	○	○	○	—	—
HF9	运行指令(扩展)	○	○	○	○	—	—	—	—	—
HFA	运行指令	○	○	○	○	○	○	○	○	○
HEE	写入设定频率(EEPROM)	○	○	○	○	○	○	○	○	○
HED	写入设定频率(RAM)	○	○	○	○	○	○	○	○	○
HFD	变频器复位	○	○	○	○	○	○	○	○	○
HF4	故障内容的成批清除	○	○	○	○	—	○	○	○	○
HFC	参数的全部清除	○	○	○	○	○	○	○	○	○
HFC	用户清除	○	○	○	○	—	○	○	—	—

3. 相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

编号		内容
通道1	通道2	
M8029		指令执行结束
M8063	M8438	串行通信错误
M8151	M8156	变频器通信中*1
M8152	M8157	变频器通信错误*1
M8153	M8158	变频器通信错误锁定*1
M8154	M8159	IVBWR指令错误*1

编号		内容
通道1	通道2	
D8063	D8438	串行通信错误的错误代码
D8150	D8155	变频器通信响应等待时间
D8151	D8156	变频器通信中的步编号*2
D8152	D8157	变频器通信错误代码*1
D8153	D8158	变频器通信错误步的锁定*2
D8154	D8159	IVBWR指令错误的参数编号*2

*1. 从STOP→RUN时清除

*2. 初始值: -1

注意要点

→ 关于其他的注意事项，请参考通信控制手册

- 不能对同一个端口使用RS(FNC 80)、RS2(FNC 87)指令和变频器通信指令(FNC 270～FNC 274)。
- 可以对同一个端口同时驱动多台变频器通信指令(FNC 270～FNC 274)。

30.3 FNC 272 - IVRD/读取变频器的参数

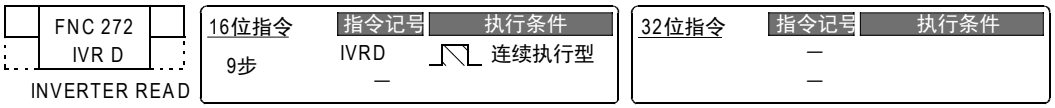
A8 A8C

概要

使用变频器一侧的计算机链接运行功能，在可编程控制器中读出变频器参数的指令。相当于A2/A2C系列的EXTR指令(K12)。

→ 关于指令的详细说明，请参考通信控制手册

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S1·)	变频器的站号(K0～K31)	BIN16位
(S2·)	变频器的参数编号	
(D·)	保存读出值的软元件编号	
n	使用的通道(K1: 通道1，K2: 通道2)	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●			
(S2·)														●	●	▲			●	●	●			
(D·)														●	●	▲			●					
n																				●	●			

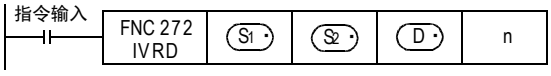
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

→ 关于指令的详细说明，请参考通信控制手册

1. 16位运算(IVRD)

从通信口n上连接的站号(S1·)的变频器中将参数(S2·)的值读出到(D·)中。



2. 相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

编号		内容
通道1	通道2	
M8029		指令执行结束
M8063	M8438	串行通信错误
M8151	M8156	变频器通信中*1
M8152	M8157	变频器通信错误*1
M8153	M8158	变频器通信错误锁定*1
M8154	M8159	IVBWR指令错误*1

编号		内容
通道1	通道2	
D8063	D8438	串行通信错误的错误代码
D8150	D8155	变频器通信响应等待时间
D8151	D8156	变频器通信中的步编号*2
D8152	D8157	变频器通信错误代码*1
D8153	D8158	变频器通信错误步的锁定*2
D8154	D8159	IVBWR指令错误的参数编号*2

*1. 从STOP→RUN时清除

*2. 初始值: -1

注意要点

→ 关于其他的注意事项，请参考通信控制手册

- 不能对同一个端口使用RS(FNC 80)、RS2(FNC 87)指令和变频器通信指令(FNC 270～FNC 274)。
- 可以对同一个端口同时驱动多台变频器通信指令(FNC 270～FNC 274)。

30.4 FNC 273 - IVWR/写入变频器的参数

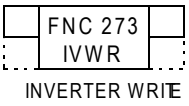
A8 A8C

概要

使用变频器一侧的计算机链接运行功能，写入变频器的参数的指令。相当于A2/A2C系列的EXTR指令(K13)。

→ 关于指令的详细说明，请参考通信控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	IVWR	连续执行型
	—	

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1·)	变频器的站号(K0～K31)	BIN16位
(S2·)	变频器的参数编号	
(S3·)	向变频器参数中写入的设定值， 或是保存设定数据的软元件编号	
n	使用的通道(K1: 通道1，K2: 通道2)	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	▲			●	●	●			
(S2·)														●	●	▲			●	●	●			
(S3·)														●	●	▲			●	●	●			
n																				●	●			

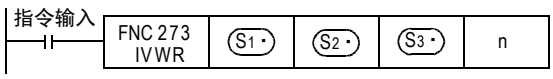
▲: 仅对应A8/A8C可编程控制器。

功能和动作说明

→ 关于指令的详细说明，请参考通信控制手册

1. 16位运算(IVWR)

向通信口n上连接的站号 (S1·) 的变频器的参数 (S2·) 中，写入 (S3·) 的值。



2. 相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

编号		内容
通道1	通道2	
M8029		指令执行结束
M8063	M8438	串行通信错误
M8151	M8156	变频器通信中*1
M8152	M8157	变频器通信错误*1
M8153	M8158	变频器通信错误锁定*1
M8154	M8159	IVBWR指令错误*1

编号		内容
通道1	通道2	
D8063	D8438	串行通信错误的错误代码
D8150	D8155	变频器通信响应等待时间
D8151	D8156	变频器通信中的步编号*2
D8152	D8157	变频器通信错误代码*1
D8153	D8158	变频器通信错误步的锁定*2
D8154	D8159	IVBWR指令错误的参数编号*2

*1. 从STOP→RUN时清除

*2. 初始值: -1

注意要点

→ 关于其他的注意事项，请参考通信控制手册

- 不能对同一个端口使用RS(FNC 80)、RS2(FNC 87)指令和变频器通信指令(FNC 270～FNC 274)。
- 可以对同一个端口同时驱动多台变频器通信指令(FNC 270～FNC 274)。
- 在D700系列使用密码功能时的注意事项

1) 发生通信错误时
变频器通信指令发生通信错误时，HC可编程控制器以3次为限*1自动重试。
因此，对于启用Pr297的“密码解除错误的次数显示”*2的D700系列，当发生密码解除错误时，如下所示，Pr297的密码解除错误次数可能和实际密码错误输入的次数不一致。
此外，对Pr297进行写入时，请不要通过顺控程序执行自动重试(变频器指令的再驱动)。

变频器通信指令发生密码解除错误的情况，以及此时的实际解除错误次数：
- 由于密码输入错误等原因， 将错误的密码写入Pr297时
  执行了1次写入指令， 而密码的解除错误次数变成3次。
- 由于噪音等原因， 未能向Pr297正确写入密码时
  密码的解除错误次数最多为3次。

2) 登录密码时
变频器通信指令中，向D700系列变频器登录密码时，将密码写入Pr297后，请重新读取Pr297，确认密码的登录是否正常结束*3。由于噪音等原因，未能正常向Pr297完成写入时，HC可编程控制器可能自动重试，并因此将登录的密码解除。

*1. 最多可以进行3次通信，包括初次通信和2次重试。

*2. 当D700系列启用Pr297的“密码解除错误的次数显示”时，密码解除错误次数到达5次后，即使输入正确密码，也不能解除读出/写入限制。
要从此状态下恢复，必须将D700系列的参数全部清除。

*3. 重新读取Pr297的值为0～4时，密码登录正常结束。

30.5 FNC 274 - IVBWR/成批写入变频器的参数

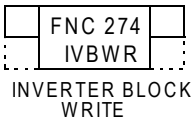
A8 A8C

概要

使用变频器一侧的计算机链接运行功能，成批写入变频器参数的指令。

→ 关于指令的详细说明，请参考通信控制手册

1. 指令格式



16位指令	指令记号	执行条件
9步	IVBWR	连续执行型
	—	

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1·)	变频器的站号(K0～K31)	BIN16位
(S2·)	变频器的参数写入个数	
(S3·)	写入到变频器中的参数表的起始软元件编号	
n	使用的通道(K1:通道1，K2：通道2)	

3. 对象软元件

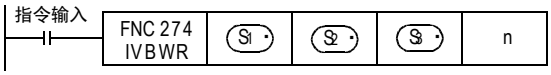
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●	●			●	●	●			
(S2·)														●	●	●			●	●	●			
(S3·)														●	●	●			●					
n																				●	●			

功能和动作说明

→ 关于指令的详细说明，请参考通信控制手册

1. 16位运算(IVBWR)

针对通信口n上连接的站号 (S1・) 的变频器，将 (S2・) (S3・) 指定的数据表格*1(参数编号和设定值)成批写入到变频器中。



*1. 数据表格的格式如下表所示。
(S2・): 写入的参数个数
(S3・): 数据表格的起始软元件编号

软元件	写入的参数编号及设定值	
(S3・)	第1个	参数编号
(S3・)+1		设定值
(S3・)+2	第2个	参数编号
(S3・)+3		设定值
≀	≀	≀
(S3・)+2(S2・)-4	第(S2・)-1个	参数编号
(S3・)+2(S2・)-3		设定值
(S3・)+2(S2・)-2	第(S2・)个	参数编号
(S3・)+2(S2・)-1		设定值

2. 相关软元件

→ 有关指令执行结束标志位的使用方法，请参考6.5.2项

编号		内容
通道1	通道2	
M8029		指令执行结束
M8063	M8438	串行通信错误
M8151	M8156	变频器通信中*2
M8152	M8157	变频器通信错误*2
M8153	M8158	变频器通信错误锁定*2
M8154	M8159	IVBWR指令错误*2

编号		内容
通道1	通道2	
D8063	D8438	串行通信错误的错误代码
D8150	D8155	变频器通信响应等待时间
D8151	D8156	变频器通信中的步编号*3
D8152	D8157	变频器通信错误代码*2
D8153	D8158	变频器通信错误步的锁定*3
D8154	D8159	IVBWR指令错误的参数编号*3

*2. 从STOP→RUN时清除

*3. 初始值: -1

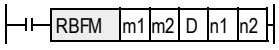
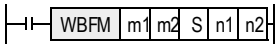
注意要点

→ 关于其他的注意事项，请参考通信控制手册

- 不能对同一个端口使用RS(FNC 80)、RS2(FNC 87)指令和变频器通信指令(FNC 270～FNC 274)。
- 可以对同一个端口同时驱动多台变频器通信指令(FNC 270～FNC 274)。

31. 数据传送3—FNC 275～FNC 279

FNC 275～FNC 279中，提供了用于特定处理的、执行比基本应用指令更加复杂处理的指令。

FNC No.	指令记号	符号	功能	参考
275	—			—
276	—			—
277	—			—
278	RBFM		BFM分割读出	31.1节
279	WBFM		BFM分割写入	31.2节

31.1 FNC 278 - RBFM/BFM分割读出

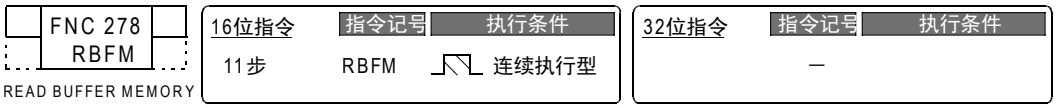
A8 A8C

概要

分几个运算周期，从特殊功能模块/单元中连续的缓冲存储区(BFM)读取数据的指令。可以将保存在通信用特殊功能模块/单元的BFM中的接收数据等分割后读出，因此非常方便。
此外，作为读取缓冲存储区(BFM)数据的指令还有FROM(FNC 78)指令。

→ 关于FROM(FNC 78)指令，请参考15.9节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
m1	单元号[0～7]	BIN16位数据
m2	缓冲存储区(BFM)的起始编号[0~32766]	
(D・)	保存从缓冲存储区(BFM)读出的数据的软元件起始编号	
n1	读出缓冲存储区(BFM)的总点数[1~32767]	
n2	每个运算周期的传送点数[1~32767]	

3. 对象软元件

操作数种类	位软元件							字软元件													其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
m1														●	●					●	●			
m2														●	●					●	●			
(D・)														▲	●				●					
n1														●	●					●	●			
n2														●	●					●	●			

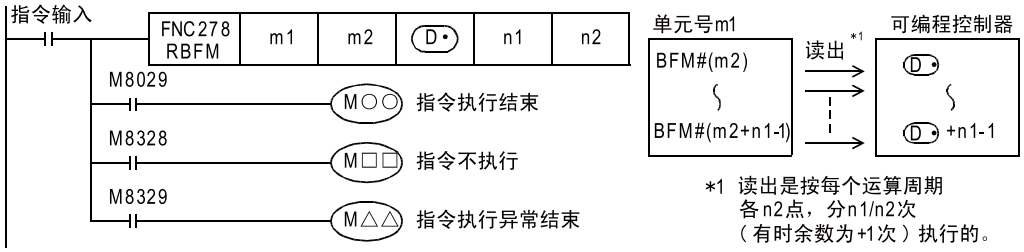
▲: 特殊数据寄存器D除外

功能和动作说明

1. 16位运算(RBFM)

从单元号m1的特殊功能模块/单元的缓冲存储区(BFM)m2号开始，将n1点按每次n2点，分n1/n2个运算周期(有余数时为+1个运算周期)，传送(读出)到可编程控制器的(D・)开始的软元件中。

→ 关于单元号、缓冲存储区(BFM)编号、注意要点、程序举例，请参考31.1.1项



- 如指令正常结束，则指令执行结束标志位M8029为ON，如指令异常结束，则指令执行异常结束标志位M8329为ON。
- 针对相同的单元号，正在执行其他步的RBFM(FNC 278)指令或WBFM(FNC 279)指令时，指令不执行的标志位M8328为ON，指令的执行处于待机状态。
  当其他的对象指令执行结束后，待机状态的指令才会解除待机状态，然后执行指令。

相关软元件

→ 关于指令执行结束、指令执行异常结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	当指令正常结束时为ON。
M8328	指令不执行	针对相同的单元号，正在执行其他步中的RBFM(FNC 278)指令或是WBFM(FNC 279)指令时为ON。
M8329	指令执行异常结束	当指令异常结束时为ON。

相关指令

指令	内容
FROM(FNC 78)	BFM的读出
TO(FNC 79)	BFM的写入
WBFM(FNC 279)	BFM分割写入

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 不存在单元号m1时。(错误代码: K6708)

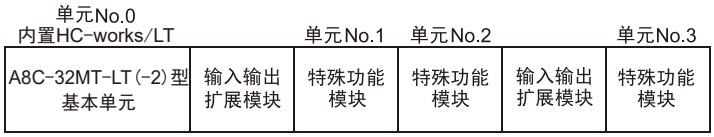
31.1.1 RBFM(FNC 278)/WBFM(FNC 279)命令的通用事项

特殊功能模块/单元的单元号和缓冲存储区的指定

→ 关于特殊功能模块/单元的连接方法、允许连接台数以及输入输出编号的使用等，请参考所使用的可编程控制器以及特殊功能模块/单元的手册

1. 特殊功能单元/模块的单元号m1

单元号是用于指定RBFM/WBFM指令对哪个设备动作。
设定范围: K0～K7



会对与可编程控制器连接的特殊功能单元/模块，自动分配单元号。
单元号是从离基本单元最近的模块开始依次为No.0→No.1→No.2…。
A8C-32MT-LT(-2)可编程控制器的情况下，由于内置HC-Link/LT主站，单元号是从离基本单元最近的模块开始依次为No.0→No.1→No.2…。

2. 缓冲存储区(BFM)编号m2

特殊功能单元/模块中，16位的RAM存储器最大内置了32767点。这个RAM存储器就称为缓冲存储区(BFM)。
缓冲存储区(BFM)的编号为#0～#32766，其内容由个特殊功能模块/单元决定。
设定范围: K0～K32766

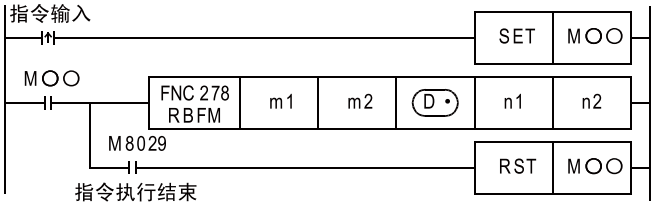
→ 关于缓冲存储区的内容，请参考所使用的特殊功能模块/单元的手册

注意要点

- 每个运算周期的传送点数较多时，可能会发生看门狗定时器错误。
  此时，请如下所示操作。
  — 更改看门狗定时器时间的方法
  通过改写D8000(看门狗定时器时间)的内容，可以更改看门狗定时器的检测时间。(初始值: K200)
  通过输入如下所示的程序，此后的顺控程序会按照新的看门狗定时器时间进行监视。



  — 更改每个运算周期的传送点数n2的方法
  请将每个运算周期的传送点数n2更改为较小的值。
- 指令执行过程中请勿中止指令的驱动。如果中止，则中断缓冲存储区(BFM)的读出/写入处理，但是在 (D・) 开始的软元件，或者缓冲存储区(BFM)中，已保存读出写入途中的数据。

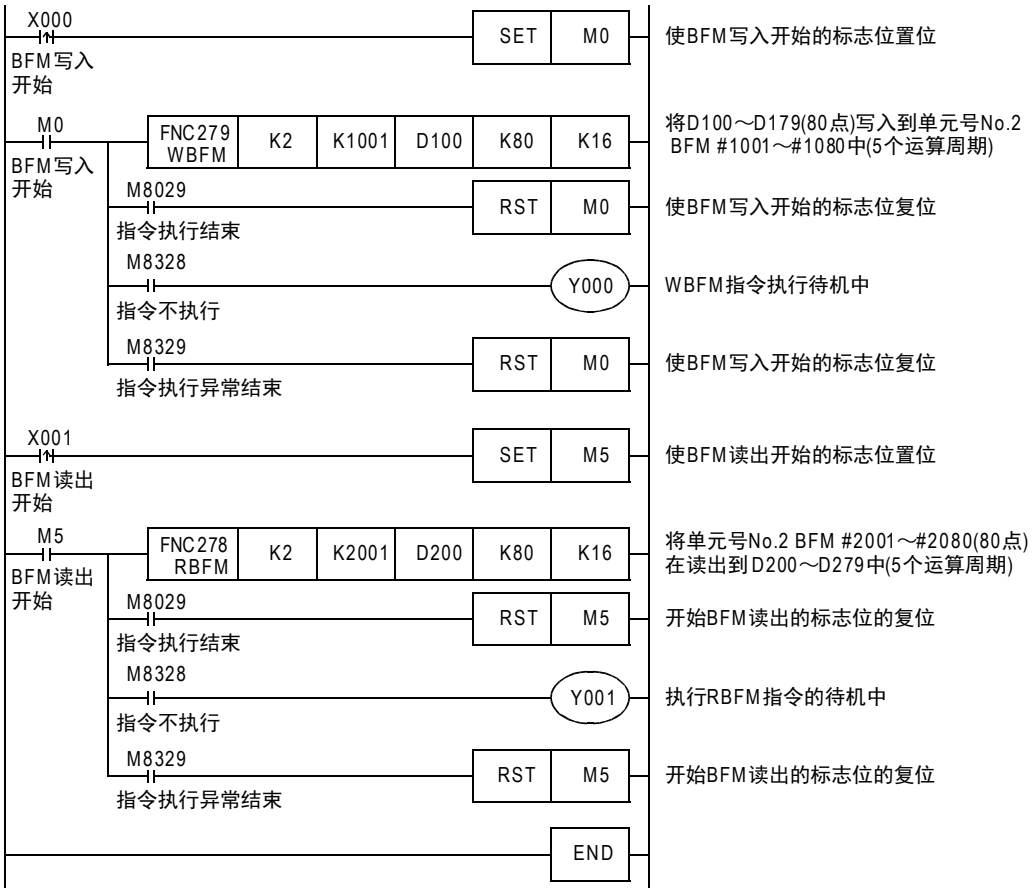


- 进行变址修饰时，使用指令开始执行时的变址寄存器的内容。
  指令执行以后，即使变址寄存器的内容改变，也不会反映到指令的处理中。
- RBFM(FNC 278)指令执行的过程中，由于 (D・) 开始的n1点的内容在变化，所以请在指令执行完成后，对 (D・) 开始的n1点执行其他的指令处理。
- WBFM(FNC 279)指令执行的过程中，请勿更新(改变) (S・) 开始的n1点的内容。如果更新，则期望的数据有可能会没有写入到缓冲存储区(BFM)中。
- RBFM(FNC 278)指令执行的过程中，请勿更新缓冲存储区(BFM)m2开始的n1点的内容。如果更新，则有可能不能读出期望的数据。

程序举例

对单元号No.2的缓冲存储区(BFM)，如下所示地读出/写入数据的程序。

- X000为ON时，按照每个运算周期16点，将D100～D179(80点)的数据写入到单元号No.2的特殊功能模块/单元的缓冲存储区(BFM)#1001～#1080中。
- X001 为 ON 时，按照每个运算周期 16 点，将单元号 No.2 的特殊功能模块 / 单元的缓冲存储区 (BFM)#2001 ～#2080(80点)中的内容，读出到D200～D279中。



31.2 FNC 279-WBFM/BFM分割写入

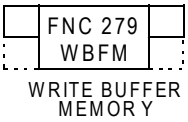
A8 A8C

概要

分几个运算周期，将数据写入到特殊功能模块/单元中连续的缓冲存储区(BFM)中的指令。由于可以将发送数据等分割后写入到通信用特殊功能模块/单元的BFM中，因此非常方便。
此外，作为写入数据到缓冲存储区(BFM)的指令还有TO(FNC 79)指令。

→ 关于TO(FNC 79)指令，请参考15.10节

1. 指令格式



16位指令	指令记号	执行条件
11步	WBFM	连续执行型

32位指令	指令记号	执行条件
	—	

2. 设定数据

操作数种类	内容	数据类型
m1	单元号[0～7]	BIN16位数据
m2	缓冲存储区(BFM)的起始编号[0~32766]	
(S・)	保存写入到缓冲存储区(BFM)的数据的软元件起始编号	
n1	写入缓冲存储区(BFM)的总点数[1～32767]	
n2	每个运算周期的传送点数[1~32767]	

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
m1														●	●					●	●			
m2														●	●					●	●			
(S・)														▲	●				●					
n1														●	●					●	●			
n2														●	●					●	●			

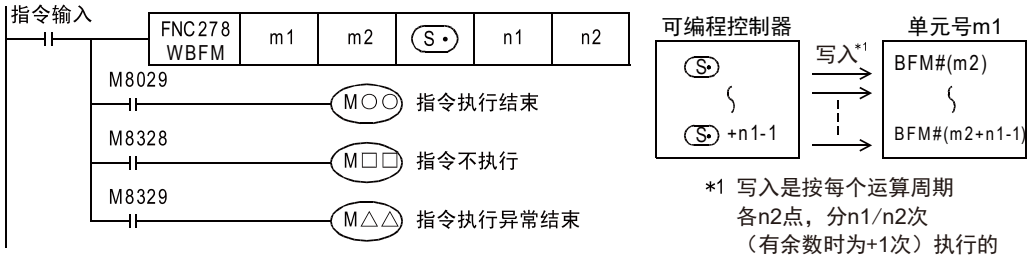
▲: 特殊数据寄存器D除外

功能和动作说明

1. 16位运算(WBFM))

将可编程控制器的(S・)开始的n1点数据，分n1/n2个运算周期(有余数时为+1个运算周期)，按每n2个点传送(写入)到单元号m1的特殊功能模块/单元的编号m2起始的缓冲存储区(BFM)中。

→ 关于单元号、缓冲存储区(BFM)编号、注意要点、程序举例，请参考31.1.1项



*1 写入是按每个运算周期各n2点，分n1/n2次（有余数时为+1次）执行的

- 如指令正常结束，则指令执行结束标志位M8029为ON，如指令异常结束，则指令执行异常结束标志位M8329为ON。
- 针对相同的单元号，正在执行其他步的RBFM(FNC 278)指令或WBFM(FNC 279)指令时，指令不执行的标志位M8328为ON，指令的执行处于待机状态。
  当其他的对象指令执行结束后，待机状态的指令才会解除待机状态，然后执行指令。

相关软元件

→ 关于指令执行结束、指令执行异常结束标志位的使用方法，请参考6.5.2项

软元件	名称	内容
M8029	指令执行结束	当指令正常结束时为ON。
M8328	指令不执行	针对相同的单元号，正在执行其他步中的RBFM(FNC 278)指令或是WBFM(FNC 279)指令时为ON。
M8329	指令执行异常结束	当指令异常结束时为ON。

相关指令

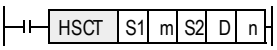
指令	内容
FROM(FNC 78)	BFM的读出
TO(FNC 79)	BFM的写入
RBFM(FNC 278)	BFM分割读出

错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 不存在单元号m1时。(错误代码：K6708)

32. 高速处理2—FNC 280～FNC 289

FNC No.	指令记号	符号	功能	参考
280	HSCT		高速计数器的表格比较	32.1节
281	—			—
282	—			—
283	—			—
284	—			—
285	—			—
286	—			—
287	—			—
288	—			—
289	—			—

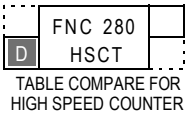
32.1 FNC 280 - HSCT/高速计数器表比较

A8 A8C

概要

将预先制作好的数据表格和高速计数器的当前值做比较，可以对最大16点输出进行置位及复位的指令。

1. 指令格式



16 位指令	指令记号	执行条件
	—	
	—	

32 位指令	指令记号	执行条件
21 步	DHSCT	连续执行型
	—	

2. 设定数据

操作数种类	内容	数据类型
(S1·)	保存数据表格的软元件起始编号	BIN16位/32位
m	数据表格数(比较点数)[1≦m≦128]	BIN32位
(S2·)	高速计数器(C235～C255)的编号	BIN32位
(D·)	动作输出软元件的起始编号	位
n	动作输出点数[1≦n≦16]	BIN32位

3. 对象软元件

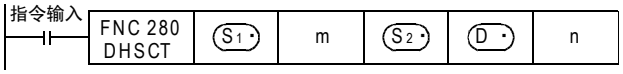
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S1·)														●	●				●					
m																				●	●			
(S2·)													▲						●					
(D·)		●	●			●													●					
n																				●	●			

▲中，仅仅可以指定高速计数器C235～C255。

功能和动作说明

1. 32位运算(DHSCT)

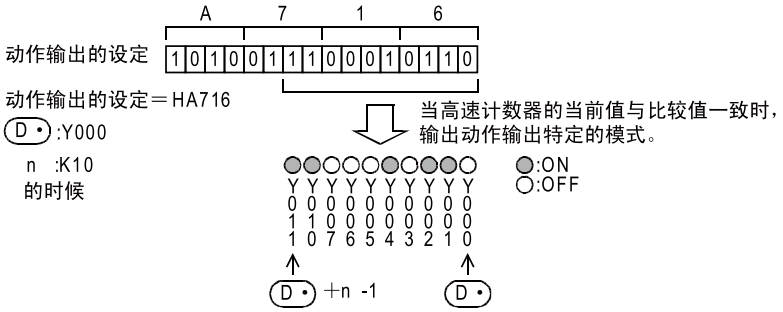
从数据表格上面开始依次对(S1·)开始的(3×m)点的数据表格(下述)和(S2·)指定的高速计数器的当前值进行比较，并将数据表格中指定的输出状态输出(置位/复位)到(D·)～(D·)+n-1中。



比较用数据表格

表格编号	比较数据	动作输出的设定 (置位[1]/复位[0])	动作输出目标软元件
0	(S1・)+1, (S1・)	(S1・)+2	(D・) ～ (D・)+n-1
1	(S1・)+4, (S1・)+3	(S1・)+5	
2	(S1・)+7, (S1・)+6	(S1・)+8	
≀	≀	≀	
m-2	(S1・)+3m-5, (S1・)+3m-6	(S1・)+3m-4	
m-1	(S1・)+3m-2, (S1・)+3m-3	(S1・)+3m-1	

动作输出的设定置位[1]/ 复位[0][最大16 点]



1) 这个指令被执行后，数据表格的最上面的数据被设定为最初的比较对象数据。

2) 如果(S2・)中指定的高速计数器的当前值和作为比较对象的表格数据一致时，在比较对象的数据表格中指定的动作输出会(D・) ～ (D・)+n-1。
此外，当(D・)中指定了输出Y时，不等到END指令再执行输出刷新，而是直接执行输出的处理。
但是，当指定了输出(Y)时，请务必指定软元件号的最低1位数为0。
例如) Y000、Y010、Y020等

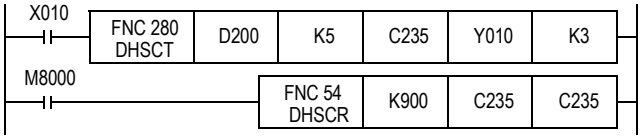
3) 2)之后的表格计数器(D8138)的当前值被「+1」。

4) 作为比较对象的表格数据转移到下一个数据中。

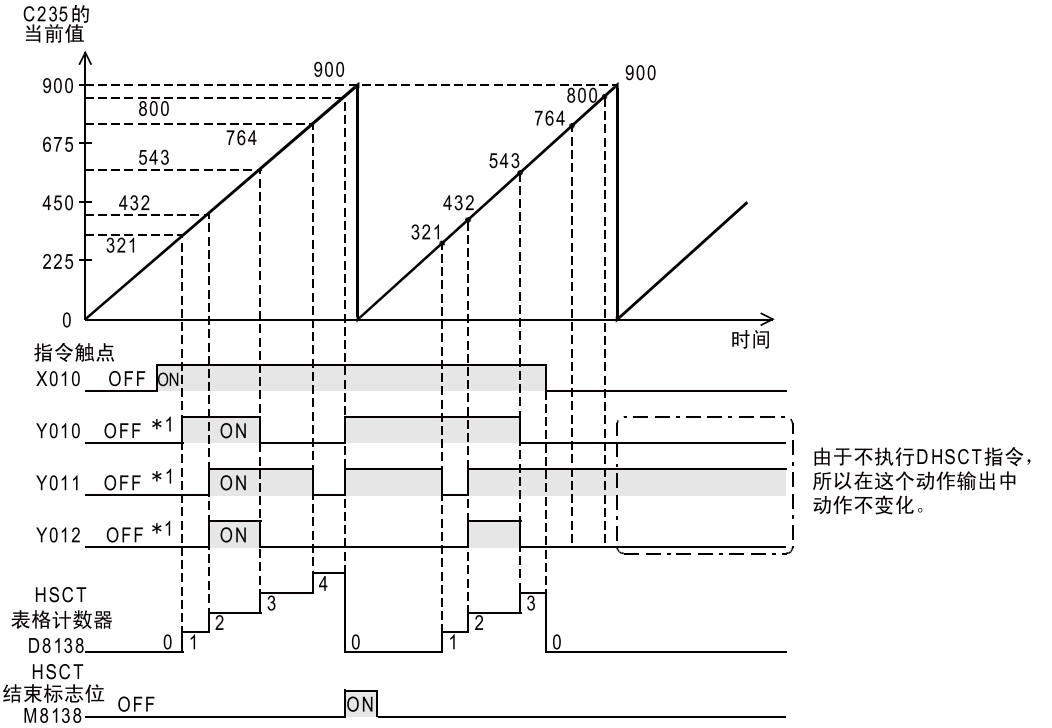
5) 在表格计数器(D8138)的当前值变为m以前，重复2)～3)的动作。
当前值变为m后结束标志位M8138置ON，返回到1)的动作。此时，表格计数器被复位(D8138=0)。

6) 指令触点OFF(指令执行OFF)后，中止指令的执行，表格计数器(D8138)复位(D8138=0)。

动作例子



表格编号	比较数据		SET/RESET模式		表格计数器(D8138)
	软元件	比较值	软元件	动作输出设定值	
0	D201,D200	K321	D202	H0001	0↓
1	D204,D203	K432	D205	H0007	1↓
2	D207,D206	K543	D208	H0002	2↓
3	D210,D209	K764	D211	H0000	3↓
4	D213,D212	K800	D214	H0003	4↓0开始重复



*1. 不执行该指令时，对输出不作处理。
在这个动作例子中，为说明起见，开始都假定为OFF。

2. 相关软元件

软元件	名称	内容
M8138	HSCT(FNC 280)结束标志位	当最后一个表格(m-1)号的动作结束时为ON。
D8138	HSCT(FNC 280)表格计数器	保存作为比较对象的表格编号。

注意要点

- 在程序中只能执行1次该指令。
  被编写2次以上时，第2次以后的指令会出现运算错误，而不被执行。
  (错误代码: K6765)
- 这个指令是在初次执行指令后的END指令中构成数据表格。因此，动作的输出要从第2次扫描以后开始动作。
- DHSCT指令(FNC 280)、DHSCS指令(FNC 53)、DHSCR指令(FNC 54)和DHSZ指令(FNC 55)可以同时(同一运算周期)执行的指令数为32个以下。第33个以后的指令，会出现运输错误，而不被执行。(错误代码: K6705)
- 在 (D・) 中指定了输出Y时，不等到END指令再执行输出刷新，就直接做输出处理。
  但是，当指定了输出(Y)时，请务必将软元件编号的最低1位数指定为0。
  例如) Y000、Y010、Y020等
- 对 (S2・) 指定的高速计数器进行变址修饰时，所有的高速计数器都作为软件计数器使用。
- 在该指令中，比较对象数据仅仅指1个(1行)的表格。
  直到与作为对象的比较数据一致为止，都不会移动到下一行表格。
  因此，象前一页中的动作例子，在比较用表格中，高速计数器的当前值为增计数时，请务必注意，指令执行时，高速计数器的当前值比表格1的比较值要小。
- 在A8/A8C可编程控制器中使用该指令时，硬件计数器(C235、C236、C237、C238、C239、C240、C244(OP)、C245(OP)、C246、C248(OP)、C251、C251)会自动地切换成软件计数器，并影响各计数器的最高频率以及综合频率。

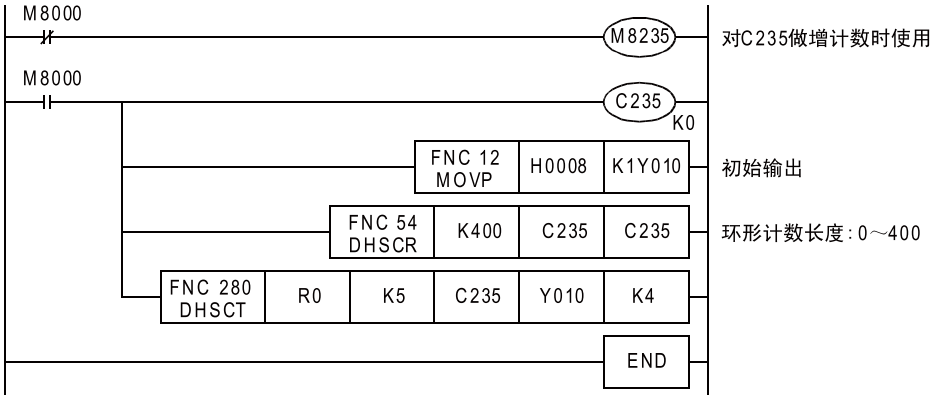
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- (S2・) 中指定了高速计数器C235～C255以外的软元件时。(错误代码: K6706)
- (S1・) 指定的软元件编号开始的3m-1个软元件超出了该软元件的末尾编号时
  (错误代码: K6706)
- (D・) 指定的软元件编号开始的n点软件超出了该软元件编号的末尾编号时
  (错误代码: K6706)
- 这个指令在程序中使用了2次以上时(错误代码: K6765)
- DHSCT指令(FNC 280)、DHSCS指令(FNC 53)、DHSCR指令(FNC 54)和DHSZ指令(FNC 55)可以同时(同一运算周期)执行的指令数为32个以下。第33个以后的指令，会出现运输错误，而不被执行。(错误代码: K6705)

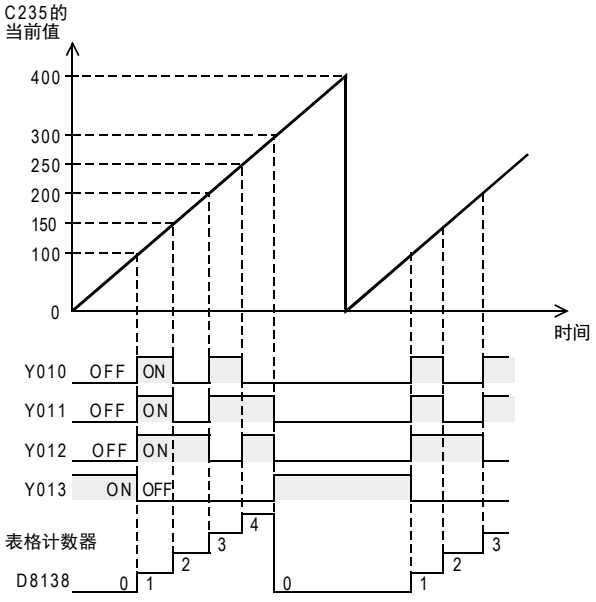
程序举例

将C235的当前值(对X000计数)和R0以后的软元件中设定的比较用数据表进行比较，然后以指定的模式输出到Y010～Y013中的程序。

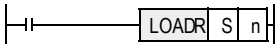
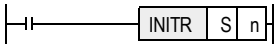
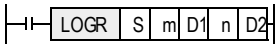
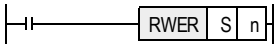
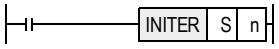


动作

表格编号	比较数据		SET/RESET模式		表格计数器(D8138)
	软元件	比较值	软元件	动作输出设定值	
0	R1,R0	K100	R2	H0007	0↓
1	R4,R3	K150	R5	H0004	1↓
2	R7,R6	K200	R8	H0003	2↓
3	R10,R9	K250	R11	H0006	3↓
4	R13,R12	K300	R14	H0008	4↓0开始重复



33. 扩展文件寄存器控制—FNC 290～FNC 299

FNC No.	指令记号	符号	功能	参考
290	LOADR		读出扩展文件寄存器	33.1节
291	SAVER		成批写入扩展文件寄存器	33.2节
292	INITR		扩展寄存器的初始化	33.3节
293	LOGR		登录到扩展寄存器	33.4节
294	RWER		扩展文件寄存器的删除・写入	33.5节
295	INITER		扩展文件寄存器的初始化	33.6节
296	—			—
297	—			—
298	—			—
299	—			—

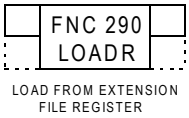
33.1 FNC 290 - LOADR/读出扩展文件寄存器

A8 A8C

概要

将保存在存储器盒(闪存、EEPROM)或主机内置EEPROM中的扩展文件寄存器(ER)的当前值读出(传送) 到可编程控制器内置RAM的扩展寄存器(R)中的指令。

1. 指令格式



16位指令	指令记号	执行条件
5步	LOADR	连续执行型
	LOADRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	保存数据的扩展寄存器的软元件编号 (作为数据传送源的扩展文件寄存器与 (S·) 的编号相同)	BIN16位
n	读出(传送)点数 (A8/A8C: 0≤n≤32767)	

3. 对象软元件

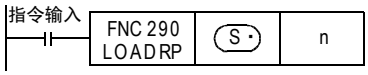
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)															●				●					
n														●						●	●			

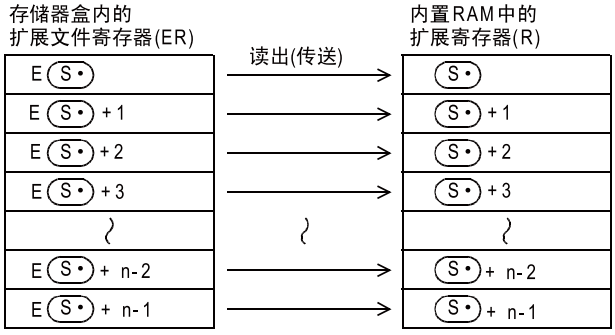
功能和动作说明

1. 16位运算((LOADR/LOADRP)

1) A8/A8C可编程控制器的情况下

将存储器盒(闪存)中保存的扩展文件寄存器(ER)(编号与 (S·) ～ (S·)+n-1的扩展寄存器编号相同)的内容(当前值)，读出(传送)到可编程控制器内置RAM的 (S·) ～ (S·)+n-1的扩展寄存器(R)中。





－以软元件的1点为单位执行读出(传送)，最多可以读出(传送)32768点。

－ 在这个指令中， 不需要像SAVER指令(FNC 291)， INITR指令(FNC 292)以及LOGR指令(FNC 293)一样以段为单位执行。

－ 当指定为n=0时， 作为n=32768执行指令。

注意要点

1. 有关存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点。

- A8/A8C可编程控制器的情况下
  存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。

  即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，也不会被计入存储器的写入次数。但是，执行SAVER(FNC 291)、LOGR(FNC 293)指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时，因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

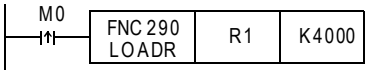
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 当要传送的软元件的末尾软元件编号超出了32767时。(错误代码: K6706)
  此时，对到末尾软元件编号的R32767为止的软元件执行读出(传送)。
- 未连接存储器盒时。(错误代码: K6771)

程序举例

当M0为ON时，存储器盒中的扩展文件寄存器ER1～ER4000(4000点)的内容(当前值)被读出(传送)到内置RAM中的扩展寄存器R1～R4000(4000点)中的程序。



存储器盒内的扩展文件寄存器			内置RAM中的扩展寄存器	
软元件编号	当前值	读出（传送）	软元件编号	当前值
ER1	K100	→	R1	K100
ER2	K50	→	R2	K50
ER3	H0003	→	R3	H0003
ER4	H0101	→	R4	H0101
⋮	⋮	⋮	⋮	⋮
ER3999	K55	→	R3999	K55
ER4000	K59	→	R4000	K59

33.2 FNC 291 - SAVER/成批写入扩展文件寄存器

A8 A8C

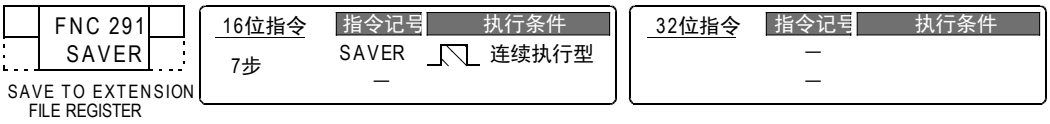
概要

用于将可编程控制器内置RAM的任意点数的扩展寄存器(R)的当前值1段(2048点)写入存储器盒(闪存)的扩展文件寄存器(ER)中时使用。

此外，A8C可编程控制器、A8可编程控制器中，还有仅写入(传送)任意点数的RWER指令(FNC 294)。如果使用RWER指令，就无需每次都执行INITR(FNC 292)指令和INITER(FNC 295)指令。

→ 关于RWER指令，请参考33.5节

1. 指令格式



2. 设定数据

操作数种类	内容	数据类型
(S·)	保存数据的扩展寄存器的软元件编号 但是，只可以指定扩展寄存器的段的起始软元件编号。	BIN16位
n	每个运算周期的写入(传送)点数0≦n≦2048	
(D·)	保存已经写入的点数的软元件编号	

3. 对象软元件

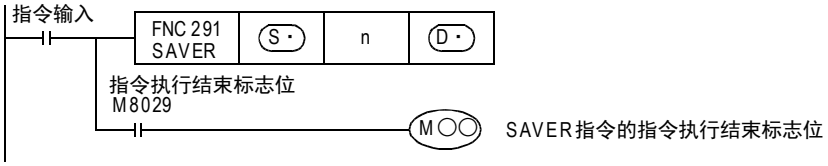
操作数种类	位软元件							字软元件												其他				
	系统・用户						位数指定					系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)															●				●					
n																				●	●			
(D·)														●					●					

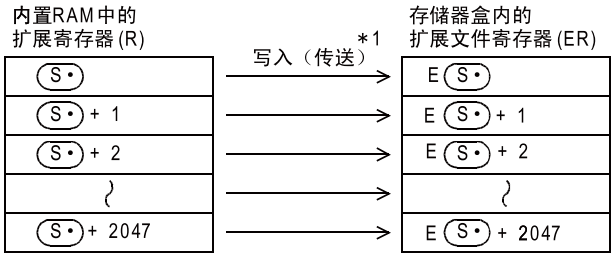
功能和动作说明

1. 16位运算(SAVER)

通过2048/n个运算周期(如有余数时，为+1个扫描)将(S·)～(S·)+2047的扩展寄存器(R)的内容(当前值)写入存储器盒(闪存)中具有相同编号的扩展文件寄存器(ER)中。

指令执行过程中，在(D·)中保存已经写入的点数。





＊1 每个运算周期写入（传送）n点。

- 以段为单位(2048点)执行写入。
  各段的起始软元件编号如下所示。

段编号	起始软元件编号	写入软元件范围	段编号	起始软元件编号	写入软元件范围
段0	R0	ER0～ER2047	段8	R16384	ER16384～ER18431
段1	R2048	ER2048～ER4095	段9	R18432	ER18432～ER20479
段2	R4096	ER4096～ER6143	段10	R20480	ER20480～ER22527
段3	R6144	ER6144～ER8191	段11	R22528	ER22528～ER24575
段4	R8192	ER8192～ER10239	段12	R24576	ER24576～ER26623
段5	R10240	ER10240～ER12287	段13	R26624	ER26624～ER28671
段6	R12288	ER12288～ER14335	段14	R28672	ER28672～ER30719
段7	R14336	ER14336～ER16383	段15	R30720	ER30720～ER32767

- 在n中指定了0的值时，作为n=2048执行指令。
- 当2048点的写入(传送)结束时，指令执行结束，指令执行结束标志位M8029变为ON。
- 在(D・)中保存已经写入的点数。

2. 相关软元件

→ 有关指令执行结束的标志位的使用方法，参考6.5.2项

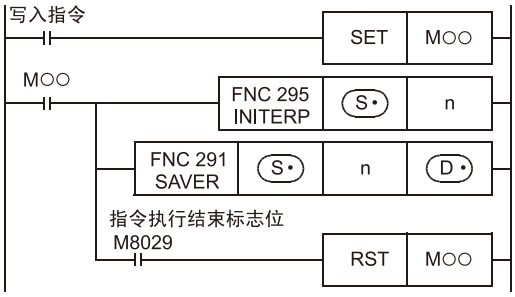
软元件编号	名称	内容
M8029	指令执行结束	当对象指令的处理结束时，指令执行结束标志位M8029为ON。 但是，作为这个标志位的对象指令有时候在程序会存在多个。因此，请在这个指令的正下方使用这个标志位的a触点，制作这个指令专用的指令执行结束标志位。

注意要点

1. 对存储器盒写入数据时的注意事项

由于存储器盒的存储介质为闪存，所以使用这个指令将数据写入到存储器盒中的扩展文件寄存器时，请务必注意以下内容。

- 所有点(2048点)的写入所需的时间大约为340ms。在n中指定了K0或K2048时，执行这个指令的运算周期会比平时长大约340ms。
  当这个时间对运算周期带来不良影响时，请使写入的执行时间在2个运算周期以上。
  如果跨多个运算周期执行写入时，请将n设定为K1～K1024。
- 指令执行过程中请勿中断。如果中断，则扩展文件寄存器中的数据可能会变成无法预料的数据。
  由于断开电源等中断了指令的执行时，请在再次上电以后，按照下一页的②的步骤再次执行指令。



2. 有关扩展文件寄存器的初始化

对作为对象的扩展文件寄存器(ER)，请在SAVER 指令的前面执行INITER(FNC 295)指令，或是INITR(FNC 292)指令。如果不执行就驱动SAVER指令时，会出现运算错误(错误代码: K6770)。
因此，请在程序中按照以下的步骤执行SAVER指令。

- A8/A8C可编程控制器

A8 A8C

① 将1个段(2048点)的扩展文件寄存器(R)的数据保存到扩展文件寄存器

1) 对作为SAVER指令的写入对象的扩展文件寄存器(ER)执行INITER(FNC 295)指令。
2) 执行SAVER命令。

② 将任意点数的扩展寄存器(R)的内容保存到扩展文件寄存器(ER)中时

请使用RWER指令。

→ 关于RWER(FNC294)指令，请参考33.5节

- A8C可编程控制器

A8C

① 将1个段(2048点)的数据保存到扩展文件寄存器(ER)中时

当扩展寄存器(R)中已经存在要保存在扩展文件寄存器(ER)中的数据时，请使用②的步骤。

1) 对作为SAVER指令的写入对象的扩展寄存器(R)，以及扩展文件寄存器(ER)执行INITR(FNC 292)指令。
2) 在作为写入对象的扩展寄存器(R)中保存(存储)数据。
3) 执行SAVER命令。

② 将保存在扩展寄存器(R)中的1个段(2048点)的数据保存到扩展文件寄存器(ER)中时

1) 使用BMOV(FNC 15)指令，将被SAVER 指令作为写入对象的扩展寄存器(R)的数据，暂时避让保存到数据寄存器，或是未使用的2048点的扩展寄存器(R)中。
2) 对作为SAVER指令的写入对象的扩展寄存器(R)、以及扩展文件寄存器(ER)执行INITR(FNC 292)指令。
3) 使用BMOV(FNC 15)指令，将1)中被暂时避让保存的数据(2048点)返回到作为写入对象的扩展寄存器(R)中。
4) 执行SAVER命令。

3. 有关存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点。

- 存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  存储器的写入次数请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。
- 即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，也不会被计入存储器的写入次数。但是，执行SAVER(FNC 291)、LOGR(FNC 293) 指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时，
  因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

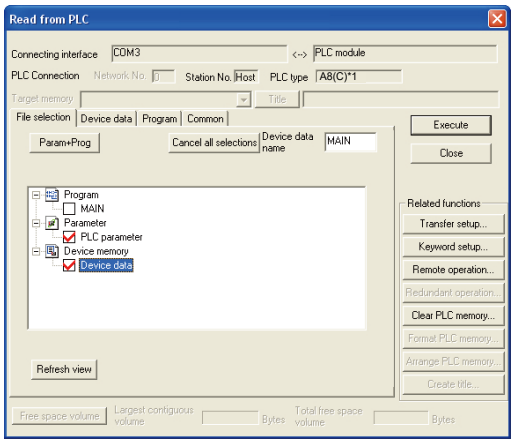
- 当(S・)中设定了扩展文件寄存器的段的起始软元件编号以外的数字时。
  (错误代码: K6706)
- 未连接存储器盒时。(错误代码: K6771)
- 存储器盒的写保护开关设置在ON时。(错误代码: K6770)
- 由于忘记初始化等，使得在数据写入后的比对中产生不一致时。(错误代码: K6770)
  发生这类错误时，如果丢失了扩展寄存器(R)的内容会有问题时，请使用以下的方法对扩展寄存器(R)的当前值(数据)进行备份。

1) 停止可编程控制器。

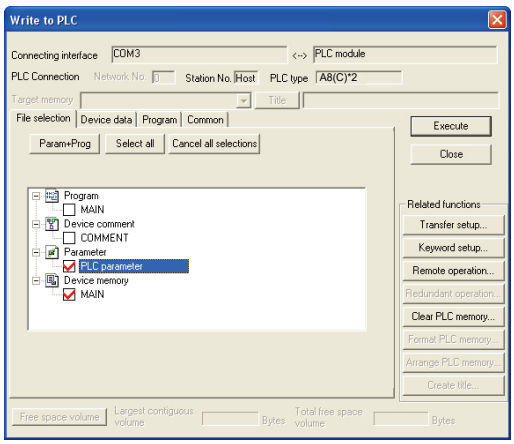
2) 在HCP-works中创建新的工程。
   允许在当前的工程中改写数据时，不需要这个步骤。

3) 在HCP-works中读出扩展寄存器(R)的内容。
   ①选择[在线→PLC读出]，显示PLC读出的窗口。
   ②点击PLC参数和软元件数据的项目，显示出选中的选项卡。
   ③点击执行按键，执行读出。
   ④读出结束后，请保存工程。

4) 请用前页的[注意要点的“对存储器盒写入数据时的注意事项”]中记载的程序更改可编程控制器中的程序。



1) 将在HCP-works中暂时避让保存的数据写入到可编程控制器中。
   ① 选择[在线→PLC写入]，显示PLC写入的窗口。
   ② 点击PLC参数和软元件数据的项目，显示出选中的选项卡。
   ③ 点击执行按键，执行写入。



2) 将可编程控制器从STOP切换成RUN，使程序运行，从而将数据保存到存储器盒中的扩展文件寄存器中。

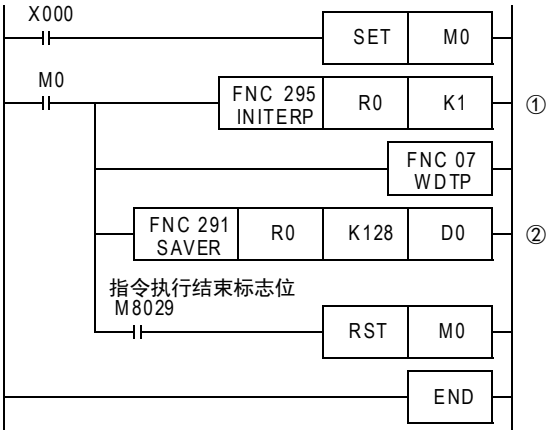
程序举例

1) A8/A8C可编程控制器。

A8 A8C

当X000为ON时，将设定数据用的扩展寄存器R10～R19(0段)的变更内容，反映到扩展文件寄存器(ER)中的程序。(每个运算周期写入128点)

程序



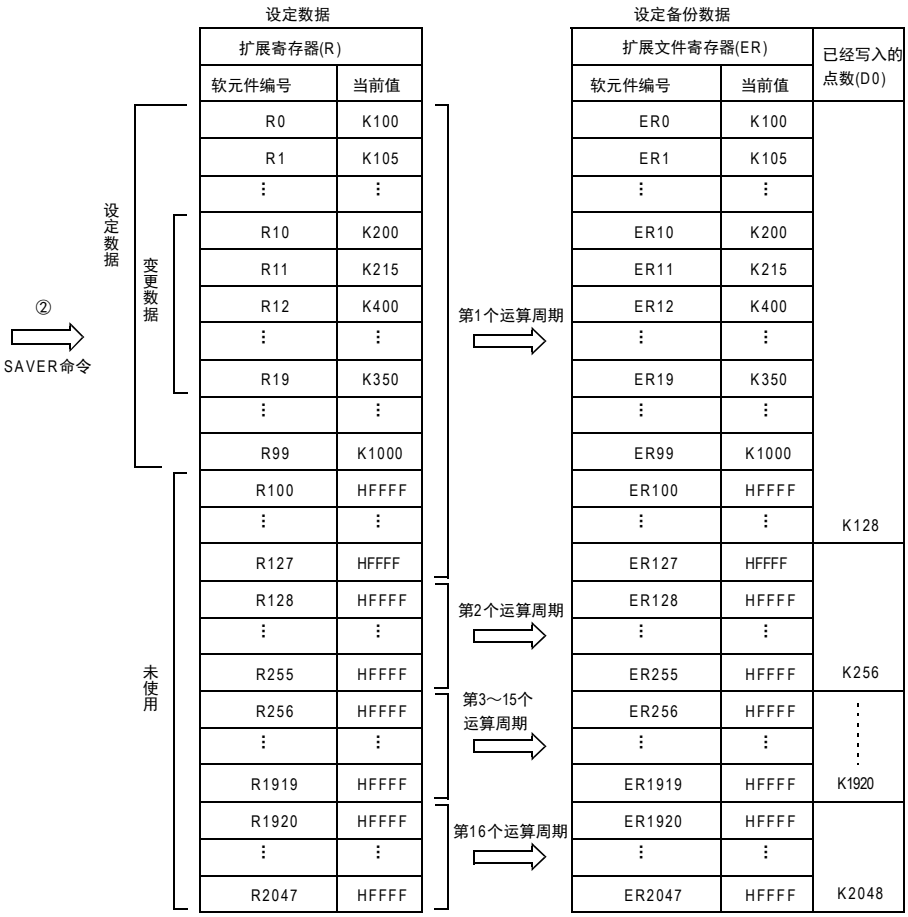
动作

设定数据 设定备份数据 设定备份数据

	扩展寄存器(R)		扩展文件寄存器(ER)		扩展文件寄存器(ER)	
	软元件编号	当前值	软元件编号	当前值	软元件编号	当前值
设定数据(A)	R0	K100	ER0	K100	ER0	HFFFF
	R1	K105	ER1	K105	ER1	HFFFF
	⋮	⋮	⋮	⋮	⋮	⋮
变更数据	R10	K200	ER10	K300	ER10	HFFFF
	R11	K215	ER11	K330	ER11	HFFFF
	R12	K400	ER12	K350	ER12	HFFFF
	⋮	⋮	⋮	⋮	⋮	⋮
	R19	K350	ER19	K400	ER19	HFFFF
	⋮	⋮	⋮	⋮	⋮	⋮
	R99	K1000	ER99	K1000	ER99	HFFFF
未使用	R100	HFFFF	ER100	HFFFF	ER100	HFFFF
	⋮	⋮	⋮	⋮	⋮	⋮
	R2047	HFFFF	ER2047	HFFFF	ER2047	HFFFF

① INITER指令 ② SAVER指令

往下一页

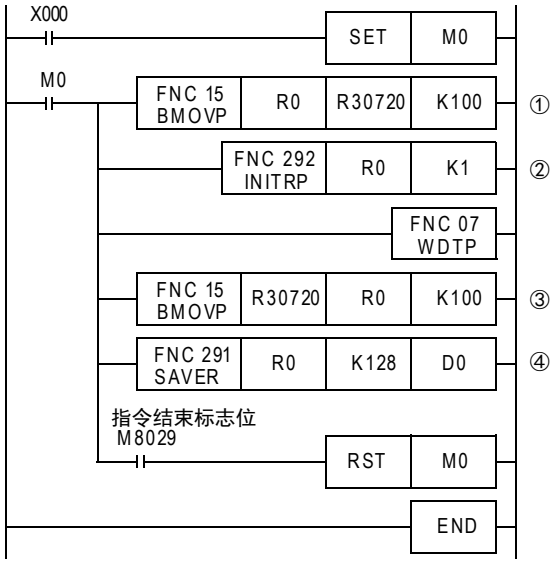

设定数据
设定备份数据
扩展寄存器(R)
扩展文件寄存器(ER)
已经写入的点数(D0)
软元件编号
当前值
R0 K100
R1 K105
R10 K200
R11 K215
R12 K400
R19 K350
R99 K1000
R100 HFFFF
R127 HFFFF
R128 HFFFF
R255 HFFFF
R256 HFFFF
R1919 HFFFF
R1920 HFFFF
R2047 HFFFF
ER0 K100
ER1 K105
ER10 K200
ER11 K215
ER12 K400
ER19 K350
ER99 K1000
ER100 HFFFF
ER127 HFFFF
ER128 HFFFF
ER255 HFFFF
ER256 HFFFF
ER1919 HFFFF
ER1920 HFFFF
ER2047 HFFFF
K128
K256
K1920
K2048
设定数据
变更数据
未使用
②
SAVER命令
第1个运算周期
第2个运算周期
第3～15个运算周期
第16个运算周期

2) A8C可编程控制器。

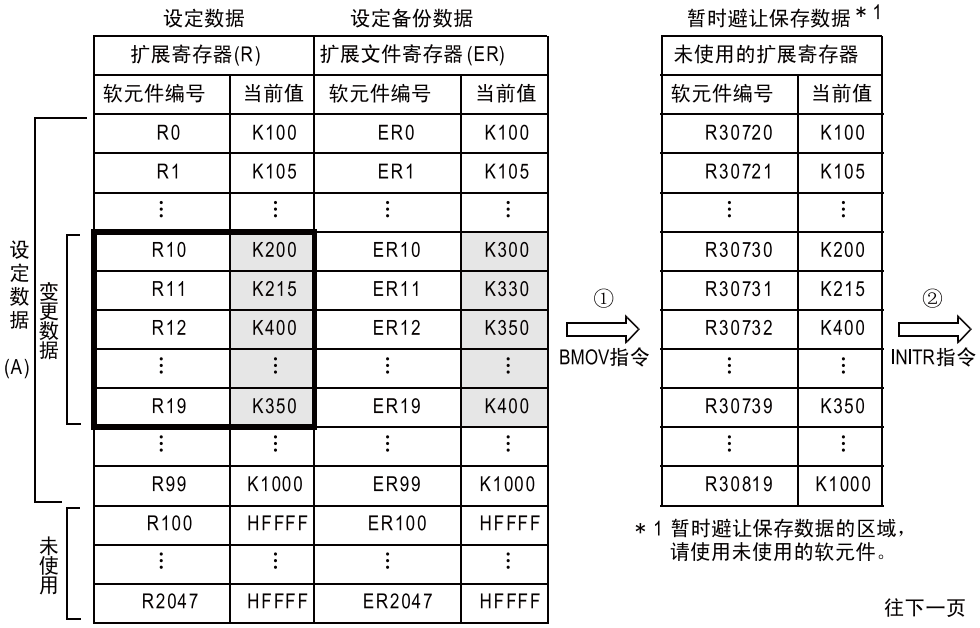
A8C

当X000为ON时，将设定数据用的扩展寄存器R10～R19(0段)的变更内容，反映到扩展文件寄存器(ER)中的程序。(每个运算周期写入128点)

程序

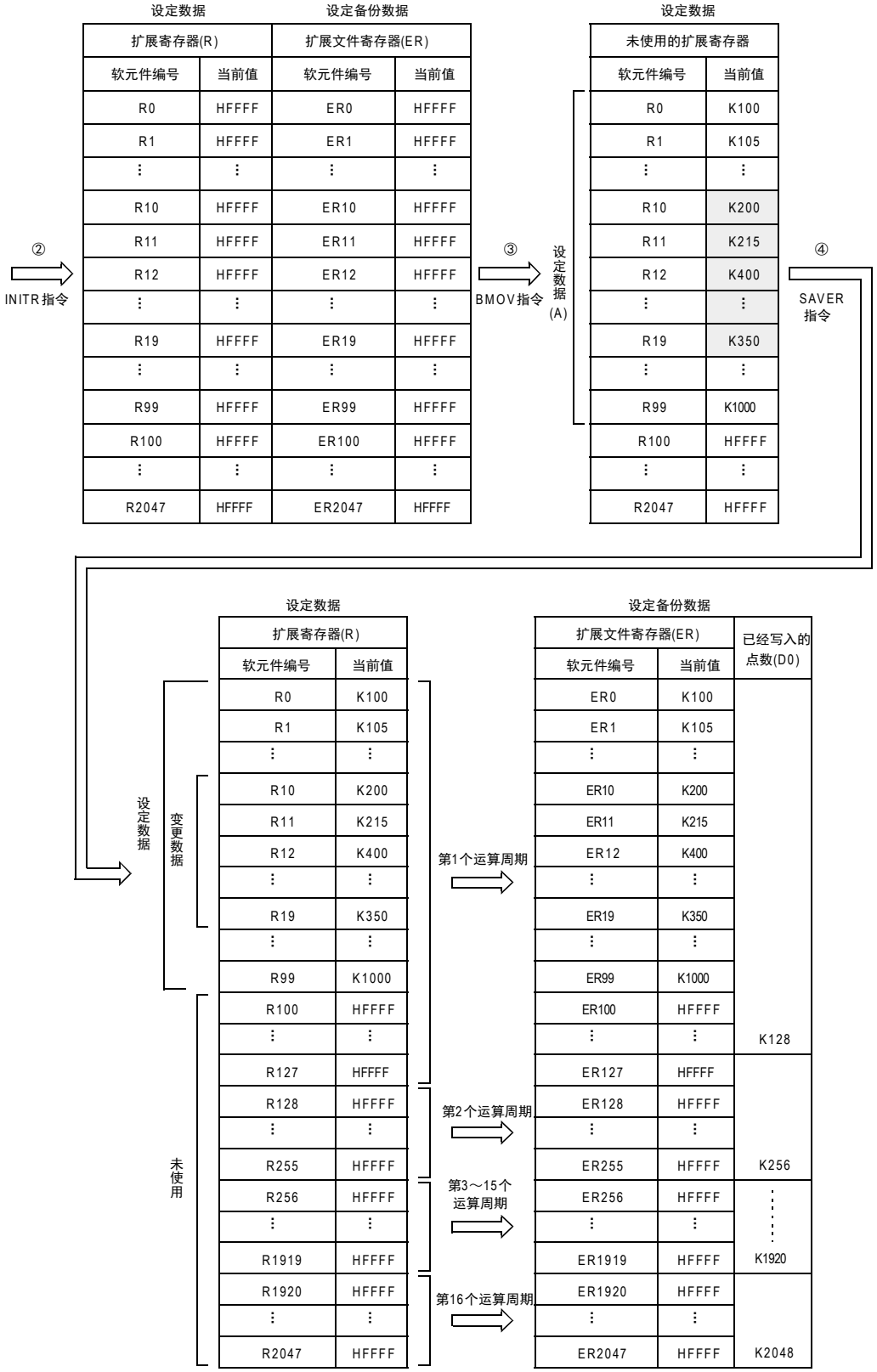


动作



＊1 暂时避让保存数据的区域，请使用未使用的软元件。

往下一页


设定数据
设定备份数据
扩展寄存器(R)
扩展文件寄存器(ER)
软元件编号
当前值
R0 HFFFF ER0 HFFFF
R1 HFFFF ER1 HFFFF
R10 HFFFF ER10 HFFFF
R11 HFFFF ER11 HFFFF
R12 HFFFF ER12 HFFFF
R19 HFFFF ER19 HFFFF
R99 HFFFF ER99 HFFFF
R100 HFFFF ER100 HFFFF
R2047 HFFFF ER2047 HFFFF
②
INITR指令
③
BMOV指令
设定数据(A)
设定数据
未使用的扩展寄存器
R0 K100
R1 K105
R10 K200
R11 K215
R12 K400
R19 K350
R99 K1000
R100 HFFFF
R2047 HFFFF
④
SAVER指令
设定数据
变更数据
未使用
设定数据
设定备份数据
扩展寄存器(R)
扩展文件寄存器(ER)
已经写入的点数(D0)
R0 K100 ER0 K100
R1 K105 ER1 K105
R10 K200 ER10 K200
R11 K215 ER11 K215
R12 K400 ER12 K400
R19 K350 ER19 K350
R99 K1000 ER99 K1000
R100 HFFFF ER100 HFFFF
R127 HFFFF ER127 HFFFF K128
R128 HFFFF ER128 HFFFF
R255 HFFFF ER255 HFFFF K256
R256 HFFFF ER256 HFFFF
R1919 HFFFF ER1919 HFFFF K1920
R1920 HFFFF ER1920 HFFFF
R2047 HFFFF ER2047 HFFFF K2048
第1个运算周期
第2个运算周期
第3～15个运算周期
第16个运算周期

33.3 FNC 292 - INITR/扩展寄存器的初始化

A8 A8C

概要

通过LOGR(FNC 293)指令开始登录数据之前，对可编程控制器内置RAM中的扩展寄存器(R)和存储器盒(闪存)中的扩展文件寄存器(ER)进行初始化(HFFFF<K-1>)时使用本指令。

此外，在A8C可编程控制器中，使用SAVER(FNC 291)指令写入数据前，对扩展寄存器(ER)进行初始化时，使用本指令。

在A8/A8C可编程控制器中，还有以段为单位只对存储器盒(闪存)中的扩展文件寄存器(ER)进行初始化(HFFFF＜K-1＞)的INITER指令(FNC 295)。

→ 有关SCL(FNC 291)指令，请参考33.2节
→ 有关RWER(FNC 293)指令，请参考33.4节
→ 有关SCL(FNC 295)指令，请参考33.6节

1. 指令格式



16位指令	指令记号	执行条件
5步	INITR	连续执行型
	INITRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S·)	要初始化的扩展寄存器和扩展文件寄存器*1的软元件编号 但是，仅可以指定扩展寄存器的段的起始软元件编号。	BIN16位
n	要初始化的扩展寄存器和扩展文件寄存器的段数	

*1. 未使用存储器盒时，不执行扩展文件寄存器(ER)的初始化。

3. 对象软元件

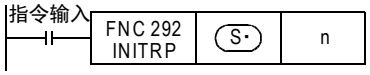
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S·)															●				●					
n																				●	●			

功能和动作说明

1. 16位运算(INITR/INITRP)

对与(S·)指定的可编程控制器内置RAM的扩展寄存器编号相同的存储器盒(闪存)中的扩展文件寄存器开头的n个段，执行初始化(写入初始值HFFFF(K-1))。

但是，按段执行初始化



各段的起始软元件编号如下所示。

段编号	起始软元件编号	初始化软元件范围
段0	R0	R0～R2047、ER0～ER2047
段1	R2048	R2048～R4095、ER2048～ER4095
段2	R4096	R4096～R6143、ER4096～ER6143
段3	R6144	R6144～R8191、ER6144～ER8191
段4	R8192	R8192～R10239、ER8192～ER10239
段5	R10240	R10240～R12287、ER10240～ER12287
段6	R12288	R12288～R14335、ER12288～ER14335
段7	R14336	R14336～R16383、ER14336～ER16383

段编号	起始软元件编号	初始化软元件范围
段8	R16384	R16384～R18431、ER16384～ER18431
段9	R18432	R18432～R20479、ER18432～ER20479
段10	R20480	R20480～R22527、ER20480～ER22527
段11	R22528	R22528～R24575、ER22528～ER24575
段12	R24576	R24576～R26623、ER24576～ER26623
段13	R26624	R26624～R28671、ER26624～ER28671
段14	R28672	R28672～R30719、ER28672～ER30719
段15	R30720	R30720～R32767、ER30720～ER32767

动作(使用存储器盒时)

- 扩展寄存器(R)[内置RAM内]

软元件编号	当前值	
	执行前	执行后
(S・)	H0010	HFFFF
(S・)+1	H0020	HFFFF
(S・)+2	H0011	HFFFF
≀	≀	≀
(S・)+(2048×n)-1	HABCD	HFFFF

- 扩展文件寄存器(ER)[存储器盒内]

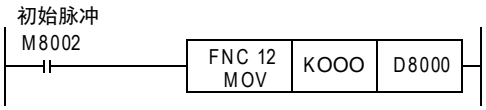
软元件编号	当前值	
	执行前	执行后
(S・)	H1234	HFFFF
(S・)+1	H5678	HFFFF
(S・)+2	H90AB	HFFFF
≀	≀	≀
(S・)+(2048×n)-1	HCDEF	HFFFF

注意要点

1. 关于多个段的初始化

安装了存储器盒时，初始化每个段需要18ms。
(没有安装存储器盒时，为1ms以下。)
对多个段执行初始化时，请务必在编程中采取下述的任意一个对策。

- 在下面的程序中将看门狗定时器的设定值D8000放大。

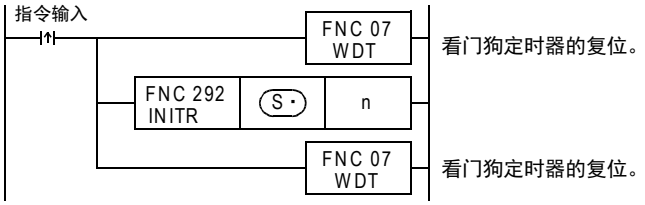


看门狗定时器的设定值基准

看门狗定时器的设定值可以按照下面的步骤将得到的值作为基准值。
但是，当这个基准值在200以下时，无需更改看门狗定时器的设定值。

1) 使用HCP-works将要运行的程序写入到可编程控制器中。
[在线]→[PLC写入]
2) 使用HCP-works的软元件测试功能将D8000(单位: ms)的当前值设定为1000。
[在线]→[调试]→[软元件测试]的[字软元件/缓冲存储区的项目]
3) 将可编程控制器置为RUN，使程序运行(也使该指令动作)。
4) 用HCP-works的软元件成批监控功能，监控最大的扫描时间D8012。(单位: 0.1ms)
5) 请将看门狗定时器最大扫描时间设置为D8012以上。
D8012中的最大的扫描时间以0.1ms为单位保存。
因此，将D8012除以10后再加上50～100，请将所得到的值作为看门狗定时器的设定值D8000(单位: ms)的目标值。

- INITR指令正前方和正后方，如下所示地编写WDT(FNC 07)指令。



INITR的处理时间超过200ms时，请使D8000(单位: ms)的值在INITR的处理时间以上。

2. 有关存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点。

- 存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  存储器的写入次数请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。
- 即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，也不会被计入存储器的写入次数。但是，执行SAVER(FNC 291)、LOGR(FNC 293)指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时，因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

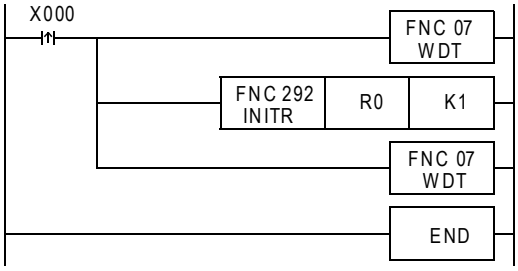
错误

以下一些情况下会发生运算错误，错误标志位M8067置ON，错误代码保存在D8067中。

- 当(S·)中设定了扩展文件寄存器的段的起始软元件编号以外的数字时。
  (错误代码: K6706)
- 要初始化的软元件的编号超出了32767时，到R32767(ER32767)为止的软元件执行初始化。
  (错误代码: K6706)
- 连接存储器盒时，写保护开关设置在ON时。(错误代码: K6770)

程序举例

对段0中的扩展寄存器R0～R2047进行初始化的程序。
但是，使用存储器盒时，请务必注意，执行了这个程序后，扩展文件寄存器ER0～ER2047也会被初始化。



- 扩展寄存器(R)[内置RAM中]

软元件编号	当前值	
	执行前	执行后
RO	H1234	HFFFF
R1	H5678	HFFFF
R2	H90AB	HFFFF
≀	≀	≀
R2047	HCDEF	HFFFF

33.4 FNC 293 - LOGR/登录到扩展寄存器

A8 A8C

概要

这个指令用于执行指定软元件的登录，并可以将已经登录的数据保存到扩展寄存器(R)以及存储器盒中的扩展文件寄存器(ER)中。

1. 指令格式

FNC 293 LOGR P

LOGGING TO EXTENSION REGISTER

16位指令	指令记号	执行条件
11步	LOGR	连续执行型
	LOGRP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	执行登录的对象软元件起始编号*1	BIN16位
m	登录的对象软元件数1≦m≦8000	
(D1)	登录中使用的软元件起始编号	
n	登录中使用的软元件的段数1≦n≦16	
(D2・)	已登录的数据数	

*1. 不能设定C200～C255。

3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)												●	●	●					●					
m														●						●	●			
(D1)															●									
n																				●	●			
(D2・)														●					●					

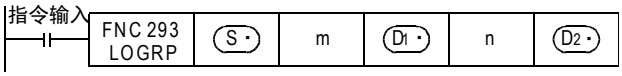
功能和动作说明

1. 16位运算(LOGR/LOGRP)

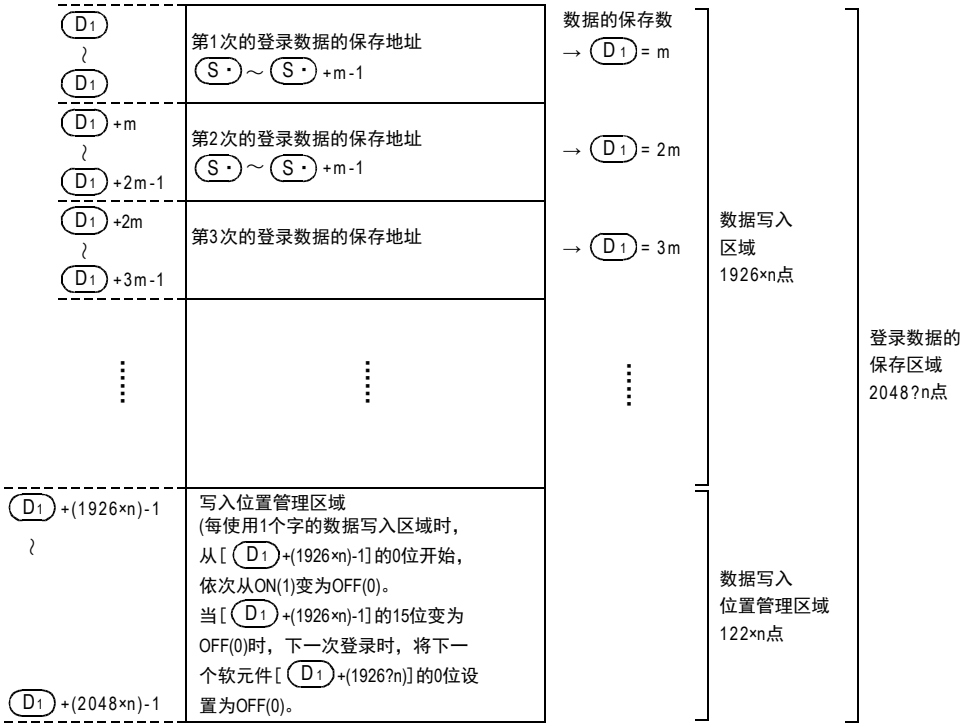
在指令被驱动期间，到(D1)开头的扩展寄存器(R)以及存储器盒中的扩展文件寄存器(ER)的n段的区域被填满为止，一直登录(S・)开始的m点的软元件。

此外，将保存的登录数据的数据数存放在(D2・)中。

未使用存储器盒时，不能对扩展文件寄存器(ER)执行写入。



登录数据的格式



各段的起始软元件编号如下所示。

段编号	起始软元件编号	写入软元件范围
段0	R0	R0～R2047、ER0～ER2047
段1	R2048	R2048～R4095、ER2048～ER4095
段2	R4096	R4096～R6143、ER4096～ER6143
段3	R6144	R6144～R8191、ER6144～ER8191
段4	R8192	R8192～R10239、ER8192～ER10239
段5	R10240	R10240～R12287、ER10240～ER12287
段6	R12288	R12288～R14335、ER12288～ER14335
段7	R14336	R14336～R16383、ER14336～ER16383

段编号	起始软元件编号	写入软元件范围
段8	R16384	R16384～R18431、ER16384～ER18431
段9	R18432	R18432～R20479、ER18432～ER20479
段10	R20480	R20480～R22527、ER20480～ER22527
段11	R22528	R22528～R24575、ER22528～ER24575
段12	R24576	R24576～R26623、ER24576～ER26623
段13	R26624	R26624～R28671、ER26624～ER28671
段14	R28672	R28672～R30719、ER28672～ER30719
段15	R30720	R30720～R32767、ER30720～ER32767

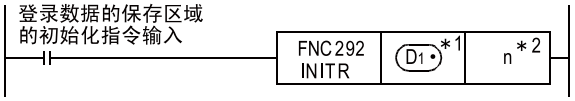
注意要点

1. 有关LOGR指令

LOGR指令连续执行时，在每个运算中都执行登录。
通过1次输入执行1次登录时，请使用脉冲执行型。

2. 使用存储器盒时的注意事项

由于存储器盒的存储介质是闪存，所以在开始登录之前，请务必以段为单位，对数据保存区域初始化。
未经初始化就执行这个指令时，有可能会出现运算错误(错误代码: K6770)



*1 请指定与LOGR指令的D1·相同的内容
*2 请指定与LOGR指令的n相同的内容

3. 有关存储器的允许写入次数

- 访问扩展文件寄存器时，请注意以下一些要点。
  存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  存储器的写入次数请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。
- 即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，就会被计入存储器的写入次数。但是，执行SAVER(FNC 291)、LOGR(FNC 293)指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时，因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

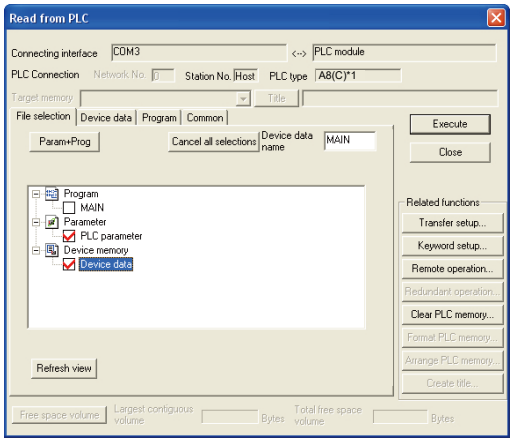
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 当(S・)中设定了扩展文件寄存器的段起始软元件编号以外的数字时。
  (错误代码:K6706)
- 写入数据时，会比较剩余区域和写入数据的量。此时，如果剩余的写入区域不够时，只写入可以写入的点数。(错误代码:K6706)
- 存储器盒的写保护开关设置在ON时。(错误代码:K6770)
- 由于忘记初始化等，使得数据写入后的比对中产生不一致时。(错误代码:K6770)
  发生这类错误时，如果丢失了扩展寄存器(R)的内容会有问题时，请使用以下的方法对扩展寄存器(R)的当前值(数据)进行备份。

1) 停止可编程控制器。
2) 在HCP-works中创建新的工程。
   允许在当前的工程中改写数据时，不需要这个步骤。
3) 在HCP-works中读出扩展寄存器(R)的内容。

① 选择[在线→PLC读出]，显示出PLC读出的窗口。
② 点击PLC参数和软元件数据的项目，显示出选中的选项卡。
③ 点击执行按键，执行读出。
④ 读出结束后，请保存工程。



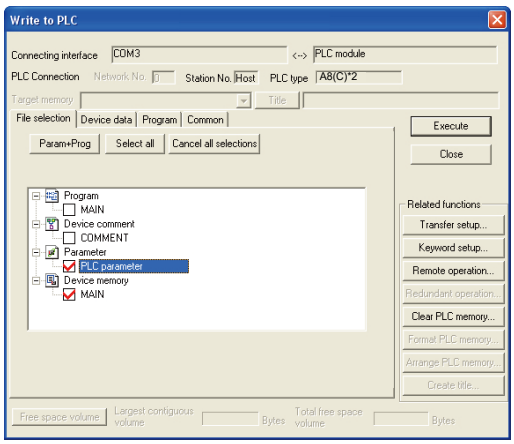
4) 请用前页的 [注意要点的“对存储器盒使用时的注意事项”] 中记载的程序更改可编程控制器中的程序。

5) 将在HCP-works中暂时避让保存的数据写入到可编程控制器中。

① 选择[在线→PLC写入]，显示PLC写入的窗口。

② 点击PLC参数和软元件数据的项目，显示出选中的选项卡。

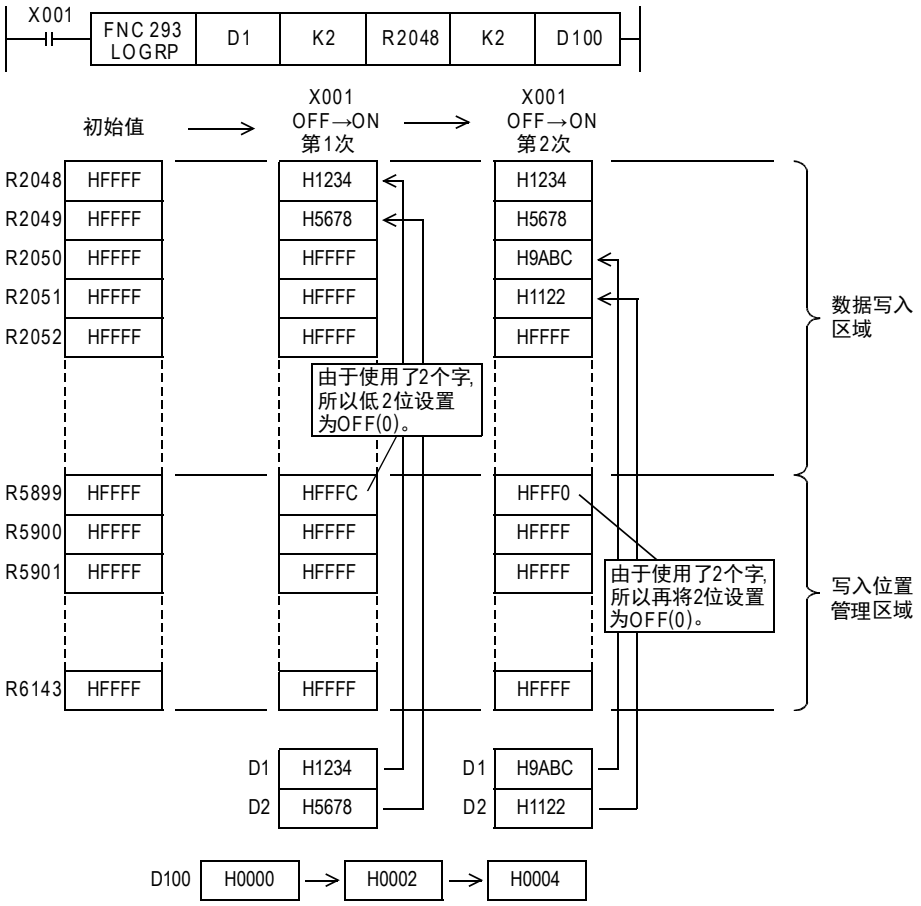
③ 点击执行按键，执行写入。



6) 将可编程控制器从STOP切换成RUN，使程序运行，从而将数据保存到存储器盒中的扩展文件寄存器中。

程序举例

每次X001为ON时，在R2048～R6143的区域中登录D1和D2的程序。



33.5 FNC 294 - RWER/扩展文件寄存器的删除·写入

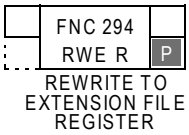
A8 A8C

概要

将可编程控制器内置RAM的扩展寄存器(R)的任意点数的当前值写入存储器盒(闪存、EEPROM)中,或者主机内置EEPROM中的扩展文件寄存器(ER)中时使用。
此外，在HCP works的A8C可编程控制器，由于不支持RWER指令(FNC 294)，
所以请使用SAVER(FNC 291)指令。

→ 关于SAVER(FNC 291)指令，请参考33.2节

1. 指令格式



16位指令	指令记号	执行条件
5步	RWER	连续执行型
	RWERP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	保存数据的扩展寄存器的软元件编号	BIN16位
n	写入(传送)点数 (A8/A8C: 0≦n≦32767)	

3. 对象软元件

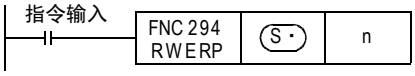
操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)															●				●					
n														●						●	●			

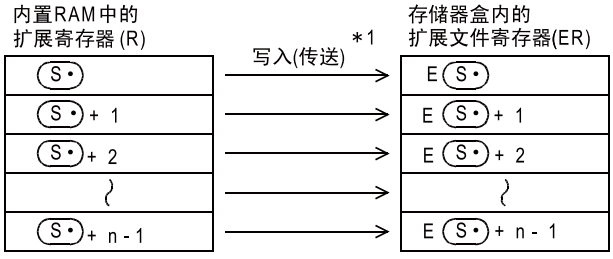
功能和动作说明

1. 16位运算((RWER)

1) A8/A8C可编程控制器的情况下

(S・)开始的n点扩展寄存器(R)的内容(当前值)，写入(传送)到相同编号的存储器盒(闪存)的扩文件寄存器(ER)中。





＊1 当执行指令时，对指定的所有点数都执行写入(传送)

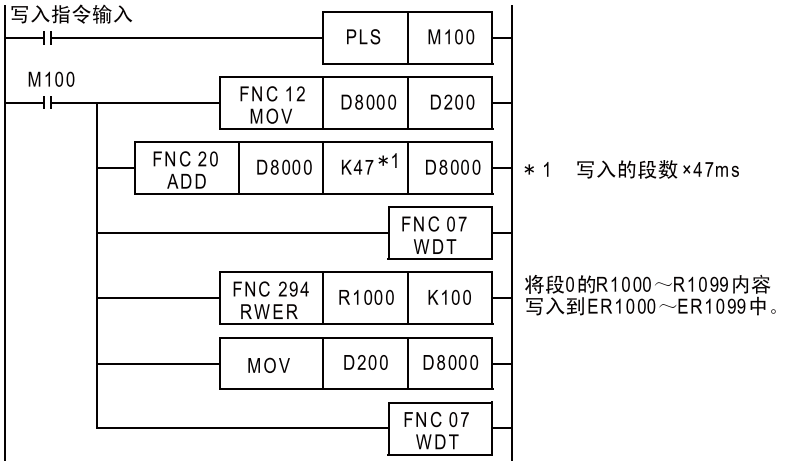
— 当指定了n=0时， 作为n=32768执行命令。

注意要点

1. 对A8/A8C可编程控制器用存储器盒(闪存)写入数据时的注意事项

由于存储器盒的存储介质为闪存，所以使用这个指令将数据写入到存储器盒中的扩展文件寄存器时，请务必注意以下的内容。

- 可以任意指定要写入的扩展寄存器，但是以段为单位执行写入动作。
  因此，写入所需的时间为每段约47ms。当要写入的扩展寄存器跨了2个段时，指令执行的时间大约为94ms。
  请在执行这个指令之前，更改看门狗定时器的设定值D8000。



各段的软元件范围如下所示。

段编号	软元件范围
段0	ER0～ER2047
段1	ER2048～ER4095
段2	ER4096～ER6143
段3	ER6144～ER8191
段4	ER8192～ER10239
段5	ER10240～ER12287
段6	ER12288～ER14335
段7	ER14336～ER16383

段编号	软元件范围
段8	ER16384～ER18431
段9	ER18432～ER20479
段10	ER20480～ER22527
段11	ER22528～ER24575
段12	ER24576～ER26623
段13	ER26624～ER28671
段14	ER28672～ER30719
段15	ER30720～ER32767

- 在指令执行过程中请勿断开电源。如果在执行过程中断电，则有可能使指令的执行被中断。
  如果指令的执行中断，则可能会丢失数据，所以在执行指令之前请务必备份数据。
  → 关于备份的方法，请参考下一页

2. 有关存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点。

- A8A/8C可编程控制器的情况下
  存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  存储器的写入次数请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。
  如要避免这种情况，必须使用脉冲执行型指令。

  即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，也不会被计入存储器的写入次数。但是，执行SAVER(FNC 291)、LOGR(FNC 293)指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时
  因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

数据备份的方法

如果丢失了扩展文件寄存器(ER)以及扩展寄存器(R)的内容会出问题时，请使用如下所述的方法对扩展寄存器(R)的当前值(数据)进行备份。

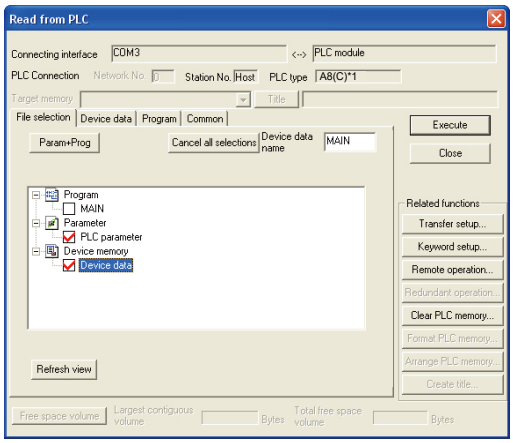
1) 停止可编程控制器。

2) 在HCP-works中创建新的工程。
允许在当前的工程中改写数据时，不需要这个步骤。

3) 在HCP-works中读出扩展文件寄存器(ER)以及扩展寄存器(R)的内容。

① 选择[在线→PLC读出]，显示PLC读出的窗口。

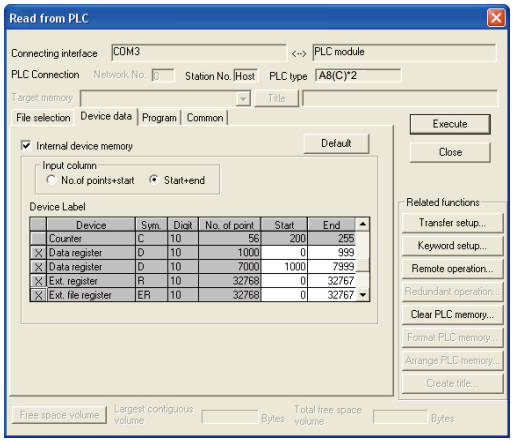
② 点击PLC参数和软元件数据的项目，显示出选中的选项卡。



① 在软元件数据选项卡中，选择扩展文件寄存器以及扩展寄存器。
在HCP-works中，不能设定扩展文件寄存器的范围。

② 点击执行按键，执行读出。

③ 读出结束后，请保存工程。



错误

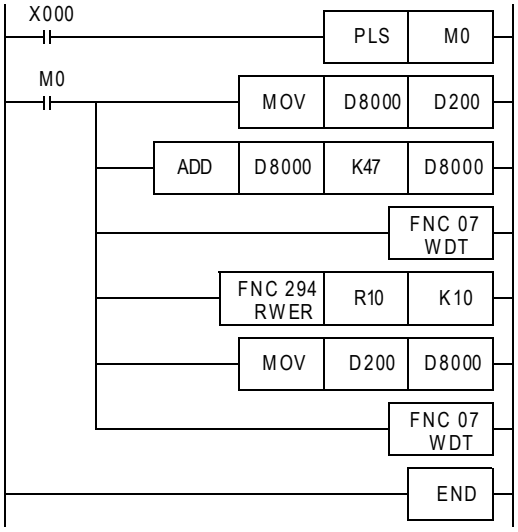
下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 要传送的软元件的末尾软元件编号超出了32767时。(错误代码: K6706)
  此时，对到末尾软元件编号的R32767为止的软元件执行读出(传送)。
- 未连接存储器盒时。(错误代码: K6771)
- 存储器盒的写保护开关设置在ON时。(错误代码: K6770)

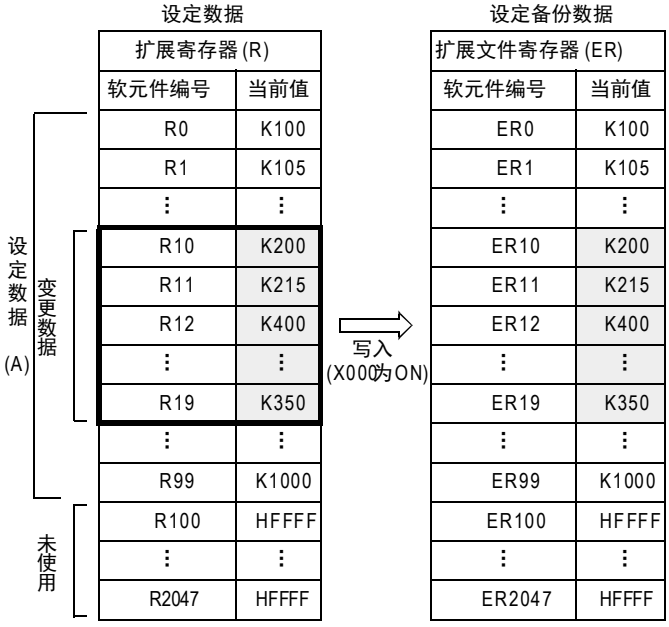
程序举例

当X000为ON时，将设定数据用的扩展寄存器R10～R19(0段)的变更内容，反映到扩展文件寄存器(ER)中的程序。

程序



动作



33.6 FNC295 - INITER/扩展文件寄存器的初始化

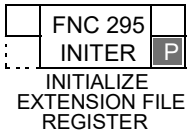
A8 A8C

概要

用于在执行SAVER指令前，对存储器盒(闪存)中的扩展文件寄存器(ER)进行初始化(HFFFF<K-1>)的情况时使用。

→ 关于SAVER(FNC 291)指令，请参考33.2节
→ 关于INITR(FNC 292)指令，请参考33.3节

1. 指令格式



16位指令	指令记号	执行条件
5步	INITER	连续执行型
	INITERP	脉冲执行型

32位指令	指令记号	执行条件
	—	
	—	

2. 设定数据

操作数种类	内容	数据类型
(S・)	与要执行初始化的扩展文件寄存器的软元件编号相同的扩展寄存器的软元件编号但是，仅可以指定扩展寄存器的段的起始软元件编号。	BIN16位
n	要初始化的扩展文件寄存器的段数	

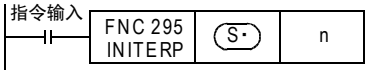
3. 对象软元件

操作数种类	位软元件							字软元件												其他				
	系统・用户							位数指定				系统・用户				特殊模块	变址			常数		实数	字符串	指针
	X	Y	M	T	C	S	D□.b	KnX	KnY	KnM	KnS	T	C	D	R	U□\G□	V	Z	修饰	K	H	E	"□"	P
(S・)															●				●					
n																				●	●			

功能和动作说明

1. 16位运算(INITER/INITERP)

对与(S・)编号相同的存储器盒(闪存)中的扩展文件寄存器(ER)开头的n个段，执行初始化(写入初始值HFFFF(K-1))。
但是，按段执行初始化



各段的起始软元件编号如下所示。

段编号	起始软元件编号	初始化软元件范围
段0	R0	ER0～ER2047
段1	R2048	ER2048～ER4095
段2	R4096	ER4096～ER6143
段3	R6144	ER6144～ER8191
段4	R8192	ER8192～ER10239
段5	R10240	ER10240～ER12287
段6	R12288	ER12288～ER14335
段7	R14336	ER14336～ER16383

段编号	起始软元件编号	初始化软元件范围
段8	R16384	ER16384～ER18431
段9	R18432	ER18432～ER20479
段10	R20480	ER20480～ER22527
段11	R22528	ER22528～ER24575
段12	R24576	ER24576～ER26623
段13	R26624	ER26624～ER28671
段14	R28672	ER28672～ER30719
段15	R30720	ER30720～ER32767

动作

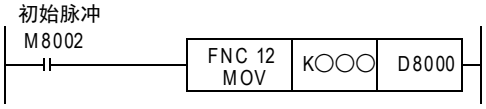
- 扩展文件寄存器(ER)[存储器盒内]

软元件编号	当前值	
	执行前	执行后
(S·)	H1234	HFFFF
(S·)+1	H5678	HFFFF
(S·)+2	H90AB	HFFFF
≀	≀	≀
(S·)+(2048×n)-1	HCDEF	HFFFF

注意要点

初始化时每段需要约25ms。
对多个段执行初始化时，请务必在编程中采取下述的任意一个对策。

- 在下面的程序中将看门狗定时器的设定值D8000放大。



看门狗定时器的设定值基准

看门狗定时器的设定值可以按照下面的步骤将得到的值作为基准值。
但是，当这个基准值在200以下时，无需更改看门狗定时器的设定值。

1) 使用HCP-works将要运行的程序写入到可编程控制器中。
[在线]→[PLC写入]

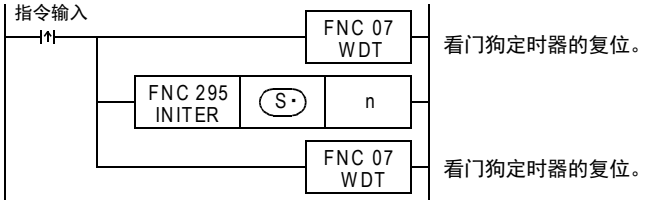
2) 使用HCP-works的软元件测试功能将D8000(单位: ms)的当前值设定为1000。
[在线]→[调试]→[软元件测试]的「字软元件/缓冲存储区的项目」

3) 用可编程控制器置为RUN，使程序运行(也使该指令动作)。

4) 用HCP-works的软元件成批监控功能，监控最大的扫描时间D8012。(单位: 0.1ms)

5) 请将看门狗定时器最大扫描时间设置为D8012以上。
D8012中的最大的扫描时间以0.1ms为单位保存。
因此，将D8012除以10后再加上50～100，请将所得到的值作为看门狗定时器的设定值D8000(单位: ms)的目标值。

- 在INITER指令的正前方和正后方，如下所示地编写WDT(FNC 07)指令。



INITER的处理时间超过200ms时，请使D8000(单位: ms)的值在INITER的处理时间以上。

2. 有关存储器的允许写入次数

访问扩展文件寄存器时，请注意以下一些要点

- 存储器盒(闪存)的允许写入次数在1万次以下。
  每执行一次INITR(FNC 292)、RWER(FNC 294)、INITER(FNC 295)指令，就会被计入存储器的写入次数。
  存储器的写入次数请勿超出允许写入次数。
  此外，使用连续执行型的指令，则每个可编程控制器的运算周期中都会执行对存储器的写入。如要避免这种情况，必须使用脉冲执行型指令。
- 即使执行LOADR(FNC 290)、SAVER(FNC 291)、LOGR(FNC 293)指令，存储器的写入次数也不会被记录。但是，执行SAVER(FNC 291)、LOGR(FNC 293)指令前，需要对写入对象的段进行初始化。使用INITR(FNC 292)、INITER(FNC 295)指令初始化时，因为每执行一次INITR(FNC 292)、INITER(FNC 295)指令，就会被计入存储器的写入次数，因此请注意存储器的写入次数。

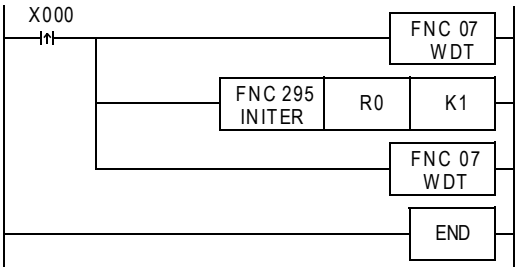
错误

下面的情况下会发生运算错误，错误标志位(M8067)为ON，错误代码保存在D8067中。

- 当(S・)设定了扩展文件寄存器(ER)的段的起始软元件编号以外的数字时。
  (错误代码: K6706)
- 要初始化的软元件的编号超出了32767时，则初始化到ER32767为止。
  (错误代码: K6706)
- 存储器盒的写保护开关设置在ON时。(错误代码: K6770)
- 未连接存储器盒时。(错误代码: K6771)

程序举例

对段0中的扩展文件寄存器ER0～ER2047进行初始化的程序。



- 扩展文件寄存器(ER)[存储器盒内]

软元件编号	当前值	
	执行前	执行后
ERO	H1234	HFFFF
ER1	H5678	HFFFF
ER2	H90AB	HFFFF
≀	≀	≀
ER2047	HCDEF	HFFFF

34. SFC程序和步进梯形图

在本章中，说明了使用HCP-works进行"SFC"和“步进梯形图”编程的要领以及顺控动作。

34.1 SFC程序

34.1.1 概要

在HC系列可编程控制器中，可以使用SFC图(SequentialFunctionChart: 顺序功能图)实现顺控。
用SFC程序可以以便于理解的方式表现基于机械动作的各工序的作用和整个控制流程。所以，顺控的设计也变得简单。
因此，即使对第三方人员也能轻易传达机械的动作，所以能够编制出便于维护以及应对规格变更、和故障发生的更加有效的程序。
此外，SFC程序和步进梯形图指令，都是按照既定的规则进行编程的，所以可以相互转换。
因此，实质内容完全一致，也可以作为熟悉的继电器梯形图使用。

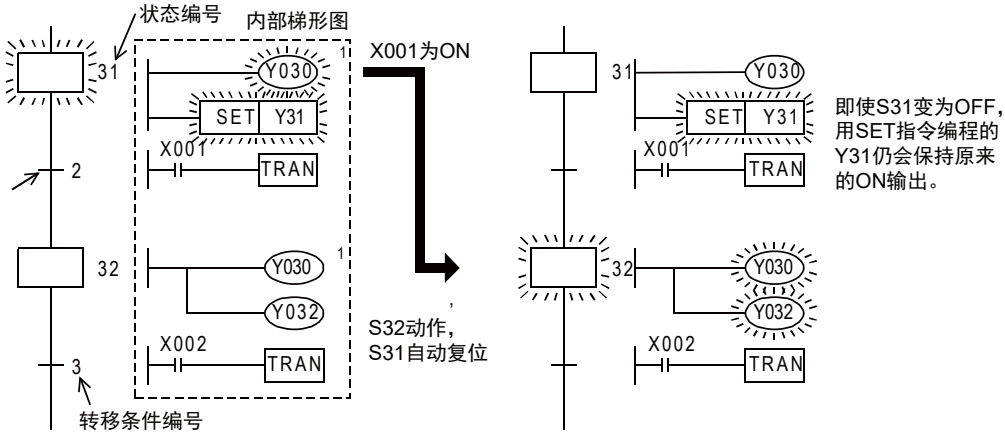
34.1.2 功能和动作说明

在SFC程序中，将状态S视作1个控制工序，在其中对输入条件和输出控制的顺序进行编程。
由于工序推进时，前工序就转为不动作，所以可以按各工序的简单的顺序来控制机械。

状态S和驱动指令的动作

SFC程序中，用状态表示机械运行的各个程序

- 当状态为ON时，与此连接的梯形图(内部梯形图)动作。
  当状态为OFF时，与此连接的内部梯形图不动作。
  1个运算周期以后，指令的OFF执行不动作。(跳转的状态)
- 当各状态之间设置的条件(转移条件)被满足时，下一个状态变为ON，此前为ON的状态变为OFF。(转移动作)
  在状态的转移过程中，仅一瞬间(1个运算周期)两个状态会同时接通。
  转移前的状态在转移后的在下一个运算周期被OFF(复位)。
  但是，通过触点指令使用转移状态S时，从转移条件被满足之后开始，触点映像被执行OFF。
- 不能重复使用同一个状态编号。

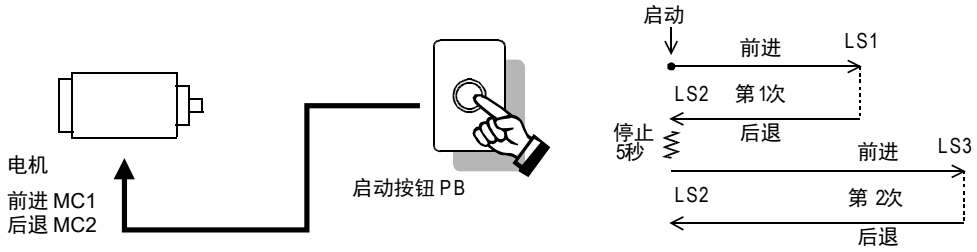


*1. 在不同的状态中，可以重复编写输出线圈。

34.1.3 SFC程序的创建步骤

按照下述的步骤创建SFC程序。

1. 动作实例



1) 按下启动按钮PB后，台车前进，限位开关LS1动作后，立即后退。
(LS1通常为OFF，只在到达前进限位处为ON。其他的限位开关也相同。)

2) 通过后退，限位开关LS2动作后，停止5秒钟以后再次前进，到限位开关LS3动作时，立即后退。

3) 此后，限位开关LS2动作时，驱动台车的电机停止。

4) 一连串的动作结束后，再次启动，则重复执行上述的动作。

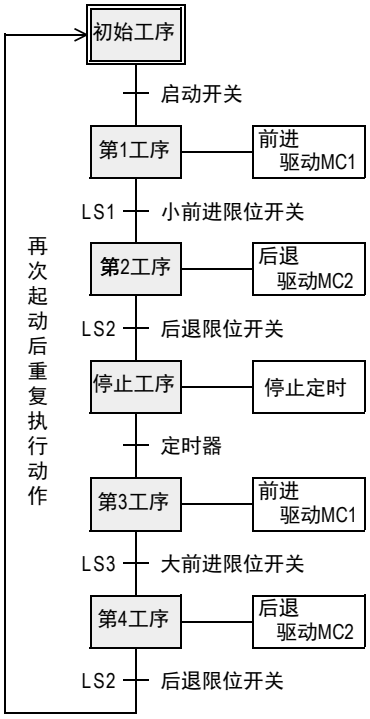
2. 工序图的创建

按照下述的步骤，创建如右图所示的工序图。

1) 将上述事例的动作分成各个工序，请按照从上至下动作的顺序用矩形表示。

2) 用纵线连接各个工序，请写入工序推进的条件。
执行重复动作的情况下，在一连串的动作结束时，请用箭头表示返回到哪个工序。

3) 请在表示工序的矩形的右边写入各个工序中执行的动作。



3. 软元件的分配

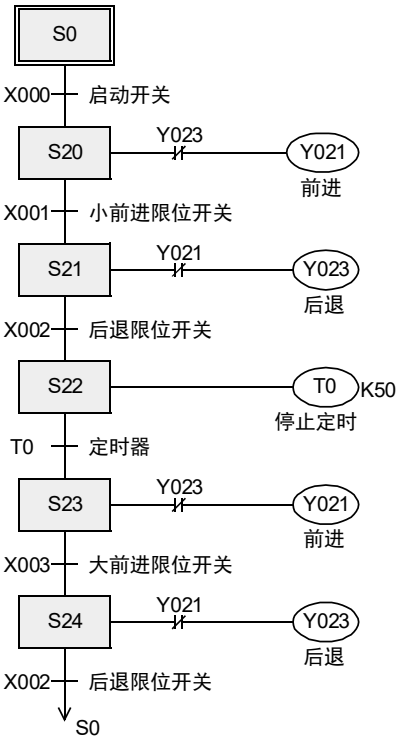
请给已经创建好的工序图分配可编程控制器的软元件。

1) 请给表示各个工序的矩形分配状态(S•)。
此时，请给初始工序中分配初始状态(S0～S9)。
第1个工序以后，请任意分配除去初始状态以外的状态编号(S20～S899等)。
(状态编号的大小与工序的顺序无关。)
在状态中，还包括即使停电也能记忆住其动作状态的停电保持用状态。
此外，S10～S19是在使用IST指令(FNC 60)时作为特殊目的使用的。

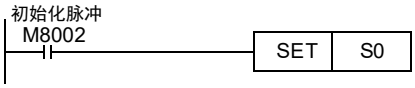
2) 请给转移条件，分配软元件(按钮开关以及限位开关连接的输入端子编号以及定时器编号)。
转移条件中可以使用a触点和b触点。
此外，有多个条件时，也可以使用AND梯形图和OR梯形图。

3) 请对各个工序执行的动作中使用的软元件(外部设备连接的输出端子编号及定时器编号)进行分配。
可编程控制器中备有多个定时器、计数器、辅助继电器等器件，可以自由地使用。
此处使用了定时器T0；这个定时器是按0.1秒时钟动作的，所以当设定值为K50时，线圈被驱动5秒后输出触点动作。
此外，有多个需要同时驱动的负载、定时器和计数器等时，也可以在1个状态中分配多个梯形图。

4) 执行重复动作以及工序的跳转时使用「　」，请指定要跳转的目标状态编号。



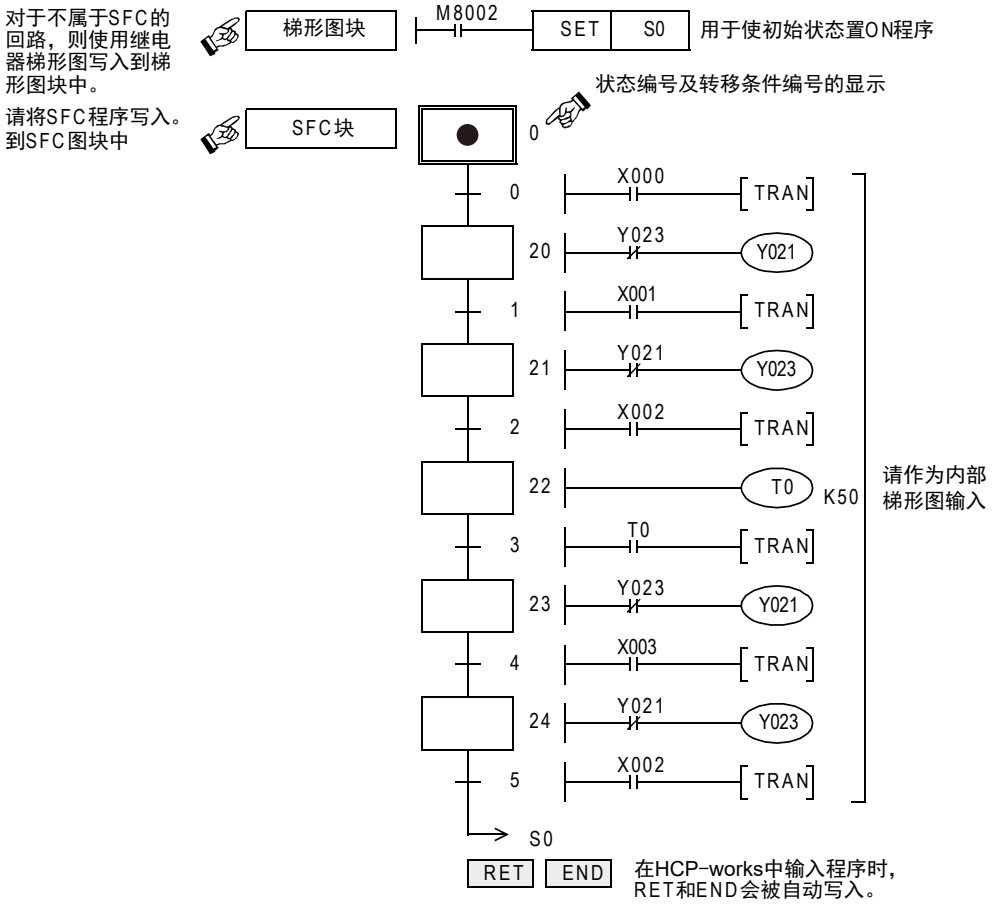
在这个例子中，仅仅说明了SFC程序的制作步骤，实际上，要使SFC的程序运行，还需要将初始状态置ON的梯形图。
请使用继电器梯形图编写使初始状态置ON的梯形图。
此时，为了使状态置ON，请使用SET指令。



4. 在HCP-works中输入及显示程序

- 请输入使初始状态置ON的继电器梯形图。
  在这个例子的梯形图块中，使用了当可编程序控制器从STOP变为RUN时，仅瞬间动作的辅助继电器M8002，使初始状态S0被置位(ON)。
- 在HCP-works中输入程序时，请把继电器梯形图的程序写入到梯形图块中，把SFC的程序写入到SFC块中。
- 表示状态内的动作的程序及转移条件，被作为状态以及转移条件的内部梯形图处理。
  请分别使用继电器梯形图编程。

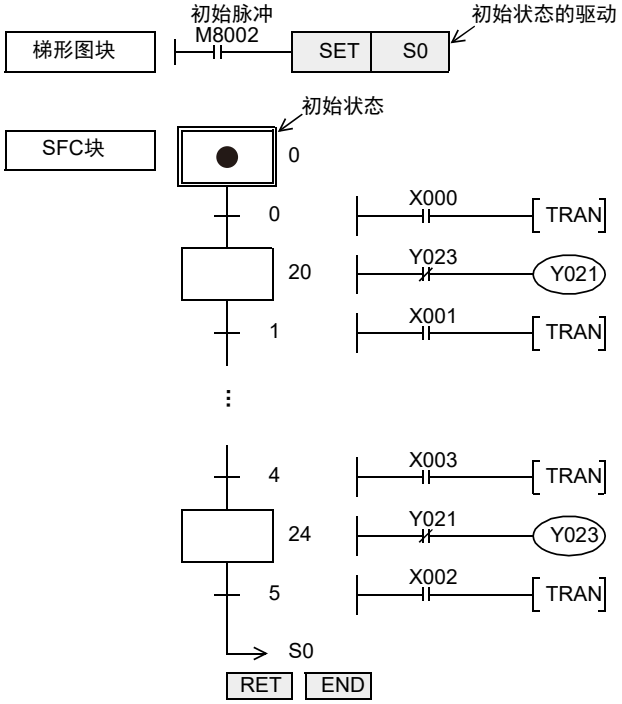
关于HCP-works的编程操作的详细内容，请参考HCP-works的操作手册。



34.1.4 初始状态的使用和作用

初始状态的使用

- 占据SFC程序的起始位置的状态称为初始状态，可以使用S0～S9的状态编号。
- 初始状态也是凭借其他的状态 (上图的例子中的 S24) 驱动的，但是在运行开始时需要预先通过其他手段来驱动。
- 如下图所示的例子中，通过使用可编程控制器从 STOP 切换到 RUN 时瞬间动作的特殊辅助继电器 M8002 进行驱动。
- 初始状态以外的一般状态，都必须通过其它状态驱动，没有被状态以外的程序驱动的情况。
- 象这样，通过STL指令以外的触点被驱动的状态称为初始状态，必须在流程的起始处作描述。



初始状态的作用

1. 作为逆转换所需的识别软元件
   - 从指令表逆转换到SFC程序时，需要识别流程的起始位置。
     因此，作为初始状态请使用S0～S9。
     使用除此以外的编号时不能执行逆转换。
   - 针对初始状态的STL指令，要比针对紧接其后的一连串状态的STL指令先编程，最后编写RET指令。
     由此，有单独的多个流程时，执行流程的相互分离。
2. 防止产生双重启动
   - 在上述的例子中，例如状态S24动作时，即使再次按下启动按钮，也无效。(由于S0不动作)
     因此，可以防止产生双重启动。

34.1.5 停电保持(保持用)状态

停电保持用状态是使用电池或EEPROM存储器对其动作状态进行备份。
在机械动作过程中发生停电后，再次通电时想要从刚才的状态继续运行时，可以使用这些状态。

34.1.6 RET指令的作用

- SFC程序中，在SFC程序的最后使用RET指令。
  但是，通过HCP-works输入SFC程序时，不需要输入RET指令。
  (会自动写入。)
- 在可编程控制器中，从0步开始到END指令之间可以制作多个SFC块。
  当梯形图块和SFC块混在一起时，分别在各个SFC程序的最后编写RET指令。

34.1.7 编制SFC程序所需的预备知识

状态内可以处理的顺控指令一览

状态		指令		
		LD/LDI/LDP/LDF,AND/ANI/ANDP/ANDF,OR/ORI/ORP/ORF,INV,MEP/MEF,OUT,SET/RST,PLS/PLF	ANB/ORB/MPS/MRD/MPP	MC/MCR
初始状态/一般状态		可以使用	可以使用*1	不可以使用
分支、汇合状态	驱动处理	可以使用	可以使用*1	不可以使用
	转移处理	可以使用	不可以使用	不可以使用

- 中断程序和子程序中不可以使用STL指令。
- 使用SFC程序(STL指令)时，在中断程序中请勿使用SET指令或OUT指令驱动状态S。
- 并非禁止在状态中使用跳转指令，而是由于使用了会产生复杂的动作，所以建议尽量不使用。

*1. 即使是驱动处理梯形图，也不能在状态(STL指令)的后面使用MPS指令。

特殊辅助继电器

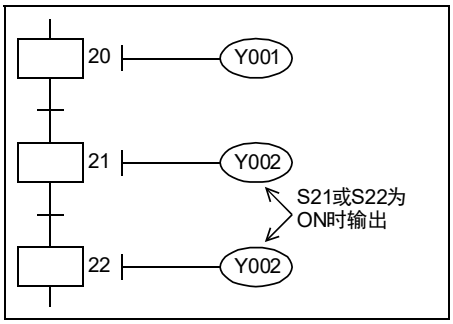
为了能够更有效地制作SFC程序，需要使用几个特殊辅助继电器，主要的内容如下表所示。

软元件编号	名称	功能及用途
M8000	RUN监控	在可编程控制器运行过程中一直为ON的继电器。 可以作为需要一直驱动的程序的输入条件以及作为可编程控制器的运行状态的显示来使用。
M8002	初始脉冲	仅仅在可编程控制器从STOP切换成RUN的瞬间(1个运算周期)为ON的继电器。 用于程序的初始设定和初始状态的置位。
M8040	禁止转移	驱动了这个继电器后，所有的状态之间都禁止转移。 此外，即使是在禁止转移的状态下，由于状态内的程序仍然动作，所以输出线圈等不会自动断开。
M8046*2	STL动作	即使S0～S899、S1000～S4095只有1个状态为ON时，M8046就会自动置ON。用于避免与其他流程同时起动，或者用作为工序的动作标志位。
M8047*2	STL监控有效	驱动了这个继电器后，将状态S0～S899，S1000～S4095中正在动作(ON)的状态的最新编号保存到D8040中，将下一个动作(ON)的状态编号保存到D8041中。以下到D8047为止依次保存动作状态(最大8点)。 • 在HCP-works的SFC监控中，即使不驱动这个继电器，也可以实现自动滚动监控。

*2. 在执行END指令时处理

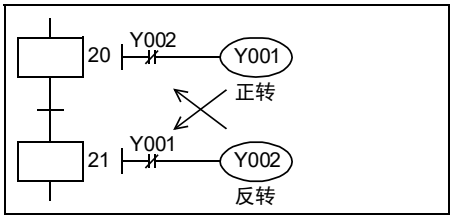
状态的动作和输出的重复使用

- 在不同状态之间，如右图所示，可以对相同的输出(Y002)进行编程。
  此时，当S21或是S22为ON时输出Y002。
  但是，在梯形图块的程序中编写了与状态中的输出线圈相同的软元件(Y002)，同时在1个状态内编写相同的输出线圈时，会执行与一般的双重线圈相同的处理，请注意。



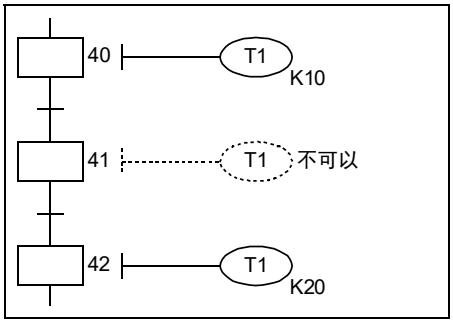
输出的互锁

- 在状态转移过程中，只有一瞬间(1个运算周期)两个状态会同时为ON。
  因此，在不可以同时接通的一对输出之间，为了避免同时为ON，请参考各可编程控制器的[使用手册]，在可编程控制器的外部设置互锁。
  此外，请同时在程序中执行如右图所示的相互的互锁。



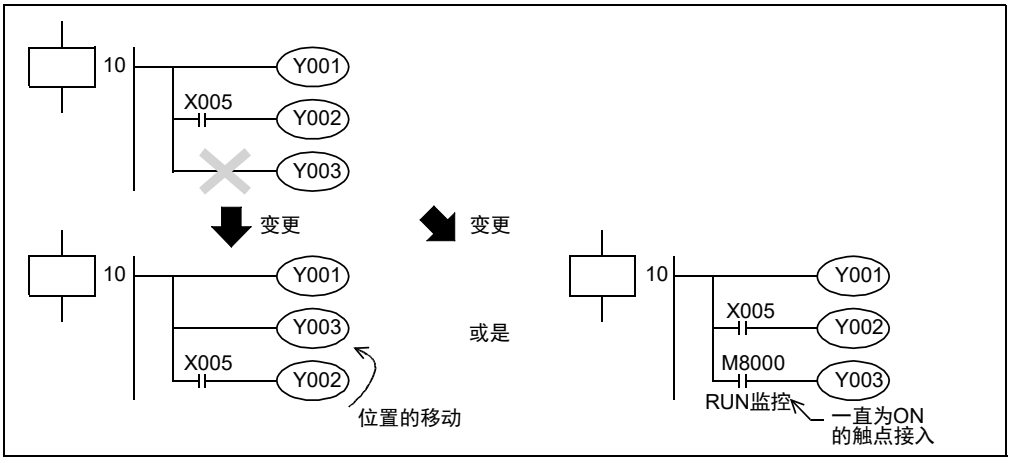
定时器的重复使用

- 定时器线圈也与输出线圈相同，可以在不同的状态中对同一个软元件进行编程，但是在相邻的状态中不能编程。如果在相邻状态中编程，则工序转移时，定时器线圈不断开，当前值不会被复位。



输出的驱动方法

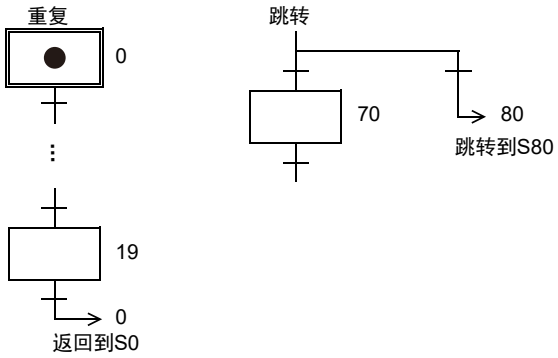
- 如下图所示，从状态内的母线开始，一旦写入LD或是LDI指令后，就不能再编写不需要触点的指令。
  请按照下图所示的梯形图进行更改。



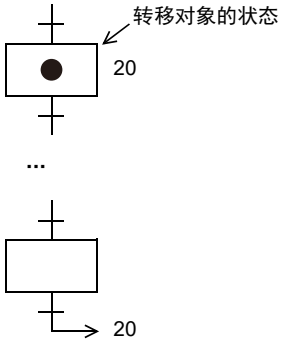
和▽的动作

转移(重复)到上方状态或转移(跳转)到下方状态，转移到被分离的其他流程的状态用[　]表示。
用[▽]表示对状态的复位处理。

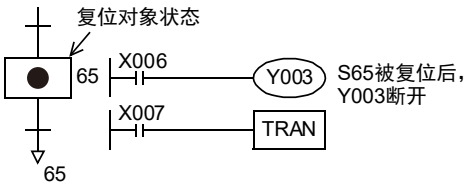
1) 转移源的程序



2) 转移目标的程序
在HCP-works中，作为移动对象的状态中会自动显示[●]。



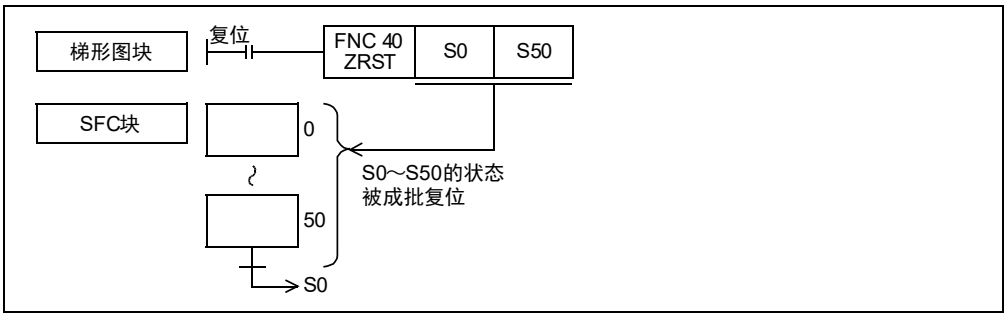
3) 复位回路图的程序
下图的例子中，通过S65中的X007对S65进行复位。
从S65对其他状态(例如S70)进行复位时也相同，但是这并非是转移动作，所以S65不被复位。
在HCP-works中，作为复位对象的状态中会自动显示[●]。



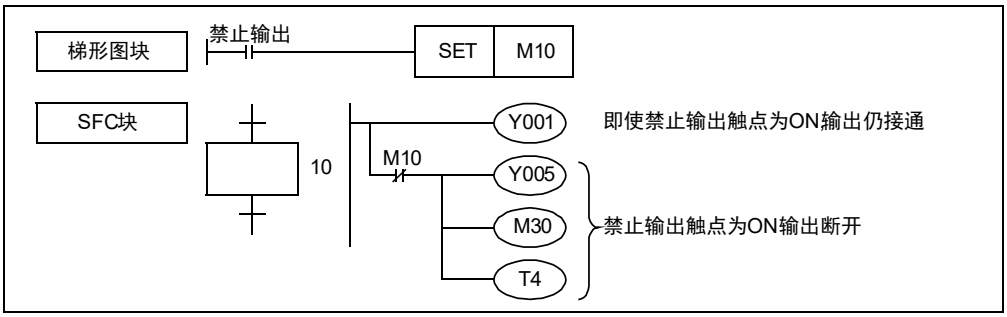
状态的成批复位和禁止输出

关于相当于紧急停止的输出禁止，请参考可编程控制器手册中记载的[安全上的注意事项]。

1) 指定状态的范围后复位
成批复位S0～S50的51点。



2) 禁止动作中的状态的任意输出

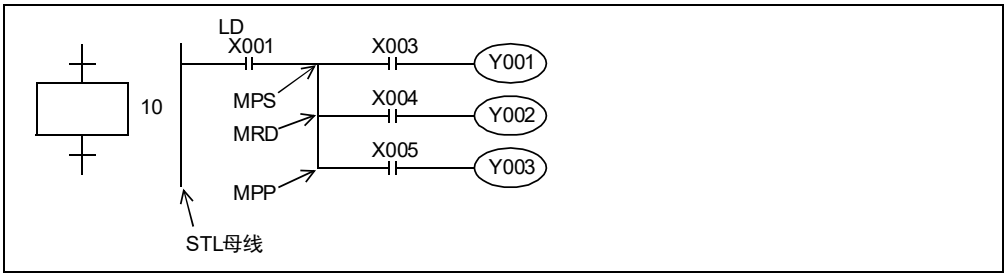


3) 断开可编程控制器的所有输出继电器(Y)
使特殊辅助继电器M8034为ON期间，顺控程序的运算继续，但是输出继电器(Y)全部变为OFF。(监控时仍然是为ON的。)



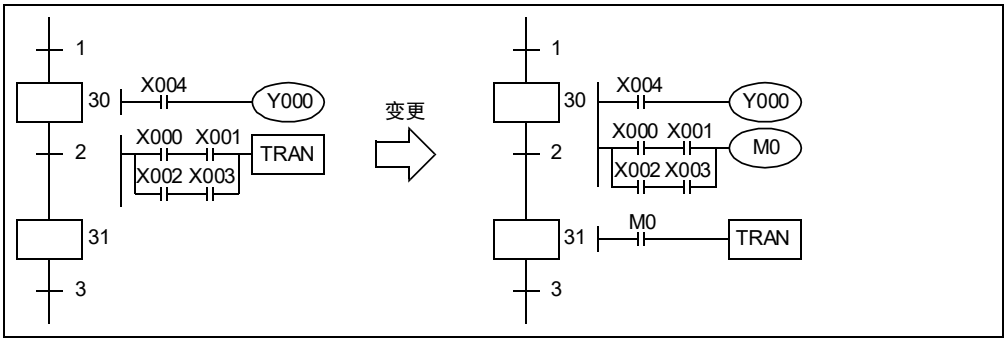
MPS/MRD/MPP指令的位置

在状态内，不能从STL的母线开始直接使用MPS/MRD/MPP指令。
如右图所示，请在LD或是LDI指令以后编程。



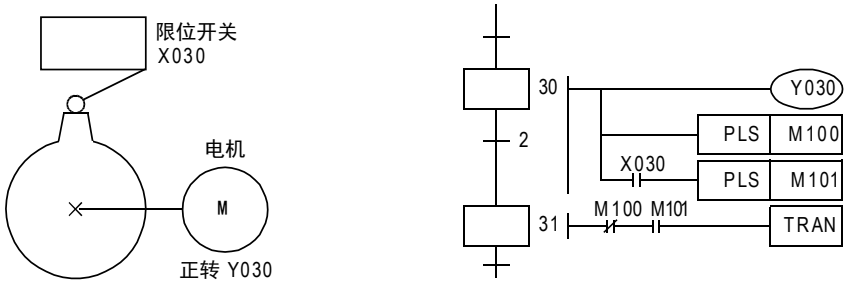
复杂的转移条件的程序

在转移条件的梯形图中，不能使用ANB,ORB,MPS,MRD,MPP 指令。
请按照下面的要领进行编程。



转移条件已成立的状态的处理

作为转移条件的限位开关X030已经动作，希望其再次动作(OFF→ON)后进行下一个转移。
此时，如下图所示，将转移条件脉冲化，S30首次动作时，通过M100使其不产生转移。

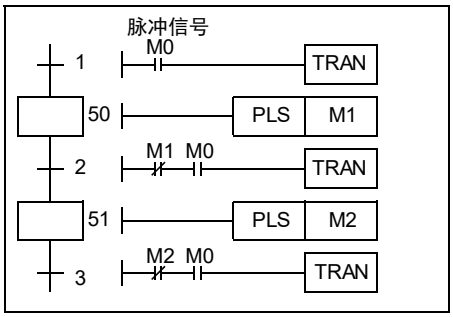


使用同一信号的状态转移

有时候，希望通过1个按键开关的ON/OFF动作等就可以进行状态转移。
为了执行这种状态转移，必须要将信号脉冲化后进行编程。
转移条件的脉冲化有以下的2种方法。

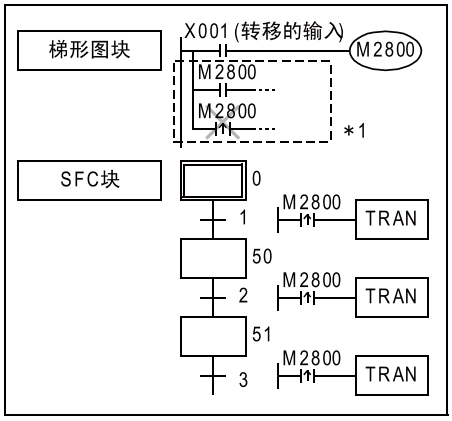
1. 使用PLS指令的步骤

当M0接通S50动作之后，转移条件M1(b触点)立即开路，在S50接通的同时，不向S51转移。当M0再次接通时，转移到S51。



2. 使用脉冲触点指令(M2800～M3017)的步骤

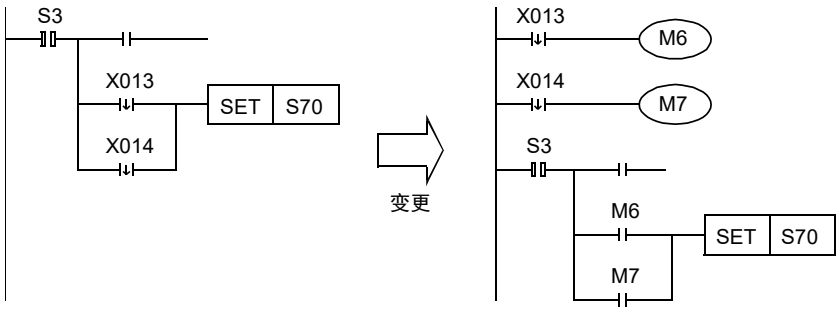
在检测上升沿，下降沿的指令(LDP, LDF, ANDP, ANDF, ORP, ORF)中，使用辅助继电器M2800～M3071可以有效地使用同一信号转移。
在上升沿、下降沿检测指令的软元件中，指定了M2800开始的软元件时，只有线圈指令以后的最初的上升沿或是下降沿检测指令被执行。因此，当X001为ON时，只有当前动作中的状态内的转移条件有1个运算周期为ON，并向下一个状态转移。



*1. 在梯形图块中的相应线圈之后，用LD,LDI,OR,ORI指令对同一编号编程不会出现问题，但是如果用LDP, LDF,ANDP,ANDF,ORP,ORF指令对同一编号编程，则这些指令被优先执行，而转移条件不动作。

使用上升沿，下降沿检测触点时的注意事项

在状态内使用了LDP,LDF,ANDP,ANDF,ORP,ORF指令的上升沿，或是下降沿检测触点时，在状态断开过程中发生变化的触点，当状态再次接通时会被检测出来。
对于状态断开时发生变化的条件，需要检测其上升沿和下降沿时，请按如下所示修改程序。



通过X013的下降沿向S70转移后，如X014为下降沿，则此时因S3已为OFF，所以无法检测X014的下降沿，当S3再次为ON时，被检测。
因此，S3第2次动作时，立即向S70转移。

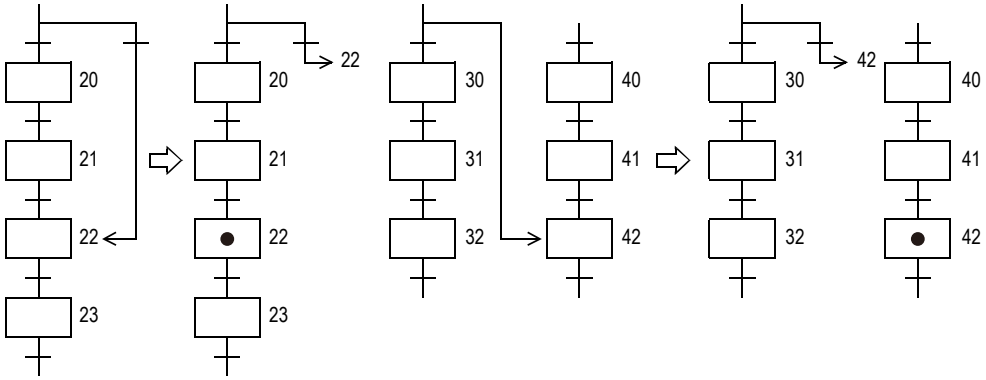
34.1.8 SFC流程的形态

表示SFC程序的单流程动作模式，和选择分支及并行分支组合使用时的动作模型。

1. 跳转・重复操作

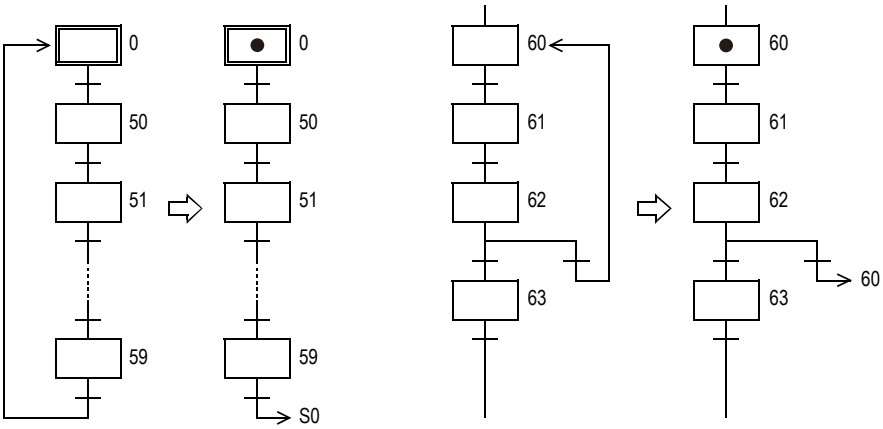
1) 跳转

直接转移到下方的状态以及转移到流程外的状态，称为跳转。用[ ]表示要转移的目标状态。



2) 重复

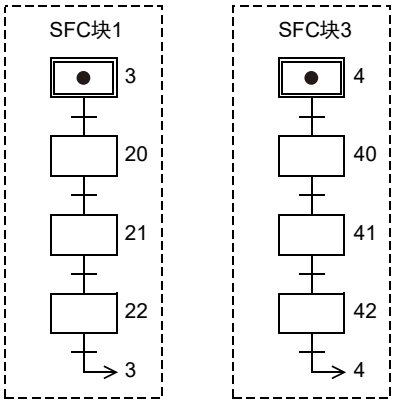
转移到上方的状态称为重复，同样地使用[]表示要转移的目标状态。



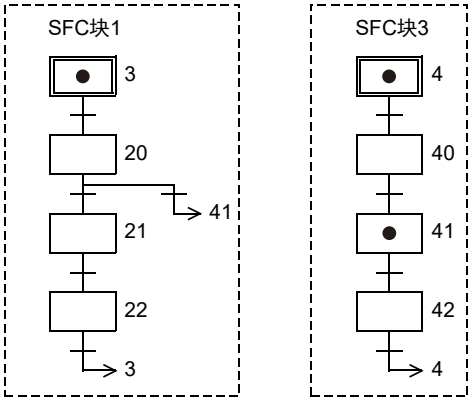
流程的分离

具有多个初始状态的SFC的程序时，将各初始状态分成程序块后编程。
在分离程序块后制作的SFC程序之间也可以转移。(跳转到流程外)
此外，在不同的块中制作的程序的状态，可以将其用作为状态的内部梯形图和转移条件的触点。

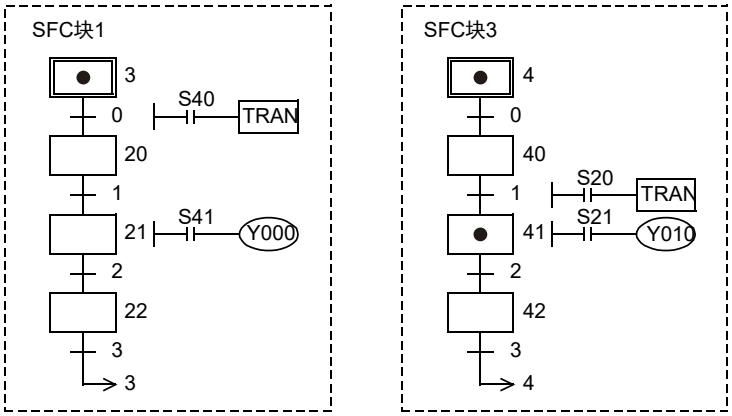
1. 流程的分离



2. 跳转到流程外



3. 使用不同的块中制作的程序的状态

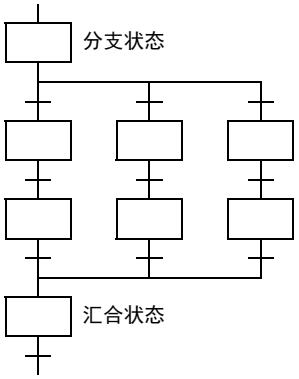


分支·汇合的复合流程

工序转移的基本类型为单流程形式的控制。对单纯动作的顺控，只需单流程就足够了，但是当介入各种各样输入条件和操作者操作时，可以通过组合使用选择分支和并行分支流程，简单地处理复杂的条件。
根据条件对多个工序执行选择处理用的分支称为选择分支，同时处理多个工序用的分支称为并行分支。

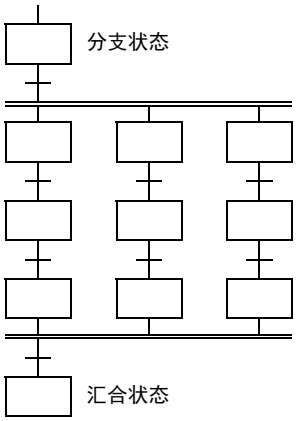
1. 选择分支

从多个流程中选择执行其中的一个流程。



2. 并行分支

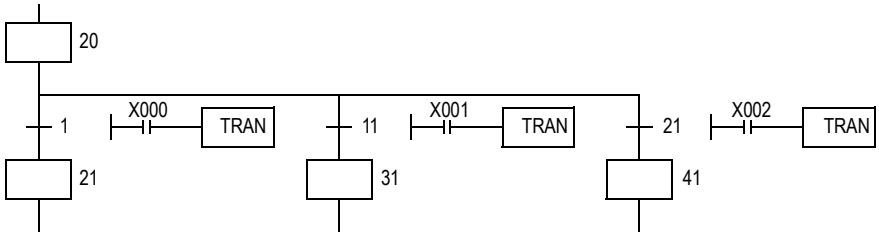
同时进行所有的多个流程



34.1.9 分支·合并状态的程序

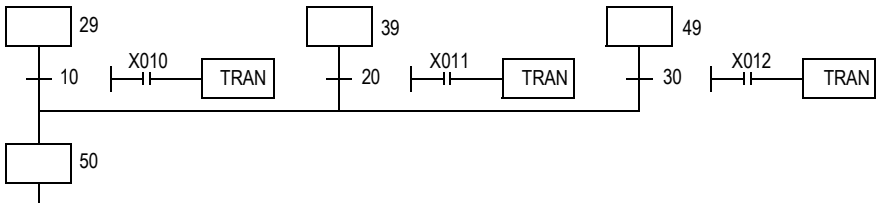
选择分支

请在分支后，编写转移条件。



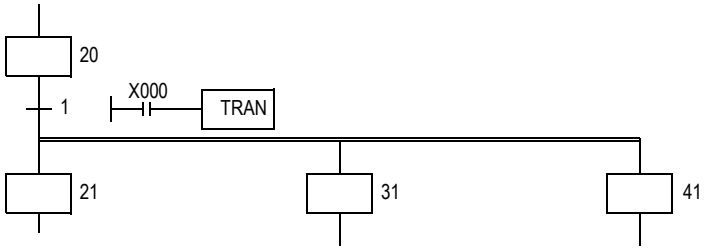
选择汇合

在编写转移条件后进行汇合。



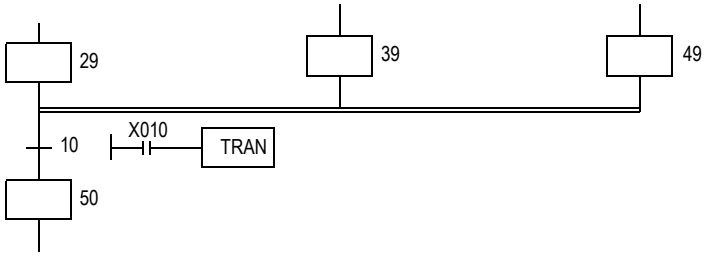
并行分支

请在编写转移条件后再分支。



并行汇合的例子

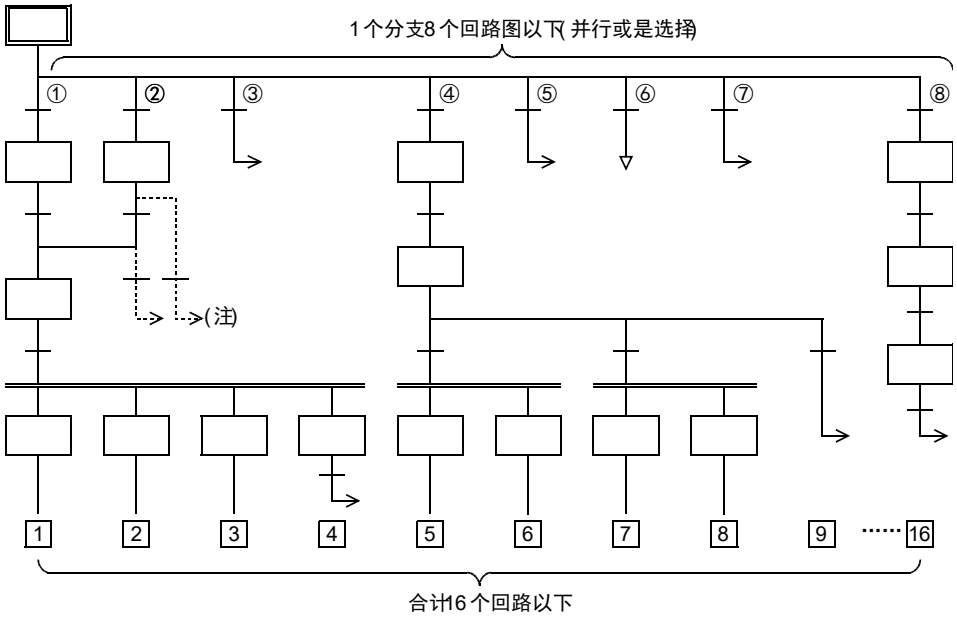
请在汇合后，再编写转移条件。



34.1.10 制作分支回路方面的规则

分支回路的限制

关于一个并行分支或是选择分支，每个分支中限制在8个回路以下。
但是，有多个并行分支和选择分支时，从整体而言，回路数被限定为每一个初始状态中16个回路以下。

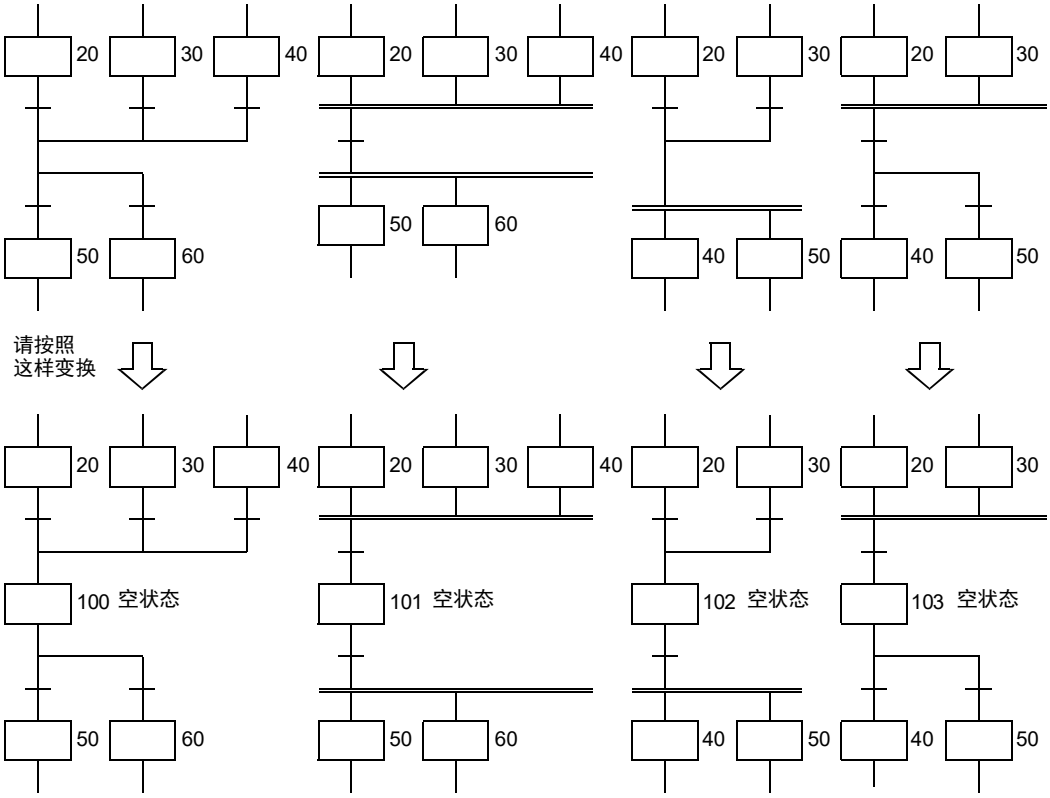


不能执行从汇合线或汇合前的状态向分支状态转移的处理，以及复位处理。
请设置空状态，必须从分支线上向分离状态进行转移，以及复位的处理。

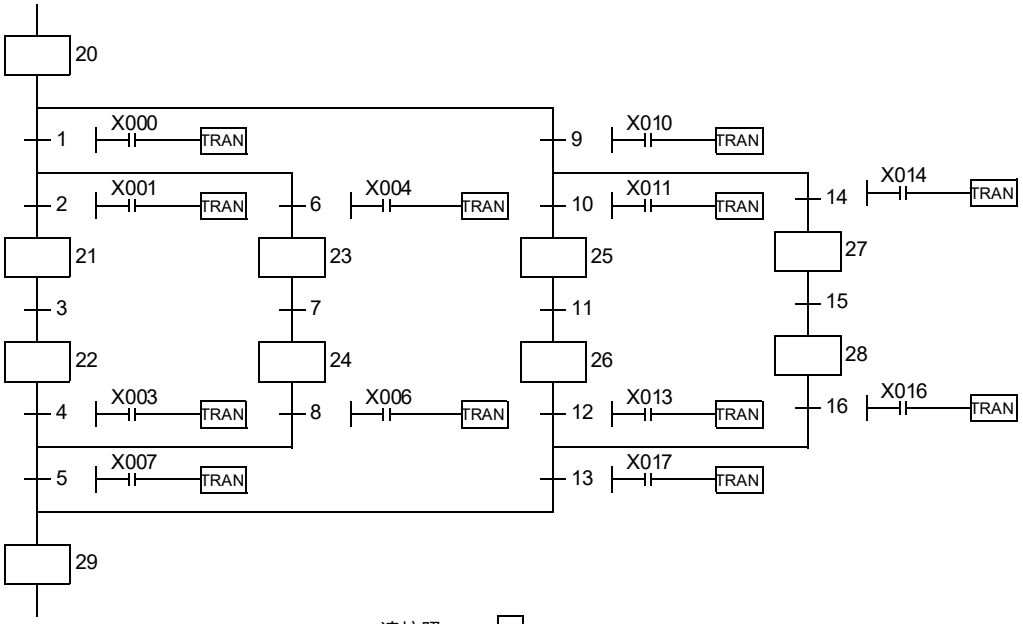
分支·汇合的合成及空状态

1. 汇合线与分支线直接连接，中间没有状态

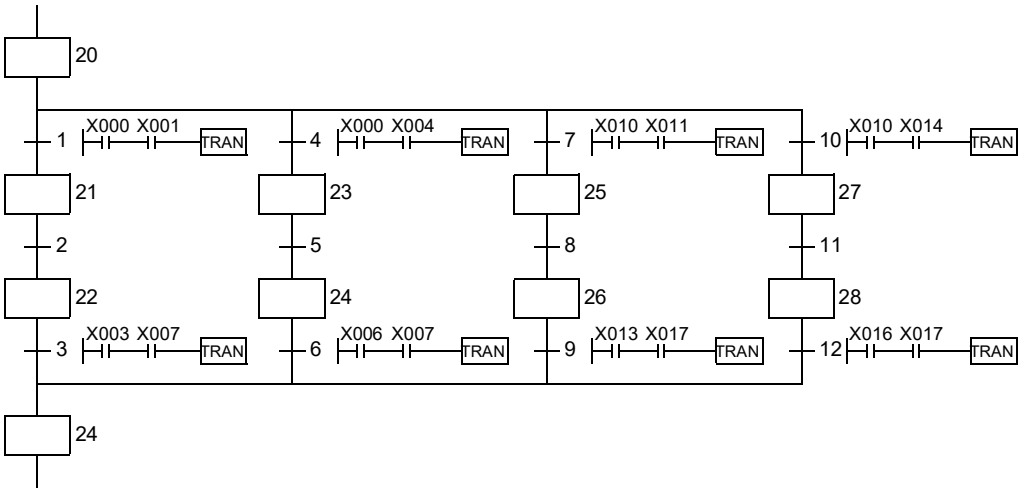
如下图所示，汇合线与分支线直接连接，中间没有状态，类似这样的情况时，建议在中间使用空状态。
状态中没有空状态专用的编号。
在程序中，请将未使用的状态编号作为空状态使用。



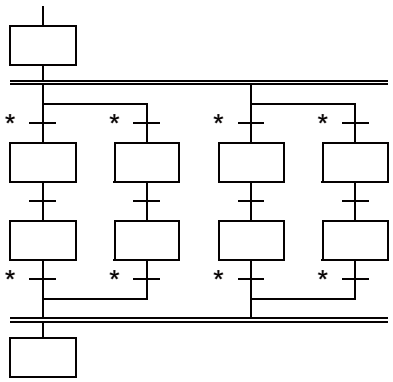
2. 请将连续的选择分支修改成分支次数较少的回路。



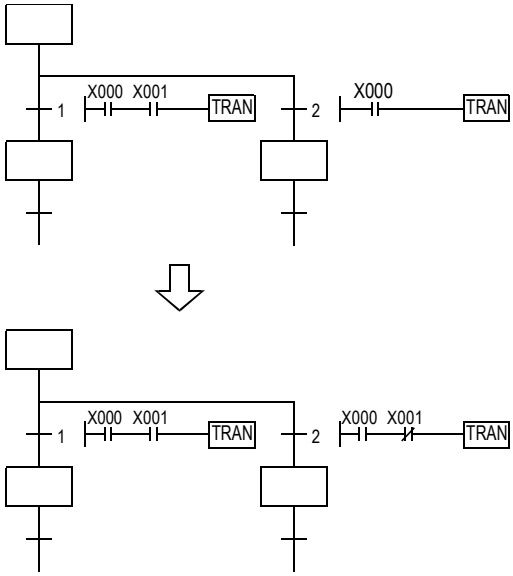
请按照
这样变换



3. 并行分支后有选择转移条件*，转移条件*后的并行汇合，不能被执行。

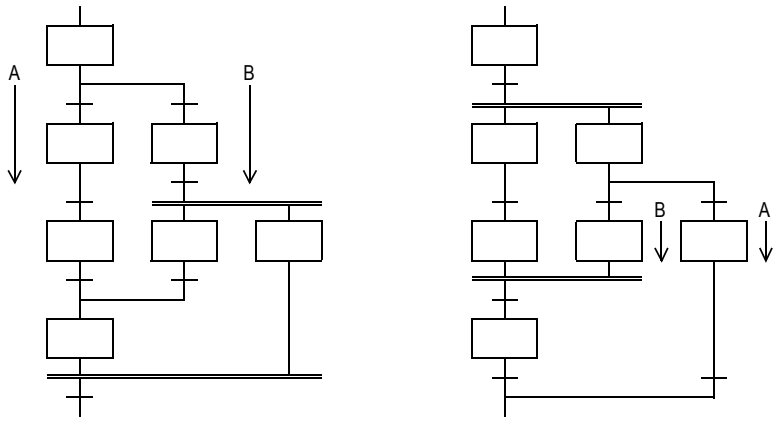


4. 象下面这样的流程不能确定是选择还是并行。请修改下面的程序。



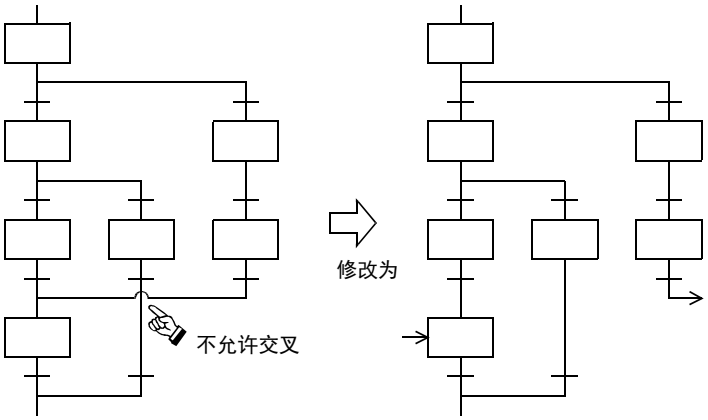
5. 如下图所示的流程都是可编程的。

B的流程没有问题，但是在A流程的情况下，并行汇合处有等待动作的状态出现，请务必注意。



6. 不能画出流程交叉的SFC程序。

如左图所示的流程要按照右侧所示的流程进行修改。根据这个可以实现以指令为基础的程序向SFC程序的逆转换。(左侧所示不能编写SFC程序。)



34.1.11 程序举例

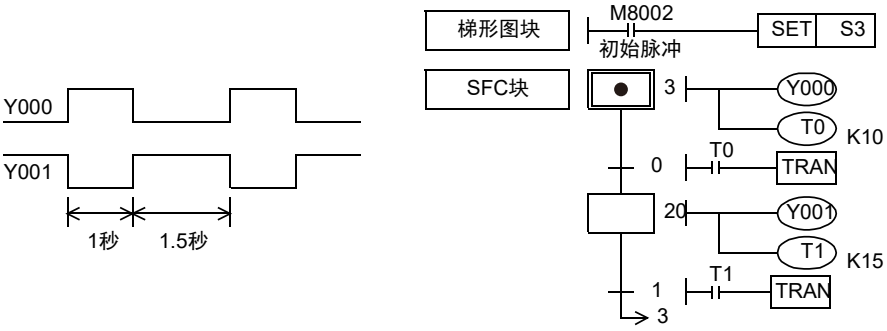
单流程的例子

1. 闪烁回路的例子

1) 使可编程控制器运行，通过初始脉冲(M8002)驱动状态S3。

2) 在S3中输出Y000，1秒钟以后转移到状态S20。

3) 在状态S20中输出Y001，1.5秒钟以后返回状态S3。

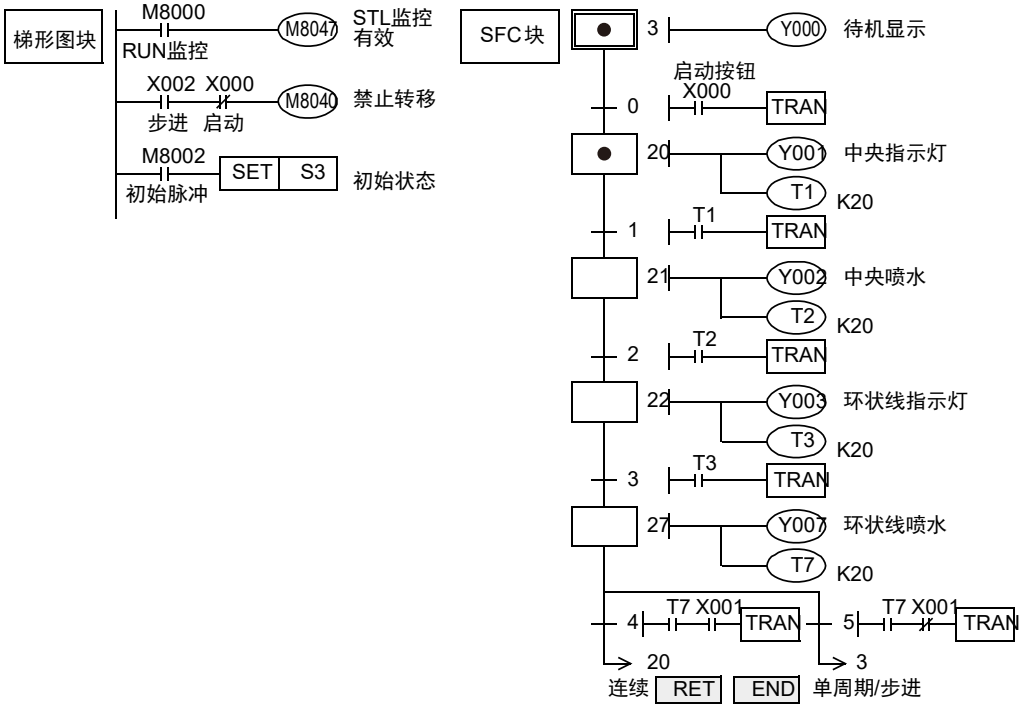


2. 喷水控制的例子

1) 单周期运行(X001=OFF,X002=OFF)
按下启动按钮X000后，按照Y000(待机显示)→Y001(中央指示灯)→Y002(中央喷水)→Y003(环状线指示灯)→Y007(环状线喷水)→Y000(待机显示)的顺序动作，然后返回待机状态。
通过预置的2秒定时器依次切换各输出。

2) 连续运行(X001=ON)
重复Y001～Y007的动作。

3) 步进运行(X002=ON)
每按一次启动按钮，各输出依次动作一次。



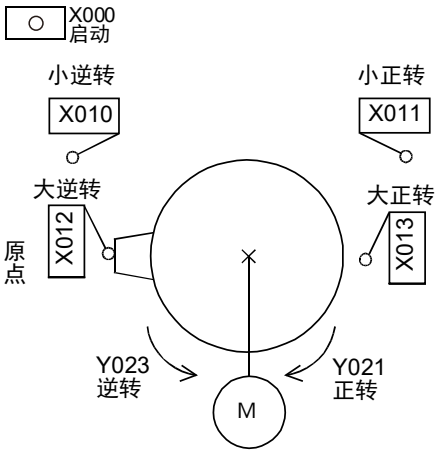
3. 凸轮轴旋转控制的例子

在正转角度的大、小 2 个位置处设有限位开关 X013、X011。

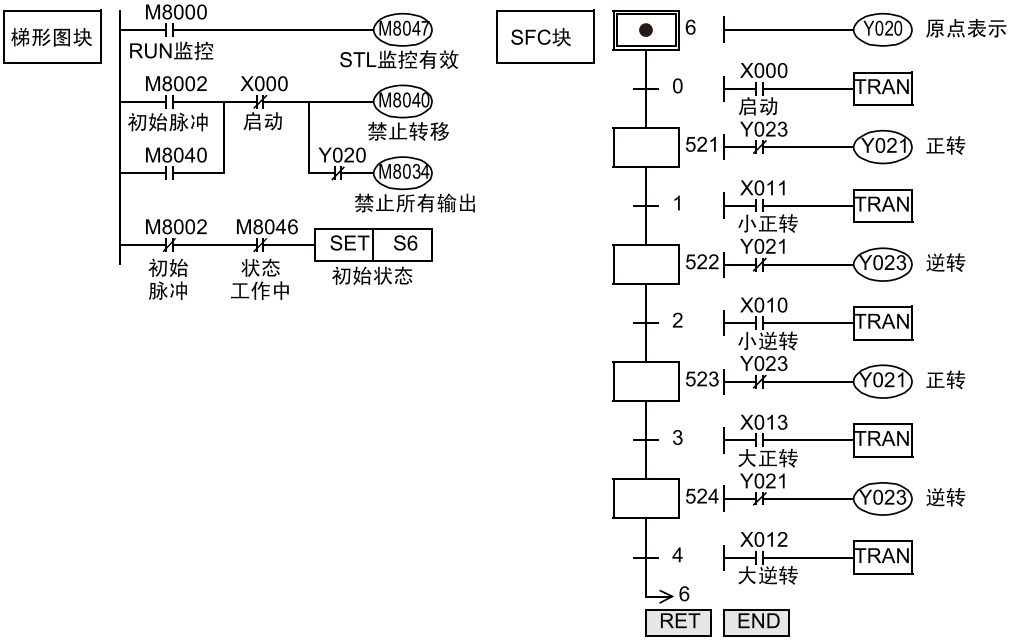
此外，在逆转角度的大、小2个位置处设有限位开关X012、X010。

按下启动按钮后，执行小正转→小逆转→大正转→大逆转的动作后停止。

限位开关X010～X013通常为OFF状态，凸轮轴达到所设定的角度时变为ON。



- 若M8047动作，则动作状态监控有效，与此同时S0～S899和S1000～S4095中只要有一个动作，
  执行END指令后M8046就动作。
- 在这个 SFC 程序中，使用了后备型的状态，即使运行过程中发生停电，只要按下启动按钮，会从这个工序开始重新执行动作。但是在按下启动按钮之前，除Y020以外的所有输出都被禁止。



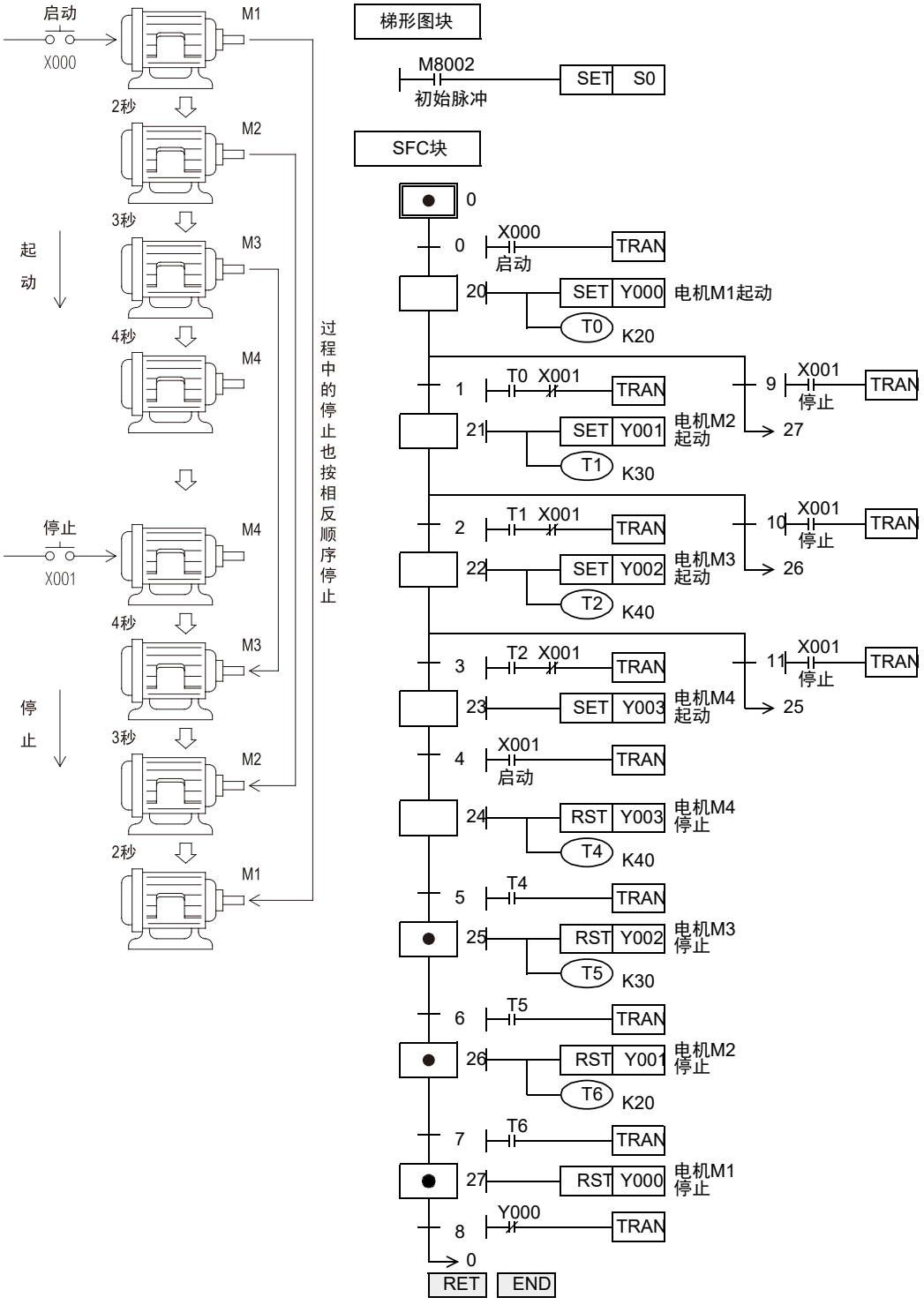
《禁止所有输出M8034》

如驱动M8034，则可编程控制器仍然运行，执行各程序，但是对外部的输出全部都变为OFF。

4. 顺序起动・停止的例子

由定时器控制电机M1～M4，按顺序起动，以相反顺序停止。

这种SFC流程是以单流程为基础执行状态的跳转。

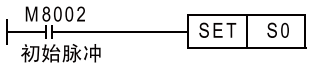


在这个SFC程序例子中，显示了根据条件跳转一部分的流程，而转移到下方的状态的例子。

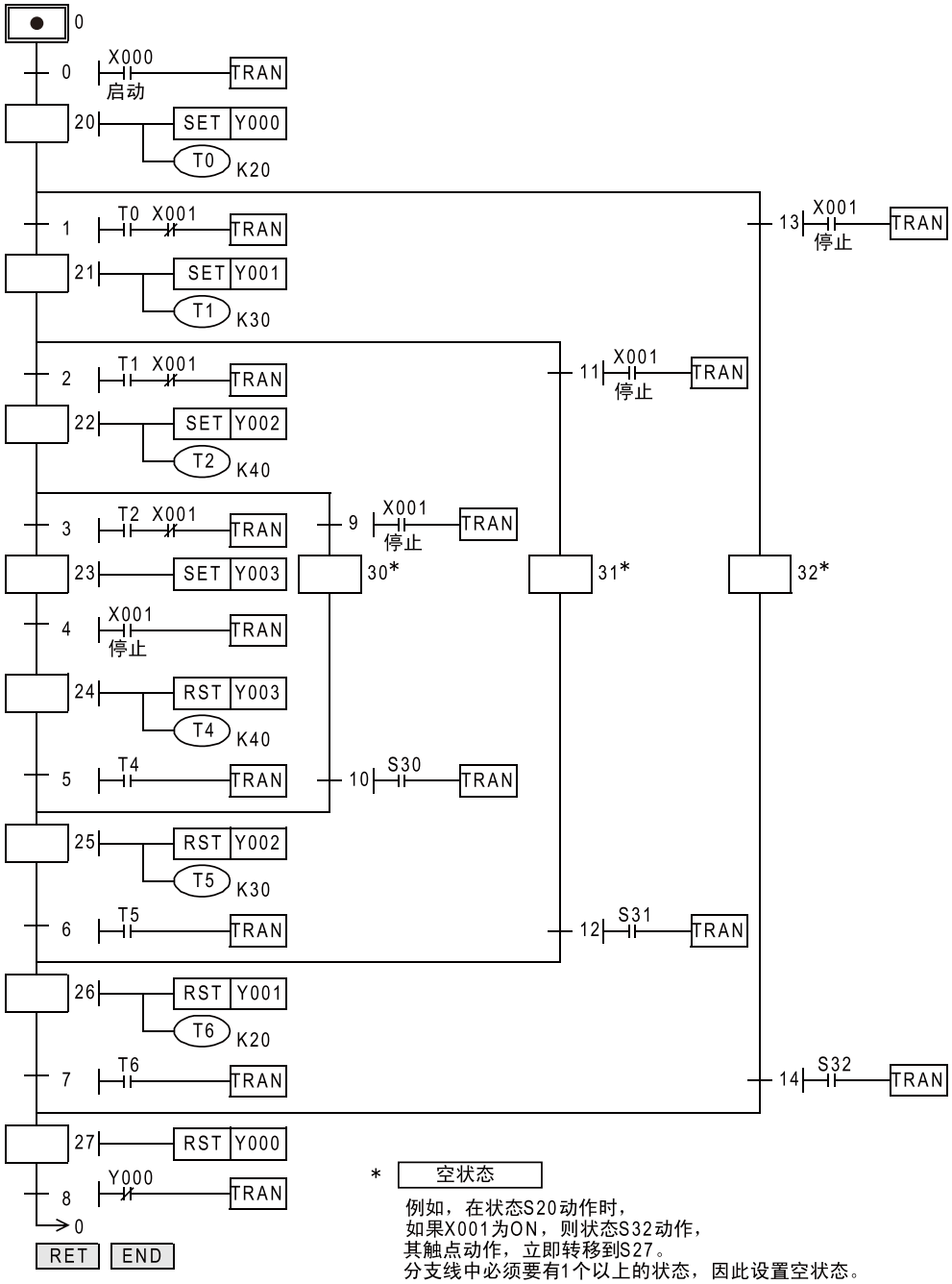
也可以跳转到上方。

上一页的部分跳转的流程，也可以用如下所示的选择分支汇合的流程来表示。
流程必须从上到下，除了分支、汇合线以外都不允许交叉。

梯形图块



SFC块



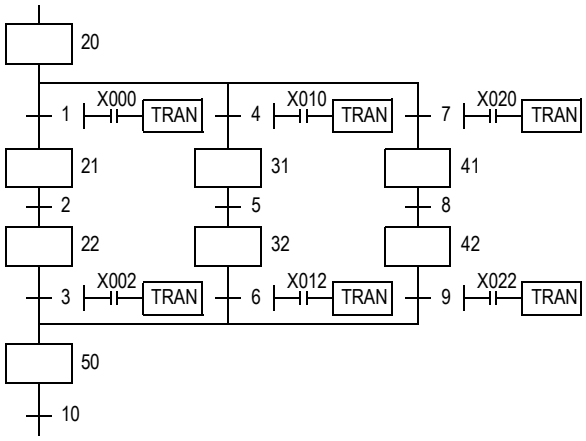
* 空状态

例如，在状态S20动作时，
如果X001为ON，则状态S32动作，
其触点动作，立即转移到S27。
分支线中必须要有1个以上的状态，因此设置空状态。

选择分支·汇合流程的例子

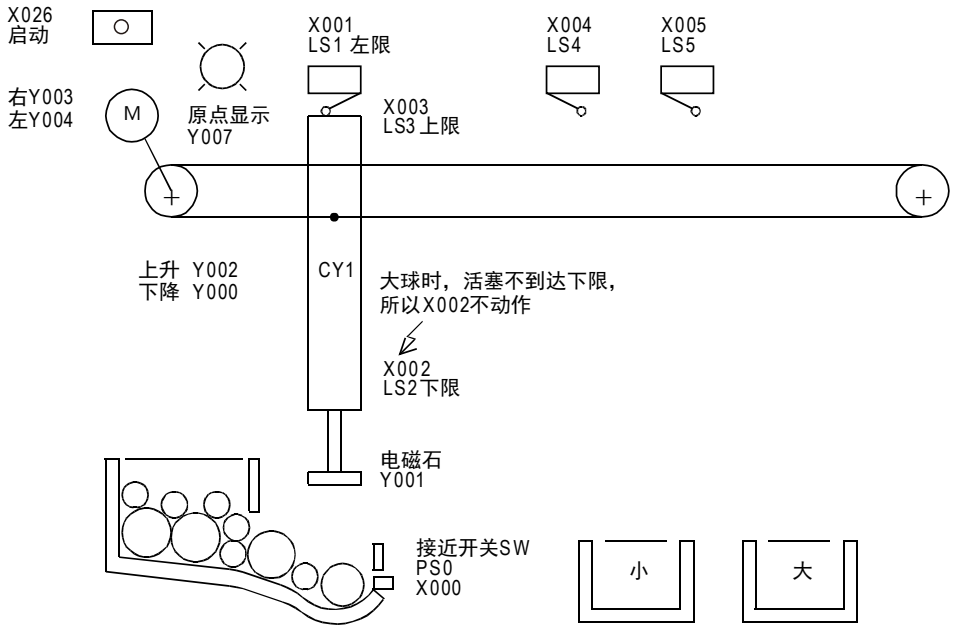
1. 选择性分支的动作

- 选择执行多个流程中的一个流程就称为选择性分支。
- 如右图的例子，必须是 X000,X010,X020 不同时为ON。
- 例如，S20 动作时，X000 为 ON，则动作状态转移到S21，S20变为不动作。
  因此，即使此后X010 和X020 动作，S31 和S41也不动作。
- 汇合状态S50是由S22、S32、S42中的一个驱动的。

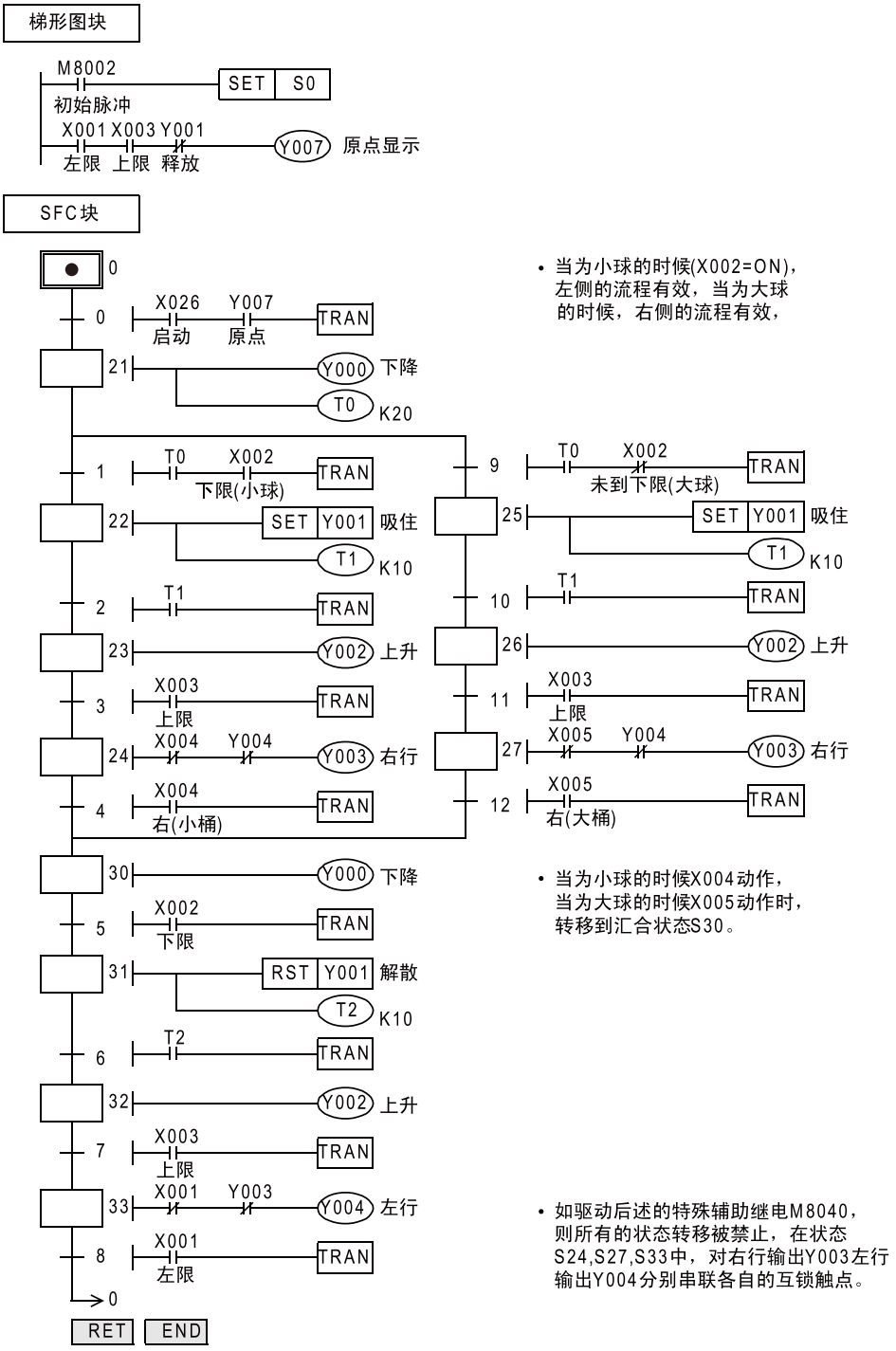


2. 大小球的选择搬运的例子

下图中显示了使用传送带，将大、小球分类传送的机械装置。
左上为原点，按照下降、吸住、上升、右行、下降、释放、上升、左行的顺序动作。
此外，当机械手臂下降，电磁铁压住大球时，下限开关LS2为OFF，压住小球时，LS2为ON。



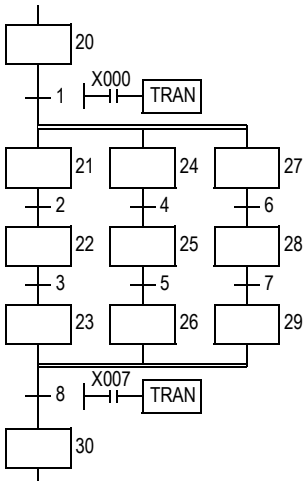
像这种大小分类选择或判别合格与否的SFC程序，就是如下图所示的选择性分支・汇合的SFC程序。



并行分支·汇合流程的例子

1. 并行分支的动作

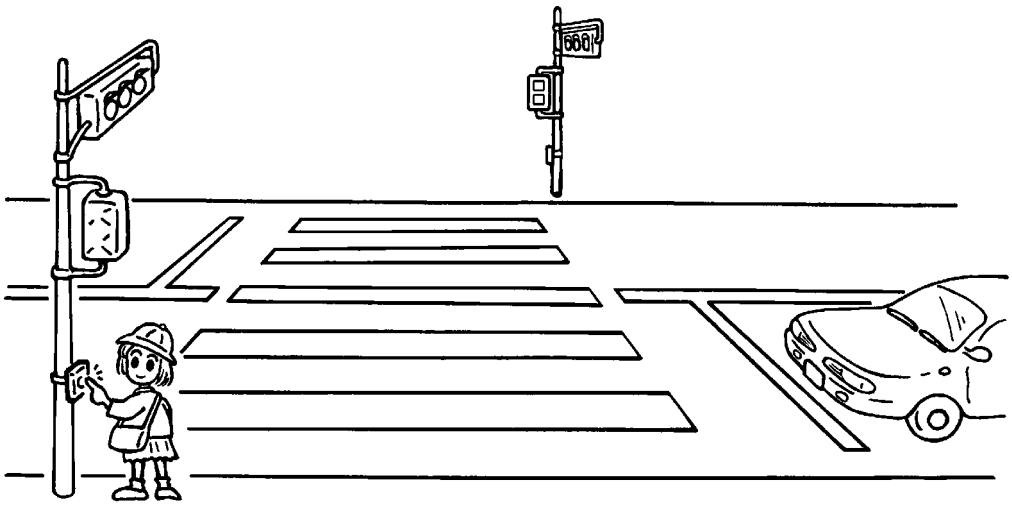
- 多个流程全部同时进行的分支就称为并行分支。
- 在左图的例子中，S20动作，X000接通后，S21、S24、S27同时动作，各流程开始动作。
- 各流程的动作全部都结束，且X007为ON时，汇合状态S30动作，转移源S23、S26、S29都不动作。
- 像这样的汇合还有一个别称，叫做等待汇合。(先结束的流程要等所有流程都动作结束，汇合之后，再继续动作。)



分别并行加工部品A、B、C，加工结束后要进行组装时，也是并行分支·汇合的流程。

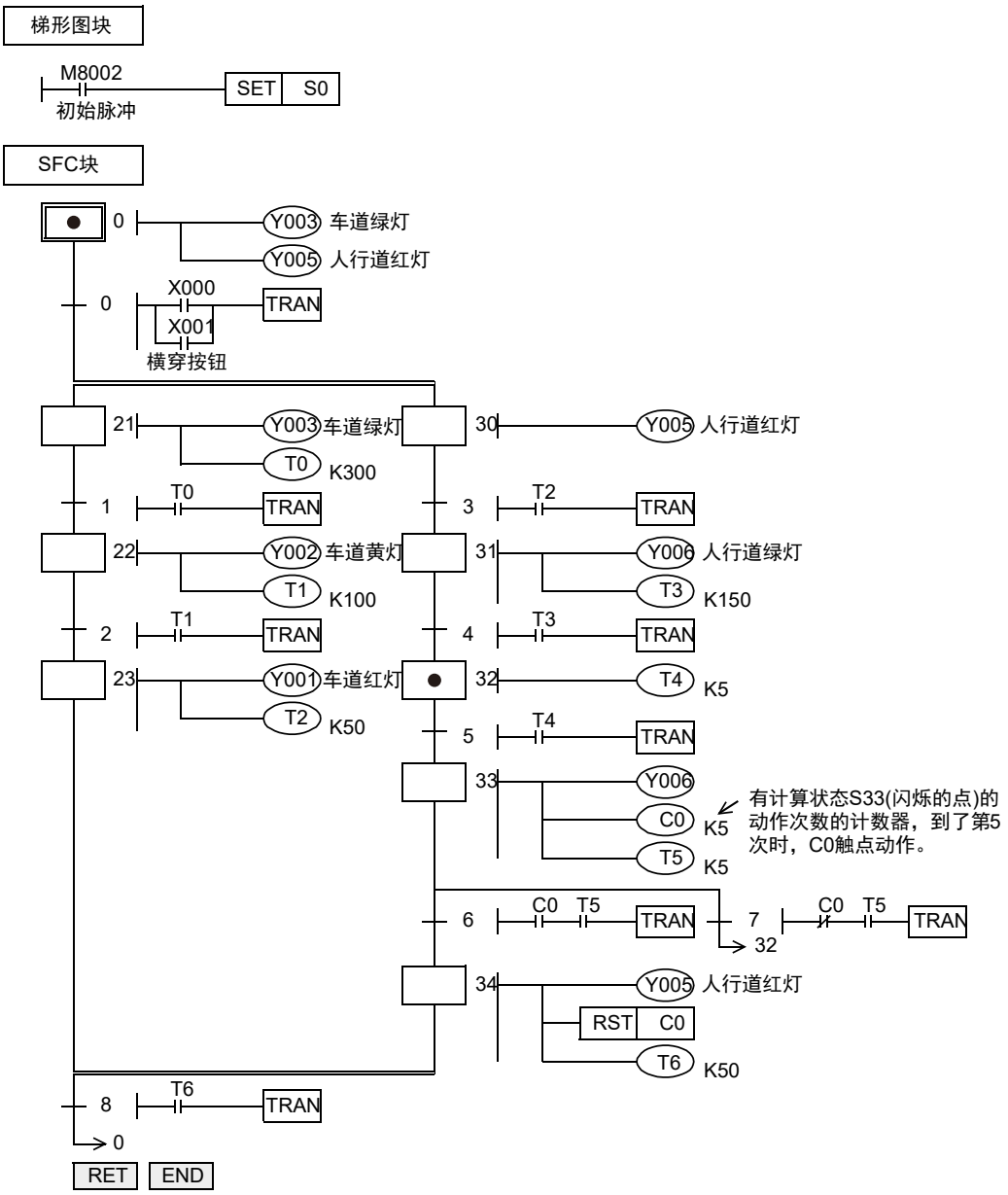
2. 按钮式人行横道的例子

如下图所示，按钮式人行横道的例子中，可以使用并行分支·汇合的流程表示。
Y003:绿灯Y002:黄灯Y001:红灯



按钮式人行横道的SFC程序如下所示，下面例举了在人行横道的绿灯的闪烁部分中，一部分流程的重复动作(跳转到上方的状态)的例子。

- 可编程控制器从STOP切换到RUN时，初始状态S0动作，平时为车道=绿灯、人行道=红灯。
- 按下横穿按钮X000或X001后，在状态S21中车道=绿灯，状态S30中人行道=红灯，状态不改变。
- 30秒以后车道=黄灯，再过10秒以后变成车道=红灯。
- 此后，定时器T2(5秒)动作后，变为人行道=绿灯。
- 15秒以后，人行道执行绿灯的闪烁。(S32=灭、S33=亮)
- 在闪烁过程中，S32、S33重复动作，但是计数器C0(设定值为5次)动作后，动作状态转移到S34，在人行道=红灯的5秒后返回到初始状态。
- 在动作过程中，即使按横穿按钮X000、X001也无效。


梯形图块
M8002
初始脉冲
SET S0
SFC块
0
Y003 车道绿灯
Y005 人行道红灯
0
X000
X001
横穿按钮
TRAN
21
Y003 车道绿灯
T0 K300
1
T0
TRAN
22
Y002 车道黄灯
T1 K100
2
T1
TRAN
23
Y001 车道红灯
T2 K50
30
Y005 人行道红灯
3
T2
TRAN
31
Y006 人行道绿灯
T3 K150
4
T3
TRAN
32
T4 K5
5
T4
TRAN
33
Y006
C0 K5
T5 K5
有计算状态S33(闪烁的点)的动作次数的计数器，到了第5次时，C0触点动作。
6
C0 T5
TRAN
7
C0 T5
TRAN
32
34
Y005 人行道红灯
RST C0
T6 K50
8
T6
TRAN
0
RET END

34.2 步进梯形图

34.2.1 概要

使用步进梯形图指令的程序，与SFC程序一样，以机械的动作为基础，对各工序分配状态S，作为连接在状态触点(STL触点)中的回路，对输入条件和输出控制的顺序进行编程。
关于编写程序的思考方法、状态的种类以及动作都与SFC程序相同，由于可以用梯形图来表示，所以其实质性内容与SFC程序完全相同，可以作为熟悉的继电器梯形图使用。
此外，在步进梯形图中，还可以采用指令表形式编程。
SFC程序和步进梯形图程序都是按照各自固定的规则编程的，可以相互转换。

本章节中，说明了与SFC程序相比，步进梯形图的书写和注意事项，以及用指令表形式的输入顺序。

34.2.2 功能和动作说明

在步进梯形图中，把状态S当作1个控制工序，在其中编写输入条件和输出控制的顺控程序。
由于工序推进时，前工序就转为不动作，所以可以按各工序的简单的顺序来控制机械。

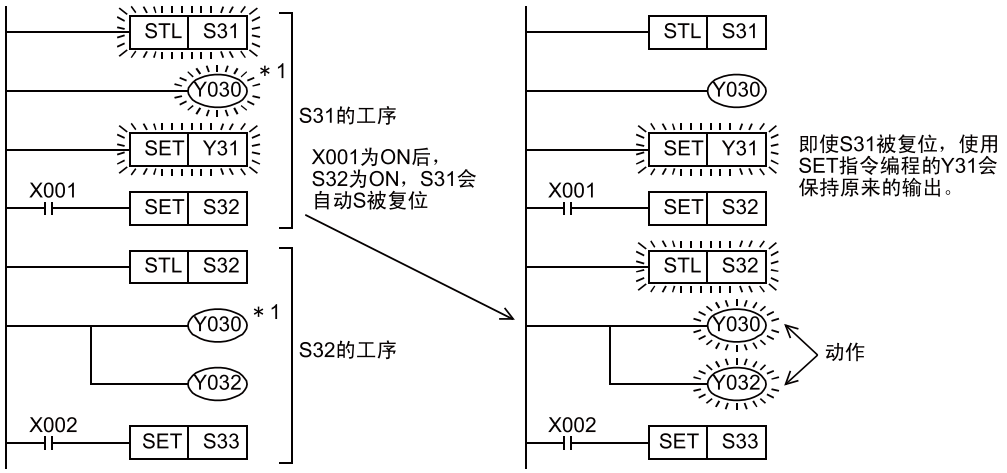
步进梯形图指令的动作

在步进梯形图中，用状态来表示机械运行的各工序。
可以采用这样的思考方式，认为状态和继电器一样，都是由驱动线圈和触点(STL触点)构成的。
在驱动线圈中使用SET指令和OUT指令，在触点中使用STL指令。

状态为ON后，通过STL触电，使与其连接的梯形图(内部梯形图)动作。
状态为OFF后，通过STL触点，使与其连接的内部梯形图不动作。
1个运算周期以后，指令的OFF执行不动作。(跳转的状态)

满足各状态的转移中设置的条件(转移条件)时，下一个状态接通，此前一直为ON的状态断开。(转移动作)
在状态的转移过程中，仅一瞬间(1个运算周期)两个状态会同时接通。
转移前的状态在转移后的在下一个运算周期被OFF(复位)。
但是，通过触点指令使用转移前状态S时，从转移条件被满足之后开始，触点映像被执行OFF。

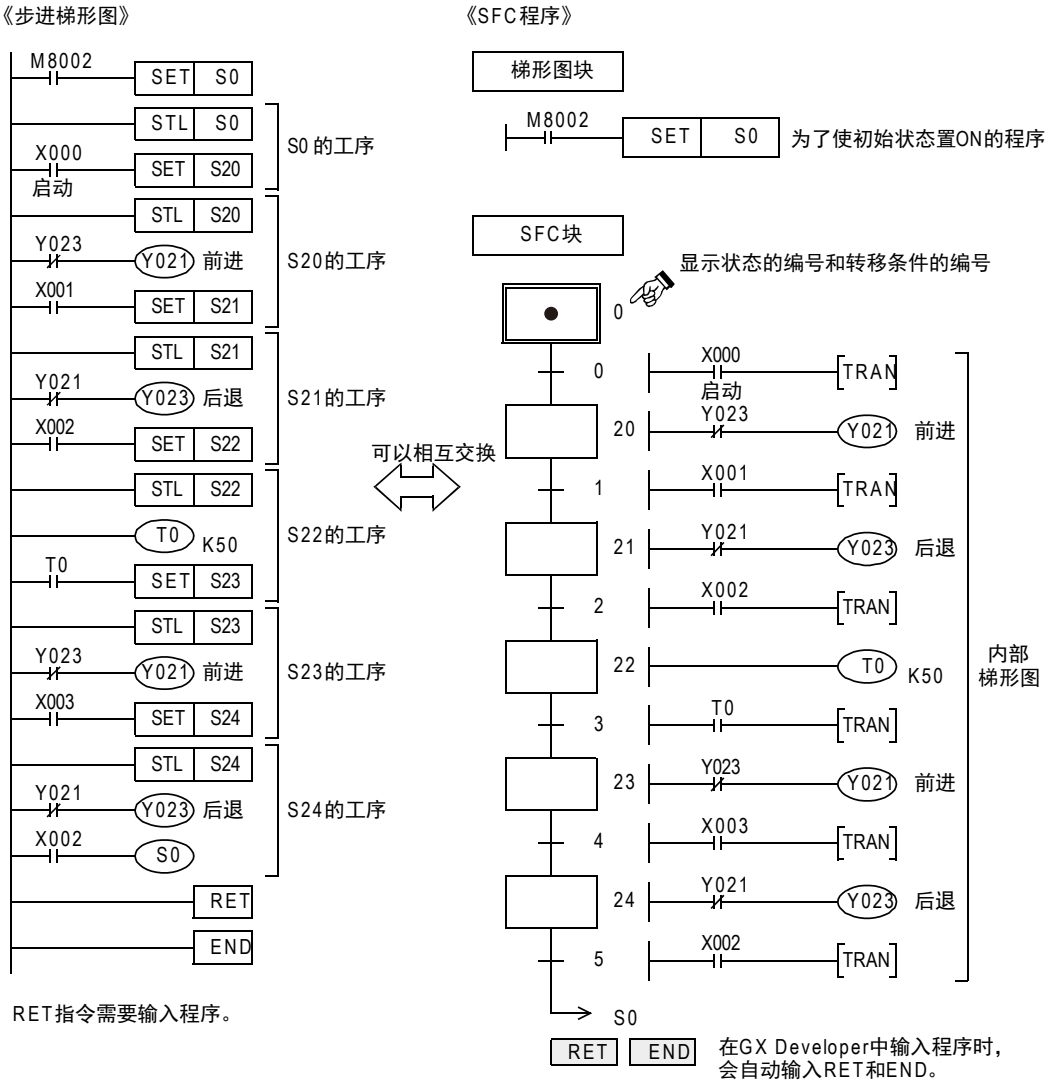
不能重复使用同一个状态编号。



*1. 在不同的状态中，可以重复编写输出线圈。

34.2.3 步进梯形图的形式

正如前面所描述的步进梯形图指令和SFC程序实质是相同的，实际的程序如下所示表示。
步进梯形图始终是继电器梯形图风格的表现形式，使用状态，可以按照机械控制的流程编写程序。

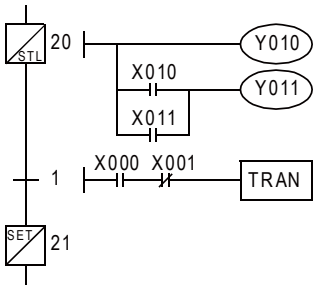


34.2.4 步进梯形图的编写(SFC程序→STL程序)

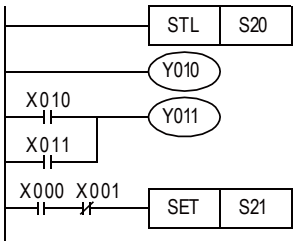
如左图所示，从SFC程序中选取1个状态作为代表。
各状态具备了对负载的驱动处理、指定转移目标以及指定其转移条件的三个功能。
以继电器顺控风格表示这个SFC程序，就是如右图所示的步进梯形图。
程序是按照先对负载的驱动处理，然后执行转移处理的顺序执行的。
当然在不带驱动负载的状态中，不需要进行负载的驱动处理。
此外，关于程序的编写步骤等，也请参考SFC程序。

→ 关于程序的编写步骤，请参考34.1.3项
→ 关于初始状态的使用及作用，请参考34.1.4项
→ 关于停电保持(保持用)状态，请参考34.1.5项
→ 关于RET指令，请参考34.1.6项

《SFC程序》



《步进梯形图》



《指令表程序》

```
0    STL    S20
1    OUT    Y010
2    LD     X010
3    OR     X011
4    OUT    Y011
5    LD     X000
6    ANI    X001
7    SET    S21   *1
```

如果将上述的程序变为指令表形式（指令表程序）就如左图所示。从STL指令开始到RET指令之间的内容，作为步进梯形图处理。

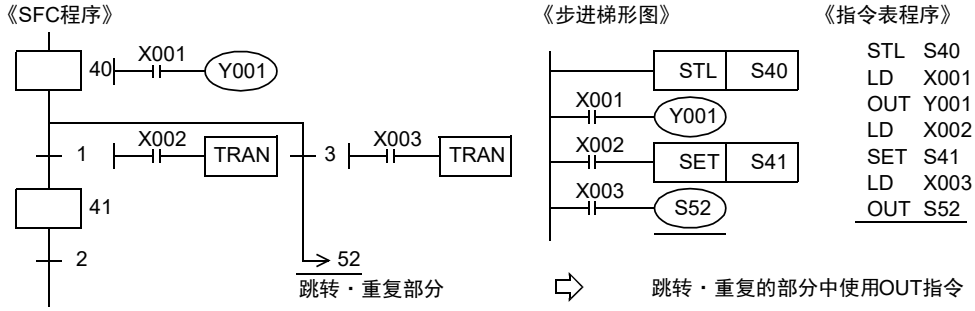
*1. 对状态的SET，RST指令为2个步长的指令。

- 对一连串的SFC程序，执行各个状态的程序，只有当所有的状态都被毫无遗漏地编程时，才可以认为编程结束。
- 对于一连串的步进梯形图，要从初始状态开始，按照要转移的状态的顺序编程。
  此外，请务必在步进梯形图的末尾编写RET指令。

跳转·重复流程的程序

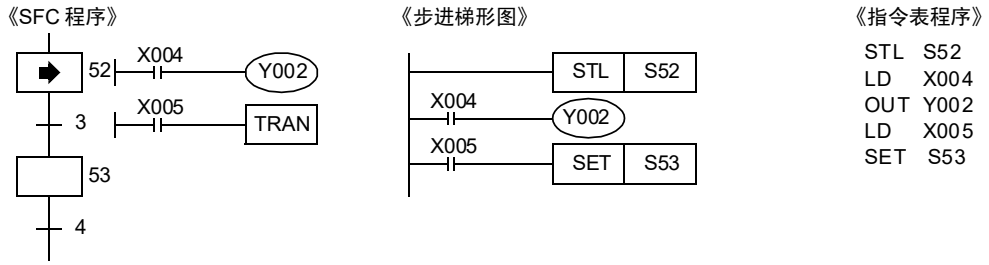
1. 转移源的程序

在跳转・重复的部分中使用OUT指令。



2. 转移目标的程序

转移目标的程序中，没有与编程有关的特别的变化。



34.2.5 编制步进梯形图所需的预备知识

请参考编写SFC程序所需的预备知识。

→ 关于编写SFC程序所需的预备知识，请参考34.1.7项

STL指令～RET指令之间可以使用的顺控指令一览

状态		指令		
		LD/LDI/LDP/LDF, AND/ANI/ANDP/ANDF, OR/ORI/ORP/ORF,INV,MEP/ MEF,OUT,SET/RST,PLS/PLF	ANB/ORB/MPS/ MRD/MPP	MC/MCR
初始状态/一般状态		可以使用	可以使用*1	不可以使用
分支、汇合状态	驱动处理	可以使用	可以使用*1	不可以使用
	转移处理	可以使用	不可以使用	不可以使用

- 中断程序和子程序中不可以使用STL指令。
- 使用SFC程序(STL指令)时，在中断程序中请勿使用SET指令或OUT指令驱动状态S。
- 并非禁止在状态中使用跳转指令，而是由于使用了会产生复杂的动作，所以建议尽量不使用。

*1. 即使是驱动处理梯形图，也不能在STL指令的后面直接使用MPS指令。

特殊辅助继电器

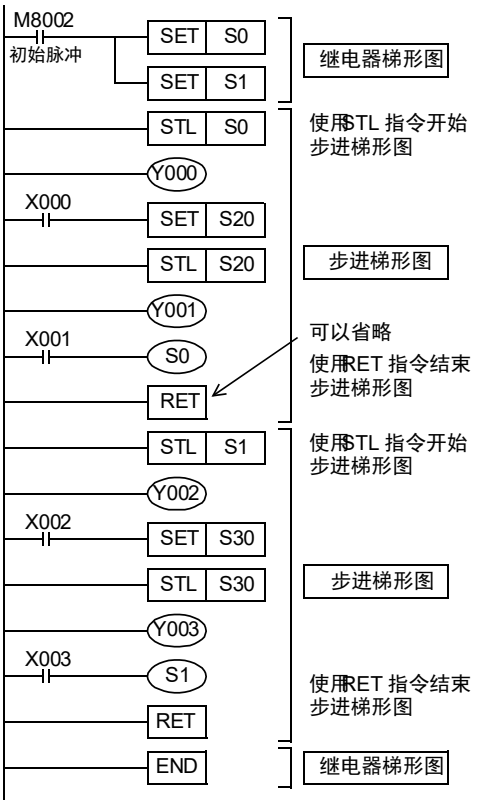
为了能够更有效地编写步进梯形图，需要使用几个特殊辅助继电器，主要的内容如下表所示。
如下所示的特殊辅助继电器与SFC程序中可以使用的相同。

软元件编号	名称	功能及用途
M8000	RUN监控	在可编程控制器运行过程中一直为ON的继电器。 可以作为需要一直驱动的程序的输入条件以及作为可编程控制器的运行状态的显示来使用。
M8002	初始脉冲	仅仅在可编程控制器从STOP切换成RUN的瞬间(1个运算周期)为ON的继电器。 用于程序的初始设定和初始状态的置位。
M8040	禁止转移	驱动了这个继电器后，所有的状态之间都禁止转移。 此外，即使是在禁止转移的状态下，由于状态内的程序仍然动作，所以输出线圈等不会自动断开。
M8046*1	STL动作	即使S0～S899,S1000～S4095只有1个状态为ON时，M8046都会自动置ON。 用于避免与其他流程同时起动，或者用作为工序的动作标志位。
M8047*1	STL监控有效	驱动了这个继电器后，将状态S0～S899，S1000～S4095中正在动作(ON)的状态的最新编号保存到D8040中，将下一个动作(ON)的状态编号保存到D8041中。以下到D8047为止依次保存动作状态(最大8点)。 • 在HCP-works的SFC监控中，即使不驱动这个继电器，也可以实现自动滚动监控。

*1. 在执行END指令时处理

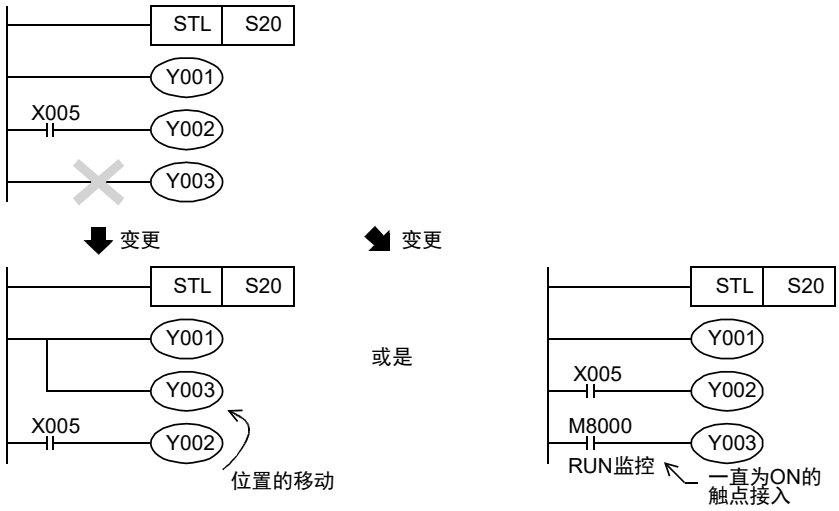
块

由于将步进梯形图作为与继电器梯形图相同的梯形图回路进行编程的，所以不需要像SFC程序那样分成继电器梯形图部分和SFC部分的块。
此外，当多个继电器梯形图和步进梯形图混在一起时，请在步进梯形图的末尾输入RET指令。
可编程控制器根据STL指令开始步进梯形图的处理，根据RET指令从步进梯形图返回到继电器梯形图的处理。
但是，紧接着不同流程的步进梯形图之后编程时(在多个流程的步进梯形图之间不存在继电器梯形图的情况)，允许省略各流程间的RET指令，只在最后的流程的末尾处编写RET指令。



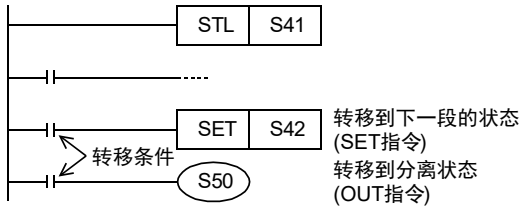
输出的驱动方法

从状态中的母线开始一旦写入LD或是LDI指令后，就不能再编写不需要触点的指令。
请按照下图所示的梯形图进行更改。



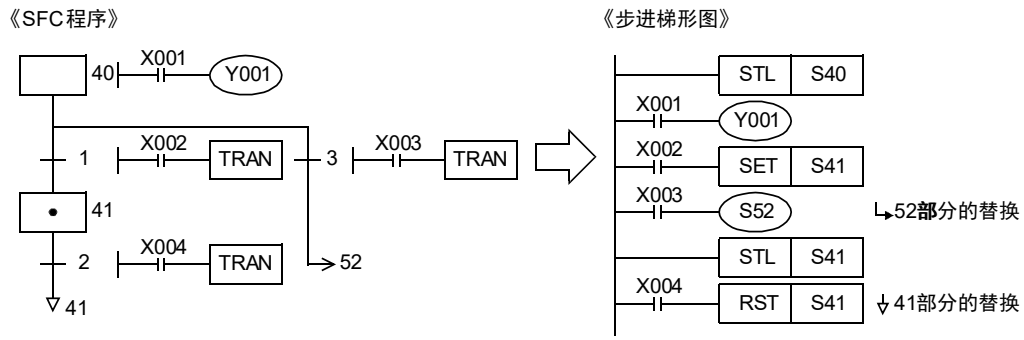
状态的转移方法

对状态使用OUT指令或是SET指令，任何一个都使转移源自动复位。
此外，还具备自保持功能。
但是，OUT指令，用于转移到SFC程序中的分离状态。



[ ]和[▽]的替换

对SFC程序中，表示重复、跳转以及转移到被分离的其他流程的状态的[ ]请使用OUT指令编程，对表示状态复位含义的[▽]请使用RST指令编程。

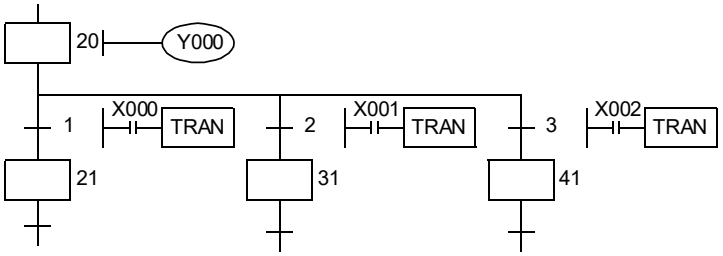


34.2.6 分支·合并状态的程序

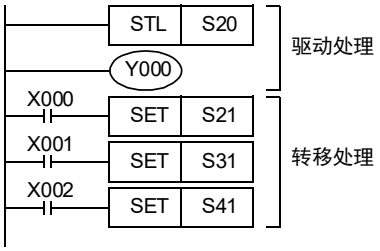
1. 选择性分支的例子

在分支、汇合的转移处理程序中请勿使用MPS、MRD、MPP、ANB、ORB的指令。
而且，即使是在驱动负载的回路中，在STL指令后面也不能使用MPS指令。
与针对一般状态的程序相同，先执行驱动处理，接着执行转移处理。
请依次执行所有的转移处理。

《SFC 程序》



《步进梯形图》



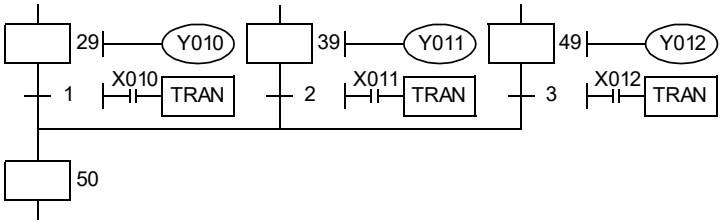
《指令表程序》

```
STL  S20
OUT  Y000
LD   X000
SET  S21
LD   X001
SET  S31
LD   X002
SET  S41
```

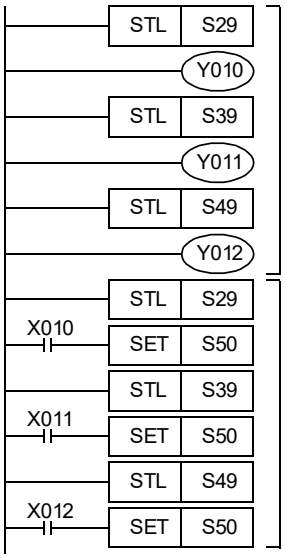
2. 选择汇合的例子

在分支、汇合的转移处理程序中请勿使用MPS、MRD、MPP、ANB、ORB的指令。
而且，即使是在驱动负载的回路中，在STL指令后面也不能使用MPS指令。
在编程的顺序中，请注意分支列和汇合列不能交叉。

《SFC 程序》



《步进梯形图》



《指令表程序》

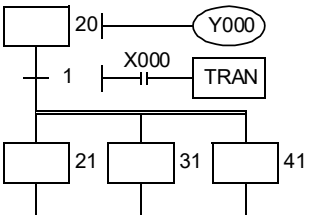
```
STL  S29
OUT  Y010
STL  S39
OUT  Y011
STL  S49
OUT  Y012
STL  S29
LD   X010
SET  S50
STL  S39
LD   X011
SET  S50
STL  S49
LD   X012
SET  S50
```

首先，仅执行汇合前状态的驱动处理。
此后，仅依次执行汇合状态的转移处理。
这是作为逆转换到SFC画面所必须遵循的规则。

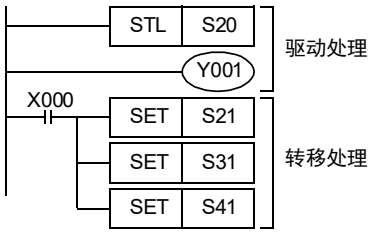
3. 并行分支的例子

在分支、汇合的转移处理程序中请勿使用MPS、MRD、MPP、ANB、ORB的指令。
而且，即使是在驱动负载的回路中，在STL指令后面也不能使用MPS指令。
与针对一般状态的程序相同，先执行驱动处理，接着执行转移处理。
请依次执行所有的转移处理。

《SFC 程序》



《步进梯形图》



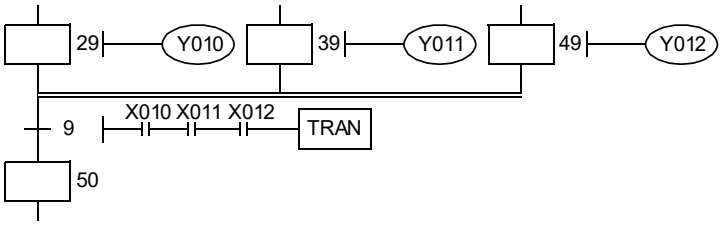
《指令表程序》

```
STL  S20
OUT  Y000
LD   X000
SET  S21
SET  S31
SET  S41
```

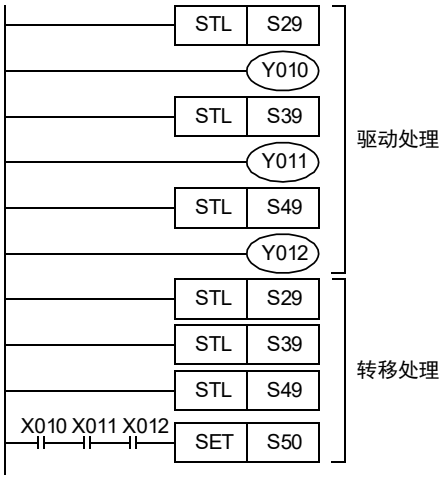
4. 并行汇合的例子

在分支、汇合的转移处理程序中请勿使用MPS、MRD、MPP、ANB、ORB的指令。
而且，即使是在驱动负载的回路中，在STL指令后面也不能使用MPS指令。
在编程的顺序中，请注意分支列和汇合列不能交叉。

《SFC 程序》



《步进梯形图》



《指令表程序》

```
STL  S29
OUT  Y010
STL  S39
OUT  Y011
STL  S49
OUT  Y012
STL  S29
STL  S39
STL  S49
LD   X010
AND  X011
AND  X012
SET  S50
```

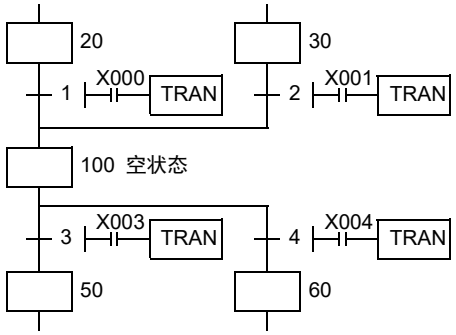
首先，仅执行汇合前状态的驱动处理。
此后，仅依次执行汇合状态的转移处理。

5. 分支・汇合的组合

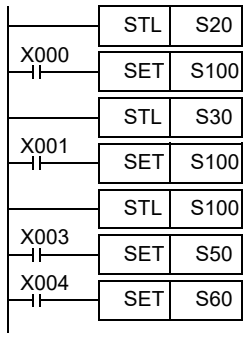
从汇合线开始直接连接分支线中间没有状态时，建议在中间使用一个空状态作为中介。
此时，请按照下图所示制作步进梯形图。

1) 选择汇合和选择分支

《SFC程序》



《步进梯形图》

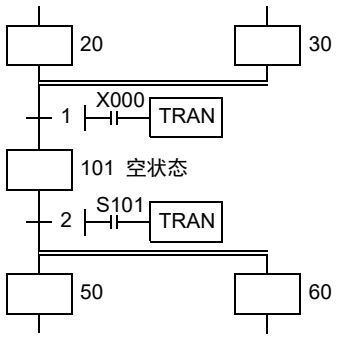


《指令表程序》

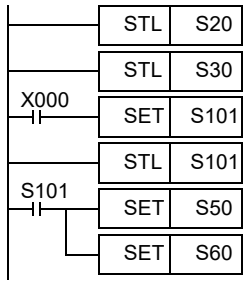
```
STL  S20
LD   X000
SET  S100
STL  S30
LD   X001
SET  S100
STL  S100
LD   X003
SET  S50
LD   X004
SET  S60
```

2) 并行汇合和并行分支

《SFC程序》



《步进梯形图》

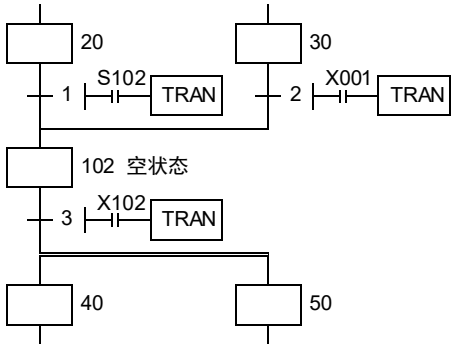


《指令表程序》

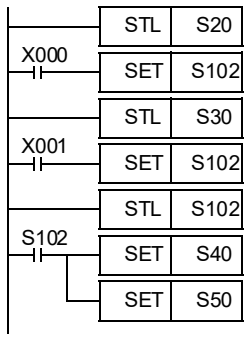
```
STL  S20
STL  S30
LD   X000
SET  S101
STL  S101
LD   S101
SET  S50
SET  S60
```

3) 选择汇合和并行分支

《SFC程序》



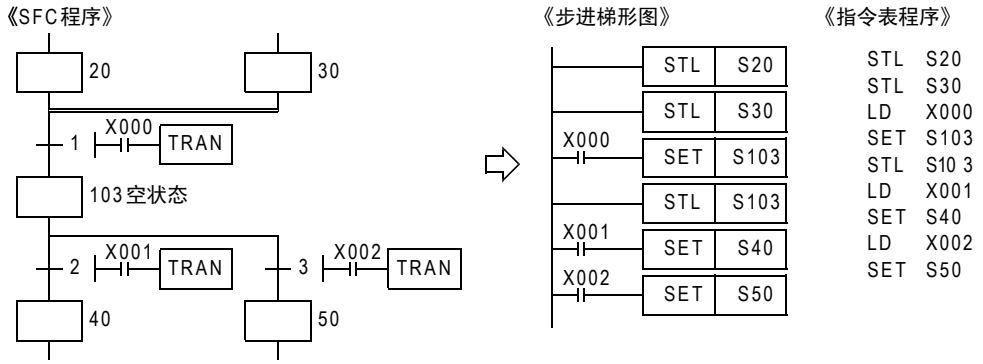
《步进梯形图》



《指令表程序》

```
STL  S20
LD   X000
SET  S102
STL  S30
LD   X001
SET  S102
STL  S102
LD   S102
SET  S40
SET  S50
```

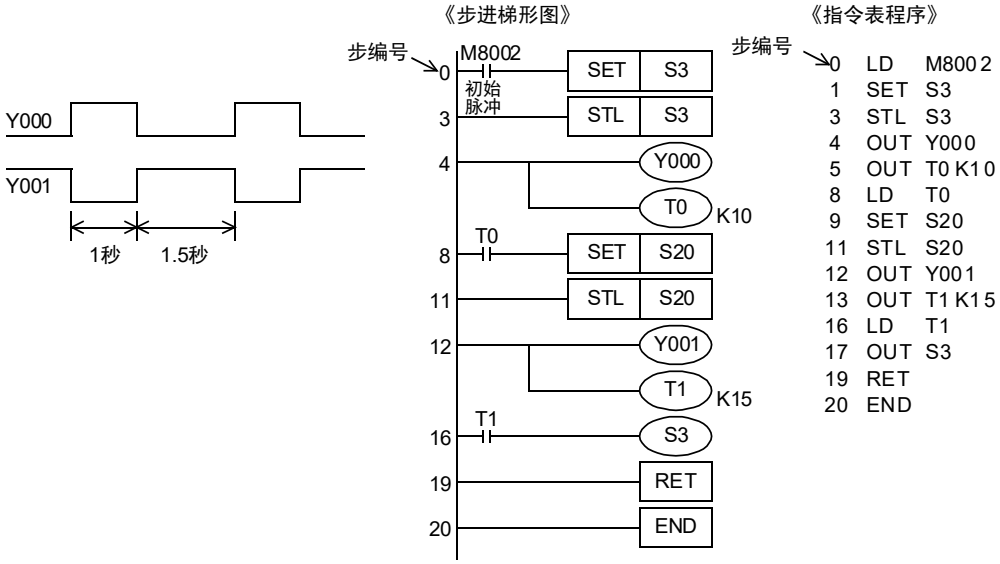
4) 并行汇合和选择分支



34.2.7 程序举例

单流程的例子

1. 闪烁回路的例子
   - 使可编程控制器运行，通过初始脉冲(M8002)驱动状态S3。
   - 在状态S3中输出Y000，1秒钟以后转移到状态S20。
   - 在状态S20中输出Y001，1.5秒钟以后返回状态S3。



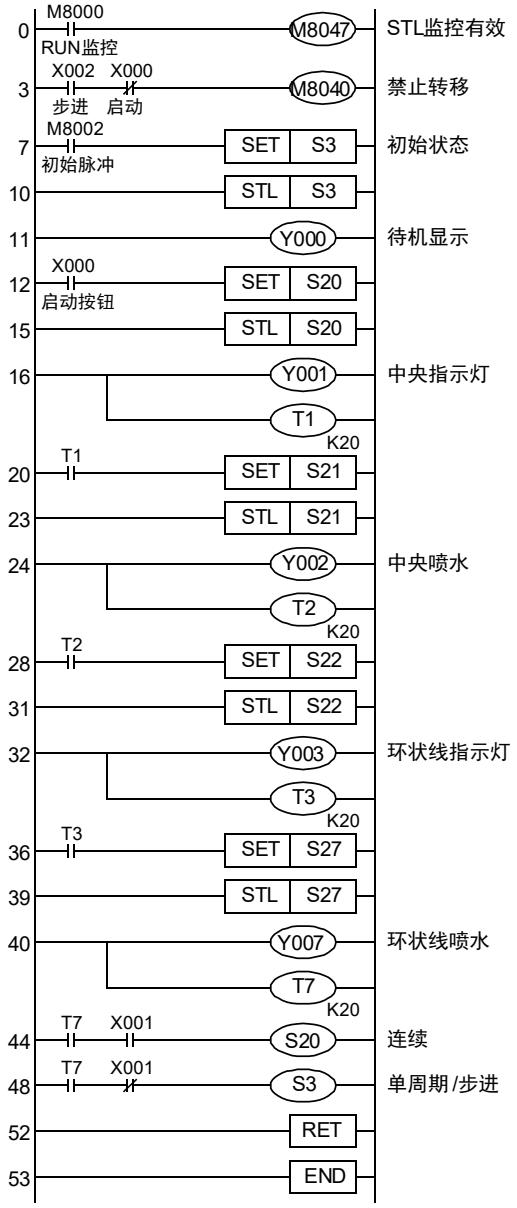
2. 喷水控制的例子

1) 单周期运行(X001=OFF,X002=OFF)
按下启动按钮X000后，按照Y000(待机显示)→Y001(中央指示灯)→Y002(中央喷水)→Y003(环状线指示灯)→Y007(环状线喷水)→Y000(待机显示)的顺序动作，然后返回待机状态。
通过预置的2秒定时器依次切换各输出。

2) 连续运行(X001=ON)
重复Y001～Y007的动作。

3) 步进运行(X002=ON)
每按一次启动按钮，各输出依次动作一次。

《步进梯形图》



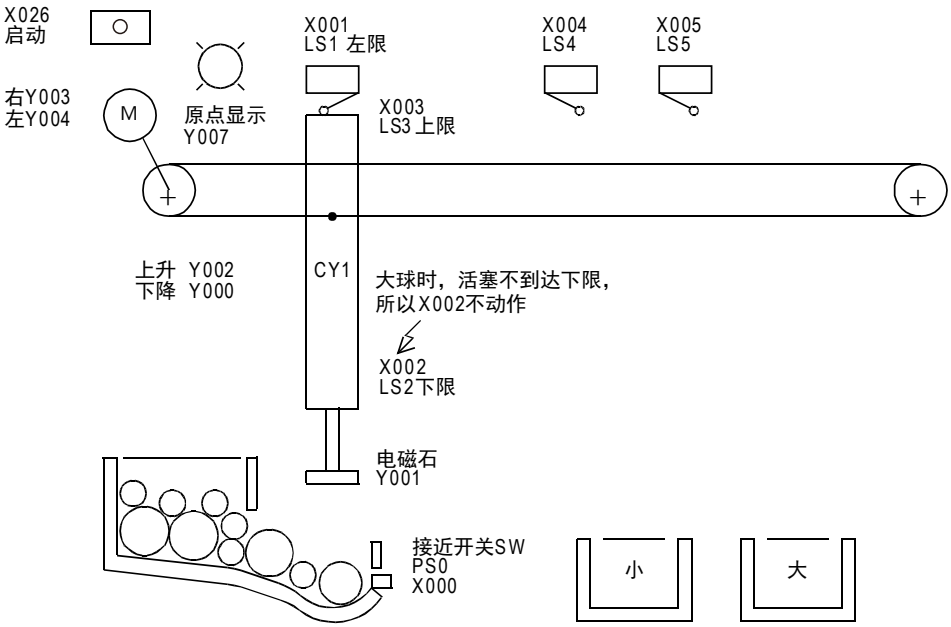
《指令表程序》

```
 0  LD    M8000
 1  OUT   M8047
 3  LD    X002
 4  ANI   X000
 5  OUT   M8040
 7  LD    M8002
 8  SET   S3
10  STL   S3
11  OUT   Y000
12  LD    X000
13  SET   S20
15  STL   S20
16  OUT   Y001
17  OUT   T1 K20
20  LD    T1
21  SET   S21
23  STL   S21
24  OUT   Y002
25  OUT   T2 K20
28  LD    T2
29  SET   S22
31  STL   S22
32  OUT   Y003
33  OUT   T3 K20
36  LD    T3
37  SET   S27
39  STL   S27
40  OUT   Y007
41  OUT   T7 K20
44  LD    T7
45  AND   X001
46  OUT   S20
48  LD    T7
49  ANI   X001
50  OUT   S3
52  RET
53  END
```

选择分支·汇合流程的例子

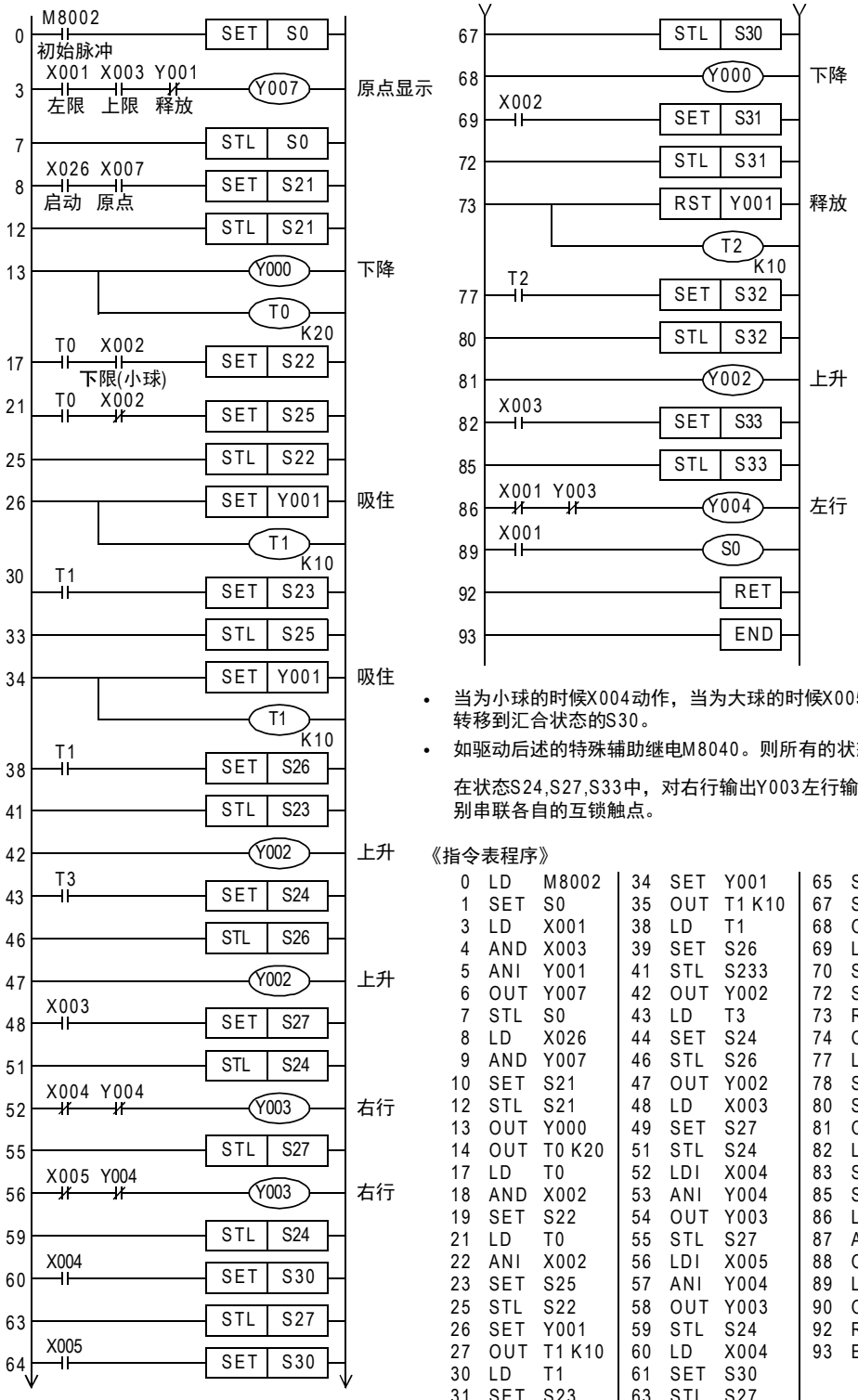
1. 大小球的选择搬运的例子

下图中显示了使用传送带，将大、小球分类传送的机械装置。
左上为原点，按照下降、吸住、上升、右行、下降、释放、上升、左行的顺序动作。
此外，当机械手臂下降，电磁铁压住大球时，下限开关LS2为OFF，压住小球时，LS2为ON。



像这种大小分类选择或判别合格与否的步进梯形图如下图所示。

《步进梯形图》



- 当为小球的时候X004动作，当为大球的时候X005动作时，转移到汇合状态的S30。
- 如驱动后述的特殊辅助继电M8040。则所有的状态转移被禁止。

  在状态S24,S27,S33中，对右行输出Y003左行输出Y004分别串联各自的互锁触点。

《指令表程序》

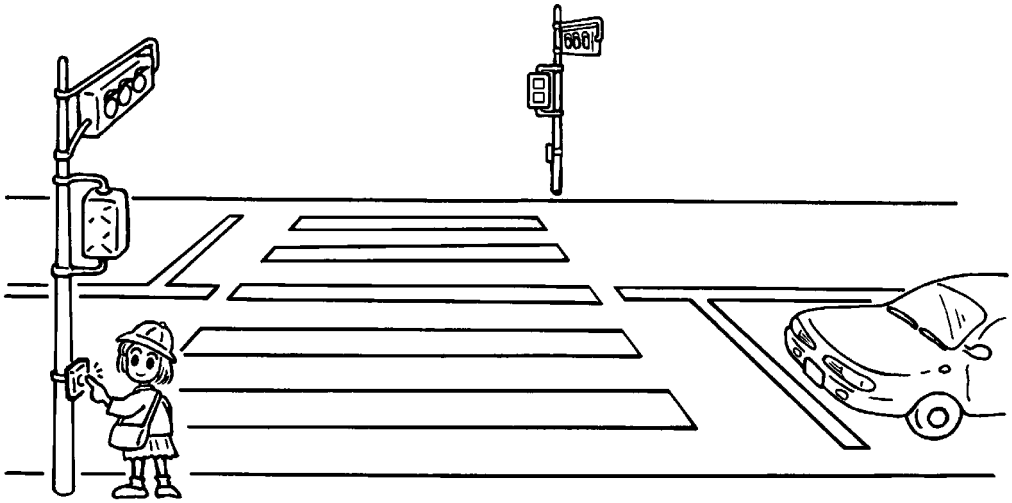
0	LD	M8002	34	SET	Y001	65	SET	S30
1	SET	S0	35	OUT	T1 K10	67	STL	S30
3	LD	X001	38	LD	T1	68	OUT	Y000
4	AND	X003	39	SET	S26	69	LD	X002
5	ANI	Y001	41	STL	S233	70	SET	S31
6	OUT	Y007	42	OUT	Y002	72	STL	S31
7	STL	S0	43	LD	T3	73	RST	Y001
8	LD	X026	44	SET	S24	74	OUT	T2 K10
9	AND	Y007	46	STL	S26	77	LD	T2
10	SET	S21	47	OUT	Y002	78	SET	S32
12	STL	S21	48	LD	X003	80	STL	S32
13	OUT	Y000	49	SET	S27	81	OUT	Y002
14	OUT	T0 K20	51	STL	S24	82	LD	X003
17	LD	T0	52	LDI	X004	83	SET	S33
18	AND	X002	53	ANI	Y004	85	STL	S33
19	SET	S22	54	OUT	Y003	86	LDI	X001
21	LD	T0	55	STL	S27	87	ANI	Y003
22	ANI	X002	56	LDI	X005	88	OUT	Y004
23	SET	S25	57	ANI	Y004	89	LD	X001
25	STL	S22	58	OUT	Y003	90	OUT	S0
26	SET	Y001	59	STL	S24	92	RET	
27	OUT	T1 K10	60	LD	X004	93	END	
30	LD	T1	61	SET	S30			
31	SET	S23	63	STL	S27			
33	STL	S25	64	LD	X005			

并行分支·汇合流程的例子

分别并行加工部品A、B、C，加工结束后要进行组装时，也是并行分支・汇合的流程。

1. 按钮式人行横道的例子

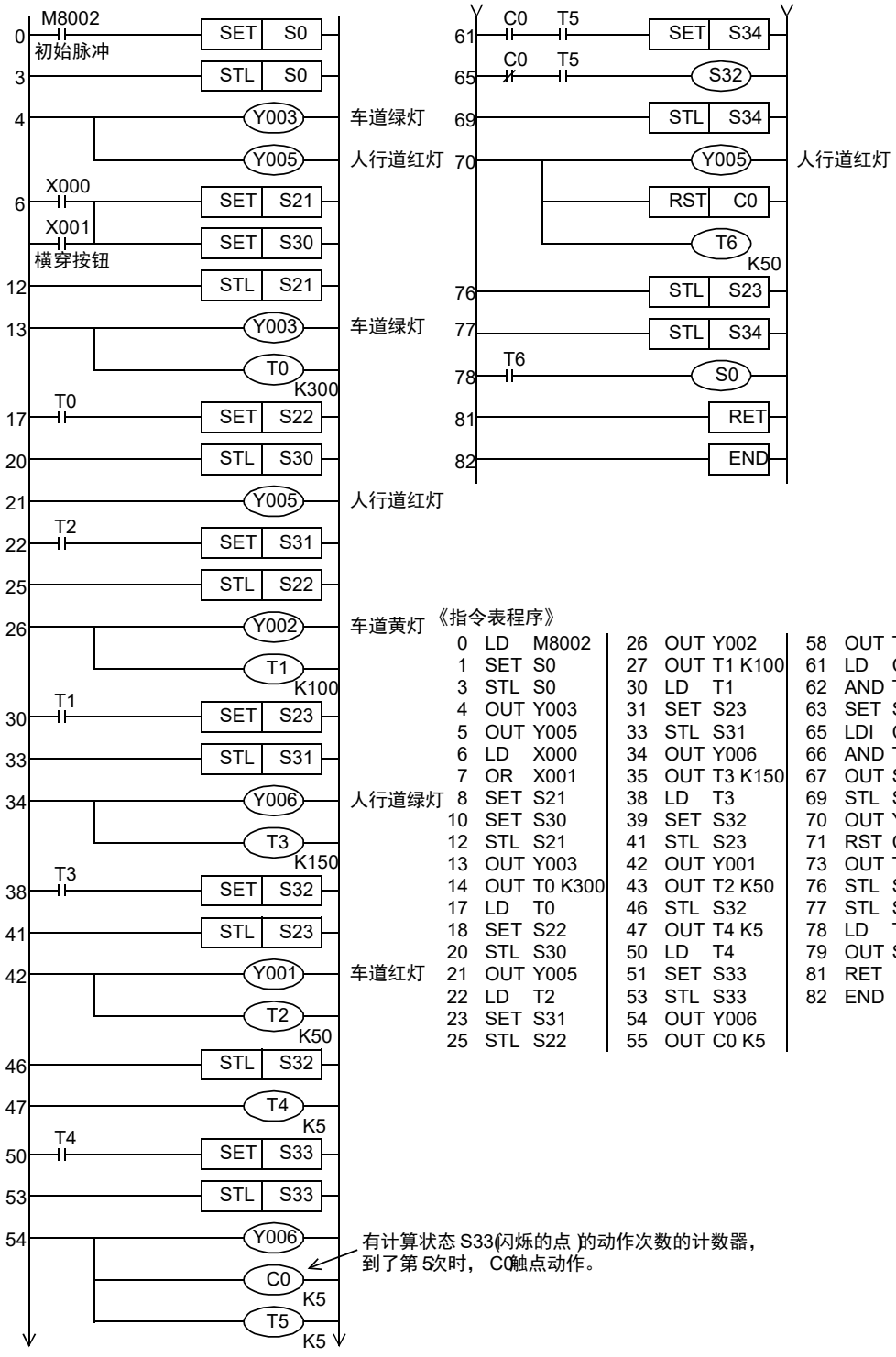
如下图所示，按钮式人行横道的例子中，可以使用并行分支・汇合的流程表示。
Y003:绿灯Y002:黄灯Y001:红灯



按钮式人行横道的SFC程序如下所示，下面例举了在人行横道的绿灯的闪烁部分中，一部分流程的重复动作(跳转到上方的状态)的例子。

- 可编程控制器从STOP切换到RUN时，初始状态S0动作，平时为车道=绿灯，人行道=红灯。
- 按下横穿按钮X000或X001后，在状态S21中车道=绿灯，状态S30中人行道=红灯，状态不改变。
- 30秒以后车道=黄灯，再过10秒以后变成车道=红灯。
- 此后，定时器T2(5秒)动作后，变为人行道=绿灯。
- 15秒以后，人行道执行绿灯的闪烁。(S32=灭、S33=亮)
- 在闪烁过程中，S32、S33重复动作，但是计数器C0(设定值为5次)动作后，动作状态转移到S34，在人行道=红灯的5秒后返回到初始状态。
- 在动作过程中，即使按横穿按钮X000、X001也无效。

《步进梯形图》



有计算状态 S33(闪烁的点)的动作次数的计数器，到了第 5次时， C0触点动作。

《指令表程序》

0	LD	M8002	26	OUT	Y002	58	OUT	T5 K5
1	SET	S0	27	OUT	T1 K100	61	LD	C0
3	STL	S0	30	LD	T1	62	AND	T5
4	OUT	Y003	31	SET	S23	63	SET	S34
5	OUT	Y005	33	STL	S31	65	LDI	C0
6	LD	X000	34	OUT	Y006	66	AND	T5
7	OR	X001	35	OUT	T3 K150	67	OUT	S32
8	SET	S21	38	LD	T3	69	STL	S34
10	SET	S30	39	SET	S32	70	OUT	Y005
12	STL	S21	41	STL	S23	71	RST	C0
13	OUT	Y003	42	OUT	Y001	73	OUT	T6 K50
14	OUT	T0 K300	43	OUT	T2 K50	76	STL	S23
17	LD	T0	46	STL	S32	77	STL	S34
18	SET	S22	47	OUT	T4 K5	78	LD	T6
20	STL	S30	50	LD	T4	79	OUT	S0
21	OUT	Y005	51	SET	S33	81	RET	
22	LD	T2	53	STL	S33	82	END	
23	SET	S31	54	OUT	Y006			
25	STL	S22	55	OUT	C0 K5			

35. 中断功能和脉冲捕捉功能

在本章中，主要说明HC可编程控制器中内置的中断功能及脉冲捕捉功能。

35.1 概要

主要说明了不受顺控程序(主程序)的运算周期的影响，将下述的中断功能作为触发信号，立即执行中断程序(中断子程序)的功能。
在一般的顺控程序处理中，由运算周期造成的延迟以及时间的偏差给机械动作带来影响，这样的情况可以得到改善。

1. 输入中断功能(外部信号输入(X)的中断)
使用输入X000～X005的输入信号，中断一般的顺控程序，优先执行中断子程序。
此外，输入中断的执行时序，可以通过指定指针的编号，指定信号的上升沿或是下降沿中任意一个。
→ 详细内容，参考35.3节

2. 输入中断延迟功能(外部信号输入(X)的中断)
使用输入X000～X005的输入信号，中断一般的顺控程序，在以1ms为单位设定的延迟时间之后优先执行中断子程序。
此外，输入中断的执行时序，可以通过指定指针的编号，指定信号的上升沿或是下降沿中任意一个。
→ 详细内容，参考35.4节

3. 定时器中断功能(按固定周期动作的定时器中断)
按照10ms～99ms的固定周期间隔，中断一般的顺控程序，优先执行中断子程序。
→ 详细内容，参考35.5节

4. 高速计数器中断功能(增计数时的中断功能)
当高速计数器的当前值达到规定值时，中断一般的顺控程序，优先执行中断子程序。
→ 详细内容，参考35.6节

5. 脉冲捕捉功能
通过输入X000～X007的输入信号从OFF变为ON，特殊辅助继电器M8170～M8177被设置为中断处理。通过在一般的顺控程序中使用这个M8170～M8177，可以方便地获取在一般的输入处理中无法获取的ON宽度的信号。
但是，如果在1个运算周期中要执行几次ON/OFF这样的处理时，请使用输入中断功能。
→ 详细内容，参考35.7节

6. 脉宽/周期测量功能
通过输入X000、X001、X003、X004的输入信号OFF→ON，将输入信号上升沿的1/6μs环形计数器值保存到特殊数据寄存器中。此外，通过输入信号的ON→OFF，将输入信号下降沿的1/6μs环形计数器值保存到特殊数据寄存器中，同时将上升沿与下降沿的计数器值的差设定为1/60后，将脉宽(10μs单位)保存到特殊数据寄存器中。
设定为脉冲周期测量模式时，通过输入信号的OFF→ON，将上一次输入信号上升沿至此次输入信号上升沿的差设定为1/60后，将10μs单位的脉冲周期保存到特殊数据寄存器中。
→ 详细内容，参考35.8节

35.2 通用事项

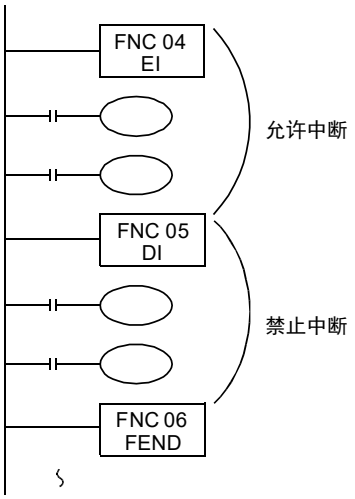
35.2.1 中断功能及脉冲捕捉功能的禁止方法

说明了中断功能及脉冲捕捉功能的禁止方法。

1. 程序的中断范围的限制[中断功能，脉冲捕捉功能]

1) 编程方法
通过编写FNC 05(DI)指令，可以设定中断禁止的区域。
在DI～EI指令之间(禁止中断区域)即使发生中断，也是被存储下来，在EI执行之后再执行。

2) 程序举例



3) 注意事项
a) 禁止中断用的特殊辅助继电器(M8050~M8059)不包括已经置ON中的中断输入。
这个特殊辅助继电器对脉冲捕捉功能无效。
b) 禁止区域的时间较长时，虽然接收中断，但是转移到中断处理会延迟。
不需要禁止中断时，请只编写EI指令。未必一定需要DI指令。

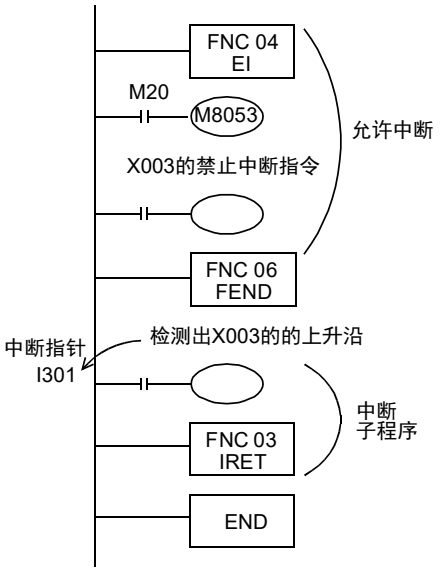
2. 中断指针(每个中断子程序)的禁止中断[中断功能]

1) 编程方法
分别准备了禁止中断用的特殊辅助继电器(M8050～M8059)。
在中断禁止标志位(M8050～M8059)为ON时，产生了禁止的中断，
此后即使中断禁止标志位断开，也不执行中断程序。

输入中断	X000～X005的输入中断分别对应M8050～M8055*1，在ON时禁止
定时器中断	I6□□～I8□□定时器中断分别对应M8056～M8058*1，在ON时禁止
高速计数器中断	I010～I060的计数器中断在M8059*1为ON时全部禁止

*1. 从RUN→STOP时清除

2) 程序举例
在如下所示的例子中，通过M20使M8053置ON时，来自X003的中断输入I301被禁止。



35.2.2 相关事项

1. 输入输出刷新功能(REF指令)的使用
通过中断程序控制输入继电器和输出继电器时，使用输入输出刷新指令(REF/FNC 50)，可获得最新的输入信息，并能立即输出运算结果，从而可以实现不受可编程控制器运算周期影响的高速控制。

2. 关于FROM/TO指令执行过程中的中断动作
通过特殊辅助继电器M8028的ON/OFF，实现以下的动作。

1) M8028=OFF时
在FROM/TO指令执行过程中自动处于禁止中断的状态，不执行输入中断和定时器中断。
在此期间产生的中断会在执行完FROM/TO指令后被立即执行。
当M8028=OFF，FROM/TO指令可以在中断程序中使用。

2) M8028=ON时
在FROM/TO指令执行过程中如产生中断，则中断执行，然后执行中断程序。
但是，当M8028=ON时，不能在中断子程序中使用FROM/TO指令。

35.2.3 使用上的注意事项(通用)

说明了使用中断功能和脉冲捕捉功能时的通用的注意要点。
关于各中断功能个别的注意要点，会在各中断功能的说明中解释。

1. 关于发生多个中断时的处理
依次发生多个中断时，先发生的优先执行，完全同时产生时，指针编号小的优先执行。
在执行中断子程序的过程中，其他的中断被禁止。

2. 想实现双重中断(中断中的中断)时[中断功能]
一般的中断子程序(程序)中禁止中断。A8/A8C的情况下，如果在中断子程序中编写EI(FNC 04)、DI(FNC 05)时,可以接收到双重的中断。

3. 关于使用定时器时的动作[中断功能]
使用普通的定时器不能执行计时，除此以外，使用1ms的累计定时器时也需要注意。
中断子程序中的定时器，请使用子程序用定时器T192～T199。

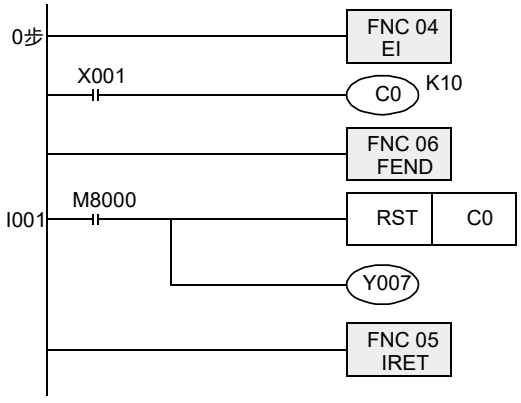
4. 禁止重复使用输入[输入中断(有/无延迟功能)、脉冲捕捉功能]
输入X000～X007用于高速计数器、输入中断、脉冲捕捉以及SPD,ZRN,DSZR,DVIT 指令和通用输入。因此，请勿重复使用输入端子。

5. 使用SFC程序(STL指令)时
使用SFC程序(STL指令)时，在中断程序中请勿使用SET指令或OUT指令驱动状态S。

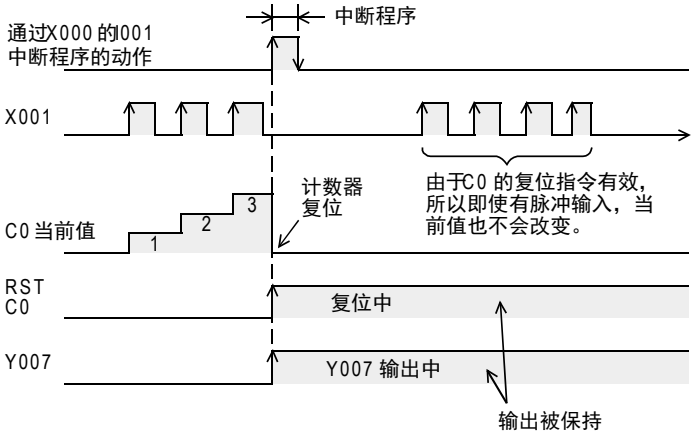
6. 关于已经置ON的软元件的动作[中断功能]
中断子程序中被置ON的软元件，在结束子程序后仍然被保持。此外，对定时器和计数器执行RST指令后，定时器和计数器的复位状态也被保持。
这些软元件在子程序内、或是子程序外执行复位和OFF运算时，请使该指令断开。

输出被保持的例子
有对X001进行计数的C0，当X000从OFF变为ON时，中断程序I001只执行1个扫描，计数器复位后输出Y007的程序。

1) 程序举例

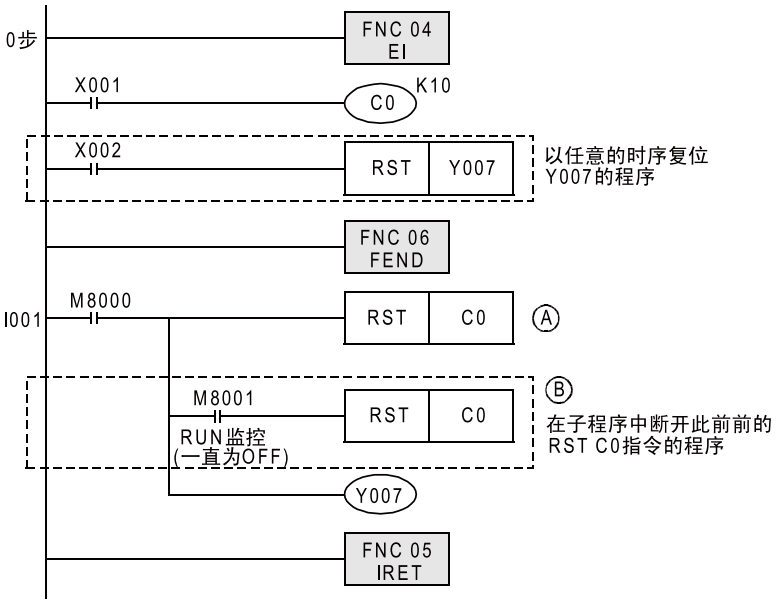


2) 时序图

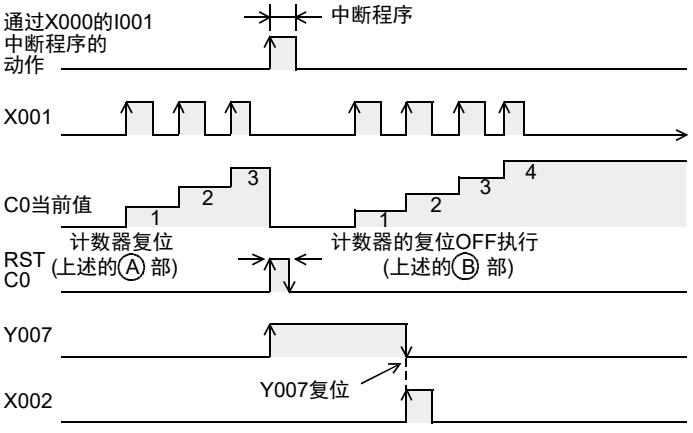


复位被保持的输出的例子(对策)

1) 程序举例



2) 时序图



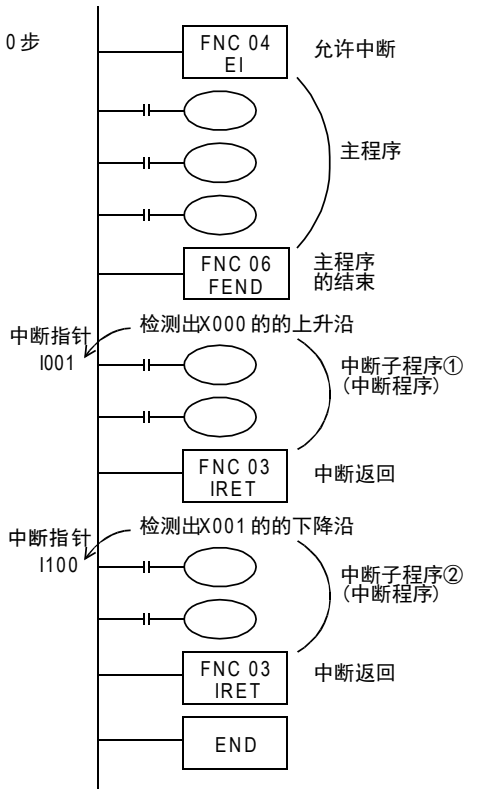
35.3 输入中断(通过外部信号中断)[不使用延迟功能]

35.3.1 输入中断(通过外部信号中断)[不使用延迟功能]

1. 概要
   使用输入X000～X005的输入信号，执行中断子程序。

2. 用途
   由于可以不受可编程控制器的运算周期的影响处理外部输入信号，所以适用于执行高速控制和获取短时间脉冲。

3. 基本程序(编程要领)



主程序
在EI指令后中断输入接收有效
此外不需要输入中断的禁止区域时，
就没有必要编写DI（禁止中断）指令。

FEND 指令为主程序的结束
中断子程序必须在END 之后描述

当X000 接通后，检测出
其上升沿，执行中断子程序①
通过RET 指令返回到主程序。

当X001 断开后，检测出其
下降沿，执行中断子程序②
通过RET 指令返回到主程序。

END 表示程序的结束

4. 中断指针(6点)的编号及动作

I □0 □
↑ ↑0: 下降沿中断 1: 上升沿中断
根据输入X000～X005为0～5

输入编号	指针编号		禁止中断的指令
	上升沿中断	下降沿中断	
X000	I001	I000	M8050*1
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. 从RUN→STOP时清除

5. 中断输入的个别禁止方法

在程序中使M8050～M8055置ON后，则其各自对应的输入编号的中断被禁止。(对应内容参考上表)

6. 注意要点

1) 禁止输入的重复使用
用作为中断指针的输入继电器的编号，请勿与使用相同输入范围的"高速计数器"、"脉冲捕捉功能"、" 脉冲密度(FNC 56)"等应用指令重复。

2) 关于输入滤波器的自动调节
指定了输入中断指针I□0□后，输入继电器的输入滤波器会被自动更改为高速读取用。
因此，不需要使用REFF(FNC 51)指令和特殊数据寄存器D8020(输入滤波器的调节)更改滤波器的调节。
此外，没有作为输入中断指针使用的输入继电器的输入滤波器以10ms(初始值)动作。

3) 输入中断的脉宽
为了能通过外部信号执行输入中断，所以需要输入下表所示的宽度以上的ON信号，或是OFF信号。

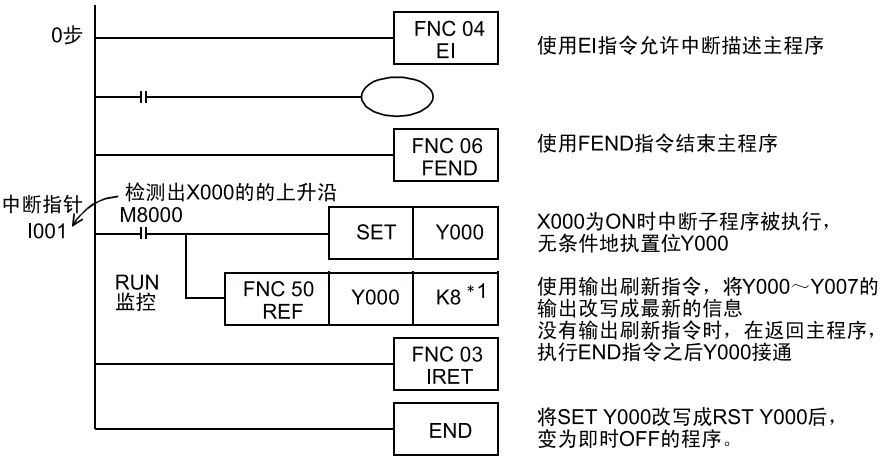
可编程控制器	输入编号	输入滤波常数设定为0时的值
A8/A8C	X000～X005	5μs*2

*2. 使用5μs的输入滤波常数时，以及用高速计数器读取50k～100kHz的脉冲时，请按如下所示设置。
- 接线长度请勿超出5m。
- 在输入端子中连接1.5kΩ(1W以上)的漏电阻，与主机的输入电流相符合，对方一侧机器的开集电极型晶体管输出的负载电流需要在20mA以上。

4) 指针编号的重复使用
对像I001和I000等那样的同一输入的上升沿中断和下降沿中断，不能被同时编写。

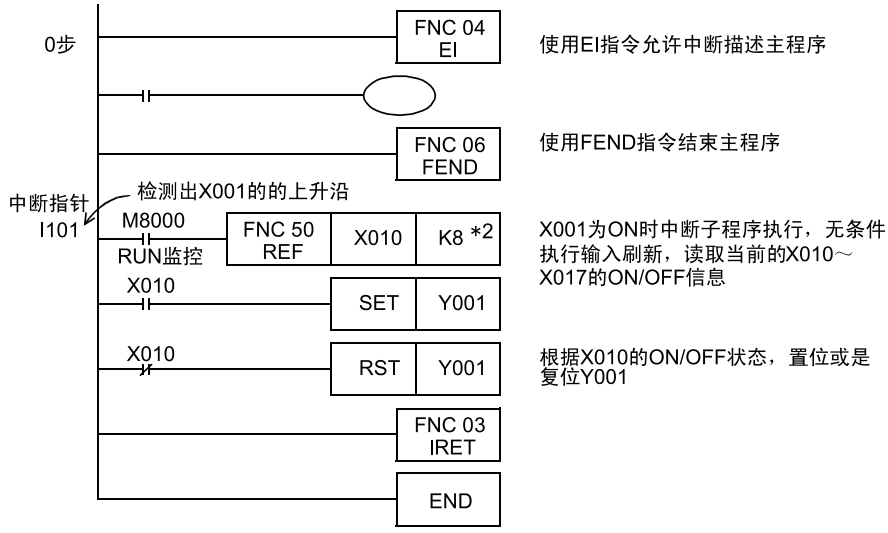
7. 程序举例

1) 上升沿外部输入中断和输出刷新(REF指令)的组合使用
通过外部输入(X000)的上升沿，即刻使输出Y007置ON的程序。



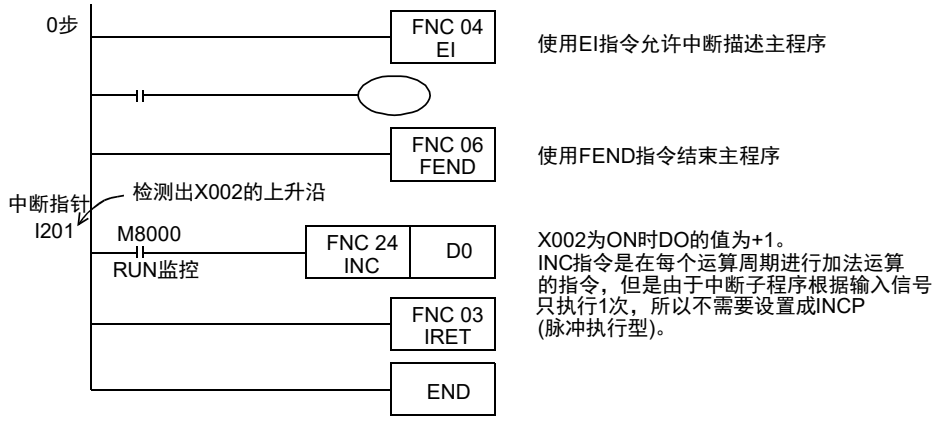
*1. 对使用REF(FNC 50)指令刷新的输入/输出的点数必须指定8的倍数。
指令了8的倍数以外时，会出现运算错误，REF(FNC 50)指令不能执行。

2) 输入中断和输入刷新(REF指令)的组合使用
使用最新的输入信息执行中断处理的程序。

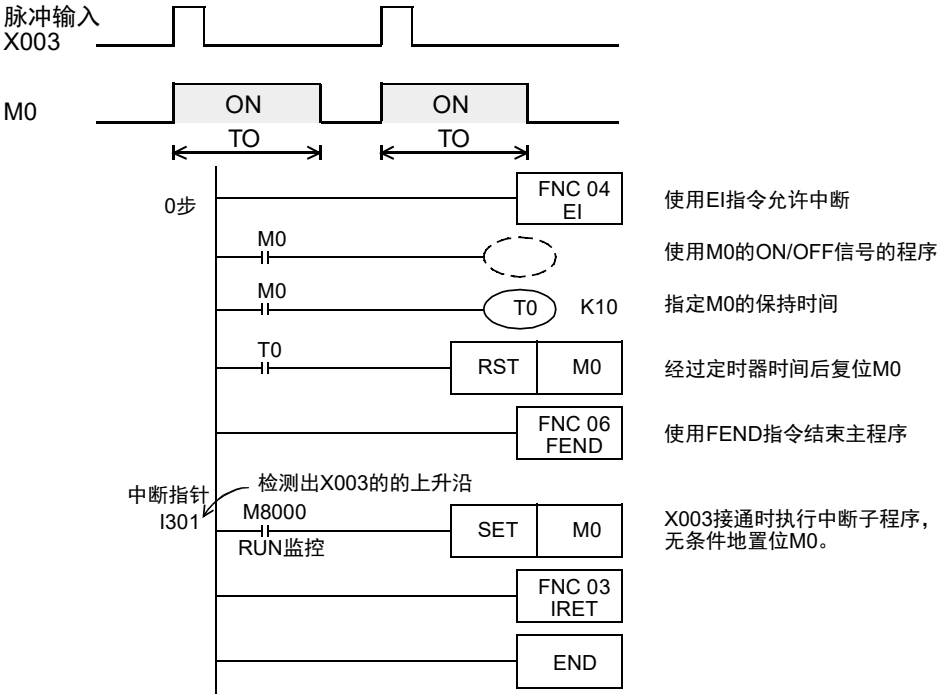


*2. 对使用REF(FNC 50)指令刷新的输入/输出的点数必须指定8的倍数。
指令了8的倍数以外时，会出现运算错误，REF(FNC 50)指令不能执行。

3) 输入发生次数的计数(与单相高速计数器相同)
对外部输入进行计数的程序。



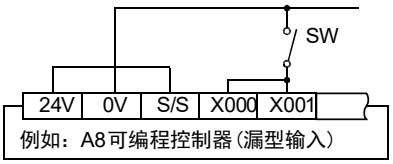
4) 短时间脉冲的捕捉
短时间脉冲为ON后，在一定时间之内保持ON的程序。

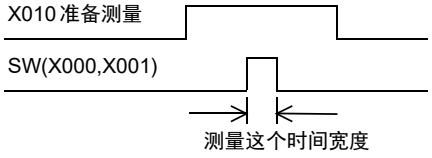


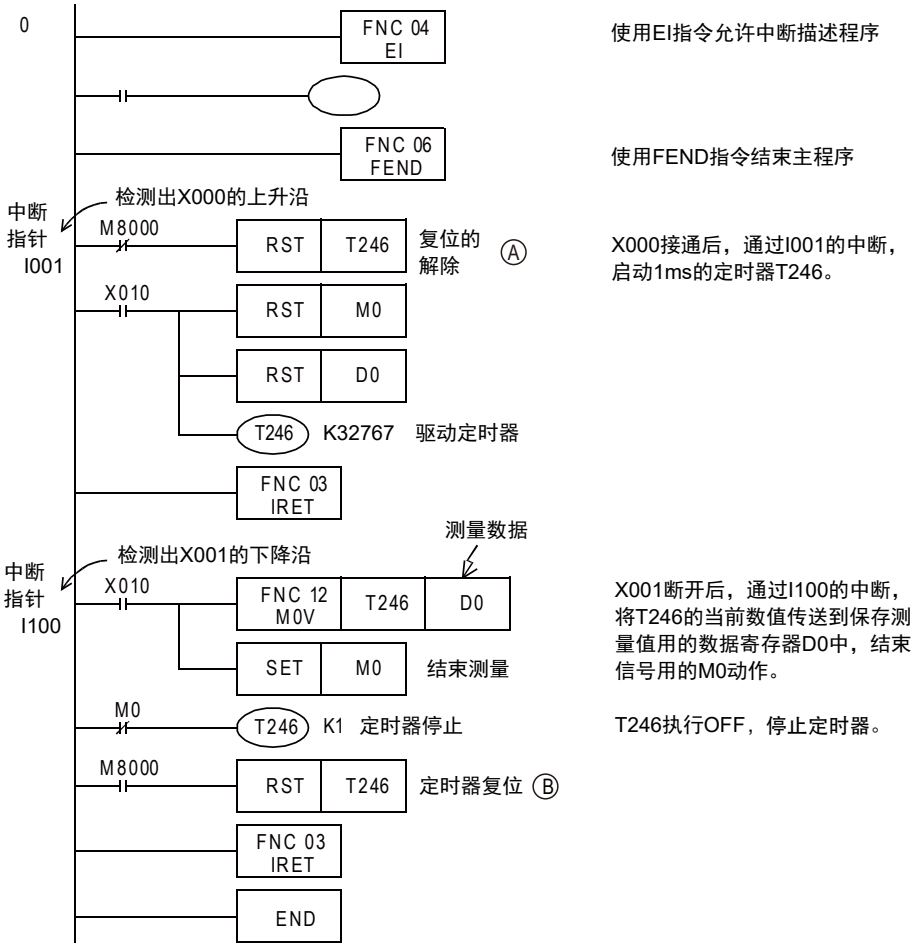
35.3.2 实用程序举例(短时间脉宽的测量程序)

如果使用累计型的1ms定时器或是特殊数据寄存器D8099(高速环形计数器)，就可以以 1ms或是0.1ms为单位测量短时间的脉冲宽度。

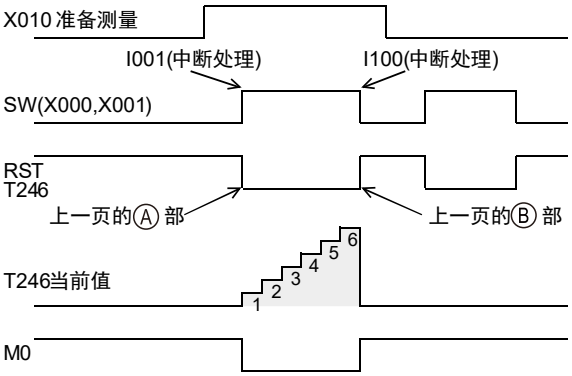
1. 使用累计型的1ms定时器测量短时间脉冲宽度的程序举例



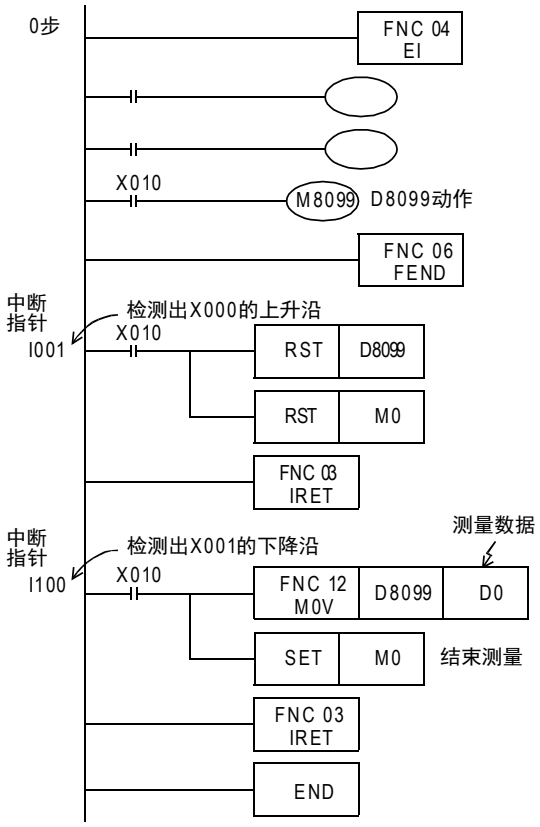




・时序图



2. 使用高速环形计数器测量短时间的脉冲宽度的程序举例(仅适用于A8/A8C可编程控制器)



使用EI指令允许中断描述程序

环形计数器动作

X000=ON：复位环形计数器，
开始测量

X001=OFF将环形计数器的值送到D0中，
结束测量

从特殊数据寄存器 M8099被驱动后的下一个运算周期开始，对0.1ms时钟进行增计数。
计数值超出32,767时，再次从0开始计算。

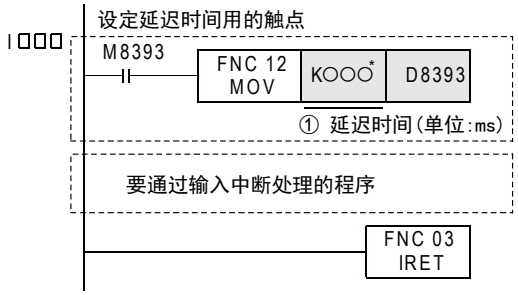
35.4 输入中断(通过外部信号中断)[使用延迟功能]

A8 A8C

1. 概要

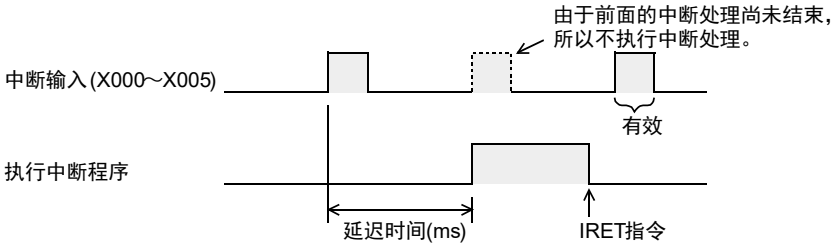
在输入中断中，有以1ms为单位的延迟执行中断子程序的功能。
通过执行以下格式的程序来指定要延迟的时间。
如使用这个延迟功能，那么在调整输入中断中使用的传感器的安装位置时，就可以进行与实际位置没有偏差的电气调节。

2. 编程要领



・指定延迟时间和程序
左图中的指定延迟时间的程序，
必须要编写在中断子程序的
开头位置中。
由于这个程序是模板程序，
所以届时只需要更改延迟时间（①）。
此外，这个时间指定（*），只可以
使用常数（K）或是数据寄存器（D）。
・中断程序的终止

3. 时序图



35.5 定时器中断(一定周期的中断)

35.5.1 定时器中断(一定周期的中断)

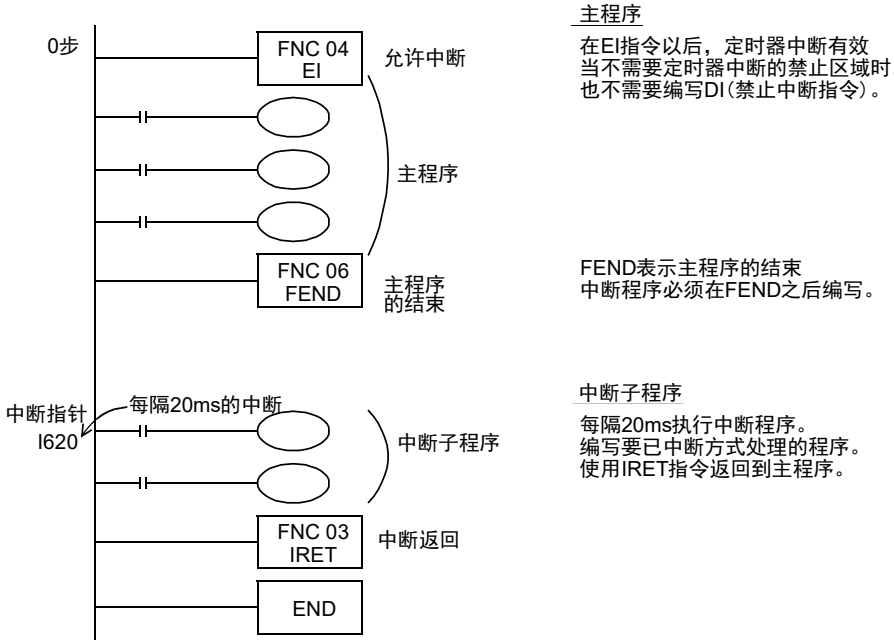
1. 概要

不受可编程控制器运算周期的影响，每隔10ms～99ms执行一次中断程序。

2. 用途

适用于当主程序的运算周期较长时，对特定程序进行高速处理，或者需要在顺控运算时间隔一定时间执行程序时的情况。

3. 基本程序(编程要领)



4. 定时器中断指针(3点)的编号及动作

I □ □ □

定时器时间10～99(ms)

定时器中断指针6,7,8

每隔指定的中断循环时间(10ms～99ms)，执行中断子程序。

在可编程控制器的运算周期以外，需要循环中断处理的控制中使用。

输入编号	中断周期(ms)	中断禁止标志位
I6□□	在指针名的□□中，输入10～99的整数。 例如:I610=每10ms的定时器中断	M8056*1
I7□□		M8057*1
I8□□		M8058*1

*1. 从RUN→STOP时清除

注意

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。

- 中断程序的处理时间较长时
- 主程序中使用了处理时间较长的指令时

5. 注意要点

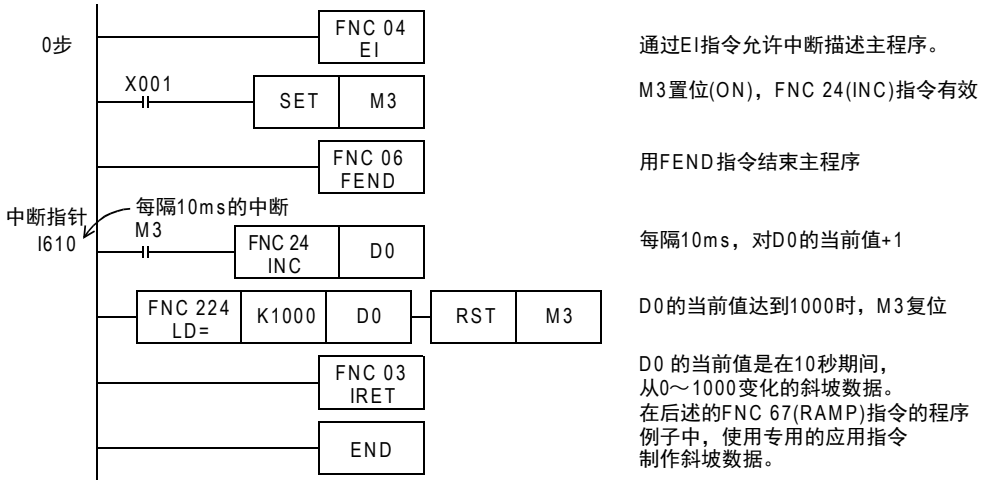
- 指针编号(I6、I7、I8)不能重复使用。
- 在程序中接通M8056～M8058后，其各自对应的定时器中断被禁止。

6. 程序举例

→ 与RAMP(FNC 67)、HKY(FNC 71)的应用指令组合使用的程序举例，请参考35.5.2项

每隔10ms对数据进行加法运算，并与设定值比较的程序举例。

1) 程序举例

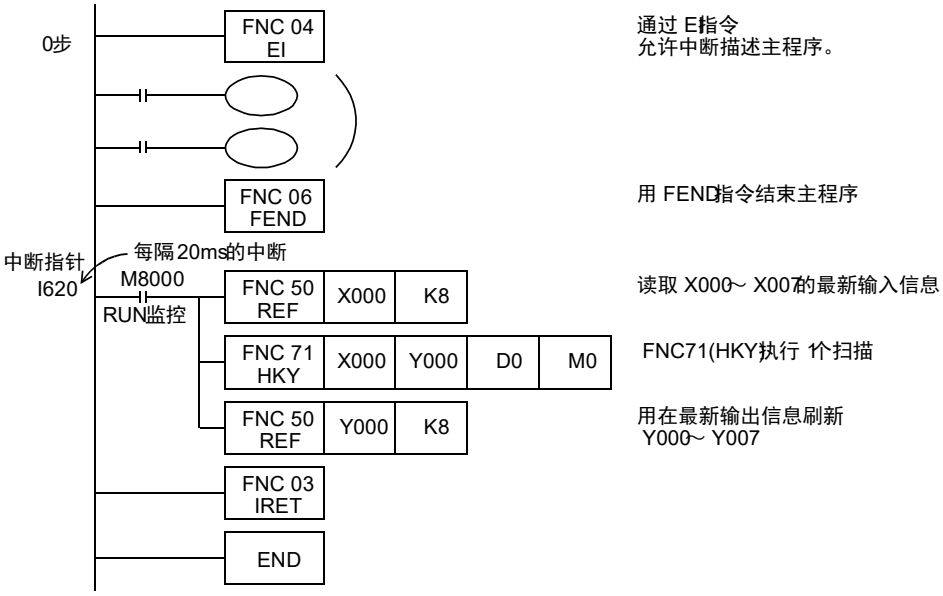


35.5.2 实用程序举例(应用指令的定时器中断程序)

RAMP(FNC 67)、HKY(FNC 71)、SEGL(FNC 74)、ARWS(FNC 75)、PR(FNC 77)指令是与扫描时间同步的，执行一连串动作的指令。
在这些指令中，由于会出现整体时间过长或在时间波动上有问题，所以使用定时器中断，使指令按照一定时间间隔执行。
此外，未使用定时器中断时，请使用恒定扫描模式。

1. HKY(FNC 71)指令的定时器中断处理

→ 关于HKY(FNC 71)指令，请参考15.2节

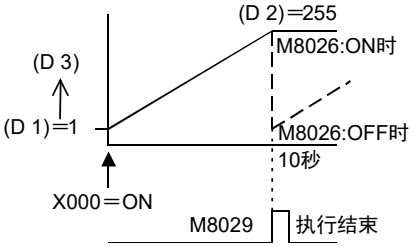


2. RAMP(FNC 67)指令的定时器中断处理

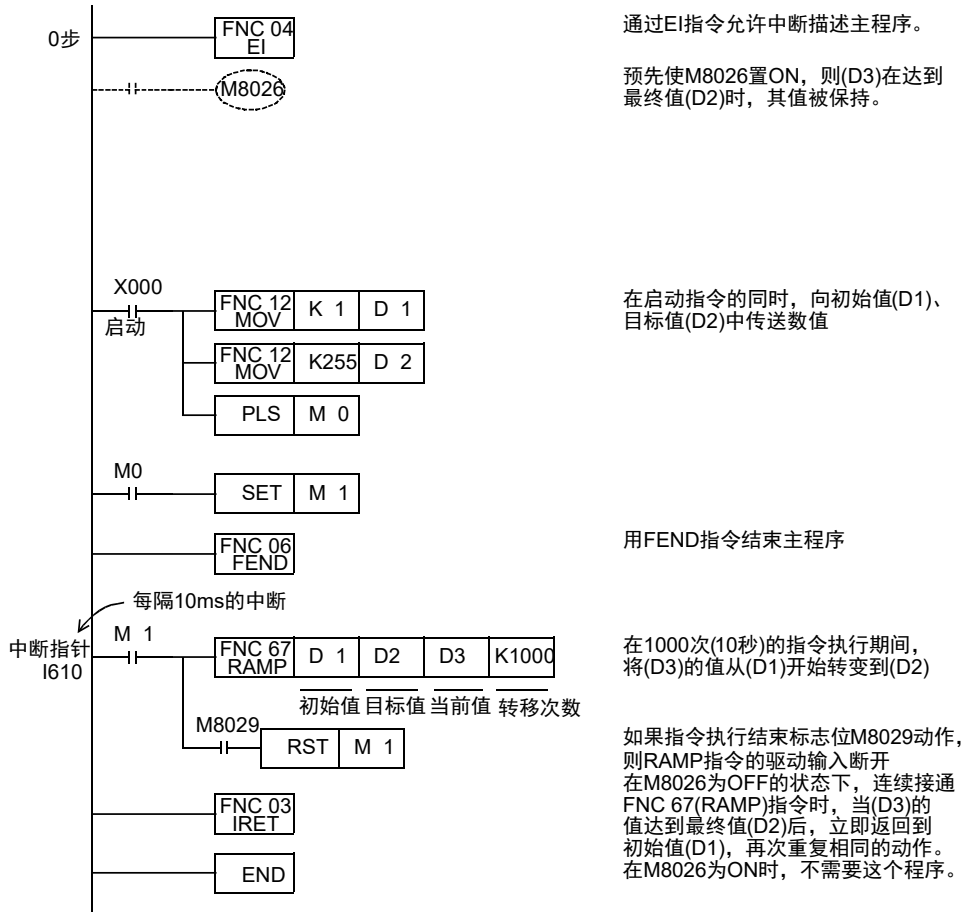
使用10ms的定时器中断，编写如下所示的斜坡信号输出回路。

→ 关于指令执行结束标志位M8029的使用方法，请参考6.5.2项
→ 关于RAMP(FNC 67)指令，请参考14.8节

1) 斜坡输出模型
D4被作为计算执行次数用的寄存器而占用。



2) 程序



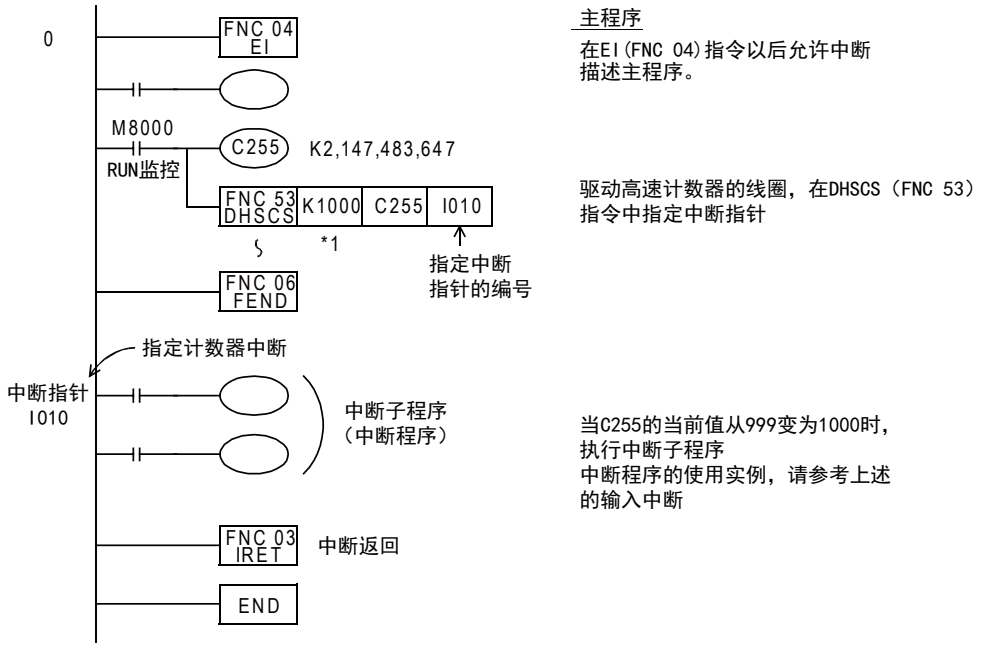
35.6 计数器中断 - 根据高速计数器的计数到位来中断

A8 A8C

1. 概要
使用高速计数器的当前值的中断。

2. 用途
与DHSCS(FNC 53)的比较置位指令一起使用，当高速计数器的当前值达到规定值时执行中断程序。

3. 基本程序(编程要领)



*1. 更改数据寄存器等中指定的比较值时，这个值在执行END指令时被更新。

4. 定时器中断指针(6点)的编号及动作

I 0 □ 0
↑计数器中断指针(1～6)

指针编号	中断禁止标志位
I010,I020,I030,I040,I050,I060	M8059*1

*1. 从RUN→STOP时清除

5. 使用高速计数器执行中断输出(Y、M)的ON/OFF
根据高速计数器的当前值，仅仅执行ON/OFF输出继电器(Y)和辅助继电器(M)的控制时，
使用DHSCS(FNC 53)、DHSCR(FNC 54)、DHSZ(FNC 55)指令可以简单地编程。

6. 注意要点
1) 指针编号的重复
不能重复使用指针编号。
2) 中断的禁止
在程序中使特殊辅助继电器M8059为ON后，则所有的计数器中断都被禁止。

35.7 脉冲捕捉功能[M8170～M8177]

执行FNC 04(EI)指令后，当输入继电器X000～X007从OFF变为ON时，特殊辅助继电器M8170～M8177通过中断处理被置位。

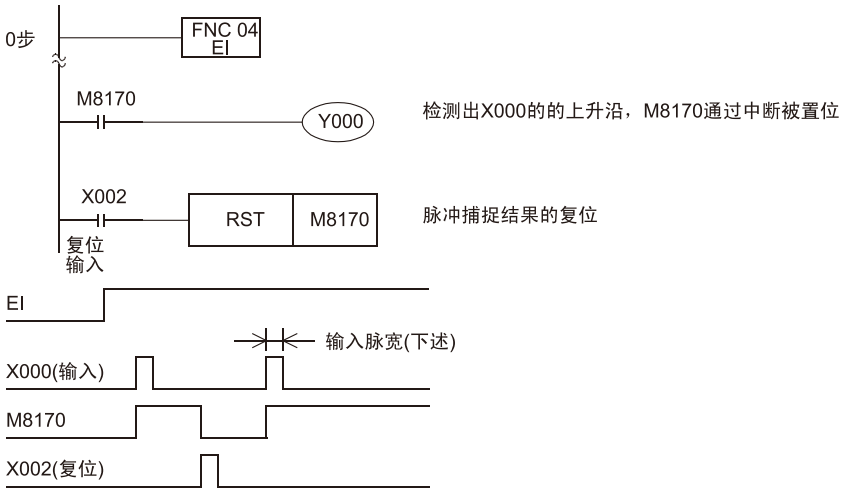
1. 输入编号和特殊辅助继电器的分配

脉冲捕捉输入	脉冲捕捉继电器
X000	M8170*1
X001	M8171*1
X002	M8172*1
X003	M8173*1
X004	M8174*1
X005	M8175*1
X006	M8176*1*2
X007	M8177*1*2

*1. 从STOP→RUN时清除

*2. 仅对应A8/A8C可编程控制器。

2. 程序举例



A8/A8C可编程控制器
[X000～X005]:5μs以上 *3
[X006,X007] :50μs以上

*3. 在5μs情况下使用脉冲捕捉功能时，以及用高速计数器读取50k～100kHz的响应频率的脉冲时，请如下所示设置。

- 接线长度请勿超出5m。
- 在输入端子中连接1.5kΩ(1W以上)的漏电阻， 与主机的输入电流相符合， 对方一侧机器的开集电极型晶体管输出的负载电流需要在20mA以上。

3. 使用上的注意事项

1) 为了再次读取输入，需要通过程序对被置位的软元件进行复位。
因此，被置位的软元件到被复位之前，不能读取新的输入。

2) 需要读取连续的短时间脉冲(输入信号)时，请使用外部输入中断功能或高速计数器功能。

3) 不需要调节滤波器的程序。

4) 与分别禁止中断用的辅助继电器M8050～M8055的动作无关而执行。

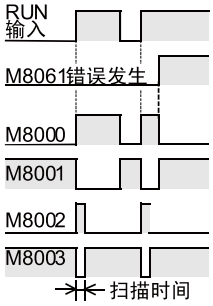
36. 特殊软元件的动作（M8000～，D8000～）

36.1 特殊软元件一览(M8000～,D8000～)

特殊辅助继电器(表中简称为特M)和特殊数据寄存器(表中简称为特D)的种类以及其功能如下所示。
此外，根据可编程控制器的系列不同，即使是同一软元件编号，有时候功能内容也可能有所不同，请务必注意。
未定义以及未记载的特殊辅助继电器和特殊数据寄存器为CPU占用的区域。
因此，请勿在顺控程序中使用。
此外，类似[M]8000、[D]8001的用[ ]框起的软元件，请不要在程序中执行驱动以及写入。

→ 关于特殊软元件的补充说明，请参考36.2节

36.1.1 特殊辅助继电器(M8000～M8511)

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
PLC状态							
[M]8000 RUN监控 a触点	 →参考36.2.1项	○	○	—	○	○	○
[M]8001 RUN监控 b触点		○	○	—	○	○	○
[M]8002 初始脉冲 a触点		○	○	—	○	○	○
[M]8003 初始脉冲 b触点		○	○	—	○	○	○
[M]8004 错误发生	• A8、A8C M8060、M8061、M8064、M8065、M8066、M8067中任意一个为ON时接通 • A1、A2、A2C、A8、A8C M8060、M8061、M8063、M8064、M8065、M8066、M8067中任意一个为ON时接通	○	○	D8004	○	○	○
[M]8005 电池电压低	当电池处于电压异常低时接通 →参考36.2.3项	○	○	D8005	—	—	—
[M]8006 电池电压低 锁存	检测出电池电压异常低时置位 →参考36.2.3项	○	○	D8006	—	—	—
[M]8007 检测出瞬间停止	检测出瞬间停止时，1个扫描为ON 即使M8007接通，如果电源电压降低的时间在D8008的时间以内时，可编程控制器的运行继续。 →参考36.2.4项	○	○	D8007 D8008	—	—	—
[M]8008 检测出停电中	检测出瞬间停电时置位，如果电源电压降低的时间超出D8008的时间，则M8008复位，可编程控制器的运行STOP(M8000=OFF)。 →参考36.2.4项	○	○	D8008	—	—	—
[M]8009 DC24V掉电	扩展单元或扩展电源单元的任意一个的DC24V掉电时接通	○	○	D8009	—	—	—

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
时钟							
[M]8010	不可以使用	—	—	—	○	○	○
[M]8011 10ms时钟	10ms周期的ON/OFF(ON: 5ms，OFF: 5ms) →参考36.2.6项	○	○	—	○	○	○
[M]8012 100ms时钟	100ms周期的ON/OFF(ON: 50ms，OFF: 50ms) →参考36.2.6项	○	○	—	○	○	○
[M]8013 1s时钟	1s周期的ON/OFF(ON: 500ms，OFF: 500ms) →参考36.2.6项	○	○	—	○	○	○
[M]8014 1min时钟	1min周期的ON/OFF(ON: 30s，OFF: 30s) →参考36.2.6项	○	○	—	○	○	○
M8015	停止计时以及预置 实时时钟用 →参考36.2.7项	○	○	—	○	○	○
M8016	时间读出后的显示被停止 实时时钟用 →参考36.2.7项	○	○	—	○	○	○
M8017	±30秒的修正 实时时钟用 →参考36.2.7项	○	○	—	○	○	○
[M]8018	检测出安装(一直为ON) 实时时钟用 →参考36.2.7项	○	○	—	○(一直为ON)		
M8019	实时时钟(RTC)错误 实时时钟用 →参考36.2.7项	○	○	—	○	○	○
标志位							
[M]8020 零位	加减法运算结果为0时接通 →关于使用方法，请参考6.5.2项	○	○	—	○	○	○
[M]8021 借位	减法运算结果超过最大的负值时接通 →关于使用方法，请参考6.5.2项	○	○	—	○	○	○
M8022 进位	加法运算结果发生进位时，或者移位结果发生溢出时接通 →关于使用方法，请参考6.5.2项	○	○	—	○	○	○
[M]8023	不可以使用	—	—	—	—	—	—
M8024*1	指定BMOV方向 (FNC 15)	○	○	—	—	○	○
M8025*2	HSC模式 (FNC 53～55)	○	○	—	—	—	—
M8026*2	RAMP模式 (FNC 67)	○	○	—	—	—	—
M8027*2	PR模式 (FNC 77)	○	○	—	—	—	—
M8028	100ms/10ms的定时器切换	—	—	—	○	—	—
	FROM/TO(FNC 78、79)指令执行过程中允许中断	○	○	—	—	—	—
[M]8029 指令执行结束	DSW(FNC 72)等的动作结束时接通 →关于使用方法，请参考6.5.2项	○	○	—	○	○	○

*1. 根据可编程控制器如下所示。
- A2/A2C可编程控制器中不被清除。
- A8/A8C可编程控制器中，从RUN→STOP时被清除。

*2. 根据可编程控制器如下所示。
- A8/A8C可编程控制器中，从RUN→STOP时被清除。

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
PLC模式							
M8030*1 电池LED 灭灯指示	驱动M8030后，即使电池电压低，可编程控制器面板上的LED也不亮灯。 →参考36.2.10项	○	○	—	—	—	—
M8031*1 非保持内存 全部清除	驱动该特殊M后，Y/M/S/T/C的ON/OFF映像区，以及T/C/D/特殊D*3，R*2的当前值被清除。 但是，文件寄存器(D)、扩展文件寄存器(ER)*2不清除。 →参考36.2.12项	○	○	—	○	○	○
M8032*1 保持内存 全部清除		○	○	—	○	○	○
M8033 内存保持 停止	从RUN到STOP时，映象存储区和数据存储区的内容按照原样保持。 →参考36.2.13项	○	○	—	○	○	○
M8034*1 禁止所有输出	可编程控制器的外部输出触点全部断开。 →参考36.2.14项	○	○	—	○	○	○
M8035 强制RUN模式	→详细内容请参考36.2.15项	○	○	—	○	○	○
M8036 强制RUN指令		○	○	—	○	○	○
M8037 强制STOP指令		○	○	—	○	○	○
[M]8038 参数的设定	通信参数设定的标志位 (设定简易PC之间的链接用) →请参考通信控制手册	○	○	D8176～D8180	○	○	○
M8039 恒定 扫描模式	M8039接通后，一直等待到D8039中指定的扫描时间到可编程控制器执行这样的循环运算。 →参考36.2.16项	○	○	D8039	○	○	○

*1. 在执行END指令时处理

*2. R、ER仅对应A8/A8C可编程控制器

*3. A1/A2/A2C可编程控制器中，特殊D不被清除。

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
步进梯形图・信号报警器(详细内容请参考ANS(FNC 46)、ANR(FNC 47)、IST(FNC 60)以及34復)							
M8040 禁止转移	驱动M8040时，禁止状态之间的转移。	○	○	—	○	○	○
[M]8041*1 转移开始	自动运行时，可以从初始状态开始转移。	○	○	—	○	○	○
[M]8042 启动脉冲	对应启动输入的脉冲输出。	○	○	—	○	○	○
M8043*1 原点回归结束	请在原点回归模式的结束状态中置位。	○	○	—	○	○	○
M8044*1 原点条件	请在检测出机械原点时驱动。	○	○	—	○	○	○
M8045 禁止所有输出复位	切换模式时，不执行所有输出的复位。	○	○	—	○	○	○
[M]8046*2 STL状态动作	当M8047接通时，S0～S899、S1000～S4095*3中任意一个为ON则接通	○	○	M8047	○	○	○
M8047*2 STL监控有效	驱动了这个特M后，D8040～D8047有效	○	○	D8040～D8047	○	○	○
[M]8048*2 信号报警器动作	当M8049接通时，S900～S999中任意一个为ON则接通	○	○	—	—	—	—
M 8049*1 信号报警器有效	驱动了这个特M时，D8049的动作有效	○	○	D8049 M8048	—	—	—

*1. 从RUN→STOP时清除

*2. 在执行END指令时处理

*3. S1000～S4095仅对应A8/A8C

禁止中断(详细内容请参考35.2.1项)							
M8050 (输入中断) I00□禁止*4	**禁止输入中断或定时器中断的特M接通时** 即使发生输入中断和定时器中断，由于禁止了相应的中断的接收，所以不处理中断程序。 例如，M8050接通时，由于禁止了中断I00□的接收，所以即使是在允许中断的程序范围内，也不处理中断程序。 **禁止输入中断或定时器中断的特M断开时** a) 发生输入中断或定时器中断时，接收中断。 b) 如果是用EI(FNC 04)指令允许中断时，会即刻执行中断程序。 但是，如用DI(FNC 05)指令禁止中断时，一直到用EI(FNC 04)指令允许中断为止，等待中断程序的执行。	○	○	—	○	○	○
M8051 (输入中断) I10□禁止*4		○	○	—	○	○	○
M8052 (输入中断) I20□禁止*4		○	○	—	○	○	○
M8053 (输入中断) I30□禁止*4		○	○	—	○	○	○
M8054 (输入中断) I40□禁止*4		○	○	—	○	○	○
M8055 (输入中断) I50□禁止*4		○	○	—	○	○	○
M8056 (定时器中断) I6□□禁止*4		○	○	—	—	—	—
M8057 (定时器中断) I7□□禁止*4		○	○	—	—	—	—
M8058 (定时器中断) I8□□禁止*4		○	○	—	—	—	—
M8059 计数器中断禁止*4	使用I010～I060的中断禁止	—	○	—	—	—	—

*4. 从RUN→STOP时清除

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
错误检测(详细内容，请参考37復)							
[M]8060	I/O构成错误	○	○	D8060	—	—	—
[M]8061	PLC硬件错误	○	○	D8061	○	○	○
[M]8062	PLC/PP通信错误	—	—	D8062	—	—	—
	串行通信错误0[通道0]	—	—	D8062	—	—	—
[M]8063*1*2	串行通信错误1[通道1]	○	○	D8063	○	○	○
[M]8064	参数错误	○	○	D8064	○	○	○
[M]8065	语法错误	○	○	D8065 D8069 D8314 D8315	○	○	○
[M]8066	回路错误	○	○	D8066 D8069 D8314 D8315	○	○	○
[M]8067*3	运算错误	○	○	D8067 D8069 D8314 D8315	○	○	○
M8068	运算错误锁存	○	○	D8068 D8312 D8313	○	○	○
M8069*4	I/O总线检测	○	○	—	—	—	—

*1. 根据可编程控制器如下所示。
— A1/A2/A2C可编程控制器中，从STOP→RUN时被清除。
— A8/A8C可编程控制器时，电源从OFF变为ON时清除

*2. A8/A8C可编程控制器串行通信错误2[通道2]为M8438。

*3. 从STOP→RUN时清除

*4. 驱动了M8069后，执行I/O总线检测。(详细内容请参考37復)

并联链接							
M 8070*5	并联链接 请在主站时驱动。	○	○	—	○	○	○
M 8071*5	并联链接 请在子站时驱动。	○	○	—	○	○	○
[M]8072	并联链接 运行过程中接通	○	○	—	○	○	○
[M]8073	并联链接 当M8070/M8071设定错误时接通	○	○	—	○	○	○

*5. 从STOP→RUN时清除

采样跟踪[A8/A8C/A2/A2C用]							
[M]8074	不可以使用	—	—	—	—	—	—
[M]8075	采样跟踪准备开始指令	○	○	D8075～D8098	—	—	—
[M]8076	采样跟踪执行开始指令	○	○		—	—	—
[M]8077	采样跟踪执行中监控	○	○		—	—	—
[M]8078	采样跟踪执行结束监控	○	○		—	—	—
[M]8079	采样跟踪系统区域	○	○		—	—	—
[M]8080～[M]8089	不可以使用	—	—	—	—	—	—

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
标志位							
[M]8090	BKCMP(FNC 194～199)指令 块比较信号	○	○	—	—	—	—
M8091	COMRD(FNC 182)、BINDA(FNC 261)指令 输出字符数切换信号	○	○	—	—	—	—
[M]8092	不可以使用	—	—	—	—	—	—
[M]8093		—	—	—	—	—	—
[M]8094		—	—	—	—	—	—
[M]8095		—	—	—	—	—	—
[M]8096		—	—	—	—	—	—
[M]8097		—	—	—	—	—	—
[M]8098		—	—	—	—	—	—
高速环形计数器							
M8099*1	高速环形计数器(0.1ms单位，16位)动作	○	○	D8099	—	—	—
[M]8100	不可以使用	—	—	—	—	—	—

*1. 在A2C中，M8099驱动后的END指令执行之后，0.1ms的高速环形计数器D8099动作。
在A8/A8C中，M8099驱动后，0.1ms的高速环形计数器D8099动作。

内存信息							
[M]8101	不可以使用	—	—	—	—	—	—
[M]8102		—	—	—	—	—	—
[M]8103		—	—	—	—	—	—
[M]8104	安装有功能扩展存储器时接通	—	—	D8104 D8105	—	—	—
[M]8105	在存储器盒写入中时接通	○	○	—	—	—	—
[M]8106	不可以使用	—	—	—	—	—	—
[M]8107	软元件注释登录的确认	○	○	D8107	—	—	—
[M]8108	不可以使用	—	—	—	—	—	—

输出刷新错误(详细内容请参考37復)							
[M]8109	输出刷新错误	○	○	D8109	—	—	—
[M]8110	不可以使用	—	—	—	—	—	—
[M]8111		—	—	—	—	—	—

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
功能扩展板[A1/A2专用]							
M8112	A2-4EX-BD: BX0的输入	—	—	—	○	○	—
	A2-2AD-BD: 通道1的输入模式切换	—	—	D8112	○	○	—
M8113	A2-4EX-BD: BX1的输入	—	—	—	○	○	—
	A2-2AD-BD: 通道2的输入模式切换	—	—	D8113	○	○	—
M8114	A2-4EX-BD: BX2的输入	—	—	—	○	○	
	A2-1DA-BD: 输出模式的切换	—	—	D8114	○	○	—
M8115	A2-4EX-BD: BX3的输入	—	—	—	○	○	—
M8116	A2-2EYT-BD: BY0的输出	—	—	—	○	○	—
M8117	A2-2EYT-BD: BY1的输出	—	—	—	○	○	—
[M]8118	不可以使用	—	—	—	—	—	—
[M]8119		—	—	—	—	—	—
RS(FNC 80)・计算机链接[通道1](详细内容请参考通信控制手册)							
[M]8120	不可以使用	—	—	—	—	—	—
[M]8121*1	RS(FNC 80)指令 发送待机标志位	○	○	—	○	○	○
M8122*1	RS(FNC 80)指令 发送请求	○	○	D8122	○	○	○
M8123*1	RS(FNC 80)指令 接收结束标志位	○	○	D8123	○	○	○
[M]8124	RS(FNC 80)指令 载波的检测标志位	○	○	—	○	○	○
[M]8125	不可以使用	—	—	—	—	—	—
[M]8126	计算机链接[通道1] 全局ON	○	○	D8127 D8128 D8129	○	○	○
[M]8127	计算机链接[通道1] 下位通信请求(ON Demand)发送中	○	○		—	—	—
M8128	计算机链接[通道1] 下位通信请求(ON Demand)错误标志位	○	○		○	○	○
M8129	计算机链接[通道1] 下位通信请求(ON Demand)字/字节的切换 RS(FNC 80)指令 判断超时的标志位	○	○		○	○	○

*1. 从RUN→STOP时，或是RS指令OFF时清除

高速计数器比较・高速表格・定位[定位为A1/A2/A2C用]							
M8130	HSZ(FNC 55)指令 表格比较模式	○	○	D8130	—	—	—
[M]8131	同上的执行结束标志位	○	○		—	—	—
M8132	HSZ(FNC 55)、PLSY(FNC 57)指令 速度模型模式	○	○	D8131 ～ D8134	—	—	—
[M]8133	同上的执行结束标志位	○	○		—	—	—
[M]8134	不可以使用	—	—	—	—	—	—
[M]8135		—	—	—	—	—	—
[M]8136		—	—	—	—	—	—
[M]8137		—	—	—	—	—	—
[M]8138	HSCT(FNC 280)指令 指令执行结束标志位	○	○	D8138	—	—	—
[M]8139	HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55) HSCT(FNC 280)指令 高速计数器比较指令执行中	○	○	D8139	—	—	—
M8140	ZRN(FNC 156)指令 CLR信号输出功能有效	—	—	—	○	○	○
[M]8141	不可以使用	—	—	—	—	—	—
[M]8142		—	—	—	—	—	—
[M]8143		—	—	—	—	—	—
[M]8144		—	—	—	—	—	—
M8145	[Y000] 脉冲输出停止指令	—	—	—	○	○	○
M8146	[Y001] 停止脉冲输出的指令	—	—	—	○	○	○
[M]8147	[Y000] 脉冲输出中的监控(BUSY/READY)	—	—	—	○	○	○
[M]8148	[Y001] 脉冲输出中的监控(BUSY/READY)	—	—	—	○	○	○
[M]8149	不可以使用	—	—	—	—	—	—

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
变频器通信功能(详细内容，请参考通信控制手册)							
[M]8150	不可以使用	—	—	—	—	—	—
[M]8151*1	变频器通信中[通道1]	○	○	D8151	—	—	—
[M]8152*1	变频器通信错误[通道1]	○	○	D8152	—	—	—
[M]8153*1	变频器通信错误[的锁定[通道1]	○	○	D8153	—	—	—
[M]8154*1	IVBWR(FNC 274)指令错误[通道1]	○	○	D8154	—	—	—
[M]8154	在每个EXTR(FNC 180)指令中被定义	—	—	—	—	—	—
[M]8155	通过EXTR(FNC 180)指令使用通信端口时	—	—	D8155	—	—	—
[M]8156*1	变频器通信中[通道2]	○	○	D8156	—	—	—
[M]8156	EXTR(FNC 180)指令中，发生通信错误或是参数错误	—	—	D8156	—	—	—
[M]8157*1	变频器通信错误[通道2]	○	○	D8157	—	—	—
	在EXTR(FNC 180)指令中发生过的通信错误被锁定	—	—	D8157	—	—	—
[M]8158*1	变频器通信错误的锁存[通道2]	○	○	D8158	—	—	—
[M]8159*1	IVBWR(FNC 274)指令错误[通道2]	○	○	D8159	—	—	—

*1. 从STOP→RUN时清除

扩展功能							
M8160*2	XCH(FNC 17)的SWAP功能	○	○	—	—	—	—
M8161*2*3	8位处理模式	○	○	—	○	○	○
M8162	高速并联链接模式	○	○	—	○	○	○
[M]8163	不可以使用	—	—	—	—	—	—
M8164*2	FROM(FNC 78)、TO(FNC 79)指令 传送点数可改变模式	—	—	D8164	—	—	—
M8165*2	SORT2(FNC 149)指令 降序排列	○	○	—	—	—	—
[M]8166	不可以使用	—	—	—	—	—	—
M8167*2	HKY(FNC 71)指令 处理HEX数据的功能	○	○	—	—	—	—
M8168*2	SMOV(FNC 13) 处理HEX数据的功能	○	○	—	—	—	—
[M]8169	不可以使用	—	—	—	—	—	—

*2. 从RUN→STOP时清除

*3. 适用于ASC(FNC 76)、RS(FNC 80)、ASCI(FNC 82)、HEX(FNC 83)、CCD(FNC 84)、CRC(FNC 188)指令*8

*4. CRC(FNC 188)指令仅对应A8/A8C可编程控制器

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
脉冲捕捉(详细内容，请参考35.7节)							
M8170*1	输入X000 脉冲捕捉	○	○	—	○	○	○
M8171*1	输入X001 脉冲捕捉	○	○	—	○	○	○
M8172*1	输入X002 脉冲捕捉	○	○	—	○	○	○
M8173*1	输入X003 脉冲捕捉	○	○	—	○	○	○
M8174*1	输入X004 脉冲捕捉	○	○	—	○	○	○
M8175*1	输入X005 脉冲捕捉	○	○	—	○	○	○
M8176*1	输入X006 脉冲捕捉	—	○	—	—	—	—
M8177*1	输入X007 脉冲捕捉	—	○	—	—	—	—

*1. 从STOP→RUN时清除
A2C/A8/A8C可编程控制器…需要EI(FNC 04)指令。
A1/A2/A2C可编程控制器……不需要EI(FNC 04)指令。

通信端口的通道设定(详细内容，请参考通信控制手册)							
M8178	并联链接 通道切换(OFF: 通道1，ON: 通道2)	○	○	—	—	—	—
M8179	简易PC间链接 通道切换*2	○	○	—	—	—	—

*2. 通过判断是否需要在设定用程序中编程，来指定要使用的通道。

→关于设定用程序，请参考通信控制手册

- 通道1: 不编程
- 通道2: 编程

简易PC间链接(详细内容，请参考通信控制手册)							
[M]8180	不可以使用	—	—	—	—	—	—
[M]8181		—	—	—	—	—	—
[M]8182		—	—	—	—	—	—
[M]8183*3	数据传送顺控错误(主站)	○	○	D8201～D8218	(M504)	○	○
[M]8184*3	数据传送顺控错误(1号站)	○	○		(M505)	○	○
[M]8185*3	数据传送顺控错误(2号站)	○	○		(M506)	○	○
[M]8186*3	数据传送顺控错误(3号站)	○	○		(M507)	○	○
[M]8187*3	数据传送顺控错误(4号站)	○	○		(M508)	○	○
[M]8188*3	数据传送顺控错误(5号站)	○	○		(M509)	○	○
[M]8189*3	数据传送顺控错误(6号站)	○	○		(M510)	○	○
[M]8190*3	数据传送顺控错误(7号站)	○	○		(M511)	○	○
[M]8191*3	数据传送顺控的执行中	○	○		(M503)	○	○
[M]8192	不可以使用	—	—	—	—	—	—
[M]8193		—	—	—	—	—	—
[M]8194		—	—	—	—	—	—
[M]8195		—	—	—	—	—	—
[M]8196		—	—	—	—	—	—
[M]8197		—	—	—	—	—	—

*3. A1可编程控制器使用( )内的编号

高速计数器倍增的指定(详细内容请参考4.7.8项)							
M8198*4*5	C251、C252、C254用1倍/4倍的切换	—	○	—	—	—	—
M8199*4*5	C253、C255、C253(OP)用1倍/4倍的切换	—	○	—	—	—	—

*4. OFF: 1倍
ON: 4倍

*5. 从RUN→STOP时清除

编号・名称		动作・功能	适用机型					
			A8	A8C	对应特殊软元件	A1	A2	A2C
计数器增/减计数的计数方向(详细内容请参考4.6节)								
M8200	C200	M8□□□动作后，与其对应的C□□□变为递减模式。 • ON: 减计数动作 • OFF: 增计数动作	○	○	—	—	○	○
M8201	C201		○	○	—	—	○	○
M8202	C202		○	○	—	—	○	○
M8203	C203		○	○	—	—	○	○
M8204	C204		○	○	—	—	○	○
M8205	C205		○	○	—	—	○	○
M8206	C206		○	○	—	—	○	○
M8207	C207		○	○	—	—	○	○
M8208	C208		○	○	—	—	○	○
M8209	C209		○	○	—	—	○	○
M8210	C210		○	○	—	—	○	○
M8211	C211		○	○	—	—	○	○
M8212	C212		○	○	—	—	○	○
M 8213	C213		○	○	—	—	○	○
M8214	C214		○	○	—	—	○	○
M8215	C215		○	○	—	—	○	○
M8216	C216		○	○	—	—	○	○
M8217	C217		○	○	—	—	○	○
M8218	C218		○	○	—	—	○	○
M8219	C219		○	○	—	—	○	○
M8220	C220		○	○	—	—	○	○
M8221	C221		○	○	—	—	○	○
M8222	C222		○	○	—	—	○	○
M8223	C223		○	○	—	—	○	○
M8224	C224		○	○	—	—	○	○
M8225	C225		○	○	—	—	○	○
M8226	C226		○	○	—	—	○	○
M8227	C227		○	○	—	—	○	○
M8228	C228		○	○	—	—	○	○
M8229	C229		○	○	—	—	○	○
M8230	C230		○	○	—	—	○	○
M8231	C231		○	○	—	—	○	○
M8232	C232		○	○	—	—	○	○
M8233	C233		○	○	—	—	○	○
M8234	C234		○	○	—	—	○	○
高速计数器增/减计数的计数方向(详细内容请参考4.7节)								
M8235	C235	M8□□□动作后，与其对应的C□□□变为递减模式。 • ON: 减计数动作 • OFF: 增计数动作	○	○	—	○	○	○
M8236	C236		○	○	—	○	○	○
M8237	C237		○	○	—	○	○	○
M8238	C238		○	○	—	○	○	○
M8239	C239		○	○	—	○	○	○
M8240	C240		○	○	—	○	○	○
M8241	C241		○	○	—	○	○	○
M8242	C242		○	○	—	○	○	○
M8243	C243		○	○	—	○	○	○
M8244	C244		○	○	—	○	○	○
M8245	C245		○	○	—	○	○	○

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
高速计数器增/减计数器的监控(详细内容，请参考4.7节)							
[M]8246	C246	○	○	—	○	○	○
[M]8247	C247	○	○	—	○	○	○
[M]8248	C248	○	○	—	○	○	○
[M]8249	C249	○	○	—	○	○	○
[M]8250	C250	○	○	—	○	○	○
[M]8251	C251	○	○	—	○	○	○
[M]8252	C252	○	○	—	○	○	○
[M]8253	C253	○	○	—	○	○	○
[M]8254	C254	○	○	—	○	○	○
[M]8255	C255	○	○	—	○	○	○
[M]8256～[M]8259	不可以使用	—	—	—	—	—	—
模拟量特殊适配器(关于各模拟量特殊适配器的内容，请参考36.2.19项)							
M8260～M8269	第1台的特殊适配器*1	○	○	—	—	—	—
M8270～M8279	第2台的特殊适配器*1	○	○	—	—	—	—
M8280～M8289	第3台的特殊适配器*1	○	○	—	—	—	—
M8290～M8299	第4台的特殊适配器*1	○	○	—	—	—	—
标志位							
[M]8300～[M]8303	不可以使用	—	—	—	—	—	—
[M]8304 零位	乘除运算结果为0时，置ON	○	○	—	—	—	—
[M]8305	不可以使用	—	—	—	—	—	—
[M]8306 进位	除法运算结果溢出时，置ON	○	○	—	—	—	—
[M]8307～[M]8311	不可以使用	—	—	—	—	—	—

C□□□为递减模式时，与其对应的M□□□□为ON。
- ON: 减计数动作
- OFF: 增计数动作

*1. 从基本单元侧计算连接的模拟量特殊适配器的台数。
*2. 连接在14点/24点型基本单元的BD连接器，或40点/60点型基本单元的BD1连接器上的功能扩展板。
*3. 连接在40点/60点型基本单元的BD2连接器上的功能扩展板。

I/O非实际安装指定错误(详细内容，请参考37復)标志位							
M8312*4	实时时钟时间数据丢失错误	—	—	—	—	—	—
[M]8313～[M]8315	不可以使用	—	—	—	—	—	—
[M]8316*5	I/O非实际安装指定错误	○	○	D8316 D8317	—	—	—
[M]8317	不可以使用	—	—	—	—	—	—
[M]8318	BFM的初始化失败 从STOP→RUN时，对于用BFM初始化功能指定的特殊扩展模块/单元，发生针对其的FROM/TO错误时接通，发生错误的单元号被保存在D8318中，BFM号被保存在D8319中。	○	○	D8318 D8319	—	—	—
[M]8319～[M]8321	不可以使用	—	—	—	—	—	—
[M]8322	辨别A8C-32MT-LT与A8C-32MT-LT-2的机型 1: A8C-32MT-LT-2 0: A8C-32MT-LT	—	—	—	—	—	—

*4. 停电保持。执行清除M8312操作或重设时间数据，将自动清除。
*5. 在LD、AND、OR、OUT指令等的软元件编号中直接指定以及通过变址间接指定时，在输入输出的软元件编号未安装的情况下为ON。

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
I/O非实际安装指定错误(详细内容，请参考37復)标志位							
[M]8323	要求内置CC-Link/LT配置	—	○	—	—	—	—
[M]8324	内蔵CC-Link/LT配置结束	—	○	—	—	—	—
[M]8325～[M]8327	不可以使用	—	—	—	—	—	—
[M]8328	指令不执行	○	○	—	—	—	—
[M]8329	指令执行异常结束	○	○	—	—	—	—
定时时钟(详细内容，情参考24.3节)・定位[A8/A8C](详细内容，请参考定位控制手册)							
[M]8330	DUTY(FNC 186)指令 定时时钟的输出1	○	○	D8330	—	—	—
[M]8331	DUTY(FNC 186)指令 定时时钟的输出2	○	○	D8331	—	—	—
[M]8332	DUTY(FNC 186)指令 定时时钟的输出3	○	○	D8332	—	—	—
[M]8333	DUTY(FNC 186)指令 定时时钟的输出4	○	○	D8333	—	—	—
[M]8334	DUTY(FNC 186)指令 定时时钟的输出5	○	○	D8334	—	—	—
[M]8335	不可以使用	—	—	—	—	—	—
M8336*1	DVIT(FNC 151)指令 中断输入指定功能有效	○	○	D8336	—	—	—
[M]8337	不可以使用	—	—	—	—	—	—
M8338	PLSV(FNC 157)指令 加减速动作	○	○	—	—	—	—
[M]8339	不可以使用	—	—	—	—	—	—
[M]8340	[Y000] 脉冲输出中监控(ON: BUSY/OFF: READY)	○	○	—	—	—	—
M8341*1	[Y000] 清除信号输出功能有效	○	○	—	—	—	—
M8342*1	[Y000] 指定原点回归方向	○	○	—	—	—	—
M8343	[Y000] 正转限位	○	○	—	—	—	—
M8344	[Y000] 反转限位	○	○	—	—	—	—
M8345*1	[Y000] 近点DOG信号逻辑反转	○	○	—	—	—	—
M8346*1	[Y000] 零点信号逻辑反转	○	○	—	—	—	—
M8347*1	[Y000] 中断信号逻辑反转	○	○	—	—	—	—
[M]8348	[Y000] 定位指令驱动中	○	○	—	—	—	—
M8349*1	[Y000] 脉冲输出停止指令	○	○	—	—	—	—
[M]8350	[Y001] 脉冲输出中监控(ON: BUSY/OFF: READY)	○	○	—	—	—	—
M8351*1	[Y001] 清除信号输出功能有效	○	○	—	—	—	—
M8352*1	[Y001] 指定原点回归方向	○	○	—	—	—	—
M8353	[Y001] 正转限位	○	○	—	—	—	—
M8354	[Y001] 反转限位	○	○	—	—	—	—
M8355*1	[Y001] 近点DOG信号逻辑反转	○	○	—	—	—	—
M8356*1	[Y001] 零点信号逻辑反转	○	○	—	—	—	—
M8357*1	[Y001] 中断信号逻辑反转	○	○	—	—	—	—
[M]8358	[Y001] 定位指令驱动中	○	○	—	—	—	—
M8359*1	[Y001] 停止脉冲输出的指令	○	○	—	—	—	—
[M]8360	[Y002] 脉冲输出中监控(ON: BUSY/OFF: READY)	○	○	—	—	—	—
M8361*1	[Y002] 清除信号输出功能有效	○	○	—	—	—	—
M8362*1	[Y002] 指定原点回归方向	○	○	—	—	—	—
M8363	[Y002] 正转限位	○	○	—	—	—	—
M8364	[Y002] 反转限位	○	○	—	—	—	—
M8365*1	[Y002] 近点DOG信号逻辑反转	○	○	—	—	—	—
M8366*1	[Y002] 零点信号逻辑反转	○	○	—	—	—	—
M8367*1	[Y002] 中断信号逻辑反转	○	○	—	—	—	—
[M]8368	[Y002] 定位指令驱动中	○	○	—	—	—	—
M8369*1	[Y002] 脉冲输出停止指令	○	○	—	—	—	—

*1. 从RUN→STOP时清除

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
定位(A8可编程控制器)(详细内容请参考定位控制手册)							
[M]8370	[Y003] 脉冲输出中监控(ON: BUSY/OFF: READY)	○	—	—	—	—	—
M8371*1	[Y003] 清除信号输出功能有效	○	—	—	—	—	—
M8372*1	[Y003] 指定原点回归方向	○	—	—	—	—	—
M8373	[Y003] 正转限位	○	—	—	—	—	—
M8374	[Y003] 反转限位	○	—	—	—	—	—
M8375*1	[Y003] 近点DOG信号逻辑反转	○	—	—	—	—	—
M8376*1	[Y003] 零点信号逻辑反转	○	—	—	—	—	—
M8377*1	[Y003] 中断信号逻辑反转	○	—	—	—	—	—
[M]8378	[Y003] 定位指令驱动中	○	—	—	—	—	—
M8379	[Y003] 脉冲输出停止指令*1	○	—	—	—	—	—

*1. 从RUN→STOP时清除

高速计数器功能(详细内容请参考4.7.5项)							
[M]8380*2	C235、C241、C244、C246、C247、C249、C251、C252、C254的动作状态	○	○	—	—	—	—
[M]8381*2	C236的动作状态	○	○	—	—	—	—
[M]8382*2	C237、C242、C245的动作状态	○	○	—	—	—	—
[M]8383*2	C238、C248、C248(OP)、C250、C253、C255的动作状态	○	○	—	—	—	—
[M]8384*2	C239、C243的动作状态	○	○	—	—	—	—
[M]8385*3	C240的动作状态	○	○	—	—	—	—
[M]8386*2	C244(OP)的动作状态	○	○	—	—	—	—
[M]8387*2	C245(OP)的动作状态	○	○	—	—	—	—
[M]8388	高速计数器的功能变更用触点	○	○	—	—	—	—
M8389	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	○	○	—	—	—	—

*2. 从STOP→RUN时清除

中断程序							
[M]8393	设定延迟时间用的触点	○	○	D8393	—	—	—
[M]8394	HCMOV(FNC 189)中断程序用驱动触点	○	○	—	—	—	—
[M]8395	C254用功能切换软元件	—	—	—	—	—	—
[M]8396	不可以使用	—	—	—	—	—	—
[M]8397		—	—	—	—	—	—
环形计数器							
M8398	1ms的环形计数(32位)动作*3	○	○	D8398, D8399	—	—	—
[M]8399	不可以使用	—	—	—	—	—	—

*3. M8398驱动后的END指令执行之后，1ms的环形计数[D8399,D8398]动作。

编号・名称	动作・功能	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
RS2(FNC 87)[通道1](详细内容请参考通信控制手册)							
[M]8400	不可以使用	—	—	—	—	—	—
[M]8401*1	RS2(FNC 87)[通道1] 发送待机标志位	○	○	—	—	—	—
M8402*1	RS2(FNC 87)[通道1] 发送请求	○	○	D8402	—	—	—
M8403*1	RS2(FNC 87)[通道1] 接收结束标志位	○	○	D8403	—	—	—
[M]8404	RS2(FNC 87)[通道1] 载波的检测标志位	○	○	—	—	—	—
[M]8405	RS2(FNC 87)[通道1] 数据设定准备就绪(DSR)标志位	○	○	—	—	—	—
[M]8406	不可以使用	—	—	—	—	—	—
[M]8407		—	—	—	—	—	—
[M]8408		—	—	—	—	—	—
M8409	RS2(FNC 87)[通道1] 判断超时的标志位	○	○	—	—	—	—

*1. 从RUN→STOP时，或是RS2指令[通道1]OFF时清除

RS2(FNC 87)[通道2]计算机链接[通道2](详细内容请参考通信控制手册)							
[M]8410～[M]8420	不可以使用	—	—	—	—	—	—
[M]8421*2	RS2(FNC 87)[通道2] 发送待机标志位	○	○	—	—	—	—
M8422*2	RS2(FNC 87)[通道2] 发送请求	○	○	D8422	—	—	—
M8423*2	RS2(FNC 87)[通道2] 接收结束标志位	○	○	D8423	—	—	—
[M]8424	RS2(FNC 87)[通道2] 载波的检测标志位	○	○	—	—	—	—
[M]8425	RS2(FNC 87)[通道2] 数据设定准备就绪(DSR)标志位	○	○	—	—	—	—
[M]8426	计算机链接[通道2] 全局ON	○	○	D8427 D8428 D8429	—	—	—
[M]8427	计算机链接[通道2] 下位通信请求(On Demand)发送中	○	○		—	—	—
M8428	计算机链接[通道2] 下位通信请求(On Demand) 错误标志位	○	○		—	—	—
M8429	计算机链接[通道2] 下位通信请求(On Demand) 字/字节的切换 RS2(FNC 87)[通道2] 判断超时的标志位	○	○		—	—	—

*2. 从RUN→STOP时，或是RS2指令[通道2]OFF时清除

错误检测(详细内容，请参考37復)							
[M]8430～[M]8437	不可以使用	—	—	—	—	—	—
M8438	串行通信错误2[通道2]	○	○	D8438	—	—	—
[M]8439～[M]8448	不可以使用	—	—	—	—	—	—
[M]8449	特殊模块错误标志位	○	○	D8449	—	—	—
[M]8450～[M]8459	不可以使用	—	—	—	—	—	—

定位[A8/A8C](详细内容，请参考定位手册)							
M 8460	DVIT(FNC 151)指令[Y000] 用户中断输入指令	○	○	D8336	—	—	—
M8461	DVIT(FNC 151)指令[Y001] 用户中断输入指令	○	○	D8336	—	—	—
M8462	DVIT(FNC 151)指令[Y002] 用户中断输入指令	○	○	D8336	—	—	—
M8463	DVIT(FNC 151)指令[Y003] 用户中断输入指令	○*3	—	D8336	—	—	—
M8464	DSZR(FNC 150)指令、ZRN(FNC 156)指令 [Y000] 清除信号软元件指定功能有效	○	○	D8464	—	—	—
M8465	DSZR(FNC 150)指令、ZRN(FNC 156)指令 [Y001] 清除信号软元件指定功能有效	○	○	D8465	—	—	—
M8466	DSZR(FNC 150)指令、ZRN(FNC 156)指令 [Y002] 清除信号软元件指定功能有效	○	○	D8466	—	—	—
M8467	DSZR(FNC 150)指令、ZRN(FNC 156)指令 [Y003] 清除信号软元件指定功能有效	○*3	—	D8467	—	—	—
[M]8468～[M]8511	不可以使用	—	—	—	—	—	—

*3. 仅当A8可编程控制器中连接了2台A8-2HC-ADP时可以使用

36.1.2 特殊数据寄存器(D8000～D8511)

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
PLC状态							
D8000 看门狗 定时器	初始值如右侧所示(1ms单位) (电源ON时从系统ROM传送过来) 通过程序改写的值，在执行了END、WDT指令后生效 →参考36.2.2项	200	200	—	200	200	200
[D]8001 PLC类型以及 系统版本	2 4 1 0 0 BCD转换值 如右侧所示 版本V1.00	24	24	D8101*1	22	26	26
[D]8002 内存容量	• 2…2K步 • 4…4K步 • 8…8K步 • 16K步以上时 D8002 为 [8] 时，在 D8102 中输入 [16]、[32]、[64]。	○ 8	○ 8	D8102	○ 2	○ 8	○ 8
[D]8003 内存种类	保存内置存储器、RAM/EEPROM/EPROM盒的种类以及保护开关的ON/OFF状态。 内容 / 内存的种类 / 保护开关 00H / RAM存储器盒 / — 01H / EPROM存储器盒 / — 02H / EEPROM存储器盒或是快闪存储器盒 / OFF 0AH / EEPROM存储器盒或是快闪存储器盒 / ON 10H / 可编程控制器内置存储器 / —	○	○	—	○	○	○
[D]8004 错误M编号	8 0 6 0 BCD转换值 8060～8068(M8004 ON时)	○	○	M8004	○	○	○
[D]8005 电池电压	3 0 BCD转换值 (0.1V单位) 电池电压的当前值(例如：3.0V)	○	○	M8005	—	—	—
[D]8006 检测出电池电压低的等级	初始值 • A8/A8C可编程控制器: 2.7V(0.1V单位) (电源ON时从系统ROM传送过来)	○	○	M8006	—	—	—

*1. 对应特殊软元件的D8101仅指A8/A8C可控制编程器。
A1/A2/A2C可编程控制器中没有对应的特殊软元件。

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
PLC状态							
[D]8007 检测出瞬间停止	保存M8007的动作次数。 电源断开时清除。	○	○	M8007	—	—	—
D8008 检测出停电的时间	初始值*1 • A8可编程控制器：10ms(AC电源型) • A8C可编程控制器：5ms(DC电源型)	○	○	M8008	—	—	—
[D]8009 DC24V掉电单元号	DC24V掉电的扩展单元中的最小输入软元件编号	○	○	M8009	—	—	—

*1. 关于A8/A8C可编程控制器，请参考36.2.4项

时钟							
[D]8010 扫描当前值	0步开始的指令累计执行时间 (0.1ms单位) →参考36.2.5项	○ 同右	○ 同右	—	○ 在显示值中，还包括了驱动M8039时的恒定扫描运行的等待时间		
[D]8011 MIN扫描时间	扫描时间的最小值 (0.1ms单位) →参考36.2.5项			—			
[D]8012 MAX扫描时间	扫描时间的最大值 (0.1ms单位) →参考36.2.5项			—			
D8013 秒	0～59秒 (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8014 分	0～59分 (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8015 时	0～23小时 (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8016 日	1～31日 (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8017 月	1～12月 (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8018 年	西历2位数(0～99) (实时时钟用) →参考36.2.7项	○	○	—	○	○	○
D8019 星期	0(日)～6(六) (实时时钟用) →参考36.2.7项	○	○	—	○	○	○

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
输入滤波器							
D 8020 输入滤波器的调节	X000～X017*1输入滤波器值 (初始值: 10ms) ➞参考36.2.9项	○	○	—	○	○	○
[D]8021	不可以使用	—	—	—	—	—	—
[D]8022		—	—	—	—	—	—
[D]8023		—	—	—	—	—	—
[D]8024		—	—	—	—	—	—
[D]8025		—	—	—	—	—	—
[D]8026		—	—	—	—	—	—
[D]8027		—	—	—	—	—	—
变址寄存器Z0，V0							
[D]8028	Z0(Z)寄存器的内容*2	○	○	—	○	○	○
[D]8029	V0(V)寄存器的内容*2	○	○	—	○	○	○

*1. A8A/8C的16点型的基本单位可达X000～X007。
A2/A2C基本单位可达X000～X007。

*2. Z1～Z7、V1～V7的内容保存在D8182～D8195中

模拟电位器[A1/A2]							
[D]8030	模拟电位器VR1的值(0～255的整数值)	—	—	—	○	○	○
[D]8031	模拟电位器VR2的值(0～255的整数值)	—	—	—	○	○	○
恒定扫描							
[D]8032	不可以使用	—	—	—	—	—	—
[D]8033		—	—	—	—	—	—
[D]8034		—	—	—	—	—	—
[D]8035		—	—	—	—	—	—
[D]8036		—	—	—	—	—	—
[D]8037		—	—	—	—	—	—
[D]8038		—	—	—	—	—	—
D8039 恒定 扫描时间	初始值: 0ms(1ms单位) (电源ON时从系统ROM传送过来) 可以通过程序改写 ➞参考36.2.16项	○	○	M8039	○	○	○

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
步进梯形图・信号报警器							
[D]8040*1 ON状态编号1	状态S0～S899、S1000～S4095*2中为ON的状态的最小编号保存到D8040中，其次为ON的状态编号保存到D8041中。以下依次将运行的状态(最大8点)保存到D8047为止。	○	○	M8047	○	○	○
[D]8041*1 ON状态编号2		○	○		○	○	○
[D]8042*1 ON状态编号3		○	○		○	○	○
[D]8043*1 ON状态编号4		○	○		○	○	○
[D]8044*1 ON状态编号5		○	○		○	○	○
[D]8045*1 ON状态编号6		○	○		○	○	○
[D]8046*1 ON状态编号7		○	○		○	○	○
[D]8047*1 ON状态编号8		○	○		○	○	○
[D]8048	不可以使用	－	－	－	－	－	－
[D]8049*1 ON状态最小编号	M8049为ON时，保存信号报警继电器S900～S999中为ON的状态的最小编号。	○	○	M8049	○	○	○
[D]8050～[D]8059	不可以使用	－	－	－	－	－	－

*1. 在执行END指令时处理

*2. S1000～S4095仅指A8/A8C可编程控制器

错误检测(详细内容，请参考37復)							
[D]8060	I/O构成错误的非实际安装I/O的起始编号被编程的输入、输出软元件没有被安装时，写入其起始的软元件编号。 (例如)X020未安装时 1 0 2 0 BCD转换值 软元件编号*3 1:输入X 0:输出Y	○	○	M8060	－	－	－
[D]8061	PLC硬件错误的错误代码编号	○	○	M8061	○	○	○
[D]8062	PLC/PP通信错误的错误代码编号	○	○	M8062	－	－	－
	串行通信错误0[通道0]的错误代码编号	－	－	M8062	－	－	－
[D]8063*4	串行通信错误1[通道1]的错误代码编号	○	○	M8063	○	○	○
[D]8064	参数错误的错误代码编号	○	○	M8064	○	○	○
[D]8065	语法错误的错误代码编号	○	○	M8065	○	○	○
[D]8066	梯形图错误的错误代码编号	○	○	M8066	○	○	○
[D]8067*5	运算错误的错误代码编号	○	○	M8067	○	○	○
D8068*5	发生运算错误的步编号的锁存	○*6	○*6	M8068	○	○	○
[D]8069*5	M8065～7的错误步编号	○*7	○*7	M8065～M8067	○	○	○

*3. A8/A8C可编程控制器可达10～337。

*4. 根据可编程控制器，如下所示。
- A1/A2/A2C可编程控制器中，从STOP→RUN时被清除
- A8/A8C可编程控制器中，电源从OFF变为ON时清除

*5. 从STOP→RUN时清除

*6. 32K步以上时，在[D8313,D8312]中保存步编号

*7. 32K步以上时，在[D8315,D8314]中保存步编号

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
并联链接(详细内容请参考通信控制手册)							
[D]8070	判断并联链接错误的时间 500ms	○	○	—	○	○	○
[D]8071	不可以使用	—	—	—	—	—	—
[D]8072		—	—	—	—	—	—
[D]8073		—	—	—	—	—	—
采样跟踪*1[A8/A8C用]							
[D]8074	在计算机中使用了采样跟踪功能时，这些软元件是被可编程控制器系统占用的区域。*1	○	○	M8075～M8079	—	—	—
[D]8075		○	○		—	—	—
[D]8076		○	○		—	—	—
[D]8077		○	○		—	—	—
[D]8078		○	○		—	—	—
[D]8079		○	○		—	—	—
[D]8080		○	○		—	—	—
[D]8081		○	○		—	—	—
[D]8082		○	○		—	—	—
[D]8083		○	○		—	—	—
[D]8084		○	○		—	—	—
[D]8085		○	○		—	—	—
[D]8086		○	○		—	—	—
[D]8087		○	○		—	—	—
[D]8088		○	○		—	—	—
[D]8089		○	○		—	—	—
[D]8090		○	○		—	—	—
[D]8091		○	○		—	—	—
[D]8092		○	○		—	—	—
[D]8093		○	○		—	—	—
[D]8094		○	○		—	—	—
[D]8095		○	○		—	—	—
[D]8096		○	○		—	—	—
[D]8097		○	○		—	—	—
[D]8098		○	○		—	—	—

*1. 采样跟踪是外围设备使用的软元件。

高速环形计数器							
D8099	0～32767(0.1ms 单位，16 位) 的递增动作的环形计数器*2	○	○	M8099	—	—	—
[D]8100	不可以使用	—	—	—	—	—	—

*2. 驱动M8099后，随着END指令的执行，0.1ms的高速环形计数器D8099动作。

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
内存信息							
[D]8101 PLC类型以及系统版本	1 6 1 0 0 BCD转换值 如右侧所示 版本V1.00	16	16	—	—	—	—
[D]8102	2…2K步 4…4K步 8…8K步 16…16K步 32…32K步 64…64K步	○ 16 64	○ 16 64	—	○ 2	○ 8	○ 8
[D]8103	不可以使用	—	—	—	—	—	—
[D]8104	功能扩展内固有的机型代码	—	—	M8104	—	—	—
[D]8105	功能扩展内存的版本(Ver.1.00=100)	—	—		—	—	—
[D]8106	不可以使用	—	—	—	—	—	—
[D]8107	软元件注释登录数	○	○	M8107	—	—	—
[D]8108	特殊模块的连接台数	○	○	—	—	—	—
输出刷新错误(详细内容请参考37復)							
[D]8109	发生输出刷新错误的Y编号	○	○	M8109	—	—	—
[D]8110	不可以使用	—	—	—	—	—	—
[D]8111		—	—	—	—	—	—
功能扩展板A1/A2专用							
[D]8112	A2-2AD-BD: 通道1的数字值	—	—	M8112	○	○	—
[D]8113	A2-2AD-BD: 通道2的数字值	—	—	M8113	○	○	—
D8114	A2-1DA-BD: 要输出的数字值	—	—	M8114	○	○	—
[D]8115～[D]8119	不可以使用	—	—	—	—	—	—
RS(FNC 80)・计算机链接[通道1](详细内容请参考通信控制手册)							
D8120*1	RS(FNC 80)指令・计算机链接[通道1] 设定通信格式	○	○	—	○	○	○
D8121*1	计算机链接[通道1] 设定站号	○	○	—	○	○	○
[D]8122*2	RS(FNC 80)指令 发送数据的剩余点数	○	○	M8122	○	○	○
[D]8123*2	RS(FNC 80)指令 接收点数的监控	○	○	M8123	○	○	○
D8124	RS(FNC 80)指令 报头<初始值: STX>	○	○	—	○	○	○
D8125	RS(FNC 80)指令 报尾<初始值: ETX>	○	○	—	○	○	○
[D]8126	不可以使用	—	—	—	—	—	—
D8127	计算机链接[通道1] 指定下位通信请求(ON Demand)的起始编号	○	○	M8126～M8129	○	○	○
D8128	计算机链接[通道1] 指定下位通信请求(ON Demand)的数据数	○	○		○	○	○
D8129*3	RS(FNC 80)指令・计算机链接[通道1] 设定超时时间	○	○		○	○	○

*1. 停电保持

*2. 从RUN→STOP时清除

编号名称	寄存器的内容		适用机型					
			A8	A8C	对应特殊软元件	A1	A2	A2C
高速计数器比较・高速表格・定位[定位为A1/A2/A2C用]								
[D]8130	HSZ(FNC 55)指令高速比较表格计数器		○	○	M8130	—	—	—
[D]8131	HSZ(FNC 55)、PLSY(FNC 57)指令 速度型式表格计数器		○	○	M8132	—	—	—
[D]8132	低位	HSZ(FNC 55)、PLSY(FNC 57)指令 速度型式频率	○	○	M8132	—	—	—
[D]8133	高位							
[D]8134	低位	HSZ(FNC 55)、PLSY(FNC 57)指令 速度型式目标脉冲数	○	○	M8132	—	—	—
[D]8135	高位							
D8136	低位	PLSY(FNC 57)、PLSR(FNC 59)指令 输出到Y000和Y001的脉冲合计数的累计	○	○	—	○	○	○
D8137	高位							
[D]8138	HSCT(FNC 280)指令 表格计数器		○	○	M8138	—	—	—
[D]8139	HSCS(FNC 53)、HSCR(FNC 54)、HSZ(FNC 55)、HSCT(FNC 280)指令 执行中的指令数		○	○	M8139	—	—	—
D8140	低位	PLSY(FNC 57)、PLSR(FNC 59)指令输出到Y000的脉冲数的累计或是使用定位指令时的当前值地址	○	○	—	○	○	○
D8141	高位							
D8142	低位	PLSY(FNC 57)、PLSR(FNC 59)指令输出到Y001的脉冲数的累计或是使用定位指令时的当前值地址	○	○	—	○	○	○
D8143	高位							
[D]8144	不可以使用		—	—	—	—	—	—
D8145	ZRN(FNC 156)、DRVI(FNC 158)、DRVA(FNC 159)指令偏差速度 初始值:0		—	—	—	○	○	○
D8146	低位	ZRN(FNC 156)、DRVI(FNC 158)、DRVA(FNC 159)指令 最高速度 • A1/A2初始值:100000	—	—	—	○	○	○
D8147	高位							
D8148	ZRN(FNC 156)、DRVI(FNC 158)、DRVA(FNC 159)指令加减速时间 (初始值:100)		—	—	—	○	○	○
[D]8149	不可以使用		—	—	—	—	—	—

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
变频器通信功能(详细内容，请参考通信控制手册)							
D8150	变频器通信的响应等待时间[通道1]	○	○	—	—	—	—
[D]8151	变频器通信的通信中的步编号[通道1] 初始值: -1	○	○	M8151	—	—	—
[D]8152*1	变频器通信的错误代码[通道1]	○	○	M8152	—	—	—
[D]8153	变频器通信的错误步的锁存[通道1] 初始值: -1	○	○	M8153	—	—	—
[D]8154	IVBWR(FNC 274)指令中发生错误的参数编号[通道 初始值: -1	○	○	M8154	—	—	—
	EXTR(FNC 180)指令的响应等待时间	—	—	—	—	—	—
D8155	变频器通信的响应等待时间[通道2]	○	○	—	—	—	—
[D]8155	EXTR(FNC 180)指令的通信中的步编号	—	—	M8155	—	—	—
[D]8156	变频器通信的通信中的步编号[通道2] 初始值: -1	○	○	M8156	—	—	—
	EXTR(FNC 180)指令的错误代码	—	—	M8156	—	—	—
[D]8157*1	变频器通信的错误代码[通道2]	○	○	M8157	—	—	—
[D]8157	EXTR(FNC 180)指令的错误步(锁存) 初始值: -1	—	—	M8157	—	—	—
[D]8158	变频器通信的错误步锁存[通道2] 初始值: -1	○	○	M8158	—	—	—
[D]8159	IVBWR(FNC 274)指令中发生错误的参数编号[通道 初始值: -1	○	○	M8159	—	—	—

*1. 从STOP→RUN时清除

<table>
<tr><th rowspan="2">编号名称</th><th rowspan="2">寄存器的内容</th><th colspan="6">适用机型</th></tr>
<tr><th>A8</th><th>A8C</th><th>对应特殊软元件</th><th>A1</th><th>A2</th><th>A2C</th></tr>
<tr><td colspan="8">扩展功能</td></tr>
<tr><td>[D]8160</td><td rowspan="4">不可以使用</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8161</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8162</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8163</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>D8164</td><td>指定FROM(FNC 78)、TO(FNC 79)传送点数</td><td>—</td><td>—</td><td>M8164</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8165</td><td rowspan="4">不可以使用</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8166</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8167</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8168</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8169</td><td>限制存取的状态
<table>
<tr><th rowspan="2">当前值</th><th rowspan="2">存取的限制状态</th><th colspan="2">程序</th><th rowspan="2">监控</th><th rowspan="2">更改当前值</th></tr>
<tr><th>读出</th><th>写入</th></tr>
<tr><td>H**00*1</td><td>第2关键字未设定</td><td>○*2</td><td>○*2</td><td>○*2</td><td>○*2</td></tr>
<tr><td>H**10*1</td><td>禁止写入</td><td>○</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>H**11*1</td><td>禁止读出/写入</td><td>×</td><td>×</td><td>○</td><td>○</td></tr>
<tr><td>H**12*1</td><td>禁止所有的在线操作</td><td>×</td><td>×</td><td>×</td><td>×</td></tr>
<tr><td>H**20*1</td><td>解除关键字</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
</table></td><td>○</td><td>○</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
</table>

*1. **在系统上使用时被清除。

*2. 通过关键字的设定状态，未限制存取。

<table>
<tr><td colspan="8">简易PC间链接(设定)(详细内容请参考通信控制手册)</td></tr>
<tr><td>[D]8170</td><td rowspan="3">不可以使用</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8171</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8172</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>[D]8173</td><td>相应的站号的设定状态</td><td>○</td><td>○</td><td>—</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>[D]8174</td><td>通信子站的设定状态</td><td>○</td><td>○</td><td>—</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>[D]8175</td><td>刷新范围的设定状态</td><td>○</td><td>○</td><td>—</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D8176</td><td>设定相应站号</td><td>○</td><td>○</td><td rowspan="5">M8038</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D8177</td><td>设定通信的子站数</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D8178</td><td>设定刷新范围</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D8179</td><td>重试的次数</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>D8180</td><td>监视时间</td><td>○</td><td>○</td><td>○</td><td>○</td><td>○</td></tr>
<tr><td>[D]8181</td><td>不可以使用</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
</table>

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
变址寄存器Z1～Z7、V1～V7							
[D]8182	Z1寄存器的内容	○	○	—	○	○	○
[D]8183	V1寄存器的内容	○	○	—	○	○	○
[D]8184	Z2寄存器的内容	○	○	—	○	○	○
[D]8185	V2寄存器的内容	○	○	—	○	○	○
[D]8186	Z3寄存器的内容	○	○	—	○	○	○
[D]8187	V3寄存器的内容	○	○	—	○	○	○
[D]8188	Z4寄存器的内容	○	○	—	○	○	○
[D]8189	V4寄存器的内容	○	○	—	○	○	○
[D]8190	Z5寄存器的内容	○	○	—	○	○	○
[D]8191	V5寄存器的内容	○	○	—	○	○	○
[D]8192	Z6寄存器的内容	○	○	—	○	○	○
[D]8193	V6寄存器的内容	○	○	—	○	○	○
[D]8194	Z7寄存器的内容	○	○	—	○	○	○
[D]8195	V7寄存器的内容	○	○	—	○	○	○
[D]8196	不可以使用	—	—	—	—	—	—
[D]8197		—	—	—	—	—	—
[D]8198		—	—	—	—	—	—
[D]8199		—	—	—	—	—	—
简易PC间链接(监控)(详细内容请参考通信控制手册)							
[D]8200	不可以使用	—	—	—	—	—	—
[D]8201*1	当前的链接扫描时间	○	○	—	(D201)	○	○
[D]8202*1	最大的链接扫描时间	○	○	—	(D202)	○	○
[D]8203*1	数据传送顺控错误计数数(主站)	○	○	M8183～M8191	(D203)	○	○
[D]8204*1	数据传送顺控错误计数数(站1)	○	○		(D204)	○	○
[D]8205*1	数据传送顺控错误计数数(站2)	○	○		(D205)	○	○
[D]8206*1	数据传送顺控错误计数数(站3)	○	○		(D206)	○	○
[D]8207*1	数据传送顺控错误计数数(站4)	○	○		(D207)	○	○
[D]8208*1	数据传送顺控错误计数数(站5)	○	○		(D208)	○	○
[D]8209*1	数据传送顺控错误计数数(站6)	○	○		(D209)	○	○
[D]8210*1	数据传送顺控错误计数数(站7)	○	○		(D210)	○	○
[D]8211*1	数据传送错误代码(主站)	○	○		(D211)	○	○
[D]8212*1	数据传送错误代码(站1)	○	○		(D212)	○	○
[D]8213*1	数据传送错误代码(站2)	○	○		(D213)	○	○
[D]8214*1	数据传送错误代码(站3)	○	○		(D214)	○	○
[D]8215*1	数据传送错误代码(站4)	○	○		(D215)	○	○
[D]8216*1	数据传送错误代码(站5)	○	○		(D216)	○	○
[D]8217*1	数据传送错误代码(站6)	○	○		(D217)	○	○
[D]8218*1	数据传送错误代码(站7)	○	○		(D218)	○	○
[D]8219～[D]8259	不可以使用	—	—	—		—	—

*1. A1可编程控制器使用( )内的编号

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
模拟量特殊适配器(关于各模拟量特殊适配器的内容，请参考36.2.19项)							
D8260～D8269	第1台的特殊适配器*1	○	○	—	—	—	—
D8270～D8279	第2台的特殊适配器*1	○	○	—	—	—	—
D8280～D8289	第3台的特殊适配器*1	○	○	—	—	—	—
D8290～D8299	第4台的特殊适配器*1	○	○	—	—	—	—
模拟量特殊适配器、模拟功能扩展板(关于各个模拟量特殊适配器、模拟功能扩展板的对应，参考36.2.18项)							
D8260～D8269	第1台功能扩展板*2	—	—	—	—	—	—
D8270～D8279	第2台功能扩展板*3	—	—	—	—	—	—
D8280～D8289	第1台特殊适配器*1	—	—	—	—	—	—
D8290～D8299	第2台特殊适配器*1	—	—	—	—	—	—

*1. 从基本单元侧计算连接的模拟量特殊适配器的台数。
*2. 连接在14点/24点型基本单元的BD连接器，或40点/60点型基本单元的BD1连接器上的功能扩展板。
*3. 连接在40点/60点型基本单元的BD2连接器上的功能扩展板。

显示模块(A8-7DM)功能(详细内容请参考可编程控制器主机的硬件手册)							
D8300	显示模块用 控制软元件(D) 初始值: K-1	○	○	—	—	—	—
D8301	显示模块用 控制软元件(M) 初始值: K-1	○	○	—	—	—	—
[D]8302*4	设定显示语言 日语: K0 英语: K0以外	○	○	—	—	—	—
[D]8303	LCD对比度设定值 初始值: K0	○	○	—	—	—	—
[D]8304～[D]8309	不可以使用	—	—	—	—	—	—

*4. 停电保持

RND(FNC 184)								
[D]8310	低位	RND(FNC 184)生成随机数用的数据 初始值: K1	○	○	—	—	—	—
[D]8311	高位							
语法・回路・运算・I/O非实际安装的指定的错误步编号(详细内容请参考37復)								
D8312	低位	发生运算错误的步编号的锁存(32bit)	○	○	M8068	—	—	—
D8313	高位							
[D]8314*5	低位	M8065～M8067的错误步编号(32bit)	○	○	M8065～M8067	—	—	—
[D]8315*5	高位							
[D]8316	低位	指定(直接/通过变址的间接指定)了未安装的I/O编号的指令的步编号	○	○	M8316	—	—	—
[D]8317	高位							
[D]8318	BFM初始化功能发生错误的单元号		○	○	M8318	—	—	—
[D]8319	BFM初始化功能 发生错误的BFM号		○	○	M8318	—	—	—
[D]8320～[D]8328	不可以使用		—	—	—	—	—	—

*5. 从STOP→RUN时清除

编号名称	寄存器的内容		适用机型					
			A8	A8C	对应特殊软元件	A1	A2	A2C
定时时钟(详细内容请参考24.3节)・定位[A8/A8C](详情请参考定位控制手册)								
[D]8329	不可以使用		—	—	—	—	—	—
[D]8330	DUTY(FNC 186)指令 定时时钟输出1用扫描数的计数器		○	○	M8330	—	—	—
[D]8331	DUTY(FNC 186)指令 定时时钟输出2用扫描数的计数器		○	○	M8331	—	—	—
[D]8332	DUTY(FNC 186)指令 定时时钟输出3用扫描数的计数器		○	○	M8332	—	—	—
[D]8333	DUTY(FNC 186)指令 定时时钟输出4用扫描数的计数器		○	○	M8333	—	—	—
[D]8334	DUTY(FNC 186)指令 定时时钟输出5用扫描数的计数器		○	○	M8334	—	—	—
D8336	DVIT(FNC 151)用中断输入的指定初始值: —		○	○	M8336	—	—	—
[D]8337～[D]8339	不可以使用		—	—	—	—	—	—
D8340	低位	[Y000] 当前值寄存器初始值: 0	○	○	—	—	—	—
D8341	高位							
D8342	[Y000] 偏差速度初始值: 0		○	○	—	—	—	—
D8343	低位	[Y000] 最高速度初始值: 100000	○	○	—	—	—	—
D8344	高位							
D8345	[Y000] 爬行速度初始值: 1000		○	○	—	—	—	—
D8346	低位	[Y000] 原点回归速度初始值: 50000	○	○	—	—	—	—
D8347	高位							
D8348	[Y000] 加速时间初始值: 100		○	○	—	—	—	—
D8349	[Y000] 减速时间初始值: 100		○	○	—	—	—	—
D8350	低位	[Y001] 当前值寄存器初始值: 0	○	○	—	—	—	—
D8351	高位							
D8352	[Y001] 偏差速度初始值: 0		○	○	—	—	—	—
D8353	低位	[Y001] 最高速度初始值: 100000	○	○	—	—	—	—
D8354	高位							
D8355	[Y001] 爬行速度初始值: 1000		○	○	—	—	—	—
D8356	低位	[Y001] 原点回归速度初始值: 50000	○	○	—	—	—	—
D8357	高位							
D8358	[Y001] 加速时间初始值: 100		○	○	—	—	—	—
D8359	[Y001] 减速时间初始值: 100		○	○	—	—	—	—
D8360	低位	[Y002] 当前值寄存器初始值: 0	○	○	—	—	—	—
D8361	高位							
D8362	[Y002] 偏差速度初始值: 0		○	○	—	—	—	—
D8363	低位	[Y002] 最高速度初始值: 100000	○	○	—	—	—	—
D8364	高位							
D8365	[Y002] 爬行速度初始值: 1000		○	○	—	—	—	—
D8366	低位	[Y002] 原点回归速度初始值: 50000	○	○	—	—	—	—
D8367	高位							
D8368	[Y002] 加速时间初始值: 100		○	○	—	—	—	—
D8369	[Y002] 减速时间初始值: 100		○	○	—	—	—	—

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
定位(A8)(详细内容请参考定位控制手册)							
D8370 D8371	低位 高位 [Y003] 当前值寄存器 初始值: 0	○*2	—	—	—	—	—
D8372	[Y003] 偏差速度初始值: 0	○*2	—	—	—	—	—
D8373 D8374	低位 高位 [Y003] 最高速度 初始值: 100000	○*2	—	—	—	—	—
D8375	[Y003] 爬行速度初始值: 1000	○*2	—	—	—	—	—
D8376 D8377	低位 高位 [Y003] 原点回归速度 初始值: 50000	○*2	—	—	—	—	—
D8378	[Y003] 加速时间初始值: 100	○*2	—	—	—	—	—
D8379	[Y003] 减速时间初始值: 100	○*2	—	—	—	—	—
[D]8380～[D]8392	不可以使用	—	—	—	—	—	—
RS2(FNC 87)[通道0](详情请参考通信控制手册)							
D8370	RS2(FNC 87)[通道0]设定通信格式	—	—	—	—	—	—
D8371	不可以使用	—	—	—	—	—	—
[D]8372*1	RS2(FNC 87)[通道0] 发送数据的剩余点数	—	—	—	—	—	—
[D]8373*1	RS2(FNC 87)[通道0] 接收点数的监控	—	—	—	—	—	—
D8374	不可以使用	—	—	—	—	—	—
[D]8375	显示通信参数[通道0]	—	—	—	—	—	—
[D]8376～[D]8378	不可以使用	—	—	—	—	—	—
D8379	RS2(FNC 87)[通道0] 设定超时时间	—	—	—	—	—	—
D8380	RS2(FNC 87)[通道0] 报头1，2<初始值: STX>	—	—	—	—	—	—
D8381	RS2(FNC 87) [通道0] 报头3，4	—	—	—	—	—	—
D8382	RS2(FNC 87)[通道0] 报尾1，2<初始值: ETX>	—	—	—	—	—	—
D8383	RS2(FNC 87)[通道0] 报尾3，4	—	—	—	—	—	—
[D]8384	RS2(FNC 87)[通道0] 接收求和(接收数据)	—	—	—	—	—	—
[D]8385	RS2(FNC 87)[通道0] 接收求和(计算结果)	—	—	—	—	—	—
[D]8386	RS2(FNC 87)[通道0] 发送求和	—	—	—	—	—	—
[D]8387～[D]8388	不可以使用	—	—	—	—	—	—
[D]8389	显示动作模式[通道0]	—	—	—	—	—	—
[D]8390～[D]8392	不可以使用	—	—	—	—	—	—

*1. 从RUN→STOP时清除

*2. 仅当A8可编程控制器中连接了2台A8-2HC-ADP时可以使用

中断程序(详细内容参考35復)							
D8393	延迟时间 →参考35.4节	○	○	M8393	—	—	—
[D]8394	不可以使用	—	—	—	—	—	—
[D]8395		—	—	—	—	—	—
[D]8396	HC-Link/LT设定信息	—	○	—	—	—	—
[D]8397	不可以使用	—	—	—	—	—	—

编号名称	寄存器的内容		适用机型					
			A8	A8C	对应特殊软元件	A1	A2	A2C
环形计数器								
D8398	低位	0～2,147,483,647(1ms单位)的递增动作的环形计数*1	○	○	M8398	—	—	—
D8399	高位							
*1. M8398驱动后，随着END指令的执行，1ms的环形计数器[D8399，D8398]动作。								
RS2(FNC 87)[通道1](详细内容请参考通信控制手册)								
D8400	RS2(FNC 87)[通道1] 设定通信格式		○	○	—	—	—	—
[D]8401	不可以使用		—	—	—	—	—	—
[D]8402*2	RS2(FNC 87)[通道1] 发送数据的剩余点数		○	○	M8402	—	—	—
[D]8403*2	RS2(FNC 87)[通道1] 接收点数的监控		○	○	M8403	—	—	—
[D]8404	不可以使用		—	—	—	—	—	—
[D]8405	显示通信参数[通道1]		○	○	—	—	—	—
[D]8406	不可以使用		—	—	—	—	—	—
[D]8407			—	—	—	—	—	—
[D]8408			—	—	—	—	—	—
D8409	RS2(FNC 87)[通道1] 设定超时时间		○	○	—	—	—	—
D8410	RS2(FNC 87)[通道1] 报头1，2<初始值: STX>		○	○	—	—	—	—
D8411	RS2(FNC 87)[通道1] 报头3，4		○	○	—	—	—	—
D8412	RS2(FNC 87)[通道1] 报尾1，2<初始值: ETX>		○	○	—	—	—	—
D8413	RS2(FNC 87)[通道1] 报尾3，4		○	○	—	—	—	—
[D]8414	RS2(FNC 87)[通道1] 接收求和(接收数据)		○	○	—	—	—	—
[D]8415	RS2(FNC 87)[通道1] 接收求和(计算结果)		○	○	—	—	—	—
[D]8416	RS2(FNC 87)[通道1] 发送求和		○	○	—	—	—	—
[D]8417	不可以使用		—	—	—	—	—	—
[D]8418			—	—	—	—	—	—
[D]8419	显示动作模式[通道1]		○	○	—	—	—	—
RS2(FNC 87)[通道2]计算机链接[通道2](详细内容，请参考通信控制手册)								
D8420	RS2(FNC 87)[通道2] 设定通信格式		○	○	—	—	—	—
D8421	计算机链接[通道2] 设定站号		○	○	—	—	—	—
[D]8422*2	RS2(FNC 87)[通道2] 发送数据的剩余点数		○	○	M8422	—	—	—
[D]8423*2	RS2(FNC 87)[通道2] 接收点数的监控		○	○	M8423	—	—	—
[D]8424	不可以使用		—	—	—	—	—	—
[D]8425	显示通信参数[通道2]		○	○	—	—	—	—
[D]8426	不可以使用		—	—	—	—	—	—
D8427	计算机链接[通道2] 指定下位通信请求(On Demand)的起始编号		○	○	M8426～M8429	—	—	—
D8428	计算机链接[通道2] 指定下位通信请求(On Demand)的数据数		○	○		—	—	—
D8429	RS2(FNC 87)[通道2]计算机链接[通道2] 设定超时时间		○	○		—	—	—
D8430	RS2(FNC 87)[通道2] 报头1，2<初始值: STX>		○	○	—	—	—	—
D8431	RS2(FNC 87)[通道2] 报头3，4		○	○	—	—	—	—
D8432	RS2(FNC 87)[通道2] 报尾1，2<初始值: ETX>		○	○	—	—	—	—
D8433	RS2(FNC 87)[通道2] 报尾3，4		○	○	—	—	—	—
[D]8434	RS2(FNC 87)[通道2] 接收求和(接收数据)		○	○	—	—	—	—
[D]8435	RS2(FNC 87)[通道2] 接收求和(计算结果)		○	○	—	—	—	—
[D]8436	RS2(FNC 87)[通道2] 发送求和		○	○	—	—	—	—
[D]8437	不可以使用		—	—	—	—	—	—
*2. 从RUN→STOP时清除								

编号名称	寄存器的内容	适用机型					
		A8	A8C	对应特殊软元件	A1	A2	A2C
错误检测(详细内容，请参考37復)							
[D]8438	串行通信错误2[通道2]的错误代码编号	○	○	M8438	—	—	—
RS2(FNC 87)[通道2]计算机链接[通道2](详细内容，请参考通信控制手册)							
[D]8439	显示动作模式[通道2]	○	○	—	—	—	—
错误检测(详细内容请参考37復)							
[D]8440～[D]8448	不可以使用	—	—	—	—	—	—
[D]8449	特殊模块错误代码	○	○	M8449	—	—	—
[D]8450～[D]8459	不可以使用	—	—	—	—	—	—
定位[A8/A8C](详情请参考定位控制手册)							
[D]8460～[D]8463	不可以使用	—	—	—	—	—	—
D8464	DSZR(FNC 150)，ZRN(FNC 156)指令[Y000]指定清除信号软元件	○	○	M8464	—	—	—
D8465	DSZR(FNC 150)、ZRN(FNC 156)指令[Y001]指定清除信号软元件	○	○	M8465	—	—	—
D8466	DSZR(FNC 150)、ZRN(FNC 156)指令[Y002]指定清除信号软元件	○	○	M8466	—	—	—
D8467	DSZR(FNC 150),ZRN(FNC 156)指令[Y003]指定清除信号软元件	○*1	—	M8467	—	—	—
[D]8468～[D]8511	不可以使用	—	—	—	—	—	—

*1. 仅当A8可编程控制器中连接了2台A8-2HC-ADP时可以使用

36.2 特殊软元件(M8000～，D8000～)的补充

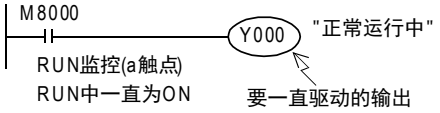
特殊软元件就是从可编程控制器运行的角度考虑,而预先准备的内置功能动作的软元件。下面说明其使用方法。

36.2.1 RUN监控，初始脉冲的使用[M8000～M8003]

1. RUN监控(M8000，M8001)

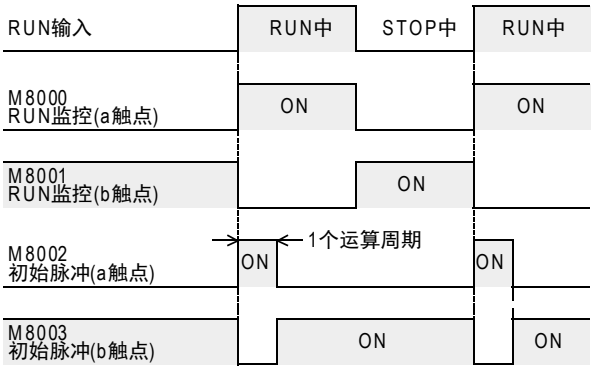
使用显示可编程控制器运行状态的RUN监控(M8000，M8001)，可以作为指令的驱动条件，也可以在显示“正常运行中”的外部显示中使用。

1) 程序举例



M8001在RUN中一直为OFF。

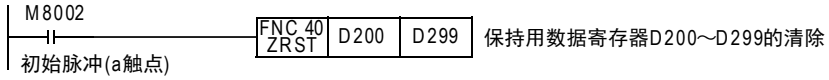
2) 标志位的动作时序



2. 初始脉冲(M8002，M8003)

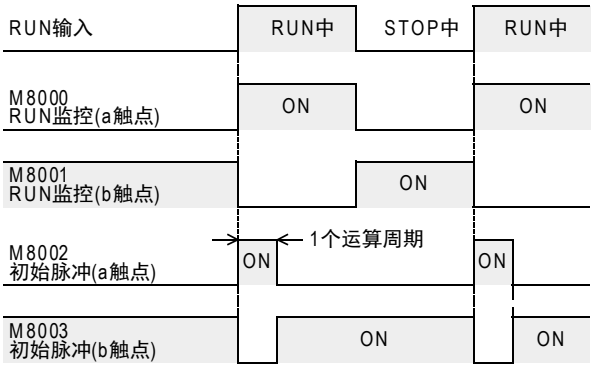
初始脉冲(M8002，M8003)在可编程控制器开始运行以后，仅瞬间(1个运算周期)为ON，或是为OFF。
这个脉冲可以作为程序的初始化或者写入规定值等程序中的初始设定信号使用。

1) 程序举例



M8003仅仅在RUN后的一瞬间(1个运算周期)为OFF。

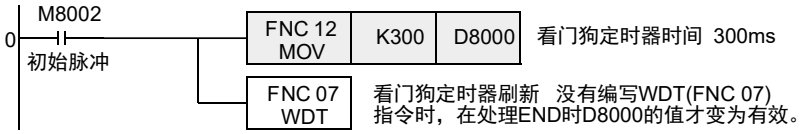
2) 标志位的动作时序



36.2.2 看门狗定时器时间[D8000]

看门狗定时器监视可编程控制器的运算(扫描)时间，在规定的时间内没有完成时，使[ERROR(ERR)] LED灯亮，所有的输出都变为OFF。
上电时从系统传送200ms的初始值，但如果执行的程序超出这个时间时，必须在程序中更改D8000的值。

1. 程序举例



2. 看门狗定时器错误的情况

如下所示的情况下，也会发生看门狗定时器错误的情况，所以请在起始步附近输入上述的程序，来延长看门狗定时器的时间，或是错开FROM/TO指令的执行时序。

1) 连接较多的特殊功能单元/模块时的注意事项
在连接较多台数的特殊功能单元/模块(定位、凸轮开关、链接、模拟量等)的系统构成中，可编程控制器运行时执行的缓冲存储区的初始化时间变长，运算时间会延长，有可能会发生看门狗定时器错误的情况。

2) 同时驱动较多的FROM/TO指令时的注意事项
当执行了多条FROM/TO 指令对多个缓冲存储区进行传送操作时，运算时间会延长，有可能会发生看门狗定时器错误的情况。

3) 高速计数器(软件计数器)较多时的注意事项
编写多个高速计数器同时对高频率脉冲进行计数时，运算时间会延长，有可能会发生看门狗定时器错误的情况。

3. 看门狗定时器复位的方法

不同于看门狗定时器时间本身的变更，使用WDT(FNC 07)指令，可以在顺控程序中对看门狗定时器进行中途复位。
在特定的顺控程序的运算时间变长的情况下，或者连接较多的特殊功能单元/模块时，建议使用 WDT(FNC 07)指令对看门狗定时器进行复位。

→ 关于WDT(FNC 07)指令，请参考8.8节

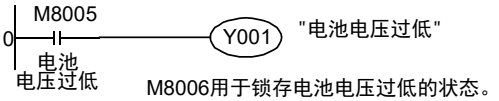
4. 更改看门狗定时器时间时的注意事项

看门狗定时器的时间最大可以设定到32,767ms，但是由此会导致运算异常的检测时序延迟。
因此，在运行没有问题的情况下，请置于初始值(200ms)的状态。

36.2.3 检测到电池电压过低[M8005，M8006]

检测内存备份用的电池电压过低的特殊软元件。
A8/A8C可编程控制器时，若检测到电池电压过低，BATT(BAT) LED灯亮。

程序举例



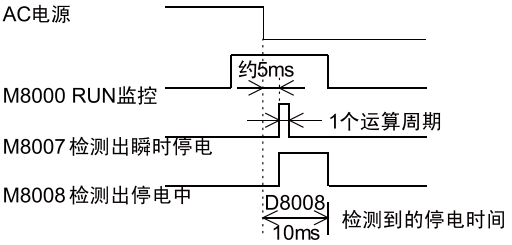
M8006用于锁存电池电压过低的状态。

36.2.4 停电检测时间[D8008，M8008，M8007]

1. A8可编程控制器(AC电源型)

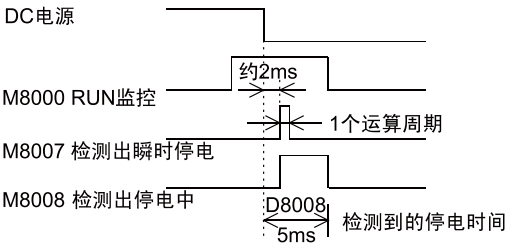
A8可编程控制器(AC电源型)允许瞬时停电的时间如下所示。

电源电压	允许瞬时停电时间
AC 100V	10ms
AC 200V	可以设定10ms～100ms在D8008中设定。 初始值: 10(ms)



2. A8/A8C可编程控制器(DC电源型)

A8/A8C可编程控制器(DC电源型)允许瞬时停电的时间为5ms。
请勿对停电检测时间D8008进行写入。



36.2.5 运算时间(监控)[D8010～D8012]

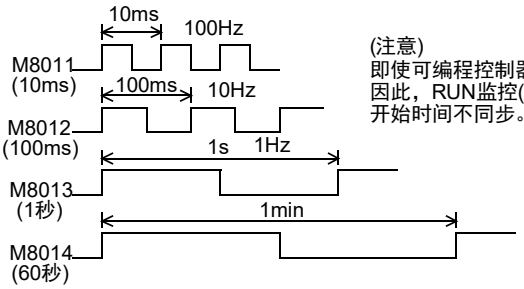
在D8010～D8012中分别保存有可编程控制器的扫描时间(运算时间)的当前、最小、最大值。(0.1ms单位)
此外，使用了恒定扫描功能时，这些值中都包含了恒定扫描时间的等待时间。

D8010 : 当前值
D8011 : 最小值
D8012 : 最大值

可以通过外围设备监控这些软元件的数值。

36.2.6 内部时钟[M8011～M8014]

具备下面4种内部时钟，在可编程控制器通电以后保持振荡。



(注意)
即使可编程控制器停止时，时钟也保持运作。
因此，RUN监控(M8000)的上升沿和时钟的开始时间不同步。

36.2.7 实时时钟[M8015～M8019，D8013～D8019]

1. 特殊辅助继电器(M8015～M8019)，特殊数据寄存器(D8013～D8019)的分配

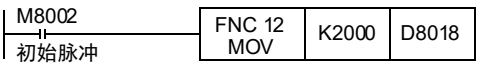
编号	名称	动作功能
M8015	时钟停止以及校时	ON时，时钟停止 在ON→OFF的边沿写入时间，再次动作。
M8016	显示时间的停止	ON时，停止显示时间(计时仍然动作)
M8017	±30秒的修正	在OFF→ON的边沿对秒进行修正。 (秒为0～29时，秒变为0。此外，秒为30～59秒时，进位到分钟，秒变为0)
M8018	安装检测	一直为ON
M8019	RTC错误	校准时间时，当特殊数据寄存器的数据超出设定范围时为ON。

编号	名称	设定值的范围	动作功能
D8013	秒	0～59	用于写入校时的初始值，或者读出当前时间。 • D8018(年)可以切换到公历4位数模式。 此外，显示公历4位数时，只能显示1980～2079年。 • 时钟精度: ±45秒/月(25°C时) • 闰年修正: 有
D8014	分	0～59	
D8015	时	0～23	
D8016	日	1～31	
D8017	月	1～12	
D8018	年	00～99(公历后2位数)	
D8019	星期	0～6(对应日～六)	

→ 实时时钟的校时方法，请参考36.2.8项

2. 年的公历4位数切换　　4位数处理时，请增加如下所示的程序。

将年作为公历
D8018在可编程控制器RUN后第2个运算周期以后，作为公历4位数动作。



1) 可编程控制器通常在公历后2位数的模式下运行。
可编程控制器运行后，执行上述指令，仅需在1个运算周期对 D8018(年) 传送[K2000(固定值)]，就可以切换为4位数模式。

2) 请在每次运行可编程控制器时执行这个程序。
此外，即使传送了K2000也仅仅是将显示变为公历4位数，对当前日期时间没有影响。

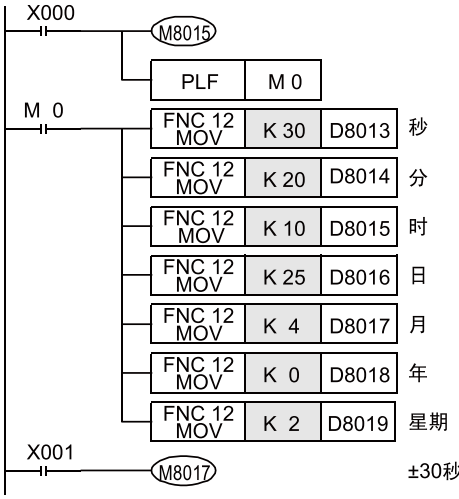
3) 公历4位数模式下，设定值[80～99]相当于[1980年～1999年]，[00～79]相当于[2000年～2079年]。
例如: 80=1980年，99=1999年，00=2000年，79=2079年

2. 通过程序更改的方法

1) 使用校时专用的TWR(FNC 167)指令的方法

→ 详细内容，参考21.8节

2) 不使用校时专用的应用指令的方法
不使用写入时间的专用指令TWR(FNC 167)时，也可以通过如下所示的程序进行设定。



内为设定项目。

例如： 2000年4月25日（二）
10时20分30秒

设定时，请设定比正确时间早2～3分钟的时间，
将左边的程序写入可编程控制器后运行，
并使X000置ON。
当达到正确时间时，将输入开关X000
从ON→OFF后，时间被设定，开始计时动作。

±30秒的修正
每当X001从OFF→ON,0～29秒时，
变为「0秒」， 30～59秒时，则「进位到分钟」

校准时的要点

a) 在M8015为OFF的状态下，不能更改日期时间的寄存器。
请务必在M8015为ON的状态下输入新的日期时间。

b) 输入设定时间时，设定比当前时间快几分钟的时间。
在达到设定时间的时候，使M8015从ON变到OFF，则新时间变为有效，且计时动作开始。

c) 在输入了表示不可能的日期时间(例如:2006年2月30日)的数值时，不能更改时间。
请重新输入正确的日期时间后操作。

3. 用编程工具变更的方法

1) 开始更改时间
强制置位(ON)M8015。

2) 在编程工具的软元件监控功能中，使用数据寄存器的当前值更改功能，对各数据寄存器输入相应的日期时间(快几分钟前的时间)。

3) 完成更改时间
在达到设定时间的时候，强制复位(OFF)M8015。

校准时的要点

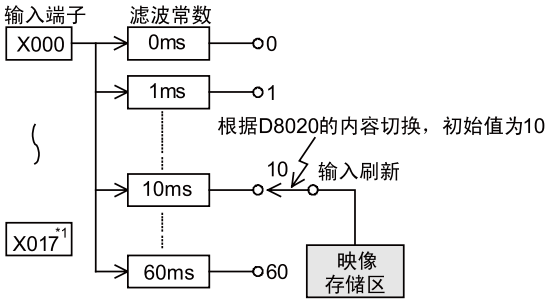
a) 在M8015为OFF的状态下，不能更改日期时间的寄存器。
请务必在M8015为ON的状态下输入新的日期时间。

b) 输入设定时间时，设定比当前时间快几分钟的时间。
在达到设定时间的时候，使M8015从ON变到OFF，则新时间变为有效，且计时动作开始。

c) 在输入了表示不可能的日期时间(例如:2006年2月30日)的数值时，不能更改时间。
请重新输入正确的日期时间后操作。

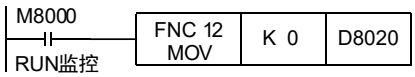
36.2.8 输入滤波器的调节[D8020]

X000～X017中分别配有0～60ms的数字滤波器回路，由特殊数据寄存器D8020的内容0～60决定使用哪个数字滤波器常数。
电源OFF→ON后，D8020的内容会自动变为10(10ms)。



1. 调节输入滤波的程序举例

执行了如下所示的程序后，滤波常数被更改为0ms。
但是，由于硬件部分设计有C-R滤波器，所以指定为0时，为下表的值。



1) A8/A8C可编程控制器

输入编号	输入滤波常数设定为0时的值
X000～X005	5 μs*1
X006,X007	50 μs
X010～X017*2	200 μs

*1. 使用5 μs的输入滤波常数时，或者用高速计数器读取50k～100kHz的脉冲时，请按如下所示进行设置。
－接线长度请勿超出5m。
－在输入端子中连接(1.5k Ω 1/2W以上)的漏电阻，与主机的输入电流相适应，对方一侧机器的开集电极型晶体管输出的负载电流需要在20mA以上。

*2. A8-16M□，A8C-16M□变为X000～X007。

- 在程序中可以多次更改这个输入滤波常数。
- 如果使用高速计数器、输入中断、脉冲捕捉(M8170～M8175)，不需要调节这个滤波常数。

36.2.9 电池[BATT(BAT)] LED,[ALM] LED灭灯指令[M8030]

如果接通M8030，则即使当内存备份用的电池电压过低时电池的LED灯也不会点亮。
当不想显示可编程控制器的电池电压过低错误时，或者想取下电池时，可以将M8030接通。
但是，在A8/A8C可编程控制器中，若以无电池运行为目的时，请勿使用M8030，详情参考后述内容。

1. 关于依据外围设备的参数设定进行无电池运行

请在参数设定中指定[batteryless mode]模式。

1) 指定了无电池运行时
通过指定[无电池运行]，可以在可编程控制器的系统一侧自动执行电池LED的灭灯控制以及如下所示的软元件保持区域的初始化。
－辅助继电器M　　－计数器C　　－状 态S
－数据寄存器D　　－定时器T　　－扩展寄存器R

2) 关于编程工具的对应
根据编程工具的版本，有些版本还未支持无电池运行，因此，在这种情况下，请依据下面的说明输入用于实现无电池运行的顺控程序。

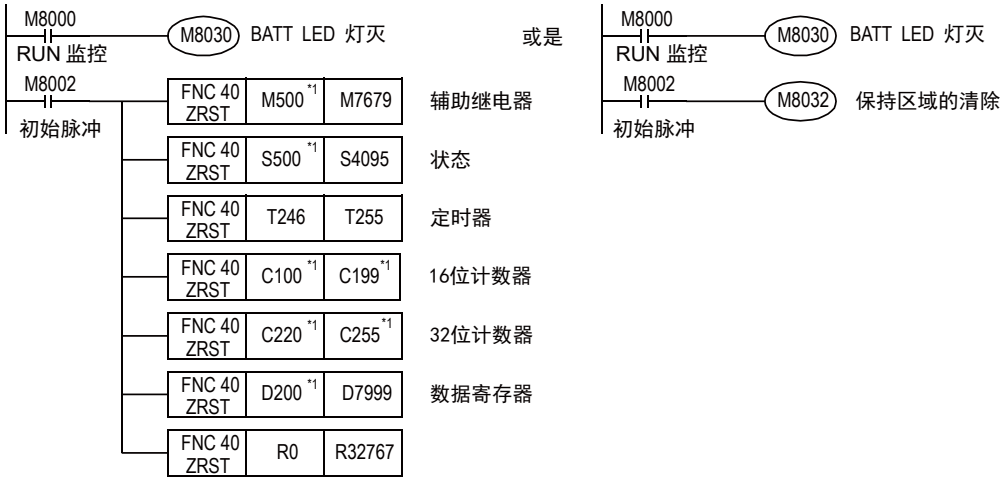
2. 可以无电池运行的条件

1) 安装有FLROM(选件的存储器盒)，确保程序不会丢失。(仅限A8/A8C可编程控制器)
2) 在控制中没有使用辅助继电器、数据寄存器等的软元件保持(停电保持)功能。
3) 没有使用采样跟踪功能。
4) 没有使用时钟(实时时钟)功能。(仅限A8/A8C可编程控制器)

3. 无电池运行用的程序举例

当不能设定参数的情况下，请编写如下所示的顺控程序。

- 清除内存备份区域的程序举例(参数的锁存范围设定为初始值时)



*1. 参数的锁存范围的设定(软元件设定)为默认值时的软元件编号。
发生变更时，请根据变更的内容进行更改。

4. 使用通信设定用软元件(D8120、D8121、D8129)时的注意事项

特殊数据寄存器D8120(通信格式)以及D8121(站号设定)、D8129(判断超时的时间)都是A8A/8C可编程控制器的电池备份的软元件。
在A8A/8C可编程控制器上使用无电池运行时，请与上述相同地进行复位，然后通过程序传送设定值。
前面提及的通信条件也有在参数设定中设定的方法。
在参数中事先设定了通信条件后，可编程控制器会在运行前将参数值写入前面提及的特殊数据寄存器中，所以建议在参数中设定。

36.2.10 清除指令[M8031，M8032]

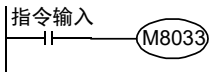
可以对可编程控制器的所有软元件(映象存储区)，执行不保持或是保持区域的清除。
M8031(不保持存储区的全部清除)，M8032(保持存储区的全部清除)

软元件编号	清除软元件
M8031 (不保持区域)	• 输出继电器(Y)，一般用辅助继电器(M)，一般用状态(S)的触点映象 • 定时器(T)的触点，计时线圈 • 一般用计数器的触点，计数线圈，复位线圈 • 一般用数据寄存器(D)的当前值 • 定时器(T)的当前值寄存器 • 一般用计数器(C)的当前值寄存器 • 一般用扩展寄存器
M 8032 (保持区域)	• 保持用辅助继电器(M)，保持用状态(S)的触点映象 • 累计定时器(T)的触点，计时线圈 • 保持用计数器和高速计数器的触点，计数线圈，复位线圈 • 保持用数据寄存器(D)的当前值寄存器 • 累计用和1ms用定时器(T)的当前值寄存器 • 保持用计数器和高速计数器的当前值寄存器 • 保持用扩展寄存器(R)

36.2.11 存储器保持停止[M8033](STOP中的输出保持)

如驱动特殊辅助继电器M8033，则可编程控制器从RUN变为STOP后，RUN时的输出状态还能原样保持。

1. 程序举例



例如通过可编程控制器驱动加热器时，加热器和其他的设备依旧被驱动，但可编程控制器停止，更改程序后可以重新运行。

36.2.12 禁止所有输出的指令[M8034]

驱动M8034后，通过清除输出内存使所有输出继电器的触点置OFF，可编程控制器在映像内存区运行。

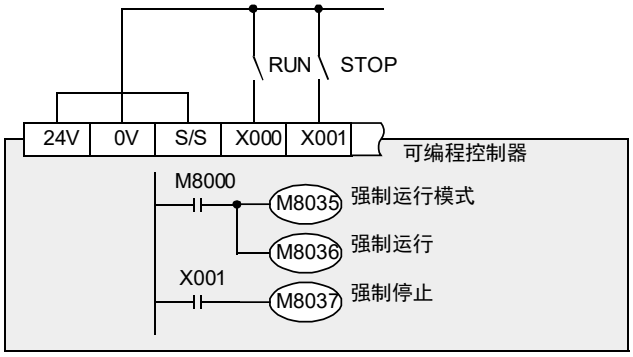
1. 程序举例



36.2.13 RUN/STOP输入的独立操作[M8035～M8037]

想通过外部的RUN，STOP按键开关控制可编程控制器的运行/停止时，参照以下步骤进行设定。
通过RUN开关的单触发输入使可编程控制器运行，通过STOP开关的单触发输入使其停止。

1. 程序举例



左图中，例举了A8可编程控制器(漏型输入)的示例。

在可编程控制器中编写上述程序。

2. 设定方法

1) 内置RUN/STOP开关固定在STOP一侧。
2) 指定RUN输入开关的输入(X)(在上述的示例中，指定了X000。)
为了使外部RUN/STOP输入有效，将X000～X017[*1]中的1点指定为RUN输入信号使用。

*1. A8-16M□，A8C-16M□，可编程控制(14点型)变为X000～X007。

a) 打开编程工具的参数设定
HCP-works中，双击工程目录的[Parameter]-[PCparameter]后显示对话框。
点击“PLCsystem(1)”选项卡后设定“RUNterminal input”。

b) 指定从STOP→RUN的切换开关的输入(X)的编号。

3) 指定STOP开关的输入(X)
用顺控程序指定任意的输入端子(可编程控制器的实际I/O)。
请参考上述的程序。
4) 将程序和参数传送至可编程控制器中。
5) 为了使参数设定有效，请将可编程控制器电源OFF一次后再次置ON。

3. 注意事项

1) 同时按下两个开关时
STOP一侧的开关优先。

2) 内置RUN/STOP开关置于RUN一侧时
在可编程控制器STOP时，内置的RUN/STOP开关设置到RUN一侧后，可以使可编程控制器运行。
但是，STOP指令需要通过分配了任意输入的STOP开关的输入来执行。(内置开关即使设置在RUN一侧，也是STOP指令优先)

4. 通过编程工具执行RUN/STOP指令

1) 使用计算机用的编程软件
具备可编程控制器的远程RUN/STOP功能。
使用这些软件，可以通过计算机发出的指令来控制可编程控制器的RUN/STOP。

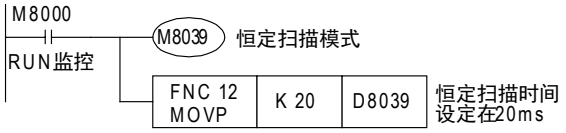
2) 使用其他编程工具时
通过强制ON/OFF操作来接通M8035(强制运行模式)和M8036(强制运行)后，可编程控制器运行。
在此，M8037(强制停止)接通后，可编程控制器停止。

3) 可编程控制器上的RUN/STOP开关设置在RUN侧时
通过编程工具执行远程STOP时，M8037(强制停止)有效。

36.2.14 恒定扫描模式[M8039，D8039](运算处理时间的固定化)

使特殊辅助继电器M8039置ON，在特殊数据寄存器D8039中写入目标的扫描时间(1ms单位)后，可编程控制器的运算周期不会低于这个值。
也就是说即使运算提早结束也将消耗剩余时间后才返回到0步。

1. 程序举例



2. 注意事项

1) 使用了与扫描同步执行的指令时
a) 使用了RAMP(FNC 67)、HKY(FNC 71)、SEGL(FNC 74)、ARWS(FNC 75)、PR(FNC 77)等
与扫描同步执行的指令时，建议使用这个恒定扫描模式，或是通过定时器中断每隔固定时间接通。
b) 当使用HKY(FNC 71)指令时
由于键盘输入的滤波器可能会导致响应延迟，所以需要使用20ms以上的扫描时间。

2) 显示扫描时间(D8010～D8012)
恒定扫描模式中指定的时间，包含在D8010～D8012的扫描时间的显示中。

36.2.15 使用STL指令、SFC图实现程序的状态控制[M8040]

驱动M8040后，即使转移条件满足，也不执行状态转移。
此外，已经停止的状态内的输出会继续保持原来的动作。

→ 关于状态内的输出复位，参考34.1.7项

36.2.16 模拟量特殊适配器[M8260～M8299，D8260～D8299]

连接了模拟量特殊适配器时，与连接台数相对应对下表的软元件分配动作及功能。
禁止写入的软元件在表中用阴影部分表示动作・功能。

→ 详细内容请参考A8/A8C系列用户手册[模拟量控制篇]

1. 特殊辅助继电器(M8260～M8299)

- A8/A8C可编程控制器

编号	动作・功能			
	A8-4AD-ADP	A8-4DA-ADP	A8-4AD-PT(W)-ADP	A8-4AD-TC-ADP
第1台				
M8260	通道1输入模式的切换	通道1输出模式的切换	选择温度单位	选择温度单位
M8261	通道2输入模式的切换	通道2输出模式的切换	未使用(不能使用)	K型、J型模式的切换
M8262	通道3输入模式的切换	通道3输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8263	通道4输入模式的切换	通道4输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8264	未使用(不能使用)	通道1输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8265	未使用(不能使用)	通道2输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8266	未使用(不能使用)	通道3输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8267	未使用(不能使用)	通道4输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8268	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
M8269	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
第2台				
M8270	通道1输入模式的切换	通道1输出模式的切换	选择温度单位	选择温度单位
M8271	通道2输入模式的切换	通道2输出模式的切换	未使用(不能使用)	K型、J型模式的切换
M8272	通道3输入模式的切换	通道3输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8273	通道4输入模式的切换	通道4输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8274	未使用(不能使用)	通道1输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8275	未使用(不能使用)	通道2输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8276	未使用(不能使用)	通道3输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8277	未使用(不能使用)	通道4输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8278	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
M8279	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
第3台				
M8280	通道1输入模式的切换	通道1输出模式的切换	选择温度单位	选择温度单位
M8281	通道2输入模式的切换	通道2输出模式的切换	未使用(不能使用)	K型、J型模式的切换
M8282	通道3输入模式的切换	通道3输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8283	通道4输入模式的切换	通道4输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8284	未使用(不能使用)	通道1输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8285	未使用(不能使用)	通道2输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8286	未使用(不能使用)	通道3输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8287	未使用(不能使用)	通道4输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8288	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
M8289	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
第4台				
M8290	通道1输入模式的切换	通道1输出模式的切换	选择温度单位	选择温度单位
M8291	通道2输入模式的切换	通道2输出模式的切换	未使用(不能使用)	K型、J型模式的切换
M8292	通道3输入模式的切换	通道3输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8293	通道4输入模式的切换	通道4输出模式的切换	未使用(不能使用)	未使用(不能使用)
M8294	未使用(不能使用)	通道1输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8295	未使用(不能使用)	通道2输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8296	未使用(不能使用)	通道3输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8297	未使用(不能使用)	通道4输出保持的解除	未使用(不能使用)	未使用(不能使用)
M8298	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)
M8299	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)	未使用(不能使用)

*1. A8可编程控制器从首批产品开始就已对应

编号	动作・功能
	A8-4AD-PNK-ADP
第1台	
M8260	选择温度单位
M8261	选择温度传感器
M8262	未使用(不能使用)
M8263	未使用(不能使用)
M8264	未使用(不能使用)
M8265	未使用(不能使用)
M8266	未使用(不能使用)
M8267	未使用(不能使用)
M8268	未使用(不能使用)
M8269	未使用(不能使用)
第2台	
M8270	选择温度单位
M8271	选择温度传感器
M8272	未使用(不能使用)
M8273	未使用(不能使用)
M8274	未使用(不能使用)
M8275	未使用(不能使用)
M8276	未使用(不能使用)
M8277	未使用(不能使用)
M8278	未使用(不能使用)
M8279	未使用(不能使用)
第3台	
M8280	选择温度单位
M8281	选择温度传感器
M8282	未使用(不能使用)
M8283	未使用(不能使用)
M8284	未使用(不能使用)
M8285	未使用(不能使用)
M8286	未使用(不能使用)
M8287	未使用(不能使用)
M8288	未使用(不能使用)
M8289	未使用(不能使用)
第4台	
M8290	选择温度单位
M8291	选择温度传感器
M8292	未使用(不能使用)
M8293	未使用(不能使用)
M8294	未使用(不能使用)
M8295	未使用(不能使用)
M8296	未使用(不能使用)
M8297	未使用(不能使用)
M8298	未使用(不能使用)
M8299	未使用(不能使用)

编号	动作・功能	
	A8-4AD-TC-ADP	A8-4AD-PNK-ADP
第1台		
M8280	选择温度单位	选择温度单位
M8281	K型、J型模式的切换	选择温度传感器
M8282	未使用(不能使用)	未使用(不能使用)
M8283	未使用(不能使用)	未使用(不能使用)
M8284	未使用(不能使用)	未使用(不能使用)
M8285	未使用(不能使用)	未使用(不能使用)
M8286	未使用(不能使用)	未使用(不能使用)
M8287	未使用(不能使用)	未使用(不能使用)
M8288	未使用(不能使用)	未使用(不能使用)
M8289	未使用(不能使用)	未使用(不能使用)
第2台		
M8290	选择温度单位	选择温度单位
M8291	K型、J型模式的切换	选择温度传感器
M8292	未使用(不能使用)	未使用(不能使用)
M8293	未使用(不能使用)	未使用(不能使用)
M8294	未使用(不能使用)	未使用(不能使用)
M8295	未使用(不能使用)	未使用(不能使用)
M8296	未使用(不能使用)	未使用(不能使用)
M8297	未使用(不能使用)	未使用(不能使用)
M8298	未使用(不能使用)	未使用(不能使用)
M8299	未使用(不能使用)	未使用(不能使用)

2. 特殊数据寄存器(D8260～D8299)

• A8/A8C可编程控制器

编号	动作・功能			
	A8-4AD-ADP	A8-4DA-ADP	A8-4AD-PT(W)-ADP	A8-4AD-TC-ADP
第1台				
D8260	通道1输入数据	通道1输出设定数据	通道1测量温度	通道1测量温度
D8261	通道2输入数据	通道2输出设定数据	通道2测量温度	通道2测量温度
D8262	通道3输入数据	通道3输出设定数据	通道3测量温度	通道3测量温度
D8263	通道4输入数据	通道4输出设定数据	通道4测量温度	通道4测量温度
D8264	通道1平均次数(1～4095)	未使用(不能使用)	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8265	通道2平均次数(1～4095)	未使用(不能使用)	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8266	通道3平均次数(1～4095)	未使用(不能使用)	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8267	通道4平均次数(1～4095)	未使用(不能使用)	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8268	错误状态	错误状态	错误状态	错误状态
D8269	机型代码=1	机型代码=2	机型代码=20，21*1	机型代码=10
第2台				
D8270	通道1输入数据	通道1输出设定数据	通道1测量温度	通道1测量温度
D8271	通道2输入数据	通道2输出设定数据	通道2测量温度	通道2测量温度
D8272	通道3输入数据	通道3输出设定数据	通道3测量温度	通道3测量温度
D8273	通道4输入数据	通道4输出设定数据	通道4测量温度	通道4测量温度
D8274	通道1平均次数(1～4095)	未使用(不能使用)	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8275	通道2平均次数(1～4095)	未使用(不能使用)	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8276	通道3平均次数(1～4095)	未使用(不能使用)	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8277	通道4平均次数(1～4095)	未使用(不能使用)	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8278	错误状态	错误状态	错误状态	错误状态
D8279	机型代码=1	机型代码=2	机型代码=20，21*1	机型代码=10
第3台				
D8280	通道1输入数据	通道1输出设定数据	通道1测量温度	通道1测量温度
D8281	通道2输入数据	通道2输出设定数据	通道2测量温度	通道2测量温度
D8282	通道3输入数据	通道3输出设定数据	通道3测量温度	通道3测量温度
D8283	通道4输入数据	通道4输出设定数据	通道4测量温度	通道4测量温度
D8284	通道1平均次数(1～4095)	未使用(不能使用)	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8285	通道2平均次数(1～4095)	未使用(不能使用)	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8286	通道3平均次数(1～4095)	未使用(不能使用)	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8287	通道4平均次数(1～4095)	未使用(不能使用)	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8288	错误状态	错误状态	错误状态	错误状态
D8289	机型代码=1	机型代码=2	机型代码=20，21*1	机型代码=10
第4台				
D8290	通道1输入数据	通道1输出设定数据	通道1测量温度	通道1测量温度
D8291	通道2输入数据	通道2输出设定数据	通道2测量温度	通道2测量温度
D8292	通道3输入数据	通道3输出设定数据	通道3测量温度	通道3测量温度
D8293	通道4输入数据	通道4输出设定数据	通道4测量温度	通道4测量温度
D8294	通道1平均次数(1～4095)	未使用(不能使用)	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8295	通道2平均次数(1～4095)	未使用(不能使用)	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8296	通道3平均次数(1～4095)	未使用(不能使用)	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8297	通道4平均次数(1～4095)	未使用(不能使用)	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8298	错误状态	错误状态	错误状态	错误状态
D8299	机型代码=1	机型代码=2	机型代码=20，21*1	机型代码=10

*1. PT-ADP变为机型代码=20。PTW-ADP变为机型代码=21。

编号	动作・功能
	A8-4AD-PNK-ADP
第1台	
D8260	通道1测量温度
D8261	通道2测量温度
D8262	通道3测量温度
D8263	通道4测量温度
D8264	通道1平均次数(1～4095)
D8265	通道2平均次数(1～4095)
D8266	通道3平均次数(1～4095)
D8267	通道4平均次数(1～4095)
D8268	错误状态
D8269	机型代码:K11
第2台	
D8270	通道1测量温度
D8271	通道2测量温度
D8272	通道3测量温度
D8273	通道4测量温度
D8274	通道1平均次数(1～4095)
D8275	通道2平均次数(1～4095)
D8276	通道3平均次数(1～4095)
D8277	通道4平均次数(1～4095)
D8278	错误状态
D8279	机型代码:K11
第3台	
D8280	通道1测量温度
D8281	通道2测量温度
D8282	通道3测量温度
D8283	通道4测量温度
D8284	通道1平均次数(1～4095)
D8285	通道2平均次数(1～4095)
D8286	通道3平均次数(1～4095)
D8287	通道4平均次数(1～4095)
D8288	错误状态
D8289	机型代码:K11
第4台	
D8290	通道1测量温度
D8291	通道2测量温度
D8292	通道3测量温度
D8293	通道4测量温度
D8294	通道1平均次数(1～4095)
D8295	通道2平均次数(1～4095)
D8296	通道3平均次数(1～4095)
D8297	通道4平均次数(1～4095)
D8298	错误状态
D8299	机型代码:K11

编号	动作・功能	
	A8-4AD-TC-ADP	A8-4AD-PNK-ADP
第1台		
D8280	通道1测量温度	通道1测量温度
D8281	通道2测量温度	通道2测量温度
D8282	通道3测量温度	通道3测量温度
D8283	通道4测量温度	通道4测量温度
D8284	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8285	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8286	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8287	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8288	错误状态	错误状态
D8289	机型代码=10	机型代码: K11
第2台		
D8290	通道1测量温度	通道1测量温度
D8291	通道2测量温度	通道2测量温度
D8292	通道3测量温度	通道3测量温度
D8293	通道4测量温度	通道4测量温度
D8294	通道1平均次数(1～4095)	通道1平均次数(1～4095)
D8295	通道2平均次数(1～4095)	通道2平均次数(1～4095)
D8296	通道3平均次数(1～4095)	通道3平均次数(1～4095)
D8297	通道4平均次数(1～4095)	通道4平均次数(1～4095)
D8298	错误状态	错误状态
D8299	机型代码=10	机型代码: K11

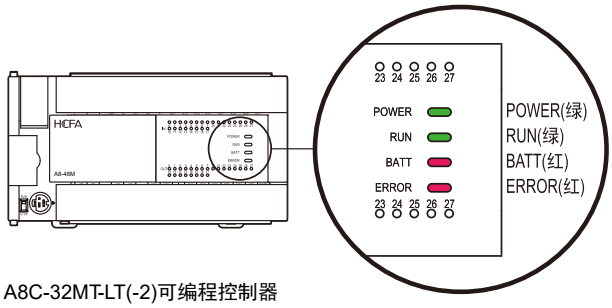
37. 故障的查看方法和错误代码一览

在程序运行中产生错误时，请参考本章的内容解决。
关于其他错误的详细内容，请分别参考通信控制手册以及所使用的可编程控制器主机的硬件手册。

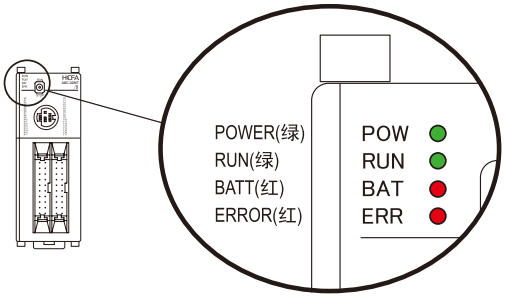
37.1 显示可编程控制器运行情况的LED及显示颜色

发生异常时，根据可编程控制器中的各种LED的亮灯情况可以确认大体的状况。

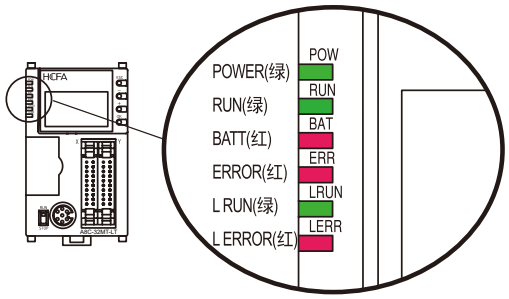
A8可编程控制器



A8C(D,DSS)可编程控制器



A8C-32MT-LT(-2)可编程控制器



37.1.1 POWER(POW) LED [灯亮/闪烁/灯灭][A8/A8C]

LED的状态	可编程控制器的状态	解决办法
灯亮	给电源端子供应了规定的电压。	电源正常。
闪烁	以下的任意一种状态。 • 没有给电源端子提供规定的电压、电流。 • 外部接线不正确。 • 可编程控制器内部有故障。	• 请确认电源电压。 • 在拆除电源电缆以外的连接电缆后，再次上电，请确认状态是否有变化。如果状态没有改善，请与三菱电机自动化(上海)有限公司或其办事处联系。
灯灭	以下的任意一种状态。 • 电源断开。 • 没有给电源端子提供规定的电压。 • 电源电缆断线。	在电源没有断开的情况下，请确认电源以及电源路径。 如果供电正确，请与三菱电机自动化(上海)有限公司或其办事处联系。

37.1.2 RUN LED [灯亮/灯灭][A8/A8C]

LED的状态	可编程控制器的状态	解决办法
灯亮	顺控程序处理执行中	显示可编程控制器的运行状态。 该指示灯根据ERROR(ERR) LED的状态(参考37.1.5项)可能会不亮。
灯灭	顺控程序处理停止中	

37.1.3 BATT(BAT) LED [灯亮/灯灭][A8/A8C]

LED的状态	可编程控制器的状态	解决办法
灯亮	电池的电压过低。	请迅速更换电池。(请参考主机手册)
灯灭	电池的电压超出了D8006中设定的数值时。	正常。

37.1.4 ERROR(ERR) LED [灯亮/闪烁/灯灭][A8/A8C]

LED的状态	可编程控制器的状态	解决办法
灯亮	可能是看门狗定时器错误，也可能是可编程控制器的硬件受损。	1) 停止可编程控制器，再次上电。 ERROR(ERR) LED灯灭时，考虑是由于看门狗定时器错误造成的。请执行下述中的某一个解决方法。 －修改程序 扫描时间的最大值(D8012)请勿超出看门狗定时器的设定值(D8000)。 －输入中断或脉冲捕捉中使用的输入， 在1个运算周期内是否频繁地ON/OFF多次。 －高速计数器中输入的脉冲(DUTY 50%)的频率是否超出规格的范围。 －增加WDT指令 请在程序中加入多个WDT指令， 在1个运算周期内复位几次看门狗定时器。 －变更看门狗定时器的设定值 通过程序， 更改看门狗定时器的设定值(D8000)， 使其值大于扫描时间的最大值(D8012)。 2) 拆下可编程控制器，提供其他的电源。 ERROR(ERR) LED灯灭时，考虑有可能是噪音带来的影响，因此请考虑以下的解决措施。 －确认接地情况， 修改接线的路径以及设置的场所。 －在电源线上加上噪音滤波器。 3) 实施了1)～2)后ERROR(ERR) LED灯仍然不灭的时候，请与三菱电机自动化(上海)有限公司或其办事处联系。
闪烁	在可编程控制器中出现了以下的任意一种错误。 • 参数错误 • 语法错误 • 回路错误	请用编程工具进行PLC诊断以及程序的检查。 关于对应解决方法，请参考37.4节
灯灭	没有出现会使可编程控制器停止的错误。	可编程控制器的动作中出现异常时，请用编程工具执行PLC诊断以及程序检查。 有可能发生[I/O构成错误]、[串行通信错误]、[运算错误]。

37.1.5 L RUN LED [仅A8C-32MT-LT(-2)]

模式	LED的状态	可编程控制器的状态	解决办法
ONLINE	灯亮	数据链接执行中	—
	灯灭	数据链接停止中	• 根据LERR. LED的状态执行各自相应的对策。
CONFIG*1	灯亮	数据链接执行中	—
	灯灭	数据链接停止中	• 根据LERR. LED的状态执行各自相应的对策。
TEST	灯亮	自折返测试正常结束	—
	灯灭	自折返测试异常结束 (在执行自折返测试时灯灭)	• 请确认给可编程控制器供电的电源。 • 请与浙江禾川科技股份有限公司或其办事处联系。

*1. 仅A8C-32MT-LT

37.1.6 L ERR. LED [仅A8C-32MT-LT(-2)]

模式	LED的状态	可编程控制器的状态	解决办法
ONLINE	灯亮	模块宕机错误 控制范围外的站点错误 RD站站号设置错误	• 请切实连接内置主站与远程I/O模块、远程设备站。 • 请使实际连接的远程 I/O 模块、远程设备站与远程站的详细信息一致。
	闪烁	所有站点异常	• 请切实连接内置主站和远程I/O模块、远程设备站。 • 请使实际连接的远程 I/O 模块、远程设备站与远程站的详细信息一致。
	灯灭	数据链接正常执行中	—
CONFIG*1	灯亮	使用的站号不一致(编辑远程站信息时进行检查)	• 请切实连接内置主站和远程I/O模块、远程设备站。 • 请使实际连接的远程 I/O 模块、远程设备站与远程站的详细信息一致。 • 请检查远程设备站的站号是否在范围内。
	闪烁	所有站异常(编辑远程站信息时进行检查)	
	灯灭	数据链接正常执行中	—
TEST	灯亮	自折返测试异常结束	• 请确认给可编程控制器供电的电源。 • 请与浙江禾川科技股份有限公司或其办事处联系。
	灯灭	自折返测试正常结束 (在执行自折返测试时灯灭)	—

*1. 仅A8C-32MT-LT

→ 详细内容，请参考所使用的可编程控制器主机的硬件篇

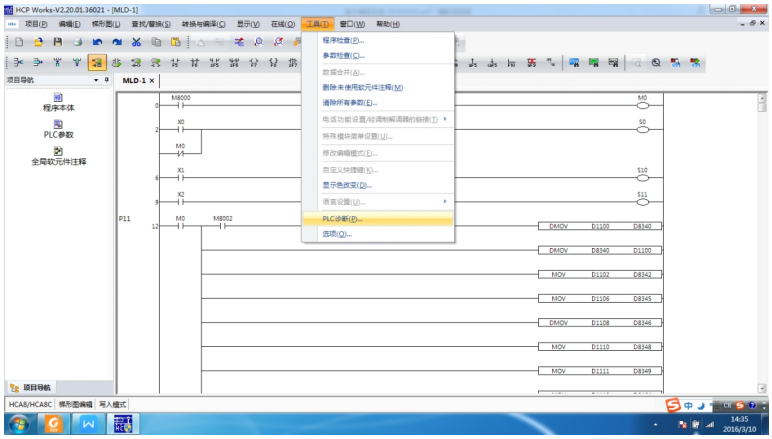
37.2 错误代码的查看方法及显示内容

37.2.1 使用HCP-works的操作及查看方法

1 连接计算机和可编程控制器。

2 执行PLC诊断。

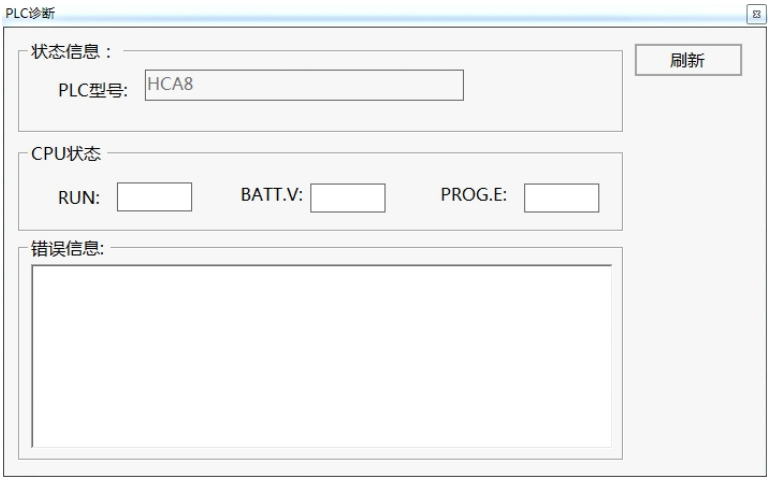
点击菜单中的[Diagnostics]-[PLC diagnostics]后，执行可编程控制器的诊断。



3 确认诊断结果。

显示下面的窗口，可以确认错误内容。

＜错误显示画面示例＞



37.3 检测错误用的软元件的补充说明

37.3.1 错误的检测(M8060～/D8060～)

M8060、M8061、M8064～M8067中任意一个接通后，其中小的编号保存到D8004中，M8004动作。

1) 可编程控制器STOP→RUN时，M8060、M8061、M8064～M8067被清除。
因M8068、D8068未被清除，请注意。

2) 驱动(ON)M8069后，输入输出扩展单元、扩展用电源单元、扩展模块出现故障时可以使可编程控制器停止(M8061 PLC硬件错误)。
驱动(ON)M8069时，执行I/O总线检查，发生错误时在D8061中写入错误代码6103，或6104，同时接通M8061。
但是，写入了6104时，M8009也接通，并将DC24V掉电的扩展用电源单元之后连接的输入输出信号写入D8009中。

3) 被编程的I/O编号的单元以及模块未被安装时，M8060动作，同时将其起始软元件编号写入D8060中。

(例如)X020未安装时

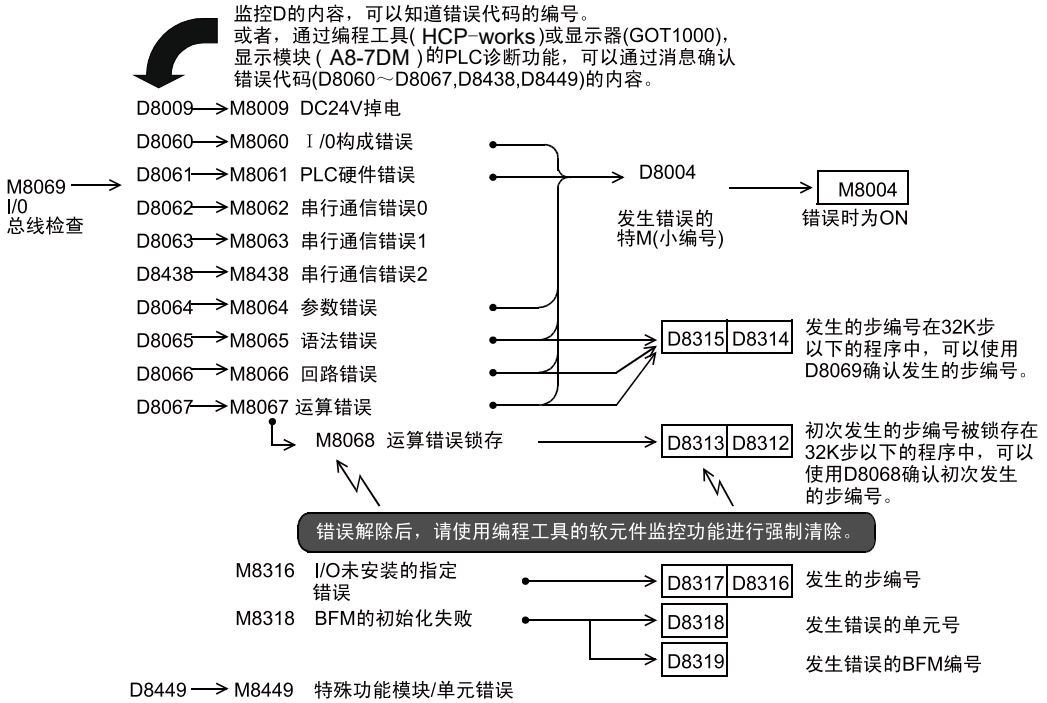


*1. A8/A8C可编程控制器可达10～337。

4) 在LD、AND、OR、OUT等指令中直接指定软元件编号，或是通过变址间接指定时，输入，输出的软元件编号实际没有被安装时，M8316接通，在D8317(高位)、D8316(低位)中写入这个指令的错误步编号。

37.3.2 检测错误用的特殊软元件的动作关系

检测错误用的特殊辅助继电器和特殊数据寄存器之间按照下面的关系动作。
从编程工具监控辅助继电器以及数据寄存器的内容，并使用PLC诊断功能查看发生的情况。



37.3.3 错误的检测时序

错误项目	ERROR LED 的状态	可编程控制器的状态	错误检测的时序		
			电源OFF→ON	STOP→RUN时	其他
M8060 I/O构成错误	灯灭	RUN	检查	检查	—
M8061 PLC硬件错误	灯亮	STOP	检查	—	一直
M8062串行通信错误0[通道0]	灯灭	RUN	—	—	从对方站点接受信号时
M8063 串行通信错误1[通道1]	灯灭	RUN	—	—	从对方站点接受信号时
M8438 串行通信错误2[通道2]	灯灭	RUN	—	—	从对方站点接受信号时
M8064 参数错误	闪烁	STOP	检查	检查	更改程序时(STOP) 传送程序时(STOP)
M8065 语法错误	闪烁	STOP			
M8066 回路错误	闪烁	STOP			
M8067 运算错误	灯灭	RUN	—	—	RUN中
M8068 运算错误的锁存	灯灭	RUN			
M8109 输出刷新错误	灯灭	RUN	—	—	一直
M8316 I/O未安装的指定错误	灯灭	RUN	—	—	RUN中
M8318 BFM的初始化失败	灯灭	RUN	—	检查	—
M8449 特殊模块错误	灯灭	RUN	—	—	一直

37.4 错误代码一览及解决方法

发生可编程控制器的程序错误时，特殊数据寄存器D8060～D8067、D8438、D8449中保存的错误代码及其解决方法如下所示。

错误代码	错误时动作	错误内容	解决方法
I/O构成错误[M8060(D8060)]			
例如1020	继续运行	未安装的I/O的起始软元件编号 (例如)X020未安装时  • 1～3位数:软元件编号 A8/A8C: 10～337 • 4位数:输入输出的种类(1=输入X，0=输出Y) 例: D8060中保存了1020时，输入的X020以后没有安装	未安装的输入继电器、输出继电器的编号被用于编写程序。 可编程控制器会继续运行，但是如有程序错误时，请修改。
串行通信错误2[M8438(D8438)]			
0000	—	无异常	
3801	继续运行	奇偶校验错误，溢出错误，帧错误	• 变频器通信、计算机链接、编程 请确认是否根据用途正确设定了通信参数。 • 简易PC间链接、并联链接等 请确认是否根据用途正确设定程序。 • 远程维护 请确认调制解调器的电源接通以及AT命令的设定内容。 • 还请确认通信电缆的接线情况。
3802		通信字符错误	
3803		通信数据的和校验不一致	
3804		数据格式错误	
3805		命令错误	
3806		监视超时	
3807		调制解调器初始化错误	
3808		简易PC间链接的参数错误	
3812		并联链接字符错误	
3813		并联链接求和校验错误	
3814		并联链接格式错误	
3820		变频器通信功能中的通信错误	
PLC硬件错误[M8061(D8061)]			
0000	—	无异常	
6101	停止运行	RAM错误	
6102		运算回路错误	
6103		I/O总线错误(M8069 ON时)	请检查扩展电缆的连接是否正确。
6104		扩展单元24V掉电(M8069 ON时)	
6105		看门狗定时器错误	采样(运算时间)超出了D8000的值。 请确认程序。
6106		I/O表制作错误错误(CPU错误)	上电时扩展单元的24V掉电。 (最长等待10秒以上仍然掉电。) 上电时，HC-Link/CT(A8C-32MT-LT(-2)内置)的输入输出分配没有做好。
6112		安装有存储器盒的情况下，执行内置HC-Link/CT的设定更改时，不能正常写入到存储器盒中。	确认存储器盒的连接
6113		安装有存储器盒的情况下，执行内置HC-Link/CT的设定更改时，由于保护开关置ON，因此不能更改设定。	将存储器盒的保护开关置为OFF。

错误代码	错误时动作	错误内容	解决方法
6114	停止运行	不能对内置HC-Link/LT主站执行设定的写入	
6115		向内置HC-Link/LT 主站的EEPROM 中写入超时错误。或者，采用自配置模式时，不能正常结束配置	
PLC/PP通信错误(D8062) 串行通信错误0[M8062(D8062)]			
0000	—	无异常	
6201	继续运行	奇偶校验错误，溢出错误，帧错误	请检查编程面板(PP)或编程用的连接器上连接的设备与可编程控制器(PLC)之间的连接是否确实连好。 如在监控可编程控制器的过程中插拔连接器，也可能发生错误。
6202		通信字符错误	
6203		通信数据的和校验不一致	
6204		数据格式错误	
6205		命令错误	
串行通信错误1 [M8063(D8063)]			
0000	—	无异常	
6301	继续运行	奇偶校验错误，溢出错误，帧错误	• 变频器通信、计算机链接、编程 请确认是否根据用途正确设定了通信参数。 • 简易PC间链接、并联链接等 请确认是否根据用途正确设定程序。 • 远程维护 请确认调制解调器的电源接通以及AT命令的设定内容。 • 还请确认通信电缆的接线情况。
6302		通信字符错误	
6303		通信数据的和校验不一致	
6304		数据格式错误	
6305		命令错误	
6306		监视超时	
6307		调制解调器初始化错误	
6308		简易PC间链接的参数错误	
6312		并联链接字符错误	
6313		并联链接求和校验错误	
6314		并联链接格式错误	
6320		变频器通信功能中的通信错误	
参数错误[M8064(D8064)]			
0000	—	无异常	
6401	停止运行	程序和校验不一致	请停止可编程控制器，正确设定参数。
6402		内存容量的设定错误	
6403		保持区域的设定错误	
6404		注释区域的设定错误	
6405		文件寄存器的区域设定错误	
6406		BFM初始值数据的和校验不一致	
6407		BFM初始值数据异常	
6409		其他的设定错误	
6411		内置HC-Link/LT设定(专用区域)的参数错误	
6412		内蔵HC-Link/LT設定(特殊设定区域)的参数和校验不一致	
6413		内蔵HC-Link/LT設定(专用区域)的参数和校验不一致	

错误代码	错误时动作	错误内容	解决方法
语法错误[M8065(D8065)]			
0000	—	无异常	
6501	停止运行	指令－软元件符号－软元件编号的组合错误	编写程序时请检查各指令的使用方法是否正确，出现错误时，请在编程模式下修改指令。
6502		在设定值前面没有OUT T、OUT C	
6503		OUT T、OUT C后面没有设定值 应用指令的操作数不够	
6504		标签编号重复 中断输入和高速计数器输入重复	
6505		软元件编号超出范围	
6506		使用了未定义的指令	
6507		标签编号(P)的定义错误	
6508		中断输入(I)的定义错误	
6509		其他	
6510		MC的嵌套编号的大小关系错误	
回路错误[M8066(D8066)]			
0000	—	无异常	
6610	停止运行	LD、LDI的连续使用次数超过9次	作为回路块整体的指令组合方法有不正确的地方时，或者成对的指令的关系不正确时，会发生这样的错误。 请在编程模式下，将指令的相互关系修改正确。
6611		相对LD、LDI指令而言，ANB、ORB指令数过多	
6612		相对LD、LDI指令而言，ANB、ORB指令数过少	
6613		MPS的连续使用次数超过12次	
6614		遗漏MPS	
6615		遗漏MPP	
6616		MPS-MRD、MPP之间的线圈遗漏，或关系错误	
6617		应该从母线开始的指令没有连接在母线上 STL、RET、MCR、P、I、 DI、EI、FOR、NEXT、SRET、IRET、 FEND、END	
6618		只能在主程序中使用的指令在主程序以外(中断、子程序等)。 STL、MC、MCR	
6619		FOR-NEXT之间有不能使用的指令 STL、RET、MC、MCR、I、IRET	
6620		FOR-NEXT嵌套超出	
6621		FOR-NEXT数的关系错误	
6622		无NEXT指令	
6623		无MC指令	
6624		无MCR指令	
6625		STL的连续使用次数超出9次	
6626		STL-RET之间有不能使用的指令 MC、MCR、I、SRET、IRET	
6627		无STL指令	
6628		在主程序中有主程序不能使用的指令 I、SRET、IRET	
6629		无P、I	
6630		无SRET、IRET指令 在子程序内有STL-RET或者MC-MCR指令	
6631		不能使用SRET指令的场所中有SRET指令	
6632		不能使用FEND指令的场所中有FEND指令	

错误代码	错误时动作	错误内容	解决方法
运算错误[M8067(D8067)]			
0000		无异常	
6701	继续运行	• 没有CJ、CALL的跳转目标地址 • 变址修饰的结果、标签未定义，以及在 P0 ～ P4095以外时 • 在 CALL指令中执行了P63。因为P63 是向END跳转的标签，所以不能在CALL指令中使用	这些是在运算执行过程中产生的错误，请修改程序或是检查应用指令的操作数的内容。 即使没有发生语法、回路错误，但是例如由于下列原因也会产生运算错误。 (例如) T500Z本身没有错误，但是如果运算结果为Z=100，则会变为T600，那样软元件编号会超出。
6702		CALL的嵌套超出6个	
6703		中断的嵌套超出3个	
6704		FOR-NEXT的嵌套超出6个	
6705		应用指令的操作数是对象软元件以外的软元件	
6706		应用指令的操作数的软元件编号范围或数据的值超出	
6707		没有设定文件寄存器的参数，就对文件寄存器进行访问	
6708		FROM/TO指令错误	在执行运算过程中出现的错误。 • 请修改程序，或是检查应用指令的操作数的内容。 • 请确认相应的设备中是否存在指定的缓冲存储区。 • 请确认扩展电缆的连接情况。
6709		其他(不正确的分支等)	这些是在运算执行过程中产生的错误，请修改程序或是检查应用指令的操作数的内容。 即使没有发生语法、回路错误，但是例如由于下列原因也会产生运算错误。 (例如) T500Z本身没有错误，但是如果运算结果为Z=100，则会变为T600，那样软元件编号会超出。
6710		参数之间的不匹配	在移位指令等中，存在源操作数和目标操作数的重复的情况
6730		采样时间(Ts)为对象范围以外 (Ts≦0)	《停止PID运算》 在控制参数的设定值或PID运算中出现数据错误。 请检查参数的内容。
6732		输入滤波器常数(α)为对象范围以外 (α<0或100≦α)	
6733		比例增益(KP)为对象范围以外 (KP<0)	
6734		积分时间(TI)为对象范围以外 (TI<0)	
6735		微分增益(KD)为对象范围以外 (KD<0或201≦KD)	
6736		微分时间(TD)为对象范围以外(TD<0)	
6740		采样时间(Ts)≦运算周期	《继续自整定》 视为采样时间(Ts)=循环时间(运算周期) 运算、继续执行。

错误代码	错误时动作	错误内容	解决方法
运算错误[M8067(D8067)]			
6742	继续运行	测量值变化量超出 (ΔPV<-32768或是32767<ΔPV)	《继续PID运算》 各参数在最大值或最小值下继续运行。
6743		偏差超出 (EV<-32768或是32767<EV)	
6744		积分计算值超出 (-32768～32767以外)	
6745		由于微分增益(KD)超出导致微分值超出	
6746		微分计算值超出 (-32768～32767以外)	
6747		PID运算结果超出 (-32768～32767以外)	
6748		PID输出上限设定值<输出下限设定值	《替换输出上限值和输出下限值→继续PID运算》 请确认对象的设定内容是否正确。
6749		PID输入变化量报警设定值、输出变化量报警设定值异常(设定值<0)	《无报警输出→继续PID运算》 请确认对象的设定内容是否正确。
6750		《步响应法》 自整定结果错误	《自整定结束→转移到PID运算》 • 自整定开始时的偏差为150以下时结束。 • 自整定结束时的偏差为开始时偏差的1/3以上时结束。 确认测量值、目标值以后，请再次执行自整定。
6751		《步响应法》 自整定动作方向不一致	《自整定强制结束→不转移到PID运算》 从自整定开始时的测量值考虑的动作方向，与自整定用的输出中的实际动作方向不一致。 请将目标值、自整定用输出值，测量值的关系都修改正确后，再次执行自整定。
6752		《步响应法》 自整定动作错误	《自整定结束→不转移到PID运算》 在自整定中，由于设定值上下变动，导致自整定不能正确动作。 请将采样时间设置成远远大于输出的变化周期的时间，或是将输入滤波器的常数放大。 在更改设定后，重新执行自整定。
6753		《极限循环法》 自整定用输出设定值异常 [ULV(上限)≦LLV(下限)]	《自整定强制结束→不转移到PID运算》 请确认对象的设定内容是否正确。
6754		《极限循环法》 自整定用PV临界值(滞后)设定值异常(SHPV < 0)	
6755		《极限循环法》 自整定状态转移异常 (管理状态转移的软元件的数据被异常改写)	《自整定强制结束→不转移到PID运算》 请确认程序中是否改写了PID指令占用的软元件。
6756		《极限循环法》 由于自整定测量时间超出导致的结果异常(τ on > τ, τ on < 0, τ < 0)	《自整定强制结束→不转移到PID运算》 自整定所需的时间超出了原先的需要。请确认采取将自整定用输出值的上下限的差(ULV-LLV) 变大，输入滤波器常数α、自整定用PV临界值SHPV的值变小等措施后，是否看到改善的效果。
6757		《极限循环法》 自整定结果的比例增益超出(KP=0～32767以外)	《自整定结束(KP=32767)→转移到PID运算》 相对输出值而言测量值(PV)的值的变化小。请通过将测量值(PV)放大10倍后输入等方法，将自整定中的PV的变化放大。

错误代码	错误时动作	错误内容	解决方法
运算错误[M8067(D8067)]			
6758	继续运行	《极限循环法》 自整定结果的积分时间超出 (TI=0～32767以外)	《自整定结束(KP=32767)→转移到PID运算》 自整定所需的时间超出了原先的需要。请确认采取将自整定用输出值的上下限的差(ULV-LLV) 变大，输入滤波器常数α、自整定用PV临界值SHPV的值变小等措施后，是否看到改善的效果。
6759		《极限循环法》 自整定结果的微分时间超出 (TD=0～32767以外)	
6760		来自伺服的ABS数据的和校验不一致	请确认与伺服的连接，及设定情况。
6762		在变频器通信指令中指定的通信端口，已经在其他的通信中被使用了。	请确认指定的通信端口是否在其他指令中被使用了。
6763		1) DSZR、DVIT、ZRN 指令中指定的输入 (X)，已经在其他的指令中被使用了。 2) DVIT指令的中断信号软元件在设定范围以外。	1) 请确认DSZR、DVIT和ZRN指令中指定的输入(X)是否在下列的用途中被使用。 －输入中断(包含延迟功能) －高速计数器C235～C255 －脉冲捕捉M8170～M8177 －SPD指令 2) 请确认DVIT指令的中断信号指定D8336的内容。
6764		脉冲输出编号，已经在定位指令、脉冲输出指令(PLSY、PWM等)中被使用了。	请确认在脉冲输出对象地址中指定的输出，是否在其他的定位指令中被驱动。
6765		应用指令的使用次数错误	请确认是否超限使用在程序中有使用次数限制的应用指令。
6770		《A8/A8C可编程控制器》 存储器盒的写入错误	
6771		存储器盒未连接	请确认是否正确安装了存储器盒。
6772		存储器盒禁止写入时的写入错误	存储器盒的保护开关为ON时，向存储器盒执行了写入。 请将保护开关置为OFF。
6773		对RUN中写入中的存储器盒的访问错误	在RUN中写入时执行了与存储器盒之间的传送(读出/写入)。
特殊模块错误[M8449(D8449)]			
□020*1	继续运行	一般数据的和校验错误	请确认扩展电缆的连接情况
□021*1		一般数据的报文异常	
□080*1		FROM/TO错误	在执行运算过程中出现的错误。 • 请修改程序，或是检查应用指令的操作数的内容。 • 请确认相应的设备中是否存在指定的缓冲存储区。 • 请确认扩展电缆的连接情况。
□090*1		外围设备访问错误	• 检查编程面板(PP)或编程用的连接器上连接的设备与可编程控制器(PLC)之间的连接是否确实连好 • 请确认扩展电缆的连接情况。

*1. □中为发生错误的特殊功能模块/单元的0～7的单元编号。

浙江禾川电子股份

400-8181-810

CALL CENTER 全国服务热线

www.hcfa.cn

浙江禾川科技股份有限公司

地址：浙江省龙游工业园区阜财路9号

电话：0570-7117888（总机）

传真：0570-7882868

网址：www.hcfa.cn

邮箱：hechuan@hcfa.cn