QCPU(Q模式)/QnACPU

MITSUBISHI

编程手册

(公共指令篇)1/2



三菱可编程控制器

MELSEC-Q

●安全警告●

（在使用产品之前必须先阅读此警告）

在使用产品之前，请先阅读此手册和此手册中介绍的相关手册。请对安全问题给予足够重视以便能正确操作此产品。

请将此手册保存到一个安全的地方，以便在需要时可以方便地得到。请送给最终用户本手册的副本。

修订本

此手册编号在封底的左下角给出。

出版日期	手册编号	版本
2003 年 09 月	SH(NA)-080450CHN-A	初版
2007 年 11 月	SH(NA)-080450CHN-B	全面改版

英文手册版本： SH-080039-M

此手册无任何工业产权，无如何其它类型产权，也无任何专利。三菱电机公司不对由于使用本手册中的说明而造成的任何可能包含的工业产权问题负责。

© 2003 三菱电机

前言

感谢使用三菱电机 MELSEC-Q 系列 (Q 模式) 和 MELSEC-QnA 系列可编程控制器。
在使用这些产品之前，请仔细阅读本手册，使你能对所使用的 Q 系列 (Q 模式)/QnA 系列可编程控制器的功能和性能足够熟悉，以便能保证正确使用该类产品。
应该送给最终用户此手册的一个副本。

目录

公共指令 1/2

安全警告 ..... A - 1
修订本 ..... A - 2
前言 ..... A - 3
目录 ..... A - 3
关于手册 ..... A - 14

1. 概述 1 - 1 到 1 - 6
1.1 相关编程手册 1 - 2
1.2 本手册中使用的总称及略称 1 - 5

2. 概述 2 - 1 到 2 - 56
2.1 指令类型 2 - 2
2.2 如何阅读指令表 2 - 4
2.3 顺序指令 2 - 6
2.3.1 触点指令 ..... 2 - 6
2.3.2 连接指令 ..... 2 - 7
2.3.3 输出指令 ..... 2 - 8
2.3.4 移位指令 ..... 2 - 8
2.3.5 主控制指令 ..... 2 - 9
2.3.6 终止指令 ..... 2 - 9
2.3.7 其它指令 ..... 2 - 9
2.4 基本指令 2 - 10
2.4.1 比较操作指令 ..... 2 - 10
2.4.2 算术运算指令 ..... 2 - 16
2.4.3 数据转换指令 ..... 2 - 21
2.4.4 数据传送指令 ..... 2 - 23
2.4.5 程序分支指令 ..... 2 - 25
2.4.6 程序执行控制指令 ..... 2 - 25
2.4.7 I/O 刷新指令 ..... 2 - 25
2.4.8 其它使用方便的指令 ..... 2 - 26
2.5 应用指令 2 - 27
2.5.1 逻辑操作指令 ..... 2 - 27
2.5.2 循环指令 ..... 2 - 30
2.5.3 位指令 ..... 2 - 31
2.5.4 处理指令 ..... 2 - 31
2.5.5 据处理指令 ..... 2 - 32
2.5.6 结构体创建指令 ..... 2 - 35
2.5.7 数据表操作指令 ..... 2 - 37
2.5.8 缓冲存储区访问指令 ..... 2 - 38
2.5.9 显示指令 ..... 2 - 38

2.5.10 调试和故障诊断指令 ..... 2 - 39
2.5.11 字符串处理指令 ..... 2 - 40
2.5.12 特殊功能指令 ..... 2 - 43
2.5.13 数据控制指令 ..... 2 - 46
2.5.14 转换指令 ..... 2 - 47
2.5.15 时钟指令 ..... 2 - 48
2.5.16 外围软元件指令 ..... 2 - 49
2.5.17 程序控制指令 ..... 2 - 49
2.5.18 其他指令 ..... 2 - 50
2.5.19 用于数据链接的指令 ..... 2 - 51
2.5.20 QCPU 指令 ..... 2 - 53
2.5.21 冗余系统指令（用于 Q4ARCPU) ..... 2 - 55
2.5.22 冗余系统指令（用于冗余 CPU) ..... 2 - 55

3. 指令组态 3 - 1 到 3 - 40
3.1 指令结构 3 - 2
3.2 数据的指定方法 3 - 3
3.2.1 使用位数据 ..... 3 - 3
3.2.2 使用字（16 位）数据 ..... 3 - 4
3.2.3 使用双字数据（32 位）..... 3 - 6
3.2.4 使用实数数据 ..... 3 - 8
3.2.5 使用字符串数据 ..... 3 - 11
3.3 变址修饰 3 - 12
3.4 间接指定 3 - 18
3.5 缩短指令处理时间 3 - 20
3.5.1 子集处理 ..... 3 - 20
3.5.2 使用通用运算寄存器（Z）的运算处理（只对于通用型 QCPU) ..... 3 - 21
3.6 编程注意事项（运算错误） 3 - 22
3.7 指令执行条件 3 - 27
3.8 计算步数 3 - 28
3.9 当 OUT、SET/RST 或 PLS/PLF 指令使用相同的软元件时的操作 3 - 32
3.10 使用文件寄存器的注意事项 3 - 37

4. 如何阅读指令 4 - 1 到 4 - 4

5. 顺序指令 5 - 1 到 5 - 56
5.1 触点指令 5 - 2
5.1.1 运行开始，串行连结，并行连接
（LD、LDI、AND、ANI、OR、ORI) ..... 5 - 2
5.1.2 脉冲运行开始，脉冲串行连接，脉冲并行连接（LDP、LDF、ANDP、ANDF、ORP、ORF) ..... 5 - 5
5.2 连接指令 5 - 7
5.2.1 梯形图块串行连接和并行连接
（ANB、ORB) ..... 5 - 7
5.2.2 运行结果推进，读取，弹出（MPS、MRD、MPP) ..... 5 - 9
5.2.3 运行结果反转（INV) ..... 5 - 12
5.2.4 运算结果脉冲变换（MEP、MEF) ..... 5 - 13
5.2.5 变址继电器运算结果的脉冲变换（EGP、EGF) ..... 5 - 14

5.3 输出指令 5 - 16
5.3.1 输出指令（不包括定时器，计数器，和报警器）(OUT) ..... 5 - 16
5.3.2 定时器 (OUT T、OUTH T) ..... 5 - 18
5.3.3 计数器 (OUT C) ..... 5 - 22
5.3.4 报警器输出 (OUT F) ..... 5 - 24
5.3.5 设定软元件（报警器除外） (SET) ..... 5 - 26
5.3.6 复位软元件（除了报警器）(RST) ..... 5 - 28
5.3.7 设定和复位报警器 (SET F、RST F) ..... 5 - 33
5.3.8 上升沿和下降沿输出 (PLS、PLF) ..... 5 - 36
5.3.9 位软元件输出取反 (FF) ..... 5 - 38
5.3.10 直接输出的脉冲变换 (DELTA(P)) ..... 5 - 40
5.4 移动指令 5 - 40
5.4.1 位软元件移动 (SFT(P)) ..... 5 - 40
5.5 主控指令 5 - 43
5.5.1 设置和复位主控 (MC、MCR) ..... 5 - 43
5.6 结束指令 5 - 47
5.6.1 结束主程序 (FEND) ..... 5 - 47
5.6.2 结束顺控程序 (END) ..... 5 - 49
5.7 其他指令 5 - 51
5.7.1 顺控程序暂停 (STOP) ..... 5 - 51
5.7.2 无处理 (NOP、NOPLF、PAGE n) ..... 5 - 53

6. 基本指令 6 - 1 到 6 - 162
6.1 比较运算指令 6 - 2
6.1.1 BIN 16 位数据比较 (=、<>、>、<=、<、>=) ..... 6 - 2
6.1.2 BIN 32 位数据比较 (D=、D<>、D>、D<=、D<、D>=) ..... 6 - 4
6.1.3 浮动小数点数据比较 (E=、E<>、E>、E<=、E<、E>=) ..... 6 - 6
6.1.4 浮动小数点数据比较（双精度）
(ED=、ED<>、ED>、ED<=、ED<、ED>=) ..... 6 - 8
6.1.5 字符串数据比较 ($=、$<>、$>、$<=、$<、$>=) ..... 6 - 11
6.1.6 BIN 块数据比较 (BKCMP □ 、BKCMP □ P) ..... 6 - 15
6.2 算术运算指令 6 - 19
6.2.1 BIN 16 位加法和减法运算 (+(P)、-(P)) ..... 6 - 19
6.2.2 BIN 32 位加法和减法运算 (D+(P)、D-(P)) ..... 6 - 23
6.2.3 BIN 16 位乘法和除法运算 (*(P)、/(P)) ..... 6 - 27
6.2.4 BIN 32 位乘法和除法运算 (D*(P)、D/(P)) ..... 6 - 29
6.2.5 BCD 4 位加法和减法运算 (B+(P)、B-(P)) ..... 6 - 31
6.2.6 BCD 8 位加法和减法运算 (DB+(P)、DB-(P)) ..... 6 - 35
6.2.7 BCD 4 位乘法和除法运算 (B*(P)、B/(P)) ..... 6 - 39
6.2.8 BCD 8 位乘法和除法运算 (DB*(P)、DB/(P)) ..... 6 - 41
6.2.9 浮点数据的加法和减法运算 (E+(P)、E-(P)) ..... 6 - 43
6.2.10 浮点数据的加法和减法运算（双精度）(ED+(P)、ED-(P)) ..... 6 - 47
6.2.11 浮点数据的乘法和除法运算 (E*(P)、E/(P)) ..... 6 - 51
6.2.12 浮点数据的乘法和除法运算（双精度）(ED*(P)、ED/(P)) ..... 6 - 53
6.2.13 块加法和减法运算 (BK+(P)、BK-(P)) ..... 6 - 56
6.2.14 链接字符串 ($+(P)) ..... 6 - 59
6.2.15 16 位 BIN 数据的递增和递减运算 (INC(P)、DEC(P)) ..... 6 - 63
6.2.16 32 位 BIN 数据的递增和递减运算 (DINC(P)、DDEC(P)) ..... 6 - 65
6.3 数据转换指令 6 - 67
6.3.1 从 BIN 数据到 4 位和 8 位 BCD 数据的转换

(BCD(P)、DBCD(P)) ..... 6 - 67
6.3.2 从 BCD 4 位和 8 位数据到 BIN 数据的转换 (BIN(P)、DBIN(P)) ..... 6 - 69
6.3.3 从 BIN 16 位和 32 位数据到浮点数据的转换
(FLT(P)、DFLT(P)) ..... 6 - 72
6.3.4 从 BIN16 和 32 位数据到浮点数据的转换 (双精度)
(FLTD(P)、DFLTD(P)) ..... 6 - 75
6.3.5 从浮点数据到 BIN 16 位和 32 位数据的转换
(INT(P)、DINT(P)) ..... 6 - 77
6.3.6 从浮点数据到 BIN16 位 /32 位数据的转换 (双精度)
(INTD(P)、DINTD(P)) ..... 6 - 80
6.3.7 从 BIN 16 位数据到 BIN 32 位数据的转换 (DBL(P)) ..... 6 - 82
6.3.8 从 BIN 32 位到 BIN 16 位数据的转换 (WORD(P)) ..... 6 - 83
6.3.9 从 BIN 16 位和 32 位数据到格雷码的转换
(GRY(P)、DGRY(P)) ..... 6 - 84
6.3.10 从格雷码到 BIN 16 位和 32 位数据的转换
(GBIN(P)、DGBIN(P)) ..... 6 - 86
6.3.11 BIN 16 位和 32 位数据的 2 进制补码 (符号取反)
(NEG(P)、DNEG(P)) ..... 6 - 88
6.3.12 浮点数据的符号取反 (单精度)(ENEG(P)) ..... 6 - 90
6.3.13 浮点数据的符号取反 (双精度)(EDNEG(P)) ..... 6 - 91
6.3.14 从块 BIN 16 位数据到 BCD 4 位数据的转换 (BKBCD(P)) ..... 6 - 92
6.3.15 从块 BCD 4 位数据到块 BIN 16 位数据的转换 (BKBIN(P)) ..... 6 - 94
6.3.16 单精度→双精度转换 (ECON(P)) ..... 6 - 96
6.3.17 双精度→单精度转换 (EDCON(P)) ..... 6 - 98
6.4 数据传送指令 6 - 100
6.4.1 16 位和 32 位数据传送 (MOV(P)、DMOV(P)) ..... 6 - 100
6.4.2 浮点数据传送 (EMOV(P)) ..... 6 - 102
6.4.3 浮点数据传送 (双精度) (EDMOV(P)) ..... 6 - 104
6.4.4 字符串传送 ($MOV(P)) ..... 6 - 106
6.4.5 16 位和 32 位数据取反传送 (CML(P)、DCML(P)) ..... 6 - 108
6.4.6 块 16 位数据传送 (BMOV(P)) ..... 6 - 111
6.4.7 相同的 16 位数据块传送 (FMOV(P)) ..... 6 - 114
6.4.8 16 位和 32 位数据交换 (XCH(P)、DXCH(P)) ..... 6 - 116
6.4.9 块 16 位数据交换 (BXCH(P)) ..... 6 - 118
6.4.10 高字节和低字节交换 (SWAP(P)) ..... 6 - 120
6.5 程序分支指令 6 - 121
6.5.1 指针分支指令 (CJ、SCJ、JMP) ..... 6 - 121
6.5.2 跳转到 END(GOEND) ..... 6 - 124
6.6 程序执行控制指令 6 - 125
6.6.1 中断禁止 / 允许指令，中断程序屏蔽 (DI、EI、IMASK) ..... 6 - 125
6.6.2 从中断程序中恢复 (IRET) ..... 6 - 134
6.7 I/O 刷新指令 6 - 136
6.7.1 I/O 刷新 (RFS(P)) ..... 6 - 136
6.8 其它方便的指令 6 - 138
6.8.1 单相输入加法或减法计数器 (UDCNT1) ..... 6 - 138
6.8.2 两相输入加法或减法计数器 (UDCNT2) ..... 6 - 141
6.8.3 教学计时器 (TTMR) ..... 6 - 144
6.8.4 特殊功能定时器 (STMR) ..... 6 - 146
6.8.5 旋转台就近控制 (ROTC) ..... 6 - 149
6.8.6 斜坡信号 (RAMP) ..... 6 - 152

6.8.7 脉冲密度测量 (SPD) ..... 6 - 155
6.8.8 固定周期脉冲输出 (PLSY) ..... 6 - 157
6.8.9 脉冲宽度调制 (PWM) ..... 6 - 159
6.8.10 矩阵输入 (MTR) ..... 6 - 161

7. 应用指令 7 - 1 到 7 - 374

7.1 逻辑运算指令 7 - 2
7.1.1 16 位和 32 位数据的逻辑乘 (WAND(P)、DAND(P)) ..... 7 - 3
7.1.2 块逻辑乘 (BKAND(P)) ..... 7 - 9
7.1.3 16 位和 32 位数据的逻辑和 (WOR(P)、DOR(P)) ..... 7 - 11
7.1.4 块逻辑和运算 (BKOR(P)) ..... 7 - 17
7.1.5 16 位和 32 位数据异或运算 (WXOR(P)、DXOR(P)) ..... 7 - 19
7.1.6 块异或运算 (BKXOR(P)) ..... 7 - 25
7.1.7 16 位和 32 位异或非运算 (WXNR(P)、DXNR(P)) ..... 7 - 27
7.1.8 块异或非运算 (BKXNR(P)) ..... 7 - 33
7.2 循环指令 7 - 35
7.2.1 16 位数据的右循环 (ROR(P)、RCR(P)) ..... 7 - 35
7.2.2 16 位数据左循环 (ROL(P)、RCL(P)) ..... 7 - 38
7.2.3 32 位数据的右循环 (DROR(P)、DRCR(P)) ..... 7 - 41
7.2.4 32 位数据左循环 (DROL(P)、DRCL(P)) ..... 7 - 44
7.3 移位指令 7 - 46
7.3.1 16 位数据的 n 位左移或右移 (SFR(P)、SFL(P)) ..... 7 - 46
7.3.2 n 位数据的 1 位左移或右移 (BSFR(P)、BSFL(P)) ..... 7 - 49
7.3.3 n 字数据的 1 字左移或右移 (DSFR(P)、DSFL(P)) ..... 7 - 51
7.4 位处理指令 7 - 53
7.4.1 字软元件的位设定和复位 (BSET(P)、BRST(P)) ..... 7 - 53
7.4.2 位测试 (TEST(P)、DTEST(P)) ..... 7 - 55
7.4.3 位软元件的成批复位 (BKRST(P)) ..... 7 - 58
7.5 数据处理指令 7 - 60
7.5.1 16 位和 32 位数据搜索 (SER(P)、DSER(P)) ..... 7 - 60
7.5.2 16 位和 32 位数据的位数据检查 (SUM(P)、DSUM(P)) ..... 7 - 63
7.5.3 8 位到 256 位的解码 (DECO(P)) ..... 7 - 65
7.5.4 将 256 位编码为 8 位 (ENCO(P)) ..... 7 - 67
7.5.5 7 段解码 (SEG(P)) ..... 7 - 69
7.5.6 16 位数据的 4 位分组 (DIS(P)) ..... 7 - 71
7.5.7 16 位数据的 4 位链接 (UNI(P)) ..... 7 - 73
7.5.8 任意数据的分离或链接 (NDIS(P)、NUNI(P)) ..... 7 - 75
7.5.9 以字节为单位的数据分离和链接 (WTOB(P)、BTOW(P)) ..... 7 - 79
7.5.10 16 位和 32 位数据的最大值查找 (MAX(P)、DMAX(P)) ..... 7 - 83
7.5.11 查找 16 位和 32 位数据的最小值 (MIN(P)、DMIN(P)) ..... 7 - 86
7.5.12 BIN16 位和 32 位数据的排序运算 (SORT、DSORT) ..... 7 - 89
7.5.13 16 位数据的总和计算 (WSUM(P)) ..... 7 - 93
7.5.14 32 位数据总和计算 (DWSUM(P)) ..... 7 - 95
7.6 结构化程序指令 7 - 97
7.6.1 FOR 到 NEXT 指令循环 (FOR、NEXT) ..... 7 - 97
7.6.2 FOR 到 NEXT 指令循环的强制结束 (BREAK(P)) ..... 7 - 100
7.6.3 子程序调用 (CALL(P)) ..... 7 - 102
7.6.4 从子程序返回 (RET) ..... 7 - 107
7.6.5 子程序输出 OFF 调用 (FCALL(P)) ..... 7 - 108
7.6.6 程序文件之间的子程序调用 (ECALL(P)) ..... 7 - 112

7.6.7 程序文件之间的子程序输出 OFF 调用 (EFCALL(P)) ..... 7 - 117
7.6.8 子程序调用 (XCALL) ..... 7 - 121
7.6.9 刷新指令 (COM) ..... 7 - 126
7.6.10 整个梯形图的变址修饰 (IX、IXEND) ..... 7 - 129
7.6.11 整个梯形图的变址修饰中修饰值的指定 (IXDEV、IXSET) ..... 7 - 133
7.7 数据表格操作指令 7 - 136
7.7.1 将数据写入数据表格 (FIFW(P)) ..... 7 - 136
7.7.2 从表格中读取最旧的数据 (FIFR(P)) ..... 7 - 138
7.7.3 从数据表格中读取最新数据 (FPOP(P)) ..... 7 - 140
7.7.4 从数据表格中删除数据和在数据表格中插入数据 (FDEL(P)、FINS(P)) ..... 7 - 142
7.8 缓冲存储区访问指令 7 - 145
7.8.1 从智能功能模块中读取 1 字 /2 字数据 (FROM(P)、DFRO(P)) ..... 7 - 145
7.8.2 将 1 字 /2 字数据写入智能功能模块 (TO(P)、DTO(P)) ..... 7 - 148
7.9 显示指令 7 - 151
7.9.1 打印 ASCII 码指令 (PR) ..... 7 - 151
7.9.2 打印注释指令 (PRC) ..... 7 - 154
7.9.3 ASCII 码 LED 显示指令 (LED) ..... 7 - 159
7.9.4 注释的 LED 显示指令 (LEDC) ..... 7 - 161
7.9.5 错误显示和报警器复位指令 (LEDR) ..... 7 - 163
7.10 调试和故障诊断指令 7 - 166
7.10.1 特殊格式故障检测 (CHKST、CHK) ..... 7 - 166
7.10.2 改变 CHK 指令的检测格式 (CHKCIR、CHKEND) ..... 7 - 170
7.10.3 设置和复位状态锁存 (SLT、SLTR) ..... 7 - 178
7.10.4 设置和复位采样跟踪 (STRA、STRAR) ..... 7 - 180
7.10.5 程序跟踪的执行、设置和复位 (PTRAEXE(P)、PTRA、PTRAR) ..... 7 - 182
7.11 字符串处理指令 7 - 184
7.11.1 从 BIN16 位或 32 位到十进制 ASCII 码的转换 (BINDA(P)、DBINDA(P)) ..... 7 - 184
7.11.2 BIN16 位或 32 位数据到十六进制 ASCII 码的转换 (BINHA(P)、DBINHA(P)) ..... 7 - 187
7.11.3 从 BCD4 位和 8 位数据到十进制 ASCII 码的转换 (BCDDA(P)、DBCDDA(P)) ..... 7 - 190
7.11.4 从十进制 ASCII 码到 BIN16 位和 32 位数据的转换 (DABIN(P)、DDABIN(P)) ..... 7 - 193
7.11.5 从十六进制 ASCII 到 BIN16 位和 32 位数据的转换 (HABIN(P)、DHABIN(P)) ..... 7 - 196
7.11.6 从十进制 ASCII 码到 BCD4 位或 8 位数据的转换 (DABCD(P)、DDABCD(P)) ..... 7 - 199
7.11.7 读取软元件注释数据 (COMRD(P)) ..... 7 - 202
7.11.8 字符串长度检测 (LEN(P)) ..... 7 - 207
7.11.9 从 BIN16 位或 32 位到字符串的转换 (STR(P)、DSTR(P)) ..... 7 - 209
7.11.10 从字符串到 BIN16 位或 32 位数据的转换 (VAL(P)、DVAL(P)) ..... 7 - 215
7.11.11 从浮点型到字符串数据的转换 (ESTR(P)) ..... 7 - 220
7.11.12 从字符串到浮点数的转换 (EVAL(P)) ..... 7 - 226
7.11.13 从十六进制 BIN 到 ASCII 码的转换 (ASC(P)) ..... 7 - 230
7.11.14 从 ASCII 码到十六进制 BIN 的转换 (HEX(P)) ..... 7 - 232
7.11.15 从字符串的右边或左边提取数据 (RIGHT(P)、LEFT(P)) ..... 7 - 234
7.11.16 字符串的任意选择和替换 (MIDR(P)、MIDW(P)) ..... 7 - 237
7.11.17 字符串查找 (INSTR(P)) ..... 7 - 241
7.11.18 浮点数到 BCD 的分解 (EMOD(P)) ..... 7 - 243

7.11.19 从 BCD 格式数据到浮点数 (EREXP(P)) ..... 7 - 246
7.12 特殊函数指令 ..... 7 - 248
7.12.1 浮点数的 SIN 运算 (单精度)(SIN(P)) ..... 7 - 248
7.12.2 浮点数的 SIN 运算 (双精度)(SIND(P)) ..... 7 - 250
7.12.3 浮点数的 COS 运算 (单精度)(COS(P)) ..... 7 - 252
7.12.4 浮点数的 COS 运算 (双精度) (COSD(P)) ..... 7 - 254
7.12.5 浮点数的 TAN 运算 (单精度)(TAN(P)) ..... 7 - 256
7.12.6 浮点数的 TAN 运算 (双精度)(TAND(P)) ..... 7 - 258
7.12.7 浮点数的 SIN^{-1} 运算 (单精度)(ASIN(P)) ..... 7 - 260
7.12.8 浮点数的 SIN^{-1} 运算 (双精度)(ASIND(P)) ..... 7 - 263
7.12.9 浮点数的 COS^{-1} 运算 (单精度)(ACOS(P)) ..... 7 - 265
7.12.10 浮点数的 COS^{-1} 运算 (双精度)(ACOSD(P)) ..... 7 - 267
7.12.11 浮点数 TAN^{-1} 运算 (单精度)(ATAN(P)) ..... 7 - 269
7.12.12 浮点数 TAN^{-1} 运算 (双精度)(ATAND(P)) ..... 7 - 271
7.12.13 从浮点数角度到弧度的转换 (单精度)(RAD(P)) ..... 7 - 273
7.12.14 从浮点数角度到弧度的转换 (双精度)(RADD(P)) ..... 7 - 275
7.12.15 从浮点数弧度到角度的转换 (单精度)(DEG(P)) ..... 7 - 277
7.12.16 从浮点数弧度到角度的转换 (双精度)(DEGD(P)) ..... 7 - 279
7.12.17 浮点数的平方根运算 (单精度)(SQR(P)) ..... 7 - 281
7.12.18 浮点数的平方根运算 (双精度)(SQRD(P)) ..... 7 - 283
7.12.19 浮点数的指数运算 (单精度)(EXP(P)) ..... 7 - 285
7.12.20 浮点数的指数运算 (双精度)(EXPD(P)) ..... 7 - 288
7.12.21 浮点数的自然对数运算 (单精度)(LOG(P)) ..... 7 - 290
7.12.22 浮点数的自然对数运算 (双精度)(LOGD(P)) ..... 7 - 292
7.12.23 随机数的产生和系列更新 (RND(P)、SRND(P)) ..... 7 - 294
7.12.24 BCD4 位和 8 位平方根 (BSQR(P)、BDSQR(P)) ..... 7 - 296
7.12.25 BCD 型 SIN 运算 (BSIN(P)) ..... 7 - 299
7.12.26 BCD 型 COS 运算 (BCOS(P)) ..... 7 - 301
7.12.27 BCD 型 TAN 运算 (BTAN(P)) ..... 7 - 303
7.12.28 BCD 型 SIN^{-1} 运算 (BASIN(P)) ..... 7 - 305
7.12.29 BCD 型 COS^{-1} 运算 (BACOS(P)) ..... 7 - 307
7.12.30 BCD 型 TAN^{-1} 运算 (BATAN(P)) ..... 7 - 309
7.13 数据控制指令 ..... 7 - 311
7.13.1 BIN 16 位和 BIN 32 位数据的高低限控制 (LIMIT(P)、DLIMIT(P)) ..... 7 - 311
7.13.2 BIN 16 位和 32 位死区控制 (BAND(P)、DBAND(P)) ..... 7 - 314
7.13.3 BIN 16 位和 BIN 32 位数据的区域控制 (ZONE(P)、DZONE(P)) ..... 7 - 317
7.14 文件寄存器切换指令 ..... 7 - 320
7.14.1 切换文件寄存器号 (RSET(P)) ..... 7 - 320
7.14.2 文件寄存器使用的设置文件 (QDRSET(P)) ..... 7 - 322
7.14.3 注释的文件设置 (QCDSET(P)) ..... 7 - 325
7.15 时钟指令 ..... 7 - 327
7.15.1 读时钟数据 (DATERD(P)) ..... 7 - 327
7.15.2 写时钟数据 (DATEWR(P)) ..... 7 - 331
7.15.3 时钟数据加法运算 (DATE+(P)) ..... 7 - 335
7.15.4 时钟数据减法运算 (DATE-(P)) ..... 7 - 337
7.15.5 时间数据转换 (从小时 / 分钟 / 秒到秒)(SECOND(P)) ..... 7 - 339
7.15.6 时间数据转换 (从秒到小时 / 分钟 / 秒)(HOUR(P)) ..... 7 - 341
7.16 外围设备指令 ..... 7 - 343

7.16.1 将信息显示到外围设备 (MSG) ..... 7 - 343
7.16.2 外围设备的键盘输入 (PKEY) ..... 7 - 345
7.17 程序控制指令 7 - 347
7.17.1 程序待机指令 (PSTOP(P)) ..... 7 - 349
7.17.2 程序输出 OFF 待机指令 (POFF(P)) ..... 7 - 350
7.17.3 程序扫描执行登记指令 (PSCAN(P)) ..... 7 - 352
7.17.4 程序低速执行登记指令 (PLOW(P)) ..... 7 - 354
7.17.5 程序执行状态检查指令 (PCHK) ..... 7 - 356
7.18 其它指令 7 - 358
7.18.1 复位看门狗定时器 (WDT(P)) ..... 7 - 358
7.18.2 定时脉冲发生 (DUTY) ..... 7 - 360
7.18.3 时间检查指令 (TIMCHK) ..... 7 - 362
7.18.4 从文件寄存器中的直接 1 字节读取 (ZRRDB(P)) ..... 7 - 363
7.18.5 写入 1 字节到文件寄存器 (ZRWRB(P)) ..... 7 - 365
7.18.6 间接地址读取操作 (ADRSET(P)) ..... 7 - 367
7.18.7 键盘的数字键输入 (KEY) ..... 7 - 368
7.18.8 批量保存或变址寄存器的恢复 (ZPUSH(P)、ZPOP(P)) ..... 7 - 372
7.18.9 对 E^2PROM 文件寄存器的批写入操作 (EROMWR(P)) ..... 7 - 374

公共指令 2/2

8. 数据链接指令 8 - 1 到 8 - 136
8.1 概要 8 - 2
8.2 网络刷新指令 8 - 10
8.2.1 网络刷新 (S(P)/J(P)/G(P).ZCOM) ..... 8 - 10
8.3 QnA 链接专用指令 8 - 18
8.3.1 从其它站中读取字软元件数据 (JP/GP.READ) ..... 8 - 18
8.3.2 从其它站读取字软元件数据 (JP/GP.SREAD) ..... 8 - 29
8.3.3 写入软元件数据到其它站 (JP/GP.WRITE) ..... 8 - 40
8.3.4 将软元件数据写入到其它站 (JP/GP.SWRITE) ..... 8 - 51
8.3.5 传送数据到其它站 (JP/GP.SEND) ..... 8 - 62
8.3.6 从其它站接收数据 (JP/GP.RECV) ..... 8 - 72
8.3.7 来自其它站的瞬时请求 (时钟数据的读 / 写、
远程 RUN/STOP)(JP/GP.REQ) ..... 8 - 81
8.3.8 从远程 I/O 站的特殊功能模块中读数据 (JP/GP.ZNFR) ..... 8 - 95
8.3.9 往远程 I/O 站的特殊功能模块中写数据 (JP/GP.ZNTO) ..... 8 - 100
8.4 A 系列兼容链接指令 8 - 105
8.4.1 从其它站读软元件数据 (MELSECNET/10、Ethernet) (J(P).ZNRD) ..... 8 - 105
8.4.2 从本地站 (MELSECNET) 读软元件数据 (J(P).ZNRD) ..... 8 - 109
8.4.3 将软元件数据写入其它站 (MELSECNET/10、以太网) (J(P).ZNWR) ..... 8 - 113
8.4.4 将数据写入本地站软元件中 (MELSECNET)(J(P).ZNWR) ..... 8 - 117
8.4.5 从远程 I/O 站的特殊功能模块读数据 (MELSECNET)(G(P).RFRP) ..... 8 - 121
8.4.6 将数据写入远程 I/O 站特殊功能模块 (MELSECNET)(G(P).RTOP) ..... 8 - 125
8.5 读 / 写路由信息 8 - 129
8.5.1 读路由信息 (S(P)/Z(P).RTREAD) ..... 8 - 129
8.5.2 登录路由信息 (S(P)/Z(P).RTWRITE) ..... 8 - 133
9. QCPU 指令 9 - 1 到 9 - 74
9.1 读取模块信息 (UNIRD(P)) 9 - 2

9.2 追踪置位 / 复位 (TRACE、TRACER) 9 - 6
9.3 写数据到指定的文件 (SP.FWRITE) 9 - 8
9.4 从指定文件中读取数据 (SP.FREAD) 9 - 19
9.5 向标准 ROM 中写入数据 (SP.DEVST) 9 - 31
9.6 从标准 ROM 中读取数据 (S(P).DEVLD) 9 - 33
9.7 从内存卡中装载程序 (PLOADP) 9 - 35
9.8 从程序内存中卸载程序 (PUNLOADP) 9 - 38
9.9 装载 + 卸载 (PSWAPP) 9 - 40
9.10 文件寄存器的高速块传送 (RBMOV(P)) 9 - 43
9.11 写入自站 CPU 共享内存 9 - 47
9.11.1 写入到自站 CPU 共享内存 (S(P).TO) 9 - 49
9.11.2 写入到自站 CPU 共享内存 (TO(P)、DTO(P)) 9 - 52
9.12 从其他站的 CPU 共享内存中读取数据 9 - 56
9.12.1 从其他站的共享内存读取数据 (FROM(P)、DFRO(P)) 9 - 57
9.13 选择性刷新指令 (COM) 9 - 63
9.14 扩展时钟指令 9 - 66
9.14.1 读取扩展时钟数据 (S(P).DATERD) 9 - 66
9.14.2 扩展时钟数据的加法运算 (S(P).DATE+) 9 - 69
9.14.3 扩展时钟数据的减法运算 (S(P).DATE-) 9 - 72

10. 冗余系统指令 (用于 Q4ARCPU) 10 - 1 到 10 - 14
10.1 CPU 启动过程中的操作模式设定指令 (S.STMODE) 10 - 2
10.2 CPU 切换时间操作模式设定指令 (S.CGMODE) 10 - 4
10.3 数据追踪指令 (S.TRUCK) 10 - 6
10.4 缓冲内存批刷新指令 (S.SPREF) 10 - 10

11. 冗余系统指令 (用于冗用系统 CPU) 11 - 1 到 11 - 4
11.1 系统切换指令 (SP.CONTSW) 11 - 2

12. 出错代码 12 - 1 到 12 - 126
12.1 出错代码表 12 - 2
12.1.1 出错代码 12 - 3
12.1.2 出错代码的读取方法 12 - 3
12.1.3 出错代码一览表 (1000 ～ 1999) 12 - 4
12.1.4 出错代码表 (2000 ～ 2999) 12 - 26
12.1.5 出错代码表 (3000 ～ 3999) 12 - 56
12.1.6 出错代码表 (4000 ～ 4999) 12 - 78
12.1.7 出错代码表 (5000 ～ 5999) 12 - 96
12.1.8 出错代码表 (6000 ～ 6999) 12 - 100
12.1.9 出错代码表 (7000 ～ 7999) 12 - 120
12.2 出错的解除 12 - 126

附录 附录 - 1 到附录 - 166
附录 1 操作处理时间 附录 - 2
附录 1.1 定义 附录 - 2

附录 1.2 基本型 QCPU 的操作处理时间 附录 - 4
附录 1.3 高性能型 QCPU/ 过程 CPU/ 冗余 CPU/QnACPU 的运算处理时间 附录 - 22
附录 1.4 通用型 QCPU 的运算处理时间 附录 - 50
附录 1.4.1 子集指令处理时间 附录 - 50
附录 1.4.1 子集指令以外的指令处理时间 附录 - 57
附录 2 CPU 之间的性能比较 附录 - 72
附录 2.1 Q/QnACPU 和 AnNCPU、AnACPU 以及 AnUCPU 之间的比较 附录 - 72
附录 2.1.1 可用的软元件 附录 - 72
附录 2.1.2 I/O 控制模式 附录 - 73
附录 2.1.3 可以被指令使用的数据 附录 - 73
附录 2.1.4 定时器比较 附录 - 74
附录 2.1.5 计数器比较 附录 - 75
附录 2.1.6 显示指令比较 附录 - 75
附录 2.1.7 改变定义格式指令（不包括用于 AnACPU 和 AnUCPU 的专用指令） 附录 - 76
附录 2.1.8 AnACPU 和 AnUCPU 专用指令 附录 - 77
附录 2.1.9 只能在一般模式中编程的指令 附录 - 77
附录 3 特殊继电器列表 附录 - 78
附录 4 特殊寄存器列表 附录 - 106
附录 5 应用程序实例 附录 - 165
附录 5.1执行 X^n、$\sqrt[n]{X}$ 操作的程序概念 附录 - 165
索引 索引 - 1 到索引 -10

关于手册

与本产品相关的手册如下所述。

请根据需要参考与借助此表。

相关手册

手册名称	手册号
QCPU 用户手册（硬件设计 / 维护点检篇） 描述了 CPU 模块、电源模块、基板、扩展电缆及存储卡等的规格。 （另售）	SH-080501CHN
QCPU 用户手册（功能解说 / 程序基础篇） 描述了创建程序所需的功能、编程方法和软元件等 （另售）	SH-080503CHN
QCPU(Q 模式)/QnACPU 编程手册 (SFC 控制指令篇) 描述了 MELSAP3 的系统构成、性能规格、功能、程序、调试和出错代码等。（另售）	SH-080283C
QCPU(Q 模式）编程手册 (MELSAP-L 篇) 描述了 MELSAP-L 的系统构成、性能规格、功能、程序、调试和出错代码等。（另售）	SH-080076
QCPU(Q 模式)/QnACPU 编程手册 (PID 控制指令篇) 描述了用于执行 PID 控制的专用指令。（另售）	SH-080240C
QCPU(Q 模式）编程手册（结构化文本篇） 描述了结构化文本 (ST) 语言的编程方法。（另售）	SH-080366E
QnPHCPU/QnPRHCPU 编程手册（过程控制指令篇） 描述了用于执行过程控制的专用指令。（另售）	SH-080449CHN
QnACPU 指南 适用于第一次使用 QnACPU 的用户。描述了从创建程序到将创建好的程序写入 CPU 模块，直到调试的一切步骤。还说明了如何最有效地使用 QnACPU。 （另售）	IB-66606
Q2A(S1)/Q3A/Q4ACPU 用户手册 描述了 Q2ACPU(S1)、Q3ACPU 和 Q4ACPU 的性能、功能函数和处理操作。以及规格说明和电源模块、存储卡和基板的处理操作。 （另售）	IB-66608
冗余 CPU 模块 Q4ARCPU 用户参考手册 描述了 Q4ARCPU 的性能、功能和使用方法。还描述了总线开关模块、系统管理模块、电源模块、存储卡和基板的规格和使用方法。（另售）	SH-080230C
CPU 模块 Q2ASCPU 用户参考手册 描述了 Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU 和 Q2ASHCPU-S1 的性能、功能和处理操作。以及电源模块、存储卡、基板的规格、使用等有关内容。（另售）	SH-080215C
QnACPU 编程参考手册 描述了如何创建程序、设备名称、参数和程序类型。（另售）	SH-080229C
QnACPU 编程手册（特殊功能篇） 描述了当使用 QnACPU 时可用于特殊功能模块的专用指令。（另售）	IB-66616
QnACPU 编程手册 (AD57 指令篇) 描述了当使用 QnACPU 时，可用于控制 AD57(S1) 型 CRT 控制器模块的专用指令集。（另售）	IB-66617

手册名称	手册编号
MELSECNET/G 网络系统参考手册（控制网络篇） 描述了 MELSECNET/G 网络系统的控制网络的规格、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080710CHN
Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇） 描述了 MELSECNET/H 网络系统的可编程控制器网络的规格、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080289C
Q 系列 MELSECNET/H 网络系统参考手册（远程 I/O 网络篇） 描述了 MELSECNET/H 网络系统的远程 I/O 网络规范、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080290C
QnA 系列 MELSECNET/10 网络系统参考手册 描述了 MELSECNET/10 的一般概述、规格、和部件名称及设置。 （另售）	IB-66620
MELSECNET、MELSECNET/B 数据链接系统参考手册 描述了 MELSECNET(II) 和 MELSECNET/B 的一般概述、规格和部件名称及设置。 （另售）	IB-66350
Q 系列以太网接口模块用户手册（应用篇） 描述了以太网模块的 E-mail 功能、可编程控制器 CPU 状态监视、MELSECNET/H 或 MELSECNET/10 网络系统通信功能、数据链接用指令通信功能、文件传输 (FTP 服务器）和其它功能。（另售）	SH-080285C
QnA 系统以太网接口模块用户手册 描述了以太网接口模块的一般概述、规格和部件名称及设置。（另售）	SH-080146
CC-Link 网络系统模块 A1S61BT11/A1SJ61BT11 用户参考手册 描述了 AJ61QBT11 和 A1SJ61QBT11 的一般概述、规格和部件名称及设置。 （另售）	SH-080212C

1 概述

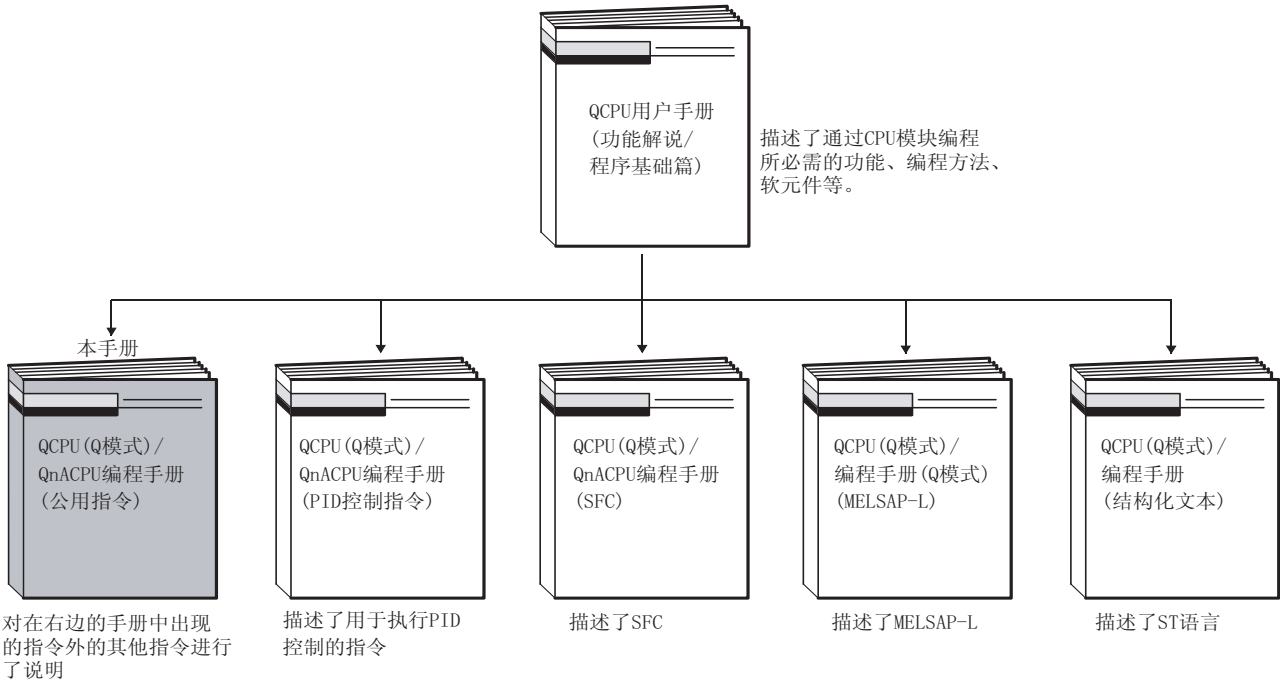
本手册描述用于在 QCPU(Q 模式)、QnACPU 和 Q2AS(H)CPU(S1) 上进行编程的公共指令。
公共指令是除了 AJ71QC24 和 AJ71PT32-S3 的特殊功能模块用指令、用于 AD57 的指令、用于 PID 控制指令、用于 SFC 和 ST 的指令之外的其它所有指令。

1.1 相关编程手册

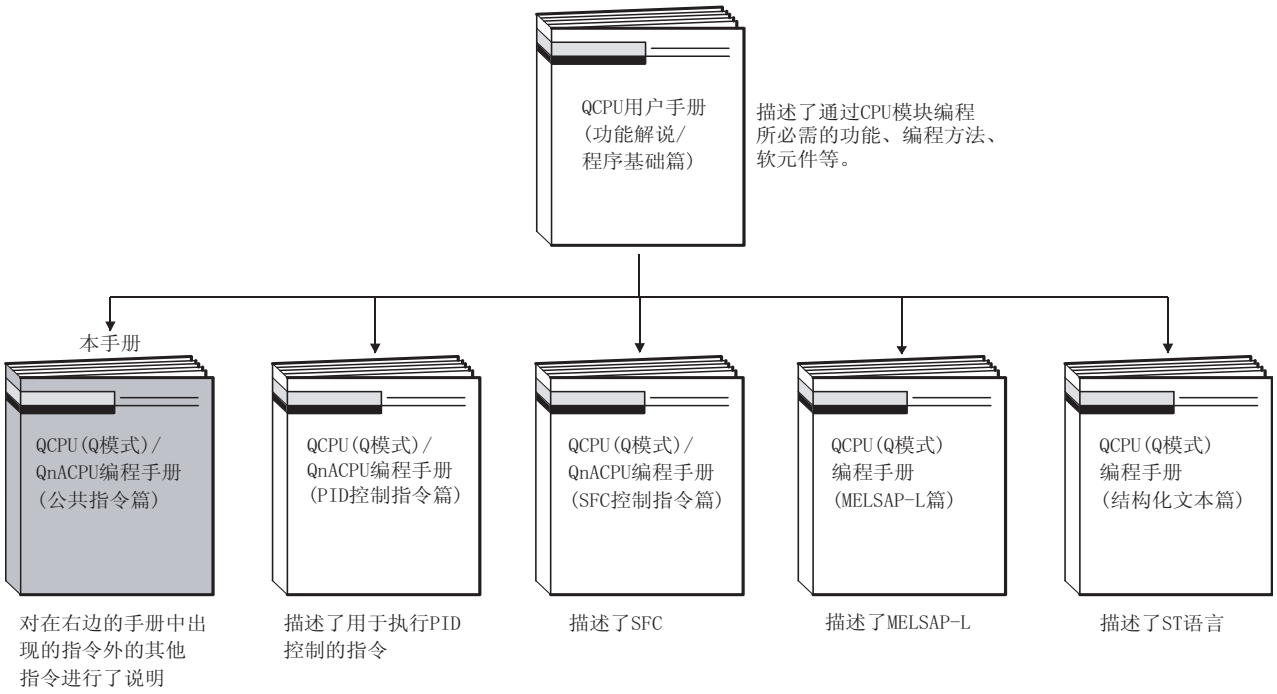
在阅读本手册之前，关于 CPU 模块编程所必需的功能、编程方法、软元件等，请参阅下列手册。

- QCPU 用户手册 (功能解说 / 程序基础篇)
- QnACPU 编程手册 (基础篇)

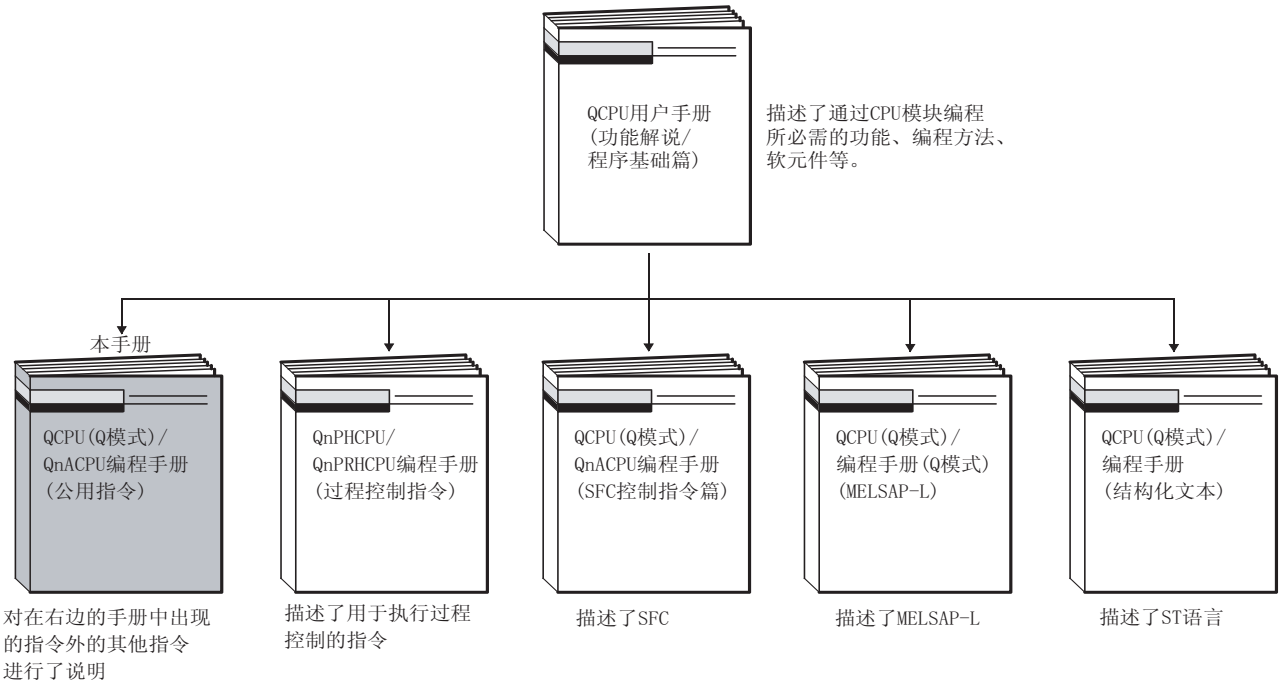
(1) Q02(H)CPU、Q06HCPU、Q12HCPU、Q25HCPU



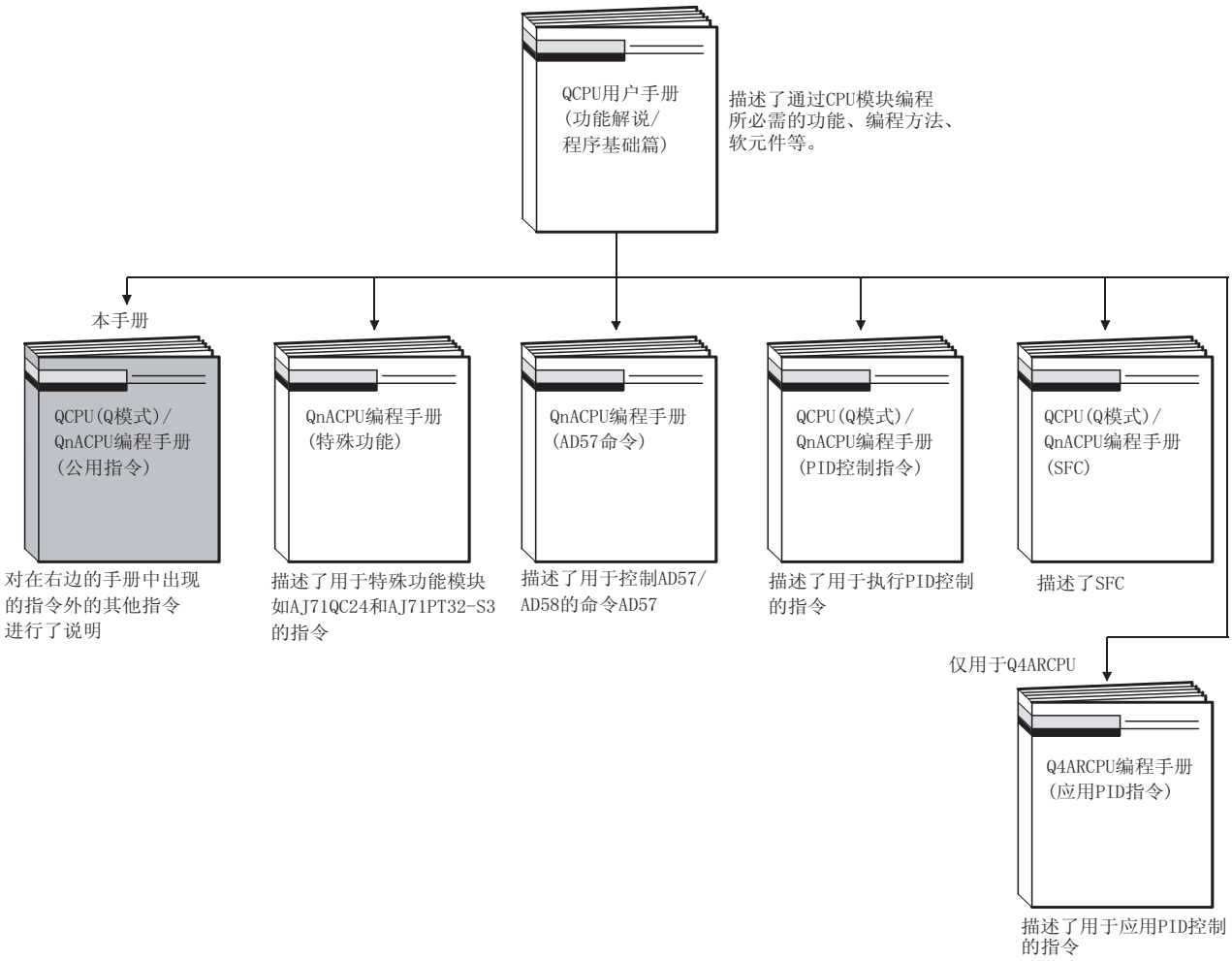
(2) Q00JCPU、Q00CPU、Q01CPU



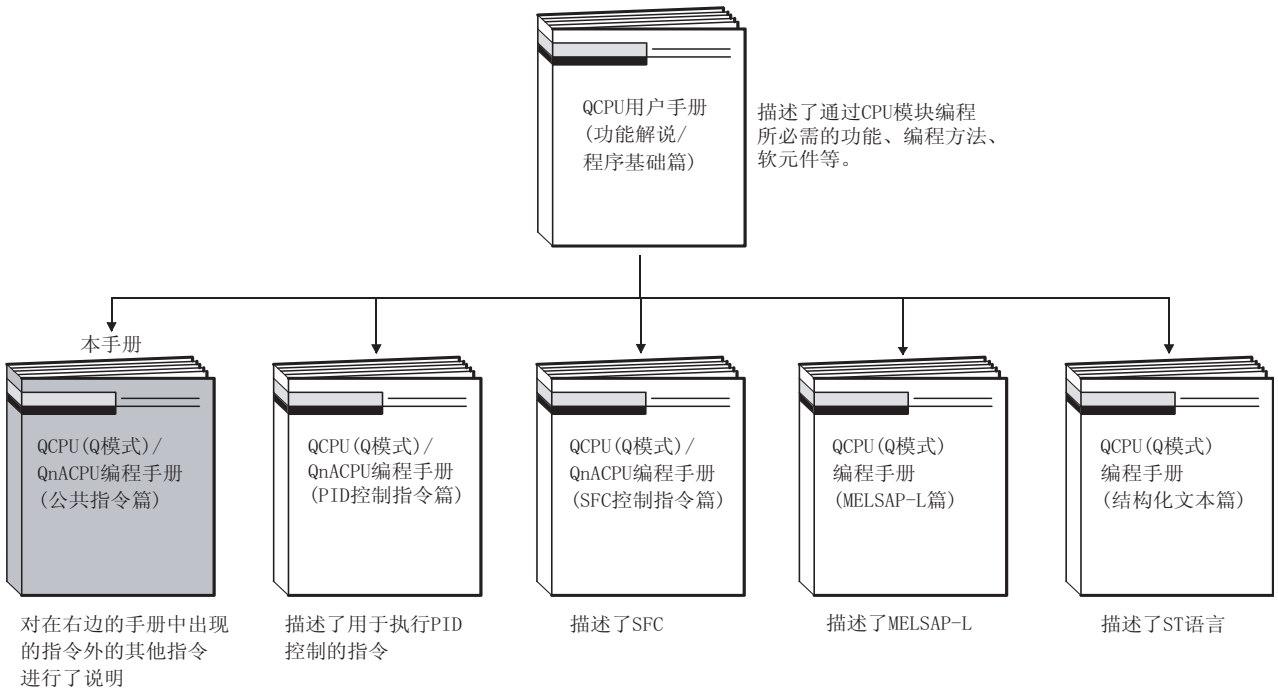
(3) Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU



(4) Q2ACPU、Q3ACPU、Q4ACPU、Q4ARCPU、Q2AS(H)CPU



(5) Q02UCPU、Q03UDCPU、Q04UDHCPU、Q06UDHCPU



1.2 本手册中使用的总称及略称

本手册中除特别注明以外，将使用如下表所示的总称和略称来介绍 Q/QnA 系列 CPU 模块的有关内容。

表 1.1 各模块的总称和略称

总称和略称	总称和略称的内容
基本型 QCPU	Q00JCPU、Q00CPU 和 Q01CPU 的总称。
高性能型 QCPU	Q02CPU、Q02HCPU、Q06HCPU、Q12HCPU 和 Q25HCPU 的总称。
过程 CPU	Q12PHCPU 和 Q25PHCPU 的总称。
冗余 CPU	Q12PRHCPU 和 Q25PRHCPU 的总称。
通用型 QCPU	Q02UCPU、Q03UDCPU、Q04UDHCPU 和 Q06UDHCPU 的总称。
QCPU	基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和通用型 QCPU 的总称。
QnCPU	Q00JCPU、Q00CPU、Q01CPU 和 Q02CPU 型 CPU 模块的总称。
QnHCPU	Q02HCPU、Q06HCPU、Q12HCPU 和 Q25HCPU- 型 CPU 模块。
QnPHCPU	Q12PHCPU 和 Q25PHCPU 型 CPU 模块的总称。
QnPRHCPU	Q12PRHCPU 和 Q25PRHCPU 型 CPU 模块的总称。
QnUCPU	Q02UCPU、Q03UDCPU、Q04UDHCPU 和 Q06UDHCPU 型 CPU 模块的总称。
QnACPU	Q2ACPU、Q2ACPU-S1、Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU、Q2ASHCPU-S1、Q3ACPU 和 Q4ACPU 型 CPU 模块的总称。
Q4ARCPU	Q4ARCPU 型 CPU 模块的总称。
QnA(R)CPU	QnACPU 和 Q4ARCPU 的总称。
ACPU	MELSEC-A 系列可编程控制器 CPU 的总称。
CPU 模块	基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU、QnACPU 和 Q4ARCPU 的总称。
Q 系列	三菱 MELSEC-Q 系列的略称。
QnA 系列	三菱 MELSEC-QnA 系列的略称。
GX Developer	Q 系列兼容 SW□D5C-GPPW 型 GPP 功能软件包的产品名。 □: 软件版本 关于适用各 CPU 模块的 GX Developer 版本，请参考 QCPU 用户手册（硬件设计 / 维护点检篇）的 “系统配置”。

表 1.1 各模块的总称和略称（续）

总称和略称	总称和略称的内容
MELSECNET/G	MELSECNET/G 网络系统的略称。
MELSECNET/H	MELSECNET/H 网络系统的略称。
MELSECNET/10	MELSECNET/10 网络系统的略称。
MELSECNET(II/、B)	MELSECNET 和 MELSECNET/B 数据链接系统的略称。
以太网	以太网网络系统的略称。
CC-Link	Control & Communication Link 的略称。
智能功能模块	智能功能模块和特殊功能模块的总称。
智能功能模块软元件	智能功能模块软元件和特殊功能模块软元件的总称。

2 指令表

2.1 指令类型

CPU 模块指令的主要类型包括顺控程序指令、基本指令、应用指令、数据链接指令、QCPU 用指令和冗余系统用指令。指令类型如下表 2.1 所示。

表 2.1 指令类型

指令类型		含义	指令类型
顺控程序指令	触点指令	操作启动，串级连接，并行连接	5
	连接指令	梯形图块连接，从操作结果产生的脉冲，存储 / 读操作结果	
	输出指令	位软元件输出，脉冲输出，输出倒置	
	移位指令	位软元件移位	
	主站控制指令	主站控制	
	终止指令	程序终止	
	其它指令	程序停止，其它指令，如和上述范畴内的操作都不同的指令	
基本指令	比较操作指令	比较操作如 ＝、 ＞、 ＜	6
	算术运算指令	BIN 或 BCD 的加法，减法，乘法或除法	
	BCD ↔ BIN 转换指令	将 BCD 转换成 BIN 和将 BIN 转换成 BCD	
	数据转移指令	传送指定的数据	
	程序分支指令	程序跳转	
	程序运行控制指令	允许或禁止中断程序	
	I/O 刷新	运行局部刷新	
	其它使用方便的指令	用于以下目的的指令：计数器增加 / 减小，教学定时器，特殊功能定时器，旋转台最短距离控制，等等	
应用指令	逻辑操作指令	逻辑操作，如逻辑和，逻辑乘等	7
	循环指令	指定数据的循环移位	
	移位指令	指定数据的移位	
	位处理指令	位置位和复位，位测试，位软元件的批复位	
	数据处理指令	16 位数据查找，数据处理如解码和编码	
	结构体创建指令	重复操作，子程序调用，梯形图中的变址修饰	
	表操作指令	读 / 写 FIFO 表	
	缓冲存储区访问指令	特殊功能模块的数据读 / 写	
	显示指令	打印 ASCII 码，LED 字符显示，等等	
	调试和故障诊断指令	检查，状态检查，采样跟踪，程序跟踪	
	字符串处理指令	CBIN/BCD 和 ASCII 之间的转化； BIN 和字符串之间的转化；浮点十进制数据和字符串之间的转化，字符串处理，等等	
	特殊功能指令	三角函数功能，角度和弧度之间的转化，指数操作，自然对数，方根运算	
	数据控制指令	最高和最低限控制，死区控制， 范围带控制	
	交换指令	文件寄存器块号码交换，文件寄存器和注释文件的指定	
	时钟指令	年，月，日，小时，分钟，秒和星期几的读 / 写；时间显示方式（时，分，秒）的变换和秒	
	外围设备指令	连接到外围设备的 I/O	
	程序控制用指令	用于转换程序执行条件的指令	
	其它指令	其它不符合以上范畴的指令，如看门狗定时器复位指令和定时时钟指令	

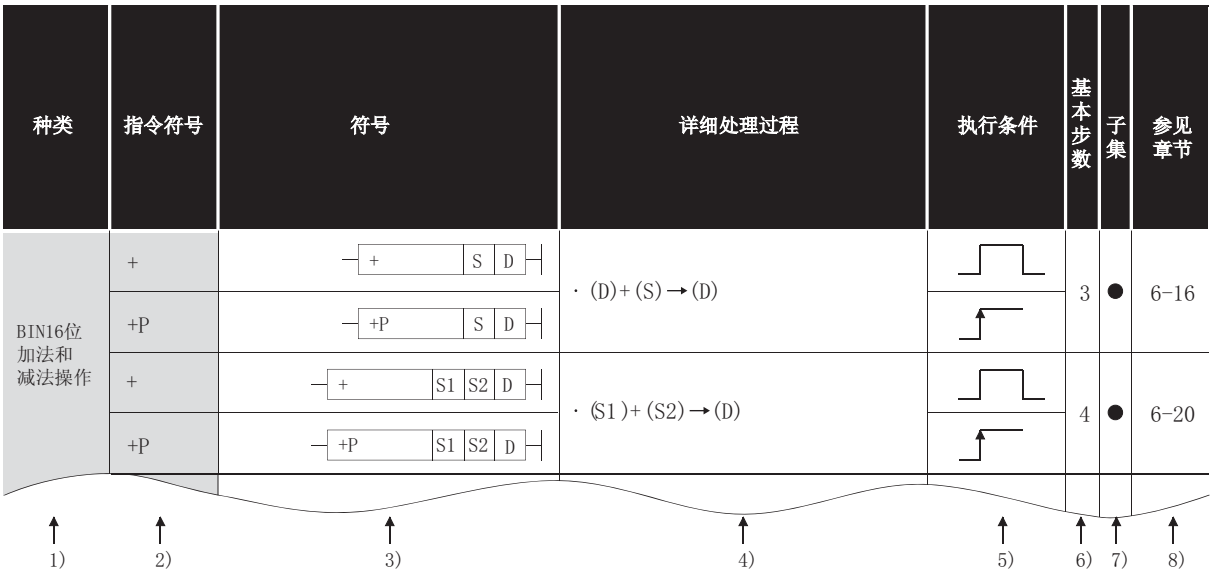
表 2.1 指令类型（续）

指令类型		含义	指令类型
数据链接指令	链接刷新指令	指定网络的刷新	8
	专门用于 QnA 链接的指令	对其它站数据的读 / 写；发往其它站的数据传输信号；发往其它站的处理请求	
	用于 A- 系列 - 兼容链接的指令	对指定站点字软元件的读 / 写，从远程 I/O 站特殊功能模块中读 / 写数据	
	路由信息读 / 写指令	读，写，和寄存器路由信息	
QCPU 指令	用于 QCPU 的指令	读取模块信息、跟踪设置 / 复位、二进制数据的读 / 写、标准 ROM 数据的读 / 写、从存储卡中安装 / 卸载 / 安装 + 卸载程序、文件寄存器中高速块传输、自站 CPU 共享存储器的写入、其他 CPU 共享存储器的读取，选择刷新指令和扩展时钟指令。	9
冗余系统指令	用于 Q4ARCPU 的指令	CPU 启动时操作模式的设置； CPU 切换时操作模式设置指令；数据跟踪；缓冲存储区批刷新	10
	用于冗余 CPU 的指令	系统切换	11

2.2 如何阅读指令表

从 2.3 节到 2.5 节的指令表是按照下列形式制作的：

表 2.2 如何阅读指令表



描述

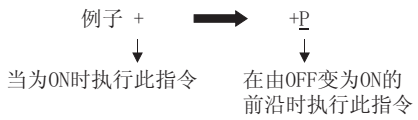
1)... 照指令的应用对它们进行分类。

2)... 表示增加到程序中的指令的指令符号。
指令代码是围绕 16 位指令建立的。下列符号用来标志只在由 OFF 变成 ON 的上升沿被执行 32 位指令，实数指令，和字符串指令：

• 32 位指令..... 在指令的第一行增加字母 “D”



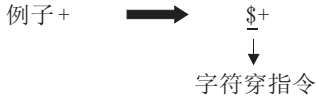
• 这些指令只在由 OFF 变成 ON 的上升沿被执行
..... 在指令的最后增加字母 “P”



• 实数指令 ..... 在指令的第一行增加字母 “E”



• 字符串指令 .... 在指令的第一行增加一个美元符号 “$”



3) . . . 在梯形图上显示符号图表

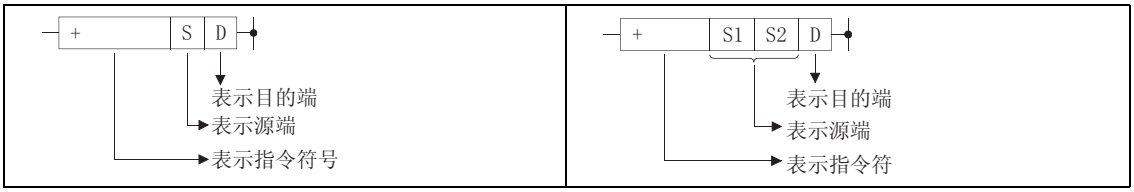


图 2.1 在梯形图上显示符号图表

目的端 . . . . . 表示数据在操作后被送往何处

源端 . . . . . 操作前存储数据的地方

4) . . . 表示由单个指令执行的操作的类型

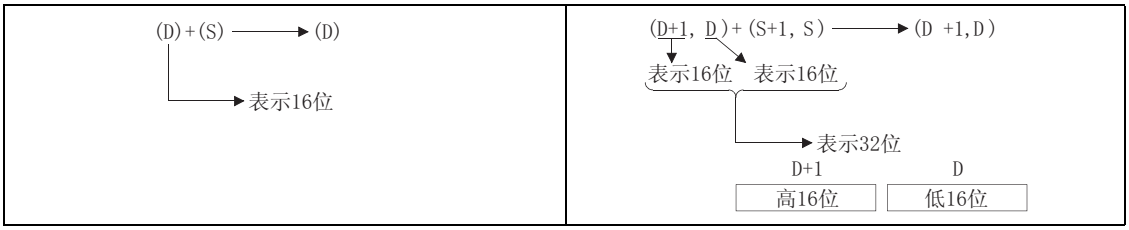


图 2.2 由单个指令执行的操作的类型

5) . . . 单个指令的执行情况细节如下所示：

符号	执行情况
无记录符号	指令在一般条件下执行，而不管指令前的 ON/OFF 状态。 如果前面的状态是 OFF，指令将执行 OFF 处理。
	条件为 ON 时执行；只有当前面的状态为 ON 时才执行指令。 如果前面的状态为 OFF，指令将不执行，并且不执行任何处理。
	一旦为 ON 时执行；只有前面的状态由 OFF 变为 ON 时在上升沿处执行指令。 此执行操作之后，即使条件保持为 ON，指令也不被执行并且不执行任何处理。
	当状态为 OFF 时执行；只有当前面的条件为 OFF 时指令才执行。 如果前面的状态为 ON，指令不被执行，并且不执行任何处理。
	一旦为 OFF 时执行；当前面的条件由 ON 变成 OFF 时在下降沿处执行指令。 此执行操作之后，即使条件保持为 OFF 指令也不被执行，并且不会执行任何处理。

6) . . . 表示单条指令的基本执行步数，

关于步数的描述参见 3.8 节。

7) . . . “ ● ”标志表示该指令可以有子处理过程。

关于子处理过程参见 3.5 节。

8) . . . 表示解释单条指令的页码。

2.3 顺序指令

2.3.1 触点指令

表 2.3 触点指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
触点	LD		• 启动逻辑操作 （启动触点逻辑操作）		*1	● *3	5-2
	LDI		• 启动 NOT 逻辑操作 （启动 b 触点逻辑操作）				
	AND		• 逻辑乘（a 触点串行连接）				
	ANI		• 逻辑乘（b 触点串行连接）				
	OR		• 逻辑或（a 触点并行连接）				
	ORI		• 逻辑或非（b 触点并行连接）				
	LDP		• 上升沿运算开始		*2	● *3	5-5
	LDF		• 下降沿运算开始				
	ANDP		• 上升沿脉冲串行连接				
	ANDF		• 下降沿脉冲串行连接				
	ORP		• 上升沿脉冲并行连接				
	ORF		• 下降沿脉冲并行连接				

*1: 步数会由于所使用软元件的不同而有变化。

软元件	步数
内部软元件，文件寄存器（R0 到 R32767）	1
直接访问输入（DX）	2
上述软元件外的其它软元件	3

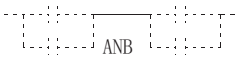
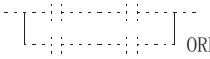
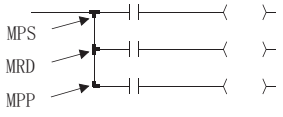
*2: 步数会由于所使用的软元件的不同和 CPU 模块类型的不同而有变化。

软元件	步数	
	QCPU	QnACPU
内部软元件，文件寄存器（R0 到 R32767）	1	2
直接访问输入（DX）	2	2
上述软元件外的其它软元件	3	3

*3: 子处理过程只在 QCPU 上有效。

2.3.2 连接指令

表 2.4 连接指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
连接	ANB		• 逻辑块之间的 AND(逻辑块之间的串行连接)		1	-	5-7
	ORB		• 逻辑块之间的 OR(逻辑块之间的串行连接)				
	MPS		• 运算结果的存储		1	-	5-9
	MRD		• 读 MPS 指令存储的运算结果				
	MPP		• 对 MPS 指令存储的运算结果的读和复位				
	INV		• 运算结果的取反		1	-	5-12
	MEP		• 运算结果上升沿脉冲化		1	-	5-13
	MEF		• 运算结果下降沿脉冲化				
	EGP		• 运算结果上升沿脉冲化 (存储在 Vn)		1	-	5-14
	EGF		• 运算结果下降沿脉冲化 (存储在 Vn)		*1		

*1: 步数会由于所使用的 CPU 模块类型的不同而有变化。

CPU 类型	基本步数
高性能型 QCPU 过程 CPU 冗余 CPU 通用型 QCPU QnACPU	1
基本型 QCPU	2

2.3.3 输出指令

表 2.5 输出指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
输出	OUT	—( )—	• 软元件输出		*1	–	5-16 5-18 5-22 5-24
	SET	—[SET D]—	• 设置软元件	*2	*1	–	5-26 5-31
	RST	—[RST D]—	• 复位软元件	*2	*1	–	5-28 5-31
	PLS	—[PLS D]—	• 在输入信号的上升沿处产生 1 个周期的程序脉冲		2	–	5-33
	PLF	—[PLF D]—	• 在输入信号的下降沿处产生 1 个周期的程序脉冲				
	FF	—[FF D]—	• 软元件输出的取反		2	–	5-36
	DELTA	—[DELTA D]—	• 直接输出的脉冲转化		2	–	5-38
	DELTAP	—[DELTAP D]—					

*1: 步数会由于所使用软元件的不同而有变化。
关于步数参见单个指令的描述说明页。

*2: 的执行条件只有在使用警报器 (F) 时才被应用。

2.3.4 移位指令

表 2.6 移位指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
移位	SFT	—[SFT D]—	• 软元件的 1 位移位		2	–	5-40
	SFTP	—[SFTP D]—					

2.3.5 主控制指令

表 2.7 主控制指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
主站控制	MC	MC n D	• 启动主控制		2	–	5-43
	MCR	MCR n	• 复位主控制		1		

2.3.6 终止指令

表 2.8 终止指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
程序结束	FEND	FEND	• 主程序的终止		1	–	5-47
	END	END	• 顺序程序的终止				5-49

2.3.7 其它指令

表 2.9 其它指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
停止	STOP	STOP	• 在输入条件得到满足后终止顺序程序。 • 将 RUN/STOP 键切换回到 RUN 位置后顺序程序重新被执行		1	–	5-51
忽略	NOP	---	• 忽略（用于程序删除或空白区）		1	–	5-53
	NOPLF	NOPLF	• 忽略（在打印输出过程中改变页码号）				
	PAGE	PAGE n	• 忽略（并行程序从页码 n 的步骤 0 开始被控制）				

2.4 基本指令

2.4.1 比较操作指令

表 2.10 比较操作指令

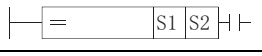
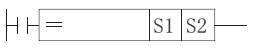
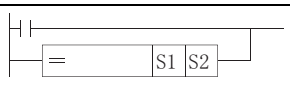
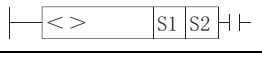
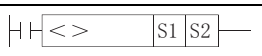
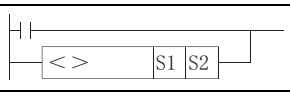
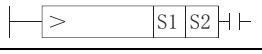
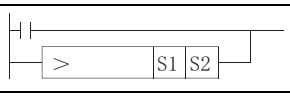
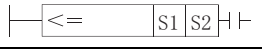
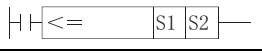
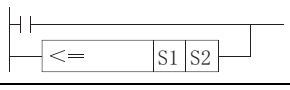
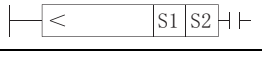
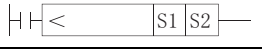
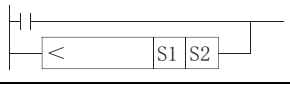
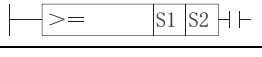
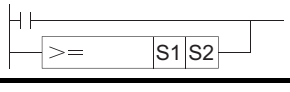
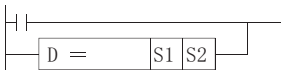
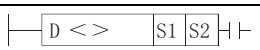
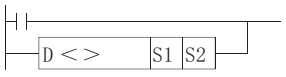
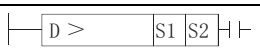
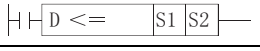
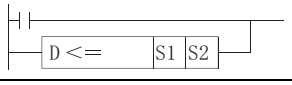
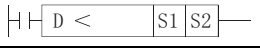
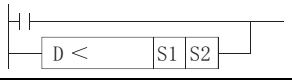
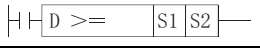
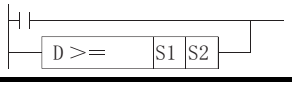
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
16 位数据比较	LD=		• 当 (S1) ＝ (S2) 时处于导通状态 • 当 (S1) ≠ (S2) 时处于不导通状态		3	●	6-2
	AND=						
	OR=						
	LD<>		• 当 (S1) ≠ (S2) 时处于导通状态 • 当 (S1) ＝ (S2) 时处于导通状态		3	●	
	AND<>						
	OR<>						
	LD>		• 当 (S1) ＞ (S2) 时处于导通状态 • 当 (S1) ≦ (S2) 时处于不导通状态		3	●	
	AND>						
	OR>						
	LD<=		• 当 (S1) ≦ (S2) 时处于导通状态当 • 当 (S1) ＞ (S2) 时处于不导通状态		3	●	
	AND<=						
	OR<=						
	LD<		• 当 (S1) ＜ (S2) 时处于导通状态 • 当 (S1) ≧ (S2) 时处于不导通状态		3	●	
	AND<						
	OR<						
	LD>=		• 当 (S1) ≧ (S2) 时处于导通状态 • 当 (S1) ＜ (S2) 时处于不导通状态		3	●	
	AND>=						
	OR>=						

表 2.10 比较操作指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
32 位数据比较	LDD=		• 当 (S1+1、S1) ＝ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ≠ (S2+1、S2) 时处于不导通状态		*1	●	6-4
	ANDD=						
	ORD=						
	LDD<>		• 当 (S1+1、S1) ≠ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＝ (S2+1、S2) 时处于不导通状态		*1	●	
	ANDD<>						
	ORD<>						
	LDD>		• 当 (S1+1、S1) ＞ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ≦ (S2+1、S2) 时处于不导通状态		*1	●	
	ANDD>						
	ORD>						
	LDD<=		• 当 (S1+1、S1) ≦ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＞ (S2+1、S2) 时处于不导通状态		*1	●	
	ANDD<=						
	ORD<=						
	LDD<		• 当 (S1+1、S1) ＜ (S2+1、S2) 时处于导通状态当 (S1+1、S1) ≧ (S2+1、S2) 时处于不导通状态		*1	●	
	ANDD<						
	ORD<						
	LDD>=		• 当 (S1+1、S1) ≧ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＜ (S2+1、S2) 时处于不导通状态		*1	●	
	ANDD>=						
	ORD>=						

*1: 执行步数可能随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	5 注 1)
	使用上述以外的软元件时	3 注 2)
基本型 QCPU 通用型 QCPU QnACPU	可以使用所有软元件	3 注 2)

注 1) 对于高性能型 QCPU、过程 CPU 或冗余 CPU，执行步数更多但是执行速度也快。
注 2) 执行步数会基于在 3.8 节中描述的条件而增加。

表 2.10 比较操作指令（续表）

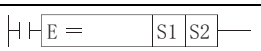
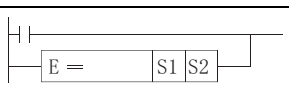
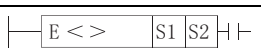
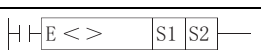
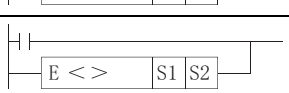
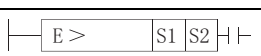
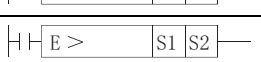
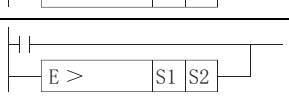
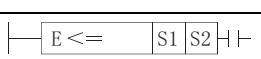
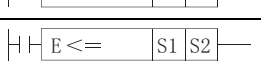
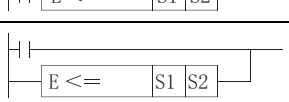
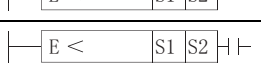
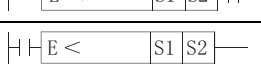
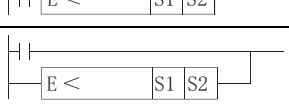
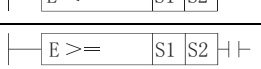
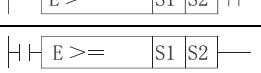
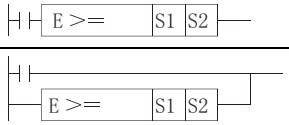
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
浮点数据比较（单精度）	LDE=		• 当 (S1+1、S1) ＝ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ≠ (S2+1、S2) 时处于不导通状态		3	－	6-6
	ANDE=						
	ORE=						
	LDE<>		• 当 (S1+1、S1) ≠ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＝ (S2+1、S2) 时处于不导通状态		3	－	
	ANDE<>						
	ORE<>						
	LDE>		• 当 (S1+1、S1) ＞ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ≦ (S2+1、S2) 时处于不导通状态		3	－	
	ANDE>						
	ORE>						
	LDE<=		• 当 (S1+1、S1) ≦ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＞ (S2+1、S2) 时处于不导通状态		3	－	
	ANDE<=						
	ORE<=						
	LDE<		• 当 (S1+1、S1) ＜ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ≧ (S2+1、S2) 时处于不导通状态		3	－	
	ANDE<						
	ORE<						
	LDE>=		• 当 (S1+1、S1) ≧ (S2+1、S2) 时处于导通状态 • 当 (S1+1、S1) ＜ (S2+1、S2) 时处于不导通状态		3	－	
	ANDE>=						
	ORE>=						

表 2.10 比较操作指令（续表）

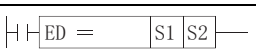
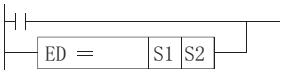
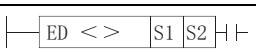
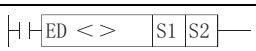
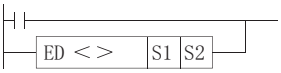
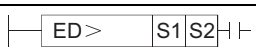
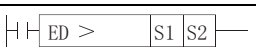
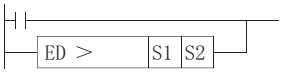
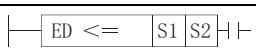
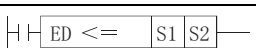
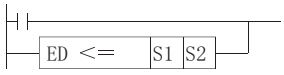
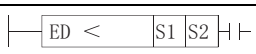
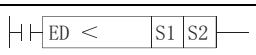
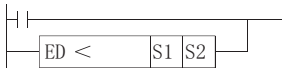
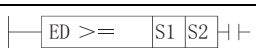
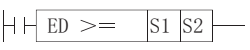
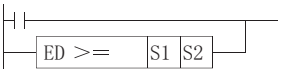
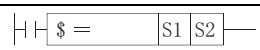
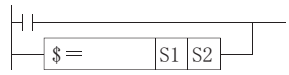
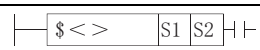
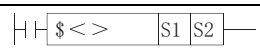
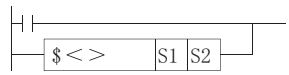
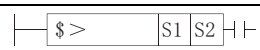
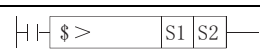
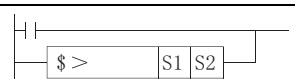
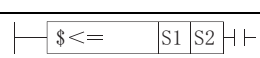
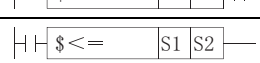
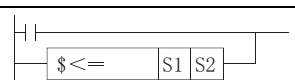
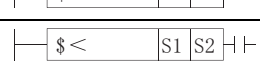
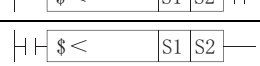
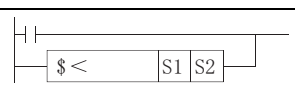
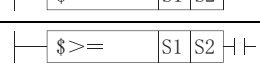
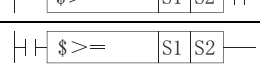
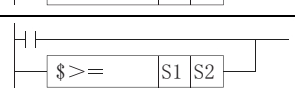
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
浮点数据比较（双精度）	LDED=		• 当（S1+3、S1+2、S1+1、S1）＝（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）≠（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	6-8
	ANDED=						
	ORED=						
	LDED<>		• 当（S1+3、S1+2、S1+1、S1）≠（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）＝（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	
	ANDED<>						
	ORED<>						
	LDED>		• 当（S1+3、S1+2、S1+1、S1）＞（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）≦（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	
	ANDED>						
	ORED>						
	LDED<=		• 当（S1+3、S1+2、S1+1、S1）≦（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）＞（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	
	ANDED<=						
	ORED<=						
	LDED<		• 当（S1+3、S1+2、S1+1、S1）＜（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）≧（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	
	ANDED<						
	ORED<						
	LDED>=		• 当（S1+3、S1+2、S1+1、S1）≧（S2+3、S2+2、S2+1、S2）时处于导通状态 • 当（S1+3、S1+2、S1+1、S1）＜（S2+3、S2+2、S2+1、S2）时处于不导通状态		3	-	
	ANDED>=						
	ORED>=						

表 2.10 比较操作指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
字符串数据比较	LD$=		• 比较字符串 S1 和字符串 S2，一次一个字符。 *2 • 当（字符串 S1）＝（字符串 S2）时处于导通状态 • 当（字符串 S1）≠（字符串 S2）时处于不导通状态		3	-	6-8
	AND$=						
	OR$=						
	LD$<>		• 比较字符串 S1 和字符串 S2，一次一个字符。 *2 • 当（字符串 S1）≠（字符串 S2）时处于导通状态 • 当（字符串 S1）＝（字符串 S2）时处于不导通状态		3	-	
	AND$<>						
	OR$<>						
	LD$>		• 比较字符串 S1 和字符串 S2，一次一个字符。 *2 • 当（字符串 S1）＞（字符串 S2）时处于导通状态 • 当（字符串 S1）≦（字符串 S2）时处于不导通状态		3	-	
	AND$>						
	OR$>						
	LD$<=		• 比较字符串 S1 和字符串 S2，一次一个字符。*2 • 当（字符串 S1）≦（字符串 S2）时处于导通状态 • 当（字符串 S1）＞（字符串 S2）时处于不导通状态		3	-	
	AND$<=						
	OR$<=						
	LD$<		• 比较字符串 S1 和字符串 S2，一次一个字符。 *2 • 当（字符串 S1）＜（字符串 S2）时处于导通状态 • 当（字符串 S1）≧（字符串 S2）时处于不导通状态		3	-	
	AND$<						
	OR$<						
	LD$>=		• 比较字符串 S1 和字符串 S2，一次一个字符。*2 • 当（字符串 S1）≧（字符串 S2）时处于导通状态 • 当（字符串 S1）＜（字符串 S2）时处于不导通状态		3	-	
	AND$>=						
	OR$>=						

*2：可以执行字符串比较的条件如下所示：

- 匹配　　：串中的所有字节都匹配。
- 较大的字符串　：如果字符串不同，取字符代码数最大的字符串。如果字符串的长度不同，取最长的字符串。
- 较小的字符串　：如果字符串不同，取字符代码数最小的字符串。如果字符串的长度不同，取最短的字符串。

表 2.10 比较操作指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BIN 块数据比较	BKCMP=	BKCMP＝ S1 S2 D n	• 在 1 字单元内从 S1 开始的 n 点数据和从 S2 开始的 n 点数据进行比较，并且将比较结果存储到从 (D) 指定的位软元件的 n 个点。		5	–	6-15
	BKCMP<>	BKCMP＜＞ S1 S2 D n					
	BKCMP>	BKCMP ＞ S1 S2 D n					
	BKCMP<=	BKCMP＜＝ S1 S2 D n					
	BKCMP<	BKCMP ＜ S1 S2 D n					
	BKCMP>=	BKCMP ＞＝ S1 S2 D n					
	BKCMP=P	BKCMP＝P S1 S2 D n					
	BKCMP<>P	BKCMP＜＞P S1 S2 D n					
	BKCMP>P	BKCMP ＞P S1 S2 D n					
	BKCMP<=P	BKCMP＜＝P S1 S2 D n					
	BKCMP<P	BKCMP ＜ P S1 S2 D n					
	BKCMP>=P	BKCMP ＞＝P S1 S2 D n					

2.4.2 算术运算指令

表 2.11 算术运算指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BIN16 位加和减运算	+	+ S D	• (D)+(S)→(D)		3	●	6-19
	+P	+P S D					
	+	+ S1 S2 D	• (S1)+(S2)→(D)		4	●	6-21
	+P	+P S1 S2 D					
	-	— S D	• (D)—(S)→(D)		3	●	6-19
	-P	—P S D					
	-	— S1 S2 D	• (S1)—(S2)→(D)		4	●	6-21
	-P	—P S1 S2 D					
BIN32 位加和减运算	D+	D+ S D	• (D+1、D)+(S+1、S)→(D+1、D)		*1	●	6-23
	D+P	D+P S D					
	D+	D+ S1 S2 D	• (S1+1、S1)+(S2+1、S2)→(D+1、D)		*2	●	6-25
	D+P	D+P S1 S2 D					
	D-	D — S D	• (D+1、D)—(S+1、S)→(D+1、D)		*1	●	6-23
	D-P	D —P S D					
	D-	D — S1 S2 D	• (S1+1、S1)—(S2+1、S2)→(D+1、D)		*2	●	6-25
	D-P	D —P S1 S2 D					
BIN16 位乘法和除法运算	*	* S1 S2 D	• (S1)×(S2)→(D+1、D)		*3	●	6-27
	*P	*P S1 S2 D					
	/	/ S1 S2 D	• (S1) / (S2) →商 (D)、余数 (D+1)		*4	●	
	/P	/P S1 S2 D					
BIN32 位乘法和除法运算	D*	D* S1 S2 D	• (S1+1、S1)×(S2+1、S2)→(D+3、D+2、D+1、D)		*4	●	6-29
	D*P	D*P S1 S2 D					
	D/	D/ S1 S2 D	• (S1+1、S1) / (S2+1、S2) →商 (D+1、D)、余数 (D+3、D+2)		*4	●	
	D/P	D/P S1 S2 D					

*1: 执行步数可能随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	5 注 1)
	使用上述以外的软元件时	3 注 2)
基本型 QCPU 通用型 QCPU QnACPU	可以使用所有软元件	3 注 2)

注 1: 对于高性能型 QCPU、过程 CPU 或冗余 CPU，执行步数更多但是执行速度也快。
注 2: 执行步数会基于在 3.8 节中描述的条件而增加。

*2: 执行步数可能随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	6 注 1)
	使用上述以外的软元件时	4 注 2)
基本型 QCPU QnACPU	可以使用所有软元件	4 注 2)
通用型 QCPU		3 注 2)

注 1: 对于高性能型 QCPU、过程 CPU 或冗余 CPU，执行步数更多但是执行速度也快。
注 2: 执行步数会基于在 3.8 节中描述的条件而增加。

*3: 执行步数可能随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
QCPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	3
	使用上述以外的软元件时	4 注 1)
QnACPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	4
	使用上述以外的软元件时	4 注 1)

注 1: 执行步数会基于在 3.8 节中描述的条件而增加。

*4: 只有在使用通用型 QCPU 时，基本步数才为 3。

表 2.11 算术运算指令（续表））

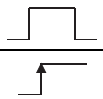
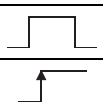
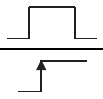
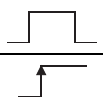
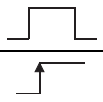
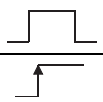
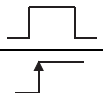
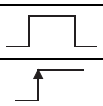
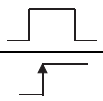
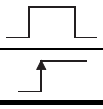
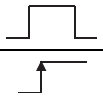
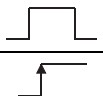
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BCD4 位加和减运算	B+	B+ S D	• (D)+(S)→(D)		3	●	6-31
	B+P	B+P S D					
	B+	B+ S1 S2 D	• (S1)+(S2)→(D)		4	-	6-33
	B+P	B+P S1 S2 D					
	B-	B－ S D	• (D)－(S)→(D)		3	●	6-31
	B-P	B－P S D					
	B-	B－ S1 S2 D	• (S1)－(S2)→(D)		4	-	6-33
	B-P	B－P S1 S2 D					
BCD8 位加和减运算	DB+	DB+ S D	• (D+1、 D)+(S+1、 S)→(D+1、 D)		3	-	6-35
	DB+P	DB+P S D					
	DB+	DB+ S1 S2 D	• (S1+1、 S1)+(S2+1、 S2)→(D+1、 D)		4	-	6-37
	DB+P	DB+P S1 S2 D					
	DB-	DB － S D	• (D+1、 D)－(S+1、 S)→(D+1、 D)		3	-	6-35
	DB-P	DB －P S D					
	DB-	DB － S1 S2 D	• (S1+1、 S1)－(S2+1、 S2)→(D+1、 D)		4	-	6-37
	DB-P	DB －P S1 S2 D					
BCD4 位乘和除运算	B*	B＊ S1 S2 D	• (S1) × (S2)→(D+1、D)		4	●	6-39
	B*P	B＊P S1 S2 D					
	B/	B/ S1 S2 D	• (S1) / (S2) →商 (D)、 余数 (D+1)		4	●	
	B/P	B/P S1 S2 D					
BCD4 位乘和除运算	DB*	DB＊ S1 S2 D	• (S1+1、S1) × (S2+1、S2) →(D+3、D+2、D+1、D)		4	-	6-41
	DB*P	DB＊P S1 S2 D					
	DB/	DB/ S1 S2 D	• (S1+1、 S1) / (S2+1、 S2) →商 (D+1、 D)、 余数 (D+3、 D+2)		4	●	
	DB/P	DB/P S1 S2 D					

表 2.11 算术运算指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
浮点数据加和减运算（单精度）	E+	E+ S D	• (D+1、 D)+(S+1、 S)→(D+1、 D)		3	● *6	6-43
	E+P	E+P S D					
	E+	E+ S1 S2 D	• (S1+1、 S1)+(S2+1、 S2)→(D+1、 D)		4 *5	● *6	6-45
	E+P	E+P S1 S2 D					
	E-	E－ S D	• (D+1、 D)－(S+1、 S)→(D+1、 D)		3	● *6	6-43
	E-P	E－P S D					
	E-	E－ S1 S2 D	• (S1+1、 S1)－(S2+1、 S2)→(D+1、 D)		4 *5	● *6	6-45
	E-P	E－P S1 S2 D					
浮点数据加和减运算（双精度）	ED+	ED+ S D	• (D+3、 D+2、 D+1、 D)+(S+3、 S+2、 S+1、 S) →(D+3、 D+2、 D+1、 D)		3	●	6-47
	ED+P	ED+P S D					
	ED+	ED+ S1 S2 D	• (S1+3、 S1+2、 S1+1、 S1)+ (S2+3、 S2+2、 S2+1、 S2)→ (D+3、 D+2、 D+1、 D)		4	●	6-49
	ED+P	ED+P S1 S2 D					
	ED-	ED － S D	• (D+3、 D+2、 D+1、 D)－(S+3、 S+2、 S+1、 S) →(D+3、 D+2、 D+1、 D)		3	●	6-47
	ED-P	ED－P S D					
	ED-	ED － S1 S2 D	• (S1+3、 S1+2、 S1+1、 S1)－ (S2+3、 S2+2、 S2+1、 S2)→ (D+3、 D+2、 D+1、 D)		4	●	6-49
	ED-P	ED －P S1 S2 D					
浮点数据乘和除运算（单精度）	E*	E* S1 S2 D	• (S1+1、S1) × (S2+1、S2)→(D+1、D)		3	● *6	6-51
	E*P	E*P S1 S2 D					
	E/	E/ S1 S2 D	• (S1+1、 S1) / (S2+1、 S2) →商 (D+1、 D)		4	● *6	
	E/P	E/P S1 S2 D					
浮点数据乘和除运算（双精度）	ED*	ED＊ S1 S2 D	• (S1+3、S1+2、S1+1、S1) × (S2+3、S2+2、S2+1、S2)→ (D+3、D+2、D+1、D)		4	● *6	6-53
	ED*P	ED＊P S1 S2 D					
	ED/	ED/ S1 S2 D	• (S1+3、 S1+2、 S1+1、 S1) / (S2+3、 S2+2、 S2+1、 S2)→ 商 (D+3、 D+2、 D+1、 D)		4	● *6	
	ED/P	ED/P S1 S2 D					

*5:　只有在使用通用型 QCPU 时，基本步数才为 3。
*6:　只有在使用通用型 QCPU 时，子集才会有效。

表 2.11 算术运算指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BIN 块加和减运算	BK+	BK+ S1 S2 D n	• 将 (S1) 开始的 n 点数据与 (S2) 开始的 n 点数据批量相加。		5	–	6-56
	BK+P	BK+P S1 S2 D n					
	BK-	BK － S1 S2 D n	• 将 (S1) 开始的 n 点数据与 (S2) 开始的 n 点数据批量相减。		5	–	
	BK-P	BK －P S1 S2 D n					
字符串数据合并	$+	$+ S D	• 将 (S) 指定的字符串与 (D) 指定的字符串连接，并从 (D) 处向前存储操作结果。		3	–	6-59
	$+P	$+P S D					
	$+	$+ S1 S2 D	• 将 (S2) 指定的字符串与 (S1) 指定的字符串连接，并从 (D) 处向前存储操作结果。		4	–	6-61
	$+P	$+P S1 S2 D					
BIN 数据增加	INC	INC D	• (D)+1→(D)		2	●	6-63
	INCP	INCP D					
	DINC	DINC D	• (D+1、 D)+1→(D+1、 D)		*7	●	6-65
	DINCP	DINCP D					
	DEC	DEC D	• (D) －1→(D)		2	●	6-63
	DECP	DECP D					
	DDEC	DDEC D	• (D+1、 D)－1→(D+1、 D)		*7	●	6-65
	DDECP	DDECP D					

*7：执行步数可能随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	3 注 1)
	使用上述以外的软元件时	2 注 2)
基本型 QCPU 通用型 QCPU QnACPU	可以使用所有软元件	2 注 2)

注 1) 对于高性能型 QCPU、过程 CPU 或冗余 CPU，执行步数更多但是执行速度也快。
注 2) 执行步数会基于在 3.8 节中描述的条件而增加。

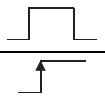
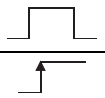
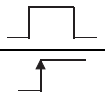
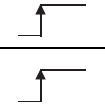
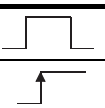
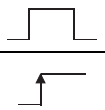
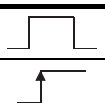
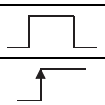
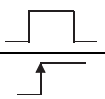
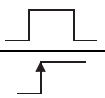
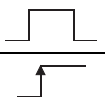
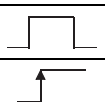
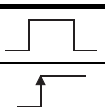
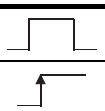
2.4.3 数据转换指令

表 2.12 数据转换指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BCD 转换	BCD	BCD S D	BCD转换 (S) → (D) BIN(0到9999)		3 *1	●	6-67
	BCDP	BCDP S D					
	DBCD	DBCD S D	BCD转换 (S+1, S) → (D+1, D) BIN(0到99999999)		3 *1	●	
	DBCDP	DBCDP S D					
BIN 转换	BIN	BIN S D	BIN转换 (S) → (D) BCD(0到9999)		3 *1	●	6-69
	BINP	BINP S D					
	DBIN	DBIN S D	BIN转换 (S+1, S) → (D+1, D) BCD(0到99999999)		3 *1	●	
	DBINP	DBINP S D					
从 BIN 转换成浮点数（单精度）	FLT	FLT S D	转换成实数 (S) → (D+1, D) BIN(－32768到32767)		3 *1	● *2	6-72
	FLTP	FLTP S D					
	DFLT	DFLT S D	转换成实数 (S+1, S) → (D+1, D) BIN(－2147483648到2147483647)		3 *1	● *2	
	DFLTP	DFLTP S D					
从 BIN 转换成浮点数（双精度）	FLTD	FLTD S D	转换成实数 (S) → (D+3, D+2, D+1, D) BIN(－32768到32767)		4	● *2	6-75
	FLTDP	FLTDP S D					
	DFLTD	DFLTD S D	转换成实数 (S+1, S) → (D+3, D+2, D+1, D) BIN(－2147483648到2147483647)		4	● *2	
	DFLTDP	DFLTDP S D					
从浮点数转换成 BIN（双精度）	INT	INT S D	转换成BIN (S+1, S) → (D) 实数(-32768到32767)		3 *1	● *2	6-77
	INTP	INTP S D					
	DINT	DINT S D	转换成BIN (S+1, S) → (D+1, D) 实数(-2147483648到2147483647)		3 *1	● *2	
	DINTP	DINTP S D					
从浮点数转换成 BIN（双精度）	INTD	INTD S D	转换成BIN N (S+3, S+2, S+1, S) → (D+1, D) 实数(－32768到32767)		3	● *2	6-80
	INTDP	INTDP S D					
	DINTD	DINTD S D	转换成BIN N (S+3, S+2, S+1, S) → (D+1, D) 实数(－2147483648到2147483647)		3	● *2	
	DINTDP	DINTDP S D					

*1: 只有在使用通用型 QCPU 时，基本步数才为 2。
*2: 只有在使用通用型 QCPU 时，子集才会有效。

表 2.12 数据转换指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BIN16 位和 32 位之间的转换	DBL	DBL S D	· (S) 转换 (D+1, D) BIN(-32768到32767)		3	–	6-82
	DBLP	DBLP S D					
	WORD	WORD S D	· (S+1, S) 转换 (D) BIN(-32768到32767)		3	–	6-83
	WORDP	WORDP S D					
BIN 转换成格雷码	GRY	GRY S D	· (S) 转换成格雷码 (D) BIN(-32768到32767)		3	–	6-84
	GRYP	GRYP S D					
	DGRY	DGRY S D	· (S+1, S) 转换成格雷码 (D+1, D) BIN(-2147483648到2147483647)		3	–	
	DGRYP	DGRYP S D					
格雷码转换成 BIN	GBIN	GBIN S D	· (S) 转换成BIN数据 (D) 格雷码(-32768到32767)		3	–	6-86
	GBINP	GBINP S D					
	DGBIN	DGBIN S D	· (S+1, S) 转换成BIN数据 (D+1, D) 格雷码(-2147483648到2147483647)		3	–	
	DGBINP	DGBINP S D					
补足到 2	NEG	NEG D	· (D) (D) BIN数据		2	–	6-88
	NEGP	NEGP D					
	DNEG	DNEG D	· (D+1, D) (D+1, D) BIN数据		2	–	
	DNEGP	DNEGP D					
	ENEG	ENEG D	· (D+1, D) (D+1, D) 实数数据		2	–	6-90
	ENEGP	ENEGP D					
	EDNEG	EDNEG D	· (D+3, D+2, D+1, D) (D+3, D+2, D+1, D) 实数数据		3	–	6-91
	EDNEGP	EDNEGP D					
块转换	BKBCD	BKBCD S D n	• 将从(S)开始的n个数据点的BIN数据批量转换成 BCD 数据，并从 (D) 处向前存储操结果。		4	–	6-92
	BKBCDP	BKBCDP S D n					
	BKBIN	BKBIN S D n	• 将从(S)开始的n个数据点的BCD数据批量转换成 BIN 数据，并从 (D) 处向前存储操作结果。		4	–	6-94
	BKBINP	BKBINP S D n					
浮点数单精度到双精度	ECON	ECON S D	· (S+1, S) 转换成双精度 (D+3, D+2, D+1, D) 32位浮点型实数		3	–	6-96
	ECONP	ECONP S D					
浮点数双精度到单精度	EDCON	EDCON S D	· (S+3, S+2, S+1, S) 转换成单精度 (D+1, D) 64位浮点型实数		3	–	6-98
	EDCONP	EDCONP S D					

2.4.4 数据传送指令

表 2.13 数据传送指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
16 位数据传送	MOV	MOV S D	· (S) → (D)		*4	●	6-100
	MOVP	MOVP S D			*1		
32 位数据传送	DMOV	DMOV S D	· (S+1,S) → (D+1,D)		*2	●	
	DMOVP	DMOVP S D					
浮点十进制数据传送	EMOV	EMOV S D	· (S+1, S) → (D+1, D) 实型数据		*2	● *3	6-102
	EMOVP	EMOVP S D					
字符串数据传送	$MOV	$MOV S D	• 将由 (S) 指定的字符串传送到由 (D) 指定的软元件，向前存储。		3	-	6-106
	$MOVP	$MOVP S D					
16 位负数据传送	CML	CML S D	· $\overline{(S)}$ → (D)		*1	●	6-108
	CMLP	CMLP S D					
32 位负数据传送	DCML	DCML S D	· $\overline{(S+1,S)}$ → (D+1,D)		*2	●	
	DCMLP	DCMLP S D					
块传送	BMOV	BMOV S D n	(S) → (D), n		4	●	6-111
	BMOVP	BMOVP S D n					
同一数据块的多路传送	FMOV	FMOV S D n	(S) → (D), n		4	●	6-114
	FMOVP	FMOVP S D n					
16 位数据交换	XCH	XCH D1 D2	· (D1) ↔ (D2)		3	●	6-116
	XCHP	XCHP D1 D2					
32 位数据交换	DXCH	DXCH D1 D2	· (D1+1,D1) ↔ (D2+1,D2)		3	●	
	DXCHP	DXCHP D1 D2					
块数据交换	BXCH	BXCH S D n	(S) ↔ (D), n		4	-	6-118
	BXCHP	BXCHP S D n					

表 2.13 数据传送指令（续）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
高字节和低字节间的交换	SWAP	SWAP D	(S) b15 到 b8 b7 到 b0 8位 8位 (D) b15 到 b8 b7 到 b0 8位 8位		3	-	6-120
	SWAPP	SWAPP D					

*1: 执行步数随使用的软元件 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
QCPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K4，无变址修饰。 • 常数： 无限制	2
	使用除上述以外的软元件时	3 注释 1)
QnACPU	可以使用所有软元件	3 注释 1)

注释 1: 执行步数会根据在 3.8 节中描述的条件而增加。

*2: 执行步数随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	3
	使用除上述以外的软元件时	3 注释 1)
基本型 QCPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制 （当使用上面的软元件 + 常数时，步数为 3。）	2
	使用除上述以外的软元件时	3 注释 1)
QnACPU	可以使用所有软元件	3 注释 1)
通用型 QCPU		2 注释 1)

注释 1: 执行步数会根据在 3.8 节中描述的条件而增加。

*3: 只有在使用 QCPU 时，子集才会有效。

*4: 执行步数随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
QCPU QnACPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K4，无变址修饰。 • 常数： 无限制	2
	使用除上述以外的软元件时	3 注释 1)

注释 1: 执行步数会根据在 3.8 节中描述的条件而增加。

2.4.5 程序分支指令

表 2.14 程序分支指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
跳转	CJ	CJ Pn	• 当输入条件满足时跳转到 Pn。		2	●	6-121
	SCJ	SCJ Pn	• 在输入条件得到满足以后的下一个扫描周期跳转到 Pn。		2	●	
	JMP	JMP Pn	• 无条件地跳转到 Pn。		2	●	
	GOEND	GOEND	• 当输入条件得到满足时，跳转到 END 指令。		1	-	6-124

2.4.6 程序执行控制指令

表 2.15 程序执行控制指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
使中断无效	DI	DI	• 禁止中断程序的运行。		1	-	6-125
使中断有效	EI	EI	• 允许中断程序的执行。		1	-	
中断无效/有效设置	IMASK	IMASK S	• 对每个中断程序禁止或允许中断。		2	-	
返回	IRET	IRET	• 在中断程序之后返回到顺序程序。		1	-	6-134

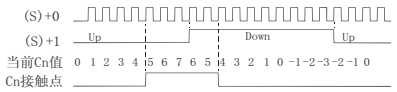
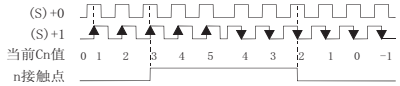
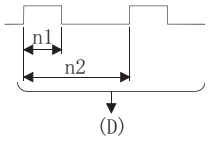
2.4.7 I/O 刷新指令

表 2.16 I/O 刷新指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
I/O 刷新	RFS	RFS S n	• 在扫描过程中刷新相关的 I/O 区域。		3	-	6-136
	RFSP	RFSP S n					

2.4.8 其它使用方便的指令

表 2.17 其它使用方便的指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
递增/递减计数器	UDCNT1	UDCNT1 S D n			4	-	6-138
	UDCNT2	UDCNT2 S D n			4	-	6-141
教学定时器	TTMR	TTMR D n	• (TTMR为ON的时间)×n ⟶ (D) n=0:1, n=1:10n, n=2:100		3	-	6-144
特殊定时器	STMR	STMR S n D	• 对从由 (D) 指定的位软元件开始的4点的操作如下所示，这取决于用于 STMR 指令的输入条件的 ON/OFF 状态： (D)+0：关闭延迟定时器输出 (D)+1：关闭定时器输出后的一拍 (D)+2：开启定时器输出后的一拍 (D)+3：启延迟定时器输出		3	-	6-146
就近控制	ROTC	ROTC S n1 n2 D	• 从停止位置到由 (S+1) 指定的位置，按照最短路径以 n1 个分区循环移动循环表。		5	-	6-149
斜坡信号	RAMP	RAMP n1 n2 D1 n3 D2	• 从 n1 到 n2，对由 D1 指定的软元件数据在 n3 个扫描周期内改变这些软元件数据。		6	-	6-152
脉冲密度	SPD	SPD S n D	• 在由 n 指定的时间周期内计数从由 (S) 指定的软元件发出的脉冲输入，并且将计数结果存储到由 (D) 指定的软元件中。		4	-	6-155
输出	PLSY	PLSY n1 n2 D	• (n1) Hz ⟶ (D) 输出n2次		4	-	6-157
脉冲宽调制	PWM	PWM n1 n2 D			4	-	6-159
矩阵输入	MTR	MTR S D1 D2 n	• 从由 (S) 指定的软元件开始，依次读取16点× n 列的数据，并将其存储到由 (D2) 指定的软元件以后。		5	-	6-161

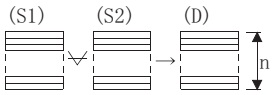
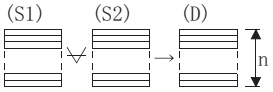
2.5 应用指令

2.5.1 逻辑操作指令

表 2.18 逻辑操作指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
逻辑积	WAND	WAND S D	· (D) ∧ (S) → (D)		3	●	7-3
	WANDP	WANDP S D					
	WAND	WAND S1 S2 D	· (S1) ∧ (S2) → (D)		4 *1	● *4	7-6
	WANDP	WANDP S1 S2 D					
	DAND	DAND S D	· (D+1, D) ∧ (S+1, S) → (D+1, D)		*2	●	7-3
	DANDP	DANDP S D					
	DAND	DAND S1 S2 D	· (S1+1, S1) ∧ (S2+1, S2) → (D+1, D)		*3	● *4	7-6
	DANDP	DANDP S1 S2 D					
	BKAND	BKAND S1 S2 D n	(S1) ∧ (S2) → (D) n		5	–	7-9
	BKANDP	BKANDP S1 S2 D n					
逻辑和	WOR	WOR S D	· (D) ∨ (S) → (D)		3	●	7-11
	WORP	WORP S D					
	WOR	WOR S1 S2 D	· (S1) ∨ (S2) → (D)		4 *1	● *4	7-14
	WORP	WORP S1 S2 D					
	DOR	DOR S D	· (D+1, D) ∨ (S+1, S) → (D+1, D)		*2	●	7-11
	DORP	DORP S D					
	DOR	DOR S1 S2 D	· (S1+1, S1) ∨ (S2+1, S2) → (D+1, D)		*3	● *4	7-14
	DORP	DORP S1 S2 D					
	BKOR	BKOR S1 S2 D n	(S1) ∨ (S2) → (D) n		5	–	7-17
	BKORP	BKORP S1 S2 D n					
专有 OR	WXOR	WXOR S D	· (D) ⊻ (S) → (D)		3	●	7-19
	WXORP	WXORP S D					
	WXOR	WXOR S1 S2 D	· (S1) ⊻ (S2) → (D)		4 *1	● *4	7-22
	WXORP	WXORP S1 S2 D					
	DXOR	DXOR S D	· (D+1, D) ⊻ (S+1, S) → (D+1, D)		*2	●	7-19
	DXORP	DXORP S D					

表 2.18 辑操作指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
专有 OR	DXOR	DXOR S1 S2 D	· (S1+1, S1) ⩛ (S2+1, S2) → (D+1, D)		*3	● *4	7-22
	DXORP	DXORP S1 S2 D					
	BKXOR	BKXOR S1 S2 D n			5	–	7-25
	BKXORP	BKXORP S1 S2 D n					
非专有逻辑求和	WXNR	WXNR S D	· $\overline{(D) \veebar (S)}$ → (D)		3	●	7-27
	WXNRP	WXNRP S D					
	WXNR	WXNR S1 S2 D	· $\overline{(S1) \veebar (S2)}$ → (D)		4 *1	● *4	7-30
	WXNRP	WXNRP S1 S2 D					
	DXNR	DXNR S D	· $\overline{(D+1, D) \veebar (S+1, S)}$ → (D+1, D)		*2	●	7-27
	DXNRP	DXNRP S D					
	DXNR	DXNR S1 S2 D	· $\overline{(S1+1, S1) \veebar (S2+1, S2)}$ → (D+1, D)		*3	● *4	7-30
	DXNRP	DXNRP S1 S2 D					
	BKXNR	BKXNR S1 S2 D n			5	–	7-33
	BKXNRP	BKXNRP S1 S2 D n					

*1: 只有在使用通用型 QCPU 时，基本步数才为 3。
*2: 执行步数随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	5 注释 1)
	使用除上述以外的软元件时	3 注释 2)
基本型 QCPU 通用型 QCPU QnACPU	可以使用所有软元件	3 注释 2)

注释 1) 对于高性能型 QCPU、过程 CPU 及冗余 CPU。执行步数更多但是执行速度也快。
注释 2) 执行步数会根据在 3.8 节中描述的条件而增加。

*3: 执行步数随使用的软元件和 CPU 模块类型的不同而变化。

CPU 类型	软元件	步数
高性能型 QCPU 过程 CPU 冗余 CPU	• 字软元件： 内部软元件（文件寄存器 ZR 除外） • 位软元件： 软元件号是 16 的倍数，其位数指定是 K8，无变址修饰。 • 常数： 无限制	6 注释 1)
	使用除上述以外的软元件时	4 注释 2)
基本型 QCPU QnACPU	可以使用所有软元件	4 注释 2)
通用型 QCPU		3 注释 2)

注释 1: 对于高性能型 QCPU、过程 CPU 及冗余 CPU，执行步数更多但是执行速度也快。
注释 2: 执行步数会根据在 3.8 节中描述的条件而增加。

*4: 只有在使用 QCPU 时，子集才会有效。

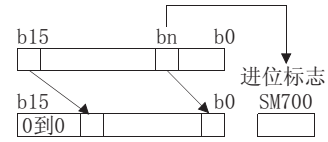
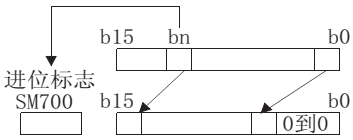
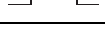
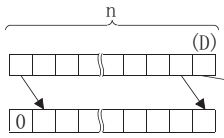
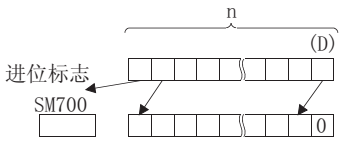
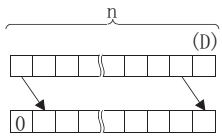
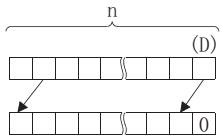
2.5.2 循环指令

表 2.19 循环指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
向右循环移位	ROR	ROR D n	b15 (D) b0 SM700 向右旋转移动n位 进位标志		3	●	7-35
	RORP	RORP D n					
	RCR	RCR D n	b15 (D) b0 SM700 向右旋转移动n位 进位标志		3	●	
	RCRP	RCRP D n					
向左循环移位	ROL	ROL D n	SM700 b15 (D) b0 进位标志 向左旋转移动n位		3	●	7-38
	ROLP	ROLP D n					
	RCL	RCL D n	SM700 b15 (D) b0 进位标志 向左旋转移动n位		3	●	
	RCLP	RCLP D n					
向右循环移位	DROR	DROR D n	(D+1) (D) b31 到 b16 b15 到 b0 SM700 向右旋转移动n位 进位标志		3	●	7-41
	DRORP	DRORP D n					
	DRCR	DRCR D n	(D+1) (D) b31 到 b16 b15 到 b0 SM700 向右旋转移动n位 进位标志		3	●	
	DRCRP	DRCRP D n					
向左循环移位	DROL	DROL D n	(D+1) (D) SM700 b31 到 b16 b15 到 b0 进位标志 向左旋转移动n位		3	●	7-44
	DROLP	DROLP D n					
	DRCL	DRCL D n	(D+1) (D) SM700 b31 到 b16 b15 到 b0 进位标志 向左旋转移动n位		3	●	
	DRCLP	DRCLP D n					

2.5.3 位指令

表 2.20 移位指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
n 位移位	SFR	SFR D n			3	●	7-46
	SFRP	SFRP D n					
	SFL	SFL D n			3	●	
	SFLP	SFLP D n					
1 位移位	BSFR	BSFR D n			3	-	7-49
	BSFRP	BSFRP D n					
	BSFL	BSFL D n			3	-	
	BSFLP	BSFLP D n					
1- 字移位	DSFR	DSFR D n			3	●	7-51
	DSFRP	DSFRP D n					
	DSFL	DSFL D n			3	●	
	DSFLP	DSFLP D n					

2.5.4 处理指令

表 2.21 位处理指令

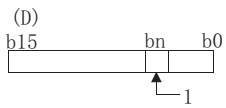
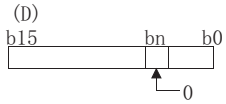
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
位置位 / 复位	BSET	BSET D n			3	●	7-53
	BSETP	BSETP D n					
	BRST	BRST D n			3	●	
	BRSTP	BRSTP D n					

表 2.21 处理指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
位测试	TEST	TEST S1 S2 D	(S1) b15 到 b0 (D) 由(S2)指定的位		4	–	7-55
	TESTP	TESTP S1 S2 D					
	DTEST	DTEST S1 S2 D	(S1) b31 到 b0 (D) 由(S2)指定的位		4	–	
	DTESTP	DTESTP S1 S2 D					
位软元件批量复位	BKRST	BKRST S n	(S) ON OFF ON ON 复位 (S) OFF OFF OFF OFF n		3	–	7-58
	BKRSTP	BKRSTP S n					

2.5.5 据处理指令

表 2.22 据处理指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
数据查找	SER	SER S1 S2 D n	(S1) (S2) n (D)：匹配号 (D + 1)：匹配数量		5	–	7-60
	SERP	SERP S1 S2 D n					
	DSER	DSER S1 S2 D n	32位 (S1) (S2) n (D)：匹配号 (D + 1)：匹配数量		5	–	
	DSERP	DSERP S1 S2 D n					
位检测	SUM	SUM S D	(S) b15 b0 (D)：1s数		3	●	7-63
	SUMP	SUMP S D					
	DSUM	DSUM S D	(S + 1) (S) (D)：1s数		3	●	
	DSUMP	DSUMP S D					
解码	DECO	DECO S D n	从8到25解码 (S) 解码 (D) n 2^n 位		4	–	7-65
	DECOP	DECOP S D n					
编码	ENCO	ENCO S D n	从256到8编码 (S) 2^n 位 编码 (D) n		4	–	7-67
	ENCOP	ENCOP S D n					

表 2.22 据处理指令（续表）

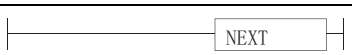
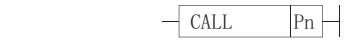
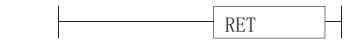
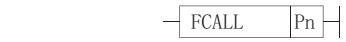
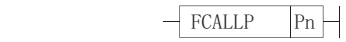
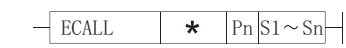
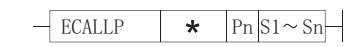
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
7 段解码	SEG	SEG S D	b3到b0 (S) (D) 7SEG		3	●	7-69
	SEGP	SEGP S D					
分解和链接	DIS	DIS S D n	• 将由 (S) 指定的 16 位数据分解成 4 位单元，并将结果存储到从 (D) 开始的 n 个数据点的低 4 位中。(n ≦ 4)		4	–	7-71
	DISP	DISP S D n					
	UNI	UNI S D n	• 链接由 (S) 指定的软元件的 n 个数据点的低 4 位，并将结果存储到由 (D) 指定的软元件中。(n ≦ 4)		4	–	7-73
	UNIP	UNIP S D n					
	NDIS	NDIS S1 D S2	• 将 (S1) 指定的软元件以后的软元件中的数据分解到 (S2) 指定的软元件以后的位中，并按顺序存储到由 (D) 指定的软元件后的软元件中。		4	–	7-75
	NDISP	NDISP S1 D S2					
	NUNI	NUNI S1 D S2	• 将 (S1) 指定的软元件以后的软元件中的数据链接到 (S2) 指定的软元件以后的位中，并按顺序存储到由 (D) 指定的软元件后的软元件中。				
	NUNIP	NUNIP S1 D S2					
	WTOB	WTOB S D n	• 将由 (S) 指定的软元件中的 16 位数据分解成 8 位数据单元，并按顺序存储到由 (D) 指定的软元件中。		4	–	7-79
	WTOBP	WTOBP S D n					
	BTOW	BTOW S D n	• 将由 (S) 指定的软元件开始的 n 个数据点的 16 位数据的低 8 位数据链接成 16 位数据单元，并按顺序存储到由 (D) 指定的软元件中。				
	BTOWP	BTOWP S D n					
搜索	MAX	MAX S D n	• 以 16 位数据为单元，对由 (S) 指定的软元件开始的 n 个数据点进行数据搜索，并将最大值存储到由 (D) 指定软元件中。		4	–	7-83
	MAXP	MAXP S D n					
	MIN	MIN S D n	• 以 16 位数据为单元，对由 (S) 指定的软元件开始的 n 个数据点进行数据搜索，并将最小值存储到由 (D) 指定软元件中。				7-86
	MINP	MINP S D n					
	DMAX	DMAX S D n	• 以 32 位数据为单元，对由 (S) 指定的软元件开始的 2n 个数据点进行数据搜索，并将最大值存储到由 (D) 指定软元件中。		4	–	7-83
	DMAXP	DMAXP S D n					
	DMIN	DMIN S D n	• 以 32 位为数据单元，对由 (S) 指定的软元件开始的 2n 个数据点进行数据搜索，并将最小值存储到由 (D) 指定软元件中。				7-86
	DMINP	DMINP S D n					

表 2.22 数据处理指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
排序	SORT	SORT S1 n S2 D1 D2 • S2：在一个运行中进行的比较数 • D1：在排序完成后变为ON的设备软元件 • D2：用于系统使用	• 以 16 位为单位，对由 (S1) 指定的软元件开始的 n 点的数据进行排序。 （需要 n × (n-1)/2 个扫描）		6	-	7-89
	DSORT	DSORT S1 n S2 D1 D2 • S2：在一个运行中进行的比较数 • D1：在排序完成后变为ON的设备软元件 • D2：用于系统使用	• 以 32 位为单位，对由 (S1) 指定的软元件开始的 2n 点的数据进行排序。 （需要 n × (n-1)/2 个扫描）				
总值计算	WSUM	WSUM S D n	• 对由 (S) 指定的软元件开始的n 个数据点的 16 位 BIN 数据进行加操作，并存储到由 (D) 指定的软元件中。		4	-	7-93
	WSUMP	WSUMP S D n					
	DWSUM	DWSUM S D n	• 对由 (S) 指定的软元件开始的n 个数据点的 32 位 BIN 数据进行加操作，并存储到由 (D) 指定的软元件中。				7-95
	DWSUMP	DWSUMP S D n					

2.5.6 结构体创建指令

表 2.23 结构体创建指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
重复数	FOR		• 在 FOR 和 NEXT 之间执行 n 次。		2	–	7-97
	NEXT				1	–	
	BREAK		• 强制结束 FOR 与 NEXT 之间的执行并跳转指针到 Pn。		3	–	7-100
	BREAKP						
子程序调用	CALL	 	• 当输入条件成立时执行子程序Pn。(S1到 Sn 是至子程序的变量。n ≦ 5)		*1 2 + n	● *3	7-102
	CALLP	 					
	RET		• 来自于子程序的复位。		1	–	7-107
	FCALL	 	• 当输入条件未成立时不执行子程序 Pn。(S1 到 Sn 是至子程序的变量。n ≦ 5)		*1 2 + n	–	7-108
	FCALLP	 					
	ECALL	  *: 文件名	• 当输入条件成立时执行指定程序的子程序 Pn。(S1 到 Sn 是至子程序的变量。n ≦ 5)		*2 3 + n	–	7-112
	ECALLP	  *: 文件名					

*1: n 表示用于子程序的变量数。
*2: n 表示用于子程序的变量数和程序名的合计。
程序名步数被计算为 “程序中的字符数 /2”(小数点以下被进位)”
*3: 只有在使用通用型 QCPU 时，子集才会有效。

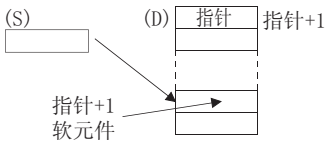
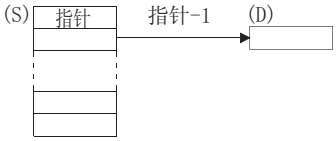
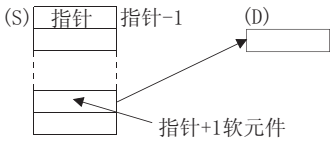
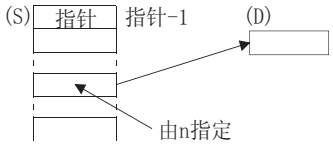
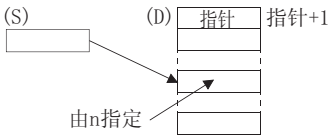
表 2.23 结构体创建指令（续）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
子程序调用	EFCALL	EFCALL * Pn EFCALL * Pn S1toSn *:文件名	• 当输入条件未成立时不执行子程序 Pn。 (S1 到 Sn 是至子程序的变量。 N ≦ 5)	⌐¬	*2 3 + n	–	7-117
	EFCALLP	EFCALLP * Pn EFCALLP * Pn S1toSn *:文件名		¬⌐			
	XCALL	XCALL Pn S1～Sn	• 当输入条件成立时执行子程序 Pn。 • 当输入条件不成立时不执行子程序 Pn。 (S1 到 Sn 是至子程序的变量。N ≦ 5)	⌐¬	*1 2 + n	–	7-121
	COM	COM	• 执行链接刷新和一般数据处理。		1	–	7-126
固定索引修改	IX	IX S 软元件修改梯形图	• 对每个用于软元件修改梯形图中的软元件进行变址修饰。		2	–	7-129
	IXEND	IXEND			1	–	
	IXDEV	IXDEV	• 将 IX 与 IXEND 之间用于进行变址修饰的修饰值存储到 D 中指定的软元件以后。		1	–	7-133
	IXSET	IXSET S D 修饰值的指定			3	–	

*1: 表示用于子程序的变量数。
*2: 表示用于子程序中的变量总数，和程序名称步数。程序名称步数被计算为 “程序中的字符数 /2（十进制余数被舍入）”。

2.5.7 数据表操作指令

表 2.24 数据表操作指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
数据表处理	FIFW	FIFW S D			3	-	7-136
	FIFWP	FIFWP S D					
	FIFR	FIFR S D			3	-	7-138
	FIFRP	FIFRP S D					
	FPOP	FPOP S D			3	-	7-140
	FPOPP	FPOPP S D					
	FDEL	FDEL S D n			4	-	7-142
	FDELP	FDELP S D n					
	FINS	FINS S D n			4	-	
	FINSP	FINSP S D n					

2.5.8 缓冲存储区访问指令

表 2.25 缓冲存储区访问指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
数据读取	FROM	FROM n1 n2 D n3	• 从智能功能模块/特殊功能模块中以16位数据为单元读取数据。	↑	5	-	7-145
	FROMP	FROMP n1 n2 D n3		↑			
	DFRO	DFRO n1 n2 D n3	• 从智能功能模块/特殊功能模块中以32位数据为单元读取数据。	↑	5	-	
	DFROP	DFROP n1 n2 D n3		↑			
数据写	TO	TO n1 n2 S n3	• 以 16 位数据为单元写数据到智能功能模块 / 特殊功能模块。	↑	5	-	7-148
	TOP	TOP n1 n2 S n3		↑			
	DTO	DTO n1 n2 S n3	• 以 32 位数据为单元写数据到智能功能模块 / 特殊功能模块。	↑	5	-	
	DTOP	DTOP n1 n2 S n3		↑			

2.5.9 显示指令

表 2.26 显示指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
ASCII 打印	PR	*SM701为OFF时 PR S D	• 从由(S)指定的软元件中将8点(16字符)的 ASCII 码输出到输出模块。	↑	3	-	7-151
	PR	*SM701为ON时 PR S D	• 从由(S)指定的软元件到00H的ASCII码输出到输出模块。				
	PRC	PRC S D	• 从由 (S) 指定的软元件将注释转换成 ASCII 码，并输出到输出模块。				7-154
显示	LED	LED S	• 从由(S)指定的软元件中将8点(16 字符)的 ASCII 码显示到 CPU 模块前的 LED 显示软元件上。	↑	2	-	7-159
	LEDC	LEDC S	• 从由 (S) 指定的软元件中将注释显示到 CPU 模块前的 LED 显示软元件上。				7-161
复位	LEDR	LEDR	• 复位报警器并显示单元显示。	↑	1	-	7-163

2.5.10 调试和故障诊断指令

表 2.27 调试和故障诊断指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
检查	CHKST	CHKST	• 当 CHKST 可执行时执行 CHK 指令 • 当CHKST处于不可执行状态时，跳至CHK指令之后的步。		1	－	7-166
	CHK	CHK 检查条件	• 在正常条件下 → SM80 : OFF、 SD80 : 0 • 在不正常条件下 → SM80 : ON、 SD80 : 故障 No.				
	CHKCIR	CHKCIR	• 在由 CHK 指令检查的梯形图模式中开始更新。		1	－	7-170
	CHKEND	CHKEND	• 在由 CHK 指令检查的梯形图模式中结束更新。				
状态闭锁	SLT	SLT	• 执行状态闭锁。	↑	1	－	7-178
	SLTR	SLTR	• 复位状态闭锁使其处于可重新执行状态。。				
采样跟踪	STRA	STRA	• 应用触发器到采样跟踪中。	↑	1	－	7-180
	STRAR	STRAR	• 复位采样跟踪以使能重新执行。				
程序跟踪	PTRA	PTRA	• 应用触发器到程序跟踪。	↑	1	－	7-182
	PTRAR	PTRAR	• 复位程序跟踪以使能重新执行。				
	PTRAEXE	PTRAEXE	• 执行程序跟踪。	⊓	1	－	
	PTRAEXEP	PTRAEXEP		↑			

2.5.11 字符串处理指令

表 2.28 字符串处理指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
BIN 到十进制 ASCII	BINDA	BINDA S D	• 将由 (S) 指定的1字BIN值转换成一个5-位数的十进制 ASCII 值，并将它存储到由 (D) 指定的字软元件中。		3	-	7-184
	BINDAP	BINDAP S D					
	DBINDA	DBINDA S D	• 将由 (S) 指定的2字BIN值转换成一个10-位数的十进制 ASCII 值，并将它存储到由 (D) 指定号码的字软元件后的字软元件单元中。		3	-	
	DBINDAP	DBINDAP S D					
BIN 到十六进制 ASCII	BINHA	BINHA S D	• 将由 (S) 指定的1字BIN值转换成一个4-位数的十六进制 ASCII 值，并将它存储到由 (D) 指定的字软元件号码后的字软元件单元中。		3	-	7-187
	BINHAP	BINHAP S D					
	DBINHA	DBINHA S D	• 将由 (S) 指定的 2 字 BIN 值转换成一个 8-位数的十六进制 ASCII 值，并将它存储到由 (D) 指定号码的字软元件后的字软元件单元中。		3	-	
	DBINHAP	DBINHAP S D					
BCD 到十进制 ASCII	BCDDA	BCDDA S D	• 将由 (S) 指定的1字BCD值转换成一个4-位数的十进制 ASCII 值，并将它存储到由 (D) 指定号码的字软元件后的字软元件单元中。		3	-	7-190
	BCDDAP	BCDDAP S D					
	DBCDDA	DBCDDA S D	• 将由 (S) 指定的2字BCD值转换成一个8-位数的十进制 ASCII 值，并将它存储到由 (D) 指定号码的字软元件后的字软元件单元中。		3	-	
	DBCDDAP	DBCDDAP S D					
十进制 ASCII 到 BIN	DABIN	DABIN S D	• 将由 (S) 指定的 5-位数的十进制ASCII 值转换成一个 1 字 BIN 值，并将它存储到由 (D) 指定号码的字软元件中。		3	-	7-193
	DABINP	DABINP S D					
	DDABIN	DDABIN S D	• 将由 (S) 指定的10-位数的十进制ASCII值转换成一个 2 字 BIN 值，并将它存储到由 (D) 指定号码的字软元件中。		3	-	
	DDABINP	DDABINP S D					
十六进制 ASCI 到 BIN	HABIN	HABIN S D	• (S) 指定的 4-位数的十六进制 ASCII 值转换成一个 1 字 BIN 值，并将它存储到由 (D) 指定号码的字软元件中。		3	-	7-196
	HABINP	HABINP S D					
	DHABIN	DHABIN S D	• 由 (S) 指定的8-位数的十六进制 ASCII 值转换成一个 2 字 BIN 值，并将它存储到由 (D) 指定号码的字软元件中。		3	-	
	DHABINP	DHABINP S D					

表 2.28 字符串处理指令（续表）

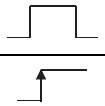
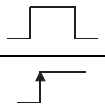
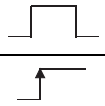
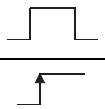
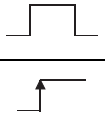
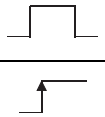
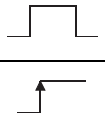
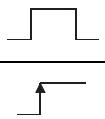
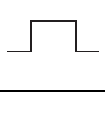
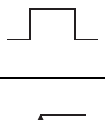
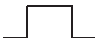
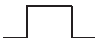
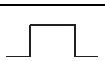
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
十进制 ASCII 到 BCD	DABCD	DABCD S D	• 将由(S)指定的一个4位的十进制ASCII值转换成 1 字 BCD 值，并将它存储到由 (D) 指定的字软元件中。		3	-	7-199
	DABCDP	DABCDP S D					
	DDABCD	DDABCD S D	• 将由(S)指定的一个8位的十进制ASCII值转换成 2 字 BIN 值，并将它存储到由 (D) 指定号码的字软元件后的字软元件单元中。		3	-	
	DDABCDP	DDABCDP S D					
软元件注释读操作	COMRD	COMRD S D	• 将从 (S) 指定的软元件开始地注释存储到由 (D) 指定的软元件中。		3	-	7-202
	COMRDP	COMRDP S D					
字符串长度检测	LEN	LEN S D	• 将从 (S) 指定的字符串中的数据长度（字节数），存储到由 (D) 指定的软元件中。		3	-	7-207
	LENP	LENP S D					
BIN 到十进制字符串	STR	STR S1 S2 D	• 将由(S2)指定的1字BIN 值，按照(S1)指定的总位数和小数点位数，转换成十进制字符串，并将它们存储到由 (D) 指定号码的软元件中。		4	-	7-209
	STRP	STRP S1 S2 D					
	DSTR	DSTR S1 S2 D	• 将由(S2)指定的2字BIN值，按照(S1)指定的总位数和小数部分的位数，转换成十进制字符串，并将它们存储到由 (D) 指定号码的软元件中。		4	-	
	DSTRP	DSTRP S1 S2 D					
十进制字符串到 BIN	VAL	VAL S D1 D2	• 将由 (S) 指定的包含小数点的字符串，转换成 1 字 BIN 值和小数部分的位数，并将它们存储到由 (D1) 和 (D2) 指定号码的软元件中。		4	-	7-215
	VALP	VALP S D1 D2					
	DVAL	DVAL S D1 D2	• 将由 (S) 指定的包含小数点的字符串，转换成 2 字 BIN 值和小数部分的位数，并将它们存储到由 (D1) 和 (D2) 指定号码的软元件中。		4	-	
	DVALP	DVALP S D1 D2					
浮点十进制数到字符串	ESTR	ESTR S1 S2 D	• 将由 (S) 指定 32 位浮点数据转换成字符串，并将它们存储到由 (D) 指定软元件。		4	-	7-220
	ESTRP	ESTRP S1 S2 D					
字符串到浮点十进制数	EVAL	EVAL S D	• 将由 (S) 指定的字符串转换成浮点十进制数据，并将它们存储到由 (D) 指定号码的软元件中。		3	-	7-226
	EVALP	EVALP S D					

表 2.28 字符串处理指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
十六进制 BIN 到 ASCII	ASC	ASC S D n	• 将由 (S) 指定号码的软元件及其之后的软元件中的 1 字 BIN 值转换成 ASCII，并只存储它们中的 n 个字符到由 (D) 指定号码的软元件中。		4	-	7-230
	ASCP	ASCP S D n					
ASCII 到十六进制 BIN	HEX	HEX S D n	• 只将由 (S) 指定号码的软元件及其之后的软元件中的 n 个 ASCII 字符转换成 BIN 值，并将它们存储到由 (D) 指定号码的软元件中。		4	-	7-232
	HEXP	HEXP S D n					
字符串处理	RIGHT	RIGHT S D n	• 将从 (S) 指定的字符串的最后开始的 n 个字符存储到由 (D) 指定的软元件中。		4	-	7-234
	RIGHTP	RIGHTP S D n					
	LEFT	LEFT S D n	• 将从 (S) 指定的字符串的开始的 n 个字符存储到由 (D) 指定的软元件中。				
	LEFTP	LEFTP S D n					
	MIDR	MIDR S1 D S2	• 将在 (S1) 指定的字符串中从 (S2) 指定的位置开始的指定数量的字符存储到由 (D) 指定的软元件中。		4	-	7-237
	MIDRP	MIDRP S1 D S2					
	MIDW	MIDW S1 D S2	• 将在 (S1) 指定的字符串中从 (S2) 指定的位置开始的指定数量的字符存储到由 (D) 指定的软元件中。				
	MIDWP	MIDWP S1 D S2					
	INSTR	INSTR S1 S2 D n	• 从字符串 (S2) 的第n个字符开始搜索字符串 (S1)，并存储匹配的位置到 (D)。		5	-	7-241
	INSTRP	INSTRP S1 S2 D n					
浮点十进制数到 BCD	EMOD	EMOD S1 S2 D	• 按照由 (S2) 指定的十进制小数部分的位数将浮点十进制小数数据 (S1) 转换成 BCD 数据，并存储到由 (D) 指定软元件中。		4	-	7-243
	EMODP	EMODP S1 S2 D					
BCD 到浮点十进制数据	EREXP	EREXP S1 S2 D	• 按照由 (S2) 指定的十进制小数部分的位数将 BCD 数据 (S1) 转换成浮点十进制小数数据，并存储到由 (D) 指定的软元件中。		4	-	7-246
	EREXPP	EREXPP S1 S2 D					

2.5.12 特殊功能指令

表 2.29 特殊功能指令

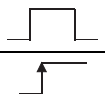
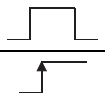
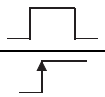
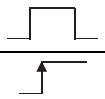
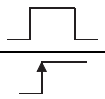
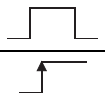
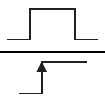
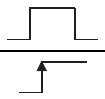
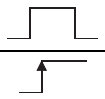
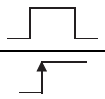
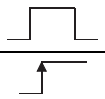
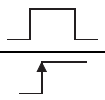
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
三角函数功能（浮点数据单精度）	SIN	SIN S D	• Sin (S+1, S) ⟶ (D+1, D)		3	-	7-248
	SINP	SINP S D					
	COS	COS S D	• Cos (S+1, S) ⟶ (D+1, D)		3	-	7-252
	COSP	COSP S D					
	TAN	TAN S D	• Tan(S+1, S) ⟶ (D+1, D)		3	-	7-256
	TANP	TANP S D					
	ASIN	ASIN S D	• Sin^{-1}(S+1, S) ⟶ (D+1, D)		3	-	7-260
	ASINP	ASINP S D					
	ACOS	ACOS S D	• Cos^{-1}(S+1, S) ⟶ (D+1, D)		3	-	7-265
	ACOSP	ACOSP S D					
	ATAN	ATAN S D	• Tan^{-1}(S+1, S) ⟶ (D+1, D)		3	-	7-269
	ATANP	ATANP S D					
三角函数功能（浮点数据双精度）	SIND	SIND S D	Sin(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-250
	SINDP	SINDP S D					
	COSD	COSD S D	Cos(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-254
	COSDP	COSDP S D					
	TAND	TAND S D	Tan(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-258
	TANDP	TANDP S D					
	ASIND	ASIND S D	Sin^{-1}(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-263
	ASINDP	ASINDP S D					
	ACOSD	ACOSD S D	Cos^{-1}(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-267
	ACOSDP	ACOSDP S D					
	ATAND	ATAND S D	Tan^{-1}(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D)		3	-	7-271
	ATANDP	ATANDP S D					

表 2.29 特殊功能指令（续表）

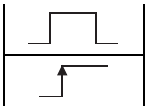
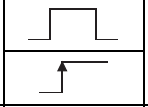
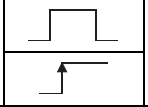
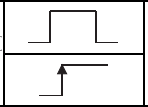
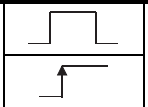
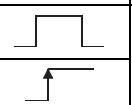
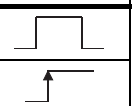
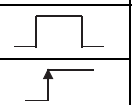
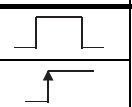
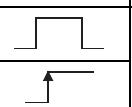
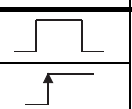
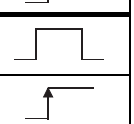
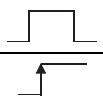
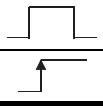
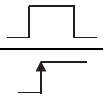
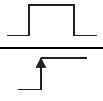
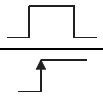
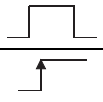
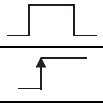
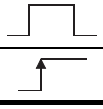
类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
在角度和弧度间的转换	RAD	RAD S D	• (S+1, S) ⟶ (D+1, D) 将角度转换成弧度		3	-	7-273
	RADP	RADP S D					
	RADD	RADD S D	· (S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, 将角度转换成弧度		3	-	7-275
	RADDP	RADDP S D					
	DEG	DEG S D	• (S+1, S) ⟶ (D+1, D) 将弧度转换成角度		3	-	7-277
	DEGP	DEGP S D					
	DEGD	DEGD S D	· (S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D 将弧度转换成角度		3	-	7-279
	DEGDP	DEGDP S D					
平方根	SQR	SQR S D	• $\sqrt{(S+1,S)}$ ⟶ (D+1 , D)		3	-	7-281
	SQRP	SQRP S D					
	SQRD	SQRD S D	$\sqrt{(S+3,\ S+2,\ S+1,\ S)}$ ⟶ (D+3, D+2, D+1,		3	-	7-283
	SQRDP	SQRDP S D					
指数操作	EXP	EXP S D	• $e^{(S+1,\ S)}$ ⟶ (D+1 , D)		3	-	7-285
	EXPP	EXPP S D					
	EXPD	EXPD S D	$e^{(S+3,\ S+2,\ S+1,\ S)}$ ⟶ (D+3, D+2, D+1, D)		3	-	7-288
	EXPDP	EXPDP S D					
自然对数	LOG	LOG S D	• Log_e(S+1 , S) ⟶ (D+1 , D)		3	-	7-290
	LOGP	LOGP S D					
	LOGD	LOGD S D	Log_e(S+3, S+2, S+1, S) ⟶ (D+3, D+2, D+1, D		3	-	7-292
	LOGDP	LOGDP S D					
随机数的产生	RND	RND D	• 产生一个随机数(从0到小于32767)并将它存储到由 (D) 指定的软元件中。		2	-	7-294
	RNDP	RNDP D					
随机数序列的更新	SRND	SRND D	• 按照存储在由 (S) 指定的软元件中的 16- 位 BIN 数据更新随机数序列。				
	SRNDP	SRNDP D					

表 2.29 特殊功能指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
平方根	BSQR	BSQR S D	• $\sqrt{(S)}$ → (D)+0 整数部分 +1 小数部分		3	-	7-296
	BSQRP	BSQRP S D					
	BDSQR	BDSQR S D	• $\sqrt{(S+1, S)}$ → (D)+0 整数部分 +1 小数部分		3	-	
	BDSQRP	BDSQRP S D					
三角函数功能	BSIN	BSIN S D	• Sin(S) → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-299
	BSINP	BSINP S D					
	BCOS	BCOS S D	• Cos(S) → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-301
	BCOSP	BCOSP S D					
	BTAN	BTAN S D	• Tan(S) → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-303
	BTANP	BTANP S D					
	BASIN	BASIN S D	• $Sin^{-1}(S)$ → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-305
	BASINP	BASINP S D					
	BACOS	BACOS S D	• $Cos^{-1}(S)$ → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-307
	BACOSP	BACOSP S D					
	BATAN	BATAN S D	• $Tan^{-1}(S)$ → (D)+0 正负号 +1 整数部分 +2 小数部分		3	-	7-309
	BATANP	BATANP S D					

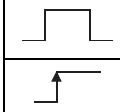
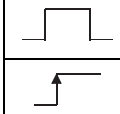
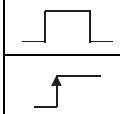
2.5.13 数据控制指令

表 2.30 数据控制指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
最高和最低限控制	LIMIT	LIMIT S1 S2 S3 D	• 当 (S3) < (S1) 时 ..... 存储 (S1) 的值到 (D) • 当 (S1) ≦ (S3) ≦ (S2) ..... 存储 (S3) 的值到 (D) • 当 (S2) < (S3) ..... 存储 (S2) 的值到 (D)		5	-	7-311
	LIMITP	LIMITP S1 S2 S3 D					
	DLIMIT	DLIMIT S1 S2 S3 D	• 当 ((S3)+1、(S3)) < ((S1)+1、S1) 时 .存储 ((S1)+1、(S1)) 的值到 ((D)+1、(D)) • 当 ((S1)+1、(S1)) ≦ ((S3)+1、(S3)) < (S2+1、S2) .存储 (S3)+1、(S3)) 的值到 ((D)+1、(D)) • 当 ((S2)、(S2)+1) < ((S3)、(S3)+1) .存储 ((S2)+1、(S2)) 的值到 ((D)+1、(D))		5	-	
	DLIMITP	DLIMITP S1 S2 S3 D					
死区控制	BAND	BAND S1 S2 S3 D	• 当 (S1) ≦ (S3) ≦ (S2) ..... 0 → (D) • 当 (S3) < (S1)..... (S3) − (S1) → (D) • 当 (S2) < (S3)..... (S3) − (S2) →		5	-	7-314
	BANDP	BANDP S1 S2 S3 D					
	DBAND	DBAND S1 S2 S3 D	• 当 ((S1)+1、(S1)) ≦ ((S3)+1、(S3)) ≦ ((S2)+1、(S2)) .... 0 → ((D)+1、(D)) • 当 ((S3)+1、(S3)) < ((S1)+1、(S1)).. ((S3)+1、(S3)) − ((S1)+1、(S1)) → ((D)+1、(D)) • 当 ((S2)+1、(S2)) < ((S3)+1、(S3)).. ((S3)+1、(S3)) − ((S2)+1、(S2)) → ((D)+1、(D))		5	-	
	DBANDP	DBANDP S1 S2 S3 D					
区域控制	ZONE	ZONE S1 S2 S3 D	• 当 (S3) = 0..... 0 → (D) • 当 (S3) > 0..... (S3)+(S2) → (D) • 当 (S3) < 0..... (S3) − (S1) → (D)		5	-	7-317
	ZONEP	ZONEP S1 S2 S3 D					
	DZONE	DZONE S1 S2 S3 D	• 当 ((S3)+1、(S3)) = 0 ..... 0 → ((D)+1、(D)) • 当 ((S3)+1、(S3)) > 0 ..... ((S3)+1、(S3))+((S2)+1、(S2)) → ((D)+1、(D)) • 当 ((S3)+1、(S3)) < 0 .... ((S3)+1、(S3)) + ((S1)+1、(S1)) → ((D)+1、(D))		5	-	
	DZONEP	DZONEP S1 S2 S3 D					

2.5.14 转换指令

表 2.31 转换指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
块号码标定	RSET	RSET S	• 将扩展文件寄存器块的号码转换成由 (S) 指定的号码。		2	-	7-320
	RSETP	RSETP S					
文件设定	QDRSET	QDRSET 文件设定	• 将文件名称用作文件寄存器。		*1 2 + n	-	7-322
	QDRSETP	QDRSETP 文件设定					
	QCDSET	QCDSET 文件设定	• 将文件名称用作注释文件。		*1 2 + n	-	7-325
	QCDSETP	QCDSETP 文件设定					

*1: n([文件名称字符数]/2) 表示一步。(小数部分舍入)

2.5.15 时钟指令

表 2.32 时钟指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
读 / 写时钟数据	DATERD	DATERD D	• (时钟设备元件) →(D)+0 年 +1 月 +2 日 +3 小时 +4 分 +5 秒 +6 星期		2	-	7-327
	DATERDP	DATERDP D					
	DATEWR	DATEWR S	• (D)+0 年 →(时钟设备元件) +1 月 +2 日 +3 小时 +4 分 +5 秒 +6 星期		2	-	7-331
	DATEWRP	DATEWRP S					
时钟数据的加 / 减	DATE+	DATE+ S1 S2 D	(S1) 小时 分钟 秒 + (S2) 小时 分钟 秒 → (D) 小时 分钟 秒		4	-	7-335
	DATE+P	DATE+P S1 S2 D					
	DATE-	DATE- S1 S2 D	(S1) 小时 分钟 秒 - (S2) 小时 分钟 秒 → (D) 小时 分钟 秒		4	-	7-337
	DATE-P	DATE-P S1 S2 D					
时钟数据传送	SECOND	SECOND S D	(S) 小时 分钟 秒 → (D) 秒(低级1) 秒(高级1)		3	-	7-339
	SECONDP	SECONDP S D					
	HOUR	HOUR S D	(S) 秒(低级16) 秒(高级16) → (D) 小时 分钟 秒		3	-	7-341
	HOURP	HOURP S D					

2.5.16 外围软元件指令

表 2.33 外围软元件指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
输入/输出到外围软元件	MSG	MSG S	• 将由 (S) 指定的信息存储到 SD738 到 SD769。 此信息将显示在外围设备上。		2	-	7-343
	PKEY	PKEY D	• 从外围软元件中输入的数据存储到由 (D) 指定的软元件上。		2	-	7-345

2.5.17 程序控制指令

表 2.34 程序控制指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
程序控制指令	PSTOP	PSTOP 文件名称	• 将指定的程序置为备用状态。		*1 2+n	-	7-349
	PSTOPP	PSTOPP 文件名称					
	POFF	POFF 文件名称	• 将指定程序的 OUT 指令线圈置为 OFF，并将指定的程序置为备用状态。		*1 2+n	-	7-350
	POFFP	POFFP 文件名称					
	PSCAN	PSCAN 文件名称	• 将指定程序登记为扫描执行类型。		*1 2+n	-	7-352
	PSCANP	PSCANP 文件名称					
	PLOW	PLOW 文件名称	• 将指定程序登记为低速执行类型。		*1 2+n	-	7-354
	PLOWP	PLOWP 文件名称					
	LDPCHK	PCHK 文件名称	• 当特定文件名称的程序被执行时进入导通状态。 • 当特定文件名称的程序不被执行时进入不导通状态。		*1 2+n	-	7-356
	ANDPCHK	PCHK 文件名称					
	ORPCHK	PCHK 文件名称					

*1: n([程序名称字符数]/2) 表示一步（小数部分舍入）

2.5.18 其他指令

表 2.35 其他指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
WDT 复位	WDT	WDT	• 在顺序程序中复位看门狗定时器。		1	–	7-358
	WDTP	WDTP					
定时时钟	DUTY	DUTY n1 n2 D	(D) n1扫描 n2扫描 SM420到SM424, SM430到SM434		4	–	7-360
时间检查	TIMCHK	TIMCHK S1 S2 D	• 如果输入条件测量到的ON时间连续超过设定时间，则将由 (D) 指定的软元件转为 ON。		4	–	7-362
在 1 一字节单元内进行直接读 / 写操作	ZRRDB	ZRRDB n D	0 低8位 ZR0 1 高8位 2 低8位 ZR1 3 高8位 n 8位 → (D)		3	–	7-363
	ZRRDBP	ZRRDBP n D					
	ZRWRB	ZRWRB n S	(S) → 0 低8位 ZR0 1 高8位 2 低8位 ZR1 3 高8位 n 8位		3	–	7-365
	ZRWRBP	ZRWRBP n S					
	ADRSET	ADRSET S D	(S) → (D) 表示指定设备元件的地址 设备元件名称		3	–	7-367
	ADRSETP	ADRSETP S D					
来自键盘的数字键输入	KEY	KEY S n D1 D2	• 将电 (S) 指定的输入单元的 8 个数据点的 ASCII 数据，转换成十六进制值到 (D1) 指定号码的软元件后，并存储。		5	–	7-368
变址寄存器的批量保存	ZPUSH	ZPUSH D	• 将变址寄存器Z0到Z15的注释存储到从(D)指定的软元件开始的区域。		2	–	7-372
	ZPUSHP	ZPUSHP D					
变址寄存器的批量恢复	ZPOP	ZPOP D	• 将存储在 (D) 指定的软元件开始的区域内的数据读到变址寄存器 Z0 到 Z15。				
	ZPOPP	ZPOPP D					
对 E2PROM 文件寄存器的批量写操作	EROMWR	EROMWR S D1 n D2	• 将批数据写入到 E2PROM 文件寄存器内。		5	–	7-374
	EROMWRP	EROMWRP S D1 n D2					

2.5.19 用于数据链接的指令

表 2.36 用于数据链接的指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
Q 链接指令：网络刷新	S.ZCOM	S.ZCOM Jn	刷新指定的网络。		5	-	8-10
	SP.ZCOM	SP.ZCOM Jn					
	S.ZCOM	S.ZCOM Un					
	SP.ZCOM	SP.ZCOM Un					
QnA 链接指令：网络刷新	J.ZCOM	J.ZCOM Jn	刷新指定的网络。		5	-	8-14
	JP.ZCOM	JP.ZCOM Jn					
	G.ZCOM	G.ZCOM Un					
	GP.ZCOM	GP.ZCOM Un					
QnA 链接指令：从其他站读取数据	JP.READ	JP.READ Jn S1 S2 D1 D2	将其他站的字软元件数据读到自站中。		9	-	8-18
	GP.READ	GP.READ Un S1 S2 D1 D2					
	JP.SREAD	JP.SREAD Jn S1 S2 D1 D2 D3			10	-	8-29
	GP.SREAD	GP.SREAD Un S1 S2 D1 D2 D3					
QnA 链接指令：将数据写入到其他站	JP.WRITE	JP.WRITE Jn S1 S2 D1 D2	将自站的数据写入到其他站的字软元件中。		10	-	8-40
	GP.WRITE	GP.WRITE Un S1 S2 D1 D2					
	JP.SWRITE	JP.SWRITE Jn S1 S2 D1 D2 D3			11	-	8-51
	GP.SWRITE	GP.SWRITE Un S1 S2 D1 D2 D3					
QnA 链接指令：发送数据	JP.SEND	JP.SEND Jn S1 S2 D	发送数据（信息）到其他站。		8	-	8-62
	GP.SEND	GP.SEND Un S1 S2 D					
QnA 链接指令：接收数据	JP.RECV	JP.RECV Jn S D1 D2	接收送往自站的数据（信息）。		8	-	8-72
	GP.RECV	GP.RECV Un S D1 D2					
QnA 链接指令：来自其他站的瞬时请求	JP.REQ	JP.REQ Jn S1 S2 D1 D2	将一个瞬时请求送往其他站并执行。		8	-	8-81
	GP.REQ	GP.REQ Un S1 S2 D1 D2					

表 2.36 用于数据链接的指令（续表）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
QnA 链接指令：从远程 I/O 站的特殊功能模块中读取数据	JP.ZNFR	JP.ZNFR Jn S1 S2 D	从远程 I/O 站的特殊功能模块中读取数据。	↑	8	–	8-95
	GP.ZNFR	GP.ZNFR Un S1 S2 D		↑			
QnA 链接指令：写数据到远程 I/O 站的特殊功能模块	JP.ZNTO	JP.ZNTO Jn S1 S2 D	写数据到远程 I/O 站的特殊功能模块。	↑	8	–	8-100
	GP.ZNTO	GP.ZNTO Un S1 S2 D		↑			
A- 系列兼容链接指令：从其他站中读取软元件数据	J.ZNRD	J.ZNRD Jn n1 S D1 n2 D2	将其他站中的软元件数据读取到自站。	↑	32	–	8-105 8-109
	JP.ZNRD	JP.ZNRD Jn n1 S D1 n2 D2		↑			
A- 系列兼容链接指令：写软元件数据到其他站。	J.ZNWR	J.ZNWR Jn n1 D1 S n2 D2	将自站的数据写入到其他站的字软元件中。	↑	32	–	8-113 8-117
	JP.ZNWR	JP.ZNWR Jn n1 D1 S n2 D2		↑			
A- 系列兼容链接指令：从远程 I/O 站的特殊功能模块中读取数据	G.RFRP	G.RFRP Un n1 D1 n2 D2	将自站的数据写入到其他站的字软元件中。	↑	11	–	8-121
	GP.RFRP	GP.RFRP Un n1 D1 n2 D2		↑			
A- 系列兼容链接指令：写数据到远程 I/O 站的特殊功能模块中	G.RTOP	G.RTOP Un n1 S n2 D	写数据到远程 I/O 站的特殊功能模块中。	↑	11	–	8-125
	GP.RTOP	GP.RTOP Un n1 S n2 D		↑			
读取路由信息	S.RTREAD	S.RTREAD n D	读取路由参数设定数据。	↑	7	–	8-129 8-131
	SP.RTREAD	SP.RTREAD n D		↑			
	Z.RTREAD	Z.RTREAD n D		↑			
	ZP.RTREAD	ZP.RTREAD n D		↑			
登记路由信息	S.RTWRITE	S.RTWRITE n S	写路由数据到由路由参数指定的区域内。	↑	8	–	8-133 8-135
	SP.RTWRITE	SP.RTWRITE n S		↑			
	Z.RTWRITE	Z.RTWRITE n S		↑			
	ZP.RTWRITE	ZP.RTWRITE n S		↑			

2.5.20 QCPU 指令

表 2.37 QCPU 指令

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
读取模块信息	UNIRD	UNIRD n1 D n2	• 读取存储在从(n)指定号码的I/O开始的区域内，由 (n2) 指定点数的模块信息，并存储它们到 (d) 指定的软元件开始的区域内。		4	-	9-2
	UNIRDP	UNIRDP n1 D n2					
跟踪设定	TRACE	TRACE	• 当 SM800，SM801，和 SM802 变为 ON 时，将设定在外围软元件上的跟踪数据存储到指定号码的内存卡中的跟踪文件内。		1	-	9-6
跟踪复位	TRACER	TRACER	• 将由 TRACE 指令设置的数据复位。		1	-	9-6
写数据到指定的文件	SP.FWRITE	SP.FWRITE U0 S0 D0 S1 S2 D1	• 写数据到指定的文件。		11	-	9-8
从指定的文件中读取数据	SP.FREAD	SP.FREAD U0 S0 D0 S1 S2 D1	• 从指定的文件中读取数据。		11	-	9-19
写数据到标准 ROM	S.DEVST	SP.DEVST n1 S n2 D	• 写数据到标准ROM中的软元件数据存储文件。		9	-	9-31
从标准 ROM 中读取数据	S.DEVLD	S.DEVLD n1 D n2	• 从标准 ROM 中的软元件数据存储文件中读取数据。		8	-	9-33
	SP.DEVLD	SP.DEVLD n1 D n2					9-33
从存储卡中装载程序	PLOADP	PLOADP S D	• 将存储在存储卡或标准存储器(驱动器0外的其他软元件）中的程序传送到启动器 0，并将程序置于备用状态。		3	-	9-35
从程序存储器中卸载程序	PUNLOADP	PUNLOADP S D	• 删除存储在标准存储器（驱动器 0）中的备用程序。		3	-	9-38
装载 + 卸载	PSWAPP	PSWAPP S1 S2 D	• 删除存储在由 (S1) 指定的标准存储器（驱动器 0）中的备用程序。然后，将存储在存储卡或由 (S2) 指定的标准存储器（驱动器 0 外的其他软元件）中的程序传送到驱动器 0，并将程序置于备用状态。		4	-	9-40
文件寄存器中的高速块传送	RBMOV	RBMOV S D n	• 将由 (S) 指定的软元件的 n 数据点的 16 位数据传送到由 (D) 指定的软元件开始的区域内。		4	-	9-43
	RBMOVP	RBMOVP S D n					

表 2.37 QCPU 指令（续）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
写入到自站 CPU 的共享存储器	S. TO	S.TO n1 n2 n3 n4 D	• 将自站的软元件数据写入到主机站 CPU 模块的共享存储区域内。		5	－	9-49
	SP. TO	SP.TO n1 n2 n3 n4 D					
	TO	TO n1 n2 S n3	• 将自站的软元件数据写入到主机站 CPU 模块的 CPU 共享存储器内。		5	－	9-52
	TOP	TOP n1 n2 S n3					
	DTO	DTO n1 n2 S n3	• 以 32 位为单位，将自站的软元件数据写入到自站 CPU 模块的共享存储器内。		5	－	
	DTOP	DTOP n1 n2 S n3					
从其他站 CPU 的共享存储器中读取数据	FROM	FROM n1 n2 D n3	• 将其他站CPU模块的CPU共享存储器区域内的软元件数据读取到主机站。		5	－	9-57
	FROMP	FROMP n1 n2 D n3					
	DFRO	DFRO n1 n2 D n3	• 以32位为单位，将其他站CPU模块的CPU共享存储器器内的软元件数据读取到自站。		5	－	
	DFROP	DFROP n1 n2 D n3					
CPU 共享存储器的自动刷新	COM	COM	• 执行智能功能模块的自动刷新，一般数据处理，和 CPU 共享存储器的自动刷新。		1	－	9-63
读取扩展时钟的数据	S.DATERD	S.DATERD D	·（时钟因子）→(D)+0 年 +1 月 +2 日 +3 小时 +4 分钟 +5 秒 +6 星期 +7 1/1000秒		6	－	9-66
	SP.DATERD	SP.DATERD D					
扩展时钟数据的加和减运算	S.DATE+	S.DATE+ S1 S2 D	(S1) 小时 分钟 秒 — 1/1000秒 + (S2) 小时 分钟 秒 — 1/1000秒 → (D) 小时 分钟 秒 — 1/1000秒		8	－	9-69
	SP.DATE+	SP.DATE+ S1 S2 D					
	S.DATE-	S.DATE S1 S2 D	(S1) 小时 分钟 秒 — 1/1000秒 − (S2) 小时 分钟 秒 — 1/1000秒 → (D) 小时 分钟 秒 — 1/1000秒		8	－	9-72
	SP.DATE-	SP.DATE S1 S2 D					

2.5.21 冗余系统指令（用于 Q4ARCPU）

表 2.38 冗余系统指令（用于 Q4ARCPU）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
在 CPU 启动时的操作模式设置	S.STMODE	S.STMODE S1 S2	• 当系统上电进行 CPU 启动时，确定 (S1) 上的操作模式，在启动前是清除 Q4ARCPU 还是不清除它们。		9	-	10-2
CPU 切换过程中的操作模式设置指令	S.CGMODE	S.CGMODE S	• 当控制由主控系统切换到备用系统时，确定 (S1) 上的操作模式，在启动前是清除 Q4ARCPU 软元件还是不清除它们。		7	-	10-4
数据跟踪	S.TRUCK	S.TRUCK S	• 在END处理过程中，按照存储在由 (S) 指定的软元件开始的区域内的参数块数据内容，对软元件存储器进行跟踪。		6	-	10-6
缓冲存储器的批量刷新	S.SPREF	S.SPREF S	• 按照存储在由 (S) 指定的软元件开始的区域内的参数块数据内容，对特殊功能模块缓冲存储器内的内容进行批量读 / 写。		6	-	10-10

2.5.22 冗余系统指令（用于冗余 CPU）

表 2.39 冗余系统指令（用于冗余 CPU）

类别	指令符号	符号	详细处理过程	执行条件	基本步数	子集	参见章节
系统切换	SP.CONTSW	SP.CONTSW S D	在通过 SP.CONTSW 指令执行的扫描的 END 处理中，进行控制系统和待机系统之间的切换。		8	-	11-2

备忘录

3 指令组态

3.1 指令结构

多数的 CPU 模块指令包括一个指令部分和一个软元件部分。

- 指令部分 ..... 显示指令的功能。
- 软元件部分 ..... 显示指令所需的数据。

软元件部分被分为源数据，目标数据和软元件号。

(1) 源数据 (S)

(a) 源数据是运算所需的数据。

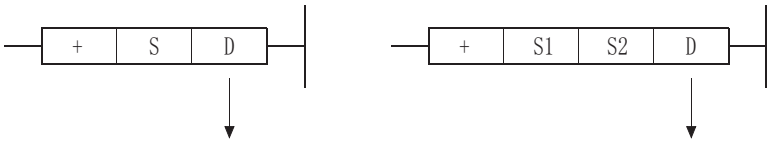
(b) 根据指定的软元件不同，源数据具有以下形式：

- 常数指定将在运算中使用的数值。..... 当程序写好以后设定，在程序执行过程中不能改变。
  当常数需要被当作变量使用时，需要进行变址修饰。
- 位软元件和字软元件 ..... 指定存储运算用数据的软元件。
  数据必须存储在指定的软元件当中，直到运算开始执行。
  通过在程序执行过程中改变指定软元件中存储的数据，在指令中要使用的数据就可以更改。

(2) 目标数据 (D)

(a) 目标数据中存储运算后的数据。
然而，某些指令要求在运算执行之前，在目标数据中存储运算中将要使用的数据。

例 BIN16 位数据的加法指令时



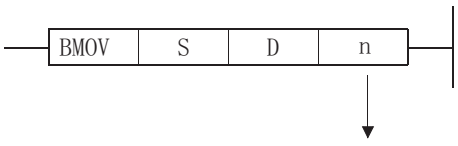
在实际的运算之前储存运算需要的数据。　　只储存运算结果。

(b) 供数据存储用的软元件必须被设置到目标中去。

(3) 软元件号和传输号 (n)

(a) 软元件号和传输号指定了在多软元件指令中要用到的软元件号和传输号。

例 块传输指令

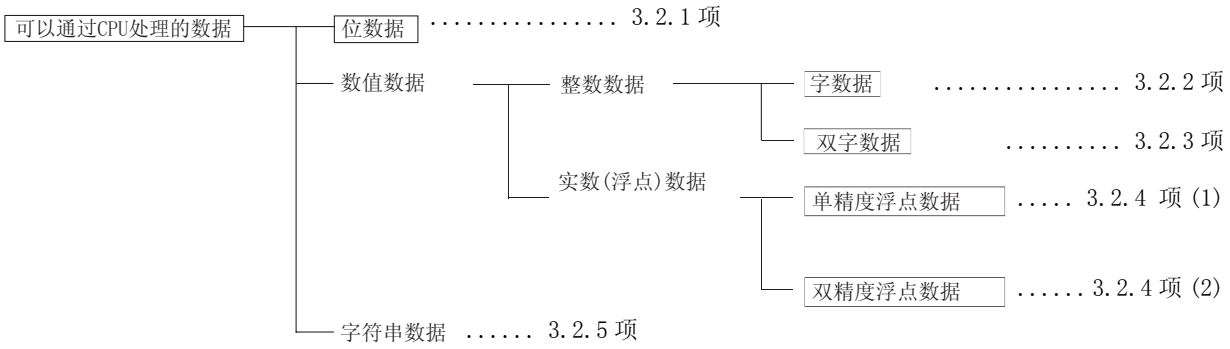


指定BMOV指令使用的传输号

(b) 软元件号或传输号可以设定在 0 和 32767 之间，然而，如果该号为 0，该指令将为无运算指令。

3.2 数据的指定方法

以下六种数据可以用于 CPU 模块指令：



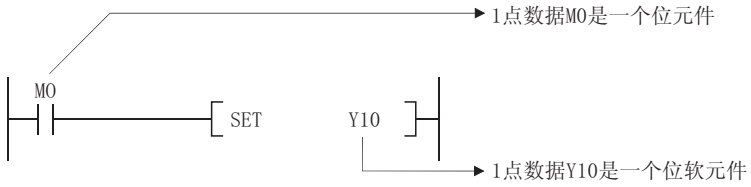
3.2.1 使用位数据

位数据是指以一位为单位使用的数据，如触点或线圈。

“位软元件”和“位指定字软元件”可以被当作位数据使用。

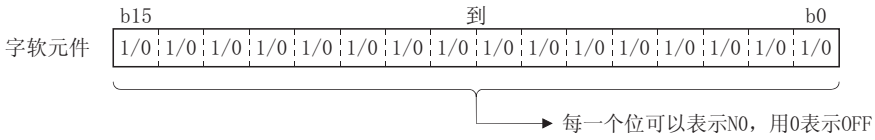
(1) 使用位软元件时

位软元件在一点数据单元中指定。



(2) 使用字软元件时

(a) 字软元件通过指定位号，使得指定位号的 1/0 被用于位数据。

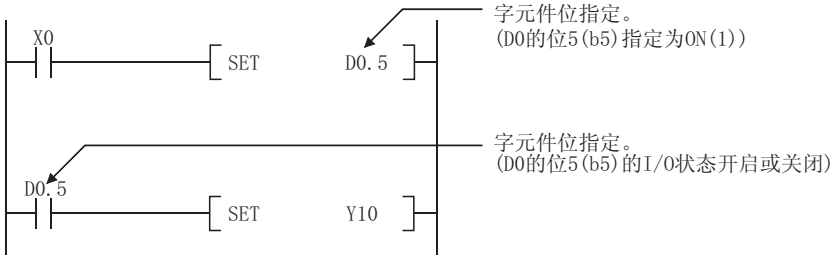


(b) 字软元件的位指定是通过指定“ **字软元件** **位号** ”来完成的。

（位号指定在十六进制中完成）。

例如 :D0 的位 5(b5) 指定为 D0.5，D0 的位 10(b10) 指定为 D0.A。

然而，对于定时器 (T)，累计定时器 (ST)，计数器 (C) 或变址寄存器 (Z) 而言，可能就没有位指定（如 Z0.0 就不存在）。



3.2.2 使用字（16 位）数据

字数据是基本指令和应用指令使用的 16 位数据。

以下两种形式的字数据可以在 CPU 模块中使用：

- 十进制常数 ..... K-32768 到 K32767
- 十六进制常数 ..... H0000 到 HFFFF

字软元件和进行了位数指定的位软元件可以作为字数据使用。

对于直接访问输入 (DX) 和直接访问输出 (DY)，字数据不能通过位数指定。（关于直接访问输入和直接访问输出，请参考以下手册。）

- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础篇）

(1) 当使用位软元件时

(a) 当数位被指定后，位软元件就可以处理字数据。

位软元件的数位指定是通过指定 “ 位号 位软元件的初始号 ” 来完成的。
位软元件的数位指定可以在 4 点 (4 位）单元中完成，并且可以指定为 K1 到 K4。（对于链接直接软元件，指定是通过 “J 网络号 \ 数位指定 位软元件的初始号 ” 完成的。如 X100 到 X10F 用于指定网络号 No.2，它是通过 J2\K4X100 来完成的）。
例如，如果 X0 被用于数位指定，下列各点需要指定：

- K1X0从 X0 到 X3 的 4 点被指定
- K2X0从 X0 到 X7 的 8 点被指定
- K3X0从 X0 到 XB 的 12 点被指定
- K4X0从 X0 到 XF 的 16 点被指定

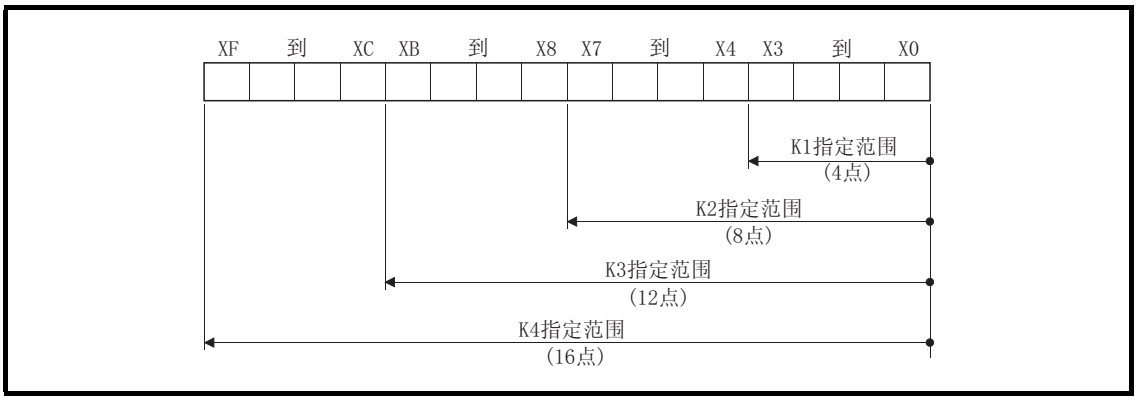


图 3.1 16 位指令的数位指定设定范围

(b) 如果在源数据 S 中数位指定已经完成，在表 3.1 表 3.1 表 3.1 中的所显示的数值就可以当作源数据进行处理。

表 3.1 当作数位指定处理的数值表

指定的数位号	通过 16 位指令
K1 (4 点)	0 到 15
K2 (8 点)	0 到 255
K3 (12 点)	0 到 4095
K4 (16 点)	-32768 到 32767

(c) 如果在源位置是由数位指定来指定的位软元件，并且目标是字软元件，那么从由源数据中的数位指定来指定的位之后，目标数据中的字软元件为 0。

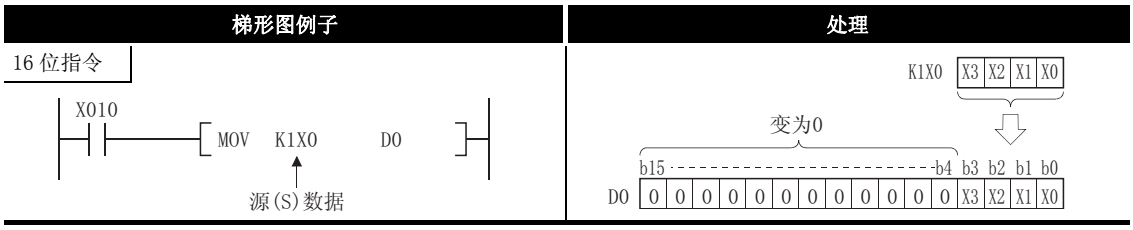


图 3.2 梯形图例子和执行的处理

(d) 一旦在目标 D 中完成数位指定，那么指定的点数作为目标使用。
在指定为数位的点数之后的位软元件不变。

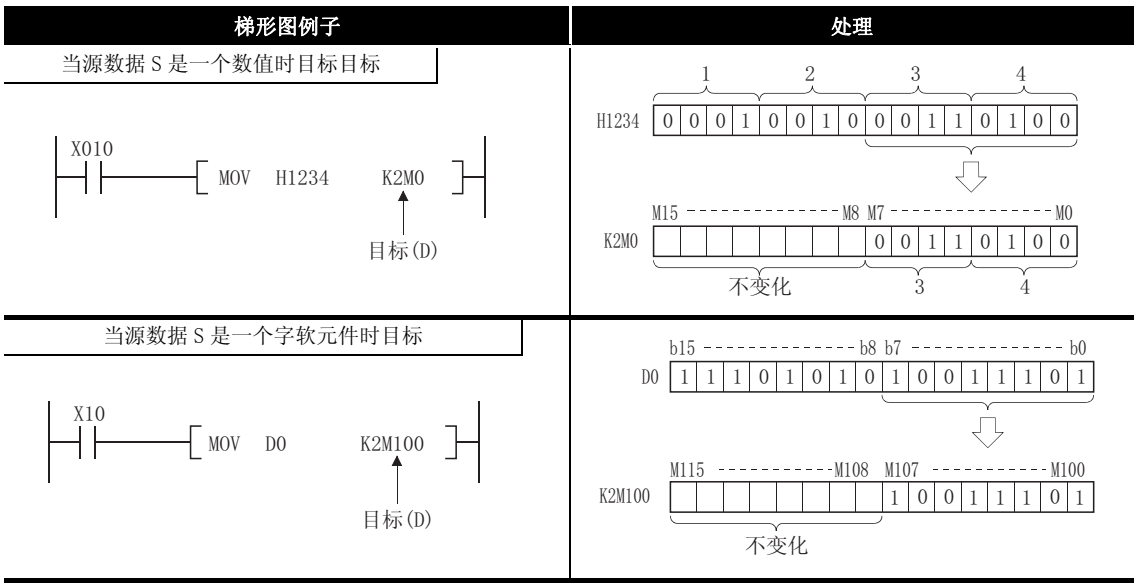
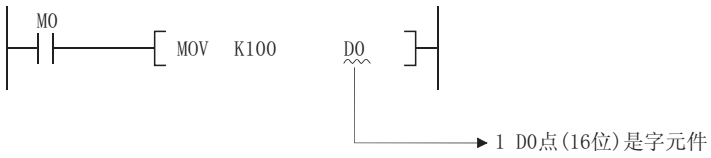


图 3.3 梯形图例子和进行的处理

(2) 当使用字软元件时

字软元件在 1 点单位 (16 位) 中被指定。



要 点

1. 进行数位指定处理时，位软元件初始软元件号可以为任意数值。
2. 数位指定不可用于直接访问 I/O(DX,DY)。

3.2.3 使用双字数据 (32 位)

双字数据是被基本指令和应用指令使用的 32 位数值数据。

两种类型的双字数据可以被 CPU 模块处理，如下所示：

- 十进制常数 ..... K-2147483648 到 K2147483647
- 十六进制常数 ..... H00000000 到 HFFFFFFFF

由数位指定来指定的字软元件和位软元件可以当作双字数据使用。

对于直接访问输入 (DX) 和直接访问输出 (DY)，双字数据的指定是不可能通过数位指定来完成的。

(1) 使用位软元件时

(a) 数位指定可以使位软元件处理双字数据。

位软元件的数位指定是通过指定 “ 数位号 和 初始位软元件号 ” 来完成的。位软元件的数位指定可以在 4 点 (4 位) 单元中完成，可用于 K1 到 K8 的指定。(对于链接直接软元件，指定是通过 “J 网络号 \ 数位 指定初始位软元件号 ” 来完成的。当 X100 到 X11F 为网络号 2 指定时，是通过 J2\K8X100 来完成的)。

例如，如果 X0 被指定用于数位指定，下列各点就可以被指定：

- K1X0 ..... X0 到 X3 的 4 点被指定
- K2X0 ..... X0 到 X7 的 8 点被指定
- K3X0 ..... X0 到 XB 的 12 点被指定
- K4X0 ..... X0 到 XF 的 16 点被指定
- K5X0..... X0 到 X13 的 20 点被指定
- K6X0..... X0 到 X17 的 24 点被指定
- K7X0..... X0 到 X1B 的 28 点被指定
- K8X0..... X0 到 X1F 的 32 点被指定

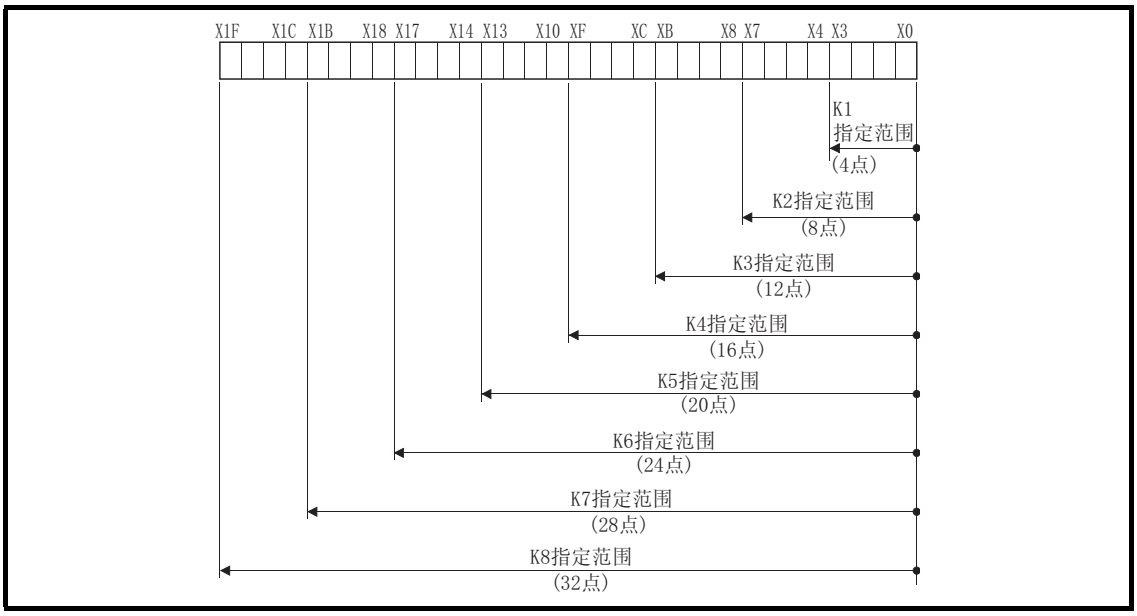


图 3.4 32 位指令位指定设定范围

(b) 如果在源数据 S 位置数位指定已经完成，在表 3.2 中所显示的数值就可以当作源数据来处理。

表 3.2 可以作为数位指定处理的数值列表

指定的数位号	对 32 位指令	指定的数位号	对 32 位指令
K1 (4 点)	0 到 15	K5 (20 点)	0 到 1048575
K2 (8 点)	0 到 255	K6 (24 点)	0 到 16777215
K3 (12 点)	0 到 4095	K7 (28 点)	0 到 268435455
K4 (16 点)	0 到 65535	K8 (32 点)	−2147483648 到 2147483647

(c) 如果在源位置是由数位指定的一位软元件，并且目标是字软元件，那么在目标数据中，由源数据中数位指定所指定的位以后，目标数据的字软元件变为 0。

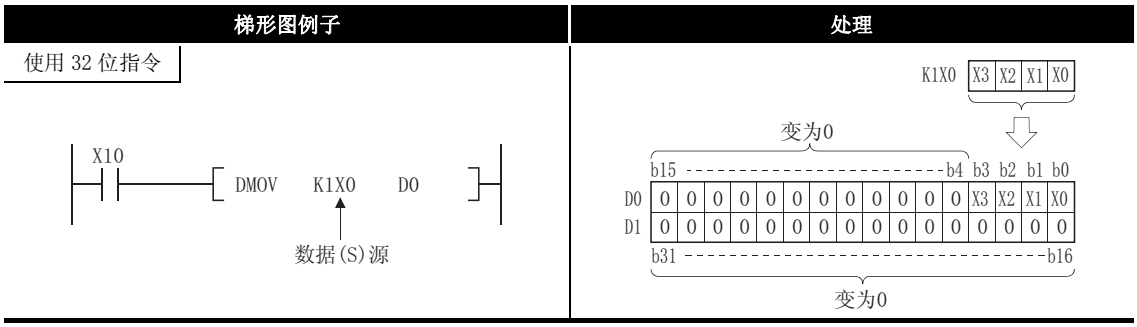


图 3.5 梯形图例子和进行的处理

(d) 一旦在目标 D 的位置完成了数位指定，指定的点数可以当作目标来使用。
点数指定为数位以后，位软元件不能变更。

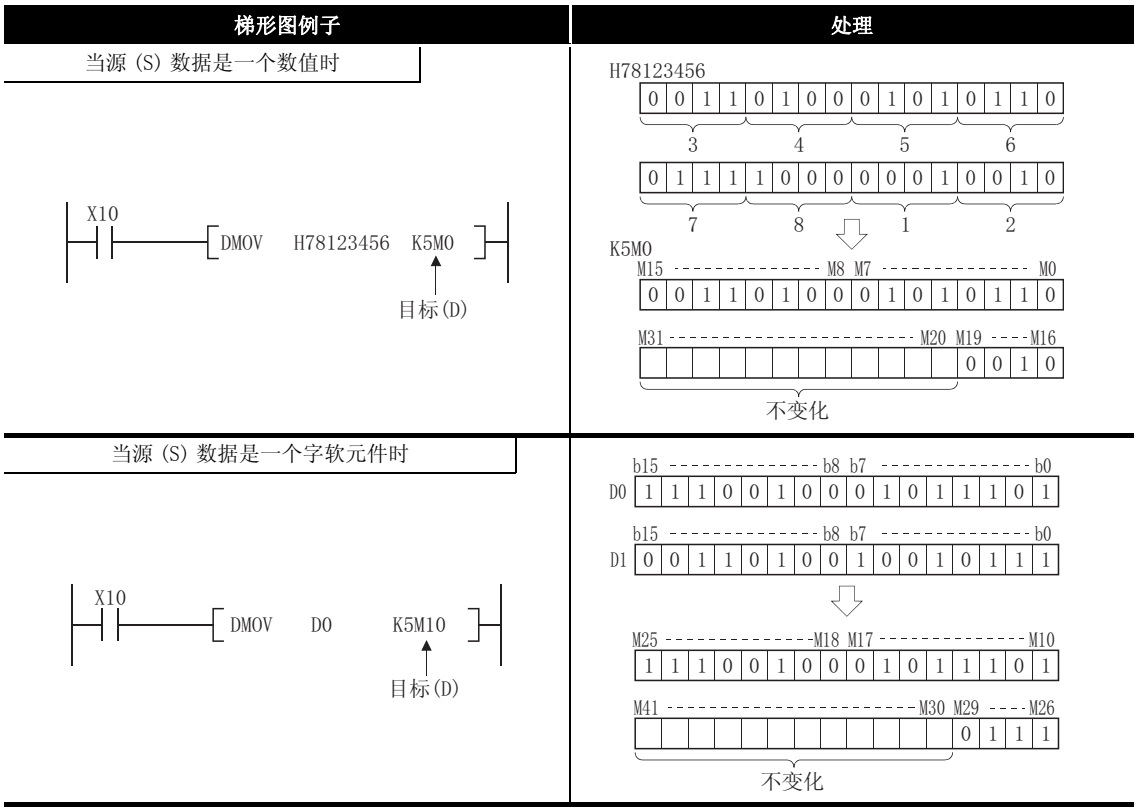


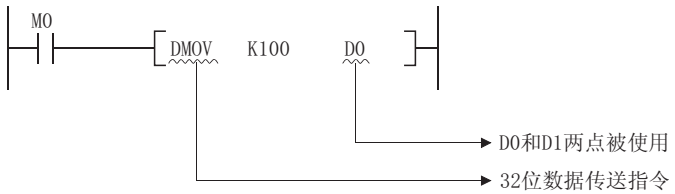
图 3.6 梯形图例子和进行的处理

要 点

1. 进行位指定处理时，位软元件初始软元件号是一个随机的值。
2. 数位指定不可用于直接访问 I/O(DX,DY)。

(2) 当使用字软元件时

字软元件可指定被数据的低 16 位使用的软元件。
一个 32 位指令使用（指定软元件号）和（指定软元件号 +1)。



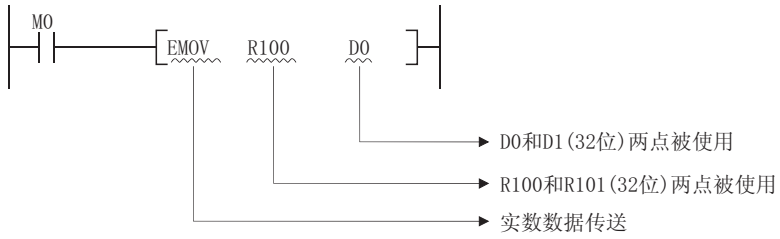
3.2.4 使用实数数据

实数数据是用于基本指令和应用指令的浮点数据。

只有字软元件能够存储实数数据。

(1) 单精度浮点数据

处理单精度浮点数据的指令指定低位 16 位中使用的软元件。
单精度浮点数据存储在（指定的软元件号）和（指定的软元件号 +1) 的 32 位中。



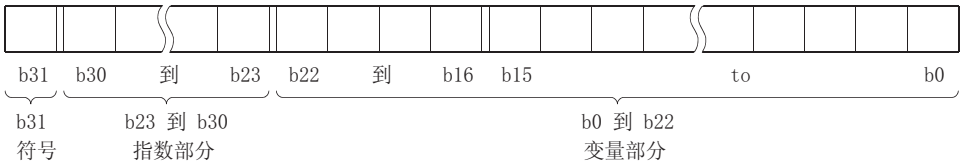
备 注

在顺控程序中，实数通过 E□指定。

浮点小数数据使用两个字软元件并以下列方式表达：

［符号］1.［变量部分］×2［指数部分］

位配置和浮点小数数据的内部代表意义如下：



- 符号　符号在 b31 处被表示
  0：正
  1：负

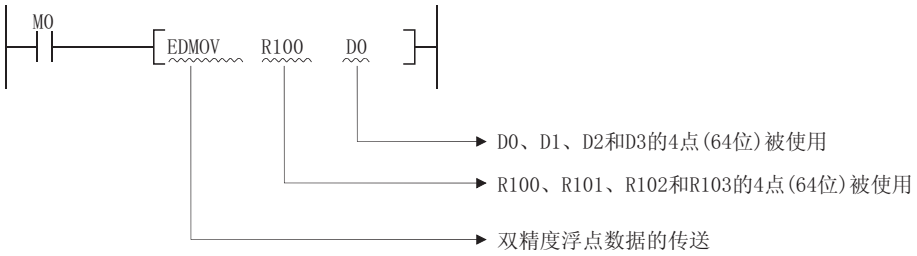
- 指数部分 2n 中的 n 是由 b23 到 b30 表示
  根据 b23 到 b30 的 BIN 值的不同，n 的值如下所示：

b23 到 b30	FFH	FEH	FDH		81	80	7FH	7EH		02	01	00
n	不使用	127	126		2	1	0	-1		-125	-126	不使用

• 变量部分　从 b0 到 b22 的 23 位，在二进制位置 1.XXXXXX.. 代表 ..XXXXXX...

(2) 双精度浮点数据

处理双精度浮点数据的指令指定低位 16 位中使用的软元件。
双精度浮点数据存储在（指定的软元件号）～（指定的软元件号 +3）的 64 位中。



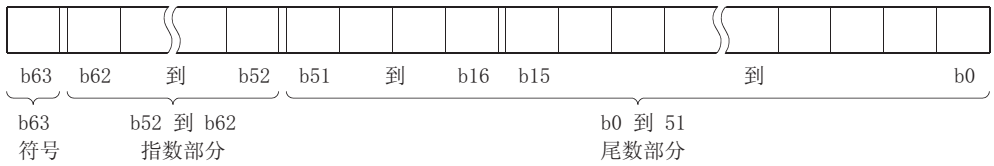
备 注

在顺控程序中，通过 E□指定浮点数据。

双精度浮点数据使用 4 个字软元件并以下列方式表达：

[符号] 1. [尾数部分] ×2 [指数部分]

双精度浮点数据内部表示时的位构成及其含义如下：



• 符号　符号在 b63 处被表示。
0：正
1：负

• 指数部分　2n 中的 n 是由 b52 到 b62 表示。
根据 b52 到 b62 的 BIN 值的不同，n 的值如下所示：

b52 到 b62	7FFH	7FEH	7FDH		400H	3FFH	3FEH	3FDH	3FCH		02H	01H	00H
n	不使用	1023	1022		2	1	0	−1	−2		−1021	−1022	不使用

• 尾数部分　在从 b0 到 b51 的 52 位中，.XXXXXX... 的值表示二进制的 1.XXXXXX..。

要 点

1. 通过外围设备的监视功能可以监视 CPU 模块的浮点数据。
2. 当浮点数据被用于表达 0 时，以下范围内的所有的数据均变为 0。
   (a) 单精度浮点数据：b0 到 b31
   (b) 双精度浮点数据：b0 到 b63
3. 浮点数据的设置范围如下。*1
   (a) 单精度浮点数据时
   $-2^{128} < $ 软元件数据 $\leqq -2^{-126}$, 0, $2^{-126} \leqq$ 软元件数据 $< 2^{128}$
   (b) 双精度浮点数据时
   $-2^{1024} < $ 软元件数据 $\leqq -2^{-1022}, 0, 2^{-1022} \leqq$ 软元件数据 $< 2^{1024}$
4. 不要在浮点数据中指定 -0（只有浮点型实数的最高位为 1 时）。（如果以 -0 进行浮点运算，将发生运算错误）。
   在以双精度进行浮点运算的内部运算的 CPU 模块中指定了 -0 时，在 CPU 模块内部将会把 -0 转换为 0 后再执行浮点运算，因此不会发生运算错误。
   在以单精度进行浮点运算的内部运算的 CPU 模块中指定了 -0 时，因为处理速度优先，在浮动点运算时原样不变地使用 -0，因此将发生运算错误。
   (a) 当 -0 被指定时，下列 CPU 模块中不会发生运算错误。
   - 内部运算设定为双精度的高性能型 QCPU *2
     （浮点运算的内部运算默认为双精度。）
   - QnACPU
   - Q2ASCPU
   - SM707 为 ON 的 Q4ARCPU

   (b) 当 -0 被指定时，下列 CPU 模块中将发生运算错误。
   - 基本型 QCPU *3
   - 内部运算设置成单精度的高性能型 QCPU *2
   - 过程 CPU
   - 冗余 CPU
   - 通用型 QCPU
   - SM707 为 OFF 的 Q4ARCPU

*1: 关于实数超出范围和输入的值无效时的运行情况，请参见以下手册：
- QCPU 用户手册（功能解说 / 程序基础篇）。

*2: 浮点运算的内部运算的单精度和双精度之间的转换是在可编程控制器参数的可编程控制器系统设定中进行。
关于浮点运算的单精度和双精度，请参见以下手册：
- QCPU 用户手册（功能解说 / 程序基础篇）。

*3: 在序列号的前五位为 “04122 或更大” 的基本型 QCPU 中，能够执行浮点运算。

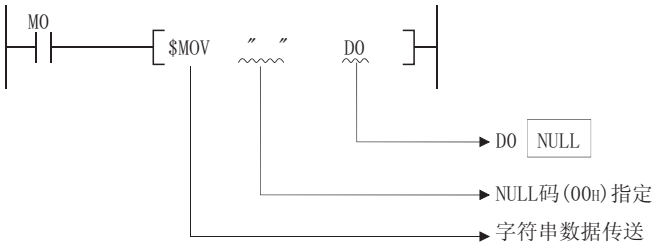
3.2.5 使用字符串数据

字符串数据是基本指令和应用指令使用的字符数据。

它包含从指定字符到 NULL 码 (00H) 的所有数据。

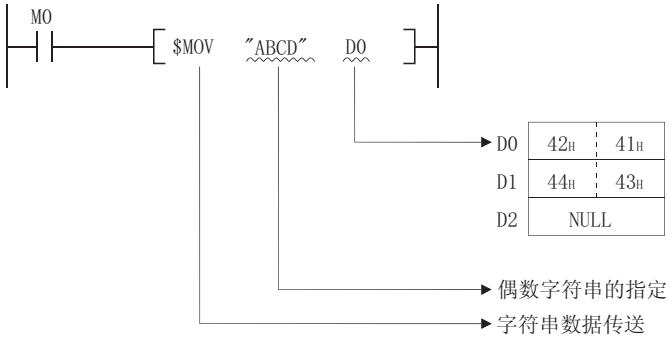
(1) 当指定字符是 NULL 码时

使用一个字来存储 NULL 码。



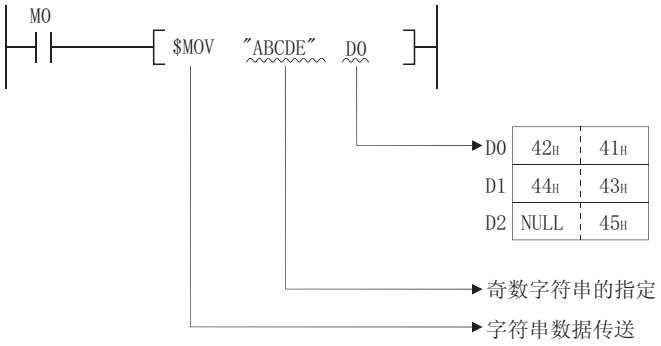
(2) 当字符数是偶数时

使用 (字符数 /2+1) 字，并且存储字符串和 NULL 码。
例如，如果 “ABCD” 被转换成 D0，那么 ABCD 就存储在 D0 和 D1 处，NULL 码存储在 D2 处。



(3) 当字符数是奇数时

使用 (字符数 /2) 字 (小数部分四舍五入)，并且存储字符串和 NULL 码。
例如，如果 “ABCD” 传送到 “D0”，字符串 (ABCDE) 和 NULL 码将会从 D0 存储到 D2 上去。
(NULL 码将被存储到最后 1 个字的高 8 位处。)



3.3 变址修饰

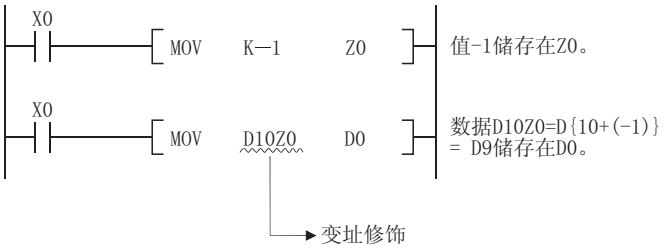
(1) 变址修饰的概要

(a) 变址修饰是通过使用变址寄存器进行的间接设定。
当在顺控程序中使用变址修饰时，使用的软元件将变为 （直接指定的软元件号）+（变址寄存器的内容）。
例如，如果指定了 D2Z2，则指定的软元件计算如下：
Z2 的内容为 3 时，D(2+3)=D5，D5 变为指定的软元件。

(b) 只有在使用通用型 QCPU 时，才可以使用 16 位变址寄存器和 32 位变址寄存器进行变址修饰。

(2) 通过 16 位变址寄存器进行的变址修饰

(a) 变址修饰示例
每一个变址寄存器都能在 -32768 到 32767 之间设定。
变址修饰按照以下方式执行：



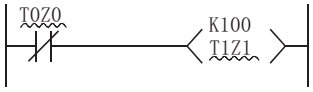
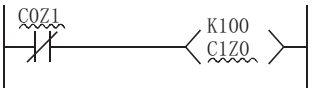
(b) 可以进行变址修饰的软元件
除去下面所列的限制，变址修饰可以应用于触点、线圈、基本指令和应用指令中使用的软元件。

1）不能使用变址修饰的软元件

软元件	含义
K、H	32 位常数
E	浮点数据
$	字符串数据
□.□	字软元件位指定
FX、FY、FD	功能软元件
P	作为标签的指针
I	作为标签的中断指针
Z	变址寄存器
S	步进继电器
TR	SFC 传送软元件 *1
BL	SFC 块软元件 *1

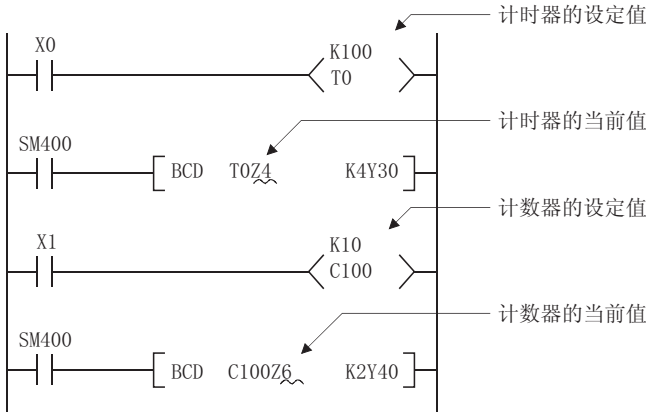
*1: SFC 传送软元件和 SFC 块软元件是供 SFC 使用的软元件。
关于如何使用这些软元件，请参考：
• QCPU/QnACPU 编程手册（SFC 控制指令篇）

2）和变址寄存器一起使用的软元件的限制

软元件	含义	应用例
T	• 只有 Z0 和 Z1 可以用作计时器的触点和线圈。	
C	• 只有 Z0 和 Z1 可以用作计数器的触点和线圈。	

备 注

对于计时器和计数器的当前值，没有变址寄存器使用编号的限制。



(c) 进行了变址修饰时与实际的处理软元件如下所示：

（当 Z0 = 20 和 Z1= −5 时）

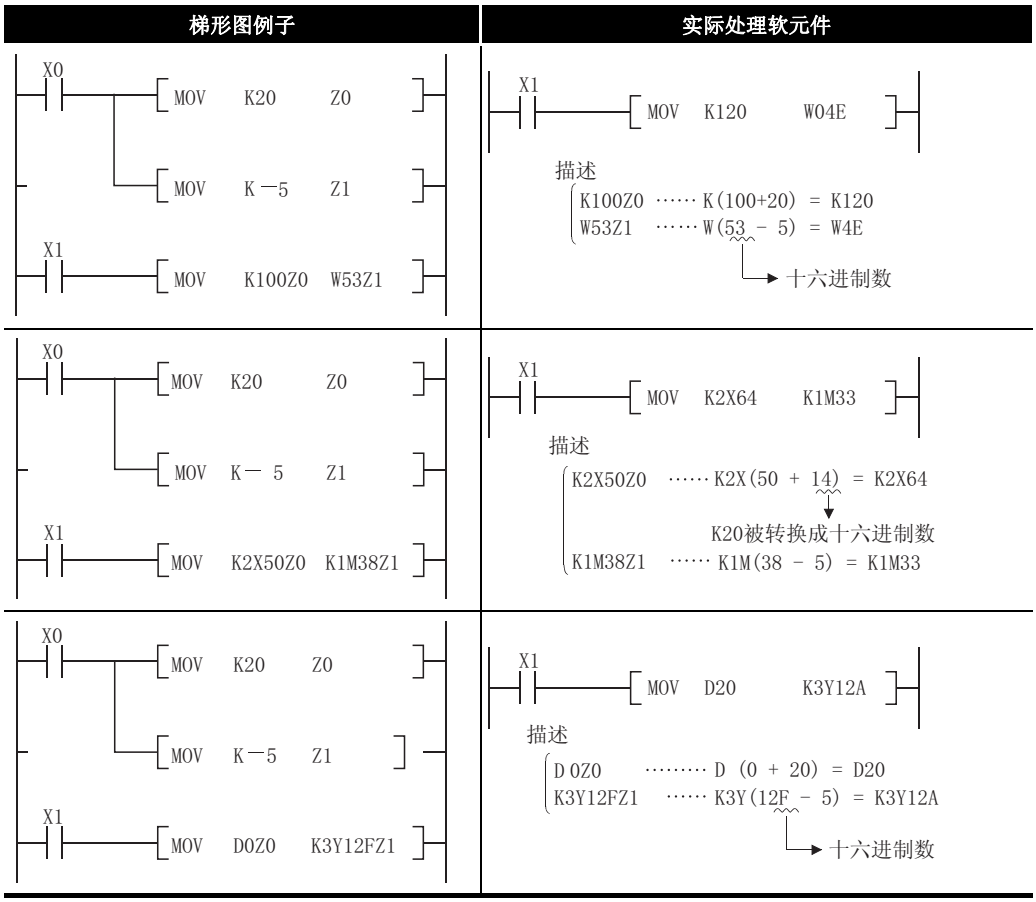


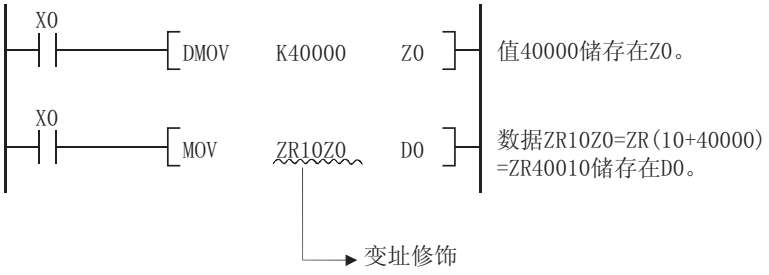
图 3.1 梯形图示例及实际的处理软元件

(3) 通过 32 位变址寄存器进行的变址修饰（只对于通用型 QCPU）

(a) 变址修饰示例

每一个变址寄存器都能在 -2147483648 到 2147483647 之间设定。

变址修饰按照以下方式执行：



(b) 变址修饰的设置方法

通过 32 位变址寄存器进行变址修饰时，在 GX Developer 的 Q 参数设置画面的软元件选项卡中指定使用的变址寄存器的起始号。

要 点

当在 Q 参数设置画面的软元件选项卡中将使用的变址寄存器的起始号进行了更改时，不要只修改参数或者进行可编程控制器写入。必须是与程序一道进行可编程控制器写入。

当参数强制性的写入可编程控制器时，将会发生 CAN'T EXE. PRG 错误。（出错代码 :2500）

(c) 可进行变址修饰的软元件

变址修饰只能用于以下软元件。

软元件	含义
ZR	连号访问型文件寄存器

(d) 变址寄存器的可用范围

下表列出了通过 32 位变址寄存器进行变址修饰时的变址寄存器可用范围。

在通过 32 位变址寄存器进行变址修饰时，使用指定的变址寄存器 (Zn) 和紧接着的下一个变址寄存器 (Zn+1)，因此请注意防止所使用的变址寄存器重叠。

设置值	使用的变址寄存器	设置值	使用的变址寄存器
Z0	Z0、Z1	Z10	Z10、Z11
Z1	Z1、Z2	Z11	Z11、Z12
Z2	Z2、Z3	Z12	Z12、Z13
Z3	Z3、Z4	Z13	Z13、Z14
Z4	Z4、Z5	Z14	Z14、Z15
Z5	Z5、Z6	Z15	Z15、Z16
Z6	Z6、Z7	Z16	Z16、Z17
Z7	Z7、Z8	Z17	Z17、Z18
Z8	Z8、Z9	Z18	Z18、Z19
Z9	Z9、Z10	Z19	不能指定

(e) 进行了变址修饰时与实际的处理软元件如下所示：

（当 Z0 = 20000 和 Z2= −5 时）

梯形图例子	实际处理软元件
X0 DMOV K20000 Z0 MOV K-20 Z2 X1 MOV ZR1000Z0 D30Z2	X1 MOV ZR21000 D10 描述 ZR1000Z0…ZR(1000+20000)=ZR21000 D30Z2……D(30-20)=D10

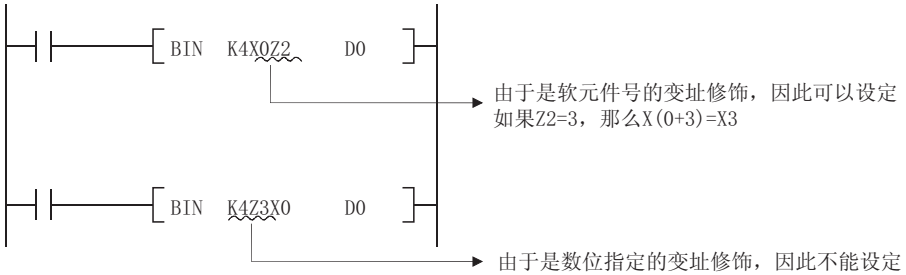
图 3.2 梯形图示例和实际处理软元件

(4) 其他

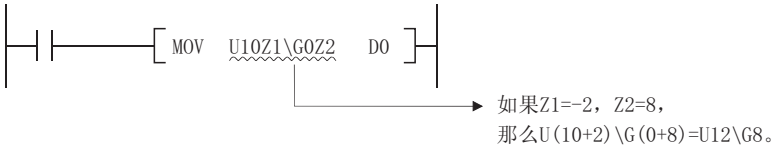
(a) 位数据

在执行数位指定时，可以进行软元件编号的变址修饰。

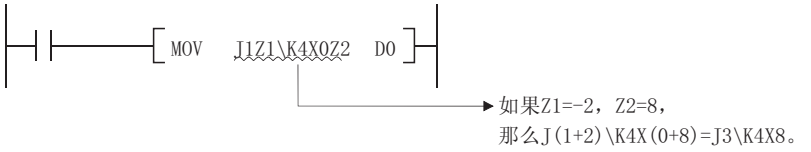
然而，不能进行位数指定的变址修饰。



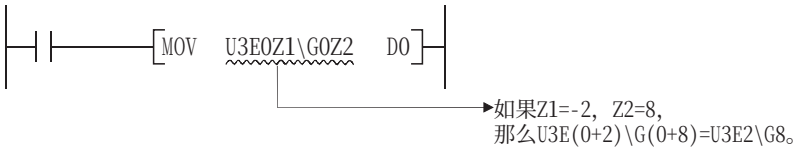
(b) 在智能功能模块软元件 *1 中，对智能功能模块的起始 I/O 地址号和缓冲存储器地址号均可以进行变址修饰。



(c) 在链接直接软元件 *1 中，对网络号和软元件号均可以进行变址修饰。



(d) 在多 CPU 共享软元件 *2 中，对 CPU 模块的起始 I/O 地址号和 CPU 共享存储器地址均可以进行变址修饰。



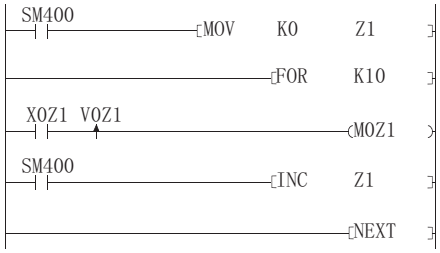
*1: 关于智能功能模块软元件和链接直接软元件，请参考以下手册：
- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础篇）

*2: 关于多 CPU 共享软元件，请参考以下手册：
- QCPU 用户手册（功能解说 / 程序基础篇）。

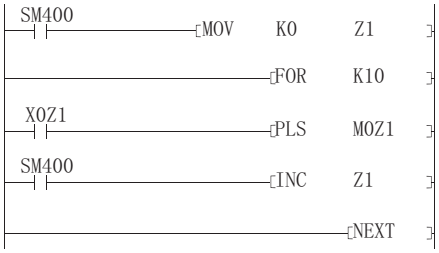
(5) 注意事项

(a) 在 FOR ～ NEXT 指令之间进行变址修饰时
通过在 FOR ～ NEXT 指令之间使用变址继电器 (V)，可以进行脉冲输出。
然而，不可以进行使用 PLS/PLF/ 脉冲（□ P) 指令的脉冲输出。

[使用变址继电器时]
(M0Z1 可以进行正常的脉冲输出。)



[未使用变址继电器时]
(M0Z1 不能进行正常的脉冲输出。)



备 注

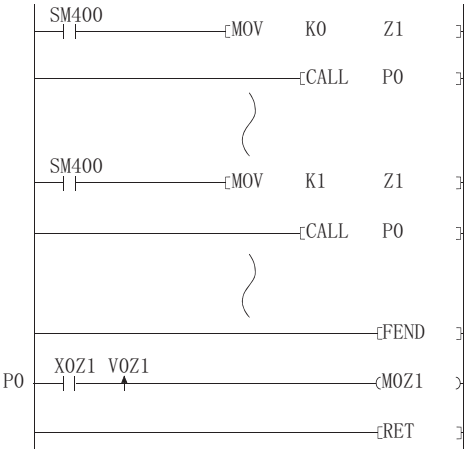
ON/OFF 数据由变址继电器 V0Z1 存储。
例如，X0 的 ON/OFF 数据由 V0 存储，X1 的数据由 V1 存储。

(b) 用 CALL 指令执行变址修饰

通过使用变址继电器 (V)，脉冲可以用 CALL 指令输出。然而，不可以进行使用 PLS/PLF/脉冲 (□ P) 指令的脉冲输出。

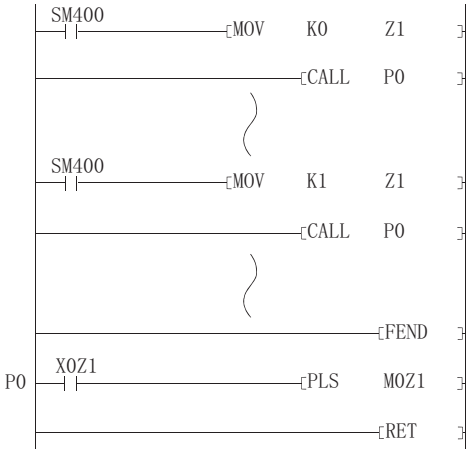
[使用变址继电器时]

(MOZ1 提供正常的脉冲输出。)



[不使用变址继电器时]

(MOZ1 不提供正常的脉冲输出。)



(c) 变址修饰时的软元件范围检查

1) 除通用型 QCPU 以外的 CPU

在变址修饰时执行时，软元件范围检查也会进行。

因此，在经过变址修饰之后的数据超出了用户所指定的软元件将围时，为了避免发生错误，这些数据将会写入到其他软元件中。(注意，如果经过变址修饰后的数据写入到用于系统的软元件中时，超出了用户所指定的软元件范围，则会出现错误。(出错代码 :1103))

在进行编程的过程中使用变址修饰时，请特别注意。

2) 通用型 QCPU

当执行了变址修饰时，软元件范围检查会也进行。

通过更改GX Developer的可编程控制器参数设置，不会进行软元件范围检查。

(d) 将通过 16 位变址寄存器进行变址修饰更改为通过 32 位变址寄存器进行变址修饰

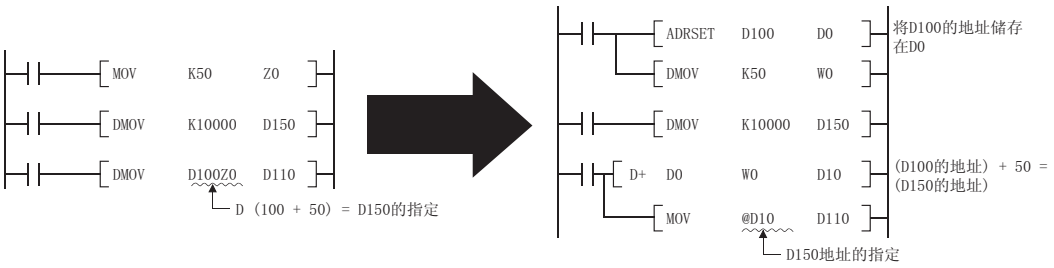
对于通过 16 位变址寄存器进行变址修饰更改为通过 32 位变址寄存器进行变址修饰，请确认程序是否有足够的空间进行变址修饰。

对于通过 32 位变址寄存器进行变址修饰，使用了指定的变址寄存器 (Zn) 和指定的变址寄存器 (Zn+1) 的下一个变址寄存器。请确认不要与将要使用的变址寄存器重叠。

3.4 间接指定

(1) 间接指定

(a) 间接指定是一种通过使用字软元件来指定软元件地址的方法，这种方法会在使用两个字软元件顺控程序（字软元件的 2 点）中使用。在变址寄存器不足的情况下，可以使用这种方法。



［使用变址寄存器时］　　［使用间接指定时］

(b) 用于指定指定软元件地址的软元件是通过 “@ +(字软元件号)”来指定的。例如，@D100 的指定将会使得 D100、D101 的内容成为软元件地址。

(c) 执行间接指定的软元件地址可以通过 ADRSET 指令来进行确认。
关于 ADRSET 指令，请参见 7.18.6 项。

(2) 能间接指定的软元件

能间接指定的 CPU 模块软元件在表 3.1 中列出：

表 3.1 能间接指定的软元件列表

软元件类型		能否被间接指定	间接指定范例
内部用户软元件	位软元件 *1	否	-----
	字软元件 *1	能	• @D100 • @D100Z2 *2
链接直接软元件	位软元件 *1	否	-----
	字软元件 *1	能 *3	• @J1\W10 • @J1Z1\W10Z2 *2
智能功能模块软元件		能 *3	• @U10\G0 • @U10Z1\G0Z2 *2
变址寄存器		否	-----
文件寄存器		能	• @R0、@ZR20000 • @R0Z1、@ZR20000Z1 *2
嵌套结构		否	-----
指针			-----
常数			-----
其他	SFC 块软元件		-----
	SFC 之后的软元件		
	指定软元件的网络号		
	指定软元件的 I/O 号		

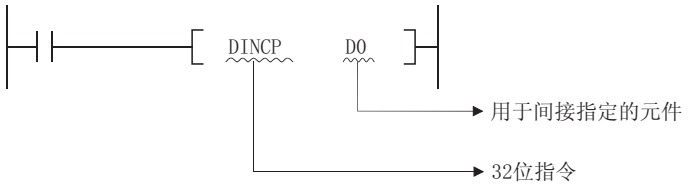
*1: 关于软元件名称，请参考使用的：
- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程参考手册（基础篇）

*2: 表示通过变址寄存器进行变址修饰。
*3: 软元件能够间接指定，然而地址不能写入 ASRSET 指令中。

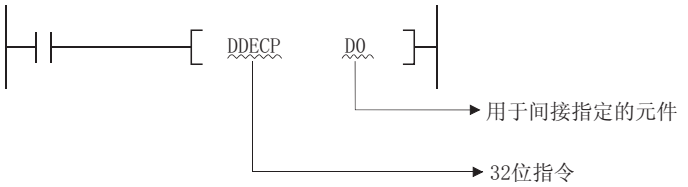
(3) 注意事项

间接指定的地址需要使用两个字来指定。
因此，用间接指定来替换变址修饰，要求使用 32 位数据的加减运算。
以下是在间接指定中，为存储在 D1 和 D0 中的软元件地址的加减而使用的梯形图。

[在间接指定中，增加 “1” 到软元件的地址中]



[在间接指定中，从软元件地址中减少 “1”]



3.5 缩短指令处理时间

3.5.1 子集处理

子集处理是用来对基本指令和应用指令中使用的位软元件设定限制的，目的是为了加快处理速度。

但是，指令符号不变。

为了缩短扫描周期，请在下面所列出的条件下运行指令。

(1) 子集处理中各软元件必须满足的条件

(a) 当使用字数据时

软元件	条件
位软元件	• 以 16 的倍数指定位软元件号。 • 数位指定只能指定为 K4。 • 不能进行变址修饰。
字软元件	• 内部软元件。 • 文件寄存器 (R、ZR*1) • 多 CPU 共享软元件 *1、*2 • 变址寄存器 (Z)/ 通用运算寄存器 (Z)*1
常数	• 没有限制

(b) 当使用双字数据时

软元件	条件
位软元件	• 以 16 的倍数指定位软元件号 • 数位指定中只能指定为 K8 • 不能进行变址修饰
字软元件	• 内部软元件 • 文件寄存器 (R、ZR*1) • 多 CPU 共享软元件 *1、*2 • 变址寄存器 (Z)/ 通用运算寄存器 (Z)*1
常数	• 没有限制

(c) 当使用位数据时

软元件	条件
位软元件	• 内部用户软元件（可以进行变址修饰）
字软元件	• 内部用户软元件的位指定 • 寄存器 (R、ZR*1) 的位指定 • 多 CPU 共享软元件的位指定 *1、*2

*1: 只对应于通用型 QCPU。

*2: 只对于多 CPU 高速通信区 (U3En\G10000 ～) 才有效。
（对 CPU 模块的起始 I/O 地址号执行了变址修饰 (U3En\G10000) 时除外。）

(2) 可进行子集处理的指令

指令类型	指令符号
触点指令	LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF
输出指令	OUT、SET、RST
比较运算指令	• =、<>、<、<=、>、>=、D=、D<>、D<、D<=、D>、D>=
算术运算	• +、－、*、/、INC、DEC、D+、D－、D*、D/、DINC、DDEC • B+、B－、B*、B/、E+、E－、E*、E/
数据变换指令	• BCD、BIN、DBCD、DBIN、FLT、DFLT、INT、DINT
数据传送指令	• MOV、DMOV、CML、DCML、XCH、DXCH • FMOV、BMOV、EMOV (with QCPU only)
程序分支指令	• CJ、SCJ、JMP
逻辑运算 *3	• WAND、DAND、WOR、DOR、WXOR、DXOR、WXNR、DXNR
旋转指令	• RCL、DRCL、RCR、DRCR、ROL、DROL、ROR、DROR
移动指令	• SFL、DSFL、SFR、DSFR
数据处理指令	• SUM、SEG
结构化指令	• FOR、CALL

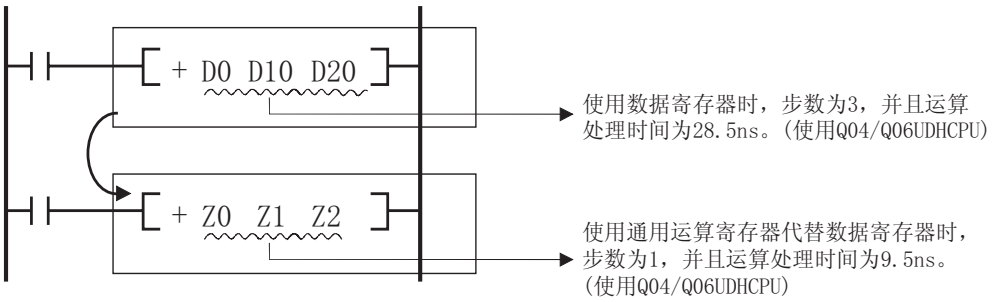
*3: 只有 QCPU 才能进行三个软元件的逻辑运算指令 WAND、DAND、WOR、DOR、WXOR、DXOR、WXNR、DXNR 的子集处理。



3.5.2 使用通用运算寄存器 (Z) 的运算处理 (只对于通用型 QCPU)

使用通用运算寄存器 (Z) 可以缩短运算处理时间。

下面是使用通用运算寄存器的程序示例。



通过可以进行子集处理的指令，可以缩短运算处理时间。

关于步数的变更，请参见 3.8 节。

关于各指令的运算时间，请参见附录 1。

要 点

因为通用运算寄存器与变址寄存器是同一种软元件，所以在进行变址修饰时，不要使通用运算寄存器的软元件号与变址寄存器的软元件号重复。

3.6 编程注意事项（运算错误）

当在 CPU 模块中执行基本指令和应用指令时，在下列情况下将会发生运算错误：

- 发生了各指令的说明页面中记载的出错时。
- 使用智能功能模块软元件时，智能功能模块没有安装在指定的 I/O 号位置。
- 使用智能功能模块软元件时，指定的缓冲存储器地址不存在。
- 使用链接软元件时，相应的网络不存在。
- 使用链接软元件时，在指定的 I/O 号位置上没有安装网络模块。
- 使用多 CPU 共享软元件时，CPU 模块没有安装在指定 CPU 模块的起始 I/O 号位置。
- 使用多 CPU 共享软元件时，指定的共享存储器地址不存在。

要 点

当进行了文件寄存器设定但未安装内存卡时，或在未进行文件寄存器设定的状态下对文件寄存器进行了的读 / 写操作时的情况如下所示：

(1) 对于 QnACPU

即使对文件寄存器执行读 / 写操作，也不会发生错误。然而，如果从文件寄存器中读取，“FFFFH”将会被存储。

(2) 对于高性能型 QCPU、过程 CPU 和冗余 CPU

即使对文件寄存器执行读 / 写操作，也不会发生错误。然而，如果从文件寄存器中读取，“0H”将会被存储。

(3) 对于通用型 QCPU

如果对文件寄存器执行读 / 写操作，将会发生 OPERATION ERROR(出错代码 :4101)。

(1) 软元件范围检查

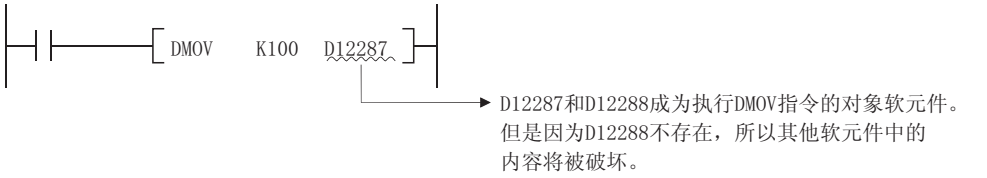
在 CPU 模块中，基本指令和应用指令中使用的软元件范围检查的情况如下所示：

(a) 用于指定各固定长的软元件 (包括 MOV 和 DMOV) 的指令

1) 对于除通用型 QCPU 以外的 CPU

不进行软元件范围检查。如果相应的软元件范围被超出，数据会被写入其他软元件。 *1

例如，在数据寄存器被分配了 12k 点时，即使超过了 D12287 点，也不会发生错误。



当执行了变址修饰时，也不进行软元件范围检查。

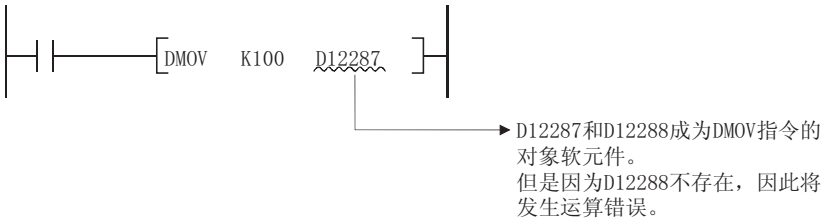
如果执行了变址修饰且超过了相应的软元件范围时，数据会被写入其他软元件。*1

*1: 关于内部软元件的分配顺序，请参见本项 (c) 字符串数据。

2) 对于通用型 QCPU

进行软元件范围检查。当软元件号超出了软元件范围，将会出现错误也会进行。

例如，如果数据寄存器被分配了 12k 点，如果超过了 D12287 点，则会出现错误。



执行了变址修饰时，将进行软元件范围检查。

通过更改 GX Developer 的可编程控制器参数设置，可以设置为不进行软元件范围检查。*2

*2: 关于将设置变更为不进行软元件范围检查的有关内容，请参见以下手册：

• QCPU 用户手册（功能解说 / 程序基础篇）

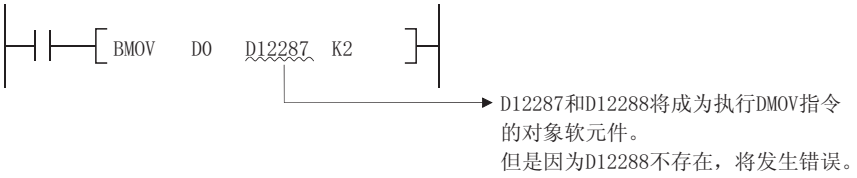
(b) 用于可变长的软元件块（包括指定传送数量的 BMOV 和 FMOV 等）的指令

1) 对于除通用型 QCPU 以外的 CPU

进行软元件范围检查。

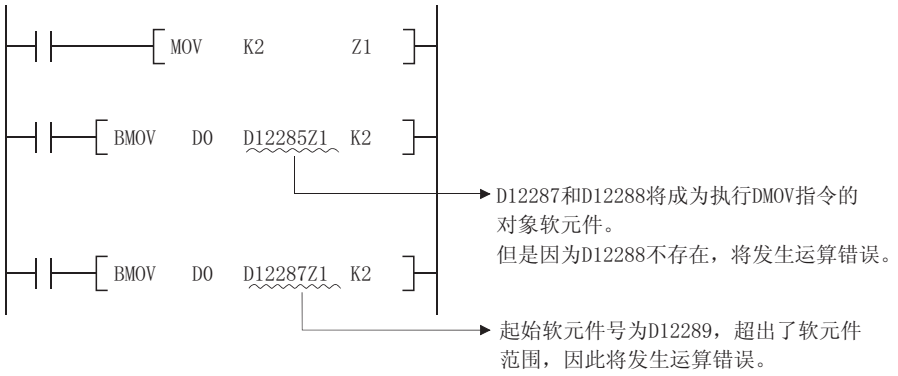
如果相应的软元件范围被超出，将发生运算错误。

例如，如果数据寄存器被分配了 12k 点，如果超过了 D12287 点，将会出现错误。



当执行了变址修饰时，将进行软元件范围检查。

但是，即使由于变址修饰导致起始软元件号超出了相应软元件范围，也不会发生错误。

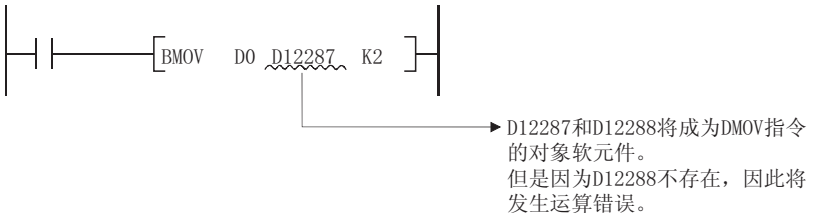


2) 对于通用型 QCPU

进行软元件范围检查。

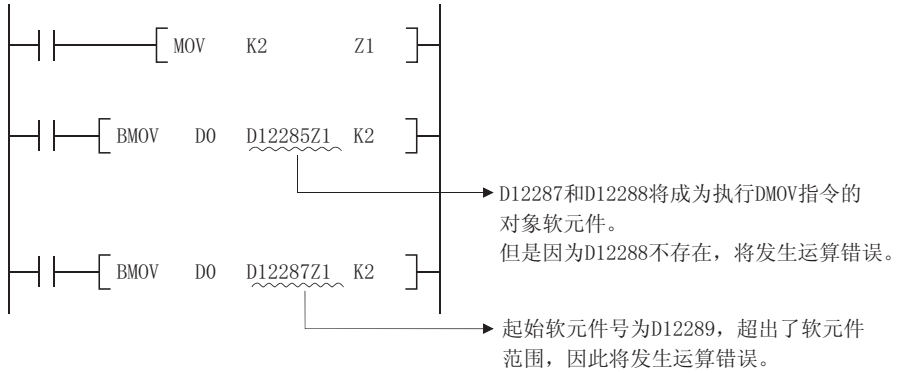
当软元件号超出了软元件范围，将发生运算错误。

例如，当数据寄存器被分配了 12k 点时，如果软元件号超过了 D12287 点，则会发生错误。



当执行了变址修饰时，将进行软元件范围检查。

由于变址修饰的结果导致起始软元件号超出了软元件范围时，将发生错误。



此外，通过更改 GX Developer 的可编程控制器参数设置，可以设置为不进行软元件范围检查。*2

*2: 关于变址修饰时设置为不进行软元件范围检查的有关内容，请参见以下手册：

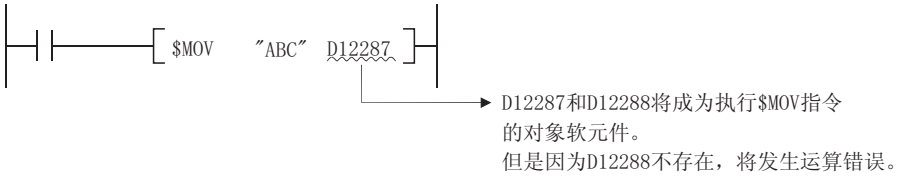
• QCPU 用户手册（功能解说 / 程序基础篇）

(c) 字符串数据

因为所有的字符串数据长度可变，因此将进行软元件的范围检查。

如果超出了相应的软元件范围，将发生运算错误。

例如，数据寄存器被分配了 12k 点时，如果超过了 D12287，将发生错误。



然而，在除通用型 QCPU 以外的 CPU 中，当执行了变址修饰时，在起始软元件号超出了软元件范围的情况下也不会发生错误，而是将访问其它的软元件。

在通用型 QCPU 中进行变址修饰时，如果起始软元件号超出了软元件范围，将会发生错误。

通过更改 GX Developer 的可编程控制器参数设置，可以设置为不进行软元件范围检查。*2

*2: 关于变址修饰时设置为不进行软元件范围检查的有关内容，请参见以下手册：

• QCPU 用户手册（功能解说 / 程序基础篇）

各软元件的分配顺序如下：

SM
SD
X
Y
M
L
B
F
SB
V
S
T触点及线圈
ST触点及线圈
C触点及线圈
T的当前值
ST的当前值
C的当前值
D
W
SW
空区
文件寄存器 (32K点)

要 点

对于通用型 QCPU，不能进行跨过内部用户软元件 (SW) 与文件寄存器 (R) 之间的变址修饰。(否则将发生错误。出错代码 :4101)

备 注

关于如何更改内部用户软元件分配，请参考以下手册：

- QCPU 用户手册 (功能解说 / 程序基础篇)
- QnACPU 编程参考手册 (基础篇)

(d) 如果通过直接的访问输出 (DY) 执行了变址修饰，将进行软元件范围检查。

(2) 软元件数据检查

在 CPU 模块中，对于基本指令和应用指令中使用的软元件的数据检查情况如下所示：

(a) 当使用 BIN 数据时
即使运算结果上溢或下溢，也不会返回出错信息。
进位标志也不会为 ON。

(b) 当使用 BCD 数据时
1) 将对各位进行是否处于 BCD 值 (0 到 9) 的检查。如果某个位超出了 0 到 9 的范围 (A 到 F) 时，将会返回运算出错信息。
2) 即使运算结果上溢或下溢，也不会返回出错信息。进位标志同样也不会为 ON。

(c) 当使用浮点数据时

1) 使用单精度浮点运算指令时，在运算结果为如下所示的情况下，将会返回出错信息。

当浮点数据的绝对值为 1.0×2^{-127} 或更低

当浮点数据的绝对值为 1.0×2^{128} 或更高

2) 使用双精度浮点运算指令时，在运算结果为如下所示的情况下，将会返回出错信息。

当浮点数据的绝对值为 1.0×2^{-1023} 或更低

当浮点数据的绝对值为 1.0×2^{1024} 或更高

(d) 当使用字符串数据时

不进行数据检测。

(3) 至缓冲存储器的访问

对缓冲存储器进行访问时，建议通过使用了智能功能模块软元件 (Un\Go ～) 的指令进行访问。

(4) 至多 CPU 共享存储器的访问

对多 CPU 共享存储器进行访问时，建议通过使用了多 CPU 共享软元件 (U3En\G10000 ～) 的指令进行访问。

3.7 指令执行条件

对于 CPU 模块顺控指令、基本指令和应用指令，存在以下四种类型的执行条件。

- 无条件执行..... 执行的指令与软元件 ON/OFF 状态无关。

  例如 LD X0、 OUT Y10

- 在 ON 的状态下执行.. 在输入条件是 ON 的情况下执行指令。

  例如 MOV 指令、 FROM 指令

- 在上升沿执行..... 指令只在输入条件的上升沿执行（当它从 OFF 转换到 ON 时）。

  例如 PLS 指令，MOVP 指令

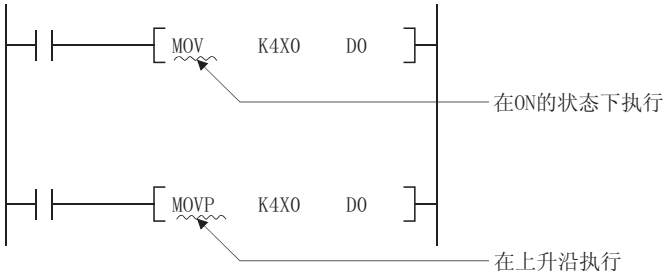
- 在下降沿执行..... 指令只在输入条件的下降沿执行（当它从 ON 转换到 OFF 时）

  例如 PLF 指令

对于线圈或与之同等的基本指令或应用指令，对于同样的指令既可以在 ON 状态下指定，也可以在上升沿执行条件下指定，在指令名称后加上一个 “P”用以说明执行条件。

- 在 ON 状态下执行的指令 指令名
- 在上升沿执行的指令 指令名 + P

对 ON 状态下执行和在上升沿条件下执行的 MOV 指令作如下指定：

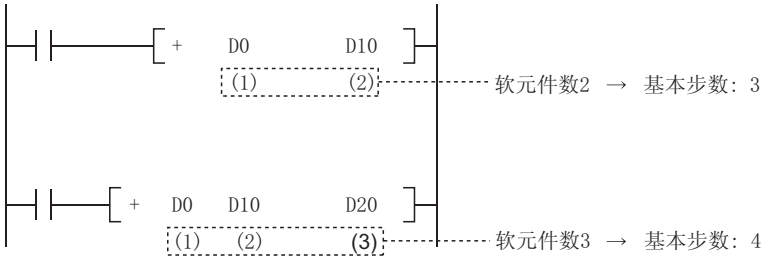


3.8 计算步数

根据使用的软元件、是否进行了间接设定，CPU 模块的顺控指令、基本指令和应用指令的步数均有不同。

(1) 计算基本步数

基本指令和应用指令的基本步数为（软元件数 +1）步。
例如，“+ 指令”应该按照以下方法计算：



(2) 增加步数的条件

当使用软元件的间接指定或增加了步数的软元件时，步数将会超过基本步数。

(a) 当软元件被间接指定时
当通过 @ 进行了间接指定时，步数在基本步数上增加一步。
例如：当一个 3 步的 MOV 指令被间接指定时（例如 MOVK4X0 @D0），将会增加一步而变为四步。

(b) 增加了步数的软元件（通用型 QCPU 除外）

增加了步数的软元件	增加的步数	示例
智能功能模块软元件	1	MOV U4\G10 D0
多 CPU 共享软元件		MOV U3E1\G0 D0
链接直接软元件		MOV J3\B20 D0
变址寄存器		MOV Z0 D0
连号访问方式文件寄存器		MOV ZR123 D0
32 位常数		DMOV K123 D0
实数常数		EMOV E0.1 D0
字符串常数	偶数时：(字符数)/2 奇数时：(字符数 +1)/2	$MOV “123” D0

(c) 增加了步数的软元件（通用型 QCPU）

1) 被进行了子集处理的指令

下表是被进行了子集处理的指令中的软元件的步数。

指令符号	增加了步数的软元件	增加的步数（指令步数）	基本步数
LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF	连号访问方式文件寄存器	1(2)	1
	多 CPU 共享软元件		
SET	连号访问方式文件寄存器	1(2)	1
	多 CPU 共享软元件		
OUT	定时器 / 计数器	3(4)	1
	连号访问方式文件寄存器	1(2)	
	多 CPU 共享软元件		
RST（位软元件）	连号访问方式文件寄存器	1(2)	1
	多 CPU 共享软元件		
RST（字软元件）	定时器 / 计数器（位 / 字软元件）	2(4)	2
	连号访问方式文件寄存器	1(3)	
	多 CPU 共享软元件	1(3)	
LD=、LD<>、LD<、LD<=、LD>、LD>=、AND=、AND<>、AND<、AND<=、AND>、AND>=、OR=、OR<>、OR<.OR<=、OR>、OR>=	通用运算寄存器 *2	-1	3
	连号访问方式文件寄存器	1	
	多 CPU 共享软元件		
LDD=、LDD<>、LDD<、LDD<=、LDD>、LDD>=、ANDD=、ANDD<>、ANDD<、ANDD<=、ANDD>、AND>=、ORD=、ORD<>、ORD<.ORD<=、ORD>、ORD>=	通用运算寄存器 *2	-1	3
	连号访问方式文件寄存器	1	
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数		
+、-、+P、-P、WAND、WOR、WXOR、WXNR、WANDP、WORP、WXORP、WXNRP（2 个软元件）	通用运算寄存器 *2	Ⓓ:-1	3
	连号访问方式文件寄存器	Ⓢ1:1、Ⓓ:3	
	多 CPU 共享软元件		
D+、D-、D+P、D-P、DAND、DOR、DXOR、DXNR、DANDP、DORP、DXORP、DXNRP（2 个软元件）	通用运算寄存器 *2	Ⓓ:-1	3
	连号访问方式文件寄存器	Ⓢ1:1、Ⓓ:3	
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数	Ⓢ1:1	
+、-、+P、-P、WAND、WOR、WXOR、WXNR、WANDP、WORP、WXORP、WXNRP（3 个软元件）*1	连号访问方式文件寄存器	Ⓢ1、Ⓢ2:1、Ⓓ:2	3
	多 CPU 共享软元件		
D+、D-、D+P、D-P、DAND、DOR、DXOR、DXNR、DANDP、DORP、DXORP、DXNRP（3 个软元件）*1	连号访问方式文件寄存器	Ⓢ1、Ⓢ2:1、Ⓓ:2	3
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数	Ⓢ1、Ⓢ2:1	
*、*P、/、/P	连号访问方式文件寄存器	Ⓢ1、Ⓢ2:1、Ⓓ:2	3
	多 CPU 共享软元件		
D*、D*P、D/、D/P、E*、E*P	连号访问方式文件寄存器	Ⓢ1、Ⓢ2:1、Ⓓ:2	3
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数	Ⓢ1、Ⓢ2:1	

指令符号	增加了步数的软元件	增加的步数（指令步数）	基本步数
INC、INCP、DEC、DECP、DINC、DINCP、DDEC、DDECP	变址寄存器 / 通用运算寄存器 *2	-1	2
	连号访问方式文件寄存器	3	
	多 CPU 共享软元件		
MOV、MOVP	连号访问方式文件寄存器	1	2
	多 CPU 共享软元件		
DMOV、DMOVP、EMOV、EMOVP	连号访问方式文件寄存器	1	2
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数		
BCD、BCDP、BIN、BINP、FLT、FLTP、CML、CMLP	连号访问方式文件寄存器	ⓢ1:1、ⓢ2:2	2
	多 CPU 共享软元件		
DBCD、DBCDP、DBIN、DBINP、INT、INTP、DINT、DINTP、DFLT、DFLTP、DCML、DCMLP	连号访问方式文件寄存器	ⓢ1:1、ⓢ2:2	2
	多 CPU 共享软元件		
	十进制常数、十六进制常数、实数常数	ⓢ1:1	

*1: 如果 ⓢ1 和 ⓢ2 使用相同的软元件，那么基本步数将增加 1 步。

*2: 使用通用运算寄存器时，步数将减少。

在进行了子集处理的指令中，在指令的多处使用了通用运算寄存器时，则步数将减少。下表列出了上述情况下各指令的步数。

指令符号	使用通用运算寄存器的位置	增加的步数（指令步数）	基本步数
LD=、LD<>、LD<、LD<=、LD>、LD>=、AND=、AND<>、AND<、AND<=、AND>、AND>=、OR=、OR<>、OR<.OR<=、OR>、OR>= LDD=、LDD<>、LDD<、LDD<=、LDD>、LDD>=、ANDD=、ANDD<>、ANDD<、ANDD<=、ANDD>、AND>=、ORD=、ORD<>、ORD<.ORD<=、ORD>、ORD>=	ⓢ1 和 ⓢ2	-2(1)	3
+、-、+P、-P、D+、D-、D+P、D-P、WAND、WOR、WXOR、WXNR、DAND、DOR、DXOR、DXNR、WANDP、WORP、WXORP、WXNRP、DANDP、DORP、DXORP、DXNRP（2 个软元件）	ⓢ1 和 ⓓ	-2(1)	3
+、-、+P、-P、D+、D-、D+P、D-P、WAND、WOR、WXOR、WXNR、DAND、DOR、DXOR、DXNR、WANDP、WORP、WXORP、WXNRP、DANDP、DORP、DXORP、DXNRP（3 个软元件）*1	ⓢ1、ⓢ2 和 ⓓ	-2(1)	3
	ⓢ1、ⓢ2 和 ⓓ	-1(2)	
	ⓢ1 和 ⓢ2（只有在 ⓓ）处指定了不增加步数的软元件时	± 0(3)	
	ⓢ1 和 ⓢ2（只有在 ⓓ 处指定了连号访问方式文件寄存器时）	+2(5)	

*1: 如果 ⓢ1 和 ⓓ 使用相同的软元件，那么基本步数将增加 1 步。

指令符号	使用了通用运算寄存器的位置	增加的步数（指令步数）	基本步数
*、*P、/、/P	ⓢ1 、ⓢ2 和 Ⓓ	-2(1)	3
	ⓢ1 、ⓢ2 和 Ⓓ	-1(2)	
D*、D*P、D/、D/P、E*、E*P	ⓢ1 、ⓢ2 和 Ⓓ	-2(1)	3
	ⓢ1 、ⓢ2 和 Ⓓ	-1(2)	
	ⓢ1 和 ⓢ2（只有在 Ⓓ 处指定了不增加步数的软元件时）	±0(3)	
	ⓢ1 和 ⓢ2（只有在 Ⓓ 处指定了连号访问方式文件寄存器时）	+2(5)	
MOV、MOVP、DMOV、DMOVP、EMOV、EMOVP	ⓢ1 和 Ⓓ	-1(1)	2
BCD、BCDP、BIN、BINP、DBCD、DBCDP、DBIN、DBINP、FLT、FLTP、DFLT、DFLTP、INT、INTP、DINT、DINTP、CML、CMLP、DCML、DCMLP	ⓢ1 和 Ⓓ	-1(1)	2

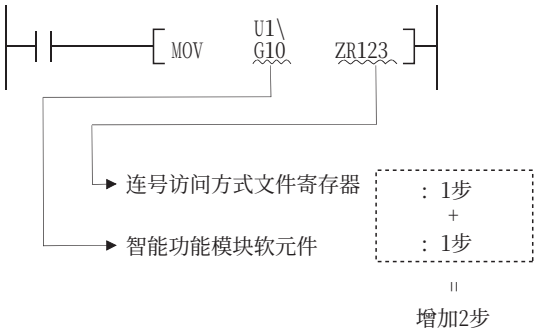
2) 未进行子集处理的指令

下表是在未进行子集处理的指令中，增加了步数的软元件的情况。

增加了步数的软元件	增加的步数	示例
智能功能模块软元件	1	MOV U4\G10 D0
多 CPU 共享软元件		MOV U3E1\G10000 D0
链接直接软元件		MOV J3\B20 D0
变址寄存器 / 通用运算寄存器		MOV Z0 D0
连号访问方式文件寄存器		MOV ZR123 D0
32 位常数		DMOV K123 D0
实数常数		EMOV E0.1 D0
字符串常数	偶数时：(字符数)/2 奇数时：(字符数 +1)/2	$MOV “123” D0

(d) 如果在 (a) ～ (c) 中描述的条件重叠，则步数将累加。

例如 如果指定了 MOV U1\G10 ZR123 ，则合计增加 2 步。



3.9 当 OUT、SET/RST 或 PLS/PLF 指令使用相同的软元件时的操作

以下描述如何在一次扫描周期中执行 OUT、SET/RST 或使用相同软元件的 PLS/PLF 的多个指令。

(1) 使用相同软元件的 OUT 指令：

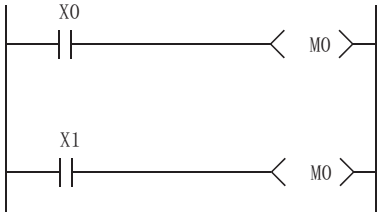
在编程过程中，一个扫描周期使用相同的软元件的 OUT 指令不要超过一次。

如果在一个扫描周期中有使用相同软元件的 OUT 指令，那么根据相关 OUT 指令的程序操作结果，每次 OUT 指令执行时，指定的软元件都将开启或关闭。

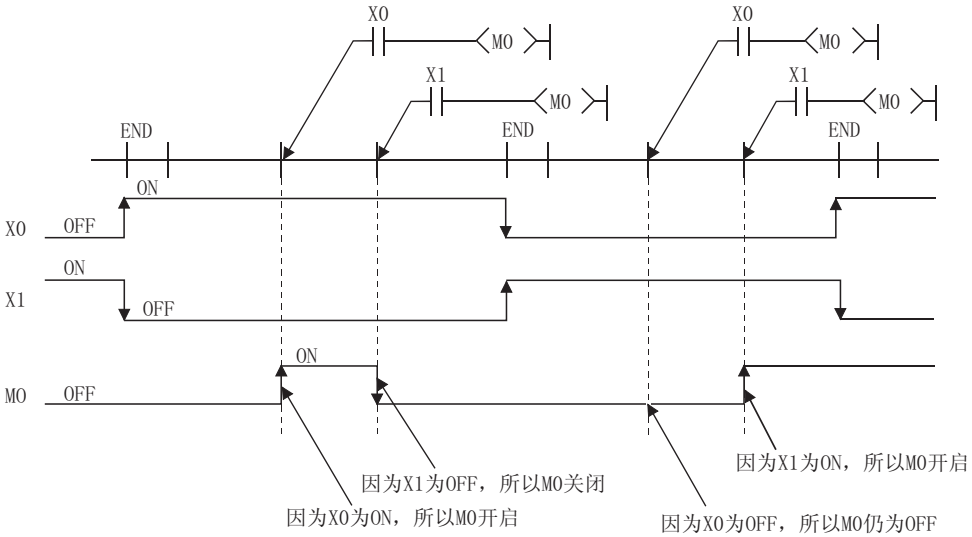
由于每个软元件的 ON 或 OFF 取决于每个 OUT 指令的执行结果，这样在一个扫描周期内软元件可能重复变为 ON 和 OFF。

以下图表为电路示例，显示如何开启 / 关闭带有输入 X0 和 X1 的同一个内部继电器 (M0)。

[电路]



[时序图]



对于刷新型 CPU 模块，当输出 (Y) 被 OUT 指令指定后，扫描周期的最后一个 OUT 指令的 ON/OFF 状态将被输出。

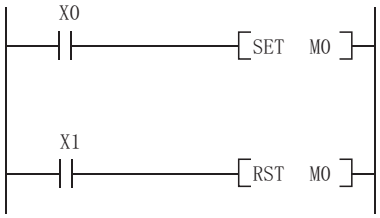
(2) 使用相同软元件的 SET/RST 指令

(a) 当 SET 指令为 ON 时，SET 指令开启指定的软元件。当 SET 指令为 OFF 时，SET 指令不进行任何操作。
为此，当两个或更多的 SET 指令在一个扫描周期中使用相同的软元件时，如果其中任何一个 SET 指令为 ON，那么指定的软软元件将会为 ON。

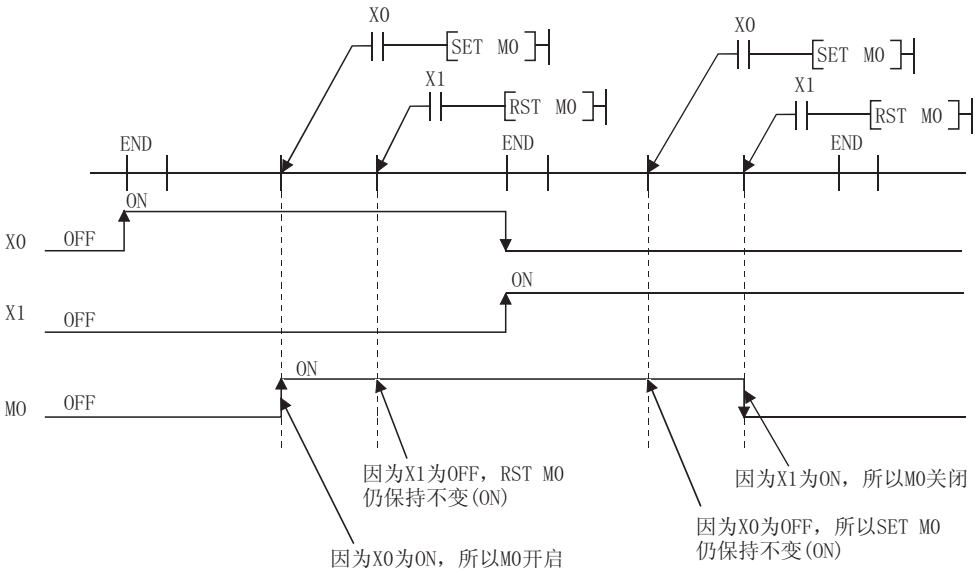
(b) 当 RST 指令为 ON 时，RST 指令关闭指定的软元件。当 RST 指令为 OFF 时，RST 指令不进行任何操作。
为此，当两个或更多的 RST 指令在一个扫描周期中使用相同的软元件时，如果任何一个 RST 指令为 ON，那么指定的软元件将会为 OFF。

(c) 如果使用同一个软元件的 SET 指令和 RST 指令编在一个扫描周期中，那么当 SET 指令为 ON 时，SET 指令开启指定的软元件。当 RST 指令为 ON 时，RST 指令关闭指定的软元件。当 SET 和 RST 命令均为 OFF 时，指定软元件的 ON/OFF 状态将不能改变。

[电路]



[时序图]



对于刷新型 CPU 模块，如果通过 SET/RST 指令指定输出 (Y)，1 个扫描中最后执行的 SET/RST 指令的 ON/OFF 状态将被输出。

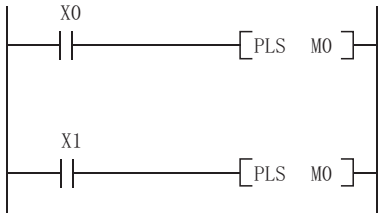
(3) 使用相同软元件的 PLS 指令

当 PLS 指令从 OFF 变为 ON 时，使指定的软元件为 ON。在其它任何时候 (OFF → OFF、ON → ON 和 ON → OFF) 均使指定的软元件为 OFF。

当在一个扫描中执行了多个相同软元件的 PLS 指令时，各 PLS 指令从 OFF 变为 ON 时，使指定的软元件为 ON。在其它任何时候，都使指定的软元件为 OFF。

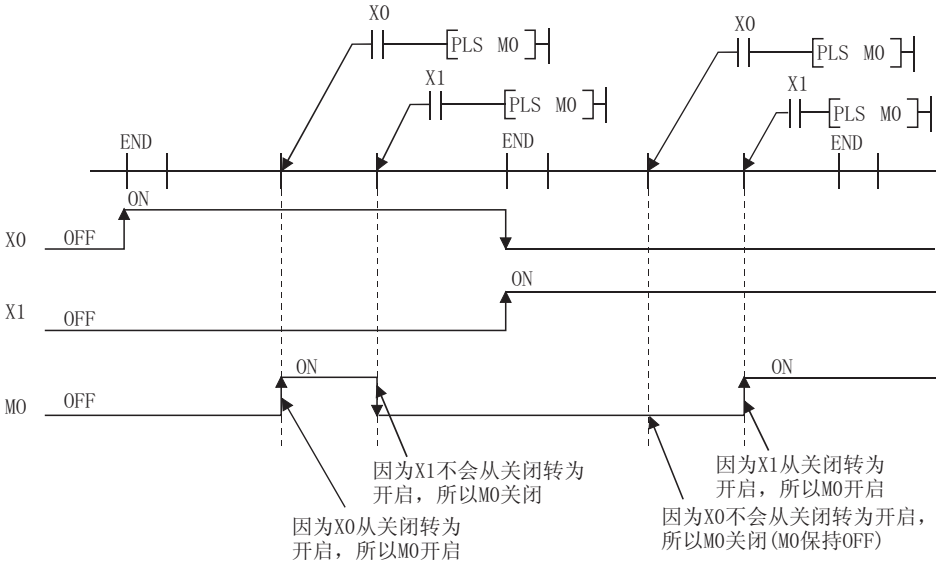
因此，在一个扫描中执行了多个相同软元件的 PLS 指令时，通过 PLS 指令置于 ON 的软元件可能在一个扫描中不为 ON。

[电路]

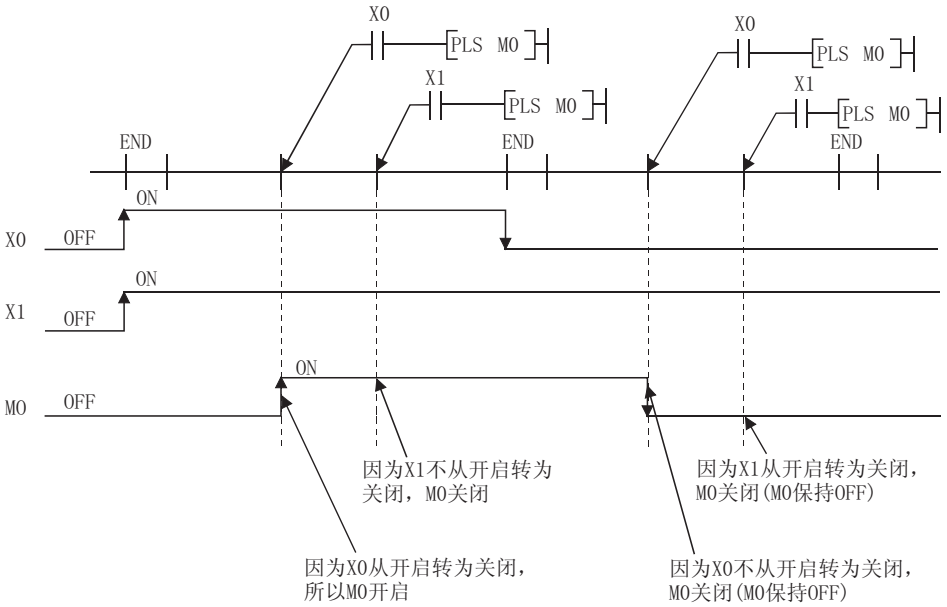


[时序图]

• X0 和 X1 的 ON/OFF 的时刻是不同的 (指定的软元件在整个扫描周期内不会开启)



• X0 和 X1 同时从关闭变为开启



对于刷新型 CPU 模块，如果通过 PLS 指令指定输出 (Y)，1 个扫描中最后执行的 PLS 指令的 ON/OFF 状态将被输出。

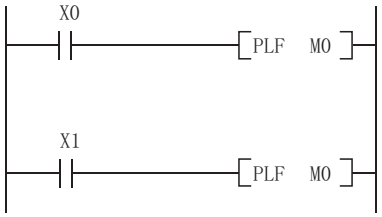
(4) 使用相同软元件的 PLF 指令

当 PLF 指令从关闭变成开启时，PLF 指令将指定的软元件开启，在其他任何时间里均将指定的软元件关闭 (OFF OFF、OFF ON、和 ON ON)。

如果在一个扫描周期中有两个或更多的使用相同的软元件的 PLF 指令时，每一个 PLF 指令在从开启变为关闭时，相应的 PLF 指令将指定的软元件关闭，并且在其他任何时间将指定的软元件将会关闭。

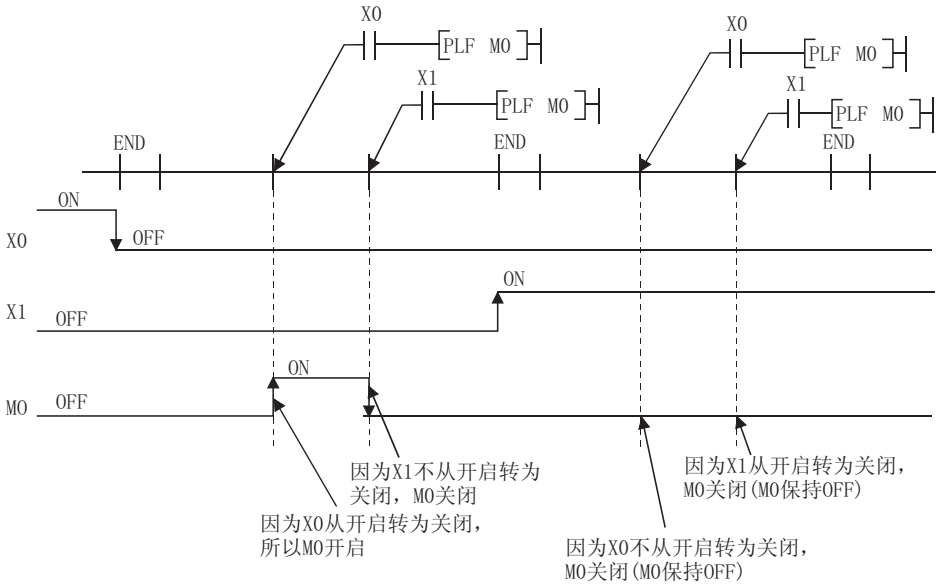
为此，当有两个或更多的使用相同软元件的 PLF 指令被编在一个扫描周期里时，在整个扫描周期内已经被开启的软元件不可以再次开启。

[电路]

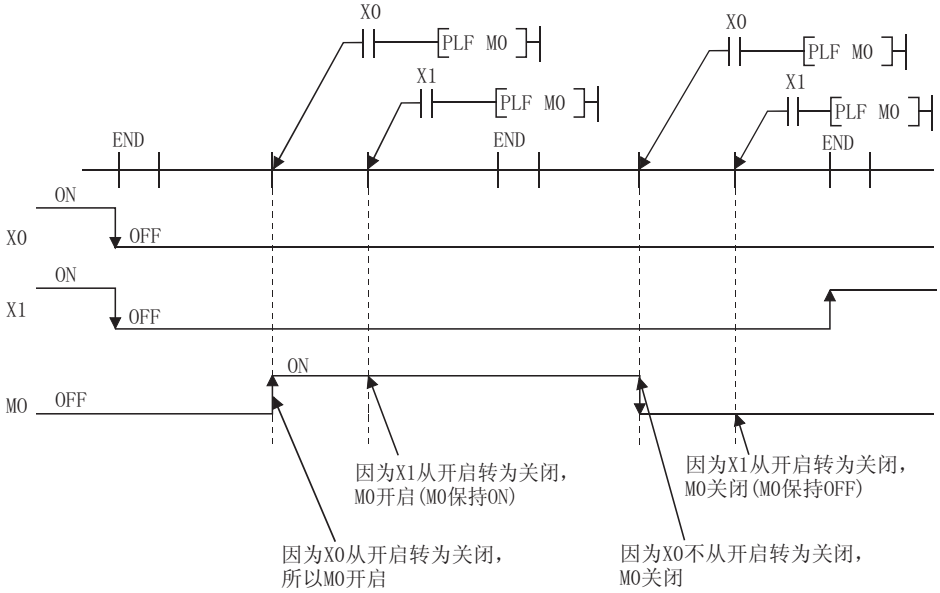


[时序图]

• X0 和 X1 ON/OFF 的时机是不同的（指定的软元件在整个扫描周期内不会开启）



• X0 和 X1 同时从开启变为关闭



对于刷新型 CPU 模块，如果通过 PLF 指令指定输出 (Y)，1 个扫描中最后执行的 PLF 指令的 ON/OFF 状态将被输出。

3.10 使用文件寄存器的注意事项

本节将解释在 QCPU 和 QnACPU 里的使用文件寄存器的注意事项。

(1) 不能使用文件寄存器的 CPU 模块

Q00JCPU 不能使用文件寄存器。当使用文件寄存器时，请使用 Q00JCPU 以外的 CPU 模块。

(2) 即将使用的文件寄存器的设定

如果要使用文件寄存器，即将使用的文件寄存器必须要用可编程控制器参数或 QDRSET 指令来设定。

（因为在 “使用的文件寄存器”中已经预设，因此 Q00CPU 和 Q01CPU 的可编程控制器参数不需要设定。）

如果即将使用的文件寄存器还没有设定，就不能使用该文件寄存器的指令执行普通的运行。

要 点

即使未在可编程控制器参数中设定使用的文件寄存器，也可以创建一个使用了文件寄存器的程序。对于除通用型 QCPU 以外的 CPU，即使将该程序写入到 CPU 模块中并执行，也不会发出出错信息。

但是，不能对文件寄存器进行正常地读取 / 写入，应加以注意。

对于通用型 QCPU，如果执行使用了文件寄存器的程序，将会发出出错信息。

(3) 文件寄存器区域的保护

(a) 高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU 和 QnACPU

当使用文件寄存器时，将文件寄存器登录到标准 RAM/ 内存卡中，预留出文件寄存器的区域。

(b) 基本类模型 QCPU（除 Q00JCPU）

文件寄存器区域已预先在标准的 RAM 中保护起来。用户不需要再保护文件寄存器区域。

下列表格列出了可以在每一个 CPU 模块中使用文件寄存器的存储器。

存储器	QnACPU	高性能型 QCPU、过程 CPU、冗余 CPU、通用型 QCPU	基本型 QCPU（除 Q00JCPU 以外）
标准 RAM	×	○	○
内存卡 *1 *2	○	○	×

○ : 能够被登录、 × : 不能被登录

*1: 当使用闪存时，只有从文件寄存器中读取能够被执行（不能执行写入到闪存）

*2: 当使用 E^2PROM 时，写入到 E^2PROM 可以通过 PROMWR 指令来执行。

备 注

关于文件寄存器的设定方法和文件寄存器区域的预留方法，请参考以下手册：

- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础骗）

(4) 文件寄存器号码的指定超过了登录的点数时

(a) 对于除通用型 QCPU 以外的 CPU

如果在文件寄存器中写入或读取的数据有比指定的点数更大的数，将不会出现错误。
然而，请注意在文件寄存器写入或读取正确的数据将不能被执行。

(b) 通用型 QCPU

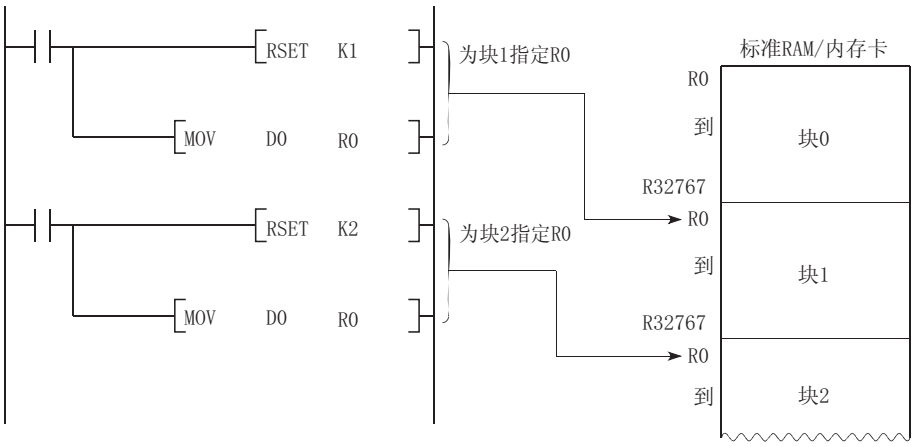
如果文件寄存器写入或读取正确的数据没有指定时，将会出现错误。（出错代码： 4101）

(5) 文件寄存器指定方法

有块切换方法和序列号访问方法两种方法来指定文件寄存器。

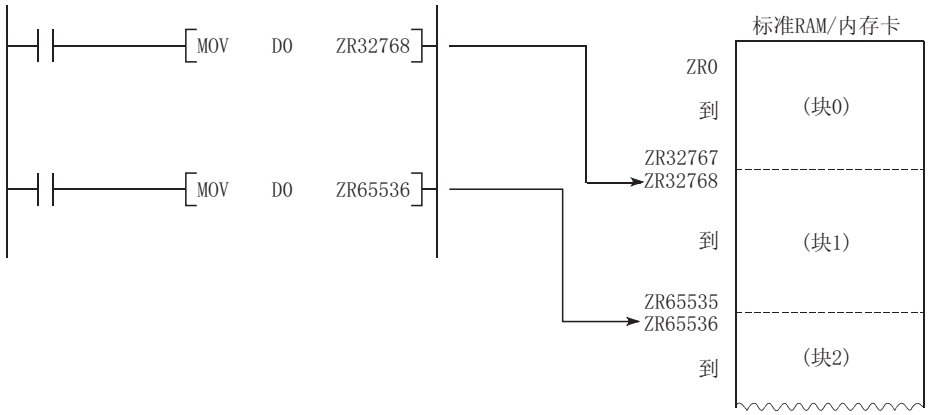
(a) 块切换方法

在块切换方法中，在 32k 点的单位（1 块）里指定使用的文件寄存器的点数。
对于 32k 点或更多的文件寄存器，通过切换 RSET 指令即将使用的块号来指定文件寄存器。
指定每一块为 R0 到 R32767。



(b) 序列号访问方法

当使用序列号访问方法时，通过连续的软元件号指定超过 32k 点的文件寄存器。
多块的文件寄存能够当作连续的文件寄存器使用。
使用 “ZR”为软元件名。



(6) 刷新文件寄存器的设置和限制

(a) 设置

软元件刷新的设置如下：

- MELSECNET/G 的刷新设置
- MELSECNET/H 的刷新设置
- CC-Link 的刷新设置
- 智能功能模块的自动刷新设置
- 多 CPU 系统的自动刷新设置

(b) 限制

指定文件寄存器进行软元件刷新的限制如下。

1) 如果通过可编程控制器参数将文件寄存器与程序指定为相同名，将不会正确执行刷新。

当使用与程序有相同名称的文件寄存器，对和可编程控制参数的［程序］标签面的末号中设置的程序有相同名的文件寄存器执行参数刷新。如果要写入或读取刷新数据，请通过 QDRSET 指令，在文件寄存器切换后，将文件寄存器指定至刷新软元件。

2) 如果文件寄存器的文件名或驱动号由 QDRSET 指令更改后，将不会正确执行刷新。

如果文件寄存器的文件名或驱动号由 QDRSET 指令更改后，在 END 指令执行时，文件设置的数据会执行链接刷新。如果要写入或读取刷新数据，请在 END 指令执行时，指定文件设置的文件寄存器。
当为除通用型 QCPU 以外的 CPU 模块中的软元件指定 “ZR”，如果驱动号由 QDRSET 指令更改后，会出现错误 (LINK PARA ERROR (3101))。(注意，当为软元件指定 “R”，将不会发生错误。)

3) 当通过 RSET 指令切换块号，在切换的块号中的文件寄存器 (R) 的数据会执行刷新。

当通过 RSET 指令切换块号，在 END 指令执行时，在块号中的文件寄存器 (R) 的数据会执行刷新。如果要写入或读取刷新数据，请在 END 指令执行时，指定块号的文件寄存器。

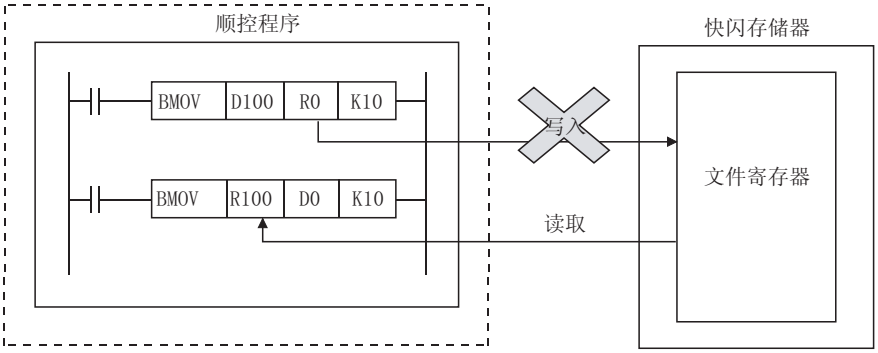
(7) 使用闪存中的文件寄存器的注意事项

本节介绍 QCPU 和 QnQCPU 中可使用文件寄存器的闪存卡的注意事项。

(a) 下列闪存可以和 QCPU 或 QnACPU 一起使用。

- QCPU: 闪存卡
- QnACPU:SRAM+ 闪存类型的存储卡

(b) 只有在顺控程序中，才可以读取闪存中的文件寄存器。
（不能在顺控程序中执行对闪存的写入）。

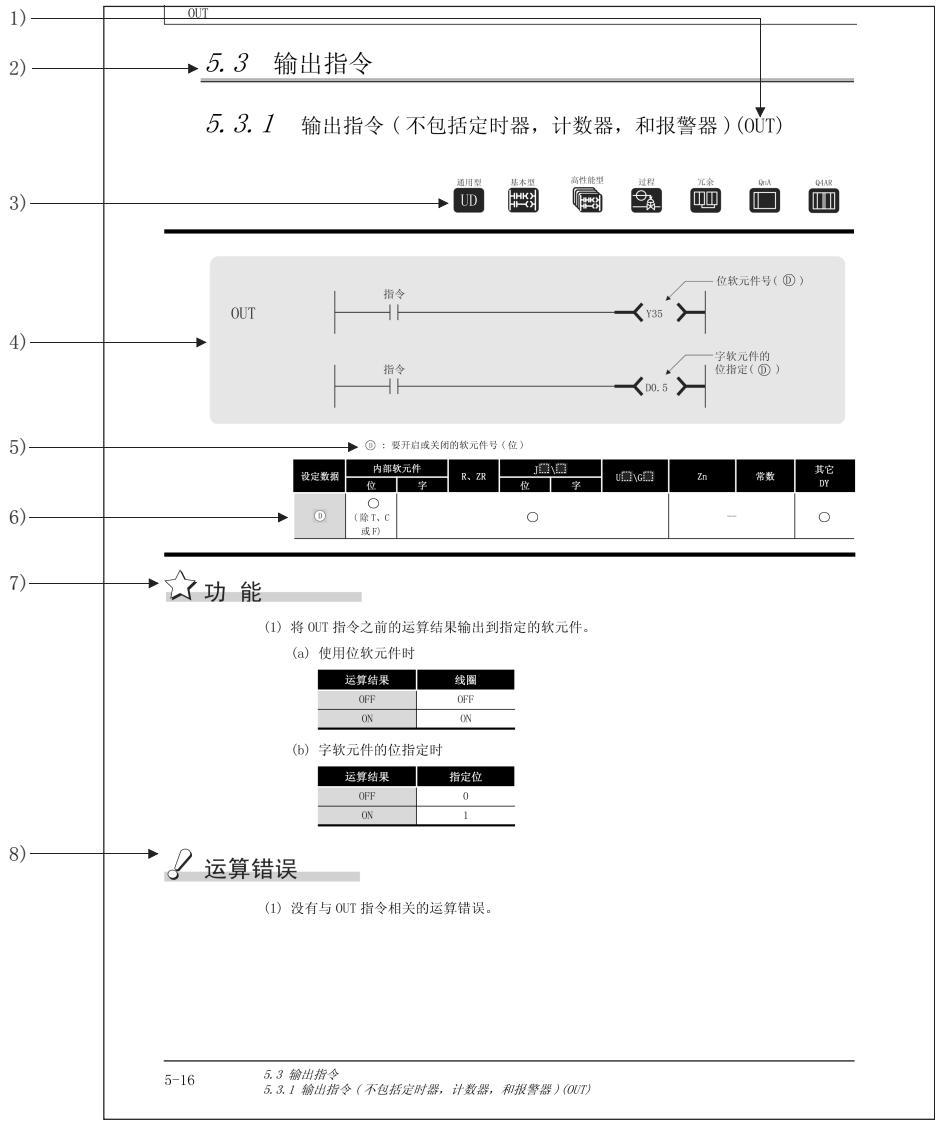


当使用用于文件寄存器的闪存，要先写入数据。

- QCPU
  使用 GX Developer，将数据写入闪存卡。
- QnACPU
  使用 IC 内存卡和读取器 / 写入器，将数据写入 SRAM+ 闪存型内存卡。

4 如何阅读指令

在后面各章节中的指令描述将以以下格式表述：

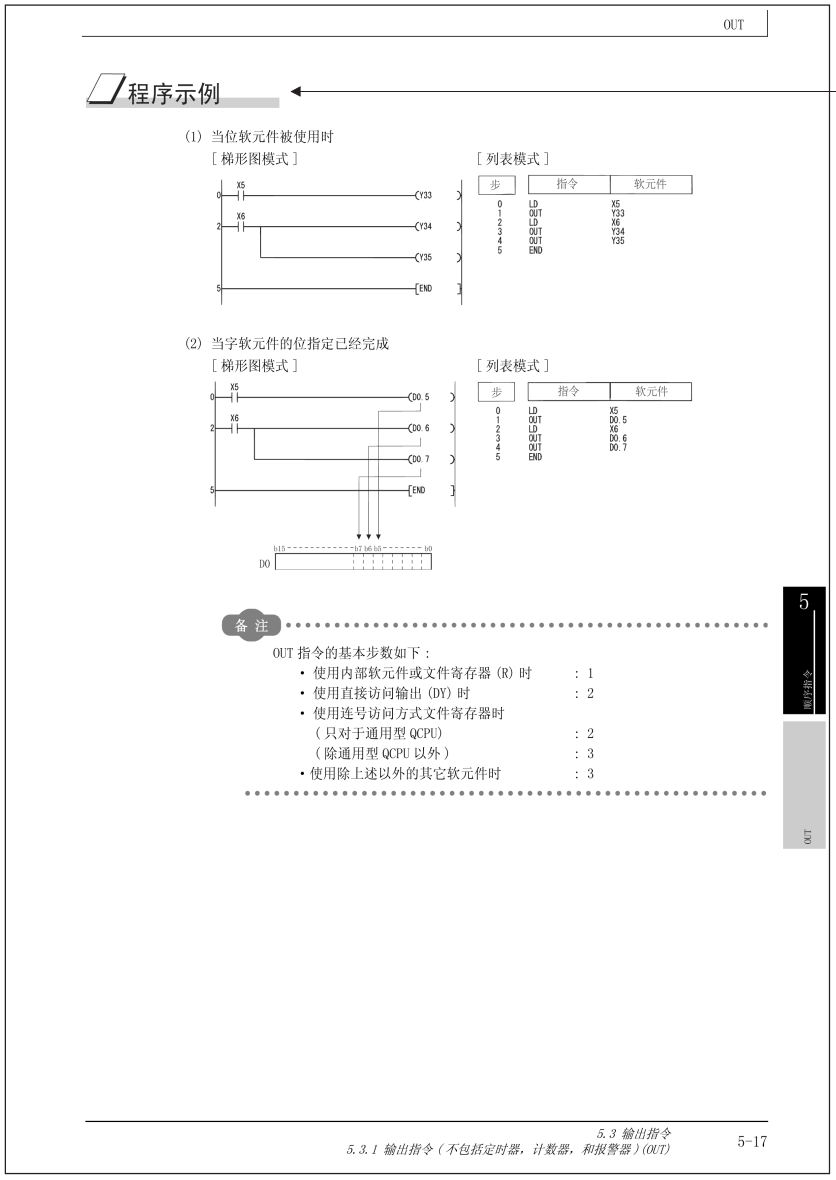


1) 用来写指令的代码（指令符号）。

2) 章节号和被描述指令的总分类。

3) 表示指令能否在各 CPU 模块类型中使用。

图标							含义
通用型 QCPU	基本型 QCPU	高性能型 QCPU	过程 CPU	冗余 CPU	QnACPU	Q4ARCPU	
通用型 UD	基本型	高性能型	过程	冗余	QnA	Q4AR	普通图标，表示可以使用相应指令。
通用型 UD	基本型	高性能型	过程	冗余	QnA	Q4AR	带 △（三角）的图标，表示可以使用但有某些限制的指令（例如：功能版本，软件版本）。
通用型 UD	基本型	高性能型	过程	冗余	QnA	Q4AR	带 ×（叉号）的图标，表示不可以使用的相应指令。



4) 表示梯形图模式的表达和指令的执行条件。

执行条件	非条件执行	在 ON 时执行	在 ON 时执行一次	在 OFF 时执行	在 OFF 时执行一次
代码记录在记述页	没有符号被记录				

5) 讨论为每个指令设定的数据和数据类型。

数据类型	含义
位	位数据或位数据中的第一个数字
BIN 16 位	BIN 16 位数据或字软元件中的第一个数字
BIN 32 位	BIN 32 位数据或双字软元件中的第一个数字
BCD 4 位	4 位 BCD 数据
BCD 8 位	8 位 BCD 数据
实数	浮动小数点数据
字符串	字符串数据
软元件名	软元件名称数据

6) 指令中可以使用的软元件用○表示。可以使用的软元件型号表示如下：

设定数据	内部软元件（系统、用户）		文件寄存器 R、ZR	直接链接软元件 *4 J□\□		智能功能模块 U□\G□	变址寄存器 Zn	常数 *5	其它 *5
	位	字		位	字				
可用软元件 *1	X、Y、M、L、SM、F、B、SB、FX、FY *2	T、ST、C、*3 D、W、SD、SW、FD、@□	R、ZR	J□\X J□\Y J□\B J□\SB	J□\W J□\SW	U□\G□	Z	K、H 、E、$	P、I、J、U、DX、DY、N、BL、TR、BL\S、V

*1: 关于各软元件的说明，请参见以下手册。

- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础骗）

*2: FX 和 FY 只能用于位数据，FD 只能用于字数据。

*3: 当 T、ST 和 C 用于除以下指令以外的情况时，只能用于字数据。
（不能用于位数据。）
[位数据中可使用的指令]
LD、LDI、AND、ANI、OR、ORI、LDP、LDF、ANDP、ANDF、ORP、ORF、OUT、RST

*4: 在 MELSECNET/G、MELSECNET/H 和 MELSECNET/10 中可以使用。

*5: 可以进行设置的软元件记录在 “常数”和 “其它”栏中。

7) 表示指令的功能。

8) 表示引起出错的条件及出错号代码。关于未记载的错误，请参见 3.6 节。

9) 以梯形图和列表这两种模式表示简单的程序示例。此外，表示该程序被执行时的各软元件的内容。

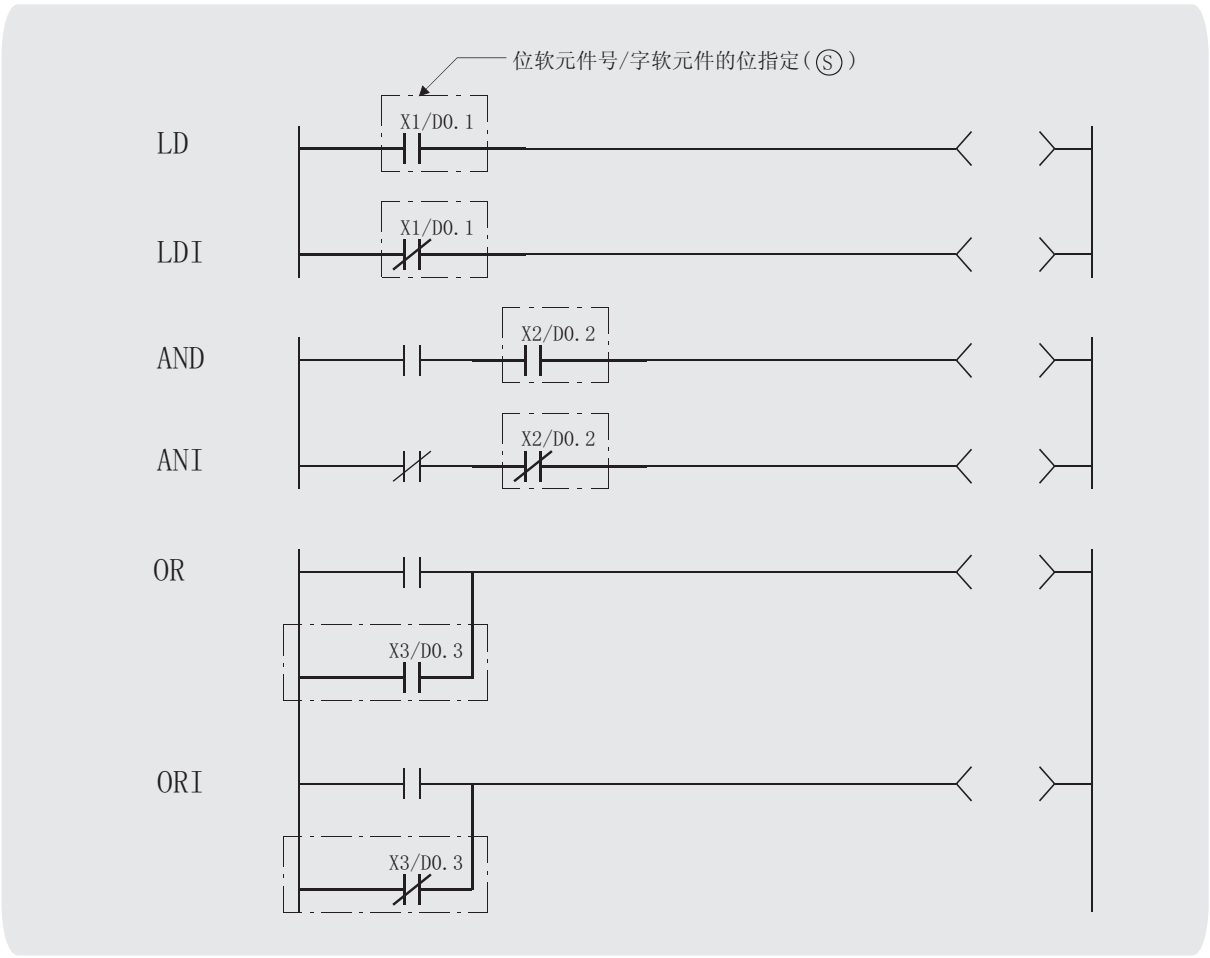
5 顺序指令

指令	含义	参见
触点指令	运行开始，串行连接，并行连接	5.1 节
连接指令	梯形图块连接，从操作结果中生成脉冲，储存 / 读取操作结果	5.2 节
输出指令	位软元件输出，脉冲输出，输出反转	5.3 节
转移指令	位软元件转移	5.4 节
主控指令	主控	5.5 节
结束指令	程序结束	5.6 节
其他指令	程序停止、无处理等不属于上述种类的指令	5.7 节

5.1 触点指令

5.1.1 运行开始，串行连结，并行连接 (LD、LDI、AND、ANI、OR、ORI)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：用作触点的软元件（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它 DX、BL
	位	字		位	字				
Ⓢ	○						--		○

功 能

LD、 LDI

(1) LD 是 A 触点运算开始指令，LD1 是 B 触点运算开始指令。其功能是从指定的软元件中读取 ON/OFF 信息*1，并将其作为运算结果。

*1: 进行了字软元件的位指定时，字软元件将根据指定位的 1/0 而 ON/OFF。

AND、 ANI

(1) AND 是触点 A 串行连接指令，ANI 是触点 B 串行连接指令，其功能是读取指定位软元件的 ON/OFF 数据*2，将该数据与至目前为止的运算结果执行 AND 运算，并将该值作为运算结果。

*2: 进行了字软元件的位指定时，字软元件将根据指定位的 1/0 而 ON/OFF。

(2) 对 AND 或 ANI 的使用没有限制，但是下列内容适用于梯形图模式中的外围使用的软元件：

(a) 写入..... 当 AND 和 ANI 为串行连接时，能够产生一个最大为 24 级的梯形图。
(b) 读取..... 当 AND 和 ANI 为串行连接时，能够显示一个最大为 24 级的梯形图。
如果数量超过 24 级，最大显示为 24 级。

OR、 ORI

(1) OR 是触点 A 单并性连接指令，ORI 是触点 B 单并行连接指令。它们从指定的软元件读取 ON/OFF 信息*3（如果字软元件位已经被指定，就变成指定位的 1/0 状态），并对到该点的运行结果执行 OR 运行，使用结果数值作为运行结果。

*3: 字软元件的位指定时，根据指定位的 1/0 而 ON/OFF。

(2) 对 OR 或 ORI 的使用没有限制，但是下列内容适用于梯形图模式中使用的外围设备：

(a) 写入..... 最多可以创建通过 23 个 OR 和 ORI 连接的梯形图。
(b) 读取..... 最多可以显示通过 23 个 OR 和 ORI 连接的梯形图。
第 24 个和后来的梯形图不能被正确显示。

备 注

字软元件位的指定以十六进制形式进行设置
D0 的位 b11 应该是 D0.0B。
字软元件的位指定的更多信息，请参见 3.2.1 项。

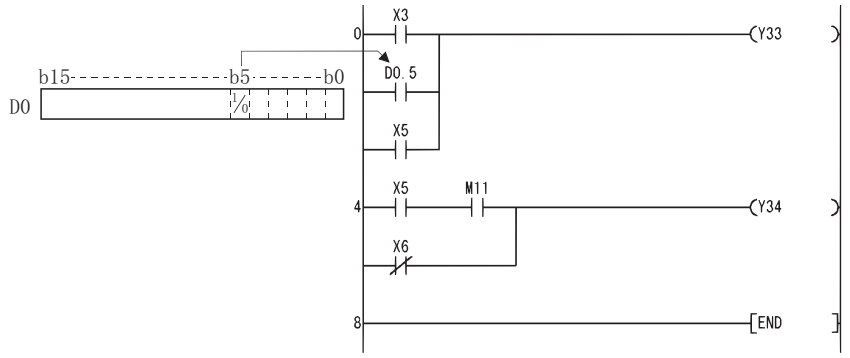
运算错误

(1) 在 LD、LDI、AND、ANI、OR 或 ORI 指令下，无运行错误。

程序示例

(1) 一个使用 LD、AND、OR 和 ORI 指令的程序

[梯形图模式]

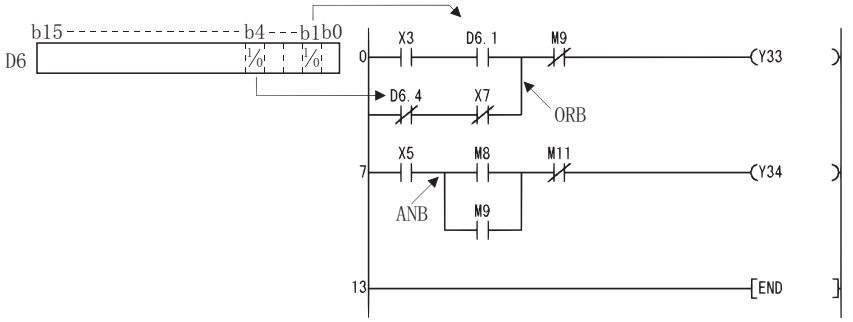


[列表模式]

步	指令	软元件
0	LD	X3
1	OR	D0.5 … 字软元件的位指定
2	OR	X5
3	OUT	Y33
4	LD	X5
5	AND	M11
6	ORI	X6
7	OUT	Y34
8	END	

(2) 一个通过使用 ANB 和 ORB 指令建立起来的链接触点程序。

[梯形图模式]

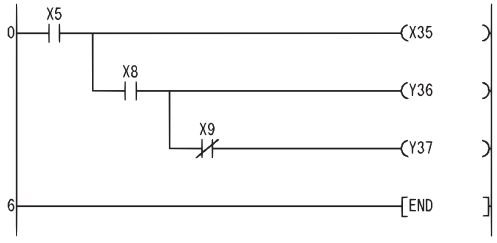


[列表模式]

步	指令	软元件
0	LD	X3
1	AND	D6.1 … 字软元件的位指定
2	LDI	D6.4
3	ANI	X7
4	ORB	
5	ANI	M9
6	OUT	Y33
7	LD	X5
8	LD	M8
9	OR	M9
10	ANB	
11	ANI	M11
12	OUT	Y34
13	END	

(3) 一个使用 OUT 指令的并行程序

[梯形图模式]

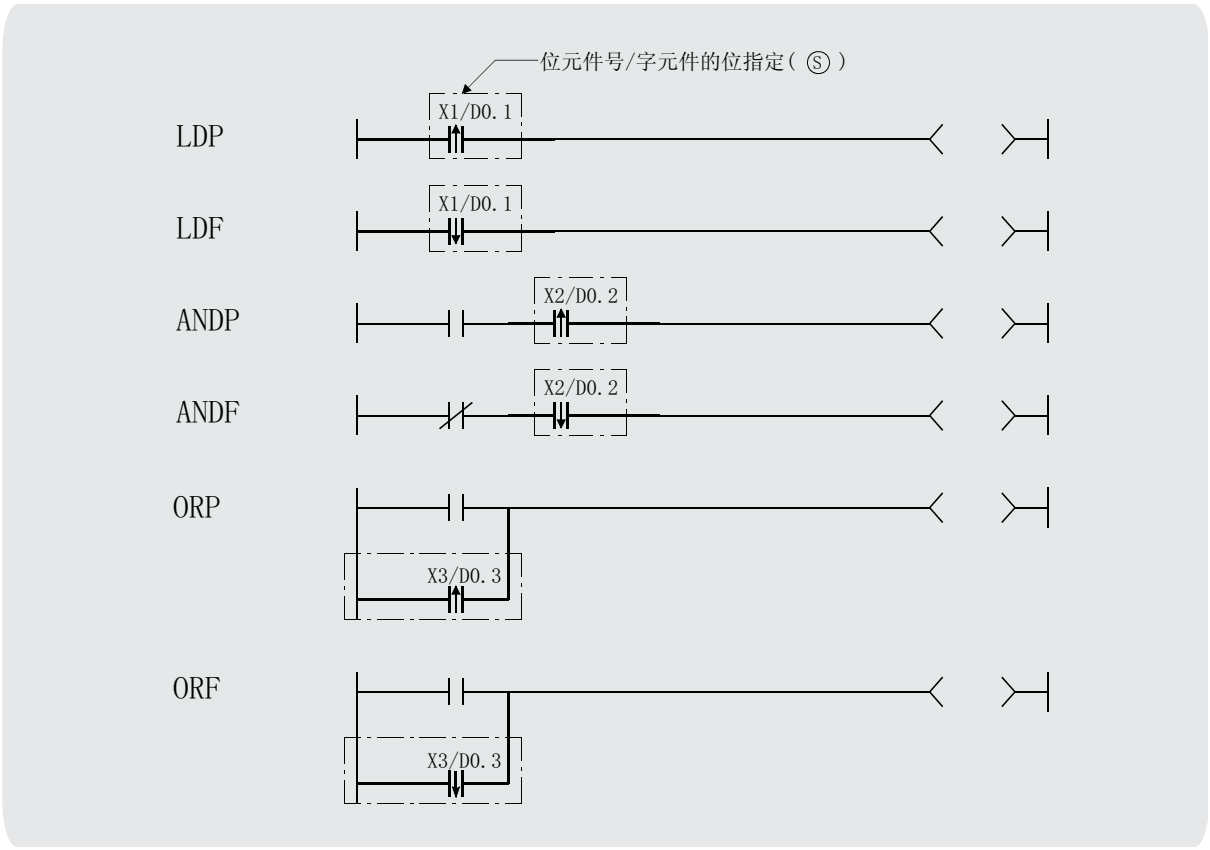


[列表模式]

步	指令	软元件
0	LD	X5
1	OUT	X35
2	AND	X8
3	OUT	Y36
4	ANI	X9
5	OUT	Y37
6	END	

5.1.2 脉冲运行开始，脉冲串行连接，脉冲并行连接（LDP、LDF、ANDP、ANDF、ORP、ORF）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



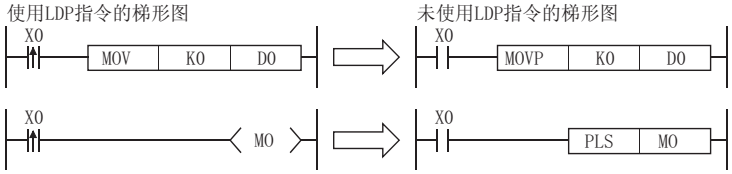
ⓢ：用作触点的软元件（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它 DX
	位	字		位	字				
ⓢ	○						--		○

功 能

LDP、 LDF

(1) LDP 是上升沿脉冲操作开始指令，并且只有在指令位软元件的上升沿（当它从 ON 转换成 OFF）才为 ON。
如果字软元件已经被指定，仅当指定的位从 0 改变为 1 时才为 ON。
如果只有一个 LDP 指令，那么它和在 ON 时（□P）执行的产生脉冲的指令具有同样作用。



(2) LDF 是下降沿脉冲操作开始指令，并且仅在指定位软元件的下降沿（从 ON 转换成 OFF 时）才为 ON。

如果字软元件被指定，那么仅当指定的位从 1 变为 0 时才为 ON。

ANDP、 ANDF

(1) ANDP 是一个上升沿脉冲串行连接指令，ANDF 是一个下降沿脉冲串行连接指令。它们对至该点的运行结果执行 AND 操作，并且将结果值作为运行结果。

供 ANDP 和 ANDF 使用的 ON/OFF 数据，如下表显示：

通过 ANDP 或 ANDF 指定的软元件		ANDP 的状态	ANDF 的状态
位软元件	字软元件的位指定		
OFF → ON	0 → 1	ON	OFF
OFF	0	OFF	
ON	1		
ON → OFF	1 → 0		ON

ORP、 ORF

(2) ORP 是上升沿脉冲并联连接指令，ORF 是下降沿脉冲串联连接指令，其功能为对至目前为止的运算结果进行 OR 运算后作为运算结果。

供 ORP 和 ORF 使用的 ON/OFF 数据，如下表显示：

通过 ORP 或 ORF 指定的软元件		ORP 的状态	ORF 的状态
位软元件	字软元件位指定		
OFF → ON	0 → 1	ON	OFF
OFF	0	OFF	
ON	1		
ON → OFF	1 → 0		ON

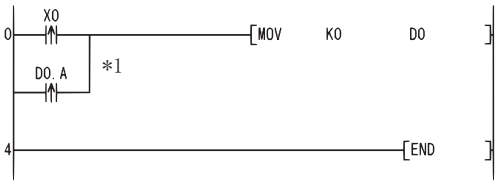
运算错误

(1) 使用 LDP，LDF，ANDP，ANDF，ORP，或 ORF，不会有运行错误。

程序示例

(1) 当在输入 X0 或者在数据寄存器 D0 的 b10（位 10）的上升沿，下述程序执行 MOV 指令：

［梯形图模式］



［列表模式］

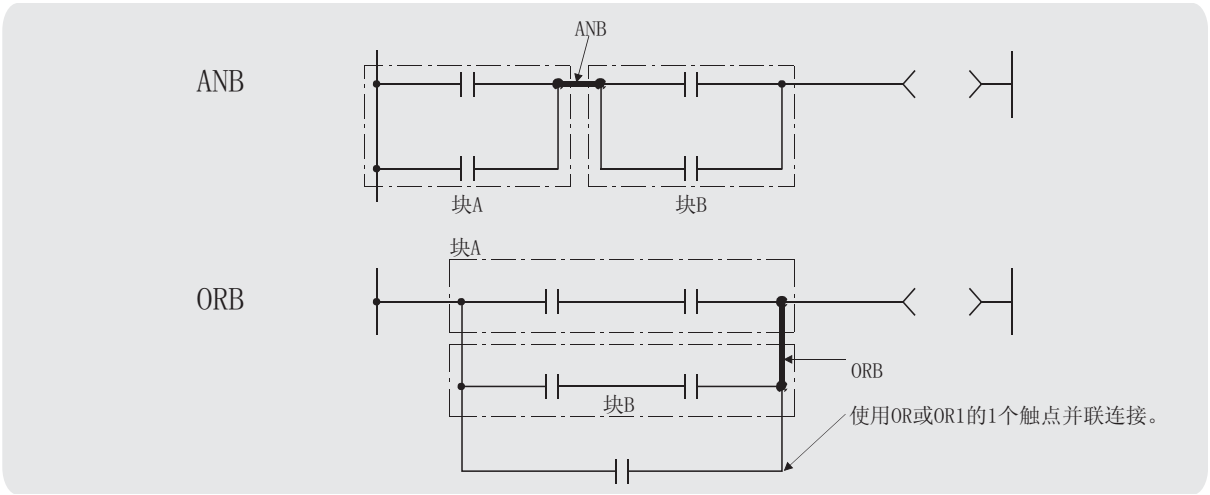
步	指令	软元件	
0	LDP	X0	
1	ORP	D0.A	
2	MOV	K0	D0
4	END		

*1: 字软元件的位指定是以十六进制的形式执行的。
D0 的 b10 应为 D0.0A.。

5.2 连接指令

5.2.1 梯形图块串行连接和并行连接 (ANB、ORB)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

ANB

(1) 在块 A 和块 B 上执行 AND 操作，并且将结果值作为运算结果。

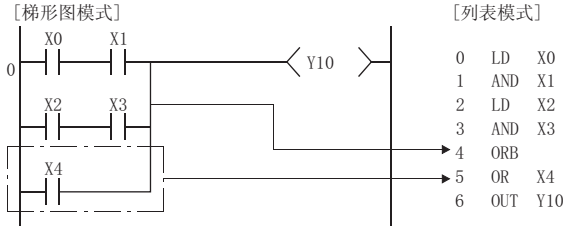
(2) ANB 的符号不是触点符号，而是连接符号。

(3) 当在列表模式下编程时，最多可以连续写入 15 条 ANB 指令 (16 块)。

ORB

(1) 在块 A 和 B 上执行一个 OR 操作，并且将结果值作为运行结果。

(2) ORB 是用于为有两个或更多触点的梯形图块执行并行连接的。
对于只有一个触点的梯形图块，使用 OR 或 ORI ；在这种情况下没有必要使用 ORB。



(3) ORB 符号不是触点符号，而是连接符号。

(4) 当在列表模式中编程时，可以连续使用最多 15 个 ORB 指令 (16 块)。

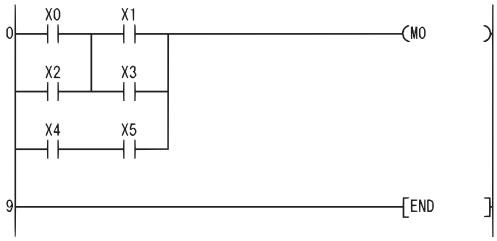
运算错误

(1) 没有与 ANB 或 ORB 相关的操作错误。

程序示例

(1) 一个使用 ANB 和 ORB 指令的程序。

[梯形图模式]

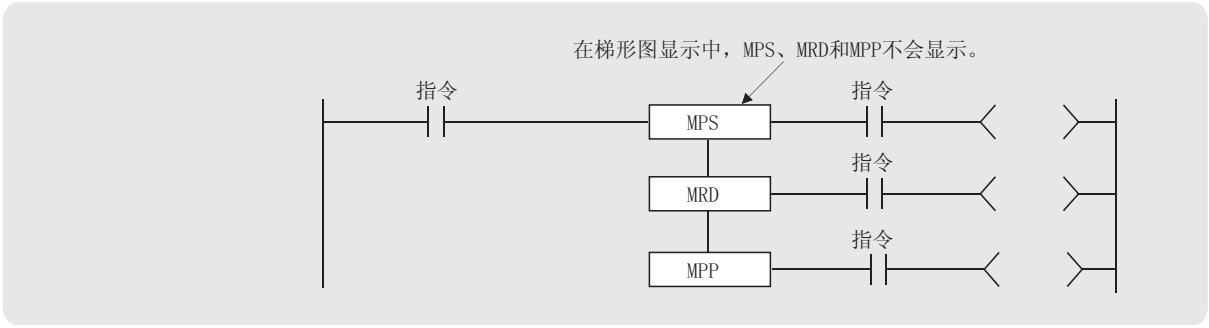


[列表模式]

步	指令	软元件
0	LD	X0
1	OR	X2
2	LD	X1
3	OR	X3
4	ANB	
5	LD	X4
6	AND	X5
7	ORB	
8	OUT	M0
9	END	

5.2.2 运行结果推进，读取，弹出 (MPS、MRD、MPP)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

MPS

(1) 在 MPS 指令之前立即将运行结果 (ON 或 OFF) 储存在存储器内。

(2) 可以连续使用最多 16 条 MPS 指令。
然而，在梯形图模式中最多能创建 11 条。
如果在该处理中使用了一个 MPP 指令，那么 MPS 指令计算使用的数量将会减少一个。

MRD

(1) 读取为 MPS 指令储存的运算结果，并且使用该结果来执行下一步操作。

MPP

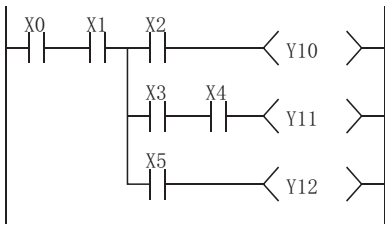
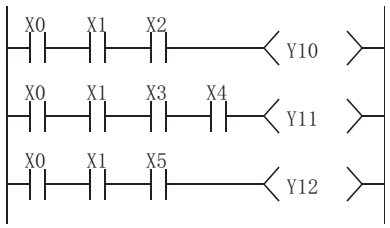
(1) 读取为 MPS 指令储存的运算结果，并且使用该结果来执行下一步操作。

(2) 清除被 MPS 指令储存的运算结果。

(3) 从 MPS 指令使用次数数量中减去 1。

要 点

1. 以下列出了已使用和未使用 MPS，MRD 和 MPP 指令的梯形图。

使用 MPS，MRD 和 MPP 指令的梯形图	未使用 MPS，MRD 和 MPP 指令的梯形图
	

2. MPS 和 MPP 指令必须使用同样的次数。
如果没有注意到这点，在外围设备中的梯形图模式中将不会正确显示梯形图。

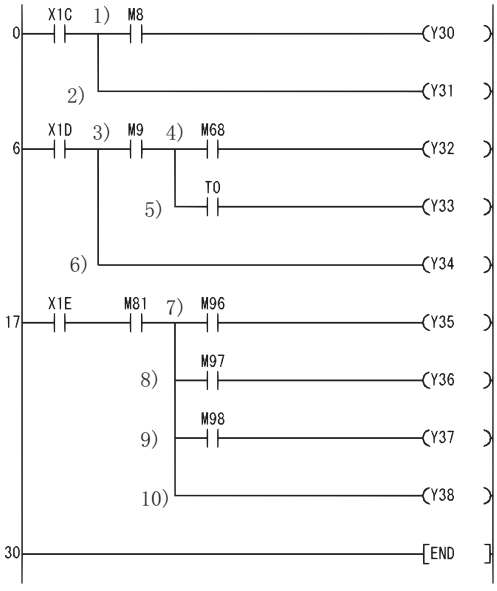
运算错误

(1) 没有与 MPS，MRD 或 MPP 指令相关的操作错误。

程序示例

(1) 一个使用 MPS，MRD 和 MPP 指令的程序。

[梯形图模式]

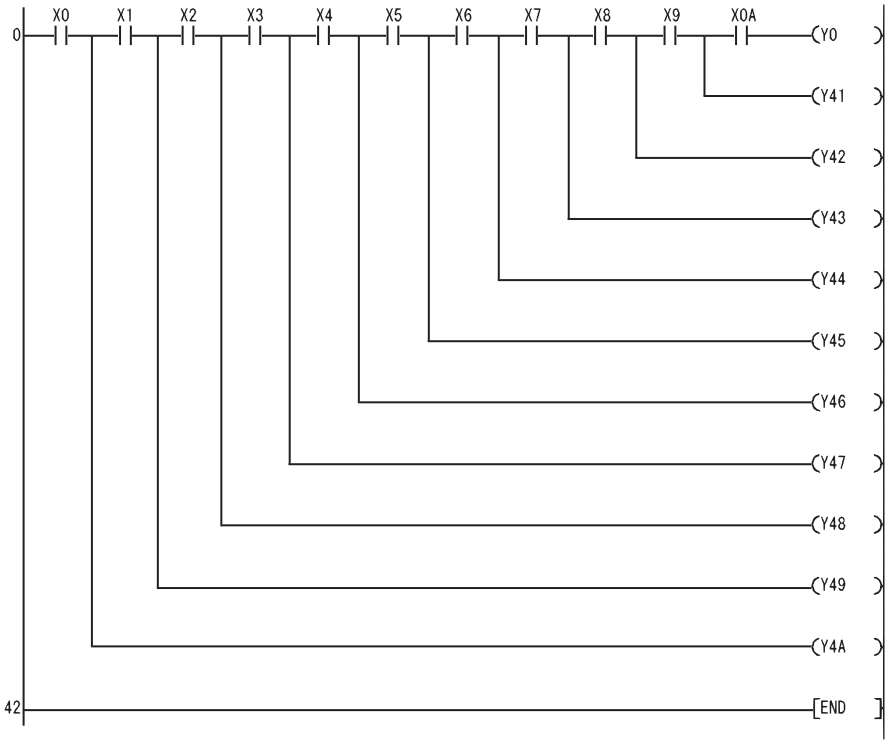


[列表模式]

	步	指令	软元件
1)	0	LD	X1C
	1	MPS	
	2	AND	M8
	3	OUT	Y30
2)	4	MPP	
	5	OUT	Y31
	6	LD	X1D
3)	7	MPS	
	8	AND	M9
4)	9	MPS	
	10	AND	M68
	11	OUT	Y32
5)	12	MPP	
	13	AND	T0
	14	OUT	Y33
6)	15	MPP	
	16	OUT	Y34
	17	LD	X1E
	18	AND	M81
7)	19	MPS	
	20	AND	M96
	21	OUT	Y35
8)	22	MRD	
	23	AND	M97
	24	OUT	Y36
9)	25	MRD	
	26	AND	M98
	27	OUT	Y37
10)	28	MPP	
	29	OUT	Y38
	30	END	

(2) 一个连续使用 MPS 和 MPP 指令的程序。

[梯形图模式]

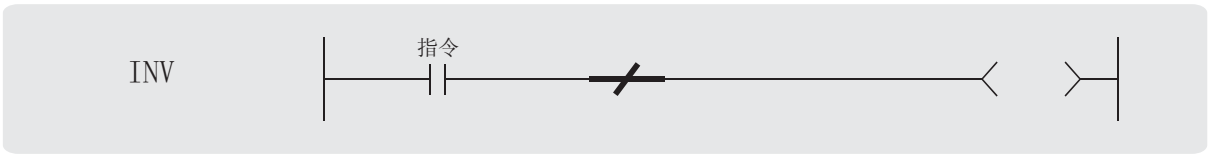


[列表模式]

步	指令	软元件
0	LD	X0
1	MPS	
2	AND	X1
3	MPS	
4	AND	X2
5	MPS	
6	AND	X3
7	MPS	
8	AND	X4
9	MPS	
10	AND	X5
11	MPS	
12	AND	X6
13	MPS	
14	AND	X7
15	MPS	
16	AND	X8
17	MPS	
18	AND	X9
19	MPS	
20	AND	X0A
21	OUT	Y0
22	MPP	
23	OUT	Y41
24	MPP	
25	OUT	Y42
26	MPP	
27	OUT	Y43
28	MPP	
29	OUT	Y44
30	MPP	
31	OUT	Y45
32	MPP	
33	OUT	Y46
34	MPP	
35	OUT	Y47
36	MPP	
37	OUT	Y48
38	MPP	
39	OUT	Y49
40	MPP	
41	OUT	Y4A
42	END	

5.2.3 运行结果反转 (INV)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

在 INV 指令之前立即转换运行结果。

在 INV 指令之前的直接运算结果	在 INV 指令执行之后的运算结果
OFF	ON
ON	OFF

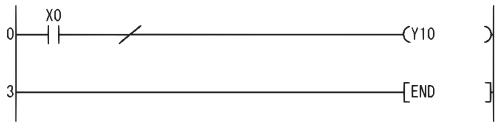
运算错误

(1) 没有与 INV 指令相关的操作错误。

程序示例

(1) 一个从 Y10 转换 X0 ON/OFF 数据和输出的程序。

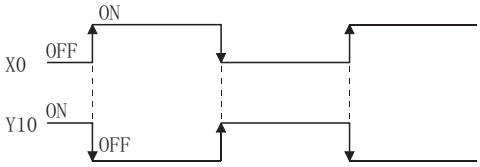
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	INV	
2	OUT	Y10
3	END	

[时序图]

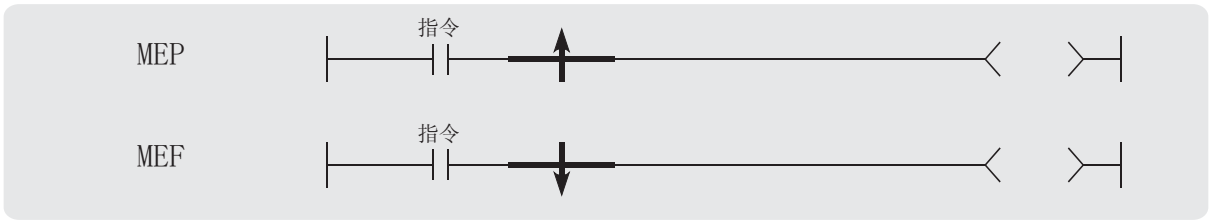


要 点

建立在计算结果基础上的 INV 指令运行只有在 INV 指令给定的情况下才能进行。
相应地，在 AND 指令相同的位置使用该指令，而 INV 指令不能在 LD 和 OR 位置上使用。

5.2.4 运算结果脉冲变换 (MEP、MEF)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

MEP

(1) 如果一直到 MEP 指令前的运算结果都为上升沿（从 OFF 到 ON），那么就为 ON（导通状态）。
如果一直到 MEP 指令前的运算结果都不为上升沿，那么就为 OFF（非导通状态）。

(2) 当多触点串行连接时，MEP 指令的使用简化了脉冲变换处理。

MEF

(1) 如果一直到 MEF 指令前的运算结果都为下降沿（从 ON 到 OFF），那么就为 ON（导通状态）。
如果一直到 MEF 指令前的运算结果都不为下降沿，那么就为 OFF（非导通状态）。

(2) 当多触点串行连接时，MEF 指令的使用简化了脉冲变换处理。

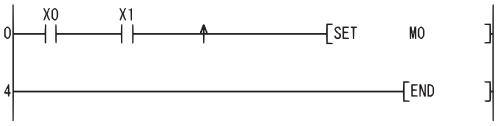
运算错误

(1) 没有与 MEP 或 MEF 指令相关的操作错误。

程序示例

(1) 在 X0 与 X1 的运算结果上执行脉冲变换的程序：

［梯形图模式］



［列表模式］

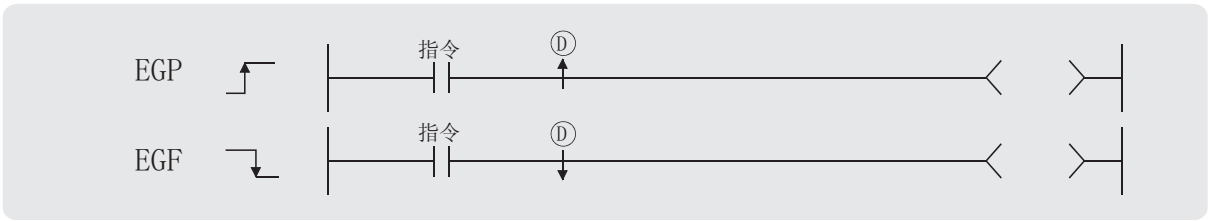
步	指令	软元件
0	LD	X0
1	AND	X1
2	MEP	
3	SET	M0
4	END	

要 点

1. 当为一个已经被副程序或 FOR 到 NEXT 指令变址的触点进行脉冲变换时，MEP 和 MEF 指令偶尔会运行不正常。
如果进行脉冲变换的触点已经被副程序或 FOR 到 NEXT 指令索引过，可以使用 EGP/EGF 指令。
2. 因为 MEP 和 MEF 指令是用在 MEP 和 MEF 指令之前的运算结果来进行操作的，因此在相同位置应该使用 AND 指令。
MEP 和 MEF 指令不能在 LD 或 OR 的位置使用。

5.2.5 变址继电器运算结果的脉冲变换（EGP、EGF）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：储存运算结果的变址继电器号（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				V
Ⓓ	--								○

功 能

EGP

(1) 执行 EGP 指令之前的运算结果被变址继电器 (V) 储存在存储器里。

(2) 执行 EGP 指令之前的运算结果在上升沿 (OFF 到 ON) 为 ON（导通状态）。
如果执行 EGP 指令之前的运算结果不是在上升沿（例如，从 ON 到 ON，从 ON 到 OFF，或从 OFF 到 OFF)，那么为 OFF（非导通状态）。

(3) EGP 指令是在副程序里使用的，并且用在 FOR 和 NEXT 指令之间通过变址修饰对指定的程序进行脉冲操作。

(4) EGP 指令可以同 AND 指令一样使用。

EGF

(1) 执行 EGF 指令之前的运算结果被边沿寄存器 (V) 储存在存储器中。

(2) 执行 EGF 指令之前的运算结果在下降沿 (ON 到 OFF) 为 ON。
如果执行 EGF 指令之前的运算结果不是为下降沿（例如从 OFF 到 ON，从 ON 到 ON，或从 OFF 到 OFF)，则为 OFF（非导通状态）。

(3) EGF 指令是在副程序里使用的，并且用在 FOR 和 NEXT 指令之间通过变址修饰对指定的程序进行脉冲操作。

(4) EGP 指令可以同 AND 指令一样使用。

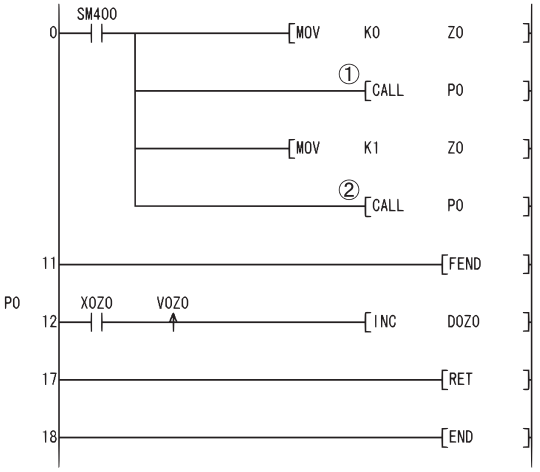
运算错误

(1) 没有与 EGP 或 EGF 指令相关的运行错误。

程序示例

(1) 在子程序中使用 EGP 指令的程序。

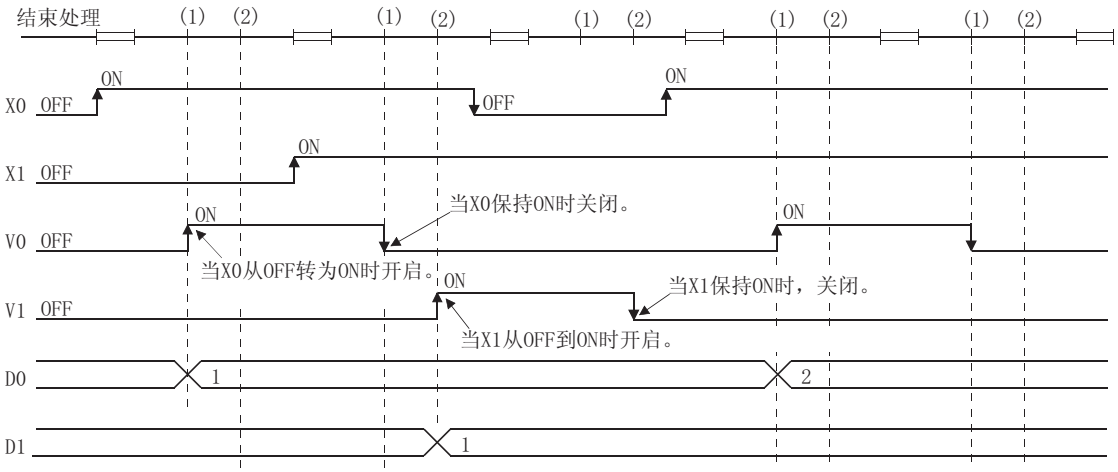
[梯形图模式]



[列表模式]

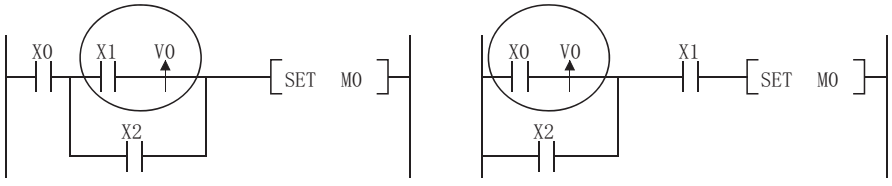
步	指令	软元件	
0	LD	SM400	
1	MOV	K0	Z0
4	CALL	P0	
6	MOV	K1	Z0
9	CALL	P0	
11	FEND		
12	P0		
13	LD	X0Z0	
14	EGP	V0Z0	
15	INC	D0Z0	
17	RET		
18	END		

[操作]



要 点

1. 因为 EGP 和 EGF 指令是按照在 EGP/EGF 指令之前的执行运算结果来执行的，因此这些指令必须在 AND 指令相同的位置上使用。
   (参见 5.1.1 项 .)
   EGP 和 EGF 指令不能在 LD 或 OR 指令的位置上使用。
2. EGP 和 EGF 指令不能在下面所示的梯形图块位置使用。



5.3 输出指令

5.3.1 输出指令（不包括定时器，计数器，和报警器）(OUT)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：要开启或关闭的软元件号（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				DY
Ⓓ	○（除T、C或F）		○				--		○

功 能

(1) 将 OUT 指令之前的运算结果输出到指定的软元件。

(a) 使用位软元件时

运算结果	线圈
OFF	OFF
ON	ON

(b) 字软元件的位指定时

运算结果	指定位
OFF	0
ON	1

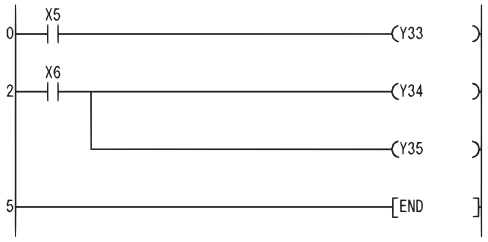
运算错误

(1) 没有与 OUT 指令相关的运算错误。

程序示例

(1) 当位软元件被使用时

[梯形图模式]

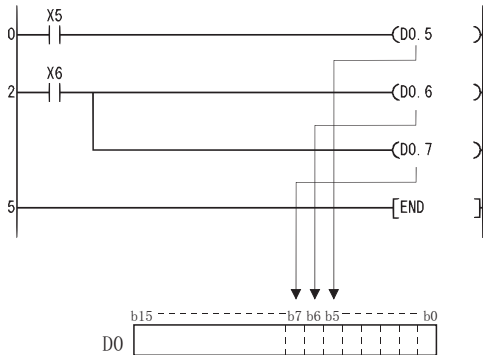


[列表模式]

步	指令	软元件
0	LD	X5
1	OUT	Y33
2	LD	X6
3	OUT	Y34
4	OUT	Y35
5	END	

(2) 当字软元件的位指定已经完成

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X5
1	OUT	D0.5
2	LD	X6
3	OUT	D0.6
4	OUT	D0.7
5	END	

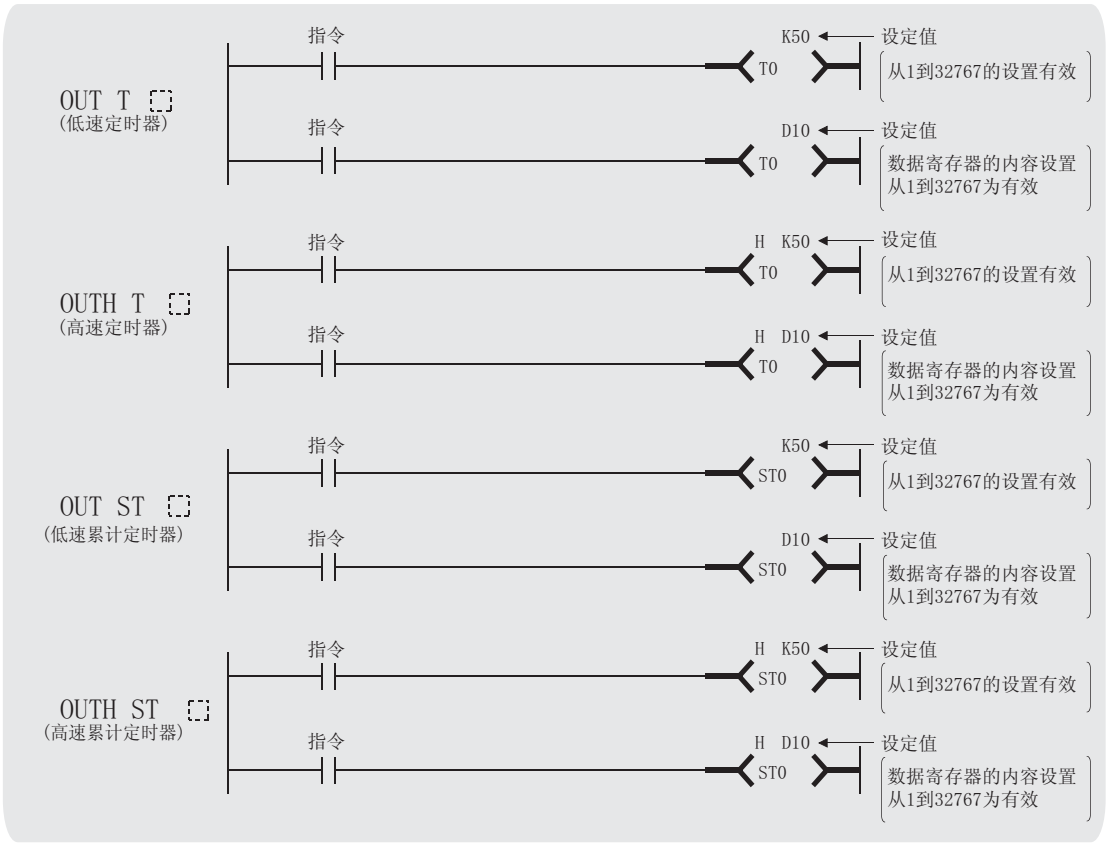
备 注

OUT 指令的基本步数如下：

- 使用内部软元件或文件寄存器 (R) 时 : 1
- 使用直接访问输出 (DY) 时 : 2
- 使用连号访问方式文件寄存器时
  (只对于通用型 QCPU) : 2
  (除通用型 QCPU 以外) : 3
- 使用除上述以外的其它软元件时 : 3

5.3.2 定时器 (OUT T、OUTH T)

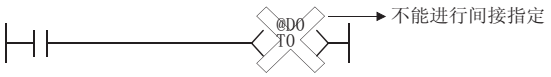
通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ　　：定时器号（位）
设定值　：定时器的设定值 (BIN16 位 *1)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K	其它
	位	字		位	字				
Ⓓ	○ (限于T)	--	--	--	--	--	--	--	--
设定值	--	○ (除T、C或F以外)	○	--	--	○	--	○ *2	--

*1: 定时器值的设置不能间接指定。



关于间接指定的更多信息，请参见 3.4 节。
*2: 定时器值的设置只能使用十进制常数 (K)。不能使用十六进制常数 (H) 和实数。

☆功 能

(1) 如果执行 OUT 指令之前的运算结果是 ON，那么定时器线圈为 ON 并且定时器计数到设定值；当到达时间到状态（总数值等于或比设定值大）时，触点应答如下：

A 触点	导通
B 触点	非导通

(2) 当到 OUT 指令的运算结果是从 ON 变换到 OFF 时，触点应答如下：

定时器类型	定时器线圈	定时器当前值	在时间到之前		时间到之后	
			A 触点	B 触点	A 触点	B 触点
低速定时器	OFF	0	非导通	导通	非导通	导通
高速定时器						
低速累计定时器	OFF	保持当前值	非导通	导通	导通	非导通
高速累计定时器						

(3) 通过使用 RST 指令在时间到之后，清除累计定时器的当前值并且将触点关闭。

(4) 负数（−32768 至 −1）不能在定时器中被设定为设定值。
如果设定值为 0，当时间 OUT 指令被执行时，定时器将会超时。

(5) 当 OUT 指令被执行时，下列处理会进行：

- OUT T□线圈变为 ON 或 OFF
- OUT T□触点变为 ON 或 OFF
- OUT T□当前值更新

如果在 OUT T□指令为 ON，JMP 指令或类似的指令被用来跳至 OUT T□指令时，不会有任何当前值更新或者进行触点 ON/OFF 操作。同样，如果在同一个扫描周期内同一 OUT T□指令被执行了两次或更多次，那么执行的重复次数的当前值将会被更新。

(6) 定时器的变址修饰只能被 Z0 或 Z1 执行。
对于定时器的设定值不能进行变址修饰。

备 注

1. 定时器的时限
定时器的时限可以在可编程控制器参数对话框的可编程控制器系统设置中设置。

定时器类型	QCPU		QnACPU	
	设定范围	设定单位	设定范围	设定单位
低速定时器 低速累计定时器	1 ms 到 1000 ms （默认值：100 ms）	1 ms	10 ms 到 1000 ms （默认值：100 ms）	10 ms
高速定时器 高速累计定时器	0.1 ms 到 100 ms （默认值：10 ms）	0.1 ms	10 ms 到 100 ms （默认值：10 ms）	1 ms

2. 关于定时器的计数方法，请参考以下手册。
- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础篇）

3. OUT T□指令的基本步数为 4 步。

运算错误

(1) 没有与 OUT T□指令相关的运行错误。

注意事项

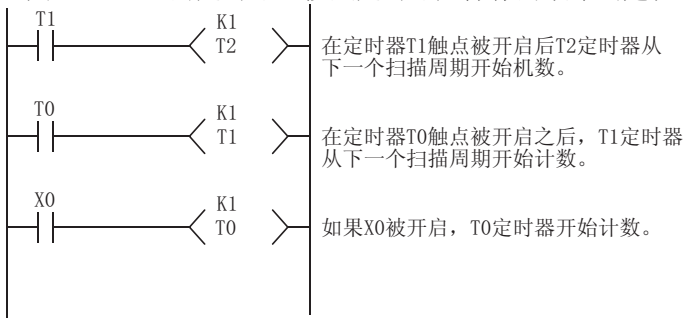
(1) 当创建的程序中定时器触点触发其他的定时器的操作，按照定时器的操作顺序创建程序－即为后操作的定时器先创建程序。

在下列情况中，如果程序是按照定时器操作的顺序来创建的，那么在相同的扫描周期内，所有的定时器均为 ON。

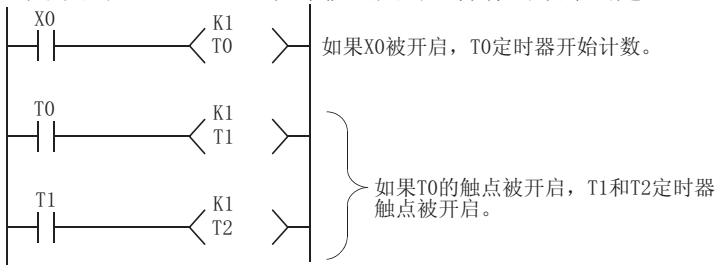
- 如果设定值小于一个扫描周期。
- 如果 “1” 被设定。

示例

- 对于 T0 至 T2 的定时器，按照定时器后操作的顺序创建程序。



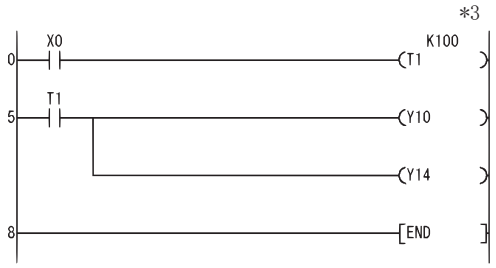
- 对于定时器 T0 至 T2，程序按照定时器操作的顺序创建。



程序示例

(1) 下列程序在 X0 已经接通后将 Y10 和 Y14 开启 10 秒。

[梯形图模式]



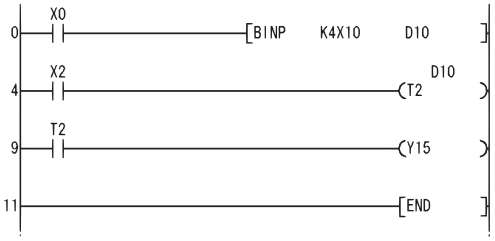
[列表模式]

步	指令	软元件	
0	LD	X0	
1	OUT	T1	K100
5	LD	T1	
6	OUT	Y10	
7	OUT	Y14	
8	END		

*3: 低速定时器的设定值表示它的默认时限 (100ms)

(2) 下列程序使用从 X10 至 X1F 的 BCD 数据作为定时器的设定值。

[梯形图模式]

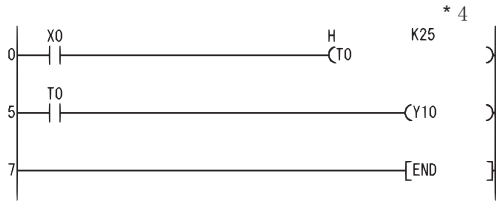


[列表模式]

步	指令	软元件	
0	LD	X0	
1	BINP	K4X10	D10
4	LD	X2	
5	OUT	T2	D10
9	LD	T2	
10	OUT	Y15	
11	END		

(3) 下列程序在 X0 接通后将 Y10 开启 250ms

[梯形图模式]



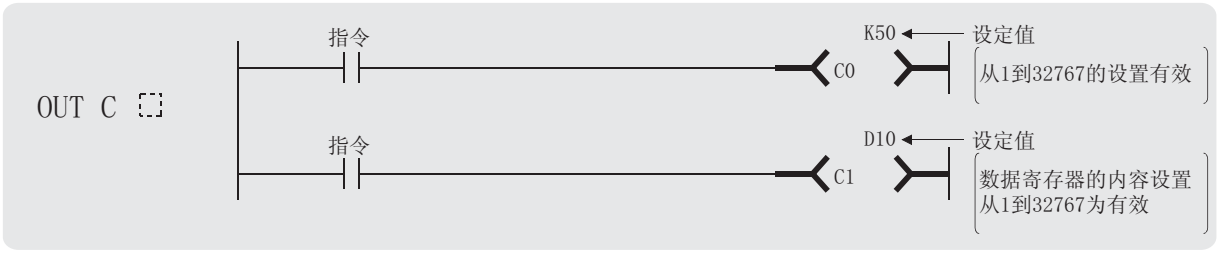
[列表模式]

步	指令	软元件	
0	LD	X0	
1	OUTH	T0	K25
5	LD	T0	
6	OUT	Y10	
7	END		

*4: 高速定时器的设定值表示其默认时限为 (10ms)

5.3.3 计数器 (OUT C)

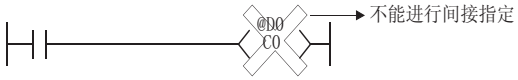
通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ　：计数器号（位）
设定值　：计数器的设定值 (BIN16 位 *1)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K	其它
	位	字		位	字				
Ⓓ	○（限于 C）	--	--	--	--	--	--	--	--
设定值	行	○（除 T、C 以外	○	--	○		--	○ *2	--

*1: 计数器值的设置不能间接指定。



关于间接指定的更多信息，请参见 3.4 节。

*2: 计数器值的设置只能为十进制常数 (K)。不能使用十六进制常数 (H) 和实数。

功 能

(1) 如果执行 OUT 指令之前的运算结果从 OFF 变为 ON，在当前值（计数值）上要加上 1，并且计数完成（当前值≧设定值）时，触点应答如下：

A 触点	导通
B 触点	非导通

(2) 在运算结果为 ON 时，不会进行任何计数。
（对于计数输出，没有必要进行脉冲变换）

(3) 如果达到计数值后，在 RST 指令执行之前，不改变计数值或触点。

(4) 定时器中设定值不能为负数（−32768 至 −1）。
如果设定值是 0，那么和 1 所进行的处理是一样

(5) 计数器线圈和触点的变址修饰只能使用 Z0 和 Z1。
不能为定时器设定执行变址修饰。

备 注

1. 关于计数器的计数方法，请参考以下手册。
   • QCPU 用户手册（功能解说 / 程序基础篇）
   • QnACPU 编程手册（基础篇）
2. OUT C□指令的基本步数为 4 步。

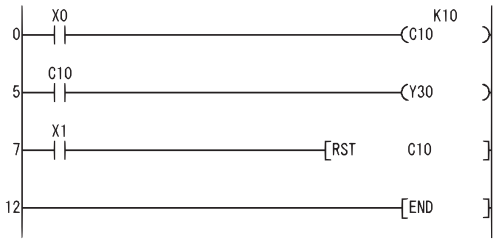
运算错误

(1) 没有与 OUT C□指令相关的运行错误。

程序示例

(1) 下列程序在 X0 接通 10 次后将 Y30 开启，当 X1 接通的时候复位计数器。

[梯形图模式]

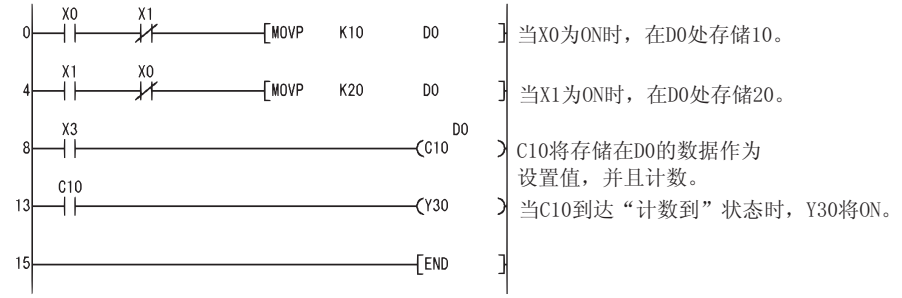


[列表模式]

步	指令	软元件	
0	LD	X0	
1	OUT	C10	K10
5	LD	C10	
6	OUT	Y30	
7	LD	X1	
8	RST	C10	
12	END		

(2) 下列程序当 X0 接通时在 10 的位置设定 C10 的值，当 X1 接通时在 20 的位置设定 C10 的值。

[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	ANI	X1	
2	MOVP	K10	D0
4	LD	X1	
5	ANI	X0	
6	MOVP	K20	D0
8	LD	X3	
9	OUT	C10	D0
13	LD	C10	
14	OUT	Y30	
15	END		

5.3.4 报警器输出 (OUT F)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ : 使其为 ON 的报警器号 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	○ (限于F)					--			

功 能

(1) 执行 OUT 指令之前的运算结果被输出到指定的报警器。

(2) 当报警器 (F) 开启的时候，出现下列应答 :

[使用 Q3A，Q4A，或 Q4ARCPU]

- 报警器号显示在 CPU 模块前面的 LED 显示软元件上，并且 “USER” LED 接通。
- 为 ON 的报警器号 (F 号) 储存在特殊寄存器里 (SD64 至 SD79)
- SD63 的值增加 1

[使用上述之外的 CPU 模块]

- “USER” / “ERR.” LED 接通。
- 为 ON 的报警器号 (F 号) 储存在特殊寄存器中 (SD64 至 SD79)。
- SD63 的值增加 1。

(3) 如果 SD63 的值是 16(当有 16 个报警器已经为 ON 时出现这种情况)，即使有一个新的报警器被开启，它的号也不会储存在 SD64 至 SD79 中。

(4) 当报警器被 OUT 指令关闭时，会出现以下反应。

[使用 Q3A、Q4A 或 Q4ARCPU 时]
线圈变为 OFF，但是 CPU 模块前面的 LED 显示器、“USER “LED 状态以及储存在 SD63 至 SD79 的内容不发生变化。
希望使 CPU 模块前面的 LED 显示器和 “USER “LED 熄灯，或从 SD63 至 SD79 中删除被 OUT F□指令关闭的报警器时，可使用 RST F□指令。

[使用上述之外的 CPU 时]
线圈变为 OFF，但是 “USER” / “ERR.” LED 的状态和储存在 SD63 至 SD79 中的内容不发生变化。
希望使 “USER” / “ERR” LED 熄灯，从 SD63 至 SD79 中删除被 OUT F□指令关闭的报警器时，可使用 RST F 指令。

运算错误

(1) 没有与 OUT F□指令有关的运行错误。

备 注

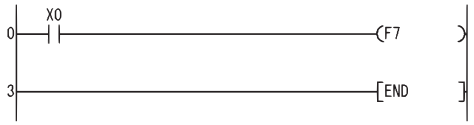
1. 关于报警器的详细内容，请参见以下手册。
   • QCPU 用户手册（功能解说 / 程序基础篇）
   • QnACPU 编程手册（基础篇）
2. OUT 模块 F□指令的基本步号为 2。
3. 下列表格显示哪一个 CPU 模块有在 CPU 模块正面的 LED 显示软元件或“USER” LED。

LED 类型	CPU 模块型号
LED 显示软元件	Q3A、Q4A、Q4AR
“USER” LED	Q2A(S1)、Q2AS(S1)、Q2ASH(S1)、高性能型 QCPU，过程 CPU，冗余 CPU、通用型 QCPU
“ERR.” LED	基本型 QCPU

程序示例

(1) 下列程序在 X0 接通的时候开启 F7，并且在从 SD64 至 SD79 上存储值 7。

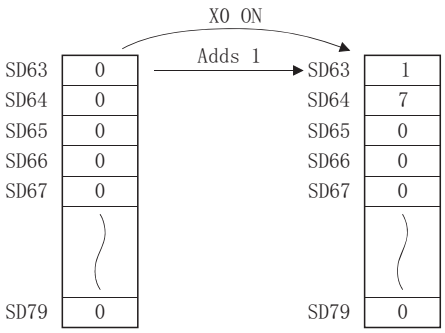
[梯形图模式]



[列表模式]

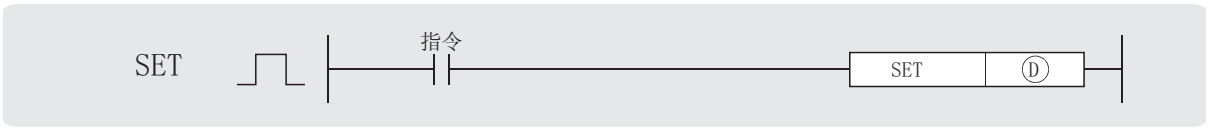
步	指令	软元件
0	LD	X0
1	OUT	F7
3	END	

[操作]



5.3.5 设定软元件（报警器除外）(SET)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：待设定 (ON) 的位软元件号 / 字软元件的位指定（位）

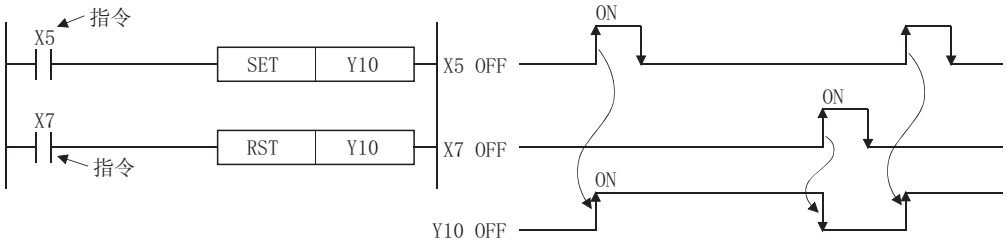
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				BL、DY
Ⓓ	○	○（除 T、C 以外）		○			--		○

功 能

(1) 当执行指令为 ON 时，指定软元件的状态如下：

软元件	软元件状态
位软元件	线圈和常数开启
当字软元件的位指令已经完成	指定位置为 1

(2) 即使 SET 输入变为 OFF 时，开启的软元件将仍然保持 ON 的状态。
被 SET 指令开启的软元件能够被 RST 指令关闭。



(3) 当执行命令为 OFF 时，软元件的状态不发生变化。

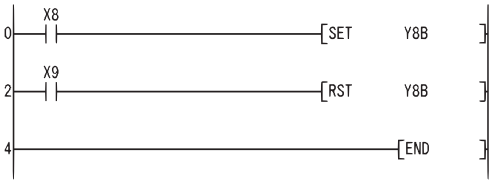
运算错误

(1) 没有与 SET 指令相关的运行错误。

程序示例

(1) 下列程序在 X8 接通时设定 Y8B(ON)，并且在 X9 接通的时候重置 Y8B(OFF)。

[梯形图模式]

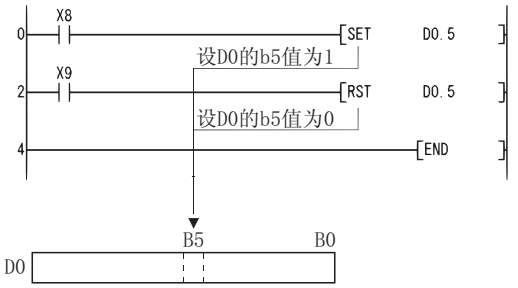


[列表模式]

步	指令	软元件
0	LD	X8
1	SET	Y8B
2	LD	X9
3	RST	Y8B
4	END	

(2) 下列程序在 X8 接通时设定 D0 位 (b5) 的值至 1，并且在 X9 接通时，设定位值至 0。

[梯形图模式]



[列表模式]

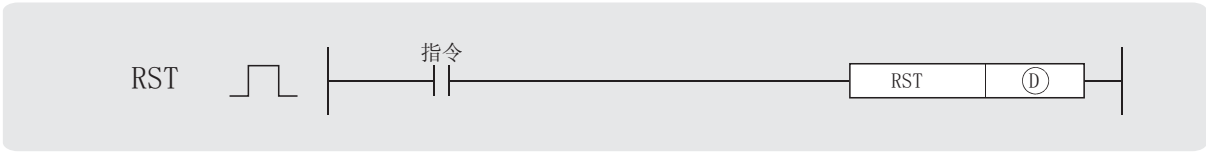
步	指令	软元件
0	LD	X8
1	SET	D0.5
2	LD	X9
3	RST	D0.5
4	END	

备 注

1. SET 指令的基本步数如下：
   - 使用内部软元件或文件寄存器 (R0 至 R32767) 时 : 1
   - 使用直接访问输出 (DY) 或 SFC 程序软元件 (BL) 时 : 2
   - 使用连号访问方式文件寄存器时
     (只对于通用型 QCPU) : 2
     (除通用型 QCPU 以外) : 3
   - 使用除上述以外的其它软元件时 : 3
2. 当 X 被当作软元件使用时，应使用实际输入中未使用的软元件号。如果使用了与实际输入软元件相同的号，那么实际输入的数据将被 SET 指令中指定的输入 X 所覆盖。

5.3.6 复位软元件（除了报警器）(RST)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：待复位的位软元件号 / 字软元件的位指定（位）。
待复位的字软元件号 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□¥□		U□¥G□	Zn	常数	其它
	位	字		位	字				DY
Ⓓ	○							--	○

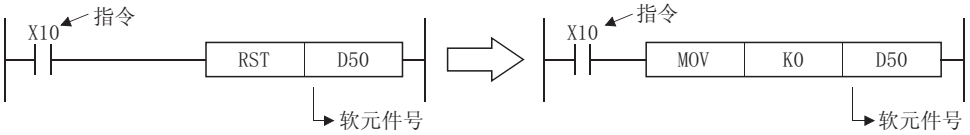
功 能

(1) 当执行命令为 ON 时，指定软元件的状态如下：

软元件	软元件状态
位软元件	将线圈和触点关闭
定时器和计数器	将当前值设定位 0，并且将线圈和触点关闭。
当字软元件的位指定已经完成	设定指定位的值到 0
除定时器和计数器之外的字软元件	将触点设置到 0

(2) 当 RST 输入为 OFF，软元件状态不改变。

(3) 被 RST 指令指定的字软元件功能等同于下列梯形图：



运算错误

(1) 没有与 RST 指令相关的运行错误。

备 注

RST 指令的基本步数如下。

a) 对于位处理

- 内部软元件（位软元件或字软元件的位指定） : 1
- 直接访问输出 : 2
- 定时器、计数器 : 4
- 使用连号访问方式文件寄存器时
  （只对于通用型 QCPU） : 2
  （除通用型 QCPU 以外） : 3
- 除上述之外 : 3

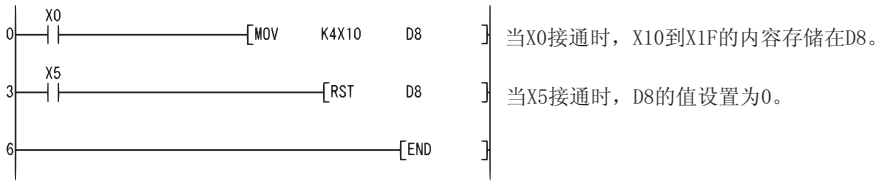
b) 对于字处理

- 内部软元件 : 2
- 变址寄存器 : 2
- 使用连号访问方式文件寄存器时
  （只对于通用型 QCPU） : 2
  （除通用型 QCPU 以外） : 3
- 除上述之外 : 3

程序示例

(1) 下列程序将数据寄存器的值设定为 0。

[梯形图模式]

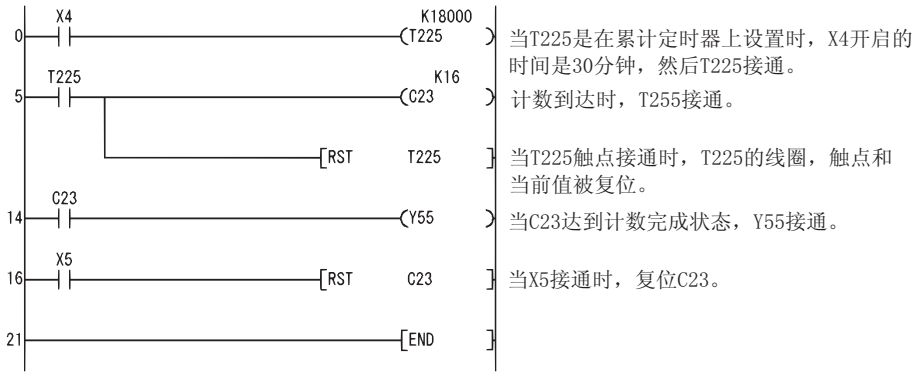


[列表模式]

步	指令	软元件
0	LD	X0
1	MOV	K4X10 D8
3	LD	X5
4	RST	D8
6	END	

(2) 下列程序复位 100ms 累计定时器和计数器。

[梯形图模式]

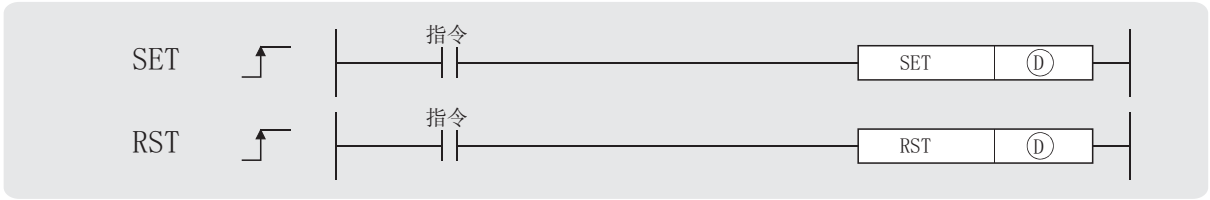


[列表模式]

步	指令	软元件	
0	LD	X4	
1	OUT	T225	K18000
5	LD	T225	
6	OUT	C23	K16
10	RST	T225	
14	LD	C23	
15	OUT	Y55	
16	LD	X5	
17	RST	C23	
21	END		

5.3.7 设定和复位报警器 (SET F、RST F)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



SET Ⓓ : 待设定的报警器号 (F 号)(位)

RST Ⓓ : 待复位的报警器号 (F 号)(位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	○ (限于 F)		--						

功 能

SET

(1) 当 SET 输入接通时被 Ⓓ 指定的报警器接通。

(2) 当报警器 (F) 接通时出现下列应答：

[使用 Q3A、Q4A 或 Q4ARCPU]

- 报警器号显示在 CPU 模块前面的 LED 显示软元件上，或 “USER” LED 亮。
- 开启的报警器号 (F 号) 连续地存储在特殊寄存器里 (SD64 至 SD79)。
- SD63 的值增加 1。

[使用 Q2A(S1)、Q2AS(S1)、Q2ASH(S1) 或 QCPU]

- “USER” LED 亮。*1
- 开启的报警器号 (F 号) 被连续地存储在特殊寄存器里 (SD63 至 SD79)。
- SD63 的值增加 1。

*1: 当使用基本型 QCPU，“ERR.” LED 变为 ON。

(3) 如果 SD63 的值为 16(当有 16 个报警器已经为 ON 时出现这种情况)，即使有新的报警器被开启，它的号也不会被 SD64 至 SD79 存储。

RST

(1) 当 RST 输入接通时，被 Ⓓ 指定的报警器被关闭。

(2) 已经断开的报警器的报警器号从特殊寄存器上 (SD64 至 SD79) 删除，并且 SD63 的值减小 1。

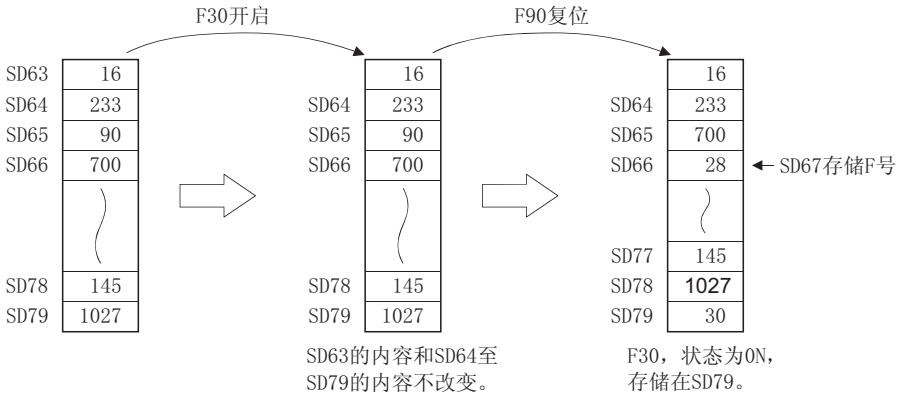
备 注

1. 关于报警器的详细内容，请参见以下手册。
   - QCPU 用户手册 (功能解说 / 程序基础篇)
   - QnACPU 编程手册 (基础篇)
2. SET F□ 、RST F□ 指令的基本步数为 2。

(3) 如果，当 SD63 的值为 16 时，通过使用 RST 指令将报警器号从 SD64 至 SD79 上删除，报警器号没有在 SD64 至 SD79 上登录的报警器将会开启，这些报警器号也将被登录。
如果所有 SD64 至 SD79 上的报警器号都被关闭，在 CPU 模块前端的 LED 显示软元件，或“USER”LED，将会被关闭。*2

*2: 当使用基本型 QCPU，“ERR.”LED 变为 OFF。

[当 SD63 为 16 时将会发生的操作]



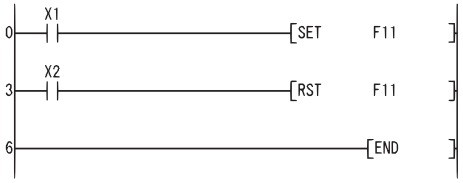
运算错误

(1) 没有与 SET F□ 或 RST F□ 指令相关的运行错误。

程序示例

(1) 下列程序在 X1 接通时将报警器 F11 开启，并且在特殊寄存器 (SD64 至 SD79) 上存储值 11。
然后，当 X2 接通时将报警器 F11 复位，并且从特殊寄存器 (SD64 至 SD79) 上删除值 11。

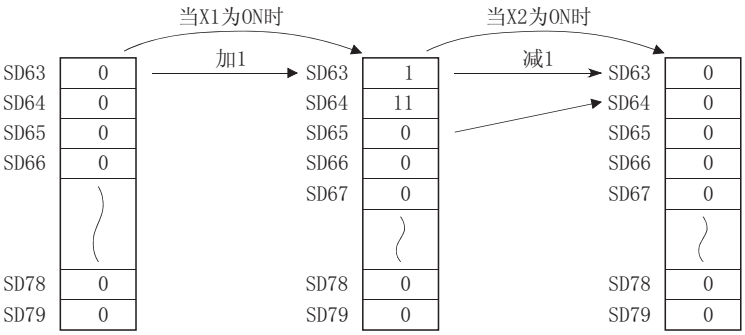
[梯形图模式]



[列表模式]

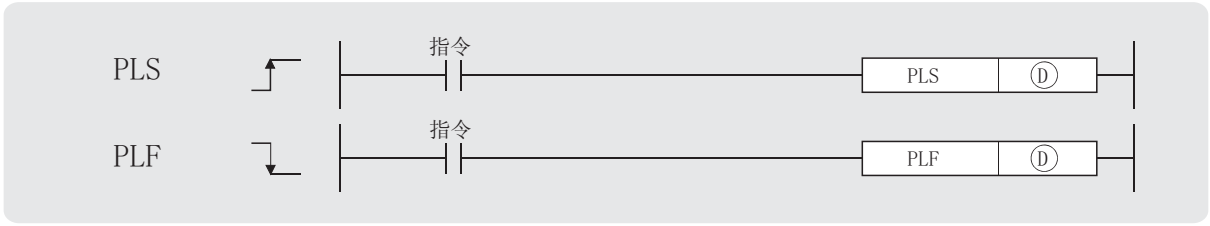
步	指令	软元件
0	LD	X1
1	SET	F11
3	LD	X2
4	RST	F11
6	END	

[动作]



5.3.8 上升沿和下降沿输出 (PLS、PLF)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ : 脉冲转换软元件（位）

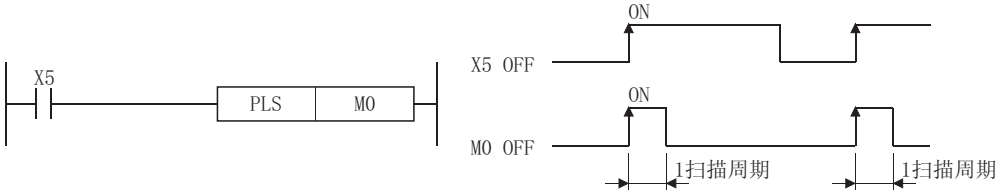
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				DY
Ⓓ	○						--		○

功 能

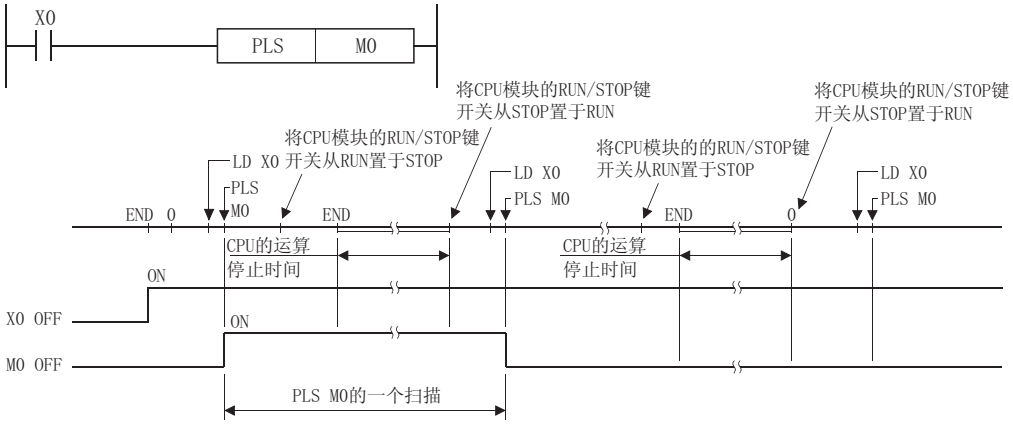
PLS

(1) 当从关闭转为开启时，PLS 指令将指定的软元件开启，在除从关闭到开启以外的其他情况下（如从 ON 到 ON，从 ON 到 OFF 或从 OFF 到 OFF），将指定的软元件关闭。

住一个扫描周期内，被 Ⓓ 指定的软元件中有一个 PLS 指令，那么这个指定的软元件在扫描周期内开启。
如果在一个扫描周期内对同一个软元件执行了多于一次的 PLS 指令，这种情况执行的操作，请参见 3.9 节。



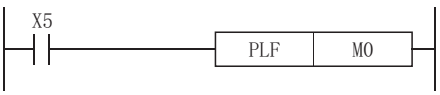
(2) 如果在执行了 PLS 指令之后，RUN/STOP 键开关从 RUN 变换到 STOP，那么即使开关被重新设置为 RUN，PLS 指令也不会再次执行。

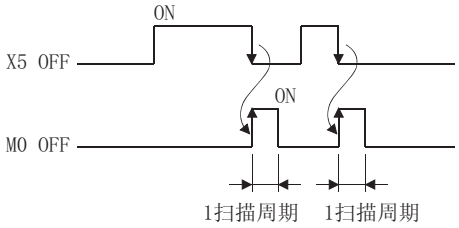


(3) 当锁存继电器 (L) 为 PLS 指令指定后，因为在第一次扫描中 PLS 指令从关闭变为开启，开启锁存继电器来关闭电源，然后再次开启电源，执行 PLS 指令来开启指定的软元件。
在电源在下一个 PLS 指令处有开启转换到关闭后，软元件在第一个扫描周期内开启。

PLF

(1) 当从开启变为关闭时，PLF 指令将指定的软元件开启，在除从开启到关闭以外的其他情况下 (例如，从 OFF 到 OFF，从 OFF 到 ON 或从 ON 到 ON)，该指令将指定的软元件关闭。
在一个扫描周期内被 Ⓓ 指定的软元件有一个 PLF 指令时，在该扫描周期内，开启指定的软元件。
如果同一个软元件在一个扫描周期内 PLF 指令执行多于一次，执行操作请参见 3.9 节。





(2) 如果在 PLF 指令执行之后，RUN/STOP 键开关从 RUN 变为 STOP，那么即使开关重新设置为 RUN，PLF 指令也不会被再次执行。

要 点

请注意如果 PLS 或 PLF 指令被 CJ 指令跳过或如果在 PLS/PLF 指令执行的位置的副程序没有被 CALL 指令回调，那么被 Ⓓ 指定的软元件开启时间会超过一个扫描周期。

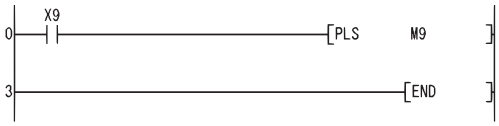
运算错误

(1) 没有与 PLS 或 PLF 指令相关的操作错误。

程序示例

(1) 当 X9 接通时，下列程序执行 PLS 指令。

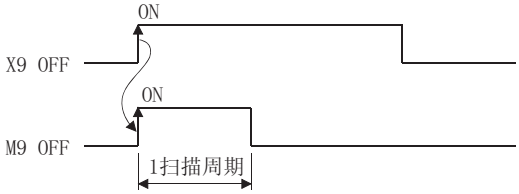
[梯形图模式]



[列表模式]

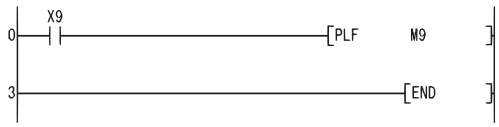
步	指令	软元件
0	LD	X9
1	PLS	M9
3	END	

[时序图]



(2) 当 X9 断开的时候，下列程序执行 PLF 指令。

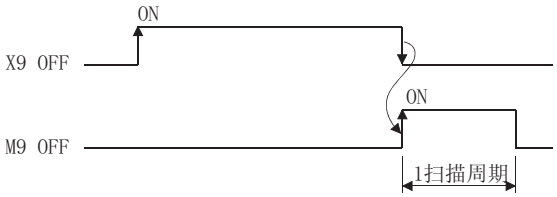
[梯形图模式]



[列表模式]

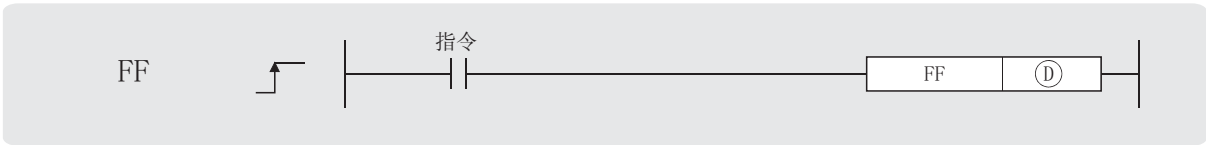
步	指令	软元件
0	LD	X9
1	PLF	M9
3	END	

[时序图]



5.3.9 位软元件输出取反 (FF)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：待取反的软元件号（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				DY
Ⓓ	○						--		○

功 能

(1) 当取反指令从 OFF 变为 ON 时，被 Ⓓ 指定的软元件状态被反转。

软元件	软元件状态	
	在 FF 执行之前	在 FF 执行之后
位软元件	OFF	ON
	ON	OFF
字软元件位指定	0	1
	1	0

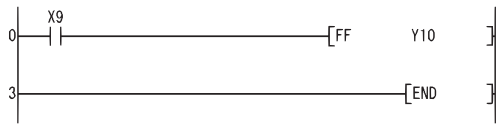
运算错误

(1) 没有与 FF 指令相关的运行错误。

程序示例

(1) 在 X9 接通时，下列程序将 Y10 的输出取反。

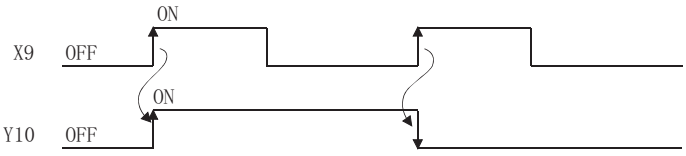
[梯形图模式]



[列表模式]

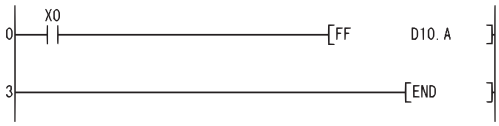
步	指令	软元件
0	LD	X9
1	FF	Y10
3	END	

[时序图]



(2) 当 X0 接通时，下列程序将 D10 的 b10（位 10）取反。

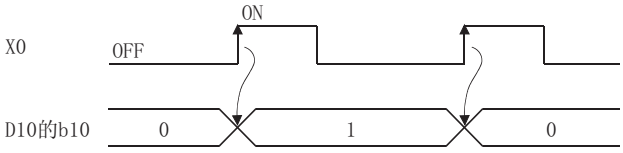
［梯形图模式］



［列表模式］

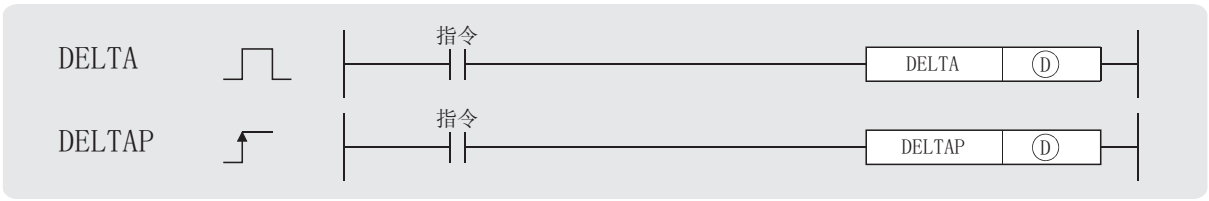
步	指令	软元件
0	LD	X0
1	FF	D10.A
3	END	

［时序图］



5.3.10 直接输出的脉冲变换 (DELTA(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ : 将要进行脉冲转换的位 (位)。

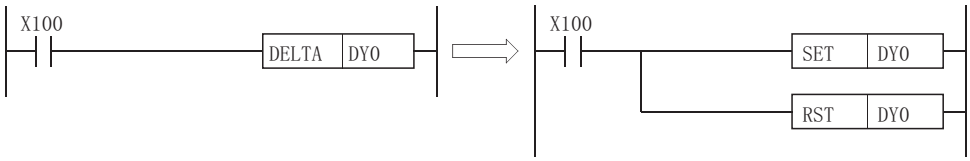
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				DY
Ⓓ	--								○

功 能

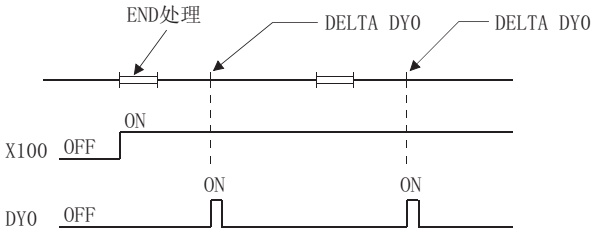
(1) 产生被 Ⓓ 指定的直接访问输出 (DY) 的脉冲输出。

如果 DELTA DY0 已经被指定，那么导致的操作将会等同于下面所示的梯形图，该梯形图使用 SET/RST 指令。

[被 DELTA 指令创建的梯形图]　　[使用 SET/RST 指令的梯形图]



[动作]



(2) DELTA(P) 指令是被智能模块 / 特殊功能模块的上升沿执行使用的指令。

运算错误

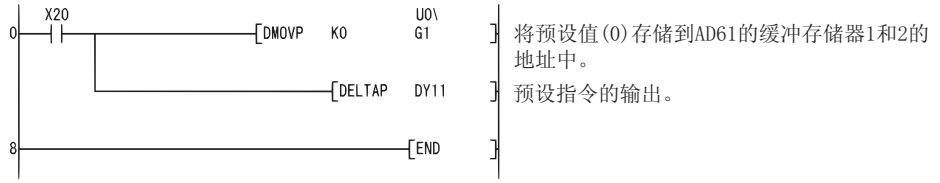
(1) 在下列情况下，会出现一个运行错误，错误标志 (SM0) 开启，并且出错代码存储在 SD0。

• 被 Ⓓ 指定的直接访问输出号超出了 CPU 模块输出范围。　　(出错代码 : 4101)

程序示例

(1) 当 X20 接通的时候，下列程序预置安装在主基板的 0 插槽的 AD61 的 CH1。

[梯形图模式]



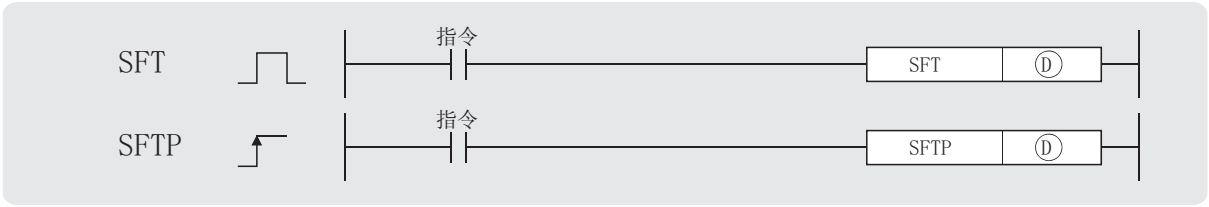
[列表模式]

步	指令	软元件	
0	LD	X20	
1	DMOVP	K0	U0¥G1
6	DELTAP	DY11	
8	END		

5.4 移动指令

5.4.1 位软元件移动 (SFT(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ：要移动的软元件号（位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它 DY
	位	字		位	字				
Ⓓ	○（除 TC 以外）						--		○

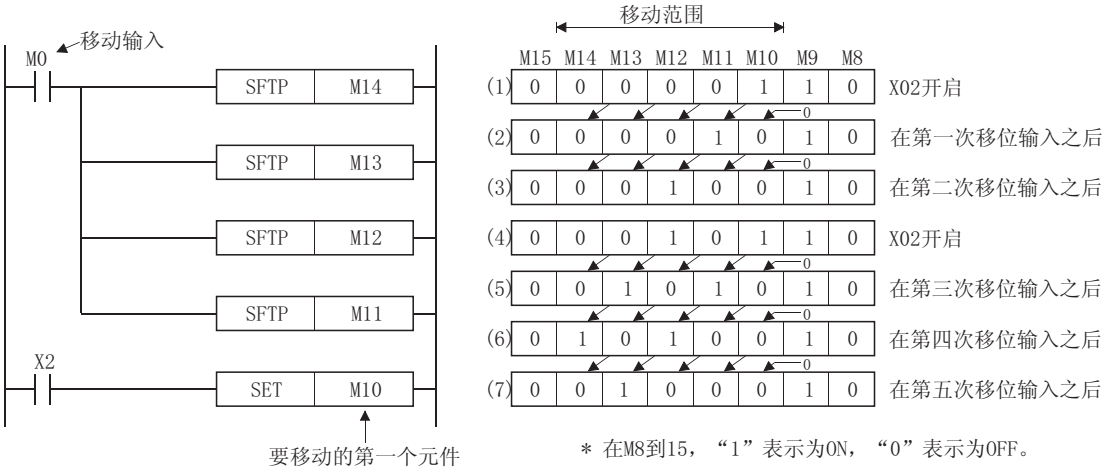
功 能

(1) 当使用位软元件时

(a) 在指定的软元件之前立即将软元件的 ON/OFF 状态移动到被 Ⓓ 指定的软元件中去，并且将先前的软元件关闭。
例如，如果 M11 已经被 SFT 指令指定，当 SET 指令被执行，那么它将会移动 M10 的 ON/OFF 状态至 M11，并且将 M10 关闭。

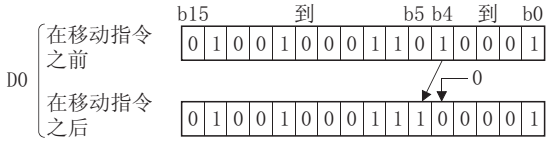
(b) 使用 SET 指令将要移动的第一个软元件开启。

(c) 当 SFT 和 SFTP 被连续使用时，将从有较大号的软元件处启动程序。



(2) 当使用字软元件的位指定

(a) 在指定的软元件之前直接将位的 1/0 状态移动到被 Ⓓ 指定的软元件的位中，并且将之前的位设置为 0。
例如，如果 D0.5(D0 的位 5[b5]) 已经被 SFT 指令指定，当这个 SFT 指令被执行时，将会移动 D0 的 b4 的 1/0 状态至 b5，并且将 b4 设置为 0。



运算错误

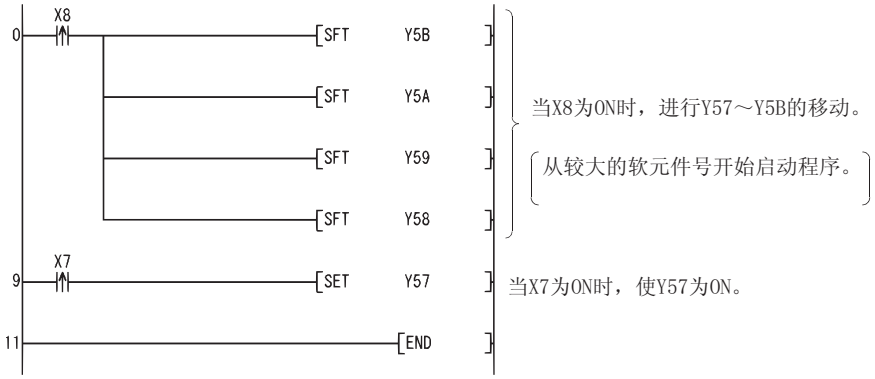
(1) 在下列情况下，将发生运算错误，错误标志 (SM0) 将 ON，出错代码将存储在 SD0 中。

•Ⓓ 指定的软元件超出了相应软元件的范围。
(只对于通用型 QCPU。) (出错代码： 4101)

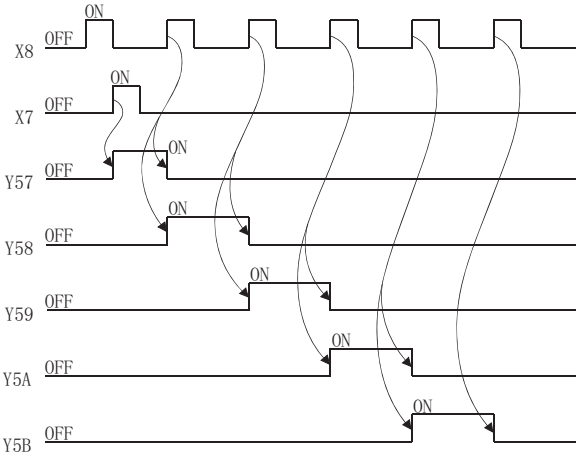
程序示例

(1) 当 X8 接通的时候，下列程序将 Y57 移动到 Y5B。

[梯形图模式]



[时序图]



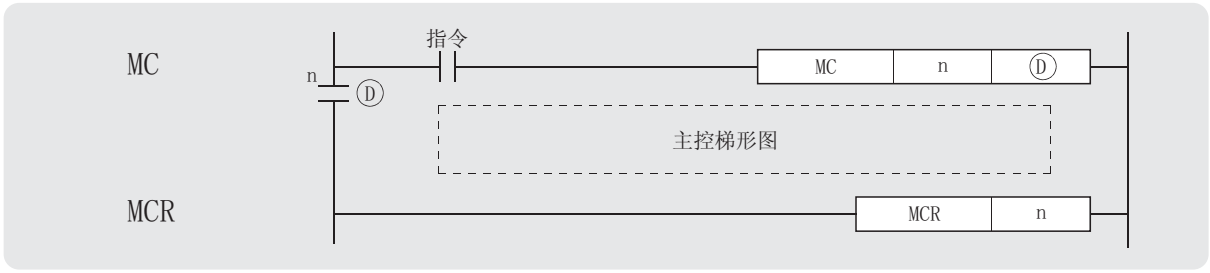
[列表模式]

步	指令	软元件
0	LDP	X8
1	SFT	Y5B
3	SFT	Y5A
5	SFT	Y59
7	SFT	Y58
9	LDP	X7
10	SET	Y57
11	END	

5.5 主控指令

5.5.1 设置和复位主控 (MC、MCR)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n ：嵌套 (N0 至 N14)（嵌套）

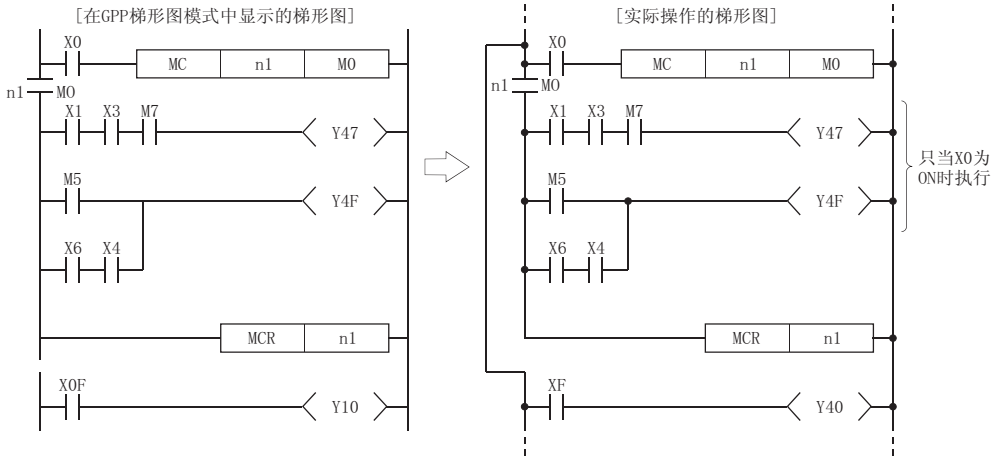
Ⓓ ：待开启的软元件号（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它	
	位	字		位	字				N	DY
n	--						--		○	--
Ⓓ	○						--		--	○

功 能

通过为梯形图开启或关闭一个公共总线，主控指令被用来生成高效的梯形图顺控程序。

使用主控的梯形图，如下所示：



备 注

当在一个外围设备中的梯形图模式中编程时，不需要在垂直母线上输入触点。在生成了梯形图之后，执行 “变换”指令时将会自动显示，然后被设置成 “读取”模式。

MC

(1) 如果当主控开始时，MC 指令的 ON/OFF 指令是 ON，那么在 MC 指令和 MCR 指令之间的运算结果将正好和所示的指令（梯形图）一样。

如果 MC ON/OFF 显示为 OFF，那么在 MC 和 MCR 指令之间的运算结果将如下所示：

软元件	软元件状态
高速定时器 低速定时器	计数器值变为 0，线圈和触点都断开。
高速累计定时器 低速累计定时器 计数器	线圈断开，但计数器的数值和触点都保持当前状态。
在 OUT 指令中的软元件	全部关闭
SET、RST SFT 基本，应用 } 在下列指令中的软元件	保持当前状态

(2) 当 MC 指令为 OFF 时，从 MC 指令到 MCR 指令的指令都将会被执行，所以扫描周期时间将不会缩短。

要 点

如果在梯形图中有不需要的触点指令（FOR - NEXT、EI、DI、等）使用了主控，那么不管 MC 指令的 ON/OFF 处在什么状态，CPU 模块都会执行这些指令。

(3) 通过改变被 Ⓓ 指定的软元件，MC 指令能够任意次使用同一个嵌套号 (N)。

(4) 当 MC 指令是 ON 时，被 Ⓓ 指定的软元件线圈被开启。

并且，在 OUT 指令或其他指令中使用这些相同的软元件将会使线圈变成双重线圈，所以被 Ⓓ 指定的软元件不应在其他指令中使用。

MCR

(1) 这是从主控中恢复的指令，并且显示操作主控范围的结束。

(2) 在 MCR 指令之前不要有触点指令。

(3) 使用具有相同的嵌套号的 MC 指令和 MCR 指令作为配套。

然而，当 MCR 指令在一个位置嵌套时，所有的主控都能因使用最低的嵌套 (N) 号而终止。（参见程序范例中的 “嵌套注意事项”）

运算错误

(1) 没有与 MC 或 MCR 指令相关的运行错误。

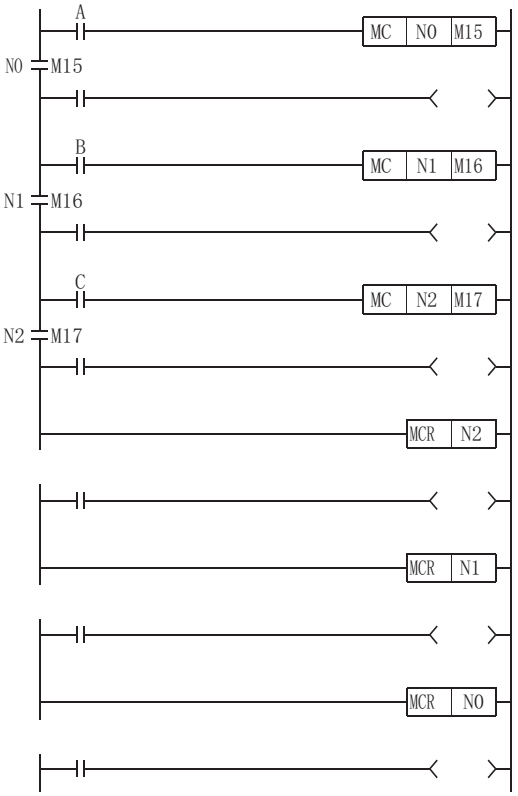
程序示例

主控指令能在嵌套中使用。

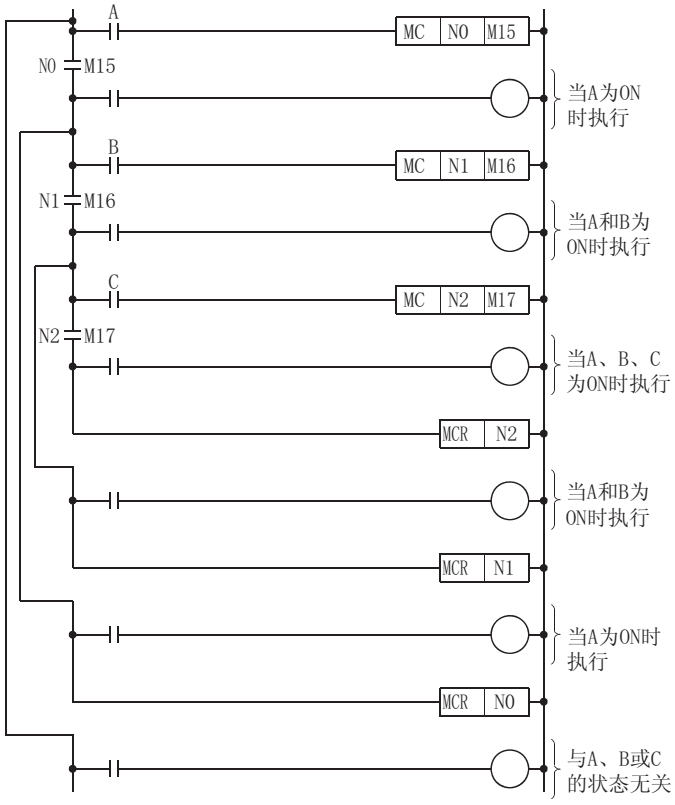
嵌套 (N) 把不同的主控区域区分开来。嵌套能在 N0 到 N14 上执行。

使用嵌套能够生成梯形图，这些梯形图连续地限制程序的执行条件。有嵌套的梯形图 如下所示

[在 GPP 梯形图模式中显示的梯形图]



[实际操作的梯形图]



使用嵌套结构的注意事项

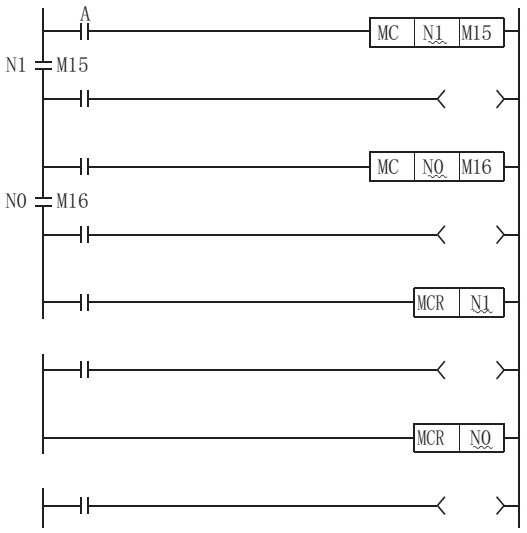
(1) 嵌套能使用达 15 次 (N0 至 N14)

使用嵌套时，在 MC 指令中嵌套是按嵌套号 (N) 从低到高插入的，在 MCR 指令中是按嵌套号从高到低插入的。

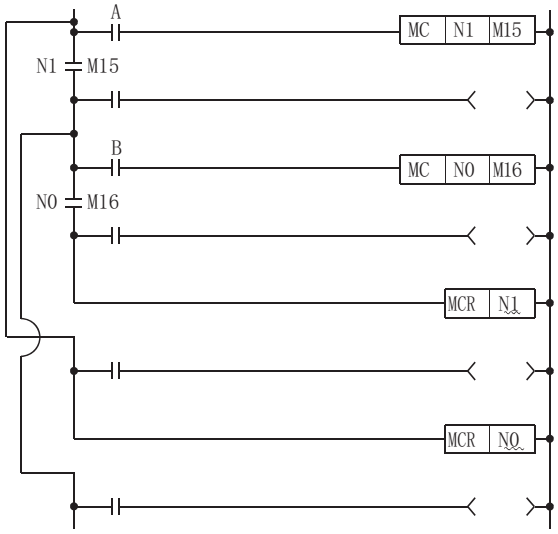
如果该顺序逆向了，将不会有嵌套结构，并且 CPU 模块也不能进行正确的操作。

例如，如果嵌套是被 MC 指令从 N1 到 N0 的顺序指定的，并且同样被 MCR 指定从 N1 到 N0 的顺序指定，那么垂直母线将会交叉并且不会产生正确的主控梯形图。

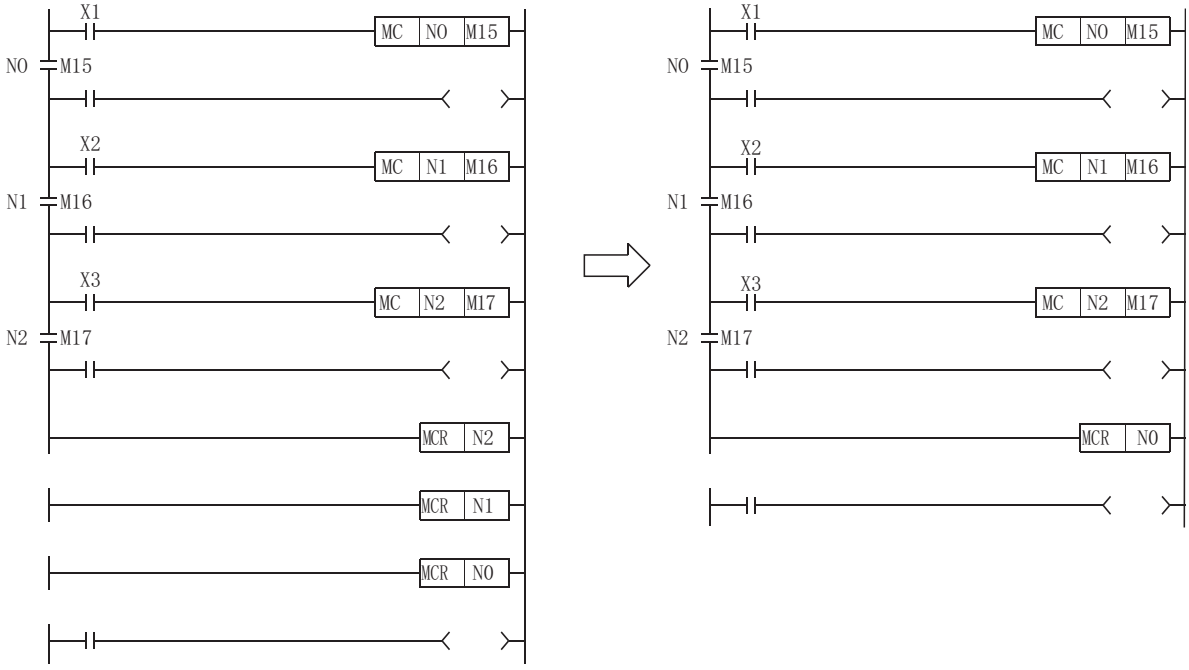
[在 GPP 梯形图模式中显示的梯形图]



[实际操作的梯形图]



(2) 如果嵌套结构导致了 MCR 指令集中在一个位置，那么所有的主控均会因使用最低的嵌套号 (N) 而被终止。

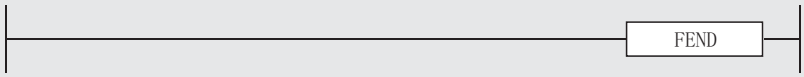


5.6 结束指令

5.6.1 结束主程序 (FEND)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

FEND



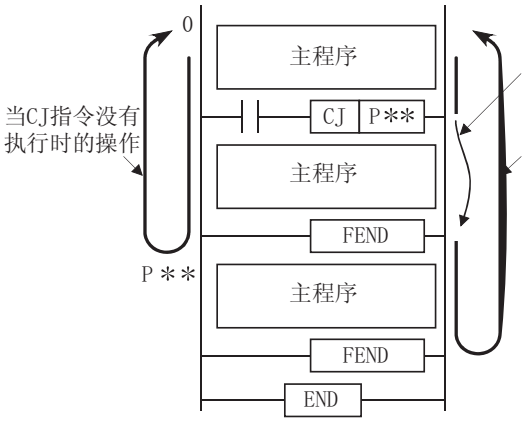
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

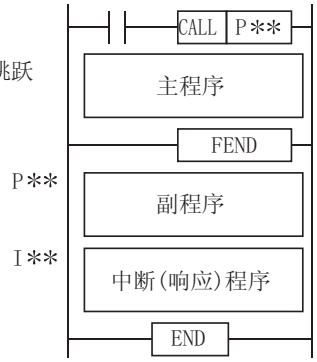
(1) 当 CJ 指令或其他指令被用来在顺控程序操作中创建一个分支时，在这种情况下使用 FEND 指令。并且当主程序从副程序或暂停（响应）程序中分离出来时，在这种情况下也使用 FEND 指令。

(2) FEND 指令的执行将会使得 CPU 模块结束正在执行的程序。

(3) 在外围设备中甚至在 FEND 指令之后的顺控程序都能够在梯形图显示中显示出来。
（外围设备持续显示梯形图直到遇到 END 指令）



(a) 当使用CJ指令时



(b) 当为副程序或中断(响应)程序时

运算错误

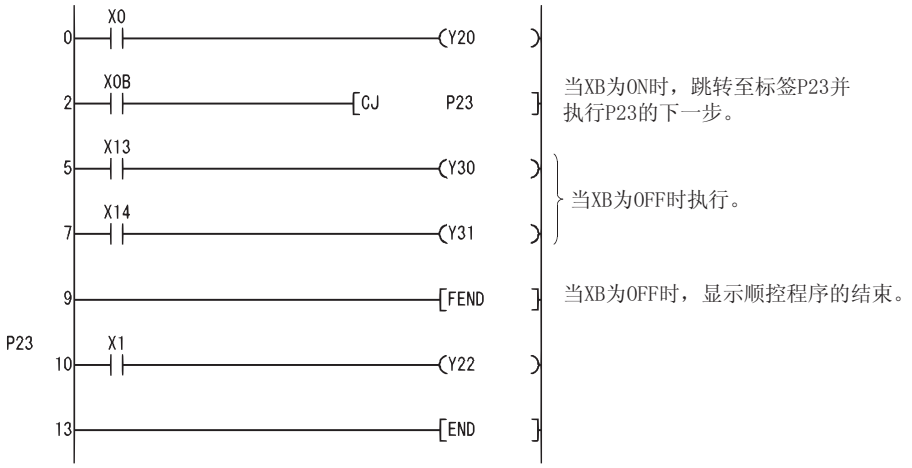
(1) 在下列情况下会出现运行错误，错误标志 (SM0) 开启，并且将出错代码存储在 SD0 中。

- FEND 指令在 CALL、FCALL、ECALL 或 EFCALL 指令执行之后，并且在 RET 指令之前执行。 (出错代码： 4211)
- FEND 指令在 FOR 指令执行之后且在 NEXT 指令执行之前执行。 (出错代码： 4200)
- FEND 指令在暂停 (响应) 程序中执行，且在 IRET 指令执行之前执行。 (出错代码： 4221)
- FEND 指令在 CHKCIR 和 CHKEND 指令之间执行。 (出错代码： 4230)
- FEND 指令在 IX 和 IXEND 指令之间执行。 (出错代码： 4231)

程序示例

(1) 下列程序使用 CJ 指令

[梯形图模式]

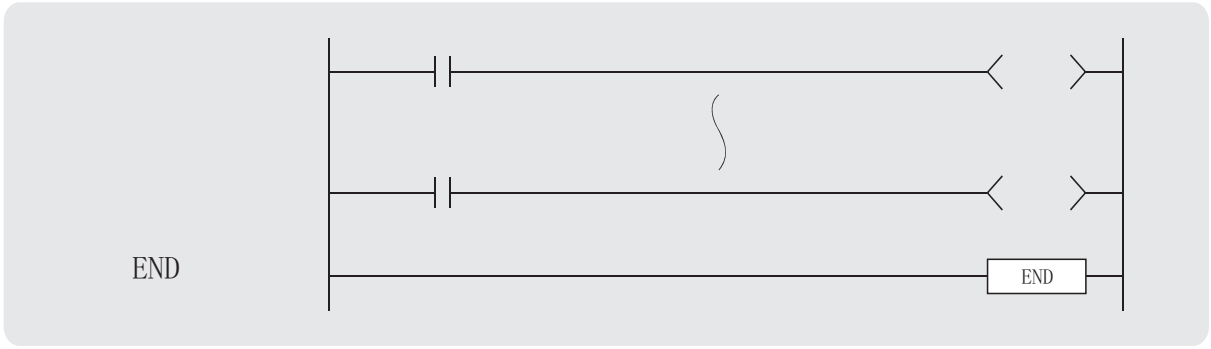


[列表模式]

步	指令	软元件
0	LD	X0
1	OUT	Y20
2	LD	X0B
3	CJ	P23
5	LD	X13
6	OUT	Y30
7	LD	X14
8	OUT	Y31
9	FEND	
10	P23	
11	LD	X1
12	OUT	Y22
13	END	

5.6.2 结束顺控程序 (END)

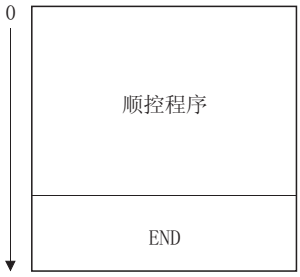
通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

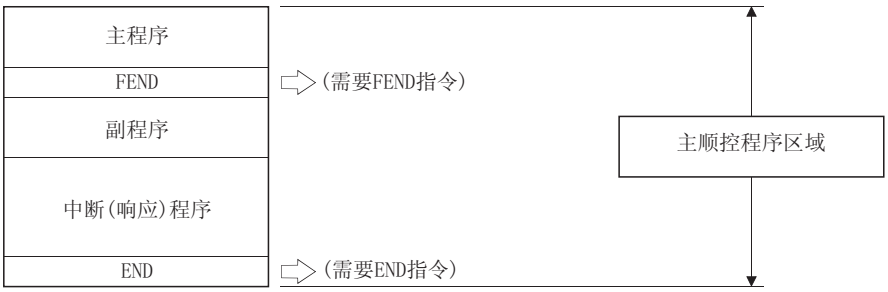
(1) 表示程序的结束，包括主程序，副程序和暂停（响应）程序。
END 指令的执行将会引起 CPU 模块结束正在执行的程序。



(2) 在主顺控程序的执行过程中不能使用 END 指令。
如果在程序执行过程中需要进行 END 处理，请使用 FEND 指令。

(3) 当在外围设备的梯形图模式中编程时，不需要输入 END 指令。

(4) 在主程序，副程序和暂停（响应）程序中，使用 END 和 FEND 指令被暂停如下：



运算错误

(1) 在下列情况下会出现运行错误，错误标志 (SM0) 开启，并且将出错代码存储在 SD0 处。

- END 指令在 RET 指令执行之前且在 CALL、FCALL、ECALL 或 EFCALL 指令执行之后执行。（出错代码： 4211）
- END 指令在 NEXT 指令执行之前且在 FOR 指令执行之后执行。（出错代码： 4200）
- END 指令在暂停（响应）程序中执行，且先于 IRET 指令的执行。（出错代码： 4221）
- END 指令在 CHKCIR 到 CHKEND 指令循环中执行。（出错代码： 4230）
- END 指令在 IX 到 IXEND 指令循环中执行。（出错代码： 4231）

5.7 其他指令

5.7.1 顺控程序暂停 (STOP)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

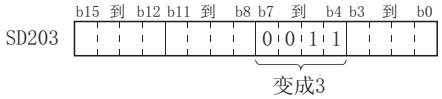


设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

(1) 当暂停输入被开启，输出 Y 被复位并且 CPU 模块的操作被停止。
（如果 RUN/STOP（键）开关被转为 STOP 设置那么会出现相同的结果。）

(2) 执行 STOP 指令将会使特殊寄存器 SD203 b4 到 b7 的值变为 3。



(3) 为了能在执行 STOP 指令之后重新执行 CPU 模块操作，应复位已经从 RUN 改变到 STOP 的 RUN/STOP 键开关，使其重新为 RUN。

运算错误

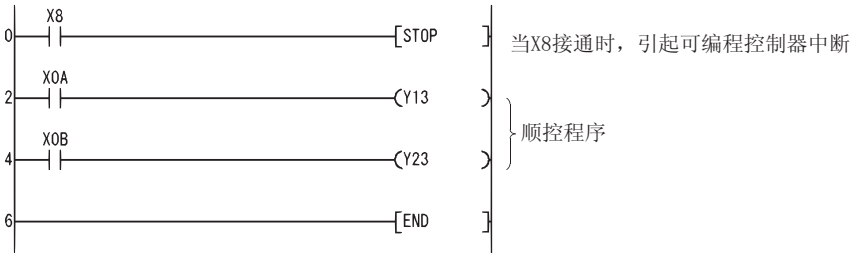
(1) 在下列情况下会出现运行错误，错误标志 (SM0) 开启，并且出错代码储存至 SD0。

- STOP 指令在 RET 指令执行之前且住 CALL/FCALL/ECALL/EFCALL/XCALL 指令执行之后执行。 (出错代码： 4211)
- STOP 指令在 NEXT 指令执行之前且在 FOR 指令执行之后执行。 (出错代码： 4200)
- STOP 指令在暂停 (响应) 程序中并且在 IRET 指令执行之前执行。(出错代码： 4221)
- STOP 指令在 CHKCIR 到 CHKEND 指令循环中执行。 (出错代码： 4230)
- STOP 指令在 IX 到 IXEND 指令循环中执行。 (出错代码： 4231)
- 在固定扫描执行类型程序中执行了 STOP 指令。 (只对于通用型 QCPU) (出错代码： 4223)

程序示例

(1) 当 X8 接通时，下列程序将 CPU 模块停止。

[梯形图模式]

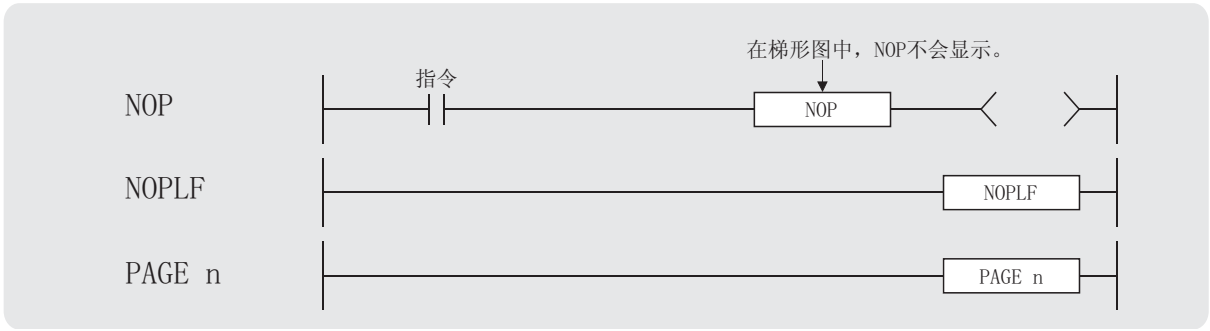


[列表模式]

步	指令	软元件
0	LD	X8
1	STOP	
2	LD	X0A
3	OUT	Y13
4	LD	X0B
5	OUT	Y23
6	END	

5.7.2 无处理 (NOP、NOPLF、PAGE n)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

NOP

(1) 这是一个无处理指令，到该点之前不会对任何处理有影响。

(2) NOP 指令适用于下列情况：

1) 为顺控程序调试插入空间。
2) 删除一个指令，但是不用改变步号（用 NOP 替换指令）
3) 临时性删除一个指令。

NOPLF

(1) 这是一个无处理指令，到该点之前不会对任何处理有影响。

(2) 使用 NOPLF 指令可以使从外围设备中打印的页在任何需要位置改变。

1) 当打印梯形图

- 如果有 NOPLF 指令，那么在梯形图块之间将会插入一个页面暂停。
- 如果 NOPLF 指令插在梯形图中间，那么梯形图将不能正确显示。不要在梯形图的中间插入 NOPLF 指令。

2) 当打印指令列表时

- 在打印 NOPLF 指令之后将会被换页。

(3) 关于外围设备的打印输出的详细资料，参见使用中的外围设备的操作手册。

PAGE n

(1) 这是一个无处理指令，对到此为止的运算不会产生任何影响。

(2) 是使外围设备也执行无处理的指令。

运算错误

(1) 没有与 NOP、NOPLF 或 PAGE 指令相关的错误。

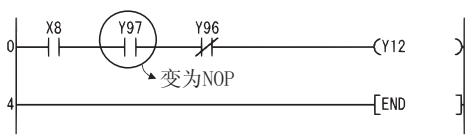
程序示例

NOP

(1) 触点关闭..... 删除 AND 或 AN1 指令

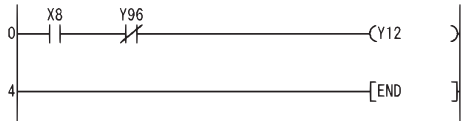
[梯形图模式]　　[列表模式]

变化前



步	指令	软元件
0	LD	X8
1	AND	Y97
2	ANI	Y96
3	OUT	Y12
4	END	

变化后

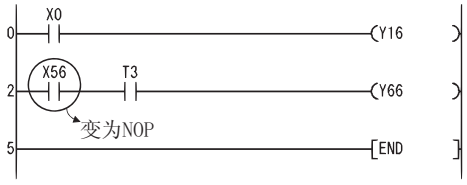


步	指令	软元件
0	LD	X8
1	NOP	
2	ANI	Y96
3	OUT	Y12
4	END	

(2) 触点关闭..... LD、LDI 变换至 NOP(请注意将 LD 和 LDI 完全改变为 NOP，改变了梯形图的性质)。

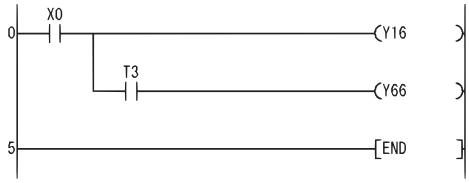
[梯形图模式]　　[列表模式]

变化前



步	指令	软元件
0	LD	X0
1	OUT	Y16
2	LD	X56
3	AND	T3
4	OUT	Y66
5	END	

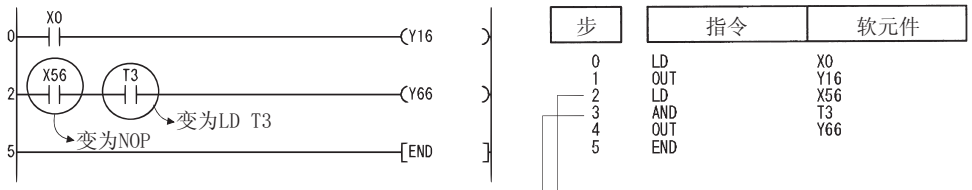
变化后



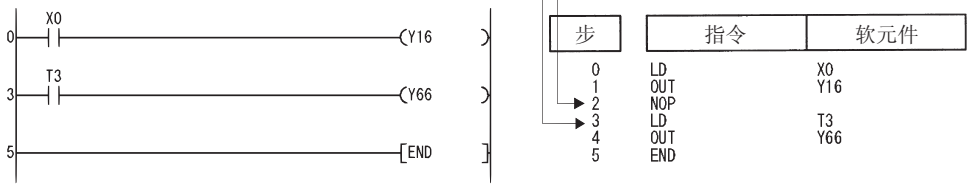
步	指令	软元件
0	LD	X0
1	OUT	Y16
2	NOP	
3	AND	T3
4	OUT	Y66
5	END	

［梯形图模式］　　［列表模式］

变化前

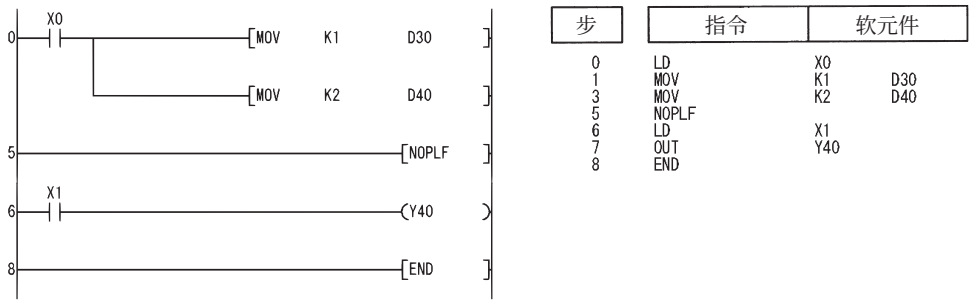


变化后

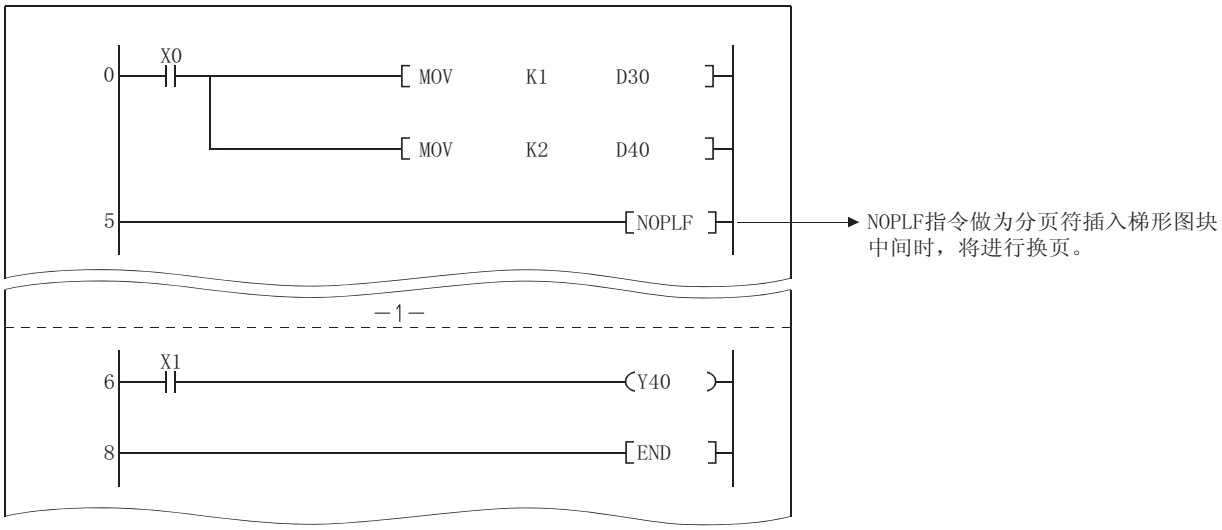


NOPLF

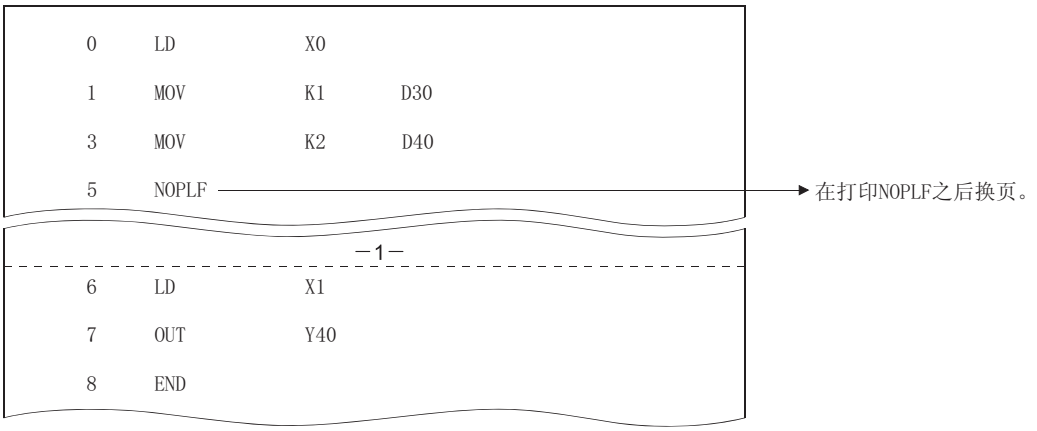
［梯形图模式］　　［列表模式］



• 打印梯形图将导致以下结果：

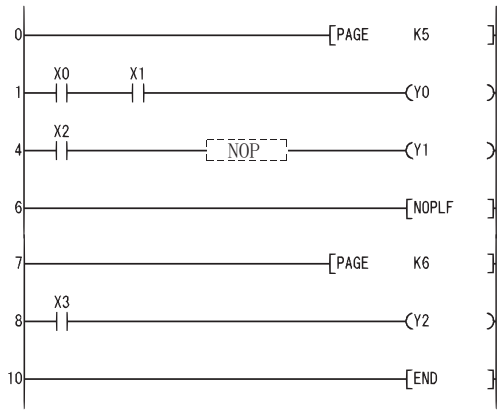


• 用 NOPLF 指令打印指令列表将会有如下结果：



PAGE n

［梯形图模式］



［列表模式］

步	指令	软元件
0	PAGE	K5
1	LD	X0
2	AND	X1
3	OUT	Y0
4	LD	X2
5	NOP	
6	OUT	Y1
7	NOPLF	
8	PAGE	K6
9	LD	X3
10	OUT	Y2
11	END	

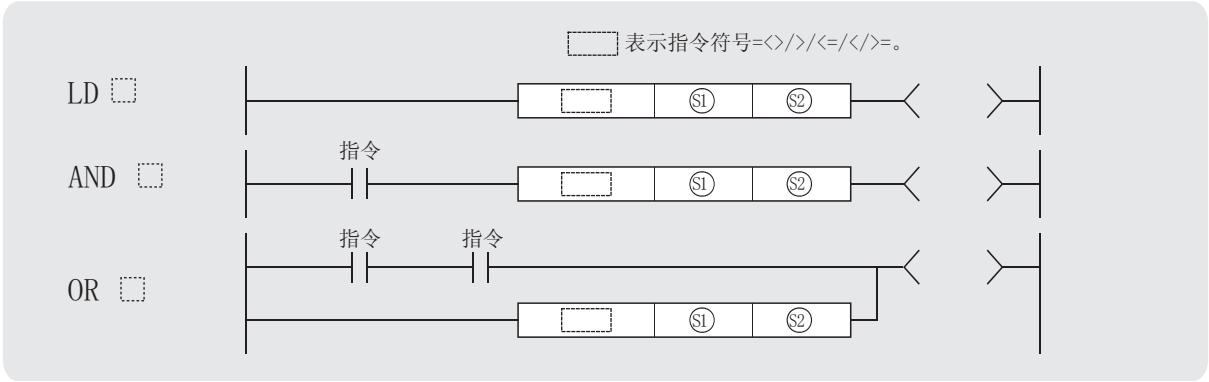
6 基本指令

分类	处理内容	参考章节
比较运算指令	数据与数据比较。	6.1 节
算术运算指令	数据与的数据加、减、乘、除、递增或递减。	6.2 节
数据转换指令	转换数据类型。	6.3 节
数据传送指令	传送指定的数据。	6.4 节
数据分支指令	程序跳转。	6.5 节
程序执行控制指令	允许 / 禁止程序中断。	6.6 节
I/O 刷新指令	刷新位软元件。	6.7 节
其它方便的指令	上 / 下计数器、教学定时器、特殊功能定时器、旋转台的就近控制等。	6.8 节

6.1 比较运算指令

6.1.1 BIN 16 位数据比较 (=、<>、>、<=、<、>=)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)、(S2) ：比较数据或者存储比较数据的软元件的起始号 (BIN16 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
(S1)				○					--
(S2)				○					--

功 能

(1) 把由 (S2) 指定的软元件的 BIN 16 位数据和由 (S1) 指定的软元件的 BIN 16 位数据作为 A 触点，并进行比较运算。

(2) 单个指令比较操作的结果如下：

指令符号 □	条件	比较操作结果	指令符号 □	条件	比较操作结果
=	(S2) = (S1)	导通	=	(S1) ≠ (S2)	不导通
<>	(S1) ≠ (S2)		<>	(S2) = (S1)	
>	(S1) > (S2)		>	(S1) ≦ (S2)	
<=	(S1) ≦ (S2)		<=	(S1) > (S2)	
<	(S1) < (S2)		<	(S1) ≧ (S2)	
>=	(S1) ≧ (S2)		>=	(S1) < (S2)	

(3) 当十六进制常数被 (S1) 和 (S2) 指定时，或最高位 (b15) 为 1 的数字值 (8 到 F) 被指定时，此值将当作负的 BIN 值被读取，用于比较。

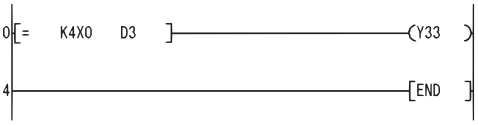
运算错误

(1) 没有与 =、<>、>、<=<、或 >= 指令相关的运行错误。

程序示例

(1) 下列程序将从 X0 到 XF 的数据与 D3 的数据进行比较，如果数据相同则将 Y33 变为 ON。

[梯形图模式]

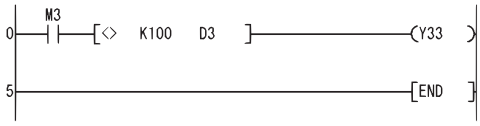


[列表模式]

步	指令	软元件	
0	LD=	K4X0	D3
3	OUT	Y33	
4	END		

(2) 下列程序将 BIN 值 K100 与 D3 中的数据进行比较，如果 D3 中的数据不是 100 则导通。

[梯形图模式]

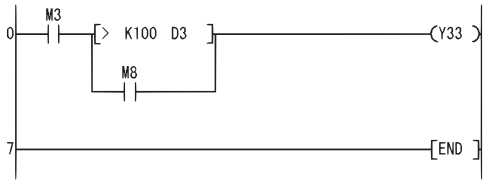


[列表模式]

步	指令	软元件	
0	LD	M3	
1	AND<>	K100	D3
4	OUT	Y33	
5	END		

(3) 下列程序将 BIN 值 100 与从 X0 到 XF 里的数据进行比较，如果 D3 的数据小于 100 则导通。

[梯形图模式]

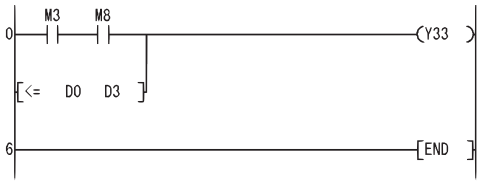


[列表模式]

步	指令	软元件	
0	LD	M3	
1	LD>	K100	D3
4	OR	M8	
5	ANB		
6	OUT	Y33	
7	END		

(4) 下列程序将 D0 和 D3 里数据进行比较，如果 D0 里的数据等于或小于 D3 里的数据则导通。

[梯形图模式]

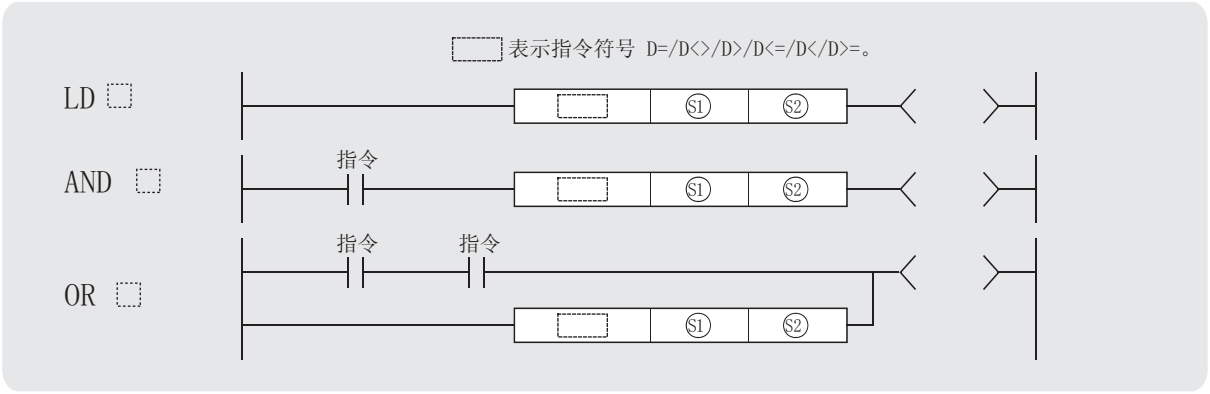


[列表模式]

步	指令	软元件	
0	LD	M3	
1	AND	M8	
2	OR<=	D0	D3
5	OUT	Y33	
6	END		

6.1.2 BIN 32 位数据比较 (D=、D<>、D>、D<=、D<、D>=)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)、(S2) ：比较数据或者存储比较数据的软元件的起始号 (BIN32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○								--
(S2)	○								--

功 能

(1) 把由 (S1) 指定的软元件的 BIN 32 位数据和由 (S2) 指定的软元件的 BIN 32 位数据作为 A 触点，进行比较操作。

(2) 单个指令比较操作的结果如下：

指令符号	条件	比较操作结果	指令符号	条件	比较操作结果
D=	(S2) = (S1)	导通	D=	(S1) ≠ (S2)	不导通
D<>	(S1) ≠ (S2)		D<>	(S2) = (S1)	
D>	(S1) > (S2)		D>	(S1) ≦ (S2)	
D<=	(S1) ≦ (S2)		D<=	(S1) > (S2)	
D<	(S1) < (S2)		D<	(S1) ≧ (S2)	
D>=	(S1) ≧ (S2)		D>=	(S1) < (S2)	

(3) 当十六进制常数被 (S1) 和 (S2) 指定时，或最高位 (b31) 为 1 的数字值 (8 到 F) 被指定时，此值将当作负的 BIN 值号被读取，用于比较。

(4) 用于比较的数据应当由 32 位指令 (DMOV 指令等) 指定。
如果由 16 位指令 (MOV 指令等) 指定，则大值和小值的比较将不能正确执行。

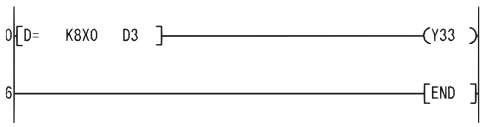
运算错误

(1) 没有与 D=、D<>、D>、D<=、D< 或 D>= 指令相关的运算错误。

程序示例

(1) 下列程序将 X0 到 X1F 的数据与 D3 和 D4 中的数据进行比较，如果 X0 到 X1F 的数据与 D3 和 D4 的数据匹配，则将 Y33 变为 ON。

[梯形图模式]

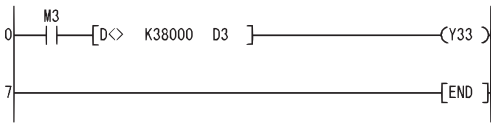


[列表模式]

步	指令	软元件	
0	LDD=	K8X0	D3
5	OUT	Y33	
6	END		

(2) 下列程序将 BIN 值 K38000 与 D3 和 D4 里的数据进行比较，如果 D3 和 D4 里的数据不是 38000 则导通。

[梯形图模式]

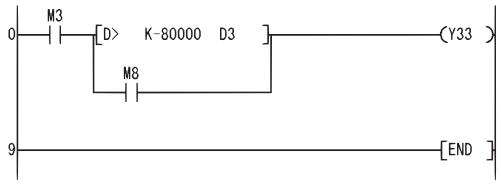


[列表模式]

步	指令	软元件	
0	LD	M3	
1	ANDD<>	K38000	D3
6	OUT	Y33	
7	END		

(3) 下列程序将 BIN 值 K－80000 与 D3 和 D4 里的数据进行比较，如果 D3 和 D4 里的数据小于 －80000 则导通。

[梯形图模式]

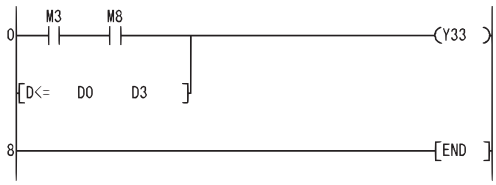


[列表模式]

步	指令	软元件	
0	LD	M3	
1	LDD>	K-80000	D3
6	OR	M8	
7	ANB		
8	OUT	Y33	
9	END		

(4) 下列程序将 D0 和 D1 里的数据与 D3 和 D4 里的数据进行比较，如果 D0 和 D1 里的数据等于或小于 D3 和 D4 里的数据则导通。

[梯形图模式]



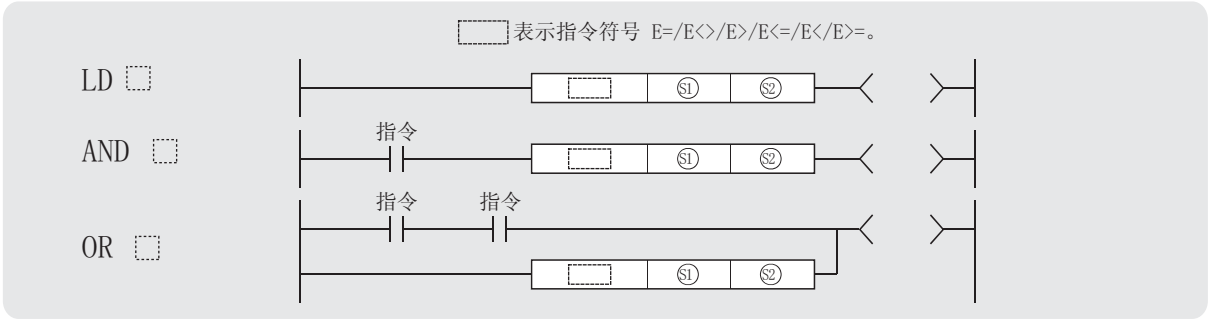
[列表模式]

步	指令	软元件	
0	LD	M3	
1	AND	M8	
2	ORD<=	D0	D3
7	OUT	Y33	
8	END		

6.1.3 浮动小数点数据比较 (E=、E<>、E>、E<=、E<、E>=)

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



S1、S2 ：比较数据或者存储比较数据的软元件的起始号。(实数)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
S1	--	○		--	○		--	○	--
S2	--	○		--	○		--	○	--

功 能

(1) 由 S1 和 S2 指定的软元件中的浮动小数点数据作为 A 触点，进行比较操作。

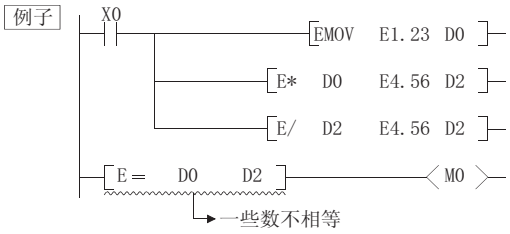
(2) 单个指令的比较操作结果如下：

□ 中的指令符号	条件	比较操作结果	□ 中的指令符号	条件	比较操作结果
E=	S2 = S1	导通	E=	S1 ≠ S2	不导通
E<>	S1 ≠ S2		E<>	S2 = S1	
E>	S1 > S2		E>	S1 ≦ S2	
E<=	S1 ≦ S2		E<=	S1 > S2	
E<	S1 < S2		E<	S1 ≧ S2	
E>=	S1 ≧ S2		E>=	S1 < S2	

要 点

注意，使用 E= 指令时，有时会导致由错误而引起的两个值不相等的情况。

例子



运算错误

(1) 在发生下列运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储到 SD0 中。

- 指定软元件的值为 −0。 *2
  （对于基本型 QCPU、高性能型 QCPU、过程控制 CPU、冗余 CPU 和 Q4ARCPU）
  （出错代码： 4100）

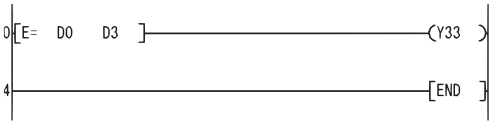
  *2: 有的 CPU 模块即使指定了 -0 也不发生运算出错。
  关于详细内容，请参照 3.2.4 项。

- 指定的软元件值超出以下范围。（仅对于通用型 QCPU）
  $0、2^{-126} \leqq |$ 指定的软元件值 $| < 2^{128}$ （出错代码： 4140）
- 指定软元件的值为 −0、非正常数字、非数字和±∞时。
  （仅对于通用型 QCPU） （出错代码： 4140）

程序示例

(1) 下列程序对 D0 和 D1 的浮动小数点实数与 D3 和 D4 的浮动小数点实数数据进行比较。

[梯形图模式]

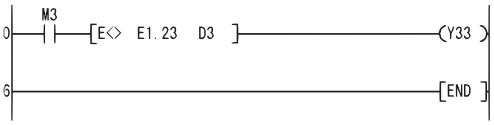


[列表模式]

步	指令	软元件	
0	LDE=	D0	D3
3	OUT	Y33	
4	END		

(2) 下列程序对浮动小数点实数 1.23 与 D3 和 D4 的浮动小数点实数数据进行比较。

[梯形图模式]

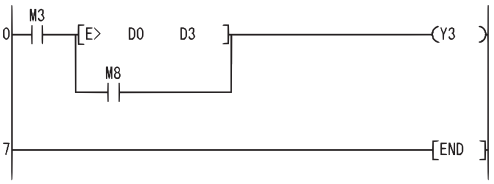


[列表模式]

步	指令	软元件	
0	LD	M3	
1	ANDE<>	E1.23	D3
5	OUT	Y33	
6	END		

(3) 下列程序对 D0 和 D1 的浮动小数点实数与 D3 和 D4 的浮动小数点实数数据进行比较。

[梯形图模式]

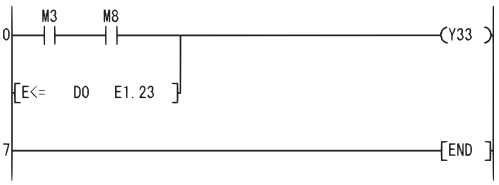


[列表模式]

步	指令	软元件	
0	LD	M3	
1	LDE>	D0	D3
4	OR	M8	
5	ANB		
6	OUT	Y3	
7	END		

(4) 下列程序对 D0 和 D1 的浮动小数点实数与浮动小数点实数数据 1.23 进行比较。

[梯形图模式]

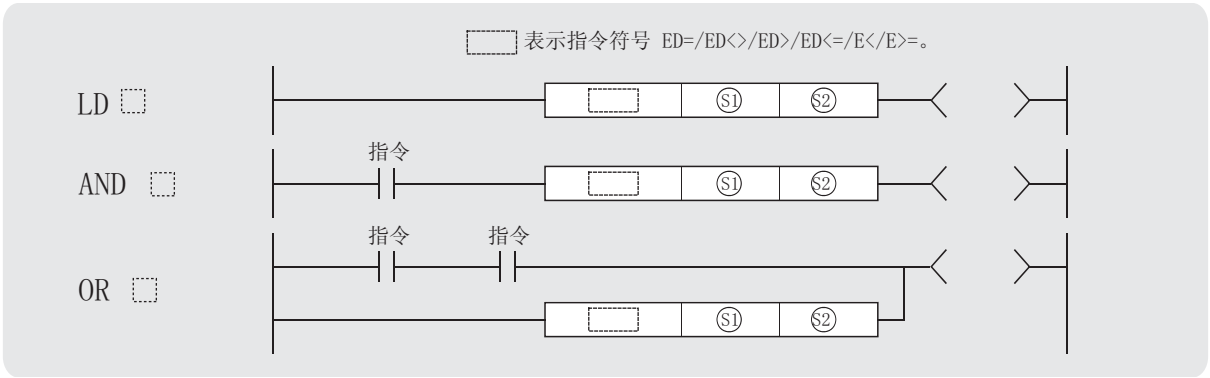


[列表模式]

步	指令	软元件	
0	LD	M3	
1	AND	M8	
2	ORE<=	D0	E1.23
6	OUT	Y33	
7	END		

6.1.4 浮动小数点数据比较（双精度）(ED=、ED<>、ED>、ED<=、ED<、ED>=)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)、(S2)：比较数据或者存储比较数据的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
(S1)	--	○		--	○		--	○	--
(S2)	--	○		--	○		--	○	--

☆功 能

(1) 把由 (S1) 指定的软元件的 64 位浮点实数与由 (S2) 指定的软元件的 64 位浮点实数作为 A（常开）触点进行比较运算。

(2) 各指令比较运算的结果如下：

内指令符号	条件	比较运算结果	内指令符号	条件	比较运算结果
ED=	(S2) = (S1)	导通	ED=	(S1) ≠ (S2)	非导通
ED<>	(S1) ≠ (S2)		ED<>	(S2) = (S1)	
ED>	(S1) > (S2)		ED>	(S1) ≦ (S2)	
ED<=	(S1) ≦ (S2)		ED<=	(S1) > (S2)	
ED<	(S1) < (S2)		ED<	(S1) ≧ (S2)	
ED>=	(S1) ≧ (S2)		ED>=	(S1) < (S2)	

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON。并且相关出错代码存储在 SD0 中。

- 指定的软元件的值不在以下范围时： （出错代码： 4140）

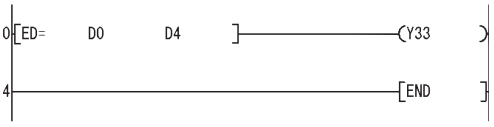
  0、$2^{-1022} \leqq$ ｜ 指定的软元件的值 ｜ $< 2^{1024}$

- 指定的软元件的值为－0 时。 （出错代码： 4140）

程序示例

(1) 下列程序将 D0 到 D3 的 64 位浮点实数数据与 D4 到 D7 的 64 位浮点实数数据进行比较。

[梯形图模式]

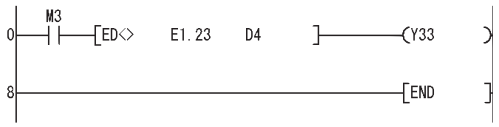


[列表模式]

步	指令	软元件	
0	LDED=	D0	D4
3	OUT	Y33	
4	END		

(2) 下列程序将浮点实数 1.23 与 D4 到 D7 的 64 位浮点实数数据进行比较。

[梯形图模式]

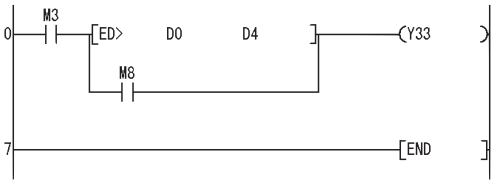


[列表模式]

步	指令	软元件	
0	LD	M3	
1	ANDED<>	E1.23	D4
7	OUT	Y33	
8	END		

(3) 下列程序将 D0 到 D3 的 64 位浮点实数数据与 D4 到 D7 的 64 位浮点实数数据进行比较。

[梯形图模式]

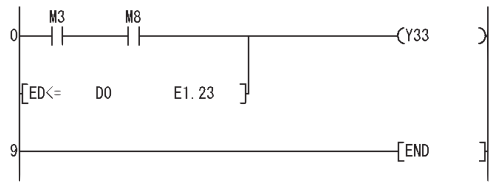


[列表模式]

步	指令	软元件	
0	LD	M3	
1	LDED>	D0	D4
4	OR	M8	
5	ANB		
6	OUT	Y33	
7	END		

(4) 下列程序将 D0 到 D3 的 64 位浮点实数数据与浮点实数 1.23 进行比较。

[梯形图模式]



[列表模式]

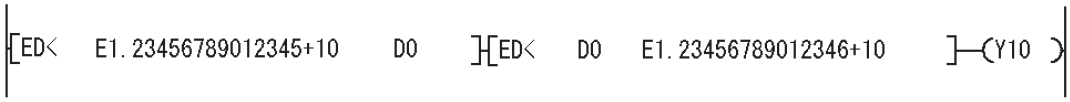
步	指令	软元件	
0	LD	M3	
1	AND	M8	
2	ORED<=	D0	E1.23
8	OUT	Y33	
9	END		

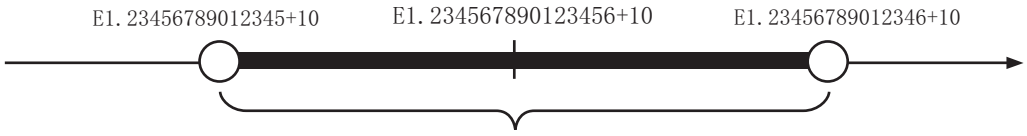
注意事项

(1) 因为可以通过 GX Developer 输入的实数位数最多为 15 位，所以本节中的指令不能比较有效位数为 16 位或以上的实数。

当用本节中的指令判断有效位数为 16 位或以上的实数的一致 / 不一致时，需要将其与待比较的实数的近似值进行比较以判断其大小。

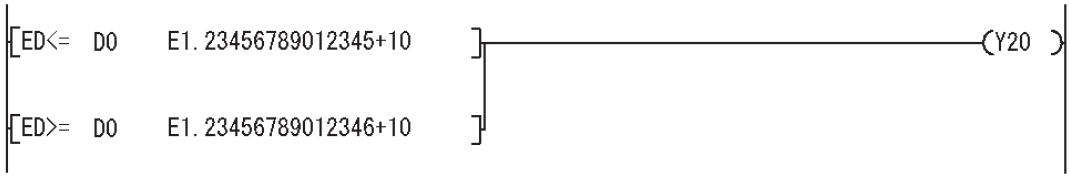
例如 当判断 E1.234567890123456+10（有效位数为 16）与双精度浮点数据的一致性时。

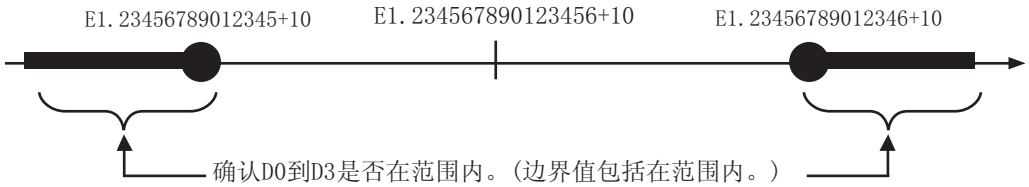




确认D0到D3是否在该范围内。（边界值不包括在范围内。）

例如 当判断 E1.234567890123456+10（有效位数为 16）与双精度浮动小数点数据的不一致性时。

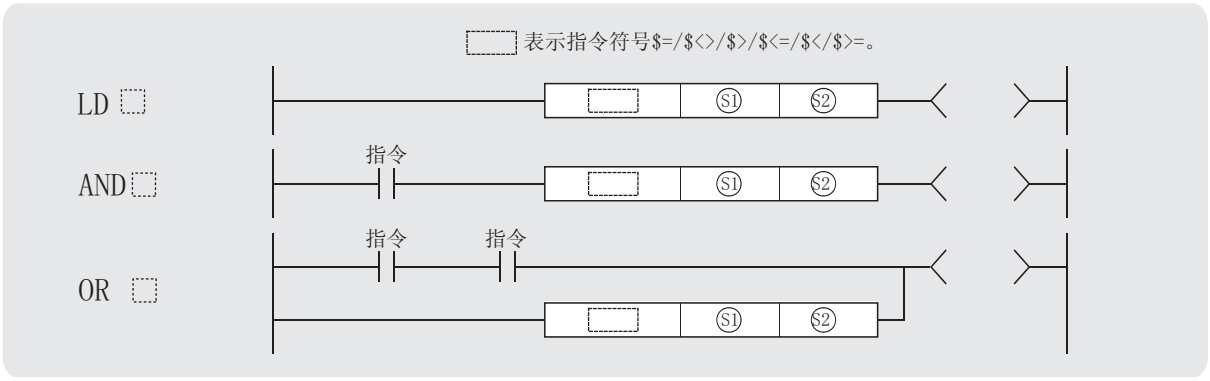




确认D0到D3是否在范围内。（边界值包括在范围内。）

6.1.5 字符串数据比较（$=、$<>、$>、$<=、$<、$>=）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



S1、S2：比较数据或者存储比较数据的软元件的起始号（字符串）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
S1	--	○		--				○	--
S2	--	○		--				○	--

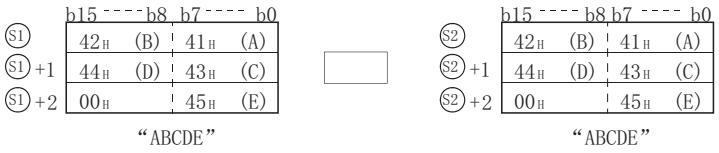
功 能

(1) 将存储在由 S2 指定的软元件里的字符串数据和存储在由 S1 指定的软元件里的字符串数据作为 A 触点，并进行比较操作。

(2) 比较操作涉及到字符串里第一个字符的 ASCII 码的字符与字符比较。

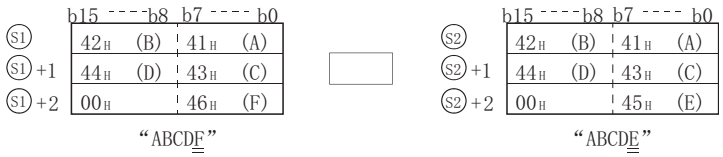
(3) S1 和 S2 字符串包含了从指定的软元件号到下一个存储了代码“00H”的软元件号中的所有字符。

(a) 如果所有字符串都匹配，则比较结果匹配。



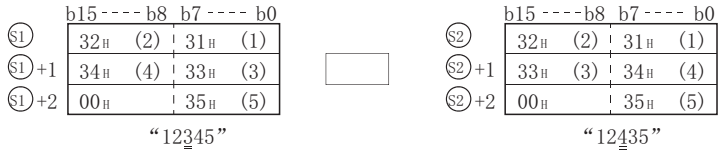
□ 中的指令符号	比较操作结果	□ 中的指令符号	比较操作结果
$=	导通	$<=	导通
$<>	不导通	$<	不导通
$>	不导通	$>=	导通

(b) 如果字符串不同，则带较大字符代码的字符串将更大。



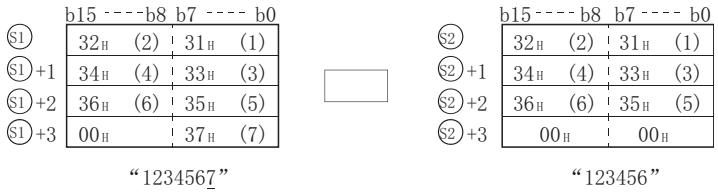
中的指令符号	比较操作结果	中的指令符号	比较操作结果
$=	不导通	$<=	不导通
$<>	导通	$<	不导通
$>	导通	$>=	导通

(c) 如果字符串不同，则第一个大小不同的字符代码将决定字符串是较大还是较小。



中的指令符号	比较操作结果	中的指令符号	比较操作结果
$=	不导通	$<=	导通
$<>	导通	$<	导通
$>	不导通	$>=	不导通

(4) 如果由 (S1) 和 (S2) 指定的字符串长度不同，则有较长字符串的数据将更大。



中的指令符号	比较操作结果	中的指令符号	比较操作结果
$=	不导通	$<=	不导通
$<>	导通	$<	不导通
$>	导通	$>=	导通

运算错误

(1) 在发生下列运行错误的情况中，错误标志 (SM0) 变为 ON，并且在 SD0 中存储出错代码。

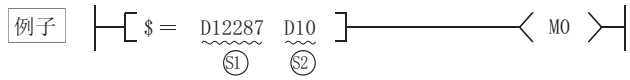
• 代码 “00H” 或非匹配在由 ⓢ1 或 ⓢ2 指定的软元件号之后的相关软元件范围内不存在。
（出错代码： 4101）

• ⓢ1 和 ⓢ2 的字符串超过 16383 字符时。 （出错代码： 4101）

要 点

在进行字符串比较的同时，字符串数据比较指令也检查软元件范围。
因此，即使字符串长度超过软元件范围，字符串数据仍然被比较。这时如果在软元件范围内检测到字符不匹配，则比较操作结果不返回运行错误就输出。

例子



ⓢ1 数据

D12287	B	A
W0	00H	C

ⓢ2 数据

D10	Z	A
D11	00H	C

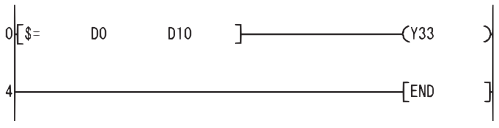
如果 ⓢ1 和 ⓢ2 的数据如上，那么 ⓢ1 的第 2 个字符与 ⓢ2 的第 2 个字符不匹配，因此比较结果为 ⓢ1 ≠ ⓢ2 （运算结果为 “不导通”）。

此时虽然 ⓢ1 的软元件范围中不存在 “00H” 代码，但是因为检测出不一致的软元件为 D12287（在软元件范围内），因此不返回运算错误信息。

程序示例

(1) 下述程序对存储在 D0 之后的字符串和 D10 之后的字符进行比较。

[梯形图模式]

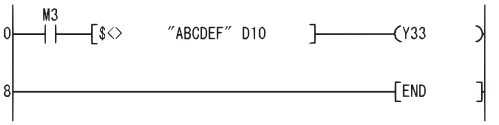


[列表模式]

步	指令	软元件	
0	LD$=	D0	D10
3	OUT	Y33	
4	END		

(2) 下述程序对字符串 “ABCDEF” 和存储在 D10 之后的字符串进行比较。

[梯形图模式]

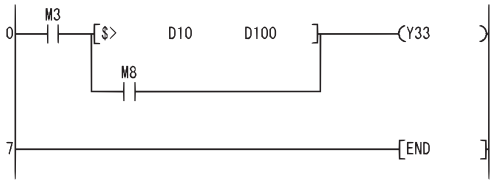


[列表模式]

步	指令	软元件
0	LD	M3
1	AND$<>	"ABCDEF" D10
7	OUT	Y33
8	END	

(3) 下述程序对存储在 D10 之后的字符串和存储在 D100 之后的字符串进行比较。

[梯形图模式]

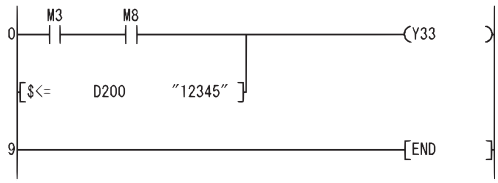


[列表模式]

步	指令	软元件	
0	LD	M3	
1	LD$>	D10	D100
4	OR	M8	
5	ANB		
6	OUT	Y33	
7	END		

(4) 下述程序对存储在 D200 之后的字符串和字符串 “12345”进行比较。

[梯形图模式]

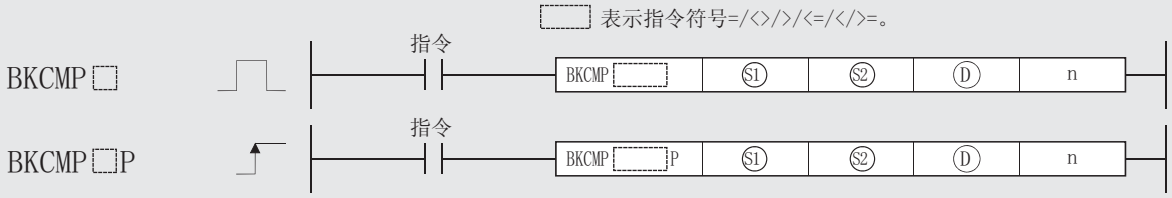


[列表模式]

步	指令	软元件	
0	LD	M3	
1	AND	M8	
2	OR$<=	D200	"12345"
8	OUT	Y33	
9	END		

6.1.6 BIN 块数据比较 (BKCMP □ 、BKCMP □ P)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ： 被比较数据或者存储被比较数据的软元件的起始号。(BIN16 位)

(S2) ： 存储比较数据的软元件的起始号。(BIN16 位)

(D) ： 存储比较操作结果的软元件的起始号 (位)。

n ： 比较数据块数 (BIN16 位)。

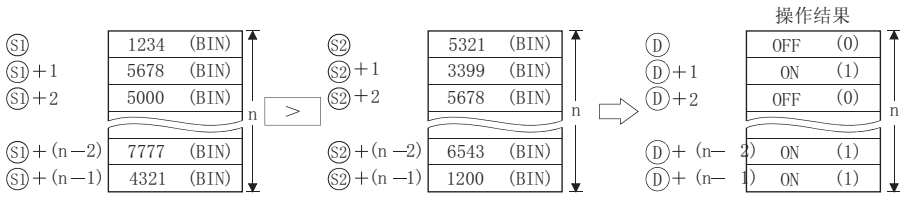
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	--	○		--				○	--
(S2)	--	○		--				--	--
(D)	○	○		--				--	--
n	○	○		○				○	--

功 能

(1) 从由 (S1) 指定的软元件号开始的第 n 点 BIN 16 位数据与由 (S2) 指定的软元件号开始的第 n 点 BIN 16 位数据进行比较，并且将结果依此向前存储到从由 (D) 指定的软元件开始的软元件中去。

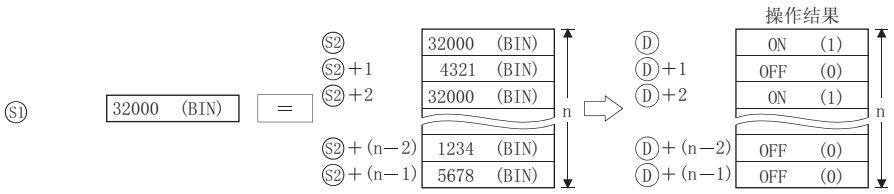
(a) 如果满足比较条件，则 (D) 指定的软元件将变为 ON。

(b) 如果比较条什不满足，则 (D) 指定的软元件将变为 OFF。



(2) 比较操作在 16 位单元中进行。

(3) 由 (S1) 指定的常数范围在 -32768 与 32767 之间 (BIN 16- 位数据)。



(4) 单个指令的比较操作结果如下：

指令符号	条件	比较操作结果	指令符号	条件	比较操作结果
BKCMP=	ⓢ2 = ⓢ1	ON (1)	BKCMP=	ⓢ1 ≠ ⓢ2	OFF (0)
BKCMP<>	ⓢ1 ≠ ⓢ2		BKCMP<>	ⓢ2 = ⓢ1	
BKCMP>	ⓢ1 > ⓢ2		BKCMP>	ⓢ1 ≦ ⓢ2	
BKCMP<=	ⓢ1 ≦ ⓢ2		BKCMP<=	ⓢ1 > ⓢ2	
BKCMP<	ⓢ1 < ⓢ2		BKCMP<	ⓢ1 ≧ ⓢ2	
BKCMP>=	ⓢ1 ≧ ⓢ2		BKCMP>=	ⓢ1 < ⓢ2	

(5) 如果所有存储在从 ⓓ 开始的 n 点比较结果都为 ON(1)，SM704(块比较信号)变为 ON。

运算错误

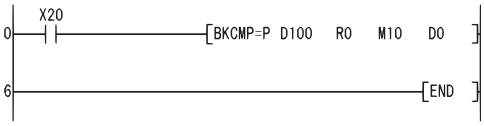
(1) 在发生下列运行错误的情况中，错误标志 (SM0) 变为 ON，并且在 SD0 中存储出错代码。

- 从由 ⓢ1 、 ⓢ2 或 ⓓ 指定的软元件开始的 n 点的软元件的范围超过相关软元件。
  (出错代码： 4101)
- 从由 ⓢ1 指定的软元件开始的 n 点软元件的范围与从 ⓓ 指定的软元件开始的 n 点软元件的范围重叠。
  (出错代码： 4101)
- 从由 ⓢ2 指定的软元件开始的 n 点软元件的范围与从 ⓓ 指定的软元件开始的 n 点软元件的范围重叠。
  (出错代码： 4101)

程序示例

(1) 下列程序在 X20 变为 ON 时执行比较运算，将存储在 D100 到 D103 的数据与存储在 R0 到 R3 的数据进行比较，并且将结果存储到从 M10 开始的地址中。

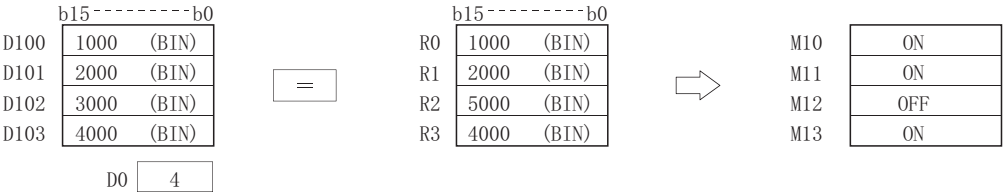
[梯形图模式]



[列表模式]

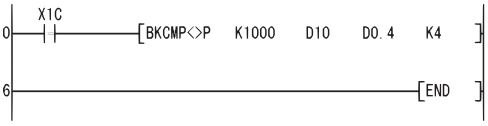
步	指令	软元件
0	LD	X20
1	BKCMP=P	D100 R0 M10 D0
6	END	

[动作]



(2) 当 X1C 变为 ON 时，下列程序执行比较运算，将常数 K1000 与存储在 D10 到 D13 的数据进行比较，并且将结果存储在 D0 的 b4 到 b7 中。

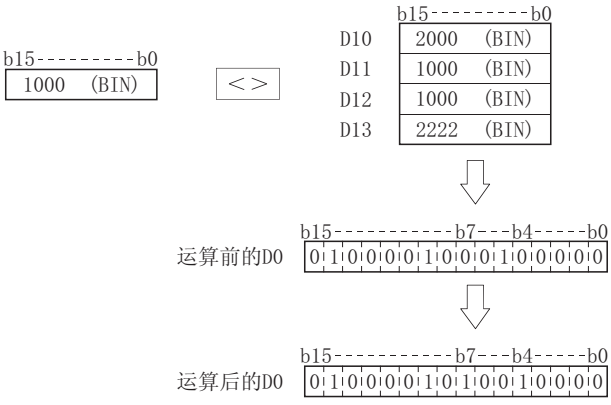
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X1C
1	BKCMP<>P	K1000 D10 D0.4 K4
6	END	

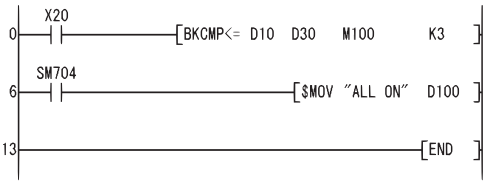
[动作]



(3) 当 X20 变为 ON 时，下列程序执行比较运算，将 D10 到 D12 的数据与 D30 到 D32 的数据进行比较，并且将结果存储在从 M100 开始的区域中。

当从 M100 开始的所有软元件均为 1(ON) 状态时，下述程序将字符串 “ALL ON”传送到从 D100 开始的地址中。

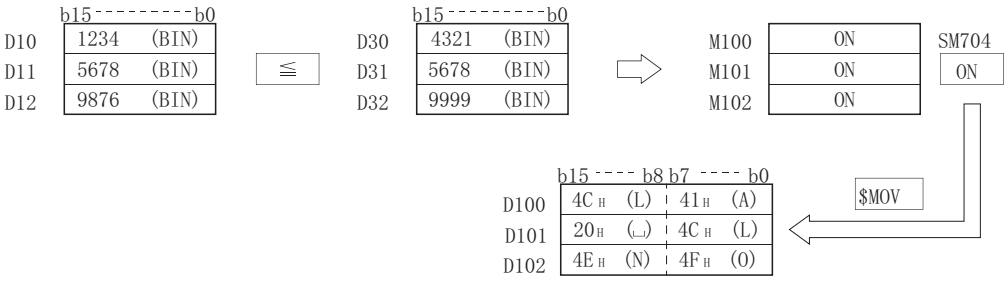
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X20
1	BKCMP<=	D10 D30 M100 K3
6	LD	SM704
7	$MOV	"ALL ON" D100
13	END	

[动作]

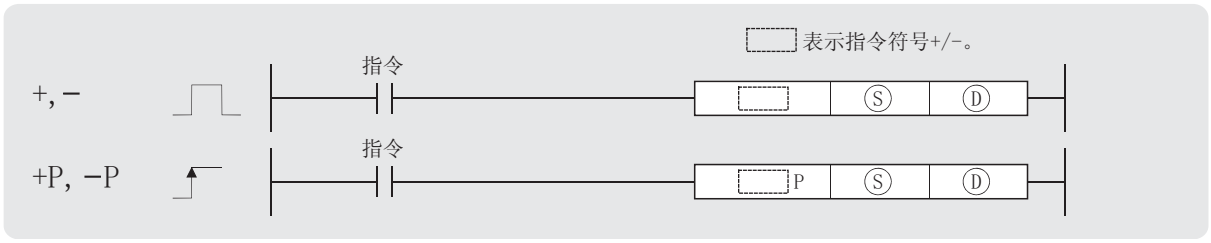


6.2 算术运算指令

6.2.1 BIN 16 位加法和减法运算 (+(P)、-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（ Ⓓ + Ⓢ → Ⓓ、Ⓓ - Ⓢ → Ⓓ）



Ⓢ ： 加数、减数或者存储加数、减数数据的软元件的起始号 (BIN16 位)。

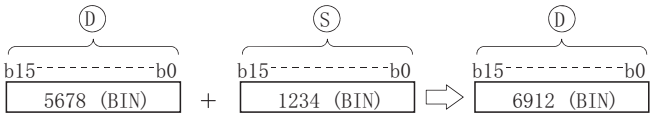
Ⓓ ： 存储被加数、被减数数据的软元件的起始号。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○						○	--
Ⓓ		○						--	--

功 能

+

(1) 将由 Ⓓ 指定的 BIN 16 位数据与由 Ⓢ 指定的 BIN 16 位数据相加，并将该加法运算的结果存储在由 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 -32768 到 32767(BIN 16 位)。

(3) 数据的正负是由它的最高有效位 (b15) 来判定的。

- 0: 正
- 1: 负

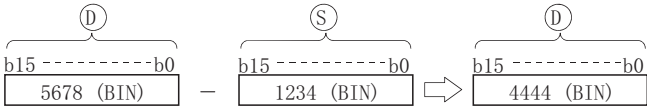
(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：在这种情况下，进位标志不变为 ON。

· K32767 (7FFFH) +K2 (0002H) → K−32767 (8001H) …… 由于位15值为“1”，因此变为负值。

· K−32768 (8000H) +K−2 (FFFEH) → K32766 (7FFEH) …… 由于位15值为“0”，因此变为正值。

－

(1) 从由 Ⓢ 指定的 BIN 16 位数据中减去由 Ⓓ 指定的 BIN 16 位数据，并将该减法运算的结果存储在由 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 -32768 到 32767(BIN 16 位)。

(3) 数据的正负是由它的最高有效位 (b15) 来判定的。

- 0：正
- 1：负

(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

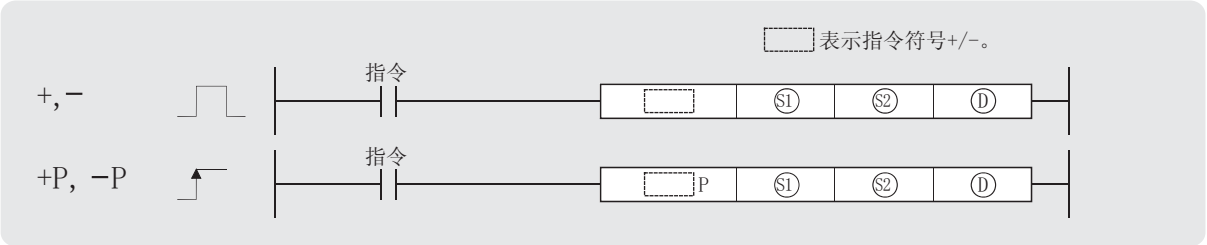
· K－32768 (8000H) － K2 (0002H) ⟶ K32766 (7FFEH) ····· 由于位15值为“0”，因此变为正值。

· K32767 (7FFFH) － K－2 (FFFEH) ⟶ K－32767 (8001H) ····· 由于位15值为“1”，因此变为负值。

运算错误

(1) 不存在与 +(P) 或 -(P) 指令相关的运行错误。

2 当设置了三个数据时（ ⓢ1 + ⓢ2 → ⓓ、 ⓢ1 - ⓢ2 → ⓓ）



ⓢ1 ： 被加数、被减数或者存储被加数、被减数的软元件的起始号（BIN16 位）。

ⓢ2 ： 加数、减数或者存储加数、减数的软元件的起始号（BIN16 位）。

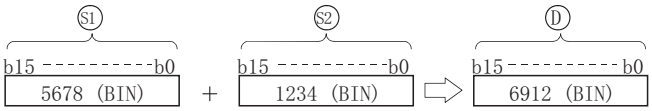
ⓓ ： 存储加法和减法运算结果的软元件的起始号（BIN16 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
ⓢ1				○				○	--
ⓢ2				○				○	--
ⓓ				○				--	--

功 能

+

(1) 将由 ⓢ1 指定的 BIN 16 位数据与由 ⓢ2 指定的 BIN 16 位数据相加，并将结果存储在由 ⓓ 指定的软元件中。



(2) 可以指定给 ⓢ1、ⓢ2 和 ⓓ 的值的范围是 -32768 到 32767(BIN 16 位)。

(3) 数据的正负是由它的最高有效位 (b15) 来判定的。

- 0：正
- 1：负

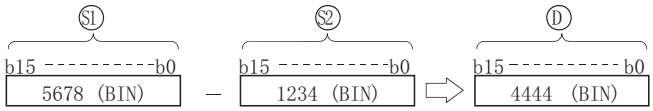
(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K32767 (7FFFH) +K2 (0002H) ⟶ K−32767 (8001H) …… 由于位15值为“1”，因此变为负值。

· K−32768 (8000H) +K−2 (FFFEH) ⟶ K32766 (7FFEH) …… 由于位15值为“0”，因此变为正值。

–

(1) 从由 (S2) 指定的 BIN 16 数据中减去由 (S1) 指定的 BIN 16 位数据，并将减法运算的结果存储在由 (D) 指定的软元件中。



(2) 可以指定给 (S1)、(S2) 和 (D) 的值的范围是 -32768 到 32767(BIN 16 位)。

(3) 数据的正负是由它的最高有效位 (b15) 来判定的。

- 0：正
- 1：负

(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K－32768 (8000H) －K 2 (0002H) ⟶ K32766 (7FFEH) ····· 由于位15值为“0”，因此变为正值。

· K32767 (7FFFH) －K－2 (FFFEH) ⟶ K－32767 (8001H) ····· 由于位15值为“1”，因此变为负值。

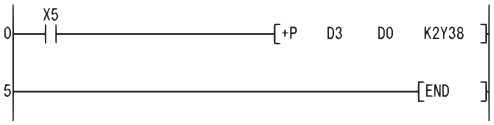
运算错误

(1) 不存在与 +(P) 或 -(P) 指令相关的运行错误。

程序示例

(1) 当 X5 变为 ON 时，以下程序将 D3 的内容与 D0 的内容相加，并将结果输出到 Y38 到 Y3F。

[梯形图模式]

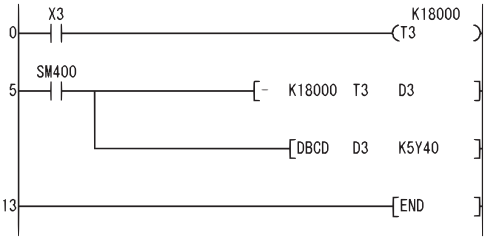


[列表模式]

步	指令	软元件
0	LD	X5
1	+P	D3 D0 K2Y38
5	END	

(2) 以下程序将定时器 T3 的改定值与当前值之间的差值输出到 Y40 到 Y53。

[梯形图模式]



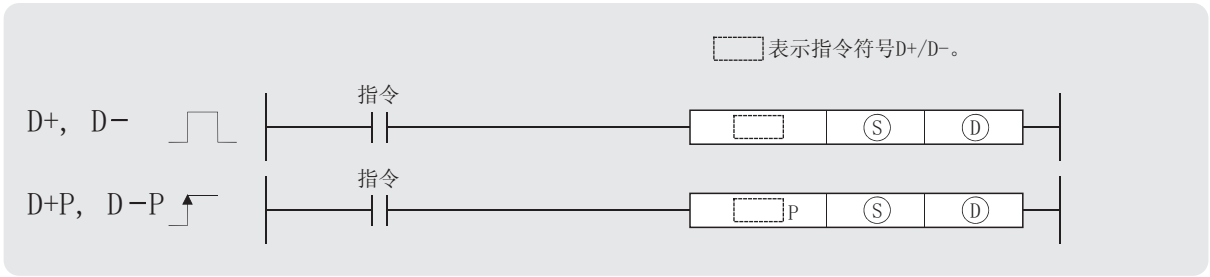
[列表模式]

步	指令	软元件
0	LD	X3
1	OUT	T3 K18000
5	LD	SM400
6	-	K18000 T3 D3
10	DBCD	D3 K5Y40
13	END	

6.2.2 BIN 32 位加法和减法运算 (D+(P)、D-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时 ((Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ)、(Ⓓ+1、Ⓓ)-(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ))



Ⓢ : 加数、减数或者存储加数、减数的软元件的起始号 (BIN32 位)。

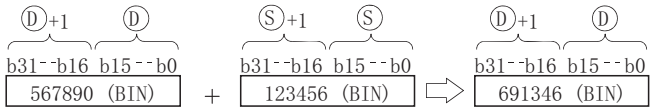
Ⓓ : 存储被加数、被减数的软元件的起始号 (BIN32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	--
Ⓓ	○							--	--

功 能

D+

(1) 将由 Ⓓ 指定的 BIN 32 位数据与由 Ⓢ 指定的 BIN 32 位数据相加，并将该加法运算的结果存储在由 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 -2147483648 到 2147483647(BIN 32 位)。

(3) 数据的正负是根据它的最高有效位 (b31) 来判定的。

- 0: 正
- 1: 负

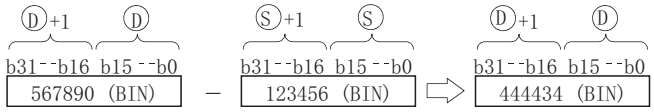
(4) 如果运算结果产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K 2147483647 (7FFFFFFFH) +K 2 (00000002H) → K-2147483647 (80000001H) …… 因为b31为1，所以值为负。

· K-214748364 8 (80000000H) +K-2 (FFFFFFFEH) → K 214748364 6 (7FFFFFFEH) …… 因为b31为0，所以值为正。

D-

(1) 从由 Ⓢ 指定的 BIN 32 位数据中减去由 Ⓓ 指定的 BIN 32 位数据，并将该减法运算的结果存储在由 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 -2147483648 到 2147483647(BIN 32 位)。

(3) 数据的正负是根据它的最高有效位 (b31) 来判定的。

- 0：正
- 1：负

(4) 如果运算结果产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

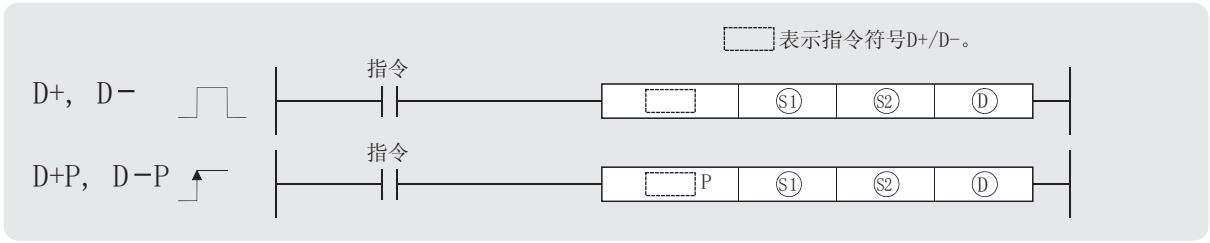
·K－2147483648 (80000000H) －K 2 (00000002H) ⟶ K 2147483646 (7FFFFFFEH) …… 因为b31为0，所以值为正。

·K 2147483647 (80000000H) －K－2 (FFFFFFFEH) ⟶ K－2147483647 (80000001H) …… 因为b31为1，所以值为负。

运算错误

(1) 不存在与 D+(P) 或 D-(P) 指令相关的运算错误。

2 当设置了三个数据时（(Ⓢ1+1、Ⓢ1)+(Ⓢ2+1、Ⓢ2)→(Ⓓ+1、Ⓓ)、(Ⓢ1+1、Ⓢ1)-(Ⓢ2+1、Ⓢ2)→(Ⓓ+1、Ⓓ)）



Ⓢ1 ： 被加数、被减数或者存储被加数、被减数的软元件的起始号（BIN32 位）。

Ⓢ2 ： 加数、减数或者存储加数、减数的软元件的起始号（BIN32 位）。

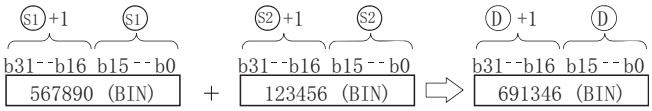
Ⓓ ： 存储加法和减法运算结果的软元件的起始号（BIN32 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ1				○				○	--
Ⓢ2				○				○	--
Ⓓ				○				--	--

功 能

D+

(1) 将由 Ⓢ1 指定的 BIN 32 位数据与由 Ⓢ2 指定的 BIN 32 位数据相加并将其结果存储在由 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ1 Ⓢ2、和 Ⓓ 的值的范围是 -2147483648 到 2147483647(BIN 32 位）。

(3) 数据的正负是根据它的最高有效位 (b31) 来判定的。

- 0：正
- 1：负

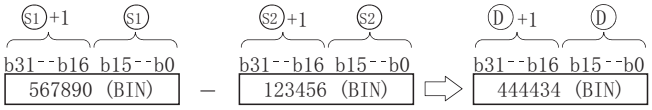
(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K2147483647 (7FFFFFFFH) +K2 (00000002H) ⟶ K-2147483647 (80000001H) … 因为b31为1，所以值为正。

· K-2147483648 (80000000H) +K-2 (FFFFFFFEH) ⟶ K2147483646 (7FFFFFFEH) …… 因为b31为0，所以值为负。

D-

(1) 从由 (S2) 指定的 BIN 32 位数据中减去由 (S1) 指定的 BIN 32 位数据，并将该减法运算的结果存储在由 (D) 指定的软元件中。



(2) 可以指定给 (S1)(S2)、和 (D) 的值的范围是 -2147483648 到 2147483647(BIN 32 位)。

(3) 数据的正负是根据它的最高有效位 (b31) 来判定的。

- 0：正
- 1：负

(4) 如果运算结果产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K−2147483648 (80000000H) −K2 (00000002H) ⟶ K2147483646 (7FFFFFFEH) …… 因为b31为0，所以值为正。

· K2147483647 (7FFFFFFFH) −K−2 (FFFFFFFEH) ⟶ K−2147483647 (80000001H) … 因为b31为1，所以值为负。

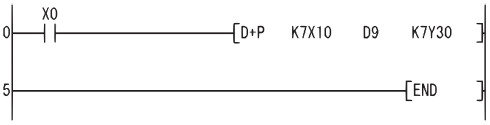
运算错误

(1) 不存在与 D+(P) 或 D-(P) 指令相关的运算错误。

程序示例

(1) 当 X0 变为 ON 时，以下程序将来自 X10 到 X28 的 28 位数据与 D9 和 D10 上的数据相加，并将运算结果输出到 Y30 到 Y4B。

[梯形图模式]

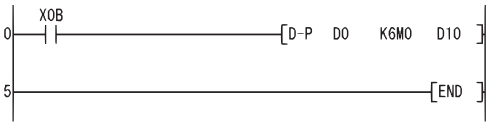


[列表模式]

步	指令	软元件
0	LD	X0
1	D+P	K7X10 D9 K7Y30
5	END	

(2) 当 XB 变为 ON 时，以下程序从 D0 和 D1 上的数据中减去来自 M0 到 M23 的数据，并将运算结果存储在 D10 和 D11 上。

[梯形图模式]

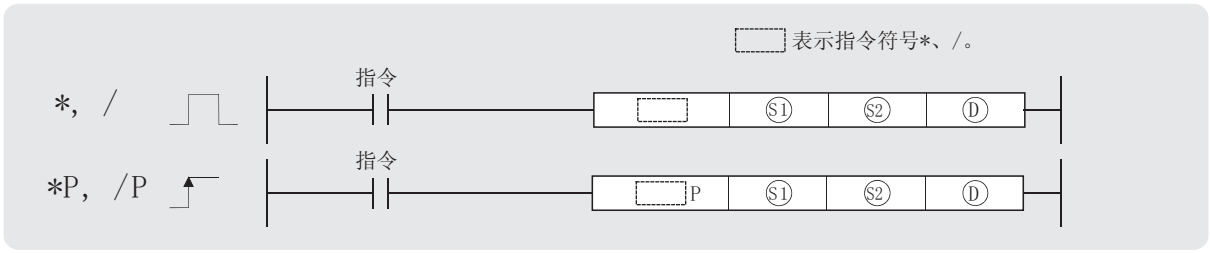


[列表模式]

步	指令	软元件
0	LD	X0B
1	D-P	D0 K6M0 D10
5	END	

6.2.3 BIN 16 位乘法和除法运算 (*(P)、/(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：被乘数、被除数或者存储被乘数、被除数的软元件的起始号 (BIN16 位)。

(S2) ：乘数、除数或者存储乘数、除数的软元件的起始号 (BIN16 位)。

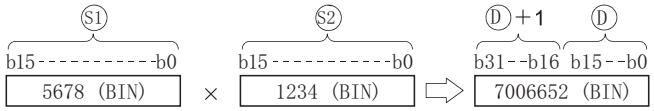
(D) ：存储乘法、除法运算的运算结果的软元件的起始号 (BIN32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)				○				○	--
(S2)				○				○	--
(D)				○				--	--

功 能

*

(1) 将由 (S1) 指定的 BIN 16 位数据与由 (S2) 指定的 BIN 16 位数据相乘，并将其结果存储在由 (D) 指定的软元件中。



(2) 如果 (D) 是位软元件，那么值由低位开始指定。

例如

K1 ..... 低 4 位 (b0 到 b3)
K4 ..... 低 16 位 (b0 到 b15)
K8 ..... 低 32 位 (b0 到 b31)

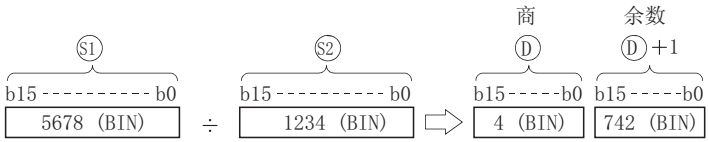
(3) 可以指定给 (S1)、(S2) 和 (D) 的值的范围是 -32768 到 32767(BIN 16 位)。

(4) (S1)、(S2) 和 (D) 的正负是根据它们的最高有效位 (对 (S1)、(S2) 来说，即为 b15 ; 对 (D)，则为 b31) 来判定的。

- 0: 正
- 1: 负

/

(1) 将由 (S1) 指定的 BIN 16 位数据除以由 (S2) 指定的 BIN 16 位数据，并将其结果存储在 (D) 指定的软元件中。



(2) 如果使用的是字软元件，那么除法运算的结果存储为 32 位，并且将商和余数都存储起来；如果使用的是位软元件，那么结果使用 16 位，并且只存储商。

商 ..... 存储在低 16 位
余数 .... 存储在高 16 位（只有在使用字软元件时才可以存储）

(3) 可以指定给 (S1) 和 (S2) 的值的范围是 -32768 到 32767(BIN 16 位)。

(4) (S1)、(S2)、(D) 和 (D)+1 的值的正负是根据它们的最高位 (b15) 来判定的。（商和余数上也附带有符号。）

- 0: 正
- 1: 负

运算错误

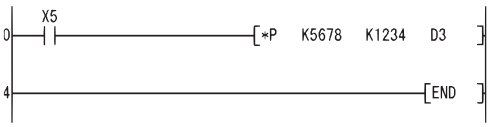
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 试图用 0 去除 (S2) 。 （出错代码 : 4100）

程序示例

(1) 当 X5 变为 ON 时，以下程序用 “1234”“5678”，并将结果存储在 D3 和 D4 中。

[梯形图模式]

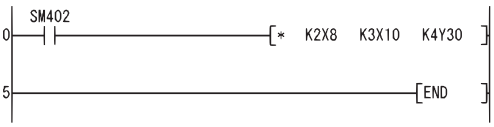


[列表模式]

步	指令	软元件
0	LD	X5
1	*P	K5678 K1234 D3
4	END	

(2) 以下程序用 X10 到 X1B 上的 BIN 数据去除 X8 到 XF 上的 BIN 数据，并将该除法运算的结果输出到 Y30 到 Y3F。

[梯形图模式]

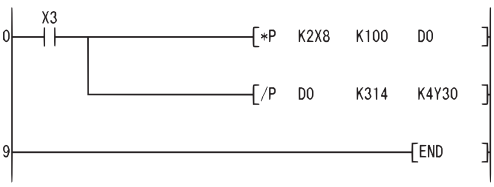


[列表模式]

步	指令	软元件
0	LD	SM402
1	*	K2X8 K3X10 K4Y30
5	END	

(3) 当 X3 为 ON 时，以下程序将 X8 到 XF 上的数据除以 3.14 的结果值输出到 Y30 到 Y3F。

[梯形图模式]

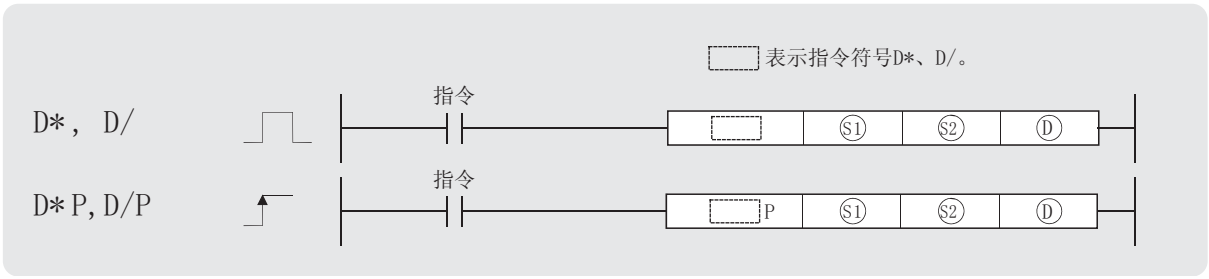


[列表模式]

步	指令	软元件
0	LD	X3
1	*P	K2X8 K100 D0
5	/P	D0 K314 K4Y30
9	END	

6.2.4 BIN 32 位乘法和除法运算 (D*(P)、D/(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ： 被乘数、被除数或者存储被乘数、被除数的软元件的起始号 (BIN32 位)。

(S2) ： 乘数、除数或者存储乘数、除数的软元件的起始号 (BIN32 位)。

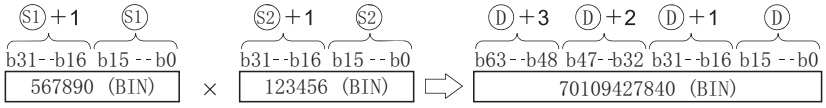
(D) ： 存储乘法、除法运算的运算结果的软元件的起始号 (BIN64 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)		○			○				--
(S2)		○			○				--
(D)		○			--				--

功 能

D*

(1) 将由 (S1) 指定的 BIN 32 位数据与由 (S2) 指定的 BIN 32 位数据相乘并将其结果存储在由 (D) 指定的软元件中。



(2) 如果 (D) 是位软元件，那么将只考虑乘法运算结果的低 32 位，而高 32 位不会被指定。

例如

K1..... 低 4 位 (b0 到 b3)
K4..... 低 16 位 (b0 到 b15)
K8..... 低 32 位 (b0 到 b31)

对乘法运算的结果来说，如果该位软元件的高 32 位也需要考虑，那么首先将该数据结果暂存在一个字软元件中，然后将字软元件中的数据传输给指定 ((D)+2) 和 ((D)+3) 的数据的位软元件中。

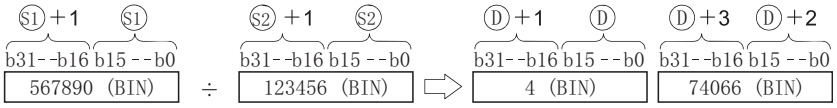
(3) 可以指定给 (S1) 和 (S2) 的值的范围是 -2147483648 到 2147483647(BIN32 位)。

(4) (S1)、(S2)、 和 (D) 的值的正负是根据它们的最高有效位 (对(S1)、(S2)来说，即为 b31 ； 对 (D)，则为 b63) 来判定的。

- 0: 正
- 1: 负

D/

(1) 用由 (S1) 指定的 BIN 32 位数据除以由 (S2) 指定的 BIN 32 位数据，并将其结果存储在 (D) 指定的软元件中。



(2) 如果使用的是字软元件，那么除法运算的结果存储为 64 位，并且将商和余数都存储起来；如果使用的是位软元件，那么结果使用 32 位，并且只存储商。

商..... 存储在低 32 位在中。
余数..... 存储在高 32 位中（只有在使用字软元件时才可以存储）。

(3) 可以指定给 (S1) 和 (S2) 的值的范围是 -2147483648 到 2147483647(BIN32 位)。

(4) (S1)、(S2)、(D) 和 (D)+2 的值的正负是根据它们的最高有效位 (b31) 来判定的。
（符号与商和余数一起使用）

- 0：正
- 1：负

运算错误

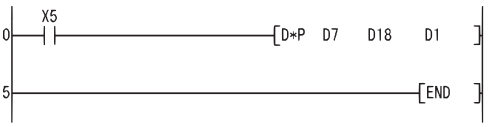
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 试图用 0 去除 (S2) 。 （出错代码： 4100）

程序示例

(1) 当 X5 为 ON 时，以下程序将 D7 和 D8 中的 BIN 数据除以 D18 和 D19 中的 BIN 数据，并将结果存储在 D1 到 D4 中。

[梯形图模式]

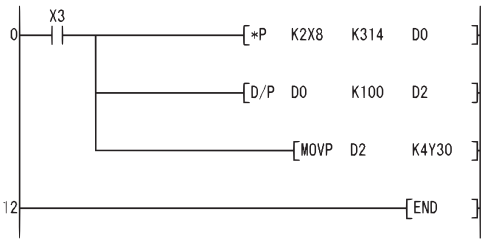


[列表模式]

步	指令	软元件		
0	LD	X5		
1	D*P	D7	D18	D1
5	END			

(2) 当 X3 为 ON 时，以下程序将 X8 到 XF 中的数据乘以 3.14 所得到的结果输出到 Y30 到 Y3F。

[梯形图模式]



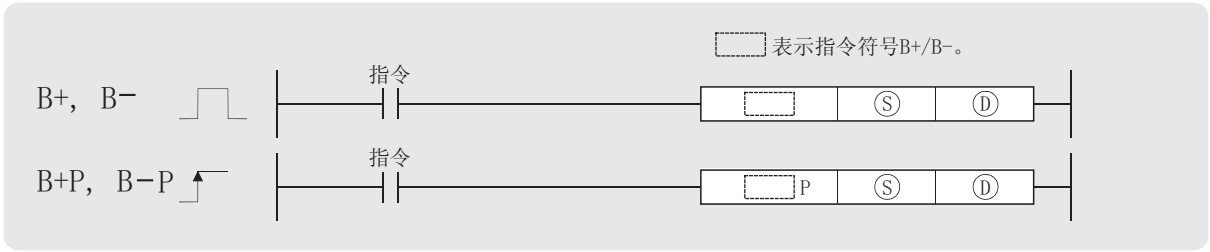
[列表模式]

步	指令	软元件		
0	LD	X3		
1	*P	K2X8	K314	D0
5	D/P	D0	K100	D2
10	MOVP	D2	K4Y30	
12	END			

6.2.5 BCD 4位加法和减法运算 (B+(P)、B-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（ ⓓ + ⓢ → ⓓ、 ⓓ - ⓢ → ⓓ）



ⓢ ：加数、减数或者存储加数、减数的软元件的起始号（BCD4位）。

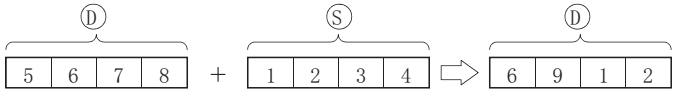
ⓓ ：被加数、被减数或者存储被加数、被减数的软元件的起始号（BCD4位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
ⓢ		○						○	--
ⓓ		○						--	--

功 能

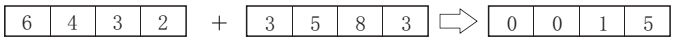
B+

(1) 将 ⓓ 指定的BCD 4位数据与 ⓢ 指定的BCD 4位数据相加，并将该加法运算的结果存储在 ⓓ 指定的软元件中。



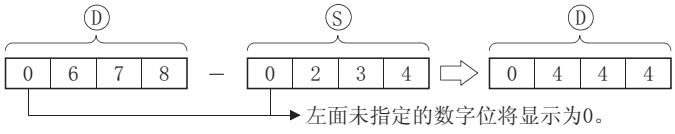
(2) 可以指定给 ⓢ 和 ⓓ 的值的范围是0到9999(BCD 4位)。

(3) 如果加法运算的结果超出了9999，那么其进位将被忽略。在这种情况下，进位标志不变为ON。



B-

(1) 从 ⓓ 指定的BCD 4位数据中减去由 ⓢ 指定的BCD 4位数据，并将该减法运算的结果存储在 ⓓ 指定的软元件中。



(2) 可以指定给 ⓢ 和 ⓓ 的值的范围是0到9999(BCD 4位)。

(3) 如果减法运算产生了下溢，则会导致以下情况：
在这种情况下，进位标志不变为 ON。

0	0	0	1	−	0	0	0	3	⇨	9	9	9	8

运算错误

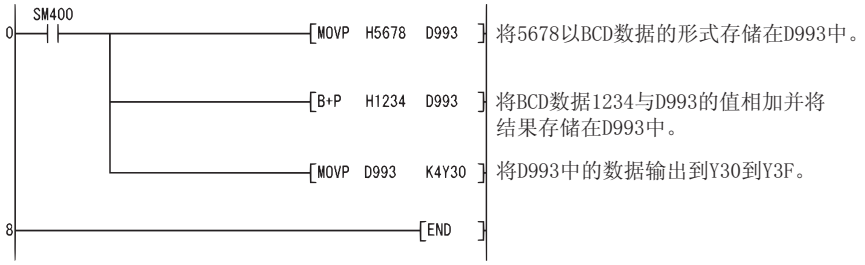
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

•Ⓢ 或 Ⓓ 上的 BCD 数据超出了 0 到 9999 的范围。 （出错代码： 4100）

程序示例

(1) 以下程序将 BCD 数据 5678 与 1234 相加，并将结果存储在 D993 中，同时将它输出到 Y30 到 Y3F。

[梯形图模式]

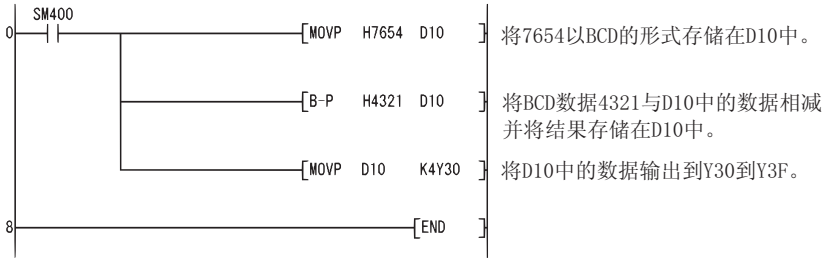


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	MOVP	H5678	D993
3	B+P	H1234	D993
6	MOVP	D993	K4Y30
8	END		

(2) 以下程序将 BCD 数据 4321 从 7654 中减去，并将结果存储住 D10 中，同时将它输出到 Y30 到 Y3F。

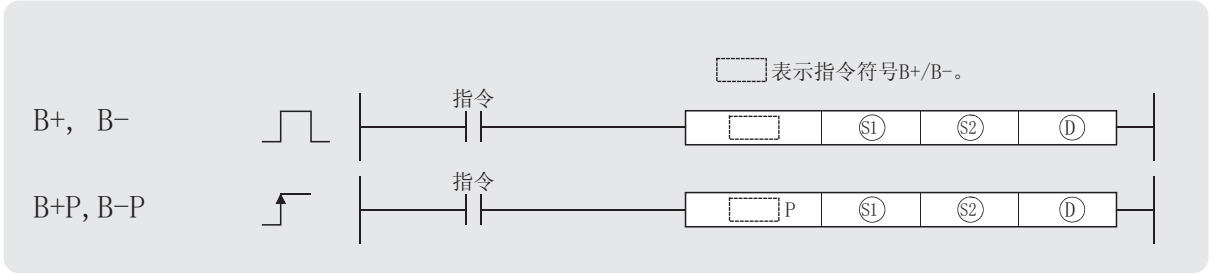
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	MOVP	H7654	D10
3	B-P	H4321	D10
6	MOVP	D10	K4Y30
8	END		

2 当设置了三个数据时（ ⓢ1 + ⓢ2 → Ⓓ、 ⓢ1 - ⓢ2 → Ⓓ）



ⓢ1 ：被加数、被减数或者存储被加数、被减数的软元件的起始号（BCD4 位）。

ⓢ2 ：加数、减数或者存储加数、减数的软元件的起始号（BCD4 位）。

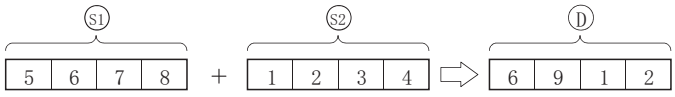
Ⓓ ：存储加法和减法运算结果的软元件的起始号（BCD4 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
ⓢ1				○				○	--
ⓢ2				○				○	--
Ⓓ				○				--	--

功 能

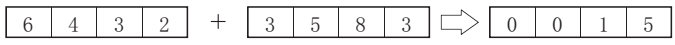
B+

(1) 将 ⓢ1 指定的 BCD 4 位数据与 ⓢ2 指定的 BCD4 位数据相加，并将该加法运算的结果存储在 Ⓓ 指定的软元件中。



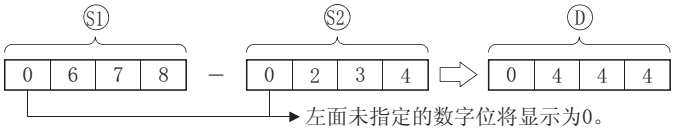
(2) 可以指定给 ⓢ1、 ⓢ2 和 Ⓓ 的值的范围是 0 到 9999(BCD 4 位)。

(3) 如果加法运算的结果超出了 9999，那么其进位将被忽略。在这种情况下，进位标志不变为 ON。



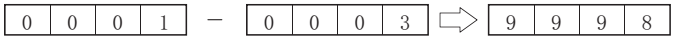
B-

(1) 从 ⓢ1 指定的 BCD 4 位数据中减去 ⓢ2 指定的 BCD 4 位数据，并将该减法运算的结果存储在 Ⓓ 指定的软元件中。



(2) 可以指定给 ⓢ1、 ⓢ2 和 Ⓓ 的值的范围是 0 到 9999(BCD 4 位)。

(3) 如果减法运算产生了下溢，则会导致以下情况：
在这种情况下，进位标志不变为 ON。



运算错误

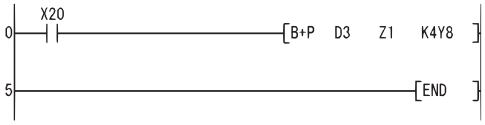
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

•(S1)、 (S2) 或 (D) 上的 BCD 数据超出了 0 到 9999 的范围。 （出错代码： 4100）

程序示例

(1) 当 X20 变为 ON 时，以下程序将 D3 中的 BCD 数据与 Z1 上的 BCD 数据相加，并将其结果输出到 Y8 到 Y17。

[梯形图模式]

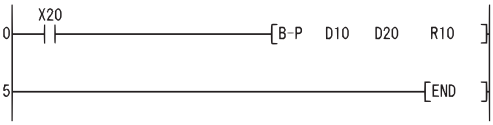


[列表模式]

步	指令	软元件		
0	LD	X20		
1	B+P	D3	Z1	K4Y8
5	END			

(2) 当 X20 变为 ON 时，以下程序将 D20 中的 BCD 数据从 D10 上的 BCD 数据中减去，并将其结果存储在 R10 中。

[梯形图模式]



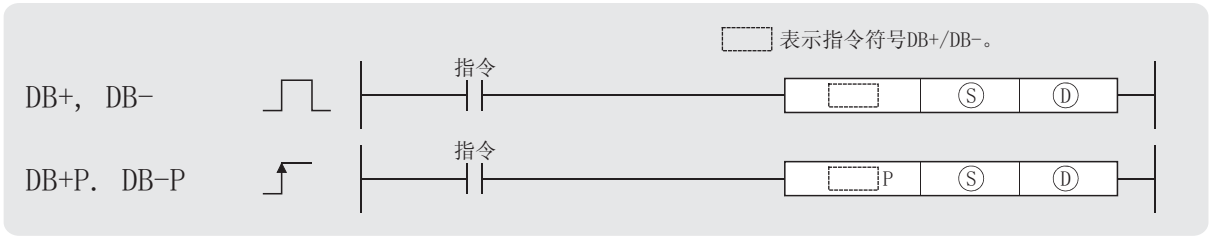
[列表模式]

步	指令	软元件		
0	LD	X20		
1	B-P	D10	D20	R10
5	END			

6.2.6 BCD 8 位加法和减法运算 (DB+(P)、DB-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时 ((Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ)、 (Ⓓ+1、Ⓓ)-(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ))



Ⓢ : 加数、减数或者存储加数、减数的软元件的起始号 (BCD8 位)。

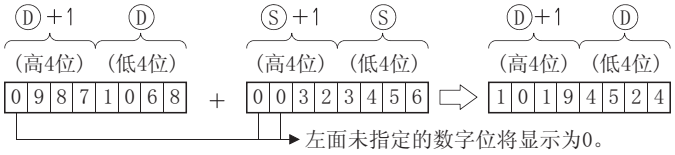
Ⓓ : 加数、被减数或者存储被加数、被减数的软元件的起始号 (BCD8 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	--
Ⓓ	○							--	--

功 能

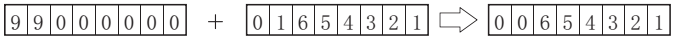
DB+

(1) 将 Ⓓ 指定的 BCD 8 位数据与 Ⓢ 指定的 BCD 8 位数据相加，并将该加法运算的结果存储在 Ⓓ 指定的软元件中。



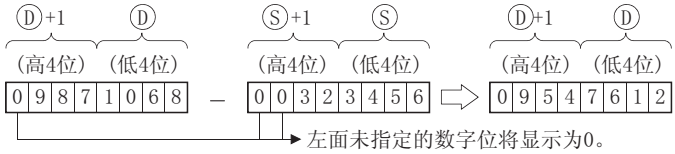
(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 0 到 99999999(BCD 8 位)。

(3) 如果加法运算的结果超出了 99999999，那么其进位将被忽略。
在这种情况下，进位标志不变为 ON。



DB-

(1) 从 Ⓓ 指定的 BCD 8 位数据中减去 Ⓢ 指定的 BCD8 位数据，并将该减法运算的结果存储在 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值的范围是 0 到 99999999(BCD 8 位)。

(3) 如果减法运算产生了下溢，则会导致以下情况：在这种情况下，进位标志不变为 ON。

1	2	3	4	5	6	7	8	−	1	2	3	4	5	6	7	9	⇨	9	9	9	9	9	9	9	9

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- Ⓢ 或 Ⓓ 上的 BCD 数据超出了 0 到 99999999 的范围。(出错代码：4100)

程序示例

(1) 以下程序将 BCD 数据 12345600 与 34567000 相加，并将结果存储在 D887 和 D888 中，同时将它们输出到 Y30 到 Y4F。

[梯形图模式]

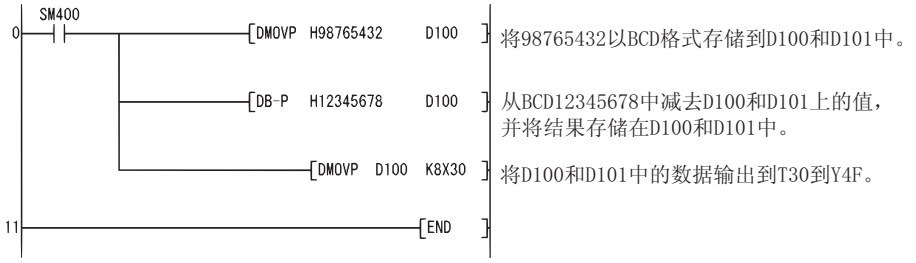


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DMOVP	H12345600	D887
4	DB+P	H34567000	D887
8	DMOVP	D887	K8Y30
11	END		

(2) 以下程序从 12345678 中减去 BCD 数据 98765432，并将结果存储在 D100 和 D101 中，同时将它输出到 Y30 到 Y4F。

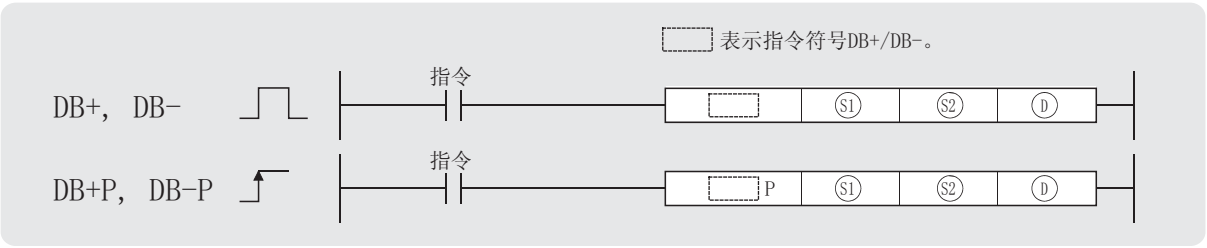
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DMOVP	H98765432	D100
4	DB-P	H12345678	D100
8	DMOVP	D100	K8X30
11	END		

2 当设置了三个数据时（(S1)+1、(S1)）+（(S2)+1、(S2)）→（(D)+1、(D)）、（(S1)+1、(S1)）-（(S2)+1、(S2)）→（(D)+1、(D)））



(S1) ：被加数、被减数或者存储被加数、被减数的软元件的起始号（BCD8 位）。

(S2) ：加数、减数或者存储加数、减数的软元件的起始号（BCD8 位）。

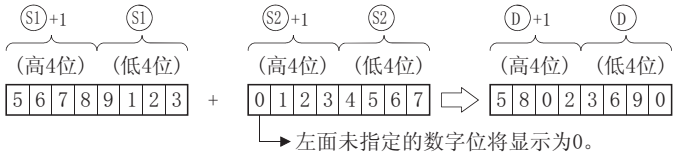
(D) ：存储加法和减法运算结果的软元件的起始号（BCD8 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)				○				○	--
(S2)				○				○	--
(D)				○				--	--

功 能

DB+

(1) 将 (S1) 指定的 BCD 8 位数据与 (S2) 指定的 BCD8 位数据相加，并将该加法运算的结果存储在 (D) 指定的软元件中。



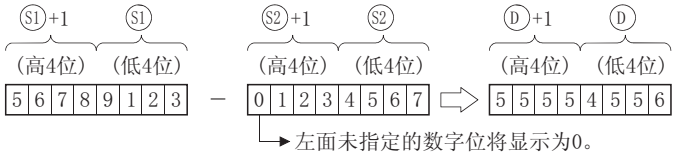
(2) 可以指定给 (S1)、(S2) 和 (D) 的值的范围是 0 到 99999999(BCD 8 位)。

(3) 如果加法运算的结果超出了 99999999，那么其进位将被忽略。
在这种情况下，进位标志不变为 ON。



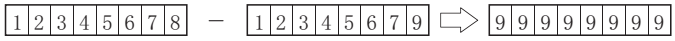
DB-

(1) 从 (S1) 指定的 BCD 8 位数据中减去 (S2) 指定的 BCD8 位数据，并将该减法运算的结果存储在 (D) 指定的软元件中。



(2) 可以指定给 (S1)、(S2) 和 (D) 的值的范围是 0 到 99999999(BCD 8 位)。

(3) 如果减法运算产生了下溢，则会导致以下情况：
在这种情况下，进位标志不变为 ON。



运算错误

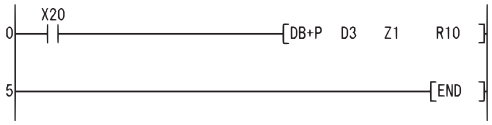
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- (S1)、(S2) 或 (D) 上的 BCD 数据超出了 0 到 99999999 的范围。(出错代码：4100)

程序示例

(1) 当 X20 变为 ON 时，以下程序将 D3、D4 中的 BCD 数据与 Z1、Z2 中的 BCD 数据相加，并将其结果存储在 R10 和 R11 中。

[梯形图模式]

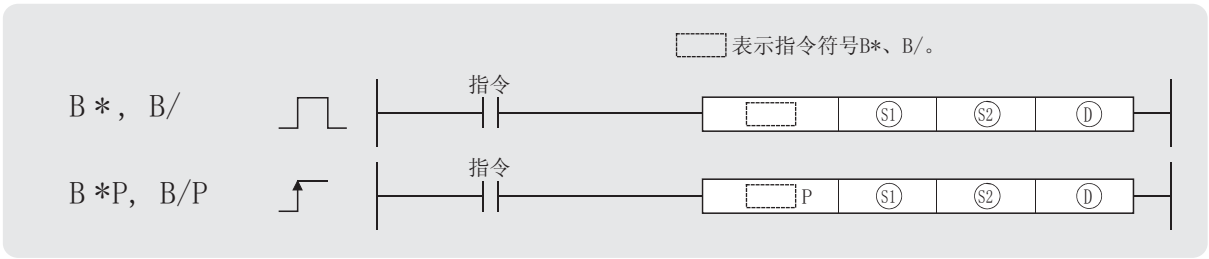


[列表模式]

步	指令	软元件		
0	LD	X20		
1	DB+P	D3	Z1	R10
5	END			

6.2.7 BCD 4 位乘法和除法运算 (B*(P)、B/(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：被乘数、被除数或者存储被乘数、被除数的软元件的起始号 (BCD4 位)。

(S2) ：乘数、除数或者存储乘数、除数的软元件的起始号 (BCD4 位)。

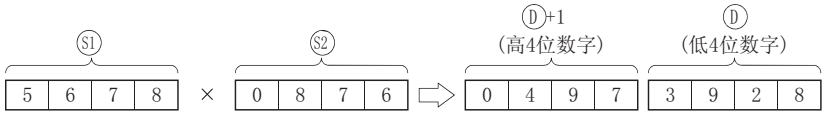
(D) ： 存储乘法、除法运算的运算结果的软元件的起始号 (BCD8 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○							○	--
(S2)	○							○	--
(D)	○							--	--

功 能

B*

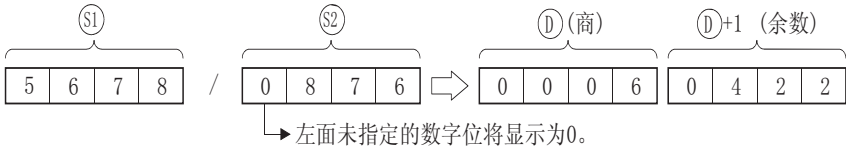
(1) 将 (S1) 指定的 BCD 数据与 (S2) 指定的 BCD 数据相乘，并将其结果存储在 (D) 指定的软元件中。



(2) 可以指定给 (S1) 和 (S2) 的值的范围是 0 到 9999(BCD 4 位)。

B/

(1) 用 (S1) 指定的 BCD 数据除以 (S2) 指定的 BCD 数据，并将其结果存储在 (D) 指定的软元件中。



(2) 使用 32 位数据来存储除法运算的结果： 商和余数

商 (BCD 4 位)　　：存储在低 16 位

余数 (BCD 4 位)　　：存储在高 16 位

(3) 如果已把 (D) 指定为位软元件，那么将不存储除法运算产生的余数。

运算错误

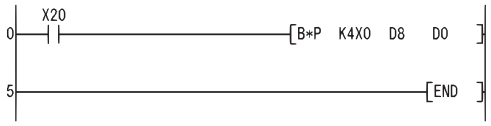
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- ⓢ1 或 ⓢ2 上的数据超出了 0 到 9999 的范围。(出错代码： 4100)
- 试图用 0 去除 ⓢ2 。(出错代码： 4100)

程序示例

(1) 当 X0B 变为 ON 时，以下程序将 X0 到 XF 中的 BCD 数据与 D8 中的 BCD 数据相乘，并将运算结果存储在 D0 和 D1 中。

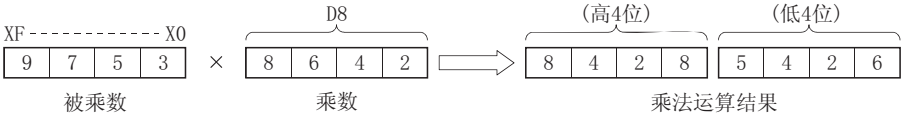
[梯形图模式]



[列表模式]

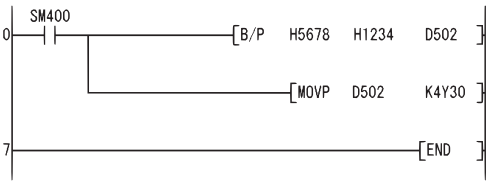
步	指令	软元件		
0	LD	X20		
1	B*P	K4X0	D8	D0
5	END			

[动作]



(2) 以下程序用 5678 除以 BCD 数据 1234，并将其结果存储在 D502 和 D503 中，同时将商输出到 Y30 到 Y3F。

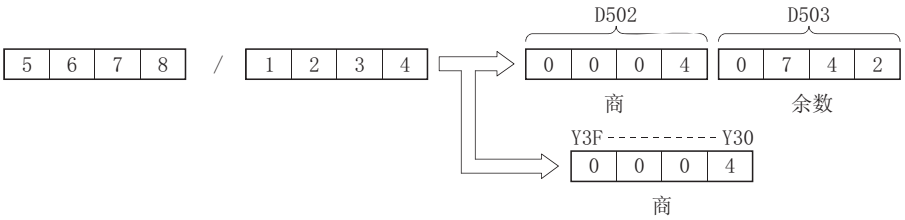
[梯形图模式]



[列表模式]

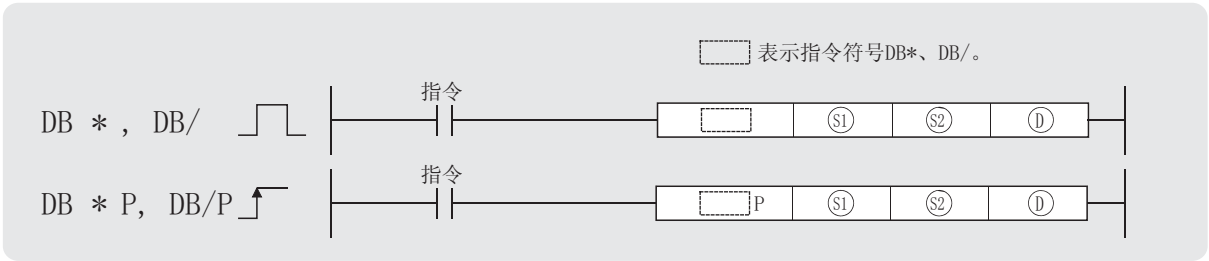
步	指令	软元件		
0	LD	SM400		
1	B/P	H5678	H1234	D502
5	MOVP	D502	K4Y30	
7	END			

[动作]



6.2.8 BCD 8 位乘法和除法运算 (DB*(P)、DB/(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：被乘数、被除数或者存储被乘数、被除数的软元件的起始号 (BCD8 位)。

(S2) ：乘数、除数或者存储乘数、除数的软元件的起始号 (BCD8 位)。

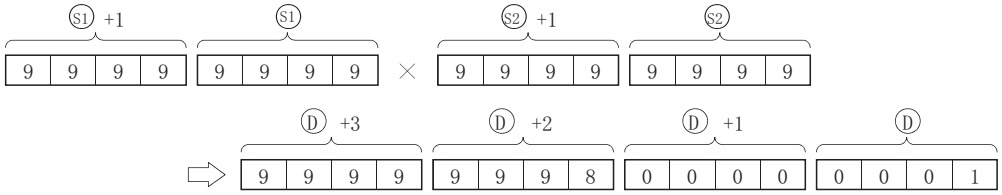
(D) ： 存储乘法、除法运算的运算结果的软元件的起始号 (BCD16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)		○				○			--
(S2)		○				○			--
(D)		○				--			--

功 能

DB*

(1) 将 (S1) 指定的 BCD 8 位数据与 (S2) 指定的 BCD 8 位数据相乘，并将其乘积存储在 (D) 指定的软元件中。



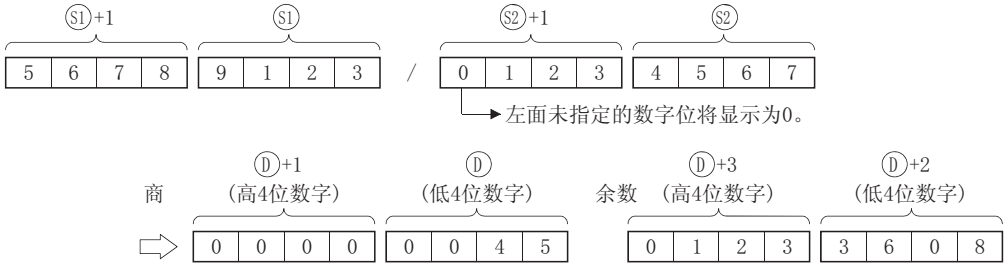
(2) 如果已把 (D) 指定为位软元件，那么乘积将使用低 8 位数字 (低 32 位)，而高 8 位数字 (高 32 位) 不会被指定。

K1.. 低 1 位数字 (b0 到 3)、 K4. 低 4 位数字 (b0 到 b15)、 K8 .. 低 8 位数字 (b0 到 b31)

(3) 可以指定给 (S1) 和 (S2) 的值的范围是 0 到 99999999(8 位 BCD)。

DB/

(1) 用 (S1) 指定的 8 位 BCD 数据除以 (S2) 指定的 8 位 BCD 数据，并将其结果存储在 (D) 指定的软元件中。



(2) 除法运算结果使用 64 位，并以商和余数的形式存储。

商 (BCD 8 位) ：存储在低 32 位
余数 (BCD 8 位) ：存储在高 32 位

(3) 如果已把 ⓓ 指定为位软元件，那么将不存储除法运算产生的余数。

运算错误

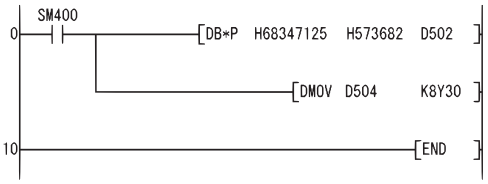
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- ⓢ1 或 ⓢ2 上的 BCD 数据超出了 0 到 99999999 的范围。(出错代码： 4100)
- 试图用 0 去除 ⓢ2 。(出错代码： 4100)

程序示例

(1) 以下程序将 BCD 数据 67347125 与 573682 相乘，并将其结果存储在 D502 到 D505 中，同时把它的高 8 位数字输出到 Y30 到 Y4F。

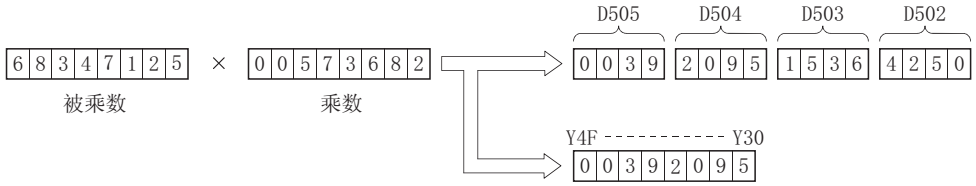
[梯形图模式]



[列表模式]

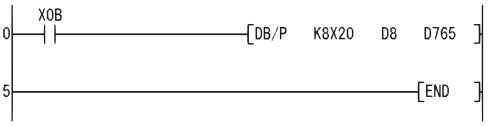
步	指令	软元件
0	LD	SM400
1	DB*P	H68347125 H573682 D502
7	DMOV	D504 K8Y30
10	END	

[动作]



(2) 当 X0B 变为 ON 时，以下程序用 X20 到 X3F 上的 BCD 数据除以 D8 到 D9 中的 BCD 数据，并将其结果存储在 D765 到 D768 中。

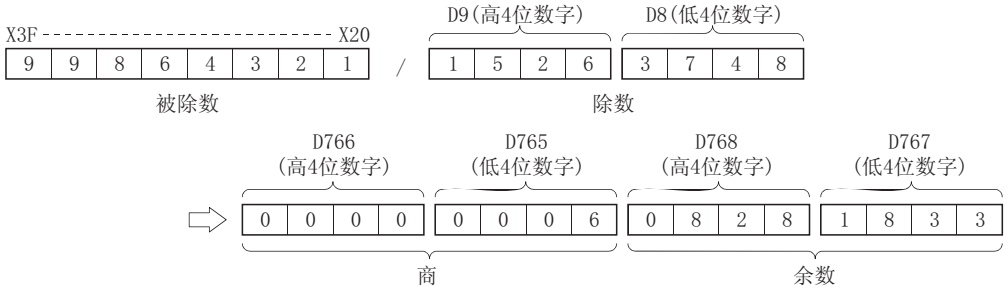
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0B
1	DB/P	K8X20 D8 D765
5	END	

[动作]

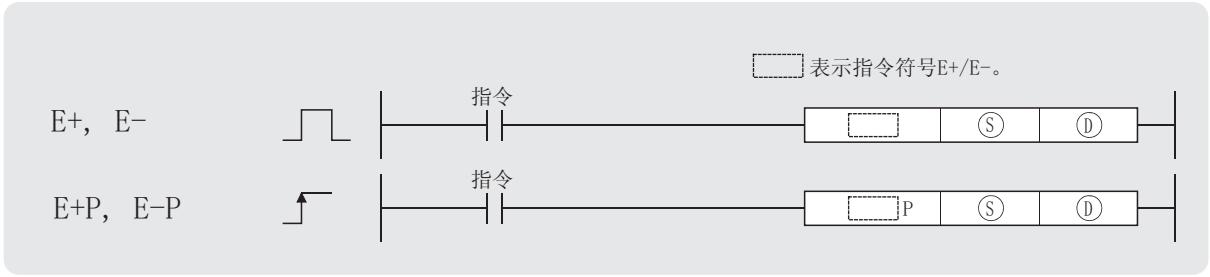


6.2.9 浮点数据的加法和减法运算 (E+(P)、E-(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。

1 当设置了两个数据时（(Ⓓ+1、Ⓓ)+(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ)、(Ⓓ+1、Ⓓ)-(Ⓢ+1、Ⓢ)→(Ⓓ+1、Ⓓ)）



Ⓢ ：加数、减数或者存储加数、减数的软元件的起始号（实数）。

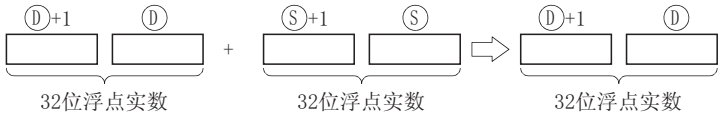
Ⓓ ：存储被加数、被减数的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--	○		--	○	--
Ⓓ	--	○		--	○		--	--	--

功 能

E+

(1) 将 Ⓓ 指定的 32 位浮点类型实数与 Ⓢ 指定的 32 位浮点类型实数相加，并将它们的和存储在 Ⓓ 指定的软元件中。

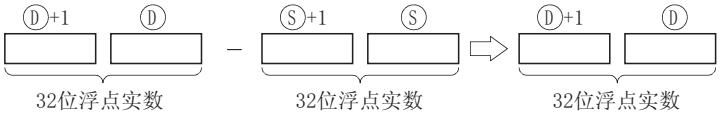


(2) 可以指定给 Ⓢ 和 Ⓓ 的值以及可以存储的值的范围如下：

0、$2^{-126} \leqq |$ 指定值（存储值）$| < 2^{128}$

E-

(1) 将 Ⓓ 指定的 32 位浮点类型实数与 Ⓢ 指定的 32 位浮点类型实数相减，并将其结果存储在 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ 和 Ⓓ 的值以及可以存储的值的范围如下：

0、$2^{-126} \leqq |$ 指定值（存储值）$| < 2^{128}$

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 指定软元件的内容或加法，减法运算的结果不为 “0”，或者不在下列范围之内：

0、$2^{-126} \leqq |$ 指定软元件的内容 $| < 2^{128}$

（对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU 和 Q4ARCPU）

（出错代码： 4100）

• 指定软元件的值为 -0 时。*2

（对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU）

（出错代码： 4100）

2: 有的 CPU 模块即使指定了 -0，也不会发生运算错误。
有关详细信息，请参阅 3.2.4 项。

• 加法、减法的结果超出了以下范围时。（发生了上溢时。）

（仅对于通用型 QCPU）

$2^{128} \leqq |$ 加法减法结果 $|$ （出错代码： 4141）

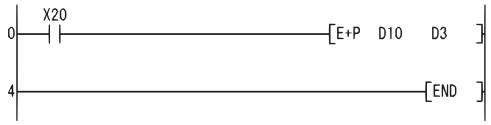
• 指定软元件的内容为－0、非正规数、非数、±∞时。

（仅对于通用型 QCPU） （出错代码： 4140）

程序示例

(1) 当 X20 变为 ON 时，以下程序将 D3 和 D4 中的浮点类型实数与 D10 和 D11 中的浮点类型实数相加，并将其结果存储在 D3 到 D4 中。

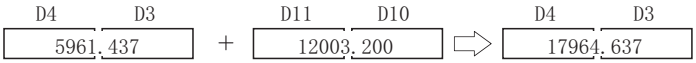
[梯形图模式]



[列表模式]

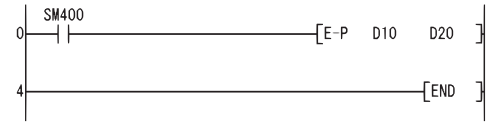
步	指令	软元件	
0	LD	X20	
1	E+P	D10	D3
4	END		

[动作]



(2) 以下程序从 D20 和 D21 上的浮点类型实数中将 D10 和 D11 中的浮点类型实数减去，并将该减法运算的结果存储在 D20 和 D21 中。

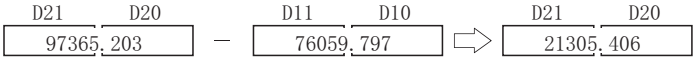
[梯形图模式]



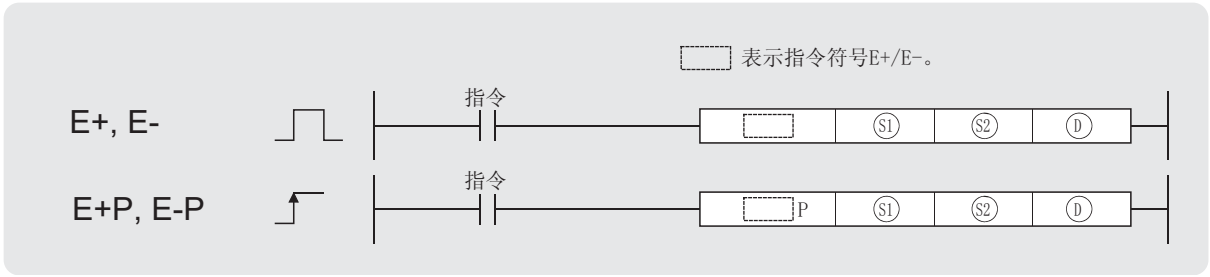
[列表模式]

步	指令	软元件	
0	LD	SM400	
1	E-P	D10	D20
4	END		

[动作]



2 当设置了三个数据时（(S1)+1、(S1)）+(S2)+1、(S2))→(D)+1、(D))、（(S1)+1、(S1)）-(S2)+1、(S2))→(D)+1、(D)))



(S1) ：被加数、被减数或者存储被加数、被减数的软元件的起始号（实数）。

(S2) ：加数、减数或者存储加数、减数的软元件的起始号（实数）。

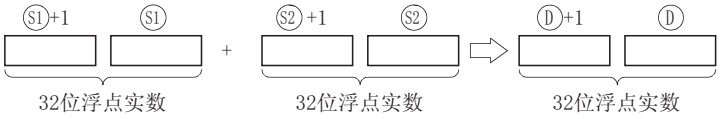
(D) ：存储加法和减法运算结果的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
(S1)	--	○		--	○		--	○	--
(S2)	--	○		--	○		--	○	--
(D)	--	○		--	○		--	--	--

☆功 能

E+

(1) 将 (S1) 指定的 32 位浮点类型实数与 (S2) 指定的 32 位浮点类型实数相加，并将它们的和存储在 (D) 指定的软元件中。

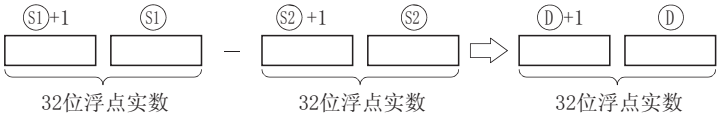


(2) 可以指定给 (S1)、(S2) 或 (D) 的值以及可以存储的值的范围如下：

0、$2^{-126} \leqq$ ｜指定值（存储值）｜$< 2^{128}$

E-

(1) 将 (S1) 指定的 32 位浮点型实数与 (S2) 指定的 32 位浮点型实数相减，并将其结果存储在 (D) 指定的软元件中。



(2) 可以指定给 (S1)，(S2) 或 (D) 的值以及可以存储的值的范围如下：

0、$2^{-126} \leqq$ ｜指定值（存储值）｜$< 2^{128}$

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 指定软元件的值或加法、减法运算的结果不在下列范围之内：

$0、\ 2^{-126} \leqq |$ 指定软元件的内容 $| < 2^{128}$

（对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU 和 Q4ARCPU）

（出错代码： 4100）

• 指定软元件的值为 -0 时。*1

（对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU）

（出错代码： 4100）

1: 有的 CPU 模块即使指定了 -0，也不会发生运算错误。
有关详细信息，请参阅 3.2.4 项。

• 加法、减法的结果超出了以下范围时。（发生了上溢时。）

（仅对于通用型 QCPU）

$2^{128} \leqq |$ 加法减法结果 $|$ （出错代码： 4141）

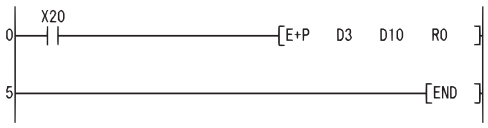
• 指定软元件的内容为 -0、非正规数、非数、$\pm\infty$ 时。

（仅对于通用型 QCPU） （出错代码： 4140）

程序示例

(1) 当 X20 变为 ON 时，以下程序将 D3 和 D4 中的 32 位浮点型实数与 D10 和 D11 中的 32 位浮点型实数相加，并将其结果输出到 R0 和 R1。

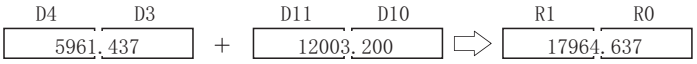
[梯形图模式]



[列表模式]

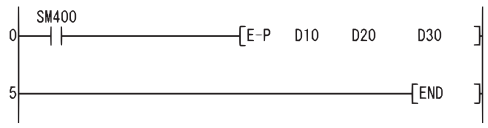
步	指令	软元件		
0	LD	X20		
1	E+P	D3	D10	R0
5	END			

[动作]



(2) 以下程序从 D11 和 D10 的 32 位浮点型实数中减去 D20 和 D21 的 32 位浮点型实数，并将其结果存储在 D30 和 D31 中。

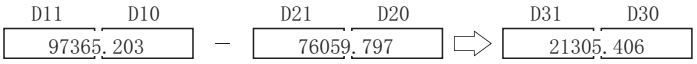
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	SM400		
1	E-P	D10	D20	D30
5	END			

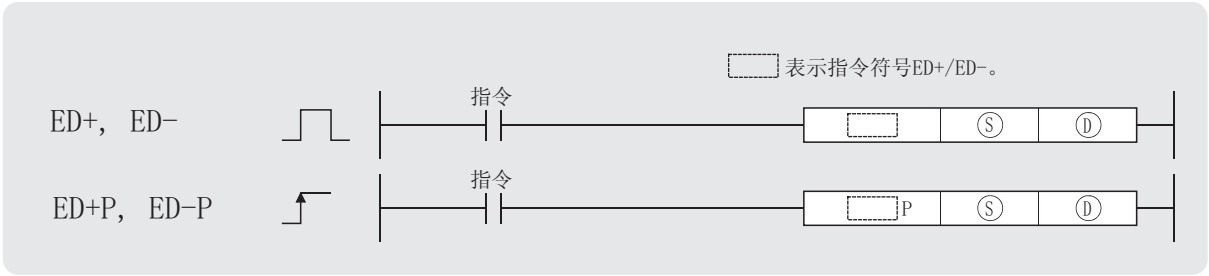
[动作]



6.2.10 浮点数据的加法和减法运算（双精度）(ED+(P)、ED-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时 ((Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)+(Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ)→(Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)、 (Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)-(Ⓢ+3、Ⓢ+2、Ⓢ+1、Ⓢ)→(Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ))



Ⓢ ：加数、减数或者存储加数、减数的软元件的起始号（实数）。

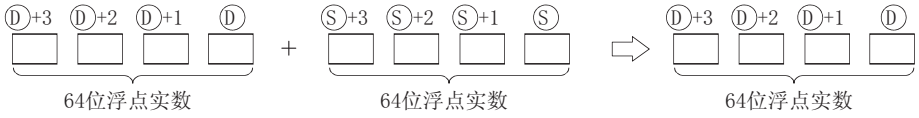
Ⓓ ： 被加数、被减数或者存储被加数、被减数的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

ED+

(1) 将 Ⓓ 指定的 64 位浮点型实数与 Ⓢ 指定的 64 位浮点型实数相加，并将运算结果存储在 Ⓓ 指定的软元件中。

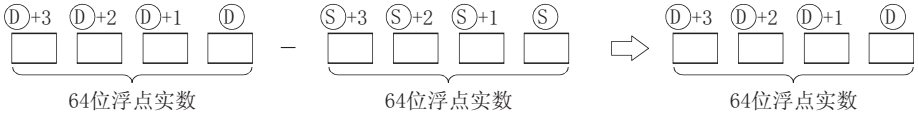


(2) 可以在 Ⓢ 和 Ⓓ 指定的值以及可以存储的值的范围如下：

0、$2^{-1022} \leqq$ | 指定值（存储值）| $< 2^{1024}$

ED-

(1) 将 ⓓ 指定的 64 位浮点型实数与 ⓢ 指定的 64 位浮点型实数相减，并将其结果存储在 ⓓ 指定的软元件中。



(2) 可以在 ⓢ 和 ⓓ 指定的值以及可以存储的值的范围如下：

0、2^{-1022} ≦ ｜指定值（存储值）｜＜ 2^{1024}

运算错误

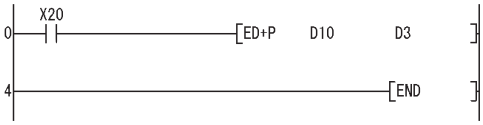
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，并且相关出错代码存储到 SD0 中。

- 指定的软元件的内容或加法结果不为 “0”，或不在以下范围： （出错代码： 4140）
  0、2^{-1022} ≦ ｜指定的软元件的内容｜＜ 2^{1024}
- 指定的软元件的值为 −0 。 （出错代码： 4140）
- 加法 / 减法结果超出以下范围。（运算结果上溢）:
  2^{1024} ≦ ｜运算结果｜ （出错代码： 4141）

程序示例

(1) 当 X20 变为 ON，以下程序将 D3 到 D6 的 64 位浮点型实数与 D10 到 D13 的 64 位浮点型实数相加，并将其结果存储在 D3 到 D6 中。

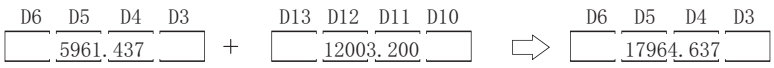
[梯形图模式]



[列表模式]

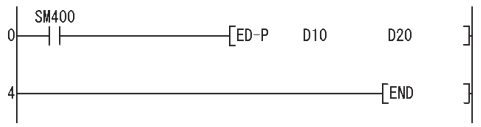
步	指令	软元件	
0	LD	X20	
1	ED+P	D10	D3
4	END		

[动作]



(2) 以下程序将 D20 到 D23 的 64 位浮点型实数与 D10 到 D13 的 64 位浮点型实数相减，并将减法运算结果存储到 D20 到 D23 中。

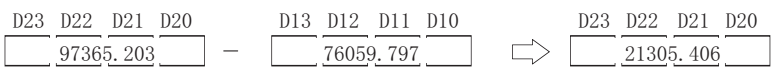
[梯形图模式]



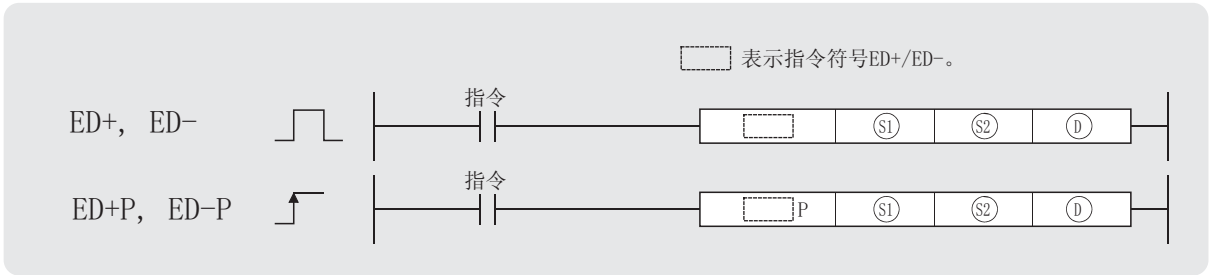
[列表模式]

步	指令	软元件	
0	LD	SM400	
1	ED-P	D10	D20
4	END		

[动作]



2 当设置了三个数据时((Ⓢ1+3、Ⓢ1+2、Ⓢ1+1、Ⓢ1)+(Ⓢ2+3、Ⓢ2+2、Ⓢ2+1、Ⓢ2)→(Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ)、(Ⓢ1+3、Ⓢ1+2、Ⓢ1+1、Ⓢ1)-(Ⓢ2+3、Ⓢ2+2、Ⓢ2+1、Ⓢ2)→(Ⓓ+3、Ⓓ+2、Ⓓ+1、Ⓓ))



Ⓢ1 ：被加数、被减数或者存储被加数、被减数的软元件的起始号（实数）。

Ⓢ2 ：加数、减数或者存储加数、减数的软元件的起始号（实数）。

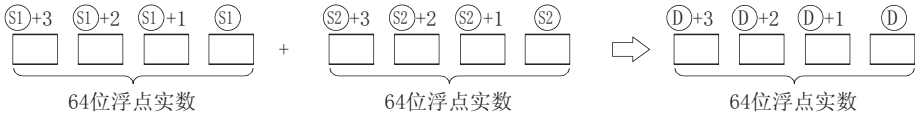
Ⓓ ：存储加法和减法运算结果的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ1	--	○		--				○	--
Ⓢ2	--	○		--				○	--
Ⓓ	--	○		--				--	--

☆功 能

ED+

(1) 将 Ⓢ1 指定的 64 位浮点型实数与 Ⓢ2 指定的 64 位浮点型实数相加，并将结果存储到 Ⓓ 指定的软元件中。

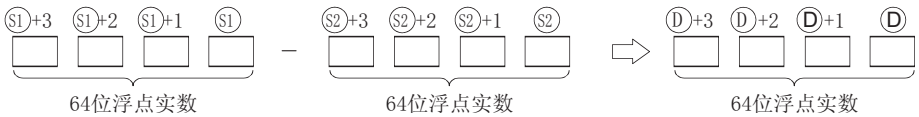


(2) 可以在 Ⓢ1、Ⓢ2 和 Ⓓ 中指定的值以及可以存储的值的范围如下：

0、$2^{-1022} \leqq$ | 指定值（存储值）| $< 2^{1024}$

ED-

(1) 将 Ⓢ1 指定的 64 位浮点型实数与 Ⓢ2 指定的 64 位浮点型实数相减，并将结果存储到 Ⓓ 指定的软元件中。



(2) 可以指定给 Ⓢ1 、Ⓢ2 和 Ⓓ 的值以及可以存储的值的范围如下：

0、$2^{-1022} \leqq$ | 指定值（存储值）| $< 2^{1024}$

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将被存储到 SD0 中。

- 指定的软元件的内容或加法结果不为 “0”，或不在以下范围： （出错代码： 4140）

  0、 2^{-1022} ≦ ｜ 指定的软元件的内容 ｜ < 2^{1024}

- 指定的软元件的值为 −0。 （出错代码： 4140）

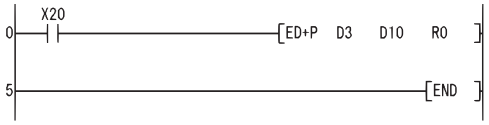
- 加法 / 减法结果超出以下范围 (运算结果上溢)

  2^{1024} ≦ ｜ 运算结果 ｜ （出错代码： 4141）

程序示例

(1) 当 X20 变为 ON，以下程序将 D3 到 D6 的 64 位浮点型实数与 D10 到 D13 的 64 位浮点型实数相加，并将结果输出到 R0 到 R3 中。

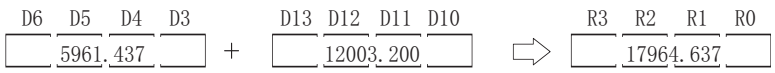
[梯形图模式]



[列表模式]

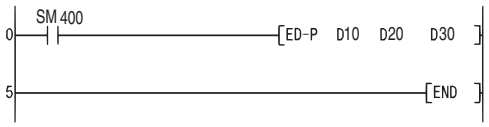
步	指令	软元件		
0	LD	X20		
1	ED+P	D3	D10	R0
5	END			

[动作]



(2) 以下程序将 D10 到 D13 的 64 位浮点型实数与 D20 到 D23 的 64 位浮点型实数相减，并将减法运算结果存储到 D30 到 D33 中。

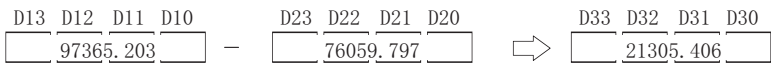
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	SM400		
1	ED-P	D10	D20	D30
5	END			

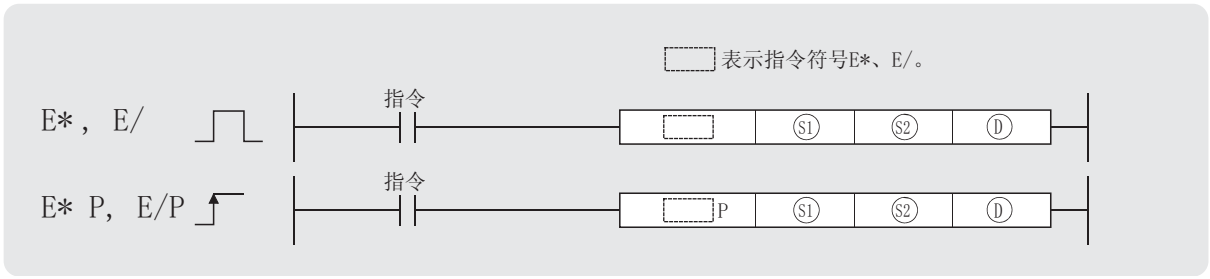
[动作]



6.2.11 浮点数据的乘法和除法运算 (E*(P)、E/(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



(S1)：被乘数、被除数或者存储被乘数、被除数的软元件的起始号（实数）。

(S2)：乘数、除数或者存储乘数、除数的软元件的起始号（实数）。

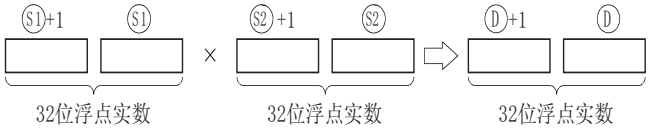
(D)：存储乘法、除法运算的运算结果的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
(S1)	--	○		--	○		--	○	--
(S2)	--	○		--	○		--	○	--
(D)	--	○		--	○		--	--	--

功 能

E*

(1) 将 (S1) 指定的 32 位浮点型实数与 (S2) 指定的 32 位浮点型实数相乘，并将相乘结果存储到(D)指定的软元件中。

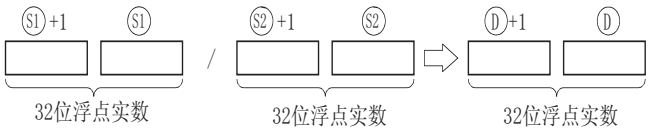


(2) 可以指定给 (S1)、(S2) 或 (D) 的值以及可以存储的值的范围如下：

0、$2^{-126} \leqq$ | 指定值（存储值）| $< 2^{128}$

E/

(1) 用 (S1) 指定的 32 位浮点型实数除以 (S2) 指定的 32 位浮点型实数，并将其结果存储到 (D) 指定的软元件中。



(2) 可以指定给 ⓢ1、 ⓢ2 或 ⓓ 的值以及可以存储的值的范围如下：

0、 2^{-126} ≦ ｜指定值（存储值）｜ < 2^{128}

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定软元件的值或乘法、除法运算的结果不在下列范围之内：

  0、 2^{-126} ≦ ｜指定软元件的内容｜ < 2^{128}

  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU、QnACPU 和 Q4ARCPU）

  （出错代码： 4100）

- 指定软元件的值为 -0 时。 *2

  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU）

  （出错代码： 4100）

  2: 有的 CPU 模块即使指定了 -0，也不会发出运算出错信息。
  有关详细信息，请参阅 3.2.4 项。

- 乘法和除法结果超出以下范围。（发生上溢。）（只对于通用型 QCPU）

  2^{128} ≦ ｜乘法和除法结果｜ （出错代码： 4141）

- 指定的值为 -0、非正规数、非数和±∞。

  （只对于通用型 QCPU） （出错代码： 4140）

程序示例

(1) 以下程序在 X20 为 ON 时，将 D3 和 D4 中的 32 位浮点型实数与 D10 和 D11 中的 32 位浮点型实数相乘，并将其结果存储在 R0 和 R1 中。

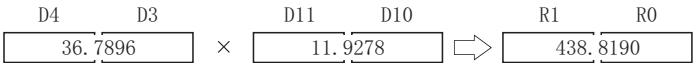
[梯形图模式]



[列表模式]

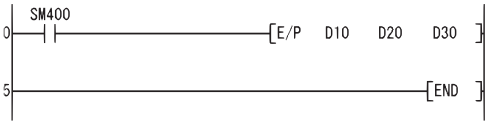
步	指令	软元件		
0	LD	X20		
1	E*P	D3	D10	R0
5	END			

[动作]



(2) 以下程序用 D20 和 D21 中的浮点类型实数去除 D10 和 D11 中的浮点类型实数，并将其结果存储在 D30 和 D31 中。

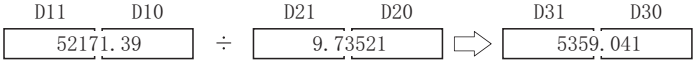
[梯形图模式]



[列表模式]

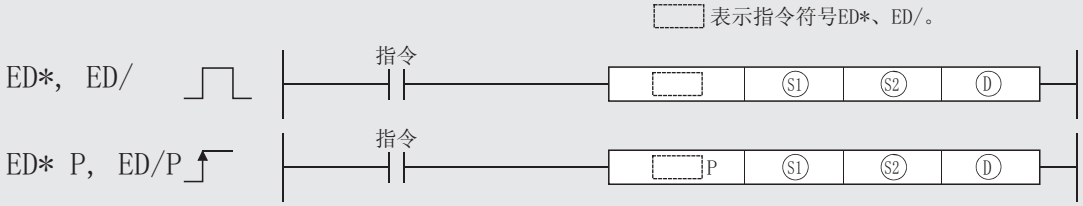
步	指令	软元件		
0	LD	SM400		
1	E/P	D10	D20	D30
5	END			

[动作]



6.2.12 浮点数据的乘法和除法运算（双精度）(ED*(P)、ED/(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：被乘数、被除数或者存储被乘数、被除数的软元件的起始号（实数）。

(S2) ：乘数、除数或者存储乘数、除数的软元件的起始号（实数）。

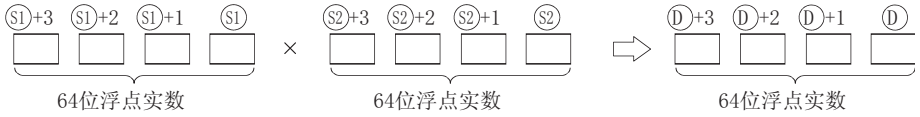
(D) ：存储乘法、除法运算的运算结果的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
(S1)	--	○		--				○	--
(S2)	--	○		--				○	--
(D)	--	○		--				--	--

功 能

ED*

(1) 将 (S1) 指定的 64 位浮点型实数与 (S2) 指定的 64 位浮点型实数相乘，并将运算结果存储在 (D) 指定的软元件中。



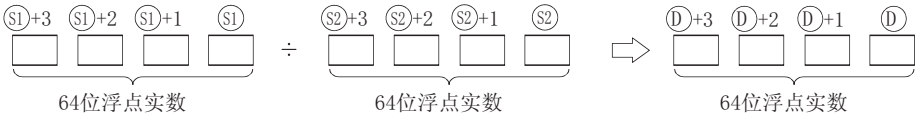
(2) 可以指定给 (S1)、(S2) 和 (D) 的值以及可以存储的值的范围如下：

0、$2^{-1022} \leqq$ | 指定值（存储值）| $< 2^{1024}$

(3) 当运算结果为 -0 或下溢时，结果作为 0 处理。

ED/

(1) 用 (S1) 指定的 64 位浮点型实数除以 (S2) 指定的 64 位浮点型实数，并将其运算结果存储在 (D) 指定的软元件中。



(2) 可以指定给 (S1)、(S2) 和 (D) 的值以及可以存储的值的范围如下：

0、2^{-1022} ≦ ｜指定值（存储值）｜ < 2^{1024}

(3) 当运算结果为 -0 或下溢时，结果做为 0 处理。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0 中。

- 指定的软元件的值或乘法结果不在以下范围：（出错代码：4140）

  0、2^{-1022} ≦ ｜指定的软元件的内容｜< 2^{1024}

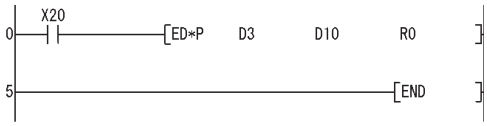
- 指定的软元件的值为 -0。（出错代码：4140）
- 除法运算的 (S2) 的值为 0。（出错代码：4100）
- 乘法 / 除法结果超出以下范围（运算结果上溢）：

  2^{1024} ≦ ｜乘法除法运算结果｜（出错代码：4141）

程序示例

(1) 以下程序在 X20 为 ON 时，将 D3 到 D6 的 64 位浮点型实数与 D10 到 D13 的 64 位浮点型实数相乘，并将其结果存储在 R0 到 R3 中。

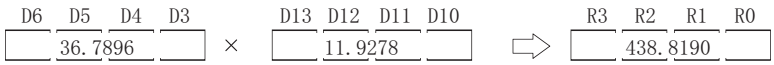
[梯形图模式]



[列表模式]

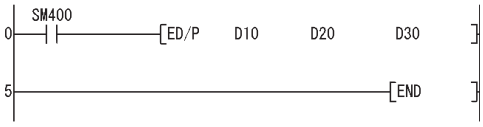
步	指令	软元件
0	LD	X20
1	ED*P	D3 D10 R0
5	END	

[动作]



(2) 以下程序用 D10 到 D13 的 64 位浮点型实数除以 D20 到 D23 的 64 位浮点型实数，并将运算结果存储在 D30 到 D33 中。

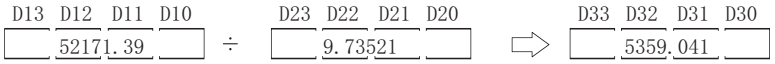
[梯形图模式]



[列表模式]

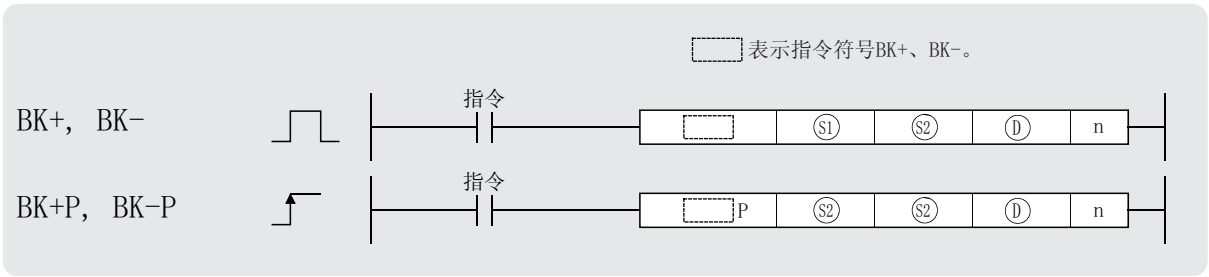
步	指令	软元件		
0	LD	SM400		
1	ED/P	D10	D20	D30
5	END			

[动作]



6.2.13 块加法和减法运算 (BK+(P)、BK-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) : 被加数、被减数或者存储被加数、被减数的软元件的起始号 (BIN16 位)。)

(S2) : 加数、减数或者存储加数、减数的软元件的起始号 (BIN16 位)。

(D) : 存储运算结果的软元件的起始号 (BIN16 位)。

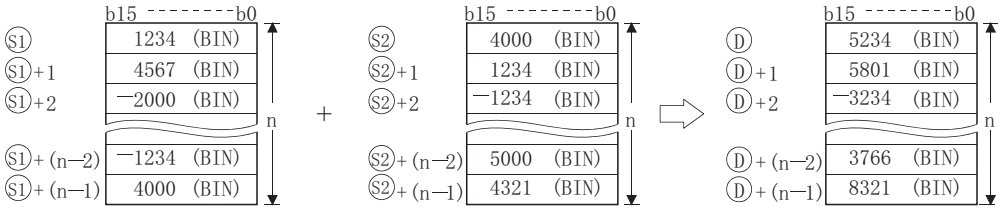
n : 加 / 减数据块的数量 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
(S1)	--	○		--				--	--
(S2)	--	○		--				○	--
(D)	--	○		--				--	--
n	○	○		○				○	--

功 能

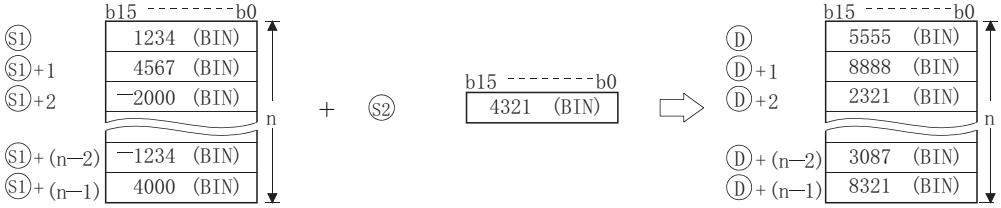
BK+

(1) 将从 (S1) 指定的软元件开始的 n 点 BIN 数据与从 (S2) 指定的软元件开始的 n 点 BIN 数据相加，并将其结果存储在以 (D) 指定的软元件中。



(2) 块加法运算在 16 位单元中进行。

(3) 可以指定给 (S2) 的常数的范围是 −32768 到 32767(BIN 16 位)。



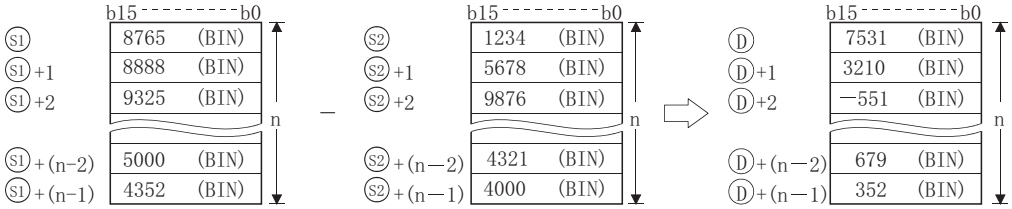
(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K32767 +K2 ⟶ K－32767
(7FFFH) (0002H) (8001H)

· K－32767 +K－2 ⟶ K32767
(8001H) (FFFEH) (7FFFH)

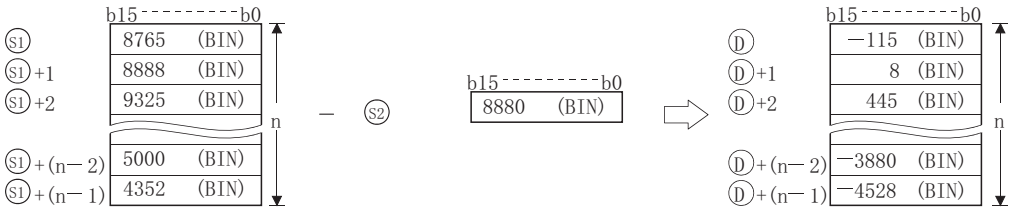
BK-

(1) 从 (S1) 指定的软元件开始的 n 点 BIN 数据中减去从 (S2) 指定的软元件开始的 n 点 BIN 数据，并将其结果存储在以 (D) 指定的软元件中。



(2) 块减法运算在 16 位单元中进行。

(3) 可以指定给 (S2) 的常数的范围是 -32768 到 32767(BIN 16 位)。



(4) 如果运算结果中产生了下溢或者上溢，则会发生以下情况：
在这种情况下，进位标志不变为 ON。

· K－32768 －K2 ⟶ K32766
(8000H) (0002H) (7FFEH)

· K32767 －K－2 ⟶ －32767
(7FFFH) (FFFEH) (8001H)

运算错误

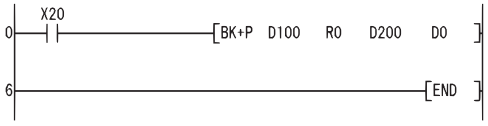
(1) 在下述情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 (S1)、(S2) 或 (D) 软元件起始的 n 位范围超出了相应软元件的范围。
(出错代码：4101)
- (S1) 和 (D) 的软元件范围重叠。(除给 (S1) 和 (D) 分配了相同软元件时以外。)
(出错代码：4101)
- (S2) 和 (D) 的软元件范围重叠。(除给 (S2) 和 (D) 分配了相同软元件时以外。)
(出错代码：4101)

程序示例

(1) 当 X20 变为 ON 时，以下程序将 D100 到 D103 的数据与存储在 R0 到 R3 的数据相加，并将运算结果存储在起始于 D200 的区域中。

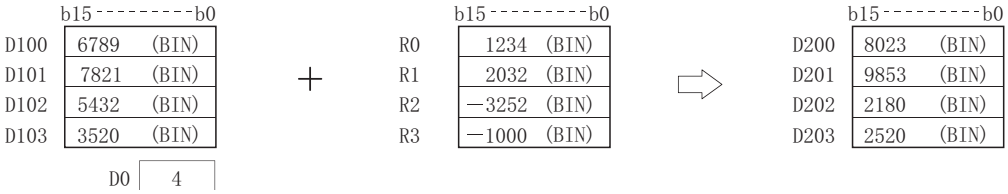
[梯形图模式]



[列表模式]

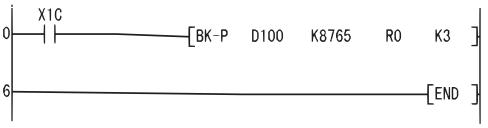
步	指令	软元件			
0	LD	X20			
1	BK+P	D100	R0	D200	D0
6	END				

[动作]



(2) 当 X1C 变为 ON 时，以下程序从 D100 到 D102 的数据中减去常数 8765，并将减法运算结果存储到起始于 R0 的区域中。

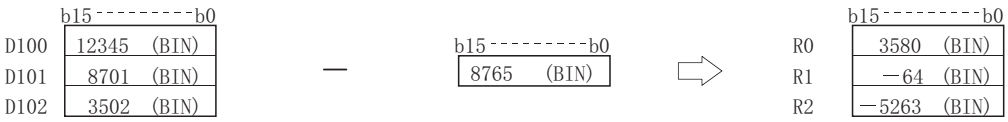
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X1C			
1	BK-P	D100	K8765	R0	K3
6	END				

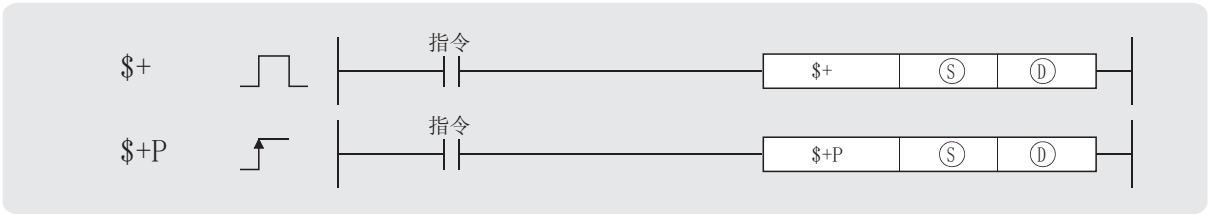
[动作]



6.2.14 链接字符串（$+(P)）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（Ⓓ+Ⓢ → Ⓓ）



Ⓢ ： 链接数据或存储链接数据的软元件的起始号（字符串）。

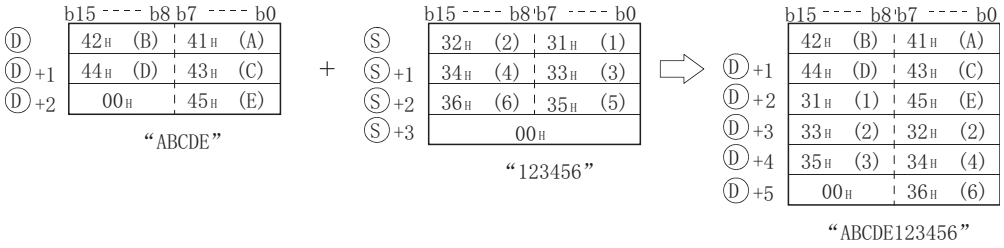
Ⓓ ： 存储被链接数据的软元件的起始号（字符串）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

(1) 将从 Ⓢ 指定的软元件号开始的软元件中存储的字符串数据追加到从 Ⓓ 指定的软元件中存储的字符串数据的后面，并存储在从 Ⓓ 指定的软元件号开始的软元件中。

该字符串数据对象是指从 Ⓓ 和 Ⓢ 指定的软元件号开始到存储 “00H”的软元件号为止的软元件中存储的字符串数据。



(2) 当链接字符串的时候，表示 Ⓓ 指定的字符串数据的结束的 “00H”将被忽略，并且，将 Ⓢ 指定的字符串数据追加到 Ⓓ 字符串的最后一个字符之后。

运算错误

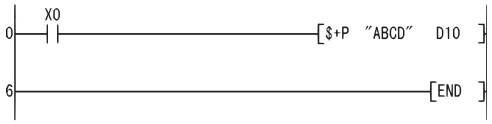
(1) 在下述情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 不能完整存储从 ⓓ 指定的软元件号到相关软元件的最终软元件号之间链接的整个字符串。 (出错代码： 4101)
- ⓢ 与 ⓓ 指定的字符串的存储软元件号相互重叠。 (出错代码： 4101)
- ⓢ 与 ⓓ 的字符串超出 16383 字符。 (出错代码： 4101)

程序示例

(1) 当 X0 变为 ON 时，以下程序将存储在 D10 到 D12 中的字符串与字符串 “ABCD”链接起来。

[梯形图模式]



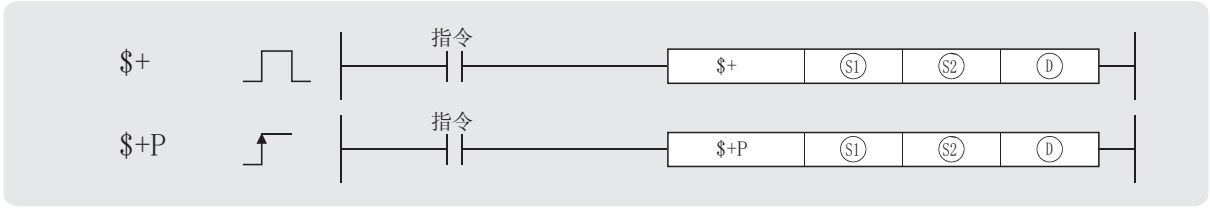
[列表模式]

步	指令	软元件
0	LD	X0
1	$+P	"ABCD" D10
6	END	

[动作]



2 当设置了三个数据时（(S1)+(S2) → (D)）



(S1) ： 链接数据或存储链接数据的软元件的起始号（字符串）。

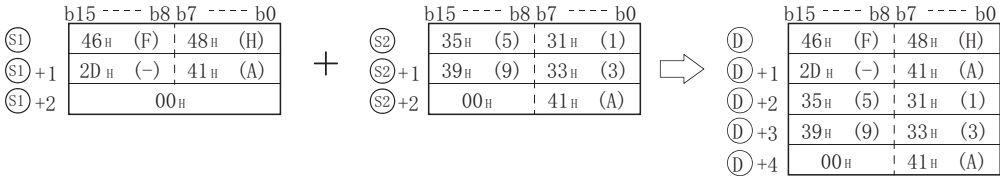
(S2) ： 被链接数据或存储被链接数据的软元件的起始号（字符串）。

(D) ： 存储链接结果的软元件的起始号（字符串）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
(S1)	--	○		--				○	--
(S2)	--	○		--				○	--
(D)	--	○		--				--	--

功 能

(1) 将从 (S2) 指定的软元件中存储的字符串数据追加到从 (S1) 指定的软元件中存储的字符串数据的后面，并存储在从 (D) 指定的软元件中。



(2) 当链接字符串时，表示 (S1) 指定的字符串数据的结束的“00H”将被忽略， (S2) 指定的字符串数据将被连接到 (S1) 字符串的最后一个字符后。

运算错误

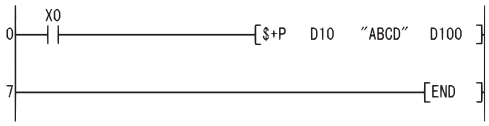
(1) 在以下情况下将发生运算错误，且错误标志将变为 ON。

- 不能完整存储从 ⓓ 指定的软元件号到相关软元件的最终软元件号之间的链接的整个字符串。 （出错代码： 4101）
- ⓢ1 与 ⓢ2 指定的字符串的存储软元件号相互重叠。 （出错代码： 4101）
- ⓢ2 与 ⓓ 指定的字符串的存储软元件号相互重叠。 （出错代码： 4101）
- ⓢ1、ⓢ2 与 ⓓ 的字符串超出 16383 字符。 （出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序将存储在 D10 到 D12 中的字符串与字符串 “ABCD”链接起来，并将它们存储在从 D100 开始的软元件中。

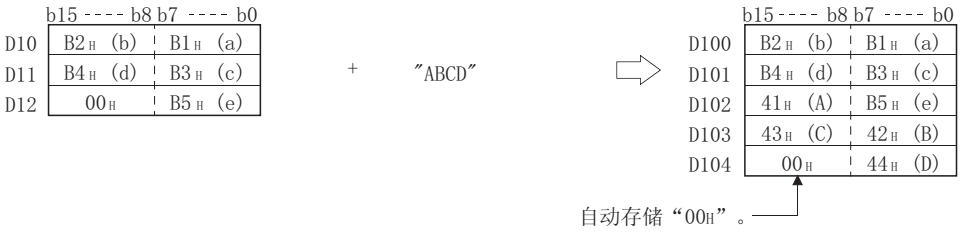
[梯形图模式]



[列表模式]

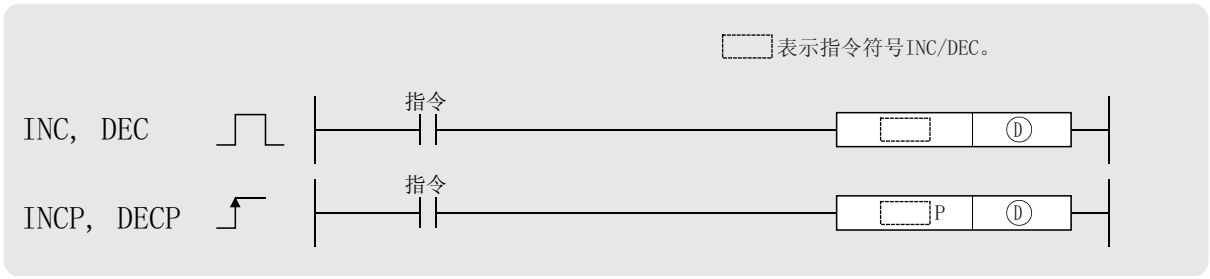
步	指令	软元件	
0	LD	X0	
1	$+P	D10	"ABCD" D100
7	END		

[动作]



6.2.15 16 位 BIN 数据的递增和递减运算（INC(P)、DEC(P)）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ ：执行 INC(+1)/DEC(-1) 运算的软元件的起始号。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ		○						--	

功 能

INC

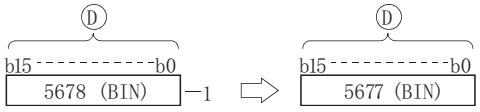
(1) 将由 Ⓓ 指定的软元件（16 位数据）加 1。



(2) 如果 Ⓓ 指定的软元件的内容是 32767，并且在该软元件上执行了 INC 或 INCP 指令，那么值 32768 将存储在 Ⓓ 指定的软元件中。

DEC

(1) 从 Ⓓ 指定的软元件（16 位数据）中减 1。



(2) 如果 Ⓓ 指定的软元件的内容是 0，并且在该软元件上执行了 DEC 或 DECP 指令，那么值 -1 将存储在 Ⓓ 指定的软元件中。

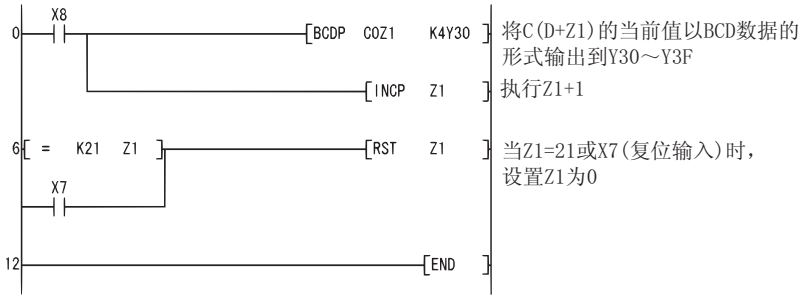
运算错误

(1) 没有与 INC(P)/DEC(P) 指令相关的运算错误。

程序示例

(1) 当 X8 变为 ON 时，存储在计数器 C0 到 C20 中的当前值以 BCD 数据的形式输出到 Y30 到 Y3F。（在当前值小于 9999 的时候）

[梯形图模式]

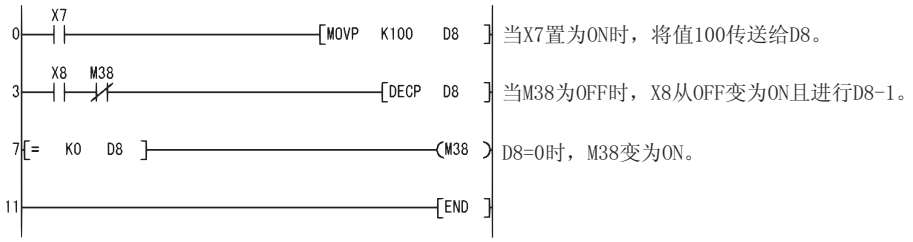


[列表模式]

步	指令	软元件	
0	LD	X8	
1	BCDP	C0Z1	K4Y30
4	INCP	Z1	
6	LD=	K21	Z1
9	OR	X7	
10	RST	Z1	
12	END		

(2) 以下是一个减法计数器程序。

[梯形图模式]

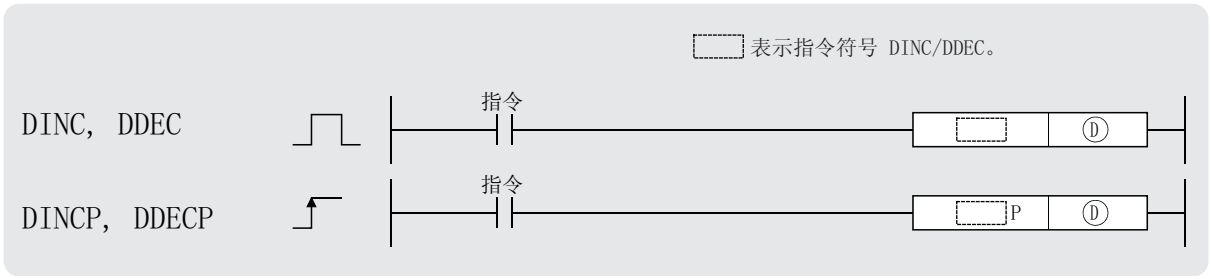


[列表模式]

步	指令	软元件	
0	LD	X7	
1	MOVP	K100	D8
3	LD	X8	
4	ANI	M38	
5	DECP	D8	
7	LD=	K0	D8
10	OUT	M38	
11	END		

6.2.16 32 位 BIN 数据的递增和递减运算 (DINC(P)、DDEC(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



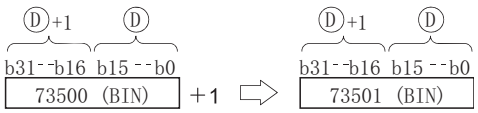
Ⓓ : 执行 DINC(+1)、DDEC(-1) 运算的软元件的起始号 (BIN32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ		○						--	

功 能

DINC

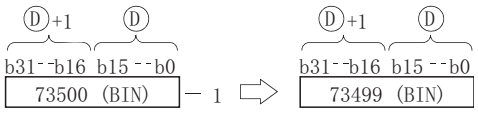
(1) 将 Ⓓ 指定的软元件 (32 位数据) 加 1。



(2) 如果 Ⓓ 指定的软元件的内容是 2147483647，并且在该软元件上执行了 DINC 或 DINCP 指令，那么值 -2147483648 将存储在 Ⓓ 指定的软元件中。

DDEC

(1) 从 Ⓓ 指定的软元件 (32 位数据) 中减 1。



(2) 如果 Ⓓ 指定的软元件的内容是 0，并且在该软元件上执行了 DDEC 或 DDECP 指令，那么值 -1 将存储在 Ⓓ 指定的软元件中。

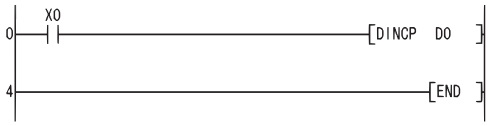
运算错误

(1) 没有与 DINC(P)/DDEC(P) 指令相关的运算错误。

程序示例

(1) 当 X0 变为 ON 时，以下程序将 D0、D1 中的数据加 1。

[梯形图模式]

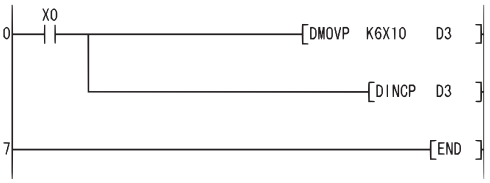


[列表模式]

步	指令	软元件
0	LD	X0
1	DINCP	D0
4	END	

(2) 当 X0 变为 ON 时，以下程序将 X10 到 X27 中设定的数据加 1，并将其结果存储在 D3 和 D4 中。

[梯形图模式]

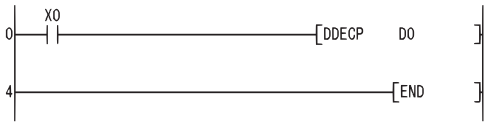


[列表模式]

步	指令	软元件
0	LD	X0
1	DMOVP	K6X10 D3
4	DINCP	D3
7	END	

(3) 当 X0 变为 ON 时，以下程序从 D0、D1 中的数据中减 1。

[梯形图模式]

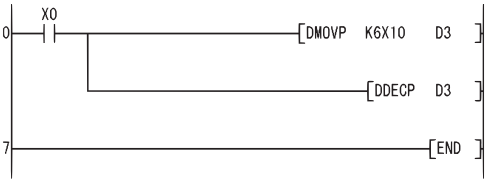


[列表模式]

步	指令	软元件
0	LD	X0
1	DDECP	D0
4	END	

(4) 当 X0 变为 ON 时，以下程序从 X10 到 X27 中的数据中减 1，并将其结果存储在 D3 和 D4 中。

[梯形图模式]



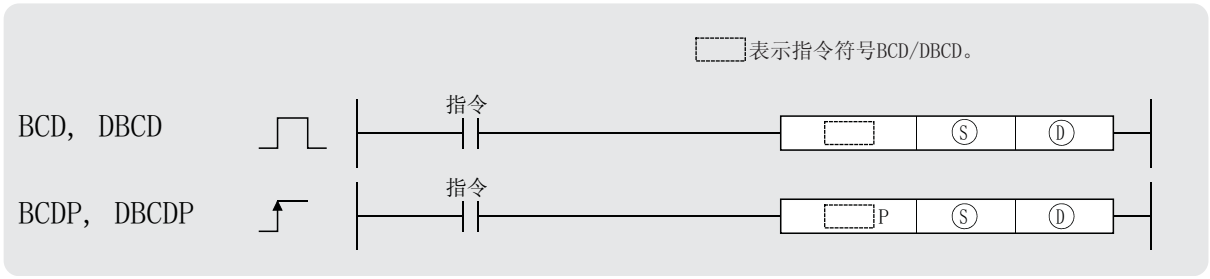
[列表模式]

步	指令	软元件
0	LD	X0
1	DMOVP	K6X10 D3
4	DDECP	D3
7	END	

6.3 数据转换指令

6.3.1 从 BIN 数据到 4 位和 8 位 BCD 数据的转换 (BCD(P)、DBCD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : BIN 数据或者存储 BIN 数据的软元件的起始号。(BIN16/32 位)

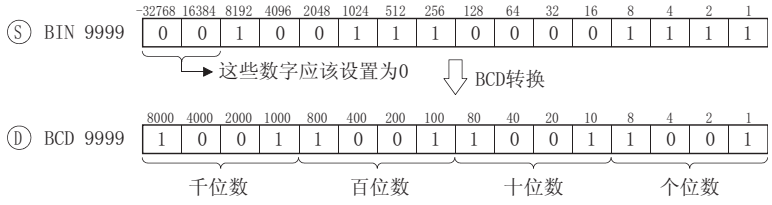
Ⓓ : 存储 BCD 数据的软元件的起始号。(BCD4/8 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○						○	--
Ⓓ		○						--	--

功 能

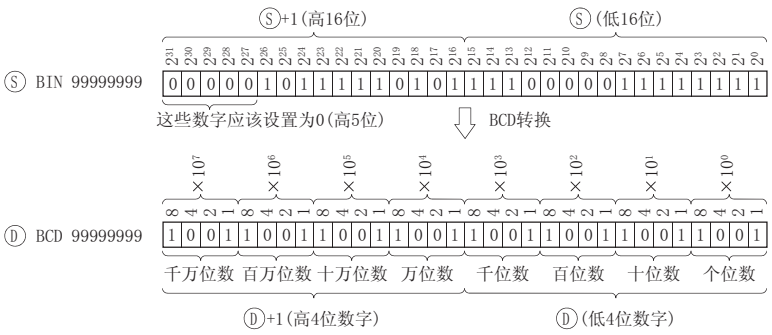
BCD

将 Ⓢ 指定的软元件中的 BIN 数据 (0 到 9999) 转换成 BCD 数据，并将它存储在 Ⓓ 指定的软元件中。



DBCD

将 Ⓢ 指定的软元件中的 BIN 数据 (0 到 99999999) 转换成 BCD 数据，并将它存储在 Ⓓ 指定的软元件中。



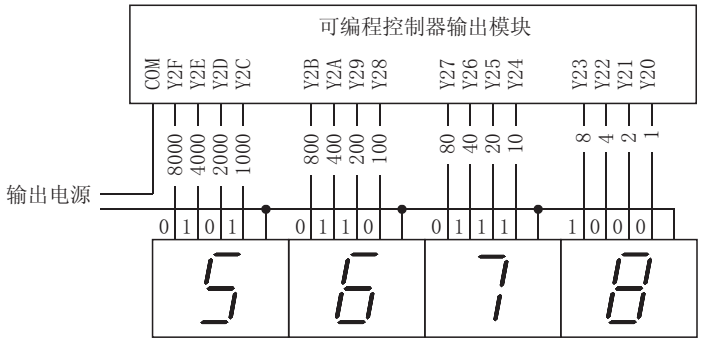
运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 发出 BCD 指令时，Ⓢ 上的数据不在 0 到 9999 范围之内。 （出错代码： 4100）
- 发出 DBCD 指令时， Ⓢ 和 Ⓢ+1 上的数据不在 0 到 99999999 范围之内。
（出错代码： 4100）

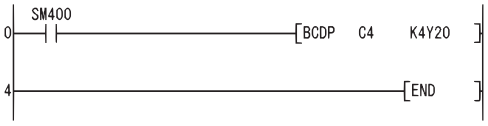
程序示例

(1) 以下程序将位于 Y20 到 Y2F 上的 C4 的当前值输出到 BCD 显示设备。



7-素子显示单元

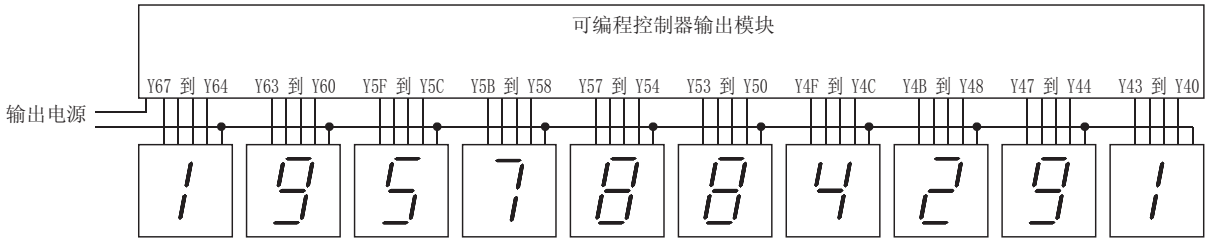
[梯形图模式]



[列表模式]

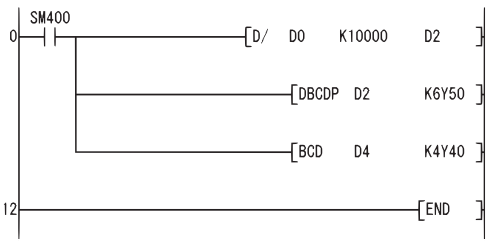
步	指令	软元件	
0	LD	SM400	
1	BCDP	C4	K4Y20
4	END		

(2) 以下程序将 D0 和 D1 上的 32 位数据输出到 Y40 到 Y67。



7-素子显示单元

[梯形图模式]

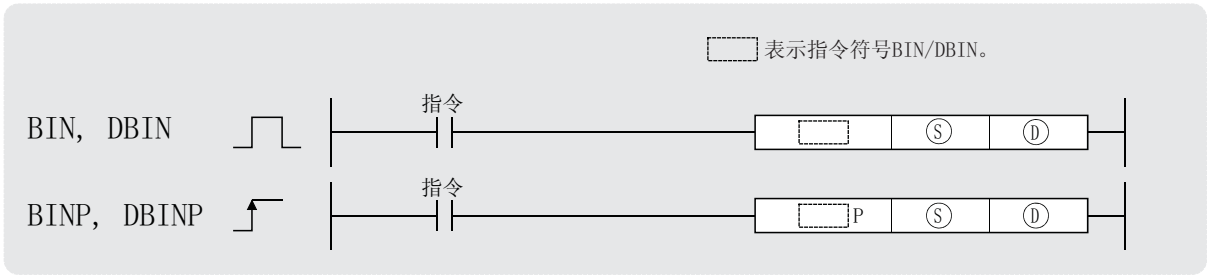


[列表模式]

步	指令	软元件		
0	LD	SM400		
1	D/	D0	K10000	D2
6	DBCDP	D2	K6Y50	
9	BCD	D4	K4Y40	
12	END			

6.3.2 从 BCD 4 位和 8 位数据到 BIN 数据的转换 (BIN(P)、DBIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： BCD 数据或者存储 BCD 数据的软元件的起始号 (BCD4/8 位)。

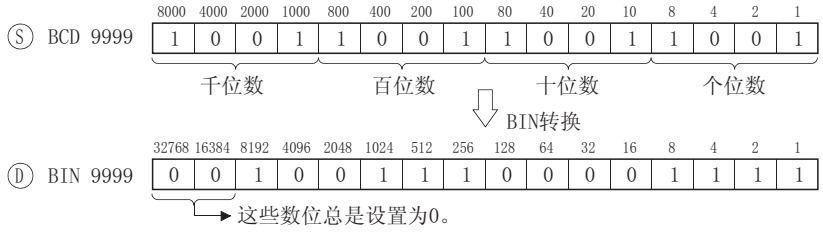
Ⓓ ： 存储 BIN 数据的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ				○				○	--
Ⓓ				○				--	--

功 能

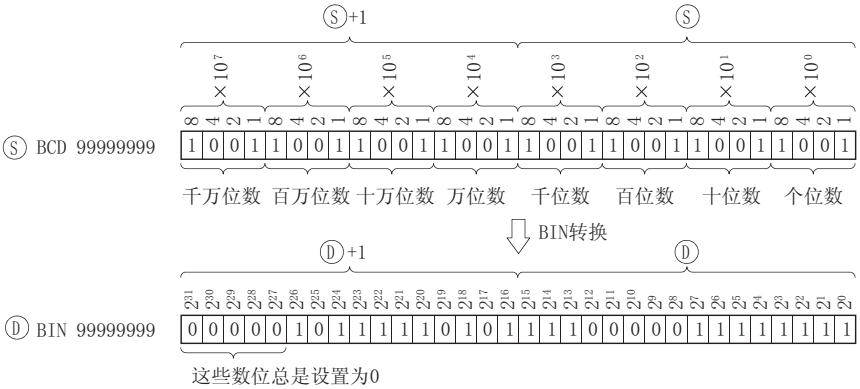
BIN

将 Ⓢ 指定的软元件中的 BCD 数据 (0 到 9999) 转换成 BIN 数据，并将它存储在 Ⓓ 指定的软元件中。



DBIN

将 Ⓢ 指定的软元件中的 BCD 数据 (0 到 99999999) 转换成 BIN 数据，并将它存储在 Ⓓ 指定的软元件中。



运算错误

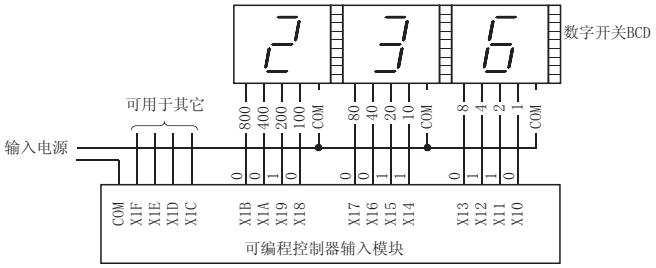
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中，指令也不会执行。

• 当指定给 ⓢ 的任何数字位的值不是 0 到 9 的数字时。 （出错代码： 4100）

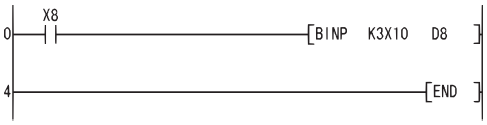
[当使用 QCPU 的时候]
当使用 QCPU 的时候，以上错误可以通过将 SM722 变为 ON 而被抑制掉。
然而，如果所指定的值超出了允许的范围，那么不管将 SM722 变为 ON 还是 OFF，指令都不会执行。
对 BINP/DBINP 指令来说，不管是否存在错误，下一步运算都要等到命令（执行条件）从 OFF 变为 ON 的时候才执行。

程序示例

(1) 当 X8 变为 ON 时，以下程序将 X10 到 X1B 中的 BCD 数据转换成 BIN 数据，并将它存储在 D8 中。



[梯形图模式]

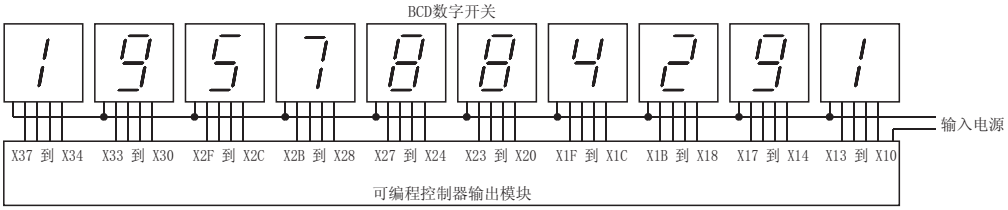


[列表模式]

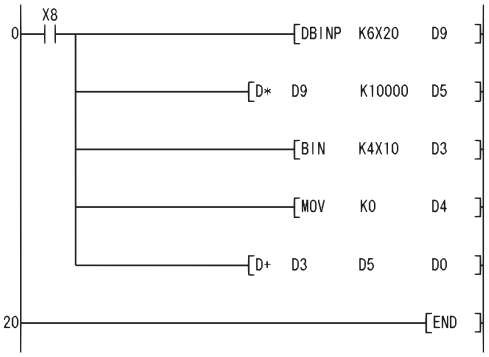
步	指令	软元件	
0	LD	X8	
1	BINP	K3X10	D8
4	END		

(2) 当 X8 变为 ON 时，以下程序将 X10 到 X37 中的 BCD 数据转换成 BIN 数据，并将其存储在 D0 和 D1 中。

（将从 X20 到 X37 中的 BCD 数据转换来的 BIN 数据与从 X10 到 X1F 中的 BCD 数据转换来的 BIN 数据相加。）



[梯形图模式]



[列表模式]

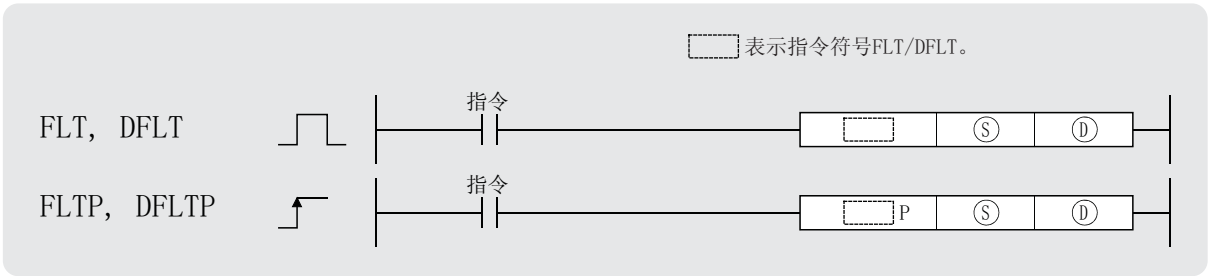
步	指令	软元件		
0	LD	X8		
1	DBINP	K6X20	D9	
4	D*	D9	K10000	D5
9	BIN	K4X10	D3	
12	MOV	K0	D4	
14	D+	D3	D5	D0
20	END			

如果在 X10 到 X37 中设定的 BCD 值大于 2147483647，由于超出了 32 位软元件所能处理的数值的范围，因此 D0 和 D1 中的值将为负。

6.3.3 从 BIN 16 位和 32 位数据到浮点数据的转换 (FLT(P)、DFLT(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



Ⓢ ： 将要转换为 32 位浮点数据的整数数据或存储整数数据的起始软元件号 (BIN16/32 位)。

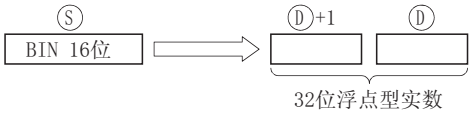
Ⓓ ： 存储已转换的 32 位浮点数据的起始软元件号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○	○		○		--
Ⓓ	--	○		--	○		--		--

功 能

FLT

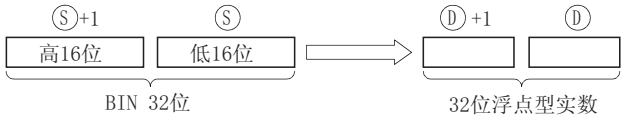
(1) 将 Ⓢ 指定的 16 位 BIN 数据转换成 32 位浮点型实数，并将其结果存储在 Ⓓ 指定的软元件号中。



(2) 可以由 Ⓢ 指定的 BIN 值的范围是 -32768 到 32767。

DFLT

(1) 将 Ⓢ 指定的 32 位 BIN 数据转换成 32 位浮点型实数，并将其结果存储在 Ⓓ 指定的软元件号中。

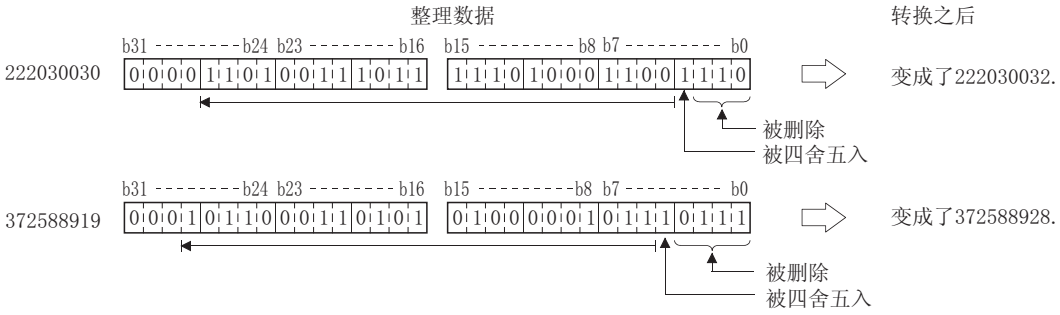


(2) 可以由 Ⓢ+1 和 Ⓢ 指定的 BIN 值的范围是 -2147483648 到 2147483647。

(3) 浮动小数点类型实数的处理是通过简单的 32 位处理来实现的。基于这样的事实，如果以二进制形式显示，那么有效数字的位数是 24 位，如果以十进制形式显示，则有效数字的位数大约是 7 位数字。

因此，如果该整数超出了 -16777216 到 16777215(24 位 BIN 值) 的范围，则可能在转换的结果值中产生错误。

对转换结果从其整型值的最高位开始的第 25 位进行四舍五入，而对其第 26 位及其以后各位中的内容全部予以删除。



运算错误

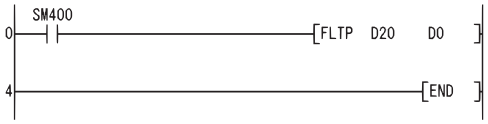
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将 ON，出错代码将被存储到 SD0 中。

- 运算结果超出了以下范围时。(发生了上溢时。)
  (仅对于通用型 QCPU)
  $2^{128} \leqq |$ 运算结果 $|$ (出错代码 : 4141)

程序示例

(1) 以下程序将 D20 上的 BIN16 位数据转换成 32 位浮点型实数，并将其结果存储在 D0 和 D1 中。

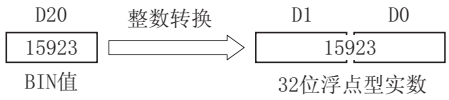
[梯形图模式]



[列表模式]

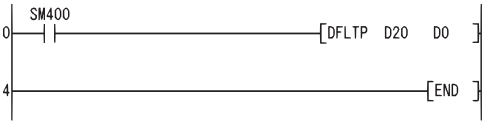
步	指令	软元件	
0	LD	SM400	
1	FLTP	D20	D0
4	END		

[动作]



(2) 以下程序将 D20 和 D21 中的 BIN32 位数据转换成 32 位浮点型实数，并将其结果存储在 D0 和 D1 中。

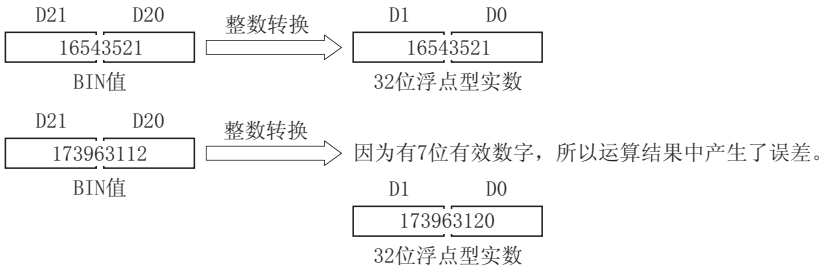
[梯形图模式]



[列表模式]

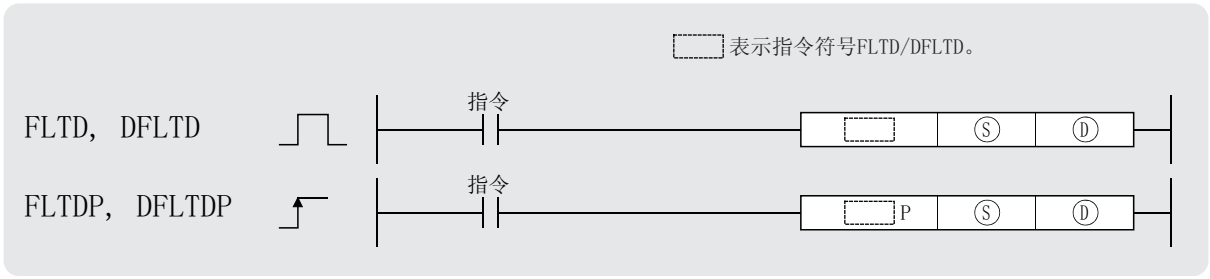
步	指令	软元件	
0	LD	SM400	
1	DFLTP	D20	D0
4	END		

[动作]



6.3.4 从 BIN16 和 32 位数据到浮点数据的转换（双精度）(FLTD(P)、DFLTD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 将要转换为 64 位浮点型实数数据的整数数据或存储整数数据的起始软元件号 (BIN16/32 位)。

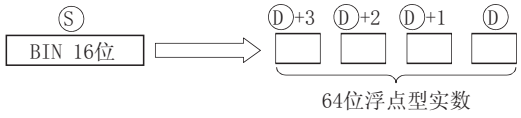
Ⓓ ： 存储已转换的 64 位浮点型实数数据的起始软元件号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	—	○		—			○		—
Ⓓ	—	○		—			—		—

功 能

FLTD

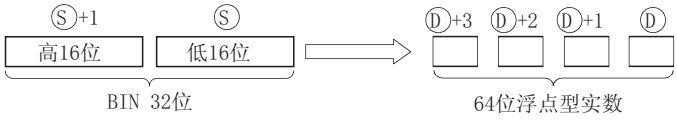
(1) 将 Ⓢ 指定的 16 位 BIN 数据转换成 64 位浮点型实数，并将其结果存储在 Ⓓ 指定的软元件号中。



(2) 可以由 Ⓢ 指定的 BIN 值的范围是 -32768 到 32767。

DFLTD

(1) 将 Ⓢ 指定的 32 位 BIN 数据转换成 64 位浮点型实数，并将其结果存储在 Ⓓ 指定的软元件号中。



(2) 可以由 Ⓢ+1 和 Ⓢ 指定的 BIN 值的范围是 -2147483648 到 2147483647。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0 中。

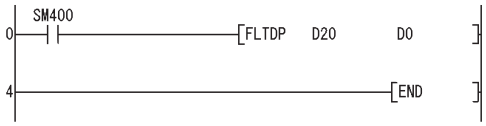
• 运算结果超出以下范围 (上溢的运算结果):

$2^{1024} \leqq$ | 运算结果 |　　(出错代码 : 4141)

程序示例

(1) 以下程序将 D20 中的 BIN16 位数据转换为 64 位浮点型实数，并将其结果存储在 D0 到 D3 中。

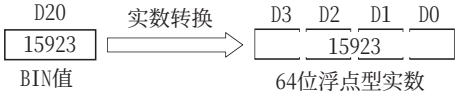
[梯形图模式]



[列表模式]

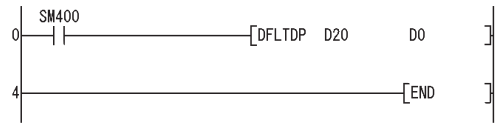
步	指令	软元件	
0	LD	SM400	
1	FLTDP	D20	D0
4	END		

[动作]



(2) 以下程序将 D20 和 D21 中的 32 位 BIN 数据转换成 64 位浮点型实数，并将其结果存储在 D0 到 D3 中。

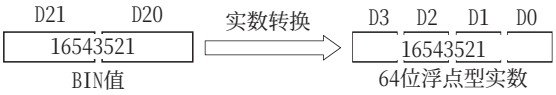
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DFLTDP	D20	D0
4	END		

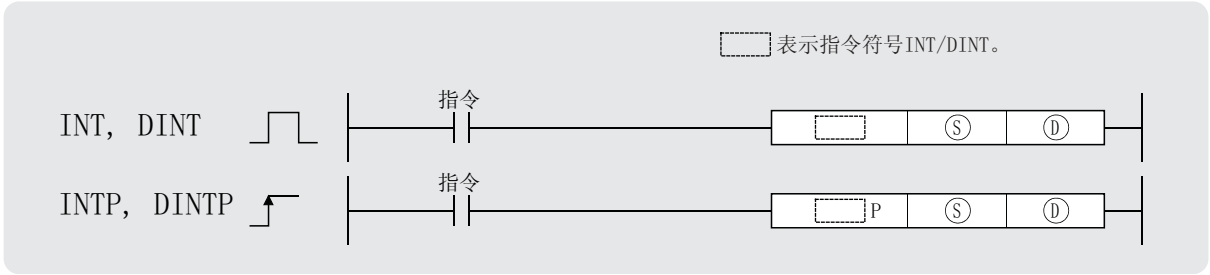
[动作]



6.3.5 从浮点数据到 BIN 16 位和 32 位数据的转换 (INT(P)、DINT(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



Ⓢ ： 将要转换为 BIN 值的 32 位浮点数据或存储浮点数据的起始软元件号（实数）。

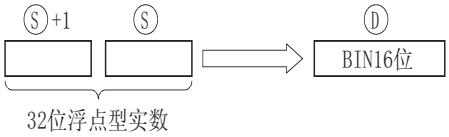
Ⓓ ： 存储转换后的 BIN 值的起始软元件号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	—	○		—	○		—	○	—
Ⓓ	○	○		○	○		○	—	—

功 能

INT

(1) 将 Ⓢ 指定的 32 位浮点数据转换成 BIN16 位数据，并将其结果存储在 Ⓓ 指定的软元件号中。



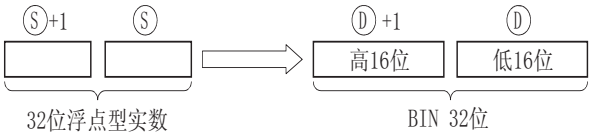
(2) 可以在 Ⓢ+1 或 Ⓢ 指定的 32 位浮点型实数的范围是 -32768 到 32767。

(3) 将存储在 Ⓓ 中的整数值以 BIN16 位值的形式存储。

(4) 转换后的数据为将该实数的小数点后第一位进行四舍五入的值。

DINT

(1) 将 Ⓢ 指定的 32 位浮点型实数转换成 BIN32 位数据，并将其结果存储在 Ⓓ 指定的软元件号中。



(2) 可以指定给 Ⓢ+1 和 Ⓢ 的浮点类型实数的范围是 -2147483648 到 2147483647。

(3) 以 BIN 32 位值的形式将整数值存储在 ⓓ+1 和 ⓓ 中。

(4) 转换后，对该实数小数点后的第一位数字四舍五入。

运算错误

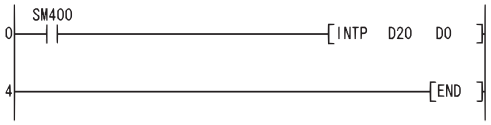
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件的内容或加法结果不为 “0” 且不在以下范围 (只对于通用型 QCPU):
  $0、2^{-126} \leqq |$ 指定的软元件的内容 $| < 2^{128}$ (出错代码： 4140)
- 指定的软元件的值为 -0、非正规数、非数和 $\pm\infty$。
  (只对于通用型 QCPU) (出错代码： 4140)
- 使用 INT 指令时， ⓢ 设置的 32 位浮点型数据的范围超出 -31768 到 32767。
  (出错代码： 4100)
- 使用 DINT 指令时， ⓢ 设置的 32 位浮点型数据的范围超出 -2147483648 到 2147483647。
  (出错代码： 4100)

程序示例

(1) 以下程序将 D20 和 D21 中的 32 位浮点型实数转换成 BIN16 位数据，并将其结果存储在 D0 中。

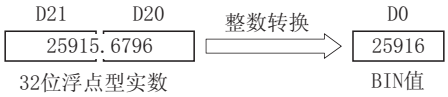
[梯形图模式]

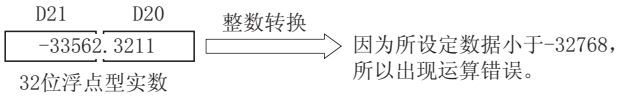


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	INTP	D20	D0
4	END		

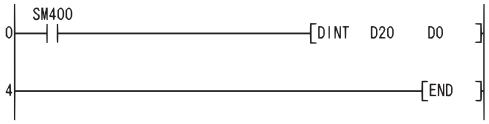
[动作]





(2) 以下程序将 D20 和 D21 中的 32 位浮点型实数转换成 BIN32 位数据，并将其结果存储在 D0 和 D1 中。

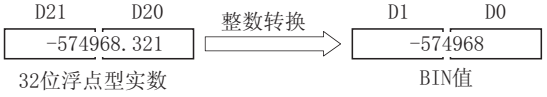
［梯形图模式］

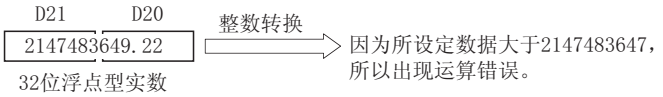


［列表模式］

步	指令	软元件	
0	LD	SM400	
1	DINT	D20	D0
4	END		

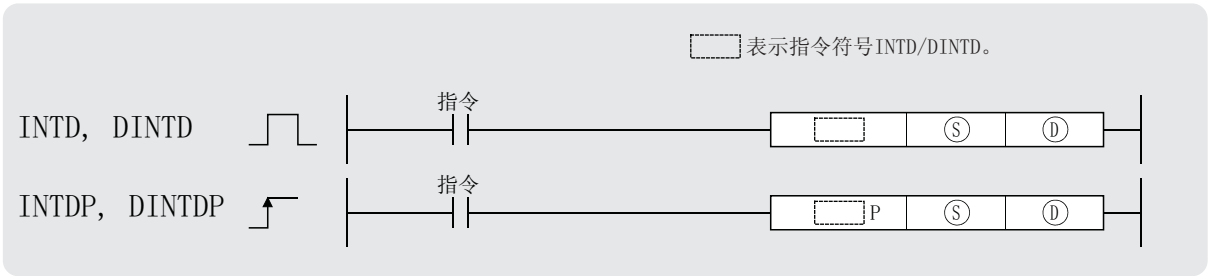
［动作］





6.3.6 从浮点数据到BIN16位/32位数据的转换（双精度）(INTD(P)、DINTD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 将要转换为BIN值的64位浮点型实数数据或存储浮点型实数数据的起始软元件号（实数）。

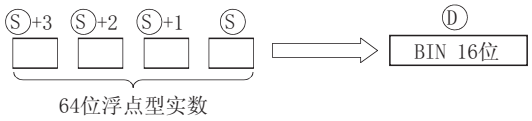
Ⓓ ： 存储转换后的BIN值的起始软元件号（BIN16/32位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--			--	○	--
Ⓓ	--	○		--			○	--	--

功 能

INTD

(1) 将 Ⓢ 指定的64位浮点型实数转换成BIN16位数据，并将其结果存储在 Ⓓ 指定的软元件号中。



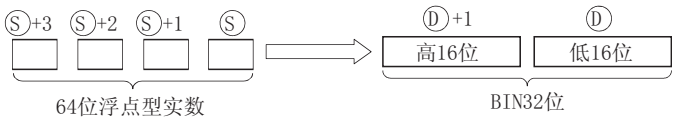
(2) 可以指定给 Ⓢ+3、Ⓢ+2、Ⓢ+1 或 Ⓢ 的64位浮点型实数的范围是-32768到32767。

(3) 以BIN16位值的形式将整数值存储在 Ⓓ 中。

(4) 转换后的数据为对该64位浮点型实数的小数点后的第一位数字进行四舍五入后的值。

DINTD

(1) 将 Ⓢ 指定的64位浮点型实数转换成BIN32位数据，并将其结果存储在 Ⓓ 指定的软元件号中。



(2) 可以指定给 Ⓢ+3、Ⓢ+2、Ⓢ+1 和 Ⓢ 的 64 位浮点类型实数的范围是 -2147483648 到 2147483647。

(3) 以 BIN32 位值的形式将整数值存储到 Ⓓ+1 和 Ⓓ 中。

(4) 转换后的数据为对该 64 位浮点型实数小数点后的第一位数字进行四舍五入后的值。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件的值不在以下范围： （出错代码： 4140）

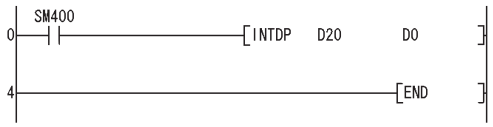
  $0、2^{-1022} \leqq |$ 指定的软元件的值 $| \leqq 2^{1024}$

- 指定的软元件的值为 -0。 （出错代码： 4140）
- 使用 INTD 指令时， Ⓢ 指定的 64 位浮点型数据的范围超出 -31768 到 32767。 （出错代码： 4100）
- 使用 DINTD 指令时， Ⓢ 指定的 64 位浮点型数据的范围超出 -2147483648 到 2147483647。 （出错代码： 4100）

程序示例

(1) 以下程序将 D20 到 D23 中的 64 位浮点型实数转换成 BIN16 位数据，并将其结果存储在 D0 中。

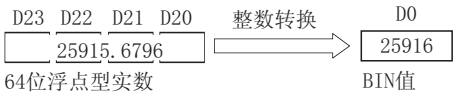
[梯形图模式]

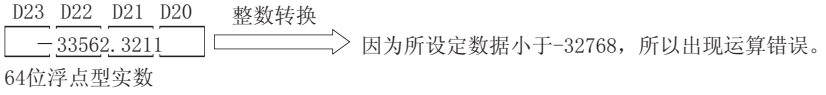


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	INTDP	D20	D0
4	END		

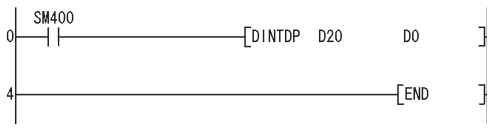
[动作]





(2) 以下程序将 D20 到 D23 中的 64 位浮点型实数转换成 BIN32 位数据，并将其结果存储在 D0 和 D1 中。

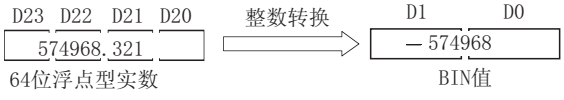
[梯形图模式]

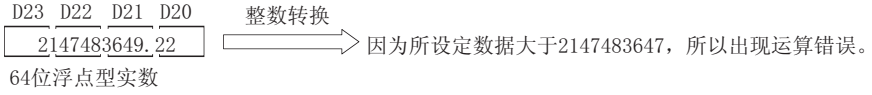


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DINTDP	D20	D0
4	END		

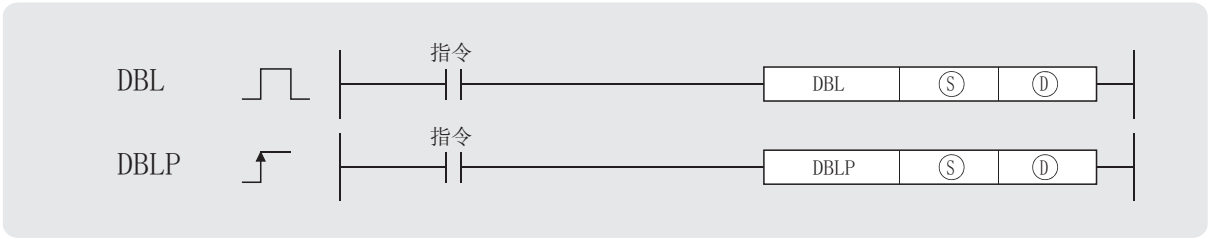
[动作]





6.3.7 从 BIN 16 位数据到 BIN 32 位数据的转换 (DBL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



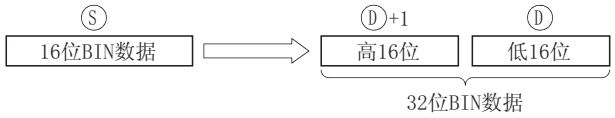
Ⓢ ： BIN16 位数据或存储 BIN16 位数据的软元件的起始号 (BIN 16 位)。

Ⓓ ： 存储转换后的 BIN32 位数据的软元件的起始号 (BIN 32 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓢ		○						○	--
Ⓓ		○						--	--

功 能

将 Ⓢ 指定的软元件中的 BIN 16 位转换为带符号的 BIN 32 位数据，并将其结果存储在 Ⓓ 指定的软元件中。



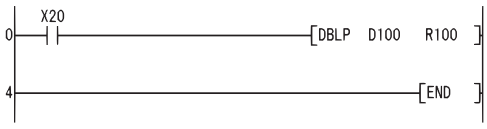
运算错误

(1) 不存在与 DBL(P) 指令相关的运行错误。

程序示例

(1) 当 X20 变为 ON 时，以下程序将存储在 D100 中的 BIN 16 位数据转换成 BIN 32 位数据，并将它存储在 R100 和 R101 中。

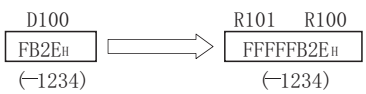
[梯形图模式]



[列表模式]

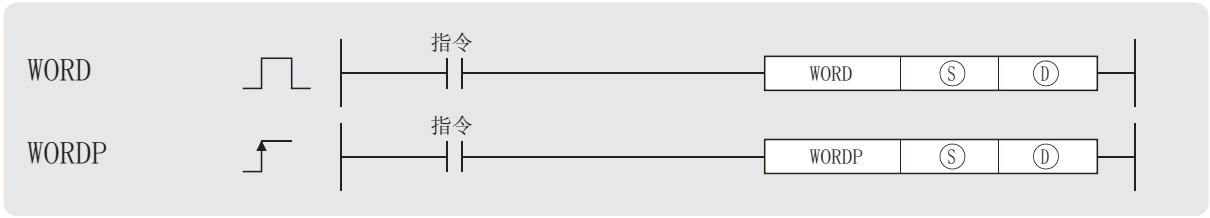
步	指令	软元件	
0	LD	X20	
1	DBLP	D100	R100
4	END		

[动作]



6.3.8 从 BIN 32 位到 BIN 16 位数据的转换 (WORD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : BIN 32 位数据或存储 BIN32 位数据的软元件的起始号。(BIN32 位)

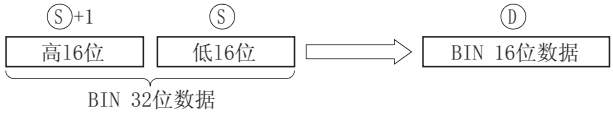
Ⓓ : 存储转换后的 BIN 16 位数据的软元件的起始号。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○						○	--
Ⓓ		○						--	--

功 能

将 Ⓢ 指定的软元件中的 BIN 32 位转换为带符号的 BIN 16 位数据，并将其结果存储在 Ⓓ 指定的软元件中。

可指定给软元件的值的范围是 -32768 到 32767。



运算错误

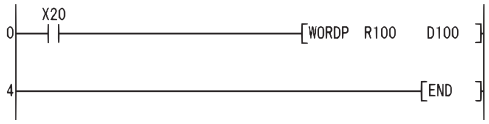
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 指定给 Ⓢ+1 和 Ⓢ 的数据的值不在 -31768 到 32767 范围之内。(出错代码 : 4100)

程序示例

(1) 当 X20 变为 ON 时，以下程序将存储在 R100 和 R101 中的 BIN 32 位数据转换成 BIN 16 位数据，并将它存储在 D100 中。

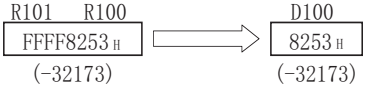
[梯形图模式]



[列表模式]

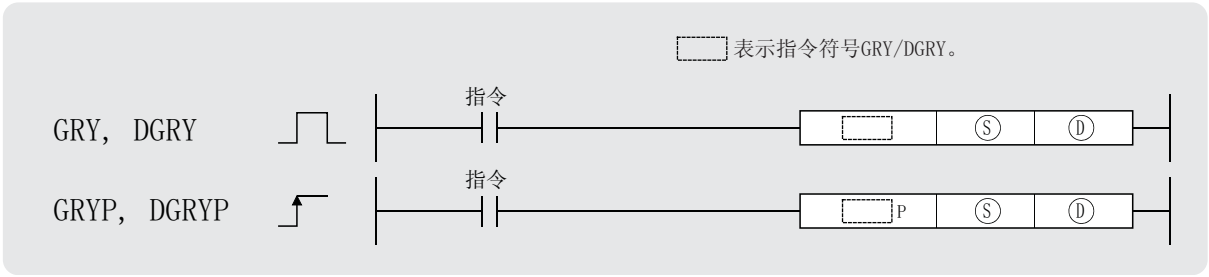
步	指令	软元件	
0	LD	X20	
1	WORDP	R100	D100
4	END		

[动作]



6.3.9 从 BIN 16 位和 32 位数据到格雷码的转换 (GRY(P)、DGRY(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： BIN 数据或存储 BIN 数据的软元件的起始号 (BIN16/32 位)。

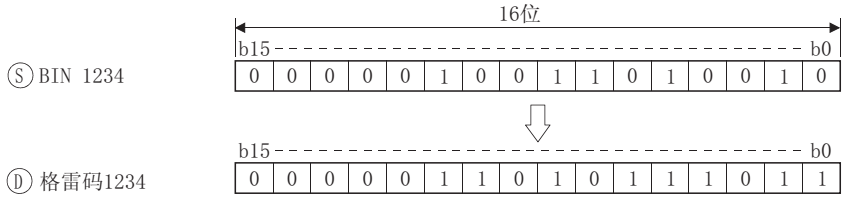
Ⓓ ： 存储转换后的格雷码的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓢ	○							○	--
Ⓓ	○							--	--

功 能

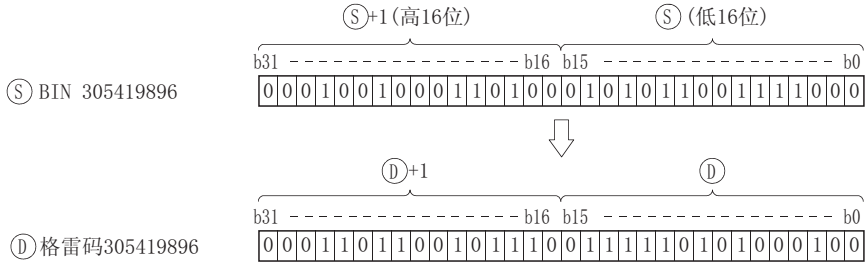
GRY

将 Ⓢ 指定的软元件中的 BIN 16 位数据转换为格雷码，并将其结果存储在 Ⓓ 指定的软元件中。



DGRY

将 Ⓢ 指定的软元件中的 BIN 32 位数据转换为格雷码，并将其结果存储在 Ⓓ 指定的软元件中。



运算错误

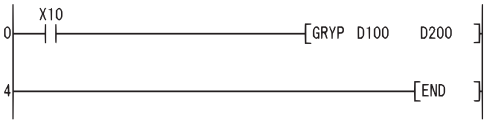
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

•Ⓢ 中的数据是负数。 (出错代码 : 4100)

程序示例

(1) 当 X10 变为 ON 时，以下程序将 D100 中的 BIN 数据转换成格雷码，并将其结果存储在 D200 中。

[梯形图模式]

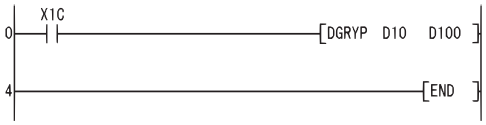


[列表模式]

步	指令	软元件	
0	LD	X10	
1	GRYP	D100	D200
4	END		

(2) 当 X1C 变为 ON 时，以下程序将 D10 和 D11 中的 BIN 数据转换成格雷码，并将它存储在 D100 和 D101 中。

[梯形图模式]

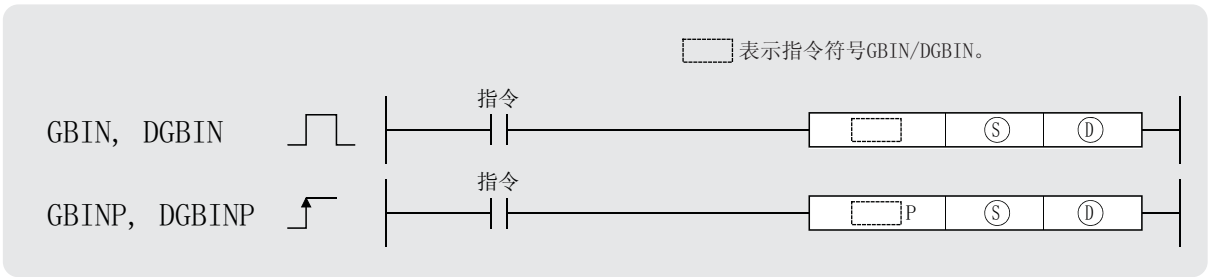


[列表模式]

步	指令	软元件	
0	LD	X1C	
1	DGRYP	D10	D100
4	END		

6.3.10 从格雷码到 BIN 16 位和 32 位数据的转换 (GBIN(P)、DGBIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 格雷码或存储格雷码的软元件的起始号 (BIN16/32 位)。

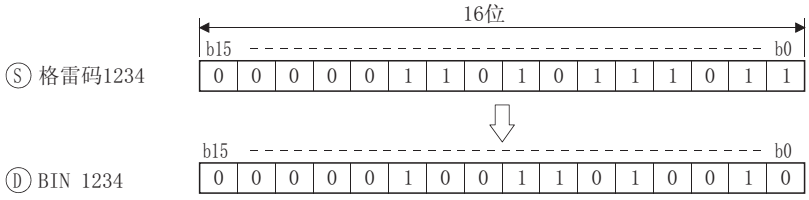
Ⓓ ： 存储转换后的 BIN 数据的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○						○	--
Ⓓ		○						--	--

功 能

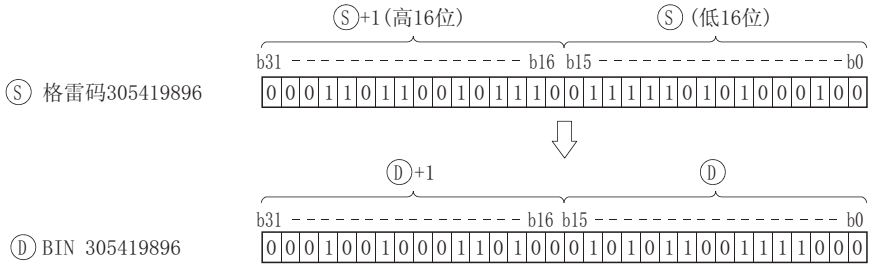
GBIN

将 Ⓢ 指定的软元件中的格雷码数据转换成 BIN 16 位数据，并存储在 Ⓓ 指定的软元件中。



DGBIN

将 Ⓢ 指定的软元件中的格雷码数据转换成 BIN 32 位数据，并存储在 Ⓓ 指定的软元件中。



运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 发出 GBIN 指令时， Ⓢ 中的数据不在 0 到 32767 范围之内。

（出错代码： 4100）

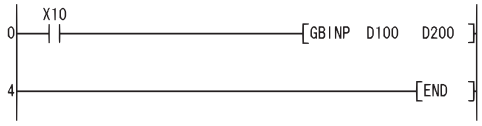
• 发出 DGBIN 指令时， Ⓢ 中的数据不在 0 到 2147483647 范围之内。

（出错代码： 4100）

程序示例

(1) 当 X10 变为 ON 时，以下程序将 D100 中的格雷码数据转换成 BIN 数据，并将其结果存储在 D200 中。

[梯形图模式]

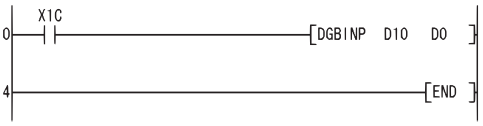


[列表模式]

步	指令	软元件	
0	LD	X10	
1	GBINP	D100	D200
4	END		

(2) 当 X1C 变为 ON 时，以下程序将 D10 和 D11 中的格雷码数据转换成 BIN 数据，并将其结果存储在 D0 和 D1 中。

[梯形图模式]

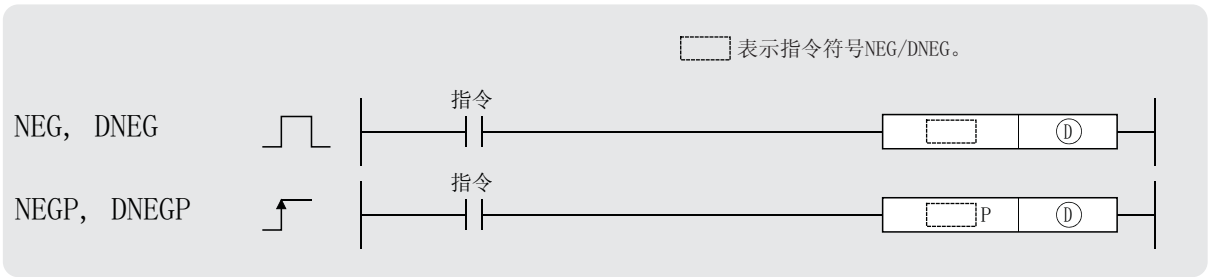


[列表模式]

步	指令	软元件	
0	LD	X1C	
1	DGBINP	D10	D0
4	END		

6.3.11 BIN 16 位和 32 位数据的 2 进制补码（符号取反）(NEG(P)、DNEG(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



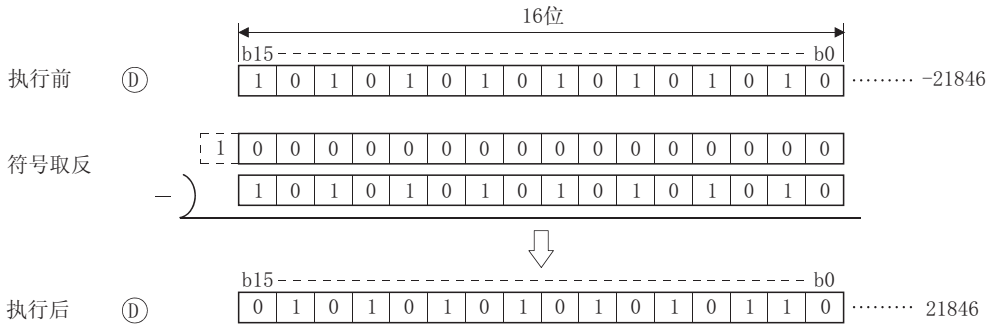
Ⓓ ： 存储用于 2 的补码操作的数据的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	○							--	

功 能

NEG

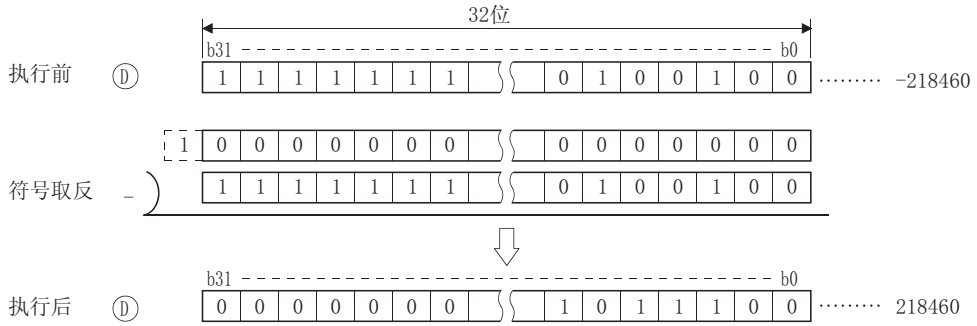
(1) 将 Ⓓ 指定的软元件中的 16 位数据的符号取反，并将其结果存储在 Ⓓ 指定的软元件中。



(2) 用于对正号和负号取反。

DNEG

(1) 将 Ⓓ 指定的软元件中的 32 位数据的符号取反，并将其结果存储在 Ⓓ 指定的软元件中。



(2) 用于对正号和负号取反。

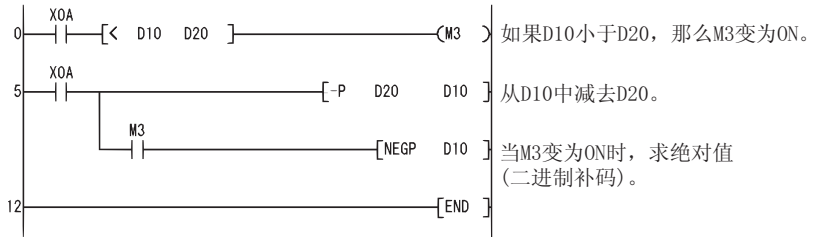
运算错误

(1) 不存在与 NEG(P) 或 DNEG(P) 指令相关的运行错误。

程序示例

(1) 当 XA 变为 ON 时，以下程序计算 D10 到 D20 中的数据的总和，如果其结果为负数，那么求它的绝对值。

[梯形图模式]



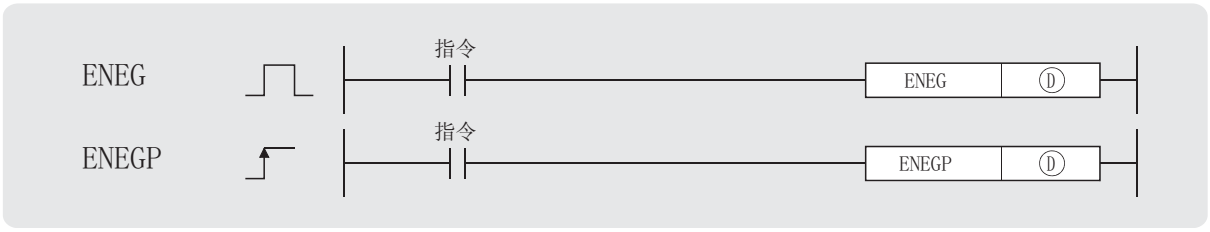
[列表模式]

步	指令	软元件	
0	LD	XOA	
1	AND<	D10	D20
4	OUT	M3	
5	LD	XOA	
6	-P	D20	D10
9	AND	M3	
10	NEGP	D10	
12	END		

6.3.12 浮点数据的符号取反（单精度）(ENEG(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



Ⓓ ： 存储符号取反的 32 位浮点数据的软元件的起始号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	--	○		--	○		--		

功 能

(1) 将 Ⓓ 指定的 32 位浮点型实数数据的符号取反，并将其结果存储在 Ⓓ 指定的软元件中。

(2) 用于对正号和负号取反。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON 相关出错代码将存储在 SD0。

- 指定的软元件的值不在以下范围。（只对于通用型 QCPU）

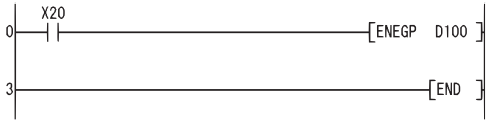
  $0、2^{-126} \leqq |$ 指定的软元件的值 $| < 2^{128}$　（出错代码： 4140）

- 指定的软元件的值为 -0、非正规数、非数、±∞。
  （只对于通用型 QCPU）　（出错代码： 4140）

程序示例

(1) 当 X20 变为 ON 时，以下程序对 D100 到 D101 中的浮动小数点类型实数的符号取反，并将其结果存储在 D100 到 D101 中。

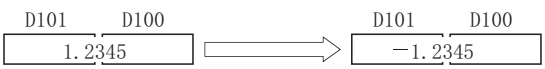
[梯形图模式]



[列表模式]

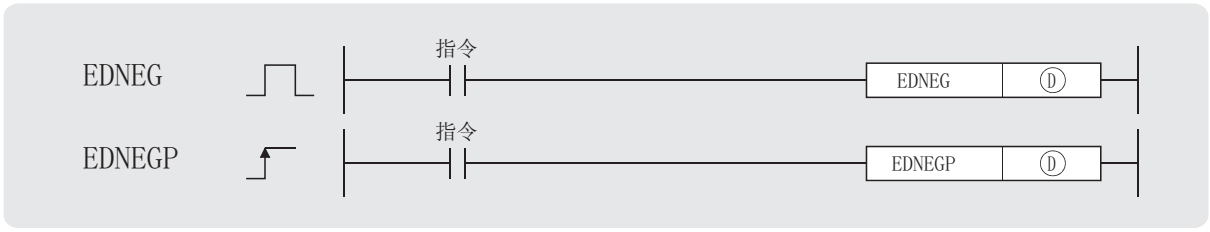
步	指令	软元件
0	LD	X20
1	ENEGP	D100
3	END	

[动作]



6.3.13 浮点数据的符号取反（双精度）(EDNEG(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ ：存储符号取反的 64 位浮点数据的软元件的起始号。（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	—	○		—					

功 能

(1) 将 Ⓓ 指定的软元件的 64 位浮点型实数的符号取反，并将其结果存储在 Ⓓ 指定的软元件中。

(2) 用于对正号和负号取反。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON 相关出错代码将存储在 SD0。

・指定的软元件的值不在以下范围： （出错代码： 4140）

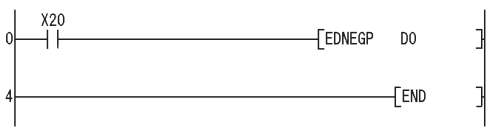
$0、2^{-1022} \leqq |$ 指定的软元件的值 $| < 2^{1024}$

・指定的软元件的值为 -0。 （出错代码： 4140）

程序示例

(1) 当 X20 变为 ON 时，以下程序对 D0 到 D3 中的 64 位浮点型实数的符号取反，并将其结果存储在 D0 到 D3 中。

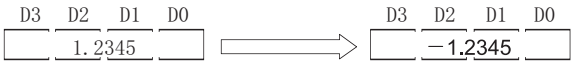
[梯形图模式]



[列表模式]

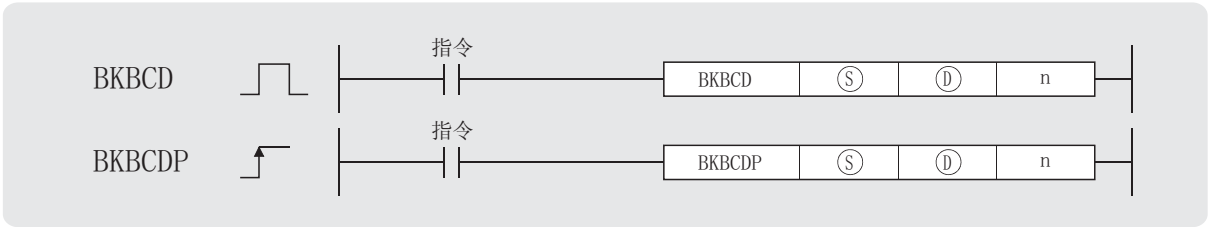
步	指令	软元件
0	LD	X20
1	EDNEGP	D0
3	END	

[动作]



6.3.14 从块 BIN 16 位数据到 BCD 4 位数据的转换 (BKBCD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 存储 BIN 数据的软元件的起始号 (BIN16 位)。

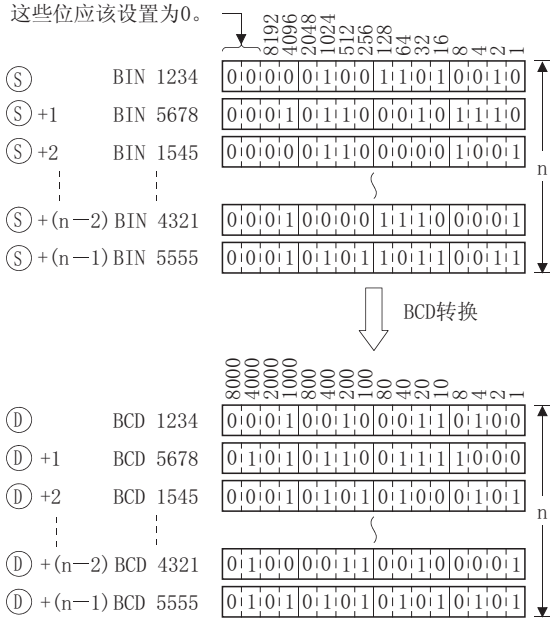
Ⓓ ： 存储转换后的 BCD 数据的软元件的起始号 (BCD4 位)。

n ： 可变数据块的数量 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	○		--					--
Ⓓ	--	○		--					--
n	○	○		--		○			--

功 能

(1) 将从 Ⓢ 指定的软元件开始的 n 点 BIN 数据 (0 到 9999) 转换成 BCD 数据，并将其结果存储在从 Ⓓ 指定的软元件开始的软元件中。



运算错误

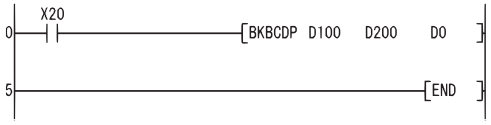
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 Ⓢ 或 Ⓓ 软元件开始的 n 点范围超出了相应软元件的范围。（出错代码： 4101）
- 从 Ⓢ 指定的软元件开始的 n 点数据不在 0 到 9999 范围之内。（出错代码： 4100）
- Ⓢ 软元件与 Ⓓ 软元件相互重叠。（出错代码： 4101）

程序示例

(1) 当 X20 变为 ON 时，以下程序将起始于 D100 的软元件中的若干个（其个数由存储在 D0 中的值决定）BIN 16 位数据转换为 BCD 数据，并将其结果存储在起始于 D200 的软元件中。

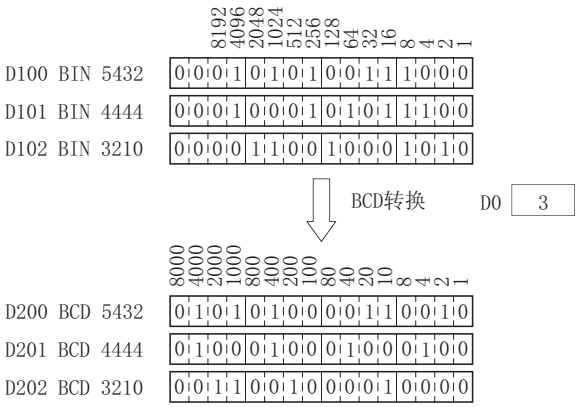
［梯形图模式］



［列表模式］

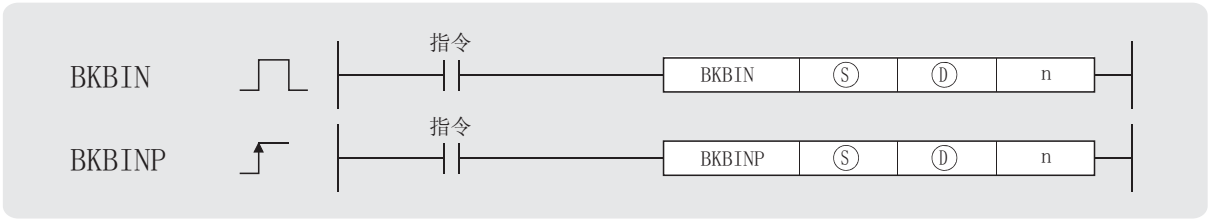
步	指令	软元件		
0	LD	X20		
1	BKBCDP	D100	D200	D0
5	END			

［动作］



6.3.15 从块 BCD 4 位数据到块 BIN 16 位数据的转换 (BKBIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 存储 BCD 数据的软元件的起始号 (BCD4 位)。

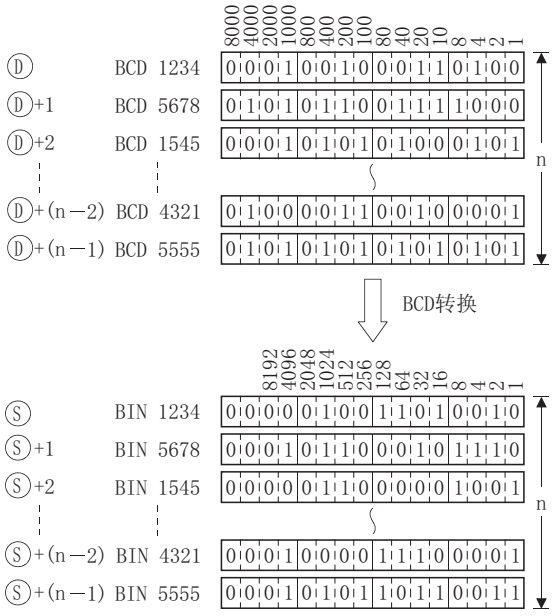
Ⓓ ： 存储转换后的 BIN 数据的软元件的起始号 (BIN16 位)。

n ： 可变数据块的数量 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	○		--					--
Ⓓ	--	○		--					--
n	○	○		○					--

功 能

(1) 将从 Ⓢ 指定的软元件开始的 n 点 BCD 数据 (0 到 9999) 转换成 BIN 数据，并将其结果存储在从 Ⓓ 指定的软元件开始的软元件中。



运算错误

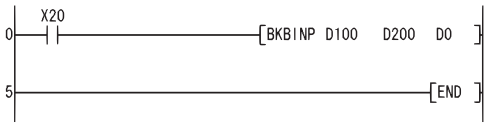
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 或 Ⓓ 软元件开始的 n 点范围超出了该软元件本身的范围。
  （出错代码： 4101）
- 从 Ⓢ 软元件开始的 n 点数据不在 0 到 9999 范围之内。
  （出错代码： 4100）
- 从 Ⓢ 软元件与 Ⓓ 软元件相互重叠。 （出错代码： 4101）

程序示例

(1) 当 X20 变为 ON 时，以下程序将起始于 D100 的软元件中的若干个（其个数由存储在 D0 中的值决定）BCD 数据转换为 BIN 数据，并将其结果存储在起始于 D200 的软元件中。

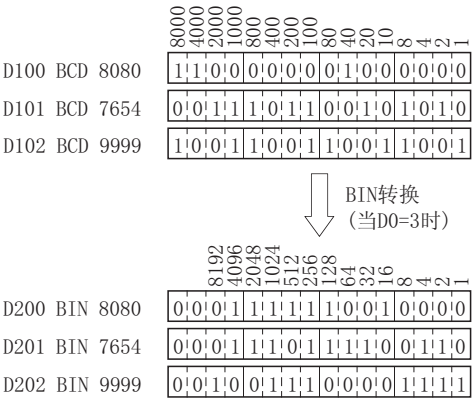
［梯形图模式］



［列表模式］

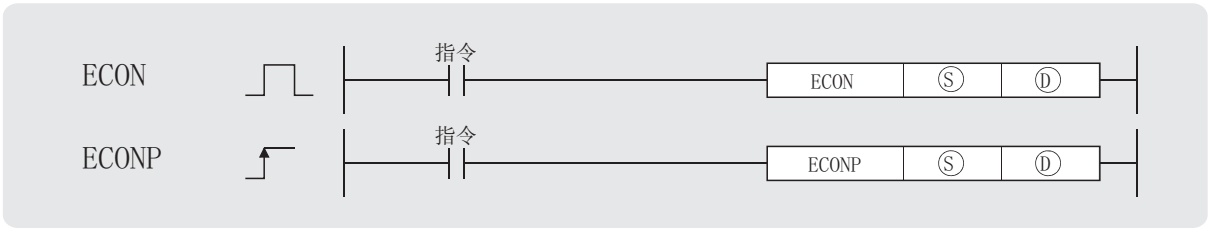
步	指令	软元件		
0	LD	X20		
1	BKBINP	D100	D200	D0
5	END			

［动作］



6.3.16 单精度→双精度转换 (ECON(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



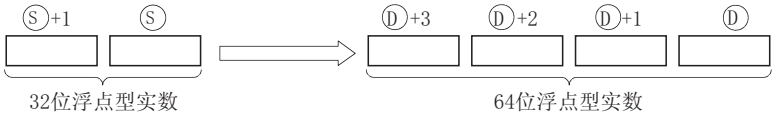
Ⓢ ：转换数据或者存储转换数据的软元件的起始号（实数（单精度））。

Ⓓ ：存储转换后数据的软元件的起始号（实数（双精度））。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	--	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

将 Ⓢ 中指定的 32 位浮点型实数转换为 64 位浮点型实数，将转换结果存储到 Ⓓ 中指定的软元件中。



运算错误

(1) 在下述情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定软元件的内容不在以下范围内。（出错代码： 4140）

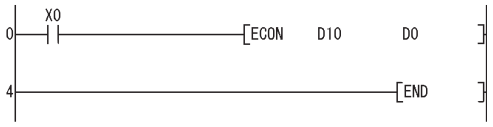
  $0、2^{-126} \leqq |$ 指定软元件的值 $| < 2^{128}$

- 指定软元件的值为 -0、非正规数、非数、$\pm\infty$。（出错代码： 4140）

程序示例

(1) 以下程序在 X0 变为 ON 时，将 D10 ～ D11 的 32 位浮点型实数转换为 64 位浮点型实数后，输出到到 D0）～ D3 中。

[梯形图模式]

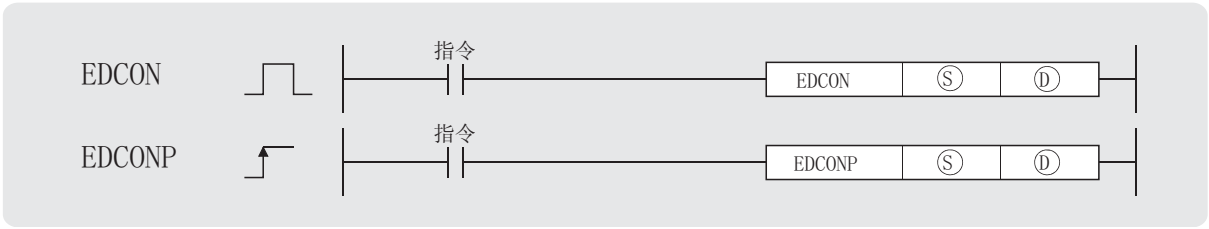


[列表模式]

步	指令	软元件	
0	LD	X0	
1	ECON	D10	D0
4	END		

6.3.17 双精度→单精度转换 (EDCON(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



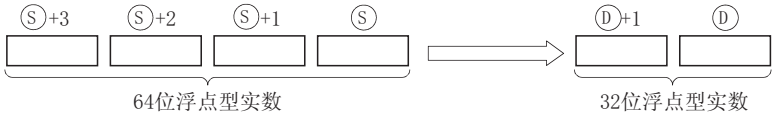
Ⓢ ：转换数据或者存储转换数据的软元件的起始号（实数（双精度））。

Ⓓ ：存储转换后数据的软元件的起始号（实数（单精度））。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--			--	○	--
Ⓓ	--	○		--			○	--	--

功 能

将 Ⓢ 中指定的 64 位浮点型实数转换为 32 位浮点型实数，将转换结果存储到 Ⓓ 中指定的软元件中。



运算错误

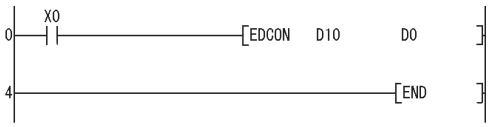
(1) 在下述情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定软元件的内容不在以下范围内：（出错代码：4140）
  $0、2^{-1022} \leqq |$ 指定软元件的值 $| < 2^{1024}$
- 指定软元件的值为 -0。（出错代码：4140）
- 转换结果超出了以下范围。（发生了上溢时。）：
  $2^{128} \leqq |$ 转换结果 $|$（出错代码：4141）

程序示例

(1) 以下程序在 X0 变为 ON 时，将 D10 ～ D13 的 64 位浮点型实数转换为 32 位浮点型实数后，输出到到 D0）～ D1 中。

[梯形图模式]



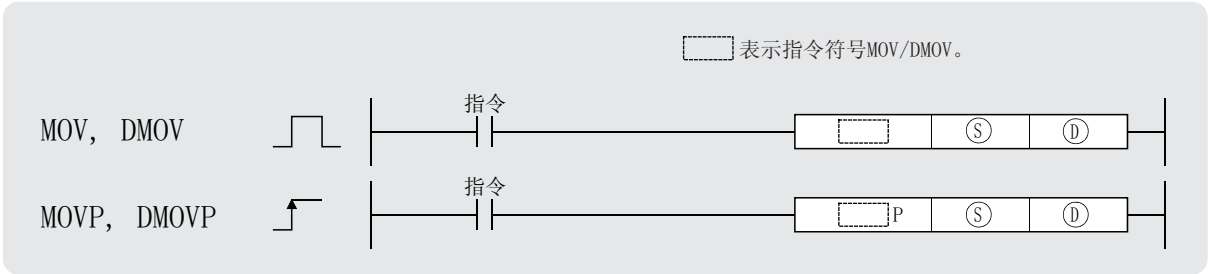
[列表模式]

步	指令	软元件	
0	LD	X0	
1	EDCON	D10	D0
4	END		

6.4 数据传送指令

6.4.1 16 位和 32 位数据传送 (MOV(P)、DMOV(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：传送源数据，或存储传送数据的软元件号 (BIN16/32 位)。

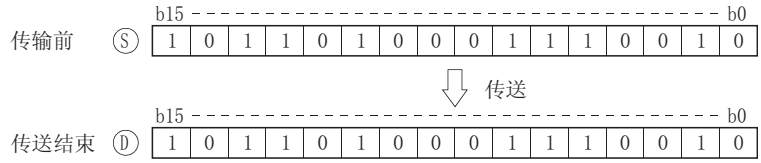
Ⓓ：传送目标软元件号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	--
Ⓓ	○							--	--

功 能

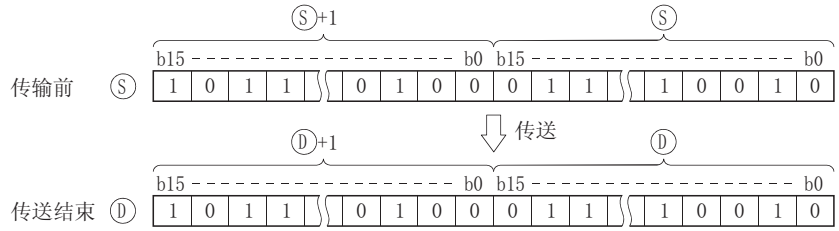
MOV

(1) 将 Ⓢ 指定的软元件中的 16 位数据传送到 Ⓓ 指定的软元件。



DMOV

(2) 将 Ⓢ 指定的软元件中的 32 位数据传送到 Ⓓ 指定的软元件。



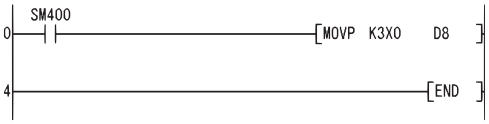
运算错误

(1) 不存在与 MOV(P) 或 DMOV(P) 指令相关的运行错误。

程序示例

(1) 以下程序将来自 X0 到 XB 的输入数据存储在 D8 中。

[梯形图模式]

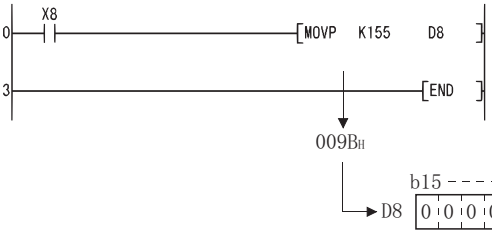


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	MOVP	K3X0	D8
4	END		

(2) 当 X8 变为 ON 时，以下程序将常数 K155 存储在 D8 中。

[梯形图模式]

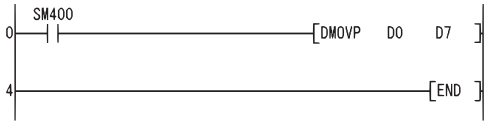


[列表模式]

步	指令	软元件	
0	LD	X8	
1	MOVP	K155	D8
3	END		

(3) 以下程序将来自 D0 和 D1 的数据存储在 D7 和 D8 中。

[梯形图模式]

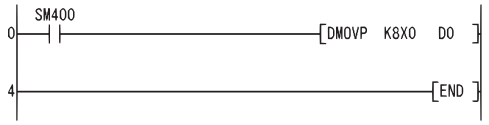


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DMOVP	D0	D7
4	END		

(4) 以下程序将来自 X0 和 X1F 的数据存储在 D0 和 D1 中。

[梯形图模式]



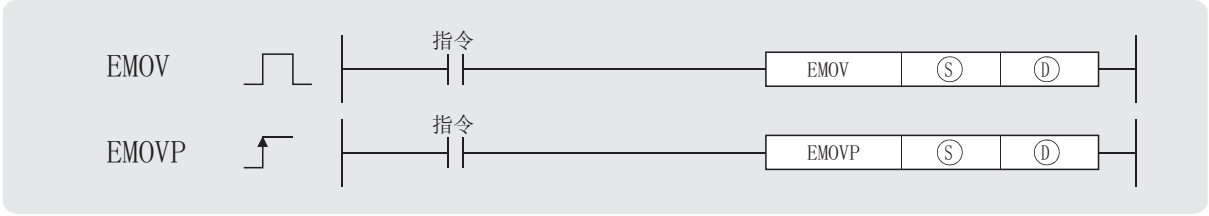
[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DMOVP	K8X0	D0
4	END		

6.4.2 浮点数据传送 (EMOV(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号的前五位数字是 04122 或以后。



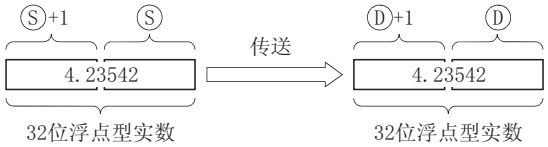
Ⓢ：传送数据，或存储传送数据的软元件号（实数）。

Ⓓ：存储已传送数据的软元件号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--	○		--	○	--
Ⓓ	--	○		--	○		--	--	--

功 能

将 Ⓢ 中指定的软元件中存储的 32 位浮点型实数数据传送至 Ⓓ 中指定的软元件。



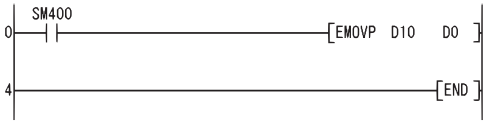
运算错误

(1) 不存在与 EMOV(P) 指令相关的运行错误。

程序示例

(1) 以下程序将 D10 和 D11 中的实数存储到 D0 和 D1 中。

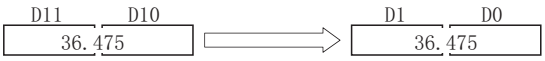
[梯形图模式]



[列表模式]

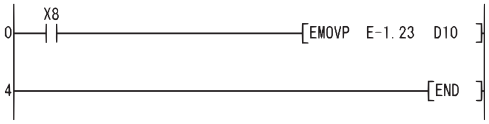
步	指令	软元件	
0	LD	SM400	
1	EMOVP	D10	D0
4	END		

[动作]



(2) 当 X8 变为 ON 时，以下程序将实数 -1.23 存储到 D10 和 D11 中。

[梯形图模式]



[列表模式]

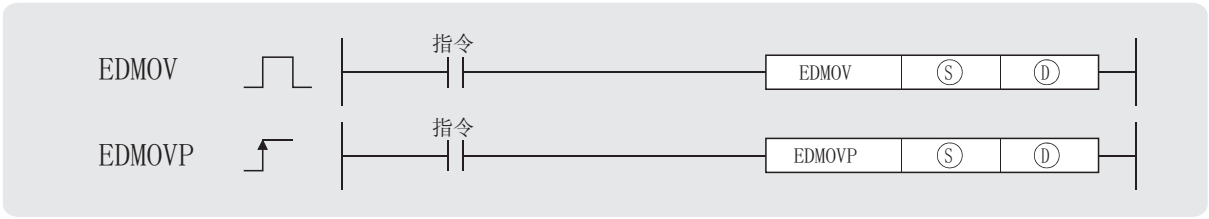
步	指令	软元件	
0	LD	X8	
1	EMOVP	E-1.23	D10
4	END		

[动作]



6.4.3 浮点数据传送（双精度）(EDMOV(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



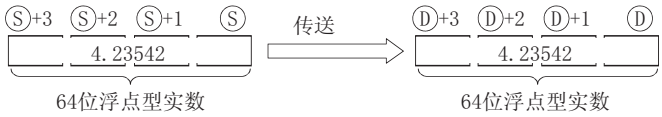
Ⓢ ：传送数据，或存储传送数据的软元件号（实数）。

Ⓓ ：存储传送数据的软元件号（实数）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

将 Ⓢ 指定的软元件中存储的 64 位浮点型实数传送到 Ⓓ 指定的软元件中。



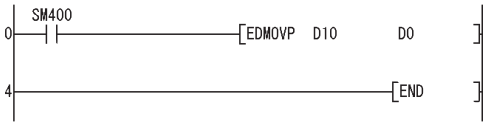
运算错误

(1) 不存在与 EDMOV(P) 指令相关的运算错误。

程序示例

(1) 以下程序将 D10 到 D13 中 64 位浮点型实数存储在 D0 到 D3 中。

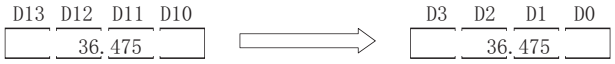
[梯形图模式]



[列表模式]

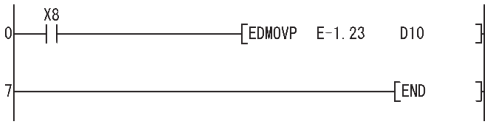
步	指令	软元件	
0	LD	SM400	
1	EDMOVP	D10	D0
4	END		

[动作]



(2) 当 X8 变为 ON 时，以下程序将实数 -1.23 存储在 D10 到 D13 中。

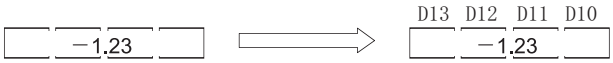
[梯形图模式]



[列表模式]

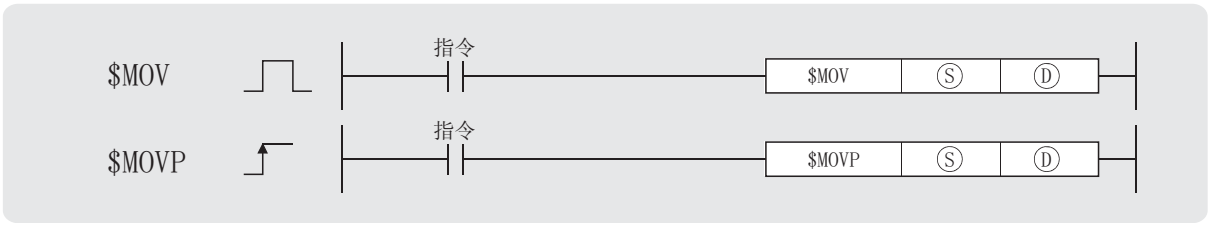
步	指令	软元件	
0	LD	X8	
1	EDMOVP	E-1.23	D10
7	END		

[动作]



6.4.4 字符串传送（$MOV(P)）

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：将传送的字符串（字符串中的最多字符数：对于 QnA/Q4AR，16 个字符；对于 QCPU，32 个字符）或者存储传送的字符串的软元件的起始号（字符串）。

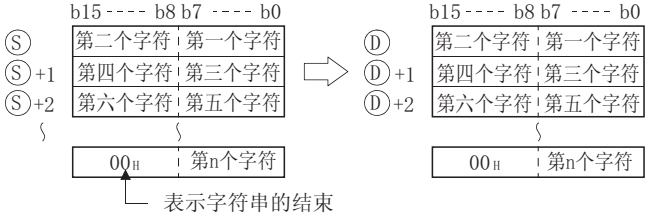
Ⓓ：存储传送字符串的软元件的起始号（字符串）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

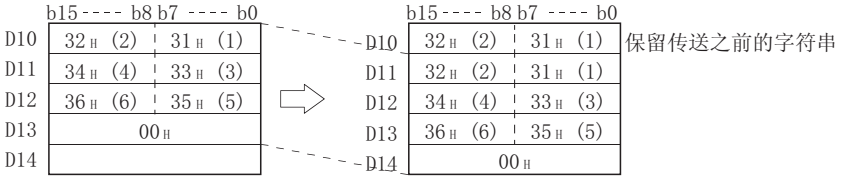
(1) 将从 Ⓢ 指定的软元件号开始的软元件中存储的字符串数据传送到从 Ⓓ 指定的软元件号开始的软元件中。

字符串传送包括对从 Ⓢ 指定的软元件号起始到存储 “00H”的软元件号为止的软元件号中存储的所有数据的传送。

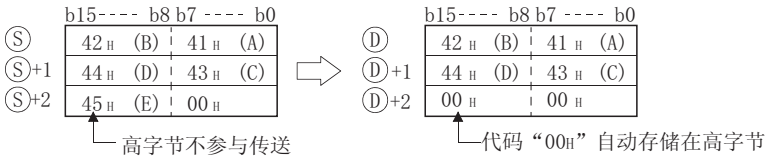


(2) 即使遇到存储将要传送的字符数据的软元件范围（Ⓢ 到 Ⓢ+n）与将要存储已完成传输的字符串数据的软元件范围（Ⓓ 到 Ⓓ+n）相互重叠的情况，执行处理时也不会发生错误。

当把存储在 D10 到 D13 中的字符串数据传送到 D11 到 D14 中时，会出现以下情况：



(3) 如果代码 “00H” 存储在 ⓢ+n 的低字节，那么它会同时存储在 ⓓ+n 的高字节和低字节。



运算错误

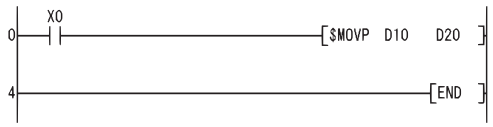
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 在 ⓢ 指定的软元件号与相应软元件号之间不存在代码 “00H”。(出错代码 : 4101)
- 不能完整存储从 ⓓ 指定的软元件号以后到相应软元件的最终软元件号之间连接的整个字符串时。 (出错代码 : 4101)
- ⓢ 的字符串超出 16383 字符。 (出错代码 : 4101)

程序示例

(1) 以下程序在 X0 变为 ON 时，将存储在 D10 到 D12 中的字符串数据传送到 D20 到 D22 中。

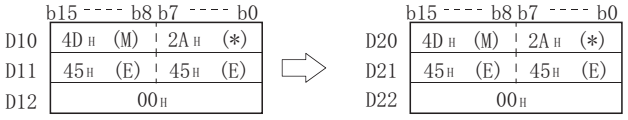
[梯形图模式]



[列表模式]

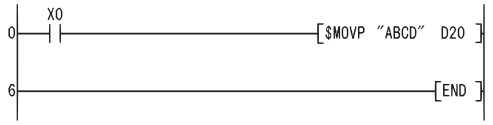
步	指令	软元件	
0	LD	X0	
1	$MOVP	D10	D20
4	END		

[动作]



(2) 以下程序在 X 变为 ON 时，将字符串 “ABCD” 传送到 D20 和 D21 中。

[梯形图模式]

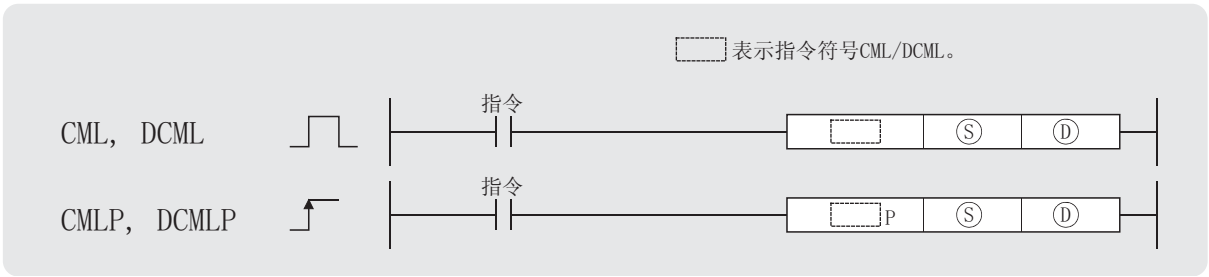


[列表模式]

步	指令	软元件	
0	LD	X0	
1	$MOVP	"ABCD"	D20
6	END		

6.4.5 16 位和 32 位数据取反传送 (CML(P)、DCML(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要取反的数据或者存储该数据的软元件号 (BIN16/32 位)。

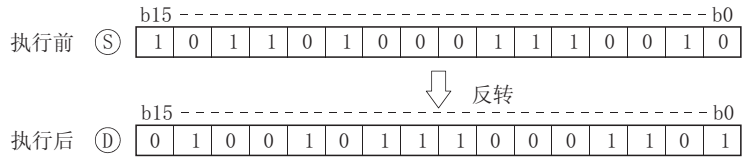
Ⓓ : 存储取反结果的软元件号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	--
Ⓓ	○							--	--

功 能

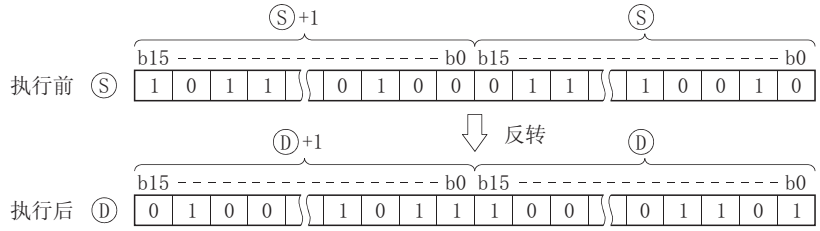
CML

(1) 将 Ⓢ 指定的 16 位数据逐位取反，并将其结果传送到 Ⓓ 指定的软元件中。



DCML

(1) 将 Ⓢ 指定的 32 位数据逐位取反，并将其结果传送到 Ⓓ 指定的软元件中。



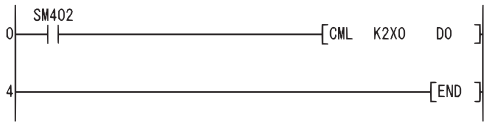
运算错误

(1) 不存在与 CML(P) 或 DCML(P) 指令相关的运行错误。

程序示例

(1) 以下程序将 X0 到 X7 中的数据取反，并将其结果传送到 D0 中。

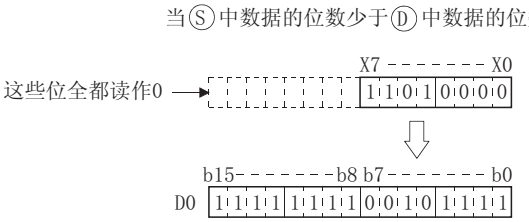
[梯形图模式]



[列表模式]

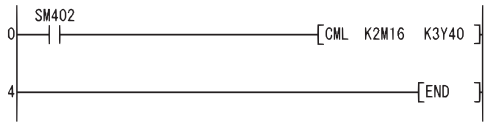
步	指令	软元件	
0	LD	SM402	
1	CML	K2X0	D0
4	END		

[动作]



(2) 以下程序将 M16 到 M23 中的数据取反，并将其结果传送到 Y40 到 Y47 中。

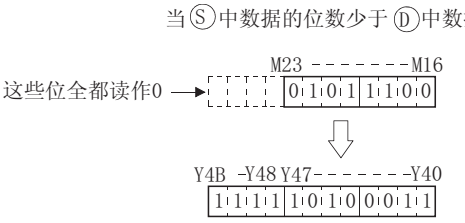
[梯形图模式]



[列表模式]

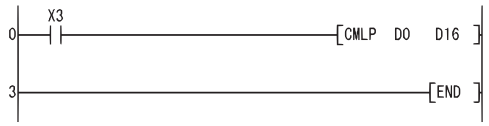
步	指令	软元件	
0	LD	SM402	
1	CML	K2M16	K3Y40
4	END		

[动作]



(3) 当 X3 变为 ON 时，以下程序将 D0 中的数据取反，并将其结果传送到 D16 中。

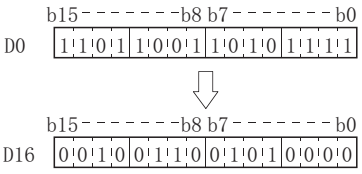
[梯形图模式]



[列表模式]

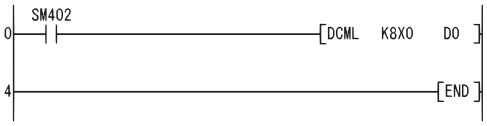
步	指令	软元件	
0	LD	X3	
1	CMLP	D0	D16
3	END		

[动作]



(4) 以下程序将 X0 到 X1F 中的数据取反，并将其结果传送到 D0 和 D1 中。

[梯形图模式]

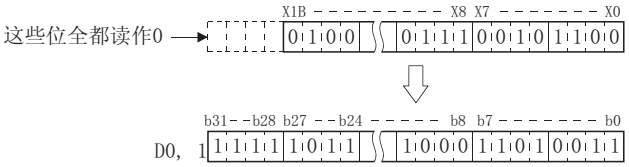


[列表模式]

步	指令	软元件	
0	LD	SM402	
1	DCML	K8X0	D0
4	END		

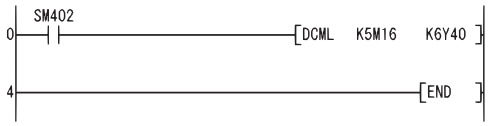
[动作]

当Ⓢ中数据的位数少于Ⓓ中数据的位数时



(5) 以下程序将 M16 到 M35 中的数据取反，并将它传送到 Y40 到 Y63 中。

[梯形图模式]

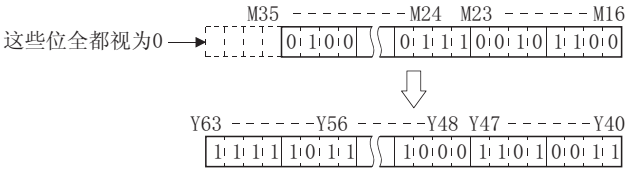


[列表模式]

步	指令	软元件	
0	LD	SM402	
1	DCML	K5M16	K6Y40
4	END		

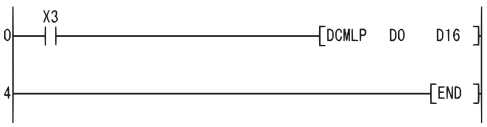
[动作]

当Ⓢ中数据的位数少于Ⓓ中数据的位数时



(6) 当 X3 变为 ON 时，以下程序将 D0 到 D1 中的数据取反，并将其结果存储在 D16 和 D17 中。

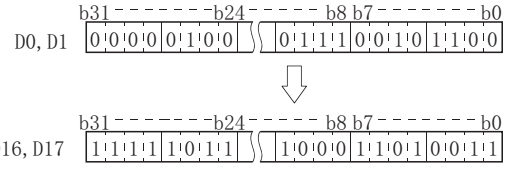
[梯形图模式]



[列表模式]

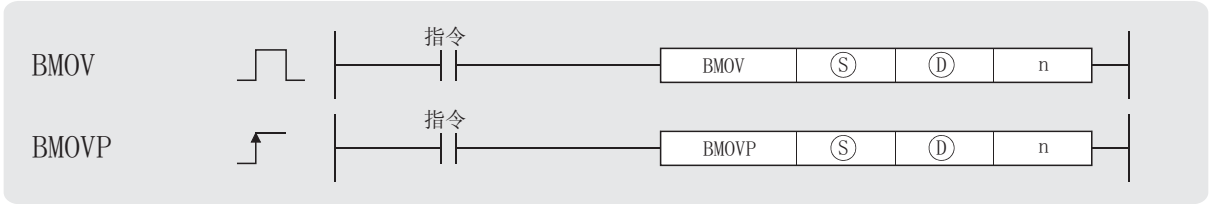
步	指令	软元件	
0	LD	X3	
1	DCMLP	D0	D16
4	END		

[动作]



6.4.6 块 16 位数据传送 (BMOV(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 存储将要传送的数据的软元件的起始号 (BIN16 位)。

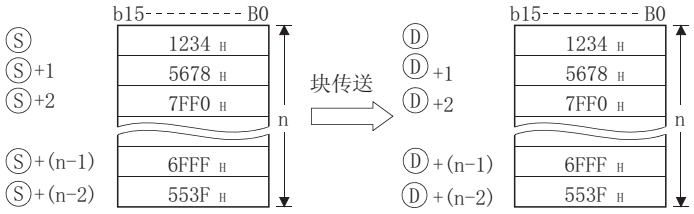
Ⓓ : 传送目标软元件的起始号 (BIN16 位)。

n : 传输的数量 (如果使用特殊功能模块软元件 (U □ \G □)，则传输数量为 1 ～ 6144(只对于 QnA))。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ			○				--		--
Ⓓ			○				--		--
n			○				○		--

功 能

(1) 将从 Ⓢ 指定的软元件开始的软元件中的 n 点 16 位数据批量传送到从 Ⓓ 指定的软元件开始的 n 点位置。



(2) 即使传送源软元件与传送目标软元件之间存在相互重叠的情况，传送也照样能够完成。
如果传送到较小的软元件号，则从 Ⓢ 开始传送；如果传送到较大的软元件号，则从 Ⓢ +(n－1) 开始传送。
但是，如下所示，当从 R 向 ZR 传送数据或从 ZR 向 R 传送数据时，传送范围 (源) 和目标范围一定不能相互重叠。
从 R 向 R 传送数据或从 ZR 向 ZR 传送数据时，不会出现问题。

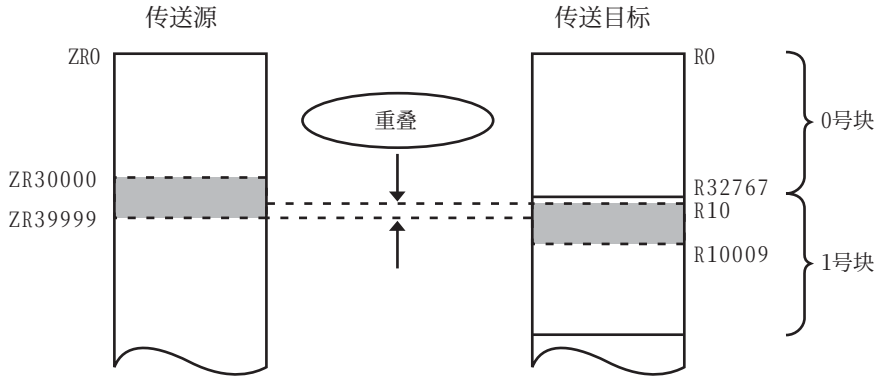
- ZR 的传送范围 ((指定的 ZR 起始号) ～ (指定的 ZR 起始号 + 传送数量 -1))
- R 的传送范围 ((指定的 R 起始号 + 文件寄存器块号× 32768) ～ (指定的 R 起始号 + 文件寄存器块号× 32768+ 传送数量 -1))

例如

当从 ZR30000（传送源）传送 10000 块数据到 R10（传送目标 1 号块），ZR 和 R 的传送范围相互重叠。

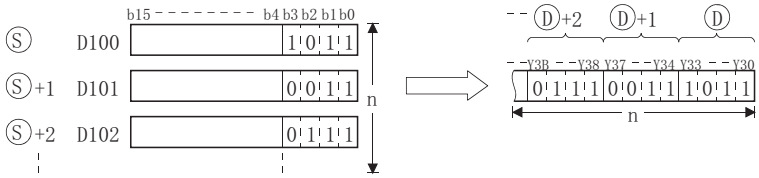
- ZR 的传送范围→（30000）～（30000+10000-1）→（30000）～（39999）
- R 的传送范围→（10+(1×32768)）～（10+(1×32768)+10000-1）
  →（32778）～（42777）

因此，32778 到 39999 的范围相互重叠，数据无法正常传送。



(3) 当 Ⓢ 是字软元件而 Ⓓ 是位软元件时，字软元件的对象将是由位软元件的数字标识所指定的位数。

如果 Ⓓ 指定了 K1Y30，那么 Ⓢ 指定的字软元件的低四位将成为该对象。



(4) 如果给 Ⓢ 和 Ⓓ 指定的都是位软元件，那么 Ⓢ 和 Ⓓ 应该总是具有相同的数字位数。

(5) 当 Ⓢ 和 Ⓓ 使用链接直接软元件和智能功能模块软元件时，只能使用 Ⓢ 和 Ⓓ 中的一个。

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

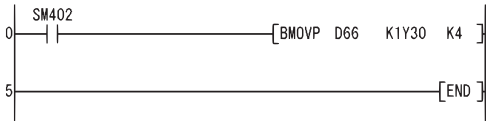
• 从 Ⓢ 或 Ⓓ 起始的 n 点软元件的范围超出了相关软元件的范围。 (出错代码： 4101)

• 使用特殊直接软元件时，传输数量超过了 6144。(QnACPU) (出错代码： 4101)

程序示例

(1) 以下程序将 D66 到 D69 中数据的低 4 位输出到 4 点单元 Y30 到 Y3F。

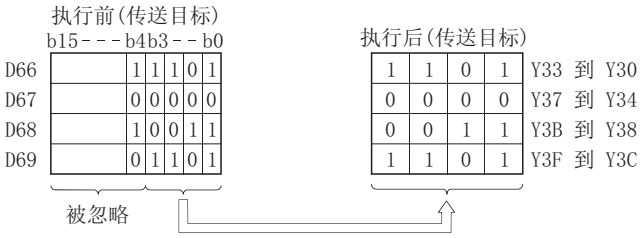
[梯形图模式]



[列表模式]

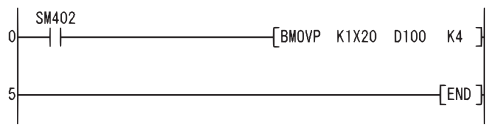
步	指令	软元件		
0	LD	SM402		
1	BMOVP	D66	K1Y30	K4
5	END			

[动作]



(2) 以下程序将 X20 到 X2F 中的数据输出到 4 点单元 D100 到 D103。

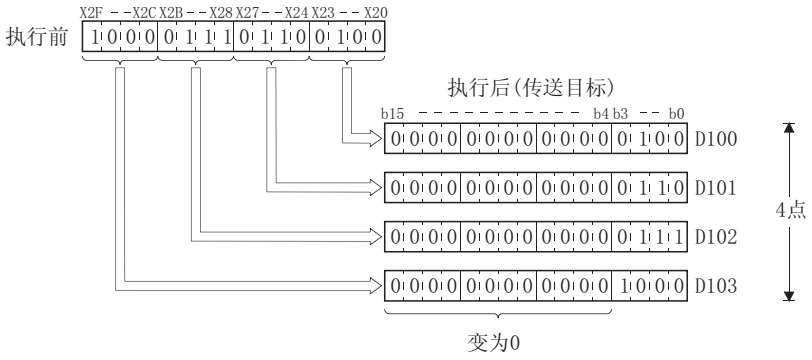
[梯形图模式]



[列表模式]

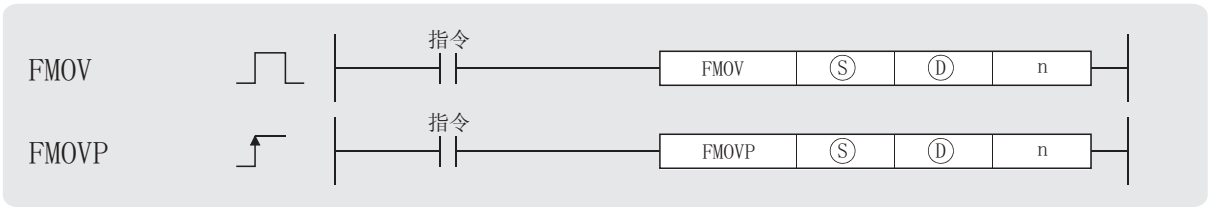
步	指令	软元件		
0	LD	SM402		
1	BMOVP	K1X20	D100	K4
5	END			

[动作]



6.4.7 相同的 16 位数据块传送 (FMOV(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 要传送的数据或存储要传送的数据的软元件的起始号 (BIN16 位)。

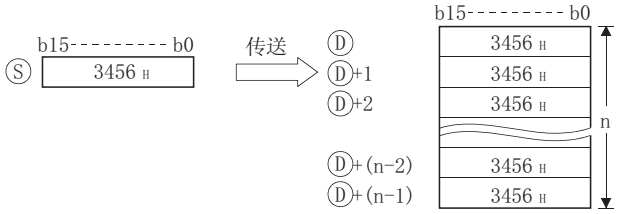
Ⓓ : 传送目标的软元件的起始号 (BIN16 位)。

n : 传输的数量 (如果使用特殊功能模块 (U □ \G □)，则传输数量为 1 ～ 6144(只对于 QnA))。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ			○				○		--
Ⓓ			○				--		--
n			○				○		--

功 能

(1) 将 Ⓢ 指定的软元件中的 16 位数据传送到从 Ⓓ 指定的软元件开始的 n 点位置。



(2) 如果 Ⓢ 指定一个字软元件而 Ⓓ 指定一个位软元件，那么由位软元件的数字标识指定的位数就是字软元件的对象位。

如果 Ⓓ 指定了 K1Y30，那么 Ⓢ 指定的字软元件的对象位将是低 4 位。



(3) 如果给 Ⓢ 和 Ⓓ 指定的都是位软元件，那么 Ⓢ 和 Ⓓ 应该总是具有相同的数字位数。

运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 从 ⓓ 起始的 n 点软元件的范围超出了该软元件范围。

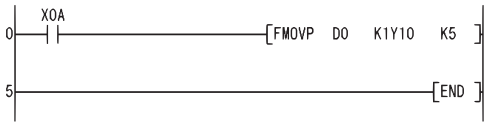
（出错代码： 4101）

• 使用特殊直接软元件时，传输数量超过了 6144。(QnACPU)　　（出错代码： 4101）

程序示例

(1) 当 XA 变为 ON 时，以下程序将 D0 中数据的低 4 位输出到 Y10 到 Y23。

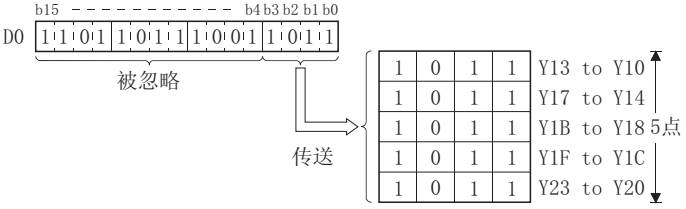
[梯形图模式]



[列表模式]

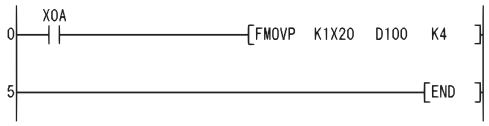
步	指令	软元件		
0	LD	X0A		
1	FMOVP	D0	K1Y10	K5
5	END			

[动作]



(2) 当 XA 变为 ON 时，以下程序将 X20 到 X23 中的数据输出到 D100 到 D103。

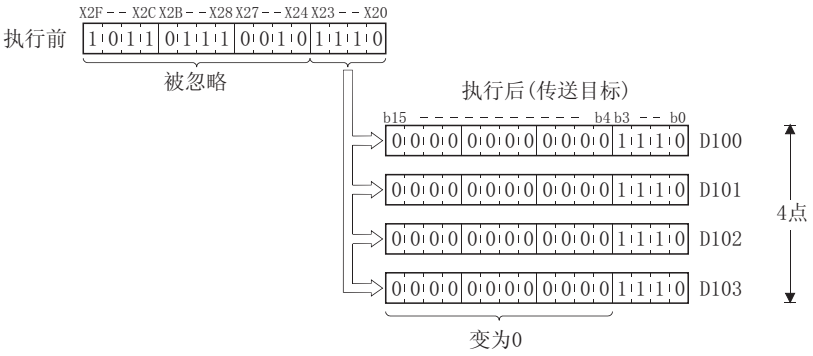
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0A		
1	FMOVP	K1X20	D100	K4
5	END			

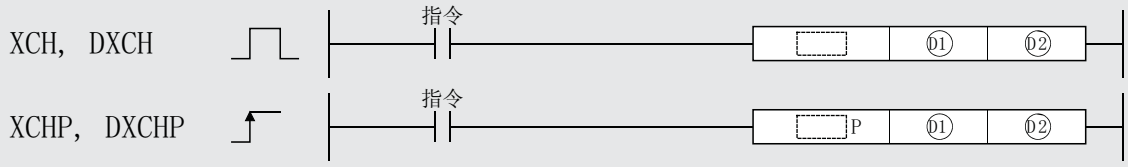
[动作]



6.4.8 16 位和 32 位数据交换 (XCH(P)、DXCH(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

表示指令符号XCH/DXCH。



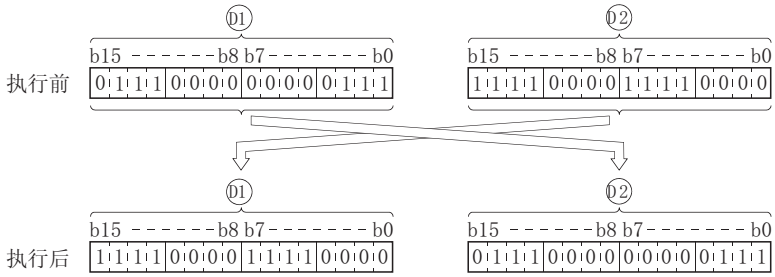
(D1)、(D2) ： 存储将进行交换的数据的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(D1)		○						--	
(D2)		○						--	

功 能

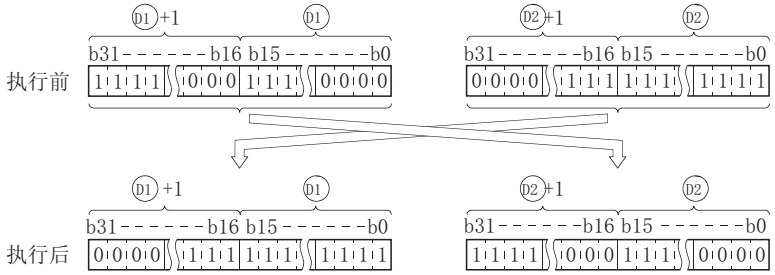
XCH

(1) 在 (D1) 和 (D2) 之间执行 16 位数据交换。



DXCH

(1) 在 (D1)+1、(D1) 和 (D2)+1、(D2) 之间执行 32 位数据交换。



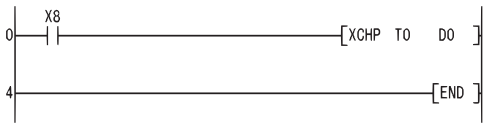
运算错误

(1) 不存在与 XCH(P) 和 DXCH(P) 指令相关的运行错误。

程序示例

(1) 当 X8 变为 ON 时，以下程序将 T0 的当前值与 D0 的内容进行交换。

[梯形图模式]

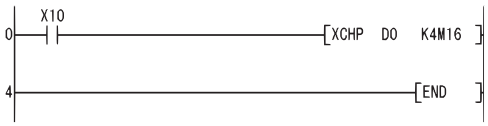


[列表模式]

步	指令	软元件	
0	LD	X8	
1	XCHP	T0	D0
4	END		

(2) 当 X10 变为 ON 时，以下程序将 D0 的内容与 M16 到 M31 中的数据进行交换。

[梯形图模式]

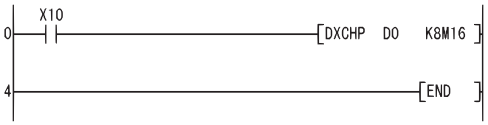


[列表模式]

步	指令	软元件	
0	LD	X10	
1	XCHP	D0	K4M16
4	END		

(3) 当 X10 变为 ON 时，以下程序将 D 和 D1 的内容与 M16 到 M47 中的数据进行交换。

[梯形图模式]

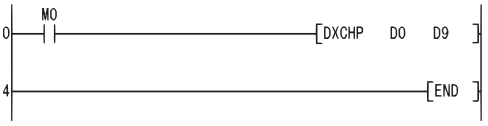


[列表模式]

步	指令	软元件	
0	LD	X10	
1	DXCHP	D0	K8M16
4	END		

(4) 当 M0 变为 ON 时，以下程序将 D0 和 D1 的内容与 D9 和 D10 的内容进行交换。

[梯形图模式]

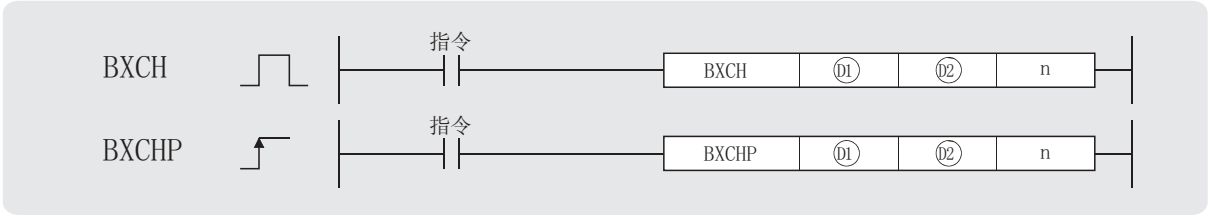


[列表模式]

步	指令	软元件	
0	LD	M0	
1	DXCHP	D0	D9
4	END		

6.4.9 块 16 位数据交换 (BXCH(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

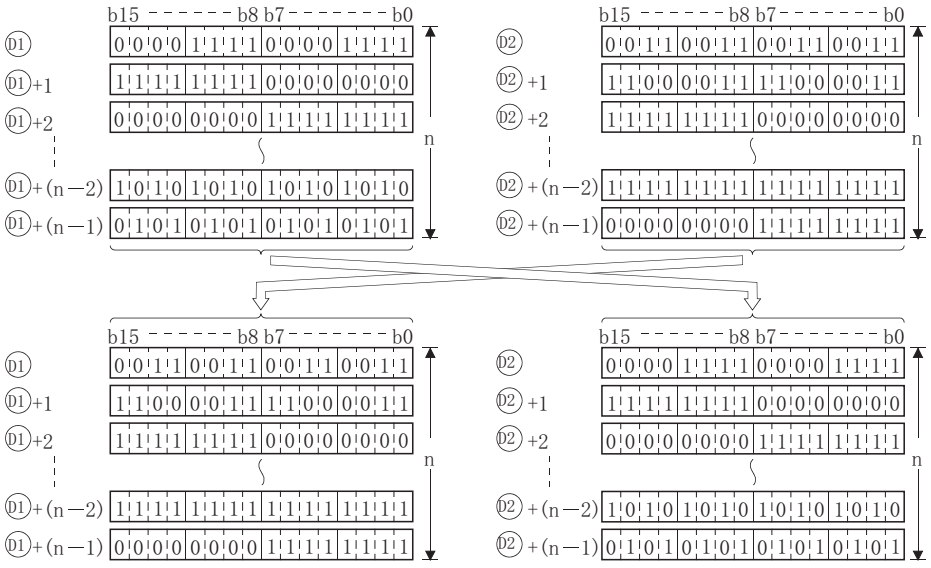


(D1)、(D2)：存储将进行交换的数据的软元件的起始号 (BIN16 位)。
n　　：交换数 (BIN16 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
(D1)	--	○		--					--
(D2)	--	○		--					--
n	○	○		○					--

功 能

(1) 将从 (D1) 指定的软元件开始的软元件中的 n 点 16 位数据与从 (D2) 指定的软元件开始的软元件中的 n 点 16 位数据进行交换。



运算错误

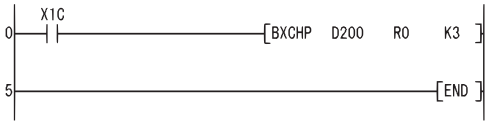
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 (D1) 或 (D2) 软元件开始的 n 点软元件范围超出了相关软元件。（出错代码： 4101）
- (D1) 和 (D2) 软元件相互重叠。（出错代码： 4101）

程序示例

(1) 当 X1C 变为 ON 时，以下程序将从 D200 起始的 3 点 16 位数据与从 R0 起始的 3 点 16 位数据进行交换。

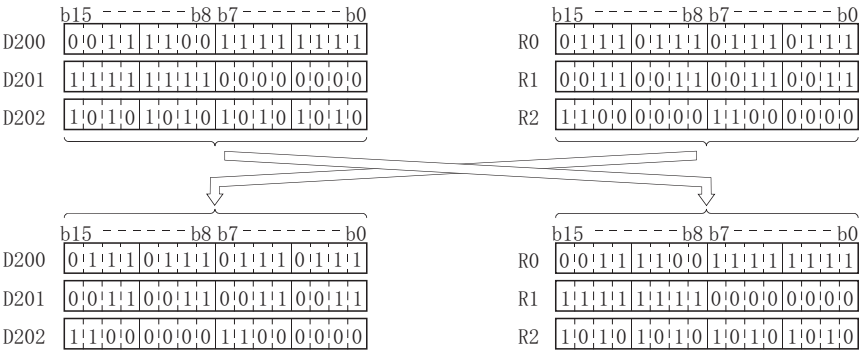
[梯形图模式]



[列表模式]

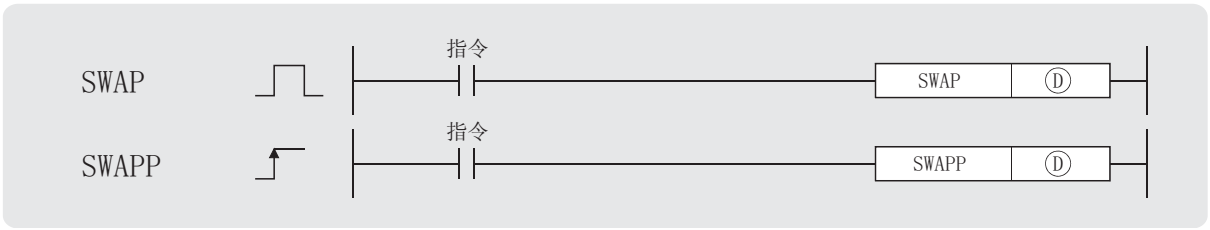
步	指令	软元件		
0	LD	X1C		
1	BXCHP	D200	R0	K3
5	END			

[动作]



6.4.10 高字节和低字节交换 (SWAP(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

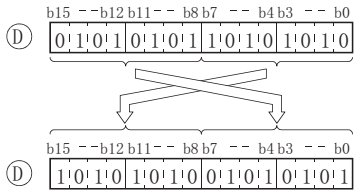


Ⓓ : 存储数据的软元件的起始号 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	○							--	

功 能

(1) 将 Ⓓ 指定的软元件的高 8 位和低 8 位进行交换。



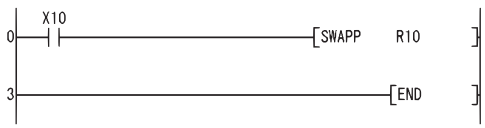
运算错误

(1) 不存在与 SWAP(P) 指令相关的运行错误。

程序示例

(1) 当 X10 变为 ON 时，以下程序将 R10 的高 8 位和低 8 位进行交换。

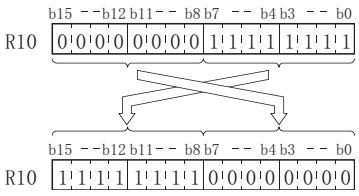
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X10
1	SWAPP	R10
3	END	

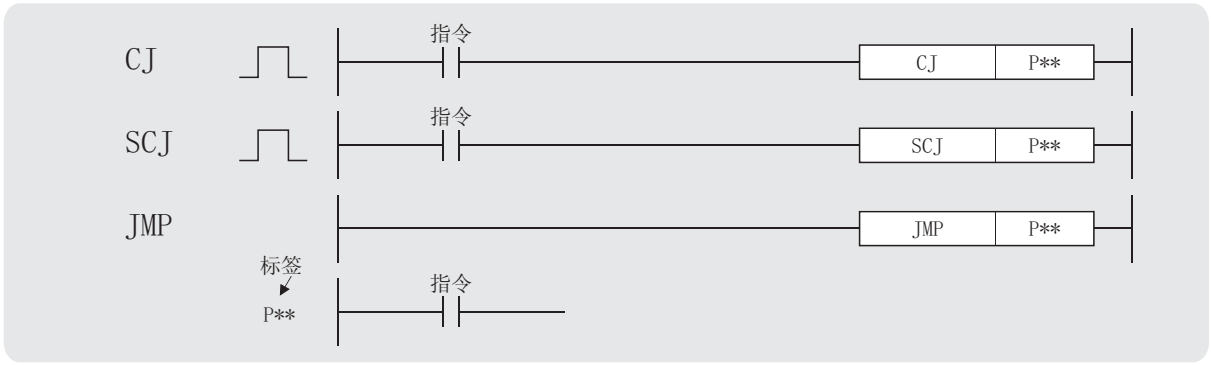
[动作]



6.5 程序分支指令

6.5.1 指针分支指令(CJ、SCJ、JMP)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



P** ：跳转目标的指针号（软元件名称）。

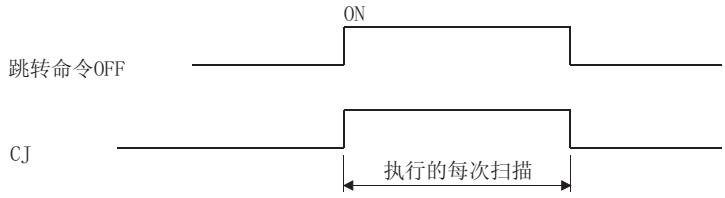
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P
P	--								○

功 能

CJ

(1) 当执行指令为 ON 时，执行同一程序文件内的指定的指针号的程序。

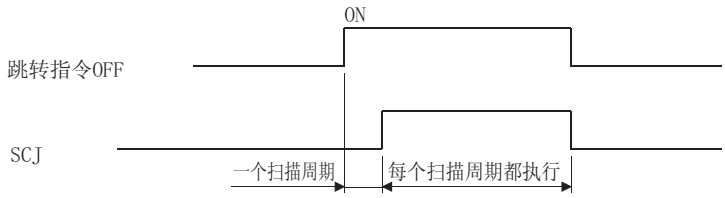
(2) 当跳转命令为 OFF 时，执行程序的下一步。



SCJ

(1) 当执行指令从 OFF 变为 ON 后，执行下一步的程序。

(2) 当跳转命令为 OFF 或者从 ON 变为 OFF 时，执行程序的下一步。



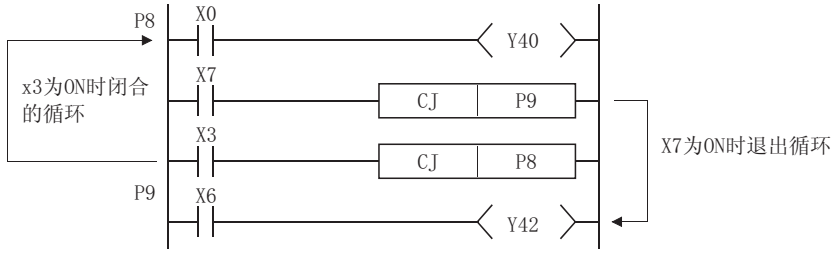
JMP

(1) 无条件地执行地同一程序文件内的指定的指针号的程序。

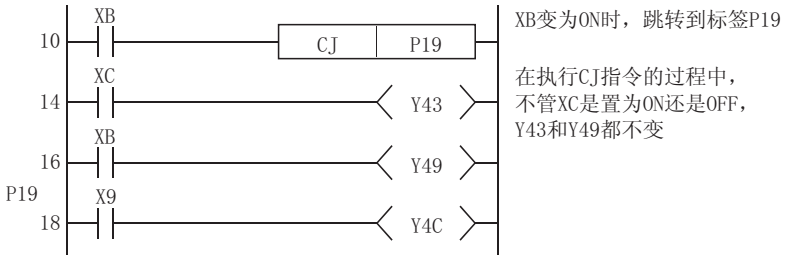
要 点

使用跳转指令时，请注意以下要点。

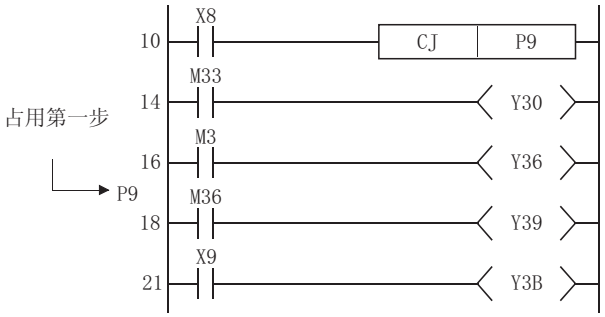
1. 在定时器线圈已经变为 ON 后，如果试图使用 CJ、SCJ 或 JMP 指令来跳转已经变为 ON 的线圈定时器，那么进行精确测量是不可能的。
2. 如果使用 CJ、SCJ 或 JMP 指令强制跳转到 OUT 指令，那么扫描时间会缩短。
3. 如果使用 CJ、SCJ 或 JMP 指令强制跳转到程序尾部，那么扫描时间会缩短。
4. CJ、SCJ 或 JMP 指令可用于跳转到当前正执行的步骤之前的某一步。
   然而，有必要考虑退出循环的方法，以便在该过程中看门狗定时器不会超时。



5. 已通过 CJ、SCJ 或 JMP 实现了的跳转的软元件不发生变化。



6. 标签 (P*) 占用第一步。



7. 跳转指令只可用于同一程序文件内的指针号码。
8. 在跳转运算期间，如果跳转到跳转范围内的一个指针号码，那么程序将从跳转目标的指针号码开始继续执行。

运算错误

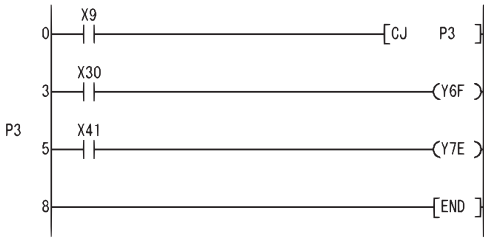
(1) 在以下情况中会返回运行错误，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 在 END 指令之前没有出现指定的指针号码。 (出错代码： 4210)
- 在同一程序里已经指定了一个并不用作标签的指针号码。 (出错代码： 4210)
- 已经指定了一个公共指针。 (出错代码： 4210)

程序示例

(1) 当 X9 变为 ON 时，以下程序跳转到 P3。

[梯形图模式]

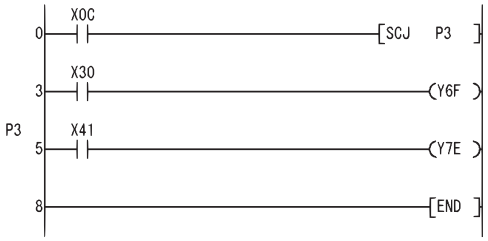


[列表模式]

步	指令	软元件
0	LD	X9
1	CJ	P3
3	LD	X30
4	OUT	Y6F
5	P3	
6	LD	X41
7	OUT	Y7E
8	END	

(2) 在 XC 变为 ON 后，以下程序从下一次扫描开始跳转到 P3。

[梯形图模式]

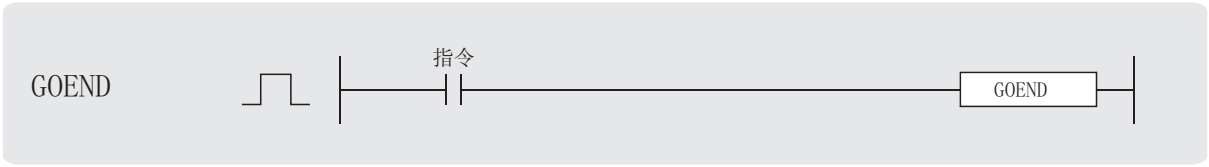


[列表模式]

步	指令	软元件
0	LD	X0C
1	SCJ	P3
3	LD	X30
4	OUT	Y6F
5	P3	
6	LD	X41
7	OUT	Y7E
8	END	

6.5.2 跳转到 END(GOEND)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

(1) 跳转到同一程序文件内的 FEND 或 END 指令。

运算错误

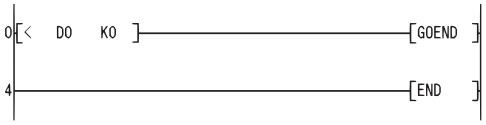
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 为 ON，并且出错代码存储住 SD0 中。

- 在执行 CALL 和 ECALL 指令之后并在执行 RET 指令之前已经执行了 GOEND 指令。 (出错代码： 4211)
- 在执行 FOR 指令之后并在执行 NEXT 指令之前已经执行了 GOEND 指令。 (出错代码： 4200)
- 在执行中断程序期间，已经在执行了 IRET 指令之前执行了 GOEND 指令。 (出错代码： 4221)
- 在 CHKCIR 和 CHKEND 指令块之间执行了 GOEND 指令。 (出错代码： 4230)
- 在 IX 和 IXEND 指令块之间执行了 GOEND 指令。 (出错代码： 4231)

程序示例

(1) 如果 D0 容纳的是一个负数，那么以下程序跳转到 END 指令。

[梯形图模式]



[列表模式]

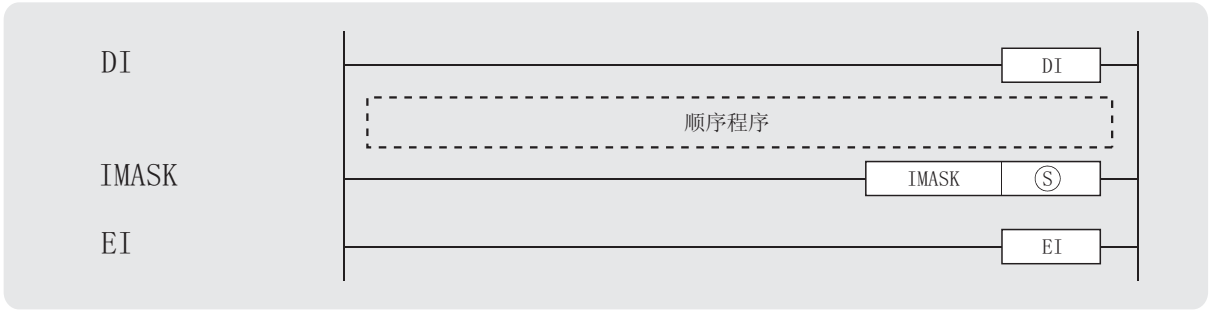
步	指令	软元件	
0	LD<	D0	K0
3	GOEND		
4	END		

6.6 程序执行控制指令

6.6.1 中断禁止 / 允许指令，中断程序屏蔽 (DI、EI、IMASK)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 使用基本型 QCPU 时



Ⓢ : 中断屏蔽数据，或者存储中断屏蔽数据的软元件的起始号 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	--	○		--					

功 能

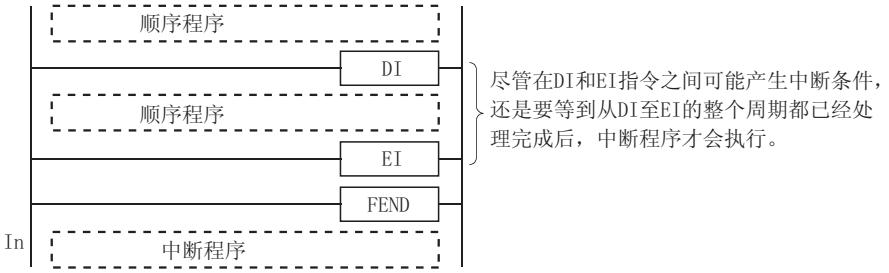
DI

(1) 即使出现了中断程序的启动条件。也只能在执行了 EI 指令之后才允许执行中断程序。

(2) 当电源为 ON 或者复位 CPU 模块时，输入一个 DI 状态。

EI

EI 指令用于清除由于执行 DI 指令而引起的中断禁止状态，并创建一个状态；在该状态下，可以执行由 IMASK 指令确认的中断指针号码所指定的中断程序。
如果 IMASK 指令不执行，则禁止 I32 到 I47。



IMASK

(1) 允许或禁止执行使用从 ⓢ 指定的软元件开始的 8 点的位模式指定的中断指针来标记的中断程序。

- 1(ON)..... 允许执行中断程序
- 0(OFF)..... 禁止执行中断程序

(2) 对应于单个位的中断指针号码如下所示：

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ⓢ	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
ⓢ+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
ⓢ+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
ⓢ+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
ⓢ+4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
ⓢ+5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
ⓢ+6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
ⓢ+7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112

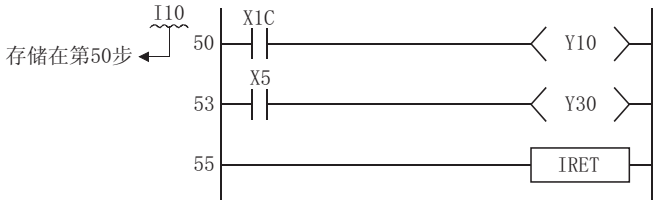
(3) 当电源为 ON 或者复位 CPU 模块时，I0 到 I31 和 I48 到 I127 的中断程序将变为允许执行状态，但 I32 到 I47 的中断程序变为禁止执行状态。

(4) 软元件 ⓢ、ⓢ+1、ⓢ+2 以及 ⓢ+3 到 ⓢ+7 的状态存储在 SD715 到 SD717 和 SD781 到 SD785(IMASK 指令屏蔽模式的存储区域）中。

(5) 尽管特殊寄存器被分隔为 SD715 到 SD717 和 SD781 到 SD785，软元件号应该被连续地指定为从 ⓢ 到 ⓢ+7。

要 点

1. 一个中断指针占用一步。



2. 关于中断条件的内容，请参照以下手册：
   - QCPU 用户手册（功能解说 / 程序基础篇）
3. 在中断程序执行期间，DI 状态（中断禁止）是激活的。
   不要在中断程序中插入 EI 指令以试图执行多个中断，即在中断程序内部嵌套运行中断程序。
4. 如果 EI 和 DI 指令位于主控制中，那么不管 MC 指令处于执行还是非执行状态，这些指令都将执行。

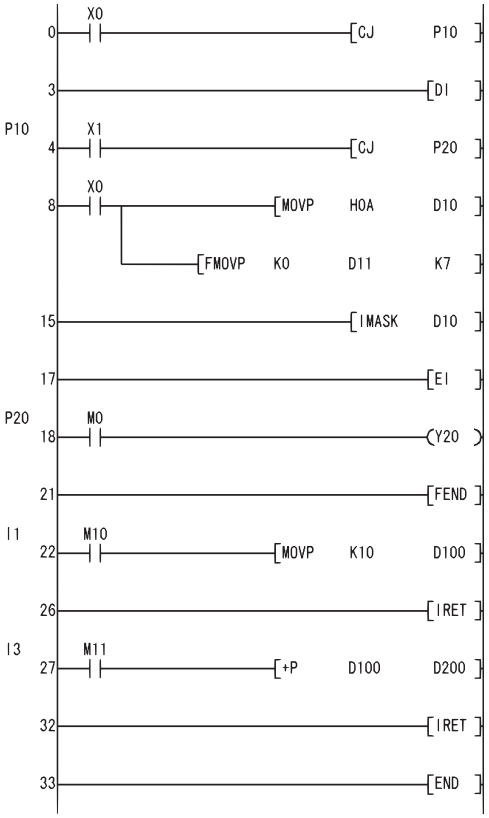
运算错误

(1) 不存在与 DI、EI 或 IMASK 指令相关的运算错误。

程序示例

(1) 以下程序旨在当 X0 为 ON 时，只允许中断指针号码为 I1 和 I3 的中断程序执行。

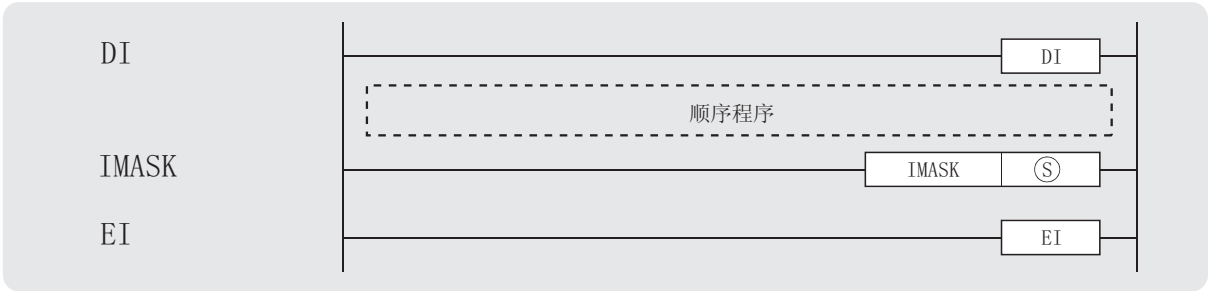
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	CJ	P10		
3	DI			
4	P10			
5	LD	X1		
6	CJ	P20		
8	LD	X0		
9	MOVP	H0A	D10	
11	FMOVP	K0	D11	K7
15	IMASK	D10		
17	EI			
18	P20			
19	LD	M0		
20	OUT	Y20		
21	FEND			
22	I1			
23	LD	M10		
24	MOVP	K10	D100	
26	IRET			
27	I3			
28	LD	M11		
29	+P	D100	D200	
32	IRET			
33	END			

2 使用高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU 时



Ⓢ : 存储中断屏蔽数据的软元件的起始号 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	--	○		--					

功 能

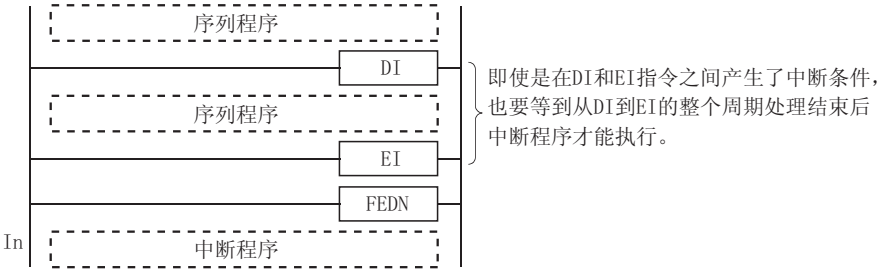
DI

(1) 即使出现了中断程序启动条件，也只能在执行了 EI 指令之后才允许执行中断程序。

(2) DI 状态在电源变为 ON 或者复位 CPU 模块时输入。

EI

EI 指令用于清除由于执行 DI 指令而引起的中断禁止状态，并创建一个状态；在该状态下，可以执行由 IMASK 指令确认的中断指针号码所指定的中断程序，也可以执行固定周期执行类型程序。
如果 IMASK 指令不执行，则禁止 I32 到 I47。



IMASK

(1) 允许或禁止执行使用从 Ⓢ 指定的软元件开始的 16 点位模式所指定的中断地址标记的中断程序。

- 1(ON)..... 允许执行中断程序
- 0(OFF)..... 禁止执行中断程序

(2) 对应于单个位的中断指针号码如下所示：

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ⓢ	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
Ⓢ+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
Ⓢ+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
Ⓢ+3	I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
Ⓢ+4	I79	I78	I77	I76	I75	I74	I73	I72	I71	I70	I69	I68	I67	I66	I65	I64
Ⓢ+5	I95	I94	I93	I92	I91	I90	I89	I88	I87	I86	I85	I84	I83	I82	I81	I80
Ⓢ+6	I111	I110	I109	I108	I107	I106	I105	I104	I103	I102	I101	I100	I99	I98	I97	I96
Ⓢ+7	I127	I126	I125	I124	I123	I122	I121	I120	I119	I118	I117	I116	I115	I114	I113	I112
Ⓢ+8	I143	I142	I141	I140	I139	I138	I137	I136	I135	I134	I133	I132	I131	I130	I129	I128
Ⓢ+9	I159	I158	I157	I156	I155	I154	I153	I152	I151	I150	I149	I148	I147	I146	I145	I144
Ⓢ+10	I175	I174	I173	I172	I171	I170	I169	I168	I167	I166	I165	I164	I163	I162	I161	I160
Ⓢ+11	I191	I190	I189	I188	I187	I186	I185	I184	I183	I182	I181	I180	I179	I178	I177	I176
Ⓢ+12	I207	I206	I205	I204	I203	I202	I201	I200	I199	I198	I197	I196	I195	I194	I193	I192
Ⓢ+13	I223	I222	I221	I220	I219	I218	I217	I216	I215	I214	I213	I212	I211	I210	I209	I208
Ⓢ+14	I239	I238	I237	I236	I235	I234	I233	I232	I231	I230	I229	I228	I227	I226	I225	I224
Ⓢ+15	I255	I254	I253	I252	I251	I250	I249	I248	I247	I246	I245	I244	I243	I242	I241	I240

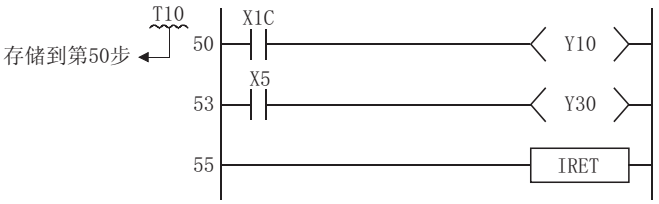
(3) 当电源为 ON 或者复位 CPU 模块时， I0 到 I31 和 I48 到 I255 的中断程序将变为允许执行状态，但 I32 到 I47 的中断程序变为禁止执行状态。

(4) 软元件 Ⓢ、Ⓢ+1、Ⓢ+2 以及Ⓢ+3 到 Ⓢ+15 的状态存储在 SD715 到 SD717 和 SD781 到 SD793(IMASK 指令屏蔽模式的存储区域）中。

(5) 尽管特殊寄存器分隔为 SD715 到 SD717 和 SD781 到 SD793，软元件号应该被连续地指定为从 Ⓢ 到 Ⓢ+15。

要 点

1. 一个中断指针占用一步。



2. 关于中断条件的内容，请参照以下手册：
   - QCPU 用户手册（功能解说 / 程序基础篇）
3. 在中断程序执行期间，DI 状态（中断禁止）是激活的。
   不要在中断程序中插入 EI 指令以试图执行多个中断，即在中断程序内部嵌套运行中断程序。
4. 如果 EI 和 DI 指令位于主控制中，那么不管 MC 指令处于执行还是非执行状态，这些指令都将执行。

运算错误

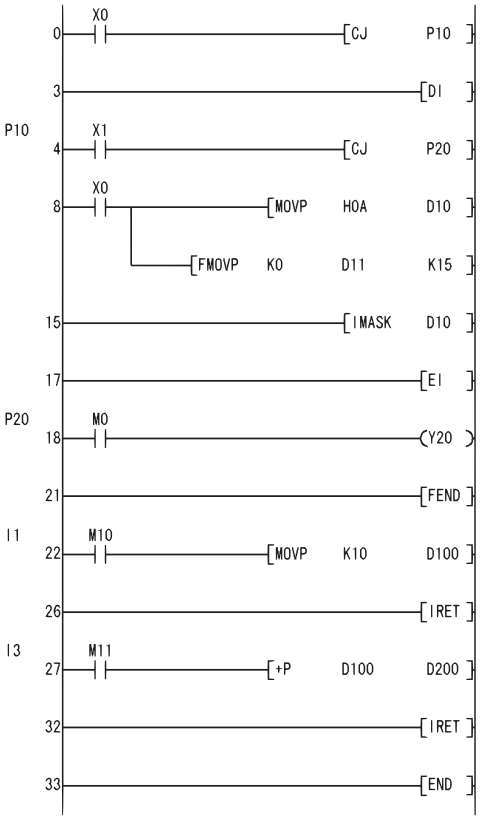
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

• 由 Ⓢ 指定的软元件超出了相应软元件的范围。
(只对于通用型 QCPU) (出错代码： 4101)

程序示例

(1) 当 X0 为 ON 时，以下程序为中断指针号码标记的中断程序创建允许执行状态。

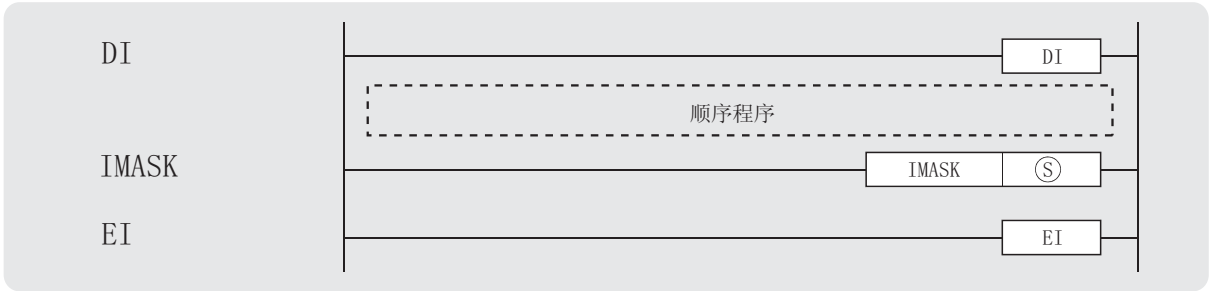
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	CJ	P10		
3	DI			
4	P10			
5	LD	X1		
6	CJ	P20		
8	LD	X0		
9	MOVP	H0A	D10	
11	FMOVP	K0	D11	K15
15	IMASK	D10		
17	EI			
18	P20			
19	LD	M0		
20	OUT	Y20		
21	FEND			
22	I1			
23	LD	M10		
24	MOVP	K10	D100	
26	IRET			
27	I3			
28	LD	M11		
29	+P	D100	D200	
32	IRET			
33	END			

3 使用 QnACPU/Q4ARCPU 时



Ⓢ : 存储中断屏蔽数据的软元件的起始号 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	--	○		--					

功 能

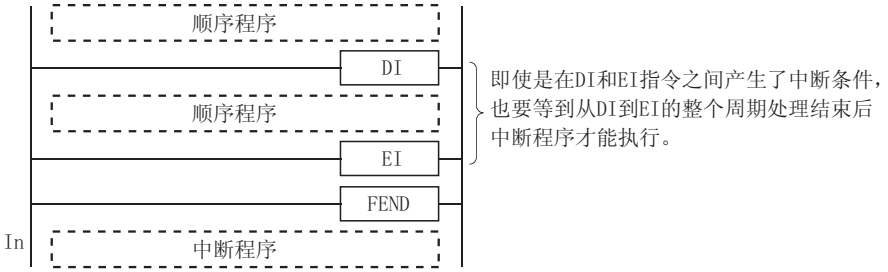
DI

(1) 即使出现了中断程序启动条件，也只能在执行了 EI 指令之后才允许执行中断程序。

(2) 当电源变为 ON 或者复位 CPU 模块时，输入一个 DI 状态。

EI

EI 指令用于清除由于执行 DI 指令而引起的中断禁止状态，并创建一个状态：在该状态下，可以执行由 IMASK 指令确认的中断指针号码所指定的中断程序。
如果 IMASK 指令不执行，则禁止 I32 到 I47。



IMASK

(1) 允许或禁止执行使用从 Ⓢ 指定的软元件开始的 3 点位模式所指定的中断地址标记的中断程序。

- 1(ON)..... 允许执行中断程序
- 0(OFF)..... 禁止执行中断程序

(2) 对应于单个位的中断指针号码如下所示：

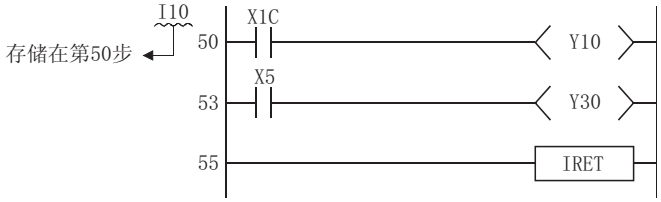
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Ⓢ	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
Ⓢ+1	I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
Ⓢ+2	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

(3) 当电源变为 ON 或者复位 CPU 模块时，中断程序 I0 到 I31 处于允许执行状态，而中断程序 I32 到 I47 处于禁止执行状态。

(4) 软元件 Ⓢ、Ⓢ+1、Ⓢ+2 的状态存储在 SD715 到 SD717(IMASK 指令屏蔽模式的存储区域）中。

要 点

1. 一个中断指针占用一步。



2. 关于中断条件的内容，请参照以下手册：
   - QnACPU 编程手册（基础篇）
3. 在中断程序执行期间，DI 状态（中断禁止）是激活的。
   不要在中断程序中插入 EI 指令以试图执行多个中断，即在中断程序内部嵌套运行中断程序。
4. 如果 EI 和 DI 指令位于主控制中，那么不管 MC 指令处于执行还是非执行状态，这些指令都将执行。

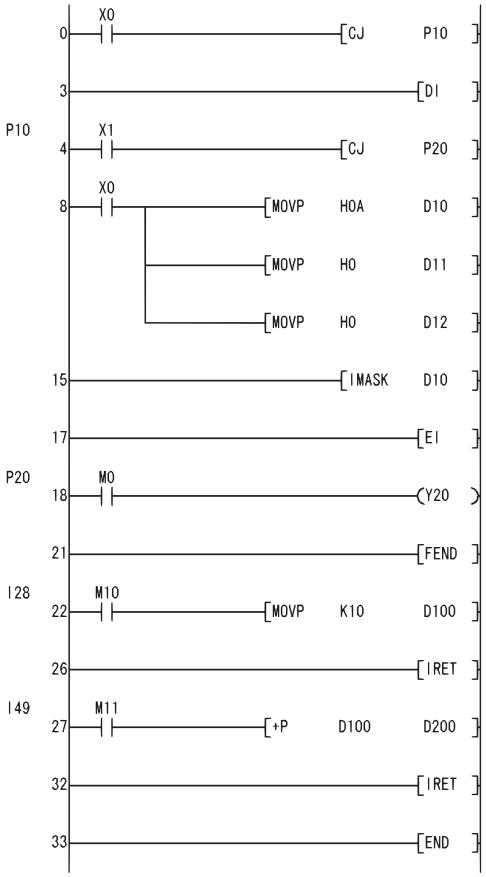
运算错误

(1) 不存在与 DI、EI 或 IMASK 指令相关的运算错误。

程序示例

(1) 当 X0 为 ON 时，以下程序为中断指针号码标记的中断程序创建允许执行状态。

[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	CJ	P10	
3	DI		
4	P10		
5	LD	X1	
6	CJ	P20	
8	LD	X0	
9	MOVP	H0A	D10
11	MOVP	H0	D11
13	MOVP	H0	D12
15	IMASK	D10	
17	EI		
18	P20		
19	LD	M0	
20	OUT	Y20	
21	FEND		
22	I28		
23	LD	M10	
24	MOVP	K10	D100
26	IRET		
27	I49		
28	LD	M11	
29	+P	D100	D200
32	IRET		
33	END		

6.6.2 从中断程序中恢复(IRET)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

(1) 表示对中断程序的处理完成。

(2) 执行 IRET 指令后，返回到对顺序程序的处理。

运算错误

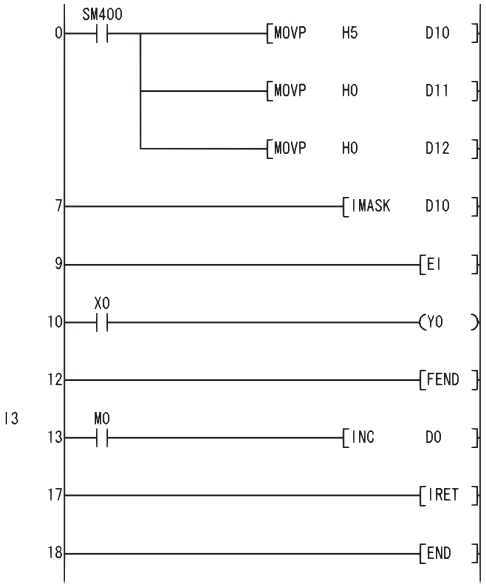
(1) 在发生以下运行错误的情况中，错误标志(SM0)为 ON，并且出错代码存储在 SD0 中。

- 没有与中断地址相应的指针。 (出错代码： 4220)
- 在执行中断程序之前已经先发出了 IRET 指令。 (出错代码： 4223)
- 在产生中断之后而在执行 IRET 指令之前已经先执行了 END、FEND、GOEND 或 STOP 指令。 (出错代码： 4221)
- 在恒定周期执行型程序中执行了 IRET 指令。 (只对于通用型 QCPU) (出错代码： 4223)

程序示例

(1) 在产生 3 号中断的时候，如果 M0 为 ON，则以下程序给 D0 加 1。

［梯形图模式］



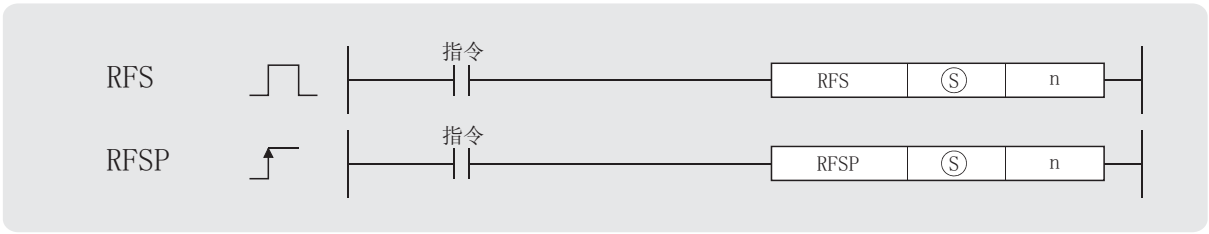
［列表模式］

步	指令	软元件	
0	LD	SM400	
1	MOVP	H5	D10
3	MOVP	H0	D11
5	MOVP	H0	D12
7	IMASK	D10	
9	EI		
10	LD	X0	
11	OUT	Y0	
12	FEND		
13	I3		
14	LD	M0	
15	INC	D0	
17	IRET		
18	END		

6.7 I/O刷新指令

6.7.1 I/O刷新 (RFS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ： 执行刷新操作的软元件的起始软元件号（位）。

n ： 要刷新的点数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○ (限于X、Y)	--							--
n	○	○							--

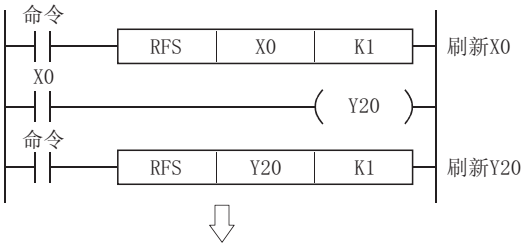
功 能

(1) 只对在一个扫描周期间扫描过的软元件进行刷新，并从外部源获取输入数据或者将数据输出到输出模块。

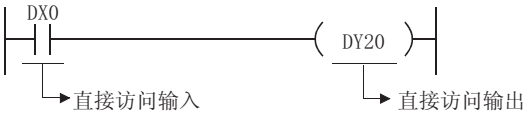
(2) 从外部源获取输入数据或者将数据输出到外部源只在执行了 END 指令之后批量执行，所以，不可能在一个扫描周期的执行期间将脉冲信号输出到外部源。
执行 I/O 刷新指令的时候，程序在执行过程中对相应软元件号的输入 (X) 或输出 (Y) 进行强制刷新。因此，可以在一个扫描周期期间将脉冲信号输出到外部源。

(3) 使用直接访问输入 (DX) 或直接访问输出 (DY) 以 1 点为单元刷新输入 (X) 或输出 (Y)。

[基于RFS指令的程序]



[基于直接访问输入和直接访问输出的程序]



运算错误

(1) 在发生以下运行错误的情况中，错误标志 (SM0) 置为 ON，并且出错代码存储在 SD0 中。

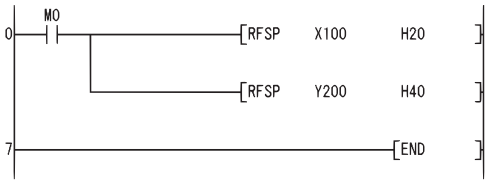
• 从 ⓢ 指定的软元件开始的 n 点范围超出了最近的范围。

（出错代码： 4101）

程序示例

(1) 当 M0 变为 ON 时，以下程序对 X100 到 X11F 和 Y200 到 Y23F 进行刷新。

[梯形图模式]



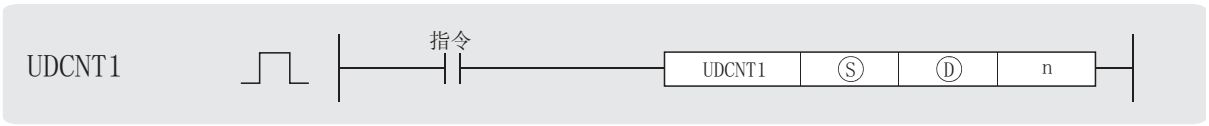
[列表模式]

步	指令	软元件	
0	LD	M0	
1	RFSP	X100	H20
4	RFSP	Y200	H40
7	END		

6.8 其它方便的指令

6.8.1 单相输入加法或减法计数器 (UDCNT1)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : Ⓢ + 0: 用于计数输入的输入号 (位)。

Ⓢ + 1: 用于设定加法、减法计数 (位)。

- OFF: 加法计数 (计数时数字增加)。
- ON: 减法计数 (计数时数字减少)。

Ⓓ : 可以根据 UDCNT1 指令开始计数的计数器号 (软元件名称)。

n : 设定值 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○ (限于 X)	--	--	--		--			--
Ⓓ	--	○ *1 (限于 C)	--	--		--			--
n	△ *1	△ *1	△ *1	○					--

*1: 不能使用局部软元件和为各个程序设置的文件寄存器。

功 能

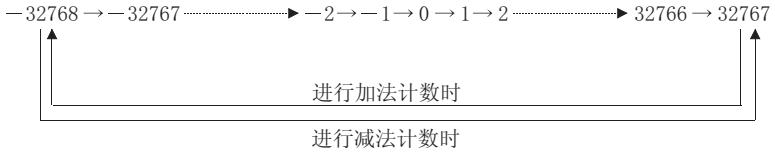
(1) 当 Ⓢ 指定的输入从 OFF 变为 ON 时， Ⓓ 指定的计数器的当前值将得到更新。

(2) 计数的方向取决于 Ⓢ+1 指定的输入状态是 ON 还是 OFF。

- OFF: 加法计数 (以增加当前值的方式计数)
- ON: 减法计数 (以减少当前值的方式计数)

(3) 计数处理的执行方式描述如下：

- 当进行加法计数时，如果当前值变得与 n 指定的设定值相同，则 Ⓓ 指定的计数器触点变为 ON。然而，即使是当 Ⓓ 指定的计数器触点变为 ON 时，对当前值的计数还将继续。(参见程序示例 (1))
- 当进行减法计数时，如果当前值变得比设定值少 1，则 Ⓓ 指定的计数器触点变为 OFF。(参见程序示例 (1))
- Ⓓ 指定的计数器是环计数器。
  在当前值为 32767 时，如果正进行加法计数，则当前值将变为 32768：另外，在当前值为 -32768 时，如果正进行减法计数，则当前值将变为 32767。
  基于当前值进行的计数处理如下所示：



(4) 当计数命令从 OFF 变为 ON 时，基于 UDCNT1 指令的计数处理开始计数；当计数命令从 ON 变为 OFF 时，中止计数。
当计数命令再一次从 OFF 变为 ON 时，从计数中止时的值重新开始计数。

(5) RST 指令清除 ⓓ 指定的计数器的当前值并将触点置为 OFF。

要 点

1. UDCNT1 指令将参数软元件数据注册到 CPU 模块的工作区，而实际的计数运算被处理为系统中断。
（当命令输入置为 OFF 或者当 CPU 模块置为 STOP 然后又置为 RUN 时，将清除注册到 CPU 模块工作区的软元件数据）。
因此，可计数的脉冲的 ON 和 OFF 时间必须比 CPU 模块的中断间隔长。CPU 模块的中断间隔如下所示。

CPU 模块类型号	中断间隔
高性能型 QCPU、过程 CPU、通用型 QCPU	1 ms
QnACPU、Q4ARCPU	5 ms

2. 在基于 UDCNT1 指令的一次计数期间（当命令输入为 ON 时），设定值不能更改。如果要更改设定值，请首先将命令输入置为 OFF。
3. 其它指令不能使用 UDCNT1 指令指定的计数器。如果这些计数器为其它指令所使用，则它们不能返回精确的计数值。
4. 在所有正执行的程序内部，UDCNT1 指令最多可以使用 6 次。第七个以及以后的 UDCNT1 指令不会得到处理。

运算错误

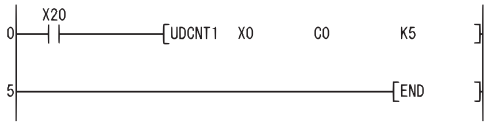
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

- 由 ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU） （出错代码：4101）

程序示例

(1) 在 X20 变为 ON 后，该程序使用 C0（加法和减法计数器）对 X0 从 OFF 变为 ON 的次数进行计数。

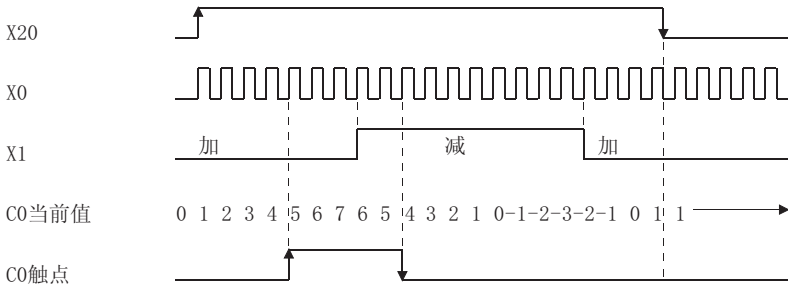
[梯形图模式]



[列表模式]

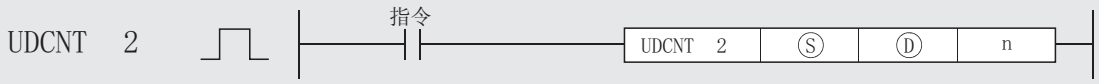
步	指令	软元件
0	LD	X20
1	UDCNT1	X0 C0 K5
5	END	

[动作]



6.8.2 两相输入加法或减法计数器 (UDCNT2)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : Ⓢ + 0: 用于计数输入的输入号 (A 相脉冲)(位)。
Ⓢ + 1: 用于计数输入的输入数字 (B 相脉冲)(位)。
Ⓓ : 可以根据 UDCNT2 指令进行计数的计数器号 (软元件名称)。
n : 设定值 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○ (限于 X)	--	--	--					--
Ⓓ	--	○ *1 (限于 C)	--	--					--
n	△ *1	△ *1	△ *1	○					--

*1: 不能使用局部软元件和为各个程序设置的文件寄存器。

功 能

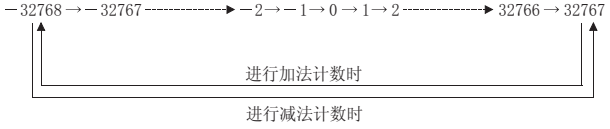
(1) Ⓓ 指定的计数器的当前值根据 Ⓢ 指定的输入状态 (A 相脉冲) 和 Ⓢ+1 指定的输入的状态 (B 相脉冲) 实现更新。

(2) 按以下方式确定计数的方向：

- 当 Ⓢ 为 ON 时，如果 Ⓢ+1 从 OFF 变为 ON，则执行加法计数运算 (计数器的当前值增加)。
- 当 Ⓢ 为 ON 时，如果 Ⓢ+1 从 ON 变为 OFF，则执行减法计数运算 (计数器的当前值减少)。
- 如果 Ⓢ 为 OFF，则不执行计数运算。

(3) 计数处理的执行方式描述如下：

- 当进行加法计数时，如果当前值变得与 n 指定的设定值相同，则 Ⓓ 指定的计数器触点变为 ON。然而，即使是当 Ⓓ 指定的计数器触点变为 ON 时，对当前值的计数还将继续。(参见程序示例 (1))
- 当进行减法计数时，如果当前值变得比设定值少上 1，则 Ⓓ 指定的计数器触点变为 OFF。(参见程序示例 (1))
- Ⓓ 指定的计数器是环计数器。当前值为 32767 时，如果正进行加法计数，则当前值将变为 32768：另外，当前值为 -32768 时，如果正进行减法计数，则当前值将变为 32767。
  基于当前值进行的计数处理如下所示：



(4) 当计数命令从 OFF 变为 ON 时，基于 UDCNT2 指令的计数处理开始计数；当计数命令从 ON 变为 OFF 时，中止计数。
当计数命令再一次从 OFF 变为 ON 时，从计数中止时的值重新开始计数。

(5) RST 指令清除 Ⓓ 指定的计数器的当前值并将触点置为 OFF。

要 点

1. UDCNT2 指令将参数软元件数据注册到 CPU 模块的工作区，而实际的计数运算被处理为系统中断。
（当命令输入置为 OFF 或者当 CPU 模块置为 STOP 然后又置为 RUN 时，将清除注册到 CPU 模块工作区的软元件数据）。
因此，可计数的脉冲的 ON 和 OFF 时间必须比 CPU 模块的中断间隔长。CPU 模块的中断间隔如下所示。

CPU 模块类型号	中断间隔
高性能型 QCPU、过程 CPU、通用型 QCPU	1 ms
QnACPU、Q4ARCPU	5 ms

2. 在基于 UDCNT2 指令的一次计数期间（当命令输入为 ON 时），设定值不能更改。如果要更改设定值，请首先将命令输入置为 OFF。
3. 其它指令不能使用 UDCNT2 指令指定的计数器。如果这些计数器为其它指令所使用，则它们不能返回精确的计数值。
4. 在所有正执行的程序内部，UDCNT2 指令最多可以使用 5 次。第六个以及以后的 UDCNT2 指令不会得到处理。

运算错误

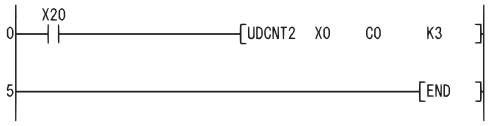
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

- 由 Ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU） （出错代码： 4101）

程序示例

(1) 在 X20 变为 ON 后，以下程序根据 X0 和 X1 的状态并按照 C0（加法和减法计数器）的指示执行一次计数运算。

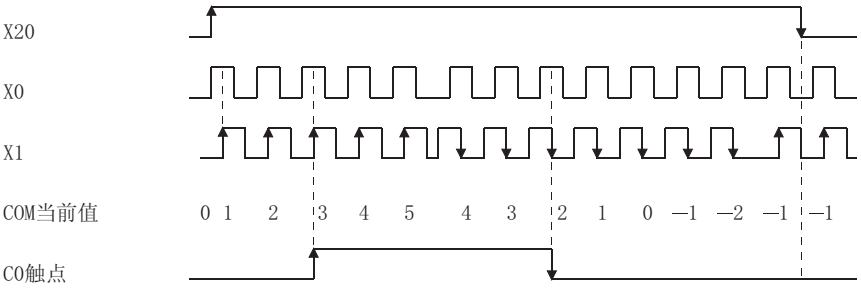
［梯形图模式］



［列表模式］

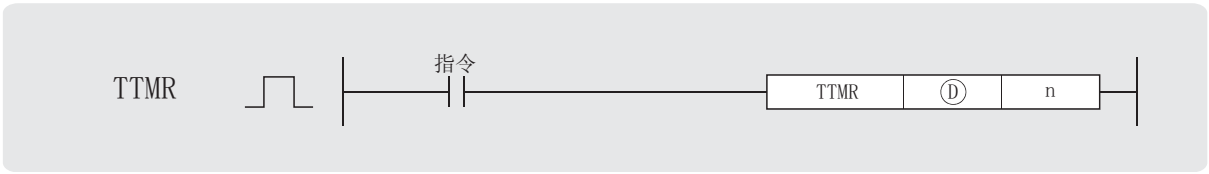
步	指令	软元件		
0	LD	X20		
1	UDCNT2	X0	C0	K3
5	END			

［动作］



6.8.3 教学计时器 (TTMR)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ : Ⓓ + 0: 存储测量值的软元件 (BIN16 位)。

Ⓓ + 1: 为 CPU 模块系统所使用 (BIN16 位)。

n : 测量值的乘数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ	--	○		--					--
n	--	○		○					--

功 能

(1) 对测量命令为 ON 的时间以秒为单位进行测量，然后给它乘以 n 指定的乘数，并将结果存储在 Ⓓ 指定的软元件中。

(2) 当测量命令从 OFF 变为 ON 时，会将 Ⓓ+0 或者 Ⓓ+1 指定的软元件清除。

(3) n 可以指定的乘数如下所示。

n	乘数
0	1
1	10
2	100

要 点

1. 时间测量在执行 TTMR 指令的时候进行。如果使用 JMP 或者类似指令跳转到 TTMR 指令，那么要获得精确的测量值是不可能的。
2. 在执行 TTMR 指令期间不要更改 n 指定的乘数。更改该乘数会导致返回值不精确。
3. TTMR 指令也可用于低速执行类型的程序中。
4. 因为 Ⓓ+1 指定的软元件为 CPU 模块系统所使用，因此用户不应该更改它的值。如果用户确实更改了这个值，那么存储在 Ⓓ 指定的软元件中的值再也不会是精确的。

(4) 如果 “n” 指定的值不在范围 0 到 2 之内，则不会执行任何运算。

运算错误

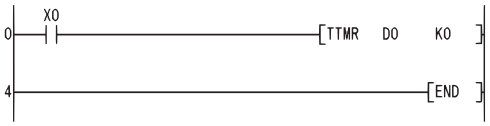
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

• 由 ⓓ 指定的软元件超出了相应软元件范围。
(只对于通用型 QCPU) (出错代码： 4101)

程序示例

(1) 以下程序将 X0 为 ON 的时间总和存储在 D0 中。

[梯形图模式]

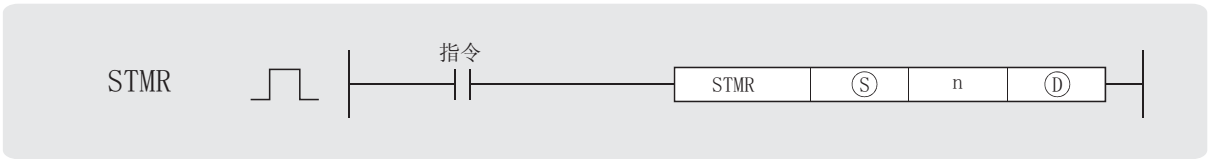


[列表模式]

步	指令	软元件	
0	LD	X0	
1	TTMR	D0	K0
4	END		

6.8.4 特殊功能定时器 (STMR)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 定时器号 (字)。

n : 设定数值 (BIN16 位)。

Ⓓ : Ⓓ + 0: OFF 延迟定时器输出 (位)。

Ⓓ + 1: OFF 后脉冲计时器输出 (位)。

Ⓓ + 2: ON 后脉冲计时器输出 (位)。

Ⓓ + 3: ON 延迟 +OFF 延迟定时器输出 (位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	△ *1	--	--					--
n	○	○	○	○					--
Ⓓ	○	--	--	--					--

*1: 只能作为定时器 (T) 数据使用。

功 能

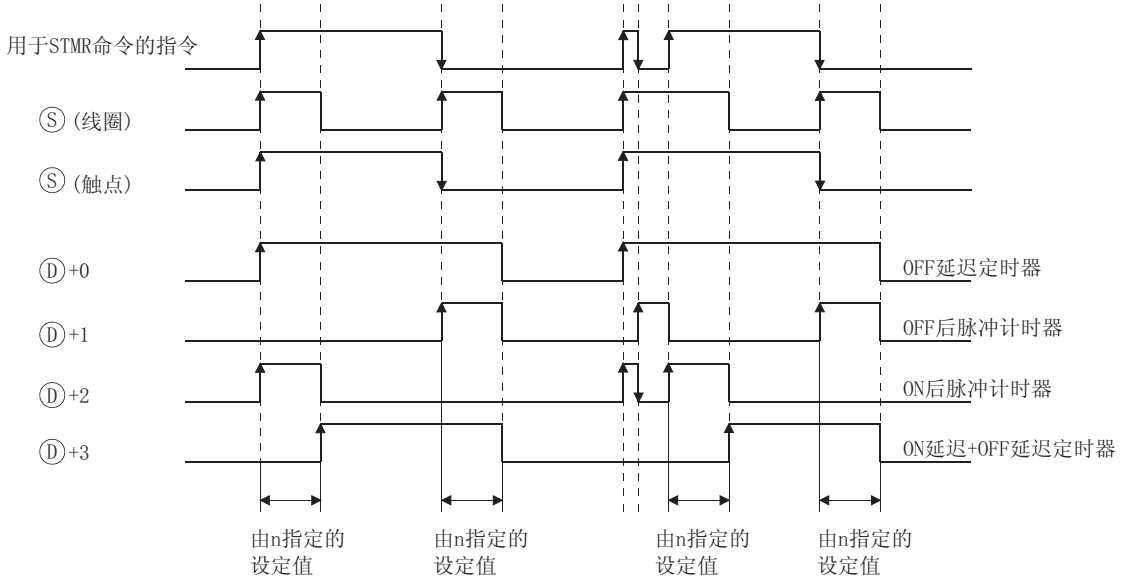
(1) STMR 指令使用从 Ⓓ 指定软元件开始的 4 点来执行 4 种类型的定时器输出。

- 关延迟定时器输出 (Ⓓ+0)
  在 STMR 指令的上升沿处变为 ON，并且在命令的下降沿后，n 设定的计数时间结束后变为 OFF。
- OFF 后脉冲定时器输出 (Ⓓ+1)
  在 STMR 指令的下降沿处变为 ON，并且在 n 指定的时间结束后变为 OFF。
- ON 后脉冲定时器输出 (Ⓓ+2)
  在 STMR 指令的上升沿处变为 ON，并且在 n 指定的时间过去后或在 STMR 指令的命令变为 OFF 时变为 OFF。
- 开延迟定时器输出 (Ⓓ+3)
  在定时器线圈的下降沿处变为 ON，并且在 SRMR 指令命令下降沿后，n 改定的计数时间结束后变 OFF。

(2) Ⓢ 指定的定时器线圈在 STMR 指令命令的上升沿处变为 ON，并开始当前值的测量。

- 当定时器线圈测量到由 n 指定的设定值的点后，进入时间到达状态并且变为 OFF。
- 如果 STMR 指令在定时器线圈到达设定时间之前变为 OFF，它将保持 ON 状态。这时定时器测量暂停。当 STMR 指令命令再次变为 ON 时，当前值清 0 并且重新开始测量。

(3) 定时器触点在 STMR 指令命令的上升沿处变为 ON，并且在下降沿到达以后，定时器线圈在 STMR 指令命令的下降沿处变为 OFF。
定时器触点只能被 CPU 模块系统使用，用户不能使用。



(4) STMR 指令指定的定时器的当前值的测量是在 STMR 指令执行时进行的。
如果 STMR 指令由于 JMP 或类似指令而跳转，它将不可能得到正确的测量。

(5) 用于 Ⓓ 指定的定时器的测量单元与低速定时器是一样的。

(6) n 值可设定在 0 到 32767 之间。
如果 n 不在 0 到 32767 之间将被作为无处理。

(7) Ⓢ 指定的定时器不能被 OUT 指令使用。
如果 STMR 和 OUT 指令使用相同的定时器号，则不能正确运行。

运算错误

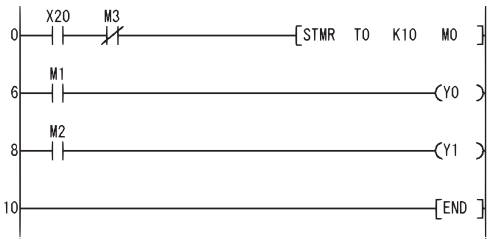
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关错误代将码存储在 SD0。

• 由 Ⓓ 指定的软元件超出了相应软元件范围。
(只对于通用型 QCPU) (出错代码：4101)

程序示例

(1) 当 X20 变 ON 时，下述程序每间隔一秒交替将 Y0 和 Y1 变 ON 和 OFF（闪烁）。
（使用 100ms 定时器）

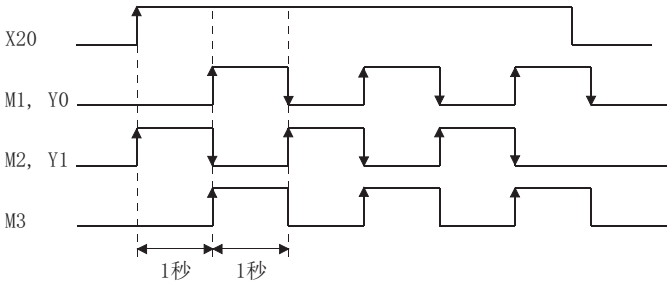
[梯形图模式]



[列表模式]

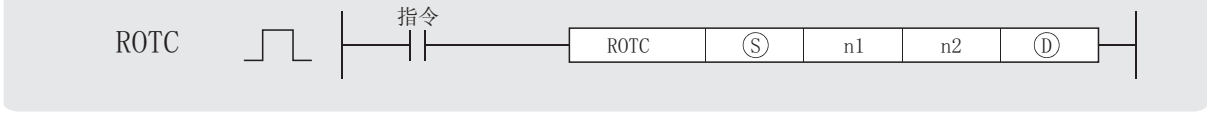
步	指令	软元件		
0	LD	X20		
1	ANI	M3		
2	STMR	T0	K10	M0
6	LD	M1		
7	OUT	Y0		
8	LD	M2		
9	OUT	Y1		
10	END			

[时序图]



6.8.5 旋转台就近控制 (ROTC)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : Ⓢ + 0: 用于测量旋转台旋转次数。(用于系统)(BIN16 位)
Ⓢ + 1: 调用站号 (BIN16 位)。
Ⓢ + 2: 调用项目号 (BIN16 位)。
n1 : 旋转台的分区数 (2 到 32767)(BIN16 位)。
n2 : 低速部分数 (0 到 n1 以下的值)(BIN16 位)。
Ⓓ : Ⓓ + 0: A 相输入信号 (位)。
Ⓓ + 1: B 相输入信号 (位)。
Ⓓ + 2: 0 点检测输入信号 (位)。
Ⓓ + 3: 高速正转输出信号 (用于系统)(位)。
Ⓓ + 4: 低速正转输出信号 (用于系统)(位)。
Ⓓ + 5: 停止输出信号 (用于系统使用)(位)。
Ⓓ + 6: 低速反转输出信号 (用于系统)(位)。
Ⓓ + 7: 高速反转输出信号 (用于系统)(位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	○		--					--
n1	○	○		○					--
n2	○	○		○					--
Ⓓ	○	--		--					--

功 能

(1) 为了将号码被 Ⓢ+2 指定的项目移走或存放在分区数与 n1 指定的值相同的旋转台上，此控制功能允许旋转台的就近路径旋转到 Ⓢ+1 指定的站号位置上。

(2) 项目号和站号是以逆时针旋转方向分配的项目来控制的。

(3) 系统使用Ⓢ+0 作为记数器来记录哪个项目号从站号 0 的哪个号开始记数。不要重写顺控程序数据。
当用户重写了此数据时，将不可能进行精确控制。

(4) n2 值应该低于 n1 指定的平台分区数。

(5) Ⓓ+0 和 Ⓓ+1 是用来检测旋转台旋转方向是向前或向后的 A 相或 B 相输入信号。
专 A 相脉冲为 ON，B 相脉冲是在上升沿或下降沿决定了旋转方向。
- 专 B 相为上升沿： 向前旋转 (顺时针方向旋转)
- 专 B 相为下降沿： 向后旋转 (逆时针方向旋转)

(6) 当项目号 0 到达 0 号站时，ⓓ+2 是状态变为 ON 时的 0 点检测输出信号。
当 ⓓ+2 指定的软元件状态变为 ON 并且 ROTC 指令被执行时， ⓢ+0 被清零。
最好先执行清零运行命令，然后用 ROTC 指令开始就近路径旋转。

(7) ⓓ+3 到 ⓓ+7 的数据包括用来控制平台运行的输出信号。
ⓓ+3 到 ⓓ+7 中的一个元件的输出信号将变为 ON 来响应 ROTC 指令的执行结果。

(8) 如果在 ROTC 指令之前的运行结果变为 OFF， ⓓ+3 到 ⓓ+7 的所有信号都将为 OFF，不执行就近路径旋转控制运行。

(9) ROTC 指令在所有程序中只能执行一次。
如果多次使用此指令将导致错误运行。

(10)当 ⓢ+0 到 ⓢ+2 的值或 n2 的值大于 n1，不执行此操作。

运算错误

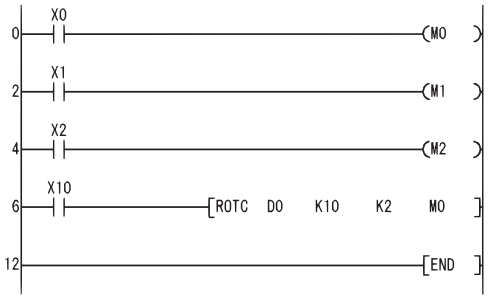
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

- 由 ⓢ 或 ⓓ 指定的软元件超出了相应软元件范围。
  (只对于通用型 QCPU)　　(出错代码： 4101)

程序示例

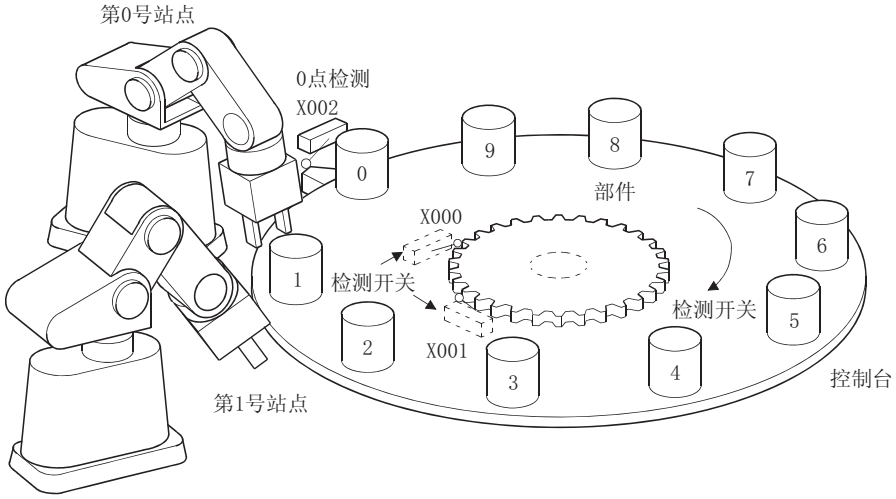
(1) 下述程序将在 D2 部分的部件存放在旋转台上 D1 部分站，前面和后面的两部分决定了在旋转台低速旋转时的马达的旋转方向和控制速度。

［梯形图模式］



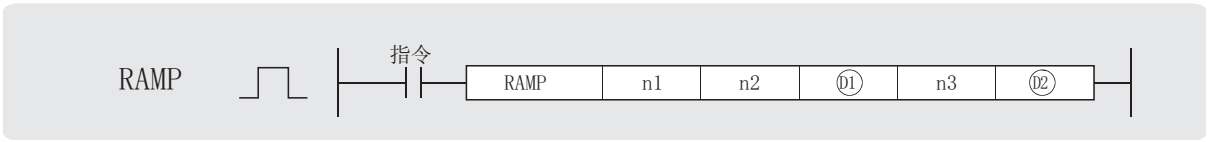
［列表模式］

步	指令	软元件			
0	LD	X0			
1	OUT	M0			
2	LD	X1			
3	OUT	M1			
4	LD	X2			
5	OUT	M2			
6	LD	X10			
7	ROTC	D0	K10	K2	M0
12	END				



6.8.6 斜坡信号 (RAMP)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n1 : 初始值 (BIN16 位)
n2 : 最终值 (BIN16 位)
D1 : D1 + 0: 当前值 (BIN16 位)
D1 + 1: 执行次数 (BIN16 位)
n3 : 移动次数 (BIN16 位)
D2 : D2 + 0: 完成软元件 (位)
D2 + 1: 用于在完成时存储数据的选择位 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	○			○				○	--
n2	○			○				○	--
D1	○			○				--	--
n3	○			○				○	--
D2	○			--				--	--

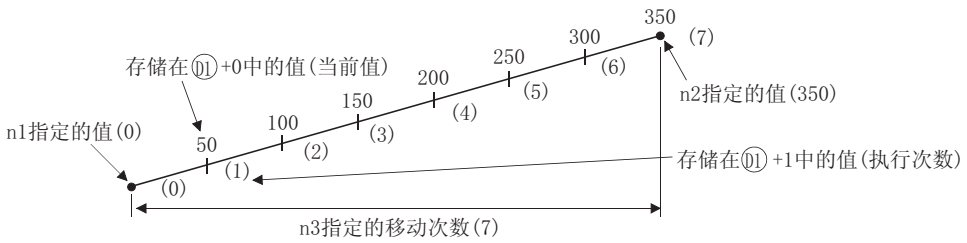
功 能

(1) 当执行指令为 ON，将执行以下处理。

- 该值按照 n3 指定的移动次数将 n1 指定的值移动到 n2 指定的值。
- 在 n3 中，为从 n1 移动到 n2 指定需要的扫描次数 (移动次数)。
  如果不在 0<n3<32768 范围内，则进行无处理。
- D1+1 为系统所用，用于存储本指令的执行次数。
- 一次 (一个扫描) 的变化值是通过下列公式计算出来的：

$$\text{一次(一个扫描)的变化值} = \frac{\text{(n2指定的值)-(n1指定的值)}}{\text{(n3指定的值)}}$$

例如 在 7 个扫描中从 0 变化到 350 时。



当在一次的变化值除不尽时，需要进行修正以通过 n3 指定的移动次数变为 n2 指定的值。因此，可能不能生成线性的斜坡。

(2) 执行了 n3 指定的移动次数的扫描后， ⓓ2 +0 指定的完成软元件将变为 ON。
完成软元件的 ON/OFF 状态以及 ⓓ1+0 的内容取决于 ⓓ2+1 指定的软元件的 ON/OFF 状态。

- 当 ⓓ2+1 为 OFF 时， ⓓ2+0 将在下一次扫描时变为 OFF，并且 RAMP 指令将从初始值开始重新开始移动。
- 当 ⓓ2+1 为 ON 时， ⓓ2+0 将保持 ON 状态不变， ⓓ1+0 的内容不变化。

(3) 在本指令的执行期间指令变为 OFF 时， ⓓ1+0 的内容在此处以后不变化。
当本指令再次变为 ON 时，RAMP 指令将从初始值开始重新开始移动。

(4) 在 ⓓ2+0 指定的完成软元件变为 ON 之前不要更改 n1 和 n2 指定的值。
因为在每次扫描都使用相同的公式来计算存储在 ⓓ1+1 中的值，所以，更改 n1 或 n2 的值可能导致急剧变化。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

- 由 ⓓ1 或 ⓓ2 指定的软元件超出了相应软元件范围。
(只对于通用型 QCPU) (出错代码： 4101)

注意事项

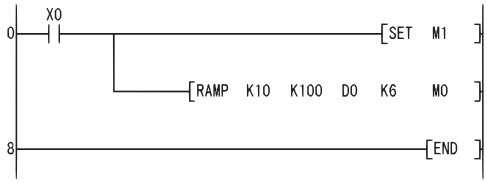
(1) 当通过位软元件对 ⓓ1 进行位数指定时，其指定只有在符合以下条件时才可以使用。

- 位数的指定： K8

程序示例

(1) 以下程序在总共 6 次的扫描中将 D0 中的内容从 10 变化到 100，并在完成移动时保存 D0 的内容。

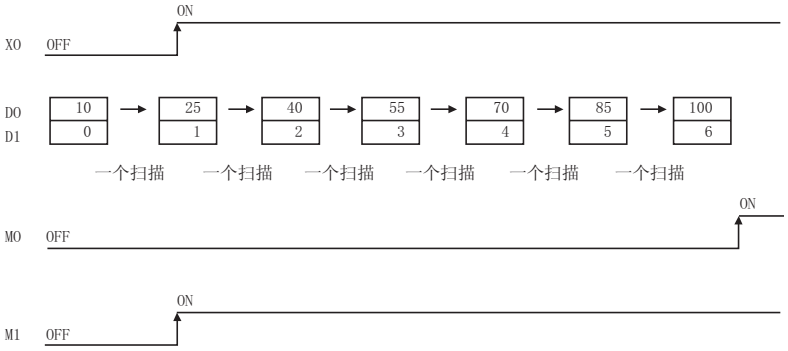
[梯形图模式]



[列表模式]

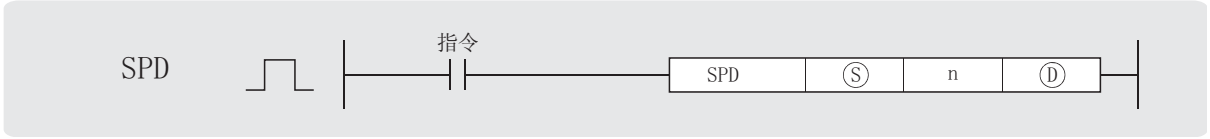
步	指令	软元件
0	LD	X0
1	SET	M1
2	RAMP	K10 K100 D0 K6 M0
8	END	

[时序图]



6.8.7 脉冲密度测量 (SPD)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 脉冲输入 (位)

n : 测量时间 (单位 :ms) (BIN16 位)

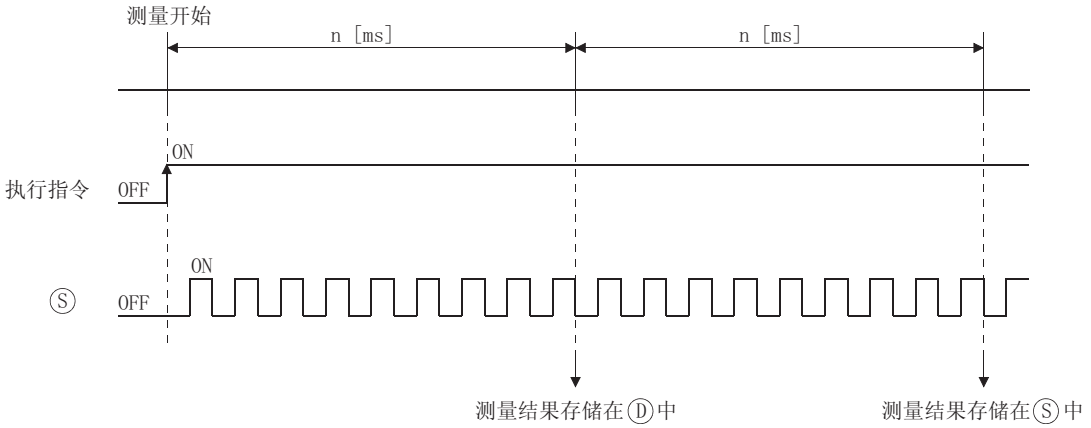
Ⓓ : 存储测量结果的软元件的起始号 (BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○ (限于 X)	--		--					--
n	△ *1	△ *1		○					--
Ⓓ	--	△ *1		--					--

*1: 不能使用局部软元件和为各个程序设置的文件寄存器。

功 能

(1) 只在 n 指定的时间内对来自 Ⓢ 指定的软元件的输入进行计数，并将计数结果存储在 Ⓓ 指定的软元件中。



(2) 如果 SPD 指令引导的测量已经完成，则从 0 开始进行下一次测量。
要中止 SPD 指令引导的测量，请将命令变为 OFF。

要 点

1. SPD 指令将参数软元件中的数据注册到 CPU 模块的工作区，而实际的计数运算在系统中断期间执行。
（当命令输入变为 OFF 或者当 CPU 模块变为 STOP 然后又变为 RUN 时，将清除注册到 CPU 模块工作区的软元件数据）。
因此，可计数的脉冲的 ON 和 OFF 时间必须比 CPU 模块的中断间隔长。CPU 模块的中断间隔如下所示。

CPU 模块类型号	中断间隔
高性能型 QCPU、过程控制 CPU、通用型 QCPU	1 ms
QnACPU、Q4ARCPU	5 ms

2. • 使用高性能型 QCPU 或过程控制 CPU 的时候：
当 n=0 时，不对指令进行处理。
• 使用 QnACPU 的时候：
当 n=0 或者 n 不是 5 的倍数时，不对指令进行处理。
3. 在所有正执行的程序内部，SPD 指令最多可以使用 6 次。第七个以及以后的 SPD 指令不会得到处理。
4. 当通过 SPD 指令执行测量时（当指令输入为 ON），无法更改设定值。在更改设定值前，首先要关闭指令输入。

运算错误

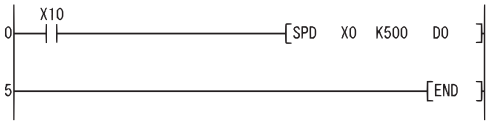
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0。

• 由 Ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU） （出错代码：4101）

程序示例

(1) 当 X10 变为 ON 时，以下程序在 500ms 的时间内对输入到 X0 的脉冲进行测量，并将结果存储在 D0 中。

[梯形图模式]

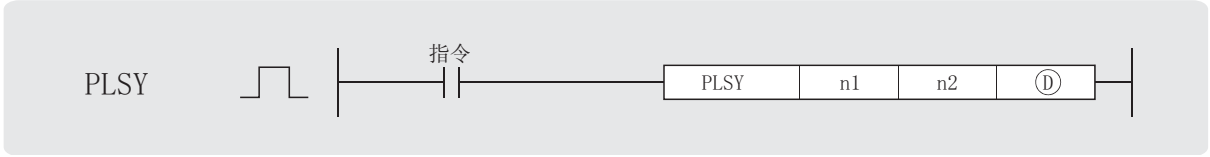


[列表模式]

步	指令	软元件		
0	LD	X10		
1	SPD	X0	K500	D0
5	END			

6.8.8 固定周期脉冲输出 (PLSY)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n1 ： 频率或存储频率的软元件号 (BIN16 位)。

n2 ： 输出次数或存储输出次数的软元件号 (BIN16 位)。

Ⓓ ： 执行脉冲输出的软元件号 (位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	○				○				--
n2	○				○				--
Ⓓ	△ *1				--				--

*1: 只能使用输出 (Y)。

功 能

(1) 将 n1 指定的频率的脉冲输出到 Ⓓ 指定的输出号 (Y) 的输出模块中，其中输出次数由 n2 指定。

(2) n1 可以指定的频率范围是 1 到 100Hz。
如果 n1 超出了范围 1 到 100Hz，则 PLSY 指令将执行无处理。

(3) n2 可以指定的输出次数的范围是 0 到 65535(0000H 到 FFFFH)。
如果 n2 设定为 “0”，脉冲将会连续输出。

(4) Ⓓ 中指定的脉冲输出只能指定与输出模块相对应的输出号。

(5) 脉冲输出开始于 PLSY 指令指令的上升沿。
PLSY 指令变为 OFF 时，脉冲输出将中止。

要 点

1. PLSY 指令将参数软元件数据注册到 CPU 模块的工作区，而实际的输出运算在系统中断期间执行。
（当命令输入变为 OFF 或者当 CPU 模块变为 STOP 然后又变为 RUN 时，将清除注册到 CPU 模块工作区的软元件数据）。
因此，可计数的脉冲的 ON 和 OFF 时间必须比 CPU 模块的中断间隔长。CPU 模块的中断间隔如下所示。

CPU 模块类型号	中断间隔
高性能型 QCPU、过程 CPU、通用型 QCPU	1 ms
QnACPU、Q4ARCPU	5 ms

2. 在通过 PLSY 指令输出脉冲的过程中，不要更改 PLSY 指令的参数（在命令输入变为 ON 时。）在更改参数之前请先将命令输入变为 OFF。
3. 在 CPU 模块执行的整个程序中，PLSY 指令只能使用一次。第二个及以后的 PLSY 都不会得到处理。

运算错误

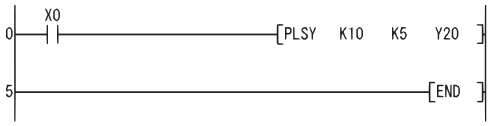
(1) 以下任何情况中出现运行错误，则错误标志 (SM0) 变为 ON。并且相关出错代码存储在 SD0。

- 由 Ⓓ 指定的软元件超出了相关软元件的范围。
（只对于通用型 QCPU)（出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序将一个 10Hz 脉冲输出到 Y20，输出次数为 5。

[梯形图模式]

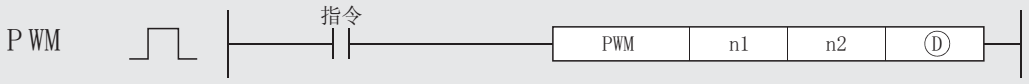


[列表模式]

步	指令	软元件		
0	LD	X0		
1	PLSY	K10	K5	Y20
5	END			

6.8.9 脉冲宽度调制 (PWM)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n1 ： ON 状态时间或存储 ON 状态时间的软元件号。(BIN16 位)

n2 ： 频率或存储频率的软元件号。(BIN16 位)

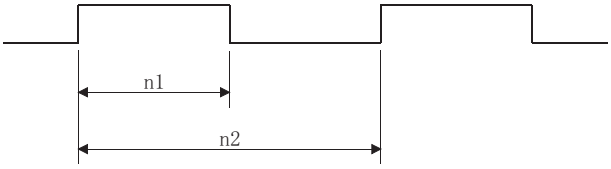
Ⓓ ： 执行脉冲输出的软元件号。(位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	○				○				--
n2	○				○				--
Ⓓ	△ *1				--				--

*1: 只能使用输出 (Y)。

功 能

(1) 在 n1 指定的状态为 ON(接通) 的时间中，将脉冲 (周期为 n2 所设定) 输出到 Ⓓ 指定的输出模块。



(2) n1 和 n2 的设置范围如下所示：

CPU 模块类型号	n1 和 n2 的设定范围 [ms] *2
高性能型 QCPU、过程控制 CPU、通用型 QCPU	1 到 65535 (0001H 到 FFFFH)
QnACPU、Q4ARCPU	5 到 65535 (0005H 到 FFFFH)

*2: 指定给 n1 的值应该小于或者等于指定给 n2 的值。

要 点

1. PWM 指令将指定的软元件数据注册到 CPU 模块的工作区。实际的输出运算被 CPU 模块处理为中断。
（当命令输入变为 OFF 或者当 CPU 模块变为 STOP 然后又变为 RUN 时，将清除注册到 CPU 模块工作区的软元件数据）。
因此，可计数的脉冲的 ON 和 OFF 时间必须比 CPU 模块的中断间隔长。CPU 模块的中断间隔如下所示。：

CPU 模块类型号	n1，n2 的中断间隔
高性能型 QCPU、过程控制 CPU、通用型 QCPU	1 ms
QnACPU、Q4ARCPU	5 ms

因此，在 CPU 模块执行的所有程序中，PWM 指令只能使用一次。

2. 在下列情况下指令不会得到执行：
   - 当 n1 和 n2 都为 0 时
   - 当 n1 和 n2 不都是 5 的倍数时（只有当使用 QnACPU 的时候）
   - 当 n2 ≥ n1 时
   - 当 PWM 指令执行了两次以上时。
3. 在通过 PWM 指令输出脉冲的过程中，不要更改 PWM 指令的参数（在命令输入变为 ON 时）。
在更改参数之前请先将命令输入变为 OFF。

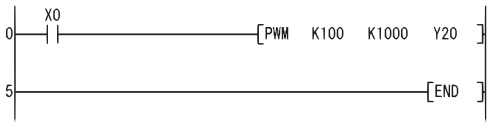
运算错误

(1) 以下任何情况中出现运行错误，则错误标志 (SM0) 变为 ON，并且相关出错代码存储在 SD0。
- 由 ⓓ 指定的软元件超出了相关软元件的范围。
（只对于通用型 QCPU） （出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序将一个 100ms 的脉冲输出到 Y20，频率为每秒钟一次。

[梯形图模式]

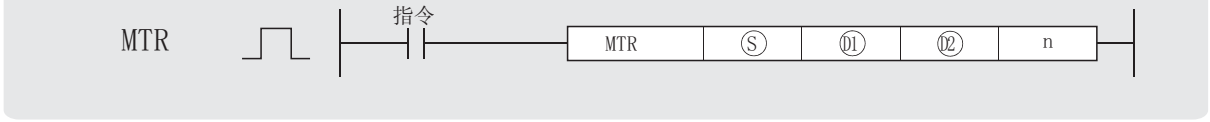


[列表模式]

步	指令	软元件		
0	LD	X0		
1	PWM	K100	K1000	Y20
5	END			

6.8.10 矩阵输入 (MTR)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 起始输入软元件。(位)

Ⓓ1 : 起始输出软元件。(位)

Ⓓ2 : 存储矩阵输入数据的起始号软元件。(位)

n : 输入行数。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○ (限于 X)		--						--
Ⓓ1	○ (限于 Y)		--						--
Ⓓ2	○		--						--
n	○		○						--

功 能

(1) 连续读取从 Ⓢ 指定的输入地址起始的 16 × n 点数据 (连续读取 n 行 16 点数据)，然后将从该运算中接收到的数据存储在 Ⓓ2 指定的软元件起始的软元件中。

(2) 在一个扫描周期中可以接收行一行 (16 点)。

(3) 从第一行到第 n 行的数据接收的过程是循序渐进的重复过程。

(4) 在从 Ⓓ2 指定的软元件起始的软元件中，第一个 16 点 (第 1 到 16 点) 存储第一行数据，下一个 16 点存储第二行数据。
因此，从 Ⓓ2 指定的软元件起始的 16 × n 点空间为 MTR 指令所占据。

(5) Ⓓ1 是用于选择将被读取的行的输出，且系统自动将它变为 ON 和 OFF。
它使用从 Ⓓ1 指定的软元件起始的 n 点。

(6) 只有可被 16 整除的软元件号才可以指定给 Ⓢ、Ⓓ1 和 Ⓓ2 。

(7) n2 的值不在范围 2 到 8 之内。

(8) 在下列情况下不执行任何处理：

- Ⓢ、Ⓓ1 或 Ⓓ2 指定的软元件号不能被 16 整除。
- Ⓢ 指定的软元件超出了实际的输入范围。
- Ⓓ1 指定的软元件超出了实际的输出范围。
- 从 Ⓓ2 指定的软元件起始的 16 × n 点空间超出了相关软元件范围。
- n2 的值不在范围 2 到 8 之内。

运算错误

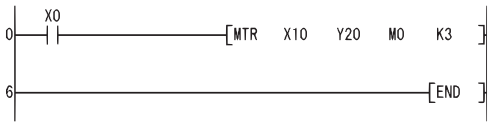
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储住 SD0 中。

- Ⓢ 指定了除输入 (X) 之外的软元件。 (出错代码： 4101)
- Ⓓ1 指定了除输出 (Y) 之外的软元件。 (出错代码： 4101)

程序示例

(1) 当 X0 变为 ON 时，以下程序接收从 X10 起始的 16 点× 3 行数据，并将它存储在从 M0 起始的软元件中。

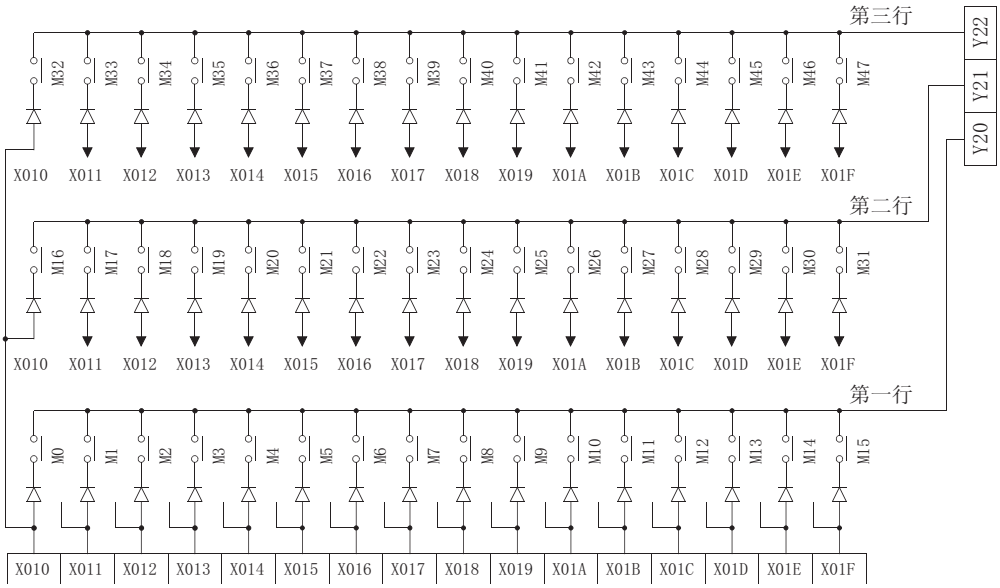
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	MTR	X10	Y20	M0	K3
6	END				

[动作]



注意事项

(1) 注意，MTR 指令对实际的输入和输出直接进行操作。

已被 MTR 指令变为 ON 的输出 Ⓓ1 不会在 MTR 命令变为 OFF 时变为 OFF。
指定的输出 Ⓓ1 在顺序程序中变为 OFF。

(2) MTR 指令的执行间隔必须比输入和输出模块响应时间的总和长。
如果设定的间隔比以上表示的值短，则不能正确读取输入。
如果顺序程序中的扫描时间太短，则选择恒定扫描并将其扫描时间改变得比响应时间的总和长。

7 应用指令

指令	含义	参考章节
逻辑运算指令	逻辑和、逻辑乘等逻辑运算	7.1 节
循环移动指令	指定数据的循环移动	7.2 节
移位指令	指定数据的移位	7.3 节
位处理指令	置位和复位位数据、位提取	7.4 节
数据处理指令	数据查找、解码、编码等数据处理	7.5 节
结构化程序指令	重复运算，子程序调用和梯形图单元中的变址修改	7.6 节
数据表运算指令	读 / 写 FIFO 表	7.7 节
缓冲存储器访问指令	读 / 写特殊功能模块的缓冲存储器	7.8 节
显示指令	将字符代码输出到外部元件，并显示在显示单元上	7.9 节
调试与故障诊断指令	检查、状态锁存、采样追踪和程序跟踪	7.10 节
字符串处理指令	字符串（ASCII 码数据）处理	7.11 节
特殊功能指令	BCD 型实数处理，浮动小数点类型实数处理	7.12 节
数据控制指令	基于输入数据范围检测的输出值控制	7.13 节
文件寄存器切换指令	设置文件寄存器切换块号码	7.14 节
时钟指令	读 / 写时钟数据	7.15 节
外围设备指令	显示外围设备消息、键输入	7.16 节
程序控制指令	切换程序执行条件的指令	7.17 节
其它指令	与以上各类不符的指令，比如看门狗计时器复位指令和计时时钟指令	7.18 节

7.1 逻辑运算指令

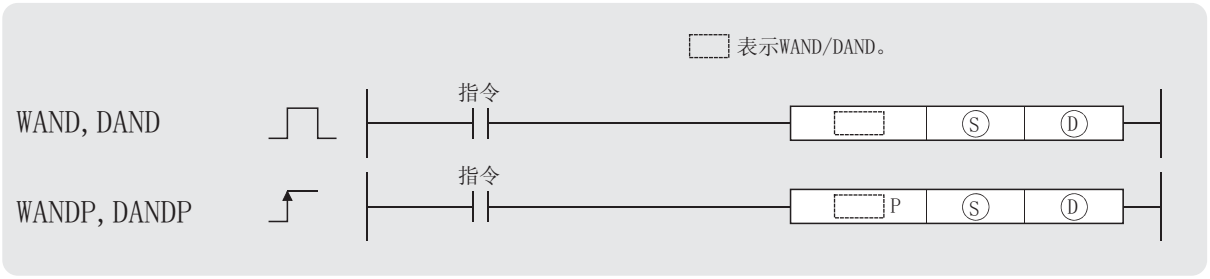
(1) 逻辑运算指令包括逻辑和及逻辑乘等，它们在 1 位单位中进行处理，讨论如下。

分类	处理详细信息	运算公式	示例		
			A	B	Y
逻辑乘 (AND)	只有当输入 A 和输入 B 都为 1 时才变为 1 否则，为 0	$Y=A \cdot B$	0	0	0
			0	1	0
			1	0	0
			1	1	1
逻辑和 (OR)	只有当输入 A 和输入 B 都为 0 时才变为 0 否则，为 1	$Y=A + B$	0	0	0
			0	1	1
			1	0	1
			1	1	1
异或 OR (XOR)	如果输入 A 和输入 B 相等，则变为 0，否则为 1	$Y=\overline{A} \cdot B + A \cdot \overline{B}$	0	0	0
			0	1	1
			1	0	1
			1	1	0
异或非 (XNR)	如果输入 A 和输入 B 相等，则变为 1，否则为 0	$Y=(\overline{A} + B)(A + \overline{B})$	0	0	1
			0	1	0
			1	0	0
			1	1	1

7.1.1 16 位和 32 位数据的逻辑乘 (WAND(P)、DAND(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（Ⓓ ∧ Ⓢ → Ⓓ、（Ⓓ +1、Ⓓ）∧（Ⓢ +1、Ⓢ）→（Ⓓ +1、Ⓓ））



Ⓢ：进行逻辑乘的数据或者存储这些数据的软元件的起始号 (BIN16/32 位)。

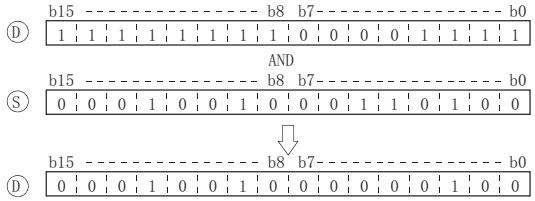
Ⓓ：存储逻辑乘运算结果的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

WAND

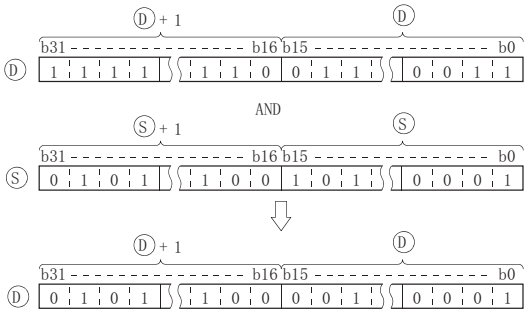
(1) 对 Ⓓ 指定的软元件中的 16 位数据的每一位与 Ⓢ 指定的软元件中的 16 位数据的对应位进行逻辑乘运算，并将结果存储到 Ⓓ 指定的软元件中。



(2) 指定位软元件时，高于位数指定点数的位软元件在运算中被作为 0。(参见程序示例 (2))

DAND

(1) 对 (S1) 指定的软元件中的 32 位数据的每一位与 (S2) 指定的软元件中的 32 位数据的对应位进行逻辑乘运算，并将结果存储到 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位数的数字位在该运算中被处理为 0。(参见程序示例 (2))

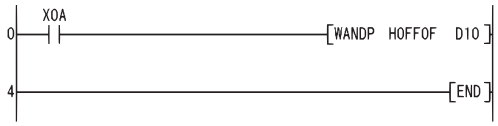
运算错误

(1) 不存在与 WAND(P) 或 DAND(P) 指令相关的运行错误。

程序示例

(1) 当 XA 为 ON 时，以下程序将 D10 中 4 位 BCD 值的十位数 (右起第二位) 上的数字屏蔽为 0。

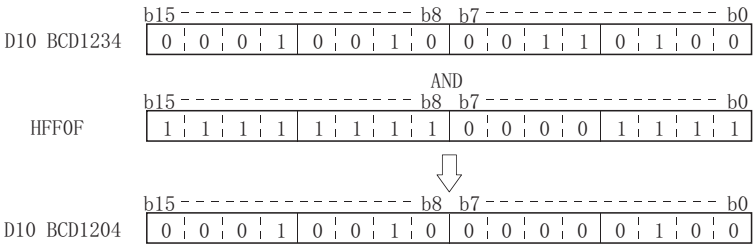
[梯形图模式]



[列表模式]

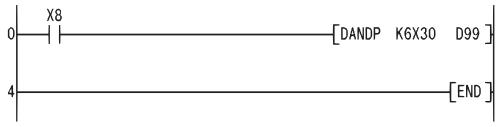
步	指令	软元件
0	LD	XOA
1	WANDP	HOFFOF D10
4	END	

[动作]



(2) 当 X8 为 ON 时，以下程序对 D99 到 D100 中的数据和 X30 到 X47 中的 24 位数据执行逻辑乘运算，并将结果存储在 D99 到 D100 中。

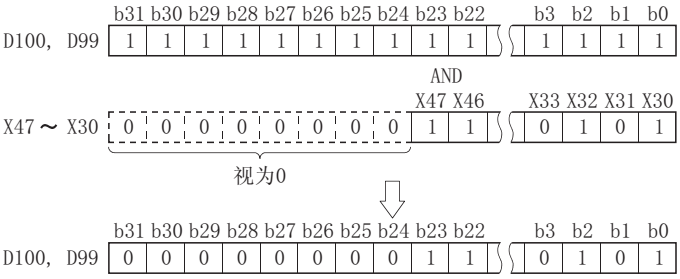
[梯形图模式]



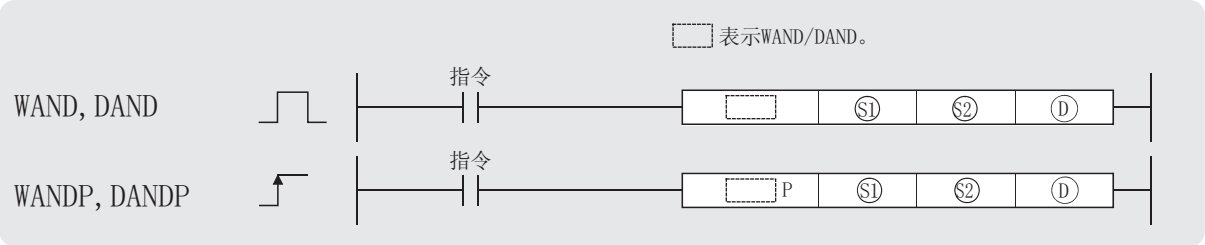
[列表模式]

步	指令	软元件	
0	LD	X8	
1	DANDP	K6X30	D99
4	END		

[动作]



2 当设置了三个数据时（(S1) ∧ (S2) → (D)、((S1)+1、(S1)) ∧ ((S2)+1、(S2)) → ((D)+1、(D))）



(S1)、(S2)：进行逻辑乘的数据，或者存储这些数据的软元件的起始号（BIN16/32 位）。

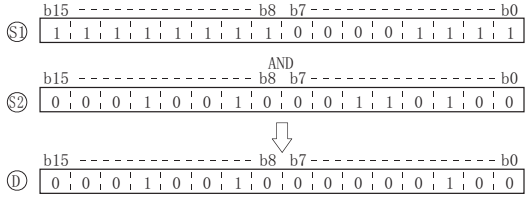
(D)　：存储逻辑乘运算结果的软元件的起始号（BIN16/32 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○							○	---
(S2)	○							○	---
(D)	○							---	---

功 能

WAND

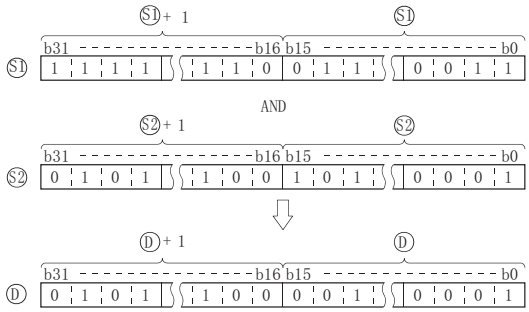
(1) 对 (S1) 指定的软元件中的 16 位数据的每一位与 (S2) 指定的软元件中的 16 位数据的对应位进行逻辑乘运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位数的数字位在该运算中被处理为 0。（参见程序示例 (1) 和 (2)）

DAND

(1) 对 (S1) 指定的软元件中的 32 位数据的每一位与 (S2) 指定的软元件中的 32 位数据的对应位进行逻辑乘运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位数的数字位在该运算中被处理为 0。（参见程序示例 (3)）

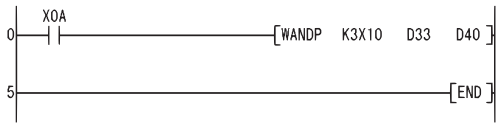
运算错误

(1) 不存在与 WAND(P) 或 DAND(P) 指令相关的运行错误。

程序示例

(1) 当 XA 为 ON 时，以下程序对 X10 到 X1B 中的数据和 D33 中的数据执行逻辑乘运算，并将结果存储在 D40 中。

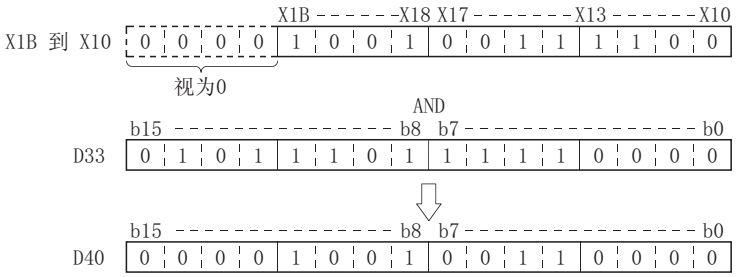
[梯形图模式]



[列表模式]

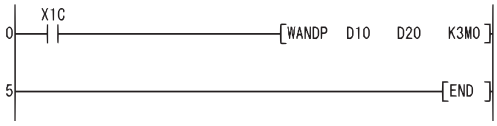
步	指令	软元件		
0	LD	X0A		
1	WANDP	K3X10	D33	D40
5	END			

[动作]



(2) 当 X1C 为 ON 时，以下程序对 D10 中的数据和 D20 中的数据执行逻辑乘运算，并将结果存储在 M0 到 M11 中。

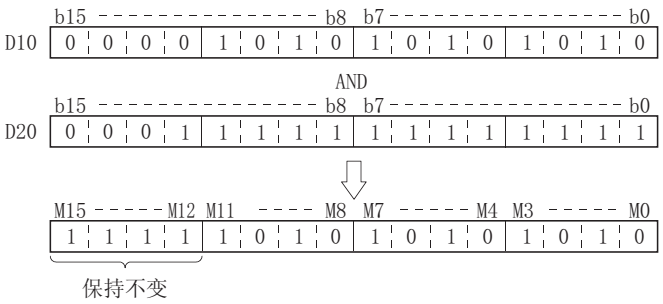
[梯形图模式]



[列表模式]

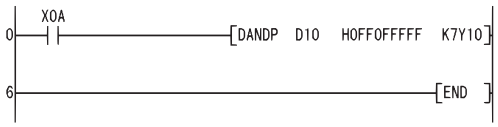
步	指令	软元件		
0	LD	X1C		
1	WANDP	D10	D20	K3M0
5	END			

[动作]



(3) 当 XA 为 ON 时，以下程序将 D10、D11 中 8 位 BCD 值十万位（右起第六位）上的数字屏蔽为 0，并将结果输出到 Y10 到 Y2B。

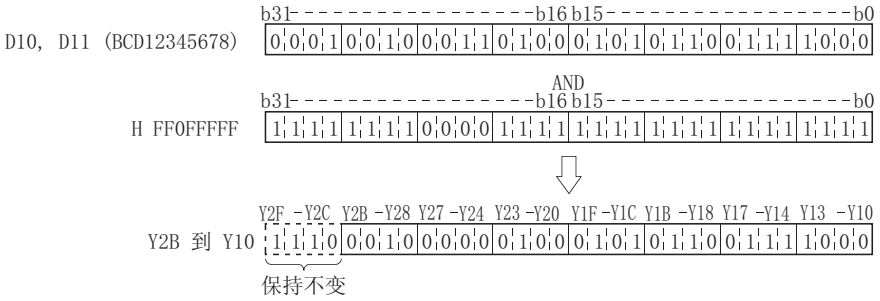
[梯形图模式]



[列表模式]

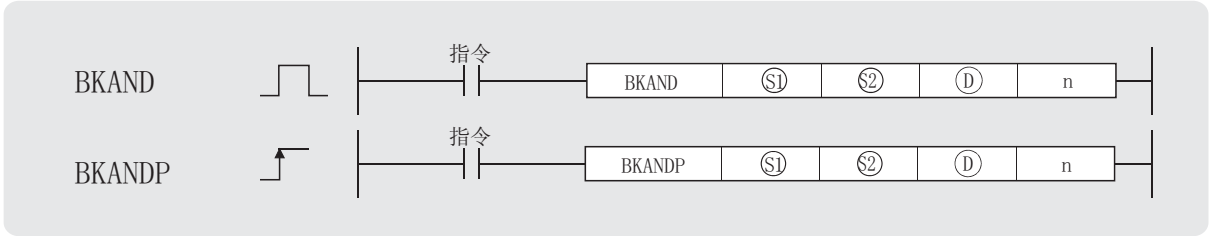
步	指令	软元件
0	LD	XOA
1	DANDP	D10 H0FF0FFFFF K7Y10
6	END	

[动作]



7.1.2 块逻辑乘 (BKAND(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



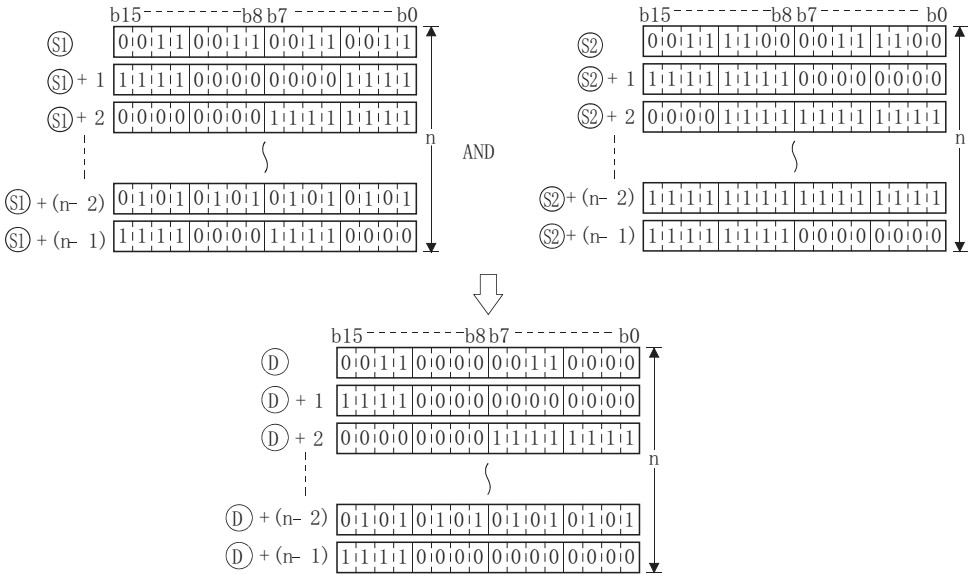
(S1)*1　：存储逻辑乘运算的数据的软元件的起始号 (BIN16 位)。
(S2)*1　：逻辑运算数据或者存储逻辑运算数据的软元件的起始号 (BIN16 位)。
(D)*1　：存储运算结果的软元件的起始号 (BIN16 位)。
n　：运算数据的块数。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)*1	---	○		---				---	---
(S2)*1	---	○		---				○	---
(D)*1	---	○		---				---	---
n	○	○		○				○	---

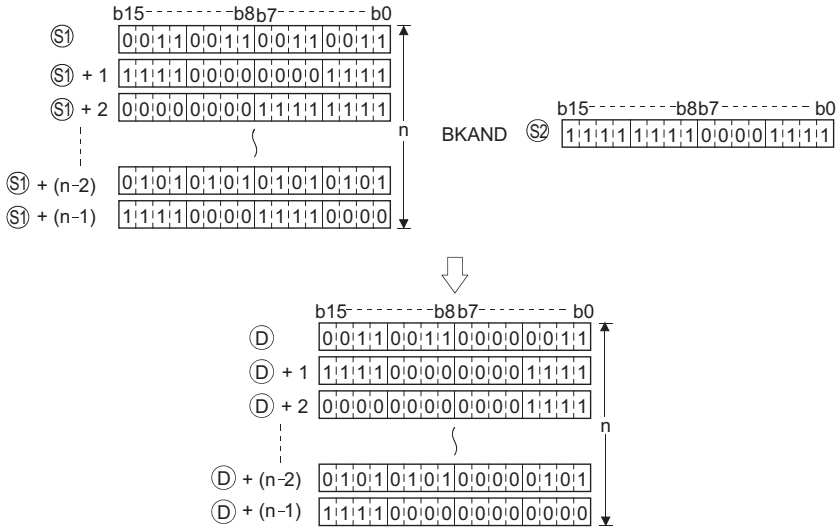
*1：可以给 (S1) 和 (D) 或者 (S2) 和 (D) 指定相同的软元件号。

功 能

(1) 对从 (S1) 指定的软元件开始的 n 点中的数据和从 (S2) 指定的软元件开始的 n 点中的数据执行逻辑乘运算，并将结果存储在 (D) 指定的软元件中。



(2) ⓢ2 可以指定的常量的范围是 -32768 到 32767(BIN 16 位)。



运算错误

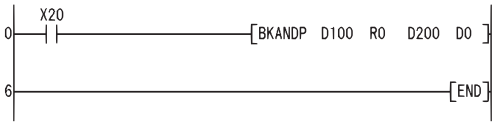
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 ⓢ1、ⓢ2 或 ⓓ 软元件开始的 n 位范围超出了该软元件的范围。
  (出错代码：4101)
- 从 ⓢ1 指定的软元件开始的 n 点软元件范围与从 ⓓ 指定的软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ1 和 ⓓ 的软元件相同时除外。) (出错代码：4101)
- 从 ⓢ2 指定的软元件开始的 n 点软元件范围与从 ⓓ 指定的软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ2 和 ⓓ 的软元件相同时除外。) (出错代码：4101)

程序示例

(1) 当 X20 变为 ON 时，以下程序对与存储在 D100 和 D0 中的值相对应的数据点数和与存储在 R0 和 D0 中的值相对应的数据点数执行逻辑乘运算，并将结果存储在从 D200 开始的软元件中。

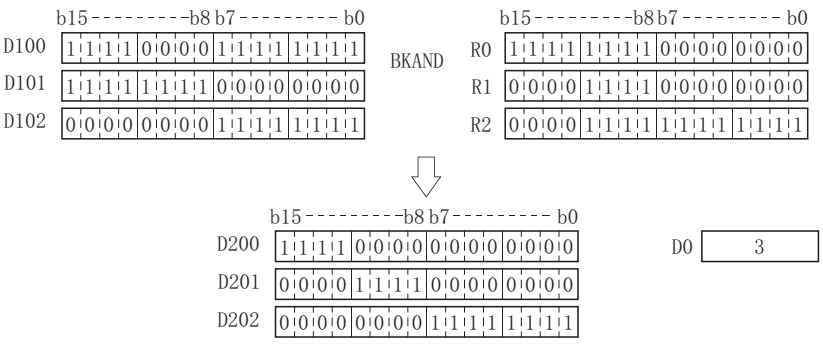
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X20			
1	BKANDP	D100	R0	D200	D0
6	END				

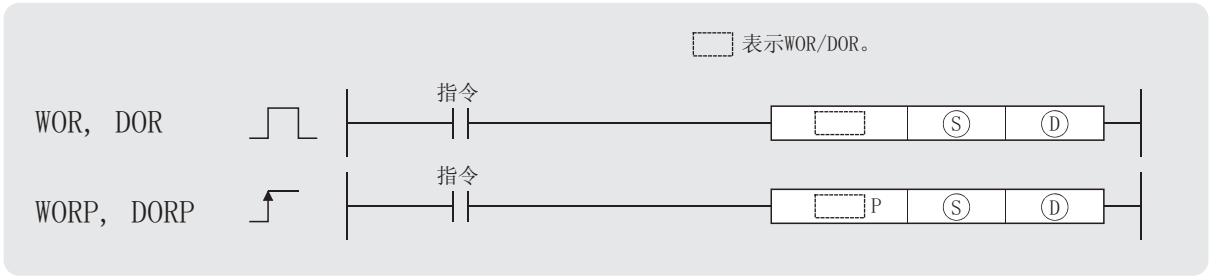
[动作]



7.1.3 16 位和 32 位数据的逻辑和 (WOR(P)、DOR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时 (Ⓓ ∨ Ⓢ → Ⓓ、(Ⓓ+1、Ⓓ) ∨ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ))



Ⓢ: 执行逻辑和的数据或者存储该数据的软元件的起始号 (BIN16/32 位)。

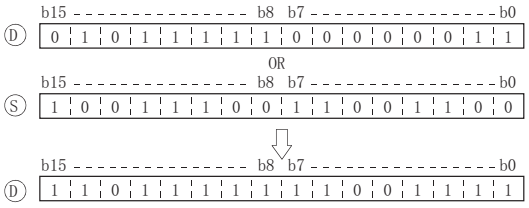
Ⓓ: 存储逻辑和结果的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

WOR

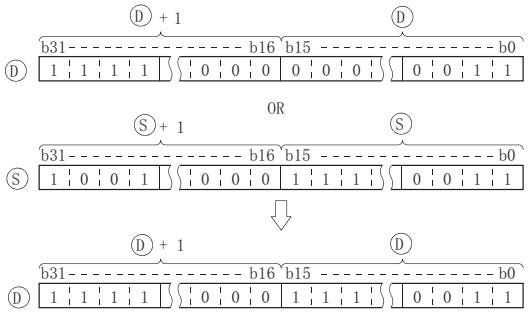
(1) 对 Ⓓ 指定的软元件中的 16 位数据的每一位与 Ⓢ 指定的软元件中的 16 位数据的对应位进行逻辑和运算，并将结果存储在 Ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

DOR

(1) 对 Ⓓ 指定的软元件中的 32 位数据的每一位与 Ⓢ 指定的软元件中的 32 位数据的对应位进行逻辑和运算，并将结果存储在 Ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

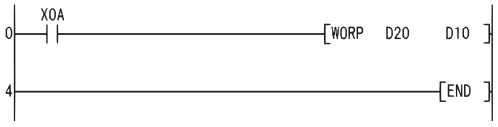
运算错误

(1) 不存在与 WOR(P) 或 DOR(P) 指令相关的运行错误。

程序示例

(1) 当 XA 变为 ON 时，以下程序对 D10 和 D20 中的数据执行逻辑和运算，并将结果存储在 D10 中。

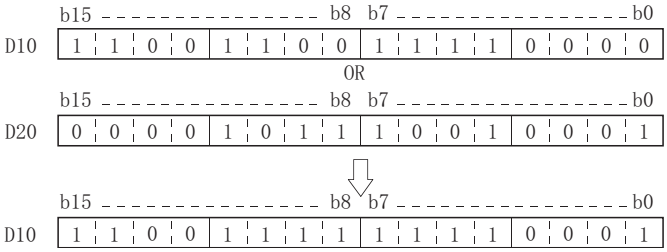
[梯形图模式]



[列表模式]

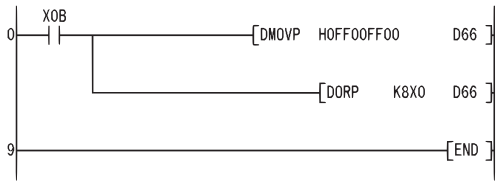
步	指令	软元件	
0	LD	XOA	
1	WORP	D20	D10
4	END		

[动作]



(2) 当 XB 变为 ON 时，以下程序对 X0 到 X1F 中的 32 位数据和 16 进制值 FF00FF00H 执行逻辑和运算，并将结果存储在 D66 和 D67 中。

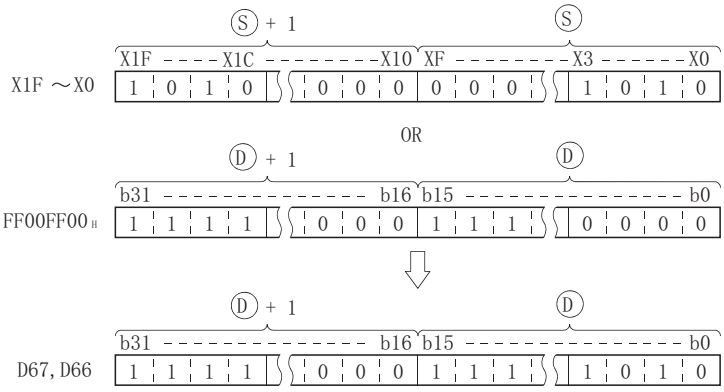
[梯形图模式]



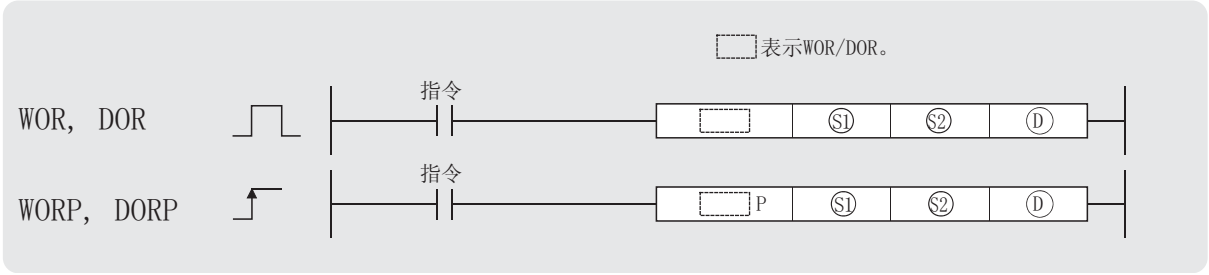
[列表模式]

步	指令	软元件	
0	LD	XOB	
1	DMOVP	HOFF00FF00	D66
4	DORP	K8X0	D66
9	END		

[动作]



2 当设置了三个数据时（Ⓢ1 ∨ Ⓢ2 → Ⓓ、（Ⓢ1+1、Ⓢ1）∨（Ⓢ2+1、Ⓢ2）→（Ⓓ+1、Ⓓ））



Ⓢ1、Ⓢ2：执行逻辑和的数据或者存储数据的软元件的起始号（BIN16/32 位）。

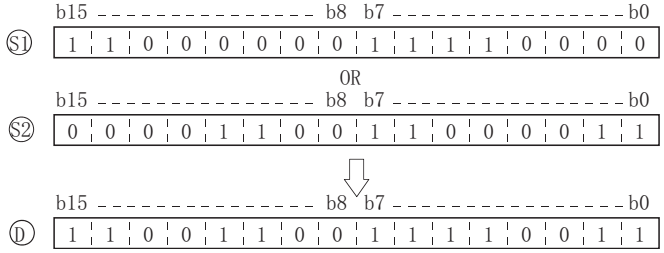
Ⓓ　　：存储逻辑和结果的软元件的起始号（BIN16/32 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ1	○							○	---
Ⓢ2	○							○	---
Ⓓ	○							---	---

功 能

WOR

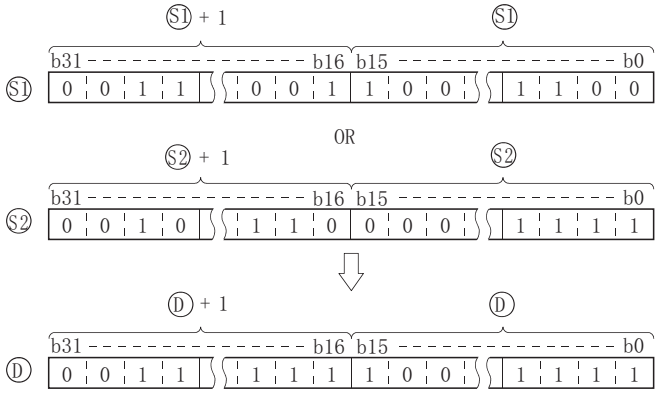
(1) 对 Ⓢ1 指定的软元件中的 16 位数据的每一位与 Ⓢ2 指定的软元件中的 16 位数据的对应位进行逻辑和运算，并将结果存储在 Ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。（参见程序示例 (1)）

DOR

(1) 对 (S1) 指定的软元件中的 32 位数据的每一位与 (S2) 指定的软元件中的 32 位数据的对应位进行逻辑和运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。(参见程序示例 (2))

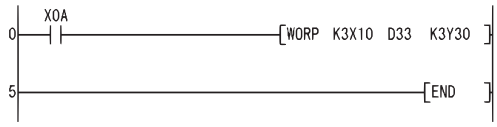
运算错误

(1) 不存在与 WOR(P) 或 DOR(P) 指令相关的运行错误。

程序示例

(1) 当 XA 为 ON 时，以下程序对 X10 到 X1B 中的数据和 D33 中的数据执行逻辑和运算，并将结果存储在 Y30 到 Y38 中。

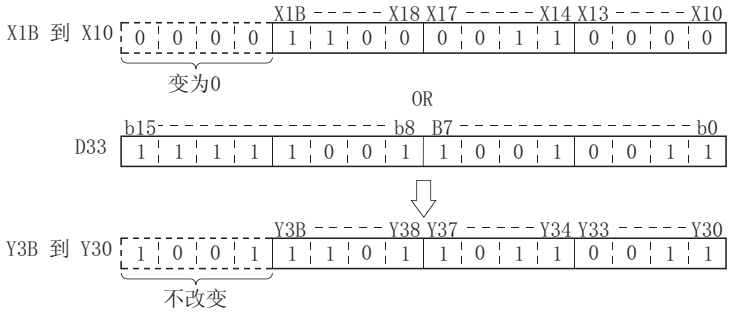
[梯形图模式]



[列表模式]

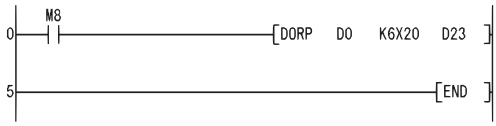
步	指令	软元件		
0	LD	XOA		
1	WORP	K3X10	D33	K3Y30
5	END			

[动作]



(2) 当 M8 为 ON 时，以下程序对 D0、D1 中的 32 位数据和 X20 到 X37 中的 24 位数据执行逻辑和运算，并将结果存储在 D23 和 D24 中。

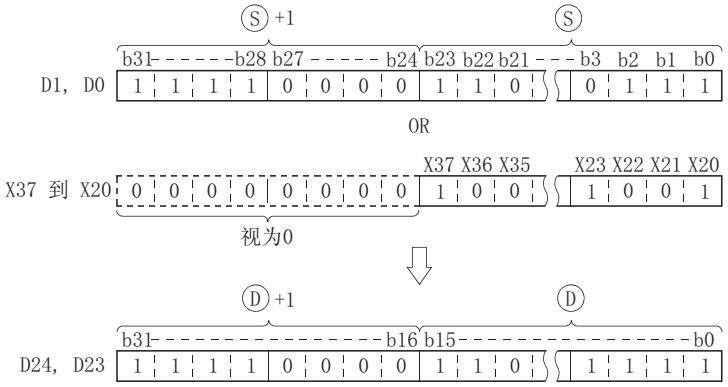
[梯形图模式]



[列表模式]

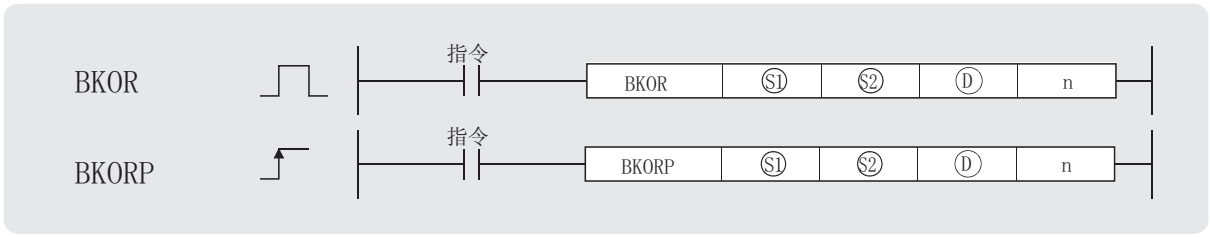
步	指令	软元件		
0	LD	M8		
1	DORP	D0	K6X20	D23
5	END			

[动作]



7.1.4 块逻辑和运算 (BKOR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)*1：存储要用于逻辑和运算的数据的软元件的起始号 (BIN 16 位)

(S2)*1：逻辑运算数据的第一个数或者存储逻辑运算数据的软元件的起始号 (BIN 16 位)

(D)*1：存储运算结果的软元件的起始号 (BIN 16 位)

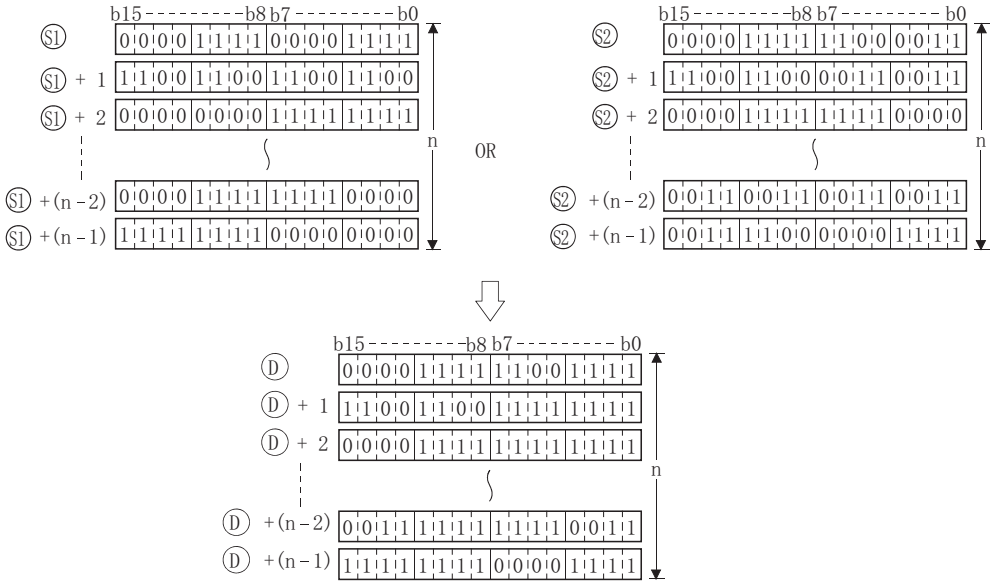
n　：运算数据个数 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)*1	---	○		---				---	---
(S2)*1	---	○		---				○	---
(D)*1	---	○		---				---	---
n	○	○		○				○	---

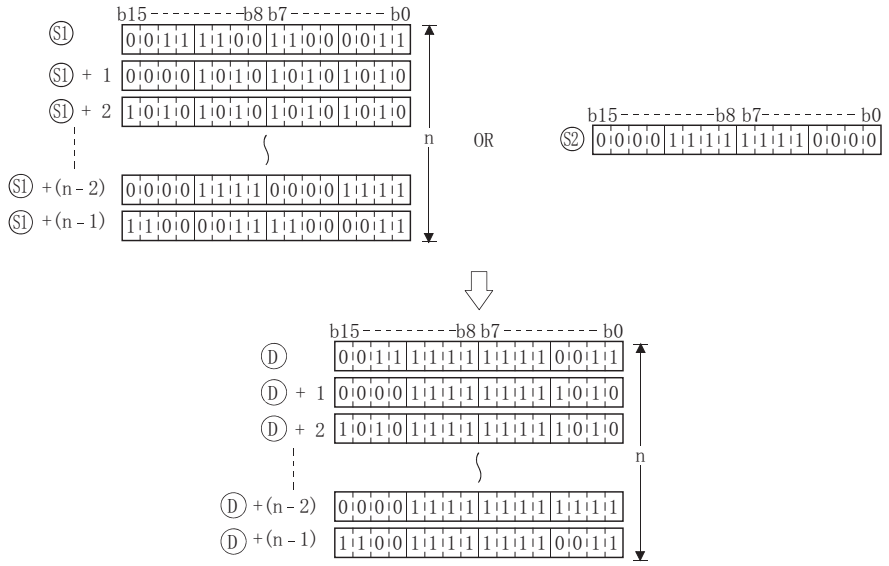
*1：可以给 (S1) 和 (D) 或者 (S2) 和 (D) 指定相同的软元件号。

功 能

(1) 对从 (S1) 指定的软元件开始的 n 点中的数据和从 (S2) 指定的软元件开始的 n 点中的数据执行逻辑和运算，并将结果存储在 (D) 指定的软元件中。



(2) ⓢ2 可以指定的常量的范围是 -32768 到 32767(BIN 16 位)。



运算错误

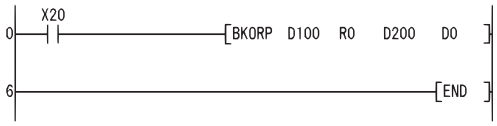
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 ⓢ1、ⓢ2 或 ⓓ 软元件开始的 n 位的范围超出了相应软元件的范围。
  (出错代码：4101)
- 从 ⓢ1 指定的软元件开始的 n 点软元件范围与从 ⓓ 指定的软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ1 和 ⓓ 的软元件相同时除外。) (出错代码：4101)
- 从 ⓢ2 指定的软元件开始的 n 点软元件范围与从 ⓓ 指定的软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ2 和 ⓓ 的软元件相同时除外。) (出错代码：4101)

程序示例

(1) 当 X20 变为 ON 时，以下程序对存储在 D100 到 D102 中的数据和存储在 R0 到 R2 中的数据执行逻辑和运算，然后，将结果存储在从 D200 开始的区域中。

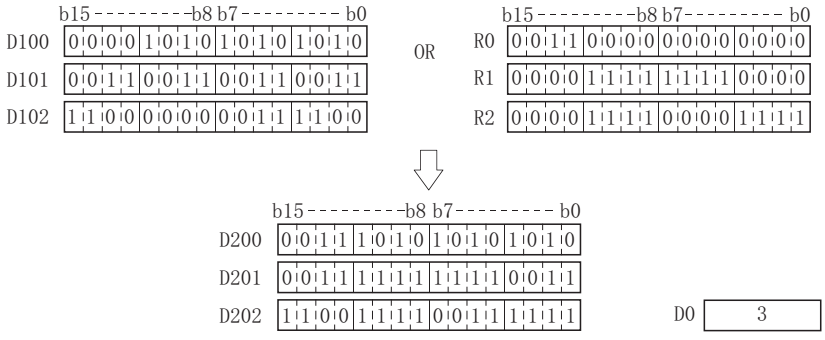
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X20
1	BKORP	D100 R0 D200 D0
6	END	

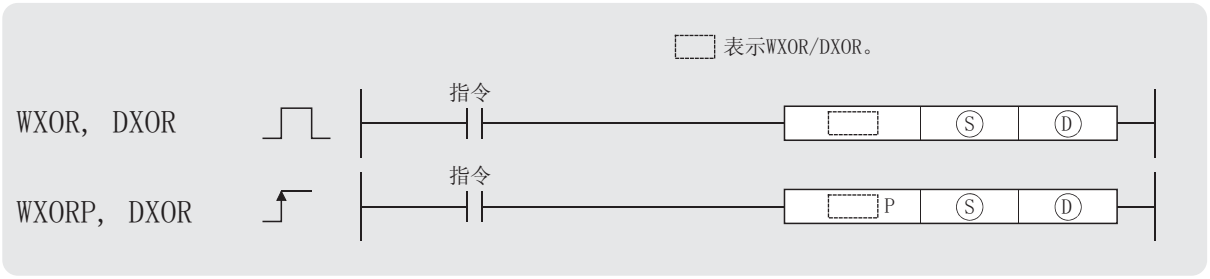
[动作]



7.1.5 16 位和 32 位数据异或运算 (WXOR(P)、DXOR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（Ⓓ ∀ Ⓢ → Ⓓ、(Ⓓ+1、Ⓓ) ∀ (Ⓢ+1、Ⓢ) → (Ⓓ+1、Ⓓ)）



Ⓢ：用于执行异或运算的数据，或者存储这些数据的软元件的起始号 (BIN 16/32 位)

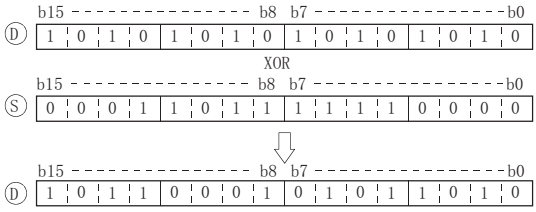
Ⓓ：存储被执行 “异或” 运算的数据的软元件的起始号 (BIN 16/32 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

WXOR

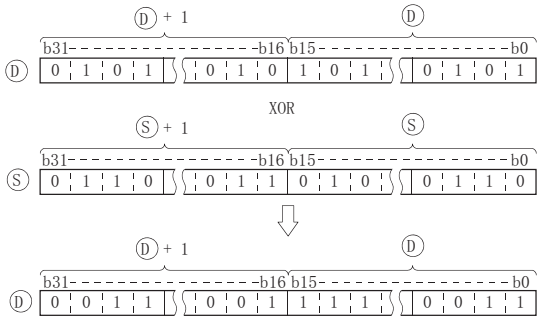
(1) 对 Ⓓ 指定的软元件中的 16 位数据的每一位与 Ⓢ 指定的软元件中的 16 位数据的对应位进行异或运算，并将结果存储在 Ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

DXOR

(1) 对 ⓓ 指定的软元件中的 32 位数据的每一位与 ⓢ 指定的软元件中的 32 位数据的对应位进行异或运算，并将结果存储在 ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

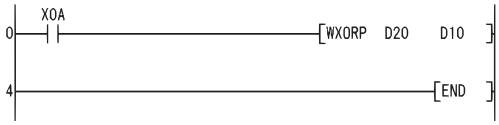
运算错误

(1) 如果没有执行变址修改，则不存在与 WXOR(P) 或 DXOR(P) 指令相关的运行错误。

程序示例

(1) 当 XA 为 ON 时，以下程序对 D10 和 D20 中的数据执行异或运算，并将结果存储在 D10 中。

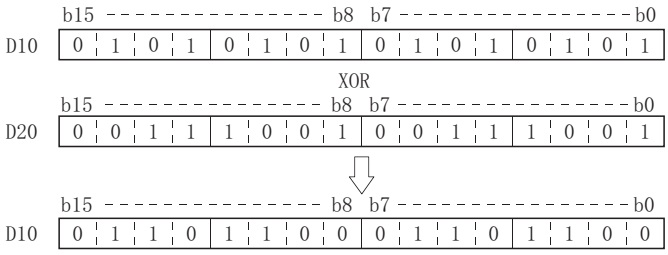
[梯形图模式]



[列表模式]

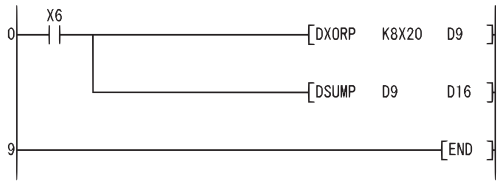
步	指令	软元件	
0	LD	XOA	
1	WXORP	D20	D10
4	END		

[动作]



(2) 当 X6 为 ON 时，以下程序将 X20 到 X3F 中的 32 位数据的位模式与 D9 到 D10 中的数据的位模式进行比较，并将存在差异的位的数目存储在 D16 中。

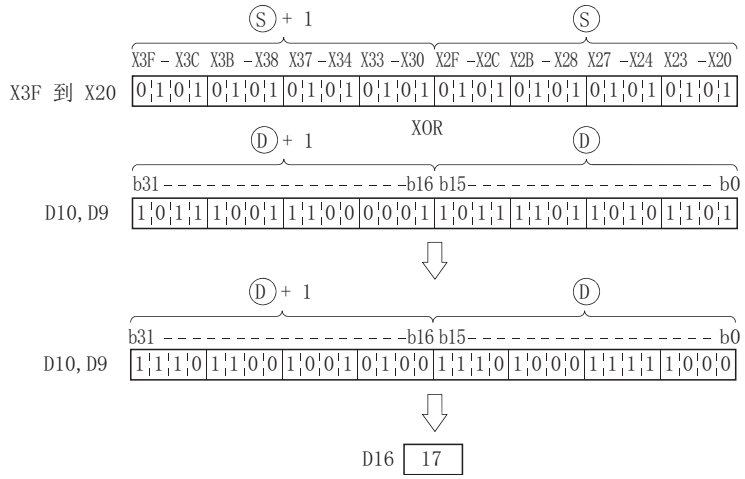
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X6	
1	DXORP	K8X20	D9
6	DSUMP	D9	D16
9	END		

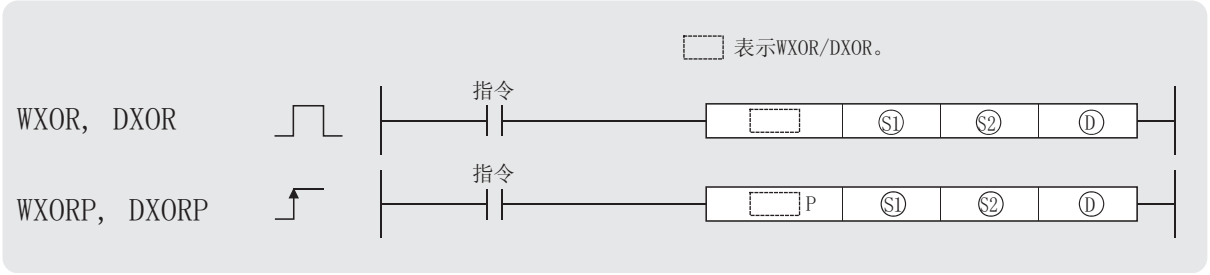
[动作]



备 注

关于 DSUMP 指令的更多内容，请参见 7.5.2 项。

2 当设置了三个数据时（(S1) ∨ (S2) → (D) （(S1)+1、(S1)）∨ （(S2)+1、(S2)）→（(D)+1、(D)））



(S1)、(S2)：用于执行异或运算的数据，或者存储这些数据的软元件的起始号（BIN16/32 位）。

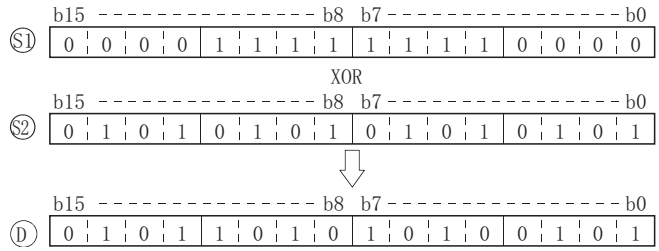
(D)　　：存储异或运算结果的软元件的起始号（BIN16/32 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○							○	---
(S2)	○							○	---
(D)	○							---	---

功 能

WXOR

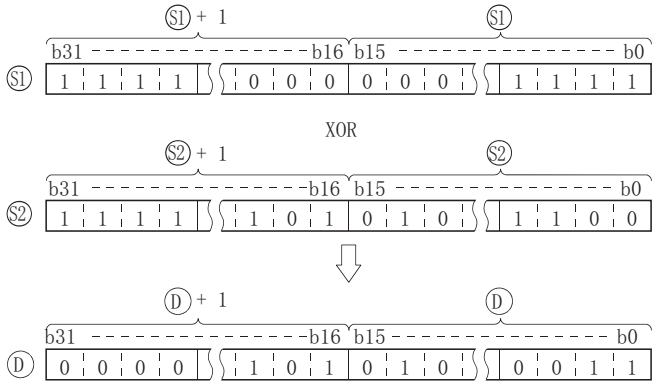
(1) 对 (S1) 指定的软元件中的 16 位数据的每一位与 (S2) 指定的软元件中的 16 位数据的对应位进行异或运算，并将结果存储在 (D) 指定的软元件中。



(2) 对于位软元件而言，高于指定位的点数的位软元件在运算中被作为 0。（参见程序示例 (1)）

DXOR

(1) 对 (S1) 指定的软元件中的 32 位数据的每一位与 (S2) 指定的软元件中的 32 位数据的对应位进行异或运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

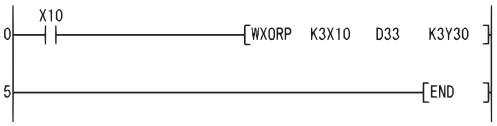
运算错误

(1) 不存在与 WXOR(P) 或 DXOR(P) 指令相关的运行错误。

程序示例

(1) 当 X10 为 ON 时，以下程序对 X10 到 X1B 中的数据和 D33 中的数据执行异或运算，并将结果输出到 Y30 到 Y3B 中。

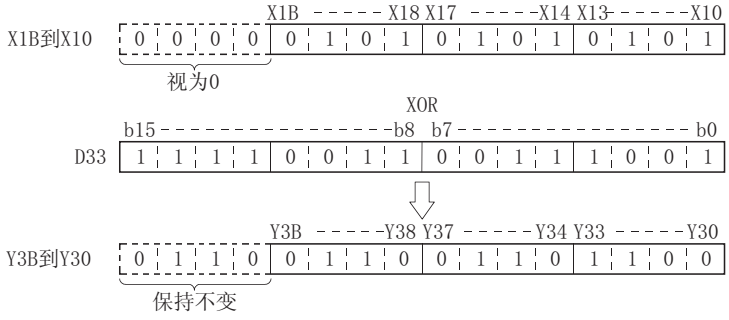
[梯形图模式]



[列表模式]

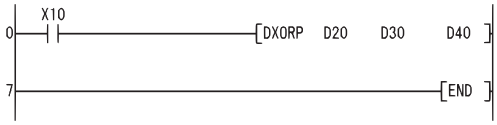
步	指令	软元件		
0	LD	X10		
1	WXORP	K3X10	D33	K3Y30
5	END			

[动作]



(2) 当 X10 为 ON 时，以下程序对 D20、D21 中的数据和 D30、D31 中的数据执行异或运算，并将结果存储在 D40 和 D41 中。

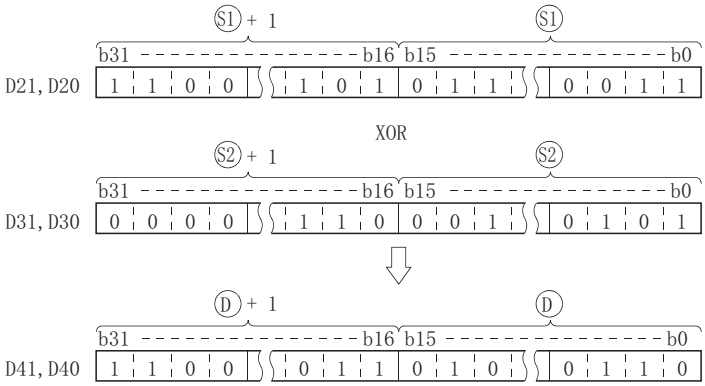
[梯形图模式]



[列表模式]

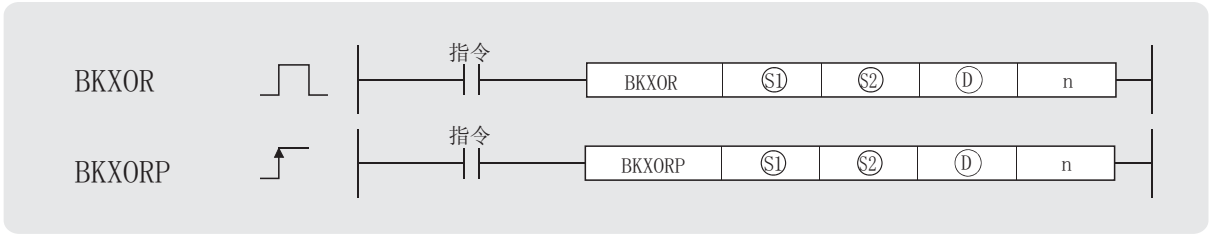
步	指令	软元件		
0	LD	X10		
1	DXORP	D20	D30	D40
7	END			

[动作]



7.1.6 块异或运算 (BKXOR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)*1 ： 存储进行逻辑运算的数据的软元件的起始号 (BIN16 位)。

(S2)*1 ： 逻辑运算数据或者存储逻辑运算数据的软元件的起始号 (BIN16 位)。

(D)*1 ： 存储运算结果的软元件的起始号 (BIN16 位)。

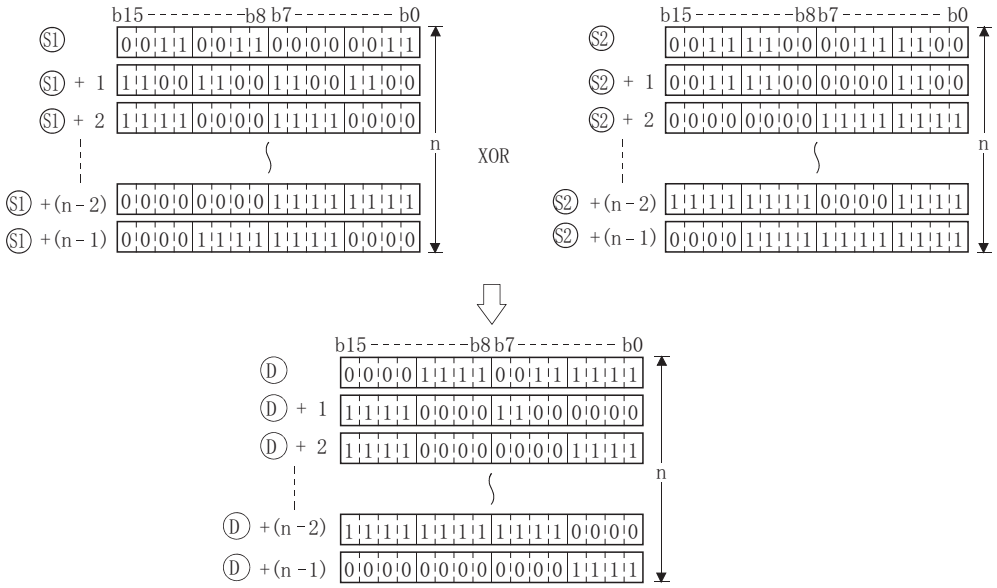
n ： 运算数据块数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)*1	---	○		---				---	---
(S2)*1	---	○		---				○	---
(D)*1	---	○		---				---	---
n	○	○		○				○	---

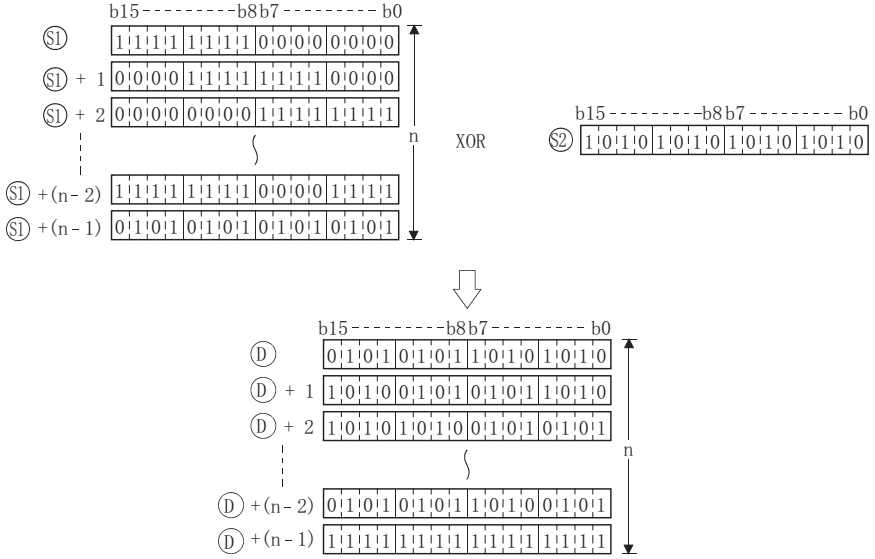
*1: 可以给 (S1) 和 (D) 或者 (S2) 和 (D) 指定相同的软元件号。

功 能

(1) 对从 (S1) 指定的软元件开始的 n 点中的数据和从 (S2) 指定的软元件开始的 n 点中的数据执行异或运算，并将结果存储在 (D) 指定的软元件中。



(2) ⓢ2 可以指定的常量的范围是 -32768 到 32767(BIN 16 位)



运算错误

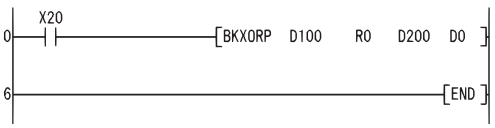
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 ⓢ1、ⓢ2 或 ⓓ 软元件开始的 n 位范围超出了相应软元件的范围。
  (出错代码 : 4101)
- 从 ⓢ1 软元件开始的 n 点软元件范围与从 ⓓ 软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ1 和 ⓓ 的软元件相同时除外。) (出错代码 : 4101)
- 从 ⓢ2 指定的软元件开始的 n 点软元件范围与从 ⓓ 指定的软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ2 和 ⓓ 的软元件相同时除外。) (出错代码 : 4101)

程序示例

(1) 当 X20 变为 ON 时，以下程序对存储在 D100 到 D102 和存储在 R0 到 R2 的数据执行异或运算，然后将异或运算的结果存储在从 D200 开始的区域。

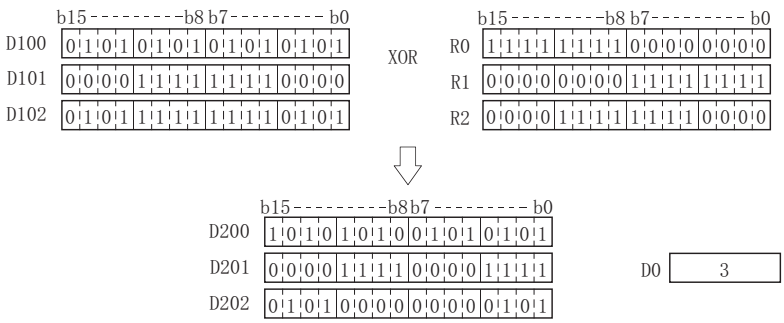
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X20			
1	BKXORP	D100	R0	D200	D0
6	END				

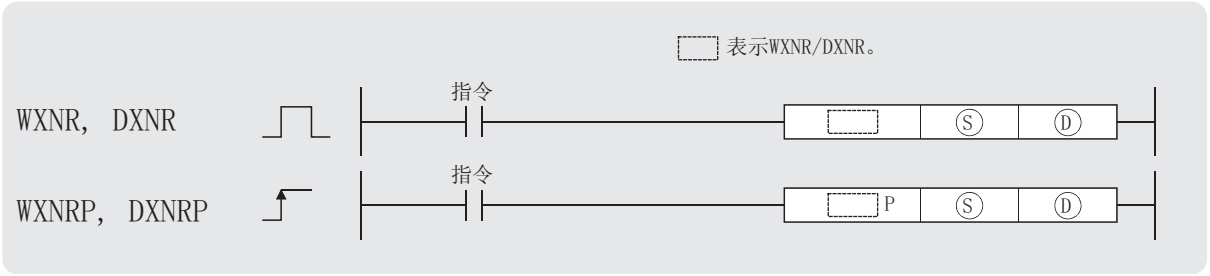
[动作]



7.1.7 16 位和 32 位异或非运算 (WXNR(P)、DXNR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当设置了两个数据时（$\overline{(D) \veebar (S)} \rightarrow (D)$、$\overline{(D)+1、(D)) \veebar ((S)+1、(S))} \rightarrow ((D)+1、(D))$）



(S)：执行异或非运算的数据，或者存储这些数据的软元件的起始号 (BIN16/32 位)。

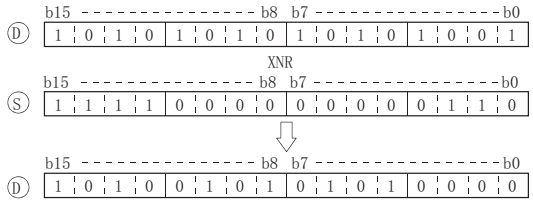
(D)：存储异或非运算结果的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S)	○							○	---
(D)	○							---	---

功 能

WXNR

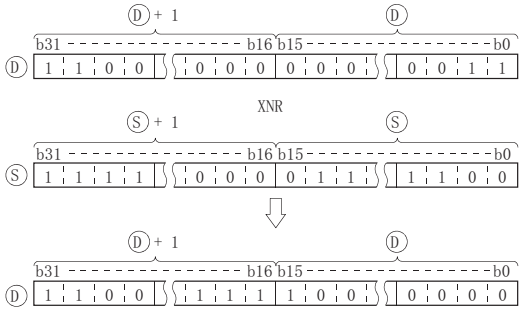
(1) 对 (D) 指定的软元件中的 16 位数据的每一位与 (S) 指定的软元件中的 16 位数据的对应位进行异或非运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

DXNR

(1) 对 ⓓ 指定的软元件中的 32 位数据的每一位与 ⓢ 指定的软元件中的 32 位数据的对应位进行异或非运算，并将结果存储在 ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

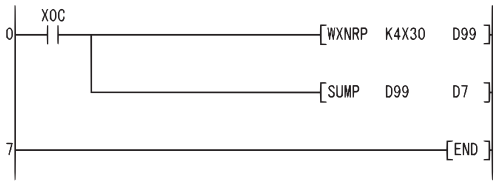
运算错误

(1) 不存在与 WXNR(P) 或 DXNR(P) 指令相关的运行错误。

程序示例

(1) 当 XC 变为 ON 时，以下程序将 X30 到 X3F 中的 16 位数据的位模式与 D99 中的数据的位模式进行比较，并将相同模式的数目存储在 D7 中。

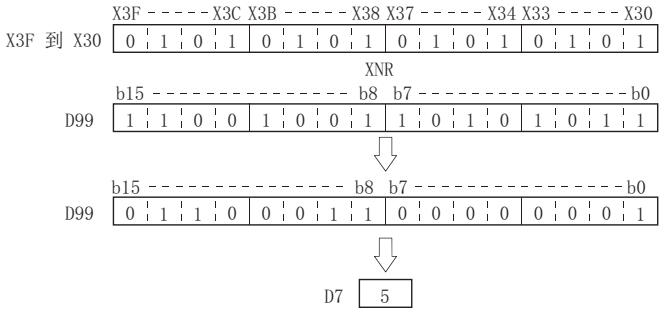
[梯形图模式]



[列表模式]

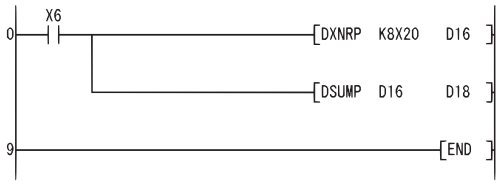
步	指令	软元件	
0	LD	X0C	
1	WXNRP	K4X30	D99
4	SUMP	D99	D7
7	END		

[动作]



(2) 当 X6 为 ON 时，以下程序将 X20 到 X3F 中的 32 位数据的位模式与 D16、D17 中的数据的位模式进行比较，并将相同位模式的数目存储在 D18 中。

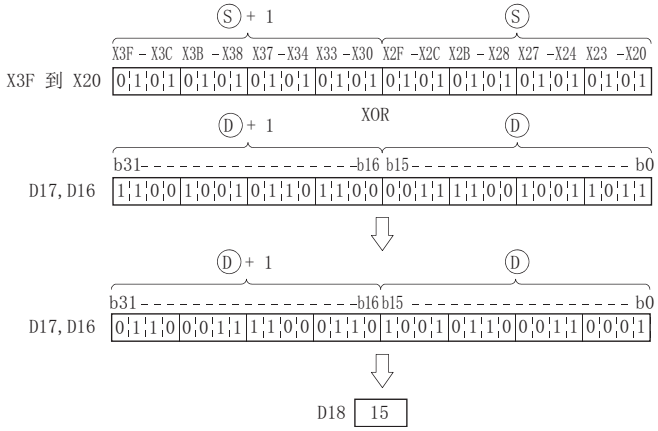
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X6	
1	DXNRP	K8X20	D16
6	DSUMP	D16	D18
9	END		

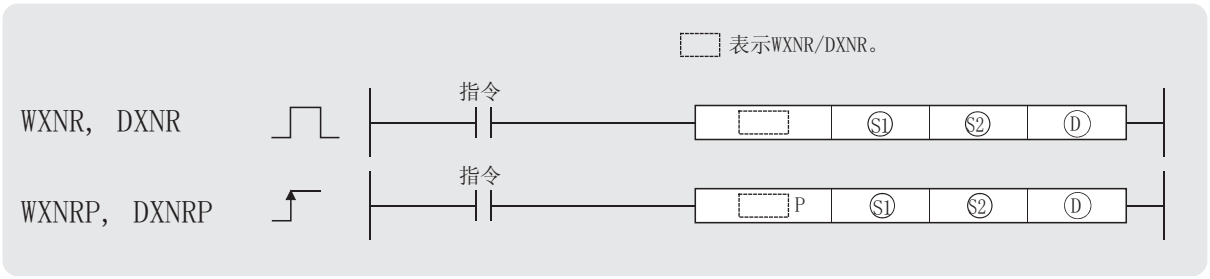
[动作]



备 注

有关 SUMP/DSUM 指令的更多信息，请参阅 7.5.2 项。

2 当设置了三个数据时（$\overline{(S1) \vee (S2)} \rightarrow (D)$、$\overline{(S1)+1、(S1)) \vee ((S2)+1、(S2))} \rightarrow ((D)+1、(D))$）



(S1)、(S2)：执行异或非运算的数据，或者存储这些数据的软元件的起始号（BIN16/32 位）。

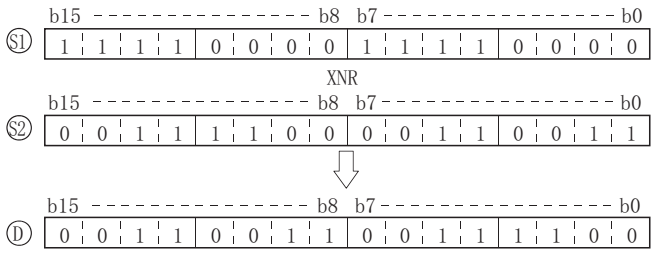
(D)　：存储异或非运算结果的软元件的起始号（BIN16/32 位）。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○							○	---
(S2)	○							○	---
(D)	○							---	---

功 能

WXNR

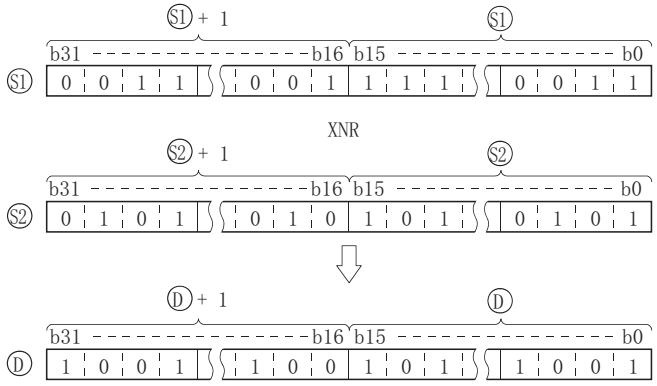
(1) 对 (S1) 指定的软元件中的 16 位数据的每一位与 (S2) 指定的软元件中的 16 位数据的对应位进行异或非运算，并将结果存储在 (D) 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

DXNR

(1) 对 ⓢ1 指定的软元件中的 32 位数据的每一位与 ⓢ2 指定的软元件中的 32 位数据的对应位进行异或非运算，并将结果存储在 ⓓ 指定的软元件中。



(2) 对位软元件而言，高于指定位的位数在该运算中被处理为 0。

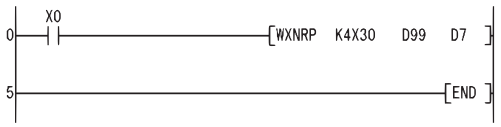
运算错误

(1) 不存在与 WXNR(P) 或 DXNR(P) 指令相关的运行错误。

程序示例

(1) 当 X0 为 ON 时，以下程序对 X30 到 X3F 中的 16 位数据和 D99 中的数据执行异或非运算，并将结果存储在 D7 中。

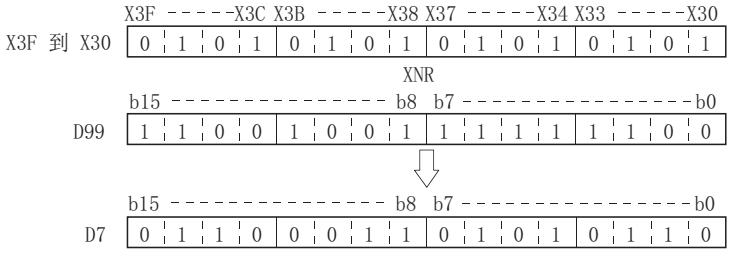
[梯形图模式]



[列表模式]

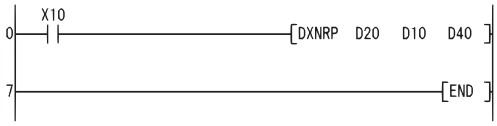
步	指令	软元件		
0	LD	X0		
1	WXNRP	K4X30	D99	D7
5	END			

[动作]



(2) 当 X10 为 ON 时，以下程序对 D20、D21 中的 32 位数据和 D10、D11 中的数据执行异或非运算，并将结果存储在 D40 和 D41 中。

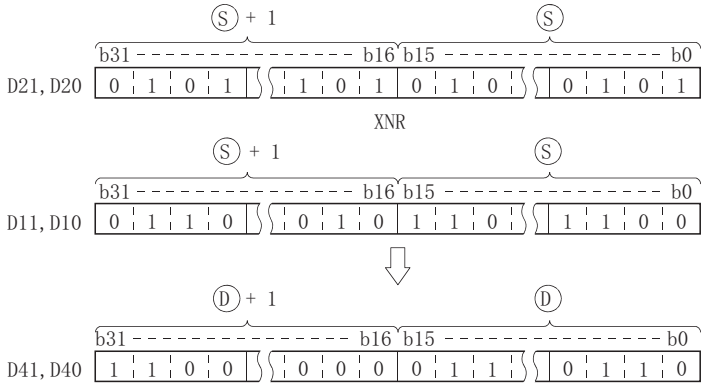
[梯形图模式]



[列表模式]

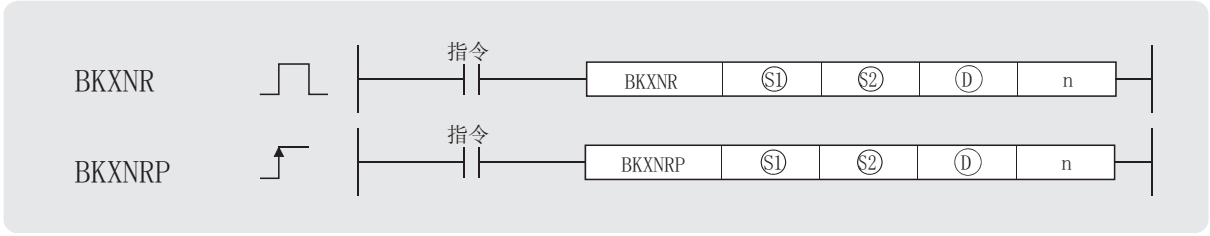
步	指令	软元件		
0	LD	X10		
1	DXNRP	D20	D10	D40
7	END			

[动作]



7.1.8 块异或非运算 (BKXNR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)*1：存储执行逻辑运算的数据的软元件的起始号 (BIN16 位)。

(S2)*1：逻辑运算数据或者存储逻辑运算数据的软元件的起始号 (BIN16 位)。

(D)*1：存储运算结果的软元件的起始号 (BIN16 位)。

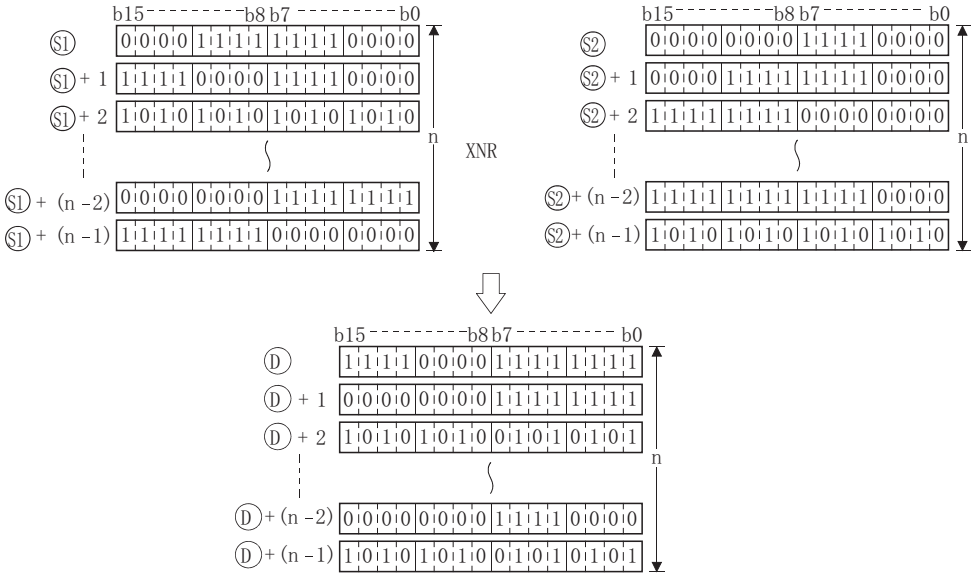
n　：运算数据块数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1) *1	---	○		—				---	---
(S2) *1	---	○		—				○	---
(D) *1	---	○		—				---	---
n	○	○		○				○	---

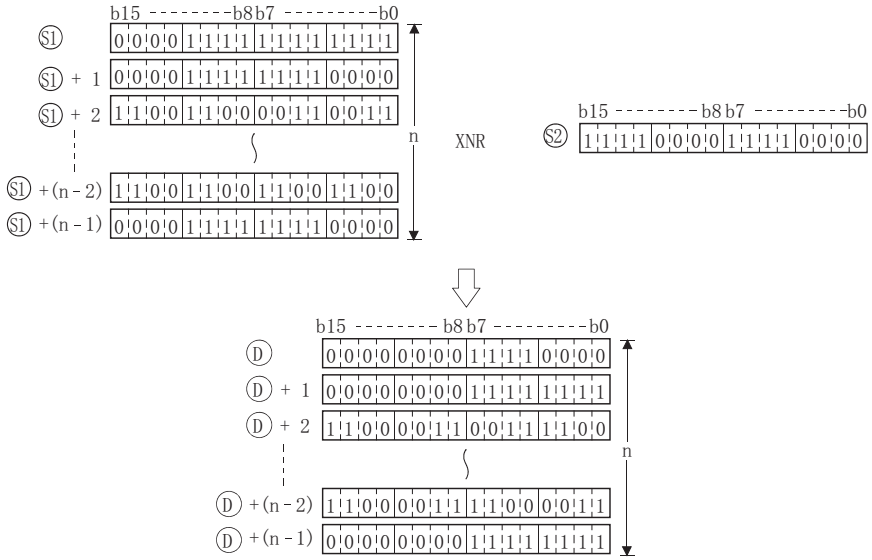
*1：可以给 (S1) 和 (D) 或者 (S2) 和 (D) 指定相同的软元件号。

功 能

(1) 对从 (S1) 指定的软元件开始的 n 点中的数据和从 (S2) 指定的软元件开始的 n 点中的数据执行异或非运算，并将结果存储在 (D) 指定的软元件中。



(2) ⓢ2 可以指定的常量的范围是 -32768 到 32767(BIN 数据)。



运算错误

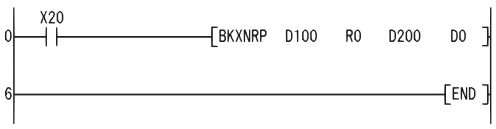
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 ⓢ1、ⓢ2 或 ⓓ 软元件开始的 n 位范围超出了相应软元件的范围。
  (出错代码 : 4101)
- 从 ⓢ1 软元件开始的 n 点软元件范围与从 ⓓ 软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ1 和 ⓓ 的软元件相同时除外。) (出错代码 : 4101)
- 从 ⓢ2 软元件开始的 n 点软元件范围与从 ⓓ 软元件开始的 n 点软元件范围相互重叠。
  (指定给 ⓢ2 和 ⓓ 的软元件相同时除外。) (出错代码 : 4101)

程序示例

(1) 当 X20 变为 ON 时，以下程序对存储在 D100 到 D102 中的数据和存储在 R0 到 R2 中的数据执行异或非运算，然后将运算结果存储在从 D200 开始的区域中。

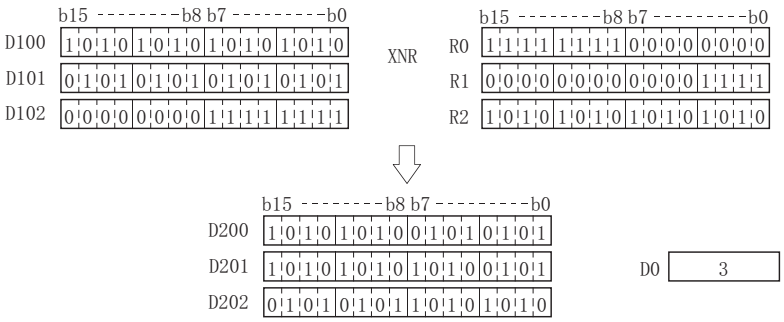
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X20			
1	BKXNRP	D100	R0	D200	D0
6	END				

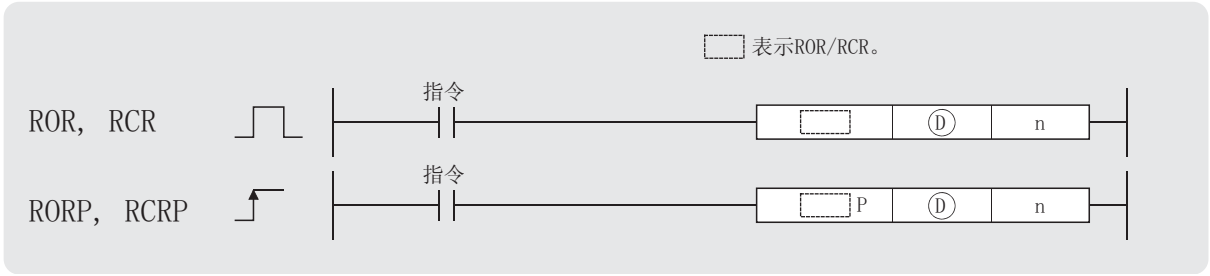
[动作]



7.2 循环指令

7.2.1 16 位数据的右循环 (ROR(P)、RCR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



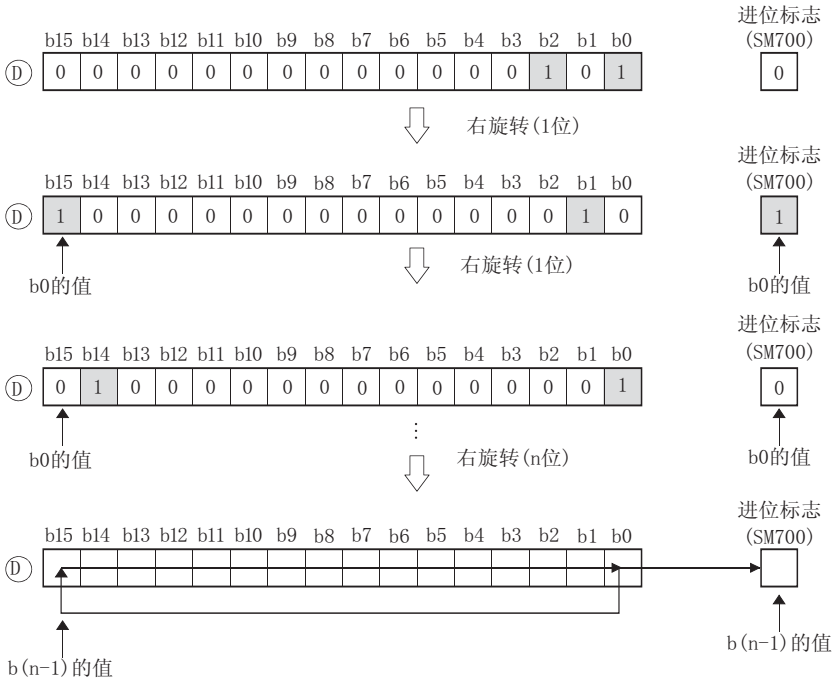
Ⓓ : 执行循环的软元件的起始号 (BIN16 位)。
n : 循环次数 (0 到 15) (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ		○						---	---
n		○						○	---

功 能

ROR

(1) 将由 Ⓓ 指定的软元件的 16 位数据，不包括进位标志，往右循环 (旋转) 移动 n 位。
进位标志的 ON/OFF 取决于 ROR 指令执行前的状态。



(2) 当位软元件被在 ⓓ 处指定后，数据循环移动指定的位数。

在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，当 n=15 且指定的位数是 12 位，则 15 除以 12 的商为 1，余数为 3，因此将循环移动 3 位。

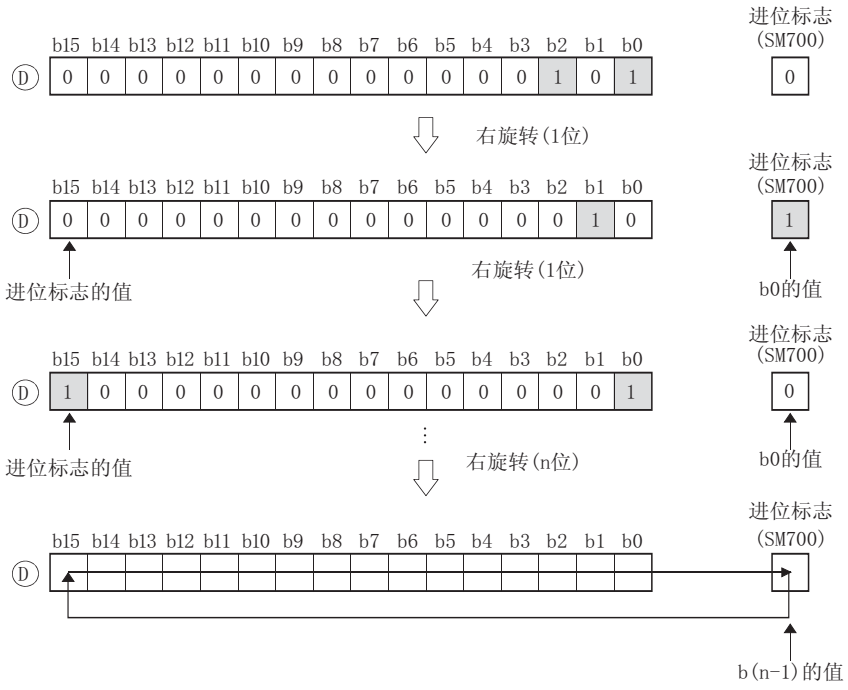
(3) 将 0 到 15 之间的任意一值指定为给 n。

当指定的 n 值为 16 或更大时，则 n 除以 16 的余数用于循环。
例如，当 n=18，18 除以 16 的余数为 2，则向右循环移动 2 位。

RCR

(1) 将由 ⓓ 指定的软元件的 16 位数据，包括进位标志，向右循环移动 n 位。

进位标志依据于它在 ROR 指令执行前的状态，变为 ON 或 OFF。



(2) 当一个位软元件被在 ⓓ 处指定时，数据将循环移动指定的位数。

在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，当 n=15 并且指定的位数是 12 位，则 15 除以 12 的商为 1，余数为 3，因此将循环移动 3 位。

(3) 将 0 到 15 之间的任意一值指定给 n。

当指定的 n 值为 16 或更大，则 n 除以 16 的余数用于循环。
例如，当 n=18，18 除以 16 的余数为 2，则向右循环移动 2 位。

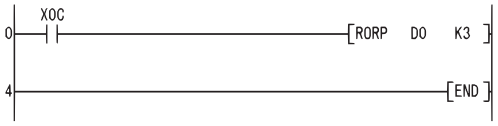
运算错误

(1) 没有与 ROR(P) 或 RCR(P) 指令有关的运行错误。

程序示例

(1) 当 XC 为 ON 时，下列程序将 D0 的内容，不包括进位标志，向右循环移动 3 位。

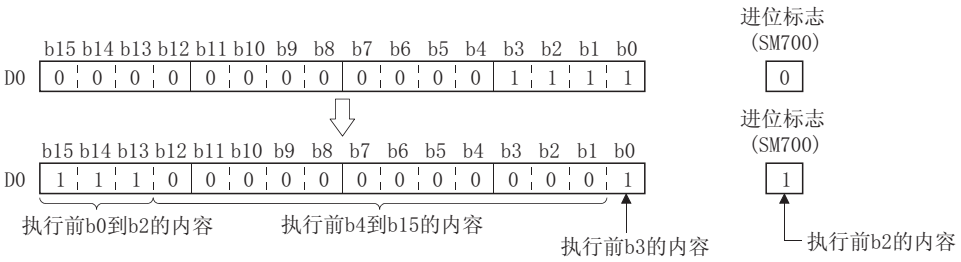
[梯形图模式]



[列表模式]

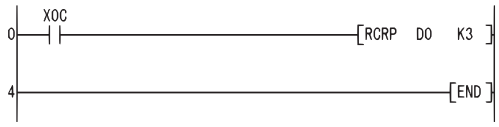
步	指令	软元件	
0	LD	XOC	
1	RORP	D0	K3
4	END		

[动作]



(2) 当 XC 为 ON 时，下列程序将 D0 的内容，包括进位标志，向右循环移动 3 位。

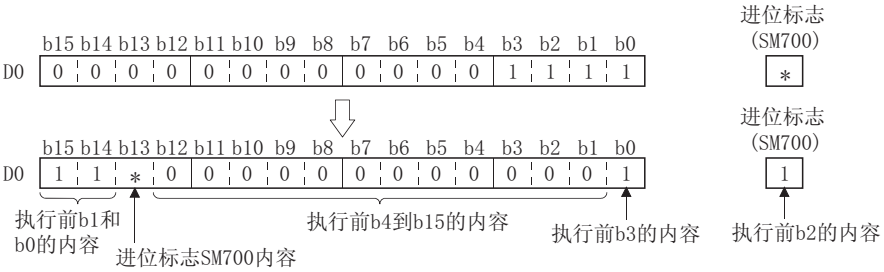
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	XOC	
1	RCRP	D0	K3
4	END		

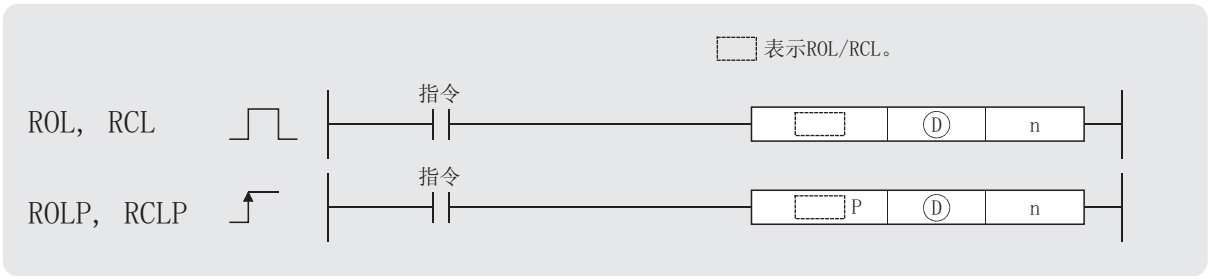
[动作]



* 进位标志依据它在RCR执行前的状态，变为ON/OFF。

7.2.2 16 位数据左循环 (ROL(P)、RCL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



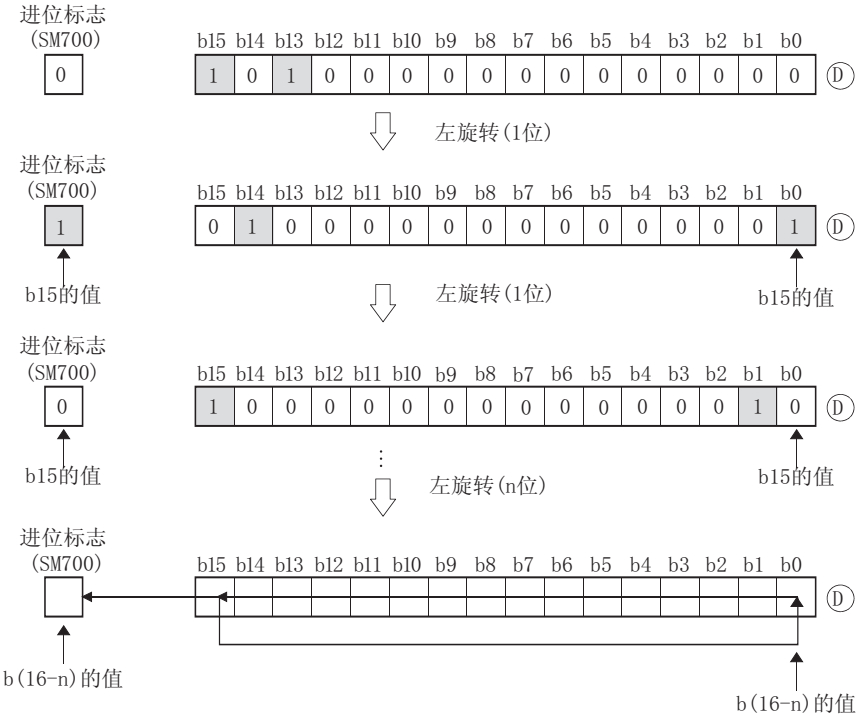
Ⓓ ：执行循环的软元件的起始号 (BIN16 位)。
n ：循环次数 (0 到 15) (BIN32 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓓ	○							---	---
n	○							○	---

功 能

ROL

(1) 将 Ⓓ 指定的软元件的 16 位数据，不包括进位标志，向左循环移动 n 位。
ROL 指令执行前的状态决定进位标志是 ON 或 OFF。



(2) 当一个位软元件在 Ⓓ 处被指定时，数据循环移动指定的位数。

在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，当 n=15 并且指定的位数是 12 位，则 15 除以 12 的商为 1，余数为 3，因此将循环移动 3 位。

(3) 将 0 到 15 中的任意一个指定给 n。

如果指定给 n 的值为 16 或更大，则 n 除以 16 的余数用于循环。
例如，当 n=18 时，因为余数为 2，因此将向右循环移动 2 位。

RCL

(1) 将 Ⓓ 指定的软元件的 16 位数据，包括进位标志，向左循环移动 n 位。

进位标志依据于它在 RCL 指令执行前的状态，变为 ON 或 OFF。



(2) 当一个位软元件在 Ⓓ 处被指定时，对数据循环移动指定的位数。

在这种情况下，实际循环移动的位数是 n 除以指定位数所得商的余数。
例如，当 n=15 且指定位数为 12 位，则 15 除以 12 的商为 1，余数为 3，因此 3 位循环。

(3) 将 0 到 15 中的任意一位指定给 n。

如果指定的 n 的值为 16 或更大，则 n 除以 16 的余数用于循环。
例如，当 n=18 时，因为余数为 2，因此将向右循环移动 2 位。

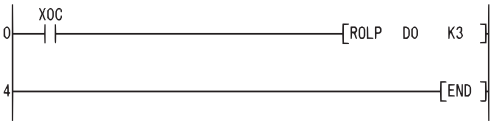
运算错误

(1) 没有与 ROL(P) 或 RCL(P) 指令有关的运行错误。

程序示例

(1) 当 XC 为 ON 时，下列程序将 D0 的内容，不包括进位标志，向左循环移动 3 位。

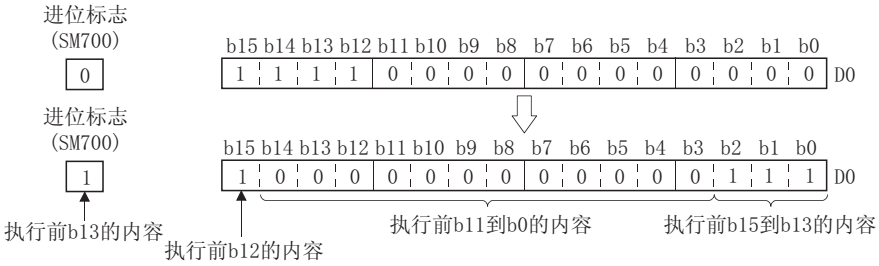
[梯形图模式]



[列表模式]

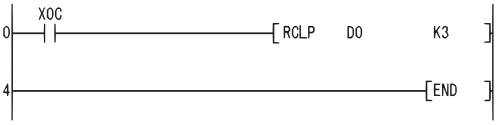
步	指令	软元件	
0	LD	XOC	
1	ROLP	D0	K3
4	END		

[动作]



(2) 当 XC 为 ON 时，下列程序将 D0 的内容，包括进位标志，向左循环移动 3 位。

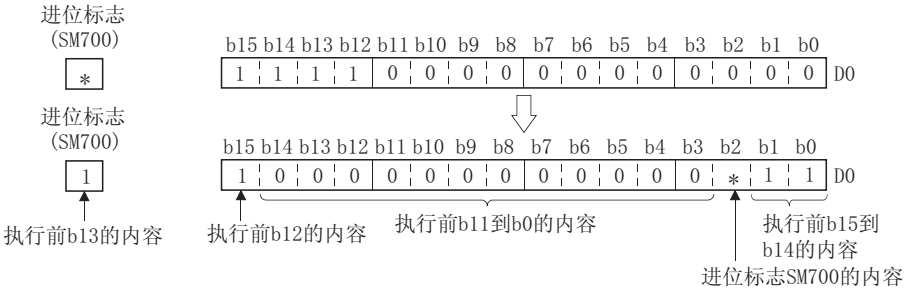
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	XOC	
1	RCLP	D0	K3
4	END		

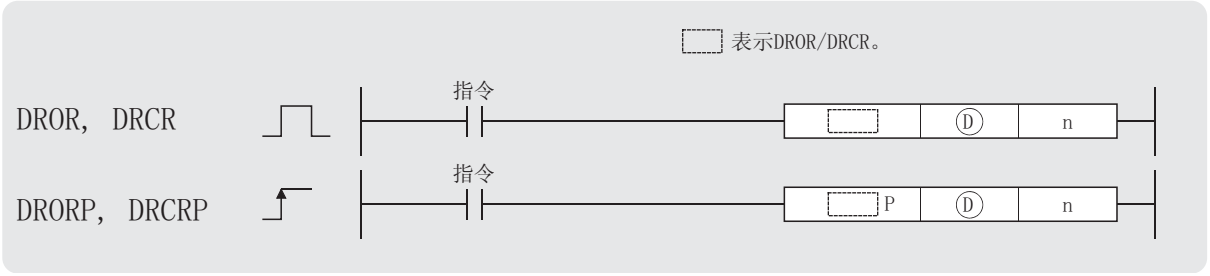
[动作]



* 进位标志依据它在RCR执行前的状态，变为ON/OFF。

7.2.3 32 位数据的右循环 (DROR(P)、DRCR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



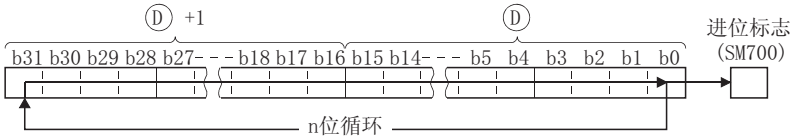
Ⓓ ：执行循环的软元件的起始号。(BIN32 位)
n ：循环次数 (0 到 31) (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ		○						---	---
n		○						○	---

功 能

DROR

(1) 将 Ⓓ 指定的软元件的 32 位数据，不包括进位标志，向右侧循环移动 n 位。
DROR 指令执行前的状态决定进位标志为 ON 或 OFF。

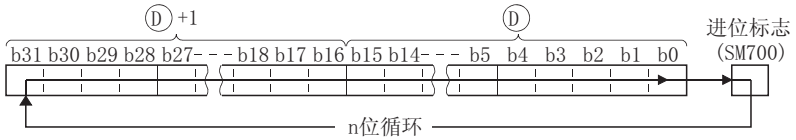


(2) 当一个位软元件在 Ⓓ 处被指定时，数据将循环移动指定的位数。
在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，如果 n=31 且数字指定位所指定的位数为 24 位，则有 7 位循环，因为计算公式为 31 除以 24 等于 1，余数为 7。

(3) 将 0 到 31 中的任意一位指定给 n。
如果指定的 n 值是 32 或更大，则 n 除以 32 的余数用于循环。
例如，当 n=34 时，因为 34 除以 32 的余数为 2，因此向右循环移动 2 位。

DRCR

(1) 在 ⓓ 指定的软元件中的 32 位数据，包括进位标志，向右循环移动 n 位。
进位标志依据于它在 DRCR 指令执行前的状态，变为 ON 或 OFF。



(2) 当一个位软元件在 ⓓ 被指定时，数据循环移动指定位数。在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，如果 n=31 并且指定位所指定的位数是 24 位，则将进行 7 位循环，因为计算公式是 31 除以 24 所得商为 1，余数是 7。

(3) 将 0 到 31 中的任意一个指定给 n。
如果指定的 n 值是 32 或更大，则 n 除以 32 的余数用于循环。例如，当 n=34 时，因为 34 除以 32 的余数为 2，因此 2 位循环到右侧。

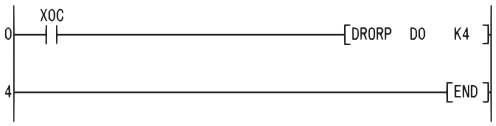
运算错误

(1) 没有与 DROR(P) 或 DRCR(P) 指令相关的运行错误。

程序示例

(1) 当 XC 为 ON 时，下列程序将 D0 和 D1 的内容，不包括进位标志，向右循环移动 4 位。

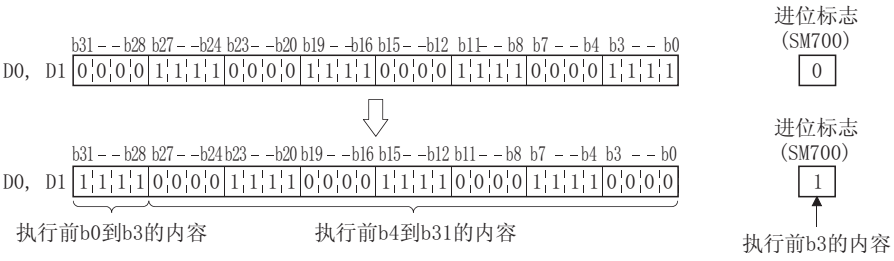
[梯形图模式]



[列表模式]

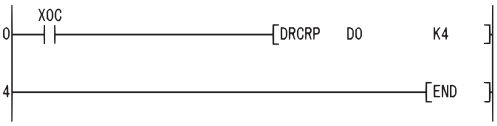
步	指令	软元件	
0	LD	XOC	
1	DRORP	D0	K4
4	END		

[动作]



(2) 当 XC 为 ON 时，下列程序将 D0 和 D1 的内容，包括进位标志，向右循环移动 4 位。

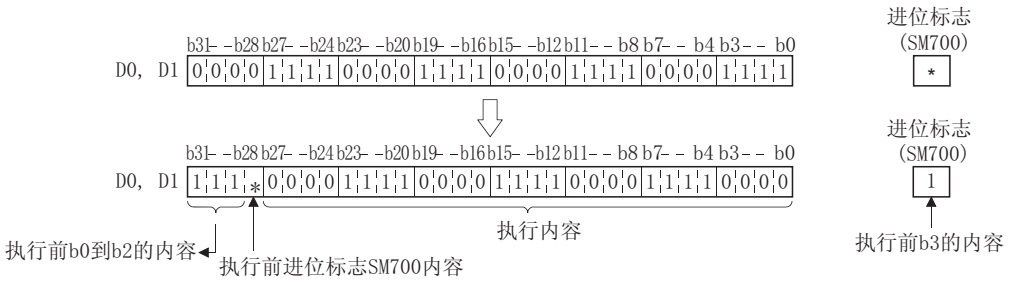
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	XOC	
1	DRCRP	D0	K4
4	END		

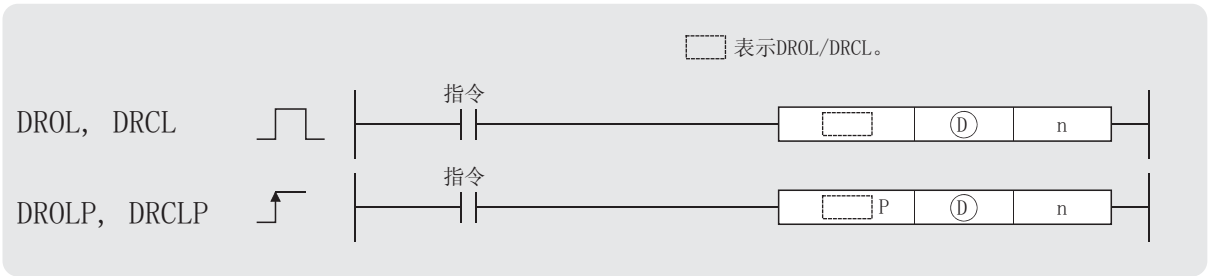
[动作]



* : 进位标志的ON/OFF状态取决于DRCR指令执行前的状态。

7.2.4 32 位数据左循环 (DROL(P)、DRCL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



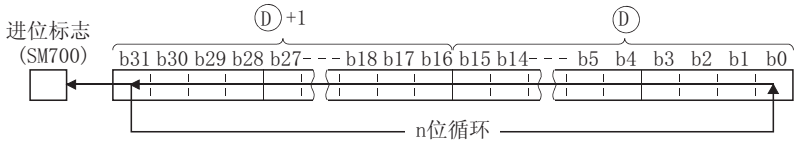
Ⓓ ：执行循环的软元件的起始号 (BIN32 位)。
n ：循环次数 (0 到 31) (BIN16 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓓ	○							---	---
n	○							○	---

功 能

DROL

(1) 将 Ⓓ 指定的软元件的 32 位数据，不包括进位标志，向左循环移动 n 位。
进位标志依据于它在执行 DROL 指令之前的状态，变为 ON 或 OFF。

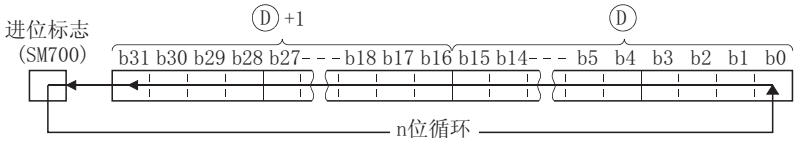


(2) 当一个位软元件在 Ⓓ 处被指定时，数据循环移动指定的位数。在这种情况下，实际循环的位数是 n 除以指定位数所得商的余数。
例如，如果 n=31 并且指定位所指定的位数是 24 位，则将循环移动 7 位，因为计算公式是 31 除以 24 所得商为 1，余数是 7。

(3) 将 0 到 31 中的任意一个指定为 n。
如果指定的 n 值是 32 或更大，则 n 除以 32 的余数用于循环。例如，当 n=34 时，因为 34 除以 32 的余数为 2，因此 2 位循环到右侧。

DRCL

(1) 在 Ⓓ 指定的软元件中，将 32 位数据，包括进位标志，向左循环移动 n 位。
进位标志依据于它在 DRCL 指令执行前的状态，变为 ON 或 OFF。



(2) 当一个位软元件在 ⓓ 处被指定时，数据循环移动指定的位数。在这种情况下，实际进行循环的位数是 n 除以指定位数所得商的余数。
例如，如果 n=31 并且位指定所指定的位数是 24 位，则将循环移动 7 位，因为计算公式是 31 除以 24 所得商为 1，余数是 7。

(3) 将 0 到 31 中的任意一个指定给 n。
如果指定的 n 值是 32 或更大，则 n 除以 32 的余数用于循环。例如，当 n=34 时，因为 34 除以 32 的余数为 2，因此 2 位循环到右侧。

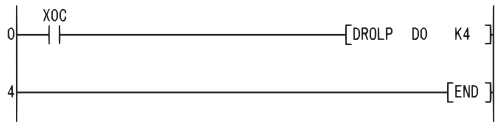
运算错误

(1) 没有与 DROL(P) 或 DRCL(P) 指令相关的运行错误。

程序示例

(1) 当 XC 为 ON 时，下列程序将 D0 和 D1 的内容，不包括进位标志，向左循环移动 4 位。

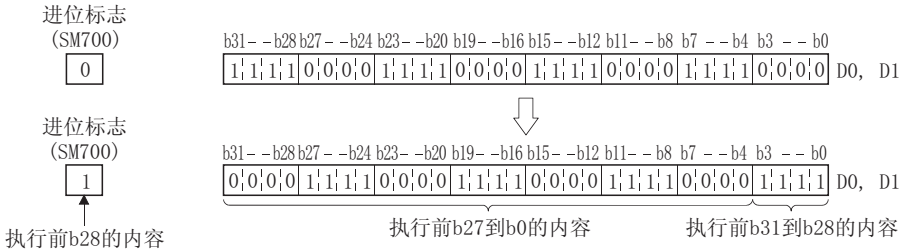
[梯形图模式]



[列表模式]

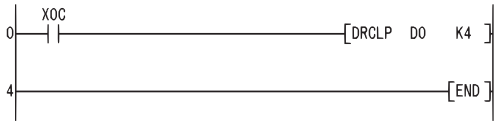
步	指令	软元件	
0	LD	XOC	
1	DROLP	D0	K4
4	END		

[动作]



(2) 当 XC 为 ON 时，下列程序循环将 D0 和 D1 的内容，包括进位标志，向左循环移动 4 位。

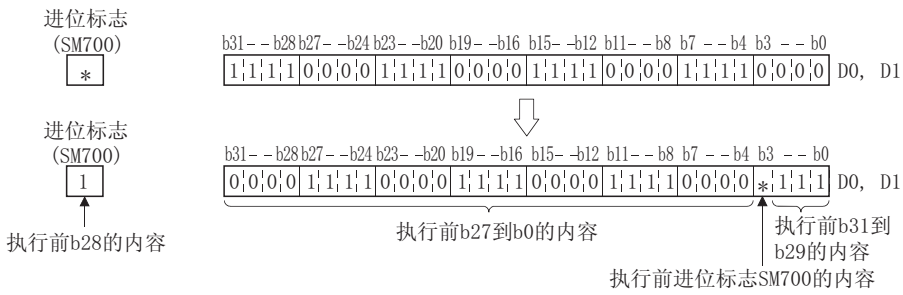
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	XOC	
1	DRCLP	D0	K4
4	END		

[动作]

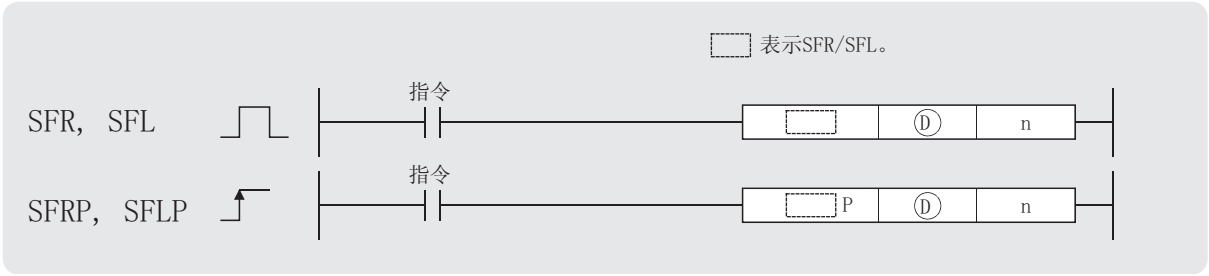


*: 进位标志的ON/OFF状态取决于DRCL指令执行前的状态。

7.3 移位指令

7.3.1 16 位数据的 n 位左移或右移 (SFR(P)、SFL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



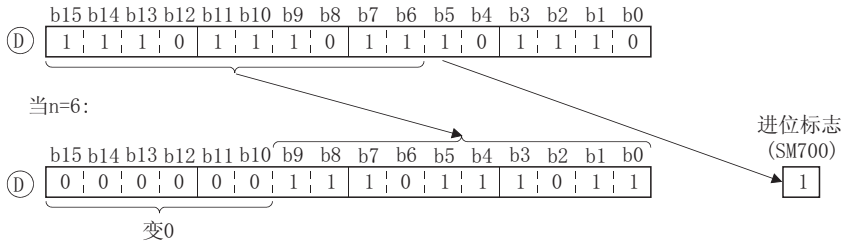
Ⓓ ：存储移位数据的软元件的起始号 (BIN16 位)。
n ：移位位数 (0 到 15)(BIN16 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓓ	○							---	---
n	○							○	---

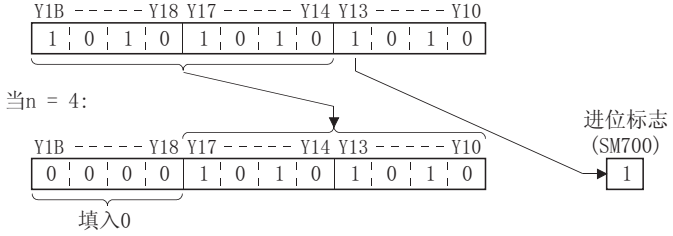
功 能

SFR

(1) 将 Ⓓ 指定的软元件中的 16 位数据右移 n 位。
从最高位开始的 n 位以 0 填入。



(2) 为 Ⓓ 指定了位软元件时，在位数指定中指定的软元件范围内向右移位。

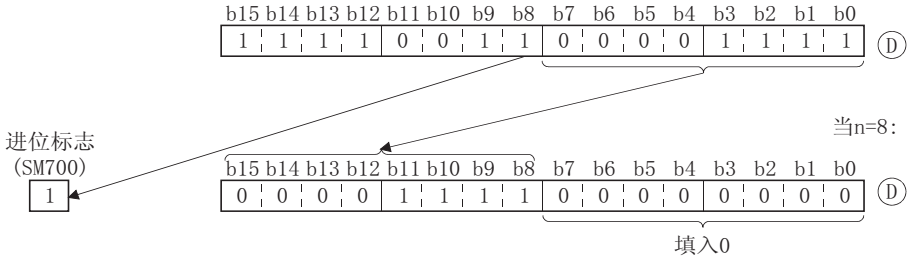


此时实际移位的位数为 n/(指定的位数)的余数。
例如，当 n=15，(指定的位数)=8 位时，15/8 =1 的余数为 “7”，数据移位为 7 位。

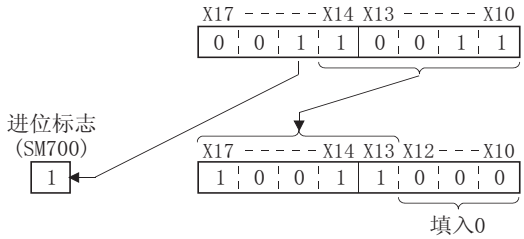
(3) 将 0 到 15 中的任意一个指定给 n。如果指定的 n 值是 16 或更大，n/16 的余数将用于向右移位。
例如，当 n=18 时，因为 18 除以 16 的余数为 2，因此向右移动 2 位。

SFL

(1) 将 ⓓ 指定的软元件中的 16 位数据左移 n 位。
从最低位开始的位到 n 位以 0 填入。



(2) 为 ⓓ 指定了位软元件时，在位数指定中指定的软元件范围内向左移位。



此时实际移位的位数为 n/(指定的位数)的余数。
例如，当 n=15，(指定的位数)=8 位时，15/8 =1 的余数为 “7”，数据移位为 7 位。

(3) 将 0 到 15 中的任意一个指定给 n。如果指定的 n 值是 16 或更大，n/16 的余数将用于向左移位。
例如，当 n=18 时， 由于 18 除以 16 的余数为 2，因此向左移动 2 位。

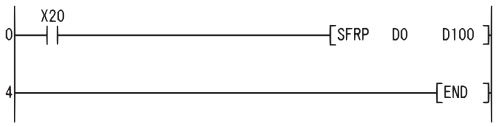
运算错误

(1) 没有与 SFR 或 SFL(P) 指令相关的运行错误。

程序示例

(1) 在 X20 为 ON 时，下列程序将 Y10 到 Y1B 的内容往右移位 D0 指定的位数。

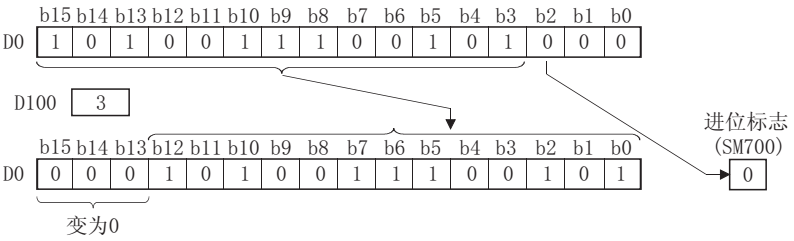
[梯形图模式]



[列表模式]

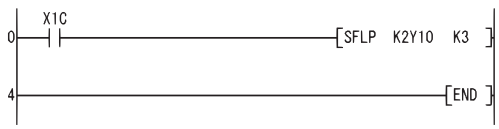
步	指令	软元件	
0	LD	X20	
1	SFRP	D0	D100
4	END		

[动作]



(2) 当 X1C 为 ON 时，下列程序将 X10 到 X17 的内容向左移位 3 位。

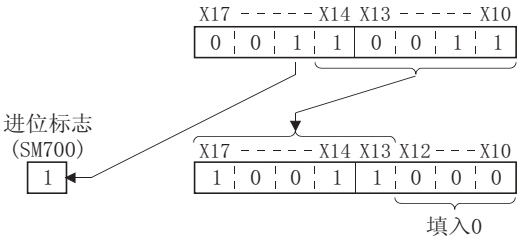
[梯形图模式]



[列表模式]

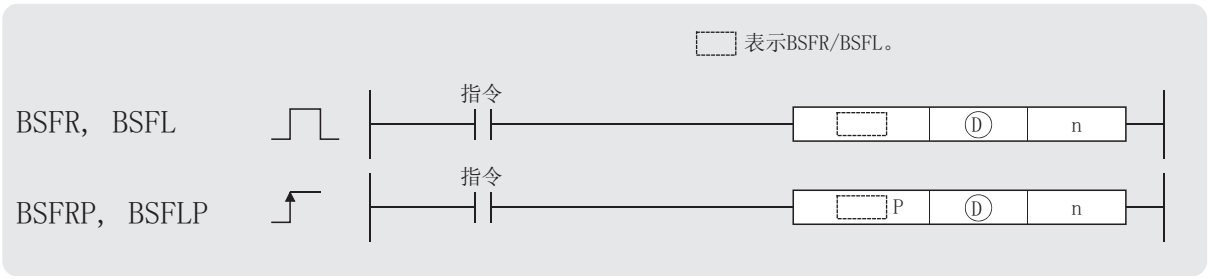
步	指令	软元件	
0	LD	X1C	
1	SFLP	K2Y10	K3
4	END		

[动作]



7.3.2 n位数据的1位左移或右移 (BSFR(P)、BSFL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



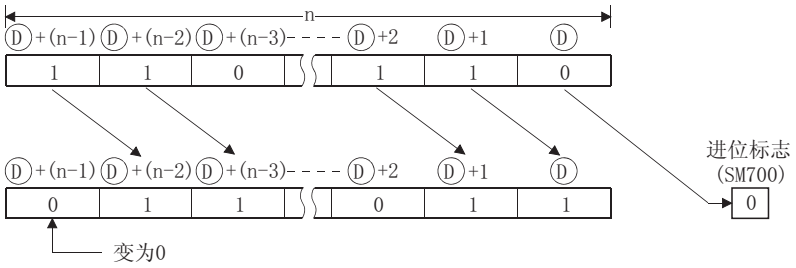
Ⓓ ：要移位的软元件的起始号（位）。
n ：执行移位的软元件数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ	○	---							---
n	○	---				○			---

功 能

BSFR

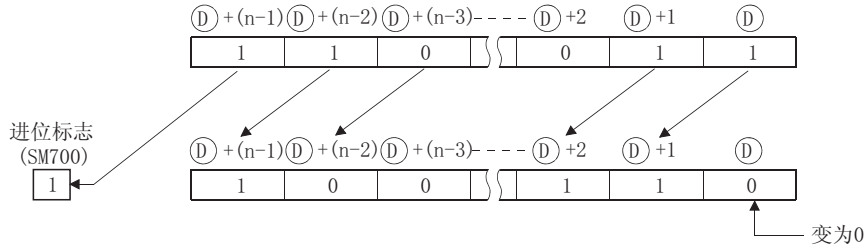
(1) 将从 Ⓓ 指定的软元件开始的 n 点数据右移 1 位。



(2) Ⓓ + (n − 1) 指定的软元件变为 0。

BSFL

(1) 将从 Ⓓ 指定的软元件开始的 n 点数据左移 1 位。



(2) Ⓓ 指定的软元件变为 0。

运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，出错代码存储在 SD0 中。

• 从 ⓓ 软元件开始的 n 点数据软元件的范围超过了相关软元件。（出错代码： 4101）

程序示例

(1) 当 X8F 为 ON 时，下列程序将 M668 到 M676 移位到右边。

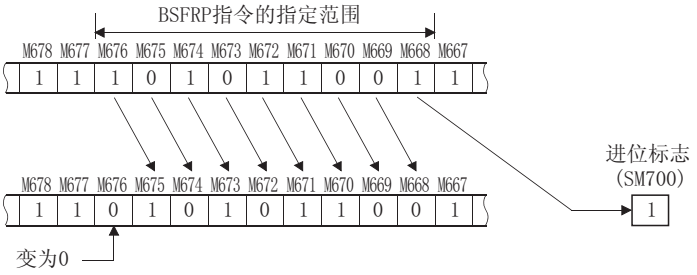
[梯形图模式]



[列表模式]

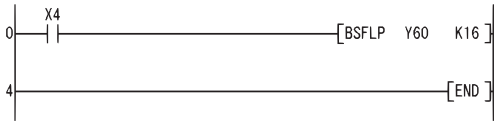
步	指令	软元件	
0	LD	X8F	
1	BSFRP	M668	K9
4	END		

[动作]



(2) 当 X4 为 ON 时，下述程序将 Y60 到 Y6F 的输出移位到左边。

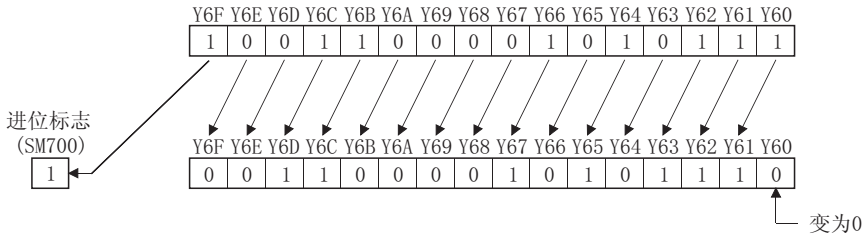
[梯形图模式]



[列表模式]

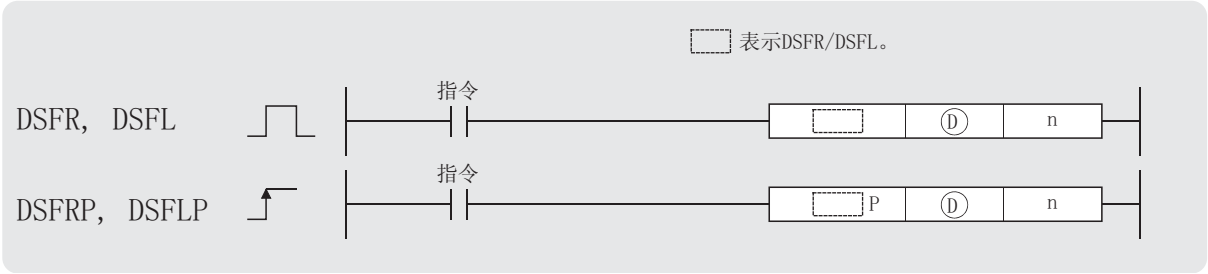
步	指令	软元件	
0	LD	X4	
1	BSFLP	Y60	K16
4	END		

[动作]



7.3.3 n字数据的1字左移或右移 (DSFR(P)、DSFL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



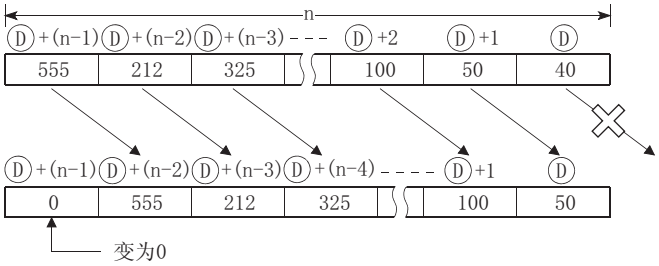
Ⓓ : 要移位的软元件的起始号 (BIN16 位)。
n : 执行移位的软元件数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ	---	○		---					---
n	○	○		○					---

功 能

DSFR

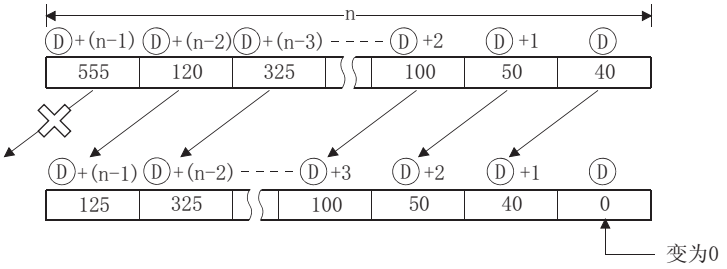
(1) 将从 Ⓓ 指定的软元件开始的 n 点数据右移 1 字。



(2) Ⓓ + (n − 1) 指定的软元件变为 0。

DSFL

(1) 将从 Ⓓ 指定的软元件开始的 n 点数据左移 1 字。



(2) Ⓓ 指定的软元件变为 0。

运算错误

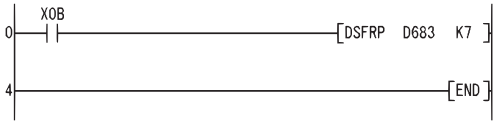
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 从 ⓓ 软元件开始的 n 点软元件的范围超过了相关软元件。 （出错代码： 4101）

程序示例

(1) 当 XB 为 ON 时，下述程序将 D683 到 D689 的内容移位到右边。

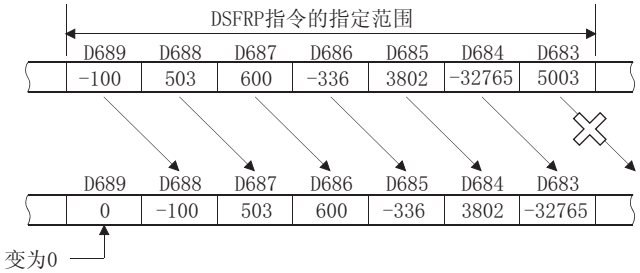
[梯形图模式]



[列表模式]

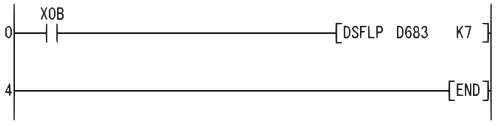
步	指令	软元件	
0	LD	XOB	
1	DSFRP	D683	K7
4	END		

[动作]



(2) 当 XB 变为 ON 时，下述程序将 D683 到 D689 的内容移位到左边。

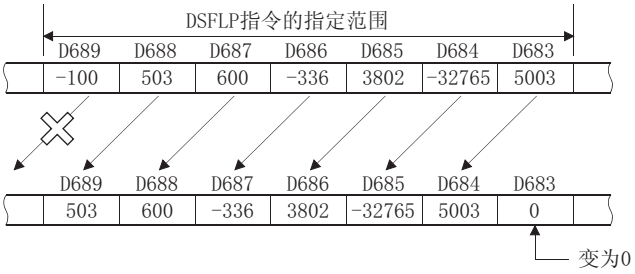
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	XOB	
1	DSFLP	D683	K7
4	END		

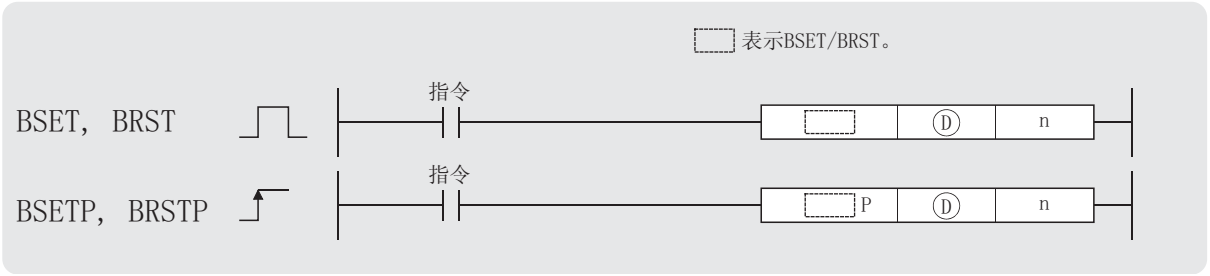
[动作]



7.4 位处理指令

7.4.1 字软元件的位设定和复位 (BSET(P)、BRST(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ ：进行置位或复位的软元件号 (BIN16 位)。
n ：进行置位或复位的位号 (0 到 15) (BIN16 位)。

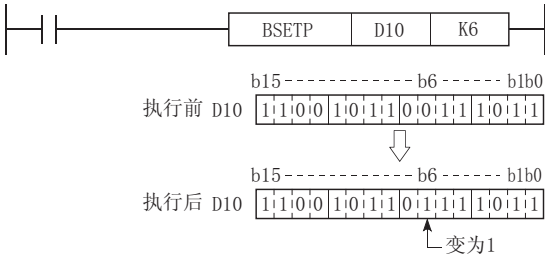
设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓓ	○							---	---
n	○							○	---

功 能

BSET

(1) 对 Ⓓ 指定的字软元件的第 n 位进行置位 (设置为 “1”)。

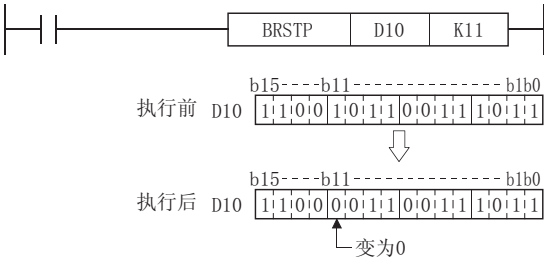
(2) n 值可以设定在 0 到 15 之间。如果此值超过 15，此运算将执行数据的低 4 位。



BRST

(1) 将 Ⓓ 指定的字软元件的第 n 位复位到 0。

(2) n 值可以设定在 0 到 15 之间。如果此值超过 15，此运算将执行数据低 4 位。



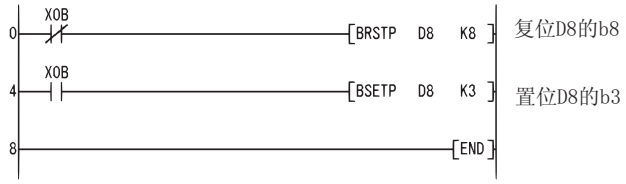
运算错误

(1) 没有与 BSET(P) 或 BRST(P) 指令相关的运行错误。

程序示例

(1) 当 XB 为 OFF 时，下列程序将 D8 的第 8 位 (b8) 复位到 0，当 XB 为 ON 时将 D8 的第 3 位 (b3) 置位到 1。

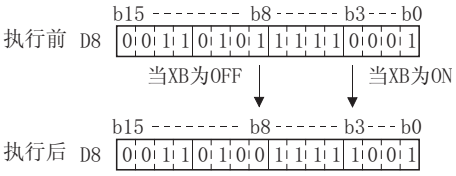
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LDI	X0B	
1	BRSTP	D8	K8
4	LD	X0B	
5	BSETP	D8	K3
8	END		

[动作]



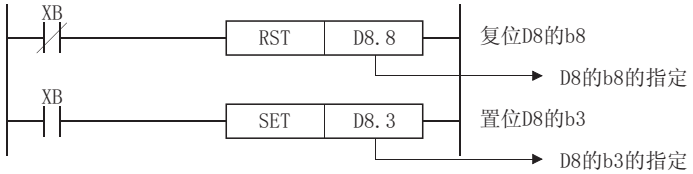
备 注

字软元件的置位或复位也可以通过字软元件的位指定来进行。

关于字软元件的位指定请参考以下手册：

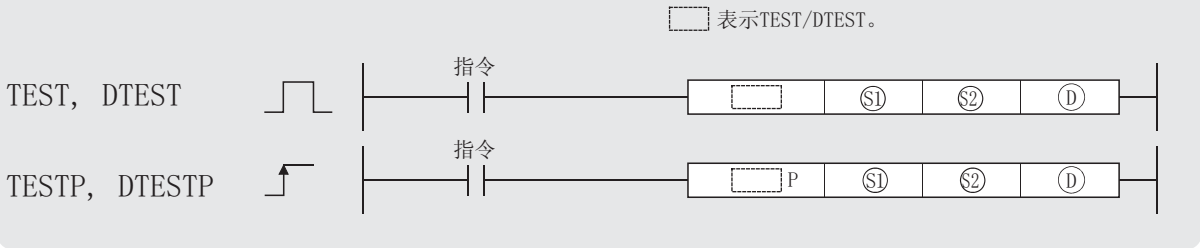
- QCPU 用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础篇）

如果使用了字软元件的位指定，程序示例 (1) 的处理过程可按如下进行。



7.4.2 位测试 (TEST(P)、DTEST(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)：存储被提取的位数据的软元件号 (BIN16/32 位)。

(S2)：被提取的位数据的位置 (0 到 15(TEST)/0 到 31(DTEST)) (BIN16/32 位)。

(D)：存储被提取的位数据的位软元件号 (位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
(S1)				○			○	---	---
(S2)				○			○	○	---
(D)				○			---	---	---

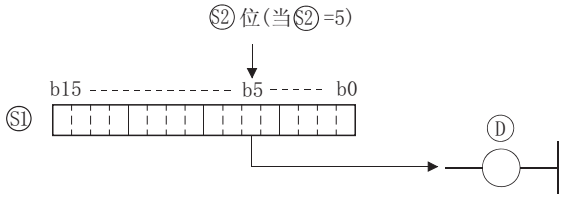
功 能

TEST

(1) (S2) 指定的字软元件内，在 (S1) 指定的位置取出位数据，并且将它写到 (D) 指定的位软元件。

(2) 当相关位为 “0” 时，(D) 指定的位软元件为 OFF，当相关位为 “1” 时，(D) 指定的位软元件为 ON。

(3) (S2) 指定的位置表示了在 1 个字数据块 (0 到 15) 里单个位的位置。
当在 (S2) 指定了 16 或更大的数时，则目标是 n 除以 16 的余数指定的位置的位数据。例如，当 n=18 时，因为 18 除以 16 的余数为 2，所以目标为在 b2 处的数据。



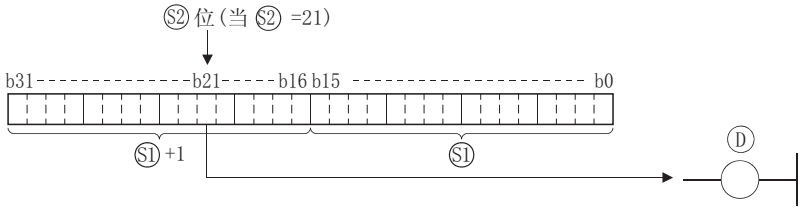
DTEST

(1) 在 (S1) 或 (S1)+1 指定的 2 字的软元件内指定的位置取出位数据，并且将它写到 (D) 指定的位软元件。

(2) 当相关位为 “0” 时，D 指定的位软元件为 OFF，当相关位为 “1” 时，(D) 指定的位软元件为 ON。

(3) ⓢ2 指定的位置表示了在 2 字数据块 (0 到 31) 里单个位的位置

当在 ⓢ2 指定了 32 或更大的数时，则目标是 n 除以 32 的余数指定位置的位数据。例如，当 n=34 时，因为 34 除以 32 的余数为 2，所以目标为在 b2 的数据。



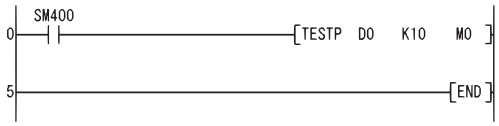
运算错误

(1) 没有与 TEST(P) 或 DTEST(P) 指令相关的运行错误。

程序示例

(1) 根据在 1 字数据块 (D0) 里的第 10 位的状态，下列程序将 M0 变为 ON 或 OFF。

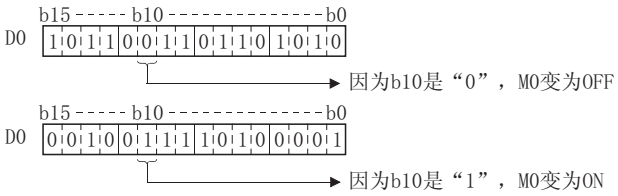
[梯形图模式]



[列表模式]

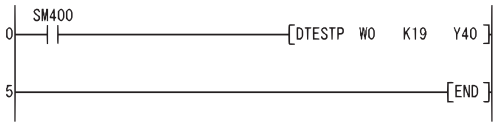
步	指令	软元件		
0	LD	SM400		
1	TESTP	D0	K10	M0
5	END			

[动作]



(2) 根据在 W0 和 W1 的 2 字数据块 (D0) 里的第 19 位的状态，下列程序将 Y40 变为 ON 或 OFF。

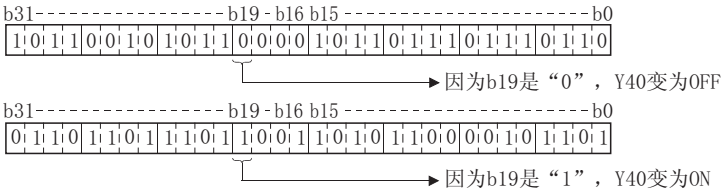
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	SM400		
1	DTESTP	W0	K19	Y40
5	END			

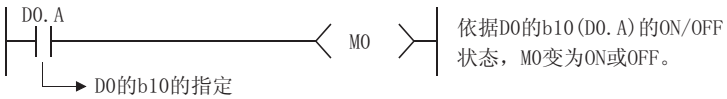
[动作]



备 注

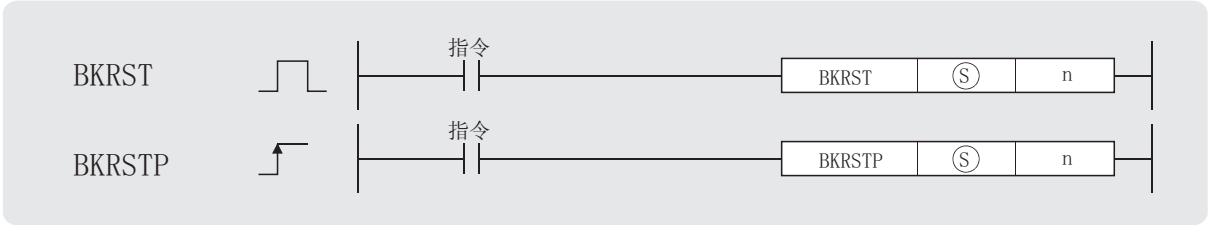
可以像使用字软元件的位指定那样使用位测试指令写程序。

如果程序示例 (1) 中的程序改变为使用字软元件的位指定，它将显示如下。



7.4.3 位软元件的成批复位 (BKRST(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：被复位的软元件的起始号（位）。
n ：被复位的软元件数 (BIN16 位)。

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
Ⓢ		○			---				---
n		○			○				---

功 能

(1) 从 Ⓢ 指定的位软元件中复位 n 位位软元件。

软元件	状态
报警器 (F)	• 将 Ⓢ 指定的报警器 (F) 号里的软元件的 n 点置为 OFF。 • 将报警器号 SD64 到 SD79 的数据置为 OFF 并且将剩余的数据向前压缩。 • 将存储在 SD64 到 SD79 里的报警器号存储在 SD63 中。
(T) 定时器 (C) 记数器	• 将从 Ⓢ 指定的定时器 (T) 或计数器 (C) 开始的 n 点当前值设置成 0，并将线圈触点置为 OFF。
其它位软元件	• 将 Ⓢ 指定的软元件的线圈或触点 n 点置为 OFF。

(2) 如果指定的软元件是 OFF，则软元件状态将不会改变。

运算错误

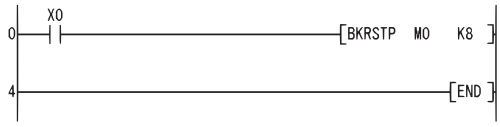
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变 ON，并且出错代码存储在 SD0 中。

• 从 Ⓢ 软元件开始的 n 点范围超过相关软元件。　（出错代码： 4101）

程序示例

(1) 当 X0 变 ON 时，下列程序将 M0 到 M7 的软元件置为 OFF。

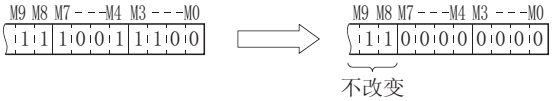
[梯形图模式]



[列表模式]

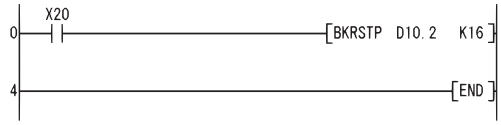
步	指令	软元件	
0	LD	X0	
1	BKRSTP	M0	K8
4	END		

[动作]



(2) 当 X20 变 ON 时，下列程序将 D10 的第二位 (b2) 到 D11 的第一位 (b1) 的数据置为 0。

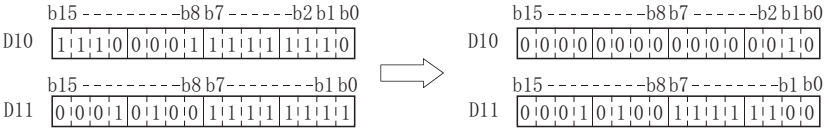
[梯形图模式]



[[列表模式]]

步	指令	软元件	
0	LD	X20	
1	BKRSTP	D10.2	K16
4	END		

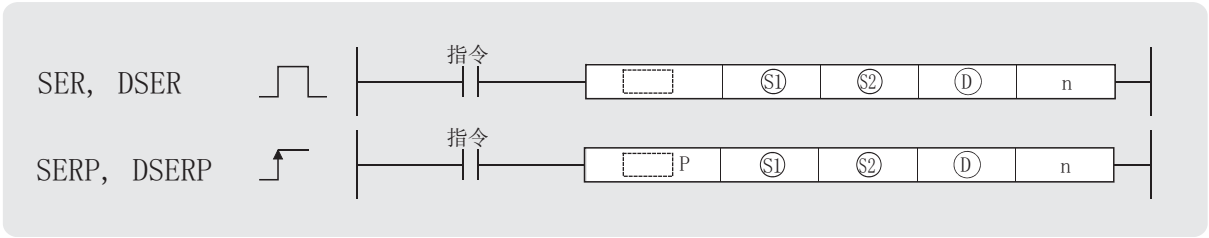
[动作]



7.5 数据处理指令

7.5.1 16 位和 32 位数据搜索 (SER(P)、DSER(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



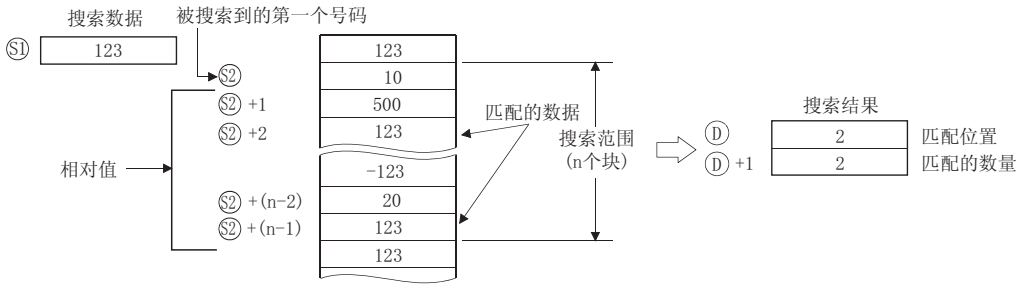
(S1) ：搜索数据或存储搜索数据的软元件的起始号 (BIN16/32 位)。
(S2) ：搜索目标的数据，或存储此类数据的软元件的起始号 (BIN16 位)。
(D) ：存储搜索结果的软元件的起始号 (BIN16 位)。
n ：搜索次数 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	○	○		○		○		○	---
(S2)	---	○		---		---		---	---
(D)	---	○		---		○		---	---
n	○	○		○		○		○	---

功 能

SER

(1) 将 (S1) 指定的软元件的 16 位数据作为入口代码来搜索从 (S2) 指定的软元件的 16 位数据开始的 n 个块。
与入口代码匹配的号码存储在 (D)+1 指定的软元件中，并且从 (S2) 开始，发现第一个匹配的软元件的点数的相关值存储在 (D) 指定的软元件中。

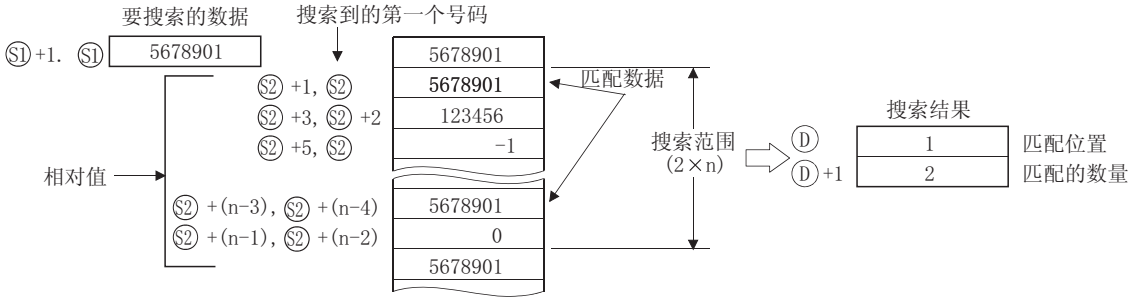


(2) 如果 n 为 0 或负，则不进行处理。

(3) 如果在搜索过程中没有发现匹配数据，则 (D) 和 (D)+1 指定的软元件变为 “0”。

DSER

(1) 将 (S1) +1， (S1) 指定的软元件的 32 位数据作为搜索的入口代码，并且对从 (S2)(2 × n 点）指定的软元件开始的 32 位数据单元里的 n 点进行搜索。
与入口代码的匹配数存储在 (D)+1 指定的软元件中，并且从 (S2) 开始，发现第一个匹配的软元件的点数的相关值存储在 (D) 指定的软元件中。



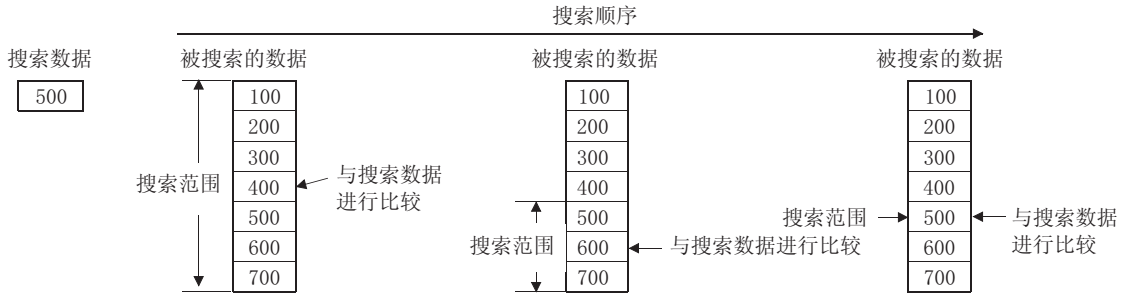
(2) 如果 n 为 0 或负，则不进行处理。

(3) 如果在搜索过程中没有发现匹配数，则 (D) 和 (D)+1 指定的软元件变为 “0”。

要 点

在 SER、DSER 指令中，当被搜索的数据为升序排序时，如果使 SM702 *1 为 ON，通过二分搜索法进行搜索，可以加快搜索处理速度。
被搜索的数据未以升序排序时，如果使 SM702 为 ON，将不能获得正常的搜索结果。

*1: SM702 是用于设定搜索方法的特殊继电器。
- SM702 为 OFF: 逐次搜索法（线性搜索法）
（从被搜索的数据的起始开始与搜索数据进行比较的方法。）
- SM702 为 ON: 二分搜索法
（该方法对于以升序排序的数据，找出位于搜索范围中间的值，将该值与要搜索的值进行比较，以确定搜索方向，锁定及缩小搜索范围。如此反复执行此操作，找出要搜索的数据。）



运算错误

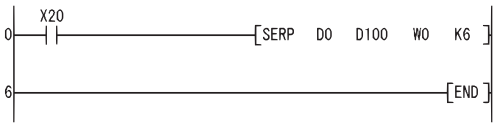
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 从 (S2) 软元件开始的 n 点数据的位置超过指定的软元件范围。
（出错代码： 4101）
- (D) 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU）
（出错代码： 4101）

程序示例

(1) 当 X20 为 ON 时，下述程序将搜索 D100 到 D105 的 D0 内容，并将搜索结果存储到 W0 和 W1。

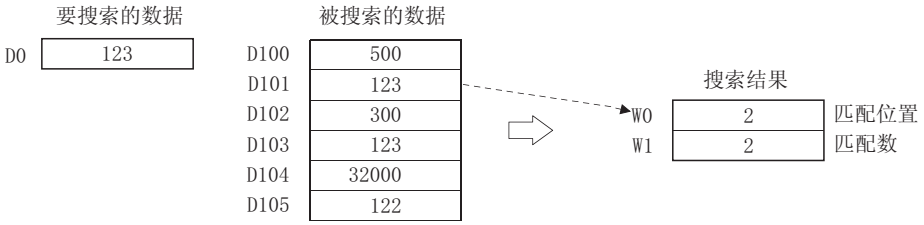
[梯形图模式]



[列表模式]

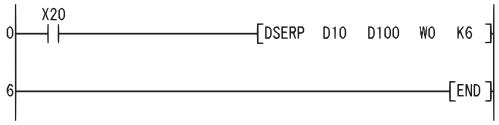
步	指令	软元件
0	LD	X20
1	SERP	D0 D100 W0 K6
6	END	

[动作]



(2) 当 X20 为 ON 时，下述程序将在 D100 到 D111 中搜索 D11 和 D10 的内容，并将搜索结果存储到 W0 和 W1。

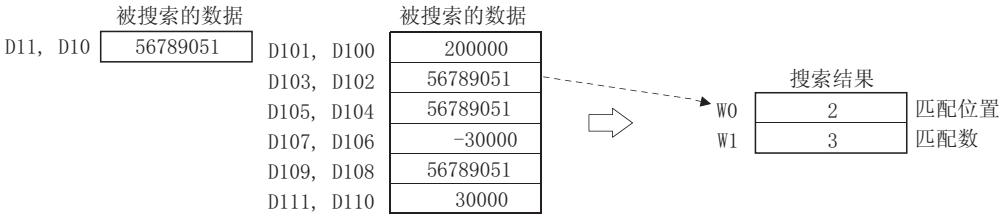
[梯形图模式]



[列表模式]

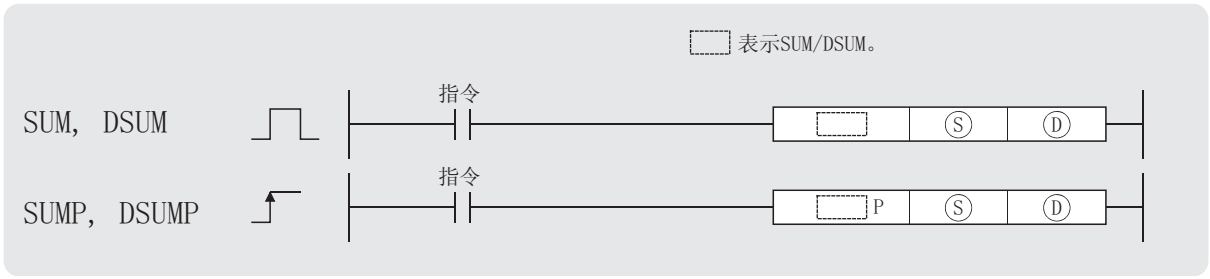
步	指令	软元件
0	LD	X20
1	DSERP	D10 D100 W0 K6
6	END	

[动作]



7.5.2 16 位和 32 位数据的位数据检查 (SUM(P)、DSUM(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：对变为“1”的位的总数进行计数的软元件的起始号 (BIN16/32 位)。

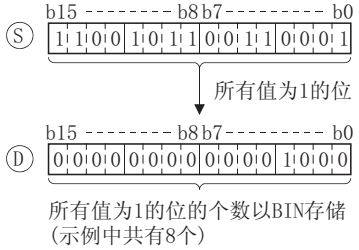
Ⓓ：存储位总数的软元件的起始号 (BIN16/32 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○						○	---
Ⓓ		○						---	---

功 能

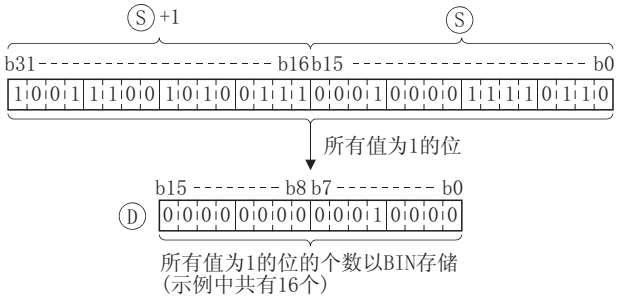
SUM

在 Ⓢ 指定的软元件里的 16 位数据中，将所有值为 1 的位的个数存储在 Ⓓ 指定的软元件中。



DSUM

在 Ⓢ 指定的软元件里的 32 位数据中，将所有值为 1 的位的总数存储在 Ⓓ 指定的软元件中。



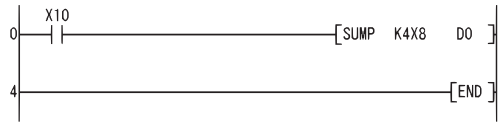
运算错误

(1) 没有与 SUM(P) 或 DSUM(P) 指令相关的运行错误。

程序示例

(1) 当 X10 为 ON 时，下列程序将 X8 到 X17 里为 ON 的位的位数存储到 D0 中。

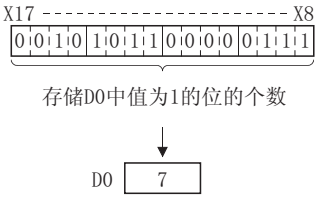
[梯形图模式]



[列表模式]

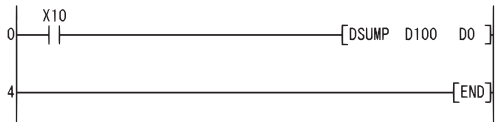
步	指令	软元件
0	LD	X10
1	SUMP	K4X8 D0
4	END	

[动作]



(2) 当 X10 为 ON 时，下列程序将 D100 和 D101 里为 ON 的位的个数存储到 D0 中。

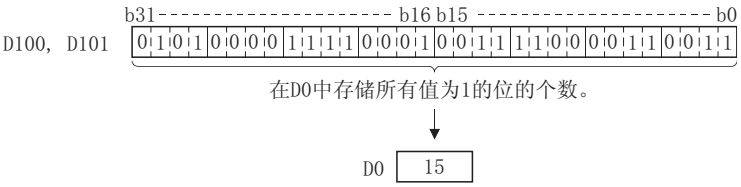
[梯形图模式]



[列表模式]

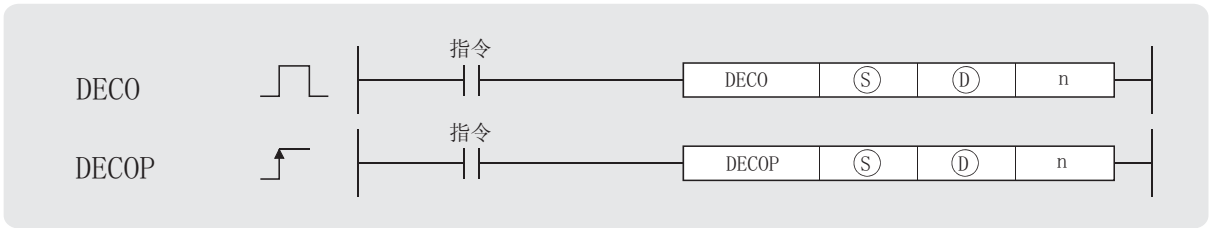
步	指令	软元件
0	LD	X10
1	DSUMP	D100 D0
4	END	

[动作]



7.5.3 8 位到 256 位的解码 (DECO(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：解码的数据或储存了解码数据的软元件号 (BIN16/32 位)。

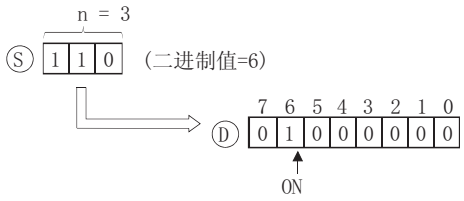
Ⓓ ：存储解码结果的软元件的起始号 (软元件名)。

n ：有效位的长度 (1 到 8)，0: 不处理 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○			○				○	---
Ⓓ	○			---				---	---
n	○			○				○	---

功 能

(1) 使与 Ⓢ 的低 n 位指定的二进制值对应的 Ⓓ 的位位置为 ON。



(2) n 值可以在 1 到 8 之间指定。

(3) 如果 n=0 则不进行处理，在 Ⓓ 指定的软元件里 2^n 位没有改变。

(4) 位软元件作为 1 位，字软元件作为 16 位。

运算错误

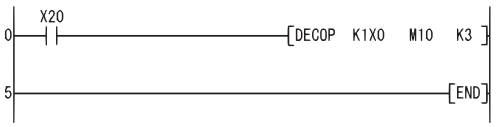
(1) 在下列情况下发生运行错误，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- n 值不在 0 到 8 范围之间。 (出错代码： 4100)
- ⓓ 里 2^n 位范围超过了相关软元件范围。 (出错代码： 4101)

程序示例

(1) 当 X20 为 ON 时，下列程序将 X0 开始的 3 位数据解码并将结果存储在 M10 中。

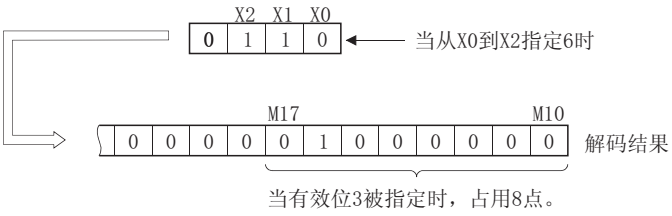
[梯形图模式]



[列表模式]

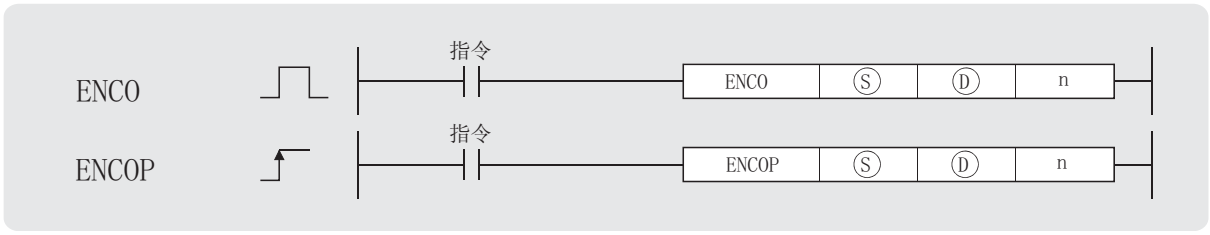
步	指令	软元件		
0	LD	X20		
1	DECOP	K1X0	M10	K3
5	END			

[动作]



7.5.4 将 256 位编码为 8 位 (ENCO(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

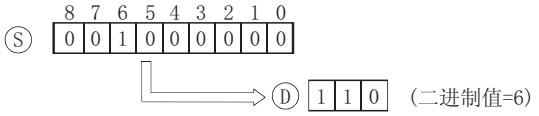


Ⓢ ：编码的数据或储存了编码数据的软元件的起始号（软元件名）。
Ⓓ ：存储编码结果的软元件号 (BIN16 位)。
n ：有效位的长度 (1 到 8)，0: 不处理 (BIN16 位)。

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ		○			---			---	---
Ⓓ		○			○			---	---
n		○			○			○	---

功 能

(1) 从 Ⓢ 的 2^n 位的数据中，将变为 “1” 的位对应的二进制值存储到 Ⓓ 中。



(2) n 的有效设置为 1 到 8。

(3) 如果 n=0，将没有运算操作，并且 Ⓓ 的内容不改变。

(4) 位软元件作为 1 位处理，字软元件作为 16 位处理。

(5) 如果有多个位为 1 时，将在最高位进行处理。

运算错误

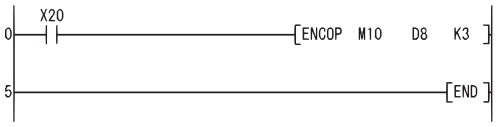
(1) 在下列情况下发生运行错误，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- n 值不在 0 到 8 范围内。 (出错代码： 4100)
- 从 Ⓢ 开始的 2^n 位的值超过 0 到 8 范围。 (出错代码： 4101)
- 从 Ⓢ 开始的 2^n 位的数据均为 “0”。 (出错代码： 4100)

程序示例

(1) 当 X20 为 ON 时，下述程序将 M10 开始的 3 位进行编码，并将结果存储在 D8 中。

[梯形图模式]



[列表模式]

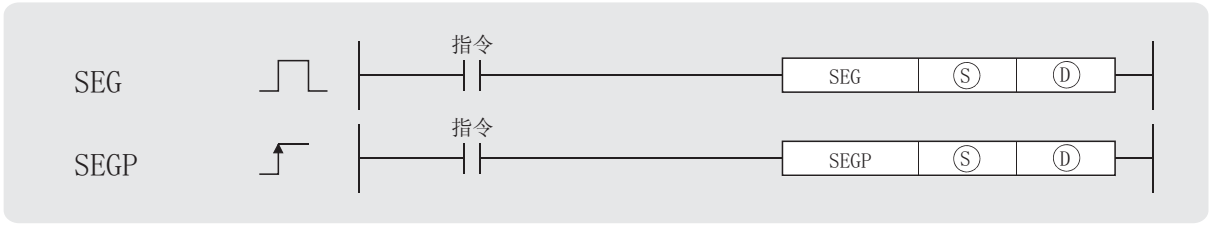
步	指令	软元件		
0	LD	X20		
1	ENCOP	M10	D8	K3
5	END			

[动作]



7.5.5 7段解码 (SEG(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：解码的数据或存储解码数据的软元件的第一个软元件号 (BIN 16 位)

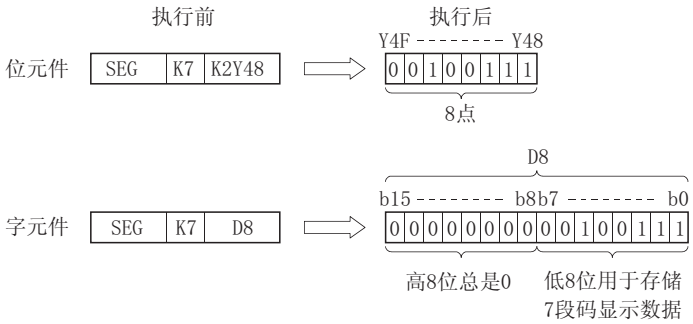
Ⓓ：存储解码结果的软元件的第一个软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

(1) 将 Ⓢ 低 4 位指定的 0 到 F 数据解码成 7 段显示数据，并存储到 Ⓓ 中。

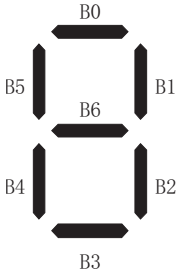
(2) 如果 Ⓓ 是一个位软元件，表示存储 7 段显示数据的软元件的起始号。如果 Ⓓ 是一个字软元件，则表示存储数据的软元件号。



运算错误

(1) 没有与 SEG(P) 指令相关的运行错误。

7- 段解码显示

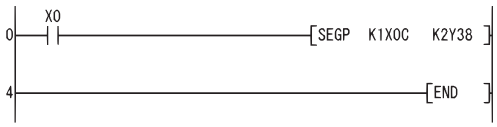
ⓢ		7 段码配置	ⓓ								显示数据
十六进制	位状态		B7	B6	B5	B4	B3	B2	B1	B0	
0	0000		0	0	1	1	1	1	1	1	0
1	0001		0	0	0	0	0	1	1	0	1
2	0010		0	1	0	1	1	0	1	1	2
3	0011		0	1	0	0	1	1	1	1	3
4	0100		0	1	1	0	0	1	1	0	4
5	0101		0	1	1	0	1	1	0	1	5
6	0110		0	1	1	1	1	1	0	1	6
7	0111		0	0	1	0	0	1	1	1	7
8	1000		0	1	1	1	1	1	1	1	8
9	1001		0	1	1	0	1	1	1	1	9
A	1010		0	1	1	1	0	1	1	1	A
B	1011		0	1	1	1	1	1	0	0	b
C	1100		0	0	1	1	1	0	0	1	C
D	1101		0	1	0	1	1	1	1	0	d
E	1110		0	1	1	1	1	0	0	1	E
F	1111		0	1	1	1	0	0	0	1	F

位元件第一位
字元件的最低位

程序示例

(1) 当 X0 为 ON 时，下列程序将 XC 到 XF 的数据转换成 7 段码显示数据并输出到 Y38 到 Y3F 中。

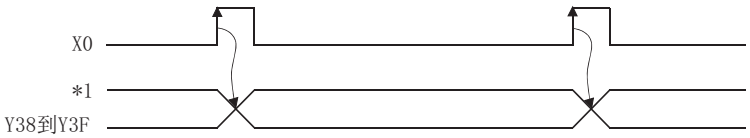
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	SEGP	K1X0C	K2Y38
4	END		

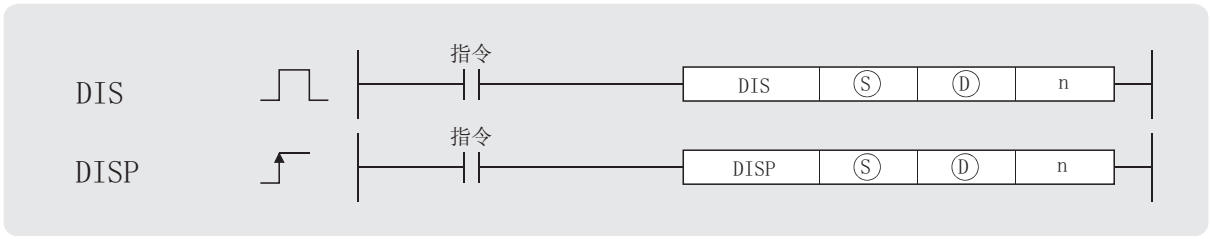
[时序图]



*1: 从Y38到Y3F中的数据在下一个数据被输出之前不会改变。

7.5.6 16 位数据的 4 位分组 (DIS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储要分离的数据的软元件的第一个软元件号 (BIN 16 位)

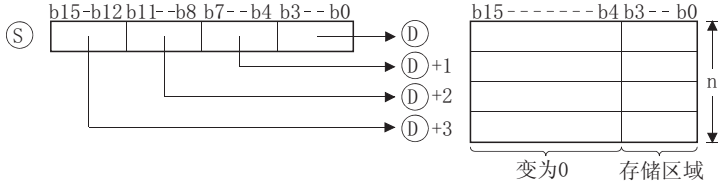
Ⓓ ：存储已分离的数据的软元件的第一个软元件号 (BIN 16 位)

n ：分离数 (1 到 4)：如果分离数为 0，则不执行任何处理 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○			○				---
Ⓓ	---	○			---				---
n	○	○			○				---

功 能

(1) 将 Ⓢ 指定的 16 位数据的低 n 位数字 (1 位数字为 4 位) 存储在从 Ⓓ 指定的软元件开始的 n 点的低 4 位。



(2) 从 Ⓢ 指定的软元件开始的 n 点的高 12 位变为 0。

(3) 可以指定给 n 的值的范围是 1 到 4。

(4) 如果 n=0，则不执行任何运算，并且从 Ⓓ 开始的 n 点中的内容将保持不变。

运算错误

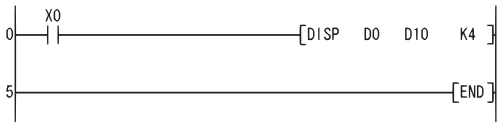
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 ⓓ 开始的 n 点范围超出了相关软元件。 (出错代码： 4101)
- n 的值超出了 0 到 4 的范围。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，以下程序将 D0 中的 16 位数据分离成 4 位一组，并存储在 D10 到 D13 中。

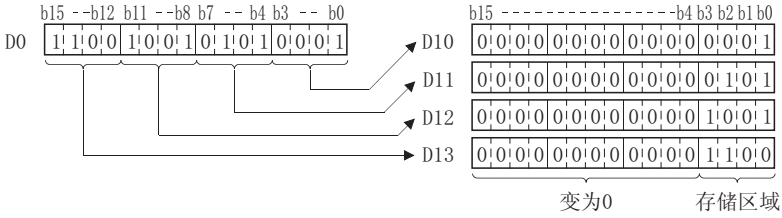
[梯形图模式]



[列表模式]

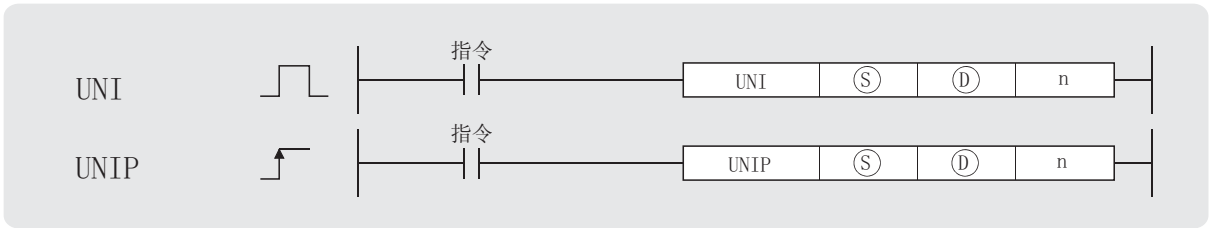
步	指令	软元件		
0	LD	X0		
1	DISP	D0	D10	K4
5	END			

[动作]



7.5.7 16 位数据的 4 位链接 (UNI(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

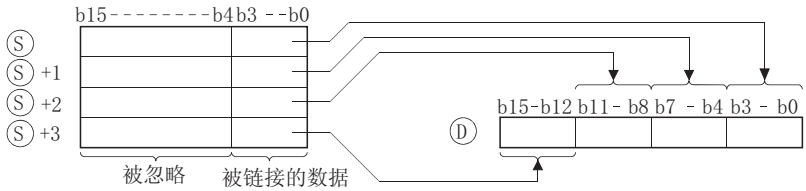


Ⓢ ：存储要链接的数据的软元件的第一个软元件号 (BIN 16 位)
Ⓓ ：存储已链接的数据的软元件的第一个软元件号 (BIN 16 位)
n ：链接数 (1 到 4)：如果链接数为 0，则不执行任何处理 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		○				---	---
n	○	○		○				○	---

功 能

(1) 将从 Ⓢ 指定的软元件开始的 n 点中的 16 位数据的低 4 位链接到 Ⓓ 指定的 16 位软元件。



(2) Ⓓ 指定的软元件的高位 (4 到 n) 数字的各位变为 0。

(3) 可以指定给 n 的值的范围是 1 到 4。

(4) 如果 n=0，则不执行任何运算，并且软元件 Ⓓ 中的内容将保持不变。

运算错误

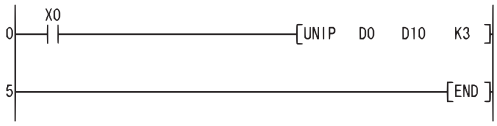
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 开始的 n 点范围超出了相关软元件。 （出错代码： 4101）
- n 的值超出了范围 0 到 4。 （出错代码： 4100）

程序示例

(1) 当 X0 为 ON 时，以下程序链接 D0 至 D2 中数据的低 4 位，并将它们存储在 D10 中。

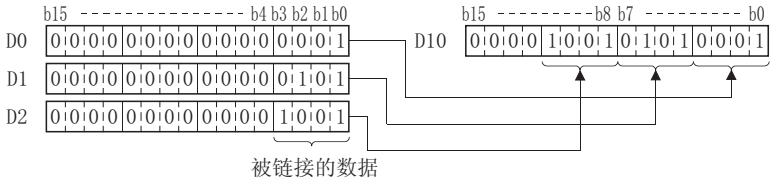
[梯形图模式]



[列表模式]

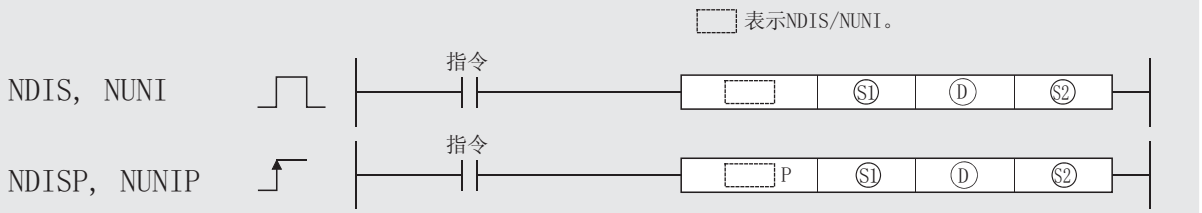
步	指令	软元件		
0	LD	X0		
1	UNIP	D0	D10	K3
5	END			

[动作]



7.5.8 任意数据的分离或链接 (NDIS(P)、NUNI(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)：存储要分离或链接的数据的软元件的第一个软元件号 (BIN 16 位)

(D)：存储已分离或链接的数据的软元件的第一个软元件号 (BIN 16 位)

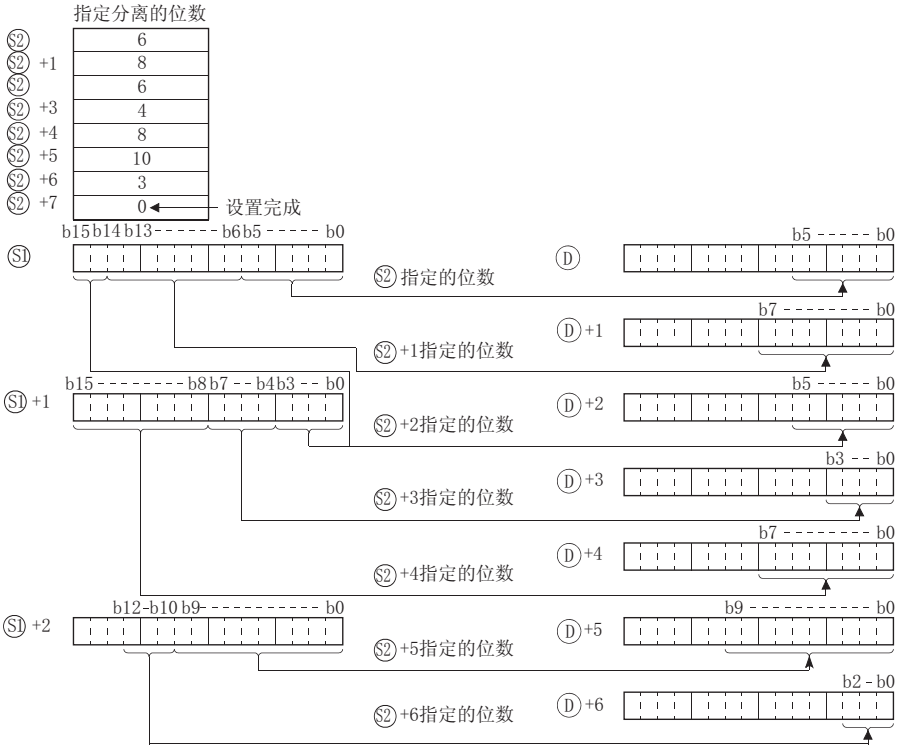
(S2)：存储进行分离或链接时将使用的单位的软元件的第一个软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	---	○		---					
(D)	---	○		---					
(S2)	---	○		---					

功 能

NDIS

(1) 将从 (S1) 指定的软元件开始的软元件中存储的数据分离成 (S2) 指定的位数，并将该数据存储在从 (D) 指定的软元件开始的软元件中。



(2) 可以指定给 ⓢ2 的分离位数的范围是 1 到 16 位。

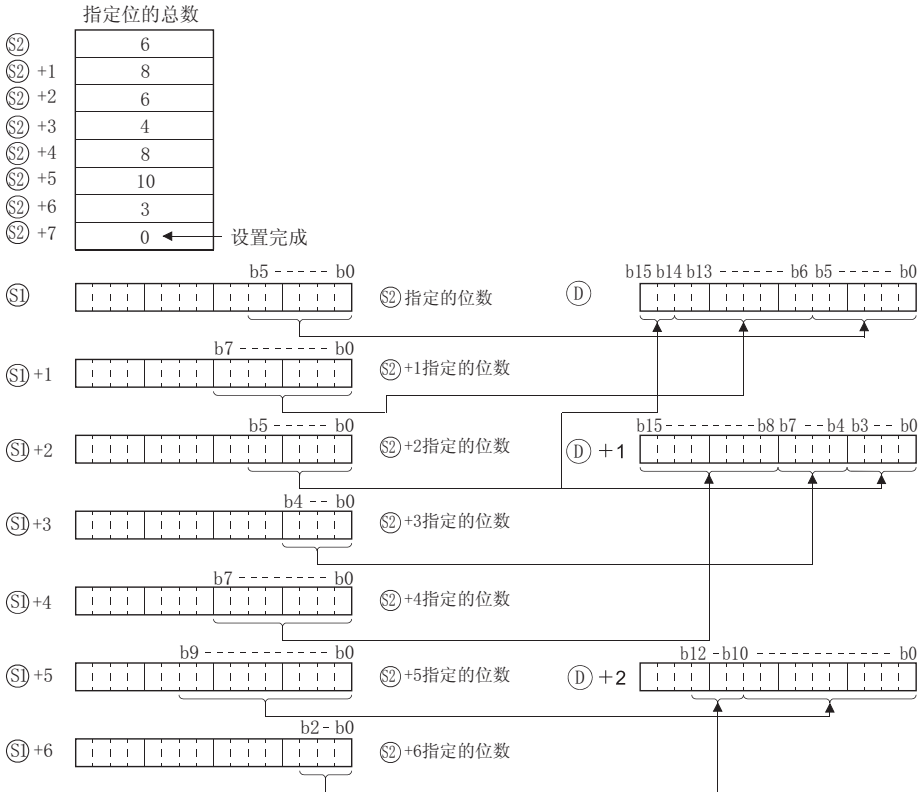
(3) 从 ⓢ2 指定的软元件号到存储“0”的软元件号之间的位都被作为分离位进行处理。

(4) 不要让分离数据的软元件范围（ⓢ1 到 ⓢ1）的结束范围）与存储分离数据的软元件范围（ⓓ 到 ⓓ）的结束范围）相互重叠。如果重叠，可能无法得到正确的运算结果。

(5) 不要为 ⓢ1、ⓢ2 和 ⓓ 指定相同的软元件号，如果为 ⓢ1、ⓢ2 和 ⓓ 指定了相同的软元件号，将无法正确进行运算。

NUNI

(1) 按照 ⓢ2 指定的位数链接存储在从 ⓢ1 指定的软元件开始的软元件中的数据的单个位，并将它们存储在从 ⓓ 指定的软元件开始的软元件中。



(2) 可以指定给 ⓢ2 的链接位数的范围是 1 到 16。

(3) 将从 ⓢ2 指定的软元件号到存储“0”的软元件号之间要被链接的位的数目进行处理。

(4) 不要让链接数据的软元件范围（ⓢ1 到 ⓢ1）的结束范围）与存储链接数据的软元件范围（ⓓ 到 ⓓ）的结束范围）相互重叠。如果重叠，可能无法得到正确的运算结果。

(5) 不要使 ⓢ1、ⓢ2 和 ⓓ 指定的软元件号相互重叠，如果重叠，将无法进行正确运算。

运算错误

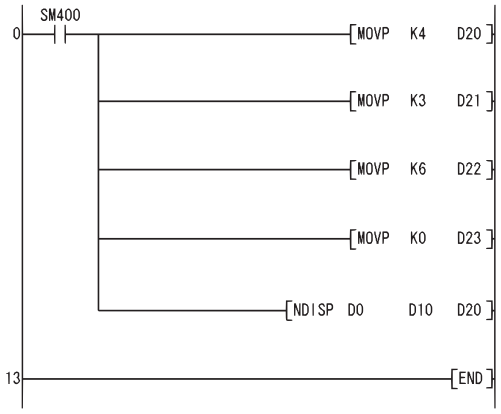
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- ⓢ2 指定的分离或链接位数，或者 ⓢ1 或 ⓓ 指定的软元件的使用范围超出了它们各自软元件的最后一个软元件号。 （出错代码： 4101）
- ⓢ2 指定的分离或链接位数不在范围 1 到 16 之内。 （出错代码： 4100）

程序示例

(1) 以下程序从 D0 中数据的低位开始分别分离出 4 位、3 位和 6 位数据，并将它们存储在 D10 到 D12 中。

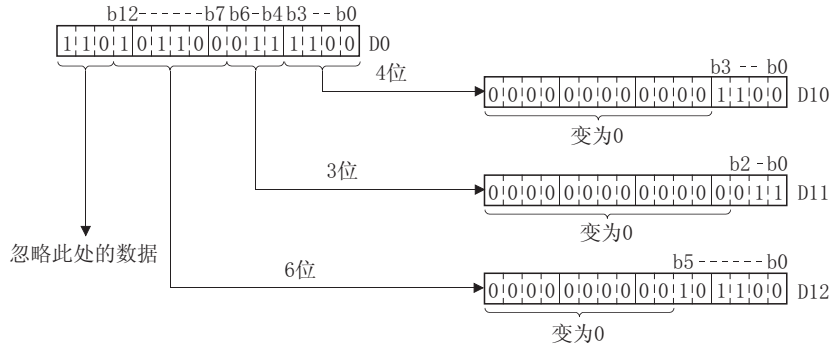
[梯形图模式]



[列表模式]

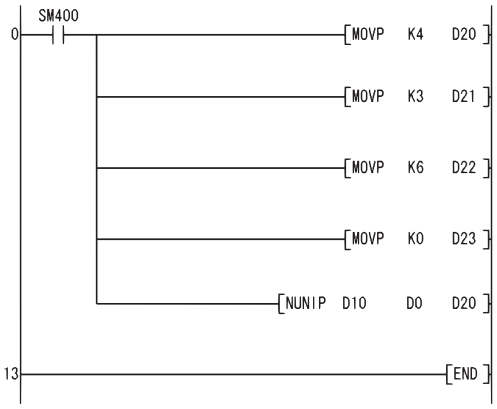
步	指令	软元件		
0	LD	SM400		
1	MOVP	K4	D20	
3	MOVP	K3	D21	
5	MOVP	K6	D22	
7	MOVP	K0	D23	
9	NDISP	D0	D10	D20
13	END			

[动作]



(2) 以下程序链接 D10 中数据的低 4 位，D11 中数据的低 3 位和 D12 中数据的低 6 位，并将它们存储在 D0 中。

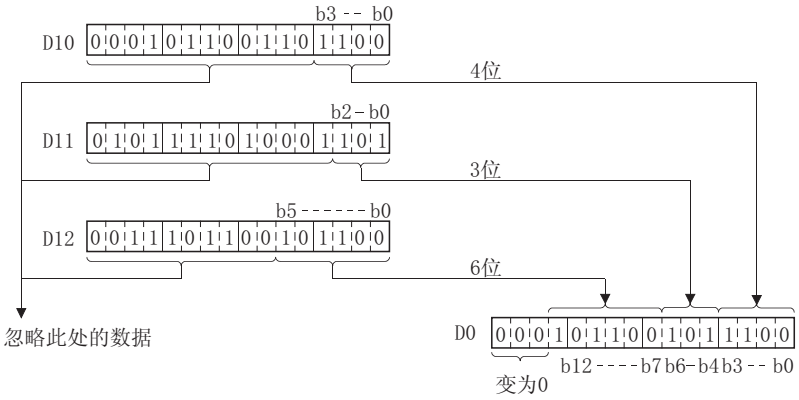
[梯形图模式]



[列表模式]

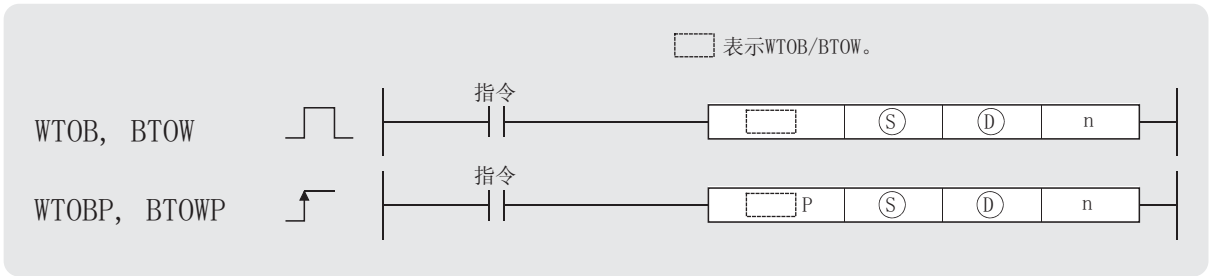
步	指令	软元件		
0	LD	SM400		
1	MOVP	K4	D20	
3	MOVP	K3	D21	
5	MOVP	K6	D22	
7	MOVP	K0	D23	
9	NUNIP	D10	D0	D20
13	END			

[动作]



7.5.9 以字节为单位的数据分离和链接(WTOB(P)、BTOW(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 存储需要以字节为单位进行分离或链接的数据的软元件的第一个软元件号(BIN 16位)

Ⓓ : 存储以字节为单位进行分离或链接的结果的软元件的第一个软元件号(BIN 16位)

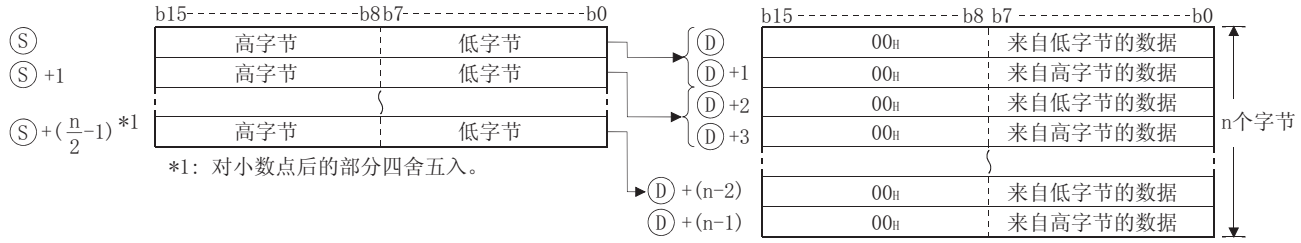
n : 要分离或链接的字节数(BIN 16位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

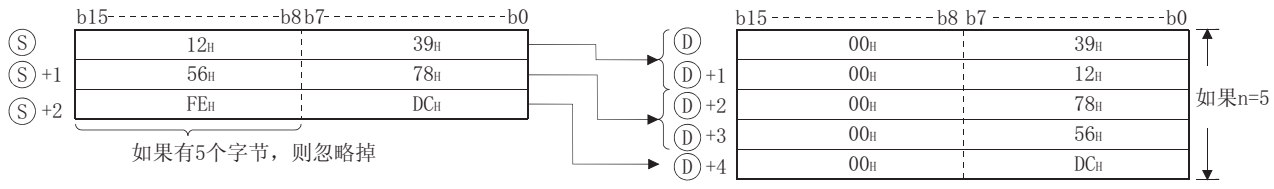
功 能

WTOB

(1) 对存储在 Ⓢ 指定的软元件之后的16位数据的n个字节进行分离，并将它们存储在 Ⓓ 指定的软元件之后的软元件中。



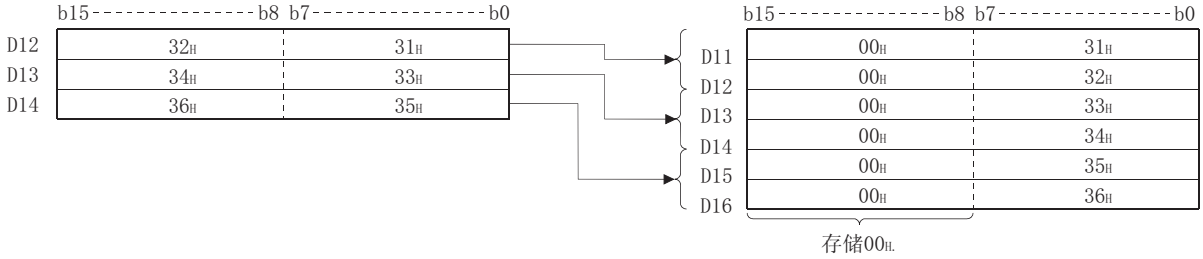
例如，如果n=5，则 Ⓢ 的低8位到(Ⓢ+2)中数据将被存储在(Ⓓ 到 Ⓓ+4)中。



(2) 通过 n 对字节数的设定将自动确定 Ⓢ 指定的 16 位数据的范围以及用于存储 Ⓓ 指定的字节数据的软元件的范围。

(3) 当 n 指定的字节数为 “0”时，不执行任何处理。

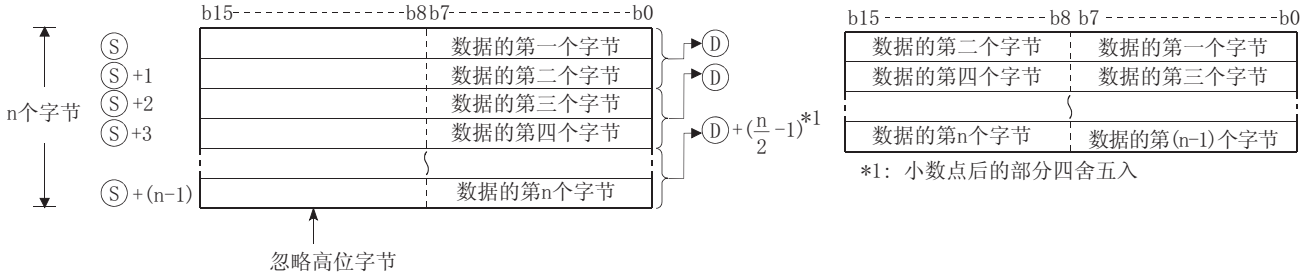
(4) 代码 “00H ”将自动存储在 Ⓓ 指定的字节存储软元件的高 8 位中。



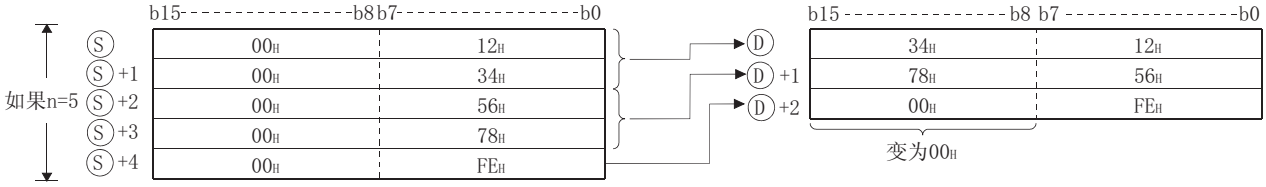
(5) 即使存储分离数据的软元件范围（Ⓢ 到 Ⓢ+($\frac{n}{2}$-1)）与存储分离数据的软元件范围（Ⓓ 到 Ⓓ+(n-1)）重叠，指令仍会正常执行。

BTOW

(1) 以字为单位链接从 Ⓢ 指定的软元件开始的软元件中的 n 字 16 位数据的低 8 位，并存储在从 Ⓓ 指定的软元件开始的软元件中。
忽略从 Ⓢ 指定的软元件开始的软元件中的 n 字数据的高 8 字节。
另外，如果 n 是奇数，那么存储数据的第 n 个字节的软元件的高 8 位将存储为 0。



例如，如果 n=5，则链接 Ⓢ 到 (Ⓢ+4) 中数据的低 8 位，且存储在 Ⓓ 到 (Ⓓ+2) 中。



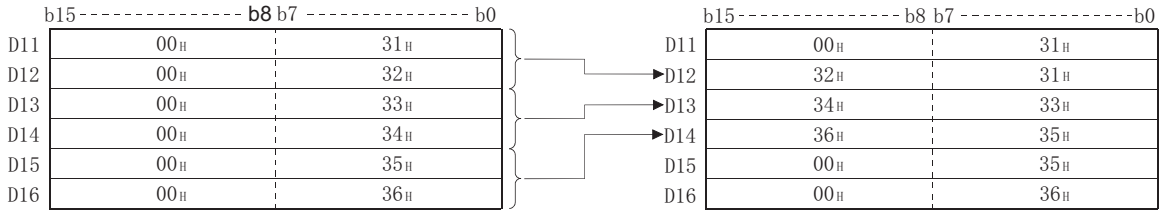
(2) 通过 n 对字节数的设定将自动确定 Ⓢ 指定的 16 位数据的范围以及用于存储 Ⓓ 指定的字节数据的软元件的范围。

(3) 当 n 指定的字节数为 “0”时，不执行任何处理。

(4) 忽略 Ⓢ 指定的字节存储软元件的高 8 位，而使用其低 8 位。

(5) 即使在如下情况中，也会正常执行的处理：存储链接（合并）数据的软元件范围（Ⓢ 到 Ⓢ+(n-1)）与存储被链接（合并）数据的软元件范围（Ⓓ 到 Ⓓ+($\frac{n}{2}$-1)）相互之间有重叠。

例如：将 D11 到 D16 的低 8 位存储到 D12 到 D14 中时，将发生下列情况。



运算错误

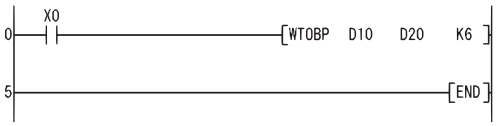
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 存储在从 Ⓢ 指定的软元件之后的软元件中，由 n 指定的字节数范围超出了相关软元件范围。（出错代码：4101）
- 存储在从 Ⓓ 指定的软元件之后的软元件中，由 n 指定的字节数范围超出了相关软元件范围。（出错代码：4101）

程序示例

(1) 当 X0 为 ON 时，以下程序以字节为单位分离 D10 到 D12 中的数据，并存储在 D20 到 D25 中。

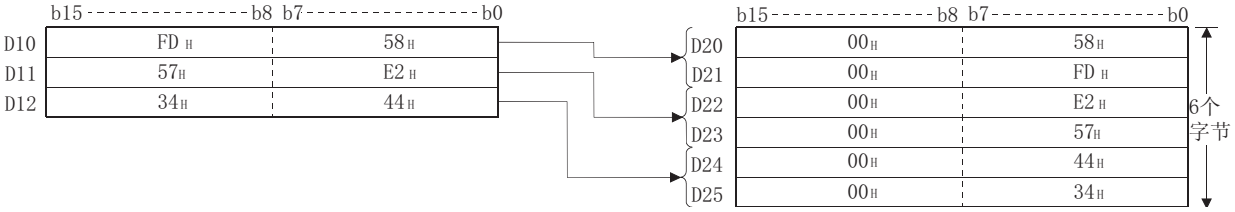
[梯形图模式]



[列表模式]

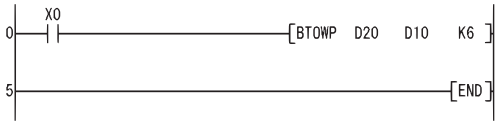
步	指令	软元件		
0	LD	X0		
1	WTOBP	D10	D20	K6
5	END			

[动作]



(2) 当 X0 为 ON 时，以下程序链接 D20 到 D25 中数据的低 8 位，并将结果存储在 D10 到 D12 中。

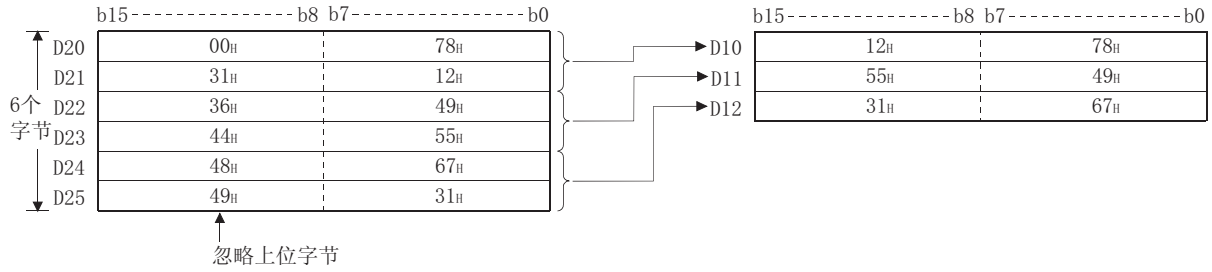
[梯形图模式]



[列表模式]

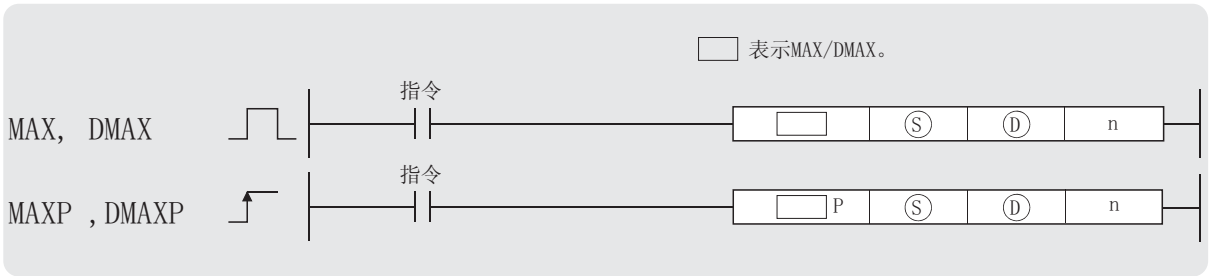
步	指令	软元件		
0	LD	X0		
1	BTOWP	D20	D10	K6
5	END			

[动作]



7.5.10 16 位和 32 位数据的最大值查找 (MAX(P)、DMAX(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：用于查找最大值的软元件的起始号 (BIN 16/32 位)

Ⓓ ：用于存储最大值查找结果的软元件的起始号 (BIN 16/32 位)

n ：查找的数据块数 (BIN 16 位)

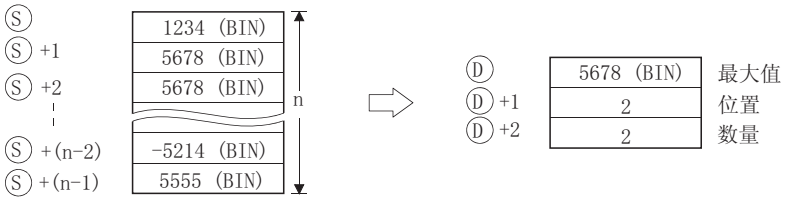
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功 能

MAX

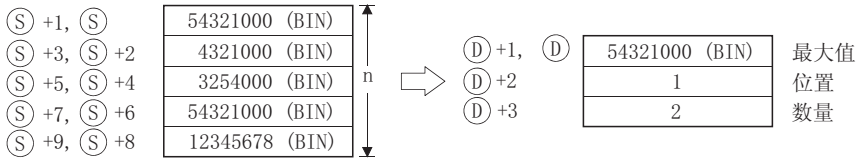
(1) 从 Ⓢ 指定的软元件开始的 n 块 16 位数据中查找最大值，并将该值存储在 Ⓓ 指定的软元件中。

从 Ⓢ 指定的软元件开始，查找到存储已发现的第一个最大值的软元件号，在 Ⓓ+1 中存储该软元件号距离软元件 Ⓢ 的点数，在 Ⓓ+2 中存储最大值的数目。



DMAX

(1) 从 Ⓢ 指定的软元件开始的 n 块 32 位数据中查找最大值，并将该值存储在 Ⓓ 和 Ⓓ+1 指定的软元件中。
从 Ⓢ 指定的软元件开始，在查找到存储着已发现的第一个最大值的软元件号时，在 Ⓓ+2 中存储该软元件号距离软元件 Ⓢ 的点数，在 Ⓓ+3 中存储最大值的数目。



运算错误

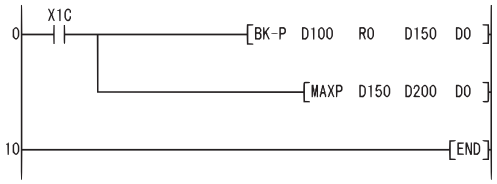
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 指定的软元件开始的 n 点范围超出了相关软元件范围。
（出错代码： 4101）
- Ⓓ 指定的软元件超过了相应软元件的范围。
（只对于通用型 QCPU） （出错代码： 4101）
-

程序示例

(1) 当 X1C 变为 ON 时，以下程序将从 D100 开始的软元件（软元件的个数由存储在 D0 中的值指定）中的数据中减去从 R0 开始的软元件（软元件的个数由存储在 D0 中的值指定）中的数据，再从该计算的结果中查找最大值，并将它存储在 D200 到 D202 中。

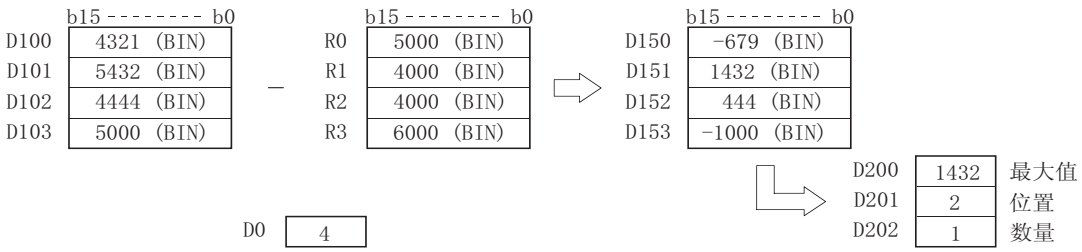
［梯形图模式］



［列表模式］

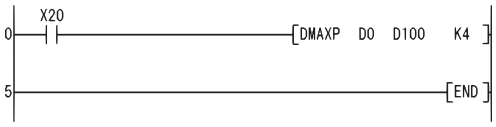
步	指令	软元件			
0	LD	X1C			
1	BK-P	D100	R0	D150	D0
6	MAXP	D150	D200	D0	
10	END				

［动作］



(2) 当 X20 变为 ON 时，以下程序从 D0 到 D7 中的 32 位数据中查找最大值，并将它存储在 D100 到 D103 中。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X20		
1	DMAXP	D0	D100	K4
5	END			

[动作]

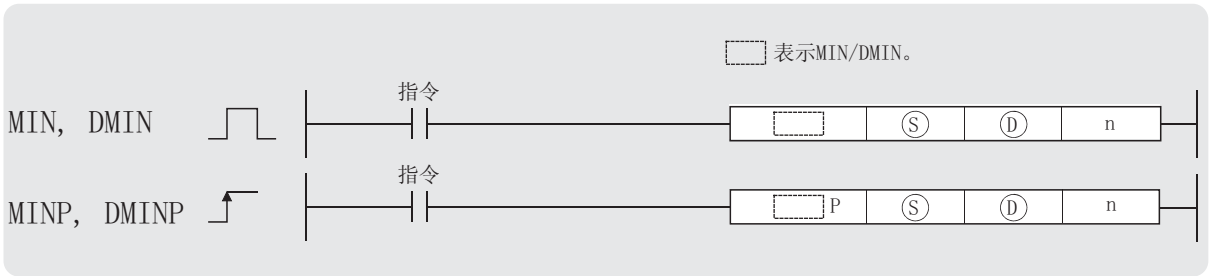
D1, D0	3786213 (BIN)
D3, D2	-3235 (BIN)
D5, D4	8744740 (BIN)
D7, D6	7141821 (BIN)

⇨

D101, D100	8744740
D102	3
D103	1

7.5.11 查找 16 位和 32 位数据的最小值 (MIN(P)、DMIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：用于查找最小值的软元件的起始号 (BIN 16/32 位)

Ⓓ ：用于存储最小值查找结果的软元件的起始号 (BIN 16/32 位)

n ：查找的数据块数 (BIN 16 位)

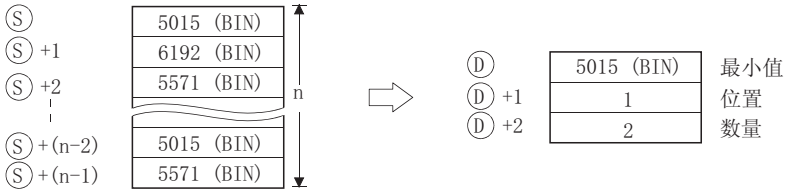
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---					---
Ⓓ	---	○		---					---
n	○	○		○					---

功 能

MIN

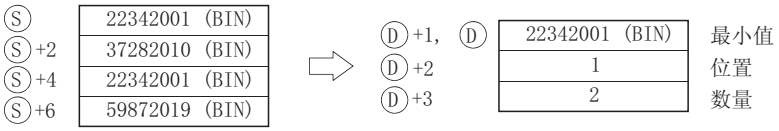
(1) 从 Ⓢ 指定的软元件开始的 n 块 16 位 BIN 数据中查找最小值，并将该值存储在 Ⓓ 指定的软元件中。

从 Ⓢ 指定的软元件开始，在查找到存储着已发现的第一个最小值的软元件号时，在 Ⓓ+1 中存储该软元件号距离软元件 Ⓢ 的点数，在 Ⓓ+2 中存储最小值的数目。



DMIN

(1) 从 ⓢ 指定的软元件开始的 n 块 32 位 BIN 数据中查找最小值，并将该值存储在 ⓓ 和 ⓓ+1 指定的软元件中。
从 ⓢ 指定的软元件开始，在查找到存储着已发现的第一个最小值的软元件号时，在 ⓓ+2 中存储该软元件号距离软元件 ⓢ 的点数，在 ⓓ+3 中存储最小值的数目。



运算错误

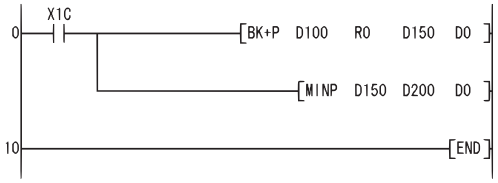
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 ⓢ 指定的软元件开始的 n 点范围超出了相关软元件。 （出错代码： 4101）
- ⓓ 指定的软元件超过了相应软元件的范围。
（只对于通用型 QCPU） （出错代码： 4101）

程序示例

(1) 当 X1C 变为 ON 时，以下程序将从 D100 开始的软元件中的数据（数据的点数由存储在 D0 中的值指定）加到从 R0 开始的软元件中的数据（数据的点数由存储在 D0 中的值指定）上，再从该计算的结果中查找最小值，并将它存储在 D200 到 D202 中。

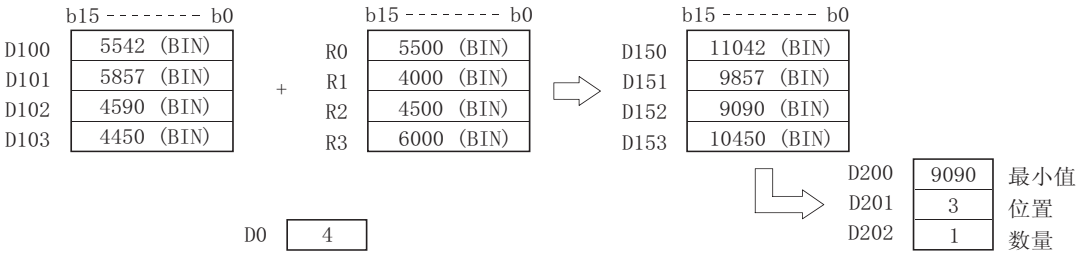
[梯形图模式]



[列表模式]

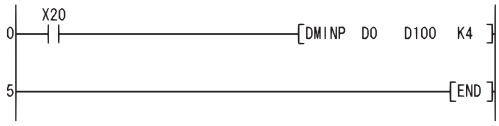
步	指令	软元件			
0	LD	X1C			
1	BK+P	D100	R0	D150	D0
6	MINP	D150	D200	D0	
10	END				

[动作]



(2) 当 X20 变为 ON 时，以下程序从 D0 到 D7 中的 32 位数据中查找最小值，并将它存储在 D100 到 D103 中。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X20		
1	DMINP	D0	D100	K4
5	END			

[动作]

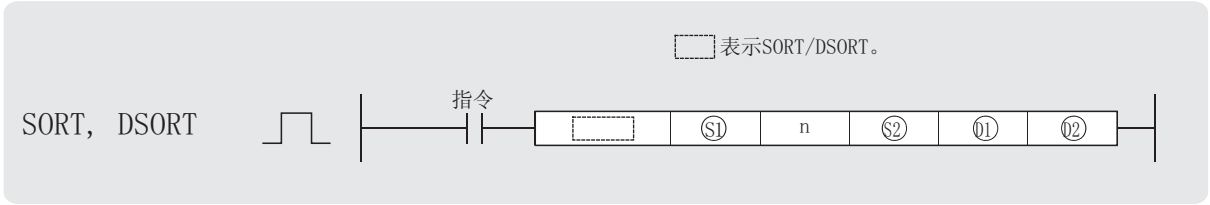
D1, D0	57020175 (BIN)
D3, D2	2070166 (BIN)
D5, D4	3596045 (BIN)
D7, D6	− 69386 (BIN)

⇨

D101, D100	− 69386
D102	4
D103	1

7.5.12 BIN16 位和 32 位数据的排序运算 (SORT、DSORT)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：将进行排序的表的起始软元件号 (BIN 16/32 位)
n ：要排序的数据块数 (BIN 16 位)
(S2) ：在一次排序运算中用于比较的数据块数 (BIN 16 位)
(D1) ：排序完成后被强制为 ON 的位软元件号 (位)
(D2) ：系统使用的软元件 (BIN 16 位)

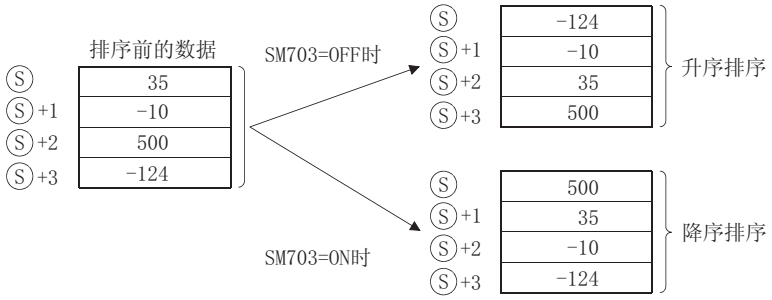
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	---	○		---					---
n	○	○		○					---
(S2)	○	○		○					---
(D1)	○	---		---					---
(D2)	---	○		---					---

功 能

SORT

(1) 对从 (S1) 开始的 n 点的 BIN16 位数据按升序 / 降序进行排序 (排列切换)。
排序方式通过 SM703 的 ON/OFF 进行指定：

- 当 SM703 为 OFF 时 ： 升序排序
- 当 SM703 为 ON 时 ： 降序排序



(2) 用 SORT 指令执行排序需要若干个扫描周期。
在排序结束时共执行的最大次数除以执行一次排序运算所比较的数据块数（由 (S2) 指定），所得到的结果值就是到排序结束共执行的扫描周期数。（小数部分四舍五入）。
如果增大 (S2) 的值，则在排序结束时扫描周期数减少，但每次扫描的时间变长。

(3) 至排序结束时执行的最多次数应通过下列公式计算：
至排序结束时执行的最多次数 =(n) × (n-1)/2[执行次数]

例如
如果 n=10，则完成排序将需要 10 × (10-1)/2=45[执行次数]。
如果上例中的 (S2) 等于 2，则至排序结束时的扫描数将为 45/2=22.5 → 23[扫描]。

(4) (D1)（完成软元件）指定的软元件在 SORT 指令执行前为 OFF，而在排序结束时为 ON。
因为在排序结束后，(D1) 指定的软元件保持为 ON 状态，所以如果需要，用户必须将它变为 OFF。

(5) 在 SORT 指令的执行期间，从 (D2) 指定的软元件开始的 2 点为系统所使用。
因此，用户不要更改 (D2) 指定的软元件开始的 2 点。
改变此 2 点可能导致出错代码返回 。 （出错代码： 4100).

(6) 如果在 SORT 指令的执行期间更改了 n 的值，则排序运算将按照更改后的排序数据块数来执行。

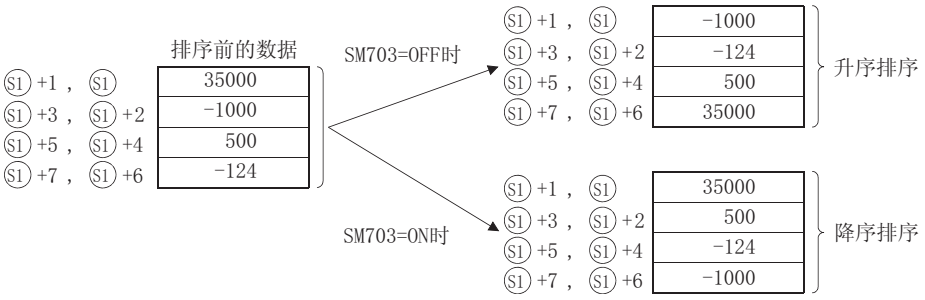
(7) 如果在执行排序运算的过程中命令变为 OFF，则排序运算将中止。
当命令重新为 ON 时，排序将从头开始。

(8) 希望排序结束时立即进行下一个排序的情况下，需先将执行指令置为 OFF，然后再 ON。

DSORT

(1) 对从 (S1) 开始的 n 点的 BIN32 位数据按升序或者降序进行排序（重新排列数据）。
排序方式通过 SM703 的 ON/OFF 进行指定：

- 当 SM703 的状态为 OFF 时 ： 升序排序
- 当 SM703 的状态为 ON 时 ： 降序排序



(2) 用 DSORT 指令执行排序需要若干个扫描周期。
在排序结束时共执行的最大次数除以执行一次排序运算所比较的数据块数（由 (S2) 指定），所得到的结果值就是到排序结束共执行的扫描周期数。（小数部分四舍五入）。
如果增大 (S2) 的值，则在排序结束时扫描周期数减少，但每次扫描的时间变长。

(3) 至排序结束时执行的最多次数应通过下列公式计算：
排序结束时执行的最多次数 =(n) × (n-1)/2[执行次数]

例如

如果 n=10，则完成排序需要 10 × (10-1)/2=45 个扫描。
如果上例中的 (S2) 等于 2，则排序结束时的扫描数将为 45/2=22.5 → 23[扫描]。

(4) (D1)（完成软元件）指定的软元件在 SORT 指令执行前为 OFF，而在排序结束时为 ON。
因为在排序结束后，(D1) 指定的软元件保持为 ON 状态，所以如果需要，用户必须将它变为 OFF。

(5) 在 DSORT 指令的执行期间，从 (D2) 指定的软元件开始的 2 点为系统所使用。
因此，用户不应更改从 (D2) 指定的软元件开始的 2 点。
改变此 2 点可能导致出错代码返回 。（出错代码： 4100).

(6) 如果在 DSORT 指令的执行期间更改了 n 的值，则排序运算将按照更改后的排序数据块数来执行。

(7) 如果在执行排序运算的过程中将命令为 OFF，则排序运算将中止。
当命令重新为 ON 时，排序将从头开始。

(8) 希望在排序结束时立即执行下一个排序的情况下，需要先将执行指令置为 OFF，然后再 ON。

运算错误

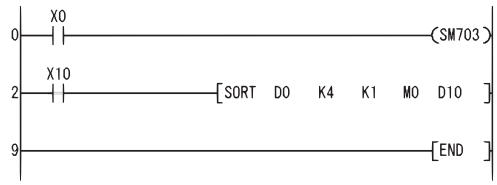
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 在 SORT(P) 指令中，从 (S1) 指定的软元件开始的 n 点的范围超出了相应软元件范围。
（出错代码： 4101)
- 在 DSORT(P) 指令中，从 (S1) 指定的软元件开始的 2 × n 点的范围超出了相应软元件范围。
（出错代码： 4101)
- 从 (S1) 指定的软元件开始的 (n/2 × n) 点的范围与从 (D2) 指定的软元件开始的 2 点的范围相互重叠。
（出错代码： 4101)
- (S2) 为 0 或负的值。（出错代码： 4100)

程序示例

(1) 当 X10 为 ON 时，以下程序对从 D0 开始的 10 点的 BIN16 位数据进行升序 / 降序排序。

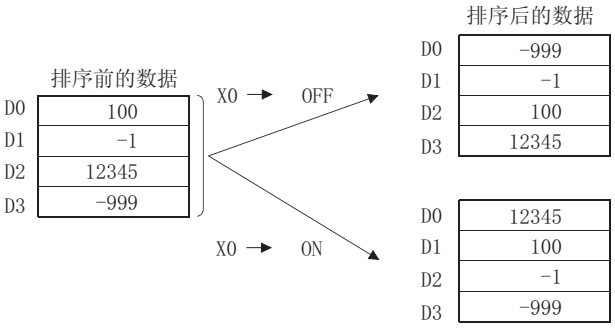
[梯形图模式]



[列表模式]

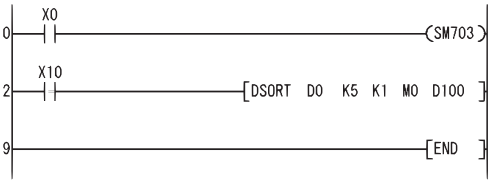
步	指令	软元件
0	LD	X0
1	OUT	SM703
2	LD	X10
3	SORT	D0 K4 K1 M0 D10
9	END	

[动作]



(2) 当 X10 为 ON 时，以下程序对从 D0 开始的 20 点的 BIN32 位数据进行升序 / 降序排序。

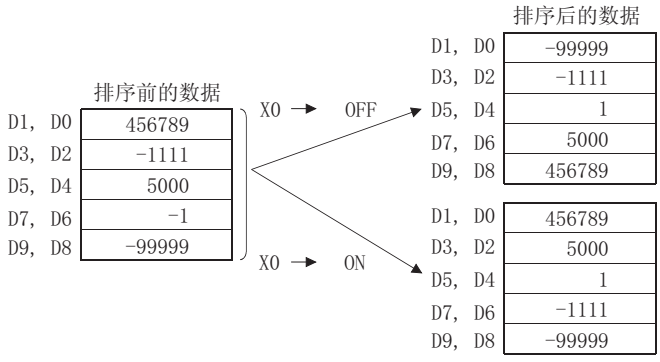
[梯形图模式]



[列表模式]

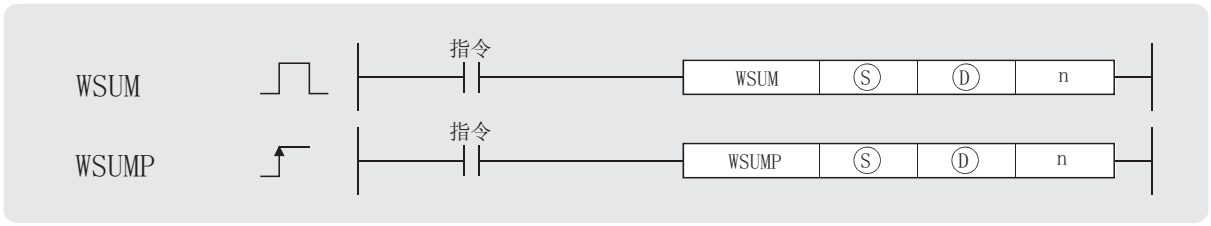
步	指令	软元件
0	LD	X0
1	OUT	SM703
2	LD	X10
3	DSORT	D0 K5 K1 M0 D100
9	END	

[动作]



7.5.13 16 位数据的总和计算 (WSUM(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储参与总和计算的数据的软元件的起始号 (BIN 16 位)

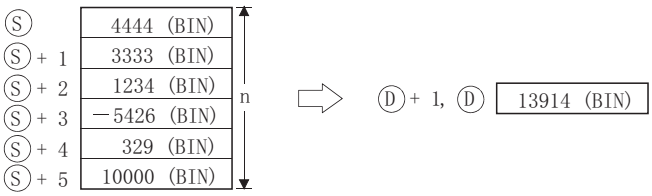
Ⓓ ：存储总和值的软元件的起始号 (BIN 32 位)

n ：数据块数 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		○				---	---
n	○	○		○				○	---

功 能

(1) 将从 Ⓢ 指定的软元件开始的 n 块 16 位 BIN 数据全都加起来，并把结果存储在 Ⓓ 指定的软元件中。



运算错误

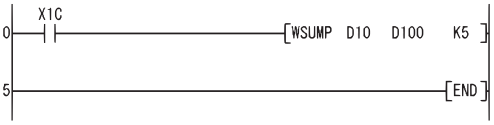
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 从 Ⓢ 软元件开始的 n 点范围超出了相关软元件。 (出错代码： 4101)

程序示例

(1) 当 X1C 变为 ON 时，以下程序将 D10 到 D14 中的 16 位 BIN 数据加起来，并存储在 D100 和 D101 中。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X1C		
1	WSUMP	D10	D100	K5
5	END			

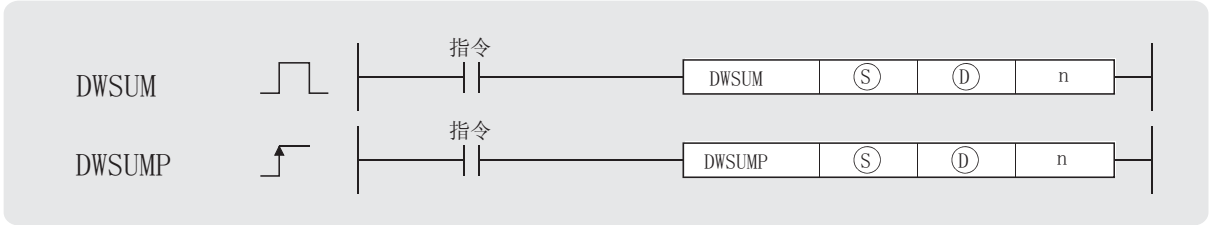
[动作]

D10	4500 (BIN)
D11	2500 (BIN)
D12	−3276 (BIN)
D13	6780 (BIN)
D14	4444 (BIN)

⇨ D101,D100 | 14948 (BIN) |

7.5.14 32 位数据总和计算 (DWSUM(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

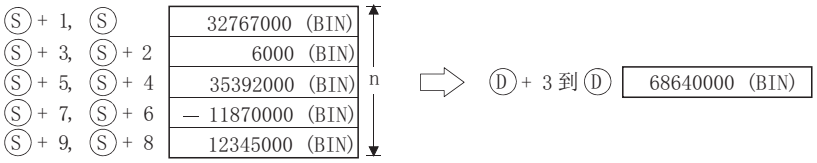


Ⓢ ：存储用于总和计算的数据的软元件的起始软元件号 (BIN 32 位)
Ⓓ ：存储总和计算结果的软元件的起始软元件号 (BIN 64 位)
n ：数据块的数目 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		---				---	---
n	○	○		○				○	---

功 能

(1) 从 Ⓢ 指定的软元件开始的全部 4 点 (4 字)32 位 BIN 数据相加，并将结果存储在 Ⓓ 指定的软元件上。



运算错误

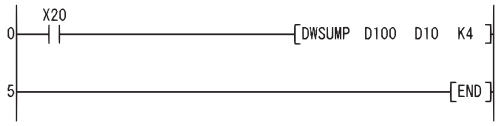
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 软元件开始的 n 点的范围超过了相关软元件。
  (出错代码： 4101)
- Ⓓ 指定的软元件超过了相应软元件的范围。
  (只对于通用型 QCPU)　(出错代码： 4101)

程序示例

(1) 当 X20 为 ON 时，以下程序将 D100 到 D107 的 32 位 BIN 数据相加，并将结果存储在 D10 到 D13 上。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X20		
1	DWSUMP	D100	D10	K4
5	END			

[动作]

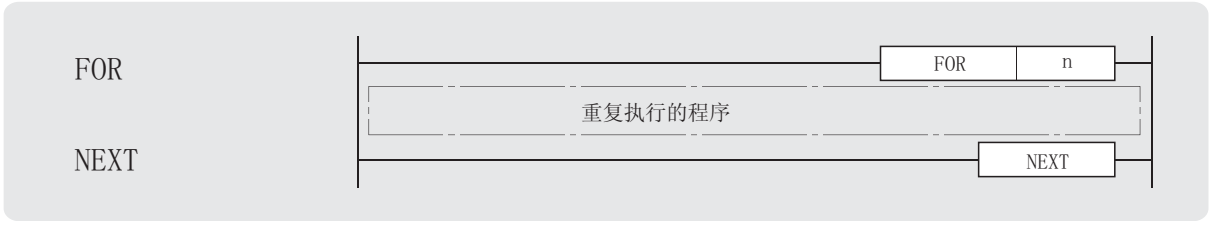
D101,D100	11245600 (BIN)
D103,D102	27543200 (BIN)
D105,D104	558800 (BIN)
D107,D106	−15675000 (BIN)

⇨ D13 到 D10 | 23672600 (BIN) |

7.6 结构化程序指令

7.6.1 FOR 到 NEXT 指令循环 (FOR、NEXT)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n ：FOR 到 NEXT 循环的重复次数 (1 到 32767)。(BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n		○							---

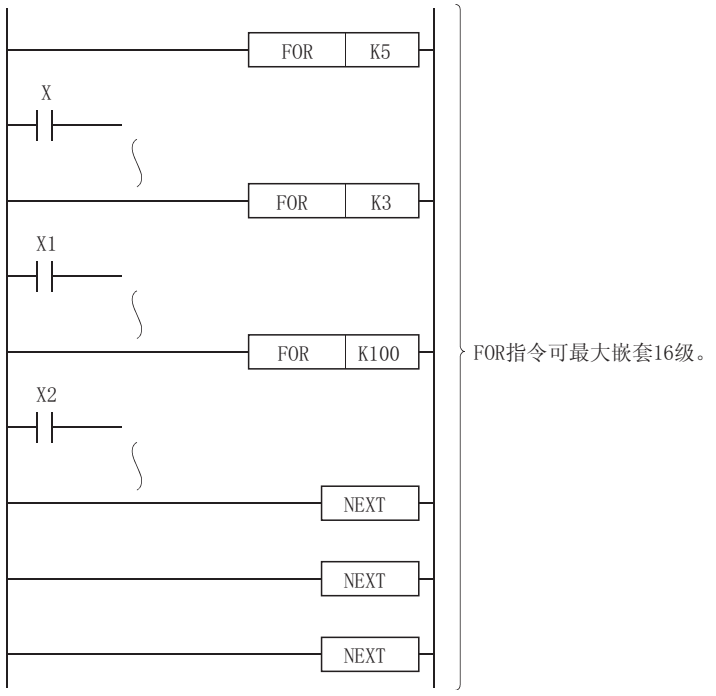
功 能

(1) 当 FOR 到 NEXT 循环中的处理无条件执行了 n 次时，将执行跟着 NEXT 指令之后的步。

(2) n 的值可指定在 1 到 32767 之间。如果指定为 -32768 到 0 之间的一个值，则当作 n=1 执行。

(3) 如果不想执行在 FOR 到 NEXT 循环内调用的处理，则使用 CJ 或 SCJ 指令跳过去。

(4) FOR 指令可最大嵌套 16 级。



运算错误

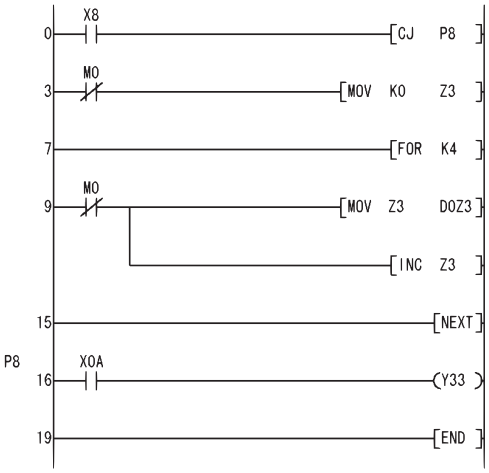
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 在一个 FOR 指令执行之后，但在一个 NEXT 指令执行之前，执行了一个 END(FEND) 指令。 (出错代码： 4200)
- 在一个 FOR 指令执行之前，执行了一个 NEXT 指令。 (出错代码： 4201)
- 已经在 FOR 到 NEXT 循环内插入一个 STOP 指令。 (出错代码： 4200)
- 当已经嵌套使用 FOR 指令时，遇到第 17 个 FOR 指令。 (出错代码： 4202)

程序示例

(1) 当 X8 为 OFF 时，以下程序执行 FOR 到 NEXT 循环，当 X8 为 ON 时，不执行。

[梯形图模式]

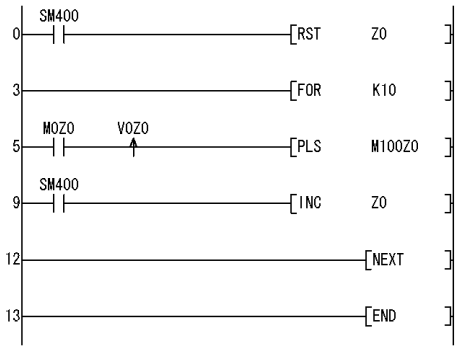


[列表模式]

步	指令	软元件	
0	LD	X8	
1	CJ	P8	
3	LDI	M0	
4	MOV	K0	Z3
7	FOR	K4	
9	LDI	M0	
10	MOV	Z3	DOZ3
13	INC	Z3	
15	NEXT		
16	P8		
17	LD	X0A	
18	OUT	Y33	
19	END		

备 注

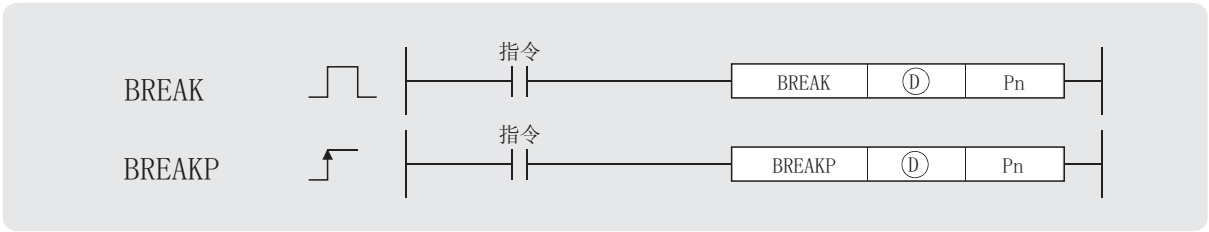
1. 为了在 FOR 到 NEXT 循环执行期间强制结束该循环的重复执行，请插入一个 BREAK 指令。有关 BREAK 指令使用的详细内容请参见 7.6.2 项。
2. 在 FOR 和 NEXT 之间对进行了变址修饰的程序进行脉冲运算时，应使用 EGP/EGF 指令。关于 EGP/EGF 指令的详情请参见 5.2.5 项。程序示范如下：



3. 在 FOR 到 NEXT 循环之外无法使用 JMP 或其它分支指令对 FOR 到 NEXT 循环执行分支操作。

7.6.2 FOR 到 NEXT 指令循环的强制结束 (BREAK(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

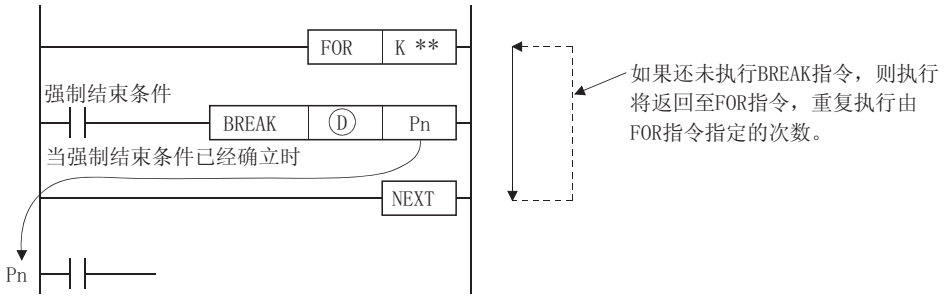


Ⓓ : 存储剩余重复次数的软元件的号码 (BIN 16 位)
Pn : 当重复处理结束时的分支目标指针号 (软元件名称 (指针))

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P
Ⓓ	○							---	---
Pn	---							---	○

功 能

(1) 强制结束 FOR 到 NEXT 指令循环的重复处理，将执行切换到 Pn 指定的指针处。对 Pn 只能指定同一程序文件中的指针。如果指定了其它程序文件中的指针，将发生运算错误。



(2) 将在 FOR 到 NEXT 循环被强制结束时剩余的重复次数存储到 Ⓓ 上。
但是，执行 BREAK 指令时的循环也包括在剩余的重复次数中。

(3) 仅在 FOR 到 NEXT 指令循环执行期间才可使用 BREAK 指令。

(4) 仅当只有一级嵌套时才可使用 BREAK 指令。
如果当有多重嵌套时强制结束，则执行同嵌套级数相同数量的 BREAK 指令。

运算错误

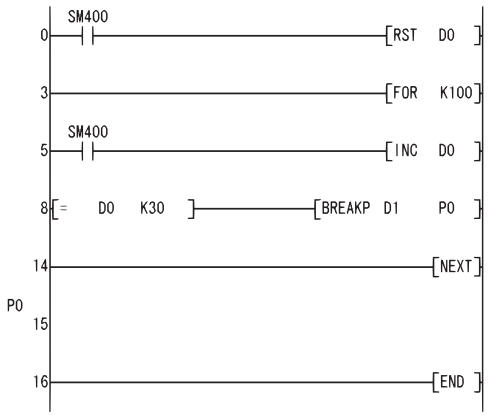
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 在一个非 FOR 到 NEXT 的指令循环内使用 BREAK 指令。 (出错代码： 4203)
- Pn 指定的指针的跳转目的地不存在。 (出错代码： 4210)
- 另一个程序文件的指针被指定给 Pn。 (出错代码： 4210)

程序示例

(1) 当 D0 值到达 30 时 (当 FOR 到 NEXT 循环被执行 30 次时)，以下程序强制结束 FOR 到 NEXT 循环。

[梯形图模式]



[列表模式]

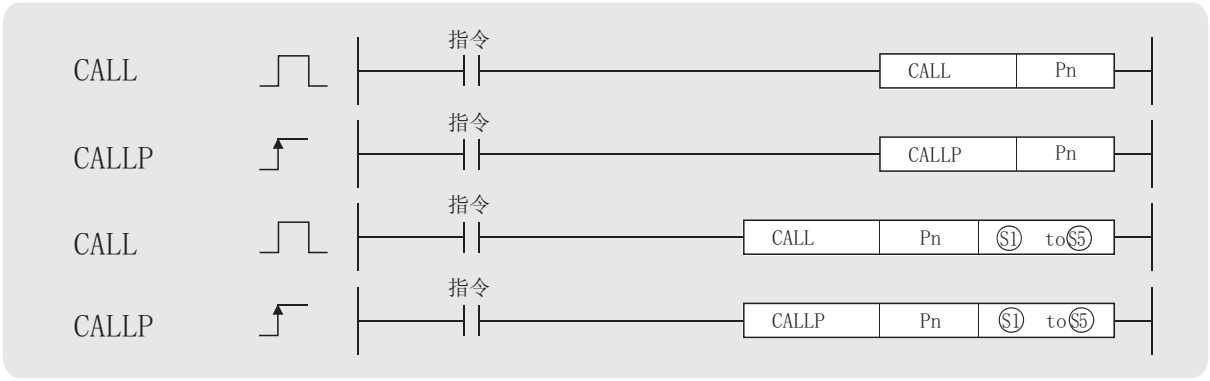
步	指令	软元件	
0	LD	SM400	
1	RST	D0	
3	FOR	K100	
5	LD	SM400	
6	INC	D0	
8	LD=	D0	K30
11	BREAKP	D1	P0
14	NEXT		
15	P0		
16	END		

备 注

当执行 BREAK 指令时，将值 71 存储在 D1 上。

7.6.3 子程序调用 (CALL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Pn　　：子程序的起始指针号（软元件名称）

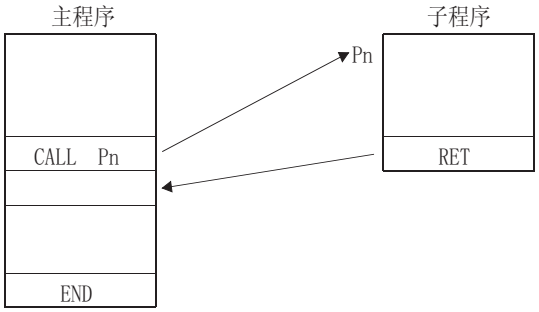
ⓢ1 到 ⓢ5 ：作为变量传送到子程序的软元件号（位、BIN 16 位、BIN 32 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它 P
	位	字		位	字				
Pn	---	---		---					○
ⓢ1 到 ⓢ5	○（除 F 之外）	○		○					---

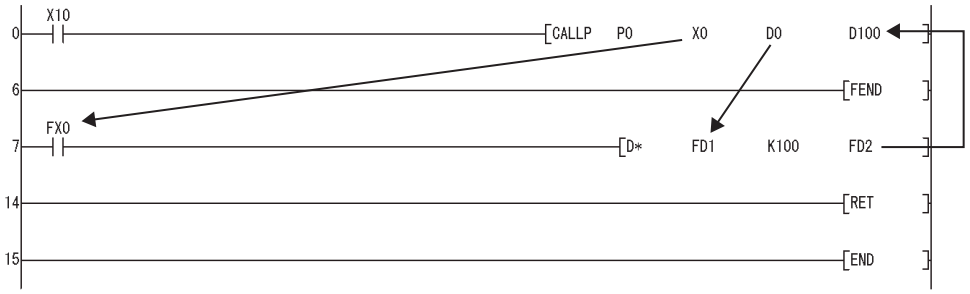
功 能

(1) 当执行 CALL(P) 指令时，执行由 Pn 指定的程序的子程序。

CALL(P) 指令可以执行由同一个程序文件中的一个指针指定的子程序和由一个公共指针指定的子程序。



(2) 当功能软元件 (FX、FY、FD) 被一个子程序使用时，用与功能软元件对应的 (S1) 到 (S5) 指定软元件。由 (S1) 到 (S5) 指定的软元件的内容简要说明如下。



(a) 在子程序执行之前，将位数据传送到 FX，将字数据传送到 FD。

(b) 在子程序执行之后，将 FY 和 FD 的内容传送到相应的软元件。

(c) 用于功能软元件的处理单元如下：

• FX、FY：位

• FD：4 字单元

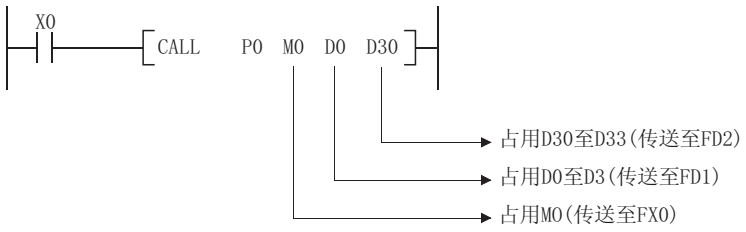
即将处理的数据大小依据变量中指定的软元件会有所不同。指定为功能软元件的软元件应该确保满足该数据量。

如果未能确保该数据量，将发生错误。

功能软元件	使用软元件	数据大小	备注
• FX • FY	位软元件	1 点	———
	当为字软元件进行位指定时	1 位	
• FD	当使用位软元件的位数指定时 *1	4 字	将要使用的指令有所不同。
	字软元件	4 字	

*1：当 (S1) 到 (S5) 指定的软元件号在位软元件的位数指定处不是 16 的倍数时，不会发生错误。

[主程序]



(3) (S1) 到 (S5) 可以和 CALL(P) 指令一起使用。

(4) 子程序将要使用的功能软元件数必须同 CALL(P) 指令中的变量数相同。而且，功能软元件和使用的 CALL(P) 变量类型应该完全一致。

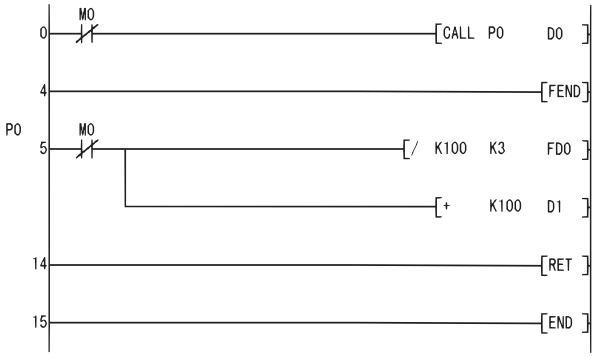
(5) 由 CALL(P) 指令指定的软元件号不应重叠。如果它们重叠，将无法获得准确的计算。

(6) CALL 指令变量中使用的软元件不应在子程序中使用。如果使用，将无法正常进行运算。
（参见以下程序示例。）

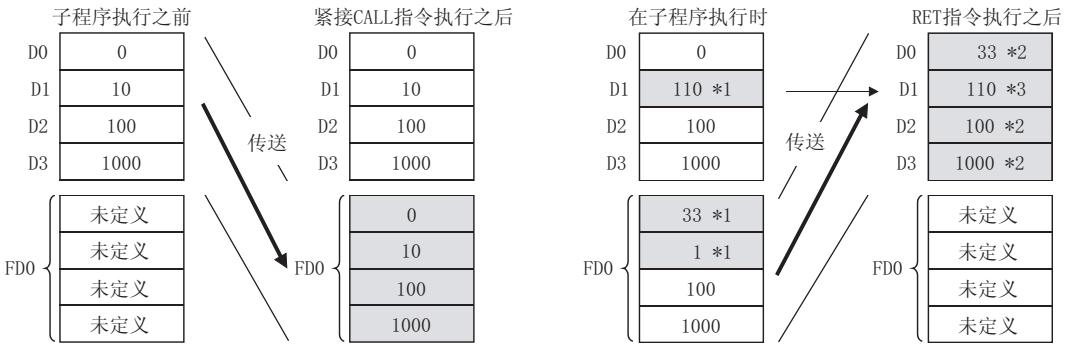
不正确的运算示例

以下程序示例为将 D0 指定到子程序的 FD0 中，且在子程序中使用 D1 时的动作。

［程序示例］



［子程序执行之后进行的运算］

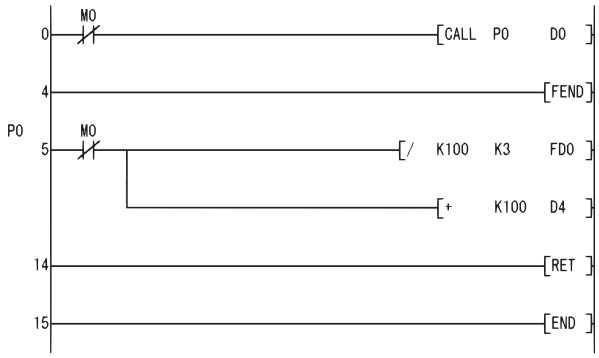


*1: 存储子程序的执行结果。
*2: 被功能软元件的值替换。
*3: D1 不反映功能软元件的值。

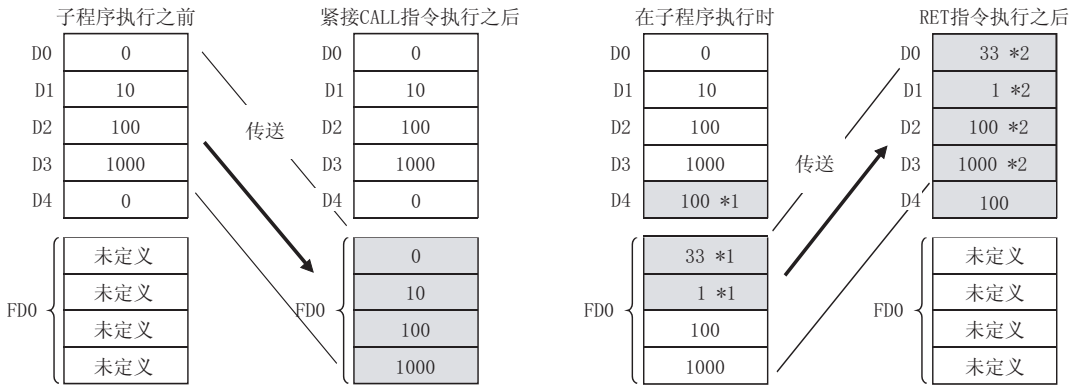
正确的运算示例

以下程序示例为将 D0 指定到子程序的 FD0 中，且在子程序中使用 D4 时的动作。

［程序示例］

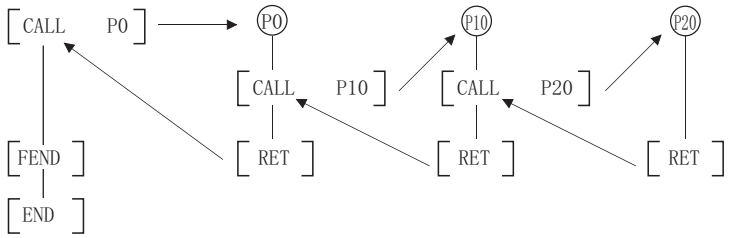


［子程序执行之后的动作］



*1: 存储子程序的执行结果。
*2: 被功能软元件的值替换。

(7) CALL(P) 指令最多可以有 16 级嵌套。
然而，该 16 级是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 和 XCALL 指令中嵌套的总级数。



(8) 在子程序内使其为 ON 的软元件将被保持，即使子程序变为未执行。
子程序执行期间变为 ON 的软元件可以通过 FCALL(P) 指令使其变为 OFF。

运算错误

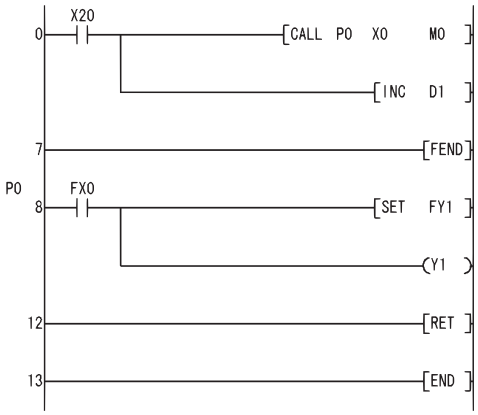
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 为变量指定的软元件无法确保数据大小。 (出错代码： 4101)
- 在 CALL(P) 指令执行之后，在执行 RET 指令之前执行了 END、FEND、GOEND 或者 STOP 指令。 (出错代码： 4211)
- 在执行 CALL(P) 指令之前执行 RET 指令。 (出错代码： 4212)
- 执行第 17 级嵌套。 (出错代码： 4213)
- CALL(P) 指令指定的指针没有子程序。 (出错代码： 4210)

程序示例

(1) 当 X20 变为 ON 时，以下程序执行一个带有变量的子程序。

[梯形图模式]



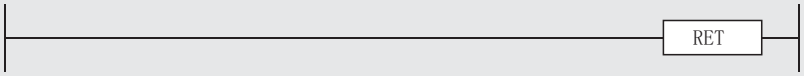
[列表模式]

步	指令	软元件
0	LD	X20
1	CALL	P0 X0 M0
5	INC	D1
7	FEND	
8	P0	
9	LD	FX0
10	SET	FY1
11	OUT	Y1
12	RET	
13	END	

7.6.4 从子程序返回 (RET)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

RET

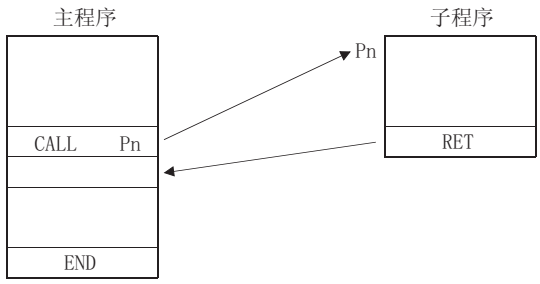


设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

(1) 表示子程序结束

(2) 当执行 RET 指令时，返回到调用子程序的 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 或者 XCALL 指令之后的步。



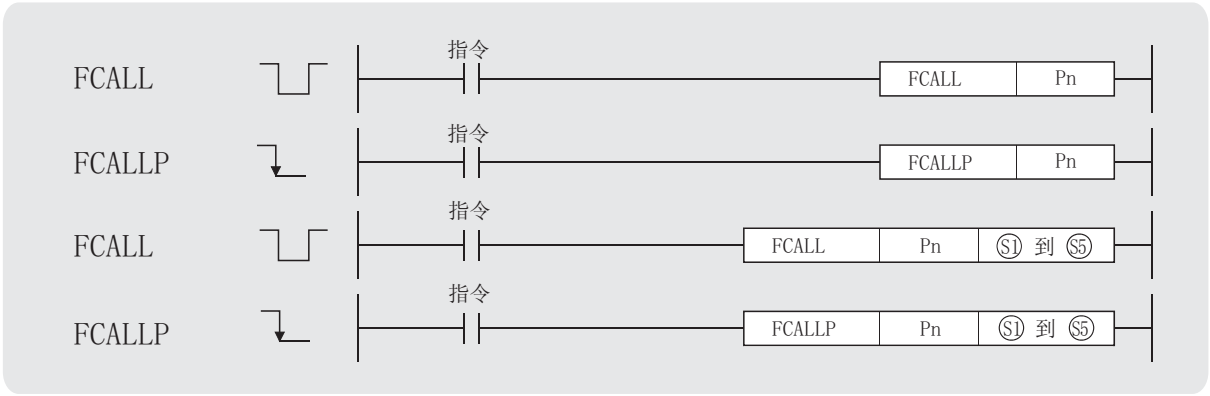
运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 在 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 或者 XCALL 指令执行之后，在执行 RET 指令之前执行 END、FEND、GOEND 或者 STOP 指令。 (出错代码： 4211)
- 在执行 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 或者 XCALL 指令之前执行 RET 指令。 (出错代码： 4212)

7.6.5 子程序输出 OFF 调用 (FCALL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Pn　　：子程序的起始指针号（软元件名称）

(S1) 到 (S5)　：作为变量传到子程序的软元件号（位、BIN 16 位、BIN 32 位）

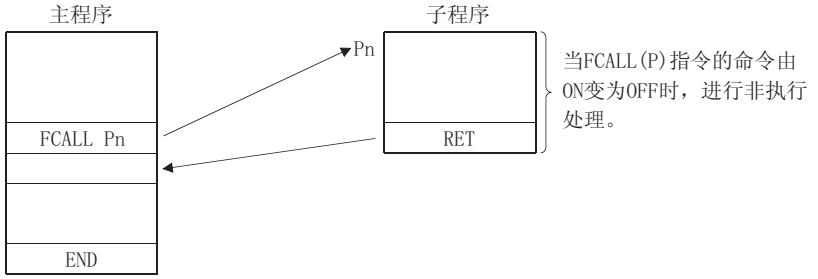
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P
Pn	---	---		---					○
(S1) 到 (S5)	○（除 F 之外）	○		○					---

功 能

(1) 当执行 FCALL(P) 时，执行由 Pn 指定的指针的子程序的非执行处理。

FCALL(P) 指令可以执行由相同程序文件中一个指针指定的子程序和由一个公共指针指定的子程序。

(a) 非执行处理同单线圈指令的条件触点为 OFF 时进行的处理完全相同。

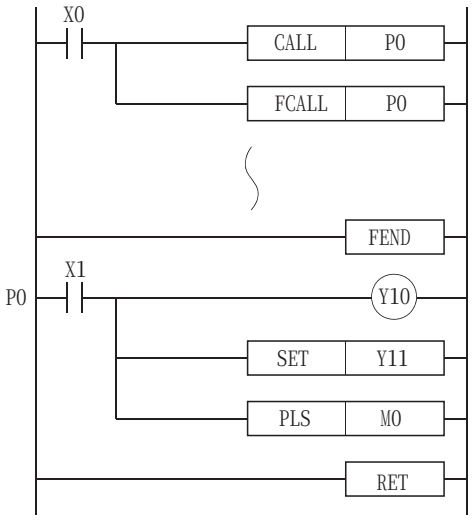


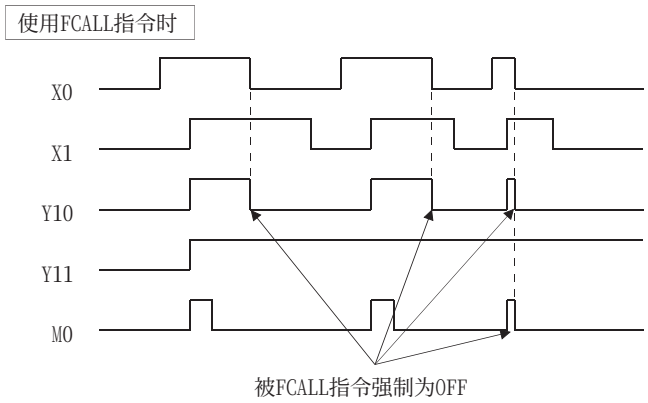
(b) 不管单个触点的状态是 ON 还是 OFF，非执行处理之后的单线圈指令的运算结果如下：

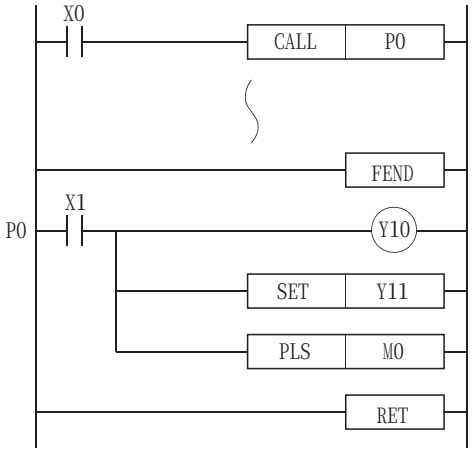
OUT 指令	.....	强制 OFF
SET 指令 RST 指令 SFT 指令 基本指令 应用指令	.....	保持状态
PLS 指令 脉冲生成指令（□ P）	.....	等同于条件触点为 OFF 时的处理
低速定时器和高速定时器的当前值	.....	0
累计定时器的当前值 计数器的当前值	.....	保持

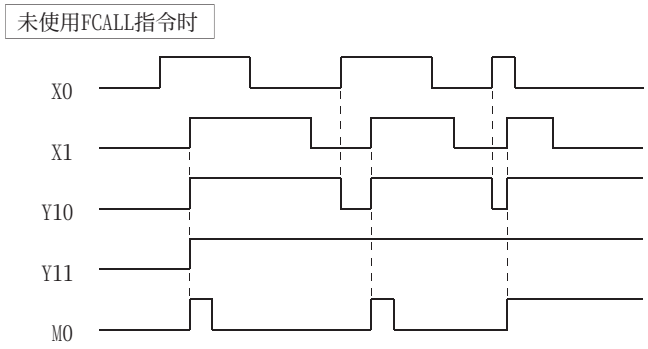
(2) FCALL(P) 指令和 CALL(P) 指令结合在一起使用。

(3) 如果未执行 FCALL(P) 指令（仅使用 CALL(P) 指令的情况），则当子程序指令变 OFF 时，由于还未执行子程序，所以将保存每个单线圈指令的输出状态。
当执行 FCALL(P) 指令时，OUT 指令和 PLS 指令（包插 □P 指令）可能被强制变为 OFF，以进行子程序非执行处理。



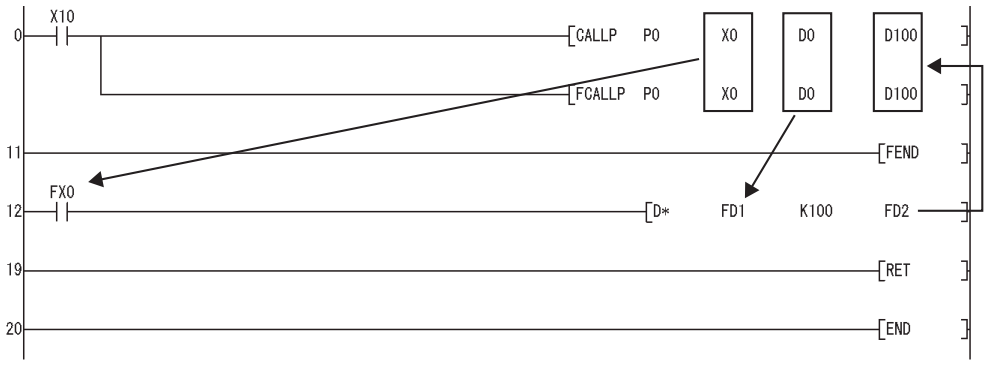






(4) 当功能软元件 (FX、FY、FD) 被一个子程序使用时，通过 ⓢ1 到 ⓢ5 指定与功能软元件对应的软元件。

由 ⓢ1 到 ⓢ5 指定的软元件的内容简要说明如下。



(a) 在子程序执行之前，将位数据传送到 FX，将字数据传送到 FD。

(b) 在子程序执行之后，将 FY 和 FD 的内容传送到相应的软元件。

(c) 功能软元件处理单元如下：

• FX、FY ： 位

• FD ： 4 字单元

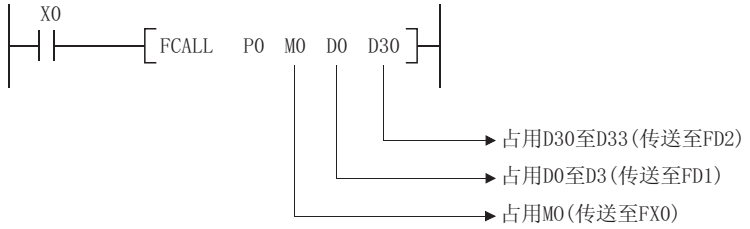
即将处理的数据大小依据变量中指定的软元件会有所不同。指定为功能软元件的软元件应该确保满足该数据量。

如果未能确保该数据量，将发生错误。

功能软元件	使用软元件	数据大小	备注
• FX • FY	位软元件	1 点	------
	当为字软元件进行位指定时	1 位	
• FD	当使用位软元件的位数指定时 *1	4 字	FD 的高 2 字变成 0
	字软元件	4 字	------

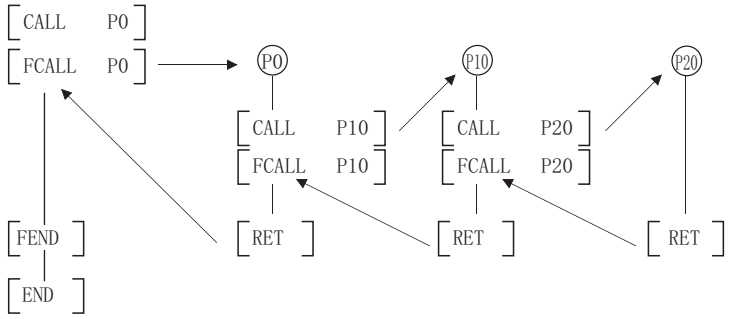
*1: 当 ⓢ1 到 ⓢ5 指定的软元件号在位软元件位数指定处不是 16 的倍数时，不会发生错误。

[主程序]



(5) FCALL(P) 指令可以使用 ⓢ1 到 ⓢ5 。

(6) FCALL(P) 指令最多可以有 16 级嵌套。
然而，该 16 级是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 和 XCALL 指令中嵌套级数的总和。



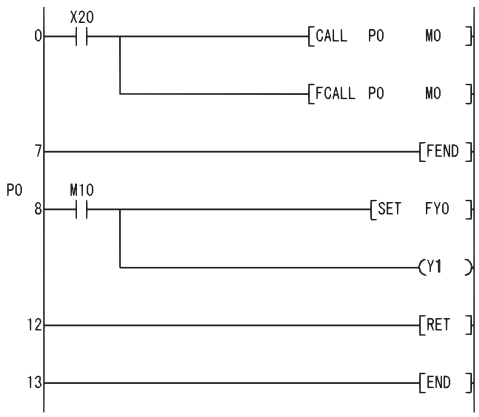
运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。
- 为变量指定的软元件无法确保数据大小。 (出错代码：4101)
- 在 FCALL(P) 指令执行之后，在执行 RET 指令之前执行 END、FEND、GOEND 或者 STOP 指令。 (出错代码：4211)
- 在执行 FCALL(P) 指令之前执行 RET 指令。 (出错代码：4212)
- 执行第 17 级嵌套。 (出错代码：4213)
- FCALL(P) 指令指定的指针的子程序不存在。 (出错代码：4210)

程序示例

(1) 当 X20 为 ON 时，以下程序执行一个带有变量的子程序，当 X20 从 ON 变到 OFF 时，强制进行非执行处理。

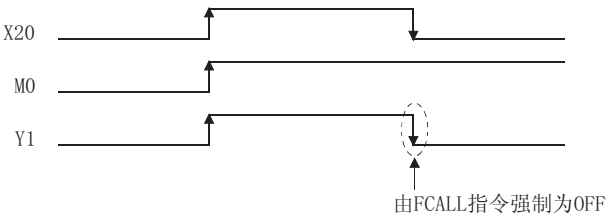
[梯形图模式]



[列表模式]

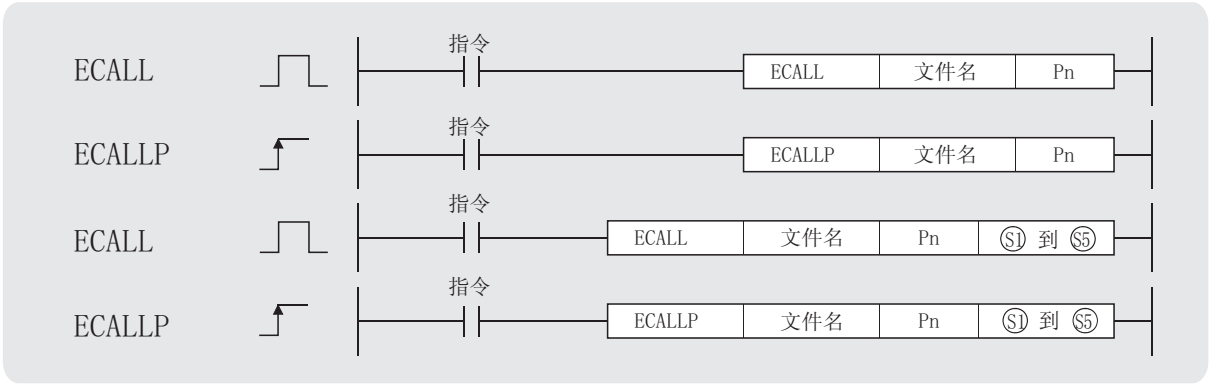
步	指令	软元件	
0	LD	X20	
1	CALL	P0	M0
4	FCALL	P0	M0
7	FEND		
8	P0		
9	LD	M10	
10	SET	FY0	
11	OUT	Y1	
12	RET		
13	END		

[动作]



7.6.6 程序文件之间的子程序调用 (ECALL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

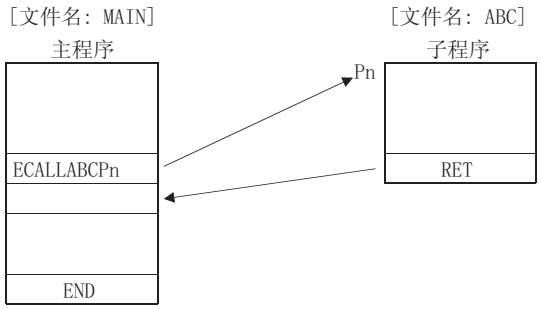


文件名　：文件名即将调用的程序文件的名称（字符串）
Pn　：子程序的起始指针号（软元件名称）
(S1)到(S5)　：作为变量传到子程序的软元件号（位、BIN 16 位、BIN 32 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	P
文件名	---	○		---					○	---
Pn	---	---		---					---	○
(S1) 到 (S5)	○（除 F 之外）	○		○					---	---

功 能

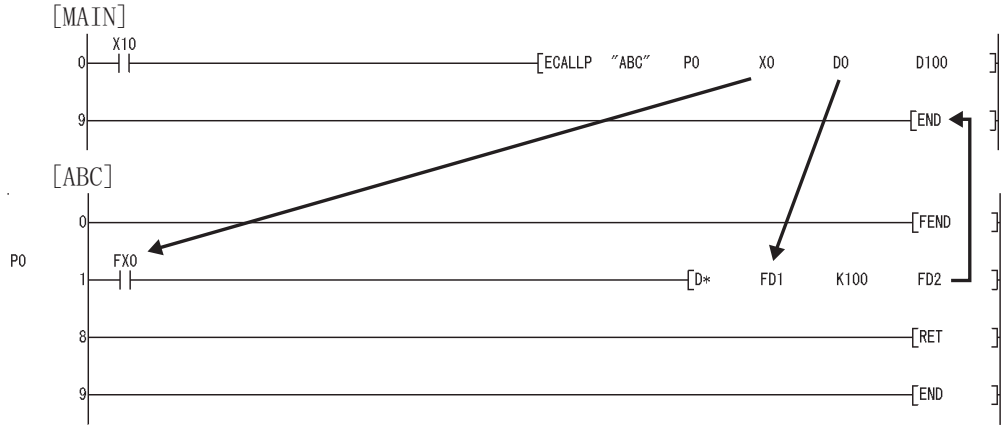
(1) 当执行 ECALL(P) 指令时，以指定的程序文件名执行 Pn 指定的指针的子程序。可使用 ECALL(P) 指令从不同的程序文件中调用使用局部指针的子程序。



(2) 文件名仅可以指定存储在驱动器 0(程序内存 / 内置 RAM) 中的程序文件的名称。

(3) 指定文件名时，不需要指定扩展名 (.QPG)。
(只对于 .QPG 文件)

(4) 当功能软元件 (FX、FY、FD) 被一个子程序使用时，通过 ⓢ1 到 ⓢ5 指定与功能软元件对应的软元件。
由 ⓢ1 到 ⓢ5 指定的软元件的内容简要说明如下。



(a) 在子程序执行之前，将位数据传送到 FX，将字数据传送到 FD。

(b) 在子程序执行之后，将 FY 和 FD 的内容传送到相应的软元件。

(c) 功能软元件的处理单元如下：

• FX、FY ： 位

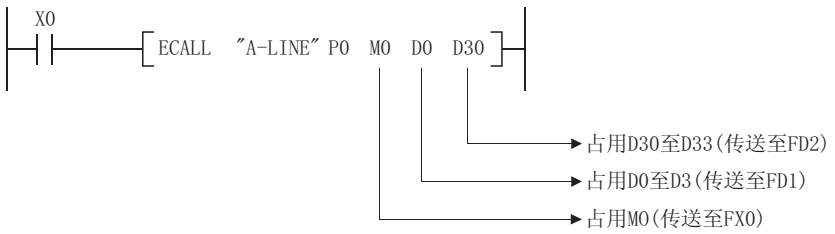
• FD ： 4 字单元

即将处理的数据大小依据变量中指定的软元件会有所不同。指定为功能软元件的软元件应该确保该数据量。
如果未能确保该数据量，将发生错误。

功能软元件	使用软元件	数据大小	备注
• FX • FY	位软元件	1 点	------
	当为字软元件进行位指定时	1 位	
• FD	当使用位软元件的位数指定时 *1	4 字	数据大小根据将要使用的指令有所不同。
	字软元件	4 字	

*1: 当 ⓢ1 到 ⓢ5 指定的软元件号在位软元件位数指定不是 16 的倍数时，不会发生错误。

[主程序]



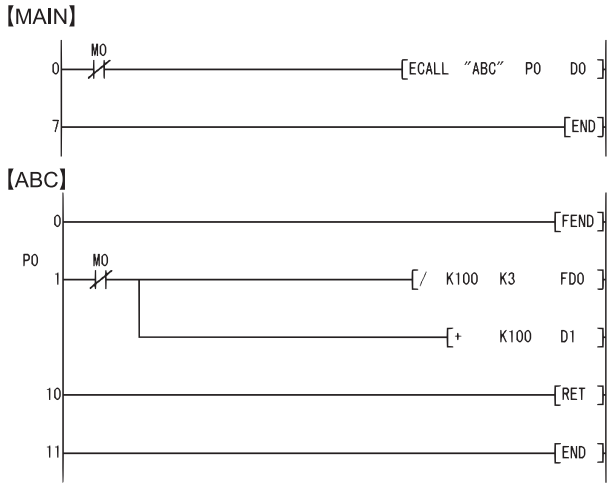
(5) ⓢ1 到 ⓢ5 可以通过 ECALL 指令使用。

(6) ECALL 指令变量中使用的软元件不能应用于子程序。
如果使用，则无法获得准确的计算。（参见以下程序示例。）

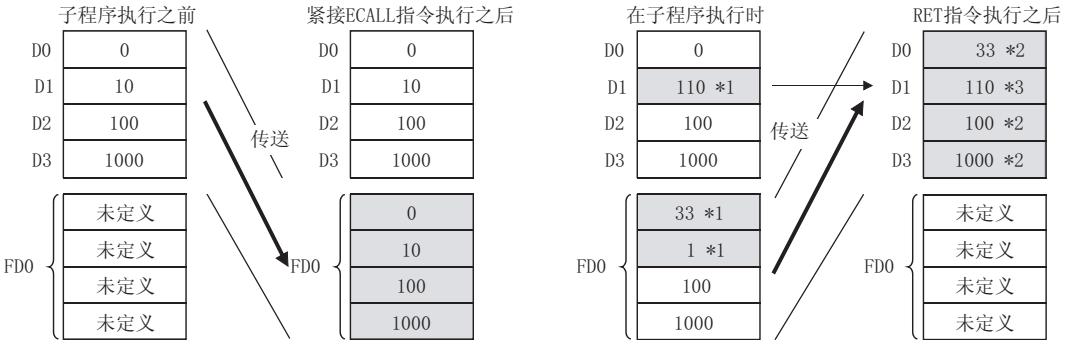
不正确的运算示例

以下程序示例显示当在子程序中将 D0 指定为 FD0 且在子程序中使用 D1 时进行的运算。

[程序示例]



[子程序执行之后进行的运算]

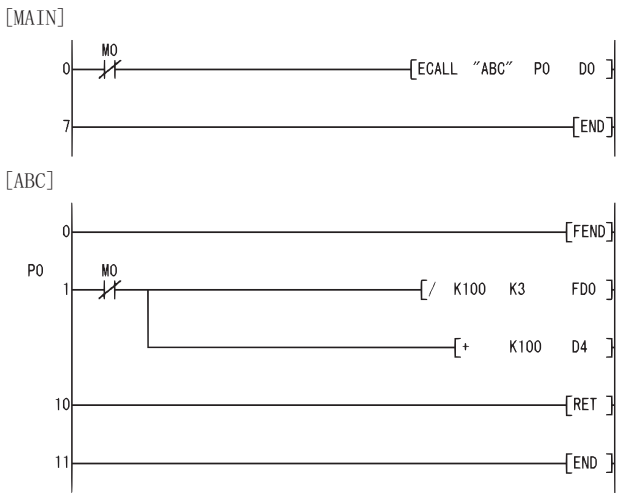


*1: 存储子程序的执行结果。
*2: 被功能软元件的值替换。
*3: D1 不能反映功能软元件的值。

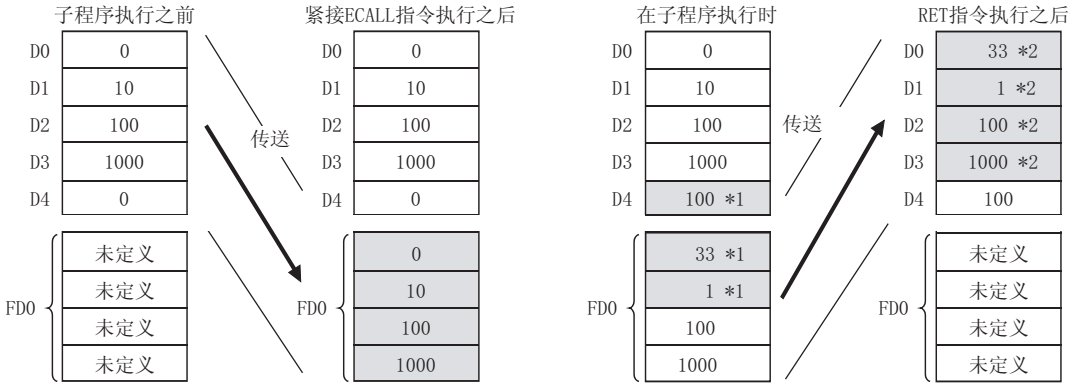
正确的运算示例

将 D0 指定到子程序的 FD0 中，在子程序中使用 D4 时的动作如下所示。

[程序示例]



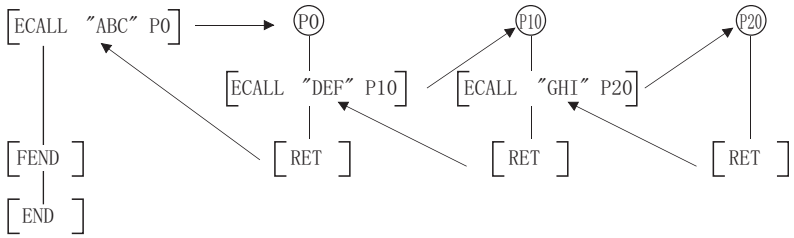
[正确的运算示例]



*1: 存储子程序的执行结果。
*2: 被功能软元件的值替换。

(7) 在 ECALL(P) 指令中由变量指定的软元件号不应重叠。
如果重叠，则无法获得准确的计算。

(8) ECALL(P) 指令最多可以有 16 级嵌套。
然而，该 16 级是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 和 XCALL 指令中嵌套的总级数。



(9) 子程序内使其为 ON 的软元件将被保持，即使子程序变为未执行。
子程序执行期间使其为 ON 的软元件可以通过 EFCALL(P) 指令使其变为 OFF。

运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

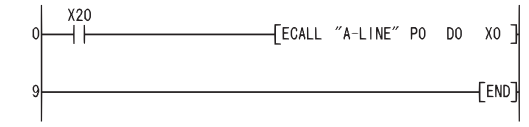
- 为变量指定的软元件无法预留出能容纳数据容量的空间。 (出错代码：4101)
- 在 ECALL(P) 指令执行之后，在执行 RET 指令之前执行了 END、FEND、GOEND 或者 STOP 指令。 (出错代码：4211)
- 在执行 ECALL(P) 指令之前执行了 RET 指令。 (出错代码：4212)
- 执行了 17 级嵌套。 (出错代码：4213)
- ECALL(P) 指令指定的指针的子程序不存在。 (出错代码：4210)
- 指定的文件不存在。 (出错代码：2410)
- 指定的文件无法执行。 (出错代码：2411)

程序示例

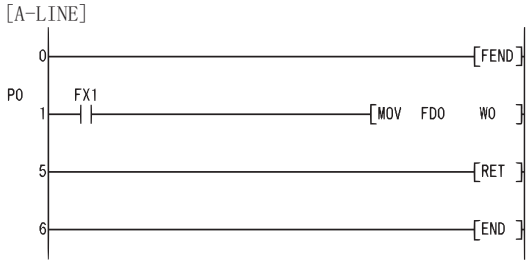
(1) 当 X20 为 ON 时，以下程序执行 P0 A-LINE.QPG。

[梯形图模式]

[MAIN]



[A-LINE]



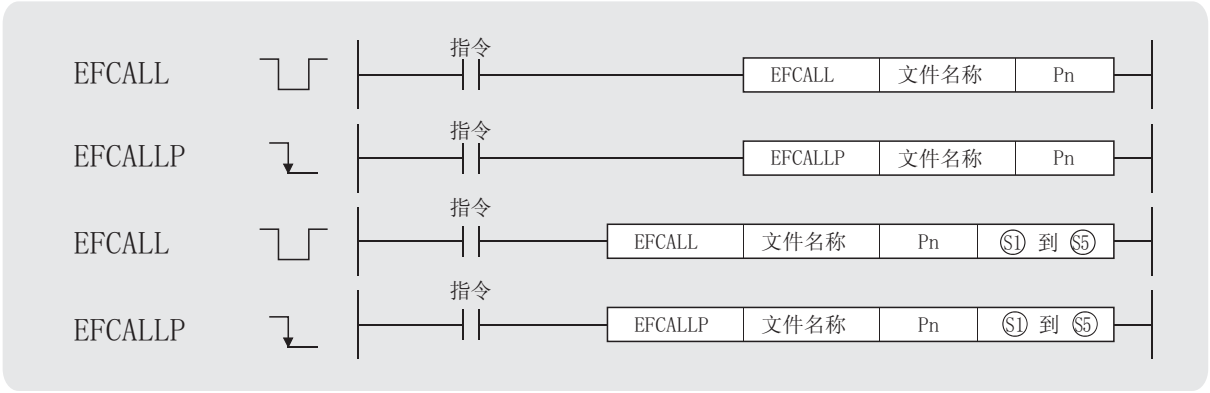
[列表模式]

步	指令	软元件		
0	LD	X20		
1	ECALL	"A-LINE" P0	D0	X0
9	END			

步	指令	软元件	
0	FEND		
1	P0		
2	LD	FX1	
3	MOV	FD0	W0
5	RET		
6	END		

7.6.7 程序文件之间的子程序输出 OFF 调用 (EFCALL(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



文件名　：即将调用的程序文件的名称（字符串）
Pn　：子程序的起始指针号（软元件名称）
(S1) 到 (S5)　：作为变量传到子程序的软元件号（位、BIN 16 位、BIN 32 位）

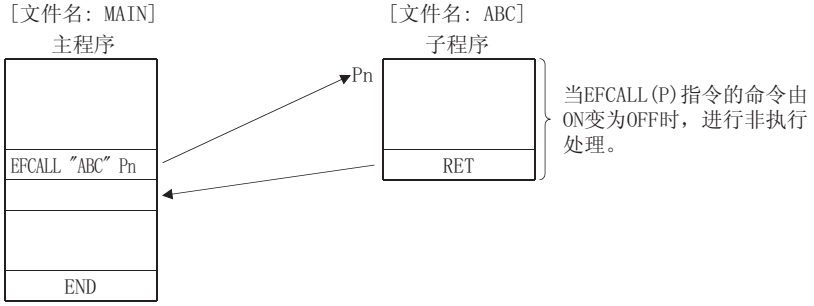
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	P
文件名称	---	○		---					○	---
Pn	---	---		---					---	○
(S1) 到 (S5)	○（除 F 之外）	○		○					---	---

功 能

(1) 当执行 EFCALL(P) 指令时，执行 Pn 指定的指针的子程序的非执行处理。

[也可使用 EFCALL(P) 指令从不同程序文件中调用使用局部指针的子程序。]

(a) 非执行处理同单线圈指令的状态触点为 OFF 时进行的处理完全相同。

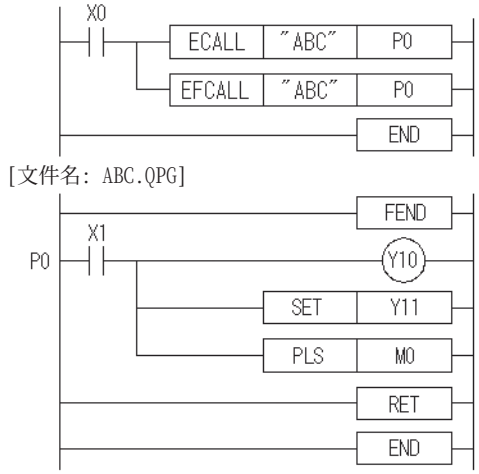


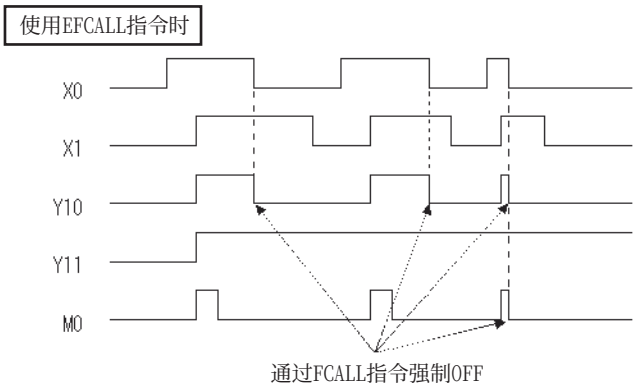
(b) 不管状态触点的 ON/OFF 状态如何，非执行处理之后的单线圈指令的运算结果如下：

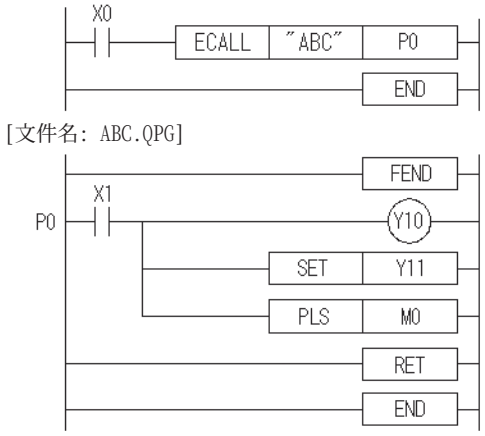
OUT 指令	.....	强制为 OFF
SET 指令 RST 指令 SFT 指令 基本指令 应用指令	.....	保持状态
PLS 指令 脉冲生成指令（□ P）	.....	等同于条件触点为 OFF 时的处理
低速定时器和高速定时器的当前值	.....	0
累计定时器当前值 计数器当前值	.....	保持

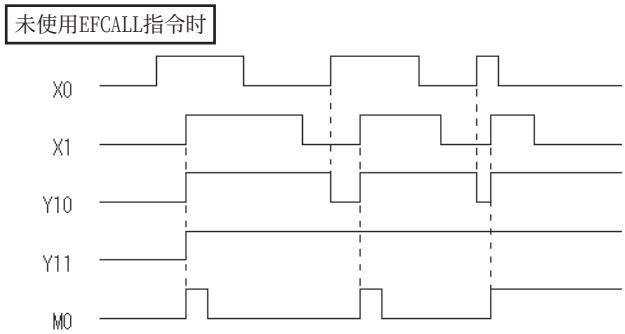
(2) EFCALL(P) 指令与 ECALL(P) 指令结合在一起使用。

(3) 如果 EFCALL(P) 指令与 ECALL(P) 指令一起使用，当执行指令为 OFF 时，执行子程序的非执行处理，因此可以将 OUT 指令和 PLS 指令（包括 □P 指令）强制变为 OFF。
如果 EFCALL(P) 指令没有与 ECALL(P) 指令一起使用，即使执行指令为 OFF，也不执行子程序的非执行处理。因此，各线圈指令的输出状态保持不变。





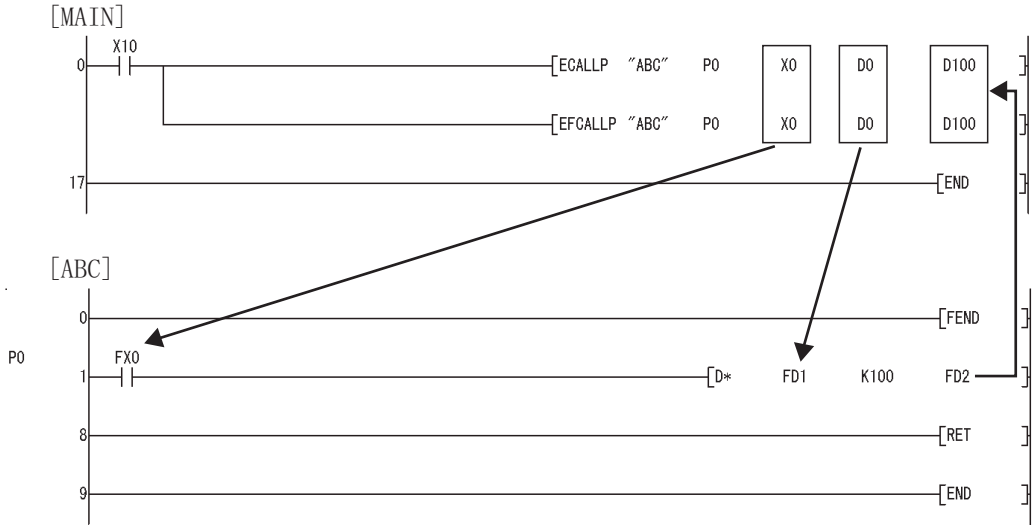




(4) 仅可以指定存储在驱动器 0(程序内存 / 内置 RAM) 中的程序文件作为文件名。

(5) 指定文件名时，不需要指定扩展名 (.QPG)。
(只对于 .QPG 文件)

(6) 当功能软元件 (FX、FY、FD) 被一个子程序使用时，通过 ⓢ1 到 ⓢ5 指定与功能软元件对应的软元件。



(a) 在子程序执行之前，将位数据传送至 FX，将字数据传送至 FD。

(b) 在子程序执行之后，将 FY 和 FD 的内容传送至相应的软元件。

(c) 功能软元件的处理单元如下：

• FX、FY ： 位

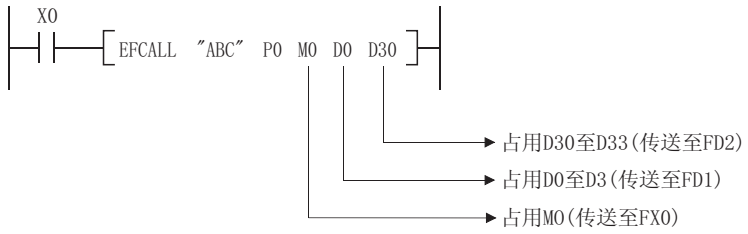
• FD ： 4 字单元

即将处理的数据大小依据变量中指定的软元件会有所不同。指定为功能软元件的软元件应该确保该数据量。
如果未能确保该数据量，将发生错误。

功能软元件	使用软元件	数据大小	备注
t FX	位软元件	1 点	---
t FY	当为字软元件进行了位指定时	1 位	
t FD	当使用位软元件的位数指定时 *1	4 字	FD 的高 2 字变为 0
	字软元件	4 字	---

*1: 当 ⓢ1 到 ⓢ5 指定的软元件号在位软元件的位数指定处不是 16 的倍数时，不会发生错误。

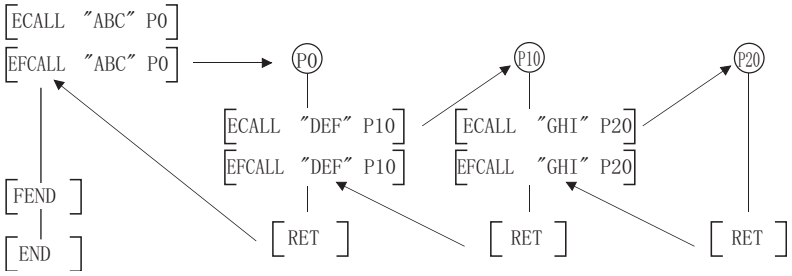
[主程序]



(7) 可通过 EFCALL(P) 指令使用 ⓢ1 到 ⓢ5 。

(8) 由子程序使用的功能软元件数必须同 EFCALL(P) 指令使用的变量数完全相同。
此外，功能软元件应该同 EFCALL(P) 指令使用的变量类型相一致。

(9) EFCALL(P) 指令最多可以用 16 级嵌套。
然而，该 16 级是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 和 XCALL 指令中嵌套的总级数。



运算错误

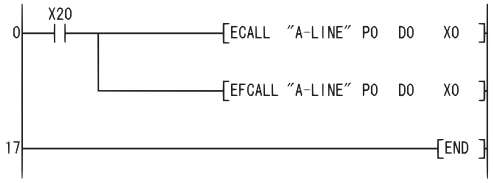
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，如果还未进行变址修改，则将并且出错代码存储在 SD0 中。

- 为变量指定的软元件无法确保数据大小。 (出错代码： 4101)
- 在 EFCALL(P) 指令执行之后，在执行 RET 指令之前执行 END、FEND、GOEND 或者 STOP 指令。 (出错代码： 4211)
- 在执行 EFCALL(P) 指令之前执行 RET 指令。 (出错代码： 4212)
- 执行第 17 级嵌套。 (出错代码： 4213)
- EFCALL(P) 指令指定指针的子程序不存在。 (出错代码： 4210)
- 指定文件不存在。 (出错代码： 4210)
- 指定文件无法执行。 (出错代码： 2411)

程序示例

(1) 当 X20 为 ON 时，以下程序执行带有变量的子程序，当 X20 由 ON 变为 OFF 时，强制进行非执行处理。

[梯形图模式]



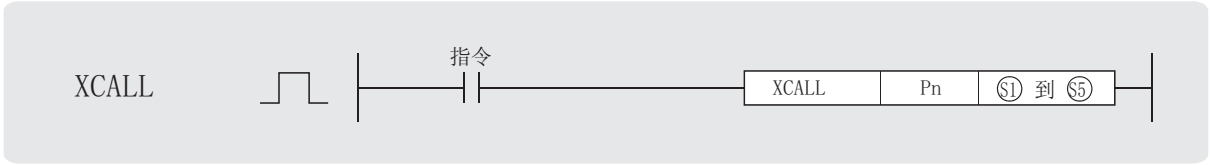
[列表模式]

步	指令	软元件
0	LD	X20
1	ECALL	"A-LINE" P0 D0 X0
9	EFCALL	"A-LINE" P0 D0 X0
17	END	

7.6.8 子程序调用 (XCALL)

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1: 序列号前五位数字是 04122 或以后。



Pn　　: 子程序的起始指针号 (软元件名称)

(S1) 到 (S5): 作为变量传到子程序的软元件号 (位、BIN 16 位、BIN 32 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它 P
	位	字		位	字				
P	---	---		---					○
(S1) 到 (S5)	○ (除 F 之外)	○		○					---

功 能

(1) XCALL 指令是进行子程序的执行和非执行处理的指令。

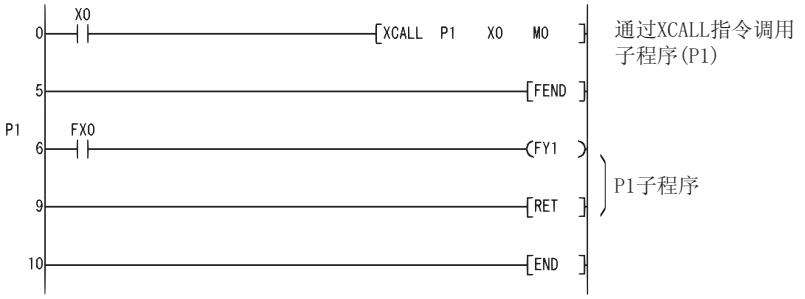
(a) 子程序的执行。
根据条件触点的 ON/OFF 状态执行各线圈指令。

(b) 当 X0 由 ON 变为 OFF 时，进行子程序 “P1” 的非执行处理。
对各线圈指令执行与条件触点为 OFF 状态时相同的处理。
与条件触点的 ON/OFF 状态无关，非执行处理之后的各线圈指令的运算结果如下所示。

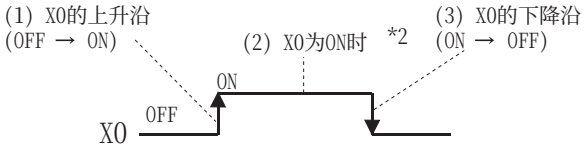
OUT 指令	..... 强制为 OFF
SET 指令 RST 指令 SFT 指令 基本指令 应用指令	..... 保持状态
PLS 指令 脉冲生成指令 (□ P)	..... 等同于条件触点为 OFF 时的处理
低速定时器和高速定时器的当前值	 0
累计定时器当前值 计数器当前值	..... 保持

(2) 根据 CPU 模块的型号，XCALL 指令的运行各有不同。以下程序示例显示各 CPU 模块的 XCALL 指令的动作。

[程序示例]



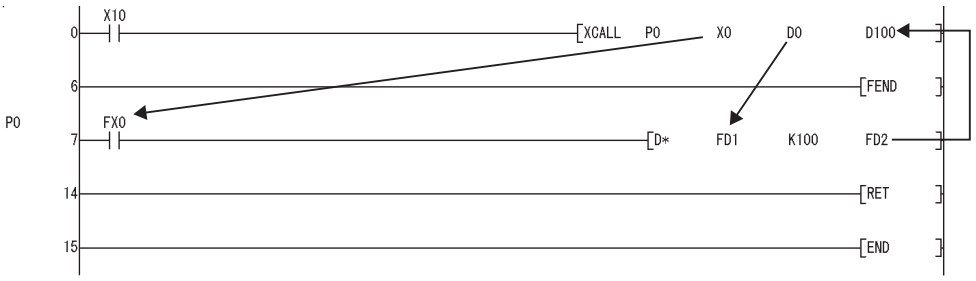
[X0 的 ON/OFF 时序]



*2: X0 为 ON 时 (2)) 的时间不包括 X0 的上升沿 (1)) 的时间。

CPU 类型	XCALL 指令的动作
• QnACPU • Q2ASCPU • Q4ARCPU • 过程控制 CPU （序列号的前五位数字 :07031 或以前） • 高性能型 QCPU （序列号的前五位数字 :06081 或以后）	1) X0 的上升沿时：无处理（不执行 “P1” 的子程序。） 2) 当 X0 为 ON 时：执行 “P1” 的子程序。 3) X0 的下降沿时：执行 “P1” 子程序的 “非执行处理”。
• 高性能型 QCPU （序列号的前五位数字 :06082 或以后。） • 过程控制 CPU （序列号的前五位数字 :07032 或以后。）	1) 通过 SM734(XCALL 指令执行条件指定）选择 X0 的上升沿时的动作。 • 当 SM734 为 OFF 时：无处理（不执行 “P1” 的子程序。） • 当 SM734 为 ON 时：执行 “P1” 的子程序。 2) 当 X0 为 ON 时：执行 “P1” 的子程序。 3) X0 的下降沿时：执行 “P1” 子程序的 “非执行处理”。
• 冗余 CPU • 基本型 QCPU • 通用型 QCPU	1) X0 的上升沿时：执行 “P1” 的子程序。 2) 当 X0 为 ON 时：执行 “P1” 的子程序。 3) X0 的下降沿时：执行 “P1” 子程序的 “非执行处理”

(3) 在子程序中使用功能软元件 (FX、FY、FD) 时，指定与 (S1) 到 (S5) 中功能软元件对应的软元件。



(a) 在子程序执行之前，将位数据传送到 FX，将字数据传送到 FD。

(b) 在子程序执行之后，将 FY 和 FD 的内容传送到相应的软元件。

(c) 功能软元件的处理单元如下：

• FX、FY ：位

• FD ：4 字单元

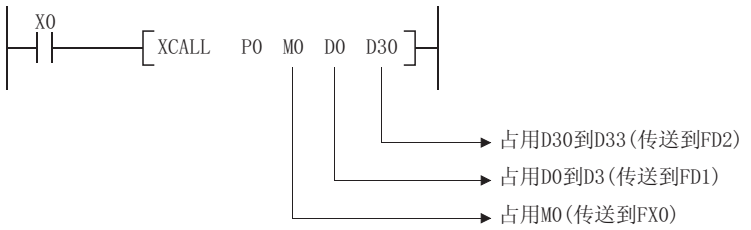
即将处理的数据大小依据变量中指定的软元件会有所不同。指定为功能软元件的软元件应该确保该数据量。

如果未能确保该数据量，将发生错误。

功能软元件	使用软元件	数据大小	备注
• FX • FY	位软元件	1 点	------
	当为字软元件进行位指定时	位	
• FD	当使用位软元件的位数指定时 *3	4 字	数据大小根据将要使用的指令有所不同。
	字软元件	4 字	

*3: 当 (S1) 到 (S5) 指定的软元件号在位软元件的位数指定处不是 16 的倍数时，不会发生错误。

[主程序]

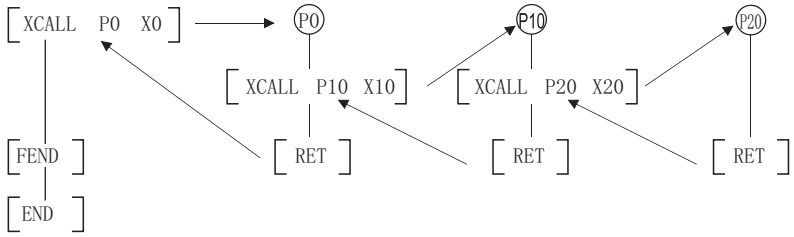


(4) (S1) 到 (S5) 可由 XCALL 指令使用。

(5) 由子程序使用的功能软元件数必须同 XCALL 指令使用的变量数完全相同。
而且，功能软元件和 XCALL 变量类型应该完全一致。

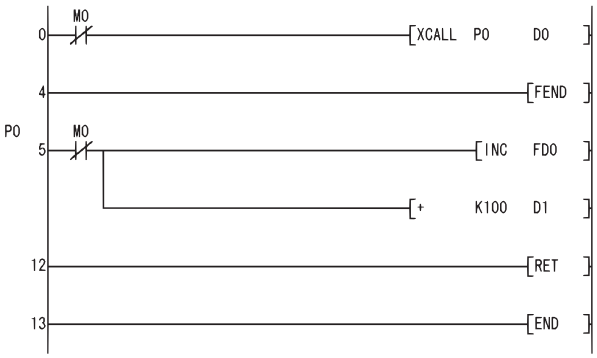
(6) XCALL 指令变量所指定的软元件号不应重叠。
如果重叠，则无法获得准确计算。

(7) XCALL 指令最多可以有 16 级嵌套。然而，该 16 级是 CALL(P)、FCALL(P)、ECALL(P)、EFCALL(P) 和 XCALL 指令中嵌套的总级数。

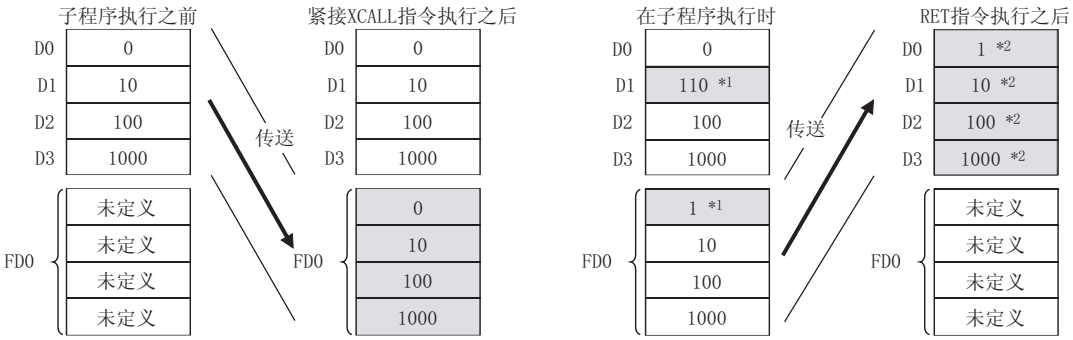


(8) XCALL 指令变量所使用的软元件不能应用于子程序。
如果使用，则无法获得准确计算。(请参见以下程序示例。)
以下程序示例显示当在子程序中将 D0 指定为 FD0 且在子程序中使用 D1 时进行的运算。

[程序示例]



[子程序执行之后进行的运算]



*1: 存储子程序的执行结果。
*2: 被功能软元件的值替换。D1 不反映子程序中的运算结果。

运算错误

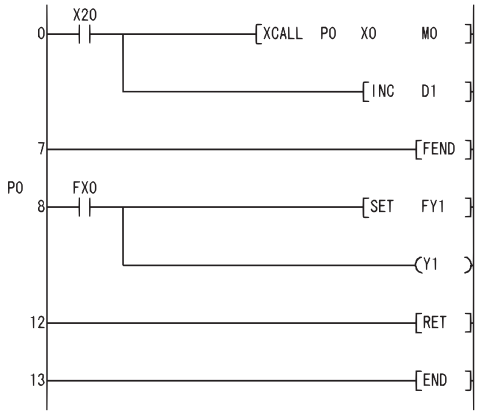
(1) 在以下情形下，发生运行错误，错误标志 (SM0) 变为 ON，出错代码存储在 SD0 中。

- 为变量指定的软元件无法确保数据大小。 (出错代码： 4101)
- 在 XCALL(P) 指令执行之后，在执行 RET 指令之前执行 END、FEND、GOEND 或者 STOP 指令。 (出错代码： 4211)
- 在执行 XCALL(P) 指令之前执行 RET 指令。 (出错代码： 4212)
- 执行第 17 级嵌套。 (出错代码： 4213)
- XCALL(P) 指令指定的指针没有子程序。 (出错代码： 4210)

程序示例

(1) 当 X20 变为 ON 时，以下程序执行带有变量的子程序。

[梯形图模式]



[列表模式]

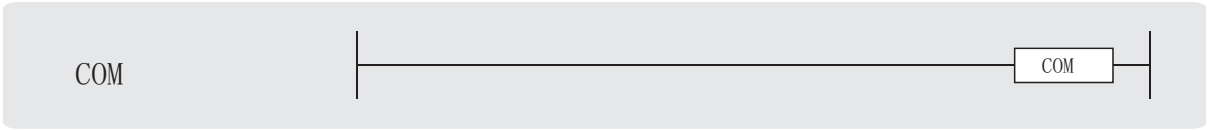
步	指令	软元件
0	LD	X20
1	XCALL	P0 X0 M0
5	INC	D1
7	FEND	
8	P0	
9	LD	FX0
10	SET	FY1
11	OUT	Y1
12	RET	
13	END	

7.6.9 刷新指令 (COM)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

以下 CPU 型号 COM 指令，请参见 9.11 节。

- 序列号为 04122 或之后的基本型 QCPU。
- 序列号为 04012 或之后的高性能型 QCPU。
- 序列号为 07032 或之后的过程 CPU。
- 冗余 CPU。
- 通用型 QCPU。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---			---					

功 能

(1) 在以下情况下，使用 COM 指令：

(a) 要提高同远程 I/O 站的传送 / 接收处理速度时。

(b) 要在执行数据链接期间，确保同使用不同扫描周期的其它站之间进行可靠的数据传送 / 接收。

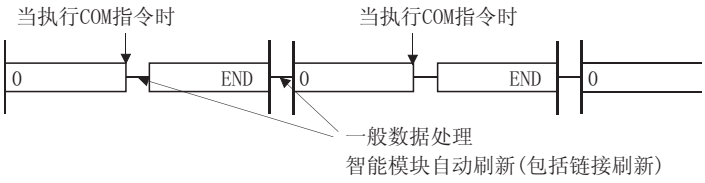
(2) 依据特殊继电器 SM775 是 ON 还是 OFF 状态，COM 指令的处理会有变化。

- SM775 为 OFF 时 ：进行自动刷新及与外围设备的通信。 *1 *2
- SM775 为 ON 时 ：仅进行与外围设备的通信。 *1

*1: 在与外围设备的通信中也进行以下处理。
- 其它站监视处理。
- 在串行通信中，对其它的智能功能模块的缓冲内存进行读取处理。

*2: 在自动刷新处理中进行以下处理。
- MELSECNET/10、MELSECNET/H 的刷新
- CC-Link 的刷新
- 智能功能模块的自动刷新 (QnACPU、Q4ARCPU 中不进行此处理。)

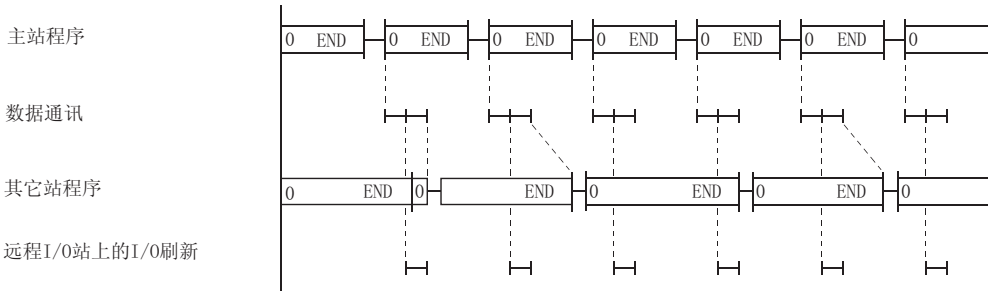
(3) 在即将执行 COM 指令的时候，CPU 模块将暂停顺控程序的处理，而进行与 END 处理期间智能功能模块的一般数据处理和自动刷新（包括链接刷新）相同的操作。
但是，不进行低速的 MELSECNET/10 或 MELSECNET/H 循环刷新。



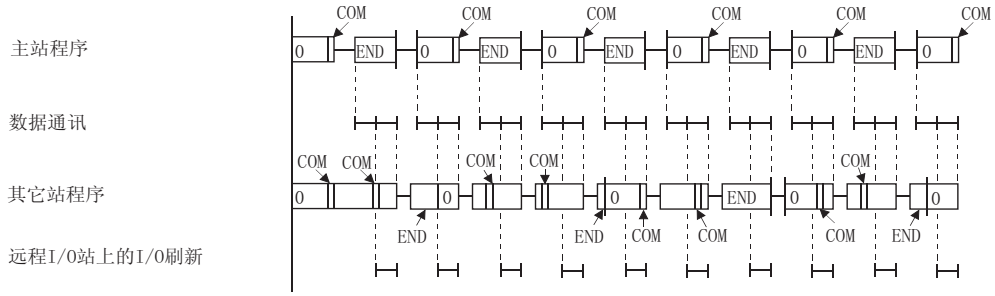
(4) COM 指令可在顺控程序中使用任意次。
但是，请注意，顺控程序的扫描周期将会因智能功能模块的一般数据处理和自动刷新（包括链接刷新）所花费的时间而增长。

(5) 使用 COM 指令的数据通讯

(a) 未使用 COM 指令时的数据通讯示例



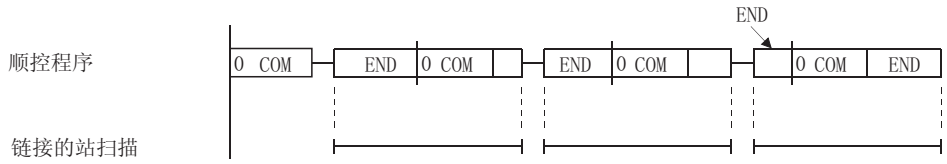
(b) 使用 COM 指令时的数据通讯示例



1) 当在主站使用 COM 指令时，可以无条件提高同远程 I/O 站的数据通讯重复次数，如上面示例 (b) 所示，从而加快数据通讯的速度。
2) 如果发生远程站顺控程序扫描周期比主站扫描周期长的情况，在主站使用 COM 指令将可以避免定时困难，在发生定时困难时数据将无法输入的，如上面示例 (a) 所示。
3) 当 COM 指令已经在其它站使用时，每次该站从主站接收到一个命令，就进行一次链接刷新。

· 第0步 ～COM指令
· COM指令 ～COM指令
· COM指令 ～END指令
} 每次在这些事件之间，链接刷新可进行一次

(6) 如果来自链接站的扫描周期比主站的顺控程序扫描周期长，在主站指定 COM 指令将不会提高数据通讯速度。



要 点

COM 指令中不能使用的程序如下所示：

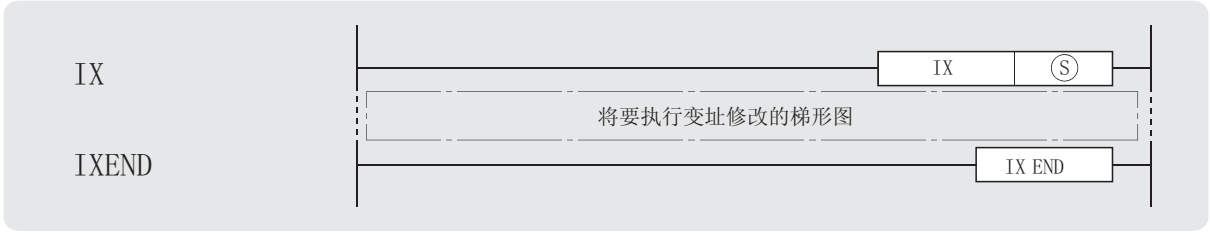
- 低速执行类型程序
- 中断程序
- 恒定周期执行类型程序

运算错误

(1) 没有与 COM 指令相关的运算错误。

7.6.10 整个梯形图的变址修饰(IX、IXEND)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ: 存储变址修改数据的软元件的起始软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---					

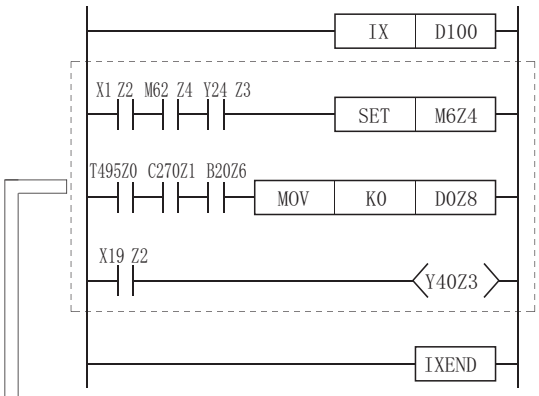
功 能

(1) 使用在变址修饰表中设定的变址修饰值，对梯形图内从 IX 指令起到 IXEND 指令为止的所有软元件进行软元件号的变址修饰。
关于变址修饰表的创建方法，请参见 7.6.11 项。
变址修改表的构成和相应的变址寄存器号如下所示：

	软元件名称	变址寄存器号		软元件名称	变址寄存器号	
Ⓢ	定时器(T)的修饰值	Z0	Ⓢ + 8	数据寄存器(D)的修饰值	Z8	
Ⓢ + 1	计数器(C)的修饰值	Z1	Ⓢ + 9	链接寄存器(W)的修饰值	Z9	
Ⓢ + 2	输入(X)的修饰值	Z2	Ⓢ + 10	文件寄存器(R)的修饰值	Z10	
Ⓢ + 3	输出(Y)的修饰值	Z3	Ⓢ + 11	缓冲寄存器I/O号(U)的修饰值	Z11	
Ⓢ + 4	内部继电器(M)的修饰值	Z4	Ⓢ + 12	缓冲寄存器(G)的修饰值	Z12	
Ⓢ + 5	锁存继电器(L)的修正值	Z5	Ⓢ + 13	链接直接软元件网络号(J)的修正值	Z13	*1
Ⓢ + 6	链接继电器(B)的修饰值	Z6	Ⓢ + 14	文件寄存器(ZR)的修正值	Z14	
Ⓢ + 7	变址继电器(V)的修饰值	Z7	Ⓢ + 15	指针(P)的修饰值	Z15	

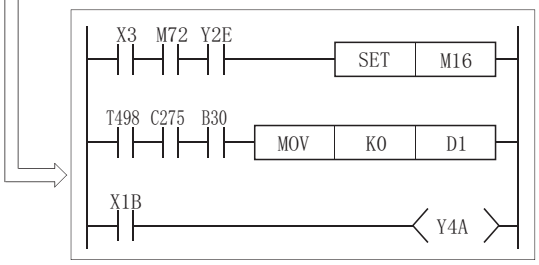
*1: 使用基本型 QCPU 时，不能使用编号为 Z10 以后的变址寄存器。

(2) 软元件号变址修改是通过为每个软元件预设修正值来实现的。将这些值加到梯形图内 IX 和 IXEND 指令之间所有使用的软元件上，并根据增加后的软元件号进行程序处理。



修饰值

D100	8	T	(Z0)
D101	5	C	(Z1)
D102	2	X	(Z2)
D103	10	Y	(Z3)
D104	10	M	(Z4)
D105	20	L	(Z5)
D106	16	B	(Z6)
D107	20	V	(Z7)
D108	1	D	(Z8)



· 将值“2”加到X1和X9 → 当作X3和X1B处理。
· 将值“10”(Ah)加到X24和Y40 → 当作Y2E和Y4A处理。
· 将值“10”加到M6和M62 → 当作M16和M72处理。
· 将值“16”(10h)加到B20 → 当作B30处理。
· 将值“8”加到T495 → 当作T498处理。
· 将值“5”加到C270 → 当作C275处理。
· 将值“1”加到D0 → 当作D1处理。

(3) 那些在输入条件建立时仅执行一次的指令，诸如 PLS、PLF 和□□□P 指令，无法通过使用 IX 到 IXEND 指令环进行变址修改。

(4) 在增加修正值使得软元件号超出了软元件范围的情形下，无法进行准确处理。

(5) 在顺控程序的在线程序改变期间，不要执行 IX 或 IXEND 指令 (RUN 期间写入)。否则，无法进行准确处理。

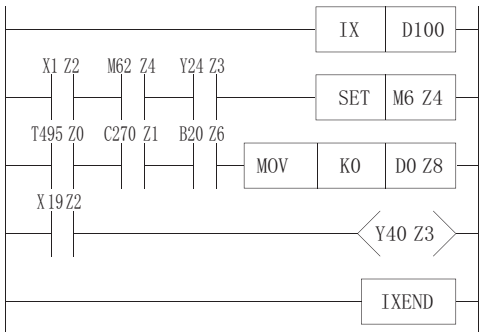
(6) 以 BIN 值为随机字软元件预设修正值，已经设定修正值的软元件的起始软元件号由 ⓢ 指定。

(7) 不要在 IX 和 IXEND 指令之间同时执行扫描执行类型程序和中断程序。

(8) 根据所使用的 GPP 功能软件包，有时需要进行程序展开或由用户创建程序。

(a) 需要由用户创建程序时（使用 GX Developer 创建时）

需要由用户将变址寄存器附加到通过 IX 和 IXEND 指令进行变址修饰的梯形图中。 *2

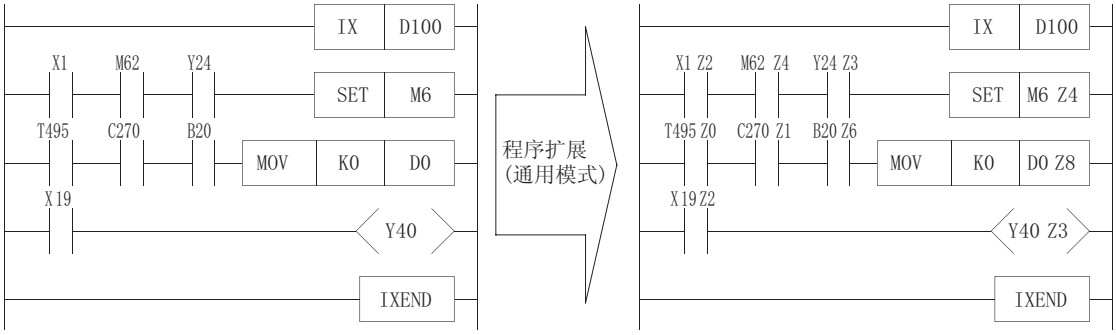


*2： IXEND 指令执行后，Zn 的值将返回到 IX 指令执行之前的原来的 Zn 值。

(b) 进行程序的展开时（使用 SW□-GPPQ 创建时）

展开程序时，通过 IX 和 IXEND 指令进行变址修饰的梯形图将被转换为使用了变址寄存器 (Zn) 的梯形图。 *3

不能通过 IX 和 IXEND 内的程序进行变址修饰。



*3： IXEND 指令执行后，Zn 的值将返回到 IX 指令执行之前的原来的 Zn 值。

要 点

1. 当使用 SW□ -GPPQ 编辑程序时（只对于 QnACPU），对于通过 IX 和 IXEND 指令进行变址修饰的梯形图，需要在 “通用模式” 下启动 SW□-GPPQ 并创建程序，执行 “程序展开”。如果在以下的 CPU 型号中启动 SW□-GPPQ，创建通过 IX 和 IXEND 指令进行变址修饰的梯形图时，将无法正常进行处理。
   - Q2A、 Q2A-S1、 Q2AS(H)、 Q2AS-S1
   - Q2AS(H)-S1、 Q3A、 Q4A、 Q4AR
2. 当在普通顺控程序和中断程序中都使用 IX 和 IXEND 指令时，应设立互锁以避免同时执行。互锁将普通顺控程序的 IX 和 IXEND 指令之间的区域作为 DI 而禁止中断。
3. IXDEV 和 IXSET 指令可以用来指定修饰值。
   关于详细内容，请参见 7.6.11 项。

运算错误

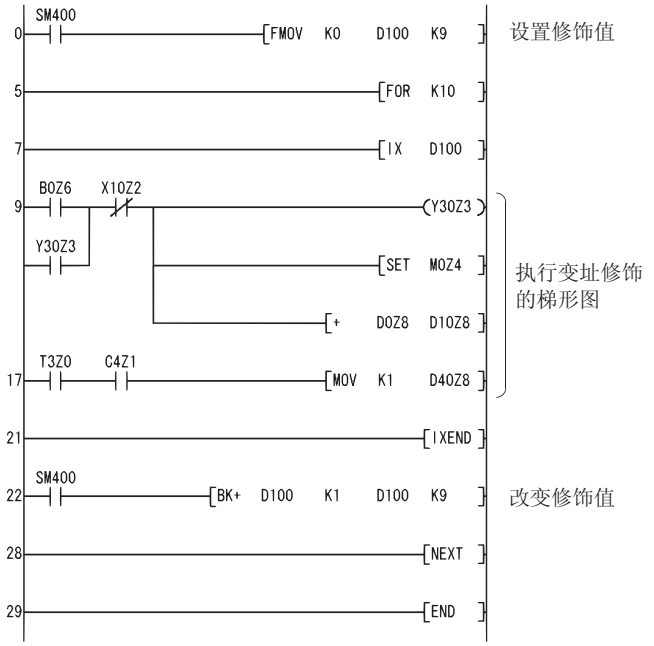
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- IX 和 IXEND 指令未一起使用。 （出错代码： 4231）
- 在 IXEND 指令执行之前，且在执行 IX 指令之后，执行 END、FEND、GOEND 或者 STOP 指令。 （出错代码： 4231）

程序示例

(1) 当改变软元件号时，以下程序执行相同梯形图 10 次。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	SM400			
1	FMOV	K0	D100	K9	
5	FOR	K10			
7	IX	D100			
9	LD	B0Z6			
10	OR	Y30Z3			
11	ANI	X10Z2			
12	OUT	Y30Z3			
13	SET	M0Z4			
14	+	D0Z8	D10Z8		
17	LD	T3Z0			
18	AND	C4Z1			
19	MOV	K1	D40Z8		
21	IXEND				
22	LD	SM400			
23	BK+	D100	K1	D100	K9
28	NEXT				
29	END				

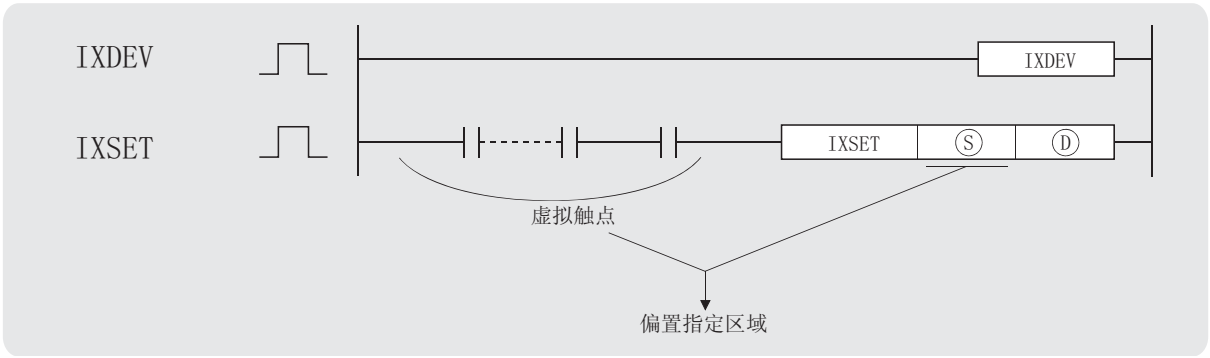
[动作]

	修饰值
D100	T的修饰值
D101	C的修饰值
D102	X的修饰值
D103	Y的修饰值
D104	M的修饰值
D105	L的修饰值
D106	B的修饰值
D107	V的修饰值
D108	D的修饰值

第1次	第2次	第3次	第10次
B0	B1	B2	B9
X10	X11	X12	X19
Y30	Y31	Y32	Y39
M0	M1	M2	M9
D0	D1	D2	D9
D10	D11	D12	D19
T3	T4	T5	T12
C4	C5	C6	C13
D40	D41	D42	D49

7.6.11 整个梯形图的变址修饰中修饰值的指定(IXDEV、IXSET)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 存储变址修改数据的软元件(仅指针) P□ 的起始软元件号
Ⓓ : 存储变址修改数据的软元件(除指针之外)的起始软元件号(BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P
Ⓢ	---	---		---		---			○
Ⓓ	---	○		---		---			---

功 能

(1) IXDEV 和 IXSET 指令用于创建 IX 和 IXEND 指令中使用的变址修饰表。

(2) 在 Ⓓ 指定的变址修改表上设定在偏置指定区域指定的软元件偏置值。

(3) 如果未进行值指定,则输入 0 值。

(4) 字软元件也通过触点(字软元件位指定)显示。
通过 D10.0 指定数据寄存器 10(D10)。
(从 0 到 F 的任何值都可用作位号。)

(5) 根据以下阐述的方法进行指定。 *1
(符号 □ 表示偏置值,符号 XX 表示随机选择。)

软元件	T	C	X	Y	M	L	V	B
指定方法	T□	C□	X□	Y□	M□	L□	V□	B□

软元件	D	W	R	U/G	J	ZR
指定方法	D□.XX	W□.XX	R□.XX	U□\G□.XX	J□\B□ *2	ZR□.XX

软元件	P
指定方法	IXSET Ⓢ Ⓓ *3

*1: 当使用基本型 QCPU 时,不能使用软元件 R、U/G、J、ZR 和 P。
*2: 将 J□\ 之后的软元件指定为 B、W、X 或 Y,并设置相应的偏置值。
*3: 使用基本型 QCPU 时,指定虚拟软元件号。 Ⓢ 为 P□ 。

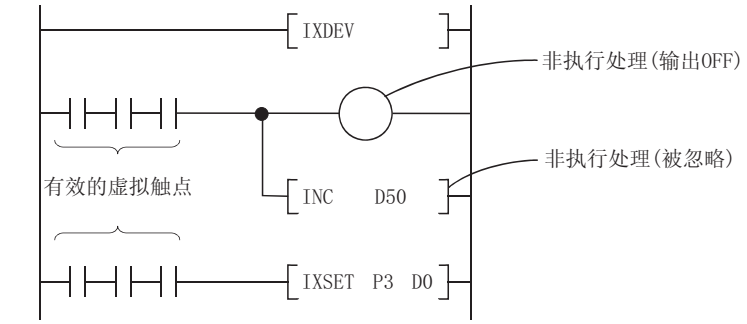
(6) 如果在偏置指定区域已经为两个相同类型的软元件设定了两个偏置，则最后设置的值有效。

(7) 应将 IXDEV 和 IXSET 指令视为一对指令。

(8) 任何从 0 到 32767 之间的 ZR 值都有效。
（偏置值是指定的软元件号除以 23768 所得的余数。）

(9) 在偏置指定区域的虚拟触点只对于 IXDEV-IXSET 指令范围内的 LD 和 AND 有效。如果记述了任何其它指令，IXDEV-IXSET 指令将不会执行。

示例



运算错误

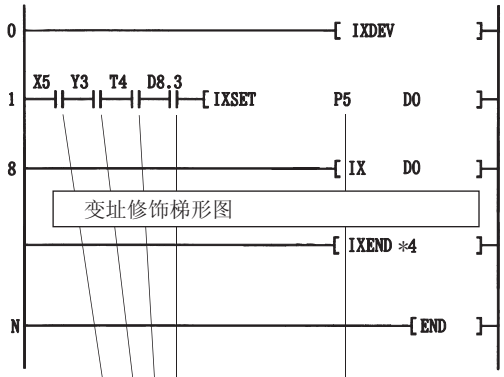
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• IXDEV 和 IXSET 指令未成对使用。 （出错代码： 4231）

程序示例

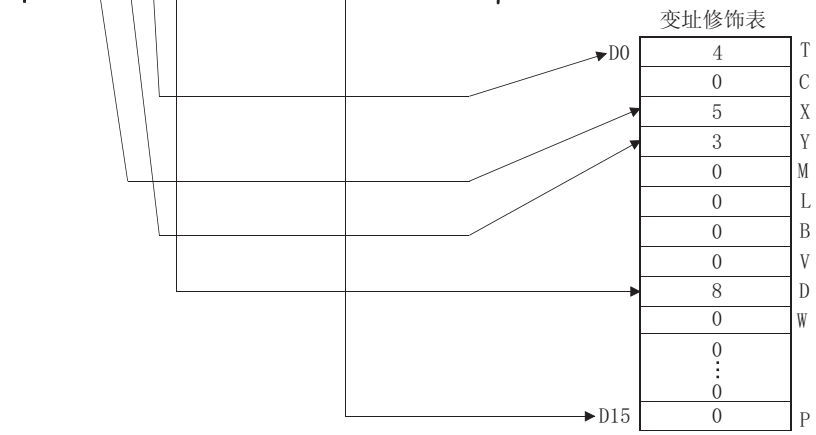
(1) 以下程序为输入 (X)、输出 (Y)、数据寄存器 (D) 和指针 (P) 更改修饰值。

[梯形图模式]



[列表模式]

步	指令	软元件
0	IXDEV	
1	LD	X5
2	AND	Y3
3	AND	T4
4	AND	D8.3
5	IXSET	P5
		D0
8	IX	D0
⋮	IXEND	
N	END	



*4: 关于使用 IX 到 IXEND 指令进行变址修饰，请参见 7.6.10 项。

7.7 数据表格操作指令

7.7.1 将数据写入数据表格 (FIFW(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：要写入表格中的数据或存储该数据的软元件号 (BIN 16 位)

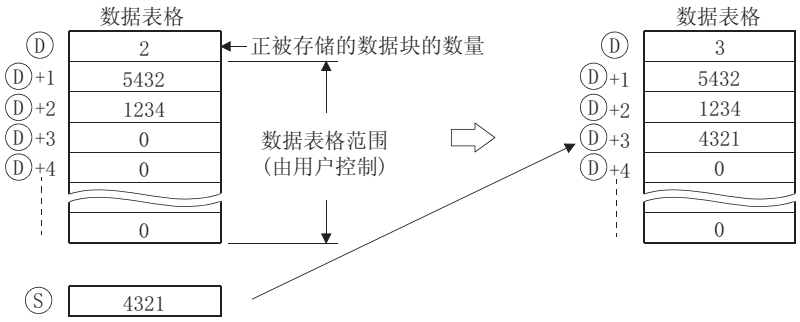
Ⓓ ：表格的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 将 Ⓢ 指定的 16 位数据存储在 Ⓓ 指定的数据表格中。

在表格中存储的数据块的数量存储在 Ⓓ 上，且将 Ⓢ 指定的数据按照顺序从 Ⓓ+1 处开始存储。



(2) 第一次执行 FIFW 指令时，应将指定软元件中的所有值都清除。

(3) 将要写入数据表格的数据块的数量和数据表格范围应由用户控制。

[参见程序示例 (2)]

运算错误

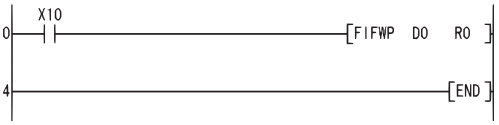
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

• 执行 FIFW 指令时，数据表格范围超出了相关软元件范围。 （出错代码： 4101）

程序示例

(1) 当 X10 为 ON 时，以下程序将数据存储在 R0 之后的 D0 上。

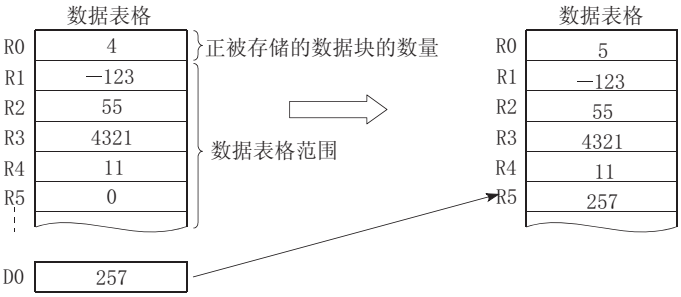
[梯形图模式]



[列表模式]

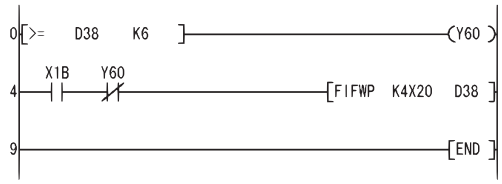
步	指令	软元件	
0	LD	X10	
1	FIFWP	D0	R0
4	END		

[动作]



(2) 当 X1B 为 ON 时，以下程序将 X20 到 X2F 的数据存储在 D38 到 D44 上，并且，如果有 6 个以上的数据块要存储，则将 Y60 变为 ON，禁用 FIFW 指令。

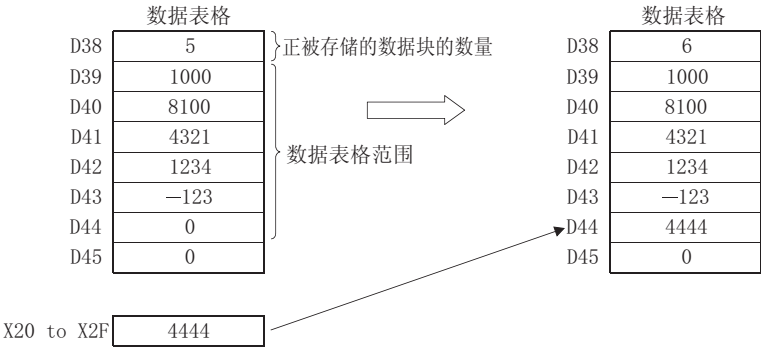
[梯形图模式]



[列表模式]

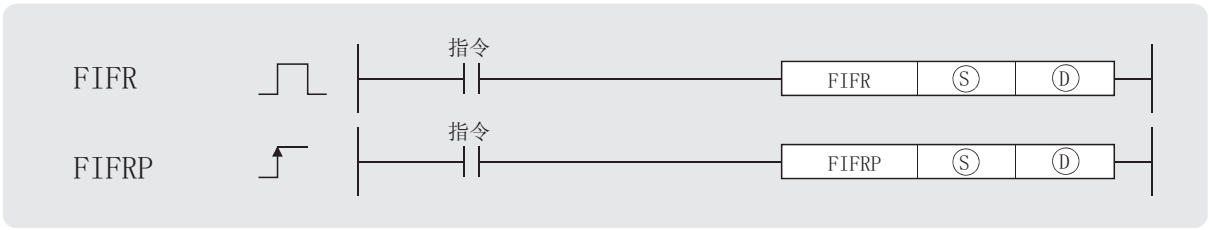
步	指令	软元件	
0	LD>=	D38	K6
3	OUT	Y60	
4	LD	X1B	
5	ANI	Y60	
6	FIFWP	K4X20	D38
9	END		

[动作]



7.7.2 从表格中读取最旧的数据 (FIFR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



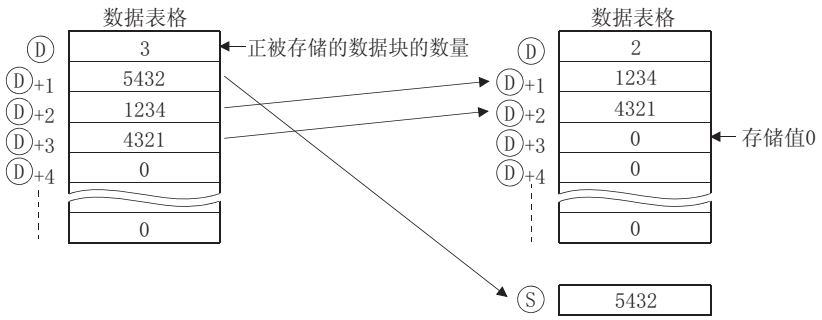
Ⓢ : 用于存储从表格中读取的数据的软元件的起始号 (BIN 16 位)

Ⓓ : 起始表格号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	○	○		○				---	
Ⓓ	---	○		---				---	

功 能

(1) 将输入到由 Ⓓ 指定的表格的最旧数据 (Ⓓ +1) 存储在 Ⓢ 指定的软元件上。
在执行 FIFR 指令之后，表格中的数据被向上压缩一个数据块。



(2) 如果存储在 Ⓓ 上的值为 0，用户应努力避免执行 FIFR 指令。[参见程序示例 (1)]

运算错误

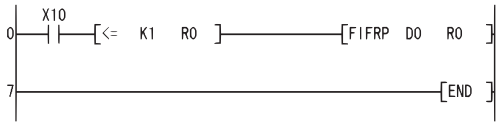
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

- 当 Ⓓ 上的值为 0 时执行了 FIFR 指令。 (出错代码 : 4100)
- 执行 FIFW 指令时，数据表格范围超出了相应软元件范围。 (出错代码 : 4101)

程序示例

(1) 当 X10 变为 ON 时，以下程序将占用 R0 到 R7 的数据表格中的在 R1 上保持的数据存储到目的点 D0 上。

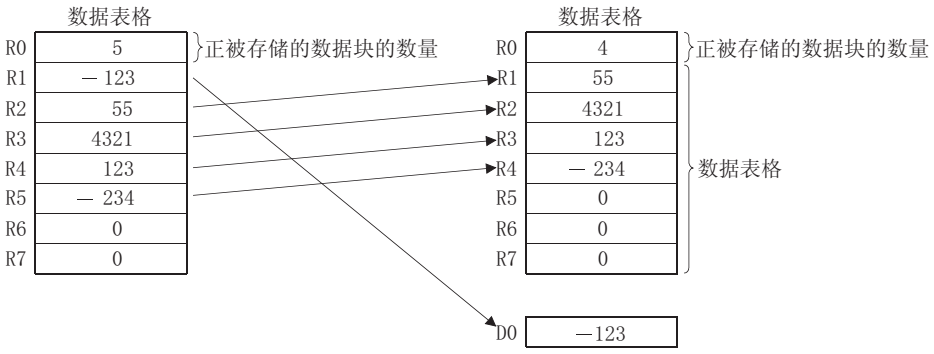
[梯形图模式]



[列表模式]

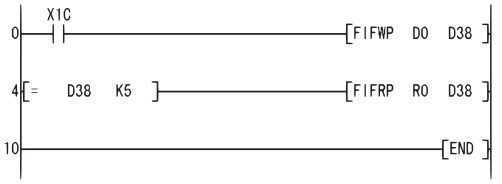
步	指令	软元件	
0	LD	X10	
1	AND<=	K1	R0
4	FIFRP	D0	R0
7	END		

[动作]



(2) 当 X1C 变为 ON 时，以下程序将 D0 中的数据存储到占用 D38 到 D43 的数据表格中，并且当表格有 5 个数据项时，将数据表格中在 D39 上保持的数据存储到 R0 上。

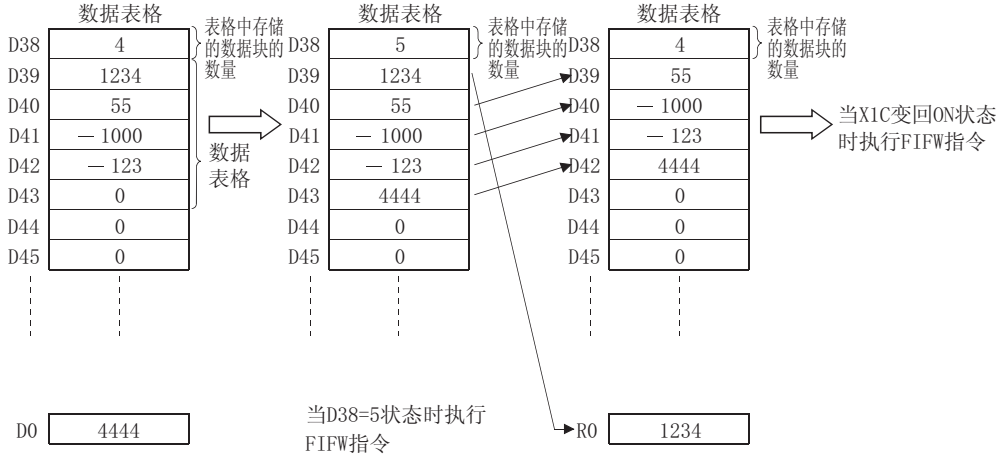
[梯形图模式]



[列表模式]

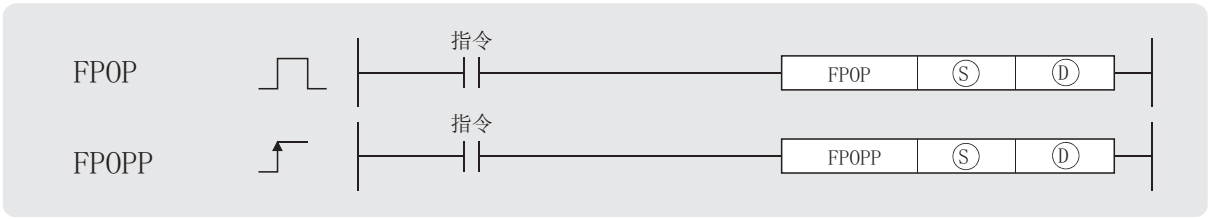
步	指令	软元件	
0	LD	X1C	
1	FIFWP	D0	D38
4	LD=	D38	K5
7	FIFRP	R0	D38
10	END		

[动作]



7.7.3 从数据表格中读取最新数据 (FPOP(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



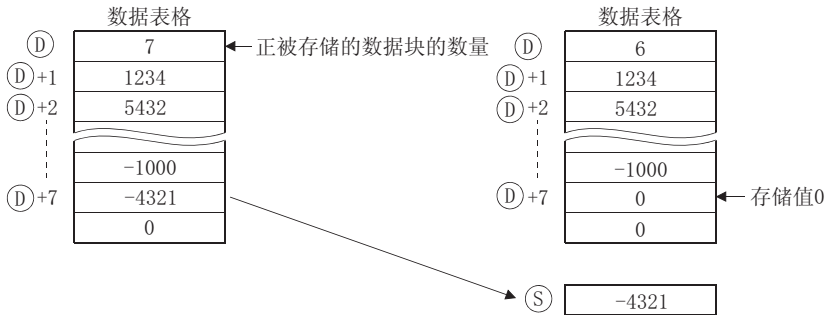
Ⓢ ：用于存储从表格中读取的数据的软元件的起始号 (BIN 16 位)

Ⓓ ：起始表格号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	○	○		○				---	
Ⓓ	---	○		---				---	

功 能

(1) 将输入到 Ⓓ 指定的表格的最新数据存储到 Ⓢ 指定的软元件上。
在执行 FPOP 指令之后，保存由 FPOP 指令读取的数据的软元件被复位到 0。



(2) 如果存储在 Ⓓ 上的值为 0，用户应努力避免执行 FPOP 指令。[参见程序示例 (1)]

运算错误

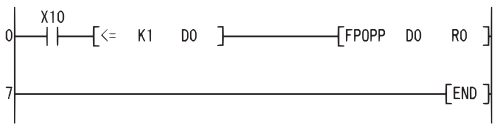
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

- 当 Ⓓ 值为 0 时，执行了 FPOP 指令。 (出错代码： 4100)
- 执行 FPOP 指令时，数据表格范围超出了对应软元件的范围。 (出错代码： 4101)

程序示例

(1) 当 X10 为 ON 时，以下程序将占用 R0 到 R7 的数据表格中存储的最新数据存储到 D0 上。

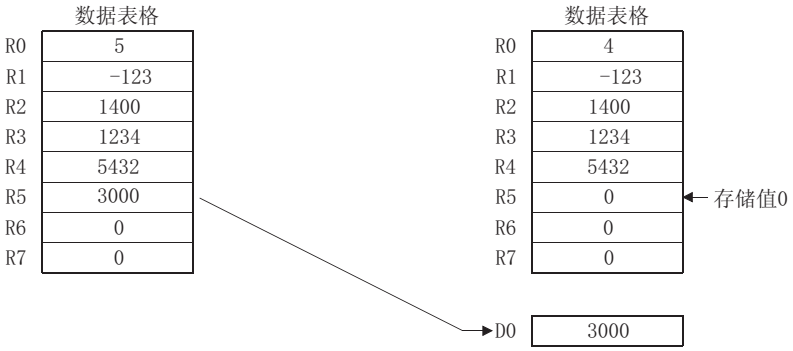
[梯形图模式]



[列表模式]

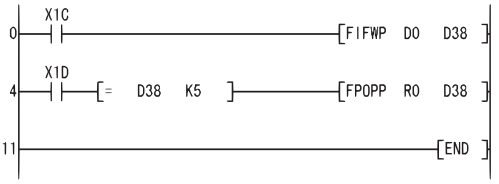
步	指令	软元件	
0	LD	X10	
1	AND<=	K1	D0
4	FPOPP	D0	R0
7	END		

[动作]



(2) 当 X1C 变为 ON 时，以下程序将原先存储在 D0 中的数据存储到占用 D38 到 D43 的数据表格中，并且当表格中的数据项数达到 5 时，将 X1D 变为 ON，将存放在数据表格中的最新数据存储到 R0 上。

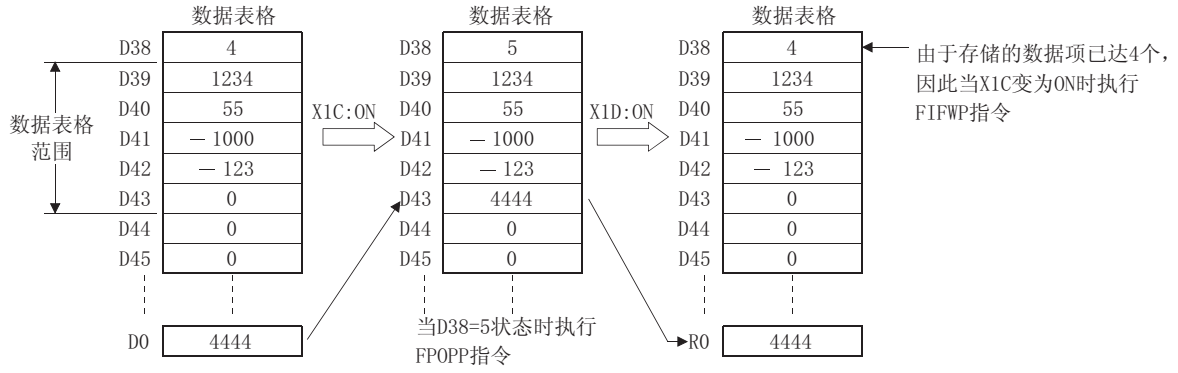
[梯形图模式]



[列表模式]

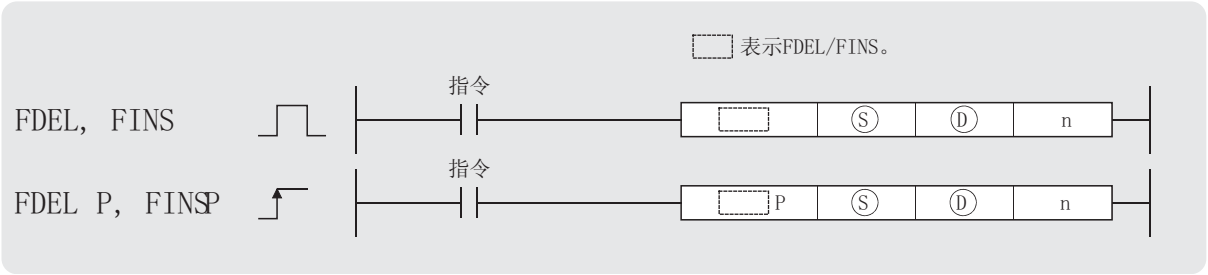
步	指令	软元件	
0	LD	X1C	
1	FIFWP	D0	D38
4	LD	X1D	
5	AND=	D38	K5
8	FPOPP	R0	D38
11	END		

[动作]



7.7.4 从数据表格中删除数据和在数据表格中插入数据 (FDEL(P)、FINS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



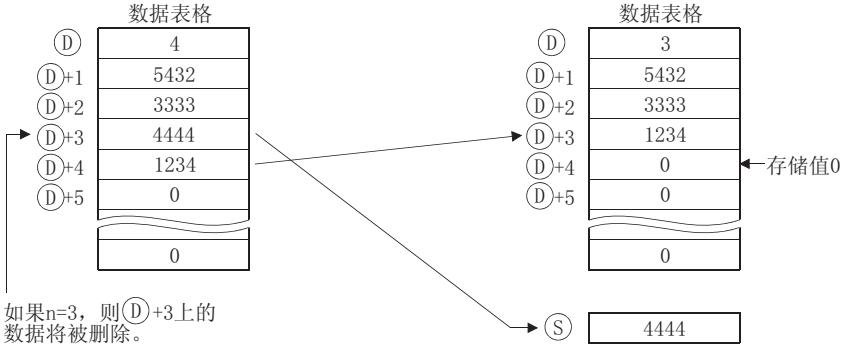
Ⓢ ：插入的数据或用于存储插入数据的软元件的起始号（BIN 16 位）
　　用于存储删除数据的软元件的起始号（BIN 16 位）
Ⓓ ：表格的起始号（BIN 16 位）
n ：表格中存放插入或删除数据的位置（BIN 16 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○				---	---
Ⓓ	---	○		---				---	---
n	○	○		○				○	---

功 能

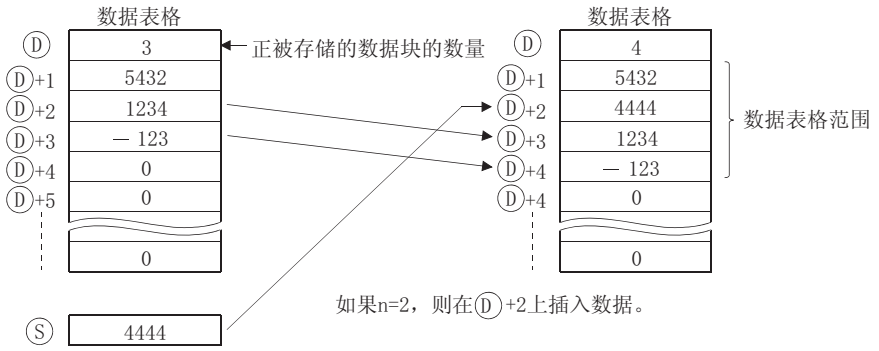
FDEL

(1) 从 Ⓓ 指定的数据表格中删除第 n 个数据，并将其存储到 Ⓢ 指定的软元件中。
在执行 FDEL 指令之后，表格中被删除的数据之后的数据分别向前填充对齐。



FINS

(1) 将 ⓢ 指定的 16 位数据插入到由 ⓓ 指定的数据表格的第 n 个数据块上。
在执行 FINS 指令之后，表格中在插入的数据块之后的数据被全部下调一个位置。



运算错误

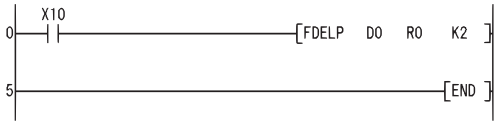
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

- 使用 FDEL 指令时，从 ⓓ 开始的第 n 个位置大于表格中数据存储的数量。
  (出错代码： 4101)
- 使用 FINS 指令时，从 ⓓ 开始的第 n 个位置大于 “数据存储的数量 +1”。
  (出错代码： 4101)
- 使用 FDEL、FINS 指令时，值 n 超出了表格 ⓓ 的软元件范围。 (出错代码： 4101)
- 当 n=0 时，执行了 FDEL 或 FINS 指令。 (出错代码： 4100)
- 当 ⓓ 的值为 0 时，执行了 FDEL 指令。 (出错代码： 4100)
- 在执行 FDEL 或 FINS 指令时，数据表格范围超出了对应软元件的范围。
  (出错代码： 4101)

程序示例

(1) 当 X10 变为 ON 时，以下程序将从占用 R0 到 R7 的表格的第 2 个位置删除数据，并将此数据存储到 D0 上。

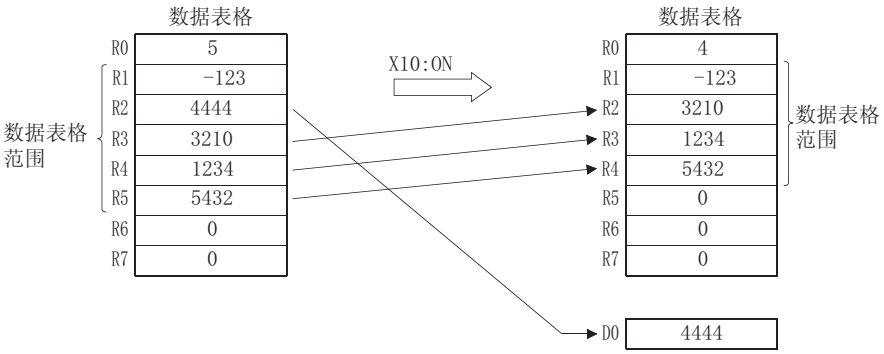
[梯形图模式]



[列表模式]

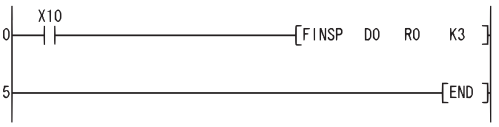
步	指令	软元件		
0	LD	X10		
1	FDELP	D0	R0	K2
5	END			

[动作]



(2) 当 X10 为 ON 时，以下程序将 D0 上的数据插入到占用 R0 到 R7 的表格的第 3 个数据块上。

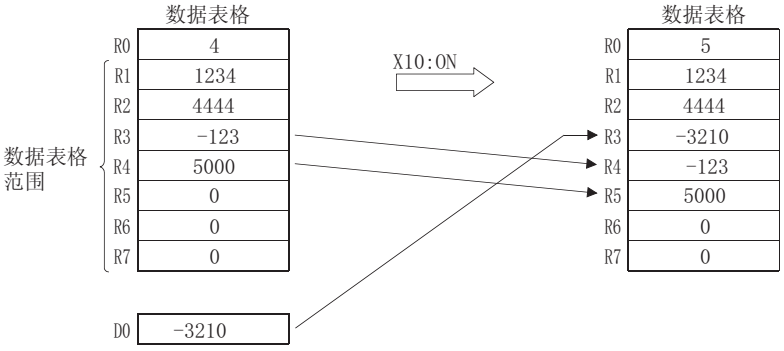
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X10		
1	FINSP	D0	R0	K3
5	END			

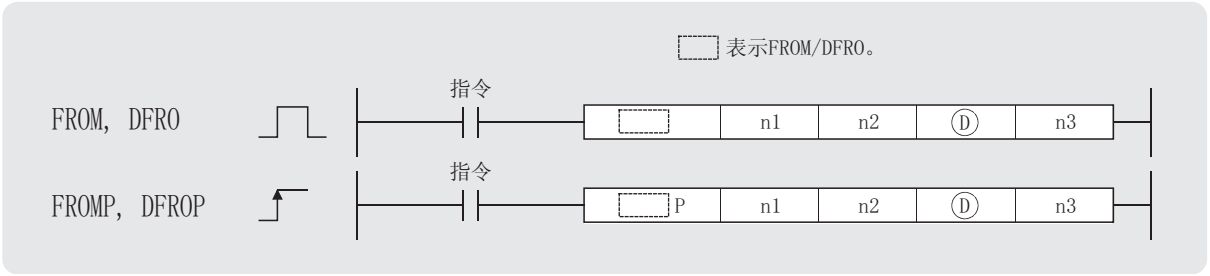
[动作]



7.8 缓冲存储区访问指令

7.8.1 从智能功能模块中读取 1 字 /2 字数据 (FROM(P)、DFRO(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



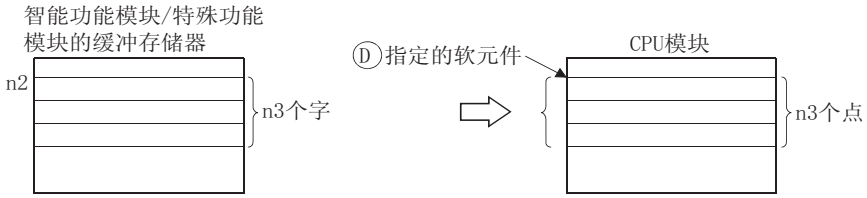
n1 : 智能功能模块 / 特殊功能模块的起始 I/O 号 (BIN 16 位)
n2 : 被读取数据的起始地址 (BIN 16 位)
Ⓓ : 存储被读取数据的软元件的起始号 (BIN 16/32 位)
n3 : 被读取数据的数量 (BIN 16 位)
- FROM(P) : 1 到 6144
- DFRO(P) : 1 到 3072(仅适用于 A/QnA)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它 U
	位	字		位	字				
n1		○				○			○
n2		○				○			---
Ⓓ		○				---			---
n3		○				○			---

功 能

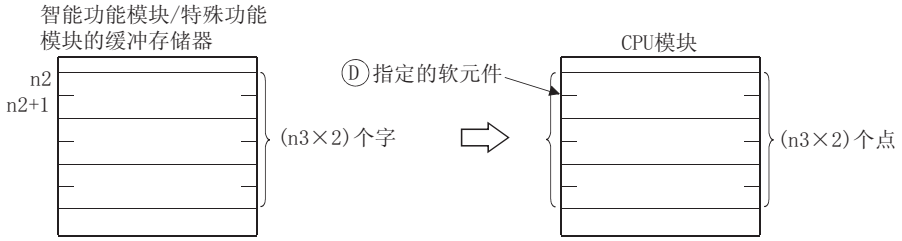
FROM

(1) n1 指定的智能功能模块 / 特殊功能模块的缓冲存储器中，从由 n2 指定的地址开始读取 n3 个字的数据。
然后，将数据存储在从 Ⓓ 指定的软元件开始的区域。



DFRO

(1) 在 n1 指定的智能功能模块 / 特殊功能模块的缓冲存储器中，从 n2 指定的地址开始读取 (n3 × 2) 字的数据。并将数据存储在从 ⓓ 指定的软元件开始的区域。



要 点

智能功能模块 / 特殊功能模块的软元件也可以用于从智能功能模块 / 特殊功能模块软元件的缓冲存储器上读取数据。

智能功能模块 / 特殊功能模块软元件的详细情况，请参考以下手册：

- QCPU 模块的用户手册 (功能解说 / 程序基础篇)
- QnACPU 编程手册 (基础篇)

运算错误

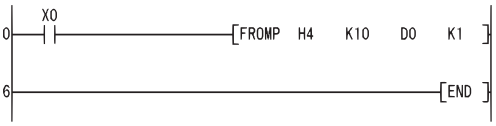
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

- 在执行指令时，没有同智能功能模块 / 特殊功能模块进行信号交换。 (出错代码： 1412)
- 在执行指令时，智能功能模块 / 特殊功能模块上检测到错误。 (出错代码： 1402)
- n1 指定的 I/O 号不是用于智能功能模块 / 特殊功能模块。 (出错代码： 2110)
- ⓓ 指定的软元件的 n3(对于 DFRO 是 2 × n3 个点) 个点超过了指定软元件的范围。 (出错代码： 4101)
- n2 指定的地址超出了缓冲存储器的范围。 (出错代码： 4101)
- n2 指定的地址是一个奇数地址。(AJ71QC24(N)) (出错代码： 4100)
- 读取数据的数量超出了 6144/3072。(仅适用于 A/QnA 模块) (出错代码： 4101)

程序示例

(1) 当 X0 变为 ON 时，以下程序将安装在 I/O 号 040 到 05F 的 A68AD 的 CH1 数字值读到 D0 上。

[梯形图模式]



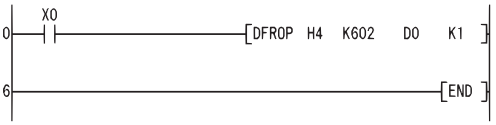
[列表模式]

步	指令	软元件			
0	LD	X0			
1	FROMP	H4	K10	D0	K1
6	END				

(2) 当 X0 变为 ON 时，以下程序将安装在 I/O 号 040 到 05F 的 AD71 的 X 轴的当前值读取到 D0 和 D1 上。

（从缓冲存储器地址 602、608 读取 2 字的数据）。

[梯形图模式]

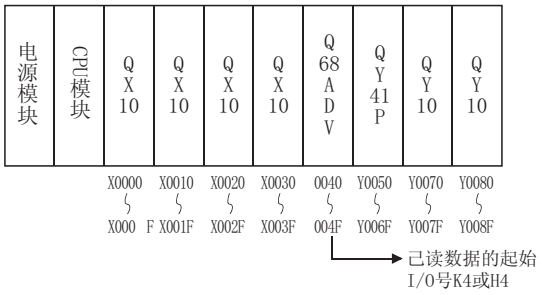


[列表模式]

步	指令	软元件			
0	LD	X0			
1	DFROP	H4	K602	D0	K1
6	END				

备 注

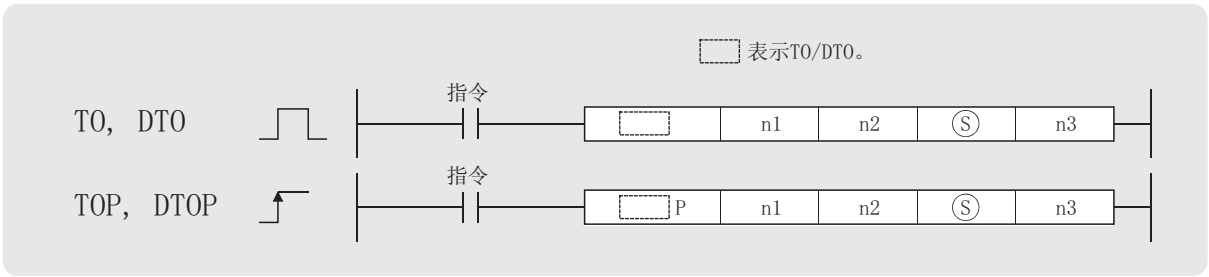
1. 当以十六进制 4 位数表示安装智能功能模块 / 特殊功能模块的插槽的起始 I/O 号时，n1 的值由高三位指定。



2. QCPU 建立了 FROM/DFRO 指令的自动互锁。

7.8.2 将 1 字 /2 字数据写入智能功能模块 (TO(P)、DTO(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



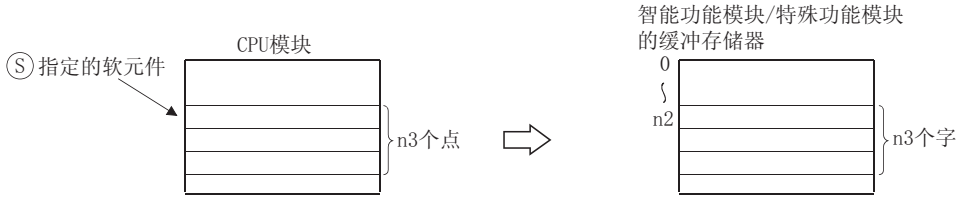
n1 ：智能功能模块 / 特殊功能模块的起始 I/O 号 (BIN 16 位)
n2 ：数据写入操作的起始地址 (BIN 16 位)
Ⓢ ：存储要写入数据的软元件的起始号 (BIN 16/32 位)
n3 ：写入数据的数量 (BIN 16 位)
- TO(P) ：1 到 6144
- DTO(P) ：1 到 3072(仅适用于 A/QnA)

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它 U
n1	○			○				○	○
n2	○			○				○	---
Ⓢ	○			---				○	---
n3	○			○				○	---

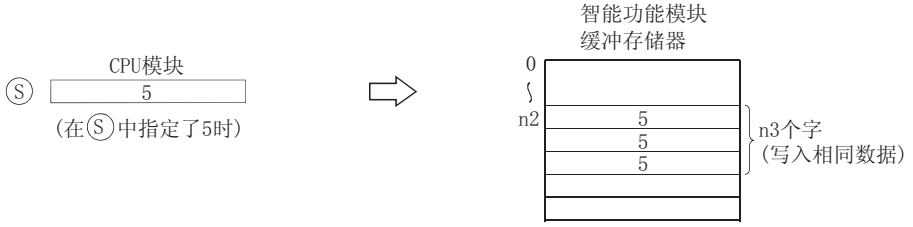
功 能

TO

从 Ⓢ 指定的软元件上将 n3 个点的数据写入到 n1 指定的智能功能模块 / 特殊功能模块的缓冲存储器中从 n2 指定的地址开始的区域内。

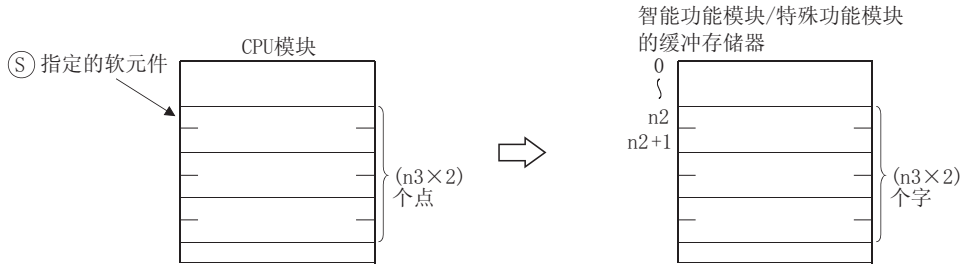


当为 Ⓢ 指定了一个常数时，将相同数据（指定给 S 的值）写入到指定缓冲存储器起始的 n3 个字区域。(Ⓢ 的指定范围为 :-32768 到 32767 或 0H 到 FFFFH。)

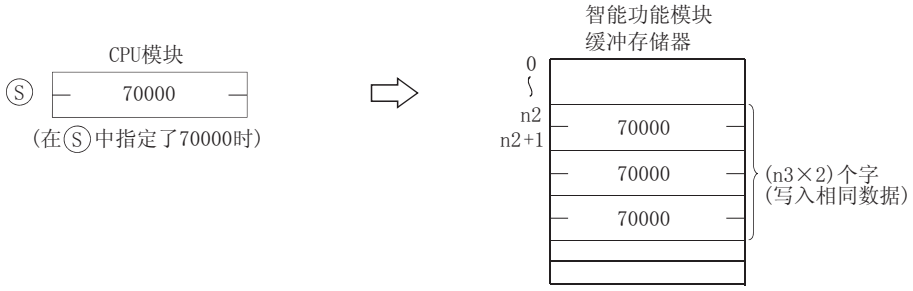


DTO

从 Ⓢ 指定的软元件上将 (n3 × 2) 个点的数据写入到 n1 指定的智能功能模块 / 特殊功能模块的缓冲存储器中从 n2 指定的地址开始的区域内。



当为 Ⓢ 指定了一个常数时，将相同数据（指定给 Ⓢ 的值）写入到指定缓冲存储器起始的 (n3 × 2) 个字的区域中。(Ⓢ 的指定范围为 :-2147483648 到 2147483647 或 0H 到 FFFFFFFFH。)



要 点

智能功能模块 / 特殊功能模块的软元件也可用于从智能功能模块 / 特殊功能模块软元件的缓冲存储器上读取数据。
智能功能模块 / 特殊功能模块软元件的详细情况，请参考以下手册 :

- QCPU 模块的用户手册（功能解说 / 程序基础篇）
- QnACPU 编程手册（基础篇）

运算错误

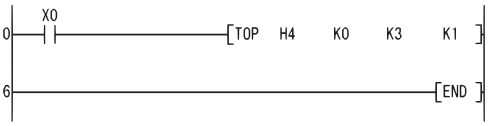
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

- 在执行指令时，没有同智能功能模块 / 特殊功能模块进行信号交换。 (出错代码 : 1412)
- 在执行指令时，智能功能模块 / 特殊功能模块上检测到错误。 (出错代码 : 1402)
- n1 指定的 I/O 号不是用于智能功能模块 / 特殊功能模块。 (出错代码 : 2110)
- Ⓢ 指定的软元件的 n3（对于 DTO 是 2 × n3 个点）个点超过了指定软元件的范围。 (出错代码 : 4101)
- n2 指定的地址超出了缓冲存储器的范围。 (出错代码 : 4101)
- n2 指定的地址是一个奇数地址 (AJ71QC24(N))。 (出错代码 : 4100)
- 读取数据的数量超出了 6144/3072。（仅适用于 A/QnA 模块） (出错代码 : 4101)

程序示例

(1) 当 X0 变为 ON 时，以下程序将安装在 I/O 号 040 到 04F 的 Q68ADV 的 CH1 和 CH2 设置 为 “A/D 转换”。(将 “3” 写入到缓冲存储器的地址 0)

[梯形图模式]

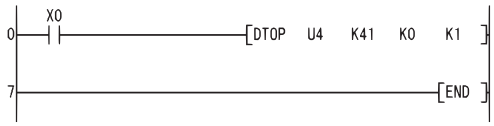


[列表模式]

步	指令	软元件
0	LD	X0
1	TOP	H4 K0 K3 K1
6	END	

(2) 当 X0 变为 ON 时，以下程序将安装在 I/O 号 040 到 05F 的 AD71 的 X 轴的当前值设置为 0。(将 0 写入缓冲存储器的地址 41、42)。

[梯形图模式]

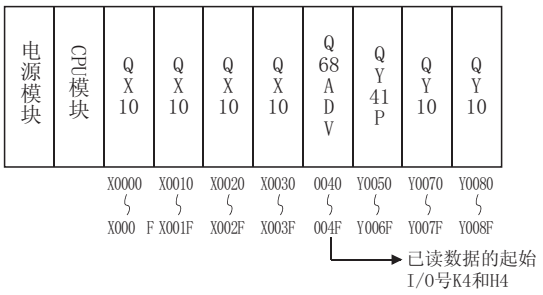


[列表模式]

步	指令	软元件
0	LD	X0
1	DTOP	U4 K41 K0 K1
7	END	

备 注

1. 当以十六进制 4 位数表示安装智能功能模块 / 特殊功能模块的插槽的起始 I/O 号时，n1 的值由高三位指定。



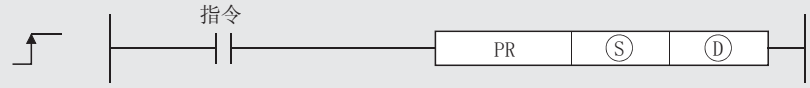
2. QCPU 建立 TO/DTO 指令的自动互锁。

7.9 显示指令

7.9.1 打印 ASCII 码指令 (PR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

PR 指令



Ⓢ : 存储 ASCII 码的软元件的起始号（字符串）

Ⓓ : 输出 ASCII 码的输出模块的起始号（位）

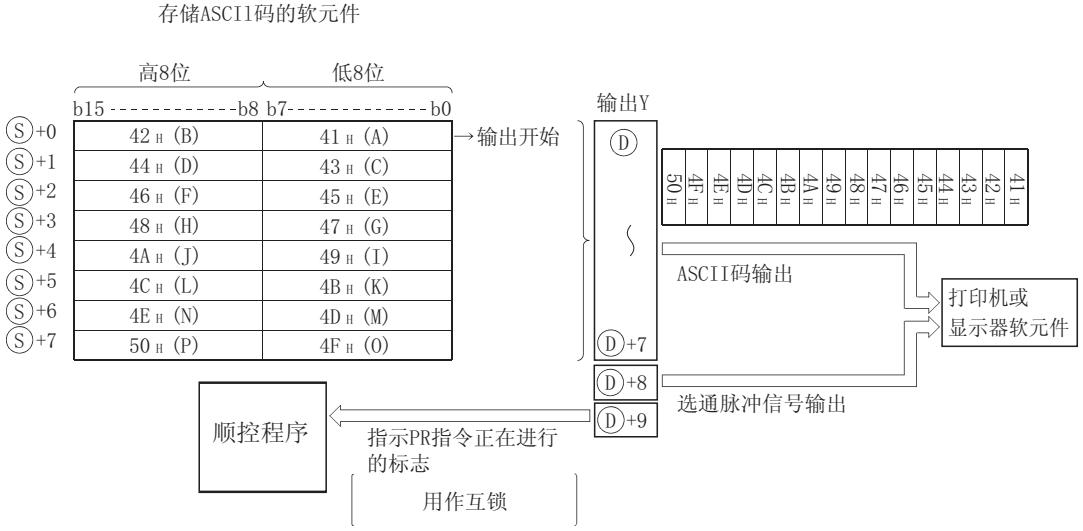
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	△*1		---			○	○	---
Ⓓ	○（限于 Y）	---		---			○	---	---

*1: 不能使用局部软元件和为各程序设置的文件寄存器。

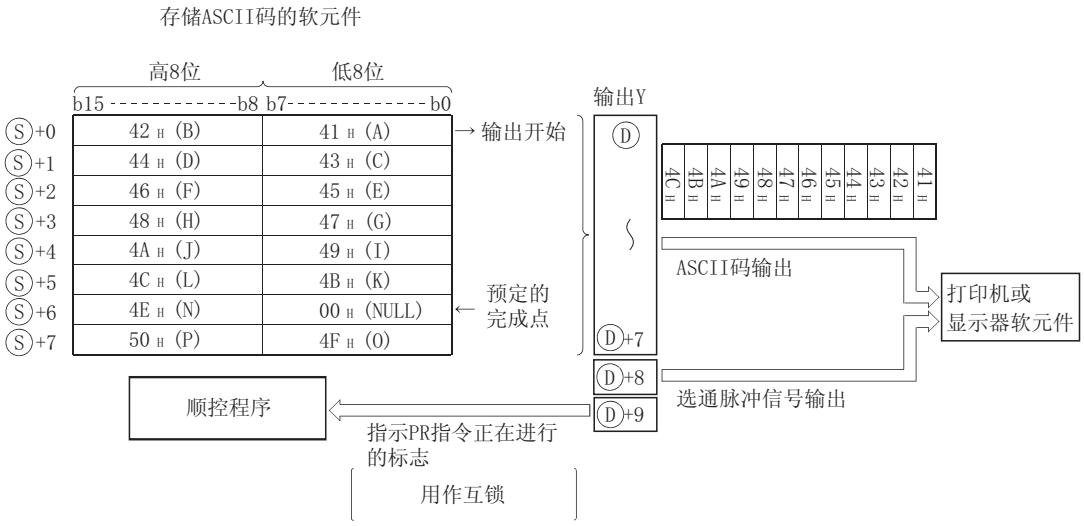
功 能

(1) 将存储在 Ⓢ 指定的软元件之后的 ASCII 码输出到 Ⓓ 指定的一个输出模块上。
输出字符数根据 SM701 的 ON/OFF 状态会有不同（输出字符数选择）。

(a) 如果 SM701 为 ON，将对从 Ⓢ 指定的软元件开始的 8 个点的字符（16 字符）进行运算。

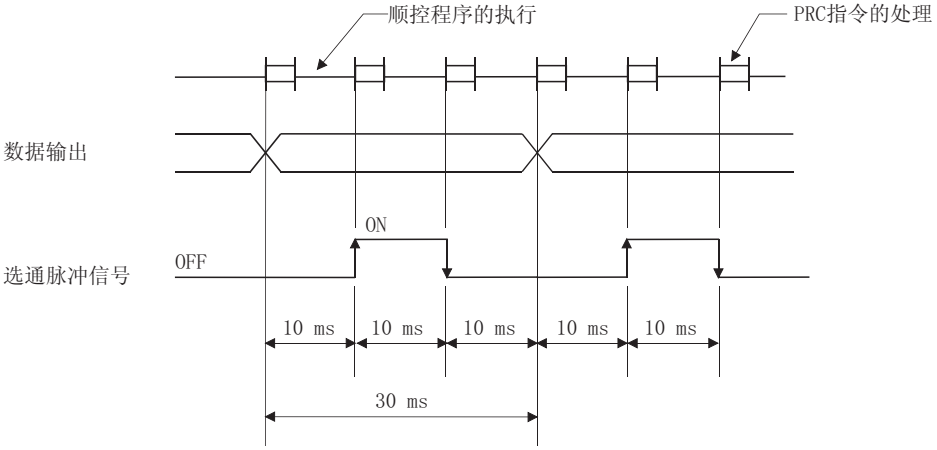


(b) 如果 SM701 为 OFF，则从 ⓢ 指定软元件到 00H 码之间的的有数据都将作为操作焦点。



(2) 输出模块使用的点数是从 ⓓ 指定的 Y 地址开始的 10 点。

(3) 来自输出模块的输出信号以每 30ms 一个字符的速率进行传送。
因此，完成 (n) 个指定字符数的传送所需要的时间是 30ms × n(ms)。
每隔 10ms，PR 指令执行数据输出，选通脉冲信号为 ON，然后选通脉冲信号变为 OFF。在以上处理之间的时间，继续执行其它指令。



(4) 除了 ASCII 码之外，输出模块也从 ⓓ + 8 软元件上输出选通脉冲信号。

(5) 在执行 PR 指令之后，PR 指令执行标志（ⓓ + 9 软元件）保持 ON 状态，直到完成指定字符数的传送。

(6) PR 和 PRC 指令可多次使用，但是最好用 PR 指令的执行标志（ⓓ + 9 软元件）建立互锁，这样它们就不会同时处于 ON 状态。

(7) 如果在 ASCII 码的输出期间，存储 ASCII 码的软元件的内容被更改，则将输出更新的数据。

运算错误

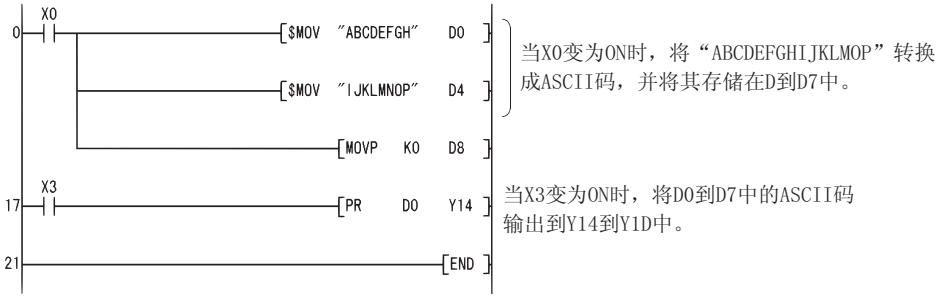
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

• 当 SM701 为 OFF 时，在 Ⓢ 指定的软元件范围内没有 00H 码。 （出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序将字符串 “ABCDEFGHIJKLMNOP”转换成 ASCII 码，并将其存储在 D0 到 D7 上，然后当 X3 变为 ON 时，将 D0 到 D7 上的 ASCII 码输出到 Y14 到 Y1D 上。

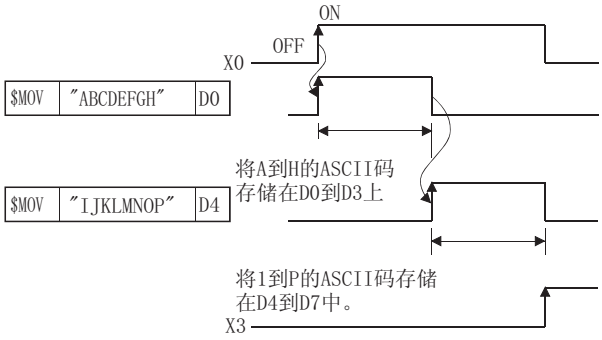
[梯形图模式]

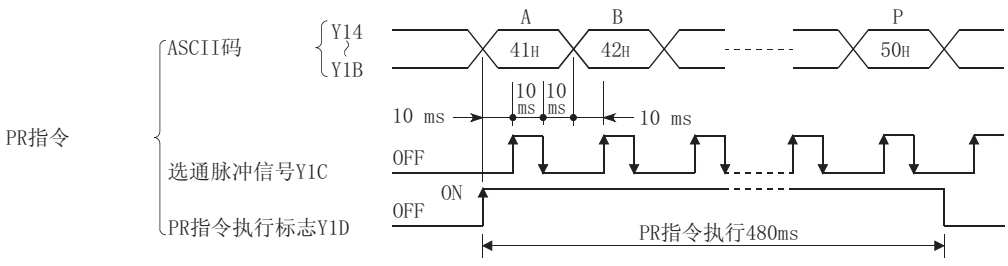


[列表模式]

步	指令	软元件	
0	LD	X0	
1	$MOV	"ABCDEFGH"	D0
8	$MOV	"IJKLMNOP"	D4
15	MOVP	K0	D8
17	LD	X3	
18	PR	D0	Y14
21	END		

[时序图]





7.9.2 打印注释指令 (PRC)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

1 当使用高性能型 QCPU/ 过程控制 CPU 时。



Ⓢ ：要打印注释的软元件的起始号（软元件名称）

Ⓓ ：要输出注释的输出模块的起始号（位）

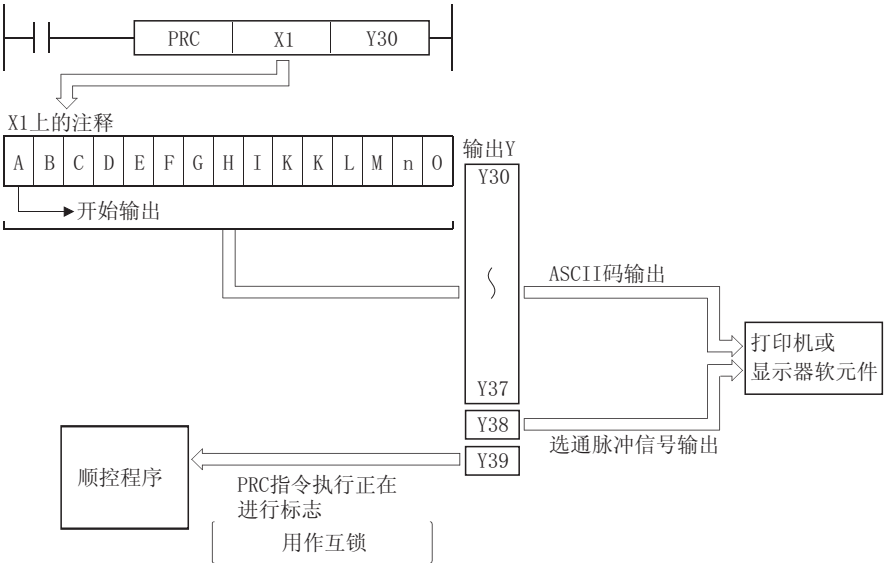
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P、I、J、U
Ⓢ	○	○		○			---	---	○
Ⓓ	○ (仅 Y)	---		---			---	---	---

功 能

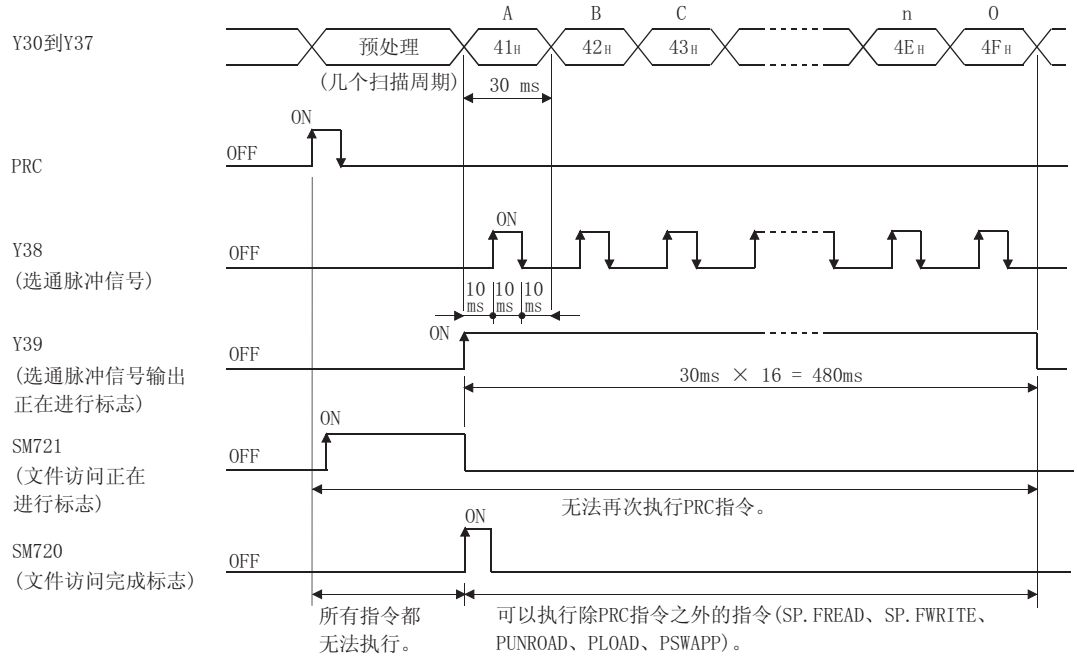
(1) 将 Ⓢ 指定的软元件上的注释 (ASCII 码）输出到 Ⓓ 指定的输出模块上。
输出字符数将根据 SM701 的 ON/OFF 状态有所不同。

- 当 SM701 为 OFF 时 ： 注释是 32 个字符
- 当 SM701 为 ON 时 ： 注释是高 16 个字符

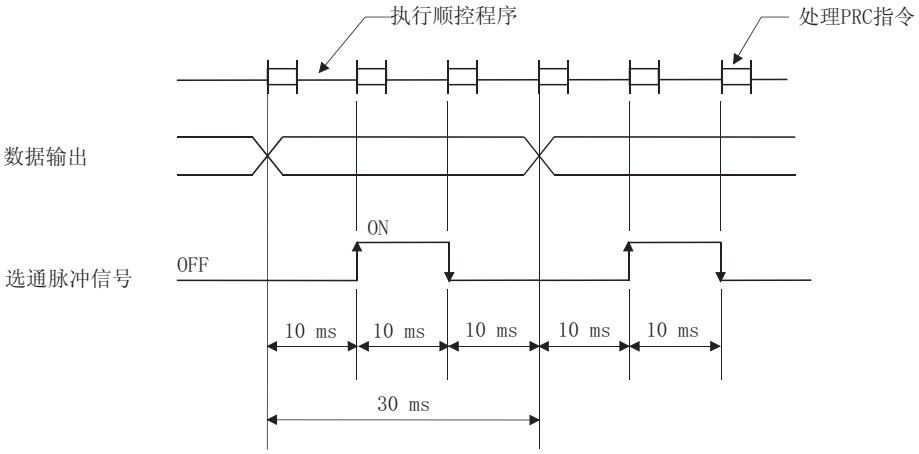
输出模块使用的点数是从 Ⓓ 指定的 Y 地址开始的 10 个点。



[时序图]



(2) 来自输出模块的输出信号以每 30ms 一个字符的速率进行传送。
因此，完成 (n) 个指定字符数的传送需要的时间是 30 × n(ms)。
每隔 10ms，PRC 指令执行数据输出，选通脉冲信号为 ON，然后选通脉冲信号为 OFF。在以上处理之间的时间，继续执行其它指令。



(3) 除了 ASCII 码之外，输出模块也从 (D) + 8 软元件上输出选通脉冲信号 (10msON、20msOFF)

(4) 执行 PRC 指令之后，PRC 指令执行标志 ((D) + 9 软元件) 保持 ON 状态，直到完成指定字符数的传送。

(5) PRC 指令可多次使用，但是最好用 PRC 指令执行标志 ((D) + 9 软元件) 建立互锁，这样它们就不会同时处于 ON 状态。

(6) 如果没有在 (S) 指定的软元件上登记注释，则不会进行处理。

(7) 当注释被读取时，指令完成后 SM720 变为 ON 一个扫描周期。SM721 在指令执行期间变为 ON。当 SM721 为 ON 时，无法执行 PRC 指令，如果试图执行，则不会进行处理。

要 点

1. PRC 指令使用的软元件注释使用存放在存储卡上的注释文件。
   它们不能使用内存中的注释文件。
2. 在参数模式中的 “可编程控制器文件设置”选项中设置 PRC 指令使用的注释文件。
   如果未给可编程控制器文件设置设定它所使用的注释文件，则无法使用 PRC 指令输出软元件注释。
3. 不要在中断程序期间执行 PRC 指令。
   否则，全导致故障出现。

运算错误

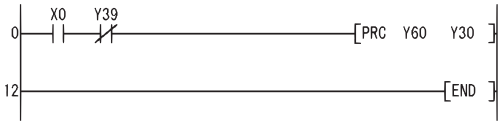
(1) 在发生以下运行错误的情况中，错误标志 (SM0) 变为 ON，且出错代码存储在 SD0 上。

• 在 RUN 期间当写入注释时执行 PRC 指令。 （出错代码： 4100）

程序示例

(1) 当 X0 变为 ON 时，程序将 Y60 的注释输出到 Y30 到 Y39 上。

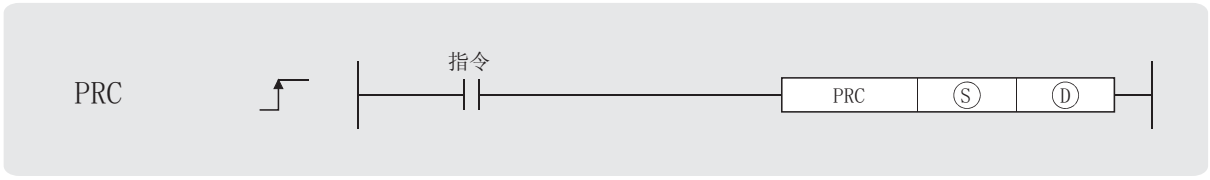
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	ANI	Y39
2	PRC	Y60 Y30
5		<;y60="aa">
12	END	

2 当使用 QnACPU/Q4ARCP 时



Ⓢ ：打印注释的软元件的起始号（软元件名称）

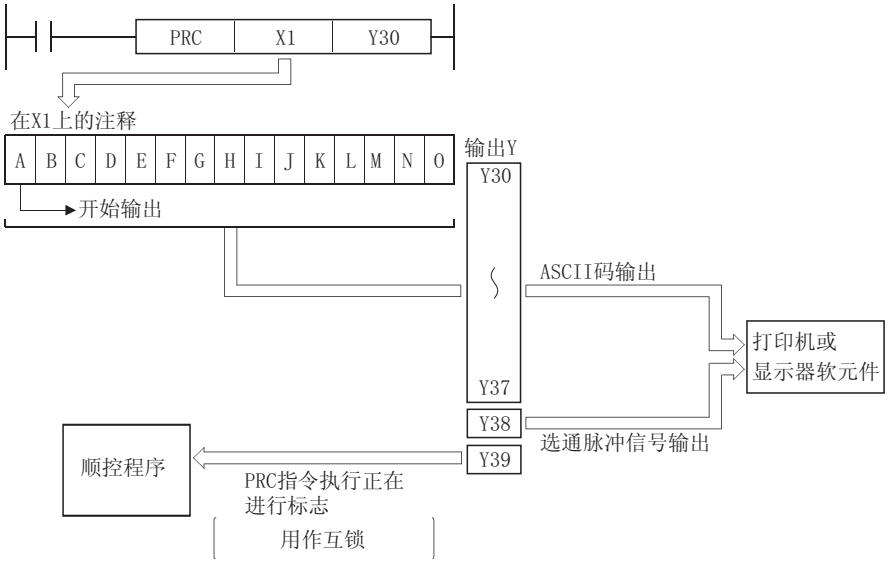
Ⓓ ：输出注释的输出模块的起始号（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				P、I、J、U、
Ⓢ	○	○		○			---	---	○
Ⓓ	○（仅 Y）	---		---			---	---	---

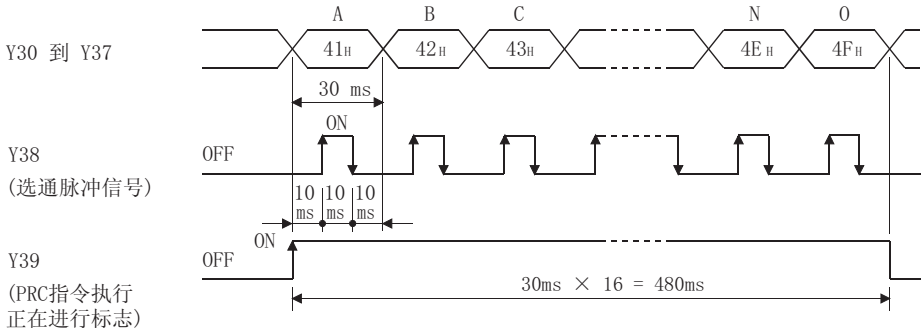
功 能

(1) 将 Ⓢ 指定的软元件上的注释（ASCII 码）输出到 Ⓓ 指定的输出模块上。
输出字符的数量将根据 SM701 的 ON/OFF 状态而有所不同。

- 当 SM701 为 OFF 时 ：注释是 32 字符
- 当 SM701 为 ON 时 ：注释是高 16 字符
  输出模块使用的点数是从 Ⓓ 指定的 Y 地址开始的 10 点。



[时序图]



(2) 来自输出模块的输出信号以每 30ms 一个字符的速率进行传送。
因此，完成 (n) 个指定字符数的传送需要的时间是 30ms × n(ms)。
每隔 10ms，PRC 指令执行数据输出，选通脉冲信号为 ON，然后选通脉冲信号为 OFF。在以上处理之间的时间，继续执行其它指令。

(3) 除了 ASCII 码之外，输出模块也从 ⓓ + 8 软元件上输出选通脉冲信号 (10msON、20msOFF)。

(4) 在执行 PRC 指令之后，PRC 指令执行标志 (ⓓ + 9 软元件) 保持 ON 状态，直到完成指定字符数的传送。

(5) PRC 指令可多次使用，但是最好用 PRC 指令执行标志 (ⓓ + 9 软元件) 建立互锁，这样它们就不会同时处于 ON 状态。

(6) 如果没有在 ⓢ 指定的软元件上登记注释，则处理将不会进行。

要 点

1. PRC 指令使用的软元件注释使用存放在存储卡上的注释文件。
不能使用内存中的注释文件。
2. 在参数模式中的 “可编程控制器文件设置” 选项中设置 PRC 指令使用的注释文件。
如果没有给可编程控制器文件设置它所使用的注释文件，则无法使用 PRC 指令输出软元件注释。

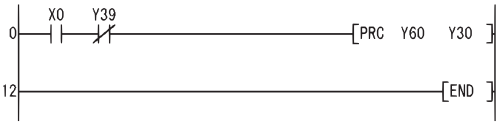
运算错误

(1) 没有同 PRC 指令相关的运行错误。

程序示例

(1) 当 X0 变为 ON 时，程序将 Y60 的注释输出到 Y30 到 Y39 上。

[梯形图模式]

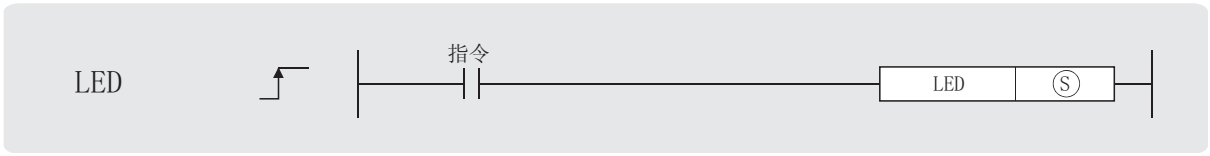


[列表模式]

步	指令	软元件
0	LD	X0
1	ANI	Y39
2	PRC	Y60 Y30
5		<:y60="aa">
12	END	

7.9.3 ASCII 码 LED 显示指令 (LED)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 存储显示数据的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---			---	○	---

功 能

(1) 在 CPU 模块前部的 LED 显示器上显示存储在从由 Ⓢ 指定的软元件开始的 8 个数据点的 ASCII 数据 (16 字符)。

显示数据

	b15----------b8	b7----------b0
Ⓢ	42H (B)	41H (A)
Ⓢ+1	44H (D)	43H (C)
Ⓢ+2	46H (F)	45H (E)
Ⓢ+3	48H (H)	47H (G)
Ⓢ+4	4AH (J)	49H (I)
Ⓢ+5	4CH (L)	4BH (K)
Ⓢ+6	4EH (N)	4DH (M)
Ⓢ+7	50H (P)	4FH (O)

CPU模块前面的LED显示器

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P

ASCII字符
正被存储的ASCII数据(十六进制数)

(2) 如果从 Ⓢ 指定的软元件开始的 8 个数据点上没有存储 ASCII 数据，则将发生以下状况：

- T、 C、 D、 W ... 空白
- R ..... 无法确定此处将显示什么 (如果文件寄存器 (R) 已清除，将是空白)

(3) LED 显示器可显示以下字符：

- 数字 0 到 9
- 字母 A 到 Z(大写字母)
- 特殊符号 <、 >、 =、 *、 /、 、 、 +、 −

(4) 在顺控程序中使用 $MOV 指令来转换包含字母和数字的 ASCII 数据。

运算错误

(1) 没有同 LED 指令相关的运行错误。

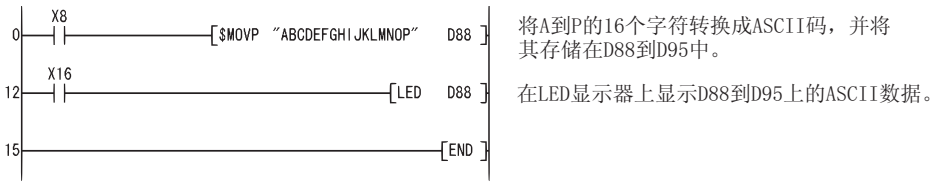
要 点

该指令仅可用于 Q3ACPU，Q4ACPU 和 Q4ARCPU。
如果在一个前面没有 LED 显示器的 CPU 模块 (QCPU、Q2ACPU、Q2AS(H)CPU) 上执行 LED 指令，则不会进行任何处理。

程序示例

(1) 当 X8 变为 ON 时，以下程序将字符串 “ABCDEFGHIJKLMNOP”转换成 ASCII 码，并将其存储在 D88 到 D95 上，然后当 X16 变为 ON 时，将 D88 到 D95 上的 ASCII 码输出到位于 CPU 模块前面的 LED 显示器上。

[梯形图模式]

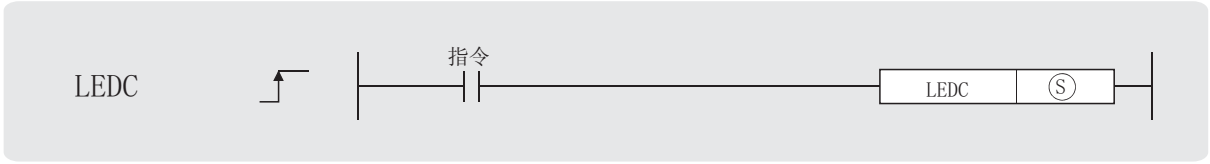


[列表模式]

步	指令	软元件
0	LD	X8
1	$MOVP	"ABCDEFGHIJKLMNOP" D88
12	LD	X16
13	LED	D88
15	END	

7.9.4 注释的 LED 显示指令 (LEDC)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ ：用于显示注释的软元件的起始号（软元件名称）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				BL、BL\S、BL\TR
Ⓢ	○							---	○

功 能

(1) 在位于 CPU 模块前面的 LED 显示器上显示存储在 Ⓢ 指定的软元件上的注释 (16 字符)。

(2) 如果 Ⓢ 指定的软元件上没有粘贴注释，或者注释超出了指定注释范围，则将发生以下状况：

Ⓢ 指定		LEDC 的动作
在注释范围之内	粘贴注释	LED 显示器上显示注释
	未粘贴注释	清除 LED 显示器
超出注释范围		无动作 (LED 显示不变)

(3) 如果注释含有 LED 显示器无法显示的字符，则不会进行准确显示。
LED 显示器可以显示以下字符：

- 数字 0 到 9
- 字母 A 到 Z(大写字母)
- 特殊符号 <、 >、 =、 ＊ 、 /、 、 、 +、 －

(4) LED 显示器上显示的注释最多可有 16 个字母，当软元件注释有 16 个或更多字母时，显示前 16 个字母。

运算错误

(1) LEDC 指令没有运行错误。

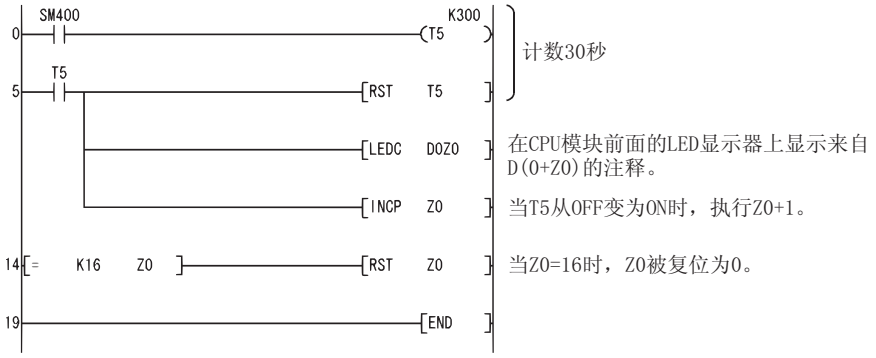
要 点

该指令仅可用于 Q3ACPU、4ACPU 和 Q4ARCPU。
如果在一个前面没有 LIED 显示器的 CPU 模块 (QCPU、Q2ACPU、Q2AS(H)CPU) 上执行，则不会进行任何处理。

程序示例

(1) 以下程序在 30 秒的间隔内显示从 D0 到 D15 的注释。

[梯形图模式]

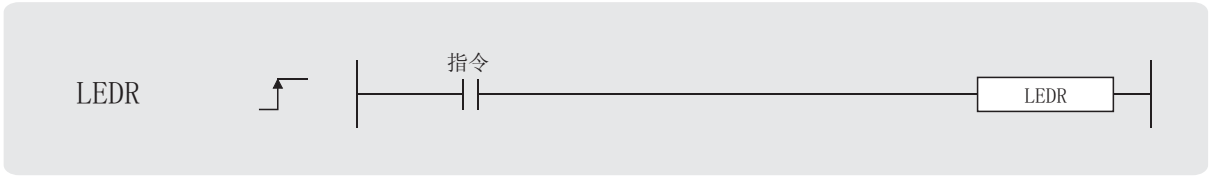


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	OUT	T5	K300
5	LD	T5	
6	RST	T5	
10	LEDC	D0Z0	
12	INCP	Z0	
14	LD=	K16	Z0
17	RST	Z0	
19	END		

7.9.5 错误显示和报警器复位指令 (LEDR)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---			---					

功 能

将自诊断错误显示复位，以便报警器显示或操作能够继续。
通过执行一次该指令，错误显示或者报警器被复位。

(1) 发生自诊断错误时的操作

(a) 如果自诊断错误是一个允许继续操作的错误
如果显示的自诊断错误是一个允许 CPU 模块继续操作的错误，则 CPU 前面的 “ERROR/ERR . ”LED 和错误显示将被复位。
必须在用户程序上复位 SM1 和 SD0，因为在此时它们不会被自动复位。
由于此时显示的错误的原因比报警器有较高的优越权，所以不采取任何复位报警器的措施。

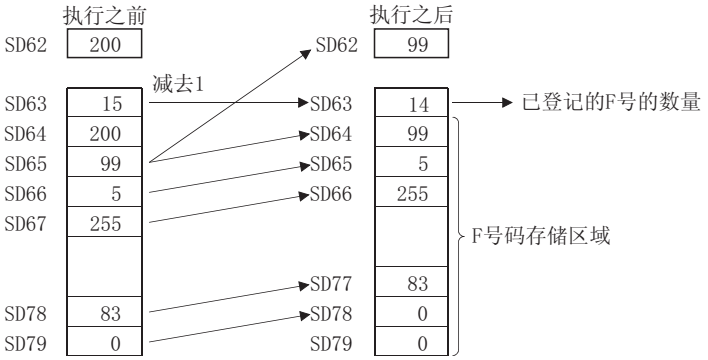
(b) 当发生电池错误时
如果在更换电池后执行 LEDR 指令，则 CPU 模块前面的 “BAT.ARM/BAT.”LED 和错误显示将被复位。
此时 SM51 变为 OFF。

(2) 当报警器 (F) 为 ON 时的操作

(a) 当 CPU 模块没有 LED 显示器时

当执行 LEDR 指令时，将进行以下操作：

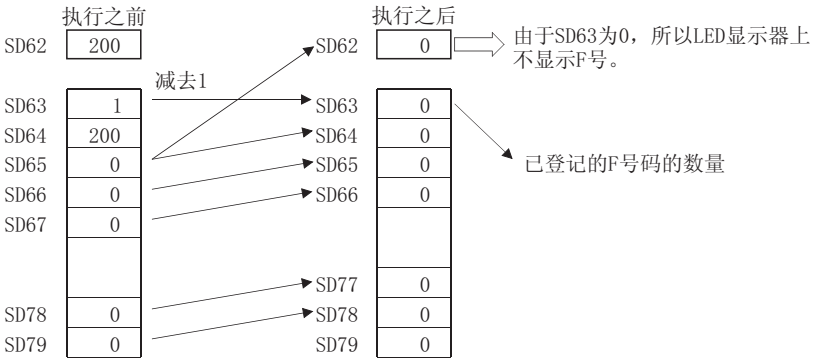
1) “USER”LED 闪烁，并被关掉。
2) 存储在 SD62 和 SD64 中的报警器 (F) 被复位，且代表 SD65 到 SD79 的 F 号被上移
3) 新存储在 SD64 上的数据被传送到 SD62。
4) SD63 上的数据减 1。但是，如果 SD63 为 0，则仍保持 0。



(b) 对于前面带有 LED 显示器的 CPU

当执行 LEDR 指令时，将进行以下操作：

1) 显示在 CPU 模块前面的 F 号将被复位。
2) “USER”闪烁，并被关掉。
3) 存储在 SD62 和 SD64 中的报警器 (F) 被复位，且代表 SD65 到 SD79 的 F 号被前移。
4) 新存储在 SD64 上的数据被传送到 SD62。
5) SD63 上的数据减 1。但是，如果 SD63 为 0，则仍保持 0。
6) 在 LED 显示器上显示存储在 SD62 上的 F 号。
   但是，如果 SD63 值为 0，则不显示。



备 注

1. 特殊寄存器 SD207 到 SD209 中设置的错误条目号和优先级的默认值如下表所示：

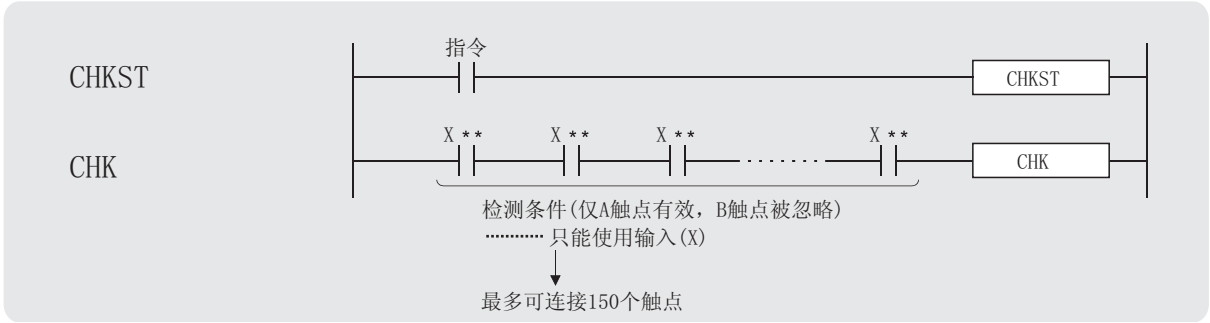
优先级	原因号（十六进制）	说明	备注
1	1	AC DOWN SINGLE PS.DOWN SINGLE PS.ERROR	断电 冗余基板电源电压下降 冗余电源模块故障
2	2	UNIT VERIFY ERR. FUSE BREAK OFF SP. UNIT ERROR	I/O 模块校验错误 保险丝熔断 特殊功能模块校验错误
3	3	OPERATION ERROR LINK PARA.ERROR SFCP OPE. ERROR SFCP EXE. ERROR	运算错误 链接参数错误 SFC 指令运算错误 SFC 程序执行错误
4	4	ICM.OPE ERROR FILE OPE. ERROR EXTEND INST. ERROR OPE. MODE DIFF. CAN'T EXE.MODE TRK.TRANS.ERR. TRK.SIZE ERROR TRK.DISCONNECT	存储卡运行错误 文件访问错误 扩展指令错误 运行状态、开关不匹配 当前模式下功能无法执行 热备数据传送错误 热备容量超出错误 热备电缆未连接或故障。
5	5	PRG.TIME OVER	恒定扫描设置时间超时错误 低速执行监视时间超时错误
6	6	CHK 指令	
7	7	报警器	
8	8	LED 指令	
9	9	BATTERY ERR.	
10	A	时钟数据	
11	B	CAN'T SWITCH STANDBY SYS.DOWN MEM.COPY EXE.	系统切换错误 待机系统不启动 / 停止错误 内存复制功能已执行

2. 如果报警器被给予了最高优先权，可通过 LEDR 指令优先将其复位。

7.10 调试和故障诊断指令

7.10.1 特殊格式故障检测（CHKST、CHK）

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



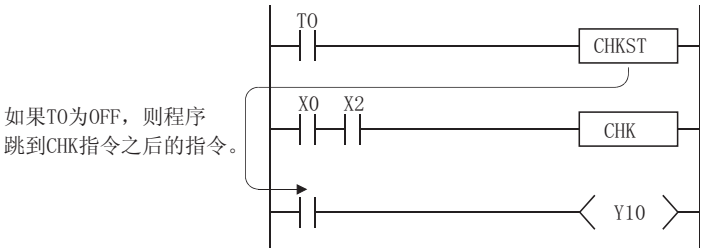
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

CHKST

(1) CHKST 指令是启动 CHK 指令的指令。

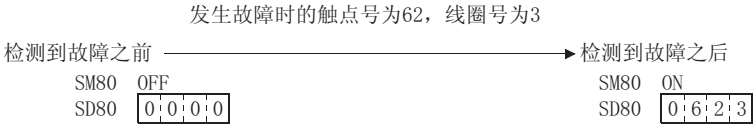
如果 CHKST 指令的命令为 OFF，则执行跳到 CHK 指令之后的步骤。如果 CHKST 指令的命令为 ON，则执行 CHK 指令。



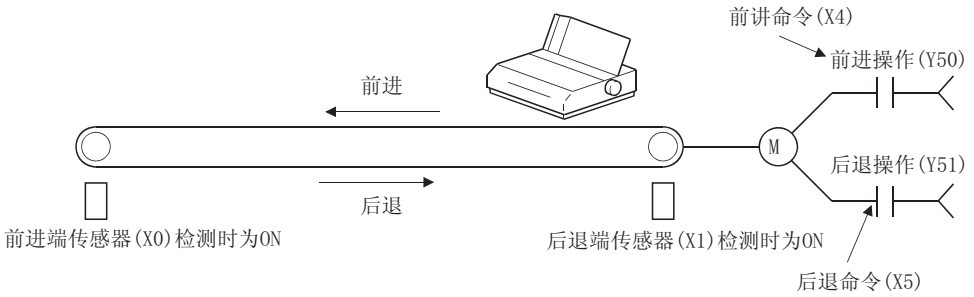
CHK

(1) CHK 指令是用于下页所示的双向操作的指令，以确认系统故障的性质。

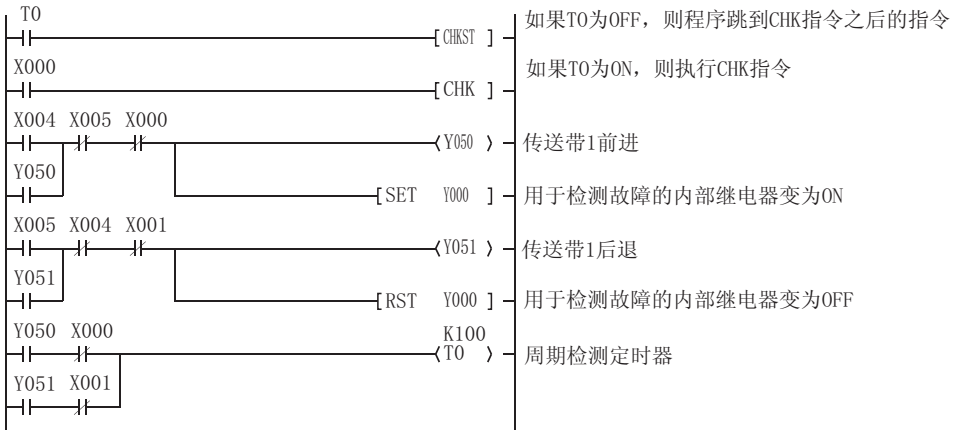
(a) 当执行 CHK 指令时，通过指定的检测条件进行故障诊断检测，如果检测出故障，则 SM80 变为 ON，且以 BCD 值将故障号存储到 SD80 上。
发现故障的触点号存储在 SD80 的高 3 位数字中（参见 (3)），发生故障的线圈号（参见 (2)）存储在 SD80 的低位数字上。



(b) CHK 指令之前的触点指令不控制 CHK 指令的执行，但却设置检测条件。



(c) 可以建立如下所示的梯形图，对上面的系统进行一个周期的检测：



(d) 当创建使用 CHK 指令的梯形图时，应考虑以下几点：

1) 前进端检测传感器和后退端检测传感器的触点号 (X□) 必须始终保持连续，此外，前进退检测传感器的触点号 (X□) 应该低于后退端检测传感器的触点号。

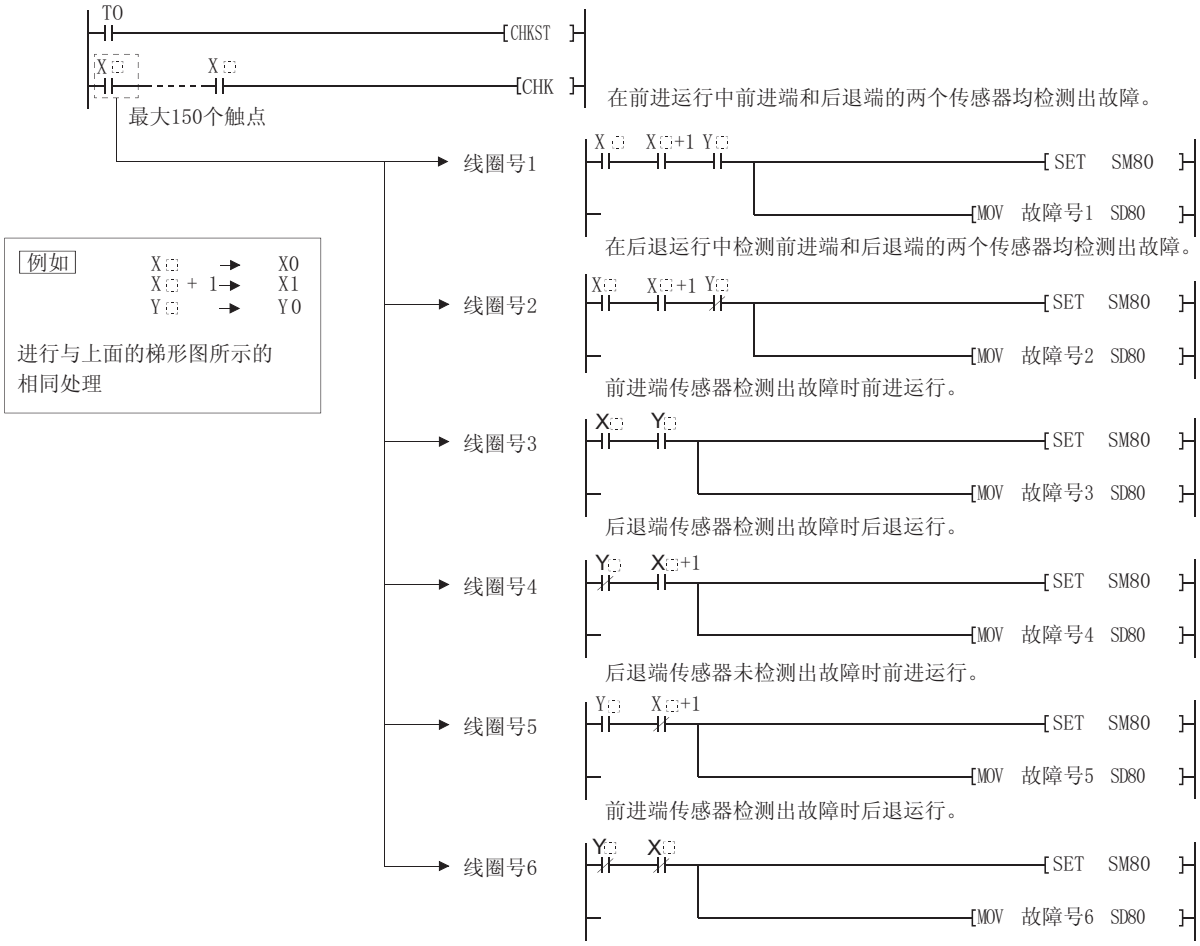
2) 对前进端检测传感器触点号 (X□) 的控制和用相同号码 (Y□) *1 的输出如下：

当正在进行前进操作时 ..... 变为 ON

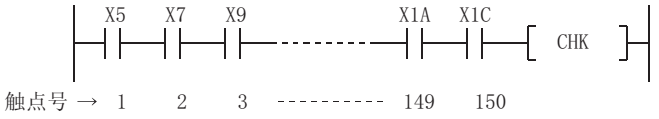
当正在进行后退操作时 ..... 变为 OFF

*1: 输出 (Y□) 被视为一个内部继电器，不能输出到一个外部软元件上。

(2) 依据指定的触点，CHK 指令执行与以下梯形图所表示的处理相同的处理。



(3) 位于左边的垂直总线上的数字 1 到 150 已经在故障检测期间被分配为触点号。



(4) 在强制执行 CHK 指令之前复位 SM80 和 SD80。
在执行 CHK 指令之后，直到 SM80 和 SD80 已经被复位，才可以再次执行 CHK 指令。
（保留 SM80 和 SD80 的内容，直到用户复位）。

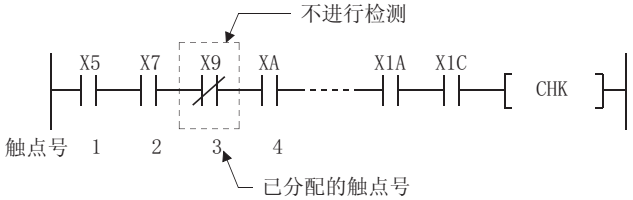
(5) 必须在 CHK 指令之前放置一个 CHKST 指令。
如果在 CHK 指令和 CHKST 指令之间存在有除 LD、LDI、AND 或 ANI 指令以外的指令，将会发生错误。（出错代码 :4235）

(6) CHK 指令可以被写入到程序的任意步中。
然而，使用 CHK 指令的数量有如下限制。

- 在所有正被执行的程序文件中最多可在两个位置使用 CHK 指令。
- 在一个程序文件中只能在一个位置使用 CHK 指令。

如果 CHK 指令的使用超出上述限制，将会发生错误。（出错代码 :4235）

(7) 在 CHK 指令之前放置 LD 和 AND 指令，以建立检测条件。检测条件无法使用其它触点指令设置。
如果已经使用 LDI 或 ANI 设置了检测条件，则将不会对它们指定的检测条件进行处理。
但是，故障检测期间的触点号也可以分配给 LDI 和 ANI 指令。



(8) 故障检测方法将依据 SM710 为 ON 状态还是 OFF 状态而有所不同。

(a) 如果 SM710 为 OFF，则逐次地对每个触点的 1 到 6 号线圈进行检测。
当执行 CHK 指令时，检测的顺序是从 1 号触点的 1 号线圈开始到 6 号线圈，然后转移到 2 号触点，从 1 号线圈开始检测。
当 n 号触点的 6 号线圈已经被检测时，CHK 指令将结束。

(b) 如果 SM710 为 ON，检测将按照线圈号顺序从 1 号触点到 n 号触点进行。
当执行 CHK 指令时，检测将从 1 号线圈的梯形图开始，按照从 1 号触点到 n 号触点的顺序，然后转移到 2 号线圈的梯形图，从 1 号触点开始。
当 6 号线圈的 n 号触点已经被检测时，CHK 指令将结束。

(9) 如果检测出一个以上的故障，第一个检测出的故障号将被存储。
此后检测的故障号将被忽略。

(10) CHK 指令不可以被低速执行类型程序使用。
如果已经在一个包含 CHK 指令的程序文件中设置了一个低速执行类型程序，将返回一个运行错误，并且将暂停 CPU 模块的操作。

运算错误

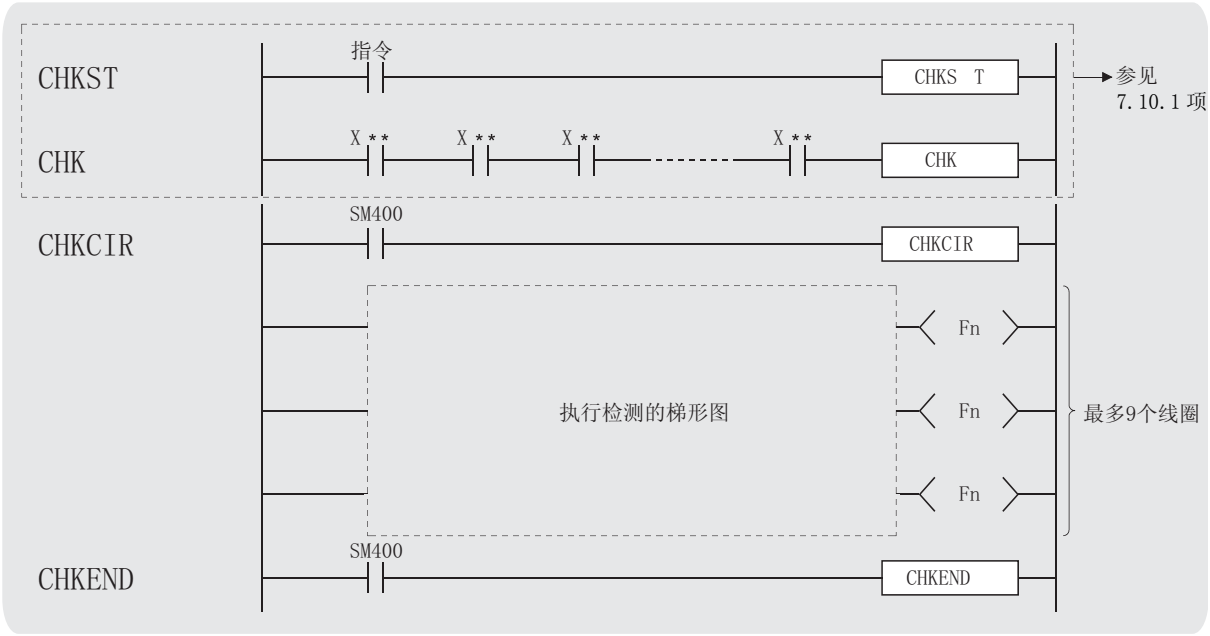
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将被存储到 SD0 中。

- 存在有并联的梯形图时。 (出错代码： 4235)
- 存在有 NOP 指令时。 (出错代码： 4235)
- 存在有 150 个以上的触点指令时。 (出错代码： 4235)
- 在 CHKST 指令之后未执行 CHK 指令时。 (出错代码： 4235)
- 未执行 CHKST 指令就执行了 CHK 指令时。 (出错代码： 4235)
- 在低速执行类型程序中使用了 CHKST 和 CHK 指令时。 (出错代码： 4235)
- 在 CHK 指令和 CHKST 指令之间存在有除 LD、LDI、AND、ANI 指令以外的指令时。 (出错代码： 4235)
- 在所有正被执行的程序中，CHK 指令用在三个或更多位置。 (出错代码： 4235)
- 在一个程序中，CHK 指令用在两个或更多位置。 (出错代码： 4235)

7.10.2 改变 CHK 指令的检测格式 (CHKCIR、CHKEND)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

1 使用 GX Developer 创建时 (高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ QnACPU/Q4ARCPU)



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

CHKCIR、CHKEND

(1) CHK 指令中使用的检测梯形图的样式可更新到任何需要的格式。实际的故障检测是通过 CHKST 和 CHK 指令执行的。

(2) 根据 CHK 指令指定的检测条件以及 CHKCIR 和 CHKEND 指令之间的梯形图执行故障检测。

备 注

有关 CHKST 和 CHK 指令的更多信息，请参见 7.10.1 节。

要 点

为了使用 CHKCIR 到 CHKEND 指令来改变 CHK 指令的检测格式，用户应该通过变址修改 (Z0) 来创建一个梯形图。

(a) 检测条件（下图中的 X2 和 X8）处指示的软元件号将被假设为用于梯形图的单个软元件号（报警器 (F) 除外）的变址修改值。

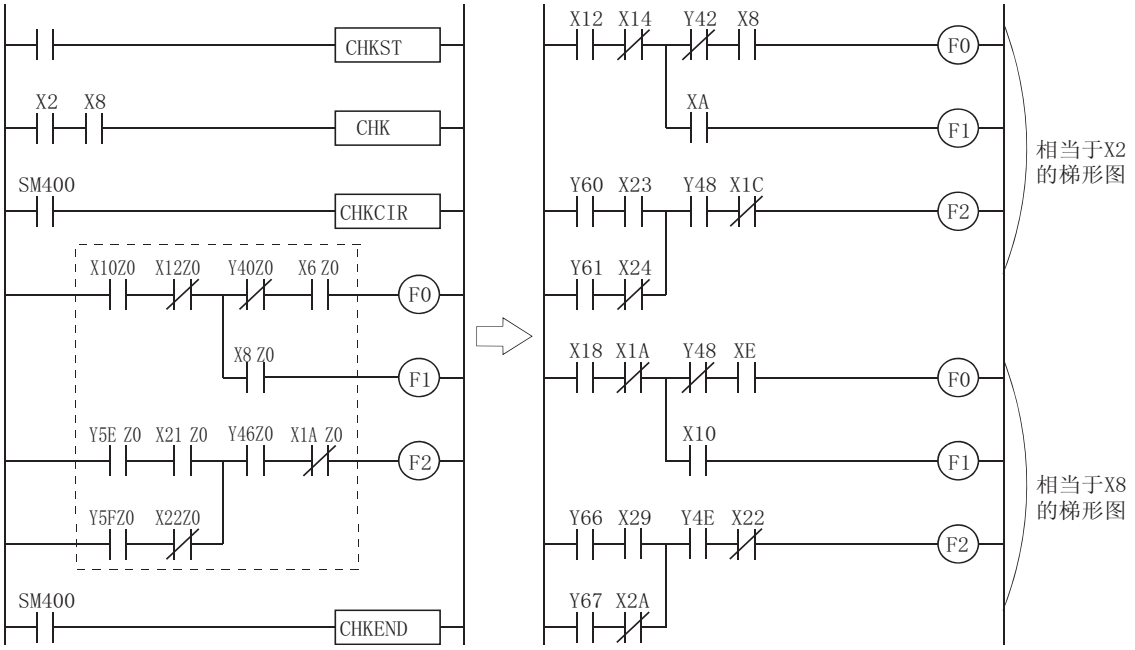
示例 下图中 ⬚ 中的 X10 如下所示：

当对应检测条件 X2 时 ..... 由 X12 执行处理
当对应检测条件 X8 时 ..... 由 X18 执行处理

但是，根据 SM710 为 ON 状态还是 OFF 状态，故障检测执行的顺序会有所不同。

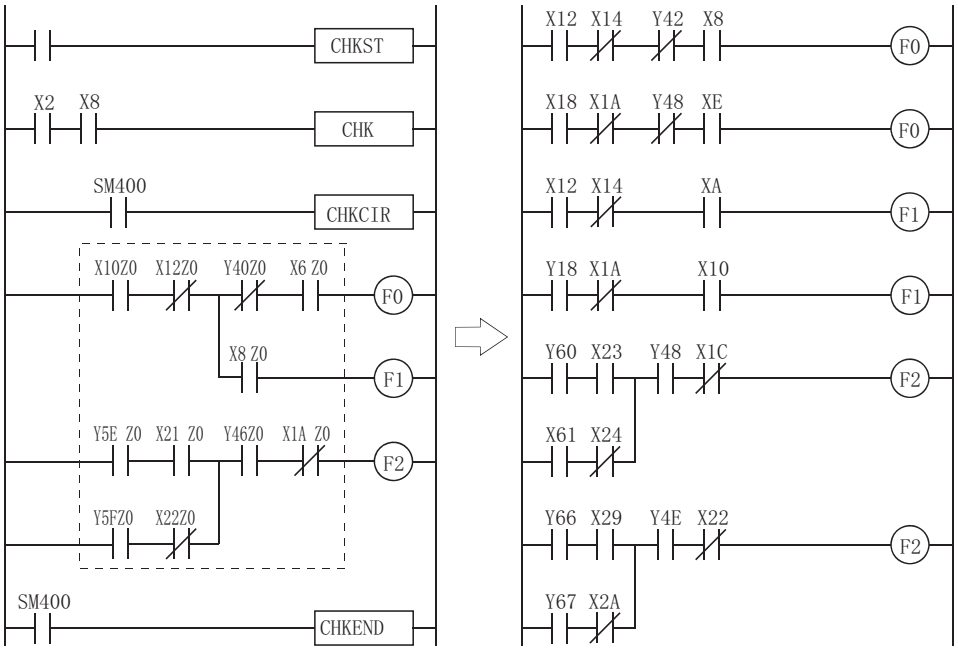
1) 如果 SM710 为 OFF，检测将依次对每个触点从 1 号线圈开始，执行到最后。

[CHKCIR 到 CHKEND 指定的梯形图]　　[CPU 模块检测顺序]



2) 如果 SM710 为 ON，检测将按照线圈号顺序，从 1 号触点执行到最后。

[CHKCIR 到 CHKEND 指定的梯形图]　　[CPU 模块检测顺序]



(b) 故障检测通过使用各种检测条件下的梯形图来检测 OUT F□ 的 ON/OFF 状态。
在所有检测条件中，SM80 将变为 ON，即使 OUT F□ 件，只有一个在梯形图模式中变为 ON 状态。
此外，与被发现处于 ON 状态的 OUT F□ 相对应的错误号（触点号和线圈号），将被从 SD80 开始按照 BCD 顺序存储起来。

(c) 梯形图中可用的指令如下所示：
触点..... LD、 LDI、 AND、 ANI、 OR、 ORI、 ANB、 ORB、 MPS、 MPP、MRD、和比较运算指令线圈
线圈..... OUT F□

(d) 以下软元件可用于梯形图触点：
输入 (X)，输出 (Y)

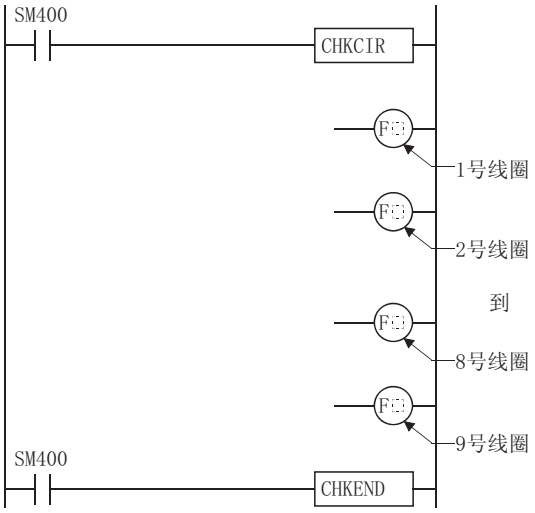
(e) 仅报警器 (F) 可用于梯形图线圈中。
但是，由于报警器 (F) 被用作虚拟触点，所以它们可设置在 F0 和任意值之间。
此外，它们可毫无困难地被重叠使用。

(f) 如果 CHR 指令执行期间使用的报警器 (F) 和某个其它非 CHK 指令环境中使用的报警器 (F) 有相同的号码，仍可执行 ON/OFF 控制，而不会产生错误。
在 CHK 指令执行期间，对它们的处理是不同的，而不是将它们看作处于不同语境中。

(g) 由于 CHK 指令中使用的报警器 (F) 实际上并没有变为 ON/OFF，如果被一个外围设备监视，它们将不会变为 ON/OFF。

(h) 最多可用 256 步创建一个梯形图。
此外，OUT F□ 最多可用 9 个线圈。

(3) 用于被 CHKCIR 到 CHKEND 指令指定的梯形图的线圈号，从 1 到 9，从头到尾，依次分配。



(4) CHKCIR 和 CHKEND 指令可在程序中任何需要的步被写入。
在所有正被执行的程序文件中，最多可在两个位置使用 CHKCIR 和 CHKEND 指令。
但是，CHKCIR 和 CHKEND 指令在单个程序文件中不能在 1 个以上的位置使用。

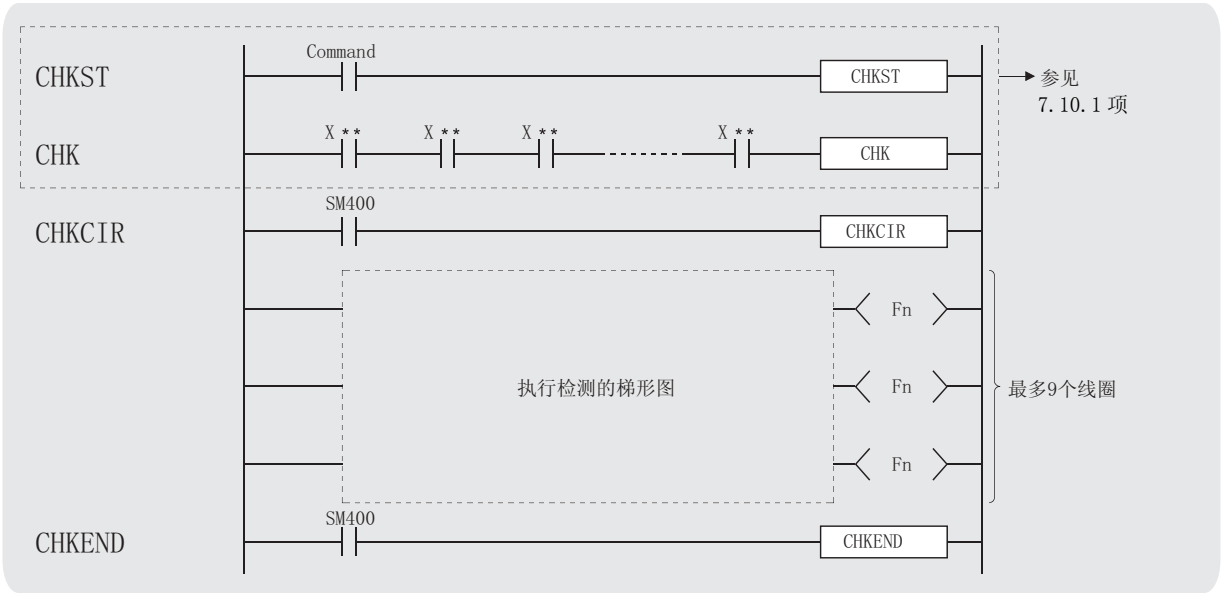
(5) CHKCIR 和 CHKEND 指令不能在低速执行类型程序中使用。
如果一个写有 CHKCIR 或 CHKEND 指令的程序文件被设置成一个低速执行类型程序，则将会发生运行错误，高性能型 QCPU/ 过程控制 CPU/QnACPU 的操作将被暂停。

运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- CHKCIR 或 CHKEND 指令在所有程序文件中出现三次或更多次。 (出错代码： 4235)
- CHKCIR 或 CHKEND 指令在单个程序文件中出现两次或更多次。 (出错代码： 4235)
- CHKCIR 指令执行之后未执行 CHKEND 指令。 (出错代码： 4230)
- 在未执行 CHKCIR 指令的情况下执行 CHKEND 指令。 (出错代码： 4230)
- CHKST 和 CHK 指令用于低速类型程序。 (出错代码： 4235)
- 在一个梯形图中有 10 个或更多个 F 实例。 (出错代码： 4235)
- 梯形图有 257 个或更多的步。 (出错代码： 4235)
- 在梯形图中碰到无法使用的软元件。 (出错代码： 4235)
- 已经在一个梯形图软元件上执行了变址修改。 (出错代码： 4235)

2 使用 SW□-GPPQ 创建时 (QnACPU/Q4ARCPU)



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

CHKCIR、 CHKEND

(1) 在 CHK 指令中将使用的检测梯形图可以更新到任何需要的格式。
实际故障检测通过 CHKST 和 CHK 指令执行。

(2) 根据 CHK 指令指定的检测条件以及 CHKCIR 和 CHKEND 指令之间的梯形图执行故障检测。

备 注

有关 CHKST 和 CHK 指令的更多信息，请参见 7.10.1 节。

要 点

当使用 CHKCIR 到 CHKEND 指令并更改 CHK 指令的检测格式时，需要在 “通用模式” 中启动 SW □ -GPPQ，以创建程序，并执行 “程序展开”。
如果在以下的 CPU 型号中启动 SW □ -GPPQ，通过 CHKCIR 到 CHKEND 指令创建一个检测格式的更改梯形图时，将无法正常进行处理。

- Q2A、 Q2A-S1、 Q2AS(H)、 Q2AS-S1
- Q2AS(H)-S1、 Q3A、 Q4A、 Q4AR

(a) 检测条件（下图中的 X2 和 X8）上指示的软元件号将被假定为梯形图中的单个软元件号（报警器 (F) 除外）的变址修改值。

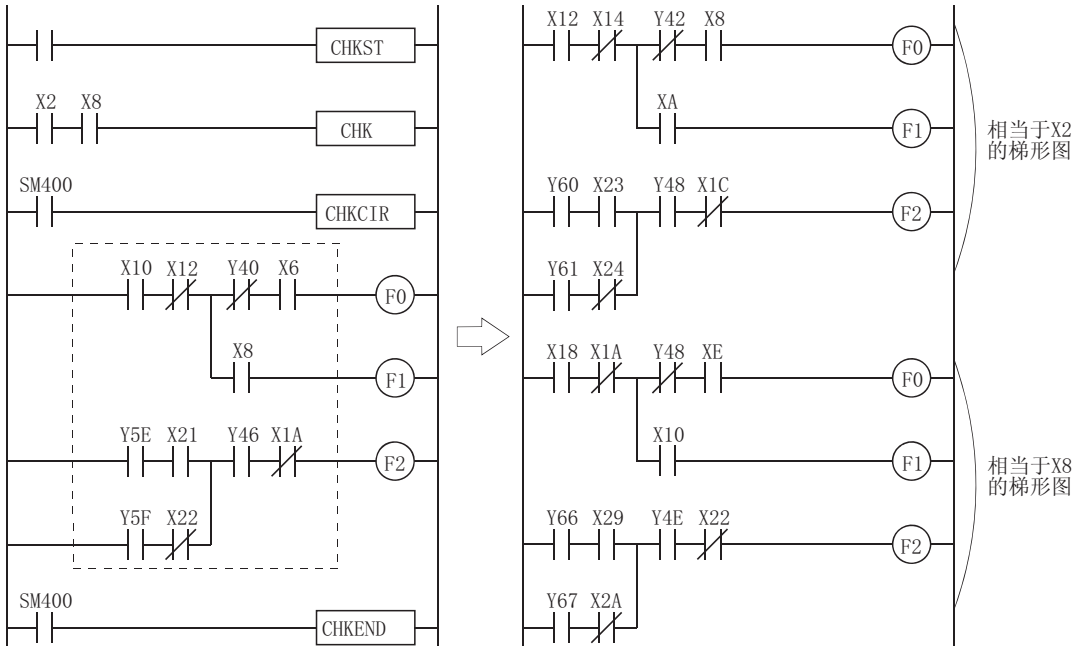
示例 下图中 ⬚ 中的 X10 如下所示：

当对应检测条件 X2 时 ..... 由 X12 执行处理
当对应检测条件 X8 时 ..... 由 X18 执行处理

但是，根据 SM710 为 ON 状态还是 OFF 状态，故障检测执行的顺序会有所不同。

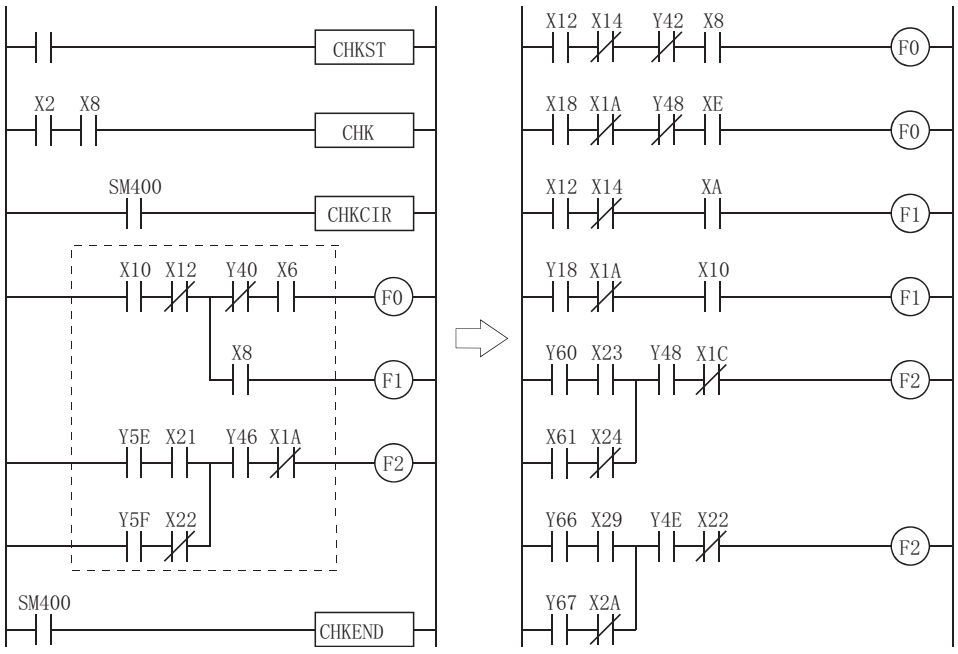
1) 如果 SM710 为 OFF，检测将依次对每个触点从 1 号线圈开始，执行到最后。

[CHKCIR 到 CHKEND 指定的梯形图]　　[QnACPU 检测顺序]



2) 如果 SM710 为 ON，检测将按照线圈号顺序从 1 号触点开始，执行到最后。

[CHKCIR 到 CHKEND 指定的梯形图]　　[QnACPU 检测顺序]



(b) 故障检测通过使用各种检测条件下的梯形图来检测 OUT F□ 的 ON/OFF 状态。
在所有检测条件中，SM80 将变为 ON，即使 OUT F□ 中只有一个在梯形图中处于 ON 状态。
此外，与被发现处于 ON 状态的 OUT F□ 相对应的错误号（触点号和线圈号），将被从 SD80 开始按照 BCD 顺序存储起来。

(c) 梯形图中可用的指令如下：
触点..... LD、 LDI、 AND、 ANI、 OR、 ORI、 ANB、 ORB、 MPS、 MPP、MRD、和比较运算指令线圈
线圈..... OUT F□

(d) 以下软元件可用于梯形图触点：
输入 (X)，输出 (Y)

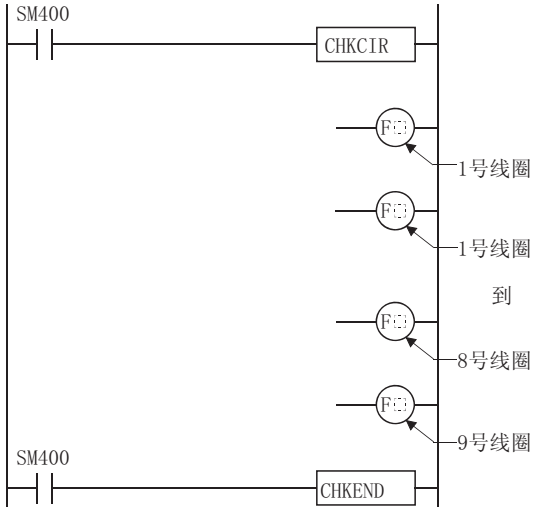
(e) 仅报警器 (F) 可用于梯形图线圈中。
但是，由于报警器 (F) 被用作虚拟触点，所以它们可设置在 F0 和任意值之间。
此外，它们可毫无困难地重叠。

(f) 如果在 CHK 指令执行期间使用的报警器 (F) 和某个其它非 CHK 指令环境中使用的报警器 (F) 有相同的号码，则仍可执行 ON/OFF 控制，而不会产生错误。
在 CHK 指令执行期间，对它们的处理是不同的，而不是将它们看作处于不同语境中。

(g) 由于 CHK 指令中使用的报警器实际上并没有变为 ON/OFF，如果受 GPPQ 监视，它们将不会变为 ON/OFF。

(h) 最多可用 256 步创建一个梯形图。
此外，OUT F□ 最多可用 9 个线圈。

(3) 用于被 CHKCIR 到 CHKEND 指令指定的梯形图的线圈号，被从 1 到 9，从头到尾，依次分配。



(4) CHKCIR 和 CHKEND 指令可在程序中任何需要的步被写入。
在所有正被执行的程序文件中，最多可在两个位置使用 CHKCIR 和 CHKEND 指令。但是，CHKCIR 和 CHKEND 指令在单个程序文件中不能在 1 个以上的位置使用。

(5) CHKCIR 和 CHKEND 指令不能在低速执行类型程序中使用。
如果一个写有 CHKCIR 或 CHKEND 指令的程序文件被设置成一个低速执行类型程序，则将发生运行错误，QnACPU 操作将被暂停。

运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- CHKCIR 或 CHKEND 指令在所有程序文件中出现三次或更多次。 (出错代码： 4235)
- CHKCIR 或 CHKEND 指令在单个程序文件中出现两次或更多次。 (出错代码： 4235)
- CHKCIR 指令执行之后未执行 CHKEND 指令。 (出错代码： 4230)
- 在未执行 CHKCIR 指令的情况下执行 CHKEND 指令。 (出错代码： 4230)
- CHKST 和 CHK 指令用于低速类型程序。 (出错代码： 4235)
- 在一个梯形图中有 10 个或更多个 F 实例。 (出错代码： 4235)
- 梯形图有 257 个或更多的步。 (出错代码： 4235)
- 在梯形图中碰到不能使用的软元件。 (出错代码： 4235)
- 已经在一个梯形图软元件上执行了变址修改。 (出错代码： 4235)

要 点

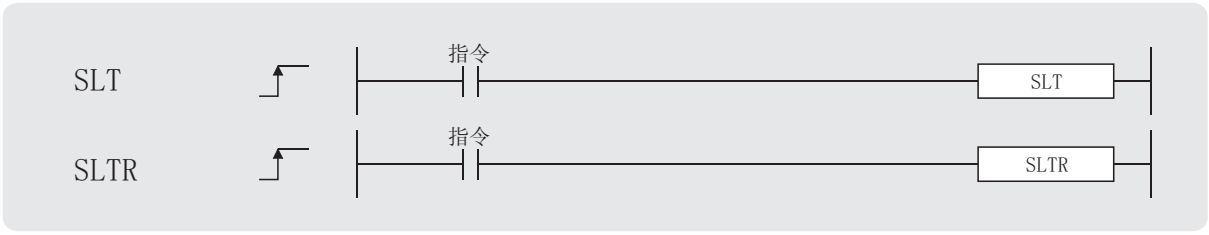
如果在一个 SW □ -GPPQ 上，在程序扩展期间发生以下错误，可能使得扩展无法执行：

•在梯形图中碰到不能使用的软元件。
•已经在一个梯形图软元件上执行了变址修改。

如果发生任何以上错误，则要改正梯形图。

7.10.3 设置和复位状态锁存 (SLT、SLTR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

SLT

(1) 当执行 SLT 指令时，在 SLT 指令执行时刻，设置在编程工具的设置软元件上的数据被存储在存储卡的文件中，供状态锁存使用。

(2) 仅当 SM806 为 ON 时，SLT 指令才有效。
如果 SM806 已经变为 OFF，则可以执行 SLT 指令，但是不会加以处理，并且不会执行状态锁存。

(3) 当执行 SLT 指令时 SM807 变为 ON，当完成状态锁存时，SM808 变为 ON。
另外，执行状态锁存的步数被存储在 SD807 上。

(4) SLT 指令已经被执行一次，将在第二次和之后试图执行时被忽略。
可以通过执行 SLTR 指令重新允许执行 SLT 指令。

(5) 可以在编程工具上监视状态锁存操作的结果。

SLTR

(1) SLTR 指令用于复位 SLT 指令。

(2) 执行 SLTR 指令允许重新执行 SLT 指令。.

备 注

1. 有关状态锁存的详细资料，请参阅使用的 CPU 模块的用户手册。
2. 有关编程工具上状态锁存的执行和监视结果，请参阅编程工具的操作手册。

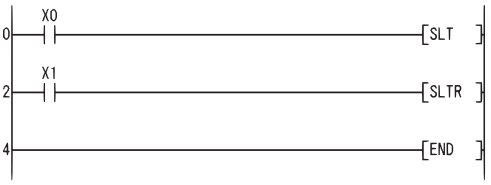
运算错误

(1) 没有和 SLT 或 SLTR 指令相关的运行错误。

程序示例

(1) 当 X0 变为 ON 时，以下程序执行 SLT 指令，当 X1 变为 ON 时，通过 SLTR 指令将其复位。

[梯形图模式]

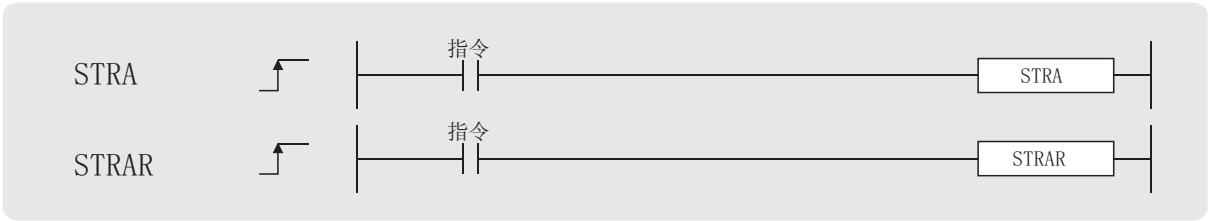


[列表模式]

步	指令	软元件
0	LD	X0
1	SLT	
2	LD	X1
3	SLTR	
4	END	

7.10.4 设置和复位采样跟踪 (STRA、STRAR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

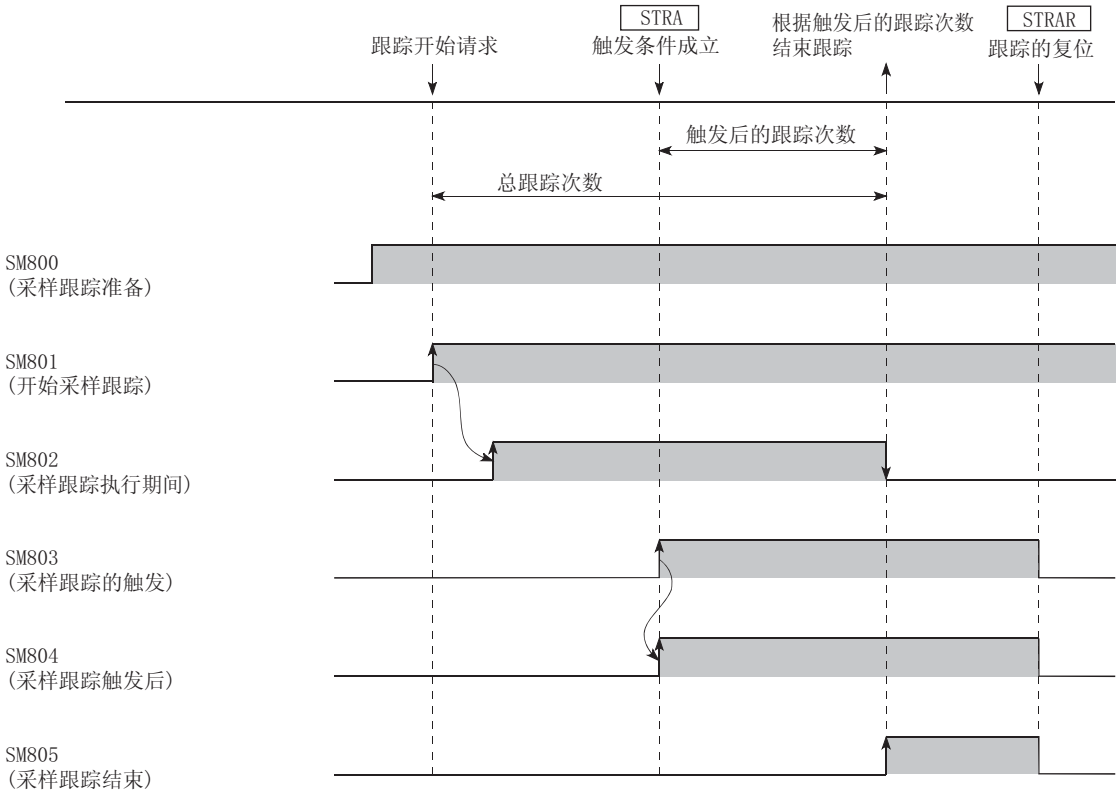


设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---			---					

功 能

采样跟踪功能用来在指定时机连续地收集在 CPU 模块指定的软元件中的数据。

通过采样跟踪功能，当 SM800、SM801 和 SM802 为 ON 时，将指定次数的跟踪结果存储到内存卡的采样跟踪文件中。



STRA

(1) STRA 指令将 SM803 变为 ON，按照跟踪条件设置的 “触发后次数”中设置的次数执行采样，锁存数据并停止采样跟踪。

(2) 在 STRA 指令执行后，完成采样跟踪，SM805 变为 ON。

(3) 在已经执行了一次 STRA 指令之后，在第二次和之后出现 STRA 指令时将被忽略。
如果执行 STRAR 指令，可以使 STRA 指令重新生效。

STRAR

(1) STRAR 指令用于复位 STRA 指令。
如果执行 STRAR 指令，可以使 STRA 指令重新生效。

(2) 如果执行 STRAR 指令，SM803 到 SM805 将变为 OFF。

备 注

1. 有关采样跟踪操作的详细资料，请参阅使用的 CPU 模块的用户手册。
2. 有关在编程工具上执行采样跟踪的情况，请参阅编程工具的操作手册。

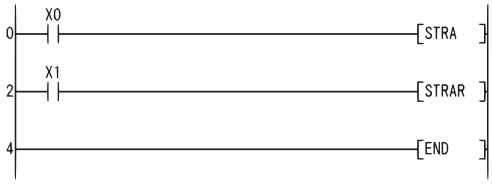
运算错误

(1) 没有和 STRA 指令相关的运行错误。

程序示例

(1) 当 X0 变为 ON 时，以下程序示例运行 STRA 指令，并且当 X1 变为 ON 时，使用 STRAR 指令复位 STRA 指令。

[梯形图模式]

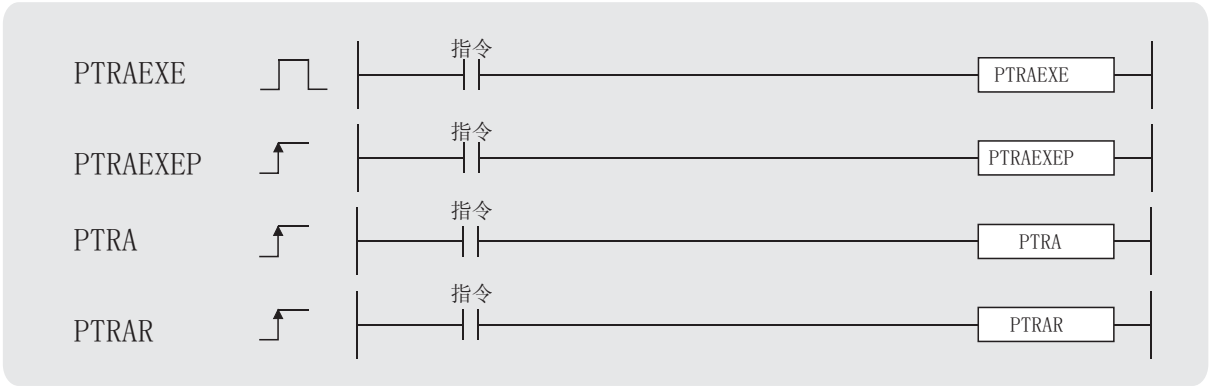


[列表模式]

步	指令	软元件
0	LD	X0
1	STRA	
2	LD	X1
3	STRAR	
4	END	

7.10.5 程序跟踪的执行、设置和复位 (PTRAEXE(P)、PTRA、PTRAR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

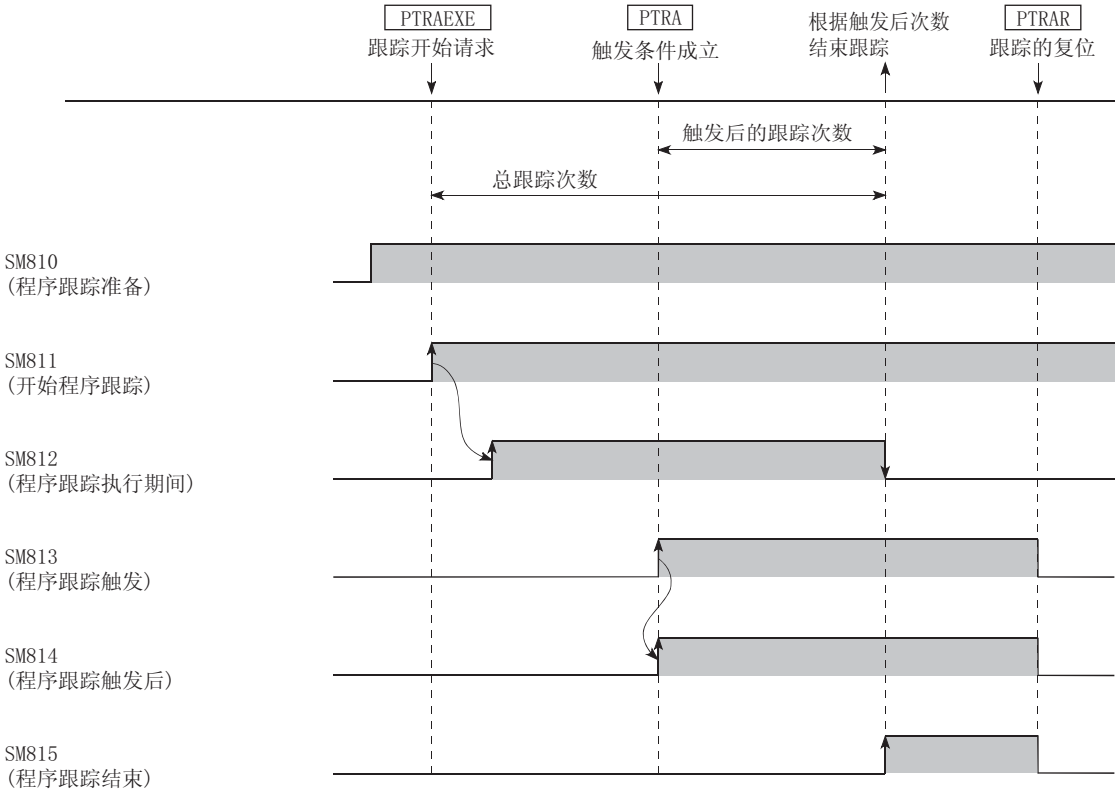


设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---			---					

功 能

程序跟踪功能用来在指定程序的指定步中收集执行状态，并将结果存储到内存卡的程序跟踪文件中。

通过程序跟踪功能，当 SM810、SM811 和 SM812 变为 ON 时，跟踪结果会被存储到内存卡的程序跟踪文件中。



PTRA

(1) PTRA 指令将 SM813 变为 ON，按照跟踪条件设置的 “触发后次数”中设置的次数执行采样后，锁存数据并停止程序跟踪。

(2) 在 PTRA 指令执行之后，完成采样跟踪，SM815 变为 ON。

(3) 在已经执行了一次 PTRA 指令之后，在第二次和之后出现 STRA 指令时将被忽略。
如果执行 PTRAR 指令，可以使 PTRA 指令重新生效。

(4) 可以使用编程工具监视程序跟踪运行结果。

PTRAR

(1) PTRAR 指令用于复位 PTRA 指令。
如果执行 PTRAR 指令，可以使 PTRA 指令重新生效。

(2) 如果执行 PTRAR 指令，SM811 到 SM815 将变为 OFF。

PTRAEXE

(1) 当程序跟踪命令变为 ON 时，执行程序跟踪。

(2) 如果 SM811 在执行程序跟踪期间变为 OFF，将暂停采样。

(3) 如果程序跟踪命令为 OFF，将不执行任何操作。

备 注

1. 有关程序跟踪操作的详细信息，请参阅使用的 CPU 模块的用户手册。
2. 有关在编程工具上执行程序跟踪的信息，请参阅编程工具的操作手册。

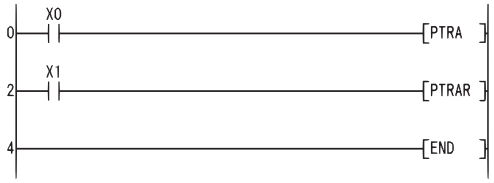
运算错误

(1) 没有和 PTRAEXE(P)、PTRA 或 PTRAR 指令相关的运行错误。

程序示例

(1) 当 X0 变为 ON 时，以下程序执行 PTRA 指令，并且当 X1 变为 ON 时，使用 PTRAR 指令复位 PTRA 指令。

[梯形图模式]



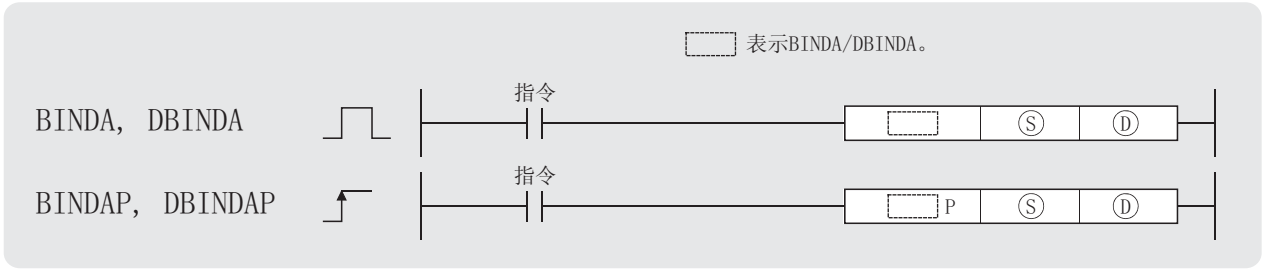
[列表模式]

步	指令	软元件
0	LD	X0
1	PTRA	
2	LD	X1
3	PTRAR	
4	END	

7.11 字符串处理指令

7.11.1 从BIN16位或32位到十进制ASCII码的转换 (BINDA(P)、DBINDA(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 要转换为 ASCII 的 BIN 数据 (BIN 16/32 位)

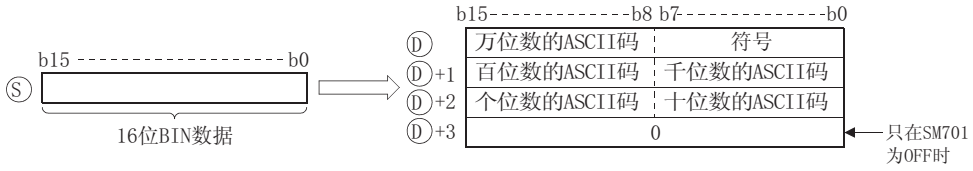
Ⓓ : 存储转换结果的软元件的起始号 (字符串)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

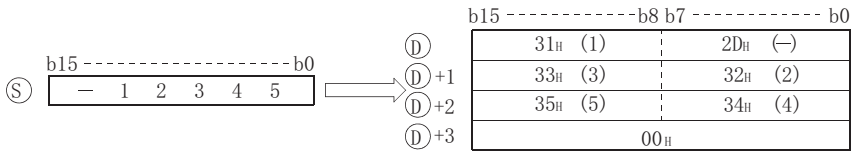
功 能

BINDA

(1) 将每个由 Ⓢ 指定的以十六进制显示的 BIN 16 位数据的单个号码转换成其各自的 ASCII 码，并从 Ⓓ 指定号码的软元件开始执行存储。



例如，如果 -12345 已经在 Ⓢ 上被指定，则以下内容将被从 Ⓓ 向前存储起来：

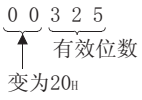


(2) 在 Ⓢ 上指定的 BIN 数据可在从 -32768 到 32767 的范围内。

(3) 存储在 Ⓓ 上的操作结果如下：

(a) 如果 BIN 数据为正，则将存储符号 “20H”，如果为负，则存储符号 “2DH”。

(b) 对有效位数前面的零，存储符号 “20H”。（执行零压缩。）

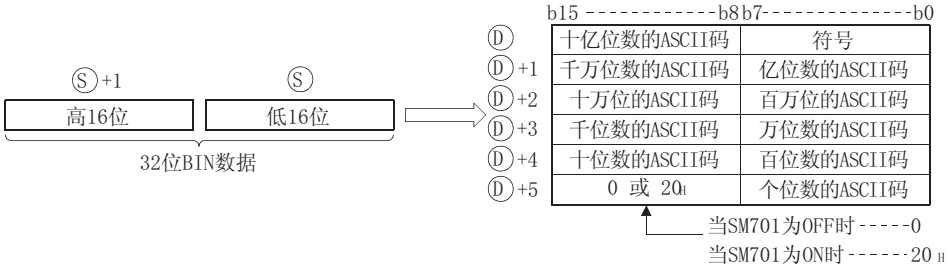


(c) 由 Ⓓ+3 指定的软元件上存储的数据依据 SM701（字符转换信号的输出号）的 ON/OFF 状态会有所不同。

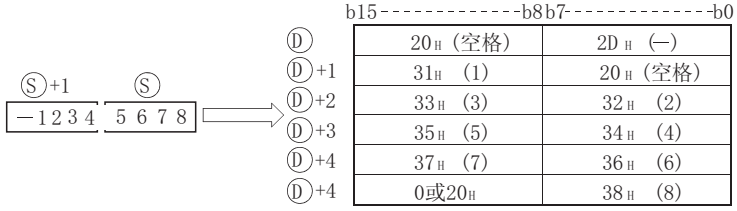
当 SM701 为 OFF 时 ..... 存储 “0”
当 SM701 为 ON 时 ..... 不变

DBINDA

(1) 当以十进制符号显示时，在每个位置为每个号码将 Ⓢ 指定的 BIN 32 位数据转换成 ASCII 码，并将其存储在 Ⓓ 指定的软元件之后。



例如，如果值 -12345678 已经被 Ⓢ 指定，以下内容将被存储在 Ⓓ 之后：

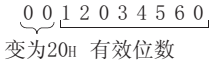


(2) 由 Ⓢ 指定的 BIN 数据可以在 -2147483648 到 2147483647 之间。

(3) 在 Ⓓ 上存储的运行结果将以下面的方法执行存储：

(a) 如果 BIN 数据为正，则存储符号 “20H”，如果为负，则存储符号 “2DH”。

(b) 对有效位数前面的零，存储符号 “20H”。（执行零压缩。）



(c) 在 Ⓓ+5 指定的软元件的高 8 位上存储的数据依据 SM701（输出选择信号的字符数）的 ON/OFF 状态，而有所不同。

当 SM701 为 OFF 时 ..... 存储 “0”
当 SM701 为 ON 时 ..... 存储 “20H”

运算错误

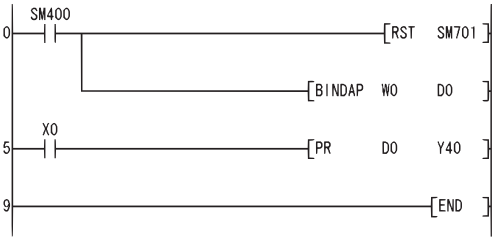
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0 中。

•ⓓ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 以下示例程序使用 PR 指令将 W0 上的 16 位 BIN 数据的十进制值转换乘 ASCII 码，并输出到 Y40 到 Y48。

[梯形图模式]

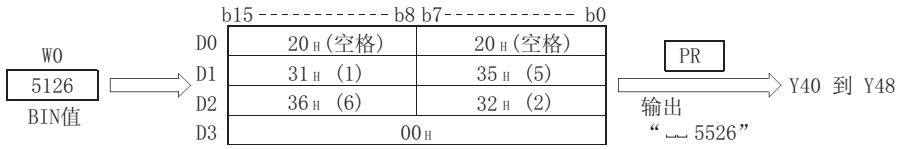


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	BINDAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

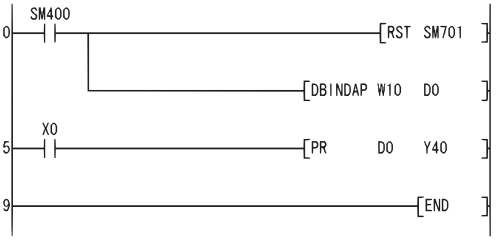
[动作]

当 X0 变为 ON 时，使用 PR 指令执行 Y40 到 Y48 的 ASCII 输出。
由于 SM701 为 OFF，所以 PR 指令将输出 ASCII 码，直到碰到 00_H。



(2) 以下程序使用 PR 指令将 W10 和 W11 上的 32 位 BIN 数据的十进制值转换成 ASCII 码，并输出到 Y40 到 Y48 上。

[梯形图模式]

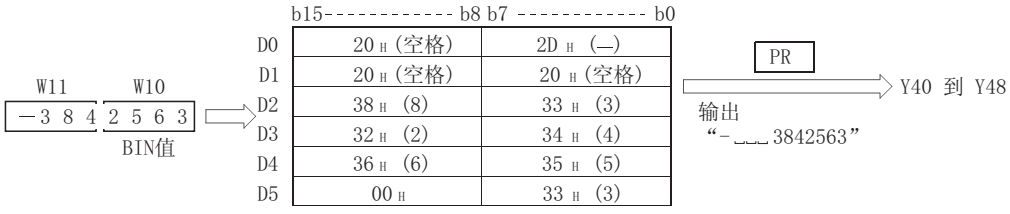


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	DBINDAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

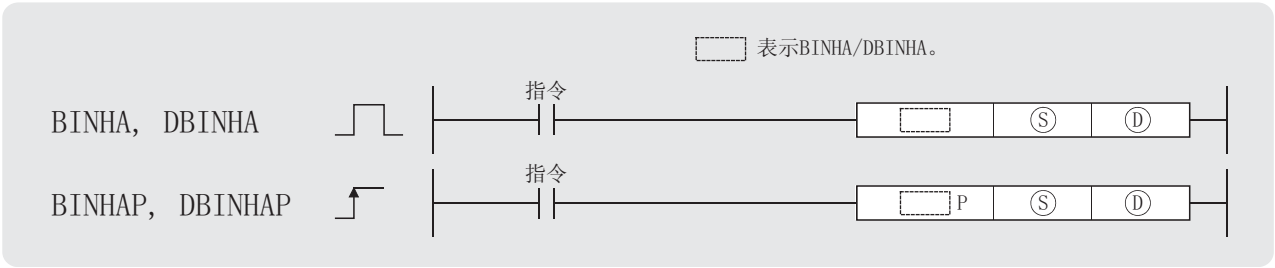
[动作]

当 X0 变为 ON 时，使用 PR 指令执行 Y40 到 Y48 的 ASCII 输出。
由于 SM701 为 OFF，所以 PR 指令将输出 ASCII 码，直到碰到 00_H。



7.11.2 从 BIN16 位或 32 位数据到十六进制 ASCII 码的转换 (BINHA(P)、DBINHA(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：要转换为 ASCII 的 BIN 数据（BIN 16/32 位）

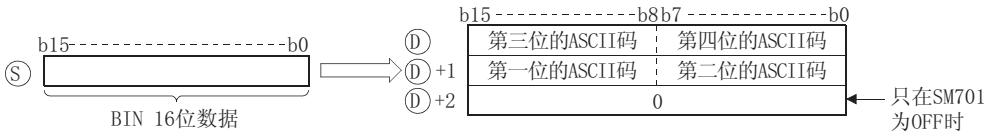
Ⓓ ：存储转换结果的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

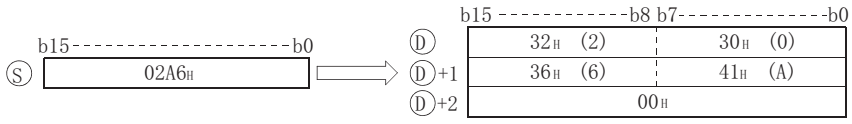
功 能

BINHA

(1) 将由 Ⓢ 指定的以十六进制显示的 BIN 16 位数据的每个号码转换成其各自的 ASCII 码，并从 Ⓓ 指定号码的软元件开始存储。



例如，如果 02A6H 已经被 Ⓢ 指定，则将按照如下所示执行存储：



(2) 由 Ⓢ 指定的 BIN 数据可在 0H 到 FFFFH 的范围内。

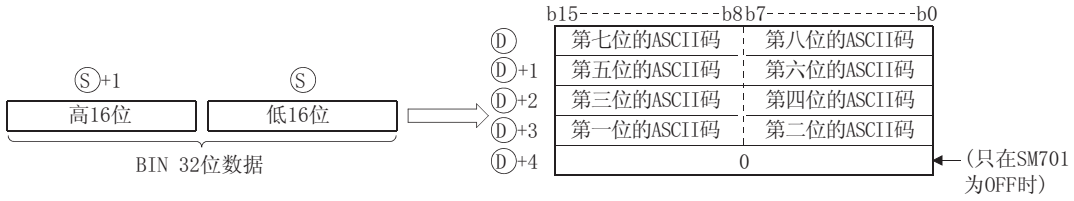
(3) 在 Ⓓ 上存储的运行结果被作为 4 位十六进制值处理。
因此，位于该值左边的作为有效位的零被作为 “0” 处理。（未执行零压缩。）

(4) 在 Ⓓ+2 指定的软元件上存储的数据依据 SM701（输出选择信号的字符数）的 ON/OFF 状态将会有所不同。

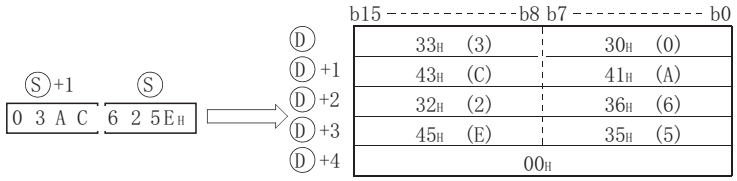
当 SM701 为 OFF 时 ..... 存储 “0”
当 SM701 为 ON 时 ..... 不变

DBINHA

(1) 当以十进制符号显示时，在每个位置为每个号将由 (S) 指定的 BIN 32 位数据转换成 ASCII 码，并将其存储在由 (D) 指定的软元件之后。



例如，如果值 03AC625EH 已经被 (S) 指定，则将其按照下面的方式存储在 (D) 之后：



(2) 由 (S) 指定的 BIN 数据可在 0H 到 FFFFFFFFH 的范围内。

(3) 在 (D) 上存储的运行结果被作为 8 位十六进制值处理。
因此，位于该值左边的作为有效位的零被作为 “0” 处理。（未执行零压缩。）

(4) 在 (D)+2 指定的软元件上存储的数据依据 SM701（输出选择信号的字符数）的 ON/OFF 状态将会有所不同。

当 SM701 为 OFF 时..... 存储 “0”
当 SM701 为 ON 时..... 不变

运算错误

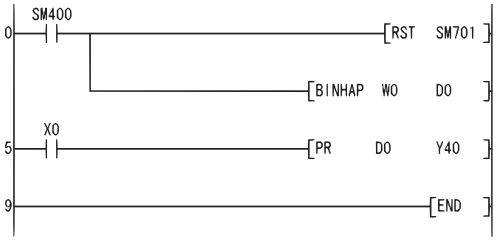
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，相关出错代码将存储在 SD0 中。

- (D) 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码：4101）

程序示例

(1) 以下程序使用 PR 指令将 W0 上的 16 位 BIN 数据的十六进制值转换成 ASCII 码形式，并输出到 Y40 到 Y48 上。

[梯形图模式]

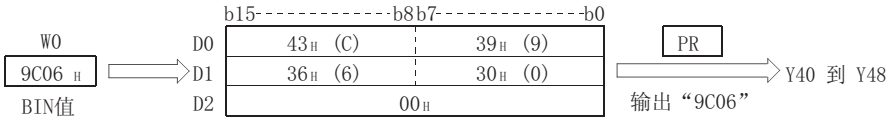


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	BINHAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

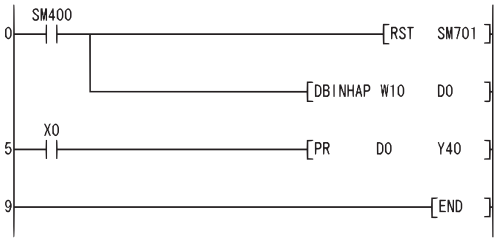
[动作]

当 X0 变为 ON 时，通过 PR 指令将 ASCII 码输出到 Y40 到 Y48 上。由于 SM701 为 OFF，PR 指令将输出 ASCII 码，直到碰到 00H。



(2) 以下程序使用 PR 指令将 W10 和 W11 上的 32 位 BIN 数据的十六进制值转换成 ASCII 码形式，并输出到 Y40 到 Y48 上。

[梯形图模式]

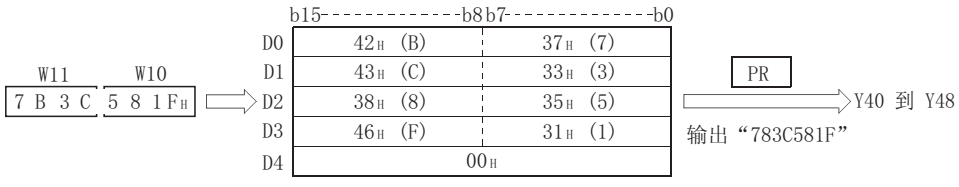


[列表模式]

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	DBINHAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

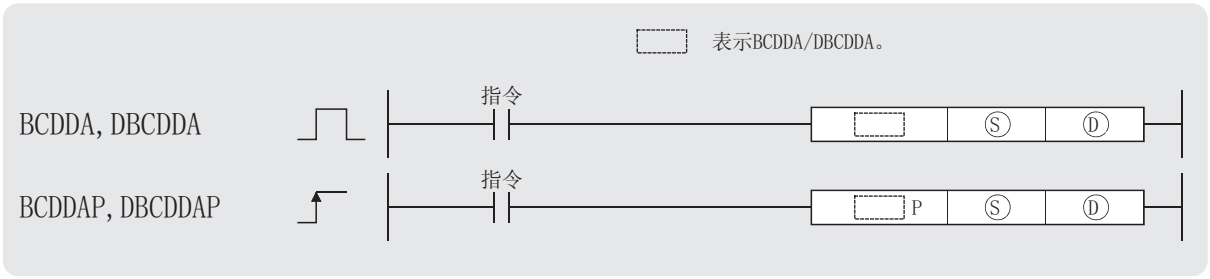
[动作]

当 X0 变为 ON 时，通过 PR 指令将 ASCII 码输出到 Y40 到 Y48 上。由于 SM701 为 OFF，PR 指令将输出 ASCII 码，直到碰到 00H。



7.11.3 从BCD4位和8位数据到十进制ASCII码的转换 (BCDDA(P)、DBCDDA(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：将被转换成ASCIII码的BCD数据（BCD 4位/8位）

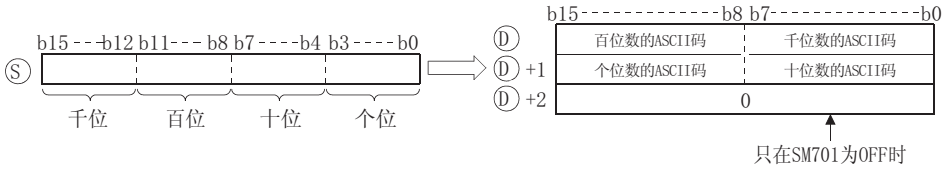
Ⓓ ：存储转换结果的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○				○			---
Ⓓ	---	○				---			---

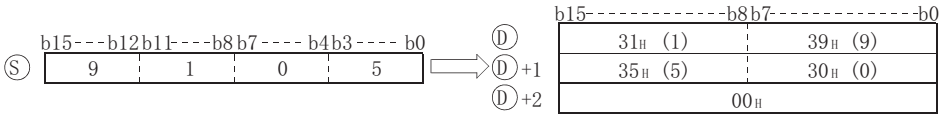
功 能

BCDDA

(1) 将 Ⓢ 指定的BCD 4位数据中的各个号码转换成它们各自的ASCII码，并从 Ⓓ 指定的软元件开始存储。

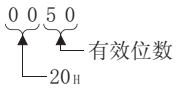


例如，如果 Ⓢ 指定了值9105，则将以下面的方式将运行结果存储在 Ⓓ 上：



(2) Ⓢ 指定的BCD数据可以在0到9999的范围内。

(3) 计算结果存储在软元件 Ⓓ 上。“有效位数”左侧的所有零都被压缩。

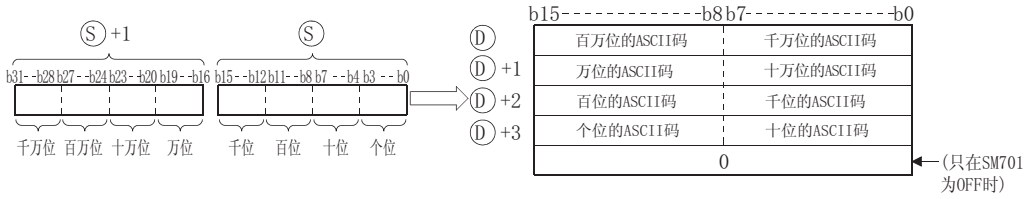


(4) 要在 ⓓ+2 指定的软元件上存储的数据依据 SM701(输出选择信号的字符数)的 ON/OFF 状态将会有所不同。

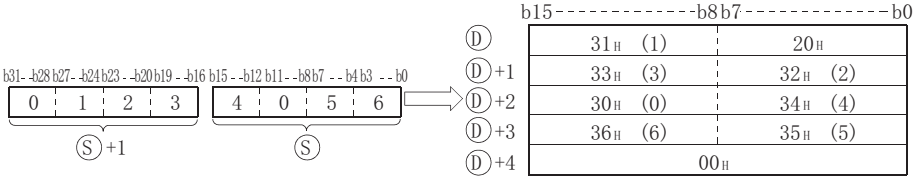
当 SM701 为 OFF 时 ..... 存储 “0”
当 SM701 为 ON 时 ..... 不变

DBCDDA

(1) 将 ⓢ 指定的 BCD 8 位数据中的各个号码转换成它们各自的 ASCII 码，并存储在 ⓓ 指定的软元件之后。

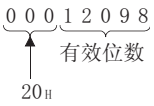


例如，如果值 01234056 已经被 ⓢ 指定，则将按照以下方式将运行结果存储在 ⓓ 之后：



(2) ⓢ 指定的 BCD 数据可以在 0 到 99999999 的范围内。

(3) 计算结果存储在软元件 ⓓ 上，“有效位数”左侧的所有零都被压缩。



(4) 要在 ⓓ+4 指定的软元件上存储的数据依据 SM701(输出选择信号的字符数)的 ON/OFF 状态将会有所不同。

当 SM701 为 OFF 时 ..... 存储 “0”
当 SM701 为 ON 时 ..... 不变

运算错误

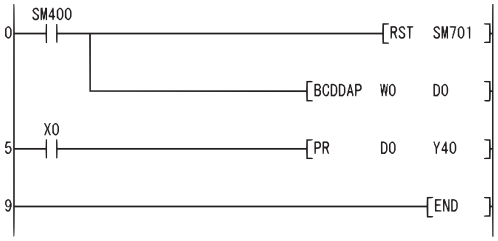
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- ⓢ 上的数据在 BCDDA 指令操作期间超出了 0 到 9999 的范围。 (出错代码： 4100)
- ⓢ 上的数据在 DBCDDA 指令操作期间超出了 0 到 99999999 的范围。 (出错代码： 4100)
- ⓓ 指定的软元件超出了相应软元件的范围。 (只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 以下程序使用 PR 指令将 BCD 4 位数据（W0 上的值）转换成十进制数，并以 ASCII 格式将其输出到 Y40 到 Y48 上。

［梯形图模式］

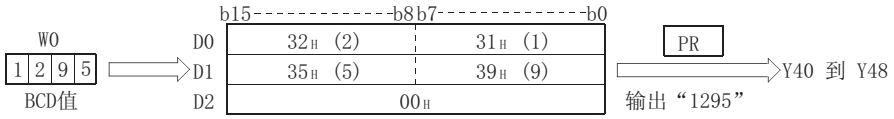


［列表模式］

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	BCDDAP	W0	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

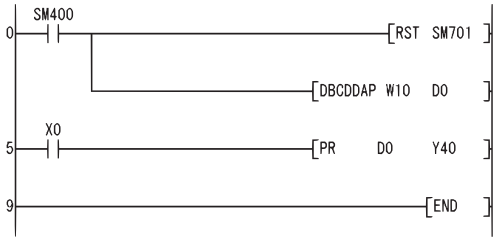
［动作］

当 X0 变为 ON 时，使用 PR 指令执行 Y40 到 Y48 的 ASCII 码输出。由于 SM701 为 OFF，PR 指令将输出 ASCII 码，直到碰到 00H。



(2) 以下程序使用 PR 指令将 BCD 8 位数据（W0 和 W11 上的值）转换成十进制数，并以 ASCII 格式将其输出到 Y40 到 Y48 上。

［梯形图模式］

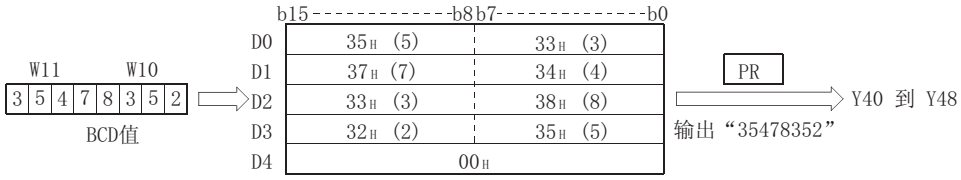


［列表模式］

步	指令	软元件	
0	LD	SM400	
1	RST	SM701	
2	DBCDDAP	W10	D0
5	LD	X0	
6	PR	D0	Y40
9	END		

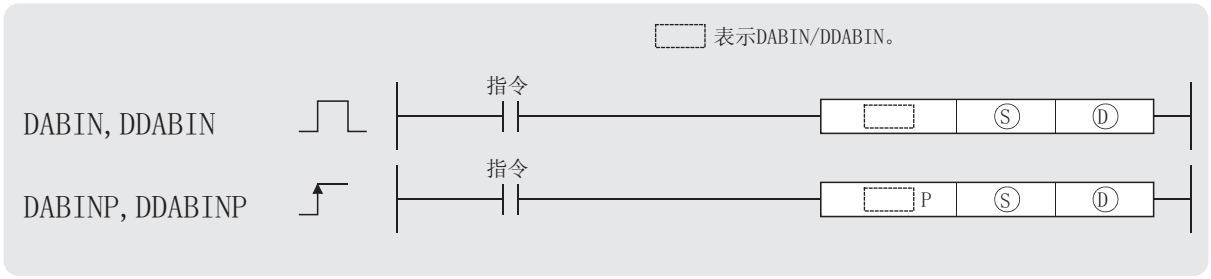
［动作］

当 X0 变为 ON 时，使用 PR 指令执行 Y40 到 Y48 的 ASCII 码输出。由于 SM701 为 OFF，PR 指令将输出 ASCII 码，直到碰到 00H。



7.11.4 从十进制 ASCII 码到 BIN16 位和 32 位数据的转换 (DABIN(P)、DDABIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储将被转换成 BIN 值的 ASCII 数据的软元件的起始号（字符串）

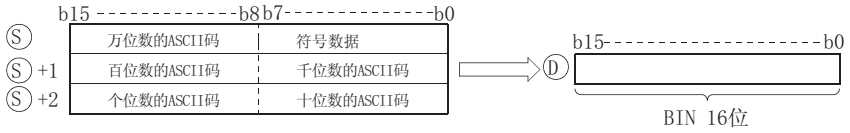
Ⓓ ：存储转换结果的软元件的起始号 (BIN 16/32 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

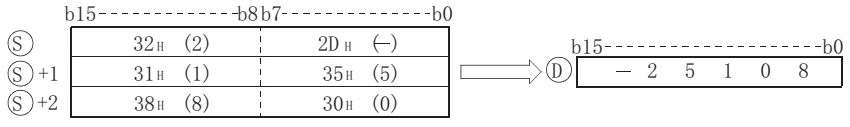
功 能

DABIN

(1) 将存储在由 Ⓢ 指定号码的软元件之后的十进制 ASCII 数据转换成 BIN 16 位数据，并将其存储在由 Ⓓ 指定号码的软元件中。



例如，如果 Ⓢ 之后的 ASCII 码被指定为值 -25108H，则将按照以下方式将运行结果存储在 Ⓓ 上：



(2) Ⓢ 到 Ⓢ+2 指定的 ASCII 数据可以在 -32768 到 32767 的范围内。

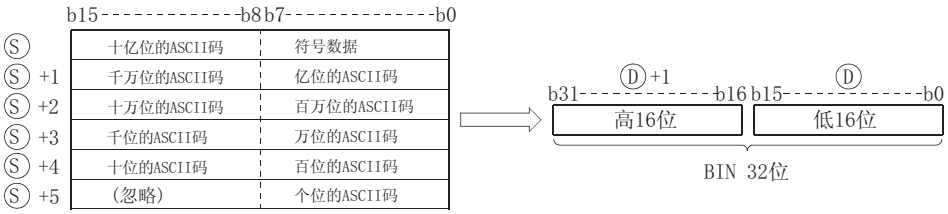
(3) 在 “符号” 数据中，如果转换的数据为正，则设置 “20H”，如果为负，则设置 “2DH”。（如果设置了除 “20H” 和 “2DH” 以外，将作为正数据处理。）

(4) 可以在 “30H” 到 “39H” 的范围内给每个位置设置 ASCII 码。

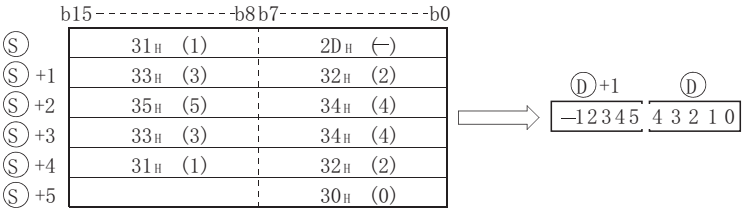
(5) 如果给单个位置设置的 ASCII 码是 “20H” 或 “00H”，则将作为 “30H” 处理。

DDABIN

(1) 将存储在 Ⓢ 指定号码的软元件之后的十进制 ASCII 数据转换成 BIN 32 位数据，并将其存储在 Ⓓ 指定号码的软元件中。



例如，如果 Ⓢ 后的 ASCII 码被指定为值 -1234543210H，则将按照以下方式将运行结果存储在 Ⓓ+1 和 Ⓓ 上：



(2) 由 Ⓢ 到 Ⓢ+5 指定的 ASCII 数据可以在 -2147483648 到 2147483647 的范围内。
此外，存储在 Ⓢ+5 的高字节上的数据将被忽略。

(3) 在 “符号” 数据中，如果转换的数据为正，则设置 “20H”，如果为负，则设置 “2DH”。
（如果设置了除 “20H” 和 “2DH” 以外，将作为正数据处理。）

(4) 可以在 “30H” 到 “39H” 的范围内给每个位置设置 ASCII 码。

(5) 如果给单个位置设置的 ASCII 码是 “20H” 或 “00H”，则将作为 “30H” 处理。

运算错误

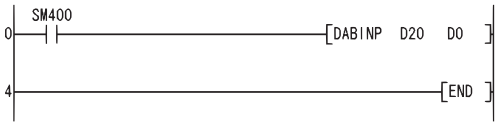
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- Ⓢ 到 Ⓢ+5 中指定的各位的 ASCII 码是 “30H” 到 “39H”，“20H” 或 “00H” 以外时。
（出错代码： 4100）
- Ⓢ 到 Ⓢ+5 中指定的 ASCII 数据超出了下述范围时： （出错代码： 4100）
使用 DABIN 指令时..... −32768 到 32767
使用 DDABIN 指令时..... −2147483648 到 2147483647
- Ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 以下程序将设置在 D20 到 D22 上的十进制，5 位 ASCII 数据和符号转换成 BIN 值，并将结果存储在 D0 上。

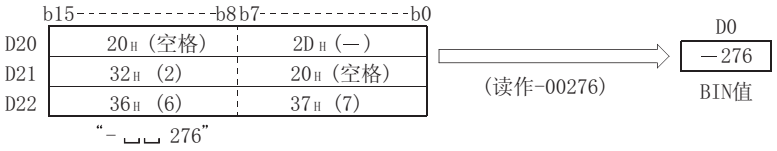
[梯形图模式]



[列表模式]

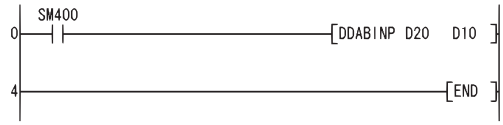
步	指令	软元件	
0	LD	SM400	
1	DABINP	D20	D0
4	END		

[动作]



(2) 以下程序将设置在 D20 到 D25 上的十进制，10 位 ASCII 数据和符号转换成 BIN 值，并将结果存储在 D10 和 D11 上。

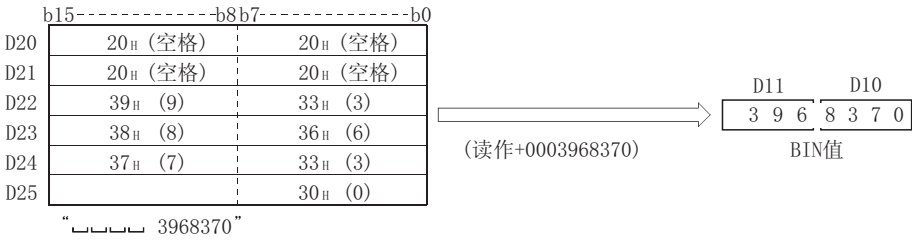
[梯形图模式]



[列表模式]

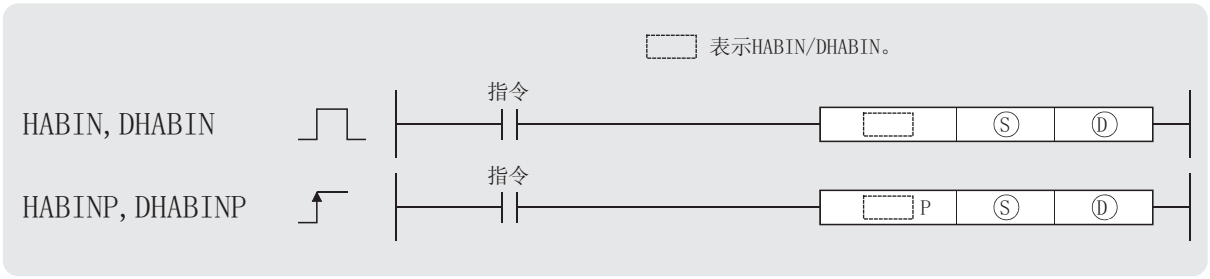
步	指令	软元件	
0	LD	SM400	
1	DDABINP	D20	D10
4	END		

[动作]



7.11.5 从十六进制 ASCII 到 BIN16 位和 32 位数据的转换 (HABIN(P)、DHABIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储将被转换成 BIN 值的 ASCII 数据的软元件的起始号（字符串）

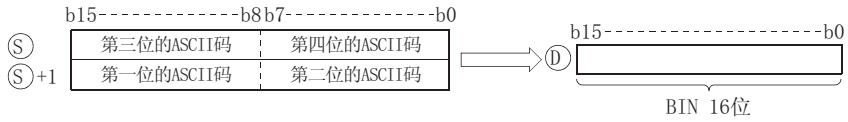
Ⓓ ：存储转换结果的软元件的起始号（BIN 16/32 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

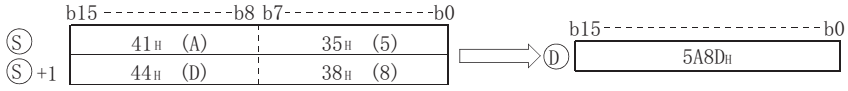
功 能

HABIN

(1) 将存储在由 Ⓢ 指定号码的软元件之后的 ASCII 数据转换成 BIN 16 位数据，并将其存储在由 Ⓓ 指定号码的软元件中。



例如，如果 Ⓢ 之后的 ASCII 码被指定为值 5A8DH，则将按照以下方式将运行结果存储在 Ⓓ 上：

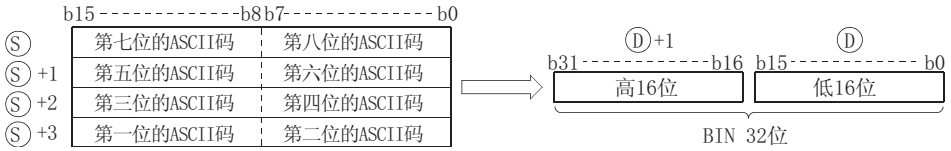


(2) Ⓢ 到 Ⓢ+1 指定的 ASCII 数据可以在 0000H 到 FFFFH 的范围内。

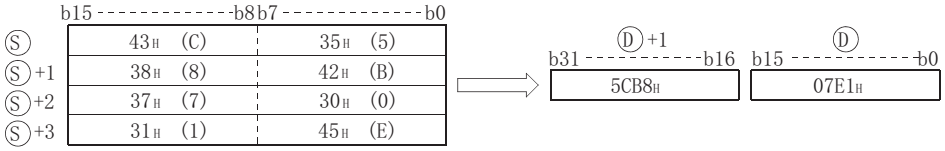
(3) ASCII 码可以在 “30H” 到 “39H” 和 “41H” 到 “46H” 的范围内。

DHABIN

(1) 将存储在由 Ⓢ 指定号码的软元件之后的十六进制 ASCII 数据转换成 BIN 32 位数据，并将其存储在由 Ⓓ 指定号码的软元件中。



例如，如果 Ⓢ 之后的 ASCII 码被指定为值 5CB807E1H，则将按照以下方式将运行结果存储在 Ⓓ+1 和 Ⓓ 上：



(2) Ⓢ 到 Ⓢ+3 指定的 ASCII 数据可以在 00000000H 到 FFFFFFFFH 范围内。

(3) ASCII 码可以在 “30H” 到 “39H” 和 “41H” 到 “46H” 的范围内。

运算错误

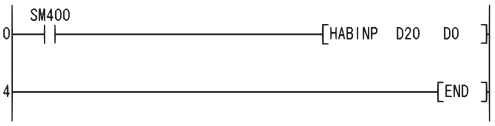
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- Ⓢ 和 Ⓢ+3 为各个号码指定的 ASCII 码超出了 “30H” 到 “39H” 和 “41H” 到 “46H” 的范围。 （出错代码： 4100）
- Ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 以下程序将设置在 D20 和 D21 上的十六进制，4 位 ASCII 数据转换成 BIN 值，并将结果存储在 D0 上。

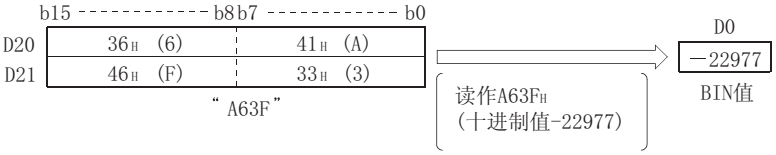
[梯形图模式]



[列表模式]

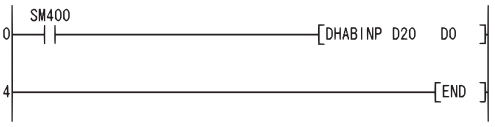
步	指令	软元件	
0	LD	SM400	
1	HABINP	D20	D0
4	END		

[动作]



(2) 以下程序将设置在 D20 到 D23 上的十六进制，8 位 ASCII 数据转换成 BIN 值，并将结果存储在 D10 和 D11 上。

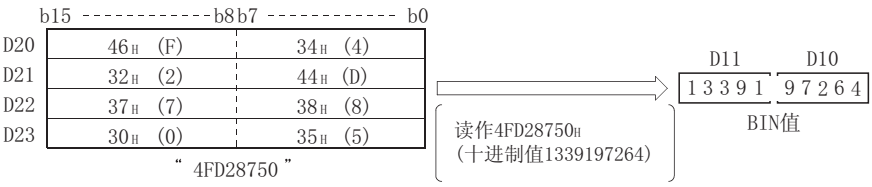
[梯形图模式]



[列表模式]

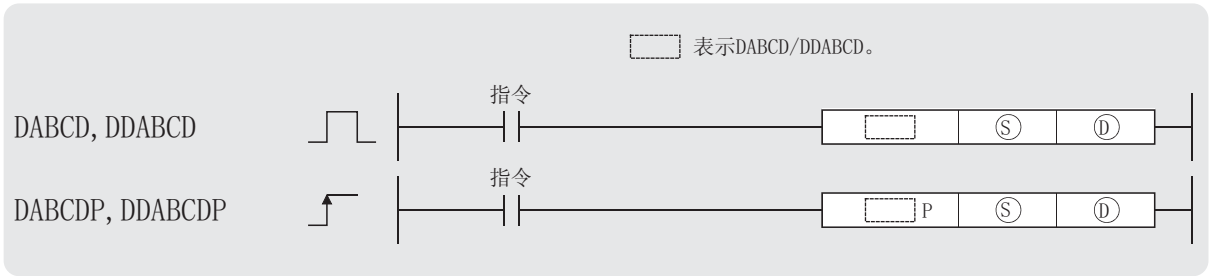
步	指令	软元件	
0	LD	SM400	
1	DHABINP	D20	D0
4	END		

[动作]



7.11.6 从十进制 ASCII 码到 BCD4 位或 8 位数据的转换 (DABCD(P)、DDABCD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储将被转换成 BCD 值的 ASCII 数据的软元件的起始号（字符串）

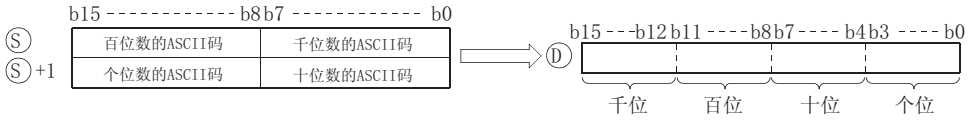
Ⓓ ：存储转换结果的软元件的起始号（BCD 4 位 /8 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

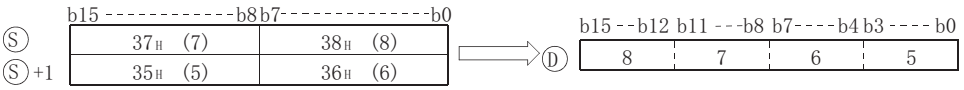
功 能

DABCD

(1) 将存储在 Ⓢ 指定号码的软元件之后的十六进制 ASCII 数据转换成 4 位 BCD 数据，并将其存储在 Ⓓ 指定号码的软元件上。



例如，如果 Ⓢ 之后的 ASCII 码被指定为值 8765H，则将按照以下方式将运行结果存储在 Ⓓ 上：



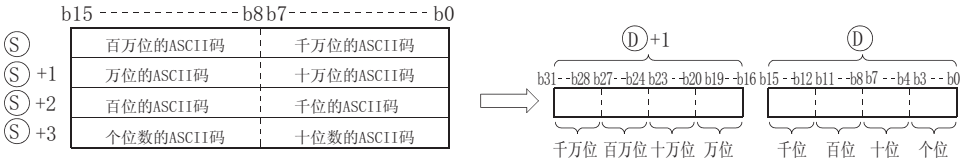
(2) Ⓢ 到 Ⓢ+1 指定的 ASCII 数据可以在 0 到 9999 的范围内。

(3) 为每个位设置的 ASCII 码可以在 “30H” 到 “39H” 的范围内。

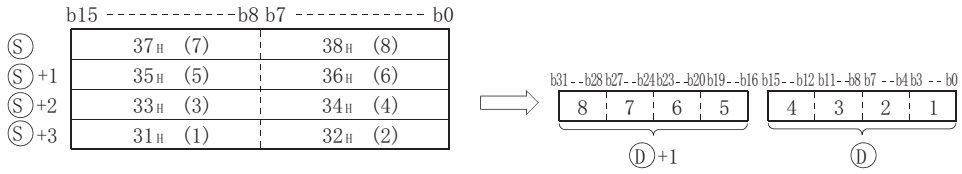
(4) 如果单个位的 ASCII 码为 “20H” 或 “00H”，则将当作 “30H” 处理。

DDABCD

(1) 将存储在由 (S) 指定号码的软元件之后的十六进制 ASCII 数据转换成 8 位 BCD 数据，并将其存储在由 (D) 指定号码的软元件上。



例如，如果 (S) 之后的 ASCII 码被指定为值 87654321H，则将按照以下方式将运行结果存储在 (D)+1 和 (D) 上：



(2) (S) 到 (S)+3 指定的 ASCII 数据可以在 0 到 99999999 的范围内。

(3) 为每个位设置的 ASCII 码可以在 “30H” 到 “39H” 的范围内。

(4) 如果单个位的 ASCII 码是 “20H” 到 “00H”，则将当作 “30H” 处理。

运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- 在 (S) 上的数据内，有字符超出了 0 到 9 的范围。 (出错代码： 4100)
- (S) 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 以下程序将 D20 到 D22 上设置的十六进制 ASCII 数据转换成 BCD 4 位数据，并将结果输出到 Y40 到 Y4F 上。

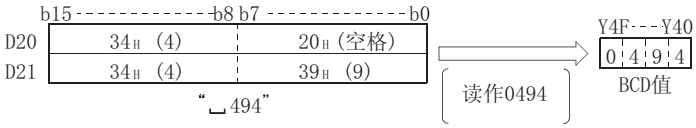
[梯形图模式]



[列表模式]

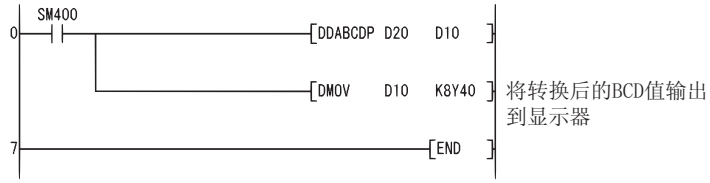
步	指令	软元件	
0	LD	SM400	
1	DABCDP	D20	K4Y40
4	END		

[动作]



(2) 以下程序将设置在 D20 到 D23 中的十进制 ASCII 数据转换成 BCD8 位数据，将结果存储在 D10 和 D11 中，并将其输出到 Y40 到 Y5F 上。

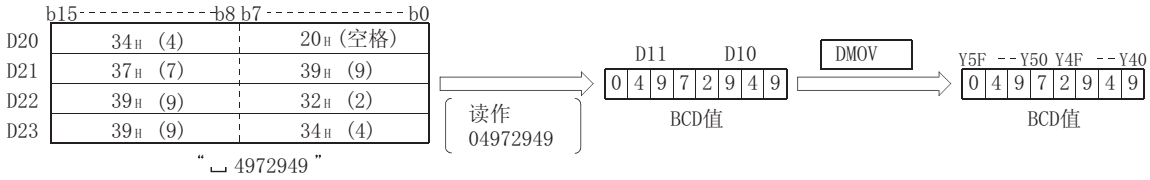
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DDABCDP	D20	D10
4	DMOV	D10	K8Y40
7	END		

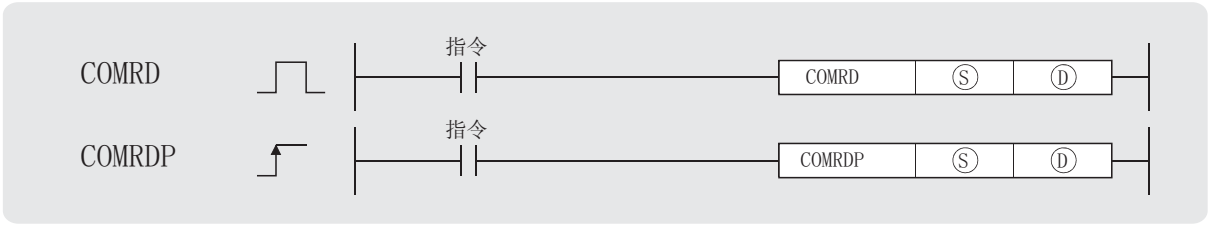
[动作]



7.11.7 读取软元件注释数据 (COMRD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当使用高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU 时

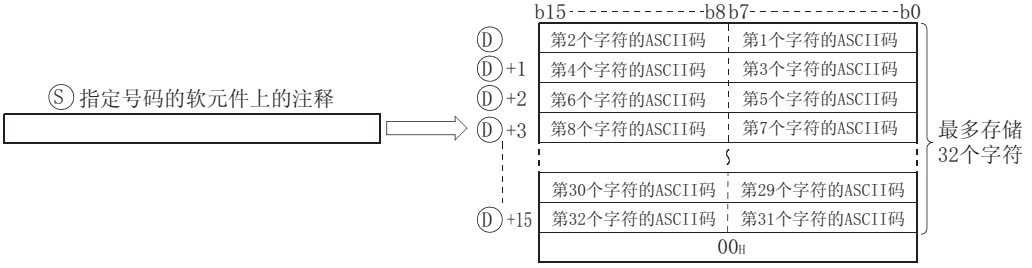


Ⓢ ：登记将被读取的注释的软元件的起始号（软元件名称）
Ⓓ ：存储已读取的注释数据的软元件的起始号（字符串）

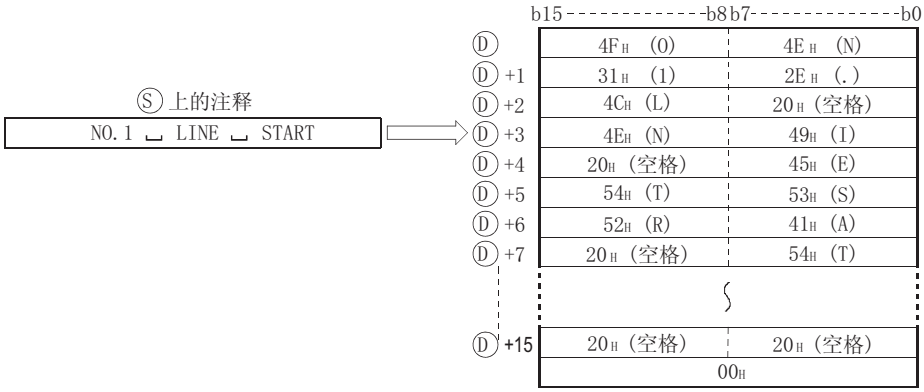
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				BL\S、BL\TR、BL、P、I、J、U
Ⓢ	○	○		○			---		○
Ⓓ	---	○		---			---		---

功 能

(1) 读取由 Ⓢ 指定的软元件上的注释，并以将其以 ASCII 码存储在 Ⓓ 指定号码的软元件之后。



例如，如果指定软元件 Ⓢ 的注释是“NO. 1␣ LINE ␣START”，则运行结果将被存储在 Ⓓ 之后，如下：



(2) 如果没有给 Ⓢ 指定的软元件登记注释，则该注释的所有字符将被处理为 “20_H”（空格）。

(3) 存储最后的 Ⓓ 字符的软元件号加 1 将依据 SM701（输出选择信号的字符数）的 ON/OFF 状态而有所不同。

当 SM701 为 OFF 时 ：不变
当 SM701 为 ON 时 ：存储 “0”

(4) 当读取注释时，该指令完成后 SM720 变为 ON 一个扫描周期，SM721 在该指令执行期间变为 ON。
当 SM721 为 ON 时，无法执行 COMRD(P) 指令，如果尝试执行，将不作处理。

要 点

1. 对于 COMRD(P) 指令使用的软元件注释，使用存储在存储卡中的注释文件。不能使用程序存储器中存储的注释文件。
2. 在参数模式下的 “可编程控制器文件设置”中设置 COMRD(P) 指令使用的注释。如果还未在可编程控制器文件设置中设置使用的注释文件，则 COMRD(P) 指令无法输出软元件注释。
3. 不要在中断程序中使用 COMRD(P) 指令。
否则，将导致产生故障。

运算错误

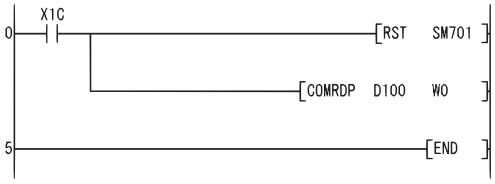
(1) 在以下情形中，返回一个错误，并且错误标志变为 ON。

- Ⓢ 指定的软元件号码的注释未登录。 （出错代码：4100）
- Ⓓ 指定的软元件号码不是字软元件。 （出错代码：4101）
- Ⓓ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码：4101）

程序示例

(1) 当 X1C 变为 ON 时，以下程序将 D100 上设置的注释以 ASCII 存储在 W0 中。

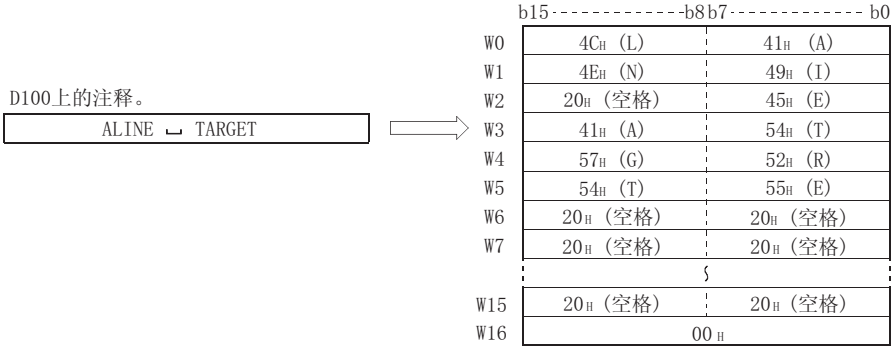
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X1C	
1	RST	SM701	
2	COMRDP	D100	W0
5	END		

[动作]



注意事项

(1) QCPU 在几次扫描后完成处理。

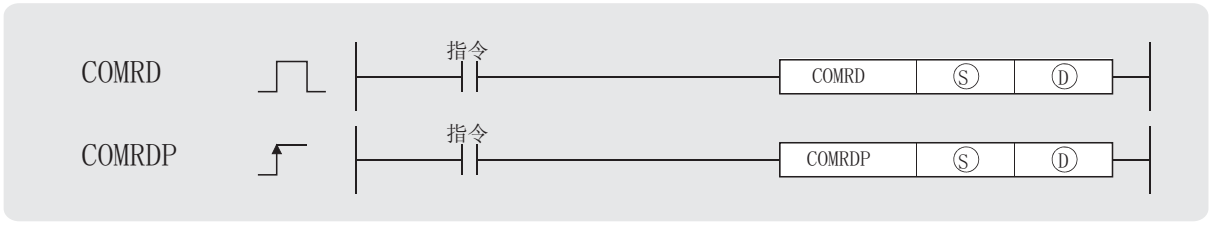
(2) 如果在该指令结束前（当 SM720 为 ON 时）将 COMRD(P)/PRC 指令的启动信号（命令）变为 ON，则不执行 COMRD(P)/PRC 指令。当 SM721 为 OFF 时，执行 COMRD(P)/PRC 指令。

(3) 不能同时访问两个或两个以上的文件注释。

(4) 以下指令由于共同使用 SM721，所以不能同时执行。

指令名	执行期间 ON	结束后 ON 一个扫描周期	异常结束后 ON
SP. FREAD SP. FWRITE	SM721	由指令指定	(指令指定的软元件)+1
PRC COMRD		SM720	无

2 使用 QnACPU/Q4ARCPU 时



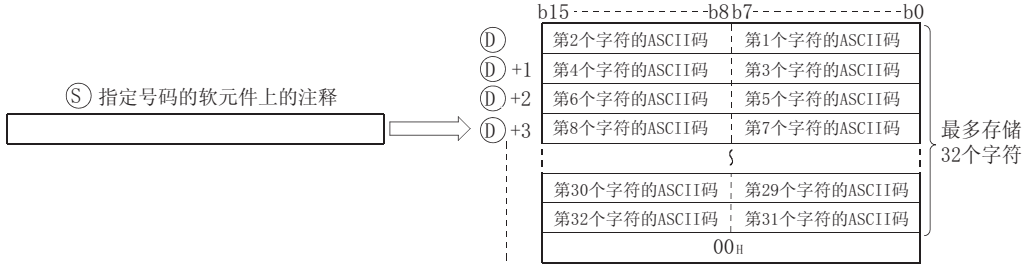
Ⓢ ：登记即将读取的注释的软元件的起始号（软元件名称）

Ⓓ ：存储已读取的注释数据的软元件的起始号（字符串）

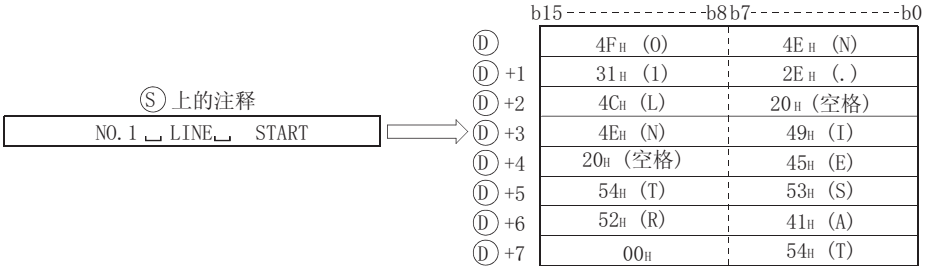
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它 BL\S、BL\TR、BL、P、I、J、U
	位	字		位	字				
Ⓢ	○	○		○			---		○
Ⓓ	---	○		---			---		---

功 能

(1) 读取由 Ⓢ 指定的软元件上的注释，并将其以 ASCII 码存储在由 Ⓓ 指定号码的软元件之后。



例如，如果由 Ⓢ 指定的软元件上的注释是 “NO. 1␣ LINE ␣START”，则运行结果将存储在 Ⓓ 后，如下：



(2) 能被 Ⓢ 指定号码的软元件仅限于注释范围内设定的那些号。

(3) 如果没有给 Ⓢ 指定的软元件登记注释，则该注释的所有字符将被处理为 “20_H”（空格）。

(4) 存储最后的 Ⓓ 字符的软元件号加 1，将依据 SM701（输出选择信号的字符数）的 ON/OFF 状态而有所不同。

当 SM701 为 OFF 时 ：存储 “0”
当 SM701 为 ON 时 ：不变

要 点

1. COMRD 指令中使用的软元件的注释，使用存储卡中存储的注释文件。不能使用内置存储器中存储的注释文件。
2. 使用参数模式中的 “可编程控制器文件设置”来设置 COMRD 指令使用的注释文件。
   如果可编程控制器文件设置中使用的注释文件还未指定，则软元件注释无法通过 COMRD 指令输出。

运算错误

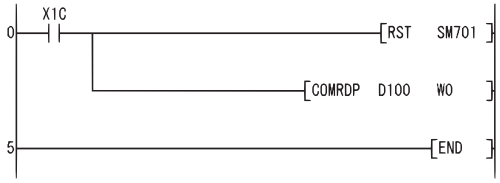
(1) 在以下情形中，返回一个错误，并且错误标志变为 ON。

- 由 Ⓢ 指定号码的软元件不在注释范围设置之内。 （出错代码： 4100）
- 由 Ⓓ 指定号码的软元件不是字软元件。 （出错代码： 4101）

程序示例

(1) 当 X1C 变为 ON 时，以下程序将 D100 上设置的注释以 ASCII 存储在 W0 中。

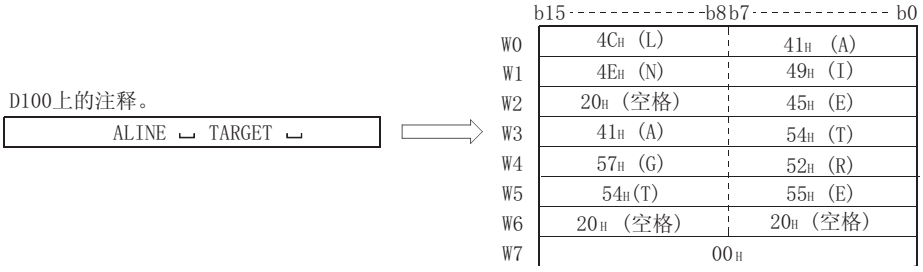
[梯形图模式]



[列表模式]

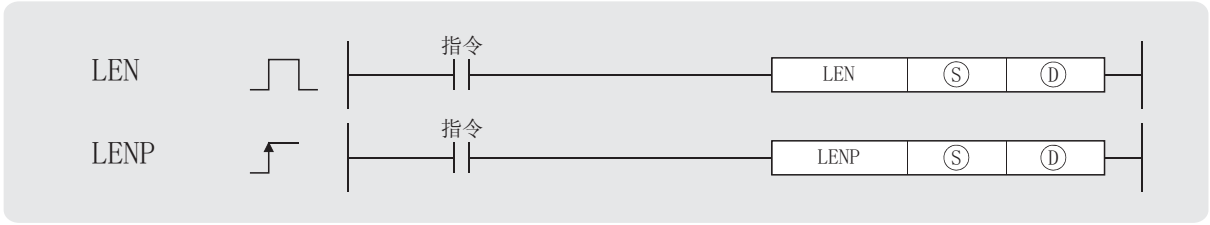
步	指令	软元件	
0	LD	X1C	
1	RST	SM701	
2	COMRDP	D100	W0
5	END		

[动作]



7.11.8 字符串长度检测 (LEN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ ：存储字符串的软元件的起始号（字符串）

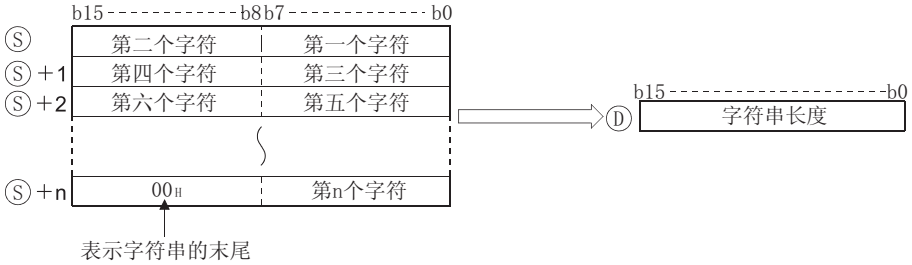
Ⓓ ：用来存储已检测字符串长度的软元件数 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	○	○		○				---	---

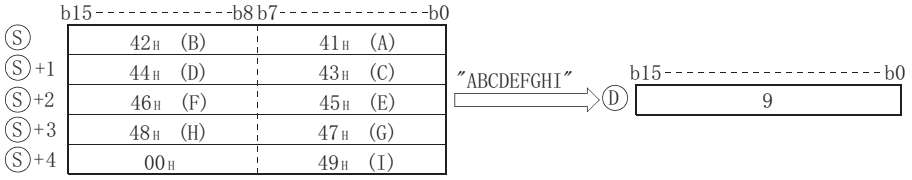
功 能

(1) 检测由 Ⓢ 指定的字符串的长度，并将其存储在由 Ⓓ 指定号码的软元件之后。

将从由 Ⓢ 指定号码的软元件开始到存储 “00H”的软元件之间的数据作为一个字符串处理。



例如，如果正从 Ⓢ 向前存储值 “ABCDEFGHI”，则值 9 将被存储在 Ⓓ 上。



运算错误

(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

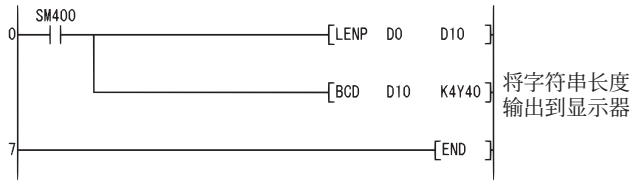
• 在由 Ⓢ 指定号码的软元件之后的相关软元件范围之内没有设定 “00H”。

（出错代码： 4101）

程序示例

(1) 以下程序将 D0 上的字符串的长度以 BCD 4 位值输出到 Y40 到 Y4F 上。

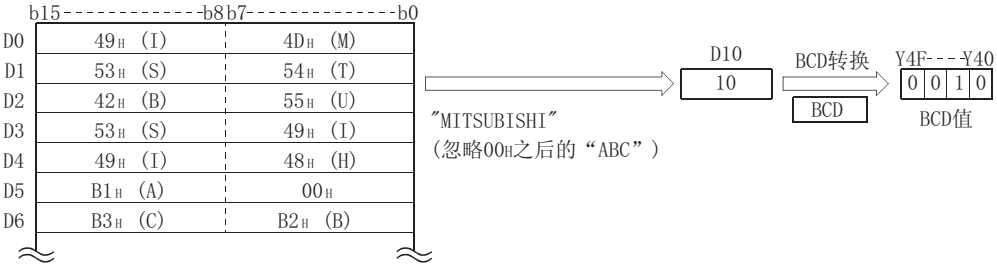
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	LENP	D0	D10
4	BCD	D10	K4Y40
7	END		

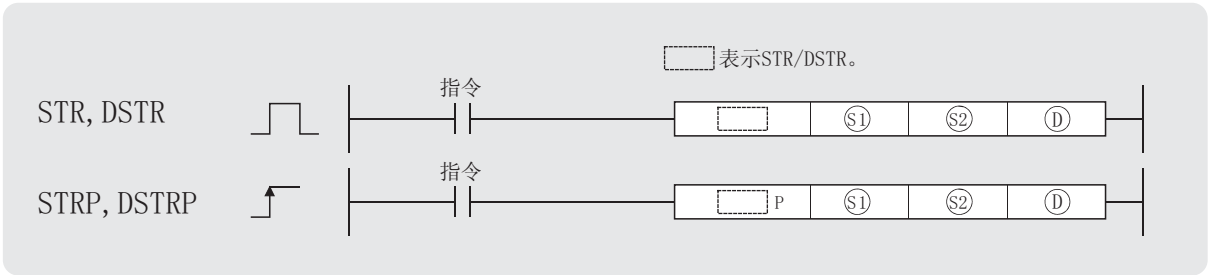
[动作]



7.11.9 从BIN16位或32位到字符串的转换 (STR(P)、DSTR(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号前五位是 04122 或以后。



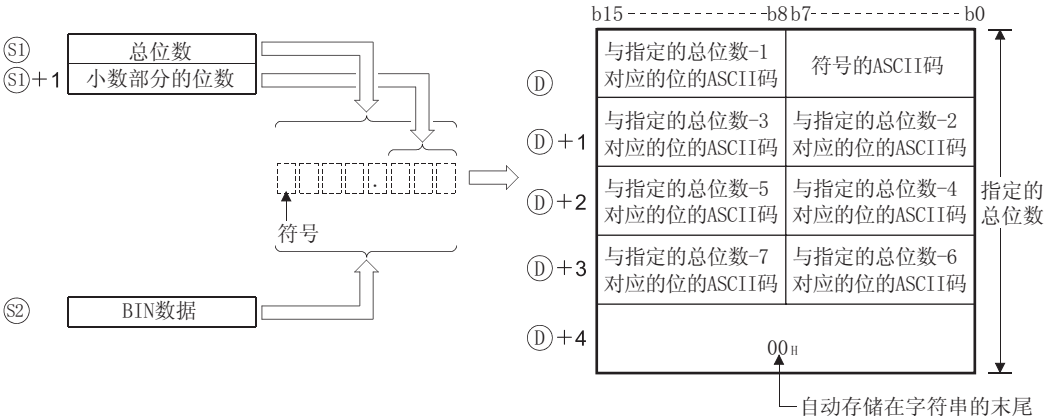
Ⓢ1 ：存储将要转换的数值的位数的软元件的起始号（BIN 16 位）
Ⓢ2 ：将被转换的 BIN 数据（BIN 16/32 位）
Ⓓ ：存储已转换字符串的软元件的起始号（字符串）

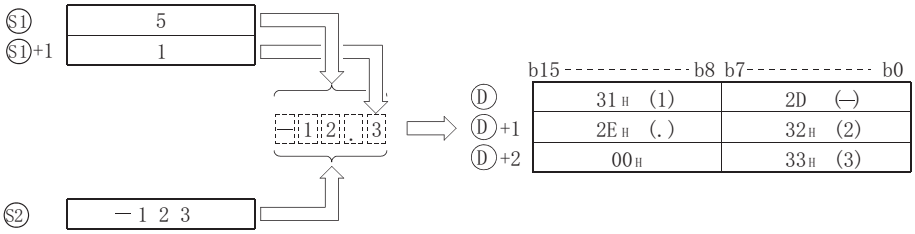
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ1	○	○		○				---	---
Ⓢ2	○	○		○				○	---
Ⓓ	---	○		---				---	---

功 能

STR

(1) 在由 Ⓢ1 指定的位置上，给由 Ⓢ2 指定的 BIN 16 位数据添加一个小数点，并将该数据转换成字符串数据，从由 Ⓓ 指定号码的软元件开始将其存储起来。





(2) ⓢ1 能够指定的总位数是 2 到 8。

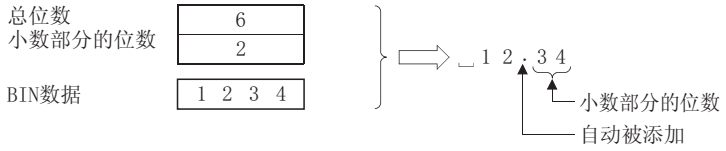
(3) 能够被 ⓢ1+1 指定的作为小数点后的一部分的位数是 0 到 5。
但是，小数点后面的位数必须小于或等于总位数减 3。

(4) 在 -32768 和 32767 之间内的 BIN 数据可在 ⓢ2 上指定。

(5) 转换之后，字符串数据按照下面所示被存储在号码在 ⓓ 之后的软元件上：

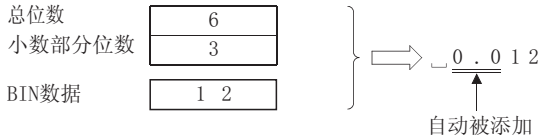
(a) 如果 BIN 数据为正，则存储符号 “20H”（空格）；如果为负，则存储符号 “2DH”（负号 -）。

(b) 如果小数点之后的位数的设置是 0 以外的任何数字，则 “2EH”(.) 将被自动存储在指定位数的第一位之前的位置。

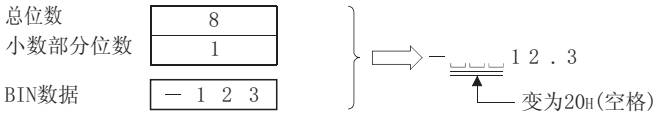


如果小数部分的位数是 0，则该值不存储 “2EH”(.)。

(c) 如果小数点之后的总位数比 BIN 数据的位数大，将自动增加一个零，并通过向右移位来转换该数据，从而变为 “0.□□□□□”。



(d) 如果除了符号和小数点之外的总位数大于 BIN 数据的位数，则 “20H”（空格）将被存储在符号和数值之间。

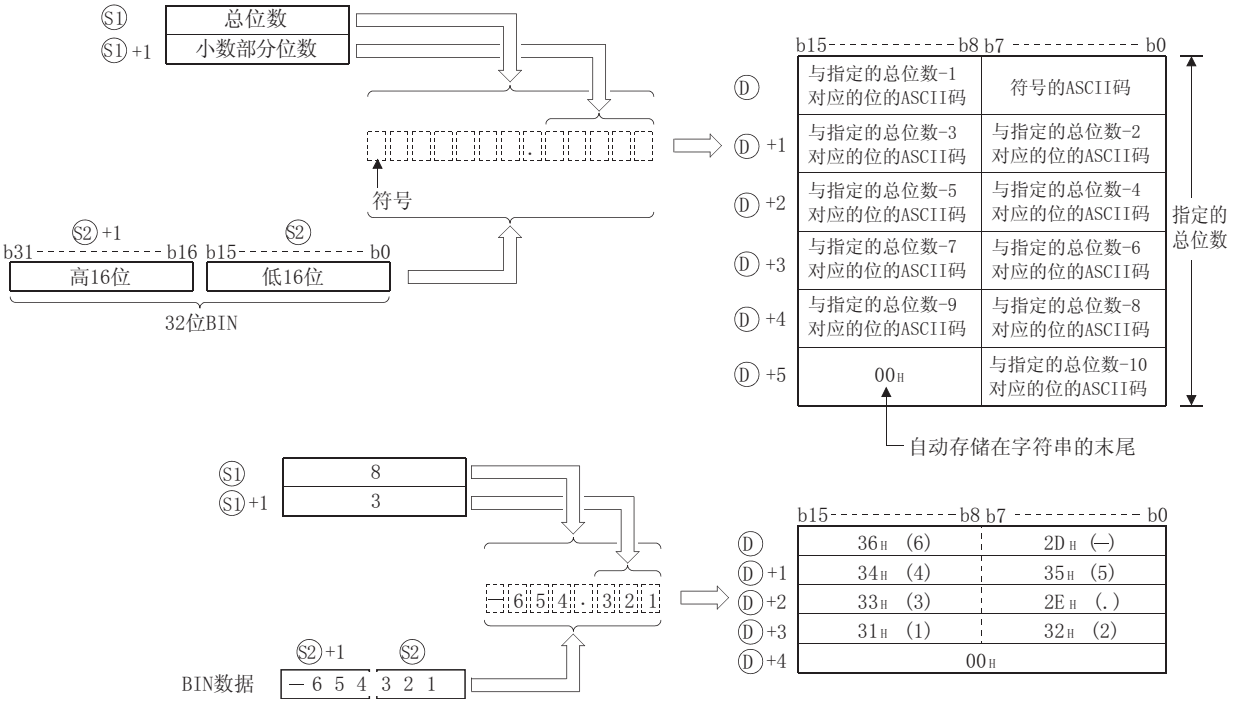


如果 BIN 的位数较大，将返回一个错误。

(e) 值 “00H” 被自动存储在已转换字符串的末端。

DSTR

(1) 在由 (S1) 指定的位置上，给由 (S2) 指定的 BIN 32 位数据添加一个小数点，将该数据转换成字符串数据，并将其存储在由 (D) 指定号码的软元件之后。



(2) (S1) 能够指定的总位数在 2 到 13 之间。

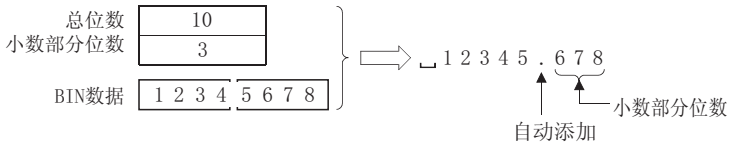
(3) 能够被 (S1)+1 指定为小数的一部分的位数是 0 到 10。
但是，小数点后面的位数必须小于或等于总位数减 3。

(4) (S1) 和 (S2)+1 可以指定的 BIN 数据在 -2147483648 到 2147483647 之间。

(5) 转换之后的字符串数据按如下所示存储在软元件中：

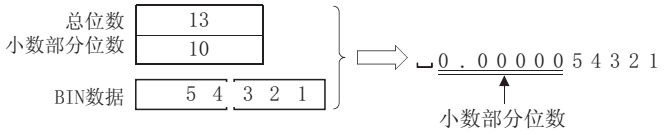
(a) 如果 BIN 数据为正，则存储符号 “20H”（空格）；如果为负，则存储符号 “2DH”（负号 -）。

(b) 如果小数点后的位数的设置是 0 以外的任何数据，则 “2EH”(-) 将被自动存储在指定位数的第一位之前。

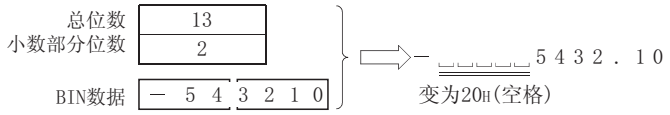


如果小数部分的位数是 0，则该值不存储 “2EH”。

(c) 如果小数部分比 BIN 数据有更多的位数，将自动增加一个 0，并将该值向右移位，这样转换后的值将变为 “ ″0.□□□□□”。



(d) 如果除了符号和小数点之外的总位数大于 BIN 数据位数，则 “20H”（空格）将被存储在符号和数值之间。



如果 BIN 的位数较大，将返回一个错误。

(e) 值 “00H”被自动存储在已转换字符串的末尾。

运算错误

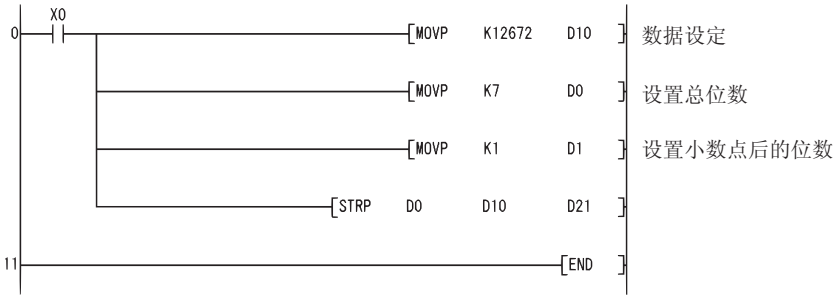
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- 由 (S1) 指定的总位数超出了以下所示的范围。 （出错代码： 4100）
  当使用 STR 指令时.....2 到 8
  当使用 DSTR 指令时.....2 到 13
- 由 (S1)+1 指定的小数部分的位数超出了以下所示的范围。 （出错代码： 4100）
  当使用 STR 指令时.....0 到 5
  当使用 DSTR 指令时.....0 到 10
- 由 (S1) 指定的总位数和由 (S1)+1 指定的小数部分的位数之间的关系不符合以下所示：
  （出错代码： 4100）
  总位数减 3 等于或大于小数部分的位数。
- 由 (S1) 指定的位数小于 “2+ (S2) 指定的 BIN 数据的位数”。
  ((S1) < 的位数 (S2) 指定的不包含符号的 BIN 数据的位数 + 符号 (+ 或 -) 的位数 + 小数点 (.) 的位数） （出错代码： 4100）
- 存储由 (D) 指定的字符串的软元件范围超出了相应的软元件范围。
  （出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序根据 D0 和 D1 的位指定转换存储在 D10 上的 BIN 16 位数据，并将结果存储在 D20 到 D23 上。

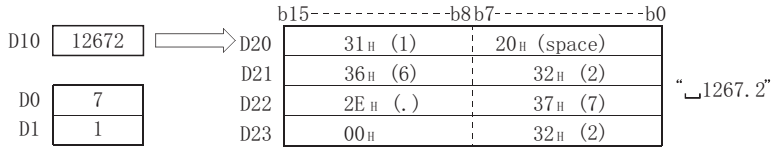
[梯形图模式]



[列表模式]

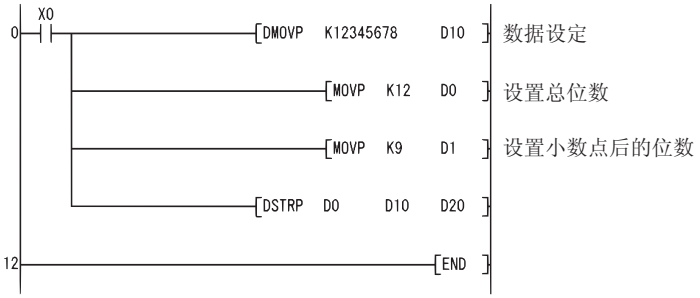
步	指令	软元件		
0	LD	X0		
1	MOVP	K12672	D10	
3	MOVP	K7	D0	
5	MOVP	K1	D1	
7	STRP	D0	D10	D21
11	END			

[动作]



(2) 当 X0 变为 ON 时，以下程序根据 D0 和 D1 上的位指定转换存储在 D10 和 D11 上的 BIN 32 位数据，并将结果存储在 D20 到 D26 上。

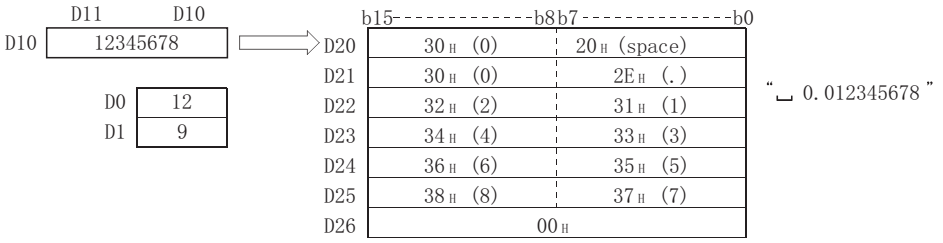
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	DMOVP	K12345678		D10
4	MOVP	K12	D0	
6	MOVP	K9	D1	
8	DSTRP	D0	D10	D20
12	END			

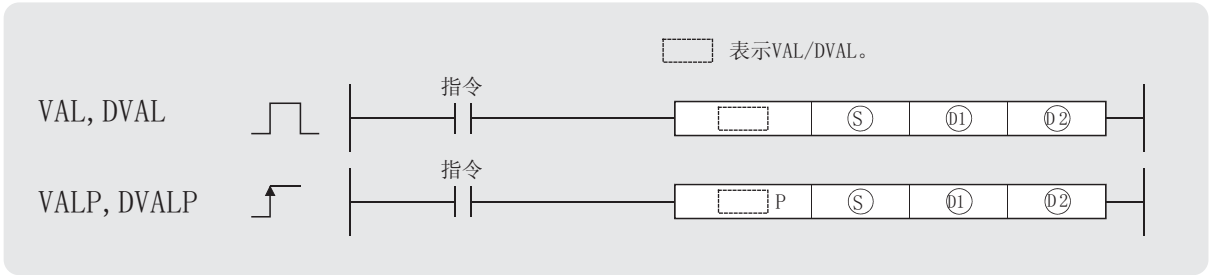
[动作]



7.11.10 从字符串到BIN16位或32位数据的转换(VAL(P)、DVAL(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号前五位是04122或以后。
（对应GX Developer：版本8.00A或以后）。



Ⓢ：存储即将转换为BIN数据的字符串的软元件的起始号（字符串）
Ⓓ1：存储转换后的BIN数据位数的软元件的起始号（BIN 16位）
Ⓓ2：存储转换后的BIN数据的软元件的起始号（BIN 16/32位）

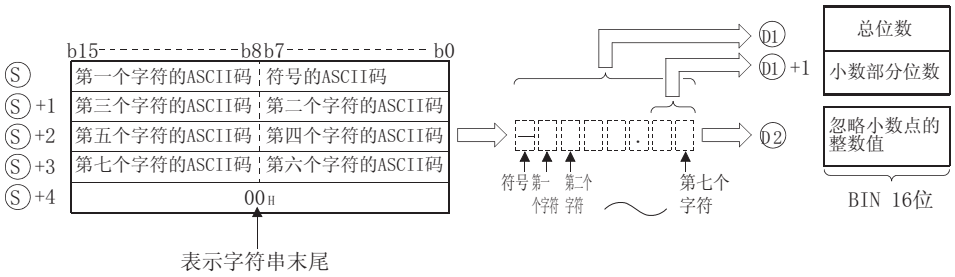
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ1	○	○		---				---	---
Ⓓ2	○	○		○				---	---

功 能

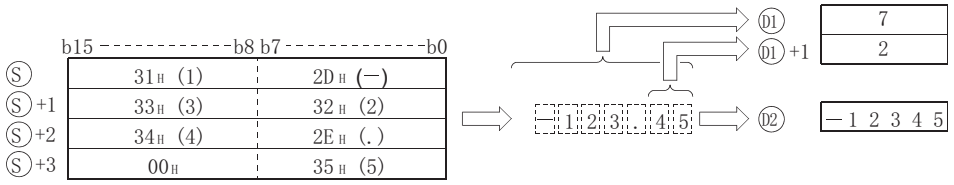
VAL

(1) 将存储在由 Ⓢ 指定号码的软元件开始的软元件上的字符串转换成BIN 16位数据，并将位数和BIN数据存储在 Ⓓ1 和 Ⓓ2 中。

对于字符串到BIN的转换，从由 Ⓢ 指定号码的软元件开始到存储“00H”的软元件点之间的所有数据将被作为字符串处理。



例如，如果 (S) 之后的软元件指定了字符串“-123.45”，则运算结果将按照以下方式存储在 (D1) 和 (D2) 上：



(2) 可在 (S) 上指定的字符串的总字符数在 2 到 8 之间。

(3) 从 (S) 指定的字符串开始的 0 到 5 个字符可成为小数部分。
但是，该数值必须不超过总位数减 3。

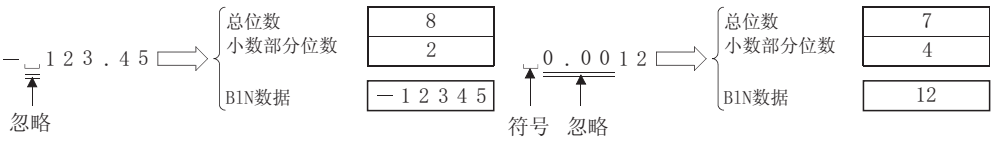
(4) 当忽略小数点时，可转换成 BIN 值的字符串的数值为 -32768 到 32767。
数值字符串，除符号和小数点之外，仅可在 “30H” 到 “39H” 的范围内指定。
当忽略小数点时，该值表示如下：
示例： “-12345.6” 被表示为 “-123456”。

(5) 如果数值为正，则存储符号 “20H”；如果为负，则存储符号 “2DH”。

(6) 为小数点设置 “2EH”。

(7) (D1) 上存储的总位数是表示数值（包括符号和小数点）的所有字符的总和。
在 (D1)+1 上存储的小数点之后的字符包括从 “2EH”(.) 往前的字符数。
在 (D2) 上存储的 BIN 数据是已被转换成 BIN 数据、忽略小数点的字符串。

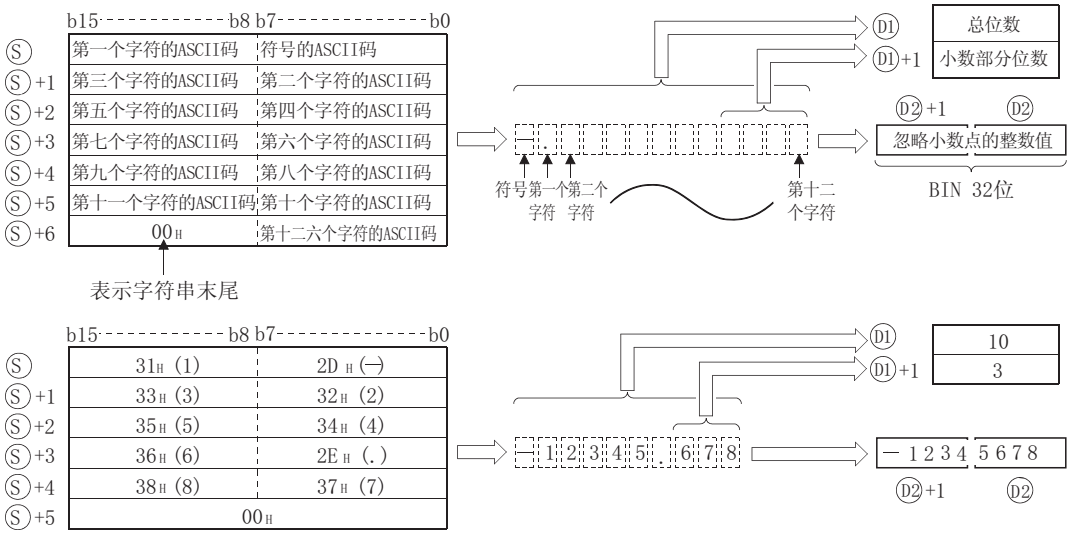
(8) 如果由 (S) 指定的字符串中在符号和第一个非零的数值之间有一个 “200H”（空格）或者一个 “30H”(0)，则在将其转换到 BIN 值期间忽略 “20H” 或 “30H”。



DVAL

(1) 将存储在由 (S) 指定号码的软元件之后的字符串转换成 BIN 32 位数据，并存储在 (D1) 和 (D2) 上。

对于从字符串到 BIN 的转换，从由 (S) 指定号码的软元件开始到存储 “00H”的软元件点之间的所有数据将被作为字符串处理。



(2) 由 (S) 指定的字符串的总字符数在 2 到 13 之间。

(3) 由 (S) 指定的字符串的 0 到 10 个字符可成为小数部分。

但是，该数值必须不超过总位数减 3。

(4) 当忽略小数点时，可转换成 BIN 值的数值字符串是那些落在从 -2147483648 到 2147483647 范围内的字符。

数值字符串，除符号和小数点之外，仅可在从 “30H”到 “39H”的范围内指定。

(5) 如果数值为正，则存储符号 “20H”″；如果为负，则存储符号 “2DH”。

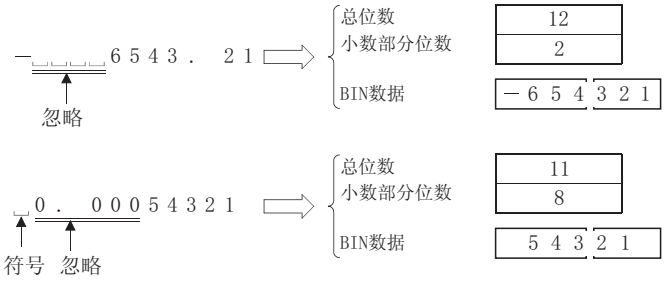
(6) 为小数点设定 “2EH”。

(7) 在 (D1) 上存储的总位数是表示数值（包括符号和小数点）的所有字符的总和。

在 (D1)+1 上存储的小数点之后的字符包括从 “2EH”(.) 往前的字符数。

在 (D2) 上存储的 BIN 数据是已被转换成 BIN 数据、忽略小数点的字符串。

(8) 如果由 Ⓢ 指定的字符串中，在符号和第一个非零的数值之间有一个 “20H”（空格）或者一个 “30H”(0)，则在将其转换到 BIN 值期间忽略 “20H” 或 “30H”。



运算错误

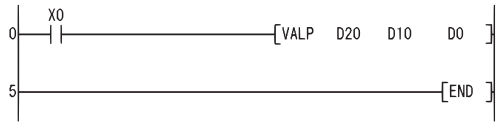
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- 由 Ⓢ1 指定的字符串的字符数超出了以下所示的范围： （出错代码： 4100）
  当使用 VAL 指令时..... 2 到 8 个字符
  当使用 DVAL 指令时..... 2 到 13 个字符
- 由 Ⓢ 指定的字符串的小数部分的字符数超出了以下所示的范围 :（出错代码： 4100）
  当使用 VAL 指令时..... 0 到 5
  当使用 DVAL 指令时..... 0 到 10
- 由 Ⓢ 指定的字符串的总字符数和小数部分的字符数之间的关系超出了以下所示的范围：
  （出错代码： 4100）
  总字符数减 3 等于或大于小数部分的字符数。
- 已经为符号设定了一个非 “20H” 或 “2DH” 的 ASCII 码。 （出错代码： 4100）
- 已经将一个非 “30H” 到 “39H” 或 “2EH”（小数点）的 ASCII 码设定给某个数的一位。
  （出错代码： 4100）
- 数值中设置的小数点不止一个。 （出错代码： 4100）
- 转换后的 BIN 数据值超出了以下范围： （出错代码： 4100）
  当使用 VAL 指令时..... －32768 到 32767
  当使用 DVAL 指令时..... －2147483648 到 2147483647
- ASCII 码 “00H” 已经被设置在由 Ⓢ 指定号码的软元件和相关软元件范围的最后软元件号之间的某个地方。 （出错代码： 4101）

程序示例

(1) 当 X0 为 ON 时，以下程序将存储在 D20 到 D22 上的字符串数据当作一个整数读取，并将其转换成一个 BIN 值，存储在 D0 上。

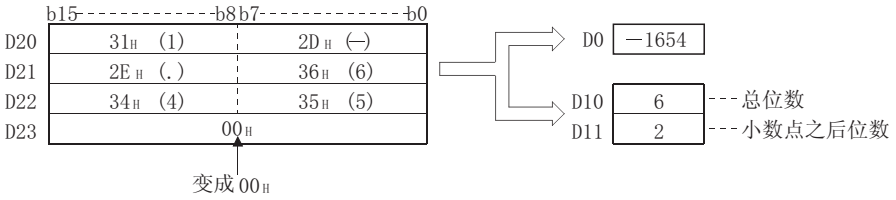
[梯形图模式]



[列表模式]

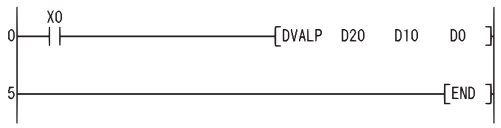
步	指令	软元件		
0	LD	X0		
1	VALP	D20	D10	D0
5	END			

[动作]



(2) 当 X0 为 ON 时，以下程序将 D20 到 D24 上的字符串数据当作一个整数读取，将其转换成一个 BIN 值，并存储在 D0 上。

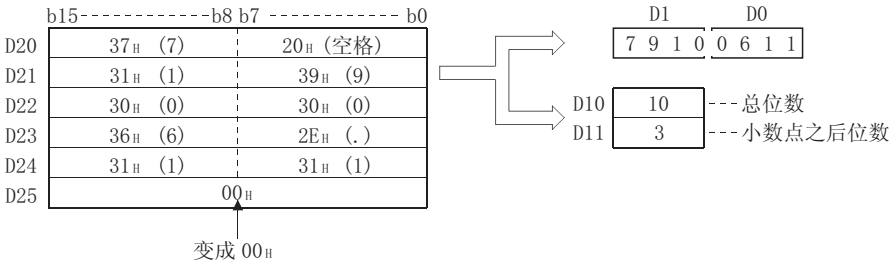
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	DVALP	D20	D10	D0
5	END			

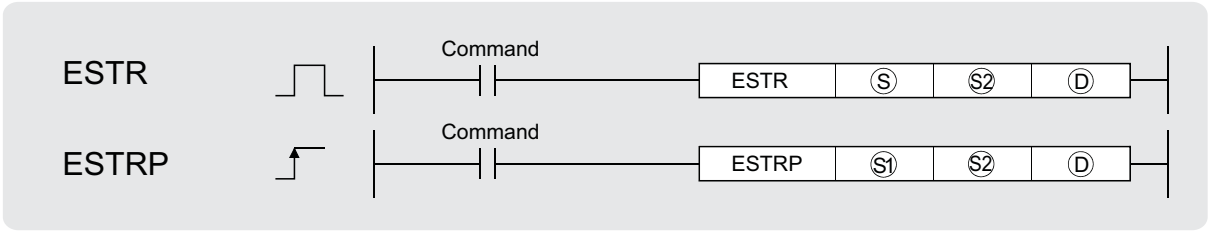
[动作]



7.11.11 从浮点型到字符串数据的转换 (ESTR(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号前五位是 04122 或以后。



(S1)：即将转换的浮点数，或者存储这类数据的软元件的起始号（实数）

(S2)：存储将要转换的数值的显示指定的软元件的起始软元件号（BIN 16 位）

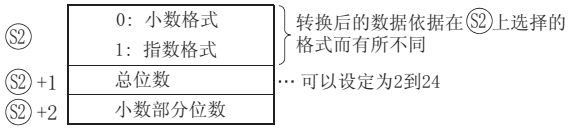
(D)：存储已转换字符串的软元件的起始软元件号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
(S1)	---	○		—	○		—	○	—
(S2)	---	○		—	—		—	—	—
(D)	---	○		—	—		—	—	—

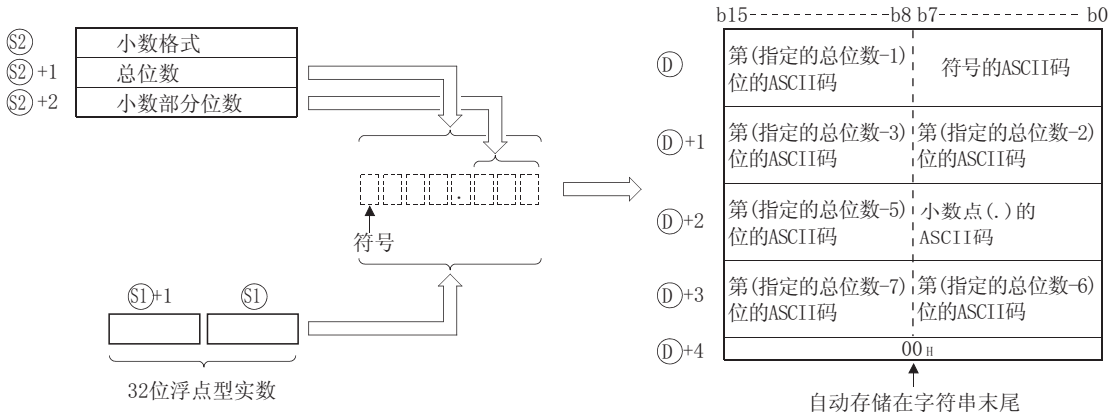
功 能

(1) 根据存储在由 (S2) 指定的软元件之后的显示指定，将由 (S1) 指定的软元件上存储的浮点型实数转换成字符串，并将结果存储在由 (D) 指定号码的软元件之后。

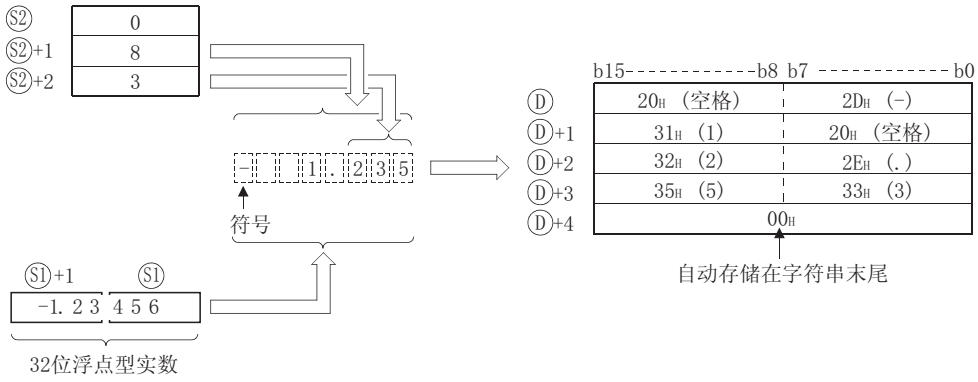
(2) 转换后的数据将依据由 (S2) 指定的显示指定而有所不同。



当使用小数格式时



例如，如果共有 8 位，其中 3 位来自小数部分，且指定的值为 -1.23456，则运算结果将被按照以下方式存储在 ⓓ 之后。



(a) ⓢ2+1 可以指定的总位数显示如下：

当小数点之后的位数是 “0” 时

..... 位数（最多 :24 ≧ 2

当小数点之后的位数是某个非 “0” 值时

..... 位数（最多 :24）≧（小数点之后的位数 +3）

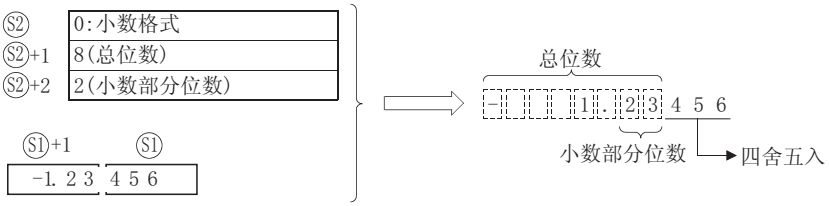
(b) 小数点后可被 ⓢ2+2 指定的位数在 0 到 7 之间。

但是，小数点后的位数应该等于或小于总位数减 3。

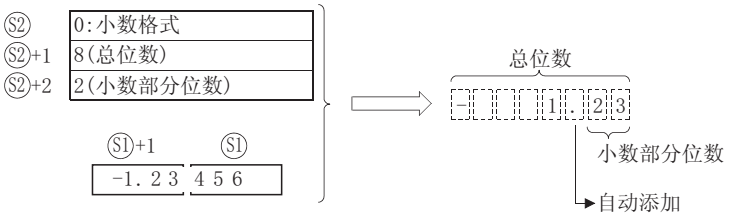
(c) 转换后，按如下所示将字符串数据存储在 ⓓ 之后的软元件上：

1) 如果浮点型实数为正，则存储符号 “20H”（空格）；如果为负，则存储符号 “2DH”（负号 -）。

2) 如果 32 位浮点型实数数据的小数部分超出了小数部分的位数范围，则将四舍五入低位的小数部分。

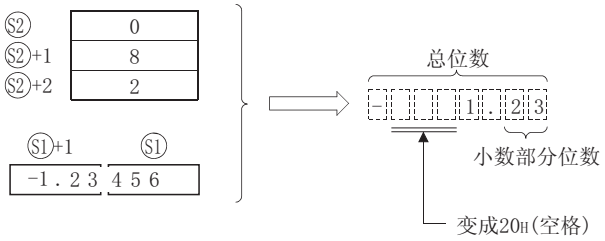


3) 如果小数点后的位数被设置成 0 以外的其他值，则 “2EH”(.) 将被自动存储在指定位数的第一位之前的位置。



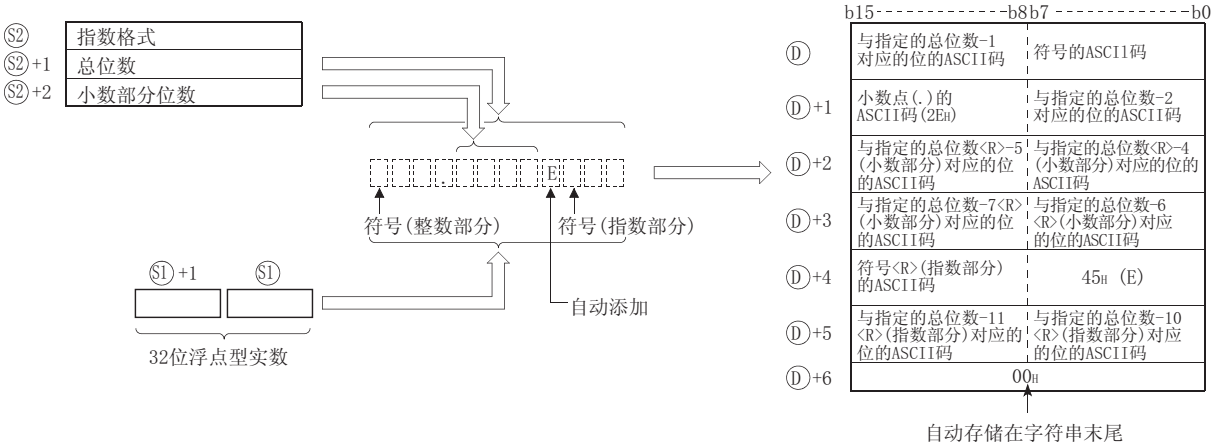
如果该数的小数部分位数为 “0”，则不存储 ASCII 码 “2EH”(.)。

4) 如果总位数，不包括符号、小数点和小数点之后的数，大于浮点型实数的小数点之前的部分，则将在符号和该数据的小数点之前的部分之间存储 “20H(空格)”。

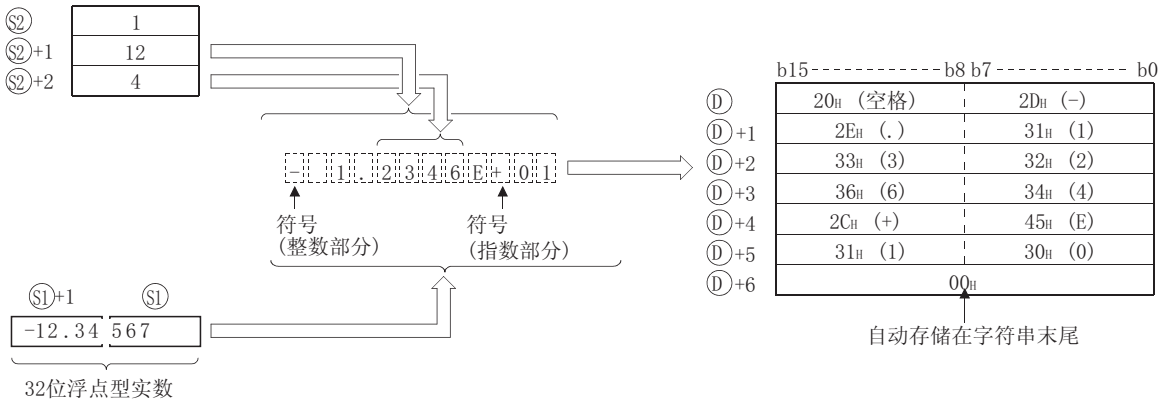


5) 在已转换的字符串的末尾自动存储值 “00H”。

当使用指数格式时



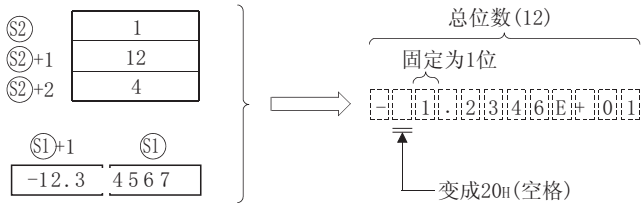
例如，如果总共有 12 位，其中小数部分有 4 位，且指定值是 -12.34567，则运算结果将按照以下方式存储在 (D) 之后。



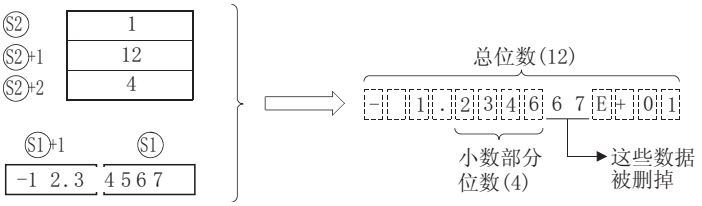
(a) ⓢ2+1 可指定的总位数显示如下：
当小数点后的位数为 “0” 时
..... 总位数 (: 24) ≧ 2
当小数点后的位数是某个非 “0” 值时
..... 总位数 (: 24) ≧ (小数部分位数 + 7)

(b) 可指定的跟在小数点之后的位数在 0 到 7 之间。
但是，小数部分的位数应等于或小于总位数减 7。

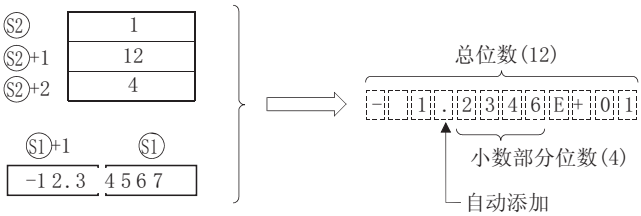
(c) 转换后，按照以下的指示将字符串数据存储在 ⓓ 之后的软元件上：
1) 如果浮点型实数的值为正，则整数前的符号将被存储为 ASCII 码 “20H” (空格)；如果是一个负值，则符号将被存储为 “2DH” (-)。
2) 整数部分被固定为一位。
“20H” (空格) 将被存储在整数和符号之间。



3) 如果浮点型实数太长，超出了总位数能容纳的范围，则小数部分的尾数将从结果中删除。



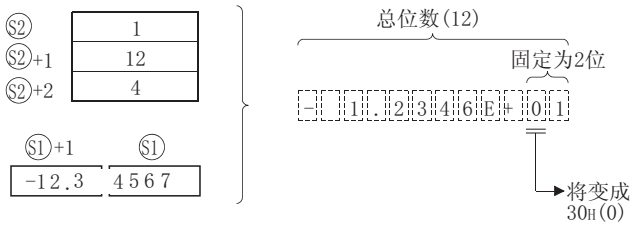
4) 如果小数点之后的位数已被设定为任何非 “0” 的值，则 “2EH” (.) 将被自动存储在指定位数第一位之前的位置。



如果该数的小数部分位数为 “0”，则将不存储 ASCII 码 “2EH” (.)。

5) 如果指数值为正，则将存储 ASCII 码 “2CH”(+) 作为该值指数部分的符号；如果指数是一个负值，则将存储 “2DH”(-)。

6) 指数部分固定为 2 位。
如果指数部分仅 1 位，则将在该数的符号和指数之间存储 ASCII 码 “30H”(0)。



7) 自动将值 “00H”存储在已转换字符串的末尾处。

运算错误

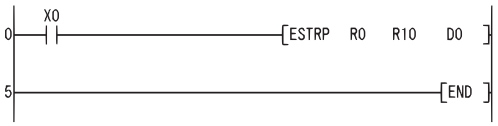
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON。

- (S1) 值不在以下所示的范围内： (出错代码： 4100)
  0、 $2^{-126} \leqq |$ (S1) $| < 2^{128}$
- (S2) 指定的格式既不是 0 也不是 1。 (出错代码： 4100)
- (S2) + 1 指定的总位数超出了以下所示的范围： (出错代码： 4100)
  当使用小数格式时
  当小数点后的位数为 “0” 时
  ..... 总位数 ≧ 2
  当小数点后的位数是某个非 “0” 值时
  ..... 总位数 ≧ (小数部分位数 + 3)
  当使用指数时
  当小数点后的数为 “0” 时
  ..... 总位数 ≧ 6
  当小数点后的位数是某个非 “0” 值时
  ..... 总位数 ≧ (小数部分位数 + 7)
- 给 (S2)+2 指定值的小数部分指定的位数超出了以下所示的范围： (出错代码： 4100)
  当使用小数格式时
  ..... 小数部分位数小于或等于总位数减 3。
  当使用指数格式时
  ..... 小数部分位数小于或等于总位数减 7。
- 已指定总位数超出 “24” 的值。 (出错代码： 4100)
- 存储由 (D) 指定的字符串的软元件范围超出了相应的软元件范围。(出错代码： 4101)
- 由 (S2) 指定的软元件超出相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)
- 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$ 。
  (只对于通用型 QCPU。) (出错代码： 4140)

程序示例

(1) 当 X0 变 ON 时，以下程序根据 R10 到 R12 上存储的转换指定，转换存储在 R0 和 R1 上的浮点型实数，并将结果存储在 D0 之后。

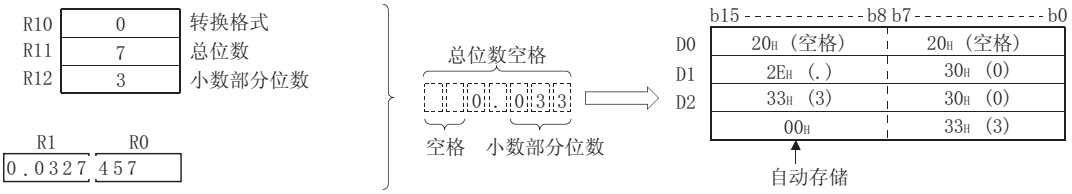
[梯形图模式]



[列表模式]

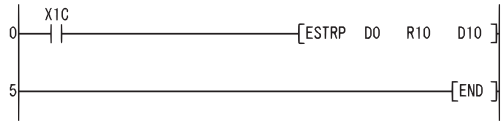
步	指令	软元件		
0	LD	X0		
1	ESTRP	R0	R10	D0
5	END			

[动作]



(2) 当 X1C 变 ON 时，以下程序根据 R10 到 R12 上存储的转换指定，转换存储在 D0 和 D1 上的浮点型实数，并将结果存储在 D0 之后。

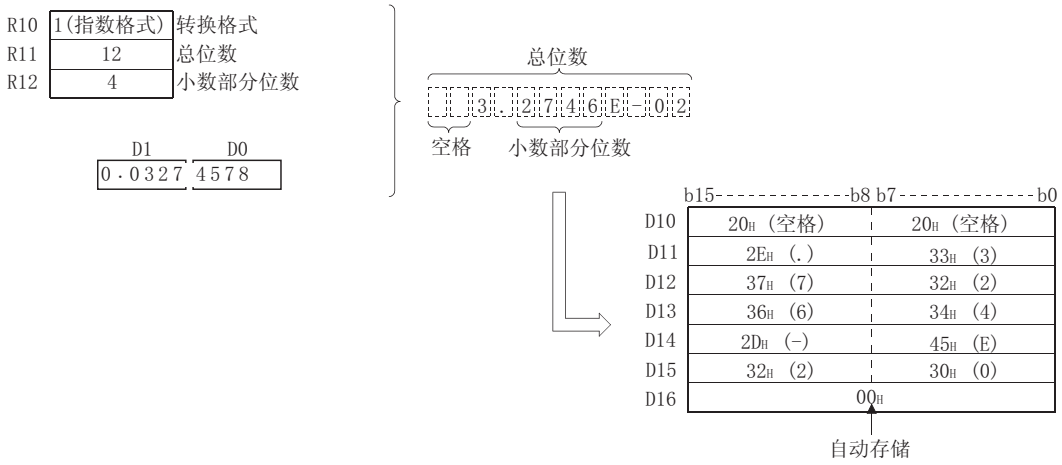
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X1C		
1	ESTRP	D0	R10	D10
5	END			

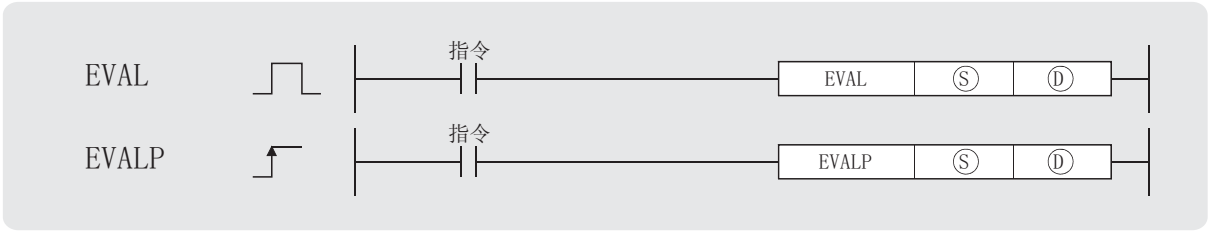
[动作]



7.11.12 从字符串到浮点数的转换 (EVAL(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：序列号前五位是 04122 或以后。



Ⓢ：即将转换为浮点型实数的字符串数据，或存储该数据的软元件的起始软元件号（字符串）

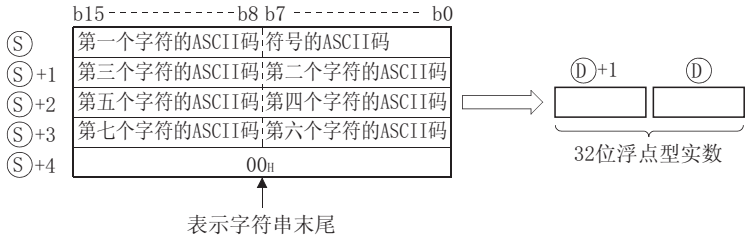
Ⓓ：转换后存储浮点型实数的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	—	○		—	—		—	○	—
Ⓓ	—	○		—	○		—	—	—

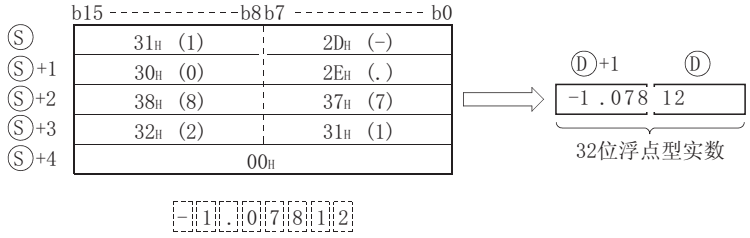
功 能

(1) 将存储在由 Ⓢ 指定号码的软元件之后的字符串转换成 32 位浮点型实数，并将结果存储在 Ⓓ 指定的软元件中。

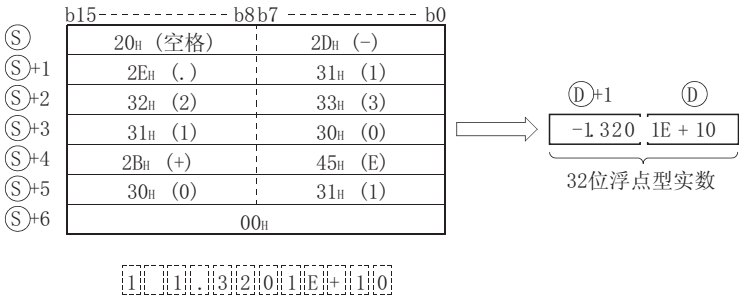
(2) 指定的字符串能够以小数格式或者指数格式转换成 32 位浮点型实数。



(a) 当使用小数格式时

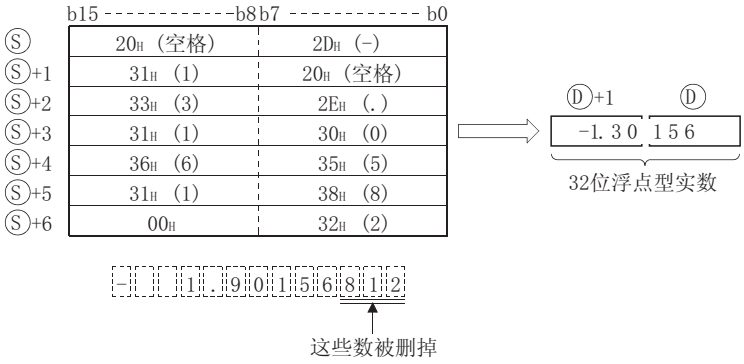


(b) 当使用指数格式时

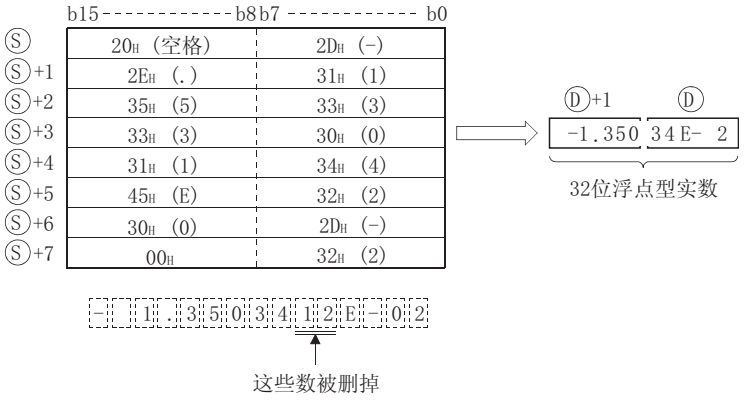


(3) 除了结果的符号、小数点和指数部分之外，由 Ⓢ 指定的即将转换成浮点型实数的字符串的6位将是有效的，自第7位开始的所有数据将从结果中删除。

(a) 当使用小数格式时



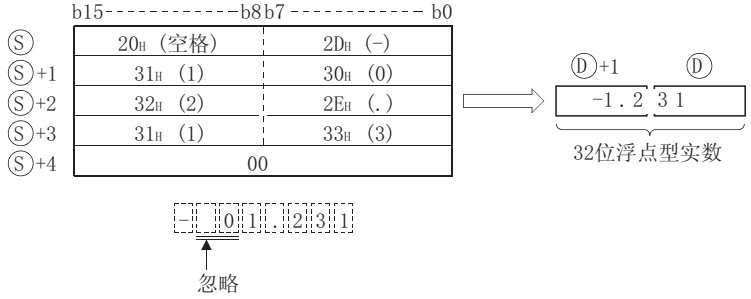
(b) 当使用指数格式时



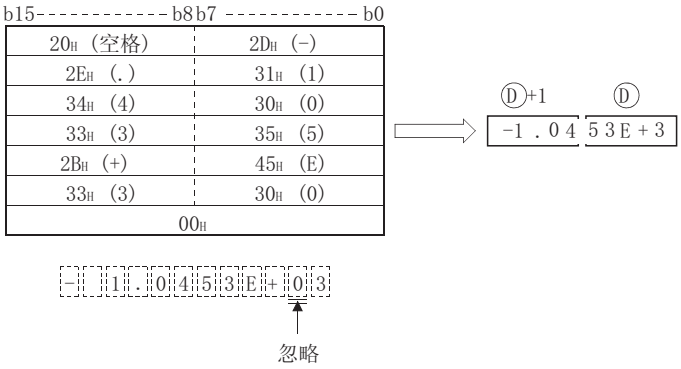
(4) 在小数点格式中，如果将符号指定为 “2BH”(+) 或者省略符号指定，将转换为正值。
如果将符号指定为 “2DH”(-)，字符串将被转换为负值。

(5) 在指数格式中，如果将指数部分的符号指定为 “2BH”(+) 或者省略符号指定，将转换为正值。
如果将指数部分的符号指定为 “2DH”(-)，字符串将被转换为负值。

(6) 如果，在 Ⓢ 指定的字符串中，除起始的零之外，有 ASCII 码 “20H”（空格）或 “30H”(0)，则在转换时将忽略它们。



(7) 如果在一个指数格式的字符串中，在字符 “E”和一个数之间存在 ASCII 码 “30H”(0)，则当进行转换时将忽略 “30H”。



(8) 如果在字符串内包含有 “20H”（空格）码时，转换时将忽略此代码。

(9) 一个字符串最多可设定 24 个字符。
字符串中的 “20H”（空格）和 “30H”(0) 码也将被计数为 1 个字符。

运算错误

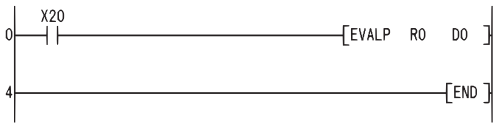
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- 整数部分或者小数部分含有一个不在 “30H”(0) 到 “39H”(9) 范围之内的字符。
（出错代码：4100）
- 字符 “2EH”(.) 在由 Ⓓ 指定的字符串中出现了两次或两次以上。
（出错代码：4100）
- 该值的指数部分包含有 “45H”(E)、“2CH”(+) 或 “45H”(E)、“2DH”(-) 之外的字符，或者有不止一个指数。
（出错代码：4100）
- 转换后的数据不在以下范围内：（出错代码：4100）
0、$2^{-126} \leqq$ ｜转换后的数据｜ $< 2^{128}$
- “00H”代码没有在 Ⓢ 到相关软元件的范围内出现。（出错代码：4101）
- 字符串中的字符数是 0 或大于 24。（出错代码：4100）

程序示例

(1) 当 X20 变为 ON 时，以下程序将存储在 R0 之后的字符串转换成浮点型实数，并将结果存储在 D0 和 D1 上。

[梯形图模式]



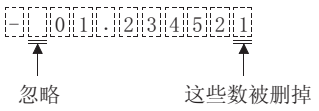
[列表模式]

步	指令	软元件	
0	LD	X20	
1	EVALP	R0	D0
4	END		

[动作]

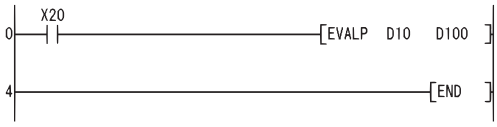
	b15 ---------- b8	b7 ---------- b0
R0	20_H (空格)	$2D_H$ (-)
R1	31_H (1)	30_H (0)
R2	32_H (2)	$2E_H$ (.)
R3	34_H (4)	33_H (3)
R4	32_H (2)	35_H (5)
R5	00_H	31_H (1)

D1	D0
-1.234	52



(2) 当 X20 变为 ON 时，以下程序将存储在 D10 之后的字符串转换成浮点型实数，并将结果存储在 D100 和 D101 上。

[梯形图模式]



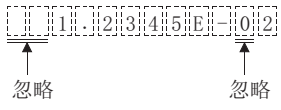
[列表模式]

步	指令	软元件	
0	LD	X20	
1	EVALP	D10	D100
4	END		

[动作]

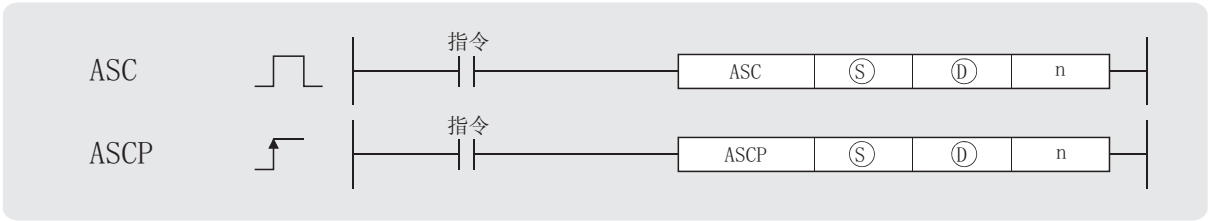
	b15 ---------- b8	b7 ---------- b0
D10	20_H (空格)	20_H (空格)
D11	$2E_H$ (.)	31_H (1)
D12	33_H (3)	32_H (2)
D13	35_H (5)	34_H (4)
D14	$2D_H$ (-)	45_H (E)
D15	32_H (2)	30_H (0)
D16	00	

D101	D100
1.234	5E -2



7.11.13 从十六进制 BIN 到 ASCII 码的转换 (ASC(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

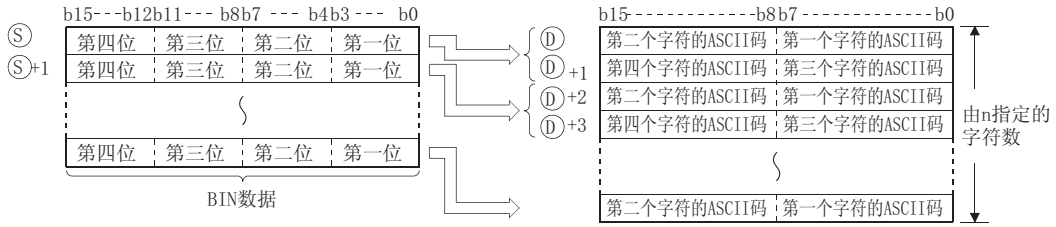


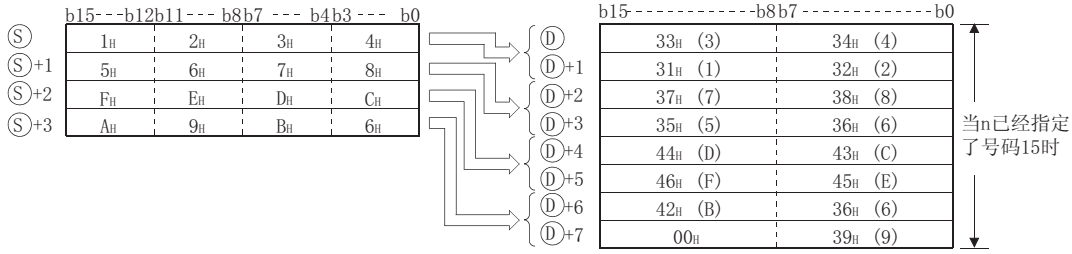
Ⓢ ：存储即将转换成字符串的 BIN 数据的软元件的起始软元件号 (BIN 16 位)
Ⓓ ：存储已转换的字符串的软元件的起始软元件号 (字符串)
n ：存储的字符数 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	—	○		—					—
Ⓓ	—	○		—					—
n	○	○		○					—

功 能

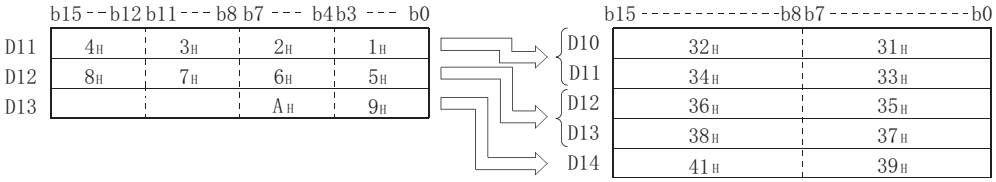
(1) 将存储在由 Ⓢ 指定号码的软元件之后的 BIN 16 位数据转换成十六进制 ASCII，并将在由 n 指定的字符数的范围内的结果存储在由 Ⓓ 指定号码的软元件之后。



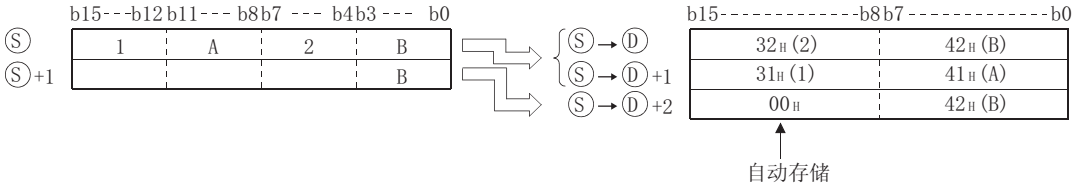


(2) 使用 n 设定字符数，使得由 Ⓢ 指定的 BIN 数据的范围和由 Ⓓ 指定的字符串存储软元件的范围被自动设定。

(3) 即使存储即将转换的 BIN 数据的软元件范围和存储转换后的 ASCII 数据的软元件范围有重叠，仍将准确地进行处理。



(4) 如果 n 已经指定了奇数个字符，则 ASCII 码 “00H” 将被自动存储在字符串存储范围内的最后一个软元件的头 8 位中。



(5) 如果 n 指定的字符数是 “0”，则将不进行转换处理。

运算错误

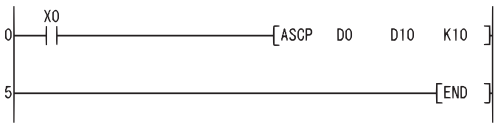
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- 在由 Ⓢ 指定号码的软元件之后，由 n 指定的字符数的范围超出了相关软元件的范围。
  （出错代码： 4101）
- 在由 Ⓓ 指定号码的软元件之后，由 n 指定的字符数的范围超出了相关软元件的范围。
  （出错代码： 4101）

程序示例

(1) 当 X0 变为 ON 时，以下程序将存储在 D0 上的 BIN 数据作为十六进制值进行读取，将其转换成字符串，并将结果存储在 D10 到 D14 上。

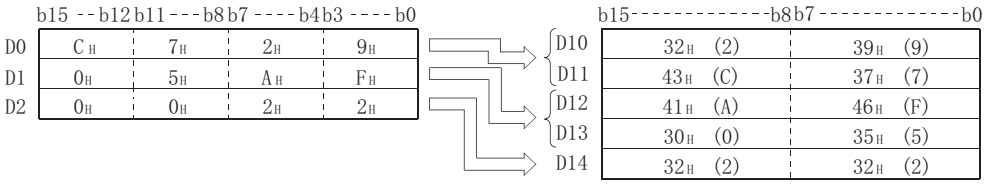
[梯形图模式]



[列表模式]

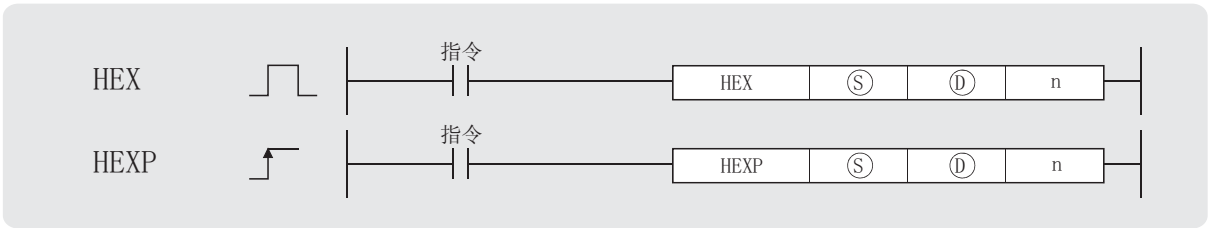
步	指令	软元件		
0	LD	X0		
1	ASCP	D0	D10	K10
5	END			

[动作]



7.11.14 从 ASCII 码到十六进制 BIN 的转换 (HEX(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

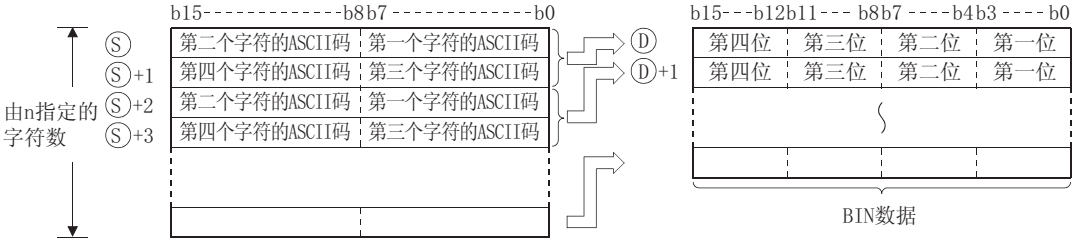


Ⓢ ：存储即将转换成 BIN 数据的字符串的软元件的起始软元件号（字符串）
Ⓓ ：存储已转换的 BIN 数据的软元件的起始软元件号 (BIN 16 位)
n ：转换的字符数 (BIN 16 位)

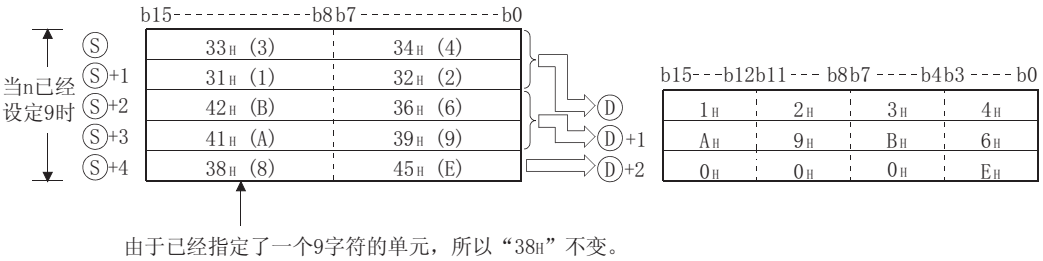
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	○		--					--
Ⓓ	--	○		--					--
n	○	○		○					--

功 能

(1) 将存储在由 Ⓢ 指定号码的软元件之后的 BIN 16 位数据转换成十六进制 ASCII，并将在由 n 指定的字符数的范围内的结果存储在由 Ⓓ 指定号码的软元件之后。

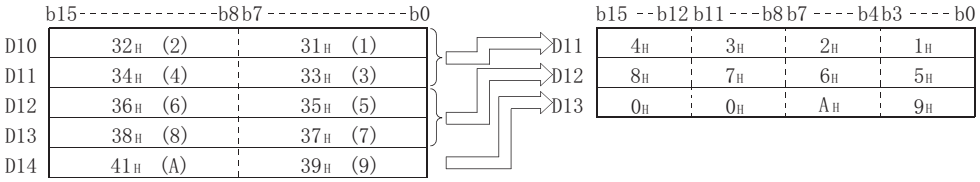


例如，如果 n 指定了数字 9，操作将如下所示：

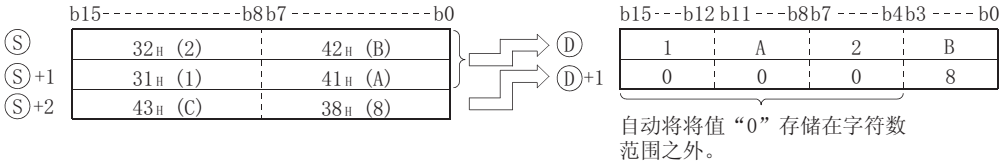


(2) 使用 n 设定字符数，使得由 Ⓢ 指定的字符串的范围和存储由 Ⓓ 指定的 BIN 数据的软元件的范围被自动设定。

(3) 即使存储即将转换的 ASCII 码的软元件范围和存储已转换的 BIN 数据的软元件范围有重叠，仍将进行准确处理。



(4) 如果 n 指定的字符数不能被 4 整除，则将自动将 “0” 存储在存储已转换 BIN 值的软元件的最后一个软元件号中指定的字符数之后。



(5) 如果 n 指定的字符数为 “0”，则将不进行转换处理。

(6) Ⓢ 可指定的 ASCII 码包括 “30H” 到 “39H” 和 “41H” 到 “46H”。

运算错误

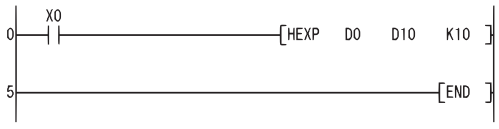
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- Ⓢ 设定了十六进制字符串之外的字符 (即这些字符不在 “30H” 到 “39H” 或 “41H” 到 “46H” 的范围内)。 (出错代码 : 4100)
- 在由 Ⓢ 指定号码的软元件之后，由 n 指定的字符数的范围超出了相关软元件范围。 (出错代码 : 4101)
- 在由 Ⓓ 指定号码的软元件之后，由 n 指定的字符数的范围超出了相关软元件范围。 (出错代码 : 4101)
- n 指定的字符数是一个负值。 (出错代码 : 4101)

程序示例

(1) 当 X0 变为 ON 时，以下程序将存储在 D0 到 D4 上的字符串转换成 BIN 数据，并将结果存储在 D10 到 D14 上。

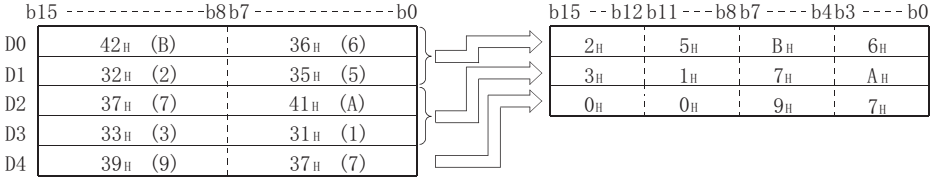
[梯形图模式]



[列表模式]

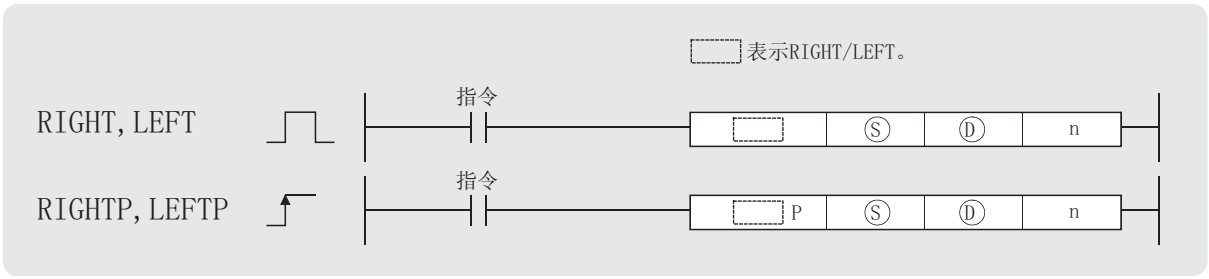
步	指令	软元件		
0	LD	X0		
1	HEXP	D0	D10	K10
5	END			

[动作]



7.11.15 从字符串的右边或左边提取数据 (RIGHT(P)、LEFT(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



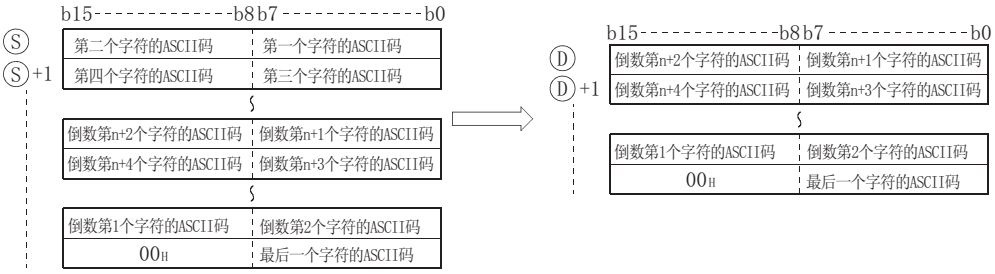
Ⓢ ：存储字符串的软元件的起始软元件号（字符串）
Ⓓ ：存储 Ⓢ 右边或左边的 n 个字符的字符串的软元件的起始软元件号（字符串）
n ：提取的字符数 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	
Ⓢ	—	○		—				—	○	—
Ⓓ	—	○		—				—	—	—
n	○	○		○				○	—	—

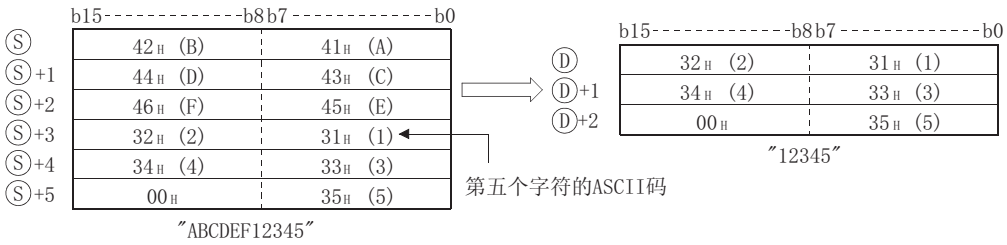
功 能

RIGHT

(1) 将存储在由 Ⓢ 指定号码的软元件之后的软元件上的字符串右侧（字符串的末尾）的 n 个字符，存储在由 Ⓓ 指定号码的软元件之后的软元件中。



当 n=5 时

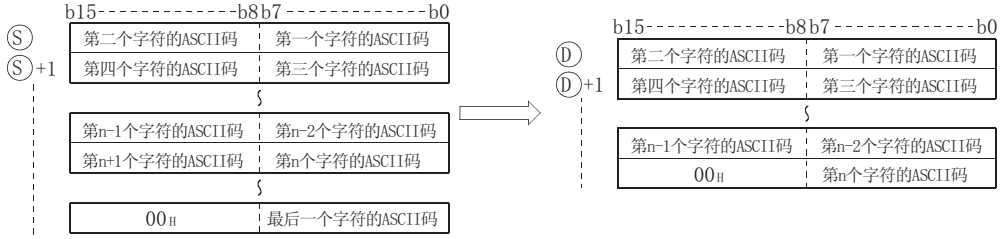


(2) 表示字符串尾的 NULL 码 (00H) 会自动添加到字符串尾。
关于字符串数据的格式，请参见 3.2.5 项。

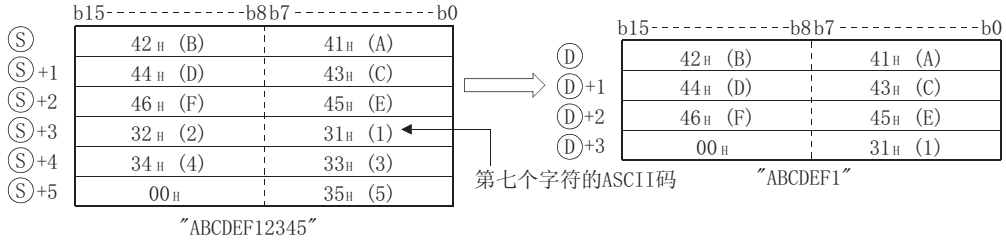
(3) 如果 n 指定的字符数为 “0”，则 NULL 代码 (00H) 将被存储在 ⓓ 上。

LEFT

(1) 将存储在由 ⓢ 指定号码的软元件之后的软元件上的字符串左侧（字符串的开始）的 n 个字符，存储在由 ⓓ 指定号码的软元件之后的软元件中。



当 n=7 时



(2) 表示字符串尾的 NULL 码 (00H) 会自动添加到字符串尾。
关于字符串数据的格式，请参见 3.2.5 项。

(3) 如果 n 指定的字符数为 “0”，则 NULL 代码 (00H) 将被存储起来。

运算错误

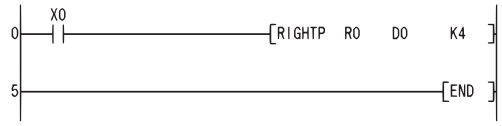
(1) 在以下发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 上。

- n 值超过了由 ⓢ 指定的字符数。 （出错代码： 4101）
- 从 ⓓ 开始的 n 个字符的范围超出了相关软元件的范围。 （出错代码： 4101）

程序示例

(1) 当 X0 变 ON 时，以下程序将 R0 上存储的字符串右侧的 4 个字符的数据存储在 D0 上。

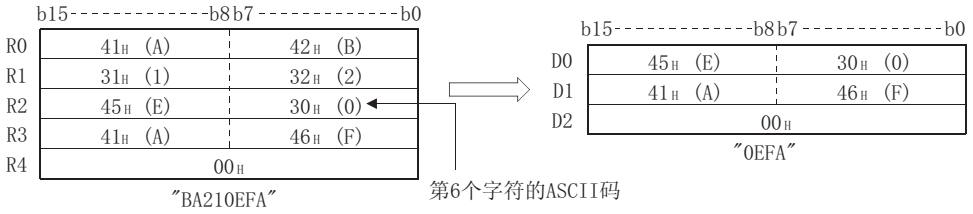
[梯形图模式]



[列表模式]

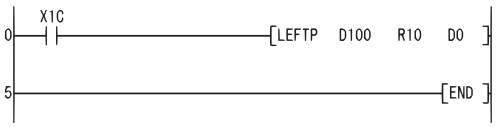
步	指令	软元件		
0	LD	X0		
1	RIGHTP	R0	D0	K4
5	END			

[动作]



(2) 当 X1C 变为 ON 时，以下程序将 D100 上存储的字符串数据左侧与 D0 中存储的值对应的字符数存储在 R10 之后。

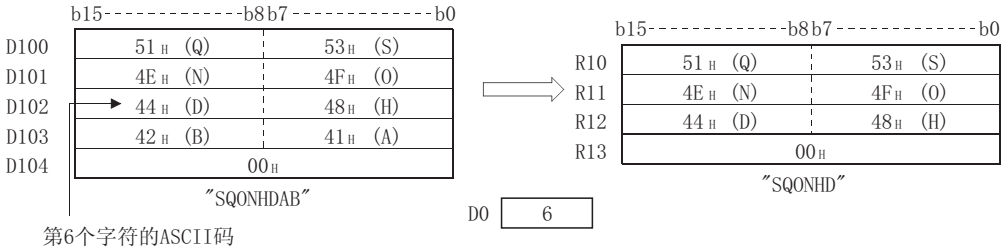
[梯形图模式]



[列表模式]

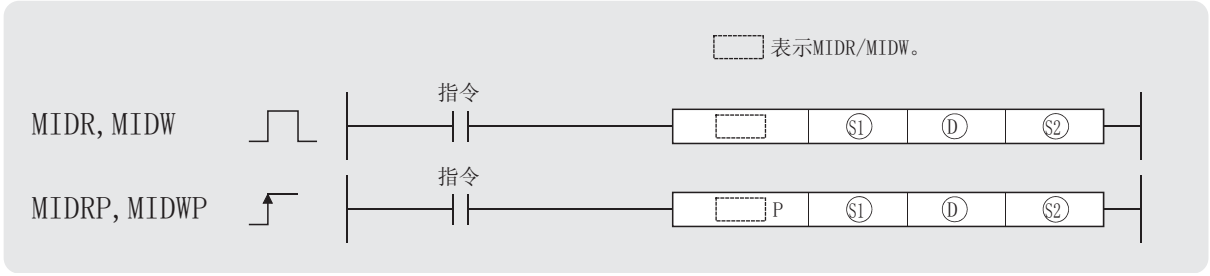
步	指令	软元件		
0	LD	X1C		
1	LEFTP	D100	R10	D0
5	END			

[动作]



7.11.16 字符串的任意选择和替换 (MIDR(P)、MIDW(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)：存储字符串的软元件的起始软元件号（字符串）

(D)：存储运算结果的字符串数据的软元件的起始软元件号（字符串）

(S2)：存储起始字符的位置和字符数的软元件的起始软元件号 (BIN 16 位)

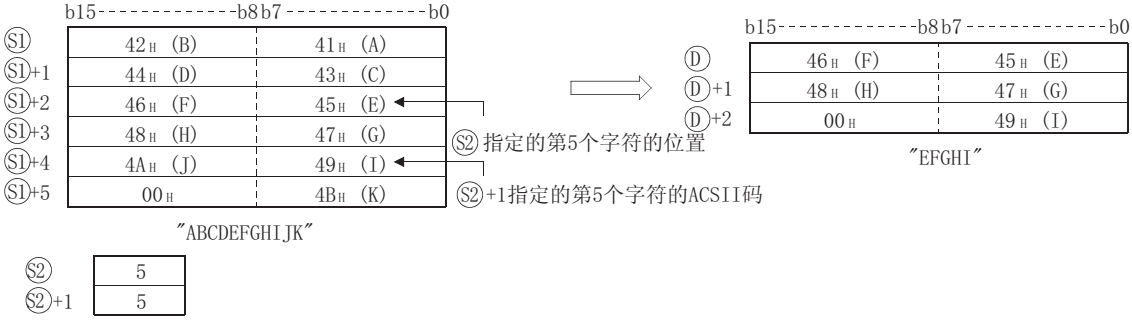
- (S2)　：起始字符的位
- (S2)+1：字符数

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
(S1)	—	○		—				○	—
(D)	—	○		—				—	—
(S2)	○	○		○				—	—

功 能

MIDR

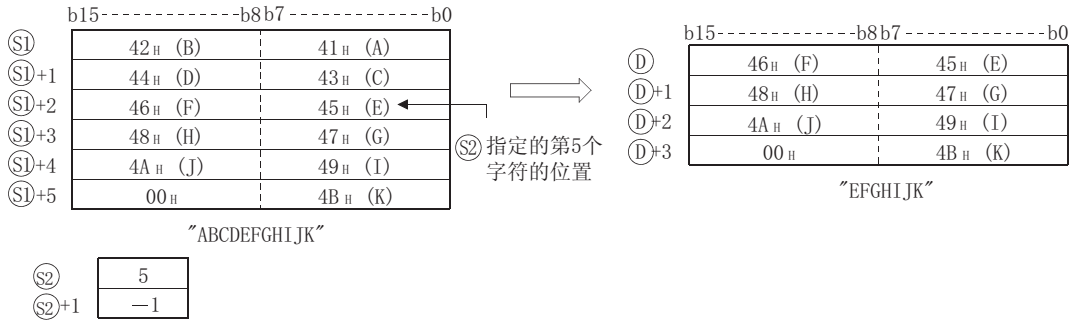
(1) 从由 (S1) 指定的软元件号开始，从正被存储的字符串的左侧，将从由 (S2) 指定的位置开始的由 (S2)+1 指定数目的字符，存储到从由 (D) 指定的软元件号开始的位置。



(2) 表示字符串尾的 NULL 码 (00H) 会自动添加到字符串尾。
关于字符串数据的格式，请参见 3.2.5 项。

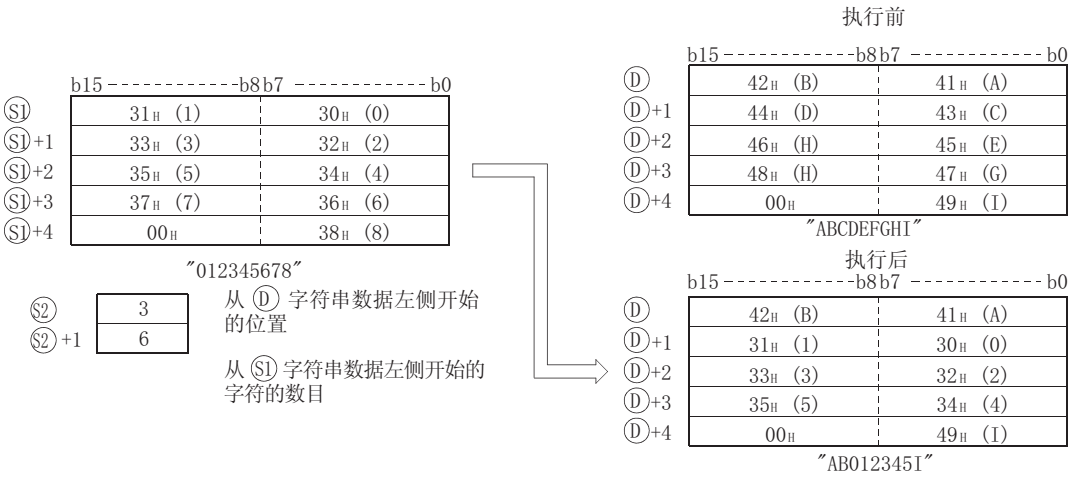
(3) 如果由 ⓢ2+1 指定的字符数是 “0”，那么不执行处理。

(4) 如果由 ⓢ2+1 指定的字符数为 “-1”，那么从由 Ⓓ 指定的软元件开始存储数据，直到碰到由 Ⓢ 指定的最后一个字符。



MIDW

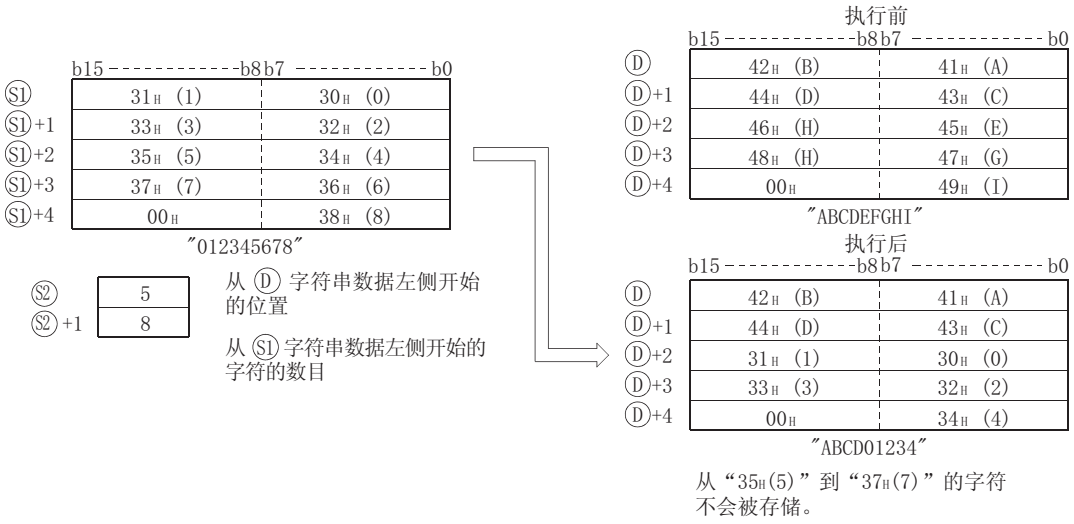
(1) 从 ⓢ1 指定的软元件号开始，从正被存储的字符串数据的左侧开始，将由 ⓢ2+1 指定数目的字符，存储到由 Ⓓ 指定的软元件号开始的，从正被存储的字符串数据的左侧，由 ⓢ2 指定的位置开始的位置中去。



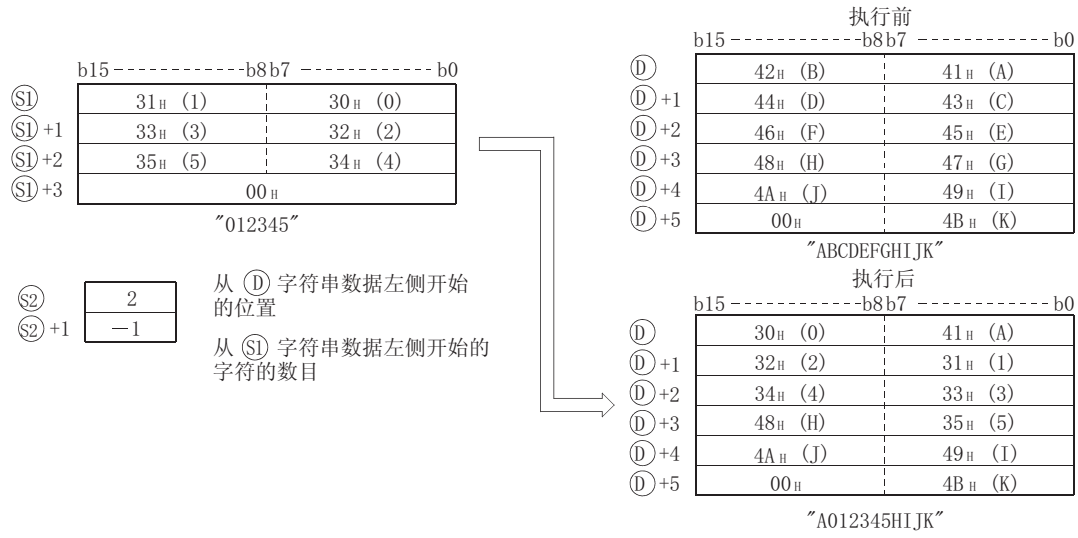
(2) 表示字符串尾的 NULL 码 (00H) 会自动添加到字符串尾。
关于字符串数据的格式，请参见 3.2.5 项。

(3) 如果由 ⓢ2+1 指定的字符数是 0，那么将不执行处理。

(4) 如果由 ⓢ2+1 指定的字符数超出了从由 Ⓓ指定的字符串数据开始的最后一个字符，那么存储数据，只到碰到最后一个字符。



(5) 如果由 ⓢ2+1 指定的字符数为 “-1”，那么从由 Ⓓ 指定的软元件开始存储字符，直到碰到由 ⓢ1 指定的最后一个字符数据。



运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

对于 MIDR 指令

- ⓢ2 的值超出了由 ⓢ1 指定的字符数。 (出错代码： 4101)
- 从位置 Ⓓ 开始的 ⓢ2+1 个字符数超出了 Ⓓ 软元件的范围。 (出错代码： 4101)
- ⓢ2+0 的值为 0。 (出错代码： 4101)
- ⓢ1 指定的软元件号以后的相应软元件范围中不存在 “00H”。 (出错代码： 4101)

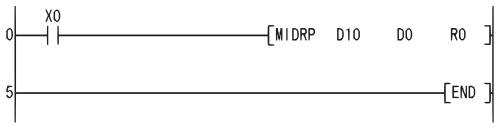
对于 MIDW 指令

- ⓢ2 的值超出了 Ⓓ 的字符数。 (出错代码： 4101)
- ⓢ2+1 的值超出了 ⓢ1 的字符数。 (出错代码： 4101)
- ⓢ2+ 的值为 0。 (出错代码： 4101)
- ⓢ1 指定的软元件号以后的相应软元件范围中不存在 “00H”。 (出错代码： 4101)

程序示例

(1) 当 X0 为 ON 时，下列程序在从 D10 开始存储的字符串中，从左边起第 6 个字符处开始，将第 3 个字符存储到从 D0 开始的软元件中。

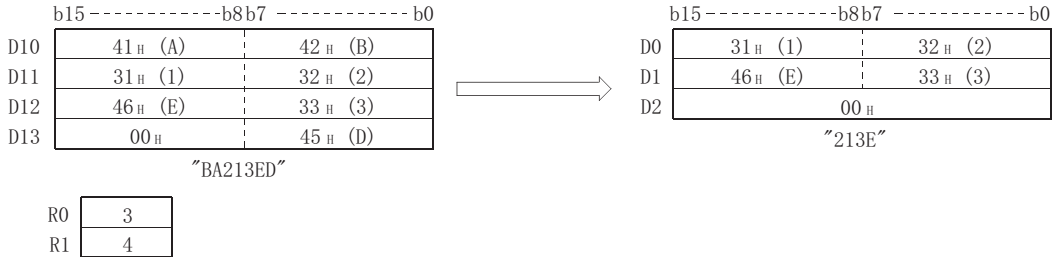
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	MIDRP	D10	D0	R0
5	END			

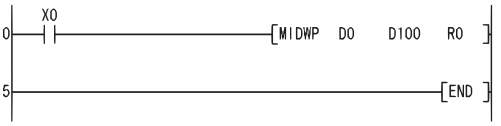
[动作]



R0	3
R1	4

(2) 当 X1C 为 ON 时，下列程序将存储在从 D0 开始的软元件中的字符串数据的 4 个字符，存储到，从 D100 开始的软元件中的字符串数据，左起第 3 个字符开始的位置中。

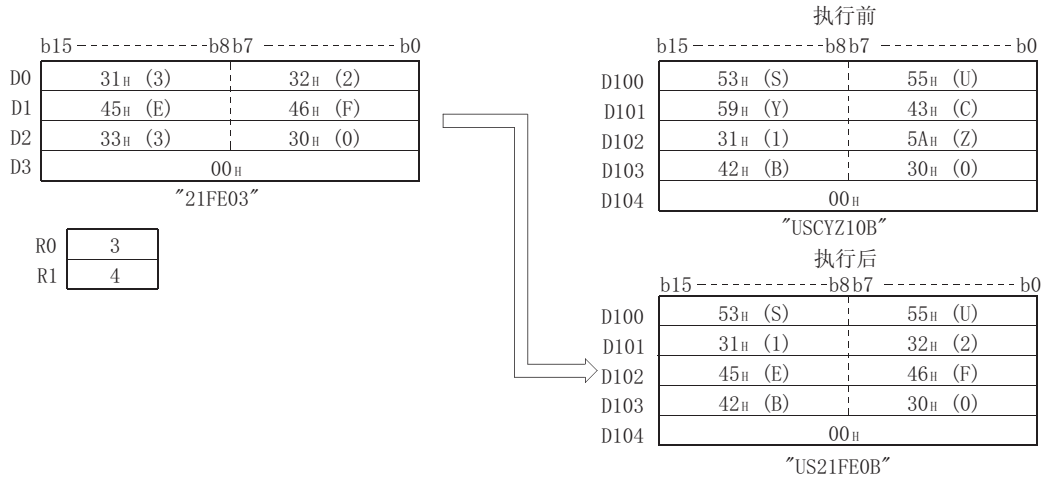
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	MIDWP	D0	D100	R0
5	END			

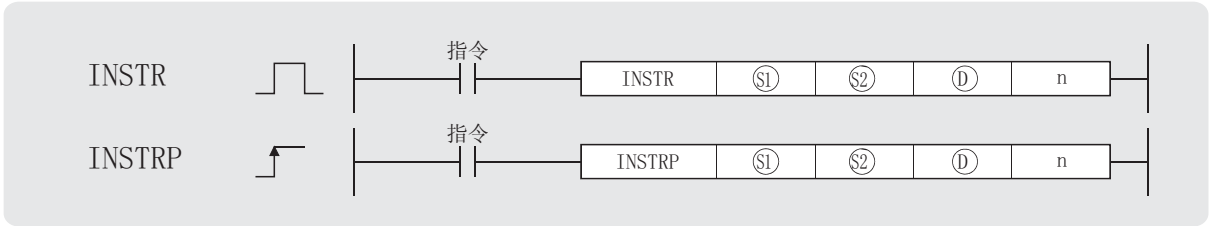
[动作]



R0	3
R1	4

7.11.17 字符串查找 (INSTR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



ⓢ1 ：要查找的字符串或者存储要查找的字符串的软元件的起始号（字符串）
ⓢ2 ：被查找的字符串或者存储被查找字符串的软元件的起始号（字符串）
Ⓓ ：存储查找结果的软元件的起始软元件号 (BIN 16 位)
n ：开始查找的位置 (BIN 16 位)

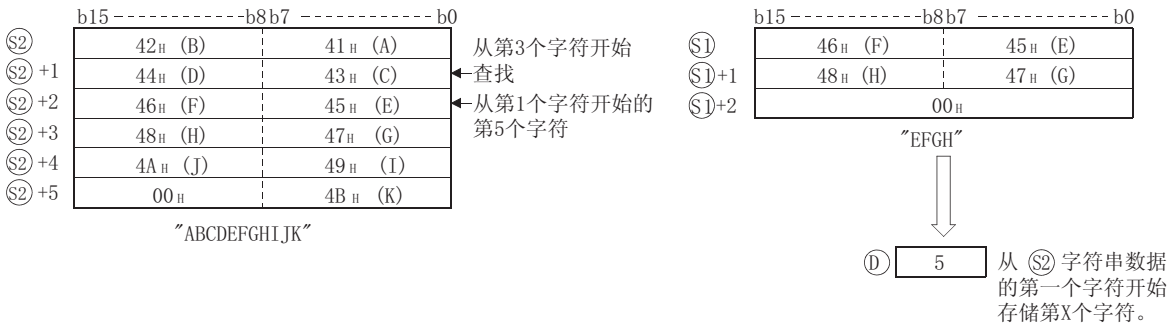
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	
ⓢ1	--	○		--				--	○	--
ⓢ2	--	○		--				--	○	--
Ⓓ	○	○		○				--	--	--
n	○	○		○				○	--	--

功 能

(1) 在由 ⓢ2 指定号码的软元件开始的软元件中存储的字符串数据中，从左起第 n 个字符串数据开始，查找由 ⓢ1 指定号码的软元件开始的软元件中存储的字符串数据，并且在由 Ⓓ 指定的软元件中存储查找结果。

查找结果存储在从由 ⓢ2 指定的字符串数据的起始字符开始的第 X 个字符中。

When n = 3



(2) 如果没有匹配的字符串数据，在 Ⓓ 中存储 “0”。

运算错误

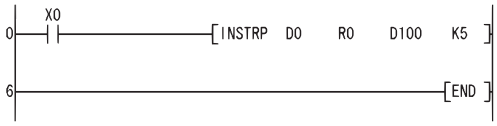
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- n 的值超出了 ⑤2 的字符数。 (出错代码： 4100)
- 在由 ⑤1、⑤2 指定的软元件之后的相应的软元件范围中不存在 00H(NULL)。 (出错代码： 4100)
- n 的值为负数或 “0”。. (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序从 R0 开始的软元件中存储的字符串数据，左起第 5 个字符处开始查找从 D0 开始的软元件中的字符串数据，并且在 D100 中存储结果。

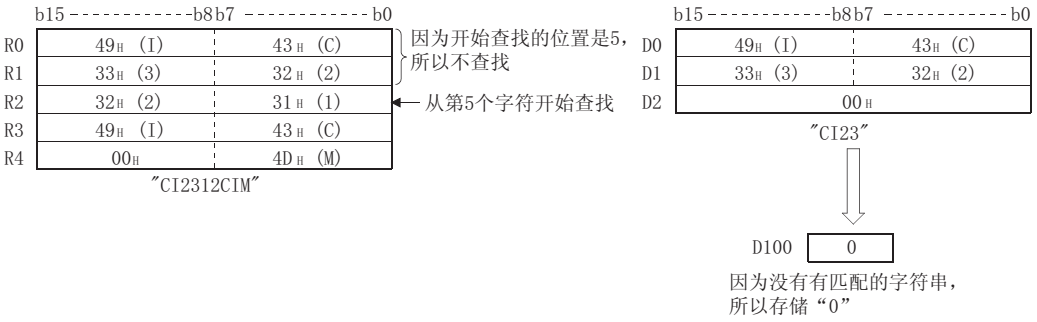
[梯形图模式]



[列表模式]

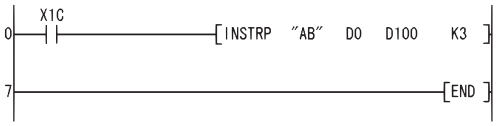
步	指令	软元件			
0	LD	X0			
1	INSTRP	D0	R0	D100	K5
6	END				

[动作]



(2) 当 X1C 为 ON 时，下列程序从 D0 开始的软元件中存储的字符串数据，左起第 3 个字符处开始查找字符数据 “AB”，并且在 D100 中存储查找结果。

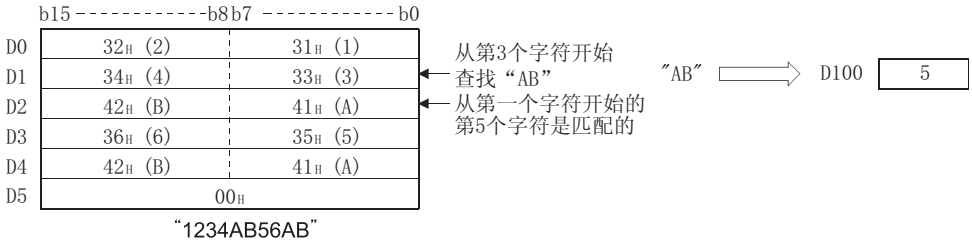
[梯形图模式]



[列表模式]

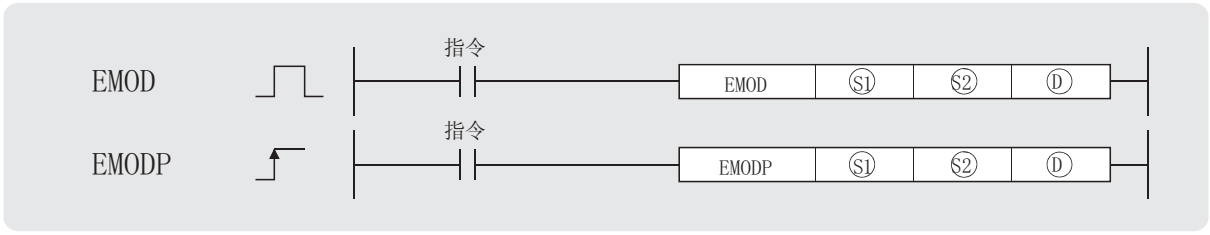
步	指令	软元件			
0	LD	X1C			
1	INSTRP	"AB"	D0	D100	K3
7	END				

[动作]



7.11.18 浮点数到 BCD 的分解 (EMOD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

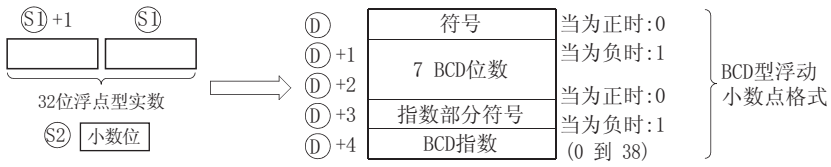


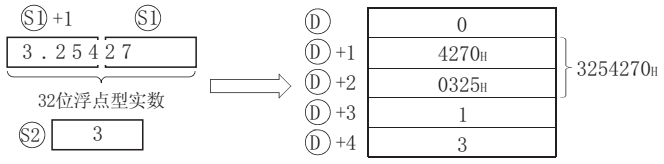
S1：32 位浮点型实数数据或存储浮点型实数数据的软元件的起始号（实数）
S2：小数部分位数据或存储该类数据的软元件的起始软元件号 (BIN 16 位)
D：将要存储 BCD 分析数据的软元件的起始软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	E	
S1	—	○		—	○		—	—	○	—
S2	○	○		○	○		○	○	—	—
D	—	○		—	—		—	—	—	—

功 能

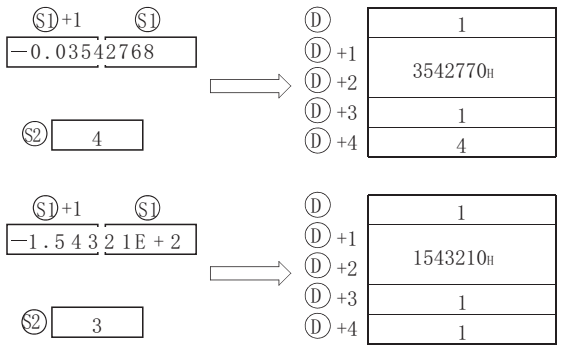
(1) 将在由 S1 指定的软元件中存储的浮点类型实数，按照由 S2 指定的软元件中存储的小数部分位数，分解为 BCD 浮点格式，并且从由 D 指定的软元件开始存储结果。



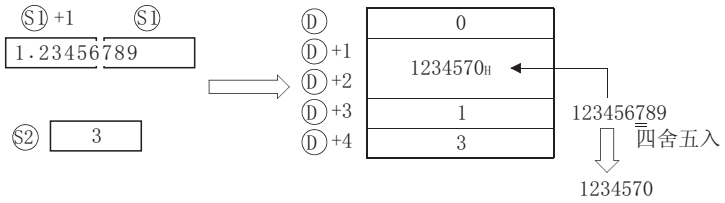


项目 S2 指定了项目 S1 的浮动小数点类型实数的小数部分的位数，在上述例子中，如下所示：

3.25427
S2 =3



(2) 在 (D)+1 和 (D)+2 中存储的有效数字的第 7 位数被四舍五入，从而变为一个 6 位数。



运算错误

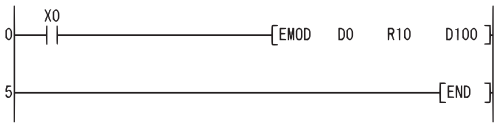
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 (S2) 指定的小数部分位数在 0 到 7 的范围之外。 (出错代码：4100)
- 由 (D) 指定的软元件范围超出了相关软元件的范围。 (出错代码：4101)
- 由 (S1) 指定的 32 位浮点型实数超出了以下范围。
  0、$2^{-126} \leqq |$ 软元件 $| < 2^{128}$ (出错代码：4100)
- 由 (D) 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码：4101)
- 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$ 。
  (只对于通用型 QCPU。) (出错代码：4140)

程序示例

(1) 当 X0 为 ON 时，下列程序将在 D0 和 D1 中存储的浮点类型实数数据，按照在 R10 中存储的值对应的小数部分的位数进行分解，并且将结果存储到从 D100 开始的软元件中。

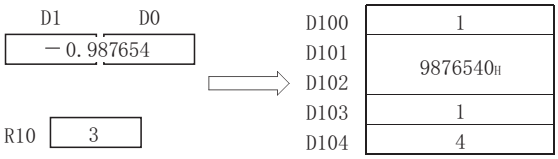
[梯形图模式]



[列表模式]

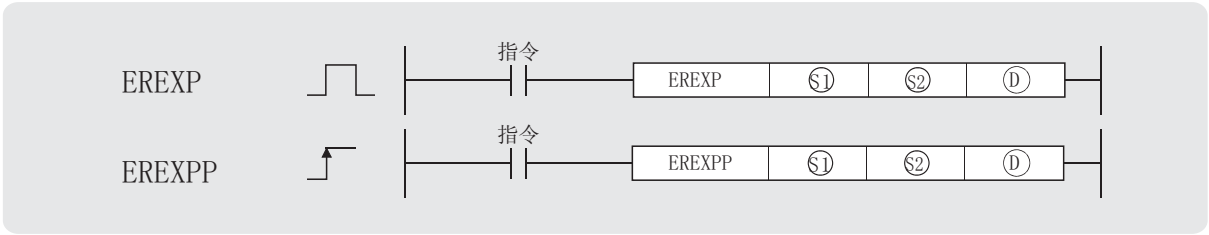
步	指令	软元件		
0	LD	X0		
1	EMOD	D0	R10	D100
5	END			

[动作]



7.11.19 从 BCD 格式数据到浮点数 (EREXP(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1)：存储 BCD 浮点格式数据的软元件的起始软元件号 (BIN 16 位)

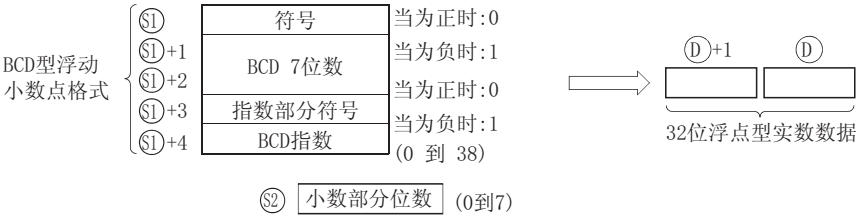
(S2)：小数部分位数据或存储该类数据的软元件 (BIN 16 位)

(D)：存储 32 位浮点型实数数据的软元件 (实数)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)	—	○		—		—	—	—	—
(S2)	○	○		○		○	○	○	—
(D)	—	○		—		○	—	—	—

功 能

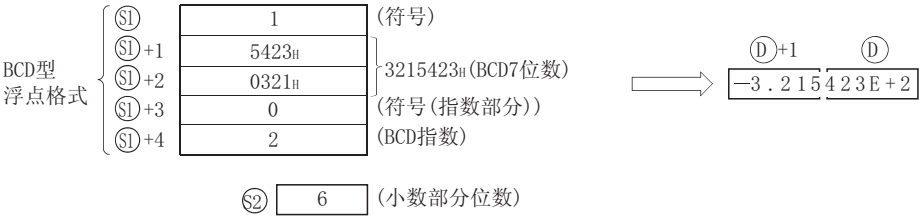
(1) 将从软元件号由 (S1) 指定的软元件开始的软元件中存储的 BCD 类型浮点格式数据，按照由 (S2) 指定的软元件中存储的小数部分位数，转换成浮点类型实数，并且在从由 (D) 指定软元件号的软元件开始的软元件中存储结果。



(2) 如果是正值，(S1) 的符号和 (S1)+3 的指数部分的符号设置为 0，如果是负值，设置为 1。

(3) 对于 (S1)+4 个 BCD 指数，可以设置为 0 到 38 的值。

(4) 小数部分位数可以设置为 0 到 7。



运算错误

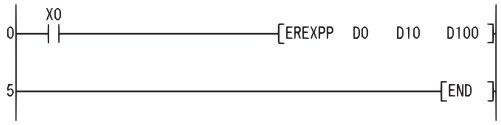
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- ⑤1 指定的格式指定既不是 0 也不是 1。 (出错代码： 4100)
- ⑤1+1 和 ⑤1+2 的各位数中存在 0 到 9 以外的值。 (出错代码： 4100)
- ⑤1+3 指定的格式指定既不是 0 也不是 1。 (出错代码： 4100)
- ⑤1+4 指定的指数数据在 0 到 38 的范围之外。 (出错代码： 4100)
- ⑤2 指定的小数部分位数在 0 到 7 的范围之外。 (出错代码： 4100)
- ⑤1 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 当 X0 为 ON 时，按照在 D10 中存储的小数部分位数，下列程序从将 D0 开始的软元件中存储的 BCD 浮点格式数据，转换为浮点类型实数数据，并且在 D100 和 D101 中存储结果。

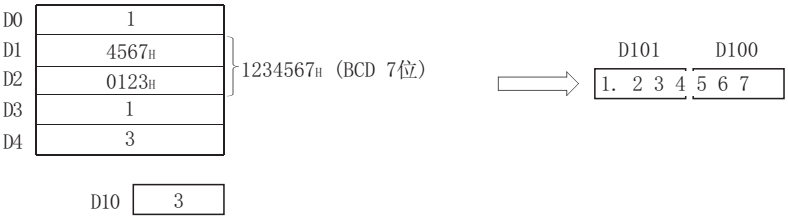
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	EREXPP	D0	D10	D100
5	END			

[动作]

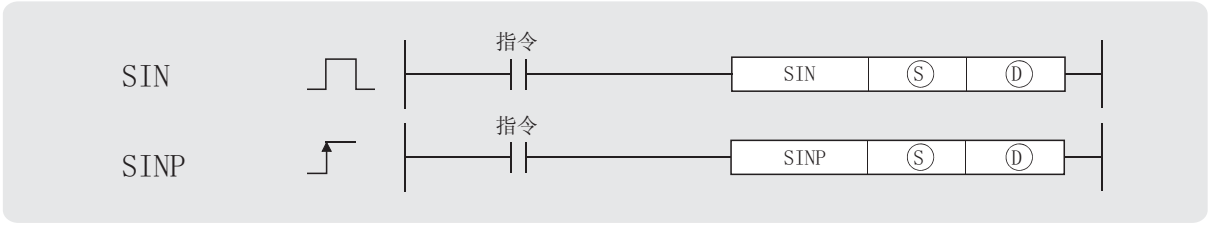


7.12 特殊函数指令

7.12.1 浮点数的 SIN 运算（单精度）(SIN(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ ：存储将执行 SIN（正弦）运算的角度数据的软元件的起始软元件号（实数）

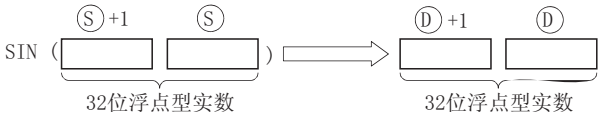
Ⓓ ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只可用于通用型 QCPU。

功 能

(1) 计算由 Ⓢ 指定的角度的 SIN（正弦）值，并且在 Ⓓ 指定软元件号的软元件中存储运算结果。



(2) 由 Ⓢ 指定的角是以弧度单位来设置的（度数 × π / 180）.
关于角度和弧度值之间的转换，请参见 RAD 和 DEG 指令。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 当指定的软元件的值为 -0 时。*3
  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  (出错代码： 4100)

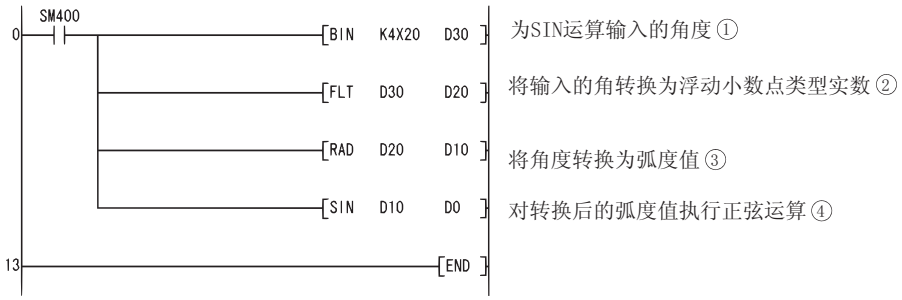
*3: 有的 CPU 模块即使指定了 -0 也不会发出运算出错信息。详细内容请参考 3.2.4 项。

- 运算结果超出了以下范围。(运算结果溢出)
  (只对于通用型 QCPU)
  $2^{128} \leqq |$ 运算结果 $|$ (出错代码： 4141)
- 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$。
  (只对于通用型 QCPU。) (出错代码： 4140)

程序示例

(1) 下列程序对存储在 X20 到 X2F 中的以 BCD4 位设置的角度执行 SIN 运算，并将结果以 32 位浮点型实数存储到 D0 到 D1 中。

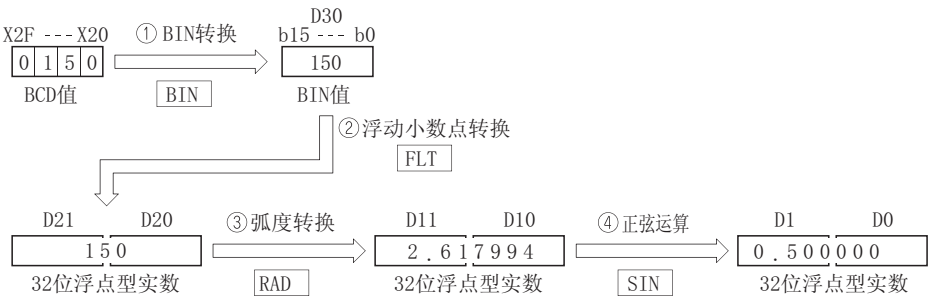
[梯形图模式]



[列表模式]

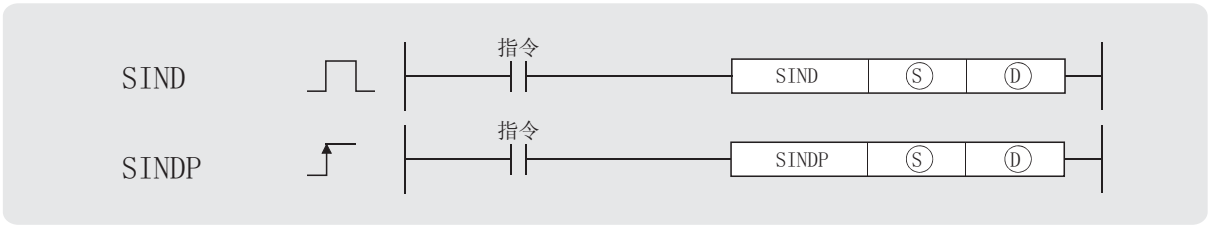
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	SIN	D10	D0
13	END		

[当 X20 到 X2F 指定的值为 150 时相关的运算]



7.12.2 浮点数的 SIN 运算（双精度）(SIND(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



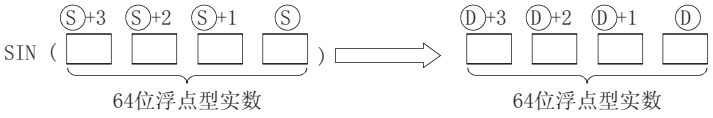
Ⓢ ：存储将执行 SIN（正弦）运算的角度数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 计算由 Ⓢ 指定的角度的 SIN（正弦）值，并且在 Ⓓ 指定的软元件中存储运算结果。



(2) 由 Ⓢ 指定的角度是以弧度单位（角度 × π / 180）设置的。
关于角度和弧度值之间的转换，请参见 RADD 和 DEGD 指令。

(3) 当运算结果为 -0 或下溢时，结果将被作为 0 处理。

运算错误

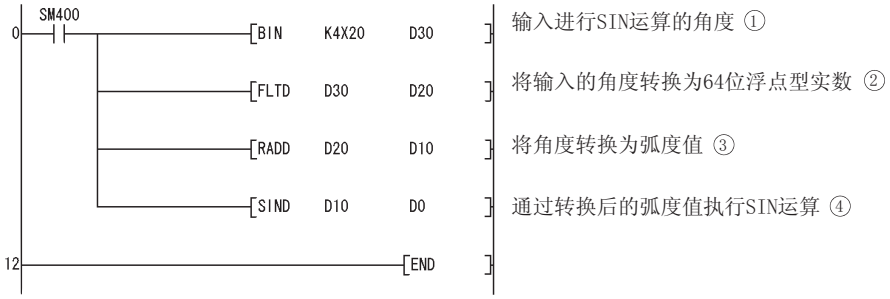
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中：（出错代码：4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。（出错代码：4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |$ 运算结果 $|$（出错代码：4141）

程序示例

(1) 下列程序对存储在 X20 到 X2F 中的以 BCD4 位设置的角度执行 SIN 运算，并将结果以 64 位浮点型实数的形式存储到 D0 到 D3 中。

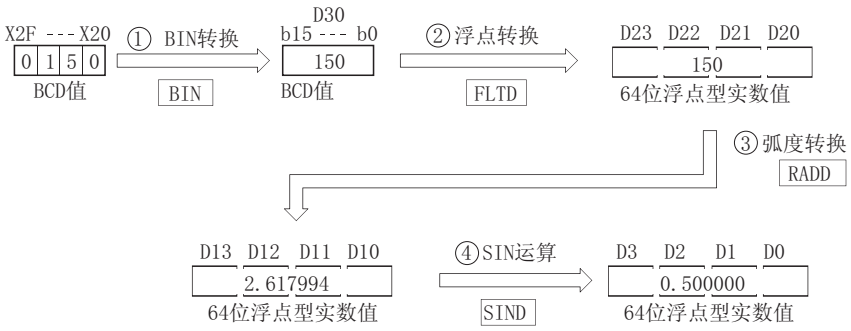
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	SIND	D10	D0
12	END		

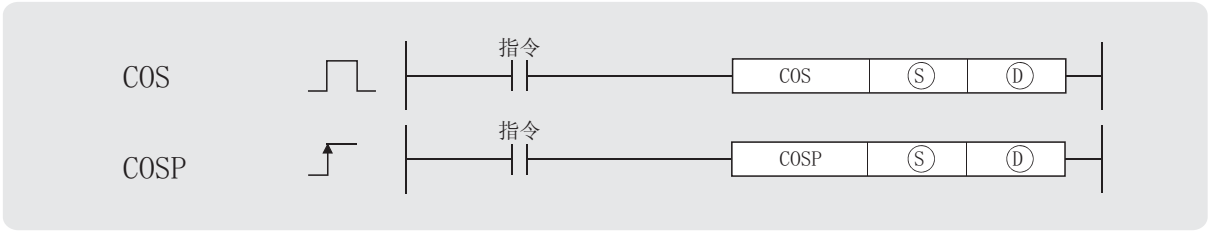
[当 X20 到 X2F 指定的值为 150 时的动作]



7.12.3 浮点数的 COS 运算（单精度）(COS(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：存储执行 COS（余弦）运算的角度数据的软元件的起始软元件号（实数）

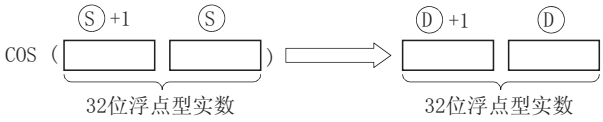
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 对由 Ⓢ 指定的角度执行 COS（余弦）运算，并且在由 Ⓓ 指定软元件号的软元件中存储运算结果。



(2) 由 Ⓢ 指定的角度是以弧度单位来设置的（角度 × π / 180）。
关于角度和弧度值之间的转换，请参见 RAD 和 DEG 指令。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 当指定的软元件的值为 -0 时。 *3
  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  (出错代码： 4100)

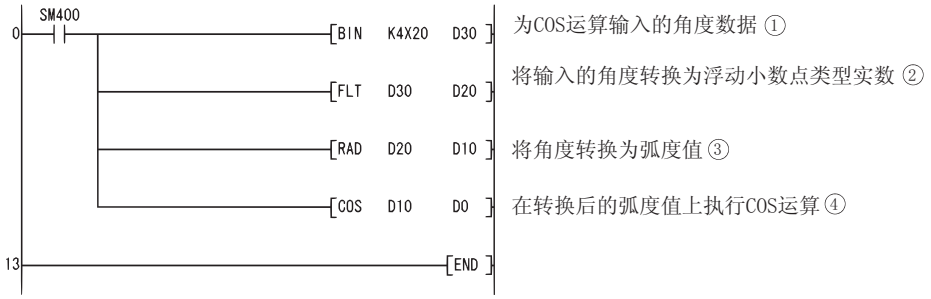
*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围。(运算结果上溢)
  (只对于通用型 QCPU)
  $2^{128} \leqq |$ 运算结果 $|$ (出错代码： 4141)
- 指定软元件的值为 -0、非正规数、非数字和 $\pm\infty$。
  (只对于通用型 QCPU。) (出错代码： 4140)

程序示例

(1) 下列程序对存储在 X20 到 X2F 的 4 个 BCD 位数指定的角度数据执行 COS 运算，并且以浮点类型实数在 D0 和 D1 中存储结果。

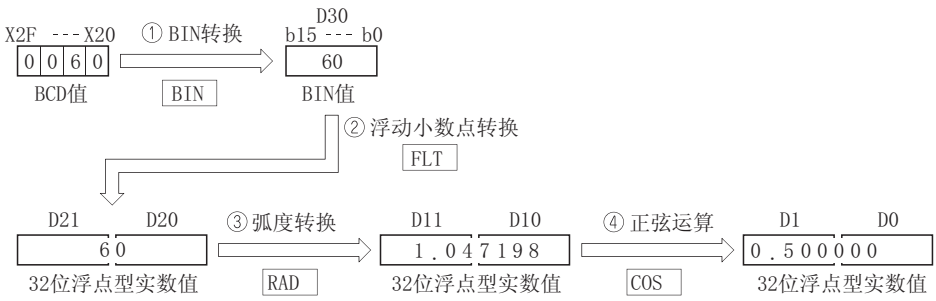
[梯形图模式]



[列表模式]

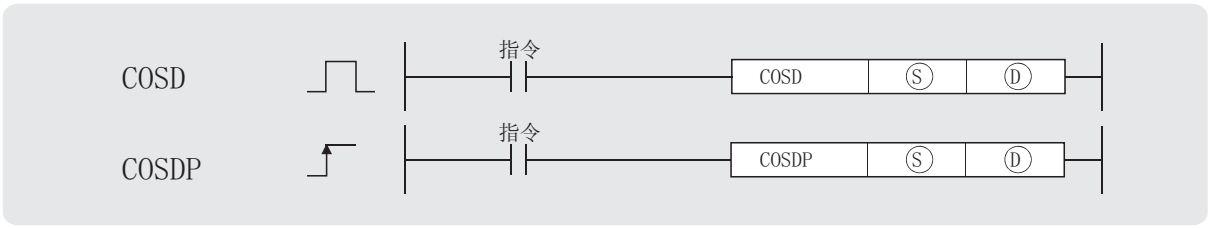
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	COS	D10	D0
13	END		

[当 X20 到 X2F 指定的值为 60 时相关的运算]



7.12.4 浮点数的 COS 运算（双精度）(COSD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



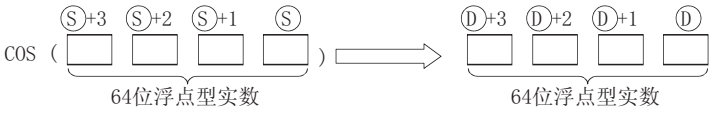
Ⓢ ：进行 COS（余弦）运算的角度数据或存储角度数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 对由 Ⓢ 指定的角度执行 COS（余弦）运算，并且在由 Ⓓ 指定的软元件中存储运算结果。



(2) 由 Ⓢ 指定的角度是以弧度单位（角度 × π / 180）设置的。
关于角度和弧度值之间的转换，请参见 RADD 和 DEGD 指令。

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

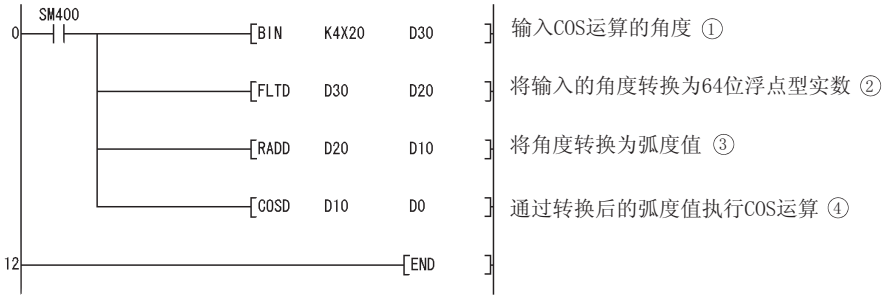
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中：（出错代码：4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。（出错代码：4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |$ 运算结果 $|$（出错代码：4141）

程序示例

(1) 下列程序对存储在 X20 到 X2F 的以 BCD4 位设定的角度数据执行 COS 运算，并且以 64 位浮点型实数的形式存储到 D0 ～ D3 中。

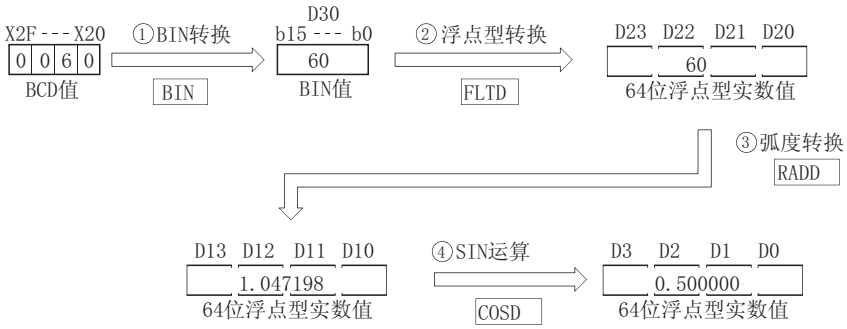
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	COSD	D10	D0
12	END		

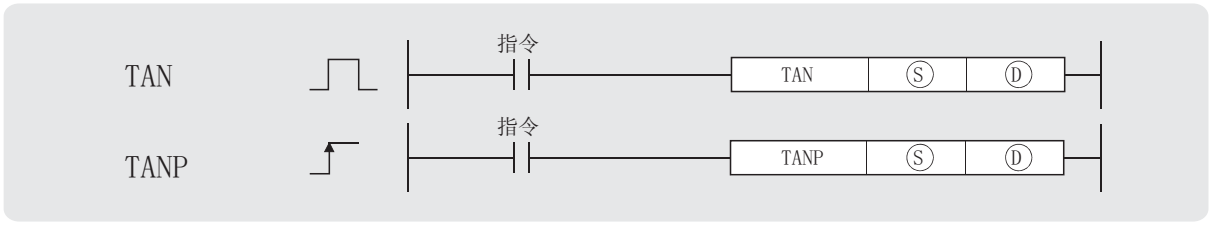
[当 X20 到 X2F 中指定的值为 60 时的动作]



7.12.5 浮点数的 TAN 运算（单精度）(TAN(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：存储将要执行 TAN（正切）运算的角度数据的软元件的起始软元件号（实数）

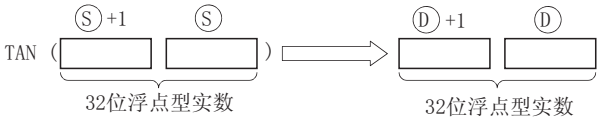
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 对由 Ⓢ 指定的角度数据执行正切 (TAN) 运算，并且在由 Ⓓ 指定的软元件中存储运算结果。



(2) Ⓢ 指定的角度是以弧度单位来设置的（角度 × π / 180）。
关于角度和弧度值之间的转换，请参见 RAD 和 DEG 指令。

(3) 当由 Ⓢ 指定的角度是 π/2 弧度，或 (3/2)π 弧度时，在弧度值的计算当中将会产生一个运行错误，所以应该小心避免此类出错。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 运算结果在下列范围之外：
  0、$2^{-126} \leqq$ ｜运算结果｜ $< 2^{128}$
  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU）
  （出错代码： 4100）
- 当指定的软元件的值为 -0 时。 *3
  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU）
  （出错代码： 4100）

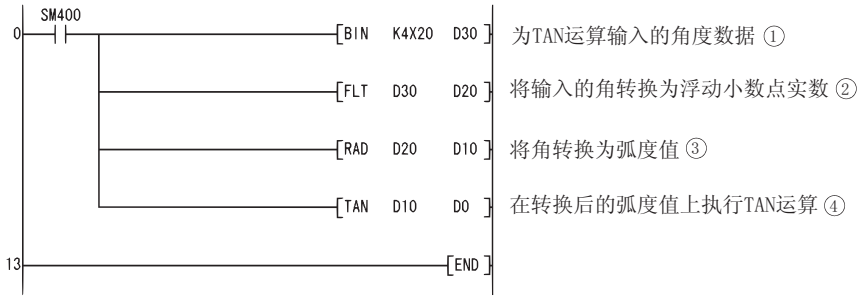
*3: 有的 CPU 墨客即使指定的是 -0，也不会发生运算错误。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围。（运算结果溢出）
  （只对于通用型 QCPU）
  $2^{128} \leqq$ ｜ 运算结果 ｜ （出错代码： 4141）
- 指定软元件的值为 -0、非正规数、非数字和±∞。
  （只对于通用型 QCPU。） （出错代码： 4140）

程序示例

(1) 下列程序对由从 X20 到 X2F 的 4 个 BIN 位数设定的角度数据执行 TAN 运算，并且在 D0 和 D1 中存储结果为浮点类型实数。

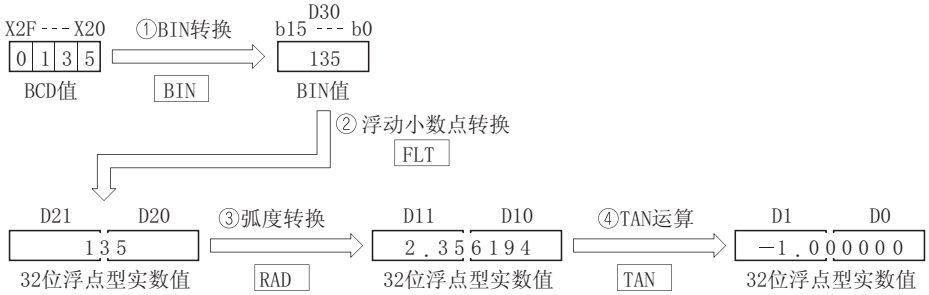
[梯形图模式]



[列表模式]

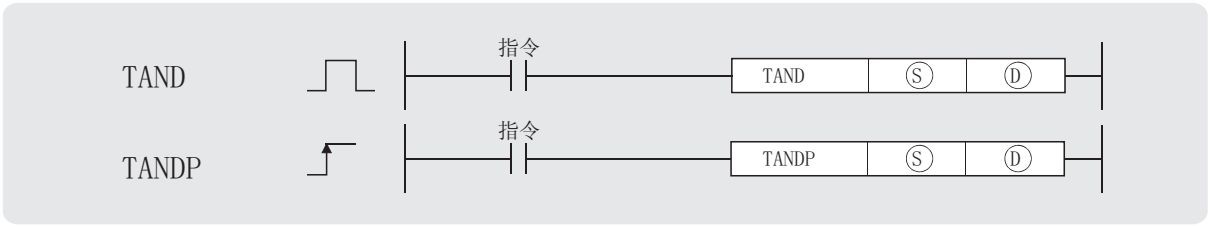
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
4	FLT	D30	D20
7	RAD	D20	D10
10	TAN	D10	D0
13	END		

[当 X20 到 X2F 的值指定为 135 时的相关运算]



7.12.6 浮点数的 TAN 运算（双精度）(TAND(P))

通用型 UD　基本型　高性能型　过程　冗余　冗余　Q4AR



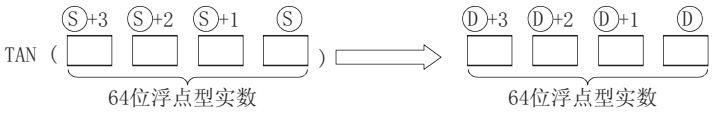
Ⓢ ：进行 TAN（正切）运算的角度数据或存储角度数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 对由 Ⓢ 指定的角度数据执行 TAN（正切）运算，并且在由 Ⓓ 指定的软元件中存储运算结果。



(2) Ⓢ 指定的角度是以弧度单位（角度 × π / 180）设置的。
关于角度和弧度之间的转换，请参见 RADD 和 DEGD 指令。

(3) 当由 Ⓢ 指定的角度是 π/2 弧度，或 (3/2)π 弧度时，在弧度值的计算当中将会产生运算误差，且不会发出出错信息，所以应加以注意。

(4) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

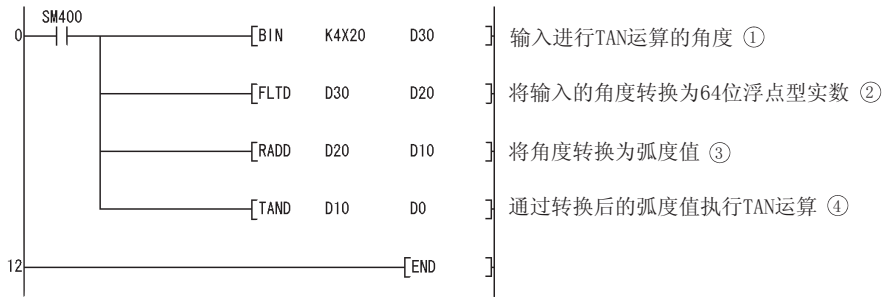
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： （出错代码： 4140）
  0、2^{-1022} ≦ ｜指定的软元件值｜< 2^{1024}
- 指定的软元件的值为 -0。 （出错代码： 4140）
- 运算结果超出了以下范围（运算结果溢出）:
  2^{1024} ≦ ｜运算结果｜ （出错代码： 4141）

程序示例

(1) 下列程序对由从 X20 到 X2F 的以 BCD4 位设定的角度数据执行 TAN 运算，并以 64 位浮点型实数存储到 D0 到 D3 中。

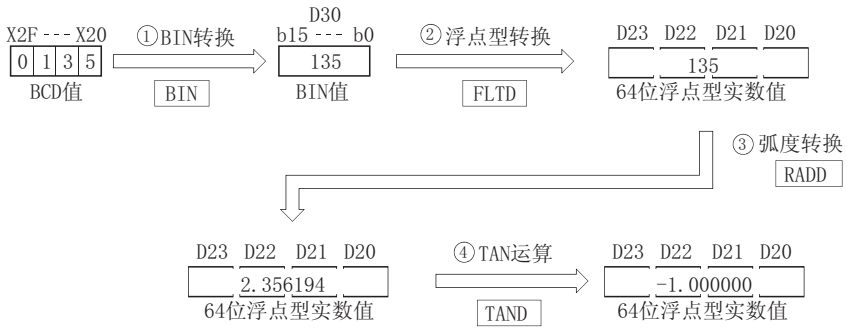
[梯形图模式]



[列表模式]

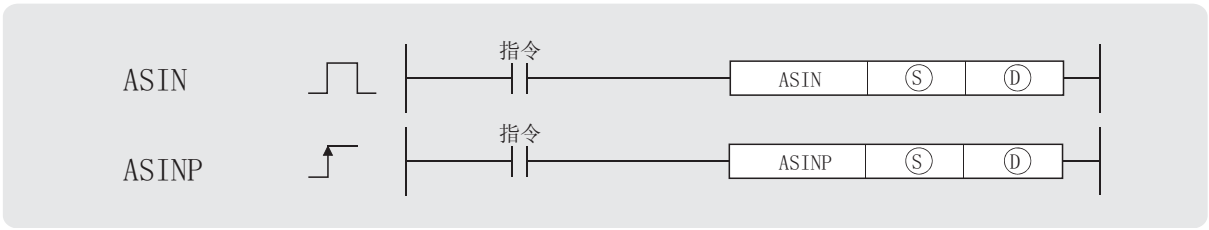
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D30
3	FLTD	D30	D20
6	RADD	D20	D10
9	TAND	D10	D0
12	END		

[当 X20 到 X2F 中指定的值为 135 时的动作]



7.12.7 浮点数的 SIN^{-1} 运算（单精度）(ASIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：存储将要执行 SIN^{-1}（反正弦）运算的 SIN 值的软元件的起始软元件号（实数）

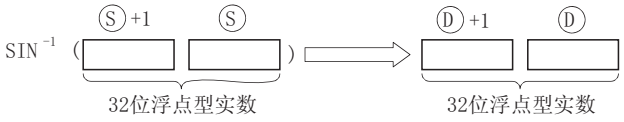
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*1	○	---
Ⓓ	---	○		---	○		○*1	---	---

*1：只适用于通用型 QCPU。

功 能

(1) 从由 Ⓢ 指定的 SIN 值计算角度，并且在由 Ⓓ 指定的字软元件中存储运算结果。



(2) 由 Ⓢ 指定的 SIN 值可在从 -1.0 到 1.0 的范围之内。

(3) 存储在 Ⓓ 中的角度（运算结果）是以弧度单位来存储的。
关于角度和弧度值之间的转换，请参见 RAD 和 DEG 指令。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 (S) 指定的值在 -1.0 到 1.0 的范围之外 （出错代码： 4100）
- 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内（只对于通用型 QCPU)：
  （出错代码： 4140）

  0、$2^{-126} \leqq |$ 指定软元件的内容 $| < 2^{128}$

- 指定的软元件值为 -0。 *2
  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  （出错代码： 4100）

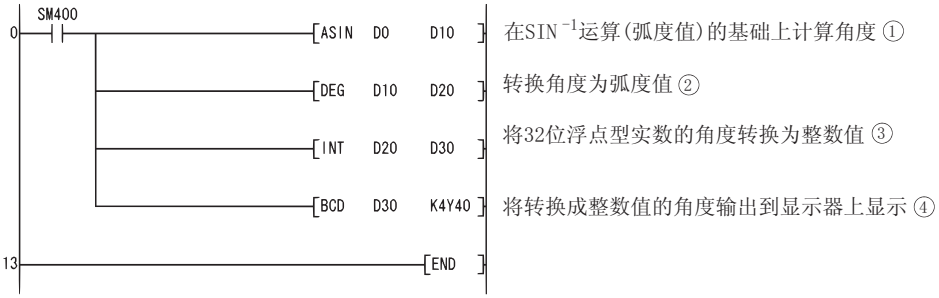
*2: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围（运算结果溢出）
  （只对于通用型 QCPU)
  $2^{128} \leqq |$ 运算结果 $|$ （出错代码： 4141）
- 指定软元件的值为 -0、非正规数、非数和±∞。
  （只对于通用型 QCPU) （出错代码： 4140）

程序示例

(1) 下列程序求出 D0 和 D1 中的 32 位浮点型实数的 SIN^{-1}(反正弦)，并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

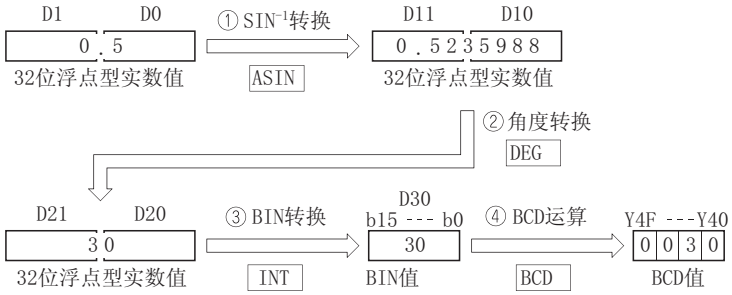
[梯形图模式]



[列表模式]

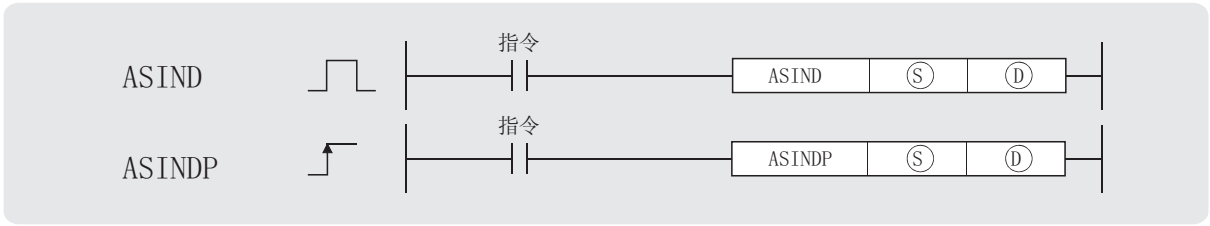
步	指令	软元件	
0	LD	SM400	
1	ASIN	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[当 D0 和 D1 的值是 0.5 时的相关运算]



7.12.8 浮点数的 SIN^{-1} 运算（双精度）(ASIND(P))

通用型 UD　基本型　高性能型　过程　冗余　冗余　Q4AR



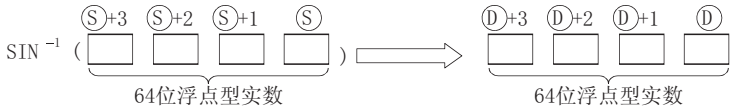
Ⓢ ：SIN^{-1}（反正弦）运算的 SIN 值或存储 SIN 值的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 从由 Ⓢ 指定的 SIN（正弦）值计算角度，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 由 Ⓢ 指定的 SIN 值可在从 -1.0 到 1.0 的范围之内。

(3) 存储在 Ⓓ 中的角度（运算结果）是以弧度单位存储的。
关于角度和弧度值之间的转换，请参见 RADD 和 DEGD 指令。

(4) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

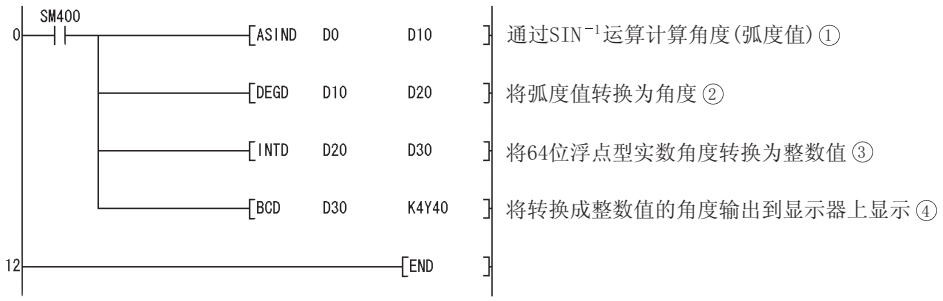
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： (出错代码： 4140)
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。 (出错代码： 4140)
- 由 ⓢ 指定的值在双精度浮点范围内但在 -1.0 到 1.0 的范围外。(出错代码： 4100)
- 运算结果超出了以下范围 (运算结果溢出):
  $2^{1024} \leqq |$ 运算结果 $|$ (出错代码： 4141)

程序示例

(1) 下列程序求出 D0 到 D3 中的 64 位浮点型实数的 SIN^{-1}(反正弦)，并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

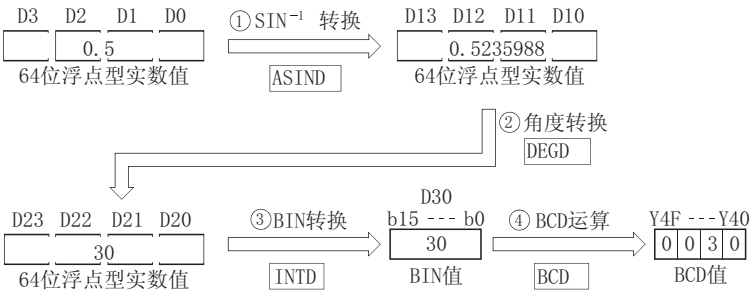
[梯形图模式]



[列表模式]

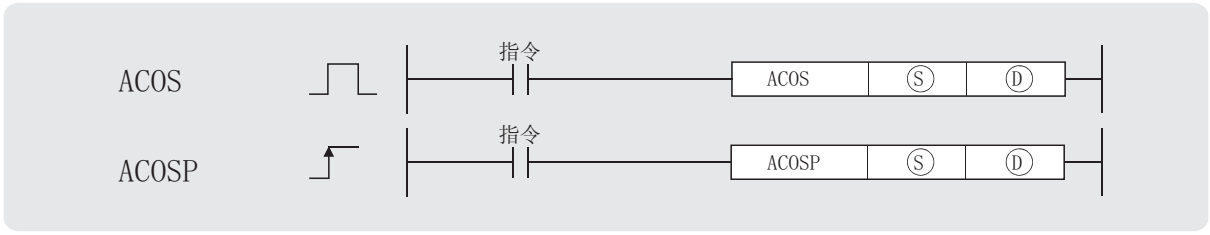
步	指令	软元件	
0	LD	SM400	
1	ASIND	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

[当 D0 和 D1 的值是 0.5 时的相关运算]



7.12.9 浮点数的 COS $^{-1}$ 运算（单精度）(ACOS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：存储将要执行 COS^{-1}（反余弦）运算的 COS 值的软元件的起始软元件号（实数）

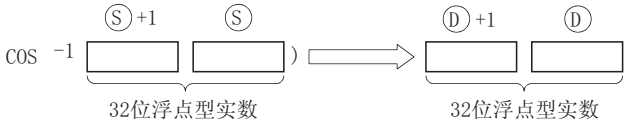
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	---	○		---	○		○*1	○	---
Ⓓ	---	○		---	○		○*1	---	---

*1：只适用于通用型 QCPU。

功 能

(1) 从由 Ⓢ 指定的 COS^{-1} 值中计算角度，并且在由 Ⓓ 指定的字软元件中存储运算结果。



(2) 由 Ⓢ 指定的 COS 值能在 -1.0 到 1.0 的范围之内。

(3) 在 Ⓓ 中存储的角度（运算结果）是按弧度单位来存储的。
关于角度数据和弧度之间转换的更多信息，请参见 DEG 和 RAD 指令的描述。

运算错误

(1) 在下列发生运行错误的情况中，出错标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的值在 -1.0 到 1.0 的范围之外 . （出错代码： 4100）
- 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内（只对于通用型 QCPU)：
  （出错代码： 4140）

  0、 2^{-126} ≦ ｜ 指定软元件的内容 ｜ < 2^{128}

- 指定的软元件值为 -0。 *2
  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  （出错代码： 4100）

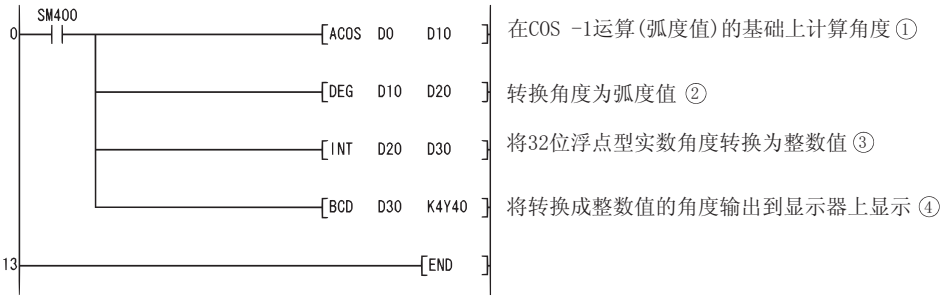
*2: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 结果超出了以下范围（运算结果溢出）
  （只对于通用型 QCPU)
  2^{128} ≦ ｜ 运算结果 ｜ （出错代码： 4141）
- 指定软元件的值为 -0、非正规数、非数和 ±∞ 。
  （只对于通用型 QCPU) （出错代码： 4140）

程序示例

(1) 下列程序求出 D0 到 D1 中的 32 位浮点型实数的 COS^{-1}（反余弦），并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

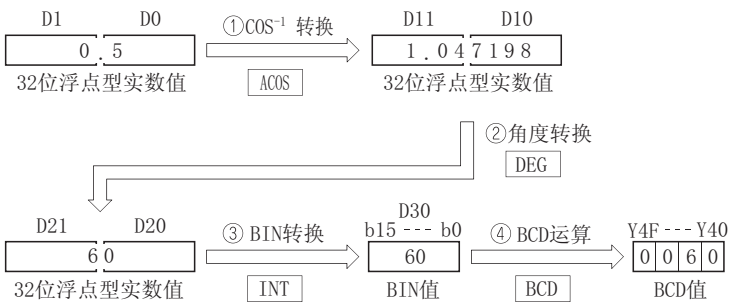
[梯形图模式]



[列表模式]

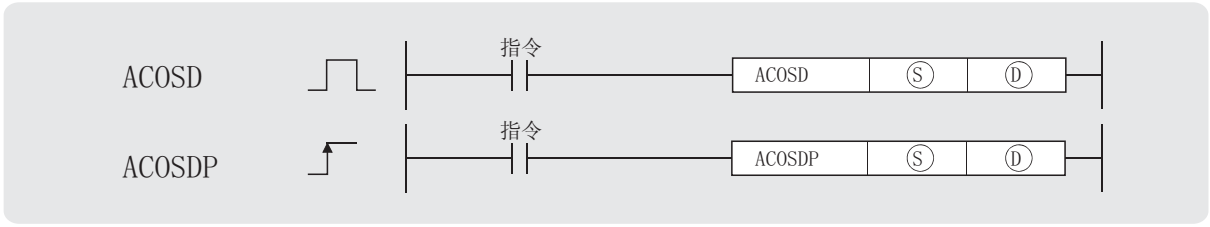
步	指令	软元件	
0	LD	SM400	
1	ACOS	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[当 D0 和 D1 的值是 0.5 时的相关运算]



7.12.10 浮点数的 COS $^{-1}$ 运算（双精度）(ACOSD(P))

通用型 UD　基本型　高性能型　过程　冗余　冗余　Q4AR



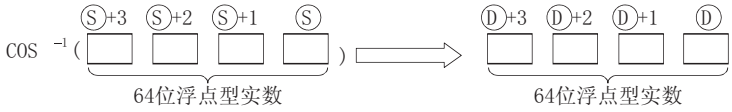
Ⓢ ：执行 COS^{-1}（反正弦）运算的 COS 值或存储 COS 值的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 从由 Ⓢ 指定的 COS（余弦）值计算角度，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 由 Ⓢ 指定的 COS 值可在从 -1.0 到 1.0 的范围之内。

(3) 存储在 Ⓓ 中的角度（运算结果）是以弧度单位存储的。
关于角度和弧度值之间的转换，请参见 RADD 和 DEGD 指令。

(4) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

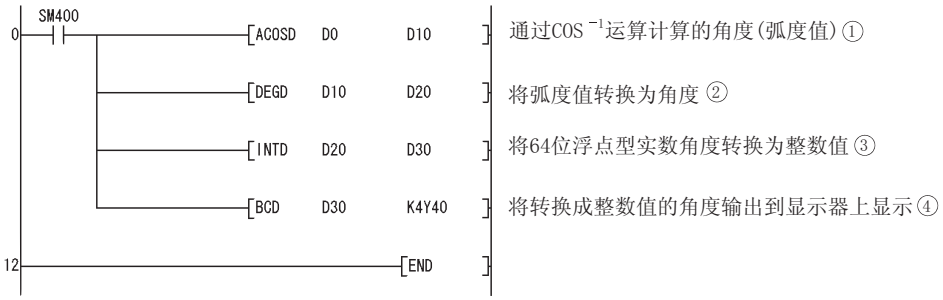
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： (出错代码： 4140)
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。 (出错代码： 4140)
- 由 (S) 指定的值在双精度浮点范围内但在 -1.0 到 1.0 的范围外。(出错代码： 4100)
- 运算结果超出了以下范围 (运算结果溢出):
  $2^{1024} \leqq |$ 运算结果 $|$ (出错代码： 4141)

程序示例

(1) 下列程序求出 D0 到 D3 中的 64 位浮点型实数的 COS^{-1}(反余弦)，并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

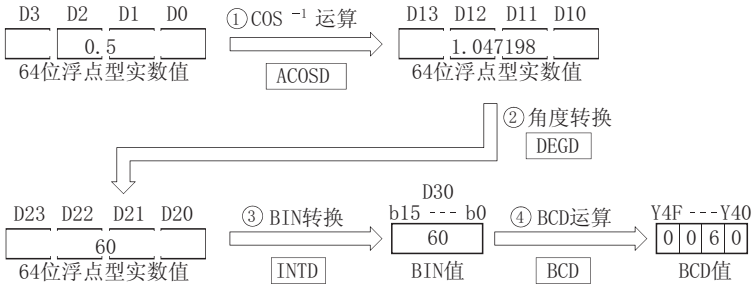
[梯形图模式]



[列表模式]

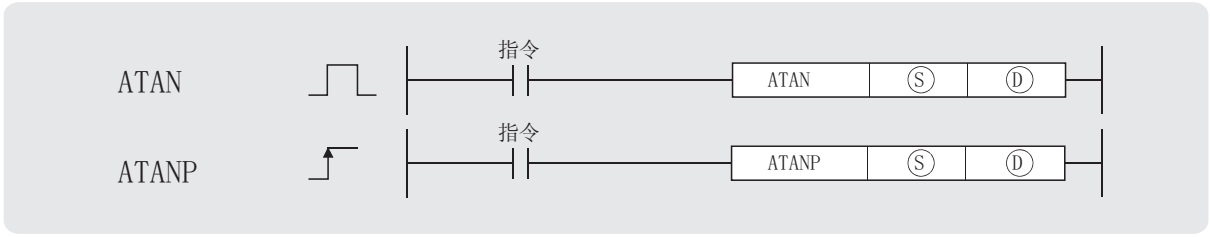
步	指令	软元件	
0	LD	SM400	
1	ACOSD	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

[当 D0 到 D3 的值是 0.5 时的动作]



7.12.11 浮点数 TAN $^{-1}$ 运算（单精度）(ATAN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：存储将要执行由 TAN^{-1}（反正切）操作的 TAN 值的软元件的起始软元件号（实数）

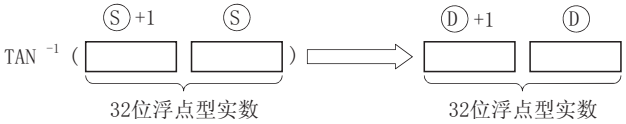
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*1	○	---
Ⓓ	---	○		---	○		○*1	---	---

*1：只适用于通用型 QCPU。

功 能

(1) 从由 Ⓢ 指定的 TAN 值中计算角度，并且在由 Ⓓ 指定的字软元件中存储运算结果。



(2) 在 Ⓓ 中存储的角度（运算结果）是按弧度单位来存储的。
关于角度数据和弧度之间转换的更多信息，请参见 DEG 和 RAD 指令的描述。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内 (只对于通用型 QCPU):
(出错代码： 4140)

$0、\ 2^{-126} \leqq |$ 指定软元件的内容 $| < 2^{128}$

• 指定的软元件值为 -0。 *2
(对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
(出错代码： 4100)

*2: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

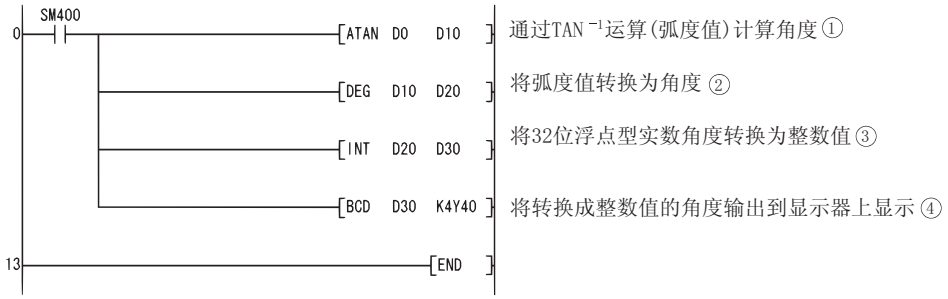
• 结果超出了以下范围 (运算结果溢出)
(只对于通用型 QCPU)
$2^{128} \leqq |$ 运算结果 $|$ (出错代码： 4141)

• 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$。
(只对于通用型 QCPU) (出错代码： 4140)

程序示例

(1) 下列程序求出 D0 和 D1 中的 32 位浮点型实数的 TAN^{-1}(反正切)，并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

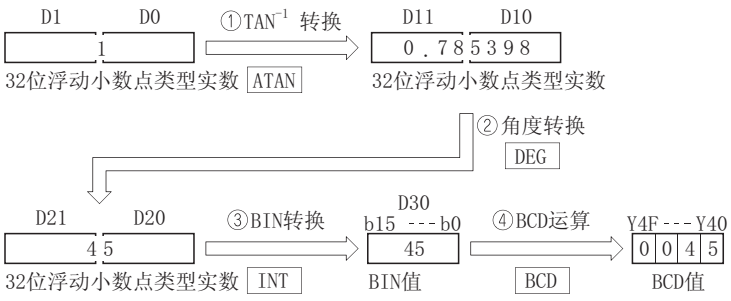
[梯形图模式]



[列表模式]

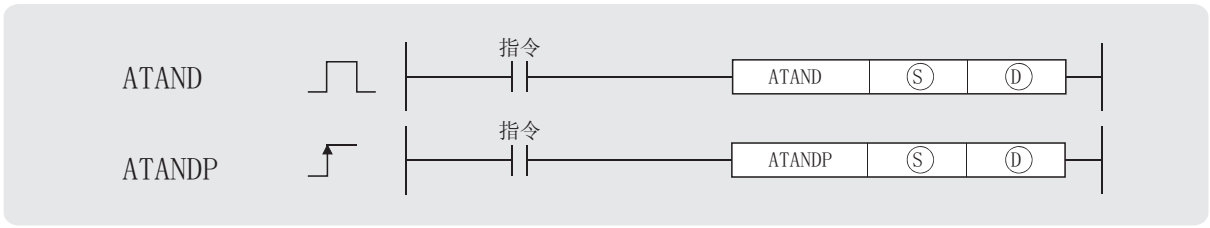
步	指令	软元件	
0	LD	SM400	
1	ATAN	D0	D10
4	DEG	D10	D20
7	INT	D20	D30
10	BCD	D30	K4Y40
13	END		

[当 D0 和 D1 的值是 1 时的相关运算]



7.12.12 浮点数 TAN $^{-1}$ 运算（双精度）(ATAND(P))

通用型 UD　基本型　高性能型　过程　冗余　冗余　Q4AR



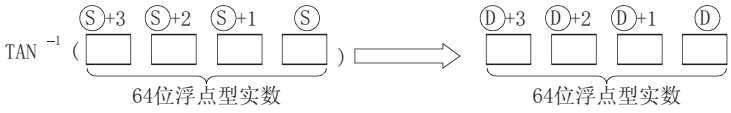
Ⓢ ：进行 TAN^{-1}（反正切）运算的 TAN 值或存储 TAN 值的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 从由 Ⓢ 指定的 TAN（正切）值计算角度，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 存储在 Ⓓ 中的角度（运算结果）是以弧度单位存储的。
关于角度和弧度值之间的转换，请参见 RADD 和 DEGD 指令。

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

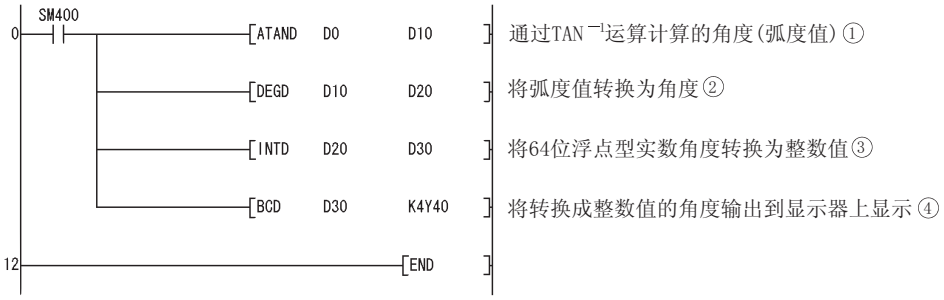
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： （出错代码： 4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件的值为 -0。 （出错代码： 4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |$ 运算结果 $|$ （出错代码： 4141）

程序示例

(1) 下列程序求出 D0 到 D3 中的 64 位浮点型实数的 TAN^{-1}(反正切)，并将该角度以 BCD4 位输出到 Y40 ～ Y4F 中。

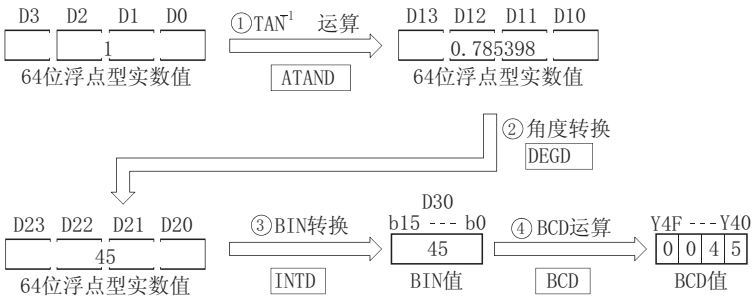
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	ATAND	D0	D10
4	DEGD	D10	D20
7	INTD	D20	D30
10	BCD	D30	K4Y40
12	END		

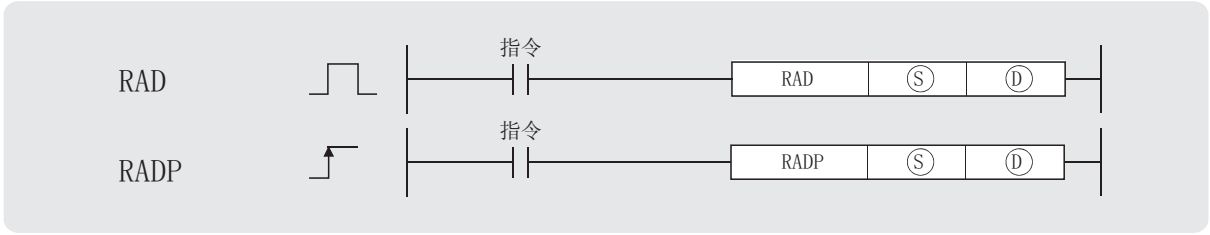
[当 D0 到 D3 的值是 1 时的动作]



7.12.13 从浮点数角度到弧度的转换（单精度）(RAD(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：存储将要转换成弧度的角度的软元件的起始软元件号（实数）

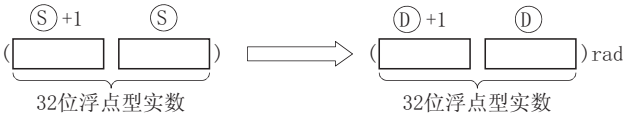
Ⓓ：在转换结束之后存储弧度值的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 将角度大小的单位从由 Ⓢ 指定的角度单位转换成弧度单位，并且在由 Ⓓ 指定软元件号的软元件中存储结果。



(2) 按照下列等式执行角度到弧度单位的转换：

$$\text{弧度单位} = \text{角度单位} \times \frac{\pi}{180}$$

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内 (只对于通用型 QCPU):
  (出错代码 : 4140)

  0、2^{-126} ≦ | 指定软元件的内容 | < 2^{128}

- 指定的软元件值为 -0。 *3
  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  (出错代码 : 4100)

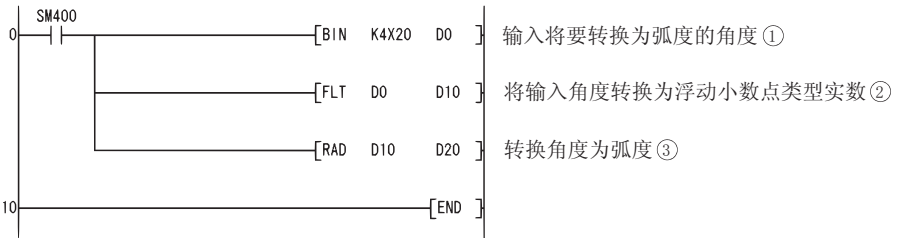
*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 结果超出了以下范围 (运算结果溢出)
  (只对于通用型 QCPU)
  2^{128} ≦ | 运算结果 | (出错代码 : 4141)
- 指定软元件的值为 -0、非正规数、非数和±∞ 。
  (只对于通用型 QCPU) (出错代码 : 4140)

程序示例

(1) 下列程序将在 X20 到 X2F 中的 4 个 BCD 位数设定的角度转换为弧度，并且在 D20 和 D21 中将结果存储为浮点类型实数。

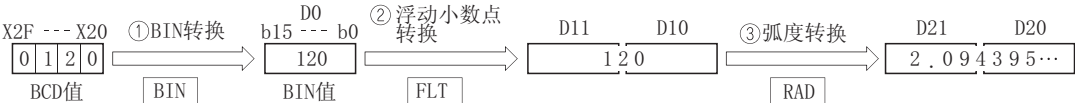
[梯形图模式]



[列表模式]

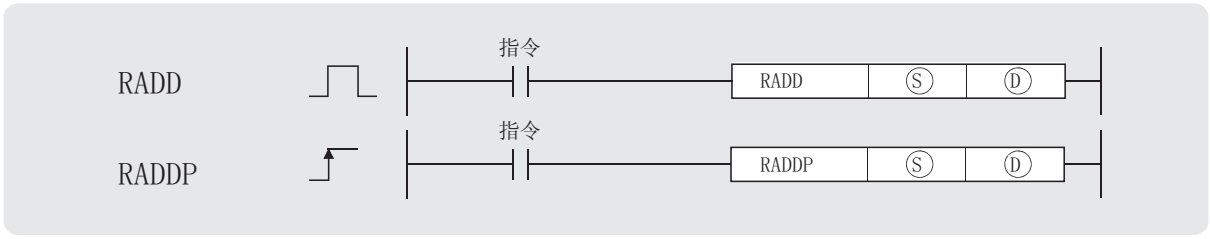
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D0
4	FLT	D0	D10
7	RAD	D10	D20
10	END		

[当 X20 到 X2F 指定的值为 120 时的相关运算]



7.12.14 从浮点数角度到弧度的转换（双精度）(RADD(P))

通用型 UD　基本型　高性能型　过程　冗余　冗余　Q4AR



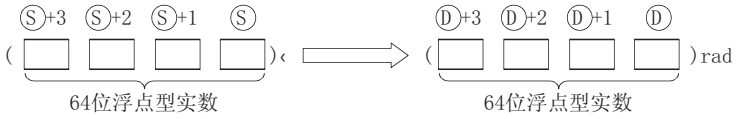
Ⓢ ：将要转换成弧度单位的角度或存储角度的软元件的起始号（实数）

Ⓓ ：存储转换为弧度单位的值的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 将角度大小的单位从 Ⓢ 指定的角度单位转换成弧度单位，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 按照下列公式执行角度到弧度单位的转换。

$$\text{弧度单位} = \text{角度单位} \times \frac{\pi}{180}$$

运算错误

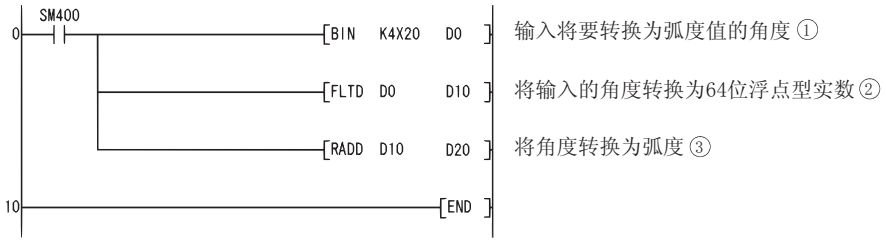
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： （出错代码： 4140）
  $0、2^{-1022} \leqq |\text{指定的软元件值}| < 2^{1024}$
- 指定的软元件值为 -0。 （出错代码： 4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |\text{运算结果}|$ （出错代码： 4141）

程序示例

(1) 下列程序将在 X20 到 X2F 中的 BCD4 位中设定的角度转换为弧度，并以 64 位浮点型实数存储到 D20 到 D23 中。

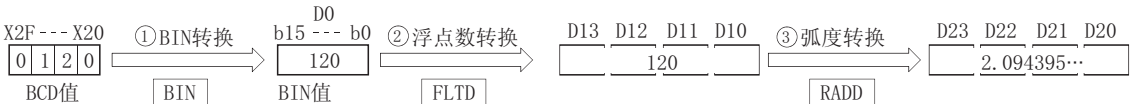
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D0
3	FLTD	D0	D10
6	RADD	D10	D20
9	END		

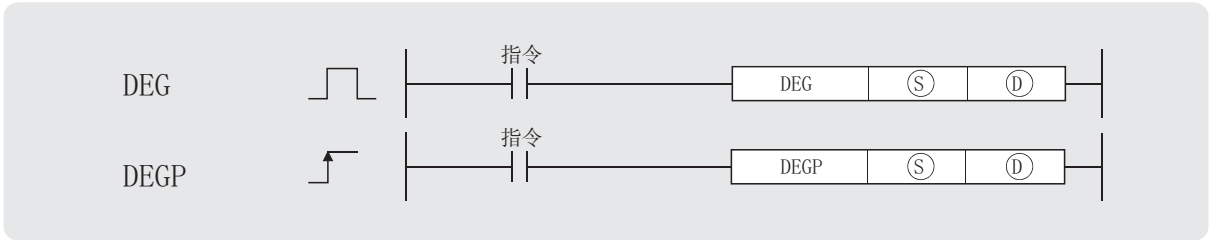
[当 X20 到 X2F 指定的值为 120 时的动作]



7.12.15 从浮点数弧度到角度的转换（单精度）(DEG(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：要转换为角度的弧度或存储弧度的软元件的起始号（实数）

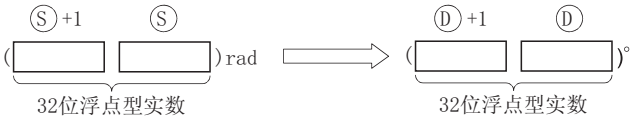
Ⓓ：存储已转换为角度的值的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 将角度大小的单位从由 Ⓢ 指定的弧度单位转换为角度单位，并且在由 Ⓓ 指定软元件号的软元件中存储结果。



(2) 按照下列等式执行从弧度到角度的转换：

$$\text{弧度单位} = \text{角度单位} \times \frac{180}{\pi}$$

运算错误

(1) 在下列情况下将发生运算错误，出错标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值为 -0。 *3
  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  (出错代码： 4100)

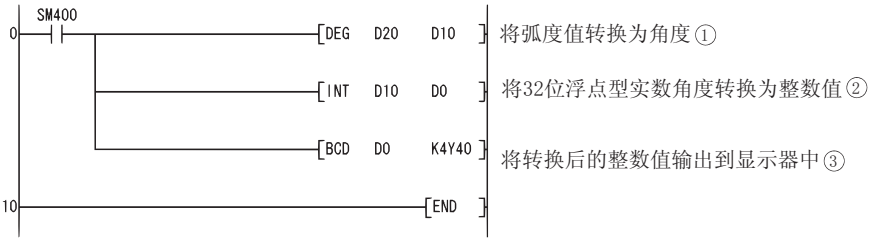
*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 结果超出了以下范围 (运算结果溢出)
  (只对于通用型 QCPU)
  $2^{128} \leqq |$ 运算结果 $|$ (出错代码： 4141)
- 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$。
  (只对于通用型 QCPU) (出错代码： 4140)

程序示例

(1) 下列程序将在 D20 和 D21 中的 32 位浮点型实数中设置的弧度值转换为角度，并以 BCD 值输出到 Y40 ～ Y4F 中。

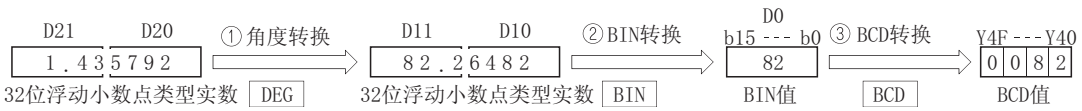
[梯形图模式]



[列表模式]

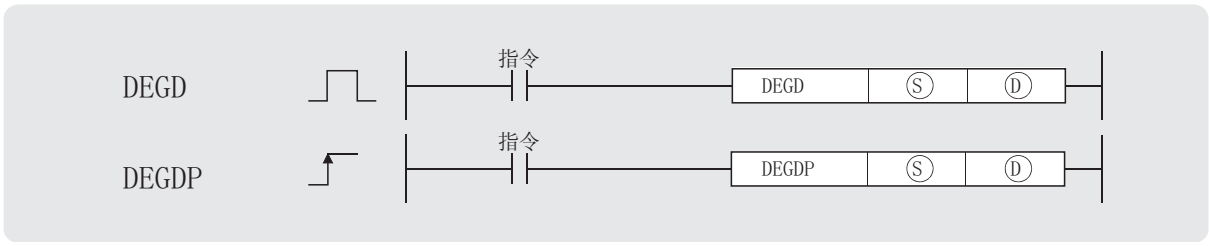
步	指令	软元件	
0	LD	SM400	
1	DEG	D20	D10
4	INT	D10	D0
7	BCD	D0	K4Y40
10	END		

[当在 D20 和 D21 中指定的值为 1.435792 时的相关运算]



7.12.16 从浮点数弧度到角度的转换（双精度）(DEGD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



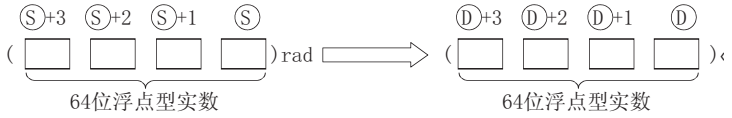
Ⓢ ：要转换为角度单位的弧度或存储弧度的软元件的起始号（实数）

Ⓓ ：存储已转换为角度单位的值的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 将角度大小的单位从由 Ⓢ 指定的弧度单位转换成角度单位，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 按照下列公式执行角度到弧度单位的转换。

$$\text{弧度单位} = \text{角度单位} \times \frac{180}{\pi}$$

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

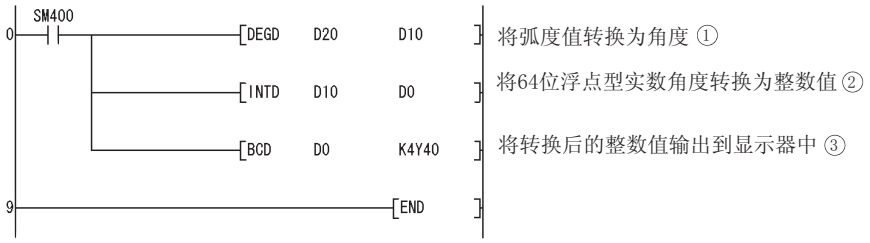
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： （出错代码： 4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。 （出错代码： 4140）
- 运算结果超出了以下范围（运算结果溢出）:
  $2^{1024} \leqq |$ 运算结果 $|$ （出错代码： 4141）

程序示例

(1) 下列程序将在 D20 和 D23 中的 64 位浮点型实数中设置的弧度值转换为角度，并以 BCD 值存储到 Y40 ～ Y4F 中。

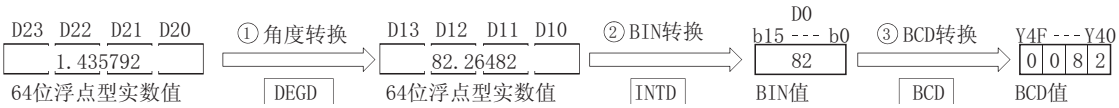
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DEGD	D20	D10
4	INTD	D10	D0
7	BCD	D0	K4Y40
9	END		

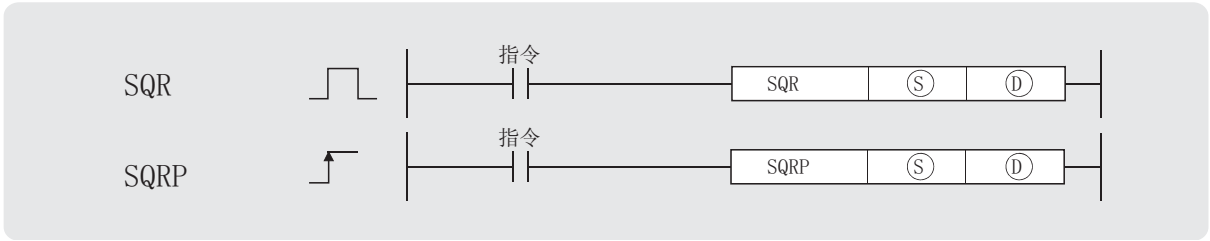
[当在 D20 和 D23 中指定的值为 1.435792 时的动作]



7.12.17 浮点数的平方根运算（单精度）(SQR(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：存储将要执行平方根运算的数据的软元件的起始软元件号（实数）

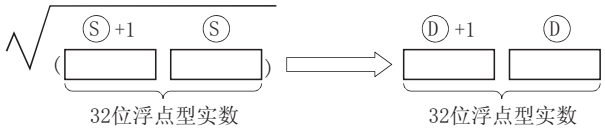
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 计算由 Ⓢ 指定的值的平方根，并且在由 Ⓓ 指定软元件号的软元件中存储运算结果。



(2) 只有正值才能由 Ⓢ 指定。（在负值上不能执行运算）

运算错误

(1) 在下列发生运行错误的情况中，出错标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的值是一个负值。 (出错代码： 4100)
- 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内 (只对于通用型 QCPU):
  (出错代码： 4140)

  0、2^{-126} ≦ | 指定软元件的内容 | < 2^{128}

- 指定的软元件值为 -0。 *3
  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  (出错代码： 4100)

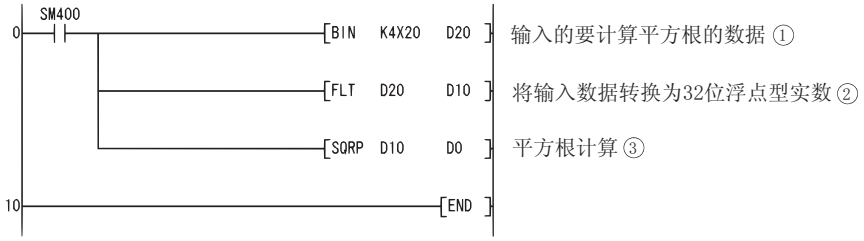
*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围 (运算结果溢出):
  (只对于通用型 QCPU)
  2^{128} ≦ | 运算结果 | (出错代码： 4141)
- 指定软元件的值为 -0、非正规数、非数和 ± ∞ 。
  (只对于通用型 QCPU) (出错代码： 4140)

程序示例

(1) 下列程序查找由从 X20 到 X2F 的 4 个 BCD 位数设置的值的平方根，并且在 D0 和 D1 中将结果存储为浮点类型实数。

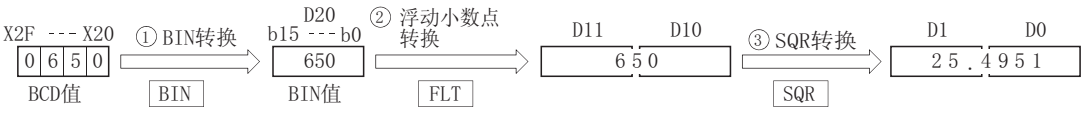
[梯形图模式]



[列表模式]

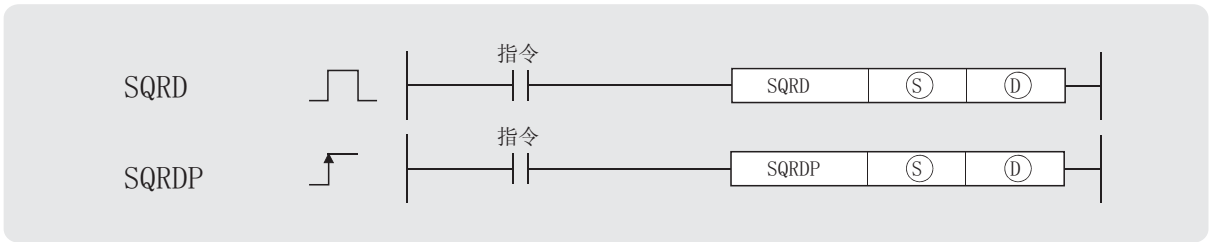
步	指令	软元件	
0	LD	SM400	
1	BIN	K4X20	D20
4	FLT	D20	D10
7	SQRP	D10	D0
10	END		

[当由 X20 到 X2F 指定的值是 650 时的运算]



7.12.18 浮点数的平方根运算（双精度）(SQRD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



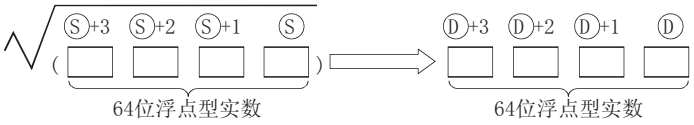
Ⓢ ：执行平方根运算的数据或存储该数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 对由 Ⓢ 指定的值进行平方根运算，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 由 Ⓢ 指定的值只能为正值。（不能以负值进行运算。）

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

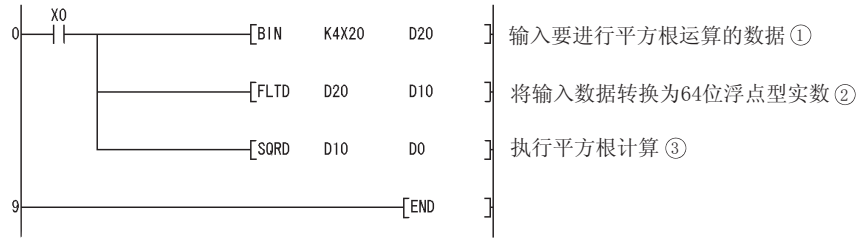
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 由 Ⓢ 指定的值是一个负值。（出错代码：4100）
- 指定的软元件值不在以下范围中：（出错代码：4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。（出错代码：4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |$ 运算结果 $|$（出错代码：4141）

程序示例

(1) 下列程序求出由 X20 到 X2F 的 BCD4 位中设置的值的平方根，并以 64 位浮点型实数存储到 D0 ～ D3 中。

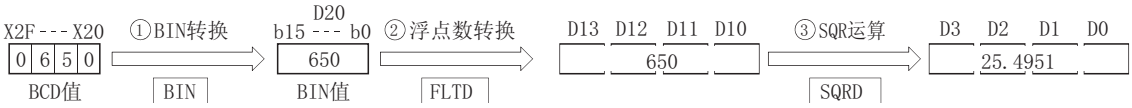
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	BIN	K4X20	D20
3	FLTD	D20	D10
6	SQRD	D10	D0
9	END		

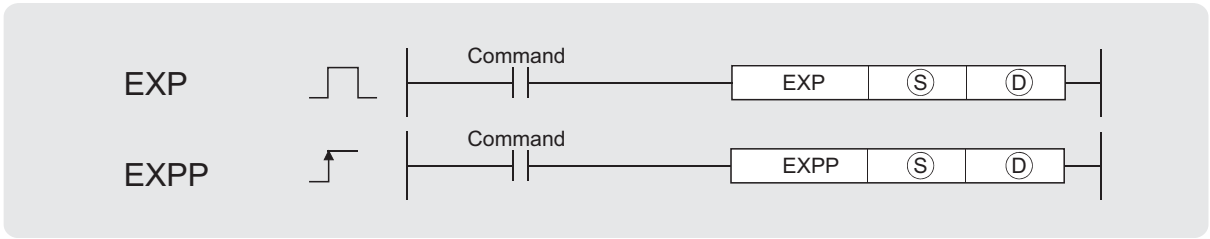
[当由 X20 到 X2F 指定的值是 650 时的动作]



7.12.19 浮点数的指数运算（单精度）(EXP(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：软元件的起始软元件号，在这些软元件中存储着指数运算将要执行的数据（实数）

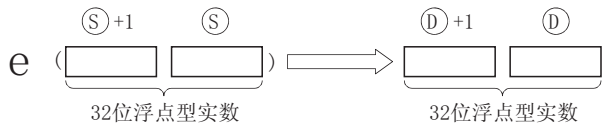
Ⓓ：运算结果由存储的软元件的起始软元件号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 计算由 Ⓢ 指定值的指数，并且在由 Ⓓ 指定的软元件中存储运算结果。



(2) 在指数运算中，底是按 (e) 是 “2.71828” 来计算的。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 运算结果不在下面表示的范围之内 : (出错代码 : 4100)

  $2^{-126} \leqq$ | 运算结果 | $\leqq 2^{128}$

  (对于高性能型 QCPU/QnACPU/Q4ARCPU)

  $2^{-126} \leqq$ | 运算结果 | $< 2^{128}$

  (对于基本型 QCPU/ 过程 CPU/ 冗余 CPU)

- 指定的软元件值为 -0。 *3

  (对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)

  (出错代码 : 4100)

*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围 (运算结果溢出):

  (只对于通用型 QCPU)

  $2^{128} \leqq$ | 运算结果 | (出错代码 : 4141)

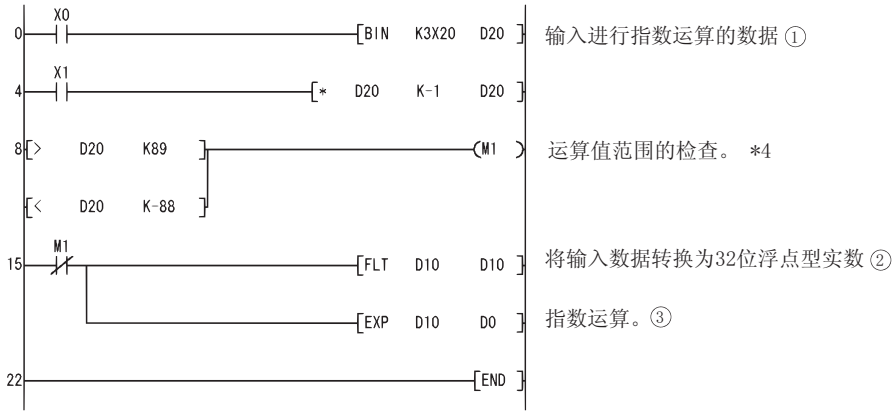
- 指定软元件的值为 -0、非正规数、非数和 $\pm\infty$。

  (只对于通用型 QCPU) (出错代码 : 4140)

程序示例

(1) 下列程序对由 X20 到 X27 中的 BCD2 位中设置的值执行指数运算，并以 32 位浮点型实数存储到 D0 和 D1 中。

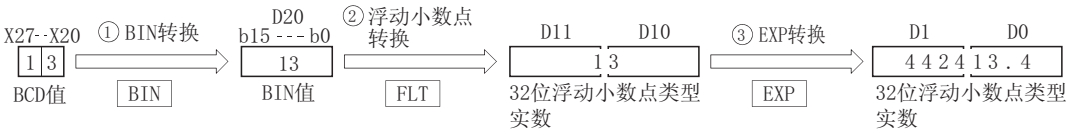
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	BIN	K3X20	D20	
4	LD	X1		
5	*	D20	K-1	D20
8	LD>	D20	K89	
11	OR<	D20	K-88	
14	OUT	M1		
15	LDI	M1		
16	FLT	D10	D10	
19	EXP	D10	D0	
22	END			

[当由 X20 到 X27 指定的值为 13 时的相关运算]



*4: 由于 $\log_e 2^{129} = 89.4$，因此如果 X20 到 X27 的 BCD 值小于 89，则运算结果将小于 2^{129} 。
由于设定超过 90 的值会发生运算错误，因此如果设定的值高于 90，应将 M0 置为 ON 以避免发生错误。

要 点

将自然对数转换为常用对数

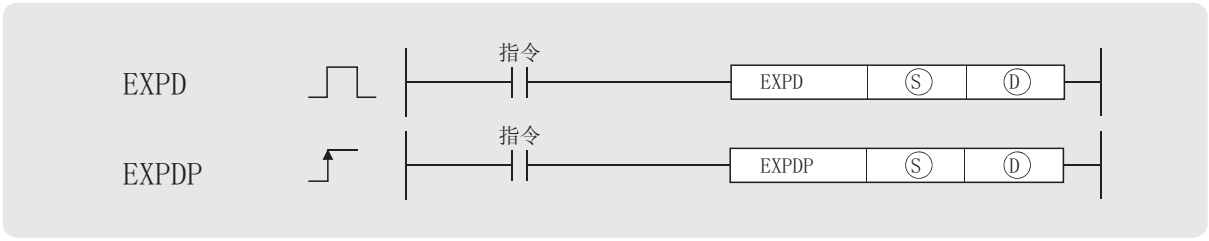
在 CPU 模块中，使用自然对数进行计算。

要获得一个常用对数值，应指定为将 (S) 中的常用对数值除以 0.43429 后所得的值。

$$10^x = e^{\frac{x}{0.43429}}$$

7.12.20 浮点数的指数运算（双精度）(EXPD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



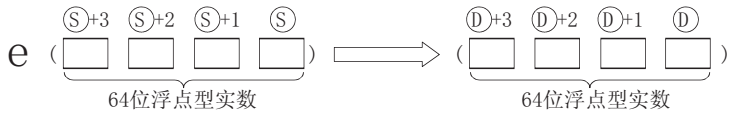
Ⓢ ：进行指数运算的数据或存储该数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 对 Ⓢ 指定的值进行指数运算，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) 在指数运算中，底 (e) 是以 “2.71828” 进行运算的。

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

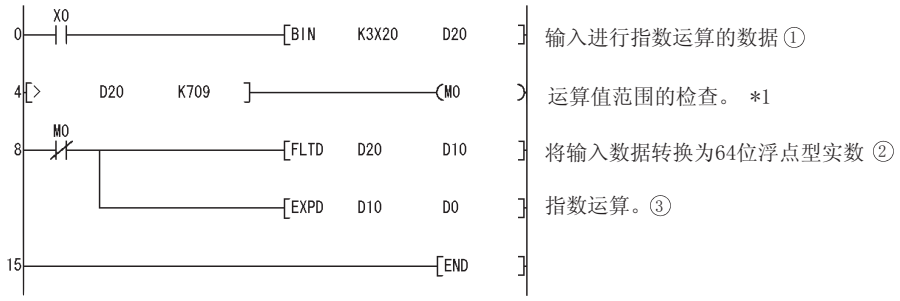
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 指定的软元件值不在以下范围中： （出错代码： 4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。 （出错代码： 4140）
- 运算结果超出了以下范围（运算结果溢出）:
  $2^{1024} \leqq |$ 运算结果 $|$ （出错代码： 4141）

程序示例

(1) 下列程序对 X20 到 X31 的 BCD2 位中设置的值进行指数运算，并以 64 位浮点型实数存储到 D0 ～ D3 中。

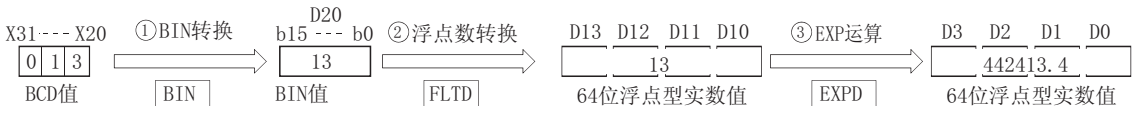
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	BIN	K3X20	D20
4	LD>	D20	K709
7	OUT	M0	
8	LDI	M0	
9	FLTD	D20	D10
12	EXPD	D10	D0
15	END		

[当由 X20 到 X31 指定的值是 13 时的动作]



*1: 由于 2^{1024} = 709.7832，因此如果 X20 到 X31 的 BCD 值小于 709，则运算结果将小于 2^{1024}。
由于设定超过 710 的值会发生运算错误，因此如果设定的值高于 710，应将 M0 置为 ON 避免发生错误

要 点

将自然对数转换为常用对数

在 CPU 模块中，使用自然对数进行计算。

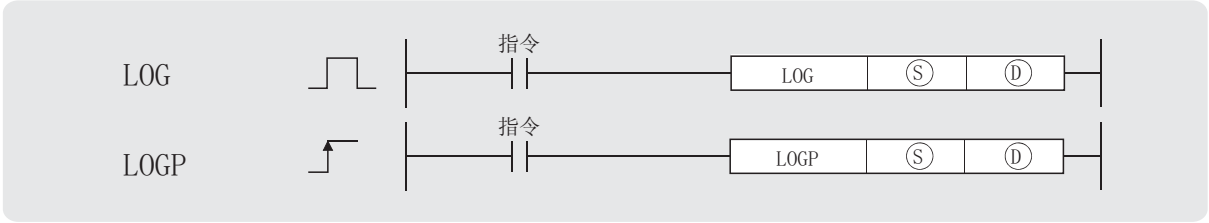
要获得一个常用对数值，应指定为将 (S) 中的常用对数值除以 0.43429 后所得的值。

$$10^{X}=e^{\frac{X}{0.43429}}$$

7.12.21 浮点数的自然对数运算（单精度）(LOG(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓢ：存储用于自然对数运算的数据的软元件的起始软元件号（实数）

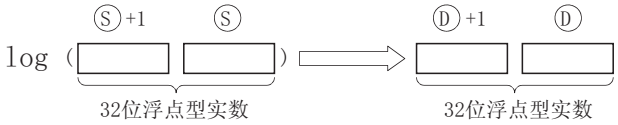
Ⓓ：存储运算结果的软元件的起始软元件号（实数）

Settiing Data	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 E	其它
	位	字		位	字				
Ⓢ	---	○		---	○		○*2	○	---
Ⓓ	---	○		---	○		○*2	---	---

*2：只适用于通用型 QCPU。

功 能

(1) 以 (e) 作底计算由 Ⓢ 指定的值的自然对数，并且在由 Ⓓ 指定的软元件中存储运算结果。



(2) 只有正值才能由 Ⓢ 指定。（不能对负数执行运算。）

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的值是负值。 （出错代码： 4100）
- 由 Ⓢ 指定的值是 0。 （出错代码： 4100）
- 指定软元件的内容以及加法运算结果不为 “0”或不在以下范围内（只对于通用型 QCPU)：
  （出错代码： 4140）

  0、2^{-126} ≦ ｜指定软元件的内容｜< 2^{128}

- 指定的软元件值为 -0。 *3
  （对于基本型 QCPU、高性能型 QCPU、过程 CPU、冗余 CPU 和 Q4ARCPU)
  （出错代码： 4100）

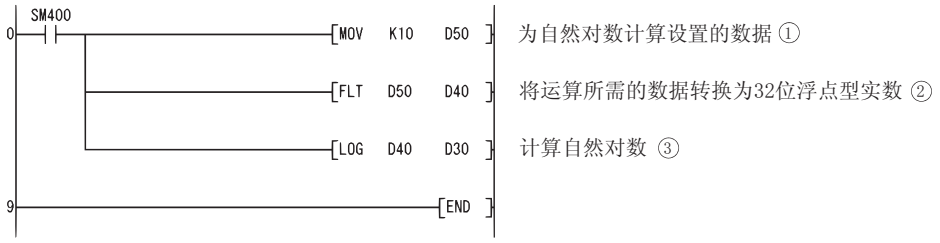
*3: 有的 CPU 模块即使指定的是 -0，也不会发出运算出错信息。详细内容请参阅 3.2.4 项。

- 运算结果超出了以下范围（运算结果溢出）:
  （只对于通用型 QCPU)
  2^{128} ≦ ｜运算结果｜ （出错代码： 4141）
- 指定软元件的值为 -0、非正规数、非数和±∞。
  （只对于通用型 QCPU) （出错代码： 4140）

程序示例

(1) 下列程序查找由 D50 设置的值 “10”的自然对数，并且在 D30 和 D31 中存储结果。

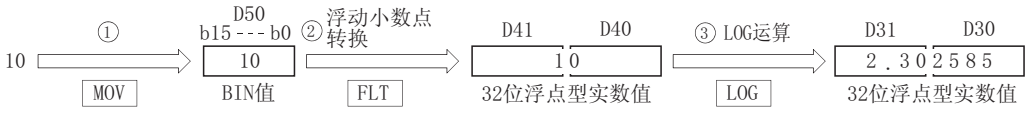
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	MOV	K10	D50
3	FLT	D50	D40
6	LOG	D40	D30
9	END		

[动作]



要 点

从自然对数到常用对数的转换

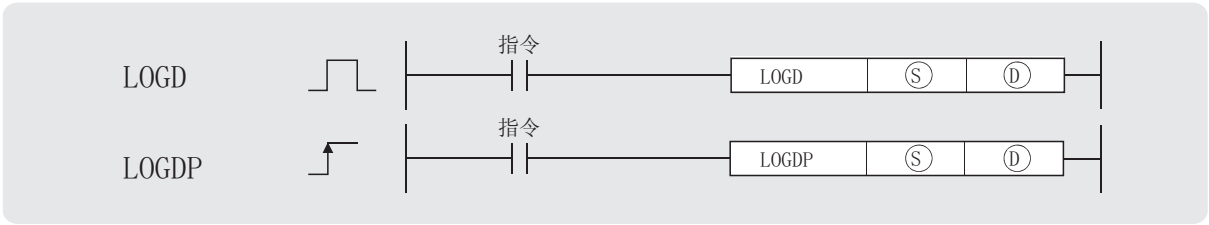
CPU 模块在运算中使用自然对数。

如果要得到常用对数值，可以用下列表达式转换自然对数的值：

$\log_{10}X = 0.43429 \times \log_e X$

7.12.22 浮点数的自然对数运算（双精度）(LOGD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



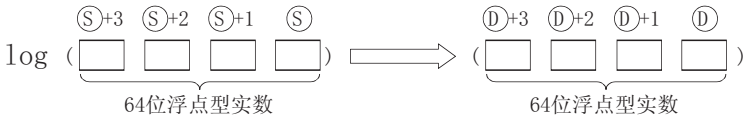
Ⓢ ：进行自然对数运算的数据或存储数据的软元件的起始号（实数）

Ⓓ ：存储运算结果的软元件的起始号（实数）

Settiing Data	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 E	其它
Ⓢ	---	○		---				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 以 Ⓢ 指定的值的自然对数 (e) 作底进行对数运算，并将运算结果存储到 Ⓓ 指定的软元件中。



(2) Ⓢ 中指定的值只能为正值。（不能以负数进行运算。）

(3) 当运算结果为 -0 或下溢时，将运算结果作为 0 处理。

运算错误

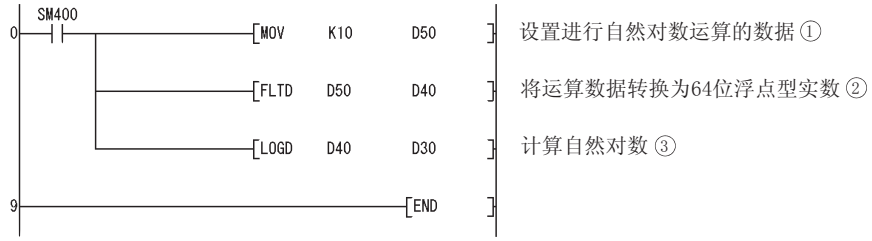
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 由 Ⓢ 指定的值是负值。（出错代码：4100）
- 由 Ⓢ 指定的值是 0。（出错代码：4100）
- 指定的软元件值不在以下范围中：（出错代码：4140）
  $0、2^{-1022} \leqq |$ 指定的软元件值 $| < 2^{1024}$
- 指定的软元件值为 -0。（出错代码：4140）
- 运算结果超出了以下范围（运算结果溢出）：
  $2^{1024} \leqq |$ 运算结果 $|$（出错代码：4141）

程序示例

(1) 下列程序求出由 D50 设置的值 “10” 的自然对数，并且在 D30 到 D33 中存储结果。

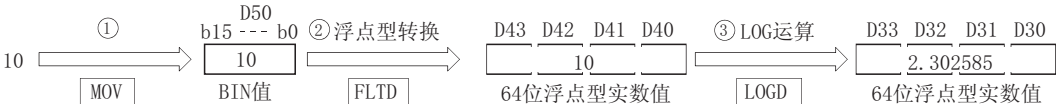
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	MOV	K10	D50
3	FLTD	D50	D40
6	LOGD	D40	D30
9	END		

[动作]



要 点

将自然对数转换为常用对数

在 CPU 模块中，使用自然对数来进行计算。

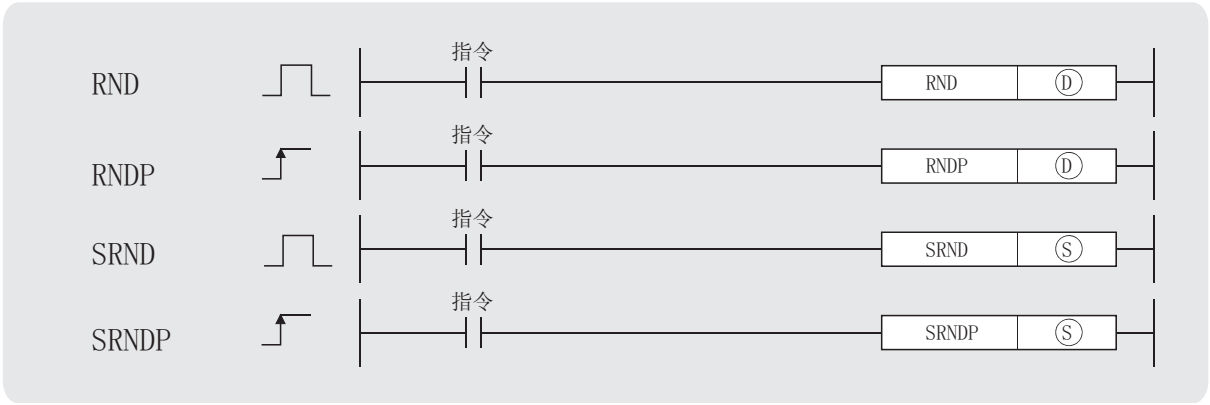
要获得一个常用对数值，可以用下列公式转换自然对数的值。

$\log_{10}X = 0.43429 \times \log_{e}X$

7.12.23 随机数的产生和系列更新 (RND(P)、SRND(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



Ⓓ：存储随机号的软元件的起始软元件号 (BIN 16 位)

Ⓢ：随机号系列数据或存储该类数据的软元件的起始软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓓ				○				---	---
Ⓢ				○				○	---

功 能

随机数产生指令用来产生与特定计算公式一致的随机数。在使用公式进行计算时，之前的计算结果将作为系数使用。
随机系列更改指令可以更改随机数产生方式。

RND

从 0 到 32767 中产生随机数，并且存储在由 Ⓓ 指定的软元件中。

SRND

根据由 Ⓢ 指定的软元件中存储的 16 位数据，更新随机数系列。

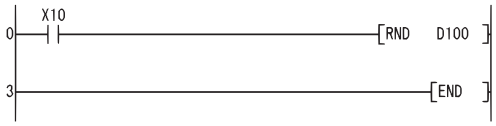
运算错误

(1) 没有与 RND(P) 或 SRND(P) 指令相关的运行错误。

程序示例

(1) 当 X10 为 ON 时，下列程序将随机数存储在 D100 中。

[梯形图模式]

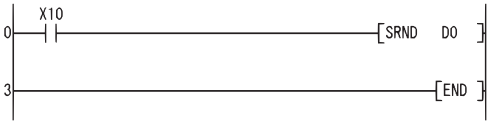


[列表模式]

步	指令	软元件
0	LD	X10
1	RND	D100
3	END	

(2) 当 X10 为 ON 时，下列程序根据 D0 的内容更新随机数系列。

[梯形图模式]

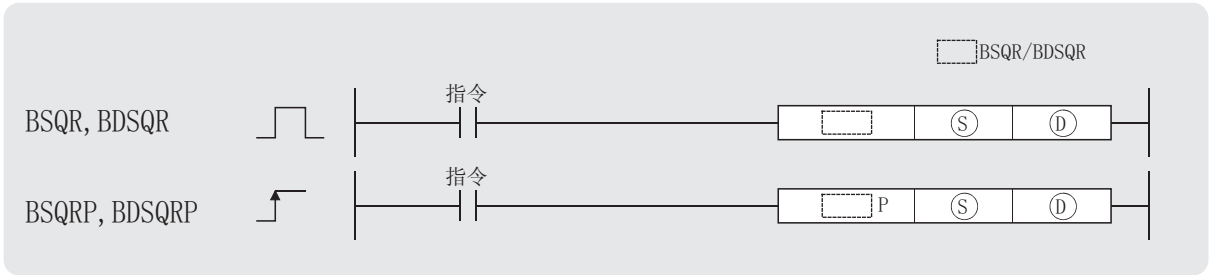


[列表模式]

步	指令	软元件
0	LD	X10
1	SRND	D0
3	END	

7.12.24 BCD4 位和 8 位平方根 (BSQR(P)、BDSQR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行平方根计算的数据或存储该类数据的软元件的软元件号 (BSQR(P): BCD 4 位、BDSQR(P): BCD 8 位)

Ⓓ : 存储平方根计算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○							○	---
Ⓓ	○							---	---

功 能

BSQR

(1) 计算在 Ⓢ 上指定的值的平方根，并且在由 Ⓓ 指定软元件号的软元件中存储运算结果。

$$\sqrt{\text{Ⓢ}} = \underbrace{\boxed{\text{Ⓓ 整数部分}}}_{\text{整数部分}} . \underbrace{\boxed{\text{Ⓓ+1 小数部分}}}_{\text{小数部分}}$$

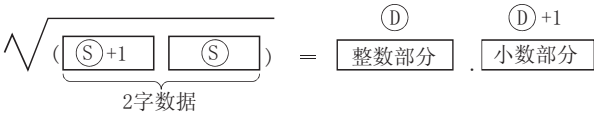
(2) 能够在 Ⓢ 中指定的值是最多 4 位数的 BCD 值 (从 0 到 9999)。

(3) Ⓓ 和 Ⓓ+1 的运算结果被存储为它们各自的 0 到 9999 之间的 BCD 值。

(4) 运算结果被统一为一个小数点后保留四位的值。
因此，第四个小数位有± 1 的误差。

BDSQR

(1) 对由 (S) 和 (S)+1 指定的值执行平方根运算，并且在由 (D) 指定的软元件中存储结果。



(2) (S) 和 (S)+1 能够指定的值最大为 8 位数的 BCD 值 (0 到 99999999)。

(3) (D) 和 (D)+1 的运算结果被存储为它们各自的 0 到 9999 之间的 BCD 值。

(4) 运算结果从小数点后第五位进行四舍五入。
因此，小数点后第四位的值有 ± 1 的误差。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 由 (S) 指定的数据不是一个 BCD 值。 (出错代码 : 4100)

程序示例

(1) 下列程序计算 BCD 值 1325 的平方根，并输出整数部分到从 Y50 到 Y5F 的 4 个 BCD 位中，输出小数部分到从 Y40 到 Y4F 的 4 个 BCD 位中。

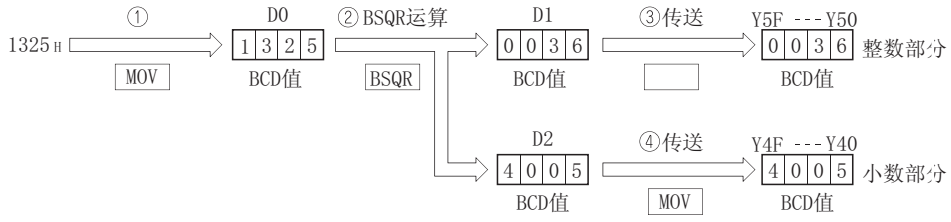
[梯形图模式]



[列表模式]

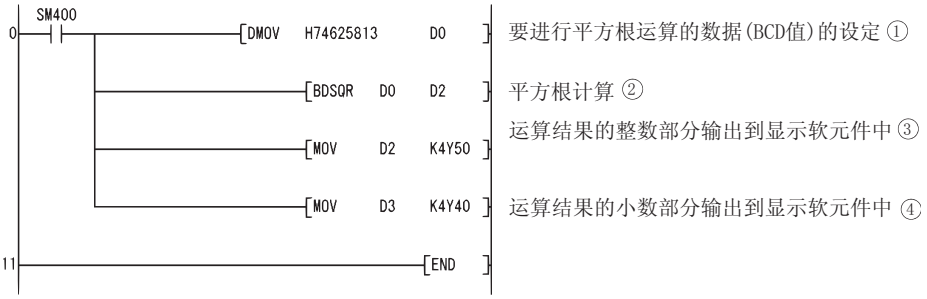
步	指令	软元件	
0	LD	SM400	
1	MOV	H1325	D0
3	BSQR	D0	D1
6	MOV	D1	K4Y50
8	MOV	D2	K4Y40
10	END		

[动作]



(2) 下列程序计算 BCD 值 74625813 的平方根，并输出结果的整数部分到从 Y50 到 Y5F 中的 4 个 BCD 位数上去，输出小数部分到从 Y40 到 Y4F 中的 4 个 BCD 位数上去。

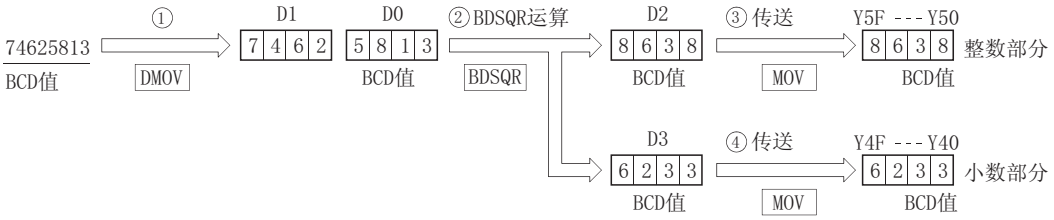
[梯形图模式]



[列表模式]

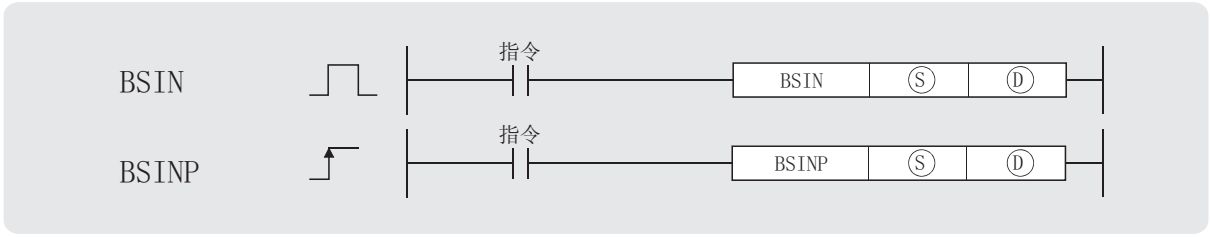
步	指令	软元件	
0	LD	SM400	
1	DMOV	H74625813	D0
4	BDSQR	D0	D2
7	MOV	D2	K4Y50
9	MOV	D3	K4Y40
11	END		

[动作]



7.12.25 BCD 型 SIN 运算 (BSIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行 SIN(正弦) 运算的数据，或者存储该类数据的软元件的软元件号 (BCD 4 位)

Ⓓ : 存储运算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○					---
Ⓓ	---	○		---					---

功 能

(1) 计算由 Ⓢ 指定的值 (角度) 的 SIN(正弦) 值，并在由 Ⓓ 指定的软元件中存储运算结果的符号，在由 Ⓓ+1 和 Ⓓ+2 指定的软元件中存储运算结果。

SIN Ⓢ = Ⓓ 符号　Ⓓ+1 整数部分 . Ⓓ+2 小数部分

(2) 由 Ⓢ 指定的值是一个在 0 度和 360 度 (以度为单位) 之间的 BCD 值。

(3) 如果结果是一个正值，在 Ⓓ 中存储的运算结果的符号将为 “0”，如果结果是一个负值，该符号为 “1”。

(4) 存储在 Ⓓ+1 和 Ⓓ+2 的运算结果是在 -1.000 和 1.000 之间的 BCD 值。

(5) 运算结果在小数点后的第五位上四舍五入。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

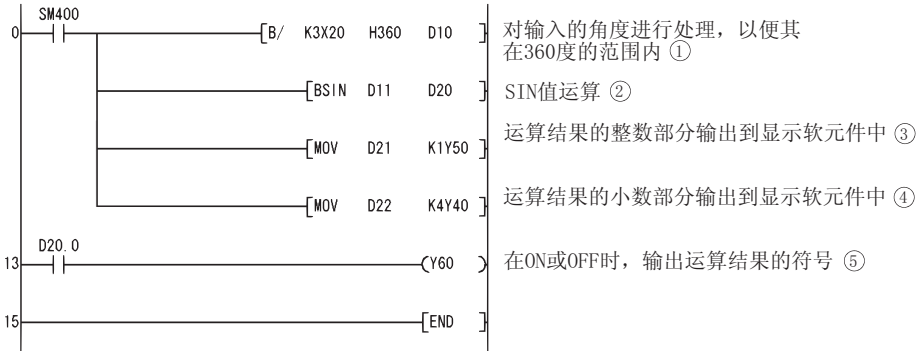
- 由 Ⓢ 指定的数据不是一个 BCD 值。 (出错代码 : 4100)
- 由 Ⓢ 指定的数据不是在 0 到 360 的范围之内。 (出错代码 : 4100)
- 由 Ⓓ 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码 : 4101)

程序示例

(1) 下面的程序示例计算由 X20 到 X28 指定的 3 位 BCD 数据的 SIN 值，并且输出一个 1 位 BCD 的整数部分数据到 Y50 到 Y53，以及 4 位 BCD 数据的小数部分到 Y40 到 Y4F。

如果运算结果是负的，Y60 变为 ON。（如果已经在 X20 到 X2F 中设置的值比 360 大，那么它将会被调整到 0 到 360 的范围里去。）

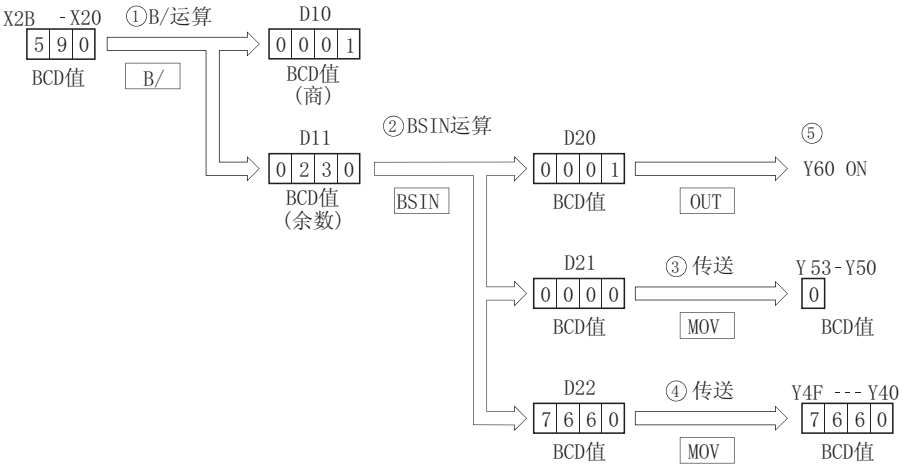
[梯形图模式]



[列表模式]

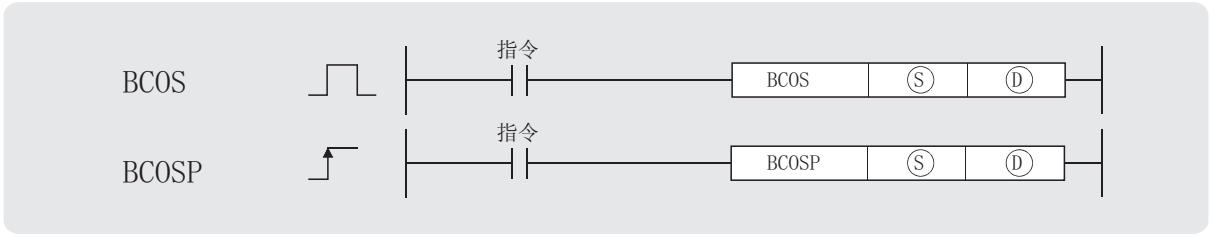
步	指令	软元件		
0	LD	SM400		
1	B/	K3X20	H360	D10
5	BSIN	D11	D20	
8	MOV	D21	K1Y50	
11	MOV	D22	K4Y40	
13	LD	D20.0		
14	OUT	Y60		
15	END			

[由 X20 到 X28 指定的值为 590 的相关运算]



7.12.26 BCD 型 COS 运算 (BCOS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行 COS(余弦)运算的数据，或存储该类数据的软元件的起始软元件号 (BCD 4 位)

Ⓓ : 存储运算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○			○				---
Ⓓ	---	○			---				---

功 能

(1) 计算由 Ⓢ 指定的值(角度)的 COS(余弦)值，然后在由 Ⓓ 指定的字软元件中存储运算结果的符号，在由 Ⓓ+1 和 Ⓓ+2 指定的字软元件中存储运算结果。

COS Ⓢ = [Ⓓ 符号] [Ⓓ+1 整数部分] . [Ⓓ+2 小数部分]

(2) 由 Ⓢ 指定的值是一个在 0 度到 360 度之间的 BCD 值(单位为度)。

(3) 如果结果是正值，存储在 Ⓓ 中的运算结果的符号将为 “0”，如果是负值，这个符号将为 “1”。

(4) 存储在 Ⓓ+1 和 Ⓓ+2 中的运算结果为在 -1.000 和 1.000 之间的 BCD 值。

(5) 运算结果在小数点后第五位上四舍五入。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

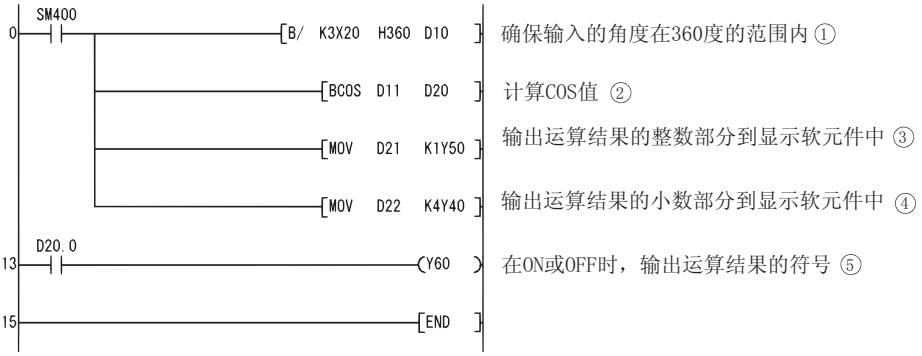
- 由 Ⓢ 指定的数据不是一个 BCD 值。 (出错代码： 4100)
- 由 Ⓢ 指定的数据不在 0 到 360 的范围之内。 (出错代码： 4100)
- 由 Ⓓ 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 下列程序计算从 X20 到 X2B 的 3 个 BCD 位指定的数据的余弦，并且输出结果的 1 个 BCD 位整数部分到 Y50 到 Y53，输出结果的 4 个 BCD 位的小数部分到从 Y40 到 Y4F。

如果运算结果为负，Y60 变为 ON。

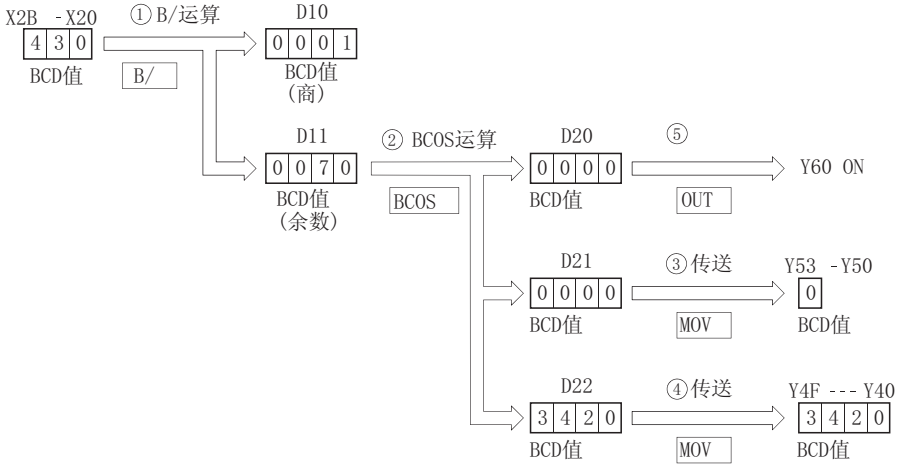
[梯形图模式]



[列表模式]

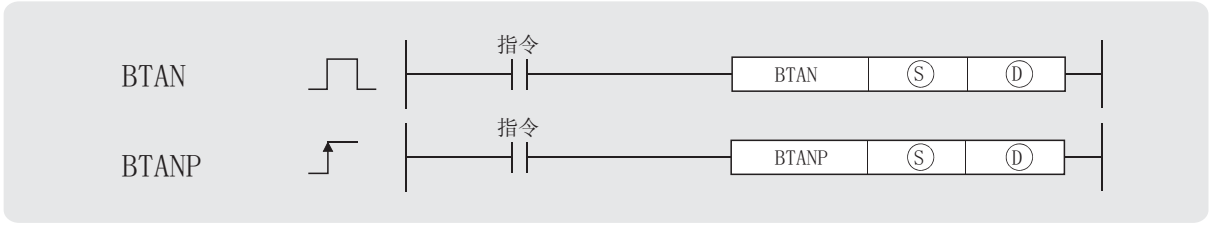
步	指令	软元件		
0	LD	SM400		
1	B/	K3X20	H360	D10
5	BCOS	D11	D20	
8	MOV	D21	K1Y50	
11	MOV	D22	K4Y40	
13	LD	D20.0		
14	OUT	Y60		
15	END			

[当由 X20 到 X2B 指定的值为 430 时的相关运算]



7.12.27 BCD 型 TAN 运算 (BTAN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：将要执行 TAN（正切）运算的数据，或者存储该类数据的软元件的起始软元件号（BCD 4 位）

Ⓓ：存储运算结果的软元件的起始软元件号（BCD 4 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○				○			---
Ⓓ	---	○				---			---

功 能

(1) 计算由 Ⓢ 指定的值（角度）的 TAN（正切）值，并且在由 Ⓓ 指定的字软元件中存储运算结果的符号，且在由 Ⓓ+1 和 Ⓓ+2 指定的字软元件中存储运算结果。

TAN Ⓢ = Ⓓ 符号　Ⓓ+1 整数部分 . Ⓓ+2 小数部分

(2) 由 Ⓢ 指定的值是一个在 0 度和 360 度之间的 BCD 值（单位为度）。

(3) 如果结果是一个正值，在 Ⓓ 中存储的运算结果的符号将为 “0”，如果结果是一个负值，该符号为 “1”。

(4) 存储在 Ⓓ+1 和 Ⓓ+2 中的运算结果是在从 -57.2901 到 57.2902 范围之内的 BCD 值。

(5) 运算结果在小数点后第五位上四舍五入。

运算错误

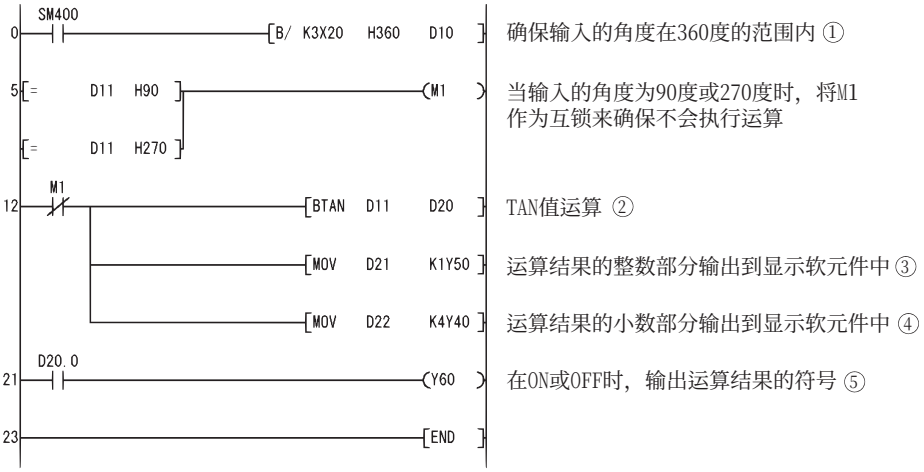
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的数据不是一个 BCD 值。（出错代码： 4100）
- 由 Ⓢ 指定的数据不在 0 到 360 的范围之内。（出错代码： 4100）
- 由 Ⓢ 指定的数据为 90 度或 270 度。（出错代码： 4100）
- 由 Ⓓ 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。）（出错代码： 4101）

程序示例

(1) 下列程序计算存储在 X20 到 X2B 上的 3 个 BCD 位的数值的正切，并且在从 Y50 到 Y53 的 4 个 BCD 位上存储结果的整数部分，在从 Y40 到 Y4F 的 4 个 BCD 位上存储存储结果的小数部分。如果运算结果为负，Y60 变为 ON。

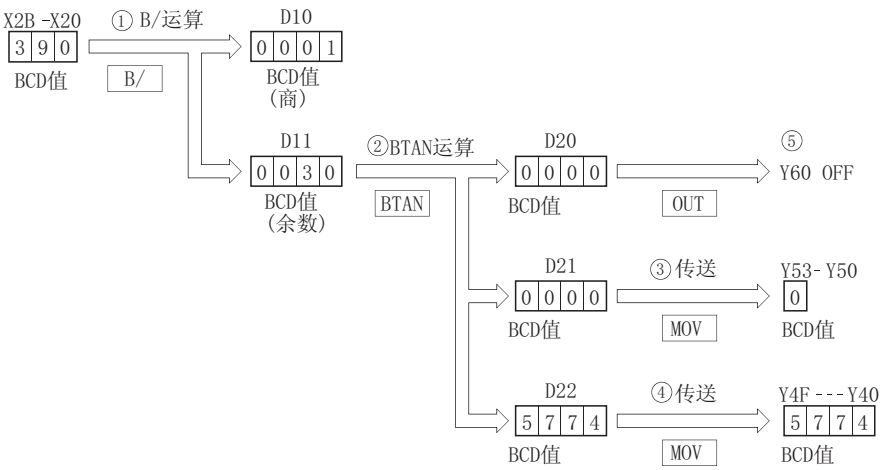
[梯形图模式]



[列表模式]

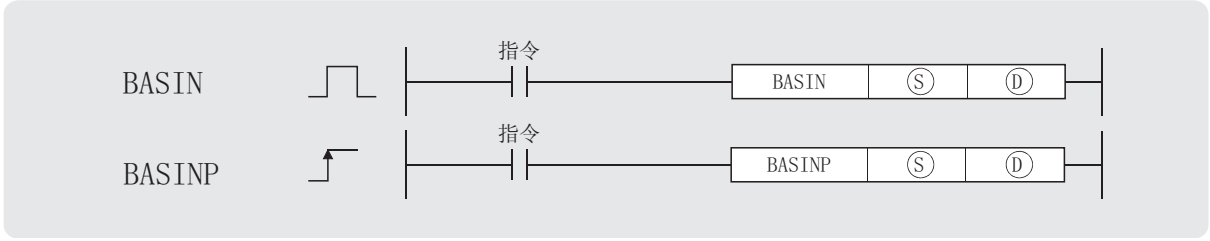
步	指令	软元件		
0	LD	SM400		
1	B/	K3X20	H360	D10
5	LD=	D11	H90	
8	OR=	D11	H270	
11	OUT	M1		
12	LDI	M1		
13	BTAN	D11	D20	
16	MOV	D21	K1Y50	
19	MOV	D22	K4Y40	
21	LD	D20.0		
22	OUT	Y60		
23	END			

[当 X20 到 X2B 的值指定为 390 时的相关运算]



7.12.28 BCD 型 SIN $^{-1}$ 运算 (BASIN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行 SIN^{-1}（反正弦）运算的数据，或存储该类数据的软元件号 (BCD 4 位)

Ⓓ : 存储运算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	○		---				---	---
Ⓓ	○	○		○				---	---

功 能

(1) 对由 Ⓢ 指定的值执行 SIN^{-1}（反正弦）运算，并且在 Ⓓ 指定的值执行 SIN -1（反正弦）运算，并且在

$$SIN^{-1}\ (\ \underset{\textcircled{S}}{\boxed{\text{符号}}}\ \underset{\textcircled{S}+1}{\boxed{\text{整数部分}}}.\underset{\textcircled{S}+2}{\boxed{\text{小数部分}}}\ ) = \textcircled{D}$$

(2) 在 Ⓢ 中设置运算数据的符号。
如果运算数据是正值，符号设置为 “0”，如果为负值，符号设置为 “1”。

(3) 在小数点和小数部分前面的部分分别以 BCD 值存储在 Ⓢ+1 和 Ⓢ+2 中（可以在 0 和 1.0000 中设置）。

(4) 存储在 Ⓓ 中的运算结果为 0 度到 90 度和 270 度到 360 度之间的 BCD 值（单位为度）。

(5) 计算结果是一个小数部分已经被四舍五入的值。

运算错误

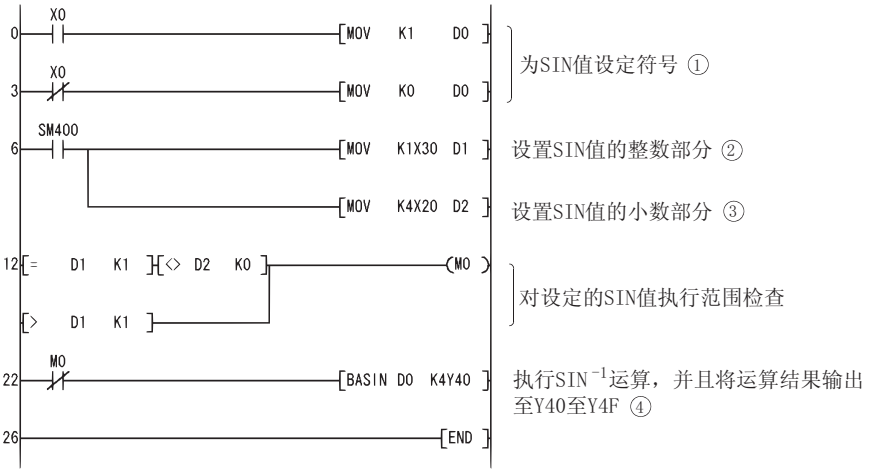
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的数据不是一个 BCD 值。 （出错代码： 4100）
- 由 Ⓢ 指定的数据不在从 -1.000 到 1.000 的范围之内。 （出错代码： 4100）
- 由 Ⓢ 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 下列程序对符号（当 X0 是 OFF 时为正，当 X0 是 ON 时为负），从 X30 到 X33 的 1 位 BCD 整数部分和从 X20 到 X2F 的 4 位 BCD 小数部分执行 SIN^{-1} 运算，并且输出从 Y40 到 Y4F 的 4 个 BCD 位中的计算出的角度。

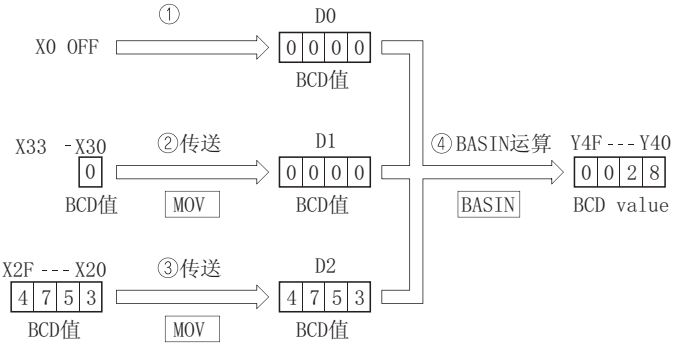
[梯形图模式]



[列表模式]

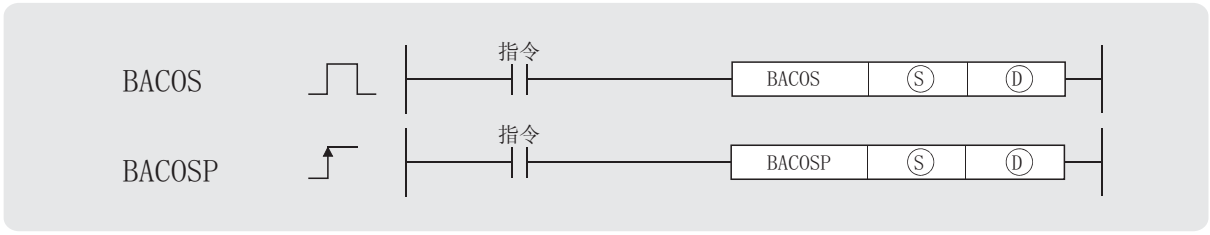
步	指令	软元件	
0	LD	X0	
1	MOV	K1	D0
3	LDI	X0	
4	MOV	K0	D0
6	LD	SM400	
7	MOV	K1X30	D1
10	MOV	K4X20	D2
12	LD=	D1	K1
15	AND<>	D2	K0
18	OR>	D1	K1
21	OUT	M0	
22	LDI	M0	
23	BASIN	D0	K4Y40
26	END		

[当 X20 到 X33 指定的值为 0.4753 时的相关运算]



7.12.29 BCD 型 COS $^{-1}$ 运算 (BACOS(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行 COS -1(反余弦)的数据，或存储该类数据的软元件的起始软元件号 (BCD 4 位)

Ⓓ : 存储运算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功 能

(1) 在由 Ⓢ 指定的值上执行 COS^{-1}(反余弦)运算，并且在 Ⓓ 指定的软元件中存储运算结果。

$$\cos^{-1}\left(\boxed{\text{Ⓢ 符号}}\ \boxed{\text{Ⓢ+1 整数部分}}.\boxed{\text{Ⓢ+2 小数部分}}\right)=\text{Ⓓ}$$

(2) 在 Ⓢ 中设置运算数据的符号。
如果运算数据是正值，符号设置为 “0”，如果为负值，符号设置为 “1”。

(3) 整数部分和小数部分前面的部分被分别以 BCD 值存储在 Ⓢ+1 和 Ⓢ+2 中。
(可以在 0 和 1.0000 中设置)

(4) 存储在 Ⓓ 中的运算结果是一个在 0 度到 180 度(单位为度)之间的 BCD 值。

(5) 计算结果是一个小数已经被四舍五入的值。

运算错误

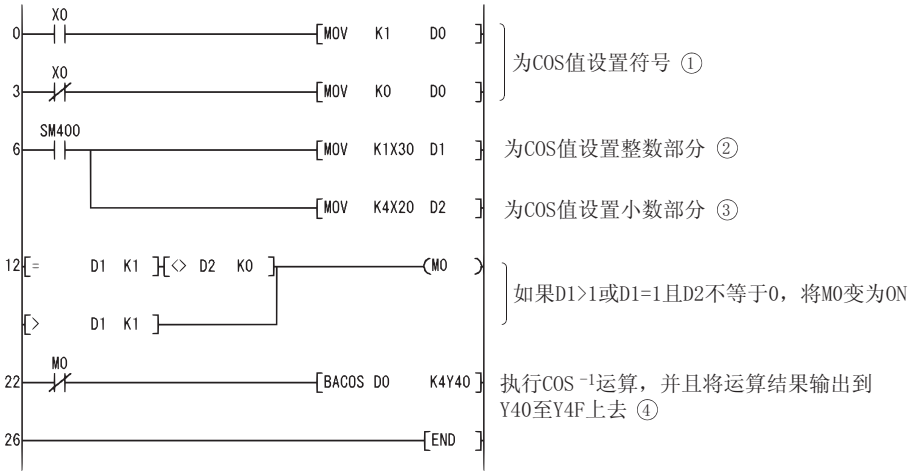
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的运算数据不是一个 BCD 值。 (出错代码： 4100)
- 由 Ⓢ 指定的运算数据不在从 -1.000 到 1.000 的范围之内。 (出错代码： 4100)
- 由 Ⓢ 指定的软元件超出了相应软元件的范围。
(只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 下列程序对符号（当 X0 是 OFF 时为正，当 X0 是 ON 时为负），从 X30 到 X33 的 1 位 BCD 整数部分和从 X20 到 X2F 的 4 位 BCD 小数部分执行 COS^{-1} 运算，并且输出从 Y40 到 Y4F 的 4 个 BCD 位中的计算出的角度。

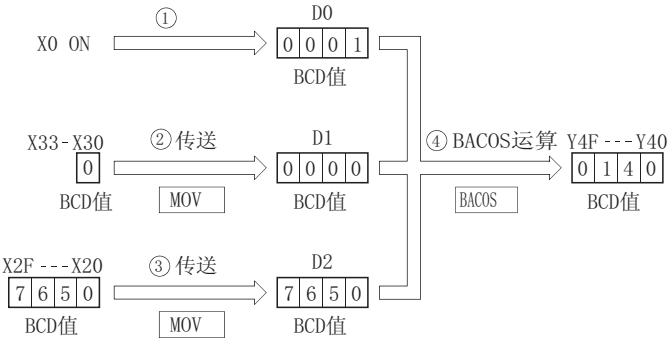
[梯形图模式]



[列表模式]

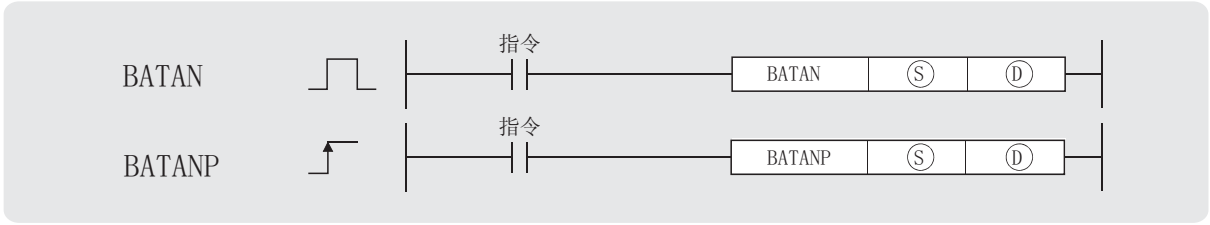
步	指令	软元件	
0	LD	X0	
1	MOV	K1	D0
3	LDI	X0	
4	MOV	K0	D0
6	LD	SM400	
7	MOV	K1X30	D1
10	MOV	K4X20	D2
12	LD=	D1	K1
15	AND<>	D2	K0
18	OR>	D1	K1
21	OUT	M0	
22	LDI	M0	
23	BACOS	D0	K4Y40
26	END		

[如果 X0 和 X20 到 X33 指定的值为 -0.7650 时的相关运算]



7.12.30 BCD 型 TAN $^{-1}$ 运算 (BATAN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 将要执行 TAN-1(反正切)运算的数据，或存储该类数据的软元件的起始软元件号 (BCD 4 位)

Ⓓ : 存储运算结果的软元件的起始软元件号 (BCD 4 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功 能

(1) 在由 Ⓢ 指定的值上执行 TAN^{-1}(反正切)运算，并且在 Ⓓ 指定的软元件中存储运算结果(角度)。

TAN $^{-1}$ (Ⓢ 符号 | Ⓢ+1 整数部分 . Ⓢ+2 小数部分) = Ⓓ

(2) 在 Ⓢ 中设置运算数据的符号。
如果运算数据是正值，符号设置为 “0”，如果为负值，符号设置为 “1”。

(3) 在小数点和小数部分前面的部分被分别以 BCD 值存储在 Ⓢ+1 和 Ⓢ+2 中。
(可以在从 0 到 9999.9999 的范围中设置此数值。)

(4) Op 存储在 Ⓓ 中的运算结果为 0 度到 90 度和 270 度到 360 度之间的 BCD 值(单位为度)。

(5) 计算结果是一个小数已经被四舍五入的值。

运算错误

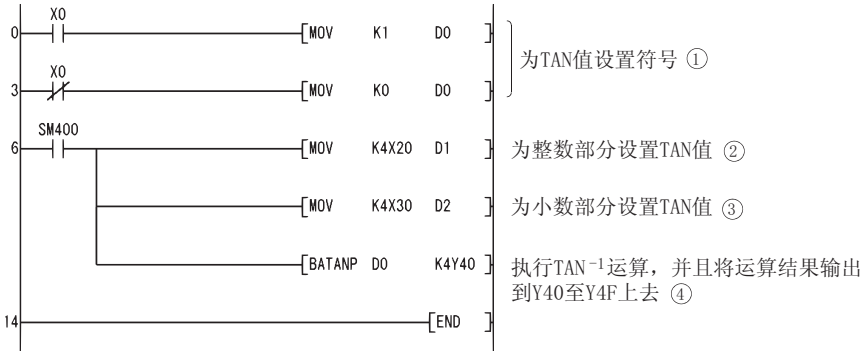
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的运算数据不是一个 BCD 值。 (出错代码： 4100)
- 由 Ⓢ 指定的软元件超出了相应软元件的范围。
(只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 下列程序对符号（当 X0 是 OFF 时为正，当 X0 是 ON 时为负），从 X20 到 X2F 的 4 位 BCD 整数部分和从 X30 到 X3F 的 4 位 BCD 小数部分执行 TAN^{-1} 运算，并且输出从 Y40 到 Y4F 的 4 个 BCD 位中的计算出的角度。

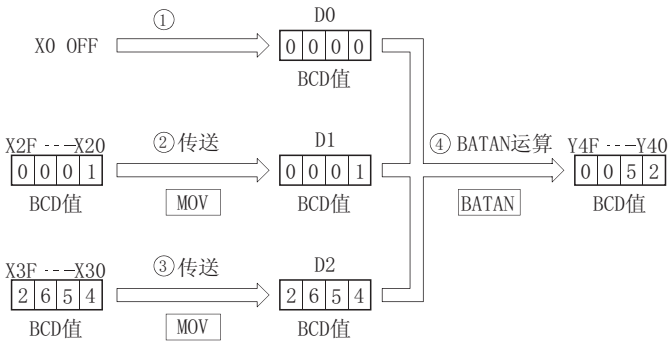
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	MOV	K1	D0
3	LDI	X0	
4	MOV	K0	D0
6	LD	SM400	
7	MOV	K4X20	D1
9	MOV	K4X30	D2
11	BATANP	D0	K4Y40
14	END		

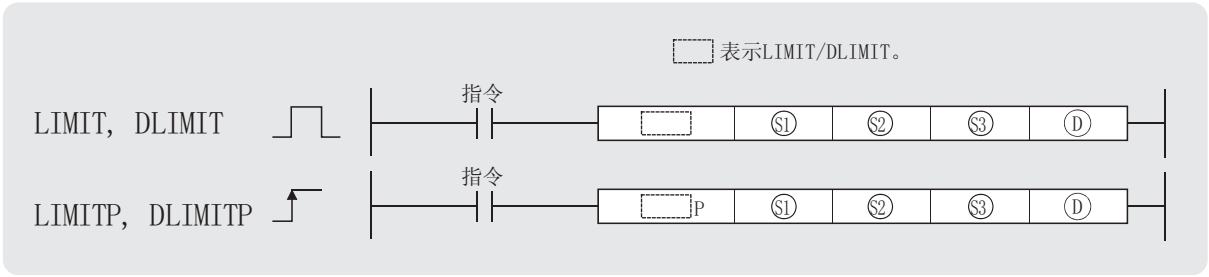
[当 X0 和 X20 到 X2F 的值指定为 1.2654 时的相关运算]



7.13 数据控制指令

7.13.1 BIN 16 位和 BIN 32 位数据的高低限控制 (LIMIT(P)、DLIMIT(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) : 低限值 (最小输出阈值) (BIN16/32 位)
(S2) : 高限值 (最大输出阈值) (BIN 16/32 位)
(S3) : 高限值和低限值控制的控制输入值 (BIN 16/32 位)
(D) : 存储由高限值和低限值控制所控制的输出值的软元件的起始号 (BIN 16/32 位)

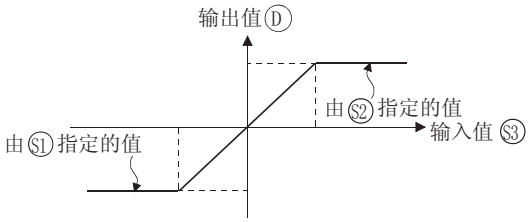
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)		○						○	---
(S2)		○						○	---
(S3)		○						○	---
(D)		○						---	---

功 能

LIMIT

(1) 检查由 (S3) 指定的输入值 (BIN 16 位值)，以确认其是否在由 (S1) 和 (S2) 指定的高限值和低限值之间，并且根据检查结果，控制在 (D) 中指定的软元件中存储的输出值。
按照下列方式控制输出值：

- 当 (S1) [低限值] > (S3) [输入值] ..... (S1) [低限值] → (D) [输出值]
- 当 (S2) [高限值] < (S3) [输入值] ..... (S2) [高限值] → (D) [输出值]
- 当 (S1) [低限值] ≦ (S3) [输入值] ≦ (S2) [高] ..... (S3) [输入值] (D) → [输出值]



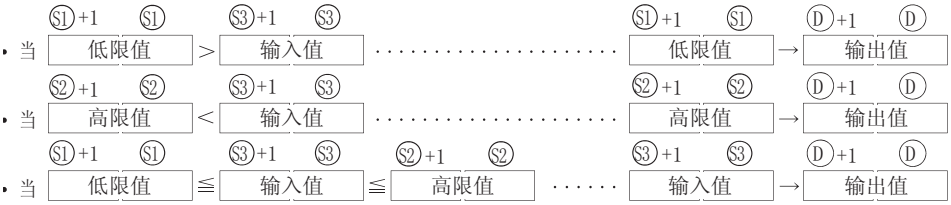
(2) 在 -32768 到 32767 范围之内的值可以在 (S1)、(S2) 和 (S3) 中指定。

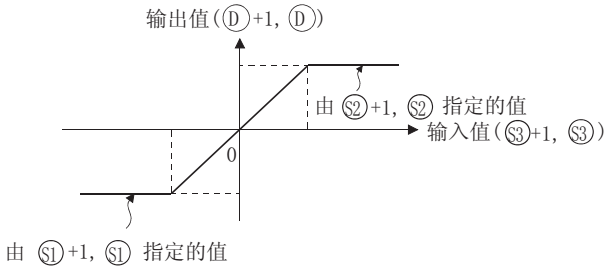
(3) 当仅基于高限值执行控制时，在 (S1) 中指定的低限值设被置为 “-32768”。

(4) 当仅基于低限值执行控制时，在 (S2) 中指定的高限值设置为 “32767”。

DLIMIT

(1) 根据由 ((S3)、(S3)+1) 指定的输入值 (BIN32 位值) 是否在由 ((S1)、(S1)+1) 和 ((S2)、(S2)+1) 指定的高限值和低限值之间，控制存储在由 ((D)、(D)+1) 指定的软元件中的输出值。





由 ((S1)、(S1)+1)、((S2)、(S2)+1) 或 ((S3)、(S3)+1) 指定的值是在 -2147483648 到 2147483647 的范围之内。

(2) 仅在高限值基础上执行控制，将由 ((S1)、(S1)+1) 指定的低限值设置为 “-2147483648”。

(3) 仅在低限值基础执行上控制，将由 ((S2)、(S2)+1) 指定的高限值设置为 “2147483647”。

运算错误

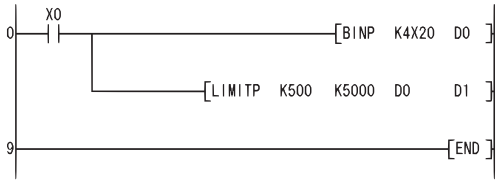
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 由 (S1) 指定的低限值比 (S2) 指定的高限值大。 (出错代码 : 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序对从 X20 到 X2F 的设置为 BCD 值的数据执行从 500 到 5000 的限制控制，并且在 D1 中存储结果。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	BINP	K4X20	D0		
4	LIMITP	K500	K5000	D0	D1
9	END				

[动作]

- 如果 500，则 D0 ＜ 500。

  例 D0＝400→D1＝500

- 当 500 ≦ D0 ≦ 5000 时，D1 变成 D0 的值。

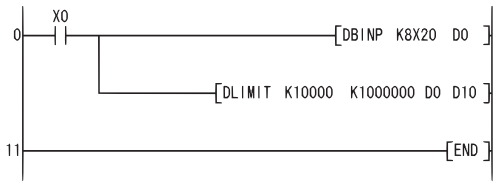
  例 D0＝1300→D1＝1300

- 当 5000 ＜ D0 时，D1 变成 5000。

  例 D0＝9600→D1＝5000

(2) 当 X0 为 ON 时，下列程序对从 X20 到 X3F 的设置为 BCD 值的数据执行从 10000 到 1000000 的限制控制。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	DBINP	K8X20	D0		
4	DLIMIT	K10000	K1000000	D0	D10
11	END				

[动作]

- 如果 (D1、D0) 小于 10000，则 (D11、D10) 变成 10000。

  例 (D1、D0)＝400→(D11、D10)＝10000

- 如果 10000 ≦ (D1、 D0) ≦ 1000000，则 (D11、D10) 变成 (D1、D0) 的值。

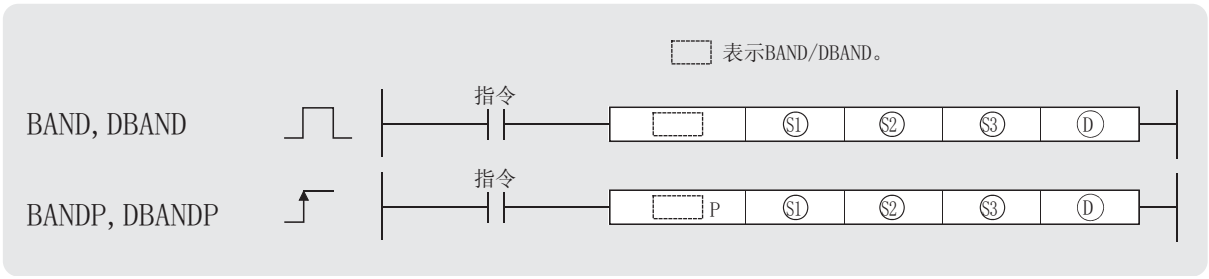
  例 (D1、D0)＝345678→(D11、D10)＝345678

- 如果 1000000 ＜ (D1、 D0)(D1、D0)，则 (D11、D10) 变成 1000000。

  例 (D1、D0)＝9876543→(D11、D10)＝1000000

7.13.2 BIN 16 位和 32 位死区控制 (BAND(P)、DBAND(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) : 死区的低限值 (无输出带) (BIN16/32 位)
(S2) : 死区的高限值 (无输出带) (BIN 16/32 位)
(S3) : 基于死区控制的控制输入值 (BIN 16/32 位)
(D) : 存储死区控制的控制输出值的软元件的起始号 (BIN 16/32 位)

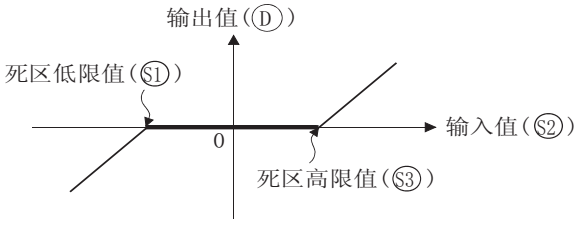
设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
(S1)		○						○	---
(S2)		○						○	---
(S3)		○						○	---
(D)		○						---	---

功 能

BAND

(1) 根据由 (S3) 指定的输入值 (BIN 16 位值) 是否在由 (S1) 和 (S2) 指定的高限值和低限值死区控制的范围之内，控制存储在由 (D) 指定的软元件中的输出值。
按照下列方式控制输出值 :

- (S1) 低限值 > (S3) 输入值 ..... (S3) 输入值 - (S1) 低限值 → (D) 低限值
- (S2) 高限值 < (S3) 输入值 ..... (S3) 输入值 - (S2) 高限值 → (D) 输出值
- (S1) 低限值 ≦ (S3) 输入值 ≦ (S2) 高限值　　0 → (D) 输出值



(2) 能由 (S1)、(S2) 和 (S3) 指定的值在从 -32768 到 32767 的范围之内。

(3) 存储在 ⓓ 中的输出值是一个带符号的 16 位 BIN 值。
因此，如果运算结果超出了从 -32768 到 32767 的范围，将会发生下列情况：

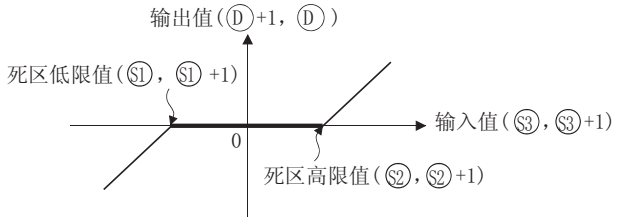
当：
- 死区低限值 ⓢ1 10
- 输入值 ⓢ3 ..... −32768

输出值=−32768 − 10= 8000_H − A_H = $7FF6_H$ = 32758

DBAND

(1) 根据由 (ⓢ3、 ⓢ3+1) 指定的输入值 (BIN 32 位值) 是否在由 (ⓢ1、 ⓢ1+1) 和 (ⓢ2、 ⓢ2+1) 指定的高限值和低限值死区控制范围之中，来控制由 ⓓ 指定的软元件中的输出值。
按照以下方式控制输出值：

- ⓢ1+1 ⓢ1 低限值 > ⓢ3+1 ⓢ3 输入值 ····· ⓢ3+1 ⓢ3 输入值 − ⓢ1+1 ⓢ1 低限值 → ⓓ+1 ⓓ 输出值
- ⓢ2+1 ⓢ2 高限值 < ⓢ3+1 ⓢ3 输入值 ····· ⓢ3+1 ⓢ3 输入值 − ⓢ2+1 ⓢ2 高限值 → ⓓ+1 ⓓ 输出值
- ⓢ1+1 ⓢ1 低限值 ≦ ⓢ3+1 ⓢ3 输入值 ≦ ⓢ2+1 ⓢ2 高限值 ····· 0 → ⓓ+1 ⓓ 输出值



(2) 由 (ⓢ1、ⓢ1+1)、 (ⓢ2、ⓢ2+1)、 或 (ⓢ3、ⓢ3+1) 指定的值在 -2147483648 到 2147483647 的范围中。

(3) 存储在 ⓓ、ⓓ+1 中的输出值是一个带符号的 32 位 BIN 值。
因此，如果运算结果超出了从 -2147483648 到 2147483647 的范围，将会发生下列情况：

当：
- 死区低限值 (ⓢ1、 ⓢ1+1) 1000
- 输入值 (ⓢ3、 ⓢ3+1) ..... −2147483648

输出值 =−2147483648−1000=80000000_H−$000003E8_H$

=$7FFFFC18_H$=2147482648

运算错误

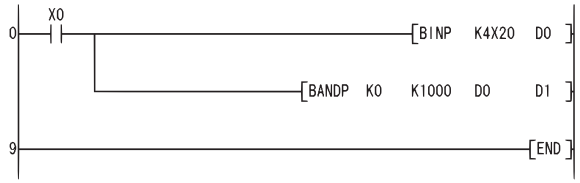
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 ⓢ1 指定的低限值比 ⓢ2 指定的高限值大。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序通过 0 和 1000 的低限和高限对由从 X20 到 X2F 的 BCD 值设置的数据执行死区控制，并且在 D1 中存储结果。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	BINP	K4X20	D0		
4	BANDP	K0	K1000	D0	D1
9	END				

[动作]

- 如果 0≦D0≦1000，则将 “0”存储在 D1) 中。

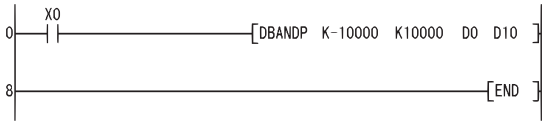
  例　D0＝500→D1＝0

- 如果 1000 ＜ D0，则 (D0)-1000 的值存储在 D1) 中。

  例　D0＝7000→D1＝6000

(2) 当 X0 为 ON 时，下列程序通过 -10000 和 10000 的低限和高限对 D0 及 D1 中设置的数据执行死区控制，并且在 D10 和 D11 中存储结果。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	DBANDP	K-10000	K10000	D0	D10
8	END				

[动作]

- 如果 (D1、 D0)＜(—10000)，(D1，D0)-(-10000) 的值存储在 (D11，D10) 中。

  例　(D1、 D0)＝－12345→(D11、 D10)＝－2345

- 如果－10000 ≦ (D1、 D0) ≦ 10000，值 0 存储在 (D11，D10) 中。

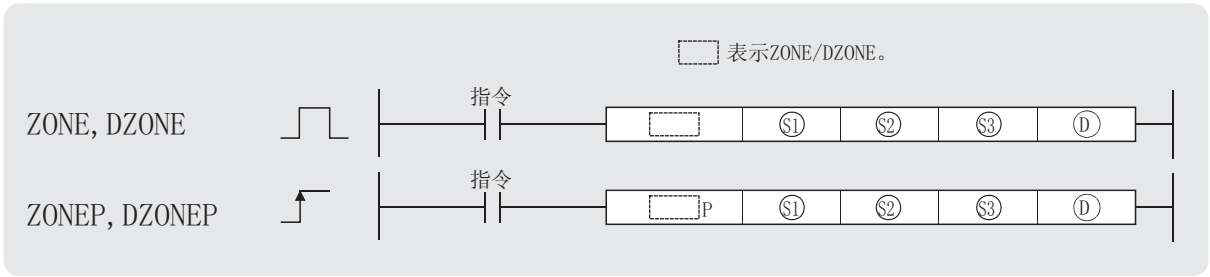
  例　(D1、 D0)＝6789→(D11、 D10)＝0

- 如果 10000 ＜ (D1、 D0)，(D1，D0)-10000 的值存储在 (D11，D10) 中。

  例　(D1、 D0)＝50000→(D11、 D10)＝40000

7.13.3 BIN 16 位和 BIN 32 位数据的区域控制 (ZONE(P)、DZONE(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) : 加到输入值上的负偏差值 (BIN 16/32 位)

(S2) : 加到输入值上的正偏差值 (BIN 16/32 位)

(S3) : 用于执行区域控制的输入值 (BIN 16/32 位)

(D) : 存储基于区域控制的输出值的软元件的起始号 (BIN 16/32 位).

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S1)				○				○	---
(S2)				○				○	---
(S3)				○				○	---
(D)				○				---	---

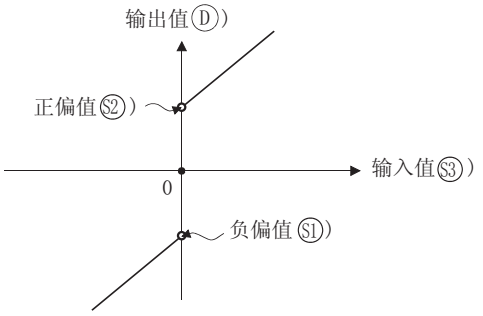
功 能

ZONE

(1) 将 (S1) 或 (S2) 指定的偏差值与由 (S3) 指定的输入值相加，并且在由 (D) 指定软元件号的软元件中存储结果。

按照下列方式计算偏差值：

- (S3) 输入值 < 0 ..... (S3) 输入值 + (S1) 负偏差值 → (D) 输出值
- (S3) 输入值 = 0 ..... 0 → (D) 输出值
- (S3) 输入值 > 0 ..... (S3) 输入值 + (S2) 负偏差值 → (D) 输出值



(2) 由 (S1)、(S2) 和 (S3) 指定的值是在 -32768 到 32767 的范围中的。

(3) 存储在 ⓓ 中的输入值是带符号的 16 位 BIN 值。

$$\text{当：}\begin{cases}\text{负偏差值 (S1)} \cdots\cdots\cdots\cdots\cdots\cdots\cdots -100\\ \text{输入值 (S3)} \cdots\cdots\cdots\cdots\cdots\cdots\cdots -32768\end{cases}$$

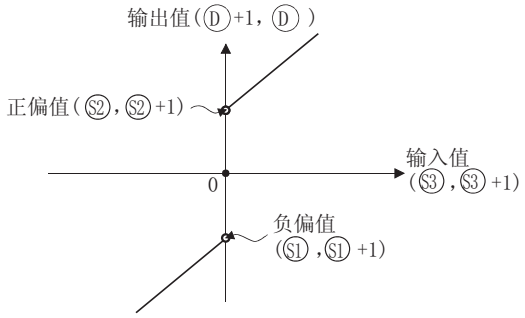
$$\text{输出值} = -32768 + (-100) = 8000_H + FF9C = 7F9C_H = 32668$$

DZONE

(1) 将由 (S1)、(S1)+1) 或 (S2)、(S2)+1) 指定的偏差值与由 (S3)、(S3)+1) 指定的输入值相加，并且在 (D)、(D)+1) 指定软元件号的软元件中存储结果。

偏差值的加法运算执行如下：

- (S3)+1 (S3) 输入值 ＜0 ····· (S3)+1 (S3) 输入值 + (S1)+1 (S1) 正偏差值 → (D)+1 (D) 输出值
- (S3)+1 (S3) 输入值 ＝0 ····· 0 → (D)+1 (D) 输出值
- (S3)+1 (S3) 输入值 ＞0 ····· (S3)+1 (S3) 输入值 + (S2)+1 (S2) 负偏差值 → (D)+1 (D) 输出值



(2) 由 (S1)、S1)+1)、(S2)、S2)+1) 或 (S3)、S3)+1) 指定的值是在从 -2147483648 到 2147483647 的范围之内的值。

(3) 存储在 (D)、D)+1) 中的值是一个带符号的 32 位 BIN 值。

因此，如果运算结果超出了从 -2147483648 到 2147483647 的范围，将会发生下列情况：

$$\text{当：}\begin{cases}\text{负偏差值 (S1)、S1)+1)} \cdots\cdots\cdots\cdots\cdots -1000\\ \text{输入值 (S3)、S3)+1)} \cdots\cdots\cdots\cdots\cdots -2147483648\end{cases}$$

$$\text{输出值} = -2147483648 + (-1000) = 80000000_H + FFFFFC18_H$$
$$= 7FFFFC18 = 2147482648.$$

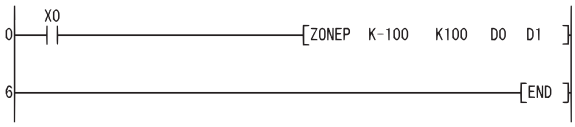
运算错误

(1) 没有与 ZONE(P) 或 DZONE(P) 指令相关的运行错误。

程序示例

(1) 当 X0 为 ON 时，下列程序对在 D0 上设置的数据执行 -100 到 100 的正偏值和负偏值的区域控制，并且在 D1 中存储控制结果。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	ZONEP	K-100	K100	D0	D1
6	END				

[动作]

- 如果 D0 ＜ 0，则 (D0)+(-100) 的值存储在 D1 中。

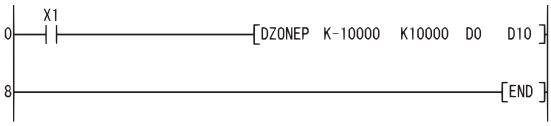
  例　D0＝－200→D1＝－300

- 如果 D0=0，值 0 存储在 D1 中。
- 如果 0 ＜ D0，(D0)+100 的值存储在 D1 中。

  例　D0＝700→D1＝800

(2) 当 X1 为 ON 时，下列程序对在 D0 和 D1 上设置的数据执行 -10000 到 10000 的正偏值和负偏值的区域控制，并且在 D10 和 D11 中存储控制结果。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X1			
1	DZONEP	K-10000	K10000	D0	D10
8	END				

[动作]

- 如果 (D1、 D0) ＜ 0，在 (D11、D10) 中存储 (D1、D0)+(-10000) 的值。

  例　(D1、D0)＝－12345→(D11、D10)＝－22345

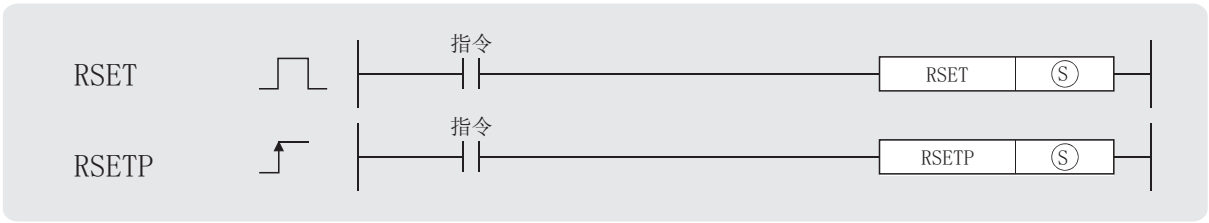
- 如果 (D1、 D0) ＝ 0，在 (D11、D10) 中存储值 0。
- 如果 0 ＜ (D1、 D0)，在 (D11、D10) 中存储 (D1、D0)+10000 的值。

  例　(D1、D0)＝50000→(D11、D10)＝60000

7.14 文件寄存器切换指令

7.14.1 切换文件寄存器号 (RSET(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 用于更改块号码的块号码数据或存储块号码数据的软元件号 (BIN 16 位)

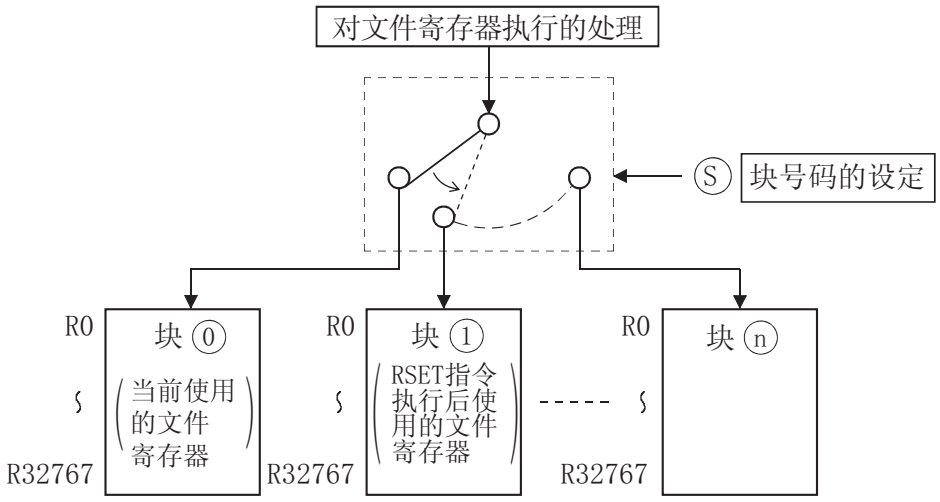
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○								---

功 能

(1) 将程序中正在使用的文件寄存器的块号码改变为由 Ⓢ 指定的软元件中存储的块号码。
在块号码改变之后，所有由顺控程序使用的文件寄存器均按块号码改变后的文件寄存器进行处理。

例如

当从 0 号块切换到 1 号块时。



要 点

将文件寄存器 (R) 指定为刷新软元件，并通过 RSET 指令切换文件寄存器的块号时，应注意以下限制。
关于文件寄存器的限制，请参见 3.10 节。

运算错误

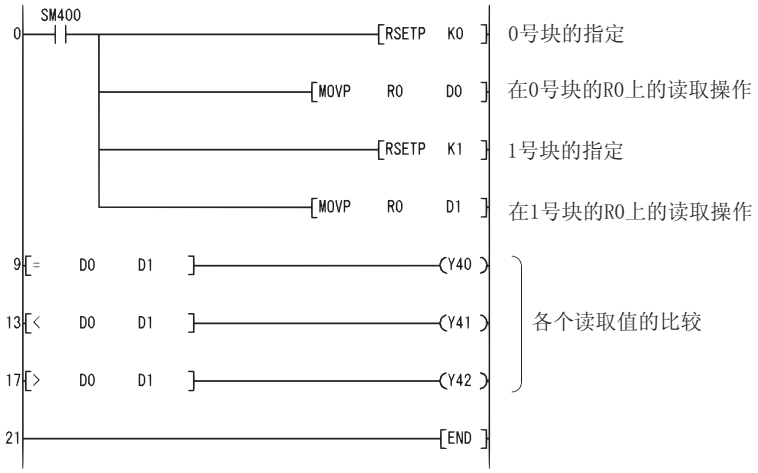
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的块号码不存在。 (出错代码： 4100)
- 由 Ⓢ 指定的块号码没有文件寄存器。 (出错代码： 4101)

程序示例

(1) 下列程序比较第 0 号块和第 1 号块的 R0。

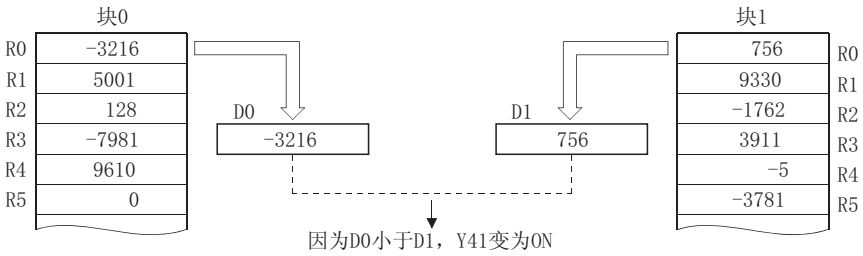
[梯形图模式]



[列表模式]

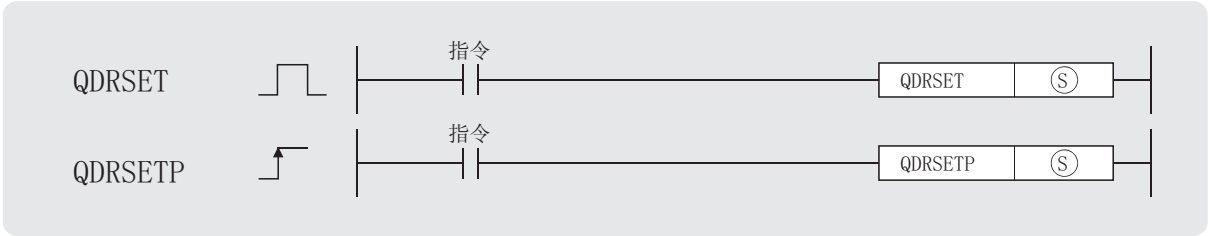
步	指令	软元件	
0	LD	SM400	
1	RSETP	K0	
3	MOVP	R0	D0
5	RSETP	K1	
7	MOVP	R0	D1
9	LD=	D0	D1
12	OUT	Y40	
13	LD<	D0	D1
16	OUT	Y41	
17	LD>	D0	D1
20	OUT	Y42	
21	END		

[动作]



7.14.2 文件寄存器使用的设置文件 (QDRSET(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 被设置的文件寄存器的驱动号 / 文件名的字符串数据，或存储字符串数据的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

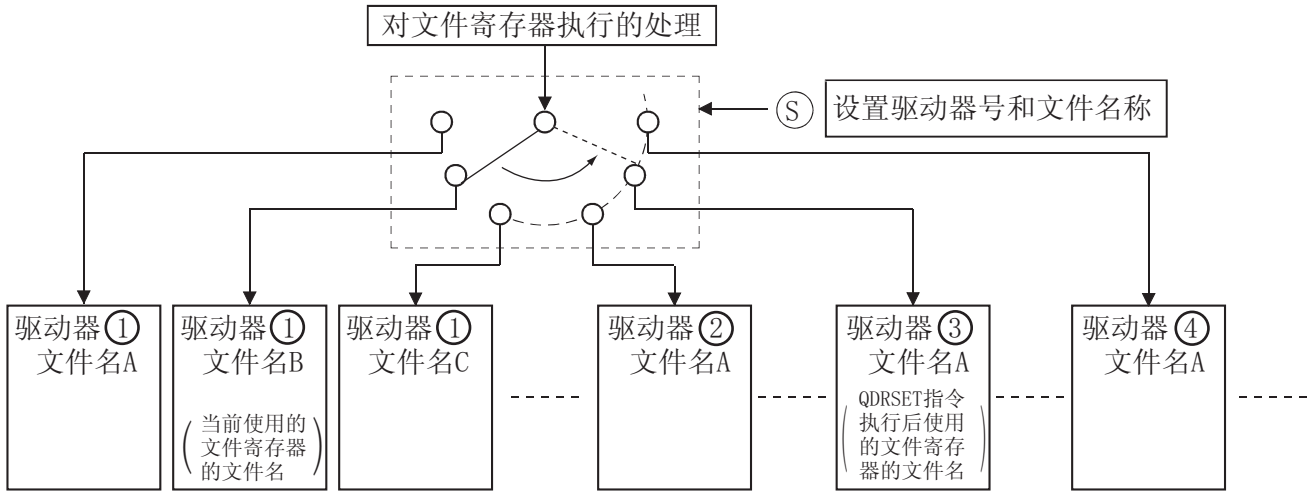
(1) 将程序使用的文件寄存器的文件名改变为由 Ⓢ 指定的软元件中存储的文件名。

在文件名改变之后，所有正在被顺控程序使用的文件寄存器均被按重命名的文件的 0 号块的文件寄存器进行处理。

通过 RSET 指令执行块号码切换。

例如

当从驱动器 1/ 文件名 B 切换到驱动器 3/ 文件名 A 时。



(2) 驱动器号能够在 1 到 4 的范围内指定。
驱动器号不能指定为驱动 0(程序内存 / 内部内存)

(3) 不需要指定文件的扩展名 (.QDR)。

(4) 通过为文件名指定 NULL 字符 (00_H) 能够删除文件名设定。

(5) 即使驱动器号和文件名已经在参数中被指定，用该指令指定的文件名也将有更高的优先级。

要 点

1. 即使通过 QDRSET 指令改变了文件名，当 CPU 模块从 STOP 切换到 RUN 时，文件名将恢复为参数指定的名称。
如果希望在 CPU 模块从 STOP 变为 RUN 后继续保持通过 QDRSET 指令更改的文件名，应使用特殊继电器 SM402 执行 QDRSET 指令，此特殊继电器当 CPU 从 STOP 进入 RUN 模式时，将会 ON 一个扫描周期。
2. 将文件寄存器指定为刷新软元件时，请不要用 QDRSET 指令改变文件寄存器的文件名。
关于文件寄存器的限制，请参见 3.10 节。

运算错误

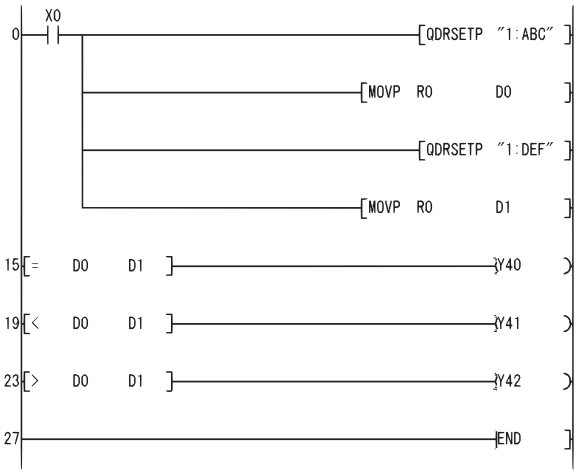
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 在由 ⓢ 指定的驱动器号中文件名不存在。 (出错代码 : 2410)

程序示例

(1) 下列程序将驱动器 1 的 ABC 的 R0 与驱动器 1 的 DEF 的 R0 进行比较。

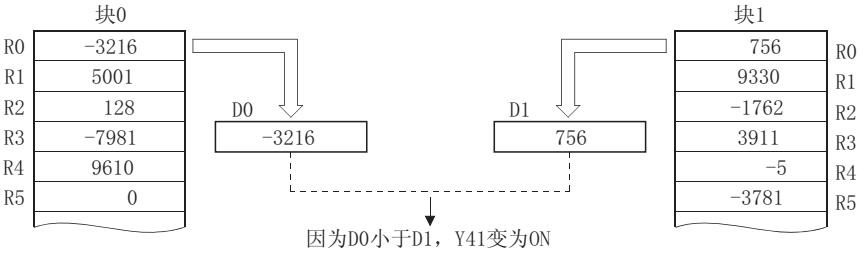
[梯形图模式]



[列表模式]

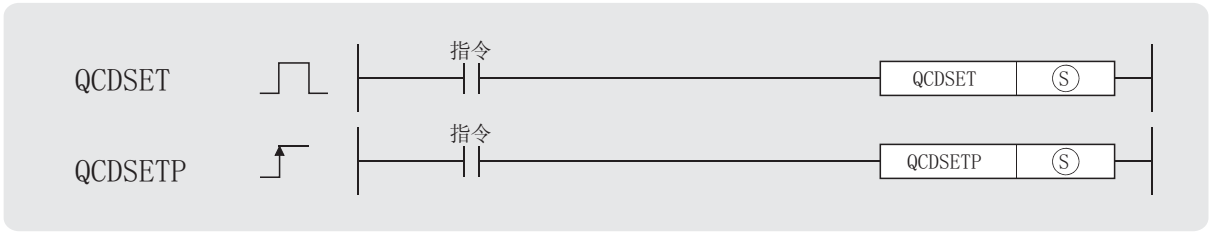
步	指令	软元件	
0	LD	X0	
1	QDRSETP	"1:ABC"	
6	MOVP	R0	D0
8	QDRSETP	"1:DEF"	
13	MOVP	R0	D1
15	LD=	D0	D1
18	OUT	Y40	
19	LD<	D0	D1
22	OUT	Y41	
23	LD>	D0	D1
26	OUT	Y42	
27	END		

[动作]



7.14.3 注释的文件设置 (QCDSET(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：设置注释文件的驱动器号 / 文件名的字符串数据，或存储字符串数据的软元件的起始号（字符串）

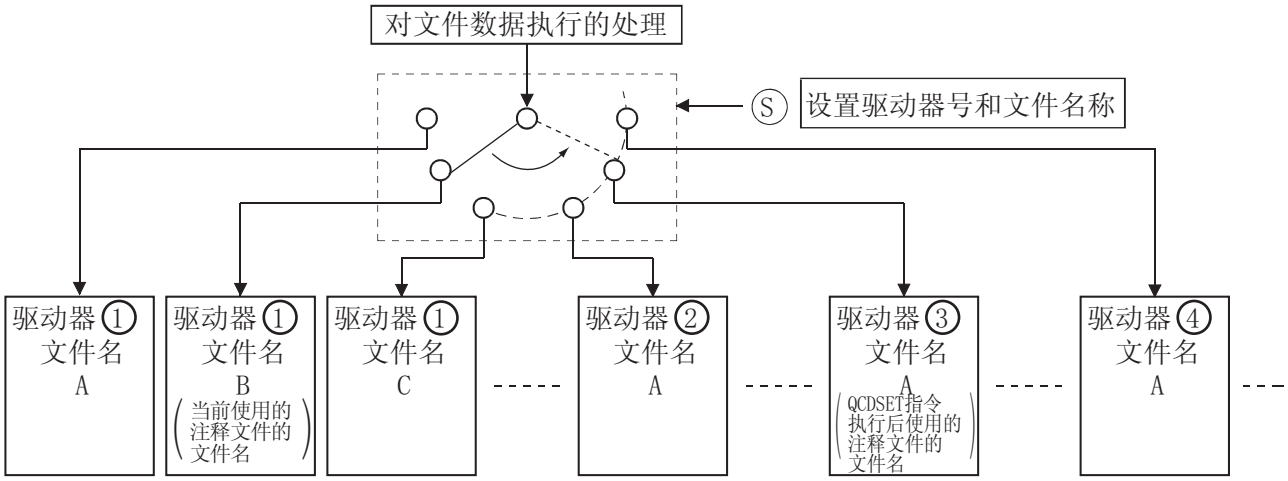
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

(1) 将程序中使用的注释文件的文件名改变成由 Ⓢ 指定的软元件中存储的文件名。
文件名改变之后，正在被顺控程序使用的注释数据将按改变后的文件名的注释数据进行处理。

例如

当从驱动器 1/ 文件名 B 切换到驱动器 3/ 文件名 A 时。



(2) 驱动器号能够在 1 到 4 的范围内指定。 *1
驱动器号不能指定为驱动器 0(程序内存 / 内部内存)。

*1: 不能为 QCPU 指定驱动器 3。

(3) 不需要指定文件的扩展名 (.QCD)。

(4) 可以通过为文件名指定 NULL 字符 (00H) 来删除文件名设定。

(5) 即使驱动器号和文件名已经在参数指定了，用该指令指定的文件名将有更高的优先级。

(6) 对于通用型 CPU，SM721 为 ON 时不能执行本指令。如果执行，将被作为无处理。

要 点

即使通过 QCDSET 指令改变了文件名，当 CPU 模块从 STOP 切换到 RUN 时，文件名将恢复为参数指定的名称。
如果希望在 CPU 模块从 STOP 变为 RUN 后继续保持通过 QDRSET 指令更改的文件名，应使用特殊继电器 SM402 执行 QDRSET 指令，此特殊继电器当 CPU 从 STOP 进入 RUN 模式时，将会 ON 一个扫描周期。

运算错误

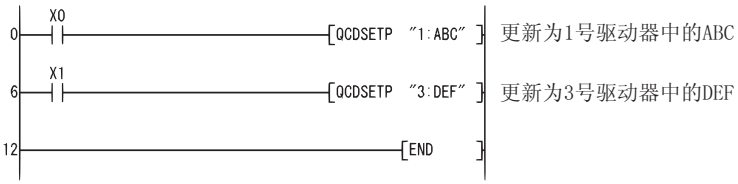
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 在由 Ⓢ 指定的驱动器号中文件名不存在。 (出错代码 : 2410)

程序示例

(1) 当 X0 为 ON 时，下列程序将目标文件切换为 0 号驱动中文件名为 ABC.QCD 的文件，并且当 X1 为 ON 时切换为 1 号驱动中文件名为 DEF.QCD 的文件。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	QCDSETP	"1:ABC"
6	LD	X1
7	QCDSETP	"3:DEF"
12	END	

注意事项

(1) 只对于通用型 QCPU，当 SM721(文件访问执行中) 为 ON 时，即使本指令的执行指令为 ON，也不执行本指令。
应在 SM721 为 OFF 时执行本指令。

7.15 时钟指令

7.15.1 读时钟数据 (DATERD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当使用 QCPU 时

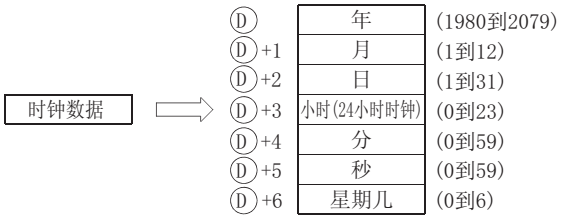


D ：存储读取的时钟数据的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
D	---	○		---					

功 能

(1) 从 CPU 模块的时钟软元件中读取 “年、月、日、分、秒和星期几”，并以 BIN 值存储在 D 指定的软元件之后。



(2) 在 D 中存储的 “年”是四位数表示的年。

(3) 在 D+6 中存储的 “星期几”为 0 到 6，代表星期日到星期六。

星期几	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存储的数据	0	1	2	3	4	5	6

(4) 闰年自动补偿。

运算错误

(1) 在下列情况下将发生运算错误，出错标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

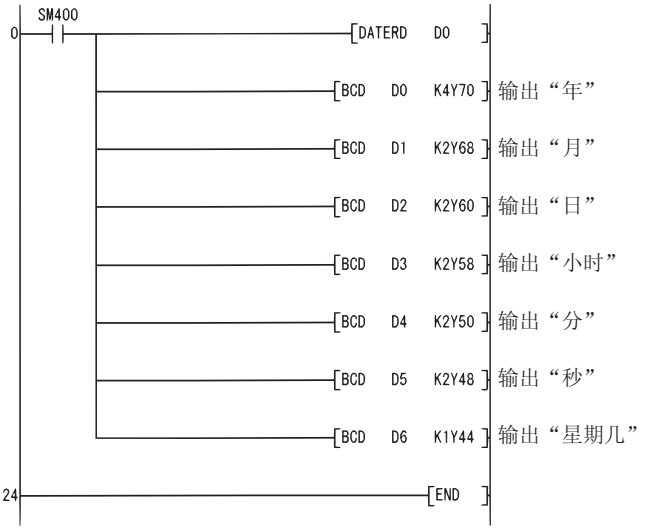
・由 D 指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 以下程序以 BCD 值输出下述时钟数据：

年 ..... Y70 到 Y7F
月 ..... Y68 到 Y6F
日 ..... Y60 到 Y67
小时 ..... Y58 到 Y5F
分钟 ..... Y50 到 Y57
秒 ..... Y48 到 Y4F
星期几 ... Y44 到 Y47

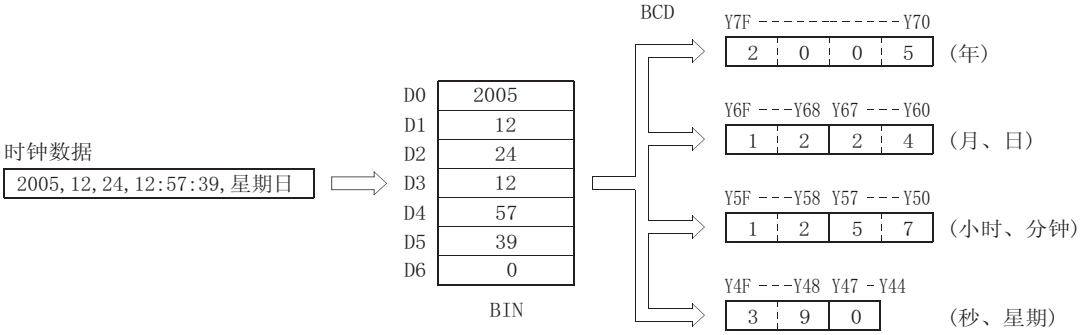
[梯形图模式]



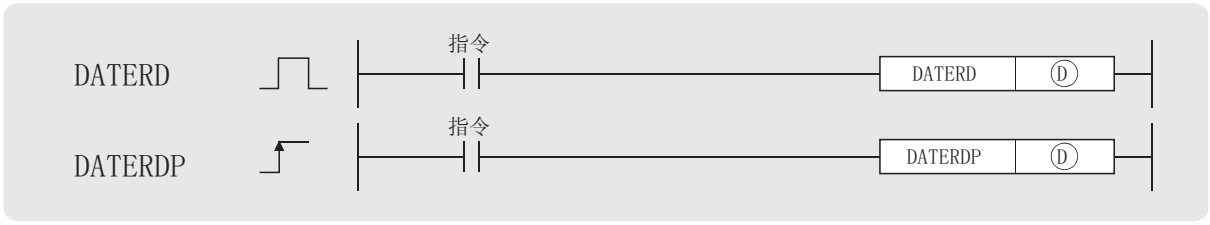
[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DATERD	D0	
3	BCD	D0	K4Y70
6	BCD	D1	K2Y68
9	BCD	D2	K2Y60
12	BCD	D3	K2Y58
15	BCD	D4	K2Y50
18	BCD	D5	K2Y48
21	BCD	D6	K1Y44
24	END		

[动作]



2 当使用 QnACPU/Q4ARCPU 时

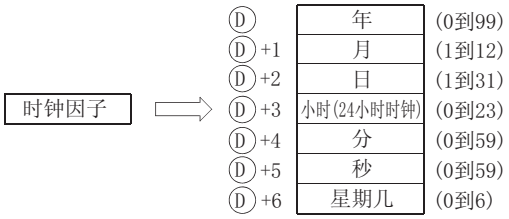


Ⓓ : 存储读取的时钟数据的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	---	○		---					

功 能

(1) 从 QnACPU 的时钟软元件中读取 “年、月、日、分、秒和星期几”，并以 BIN 值存储在 Ⓓ 指定的软元件之后。



(2) 在 Ⓓ 中存储的 “年”是年份的低 2 位。
例如，1995 年存储为 “95”。

(3) 在 Ⓓ+6 中存储的 “星期几”为 0 到 6，代表星期日到星期六。

星期几	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存储的数据	0	1	2	3	4	5	6

(4) 闰年自动补偿。

运算错误

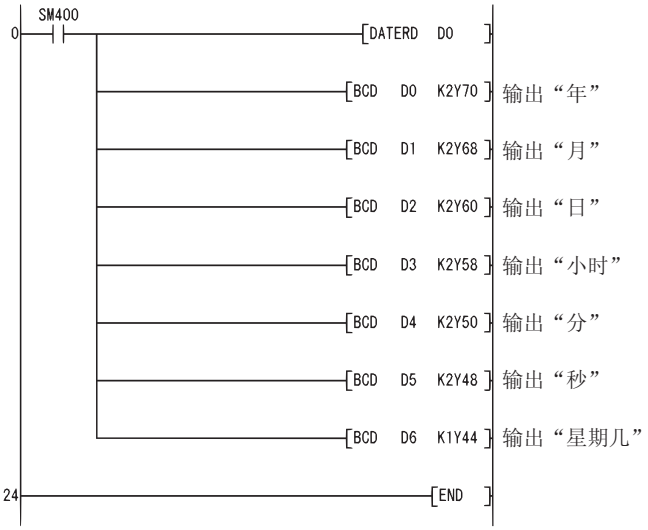
(1) 没有与 DATERD(P) 指令相关的运行错误。

程序示例

(1) 以下程序以 BCD 值输出下述时钟数据：

年..... Y70 到 Y77
月..... Y68 到 Y6F
日..... Y60 到 Y67
小时..... Y58 到 Y5F
分钟..... Y50 到 Y57
秒..... Y48 到 Y4F
星期几... Y44 到 Y47

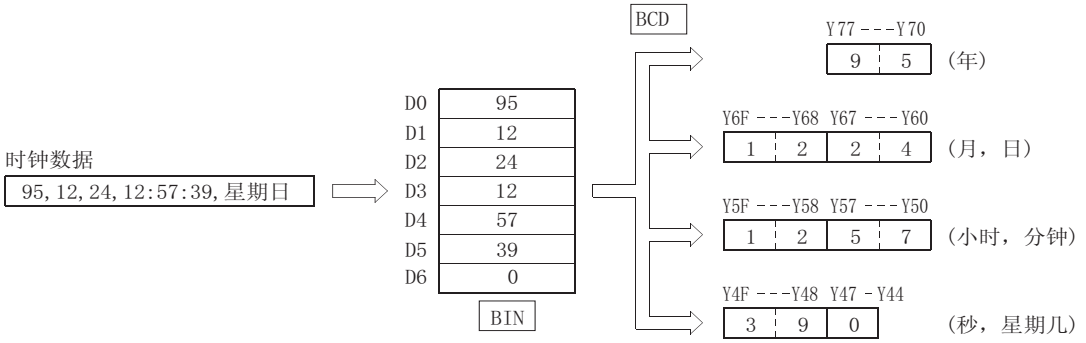
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	DATERD	D0	
3	BCD	D0	K2Y70
6	BCD	D1	K2Y68
9	BCD	D2	K2Y60
12	BCD	D3	K2Y58
15	BCD	D4	K2Y50
18	BCD	D5	K2Y48
21	BCD	D6	K1Y44
24	END		

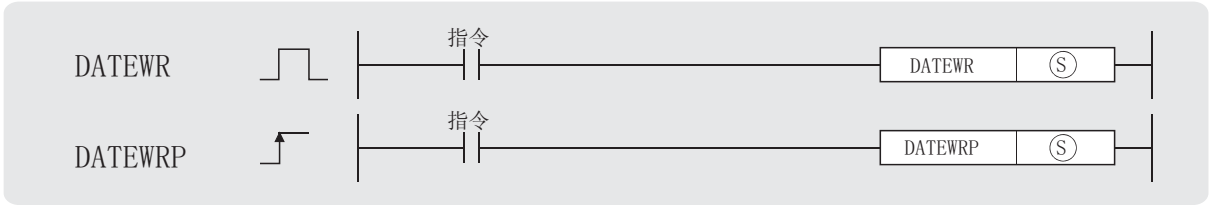
[动作]



7.15.2 写时钟数据 (DATEWR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

1 当使用 QCPU 时



Ⓢ : 存储写入到时钟因子中的时钟数据的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---					

功 能

(1) 存储在由 Ⓢ 指定号码的软元件之后的时钟数据，被写到 CPU 模块时钟软元件中。

Ⓢ	年
Ⓢ+1	月
Ⓢ+2	日
Ⓢ+3	小时
Ⓢ+4	分
Ⓢ+5	秒
Ⓢ+6	星期几

→ 时钟软元件

(2) 每一个项目均被设置为 BIN 值。

(3) 在 Ⓢ 中的 “年” 是由在 1980 到 2079 年之间的 4 位数进行设置的。
(如果年份设定使用了特殊纪元名，闰年将无法自动调整。)

(4) Ⓢ+1 指定的 “月” 是用从 1 到 12 的值 (1 月到 12 月) 来表示的。

(5) Ⓢ+2 指定的 “日” 是用从 1 到 31 的值进行设置的。

(6) Ⓢ+3 指定的 “小时” 是用从 0 到 23 的值 (使用 24 小时制时钟，从 0 小时到 23 小时) 来表示的。(使用 24 小时制时钟)

(7) Ⓢ+4 指定的 “分钟” 是用从 0 到 59 的值进行设置的。

(8) Ⓢ+5 指定的 “秒” 是用从 0 到 59 的值进行设置的。

(9) Ⓢ+6 指定的 “星期几” 是用从 0 到 6 的值 (星期日到星期六) 来表示的。

星期几	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存储的数据	0	1	2	3	4	5	6

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

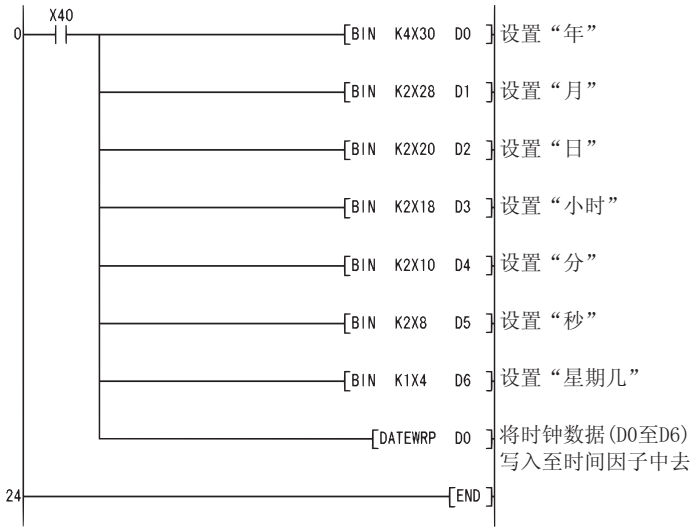
- 数据的个别项目设置在设定范围之外。 （出错代码： 4100）
- 由 ⓢ 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 当 X40 为 ON 时，下列程序以 BCD 值将以下时钟数据写入到时钟软元件中。

年..... X30 到 X3F　　小时..... X18 到 X1F
月..... X28 到 X2F　　分钟..... X10 到 X17
日..... X20 到 X27　　秒..... X8 到 XF
星期几..... X4 到 X7

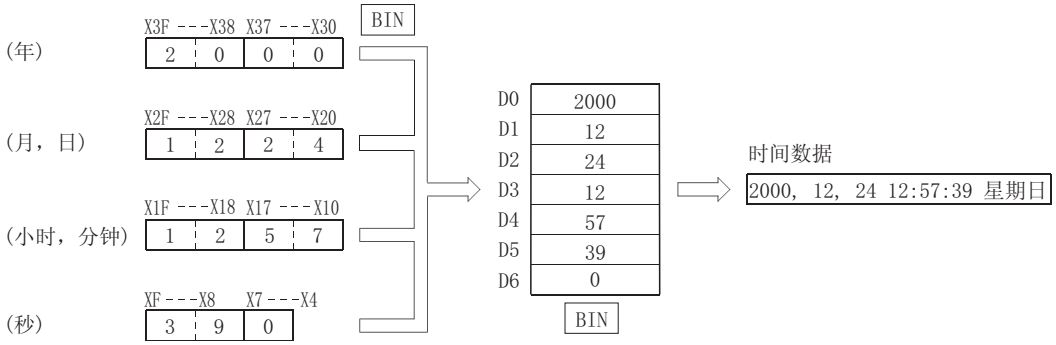
[梯形图模式]



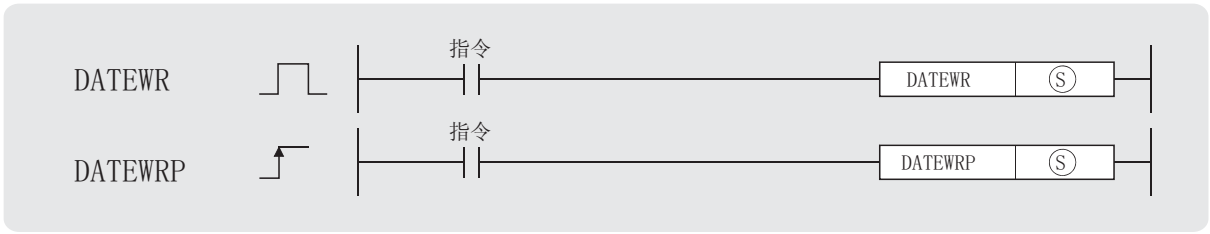
[列表模式]

步	指令	软元件	
0	LD	X40	
1	BIN	K4X30	D0
4	BIN	K2X28	D1
7	BIN	K2X20	D2
10	BIN	K2X18	D3
13	BIN	K2X10	D4
16	BIN	K2X8	D5
19	BIN	K1X4	D6
22	DATEWRP	D0	
24	END		

[动作]



2 当使用 QnACPU/Q4ARCPU 时

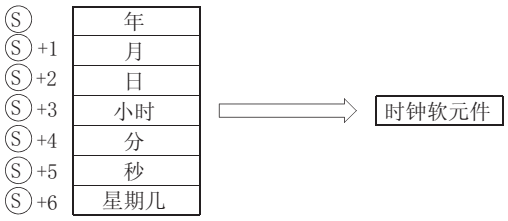


Ⓢ : 存储写入到时钟因子中的时钟数据的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---					

功 能

(1) 将在由 Ⓢ 指定号码的软元件之后的时钟数据写入到 CPU 模块时钟软元件中。



(2) 每一个项目均被设置为 BIN 值。

(3) 由 Ⓢ 指定的 “年” 被设置为年份的四位数的后两位数。
(如果年份设定使用了特殊纪元名，闰年将无法自动调整。)1995 年应设定为 “95”。

(4) Ⓢ+1 指定的 “月” 是用从 1 到 12 的值来表示的。(1 月到 12 月)

(5) Ⓢ+2 指定的 “日” 是用从 1 到 31 的值进行设置的。

(6) Ⓢ+3 指定的 “小时” 是用从 0 到 23 的值进行设置的 (使用 24 小时制时钟，从 0 小时到 2300 小时) (使用 24 小时制时钟)。

(7) Ⓢ+4 指定的 “分钟” 是用从 0 到 59 的值进行设置的。

(8) Ⓢ+5 指定的 “秒” 是用从 0 到 59 的值进行设置的。

(9) Ⓢ+6 指定的 “星期几” 是用从 0 到 6 的值来表示的 (星期日到星期六)。

星期几	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存储的数据	0	1	2	3	4	5	6

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

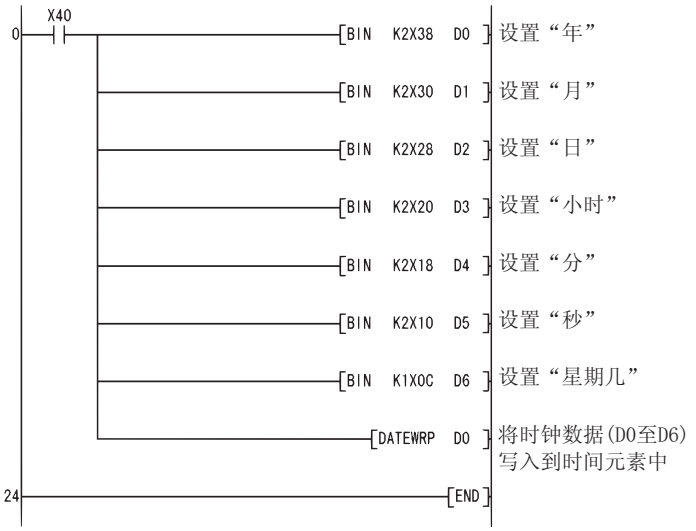
• 数据的个别项目设置在设定范围之外。 （出错代码： 4100）

程序示例

(1) 当 X40 为 ON 时，下列程序将以下时钟数据做为 BCD 值写入到时钟软元件中去。

年..... X38 到 X3F　　小时..... X20 到 X27
月..... X30 到 X37　　分钟..... X18 到 X1F
日..... X28 到 X2F　　秒..... X10 到 X17
星期几..... XC 到 XF

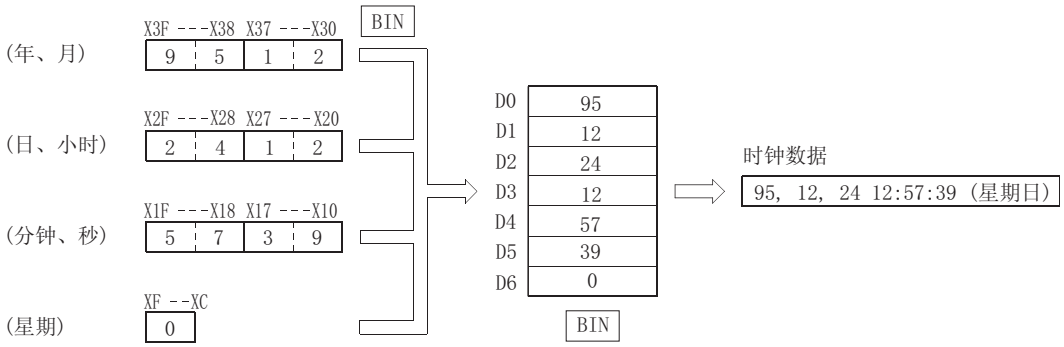
[梯形图模式]



[列表模式]

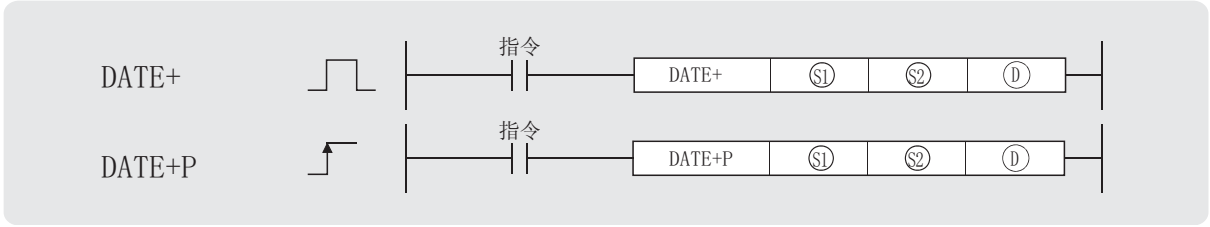
步	指令	软元件	
0	LD	X40	
1	BIN	K2X38	D0
4	BIN	K2X30	D1
7	BIN	K2X28	D2
10	BIN	K2X20	D3
13	BIN	K2X18	D4
16	BIN	K2X10	D5
19	BIN	K1X0C	D6
22	DATEWRP	D0	
24	END		

[动作]



7.15.3 时钟数据加法运算 (DATE+(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) ：存储被进行加法运算的时钟数据的软元件的起始号 (BIN 16 位)

(S2) ：存储进行加法运算的时间数据的软元件的起始号 (BIN 16 位)

(D) ：存储时钟 (时间) 数据加法结果的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	---	○		---					
(S2)	---	○		---					
(D)	---	○		---					

功 能

(1) 将存储在由 (S2) 指定软元件号的软元件之后的软元件中的时间数据，加到存储在从由 (S1) 指定软元件号的软元件开始的软元件中的时间数据上，并且在从由 (D) 指定软元件号的软元件开始的软元件中存储相加的结果。

		数据范围			数据范围			数据范围
(S1)	小时	(0到23)	(S2)	小时	(0到23)	(D)	小时	(0到23)
(S1)+1	分钟	(0到59) +	(S2)+1	分钟	(0到59) ⇨	(D)+1	分钟	(0到59)
(S1)+2	秒	(0到59)	(S2)+2	秒	(0到59)	(D)+2	秒	(0到59)

例如：将时间 7:48:10 与 6:32:40 相加会发生以下运算：

(S1)	小时：6		(S2)	小时：7		(D)	小时：14
(S1)+1	分钟：32	+	(S2)+1	分钟：48	⇨	(D)+1	分钟：20
(S1)+2	秒：40		(S2)+2	秒：10		(D)+2	秒：50

(2) 如果时间相加的结果超过了 24 小时，将会从相加的结果中减去 24 小时，从而得到最终的运算结果。

例如：如果时间 20:20:20 与 14:20:30 相加，结果不会是 34:40:50，而是 10:40:50。

(S1)	小时：14		(S2)	小时：20		(D)	小时：10
(S1)+1	分钟：20	+	(S2)+1	分钟：20	⇨	(D)+1	分钟：40
(S1)+2	秒：30		(S2)+2	秒：20		(D)+2	秒：50

备 注

关于能够设置为小时、分钟和秒的数据的更多信息，请参见 7.15.2 项。

运算错误

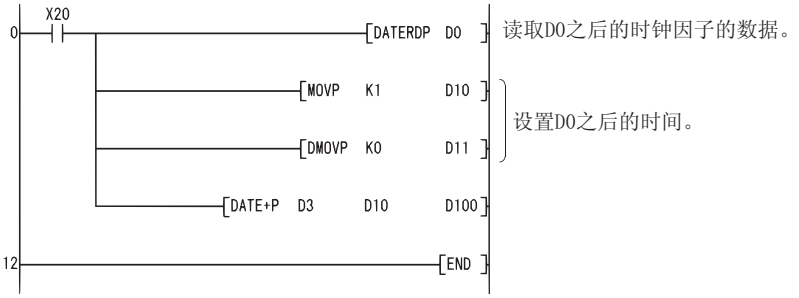
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 (S1) 和 (S2) 设置的数据在设定范围之外。 （出错代码： 4100）
- 由 (S1)、(S2) 或 (D) 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 当 X20 为 ON 时，下列程序将从时钟软元件中读取的时钟数据加上 1 小时，并在 D100 之后存储结果。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X20		
1	DATERDP	D0		
3	MOVP	K1	D10	
5	DMOVP	K0	D11	
8	DATE+P	D3	D10	D100
12	END			

[动作]

- 由 DATERDP 指令触发的时钟数据读取操作。

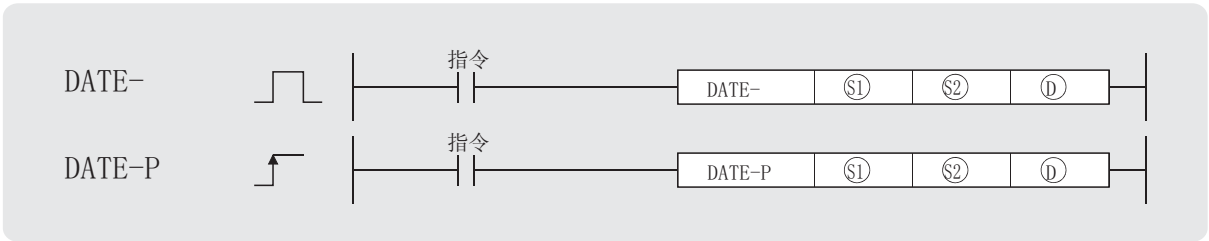
时钟软元件			
D0	95	年	
D1	5	月	
D2	15	日	
D3	10	小时	时间数据
D4	23	分	
D5	41	秒	
D6	2	星期几	

- 由 DATE+P 指令触发的加法运算。

D3	小时：10		D10	小时：1		D100	小时：11
D4	分钟：23	+	D11	分钟：0	⇒	D101	分钟：23
D5	秒：41		D12	秒：0		D102	秒：41

7.15.4 时钟数据减法运算 (DATE-(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



(S1) : 存储被进行减法运算的时钟数据的软元件的起始号 (BIN 16 位)

(S2) : 存储进行减法运算的时间数据的软元件的起始号 (BIN 16 位)

(D) : 存储时钟 (时间) 数据减法运算结果的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	---	○				---			
(S2)	---	○				---			
(D)	---	○				---			

功 能

(1) 从在由 (S1) 指定软元件号的软元件中存储的时间数据中，减去从由 (S2) 指定软元件号的软元件开始的软元件中存储的时间数据，并且在从由 (D) 指定软元件号的软元件开始的软元件中存储减法运算结果。

		数据范围			数据范围			数据范围
(S1)	时	(0到23)	(S2)	时	(0到23)	(D)	时	(0到23)
(S1)+1	分钟	(0到59)	(S2)+1	分钟	(0到59)	(D)+1	分钟	(0到59)
(S1)+2	秒	(0到59)	(S2)+2	秒	(0到59)	(D)+2	秒	(0到59)

例如：如果从时钟时间 10:40:20 中减去时钟时间 3:50:10，运算执行如下：

(S1)	时：10		(S2)	时：3		(D)	时：6
(S1)+1	分钟：40	−	(S2)+1	分钟：50	⇨	(D)+1	分钟：50
(S1)+2	秒：20		(S2)+2	秒：10		(D)+2	秒：10

(2) 如果减法运算的结果是一个负数，那么将在结果上加上 24，从而得到最终的运算结果。
例如：如果从 4:50:32 中减去时钟时间 10:42:12，结果不会是 -6:8:20，而是 18:8:20。

(S1)	小时：4		(S2)	小时：10		(D)	小时：18
(S1)+1	分钟：50	−	(S2)+1	分钟：42	⇨	(D)+1	分钟：8
(S1)+2	秒：32		(S2)+2	秒：12		(D)+2	秒：20

备 注

关于能够设置为小时、分钟和秒的数据的更多信息，请参见 7.15.2 项。

运算错误

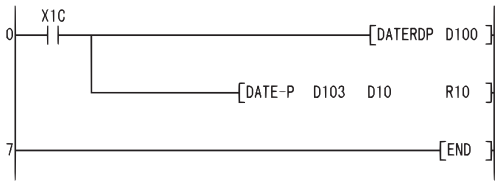
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 (S1) 和 (S2) 设置的数据在设定范围之外。 （出错代码： 4100）
- 由 (S1) 、(S2) 或 (D) 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。） （出错代码： 4101）

程序示例

(1) 当 X1C 为 ON 时，下列程序从时钟软元件中读取的时钟数据中减去从 D10 开始的软元件中存储的时间数据，并且在从 R10 开始的软元件中存储结果。

[梯形图模式]

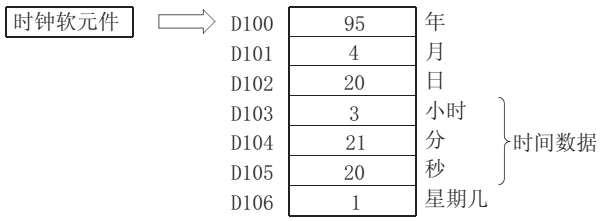


[列表模式]

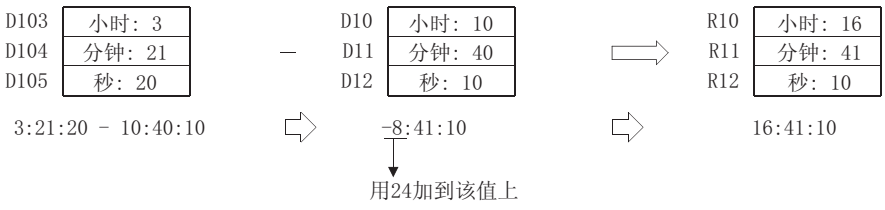
步	指令	软元件		
0	LD	X1C		
1	DATERDP	D100		
3	DATE-P	D103	D10	R10
7	END			

[动作]

- 由 DATERDP 触发的时间数据读取运算。

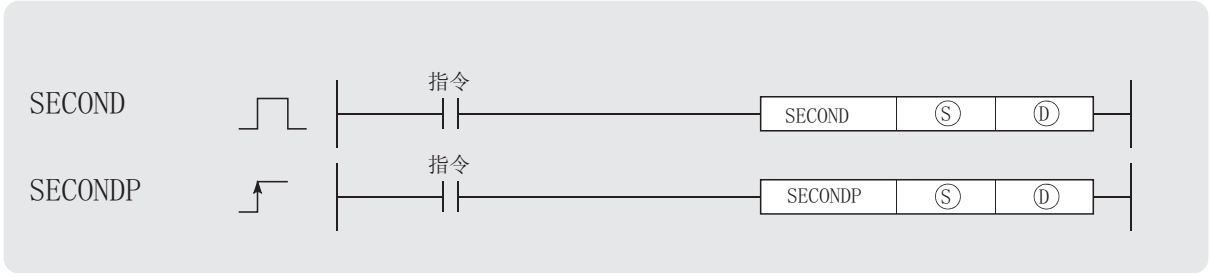


- 由 DATE-P 指令触发的减法运算（当 D10 到 D12 已经指定了 10 小时，40 分钟和 10 秒时）



7.15.5 时间数据转换（从小时／分钟／秒到秒）(SECOND(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



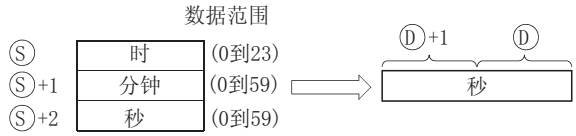
Ⓢ：存储转换之前的时钟数据的软元件的起始号（BIN 16 位）

Ⓓ：存储转换之后的时钟数据的软元件的起始号（BIN 32 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	---	○		---				---	
Ⓓ	○	○		○				---	

功 能

(1) 将存储在由 Ⓢ 指定的软元件之后的时间数据转换为秒，并且在由 Ⓓ 指定的软元件中存储转换结果。



例如，如果值为 4 小时，29 分钟，31 秒，转换执行如下：



运算错误

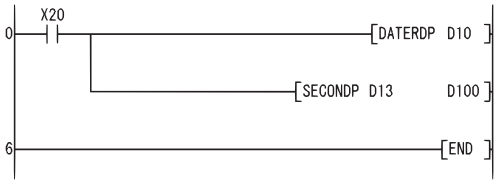
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的数据在可接受的范围之外。（出错代码：4100）
- 由 Ⓢ 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。）（出错代码：4101）

程序示例

(1) 当 X20 变为 ON 时，下列程序将从时钟软元件中读取的时钟时间数据转换为秒，并且在 D100 和 D101 中存储结果。

[梯形图模式]

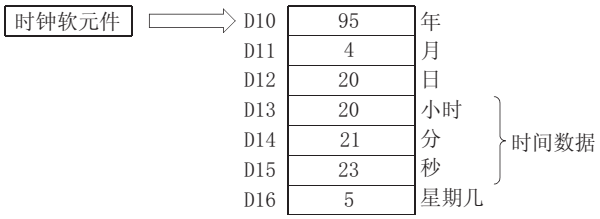


[列表模式]

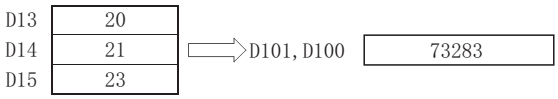
步	指令	软元件	
0	LD	X20	
1	DATERDP	D10	
3	SECONDP	D13	D100
6	END		

[动作]

• 由 DATERDP 触发的时间数据读取运算。



• 当由 SECONDP 指令触发时转换为秒。



7.15.6 时间数据转换（从秒到小时／分钟／秒）(HOUR(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



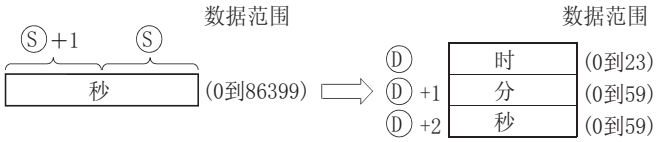
Ⓢ：存储转换之前的时钟数据的软元件的起始号（BIN 32 位）

Ⓓ：存储转换之后的时钟数据的软元件的起始号（BIN 16 位）

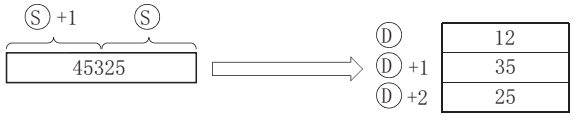
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	○	○		○				○	---
Ⓓ	---	○		---				---	---

功 能

(1) 将存储在由 Ⓢ 指定的软元件号中的秒格式数据转换为小时／分钟／秒格式，并且在从由 Ⓓ 指定的软元件开始的区域存储转换结果。



例如，如果指定值为 45325 秒，转换执行如下：



运算错误

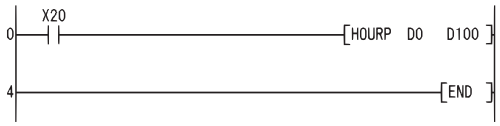
(1) 政治以下情况下将发生运算错误，错误标志（SM0）将变为 ON，出错代码将存储在 SD0 中。

- 由 Ⓢ 设置的数据在设定范围之外。（出错代码：4100）
- 由 Ⓓ 指定的软元件超出了相应软元件的范围。
  （只对于通用型 QCPU。）（出错代码：4101）

程序示例

(1) 当 X20 为 ON 时，下列程序将存储在 D0 和 D1 中的秒数据转换为时、分、秒格式，并且在从 D100 开始的软元件中存储结果。

[梯形图模式]

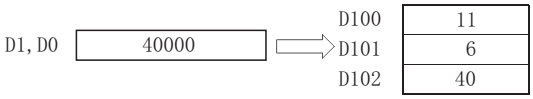


[列表模式]

步	指令	软元件	
0	LD	X20	
1	HOURP	D0	D100
4	END		

[动作]

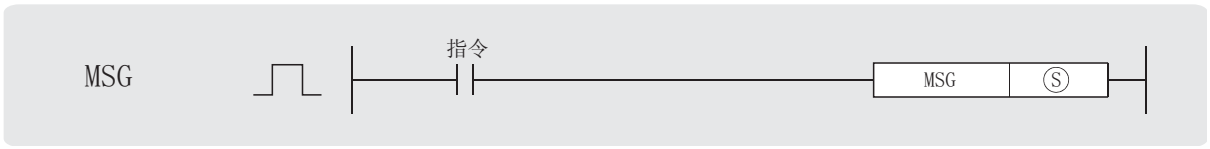
• 通过 HOURP 指令转换为时、分和秒格式。(当 D1 和 D0 中指定了 40000 秒时)



7.16 外围设备指令

7.16.1 将信息显示到外围设备 (MSG)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

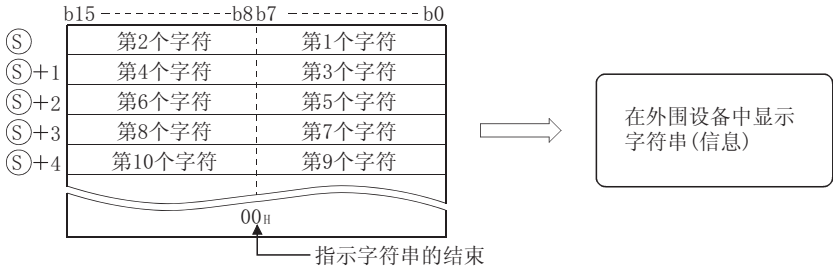


Ⓢ ：要显示的字符串数据，或存储字符串数据的软元件的起始（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

(1) 将从由 Ⓢ 指定软元件号的软元件开始的软元件中存储的字符串数据，作为一条信息显示到在终端模式中设置的外围设备中。
该字符串数据的结束位置用 00H 作为标记。



(2) 在外围设备中显示的字符串数据最多能包括 64 个字符。

(3) 由 Ⓢ 指定的字符串数据存储在从 SD738 到 SD769 的位置上（信息存储区域）。

(4) 当执行 MSG 指令时，SM738(MSG 指令执行信号）被变为 ON。
当 SM738 变为 ON 时，将不会处理任何其它被执行的 MSG 指令。

(5) 当在外围设备中显示字符串（信息）时，SM738 变为 OFF，并且从 SD738 到 SD769 上存储的字符串数据被清除 (00H)。

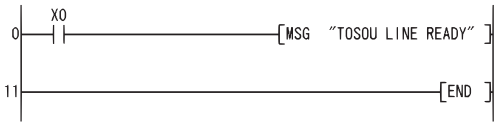
运算错误

(1) 没有与 MSG 指令相关的运行出错。

程序示例

(1) 当 X0 变为 ON 时，下列程序传送信息 “TOSOU LINE READY”以将其显示在一个外围设备上。

[梯形图模式]



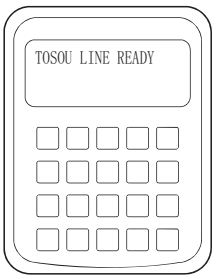
[列表模式]

步	指令	软元件
0	LD	X0
1	MSG	"TOSOU LINE READY"
11	END	

[动作]

	b15----------b8	b7----------b0
SD738	4F$_H$(O)	54$_H$(T)
SD739	4F$_H$(O)	53$_H$(S)
SD740	20$_H$(空格)	55$_H$(U)
SD741	49$_H$(I)	4C$_H$(L)
SD742	45$_H$(E)	4E$_H$(N)
SD743	52$_H$(R)	20$_H$(空格)
SD744	41$_H$(A)	45$_H$(E)
SD745	59$_H$(Y)	44$_H$(D)
SD746	00$_H$	

显示



备 注

关于讨论的如何在终端模式中操作外围设备，请参考外围设备操作手册。

7.16.2 外围设备的键盘输入 (PKEY)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓓ ：存储键盘输入的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓓ	---	○		---					

功 能

(1) 在执行 PKEY 指令时，17 个字被从由 Ⓓ 指定的软元件中清除，并且 SM736(PKEY 指令执行中标记) 变为 ON。

另外，SM737(键盘输入接收标记) 变为 OFF。

当 PKEY 指令执行结束时，在由 Ⓓ 指定的软元件之后，在终端模式中被指定的外围设备中的键盘输入数据被存储为 ASCII 码。

(2) 当 PKEY 指令变为 OFF 时，SM736 和 SM737 均变为 OFF。

(3) 当键盘输入的一个字符被从外围设备中接收到时，SM737 变为 ON，并且当键盘输入的数据被 QnACPU 存储时，SM737 变为 OFF。

然而当 SM737 是 ON 时，来自外围设备的键盘输入将不会被接受。

(4) 当从外围设备中键入 “CR” 时， “CR” 的 ASCII 码 (0DH) 将附加在键入数据的末尾，从而结束从外围设备的键入操作。

(5) 能够从外围设备中输入的数据量为 32 个字符。

如果从外围设备中已经输入了 32 个字符，即使可能没有输入 “CR”，来自外围设备的键盘输入也会结束。

(6) 对由 Ⓓ 指定的软元件之后的数据，存储执行如下：

	计数器		
Ⓓ			→ 通过键盘输入的字符数被存储为BIN
Ⓓ+1	第2个字符	第1个字符	
Ⓓ+2	第4个字符	第3个字符	
Ⓓ+3	第6个字符	第5个字符	
			最多16个字
Ⓓ+15	第30个字符	第29个字符	
Ⓓ+16	第32个字符	第31个字符	

(7) PKEY 指令不能在两个或更多的位置上同时执行。

如果需要在两个或更多的位置上使用 PKEY 指令，可以用 SM736(PKEY 指令执行中标记) 来建立一个互锁，从而使得它们可以同时被执行。

运算错误

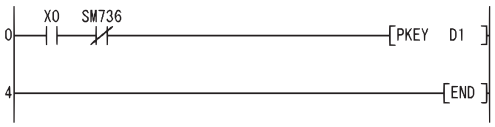
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 试图去存储超过由 ⓓ 指定的软元件范围的键盘输入数据。　　（出错代码： 4101）

程序示例

(1) 在 X0 变为 ON 后，下列程序在 D1 之后从外围设备中输入字符 “TOSOU LINE READY”。

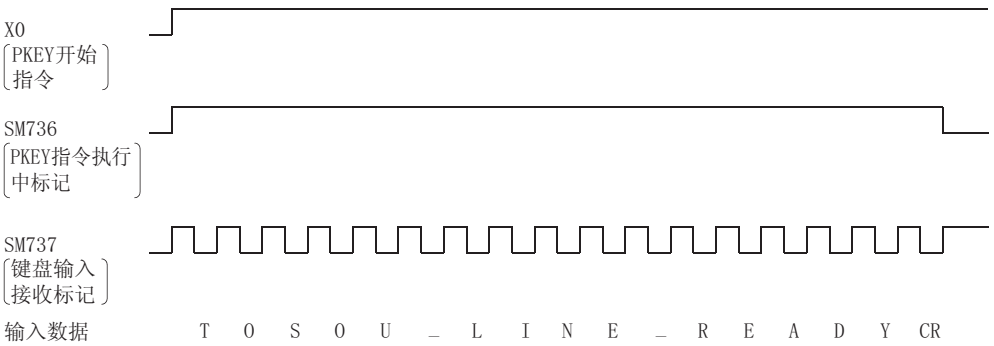
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	ANI	SM736
2	PKEY	D1
4	END	

[动作]



[键盘输入数据存储]

D1	17	
D2	4FH	54H
D3	4FH	53H
D4	20H	55H
D5	49H	4CH
D6	45H	4EH
D7	4CH	20H
D8	41H	45H
D9	59H	44H
D10	00	0DH

7.17 程序控制指令

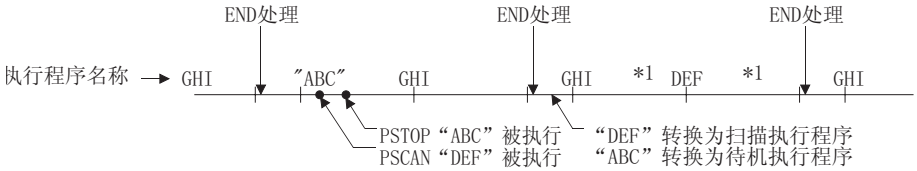
(1) 当执行类型被程序控制指令转换时的处理如下：

改变之前的执行类型	执行的指令			
	PSCAN	PSTOP	POFF	PLOW
扫描执行类型	保持扫描执行类型不变。	变为待机类型。	在下一个扫描中输出被变为 OFF，在这之后的下一个扫描开始变成待机类型。	变为低速执行类型。
初始执行类型	变为扫描执行类型。			
待机类型		保持待机类型不变。	无处理	
低速执行类型	低速执行类型的执行被中断，从下一个扫描开始变为扫描执行类型。（从 0 步开始执行）	低速执行类型的执行被中断，从下一个扫描开始变为待机类型。	低速执行类型的执行被中断，在下一个扫描中输出变为 OFF。 在这之后的下一个扫描开始变为待机类型。	保持低速执行类型不变。
恒定周期执行类型	变为扫描执行类型。	变为待机类型。	在下一个扫描中输出被变为 OFF。 在这之后的下一个扫描开始变成待机类型。	变为低速执行类型。

要 点

如果将恒定周期执行类型程序改变为其它执行类型，将无法恢复为恒定周期执行类型。

(2) 因为 PSCAN 和 PSTOP 指令执行的程序执行类型转换是出现在 END 处理的时候，所以这类转换不可能出现在程序执行中。
当在同一个扫描中出现了为同一个程序设置了不同的执行类型时，那么执行类型就是最后执行的执行转换指令指定的那个执行类型。



*1: "GHI" 和 "DEF" 程序执行的顺序是由程序设定参数决定的。

在以下时序中，从恒定周期执行类型程序转换为执行类型程序。

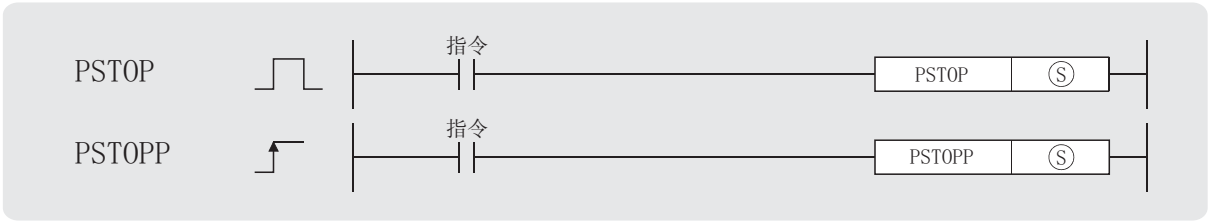
(a) 对于通用型 QCPU
在程序控制指令执行后的 END 处理时停止恒定周期执行类型的执行，执行类型被改变。

(b) 对于除通用型 QCPU 以外的 CPU 模块
在程序控制指令执行时停止恒定周期执行类型的执行，执行类型在 END 处理时被改变。

(3) 当执行 POFF 指令时，在下一个扫描时输出变为 OFF，并且在第三个扫描和后续扫描中执行类型将变为待机类型。
如果是在输出 OFF 处理之前被执行，则程序控制指令将被忽略。

7.17.1 程序待机指令 (PSTOP(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ：将要设置为待机状态的程序文件名的字符串数据，或存储字符串数据的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

(1) 将存储在由 Ⓢ 指定的软元件中的文件名程序设置为待机状态。

(2) 只有存储在 0 号驱动器（程序内存 / 内部 RAM）中的程序才能被设置为待机类型。

(3) 当 END 处理被执行时，指定的程序被设置为待机状态。

(4) 即使程序执行类型已经在参数中指定，该指令也将有更高的优先级。

(5) 不需要指定文件名的扩展名（.QPG）。
（只有 .QPG 文件才能进行操作）

运算错误

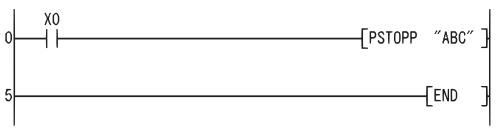
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定文件名的程序不存在。（出错代码：2410）
- 由 Ⓢ 指定的文件名的程序类型为 SFC 程序。（出错代码：2412）
- Ⓢ 的文件名存储目标软元件超出了相应软元件的范围。（出错代码：4101）

程序示例

(1) 当 X0 变为 ON 时，下列程序将文件名为 ABC 的程序设置到待机状态中。

[梯形图模式]

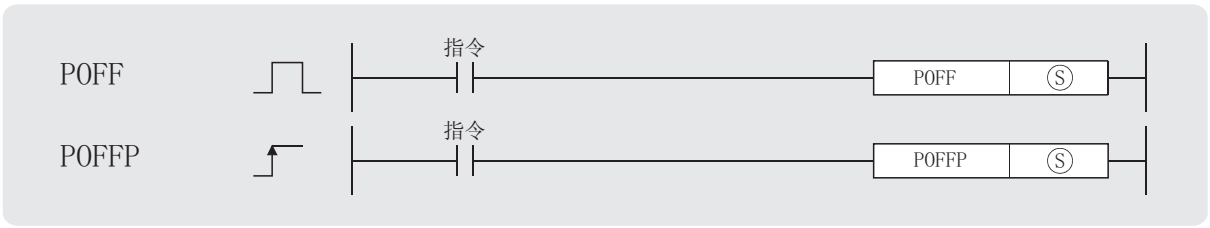


[列表模式]

步	指令	软元件
0	LD	X0
1	PSTOPP	"ABC"
5	END	

7.17.2 程序输出 OFF 待机指令 (POFF(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 变为 OFF 输出来指定待机状态的程序的文件名，或存储文件名的软元件（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

(1) 改变文件名存储在由 Ⓢ 指定的软元件中的程序的执行类型。

- 扫描执行类型：　在下一个扫描中变为 OFF 输出（无执行处理）。
  在后续扫描中程序被设置为待机类型。
- 低速执行类型：　停止低速执行类型的执行，并在下一个扫描变为 OFF 输出。程序在下一个扫描之后的扫描中被设置为待机类型。

(2) 只有存储在 0 号驱动器（程序内存 / 内部 RAM）中的程序能够被设置为待机类型。

(3) 即使程序执行类型已经在参数中指定，该指令也将有更高的优先级。

(4) 不需要指定文件名的扩展名 (.QPG)。（只有 .QPG 文件才能进行操作。）

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定文件名的程序不存在。　（出错代码：2410）
- Ⓢ 的文件名存储目标软元件超出了相应软元件的范围。　（出错代码：4101）

备 注

1. 非执行处理是指每个线圈指令的处理和条件设置为 OFF 状态时的处理是一样的。

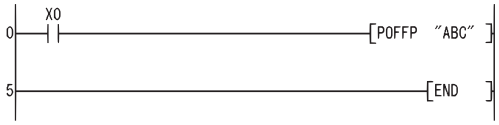
2. 不管条件触点是 ON 还是 OFF，在非执行处理之后的每个线圈指令的运算结果如下：

OUT 指令		..... 强制为 OFF
SET 指令 RST 指令 SFT 指令 基本指令 应用指令	)	..... 状态被保持
PLS 指令 脉冲转换指令 (□ P)	)	..... 同条件触点为 OFF 时的处理一致
低速 / 高速定时器的当前值		 0
累计定时器的当前值 计数器的当前值	)	..... 被保持

程序示例

(1) 当 X0 变为 ON 时，下列程序将文件名为 ABC 的程序设置为不可执行并且将其放置在待机状态中。

[梯形图模式]

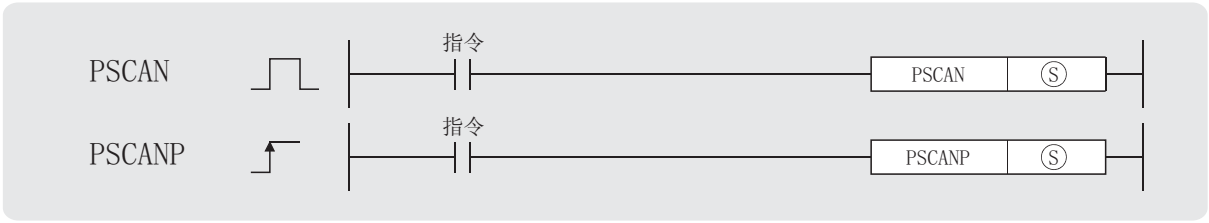


[列表模式]

步	指令	软元件
0	LD	X0
1	POFFP	"ABC"
5	END	

7.17.3 程序扫描执行登记指令 (PSCAN(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 被置为扫描执行类型的程序的文件名，或存储文件名的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	---	○		---				○	---

功 能

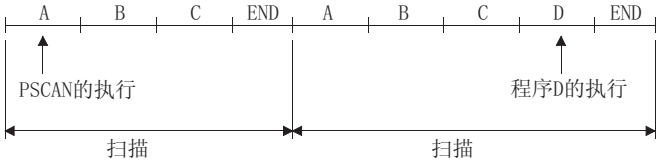
(1) 将文件名存储在由 Ⓢ 指定的软元件中的程序设置为扫描执行类型。

(2) 只有存储在 0 号驱动器（程序内存 / 内部 RAM）中的程序能够被设置为扫描执行类型。

(3) 指定的程序假定扫描执行类型带有 END 处理。

例

当存在程序 A、B 和 C，并且程序 A 执行程序 D 的 “PSCAN”。



(4) 即使程序执行类型已经在参数中指定，该指令也将有更高的优先级。

(5) 不需要指定文件名的扩展名（.QPG）。（只有 .QPG 文件才能进行操作。）

运算错误

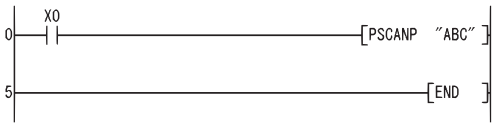
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定文件名的程序不存在。 (出错代码： 2410)
- ⓢ 的文件名存储目标软元件超出了相应软元件的范围。 (出错代码： 4101)
- 指定的文件名为 SFC 程序，并且其它文件名的 SFC 程序已经启动。(SFC 程序重复启动错误)
  (只对于通用型 QCPU) (出错代码： 4131)
  (对于高性能型 QCPU、过程 CPU、冗余 CPU 和 QnACPU) (出错代码： 2504)

程序示例

(1) 当 X0 变为 ON 时，下列程序将文件名为 ABC 的程序设置到扫描执行状态中。

[梯形图模式]

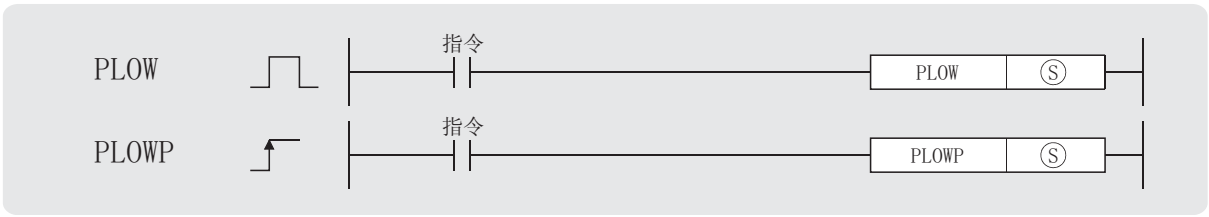


[列表模式]

步	指令	软元件
0	LD	X0
1	PSCANP	"ABC"
5	END	

7.17.4 程序低速执行登记指令 (PLOW(P))

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 要设定为低速执行类型的程序的文件名，或存储文件名的软元件的起始号（字符串）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			$	
Ⓢ	---	○		---				○	---

功 能

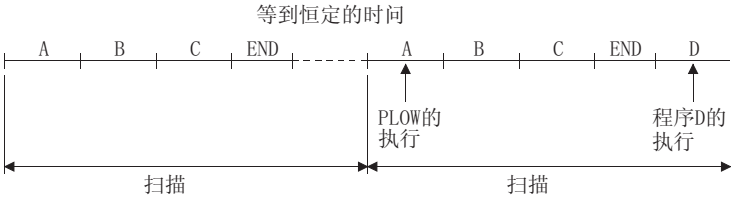
(1) 将文件名存储在由 Ⓢ 指定的软元件中的程序设置为低速执行类型。

(2) 只有存储在 0 号驱动器（程序内存 / 内部 RAM）中的程序能够被设置为低速执行类型。

(3) 指定的程序假定低速执行类型的程序带有 END 处理。

例

当存在程序 A、B 和 C，并且程序 A 执行程序 D 的 “PLOW”。（假定恒定扫描已经被设置）



(4) 即使程序执行类型已经在参数中指定，该指令也将有更高的优先级。

(5) 不需要指定文件名的扩展名 (.QPG)。
（只有 .QPG 文件才能进行操作。）

运算错误

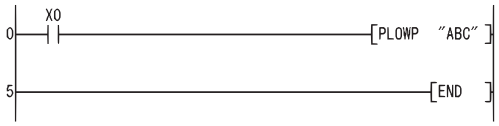
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 指定文件名的程序不存在。 （出错代码： 2410）
- 在文件名已经被指定的程序中有一个 CHK 指令。 （出错代码： 4235）

程序示例

(1) 当 X0 变为 ON 时，下列程序将文件名为 ABC 的程序设置到低速执行状态中。

［梯形图模式］

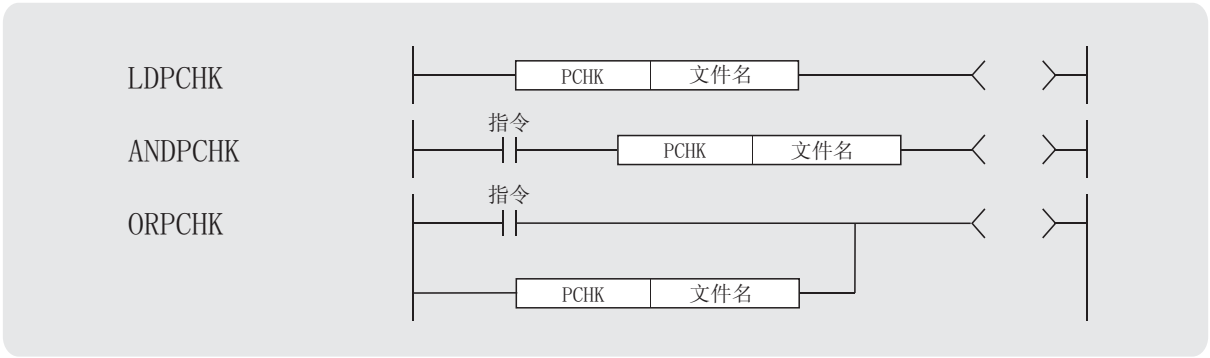


［列表模式］

步	指令	软元件
0	LD	X0
1	PLOWP	"ABC"
5	END	

7.17.5 程序执行状态检查指令 (PCHK)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



ⓢ : 进行执行状态检查的程序的文件名 (字符串)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
ⓢ	---							○	---

功 能

(1) 检查指定文件名的程序是否在执行。

(2) 当指定文件名的程序在执行时该指令导通，当程序为非执行时该指令不导通。

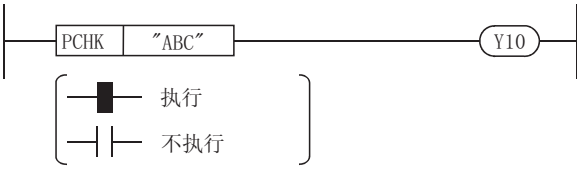
(3) 指定文件名时不需要指定扩展名 (.QPG)。
例如，当文件名为 ABC.QPG 时，指定 “ABC”。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。
• 指定文件名的程序不存在。 (出错代码 : 2410)

程序示例

(1) 当执行程序文件 “ABC.QPG” 时，该程序保持 Y10 为 ON。



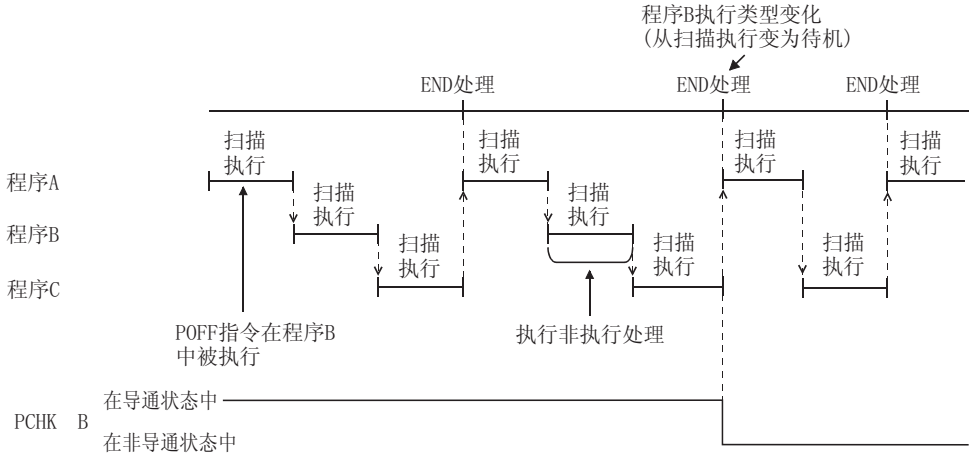
备 注

“非执行”表示程序执行类型为待机类型。
“执行”表示程序执行类型为扫描执行类型（包括输出 OFF（在非执行处理过程中）的过程），低速执行类型或恒定固定扫描执行类型。

要 点

当指定文件名的程序（目标程序）在执行中时，PCHK 指令处于导通状态，并且当程序是非执行时，该指令也是处于非导通状态。
当目标程序被用 POFF 指令设置为非执行（待机类型）时，在目标程序的非执行处理被执行过程中，PCHK 指令处于导通状态。
当在非执行处理结束处的扫描周期的 END 处理时，目标程序被设置为非执行（待机类型），并且 PCHK 指令被置为非导通状态。
因此请注意，当 PCHK 指令在非执行处理已经被 POFF 指令结束的程序中被执行，PCKK 指令可能被置为导通状态。

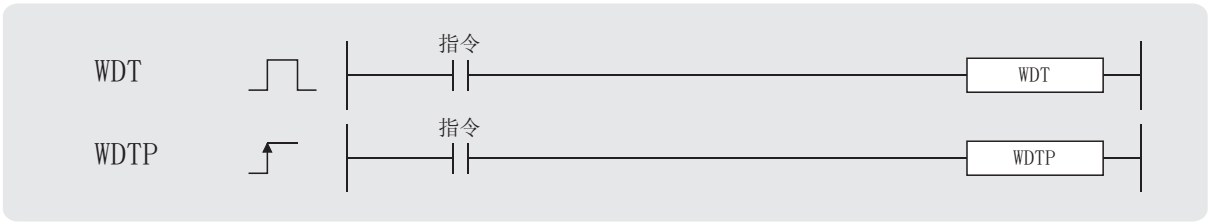
当程序 A 执行程序 B 的 POFF 指令并且程序 C 执行程序 B 的 PCHK 指令时，下列图表表示所执行的运算，其中程序是按程序 A、程序 B 和程序 C 的顺序来执行的。



7.18 其它指令

7.18.1 复位看门狗定时器 (WDT(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



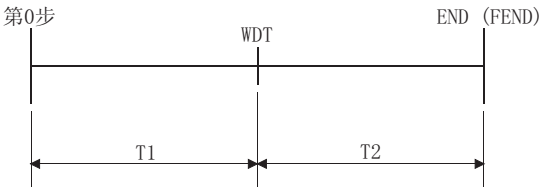
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
---	---		---						

功 能

(1) 在顺控程序的执行过程中复位看门狗定时器。

(2) 该指令用在由于主要的条件，扫描时间超过为看门狗定时器设置的值的情况中。
如果在每个扫描中，扫描时间均超过了看门狗定时器设定值，那么在外围设备设定中改变看门狗定时器的设定。

(3) 注意既不是从 0 步到 WDT 指令的 t1，也不是从 WDT 到 END(FEND) 指令的 t2 超过了看门狗定时器的设定值。



(4) 在一个扫描中 WDT 指令能够被使用两次或更多次，但是，因为在产生错误的过程中直到输出变为 OFF 所需要的时间的原因，在这种情况中应该格外注意。

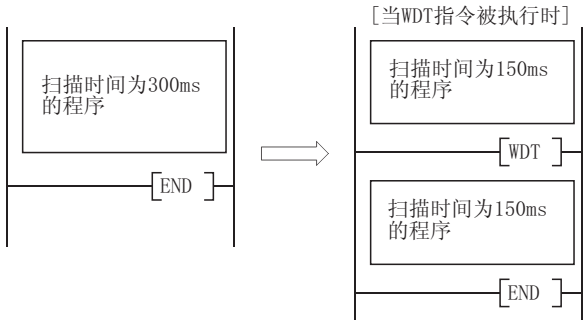
(5) 即使 WDT 或 WDTP 指令被执行，存储在特殊寄存器中的扫描时间值也不会被清除。
相应的，当为特殊寄存器设置的扫描时间的值大于用参数设置的看门狗定时器的值时，也有这种情况发生。

运算错误

(1) 没有与 WDT(P) 指令相关的运行错误。

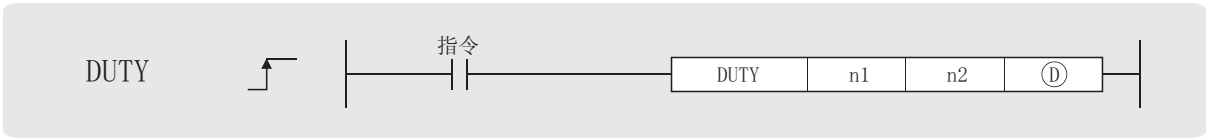
程序示例

(1) 下列程序设定了 200ms 的看门狗定时器从 0 步到 END(FEND) 指令的程序执行基于执行条件需要 300ms。



7.18.2 定时脉冲发生 (DUTY)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



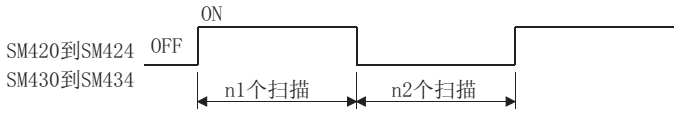
n1 : 使其为 ON 的扫描个数 (BIN 16 位)
n2 : 使其为 OFF 的扫描个数 (BIN 16 位)
Ⓓ : 用户定时时钟 (SM420 到 SM424，SM430 到 SM434)(位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	○			○					---
n2	○			○					---
Ⓓ	○*1			---					---

*1: 只能使用 SM420 到 SM424，SM430 到 SM434。

功 能

(1) 为由 Ⓓ 指定的用户定时时钟 (SM420 到 SM424、SM430 到 SM434)，将 n1 个扫描周期变为 ON，并将 n2 个扫描周期变为 OFF。



(2) 扫描执行类型程序使用 SM420 到 SM424，低速执行类型程序使用 SM430 到 SM434。

(3) 如果 n1 和 n2 都被设置为 0，将会发生下列情况：

(a) n1=0、 n2≧0　　SM420 ～ SM424/SM430 ～ SM434 将保持 OFF。

(b) n1>0、 n2=0　　SM420 ～ SM424/SM430 ～ SM434 将保持 ON。

(4) 当 DUTY 指令被执行时，被 n1、n2 指定的数据和 Ⓓ 在系统中登记，并且定时脉冲被 END 处理变为 ON 和 OFF。

运算错误

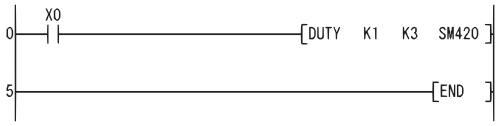
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 被 Ⓓ 指定的软元件不是在从 SM420 到 SM424 或从 SM430 到 SM434 的范围内。 (出错代码： 4101)
- n1 和 n2 的值小于 0。 (出错代码： 4100)

程序示例

(1) 如果 X0 为 ON，下列程序将 SM420 变为 ON 一个扫描周期，并将其变为 OFF 3 个扫描周期。

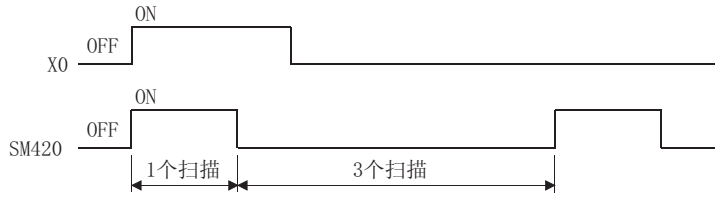
[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	X0		
1	DUTY	K1	K3	SM420
5	END			

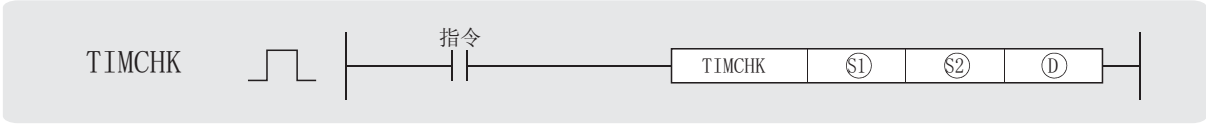
[动作]



7.18.3 时间检查指令 (TIMCHK)

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1：系列号的前五位数是 04122 或以后。



S1 ：存储测量的当前值的软元件 (BIN 16 位)
S2 ：存储测量的设定值的软元件 (BIN 16 位)
D ：当“时间到”时使其为 ON 的软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
S1	---	○		---					---
S2	○	○		○					---
D	○	---		---					---

功 能

(1) 测量作为条件使用的软元件的 ON 时间，并且如果条件软元件保持 ON 的时间比由 S2 指定的软元件中设定的时间要长，那么由 D 指定的软元件被变为 ON。

(2) 由 S1 指定的软元件的当前值被清除为 0，并且由 D 指定的软元件在输入条件的上升沿被变为 OFF。

如果输入条件变为 OFF，由 S1 指定的软元件的当前值和由 D 指定的软元件的 ON 状态将被保持。

(3) 测量的设定值是以 100ms 为单位设置的。

运算错误

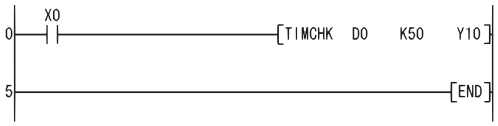
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 不能被指定的软元件已经被指定。 (出错代码： 4100)

程序示例

(1) 在下面这个程序中，X0 的 ON 时间被设置为 5s，当前值存储软元件被设置为 D0，并且在时间到达时要变为 ON 的软元件被设置为 Y10。

[梯形图模式]

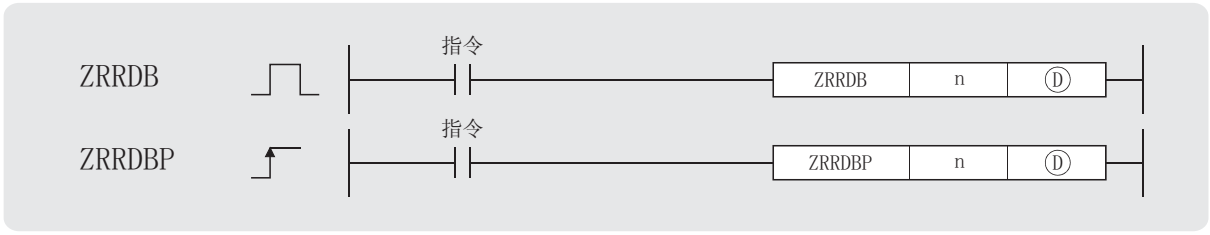


[列表模式]

步	指令	软元件		
0	LD	X0		
1	TIMCHK	D0	K50	Y10
5	END			

7.18.4 从文件寄存器中的直接 1 字节读取 (ZRRDB(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



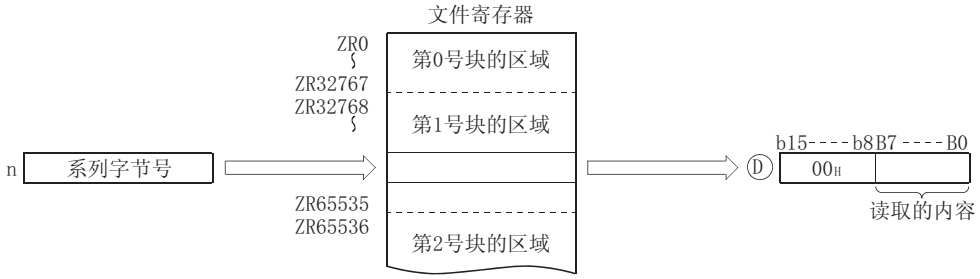
n ：将被读取的文件寄存器的系列字节号 (BIN 32 位)

Ⓓ ：存储读取数据的软元件号 (BIN 16 位)

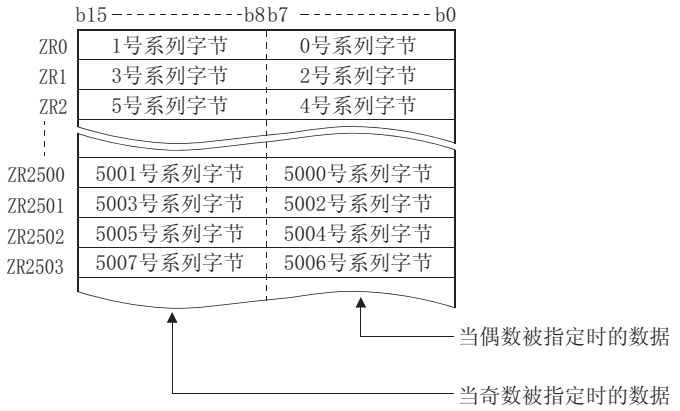
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n		○						○	---
Ⓓ		○						---	---

功 能

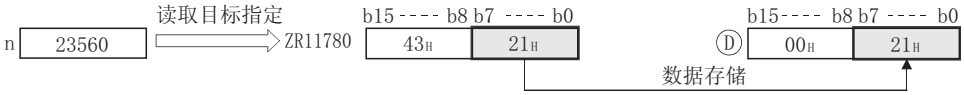
(1) 读取被 n 指定的系列字节号，n 不表示块号，并且存储在由 Ⓓ 指定的软元件中的低 8 位上。由 Ⓓ 指定的软元件的高 8 位将变为 00H。



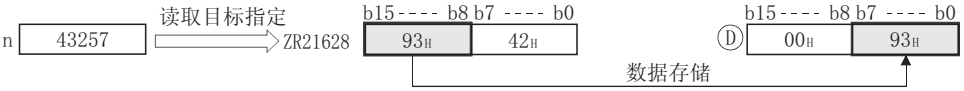
(2) 文件寄存器号和系列字节号的对应关系如下：



(a) 如果 n 的值已被指定为 23560，将会读取在 ZR11780 中低 8 位上的数据。



(b) 如果 n 的值已被指定为 43257，将会读取在 ZR21628 中高 8 位上的数据。



运算错误

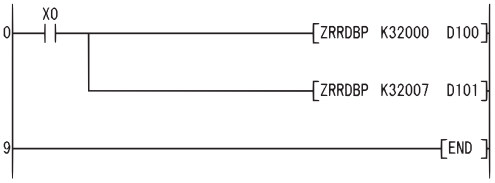
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 超出允许的指定范围的软元件号 (系列字节号) 已经被指定。 (出错代码 : 4101)

程序示例

(1) 当 X0 为 ON 时，下列程序读取 ZR16000 的低位和 R16003 的高位，并且在 D100 和 D101 中存储结果。

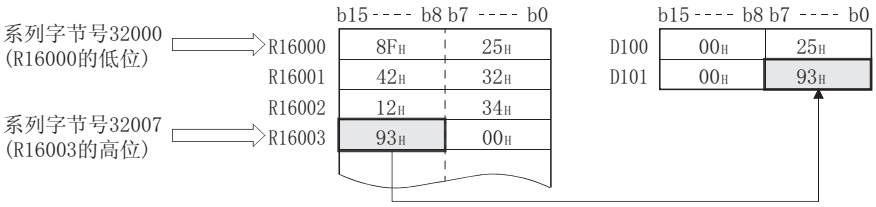
[梯形图模式]



[列表模式]

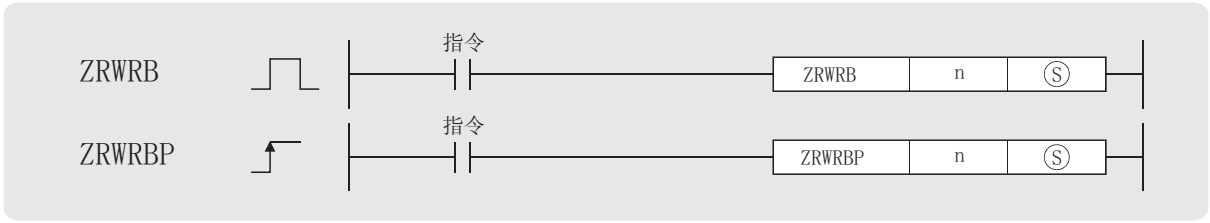
步	指令	软元件	
0	LD	X0	
1	ZRRDBP	K32000	D100
5	ZRRDBP	K32007	D101
9	END		

[动作]



7.18.5 写入 1 字节到文件寄存器 (ZRWRB(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n：将被写入的文件寄存器的系列字节号 (BIN 32 位)

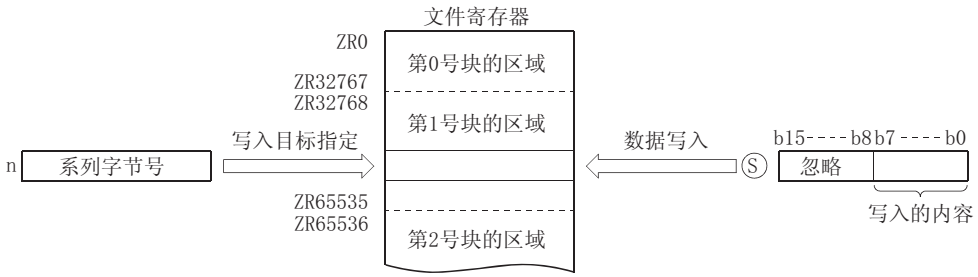
Ⓓ：存储被写入的数据的软元件号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n		○							---
Ⓢ		○							---

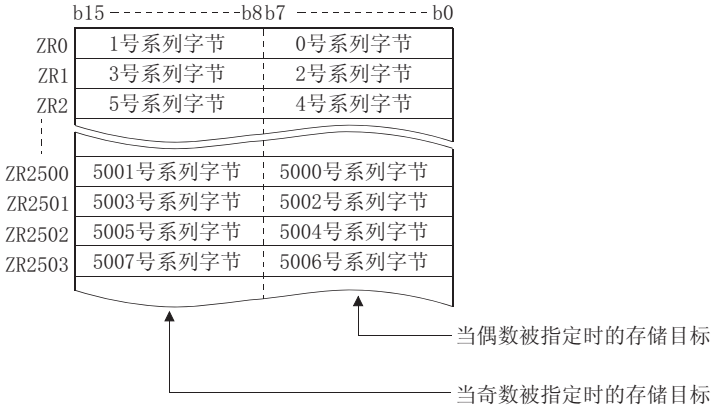
功 能

(1) 将在由 Ⓢ 不表示块号指定的软元件中存储的数据写入到被 n 指定系列字节号的文件寄存器中去。

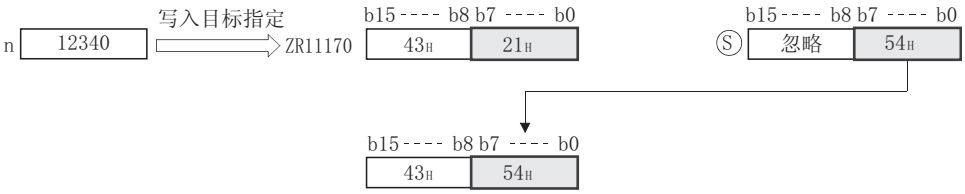
由 Ⓢ 指定的软元件中的数据的高 8 位将被忽略。



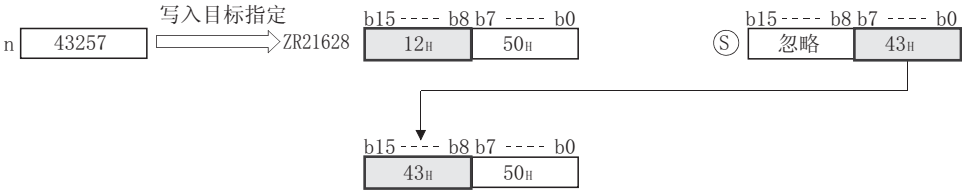
(2) 文件寄存器号和系列字节号的对应关系如下：



如果 n 的值被指定为 12340，那么 ZR11170 的低 8 位将被写入。



当 n 值指定为 43257 时，数据将被写入到 ZR21628 的高 8 位。



运算错误

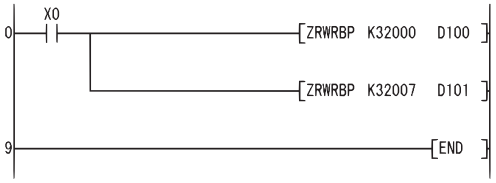
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 超出允许的指定范围的软元件号 (系列字节号) 已经被指定。 (出错代码 : 4101)

程序示例

(1) 当 X0 变为 ON 时，下列程序将 D100 和 D101 中低位上的数据写入到 R16000 的低位和 R16003 的高位上去。

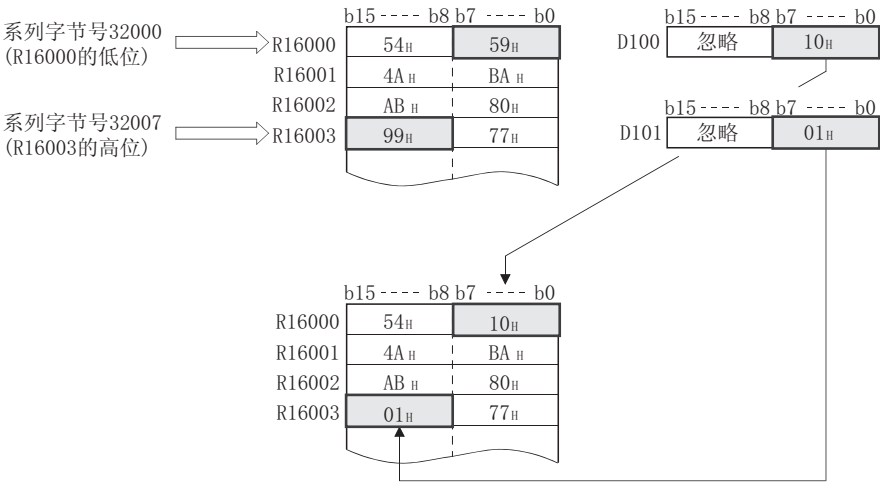
[梯形图模式]



[列表模式]

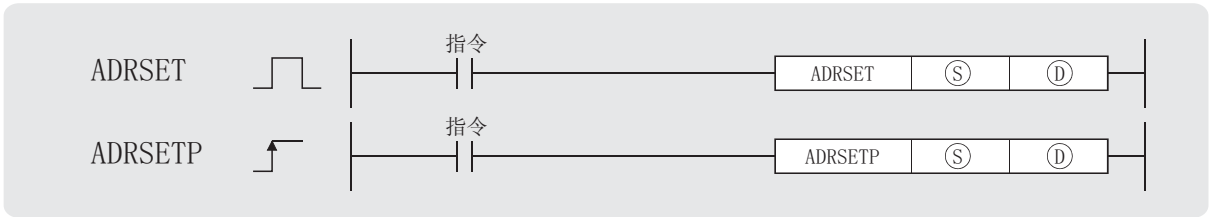
步	指令	软元件	
0	LD	X0	
1	ZRWRBP	K32000	D100
5	ZRWRBP	K32007	D101
9	END		

[动作]



7.18.6 间接地址读取操作 (ADRSET(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓢ : 用于读取间接地址的软元件号 (软元件名称)

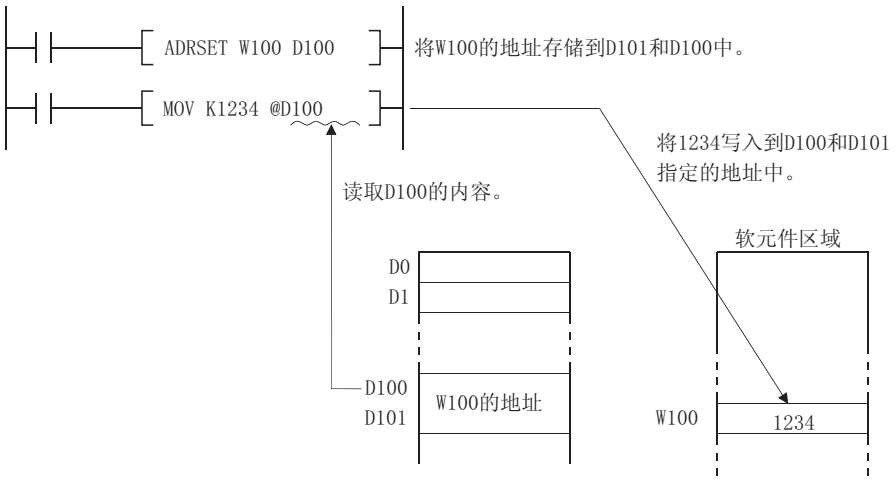
Ⓓ : 存储由 Ⓢ 指定的软元件的间接地址的软元件号 (BIN 32 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ		○			---				
Ⓓ		○			---				

功 能

(1) 在 Ⓓ+1 和 Ⓓ 中存储由 Ⓢ 指定的软元件的间接地址。

当间接软元件地址被顺控程序执行时，使用由 Ⓓ 指定的软元件中存储的地址。



(2) 不能在 Ⓢ 中进行位软元件指定。

运算错误

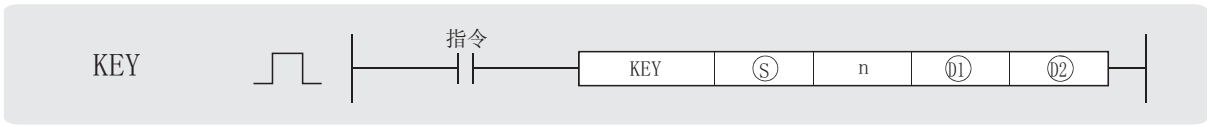
(1) 没有与 ADRSET(P) 指令相关的运算错误。

备 注

关于间接指定的更多信息，请参见 3.4 节。

7.18.7 键盘的数字键输入 (KEY)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR

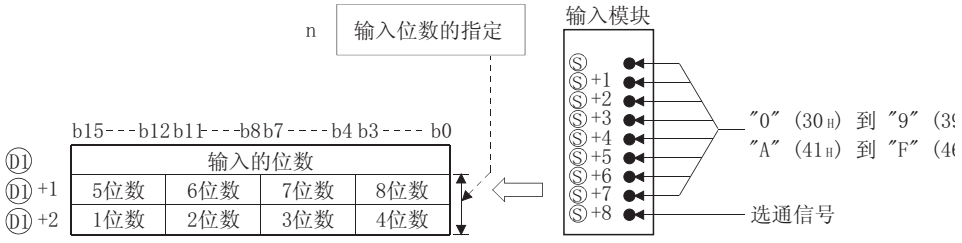


(S) : 进行数字输入 (X) 的软元件的起始号 (位)
n : 进行数字输入的位数 (BIN 16 位)
(D1) : 存储输入的数字的软元件的起始号 (BIN 16 位)
(D2) : 在输入结束时变为 ON 的位软元件的号码 (位)

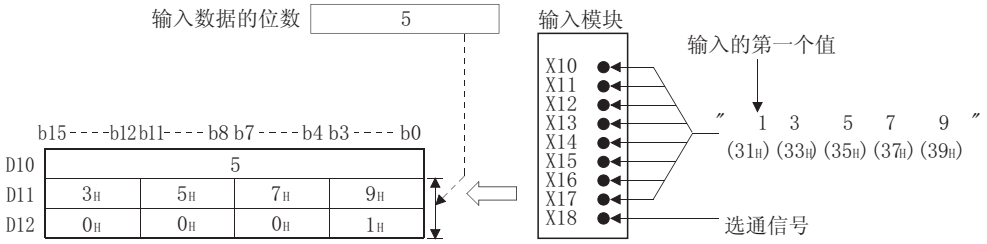
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
(S)	○ (仅 X)	---		---			---		---
n	○	○		○			○		---
(D1)	---	○		---			---		---
(D2)	○	○		○			---		---

功 能

(1) 从由 (S) 指定的输入 (X) 的 8 点中取出 ASCII 数据，将其转换为十六进制值，并且在从由 (D1) 指定的软元件开始的软元件中存储结果。

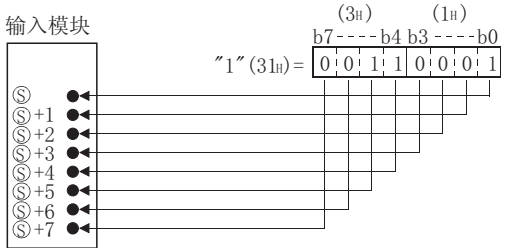


例如，当位数 (n) 已被设置为 5，并且已经通过输入模块的 X10 到 X18 输入值 “31”、“33”、“35”、“37” 和 “39” 时，将会发生下列情况：



(2) 输入到由 ⓢ 指定的输入 (X) 的数字输入在 ⓢ 到 ⓢ+7 上进行位扩展，并按照相应的数字以 ASCII 码输入。

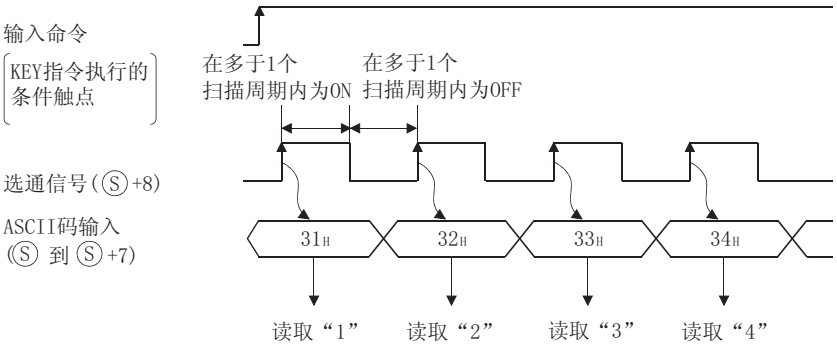
能够输入的 ASCII 码是在从 30H(0) 到 39H(9)，和从 41H(A) 到 46H(F) 的范围之内的。



(3) 在 ASCII 码输入到 ⓢ 到 ⓢ+7 之后，在 ⓢ+8 上的选通信号变为 ON，从而在内部合并指定的数字。

选通信号应该在顺控程序中的多于一个的扫描周期内保持其 ON 或 OFF 状态。

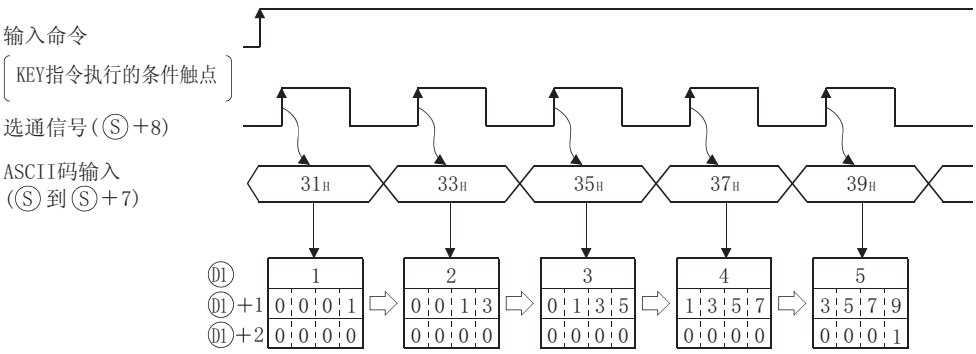
如果时间少于 1 个扫描周期，当数据被正确合并时将会发生几种情况。



(4) 在输入指定的位数之前，输入指令 (KEY 指令执行条件触点) 应该总保持 ON。

如果输入指令变为 OFF，那么 KEY 指令就不能被执行。

(5) 实际从 ⓓ1 中取出的数字的位数将会在 ⓓ1 指定的软元件中存储，并且这些位数将会在 ⓓ1+1 和 ⓓ1+2 中转换为 ASCII 码，转换为十六进制 BIN 值，并且存储起来



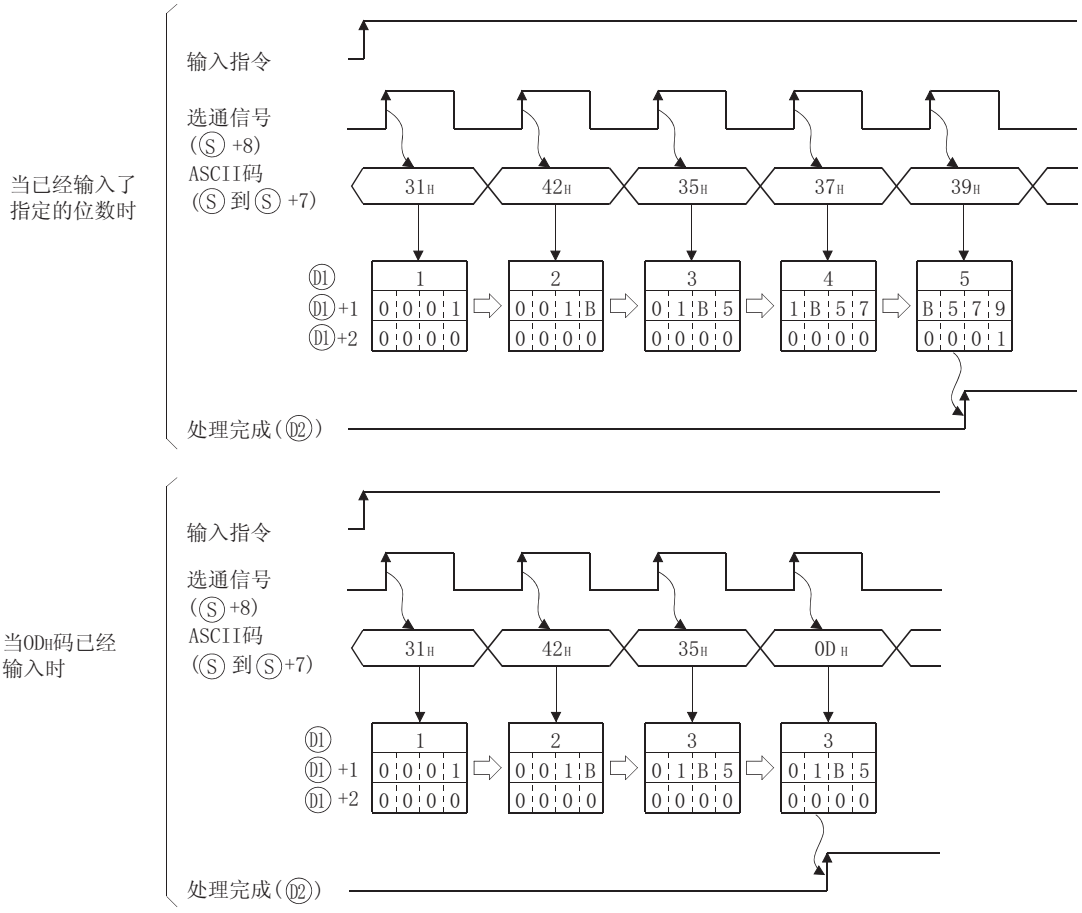
(6) 能够被 n 指定的位数为 1 到 8。

(7) 当进行了以下输入时，对输入数据内部的读取将结束，(D2) 指定的位软元件将 ON。

• 输入了 n 指定的位数时。

• 输入了 “0DH”码时。

例如，如果 n=5，在指定位置上的运算表示如下：



如果输入处理要执行第二次，就有必要清除存储在 (D1) 上的输入数据和输入的位数，并且要运行一个用户程序，该用户程序将由 (D2) 指定的位软元件变为 OFF。

如果 (D1) 没有被清除并且 (D2) 没有被变为 OFF，那下一个输入处理就不能被执行。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

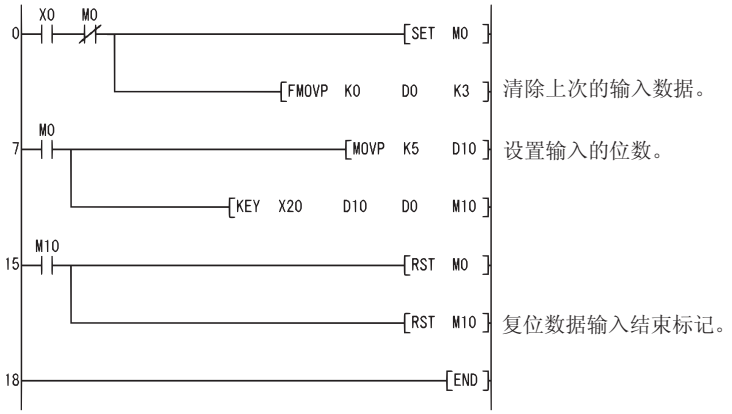
• 被 (D) 指定的软元件不是一个输入 (X) 软元件。 (出错代码：4100)

• 被 n 指定的位数在 1 到 8 的范围之外。 (出错代码：4100)

程序示例

(1) 当 X0 变为 ON 时，下列程序合并来自连接的连接到 X20 到 X28 的数字键盘的 5 位或更少位的数据，并且在从 D0 开始的位置存储这些数据。

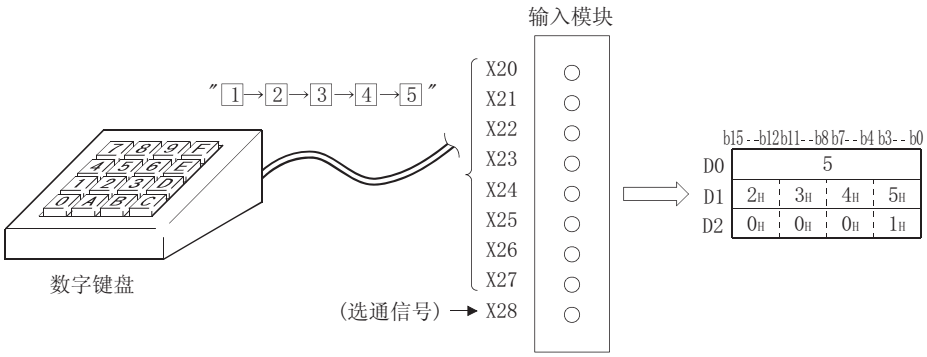
[梯形图模式]



[列表模式]

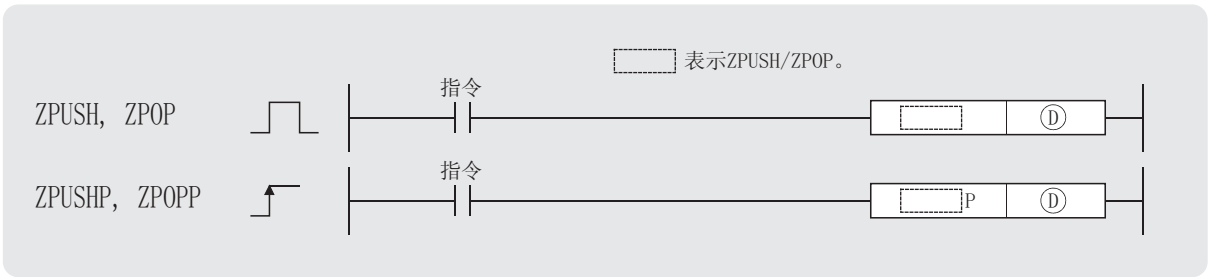
步	指令	软元件			
0	LD	X0			
1	ANI	M0			
2	SET	M0			
3	FMOVP	K0	D0	K3	
7	LD	M0			
8	MOVP	K5	D10		
10	KEY	X20	D10	D0	M10
15	LD	M10			
16	RST	M0			
17	RST	M10			
18	END				

[动作]



7.18.8 批量保存或变址寄存器的恢复(ZPUSH(P)、ZPOP(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



Ⓓ : 保存 / 恢复变址寄存器的内容的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	---	○		---					

功 能

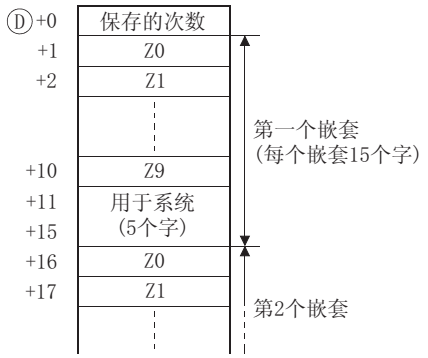
ZPUSH

(1) 将下列变址寄存器的内容保存到由 Ⓓ 指定的软元件的后面。
(当变址寄存器的内容已保存， Ⓓ + 0(保存的次数) 会增加 1。.)
- 基本类型 QCPU: ..... Z0 到 Z9
- 高性能型 QCPU/ 过程 CPU/QnACPU/ 冗余 CPU/Q4ARCPU: ..... Z0 到 Z15
- 通用型 QCPU: ..... Z0 到 Z19

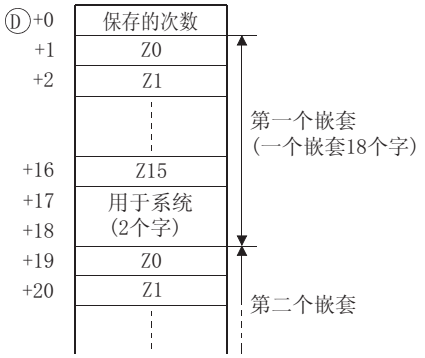
(2) ZPOP 指令是用于数据恢复的，在 ZPUSH 到 ZPOP 的周期循环中可以使用嵌套。

(3) 如果已经有嵌套，每当 ZPUSH 指令被执行时，在 Ⓓ 之后使用的区域将会被添加上去，所以这个区域从开始就要保持足够大，从而能容纳该指令将要被使用的次数。

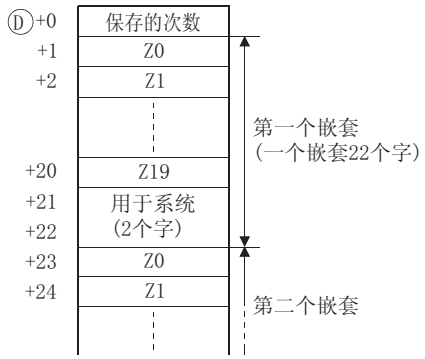
(4) 在 Ⓓ 之后使用的区域的组成如下：
- 使用基本型 QCPU 时



- 使用高性能型 QCPU/ 过程 CPU/ 冗余 CPU/QnACPU/Q4ARCPU 时



• 使用通用型 QCPU 时



ZPOP

(1) 将 Ⓓ 指定的软元件之后的区域内保存的内容恢复到变址寄存器中。(当保存的内容被读取到变址寄存器， Ⓓ + 0(保存的次数) 将被减 1。)

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

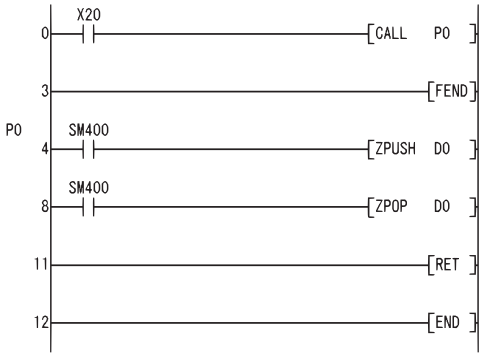
• 要在 Ⓓ 中使用的和以后被 ZPUSH(P) 指令使用的点的个数范围超过了相应的软元件范围。
(出错代码： 4101)

• 在 ZPOP(P) 指令中 Ⓓ+ 0 的内容 (已进行保存的个数) 为 0。 (出错代码： 4100)

程序示例

(1) 下列程序在调用使用变址寄存器的 P0 之后的子函数之前，将变址寄存器的内容保存到 D0 之后的区域中去。

[梯形图模式]

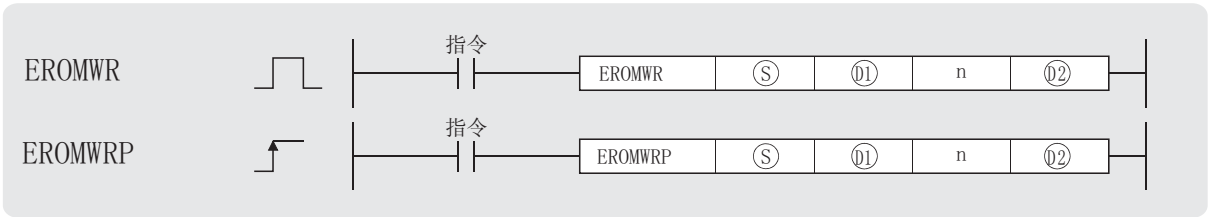


[列表模式]

步	指令	软元件
0	LD	X20
1	CALL	P0
3	FEND	
4		P0
5	LD	SM400
6	ZPUSH	D0
8	LD	SM400
9	ZPOP	D0
11	RET	
12	END	

7.18.9 对 E²PROM 文件寄存器的批写入操作 (EROMWR(P))

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 存储写入数据的软元件的起始号 (BIN 16 位)
Ⓓ1 : 写入目标文件寄存器的软元件的起始号 (BIN 16 位)
n : 要写入的字数 (BIN 16 位)
Ⓓ2 : 结束位软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	---	△*1	---	---					---
Ⓓ1	---	---	△*1	---					---
n	△*1	△*1	△*1	○					---
Ⓓ2	△*1	---	---	---					---

*1 : 不能使用局部软元件和各个程序设置的文件寄存器。

功 能

(1) 从由 Ⓢ 指定的软元件开始写入 n 个字的数据到由 Ⓓ1 指定的 E²PROM 驱动器中的文件寄存器中去。

(2) 当写入操作已经结束时，由 Ⓓ2 指定的位软元件被变为 ON，并且在一个扫描周期之后再自动变为 OFF。

(3) 数据写入操作是被 END 处理执行的，在每一个处理周期内写入 64 个字。因此，指定字数的写入操作需要 n 除以 64 个扫描周期 (小数部分进位)。
另外注意，处理过程的扫描时间被延长大约 10ms。

(4) 不要在写入处理过程中更新 Ⓢ 之后的数据。
如果在写入处理过程中更新了 Ⓢ 之后的数据，将会丢失一些数据。

运算错误

(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 被 n 指定的范围超过了 Ⓢ 或 Ⓓ1 的相关软元件的范围。 (出错代码 : 4101)
- 被 Ⓓ1 指定的文件寄存器不存在或者不是一个 E²PROM 文件寄存器。 (出错代码 : 4101)

索引

[符号]

-(BIN16 位减法运算) 6-19
$+(链接字符串) ..... 6-59、6-61
$=、$<>、$>、$<=、$<、$>=
(字符串数据比较) 6-11
$MOV(字符串转移) 6-106
*(BIN16 位乘法运算) 6-27
/(BIN16 位除法运算) 6-27
+(BIN16 位加法运算) 6-19
<(BIN16 位数据比较) 6-2
<=(BIN16 位数据比较) 6-2
<>(BIN16 位数据比较) 6-2
=(BIN16 位数据比较) 6-2
>(BIN16 位数据比较) 6-2
>=(BIN16 位数据比较) 6-2

[数字]

16 位否定移动 (CML) 6-108
16 位数据的 4 位分解 (DIS) 7-71
16 位数据的 4 位链接 (UNI) 7-73
16 位数据的 n 位右移 (SFR) 7-46
16 位数据的 n 位左移 (SFL) 7-46
16 位数据的逻辑和 (WOR) 7-11
16 位数据的逻辑积 (WAND) 7-3
16 位数据的右旋 (ROR、RCR) 7-35
16 位数据的总和计算 (WSUM) 7-93
16 位数据的最大值搜索 (MAX) 7-83
16 位数据的最小值搜索 (MIN) 7-86
16 位数据的左旋 (ROL、RCL) 7-38
16 位数据检查 (SUM) 7-63
16 位数据交换 (XCH) 6-116
16 位数据块转移 (FMOV) 6-114
16 位数据搜索 (SER) 7-60
16 位数据异或运算 (WXOR) 7-19
16 位数据与或运算 (WXNR) 7-27
16 位死区控制 (BAND) 7-314
16 位移动 (MOV) 6-100
1 相输入增大或减小 (UDCNT1) 6-138
256 到 8 位编码 (ENCO) 7-67
2 相输入增大或减小 (UDCNT2) 6-141
32 位否定移动 (DCML) 6-108
32 位数据的逻辑和 (DOR) 7-11
32 位数据的逻辑积 (DAND) 7-3
32 位数据的右旋 (DROR、DRCL) 7-41
32 位数据的总和计算 (DWSUM) 7-95
32 位数据的最大值搜索 (DMAX) 7-83
32 位数据的最小值搜索 (DMIN) 7-86
32 位数据的左旋 (DROL、DRCL) 7-44
32 位数据检查 (DSUM) 7-63
32 位数据交换 (DXCH) 6-116
32 位数据搜索 (DSER) 7-60
32 位数据异或运算 (DXOR) 7-19
32 位数据与或运算 (DXNR) 7-27
32 位死区控制 (DBAND) 7-314
32 位转移 (DMOV) 6-100
7 段解码 (SEG) 7-69
8 位到 256 位解码 (DECO) 7-65

[A]

ACOS(浮点数据的 COS-1 运算 (单精度)) 7-265
ACOSD(浮点数据的 COS-1 运算 (双精度)) 7-267
ADRSET(间接地址读取) 7-367
ANB(梯形图块系列) 5-7
AND(=、<>、>、<=、<、>=)
(BIN16 位数据比较) 6-2
AND(A 触点串行连接) 5-2
AND(D=、D<>、D>、D<=、D<、D>=)
(BIN32 位数据比较) 6-4
AND(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较 (单精度)) 6-6
AND(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较 (双精度)) 6-8
AND($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较) 6-11
ANDF(脉冲串行连接 / 上升沿) 5-5
ANDP(脉冲串行连接 / 下降沿) 5-5
ANI(B 触点串行连接) 5-2
ASC(十六进制 BIN 到 ASCII 的转换) 7-230
ASCII 到十六进制 BIN 的转换 (HEX) 7-232
ASCII 码 LED 显示指令 (LED) 7-159
ASIN(浮点数据的 SIN-1 运算 (单精度)) 7-260
ASIND(浮点数据的 SIN-1 运算 (双精度)) 7-263
ATAN(浮点数据的 TAN-1 运算 (单精度)) 7-269
ATAND(浮点数据的 TAN-1 运算 (双精度)) 7-271
A 触点并行连接 (OR) 5-2
A 触点串行连接 (AND) 5-2
A 触点运算开始 (LD) 5-2

[B]

B-(BCD4 位减法运算) 6-31
B*(BCD4 位乘法运算) 6-39
B/(BCD4 位除法运算) 6-39
B+(BCD4 位加法运算) 6-31
BACOS(BCD 类型 COS-1 运算) 7-307
BAND(16 位死区控制) 7-314
BASIN(BCD 类型 SIN-1 运算) 7-305
BATAN(BCD 类型 TAN-1 运算) 7-303
BCD(BIN 数据到 4 位数) 6-67
BCD4 位数据乘法和除法运算 (B*、B/) 6-39
BCD4 位数据到十进制 ASCII 的转换 (BCDDA) ... 7-190
BCD4 位数据加法和减法运算 (B+、B-) 6-31
BCD4 位数据平方根 (BSQR) 7-296
BCD8 位加法和减法运算 (DB+、DB-) 6-35
BCD8 位数据乘法和除法运算 (DB*、DB/) 6-41
BCD8 位数据到十进制 ASCII 的转换 (DBCDDA) .. 7-190
BCD8 位数据平方根 (BDSQR) 7-296
BCDDA(BCD4 位数据到十进制 ASCII 的转换) ... 7-190
BCD 的转换
  BIN 数据到 BCD4 位数据的转换 (BCD) 6-67
  BIN 数据到 BCD8 位数据的转换 (DBCD) 6-67
BCD 格式数据到浮点数 (EREXP) 7-246
BCD 类型 COS-1 运算 (BACOS) 7-307
BCD 类型 COS 运算 (BCOS) 7-301
BCD 类型 SIN-1 运算 (BASIN) 7-305

BCD 类型 SIN 运算 (BSIN) 7-299
BCD 类型 TAN-1 运算 (BATAN) 7-309
BCD 类型 TAN 运算 (BTAN) 7-303
BCOS(BCD 类型 COS 运算) 7-301
BDSQR(BCD8 位数据平方根) 7-296
BIN(BCD4 位数据到 BIN 数据) 6-69
BIN 16 位乘法和除法运算 (*、/) 6-27
BIN 16 位到 BIN32 位的转换 (DBL) 6-82
BIN 16 位到格雷码的转换 (GRY) 6-84
BIN 16 位加法和减法运算 (+、-) 6-19
BIN 16 位数据比较 (=、<>、>、<=、<、>=) 6-2
BIN 16 位数据排序 (SORT) 7-89
BIN 32 位乘法和除法运算 (D*、D/) 6-29
BIN 32 位到 BIN16 位的转换 (WORD) 6-83
BIN 32 位到格雷码的转换 (DGRY) 6-84
BIN 32 位加法和减法运算 (D+、D-) 6-23
BIN 32 位数据比较
(D=、D<>、D>、D<=、D<、D>=) 6-4
BIN 32 位数据排序 (DSORT) 7-89
BIN16 位到浮点数据的转换 (单精度)(FLT) 6-72
BIN16 位到浮点数据的转换 (双精度)(FLTD) ... 6-75
BIN16 位到十进制 ASCII 的转换 (BINDA) 7-184
BIN16 位到十六进制 ASCII 的转换 (BINHA) 7-187
BIN16 位到字符串的转换 (STR) 7-209
BIN16 位数据的 2 位补码 (NEG) 6-88
BIN16 位数据的死区控制 (ZONE) 7-317
BIN32 位到浮点数据的转换 (单精度)
(DFLT) 6-72
BIN32 位到浮点数据的转换 (双精度)
(DFLTD) 6-75
BIN32 位到十进制 ASCII 的转换 (DBINDA) 7-184
BIN32 位到十六进制 ASCII 的转换 (DBINHA) ... 7-187
BIN32 位到字符串的转换 (DSTR) 7-209
BIN32 位数据的 2 位补码 (DNEG) 6-88
BIN32 位数据的上限和下限控制 (DLIMIT) 7-311
BIN32 位数据的死区控制 (DZONE) 7-317
BINDA(BIN16 位到十进制 ASCII 的转换) 7-184
BINHA(BIN16 位到十六进制 ASCII 的转换) 7-187
BIN 的转换
  BCD4 位数据到 BIN16 位的转换 (BIN) 6-69
  BCD8 位数据到 BIN32 位的转换 (DBIN) 6-69
  浮点数据到 BIN16 位的转换 (单精度)
  (INT) 6-77
  浮点数据到 BIN16 位的转换 (双精度)
  INTD) 6-80
  浮点数据到 BIN32 位的转换 (单精度)
  (DINT) 6-77
  浮点数据到 BIN32 位的转换 (双精度)
  (DINTD) 6-80
BIN 块数据比较 (BKCMP □) 6-15
BK-(块减法运算) 6-56
BK+(块加法运算) 6-56
BKAND(块逻辑积) 7-9
BKBCD(块 BIN16 位数据到块 BCD4 位数据的转换) 6-92
BKBIN(块 BCD4 位数据到块 BIN16 位数据的转换) 6-94
BKCMP □ (BIN 块数据比较) 6-15
BKOR(块逻辑和运算) 7-17
BKRST(位软元件的批量复位) 7-58
BKXNR(块与或运算) 7-33
BKXOR(块异或运算) 7-25
BMOV(16 位数据块移动) 6-111
BREAK(FOR 到 NEXT 指令循环的强制结束) 7-100
BRST(字软元件的位复位) 7-53
BSET(字软元件的位置位) 7-53
BSFL(n 位数据的 1 位左移) 7-49
BSFR(n 位数据的 1 位右移) 7-49
BSIN(BCD 类型 SIN 运算) 7-299
BSQR(BCD4 位数据平方根) 7-296
BTAN(BCD 类型 TAN 运算) 7-303
BTOW(以字节为单位进行数据链接) 7-79
BXCH(块 16 位数据交换) 6-118
B 触点运算开始 (LDI) 5-2
报警器输出 (OUT F) 5-24
比较 (BIN16 位数据) 6-2
比较 (BIN32 位数据) 6-4
比较 (字符串数据) 6-11
比较操作指令 6-2
比较操作指令列表 2-10
编程注意事项 3-22
变址寄存器的批量存储 (ZPUSH) 7-372
变址寄存器的批量恢复 (ZPOP) 7-372
变址修饰 3-12
变址修饰中的修改值的指定 (IXDEV、IXSET) ... 7-133
并行连接 (OR、ORI) 5-2
并行连接 (ORB) 5-7

[C]

CALL(子程序调用) 7-102
CHKCIR(改变 CHK 指令的检查格式) 7-170
CHKEND(改变 CHK 指令的检查格式) 7-170
CHKST、CHK(特殊格式故障检查) 7-166
CJ(指针分支指令) 6-121
CML(16 位否定移动) 6-108
COM(刷新指令) ..... 7-126、9-63
COMRD(读软元件注释数据) 7-202
COS(浮点数据的 COS 运算 (单精度)) 7-252
COSD(浮点数据的 COS 运算 (双精度)) 7-254
CPU 启动时的动作模式设定指令 (S.STMODE) 10-2
CPU 切换时的动作模式设定指令 (S.CGMODE) 10-4
步数 3-28
乘法
  BCD4 位数据 (B*) 6-39
  BCD8 位数据 (DB*) 6-41
  BIN16 位 (*) 6-27
  BIN32 位 (D*) 6-29
  浮点数据乘法运算 (单精度)(E*) 6-51
  浮点数据乘法运算 (双精度)(ED*) 6-53
程序备用指令 (PSTOP) 7-349
程序低速执行登记 (PLOW) 7-354
程序分支指令表 2-25
程序跟踪的复位 (PTRAR) 7-182
程序跟踪的设置 (PTRA) 7-182
程序跟踪的执行 (PTRAEXE) 7-182
程序控制指令 2-49
程序扫描执行登记指令 (PSCAN) 7-352
程序输出 OFF 备用指令 (POFF) 7-350

索引

程序文件之间的输出 OFF 调用 (EFCALL)..... 7-117
程序执行控制指令表 2-25
程序执行状态检查指令 (PCHK)..... 7-356
出栈 (MPS)..... 5-9
除法运算
BCD4 位数据 (B/) 6-39
BCD8 位除法运算 (DB/) 6-41
BIN16 位 (/) 6-27
浮点数据的除法运算 (单精度)(E/)..... 6-51
浮点数据的除法运算 (双精度)(ED/)..... 6-53
触点指令 2-6
触点指令
并行连接 (OR、ORI)..... 5-2
串行连接 (AND、ANI)..... 5-2
脉冲并行连接 (ORF、ORP)..... 5-5
脉冲串行连接 (ANF、ANP)..... 5-5
脉冲运算开始 (LDF、LDP)..... 5-5
运算开始 (LD、LDI)..... 5-2
串行连接 (ANB) 5-7
串行连接 (AND、ANI)..... 5-2
从本地站读取软元件数据 (ZNRD)..... 8-109
从标准 ROM 读数据 (S.DEVLD)..... 9-33
从程序存储器中卸载程序 (PUNLOADP)..... 9-38
从存储器载入程序 (PLOADP)..... 9-35
从其它站的 CPU 共享存储器读取数据 (FROM)... 9-57
从其它站读取软元件数据 (ZNRD)..... 8-105
从数据表读取最旧数据 (FIFR)..... 7-138
从数据表读取最新数据 (FPOP)..... 7-140
从数据列表中删除数据 (FDEL)..... 7-142
从右边提取字符串数据 (RIGHT)..... 7-234
从指定文件中读取数据 (SP.FREAD)..... 9-19
从智能功能模块读取 2 字数据 (DFRO)..... 7-145
从智能功能模块中读取 1 字数据 (FROM)..... 7-145
从中断程序中恢复 (IRET)..... 6-134
从子程序的返回 (RET)..... 7-107
从左边提取字符串数据 (LEFT)..... 7-234
错误显示和报警器复位指令 (LEDR) 7-163

[D]

D-(BIN32 位数据减法运算) 6-23
D*(BIN32 位数据乘法运算) 6-29
D/(BIN32 位数据除法运算) 6-29
D+(BIN32 位数据加法运算) 6-23
D=、D<>、D>、D<=、D<、D>=
(BIN32 位数据比较) 6-4
DABCD(十进制 ASCII 到 BCD4 位数据的转换).. 7-199
DABIN(十进制 ASCII 到 BIN16 位的转换) 7-193
DAND(32 位数据的逻辑乘积) 7-3
DATE-(时钟数据减法运算) 7-337
DATE+(时钟数据加法运算) 7-335
DATERD(读时钟数据) 7-327
DATEWR(写时钟数据) 7-331
DB-(BCD8 位减法运算) 6-35
DB*(BCD8 位乘法运算) 6-41
DB/(BCD 8 位除法运算) 6-41
DB+(BCD8 位加法运算) 6-35
DBAND(32 位死区控制) 7-314
DBCD(BIN 到 BCD8 位数的转换) 6-67
DBCDDA(BCD8 位数据到十进制 ASCII 的转换).. 7-190
DBIN(BCD8 位数据到 BIN16 位的转换) 6-69
DBINDA(BIN32 位到十进制 ASCII 的转换) 7-184
DBINHA(BIN32 位到十六进制 ASCII 的转换) ... 7-187
DBL(BIN16 位到 BIN32 位的转换) 6-82
DCML(32 位否定传送) 6-108
DDABCD
(十进制 ASCII 到 BCD8 位数据的转换) 7-199
DDABIN(十进制 ASCII 到 BIN32 位的转换) 7-193
DDEC(32 位 BIN 数据递减) 6-65
DEC(16 位 BIN 数据递减) 6-63
DECO(8 位到 256 位解码) 7-65
DEG(浮点弧度到角度的转换 (单精度)) 7-277
DEGD(浮点弧度到角度的转换 (双精度)) 7-279
DELTA(直接输出的脉冲转换) 5-38
DFLT(BIN32 位到浮点数据的转换
(单精度)) 6-72
DFLTD(BIN32 位到浮点数据的转换
(双精度)) 6-75
DFRO(从智能功能模块读取 2 字数据) 7-145
DGBIN(格雷码到 BIN16 位的转换) 6-86
DGRY(BIN32 位到格雷码的转换) 6-84
DHABIN(十六进制 ASCII 到 BIN32 位的转换).. 7-196
DI(中断禁止) 6-125
DINC(递增 32 位 BIN) 6-65
DINT(浮点数据到 BIN32 位的转换
(单精度)) 6-77
DINTD(浮点数据到 BIN32 位的转换
(双精度)) 6-80
DIS(16 位数据的 4 位分组) 7-71
DLIMIT(BIN32 位数据的上限和下限控制) 7-311
DMAX(32 位数据的最大值搜索) 7-83
DMIN(32 位数据的最小值搜索) 7-86
DMOV(32 位转移) 6-100
DNEG(BIN32 位数据的 2 位补码) 6-88
DOR(32 位数据的逻辑和) 7-11
DRCL(32 位数据的左循环) 7-44
DRCR(32 位数据的右循环) 7-41
DROL(32 位数据的左循环) 7-44
DROR(32 位数据的右循环) 7-41
DSER(32 位数据搜索) 7-60
DSFL(n 字数据左移 1 字) 7-51
DSFR(n 字数据右移 1 字) 7-51
DSORT(BIN32 位数据排序) 7-89
DSTR(BIN32 位到字符串的转换) 7-209
DSUM(32 位数据检查) 7-63
DTEST(位测试) 7-55
DTO(智能功能模块 2 字数据写运算) 7-148
DUTY(定时脉冲的产生) 7-360
DVAL(字符串到 BIN32 位的转换) 7-215
DWSUM(32 位数据的总数计算) 7-95
DXCH(16 位数据交换) 6-116
DXNR(32 位数据否定排它逻辑和运算) 7-27
DXOR(32 位排它或运算) 7-19
DZONE(BIN32 位数据的死区控制) 7-317
打印 ASCII 码指令 (PR)..... 7-151
打印注释指令 (PRC)..... 7-154
单精度到双精度的转换 (ECON)..... 6-96

登记路由信息 (RTWRITE)..... 8-133
低速定时器 (OUT T)..... 5-18
低速累计定时器 (OUTH ST)..... 5-18
递减
16 位 BIN(DEC) 6-63
32 位 BIN(DDEC) 6-65
递增
BIN 16 位 (INC) 6-63
BIN 32 位 (DINC) 6-65
调试和故障诊断指令 2-39
定时脉冲的产生 (DUTY)..... 7-360
定时器 (OUT T)..... 5-18
读路径参数 (RTREAD)..... 8-129
读模块信息 (UNIRD)..... 9-2
读取 (MRD)..... 5-9
读取扩展时钟数据 (S.DATERD)..... 9-66
读软元件注释数据 (COMRD)..... 7-202
读时钟数据 (DATERD)..... 7-327

[E]

E-(浮点数减法运算)(单精度))..... 6-43、6-45
E*(浮点数乘法运算)(单精度))..... 6-51
E/(浮点数除法运算)(单精度))..... 6-51
E+(浮点数加法运算)(单精度))..... 6-43、6-45
E=、E<>、E>、E<=、E<、E>=
(浮点数据比较(单精度))..... 6-6
ECALL(在程序文件之间的子集调用)..... 7-112
ECON(单精度到双精度的转换)..... 6-96
ED-(浮点数减法运算)(双精度))..... 6-47、6-49
ED*(浮点数乘法运算)(双精度))..... 6-53
ED/(浮点数除法运算)(双精度))..... 6-53
ED+(浮点数加法运算)(双精度))..... 6-47、6-49
ED=、ED<>、ED>、ED<=、ED<、ED>=
(浮点数据比较(双精度))..... 6-8
EDCON(双精度到单精度的转换)..... 6-98
EDMOV(浮点数据转移(双精度))..... 6-104
EDNEG(浮点数的符号取反(双精度)) 6-91
EFCALL(在程序文件之间的输出 OFF 调用)..... 7-117
EGF(脉冲运算结果/上升沿)..... 5-14
EGP(脉冲运算结果/下降沿)..... 5-14
EI(中断允许)..... 6-125
EMOD(浮点数到 BCD)..... 7-243
EMOV(浮点数据转移(单精度))..... 6-102
ENCO(256 到 8 位编码)..... 7-67
END(结束顺控程序)..... 5-49
ENEG(浮点数符号取反(单精度))..... 6-90
EREXP(BCD 格式数据到浮点数)..... 7-246
EROMWR(批量写运算到 E2PROM 文件寄存器)... 7-374
ESTR(浮点数到字符串的转换)..... 7-220
EVAL(字符串到浮点数的转换)..... 7-220
EXP(浮点数据的指数运算(单精度))..... 7-285
EXPD(浮点数据的指数运算(双精度))..... 7-288

[F]

FCALL(子程序输出 OFF 调用)..... 7-108
FDEL(从数据表中删除数据)..... 7-142
FEND(结束主程序)..... 5-47
FF(位软元件输出取反)..... 5-36
FIFR(从数据表读取旧数据)..... 7-138
FIFW(写数据到数据表)..... 7-136
FINS(在数据表中插入数据)..... 7-142
FLT(BIN16 位到浮点数据的转换(单精度))..... 6-72
FLTD(BIN16 位到浮点数据的转换(双精度))... 6-75
FMOV(16 位数据块传送)..... 6-114
FOR(FOR 到 NEXT)..... 7-97
FOR 到 NEXT(FOR、NEXT)..... 7-97
FOR 到 NEXT 指令循环的强制结束 (BREAK)..... 7-100
FPOP(从数据表读取新数据)..... 7-140
FROM(从智能功能模块读取 1 字数据)..... 7-145
FROM(其它站点的 CPU 共享存储区读取数据)... 9-57
发送数据到其它站点 (SEND)..... 8-62
浮点弧度到角度的转换(单精度)(DEG)..... 7-277
浮点弧度到角度的转换(双精度)(DEGD)..... 7-279
浮点角度到弧度的转换(单精度)(RAD)..... 7-273
浮点角度到弧度的转换(双精度)(RADD)..... 7-275
浮点数到 BCD(EMOD)..... 7-243
浮点数据比较(单精度)
(E=、E<>、E>、E<=、E<、E>=) 6-6
浮点数据比较(双精度)
(ED=、ED<>、ED>、ED<=、ED<、ED>=) 6-8
浮点数据乘除法运算(单精度)(E*、E/)..... 6-51
浮点数据乘除法运算(双精度)(ED*、ED/)..... 6-53
浮点数据传送(单精度)(EMOV)..... 6-102
浮点数据传送(双精度)(EDMOV)..... 6-104
浮点数据到字符串的转换 (ESTR)..... 7-220
浮点数据的 COS^{-1} 运算(单精度)(ACOS) 7-265
浮点数据的 COS^{-1} 运算(双精度)(ACOSD) 7-267
浮点数据的 COS 运算(单精度)(COS)..... 7-252
浮点数据的 COS 运算(双精度)(COSD)..... 7-254
浮点数据的 SIN^{-1} 运算(单精度)(ASIN) 7-260
浮点数据的 SIN^{-1} 运算(双精度)(ASIND) 7-263
浮点数据的 SIN 运算(单精度))(SIN)..... 7-248
浮点数据的 SIN 运算(双精度))(SIND)..... 7-250
浮点数据的 TAN^{-1} 运算(单精度)(ATAN) 7-269
浮点数据的 TAN^{-1} 运算(双精度)(ATAND) 7-271
浮点数据的 TAN 运算(单精度)(TAN)..... 7-256
浮点数据的 TAN 运算(双精度)(TAND)..... 7-258
浮点数据的指数运算(单精度)(EXP)..... 7-285
浮点数据的指数运算(双精度)(EXPD)..... 7-288
浮点数据的转换(单精度)(FLT、DFLT)..... 6-72
浮点数据的转换(双精度)(FLTD、DFLTD)..... 6-75
浮点数据的自然对数运算(单精度)(LOG)..... 7-290
浮点数据的自然对数运算(双精度)(LOGD).... 7-292
浮点数据符号取反(单精度)(ENEG) 6-90
浮点数据符号取反(双精度)(EDNEG)..... 6-91
浮点数区域的平方根运算(单精度)(SQR)..... 7-281
浮点数区域的平方根运算(双精度)(SQRD).... 7-283
浮动点数据的加法和减法运算(单精度)
(E+、E-) ..... 6-43、6-45
浮动点数据的加法和减法运算(双精度)
(ED+、ED-) ..... 6-47、6-49
复位报警器 (RST F)..... 5-31
复位采样跟踪 (STRAR)..... 7-180
复位看门狗定时器 (WDT)..... 7-358
复位软元件 (RST)..... 5-28、5-31
复位主控制 (MCR)..... 5-43
复位状态闭锁 (SLTR)..... 7-178

索引

[G]

GBIN(格雷码到BIN16位的转换)..... 6-86
GOEND(跳转到END)..... 6-124
GRY(BIN16位到格雷码的转换)..... 6-84
改变CHK指令的检查格式(CHKCIR、CHKEND)... 7-170
高速定时器(OUTH T)..... 5-18
高速累计定时器(OUTH ST)..... 5-18
格雷码到BIN16位的转换(GBIN)..... 6-86
格雷码到BIN32位的转换(DGBIN)..... 6-86

[H]

HABIN(十六进制ASCII到BIN16位的转换)... 7-196
HEX(ASCII到十六进制BIN的转换)..... 7-232
HOUR(改变时间数据格式)..... 7-341
恒定周期脉冲输出(PLSY)..... 6-157
缓冲存储区访问指令..... 2-38
缓冲存储区批量刷新指令(S.SPREF)..... 10-10
换页(NOPLF)..... 5-53
换页(PAGE n)..... 5-53

[I]

I/O刷新(RFS)..... 6-136
I/O刷新指令列表..... 2-25
IMASK(中断程序掩码)..... 6-125
INC(递增16位BIN)..... 6-63
INSTR(字符串搜索)..... 7-241
INT(浮点数据到BIN16位(单精度))..... 6-77
INTD(浮点数据到BIN16位(双精度))..... 6-80
INV(运算结果取反)..... 5-12
IRET(从中断程序恢复)..... 6-134
IX、IXEND(整个梯形图的变址修饰)..... 7-129
IXDEV(变址修饰中修饰值的指定)..... 7-133
IXSET(变址修饰中修饰值的指定)..... 7-133

[J]

JMP(指针分支)..... 6-121
基本指令..... 2-10
计数器(OUT C)..... 5-22
加法
  BCD4位加法运算(B+)..... 6-31
  BCD8位加法运算(DB+)..... 6-35
  BIN16位加法运算(+)..... 6-19
  BIN32位加法运算(D+)..... 6-23
  浮点数据的加法运算
  (单精度)(E+)..... 6-43、6-45
  浮点数据的加法运算
  (双精度)(ED+)..... 6-47、6-49
  块加法运算(BK+)..... 6-56
间接地址读取(ADRSET)..... 7-367
间接指定..... 3-18
减法
  BCD4位减法运算(B-)..... 6-31
  BCD8位减法运算(DB-)..... 6-35
  BIN16位减法运算(-)..... 6-19
  BIN32位减法运算(D-)..... 6-23
  浮点数减法运算(单精度)(E-)..... 6-43、6-45
  浮点数减法运算(双精度)(ED-)..... 6-47、6-49
  块的减法运算(BK-)..... 6-56
将n位的数据右移1位(BSFR)..... 7-49
将n位的数据左移1位(BSFL)..... 7-49
将n字的数据右移1字(DSFR)..... 7-51
将n字的数据左移1字(DSFL)..... 7-51
教学定时器(TTMR)..... 6-144
接收
  浮点数的符号取反(单精度)(ENEG)..... 6-90
  浮点数的符号取反(双精度)(EDNEG)..... 6-91
  位软元件输出取反(FF)..... 5-36
  运算结果取反(INV)..... 5-12
结构体创建指令..... 2-35
结束顺控程序(END)..... 5-49
结束主程序(FEND)..... 5-47
矩阵输入(MTR)..... 6-161

[K]

KEY(通过键盘的数字键输入)..... 7-368
块16位交换(BXCH)..... 6-118
块16位转移(BMOV)..... 6-111
块BCD4位数数据到块BIN16位数据的转换
(BKBIN)..... 6-94
块BIN16位数据到BCD4位数据的转换
(BKBCD)..... 6-92
块加法运算(BK+)..... 6-56
块减法运算(BK-)..... 6-56
块逻辑乘(BKAND)..... 7-9
块逻辑和运算(BKOR)..... 7-17
块异或运算(BKXOR)..... 7-25
块与或运算(BKXNR)..... 7-33
扩展时钟数据的加法运算(S.DATE+)..... 9-66
扩展时钟数据的减法运算(S.DATE-)..... 9-66

[L]

LD($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较)..... 6-11
LD(=、<>、>、<=、<、>=)
(BIN16位数据比较)..... 6-2
LD(A触点运算开始)..... 5-2
LD(D=、D<>、D>、D<=、D<、D>=)
(BIN32位数据比较)..... 6-4
LD(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较(单精度))..... 6-6
LD(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较(双精度))..... 6-8
LDF(脉冲运算开始/下降沿)..... 5-5
LDI(B触点运算开始)..... 5-2
LDP(脉冲运算开始/上升沿)..... 5-5
LED(ASCII码LED显示指令)..... 7-159
LEDC(注释的LED显示指令)..... 7-161
LEDR(错误显示和报警器复位指令)..... 7-163
LEFT(从左边提取字符串数据)..... 7-234
LEN(字符串长度检测)..... 7-207
LIMIT(BIN16位的上下限控制)..... 7-311
LOG(浮点数据的自然对数运算(单精度))..... 7-290
LOGD(浮点数据的自然对数运算(双精度))... 7-292
来自其它站点的瞬时请求(REQ)..... 8-81
来自外围软元件的键盘输入(PKEY)..... 7-345
连接指令

连接指令列表 2-7
链接字符串 ($+) 6-59
梯形图块并行连接 (ORB) 5-7
梯形图块串行连接 (ANB) 5-7
链接刷新指令 2-51
链接字符串 ($+) ..... 6-59、6-61
逻辑和 7-2
逻辑积 7-2
逻辑运算指令 2-27

[M]

MAX(16 位数据的最大值搜索) 7-83
MC(主控制置位) 5-43
MCR(主控制复位) 5-43
MEF(脉冲运算结果 / 下降沿) 5-13
MEP(脉冲运算结果 / 上升沿) 5-13
MIDR(字符串中任意选择) 7-237
MIDW(字符串中任意替换) 7-237
MIN(16 位数据的最小值搜索) 7-86
MOV(16 位传送) 6-100
MPP(运算结果入栈) 5-9
MPS(运算结果出栈) 5-9
MRD(操作结果读) 5-9
MSG(显示信息到外围设备) 7-343
MTR(矩阵输入) 6-161
脉冲并行连接 (ORF、ORP) 5-5
脉冲串行连接 (ANDF、ANDP) 5-5
脉冲宽度调制 (PWM) 6-159
脉冲密度测量 (SPD) 6-155
脉冲运算结果
变址继电器运算结果的脉冲转换 (EGF、EGP) .. 5-14
运算结果转换 (MEF、MEP) 5-13
脉冲运算开始 (LDF、LDP) 5-5
脉冲转换
(EGF、EGP) 5-14
(MEF、MEP) 5-13
(DELTA) 5-38

[N]

NEG(BIN16 位数据的 2 位补码) 6-88
NEXT(FOR 到 NEXT) 7-97
NOP 5-53
NOP(无处理) 5-53
NOPLF(无处理换页) 5-53
NUNI(任意数据的链接) 7-75

[O]

OR($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较) 6-11
OR(=、<>、>、<=、<、>=)
(BIN16 位数据比较) 6-2
OR(A 触点并行连接) 5-2
OR(D=、D<>、D>、D<=、D<、D>=)
(BIN32 位数据比较) 6-4
OR(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较 (单精度)) 6-6
OR(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较 (双精度)) 6-8
ORB(梯形图块并行连接) 5-7
ORF(脉冲并行连接 / 下降沿) 5-5
ORI(B 触点并行连接) 5-2
ORP(脉冲并行连接 / 上升沿) 5-5
Or 取反 (ORI) 5-2
OUT
报警器输出 (OUT F) 5-24
低速定时器 (OUT T) 5-18
低速累计定时器 (OUT ST) 5-18
高速定时器 (OUTH T) 5-18
高速累计定时器 (OUTH ST) 5-18
计数器 (OUT C) 5-22
输出 (OUT) 5-16

[P]

PAGE(无处理换页) 5-53
PCHK(程序执行状态检查指令) 7-356
PLF(下降沿输出) 5-33
PLOADP(从存储卡中装载程序) 9-35
PLOW(程序低速执行登记) 7-354
PLS(上升沿输出) 5-33
PLSY(固定周期脉冲输出) 6-157
POFF(程序输出 OFF 备用指令) 7-350
PR(打印 ASCII 码指令) 7-151
PRC(打印注释指令) 7-154
PSCAN(程序扫描执行登记指令) 7-352
PSTOP(程序备用指令) 7-349
PSWAPP(装载 + 卸载) 9-40
PTRA(程序跟踪的设置) 7-182
PTRAEXE(程序跟踪的执行) 7-182
PTRAR(程序跟踪的复位) 7-182
PUNLOADP(从程序存储器中卸载程序) 9-38
PWM(脉冲宽度调制) 6-159

[Q]

QCDSET(注释用文件的设置) 7-325
QCPU 专用指令 2-53
QDRSET(文件寄存器用文件的设置) 7-322
QnA 链接专用指令 8-18
其它方便指令 2-6
其它站点读字软元件数据 (READ) 8-18
其它站点读字软元件数据 (SREAD) 8-29
其它站点接收数据 (RECV) 8-72
其它指令 5-51
顺控指令 2-6
应用指令 2-27
切换文件寄存器号码 (RSET) 7-320
取反
位软元件输出取反 (FF) 5-36
运算结果取反 (INV) 5-12

[R]

RAD(浮点数角度到弧度的转换 (单精度)) ... 7-273
RADD(浮点数角度到弧度的转换 (双精度)) ... 7-275
RAMP(斜坡信号) 6-152
RBMOV(文件寄存器的高速块传送) 9-43
RCL(16 位数据的左旋) 7-38
RCR(16 位数据的右旋) 7-35

索引

READ(从其它站读取字软元件数据)..... 8-18
RECV(从其它站接收数据)..... 8-72
REQ(来自其它站的瞬时请求)..... 8-81
RET(从子程序的返回)..... 7-107
RFRP(从远程 I/O 站的特殊功能模块中
读数据)..... 8-121
RFS(I/O 刷新)..... 6-136
RIGHT(从右边提取字符串数据)..... 7-234
RND(随机数生成和序列更新)..... 7-294
ROL(16 位数据的左旋)..... 7-38
ROR(16 位数据的右旋)..... 7-35
ROTC(旋转台就近控制)..... 6-149
RSET(文件寄存器的块号切换)..... 7-320
RST
  复位报警器 (RST F)..... 5-31
  复位软元件 (RST)..... 5-28
RTOP(写数据到远程 I/O 站的
特殊功能模块)..... 8-125
RTREAD(读路由信息)..... 8-129
RTWRITE(登录路由信息)..... 8-133
任意数据的分解 (NDIS)..... 7-75
任意数据的链接 (NUNI)..... 7-75
冗余系统指令(对于 Q4ARCPU)..... 2-55
如何读指令..... 4-2
如何读指令表..... 2-4
入栈 (MPP)..... 5-9
软元件范围检查..... 3-22

[S]

S.CGMODE(CPU 切换时间动作模式设置)..... 10-4
S.DATE-(扩展时钟数据减法运算)..... 9-66
S.DATE+(扩展时钟数据加法运算)..... 9-66
S.DATERD(读扩展时钟数据)..... 9-66
S.DEVLD(从标准 ROM 读数据)..... 9-33
S.SPREF(缓冲存储器批量刷新)..... 10-10
S.STMODE(CPU 启动时的动作模式设置指令).... 10-2
S.TO(向 CPU 共享存储器写入)..... 9-49
S.TRUCK(数据跟踪指令)..... 10-6
SCJ(指针分支指令)..... 6-121
SECOND(改变时间数据格式)..... 7-339
SEG(7 段解码)..... 7-69
SEND(发送数据到其它站点)..... 8-62
SER(16 位数据搜索)..... 7-60
SET
  设置报警器 (SET F)..... 5-31
  设置软元件 (SET)..... 5-26
SFL(16 位数据的 n 位左移)..... 7-46
SFR(16 位数据的 n 位右移)..... 7-46
SFT(位软元件移位)..... 5-40
SIN(浮点数据的 SIN 运算)(单精度))..... 7-248
SIND(浮点数据的 SIN 运算(双精度))..... 7-250
SLT(设置状态闭锁)..... 7-178
SLTR(复位状态闭锁)..... 7-178
SORT(BIN16 位数据排序)..... 7-89
SP.CONTSW(系统转换指令)..... 11-2
SP.DEVST(写数据到标准 ROM)..... 9-31
SP.FREAD(从指定文件中读取数据)..... 9-19
SP.FWRITE(写数据到指定的文件)..... 9-8
SPD(脉冲密度测量)..... 6-155
SQR(浮点数区域的平方根运算(单精度))..... 7-281
SQRD(浮点数区域的平方根运算(双精度))... 7-283
SREAD(从其它站读取字软元件数据)..... 8-29
SRND(随机数生成和序列更新)..... 7-294
STMR(特殊功能定时器)..... 6-146
STOP(顺控程序停止)..... 5-51
STR(BIN16 位到字符串的转换)..... 7-209
STRA(设置采样跟踪)..... 7-180
STRAR(复位采样跟踪)..... 7-180
SUM(16 位数据检查)..... 7-63
SWAP(最高和最低字节交换)..... 6-120
SWRITE(写软元件数据到其它站)..... 8-51
上升沿输出 (PLS)..... 5-33
设置报警器 (SET F)..... 5-31
设置采样跟踪 (STRA)..... 7-180
设置软元件 (SET)..... 5-26、5-31
设置主控制 (MC)..... 5-43
设置状态闭锁 (SLT)..... 7-178
十进制 ASCII 到 BCD4 位的转换 (DABCD)..... 7-199
十进制 ASCII 到 BCD8 位的转换 (DDABCD)..... 7-199
十进制 ASCII 到 BIN16 位的转换 (DABIN)..... 7-193
十进制 ASCII 到 BIN32 位的转换 (DDABIN)..... 7-193
十六进制 ASCII 到 BIN16 位的转换 (HABIN).... 7-196
十六进制 ASCII 到 BIN32 位的转换 (DHABIN)... 7-196
十六进制 BIN 到 ASCII 的转换 (ASC)..... 7-230
时间检查指令 (TIMCHK)..... 7-362
时间数据转换 (HOUR)..... 7-341
时间数据转换 (SECOND)..... 7-339
时钟数据加法运算 (DATE+)..... 7-335
时钟数据减法运算 (DATE-)..... 7-337
时钟指令..... 2-48
实数数据..... 3-8
输出取反 (FF)..... 5-36
输出指令 (OUT)..... 5-16
输出指令 (OUT)..... 5-16
输出指令列表..... 2-8
数据处理指令..... 2-32
数据跟踪指令 (S.TRUCK)..... 10-6
数据控制指令..... 2-46
数据链接指令..... 2-51
数据链接指令..... 2-51
数据列表操作指令..... 2-37
数据转换指令..... 6-67
数据转换指令列表..... 2-21
数字键输入 (KEY)..... 7-368
刷新指令 (COM)..... 7-126
双精度到单精度的转换 (EDCON)..... 6-98
双字数据..... 3-6
顺控程序停止 (STOP)..... 5-51
顺控指令..... 2-6
算术运算指令..... 2-16
随机数生成 (RND/SRND)..... 7-294

[T]

TAN(浮点数据的 TAN 运算(单精度))..... 7-256
TAND(浮点数据的 TAN 运算(双精度))..... 7-258
TEST(位测试)..... 7-55
TIMCHK(时间检查指令)..... 7-362
TO(写 1 字数据到智能功能模块)..... 7-148

TRACE(跟踪设置) 9-6
TRACER(跟踪复位) 9-6
TTMR(教学定时器) 6-144
特殊定时器 (STMR) 6-146
特殊格式故障检查 (CHKST、CHK) 7-166
特殊功能指令 2-43
梯形图块并行连接 (ORB) 5-7
梯形图块串行连接 (ANB) 5-7
跳转到 END(GOEND) 6-124
通过键盘的数字键输入 (KEY) 7-368

[U]

UDCNT1(计数器 1 相输入增大或减小) 6-138
UDCNT2(计数器 2 相输入增大或减小) 6-141
UNI(16 位数据的 4 位链接) 7-73
UNIRD(读模块信息) 9-2

[V]

VAL(字符串到 BIN16 位的转换) 7-215

[W]

WAND(16 位数据的逻辑积) 7-3
WDT(复位看门狗定时器) 7-358
WOR(16 位数据的逻辑和) 7-11
WORD(BIN32 位到 BIN16 位的转换) 6-83
WRITE(向其它站写入数据) 8-40
WSUM(16 位数据的总数计算) 7-93
WTOB(以字节为单位的数据分解) 7-79
WXNR(16 位数据否定排它逻辑和运算) 7-30
WXNR(16 位数据与或运算) 7-27
WXOR(16 位数据排它或运算) ..... 7-19、7-22
外围设备指令 2-49
网络刷新指令 (ZCOM) 8-10
位测试 (TEST/DTEST) 7-55
位处理指令 2-31
位软元件的批量复位 (BKRST) 7-58
位软元件的位数指定 3-4
位软元件输出反转 (FF) 5-36
位软元件移动 (SET) 5-40
位数据 3-3
位数指定 3-4
文件寄存器 1 字节写 (ZRWRB) 7-365
文件寄存器的高速块转移 (RBMOV) 9-43
文件寄存器直接读取 1 字节数据 (ZRRDB) 7-363
无处理 (NOP、NOPLF、PAGE) 5-53

[X]

XCALL(子程序调用) 7-121
XCH(32 位数据交换) 6-116
系统转换 (SP.CONTSW) 11-2
显示信息到外围设备 (MSG) 7-343
显示指令 2-38
相关程序手册 1-2
向 E^2PROM 文件寄存器的批量写入操作
(EROMWR) 7-374
斜坡信号 (RAMP) 6-152
写 1 字数据到智能功能模块 (TO) 7-148
写 2 字数据到智能功能模块 (DTO) 7-148
写入自站 CPU 的 CPU 共享存储器 CPU- 9-47
写入自站 CPU 的 CPU 共享存储器 CPU-S.TO 9-49
写入自站 CPU 的 CPU 共享存储器 CPU-TO 9-52
写软元件数据到本地站 (ZNWR) 8-117
写软元件数据到其它站 (ZNWR) 8-113
写时钟数据 (DATEWR) 7-331
写数据到标准 ROM(SP.DEVST) 9-31
写数据到其它站 (WRITE) 8-40
写数据到数据表 (FIFW) 7-136
写数据到指定的文件 (SP.FWRITE) 9-8
写字软元件数据到其它站 (SWRITE) 8-51
循环表近路径循环移位控制 (ROTC) 6-149
循环指令 2-30

[Y]

移位指令 ..... 5-40、7-46
(应用指令) 2-27
移位指令表(顺控程序) 2-8
以字节为单位的数据分解 (WTOB) 7-79
以字节为单位的数据链接 (BTOW) 7-79
应用指令 2-27
用于文件寄存器的设置文件 (QDRSET) 7-322
用于注释的文件设置 (QCDSET) 7-325
与远程 I/O 站通讯
从远程 I/O 站的特殊功能模块中读数据
(RFRP) 8-121
从远程 I/O 站的特殊功能模块中读数据
(ZNFR) 8-95
向远程 I/O 站的特殊功能模块中写数据
(RTOP) 8-125
向远程 I/O 站的特殊功能模块中写数据
(ZNTO) 8-100
运算错误 3-22
运算结果出栈 (MPS) 5-9
运算结果读 (MRD) 5-9
运算结果取反 (INV) 5-12
运算结果入栈 (MPP) 5-9
运算开始 (LD、LDI) 5-2

[Z]

ZCOM(网络刷新指令) 8-10
ZNFR(从远程 I/O 站的特殊功能模块中
读取数据) 8-95
ZNRD(从本地站读取软元件数据) 8-109
ZNRD(从其它站读取软元件数据) 8-105
ZNTO(写数据到远程 I/O 站中的
特殊功能模块) 8-100
ZNWR(写软元件数据到本地站) 8-117
ZNWR(写软元件数据到其它站) 8-113
ZONE(BIN16 位数据的死区控制) 7-317
ZPOP(变址寄存器的批量恢复) 7-372
ZPUSH(变址寄存器的批量保存) 7-372
ZRRDB(从文件寄存器中读取 1 字节) 7-363
ZRWRB(文件寄存器中写入 1 字节) 7-365
在程序文件之间的子程序调用 (ECALL) 7-112
在数据表中插入数据 (FINS) 7-142
增大 / 减小计数器
计数器 1 相输入增大或减小 (UDCNT1) 6-138

索引

计数器 2 相输入增大或减小 (UDCNT2)..... 6-141
直接输出的脉冲转换 (DELTA)..... 5-38
指定数据 3-3
指令类型 2-2
指令执行的条件 3-27
指针分支指令 (CJ、SCJ、JMP)..... 6-121
中断程序掩码 (IMASK)..... 6-125
中断禁止 (DI)..... 6-125
中断允许 (EI)..... 6-125
终端指令表 2-9
主控制指令 5-43
注释的 LED 显示指令 (LEDC)..... 7-161
转换
BCD4 位数据到 BIN 数据的转换 (BIN) 6-69
BCD8 位数据到 BIN 数据的转换 (DBIN) 6-69
BIN16 位到浮点数据的转换 (单精度)
(FLT) 6-72
BIN16 位到浮点数据的转换 (双精度)
(FLTD) 6-75
BIN16 位到格雷码的转换 (GRY) 6-84
BIN16 位数据到 BIN32 位的转换 (DBL) 6-82
BIN32 位到 BIN16 位的转换 (WORD) 6-83
BIN32 位到浮点数据的转换 (单精度)
(DFLT) 6-72
BIN32 位到浮点数据的转换 (双精度)
(DFLTD) 6-75
BIN32 位到格雷码的转换 (DGRY) 6-84
BIN 到 BCD4 位数据的转换 (BCD) 6-67
BIN 到 BCD8 位数据的转换 (DBCD) 6-67
单精度到双精度的转换 (ECON)..... 6-96
浮点数据到 BIN16 位的转换 (单精度)
(INT) 6-77
浮点数据到 BIN16 位的转换 (双精度)
(INTD) 6-80
浮点数据到 BIN32 位的转换 (单精度)
(DINT) 6-77
浮点数据到 BIN32 位的转换 (双精度)
(DINTD) 6-80
格雷码到 BIN16 位的转换 (GBIN)..... 6-86
格雷码到 BIN32 位的转换 (DGBIN)..... 6-86
双精度到单精度的转换 (EDCON)..... 6-98
转换指令 2-47
装载 + 卸载 (PSWAPP)..... 9-40
子程序的输出 OFF 调用 (FCALL)..... 7-108
子程序调用 (CALL)..... 7-102
子程序调用 (XCALL)..... 7-121
子程序输出 OFF 调用 (FCALL)..... 7-108
子集处理 3-20
字符串长度检测 (LEN)..... 7-207
字符串处理指令 2-40
字符串到 BIN16 位的转换 (VAL)..... 7-215
字符串到 BIN32 位的转换 (DVAL)..... 7-215
字符串到浮点数据的转换 (EVAL)..... 7-226
字符串的任意选择和替换 (MIDR)..... 7-237
字符串的任意选择替换 (MIDW)..... 7-237
字符串数据 3-11
字符串数据比较 6-11
字符串搜索 (INSTR)..... 7-241
字符串转移 ($MOV)..... 6-106
字软元件的位复位 (BRST)..... 7-53
字软元件的位指定 3-3
字软元件的位置位 (BSET)..... 7-53
字数据 3-4
最高和最低字节交换 (SWAP)..... 6-120

质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱责任的故障或缺陷（以下称“故障”），则经销商或三菱服务公司将负责免费维修。

注意如果需要在国内现场或海外维修时，则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试，三菱将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或货到目的地日的一年内。

注意产品从三菱生产并出货之后，最长分销时间为 6 个月，生产后最长的免费质保期为 18 个月。维修零部件的免费质保期不得超过修理前的免费质保期。

［ 免费质保范围 ］

(1) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。

(2) 以下情况下，即使在免费质保期内，也要收取维修费用。

1. 因不适当存储或搬运、用户粗心或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。
2. 因用户未经批准对产品进行改造而导致的故障等。
3. 对于装有三菱产品的用户设备，如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。
4. 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。
5. 因火灾或异常电压等外部因素以及因地震、雷电、大风和水灾等不可抗力而导致的故障。
6. 根据从三菱出货时的科技标准还无法预知的原因而导致的故障。
7. 任何非三菱或用户责任而导致的故障。

2. 产品停产后的有偿维修期限

(1) 三菱在本产品停产后的 7 年内受理该产品的有偿维修。
停产的消息将以三菱技术公告等方式予以通告。

(2) 产品停产后，将不再提供产品（包括维修零件）。

3. 海外服务

在海外，维修由三菱在当地的海外 FA 中心受理。注意各个 FA 中心的维修条件可能会不同。

4. 意外损失和间接损失不在质保责任范围内

无论是否在免费质保期内，对于任何非三菱责任的原因而导致的损失、机会损失、因三菱产品故障而引起的用户利润损失、无论能否预测的特殊损失和间接损失、事故赔偿、除三菱以外产品的损失赔偿、用户更换设备、现场机械设备的再调试、运行测试及其它作业等，三菱将不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变，恕不另行通知。

6. 产品应用

(1) 在使用三菱 MELSEC 通用可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效保险功能。

(2) 三菱通用可编程控制器是以一般工业用途等为对象设计和制造的。因此，可编程控制器的应用不包括那些会影响公共利益的应用，如核电厂和其它由独立供电公司经营的电厂以及需要特殊质量保证的应用如铁路公司或用于公用设施目的的应用。

另外，可编程控制器的应用不包括航空、医疗应用、焚化和燃烧设备、载人设备、娱乐及休闲设施、安全装置等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时的应用。

然而，对于这些应用，假如用户咨询当地三菱代表机构，提供有特殊要求方案的大纲并提供满足特殊环境的所有细节及用户自主要求，则可以进行一些应用。

Microsoft、Windows、Windows NT 是美国 Microsoft Corporation 公司在美国及其它国家的注册商标。
Adobe、Acrobat 是 Adobe Systems Incorporated 公司的注册商标。
Pentium、Celeron 是 Intel Corporation 公司在美国及其它国家的商标及注册商标。
Ethernet 是美国 Xerox.co.ltd 公司的注册商标。
PC-9800、PC98-NX 是日本电气公司的注册商标。
其它本文中的公司名、商品名是各公司的商标或注册商标。

QCPU(Q模式)/QnACPU

编程手册(公共指令篇) 1/2

技术服务热线:

800-828-9910

服务时间: **9:00~12:00**

13:00~17:00(节假日除外)

三菱电机自动化(上海)有限公司

地址：上海市黄浦区新昌路80号智富广场4楼
邮编：200003
电话：021-61200808　传真：021-61212444
网址：www.mitsubishielectric-automation.cn

书号	SH(NA)-080450CHN-B(0711)STC
印号	STC-Q/QnaCPU-CI-PM(0711)

内容如有更改
恕不另行通知

QCPU(Q模式)/QnACPU

MITSUBISHI

编程手册

(公共指令篇)2/2



三菱可编程控制器

MELSEC-Q

●安全警告●

（在使用产品之前必须先阅读此警告）

在使用产品之前，请先阅读此手册和此手册中介绍的相关手册。请对安全问题给予足够重视以便能正确操作此产品。

请将此手册保存到一个安全的地方，以便在需要时可以方便地得到。请送给最终用户本手册的副本。

修订本

此手册编号在封底的左下角给出。

出版日期	手册编号	版本
2003 年 09 月	SH(NA)-080450CHN-A	初版
2007 年 11 月	SH(NA)-080450CHN-B	全面改版

英文手册版本： SH-080039-M

此手册无任何工业产权，无如何其它类型产权，也无任何专利。三菱电机公司不对由于使用本手册中的说明而造成的任何可能包含的工业产权问题负责。

© 2003 三菱电机

前言

感谢使用三菱电机 MELSEC-Q 系列 (Q 模式) 和 MELSEC-QnA 系列可编程控制器。
在使用这些产品之前，请仔细阅读本手册，使你能对所使用的 Q 系列 (Q 模式)/QnA 系列可编程控制器的功能和性能足够熟悉，以便能保证正确使用该类产品。
应该送给最终用户此手册的一个副本。

目录

公共指令 1/2

安全警告 ..... A - 1
修订本 ..... A - 2
前言 ..... A - 3
目录 ..... A - 3
关于手册 ..... A - 14

1. 概述 1 - 1 到 1 - 6
1.1 相关编程手册 1 - 2
1.2 本手册中使用的总称及略称 1 - 5

2. 概述 2 - 1 到 2 - 56
2.1 指令类型 2 - 2
2.2 如何阅读指令表 2 - 4
2.3 顺序指令 2 - 6
2.3.1 触点指令 ..... 2 - 6
2.3.2 连接指令 ..... 2 - 7
2.3.3 输出指令 ..... 2 - 8
2.3.4 移位指令 ..... 2 - 8
2.3.5 主控制指令 ..... 2 - 9
2.3.6 终止指令 ..... 2 - 9
2.3.7 其它指令 ..... 2 - 9
2.4 基本指令 2 - 10
2.4.1 比较操作指令 ..... 2 - 10
2.4.2 算术运算指令 ..... 2 - 16
2.4.3 数据转换指令 ..... 2 - 21
2.4.4 数据传送指令 ..... 2 - 23
2.4.5 程序分支指令 ..... 2 - 25
2.4.6 程序执行控制指令 ..... 2 - 25
2.4.7 I/O 刷新指令 ..... 2 - 25
2.4.8 其它使用方便的指令 ..... 2 - 26
2.5 应用指令 2 - 27
2.5.1 逻辑操作指令 ..... 2 - 27
2.5.2 循环指令 ..... 2 - 30
2.5.3 位指令 ..... 2 - 31
2.5.4 处理指令 ..... 2 - 31
2.5.5 据处理指令 ..... 2 - 32
2.5.6 结构体创建指令 ..... 2 - 35
2.5.7 数据表操作指令 ..... 2 - 37
2.5.8 缓冲存储区访问指令 ..... 2 - 38
2.5.9 显示指令 ..... 2 - 38

2.5.10 调试和故障诊断指令 ..... 2 - 39
2.5.11 字符串处理指令 ..... 2 - 40
2.5.12 特殊功能指令 ..... 2 - 43
2.5.13 数据控制指令 ..... 2 - 46
2.5.14 转换指令 ..... 2 - 47
2.5.15 时钟指令 ..... 2 - 48
2.5.16 外围软元件指令 ..... 2 - 49
2.5.17 程序控制指令 ..... 2 - 49
2.5.18 其他指令 ..... 2 - 50
2.5.19 用于数据链接的指令 ..... 2 - 51
2.5.20 QCPU 指令 ..... 2 - 53
2.5.21 冗余系统指令（用于 Q4ARCPU) ..... 2 - 55
2.5.22 冗余系统指令（用于冗余 CPU) ..... 2 - 55

3. 指令组态 3 - 1 到 3 - 40
3.1 指令结构 3 - 2
3.2 数据的指定方法 3 - 3
3.2.1 使用位数据 ..... 3 - 3
3.2.2 使用字（16 位）数据 ..... 3 - 4
3.2.3 使用双字数据（32 位）..... 3 - 6
3.2.4 使用实数数据 ..... 3 - 8
3.2.5 使用字符串数据 ..... 3 - 11
3.3 变址修饰 3 - 12
3.4 间接指定 3 - 18
3.5 缩短指令处理时间 3 - 20
3.5.1 子集处理 ..... 3 - 20
3.5.2 使用通用运算寄存器（Z）的运算处理（只对于通用型 QCPU) ..... 3 - 21
3.6 编程注意事项（运算错误） 3 - 22
3.7 指令执行条件 3 - 27
3.8 计算步数 3 - 28
3.9 当 OUT、SET/RST 或 PLS/PLF 指令使用相同的软元件时的操作 3 - 32
3.10 使用文件寄存器的注意事项 3 - 37

4. 如何阅读指令 4 - 1 到 4 - 4

5. 顺序指令 5 - 1 到 5 - 56
5.1 触点指令 5 - 2
5.1.1 运行开始，串行连结，并行连接
（LD、LDI、AND、ANI、OR、ORI) ..... 5 - 2
5.1.2 脉冲运行开始，脉冲串行连接，脉冲并行连接（LDP、LDF、ANDP、ANDF、ORP、ORF) ..... 5 - 5
5.2 连接指令 5 - 7
5.2.1 梯形图块串行连接和并行连接
（ANB、ORB) ..... 5 - 7
5.2.2 运行结果推进，读取，弹出（MPS、MRD、MPP) ..... 5 - 9
5.2.3 运行结果反转（INV) ..... 5 - 12
5.2.4 运算结果脉冲变换（MEP、MEF) ..... 5 - 13
5.2.5 变址继电器运算结果的脉冲变换（EGP、EGF) ..... 5 - 14

5.3 输出指令 5 - 16
5.3.1 输出指令（不包括定时器，计数器，和报警器）(OUT) ..... 5 - 16
5.3.2 定时器 (OUT T、OUTH T) ..... 5 - 18
5.3.3 计数器 (OUT C) ..... 5 - 22
5.3.4 报警器输出 (OUT F) ..... 5 - 24
5.3.5 设定软元件（报警器除外） (SET) ..... 5 - 26
5.3.6 复位软元件（除了报警器）(RST) ..... 5 - 28
5.3.7 设定和复位报警器 (SET F、RST F) ..... 5 - 33
5.3.8 上升沿和下降沿输出 (PLS、PLF) ..... 5 - 36
5.3.9 位软元件输出取反 (FF) ..... 5 - 38
5.3.10 直接输出的脉冲变换 (DELTA(P)) ..... 5 - 40
5.4 移动指令 5 - 40
5.4.1 位软元件移动 (SFT(P)) ..... 5 - 40
5.5 主控指令 5 - 43
5.5.1 设置和复位主控 (MC、MCR) ..... 5 - 43
5.6 结束指令 5 - 47
5.6.1 结束主程序 (FEND) ..... 5 - 47
5.6.2 结束顺控程序 (END) ..... 5 - 49
5.7 其他指令 5 - 51
5.7.1 顺控程序暂停 (STOP) ..... 5 - 51
5.7.2 无处理 (NOP、NOPLF、PAGE n) ..... 5 - 53

6. 基本指令 6 - 1 到 6 - 162
6.1 比较运算指令 6 - 2
6.1.1 BIN 16 位数据比较 (=、<>、>、<=、<、>=) ..... 6 - 2
6.1.2 BIN 32 位数据比较 (D=、D<>、D>、D<=、D<、D>=) ..... 6 - 4
6.1.3 浮动小数点数据比较 (E=、E<>、E>、E<=、E<、E>=) ..... 6 - 6
6.1.4 浮动小数点数据比较（双精度）
(ED=、ED<>、ED>、ED<=、ED<、ED>=) ..... 6 - 8
6.1.5 字符串数据比较 ($=、$<>、$>、$<=、$<、$>=) ..... 6 - 11
6.1.6 BIN 块数据比较 (BKCMP □ 、BKCMP □ P) ..... 6 - 15
6.2 算术运算指令 6 - 19
6.2.1 BIN 16 位加法和减法运算 (+(P)、-(P)) ..... 6 - 19
6.2.2 BIN 32 位加法和减法运算 (D+(P)、D-(P)) ..... 6 - 23
6.2.3 BIN 16 位乘法和除法运算 (*(P)、/(P)) ..... 6 - 27
6.2.4 BIN 32 位乘法和除法运算 (D*(P)、D/(P)) ..... 6 - 29
6.2.5 BCD 4 位加法和减法运算 (B+(P)、B-(P)) ..... 6 - 31
6.2.6 BCD 8 位加法和减法运算 (DB+(P)、DB-(P)) ..... 6 - 35
6.2.7 BCD 4 位乘法和除法运算 (B*(P)、B/(P)) ..... 6 - 39
6.2.8 BCD 8 位乘法和除法运算 (DB*(P)、DB/(P)) ..... 6 - 41
6.2.9 浮点数据的加法和减法运算 (E+(P)、E-(P)) ..... 6 - 43
6.2.10 浮点数据的加法和减法运算（双精度）(ED+(P)、ED-(P)) ..... 6 - 47
6.2.11 浮点数据的乘法和除法运算 (E*(P)、E/(P)) ..... 6 - 51
6.2.12 浮点数据的乘法和除法运算（双精度）(ED*(P)、ED/(P)) ..... 6 - 53
6.2.13 块加法和减法运算 (BK+(P)、BK-(P)) ..... 6 - 56
6.2.14 链接字符串 ($+(P)) ..... 6 - 59
6.2.15 16 位 BIN 数据的递增和递减运算 (INC(P)、DEC(P)) ..... 6 - 63
6.2.16 32 位 BIN 数据的递增和递减运算 (DINC(P)、DDEC(P)) ..... 6 - 65
6.3 数据转换指令 6 - 67
6.3.1 从 BIN 数据到 4 位和 8 位 BCD 数据的转换

(BCD(P)、DBCD(P)) ..... 6 - 67
6.3.2 从 BCD 4 位和 8 位数据到 BIN 数据的转换 (BIN(P)、DBIN(P)) ..... 6 - 69
6.3.3 从 BIN 16 位和 32 位数据到浮点数据的转换
(FLT(P)、DFLT(P)) ..... 6 - 72
6.3.4 从 BIN16 和 32 位数据到浮点数据的转换 (双精度)
(FLTD(P)、DFLTD(P)) ..... 6 - 75
6.3.5 从浮点数据到 BIN 16 位和 32 位数据的转换
(INT(P)、DINT(P)) ..... 6 - 77
6.3.6 从浮点数据到 BIN16 位 /32 位数据的转换 (双精度)
(INTD(P)、DINTD(P)) ..... 6 - 80
6.3.7 从 BIN 16 位数据到 BIN 32 位数据的转换 (DBL(P)) ..... 6 - 82
6.3.8 从 BIN 32 位到 BIN 16 位数据的转换 (WORD(P)) ..... 6 - 83
6.3.9 从 BIN 16 位和 32 位数据到格雷码的转换
(GRY(P)、DGRY(P)) ..... 6 - 84
6.3.10 从格雷码到 BIN 16 位和 32 位数据的转换
(GBIN(P)、DGBIN(P)) ..... 6 - 86
6.3.11 BIN 16 位和 32 位数据的 2 进制补码 (符号取反)
(NEG(P)、DNEG(P)) ..... 6 - 88
6.3.12 浮点数据的符号取反 (单精度)(ENEG(P)) ..... 6 - 90
6.3.13 浮点数据的符号取反 (双精度)(EDNEG(P)) ..... 6 - 91
6.3.14 从块 BIN 16 位数据到 BCD 4 位数据的转换 (BKBCD(P)) ..... 6 - 92
6.3.15 从块 BCD 4 位数据到块 BIN 16 位数据的转换 (BKBIN(P)) ..... 6 - 94
6.3.16 单精度→双精度转换 (ECON(P)) ..... 6 - 96
6.3.17 双精度→单精度转换 (EDCON(P)) ..... 6 - 98
6.4 数据传送指令 6 - 100
6.4.1 16 位和 32 位数据传送 (MOV(P)、DMOV(P)) ..... 6 - 100
6.4.2 浮点数据传送 (EMOV(P)) ..... 6 - 102
6.4.3 浮点数据传送 (双精度) (EDMOV(P)) ..... 6 - 104
6.4.4 字符串传送 ($MOV(P)) ..... 6 - 106
6.4.5 16 位和 32 位数据取反传送 (CML(P)、DCML(P)) ..... 6 - 108
6.4.6 块 16 位数据传送 (BMOV(P)) ..... 6 - 111
6.4.7 相同的 16 位数据块传送 (FMOV(P)) ..... 6 - 114
6.4.8 16 位和 32 位数据交换 (XCH(P)、DXCH(P)) ..... 6 - 116
6.4.9 块 16 位数据交换 (BXCH(P)) ..... 6 - 118
6.4.10 高字节和低字节交换 (SWAP(P)) ..... 6 - 120
6.5 程序分支指令 6 - 121
6.5.1 指针分支指令 (CJ、SCJ、JMP) ..... 6 - 121
6.5.2 跳转到 END(GOEND) ..... 6 - 124
6.6 程序执行控制指令 6 - 125
6.6.1 中断禁止 / 允许指令，中断程序屏蔽 (DI、EI、IMASK) ..... 6 - 125
6.6.2 从中断程序中恢复 (IRET) ..... 6 - 134
6.7 I/O 刷新指令 6 - 136
6.7.1 I/O 刷新 (RFS(P)) ..... 6 - 136
6.8 其它方便的指令 6 - 138
6.8.1 单相输入加法或减法计数器 (UDCNT1) ..... 6 - 138
6.8.2 两相输入加法或减法计数器 (UDCNT2) ..... 6 - 141
6.8.3 教学计时器 (TTMR) ..... 6 - 144
6.8.4 特殊功能定时器 (STMR) ..... 6 - 146
6.8.5 旋转台就近控制 (ROTC) ..... 6 - 149
6.8.6 斜坡信号 (RAMP) ..... 6 - 152

6.8.7 脉冲密度测量 (SPD) ..... 6 - 155
6.8.8 固定周期脉冲输出 (PLSY) ..... 6 - 157
6.8.9 脉冲宽度调制 (PWM) ..... 6 - 159
6.8.10 矩阵输入 (MTR) ..... 6 - 161

7. 应用指令 7 - 1 到 7 - 374

7.1 逻辑运算指令 7 - 2
7.1.1 16 位和 32 位数据的逻辑乘 (WAND(P)、DAND(P)) ..... 7 - 3
7.1.2 块逻辑乘 (BKAND(P)) ..... 7 - 9
7.1.3 16 位和 32 位数据的逻辑和 (WOR(P)、DOR(P)) ..... 7 - 11
7.1.4 块逻辑和运算 (BKOR(P)) ..... 7 - 17
7.1.5 16 位和 32 位数据异或运算 (WXOR(P)、DXOR(P)) ..... 7 - 19
7.1.6 块异或运算 (BKXOR(P)) ..... 7 - 25
7.1.7 16 位和 32 位异或非运算 (WXNR(P)、DXNR(P)) ..... 7 - 27
7.1.8 块异或非运算 (BKXNR(P)) ..... 7 - 33
7.2 循环指令 7 - 35
7.2.1 16 位数据的右循环 (ROR(P)、RCR(P)) ..... 7 - 35
7.2.2 16 位数据左循环 (ROL(P)、RCL(P)) ..... 7 - 38
7.2.3 32 位数据的右循环 (DROR(P)、DRCR(P)) ..... 7 - 41
7.2.4 32 位数据左循环 (DROL(P)、DRCL(P)) ..... 7 - 44
7.3 移位指令 7 - 46
7.3.1 16 位数据的 n 位左移或右移 (SFR(P)、SFL(P)) ..... 7 - 46
7.3.2 n 位数据的 1 位左移或右移 (BSFR(P)、BSFL(P)) ..... 7 - 49
7.3.3 n 字数据的 1 字左移或右移 (DSFR(P)、DSFL(P)) ..... 7 - 51
7.4 位处理指令 7 - 53
7.4.1 字软元件的位设定和复位 (BSET(P)、BRST(P)) ..... 7 - 53
7.4.2 位测试 (TEST(P)、DTEST(P)) ..... 7 - 55
7.4.3 位软元件的成批复位 (BKRST(P)) ..... 7 - 58
7.5 数据处理指令 7 - 60
7.5.1 16 位和 32 位数据搜索 (SER(P)、DSER(P)) ..... 7 - 60
7.5.2 16 位和 32 位数据的位数据检查 (SUM(P)、DSUM(P)) ..... 7 - 63
7.5.3 8 位到 256 位的解码 (DECO(P)) ..... 7 - 65
7.5.4 将 256 位编码为 8 位 (ENCO(P)) ..... 7 - 67
7.5.5 7 段解码 (SEG(P)) ..... 7 - 69
7.5.6 16 位数据的 4 位分组 (DIS(P)) ..... 7 - 71
7.5.7 16 位数据的 4 位链接 (UNI(P)) ..... 7 - 73
7.5.8 任意数据的分离或链接 (NDIS(P)、NUNI(P)) ..... 7 - 75
7.5.9 以字节为单位的数据分离和链接 (WTOB(P)、BTOW(P)) ..... 7 - 79
7.5.10 16 位和 32 位数据的最大值查找 (MAX(P)、DMAX(P)) ..... 7 - 83
7.5.11 查找 16 位和 32 位数据的最小值 (MIN(P)、DMIN(P)) ..... 7 - 86
7.5.12 BIN16 位和 32 位数据的排序运算 (SORT、DSORT) ..... 7 - 89
7.5.13 16 位数据的总和计算 (WSUM(P)) ..... 7 - 93
7.5.14 32 位数据总和计算 (DWSUM(P)) ..... 7 - 95
7.6 结构化程序指令 7 - 97
7.6.1 FOR 到 NEXT 指令循环 (FOR、NEXT) ..... 7 - 97
7.6.2 FOR 到 NEXT 指令循环的强制结束 (BREAK(P)) ..... 7 - 100
7.6.3 子程序调用 (CALL(P)) ..... 7 - 102
7.6.4 从子程序返回 (RET) ..... 7 - 107
7.6.5 子程序输出 OFF 调用 (FCALL(P)) ..... 7 - 108
7.6.6 程序文件之间的子程序调用 (ECALL(P)) ..... 7 - 112

7.6.7 程序文件之间的子程序输出 OFF 调用 (EFCALL(P)) ..... 7 - 117
7.6.8 子程序调用 (XCALL) ..... 7 - 121
7.6.9 刷新指令 (COM) ..... 7 - 126
7.6.10 整个梯形图的变址修饰 (IX、IXEND) ..... 7 - 129
7.6.11 整个梯形图的变址修饰中修饰值的指定 (IXDEV、IXSET) ..... 7 - 133
7.7 数据表格操作指令 7 - 136
7.7.1 将数据写入数据表格 (FIFW(P)) ..... 7 - 136
7.7.2 从表格中读取最旧的数据 (FIFR(P)) ..... 7 - 138
7.7.3 从数据表格中读取最新数据 (FPOP(P)) ..... 7 - 140
7.7.4 从数据表格中删除数据和在数据表格中插入数据 (FDEL(P)、FINS(P)) ..... 7 - 142
7.8 缓冲存储区访问指令 7 - 145
7.8.1 从智能功能模块中读取 1 字 /2 字数据 (FROM(P)、DFRO(P)) ..... 7 - 145
7.8.2 将 1 字 /2 字数据写入智能功能模块 (TO(P)、DTO(P)) ..... 7 - 148
7.9 显示指令 7 - 151
7.9.1 打印 ASCII 码指令 (PR) ..... 7 - 151
7.9.2 打印注释指令 (PRC) ..... 7 - 154
7.9.3 ASCII 码 LED 显示指令 (LED) ..... 7 - 159
7.9.4 注释的 LED 显示指令 (LEDC) ..... 7 - 161
7.9.5 错误显示和报警器复位指令 (LEDR) ..... 7 - 163
7.10 调试和故障诊断指令 7 - 166
7.10.1 特殊格式故障检测 (CHKST、CHK) ..... 7 - 166
7.10.2 改变 CHK 指令的检测格式 (CHKCIR、CHKEND) ..... 7 - 170
7.10.3 设置和复位状态锁存 (SLT、SLTR) ..... 7 - 178
7.10.4 设置和复位采样跟踪 (STRA、STRAR) ..... 7 - 180
7.10.5 程序跟踪的执行、设置和复位 (PTRAEXE(P)、PTRA、PTRAR) ..... 7 - 182
7.11 字符串处理指令 7 - 184
7.11.1 从 BIN16 位或 32 位到十进制 ASCII 码的转换 (BINDA(P)、DBINDA(P)) ..... 7 - 184
7.11.2 BIN16 位或 32 位数据到十六进制 ASCII 码的转换 (BINHA(P)、DBINHA(P)) ..... 7 - 187
7.11.3 从 BCD4 位和 8 位数据到十进制 ASCII 码的转换 (BCDDA(P)、DBCDDA(P)) ..... 7 - 190
7.11.4 从十进制 ASCII 码到 BIN16 位和 32 位数据的转换 (DABIN(P)、DDABIN(P)) ..... 7 - 193
7.11.5 从十六进制 ASCII 到 BIN16 位和 32 位数据的转换 (HABIN(P)、DHABIN(P)) ..... 7 - 196
7.11.6 从十进制 ASCII 码到 BCD4 位或 8 位数据的转换 (DABCD(P)、DDABCD(P)) ..... 7 - 199
7.11.7 读取软元件注释数据 (COMRD(P)) ..... 7 - 202
7.11.8 字符串长度检测 (LEN(P)) ..... 7 - 207
7.11.9 从 BIN16 位或 32 位到字符串的转换 (STR(P)、DSTR(P)) ..... 7 - 209
7.11.10 从字符串到 BIN16 位或 32 位数据的转换 (VAL(P)、DVAL(P)) ..... 7 - 215
7.11.11 从浮点型到字符串数据的转换 (ESTR(P)) ..... 7 - 220
7.11.12 从字符串到浮点数的转换 (EVAL(P)) ..... 7 - 226
7.11.13 从十六进制 BIN 到 ASCII 码的转换 (ASC(P)) ..... 7 - 230
7.11.14 从 ASCII 码到十六进制 BIN 的转换 (HEX(P)) ..... 7 - 232
7.11.15 从字符串的右边或左边提取数据 (RIGHT(P)、LEFT(P)) ..... 7 - 234
7.11.16 字符串的任意选择和替换 (MIDR(P)、MIDW(P)) ..... 7 - 237
7.11.17 字符串查找 (INSTR(P)) ..... 7 - 241
7.11.18 浮点数到 BCD 的分解 (EMOD(P)) ..... 7 - 243

7.11.19 从 BCD 格式数据到浮点数 (EREXP(P)) ..... 7 - 246
7.12 特殊函数指令 ..... 7 - 248
7.12.1 浮点数的 SIN 运算 (单精度)(SIN(P)) ..... 7 - 248
7.12.2 浮点数的 SIN 运算 (双精度)(SIND(P)) ..... 7 - 250
7.12.3 浮点数的 COS 运算 (单精度)(COS(P)) ..... 7 - 252
7.12.4 浮点数的 COS 运算 (双精度) (COSD(P)) ..... 7 - 254
7.12.5 浮点数的 TAN 运算 (单精度)(TAN(P)) ..... 7 - 256
7.12.6 浮点数的 TAN 运算 (双精度)(TAND(P)) ..... 7 - 258
7.12.7 浮点数的 SIN^{-1} 运算 (单精度)(ASIN(P)) ..... 7 - 260
7.12.8 浮点数的 SIN^{-1} 运算 (双精度)(ASIND(P)) ..... 7 - 263
7.12.9 浮点数的 COS^{-1} 运算 (单精度)(ACOS(P)) ..... 7 - 265
7.12.10 浮点数的 COS^{-1} 运算 (双精度)(ACOSD(P)) ..... 7 - 267
7.12.11 浮点数 TAN^{-1} 运算 (单精度)(ATAN(P)) ..... 7 - 269
7.12.12 浮点数 TAN^{-1} 运算 (双精度)(ATAND(P)) ..... 7 - 271
7.12.13 从浮点数角度到弧度的转换 (单精度)(RAD(P)) ..... 7 - 273
7.12.14 从浮点数角度到弧度的转换 (双精度)(RADD(P)) ..... 7 - 275
7.12.15 从浮点数弧度到角度的转换 (单精度)(DEG(P)) ..... 7 - 277
7.12.16 从浮点数弧度到角度的转换 (双精度)(DEGD(P)) ..... 7 - 279
7.12.17 浮点数的平方根运算 (单精度)(SQR(P)) ..... 7 - 281
7.12.18 浮点数的平方根运算 (双精度)(SQRD(P)) ..... 7 - 283
7.12.19 浮点数的指数运算 (单精度)(EXP(P)) ..... 7 - 285
7.12.20 浮点数的指数运算 (双精度)(EXPD(P)) ..... 7 - 288
7.12.21 浮点数的自然对数运算 (单精度)(LOG(P)) ..... 7 - 290
7.12.22 浮点数的自然对数运算 (双精度)(LOGD(P)) ..... 7 - 292
7.12.23 随机数的产生和系列更新 (RND(P)、SRND(P)) ..... 7 - 294
7.12.24 BCD4 位和 8 位平方根 (BSQR(P)、BDSQR(P)) ..... 7 - 296
7.12.25 BCD 型 SIN 运算 (BSIN(P)) ..... 7 - 299
7.12.26 BCD 型 COS 运算 (BCOS(P)) ..... 7 - 301
7.12.27 BCD 型 TAN 运算 (BTAN(P)) ..... 7 - 303
7.12.28 BCD 型 SIN^{-1} 运算 (BASIN(P)) ..... 7 - 305
7.12.29 BCD 型 COS^{-1} 运算 (BACOS(P)) ..... 7 - 307
7.12.30 BCD 型 TAN^{-1} 运算 (BATAN(P)) ..... 7 - 309
7.13 数据控制指令 ..... 7 - 311
7.13.1 BIN 16 位和 BIN 32 位数据的高低限控制 (LIMIT(P)、DLIMIT(P)) ..... 7 - 311
7.13.2 BIN 16 位和 32 位死区控制 (BAND(P)、DBAND(P)) ..... 7 - 314
7.13.3 BIN 16 位和 BIN 32 位数据的区域控制 (ZONE(P)、DZONE(P)) ..... 7 - 317
7.14 文件寄存器切换指令 ..... 7 - 320
7.14.1 切换文件寄存器号 (RSET(P)) ..... 7 - 320
7.14.2 文件寄存器使用的设置文件 (QDRSET(P)) ..... 7 - 322
7.14.3 注释的文件设置 (QCDSET(P)) ..... 7 - 325
7.15 时钟指令 ..... 7 - 327
7.15.1 读时钟数据 (DATERD(P)) ..... 7 - 327
7.15.2 写时钟数据 (DATEWR(P)) ..... 7 - 331
7.15.3 时钟数据加法运算 (DATE+(P)) ..... 7 - 335
7.15.4 时钟数据减法运算 (DATE-(P)) ..... 7 - 337
7.15.5 时间数据转换 (从小时 / 分钟 / 秒到秒)(SECOND(P)) ..... 7 - 339
7.15.6 时间数据转换 (从秒到小时 / 分钟 / 秒)(HOUR(P)) ..... 7 - 341
7.16 外围设备指令 ..... 7 - 343

7.16.1 将信息显示到外围设备 (MSG) ..... 7 - 343
7.16.2 外围设备的键盘输入 (PKEY) ..... 7 - 345
7.17 程序控制指令 7 - 347
7.17.1 程序待机指令 (PSTOP(P)) ..... 7 - 349
7.17.2 程序输出 OFF 待机指令 (POFF(P)) ..... 7 - 350
7.17.3 程序扫描执行登记指令 (PSCAN(P)) ..... 7 - 352
7.17.4 程序低速执行登记指令 (PLOW(P)) ..... 7 - 354
7.17.5 程序执行状态检查指令 (PCHK) ..... 7 - 356
7.18 其它指令 7 - 358
7.18.1 复位看门狗定时器 (WDT(P)) ..... 7 - 358
7.18.2 定时脉冲发生 (DUTY) ..... 7 - 360
7.18.3 时间检查指令 (TIMCHK) ..... 7 - 362
7.18.4 从文件寄存器中的直接 1 字节读取 (ZRRDB(P)) ..... 7 - 363
7.18.5 写入 1 字节到文件寄存器 (ZRWRB(P)) ..... 7 - 365
7.18.6 间接地址读取操作 (ADRSET(P)) ..... 7 - 367
7.18.7 键盘的数字键输入 (KEY) ..... 7 - 368
7.18.8 批量保存或变址寄存器的恢复 (ZPUSH(P)、ZPOP(P)) ..... 7 - 372
7.18.9 对 E^2PROM 文件寄存器的批写入操作 (EROMWR(P)) ..... 7 - 374

公共指令 2/2

8. 数据链接指令 8 - 1 到 8 - 136
8.1 概要 8 - 2
8.2 网络刷新指令 8 - 10
8.2.1 网络刷新 (S(P)/J(P)/G(P).ZCOM) ..... 8 - 10
8.3 QnA 链接专用指令 8 - 18
8.3.1 从其它站中读取字软元件数据 (JP/GP.READ) ..... 8 - 18
8.3.2 从其它站读取字软元件数据 (JP/GP.SREAD) ..... 8 - 29
8.3.3 写入软元件数据到其它站 (JP/GP.WRITE) ..... 8 - 40
8.3.4 将软元件数据写入到其它站 (JP/GP.SWRITE) ..... 8 - 51
8.3.5 传送数据到其它站 (JP/GP.SEND) ..... 8 - 62
8.3.6 从其它站接收数据 (JP/GP.RECV) ..... 8 - 72
8.3.7 来自其它站的瞬时请求 (时钟数据的读 / 写、
远程 RUN/STOP)(JP/GP.REQ) ..... 8 - 81
8.3.8 从远程 I/O 站的特殊功能模块中读数据 (JP/GP.ZNFR) ..... 8 - 95
8.3.9 往远程 I/O 站的特殊功能模块中写数据 (JP/GP.ZNTO) ..... 8 - 100
8.4 A 系列兼容链接指令 8 - 105
8.4.1 从其它站读软元件数据 (MELSECNET/10、Ethernet) (J(P).ZNRD) ..... 8 - 105
8.4.2 从本地站 (MELSECNET) 读软元件数据 (J(P).ZNRD) ..... 8 - 109
8.4.3 将软元件数据写入其它站 (MELSECNET/10、以太网) (J(P).ZNWR) ..... 8 - 113
8.4.4 将数据写入本地站软元件中 (MELSECNET)(J(P).ZNWR) ..... 8 - 117
8.4.5 从远程 I/O 站的特殊功能模块读数据 (MELSECNET)(G(P).RFRP) ..... 8 - 121
8.4.6 将数据写入远程 I/O 站特殊功能模块 (MELSECNET)(G(P).RTOP) ..... 8 - 125
8.5 读 / 写路由信息 8 - 129
8.5.1 读路由信息 (S(P)/Z(P).RTREAD) ..... 8 - 129
8.5.2 登录路由信息 (S(P)/Z(P).RTWRITE) ..... 8 - 133
9. QCPU 指令 9 - 1 到 9 - 74
9.1 读取模块信息 (UNIRD(P)) 9 - 2

9.2 追踪置位 / 复位 (TRACE、TRACER) 9 - 6
9.3 写数据到指定的文件 (SP.FWRITE) 9 - 8
9.4 从指定文件中读取数据 (SP.FREAD) 9 - 19
9.5 向标准 ROM 中写入数据 (SP.DEVST) 9 - 31
9.6 从标准 ROM 中读取数据 (S(P).DEVLD) 9 - 33
9.7 从内存卡中装载程序 (PLOADP) 9 - 35
9.8 从程序内存中卸载程序 (PUNLOADP) 9 - 38
9.9 装载 + 卸载 (PSWAPP) 9 - 40
9.10 文件寄存器的高速块传送 (RBMOV(P)) 9 - 43
9.11 写入自站 CPU 共享内存 9 - 47
9.11.1 写入到自站 CPU 共享内存 (S(P).TO) 9 - 49
9.11.2 写入到自站 CPU 共享内存 (TO(P)、DTO(P)) 9 - 52
9.12 从其他站的 CPU 共享内存中读取数据 9 - 56
9.12.1 从其他站的共享内存读取数据 (FROM(P)、DFRO(P)) 9 - 57
9.13 选择性刷新指令 (COM) 9 - 63
9.14 扩展时钟指令 9 - 66
9.14.1 读取扩展时钟数据 (S(P).DATERD) 9 - 66
9.14.2 扩展时钟数据的加法运算 (S(P).DATE+) 9 - 69
9.14.3 扩展时钟数据的减法运算 (S(P).DATE-) 9 - 72

10. 冗余系统指令 (用于 Q4ARCPU) 10 - 1 到 10 - 14
10.1 CPU 启动过程中的操作模式设定指令 (S.STMODE) 10 - 2
10.2 CPU 切换时间操作模式设定指令 (S.CGMODE) 10 - 4
10.3 数据追踪指令 (S.TRUCK) 10 - 6
10.4 缓冲内存批刷新指令 (S.SPREF) 10 - 10

11. 冗余系统指令 (用于冗用系统 CPU) 11 - 1 到 11 - 4
11.1 系统切换指令 (SP.CONTSW) 11 - 2

12. 出错代码 12 - 1 到 12 - 126
12.1 出错代码表 12 - 2
12.1.1 出错代码 12 - 3
12.1.2 出错代码的读取方法 12 - 3
12.1.3 出错代码一览表 (1000 ～ 1999) 12 - 4
12.1.4 出错代码表 (2000 ～ 2999) 12 - 26
12.1.5 出错代码表 (3000 ～ 3999) 12 - 56
12.1.6 出错代码表 (4000 ～ 4999) 12 - 78
12.1.7 出错代码表 (5000 ～ 5999) 12 - 96
12.1.8 出错代码表 (6000 ～ 6999) 12 - 100
12.1.9 出错代码表 (7000 ～ 7999) 12 - 120
12.2 出错的解除 12 - 126

附录 附录 - 1 到附录 - 166
附录 1 操作处理时间 附录 - 2
附录 1.1 定义 附录 - 2

附录 1.2 基本型 QCPU 的操作处理时间 附录 - 4
附录 1.3 高性能型 QCPU/ 过程 CPU/ 冗余 CPU/QnACPU 的运算处理时间 附录 - 22
附录 1.4 通用型 QCPU 的运算处理时间 附录 - 50
附录 1.4.1 子集指令处理时间 附录 - 50
附录 1.4.1 子集指令以外的指令处理时间 附录 - 57
附录 2 CPU 之间的性能比较 附录 - 72
附录 2.1 Q/QnACPU 和 AnNCPU、AnACPU 以及 AnUCPU 之间的比较 附录 - 72
附录 2.1.1 可用的软元件 附录 - 72
附录 2.1.2 I/O 控制模式 附录 - 73
附录 2.1.3 可以被指令使用的数据 附录 - 73
附录 2.1.4 定时器比较 附录 - 74
附录 2.1.5 计数器比较 附录 - 75
附录 2.1.6 显示指令比较 附录 - 75
附录 2.1.7 改变定义格式指令（不包括用于 AnACPU 和 AnUCPU 的专用指令） 附录 - 76
附录 2.1.8 AnACPU 和 AnUCPU 专用指令 附录 - 77
附录 2.1.9 只能在一般模式中编程的指令 附录 - 77
附录 3 特殊继电器列表 附录 - 78
附录 4 特殊寄存器列表 附录 - 106
附录 5 应用程序实例 附录 - 165
附录 5.1执行 X^n、$\sqrt[n]{X}$ 操作的程序概念 附录 - 165
索引 索引 - 1 到索引 -10

关于手册

与本产品相关的手册如下所述。

请根据需要参考与借助此表。

相关手册

手册名称	手册号
QCPU 用户手册（硬件设计 / 维护点检篇） 描述了 CPU 模块、电源模块、基板、扩展电缆及存储卡等的规格。 （另售）	SH-080501CHN
QCPU 用户手册（功能解说 / 程序基础篇） 描述了创建程序所需的功能、编程方法和软元件等 （另售）	SH-080503CHN
QCPU(Q 模式)/QnACPU 编程手册 (SFC 控制指令篇) 描述了 MELSAP3 的系统构成、性能规格、功能、程序、调试和出错代码等。（另售）	SH-080283C
QCPU(Q 模式）编程手册 (MELSAP-L 篇) 描述了 MELSAP-L 的系统构成、性能规格、功能、程序、调试和出错代码等。（另售）	SH-080076
QCPU(Q 模式)/QnACPU 编程手册 (PID 控制指令篇) 描述了用于执行 PID 控制的专用指令。（另售）	SH-080240C
QCPU(Q 模式）编程手册（结构化文本篇） 描述了结构化文本 (ST) 语言的编程方法。（另售）	SH-080366E
QnPHCPU/QnPRHCPU 编程手册（过程控制指令篇） 描述了用于执行过程控制的专用指令。（另售）	SH-080449CHN
QnACPU 指南 适用于第一次使用 QnACPU 的用户。描述了从创建程序到将创建好的程序写入 CPU 模块，直到调试的一切步骤。还说明了如何最有效地使用 QnACPU。 （另售）	IB-66606
Q2A(S1)/Q3A/Q4ACPU 用户手册 描述了 Q2ACPU(S1)、Q3ACPU 和 Q4ACPU 的性能、功能函数和处理操作。以及规格说明和电源模块、存储卡和基板的处理操作。 （另售）	IB-66608
冗余 CPU 模块 Q4ARCPU 用户参考手册 描述了 Q4ARCPU 的性能、功能和使用方法。还描述了总线开关模块、系统管理模块、电源模块、存储卡和基板的规格和使用方法。（另售）	SH-080230C
CPU 模块 Q2ASCPU 用户参考手册 描述了 Q2ASCPU、Q2ASCPU-S1、Q2ASHCPU 和 Q2ASHCPU-S1 的性能、功能和处理操作。以及电源模块、存储卡、基板的规格、使用等有关内容。（另售）	SH-080215C
QnACPU 编程参考手册 描述了如何创建程序、设备名称、参数和程序类型。（另售）	SH-080229C
QnACPU 编程手册（特殊功能篇） 描述了当使用 QnACPU 时可用于特殊功能模块的专用指令。（另售）	IB-66616
QnACPU 编程手册 (AD57 指令篇) 描述了当使用 QnACPU 时，可用于控制 AD57(S1) 型 CRT 控制器模块的专用指令集。（另售）	IB-66617

手册名称	手册编号
MELSECNET/G 网络系统参考手册（控制网络篇） 描述了 MELSECNET/G 网络系统的控制网络的规格、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080710CHN
Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇） 描述了 MELSECNET/H 网络系统的可编程控制器网络的规格、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080289C
Q 系列 MELSECNET/H 网络系统参考手册（远程 I/O 网络篇） 描述了 MELSECNET/H 网络系统的远程 I/O 网络规范、操作步骤和设置、参数设置、编程和故障排除方法。 （另售）	SH-080290C
QnA 系列 MELSECNET/10 网络系统参考手册 描述了 MELSECNET/10 的一般概述、规格、和部件名称及设置。 （另售）	IB-66620
MELSECNET、MELSECNET/B 数据链接系统参考手册 描述了 MELSECNET(II) 和 MELSECNET/B 的一般概述、规格和部件名称及设置。 （另售）	IB-66350
Q 系列以太网接口模块用户手册（应用篇） 描述了以太网模块的 E-mail 功能、可编程控制器 CPU 状态监视、MELSECNET/H 或 MELSECNET/10 网络系统通信功能、数据链接用指令通信功能、文件传输 (FTP 服务器）和其它功能。（另售）	SH-080285C
QnA 系统以太网接口模块用户手册 描述了以太网接口模块的一般概述、规格和部件名称及设置。（另售）	SH-080146
CC-Link 网络系统模块 A1S61BT11/A1SJ61BT11 用户参考手册 描述了 AJ61QBT11 和 A1SJ61QBT11 的一般概述、规格和部件名称及设置。 （另售）	SH-080212C

8 数据链接指令

类别	处理内容	参考章节
网络刷新指令	刷新指定的网络模块。	8.2 节
QnA 链接专用指令	从目标网络号的目标站中读取 QnACPU 软元件数据。	8.3.1 项
		8.3.2 项
	向目标网络号的目标站的 QnACPU 软元件写入数据。	8.3.3 项
		8.3.4 项
	将数据写入目标网络号的目标站（网络模块）。	8.3.5 项
	将通过 SEND 指令发送的数据读取到 QnACPU 中。	8.3.6 项
	读 / 写时钟数据到其它站并执行远程 RUN/STOP。	8.3.7 项
	从远程 I/O 站的特殊功能模块读取数据。	8.3.8 项
	将数据写入远程 I/O 站的特殊功能模块。	8.3.9 项
A 系列兼容链接指令	从目标网络号的目标站读取 QnACPU 软元件数据。	8.4.1 项
	读取本地站的软元件数据。（只对主站有效）	8.4.2 项
	将数据写入目标网络号的目标站的 QnACPU 软元件。	8.4.3 项
	将数据写入本地站的软元件。（只对主站有效）	8.4.4 项
	从远程 I/O 站的特殊功能模块读取数据。	8.4.5 项
	将数据写入远程 I/O 站的特殊功能模块。	8.4.6 项
路由信息读 / 写指令	读取路由参数指定的数据。	8.5.1 项
	将路由数据写入路由参数指定的区域。	8.5.2 项

备 注

在本章中，如果没有特别指出，指令名按如下方式略称。

- S(P)/J(P)/G(P).ZCOM → ZCOM
- JP/GP.READ → READ
- JP/GP.SREAD → SREAD
- JP/GP.WRITE → WRITE
- JP/GP.SWRITE → SWRITE
- JP/GP.SEND → SEND
- JP/GP.RECV → RECV
- JP/GP.REQ → REQ
- JP/GP.ZNFR → ZNFR
- JP/GP.ZNTO → ZNTO
- J(P).ZNRD → ZNRD
- J(P).ZNWR → ZNWR
- G(P).RFRP → RFRP
- G(P).RTOP → RTOP
- S(P)/Z(P).RTREAD → RTREAD
- S(P)/Z(P).RTWRITE → RTWRITE

8.1 概要

QCPU/QnACPU 可以通过 MELSECNET/G、MELSECNET/H、MELSECNET/10、以太网、MELSECNET/(II，/B)、CC-Link 和串行通信来进行数据链接。*1

链接专用指令是指，用于 CPU 模块从连接在 MELSECNET/G、MELSECNET/H、MELSECNET/10 或 MELSECNET/(II，/B) 中的其它站中读取数据到自站中的指令。

(1) 链接专用指令的类型

链接专用指令包括网络刷新指令、QnA 链接专用指令和 A 系列兼容链接指令。

- 网络刷新指令 : 用于进行指定网络模块的刷新处理。
- QnA 链接专用指令 : 用于 QnACPU 的链接专用指令。
  可以使用网络模块的通道 1 ～通道 8 的 8 个通道进行通讯。
  可以通过控制数据区实现精确控制。
- A 系列兼容链接指令 : 与 AnACPU/AnUCPU 专用指令相同的指令。
- 路由信息读取 / 写入指令 : 读取 / 写入通过路由参数指定的数据。

适用于 MELSECNET/G、MELSECNET/H、MELSECNET/10 或 MELSECNET/(II，/B) 的链接指令是有限的。
当使用 MELSECNET/G、MELSECNET/H 或 MELSECNET/10 时，根据在目标站中的 CPU 模块是 QCPU、QnACPU 或 ACPU，可以使用的指令是不同的。

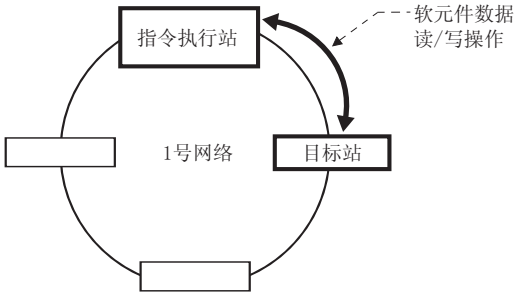
指令分类	访问源 CPU 模块	目标 CPU 模块		
		QCPU	QnACPU	ACPU
QnA 链接专用指令	QCPU	× *2	× *2	× *2
	QnACPU	○	○	×
	ACPU	×	×	×
A 兼容链接专用指令	QCPU	× *2	× *2	× *2
	QnACPU	○	○	○
	ACPU	×	×	×

*1 :MELSECNET/G 可以与系列号前五位数是 09012 或以后的高性能型 QCPU 或通用型 QCPU 连接。
*2 :QCPU 可用的指令在相应网络模块的手册中进行说明。
关于可以使用的指令，请参照下列手册：
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（远程 I/O 网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

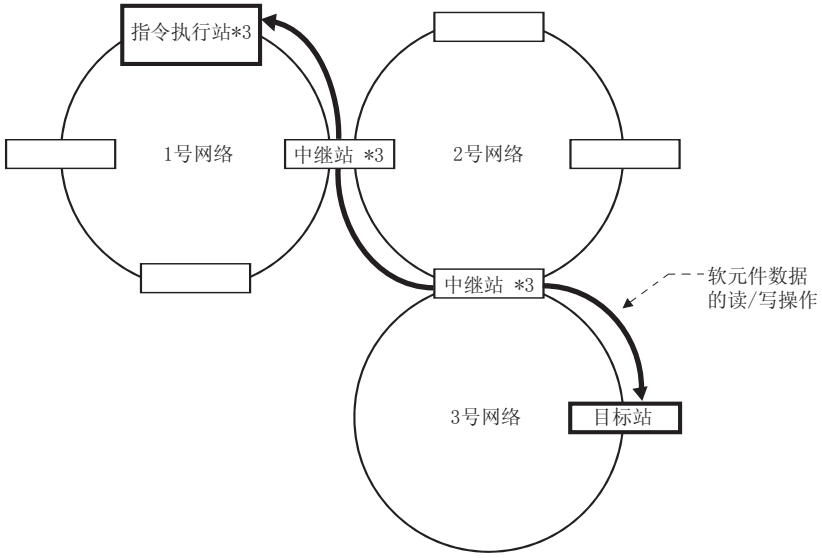
(2) 数据读 / 写范围

(a) 在 MELSECNET/G、MELSECNET/H、MELSECNET/10 中，自站不仅可以对连接在同一网络上的其它站，而且可以对指定网络号的其它连接站执行数据读 / 写操作。

(1) 对同一网络中的连接站进行的读 / 写操作
对同一网络中的连接站进行读 / 写操作时，应将目标站的网络号设定为与自站的网络号一致。

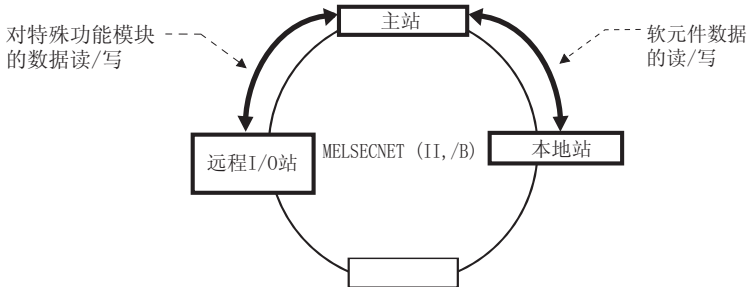


(2) 对指定网络号的其它连接站的读 / 写操作
对指定网络号的连接站进行读 / 写操作时，使用自站网络内的连接站作为中继站，并设置要进行软元件数据读 / 写操作的目标网络号。



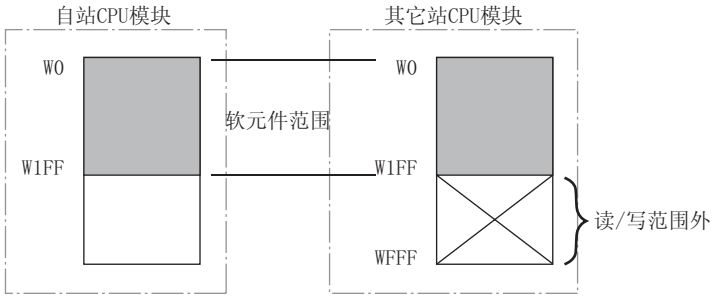
*3: 需要进行路由参数的设定

(b) 在 MELSECNET (II, /B) 中，可以从主站对本地站或远程 I/O 站进行读 / 写操作。



要 点

对于通过数据链接指令进行数据读 / 写的其它站 CPU 模块的软元件，必须在自站 CPU 模块可使用的软元件范围内进行指定。
当其它站 CPU 模块与自站 CPU 模块中可使用的软元件范围不同时必须注意。



(3) QnA 链接专用指令

下列是使用专用 QnA 链接指令时的注意事项。

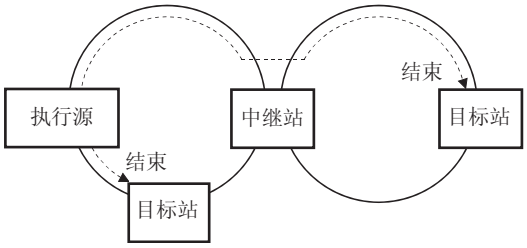
(a) 数据链接 MELSECNET/10 网络模块的同步执行指令有八个使用数据链接指令的通讯区域。用相应的一个通讯区域，同时用 MELSECNET/10 网络模块对数据链执行多指令是不可能的。
如果在 QnACPU 同样的通讯区域用来执行数据链接的一个以上的指令，则由程序来完成，此程序将用数据链接单个指令的结束软元件逐步执行。

(b) 传送结束确认

当用 QnA 专用链接指令进行数据传送时，可以选择传送结束确认或不确认。
（当用 QnA 专用链接指令进行读操作时，只能选择结束确认。）

• 传送结束确认

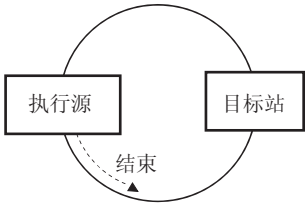
：此为对指定目标站指定通道数据写入的指令结束。
因为处理过程不等待对目标站的写结束通知，因此执行源的指令结束就会更早。



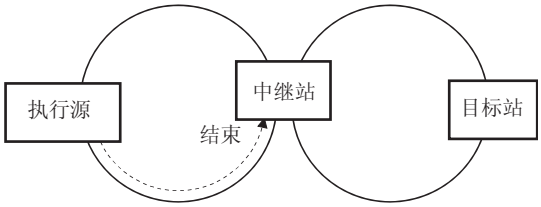
• 无传送结束确认

：如果目标站在自站网络上，从自站传送了数据时指令结束。
如果目标站在另外一个网络上，当数据到达自网络的中继站时就为指令结束。
由于不等待对目标站的写入结束的通知，因此执行源的指令结束变快。

[当目标站在自站网络中时]



[当目标站在不同网络中时]



要 点

1. 为了提高数据可靠性，建议设定 “传送结束确认”并执行此指令。
2. 如果设定成 “无传送结束确认”，如果传送的数据有错误时，只要通讯本身正常结束，传送站将正常结束。
当从多站执行指令到相应同一站时，即使传送的数据内容是正确的，目标站仍然发送 “接受缓冲器满”错误。
然而，在这种情况下，传送站将送出正常结束。

(4) 网络模块通道

(a) 对于 SEND、RECV、READ、WRITE、REQ、ZNRD、ZNWR、SREAD 或 SWRITE 指令

网络模块有八个通道用于执行指令。

八个通道可以同时使用，但同一通道不能被多条指令使用。

应创建一个通过链接特殊继电器 (SB) 进行互锁的程序，避免在同一个通道中同时执行多个指令。



每一指令的互锁信号说明如下：

指令	ZNRD*3	ZNWR*4	—	—	—	—	—	—
	SEND、 RECV、 READ、 WRITE、 REQ、 SREAD、 SWRITE							
	通道 1	通道 2	通道 3	通道 4	通道 5	通道 6	通道 7	通道 8
第 1 个	SB030	SB032	SB034	SB036	SB038	SB03A	SB03C	SB03E
第 2 个	SB230	SB232	SB234	SB236	SB238	SB23A	SB23C	SB23E
第 3 个	SB430	SB432	SB434	SB436	SB438	SB43A	SB43C	SB43E
第 4 个	SB630	SB632	SB634	SB636	SB638	SB63A	SB63C	SB63E

*3 : ZNRD 总是使用通道 1。

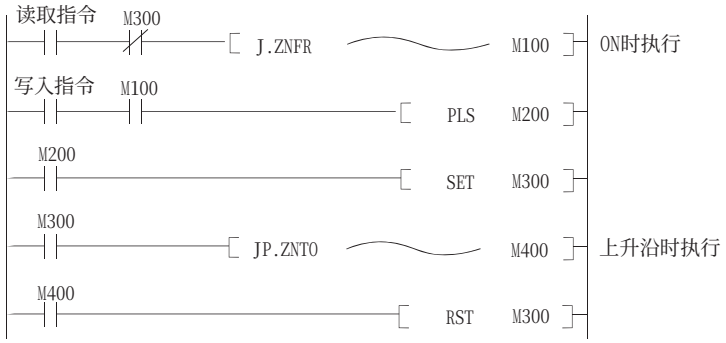
*4 : ZNWR 总是使用通道 2。

(b) 对于 ZNFR 或 ZNTO 指令

不能对安装在远程 I/O 站的相同 I/O 号的特殊功能模块同时执行 ZNFR 和 ZNTO 指令。

应创建一个互锁程序，使前一个指令未结束时，下一指令不会执行。

例 在 ON 时执行（读取）及在上升沿时执行（写入）时



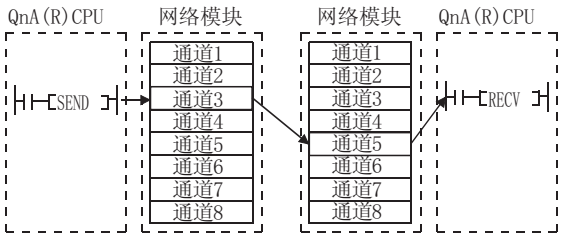
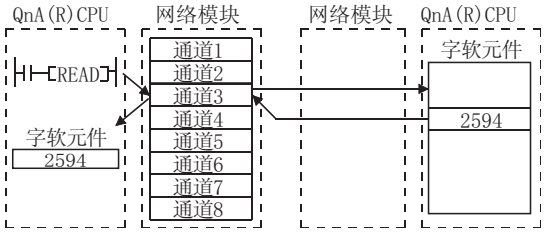
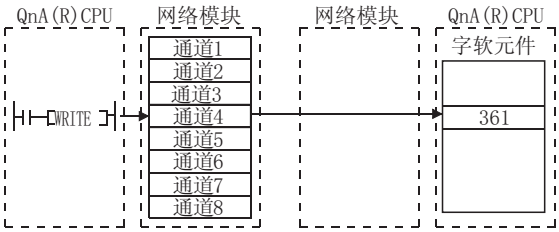
要 点

注意 ZNFR、ZNTO 指令与 (a) 中说明的 SEND、RECV、READ、WRITE、REQ、ZNRD、ZNWR、SREAD、SWRITE 指令不同，没有可以显示指令执行状态的特殊链接继电器 (SB)。

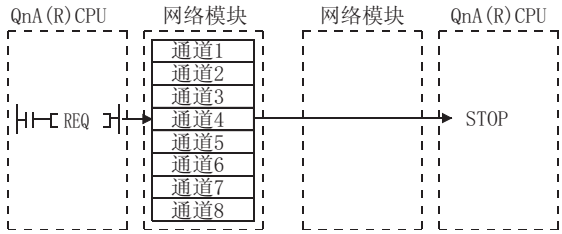
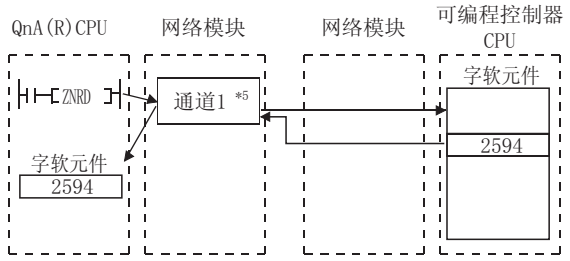
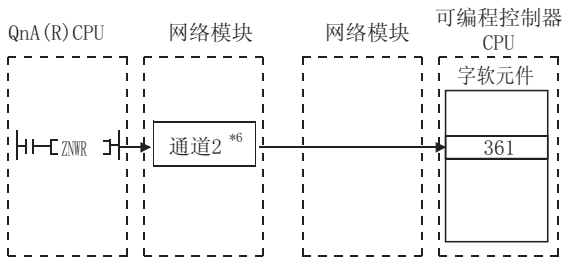
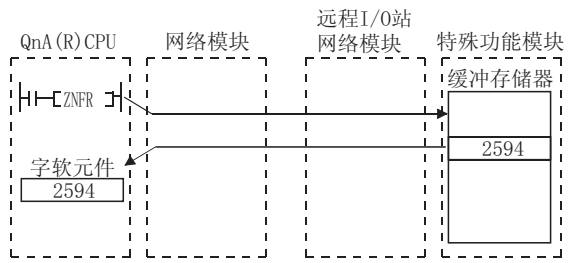
(5) 各指令的概要

以下说明可以用于 QnACPU 的指令。

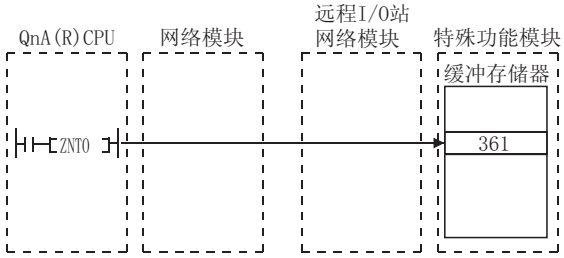
通道 1 到 8 是 SEND/RECV/READ/WRITE/REQ/SREAD/SWRITE 指令的公共区域。

指令	详细说明	指令执行站（自站）	目标站		
		站类型	站类型	可编程控制器 CPU 类型	
				QCPU QnA(R) CPU	ACPU
SEND RECV	QnA(R)CPU 站之间数据发送 (SEND) 与接收 (RECV)。 	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	×
READ SREAD	从其它站的字软元件中读数据。 （通过 SREAD，可以打开目标站上的软元件） 	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	×
WRITE SWRITE	对其它站的字软元件写入数据。 （通过 SWRITE，可以打开目标站上的软元件） 	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	×

在指令格式上没有操作区别 JP. [ ] 和 GP. [ ]、和 J. [ ] 和 G. [ ]。

指令	详细说明	指令执行站（自站）	目标站		
		站类型	站类型	可编程控制器CPU类型	
				QCPU QnA(R) CPU	ACPU
REQ	对其它站执行“远程RUN/STOP”和“时钟数据读/写”。 	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	×
ZNRD	从另一个站的字软元件中读数据。  *5：通道1固定	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	○
ZNWR	对另一个站的字软元件写数据。  *6：通道2固定	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	控制站 普通站 远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	○	○
ZNFR	从远程I/O站的特殊功能模块读取缓冲存储器数据。 	远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	远程I/O站 AJ72QLP25(G) AJ72QBR15 AJ72QLR25 AJ72LP25(G) AJ72BR25 AJ72LR25	－	－

（转下页）

指令	详细说明	指令执行站（自站）	目标站		
		站类型	站类型	可编程控制器 CPU 类型	
				QCPU QnA(R) CPU	ACPU
ZNTO	对远程 I/O 站的特殊功能模块写入缓冲存储器数据。  	远程主站 多远程主站 并行远程主站 多远程子主站 并行远程子主站	远程 I/O 站 AJ72QLP25(G) AJ72QBR15 AJ72QLR25 AJ72LP25(G) AJ72BR25 AJ72LR25	-	-

要 点

链接专用指令注意事项

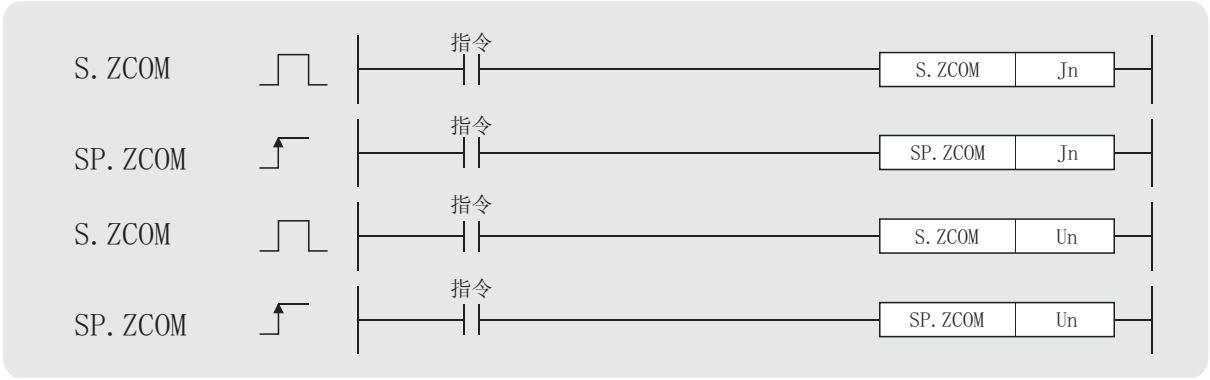
1. 在同时使用 QnA(R)CPU 和 AnUCPU 的系统中，不要从 QnA(R)CPU 对其它站的 AnUCPU 执行以下无法执行的指令。
   如果已经执行，AnUCPU 可能会进入 “MAIN CPU DOWN” 或 “WDT ERROR” 状态而停止运行。
   (1) SEND (2) READ (3) SREAD (4) WRITE (5) SWRITE and (6) REQ
2. 如果希望对同一网络中的全部站执行这些指令，应通过组指定只对 QnA(R)CPU 执行指令。

8.2 网络刷新指令

8.2.1 网络刷新 (S(P)/J(P)/G(P).ZCOM)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

(1) 使用了 QCPU 时



Jn ： 自站网络号 (BIN16 位)

Un ： 自站网络模块的起始 I/O 号 (BIN16 位)

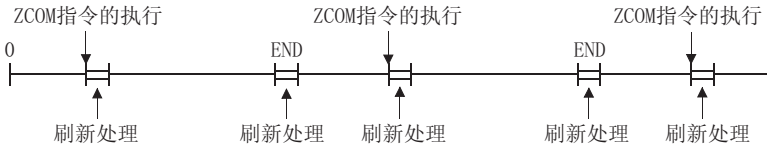
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

ZCOM 指令用于在顺控程序执行过程中的任意时间执行刷新。
ZCOM 指令执行的刷新目标如下所示：

- MELSECNET/G 的刷新 (当设定了刷新参数时)
- MELSECNET/H 的刷新 (当设定了刷新参数时)
- CC-Link 的自动刷新 (当设定了刷新软元件时)
- 智能功能模块的自动刷新 (当设定了自动刷新时)

功 能

(1) 当执行了 ZCOM 指令后，CPU 模块暂时停止处理顺控程序，而执行由 Jn/Un 指定的网络模块的刷新处理。



(2) ZCOM 指令不执行下列通用数据处理

(a) CPU 模块与编程工具之间的通讯处理

(b) 其它站的监视处理

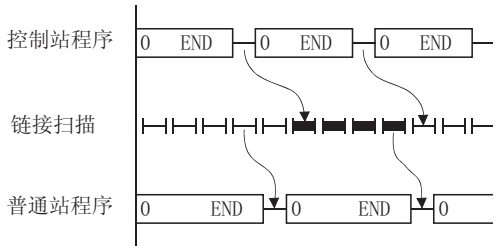
(c) 通过串行通讯模块，读取另外一个智能功能模块缓冲内存的处理

(d) MELSECNET/H 数据的低速周期传送

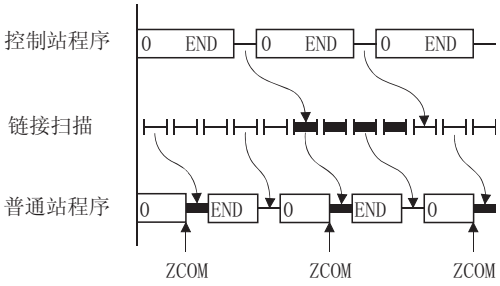
(3) 可编程控制器网络 *1

(a) 当自站顺控程序的扫描时间长于其它站的扫描时间时，ZCOM 指令用来保证来自其它站的数据被正确合并。

(1) 未使用 ZCOM 指令时的数据通讯举例



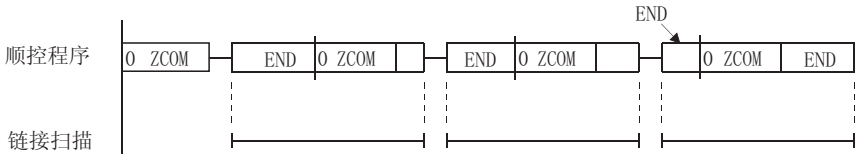
(2) 使用了 ZCOM 指令时的数据通讯举例



关于可编程控制器网络 *1 的传送延迟时间，请参照以下手册：

- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）

(b) 当链接扫描时间长于顺控程序扫描时间时，即使使用 ZCOM 指令数据通讯也不能进行。

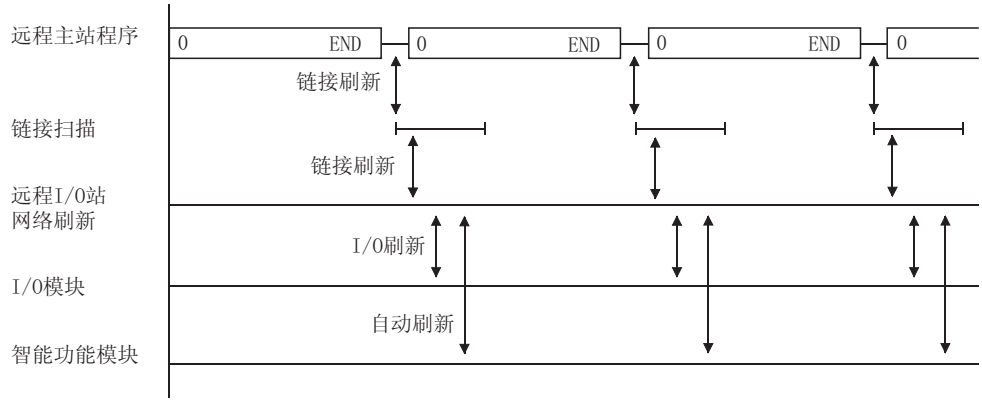


*1 ：在 MELSECNET/G 网络系统中，为控制网络。

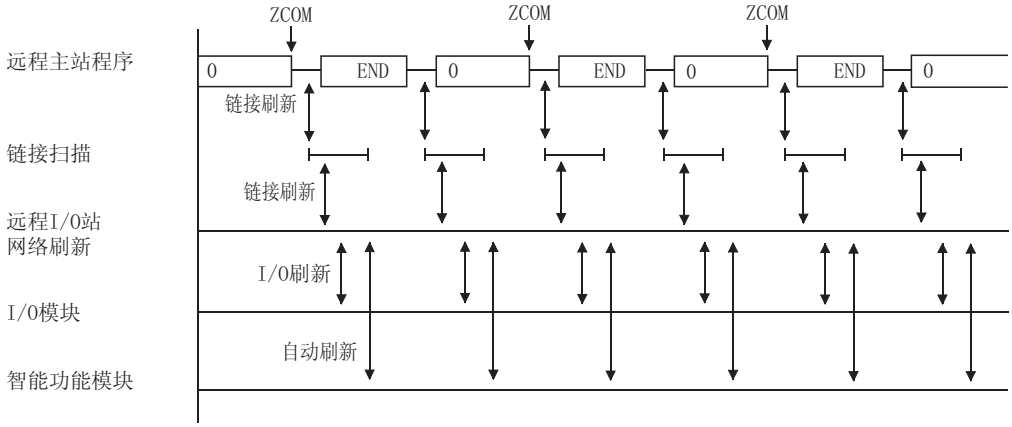
(4) 远程 I/O 网络

远程主站的链接刷新由 CPU 模块的 “END 处理”来执行。
因为链接扫描是在链接刷新结束时执行，链接扫描与 CPU 模块程序 “同步”。
在远程主站使用 “ZCOM”指令时，链接刷新在 ZCOM 指令执行点执行，并且在链接刷新结束时执行链接扫描。
因此，在远程主站使用 ZCOM 指令可以加速远程 I/O 站的发送或接收处理速度。

(1) 不使用 ZCOM 指令时



(2) 使用 ZCOM 指令时



远程 I/O 网络的传送延迟时间的详细内容，请参照以下手册：

• Q 系列 MELSECNET/H 网络系统参考手册（远程 I/O 网络）

(5) ZCOM 指令可以在顺控程序中无限量使用，然而，每执行一次刷新操作都将使顺控程序扫描时间延长，并增加了刷新操作所需要的时间。

(6) 在自变量中指定 “Un”可将目标指定为网络模块或智能模块。
在这种情况下，智能功能模块的缓冲内存将自动刷新（代替 FROM/TO 指令）。

(7) 只有在使用通用型 QCPU 时，执行 ZCOM 指令时的处理中断才有效。然而，当刷新数据用于中断程序时，数据会被分割。

要 点

1. ZCOM 指令不能用于固定周期执行类型的程序或中断程序。
2. 冗余 CPU 在使用 ZCOM 指令时有限制。
有关详细内容请参见以下手册。
• QnPRHCPU 用户手册（冗余系统篇）

运算错误

(1) 在以下发生运行错误情况中，错误指示 (SM0) 变为 ON，并且出错代码存储在 SD0 中。
• 指定的网络号没有连接到自站 （出错代码： 4102)
• 指定首 I/O 号的模块不是网络模块或链接模块（除通用型 QCPU)
（出错代码： 2111)
• 指定起始 I/O 号的模块不是网络模块或链接模块。（只对于通用型 QCPU)
（出错代码： 4102)

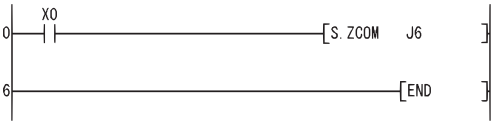
要 点

希望只与外围设备进行通讯时，应使用 COM 指令。（参见 7.6.9 项、9.13 节）。

程序示例

(1) 当 X0 变为 ON 时，下列程序对网络号为 6 的网络模块进行链接刷新操作。

[梯形图模式]

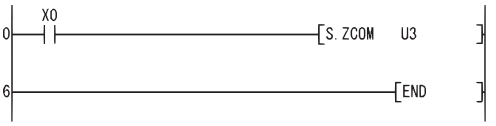


[列表模式]

步	指令	软元件
0	LD	X0
1	S.ZCOM	J6
6	END	

(2) 当 X0 变为 ON 时，下列程序对安装在 I/O 号从 X/Y30 到 X/Y4F 的网络模块进行链接刷新操作。

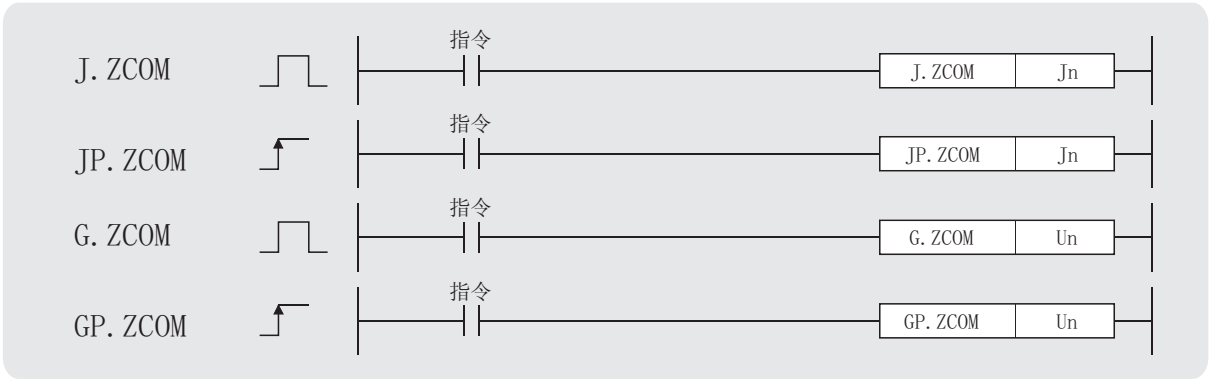
[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	S.ZCOM	U3
6	END	

(2) 使用 QnACPU/Q4ARCPU 时



Jn : 自站网络号 (BIN16 位)

Un : 自站网络模块的起始 I/O 号 (BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

在下列情况下使用 ZCOM 指令：

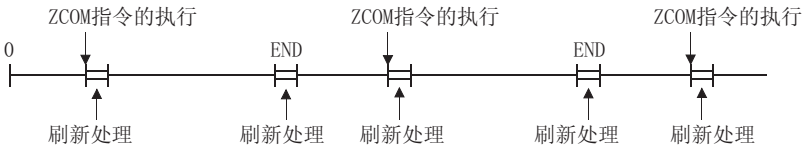
(a) 在远程 I/O 网络中，希望加快与远程 I/O 站的通讯处理速度时。

(b) 在可编程控制器网络中，需要与不同扫描时间的其它站进行可靠的数据交换时。

功 能

(1) 当执行了 ZCOM 指令时，QnACPU 暂时停止顺控程序的处理，来进行 Jn/Un 指定的网络模块的刷新处理。

注意不能执行 MELSECNET/10 的慢速周期刷新。



(2) ZCOM 不能执行下列一般数据处理。

(a) CPU 模块与编程工具间的通讯处理

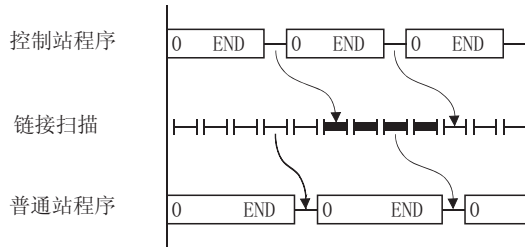
(b) 其它站的监视处理

(c) 通过串行通讯模块对另一个智能功能模块缓冲内存的读处理

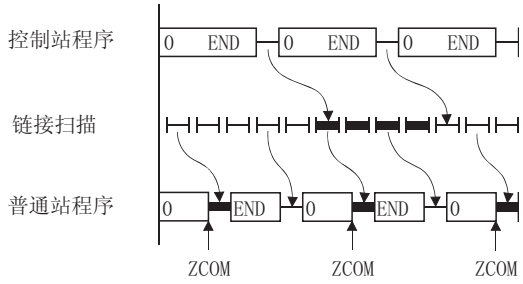
(3) 可编程控制器到可编程控制器网络

(a) 当自站顺控程序的扫描时间长于其它站的扫描时间，ZCOM 指令用来保证来自其它站的数据正确组合。

(1) 未使用 ZCOM 指令时的数据通讯举例



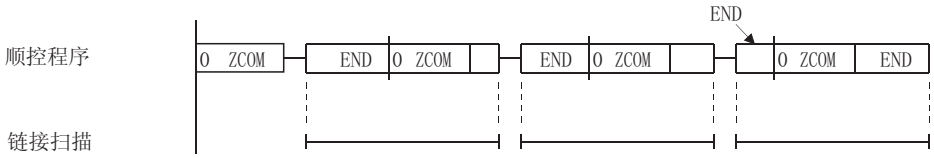
(2) 使用了 ZCOM 指令时的数据通讯举例



关于可编程控制器网络的传送延迟时间，请参考以下手册：

• QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册

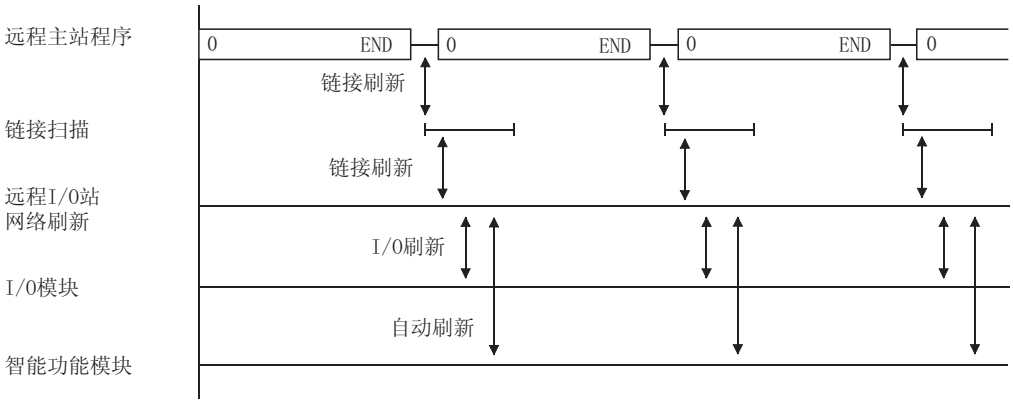
(b) 当链接扫描时间长于顺控程序的扫描时间时，即使使用 ZCOM 指令也不能加快数据通讯的速度。



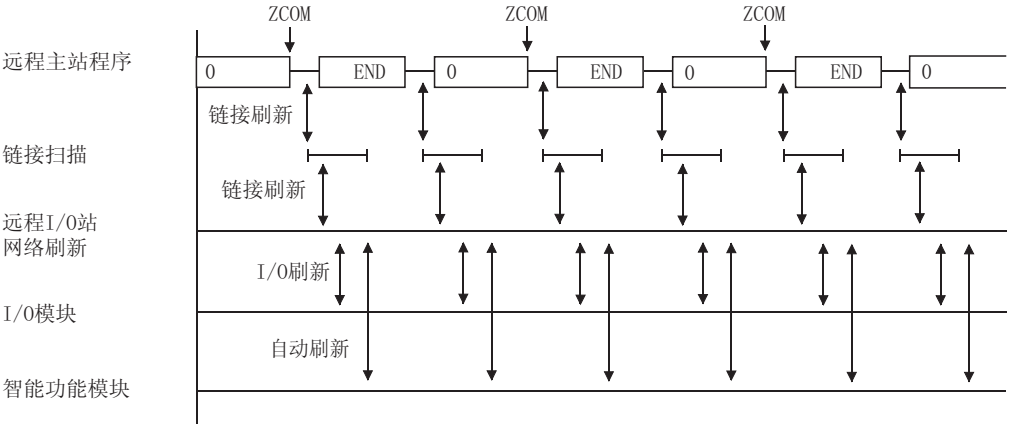
(4) 远程 I/O 网络

通过 CPU 模块的 “END 处理”执行远程主站的链接刷新。
因为链接扫描是在链接刷新结束时执行，链接扫描与 CPU 模块程序 “同步”。
在远程主站使用 ZCOM 指令时，链接刷新是在 ZCOM 指令执行点完成的，而链接扫描是在链接刷新结束完成的。
因此，在远程主站使用 ZCOM 指令可加速远程 I/O 站的发送 / 接收处理速度。

(1) 没有使用 ZCOM 指令时



(2) 使用了 ZCOM 指令时



关于 I/O 网络的传送延迟时间的详细内容，请参考以下手册：

• QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册

(5) ZCOM 指令可以在顺控程序中无限次使用。

然而，每进行一次刷新操作都将使顺控程序扫描时间延长，增加了刷新操作所需要的时间。

运算错误

(1) 在以下发生运行错误情况中。错误指示 (SM0) 开，并且出错代码储存在 SD0 中。

- 指定的网络号没有连接到自站上 （出错代码： 4102）
- 指定首 I/O 号的模块不是网络模块或链接模块 （出错代码： 4102）

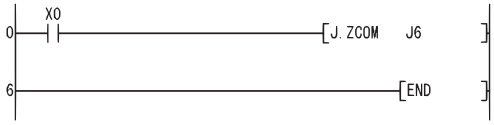
要 点

希望只与外围设备进行通讯时，应使用 COM 指令。（参见 7.6.9 项、9.13 节）。

程序示例

(1) 当 X0 变为 ON 时，下列程序将对网络号为 6 的网络模块进行链接刷新操作。

[梯形图模式]

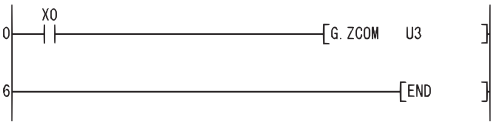


[列表模式]

步	指令	软元件
0	LD	X0
1	J. ZCOM	J6
6	END	

(2) 当 X0 变为 ON 时，下列程序将对安装在 I/O 号为从 X/Y30 到 X/Y4F 的网络模块进行链接刷新操作。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	X0
1	G. ZCOM	U3
6	END	

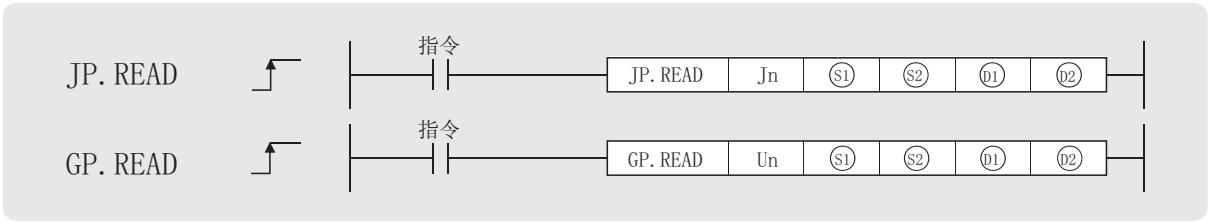
8.3 QnA 链接专用指令

8.3.1 从其它站中读取字软元件数据 (JP/GP.READ)



*1: 关于 QCPU 可以使用的 READ 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10 或以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
S1	--	△*3	○	--					
S2	--	△*3	○	--					
D1	--	△*3	○	--					
D2	△*2	△*3	○	--					

*2: 可以使用 X、Y、M、L、F、V 和 B。
*3: 可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Jn	自站网络号。*4	BIN 16 位
Un	自站网络模块的起始 I/O 号。*5	
S1	存储控制数据的自站的起始软元件。	软元件名称
S2	存储读取数据的目标站的起始软元件。	
D1	存储读取数据的自站的起始软元件。	
D2	指令结束时，软元件将在 1 次扫描期间为 ON。*6	位

*4 : 自站网络号的指定如下。
- 1 到 239: 网络号
- 254: 由设定为其它站可以访问的有效模块来指定的网络。

*5 : 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 : 局部软元件和各程序的文件寄存器均不能作为设定数据使用的软元件。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 READ 指令才能被执行。
   (READ 指令不能在连接到 MELSECNET/10 或以太网的 ACPU 中执行。)
2. 对于通过 READ 指令进行数据读取的目标站 CPU 模块的软元件，应在自站 CPU 模块可使用的范围内进行指定。
   (目标站 CPU 模块的读取目标的起始软元件号 (S2)) + (读取点数 - 1)
   ≦ (指定的最终软元件号 (S2) (自站 CPU 模块))

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	有传送结束确认：将位 0(b0) 设定成 1(固定方式)。	0001H 0081H	用户
	异常时结束类型	设置异常完成时的时钟数据设定状态。 不设定时钟数据 ：将位 7(b7) 设定成 0。 设定时钟数据 ：将位 7(b7) 设定成 1。 (附加在 (S1)+11 的后面)		
(S1) + 1	结束状态	存储指令结束时的状态。 0 ：没有错误(正常结束) 不是 0：出错代码*7	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*8	1 到 8	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	设定读取软元件数据的目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	设定目标站的站号。	1 到 64	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	通过 READ 指令进行的读取操作在 (S1)+8 指定的监控时间内未完成时，设定重复传送的次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定通过 READ 指令进行读取时的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 中指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒	用户
(S1) + 9	读取数据长度	设定要读取的数据块数目。	1 到 480*10	用户
(S1) + 10	虚拟	禁止使用	--	
(S1) + 11	时钟设定标记 (只在异常时设定)	存储时钟数据的有效 / 无效状态。 0 ：时钟数据无效 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 (只在异常时设定)	月和年(年份后 2 位)以 BCD 编码存储。 高 8 位：月(01H 到 12H)、低 8 位：年(00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。. 高 8 位 ：小时(00H 到 23H)、低 8 位：日(01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒(00H 到 59H)、低 8 位：分钟(00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期(00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站号。但如果 (S1)+1 结束状态是“通道使用中”，则不进行存储。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站号。但如果 (S1)+1 结束状态是“通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*7 ： 关于出错代码，请参照以下手册：
- QnA/Q4AR 用 MELSECNET/10 网络系统参考手册
- QnA 系统以太网接口模块用户手册(详细篇)

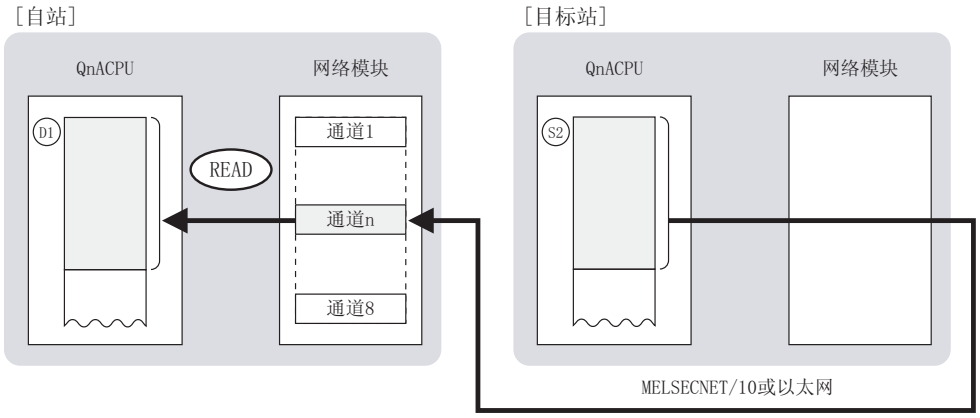
*8 ： 关于网络模块的通道的详细内容，请参照 8.1 节。

*9 ： 在 Jn 中设定了 254 时进行此指定。

*10: 如果((D1) 中指定的软元件号)+(接收数据长度)超出了 (D1) 中指定的软元件范围，超出了指定范围的点数数据会存储在下一个软元件的起始部分。

☆功 能

(1) 将控制数据的目标站网络号 (Ⓢ1 + 4) 和目标站号 (Ⓢ1 + 5) 中指定的、MELSECNET/10 或以太网连接站的 Ⓢ2 中指定的的字软元件以后的数据，保存到自站的 Ⓓ1 中指定的软元件的后面。
当目标站中软元件数据的读取操作已经完成时， Ⓓ2 中指定的结束软元件将会变为 ON。



(2) 可以对连接在 MELSECNET/10 或者以太网中的指定站以及连接在自站网络中站点执行软元件数据的读取。

(3) 对同一个通道不能在 2 处以上执行链接专用指令。
如果在 2 处以上执行条件同时成立，将会自动执行握手（数据交换），后面的链接专用指令将无法处理。

(4) 可以通过以下方式确认 READ 指令的执行状态和正常 / 异常结束：

(a) 通讯指示标志 *11 : READ 指令执行时为 ON，在读取结束时的扫描的 END 处理时变为 OFF。

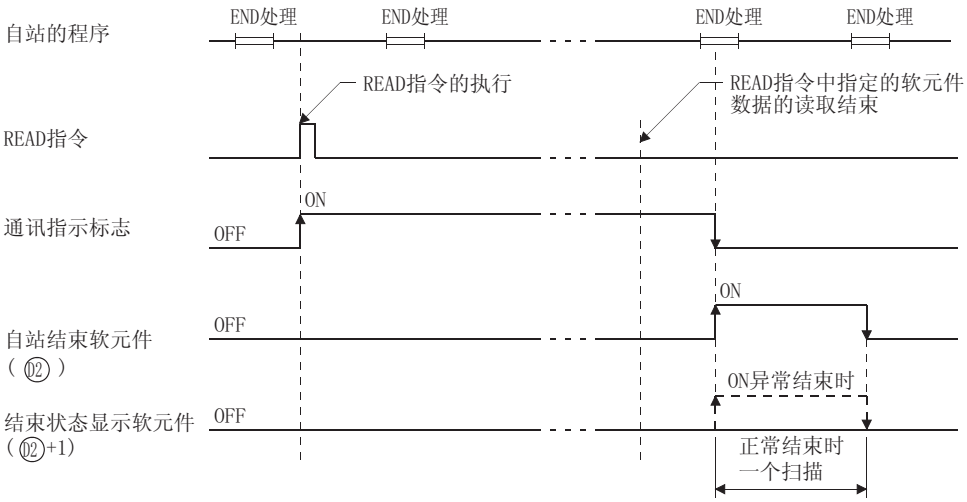
(b) 自站结束软元件 (Ⓓ2) : 通过 READ 指令进行的读取结束的扫描的 END 处理时变为 ON，并在进行下一个 END 处理时变为 OFF。

(c) 结束状态显示软元件 (Ⓓ2+1) : 根据 READ 指令结束时的状态，变为 ON 或 OFF。
- 正常结束：保持 OFF 状态不变。
- 异常结束：READ 指令结束时的扫描的 END 处理时变为 ON，并在进行下一个 END 处理时变为 OFF。

*11: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[READ 指令执行时的自站的动作]



运算错误

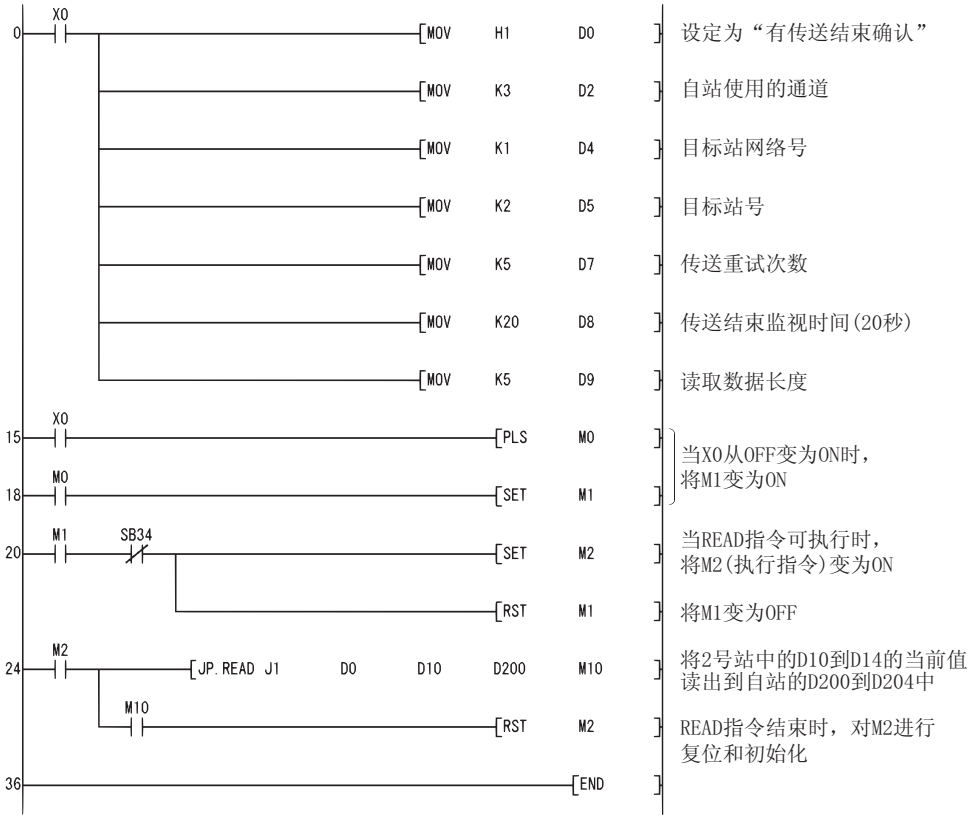
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4100)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序将连接在 MELSECNET/10 中的 2 号站的 D10 到 D14 的数据读取到自站的 D200 到 D204 中。

[梯形图模式]



[列表模式]

步	指令	软元件				
0	LD	X20				
1	MOV	H1	D0			
3	MOV	K3	D2			
5	MOV	K1	D4			
7	MOV	K2	D5			
9	MOV	K5	D7			
11	MOV	K20	D8			
13	MOV	K5	D9			
15	LD	X20				
16	PLS	M0				
18	LD	M0				
19	SET	M1				
20	LD	M1				
21	ANI	SB34				
22	SET	M2				
23	RST	M1				
24	LD	M2				
25	JP.READ	J1	D0	D10	D200	M10
34	AND	M10				
35	RST	M2				
36	END					

2 关于 CC-Link、串行通信
在软件版本 J 或更新版本的主站模块 (A(1S)J61QBT11) 中可以使用 CC-Link。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○				--		
(S2)	--	△*2	○				--		
(D1)	--	△*2	○				--		
(D2)	△*1	△*2	○				--		

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Un	自站起始 I/O 号。*3	BIN 16 位
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储读取数据的目标站的起始软元件。	
(D1)	存储读取数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的软元件。*4	位

*3 : 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件及各程序的文件寄存器均不能作为设定数据用的软元件。关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 READ 指令才能被执行。
(READ 指令不能在连接到 CC-Link 或串行通信的 ACPU 中执行。)
2. 对于通过 READ 指令进行数据读取的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
（目标站 CPU 模块的读取目标的起始软元件号 (S2) ）+（读取点数 - 1）
≦（(S2) 指定的最终软元件号（自站 CPU 模块））

控制数据

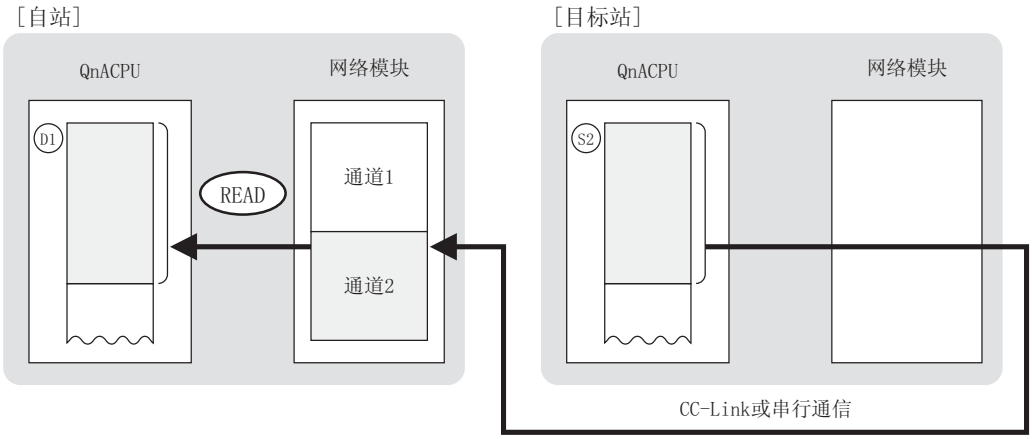
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0001H：固定的（有传送结束确认）。	0001H	用户
(S1) + 1	结束状态	存储指令结束时的状态。 0 ：没有错误（正常结束） 不是 0：出错代码 *5	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 ：使用 CH1 侧 2 ：使用 CH2 侧	1、2	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 (CC-Link)	指定存储数据的目标站号。 0：指定的主站 1 到 64：指定的本地站	0 到 64	用户
	特殊功能模块站号 （串行通信模块）	指定存储数据的目标站号。 0 到 31：站点号	0 到 31	
(S1) + 7	传送重试次数	如果通过 READ 指令进行的读取操作在 (S1)+8 指定的监控时间内未完成，则设定传送重试次数。	0 到 15	用户
	传送重试结果	存储重复处理的次数。		系统
(S1) + 8	传送结束监视时间	以秒为单位设定 READ 指令读取的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 指定的次数传送重试。	1 到 32767 0：1010 秒固定的（默认）	用户
(S1) + 9	读取数据长度	设定要读取的数据块数目。	1 到 480*6	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12	虚拟	禁止使用	--	--
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 ：关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）（新增调制解调器功能版）
- CC-Link 系统主站 / 本地站模块用户手册（型号 AJ61QBT11/A1SJ61QBT11）

*6 ：当（由 (D1) 指定的软元件号）+（接收数据长度）超出了由 (D1) 指定的软元件的范围时，超出了指定范围的点数的数据存储在下一个软元件的起始部分。

☆功 能

(1) 把存储在控制数据的主站 / 本地站号或特殊功能模块站号 (Ⓢ1 + 6) 所指定的站的Ⓢ2 中指定的字软元件后面的数据，保存到自站的 Ⓓ1 中指定的软元件后面的软元件中。
当目标站中软元件数据的读取操作已经完成时， Ⓓ2 中指定的结束软元件将会变为 ON。



(2) 对于 CC-Link 或串行通信的连接站，可以进行软元件数据的读取。

(3) 在同一通道中，链接专用指令不能在多个位置上同时执行。
当在两个或更多位置同时执行的条件成立时，将自动进行握手 (数据交换)，因此后面的链接专用指令将无法进行处理。

(4) READ 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *7 : READ 指令执行时变为 ON，在读取操作结束时的扫描的 END 处理时变为 OFF。

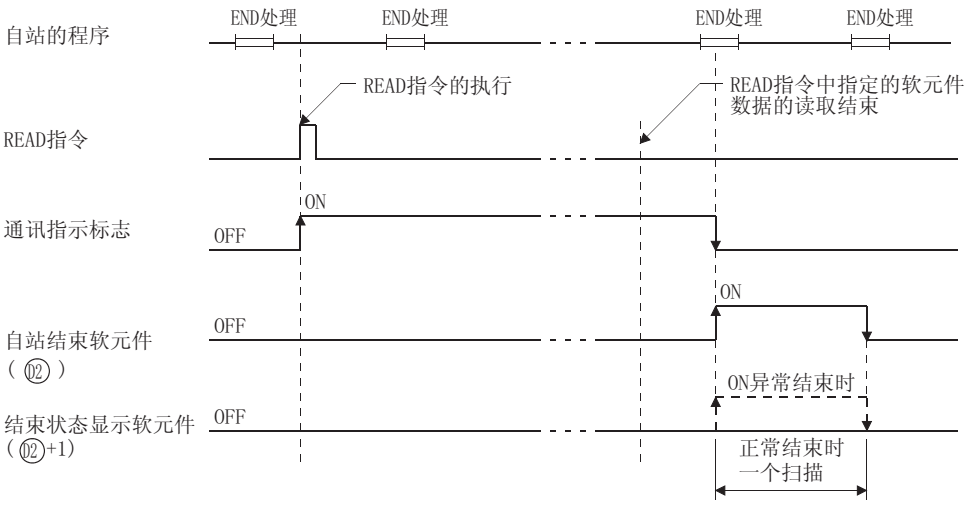
(b) 自站结束软元件 (Ⓓ2) : 通过 READ 指令执行的读取操作结束时的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(c) 结束状态显示软元件 (Ⓓ2+1) : 根据 READ 指令结束时的状态而 ON/OFF。
- 正常结束：保持 OFF 状态不变。
- 异常结束：在 READ 指令结束时的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*7: 所用通道的通讯指示标志如下所示：

通道号	1	2
通讯指示标志号 .	SB30	SB32

[READ 指令执行时的自站的动作]



运算错误

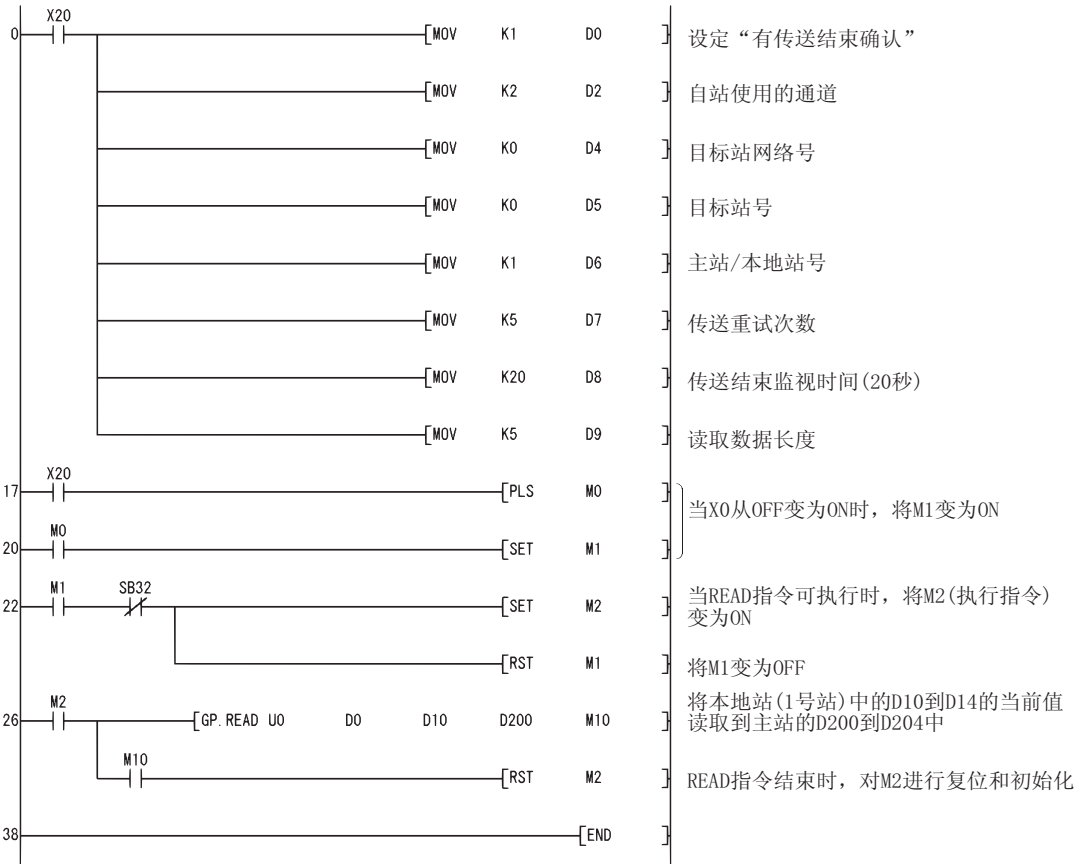
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4100)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序将连接在 CC-Link 中的本地站（1 号站）的 D10 到 D14 的数据读取到主站的 D200 到 D204 中。

[梯形图模式]



[列表模式]

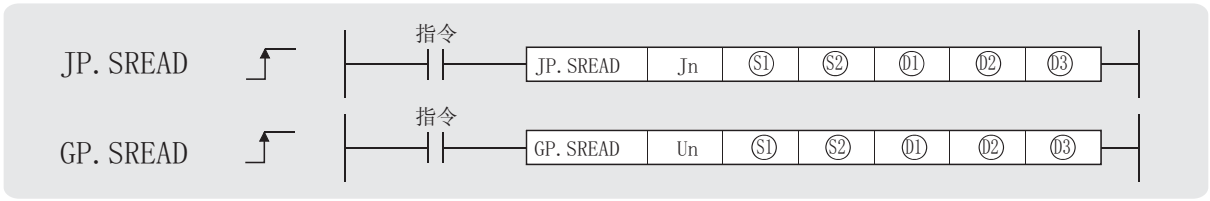
步	指令	软元件				
0	LD	X20				
1	MOV	K1	D0			
3	MOV	K2	D2			
5	MOV	K0	D4			
7	MOV	K0	D5			
9	MOV	K1	D6			
11	MOV	K5	D7			
13	MOV	K20	D8			
15	MOV	K5	D9			
17	LD	X20				
18	PLS	M0				
20	LD	M0				
21	SET	M1				
22	LD	M1				
23	ANI	SB32				
24	SET	M2				
25	RST	M1				
26	LD	M2				
27	GP.READ	U0	D0	D10	D200	M10
36	AND	M10				
37	RST	M2				
38	END					

8.3.2 从其它站读取字软元件数据（JP/GP.SREAD）

通用型 UD *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 SREAD 指令的详细内容，请参照以下手册。
MELSECNET/G 网络系统参考手册（控制网络篇）
• Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
• Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	—	△*3	○			—			
(S2)	—	△*3	○			—			
(D1)	—	△*3	○			—			
(D2)	△*2	△*3	○			—			
(D3)	△*2	△*3	○			—			

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

设定数据

设定数据	含义	数据类型
Jn	自站网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储要读取数据的目标站的起始软元件。	
(D1)	存储读取数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的自站软元件。*6	位
(D3)	指令结束后将要 ON 一个扫描的目标站软元件。*6	

*4 ： 自站网络号的指定如下。
• 1 到 239：网络号
• 254：设定为其它站可以访问的有效模块中指定的网络。
*5 ： 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ： 局部软元件及各程序的文件寄存器均不能作为设定数据用的软元件。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 SREAD 指令才能被执行。
   (SREAD 指令不能在连接到 MELSECNET/10 或以太网的 ACPU 中执行。)
2. 对于通过 SREAD 指令进行数据读取的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
   (目标站 CPU 模块的读取目标的起始软元件号 (S2)) + (读取点数 - 1)
   ≦ ((S2) 指定的最终软元件号 (自站 CPU 模块))

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	有传送结束确认：将位 0(b0) 设定成 1(固定)。	0001H 0081H	用户
	异常时结束类型	设置异常结束时的时钟数据设定状态。 不设定时钟数据：将位 7(b7) 设定成 0。 设定时钟数据：将位 7(b7) 设定成 1。 (附加在 (S1)+11 的后面)		
(S1) + 1	结束状态	存储指令结束时的状态。 0 : 没有错误(正常结束) 不是 0：出错代码 *7	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*8	1 到 8	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	设定进行软元件数据读取的目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	设定目标站的站号。	1 到 64	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	如果通过 SREAD 指令进行的读取操作在 (S1)+8 指定的监控时间内未完成，则设定传送重试次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定通过 SREAD 指令进行读取时的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒	用户
(S1) + 9	读取数据长度	设定要读取的数据块数目。	1 到 480*10	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 (只在异常时设定)	存储时钟数据的有效 / 无效状态。 0：时钟数据无效。 1：时钟数据有效。	--	系统
(S1) + 12	时钟数据 (只在异常时设定)	月和年(年份后 2 位)以 BCD 编码存储。. 高 8 位：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位：00H、低 8 位：星期 (00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站号。但如果 (S1)+1 结束状态是 “通道使用中”，则不进行存储。	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站号。但如果 (S1)+1 结束状态是 “通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*7 : 关于异常时的出错代码，请参照以下手册：
- QnA/Q4AR 用 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册(详细篇)

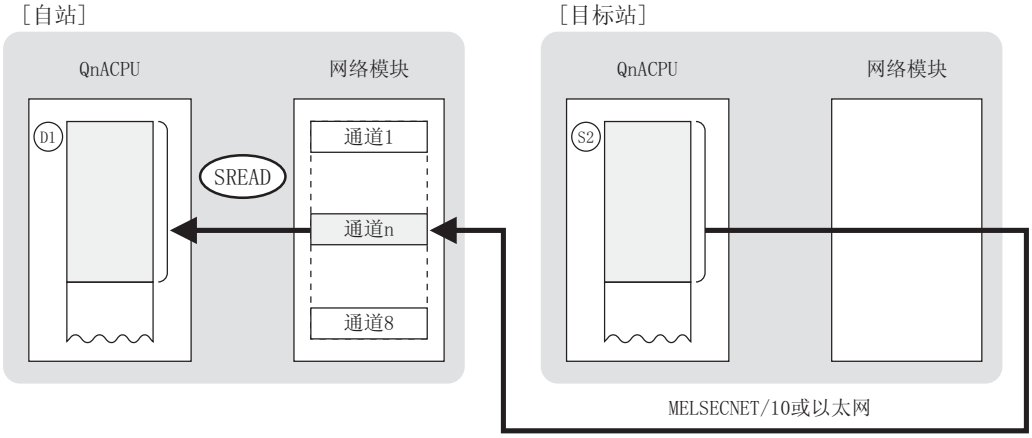
*8 : 关于网络模块的通道的详细内容，请参照 8.1 节。

*9 : 在 Jn 中设定了 254 时进行此指定。

*10: 如果(由 (D1)) 指定的软元件号)+(接收数据长度)超出了由 (D1) 指定的软元件范围，超出了指定范围的点数数据会存储在下一个软元件的起始部分。

☆功 能

(1) 将控制数据的目标站网络号 (Ⓢ1 + 4) 和目标站站号 (Ⓢ1 + 5) 所指定的 MELSECNET/10 或以太网连接站的 Ⓢ2 中指定的字软元件后面的数据，保存到自站的 Ⓓ1 指定的软元件后面的软元件中。
当目标站中软元件数据的读取操作已经完成时，Ⓓ2 中指定的结束软元件将变为 ON。
在目标站中，当 Ⓢ2 中指定的软元件数据传送已经完成时，Ⓓ3 中指定的软元件将变为 ON。



(2) 除了连接在自站网络中的站之外，对于连接在 MELSECNET/10 或以太网中的指定网络号的站也可以执行软元件数据的读取操作。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) SREAD 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *11 : SREAD 指令执行时变为 ON，在读取操作结束时的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (Ⓓ2) : 在通过 SREAD 指令执行的读取操作结束时的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

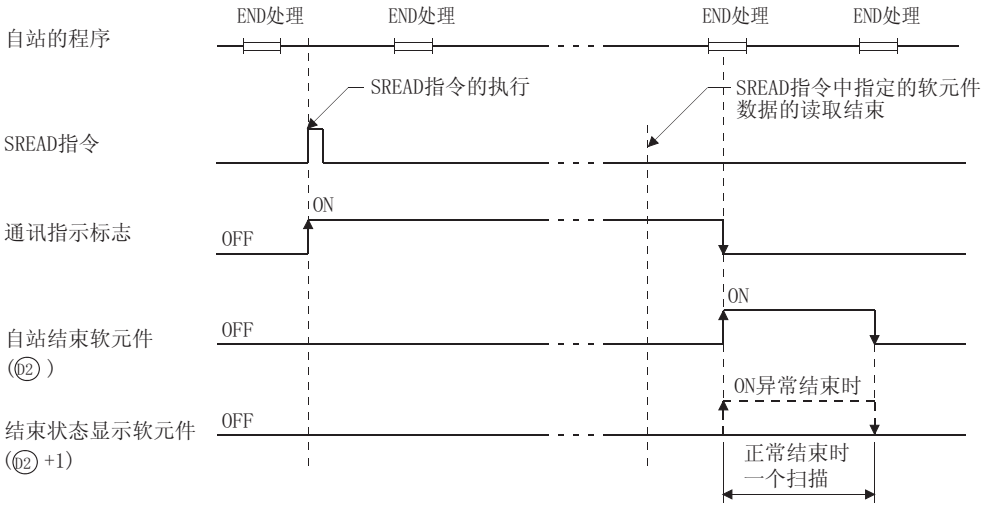
(c) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 SREAD 指令结束时的状态而 ON/OFF。
- 正常结束：保持 OFF 状态不变。
- 异常结束：在 SREAD 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(d) 目标站的结束软元件 (Ⓓ3) : 当由 SREAD 指令指定的软元件的数据传送已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

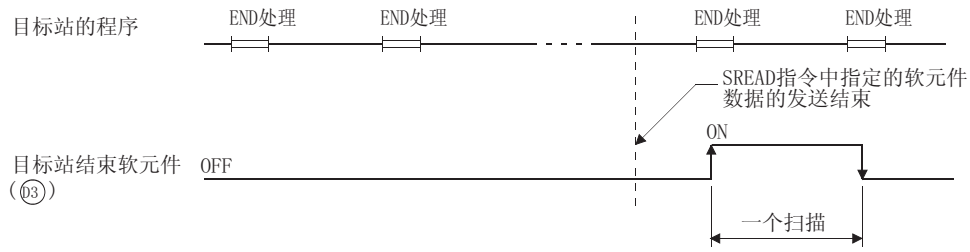
*11: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[SREAD 指令执行时的自站的动作]



[SREAD 指令执行时的目标站的动作]



运算错误

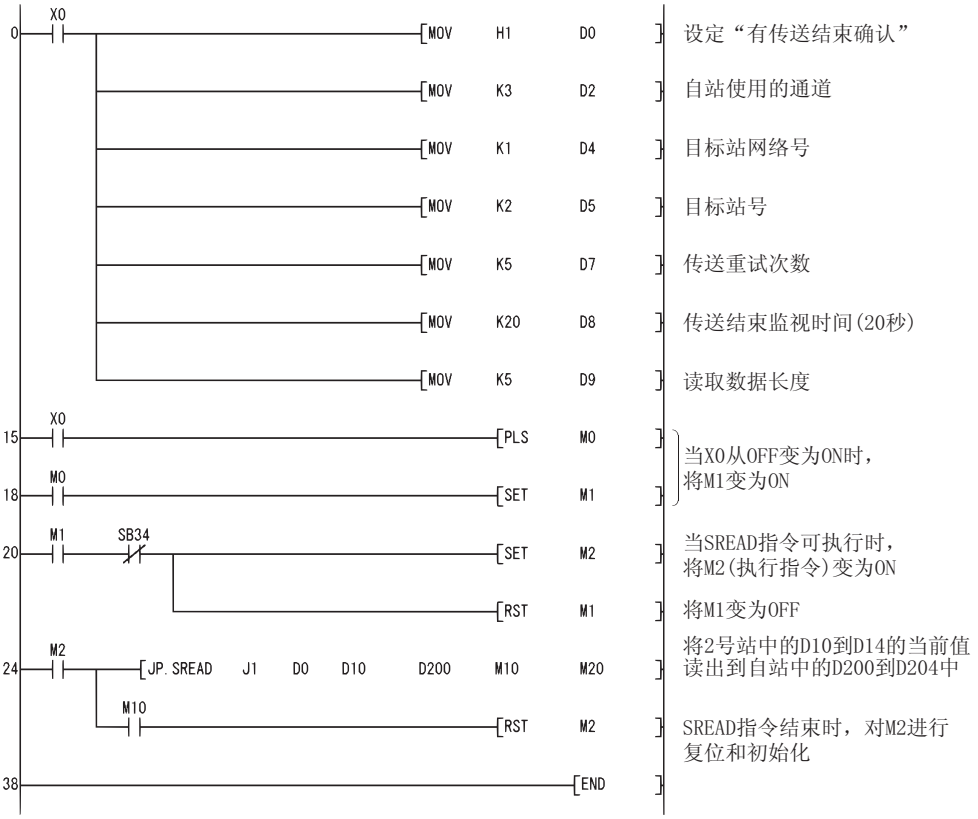
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码 : 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码 : 4100)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码 : 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序将连接到 MELSECNET/10 的 2 号站的 D10 ～ D14 的数据读取到自站的 200 ～ D204 中。

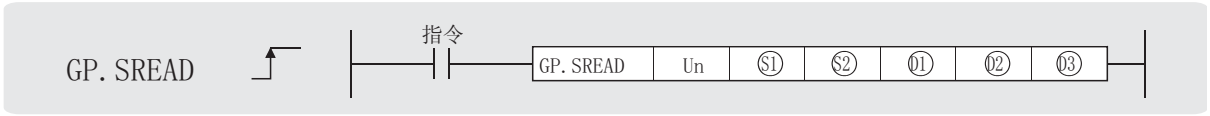
[梯形图模式]



[列表模式]

步	指令	软元件					
0	LD	X0					
1	MOV	H1	D0				
3	MOV	K3	D2				
5	MOV	K1	D4				
7	MOV	K2	D5				
9	MOV	K5	D7				
11	MOV	K20	D8				
13	MOV	K5	D9				
15	LD	X0					
16	PLS	M0					
18	LD	M0					
19	SET	M1					
20	LD	M1					
21	ANI	SB34					
22	SET	M2					
23	RST	M1					
24	LD	M2					
25	JP.SREAD	J1	D0	D10	D200	M10	M20
36	AND	M10					
37	RST	M2					
38	END						

2 对于 CC-Link、串行通信
软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 可以使用 CC-Link。



设定数据	内部软件元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○	--					
(S2)	--	△*2	○	--					
(D1)	--	△*2	○	--					
(D2)	△*1	△*2	○	--					
(D3)	△*1	△*2	○	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

◯设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。*3	BIN 16 位
(S1)	存储控制数据的自站的起始软元件号。	软元件名称
(S2)	存储读取数据的目标站的起始软元件。	
(D1)	存储读取数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的自站软元件。*4	位
(D3)	指令结束后将 ON 一个扫描的目标站软元件。*4	

*3 : 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
• QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 SREAD 指令才能被执行。
(SREAD 指令不能在连接到 CC-Link 或串行通信的 ACPU 中执行。)
2. 对于通过 SREAD 指令进行数据读取的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
（目标站 CPU 模块的读取目标的起始软元件号 (S2) ）+（读取点数 - 1）
≦ ((S2) 指定的最终软元件号（自站 CPU 模块））

控制数据

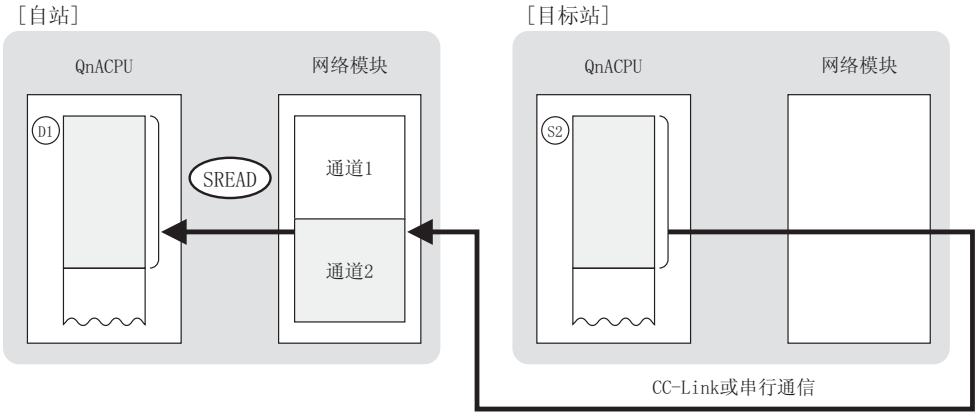
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0001H: 固定（有传送结束确认）	0001H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 : 没有错误（正常结束） 不是 0: 出错代码 *5	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 (CC-Link)	指定存储数据的目标站号。 0: 指定的主站 1 到 64: 指定的本地站	0 到 64	用户
	特殊功能模块站号 （串行通信模块）	指定存储数据的目标站号。 0 到 31: 站点号	0 到 31	用户
(S1) + 7	传送重试次数	如果通过 SREAD 指令进行的读取操作在 (S1)+8 指定的监控时间内未完成，设定传送重试次数。	0 到 15	用户
	重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定通过 SREAD 指令读取时的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0: 固定为 10 秒 （默认）	用户
(S1) + 9	读取数据长度	设定要读取的数据块数目。	1 到 480*6	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12	虚拟	禁止使用	--	--
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 : 关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）(新增调制解调器版)
- CC-Link 系统主站 / 自站模块用户手册（型号 AJ61QBT11/A1SJ61QBT11)(详细篇)

*6 : 当（由 (D1)）指定的软元件号)+(接收数据长度）超出了由 (D1) 指定的软元件的范围时，超出了指定范围的点数的数据存储在下一个软元件的起始部分。

功 能

(1) 将控制数据的主站 / 本地站号或特殊功能模块站号 (Ⓢ1+6) 所指定的站的 Ⓢ2 中指定的字软元件后面的数据，保存到自站的 Ⓓ1 指定的软元件后面的软元件中。
当目标站中软元件数据的读取操作已经完成时， Ⓓ2 指定的结束软元件将变为 ON。
在目标站中，当 Ⓢ2 指定的软元件数据传送已经完成时， Ⓓ3 指定的软元件将变为 ON。



(2) 对 CC-Link 或串行通信连接的站可以进行软元件数据的读取。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) SREAD 指令的执行状态以及其正常异常结束可以通过以下方式确认。

(a) 通讯指示标志*7 : SREAD 指令执行时变为 ON，在读取结束时的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (Ⓓ2) : 在通过 SREAD 指令执行的读取结束时的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

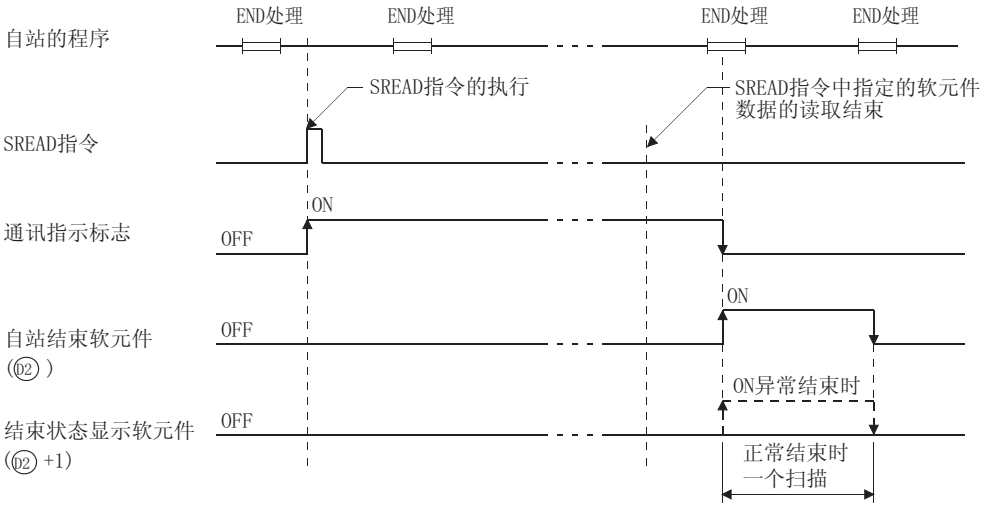
(c) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 SREAD 指令结束时的状态而 ON/OFF。
- 正常结束： 保持 OFF 状态不变
- 异常结束： 当 SREAD 指令结束时的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(d) 目标站的结束软元件 (Ⓓ3) : 当由 SREAD 指令指定的软元件的数据传送已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

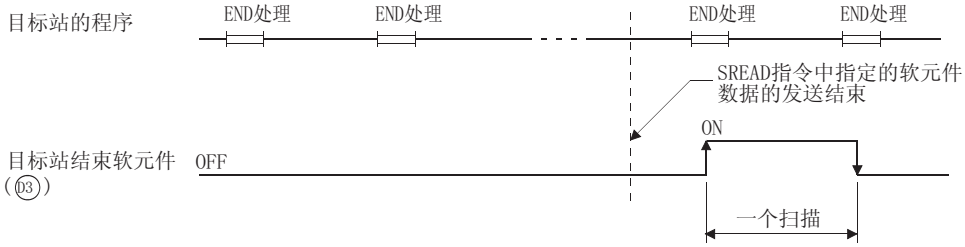
*7: 所用通道的通讯指示标志如下所示：

通道号	1	2
通讯指示标志号	SB30	SB32

[SREAD 指令执行时的自站的动作]



[SREAD 指令执行时的目标站的动作]



运算错误

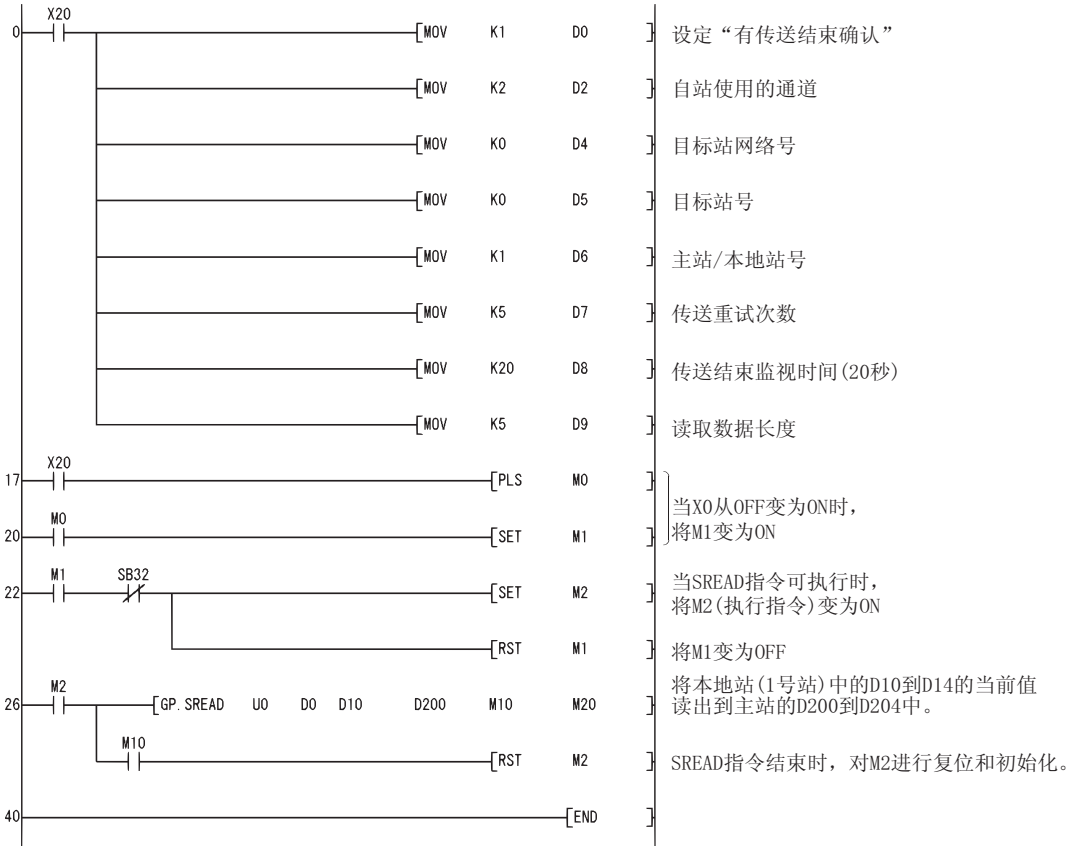
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4100)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，下列程序将连接到 CC-Link 的本地站 (1 号站) 的 D10 到 D14 的数据读取到主站的 D200 到 D204 中。

[梯形图模式]



[列表模式]

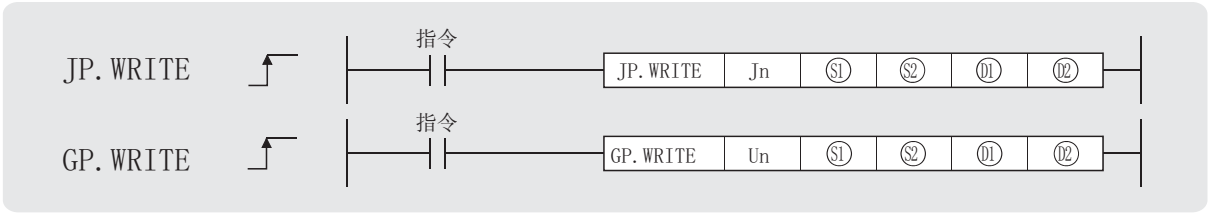
步	指令	软元件					
0	LD	X20					
1	MOV	K1	D0				
3	MOV	K2	D2				
5	MOV	K0	D4				
7	MOV	K0	D5				
9	MOV	K1	D6				
11	MOV	K5	D7				
13	MOV	K20	D8				
15	MOV	K5	D9				
17	LD	X20					
18	PLS	M0					
20	LD	M0					
21	SET	M1					
22	LD	M1					
23	ANI	SB32					
24	SET	M2					
25	RST	M1					
26	LD	M2					
27	GP.SREAD	U0	D0	D10	D200	M10	M20
38	AND	M10					
39	RST	M2					
40	END						

8.3.3 写入软元件数据到其它站（JP/GP.WRITE）

通用型 *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 WRITE 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*3	○				--		
(S2)	--	△*3	○				--		
(D1)	--	△*3	○				--		
(D2)	△*2	△*3	○				--		

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

◯设定数据

设定数据	含义	数据类型
Jn	自站网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储写入数据的自站的起始软元件。	
(D1)	存储写入数据的目标站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的软元件。*6	位

*4 ：自站网络号应按如下方式指定。
- 1 到 239：网络号
- 254：设定为其它站可以访问的有效模块中指定的网络。

*5 ：自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ：局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 WRITE 指令才能被执行。
   (WRITE 指令不能在连接到 MELSECNET/10 或以太网的 ACPU 中执行。)
2. 只有在仅由 QnACPU 构成的网络中，才可以在 WRITE 指令中将目标站号指定为“FFH(对象网络的所有站)”。
   在由 QnACPU 和 ACPU 混合构成的网络中，不能指定 “FFH”。
3. 对于通过 WRITE 指令进行数据写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
   (目标站 CPU 模块的写入目标的起始软元件号 ⑤2) + (写入的点数 - 1)
   ≦ (⑤2 指定的最终软元件号(自站 CPU 模块))

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	无传送结束确认：将位 0(b0) 设定成 0。 有传送结束确认：将位 0(b0) 设定为 1。	0000H*7 0001H 0080H 0081H	用户
	异常结束类型	设置异常结束时时钟数据的设定状态。 不设定时钟数据：设定位 7(7) 为 0。 设定时钟数据：设定位 7(b7) 为 1。 （附加在 (S1)+11 的后面）		
(S1) + 1	结束状态	存储指令结束时的状态。 0 ：没有错误（正常结束） 不是 0：出错代码*8	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*7	1 到 8	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	设定目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	设定目标站的站号。 （只有在执行类型为 0 时，才可以设定 81H 到 89H，FFH）	1 到 64：站点号 81H 到 89H： 组指定 FFH：目标网络 的全部站	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	当 (S1) 指定的执行类型为 1 时有效，并且当 (S1)+8 指定的监视时间内没有结束时，设定要进行的传送重试次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒 只有执行类型为 1 时有效。	用户
(S1) + 9	写入数据长度	设定要写入的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 （只在异常时设定）	存储时钟数据的有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 （只在异常时设定）	月和年（年份后 2 位）以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站点号，如果 (S1)+1 结束状态是“通道使用中”，则不进行存储。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站号，如果 (S1)+1 结束状态是“通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*7 ：关于传送结束的有 / 无、网络模块的通道的详细内容，请参照 8.1 节。

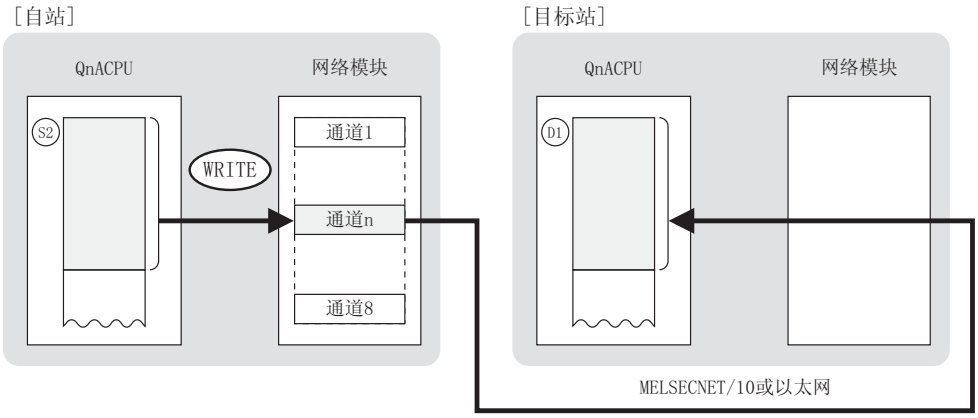
*8 ： 关于异常时的出错代码，请参照以下手册：

- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册

*9 ： 在 Jn 中设定了 254 时进行此指定。

功 能

(1) 将存储在自站的 (S2) 中指定的软元件后面的数据，存储到控制数据的目标站网络号 ((S1)+4) 和目标站站号 ((S1)+5) 所指定的 MELSECNET/10 或以太网的连接站的 (D1) 中指定的字软元件后面的软元件中。
当至目标站的软元件数据的写入操作完成时， (D2) 指定的结束软元件将变为 ON。



(2) 除了连接到自站网络中的站之外，对于连接到 MELSECNET/10 或以太网的指定网络号的站也可以执行软元件数据的写入操作。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) WRITE 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *10 : WRITE 指令执行时变为 ON，在写入操作结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 ((D2)) : 在 WRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

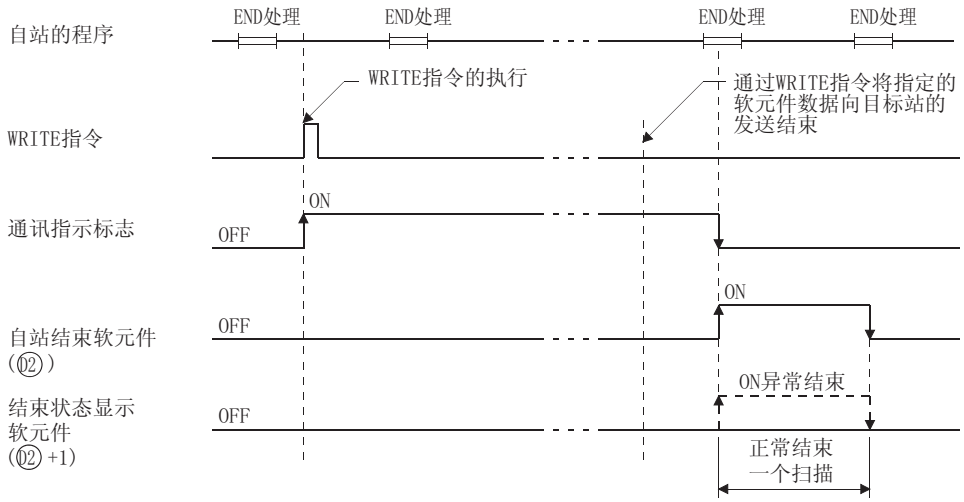
(c) 结束时的状态显示软元件 ((D2)+1) : 根据 WRITE 指令结束时的状态而 ON/OFF。

- 正常结束： 保持 OFF 状态不变
- 异常结束： 在 WRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*10: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[WRITE 指令执行时的自站的动作]



运算错误

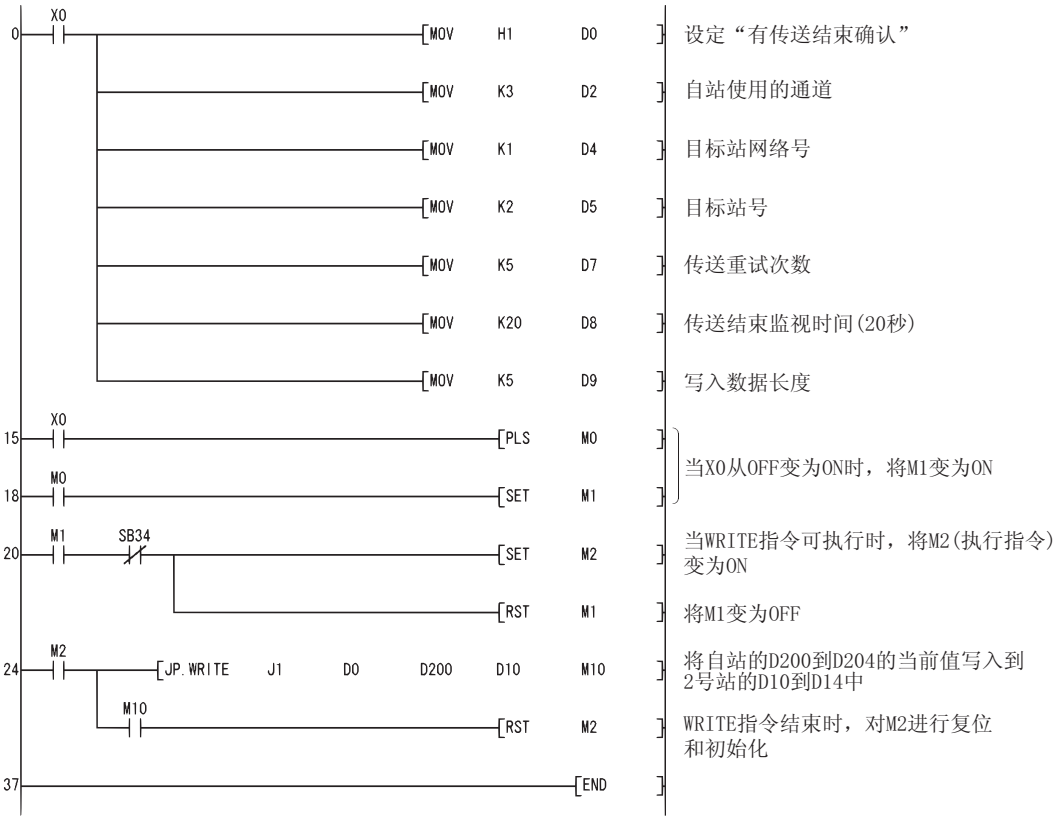
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X0 为 ON 时，下列程序将自站的 D200 到 D204 的数据写入到连接到 MELSECNET/10 的 2 号站的 D10 到 D14 中。

[梯形图模式]



[列表模式]

步	指令	软元件				
0	LD	X0				
1	MOV	H1	D0			
3	MOV	K3	D2			
5	MOV	K1	D4			
7	MOV	K2	D5			
9	MOV	K5	D7			
11	MOV	K20	D8			
13	MOV	K5	D9			
15	LD	X0				
16	PLS	M0				
18	LD	M0				
19	SET	M1				
20	LD	M1				
21	ANI	SB34				
22	SET	M2				
23	RST	M1				
24	LD	M2				
25	JP.WRITE	J1	D0	D200	D10	M10
35	AND	M10				
36	RST	M2				
37	END					

2 对于 CC-Link、串行通信
在软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 中可以使用 CC-Link。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○	--					
(S2)	--	△*2	○	--					
(D1)	--	△*2	○	--					
(D2)	△*1	△*2	○	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。*3	BIN 16 位
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储写入数据的目标站的起始软元件。	
(D1)	存储写入数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的软元件。*4	位

*3 : 自站的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时才可以执行 WRITE 指令。
(WRITE 指令不能在连接到 CC-Link 或串行通信的 ACPU 中执行。)
2. 对于通过 WRITE 指令进行数据写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
（目标站 CPU 模块的写入目标的起始软元件号 (S2)）+（写入的点数 - 1）
≦（(S2) 指定的最终软元件号（自站 CPU 模块））

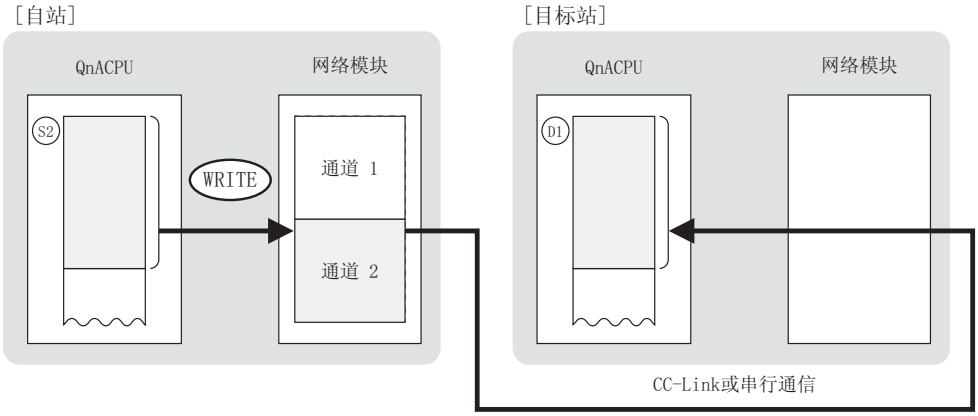
控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0000H ：无传送结束确认 0001H ：有传送结束确认	0000H*5 0001H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误（正常结束） 不是 0：出错代码 *6	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 ：使用 CH1 侧 2 ：使用 CH2 侧	1、2	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 (CC-Link)	指定存储数据的目标站号。 0：指定的主站　　1 到 64：指定的本地站	0 到 64	用户
	特殊功能模块站号 （串行通信模块）	指定存储数据的目标站号。 0 到 31：站号	0 到 31	用户
(S1) + 7	传送重试次数	当 (S1) 指定的执行类型为 1 时有效，当 (S1)+8 指定的监视时间内没有完成结束时，设定将要进行的传送重试次数。	0 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送在设定时间内未完成，将按照 (S1)+7 指定的次数执行传送重试。	1 到 32767 0：固定为 10 秒（默认） 只在执行类型为 1 时有效	用户
(S1) + 9	写入数据长度	设定要写入的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12	虚拟	禁止使用	--	--
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 ：关于有 / 无传送结束确认，请参照 8.1 节。
*6 ：关于异常时的出错代码，请参照以下手册：
- 串行通信模块用户手册（详细篇）（新增调制解调器版）
- CC-Link 系统主站 / 本地站模块用户手册（详细篇）（型号 AJ61QBT11/A1SJ61QBT11）

☆功 能

(1) 将从自站 (S2) 指定的软元件后面的数据，存储到控制数据的主站 / 本地站号或特殊功能模块站号 ((S1)+6) 指定的站的 (D1) 中指定的字软元件后面的字软元件中。
当向目标站的软元件数据写入结束时， (D2) 指定的结束软元件将变为 ON。



(2) 在用 CC-Link 或串行通信连接的站中可以进行软元件数据的写入。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) WRITE 指令的执行状态以及其正常或异常结束可以通过以下方式确认。

(d) 通讯指示标志 *7 : WRITE 指令执行时变为 ON，在写入结束的扫描的 END 处理时变为 OFF。

(e) 自站结束软元件 ((D2)) : 在 WRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

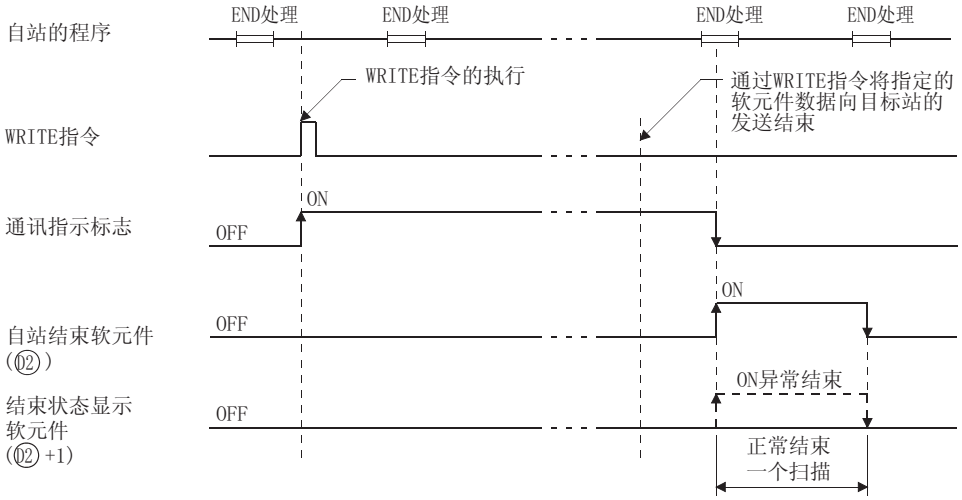
(f) 结束时的状态显示软元件 ((D2)+1) : 根据 WRITE 指令结束时的状态而 ON/OFF。

- 正常结束：保持 OFF 状态不变
- 异常结束：在 WRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*7：所用通道的通讯指示标志如下所示：

通道号	1	2
通讯指示标志号	SB30	SB32

[WRITE 指令执行时的自站的动作]



运算错误

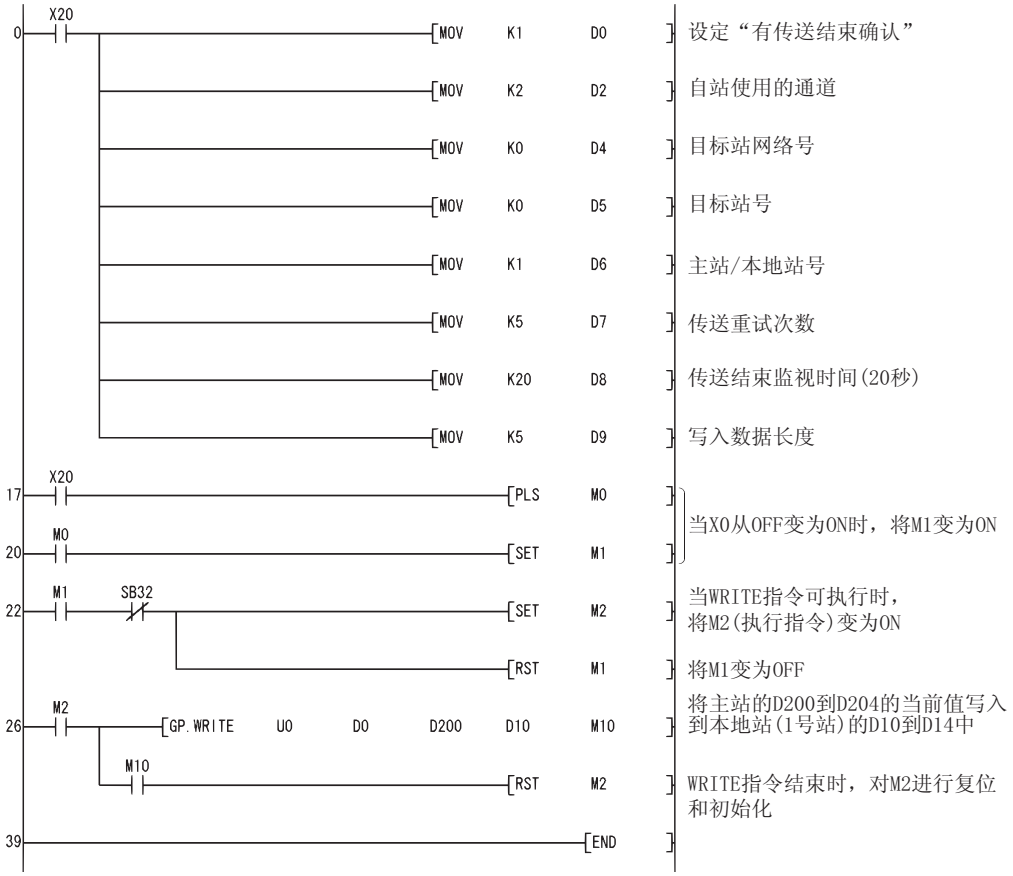
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X0 为 ON 时，下列程序将主站的 D200 到 D204 的数据写入到连接在 CC-Link 中的本地站（1 号站）的 D10 到 D14 中。

[梯形图模式]



[列表模式]

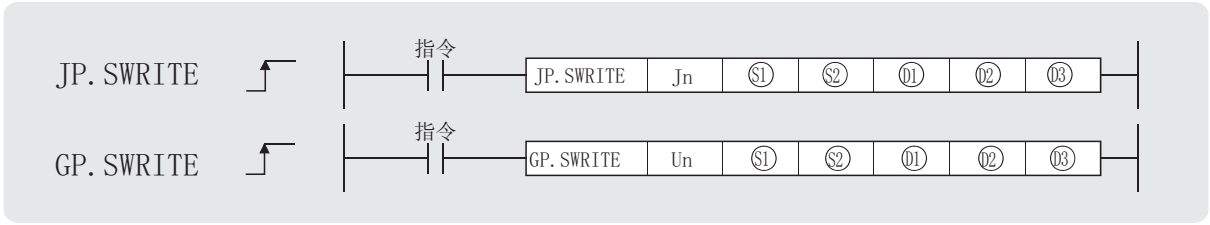
步	指令	软元件				
0	LD	X20				
1	MOV	K1	D0			
3	MOV	K2	D2			
5	MOV	K0	D4			
7	MOV	K0	D5			
9	MOV	K1	D6			
11	MOV	K5	D7			
13	MOV	K20	D8			
15	MOV	K5	D9			
17	LD	X20				
18	PLS	M0				
20	LD	M0				
21	SET	M1				
22	LD	M1				
23	ANI	SB32				
24	SET	M2				
25	RST	M1				
26	LD	M2				
27	GP.WRITE	U0	D0	D200	D10	M10
37	AND	M10				
38	RST	M2				
39	END					

8.3.4 将软元件数据写入到其它站（JP/GP.SWRITE）

通用型 UD *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 SWRITE 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*3	○	--					
(S2)	--	△*3	○	--					
(D1)	--	△*3	○	--					
(D2)	△*2	△*3	○	--					
(D3)	△*2	△*3	○	--					

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

设定数据

设定数据	含义	数据类型
Jn	自站网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储写入数据的目标站的起始软元件。	
(D1)	存储写入数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的自站软元件。*6	位
(D3)	指令结束后将 ON 一个扫描的目标站软元件。*6	

*4 ：自站网络号应按如下方式指定。
- 1 到 239：网络号
- 254：设定为其它站可以访问的有效模块中指定的网络。

*5 ： 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ： 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 SWRITE 指令才能被执行。
   (SWRITE 指令不能在连接到 MELSECNET/10 或以太网的 ACPU 中执行。)
2. 只有在仅由 QCPU/QnACPU 构成的网络中，才可以在 SWRITE 指令中将目标站号指定为 “FFH(在对象网络中的所有站)”。
   在由 QCPU/QnACPU 和 ACPU 混合构成的网络中，不能指定 “FEH”。
3. 对于通过 SWRITE 指令进行写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
   (Ⓢ2 目标站 CPU 模块的写入的目标的起始软元件号) + (写入的点数 - 1)
   ≦ (Ⓢ2 指定的最终软元件号(自站 CPU 模块))

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	无传送结束确认：将位 0(b0) 设定成 0。 有传送结束确认：将位 0(b0) 设定为 1。	0000H*7 0001H 0080H 0081H	用户
	异常结束类型	设置异常结束时的时钟数据的设定状态。 不设定时钟数据：将位 7(b7) 设定为 0。 设定时钟数据：将位 7(b7) 设定为 1。 （附加在 (S1)+11 的后面）		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误（正常结束） 不是 0：出错代码 *8	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*7	1 到 8	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	设定目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	设定目标站的站号。 只有在执行类型为 0 时，才可以设定 81H 到 89H，FFH)	1 到 64：站点号 81H 到 89H：组指定 FFH：对象网络中的全部站	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	当 (S1) 指定的执行类型为 1 时有效，并且当被 (S1)+8 指定的监视时间内没有结束时，设定将要进行的传送重试次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送操作在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒 只有执行类型为 1 时有效	用户
(S1) + 9	写入数据长度	设定要写入的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 （只在异常时设定）	存储时钟数据有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 （只在异常时设定）	月和年（年份后 2 位）以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站号。但如果 (S1)+1 结束状态是 “通道使用中”，则不进行存储。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站点号。但如果 (S1)+1 结束状态是 “通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*7 ：关于传送结束有 / 无、网络模块的通道的详细内容，请参照 8.1 节。

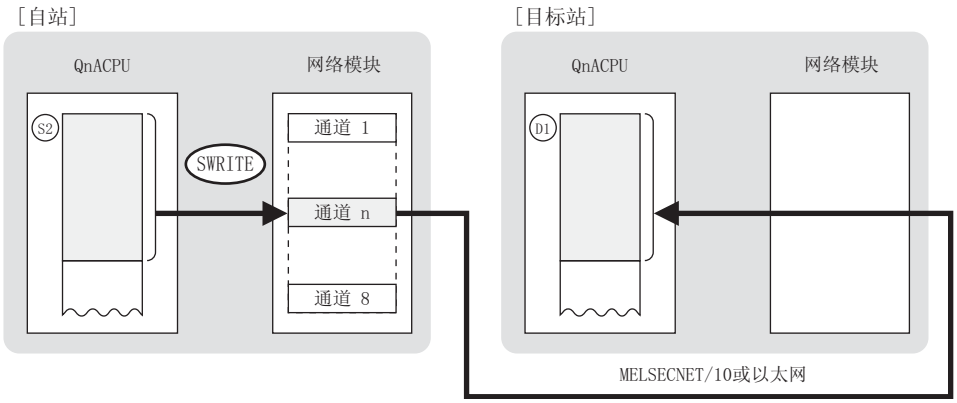
*8 ：关于异常时的出错代码，请参照以下手册：

- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册（详细篇）

*9 ：在 Jn 中设定了 254 时进行此指定。

☆功 能

(1) 将自站的 (S2) 中指定的软元件后面的数据，存储到控制数据的目标站网络号 ((S1)+4) 和目标站号 ((S1)+5) 中指定的 MELSECNET/10 或以太网模块连接站的 (D1) 中指定的字软元件后面的软元件中。
当目标站中 (S2) 指定的软元件数据的写入操作完成时，(D2) 中指定的结束软元件将变为 ON。同样，当目标站中 (S2) 指定的软元件数据的写入完成时，(D3) 指定的软元件将变为 ON。



(2) 除了连接在自站网络中的站之外，对于连接在 MELSECNET/10 或以太网的指定网络号的站也可以执行软元件数据的写入。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) SWRITE 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *10 : SWRITE 指令执行时变为 ON，在写入结束时的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 ((D2)) : 在 SWRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(c) 结束时的状态显示软元件 ((D2)+1) : 根据 SWRITE 指令结束时的状态而 ON/OFF。

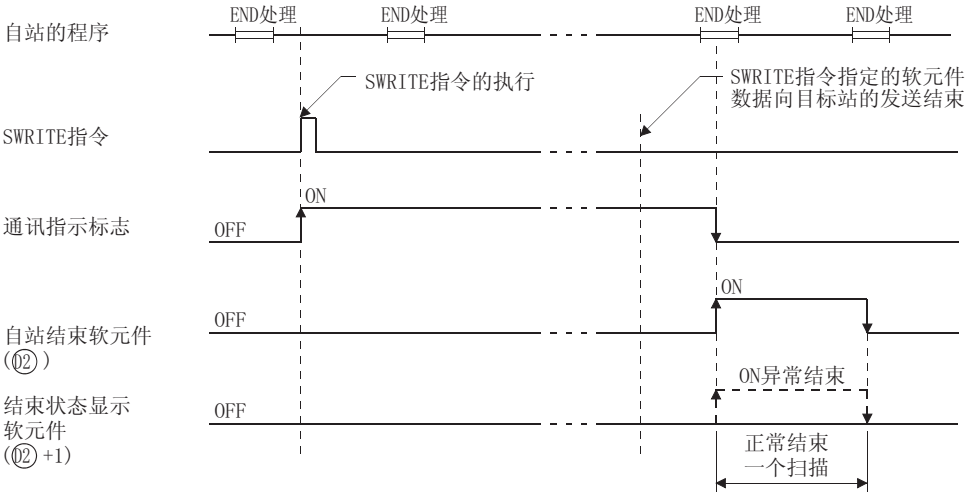
- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : 当 SWRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(d) 目标站的结束软元件 ((D3)) : 当 SWRITE 指令指定的软元件的数据传送已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

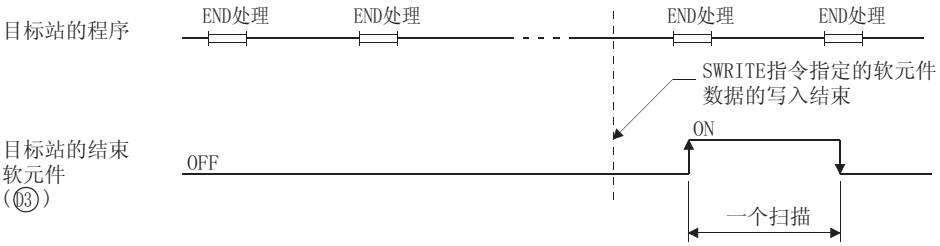
*10: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[SWRITE 指令执行时的自站的动作]



[SWRITE 指令执行时的目标站的动作]



运算错误

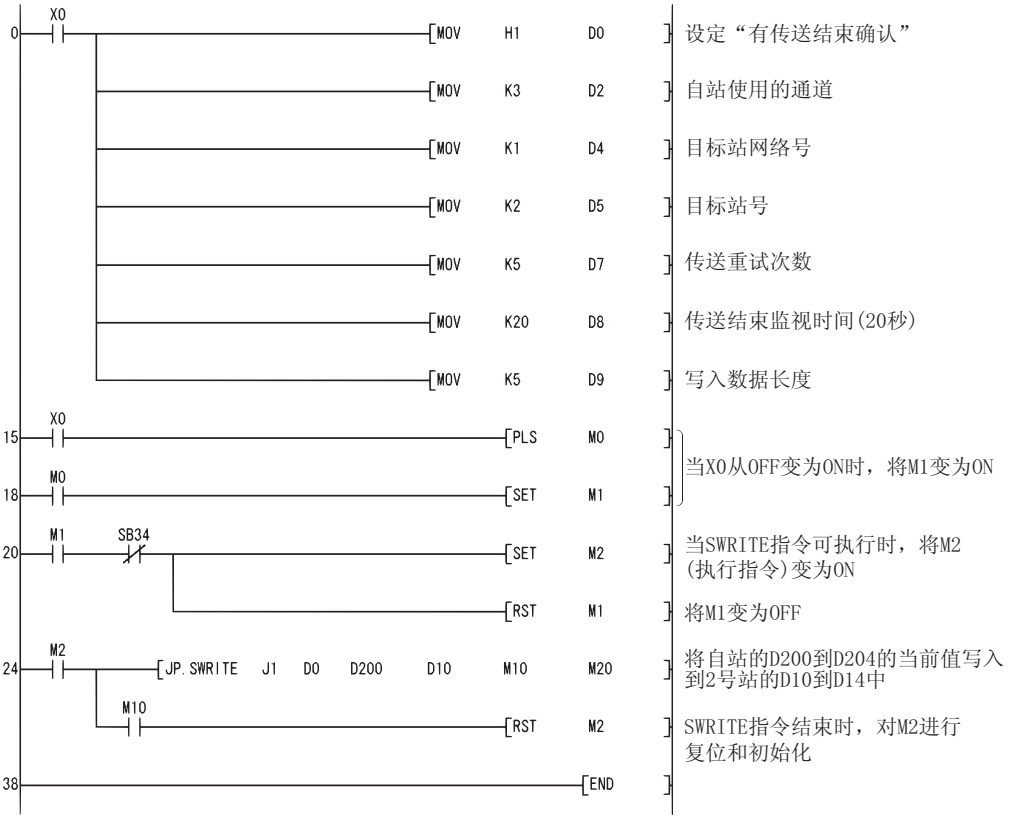
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X0 为 ON 时，下列程序将自站的 D200 到 D204 的数据写入到连接在 MELSECNET/10 中的 2 号站的 D10 到 D14 中。

[梯形图模式]



[列表模式]

步	指令	软元件					
0	LD	X0					
1	MOV	H1	D0				
3	MOV	K3	D2				
5	MOV	K1	D4				
7	MOV	K2	D5				
9	MOV	K5	D7				
11	MOV	K20	D8				
13	MOV	K5	D9				
15	LD	X0					
16	PLS	M0					
18	LD	M0					
19	SET	M1					
20	LD	M1					
21	ANI	SB34					
22	SET	M2					
23	RST	M1					
24	LD	M2					
25	JP.SWRITE		J1	D0	D200	D10	M10	M20
36	AND	M10					
37	RST	M2					
38	END						

2 对于 CC-Link、串行通信
在软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 中可以使用 CC-Link。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○	--					
(S2)	--	△*2	○	--					
(D1)	--	△*2	○	--					
(D2)	／*1	／*2	○	--					
(D3)	／*1	／*2	○	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

◯设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。*3	BIN 16 位
(S1)	存储控制数据的自站的起始软元件。	软元件名称
(S2)	存储写入数据的目标站的起始软元件。	
(D1)	存储写入数据的自站的起始软元件。	
(D2)	指令结束后将 ON 一个扫描的自站软元件。*4	位
(D3)	指令结束后将 ON 一个扫描的目标站软元件。*4	

*3 : 自站的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

要 点

1. 只有当目标站是 QCPU/QnACPU 时 SWRITE 指令才能被执行。
(SWRITE 指令不能在连接到 CC-Link 或串行通信的 ACPU 中执行。)
2. 对于通过 SWRITE 指令进行数据写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
（目标站 CPU 模块的写入目标的起始软元件号 (S2) ）+（写入的点数 - 1）
≦（ (S2) 指定的最终软元件号（自站 CPU 模块））

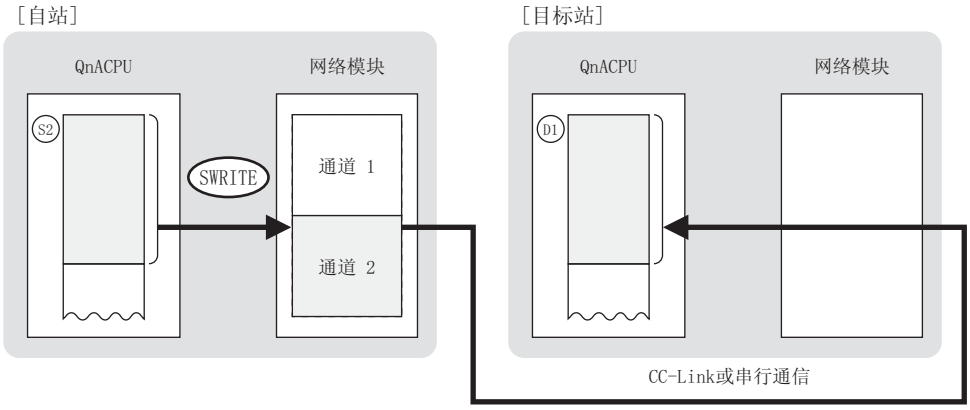
控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0000H : 无传送结束确认 0001H : 有传送结束确认	0000H*5 0001H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 : 没有错误（正常结束） 不是 0：出错代码 *6	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 (CC-Link)	指定存储数据的目标站号。 0：指定的主站 1 到 64：指定的本地站	0 到 64	用户
	特殊功能模块站号 （串行通信模块）	指定存储数据的目标站号。 0 到 31：站点号	0 到 31	
(S1) + 7	传送重试次数	当 (S1) 指定的执行类型为 1 时有效，当 (S1)+8 指定的监视时间内没有完成时，设定将要进行的传送重试次数。	0 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒（默认）只在执行类型为 1 时有效 1	用户
(S1) + 9	写入数据长度	设定要写入的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12	虚拟	禁止使用	--	--
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 : 关于有 / 无传送结束确认，请参照 8.1 节。
*6 : 关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）（新增调制解调器版）
- CC-Link 系统主站 / 自站模块用户手册（详细篇）（型号 AJ61QBT11/A1SJ61QBT11）

功 能

(1) 将自站的 (S2) 指定的软元件后面的数据，存储到控制数据的主站 / 本地站号或特殊功能模块站号 ((S1)+6) 指定的站的 (D1) 中指定的字软元件后面的字软元件中。
当向目标站的软元件数据的写入结束时， (D2) 指定的结束软元件将变为 ON。
同样，当目标站中 (S2) 指定的软元件数据的写入完成时， (D3) 指定的软元件将变为 ON。



(2) 对于 CC-Link 或串行通信的连接站可以进行软元件数据的写入。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) SWRITE 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(e) 通讯指示标志 *7 : SWRITE 指令执行时变为 ON，在写入结束的扫描的 END 处理时变为 OFF。

(f) 自站结束软元件 ((D2)) : 在 SWRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

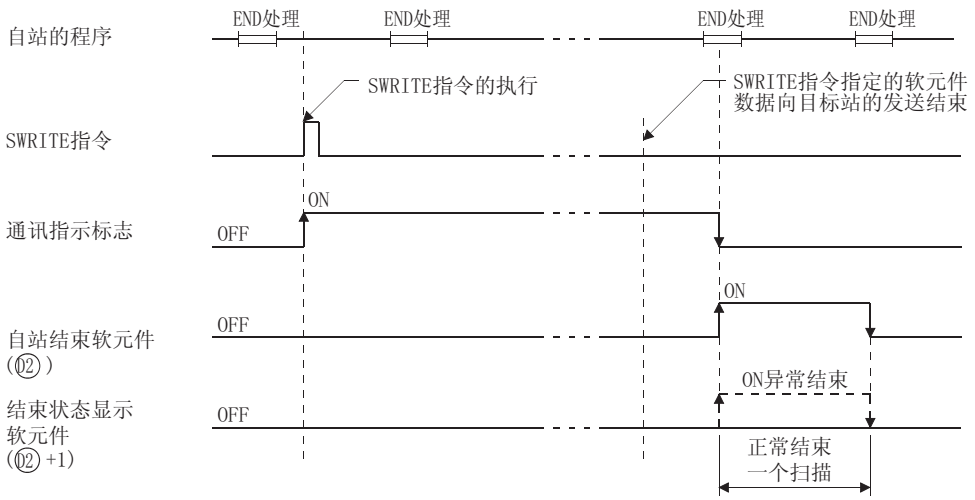
(g) 结束时的状态显示软元件 ((D2)+1) : 根据 SWRITE 指令结束时的状态而 ON/OFF。
- 正常结束： 保持 OFF 状态不变。
- 异常结束： 在 SWRITE 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(h) 目标站的结束软元件 ((D3)) : 当 SWRITE 指令指定的软元件的数据传送结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

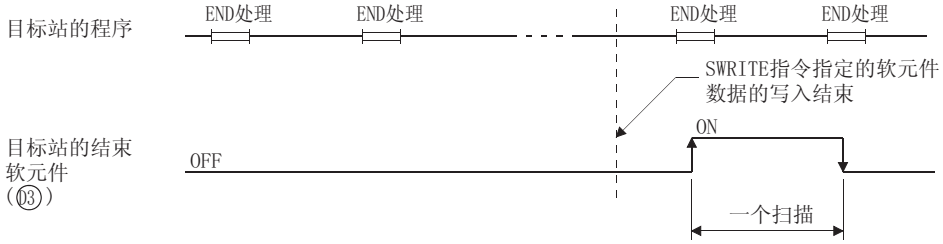
*7: 所用通道的通讯指示标志如下所示：

通道号	1	2
通讯指示标志号	SB30	SB32

[SWRITE 指令执行时的自站的动作]



[SWRITE 指令执行时的目标站的动作]



运算错误

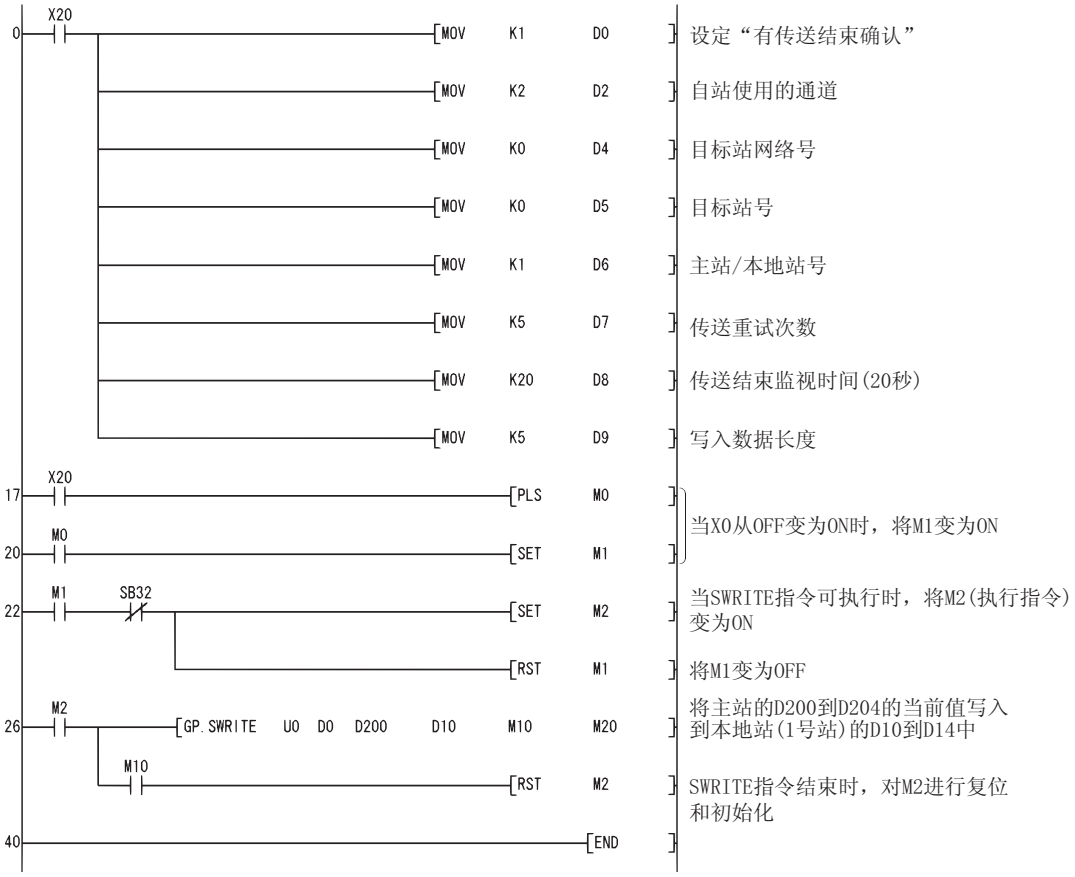
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X20 为 ON 时，下列程序写入主站数据 D200 至 D204 到连接到 CC-Link 的本地站 (1 号站) 的 D10 到 D14。

[梯形图模式]



[列表模式]

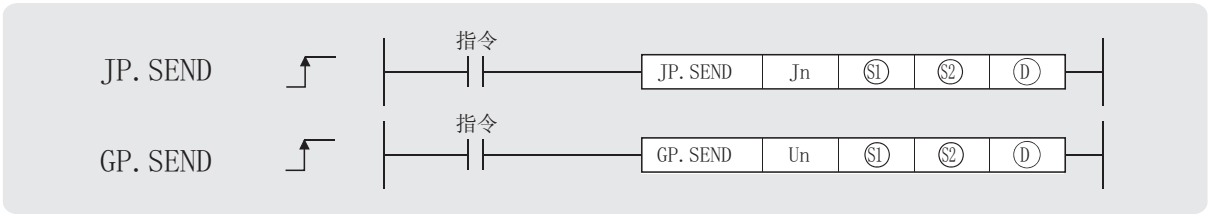
步	指令	软元件					
0	LD	X20					
1	MOV	K1	D0				
3	MOV	K2	D2				
5	MOV	K0	D4				
7	MOV	K0	D5				
9	MOV	K1	D6				
11	MOV	K5	D7				
13	MOV	K20	D8				
15	MOV	K5	D9				
17	LD	X20					
18	PLS	M0					
20	LD	M0					
21	SET	M1					
22	LD	M1					
23	ANI	SB32					
24	SET	M2					
25	RST	M1					
26	LD	M2					
27	GP.SWRITE		U0	D0	D200	D10	M10	M20
38	AND	M10					
39	RST	M2					
40	END						

8.3.5 传送数据到其它站（JP/GP.SEND）

通用型 *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 SEND 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*3	○	--					
(S2)	--	△*3	○	--					
(D)	△*2	△*3	○	--					

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Jn	自站的网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
(S1)	存储控制数据的软元件的起始号。	软元件名称
(S2)	存储将要发送的数据的软元件的起始号。	
(D)	在发送结束时， ON 一个扫描的软元件。*6	位

*4 ： 自站网络号应按如下方式指定。
- 1 到 239：网络号
- 254：设定为其它站可以访问的有效模块中指定的网络。

*5 ： 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ： 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	无传送结束确认：将位 0(b0) 设定成 0。 有传送结束确认：设定 1 到位 0(b0)。	0000H*7 0001H 0080H 0081H	用户
	异常结束类型	设置异常结束时的时钟数据的设定状态。 不设定时钟数据：将位 7(b7) 设定为 0。 设定时钟数据：将位 7(b7) 设定为 1。 （附加在 (S1)+11 的后面）		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误（正常结束） 不是 0：出错代码*8	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*7	1 到 8	用户
(S1) + 3	目标站使用的通道	为存储数据的目标站指定通道。	1 到 8	用户
(S1) + 4	目标站网络号	设定目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	指定目标站的站号。 （仅当执行类型为 0 时才可以设置 81H 到 89H，FFH）	1 到 64：站点号 81H 到 89H： 组指定 FFH：目标网络的全部站	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	当 (S1) 中指定的执行类型为 1 时有效，并且当 (S1)+8 指定的监视时间内没有结束时，设定将要进行的传送重试次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送操作在设定时间内未完成，将按照 (S1)+7 指定的次数传送重试。	1 到 32767 0：固定为 10 秒 只有执行类型为 1 时有效。1	用户
(S1) + 9	发送数据长度	设定要发送的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 （只在异常时设定）	存储时钟数据有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 （只在异常时设定）	月和年（年份后 2 位）以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站号。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站号。但如果 (S1)+1 结束状态是“通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*7 ：关于有 / 无传送结束确认、网络模块的通道的详细内容，请参照 8.1 节 .

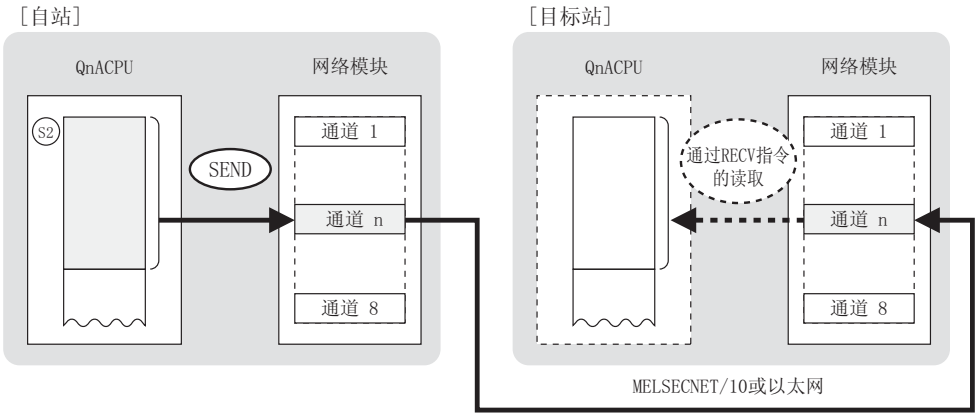
*8 ：关于异常时的出错代码，请参照以下手册：

- 出错代码为 F000H 到 FFFFH 时
  QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
  QnA 系列以太网接口模块用户手册（详细篇）
- 当出错代码为 4000H 到 4FFFH，请参照目标站的 QnACPU 用户手册（详细篇）中的“一般数据处理时返回到请求源的出错代码”。
- 当出错代码不为以上情况，则错误原因为目标站的 QnACPU。
  请参照目标站的 QnACPU 用户手册（详细篇）中的故障排除。

*9 ：在 Jn 中设定了 254 时进行此指定。

☆功 能

(1) 将自站的 (S2) 指定的软元件后面的数据，存储到控制数据的目标站网络号 ((S1)+4) 和目标站号 ((S1)+5) 中指定的 MELSECNET/10 或以太网模块的网络模块的指定通道中。
当至目标站的软元件数据的写入完成时， (D) 指定的结束软元件将变为 ON。



(2) 除了连接在自站网络中的站之外，对于 MELSECNET/10 或以太网的指定网络号的连接站也可以执行软元件数据的写入。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) SEND 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *10 : SEND 指令执行时变为 ON，在写入结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 ((D)) : 通过 SEND 指令执行的写入结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

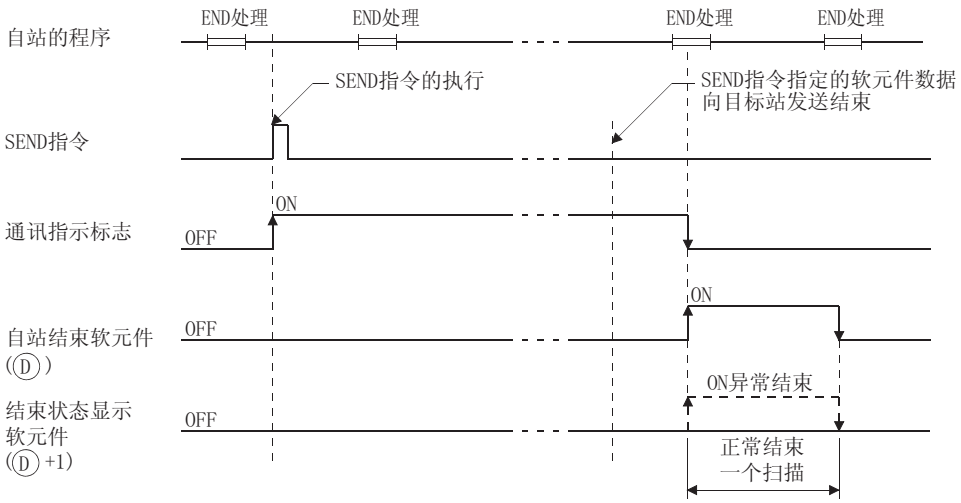
(c) 结束时的状态显示软元件 ((D)+1) : 根据 SEND 指令结束时的状态而 ON/OFF。

- 正常结束： 保持 OFF 状态不变。
- 异常结束： 当 SEND 指令已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*10: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[SEND 指令执行时的自站的动作]



运算错误

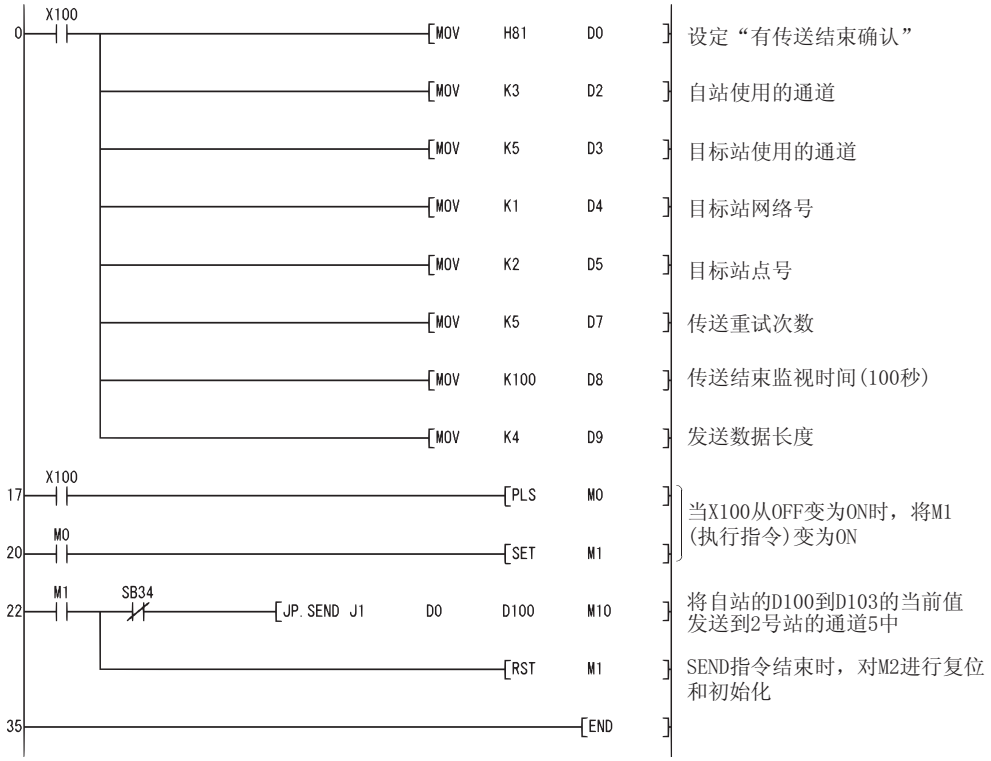
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X100 为 ON 时，下列程序将连接在 MELSECNET/10 中的自站的 D100 到 D103 的数据发送到 2 号站的通道 5 中。

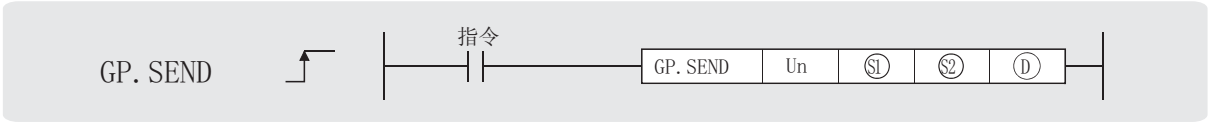
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X100			
1	MOV	H81	D0		
3	MOV	K3	D2		
5	MOV	K5	D3		
7	MOV	K1	D4		
9	MOV	K2	D5		
11	MOV	K5	D7		
13	MOV	K100	D8		
15	MOV	K4	D9		
17	LD	X100			
18	PLS	M0			
20	LD	M0			
21	SET	M1			
22	LD	M1			
23	MPS				
24	ANI	SB34			
25	JP.SEND	J1	D0	D100	M10
33	MPP				
34	RST	M1			
35	END				

2 对于 CC-Link、串行通信
在软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 中可以使用 CC-Link。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○	--					
(S2)	--	／*2	○	--					
(D)	／*1	／*2	○	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。 *3	BIN 16 位
(S1)	存储控制数据的软元件的起始号。	软元件名称
(S2)	存储发送数据的软元件的起始号。	
(D)	指令结束后，ON 一个扫描的软元件。*4	位

*3 ： 自站的起始 I/O 号的指定范围为 0 到 FEH。
*4 ： 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

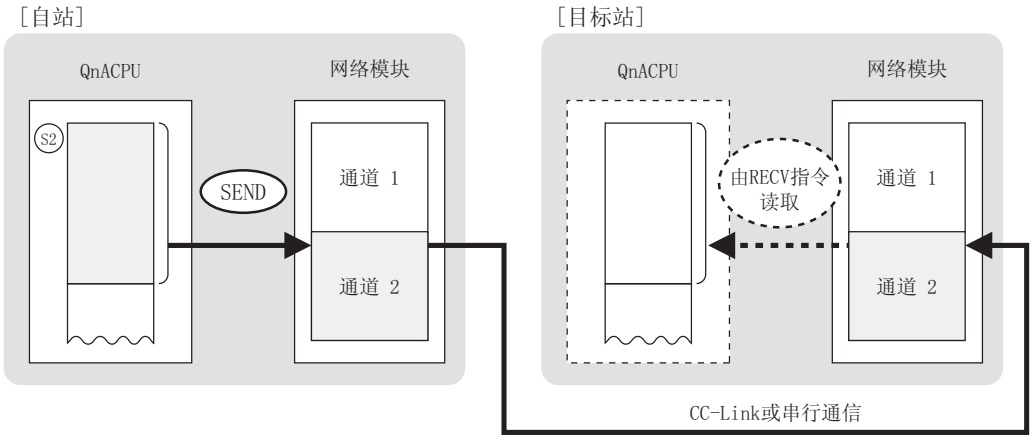
控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0000H : 无传送结束确认 0001H : 有传送结束确认	0000H*5 0001H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 : 没有错误（正常结束） 不是 0：出错代码 *6	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	用户
(S1) + 3	目标站使用的通道	为存储数据的目标站指定通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	用户
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 (CC-Link)	指定存储数据的目标站号。 0：指定的主站 1 到 64：指定的本地站	0 到 64	用户
	特殊功能模块站号 （串行通信模块）	指定存储数据的目标站号。 0 到 31：站点号	0 到 31	
(S1) + 7	传送重试次数	当 (S1) 指定的执行类型为 1 时有效，当 (S1)+8 指定的监视时间内没有完成结束时，设定将要进行的传送重试次数。	0 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果读取操作在设定时间内未完成，将按照 (S1)+7 指定的次数进行传送重试。	1 到 32767 0：固定为 10 秒（默认） 只在执行类型为 1 时有效	用户
(S1) + 9	发送数据长度	设定要发送的数据块数目。	1 到 480	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12				
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 : 关于有 / 无传送结束确认，请参照 8.1 节。
*6 : 关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）（新增调制解调器版）
- CC-Link 系统主站 / 自站模块用户手册（详细篇）（型号 AJ61QBT11/A1SJ61QBT11）

功 能

(1) 将自站的 ⓢ2 指定的软元件后面的数据，存储到控制数据的主站 / 本地站号或特殊功能模块站号 (ⓢ1+6) 中指定的模块通道中。
当向目标站的软元件数据的写入结束时，由 Ⓓ 指定的结束软元件将变为 ON。



(2) 对于 CC-Link 或串行通信的连接站可以进行软元件数据的写入。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手 (数据交换)，因此后面的链接专用指令将无法进行处理。

(4) SEND 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(d) 通讯指示标志 *7 : SEND 指令执行时变为 ON，在写入结束的扫描的 END 处理时变为 OFF。

(e) 自站结束软元件 (Ⓓ) : SEND 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

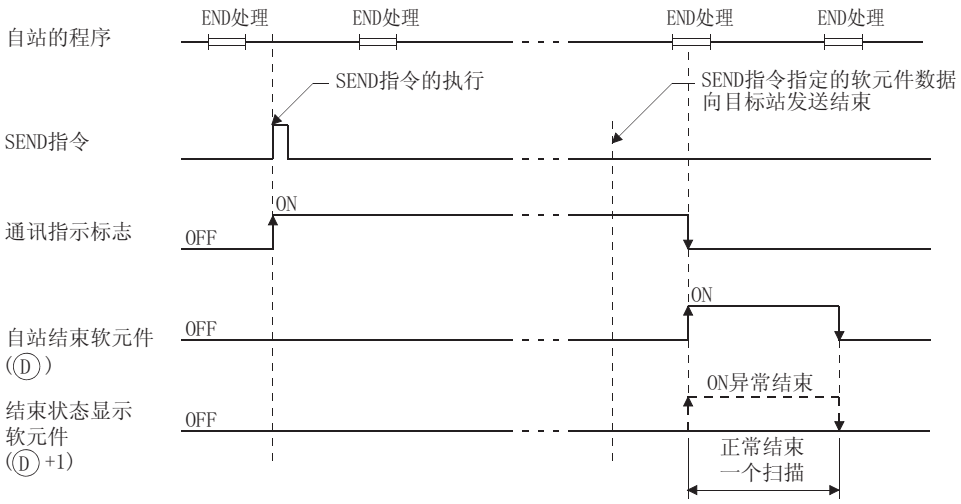
(f) 结束时的状态显示软元件 (Ⓓ+1) : 根据 SEND 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : 当 SEND 指令已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*7: 所用通道的通讯指示标志如下所示 :

通道号	1	2
通讯指示标志号	SB30	SB32

[SEND 指令执行时的自站的动作]



运算错误

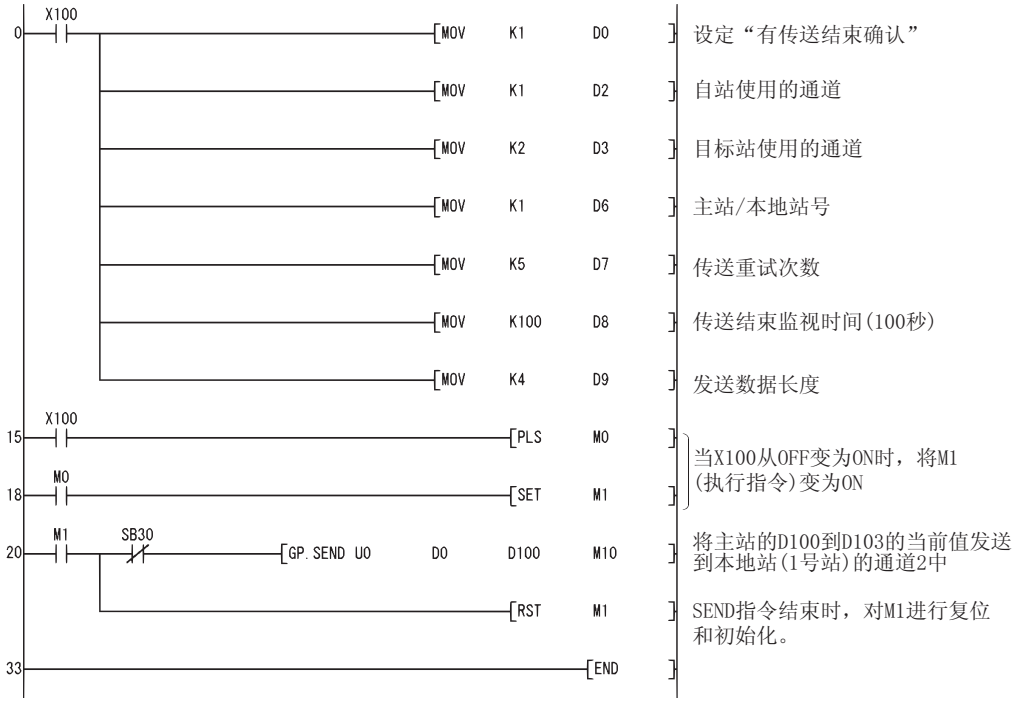
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X100 为 ON 时，下列程序将连接在 CC-Link 中的主站的 D100 到 D103 的数据发送到本地站 (1 号站) 的通道 2 中。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X100			
1	MOV	K1	D0		
3	MOV	K1	D2		
5	MOV	K2	D3		
7	MOV	K1	D6		
9	MOV	K5	D7		
11	MOV	K100	D8		
13	MOV	K4	D9		
15	LD	X100			
16	PLS	M0			
18	LD	M0			
19	SET	M1			
20	LD	M1			
21	MPS				
22	ANI	SB30			
23	GP.SEND	U0	D0	D100	M10
31	MPP				
32	RST	M1			
33	END				

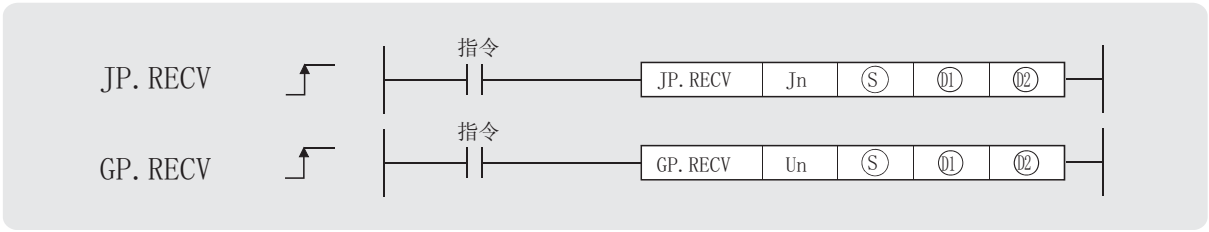
8.3.6 从其它站接收数据（JP/GP.RECV）

通用型 *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 RECV 指令的详细内容，请参照以下手册。

- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
ⓢ	--	△*3	○	--					
ⓓ1	--	△*3	○	--					
ⓓ2	△*2	△*3	○	--					

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

○设定数据

设定数据	含义	数据类型
Jn	自站网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
ⓢ	存储控制数据的软元件的起始号。	软元件名称
ⓓ1*7	存储发送数据的软元件的起始号。	
ⓓ2	在发送结束时，ON 一个扫描的软元件。*6	位

*4 ： 自站网络号应按如下方式指定。
- 1 到 239：网络号
- 254：设定为其它站可以访问的有效模块中指定的网络。

*5 ： 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ： 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

*7 ： 当接收数据软元件范围超过指定的限制，那么 4031H（软元件范围上限）将被设置到结束状态中。

控制数据

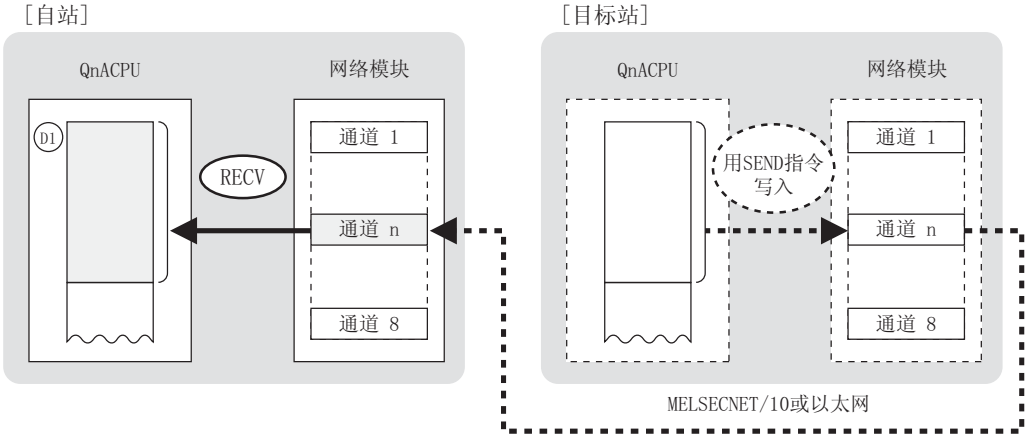
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	无数据时等待（在数据接收到之前等待一定的时间） ：将位 0(b0) 设定成 0。（固定）	0000H 0080H	用户
	异常结束类型	设定异常结束时有 / 无钟数据设定。 不设定时钟数据：将位 7(b7) 设定为 0。 设定时钟数据：位 7(b7) 设定为 1。 （附加在 (S1)+11 的后面）		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误（正常结束） 不是 0：出错代码 *8	--	系统
(S1) + 2	自站使用的通道	指定存储读取数据的通道。*9	1 到 8	用户
(S1) + 3	目标站使用的通道	存储发送站使用的通道。	1 到 8	系统
(S1) + 4	目标站网络号	存储发送站的网络号。	1 到 239	系统
(S1) + 5	目标站号	存储发送站的站号。	1 到 64	系统
(S1) + 6	虚拟	禁止使用	--	
(S1) + 7	虚拟	禁止使用	--	
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果接收在 WDT 时间内未完成，将会出现异常结束。	1 到 32767 0：固定为 10 秒 只有执行类型为 1 时有效。	用户
(S1) + 9	接收数据长度	设定要接收的数据块数目。	1 到 480	系统
(S1) + 10	虚拟	禁止使用	--	
(S1) + 11	时钟设定标记 （只在异常时设定）	存储时钟数据有效 / 无效状态。 0：时钟数据无效。 1：时钟数据有效。	--	系统
(S1) + 12	时钟数据 （只在异常时设定）	月和年（年份后 2 位）以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的站号。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站号。但如果 (S1)+1 结束状态是 “通道使用中”，则不进行存储。 1 到 64：站点号	--	系统

*8 ： 关于异常时的出错代码，请参照以下手册：
- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册（详细篇）

*9 ： 关于网络模块的通道的详细内容，请参照 8.1 节。

☆功 能

(1) 从连接在 MELSECNET/10 或以太网中的站中将以 SEND 指令发送的数据，通过自站的网络模块存储到 (D1) 指定的软元件以后。
当从网络模块软元件数据的读取完成时，(D2) 指定的结束软元件将变为 ON。



(2) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(3) RECV 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *10 : RECV 指令执行时变为 ON，在读取结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (D2) : 通过 RECV 指令执行的读取结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

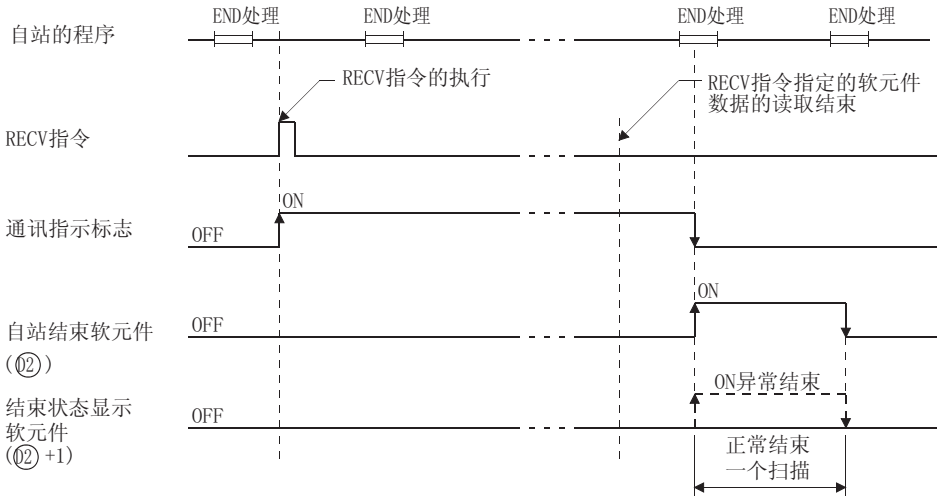
(c) 结束时的状态显示软元件 (D2)+1) : 根据 RECV 指令结束时的状态而 ON/OFF。

- 正常结束：保持 OFF 状态不变。
- 异常结束：当 RECV 指令结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*10: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[RECV 指令执行时的自站的动作]



运算错误

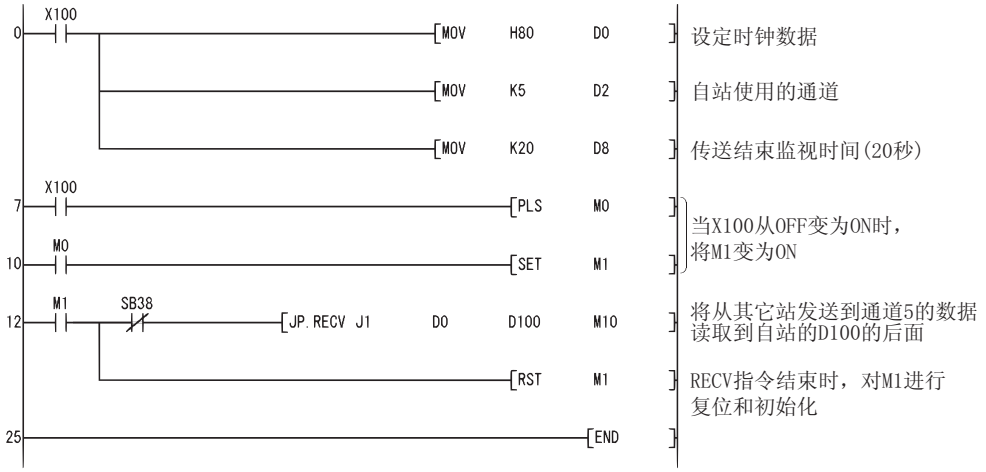
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X100 为 ON 时，下列程序将从连接在 MELSECNET/10 中的其它站发送至自站的通道 5 的数据读取到 D100 的后面。

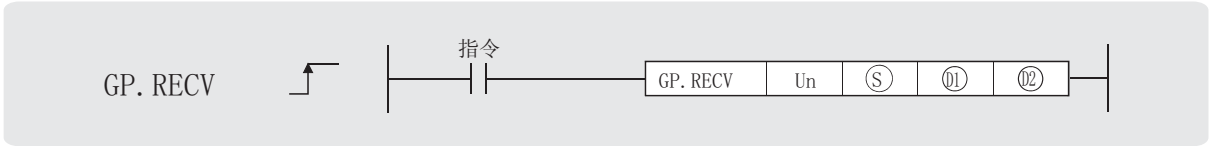
[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X100			
1	MOV	H80	D0		
3	MOV	K5	D2		
5	MOV	K20	D8		
7	LD	X100			
8	PLS	M0			
10	LD	M0			
11	SET	M1			
12	LD	M1			
13	MPS				
14	ANI	SB38			
15	JP.RECV	J1	D0	D100	M10
23	MPP				
24	RST	M1			
25	END				

2 对于 CC-Link、串行通信
在软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 中可以使用 CC-Link。



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S)	--	△*2	○	--					
(D1)	--	△*2	○	--					
(D2)	△*1	△*2	○	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

◯设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。*3	BIN 16 位
(S)	存储控制数据的软元件的起始号。	软元件名称
(D1)*5	存储接收数据的软元件的起始号。	
(D2)	接收结束后，ON 一个扫描的软元件。*4	位

*3 : 自站的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）
*5 : 当接收数据软元件范围超过指定的限制，则 4031H（软元件范围上限）将被设置到结束状态中。

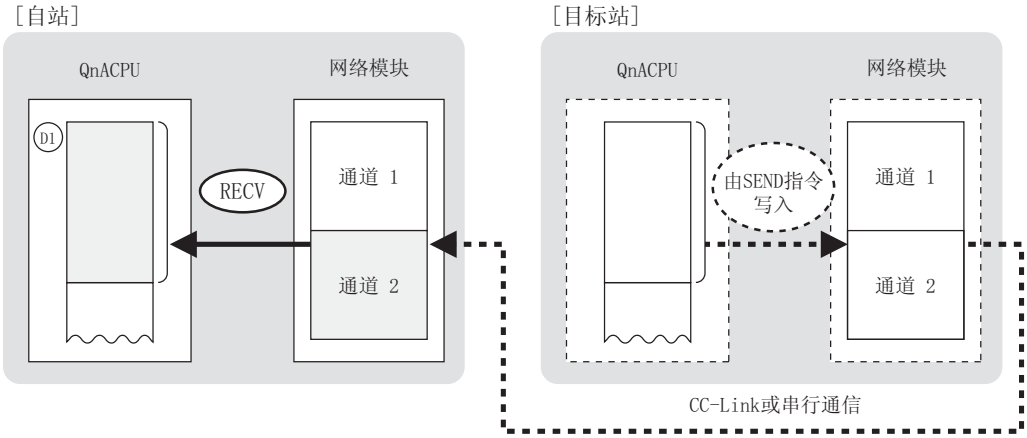
控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0000H: 固定（无传送结束确认）	0000H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 : 没有错误（正常结束） 不是 0: 出错代码 *6	--	系统
(S1) + 2	自站使用的通道	指定存储读取数据的通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	用户
(S1) + 3	目标站使用的通道	存储发送站使用的通道。 1 : 使用 CH1 侧 2 : 使用 CH2 侧	1、2	系统
(S1) + 4	目标站网络号	固定为 0	0	系统
(S1) + 5	目标站号	固定为 0	0	系统
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	虚拟	禁止使用	--	--
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果接收在 WDT 时间内未完成，将会出现异常结束。	1 到 32767 0: 固定为 10 秒 只在执行类型为 1 时有效	用户
(S1) + 9	接收数据长度	设定要接收的数据块数目。	1 到 480	系统
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12				
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*6 : 关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）(新增调制解调器版）
- CC-Link 系统主站 / 自站模块用户手册（详细篇）(型号 AJ61QBT11/A1SJ61QBT11)

功 能

(1) 将由 SEND 指令从连接在 CC-Link 或串行通信模块的站发送的数据，通过自站的网络模块存储到 (D1) 指定的软元件的后面。
当从网络模块的软元件数据的读取完成时，(D2) 指定的结束软元件将变为 ON。



(2) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(3) RECV 指令的执行状态以及其正常或异常结束可以通过以下方式确认。

(d) 通讯指示标志 *7 : RECV 指令执行时变为 ON，在读取结束的扫描的 END 处理时变为 OFF。

(e) 自站结束软元件 (D2) : 通过 RECV 指令执行的读取结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

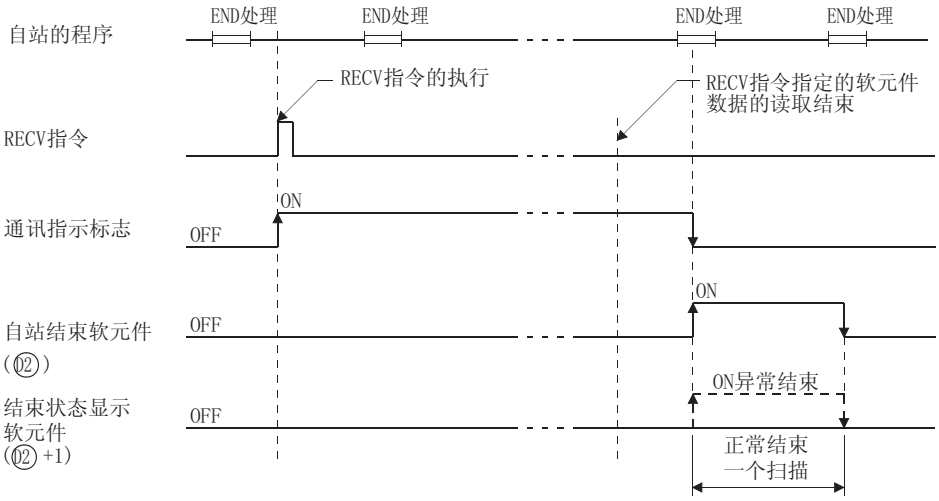
(f) 结束时的状态显示软元件 (D2+1) : 根据 RECV 指令结束时的状态而 ON/OFF。

- 正常结束: 保持 OFF 状态不变。
- 异常结束: 当 RECV 指令结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*7: 所用通道的通讯指示标志如下所示：

通道号	1	2
通讯指示标志号	SB30	SB32

[RECV 指令执行时的自站的动作]



运算错误

(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

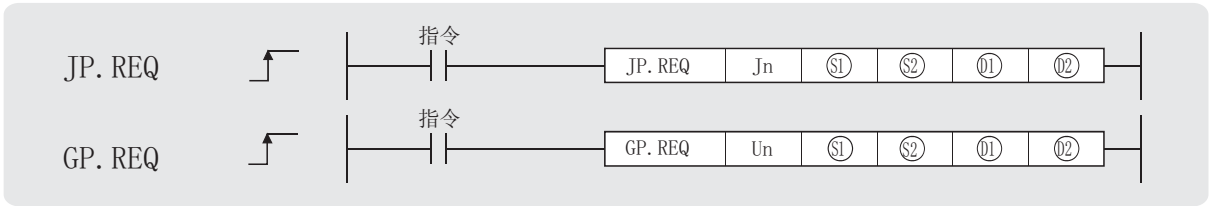
- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

8.3.7 来自其它站的瞬时请求（时钟数据的读 / 写、远程 RUN/STOP)(JP/GP.REQ)

通用型 UD *1　基本型 *1　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：关于 QCPU 可以使用的 REQ 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）

1 对于 MELSECNET/10、以太网



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*3	○	--					
(S2)	--	△*3	○	--					
(D1)	--	∧*3	○	--					
(D2)	∧*2	∧*3	○	--					

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。

设定数据

设定数据	含义	数据类型
Jn	自站的网络号。*4	BIN 16 位
Un	自站的网络模块的起始 I/O 号。*5	
(S1)	存储控制数据的软元件的起始号。	软元件名称
(S2)	存储请求数据的软元件的起始号。	
(D1)	存储响应数据的软元件的起始号。	
(D2)	在指令结束时，ON 一个扫描的软元件。*6	位

*4 ：自站网络号应按如下方式指定。
- 1 到 239：网络号
- 254：设定为其它站可以访问的有效模块中指定的网络。

*5 ：自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 ：局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

要 点

仅当目标站为 QCPU/QnACPU 站时，才可以执行 REQ 指令。
（在连接到 MELSECNET/10 或以太网的 ACPU 站中不能执行 REQ 指令。）

控制数据

软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	有传送结束确认：将位 0(b0) 和位 4(b4) 设定为 1(固定)。	0011H 0091H	用户
	异常结束类型	设置异常结束时的时钟数据的设定状态。 不设定时钟数据：将位 7(b7) 设定为 0。 设定时钟数据：将位 7(b7) 设定为 1。 (附加在 (S1)+11 的后面)		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误(正常结束) 不是 0：出错代码 *7	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。*8	1 到 8	用户
(S1) + 3	虚拟	禁止使用		
(S1) + 4	目标站网络号	设定目标站的网络号。	1 到 239 254*9	用户
(S1) + 5	目标站号	设定目标站的站号。 1 到 64：站点号 81H 到 89H：组指定(只允许写入时钟数据和远程 RUN/STOP)。 FFH ：目标网络号中的全部站(只允许写入时钟数据和远程 RUN/STOP)。	1 到 64： 站点号 81H 到 89H： 目标网络中 的全部站	用户
(S1) + 6	虚拟	禁止使用	--	--
(S1) + 7	传送重试次数	如果传送在 (S1)+8 指定的监控时间内未完成，设定重复次数。	1 到 15	用户
	传送重试结果	存储重复处理的次数。		系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送在设定时间内未完成，将会按 (S1)+7 指定的次数重新传送。	1 到 32767 0：固定为 10 秒	用户
(S1) + 9	请求数据长度	指定请求数据长度。 2 ：读时钟数据时。 6 ：写时钟数据时。 3 ：远程 STOP 时。 4 ：远程 RUN 时。	2、6、3、4	用户
(S1) + 10	响应数据长度	指定响应数据长度。 2 ：远程 RUN/STOP 时。 2 ：写时钟数据时。 6 ：读时钟数据时。	2、6	系统
(S1) + 11	时钟设定标记 (只在异常时设定)	存储时钟数据有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 (只在异常时设定)	月和年(年份后 2 位)以 BCD 编码存储。 高 8 位：月(01H 到 12H)、低 8 位：年(00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位：小时(00H 到 23H)、低 8 位：日(01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位：秒(00H 到 59H)、低 8 位：分钟(00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位：00H、低 8 位：日 of 星期(00H 到 06H)		
(S1) + 16	检测出异常的网络号	存储检测出异常的网络号。但如果 ((S1)+1) 的结束状态是“通道使用中”，则不进行存储。 1 到 239：网络号	--	系统
(S1) + 17	检测出异常的站号	存储检测出异常的站点号。但如果 ((S1)+1) 的结束状态是“通道使用中”，则不进行存储。 1 到 64 ：站点号	--	系统

*7 ：关于异常时的出错代码，请参照以下手册：
- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册

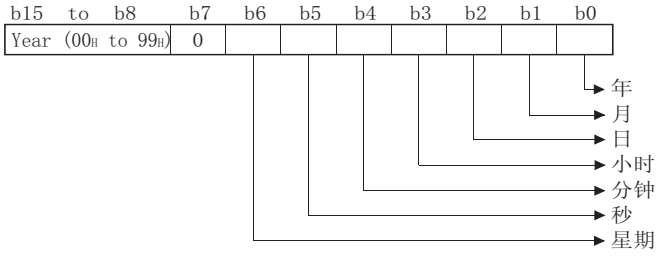
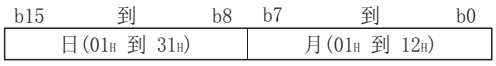
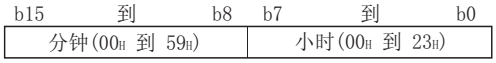
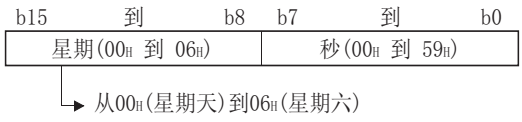
*8 ：关于网络模块的通道的详细内容，请参照 8.1 节。

*9 ：在 Jn 中设定了 254 时进行此指定。

[请求数据 / 响应数据]

(1)在对时钟数据的读 / 写操作中

(1) 请求数据

软元件	项目	含义	读时钟数据	写时钟数据
(S2)	请求类型	0001H: 读时钟数据 0011H: 写时钟数据 (由 (S1)+5 指定站号时) 0031H: 写时钟数据 (由 (S1)+5 指定全部站或组时)	○	○
(S2) + 1	子请求类型	0002H: 读时钟数据 0001H: 写时钟数据	○	○
(S2) + 2	更新格式被更新的年份	(1) 更新格式 (位 0 到 7) 指定写入 (S2)+2 的高位字节到(S2)+5 的项目。 0: 更新 1: 不更新 2) 更新年份 (位 8 到 15) 年 (年份的最后两位) 以 BCD 码形式存储。 	--	○
(S2) + 3	被更新的月份和日	月和日 (年份的最后两位) 以 BCD 码形式存储。 	--	○
(S2) + 4	被更新的小时和分钟	小时和分钟以 BCD 码形式存储。 	--	○
(S2) + 5	被更新的秒和星期	秒和星期以 BCD 码形式存储。 	--	○

(2) 响应数据

软元件	项目	含义	读时钟数据	写时钟数据
(D1)	请求类型	0081H: 读时钟数据 0091H: 写时钟数据（由 (S2) 指定 11H 时） 00B1H: 写时钟数据（由 (S2) 指定 31H 时）	○	○
(D1) + 1	子请求类型	0002H: 读时钟数据 0001H: 写时钟数据	○	○
(D1) + 2	被读取的月份和年份	月和年（年份后 2 位）以 BCD 编码存储。 b15 到 b8 | b7 到 b0 月(01H 到 12H) | 年(00H 到 99H)	○	--
(D1) + 3	被读取的小时和日	小时和日以 BCD 编码存储。 b15 到 b8 | b7 到 b0 小时(00H 到 23H) | 天(01H 到 31H)	○	--
(D1) + 4	被读取的秒和分钟	秒和分钟以 BCD 编码存储。 b15 到 b8 | b7 到 b0 秒(00H 到 59H) | 分钟(00H 到 59H)	○	--
(D1) + 5	被读取的星期	星期以 BCD 编码存储。 b15 到 b8 | b7 到 b0 00H | 星期(00H 到 06H) 从00H(星期天)到06H(星期六)	○	--

(2) 在远程 RUN/STOP 期间

(1) 请求数据

软元件	项目	含义	远程 RUN	远程 STOP
(S2)	请求类型	0010H ：（由 (S1) +5 指定站号时） 0030H ：（由 (S1) +5 指定全部站或组时）	○	○
(S2) + 1	子请求类型	0001H ：远程 RUN 0002H ：远程 STOP	○	○
(S2) + 2	模式	指定是否强制执行远程 RUN。 0001H ：不强制执行 0003H ：强制执行（远程 STOP 时的设定） 强制执行是指，对执行了远程 STOP 的站不能执行远程 RUN 时，通过其它站强制执行远程 RUN 操作。	○	○
(S2) + 3	清除模式	指定 QnACPU 被执行了远程 RUN 时的软元件内存的状态。 0000H ：不清除（远程 STOP 时的设定） 0001H ：清除（除锁存范围外 0002H ：清除（包括锁存范围）	○	--

(2) 响应数据

软元件	项目	含义	远程 RUN	远程 STOP
(D1)	请求类型	0090H ：（由 (S2) 指定 10H 时） 00B0H ：（由 (S2) 指定 30H 时）	○	○
(D1) + 1	子请求类型	0001H ：远程 RUN 0002H ：远程 STOP	○	○

要 点

1. 当目标站的 QnACPU 的 RUN/STOP 键开关设为 RUN 时，远程 RUN/STOP 功能被启动。

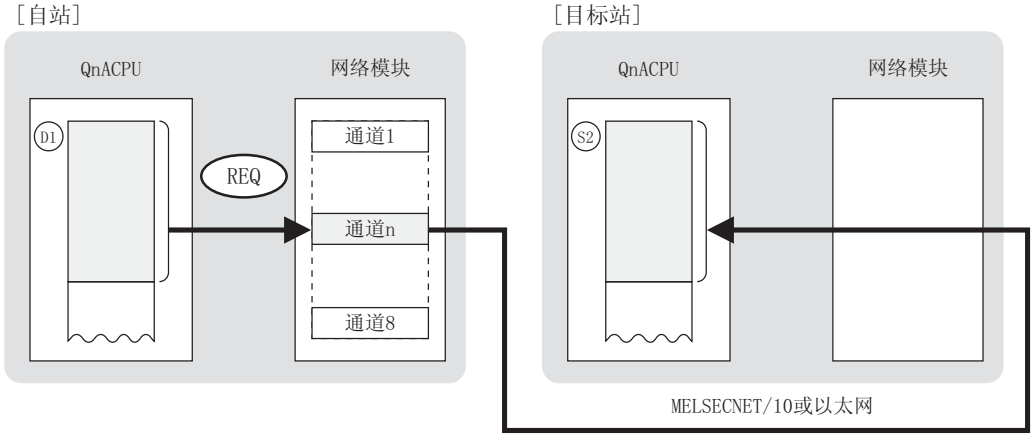
2. 当目标站的 QnACPU 的系统保护功能开启时（系统保护开关 SW5 状态为 ON），不能进行远程 RUN/STOP 操作。

3. 当目标站已经被另一个站进行了远程 STOP/PAUSE 时，如果 (S2)+2 的模式为“（不强制执行（0001H）”，则 RUN 操作不能被强制执行。

4. 如果进行了远程 RUN/STOP 的目标站的 QnACPU 被复位，则远程 RUN/STOP 信息将丢失。

功 能

(1) 对控制数据的目标站网络号（(S1) + 4）和目标站号（(S1) + 5）中指定的 MELSECNET/10 或以太网连接站的 (S2) 中指定的软元件之后的区域中存储的请求数据进行发送，以执行服务请求。当来自目标站的软元件数据请求结束时，(D2) 指定的结束软元件将变为 ON。



(2) 对于除自站网络连接站以外，对 MELSECNET/10 或以太网的指定网络号的连接站也可以进行软元件数据的发送。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) REQ 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 通讯指示标志 *10 : REQ 指令执行时变为 ON，在传送结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (D2) : 通过 REQ 指令执行的传送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

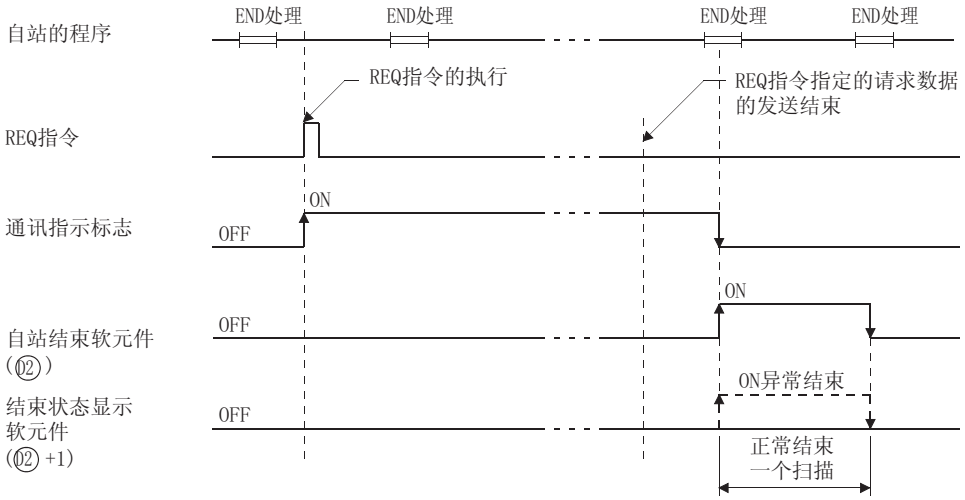
(c) 结束时的状态显示软元件 (Ⓓ2+1)：根据 REQ 指令结束时的状态而 ON/OFF。

- 正常结束： 保持 OFF 状态不变。
- 错误结束： 在 REQ 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*10: 所用通道的通讯指示标志如下所示：

通道号	1	2	3	4	5	6	7	8
通讯指示标志号	SB30	SB32	SB34	SB36	SB38	SB3A	SB3C	SB3E

[REQ 指令执行时的自站的动作]



运算错误

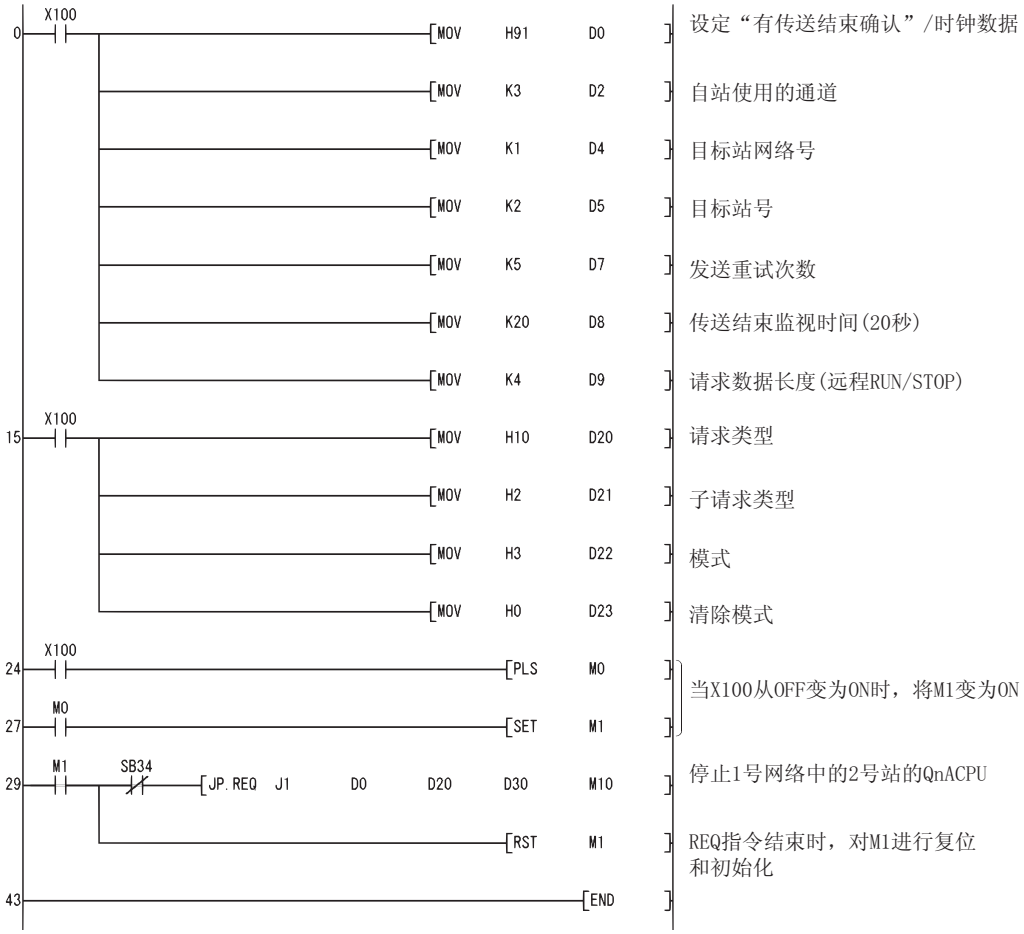
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X100 为 ON 时，下列程序对连接在 MELSECNET/10 中的 1 号网络 2 号站的 QnACPU 执行“STOP”操作。

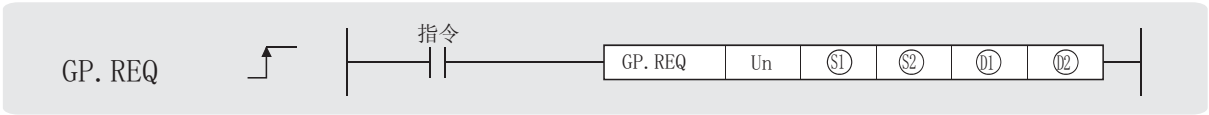
[梯形图模式]



[列表模式]

步	指令	软元件				
0	LD	X100				
1	MOV	H91	D0			
3	MOV	K3	D2			
5	MOV	K1	D4			
7	MOV	K2	D5			
9	MOV	K5	D7			
11	MOV	K20	D8			
13	MOV	K4	D9			
15	LD	X100				
16	MOV	H10	D20			
18	MOV	H2	D21			
20	MOV	H3	D22			
22	MOV	H0	D23			
24	LD	X100				
25	PLS	M0				
27	LD	M0				
28	SET	M1				
29	LD	M1				
30	MPS					
31	ANI	SB34				
32	JP.REQ	J1	D0	D20	D30	M10
41	MPP					
42	RST	M1				
43	END					

2 对于 CC-Link、串行通信
软件版本 J 或更新版本的主模块 (A(1S)J61QBT11) 可以使用 CC-Link。.



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	△*2	○				--		
(S2)	--	△*2	○				--		
(D1)	--	△*2	○				--		
(D2)	∧*1	∧*2	○				--		

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。

设定数据

设定数据	含义	数据类型
Un	自站的起始 I/O 号。*3	BIN 16 位
(S1)	存储控制数据的软元件的起始号。	软元件名称
(S2)	存储请求数据的软元件的起始号。	
(D1)	存储响应数据的软元件的起始号。	
(D2)	指令结束后，ON 一个扫描的软元件。*4	位

*3 : 自站的起始 I/O 号的指定范围为 0 到 FEH。
*4 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
• QnACPU 编程手册（特殊功能模块篇）

要 点

只有当目标站是 QCPU/QnACPU 时才可以执行 REQ 指令。
(REQ 指令不能在连接到 CC-Link 或串行通信的 ACPU 中执行。)

控制数据

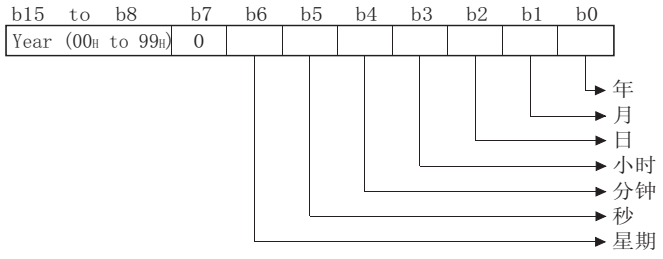
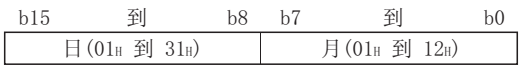
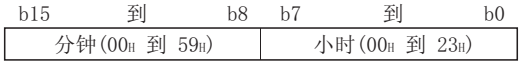
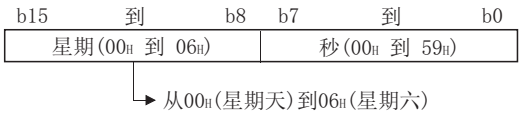
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	0011H: 固定（有传送结束确认））	0011H	用户
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误（正常结束） 不是 0 ：出错代码 *5	--	系统
(S1) + 2	自站使用的通道	指定自站使用的通道。 1 ：使用 CH1 侧 2 ：使用 CH2 侧	1、2	用户
(S1) + 3	虚拟	禁止使用	--	--
(S1) + 4	目标站网络号	固定为 0	0	用户
(S1) + 5	目标站号	固定为 0	0	用户
(S1) + 6	主站 / 本地站号 CC-Link	指定存储数据的目标站号。 0 ：指定的主站 1 到 64：指定的本地站	0 到 64	用户
	特殊功能模块站号（串行通信模块）	指定存储数据的目标站号。 0 到 31 ：站点号	0 到 31	用户
(S1) + 7	传送重试次数	如果传送在 (S1)+8 指定的监控时间内未完成，设定重复次数。	0 到 15	用户
	传送重试结果	存储重复处理的次数。	--	系统
(S1) + 8	传送结束监视时间	以秒为单位设定指令结束的监视时间。 如果传送在设定时间内未完成，将会按 (S1)+7 指定的次数重新传送。	1 到 32767 0：固定为 10 秒（默认）	用户
(S1) + 9	请求数据长度	指定请求数据长度。 2 ：读时钟数据时。 3 ：远程 STOP 时。 4 ：远程 RUN 时。 6 ：写时钟数据时。	2、6、3、4	用户
(S1) + 10	响应数据长度	指定响应数据长度。 2 ：远程 RUN/STOP 时。 2 ：写时钟数据时。 6 ：读时钟数据时。	2、6	系统
(S1) + 11	虚拟	禁止使用	--	--
(S1) + 12	虚拟	禁止使用	--	--
(S1) + 13				
(S1) + 14				
(S1) + 15				
(S1) + 16	检测出异常的网络号	固定为 0	--	系统
(S1) + 17	检测出异常的站号	固定为 0	--	系统

*5 ：关于异常时的出错代码，请参照以下手册。
- 串行通信模块用户手册（详细篇）（新增调制解调器版）
- CC-Link 系统主站 / 自站模块用户手册（详细篇（型号 AJ61QBT11/A1SJ61QBT11）

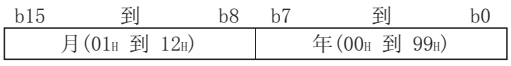
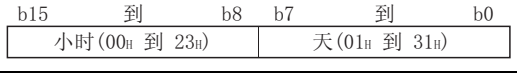
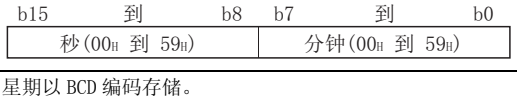
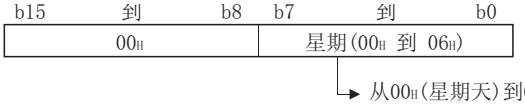
[请求数据 / 响应数据]

(1) 时钟数据的读 / 写时

(1) 请求数据

软元件	项目	含义	读时钟数据	写时钟数据
(S2)	请求类型	0001H : 读时钟数据 0011H : 写时钟数据	○	○
(S2) + 1	子请求类型	0002H : 读时钟数据 0001H : 写时钟数据	○	○
(S2) + 2	更新格式被更新的年份	(1) 更新格式 (位 0 到 7) 指定写入 (S2)+2 的高位字节 (S2)+5 的项目。 0: 更新 1: 不更新 2) 更新年份 (位 8 到 15) 年 (年份的最后两位) 以 BCD 码形式存储。 	--	○
(S2) + 3	被更新的月份和日	月和日 (年份的最后两位) 以 BCD 码形式存储。 	--	○
(S2) + 4	被更新的小时和分钟	小时和分钟以 BCD 码形式存储。 	--	○
(S2) + 5	被更新的秒和星期	秒和星期以 BCD 码形式存储。 	--	○

(2) 响应数据

软元件	项目	含义	读时钟数据	写时钟数据
(D1)	请求类型	0081H: 读时钟数据 0091H: 写时钟数据	○	○
(D1) + 1	子请求类型	0002H: 读时钟数据 0001H: 写时钟数据	○	○
(D1) + 2	被读取的月份和年份	月和年 (年份后 2 位) 以 BCD 编码存储。 	○	--
(D1) + 3	被读取的小时和日	小时和日以 BCD 编码存储。 	○	--
(D1) + 4	被读取的秒和分钟	秒和分钟以 BCD 编码存储。 	○	--
(D1) + 5	被读取的星期	星期以 BCD 编码存储。 	○	--

(2) 远程 RUN/STOP 时

(1) 请求数据

软元件	项目	含义	远程 RUN	远程 STOP
(S2)	请求类型	0010H	○	○
(S2) + 1	子请求类型	0001H : 远程 RUN 0002H : 远程 STOP	○	○
(S2) + 2	模式	指定是否强制执行远程 RUN 操作。 0001H : 不强制执行 0003H : 强制执行（远程 STOP 时的设定） 强制执行是指，对执行了远程 STOP 的站不能执行远程 RUN 时，通过其它站强制执行远程 RUN 操作。	○	○
(S2) + 3	清除模式	指定对 QnACPU 进行了远程 RUN 时的软元件内存的状态。 0000H : 不清除（远程 STOP 时的设定） 0001H : 清除（除锁存范围外） 0002H : 清除（包括锁存范围）	○	--

(2) 响应数据

软元件	项目	含义	远程 RUN	远程 STOP
(D1)	请求类型	0090H	○	○
(D1) + 1	子请求类型	0001H : 远程 RUN 0002H : 远程 STOP	○	○

要 点

1. 当目标站的 QnACPU 的 RUN/STOP 键开关设为 RUN 时，远程 RUN/STOP 功能有效。

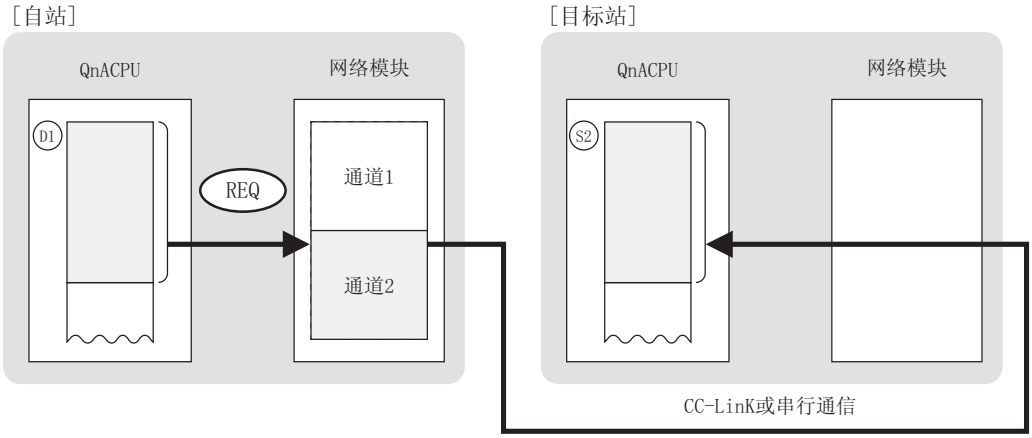
2. 当目标站的 QnACPU 的系统保护功能开启时（系统保护开关 SW5 状态为 ON），不能进行远程 RUN/STOP 操作。

3. 当目标站已经被另一个站进行了远程 STOP/PAUSE 时，如果 (S2)+2 的模式为“（不强制执行 (0001H)”，则 RUN 操作不能被强制执行。

4. 如果进行了远程 RUN/STOP 的目标站的 QnACPU 被复位，则远程 RUN/STOP 的信息将丢失。

☆功 能

(1) 对控制数据的主站 / 本地站号或特殊功能模块站号（(S1)+6）中指定的站的 (S2) 中指定的软元件后面的区域中的请求数据进行发送，以执行服务请求。
当来自目标站的软元件数据的请求结束时，(D2) 指定的结束软元件将变为 ON。



(2) 对于 CC-Link 或串行通信的连接站可以进行软元件数据的发送。

(3) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手（数据交换），因此后面的链接专用指令将无法进行处理。

(4) REQ 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(d) 通讯指示标志 *6 : REQ 指令执行时变为 ON，在传送结束的扫描的 END 处理时变为 OFF。

(e) 自站结束软元件 ((D2)) : 通过 REQ 指令执行的传送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

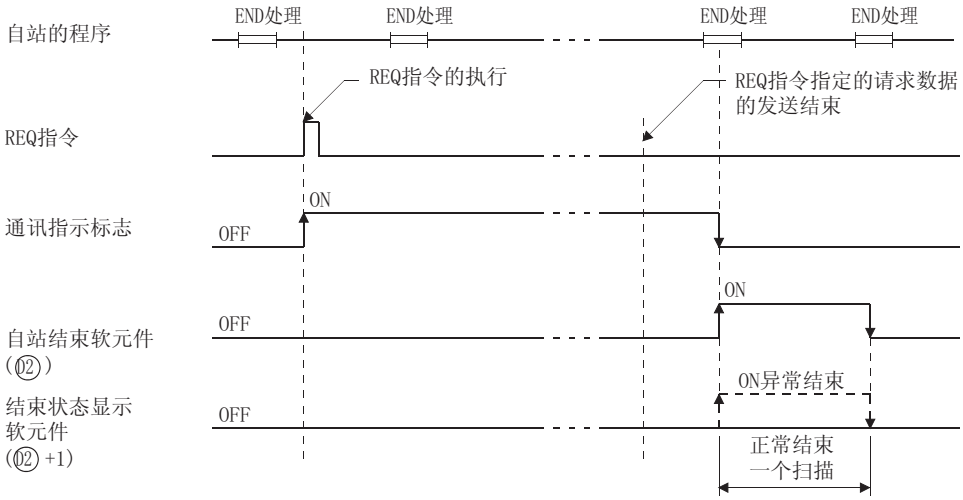
(f) 结束时的状态显示软元件 ((D2)+1) : 根据 REQ 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 错误结束 : 当 REQ 指令已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

*6: 所用通道的通讯指示标志如下所示 :

通道号	1	2
通讯指示标志号	SB30	SB32

[REQ 指令执行时的自站的动作]



运算错误

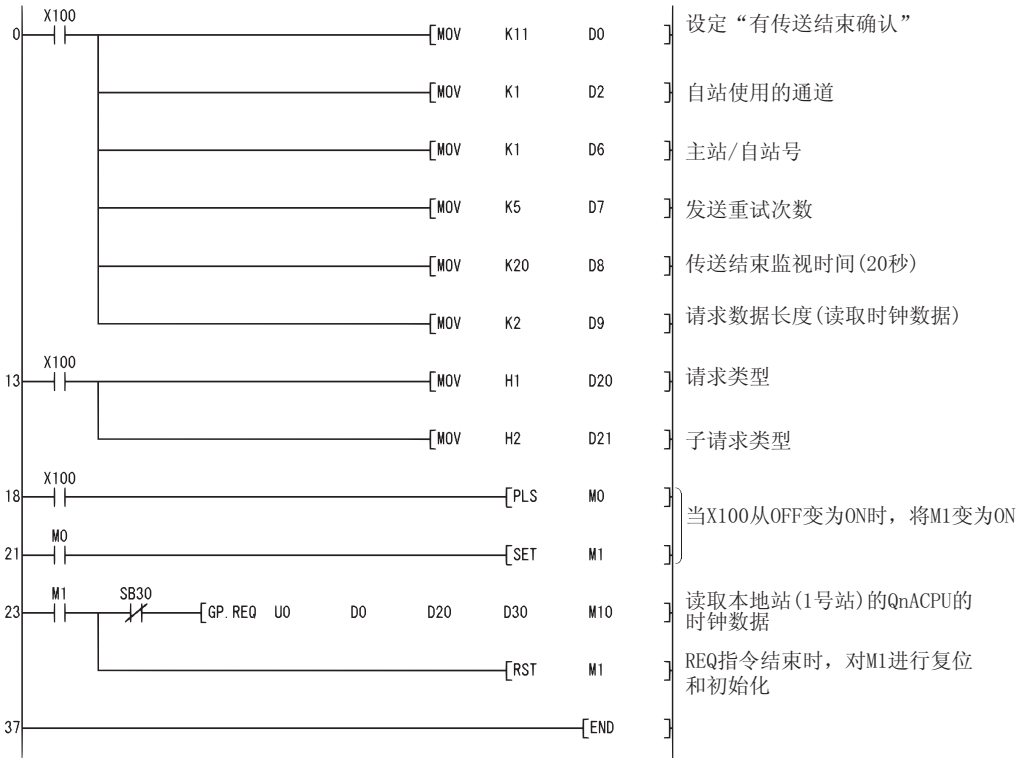
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的起始 I/O 号模块不是网络模块。 (出错代码： 2111)

程序示例

(1) 当 X100 为 ON 时，下列程序读取连接到 CC-Link 的本地站 (1 号站) 的 QnACPU 的时钟数据。

[梯形图模式]

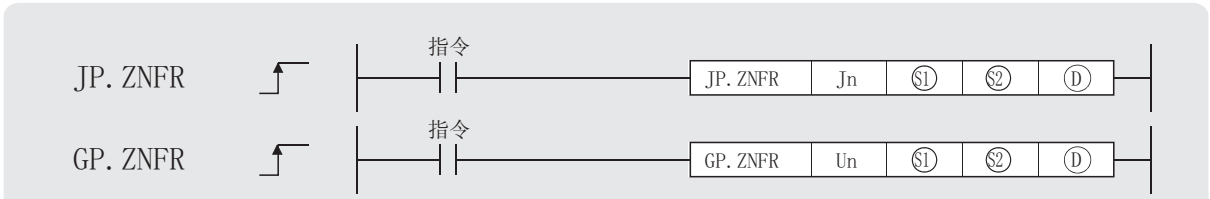


[列表模式]

步	指令	软元件				
0	LD	X100				
1	MOV	K11	D0			
3	MOV	K1	D2			
5	MOV	K1	D6			
7	MOV	K5	D7			
9	MOV	K20	D8			
11	MOV	K2	D9			
13	LD	X100				
14	MOV	H1	D20			
16	MOV	H2	D21			
18	LD	X100				
19	PLS	M0				
21	LD	M0				
22	SET	M1				
23	LD	M1				
24	MPS					
25	ANI	SB30				
26	GP.REQ	U0	D0	D20	D30	M10
35	MPP					
36	RST	M1				
37	END					

8.3.8 从远程 I/O 站的特殊功能模块中读数据 (JP/GP.ZNFR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字 --				
(S1)	--	△*2	○	--					
(S2)	--	／*3	--	--					
(D)	／*1	--	--	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只能使用链接寄存器。

○设定数据

设定数据	含义	数据类型
Jn	自站的网络号 *4	BIN 16 位
Un	网络模块中自站的起始输入输出 (I/O 号)*5	
(S1)	存储控制数据的自站软元件的起始软元件号。	软元件名称
(S2)	在读操作完成后将要存储数据的本地站链接寄存器 (W) 的起始软元件号。	
(D)	在指令结束时，软元件变为 ON 一个扫描周期。*6	位

*4 : 自站的网络号指定如下：
- 1 到 239: 网络号
- 254: 设定为其它站可以访问的有效模块中指定的网络。

*5 : 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。

*6 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

控制数据

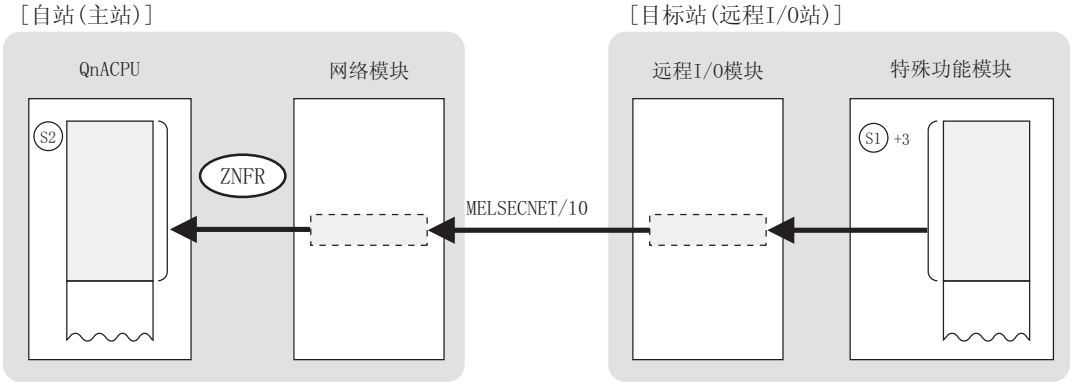
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	有传送结束确认：将位 0(b0) 设定成 1(固定的)。	0001H 0081H	用户
	异常结束类型	设置异常结束时的时钟数据的设定状态。 不设定时钟数据：将位 7(b7) 设定成 0。 设定时钟数据：将位 7(b7) 设定成 1。 (附加在 (S1)+11 的后面)		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误(正常结束) 不是 0：出错代码*7	--	系统
(S1) + 2	虚拟	禁止使用	--	--
(S1) + 3	缓冲存储区地址	指定缓冲存储区的起始地址。	*8	用户
(S1) + 4	虚拟	禁止使用	--	--
(S1) + 5	目标站号	为读软元件数据的远程 I/O 站设定站号。	1 到 64	用户
(S1) + 6	模块号指定	指定安装在目标站中的特殊功能模块的模块号。	1 到 64	用户
(S1) + 7	虚拟	禁止使用	--	--
(S1) + 8	虚拟	禁止使用	--	--
(S1) + 9	读取数据长度	设定要读取的数据块数目。	1 到 256	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 (只在异常时设定)	存储时钟数据有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 (只在异常时设定)	月和年(年份后 2 位)以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		

*7 ：关于异常时的出错代码，请参照以下手册：
- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系统以太网接口模块用户手册(详细篇)

*8 ：请参照进行读取的特殊功能模块的手册。

功 能

(1) 将控制数据的目标站号（(S1)+5）站和（(S1)+6）中指定的 MELSECNET/10 远程 I/O 站的特殊功能模块的缓冲存储区的数据，存储到 (S2) 指定的软元件后面的软元件中。
当来自远程 I/O 站的数据读取结束时，(D) 指定的结束软元件将变为 ON。



(2) 当一个远程站是来自 MELSECNET/10 网络的远程站时，读自这个远程 I/O 站的软元件数据只能被连接到同一网络的远程 I/O 站处理。

(3) ZNFR 指令不能在同一特殊功能模块的两个位置上同时执行。
当确定了指令在两个或多个位置同时执行的条件时，将自动处理握手操作，因此已执行的 ZNFR 指令将不会被处理。

(4) ZNFR 指令的执行过程中的互锁信号以及其正常 / 异常结束可以通过以下方式确认。

(a) 自站结束软元件（(D)） : 在 ZNFR 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

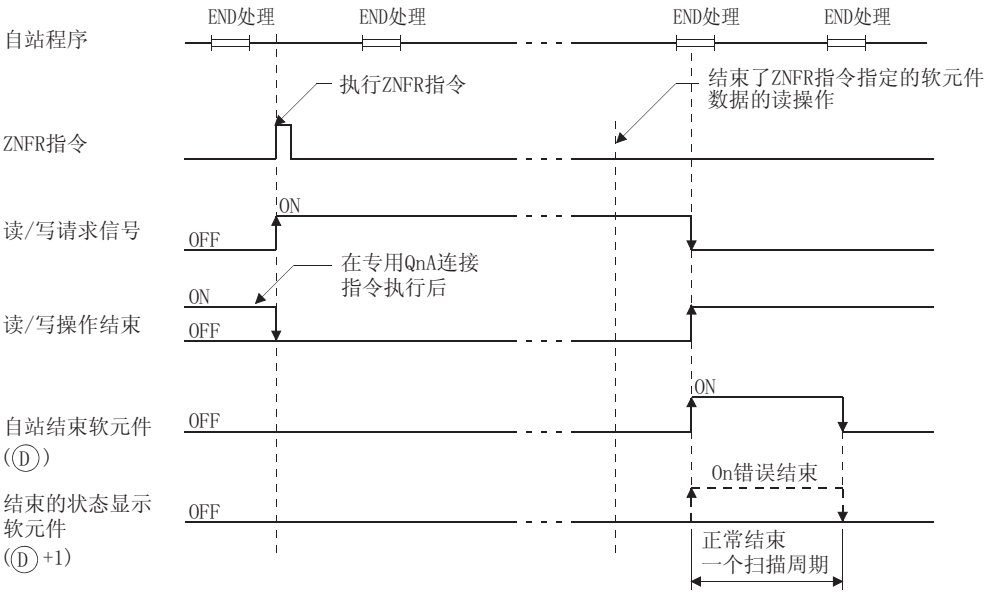
(b) 结束时的状态显示软元件（(D)+1）: 根据 ZNFR 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 错误结束 : 当 ZNFR 指令已经结束时，在扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(c) 每一个站的循环传送状态 (SW74 到 77)

: 网络中所有站的循环传送状态都以位信息形式存储，并且当一个站的循环传送发生异常网络中所有站的循环传送状态都以位信息形式存储，并且当一个站的循环传送发生异常，它的相应位被设定为 ON。
写一个程序以便 ZNFR 指令只在那些 ZNFR 执行目标站中位状态为 OFF(循环传送状态为正常) 的站上执行。

[在 ZNFR 指令执行期间自站的操作]



(5) ⓢ2 指定的链接寄存器 (W) 设为网络参数 “M ← R(从远程 I/O 站到主站)”，并且属于由链接刷新参数分配给 CPU 模块的范围内。

(6) 为了执行 ZNFR 指令，需要一个 OS 能使用的链接继电器和链接寄存器。
对于每个特殊功能模块，OS 要使用的链接继电器和链接寄存器号如下所示 :

- 对 M → R(从主站到远程 I/O 站) ... 链接继电器 :4，链接寄存器 :4
- 对 M ← R(从远程 I/O 站到主站) ... 链接继电器 :4，链接寄存器 :4

运算错误

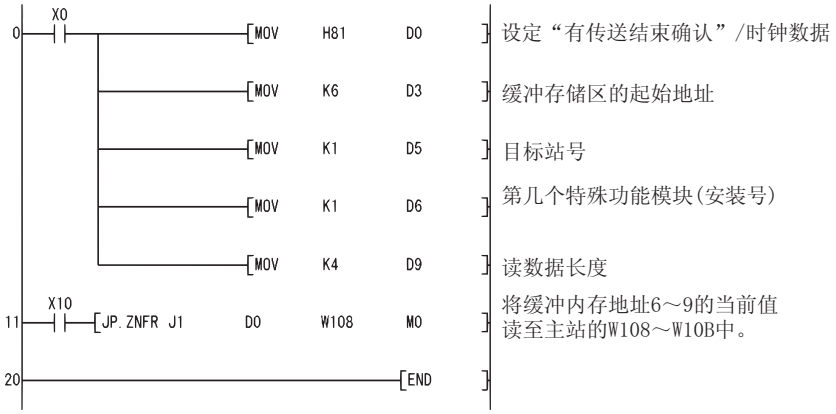
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的 I/O 地址号模块不是一个网络模块。 (出错代码： 4102)
- 由 (S2) 指定的链接寄存器 (W) 的范围超出了网络参数 “M 站← R 站 (主站←远程 I/O 站)” 中设定的范围。 (出错代码： 4101)

程序示例

(1) 以下程序将 1 号网络的 1 号远程 I/O 站的第 1 个特殊功能模块的缓冲内存地址 6 ～ 9 的内容读至主站的 W108 ～ W10B 中。
当将以下程序用于实际中时，应参阅 8.1 节创建互锁电路。

[梯形图模式]

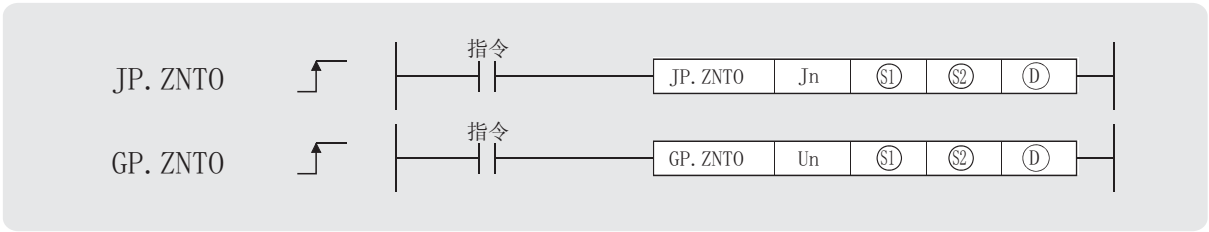


[列表模式]

步	指令	软元件			
0	LD	X0			
1	MOV	H81	D0		
3	MOV	K6	D3		
5	MOV	K1	D5		
7	MOV	K1	D6		
9	MOV	K4	D9		
11	LD	X10			
12	JP.ZNFR	J1	D0	W108	M0
20	END				

8.3.9 往远程 I/O 站的特殊功能模块中写数据 (JP/GP.ZNTO)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它其它
	位	字		位	字				
(S1)	--	△*2	○	--					
(S2)	--	△*3	--	--					
(D)	△*1	--	--	--					

*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只能使用链接寄存器。

○设定数据

设定数据	含义	数据类型
Jn	自站的网络号。*4	BIN 16 位
Un	网络模块中自站的起始输入输出 (I/O) 号。*5	
(S1)	存储控制数据的自站软元件的起始软元件号。	软元件名称
(S2)	存储写入数据的自站链接寄存器 (W) 的起始软元件号。	
(D)	在指令结束时，软元件变为 ON 一个扫描周期。*6	位

*4 : 自站的网络号指定如下：
- 1 到 239: 网络号
- 254: 设定为其它站可以访问的有效模块中指定的网络。

*5 : 自站网络模块的起始 I/O 号的指定范围为 0 到 FEH。
*6 : 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

控制数据

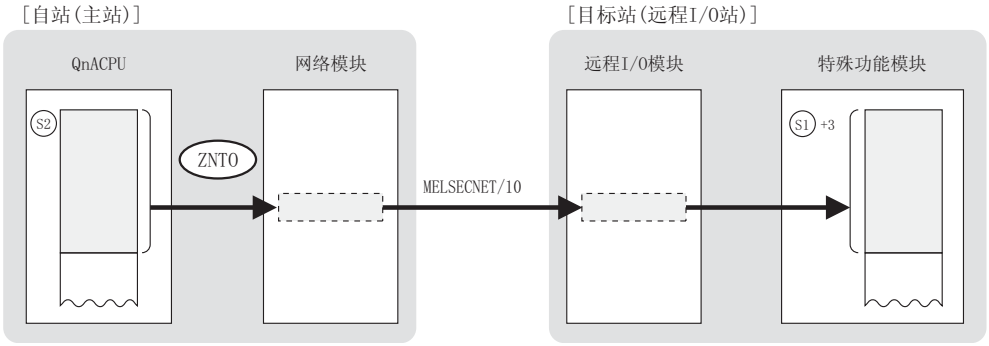
软元件	项目	设定数据	设定范围	设定方
(S1) + 0	执行类型	有传送结束确认：将位 0(b0) 设定成 1(固定)。	0001H 0081H	用户
	异常结束类型	设置异常结束时的时钟数据的设定状态。 不设定时钟数据：将位 7(b7) 设定成 0。 设定时钟数据：将位 7(b7) 设定成 1。 (附加在 (S1)+11 的后面)		
(S1) + 1	结束状态	存储指令结束后的状态。 0 ：没有错误(正常结束) 不是 0：出错代码 *7	--	系统
(S1) + 2	虚拟	禁止使用	--	--
(S1) + 3	缓冲存储区地址	指定缓冲存储区的起始地址。	*8	用户
(S1) + 4	虚拟	禁止使用	--	--
(S1) + 5	目标站号	为读远程软元件数据的远程 I/O 站设定站号。	1 到 64	用户
(S1) + 6	模块号	指定安装在目标站中的特殊功能模块的模块号。	1 到 64	用户
(S1) + 7	虚拟	禁止使用	--	--
(S1) + 8	虚拟	禁止使用	--	--
(S1) + 9	写入数据长度	设定要写入的数据块数目。	1 到 256	用户
(S1) + 10	虚拟	禁止使用	--	--
(S1) + 11	时钟设定标记 (只在异常时设定)	存储时钟数据有效 / 无效状态。 0 ：时钟数据无效。 1 ：时钟数据有效。	--	系统
(S1) + 12	时钟数据 (只在异常时设定)	月和年(年份后 2 位)以 BCD 编码存储。 高 8 位 ：月 (01H 到 12H)、低 8 位：年 (00H 到 99H)	--	系统
(S1) + 13		小时和日以 BCD 编码存储。 高 8 位 ：小时 (00H 到 23H)、低 8 位：日 (01H 到 31H)		
(S1) + 14		秒和分钟以 BCD 编码存储。 高 8 位 ：秒 (00H 到 59H)、低 8 位：分钟 (00H 到 59H)		
(S1) + 15		星期以 BCD 编码存储。 高 8 位 ：00H、低 8 位：星期 (00H 到 06H)		

*7 ：关于异常时的出错代码，请参照以下手册：
- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册

*8 ：请参照即将处理写操作的特殊功能模块的使用手册。

☆功 能

(1) 将 ⓢ2 指定的软元件后面存储的数据写到控制数据的目标站号（ⓢ1 + 5）和站（ⓢ1 + 6）中指定的 MELSECNET/10 的远程 I/O 站的特殊功能模块的缓冲存储区中。
当至远程 I/O 站的数据写入结束时， Ⓓ 指定的结束软元件将变为 ON。



(2) 通过远程 I/O 站进行软元件数据的写入时，只有在 MELSECNET/10 的远程主站→连接在同一网络的远程 I/O 站的情况下才可以执行。
从可编程控制器网络的站或使用路由器不能执行此操作。

(3) 不能从多处对同一特殊功能模块执行 ZNTO 指令。
当在两个或更多位置执行条件同时成立时，将自动执行握手（数据交换），因此后面的 ZNTO 指令将不会被处理。

(4) ZNTO 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

(a) 读 / 写请求信号 : QnA 专用链接指令执行时变为 ON，在读 / 写结束的扫描的 END 处理时变为 OFF。

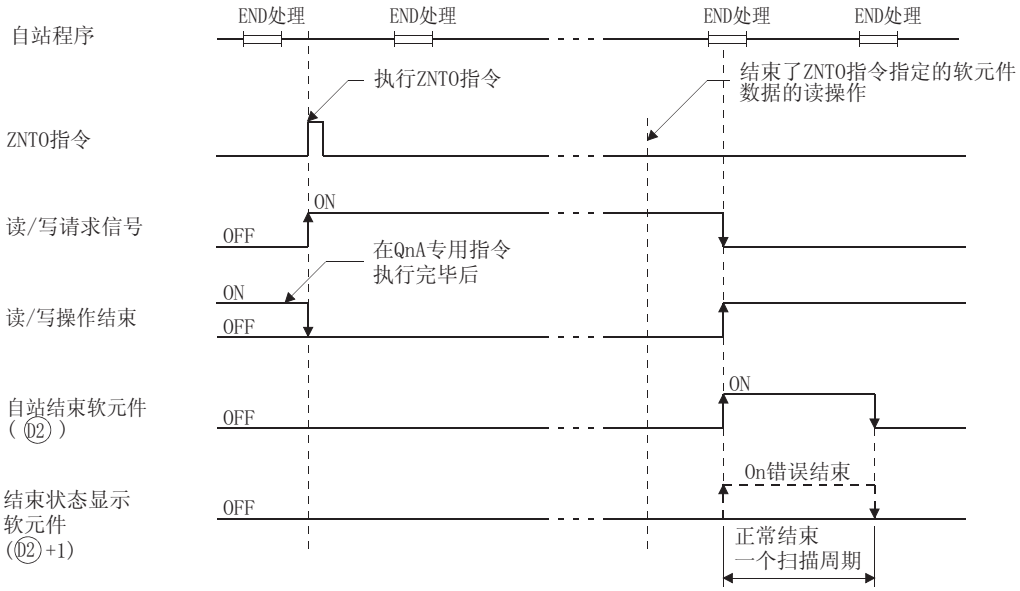
(b) 读 / 写结束信号 : QnA 专用链接指令执行时变为 OFF，在读 / 写结束的扫描的 END 处理时变为 ON。

(c) 自站结束软元件（Ⓓ） : 在 ZNTO 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(d) 结束时的状态显示软元件（Ⓓ+1）: 根据 ZNTO 指令结束时的状态而 ON/OFF。

- 正常结束： 保持 OFF 状态不变。
- 错误结束： 当 ZNTO 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[在 ZNTO 指令执行期间自站的操作]



(5) Ⓢ2 中可指定的链接寄存器 (W) 是在网络参数的 “M 站→ R 站 (主站→远程 I/O 站)” 中设定，并且在链接刷新参数分配给 CPU 模块的范围内。

(6) 为了执行 ZNTO 指令，需要一个 OS 能使用的链接继电器和链接寄存器。
对于每个特殊功能模块，OS 要使用的链接继电器和链接寄存器号如下所示：

- 对 M → R(从主站到远程 I/O 站) ... 链接继电器 :4，链接寄存器 :4
- 对 M ← R(从远程 I/O 站到主站) ... 链接继电器 :4，链接寄存器 :4

运算错误

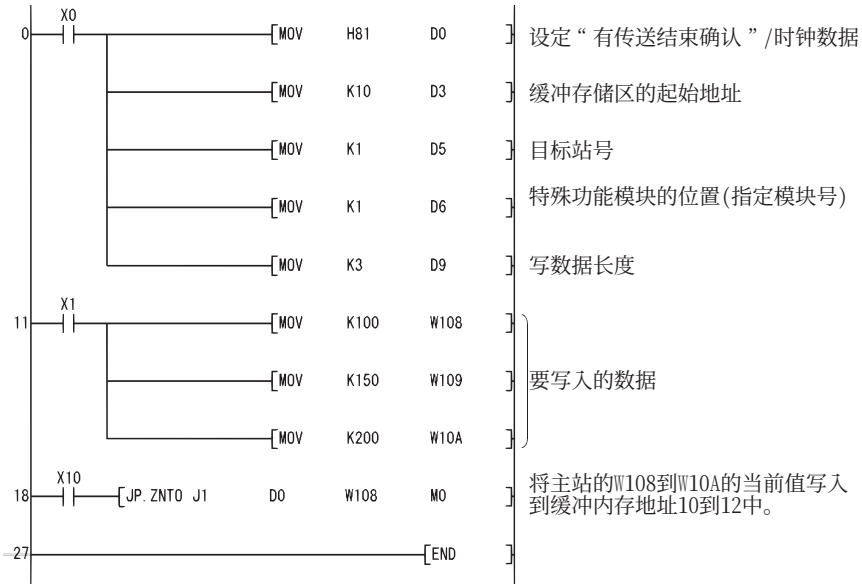
(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 控制数据内容不在设定范围内。 (出错代码： 4100)
- Jn 指定的网络号没有连接到自站。 (出错代码： 4102)
- Un 指定的 I/O 地址号模块不是一个网络模块。 (出错代码： 4102)
- Ⓢ2 指定的链接寄存器 (W) 的范围超出了网络参数的 “M 站← R 站 (主站←远程 I/O 站)” 中设定的范围。 (出错代码： 4101)

程序示例

(1) 以下程序将主站的 W108 到 W10A 中的数据写入到 1 号网络的 1 号远程 I/O 站的第 1 个特殊功能模块的缓冲内存地址 10 到 12 中。
将以下程序用于实际应用中时，请参照 8.1 节的描述建立互锁电路。

[梯形图模式]



[列表模式]

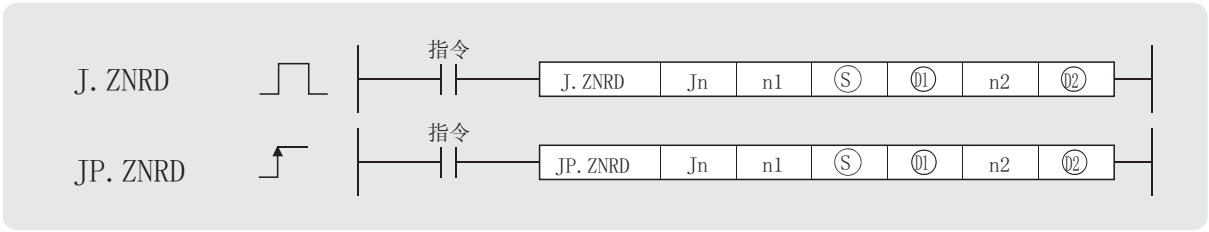
步	指令	软元件			
0	LD	X0			
1	MOV	H81	D0		
3	MOV	K10	D3		
5	MOV	K1	D5		
7	MOV	K1	D6		
9	MOV	K3	D9		
11	LD	X1			
12	MOV	K100	W108		
14	MOV	K150	W109		
16	MOV	K200	W10A		
18	LD	X10			
19	JP.ZNTO	J1	D0	W108	M0
27	END				

8.4 A 系列兼容链接指令

8.4.1 从其它站读软元件数据 (MELSECNET/10、Ethernet) (J(P).ZNRD)



*1: 关于 QCPU 可以使用的 ZNRD 指令的详细内容，请参照以下手册。
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）



Jn : 目标网络号 (BIN16 位)
n1 : 目标站号 (BIN16 位)
Ⓢ : 存储读取数据的目标站的起始软元件（软元件名称）
Ⓓ1 : 存储读取数据的自站的起始软元件（软元件名称）
n2 : 接收数据长度（软元件名称）
Ⓓ2 : 指令结束时 ON 一个扫描的软元件（位））

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	△*2	△*3	○	--				○	--
Ⓢ	--	△*4	--	--				--	--
Ⓓ1	--	△*3	△*5	--				--	--
n2	△*2	△*3	○	--				○	--
Ⓓ2	△*2	△*3	○	--				--	--

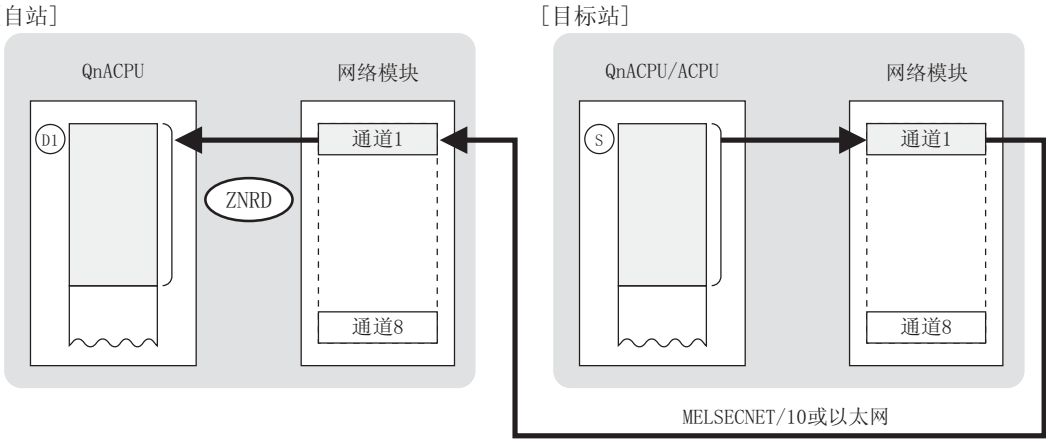
*2: 可以使用 X、Y、M、L、F、V 和 B。
*3: 可以使用 T、ST、C、D 和 W。
*4: 只可以使用 T、C、D 和 W。
*5: 只有当目标为 QnACPU 时可以使用。

要 点

1. 对于通过 ZNRD 指令进行数据读取的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
（目标站 CPU 模块的读取目标的起始软元件号 Ⓢ）+（读取点数 - 1）
≦（Ⓢ 指定的最终软元件号（自站 CPU 模块））
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
关于详细内容，请参考以下手册。
- QnACPU 编程手册（特殊功能模块篇）

☆功 能

(1) 在 MELSECNET/10 或以太网的数据链接系统中，将 Jn • n1 中指定网络号的连接站的 Ⓢ 中指定的软元件后面的数据，读到 Ⓓ1 指定的软元件后面的软元件中。
从目标站读取软元件数据结束时 Ⓓ2 指定的结束软元件 ON 一个扫描。



(2) 当自站连接到 MELSECNET/10 或以太网时，可以从连接到自站网络的目标站执行软元件数据的读取。
注意必须设定路由参数。

(3) Jn 中可指定的网络号为 1 到 239。
指定了 0(J0) 时与 MELSECNET(II，/B) 的指定相同。(参照 8.4.2 项)

(4) 在 n1 中可指定的站号为 1 到 64。

(5) 对于 AnUCPU 站可以像 QnACPU 站一样，通过 ZNRD 指令执行来自其它站的读取。

(6) n2 中可指定的接收数据长度 (点数) 为 1 到 230。
当除 AnUCPU 外的 ACPU 被指定为目标站时，可以接受的请求数据块数最多为 32，但如果设定为 33 以上，也不会出现错误。
当在 n2 中指定 0 时，将不执行处理。

(7) 在同一通道中，不能在多个位置上同时执行链接专用指令。
当在两个或更多位置同时执行的条件成立时，将自动执行握手 (数据交换)，因此后面的链接专用指令将无法进行处理。

(8) 对于 ZNRD 指令，自站和目标站都使用网络模块的通道 1。
不能对指定网络模块同时执行多个 ZNRD 指令，也不能同时执行多个使用通道 1 的 QnA 链接专用指令。应建立一个基于读 / 写请求信号和结束软元件的互锁，以防止同时执行多个链接专用指令。

(9) ZNRD 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

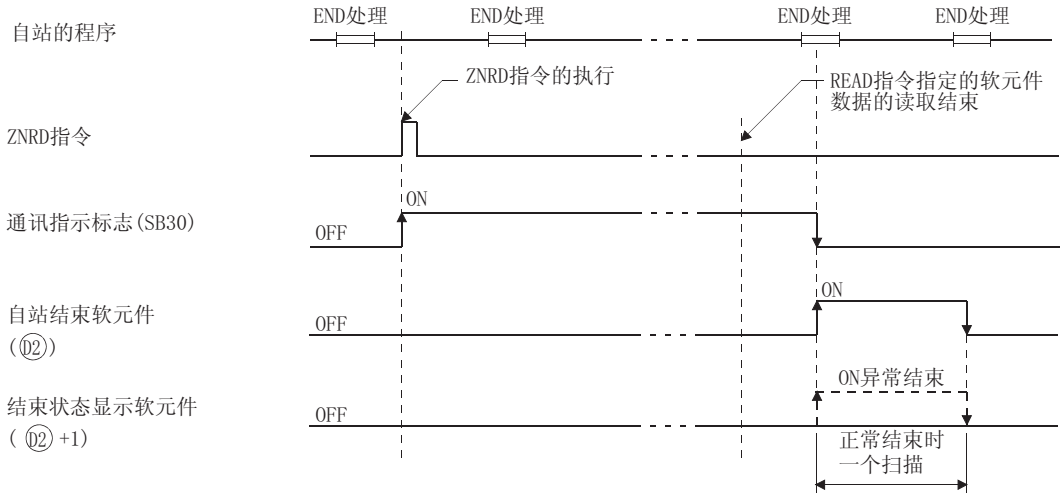
(a) ZNRD 指示标志 (SB30) : ZNRD 指令执行时变为 ON，在读取结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (Ⓓ2) : 在 ZNRD 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(c) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 ZNRD 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : 在 ZNRD 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[ZNRD 指令执行时的自站的动作]



(10)ZNRD 指令的正常 / 错误结束可以用 ZNRD 指令处理结束寄存器 (SW31) 来确定。

正常结束时，SW31 为 0。

错误结束时，SW31 存储了适合的出错代码。*6

*6 : 错误结束时的出错代码的具体细节请参照以下手册 :

- QnA/Q4AR MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册 (详细篇)

运算错误

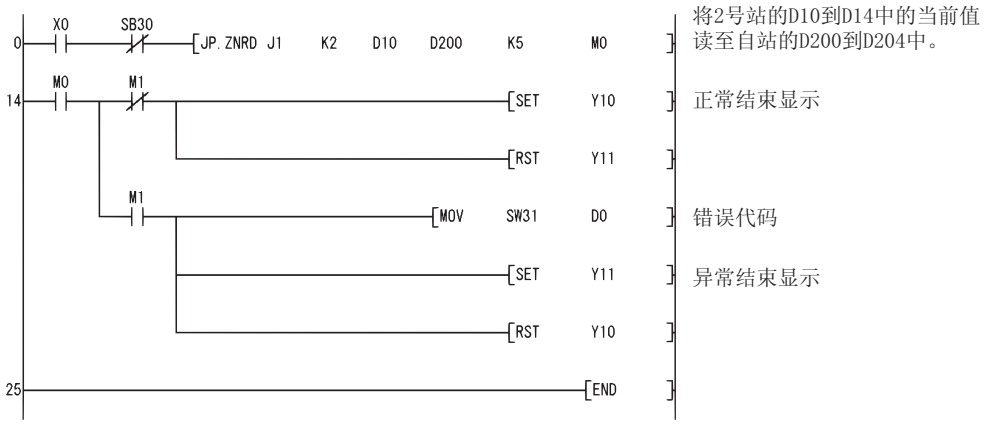
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 指定的软元件起的 n2 点数超过了相应软元件的最终软元件号。 (出错代码： 4101)
- Jn 指定的网络号不存在。 (出错代码： 4102)
- n1 指定的站号不存在。 (出错代码： 4100)
- n2 指定的接收数据长度不在 1 到 230 的范围内。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，以下程序将连接在 MELSECNET/10 中的 2 号站的 D10 到 D14 中的数据读至自站的 D200 到 D204 中。

[梯形图模式]

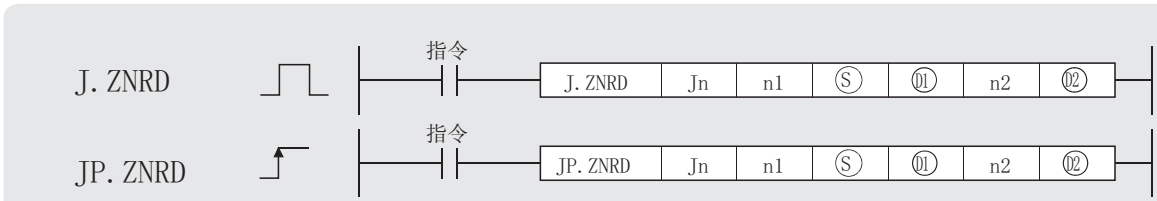


[列表模式]

步	指令	软元件					
0	LD	X0					
1	ANI	SB30					
2	JP.ZNRD	J1	K2	D10	D200	K5	M0
14	LD	M0					
15	MPS						
16	ANI	M1					
17	SET	Y10					
18	RST	Y11					
19	MPP						
20	AND	M1					
21	MOV	SW31	D0				
23	SET	Y11					
24	RST	Y10					
25	END						

8.4.2 从本地站 (MELSECNET) 读软元件数据 (J(P).ZNRD)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Jn : 目标网络号（固定为 0(J0))(BIN16 位）

n1 : 目标本地站号 (BIN16 位）

Ⓢ : 存储读取数据的本地站的起始软元件（软元件名称）

Ⓓ1 : 存储读取数据的自站的起始软元件（软元件名称）

n2 : 接收数据长度（软元件名称）

Ⓓ2 : 在指令结束时 ON 一个扫描的软元件（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	△*1	△*2	○	--				○	--
Ⓢ	--	△*3	--	--				--	--
Ⓓ1	--	△*2	○	--				--	--
n2	--	△*2	○	--				○	--
Ⓓ2	△*1	△*2	○	--				--	--

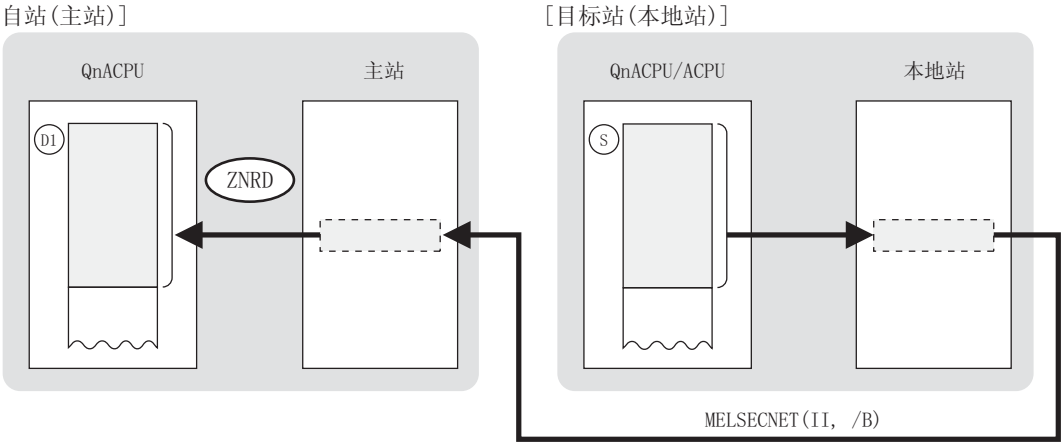
*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只可以使用 T、C、D 和 W。d

要 点

1. 对于通过 ZNRD 指令进行数据读取的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
   （Ⓢ 目标站 CPU 模块的读取目标的起始软元件号）+（读取点数 - 1）
   ≦（Ⓢ 指定的最终软元件号（自站 CPU 模块））
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
   关于详细内容，请参考以下手册。
   • QnACPU 编程手册（特殊功能模块篇）

☆功 能

(1) 在 MELSECNET(II，/B) 中，从 n1 指定的本地站的 Ⓢ 中指定的字软元件中读取两点数据，并把它存储到 Ⓓ1 中指定的自站的软元件的后面。
当来自本地站的软元件数据的读取结束时，Ⓓ2 指定的结束软元件将 ON1 个扫描。



(2) 只有通过 MELSECNET 的主站才可以从本地站中读取软元件数据。

(3) 在 MELSECNET(II，/B) 系统中，目标网络号 (Jn) 固定为 0(J0)。
在 Jn 中指定了 1 到 239 时，与 MELSECNET/10 和以太网的指定相同。(参照 8.4.1 项)

(4) 站号 1 到 64 在 n1 处指定。
当为 MELSECNET/B 时，1 到 31 的站号在 n1 处指定。

(5) Ⓢ 指定的软元件范围如下所示：

	MELSECNET 模式	MELSECNET II 到模式
T	T0 到 T255	T0 到 T2047
C	C0 到 C255	C0 到 C1023
D	D0 到 D1023	D0 到 D6143
W	W0 到 W3FF	W0 到 WFFF

(6) 接收数据长度 (点数)1 到 32 在 n2 处指定。当在 n2 处指定为 0 时，不执行任何进程。

(7) ZNRD 和 ZNWR 指令不能同时执行。
当确定了指令在两个或两个以上位置同时执行的条件时，由于自动处理握手操作，因此后续 ZNRD 和 ZNWR 指令的处理将不能完成。
要执行多个 ZNRD 和 ZNWR 指令，采用结束软元件，并随后执行指令。

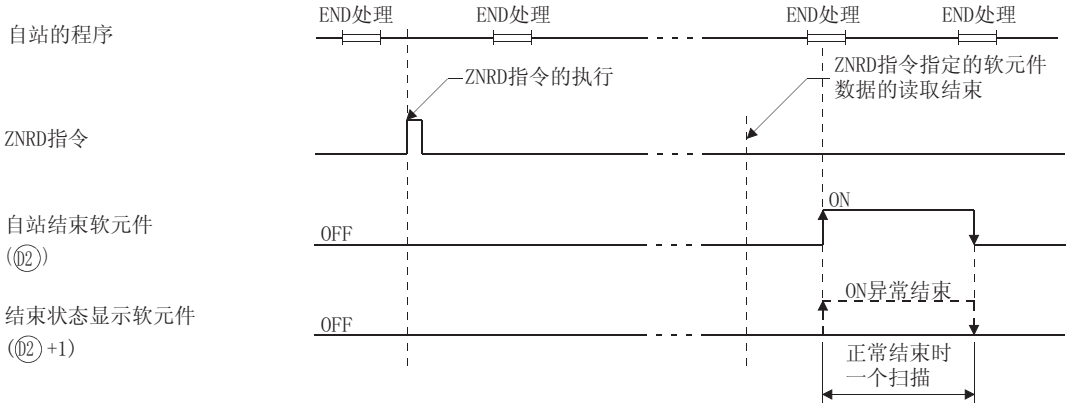
(8) ZNRD 指令的正常 / 异常结束可以通过以下方式确认。

(a) 自站结束软元件 (Ⓓ2) : 通过 ZNRD 指令执行的发送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(b) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 ZNRD 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : ZNRD 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[ZNRD 指令执行时的自站的动作]



运算错误

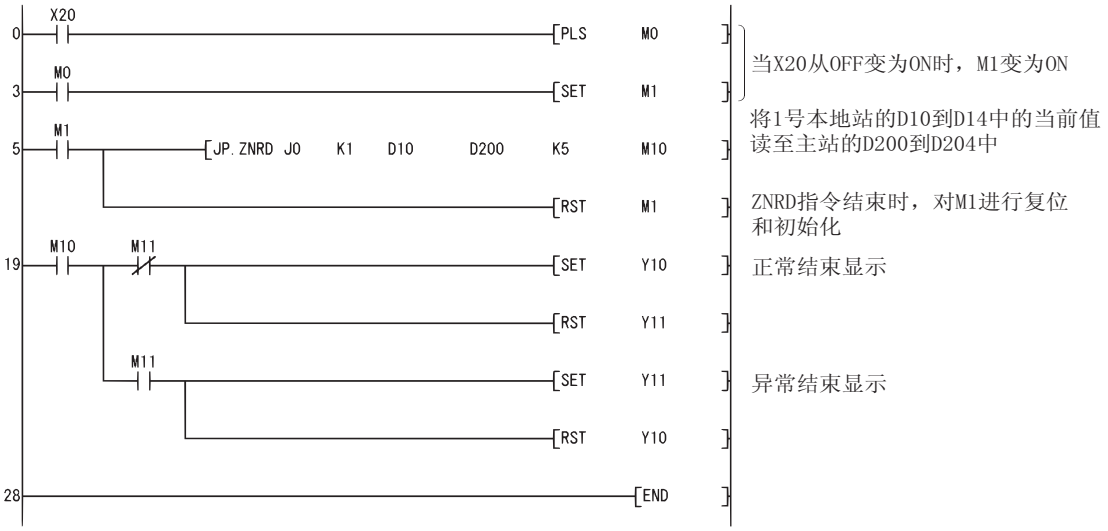
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 指定的软元件起的 n2 点数超过了相应软元件范围的最终软元件号。 (出错代码 : 4101)
- Jn 指定的网络号不存在。 (出错代码 : 4100)
- n1 指定的站号不存在。 (出错代码 : 4102)
- n2 指定的接收数据长度不在 1 到 32 的范围内。 (出错代码 : 4100)

程序示例

(1) 当主站的 X20 为 ON 时，以下程序将 1 号本地站的 D10 到 D14 中的数据读至主站的 D200 到 D204 中。

[梯形图模式]



[列表模式]

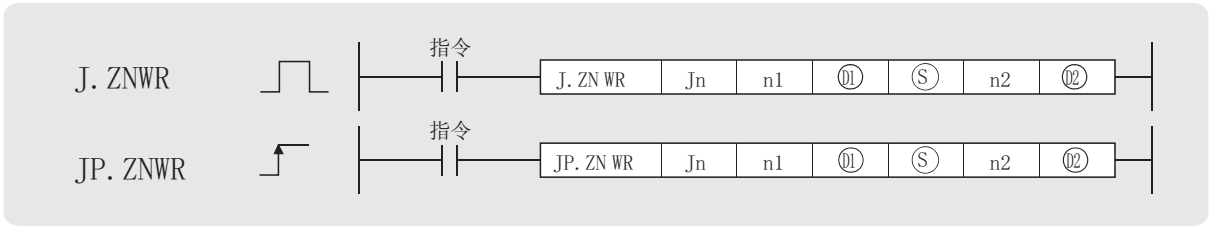
步	指令	软元件					
0	LD	X20					
1	PLS	M0					
3	LD	M0					
4	SET	M1					
5	LD	M1					
6	JP.ZNRD	J0	K1	D10	D200	K5	M10
18	RST	M1					
19	LD	M10					
20	MPS						
21	ANI	M11					
22	SET	Y10					
23	RST	Y11					
24	MPP						
25	AND	M11					
26	SET	Y11					
27	RST	Y10					
28	END						

8.4.3 将软元件数据写入其它站（MELSECNET/10、以太网）(J(P).ZNWR)



*1：关于 QCPU 可以使用的 ZNWR 指令的详细内容，请参照以下手册。

- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- Q 系列以太网接口模块用户手册（应用篇）



Jn ：目标网络号（BIN16 位）

n1 ：目标站号（BIN16 位）

(D1) ：写入数据的目标站的起始软元件（软元件名称）

(S) ：存储写入数据的自站的起始软元件（软元件名称）

n2 ：写入点数（软元件名称）

(D2) ：指令结束时 ON 一个扫描的软元件（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	△*2	△*3	○	--				○	--
(D1)	--	△*4	--	--				--	--
(S)	--	△*3	△*5	--				--	--
n2	--	△*3	○	--				○	--
(D2)	△*2	△*3	○	--				--	--

*2：可以使用 X、Y、M、L、F、V 和 B。
*3：可以使用 T、ST、C、D 和 W。
*4：只可以使用 T、C、D 和 W。
*5：只有当目标为 QnACPU 时可以使用。

要 点

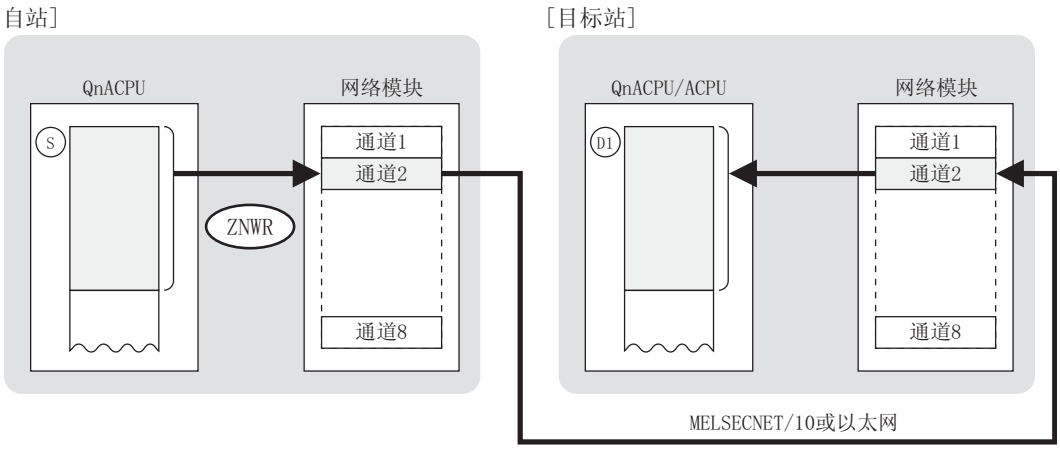
1. 对于通过 ZNWR 指令进行数据写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。

   （目标站 CPU 模块的写入目标的起始软元件号 (S) ）+（写入点数 - 1）

   ≦（(S) 指定的最终软元件号（自站 CPU 模块））
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
   关于详细内容，请参考以下手册。
   - QnACPU 编程手册（特殊功能模块篇）

☆功 能

(1) 在 MELSECNET/10 或以太网数据链接系统中，将自站的 Ⓢ 中指定的字软元件开始的 n2 点数的数据，存储到 Jn・n1 中指定的网络号的连接站的 Ⓓ1 中指定的软元件的后面。
至目标站的软元件数据的写入结束时， Ⓓ2 中指定的结束软元件将变为 ON。



(2) 只有连接在 MELSECNET/10 或以太网中的自站的网络的连接站，才可以进行自站的软元件数据的写入。
注意必须设定路由参数。

(3) 在 Jn 中可指定的网络号为 1 到 239。
指定了 0(J0) 时，与 MELSECNET(II，/B) 的指定相同。(参照 8.4.4 项)

(4) 在 n1 中可指定的站号为 1 ～ 64。

(5) n2 中可指定的发送数据长度为 1 到 230。
当在 n2 中指定了 0 时，将不执行处理。

(6) 对于 AnUCPU 站可以像 QnACPU 站一样，可以通过 ZNWR 指令对其它站进行写入。

(7) 在 ZNWR 指令中，自站和目标站都使用网络模块的通道 2。
不能对指定网络模块同时执行多个 ZNWR 指令，也不能同时执行多个使用通道 2 的 QnA 链接专用指令。
应建立一个基于读 / 写请求信号和结束软元件的互锁，以防止同时执行多个链接专用指令。

(8) ZNWR 指令的执行状态以及其正常 / 异常结束可以通过以下方式确认。

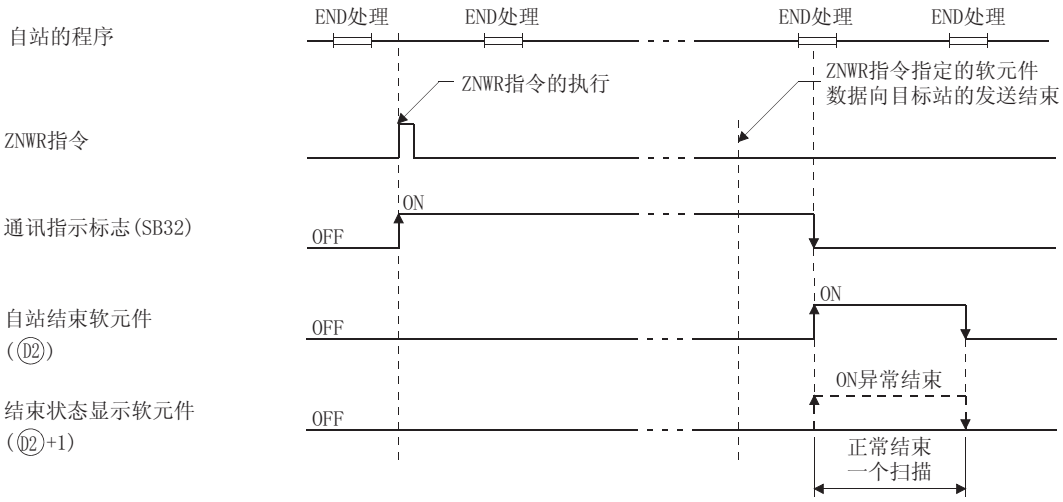
(a) ZNWR 指示标志 (SB32) : ZNWR 指令执行时变为 ON，在写入结束的扫描的 END 处理时变为 OFF。

(b) 自站结束软元件 (Ⓓ2) : 在 ZNWR 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(c) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 ZNWR 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : 在 ZNWR 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[ZNWR 执行时的自站的动作]



(9) ZNWR 指令的正常 / 错误结束可以用 ZNWR 指令处理结束寄存器 (SW33) 来确定。
正常结束时，SW33 为 0。
错误结束时，SW33 存储了适合的出错代码。*6

*6 : 错误结束时的出错代码的具体细节请参照以下手册 :
- QnA/Q4AR MELSECNET/10 网络系统参考手册
- QnA 系列以太网接口模块用户手册 (详细篇)

运算错误

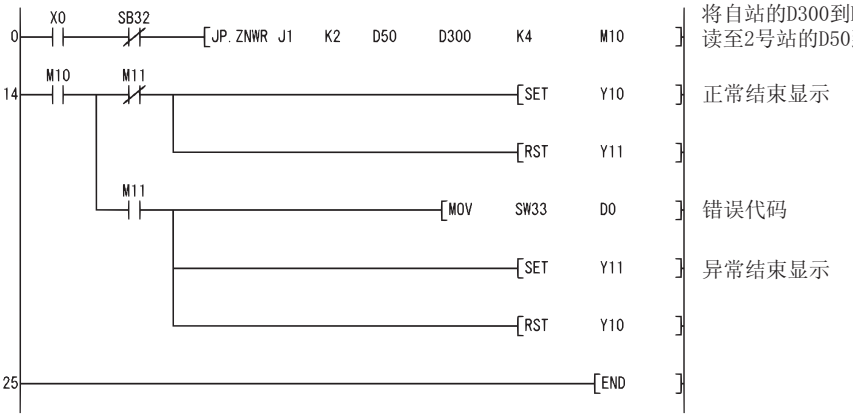
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 从 Ⓢ 指定的软元件起的 n2 点数超过了相应软元件的最终软元件号。 (出错代码： 4101)
- Jn 指定的网络号不存在。 (出错代码： 4102)
- n1 指定的站号不存在。 (出错代码： 4100)
- n2 指定的接收数据长度不在 1 到 230 的范围内。 (出错代码： 4100)

程序示例

(1) 当 X0 为 ON 时，以下程序将自站的 D300 到 D303 中的数据，写入连接在 MELSECNET/10 中的 2 号站的 D50 到 D53 中。

[梯形图模式]

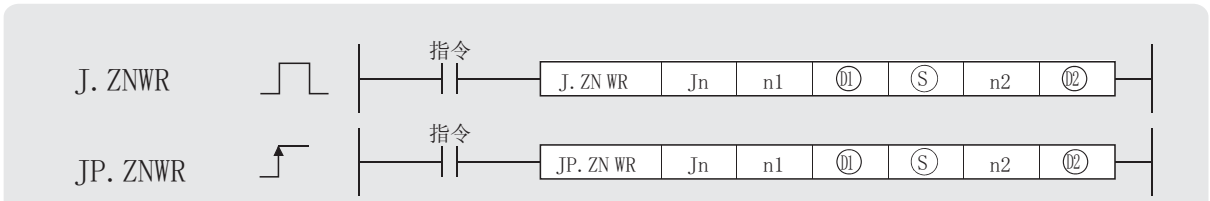


[列表模式]

步	指令	软元件
0	LD	X0
1	ANI	SB32
2	JP.ZNWR	J1 K2 D50 D300 K4 M10
14	LD	M10
15	MPS	
16	ANI	M11
17	SET	Y10
18	RST	Y11
19	MPP	
20	AND	M11
21	MOV	SW33 D0
23	SET	Y11
24	RST	Y10
25	END	

8.4.4 将数据写入本地站软元件中 (MELSECNET)(J(P).ZNWR)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Jn ：目标网络号（固定为 0(J0))(BIN16 位）

n1 ：目标本地站号 (BIN16 位）

(D1) ：写入数据的本地站的起始软元件（软元件名称）

(S) ：存储写入数据的自站的起始软元件（软元件名称）

n2 ：发送数据长度（软元件名称）

(D2) ：在指令结束时 ON 一个扫描的软元件（位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	∧*1	∧*2	○	--				○	--
(D1)	--	∧*3	--	--				--	--
(S)	--	∧*2	○	--				--	--
n2	--	∧*2	○	--				○	--
(D2)	△*1	△*2	○	--				--	--

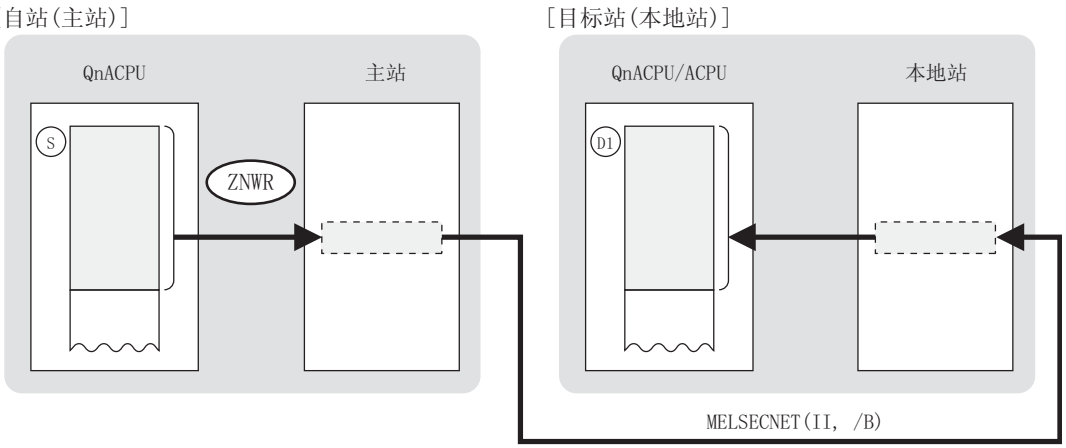
*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只可以使用 T、C、D 和 W。

要 点

1. 对于通过 ZNWR 指令进行数据写入的目标站 CPU 模块的软元件，应在可应用于自站 CPU 模块的范围内进行指定。
   （(S) 目标站 CPU 模块的写入目标的起始软元件号）+（写入点数 - 1）
   ≦（(S) 指定的最终软元件号（自站 CPU 模块））
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
   关于详细内容，请参考以下手册。
   • QnACPU 编程手册（特殊功能模块篇）

☆功 能

(1) 在 MELSECNET(II，/B) 中，将 Ⓢ 中指定的自站的字软元件开始的 n2 点数的数据，存储到 n1 中指定的本地站的 Ⓓ1 中指定的软元件后面。
当对本地站的软元件数据的写入结束时， Ⓓ2 中指定的结束软元件将 ON 一个扫描。



(2) 只有在 MELSECNET(II，/B) 的主站中，才可以通过本地站执行软元件数据的写入。

(3) 在 MELSECNET(II，/B) 系统中，目标网络号 (Jn) 固定为 0(J0)。
在 Jn 中指定了 1 到 239 时，与 MELSECNET/10 或以太网的指定相同。(参照 8.4.3 项)

(4) 站号 1 到 64 可以在 n1 处指定。
在 MELSECNET/B 中，1 到 31 的站号在 n1 处指定。

(5) Ⓓ1 指定的软元件范围如下所示。

	MELSECNET 模式	MELSECNET II 模式
T	T0 到 T255	T0 到 T2047
C	C0 到 C255	C0 到 C1023
D	D0 到 D1023	D0 到 D6143
W	W0 到 W3FF	W0 到 WFFF

(6) n2 指定的传送数据长度 (点数) 为 1 到 32。
当在 n2 处指定为 0 时，不执行任何处理。

(7) ZNRD 和 ZNWR 指令不能同时执行。
使用结束软元件预设一个互锁，以便 ZNRD 和 ZNWR 指令将不会同时执行。
当 ZNRD 和 ZNWR 指令执行一次以上时，是用一个结束软元件使得它们可以连续执行。

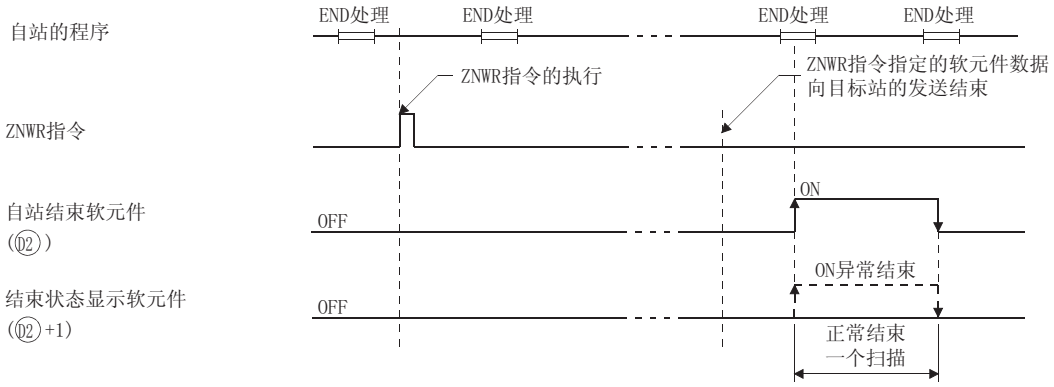
(8) ZNWR 指令的正常 / 异常结束可以通过以下方式确认。

(a) 自站结束软元件 (Ⓓ2) : 通过 ZNWR 指令执行的传送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(b) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 ZNWR 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变
- 异常结束 : 在 ZNWR 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[ZNWR 执行时自站的动作]



运算错误

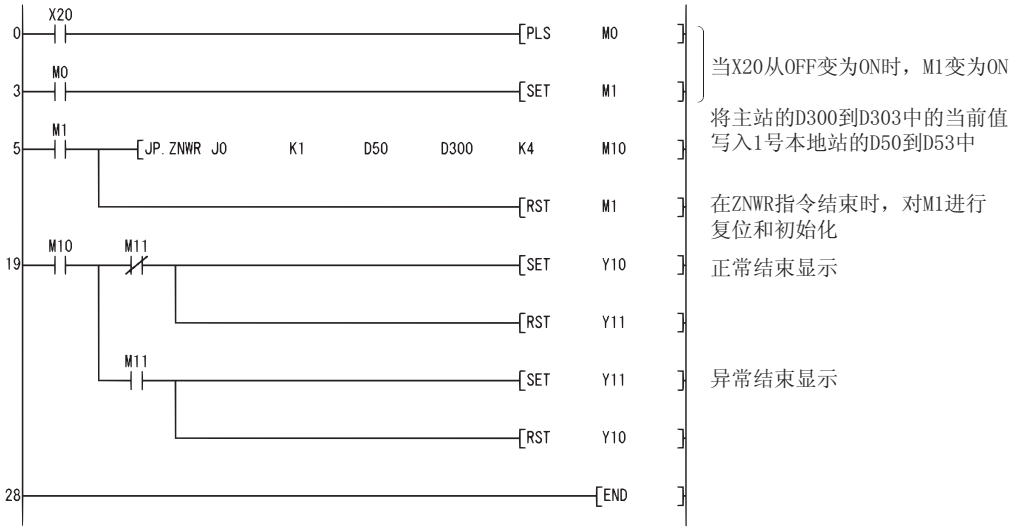
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 来自 Ⓢ 指定的软元件的 n2 点超过了相应软元件范围的最终软元件号。 (出错代码 : 4101)
- Jn 指定的网络号不存在。 (出错代码 : 4100)
- n1 指定的站号不存在。 (出错代码 : 4102)
- n2 指定的接收数据长度不在 1 到 32 的范围内。 (出错代码 : 4100)

程序示例

(1) 当主站的 X20 为 ON 时，以下程序将主站的 D300 到 D303 中的数据写入 1 号本地站的 D50 到 D53 中。

[梯形图模式]

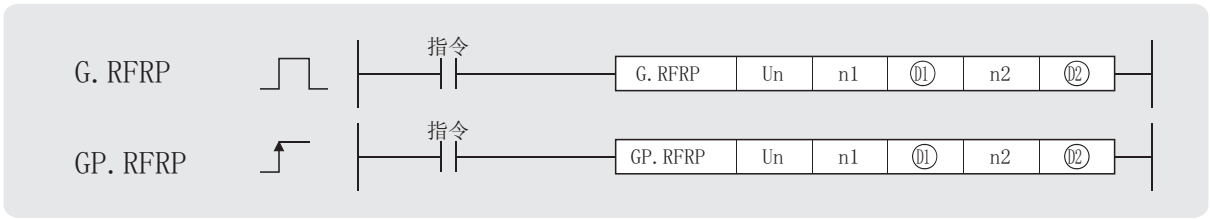


[列表模式]

步	指令	软元件
0	LD	X20
1	PLS	M0
3	LD	M0
4	SET	M1
5	LD	M1
6	JP.ZNWR	J0 K1 D50 D300 K4 M10
18	RST	M1
19	LD	M10
20	MPS	
21	ANI	M11
22	SET	Y10
23	RST	Y11
24	MPP	
25	AND	M11
26	SET	Y11
27	RST	Y10
28	END	

8.4.5 从远程 I/O 站的特殊功能模块读数据 (MELSECNET)(G(P).RFRP)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Un : 为执行数据读取的特殊功能模块分配的主站的起始 I/O 号 (BIN16 位)

n1 : 存储读取数据的特殊功能模块的缓冲存储器的起始号 (BIN16 位)

D1 : 存储读取数据的自站的链接寄存器的起始号 (软元件名称)

n2 : 接收数据长度 (BIN16 位)

D2 : 在指令结束时 ON 一个扫描的软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	△*1	△*2	○	--				○	--
D1	--	△*3	--	--				--	--
n2	△*1	△*2	○	--				○	--
D2	△*1	△*2	○	--				--	--

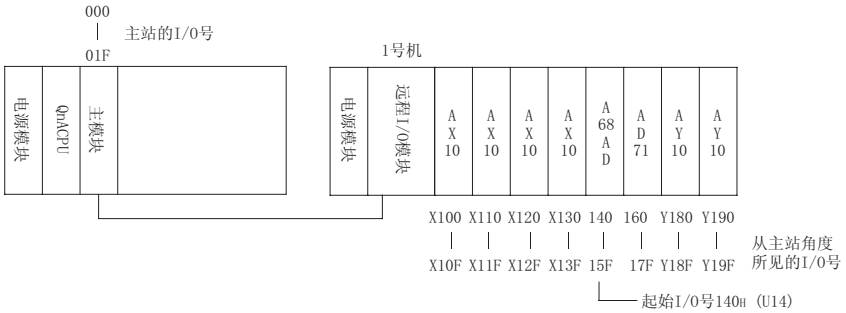
*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只可以使用链接寄存器。

要 点

1. 将从主站角度所见的特殊功能模块的起始 I/O 号指定到 Un。
   Un 中指定的特殊功能模块的起始 I/O 号是将从主站角度所见的特殊功能模块的起始 I/O 号以 4 位表示时的高 3 位进行设定的。
   例如，X/Y0140 的情况下其高 3 位为 014 因此指定为 “U14”。

   例如

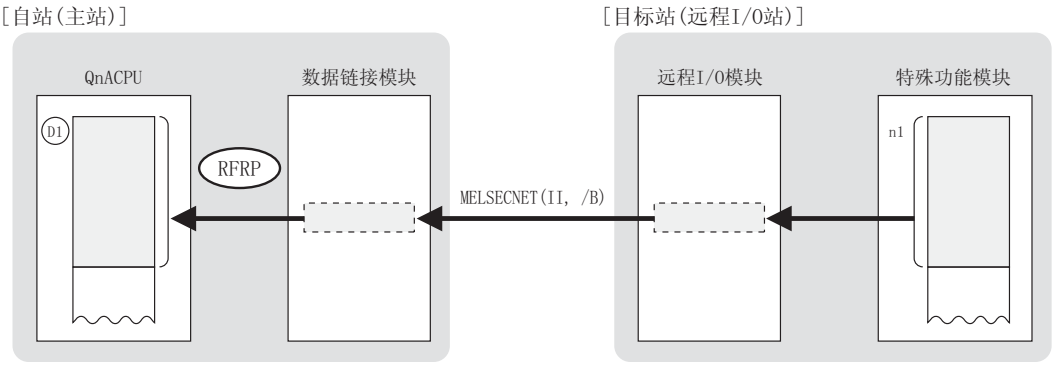
   假设网络参数 X100 到 X17F 和 Y140 到 Y19F 分配到 1 号远程 I/O 站。



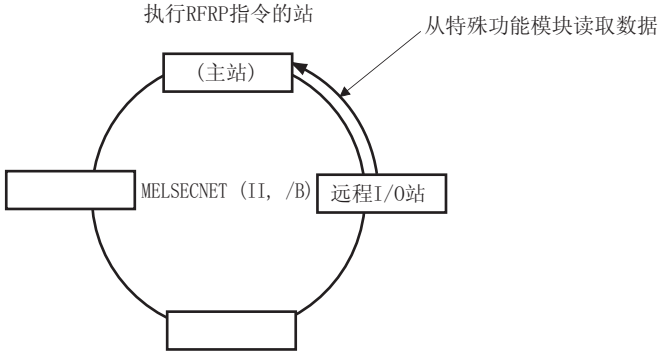
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
   关于详细内容，请参考以下手册。
   • QnACPU 编程手册 (特殊功能模块篇)

☆功 能

(1) 在 MELSECNET(II，/B) 中，将 Un 指定的 I/O 号中分配的远程 I/O 站特殊功能模块的 n1 中指定的缓冲存储器中的 n2 点的数据，存储到 ⓓ1 指定的自站（主站）的链接软元件后面的软元件中。
在来自远程 I/O 站的软元件数据的读取结束时，ⓓ2 指定的结束软元件将 ON 一个扫描。



(2) 对远程 I/O 站的软元件数据的读取只能通过 MELSECNET(II，/B) 主站进行。
而且，只有占用 32 个 I/O 点的特殊功能模块才可以执行 RFRP 指令。



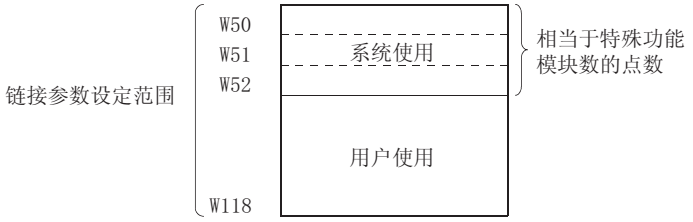
(3) 如果特殊功能模块发生异常，X(n+1)D 为 ON 时执行了 RFRP 指令，YnD 将自动变为 ON 且执行错误复位。
但是，在错误复位处理过程中不能进行通讯处理。

(4) 关于 n1 指定的特殊功能模块缓冲存储器的有关内容，请参阅特殊功能模块的使用手册。

(5) n2 中可指定的接收数据长度（点数）为 1 到 16。

(6) ⓓ1 中指定的链接寄存器的设定范围为网络参数中 “主站←远程 I/O 站”中设定的范围。

(7) 即使使用 RFRP 指令对特殊功能模块执行只读操作，也必须在网络参数中设定系统使用的“主站→远程 I/O 站”的链接寄存器。
从网络参数中设置为 “主站→远程 I/O 站”的链接寄存器的起始部分开始，到相当于相应远程 I/O 站中安装的特殊功能模块个数为止的区域，都为系统使用。
例如，在网络参数中 “主站→远程 I/O 站”设定为 W50 到 W118，由于相应远程 I/O 站中安装了 3 个特殊功能模块，因此 W50 到 W52 将保留给系统使用。



(8) 对同一特殊功能模块，RFRP、RTOP 指令不能同时执行。
当在 2 处以上同时执行条件成立时，将自动进行握手（数据交换），因此后面的 RFRP/RTOP 指令将不会被处理。
如果对同一特殊功能模块既执行 RFRP 又执行 RTOP 指令，则应使用结束软元件后依次执行。

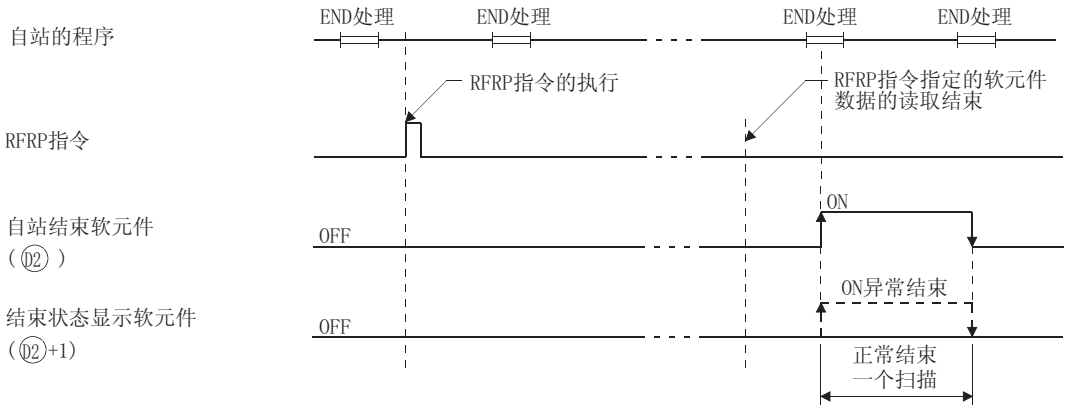
(9) RFRP 指令的正常 / 异常结束可以通过以下方式确认。

(a) 自站结束软元件 (Ⓓ2) ：通过 RFRP 指令执行的发送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(b) 结束时的状态显示软元件 (Ⓓ2+1)：根据 RFRP 指令结束时的状态而 ON/OFF。

- 正常结束： 保持 OFF 状态不变
- 异常结束： 在 RFRP 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[RFRP 指令执行时的自站的动作]



运算错误

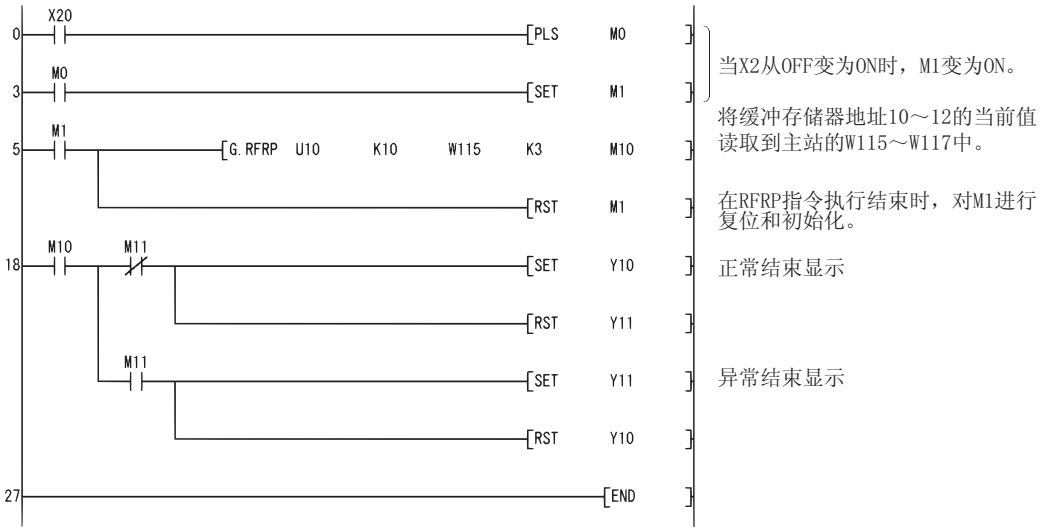
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- Un 指定的 I/O 号与远程 I/O 站的 I/O 号不一致。 (出错代码： 4102)
- Un 指定的 I/O 号不是一个特殊功能模块的起始 I/O 号。 (出错代码： 4102)
- (D1) 指定的软元件起的 n2 点数超过了相应软元件的最终软元件号。(出错代码： 4101)
- n2 指定的点数超过了 1 到 16 的范围。(出错代码： 4100)
- (D1) 指定的链接寄存器 (W) 的范围超过了网络参数的 “主站←远程 I/O 站” 中设定的范围。 (出错代码： 4102)

程序示例

(1) X20 变为 ON 时，以下程序将 1 号远程 I/O 站的第 1 个特殊功能模块的缓冲存储器地址 10 ～ 12 的数据，读取到主站的 W115 ～ W117 中。

[梯形图模式]

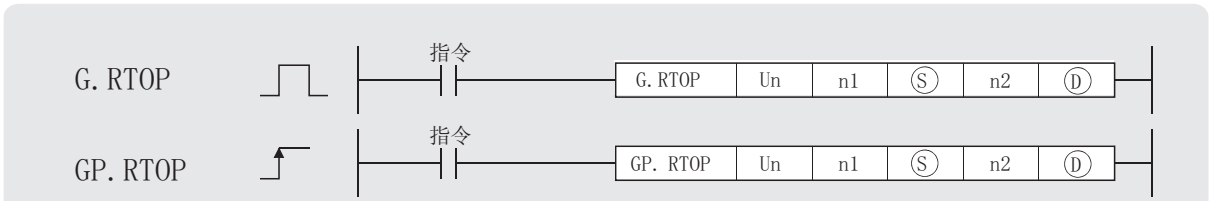


[列表模式]

步	指令	软元件				
0	LD	X20				
1	PLS	M0				
3	LD	M0				
4	SET	M1				
5	LD	M1				
6	G.RFRP	U10	K10	W115	K3	M10
17	RST	M1				
18	LD	M10				
19	MPS					
20	ANI	M11				
21	SET	Y10				
22	RST	Y11				
23	MPP					
24	AND	M11				
25	SET	Y11				
26	RST	Y10				
27	END					

8.4.6 将数据写入远程 I/O 站特殊功能模块 (MELSECNET)(G(P).RTOP)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Un : 为执行数据写入的特殊功能模块分配的主站的起始 I/O 号 (BIN16 位)

n1 : 存储写入数据的特殊功能模块的缓冲存储器的起始号 (BIN16 位)

Ⓢ : 存储写入数据的自站链接寄存器的起始号 (软元件名称)

n2 : 发送数据长度 (BIN16 位)

Ⓓ : 在指令结束时 ON 一个扫描软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	△*1	△*2	○	--				○	--
Ⓢ	--	△*3	--	--				--	--
n2	∧*1	∧*2	○	--				○	--
Ⓓ	∧*1	∧*2	○	--				--	--

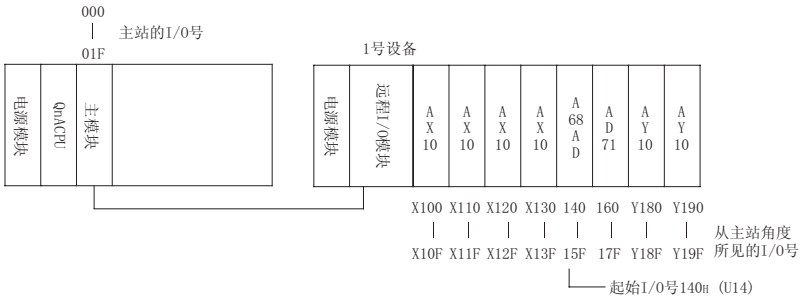
*1: 可以使用 X、Y、M、L、F、V 和 B。
*2: 可以使用 T、ST、C、D 和 W。
*3: 只可以使用链接寄存器。

要 点

1. 将从主站角度所见的特殊功能模块的起始 I/O 号指定到 Un。
   Un 中指定的特殊功能模块的起始 I/O 号是将从主站角度所见的特殊功能模块的起始 I/O 号以 4 位表示时的高 3 位进行设定的。
   例如，X/Y0140 的情况下其高 3 位为 014 因此指定为 “U14”。’

   示例

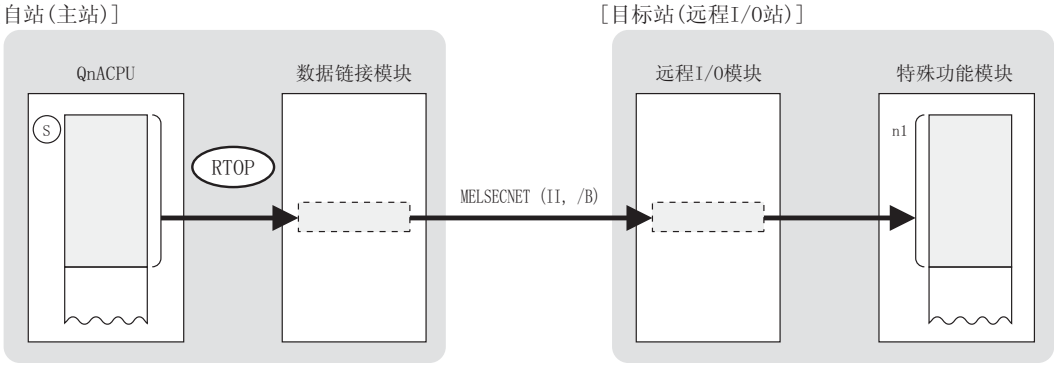
   假设网络参数 X100 到 X17F 和 Y140 到 Y19F 分配到 1 号远程 I/O 站。



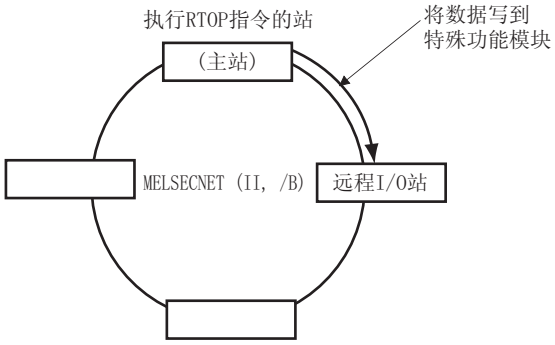
2. 局部软元件和各程序的文件寄存器均不能作为设定数据用的软元件使用。
   关于详细内容，请参考以下手册。
   • QnACPU 编程手册 (特殊功能模块篇)

功 能

(1) 在 MELSECNET(II, /B) 中，将自站（主站）的 ⓢ 中指定的软元件起 n2 点数的数据，存储至 Un 中指定的 I/O 号中分配的远程 I/O 站的特殊功能模块的 n1 中指定的缓冲存储器的后面。
当对远程 I/O 站的软元件数据的写入结束时， ⓓ 指定的结束软元件将 ON 一个扫描。



(2) 对远程 I/O 站的软元件数据的写入操作只能通过 MELSECNET(II, /B) 的主站进行。
此外，RTOP 指令的执行只能对占用 32 个 I/O 点的特殊功能模块进行。

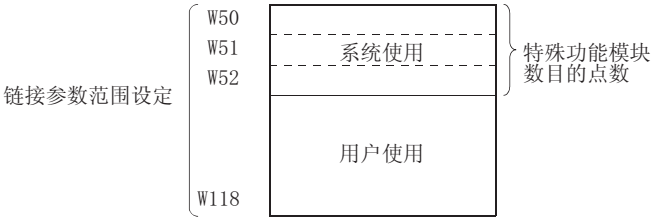


(3) 当特殊功能模块发生一个错误且 X(n+1)D 变为 ON 时，执行 RTOP 指令，YnD 将自动变为 ON，并且执行一个错误复位操作。
但是，在错误复位处理中，不能进行通讯处理。

(4) 关于 n1 指定的特殊功能模块缓冲存储器的附加信息，请参照特殊功能模块使用手册。

(5) n2 处的发送数据长度（点数）可设为 1 到 16。

(6) Ⓢ 指定的链接寄存器的范围在链接参数 “从主站到远程 I/O 站”的选择中设定。
而在链接寄存器中，从由链接参数选择中设定的 “从主站到远程 I/O 站”开始处到包含装在相关远程 I/O 站上的特殊模块个数为上的区域是系统保留使用的。
例如，连接参数中 “从主站→远程 I/O 站”设定为 W50 到 W118，则由于所讨论的安装在远程 I/O 站有 3 个特殊功能模块，W50 到 W52 将保留给系统使用。



(7) 对同一特殊功能模块，RFRP 和 RTOP 指令不能同时执行。
当在两个或多个位置同时执行指令的条件建立时，将自动处理握手操作，因此随后执行的 RFRP 和 RTOP 指令将不会被处理。
如果同一特殊功能模块希望执行多 RFRP/RTOP 指令，使用一个结束软元件以便他们将连续的执行。

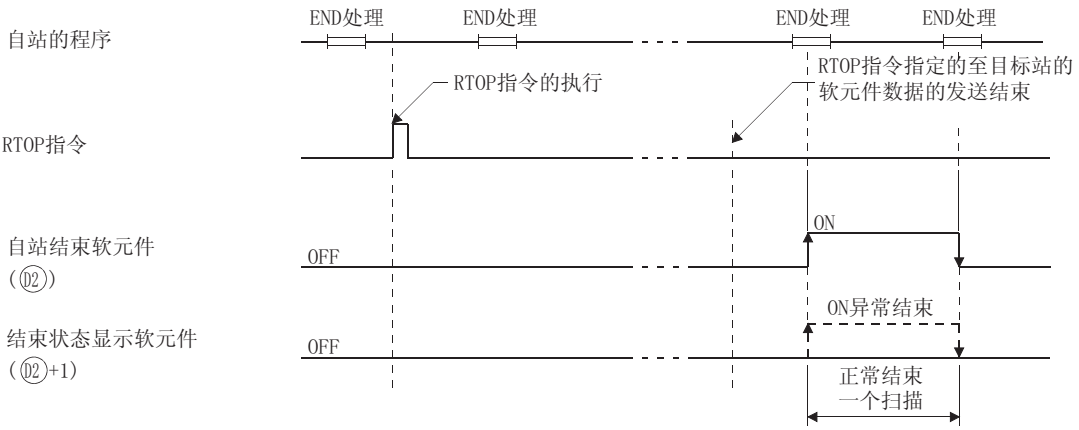
(8) RTOP 指令的正常 / 异常结束可以通过以下方式确认。

(a) 自站结束软元件 (Ⓓ2) : 通过 RTOP 指令执行的传送结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

(b) 结束时的状态显示软元件 (Ⓓ2+1) : 根据 RTOP 指令结束时的状态而 ON/OFF。

- 正常结束 : 保持 OFF 状态不变。
- 异常结束 : 在 RTOP 指令结束的扫描的 END 处理时变为 ON，在下一个 END 处理时变为 OFF。

[RTOP 指令执行时的自站的动作]



运算错误

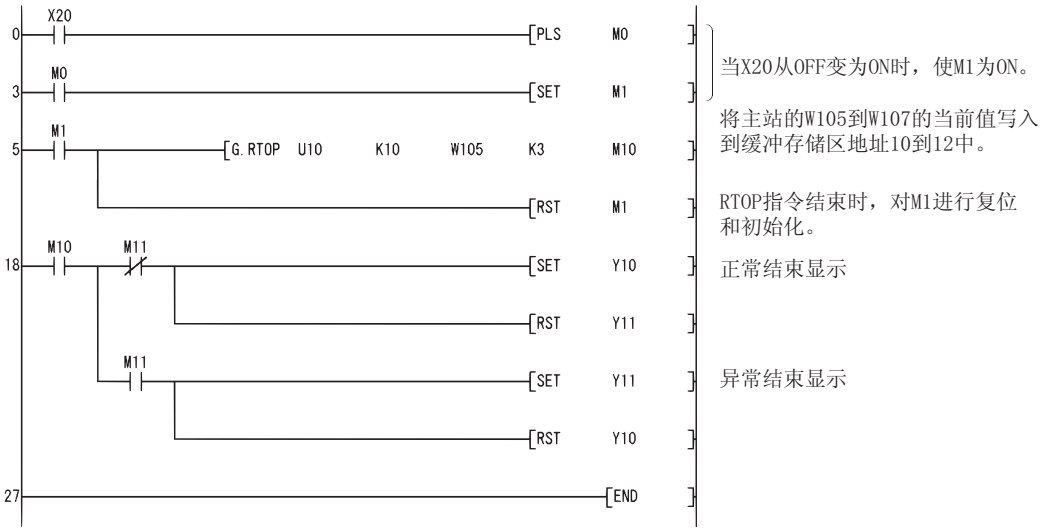
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- Un 指定的 I/O 号与远程 I/O 站的 I/O 号不一致。 (出错代码： 4102)
- Un 指定的 I/O 号不是一个特殊功能模块的起始 I/O 号。 (出错代码： 4102)
- ⓢ 指定的软元件起的 n2 点超过了相应软元件的最终软元件号。 (出错代码： 4101)
- n2 指定的点数超过了 1 到 16 的范围。 (出错代码： 4100)
- ⓓ1 指定的链接寄存器 (W) 的范围超过了网络参数的 “主站←远程 I/O 站”中设定的范围。 (出错代码： 4102)

程序示例

(1) 当 X20 为 ON 时，下列程序将主站的 W105 到 W107 的数据写入至 1 号远程 I/O 站的第 1 个特殊功能模块的缓冲存储区地址 10 ～ 12 中。

[梯形图模式]



[列表模式]

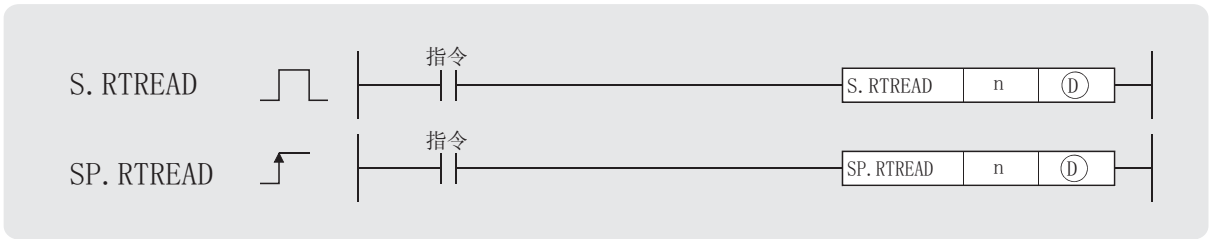
步	指令	软元件				
0	LD	X20				
1	PLS	M0				
3	LD	M0				
4	SET	M1				
5	LD	M1				
6	G.RTOP	U10	K10	W105	K3	M10
17	RST	M1				
18	LD	M10				
19	MPS					
20	ANI	M11				
21	SET	Y10				
22	RST	Y11				
23	MPP					
24	AND	M11				
25	SET	Y11				
26	RST	Y10				
27	END					

8.5 读 / 写路由信息

8.5.1 读路由信息 (S(P)/Z(P).RTREAD)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

(1) 使用高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU 时



n : 传送目标网络号 (1 到 239)(BIN16 位)

Ⓓ : 存储读取数据的软元件的起始号 (软元件名称)

用户

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n	○	○		--				○	--
Ⓓ	--	○		--				--	--

功 能

(1) 从由 n 指定目标网络读取数据，使用在路由参数中设定的路由信息，并且存储到从 Ⓓ 开始的软元件中去。

(2) 如果在路由参数中设定的 n 所指定的传输目标网络中没有数据，则将 0 存储到从 Ⓓ 开始的软元件中去。

(3) 从 Ⓓ 开始的软元件中存储的数据内容如下所示。

		(各数据的范围)
Ⓓ+0	中继网络号	(1到239)
+1	中继站号	(1到120)
+2	虚拟	

运算错误

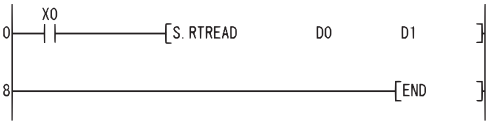
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- n 指定的数据不在 1 到 239 的范围内。 (出错代码 : 4100)
- Ⓓ 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU) (出错代码 : 4101)

程序示例

(1) 当 X0 为 ON 时，以下程序从 D0 指定网络号中读取路由信息。

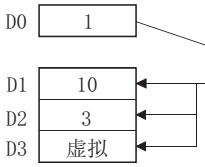
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	S.RTREAD	D0	D1
8	END		

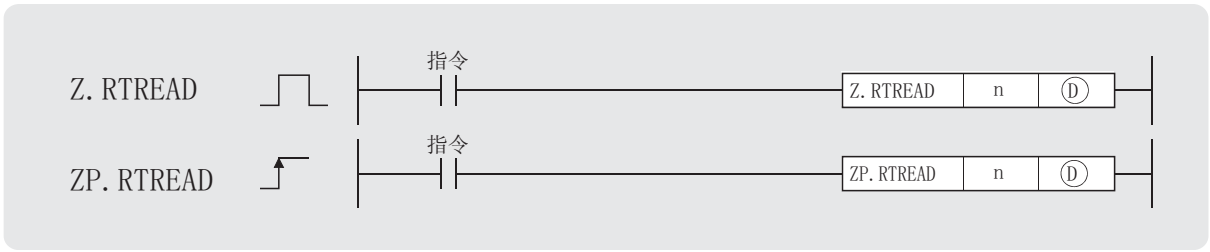
[动作]



[路由参数设定内容]

发送指定网络号	中继网络号	中继网络号
1	10	3
2	10	2
3	10	1

(2) 使用 QnACPU/Q4ARCPU 时



n : 传送目标网络号 (1 到 239)(BIN16 位)

Ⓓ : 存储读取数据的软元件的起始号 (软元件名称)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字 —				
n	○	○		--				○	--
Ⓓ1	--	○		--				--	--

功 能

(1) 从由 n 指定的目标网络中读取数据，使用在路由参数中设定的路由信息，并且存储到从 Ⓓ 开始的软元件中去。

(2) 如果在路由参数中设定的 n 所指定的传输目标网络中没有数据，则将 0 存储到从 Ⓓ 开始的软元件中去。

(3) 从 Ⓓ 开始的软元件中存储的数据内容如下所示。

(单个数据范围)

Ⓓ+0	继电器网络号	(1到239)
+1	继电器站号	(1到64)
+2	路由站号	(1到64)

运算错误

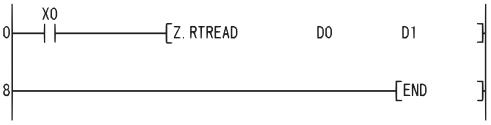
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• n 指定的数据不在 1 到 239 的范围内。 (出错代码 : 4100)

程序示例

(1) 当 X0 为 ON 时，以下程序从 D0 指定网络号中读取路由信息。

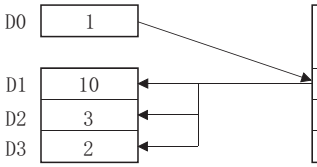
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	Z.RTREAD	D0	D1
8	END		

[动作]



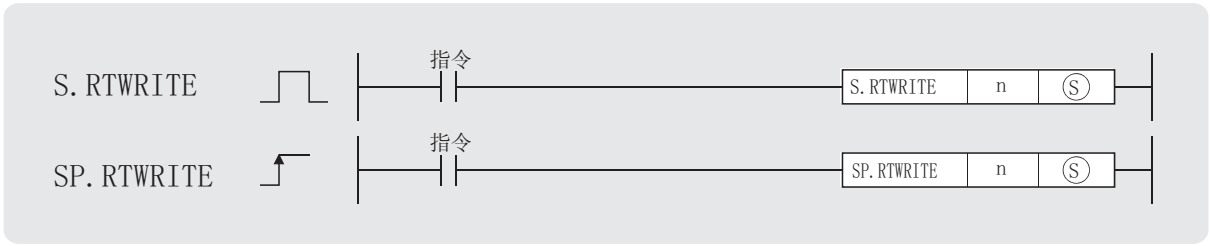
[路由参数设定内容]

	发送指定网络号	发送指定网络号	中继器站号	路由站号
D1	1	10	3	2
D2	2	10	2	2
D3	3	10	1	2

8.5.2 登录路由信息 (S(P)/Z(P).RTWRITE)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

(1) 使用高性能型 QCPU/ 过程控制 CPU/ 冗余 CPU/ 通用型 QCPU 时



n : 传送目标网络号 (1 到 239)(BIN16 位)

Ⓢ : 存储写入数据的软元件的起始号 (软元件名称)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n	○	○		--				○	--
Ⓢ	--	○		--				--	--

功 能

(1) 将 Ⓢ 后面的路由数据登录到路由参数的 n 中指定的传送目标网络号的区域中。

(2) 设定在 Ⓢ 后面的数据的内容如下所示。

(各数据的范围)

Ⓢ+0	中继网络号	(0到239)
+1	中继站号	(0到120)
+2	虚拟	

(3) 如果 n 指定的传送目标网络号在路由参数中设定，将被 Ⓢ 后面的数据所更新。

(4) 如果 Ⓢ 后面的数据 (Ⓢ+0 到Ⓢ+2) 均为 0，n 指定的传送目标网络号的数据将从路由参数中删除。

运算错误

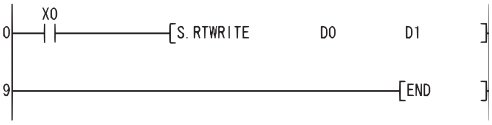
(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- n 指定的数据不在 1 到 239 的范围内。 (出错代码 : 4100)
- Ⓢ 后面的数据超过了各设定范围。 (出错代码 : 4100)
- 登记在网络参数的路由参数中的路由信息和登记在 RTWRITE 指令中的路由信息的总数超过了 64。 (出错代码 : 4100)
- Ⓢ 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码 : 4101)

程序示例

(1) 当 X0 为 ON 时，以下程序将 D1 到 D3 指定的路由信息写到 D0 指定网络号的网络模块中。

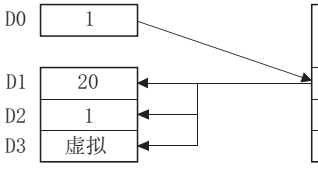
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	S.RTWRITE	D0	D1
9	END		

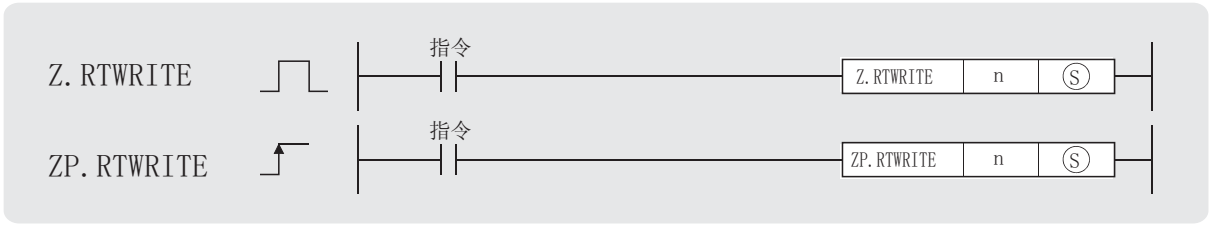
[动作]



[路由参数的设定内容]

传送目标网络号	中继网络号	中继站号
1	20	1
2	10	2
3	10	1

(2) 使用 QnACPU/Q4ARCPU 时



n : 传送目标网络号 (1 到 239)(BIN16 位)

Ⓢ : 存储写入数据的软元件的起始号 (软元件名称)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n	○	○		--				○	--
Ⓢ		○		--				--	--

功 能

(1) 将 Ⓢ 后面的路由数据登录到路由参数的 n 所指定的传送目标网络号的区域中。

(2) Ⓢ 后面的设定数据的内容如下所示。

		(各数据的范围)
Ⓢ+0	中继网络号	(0到239)
+1	中继站号	(0到64)
+2	路由站号	(0到64)

(3) 如果 n 指定的传送目标网络号在路由参数中设定，将被 Ⓢ 后面的数据所更新。

(4) 如果 Ⓢ 后面的数据 (Ⓢ+0 到Ⓢ+2) 均为 0，则 n 指定的传送目标网络号的数据将从路由参数中删除。

运算错误

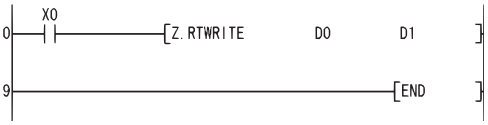
(1) 在以下发生运行错误情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- n 指定的数据不在 1 到 239 的范围内。 (出错代码 : 4100)
- 从 Ⓢ 开始的数据超过了单个设定范围。 (出错代码 : 4100)

程序示例

(1) 当 X0 为 ON 时，以下程序将 D1 到 D3 指定的路由信息写到 D0 指定网络号的网络模块中。

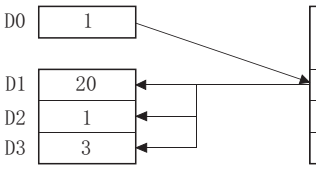
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	X0	
1	Z.RTWRITE	D0	D1
9	END		

[动作]



[路由参数的设定内容 g]

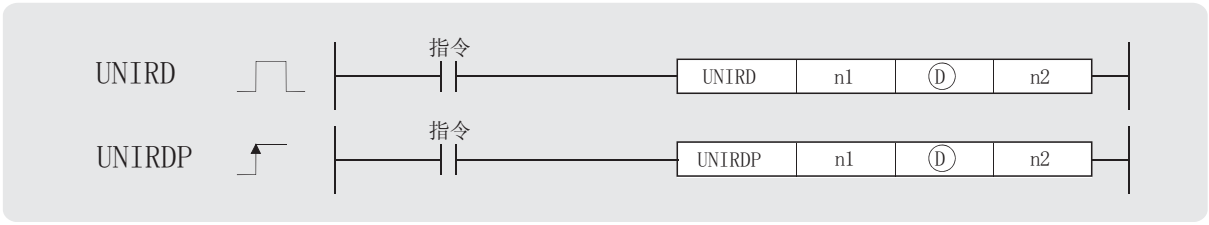
传送目标网络号	中继网络号	中继站号	路由站号
1	20	1	3
2	10	2	2
3	10	1	2

9 QCPU 指令

分类	详细处理内容	参考章节
读取模块信息	读取指定模块状态的详细信息。	9.1 节
追踪置位 / 复位	触发和复位采样追踪。	9.2 节
写入数据到指定文件	写入数据到指定的文件中。	9.3 节
从指定的文件中读取数据	从指定的文件中读取数据。	9.4 节
写入数据到标准 ROM	写入数据到标准 ROM 中的软元件数据存储文件。	9.5 节
从标准 ROM 中读取数据	从标准 ROM 中的软元件数据存储文件中读取数据。	9.6 节
从内存卡装载程序	将存储在内存卡或标准 ROM 中的程序传送到程序存储器中使其处于备用状态。	9.7 节
从程序内存中卸载程序	删除程序内存中的备用状态程序。	9.8 节
装载 + 卸载	从程序内存中卸载程序然后从内存卡装载程序。	9.9 节
文件寄存器的高速块传输	批量高速传送指定数据。	9.10 节
写入到自站的 CPU 共享内存	将自站 CPU 的软元件数据写入到自站 CPU 模块的 CPU 共享内存。	9.11 节
从另一个站的 CPU 共享内存中读取	从另一个站的 CPU 模块的 CPU 共享内存读取软元件数据到自站 CPU。	9.12 节
选择刷新指令	自动刷新 CPU 模块的 CPU 共享内存。	9.13 节
扩展时钟指令	扩展时钟数据的读取、加法 / 减法。	9.14 节

9.1 读取模块信息 (UNIRD(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n1： 将从模块信息被读取的模块开始的起始 I/O 号整除 16 所获得的值 (0 到 FFn) (BIN 16 位)
Ⓓ: 储存模块信息的软元件起始号 (软元件名称)
n2： 读取数据的点数 (0 到 256) (BIN 16 位)

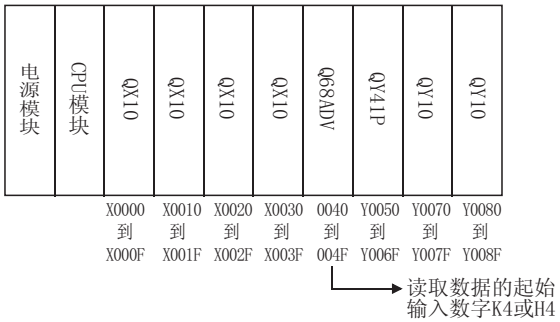
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	○	○		--				○	--
Ⓓ	--	○		--				--	--
n2	○	○		--				○	--

功 能

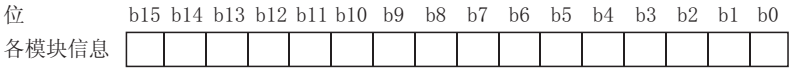
(1) 从 n1 指定的模块中读取尽可能多的 n2 指定的模块信息，(通过将起始 I/O 号整除 16 获得该值)，同时在软元件里存储该信息，然后由 Ⓓ 指定。
(读取实际安装模块的状态而不是通过 I/O 分配指定的模块类型。)

备 注

当它用 16 进位制符号表达成 4 位数时，n1 的值由读取模块信息的插槽中的起始 I/O 号的前三位数来指定。



模块信息细节描述如下：



位	项目	含义	
b0	I/O 点的数目	000：16 点	001：32 点
b1		010：48 点	011：64 点
b2		100：128 点	101：256 点
		110：512 点	111：1024 点
b3	模块类型	000：输入模块	
b4		001：输出模块	
b5		010：I/O 混合模块	
		011：智能功能模块	
b6	外部电源状态（为将来扩充）	1：外部电源已连接。0：外部电源未连接	
b7	保险丝熔断的有 / 无	1：有保险丝熔断	0：正常
b8	在线模块更换状态 / 从待机系统执行	1：正在进行在线模块更换，或试图通过冗余系统中的待机系统读取扩展基板上的模块信息。*1 0：除以上之外。	
b9	轻微 / 中等错误状态	1：出现轻微 / 中等错误	0：正常
b10	模块错误状态	00：没有模块错误	01：轻微错误
b11		10：中度错误	11：严重错误
b12	模块备用状态	1：正常	0：有模块错误
b13	空位	固定为 0	
b14	A/Q 模块	1：A 系列模块	0：Q 系列模块
b15	模块安装状态	1：模块已经安装	0：模块没有安装

*1：在其它 CPU 控制的模块的在线模块更换过程中，多 CPU 系统中使用的通用型 QCPU 仍将处于 ON 状态。

运算错误

(1) 在以下出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

[高性能型 QCPU、过程 CPU、冗余 CPU 和通用型 QCPU]

- 当 n1 不属于 0 到 FF_H （出错代码： 4100）
- 当 n2 不属于 0 到 256 （出错代码： 4100）
- 当 n1 和 n2 等于或远大于 257 （出错代码： 4100）

[Q00/Q01CPU]

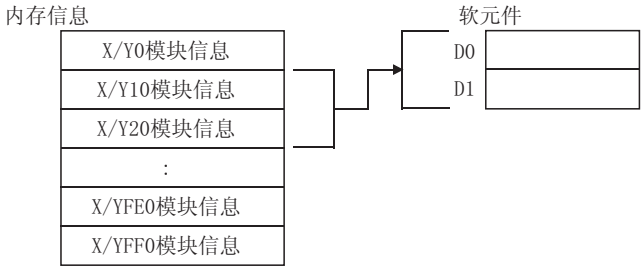
- 当 n1 不属于 0 到 $3F_H$ （出错代码： 4100）
- 当 n2 不属于 0 到 64 （出错代码： 4100）
- 当 n1 和 n2 等于或远大于 65 （出错代码： 4100）

[Q00JCPU]

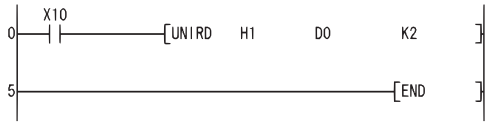
- 当 n1 不属于 0 到 F_H （出错代码： 4100）
- 当 n2 不属于 0 到 16 （出错代码： 4100）
- 当 n1 和 n2 等于或远大于 17 （出错代码： 4100）

程序示例

(1) 当 X10 为 ON 时，下列程序从 D0 开始，在 I/O 号码 10H 到 20H 存储模块信息。



[梯形图模式]

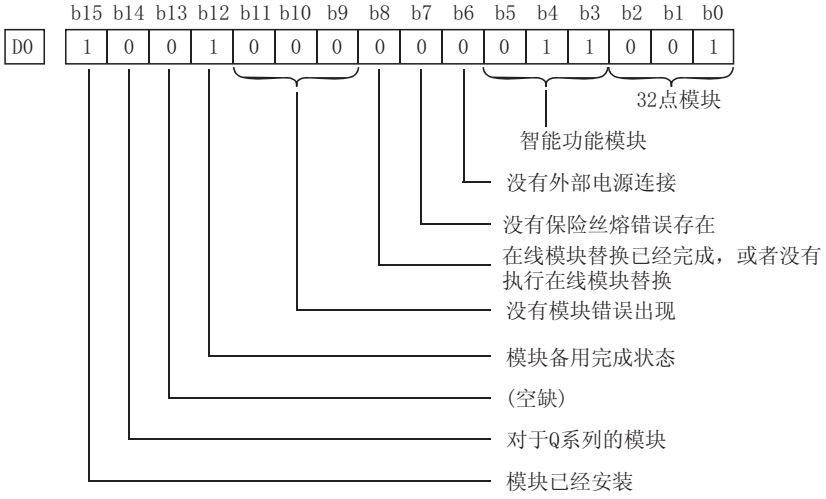


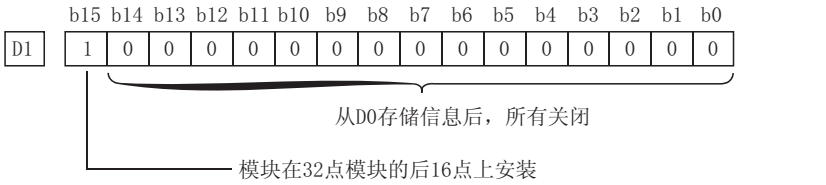
[列表模式]

步	指令	软元件		
0	LD	X10		
1	UNIRD	H1	D0	K2
5	END			

读出结果（当从 D0 读时）

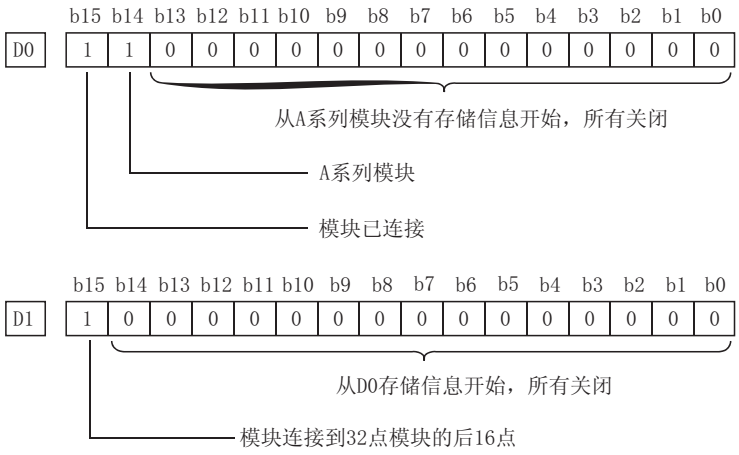
(a) 对于 Q 系列的 32 点智能功能模块





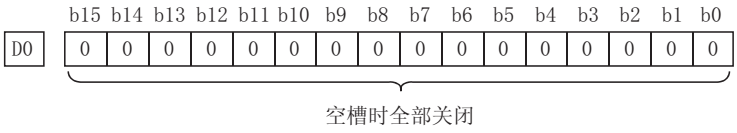
• 使用 48 或 64 点模块，D1 的相同内容被分别存储在 D2 或 D2 和 D3 之间。

(b) 对于 A 系列的 32 点模块

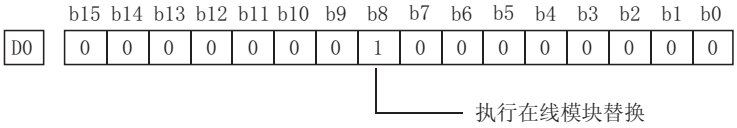


• 使用 48 或 64 点模块，D1 的相同内容被分别存储在 D2 或 D2 和 D3 之间。

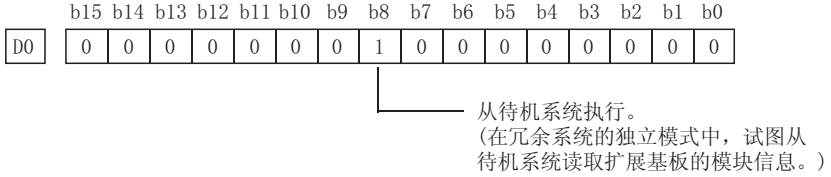
(c) 空槽



(d) 执行在线模块替换

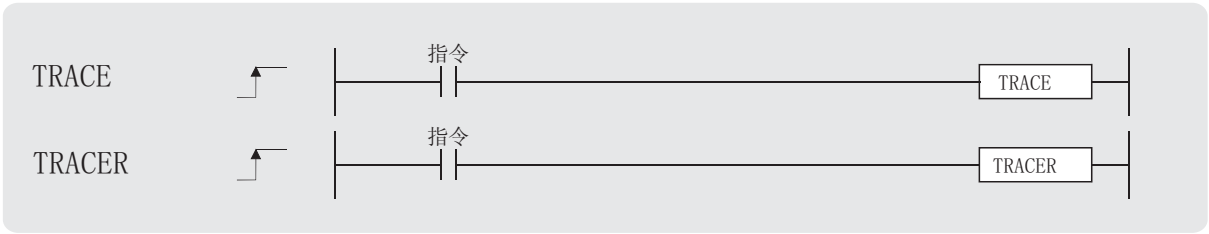


(e) 在冗余系统的独立模式中，试图从待机系统读取扩展基板的模块信息。



9.2 追踪置位 / 复位 (TRACE、TRACER)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

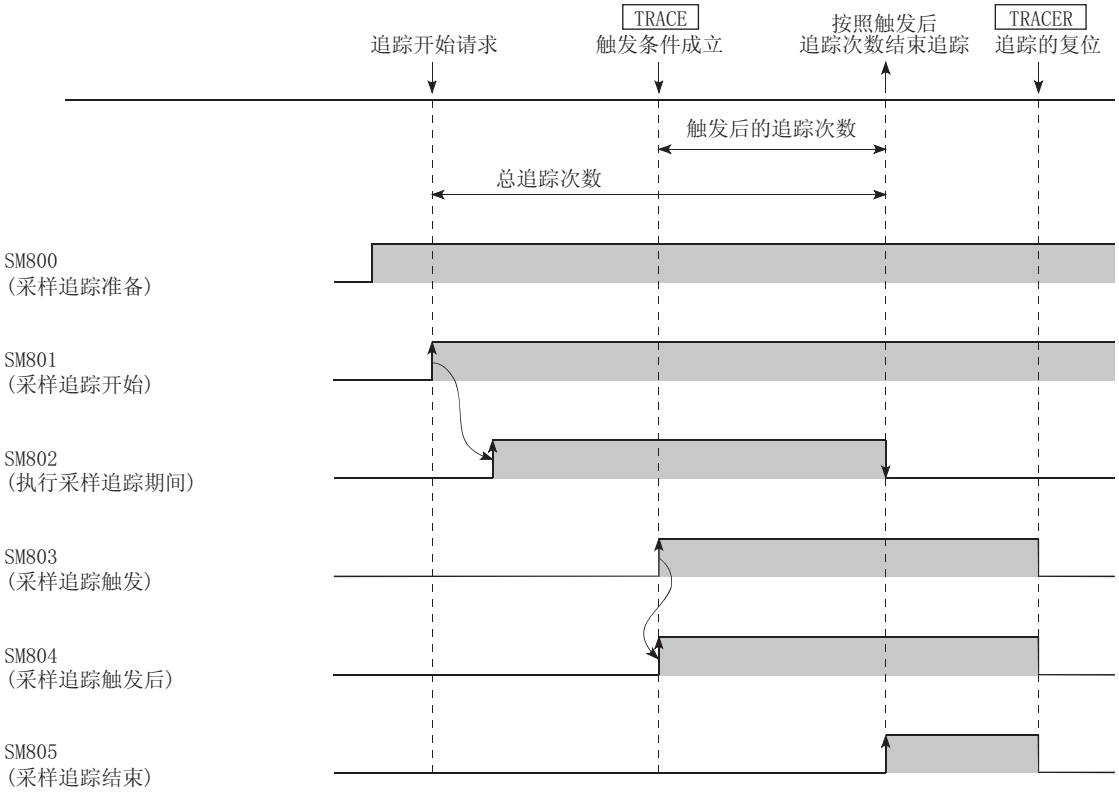


设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

采样追踪功能是在指定时间连续收集 CPU 模块的指定软元件数据的功能。

当 SM800、SM801 和 SM802 为 ON 时，通过采样样本追踪功能，从追踪运算的指定次数得出的追踪结果被存储到内存卡的采样追踪文件中。



TRACE

(1) TRACE 指令将 SM803 变为 ON，按照追踪条件设定中的 “触发后次数”中设定的次数执行采样后，锁存数据并停止采样追踪。

(2) 在追踪执行的过程中，如果 SM801 为 OFF，取样结束。

(3) 在 TRACE 指令执行和追踪结束后，SM805 为 ON。

(4) 如果 TRACE 指令已经执行，第二个及以后的 TRACE 指令将被忽略。
当 TRACE 指令执行时，TRACE 指令能够再次执行。

TRACER

(1) TRACER 指令是用于复位 TRACE 指令的指令。
当执行了 TRACER 指令后，则可以再次执行 TRACE 指令。

(2) 当 TRACER 指令执行后，SM803 到 SM805 为 OFF。

备 注

1. 在使用追踪操作过程中，可以参考以下手册：
   • CPU 模块的用户手册（功能解说 / 程序基础篇）
2. 在使用 GX Developer 执行追踪时，可以参考 GX Developer 的操作手册。

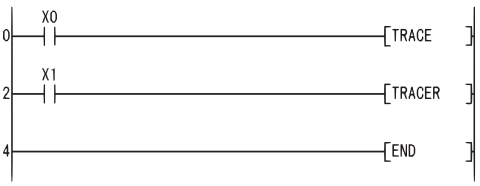
运算错误

(1) 没有与 TRACE 或 TRACER 指令相关的运算错误。

程序示例

(1) 当 X0 为 ON 时，下列程序执行 TRACE 指令，当 X1 为 ON 时，可以用 TRACER 指令复位 TRACE 指令。

[梯形图模式]

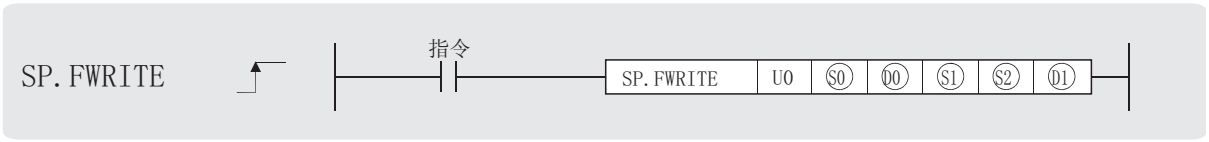


[列表模式]

步	指令	软元件
0	LD	X0
1	TRACE	
2	LD	X1
3	TRACER	
4	END	

9.3 写数据到指定的文件 (SP.FWRITE)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	
S0	○	○		--				○	--	--
D0	--	○		--				--	--	--
S1	--	○		--				--	--	--
S2	--	○		--				--	○	--
D1	∧*1	∧*1		--				--	--	--

*1：本地软元件和指定用于各程序的软元件不能使用。

○ 设定数据 / 控制数据

设定数据	含义			设定范围	设定者	数据类型
U0	虚拟			--	--	
S0	驱动指定			2	用户	
D0	存储控制数据的软元件的起始号。控制数据如下所示。					
	软元件	项目	含义 / 设定数据	设定范围	设定者	
	D0	执行 / 完成类型	指定执行类型。 0000H ：二进制数据写入 0100H ：在 CSV 格式转换后写入数据	0000H 0100H	用户	
	D0+1	（未使用）	由系统使用。	--	系统	
	D0+2	写入结果（已写数据数）	包含相对于 S2 指定数据的实际写入数据的数目。数值单位取决于 D0+7 的数据类型。	--	系统	
	D0+3	（未使用）	--	--	--	
	D0+4 D0+5	文件的位置	当 D0 中指定了二进制数据写入时，设定文件位置。 00000000H ：从文件的起始开始 00000001H 到 FFFFFFFEH：从指定的位置开始。值的单位取决于数据类型指定。 FFFFFFFFH ：附加在文件的最后。 D0 中指定 CSV 格式写入时 • 对于序列号的前五位是 01111 或以前的高性能型 QCPU，必须设置为文件的起始 (0H)。 • 对于序列号的前五位是 01112 或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU，需设置文件的位置。 00000000H 到 FFFFFFFEH：从文件的起始开始。 FFFFFFFFH ：附加在文件的最后。	00000000H 到 FFFFFFFFH	用户	BIN 16 位

设定数据	含义			设定范围	设定者	数据类型
(D0)	(D0)+6	列数指定	在 (D0) 中指定了二进制写入时必须设置为 0。 在 (D0) 中指定了 CSV 格式写入时，设置进行写入的列数 0 : 没有列，默认为一行。 除了 0 以外：设置指定的列数。	0H 到 FFFFH (0 到 65535)	用户	BIN 16 位
	(D0)+7	数据类型指定	0：字 1：字节	0、1	用户	
(S1)	存储文件名称的软元件的起始号。文件名称如下所示：					
	软元件	项目	含义 / 数据设定	设定范围	设定者	
	(S1) 到 (S1)+□	文件名称字符串	指定文件名称的字符串。 • 当省去扩展名时，也省掉“.”（句点）。 • 限制文件名为 8 个字符 + 句点 +3 个字符以内。 • 当使用了 9 个或者更多字符时，不论扩展名存在与否都将被省略，扩展名变为 BIN 或 CSV。	字符串	用户	
(S2)	存储数据的软元件的起始号。写入的数据如下所示：					
	软元件	项目	含义 / 设定数据	设定范围	设定者	
	(S2)	请求写入的数据数目	指定请求写入的数据数目（以字为单位）。 在 (D0)+7 中指定字节时，也应进行字换算，以字单位进行指定。	1 到 480 1 到 32767 *2	用户	
	(S2)+1 到 (S2)+□	写入数据	请求写入的数据。	0000H 到 FFFFH		
(D1)	在指令完成时变为 ON 的位软元件。（但是，异常结束时 (D1)+1 也将为 ON。）					位
	软元件	项目	含义 / 设定数据	设定范围	设定者	
	(D1)	结束信号	指示指令的完成。 ON：结束 OFF：未结束	--	系统	
	(D1)+1	异常结束信号	表明指令是正常结束或是异常结束。 ON：异常结束 OFF：正常结束	--		

*2：表示只适用于通用型 QCPU 的范围。

注意事项

(1) 在 ⓢ⓪ (驱动指定)，只可以设置 ATA 卡驱动 (2)。
注意当 FLASH 卡装载时，SP.FWRITE 指令不能执行写入的功能。
不能设置 SRAM 卡，标准 RAM 或标准 ROM 驱动。

(2) 对于 CSV 设置，已写数据是十进制值。

例如　字符 “A”(41H) → “65”被写入
处理范围 : -32768 到 32767

(3) 对于二进制写入，字指定中文件位置的设置范围是从 00000000H 到 7FFFFFFFH 和 FFFFFFFFH。

☆功 能

(1) 指定的数据数目写入到指定的文件。

在控制数据中设置执行 / 结束类型，从而指定是否不通过任何转换写入二进制数据或者在写入之前转换二进制数据成 CSV 格式数据。

（写入目标只有 ATA 卡）

(2) 在检测到 FWRITE 的指令结束和 END 指令执行后，结束信号位软元件（(D1)+1）自动为 ON。在结束指令执行后开始下一个扫描时，位软元件为 OFF。

对于 FWRITE 指令使用位软元件作为执行结束的标志。

当 FWRITE 指令异常结束时，错误结束软元件（ (D1) +1）与执行结束软元件（ (D1) +0）同步 ON/OFF。对于 FWRITE 指令使用这个软元件作为出现错误结束的标志。

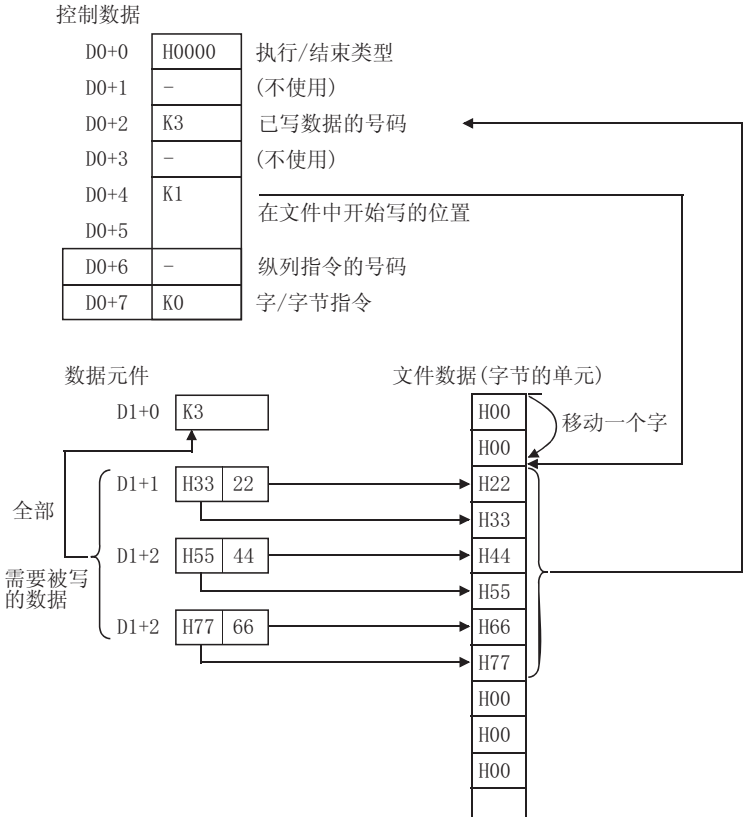
SM721 在指令执行时是为 ON。

当 SM721 为 ON 时，FWRITE 指令不能执行。（如果尝试，没有执行处理）

当一个错误在指令执行之前被检测到（在 SM721 为 ON 之前），执行结束软元件（ (D1) +0），错误结束软元件（ (D1) +1）和 SM721 不变为 ON。

(3) 应将数据的请求写入数据数（(S2)）和文件位置（(D0)+4 和 (D0)+5）的处理单位设置为字单位。

如下实例表示当指定了请求写入数据数和文件位置时，写入二进制数据的方法。



(4) 当写入二进制数据时

(a) 如果目标文件的扩展名是省略的，使用 “BIN”作为扩展名。

(b) 当指定文件不存在时，将建立一个新的文件，添加数据，同时从文件的开始部分存储。
新文件的属性使用档案属性来设定。

(c) 当存储数据的大小超过写入的过程中文件里面存在的区域，多余的数据将被添加 / 存储。

(d) 如果指定的文件位置大于存在文件的尺寸：

- 高性能型 QCPU 的系列号的前五位数字是 “01111”或更小将会出现错误。
- 对于序列号的前五位数字是 “01112”或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU，将变为以 0 点执行写入同时正常地结束。

(e) 当数据添加 / 存储，缺乏足够的空间时，错误将出现。
在这种情况下，成功添加 / 存储的数据仍然保留在中间存储器中。
在尽可能多的数据被添加 / 存储后，错误结束将指示。

(5) 在 CSV 格式转换后写入数据

(a) 如果扩展名省略，可以使用 “CSV”作为扩展名。

(b) 当存在的文件被指定时：
[系列号的前 5 位数是 “01111”或更早的高性能型 QCPU]
文件内容将被全部删除同时数据从开始处存储。
[序列号的前 5 位数是 “01112”或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU]

- 当不是 FFFFFFFFH 设置在 (Ⓓ+4、 Ⓓ+5) 时，文件内容将被全部删除同时数据从开始处存储。
- 当 FFFFFFFFH 设置在 (Ⓓ+4、 Ⓓ+5) 时，数据将从文件的末尾开始存储。

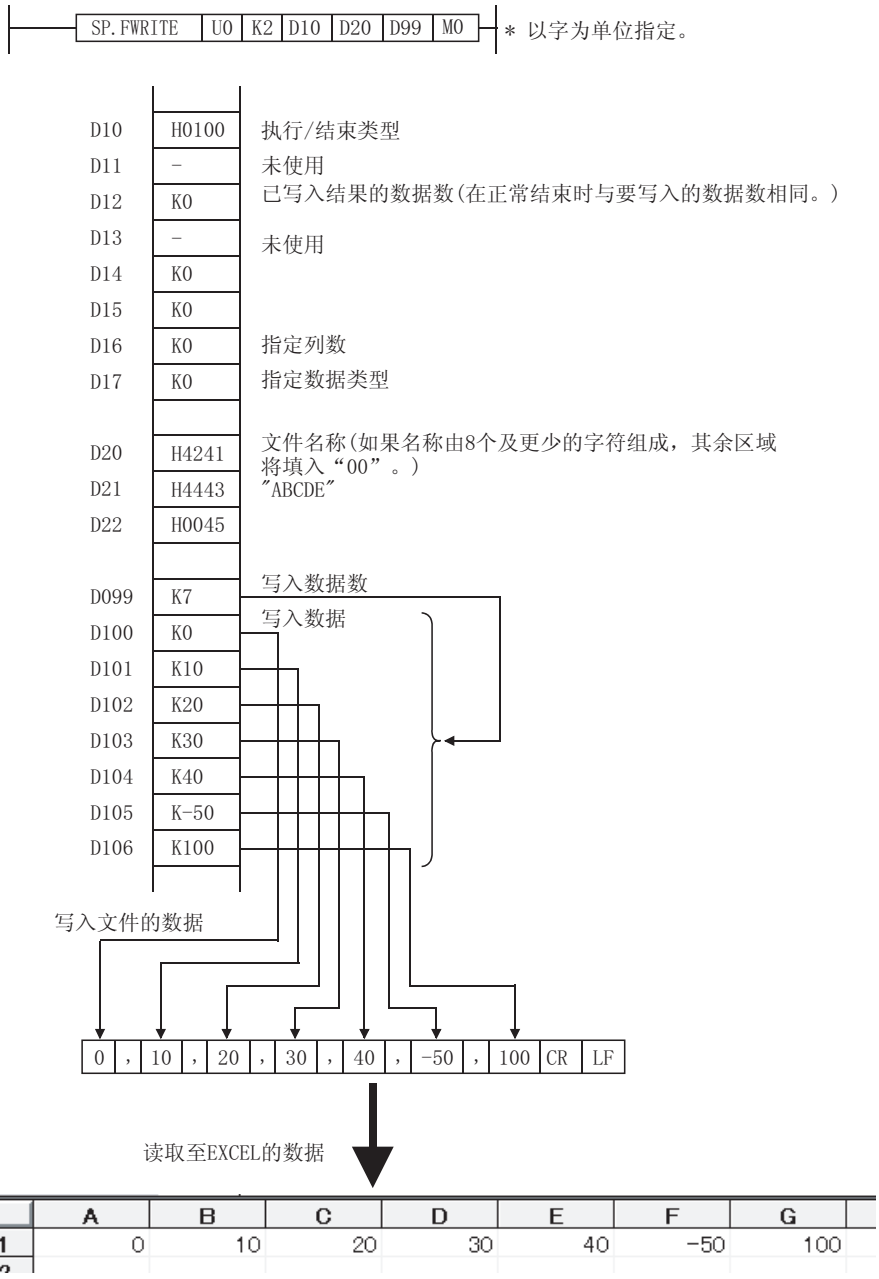
(c) 当指定文件不存在时，将建立一个新的文件，数据从文件的开始部分添加 / 存储。
新文件的属性使用档案属性来设定。

(d) 当数据添加 / 存储，缺乏足够的空间时，错误将出现。
在这种情况下，成功添加 / 存储的数据仍然保持在中间存储器中。
在尽可能多的数据被添加 / 存储后，错误结束将指示。

(e) 当列的指定列数是 “0”时，数据作为单行数据存储在CSV格式文件中。
以下是这种情况的一个实例：

示例

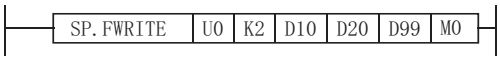
当数据在CSV格式转换后写入时，同时指定的列数是不为 “0”：



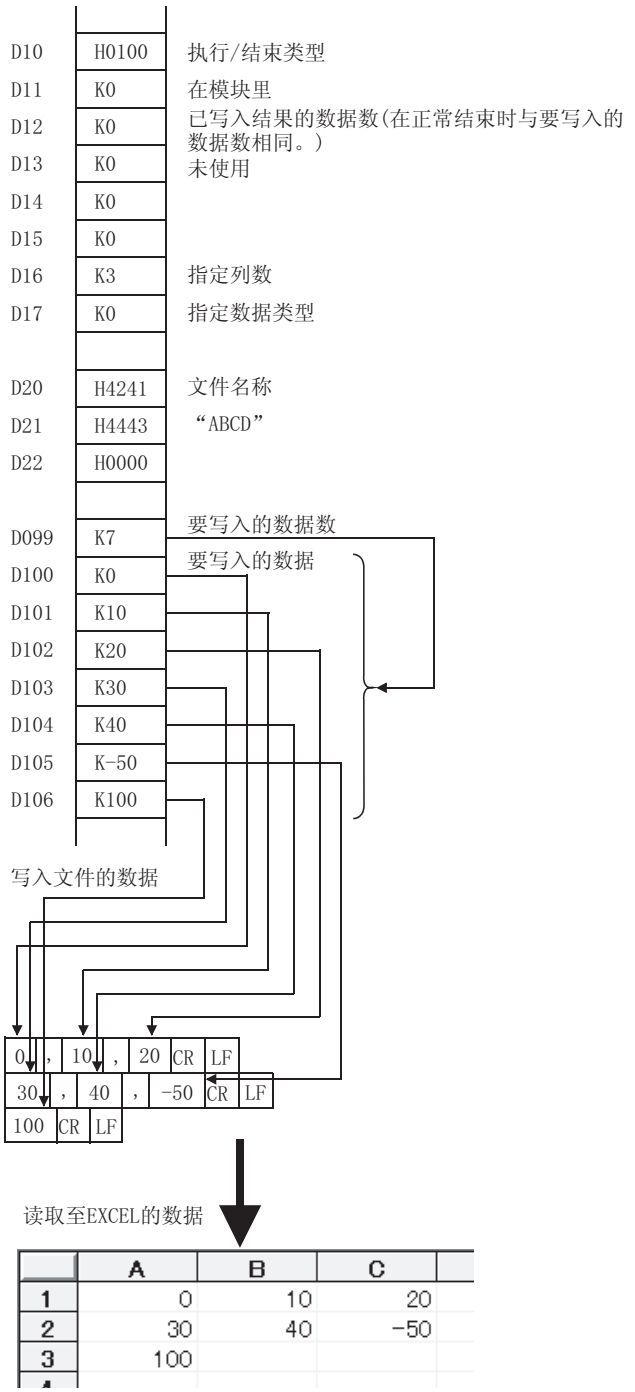
(f) 当数据在 CSV 格式转换后写入，同时指定的列数不是 “0”，数据作为表格数据以指定的列数存储在 CSV 模格式的文件里面。
以下是该情况的一个实例：

示例

当数据在 CSV 格式转换后写入，同时列的指定号码不是 “0”。



＊ 以字为单位指定。

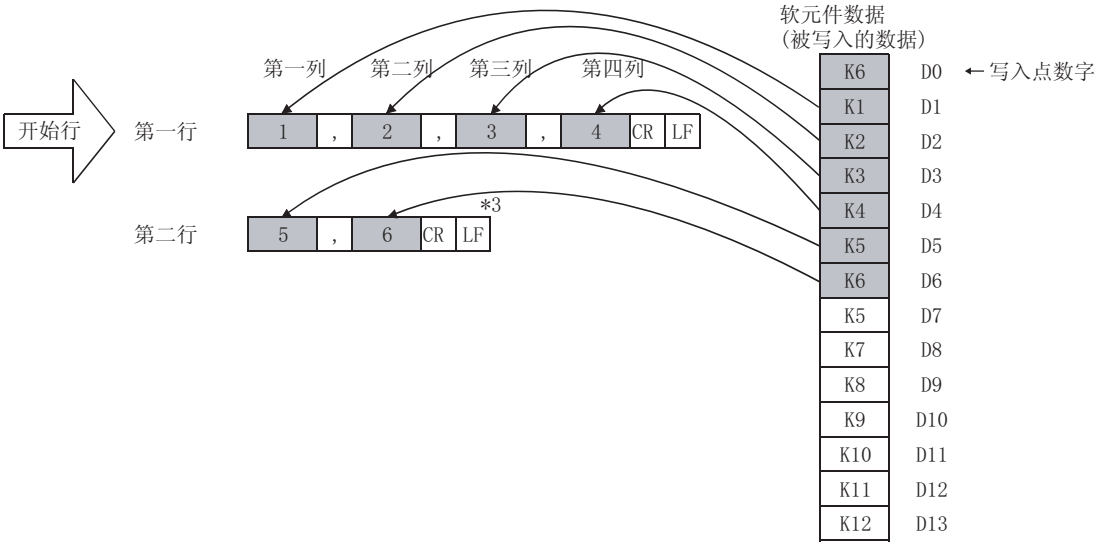


	A	B	C
1	0	10	20
2	30	40	-50
3	100		

(g) 在序列号的前五位数是 01112 或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU 中添加数据时的情况如下所示。

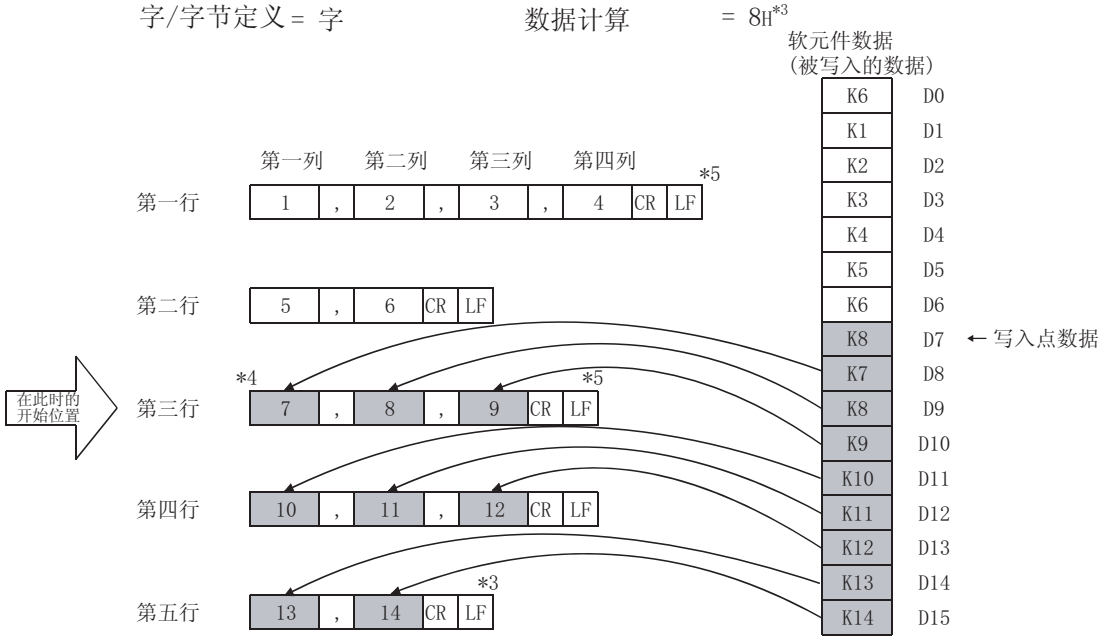
[指定文件到数据被写入的的位置](如果一个文件存在，删除它并重新创建一个文件。)

执行类型 = CSV格式　　文件位置 = 0H(创建新的文件)
列指定 = 4H*3*5　　写入起始软元件 = D0
字/字节指定 = 字　　数据计算 = 6H*3



[在添加模式中，在文件的末尾处添加]

执行类型 = CSV格式　　文件位置 = FFFFFFFFH(连续模式)
列指定 = 3H*3*5　　写入起始软元件 = D7
字/字节定义 = 字　　数据计算 = 8H*3



*3: 除非写入点的数字设定成 “列指定” 的整倍数，列的数目是随机的。
*4: 由于最后的数据总是紧随行供给代码，添加通常在添加模式中开始于新一行的开始部分。
*5: 在添加模式中，如果 “列指定” 从先前写入的模式中改变，列数将改变。

(h) 在中断程序中不要执行 SP.FWRITE 指令。
(如果执行，将无法保证正常运行。)

运算错误

(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

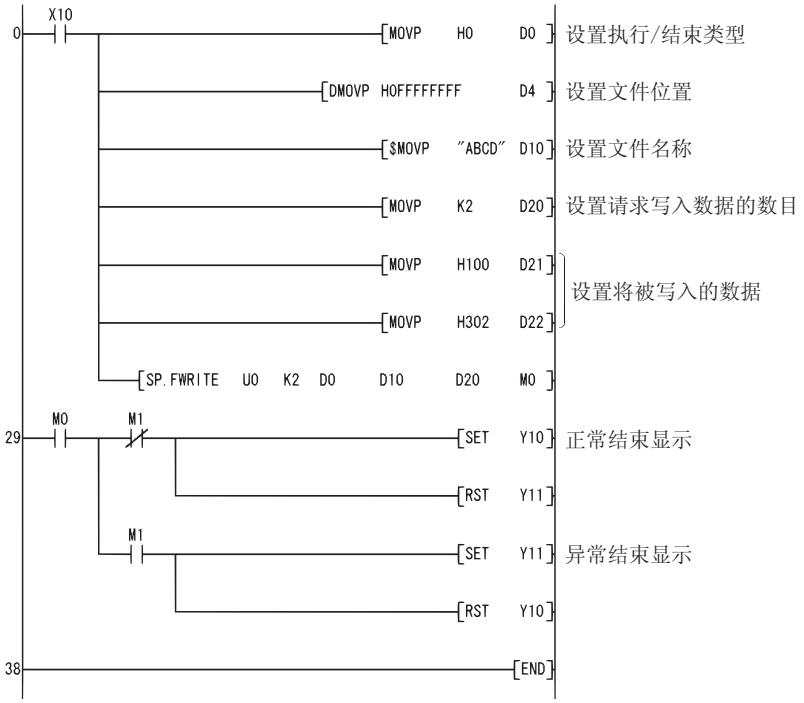
- 通过驱动指定软元件 (S0) 指定的驱动包含了除 ATA 卡外的中间存储器。 (出错代码： 4100)
- 设置在控制数据 (D0) 以后的值超出了设置范围。 (出错代码： 4100)
- “请求写入数据的数目”((S2)) 中指定的值超出了设置范围，或超出了 ((S2)+1) 后面的软元件范围。 (出错代码： 4101)
- ATA 卡的空余空间不足。 (出错代码： 4100)
- 在试图创建新文件时，无空余的空间。 (出错代码： 4100)
- 指定了无效的软元件。 (出错代码： 4004)
- ATA 卡出现访问异常。 (出错代码： 4100)
- 文件名称 ((S1)) 设定的值无法使用。 (出错代码： 4100)
- 文件名称 ((S1)) 的属性为 “只读”。 (出错代码： 4100)
- 由 (D0) 或 (D1) 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)

程序示例

(1) 当 X10 为 ON 时，下列程序添加二进制数据 (H00、H01、H02、H03) 的 4 个字节到插入的驱动 2 的内存卡的文件 “ABCD.BIN” 中。

• 假设来自 (D0) 的 8 个点为控制数据软元件保留。

[梯形图模式]



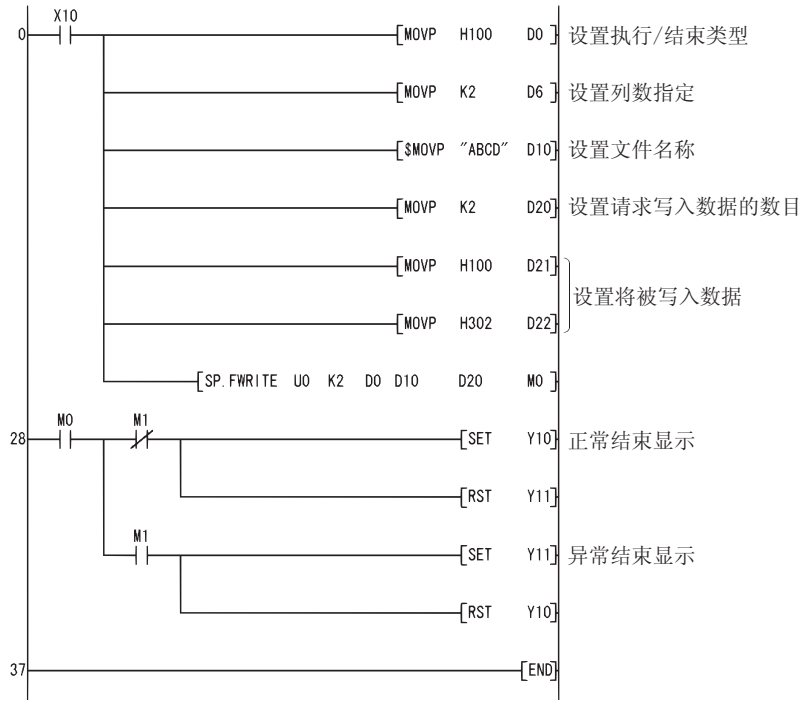
[列表模式]

步	指令	软元件					
0	LD	X10					
1	MOVP	H0	D0				
3	DMOVP	H0FFFFFFFF	D4				
6	$MOVP	"ABCD"	D10				
11	MOVP	K2	D20				
13	MOVP	H100	D21				
15	MOVP	H302	D22				
17	SP.FWRITE		U0	K2	D0	D10	D20	M0
29	LD	M0					
30	MPS						
31	ANI	M1					
32	SET	Y10					
33	RST	Y11					
34	MPP						
35	AND	M1					
36	SET	Y11					
37	RST	Y10					
38	END						

(2) 当 X10 为 ON 时，下列程序将四个字节 (00H、01H、02H 和 03H) 作为两列的 CSV 格式文件，以文件名 “ABCD.CSV” 创建到驱动 1 中安装的内存卡中。

• 写入的文件如下所示：

[梯形图模式]



[列表模式]

步	指令	软元件						
0	LD	X10						
1	MOVP	H100	D0					
3	MOVP	K2	D6					
5	$MOVP	"ABCD"	D10					
10	MOVP	K2	D20					
12	MOVP	H100	D21					
14	MOVP	H302	D22					
16	SP.FWRITE		U0	K2	D0	D10	D20	M0
28	LD	M0						
29	MPS							
30	ANI	M1						
31	SET	Y10						
32	RST	Y11						
33	MPP							
34	AND	M1						
35	SET	Y11						
36	RST	Y10						
37	END							

• 在 (D0) 中预留 8 点，作为控制数据用软元件。

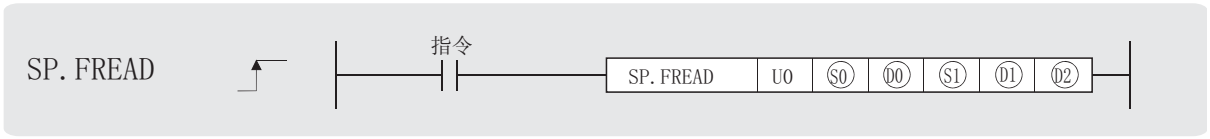
0	,	1	,	CR	LF	被写入文件的内容
2	,	3	,	CR	LF	

由EXCEL读取的数据

	A	B
1	0	1
2	2	3

9.4 从指定文件中读取数据 (SP. FREAD)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



设定数据	内部软元件		R、RZ	J□\□		U□\G□	Zn	常数		其它
	位	字		位	字			K、H	$	
(S0)	○	○		--				○	--	--
(D0)	--	○		--				--	--	--
(S1)	--	○		--				--	--	--
(D1)	--	○		--				--	○	--
(D2)	△*1	△*1		--				--	--	--

*1：本地软元件和指定用于各程序的软元件不能使用。

○ 设定数据 / 控制数据

设定数据	含义			设定范围	设定者	数据类型
U0	虚拟			--	--	BIN 16 位
(S0)	驱动指定			2	用户	
(D0)	存储控制数据的软元件的起始号。 控制数据如下所示。					
	软元件	项目	含义 / 设定数据	设定范围	设定者	
	(D0)	执行 / 完成类型	指定执行类型。 0000H：读取二进制数据。 0100H：在 CSV 格式转换后读取数据。	0000H 0100H	用户	
	(D0)+1	（未使用）	由系统使用。	--	系统	
	(D0)+2	请求读取数据数	指定希望读取的数据数目。 （单位：字） 即使在 (D0)+7 中的数据类型指定中指定了字节时，也应换算为字位单位（16 位）后进行指定。	1 到 480 1 到 32767*2	用户	
	(D0)+3	（未使用）	--	--	--	

*2：表示只适用于通用型 QCPU 的范围。

设定数据	含义			设定范围	设定者	数据类型
(D0)	(D0)+4 (D0)+5	文件位置	在 (D0) 中指定了二进制数据读取时，指定文件中开始读取的位置。 00000000H：从文件的起始开始 00000001H 到 FFFFFFFEH ：从指定的位置开始。 值的单位取决于数据类型指定。 FFFFFFFFH：不能设置。 在 (D0) 中指定了 CSV 格式读取时 • 对于序列号的前五位是 01111 或以前的高性能型 QCPU，必须设置为文件的起始(0H)。 • 对于序列号的前五位数字是 01112 或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU，应设置为文件位置（行）。 00000000H：从文件的起始位置读取数据。 00000001H 到 FFFFFFFEH ：从指定的行开始读取数据。 FFFFFFFFH：从先前读取的位置开始，继续读取数据。	00000000H 到 FFFFFFFFH	用户	BIN 16 位
	(D0)+6	列数指定	在 (D0) 中指定了二进制读取时必须设置为 0。 在 (D0) 中指定了 CSV 格式读取时，设置进行读取的列数。 0：没有列，视为一行。 除了 0 以外：视为指定数的列。	0H 到 FFFFH (0 到 65535)	用户	
	(D0)+7	数据类型指定	0：字 1：字节	0、1	用户	
(S1)	存储文件名称的软元件的起始号。文件名称如下所示：					
	软元件	项目	Contents/ 设定数据	设定范围	设定者	
	(S1) 到 (S1)+□	文件名称字符串	指定文件名称的字符串。 • 当省去扩展名时，也省掉 “.”（句点）。 • 限制文件名为 8 个字符 + 句点 +3 个字符以内。 • 当使用了 9 个或以上字符时，不论扩展名存在与否将被省略，扩展名变为 BIN 或 CSV。	字符串	用户	
(D1)	存储读取的数据的软元件的起始号。					
	软元件	项目	含义 / 设定数据	设定范围	设定者	
	(D1)	读取结果（读数据的数目）	对于 (D0)+2. 指定的数据数，设置实际读取的数据数目。值的单位根据数据类型指定。	--	系统	
	(D1)+1 到 (D1)+□	将被读取的数据	要读取的数据。	--	系统	

<table>
<tr><th>设定数据</th><th colspan="3">含义</th><th>设定范围</th><th>设定者</th><th>数据类型</th></tr>
<tr><td rowspan="4">(D2)</td><td colspan="3">指令完成时为 ON 的位软元件。
(但在异常结束时 (D2)+1 也为 ON)</td><td></td><td></td><td rowspan="4">位</td></tr>
<tr><td>软元件</td><td>项目</td><td>含义 / 数据设置</td><td>设定范围</td><td>设定者</td></tr>
<tr><td>(D2)</td><td>结束信号</td><td>指示指令的完成。
ON: 结束
OFF: 未结束</td><td>--</td><td rowspan="2">系统</td></tr>
<tr><td>(D2)+1</td><td>异常结束信号</td><td>表明指令是正常结束或者是异常结束。
ON: 异常结束
OFF: 正常结束</td><td>--</td></tr>
</table>

注意事项

(1) 在 (S0)(驱动指定)中，只可以设置 ATA 卡的驱动 (2)。

注意当安装了 FLASH 卡时，不能通过 SP.FREAD 指令进行读取。
不能设置 SRAM 卡、标准 RAM 卡或标准 ROM 的驱动。

(2) 对于 CSV 设置，写入的数据是十进制值。

例如

字符 “A”(41H) → “65”被写入。
处理范围 : -32768 到 32767

(3) 在进行二进制读取时，字指定中文件位置的设置范围是从 00000000H 到 7FFFFFFFH。

功 能

(1) 从指定文件中读取数据

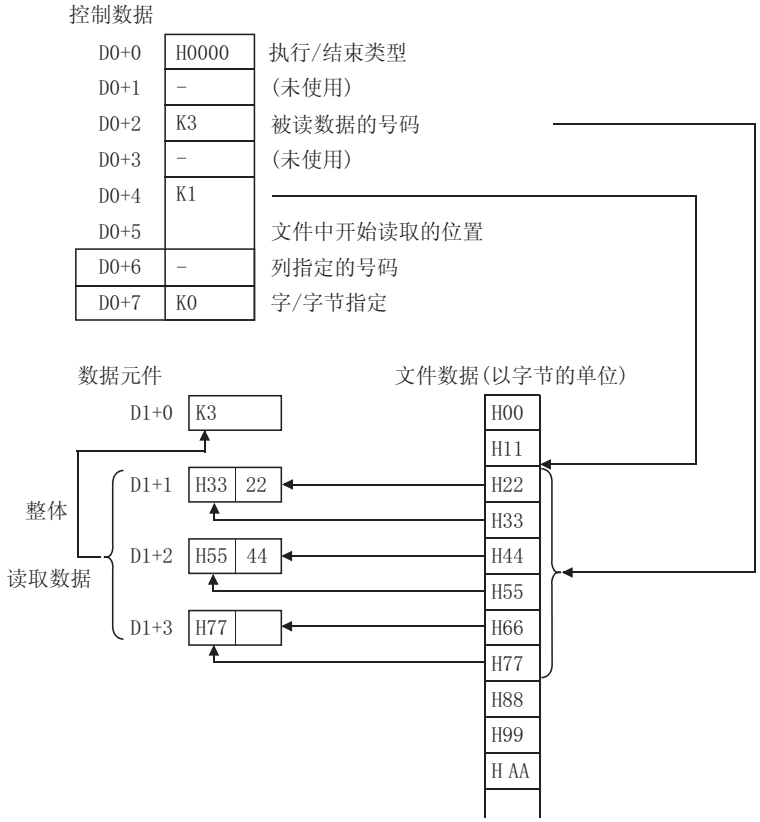
设置在控制数据中的执行 / 结束类型，从而指令是不通过任何转换读二进制数据或者在读之前转换二进制数据成 CSV 格式数据。(读取目标只有 ATA 卡。)

(2) 处理结束 ((D2)) 的位软元件在检测到指令结束的 END 处理时将自动变为 ON，在下一个扫描的 END 指令执行时变为 OFF。

使用位软元件作为 SP.FWRITE 指令执行结束的标志。
当 SP.FWRITE 指令异常结束时，异常结束 ((D2)+1) 的软元件与处理结束 ((D2)) 的软元件将同时 ON/OFF，因此可以将该软元件作为 SP.FWRITE 指令异常结束的标志使用。
此外，SM721 在指令执行过程中处于 ON 状态。
当 SM721 为 ON 时，SP.FWRITE 指令不能执行。(即使执行也将被作为无处理。)
如果在指令执行时检测到错误 (在 SM721 为 ON 之前)，处理结束软元件 ((D1)+0)、异常结束软元件 ((D1)+1) 和 SM721 不变为 ON。

(3) 数据的请求读取的数据数 (Ⓓ0+2)、文件位置 (Ⓓ0+4 和 Ⓓ0+5) 以及读取数据软元件大小 (Ⓓ1) 的处理单位是以字单位指定。

二进制读取时，分别指定时的数据读取方法如下所示：



(4) 当读取二进制数据时

(a) 如果目标文件的扩展名时省略的，使用 “BIN”作为扩展名。

(b) 如果指定文件不存在，一个错误将出现。

(c) 如果指定的位置大于存在文件的大小：

- 高性能的 QCPU 模型的系列号的前五位数字是 “01111”或更早将会出现错误。
- 对于序列号的前五位数字是 01112 或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU，将变为以 0 点执行读取并正常结束。

(5) 在 CSV 格式转换后读取数据

(a) CSV 格式文件的元素按每行读取 (EXCEL 的单元)。数量值和字符串被转化成二进制数据存储在软元件里面。

(b) 如果扩展名省略，“.CSV”作为扩展名。

(c) 当指定文件不存在时，错误将出现。

(d) 文件开始时的元素通过由被读数据号码 (Ⓓ0+2) 指定的数字而读取。

当文件最后的数据在数据的指定数字读取之前得到时：

- 系列号的前 5 位数是 “01111”或更小的高性能型 QCPU 将导致错误。
- 对于序列号的前五位数字是 01112 或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU，将读取允许读取的最大点数。

(e) 当指定的列数是 “0”时，数据在 CSV 格式文件里通过忽略行来读取。

示例 在 CSV 格式转换后读取数据并指定列数为 0 时：

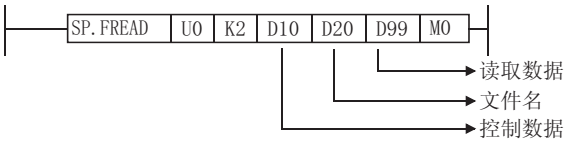
数据由EXCEL创建

	A	B	C
1	主体/分项		测量值
2	长度	1	3
3	温度	-21	

存储在CSV格式里的数据

主体/分项	,	,	测量值	CR	LF
长度	,	1	, 3	CR	LF
温度	,	-21	,	CR	LF

读到软元件的数据



控制数据

D10	H0100	执行/结束类型
D11	-	未使用读取数据的数目
D12	K9	请求
D13	-	未使用
D14	K0	
D15	K0	
D16	K0	列数指定
D17	K0	数据类型指定
D20	H4241	文件名称
D21	H4443	“ABCDE”
D22	H0045	

存储被读取数据的数

读取数据

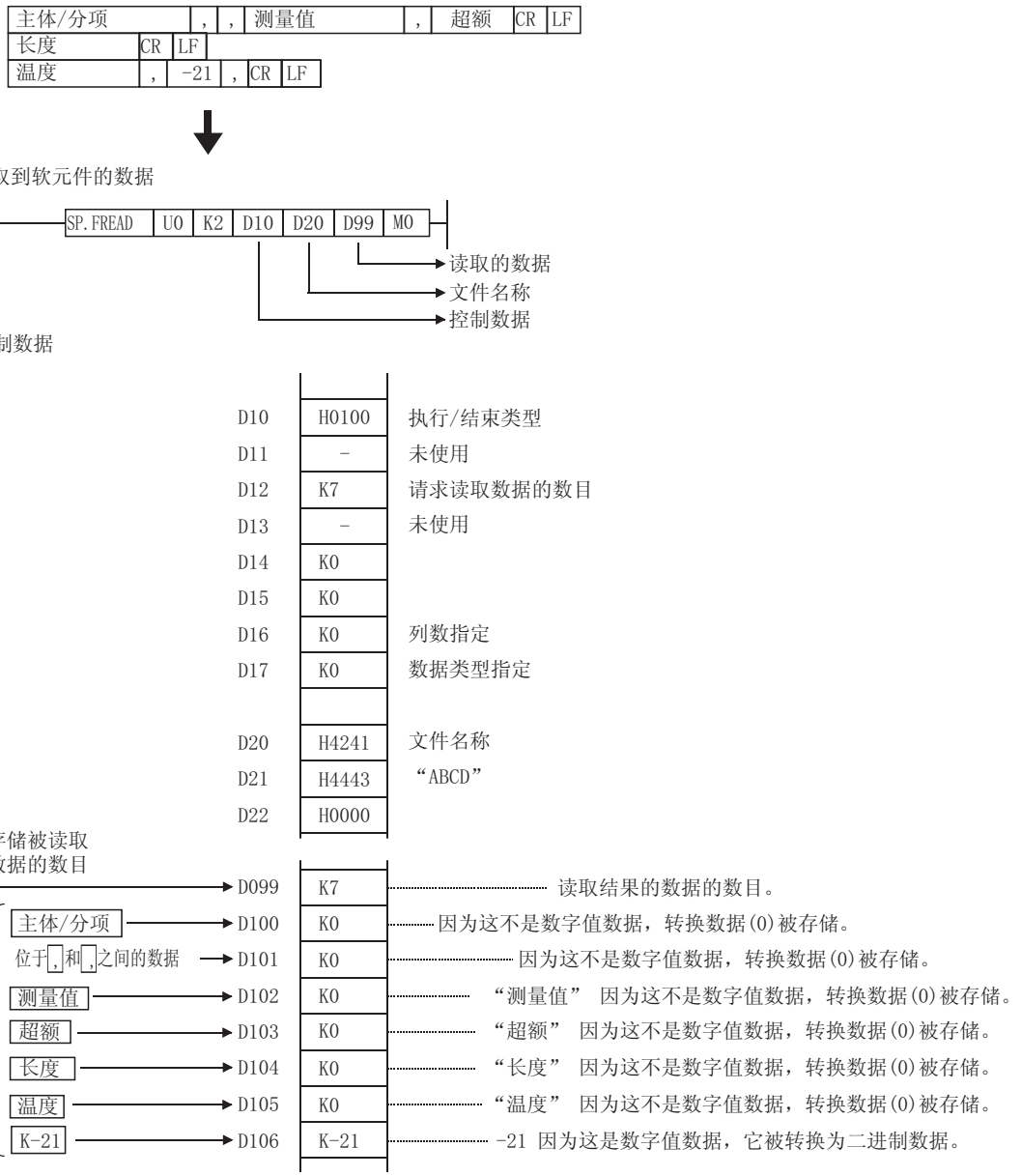
D099	K9	读取结果的数据的数目。
主体/分项 → D100	K0	“主体/分项” 因为这不是数字值数据，转换数据(0)被存储。
位于 , 和 , 之间的数据 → D101	K0	“” 因为这不是数字值数据，转换数据(0)被存储。
测量值 → D102	K0	“测量值” 因为这不是数字值数据，转换数据(0)被存储。
长度 → D103	K0	“长度” 因为这不是数字值数据，转换数据(0)被存储。
1 → D104	K1	因为这是数字值数据，它被转换为二进制数据。
3 → D105	K9	因为这是数字值数据，它被转换为二进制数据。
温度 → D106	K0	“温度” 因为这不是数字值数据，转换数据(0)被存储。
K-21 → D107	K-21	-21因为这是数字值数据，它被转换为二进制数据。
位于 , 和 CR 之间的数据 → D108	K0	“” 因为这不是数字值数据，转换数据(0)被存储。

如果列数在每行都有变化，数据同样可以忽略行来读取。

要 点

在 EXCEL 中不能创建这样的文件。当用户修改 CSV 文件后才会发生。

示例 当读取数据时，各行的列数不一样的情况：



(f) 当数据在 CSV 格式转换后读取，同时指定的列数不是 “0”，数据作为表格数据同指定的列数存储在 CSV 格式的文件里面。位于指定列外的元素需要忽略。

示例 当在 CSV 格式转换后读数据并且指定列数为 0 以外时：

由EXCEL创建的数据

	A	B	C
1	主体/分项		测量值
2	长度	1	3
3	温度	-21	

在CSV格式里存储的数据

主体/分项	,	,	测量值	CR	LF			
长度	,	1	,	3	CR	LF		
温度	,	-21	,	CR	LF			

位于指定纵列外的元素被忽略

读取到软元件的数据

SP.FREAD	U0	K2	D10	D20	D99	M0

D99 → 读取数据
D20 → 文件名称
D10 → 控制数据

控制数据

D10	H0100	执行/结束类型
D11	-	未使用
D12	K6	被读数据数目
D13	-	未使用
D14	K0	
D15	K0	
D16	K2	列指定数目
D17	K0	字/字节指定
D20	H4241	文件名
D21	H4443	“ABCD”
D22	H0000	

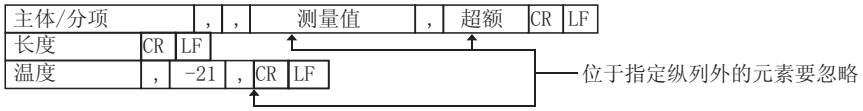
存储被读数据的数目

读取数据			
	D099	K	读取数据的数目
主体/分项	D100	K0	“主体/分项”因为这不是数字值数据，转换数据(0)被存储。
数据位于 , 和 ,	D101	K0	“”因为这不是数字值数据，转换数据(0)被存储。
长度	D102	K0	“长度”因为这不是数字值数据，转换数据(0)被存储。
1	D103	K1	1因为这是数字值数据，它被转换为二进制数据。
温度	D104	K0	“温度”因为这不是数字值数据，转换数据(0)被存储。
K-21	D105	K-21	-21因为这是数字值数据，它被转换为二进制数据。

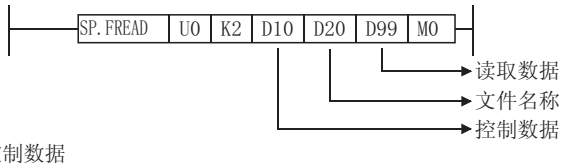
如果每行的列数都有变化，指定列外的元素将被忽略，同时 “0”被添加到元素不存在的地方。

示例

当读取数据时，各行的列数不一样的情况：



读取到软元件的数据



控制数据

D10	H0100	执行/结束类型
D11	-	未使用
D12	K6	被读数据数目
D13	-	未使用
D14	K0	
D15	K0	
D16	K2	列指定数目
D17	K0	字/字节指定
D20	H4241	文件名
D21	H4443	“ABCD”
D22	H0000	

存储被读数据的数目

读取数据

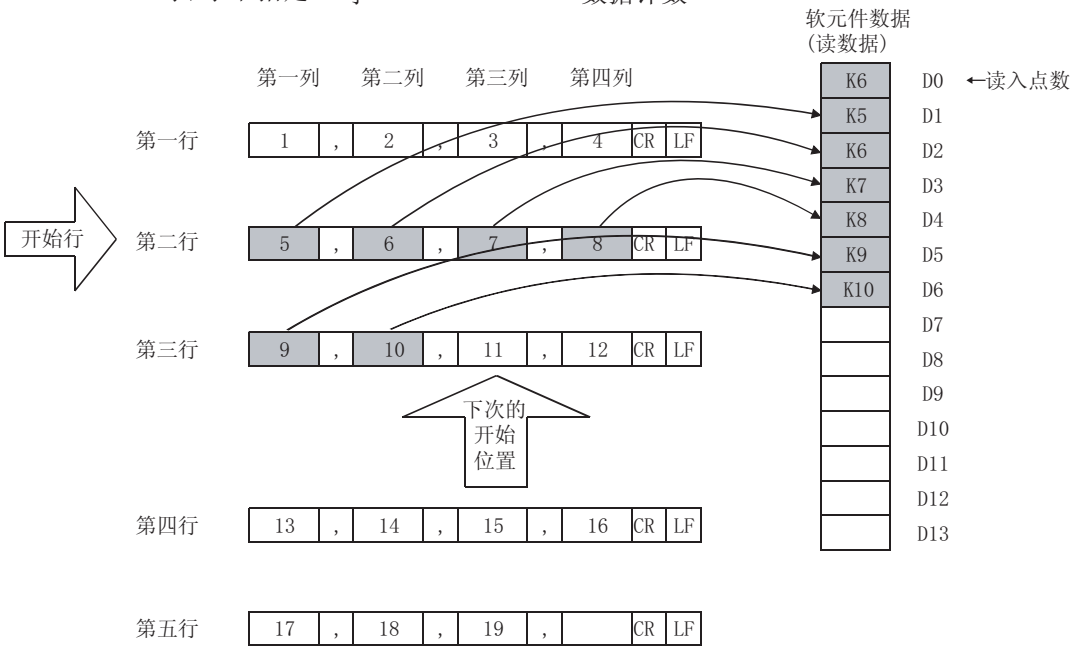
	D099	K6	读取数据的数目
主体/分项	D100	K0	“主体/分项”因为这不是数字值数据，转换数据(0)被存储。
数据位于, 和 ,	D101	K0	“”因为这不是数字值数据，转换数据(0)被存储。
长度	D102	K0	“长度”因为这不是数字值数据，转换数据(0)被存储。
无数据	D103	K0	无数据 因为没有元素存在，转换数据(0)被添加。
温度	D104	K0	“温度”因为这不是数字值数据，转换数据(0)被存储。
K-21	D105	K-21	-21 因为这是数字值数据，它被转换为二进制数据。

(g) 在序列号的前五位是 “01112”或以后的高性能型 QCPU/ 过程 CPU/ 冗余 CPU/ 通用型 QCPU 中，可以分为多次进行读取。

[指定你要开始读取的行]

执行类型 = CSV格式
列指定 = 4H
字/字节指定 = 字

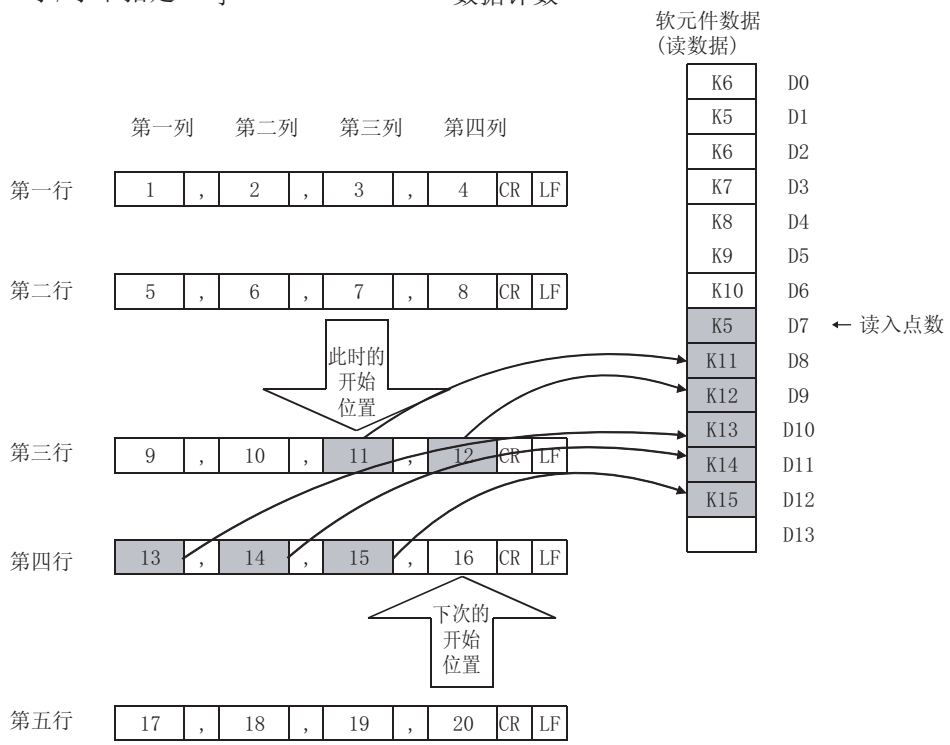
开始行数 = 2H
读入起始软元件 = D0
数据计数 = 6H



[在连续模式里，从先前读取位置的末尾处继续读取]

执行类型 = CSV模式
列指定 = 4H
字/字节指定 = 字

开始行数 = FFFFFFFFH (连续模式)
读取开始软元件 = D7
数据计数 = 5H



- 在持续模式中执行数据读取时，如果 “执行类型”，“列指定”和 “字 / 字节指定”的设定不同与先前的情况，此前的添加不能正常执行。
- 如果在另一设定中的 SP.FREAD 指令和 SP.FWRITE 指令在持续模式里数据被持续读取的时候进行执行，先前的添加不能正常工作。

(h) 当数据在 CSV 格式转换后读取时，超过范围的数字值和在目标 CSV 格式文件里的除了数字值的元素将被转换为 “0H”。

(i) 当数据在 CSV 格式转换后读取时，数字值按如下方式读取和转换：

在 CSV 格式里的数字值		-32768 到 -1	0 到 32767	32768 到 65535
字软元件	无符号	32768 到 65535	0 到 32767	32768 到 65535
	有符号	-32768 到 -1	0 到 32767	-32768 到 -1

(j) 在中断程序中不能执行此指令。
（否则，将导致误动作。）

运算错误

(1) 在下列情况下将发生运算错误，SD0 将变为 ON，出错代码将存储在 SD0 中。

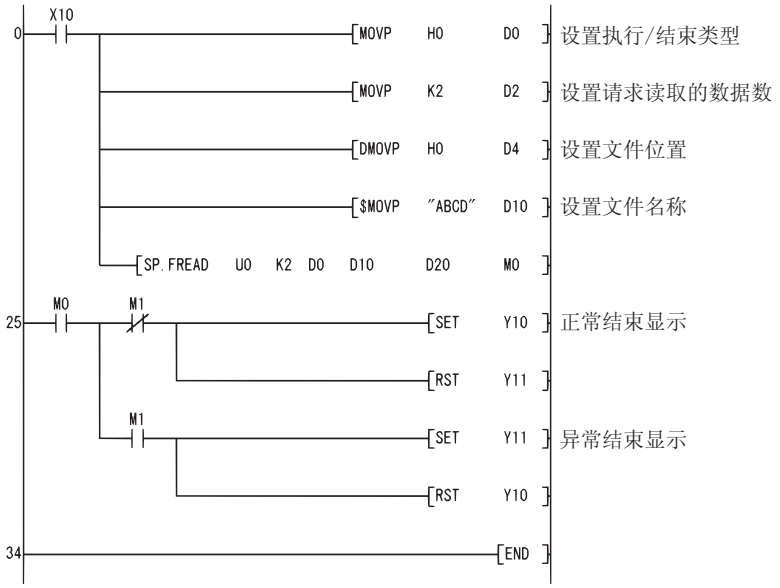
- 通过驱动指定软元件（(S0)）指定的驱动为 ATA 卡以外的中间存储器。
（出错代码：4100）
- 设定在控制数据（(D0)）后面的值超出了设置范围。（(D0)+2 除外）（出错代码：4100）
- 在 “读取数据块数”（(D0)+2）中指定的值超出了设置范围。（出错代码：4101）
- 指定了无效的软元件。（出错代码：4004）
- 在文件名称字符串（(S1)）后面指定的文件名不在指定的驱动器内。（出错代码：2410）
- 读取数据的大小超过了读取软元件的容量。（出错代码：4101）
- 当二进制数据被读取时，文件的数据数小于请求读取的数据数（(D0)+2）中指定的容量。
（序列号的前五位数字是 “01111” 或以前的高性能型 QCPU。）（出错代码：4100）
- ATA 卡出现访问异常。（出错代码：4100）
- (D0) 或 (D2) 中指定的软元件超出了相应软元件的范围。
（只对于通用型 QCPU。）（出错代码：4101）

程序示例

(1) 当 X10 为 ON 时，下列程序从嵌入到驱动器 2 的内存卡的文件 “ABCD.BIN”的开始处读取二进制数据的四个字节。

- 假设来自 D0 的 8 个数据点为控制数据软元件保留
- 假设来自 D20 的 100 个字节为读取软元件保留

[梯形图模式]



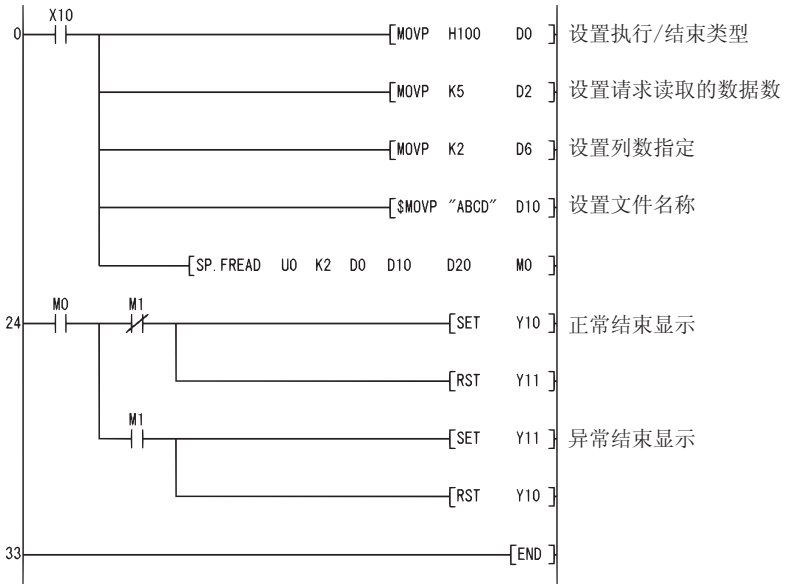
[列表模式]

步	指令	软元件					
0	LD	X10					
1	MOVP	H0	D0				
3	MOVP	K2	D2				
5	DMOVP	H0	D4				
8	$MOVP	"ABCD"	D10				
13	SP.FREAD	U0	K2	D0	D10	D20	M0
25	LD	M0					
26	MPS						
27	ANI	M1					
28	SET	Y10					
29	RST	Y11					
30	MPP						
31	AND	M1					
32	SET	Y11					
33	RST	Y10					
34	END						

(2) 当 X10 为 ON 时，下列程序从嵌入到插槽 0 中的内存卡中以 CSV 格式读取文件 “ABCD.CSV”，作为两列表格数据。

- 假设来自 D0 的 8 个数据点为控制数据软元件保留
- 假设来自 D20 的 100 字节为读取软元件保留
- 假设目标 CSV 格式文件只包括数字值

[梯形图模式]

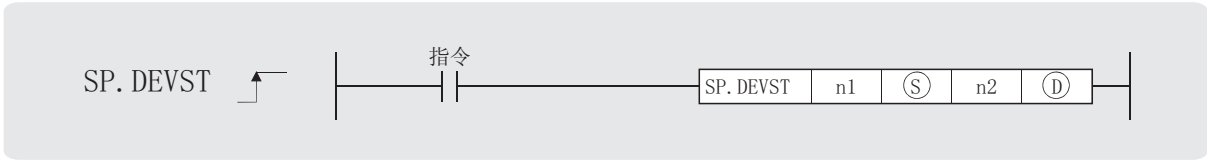


[列表模式]

步	指令	软元件					
0	LD	X10					
1	MOVP	H100	D0				
3	MOVP	K5	D2				
5	MOVP	K2	D6				
7	$MOVP	"ABCD"	D10				
12	SP.FREAD	U0	K2	D0	D10	D20	M0
24	LD	M0					
25	MPS						
26	ANI	M1					
27	SET	Y10					
28	RST	Y11					
29	MPP						
30	AND	M1					
31	SET	Y11					
32	RST	Y10					
33	END						

9.5 向标准 ROM 中写入数据 (SP.DEVST)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



n1: 软元件数据存储文件的写入偏置值（以 16 位字指定）(BIN32 位）

Ⓢ: 写入标准 ROM 的起始软元件号（软元件名称）

n2: 写入点数 (BIN16 位）

Ⓓ: Ⓓ+0 : 结束软元件（位）

Ⓓ +1: 异常结束软元件（位）

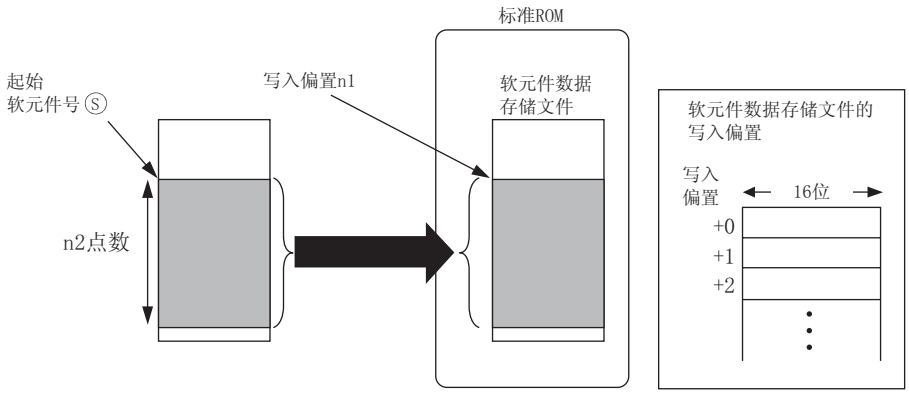
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	--	○	○	--				○	--
Ⓢ	--	○	○	--				--	--
n2	--	○	○	--				○	--
Ⓓ	∧*1	--	∧*1	--				--	--

*1: 局部软元件以及各程序的软元件不可使用。

☆功 能

(1) 将 Ⓢ 指定的软元件的 n2 指定点数的软元件数据写入到标准 ROM 中的软元件数据存储文件的 n1 中指定的写入偏置。

n1 是从软元件数据存储文件起始开始的偏置并由字偏置（以 16 位字为单位）指定。



(2) 标准 ROM 中的软元件数据写入位置结束软元件 (Ⓓ+0) 在检测到本指令的处理结束的 END 指令执行时将自动变为 ON，并在下一次扫描的 END 指令变为 OFF，因此被作为本指令的执行结束标志使用。

(3) 当本指令异常结束时，异常结束软元件 (Ⓓ+1) 与结束软元件 (Ⓓ+0) 同时变为 ON/OFF，因此被作为本指令的异常结束标志使用。

(4) 此指令执行时，SM721 变为 ON。
当 SM721 已经变为 ON 时，则不能执行此指令。(如果执行，则不会执行处理。)

(5) 在执行此指令的过程中检测到错误时，结束软元件 (Ⓓ+0)、异常结束软元件 (Ⓓ+1) 和 SM721 不会变为 ON。

运算错误

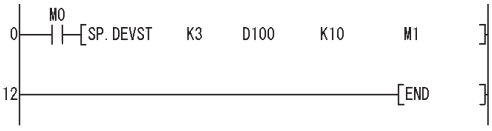
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- n1 指定的写入偏置超出了软元件数据存储文件的范围。 (出错代码： 4100)
- 从 n1 指定的写入偏置起的 n2 点数超出了软元件数据存储文件的范围。 (出错代码： 4100)
- Ⓢ 指定的软元件起的 n2 点数范围超出了相应软元件。 (出错代码： 4141)
- 未在 GX Developer 的可编程控制器参数的 “可编程控制器文件”中设置软元件数据存储文件。 (出错代码： 2410)
- Ⓓ 指定的软元件超出了相应软元件的范围。 (出错代码： 4101)

程序示例

(1) 当 M0 为 ON 时，下列程序将 D100 的 10 点数据写入到标准 ROM 中的软元件数据存储文件中。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	M0
1	SP.DEVST	K3 D100 K10 M1
12	END	

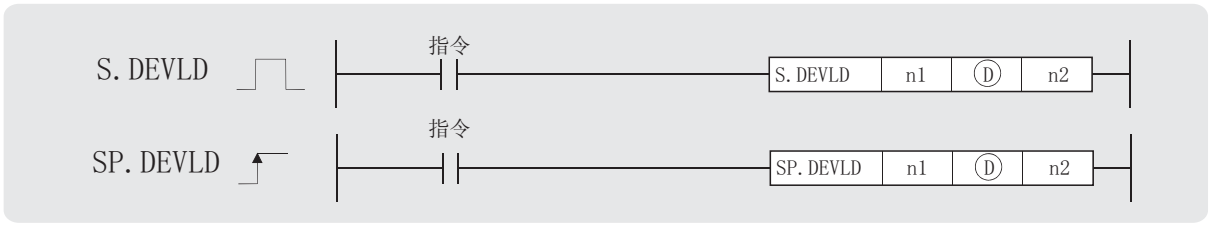
注意事项

(1) 写入标准 ROM 的值是执行本指令时的值。

(2) 由于 SP.DEVST 指令的执行，标准 ROM 的写入计数指标 (SD687 和 SD688) 将会增加。如果标准 ROM 的写入计数指标超过了 10 万次，将会出现 FLASH ROM ERROR(出错代码： 1610)。

9.6 从标准 ROM 中读取数据 (S(P).DEVLD)

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR



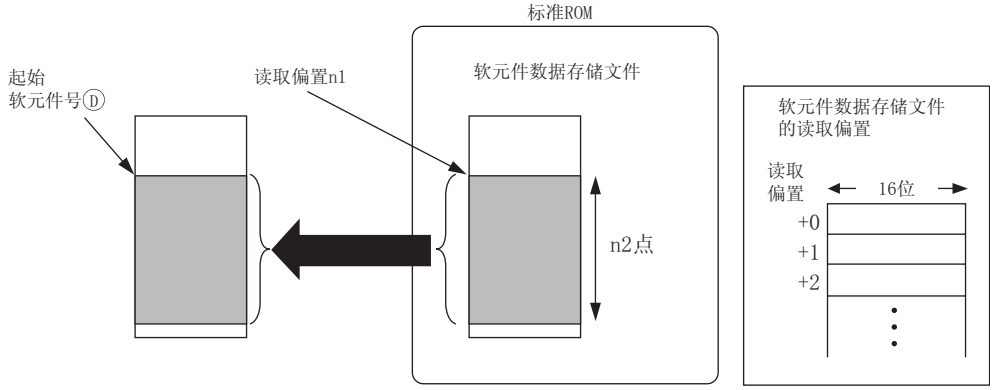
n1 ：软元件数据存储文件的读取偏置值（以 16 位字指定）(BIN32 位）
Ⓓ ：从标准 ROM 读取的起始软元件号（软元件名称）
n2 ：读取点数 (BIN16 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			E	
n1	—	○		—				○	—
Ⓓ	—	○		—				—	—
n2	—	○		—				○	—

功 能

(1) 从标准 ROM 中的软元件数据存储文件的 n1 指定的读取偏置中，读取 n2 中指定点数的软元件数据，将其存储到 Ⓓ 中指定的软元件中。

n1 是从软元件数据存储文件起始开始的偏置并由字偏置（以 16 位字为单位）指定。



运算错误

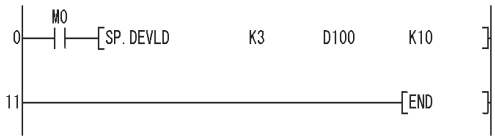
(1) 在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- n1 指定的地址超出了标准 ROM 的范围。 (出错代码： 4100)
- 从 n1 指定的地址起的 n2 点数超出了标准 ROM 的范围。 (出错代码： 4100)
- 从 ⓓ 指定的软元件起的 n2 点数范围超出了相应软元件。 (出错代码： 4101)
- 未在 GX Developer 的可编程控制器参数的 “可编程控制器文件”中设置软元件数据存储文件。 (出错代码： 2410)

程序示例

(1) 当 M0 为 ON 时，下列程序从标准 ROM 中的软元件数据存储文件中读取 10 点数据到 D100 中。

[梯形图模式]

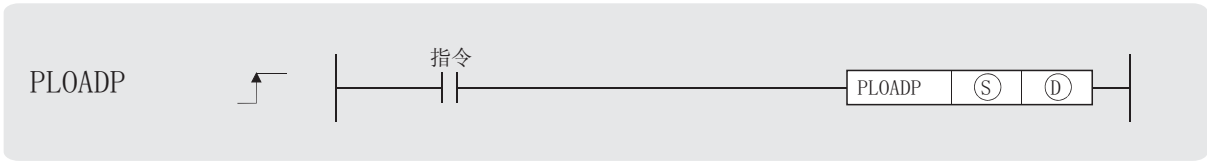


[列表模式]

步	指令	软元件		
0	LD	M0		
1	SP.DEVLD	K3	D100	K10
11	END			

9.7 从内存卡中装载程序 (PLOADP)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 存储要装载程序的驱动号码，文件名的字符串数据，或者存储字符串数据的软元件的起始号 (BIN 16 位) *1

Ⓓ : 指令结束时，软元件变为 ON 一个扫描周期 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	△*2	--		--				--	--

*1: 指定为 “< 驱动号 >:< 文件名称 >”。例如) 1:MAIN

*2: 局部软元件不可使用。

功 能

(1) 存储在内存卡或标准 ROM 的程序将转移到程序内存。(驱动 0)

如果转移的文件没有注册到可编程控制器参数对话框的程序设定里，在 CPU 模块里的设定的程序将设为待命类型。

在这时，可编程控制器参数对话框的设定不再变化。

(用 PLOADP 指令转移一个程序，再程序内存中需要一个持续的空余空间。)

(2) 使用 PLOADP 指令添加的程序在未使用的程序号码中分配为最小号。

(如果要手动分配程序号，应将分配的程序号存储到 SD720 中)

下面的示例为通过 PLOADP 指令添加 “MAIN6”。

(a) 当程序号码连续的被设定，新的程序在现在程序号码的末尾进行添加。

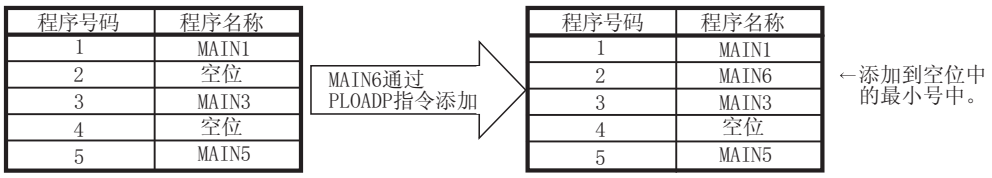
当 1 和 5 号程序已被设定，新的程序添加作为程序 6。

程序号码	程序名称
1	MAIN1
2	MAIN2
3	MAIN3
4	MAIN4
5	MAIN5

MAIN6通过 PLOADP指令添加

程序号码	程序名称	
1	MAIN1	
2	MAIN2	
3	MAIN3	
4	MAIN4	
5	MAIN5	
6	MAIN6	←在末尾添加

(b) 当有多个未装入程序的空闲的程序号时，PLOADP 指令指定的程序将被添加到最小号码中。
（使用 PUNLOADP 指令删除程序时将会产生空闲的程序号。）
当程序号 2 及 4 为空位时，程序将被添加到程序号 2 中。



(3) 驱动 1、2 和 4 可以指定（驱动 3 不可以指定。）

- 驱动 1：内存卡 (RAM)
- 驱动 2：内存卡 (ROM)
- 驱动 4：标准 ROM

(4) 对于文件名称扩展名 (.QPG) 不需要指定。

(5) 在指令结束时的 END 处理过程中，由 ⓓ 指定的位软元件变为 ON。在下一个 END 程序处理的时候，位软元件变为 OFF。

(6) PLOADP、PUNLOADP 和 PSWAPP 指令不能同时执行。
如果两个或更多的上述指令同时执行，后面的指令将不能执行。
当使用上述指令时，请用户采取互锁措施避免同时执行。

(7) 在中断程序时不能执行这个指令。
（在中断程序里执行这个指令可以导致功能故障）

(8) 在执行通过 PLOADP 指令传送到程序内存中的文件时，应通过 PSCAN 指令设置为 “扫描执行类型”。（参见 7.17.3 项）

(9) 装载文件的可编程控制器文件设定如下：

(a) 各程序的文件的使用方法
通过本指令传送的程序的文件寄存器、软元件初始值、注释和局部软元件的使用方法全部为 “按照可编程控制器文件设置”。
然而，通过本指令传送程序时，如果满足了以下两个条件，将会返回错误。

- 在可编程控制器文件设置中设置为 “使用局部软元件”。
- 程序内存的程序数量超过了参数设置中设定的程序数量。

在通过本指令传送的程序中使用局部软元件时，应在参数中注册一个虚拟程序文件，然后使用 PUNLOADP 指令删除虚拟程序文件后，使用 PLOADP 指令装载程序。

(b) I/O 刷新设置
通过本指令传送的程序的 I/O 刷新设置对于输入和输出均为 “无设置”。

(10)“PLOADP”指令和 “在运行时写入”处理不能同时执行。

(a) 在 PLOADP 指令的处理过程中，需要一个运行时候的写入，运行的写入会延迟。
在 PLOADP 指令处理结束后，运行中的处理再开始。

(b) 当 PLOADP 指令在运行中写入的时候执行，PLOADP 指令的处理将延迟。
在运行中处理结束后，PLOADP 指令的处理再开始。

运算错误

(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

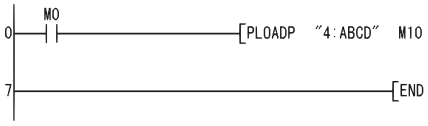
- 文件名称在由 Ⓢ 指令的驱动号码中不存在。 (出错代码： 2410)
- 由 Ⓢ 指定的驱动号无效。 (出错代码： 4100)
- 在 0 驱动中，没有足够的内存来装载指定的文件。 (出错代码： 2413)
- 下面显现的程序数字已经在程序内存中登记。 (出错代码： 4101)
- 在 SD720 中存储的程序号码已经使用，或大于下面显示的最大程序数字。 (出错代码： 4101)
- 程序文件同已经存在的将装载的程序文件相同。 (出错代码： 2410)
- 本地驱动的文件大小不能保留。 (出错代码： 2401)

CPU 类型名称	程序内存（文件号码）	最大程序号码
Q02 (H) CPU	28	28
Q06HCPU	60	60
Q12HCPU	124	124
Q25HCPU	124	124
Q12PHCPU	124	124
Q25PHCPU	124	124

程序示例

(1) 当 M0 为 ON 时，下列程序装载存储在驱动 4 中的文件 “ABCD.QPG”到驱动 0，同时程序处于备用状态。

[梯形图模式]

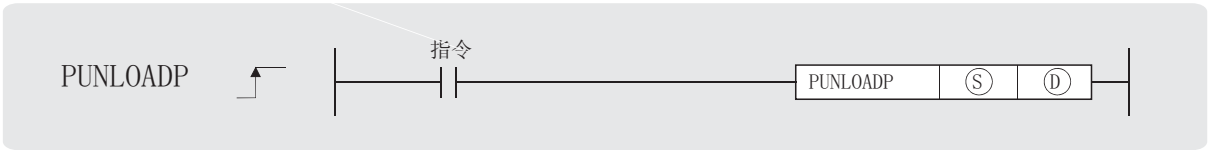


[列表模式]

步	指令	软元件
0	LD	M0
1	PLOADP	"4:ABCD" M10
7	END	

9.8 从程序内存中卸载程序 (PUNLOADP)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 将被卸载的程序文件名称的字符串数据或存储字符串数据的软元件的起始号 (BIN16 位)

Ⓓ : 指令结束时 ON 一个扫描的软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	△*1	--		--				--	--

*1: 局部软元件不能使用。

功 能

(1) 存储在程序内存 (驱动 0) 的备用程序将从程序内存中删除。
(用 PSCAN 指令设置为 “扫描执行类型”的程序或用 PLOW 指令设置为 “低速执行类型”的程序不能被删除。)

(2) 被 PUNLOADP 指令删除的程序号码为空位。
当程序号码为 1 到 5 在可编程控制器参数对话框的设定中已经设定，用 PUNLOADP 指令删除 2 号程序使得程序号码 2 变为空位。

程序号码	程序名称
1	MAIN1
2	MAIN2
3	MAIN3
4	MAIN4
5	MAIN5

用PLOADP指令
删除“MAINS”

程序号码	程序名称
1	MAIN1
2	空位
3	MAIN3
4	MAIN4
5	MAIN5

←2号程序已删除

(3) 对于文件名称不必指定扩展名 (.QPG)

(4) 由 Ⓓ 指令的位软元件在 PUNLOAD 指令结束的 END 程序处理中继续。在下一个 END 处理时，位软元件为 OFF。

(5) PLOADP、PUNLOADP 和 PSWAPP 指令不能同时执行。
如果两个或更多的上述指令同时执行，随后需要执行的指令不能被执行。
当使用上述指令时，以用户来提供互锁避免同时执行。

(6) 当可编程控制器在为 OFF 后为 ON 或者 CPU 模块在 PUNLOADP 指令执行后复位设置，下列操作将被执行。

(a) 当在可编程控制器参数对话框中已经进行了启动设置，在启动设置位置的程序将转移到程序内存。
当用 PUNLOADP 指令删除的程序不能执行，从启动设置和可编程控制器参数对话框的程序设定中删除相应的文件名称。

(b) 当在可编程控制器参数对话框的启动文件没有设定，“FILE SET ERROR”（出错代码:2400）将出现。
1) 当用 PUNLOADP 指令删除的文件不能执行，从可编程控制器参数对话框的程序设定中删除相应程序名称。
2) 当用 PUNLOADP 指令删除的程序再次执行时，把相应程序写入 CPU 模块。

(7) 在中断程序里不要执行 PUNLOADP 指令。
（否则，将导致功能故障）

(8) 从程序内存中用这个指令将删除的程序将先用 PSTOP 指令设置为 “待命执行类型”。
（参见 7.17.1 项）

(9) “PUNLOADP”指令和 “在运行时写入”处理不能同时执行。

(a) 在 PUNLOADP 指令的处理过程中，需要一个运行时候的写入，运行的写入会延迟。
在 PUNLOADP 指令处理结束后，运行中的处理再开始。

(b) 当 PUNLOADP 指令在运行中写入的时候执行，PUNLOADP 指令的处理将延迟。
在运行中处理结束后 PUNLOADP 指令的处理再开始。

运算错误

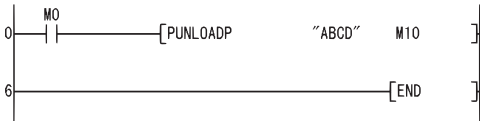
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 Ⓢ 指定的文件名称不存在。 （出错代码： 2410）
- 由 Ⓢ 指定的程序不是备用程序。或者是当前正在执行的程序。 （出错代码： 4101）

程序示例

(1) 当 M0 从 OFF 变为 ON 时，下列文件从内存中删除存储在驱动 0 的 “ABCD.QPG”文件。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	M0
1	PUNLOADP	"ABCD" M10
6	END	

9.9 装载 + 卸载 (PSWAPP)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



(S1) : 将被卸载的程序文件名称的字符串数据或存储字符串数据的软元件的起始号 (BIN16 位)

(S2) : 存储装载程序的驱动号、文件名称的字符串数据或存储字符串数据的软元件的起始号 (BIN16 位) *1

(D) : 指令结束时 ON 一个扫描的软元件 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 $	其它
	位	字		位	字 --				
(S1)	--	○		--				○	--
(S2)	--	○		--				○	--
(D)	△*2	--		--				--	--

*1: 指定为 “< 驱动号 >:< 文件名称 >”。例如) 1:MAIN

*2: 局部软元件不可使用。

功 能

(1) 将 (S1) 指定的程序内存 (驱动 0) 中存储的备用类型程序从程序内存中删除，然后将 (S2) 指定的内存卡或标准 ROM 中存储由的程序传送到程序内存中，使其为备用状态。
(将程序传送到程序内存中时，程序内存中必须有连续的空余空间。)
由 PSCAN 指令设定为 “扫描执行类型” 的程序或由 PLOW 指令设定为 “低速执行类型” 的程序不能删除。

(2) 用 PSWAPP 指令转移到内存卡的程序和从程序内存中删除的程序的程序号码相同。
(如果在程序从程序内存删除之前有一个为 ON 的程序号码，将要转移到程序内存的程序将没有为 ON 的程序号码。)
当程序号码 2 为 ON 时，转移到程序内存的程序通过 PSWAPP 指令用程序交换方式注册为 3 号程序。

程序号码	程序名称
1	MAIN1
2	空位
3	MAIN3
4	MAIN4
5	MAIN5

通过PAWAPP指令
MAIN3和MAIN6交换

程序号码	程序名称	
1	MAIN1	
2	空位	
3	MAIN6	←MAIN6进入
4	MAIN4	
5	MAIN5	

(3) 驱动器 1、2、4 可以被指定（驱动 3 不可以接受指令）

- 驱动 1: 内存卡 (RAM)
- 驱动 2: 内存卡 (ROM)
- 驱动 4: 标准 ROM

(4) 对于一个文件名称不必指定一个扩展名 (.QPG)。

(5) 由 ⓓ 指定的位软元件在指令结束的 END 扫描处理过程中为 ON，在下一个 END 处理的阶段为 OFF。

(6) PLOADP、PUNLOADP 和 PSWAPP 指令不能同时执行。
如果两个或更多的上述指令同时执行，随后需要执行的指令不能工作。
当使用上述指令时，由用户来提供互锁避免同时执行。

(7) 在 PSWAPP 指令执行后，进行了可编程控制器的 OFF → ON 或者 CPU 模块的复位时，将执行下列动作。

(a) 当在可编程控制器参数对话框中已经进行了启动设置，启动设置位置的程序将转移到程序内存。
当用 PSWAPP 指令替换的程序执行时，改变启动设置和可编程控制器参数对话框中的程序设定为相应的程序名称。

(b) 当在可编程控制器参数对话框中没有进行启动设置，“FILE SET ERROR”（出错代码 :2400）将出现。
1) 当被 PSWAPP 指令替换的程序被执行时，改变可编程控制器参数对话框的程序设定为相应的程序名称。
2) 在可编程控制器参数对话框的程序设置中执行程序设定时，把相应程序重新写入 CPU 模块。

(8) 在中断程序中不能执行这个指令。
（在中断程序中执行这个指令将导致故障）

(9) 交换程序的可编程控制器文件设定如下：

(a) 每个程序的文件使用
所有文件寄存器、软元件初始值、注释和交换程序的本地软元件设定为 “跟随可编程控制器文件设定”。

(b) I/O 刷新设置
对于交换程序的 I/O 刷新设置对于输入和输出都无法使用。

(10)“PSWAPP”指令和 “在 RUN 过程中写入”处理不能同时执行。

(a) 当在 PSWAPP 指令的处理过程中，有一个 RUN 过程中的写入请求时，RUN 过程中的写入会延迟。
RUN 过程中的写入在 PSWAPP 指令处理结束后开始。

(b) 如果在 RUN 中写入的过程中执行 PSWAPP 指令，PSWAPP 指令的处理将等待。
在 RUN 中写入的处理结束后，进行 PSWAPP 指令的处理。

运算错误

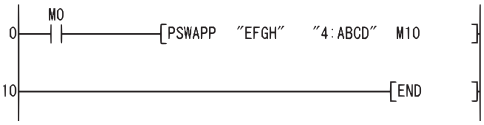
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 由 ⓢ1 和 ⓢ2 指定的驱动号码或文件名称不存在。 (出错代码：2410)
- 由 ⓢ1 指定的驱动号码无效。 (出错代码：4100)
- 在驱动 0 中没有足够的容量装载指定的程序。 (出错代码：2413)
- 由 ⓢ1 指定的程序不处于备用状态或正在执行。 (出错代码：4101)

程序示例

(1) 当 M0 由为 OFF 到为 ON，下列程序将从内存中删除存储在驱动 0 中的文件 “EFGH.QPG”，装载存储在驱动 4 中的文件 “ABCD.QPG” 到驱动 0，同时置于备用状态。

[梯形图模式]

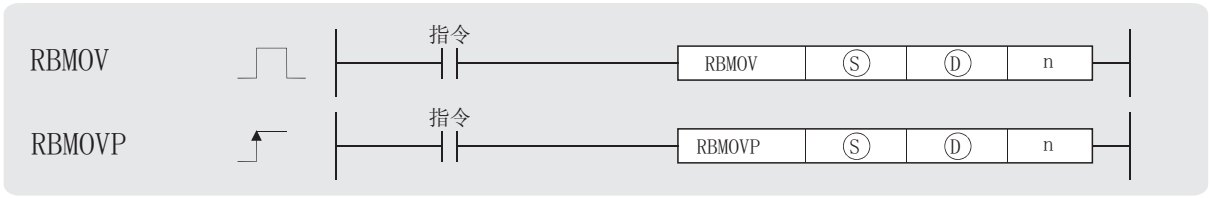


[列表模式]

步	指令	软元件
0	LD	M0
1	PSWAPP	"EFGH" "4:ABCD" M10
10	END	

9.10 文件寄存器的高速块传送 (RBMOV(P))

通用型 UD　基本型　高性能型　过程　冗余　QnA　Q4AR

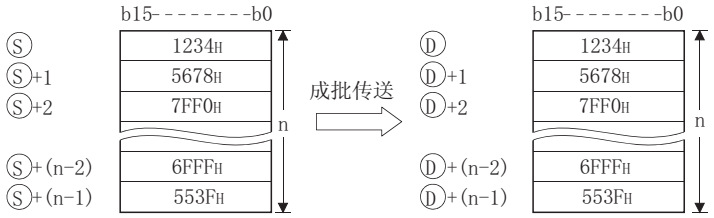


Ⓢ : 存储传送数据的软元件的起始号 (BIN16 位)
Ⓓ : 传送目标软元件的起始号 (BIN16 位)
n : 传送数据的数目 (BIN16 位)

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它字
Ⓢ		○					--		--
Ⓓ		○					--		--
n		○					○		--

功 能

(1) 由 Ⓢ 指定的软元件开始的 “n” 点 16 位数据成批传送到由 Ⓓ 指定的软元件开始的 “n” 点区域。



(2) 即使传送源和传送目标的软元件之间有重叠，也可以进行传送。
传送到较小的软元件号时，数据从 Ⓢ 开始传送。传送到较大软元件号时，数据从 Ⓢ+(n-1) 开始传送。

然而，当从 R 传送数据到 ZR 或从 ZR 传送数据到 R 时，应按以下方式避免 ZR 与 Z 的各传送范围重叠。

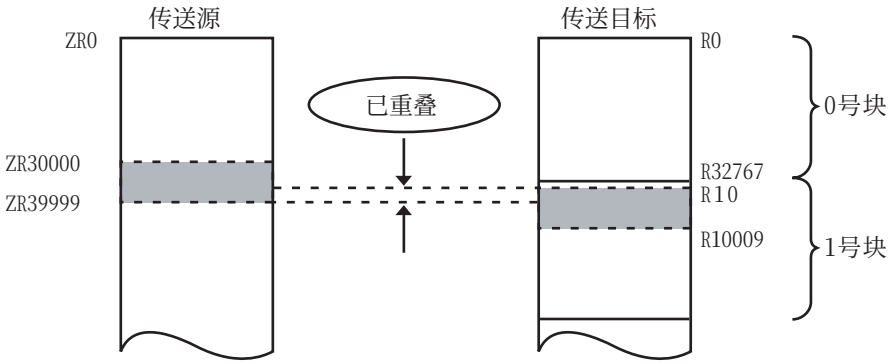
- ZR 的传送范围 ((指定的 ZR 起始号) 到 (指定的 ZR 起始号 + 传送数量 -1))
- R 的传送范围 ((指定的 R 起始号 + 文件寄存器块号× 32768) 到 (指定的 R 起始号 + 文件寄存器块号× 32768+ 传送数量 -1))

示例

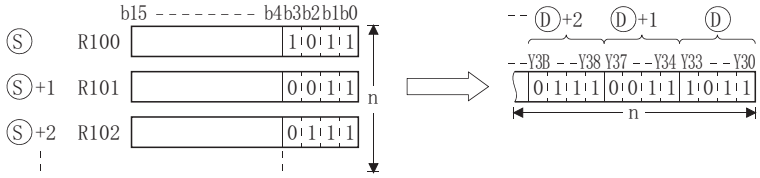
当从 ZR30000（源）传送 10000 点数据到 R10（传送目标的 1 号块）时，ZR 和 R 的传送范围互相重叠。

- ZR 的传送范围 → (30000) 到 (30000+10000-1) →(30000) 到 (39999)
- R 的传送范围 → (10+(1×32768)) 到 (10+(1×32768)+10000-1)
  → (32778) 到 (42777)

因此，范围 32778 到 39999 互相重叠。



(3) 如果 Ⓢ 是一个字软元件而 Ⓓ 是一个位软元件，字软元件的对象是由位软元件数字定义指定的位数。
当 K1Y30 由 Ⓓ 指定时，由 Ⓢ 指定的字软元件较低四位作为对象。



运算错误

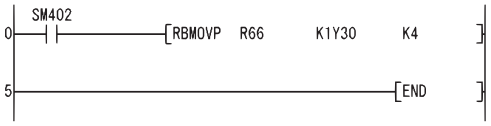
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 始于 Ⓢ 或 Ⓓ 的 “n” 点软元件范围超过了现有软元件。 （出错代码：4101）
- 文件寄存器不是由 Ⓢ 或 Ⓓ 指定的。 （出错代码：4101）

程序示例

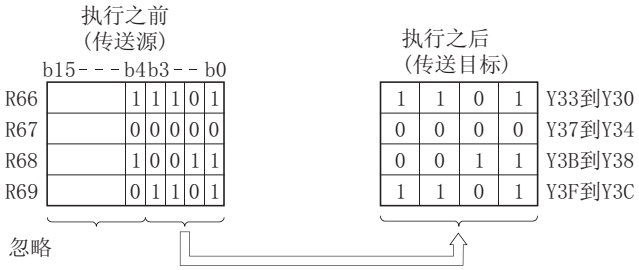
(1) 下列程序以 4 点为单位将 R66 ～ R69 的低 4 位的数据输出到 Y30 ～ Y3F 中。

[梯形图模式]



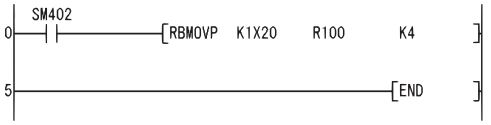
[列表模式]

步	指令	软元件		
0	LD	SM402		
1	RBMOVP	R66	K1Y30	K4
5	END			



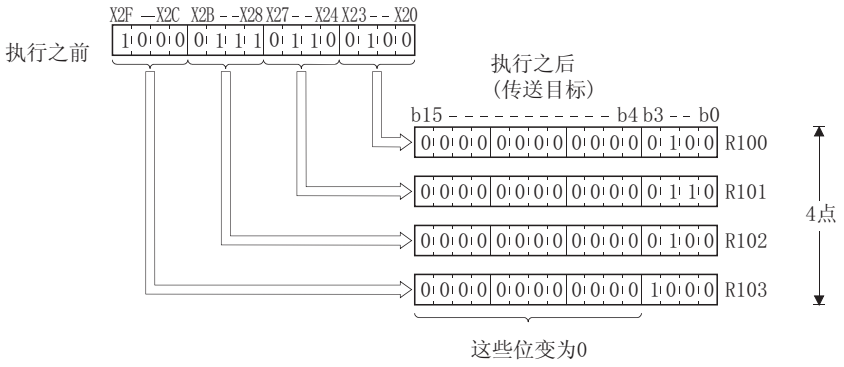
(2) 下列程序以 4 点为单位将 X20 ～ X2F 中的数据输出到 R100 ～ R103 中。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	SM402		
1	RBMOVP	K1X20	R100	K4
5	END			



要 点

RBMOV(P) 指令对于在 QnHCPU/QnPHCPU/QnPRHCPU 中批量传送大量文件寄存器数据时很有用。
对于 QnUCPU，RBMOV(P) 的处理速度与 RBMOV 指令相同。
RBMOV 和 BMOV 指令的处理速度对比表如下：

(1)从文件寄存器传送到内部软元件 / 内部软元件传送到文件寄存器时。

CPU	存储文件寄存器的目标内存	RBMOV			BMOV	
		1 字	1000 字	10000 字	1 字	10000 字
QnHCPU QnPHCPU QnPRHCPU	标准 RAM	20.0μs	91.0μs	775.0μs	7.5μs	720.0μs
	SRAM 卡	22.0μss	305.0μs	2900.0μs	8.0μs	3900.0μs
	闪存卡 *1	22.0μss	405.0μs	3950.0μs		4250.0μs
QnCPU	标准 RAM	45.0μs	215.0μs	1850.0μs	17.5μs	1700.0μs
	SRAM 卡	49.0μs	540.0μs	5150.0μs	18.0μs	5050.0μs
	闪存卡 *1					5800.0μs

*1 ： 当文件寄存器存储在闪存卡时，不会执行从内部软元件到文件寄存器的传送操作。

(2)从文件寄存器传送到文件寄存器时

CPU	存储文件寄存器的目标内存	RBMOV			BMOV	
		1 字	1000 字	10000 字	1 字	10000 字
QnHCPU QnPHCPU QnPRHCPU	标准 RAM	20.0μs	91.0μs	775.0μs	7.5μs	720.0μs
	SRAM 卡	22.5μs	545.0μs	5300.0μs	8.5μs	7050.0μs
QnCPU	标准 RAM	45.5μs	215.0μs	1850.0μs	17.5μs	1700.0μs
	SRAM	50.0μs	870.0μs	8350.0μs	18.5μs	8600.0μs

9.11 写入自站 CPU 共享内存

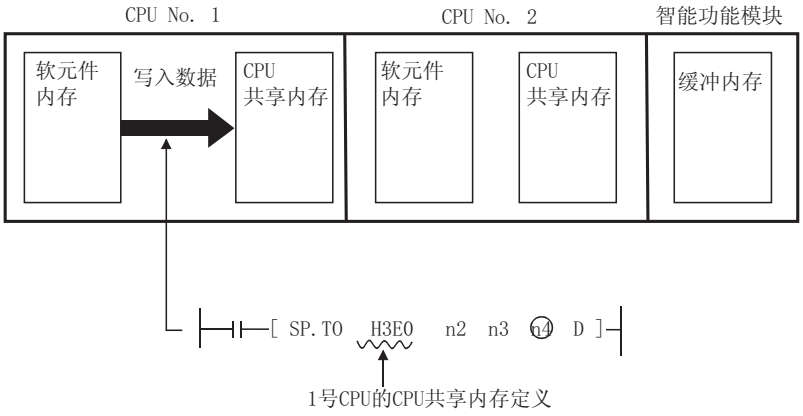
S.TO 或 TO 指令用来在多 CPU 系统中执行写入到自站 CPU 共享内存。S.TO 或 TO 都可以用来执行写入 CPU 共享内存的操作。

下列表格表明 S.TO 和 TO 指令的可用性。

CPU 模块型号名称		S.TO 指令	TO 指令
基本型 QCPU	Q00CPU Q01CPU	可使用	可使用
高性能型 QCPU	Q02 (H) CPU Q06HCPU Q12HCPU Q25HCPU	可使用	不能使用
过程 CPU	Q12PHCPU Q25PHCPU	可使用	不能使用
冗余 CPU	Q12PRHCPU Q25PRHCPU	不能使用	不能使用
通用型 QCPU	Q02UCPU Q03UDCPU Q04UDHCPU Q06UDHCPU	可使用	可使用

(1) S.TO 指令的操作

S.TO 指令能够写入数据到自站 CPU 模块的 CPU 共享内存。
下列数据表明当 S.TO 指令在 1 号 CPU 执行时程序的处理过程。

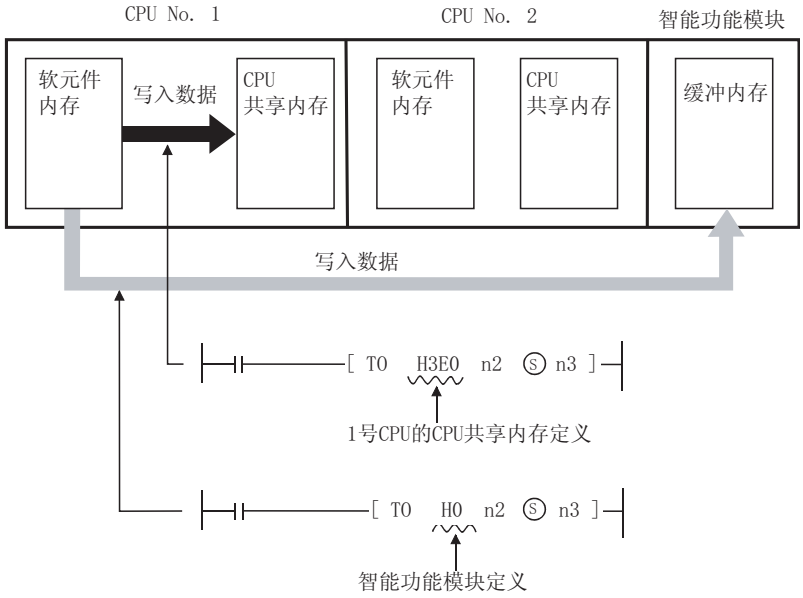


(2) TO 指令的操作

TO 指令能够写入软元件内存数据到下列内存。

- 自站 CPU 模块的 CPU 共享内存。
- 智能功能模块的缓冲内存。

下列数据表明当 TO 指令在 1 号 CPU 执行时的处理过程。



要 点

S.TO 和 TO 指令都可以被基本型 QCPU(Q00CPU 或 Q01CPU) 和通用型 QCPU 用于向 CPU 共享内存写入数据。然而，对自站的 CPU 共享内存进行写入时，建议使用 TO 指令，因为使用 TO 指令可以减少步数和处理时间。

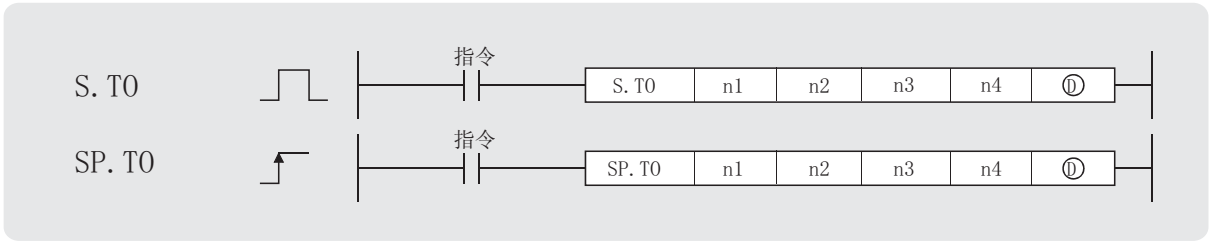
备 注

关于用 TO 指令写入到智能功能模块的缓冲内存，参考 7.8.2 项。

9.11.1 写入到自站 CPU 共享内存 (S(P).TO)

通用型 UD　基本型 *1　高性能型 *2　过程　冗余　QnA　Q4AR

*1：系列号前五位数为“04122”或以后。
*2：功能版本 B 或以后的版本。

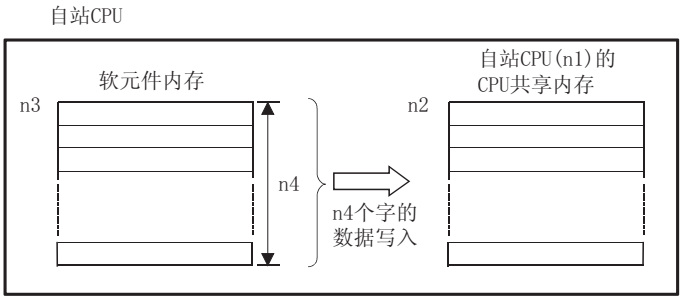


n1 ：自站 CPU 起始 I/O 号 (BIN16 位)
n2 ：写入目标的自站的 CPU 共享内存地址 (BIN16 位)
　　基本型 QCPU:0 到 511
　　高性能型 QCPU、过程 CPU、通用型 QCPU:0 到 4095
n3 ：存储写入数据的软元件的起始号 (BIN16 位)
n4 ：写入数据块的数目 (BIN16 位)
　　基本型 QCPU:1 到 320
　　高性能型 QCPU、过程 CPU:1 到 256
　　通用型 QCPU:1 到 2048
D ：在写入完成时 ON 一个扫描的自站 CPU 软元件 (位)

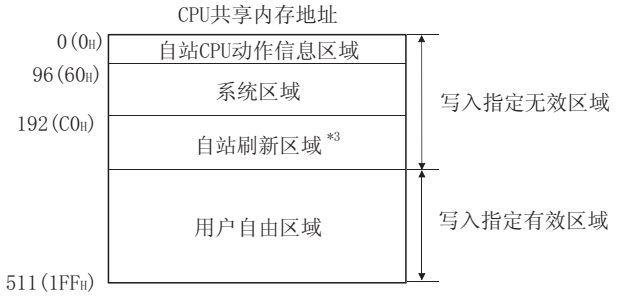
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
n1	—	○		—				○	—
n2	—	○		—				○	—
n3	—	○		—				—	—
n4	—	○		—				○	—
D	○	○		○				—	—

功 能

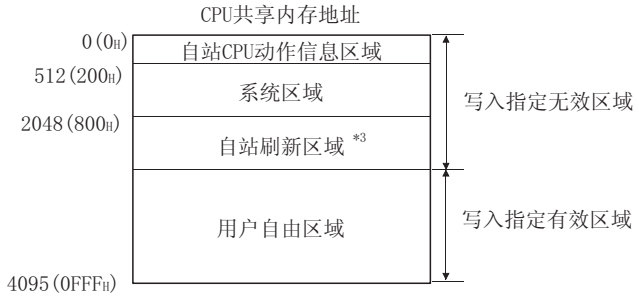
(1) 在软元件里写入 n4 个字的数据，从自站 CPU 在 n3 中的号码开始到自站 CPU 模块在 n2 指定的共享内存地址。
写入结束时，D 指定的结束位将变为 ON。



(a) 基本型 QCPU 的 CPU 共享内存地址



(b) 高性能型 QCPU、过程 CPU 和通用型 QCPU*4 的 CPU 共享内存地址。



*3: 当自动刷新设定没有使用时，作为用户空余区域使用。
当自动刷新设定完成时，自动开始传送范围和其后区域作为用户空余空间使用。

*4 : 不能使用 S(P).TO 指令将数据写入通用型 QCPU 的多 CPU 高速通信区域。

(2) 当写入端数字是 0 时，处理没有执行同时结束软元件也不变为 ON。

(3) 在每个站的一次扫描中只有一个 S.TO 指令可以执行。
如果执行条件在两个或更多地方不能同时使用，随后执行的 S.TO 指令由于自动握手而没有被处理。

(4) 根据目标 CPU 模块，可以被写入的数据数量有所不同。

CPU 模块	写入点数
基本型 QCPU	1 到 320
高性能型 QCPU、过程 CPU	1 到 256
通用型 QCPU	1 到 2048

要 点

也可以使用智能功能模块软元件执行至 CPU 共享内存的数据写入。
关于智能功能模块软元件，请参考以下手册：
• QCPU 用户手册（功能解说 / 程序基础篇）

运算错误

在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

(1) 指定数据在下列范围之外 （出错代码： 4101）

- 写入点数 (n4) 超出了设定数据的指定范围。
- 写入目标自站的 CPU 共享内存地址 (n2) 的开始超出 CPU 共享内存地址范围。
- 写入目标自站的 CPU 共享内存地址 (n2)+ 写入点数 (n4)，超出了 CPU 的共享内存地址范围。
- 存储写入数据的第一个软元件号 (n3) 加上写入点数 (n4)，超出软元件范围。

(2) 当为写入目标的自站 CPU 共享内存地址 (n2) 指定了自站 CPU 动作信息区域、系统区域或自站 CPU 刷新区域时。

（高性能型 QCPU、过程 CPU） （出错代码： 4101）

（基本型 QCPU、通用型 QCPU） （出错代码： 4111）

(3) 当自站 CPU 的起始 I/O 号 (n1) 不是自站 CPU 时。

（高性能型 QCPU、过程 CPU） （出错代码： 2107）

（基本型 QCPU、通用型 QCPU） （出错代码： 4112）

(4) 在 CPU 模块的起始 I/O 号指定的位置上未安装 CPU 模块时。 （出错代码： 2110）

(5) 在 n1 里指定的数字不包括 3E0H/3E1H/3E2H/3E3H （出错代码： 4100）

(6) 指定的指令不合适 （出错代码： 4002）

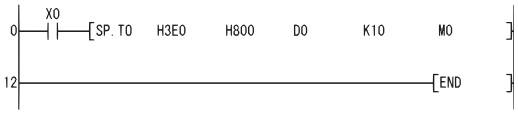
(7) 指定的软元件号码错误 （出错代码： 4003）

(8) 不能使用的软元件被指定 （出错代码： 4004）

程序示例

(1) 当 X0 为 ON 时，存储 10 点数据的程序从在 D0 开始，进入自站 CPU 共享内存的地址 800H。

[梯形图模式]



[列表模式]

步	指令	软元件				
0	LD	X0				
1	SP.TO	H3E0	H800	D0	K10	M0
12	END					

备 注

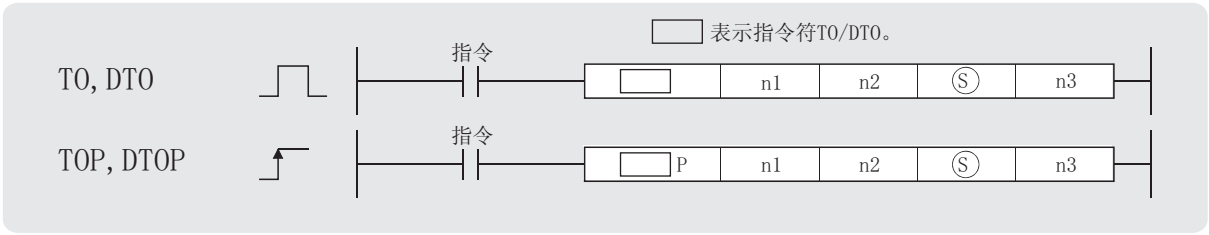
在 n1 里，指定代表 CPU 模块插槽的起始 I/O 号的 4 个 16 进制数字的上面三位。

	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 号	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

9.11.2 写入到自站 CPU 共享内存 (TO(P)、DTO(P))

通用型 UD　基本型 *1　高性能型　过程　冗余　QnA　Q4AR

*1: 系列号前五位数为 “04122”或之后的 Q00CPU/Q01CPU。



n1 : 自站 CPU 起始 I/O 号 (BIN16 位)
- 基本型 QCPU: 3E0H
- 通用型 QCPU: 3E0H 到 3E3H

n2 : 写入目标自站 CPU 的 CPU 共享内存地址 (BIN16 位)
- 基本型 QCPU: 192 到 511
- 通用型 QCPU: 2048 到 4095、 10000 到 24335 *2

Ⓢ : 要写入的数据或存储写入数据的软元件的起始号 (BIN16 位)

n3 : 写入数据块的数目 (BIN16 位)
- 基本型 QCPU: TO(P): 1 到 320、 DTP(P) : 1 到 160
- 通用型 QCPU: TO(P): 1 到 14336*2、 DTP(P) : 1 到 7168 *2

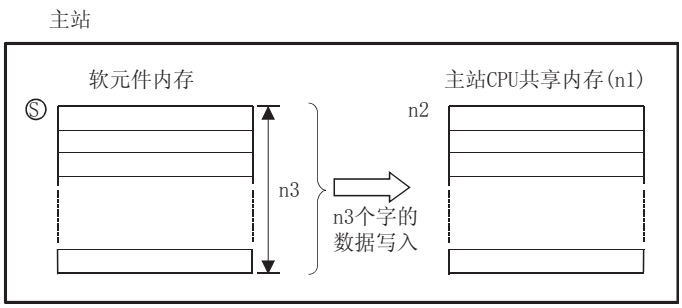
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它 U
	位	字		位	字				
n1	○			○				○	○
n2	○			○				○	--
Ⓢ	○			--				○	--
n3	○			○				○	--

*2: 根据多 CPU 高速通信功能的自动刷新设定范围，设定范围有所不同。

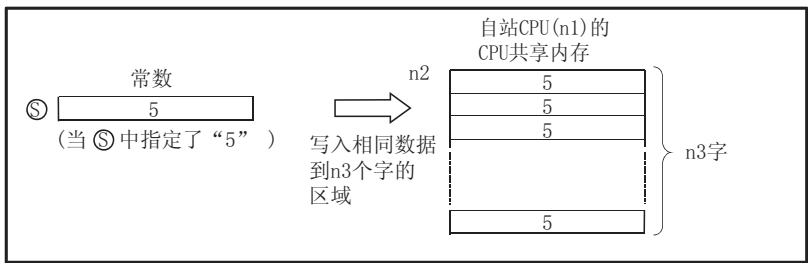
功 能

TO

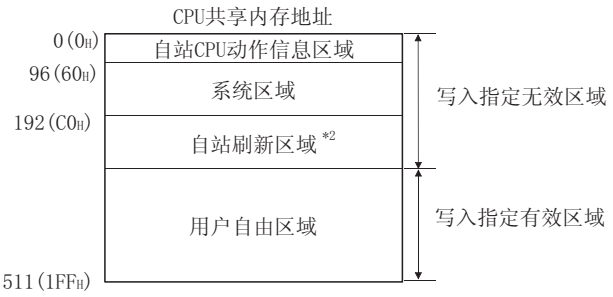
(1) 在软元件里写入 n3 个字的数据，从自站 CPU 模块里 Ⓢ 中的数字开始，到在 n2 中指定的 CPU 共享内存地址，和其后的自站 CPU 模块区域。



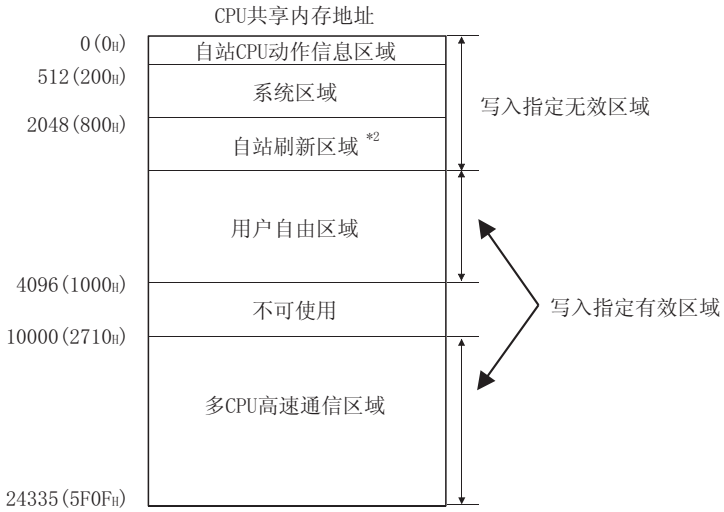
在 Ⓢ 中指定了常数时，将相同数据（指定给到 Ⓢ 的值）写入到指定的 CPU 共享内存起的 n3 个字的区域。



(a) 基本型 QCPU 的 CPU 共享内存地址



(b) 通用型 QCPU 的 CPU 共享内存地址 *3



*2: 当未进行自动刷新设定时，可以作为用户自由区域使用。
另外，当进行了自动刷新设定时，自动刷新传送范围后面的区域可以作为用户自由空间使用。

*3: 在 Q02UCPU 中，数据不能写入到多 CPU 高速通信区域。

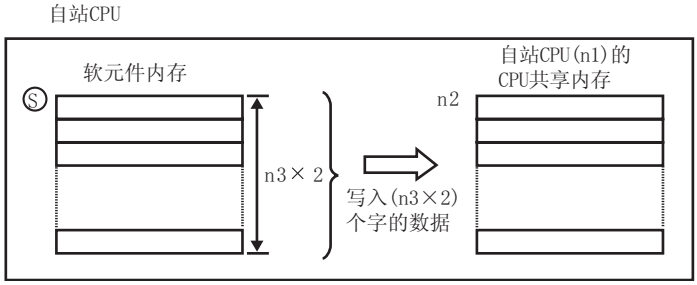
(2) 当写入数字是 0 时，没有执行处理。

(3) 根据目标 CPU 模块，写入的数据数量有所不同。

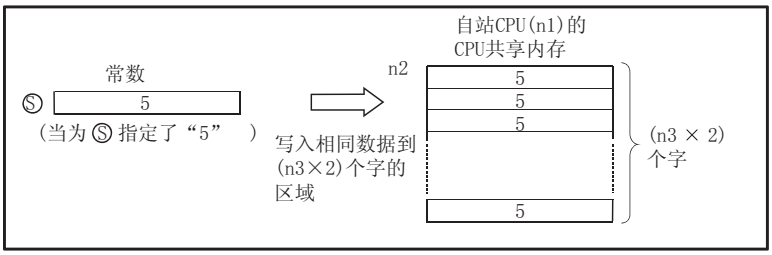
CPU 模块	写入点数
基本型 QCPU	1 到 320
通用型 QCPU	1 到 14336

DTO

(1) 将自站 CPU 模块的 Ⓢ 起的 (n3×2) 个字的软元件数据，写入到自站 CPU 模块的 n2) 中指定的 CPU 共享内存地址的后面。



在 Ⓢ 中指定了常数时，将相同数据（指定给到Ⓢ 的值）写入到指定的 CPU 共享内存起的 (n3 × 2) 字的区域。



(2) 当写入点数为 0 时，不执行处理。

(3) 根据写入目标 CPU 模块，可以被写入的数据数量有所不同。

CPU 模块	写入点数
基本型 QCPU	1 到 160
通用型 QCPU	1 到 7168

要 点

智能功能模块软元件也可以用来执行至 CPU 共享内存的数据写入。
关于智能功能模块软元件，请参考以下手册：
• QCPU 用户手册（功能解说 / 程序基础篇）

运算错误

在下列情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

(1) 指定数据在下列范围之外。 (出错代码： 4101)

- 写入点数 (n3) 超出了设定数据的指定范围。
- 写入目标自站 CPU 的 CPU 共享内存地址 (n2)+ 写入点数 (n3)，超过了 CPU 共享内存的范围。
- 存储写入数据 Ⓢ 的起始软元件号 + 写入点数 (n3) 超过了软元件范围。
- 写入目标自站 CPU 的 CPU 共享内存地址 (n2) 的起始中指定的值超过了写入有效区域。

(2) 写入目标自站 CPU 的 CPU 共享内存地址 (n2) 的起始是无效值。 (出错代码： 4111)

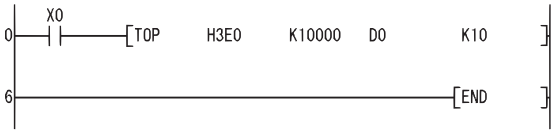
(3) (n1) 中指定的 I/O 号不是自站 CPU。(使用其它 CPU 的多 CPU 高速通信区域时除外。) (出错代码： 4112)

(4) 在 CPU 模块的起始 I/O 号指定的位置上未安装 CPU 模块。 (出错代码： 2110)

程序示例

(1) 当 X0 为 ON 时，下列程序从 D0 存储 10 点的数据到 1 号 CPU 的 CPU 共享内存的地址 10000 的后面。

[梯形图模式]

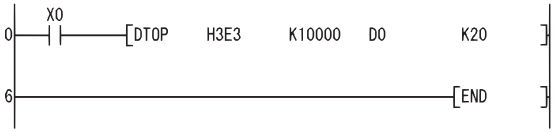


[列表模式]

步	指令	软元件				
0	LD	X0				
1	TOP	H3E0	K10000	D0	K10	
6	END					

(2) 当 X0 为 ON 时，下列程序从 D0 存储 20 点的数据到 4 号 CPU 的 CPU 共享内存的地址 10000 的后面。

[梯形图模式]



[列表模式]

步	指令	软元件				
0	LD	X0				
1	DTOP	H3E3	K10000	D0	K20	
6	END					

备 注

n1 由 4 位 16 进制数据的前三个数字指定，4 位 16 进制数据表示安装在 CPU 模块的插槽的起始 I/O 号。

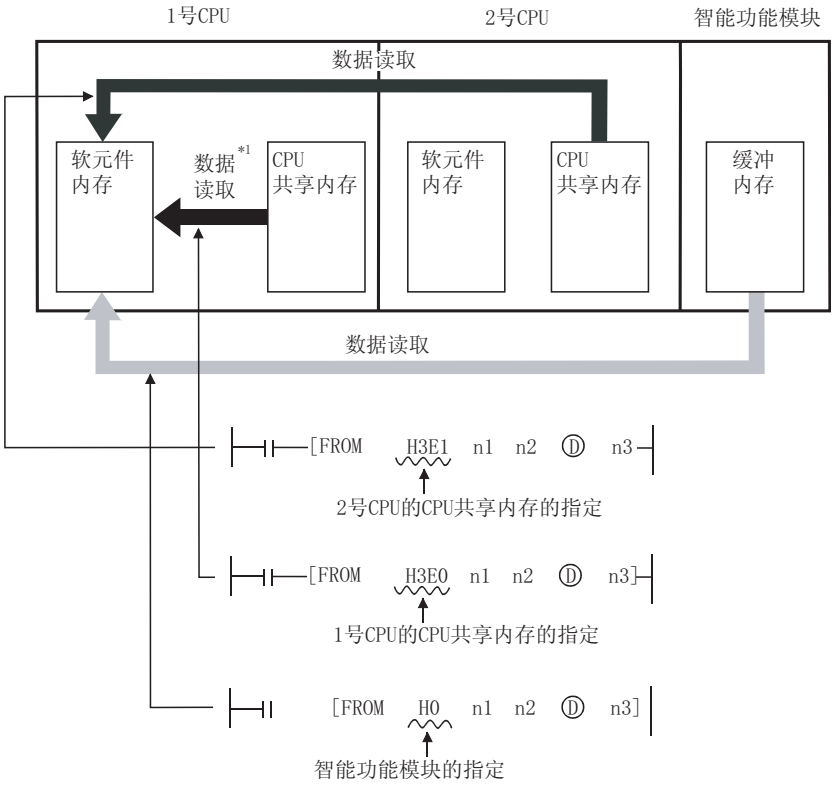
	CPU 插槽	0 号插槽	1 号插槽	2 号插槽
起始 I/O 号	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

9.12 从其他站的 CPU 共享内存中读取数据

多 CPU 系统的 FROM(P)/DFRO(P) 指令可以从以下内存中执行读取。

- 智能功能模块的缓冲内存
- 另一站 CPU 模块的 CPU 共享内存
- 自站 CPU 模块的 CPU 共享内存（只可对基本型 QCPU 和通用型 QCPU 执行）

下图表明当 FROM(P) 指令在 1 号 CPU 中执行时的处理。



*1 : 只能对基本型 QCPU 和通用型 QCPU 执行。

备 注

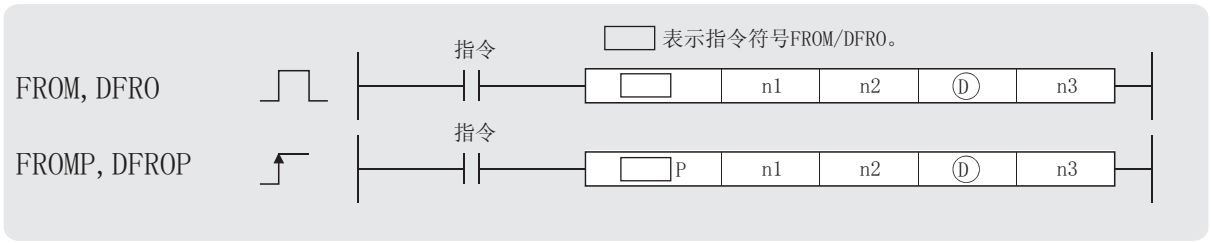
关于通过 FROM/DFRO 指令从智能功能模块的缓冲内存中读取数据的内容，请参阅 7.8.1 项。

9.12.1 从其他站的共享内存读取数据 (FROM(P)、DFRO(P))

通用型 UD　基本型 *1　高性能型 *2　过程　冗余　QnA　Q4AR

*1：系列号前五位数为“04122”或之后的 Q00CPU/Q01CPU。
*2：功能版本 B 或以后的版本。

(1) 使用基本型 QCPU、通用型 QCPU 时。



n1 ：读取目标 CPU 模块的起始 I/O 号 (BIN16 位)
- 基本型 QCPU：3E0H 到 3E2H
- 通用型 QCPU：3E0H 到 3E3H

n2 ：读取数据的起始地址 (BIN16 位)
- 基本型 QCPU：0 到 512
- 通用型 QCPU：0 到 4095、10000 到 24335 *3

Ⓓ ：存储读取数据的软元件的起始号 (BIN16 位)

n3 ：读取数据的数目 (BIN 16 位)
- 基本型 QCPU：FROM(P)：1 到 512、DFRO(P) ：1 到 256
- 通用型 QCPU：FROM(P)：1 到 14336*3、DRRO(P) ：1 到 7168 *3

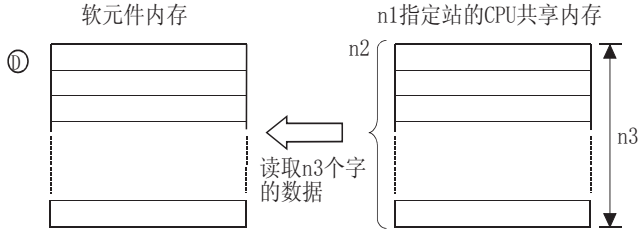
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字			K、H	U
n1	—	○		○				○	○
n2	—	○		○				○	—
Ⓓ	—	○		—				—	—
n3	—	○		○				○	—

*3：根据多 CPU 高速通信功能的自动刷新设定范围，设定范围有所不同。

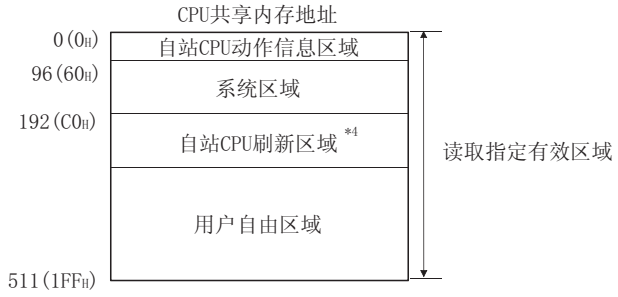
☆功 能

FROM

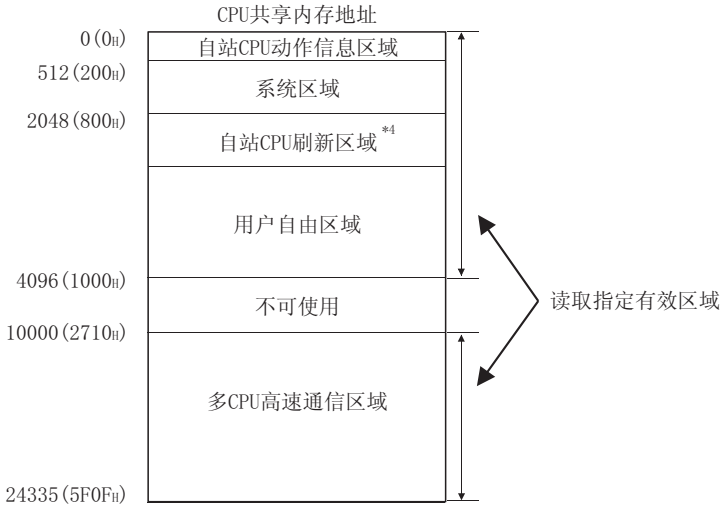
(1) 读取从由 n1 指定的 CPU 模块中，从由 n2 指定的 CPU 共享地址中读取 n3 个字的数据，同时存储在由 Ⓓ 指定的软元件其后的区域中。



(a) 基本型 QCPU 的 CPU 共享内存地址



(b) 通用型 QCPU 的 CPU 共享内存地址 *5



*4: 当未进行自动刷新设定时，可作为用户自由区域使用。
即使进行了自动刷新设定，自动刷新传送范围后面的区域也可作为用户自由空间使用。

*5: 在 Q02UCPU 中，不能从多 CPU 高速通信区域进行读取。

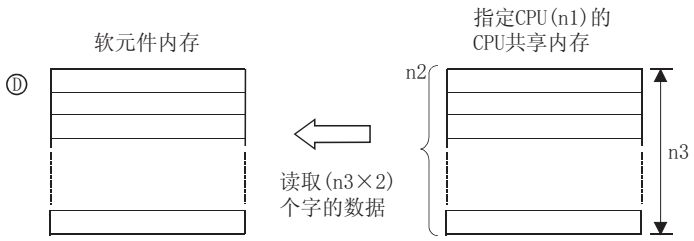
(2) 当 0 指定在 n3 里作为读取数据的数目时，不执行处理。

(3) 根据目标 CPU 模块，被读数据的数目有所不同。

CPU 模块	读取点数
基本型 QCPU	1 到 512
通用型 QCPU	1 到 14336

DFRO

(1) 从 n1 指定的 CPU 模块的 n2 指定的 CPU 共享内存地址读取 (n3 × 2) 个字的数据，并将其存储到 ⓓ 指定的软元件的后面。



(2) 当在 n3 中将 0 指定为将要读取的数据数目，将不执行处理。

(3) 根据目标 CPU 模块，可以被读取的数据数目有所不同。

CPU 模块	读取点数
基本型 QCPU	1 到 256
通用型 QCPU	1 到 7168

要 点

智能功能模块软元件也可以用来执行从 CPU 共享内存的数据读取。
关于智能功能模块软元件，请参考以下手册：
• QCPU 用户手册（功能解说 / 程序基础篇）

运算错误

在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

(1) 指定数据在下列范围之外。 （出错代码：4101）
- 执行读取的 CPU 共享内存地址 (n2) 的起始地址超出了 CPU 共享内存的范围。
- 执行读取的 CPU 共享内存地址 (n2)+ 读取点数 (n3) 超出了 CPU 共享内存范围。
- 读取数据存储软元件数目 (D)+ 读取点数 (n3) 超出了指定的软元件范围。

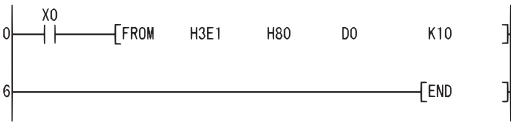
(2) 智能功能模块指定的起始 I/O 号是自站的号码。 （出错代码：2114）

(3) 智能功能模块和 CPU 模块在指定使用智能功能模块的输入输出。
起始号的位置不存在。 （出错代码：2110）

程序示例

(1) 当 X0 为 ON 时，下面的程序将 2 号 CPU 的 CPU 共享内存中从地址 C0H 开始的 10 个点的数据存储到 D0 和其后的区域内。

[梯形图模式]

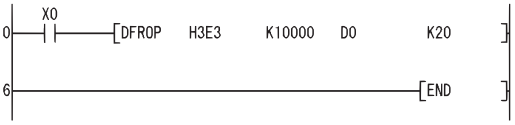


[列表模式]

步	指令	软元件			
0	LD	X0			
1	FROM	H3E1	H80	D0	K10
6	END				

(2) 当 X0 为 ON 时，下面的程序将 4 号 CPU 的 CPU 共享内存中从地址 10000 开始的 20 个点的数据存储到 D0 和其后的区域内。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	DFROP	H3E3	K10000	D0	K20
6	END				

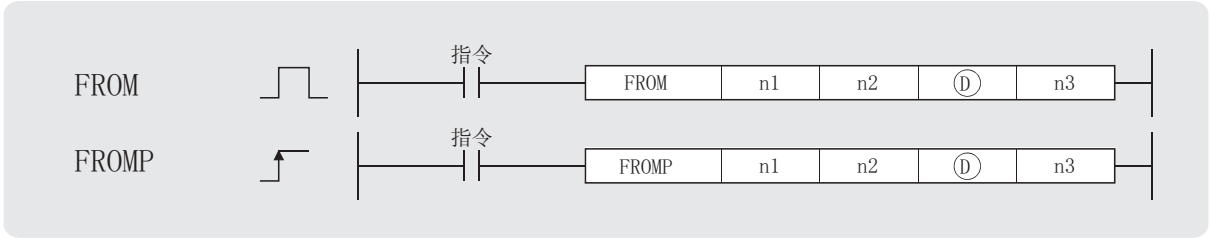
备 注

在 n1 里，指定代表 CPU 模块的插槽的起始 I/O 号的 4 个 16 进制数字的上面三位。

	CPU 插槽	插槽 0	插槽 1	插槽 2
起始 I/O 号	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

对于从 FROM 到 TO 指令，QCPU 提供了自动互锁。

(2) 使用高性能型 QCPU、过程 CPU 时



n1 ： 读取目标 CPU 模块的起始 I/O 号 (BIN16 位)

n2 ： 要读取的数据的地址 (BIN16 位)

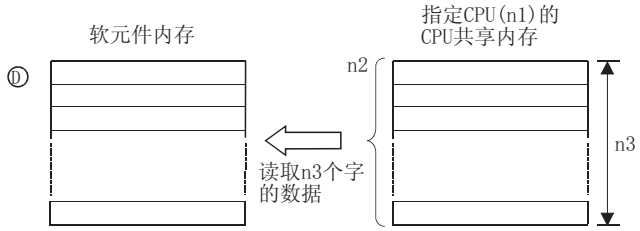
Ⓓ ： 存储读取的数据的软元件的起始号 (BIN16 位)

n3 ： 读取数据的数目 (BIN16 位)

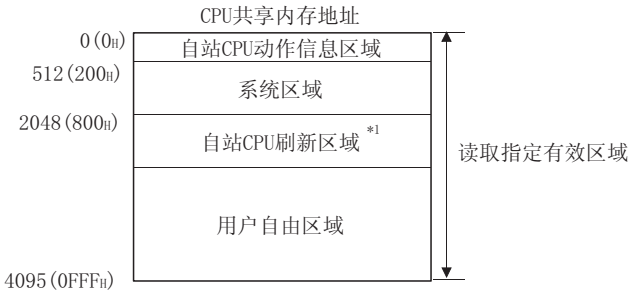
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它 U
	位	字		位	字				
n1	--	○		○				○	○
n2	--	○		○				○	--
Ⓓ	--	○		--				--	--
n3	--	○		○				○	--

功 能

(1) 从 n1 指定的 CPU 模块的 n2 中指定的 CPU 共享内存地址读取 n3 个字的数据，并将其存储到 Ⓓ 指定的软元件的后面区域。



高性能型 QCPU 和过程 CPU 的 CPU 共享内存地址



*1: 当未进行自动刷新设定时，可作为用户自由区域使用。
即使进行了自动刷新设定，自动刷新传送范围后面的区域也可作为用户自由空间使用。

(2) 当在 n3 中将 0 指定为将要读取的数据数目时，将不执行处理。

(3) 根据目标 CPU 模块，被读数据的数目有所不同。

模块	读取点数
高性能型 QCPU 过程 CPU	1 到 4096

要 点

智能功能模块软元件也可以用来执行 CPU 共享内存的数据读取。
关于智能功能模块软元件，请参考以下手册：
• QCPU 用户手册（功能解说 / 程序基础篇）

运算错误

在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

(1) 指定数据在下列范围之外。 （出错代码： 4101）
- 将执行读取的 CPU 共享内存的起始地址 (n2) 超出了 CPU 共享内存的范围。
- 读取数据的 CPU 共享内存的地址 (n2)+ 读取点数 (n3) 超出了 CPU 共享内存的范围。
- 读取数据存储软元件号 (D)+ 读取点数 (n3) 超出了指定软元件的范围。

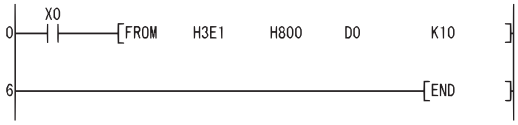
(2) 在 CPU 模块的起始 I/O 号指定的位置上未安装 CPU 模块。 （出错代码： 2110）

(3) 进行读取的 CPU 共享内存地址 (n2) 的起始是无效值。
(4097 到 9999) （出错代码： 4101）

程序示例

(1) 当 X0 为 ON 时，下面的程序将 2 号 CPU 的 CPU 共享内存中从地址 800H 开始的 10 个点的数据存储到 D0 和其后的区域内。

[梯形图模式]



[列表模式]

步	指令	软元件			
0	LD	X0			
1	FROM	H3E1	H800	D0	K10
6	END				

备 注

n1 由 4 位 16 进制数据的前三个数字指定，4 位 16 进制数据表示安装在 CPU 模块中的插槽的起始 I/O 号。

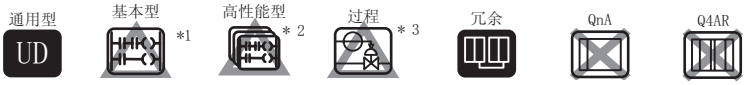
	CPU 插槽	0 号插槽	1 号插槽	2 号插槽
起始 I/O 号	3E00	3E10	3E20	3E30
n1	3E0	3E1	3E2	3E3

在 QCPU 中可以对 FROM 和 TO 指令进行自动互锁。

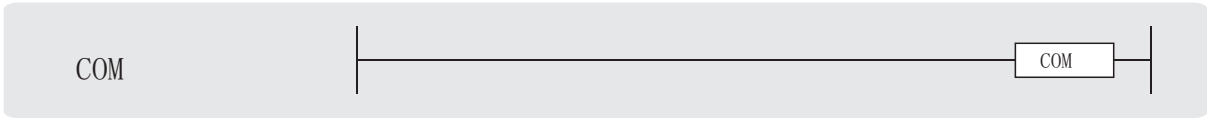
9.13 选择性刷新指令 (COM)

下列 CPU 模块的 COM 指令可以参考 7.6.9 项。

- 系列号 04121 或更早的基本类型 QCPU
- 系列号 04011 或更早的高性能类型 QCPU
- 序列号 07031 或以前的过程 CPU



*1: 序列号前五位为 “04122” 或以后。
*2: 序列号前五位为 “04012” 或以后。
*3: 序列号前五位为 “07032” 或以后。



设定数据	内部软元件		Z、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
--	--								

功 能

(1) 当 COM 指令执行时，将执行下列刷新操作。

- I/O 刷新
- CC-Link 刷新
- MELSECNET/G 刷新
- MELSECNET/H 刷新
- 智能功能模块的自动刷新
- 使用多 CPU 系统的 QCPU 标准区域的自动刷新
- 多 CPU 系统组外的全部模块的输入 / 输出数据的获取
- 使用多 CPU 系统的多 CPU 高速通信区域的自动刷新
- 与外围设备的通讯

备 注

在与外围设备通讯中，将执行下列操作。

- 其它站监视
- 通过串行通讯模块读取另一智能功能模块的缓冲内存

(2) 将 SM775 变为 OFF，以刷新除了 I/O 刷新之外的所有刷新条目。

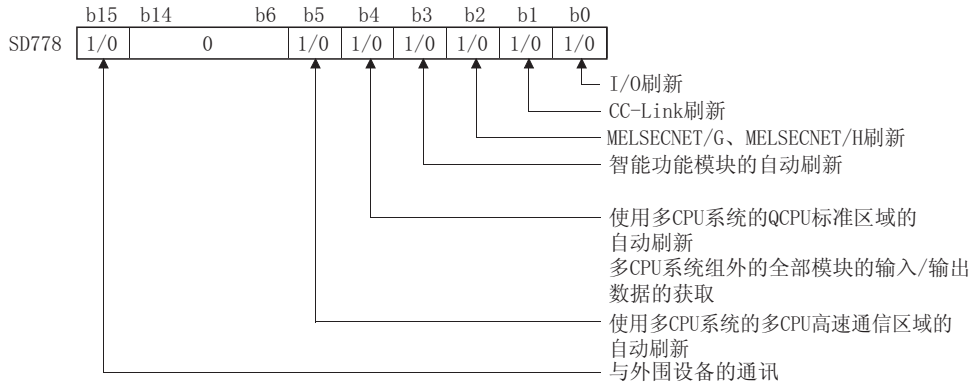
(3) 当选择刷新条目时

(a) 通过 SD778 选择刷新条目，同时设置 SM775 到 ON
下图说明了可以通过将 SM775 变为 ON 和使用 ON/OFF SD778 来指定的刷新条目 .

刷新项目	当 SM775 为 OFF	当 SM775 为 ON
I/O 刷新	不执行	执行或不执行可以选择
CC-Link 刷新	执行	
MELSECNET/G 刷新		
MELSECNET/H 刷新		
智能功能模块的自动刷新		
使用多 CPU 系统的 QCPU 标准区域的自动刷新		
多 CPU 系统组外的全部模块的输入 / 输出数据的获取		
使用多 CPU 系统的多 CPU 高速通信区域的自动刷新		
与外围设备的通讯		

(b) 通过 SD778 的 b0 ～ b5、b15 来选择刷新项目。
SD778 的各个位的执行 / 不执行可按如下方式指定 :

SD778 的位	执行	不执行
b0 到 b5	1	0
b15	0	1



例如

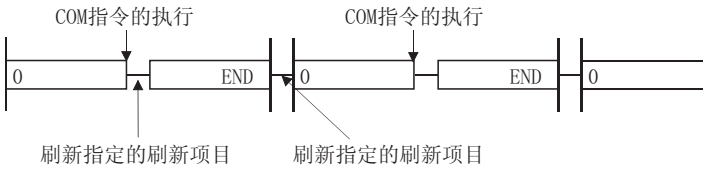
为了只使远程 I/O 站的发送 / 接收处理更快，只用 MELSECNET/H 刷新。
(只设定 SD778 的 b2 和 b15 到 1(SD778:H8004)。)

要 点

使用 COM 指令的多 CPU 间刷新在下列情况执行。

- 从其它 CPU 的接收动作: 当 SD778 的 b4(CPU 共享内存的自动刷新) 为 1 时。
- 从自站 CPU 的发送动作: 当 SD778 的 b15(与外围设备的通讯的执行 / 不执行) 为 0 时。

(4) 在 COM 指令执行的时候，CPU 模块中止了顺序程序的处理，同时刷新指定的刷新条目。



(5) 在顺序程序里 COM 指令可以使用任意次数。
然而，注意 SD778 中选择的与外围设备的通讯、刷新项目的刷新所用的时间会延长顺序程序的扫描时间。

(6) 只有使用通用型 QCPU 时，在 COM 指令的执行过程中才可以允许中断。
然而，注意如果刷新数据被中断程序等使用，有可能导致数据背离。

(7) 下列表列出了 COM 指令的刷新项目。

CPU 模块型号	功能版本	序列号	SM775	刷新项目
Q00JCPU Q00CPU Q01CPU	A	04021 或更早	OFF	刷新所有的刷新项目。
			ON	只与外围设备通讯。
	B	04022 或更早	OFF	刷新所有的刷新项目。
			ON	刷新由 SD778 选择的刷新项目。
Q02CPU	A	----	ON/OFF	刷新所有的刷新项目。
Q02HCPU Q06HCPU Q12HCPU Q25HCPU	B	04011 或更早	OFF	刷新所有的刷新项目。
			ON	只与外围设备通讯。
		04012 或更早	OFF	刷新所有的刷新项目。
			ON	刷新由 SD778 选择的刷新项目。
Q12PHCPU Q25PHCPU	C	07031 或更早	OFF	刷新所有的刷新项目。
			ON	只与外围设备通讯。
		07032 或更早	OFF	刷新所有的刷新项目。
			ON	刷新由 SD778 选择的刷新项目。
Q12PRHCPU Q25PRHCPU	D	----	OFF	刷新所有的刷新项目。
			ON	刷新由 SD778 选择的刷新项目。
Q02UCPU Q03UDCPU Q04UDHCPU Q06UDHCPU	B	----	OFF	刷新由 SD778 选择的刷新项目。
			ON	刷新所有的刷新项目。

要 点

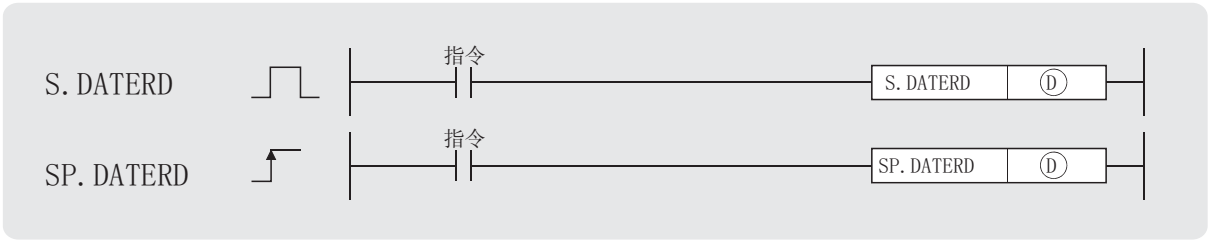
1. 在低速执行类型程序、恒定扫描执行类型程序或中断程序里，COM 指令不能使用。
2. 对于冗余 CPU，在使用 COM 指令时有所限制。
有关详细内容，请参照以下手册。
• QnPRHCPU 用户手册（冗余系统篇）

9.14 扩展时钟指令

9.14.1 读取扩展时钟数据 (S(P).DATERD)

通用型 UD　基本型　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：序列号前五位为“07032”或以后。

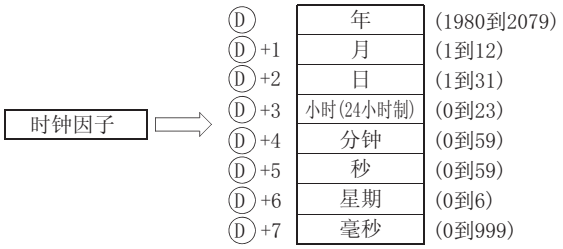


Ⓓ：存储读取时钟数据的软元件的起始号 (BIN 16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓓ	--	○		--					

功 能

(1) 从 CPU 模块的时钟因子读取 “年、月、日、小时、分钟、秒、星期和毫秒”，并将其以 BIN 值存储到 Ⓓ 指定的软元件的后面。



(2) Ⓓ 中的 “年”以 4 位年份存储。

(3) Ⓓ+6 中的 “星期”存储为 0 到 6，表示星期日到星期六。

星期	星期日	星期一	星期二	星期三	星期四	星期五	星期六
存储数据	0	1	2	3	4	5	6

(4) 闰年自动调整。

运算错误

(1) 在以下情况下将发生运算错误，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

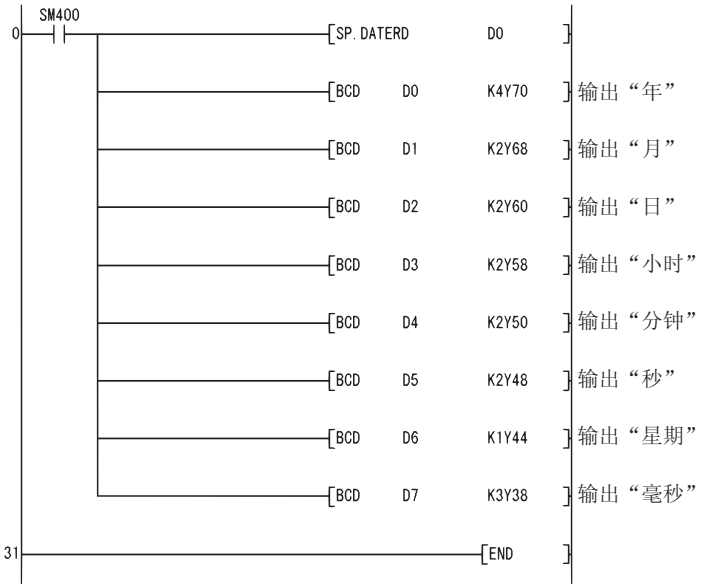
- 由 Ⓓ 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU)　　(出错代码： 4101)

程序示例

(1) 下列程序将以下时钟数据输出为 BCD 值：

年 ..... Y70 到 Y7F
月 ..... Y68 到 Y6F
日 ..... Y60 到 Y67
小时 ..... Y58 到 Y5F
分钟 ..... Y50 到 Y57
秒 ..... Y48 到 Y4F
星期 ..... Y44 到 Y47
毫秒 ..... Y38 到 Y43

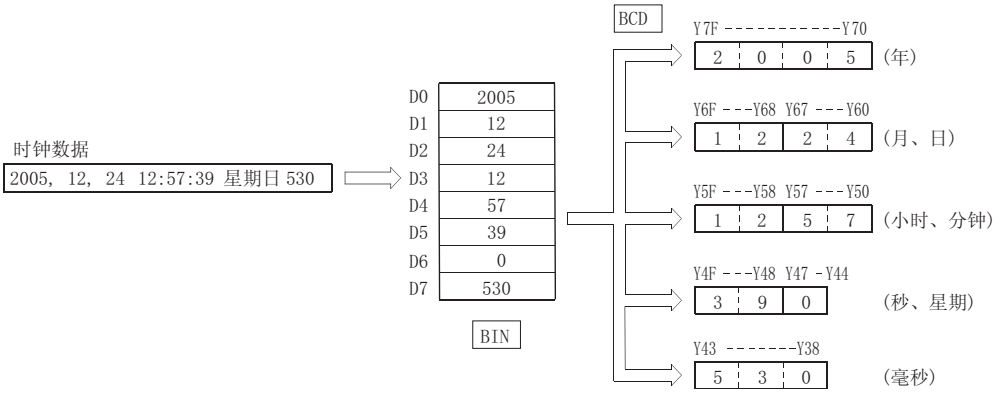
[梯形图模式]



[列表模式]

步	指令	软元件	
0	LD	SM400	
1	SP.DATERD		D0
7	BCD	D0	K4Y70
10	BCD	D1	K2Y68
13	BCD	D2	K2Y60
16	BCD	D3	K2Y58
19	BCD	D4	K2Y50
22	BCD	D5	K2Y48
25	BCD	D6	K1Y44
28	BCD	D7	K3Y38
31	END		

[动作]



注意事项

(1) 本指令用来读取时钟数据并存储到指定的软元件。即使为 CPU 模块设定的时钟数据是错误的，也会执行。（例如：二月 30 日）

当使用 DATEWR 指令或 GX Developer 设定时钟数据时，确认要设置正确的数据。

(2) 读取毫秒的时钟数据时的误差最多为 2ms。(CPU 模块中的时钟因子记忆的数据和用本功能读取的数据的误差。)

(3) 在为位软元件指定位数时，只有在满足下列 (a) 和 (b) 条件时才可以使用。

(a) 位数指定 :K4

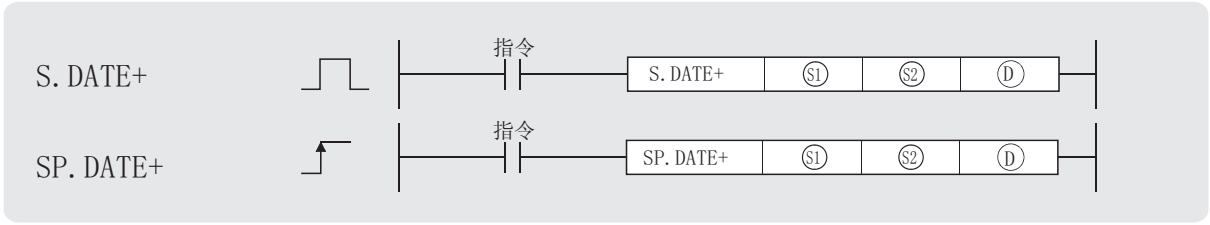
(b) 软元件的起始 :16 的倍数

当未满足上述 (a) 和 (b) 条件时，将会出现 INSTRCT CODE ERR.（出错代码 :4004)。

9.14.2 扩展时钟数据的加法运算 (S(P).DATE+)

通用型 UD　基本型　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：序列号前五位为“07032”或以后。



ⓢ1 ：存储进行了加法运算的时钟数据的软元件的起始号（BIN16 位）

ⓢ2 ：存储进行加法运算的时间数据的软元件的起始号（BIN16 位）

Ⓓ ：存储时钟（时间）数据加法结果的软元件的起始号（BIN16 位）

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
ⓢ1	—	○		—					
ⓢ2	—	○		—					
Ⓓ	—	○		—					

功 能

(1) 将 ⓢ2 指定的时间数据，加到 ⓢ1 指定的时钟数据上，并将结果存储到 Ⓓ 指定的软元件后的后面。

		设定数据				设定数据				设定数据
ⓢ1	小时	(0到23)		ⓢ2	小时	(0到23)		Ⓓ	小时	(0到23)
ⓢ1+1	分钟	(0到59)		ⓢ2+1	分钟	(0到59)		Ⓓ+1	分钟	(0到59)
ⓢ1+2	秒	(0到59)	+	ⓢ2+2	秒	(0到59)	⇨	Ⓓ+2	秒	(0到59)
ⓢ1+3	—			ⓢ2+3	—			Ⓓ+3	—	
ⓢ1+4	毫秒	(0到999)		ⓢ2+4	毫秒	(0到999)		Ⓓ+4	毫秒	(0到999)

例如，将时间 7:48:10:500 加到 6:32:40:875 时会出现下列结果：

ⓢ1	小时：6		ⓢ2	小时：7		Ⓓ	小时：14
ⓢ1+1	分钟：32		ⓢ2+1	分钟：48		Ⓓ+1	分钟：20
ⓢ1+2	秒：40	+	ⓢ2+2	秒：10	⇨	Ⓓ+2	秒：51
ⓢ1+3	—		ⓢ2+3	—		Ⓓ+3	—
ⓢ1+4	毫秒：875		ⓢ2+4	毫秒：500		Ⓓ+4	毫秒：375

(2) 如果时间相加的结果超过了 24 小时，将会从和中减去 24 小时后作为最终结果。

例如，将时间 20:20:20:500 加到 14:20:30:875 时，结果不是 34:40:51:375，而是 10:40:51:375。

(S1)	小时：14		(S2)	小时：20		(D)	小时：10
(S1)+1	分钟：20		(S2)+1	分钟：20		(D)+1	分钟：40
(S1)+2	秒：30	+	(S2)+2	秒：20	⇨	(D)+2	秒：51
(S1)+3	—		(S2)+3	—		(D)+3	—
(S1)+4	毫秒：875		(S2)+4	毫秒：500		(D)+4	毫秒：375

要 点

(S1)+3、(S2)+3 和 (D)+3 的软元件不用于运算。

对于 S(P).DATERD 指令读取的时钟数据可以直接相加。

(D)	小时
(D)+1	分钟
(D)+2	秒
(D)+3	星期
(D)+4	毫秒

当使用S(P).DATERD指令读取时钟数据时，星期会插入到“秒”和“毫秒”之间。

由于在S(P).DATE+指令中不对星期执行计算，所以数据可以直接相加。

运算错误

(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 由 (S1) 和 (S2) 设定的数据超出了范围。(参见功能 (1)) (出错代码： 4100)
- 由 (S1) 、 (S2) 和 (D) 指定的软元件超出了相应软元件的范围。(只对于通用型 QCPU。) (出错代码： 4101)

注意事项

(1) 位软元件的位数指定只有在满足下列 (a) 和 (b) 条件时才可以使用。

(a) 位数指定 :K4

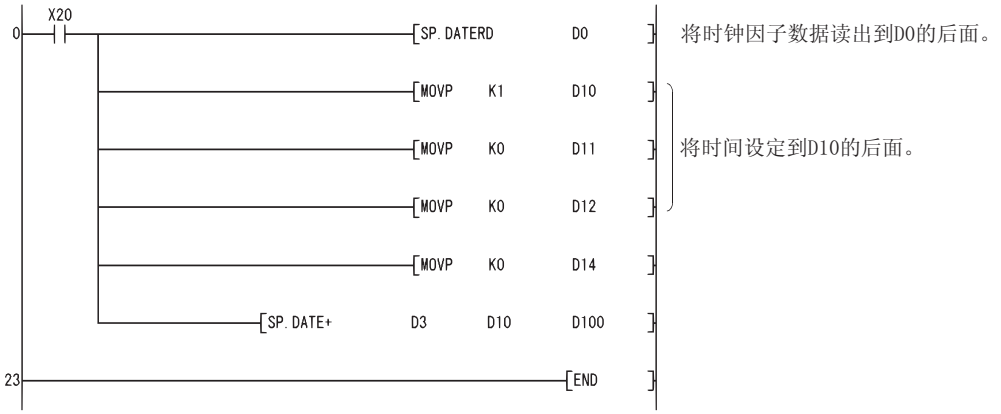
(b) 软元件的起始 :16 的倍数

当未满足上述 (a) 和 (b) 条件时，将会出现 INSTRCT CODE ERR.(出错代码 :4004)

程序示例

(1) 当 X20 为 ON 时，下列程序在从时钟因子读取的时钟数据上加 1 小时，并将结果存储到 D100 的后面的区域。

[梯形图模式]

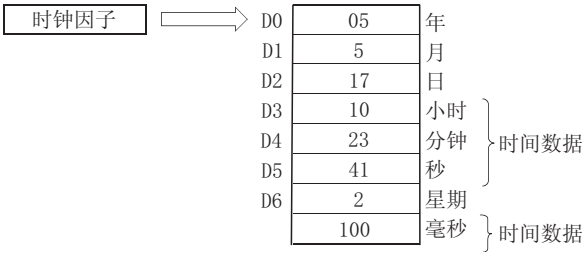


[列表模式]

步	指令	软元件		
0	LD	X20		
1	SP.DATERD		D0	
7	MOVP	K1	D10	
9	MOVP	K0	D11	
11	MOVP	K0	D12	
13	MOVP	K0	D14	
15	SP.DATE+	D3	D10	D100
23	END			

[动作]

• 通过 SP.DATERD 指令的时间数据的读取



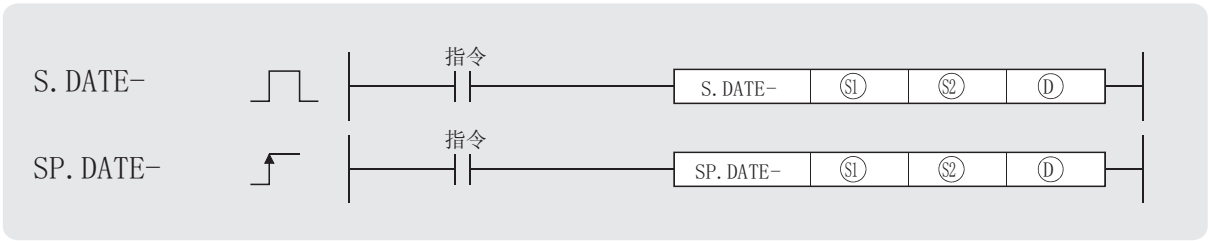
• 通过 SP.DATE+ 指令相加



9.14.3 扩展时钟数据的减法运算 (S(P).DATE-)

通用型 UD　基本型　高性能型 *1　过程 *1　冗余 *1　QnA　Q4AR

*1：序列号前五位为“07032”或以后。



(S1)：存储被执行减法运算的时钟数据的软元件的起始号 (BIN16 位)

(S2)：存储进行减法运算的时间数据的软元件的起始号 (BIN16 位)

(D)：存储时钟 (时间) 数据减法结果的软元件的起始号 (BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
(S1)	--	○		--					
(S2)	--	○		--					
(D)	--	○		--					

功 能

(1) 将 (S1) 中指定的时间数据减去 (S2) 中指定的时钟数据，并将结果存储到 (D) 指定的软元件号的后面。

		设定数据			设定数据			设定数据
(S1)	小时	(0到23)	(S2)	小时	(0到23)	(D)	小时	(0到23)
(S1)+1	分钟	(0到59)	(S2)+1	分钟	(0到59)	(D)+1	分钟	(0到59)
(S1)+2	秒	(0到59)	(S2)+2	秒	(0到59)	(D)+2	秒	(0到59)
(S1)+3	—		(S2)+3	—		(D)+3	—	
(S1)+4	毫秒	(0到999)	(S2)+4	毫秒	(0到999)	(D)+4	毫秒	(0到999)

例如，从时钟时间 10:40:20:875 中减去时钟时间 3:50:10:500 时，其运算如下：

(S1)	小时：10	(S2)	小时：3	(D) 小时：6
(S1)+1	分钟：40	(S2)+1	分钟：50	(D)+1 分钟：50
(S1)+2	秒：20	(S2)+2	秒：10	(D)+2 秒：10
(S1)+3	—	(S2)+3	—	(D)+3 —
(S1)+4	毫秒：875	(S2)+4	毫秒：500	(D)+4 毫秒：375

(2) 如果减法结果是负数，则将其加上 24 小时后作为最终运算结果。

例如，从 4:50:32:875 中减去时钟时间 10:42:12:500 时，结果不是 -6:8:20:375，而是 18:8:20:375。

(S1)	小时：4		(S2)	小时：10		(D)	小时：18
(S1)+1	分钟：50		(S2)+1	分钟：42		(D)+1	分钟：8
(S1)+2	秒：32	—	(S2)+2	秒：12	⇨	(D)+2	秒：20
(S1)+3	—		(S2)+3	—		(D)+3	—
(S1)+4	毫秒：875		(S2)+4	毫秒：500		(D)+4	毫秒：375

要 点

(S1)+3、(S2)+3 和 (D)+3 的软元件不用于运算。

用 S(P).DATERD 指令读取的时钟数据可以直接相减。

(D)	小时
(D)+1	分钟
(D)+2	秒
(D)+3	星期
(D)+4	毫秒

当使用S(P).DATERD指令读取时钟数据时，星期会插入到“秒”和“毫秒”之间。

由于在S(P).DATE-指令中不对星期执行计算，所以数据可以直接相减。

运算错误

(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码存储在 SD0 中。

- 由 (S1) 和 (S2) 设定的数据超出了范围。(参见功能 (1)) (出错代码： 4100)
- 由 (S1)、(S2) 和 (D) 指定的软元件超出了相应软元件的范围。
  (只对于通用型 QCPU。) (出错代码： 4101)

注意事项

(1) 位软元件的位数指定只有在满足下列 (a) 和 (b) 条件时才可以使用。

(a) 位指定 :K4

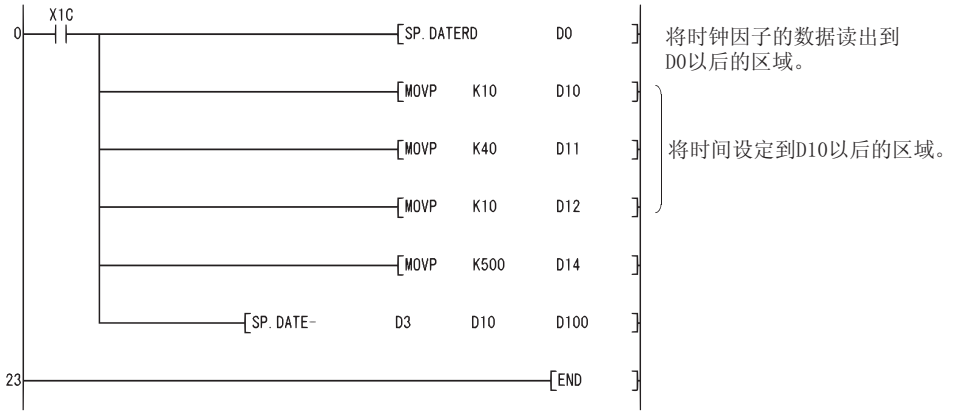
(b) 软元件的起始 :16 的倍数

当未满足上述 (a) 和 (b) 条件时，将会出现 INSTRCT CODE ERR.(出错代码 :4004)

程序示例

(1) 当 X1C 为 ON 时，下列程序从读取自时钟因子的时钟数据中减去存储在 D10 的后面的时间数据，并将结果存储到 D100 的以后的区域。

[梯形图模式]

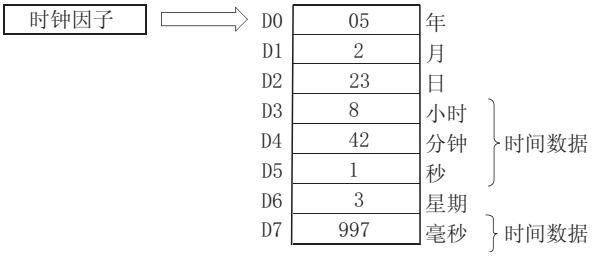


[列表模式]

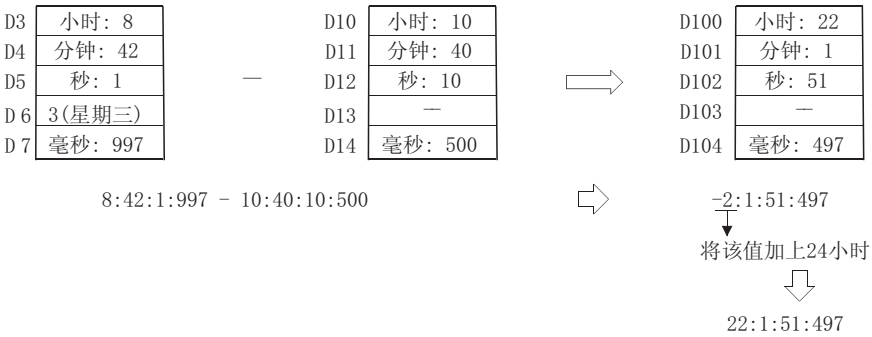
步	指令	软元件		
0	LD	X1C		
1	SP.DATERD		D0	
7	MOVP	K10	D10	
9	MOVP	K40	D11	
11	MOVP	K10	D12	
13	MOVP	K500	D14	
15	SP.DATE-	D3	D10	D100
23	END			

[动作]

• 通过 SP.DATERD 指令进行的时间数据的读取



• 用 SP.DATE- 指令相减

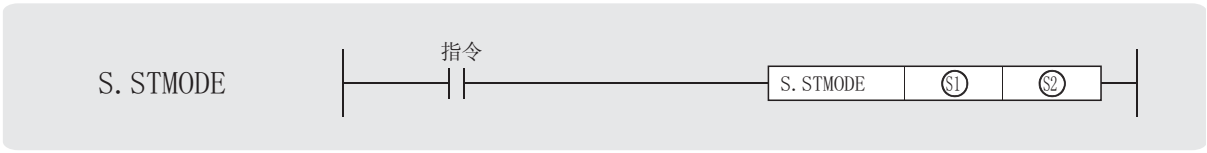


10 冗余系统指令（用于 Q4ARCPU）

分类	处理内容	参考章节
CPU 启动时动作模式设定指令	在 (S1) 中指定通过电源 ON 启动 Q4ARCPU 时，是否启动之前清除 Q4ARCPU 的软元件。	10.1 节
CPU 转换时动作模式设定指令	在 (S1) 中指定将控制从控制系统切换到待机系统时，是否启动之前清除 Q4ARCPU 的软元件。	10.2 节
数据追踪指令	在 END 指令的处理过程中，为了与存储在从 (S) 中指定的软元件后面的参数块数据内容保持一致，执行软元件内存的追踪。	10.3 节
缓冲内存批量刷新指令	为了与存储在 (S) 中指定的软元件的后面的参数块数据内容保持一致，批量读取 / 写入特殊功能模块的缓冲内存的内容。	10.4 节

10.1 CPU 启动过程中的操作模式设定指令 (S.STMODE)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



(S1) : 动作模式设定 (BIN16 位)

(S2) : 初始启动切换断电时间 (BIN16 位)

设定数据	内部软元件 位	内部软元件 字	R、ZR	J□\□ 位	J□\□ 字	U□\G□	Zn	常数 K、H	其它
(S1)	--							○	--
(S2)	--							○	--

功 能

(1) 当通过电源 ON 启动 Q4ARCPU 时，可以指定在启动时是否清除 Q4ARCPU 的软元件。使用 (S1) 进行此指定。

(2) 当指定了热启动模式，在某一特定时间出现了电源暂时中断，可以执行自动数据清除和重新启动。在这种情况下，在 (S2) 下指定转换电源超出时间。

(3) 当电源打开时执行这个指令。
由于这个原因，即使指令触点关闭也没有问题。这个指令触点变成虚触点。
当指令触点在程序执行时打开，NOP 处理将执行。

(4) 在每个系统里只能建立其中一个指令。
如果有多重程序文件，这个指令只能建立在一个文件里。
如果其中的指令多于一个，操作不能保证。

(5) 在 (S1) 里通过设定 0 或 1 可以执行。
0: 初始启动模式 (清除锁存范围外的软元件)
1: 热启动模式 (不清除 *1)

*1 : 清除下列信息
- 变址寄存器
- 信号流 (操作结果)

此外，预设特殊继电器 SM 和特殊寄存器 SD。

(6) 在 (S2) 里可以从 0 到 65535 进行指定 (单位 S)
当指定为 0 时，初始启动模式转换不能执行。
如果一个超过 32767 的数字被设定，在执行时使用 16 进制数。

运算错误

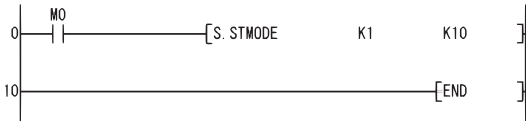
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 当超过指定允许范围的值被指定。 (出错代码 : 4104)

程序示例

(1) 当电源打开同时设定暂时转换电源超出时间为 10s 时，程序在热启动模式下启动 CPU 模块。

[梯形图模式]

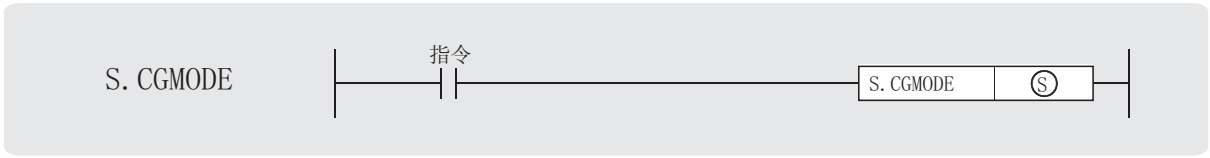


[列表模式]

步	指令	软元件	
0	LD	M0	
1	S.STMODE	K1	K10
10	END		

10.2 CPU 切换时间操作模式设定指令 (S.CGMODE)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 动作模式设定 (BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--							○	--

功 能

(1) 当控制从控制系统切换到待机 (备用) 系统时，本指令可以指定在切换时是否清除 Q4ARCPU 的软元件。该指定是在 Ⓢ 中进行。

(2) 当电源打开时这个指令从 STOP 转变到 RUN 执行。

因为这个原因，即使这个指令触点关闭也没有问题。这个指令触点将变成虚触点。
当指令触点在程序执行时打开，NOP 处理将执行。

(3) 在一个系统里只能建立其中一个指令。

即使有多重程序文件，只能在一个文件里建立这个指令。
如果其中不只一个指令存在，操作不能保证。

(4) 在 Ⓢ 里设定 0 或 1。

0: 初始启动模式 (清除闭锁范围外的软元件)
1: 热启动模式 (不清除 *1)

*1: 使全部信号流 (运算结果) 为 ON。
预置特殊继电器 SM 和特殊寄存器 SD。

运算错误

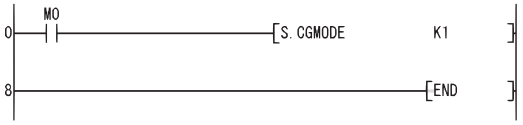
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

• 当一个超过指定范围的值被指定 (出错代码 : 4104)

程序示例

(1) 当从控制系统切换到备用系统，这个程序在热启动模式下启动 CPU 模块。

[梯形图模式]

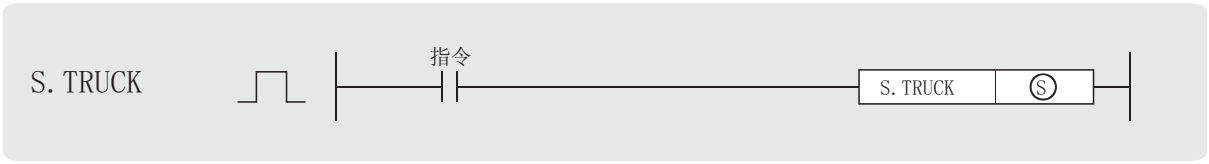


[列表模式]

步	指令	软元件
0	LD	M0
1	S.CGMODE	K1
8	END	

10.3 数据追踪指令 (S. TRUCK)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ: 参数块的起始软元件 (BIN16 位)

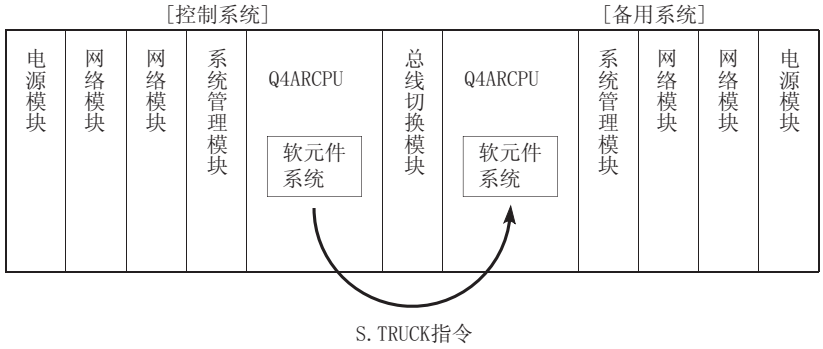
设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	--	○*1	○	--					

*1: 仅为锁存的软元件。

功 能

(1) 在本指令执行的各个扫描的 END 处理时，Q4ARCPU 跟随 Ⓢ 中指定的软元件中存储的参数块数据内容，执行软元件内存的追踪。*2

*2: 追踪（热备）是指，将控制系统 Q4ARCPU 的软元件内存的内容传送到待机（备用）系统 Q4ARCPU 的软元件内存中的功能。追踪是在 END 处理时执行。



(2) 当电源打开同时系统从复位中启动，Q4ARCPU 读取参数块内容。（因此，如果参数块内容改变，系统必需重新启动。）

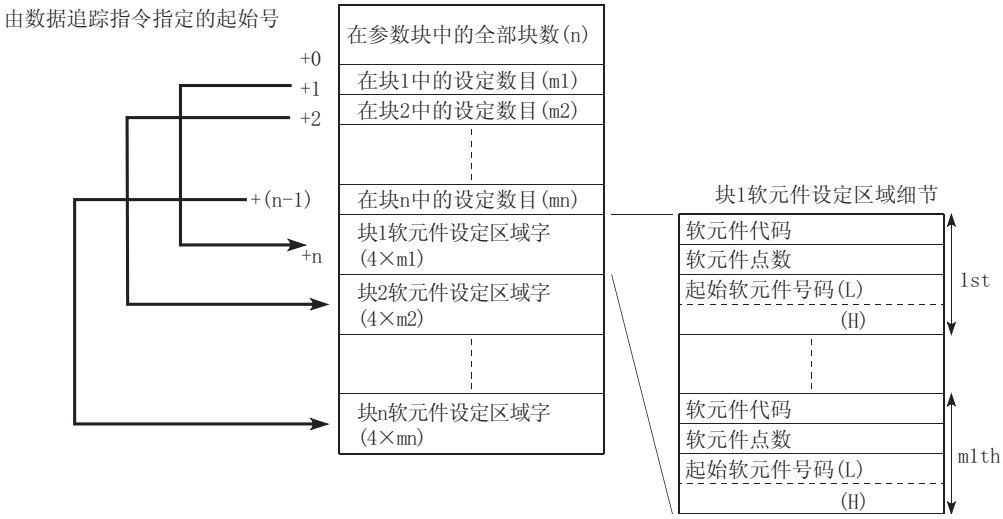
参数块由多个块构成。在指定要执行的块特殊继电器从 SM1520 到 SM1583 执行和特殊继电器关闭后，执行这个指令。

(3) 在一个系统里只能建立其中一个指令

即使有多个程序文件，只能在一个文件里建立这个指令。

如果多于一个文件存在，操作不能保证。

(4) 参数块有如下的构成



该块设定了软元件存储类型、触点、起始号码等。
块追踪对应于传送触发器 SM1520 到 1583 的为 ON 的传送触发器块的内容 (SM1520 → 块 1、SM1521 → 块 2、．．．、SM1583 → 块 64)。

要 点

数据追踪指令传送触发器 SM1520 到 SM1583 可以使用同样的传送触发器号码作为缓冲存储器批更新指令。

参数块设定指令内容

(a) 参数块的全部块数
参数块是一个多个块的集合。设定有多少块可以使用。

(b) 设置块 n 的数目
对于每个块的软元件设定区域，设定了设定软元件的数目。

(c) 块 n 软元件设置区域
这个可以设定实际追踪数据的软元件存储器的类型、点数、起始软元件号码。
1) 软元件代码设定软元件存储器的类型
设定如下：

软元件	软元件代码	软元件	软元件代码	软元件	软元件代码	软元件	软元件代码
X	0	B	5	C	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	V	7	W	12	SW	17
L	3	ST	8	R	13	SM	18
S	4	T	9	ZR	14	SD	19

T/C 包括触点号码、线圈和当前值。

2) 触点的软元件号码设定了将要追踪的软元件号码。
设定可以在十进制或十六进制下执行。位软元件设定为 16 的倍数。
3) 起始软元件号码设定软元件存储器的起始号码。
设定可以在十进制或十六进制下执行。位软元件设定为 16 的倍数 (0、16、32...)

备 注

当设定参数块时，应用下列限制。

- m1 + m2 + ... mn ≦ 2048。 n ≦ 64
- 当设定区域从 m1 到 mn 是 0 时，块的数目设定为 m=0。
对于设定不能执行的块数字设定为 0，使得有可能跳越一个块。
- 在一个扫描的 END 的处理时可追踪的一个块中的点数为，块的总追踪点数 =48k 字。
如果超过了 48K 字，将检测出错误，且该追踪不能执行。
- 当指定一个位软元件作为执行追踪的软元件，设定软元件的点数和起始软元件号码为 16 的倍数。.
- 当计时器和计算器作为追踪的软元件被指定，下面的公式用来计算追踪软元件的实际数目。
追踪软元件的点数 = 设定软元件的点数 × (1 + 1/8)

(5) 追踪执行模式的两种类型如下。可以使用继电器 SM1518 来选择。
在扫描 END 处理让 SM1518 ON/OFF 后，选择内容有效。

(a) 批通讯模式 (SM1518: 为 OFF 时)
如果当追踪执行时，备用系统使用追踪存储器时，控制系统在等待备用系统处理结束后执行追踪处理。
如果控制系统 CPU 产生追踪处理等待时间，这个时间累积将增加扫描时间。

(b) 在重复模式里 (SM1518: 为 ON 时)
当追踪执行时，备用系统使用追踪存储，控制系统在没有执行追踪处理的情况下，重复执行 END 处理。
当追踪处理被重复时，下列追踪要求不能接收，控制系统 CPU 不能产生追踪处理等待时间，所以扫描时间不会延长。

(6) 当一个特定的块追踪处理已经结束，Q4ARCPU 的追踪结束标志变为 ON 1 个扫描周期。
追踪结束标志对应每个块和从 SM1712 到 SM1775 的特殊继电器。
(块 1→SM1712、块 2→SM1713, ..., 块 64→SM1775)

运算错误

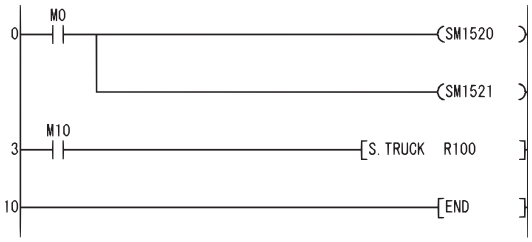
(1) 在下列发生运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 即使当文件寄存器 R 在指定的参数块被指定，文件寄存器的文件也不存在 (出错代码： 2402)
- 当超过指定范围的值被指定。 (出错代码： 4104)
- 当追踪软元件的点数超过 48K 个字。 (出错代码： 4104)

程序示例

(1) 该程序从存储在 R100 的参数块 1 和 2 执行追踪。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	M0
1	OUT	SM1520
2	OUT	SM1521
3	LD	M10
4	S.TRUCK	R100
10	END	

(2) 这是个当内部继电器 M0 到 M95、M320 到 M639 和数据寄存器 D0 到 D29、D600 到 D699 被追踪的示例参数块。

R100	K2	参数块总块数	
R101	K2	块1的设定数目	
R102	K2	块2的设定数目	
R103	K2	软元件代码(M)	块1软元件设定区域
R104	K96	软元件点数(96点)	
R105	K0	起始软元件号码(M0)	
R106	K0		
R107	K2	软元件代码(M)	
R108	K320	软元件点数(320点)	
R109	K320	起始软元件号码(M320)	
R110	K0		
R111	K11	软元件号码(D)	块2软元件设定区域
R112	K30	软元件点数(30点)	
R113	K0	起始软元件号码(D0)	
R114	K0		
R115	K11	软元件代码(D)	
R116	K100	软元件点数(100点)	
R117	K600	起始软元件号码(D600)	
R118	K0		

(a) 块 1 软元件设定区域
指定内部继电器 M0 到 M95、M320 到 M639。

(b) 块 2 软元件设定区域
指定数据寄存器 D0 到 D29、D600 到 D699。

10.4 缓冲内存批刷新指令 (S. SPREF)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



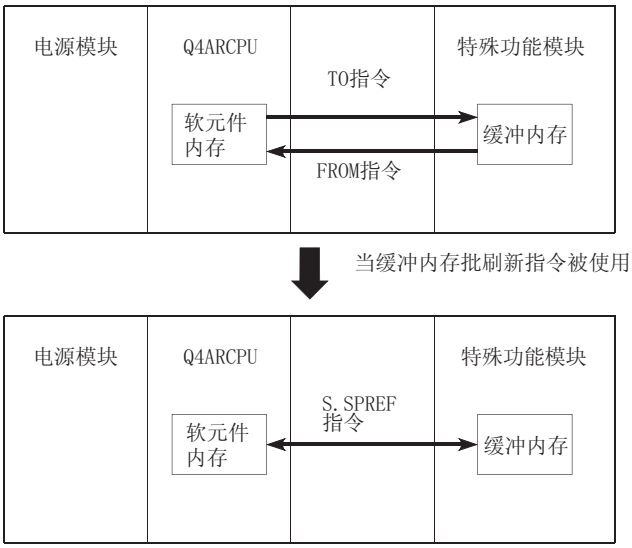
Ⓢ : 参数块的起始软元件 (BIN16 位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数	其它
	位	字		位	字				
Ⓢ	--	○*1	○	--					

*1: 仅锁存软元件。

功 能

(1) 特殊功能模块缓冲内存的内容与存储在 Ⓢ 指定的软元件和其后区域里的参数块数据的内容相一致，都是批读取 / 写入。



通过FROM指令/TO指令对于每个指令，特殊功能模块缓冲内存的读取/写入都执行。

S. SPREF指令在读取/写入特殊功能模块缓冲内存的同时执行。

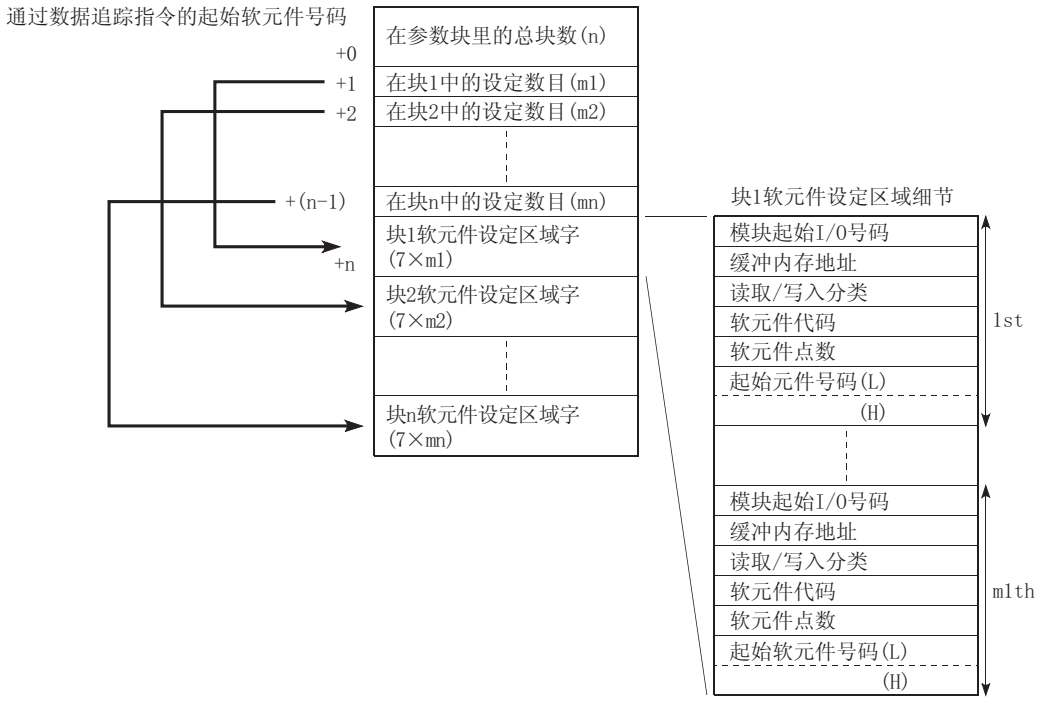
要 点

1. 对于在 MELSECNET(II)、/B、/10 中的远程 I/O 站的特殊的功能模块，缓冲内存批刷新指令不能执行。
2. 对于在 MELSECNET/MINI-S3 中的特殊功能模块，缓冲内存批刷新指令不能执行。

(2) 在该指令执行前设定参数块内容。

参数块由多个块组成。该指令在将要执行的块的特殊继电器 SM1520 到 SM1583 设定后执行。(特殊继电器变为 ON)。

(3) 参数块构成如下：



该块用来设定执行批刷新、特殊功能模块缓冲内存和缓冲内存的特殊功能模块的读 / 写软元件内存类型、点数、起始号码等。
根据传送触点 SM1520 到 SM1583，该块根据变为 ON 的传送触点块的内容执行批刷新。
(SM1520 → 块 1、SM1521 → 块 2,..., SM1583 → 块 64)。

要 点

缓冲内存批刷新指令传送触点 SM1520 到 SM1583 能够同数据传送指令一样，使用同样的传送触点号码。

参数块设定项目内容

(a) 参数块总块数
参数块是一个多个块的集合。这可以设定有多少块被使用。

(b) 块 n 的设定数目
这可以设定每个块软元件的设定区域的软元件的数目。

(c) 块 n 的软元件设定区域

这对于将被批刷新的特殊功能模块的缓冲内存和软元件内存，可以用来实际设定其类型，点数和起始软元件号码。

(1) 模块起始 I/O 号码设定了目标特殊功能模块的起始 I/O 号码。
当数字用 3 位的 16 进制数表示时，设定时取其前两位数。
(当 X/Y100 时，设定到 10H。)

(2) 将缓冲内存地址设定为要刷新的特殊功能模块的缓冲内存的起始地址。
该设定用十进制或十六进制进行。

(3) 读取 / 写入分类设定是否读取或写入缓冲内存。
该设定是 :0: 读取，1: 写入

(4) 软元件代码设定了软元件内存类型。
该设定如下表：

软元件	软元件代码	软元件	软元件代码	软元件	软元件代码	软元件	软元件代码
X	0	B	5	C	10	Z	15
Y	1	F	6	D	11	SB	16
M	2	--	--	W	12	SW	17
L	3	ST	8	R	13	SM	18
--	--	T	9	ZR	14	SD	19

T/C 只是当前值。

(5) 软元件数设定了追踪的软元件数目。
该设定用十进制或十六进制。
位软元件设定为 16 的倍数。

(6) 起始软元件号码设定了软元件内存起始号码。
该设定用十进制或十六进制。
位软元件用 16 的倍数设定 (0、16、32...)

备 注

当设定参数块时，应用如下限制。

- m1 + m2 + ... mn ≦ 256 n ≦ 64。
- 当 m1 到 mn 设定区间是 0 时，块数字设定为 m=0。

当没有块设定时，块数设定为 0，跳越这些块。

运算错误

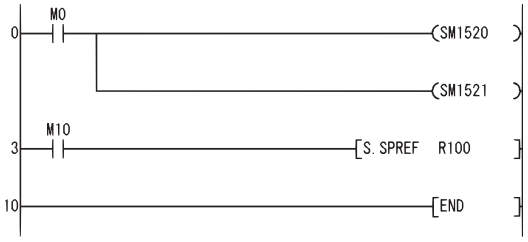
(1) 在下列出现运行错误的情况中，错误标志 (SM0) 变为 ON，并且出错代码存储在 SD0 中。

- 当超过指定范围的值被指定。 (出错代码： 4104)

程序示例

(1) 该程序从存储在 R100 和其后的紧随参数块的块 1 和块 2 刷新缓冲内存。

[梯形图模式]



[列表模式]

步	指令	软元件
0	LD	M0
1	OUT	SM1520
2	OUT	SM1521
3	LD	M10
4	S. SPREF	R100
10	END	

(2) 这是一个参数块设定在 X/Y20 特殊功能模块缓冲内存从 0 到 3、10 到 11 的内容和 X/Y100 特殊功能模块缓冲内存 110 到 119 内容被刷新时的例子。

R100	K2	参数块总块数	
R101	K2	块1设定数目	
R102	K1	块2设定数目	
R103	H2	模块起始I/O号码(X/Y20)	块1软元件设定区域
R104	K0	缓冲内存地址(地址0)	
R105	K0	读取/写入类别(读取)	
R106	K13	软元件代码(R)	
R107	K4	软元件点数(4点)	
R108	K0	起始软元件号码(R0)	
R109	K0		
R110	H2	模块起始I/O号码(X/Y20)	
R111	K10	缓冲内存地址(地址10)	
R112	K1	读取/写入类别(写入)	
R113	K13	软元件代码(R)	
R114	K2	软元件点数(2点)	
R115	K10	起始软元件号码(R10)	
R116	K0		
R117	H10	模块起始I/O号码(X/Y100)	块2软元件设定区域
R118	K110	缓冲内存地址(地址110)	
R119	K0	读取/写入类别(读取)	
R120	K11	软元件代码(D)	
R121	K10	软元件点数(10点)	
R122	K110	起始软元件号码(D110)	
R123	K0		

(a) 块 1 软元件设定区域

- 读取 X/Y20 特殊功能模块缓冲内存 0 到 3 内容到 R0 到 R3。
- 写入 R10 和 11 的内容到 X/Y20 的特殊功能模块缓冲内存 10、11。

(b) 块 2 软元件设定区域

- 读取 X/Y100 特殊功能模块缓冲内存 110 到 119 内容到 D110 到 D119。

备忘录

11 冗余系统指令（用于冗用系统 CPU）

分类	处理详细内容	参考章节
系统切换指令	在执行了 SP.CONTSW 指令的 END 处理时，进行控制系统和备用系统之间的切换。	11.1 节

11.1 系统切换指令 (SP.CONTSW)

通用型 基本型 高性能型 过程 冗余 QnA Q4AR



Ⓢ : 用于确定发出系统切换请求处理的 0 以外的值 (BIN16 位)

Ⓓ : 异常结束软元件号 (位)

设定数据	内部软元件		R、ZR	J□\□		U□\G□	Zn	常数 K、H	其它
	位	字		位	字				
Ⓢ	--	○		--				○	--
Ⓓ	○	○*1		--				--	--

*1: 可用于字软元件的位指定。

功 能

(1) 在执行了 SP.CONTSW 指令的扫描的 END 处理时，在控制系统与待机 (备用) 系统之间进行切换。

(2) 通过 SP.CONTSW 指令进行系统切换时，必须先将 “手动转换有效标志 (SM1592)” 置为 ON(有效)。

(3) 当使用多个 SP.CONTSW 指令时，可通过 Ⓢ 来确定发生了系统切换的程序的处理块。

在 Ⓢ 中，应指定范围在 -32768 到 -1 和 1 到 32767(1H 到 FFFFH) 内的值。

当系统切换正常结束时，通过 SP.CONTSW 指令指定的 Ⓢ 值将被存储到出错公共信息的 “系统切换指令变量 (SD6)” 中。*2

当在同一个扫描中执行了多个 SP.CONTSW 指令时，最先执行的 SP.CONTSW 指令的变量将被存储到系统切换指令变量 (SD6) 中。

(4) 当系统切换正常结束时，通过 SP.CONTSW 指令指定的 Ⓢ 值将被存储到新控制系统 CPU 模块的 “系统切换指令变量 (SD1602)” 中。*3

通过从新控制系统 CPU 模块读取 SD1602 值，系统切换所使用的 SP.CONTSW 指令可以得到确认。

*2 : 通过 SP.CONTSW 指令指定的 Ⓢ 值可以在 GX Developer 的可编程控制器诊断对话框的出错公共信息中得到确认。

*3 : 新控制系统 CPU 模块是指，通过 SP.CONTSW 指令进行系统切换，从待机 (备用) 系统切换为控制系统的 CPU 模块。

(5) 当无法通过 SP.CONTSW 指令进行系统切换时，控制系统 CPU 模块会将异常结束软元件变为 ON。

(a) 在 SP.CONTSW 指令执行时由于下列原因而检测到 OPERATION ERROR 时，异常结束软元件会在指令执行时变为 ON。

- 在执行的 SP.CONTSW 指令的 Ⓢ 中指定了 0 时。
- “手动切换有效标志 (SM1592)”为 OFF 时。
- 在独立模式的待机系统中执行了 SP.CONTSW 指令时。
- 在调试模式中执行了 SP.CONTSW 指令时。

(b) 如果系统因为下表中的原因而无法切换，则在 END 处理的系统切换执行时，异常结束软元件变为 ON。

原因号	系统切换失败的原因
0	正常结束。
1	追踪（热备）电缆脱落或出现故障。
2	在待机系统中发生硬件故障、断电、复位或看门狗时钟错误。
3	在控制系统中发生看门狗时钟错误。
4	追踪（热备）传输的准备中。
5	通讯超时。
6	待机系统中出现停止错误。（除看门狗时钟错误以外）
7	控制系统与待机系统的运行状态不同。
8	正在进行从控制系统向待机系统的内存复制。
9	正在进行 RUN 中写入。
10	待机系统中检测出网络故障。

当因为无法进行系统切换而导致异常结束软元件变为 ON 时，将存储 16 到 “系统切换的原因 (SD1588)”，存储上表中的原因号到 “系统切换失败的原因 (SD1589)”中。

(6) 应使用用户程序或 GX Developer 将状态为 ON 的异常结束位变为 OFF。
如果在异常结束软元件为 ON 的状态下，通过 SP.CONTSW 指令执行正常系统转换操作，新待机系统 CPU 模块的异常结束软元件也会变为 OFF。
但是，由于 SP.CONTSW 指令以外的原因导致发生了系统切换时，异常结束软元件不会变为 OFF。

运算错误

(1) 在下列情况下将发生运算错误， 错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 执行 SP.CONTSW 指令时，在 Ⓢ 中指定了 0。 （出错代码： 4100）
- 执行 SP.CONTSW 指令时，“手动切换有效标志 (SM1592)”为 OFF（无效）状态。 （出错代码： 4120）
- 在独立模式中的待机系统 CPU 模块中执行了 SP.CONTSW 指令。 （出错代码： 4121）
- 在调试模式中执行了 SP.CONTSW 指令。 （出错代码： 4121）

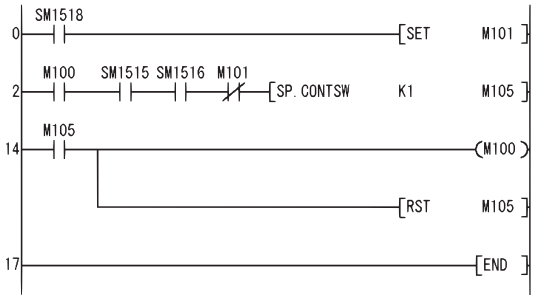
(2) 如果由于以下原因导致无法执行系统切换，错误标志 (SM0) 将变为 ON，出错代码将存储在 SD0 中。

- 追踪电缆脱落或出现故障。 (出错代码：6220)
- 待机系统中发生了硬件故障、断电、复位或看门狗时钟错误。 (出错代码：6220)
- 在控制系统中发生了看门狗时钟错误。 (出错代码：6220)
- 跟踪传输准备中。 (出错代码：6220)
- 通讯超时。 (出错代码：6220)
- 待机系统中发生停止错误。(除看门狗时钟错误以外) (出错代码：6220)
- 控制系统与待机系统的运行状态不同。 (出错代码：6220)
- 正在从控制系统向待机系统进行内存复制。 (出错代码：6220)
- 正在执行 RUN 中写入。 (出错代码：6220)
- 待机系统中检测出网络故障。 (出错代码：6220)

程序示例

(1) 下列程序在系统切换指令 (M100) 的上升沿执行系统切换。
如果系统切换指令 (M100) 保持 ON 状态不变，系统切换后，新控制系统 CPU 模块也会执行 SP.CONTSW 指令，因此，将 M101 附加到执行条件中作为连续切换防止用标志。

[梯形图模式]



[列表模式]

步	指令	软元件		
0	LD	SM1518		
1	SET	M101		
2	LD	M100		
3	AND	SM1515		
4	AND	SM1516		
5	ANI	M101		
6	SP.CONTSW		K1	M105
14	LD	M105		
15	OUT	M100		
16	RST	M105		
17	END			

12 出错代码

12.1 出错代码表

CPU 模块在以下情况下发生异常时，通过自诊断功能显示出错信息（在 LED 上显示、显示器的信息显示），并将出错信息存储到特殊继电器 SM 和特殊寄存器 SD 中。

- 可编程控制器电源上电时。
- CPU 模块从 STOP 切换到 RUN 时。
- CPU 模块运行过程中。

此外，如果从外围设备 (GX Developer)、智能功能模块或者网络系统向 CPU 模块发送通讯请求时发生出错，CPU 模块将向请求源返回出错代码 (4000H 到 4FFFH)。

以下介绍 CPU 模块中发生的出错内容及出错相应处理方法等。

(1) 出错代码表的使用方法

12.1 项出错代码表 (1000 ～ 1999) ～ 12.1.9 项出错代码表 (7000 ～ 10000) 的使用方法如下所示。

(a) 关于出错代码、公共信息、个别信息

出错代码、公共信息、个别信息的各标题栏的括号内的字符表示存储各信息的特殊寄存器编号。

(b) 关于对应 CPU

0	: 对应所有的 QnACPU 和 QCPU。
QCPU	: 对应所有的 Q 系列 CPU 模块。
Q00J/Q00/Q01	: 对应基本型 QCPU。
Qn(H)	: 对应高性能型 QCPU。
QnPH	: 对应过程 CPU。
QnPRH	: 对应冗余 CPU。
QnU	: 对应通用型 QCPU。
QnA	: 对应 QnA 系列和 Q2ASCPU 系列。
Rem	: 对应 MELSECNET/H 远程 I/O 模块。
各 CPU 模块的型号	: 仅对应记载的 CPU 模块。（例如：Q4AR、Q2AS）

12.1.1 出错代码

出错中包含有通过 CPU 模块的自诊断功能检测出的出错，以及与 CPU 模块通信时检测出的出错。

出错的检测类型、出错检测位置以及出错代码的关系如下表所示。

出错检测类型	出错检测位置	出错代码	出错内容的参照目标
通过 CPU 模块的自诊断功能检测出	CPU 模块	1000 到 10000*1*2	12.1.3 项到 12.1.9 项
在和 CPU 模块的通讯过程中检测出	CPU 模块	4000H 到 4FFFH	• Q 系列： QCPU 用户手册（硬件设计 / 维护点检篇） • QnA 系列： 所使用的 CPU 模块的用户手册（详细篇）
	串行通信模块等	7000H 到 7FFFH	串行通信模块用户手册等
	CC-Link 模块	B000H 到 BFFFH	CC-Link 系统主站 / 本地站模块用户手册
	以太网模块	C000H 到 CFFFH	以太网接口模块用户手册
	MELSECNET/G 网络模块	E000H 到 EFFFH	MELSECNET/G 网络参考手册
	MELSECNET/H、MELSECNET/10 网络模块	F000H 到 FFFFH	• Q 系列： Q 系列 MELSECNET/H 网络系统参考手册 • QnA 系列： QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册

1: CPU 模块的出错代码被分为轻度异常、中度异常、严重异常三类。
- 轻度异常：电池出错等 CPU 模块继续运行错误
- 中度异常：WDT 出错等 CPU 模块停止运行错误（出错代码 :1300 ～ 10000）
- 严重异常：RAM 异常等 CPU 模块停止运行错误（出错代码 :1000 ～ 1299）

“继续运行错误”与 “停止运行错误”可以通过 12.1.3 项到 12.1.9 项出错代码表的 “CPU 动作状态”中判别。

2: 在检测出参照目标的出错代码表中未记载的出错代码时，应向附近的 FA 中心咨询。

12.1.2 出错代码的读取方法

如果发生了出错，相应出错代码、出错消息等可以通过外围设备 (GX Developer) 按照如下所示进行读取。

关于外围设备的操作方法，请参阅 GX Developer 操作手册以及 SW □ IVD/NX-GPPQ 操作手册（在线篇）。

12.1.3 出错代码一览表 (1000 ～ 1999)

以下介绍出错代码 1000 ～ 1999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1000	MAIN CPU DOWN	-	-	熄灭	闪烁	停止	常时	
1001								
1002								
1003								
1004								
1005								
1006								
1007								
1008								
1009								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
1000	主 CPU 失控或故障 • 噪声或者其它原因引起的误动作 • 硬件故障	• 采取降低噪声措施。 • 复位 CPU 模块，然后重新运行。如果相同的出错再次显示，这表明是 CPU 模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
1001			Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU
1002			
1003			
1004			
1005			Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU Rem
1006			Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU
1007			Qn(H) QnPH QnPRH
1008			
1009	• 在电源模块、CPU 模块、基板或者扩展基板上检测出错误。 • 使用冗余类型基板时，检测出两个冗余电源模块都发生故障。此外，检测出冗余基板故障。	复位 CPU 模块，然后重新运行。如果再次检测出相同的错误，则可能是电源模块、CPU 模块、主基板、扩展基板或者扩展基板有故障。 （请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Q00J/Q00/Q01*4 Qn(H)*6 QnPH QnPRH QnU

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1010	END NOT EXECUTE	－	－	熄灭	闪烁	停止	当执行 END 指令时	
1020	SFCP. END ERROR	－	－	熄灭	闪烁	停止	当执行 SFC 程序时	
1035	MAIN CPU DOWN	－	－	熄灭	闪烁	停止	常时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1010	整个程序在 END 指令未被执行的情况下运行完毕。 • 当 END 指令执行时，由于噪声等原因，它被读取为其它指令。 • END 指令由于某种原因被更改为其它指令。	• 采取降低噪声措施。 • 复位 CPU 模块，然后重新运行。如果相同的出错再次显示，则表明是 CPU 模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1020	由于噪声或者其他原因，SFC 程序未能正常结束。 • 由于噪声或者相似原因，SFC 程序未能正常结束。 • 由于某种原因，SFC 程序未能正常结束。		Q00J/Q00/Q01*4 QnPH QnU
	1035	检测出 CPU 失控或故障 • 噪声或者其它原因引起的误动作 • 硬件故障	• 采取降低噪声措施。 • 复位 CPU 模块，然后重新运行。如果相同的出错再次显示，这表明是 CPU 模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnU

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为“04101”以后的 CPU 为对象。
*7: 以序列号的高 5 位为“07032”以后的 CPU 为对象。
*8: 以序列号的高 5 位为“08032”以后的 CPU 为对象。
*9: 以序列号的高 5 位为“09012”以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1101	RAM ERROR	－	－	熄灭	闪烁	停止	电源接通 / 复位时 / 当执行 END 指令时	
1102							电源接通 / 复位时 / 当执行 END 指令时	
1103							电源接通 / 复位时	
							电源接通 / 复位时 / 当执行 END 指令时	
1104	RAM ERROR	－	－	熄灭	闪烁	停止	电源接通 / 复位时	
1105								
1106	RAM ERROR	－	－	熄灭	闪烁	停止	STOP → RUN/ 当执行 END 指令时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1101	存储顺控程序的 CPU 模块中的内置 RAM/程序内存故障。	• 采取减少噪声措施。 • 复位 CPU 模块，并重新运行。如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1102	CPU 模块中的工作区 RAM 有故障。	• 采取减少噪声措施。 • 复位 CPU 模块，并重新运行。如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1103	CPU 模块中的软元件内存有故障。	• 采取减少噪声措施。 • 变址修饰时，检查变址寄存器的值是否在软元件范围内。 • 复位 CPU 模块，并重新运行。如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
		• CPU 模块中的软元件内存有故障。 • 由于变址修饰访问了超出范围的软元件，系统软元件被改写。		Qn(H)*8 QnPH*8 QnPRH*9
	1104	CPU 模块中的地址 RAM 有故障。	• 采取减少噪声措施。 • 复位 CPU 模块，并重新运行。如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1105	CPU 模块中的系统 RAM 有故障。		Q4AR
		CPU 模块中的 CPU 共享内存有故障。		Q00J/Q00/Q01 QnU
		CPU 模块中的 CPU 内存有故障。		Qn(H)*4 QnPH QnPRH QnU
	1106	电池已耗尽。 CPU 模块中的程序内存有故障。	• 检查电池，看其是否已经耗尽。如果电池已耗尽，更换电池。 • 采取减少噪声措施。 • 格式化程序内存，将所有文件写到可编程控制器，然后复位 CPU 模块，并重新运行。 如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H) QnPH*7 QnPRH

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1107	RAM ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
1108								
1109							常时	
1110	TRK. CIR. ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
1111		-	-	熄灭	闪烁	停止		
1112	TRK. CIR. ERROR	-	-	熄灭	闪烁	停止	运行过程中	
1113								
1115	TRK. CIR. ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
1116	TRK. CIR. ERROR	-	-	熄灭	闪烁	停止	运行过程中	
1150	RAM ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1107	CPU 模块中的工作区 RAM 有故障。	CPU 模块硬件故障。 （请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnPRH
	1108			
	1109			Qn(H)*8 QnPH*8 QnPRH*9
	1110	在对热备硬件的初始化检查中检测到故障。		QnPRH
	1111	检测到热备硬件故障。		
	1112	• 在运行过程中检测到热备硬件故障。 • 在未将待机系统断电或者复位的状态下对热备电缆进行了插拔。 • 没有用连接器固定螺钉固定热备电缆。 • 由于未按冗余系统启动步骤进行操作，在启动时出错。	• 在检查热备电缆已经连接后，启动。如果相同的出错再次显示，则原因是热备电缆或者 CPU 模块的硬件故障。（ 联系您当地的三菱代理商，说明故障症状。） • 确认冗余系统启动顺序，并重新执行启动操作。对于详细信息，参考 QnPRHCPU 用户手册（ 冗余系统）。	QnPRH
	1113			
	1115	热备硬件的初始化检查检测到故障。	CPU 模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	
	1116	• 在运行过程中检测到热备硬件故障。 • 在未将待机系统断电或者复位的状态下对热备电缆进行了插拔。 • 没有用连接器固定螺钉固定热备电缆。 • 由于未按冗余系统启动步骤进行操作，在启动时出错。	• 在检查热备电缆已经连接后，启动。如果相同的出错再次显示，则原因是热备电缆或者 CPU 模块的硬件故障。（ 联系您当地的三菱代理商，说明故障症状。） • 确认冗余系统启动顺序，并重新执行启动操作。对于详细信息，参考 QnPRHCPU 用户手册（ 冗余系统）。	
	1150	多 CPU 高速通信区域的 CPU 内存检测到故障。	• 采取减少噪声措施。 • 复位 CPU 模块，并重新运行。 如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnU*10

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1160	RAM ERROR	－	－	熄灭	闪烁	停止	程序执行时	
1161							程序执行时	
1162							电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1160	CPU 模块的程序内存被改写。	• 采取减少噪声措施。 • 格式化程序内存，将所有文件写到可编程控制器，然后复位 CPU 模块，并重新运行。 如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnU
	1161	CPU 模块内的软元件内存数据被改写。	• 采取减少噪声措施。 如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	
	1162	CPU 模块的电池保持的数据检测到故障。（设置不进行自动格式化时发生）	• 采取减少噪声措施。 • 更换 CPU 或 SRAM 卡的电池。 如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1200	OPE. CIRCUIT ERR.	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
1201								
1202								
1203	OPE. CIRCUIT ERR.	-	-	熄灭	闪烁	停止	当执行 END 指令时	
1204								
1205								
1206	OPE. CIRCUIT ERR.	-	-	熄灭	闪烁	停止	当指令执行时	
1300	FUSE BREAK OFF	模块号（插槽号）［对于远程 I/O 网络］网络号 / 站号	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	常时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)

*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。

*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1200	CPU 模块内用于变址修饰的运算电路运行不正常。	CPU 模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1201	CPU 模块内的硬件（逻辑）运行不正常。		
	1202	CPU 模块内用于顺控处理的运算电路运行不正常。		
	1203	CPU 模块中用于变址修饰的运算电路运行不正常。		Q4AR QnPRH
	1204	CPU 模块内的硬件（逻辑）运行不正常。		
	1205	CPU 模块内用于顺控处理的运算电路运行不正常。		
	1206	CPU 模块内的 DSP 运算电路运行不正常。		Q4AR
	1300	有保险丝熔断的输出模块。	•检查输出模块的 FUSE .LED，更换 LED 点亮的模块。 （保险丝熔断的模块也可以使用 GX Developer 来确认。检查特殊寄存器 SD1300 到 SD1331，如果和模块相对应位是 “1”，则模块的保险丝熔断。） • 当 GOT 是总线连接到主基板或者扩展基板时，检查扩展电缆的连接状态和 GOT 的接地状态。	Qn(H) QnPH QnPRH QnU Rem
		有保险丝熔断的输出模块。	检查输出模块的 ERR. LED，并更换 LED 点亮的模块。 （保险丝熔断的模块也可以使用 GX Developer 来确认。检查特殊寄存器 SD1300 到 SD1331，如果和模块相对应位是 “1”，则模块的保险丝熔断。）	Q00J/Q00/Q01

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1300	FUSE BREAK OFF	模块号（插槽号）[对于远程 I/O 网络]号 / 站号	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	常时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1300	有保险丝熔断的输出模块。	• 检查输出模块的 ERR. LED，并更换 LED 点亮的模块的保险丝。 • 使用外围设备读取出错的公共信息，并更换对应读取的数值（模块号）的输出模块上的保险丝。 也可以用外围设备监视特殊寄存器 SD1300 到 SD1331，更换其位的值为“1”的输出模块的保险丝。 • GOT 是总线连接到主基板或者扩展基板时，检查扩展电缆的连接状态和 GOT 的接地状态。	QnA Q4AR
		• 有保险丝熔断的输出模块。 • 输出负载用的外部电源关断或者未连接。	• 检查输出模块的 ERR.LED，并更换 LED 点亮的模块。 • 使用外围设备读取出错的公共信息，并更换对应读取的数值（模块号）的输出模块上的保险丝。 也可以用外围设备监视特殊寄存器 SD1300 到 SD1331，更换其位的值为“1”的输出模块的保险丝。 • 检查用于输出负载的外部电源是 ON 还是 OFF。 • GOT 是总线连接到主基板或者扩展基板时，检查扩展电缆的连接状态和 GOT 的接地状态。	Q2AS

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为“04101”以后的 CPU 为对象。
*7: 以序列号的高 5 位为“07032”以后的 CPU 为对象。
*8: 以序列号的高 5 位为“08032”以后的 CPU 为对象。
*9: 以序列号的高 5 位为“09012”以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1310	I/O INT. ERROR	－	－	熄灭	闪烁	停止	发生中断时	
1311								
1401	SP. UNIT DOWN	模块号（插槽号）	－	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	电源接通 / 复位 / 当智能功能模块被访问时	
				熄灭	闪烁	停止 *2	电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1310	虽然没有中断模块存在，但是有中断发生。	安装的模块中有硬件故障。因此，检查安装的模块和更换有故障的模块。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	1311	检测到来自中断模块以外的模块发出的中断请求。	采取措施，防止从中断模块以外的模块发出中断。	Q00J/Q00/Q01*4 QnU
		从在可编程控制器参数中未进行中断指针设置的模块上检测到中断请求。	• 在可编程控制器参数对话框的可编程控制器系统设置中修改中断指针设置。 • 采取措施，防止未在可编程控制器参数对话框的可编程控制器系统设置中进行中断指针设置的模块发送中断请求。修改网络参数的中断设置。修改智能功能模块缓冲存储区的中断设置。修改 QD51 的基本程序。	Q00J/Q00/Q01*5 QnPRH QnU
	1401	• 在进行初始化处理时，智能功能模块 / 特殊功能模块没有响应。 • 智能功能模块 / 特殊功能模块的缓冲存储区的大小有错误。 • 安装了不支持的模块。	安装了不支持的模块时，卸下不支持的模块。 安装了支持的对应模块时，可能是智能功能模块 / 特殊功能模块、CPU 模块或者基板的硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU Rem
		在进行了可编程控制器参数的 I/O 分配设置的情况下，在进行初始化处理时，特殊功能模块没有返回信号。（当发生错误时，在公共信息中存储相应特殊功能模块的起始 I/O 号码。）	CPU 模块、基板或者被访问的特殊功能模块有硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnA

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1402	SP. UNIT DOWN	模块号（插槽号）	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	当执行智能功能模块访问指令时	
				熄灭	闪烁	停止	在 FROM/TO 指令集的执行过程中	
1403	SP. UNIT DOWN	模块号（插槽号）	－	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	当执行 END 指令时	
				熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	常时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1402	通过程序访问了智能功能模块 / 特殊功能模块，但是没有任何响应。	智能功能模块 / 特殊功能模块、CPU 模块或者基板的硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU Rem
		在执行 FROM/TO 指令的过程中，访问了特殊功能模块，但是没有响应。 （当发生错误时，在个别信息中存储相应程序出错位置。）	CPU 模块、基板或者被访问的特殊功能模块有硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnA
	1403	• 安装了不支持的模块。	安装了不支持的模块时，卸下不支持的模块。 安装了支持的对应模块时，可能是智能功能模块 / 特殊功能模块、CPU 模块或者基板的硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU
		• 当执行 END 指令时，智能功能模块 / 特殊功能模块没有响应。 • 在智能功能模块 / 特殊功能模块上检测到出错。 • 在运行过程中，I/O 模块（智能功能模块 / 特殊功能模块）被松动、完全卸载或者被安装。	CPU 模块、基板或者被访问的智能功能模块 / 特殊功能模块有硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU Rem

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1411	CONTROL-BUS. ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时	
1412	CONTROL-BUS. ERR.	模块号（插槽号）	程序出错位置	熄灭	闪烁	停止	在执行 ROM/TO 指令的过程中。	
1413	CONTROL-BUS. ERR.	-	-	熄灭	闪烁	停止	常时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1411	已经进行了参数的 I/O 分配，但是在初始化通讯过程中，不能访问智能功能模块 / 特殊功能模块。（在出错发生时，相应的智能功能模块 / 特殊功能模块的起始 I/O 号被存储在公共信息中。）	复位 CPU 模块，然后重新运行。如果相同的出错再次显示，智能功能模块 / 特殊功能模块、CPU 模块或者基板有故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○ Rem
	1412	由于带智能功能模块 / 特殊功能模块的系统总线出错，FROM/TO 指令不能执行。 （在出错发生时，程序出错位置被存储在个别信息中。）		○
	1413	使用不兼容多 CPU 系统 CPU 模块构建多 CPU 系统。	• 从主基板上卸载和多 CPU 系统不兼容的 CPU 模块，或者将和多 CPU 系统不兼容的 CPU 模块更换为和多 CPU 系统兼容的 CPU 模块。 • 智能功能模块、CPU 模块或者基板有故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Q00J/Q00/Q01*4 Qn(H)*4 QnPH
		在系统总线上检测到出错。 • 系统总线自检测出错。 • CPU 模块自检测出错。	复位 CPU 模块，然后重新运行。如果相同的出错再次显示，则可能是智能功能模块、CPU 模块或者基板有故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU Rem

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
1500	AC/DC DOWN	－	－	点亮	熄灭	继续运行	常时	
1510	DUAL DC DOWN 5V	－	－	点亮	点亮	继续运行	常时	
	SINGLE PS. DOWN	基板号 / 电源号	－	点亮	点亮	继续运行	常时	
1520	DC DOWN 5V	－	－	熄灭	闪烁	停止	常时	
	SINGLE PS. ERROR	基板号 / 电源号	－	点亮	点亮	继续运行	常时	
1530	DC DOWN 24V	－	－	点亮	点亮	继续运行	常时	
1600	BATTERY ERROR*3	驱动器名称	－	点亮	熄灭	继续运行	常时	
1601								
1602					点亮			
1610	FLASH ROM ERROR	－	－	点亮	点亮	继续运行	当写入 ROM 时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 对于除远程 I/O 模块以外的 QCPU，可以通过参数选择各智能功能模块的出错停止 / 继续运行。
*3: 发生了 BATTERY ERROR 时，BAT. ALM LED 将亮灯。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	1500	• 电源发生瞬时掉电。 • 电源关闭。	检查电源。	○ Rem
	1510	电源冗余扩展基板上的两个电源模块中的一个的电源电压 (100-240V) 下降到额定电压的 85 % 以下。(只有冗余系统的控制系统才可以检测。)	检查电源模块的供电电压。如果电压无异常，则更换电源模块。	Q4AR
		冗余基板上的电源模块中的一个的电源电压下降。	检查安装在冗余基板上的冗余电源模块的供电电压。	Qn(H)*6 QnPH*6 QnPRH Rem
	1520	扩展基板的电源模块的电源电压 (100-240V) 下降到额定电压的 85 % 以下。 (单独系统或冗余系统的控制系统才可以检测。)	检查电源模块供电电压。如果电压无异常，更换电源模块。	Q4AR
		在冗余电源系统的某个冗余电源模块中检测到故障。	冗余电源模块的硬件故障。 (联系附近的三菱 FA 中心或代理商，说明故障症状。)	Qn(H)*6 QnPH*6 QnPRH Rem
	1530	给系统管理模块 AS92R 供电的 24VDC 电压下降到额定电压的 90% 以下。(冗余系统的控制系统或待机系统才可以检测。)	检查给系统管理模块 AS92R 供电的 24VDC 电源。	Q4AR
	1600	• CPU 模块电池的电压下降到低于规定的水平。 • CPU 模块电池没有安装导线连接器。	• 更换电池。 • 如果电池是用于程序内存、标准 RAM 或者用于备份电源，安装导线连接器。	○
	1601	存储卡 1 上的电池的电压下降到低于规定水平。	更换电池。	QnA Qn(H) QnPH QnPRH QnU
	1602	存储卡 2 上的电池的电压下降到低于规定水平。	更换电池。	QnA
	1610	写入闪存 ROM(标准 ROM 和系统预留区域) 次数超过 10 万次。 (写入次数 >10 万次)	更换 CPU 模块。	QnU

*4: 以功能版本 B 以后的 CPU 为对象。
*5: 以功能版本 A 的 CPU 为对象。
*6: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*7: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*8: 以序列号的高 5 位为 “08032” 以后的 CPU 为对象。
*9: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*10: 除 Q02UCPU 外的通用型 QCPU。

12.1.4 出错代码表 (2000 ～ 2999)

以下介绍出错代码 2000 ～ 2999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2000	UNIT VERIFY ERR.	模块号（插槽号）[对于远程 I/O 网络]网络号 / 站号	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行 END 指令时	
2001	UNIT VERIFY ERR.	模块号（插槽号）	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	当执行 END 指令时	
2010	BASE LAY ERROR	基板号	-	熄灭	闪烁	停止	电源接通 / 复位时	
2011								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2000	在多 CPU 系统中安装了多 CPU 系统不兼容的 CPU 模块。	将多 CPU 系统不兼容的 CPU 模块更换为多 CPU 系统兼容的 CPU 模块。	Qn(H)*3 QnPH
		I/O 模块实际安装情况与电源 ON 时的 I/O 模块信息不符。 • 在运行过程中 I/O 模块（包括智能功能模块）松脱或者被插拔。	通过 GX Developer 读取出错公共信息，检查或者更换对应数值（模块号）的模块。 也可以使用 GX Developer 监视特殊寄存器 SD150 到 SD157，检查或更换其数据位变为 “1” 的模块。	Q00J/Q00/Q01
		I/O 模块实际安装情况在电源 ON 时与参数设置不符。 • 在运行过程中 I/O 模块（或者智能功能模块 / 特殊功能模块）松脱或者被插拔。	• 使用外围设备读取出错的公共信息，检查或者更换对应数值（模块号）的模块。 • 也可以在外围设备上监视特殊寄存器 SD1400 到 SD1431，更换其位数据是 “1” 的模块。 • 当 GOT 是总线连接到主基板或者扩展基板时，检查扩展电缆的连接状态和 GOT 的接地状态。	QnA Qn(H) QnPH QnPRH QnU Rem
	2001	运行过程中，模块被安装到设定为 “空余” 的插槽中。	运行过程中，不要安装模块到设定了 CPU 模块 “空余” 设定的插槽中。	Q00J/Q00/Q01*3 QnU
	2010	• 使用的扩展基板超过了可用扩展基板数。 • 当 GOT 是总线连接时，GOT 的电源变为 OFF 时 CPU 模块被复位	• 使用容许数量内的扩展基板。 • 重新给可编程控制器和 GOT 上电。	Q00J/Q00/Q01*3 QnPRH Q02U
	2011	QA1S6 □B 或者 QA6 □B 被用作基板。	不要将 QA1S6 □B 或者 QA6 □B 用作基板。	Q00J/Q00/Q01*3 QnPRH QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2012	BASE LAY ERROR	基板号	-	熄灭	闪烁	停止	电源接通 / 复位时	
2013								
2020	EXT.CABLE ERR.	-	-	熄灭	闪烁	停止	电源接通 / 复位时 / 当执行 END 指令时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2012	冗余系统中，当 GOT 是总线连接到主基板时。 兼容扩展基板的冗余系统 CPU 检测出如下故障。 • 第 1 级扩展基板连接了非 Q6 □WRB 的其它基板。 • 第 1 级扩展基板未连接 Q6 □WRB 时，第 2 到 7 级扩展基板连接了基板。 • 其它系统的 CPU 与扩展基板不兼容。 • 连接了 Q5 □B、QA1S6 □B 或者 QA6 □B。 • 两系统主基板的插槽数不一致。 • 不能正确读出 Q6 □WRB 信息。	• 卸载 GOT 连接到主基板的总线连接电缆。 • 使用 Q6 □WRB（固定为第 1 级扩展基板） • 在其它系统中使用与扩展基板兼容的 CPU。 • 不要将 Q5 □B、QA1S6 □B 或者 QA6 □B 用作基板。 • 使用同样插槽数的主基板。 • Q6 □WRB 的硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnPRH*6
	2013	冗余系统中，检测出 Q6 □WRB 的基板号为第 1 级以外。	• Q6 □WRB 的硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	
	2020	冗余系统中检测出如下故障。 • 电源接通 / 复位时，待机系统中检测出控制系统与 Q6 □WRB 的连线故障。 • END 处理时，检测出待机系统与 Q6 □WRB 的连线故障。	察看 Q6 □WRB 与主基板之间是否正确连接了扩展电缆，如果未正确连接，应关闭主基板电源连接扩展电缆。 如果电缆连接正确，可能是 CPU 模块、Q6 □WRB 或者扩展电缆故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2100	SP. UNIT LAY ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2100	安装了 QI60 的插槽在可编程控制器参数 I/O 分配设置中被设定为智能（智能功能模块）或中断（中断模块）以外的其它模块。	重新进行设定，使可编程控制器参数 I/O 分配和实际安装状态相匹配。	Qn(H)*3 QnPH QnPRH
		• 在可编程控制器参数 I/O 分配设置中，智能功能模块被分配到了为 I/O 模块预留的位置；或者进行了相反的设置。 • 在可编程控制器参数 I/O 分配设置中，在 CPU 模块的位置上分配了除 CPU 模块外的其它模块，或将其设置为空余。或者进行了相反的设置。 • 在可编程控制器参数 I/O 分配设置中，对无开关设置的模块进行了开关设置。 • 在可编程控制器参数 I/O 分配设置中，将智能功能模块的分配点数设置为小于安装模块点数的值。	• 重新进行可编程控制器参数 I/O 分配设定，以符合智能功能模块和 CPU 模块的实际安装状态。 • 删除可编程控制器参数的 I/O 分配设置中的开关设置。	Qn(H) QnPH QnPRH QnU Rem
		• 在可编程控制器参数 I/O 分配设置中，智能功能模块被分配到了为 I/O 模块预留的位置；或者进行了相反的设置。 • 在可编程控制器参数 I/O 分配设置中，在 CPU 模块的位置上分配了除 CPU 模块外的其它模块，或将其设置为空余。或者进行了相反的设置。 • 在可编程控制器参数 I/O 分配设置中，将智能功能模块的分配点数设置为小于安装模块点数的值。	重新进行可编程控制器参数 I/O 分配设定，以符合智能功能模块和 CPU 模块的实际安装状态。	Q00J/Q00/Q01
		在可编程控制器参数 I/O 分配设置中，特殊功能模块被分配到为 I/O 模块预留的位置；或者相反。	重新进行可编程控制器参数 I/O 分配设定，以符合特殊功能模块的实际安装状态。	QnA

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2101	SP.UNIT LAY ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时	
2102								
2103								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
2101	安装了 13 个以上可以对 CPU 模块进行中断启动的 A 系列特殊功能模块 (A1SI61 除外)。	应安装 12 个以下可以对 CPU 模块进行中断启动的 A 系列特殊功能模块 (A1SI61 除外)。	Qn(H)
	安装了 13 个以上可以对 CPU 模块进行中断启动的特殊功能模块 (A(1S)I61 除外)。	应安装 12 个以下可以对 CPU 模块进行中断启动的特殊功能模块 (A(1S)I61 除外)。	QnA
2102	安装了 7 个以上的 A1SD51S。	应安装 6 个以下的 A1SD51S。	Qn(H)
	安装了 7 个以上的串行通讯模块等 (A(1S)J71QC24 除外)。	应安装 6 个以下的串行通讯模块等 (A(1S)J71QC24 除外)。	QnA
2103	• 在单 CPU 系统中安装了 2 个以上的 QI60/A1SI61 模块。 • 在多 CPU 系统中，2 个以上的 QI60/A1SI61 模块被设定到同一个控制 CPU 中。 • 在多 CPU 系统中，安装了 2 个以上的 A1SI61 模块。	• 将单 CPU 系统中安装的 QI60/A1SI61 模块数减少到一个。 • 在多 CPU 系统中，将设定到相同控制 CPU 的 QI60/A1SI61 模块数改为只有一个。 • 将多 CPU 系统中 A1SI61 模块数减少到只有一个。当多 CPU 系统中各个 QCPU 分别使用中断模块时，将其改为 QI60。((使用一个 A1SI61 模块 + 最多三个 QI60 模块) 或者只使用 QI60 模块。)	Qn(H)*3 QnPH
	安装了 2 个或者更多 QI60、A1SI61 中断模块。	只安装 1 个 QI60、A(1S)I61 模块。	Qn(H) QnPRH
	安装了 QI60。	卸载 QI60。	Rem
	安装了 2 个或者更多 A1SI61 中断模块。	只安装 1 个 AI61 模块。	QnA
	安装了 2 个或者更多 QI60 模块。	将 QI61 模块数减少到一个。	Q00J/Q00/Q01*5
	安装了 2 个或者更多没有进行中断指针设置的 QI60 模块。	• 将 QI60 模块数减少到一个。 • 为第二个 QI60 模块和其后的模块设定中断指针。	Q00J/Q00/Q01*3 QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2104	SP.UNIT LAY ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时	
2105								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2104	在 MELSECNET/MINI 自动刷新网络参数设置中，设定的模块分配和实际链接站号的模块型号不同。	重新设定网络参数 MELSECNET/MINI 自动刷新设置的模块分配，使它符合实际链接站号的模块。	QnA
	2105	分配了太多可以使用专用指令的特殊功能模块（ 安装的模块数）。 （如下所示，总数是大于 1344。） (AD59 安装个数 × 5) (AD57(S1)/AD58 安装个数 × 8) (AJ71C24(S3/S6/S8) 安装个数 × 10) (AJ71UC24 安装个数 × 10) (AJ71C21(S1) 安装个数 × 29) (AJ71PT32-S3/AJ71T32-S3 安装个数 ×125) * (AJ71QC24(R2,R4) 安装个数 × 29) (AJ71ID1(2)-R4 安装个数 × 8) +(AD75 安装个数 × 12) 合计 >1344 *: 当使用扩展模式时。	减少安装的特殊功能模块数。	

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2106	SP.UNIT LAY ERR.	模块号	-	熄灭	闪烁	停止	电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2106	• 在整个系统中安装了总数 3 个或者更多 MELSECNET/H 和 MELSECNET/G 模块。 • 在整个系统安装了 3 个或者更多 Q 系列以太网接口模块。	• 将整个系统中安装的 MELSECNET/H 和 MELSECNET/G 模块数减少到总数 2 个或者更少。 • 将整个系统中安装的以太网接口模块数减少到 2 个或者更少。	Q02U
		• 在整个系统中安装了总数 5 个或者更多 MELSECNET/H 和 MELSECNET/G 模块。 • 在整个系统中安装了 5 个或者更多 Q 系列以太网接口模块。	• 将整个系统中安装的 MELSECNET/H 和 MELSECNET/G 模块数减少到总数 4 个或者更少。 • 将整个系统中安装的以太网接口模块数减少到 4 个或者更少。	QnU*7
		• 在整个系统中安装了 3 个或者更多 MELSECNET/G 模块。 • 在整个系统中安装了总数 5 个或者更多 MELSECNET/H 和 MELSECNET/G 模块。	• 将整个系统中安装的 MELSECNET/G 模块数减少到 2 个或者更少。 • 将整个系统中安装的 MELSECNET/H 和 MELSECNET/G 模块数减少到总数 4 个或者更少。	Qn(H)*6
		• 安装了 5 个或者更多 MELSECNET/H 模块。 • 安装了 5 个或者更多 Q 系列以太网接口模块。	• 将安装的 MELSECNET/H 模块数减少到 4 个或者更少。 • 将安装的 Q 系列以太网接口模块数减少到 4 个或者更少。	Qn(H) QnPH QnPRH Rem
		• 安装了 2 个或者更多 MELSECNET/H 模块。 • 安装了 2 个或者更多 Q 系列以太网接口模块。 • 安装了 3 个或者更多 Q 系列 CC-Link 模块。	• 将安装的 MELSECNET/H 模块数减少到 1 个或者更少。 • 将安装的 Q 系列以太网接口模块数减少到 1 个或者更少。 • 将安装的 Q 系列 CC-Link 模块数减少到 2 个或者更少。	Q00J/Q00/Q01
		• 在 MELSECNET/H 网络系统中，存在相同的网络号和站号。	• 检查网络号和站号。	QCPU Rem
		• 安装了 5 个或者更多 AJ71QLP21 和 AJ71QBR11 模块。 • 安装了 3 个或者更多 AJ71AP21/R21 和 AJ71AT21B 模块。 • 安装的 AJ71QLP21、AJ71QBR11、AJ71AP21/R21 和 AJ71AT21B 模块的总数超过了 5。 • 在 MELSECNET/10 网络系统中，存在相同的网络号和站号。 • 在 MELSECNET(II) 或者 MELSECNET/B 数据链接系统中，同时存在 2 个以上的主站或者本地站。	• 将 AJ71QLP21 和 AJ71QBR11 模块数减少到 4 个或者更少。 • 将 AJ71AP21/R21 和 AJ71AT21B 模块数减少到 2 个或者更少。 • 将 AJ71QLP21、AJ71QBR11、AJ71AP21/R21 和 AJ71AT21B 模块的总数减少到 4 个或者更少。 • 检查网络号和站号。 • 检查站号。	QnA

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2107	SP.UNIT LAY ERR.	模块号（插槽号）	–	熄灭	闪烁	停止	电源接通 / 复位时	
2108								
2109						停止 / 继续运行 *2		
2110	SP.UNIT ERROR	模块号（插槽号）	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当指令执行时	
2111								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2107	在可编程控制器参数 I/O 分配设置中起始 X/Y 和其他模块的起始 X/Y 重复。	重新设定可编程控制器参数 I/O 分配设置，使其符合智能功能模块 / 特殊功能模块的实际状态。	○ Rem
	2108	• 安装了 A2USCPU 专用的网络模块 A1SJ71LP21、A1SJ71BR11、A1SJ71AP21、A1SJ71AR21 或者 A1SJ71AT21B。 • 安装了 Q2AS 专用的网络模块 A1SJ71QLP21 或者 A1SJ71QBR11。	将网络模块更换为 MELSECNET/H 模块。	Qn(H)
		安装了用于和 AnUCPU 网络模块一起使用的 A(1S)J71LP21 或者 A(1S)J71BR11。	将网络模块更换为 A(1S)J71QLP21 或者 A(1S)J71QBR11。	QnA
	2109	当冗余系统处于热备模式时，控制系统和待机系统的模块配置不同。	检查待机系统的模块配置。	Q4AR
	2110	• FROM/TO 指令指定的位置不是智能功能模块 / 特殊功能模块。 • FROM/TO 指令指定了没有包含缓冲存储区的模块。 • 被访问的智能功能模块 / 特殊功能模块、网络模块有故障。 • 在以 CPU 共享存储区为对象的指令中指定了没有安装的站。	• 使用 GX Developer 读取出错的个别信息，检查和数值（程序出错位置）相对应的 FROM/TO 指令，必要时进行修改。 • 被访问的智能功能模块 / 特殊功能模块有硬件故障。更换有故障的模块。或者请联系附近的三菱 FA 中心或代理商，说明故障症状。	Q00J/Q00/Q01 Qn(H)*3 QnPH QnPRH QnU
	2111	• 由链接直接软元件（J □ \ □）指定的位置不是网络模块。 • 在运行过程中，I/O 模块（智能功能模块 / 特殊功能模块）被卸载、或者被安装。		○

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2112	SP.UNIT ERROR	模块号（插槽号）	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当指令执行时 /STOP → RUN	
2113	SP.UNIT ERROR	FFFF$_H$（固定）	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	当指令执行时 /STOP → RUN	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2112	• 特殊功能模块以外模块被特殊功能模块专用指令指定。或者，它不是相应的特殊功能模块。 • 特殊功能模块专用指令指定的模块型号和参数 I/O 分配指定的模块型号不同。	• 使用外围设备读取出错的个别信息，检查和数值（程序出错位置）相对应的特殊功能模块专用指令（网络指令），以进行修改。 • 依照特殊功能模块专用指令设定，通过可编程控制器参数 I/O 分配设定模块型号。 实例）对 AJ71QC24N，设定为 AJ71QC24。	QnA
		• 智能功能模块 / 特殊功能模块以外的模块被智能功能模块 / 特殊功能模块专用指令指定。 或者，它不是相应的智能功能模块 / 特殊功能模块。 • 被网络专用指令指定的网络号不存在。 或者中转目标网络不存在。	使用外围设备读取出错的个别信息，检查和数值（程序出错位置）相对应的智能功能模块 / 特殊功能模块专用指令（网络指令），以进行修改。	QCPU Rem
	2113	网络模块以外的模块被网络专用指令指定。		QnA Qn(H) QnPH

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2114	SP. UNIT ERROR	模块号（插槽号）	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行	当指令执行时 /STOP → RUN	
2115								
2116								
2117								
2118							当指令执行时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
2114	将用于其他站的指令用于了自站 CPU。（不能用于自站 CPU 的指令。）	使用 GX Developer 出错的读取个别信息，检查对应数值（程序出错位置）的程序，并进行改正。	Q00J/Q00/Q01*3 Qn(H)*3 QnPH QnU
2115	将用于指定自站的指令用于了其他站 CPU。（不能用于其他站 CPU 的指令。）		Q00J/Q00/Q01*3 Qn(H)*3 QnPH
2116	• 将不能用于其他 CPU 管理的模块的指令用于其他 CPU 管理的模块。 • 对其他 CPU 管理的 A/QnA 模块执行了指令。		Q00J/Q00/Q01*3 Qn(H)*3 QnPH QnU
2117	指定了不能在多 CPU 系统专用指令中被指定的 CPU 模块。		Q00J/Q00/Q01*3 Qn(H)*3 QnPH QnU
2118	在多 CPU 系统的可编程控制器参数中，将在线模块更换设置设定为允许时，在 FROM 指令 / 智能功能模块软元件 (U □ \G □）中指定了其他 CPU 控制的智能功能模块。	• 当在多 CPU 系统中执行在线模块更换时，改正程序使程序不访问其他 CPU 控制的智能功能模块。 • 当在多 CPU 系统中访问其他 CPU 控制的智能功能模块时，在参数中将在线模块更换设置设定为禁止。	Qn(H)*3 QnPH QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2120	SP. UNIT LAY ERR.	－	－	熄灭	闪烁	停止	电源接通 / 复位时	
2121								
2122								
2124								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
2120	Q5 □ B、Q6 □ B 及 QA1S6 □ B、QA6 □ B 的配置不正确	重新进行基板配置。	Q00J/Q00/Q01*4 Qn(H) QnPH
2121	CPU 模块被安装在除 CPU 插槽或 0 ～ 2 号插槽以外的插槽中。	检查 CPU 模块的安装位置并重新安装到正确的插槽。	Qn(H) QnPH Rem
2122	将 QA1S6 □ B、QA6 □ B 作为主基板使用。	采用适用的基板作为主基板。	Qn(H) QnPH QnPRH Rem
2124	• 模块安装在第 65 个或者更后的插槽。 • 模块安装在基板分配设置指定的插槽之后的插槽。 • 模块安装在第 4,096I/O 点之后。 • 模块占用 I/O 点超过了第 4,096 点 I/O 点。	• 卸载安装在第 65 个或者更高的插槽的模块。 • 卸载安装在基板分配设置指定的插槽之后的插槽中的模块。 • 卸载安装在第 4,096 点以后的 I/O 点的模块。 • 更换最终模块使其占用点数不超过第 4,096 点。	Qn(H) QnPH QnPRH QnU*7 Rem
	• 模块安装在第 37 个或者更后的插槽。 • 模块安装在基板分配设置指定的插槽之后的插槽。 • 模块安装在第 2,048I/O 点之后。 • 模块占用 I/O 点超过了第 2,048 点 I/O 点。	• 卸载安装在第 37 个或者以后的插槽的模块。 • 卸载安装在基板分配设置指定的插槽之后的插槽中的模块。 • 卸载安装在第 2,048 点以后的 I/O 点的模块。 • 更换最终模块使其占用点数不超过第 2,048 点。	Q02U
	• 模块安装在第 25 个或者更后的插槽。（对于 Q00J，是第 17 或者之后的插槽） • 模块安装在基板分配设置指定的插槽之后的插槽。 • 模块安装在第 1,024I/O 点之后。（对于 Q00J，是 256 或者更后的点） • 模块安装在第 1,024I/O 点的界限上。（对于 Q00J，是 256 的界限）	• 卸载安装在第 25 或者更后的插槽中的模块。（对于 Q00J，是第 17 或者更后的插槽） • 卸载安装在高于基板分配设置指定的号码的插槽中的模块。 • 卸载安装在第 1,024 点的 I/O 点之后的模块。（对于 Q00J，是 256 或者之后的点） • 更换最终模块使其占用点数不超过 1,024 点。（对于 Q00J，是 256 点）	Q00J/Q00/Q01
	增加了 5 级或者更多扩展基板。（对 Q00J 是 3 级扩展基板）	卸载第 5 级及其后的扩展基板。（对 Q00J 是 3 级扩展基板以后）	Q00J/Q00/Q01*4

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2125	SP. UNIT LAY. ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时	
2126								
2128	SP. UNIT LAY. ERR.	模块号	-	熄灭	闪烁	停止	电源接通 / 复位时	
2150	SP. UNIT VER. ERR.	模块号（插槽号）	-	熄灭	闪烁	停止	电源接通 / 复位时 / 可编程控制器写入时	
2151								
2200	MISSING PARA.	驱动器名称	-	熄灭	闪烁	停止	电源接通 / 复位时 / STOP → RUN	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2125	• 安装了 QCPU 不能识别的模块。 • 智能功能模块 / 特殊功能模块没有响应。	• 安装可用的模块。 • 智能功能模块 / 特殊功能模块有硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	QCPU Rem
	2126	在多 CPU 系统中 CPU 模块的配置处于下列状态之一： • CPU 模块的左侧 (QCPU 与运动控制器之间）有空闲插槽。 • 在高性能型 QCPU/ 过程 CPU 的左侧安装了除高性能型 QCPU/ 过程 CPU 以外的模块（包括运动控制 CPU)。	• 在空插槽上安装模块。（将空插槽设置到 CPU 模块的右侧。） • 卸下安装在高性能型 QCPU/ 过程 CPU 左侧的其它模块，并将高性能型 QCPU/ 过程 CPU 填补安装在空出来的插槽中。 将运动控制 CPU 安装在高性能型 QCPU/ 过程控制 CPU 的右侧。	Qn(H)*3 QnPH
	2128	冗余系统的扩展基板安装了不可用的模块。	• 卸下安装在扩展基板上的不可用的模块。	QnPRH*6
	2150	在多 CPU 系统中，和多 CPU 系统不兼容的智能功能模块的控制 CPU 被设定为 1 号 CPU 以外的 CPU。	• 更换智能功能模块为一个兼容多 CPU 系统（功能版本 B) 的模块。 • 将和多 CPU 系统不兼容的智能功能模块的控制 CPU 设定为 1 号 CPU。	Q00J/Q00/Q01 QnPH QnU
	2151	下列和冗余系统不兼容的模块被安装到冗余系统。 • MELSECNET/H • 以太网	使用下列和冗余系统兼容的模块。 • MELSECNET/H • 以太网	QnPRH
	2200	被 DIP 开关指定为有效参数驱动器的驱动器中没有参数文件。	• 检查和改正 DIP 开关设定的有效参数驱动器设置。 • 设定参数文件到被 DIP 开关指定为有效参数驱动器的驱动器。	Qn(H) QnPH QnPRH
		程序内存中没有参数文件。	设定参数文件到程序内存。	Q00J/Q00/Q01
		参数有效的所有驱动器中没有参数文件。	将参数文件设定到参数有效驱动器中。	QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2210	BOOT ERROR	驱动器名称	-	熄灭	闪烁	停止	电源接通 / 复位时	
2211								
2220	RESTORE ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
2221								
2300	ICM. OPE. ERROR	驱动器名称	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当存储卡被插入或者卸载时	
2301								
2302								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2210	引导文件的内容不正确。	检查引导设置。	Q00J/Q00/Q01*3 Qn(H) QnPH QnPRH QnU
		DIP 开关的参数有效驱动指定的驱动器中没有引导文件，即使 DIP 开关的引导开关为 ON。	检查和改正由 DIP 开关设定的有效参数驱动器设置。 设定引导文件到被参数驱动器 DIP 开关指定的驱动器。	QnA
	2211	在引导时文件格式化处理失败。	• 重新进行引导。 • CPU 模块有硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H) QnPRH QnU
	2220	• 软元件数据备份功能备份的软元件信息（点数）与可编程控制器参数的软元件点数不同。 发生本出错后，使备份时的软元件点数与可编程控制器参数的软元件点数一致，或者在删除备份数据前通过电源接通 / 复位进行恢复。	• 将可编程控制器参数中设定的软元件点数与备份时的软元件点数设定为一致后，进行电源 OFF → ON 或者复位→复位解除。 • 删除备份数据，将电源 OFF → ON 或者 CPU 复位→复位解除。	QnU
	2221	• 软元件数据备份功能备份的软元件信息不完整。（可能备份时电源关闭 / 复位） 发生本出错时，不要恢复数据。或者删除发生本出错时的不完整软元件信息。	CPU 复位并重新运行。	QnU
	2300	• 在未将存储卡插拔开关切换为 OFF 的状况下，将存储卡拔出。 • 并未安装存储卡时将存储卡插拔开关置于 ON。	• 在将存储卡插拔开关置为 OFF 后，拆卸存储卡。 • 在插入存储卡后，将存储卡插拔开关置于 ON。	QnA Qn(H) QnPH QnPRH QnU
	2301	• 存储卡没有格式化。 • 存储卡格式化状态不正确。 • 闪存卡中不存在 QCPU 文件。	• 格式化存储卡。 • 重新格式化存储卡。 • 将 QCPU 文件写入到闪存卡中。	
		检测出 SRAM 卡故障。（未设置自动格式化时发生）	更换 SRAM 卡电池后格式化 SRAM 卡。	QnU
	2302	安装了 CPU 模块不能使用的存储卡。	• 格式化存储卡。 • 重新格式化存储卡。 • 检查存储卡。	QnA Qn(H) QnPH QnPRH QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2400	FILE SET ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位时 / 可编程控制器写入时	
2401	FILE SET ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位时 / 可编程控制器写入时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2400	通过不能进行标准 ROM 自动写入的 CPU 模块执行了至标准 ROM 的自动写入。（安装了选定为通过引导文件进行至标准 ROM 自动写入的存储卡，并将 DIP 开关的参数有效驱动器设定为存储卡。）	• 通过可进行标准 ROM 自动写入的 CPU 模块执行至标准 ROM 自动写入。 • 使用 GX Developer，将参数和程序写入到标准 ROM 中。 • 更换存储卡为没有设定为标准 ROM 自动写入的存储卡，并进行从存储卡引导的启动操作。	Qn(H)*3 QnPH QnPRH
		在参数的可编程控制器文件设置中指定的文件找不到。	• 使用外围设备读取出错的个别信息，检查对应于数值（参数号）的参数驱动器名称和文件名称，并改正。 • 创建使用参数创建的文件，并将它装载到 CPU 模块中。	○
		为功能版本 “B” 的 QnACPU 增加的以太网参数，被设定到功能版本不是 “B” 的 QnACPU。	将 QnACPU 更换为功能版本为 “B” 的。 删除以太网参数。	QnA
	2401	执行引导操作或者自动写标准 ROM 时超过了程序内存容量。	• 检查和改正参数（引导设置）。 • 删除程序内存中不需要的文件。 • 在参数中为引导选择 “Clear program memory”，在程序内存被清除后进行引导。	Qn(H)*3 QnPH QnPRH
		执行引导操作时超过了程序内存容量。		QnU
		参数指定的文件无法创建。	• 使用外围设备读取出错的个别信息，检查对应于数值（参数号）的参数驱动器名称和文件名称，并改正。 • 检查存储卡中剩余的空间。	○
		虽然在可编程控制器文件设定中设置了使用软元件数据存储文件，但标准 ROM 中创建软元件数据存储文件必需的空间不足。	确保标准 ROM 的可用空间。	QnU

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2402	FILE SET ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位时 / 可编程控制器写入时	
2410	FILE OPE. ERROR	文件名称 / 驱动器名称	程序出错位置	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
2411								
2412								
2413								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	2402	虽然已经在双工通讯设置 / 热备设置中设定了文件寄存器，但是文件寄存器不存在。	确认文件寄存器和参数。	Q4AR
	2410	• 指定的程序在程序内存中不存在。 当执行 ECALL、EFCALL、PSTOP、PSCAN、POFF 或者 PLOW 指令时，可能发生此错误。 • 指定的文件不存在。	• 使用外围设备读取出错的个别信息，检查和数值（程序位置）相对应的程序，并改正。 使用参数创建文件，将其装载到 CPU 模块。 • 指定的文件不存在时，将文件写入到对象存储器中。或者重新通过指令进行文件指定。	QnA Qn(H) QnPH QnPRH QnU
	2411	• 顺控程序不能指定的文件。（如注释文件） • 指定的程序在程序内存中存在，但是没有在参数的程序设置对话框中登记。当执行 ECALL、EFCALL、PSTOP、PSCAN、POFF 或者 PLOW 指令时，可能发生本出错。	使用外围设备读取出错的个别信息，检查和数值（程序位置）相对应的程序，并改正。	
	2412	是不能被顺控程序指定的 SFC 程序文件。	使用外围设备读取出错的个别信息，检查和数值（程序位置）相对应的程序，并改正。	QnA Qn(H) QnPH QnPRH QnU
	2413	数据不能被写入到由顺控程序指定的文件中。	使用外围设备读取出错的个别信息，检查和数值（程序位置）相对应的程序，并改正。确认指定的文件是否处于写保护状态。	QnA Qn(H) QnPH QnPRH

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
2500	CAN'T EXE. PRG.	文件名称 / 驱动器名称	-	熄灭	闪烁	停止	电源接通 / 复位时	
2501								
2502								
2503								
2504								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 可以通过参数选择各模块的出错停止 / 继续运行。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
2500	• 存在有使用了超出可编程控制器参数软元件设置中设定的软元件分配范围的软元件的程序文件。 • 在可编程控制器参数的软元件设置被改变后，只有参数被写入可编程控制器。	• 使用外围设备读取出错的公共信息，检查并确认与数值（文件名称）相对应的可编程控制器参数的软元件设置的软元件分配和程序文件软元件分配，如果需要进行改正。 • 在更改了可编程控制器参数的软元件设置时，应对参数及程序文件批量地进行可编程控制器写入。	○
	在可编程控制器参数的变址修饰被改变后，只有参数被写入可编程控制器。	在更改了可编程控制器参数的变址修饰时，应对参数及程序文件批量地进行可编程控制器写入。	QnU
2501	虽然在可编程控制器参数程序设置中设定了 “无”，但却有多个程序文件存在。	将编辑可编程控制器参数的程序设置修改为 “有”。或者删除不需要的程序文件。	QnA Qn(H) QnPH QnPRH QnU
	• 存在有三个以上的程序文件。 • 程序名称和程序内容不符。	• 删除不需要的程序文件。 • 使程序名称和程序内容相匹配。	Q00J/Q00/Q01
2502	程序文件不正确。 或者文件内容不是用于顺控程序的。	检查程序文件的类型是否是 ***.QPG，并检查文件内容以确认它们是否为顺控程序。	○
	程序文件不是用于冗余 CPU 的。	使用 GX Developer 或者 PX Developer 创建可编程控制器类型被设定为用于冗余 CPU (Q12PRH/Q25PRH) 的程序，并写程序到 CPU 模块。	QnPRH
2503	根本没有程序文件。	• 检查程序配置。 • 检查参数和程序配置。	○
2504	指定了两个或者更多 SFC 常规程序或者控制程序。		QnA Qn(H) QnPH QnPRH QnU
	有两个或者更多 SFC 程序。	将 SFC 程序减少到一个。	Q00J/Q00/Q01*3

*3: 以功能版本 B 以后的 CPU 为对象。
*4: 以功能版本 A 的 CPU 为对象。
*5: 以序列号的高 5 位为 “04101” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

12.1.5 出错代码表 (3000 ～ 3999)

以下介绍出错代码 3000 ～ 3999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3000 3001 3002	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
3000	在多 CPU 系统中，在可编程控制器参数的中断指针设置中指定了由其它 CPU 控制的智能功能模块。	• 指定由自站 CPU 控制的智能功能模块的起始 I/O 编号。 • 删除可编程控制器参数的中断指针设置。	Qn(H)*1 QnPH QnU
	在可编程控制器参数的定时器时限设置、RUN-PAUSE 触点、公共指针号、一般数据处理、空余插槽数、系统中断设置的各设置中超过了 CPU 模块的可使用范围。	• 使用外围设备读取出错的详细信息，检查对应数值（参数号）的参数项目，需要时改正。 • 将修改后的参数再次写入到 CPU 模块中，重新接通可编程控制器的电源，或者对 CPU 模块进行复位。 • 如果仍然发生相同的错误，则可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○ Rem
	在程序内存检查中，设置的检查容量超出了 CPU 模块的可用范围。		QnPH QNPRH*4
	出错个别信息（特殊寄存器 SD16）的参数设置非法。		○
	在可编程控制器文件设置中设置文件寄存器的指定驱动器为“存储卡 ROM”和“使用下述文件”或“使用与程序同名文件”（其中之一）时，ATA 卡被设置到可编程控制器卡插槽。		QnU
3001	参数的内容已被损坏。		○ Rem
3002	在可编程控制器参数的可编程控制器文件设置中为文件寄存器选择了“使用下述文件”时，虽然设定了文件寄存器的容量，但是指定的文件不存在。		QnA Qn(H) QnPH QnPRH
	在可编程控制器参数的可编程控制器文件设置中为文件寄存器设置了“使用下述文件”但未设定文件寄存器的容量时，指定的对象存储器不存在文件寄存器文件。		QnU
	在可编程控制器参数的可编程控制器文件设置中为软元件数据恢复文件设置了“使用下述文件”但未设定容量时，对象存储器不存在软元件数据恢复文件。		QnU

*3: 以序列号的高 5 位为“04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为“07032”以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3003	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	当执行 END 指令时	
	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
3004	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
		多 CPU 系统的自动刷新范围超过了文件寄存器容量。	更换为可进行全部范围刷新的文件寄存器文件。	Qn(H)*1 QnPH QnU
	3003	可编程控制器参数软元件设置中的软元件数设置超过了 CPU 模块的允许范围。	• 使用外围设备读取出错的详细信息，检查对应数值（参数号）的参数项目，需要时改正。 • 如果参数改正后，仍然产生出错，可能是 CPU 模块的内置 RAM 或者程序内存卡的存储器故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○
	3004	参数文件不正确。 或者文件的内容不是参数。	检查参数文件的类型是否是＊＊＊.QPA，并检查文件内容，以确认它们是否为参数。	○

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3005	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
3006	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
3007								
3009								

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3005	参数的内容不完整。	• 使用外围设备读取出错的详细信息，检查对应数值（参数号）的参数项目，需要时改正。 • 将修改后的参数再次写入到 CPU 模块中，重新接通可编程控制器的电源，或者对 CPU 模块进行复位。 • 如果相同的出错再次显示，可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H)*6
	3006	• 在 Q02CPU 中设定了高速中断。 • 在多 CPU 系统中设定了高速中断。 • 当使用 QA1S6 □ B 或者 QA6 □ B 时，设定了高速中断。 • 在高速中断指定的 I/O 地址中没有安装模块。	• 删除 Q02CPU 的高速中断设置。希望使用高速中断时，应将 CPU 模块更换为 Q02H/Q06H/Q12H/Q25HCPU 中的一种。 • 在用于多 CPU 系统时，删除高速中断设置。希望使用高速中断时，应将系统更换为单 CPU 系统。 • 使用 QA1S6 □ B 或者 QA6 □ B 时，删除设置的高速中断。希望使用高速中断时，不要使用 QA1S6 □ B/QA6 □ B。 • 重新检查高速中断设置指定的 I/O 地址。	Qn(H)*3
	3007	由 DIP 开关指定作为有效参数驱动器的驱动器中的参数文件不能用于 CPU 模块。	使用 GX Developer 创建参数，将它们写到由 DIP 开关指定作为有效参数驱动器的驱动器中。	QnPRH
	3009	在多 CPU 系统中，用于 AnS、A、Q2AS 和 QnA 的模块被设定到多个控制 CPU 中。	重新设定参数 I/O 分配，使它们在一个 CPU 模块的控制下。（改变多 CPU 系统中所有 CPU 的参数。）	Qn(H)*1

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3010 3012	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
3013	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
3014	PARAMETER ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3010	CPU 模块的参数设定的个数和在多 CPU 系统中实际使用的个数不同。	使（多 CPU 设置中的 CPU 模块数）-（在 I/O 分配中被设定为空闲的 CPU 数）和实际安装的 CPU 模块数相匹配。	Qn(H)*1 QnPH
	3012	在多 CPU 系统中，多 CPU 设置、控制 CPU 设置与基准 CPU 的设置不同。	使可编程控制器参数中的多 CPU 设置及控制 CPU 设置与基准 CPU(1 号 CPU) 设置相匹配。	Q00/Q01*1 Qn(H)*1 QnU
	3013	在多 CPU 系统中，进行了如下所示的多 CPU 自动刷新设置。 • 当位软元件被指定为刷新软元件时，为起始刷新软元件指定了 16 的倍数以外的号码。 • 指定的软元件不是可以指定的软元件。 • 发送点数是奇数。	检查多 CPU 自动刷新参数中的下列项目，进行改正。 • 当指定位软元件时，为起始刷新软元件指定 16 的倍数。 • 指定可以为刷新软元件指定的软元件。 • 将发送点数设定为偶数。	Qn(H)*1 QnPH
		在多 CPU 系统中，多 CPU 自动刷新设置如下。 • 传送点数的总数超过了刷新点的最大数。	在多 CPU 设置的刷新设置中进行以下检查，如果需要则改正设置。 • 使传送点数的总数在刷新点的最大数之内。	Q00/Q01*1
		在多 CPU 系统中，进行了如下所示的多 CPU 自动刷新设置。 • 指定的软元件不是可以指定的软元件。 • 发送点数是奇数。 • 传送点数的总数超过了刷新点的最大数。	在多 CPU 设置的刷新设置中进行以下检查，如果需要则改正设置。 • 指定可以为刷新软元件指定的软元件。 • 将发送点数设定为偶数。 • 使传送点数的总数在刷新点的最大数之内。	QnU
	3014	• 在多 CPU 系统中，在线模块更换参数（多 CPU 系统参数）设置和基准 CPU 的设置不同。 • 在多 CPU 系统中，在线模块更换设置被设置为允许，但是安装的 CPU 模块并不支持模块的在线更换。	• 使在线模块更换参数和基准 CPU 的参数相匹配。 • 如果安装了不支持在线模块更换的 CPU 模块，将其更换为支持在线模块更换的 CPU 模块。	Qn(H) QnPH QnU

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息 s	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3015	PARAMETER ERROR	文件名称 / 驱动器名称	参数号 / CPU 号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入	
3016	PARAMETER ERROR	文件名称 / 驱动器名称	参数号 / CPU 号	熄灭	闪烁	停止	电源接通 / 复位 / 可编程控制器写入时	
3040 3041	PARAMETER ERROR	–	–	熄灭	闪烁	停止	电源接通 / 复位时	
3042	PARAMETER ERROR	–	–	熄灭	闪烁	停止	电源接通 / 复位时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3015	多 CPU 系统构成时，参数设置与 CPU 校验不一致。	使用外围设备读取出错的详细信息，检查对应数值（参数号 /CPU 号）的参数项目和对象 CPU 参数，需要时改正。	QnU
	3016	多 CPU 同步启动设置中，同步启动的对象设置了与多 CPU 同步启动不兼容的 CPU 模块。	从设置中删除与多 CPU 同步启动设置不兼容的 CPU 模块。	QnU*7
	3040	参数文件已损坏。	通过 GX Developer 将 “可编程控制器参数 / 网络参数 / 远程口令”写入到参数有效驱动器中后，执行系统电源的再接通操作或者 CPU 模块的复位操作。如果仍然发生相同的错误，则可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H)*4 QnPH*4 QnPRH*4
	3041	智能功能模块参数文件已损坏。	通过 GX Developer 将 “智能功能模块参数”写入到参数有效驱动器中后，执行系统电源的再接通操作或者 CPU 模块的复位操作。如果仍然发生相同的错误，则可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H)*4 QnPH*4 QnPRH*4
	3042	存储了远程口令设置内容的系统文件已损坏。	• 通过 GX Developer 将 “可编程控制器参数 / 网络参数 / 远程口令”写入到参数有效驱动器中后，执行系统电源的再接通操作或者 CPU 模块的复位操作。 • 如果仍然发生相同的错误，则可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。） • 在将参数有效驱动器设置为 “程序内存”以外时，通过引导文件设置进行将参数文件 (PARAM) 传送至程序内存中的设置。 通过 GX Developer 将 “可编程控制器参数 / 网络参数 / 远程口令”写入到参数有效驱动器中后，执行系统电源的再接通操作或者 CPU 模块的复位操作。如果仍然发生相同的错误，则可能是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H)*4 QnPH*4 QnPRH*4

*3: 以序列号的高 5 位为 “04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032”以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102”以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3100	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3100	在多 CPU 系统中，在 MELSECNET/G 的网络参数的起始 I/O 号中指定了其它 CPU 控制的 MELSECNET/G 模块。	• 删除由其它 CPU 控制的 MELSECNET/G 模块的网络参数。 • 将设置更改为由自站 CPU 控制的 MELSECNET/G 模块的起始 I/O 号。	Qn(H)*6 QnU
		在普通站中运行的 MELSECNET/G 模块的网络参数被改写为控制站，或者在控制站中运行的 MELSECNET/G 模块的网络参数被改写为普通站。（网络参数可以通过复位而反映在模块侧。）	复位 CPU 模块。	
		• 实际安装的模块数和 MELSECNET/G 的模块设置参数中指定的数目不同。 • 实际安装的模块的起始 I/O 号码和 MELSECNET/G 的网络参数中设定的不同。 • 参数中的某些数据不能处理。 • MELSECNET/G 的站点类型在电源接通的情况下被改变。（改变站点类型需要 RESET → RUN。）	• 检查网络参数和安装状态，如果它们不同，使网络参数与安装状态相匹配。如果改正了任何一个网络参数，则将其写到 CPU 模块。 • 确认扩展基板的扩展级数设置。 • 检查扩展基板和扩展电缆的连接状态。 当 GOT 是总线连接到主基板和扩展基板时，也要检查它们的连接状态。 如果在上面的检查之后还发生出错，可能的原因是硬件故障。（联系您当地的三菱代理商，请说明故障症状。）	
		• MELSECNET/G 模块指定为 MELSECNET/H 模块网络参数的起始 I/O 号码。 • MELSECNET/H 模块指定为 MELSECNET/G 模块网络参数的起始 I/O 号码		
		• 虽然安装了 MELSECNET/G 模块，但是 MELSECNET/G 网络参数没有设置。 • 虽然安装了 MELSECNET/G 模块和 MELSECNET/H 模块，但是 MELSECNET/H 模块网络参数没有设置。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3101	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	当执行 END 指令时	
	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3101	链接刷新范围超出了文件寄存器的容量。	更换文件寄存器文件为匹配全局刷新的文件寄存器文件。	Qn(H)*1 QnPH QnPRH QnU
		• 当 MELSECNET/H 模块的站号是 0 时，进行了可编程控制器之间的网络参数设置。 • 当 MELSECNET/H 模块的站号是 0 以外的数字时，进行了远程主站的参数设置。	改正网络参数中 MELSECNET/H 模块的类型或者站号，使其符合实际使用的系统。	Qn(H)*1 QnPH QnPRH
		MELSECNET/G 刷新参数超出范围。	• 检查网络参数和安装状态，如果它们不同，匹配网络参数和安装状态。 如果改正了任何一个网络参数，将其写到 CPU 模块。 • 确认扩展基板的扩展级数的设置。 • 检查扩展基板和扩展电缆的连接状态。 当 GOT 是总线连接到主基板和扩展基板，也要检查它们的连接状态。 如果在上面的检查之后还发生出错，可能的原因是硬件故障。（联系您当地的三菱代理商，请说明故障症状。）	Qn(H)*6 QnU
		• 网络参数指定的网络号不同于实际安装的网络号。 • 网络参数指定的起始 I/O 号不同于实际安装的 I/O 模块。 • 网络参数指定的网络级不同于的实际安装的网络。 • MELSECNET/H 和 MELSECNET/10 的网络刷新参数超过了指定的范围。		○
		使用不支持 MELSECNET/H 多重远程 I/O 网络的模块配置了多远程 I/O 网络。	使用支持 MELSECNET/H 多远程 I/O 网络的模块。	QnPH
		• MELSECNET/H 远程主站的 A 系统被设定为 0 号站以外的站点。 • MELSECNET/H 远程主站的 B 系统被设定为 0 号站。	• 在 CPU 模块中将 A 系统侧的 MELSECNET/H 远程主站设置为 0 号站。 • 在 CPU 模块中将 B 系统侧的 MELSECNET/H 远程主站设置为 1 ～ 64 号站中的一个。	QnPRH

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3102	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
3103	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3102	检测到 MELSECNET/G 网络参数出错。	• 改正和写网络参数。 • 如果改正之后还发生出错，这表示是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Qn(H)*6 QnU
		• 网络模块检测到网络参数出错。 • 检测到 MELSECNET/H 网络参数出错。		○
		双工通讯设置中指定的站号不正确。 • 没有对站连续编号。 • 没有为常规站上的 CPU 模块进行双工通讯设置。	参考网络模块的故障排除，如果出错是由于不正确的双工通讯设置引起的，重新检查网络参数的双工通讯设置。	QnPRH
		安装了序列号的高 5 位为 “09041”以前的 MELSECNET/G 模块。	安装序列号的高 5 位为 “09042”以后的 MELSECNET/G 模块。	QnU
	3103	在多 CPU 系统中，受其它站控制的 Q 系列以太网接口模块被指定为以太网网络参数的起始 I/O 号。	• 删除受其它站点控制的 Q 系列以太网接口模块的以太网网络参数。 • 将设置更改为受自站控制的 Q 系列以太网接口模块的起始 I/O 号。	Q00/Q01*1 Qn(H)*1 QnPH QnU
		• 虽然在以太网模块数设置参数设置将模块数设定为 1 或者更大的数，但是实际安装的模块数是 0。 • 以太网网络参数的起始 I/O 号和实际安装的模块的 I/O 号不同。	• 改正和写网络参数。 • 如果改正之后还发生出错，这表明是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○ Rem
		• 在参数设定的 I/O 号的位置上没有 AJ71QE71 存在。 • I/O 号码指定重复。 • 网络参数的数目和安装的 AJ71QE71 不同。 • 设定的以太网（参数 + 专用指令）超过 5 个。		QnA
		• 在冗余系统的扩展基板上安装了网络类型设置为 “以太网（主基板）”的以太网模块。 • 在冗余系统的主基板上安装了网络类型设置为 “以太网（扩展基板）”的以太网模块。		QnPRH*6

*3: 以序列号的高 5 位为 “04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032”以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102”以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3104	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
3105								
3106	LINK PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	当执行 END 指令时	
	LINK PARA. ERROR	文件名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
3107	LINK PARA. ERROR	文件名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3104	• 以太网和 MELSECNET/10 使用相同的网络号。 • 网络参数设定的网络号、站号或者组号超过范围。 • I/O 号指定时超出了使用的 CPU 模块的范围。 • 以太网特有参数设置不正常。	• 改正和写网络参数。 • 如果改正之后还发生出错，这表明是硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	○ Rem
	3105	在多 CPU 系统中，受其它站控制的 Q 系列 CC-Link 模块被指定为 CC-Link 网络参数的起始 I/O 号。	• 删除受其它站点控制的 Q 系列 CC-Link 模块的 CC-Link 网络参数。 • 将设置更改为受自站控制的 Q 系列 CC-Link 模块的起始 I/O 号码。	Q00/Q01*1 Qn(H)*1 QnPH QnU
		• 虽然在网络参数中设定的 CC-Link 模块数是一个或者更多，实际安装的模块数却是 0。 • 公共参数中的起始 I/O 号和实际安装的 I/O 号不同。 • CC-Link 模块数设置参数的站类型不同于实际安装的站类型。	• 改正和写入网络参数。 • 如果改正之后还发生出错，这表明是硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	○ Rem
		• 在冗余系统的扩展基板上安装了站类型设置为 “主站（冗余功能对应）” 的 CC-Link 模块。 • 在冗余系统的主基板上安装了站类型设置为 “主站” 的 CC-Link 模块。		QnPRH*6
		以太网参数的内容不正确。	改正参数后重新写入。	QnA
	3106	CC-Link 链接刷新范围超过文件寄存器容量。	将文件寄存器文件更换为全局范围内可以刷新的文件寄存器文件	Qn(H)*1 QnPH QnPRH QnU
		用于 CC-Link 的网络刷新参数超过范围。	检查参数设置。	QCPU Rem
	3107	• CC-Link 参数设置不正确。 • 设定模式不能用于安装的 CC-Link 模块的版本。	检查参数设置。	○ Rem

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3200	SFC PARA. ERROR	文件名称	参数号	熄灭	闪烁	停止	STOP→ RUN	
3201								
3202								
3203								
3300	SP.PARA ERROR	文件名称	参数号 *2	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
3301								
3302								
3303	SP.PARA ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	3200	参数设置非法。 • 虽然在可编程控制器参数对话框的 SFC 设置中将块 0 设定为“Automatic start”，但是块 0 并不存在。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA Q00J/Q00/Q01*1 QnPH QnPRH QnU
	3201	块参数设置非法。		QnA Qn(H) QnPH QnPRH
	3202	在可编程控制器参数对话框的软元件设置中指定的步继电器数少于程序中已经使用的个数。		
	3203	在可编程控制器参数对话框的程序设置中指定的 SFC 程序的执行类型不是扫描执行类型。		QnA Qn(H) QnPH QnPRH QnU
	3300	在 GX Configurator 上设定的智能功能模块参数中的起始 I/O 号码和实际 I/O 号不同。	检查参数设置。	QCPU Rem
	3301	智能功能模块的刷新设置超过文件寄存器容量。	更换文件寄存器文件为允许全局内刷新的文件寄存器文件。	Q00J/Q00/Q01 Qn(H)*1 QnPH QnPRH QnU
		智能功能模块的刷新参数设置超出可用范围。	检查参数设置。	QCPU Rem
	3302	智能功能模块的刷新参数异常。	检查参数设置。	QCPU
	3303	在多 CPU 系统中，自动刷新设置或者相似参数设置被设定到受其它站控制的智能功能模块。	• 删除由其它 CPU 控制的智能功能模块的自动刷新设置或者相似参数设置。 • 将设置更改为由自站 CPU 控制的智能功能模块的自动刷新设置或者相似参数设置。	Q00/Q01*1 Qn(H)*1 QnPH QnU

*3: 以序列号的高 5 位为“04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为“07032”以后的 CPU 为对象。
*5: 以序列号的高 5 位为“08102”以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为“09012”以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
3400	REMOTE PASS. ERR.	–	–	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
3401								

*1: 以功能版本 B 以后的 CPU 为对象。
*2 参数号为 GX Configurator 设置的智能功能模块的参数起始 I/O 号除以 10H 的值。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
3400	远程密码文件中的目标模块的起始 I/O 号码被设定为 0H 到 0FF0H 以外。	将目标模块的起始 I/O 号码更改为 0H 到 0FF0H 范围之内。	Qn(H)*1 QnPH QnPRH QnU*7 Rem
	远程密码文件中的目标模块的起始 I/O 号码被设定为 0H 到 07E0H 以外。	将目标模块的起始 I/O 号码更改为 0H 到 07E0H 范围之内。	Q02U
	远程密码的目标模块的起始 I/O 号码在下列范围之外。 • Q00JCPU: 0H 到 1E0H • Q00CPU/Q01CPU: 0H 到 3E0H	将远程密码的目标模块的起始 I/O 号码更改为下列范围之内的号码。 • Q00JCPU: 0H 到 1E0H • Q00CPU/Q01CPU: 0H 到 3E0H	Q00J/Q00/Q01*1
3401	由于下列原因之一，远程密码文件的起始 I/O 号码指定的位置不正确： • 模块没有安装。 • 不是智能功能模块（I/O 模块） • 智能功能模块不是 Q 系列串行通讯模块、调制解调器接口模块或者以太网模块。 • 是功能版本 A 的 Q 系列串行通讯模块或者以太网模块。	将功能版本为 B 或者更高的 Q 系列串行通讯模块、调制解调器接口模块或者以太网模块安装到远程密码文件的起始 I/O 号指定的位置	Qn(H)*1 QnPH QnPRH QnU Rem
	在远程密码的起始 I/O 号指定的插槽中未安装下列模块中的一个。 • 功能版本为 B 或者更高的串行通讯模块 • 功能版本为 B 或者更高的以太网模块 • 功能版本为 B 或者更高的调制解调器接口模块	在远程密码的起始 I/O 号指定的位置安装下列模块中的一个。 • 功能版本为 B 或者更高的串行通讯模块 • 功能版本为 B 或者更高的以太网模块 • 功能版本为 B 或者更高的调制解调器接口模块	Q00J/Q00/Q01*1
	在多 CPU 系统中指定了受其它 CPU 控制的功能版本为 B 或者更高的 Q 系列串行通讯模块、调制解调器接口模块或者以太网模块。	• 将其更换为主站 CPU 控制的功能版本为 B 或者更高的以太网模块。 • 删除远程密码设置。	Qn(H)*1 QnPH QnU

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “08102” 以后的 MELSECNET/H 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

12.1.6 出错代码表 (4000 ～ 4999)

以下介绍出错代码 4000 ～ 4999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4000	INSTRCT. CODE ERR	程序出错位置	-	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN/ 当执行指令时	
4001								
4002								
4003								
4004								
4010	MISSING END INS.	程序出错位置	-	熄灭	闪烁	停止	电源接通 / 复位 /STOP → RUN 时	
4020	CAN'T SET(P)	程序出错位置	-	熄灭	闪烁	停止		
4021								
4030	CAN'T SET(I)	程序出错位置	-	熄灭	闪烁	停止		

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)

*2: 以功能版本 B 以后的 CPU 为对象。

出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
4000	• 程序包含不能被解码的指令代码。 • 程序中包含不能使用的指令。	使用外围设备读取出错的公共信息，检查和数值（程序出错位置）相对应的出错步，并改正错误。	○
4001	程序包含 SFC 专用指令，尽管它不是 SFC 程序。		QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
4002	• 程序指定的专用指令有不合法指令名称。 • 程序中指定的专用指令不能被指定的模块执行。		○ Rem
4003	程序指定的专用指令有不正确的软元件数。		
4004	程序指定的专用指令包含不能使用的软元件。		
4010	程序中没有 END(FEND) 指令。		○
4020	程序使用的内部文件指针的总数超过了参数中设定的内部文件指针数。		QnA Qn(H) QnPH QnPRH QnU
4021	• 各文件中使用的公共指针号有重复。 • 各文件中使用的本地指针号有重复。		○
4030	各文件中使用的中断指针号有重复。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4100	OPERATION ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
4101								
4102								
4103								
4104								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

出错代码(SD0)	出错内容及原因	对策	相应的 CPU
4100	指令中包含有不能处理的数据。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	○ Rem
	通过 FREAD/FWRITE 指令访问 ATA 卡时发生异常。	• 采取抗噪声措施。 • 使 CPU 模块复位后再次 RUN。如果仍然显示了相同的出错，则表明 ATA 卡的硬件异常。（请联系附近的 FA 中心或代理商，说明故障症状。）	Qn(H) QnPH QnPRH
4101	• 通过指令处理的数据的设置使用数超过了可用范围。 • 通过指令指定的软元件的存储数据、常量超过了可用范围。 • 在至自站 CPU 共享存储区的写入中，将禁止写入的区域指定为写入目标地址。 • 通过指令指定的软元件的存储数据重复。 • 通过指令指定的软元件超出了软元件点数范围。 • 通过指令指定的中断指针号超过了可用范围。	通过外部设备读取出错的公共信息，对该数值（程序出错位置）对应的出错步进行检查、修改。	○ Rem
	• 通过指令指定的文件寄存器的存储数据超过了可用范围，或者文件寄存器未设置。		QnU
4102	在多 CPU 系统中，为受其它站控制的网络模块指定了链接直接软元件(J□\□)。	• 从程序中删除链接直接软元件，其指定了由其它 CPU 控制的网络模块。 • 使用链接直接软元件，指定由自站 CPU 控制的网络模块。	Q00/Q01*2 Qn(H)*2 QnPH QnU
	• 为网络专用指令指定的网络号或者站号有错误。 • 链接直接软元件（J □\ □ ）设置不正确。 • 为专用指令指定的模块号 / 网络号 / 字符串数超过允许的范围。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	○ Rem
	• 专用指令指定的字符串的指定（ “” ）不是可使用的字符号。		QnU
4103	PID 专用指令的配置不正确。		QnA Q00J/Q00/Q01*2 Qn(H) QnPRH QnU
4104	设置数超过范围。	使用外围设备读取出错的公共信息，并检查和改正对应数值（程序出错位置）的程序。	Q4AR

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4105	OPERATION ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
4107								
4108								
4109	OPERATION ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
4111	OPERATION ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *2	当执行指令时	
4112								
4113	OPERATION ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行	当执行指令时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4105	设置了程序内存检查时，执行了 PLOADP/PUNLOADP/PSWAPP 指令。	• 删除程序内存检查设置。 • 使用程序内存检查时，删除 PLOADP/PUNLOADP/PSWAPP 指令。	QnPH*5
	4107	通过一个 CPU 模块执行了 33 条以上的多 CPU 专用指令。	使用多 CPU 专用指令的结束位进行互锁，防止在一个 CPU 模块中执行 33 条以上的多 CPU 专用指令。	Q00/Q01*2 Qn(H)*2 QnPH Q02U
		CC-Link 指令的执行数超过 32。	将 CC-Link 指令的执行数设定为 32 以下。	QnA
	4108	执行 CC-Link 指令时，没有设定 CC-Link 参数。	在设置了 CC-Link 参数之后，执行 CC-Link 指令。	
	4109	设置了高速中断时执行了 PR、PRC、UDCNT1、UDCNT2、PLSY 和 PWM 指令。	删除高速中断设置。 当使用高速中断时，删除 PR、PRC、UDCNT1、UDCNT2、PLSY 和 PWM 指令。	Qn(H)*3
	4111	试图用指令在主站 CPU 模块的 CPU 共享存储区的写入禁止区域中执行写入操作。	使用 GX Developer 读取出错的公共信息，并检查和改正对应数值（程序出错位置）的出错步。	Q00/Q01*2 QnU
	4112	指定了不能用多 CPU 专用指令指定的 CPU 模块。		
	4113	• 执行 SP.DEVST 指令时，当天写入标准的次数超过了 SD695 指定的值。 • 设定了超出 SD695 指定范围的值。	• 检查 SP.DEVST 指令为正确的执行次数。 • 次日或以后再次执行 SP.DEVST 指令，或者调整 SD695 的值。 • 改正超出范围的 SD695 值。	QnU

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4120	OPERATION ERROR	程序出错位置	-	熄灭 / 亮灯	闪烁 / 亮灯	停止 / 继续运行 *2	当执行指令时	
4121								
4122	OPERATION ERROR	程序出错位置	-	熄灭 / 亮灯	闪烁 / 亮灯	停止 / 继续运行	当执行指令时	
4130	OPERATION ERROR	程序出错位置	-	熄灭 / 亮灯	闪烁 / 亮灯	停止 / 继续运行	当执行 END 指令时 / 当执行其它指令时	
4131	OPERATION ERROR	程序出错位置	-	熄灭 / 亮灯	闪烁 / 亮灯	停止 / 继续运行 *2	当执行指令时	
4140	OPERATION ERROR	程序出错位置	-	熄灭 / 亮灯	闪烁 / 亮灯	停止 / 继续运行 *2	当执行指令时	
4141								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4120	由于手动切换允许标志（特殊寄存器 SM1592）为 OFF，所以不能由控制系统切换指令（SP.CONTSW）执行手动切换。	通过 SP.CONTSW 指令执行控制系统切换时，将手动切换允许标志（特殊寄存器 SM1592）置于 ON 后再进行操作。	QnPRH
	4121	• 在独立模式下，在待机系统的 CPU 模块中执行了控制系统切换指令（SP.CONTSW）。 • 在调试模式下，执行了控制系统切换指令（SP.CONTSW）。	• 重新检查用于 SP.CONTSW 指令的互锁信号，并确保 SP.CONTSW 指令是只能在控制系统中执行。（由于 SP.CONTSW 指令不能在待机系统中执行，建议使用运行模式信号或者类似的信号来提供互锁。） • 由于在调试模式中不能执行 SP.CONTSW 指令，重新检查和运行模式相关的互锁信号。	
	4122	• 对冗余系统扩展基板上安装的模块，执行了专用指令。 • 在独立模式下，对待机系统的扩展基板上安装的智能功能模块，执行了访问指令。	• 删除用于扩展基板上安装的模块的专用指令。 • 删除对待机系统的扩展基板上安装的智能功能模块进行访问的指令。	QnPRH*6
	4130	以 ATA 卡内的注释文件作为对象，执行了 SFC 步注释读取指令（S(P).SFCSCOMR）、SFC 转移条件注释读取指令（S(P).SFCTCOMR）。	将对象注释文件设置为 ATA 卡内的注释文件以外。	Qn(H)*4 QnPH*5 QnPRH
	4131	另一个 SFC 程序还没有结束时，通过指令启动了 SFC 程序。	检查指令指定的 SFC 程序，或者检查 SFC 程序的执行状态。	QnU
	4140	输入数据用特殊值（“-0”、非正规数、非数、±∞）进行了运算。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnU
	4141	运算时发生了溢出。		

*3: 以序列号的高 5 位为“04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为“07012”以后的 CPU 为对象。
*5: 以序列号的高 5 位为“07032”以后的 CPU 为对象。
*6: 以序列号的高 5 位为“09012”以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4200	FOR NEXT ERROR	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4201	FOR NEXT ERROR	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4202								
4203								
4210	CAN'T EXECUTE(P)	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4211								
4212								
4213								
4220	CAN'T EXECUTE(I)	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4221								
4223								
4225	CAN'T EXECUTE(I)	-	-	熄灭	闪烁	停止	电源接通 / 复位时	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码(SD0)	出错内容及原因	对策	相应的 CPU
	4200	在 FOR 指令执行之后未执行 NEXT 指令。 或者 NEXT 指令个数少于 FOR 指令的个数。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	○
	4201	在未执行 FOR 指令的情况下执行了 NEXT 指令。 或者 NEXT 指令的个数多于 FOR 指令的个数。		
	4202	嵌套超过了 16 级。	将嵌套级数保持为 16 以内。	
	4203	在未执行 FOR 指令的情况下执行了 BREAK 指令。	使用外围设备读取出错的公共信息，检查对应数值（ 程序出错位置 ）的出错步，并改正错误。	
	4210	执行了 CALL 指令，但是指定的指针处没有子程序。		
	4211	执行的子程序中不存在 RET 指令。		
	4212	在主程序中，RET 指令在 FEND 指令之前。		
	4213	嵌套超过了 16 级。	将嵌套级数保持为 16 以内。	
	4220	虽然有中断输入发生，但是相应的中断指针却不存在。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置） 的出错步，并改正错误。	○
	4221	在执行的中断程序中不存在 IRET 指令。		
	4223	在主程序中，IRET 指令在 FEND 指令之前。		
		• 固定扫描周期执行类型程序中执行了 IRET 指令。 • 固定扫描周期执行类型程序中执行了 STOP 指令。		QnU
	4225	冗余系统扩展基板上安装的模块设置了中断指针。	扩展基板上安装的模块设置的中断指针不能使用，删除设置。	QnPRH*6

*3: 以序列号的高 5 位为 “04012”以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012”以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032”以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4230	INST. FORMAT ERR.	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4231								
4235								
4300	EXTEND INST. ERR.	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
4301								
4350	OPERATION ERROR	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4351								
4352								
4353								
4354								
4355								
4400	SFCP. CODE ERROR	程序出错位置	-	熄灭	闪烁	停止	STOP → RUN	
4410	CAN'T SET(BL)	程序出错位置	-	熄灭	闪烁	停止	STOP → RUN	
4411								
4420	CAN'T SET(S)	程序出错位置	-	熄灭	闪烁	停止	STOP → RUN	

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4230	CHK 和 CHKEND 指令的数目不相等。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA Qn(H) QnPH
	4231	IX 和 IXEND 指令的数目不相等。		○
	4235	CHK 指令的检查条件的配置不正确。 或者在低速执行类型程序中使用了 CHK 指令。		QnA Qn(H) QnPH
	4300	MELSECNET/MINI-S3 主站模块控制指令的目标是错误的。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA
	4301	AD57/AD58 控制指令的目标是错误的。		
	4350	• 程序指定的多 CPU 高速总线专用指令指定的对象 CPU 模块出错。 • 指定了被设置为预约的 CPU。 • 指定了未安装的 CPU。 • 对象 CPU 的起始 I/O 号除以 16(n1) 超出了 3E0H 到 3E3H 的范围。 • 指定了不能执行指令的 CPU 模块。 • 单 CPU 系统中执行指令。 • 指定了自 CPU。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnU
	4351	• 程序指定的多 CPU 高速总线专用指令，不能在指定的对象 CPU 模块中执行。 • 指令名出错。 • 指定了对象 CPU 模块不支持的指令。		QnU*7
	4352	程序指定的多 CPU 高速总线专用指令的软元件数出错。		
	4353	程序指定了多 CPU 高速总线专用指令的不能使用的软元件。		QnU
	4354	指定了多 CPU 高速总线专用指令的不能处理的字符串。		QnU*7
	4355	程序指定的多 CPU 高速总线专用指令的读取 / 写入数据次数（请求 / 接收数据次数）无效。		
	4400	SFC 程序中没有 SFCP 或者 SFCPEND 指令。	使用 GX Developer 重新写程序到 CPU 模块。	QnA Qn(H) QnPH QnPRH
	4410	SFC 程序指定的块号超出范围。		QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
	4411	SFC 程序中，块号指定重复。		
	4420	SFC 程序指定的步号超出范围。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4421	CAN'T SET(S)	程序出错位置	-	熄灭	闪烁	停止	STOP → RUN	
4422								
4423								
4430	SFC EXE. ERROR	文件名称 / 驱动器名称	-	熄灭	闪烁	停止	STOP → RUN	
4431								
4432								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4421	所有 SFC 程序中的总步数超出最大值。	使用 GX Developer 重新写程序到 CPU 模块。	QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
	4422	SFC 程序中步号指定重复。		
	4423	各块的（最大步号 +1）的总数超出总步进继电器数。	改正各块的（最大步号 +1）的总数，避免超出步进继电器的总数。	Q00J/Q00/Q01*2 QnU
	4430	SFC 程序不能被执行。 • 块数据设置的数据非法。 • 块数据设置的 SFC 数据软元件超出可编程控制器参数中设定的软元件设置范围。	• 使用 GX Developer 重新写程序到 CPU 模块。 • 在改正 SFC 数据软元件的设置之后，将其写入 CPU 模块。 • 在改正可编程控制器参数中设定的软元件设置范围之后，将其写入 CPU 模块。	
	4431	SFC 程序不能执行。 • 块参数设置异常。	使用 GX Developer，重新写程序到 CPU 模块。	
	4432	SFC 程序不能执行。 • SFC 程序的结构非法。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4500	SFCP. FORMAT ERR.	程序出错位置	-	熄灭	闪烁	停止	STOP → RUN	
4501								
4502								
4503								
4504								
4505								
4506								
4600	SFCP. OPE. ERROR	程序出错位置	-	熄灭 / 点亮	闪烁 / 点亮	停止 / 继续运行 *1	当执行指令时	
4601								
4602								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4500	SFC 程序中，BLOCK 指令数和 BEND 指令数不相等。	使用外围设备，重新写程序到 CPU 模块。	QnA Qn(H) QnPH QnPRH
	4501	SFC 程序中，STEP* ～ TRAN* ～ TSET ～ SEND 指令的配置不正确。		
	4502	SFC 程序的结构非法。 ・在 SFC 程序块中，不存在 STEPI* 指令。		QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
	4503	SFC 程序的结构非法。 ・在 TSET 指令中指定的步不存在。 ・在跳跃转移中，主步号被指定为目标步号。	・使用 GX Developer，重新写程序到 CPU 模块。 ・使用 GX Developer 读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	
	4504	SFC 程序的结构非法。 ・在 TAND 指令中指定的步不存在。	使用 GX Developer，重新写程序到 CPU 模块。	
	4505	SFC 程序的结构非法。 ・在步的运算输出中，为主步指定了 SETSn/BLmSn 或者 RSTSn/BLmSn 指令。	使用 GX Developer 读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	Q00J/Q00/Q01*2 QnU
	4506	SFC 程序的结构非法。 ・在复位步中，主步号被指定为目标步。		
	4600	SFC 程序包含不能处理的数据。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA Qn(H) QnPH QnPRH
	4601	超出 SFC 程序可以指定的软元件范围。		
	4602	在 SFC 程序中的块控制中在执行 START 指令之前执行了 END 指令。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
4610	SFCP. EXE. ERROR	程序出错位置	-	点亮	点亮	继续运行	STOP → RUN	
4611								
4620	BLOCK EXE. ERROR	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4621								
4630	STEP EXE. ERROR	程序出错位置	-	熄灭	闪烁	停止	当执行指令时	
4631								
4632								
4633								

*1: 可以通过参数设置发生错误时的 CPU 的运行状态。(LED 显示也连动变化。)
*2: 以功能版本 B 以后的 CPU 为对象。

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	4610	通过 SFC 程序进行继续运行启动时的激活步信息不正确。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。 程序自动地进行初始化启动。	QnA Qn(H) QnPH QnPRH
	4611	通过 SFC 程序中指定继续运行启动时，在 RUN 过程中被复位。		
	4620	通过 SFC 程序对已经启动的块执行了启动操作。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	
	4621	通过 SFC 程序试图在不存在的块上执行启动操作。	• 使用 GX Developer 读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。 • 如果特殊继电器 SM321 为 OFF，将其变为 ON。	QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
	4630	通过 SFC 程序在已经启动的块上执行了启动操作。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA Qn(H) QnPH QnPRH
	4631	• 通过 SFC 程序对不存在的步执行了启动。 或者通过 SFC 程序对不存在的步进行了结束指定。 • 通过 SFC 程序对不存在的转移条件进行了强制转移。 或者通过 SFC 程序对不存在的转移条件的强制转移进行了解除。	• 使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。 • 如果特殊继电器 SM321 为 OFF，将其变为 ON。	QnA Q00J/Q00/Q01*2 Qn(H) QnPH QnPRH QnU
	4632	在可以被 SFC 程序指定的块中，有太多的同时激活步。	使用外围设备读取出错的公共信息，检查对应数值（程序出错位置）的出错步，并改正错误。	QnA Qn(H) QnPH QnPRH QnU
	4633	在可以被指定的所有块中，有太多的同时激活步。		

*3: 以序列号的高 5 位为 “04012” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07012” 以后的 CPU 为对象。
*5: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。
*7: 除 Q02UCPU 外的通用型 QCPU。

12.1.7 出错代码表 (5000 ～ 5999)

以下介绍出错代码 5000 ～ 5999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
5000 5001	WDT ERROR	时间（设定值）	时间 （实际测量值）	熄灭	闪烁	停止	常时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	5000	• 初始化执行类型程序的扫描时间超出了在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的初始化执行监视时间。	• 用外围设备读取出错的个别信息，检查它的值（时间），并缩短扫描时间。 • 在可编程控制器参数的可编程控制器 RAS 设置中更改初始执行监视时间或者 WDT 设置值。 • 解除由跳跃转移引起的无限循环。	QnA Qn(H) QnPH QnPRH QnU
		• 待机系统发生了电源 OFF。 • 在待机系统的电源未 OFF 或者复位的情况下，进行了热备电缆的插拔操作。 • 热备电缆没有用连接器固定螺钉固定。	• 由于待机系统的掉电使控制系统的扫描时间延迟，因此应重新设定 WDT 值，并将控制系统扫描时间的延迟考虑在内。 • 当运行期间热备电缆脱落时，应牢固地连接电缆后重启系统。 如果相同的出错再次显示，原因可能是热备电缆或者 CPU 模块的硬件故障。（请联系附近的三菱服务中心或者代理商，说明故障症状。）	QnPRH
	5001	• 程序扫描时间超出在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的 WDT 值。	• 用外围设备读取出错的个别信息，检查它的值（时间），并缩短扫描时间。 • 在可编程控制器参数的可编程控制器 RAS 设置中更改初始执行监视时间或者 WDT 设置值。 • 解除由跳跃转移引起的无限循环。	○
		• 待机系统发生了电源 OFF。 • 在待机系统的电源未 OFF 或者复位的情况下，进行了热备电缆的插拔操作。 • 热备电缆没有用连接器固定螺钉固定。	• 由于待机系统的掉电使控制系统的扫描时间延迟，因此应重新设定 WDT 值，并将控制系统扫描时间的延迟考虑在内。 • 当运行期间热备电缆脱落时，应牢固地连接电缆后重启系统。 如果相同的出错再次显示，原因可能是热备电缆或者 CPU 模块的硬件故障。（请联系附近的三菱服务中心或者代理商，说明故障症状。）	QnPRH

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
5010	PRG. TIME OVER	时间（设定值）	时间（实际测量值）	点亮	点亮	继续运行	常时	
5011								

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	5010	程序扫描时间超出了在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的恒定扫描时间。	• 重新检查恒定扫描设置时间。 • 重新检查可编程控制器参数中的恒定扫描时间和低速程序执行时间，以保证有足够的恒定扫描的剩余时间。	QnA Qn(H) QnPH QnPRH QnU
		在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的低速程序执行时间超出了恒定扫描的剩余时间。		QnA Qn(H) QnPH QnPRH
		程序扫描时间超出在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的恒定扫描时间。	重新检查可编程控制器参数中的恒定扫描时间，以保证有足够的恒定扫描时间。	Q00J/Q00/Q01
	5011	低速执行类型程序的扫描时间超出在可编程控制器参数对话框的可编程控制器 RAS 设置中指定的低速执行监视时间。	从外围设备读取出错的个别信息，检查其中的数值（时间），如果需要则缩短扫描时间。在可编程控制器参数的可编程控制器 RAS 设置中更改低速执行监视时间。	QnA Qn(H) QnPH

12.1.8 出错代码表 (6000 ～ 6999)

以下介绍出错代码 6000 ～ 6999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6000	PRB. VERIFY ERR.	文件名称	-	熄灭	闪烁	停止	常时	
	FILE DIFF.	文件名称	-	熄灭	闪烁	停止	电源接通 / 复位 / 连接热备电缆 / 备份模式改变 /RUN 中写入结束 / 系统切换 / 双系统 RUN 切换时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6000	冗余系统中的控制系统和待机系统的程序和参数设置不同。(只在冗余系统的待机系统中才可检测上述内容。)	使控制系统和待机系统的程序和参数相同。	Q4AR
		在冗余系统中，控制系统和待机系统的程序和参数不相同。 检测到的、在两个系统之间有差别的文件类型可以通过出错公共信息的文件名称来确认。 • 程序不同。 (文件名称 = ********.QPG) • 可编程控制器参数 / 网络参数 / 冗余参数不同。 (文件名称 = PARAM.QPA) • 远程密码不同。 (文件名称 = PARAM.QPA) • 智能功能模块参数不同。 (文件名称 = IPARAM.QPA) • 软元件初始值不同。 (文件名称 = ********.QDI) • CPU 内可用于多个程序块运行中写入的区域容量不同。 (文件名称 = MBOC.QMB) (只在冗余系统的待机系统中才可检测上述内容。)	• 使控制系统和待机系统的程序和参数相同。 • 按照下面的步骤 1) 或 2)，执行可编程控制器校验，以确认两个系统的文件之间的差异，然后改正错误的文件，并将改正后的文件重新写入可编程控制器。 1) 使用 GX Developer 或者 PX Developer 读取 A 系统的程序 / 参数后，与 B 系统的程序 / 参数进行校验。 2) 将 GX Developer 或者 PX Developer 在离线环境中保存的程序 / 参数与写入到两个系统的 CPU 模块中的程序 / 参数进行校验。 • 在两个系统的可用于多个程序块运行中写入的区域容量不相同时，采取以下对策 1) 或者 2)。 1) 使用从控制系统到待机系统的内存复制功能，将控制系统的程序内存的内容复制到待机系统。 2) 格式化两个系统的 CPU 模块程序内存。(将两个系统的可用于多个程序块运行中写入的区域容量设置为相同。)	QnPRH

*1: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6001	FILE DIFF.	-	-	熄灭	闪烁	停止	电源接通 / 复位 / 连接热备电缆 / 运行模式改变时	
6010	MODE. VERIFY ERR.	-	-	点亮	点亮	继续运行	常时	
	OPE. MODE DIFF.	-	-	点亮	点亮	继续运行	常时	
6020	OPE. MODE DIFF.	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
6030	UNIT LAY. DIFF.	模块号	-	熄灭	闪烁	停止	电源接通 / 复位 / 连接热备电缆 / 运行模式改变时	
6035	UNIT LAY. DIFF.	-	-	熄灭	闪烁	停止		
6036	UNIT LAY. DIFF.	模块号	-	熄灭	闪烁	停止	常时	
6040	CARD TYPE DIFF.	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
6041		-	-	熄灭	闪烁	停止	电源接通 / 复位时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6001	在冗余系统中，由 DIP 开关设定的有效参数驱动器设置（SW2、SW3）不同。	将控制系统和待机系统的 DIP 开关设定的有效参数驱动器设置（SW2、SW3）设置为相同。	QnPRH
	6010	冗余系统中控制系统和待机系统的运行状态不相同。（只在冗余系统的待机系统中才可检测上述内容。）	使控制系统和待机系统的运行状态同步。	Q4AR
		冗余系统中控制系统和待机系统的运行状态不相同。（只在冗余系统的待机系统中才可检测上述内容。）	使控制系统和待机系统的运行状态同步。	QnPRH
	6020	电源接通 / 复位时，冗余系统的控制系统和待机系统的 RUN/STOP 开关设置不相同。（在冗余系统的控制系统或待机系统中可检测上述内容。）	将控制系统和待机系统的 RUN/STOP 开关设定为相同的设置。	
	6030	• 在冗余系统中，控制系统和待机系统的模块配置不同。 • 两个系统的网络模块模式设置不同。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	• 匹配控制系统和待机系统的模块配置。 • 在网络参数对话框的冗余设置中，匹配 B 系统和 A 系统的模式设置。	
	6035	在冗余系统中，控制系统和待机系统的 CPU 模块型号不同。（只在冗余系统的待机系统中才可检测上述内容。）	匹配控制系统和待机系统的模块型号。	
	6036	在冗余系统中，在控制系统和待机系统的 MELSECNET/H 多重远程 I/O 网络的远程 I/O 配置中检测到差异。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	检查 MELSECNET/H 多重远程 I/O 网络的网络电缆是否断开。	
	6040	在冗余系统中，控制系统和待机系统的存储卡安装状态（已安装 / 未安装）不同。	匹配控制系统和待机系统的存储卡安装状态（已安装 / 未安装）。	
	6041	在冗余系统中，控制系统和待机系统的存储卡类型不同。	匹配控制系统和待机系统的存储卡类型。	

*1: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6050	CAN'T EXE. MODE	-	-	点亮	点亮	继续运行	常时	
6060	CPU MODE DIFF.	-	-	熄灭	闪烁	停止	电源接通 / 复位 / 连接热备电缆时	
6061	CPU MODE DIFF.	-	-	熄灭	闪烁	停止	当执行 END 指令时	
6062	CPU MODE DIFF.	-	-	熄灭	闪烁	停止	电源接通 / 复位 / 连接热备电缆时	
6100	TRUCKINERR.	-	-	点亮	点亮	继续运行	电源接通 / 复位 /STOP → RUN 时	
	TRK. TRANS. ERR.	热备通信中通信数据的类型	-	点亮	点亮	继续运行	常时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6050	执行了在调试模式中或者运行模式（备用 / 独立模式）中不能执行的功能。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	执行可以在调试模式中或者运行模式（备用 / 独立模式）中可执行的功能。	QnPRH
	6060	在冗余系统中，控制系统和待机系统的运行模式（备用 / 独立）不同。 （只在冗余系统的待机系统中才可检测上述内容。）	匹配控制系统和待机系统的运行模式。	
	6061			
	6062	系统 A 和 B 处于相同的系统状态（控制系统）。（只在冗余系统的系统 B 中才可检测上述内容。）	将处于停止错误状态的 CPU 模块（系统 B）的电源关闭，然后再接通。	QnPRH
	6100	在初始化过程中，检测到 CPU 模块热备内存出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	CPU 模块的硬件故障。（请联系您当地的三菱服务中心或者代理商，说明故障症状。更换 CPU 模块时，应先更换待机系统的 CPU 模块，然后更换控制系统的 CPU 模块。）	Q4AR
		• 在热备数据传输过程中发生出错（例如，超过重试次数）。 （此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵循冗余系统启动步骤，在启动时发生错误。	• 在检查 CPU 模块或者热备电缆之后，进行再启动操作。如果出错仍然发生，这表示 CPU 模块或者热备电缆有故障。（请联系您当地的三菱服务中心或者代理商，说明故障症状。） • 确认冗余系统启动步骤，并重新执行启动。	QnPRH

*1: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
	TRUCKIN ERR.	-	-	点亮	点亮	继续运行	当执行 END 指令时	
6101	TRK. TRANS. ERR.	热备通信中通信数据的类型	-	点亮	点亮	继续运行	常时	
6102		-						
6103								
6105		热备通信中通信数据的类型						
6106								

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
		在热备的数据交换过程中，CPU 模块检测到出错。（在冗余系统的控制系统或待机系统中可检测上述内容。）	检查其它站的情况。	Q4AR
	6101	• 在热备（数据发送）过程中，发生超时错误。 （此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵守冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	• 检查 CPU 模块或者热备电缆。如果出错仍然发生，这表示 CPU 模块或者热备电缆有故障。 （请联系您当地的三菱 FA 中心或代理商，说明故障症状。） • 确认冗余系统启动步骤，并重新执行启动操作。	QnPRH
	6102	在热备（数据接收）过程中，发生数据总和出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）		
	6103	• 在热备（数据接收）过程中，发生数据出错（数据总和以外的情况）。 （此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵守冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）		
	6105	• 在热备（ 数据发送）过程中，发生出错（例如，超过重试次数）。 （此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵守冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）		
	6106	• 在热备（数据发送）过程中，发生超时出错。 （此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵守冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）		

*1: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6107	TRK. TRANS. ERR.	-	-	点亮	点亮	继续运行	常时	
6108								
6110	TRK. SIZE ERROR	热备容量超出出错因子	-	点亮	点亮	继续运行	当执行 END 指令时	
6111	TRK. SIZE ERROR	-	-	点亮	点亮	继续运行	当执行 END 指令时	
6112	TRK. SIZE ERROR	-	-	点亮	点亮	继续运行	当执行 END 指令时	
6120	TRK. CABLE ERR.	-	-	熄灭	闪烁	停止	电源接通 / 复位时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6107	在热备（数据接收）过程中，发生数据总和出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	• 检查 CPU 模块或者热备电缆。如果出错仍然发生，这表示 CPU 模块或者热备电缆有故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。） • 确认冗余系统启动步骤，并重新执行启动操作。	QnPRH
	6108	• 在热备（数据接收）过程中，发生数据出错（数据总和以外的情况）。（此出错也可能是由热备电缆松动或者其它系统断电（包括复位）引起的。） • 由于没有遵守冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）		
	6110	热备容量超出允许的范围。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	重新检查热备容量。	
	6111	控制系统没有足够的文件寄存器容量用于热备设置中指定的文件寄存器。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	切换为容量高于在热备设置中指定的文件寄存器容量的文件寄存器。	
	6112	从控制系统热备发送了超过待机系统文件寄存器容量的文件。（只在冗余系统的待机系统中才可检测上述内容。）	切换为容量高于在热备设置中指定的文件寄存器容量的文件寄存器。	
	6120	• 在没有连接热备电缆的情况下执行了启动操作。 • 在热备电缆有故障的情况下执行了启动操作。 • 由于 CPU 模块端的热备通信硬件有故障，所以不能通过热备电缆和其它系统进行通讯。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	在连接热备电缆之后再执行启动。如果相同的出错仍然发生，这表示热备电缆或者 CPU 模块端的热备通信硬件有故障。 （请联系附近的三菱 FA 中心或代理商，说明故障症状。）	

*1: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6130	TRK. DISCONNECT	-	-	点亮	点亮	继续运行	常时	
6140	TRK. INIT. ERROR	-	-	熄灭	闪烁	停止	电源接通 / 复位时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6130	• 热备电缆脱落。 • 在 CPU 运行过程中，热备电缆出现故障。 • CPU 模块端热备通信硬件出现故障。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	• 如果热备电缆脱落，连接热备电缆到两个系统的 CPU 模块的连接器。 • 如果将热备电缆连接到两个系统的 CPU 模块的连接器并进行出错解除后，出错仍然没有被消除，则表明热备电缆或者 CPU 模块端的热备通信硬件有故障。 （ 请联系您当地的三菱 FA 中心或代理商，说明故障症状。）	QnPRH
	6140	• 在电源上电 / 复位时，其它系统在初始化通讯过程中没有响应。 • 由于没有遵循冗余系统启动步骤，在启动时发生出错。 （在冗余系统的控制系统或待机系统中可检测上述内容。）	• 将相应 CPU 模块的电源关闭，然后再接通，或者复位 CPU 模块后再解除复位。如果相同的出错仍然发生，这表示 CPU 模块有故障。 （请联系您当地的三菱 FA 中心或代理商，说明故障症状。） • 确认冗余系统启动步骤，并重新执行启动。	

*1: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6200	CONTROL EXE.	系统切换原因	-	点亮	熄灭	继续运行	常时	
	CONTROL EXE.	系统切换原因	-	点亮	熄灭	无出错	常时	
6210	CONTROL WAIT	系统切换原因	-	点亮	熄灭	继续运行	常时	
	STANDBY	系统切换原因	-	点亮	熄灭	无出错	常时	

	出错代码(SD0)	出错内容及原因	对策	相应的 CPU
	6200	冗余系统中，待机系统被切换为控制系统。（只在冗余系统的待机系统中才可检测上述内容。）	检查控制系统状况。	Q4AR
		在冗余系统中，待机系统被切换为控制系统。（被从待机系统切换为控制系统的 CPU 检测到） 由于此出错代码不是 CPU 模块的异常信息，而是表示它的状态，所以出错代码和出错信息并没有被存储到 SD0 到 26，而是每次发生系统切换时被存储在出错历史记录中。 （使用 GX Developer 读取出错历史记录，确认出错信息。）	-	QnPRH
	6210	冗余系统中的控制系统被切换为待机系统。（只在冗余系统的待机系统中才可检测上述内容。）	检查控制系统状况。	Q4AR
		在冗余系统中，控制系统被切换为待机系统。（被从控制系统切换为待机系统的 CPU 检测到） 由于此出错代码不是 CPU 模块的异常信息，而是表示它的状态，所以出错代码和出错信息并没有被存储到 SD0 到 26，而是每次发生系统切换时被存储在出错历史记录中。 （使用 GX Developer 读取出错历史记录，确认出错信息。）	-	QnPRH

*1: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6220	CAN'T EXE. CHANGE	系统切换原因	-	点亮	点亮	继续运行	请求切换时	
	CAN'T SWITCH	系统切换原因	系统不能切换的原因	点亮	点亮	继续运行	请求切换时	
6221	CAN'T EXE. CHANGE	系统切换原因	-	点亮	点亮	继续运行	请求切换时	
6230	DUAL SYS. ERROR	-	-	点亮	点亮	继续运行	常时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6220	• 由于冗余系统中的待机系统处于出错等状态中，控制系统不能被切换为待机系统。 • 当试图执行系统切换时，由于控制系统的网络异常，控制系统未能被切换为待机系统。 （只在冗余系统的控制系统中才可检测上述内容。）	检查待机系统状况。	Q4AR
		在冗余系统中，由于待机系统异常 / 热备电缆异常 / 在独立模式下进行在线模块更换等，不能执行系统切换。下列给出了在控制系统上执行系统切换的原因。 • 根据 SP.CONTSW 指令的系统切换 • 来自网络模块的系统切换请求	• 检查待机系统的状态，并解除出错。 • 等待在线模块更换结束。	QnPRH
	6221	由于总线切换模块异常，不能执行切换。（只在冗余系统的控制系统中才可检测上述内容。）	这是总线切换模块硬件故障。（请联系附近的三菱 FA 中心或代理商，说明故障症状。）	Q4AR
	6230	安装在待机系统 CPU 中的链接模块是远程主站。	检查系统配置状态。	

*1: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6300	STANDBY SYS. DOWN	-	-	点亮	点亮	继续运行	常时	
6310	CONTROL SYS. DOWN	-	-	熄灭	闪烁	停止	常时	

	出错代码(SD0)	出错内容及原因	对策	相应的 CPU
	6300	在备份模式中，检测到下列出错情况中的一个。 ・在冗余系统中，待机系统没有启动。 ・在冗余系统中，待机系统发生停止出错。 ・处于调试模式的 CPU 模块被连接到运行中控制系统。 （只在冗余系统的控制系统中才可检测上述内容。）	・检查待机系统电源是否接通，如果未接通，通电使其运行。 ・检查待机系统是否处于复位状态，如果处于复位状态则进行复位解除。 ・检查待机系统是否有停止出错，如果其有出错，排除出错因素并重启系统。 ・当处于调试模式的 CPU 模块被连接到以备份模式运行的控制系统时，重新进行连接，使得控制系统和待机系统正确组合。	QnPRH
	6310	在备份模式中检测到下列出错情况中的一个。 ・在冗余系统中，控制系统没有启动。 ・在冗余系统中，控制系统发生停止出错。 ・处于调试模式的 CPU 模块被连接到运行的待机系统。 ・由于没有遵循冗余系统启动步骤，在启动时发生出错。 （只在冗余系统的待机系统中才可检测上述内容。）	・待机系统存在，但是控制系统不存在。 ・检查待机系统以外的系统的电源是否接通，如果未接通，则接通电源。 ・检查待机系统以外系统是否处于复位状态，如果处于复位状态，则进行复位解除。 ・检查待机系统以外的系统是否发生停止出错，如果有出错发生，排除出错因素，将控制系统和待机系统设定为相同的运行状态，然后重启系统。 ・当处于调试模式的 CPU 模块被连接到运行于备份模式的待机系统时，重新进行连接，使得控制系统和待机系统正确组合。 ・确认冗余系统启动步骤，并重新执行启动。	QnPRH

*1: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
6311	CONTROL SYS. DOWN	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
6312								
6313	CONTROL SYS. DOWN	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
6400	PRG. MEM. CLEAR	-	-	熄灭	闪烁	停止	执行从控制系统到待机系统的内存复制时	
6410	MEM.COPY EXE	-	-	点亮	点亮	继续运行	执行从控制系统到待机系统的内存复制时	
6500	TRK. PARA. ERROR	文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位时	
6501		文件名称 / 驱动器名称	参数号	熄灭	闪烁	停止	电源接通 / 复位时	

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	6311	・在冗余系统中，由于没有从控制系统发送一致性检查数据，所以其它系统不能作为待机系统启动。 ・由于没有遵守冗余系统启动步骤，在启动时发生出错。 （只在冗余系统的待机系统中才可检测上述内容。）	・更换热备电缆。如果相同的出错仍然发生，这表示 CPU 模块有故障。（ 请联系您当地的三菱 FA 中心或代理商，说明故障症状。） ・确认冗余系统启动步骤，并重新执行启动操作。	QnPRH
	6312			
	6313	在冗余系统中，控制系统检测到系统配置异常，并已通知待机系统（本系统）。	确认基板间连接、系统配置（类型 / 个数 / 参数）正常后，重新启动系统。	QnPRH*1
	6400	执行了从控制系统到待机系统的内存复制，程序内存被清除。	在从控制系统到待机系统的内存复制完成后，将电源切换为 OFF，然后切换为 ON，或者进行复位。	QnPRH
	6410	执行了从控制系统到待机系统的内存复制。（只在冗余系统的控制系统中才可检测上述内容。）	－	
	6500	在可编程控制器参数对话框的热备设置中指定的文件寄存器文件不存在。	使用 GX Developer 读取出错的个别信息，并检查和改正驱动器名称和文件名称。创建指定的文件。	QnPRH
	6501	在可编程控制器参数对话框的热备设置的软元件详细设置中指定的文件寄存器范围超出了指定的文件寄存器文件容量。	使用 GX Developer 读取出错的个别信息，并增加文件寄存器容量。	

*1: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

12.1.9 出错代码表 (7000 ～ 7999)

以下介绍出错代码 7000 ～ 7999 的出错信息、异常内容及原因以及处理方法等。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
7000	MULTI CPU DOWN	模块号 (CPU 号)	-	熄灭	闪烁	停止	常时	
7002	MULTI CPU DOWN	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7003								
7004	MULTI CPU DOWN	模块号 (CPU 号)	-	熄灭	闪烁	停止	常时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2: 在基本模式 QCPU 中，可以通过 LED 显示优先顺序设置用特殊寄存器 (SD207 ～ DS209) 进行熄灯。
(高性能模式 QCPU、过程 CPU、冗余 CPU 和通用型 QCPU 中只能为熄灯状态)

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	7000	• 在多 CPU 系统中，在运行模式为“All station stop by stop error of PLC”的 CPU 模块中发生了异常。 • 在多 CPU 系统中，安装了与多 CPU 系统不兼容的 CPU 模块。	• 使用 GX Developer 读取出错的个别信息，确认引起 CPU 模块异常的出错，并排除出错原因。 • 从主基板上卸载与多 CPU 系统不兼容的 CPU 模块。	Q00/Q01*1 Qn(H)*1 QnPH QnU
		在多 CPU 系统中，电源 ON 时由于 1 号 CPU 停止出错，导致其它 CPU 无法启动 (2 号到 4 号 CPU 中发生)。	使用 GX Developer 读取出错的个别信息，确认引起 CPU 模块异常的出错，并排除出错原因。	Q00/Q01*1 Qn(H)*1 QnPH QnU
	7002	• 多 CPU 系统中，在初始化通讯阶段，目标 CPU 模块没有响应。 • 在多 CPU 系统中，安装了不兼容多 CPU 系统的 CPU 模块。	• 复位 CPU 模块，然后重新运行。如果相同的出错再次显示，这表明 CPU 模块中的一个有硬件故障。(请联系附近的三菱 FA 中心或代理商，说明故障症状。) • 从主基板卸载不兼容多 CPU 系统的 CPU 模块。 或者将不兼容多 CPU 系统的 CPU 模块更换为兼容多 CPU 系统的 CPU 模块。	Q00/Q01*1 Qn(H)*1 QnPH
		多 CPU 系统中，在初始化通讯阶段，目标 CPU 模块没有响应。	复位 CPU 模块，然后重新运行。如果相同的出错再次显示，这表明 CPU 模块中的一个有硬件故障。(请联系附近的三菱 FA 中心或代理商，说明故障症状。)	QnU
	7003	多 CPU 系统中，在初始化通讯阶段，目标 CPU 模块没有响应。	复位 CPU 模块，然后重新运行。如果相同的出错再次显示，这表明 CPU 模块有硬件故障。(请联系附近的三菱 FA 中心或代理商，说明故障症状。)	Q00/Q01*1 Qn(H)*1 QnPH
	7004	在多 CPU 系统中，在 CPU 模块间发生了数据通讯错误。	• 检查系统配置，确认模块是否安装在超过 I/O 点数的插槽中。 • 当系统配置没有问题时，这表示 CPU 模块有硬件故障。(请联系当地的三菱 FA 中心或代理商，说明故障症状。)	Q00/Q01*1 QnU

*3: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
7010	MULTI EXE. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7011	MULTI EXE. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7013	MULTI EXE. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7020	MULTI CPU ERROR	模块号 (CPU 号)	-	点亮	点亮	继续运行	常时	
7030	CPU LAY. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7031								

*1: 以功能版本 B 以后的 CPU 为对象。

*2: 在基本模式 QCPU 中，可以通过 LED 显示优先顺序设置用特殊寄存器 (SD207 ～ DS209) 进行熄灯。
(高性能模式 QCPU、过程 CPU、冗余 CPU 和通用型 QCPU 中只能为熄灯状态)

	出错代码(SD0)	出错内容及原因	对策	相应的 CPU
	7010	• 在多 CPU 系统中，安装了有故障的 CPU 模块。 • 在多 CPU 系统中，安装了不兼容多 CPU 系统的 CPU 模块。(通过与多 CPU 系统兼容的 CPU 模块检测出错。) • 在多 CPU 系统中，在电源 ON 的状态下对 2 ～ 4 号 CPU 进行了复位。(通过复位解除的 CPU 检测出错。) • 在多 CPU 系统中，个人计算机 CPU 模块使用了版本为 1.06 或者更低的 QFB(总线接口驱动器)。	• 使用 GX Developer 读取出错的个别信息，并更换有故障的 CPU 模块。 • 将不兼容多 CPU 系统的 CPU 模块更换为兼容多 CPU 系统的 CPU 模块。 • 不要复位 2 ～ 4 号 CPU 模块中的任何一个。 • 使用 QFB 版本为 1.07 以后的个人计算机 CPU 模块。 • 复位 1 号 CPU 模块，并重启多 CPU 系统。	Q00/Q01*1 Qn(H)*1 QnPH QnU
	7011	在多 CPU 系统中，设定了下列设置中的一个。 • 对不能进行多 CPU 间自动刷新的 CPU 模块设定了多 CPU 自动刷新设置。 • 为不可进行以下设置的的 CPU 模块进行了 “I/O sharing when use multiple CPUs” 设置。	• 改正多 CPU 自动刷新设置。 • 改正 “I/O sharing when use multiple CPUs” 设置。	Q00/Q01*1 QnU
		系统配置未符合多 CPU 高速通信功能。 • 1 号 CPU 未使用 QnUD(H)CPU。 • 未使用多 CPU 高速主基板 (Q3 □ DB)。 • 对不兼容多 CPU 高速通信功能的 CPU 模块，通信范围设置到 0 点以外。 • 对不兼容多 CPU 的 CPU 模块，通信范围设置到 0 点以外。	将系统配置变更为满足多 CPU 高速通信功能的使用条件。	QnU*3
	7013	CPU 插槽或插槽 0 到 2 安装了 Q172(H)CPU(N) 或 Q173(H)CPU(N)。 (有可以导致模块故障)	卸载 Q172(H)CPU(N) 或 Q173(H)CPU(N)。	QnU
	7020	在多 CPU 系统中，在处于 “where system stop was not selected” 的运行模式的 CPU 上发生了出错。 (通过未发生出错的 CPU 模块检测出出错。)	使用 GX Developer 读取出错的个别信息，确认引起 CPU 模块异常的出错，并排除出错原因。	Q00/Q01*1 Qn(H)*1 QnPH QnU
	7030	在超出了可编程控制器参数对话框的多 CPU 设置中指定的 CPU 模块个数的 CPU 安装插槽 (CPU 插槽，I/O 插槽 0、1) 中发生了分配出错。	• 使可编程控制器参数对话框的多 CPU 设置中指定的 CPU 模块个数符合实际安装的 CPU 模块 (包括 CPU(空闲)) 的个数。 • 使在可编程控制器参数对话框的 I/O 分配设置中指定的类型和 CPU 模块配置相一致。	Q00J/Q01/Q01*1 QnU
	7031	在可编程控制器参数对话框的多 CPU 设置中指定的 CPU 个数未被超出的情况下，发生了分配出错。		

*3: 除 Q02UCPU 外的通用型 QCPU。

出错代码 (SD0)	出错信息	公共信息 (SD5 ～ 15)	个别信息 (SD16 ～ 26)	LED 状态		CPU 运行状态	检测时间	
				RUN	ERROR			
7032	CPU LAY. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7035	CPU LAY. ERROR	模块号 (插槽号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
7036	CPU RAY. ERROR	模块号 (CPU 号)	-	熄灭	闪烁	停止	电源接通 / 复位时	
8031	INCORRECT FILE	-	文件诊断信息	熄灭	熄灭	停止	电源接通 / 复位 /STOP → RUN/ 可编程控制器写入时	
9000	F****	程序出错位置	报警器号	点亮	点亮 / 熄灭 *2	继续运行	当执行指令时	
				USER LED 点亮				
9010	<CHK>ERR ***-***	程序出错位置	报警器号	点亮	熄灭	继续运行	当执行指令时	
				USER LED 点亮				
9020	BOOT OK	-	-	熄灭	闪烁	停止	电源接通 / 复位时	
10000	CONT. UNIT ERROR	-	-	熄灭	闪烁	继续运行	常时	

*1: 以功能版本 B 以后的 CPU 为对象。

*2: 在基本模式 QCPU 中，可以通过 LED 显示优先顺序设置用特殊寄存器 (SD207 ～ DS209) 进行熄灯。
(高性能模式 QCPU、过程 CPU、冗余 CPU 和通用型 QCPU 中只能为熄灯状态)

	出错代码 (SD0)	出错内容及原因	对策	相应的 CPU
	7032	在多 CPU 系统中安装的 CPU 模块个数有错误。	将系统配置为各 CPU 的模块安装个数不超过规格中规定的最大可安装模块数。	Q00/Q01*1 QnU
	7035	CPU 模块被安装在不可用的插槽中。	将 CPU 模块安装到可用插槽中。	Q00J/Q00/Q01*1 QnPRH QnU
	7036	多 CPU 设置中设置的自 CPU 号与 CPU 模块安装位置确定的自 CPU 号不一致。	• 将 CPU 模块安装到正确的插槽。 • 将多 CPU 设置的自 CPU 号设置修改为 CPU 模块安装位置确定的自 CPU 号。	QnU*3
	8031	检测出存储文件（有效的参数文件）异常。	将个别信息的 SD17 ～ SD22 显示的文件写入到个别信息的 SD16(L) 显示的驱动器中后，进行电源 OFF → ON 或者复位→复位解除。如果相同的出错再次显示，可能是 CPU 模块硬件故障。（联系附近的三菱 FA 中心或代理商，说明故障症状。）	QnU
	9000	报警器 (F) 为 ON。	使用外围设备读取出错的个别信息，并检查对应数值（报警器号）的程序。	○
	9010	CHK 指令检测到出错。	使用外围设备读取出错的个别信息，并检查对应数值（出错编号）的程序。	QnA Qn(H) QnPH QnPRH
	9020	在自动写入标准 ROM 的过程中，到 ROM 的数据存储正常结束。 (BOOT LED 也闪烁。)	使用 DIP 开关将有效参数驱动器设置为标准 ROM。然后，重新将电源接通，执行从标准 ROM 的引导运行。	Qn(H)*1 QnPH QnPRH
	10000	在多 CPU 系统中，在过程 CPU/ 高性能型 QCPU 以外的 CPU 模块中发生出错。	使用 GX Developer 连接到相应的 CPU 模块，确认产生的出错的详细信息。	Qn(H)*1 QnPH

*3: 除 Q02UCPU 外的通用型 QCPU。

12.2 出错的解除

对于 Q 系列 CPU 模块，只有当出错为继续运行出错时，才可以进行出错解除操作。
出错解除应遵循如下所示的步骤。

1) 排除出错的原因。
2) 将要被解除的出错代码存储到特殊寄存器 SD50 中。
3) 对特殊继电器 SM50 进行 OFF → ON 操作。
4) 解除要解除的出错。

通过出错解除使 CPU 模块复位后，与出错相关的特殊继电器、特殊寄存器和 LED 将返回到出错发生之前的状态。
如果在出错解除后相同的出错再次发生，它将被重新登记到出错历史中。

当对检测出的多个报警器 (F) 进行解除时，只有最先被检测出的报警号的报警器被解除。

关于出错解除的详细信息，请参考下面的手册。
→ QCPU 用户手册（功能解说 / 程序基础篇）

要 点

(1) 将要解除的出错代码存储到 SD50 中后进行了出错解除的情况下，下一位的出错代码编号可以被忽略。
（实例）
如果出错代码 2100 和 2101 发生，则出错代码 2100 被解除时出错代码 2101 也将被解除。
如果出错代码 2100 和 2111 发生，当出错代码 2100 被解除时，出错代码 2111 不会被解除。

(2) 对于 CPU 模块以外原因引起的出错，即使通过特殊继电器 (SM50) 和特殊寄存器 (SD50) 进行了出错解除，也不能消除出错原因。
（实例）
由于 “SP.UNIT DOWN” 是发生在基板（包括扩展电缆）、智能功能模块等元件上的出错，因此即使通过特殊继电器 (SM50) 和特殊寄存器 (SD50) 进行了出错解除，也不能消除出错原因。
应参照出错代码表，消除出错原因。

附 附录

附录 1　操作处理时间

附录 1.1　定义

(1) 在 Q/QnACPU 中，操作处理时间是下列时间的总和：

- 每个指令处理时间的总和
- END 处理时间
- I/O 刷新时间

(2) 指令处理时间

这是列在附录 1.2 和附录 1.3 中的每条指令处理时间的总和

(3) END 处理时间

END 处理时间是下列时间的总和：

- 在附录 1.2 和附录 1.3 中指出的 END 指令时间
- MELSECNET 相关的刷新时间 *1
- 与外围设备间的通讯处理时间 *2
- 与如串行通讯模块等设备的通讯时间 *3

*1:　关于 MELSECNET 相关的刷新时间，请参照以下手册：
- MELSECNET/G 网络系统参考手册（控制网络篇）
- Q 系列 MELSECNET/H 网络系统参考手册（可编程控制器网络篇）
- QnA/Q4AR 系列 MELSECNET/10 网络系统参考手册
- MELSECNET、MELSECNET/B 数据链接系统参考手册

*2:　关于与外围设备的通讯处理时间，请参考以下手册：
- QCPU 用户手册（功能解说 / 程序基础篇）
- Q2A (S1)/Q3A/Q4ACPU 用户手册（详细篇）
- Q2AS(H)CPU(S1) 用户手册（详细篇）

*3:　关于与串行通讯模块的通讯处理时间，请参照相关模块的用户手册。

(4) I/O 刷新时间

I/O 刷新时间 = (输入点数 /16) × N1 + (输出点数 /16) × N2

对于 N1 和 N2，请参考下表。

(a) QCPU

CPU 模块	N1					N2				
	Q3□B、Q3□SB、Q3□RB、Q3□DB	Q5□B、Q6□B、Q6□RB	Q6□WRB	QA1S6□B	QA6□B	Q3□B、Q3□SB、Q3□RB、Q3□DB	Q5□B、Q6□B、Q6□RB	Q6□WRB	QA1S6□B	QA6□B
Q00JCPU	2.05μs	2.95μs	--	--	--	1.25μs	2.20μs	--	--	--
Q00CPU	2.00μs	2.75μs	--	--	--	1.20μs	2.05μs	--	--	--
Q01CPU	1.95μs	2.70μs	--	--	--	1.15μs	2.00μs	--	--	--
Q02CPU	2.2μs	2.9μs	--	4.3μs	4.3μs	1.3μs	2.1μs	--	3.5μs	3.5μs
Q02HCPU Q06HCPU Q12HCPU Q25HCPU	1.7μs	2.4μs	--	3.7μs	3.7μs	1.3μs	2.1μs	--	3.5μs	3.5μs
Q12PHCPU Q25PHCPU	1.7μs	2.4μs	--	--	--	1.3μs	2.1μs	--	--	--
Q12PRHCPU Q25PRHCPU	1.7μs	--	2.4μs	--	--	1.3μs	--	2.1μs	--	--
Q02UCPU	1.5μs	2.3μs	--	--	--	1.0μs	1.8μs	--	--	--
Q03UDCPU Q04UDHCPU Q06UDHCPU	1.5μs	2.3μs	--	--	--	1.0μs	1.8μs	--	--	--

(b) QnA(R)CPU

CPU 模块	N1	N2
Q2ASCPU(S1) Q2ACPU	5.2μs	5.0μs
Q3ACPU	4.8μs	4.7μs
Q2ASHCPU(S1) Q4ACPU Q4ARCPU	4.4μs	4.3μs

附

附录

附录 1.2 基本型 QCPU 的操作处理时间

每条指令的处理时间列在下表中。

指令的处理时间会由于指令源和目标的种类不同而产生一定的变化，因此包含在下表中的值应作为处理时间的一套基本准则，而不能作为严格准确的时间来计算。

(1) 顺控指令

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
LD LDI AND ANI OR ORI	X0	0.20	0.16	0.10
	D0.0	0.30	0.24	0.15
LDP LDF ANDP ANDF ORP ORF	X0	0.30	0.24	0.15
	D0.0			
ANB ORB MPS MRD MPP	--	0.20	0.16	0.10
INV	当不被执行时	0.20	0.16	0.10
	当被执行时			
MEP MEF	当不被执行时	0.30	0.24	0.15
	当被执行时			
EGP	当不被执行时 (OFF→OFF) (ON→ON)	0.20	0.16	0.10
	当被执行时 (OFF→ON) (ON→OFF)			
EGF	当不被执行时 (OFF→OFF) (ON→ON)	17	9.5	9.4
	当被执行时 (OFF→ON) (ON→OFF)	18	14	14

指令		条件（软元件）			处理时间（μs）		
					Q00JCPU	Q00CPU	Q01CPU
OUT	Y	无变化时	(OFF→OFF) (ON→ON)		0.20	0.16	0.10
		变化时	(OFF→ON) (ON→OFF)		0.20	0.16	0.10
	D0.0	无变化时	(OFF→OFF) (ON→ON)		0.40	0.32	0.20
		变化时	(OFF→ON) (ON→OFF)		0.40	0.32	0.20
	F	OFF 时			24	20	19
		ON 时	显示时		260	210	200
			显示完成时		205	165	155
	T	当不被执行时			1.1	0.88	0.55
		当被执行时	时间到后		1.1	0.88	0.55
			加法运算时	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
	C	当不被执行时			1.1	0.88	0.55
		当被执行时	时间到后		1.1	0.88	0.55
			加法运算时	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
OUTH	T	当不被执行时			1.1	0.88	0.55
		当被执行时	时间到后		1.1	0.88	0.55
			加法运算时	K	1.1	0.88	0.55
				D	1.2	0.96	0.60
SET	Y	当不被执行时			0.20	0.16	0.10
		当被执行时	无变化时 (ON→ON)		0.20	0.16	0.10
			变化时 (OFF→ON)		0.20	0.16	0.10
	D0.0	当不被执行时			0.40	0.32	0.20
		当被执行时	无变化时 (ON→ON)		0.40	0.32	0.20
			变化时 (OFF→ON)		0.40	0.32	0.20
	F	当不被执行时			0.50	0.44	0.25
		当被执行时	显示时		255	205	195
			显示完成时		195	160	150
RST	Y	当不被执行时			0.20	0.16	0.10
		当被执行时	无变化时 (OFF→OFF)		0.20	0.16	0.10
			变化时 (ON→OFF)		0.20	0.16	0.10
	D0.0	当不被执行时			0.40	0.32	0.20
		当被执行时	无变化时 (ON→ON)		0.40	0.32	0.20
			变化时 (OFF→ON)		0.40	0.32	0.20
	SM	当不被执行时			0.20	0.16	0.10
		当被执行时			0.20	0.16	0.10
	F	当不被执行时			0.48	0.44	0.25
		当被执行时	显示时		75	69	65
			显示完成时		43	35	33
	T、C	当不被执行时			0.80	0.64	0.40
		当被执行时			1.0	0.80	0.50
	D	当不被执行时			0.40	0.32	0.20
		当被执行时			0.60	0.48	0.30
	Z	当不被执行时			0.50	0.40	0.25
		当被执行时			9.4	7.9	7.4
	R	当不被执行时			--	0.32	0.20
		当被执行时			行	0.48	0.30

指令		条件（软元件）	处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
PLS			12	9.5	9.2
PLF			11	9.5	8.9
FF	Y	当不被执行时	0.68	0.40	0.25
		当被执行时	7.5	6.2	5.7
DELTA	DY0	当不被执行时	0.50	0.40	0.25
		当被执行时	26	21	21
DELTAP	DY0	当不被执行时	0.48	0.40	0.25
		当被执行时	58	45	43
SFT SFTP		当不被执行时	0.50	0.34	0.25
		当被执行时	12	8.7	8.3
MC		M0	0.40	0.32	0.20
		D0.0	3.3	2.9	2.8
MCR		--	0.20	0.16	0.10
FEND END		执行错误检查时	660	600	520
		不执行错误检查时 （• 电池检查） （• 保险丝熔断检查） （• I/O 模块校验）	660	600	520
NOP		--	0.20	0.16	0.10
NOPLF PAGE		--	0.20	0.16	0.10

(2) 基本指令

当指令不被执行时，处理时间按如下方式计算：

Q00JCPU ... 0.20 × （每条指令的步数 + 1)μs

Q00CPU ... 0.16 × （每条指令的步数 + 1)μs

Q01CPU ... 0.10 × （每条指令的步数 + 1)μs

指令	条件（软元件）		处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
LD =	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
OR =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LD < >	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND < >	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40

指令	条件（软元件）		处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
OR < >	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LD >	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND >	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
OR >	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LD < =	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND < =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
OR < =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LD <	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND <	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
OR <	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LD > =	导通时		0.80	0.64	0.40
	不导通时		0.80	0.64	0.40
AND > =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
OR > =	当不被执行时		0.70	0.56	0.35
	当被执行时	导通时	0.80	0.64	0.40
		不导通时	0.80	0.64	0.40
LDD =	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
ORD =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
LDD < >	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD < >	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50

指令	条件（软元件）		处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
ORD < >	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
LDD >	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD >	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
ORD >	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
LDD < =	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD < =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
ORD < =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
LDD <	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD <	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
ORD <	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
LDD > =	导通时		1.0	0.80	0.50
	不导通时		1.0	0.80	0.50
ANDD > =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
ORD > =	当不被执行时		0.80	0.64	0.40
	当被执行时	导通时	1.0	0.80	0.50
		不导通时	1.0	0.80	0.50
BKCMP = (S1) (S2) (D) n	n = 1		130	105	97
BKCMP = P (S1) (S2) (D) n	n = 96		205	175	165
BKCMP<> (S1) (S2) (D) n	n = 1		130	105	98
BKCMP<>P (S1) (S2) (D) n	n = 96		210	180	165
BKCMP> (S1) (S2) (D) n	n = 1		130	105	97
BKCMP>P (S1) (S2) (D) n	n = 96		210	180	165
BKCMP>= (S1) (S2) (D) n	n = 1		130	105	98
BKCMP>=P (S1) (S2) (D) n	n = 96		205	175	165

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
BKCMP< (S1) (S2) (D) n	n = 1	130	105	98
BKCMP<P (S1) (S2) (D) n	n = 96	210	180	165
BKCMP<= (S1) (S2) (D) n	n = 1	130	105	97
BKCMP<=P (S1) (S2) (D) n	n = 96	205	175	165
+ (S) (D) +P (S) (D)	当被执行时	1.0	0.80	0.50
+ (S1) (S2) (D) +P (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
- (S) (D) - P (S) (D)	当被执行时	1.0	0.80	0.50
- (S1) (S2) (D) - P (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
D+ (S) (D) D+P (S) (D)	当被执行时	1.3	1.04	0.65
D+ (S1) (S2) (D) D+P (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
D - (S) (D) D - P (S) (D)	当被执行时	1.3	1.04	0.65
D - (S1) (S2) (D) D - P (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
* (S1) (S2) (D) * P (S1) (S2) (D)	当被执行时	1.1	0.88	0.55
/ (S1) (S2) (D) /P (S1) (S2) (D)	--	19	16	15
D * (S1) (S2) (D) D * P (S1) (S2) (D)	--	41	34	31
D/ (S1) (S2) (D) D/P (S1) (S2) (D)	--	28	23	21
B+ (S) (D) B+P (S) (D)	--	34	28	26
B+ (S1) (S2) (D) B+P (S1) (S2) (D)	--	47	39	37
B - (S) (D) B - P (S) (D)	--	34	28	26
B - (S1) (S2) (D) B - P (S1) (S2) (D)	--	48	40	38
DB+ (S) (D) DB+P (S) (D)	--	58	48	44
DB+ (S1) (S2) (D) DB+P (S1) (S2) (D)	--	60	49	46
DB - (S) (D) DB - P (S) (D)	--	59	48	45
DB - (S1) (S2) (D) DB - P (S1) (S2) (D)	--	60	51	45

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
B ＊ (S1) (S2) (D) B ＊ P (S1) (S2) (D)	--	42	35	33
B/ (S1) (S2) (D) B/P (S1) (S2) (D)	--	48	40	37
DB ＊ (S1) (S2) (D) DB ＊ P (S1) (S2) (D)	--	140	120	110
DB/ (S1) (S2) (D) DB/P (S1) (S2) (D)	--	83	69	65
BK + (S1) (S2) (D) n	n = 1	105	86	80
BK + P (S1) (S2) (D) n	n = 96	185	155	140
BK - (S1) (S2) (D) n	n = 1	105	86	80
BK - P (S1) (S2) (D) n	n = 96	185	155	140
INC INCP	--	0.70	0.56	0.35
DINC DINCP	--	0.90	0.72	0.45
DEC DECP	--	0.70	0.56	0.35
DDEC DDECP	--	0.90	0.72	0.45
BCD BCDP	--	20	16	15
DBCD DBCDP	--	26	21	20
BIN BINP	--	19	16	15
DBIN DBINP	--	22	18	17
DBL DBLP	--	19	16	15
WORD WORDP	--	23	19	17
GRY GRYP	--	19	16	15
DGRY DGRYP	--	23	19	17
GBIN GBINP	--	52	42	40
DGBIN DGBINP	--	110	88	84
NEG NEGP	--	16	13	12
DNEG DNEGP	--	19	17	15

指令		条件（软元件）	处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
BKBCD (S) (D) n		n = 1	78	63	57
BKBCDP (S) (D) n		n = 96	315	275	250
BKBIN (S) (D) n		n = 1	74	61	57
BKBINP (S) (D) n		n = 96	285	255	230
MOV MOVP		(S) = D0、(D) = D1	0.70	0.56	0.35
		(S) = D0、(D) = J1 \ W1	155	130	120
DMOV DMOVP		(S) = D0、(D) = D1	0.90	0.72	0.45
		(S) = D0、(D) = J1 \ W1	165	135	120
$MOV		0 字	46	38	35
$MOVP		32 字	98	80	73
CML CMLP		--	0.70	0.56	0.35
DCML DCMLP		--	0.90	0.72	0.45
BMOV (S) (D) n		n = 1	27	21	20
BMOVP (S) (D) n		n = 96	72	62	53
FMOV (S) (D) n		n = 1	23	19	17
FMOVP (S) (D) n		n = 96	48	41	36
XCH XCHP		--	7.6	6.3	5.7
DXCH DXCHP		--	9.5	8.0	7.1
BXCH (D1) (D2) n		n = 1	62	51	48
BXCHP (D1) (D2) n		n = 96	165	140	125
SWAP SWAPP		--	17	14	13
CJ		--	10	8.5	8.1
SCJ		--	10	8.5	8.1
JMP		--	11	8.5	8.1
GOEND		--	3.3	2.9	2.8
DI		--	13	12	11
EI		--	14	11	11
IMASK		--	41	34	35
IRET		--	205	170	155
RFS RFSP	X	n = 1	55	46	43
		n = 96	79	64	59
	Y	n = 1	54	45	41
		n = 96	73	61	56

(3) 应用指令

当指令不执行时，处理时间按如下方式计算：

Q00JCPU .. 0.20 × (每条指令的步数 + 1)μs

Q00CPU ... 0.16 × (每条指令的步数 + 1)μs

Q01CPU ... 0.10 × (每条指令的步数 + 1)μs

指令	条件（软元件）	处理时间 (μs)		
		Q00JCPU	Q00CPU	Q01CPU
WAND (S) (D) WANDP (S) (D)	当被执行时	1.0	0.80	0.50
WAND (S1) (S2) (D) WANDP (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
DAND (S) (D) DANDP (S) (D)	当被执行时	1.3	1.04	0.65
DAND (S1) (S2) (D) DANDP (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
BKAND (S1) (S2) (D) n	n = 1	110	87	79
BKANDP (S1) (S2) (D) n	n = 96	185	155	140
WOR (S) (D) WORP (S) (D)	当被执行时	1.0	0.80	0.50
WOR (S1) (S2) (D) WORP (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
DOR (S) (D) DORP (S) (D)	当被执行时	1.3	1.04	0.65
DOR (S1) (S2) (D) DORP (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
BKOR (S1) (S2) (D) n	n = 1	110	87	81
BKORP (S1) (S2) (D) n	n = 96	185	155	140
WXOR (S) (D) WXORP (S) (D)	当被执行时	1.0	0.80	0.50
WXOR (S1) (S2) (D) WXORP (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
DXOR (S) (D) DXORP (S) (D)	当被执行时	1.3	1.04	0.65
DXOR (S1) (S2) (D) DXORP (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
BKXOR (S1) (S2) (D) n	n = 1	110	87	81
BKXORP (S1) (S2) (D) n	n = 96	185	155	140

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
WXNR (S) (D) WXNRP (S) (D)	当被执行时	1.0	0.80	0.50
WXNR (S1) (S2) (D) WXNRP (S1) (S2) (D)	当被执行时	1.2	0.96	0.60
DXNR (S) (D) DXNRP (S) (D)	当被执行时	1.3	1.04	0.65
DXNR (S1) (S2) (D) DXNRP (S1) (S2) (D)	当被执行时	1.5	1.2	0.75
BKXNR (S1) (S2) (D) n	n = 1	110	87	82
BKXNRP (S1) (S2) (D) n	n = 96	185	155	140
ROR (D) n	n = 1	13	11	9.7
RORP (D) n	n = 15	13	11	9.7
RCR (D) n	n = 1	15	12	12
RCRP (D) n	n = 15	15	13	12
ROL (D) n	n = 1	13	11	10
ROLP (D) n	n = 15	13	11	10
RCL (D) n	n = 1	15	13	12
RCLP (D) n	n = 15	16	13	12
DROR (D) n	n = 1	15	12	12
DRORP (D) n	n = 31	15	13	12
DRCR (D) n	n = 1	17	14	14
DRCRP (D) n	n = 31	18	16	15
DROL (D) n	n = 1	14	13	12
DROLP (D) n	n = 31	14	13	12
DRCL (D) n	n = 1	18	15	14
DRCLP (D) n	n = 31	20	17	16
SFR (D) n	n = 1	13	10	9.7
SFRP (D) n	n = 15	13	11	9.5
SFL (D) n	n = 1	12	10	9.5
SFLP (D) n	n = 15	12	9.8	9.5
BSFLR (D) n	n = 1	42	35	33
BSFLRP (D) n	n = 96	69	58	54
BSFL (D) n	n = 1	41	34	32
BSFLP (D) n	n = 96	63	53	50
DSFR (D) n	n = 1	19	16	15
DSFRP (D) n	n = 96	71	61	53
DSFL (D) n	n = 1	19	16	15
DSFLP (D) n	n = 96	70	60	52

附

附录

指令	条件（软元件）		处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
BSET (D) n	n = 1		27	22	20
BSETP (D) n	n = 15		27	22	20
BRST (D) n	n = 1		27	22	21
BRSTP (D) n	n = 15		27	22	21
TEST (S1) (S2) (D) TESTP (S1) (S2) (D)	--		35	30	27
DTEST (S1) (S2) (D) DTESTP (S1) (S2) (D)	--		37	31	28
BKRST (S) n	n = 1		49	41	38
BKRSTP (S) n	n = 96		64	54	50
SER (S1) (S2) (D) n SERP (S1) (S2) (D) n	n = 1	所有的都匹配	56	54	42
		一个都不匹配	56	54	42
	n = 96	所有的都匹配	280	240	220
		一个都不匹配	280	240	220
DSER (S1) (S2) (D) n DSERP (S1) (S2) (D) n	n = 1	所有的都匹配	71	67	53
		一个都不匹配	71	67	54
	n = 96	所有的都匹配	495	415	375
		一个都不匹配	500	415	375
SUM SUMP	(S) = 0		32	26	25
	(S) = $FFFF_H$		27	22	21
DSUM DSUMP	(S) = 0		54	44	42
	(S) = $FFFFFFFF_H$		54	44	42
DECO (S) (D) n	n = 2		60	50	46
DECOP (S) (D) n	n = 8		80	65	61
ENCO (S) (D) n ENCOP (S) (D) n	n = 2	M1 = ON	66	55	51
		M4 = ON	66	54	51
	n = 8	M1 = ON	90	76	71
		M256 = ON	76	74	71
SEG SEGP	--		8.0	6.8	6.1
DIS (S) (D) n	n = 1		47	39	36
DISP (S) (D) n	n = 4		53	43	40
UNI (S) (D) n	n = 1		54	44	41
UNIP (S) (D) n	n = 4		60	49	46
NDIS (S1) (D) (S2) NDISP (S1) (D) (S2)	--		92	76	38
NUNI (S1) (D) (S2) NUNIP (S1) (D) (S2)	--		47	39	36

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
WTOB (S) (D) n	n = 1	56	46	42
WTOBP (S) (D) n	n = 96	190	155	145
BTOW (S) (D) n	n = 1	56	46	42
BTOWP (S) (D) n	n = 96	190	155	145
MAX (S) (D) n	n = 1	48	40	36
MAXP (S) (D) n	n = 96	300	240	235
MIN (S) (D) n	n = 1	48	40	36
MINP (S) (D) n	n = 96	300	240	235
DMAX (S) (D) n	n = 1	52	43	39
DMAXP (S) (D) n	n = 96	600	490	460
DMIN (S) (D) n	n = 1	52	43	39
DMINP (S) (D) n	n = 96	585	475	445
SORT (S1) n (S2) (D1) (D2)	n = 1	66	55	50
	n = 96	105	86	80
DSORT (S1) n (S2) (D1) (D2)	n = 1	98	57	52
	n = 96	115	96	88
WSUM (S) (D) n	n = 1	52	43	40
WSUMP (S) (D) n	n = 96	175	140	135
DWSUM (S) (D) n	n = 1	61	51	46
DWSUMP (S) (D) n	n = 96	515	420	395
FOR n	n = 0	11	8.9	8.1
NEXT	--	8.8	7.3	6.8
BREAK BREAKP	--	37	30	28
CALL Pn CALLP Pn	--	17	14	13
CALL Pn (S1)到(S5) CALLP Pn (S1)到(S5)	--	245	200	190
RET	返回正常程序	16	13	12
FCALL Pn FCALLP Pn	--	29	24	22
FCALL Pn (S1)到(S5) FCALLP Pn (S1)到(S5)	--	250	205	190

附

附录

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
COM	--	110	77	72
IX	--	65	54	51
IXEND	--	30	26	25
IXDEV + IXSET	触点数为 1	145	120	110
	触点数为 14	770	630	585
FIFW FIFWP	数据点数为 0	36	32	28
	数据点数为 96	36	32	28
FIFR FIFRP	数据点数为 1	45	41	36
	数据点数为 96	93	82	70
FPOP FPOPP	数据点数为 1	40	37	32
	数据点数为 96	40	37	32
FINS FINSP	数据点数为 0	53	44	38
	数据点数为 96	100	89	76
FDEL FDELP	数据点数为 1	60	50	43
	数据点数为 96	110	95	82
FROM n1 n2 (D) n3	n3 = 1	125	105	93
FROMP n1 n2 (D) n3 *1	n3 = 1000	740	695	685
DFRO n1 n2 (D) n3	n3 = 1	130	110	100
DFROP n1 n2 (D) n3 *1	n3 = 500	745	695	675
TO n1 n2 (S) n3	n3 = 1	120	105	92
TOP n1 n2 (S) n3 *1	n3 = 1000	735	680	645
DTO n1 n2 (S) n3	n3 = 1	130	110	99
DTOP n1 n2 (S) n3 *1	n3 = 500	740	680	640

*1 ： FROM/TO 指令的处理时间随插槽数和安装的模块数而变化。
（CPU 的处理时间也随扩展基板类型而变化。）

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
LIMIT LIMITP	--	34	28	26
DLIMIT DLIMITP	--	41	34	30
BAND BANDP	--	33	28	25
DBAND DBANDP	--	40	34	30
ZONE ZONEP	--	31	25	24
DZONE DZONEP	--	37	29	28
RSET RSETP	--	--	18	16
DATERD DATERDP	--	30	25	23
DATEWR DATEWRP	--	69	57	54
DATE+ DATE+P	无进位	47	39	36
	有进位	50	42	38
DATE - DATE - P	无进位	47	40	36
	有进位	50	42	38
SECOND SECONDP	--	28	24	22
HOUR HOURP	--	38	32	29
WDT WDTP	--	18	15	14
DUTY	--	41	36	32
ZRRDB ZRRDBP	--	--	24	22
ZRWRB ZRWRBP	--	--	27	24
ADRSET ADRSETP	--	23	19	18
ZPUSH ZPUSHP	--	38	33	30
ZPOP ZPOPP	--	37	31	29
ZCOM	--	105	82	80

(4) QCPU 指令的处理时间（只包括 QCPU 指令）

指令	条件（软元件）	处理时间（μs）		
		Q00JCPU	Q00CPU	Q01CPU
UNIRD UNIRDP	n = 1	96	80	74
	n = 16	440	370	340

(5) 可以被头五位系列号为 “04122”或更大数字的产品执行的指令

指令	条件（软元件）			处理时间（μs）		
				Q00JCPU	Q00CPU	Q01CPU
LDE =	单精度	导通时		43.0	35.5	33.0
		不导通时		46.0	38.0	35.5
ANDE =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	35.5	29.5	26.5
			不导通时	42.0	35.0	32.5
ORE =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	42.0	35.0	32.5
			不导通时	37.0	31.0	28.5
LDE < >	单精度	导通时		46.0	38.0	35.5
		不导通时		43.5	36.0	33.0
ANDE < >	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	38.5	31.5	29.0
			不导通时	39.5	33.0	30.5
ORE < >	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	45.0	37.5	35.0
			不导通时	34.5	29.0	26.5
LDE >	单精度	导通时		46.0	37.5	35.5
		不导通时		46.0	38.5	35.0
ANDE >	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	38.5	32.0	29.0
			不导通时	42.0	35.0	32.5
ORE >	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	45.0	37.5	34.5
			不导通时	37.0	31.0	29.0
LDE < =	单精度	导通时		45.5	37.5	35.0
		不导通时		46.5	38.5	35.5
ANDE < =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	38.5	31.5	29.0
			不导通时	42.5	35.5	32.5
ORE < =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	45.0	37.5	34.5
			不导通时	37.5	31.5	28.5
LDE <	单精度	导通时		45.5	37.5	35.0
		不导通时		46.5	38.5	35.5
ANDE <	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	38.0	31.5	29.0
			不导通时	42.5	35.5	32.5
ORE <	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	45.0	37.5	34.5
			不导通时	37.5	31.5	29.0
LDE > =	单精度	导通时		45.5	38.0	35.5
		不导通时		46.5	38.0	35.0
ANDE > =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	38.5	32.0	29.0
			不导通时	42.5	35.5	32.5

指令	条件（软元件）			处理时间（μs）		
				Q00JCPU	Q00CPU	Q01CPU
ORE > =	单精度	当不被执行时		1.5	1.2	1.0
		当被执行时	导通时	45.0	38.5	34.5
			不导通时	37.5	31.0	28.5
E+ (S) (D) E+P (S) (D)	单精度	(S) = 0、(D) = 0		29.5	25.0	23.0
		(S) = 2^{127}、(D) = 2^{127}		65.5	60.5	49.5
E+ (S1) (S2) (D) E+P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0		31.0	27.0	24.0
		(S1) = 2^{127}、(S2) = 2^{127}		66.5	56.0	51.0
E - (S) (D) E -P (S) (D)	单精度	(S) = 0、(D) = 0		29.5	25.0	23.0
		(S) = 2^{127}、(D) = 2^{127}		48.5	41.0	37.5
E - (S1) (S2) (D) E -P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0		31.0	27.0	24.0
		(S1) = 2^{127}、(S2) = 2^{127}		50.5	42.5	38.5
E* (S1) (S2) (D) E*P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0		30.0	25.5	23.0
		(S1) = 2^{127}、(S2) = 2^{127}		65.5	55.0	49.5
E/ (S1) (S2) (D) E/P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 1		30.0	26.0	23.0
		(S1) = 2^{127}、(S2) = -2^{126}		69.5	57.5	53.0
INT INTP	单精度	(S) = 0		21.5	18.5	16.0
		(S) = 32766.5		38.0	32.0	29.5
DINT DINTP	单精度	(S) = 0		23.0	19.5	17.5
		(S) = 1234567890.3		42.0	35.5	32.0
FLT FLTP	单精度	(S) = 0		22.5	19.5	17.0
		(S) = $7FFF_H$		26.5	23.0	20.0
DFLT DFLTP	单精度	(S) = 0		23.0	20.0	17.5
		(S) = $7FFFFFFF_H$		26.0	23.5	19.5
ENEG ENEGP	(S) = 0			20.5	17.0	15.5
	(S) = E - 1.0			31.5	26.0	24.0
EMOV EMOVP	--			1.5	1.2	1.0
ESTR ESTRP	--			604.0	686.0	831.0
EVAL EVALP	小数点格式　所有的 2 位			138.0	148.0	196.0
	指数格式　所有的 6 位			164.0	177.0	214.0

附

附录

指令	条件（软元件）		处理时间（μs）		
			Q00JCPU	Q00CPU	Q01CPU
SIN SINP	单精度		204.0	173.0	157.0
COS COSP	单精度		187.0	158.0	144.0
TAN TANP	单精度		224.0	190.0	173.0
RAD RADP	单精度		51.0	43.0	39.0
DEG DEGP	单精度		51.0	43.0	39.0
SQR SQRP	单精度		60.0	51.0	46.5
EXP EXPP	单精度	Ⓢ = -10	306.0	259.0	235.0
		Ⓢ = 1	306.0	259.0	235.0
LOG LOGP	单精度	Ⓢ = 1	73.0	61.5	56.0
		Ⓢ = 10	301.0	255.0	232.0
RND RNDP	--		12.5	11.0	10.0
SRND SRNDP	--		13.5	12.0	11.0

指令名称	条件 / 处理点数		处理时间 (μs)		
			Q00JCPU	Q00CPU	Q01CPU
COM *2	带 CPU 共享存储区的自动刷新	刷新范围 :2k 字 (0.5k 字被平均分配到所有站点)	--	920	880
	不带 CPU 共享存储区的自动刷新	--	--	150	135
FROM	从主站点的 CPU 共享存储区读	n3 = 1	--	100	90
		n3 = 320	--	440	420
	从其它站点的 CPU 共享存储区读	n3 = 1	--	110	105
		n3 = 320	--	305	290
TO	向主站点的 CPU 共享存储区写	n3 = 1	--	100	95
		n3 = 320	--	440	425
S.TO	向主站点的CPU共享存储区写	n4 = 1	--	205	195
		n4 = 320	--	545	525

*2: 如果与多 CPU 系统的其它 CPU 的处理互相重叠，处理时间最多增加下列时间。

对于只有主基板的系统

(指令处理时间增加) = 4 × 0.54 × (处理点数) × (其它 CPU 的数目) (μs)

对于有扩展基板的系统

(指令处理时间增加) = 4 × 1.30 × (处理点数) × (其它 CPU 的数目) (μs)

附录 1.3 高性能型 QCPU/ 过程 CPU/ 冗余 CPU/QnACPU 的运算处理时间

每条指令的处理时间列在下表中：

指令的处理时间会由于指令源和目标的种类不同而产生一定的变化，因此包含在下表中的值应作为处理时间的一套基本准则，而不能作为准确的时间来计算。

(1) 顺控指令

指令	条件（软元件）	处理时间（μs）							
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
LD LDI AND ANI OR ORI	--	0.20	0.15	0.075		0.079	0.034	0.034	0.034
LDP LDF ANDP ANDF ORP ORF	--	6.6	5.0	2.5		0.158	0.068	0.068	0.068
ANB ORB MPS MRD MPP	--	0.20	0.15	0.075		0.079	0.034	0.034	0.034
INV	当不被执行时	2.4	1.8	0.9		0.079	0.034	0.034	0.034
	当被执行时								
MEP MEF	当不被执行时	2.0	1.5	0.75		0.173	0.073	0.073	0.073
	当被执行时								
EGP EGF	当不被执行时 (OFF→OFF) (ON→ON)	0.6	0.3	0.15		0.158	0.068	0.068	0.068
	当被执行时 (OFF→ON) (ON→OFF)								

指令	条件（软元件）				处理时间（μs）						
					Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
OUT	无变化时 (OFF→OFF) (ON→ON)				0.40	0.30	0.15		0.158	0.068	0.068	0.068
	变化时 (OFF→ON) (ON→OFF)				0.40	0.30	0.15		0.158	0.068	0.068	0.068
	F	OFF 时 ON 时			7.0	5.3	2.7		2.8	1.2	1.2	1.2
		ON 时	显示时		167	126	63		162	69.7	69.7	69.7
			显示完成时		166	125	62		126	54	54	54
	T	当不被执行时			1.6	1.2	0.6		0.63	0.27	0.27	0.27
		当被执行时	时间到后		1.6	1.2	0.6		0.63	0.27	0.27	0.27
			加法运算时	K	1.6	1.2	0.6		0.63	0.27	0.27	0.27
				D	1.6	1.2	0.6		0.63	0.27	0.27	0.27
	C	当不被执行时			1.6	1.2	0.6		0.63	0.27	0.27	0.27
		当被执行时	时间到后		1.6	1.2	0.6		0.63	0.27	0.27	0.27
			加法运算时	K	1.6	1.2	0.6		0.63	0.27	0.27	0.27
				D	1.6	1.2	0.6		0.63	0.27	0.27	0.27
OUTH	T	当不被执行时			1.6	1.2	0.6		0.63	0.27	0.27	0.27
		当被执行时	时间到后		1.6	1.2	0.6		0.63	0.27	0.27	0.27
			加法运算时	K	1.6	1.2	0.6		0.63	0.27	0.27	0.27
				D	1.6	1.2	0.15		0.63	0.27	0.27	0.27
SET	当不被执行时				0.40	0.30	0.15		0.158	0.068	0.068	0.068
	当被执行时	无变化时 (ON→ON)			0.40	0.30	0.15		0.158	0.068	0.068	0.068
		变化时 (OFF→ON)			0.40	0.30	0.45		0.158	0.068	0.068	0.068
	F	当不被执行时			1.2	0.90	104		0.47	0.20	0.20	0.20
		当被执行时	显示时		277	208	0.45		161	69	69	69
			显示完成时		1.2	0.90	0.15		0.47	0.20	0.20	0.20
RST	当不被执行时				0.40	0.30	0.15		0.158	0.068	0.068	0.068
	当被执行时	无变化时 (OFF→OFF)			0.40	0.30	0.15		0.158	0.068	0.068	0.068
		变化时 (ON→OFF)			0.40	0.30	0.15		0.158	0.068	0.068	0.068
	SM	当不被执行时			0.40	0.30	0.15		0.158	0.068	0.068	0.068
		当被执行时			0.40	0.30	0.15		0.158	0.068	0.068	0.068
	F	当不被执行时			1.2	0.90	0.45		0.47	0.20	0.20	0.20
		当被执行时	显示时		148	112	56		90	38	38	38
			显示完成时		1.2	0.90	0.45		0.47	0.20	0.20	0.20
	T、C	当不被执行时			1.4	1.1	0.6		0.63	0.27	0.27	0.27
		当被执行时							0.63	0.27	0.27	0.27
	D	当不被执行时			0.60	0.45	0.23		0.24	0.10	0.10	0.10
		当被执行时							0.24	0.10	0.10	0.10
	Z	当不被执行时			1.2	0.90	0.45		0.47	0.20	0.20	0.20
		当被执行时			10.8	8.1	4.1		4.3	1.9	1.9	1.9
	R	当不被执行时			1.0	0.75	0.38		0.40	0.17	0.17	0.17
		当被执行时							0.40	0.17	0.17	0.17
PLS PLF	--				2.6	2.0	0.98		1.0	0.44	0.44	0.44
FF	Y	当不被执行时			1.2	0.90	0.45		0.47	0.20	0.20	0.20
		当被执行时							0.47	0.20	0.20	0.20
DELTA DELTAP	DY0	当不被执行时			1.2	0.90	0.45		0.47	0.20	0.20	0.20
		当被执行时			16.8	14.1	11.1		5.9	2.6	2.6	2.6

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
SFT SFTP	当不被执行时	1.2	0.90	0.45		0.47	0.20	0.20	0.20
	当被执行时	4.2	3.2	1.6		1.66	0.71	0.71	0.71
MC	--	0.60	0.45	0.23		0.24	0.10	0.10	0.10
MCR	--	0.20	0.15	0.075		0.079	0.034	0.034	0.034
FEND END	执行错误检查时	1643	1236	618		380	150	150	500
	不执行错误检查时 （・电池检查） （・保险丝熔断检查） （・I/O 模块校验）	1106	832	416		380	150	150	500
NOP	--	0.2	0.15	0.075		0.079	0.034	0.034	0.034
NOPLF PAGE	--	0.2	0.15	0.075		0.079	0.034	0.034	0.034

(2) 基本指令

当指令不被执行时，处理时间按如下方式计算：

Q2ACPU(S1)、 Q2ASCPU(S1) ..... 0.20 × （每条指令的步数 + 1)μs

Q3ACPU..... 0.15 × （每条指令的步数 + 1)μs

Q2ASHCPU(S1)、Q4ACPU、Q4ARCPU .. 0.075 × （每条指令的步数 + 1)μs

Q02CPU..... 0.079 × （每条指令的步数 + 1)μs

Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU、Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU

..... 0.034 × （每条指令的步数 + 1)μs

指令	条件（软元件）		处理时间 (μs)						
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
LD =	导通时		3.8	2.9	1.5		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10
OR =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LD < >	导通时		4.4	3.3	1.7		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND < >	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10
OR < >	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LD >	导通时		4.4	3.3	1.7		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND >	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10
OR >	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LD < =	导通时		4.4	3.3	1.7		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND < =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10

指令	条件（软元件）		处理时间（μs）						
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
OR < =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LD <	导通时		4.4	3.3	1.7		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND <	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10
OR <	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LD > =	导通时		4.4	3.3	1.7		0.24	0.10	0.10	0.10
	不导通时		3.6	2.7	1.4		0.24	0.10	0.10	0.10
AND > =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
		不导通时	3.2	2.4	1.2		0.24	0.10	0.10	0.10
OR > =	当不被执行时		1.4	1.1	0.55		0.24	0.10	0.10	0.10
	当被执行时	导通时	3.8	2.9	1.5		0.24	0.10	0.10	0.10
		不导通时	2.8	2.1	1.1		0.24	0.10	0.10	0.10
LDD =	导通时		5.0	3.8	1.9		0.55	0.24	0.24	0.24
	不导通时		4.2	3.2	1.6		0.39	0.17	0.17	0.17
ANDD =	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
		不导通时	3.8	2.9	1.5		0.39	0.17	0.17	0.17
ORD =	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	4.4	3.3	1.7		0.55	0.24	0.24	0.24
		不导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
LDD < >	导通时		5.0	3.8	1.9		0.55	0.24	0.24	0.24
	不导通时		4.2	3.2	1.6		0.55	0.24	0.24	0.24
ANDD < >	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
		不导通时	3.8	2.9	1.5		0.55	0.24	0.24	0.24
ORD < >	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	4.4	3.3	1.7		0.55	0.24	0.24	0.24
		不导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
LDD >	导通时		3.8	2.9	1.5		0.55	0.24	0.24	0.24
	不导通时		4.2	3.2	1.6		0.55	0.24	0.24	0.24
ANDD >	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	2.8	2.1	1.1		0.55	0.24	0.24	0.24
		不导通时	3.8	2.9	1.5		0.55	0.24	0.24	0.24
ORD >	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	3.8	2.9	1.5		0.55	0.24	0.24	0.24
		不导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
LDD < =	导通时		4.4	3.3	1.7		0.55	0.24	0.24	0.24
	不导通时		3.6	2.7	1.4		0.55	0.24	0.24	0.24
ANDD < =	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
		不导通时	3.2	2.4	1.2		0.55	0.24	0.24	0.24
ORD < =	当不被执行时		1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时	导通时	4.4	3.3	1.7		0.55	0.24	0.24	0.24
		不导通时	2.8	2.1	1.1		0.55	0.24	0.24	0.24

指令	条件（软元件）			处理时间（μs）							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
LDD <	导通时			3.8	2.9	1.5		0.55	0.24	0.24	0.24
	不导通时			4.2	3.2	1.6		0.55	0.24	0.24	0.24
ANDD <	当不被执行时			1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时		导通时	2.8	2.1	1.1		0.55	0.24	0.24	0.24
			不导通时	3.8	2.9	1.5		0.55	0.24	0.24	0.24
ORD <	当不被执行时			1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时		导通时	3.8	2.9	1.5		0.55	0.24	0.24	0.24
			不导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
LDD > =	导通时			4.4	3.3	1.7		0.55	0.24	0.24	0.24
	不导通时			3.6	2.7	1.4		0.55	0.24	0.24	0.24
ANDD > =	当不被执行时			1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时		导通时	3.4	2.6	1.3		0.55	0.24	0.24	0.24
			不导通时	3.2	2.4	1.2		0.55	0.24	0.24	0.24
ORD > =	当不被执行时			1.4	1.1	0.55		0.39	0.17	0.17	0.17
	当被执行时		导通时	4.4	3.3	1.7		0.55	0.24	0.24	0.24
			不导通时	2.8	2.1	1.1		0.55	0.24	0.24	0.24
LDE = *1	单精度	导通时		--	--	--	35	93	40	6.4	6.4
								14.9	6.4		
		不导通时		--	--	--	87	92	40	6.4	6.4
								14.9	6.4		
	双精度	导通时		235	177	89	--	93	40	--	--
								14.9	6.4		
		不导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
ANDE = *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	93	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	86	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	--	--	--	--
		当被执行时	导通时	234	176	88	--	93	40	--	--
								14.9	6.4		
			不导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
ORE = *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	35	93	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	86	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	234	176	88	--	93	40	--	--
								14.9	6.4		
			不导通时	230	172	86	--	92	40	--	--
								14.9	6.4		

*1 ： Qn/QnH 的处理时间随 CPU 模块的系列号而变化。
上行：系列号的首五位是 “05031”或更小的数字
下行：系列号的首五位是 “05032”或更大的数字
当指令不执行的条件得到满足时，在上行和下行之间没有差别。

指令	条件（软元件）			处理时间（μs）							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
LDE<> *1	单精度	导通时		--	--	--	35	92	40	6.4	6.4
								14.9	6.4		
		不导通时		--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
		不导通时		234	176	88	--	92	40	--	--
								14.9	6.4		
ANDE<> *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	93	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
ORE<> *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	行	--	--	35	93	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	231	174	87	--	93	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
LDE> *1	单精度	当不被执行时		--	--	--	35	92	40	6.4	6.4
								14.9	6.4		
		导通时		--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	不导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
				234	176	88	--	92	40	--	--
								14.9	6.4		
ANDE> *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	93	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		

*1 ： Qn/QnH 的处理时间随 CPU 模块的系列号而变化。
上行：系列号的首五位是 “05031”或更小的数字
下行：系列号的首五位是 “05032”或更大的数字
当指令不执行的条件得到满足时，在上行和下行之间没有差别。

指令	条件（软元件）			处理时间（μs）							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
ORE> *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	93	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	231	174	87	--	93	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
LDE<= *1	单精度	导通时		--	--	--	34	93	40	6.4	6.4
								14.9	6.4		
		不导通时		--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	导通时		235	177	89	--	93	40	--	--
								14.9	6.4		
		不导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
ANDE<= *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	86	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
			不导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
ORE<= *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	86	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
			不导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
LDE< *1	单精度	导通时		--	--	--	35	92	40	6.4	6.4
								14.9	6.4		
		不导通时		--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
		不导通时		234	176	88	--	92	40	--	--
								14.9	6.4		

*1 ： Qn/QnH 的处理时间随 CPU 模块的系列号而变化。
上行：系列号的首五位是 “05031”或更小的数字
下行：系列号的首五位是 “05032”或更大的数字
当指令不执行的条件得到满足时，在上行和下行之间没有差别。

指令	条件（软元件）			处理时间（μs）							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
ANDE< *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
ORE< *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	93	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	88	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	231	174	87	--	93	40	--	--
								14.9	6.4		
			不导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
LDE>= *1	单精度	导通时		--	--	--	35	93	40	6.4	6.4
								14.9	6.4		
		不导通时		--	--	--	87	92	40	6.4	6.4
								14.9	6.4		
	双精度	导通时		235	177	89	--	93	40	--	--
								14.9	6.4		
		不导通时		231	174	87	--	92	40	--	--
								14.9	6.4		
ANDE>= *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	87	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
			不导通时	231	174	87	--	92	40	--	--
								14.9	6.4		

*1 ： Qn/QnH 的处理时间随 CPU 模块的系列号而变化。
上行：系列号的首五位是 “05031”或更小的数字
下行：系列号的首五位是 “05032”或更大的数字
当指令不执行的条件得到满足时，在上行和下行之间没有差别。

指令	条件（软元件）			处理时间（μs）							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
ORE>= *1	单精度	当不被执行时		--	--	--	0.55	0.55	0.24	0.24	0.24
		当被执行时	导通时	--	--	--	34	92	40	6.4	6.4
								14.9	6.4		
			不导通时	--	--	--	86	92	40	6.4	6.4
								14.9	6.4		
	双精度	当不被执行时		1.4	1.1	0.55	--	0.55	0.24	--	--
		当被执行时	导通时	234	176	88	--	92	40	--	--
								14.9	6.4		
			不导通时	230	172	86	--	92	40	--	--
								14.9	6.4		
LD$ =	导通时			97	73	37		38	16	16	16
	不导通时			81	61	31		34	15	15	15
AND$ =	当不被执行时			1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时		导通时	96	72	36		39	17	17	17
			不导通时	81	61	31		32	14	14	14
OR$ =	当不被执行时			1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时		导通时	97	73	37		40	17	17	17
			不导通时	80	60	30		33	14	14	14
LD$ < >	导通时			83	62	31		32	14	14	14
	不导通时			97	73	37		40	17	17	17
AND$ < >	当不被执行时			1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时		导通时	80	60	30		33	14	14	14
			不导通时	96	72	36		39	17	17	17
OR$ < >	当不被执行时			1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时		导通时	81	61	31		32	14	14	14
			不导通时	96	72	36		39	17	17	17
LD$ >	导通时			83	62	31		32	14	14	14
	不导通时			97	73	37		40	17	17	17

*1 ： Qn/QnH 的处理时间随 CPU 模块的系列号而变化。
上行：系列号的首五位是 “05031”或更小的数字
下行：系列号的首五位是 “05032”或更大的数字
当指令不执行的条件得到满足时，在上行和下行之间没有差别。

指令	条件（软元件）		处理时间（μs）						
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
AND$ >	当不被执行时		1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时	导通时	80	60	30		33	14	14	14
		不导通时	96	72	36		39	17	17	17
OR$ >	当不被执行时		1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时	导通时	81	61	31		32	14	14	14
		不导通时	96	72	36		39	17	17	17
LD$ < =	导通时		97	73	37		40	17	17	17
	不导通时		81	61	31		32	14	14	14
AND$ < =	当不被执行时		1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时	导通时	96	72	36		39	17	17	17
		不导通时	81	61	31		32	14	14	14
OR$ < =	当不被执行时		1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时	导通时	97	73	37		40	17	17	17
		不导通时	80	60	30		33	14	14	14
LD$ <	导通时		81	61	31		32	14	14	14
	不导通时		97	73	37		40	17	17	17
AND$ <	当不被执行时		1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时	导通时	80	60	30		32	14	14	14
		不导通时	96	72	36		39	16	16	16
OR$ <	当不被执行时		1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时	导通时	81	61	31		32	14	14	14
		不导通时	96	72	36		39	16	16	16
LD$ > =	导通时		97	73	37		40	17	17	17
	不导通时		81	61	31		32	14	14	14
AND$ > =	当不被执行时		1.4	1.1	0.55		0.56	0.23	0.23	0.23
	当被执行时	导通时	96	72	36		39	16	16	16
		不导通时	81	61	31		32	14	14	14
OR$ > =	当不被执行时		1.4	1.1	0.55		0.56	0.24	0.24	0.24
	当被执行时	导通时	97	73	37		39	17	17	17
		不导通时	80	60	30		32	14	14	14
BKCMP = (S1) (S2) (D) n	n = 1		120	90	45		48	21	21	21
BKCMP = P (S1) (S2) (D) n	n = 96		367	276	138		142	61	61	61
BKCMP <> (S1) (S2) (D) n	n = 1		123	92	46		48	21	21	21
BKCMP <>P (S1) (S2) (D) n	n = 96		346	260	130		150	65	65	65
BKCMP > (S1) (S2) (D) n	n = 1		123	92	96		48	21	21	21
BKCMP >P (S1) (S2) (D) n	n = 96		366	275	138		142	61	61	61
BKCMP >= (S1) (S2) (D) n	n = 1		121	91	46		48	21	21	21
BKCMP >=P (S1) (S2) (D) n	n = 96		386	290	145		150	65	65	65
BKCMP < (S1) (S2) (D) n	n = 1		121	91	96		48	21	21	21
BKCMP <P (S1) (S2) (D) n	n = 96		366	275	138		158	68	68	68
BKCMP <= (S1) (S2) (D) n	n = 1		121	91	46		48	21	21	21
BKCMP <=P (S1) (S2) (D) n	n = 96		348	261	131		150	65	65	65

指令	条件（软元件）	处理时间（μs）							
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
+ (S) (D) +P (S) (D)	当被执行时	2.4	1.8	0.9		0.39	0.17	0.17	0.17
+ (S1) (S2) (D) +P (S1) (S2) (D)	当被执行时	2.7	2.0	1.0		0.47	0.20	0.20	0.20
- (S) (D) - P (S) (D)	当被执行时	2.4	1.8	0.9		0.39	0.17	0.17	0.17
- (S1) (S2) (D) - P (S1) (S2) (D)	当被执行时	2.6	2.0	1.0		0.47	0.20	0.20	0.20
D+ (S) (D) D+P (S) (D)	当被执行时	2.8	2.1	1.1		0.71	0.31	0.31	0.31
D+ (S1) (S2) (D) D+P (S1) (S2) (D)	当被执行时	3.2	2.4	1.2		0.79	0.34	0.34	0.34
D - (S) (D) D - P (S) (D)	当被执行时	2.8	2.1	1.1		0.71	0.30	0.30	0.30
D - (S1) (S2) (D) D - P (S1) (S2) (D)	当被执行时	3.2	2.4	1.2		0.79	0.34	0.34	0.34
* (S1) (S2) (D) * P (S1) (S2) (D)	当被执行时	2.8	2.1	1.1		0.47	0.20	0.20	0.20
/ (S1) (S2) (D) /P (S1) (S2) (D)	--	6.8	5.1	2.6		2.7	1.2	1.2	1.2
D * (S1) (S2) (D) D * P (S1) (S2) (D)	--	20	15	7.5		7.9	3.4	3.4	3.4
D/ (S1) (S2) (D) D/P (S1) (S2) (D)	--	36	27	13.5		14	6.1	6.1	6.1
B+ (S) (D) B+P (S) (D)	--	5.5	4.1	2.1		2.2	1.0	1.0	1.0
B+ (S1) (S2) (D) B+P (S1) (S2) (D)	--	13	9.6	4.8		5.0	2.2	2.2	2.2
B - (S) (D) B - P (S) (D)	--	5.2	3.9	2.0		2.0	0.9	0.9	0.9
B - (S1) (S2) (D) B - P (S1) (S2) (D)	--	13	9.4	4.7		4.9	2.1	2.1	2.1
DB+ (S) (D) DB+P (S) (D)	--	29	22	11		12	5.0	5.0	5.0
DB+ (S1) (S2) (D) DB+P (S1) (S2) (D)	--	32	24	12		12	5.3	5.3	5.3
DB - (S) (D) DB - P (S) (D)	--	29	22	11		11	4.8	4.8	4.8
DB - (S1) (S2) (D) DB - P (S1) (S2) (D)	--	32	24	12		12	5.2	5.2	5.2
B * (S1) (S2) (D) B * P (S1) (S2) (D)	--	9.4	7.1	3.6		3.7	1.6	1.6	1.6

指令	条件（软元件）		处理时间（μs）							
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
B/ (S1) (S2) (D) B/P (S1) (S2) (D)	--		9.4	7.1	3.6		3.8	1.6	1.6	1.6
DB * (S1) (S2) (D) DB * P (S1) (S2) (D)	--		62	46	23		24	10	10	10
DB/ (S1) (S2) (D) DB/P (S1) (S2) (D)	--		69	52	26		27	12	12	12
E+ (S) (D) E+P (S) (D)	单精度	(S) = 0、(D) = 0	--	--	--	35	1.8	0.78	0.78	0.78
		(S) = 2^{127}、(D) = 2^{127}	--	--	--	35	1.8	0.78	0.78	0.78
	双精度	(S) = 0、(D) = 0	54	40	20	--	203	87	--	--
		(S) = 2^{127}、(D) = 2^{127}	524	394	197	--	203	87	--	--
E+ (S1) (S2) (D) E+P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	--	--	--	35	2.4	1.1	1.1	1.1
		(S1) = 2^{127}、(S2) = 2^{127}	--	--	--	35	2.4	1.1	1.1	1.1
	双精度	(S1) = 0、(S2) = 0	54	40	20	--	209	90	--	--
		(S1) = 2^{127}、(S2) = 2^{127}	524	394	197	--	209	90	--	--
E - (S) (D) E -P (S) (D)	单精度	(S) = 0、(D) = 0	--	--	--	35	1.8	0.78	0.78	0.78
		(S) = 2^{127}、(D) = 2^{127}	--	--	--	35	1.8	0.78	0.78	0.78
	双精度	(S) = 0、(D) = 0	54	40	20	--	202	87	--	--
		(S) = 2^{127}、(D) = 2^{127}	515	387	194	--	202	87	--	--
E - (S1) (S2) (D) E -P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	--	--	--	35	2.4	1.1	1.1	1.1
		(S1) = 2^{127}、(S2) = 2^{127}	--	--	--	36	2.4	1.1	1.1	1.1
	双精度	(S1) = 0、(S2) = 0	55	41	21	--	210	90	--	--
		(S1) = 2^{127}、(S2) = 2^{127}	520	391	146	--	210	90	--	--
E* (S1) (S2) (D) E*P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	--	--	--	35	2.4	1.1	1.1	1.1
		(S1) = 2^{126}、(S2) = 2^{127}	--	--	--	36	2.4	1.1	1.1	1.1
	双精度	(S1) = 0、(S2) = 0	55	41	21	--	222	96	--	--
		(S1) = 2^{126}、(S2) = 2^{127}	567	426	218	--	222	96	--	--
E/ (S1) (S2) (D) E/P (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 1	--	--	--	37	12	5.2	5.2	5.2
		(S1) = 2^{127}、(S2) = - 2^{126}	--	--	--	38	12	5.2	5.2	5.2
	双精度	(S1) = 0、(S2) = 1	149	112	56	--	369	159	--	--
		(S1) = 2^{127}、(S2) = - 2^{126}	1109	834	417	--	369	159	--	--

指令	条件（软元件）		处理时间（μs）							
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
$+ (S) (D) $+P (S) (D)	--		179	134	67		68	29	29	29
$+ (S1) (S2) (D) $+P (S1) (S2) (D)	--		206	155	78		81	35	35	35
INC INCP	--		1.9	1.4	0.7		0.32	0.14	0.14	0.14
DINC DINCP	--		2.3	1.7	0.9		0.47	0.20	0.20	0.20
DEC DECP	--		1.9	1.4	0.7		0.32	0.14	0.14	0.14
DDEC DDECP	--		2.3	1.7	0.9		0.47	0.20	0.20	0.20
BCD BCDP	--		2.7	2.0	1.0		1.1	0.48	0.48	0.48
DBCD DBCDP	--		7.9	5.9	3.0		3.2	1.4	1.4	1.4
BIN BINP	--		2.7	2.0	1.0		1.0	0.44	0.44	0.44
DBIN DBINP	--		4.8	3.6	1.8		1.9	0.82	0.82	0.82
INT INTP	单精度	(S) = 0	--	--	--	7.5	3.2	1.4	1.4	1.4
		(S) = 32766.5	--	--	--	20	3.2	1.4	1.4	1.4
	双精度	(S) = 0	20	15	7.5	--	22	9.3	--	--
		(S) = 32766.5	54	40	20	--	22	9.3	--	--
DINT DINTP	单精度	(S) = 0	--	--	--	7.5	2.5	1.1	1.1	1.1
		(S) = 1234567890.3	--	--	--	22	2.5	1.1	1.1	1.1
	双精度	(S) = 0	20	15	7.5	--	24	10	--	--
		(S) = 1234567890.3	59	44	22	--	24	10	--	--
FLT FLTP	单精度	(S) = 0	--	--	--	10	2.1	0.92	0.92	0.92
		(S) = $7FFF_H$	--	--	--	21	2.1	0.92	0.92	0.92
	双精度	(S) = 0	27	20	10	--	22	9.6	--	--
		(S) = $7FFF_H$	55	41	21	--	22	9.6	--	--
DFLT DFLTP	单精度	(S) = 0	--	--	--	11	2.1	0.88	0.88	0.88
		(S) = $7FFFFFFF_H$	--	--	--	21	2.1	0.88	0.88	0.88
	双精度	(S) = 0	28	21	11	--	26	11	--	--
		(S) = $7FFFFFFF_H$	56	42	21	--	26	11	--	--

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
DBL DBLP	--	12	8.6	4.3		4.5	1.9	1.9	1.9
WORD WORDP	--	12	9.0	4.5		4.7	2.0	2.0	2.0
GRY GRYP	--	12	9.0	4.5		4.7	2.0	2.0	2.0
DGRY DGRYP	--	14	10	5.0		5.3	2.3	2.3	2.3
GBIN GBINP	--	46	34	17		18	7.7	7.7	7.7
DGBIN DGBINP	--	83	62	31		32	14	14	14
NEG NEGP	--	9.3	7	3.5		3.6	1.6	1.6	1.6
DNEG DNEGP	--	11	8.2	4.1		4.3	1.8	1.8	1.8
ENEG ENEGP	--	9.8	7.4	3.7		3.9	1.7	1.7	1.7
BKBCD Ⓢ Ⓓ n	n = 1	102	76	38		38	17	17	17
BKBCDP Ⓢ Ⓓ n	n = 96	272	204	102		99	43	43	43
BKBIN Ⓢ Ⓓ n	n = 1	102	76	38		38	17	17	17
BKBINP Ⓢ Ⓓ n	n = 96	272	204	102		99	43	43	43
MOV MOVP	Ⓢ = D0、 Ⓓ = D1	0.6	0.5	0.225		0.24	0.10	0.10	0.10
	Ⓢ = D0、 Ⓓ = J1 \ W1	392*1	305*1	176*1		--	--	--	--
		391*1	299*1	165*1		--	--	--	--
		--	--	--		140*1	60*1	60*1	60*1
DMOV DMOVP	Ⓢ = D0、 Ⓓ = D1	2.4	1.8	0.9		0.47	0.20	0.20	0.20
	Ⓢ = D0、 Ⓓ = J1 \ W1	400*1	313*1	183*1		--	--	--	--
		395*1	301*1	167*1		--	--	--	--
		--	--	--		147*1	64*1	64*1	64*1
EMOV EMOVP	--	12	8.6	4.3		0.63	0.27	0.27	0.27
$MOV $MOVP	--	100	75	38		40	17	17	17
CML CMLP	--	2.0	1.5	0.8		0.40	0.17	0.17	0.17
DCML DCMLP	--	2.4	1.8	0.9		0.55	0.24	0.24	0.24

*1 ： 上面的一行表示使用 A38B/A1S38B 和扩展基板时的处理时间。
中间一行表示使用 A38HB/A1S38HB 时的处理时间。
下面一行表示使用 Q3128 时的处理时间。

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
BMOV ⓢ ⓓ n	n = 1	43	32	16		17	7.1	7.1	7.1
BMOVP ⓢ ⓓ n	n = 96	81	61	31		32	14	14	14
FMOV ⓢ ⓓ n	n = 1	18	13	6.5		6.7	2.9	2.9	2.9
FMOVP ⓢ ⓓ n	n = 96	36	27	14		14	6.1	6.1	6.1
XCH XCHP DXCH DXCHP	--	3.1	2.3	1.2		1.3	0.54	0.54	0.54
BXCH ⓓ1 ⓓ2 n	n = 1	77	58	29		31	13	13	13
BXCHP ⓓ1 ⓓ2 n	n = 96	213	160	80		84	36	36	36
SWAP SWAPP	--	9.2	6.9	3.5		3.7	1.6	1.6	1.6
CJ	--	7.8	5.8	2.9		3.2	1.4	1.4	1.4
SCJ	--	7.8	5.8	2.9		3.2	1.4	1.4	1.4
JMP	--	8.0	6.0	3.0		3.2	1.4	1.4	1.4
GOEND	--	2.0	1.5	0.75		0.39	0.34	0.34	0.34
DI	--	2.3	1.7	0.9		0.95	0.41	0.41	0.41
EI	--	3.1	2.3	1.2		1.3	0.54	0.54	0.54
IMASK	--	8.1	6.5	3.3		11	4.6	4.6	4.6
IRET	--	4.0	3.0	1.5		1.6	0.68	0.68	0.68
RFS	n = 1	31.3	23.4	11.7		6.7	4.7	4.7	4.7
RFSP	n = 96	97.6	72.8	36.4		19	13	13	13
UDCNT1	--	42.6	31.8	15.9		15	6.5	6.5	--
UDCNT2	--	44.6	33.3	16.7		16	6.8	6.8	--
TTMR	--	25.9	19.3	9.7		10	4.4	4.4	--
STMR	--	41.7	31.1	15.6		20	7.1	7.1	--
ROTC	--	66.1	49.3	24.7		26	11	11	--
RAMP	--	45.4	33.9	17.0		18	7.7	7.7	--
SPD	--	48.9	36.5	18.3		19	8.3	8.3	--
PLSY	--	26.9	20.1	10.1		10	4.5	4.5	--
PWM	--	32.8	24.5	12.3		9.1	3.9	3.9	--
MTR	--	29.2	21.8	10.9		11	4.9	4.9	--

(3) 应用指令

指令不执行时的处理时间按照如下方式计算：

Q2ACPU(S1)、Q2ASCPU(S1) ..... 0.20 × (每条指令的步数 + 1)μs

Q3ACPU ..... 0.15 × (每条指令的步数 + 1)μs

Q2ASHCPU(S1)、Q4ACPU、Q4ARCPU .. 0.075 × (每条指令的步数 + 1)μs

Q02CPU ..... 0.079 × (每条指令的步数 + 1)μs

Q02HCPU、Q06HCPU、Q12HCPU、Q25HCPU、Q12PHCPU、Q25PHCPU、Q12PRHCPU、Q25PRHCPU

..... 0.034 × (每条指令的步数 + 1)μs

指令	条件（软元件）	处理时间(μs)						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
WAND (S) (D) WANDP (S) (D)	当被执行时	2.4	1.8	0.9		0.39	0.17	0.17	0.17
WAND (S1) (S2) (D) WANDP (S1) (S2) (D)	当被执行时	9.5	7.1	3.6		0.47	0.20	0.20	0.20
DAND (S) (D) DANDP (S) (D)	当被执行时	3.0	2.3	1.2		0.71	0.31	0.31	0.31
DAND (S1) (S2) (D) DANDP (S1) (S2) (D)	当被执行时	19	14	7.0		0.79	0.34	0.34	0.34
BKAND (S1) (S2) (D) n	n = 1	89	67	34		36	16	16	16
BKANDP (S1) (S2) (D) n	n = 96	184	138	69		74	32	32	32
WOR (S) (D) WORP (S) (D)	当被执行时	2.4	1.8	0.9		0.40	0.17	0.17	0.17
WOR (S1) (S2) (D) WORP (S1) (S2) (D)	当被执行时	9.5	7.1	3.6		0.47	0.20	0.20	0.20
DOR (S) (D) DORP (S) (D)	当被执行时	3.0	2.3	1.2		0.71	0.31	0.31	0.31
DOR (S1) (S2) (D) DORP (S1) (S2) (D)	当被执行时	19	14	7.0		0.79	0.34	0.34	0.34
BKOR (S1) (S2) (D) n	n = 1	89	67	34		36	16	16	16
BKORP (S1) (S2) (D) n	n = 96	184	138	69		74	32	32	32
WXOR (S) (D) WXORP (S) (D)	当被执行时	2.4	1.8	0.9		0.39	0.17	0.17	0.17
WXOR (S1) (S2) (D) WXORP (S1) (S2) (D)	当被执行时	17.2	7.1	3.6		0.47	0.20	0.20	0.20
DXOR (S) (D) DXORP (S) (D)	当被执行时	3.0	2.3	1.2		0.71	0.31	0.31	0.31
DXOR (S1) (S2) (D) DXORP (S1) (S2) (D)	当被执行时	19	14	7.0		0.79	0.34	0.34	0.34
BKXOR (S1) (S2) (D) n	n = 1	89	67	34		36	16	16	16
BKXORP (S1) (S2) (D) n	n = 96	184	138	69		74	32	32	32

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
WXNR (S) (D) WXNRP (S) (D)	当被执行时	2.4	1.8	0.9		0.40	0.17	0.17	0.17
WXNR (S1) (S2) (D) WXNRP (S1) (S2) (D)	当被执行时	9.5	7.1	3.6		0.47	0.20	0.20	0.20
DNXR (S) (D) DNXRP (S) (D)	当被执行时	3.0	2.3	1.2		0.71	0.31	0.31	0.31
DNXR (S1) (S2) (D) DNXRP (S1) (S2) (D)	当被执行时	24	18	9		0.79	0.34	0.34	0.34
BKNXOR (S1) (S2) (D) n	n = 1	89	67	34		36	16	16	16
BKNXORP (S1) (S2) (D) n	n = 96	184	138	69		74	32	32	32
ROR (D) n	n = 1	5.0	3.8	1.9		2.0	0.85	0.85	0.85
RORP (D) n	n = 15	5.0	3.8	1.9		2.0	0.85	0.85	0.85
RCR (D) n	n = 1	4.0	3.0	1.5		1.6	0.68	0.68	0.68
RCRP (D) n	n = 15	4.0	3.0	1.5		1.6	0.68	0.68	0.68
ROL (D) n	n = 1	5.0	3.8	1.9		2.0	0.85	0.85	0.85
ROLP (D) n	n = 15	5.0	3.8	1.9		2.0	0.85	0.85	0.85
RCL (D) n	n = 1	4.0	3.0	1.5		1.6	0.68	0.68	0.68
RCLP (D) n	n = 15	4.0	3.0	1.5		1.6	0.68	0.68	0.68
DROR (D) n	n = 1	9.8	7.4	3.7		3.9	1.7	1.7	1.7
DRORP (D) n	n = 31	10	7.8	3.9		4.0	1.7	1.7	1.7
DRCR (D) n	n = 1	11	8.1	4.1		4.3	1.8	1.8	1.8
DRCRP (D) n	n = 31	11	8.3	4.2		4.3	1.9	1.9	1.9
DROL (D) n	n = 1	9.8	7.4	3.7		3.9	1.7	1.7	1.7
DROLP (D) n	n = 31	10	7.8	3.9		4.0	1.7	1.7	1.7
DRCL (D) n	n = 1	11	8.1	4.1		4.3	1.8	1.8	1.8
DRCLP (D) n	n = 31	11	8.3	4.2		4.3	1.9	1.9	1.9
SFR (D) n	n = 1	4.4	3.3	1.7		1.7	0.75	0.75	0.75
SFRP (D) n	n = 15	5.0	3.8	1.9		2.0	0.85	0.85	0.85
SFL (D) n	n = 1	4.4	3.3	1.7		1.7	0.75	0.75	0.75
SFLP (D) n	n = 15	5.0	3.8	1.9		2.0	0.85	0.85	0.85
BSFLR (D) n	n = 1	51	38	19		20	8.6	8.6	8.6
BSFLRP (D) n	n = 96	60	45	23		24	10	10	10
BSFL (D) n	n = 1	49	37	19		20	8.5	8.5	8.5
BSFLP (D) n	n = 96	58	44	22		23	10	10	10
DSFR (D) n	n = 1	3.6	2.6	1.3		1.3	0.58	0.58	0.58
DSFRP (D) n	n = 96	63	47	24		25	11	11	11
DSFL (D) n	n = 1	3.6	2.6	1.3		1.3	0.58	0.58	0.58
DSFLP (D) n	n = 96	65	49	25		26	11	11	11

指令	条件（软元件）		处理时间（μs）							
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
BSET Ⓓ n	n = 1		20	15	7.5		7.6	3.3	3.3	3.3
BSETP Ⓓ n	n = 15		20	15	7.5		7.6	3.3	3.3	3.3
BRST Ⓓ n	n = 1		20	15	7.5		7.6	3.3	3.3	3.3
BRSTP Ⓓ n	n = 15		20	15	7.5		7.6	3.3	3.3	3.3
TEST Ⓢ1 Ⓢ2 Ⓓ TESTP Ⓢ1 Ⓢ2 Ⓓ	--		21	16	8.0		8.2	3.5	3.5	3.5
DTEST Ⓢ1 Ⓢ2 Ⓓ DTESTP Ⓢ1 Ⓢ2 Ⓓ	--		24	18	9.0		9.2	3.9	3.9	3.9
BKRST Ⓢ n	n = 1		45	34	17		18	7.8	7.8	7.8
BKRSTP Ⓢ n	n = 96		49	37	19		19	8.2	8.2	8.2
SER Ⓢ1 Ⓢ2 Ⓓ n	n = 1	所有的都匹配	58	44	22		22	9.6	9.6	9.6
		一个都不匹配	57	43	21		21	8.9	8.9	8.9
SERP Ⓢ1 Ⓢ2 Ⓓ n	n = 96	所有的都匹配	293	220	110		115	49	49	49
		一个都不匹配	340	256	128		133	57	57	57
DSER Ⓢ1 Ⓢ2 Ⓓ n	n = 1	所有的都匹配	61	46	23		23	9.9	9.9	9.9
		一个都不匹配	58	44	22		23	9.7	9.7	9.7
DSERP Ⓢ1 Ⓢ2 Ⓓ n	n = 96	所有的都匹配	354	266	133		142	61	61	61
		一个都不匹配	354	266	133		132	57	57	57
SUM SUMP	Ⓢ = 0 Ⓢ = $FFFF_H$		9.8	7.4	3.7		3.9	1.7	1.7	1.7
DSUM DSUMP	Ⓢ = 0		12	9.0	4.5		4.7	2.0	2.0	2.0
	Ⓢ = $FFFFFFFF_H$		31	23	12		12	5.0	5.0	5.0
DECO Ⓢ Ⓓ n	n = 2		48	36	18		20	8.6	8.6	8.6
DECOP Ⓢ Ⓓ n	n = 8		62	47	24		27	12	12	12
ENCO Ⓢ Ⓓ n	n = 2	M1 = ON	52	39	20		21	9.1	9.1	9.1
		M4 = ON	52	39	20		21	9.1	9.1	9.1
ENCOP Ⓢ Ⓓ n	n = 8	M1 = ON	65	49	25		28	12	12	12
		M256 = ON	65	49	25		26	11	11	11
SEG SEGP	--		3.2	2.4	1.2		1.3	0.54	0.54	0.54
DIS Ⓢ Ⓓ n	n = 1		46	34	17		18	7.7	7.7	7.7
DISP Ⓢ Ⓓ n	n = 4		51	38	19		19	8.3	8.3	8.3
UNI Ⓢ Ⓓ n	n = 1		53	40	20		21	8.9	8.9	8.9
UNIP Ⓢ Ⓓ n	n = 4		57	43	22		23	9.7	9.7	9.7
NDIS Ⓢ1 Ⓓ Ⓢ2 NDISP Ⓢ1 Ⓓ Ⓢ2	--		104	78	39		41	18	18	18
NUNI Ⓢ1 Ⓓ Ⓢ2 NUNIP Ⓢ1 Ⓓ Ⓢ2	--		105	79	40		42	18	18	18
WTOB Ⓢ Ⓓ n	n = 1		125	94	47		47	20	20	20
WTOBP Ⓢ Ⓓ n	n = 96		257	193	97		99	43	43	43
BTOW Ⓢ Ⓓ n	n = 1		121	91	46		45	19	19	19
BTOWP Ⓢ Ⓓ n	n = 96		233	175	88		89	38	38	38

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
MAX Ⓢ Ⓓ n	n = 1	43	32	16		17	7.1	7.1	7.1
MAXP Ⓢ Ⓓ n	n = 96	318	239	120		136	59	59	59
MIN Ⓢ Ⓓ n	n = 1	43	32	16		17	7.1	7.1	7.1
MINP Ⓢ Ⓓ n	n = 96	436	326	163		159	69	69	69
DMAX Ⓢ Ⓓ n	n = 1	71	53	27		27	12	12	12
DMAXP Ⓢ Ⓓ n	n = 96	427	321	161		181	78	78	78
DMIN Ⓢ Ⓓ n	n = 1	71	53	27		27	12	12	12
DMINP Ⓢ Ⓓ n	n = 96	268	201	101		112	48	48	48
SORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n = 1	43	32	16		16	7.1	7.1	7.1
	n = 96	40*1	30*1	15*1		14	6.2	6.2	6.2
DSORT Ⓢ1 n Ⓢ2 Ⓓ1 Ⓓ2	n = 1	44	33	17		17	7.1	7.1	7.1
	n = 96	43*1	32*1	16*1		16	6.8	6.8	6.8
WSUM Ⓢ Ⓓ n	n = 1	41.5	31.1	15.6		16.4	7.1	7.1	7.1
WSUMP Ⓢ Ⓓ n	n = 96	173.2	129.9	65		68.4	29.5	29.5	29.5
DWSUM Ⓢ Ⓓ n	n = 1	47.9	35.9	18		18.9	8.2	8.2	8.2
DWSUMP Ⓢ Ⓓ n	n = 96	330	247.5	123.8		130.4	56.1	56.1	56.1
FOR n	n = 0	5.2	3.9	2.0		2.3	1.0	1.0	1.0
NEXT	--	8.0	6.0	3.0		3.3	1.4	1.4	1.4
BREAK BREAKP	--	26	19	9.5		11	4.6	4.6	4.6
CALL Pn	文件内指针	5.1	3.8	1.9		2.1	0.88	0.88	0.88
CALLP Pn	公共指针	85	64	32		33	14	14	14
CALL Pn Ⓢ1 到 Ⓢ5 CALLP Pn Ⓢ1 到 Ⓢ5	--	348	261	131		135	58	58	58
RET	返回正常程序	7.5	5.6	2.8		2.9	1.3	1.3	1.3
	返回到其他程序	51	38	19		20	8.5	8.5	8.5
FCALL Pn	文件内指针	8.8	6.6	3.3		3.6	1.6	1.6	1.6
FCALLP Pn	公共指针	48	36	18		20	8.7	8.7	8.7
FCALL Pn Ⓢ1 到 Ⓢ5 FCALLP Pn Ⓢ1 到 Ⓢ5	--	338	254	127		134	57	57	57
ECALL * Pn ECALLP * Pn *: 程序名称	--	187	140	70		77	33	33	33
ECALL * Pn Ⓢ1 到 Ⓢ5 ECALLP * Pn Ⓢ1 到 Ⓢ5 *: 程序名称	--	515	387	144		162	70	70	70
EFCALL * Pn EFCALLP * Pn *: 程序名称	--	188	141	71		78	34	34	34
EFCALL * Pn Ⓢ1 到 Ⓢ5 EFCALLP * Pn Ⓢ1 到 Ⓢ5 *: 程序名称	--	516	388	194		200	86	86	86

*1：表示为执行完指令所需要的扫描时间的延长。

指令	条件（软元件）		处理时间（μs）						
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
COM	--		137	103	52		55	16	16	16
IX	--		31	23	12		12	5.2	5.2	5.2
IXEND	--		12	8.9	4.5		4.7	2.0	2.0	2.0
IXDEV + IXSET	触点数为 1		127	95	46		48	21	21	21
	触点数为 14		238	179	85		93	40	40	40
FIFW	数据点数为 0		27	20	10		11	4.5	4.5	4.5
FIFWP	数据点数为 96		27	20	10		11	4.5	4.5	4.5
FIFR	数据点数为 1		34	25	13		13	5.6	5.6	5.6
FIFRP	数据点数为 96		79	59	30		32	14	14	14
FPOP	数据点数为 1		46	34	17		16	7.0	7.0	7.0
FPOPP	数据点数为 96		46	34	17		16	7.0	7.0	7.0
FINS	数据点数为 0		48	36	18		20	8.4	8.4	8.4
FINSP	数据点数为 96		96	72	36		36	15	15	15
FDEL	数据点数为 1		47	35	18		19	7.5	7.5	7.5
FDELP	数据点数为 96		97	73	37		39	15	15	15
FROM n1 n2 (D) n3 FROMP n1 n2 (D) n3 *1	n3 = 1		253	217	160		--	--	--	--
			252	210	154		--	--	--	--
			--	--	--		47	22	22	22
	n3 = 1000		4514	4286	4150		--	--	--	--
			2855	2127	2038		--	--	--	--
			--	--	--		476	437	437	437
DFRO n1 n2 (D) n3 DFROP n1 n2 (D) n3 *1	n3 = 1		260	221	165		--	--	--	--
			257	214	156		--	--	--	--
			--	--	--		51	24	24	24
	n3 = 500		4543	4271	4082		--	--	--	--
			2883	2129	2064		--	--	--	--
			--	--	--		478	437	437	437
TO n1 n2 (D) n3 TOP n1 n2 (D) n3 *1	n3 = 1		276	217	162		--	--	--	--
			254	211	154		--	--	--	--
			--	--	--		48	20	20	20
	n3 = 1000		4500	4319	4188		--	--	--	--
			2878	2155	2043		--	--	--	--
			--	--	--		479	412	412	412
DTO n1 n2 (D) n3 DTOP n1 n2 (D) n3 *1	n3 = 1		260	221	165		--	--	--	--
			257	216	157		--	--	--	--
			--	--	--		50	23	23	23
	n3 = 500		4471	4315	4198		--	--	--	--
			2819	2172	2062		--	--	--	--
			--	--	--		457	416	416	416
PR	SM7010N	变量 1 个字符	83	62	31		33	11	11	--
		变量 32 个字符	123	92	46		48	18	18	--
	SM7010FF		54	40	20		21	7.8	7.8	--
PRC	--		400	301	151		181	16	16	--
LED	显示时		223	167	84		--	--	--	--
	显示完成时		79	59	30		--	--	--	--

*1 ： 上面的一行表示当使用 A38B/A1S38B 和扩展基板时的处理时间。
中间一行表示使用 A38HB/A1S38HB 时的处理时间。
下面的一行表示当 Q312B 用来执行在插槽 0 中的 QJ71C24 指令的处理时间。
FROM/TO 指令的处理时间会随插槽数和安装的模块数不同而有变化。
(QnCPU/QnHCPU 中的处理时间也会随扩展基板的不同而有变化。)

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
LEDC	显示时	559	420	210		--	--	--	--
	显示完成时	413	310	155		--	--	--	--
LEDR	没有显示 → 没有显示	18	13	6.5		0.40	0.17	0.17	0.17
	LED 指令执行 → 没有显示	205	154	77		103	44	44	44
CHKST	--	15	11	5.5		5.8	2.5	2.5	2.5
CHK	触点 1 无错误	61	46	23		24	10	10	10
	触点 150 无错误	4232	3182	1591		1676	721	721	721
	触点 1 有错误	224	168	84		88	38	38	38
CHKCIR	10 步	15	11	5.5		5.8	2.5	2.5	2.5
SLT	所有的内部软元件	2399	1804	902		--	--	--	--
	文件寄存器 8K 点	7254	5454	2727		--	--	--	--
	SLT 执行完成	15	11	5.5		--	--	--	--
SLTR	--	1.1	0.8	0.4		--	--	--	--
STRA	开始	47	35	18		--	--	--	--
	STRA 执行完成	15	11	5.5		--	--	--	--
STRAR	--	1.1	0.8	0.4		--	--	--	--
PTRA	--	15	11	5.5		--	--	--	--
PTRAR	--	15	11	5.5		--	--	--	--
PTRAEXE PTRAEXEP	运行中	1.6	1.2	0.6		--	--	--	--
	过程跟踪中	169	127	64		--	--	--	--
BINDA BINDAP	(S) = 1	40	30	15		15	6.7	6.7	6.7
	(S) = -32768	60	45	23		24	10	10	10
DBINDA DBINDAP	(S) = 1	63	47	44		43	18	18	18
	(S) = -2147483648	217	163	82		86	37	37	37
BINHA BINHAP	(S) = 1	46	34	17		18	7.7	7.7	7.7
	(S) = $FFFF_H$	48	36	18		19	8.2	8.2	8.2
DBINHA DBINHAP	(S) = 1	59	44	22		23	10	10	10
	(S) = $FFFFFFFF_H$	62	46	23		24	10	10	10
BCDDA BCDDAP	(S) = 1	58	43	22		23	9.8	9.8	9.8
	(S) = 9999	54	40	20		21	8.9	8.9	8.9
DBCDDA DBCDDAP	(S) = 1	61	46	23		22	9.5	9.5	9.5
	(S) = 99999999	75	56	28		29	13	13	13
DABIN DABINP	(S) = 1	133	100	50		57	25	25	25
	(S) = -32768	145	109	55		58	25	25	25
DDABIN DDABINP	(S) = 1	241	181	91		92	40	40	40
	(S) = -2147483648	268	201	101		106	46	46	46
HABIN HABINP	(S) = 1	32	24	12		13	5.8	5.8	5.8
	(S) = $FFFF_H$	38	28	14		15	6.4	6.4	6.4
DHABIN DHABINP	(S) = 1	54	40	20		22	9.5	9.5	9.5
	(S) = $FFFFFFFF_H$	63	47	24		25	11	11	11

指令	条件（软元件）		处理时间（μs）							
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
DABCD DABCDP	Ⓢ = 1		36	27	14		16	6.9	6.9	6.9
	Ⓢ = 9999		42	31	16		17	7.2	7.2	7.2
DDABCD DDABCDP	Ⓢ = 1		63	47	24		25	11	11	11
	Ⓢ = 99999999		75	56	28		29	13	13	13
COMRD COMRDP	--		36	27	14		40	17	17	17
LEN LENP	1 个字符		48	36	18		18	8.0	8.0	8.0
	96 个字符		229	172	86		86	37	37	37
STR STRP	--		132	99	50		53	23	23	23
DSTR DSTRP	--		285	214	107		123	53	53	53
VAL VALP	--		258	194	97		95	41	41	41
DVAL DVALP	--		402	302	151		166	72	72	72
ESTR ESTRP	--		1337	1005	503		564	243	243	243
EVAL EVALP	小数点格式所有 2 位指定		242	182	91		100	43	43	43
	指数格式所有 6 位指定		306	230	115		127	55	55	55
ASC Ⓢ Ⓓ n	n = 1		164	123	62		64	28	28	28
ASCP Ⓢ Ⓓ n	n = 96		780	586	293		289	125	125	125
HEX Ⓢ Ⓓ n	n = 1		161	121	61		60	26	26	26
HEXP Ⓢ Ⓓ n	n = 96		826	621	311		343	148	148	148
RIGHT Ⓢ Ⓓ n	n = 1		131	98	49		49	21	21	21
RIGHTP Ⓢ Ⓓ n	n = 96		354	266	133		131	56	56	56
LEFT Ⓢ Ⓓ n	n = 1		129	97	49		50	21	21	21
LEFTP Ⓢ Ⓓ n	n = 96		354	266	133		131	56	56	56
MIDR MIDRP	--		141	106	53		53	23	23	23
MIDW MIDWP	--		341	256	128		128	55	55	55
INSTR INSTRP	不匹配		156	117	59		58	25	25	25
	匹配	开始	141	106	53		55	24	24	24
		最后	155	116	58		58	25	25	25

指令	条件（软元件）		处理时间（μs）							
			Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
EMOD EMODP	--		1313	987	494		527	227	227	227
EREXP EREXPP	--		4423	3325	1663		1656	713	713	713
SIN SINP	单精度		--	--	--	35	115	50	50	50
	双精度		4921	3700	1850	-	1945	837	--	--
COS COSP	单精度		--	--	--	35	122	53	53	53
	双精度		6462	4858	2429	--	2618	1127	--	--
TAN TANP	单精度		--	--	--	38	123	53	53	53
	双精度		6515	4898	2449	--	2618	1127	--	--
ASIN ASINP	单精度		--	--	--	44	111	48	48	48
	双精度		890	669	335	--	2491	1072	--	--
ACOS ACOSP	单精度		--	--	--	44	115	49	49	49
	双精度		801	602	301	--	2367	1019	--	--
ATAN ATANP	单精度		--	--	--	39	157	68	68	68
	双精度		7818	5878	2939	--	3140	1352	--	--
RAD RADP	单精度		--	--	--	31	17	7.2	7.2	7.2
	双精度		465	349	175	--	24	10	--	--
DEG DEGP	单精度		--	--	--	31	17	7.2	7.2	7.2
	双精度		492	369	185	--	23	9.9	--	--
SQR SQRP	单精度		--	--	--	39	28	12	12	12
	双精度		4520	3398	1699	--	1812	780	--	--
EXP EXPP	单精度	(S) = -10	--	--	--	37	129	56	56	56
		(S) = 1	--	--	--	37				
	双精度	(S) = -10	5871	4414	2207	--	2386	1026	--	--
		(S) = 1	5950	4474	2237	--				
LOG LOGP	单精度	(S) = 1	--	--	--	37	113	49	49	49
		(S) = 10	--	--	--	37				
	双精度	(S) = 1	1191	896	448	--	2146	924	--	--
		(S) = 10	6839	5142	2571	--				
RND RNDP	--		10	7.5	3.8		3.9	1.7	1.7	1.7
SRND SRNDP	--		8.8	6.6	3.3		3.5	1.5	1.5	1.5

指令	条件（软元件）	处理时间（μs）							
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
BSQR BSQRP	Ⓢ = 0	16	12	6.0		6.2	2.7	2.7	2.7
	Ⓢ = 9999	97	73	37		38	16	16	16
BDSQR BDSQRP	Ⓢ = 0	17	13	6.5		6.2	2.7	2.7	2.7
	Ⓢ = 99999999	88	66	33		38	16	16	16
BSIN BSINP	--	30	22	11		12	5.1	5.1	5.1
BCOS BCOSP	--	32	24	12		12	5.2	5.2	5.2
BTAN BTANP	--	30	22	11		12	5.2	5.2	5.2
BASIN BASINP	--	52	39	20		20	8.7	8.7	8.7
BACOS BACOSP	--	53	40	20		21	9.0	9.0	9.0
BATAN BATANP	--	56	42	21		22	9.6	9.6	9.6
LIMIT LIMITP	--	24	18	9.0		10	4.3	4.3	4.3
DLIMIT DLIMITP	--	28	21	11		11	4.7	4.7	4.7
BAND BANDP	--	24	18	9.0		9.8	4.2	4.2	4.2
DBAND DBANDP	--	28	21	11		11	4.9	4.9	4.9
ZONE ZONEP	--	24	18	9.0		9.1	3.9	3.9	3.9
DZONE DZONEP	--	28	21	11		11	4.6	4.6	4.6
RSET RSETP	--	19	14	7.0		6.8	2.9	2.9	2.9
QDRSET QDRSETP	--	322	242	121		205	88	88	88
QCDSET QCDSETP	--	218	164	82		147	63	63	63
DATERD DATERDP	--	36	27	14		13	5.5	5.5	5.5
DATEWR DATEWRP	--	42	31	16		15	6.4	6.4	6.4

指令	条件（软元件）	处理时间（μs）						
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
DATE+ DATE+P	无进位	60	45	23		13	5.4	5.4	5.4
	有进位	60	45	23		13	5.4	5.4	5.4
DATE - DATE - P	无进位	59	44	22		12	5.2	5.2	5.2
	有进位	60	45	23		12	5.2	5.2	5.2
SECOND SECONDP	--	27	20	10		10	4.5	4.5	4.5
HOUR HOURP	--	31	23	12		12	5.2	5.2	5.2
MSG	1 个字符	7.2	5.4	2.7		3.0	1.3	1.3	1.3
	32 个字符	7.4	5.6	2.8		3.0	1.3	1.3	1.3
PKEY	初次	51	38	19		20	8.6	8.6	8.6
	无受理	48	36	18		19	8.2	8.2	8.2
PSTOP PSTOPP	--	122	92	46		79	34	34	34
POFF POFFP	--	120	90	45		79	34	34	34
PSCAN PSCNAP	--	122	92	46		75	32	32	32
PLOW PLOWP	--	124	93	47		80	34	34	--
WDT WDTP	--	12	8.7	4.4		5.9	2.6	2.6	2.6
DUTY	--	1.6	1.2	0.6		9.3	4.0	4.0	4.0
ZRRDB ZRRDBP	--	19	14	6.9		7.9	3.4	3.4	3.4
ZRWRB ZRWRBP	--	21	16	7.8		9.4	4.0	4.0	4.0
ADRSET ADRSETP	--	13	9.3	4.7		4.9	2.1	2.1	2.1
KEY	--	43.4	32.4	16.2		17	7.3	7.3	--
ZPUSH ZPUSHP	--	27.6	20.6	10.3		11	4.7	4.7	4.7
ZPOP ZPOPP	--	12.7	9.5	4.8		5.1	2.2	2.2	2.2
EROMWR EROMWRP	--	62.6	46.7	23.4		--	--	--	--

指令	条件（软元件）	处理时间（μs）							
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
ZCOM	--	4296.6	3206.4	1603.2		691	289	289	289
READ	--	770.6	575.1	287.6		--	--	--	--
SREAD	--	858.9	641.0	320.5		--	--	--	--
WRITE	--	791.9	591.0	295.5		--	--	--	--
SWRITE	--	848.6	633.3	316.6		--	--	--	--
SEND	--	575.7	429.6	214.8		--	--	--	--
RECV	--	375.9	280.5	140.3		--	--	--	--
REQ	--	527.4	393.6	196.8		--	--	--	--
ZNFR	--	982.1	732.9	366.5		--	--	--	--
ZNTO	--	989.3	738.3	369.2		--	--	--	--
ZNRD	MELSECNET/10	598.6	446.7	223.4		--	--	--	--
	MELSECNET (II)	649.2	484.5	242.3		--	--	--	--
ZNWR	MELSECNET/10	614.3	458.4	229.2		--	--	--	--
	MELSECNET (II)	665.6	496.7	248.4		--	--	--	--
RFRP	--	590.9	441.0	220.5		--	--	--	--
RTOP	--	588.8	439.4	219.7		--	--	--	--

(4) QCPU 指令的处理时间（只包括 QCPU 指令）

(a) 功能版本 A 中可用的指令

指令	条件（软元件）		处理时间（μs）							
			Q2A、QA2S	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
UNIRD	--		--	--	--	--	79	34	34	34
TRACE	开始		--	--	--	--	176	76	76	76
	STRA 执行结束		--	--	--	--	6.3	2.7	2.7	2.7
TRACER	--		--	--	--	--	19	8.2	8.2	8.2
SP.FWRITE	--		--	--	--	--	84	36	36	36
SP.FREAD	--		--	--	--	--	82	35	35	35
PLOADP	--		--	--	--	--	58	25	25	--
PUNLOADP	--		--	--	--	--	272	117	117	--
PSWAPP	--		--	--	--	--	308	133	133	--
RBMOV	传送时间 使用 SRAM 卡的文件 R →文件 R 的传送	1 点	--	--	--	--	69	29	29	29
		1000 点	--	--	--	--	580	308	308	308

(b) 功能版本 B 中可用的指令

指令	条件 / 处理点数			处理时间 (μs)							
				Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
COM *1	带 CPU 共享存储区的自动刷新	刷新范围 :2k 字 (0.5k 字被平均分配到各个 CPU)		--	--	--	--	720	660	660	--
		刷新范围 :4k 字 (1k 字被平均分配到各个 CPU)		--	--	--	--	860	730	730	--
	不带 CPU 共享存储区的自动刷新	--		--	--	--	--	43	20	20	20
FROM *1	从其它站的 CPU 共享存储区读取数据	n3 = 1						59	29	29	--
		n3 = 1000		--	--	--	--	530	500	500	--
	从智能功能模块的缓冲存储区读取数据 *2	n3 = 1	主基板	--	--	--	--	51	24	24	--
			扩展基板	--	--	--	--	54	27	27	--
		n3 = 1000	主基板	--	--	--	--	540	480	480	--
			扩展基板	--	--	--	--	1100	1050	1050	--
S.TO	向自站的 CPU 共享存储区内写数据	n3 = 1 (“TO”指令) n4 = 1 (“S.TO”指令)		--	--	--	--	74	33	33	--
		n2 = 256		--	--	--	--	126	54	54	--
S (P). DATERD *3	读取扩展时钟的数据	--		--	--	--	--	25	11	11	11
S (P). DATE+ *3	扩展时钟数据加法运算	--		--	--	--	--	38	17	17	17
S (P). DATE- *3	扩展时钟数据减法运算	--		--	--	--	--	38	17	17	17

*1 : 在一个多 CPU 系统中，如果指令的处理和其它站点的处理重叠，则处理时间增加下列时间中的最大值。

对于只包含主基板单元的系统

(指令处理时间的增加) = 0.54 × (被处理的点数) × (其它站点数)(μs)

对于包括扩展基板的系统

(指令处理时间的增加) = 1.30 × (被处理的点数) × (其它站点数)(μs)

*2 : 在一个多 CPU 系统中，处于自站控制下的智能功能模块的指令处理时间和处于其它站点控制下的智能功能模块的处理时间相同。

*3 : 可用于序列号前五位为 “07032”或以后的产品。

(5) 冗余系统指令 (只用于 Q4ARCPU)

指令	条件 (软元件)	处理时间 (μs)							
		Q2A、Q2AS	Q3A	Q4A、Q2ASH	Q4AR	Qn	QnH	QnPH	QnPRH
S.STMODE	--	--	--	--	22.6	--	--	--	--
S.CGMODE	--	--	--	--	19.4	--	--	--	--
S.TRUCK	--	--	--	--	43.7	--	--	--	--
S.SPREF	缓冲存储区起始地址 =0 当使用 A68AD 时	--	--	--	407.1	--	--	--	--
	缓冲存储区起始地址 =0 当使用 A62DA 时	--	--	--	331.4	--	--	--	--

(6) 冗余系统指令 (对于冗余 CPU)

指令	条件 (软元件)	处理时间 (μs)							
		Q2A、Q2AS	Q3A	Q4A、Q4AR	Q4AR	Qn	QnH	QnPH	QnPRH
SP.CONTSW	--	--	--	--	--	--	--	--	9.6

附录 1.4　通用型 QCPU 的运算处理时间

各指令的运算时间在本页及以后页面的表中列出。

根据指令源和指令目标的特性，运算处理时间会有所不同，因此应将下表中的值做为处理时间的大致标准进行参照。

附录 1.4.1　子集指令处理时间

以下说明子集指令处理时间。

要 点

1. 指令中使用的软元件满足以下 (a) 和 (b) 中的一项时，则适用于下述 (1) 中列出的子集指令处理时间表。
   (a) 位软元件时
   - 指定内部用户软元件。
   - 指定位软元件号为 16 的倍数。
   - 位数指定仅指定 K4。
     (当使用双字数据时指定为 K8。)
   - 不执行变址修饰。

   (b) 字软元件时
   - 使用内部用户软元件。

   当使用不是以上所列出的软元件时，应参照 (2) 中列出增加时间，对各指令的处理时间进行加法运算。
   对于 OUT/SET/RST 指令，由于使用内部用户软元件 F、T(ST)、C 软元件时处理时间各不相同，因此应参照 (3) 中列出的增加时间，对各指令的处理时间进行加法运算。
2. 在通用型 QCPU 中，由于高速缓冲存储器功能的影响，各指令的处理时间不恒定，因此对最小值和最大值进行了记载。

(1) 子集指令处理时间表

类别	指令	条件 (软元件)		处理时间 (μs)					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
顺控指令	LD LDI AND ANI OR ORI LDP LDF ANDP ANDF ORP ORF	当被执行时		0.040		0.020		0.0095	
	OUT	无变化时		0.040		0.020		0.0095	
		变化时							
	SET RST	当不被执行时		0.040		0.020		0.0095	
		当被执行时	无变化时						
			变化时						

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LD=	导通时		0.120		0.060		0.0285	
		不导通时							
	AND=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LD<>	导通时		0.120		0.060		0.0285	
		不导通时							
	AND<>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR<>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LD>	导通时		0.120		0.060		0.0285	
		不导通时							
	AND>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LD<=	导通时		0.120		0.060		0.0285	
		不导通时							
	AND<=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR<=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LD<	导通时		0.120		0.060		0.0285	
		不导通时							
	AND<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LD>=	导通时		0.120		0.060		0.0285	
		不导通时							
	AND>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	OR>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDD=	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LDD<>	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD<>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD<>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LDD>	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD>	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LDD<=	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD<=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD<=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LDD<	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	LDD>=	导通时		0.120		0.060		0.0285	
		不导通时							
	ANDD>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						
	ORD>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时						
			不导通时						

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	+ (S) (D)	当被执行时		0.120		0.060		0.0285	
	+ (S1) (S2) (D)	当被执行时		0.160		0.080		0.038	
	- (S) (D)	当被执行时		0.120		0.060		0.0285	
	- (S1) (S2) (D)	当被执行时		0.160		0.080		0.038	
	D + (S) (D)	当被执行时		0.120		0.060		0.0285	
	D + (S1) (S2) (D)	当被执行时		0.160		0.080		0.038	
	D - (S) (D)	当被执行时		0.120		0.060		0.0285	
	D - (S1) (S2) (D)	当被执行时		0.160		0.080		0.038	
	* (S1) (S2) (D)	当被执行时		0.180		0.120		0.057	
	/ (S1) (S2) (D)	当被执行时		0.280		0.220		0.110	
	D * (S1) (S2) (D)	当被执行时		0.260		0.200		0.095	
	D/ (S1) (S2) (D)	当被执行时		0.400		0.340		0.170	
	B + (S) (D)	当被执行时		3.300	8.300	3.300	5.500	3.000	4.100
	B + (S1) (S2) (D)	当被执行时		4.600	6.200	4.600	6.200	4.200	5.900
	B - (S) (D)	当被执行时		3.300	9.000	3.300	4.400	2.900	3.800
	B - (S1) (S2) (D)	当被执行时		4.600	8.200	4.600	6.300	2.900	4.600
	B * (S1) (S2) (D)	当被执行时		4.000	8.200	4.000	4.800	3.400	4.800
	B/ (S1) (S2) (D)	当被执行时		4.200	12.400	4.200	5.700	3.700	5.200
	E + (S) (D)	单精度	(S) = 0、(D) = 0	0.180		0.120		0.057	
			(S) = 2^{127}、(D) = 2^{127}	0.180		0.120		0.057	
	E + (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	0.220		0.140		0.0665	
			(S1) = 2^{127}、(S2) = 2^{127}	0.220		0.140		0.0665	
	E - (S) (D)	单精度	(S) = 0、(D) = 0	0.180		0.120		0.057	
			(S) = 2^{127}、(D) = 2^{127}	0.180		0.120		0.057	
	E - (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	0.220		0.140		0.0665	
			(S1) = 2^{127}、(S2) = 2^{127}	0.220		0.140		0.0665	
	E * (S1) (S2) (D)	单精度	(S1) = 0、(S2) = 0	0.180		0.120		0.057	
			(S1) = 2^{127}、(S2) = 2^{127}	0.180		0.120		0.057	
	E/ (S1) (S2) (D)	单精度	(S1) = 2^{127}、(S2) = 2^{127}	5.100	14.100	4.500	5.700	3.900	4.900
	INC	当被执行时		0.080		0.040		0.019	
	DINC	当被执行时		0.080		0.040		0.019	
	DEC	当被执行时		0.080		0.040		0.019	
	DDEC	当被执行时		0.080		0.040		0.019	
	BCD	当被执行时		0.160		0.120		0.057	
	DBCD	当被执行时		0.240		0.200		0.095	
	BIN	当被执行时		0.100		0.060		0.0285	
	DBIN	当被执行时		0.100		0.060		0.0285	

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	FLT	单精度	ⓢ = 0	0.100		0.100		0.0475	
			ⓢ = 7FFF$_H$	0.140		0.100		0.0475	
	DFLT	单精度	ⓢ = 0	0.140		0.100		0.0475	
			ⓢ = 7FFFFFFF$_H$	0.140		0.100		0.0475	
	INT	单精度	ⓢ = 0	0.140		0.100		0.0475	
			ⓢ = 32766.5	0.140		0.100		0.0475	
	DINT	单精度	ⓢ = 0	0.140		0.100		0.0475	
			ⓢ = 1234567890.3	0.140		0.100		0.0475	
	MOV	--		0.080		0.040		0.019	
	DMOV	--		0.080		0.040		0.019	
	EMOV	--		0.080		0.040		0.019	
	CML	--		0.080		0.040		0.019	
	DCML	--		0.080		0.040		0.019	
	BMOV	n = 1		7.600	16.000	6.300	8.200	5.400	7.000
		n = 96		8.200	16.500	7.100	8.800	5.900	7.600
	FMOV	n = 1		6.200	10.500	5.300	7.600	4.400	6.800
		n = 96		5.300	12.600	5.300	5.900	4.200	4.800
	XCH	--		2.500	6.000	2.500	2.900	1.800	2.300
	DXCH	--		2.800	7.900	2.800	3.700	2.100	2.900
	CJ	--		1.900	10.100	1.800	2.800	1.400	2.400
	SCJ	--		1.900	10.100	1.800	2.800	1.400	2.400
	JMP	--		1.900	10.100	1.800	2.800	1.100	2.400
应用指令	WAND ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	WAND ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	DAND ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	DAND ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	WOR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	WOR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	DOR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	DOR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	WXOR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	WXOR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	DXOR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	DXOR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	WXNR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	WXNR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	
	DXNR ⓢ ⓓ	当被执行时		0.120		0.060		0.0285	
	DXNR ⓢ1 ⓢ2 ⓓ	当被执行时		0.160		0.080		0.038	

类别	指令	条件（软元件）	处理时间（μs）					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
应用指令	ROR (D) n	n = 1	2.300	7.800	2.300	3.100	1.700	2.500
		n = 15	2.400	7.800	2.400	3.100	1.800	2.500
	RCR (D) n	n = 1	2.300	3.900	2.300	3.900	1.700	3.200
		n = 15	2.400	4.100	2.400	4.100	1.700	3.200
	ROL (D) n	n = 1	2.500	4.600	2.400	3.300	1.800	3.200
		n = 15	2.400	4.600	2.400	3.300	1.800	3.200
	RCL (D) n	n = 1	2.400	7.500	2.400	2.700	1.800	2.100
		n = 15	2.500	7.500	2.400	2.800	1.800	2.200
	DROR (D) n	n = 1	2.400	10.300	2.400	3.400	1.900	2.700
		n = 31	2.500	10.300	2.500	3.400	1.900	2.700
	DRCR (D) n	n = 1	2.500	12.700	2.500	4.800	1.900	4.200
		n = 31	2.500	12.700	2.500	4.900	1.900	4.200
	DROL (D) n	n = 1	2.500	11.800	2.500	3.900	1.800	3.200
		n = 31	2.500	11.800	2.500	3.900	1.800	3.300
	DRCL (D) n	n = 1	2.500	5.100	2.500	4.800	1.900	3.800
		n = 31	2.500	5.100	2.500	4.600	1.900	3.800
	SFR (D) n	n = 1	2.400	6.100	2.400	3.900	1.700	2.600
		n = 15	2.300	5.700	2.300	3.900	1.800	2.600
	SFL (D) n	n = 1	2.400	4.300	2.400	4.300	1.800	2.700
		n = 15	2.400	4.300	2.400	4.300	1.800	2.700
	DSFR (D) n	n = 1	3.300	12.000	2.700	4.800	2.200	4.300
		n = 96	32.600	42.200	32.600	35.900	23.900	26.100
	DSFL (D) n	n = 1	3.300	8.200	2.700	4.600	2.100	4.000
		n = 96	32.600	37.700	32.600	35.300	23.700	25.800
	SUM	(S) = 0	3.400	6.700	3.400	4.300	2.900	3.600
		(S) = $FFFF_H$	3.500	6.700	3.500	4.200	2.900	3.600
	SEG	当被执行时	2.100	5.900	2.100	2.800	1.500	2.100
	FOR	--	0.920	4.500	0.920	1.500	0.590	1.200
	CALL Pn	文件内指针	2.700	4.800	2.600	4.000	2.300	3.600
		公共指针	4.400	5.700	3.600	5.300	3.200	4.900
	CALL Pn (S1) 到 (S5)	--	28.700	42.600	28.700	33.400	26.100	29.300

备 注

对于表中未记载上升沿指令（□P）的指令，其处理时间与 ON 时执行的指令相同。

例如 MOVP 指令，WANDP 指令等。

(2) 当使用文件寄存器或多 CPU 共享软元件时，所增加时间的列表。

软元件名称		数据	软元件指定位置	处理时间 (μs)		
				Q02UCPU	Q03UDCPU	Q04/Q06UDHCPU
文件寄存器 (R)	使用标准 RAM 时	位	源	0.100	0.100	0.048
			目标	0.100	0.100	0.038
		字	源	0.100	0.100	0.048
			目标	0.100	0.100	0.038
		双字	源	0.200	0.200	0.095
			目标	0.200	0.200	0.086
	使用 SRAM 卡时 (Q2MEM-1MBS、Q2MEM-2MBS)	位	源	0.220	0.220	0.200
			目标	0.180	0.180	0.162
		字	源	0.220	0.220	0.200
			目标	0.180	0.180	0.162
		双字	源	0.440	0.440	0.399
			目标	0.380	0.380	0.361
	使用 SRAM 卡时 (Q3MEM-4MBS、Q3MEM-8MBS)	位	源	0.160	0.160	0.152
			目标	0.140	0.140	0.133
		字	源	0.160	0.160	0.152
			目标	0.140	0.140	0.133
		双字	源	0.320	0.320	0.304
			目标	0.300	0.300	0.295
文件寄存器 (ZR)	使用标准 RAM 时	位	源	0.120	0.120	0.057
			目标	0.120	0.120	0.048
		字	源	0.120	0.120	0.057
			目标	0.120	0.120	0.048
		双字	源	0.220	0.220	0.105
			目标	0.220	0.220	0.095
	使用 SRAM 卡时 (Q2MEM-1MBS、Q2MEM-2MBS)	位	源	0.240	0.240	0.209
			目标	0.200	0.200	0.171
		字	源	0.240	0.240	0.209
			目标	0.200	0.200	0.171
		双字	源	0.460	0.460	0.409
			目标	0.400	0.400	0.371
	使用 SRAM 卡时 (Q3MEM-4MBS、Q3MEM-8MBS)	位	源	0.180	0.180	0.162
			目标	0.160	0.160	0.143
		字	源	0.180	0.180	0.162
			目标	0.160	0.160	0.143
		双字	源	0.340	0.340	0.314
			目标	0.320	0.320	0.304
模块访问软元件 (Un\G□、U3En\G0 到 G4095)		位	源	12.000	11.700	11.200
			目标	17.300	15.400	15.300
		字	源	9.700	9.460	9.410
			目标	33.000	19.000	19.000
		双字	源	24.200	11.000	10.900
			目标	34.800	18.800	18.700
模块访问软元件 (多 CPU 高速通信区域) (U3En\G10000)		位	源	--	0.220	0.181
			目标	--	0.140	0.105
		字	源	--	0.220	0.181
			目标	--	0.140	0.105
		双字	源	--	0.500	0.437
			目标	--	0.340	0.285
链接直接软元件 (Jn\□)		位	源	32.900	32.700	31.300
			目标	67.300	52.300	29.900
		字	源	37.200	28.500	17.300
			目标	37.000	27.500	14.700
		双字	源	39.500	30.300	18.100
			目标	41.900	30.600	15.700

(3) 当在 OUT/SET/RST 指令中使用 F/T(ST)/C 软元件时，所增加时间的列表。

指令名称	软元件名称	条件		处理时间 (μs)		
				Q02UCPU	Q03UDCPU	Q04/Q06UDHCPU
OUT	F	当不被执行时		2.100	1.940	1.570
		当被执行时	显示时	68.800	39.930	38.090
			显示完成时	61.570	39.750	37.980
	T(ST)、C	当不被执行时		0.155	0.060	0.030
		当被执行时	时间到后	0.120	0.060	0.030
			加法运算时	0.120	0.060	0.030
SET	F	当不被执行时		0.000	0.000	0.000
		当被执行时	显示时	68.560	42.900	40.600
			显示完成时	65.650	39.270	37.900
RST	F	当不被执行时		0.000	0.000	0.000
		当被执行时	显示时	55.800	45.260	36.600
			显示完成时	29.150	19.020	16.190
	T(ST)、C	当不被执行时		0.120	0.060	0.030
		当被执行时		0.120	0.060	0.030

附录 1.4.1 子集指令以外的指令处理时间

下列表格说明子集指令以外的指令处理时间。

(1) 子集指令以外的指令处理时间的列表

类别	指令	条件（软元件）	处理时间 (μs)					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
顺控指令	ANB ORB MPS MRD MPP	--	0.040		0.020		0.0095	
	INV	当不被执行时	0.040		0.020		0.0095	
		当被执行时						
	MEP MEF	当不被执行时	0.040		0.020		0.0095	
		当被执行时						
	EGP EGF	当不被执行时	0.040		0.020		0.0095	
		当被执行时						
	PLS	--	1.300	1.600	1.300	1.600	0.890	1.100
	PLF	--	1.600	1.700	1.500	1.600	0.940	1.200
	FF	当不被执行时	0.080		0.040		0.0185	
		当被执行时	1.200	1.500	1.200	1.500	0.790	0.910
	DELTA	当不被执行时	0.080		0.040		0.0185	
		当被执行时	2.800	3.600	2.800	3.600	2.400	3.200
	SFT	当不被执行时	0.800		0.040		0.0185	
		当被执行时	1.600	6.600	1.600	3.300	1.100	2.700
	MC	--	0.080		0.040		0.0185	
	MCR	--	0.040		0.040		0.0185	
	FEND END	执行错误检查时	175.000	252.000	108.000	130.000	75.800	89.300
		不执行错误检查时	175.000	221.000	107.000	124.000	75.800	89.800
	STOP	--	--		--		--	

类别	指令	条件（软元件）			处理时间（μs）					
					Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值
顺控指令	NOP NOPLF PAGE	--			0.040		0.020		0.0095	
基本指令	LDE=	单精度	导通时		4.700	10.100	3.700	4.700	3.300	4.300
			不导通时		4.700	10.100	3.800	5.000	3.400	4.500
	ANDE=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.200	12.500	3.300	5.800	3.000	5.100
				不导通时	4.400	11.900	3.500	5.600	3.000	5.200
	ORE=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.600	10.800	3.600	4.500	3.200	4.200
				不导通时	4.500	9.800	3.500	4.800	3.200	4.300
	LDE< >	单精度	导通时		4.700	7.700	4.000	4.700	3.600	4.200
			不导通时		4.600	8.200	3.900	4.500	3.500	4.000
	ANDE< >	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.300	14.200	3.300	5.100	3.000	4.800
				不导通时	4.400	14.200	3.500	5.000	3.100	4.600
	ORE< >	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.600	6.700	3.600	6.000	3.300	5.500
				不导通时	4.400	6.600	3.500	5.800	3.100	5.300
	LDE>	单精度	导通时		4.700	13.700	3.800	5.000	3.300	4.600
			不导通时		4.600	13.700	3.700	4.900	3.300	4.400
	ANDE>	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.300	8.100	3.500	4.700	3.100	4.200
				不导通时	4.200	8.100	3.600	4.500	3.100	4.000
	ORE>	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.600	8.500	3.600	5.100	3.300	4.600
				不导通时	4.400	8.100	3.500	4.800	3.200	4.500
	LDE<=	单精度	导通时		4.700	11.100	3.800	5.600	3.400	5.200
			不导通时		4.700	9.600	3.800	5.600	3.400	5.100
	ANDE<=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.100	7.800	3.200	4.600	2.800	4.200
				不导通时	4.400	8.200	3.500	5.000	3.100	4.500
	ORE<=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.500	10.300	3.700	5.800	3.400	5.400
				不导通时	4.400	9.800	3.800	5.700	3.300	5.300
	LDE<	单精度	导通时		4.700	11.500	4.000	5.400	3.500	4.900
			不导通时		4.700	10.900	4.000	5.200	3.500	4.900
	ANDE<	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.300	9.200	3.400	4.600	3.000	4.200
				不导通时	4.400	9.400	3.500	4.900	3.100	4.400
	ORE<	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.600	10.400	3.600	5.200	3.300	4.900
				不导通时	4.400	9.800	3.400	4.900	3.200	4.500

类别	指令	条件（软元件）			处理时间（μs）					
					Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDE>=	单精度	导通时		4.700	12.200	3.800	6.000	3.300	5.500
			不导通时		4.700	11.800	3.800	5.900	3.400	5.400
	ANDE>=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.100	6.700	3.200	4.800	2.900	4.600
				不导通时	4.400	7.000	3.500	5.400	3.100	5.100
	ORE>=	单精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.600	14.000	3.600	5.200	3.300	4.700
				不导通时	4.500	14.300	3.500	5.200	3.200	4.700
	LDED=	双精度	导通时		4.200	21.000	4.100	7.700	3.500	7.200
			不导通时		5.100	21.900	4.300	8.100	3.800	7.400
	ANDED=	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	3.800	17.800	3.600	7.600	3.200	7.000
				不导通时	4.100	18.100	3.900	7.700	3.400	7.400
	ORED=	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.100	23.800	3.800	8.800	3.400	8.300
				不导通时	4.900	25.500	4.000	9.300	3.700	8.800
	LDED<>	双精度	导通时		5.100	23.500	4.400	8.200	3.900	7.700
			不导通时		4.200	22.600	4.100	7.900	3.500	7.500
	ANDED<>	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.000	18.800	3.800	7.600	3.300	7.200
				不导通时	4.000	18.700	3.800	7.700	3.400	7.300
	ORED<>	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	5.000	25.200	4.100	9.300	3.700	8.900
				不导通时	4.100	23.400	3.800	8.900	3.400	8.400
	LDED>	双精度	导通时		5.100	25.100	4.300	8.100	3.800	7.500
			不导通时		4.200	23.400	4.100	7.800	3.500	7.200
	ANDED>	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	4.000	19.500	3.800	7.700	3.300	7.300
				不导通时	4.100	19.700	4.000	7.900	3.500	7.500
	ORED>	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	5.000	24.200	4.100	9.300	3.700	8.800
				不导通时	4.900	25.800	4.100	9.300	3.700	8.800
	LDED<=	双精度	导通时		4.200	22.500	4.000	8.000	3.500	7.400
			不导通时		4.200	13.500	4.100	9.400	3.600	8.800
	ANDED<=	双精度	当不被执行时		0.120		0.060		0.0825	
			当被执行时	导通时	4.000	19.600	3.800	7.700	3.300	7.200
				不导通时	4.100	19.700	3.900	7.700	3.500	7.400
	ORED<=	双精度	当不被执行时		0.120		0.060		0.0825	
			当被执行时	导通时	5.000	26.300	4.100	9.600	3.700	9.200
				不导通时	5.000	25.200	4.100	9.600	3.700	9.200

类别	指令	条件（软元件）			处理时间（μs）					
					Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
					最小值	最大值	最小值	最大值	最小值	最大值
基本指令	LDED<	双精度	导通时		5.100	25.000	4.300	8.300	3.800	7.600
			不导通时		4.200	24.100	3.700	7.900	3.500	7.400
	ANDED<	双精度	当不被执行时		0.120		0.060		0.0825	
			当被执行时	导通时	4.000	19.400	3.800	7.800	3.300	7.300
				不导通时	4.100	19.700	3.900	7.900	3.400	3.900
	ORED<	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	5.000	25.100	4.100	9.600	3.700	9.200
				不导通时	5.000	25.100	4.000	9.600	3.700	9.200
	LDED>=	双精度	导通时		4.200	13.100	4.100	9.600	3.600	9.000
			不导通时		4.300	13.100	4.100	9.600	3.600	8.900
	ANDED>=	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	3.900	19.500	3.800	7.900	3.400	7.400
				不导通时	4.100	19.800	3.900	8.100	3.400	7.500
	ORED>=	双精度	当不被执行时		0.120		0.060		0.0285	
			当被执行时	导通时	5.000	25.100	4.100	9.600	3.700	9.200
				不导通时	4.200	18.500	4.000	7.200	3.600	6.600
	LD$=	导通时			5.500	14.900	5.300	8.900	4.700	8.100
		不导通时			5.500	15.600	4.700	9.000	4.200	8.200
	AND$=	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		5.200	13.800	4.400	6.800	3.900	6.400
			不导通时		5.300	14.500	4.500	6.700	4.000	6.300
	OR$=	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		5.500	14.900	5.100	8.200	4.200	7.600
			不导通时		5.300	14.600	5.000	8.100	4.000	7.200
	LD$< >	导通时			5.600	15.200	4.800	8.100	4.300	7.500
		不导通时			5.600	15.400	4.700	8.400	4.200	7.800
	AND$< >	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		4.300	21.500	4.300	5.500	4.100	5.100
			不导通时		4.500	23.400	4.500	5.900	4.400	5.400
	OR$< >	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		5.400	17.700	5.200	7.300	4.100	6.700
			不导通时		5.300	19.400	5.100	7.200	4.100	6.700
	LD$>	导通时			6.400	19.200	4.800	7.200	4.300	6.700
		不导通时			5.600	20.100	4.800	7.700	4.200	7.100
	AND$>	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		4.500	15.400	4.500	7.100	4.000	6.700
			不导通时		4.600	15.300	4.600	7.600	4.300	7.000
	OR$>	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		5.400	20.000	5.100	6.800	4.300	6.200
			不导通时		5.400	22.100	5.200	7.200	4.300	6.600
	LD$<=	导通时			5.800	12.800	5.000	6.300	4.400	5.700
		不导通时			6.300	13.900	4.800	6.400	4.200	5.800
	AND$<=	当不被执行时			0.120		0.060		0.0285	
		当被执行时	导通时		6.000	16.000	4.600	7.600	4.100	7.200
			不导通时		6.100	16.200	4.700	7.700	4.200	7.300

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	OR$<=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时	4.700	14.600	4.700	7.700	4.400	7.200
			不导通时	4.600	14.400	4.600	7.600	4.400	7.100
	LD$<	导通时		4.800	17.000	4.800	8.100	4.500	7.500
		不导通时		5.500	18.000	5.000	8.300	4.500	7.900
	AND$<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时	5.900	13.400	4.500	7.100	4.000	6.600
			不导通时	6.200	14.500	4.900	7.500	4.400	7.100
	OR$<	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时	6.200	18.700	5.100	7.800	4.100	7.200
			不导通时	5.400	19.700	5.000	8.100	4.100	7.600
	LD$>=	导通时		4.800	10.000	4.800	6.700	4.500	6.200
		不导通时		5.500	11.200	5.000	6.700	4.400	6.300
	AND$>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时	4.400	21.600	4.400	6.800	4.100	6.300
			不导通时	4.500	21.800	4.500	7.000	4.200	6.600
	OR$>=	当不被执行时		0.120		0.060		0.0285	
		当被执行时	导通时	5.400	15.400	5.400	6.600	4.100	5.800
			不导通时	5.300	15.300	5.300	6.300	4.100	5.700
	BKCMP = (S1) (S2) (D) n	n = 1		8.200	22.600	8.200	10.700	7.500	10.000
		n = 96		57.400	72.500	57.400	61.800	46.400	48.700
	BKCMP<> (S1) (S2) (D) n	n = 1		8.200	22.500	8.200	10.700	7.500	10.000
		n = 96		59.500	74.500	59.500	63.300	45.600	50.400
	BKCMP> (S1) (S2) (D) n	n = 1		8.200	23.100	8.200	10.800	7.500	10.100
		n = 96		59.500	74.400	59.500	63.400	47.700	50.500
	BKCMP<= (S1) (S2) (D) n	n = 1		8.200	22.500	8.200	10.600	7.500	10.000
		n = 96		57.400	72.400	57.400	61.700	46.400	49.000
	BKCMP< (S1) (S2) (D) n	n = 1		8.300	23.000	8.300	10.600	7.500	10.000
		n = 96		59.500	74.500	59.500	63.600	47.600	50.500
	BKCMP>= (S1) (S2) (D) n	n = 1		8.200	22.500	8.200	10.900	7.500	10.000
		n = 96		57.400	72.400	57.400	62.000	46.400	48.900
	DB + (S) (D)	当被执行时		4.900	7.500	4.900	7.000	4.600	6.400
	DB + (S1) (S2) (D)	当被执行时		5.200	11.000	5.200	7.300	4.800	6.700
	DB - (S) (D)	当被执行时		4.900	10.200	4.900	6.600	4.700	6.000
	DB - (S1) (S2) (D)	当被执行时		5.200	8.600	5.200	7.500	4.800	6.600
	DB * (S1) (S2) (D)	当被执行时		8.300	22.200	8.300	12.100	8.100	11.600
	DB/ (S1) (S2) (D)	当被执行时		6.100	19.200	6.100	9.100	5.800	8.800

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ED + (S) (D)	双精度	(S) = 0、(D) = 0	4.800	16.800	4.800	8.000	4.300	7.200
			(S) = 2^{1023}、(D) = 2^{1023}	4.800	16.800	4.800	8.000	4.300	7.200
	ED + (S1) (S2) (D)	双精度	(S1) = 0、(S2) = 0	7.100	20.100	5.500	9.800	4.800	9.200
			(S1) = 2^{1023}、(S2) = 2^{1023}	7.100	20.100	5.500	9.800	4.800	9.200
	ED - (S) (D)	双精度	(S) = 0、(D) = 0	5.000	17.300	5.000	8.200	4.400	7.500
			(S) = 2^{1023}、(D) = 2^{1023}	5.000	17.300	5.000	8.200	4.400	7.500
	ED - (S1) (S2) (D)	双精度	(S1) = 0、(S2) = 0	6.000	16.300	4.400	8.100	3.800	7.500
			(S1) = 2^{1023}、(S2) = 2^{1023}	6.000	16.300	4.400	8.100	3.800	7.500
	ED * (S1) (S2) (D)	双精度	(S1) = 0、(S2) = 0	10.500	22.300	5.800	9.500	5.100	8.800
			(S1) = 2^{1023}、(S2) = 2^{1023}	10.500	22.300	5.800	9.500	5.100	8.800
	ED / (S1) (S2) (D)	双精度	(S1) = 2^{1023}、(S2) = 2^{1023}	7.500	27.200	6.600	10.600	5.900	10.000
	BK + (S1) (S2) (D) n	n = 1		12.100	19.700	9.100	11.200	8.500	10.600
		n = 96		61.700	69.300	60.700	62.900	44.600	47.000
	BK - (S1) (S2) (D) n	n = 1		12.100	20.600	9.700	12.000	8.900	11.300
		n = 96		61.700	70.200	61.300	63.600	45.600	47.900
	$ + (S) (D)	--		14.400	34.000	8.800	14.600	8.100	13.900
	$ + (S1) (S2) (D)	--		9.200	22.900	7.300	11.100	6.500	10.300
	FLTD	双精度	(S) = 0	4.000	8.900	2.300	5.000	1.800	4.700
			(S) = $7FFF_H$	3.400	9.000	2.500	5.200	2.200	4.800
	DFLTD	双精度	(S) = 0	4.100	10.800	2.400	5.200	2.000	4.900
			(S) = $7FFFFFFF_H$	3.600	10.800	2.700	5.400	2.300	5.100
	INTD	双精度	(S) = 0	3.500	9.300	2.700	4.100	2.200	4.100
			(S) = 32766.5	5.100	19.500	3.700	5.900	3.200	5.600
	DINTD	双精度	(S) = 0	2.600	6.800	2.600	3.900	2.200	3.400
			(S) = 1234567890.3	3.400	11.700	3.400	5.600	3.000	5.100
	DBL	当被执行时		2.700	3.800	2.700	3.400	2.300	2.700
	WORD	当被执行时		2.900	7.000	2.900	4.300	2.600	3.600
	GRY	当被执行时		2.700	6.100	2.700	3.900	2.300	3.400
	DGRY	当被执行时		2.900	4.600	2.900	3.500	2.500	3.000
	GBIN	当被执行时		4.000	8.200	4.000	4.800	3.800	4.300
	DGBIN	当被执行时		5.500	8.000	5.500	6.100	5.000	5.900
	NEG	当被执行时		2.400	4.100	2.400	3.900	2.000	3.300
	DNEG	当被执行时		2.500	4.300	2.500	3.700	2.500	3.300

类别	指令	条件（软元件）	处理时间（μs）					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
基本指令	ENEG	浮点 = 0	2.500	3.400	2.500	3.300	2.300	2.800
		浮点 = -1.0	2.700	4.500	2.700	4.500	2.500	3.900
	EDNEG	浮点 = 0	2.200	3.500	2.200	3.500	1.800	3.100
		浮点 = -1.0	2.400	3.500	2.400	3.500	1.900	3.000
	BKBCD (S) (D) n	n = 1	9.700	22.000	6.600	8.900	5.900	8.200
		n = 96	74.200	86.500	71.300	74.100	61.000	63.400
	BKBIN (S) (D) n	n = 1	8.900	16.300	6.500	9.800	5.600	9.300
		n = 96	58.500	65.100	56.300	59.500	49.200	52.500
	ECON	--	4.300	6.800	2.600	5.400	2.100	4.500
	EDCON	--	2.800	5.400	2.800	5.400	2.500	5.400
	EDMOV	--	3.200	7.800	2.300	5.500	1.700	5.000
	$MOV	传送的字符串 = 0	4.500	13.900	4.000	6.300	3.400	5.600
		传送的字符串 = 32	15.400	17.500	14.600	16.500	11.400	13.300
	BXCH (D1) (D2) n	n = 1	8.700	15.200	6.200	7.900	5.500	7.300
		n = 96	67.200	74.000	67.000	68.800	47.300	49.300
	SWAP	--	2.400	2.700	2.400	2.700	1.900	2.200
	GOEND	--	0.500		0.500		0.500	
	DI	--	1.800	2.200	1.800	2.200	1.500	1.800
	EI	--	3.100	3.800	3.100	3.800	3.000	3.300
	IMASK	--	9.800	25.000	9.800	13.300	7.200	10.500
	IRET	--	1.000		1.000		1.000	
	RSF X n	n = 1	4.300	16.100	4.200	5.900	3.700	5.600
		n = 96	11.400	23.700	11.400	13.800	10.700	12.400
	RSF Y n	n = 1	3.800	10.000	3.800	4.800	3.400	4.800
		n = 96	8.500	15.200	8.500	9.500	8.100	8.900
	UDCNT1	--	1.000	2.000	0.900	1.500	0.500	0.983
	UDCNT2	--	1.000	4.000	0.900	1.700	0.600	1.300
	TTMR	--	3.900	6.100	3.900	6.100	3.400	5.400
	STMR	--	7.200	30.000	6.800	13.500	5.800	12.500
	ROTC	--	15.200	16.100	9.000	10.500	8.000	9.400
	RAMP	--	5.900	18.300	5.900	8.800	5.200	8.400
	SPD	--	1.000	2.800	0.900	1.900	0.500	1.400
	PLSY	--	3.500	4.700	1.900	2.200	1.500	1.800
	PWM	--	3.400	3.400	1.200	1.600	0.900	1.200
	MTR	--	20.500	28.400	10.400	19.800	9.400	10.000

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
应用指令	BKAND (S1) (S2) (D) n	n = 1		12.100	20.100	9.000	11.700	8.300	11.000
		n = 96		57.400	63.200	57.400	63.100	43.800	47.300
	BKOR (S1) (S2) (D) n	n = 1		7.700	13.200	7.700	10.000	7.700	9.500
		n = 96		57.400	62.800	57.400	61.900	44.300	45.800
	BKXOR (S1) (S2) (D) n	n = 1		7.800	13.200	7.800	10.100	7.300	9.200
		n = 96		57.300	62.800	57.300	61.500	43.800	45.800
	BKXNR (S1) (S2) (D) n	n = 1		7.800	14.100	7.800	9.600	7.600	8.900
		n = 96		57.400	62.900	57.400	61.400	43.900	45.300
	BSFR (D) n	n = 1		3.700	6.300	3.700	5.400	3.200	4.800
		n = 96		10.200	12.800	6.900	9.000	5.800	7.700
	BSFL (D) n	n = 1		4.500	8.900	4.100	5.900	3.400	5.100
		n = 96		10.100	14.300	7.100	9.100	6.000	7.900
	BSET (D) n	n = 1		3.000	3.400	3.000	3.400	2.500	2.800
		n = 15		3.000	3.500	3.000	3.500	2.500	2.800
	BRST (D) n	n = 1		3.000	3.400	3.000	3.400	2.600	2.800
		n = 15		3.000	3.400	3.000	3.400	2.500	2.800
	TEST	当被执行时		4.400	6.900	4.400	5.300	3.700	4.700
	DTEST	当被执行时		4.500	7.000	4.500	5.400	3.900	4.800
	BKRST (S) n	n = 1		4.300	5.200	4.300	4.600	3.700	4.100
		n = 96		6.500	13.200	6.000	6.800	5.100	6.000
	SER (S1) (S2) (D) n	n = 1	所有的都匹配	5.000	5.300	4.900	5.300	4.200	4.600
			一个都不匹配	5.000	5.300	5.000	5.300	4.200	4.600
		n = 96	所有的都匹配	32.300	35.900	32.300	32.900	25.900	26.300
			一个都不匹配	32.400	35.900	32.400	32.900	25.900	26.300
	DSER (S1) (S2) (D) n	n = 1	所有的都匹配	6.800	10.200	6.100	6.500	5.400	5.700
			一个都不匹配	6.800	10.200	6.200	6.600	5.500	5.900
		n = 96	所有的都匹配	52.800	56.300	52.800	54.200	41.200	41.800
			一个都不匹配	52.800	56.300	52.800	54.200	41.200	41.800
	DSUM (S) (D)	(S) = 0		3.700	4.100	3.700	4.100	3.300	3.600
		(S) = FFFFFFFF$_H$		3.800	4.100	3.800	4.100	3.200	3.700
	DECO (S) (D) n	n = 2		6.000	16.400	6.000	7.500	5.300	6.900
		n = 8		8.100	15.200	8.100	9.300	6.800	7.800
	ENCO (S) (D) n	n = 2	M1 = ON	5.300	6.300	5.300	5.700	4.700	5.100
			M4 = ON	5.200	6.200	5.200	5.700	4.600	5.000
		n = 8	M1 = ON	10.400	17.900	10.400	11.400	9.000	10.000
			M256 = ON	5.700	13.300	5.700	6.800	5.100	6.100

类别	指令	条件（软元件）	处理时间 (μs)					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
应用指令	DIS (S) (D) n	n = 1	5.000	10.900	4.400	5.300	3.800	4.600
		n = 4	5.400	11.300	4.800	5.700	4.000	5.000
	UNI (S) (D) n	n = 1	5.500	8.900	5.000	5.300	3.500	4.800
		n = 4	6.200	9.600	5.600	6.000	4.000	5.100
	NDIS	当被执行时	11.000	16.300	11.000	13.100	11.000	13.200
	NUNI	当被执行时	10.600	16.000	10.600	12.700	7.300	13.200
	WTOB (S) (D) n	n = 1	5.000	6.300	5.000	6.500	4.400	5.800
		n = 96	36.000	36.200	36.000	38.400	28.200	29.300
	BTOW (S) (D) n	n = 1	5.100	5.600	5.100	6.100	4.600	5.500
		n = 96	29.900	30.400	29.900	32.000	22.800	23.800
	MAX (S) (D) n	n = 1	4.300	6.500	4.300	6.900	4.000	6.100
		n = 96	32.000	34.300	31.200	33.500	24.700	27.000
	MIN (S) (D) n	n = 1	4.400	5.300	4.400	6.800	4.000	6.000
		n = 96	30.300	31.700	30.300	34.800	26.500	28.300
	DMAX (S) (D) n	n = 1	4.800	14.200	4.800	9.100	4.800	8.100
		n = 96	56.400	68.000	56.400	62.200	47.100	49.600
	DMIN (S) (D) n	n = 1	4.800	9.300	4.800	6.800	4.300	5.900
		n = 96	55.400	62.800	55.400	60.200	45.400	47.400
	SORT (S1) n (S2) (D1) (D2)	n = 1	6.200	12.200	6.200	9.300	5.600	8.800
		n = 96	6.200	13.100	6.200	9.400	5.600	8.600
	DSORT (S1) n (S2) (D1) (D2)	n = 1	6.200	10.500	6.200	9.300	5.600	8.200
		n = 96	6.100	10.500	6.100	9.100	5.600	8.400
	WSUM (S) (D) n	n = 1	4.800	6.200	4.800	6.200	4.200	5.500
		n = 96	26.900	28.700	26.900	28.700	21.300	22.300
	DWSUM (S) (D) n	n = 1	5.500	7.000	5.500	7.000	4.800	6.100
		n = 96	53.000	56.300	53.000	56.300	42.700	44.000
	NEXT	--	0.620	0.770	0.620	0.770	0.390	0.600
	BREAK	--	21.300	17.900	10.400	5.500	9.100	5.000
	RET	返回正常程序	2.000	3.000	2.000	3.000	1.600	2.600
		返回其它程序	2.300	4.900	2.300	3.700	2.000	3.100
	FCALL Pn	文件内指针	3.300	5.300	3.100	4.400	2.700	3.600
		公共指针	4.900	6.600	4.000	5.700	3.600	5.100
	FCALL Pn (S1) 到 (S5)	--	19.800	23.700	19.300	21.500	16.500	18.600
	ECALL * Pn *: 程序名称	--	75.700	134.000	70.300	82.300	65.900	77.600
	ECALL * Pn (S1) 到 (S5) *: 程序名称	--	109.000	173.000	101.000	114.000	91.800	105.000
	EFCALL * Pn *: 程序名称	--	76.200	134.000	70.700	82.800	66.200	78.100
	EFCALL * Pn (S1) 到 (S5) *: 程序名称	--	90.500	170.000	86.500	107.000	78.800	91.600
	XCALL	--	3.800	6.400	3.800	5.700	3.700	5.200

附 附录

类别	指令	条件（软元件）	处理时间（μs）					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
应用指令	COM	当只选择 I/O 刷新时	4.500	6.400	4.500	6.400	4.400	5.800
		当只选择 CC-Link 刷新时（主站侧）	10.600	32.100	10.400	25.100	9.800	25.100
		当只选择 CC-Link 刷新时（本地站侧）	28.500	56.900	21.800	28.800	21.500	28.200
		当只选择 MELSECNET/H 刷新时（控制站侧）	75.000	92.200	27.000	31.600	26.600	30.900
		当只选择 MELSECNET/H 刷新时（常规站侧）	31.400	52.400	19.700	26.000	19.200	25.100
		当只选择智能自动刷新时	7.100	9.700	7.100	9.700	7.100	8.900
		当只选择组外 I/O 时（仅输入）	15.600	42.600	1.500	6.800	1.500	6.300
		当只选择组外 I/O 时（仅输出）	25.500	105.000	16.500	31.500	14.400	30.100
		当只选择组外 I/O 时（两种 I/O 方式）	25.500	103.000	17.400	31.100	14.000	30.100
		当只选择多 CPU 高速通信区域刷新时	--		1.100	6.400	0.800	5.000
		当只选择与外围设备通讯时	11.400	18.900	7.100	5.600	6.600	9.600
	FIFW	数据点数为 = 0	3.700	10.100	3.700	5.300	3.200	4.600
		数据点数为 = 96	3.800	5.200	3.800	4.400	3.300	3.800
	FIFR	数据点数为 = 0	4.400	5.800	4.300	5.000	3.800	4.400
		数据点数为 = 96	33.500	35.200	33.500	35.500	24.800	25.700
	FPOP	数据点数为 = 0	4.400	10.800	4.300	5.900	3.800	5.300
		数据点数为 = 96	4.400	10.800	4.300	5.900	3.700	5.400
	FINS	数据点数为 = 0	5.000	10.700	4.800	5.900	3.700	5.300
		数据点数为 = 96	4.400	10.900	4.300	5.900	3.700	5.300
	FDEL	数据点数为 = 0	4.900	11.300	4.900	6.500	4.200	5.800
		数据点数为 = 96	34.200	35.900	34.200	35.900	25.400	25.900
	FROM n1 n2 (D) n3	n3 = 1	12.200	36.300	10.500	16.900	10.000	16.400
		n3 = 1000	465.000	539.000	423.000	455.000	421.000	447.000
	DFRO n1 n2 (D) n3	n3 = 1	11.300	42.200	11.300	18.500	10.800	18.000
		n3 = 500	457.000	535.000	422.000	451.000	421.000	450.000
	TO n1 n2 (S) n3	n3 = 1	8.400	22.600	8.400	16.800	8.100	16.800
		n3 = 1000	483.000	539.000	466.000	498.000	466.000	494.000
	DTO n1 n2 (S) n3	n3 = 1	8.800	29.600	8.800	17.400	8.300	16.900
		n3 = 500	476.000	537.000	465.000	497.000	465.000	497.000
	LEDR	不显示→不显示	5.100	5.100	2.400	2.600	1.900	2.000
		LED 指令执行→不显示	35.700	89.200	28.100	39.400	24.400	35.800
	BINDA (S) (D)	(S) = 1	4.900	6.500	4.900	6.500	4.300	5.600
		(S) = -32768	7.200	8.700	7.200	8.700	6.500	8.000
	DBINDA (S) (D)	(S) = 1	5.700	7.100	5.700	7.100	4.900	6.300
		(S) = -2147483648	10.400	12.200	10.400	12.000	9.600	11.000

类别	指令	条件（软元件）	处理时间（μs）					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
应用指令	BINHA (S) (D)	(S) = 1	4.400	5.900	4.400	5.900	3.800	5.200
		(S) = FFFF$_H$	4.400	5.800	4.400	5.800	3.700	5.200
	DBINHA (S) (D)	(S) = 1	5.200	6.700	5.200	6.700	4.600	6.000
		(S) = FFFFFFFF$_H$	5.100	6.500	5.100	6.500	4.600	6.000
	BCDDA (S) (D)	(S) = 1	4.300	5.800	4.300	5.800	3.600	5.000
		(S) = 9999	4.700	6.100	4.700	6.100	4.100	5.400
	DBCDDA (S) (D)	(S) = 1	4.800	6.300	4.800	6.300	4.000	5.500
		(S) = 99999999	5.600	7.100	5.600	7.100	4.900	6.300
	DABIN (S) (D)	(S) = 1	6.500	9.000	6.500	8.500	5.800	7.800
		(S) = -32768	6.300	8.900	6.300	8.300	5.600	7.700
	DDABIN (S) (D)	(S) = 1	9.400	12.000	9.400	11.500	8.500	10.500
		(S) = -2147483648	9.100	11.600	9.100	11.200	8.100	10.200
	HABIN (S) (D)	(S) = 1	4.900	7.500	4.900	7.100	4.400	6.400
		(S) = FFFF$_H$	5.100	8.100	5.100	7.300	4.600	6.500
	DHABIN (S) (D)	(S) = 1	6.000	8.500	6.000	8.100	5.300	7.300
		(S) = FFFFFFFF$_H$	6.300	8.900	6.300	8.500	5.600	7.700
	DABCD (S) (D)	(S) = 1	5.000	7.500	5.000	7.100	4.400	6.300
		(S) = 9999	5.000	7.500	5.000	7.100	4.300	6.300
	DDABCD (S) (D)	(S) = 1	6.200	8.800	6.200	8.300	5.500	7.400
		(S) = 99999999	6.200	8.800	6.200	8.300	5.500	7.500
	COMRD	--	97.300	97.400	51.600	52.400	50.900	51.200
	LEN	1 个字符	4.100	6.600	4.100	6.200	3.600	5.500
		96 个字符	19.800	22.400	19.800	22.200	16.800	18.700
	STR	--	6.900	14.400	6.900	11.100	6.600	10.400
	DSTR	--	10.200	20.800	10.200	12.500	9.600	11.500
	VAL	--	9.800	23.900	9.800	14.200	8.900	13.000
	DVAL	--	14.000	33.100	14.000	18.700	12.700	16.800
	ESTR	--	22.100	52.400	18.700	24.100	17.900	23.100

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
应用指令	EVAL	小数点格式全部2位指定		23.300	36.500	23.300	30.400	22.800	29.000
		指数格式全部6位指定		23.300	36.400	23.300	30.500	22.500	29.000
	ASC (S) (D) n	n = 1		5.600	19.700	5.600	9.000	5.400	8.300
		n = 96		30.200	44.700	28.700	32.100	25.200	28.400
	HEX (S) (D) n	n = 1		7.500	23.100	6.000	9.700	5.400	9.000
		n = 96		37.500	53.300	35.600	39.800	31.300	35.000
	RIGHT (S) (D) n	n = 1		7.600	11.400	7.600	9.400	7.300	6.600
		n = 96		36.300	46.000	36.300	40.000	29.200	31.600
	LEFT (S) (D) n	n = 1		6.500	16.100	6.500	8.900	5.900	8.200
		n = 96		36.200	46.200	36.200	39.700	29.200	31.500
	MIDR	--		9.500	19.100	9.500	12.100	8.100	10.300
	MIDW	--		10.300	18.200	10.300	12.000	8.800	10.200
	INSTR	不匹配		19.300	29.000	19.300	21.800	16.600	18.400
		匹配	起始	10.300	20.000	10.300	12.800	9.100	10.900
			最终	51.100	60.800	51.100	54.200	42.700	44.900
	EMOD	--		10.300	15.300	10.300	11.800	9.600	11.000
	EREXP	--		19.300	22.300	19.300	21.000	18.800	20.100
	SIN	单精度		4.500	9.900	4.500	6.200	4.100	5.700
	COS	单精度		4.300	8.200	4.300	6.000	4.000	5.600
	TAN	单精度		5.100	7.200	5.100	7.200	5.100	6.700
	ASIN	单精度		6.100	13.700	6.100	8.900	5.900	8.500
	ACOS	单精度		6.800	11.100	6.800	9.300	6.700	8.900
	ATAN	单精度		4.000	6.900	4.000	6.500	3.900	6.000
	SIND	双精度		9.600	26.000	8.800	14.300	8.500	13.800
	COSD	双精度		10.000	26.900	9.300	15.100	8.800	14.600
	TAND	双精度		11.400	25.300	11.200	16.900	10.800	16.500
	ASIND	双精度		12.100	30.800	12.000	17.100	11.600	16.600
	ACOSD	双精度		11.700	28.000	11.700	16.500	11.200	16.200
	ATAND	双精度		9.700	22.000	9.500	14.200	9.100	13.800
	RAD	单精度		2.500	4.800	2.500	4.800	2.100	4.300
	RADD	双精度		4.100	16.400	4.000	9.600	3.600	9.200
	DEG	单精度		2.500	4.700	2.500	4.700	2.200	4.400
	DEGD	双精度		5.000	18.100	4.300	9.000	3.800	9.000
	SQR	单精度		3.500	9.300	3.000	4.600	2.600	4.300
	SQRD	双精度		5.700	25.400	5.600	11.500	5.200	11.000
	EXP (S) (D)	单精度	(S) = -10	4.000	13.000	4.000	6.100	3.800	5.500
			(S) = 1	4.000	13.000	4.000	6.100	3.800	5.600
	EXPD (S) (D)	双精度	(S) = -10	8.800	27.600	8.700	13.900	8.200	13.500
			(S) = 1	8.500	27.300	8.400	13.600	8.000	13.200
	LOG (S) (D)	单精度	(S) = 1	4.100	8.100	4.100	6.900	3.800	6.400
			(S) = 10	6.200	10.300	5.600	8.200	5.200	7.700
	LOGD (S) (D)	双精度	(S) = 1	9.500	28.300	8.100	13.000	7.700	12.500
			(S) = 10	11.100	29.900	9.700	14.800	9.200	14.300

类别	指令	条件（软元件）	处理时间（μs）					
			Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
			最小值	最大值	最小值	最大值	最小值	最大值
应用指令	RND	--	1.200	2.300	1.200	2.300	0.800	1.800
	SRND	--	1.400	2.400	1.400	2.400	1.100	2.000
	BSQR (S) (D)	(S) = 0	1.800	3.300	1.800	3.300	1.600	2.800
		(S) = 9999	5.100	8.800	5.100	8.800	5.100	8.000
	BDSQR (S) (D)	(S) = 0	1.900	3.700	1.900	3.400	1.500	3.000
		(S) = 99999999	7.500	10.900	7.500	10.200	7.500	9.900
	BSIN	--	8.700	20.200	8.600	15.100	8.100	14.500
	BCOS	--	7.800	14.400	7.800	14.400	7.800	13.700
	BTAN	--	9.000	17.000	9.000	13.800	9.000	13.300
	BASIN	--	12.200	15.100	10.600	13.400	10.100	12.800
	BACOS	--	13.100	14.900	11.600	14.400	11.100	14.100
	BATAN	--	11.400	15.700	9.800	11.700	9.100	10.900
	LIMIT	--	5.200	5.400	2.800	3.100	2.400	2.700
	DLIMIT	--	5.700	5.900	3.200	3.500	2.800	3.000
	BAND	--	5.400	6.300	3.000	4.300	2.700	3.800
	DBAND	--	5.800	6.900	3.600	5.100	3.300	4.600
	ZONE	--	5.200	11.100	3.000	4.700	2.600	4.300
	DZONE	--	5.700	10.800	3.400	5.000	3.000	4.600
	RSET	标准 RAM	3.000	16.400	3.000	6.300	2.700	5.900
		SRAM 卡	3.000	16.400	3.000	6.400	2.600	5.800
	QDRSET	SRAM 卡到标准 RAM	230.000	327.000	120.000	134.000	115.000	134.000
		标准 RAM 到 SRAM 卡	997.000	1066.000	533.000	560.000	520.000	553.000
	QCDSET	SRAM 卡到标准 ROM	525.000	690.000	306.000	346.000	305.000	346.000
		标准 ROM 到 SRAM 卡	490.000	655.000	311.000	342.000	300.000	334.000
	DATERD	--	5.100	14.700	3.200	5.000	2.500	4.200
	DATEWR	--	7.100	23.000	4.900	9.700	4.100	8.900
	DATE +	无进位	6.500	13.100	5.100	8.000	4.700	6.600
		有进位	5.700	21.200	5.700	8.000	4.600	6.500
	DATE -	无进位	6.500	11.500	5.800	8.500	4.600	7.000
		有进位	5.700	17.200	5.700	7.400	4.600	6.500
	SECOND	--	2.600	5.900	2.600	3.900	2.200	3.400
	HOUR	--	3.000	5.300	2.900	4.800	2.400	4.300
	S.DATERD	--	6.400	10.200	5.600	8.000	4.800	7.100
	S.DATE +	无进位	9.700	15.600	9.100	11.700	7.400	10.000
		有进位	9.700	15.700	8.900	11.800	7.400	10.000
	S.DATE -	无进位	10.600	22.900	9.000	12.000	7.400	10.300
		有进位	9.800	22.100	9.000	12.200	7.500	10.200
	PSTOP	--	120.000	249.000	61.400	84.500	56.600	79.800
	POFF	--	121.000	246.000	61.800	84.500	57.200	79.800
	PSCAN	--	126.000	232.000	64.900	84.700	60.100	79.900
	WDT	--	1.300	3.800	1.300	2.700	1.100	2.400
	DUTY	--	4.900	24.300	4.900	10.100	4.800	9.600
	TIMCHK	--	3.800	14.000	3.800	5.300	3.500	4.700
	ZRRDB	标准 RAM 的文件寄存器	2.400	2.600	2.400	2.600	1.800	2.100
		SRAM 卡的文件寄存器	2.500	2.800	2.500	2.800	2.000	2.300

类别	指令	条件（软元件）		处理时间（μs）					
				Q02UCPU		Q03UDCPU		Q04/Q06UDHCPU	
				最小值	最大值	最小值	最大值	最小值	最大值
应用指令	ZRWRB	标准 RAM 的文件寄存器		3.100	3.300	3.000	3.300	2.400	2.700
		SRAM 卡的文件寄存器		3.300	3.600	3.200	3.600	2.600	3.000
	ADRSET	--		4.200	4.900	2.600	3.100	2.100	2.600
	ZPUSH	--		6.900	14.000	6.900	9.200	5.800	7.500
	ZPOP	--		7.500	12.500	7.500	8.800	5.800	6.400
	S.ZCOM	当装载 CC-Link 模块时（主站侧）		20.600	55.000	19.600	26.500	19.300	26.000
		当装载 CC-Link 模块时（本地站侧）		20.600	66.100	19.600	26.500	19.100	26.200
		当装载 MELSECNET/H 模块时（控制站侧）		102.000	180.000	53.500	73.500	53.000	72.700
		当装载 MELSECNET/H 模块时（常规站侧）		29.800	102.000	29.800	41.200	29.800	40.600
	S.RTREAD	--		6.700	33.500	5.900	11.000	5.400	10.500
	S.RTWRITE	--		8.300	26.000	6.700	11.100	6.000	10.400
	UNIRD n1 (D) n2	n2 = 1		4.000	29.100	4.000	8.400	3.700	8.000
		n2 = 16		12.500	37.600	12.500	17.000	12.200	16.600
	TRACE	开始		96.600	103.000	46.600	48.300	43.800	44.700
	TRACER	--		3.800	13.600	3.300	6.800	2.600	6.000
	RBNOV (S) (D) n	使用 SRAM 卡的文件 R→文件 R 的传送时间	1 点	9.200	12.100	9.200	12.100	9.100	11.500
			1000 点	489.000	544.000	489.000	509.000	464.000	487.000
	SP.FWRITE	--		87.000	144.000	4.100	55.600	3.700	53.700
	SP.FREAD	--		127.000	140.000	4.600	54.600	4.000	53.000
	SP.DEVST	--		8.100	98.000	4.500	36.500	4.000	34.500
	S.DEVLD	--		13.000	43.000	11.000	17.800	10.000	17.000
多 CPU 专用指令	S.TO n1 n2 n3 n4 (D)	向自站 CPU 共享存储区写数据	n4 = 1	105.000	105.000	34.700	34.900	33.500	34.400
			n4 = 320	154.000	154.000	85.900	87.600	75.200	75.500
	TO n1 n2 (S) n3	向自站 CPU 共享存储区写数据	n3 = 1	6.800	18.600	5.600	10.200	3.300	9.900
			n3 = 320	38.400	53.000	36.700	42.400	34.300	39.200
	DTO n1 n2 (S) n3	向自站 CPU 共享存储区写数据	n3 = 1	7.000	20.600	5.000	12.100	3.100	10.500
			n3 = 320	65.200	81.000	59.100	66.800	55.300	65.100
	FROM n1 n2 (D) n3	从自站 CPU 共享存储区读数据	n3 = 1	7.300	23.400	3.300	12.700	2.400	9.600
			n3 = 320	51.300	71.700	50.900	64.400	45.200	48.200
		从其它 CPU 共享存储区读数据	n3 = 1	16.600	37.000	11.600	17.700	10.600	13.900
			n3 = 320	153.000	185.000	142.000	160.000	142.000	149.000
			n3 = 1000	432.000	485.000	431.000	463.000	422.000	448.000
	DFRO n1 n2 (D) n3	从自站 CPU 共享存储区读数据	n3 = 1	7.400	24.800	6.700	12.600	2.800	9.900
			n3 = 320	98.700	111.000	96.500	105.000	84.300	96.700
		从其它 CPU 共享存储区读数据	n3 = 1	16.600	47.300	12.900	20.800	12.200	17.100
			n3 = 320	278.000	339.000	277.000	299.000	274.000	291.000
			n3 = 1000	841.000	892.000	838.000	860.000	835.000	857.000

备 注

对于表中未记载上升沿执行指令（□P）的指令，其处理时间与 ON 时执行的指令相同。

例如 WORDP 指令和 TOP 指令等。

(2) 当使用文件寄存器、模块访问软元件、链接直接软元件时所增加时间的列表

软元件名称		数据	软元件指定位置	处理时间 (μs)		
				Q02UCPU	Q03UDCPU	Q04/Q06UDHCPU
文件寄存器 (R)	使用标准 RAM 时	位	源	0.100	0.100	0.048
			目标	0.100	0.100	0.038
		字	源	0.100	0.100	0.048
			目标	0.100	0.100	0.038
		双字	源	0.200	0.200	0.095
			目标	0.200	0.200	0.086
	使用 SRAM 卡时 (Q2MEM-1MBS、Q2MEM-2MBS)	位	源	0.220	0.220	0.200
			目标	0.180	0.180	0.162
		字	源	0.220	0.220	0.200
			目标	0.180	0.180	0.162
		双字	源	0.440	0.440	0.399
			目标	0.380	0.380	0.361
	使用 SRAM 卡时 (Q3MEM-4MBS、Q3MEM-8MBS)	位	源	0.160	0.160	0.152
			目标	0.140	0.140	0.133
		字	源	0.160	0.160	0.152
			目标	0.140	0.140	0.133
		双字	源	0.320	0.320	0.304
			目标	0.300	0.300	0.295
文件寄存器 (ZR)	使用标准 RAM 时	位	源	0.120	0.120	0.057
			目标	0.120	0.120	0.048
		字	源	0.120	0.120	0.057
			目标	0.120	0.120	0.048
		双字	源	0.220	0.220	0.105
			目标	0.220	0.220	0.095
	使用 SRAM 卡时 (Q2MEM-1MBS、Q2MEM-2MBS)	位	源	0.240	0.240	0.209
			目标	0.200	0.200	0.171
		字	源	0.240	0.240	0.209
			目标	0.200	0.200	0.171
		双字	源	0.460	0.460	0.409
			目标	0.400	0.400	0.371
	使用 SRAM 卡时 (Q3MEM-4MBS、Q3MEM-8MBS)	位	源	0.180	0.180	0.162
			目标	0.160	0.160	0.143
		字	源	0.180	0.180	0.162
			目标	0.160	0.160	0.143
		双字	源	0.340	0.340	0.314
			目标	0.320	0.320	0.304
模块访问软元件 (Un\G□、U3En\G0 到 G4095)		位	源	12.000	11.700	11.200
			目标	17.300	15.400	15.300
		字	源	9.700	9.460	9.410
			目标	33.000	19.000	19.000
		双字	源	24.200	11.000	10.900
			目标	34.800	18.800	18.700
模块访问软元件 (多 CPU 高速通信区域) (U3En\G10000)		位	源	--	0.220	0.181
			目标	--	0.140	0.105
		字	源	--	0.220	0.181
			目标	--	0.140	0.105
		双字	源	--	0.500	0.437
			目标	--	0.340	0.285
链接直接软元件 (Jn\□)		位	源	32.900	32.700	31.300
			目标	67.300	52.300	29.900
		字	源	37.200	28.500	17.300
			目标	37.000	27.500	14.700
		双字	源	39.500	30.300	18.100
			目标	41.900	30.600	15.700

附录 2　CPU 之间的性能比较

附录 2.1　Q/QnACPU 和 AnNCPU、AnACPU 以及 AnUCPU 之间的比较

附录 2.1.1　可用的软元件

附表 2.1 软元件比较

软元件名称		QCPU			QnACPU	AnUCPU	AnACPU	AnNCPU
I/O 点数*9		Q00J：256 点 Q00：1024 点 Q01：1024 点	Q02 Q02H Q06H Q12H Q25H Q12PH Q25PH Q12PRH Q25PRH Q03UD Q04UDH Q06UDH Q02U　：2048 点	4096 点	—— Q2A：512 点 Q2A-S1：1024 点 Q3A：2048 点 Q4A：4096 点	—— A2U：512 点 A2U-S1：1024 点 A3U：2048 点 A4U：4096 点	—— A2A：512 点 A2A-S1：1024 点 A3A：2048 点 ——	A1N：256 点 A2N：512 点 A2N-S1：1024 点 A3N：2048 点 ——
I/O 软元件点数*8		2048 点*1	8192 点*1			8192 点	与各 CPU 的 I/O 软元件点数相同	
内部继电器		8192 点*1				总共 8192 点	总共 8192 点	总共 2048 点
锁存继电器		2048 点*1	8192 点*1					
步进继电器	顺控程序	——	——					——
	SFC	2048 点*6	8192 点			——	——	——
报警器		1024 点*1	2048 点*1			2048 点	2048 点	256 点
变址继电器		1024 点*1	2048 点*1			——	——	——
链接继电器		2048 点*1	8192 点*1			8192 点	4096 点	1024 点
链接特殊继电器		1024 点	2048 点			56 点	56 点	56 点
定时器		512 点*1	2048 点*1			总共 2048 点	总共 2048 点	总共 256 点
累计定时器		0 点*1						
计数器		512 点*1	1024 点*1			1024 点	1024 点	256 点
数据寄存器		11136 点*1	12288 点*1			8192 点	6144 点	1024 点
链接寄存器		2048 点*1	8192 点*1			8192 点	4096 点	1024 点
链接特殊寄存器		1024 点	2048 点			56 点	56 点	56 点
功能输入		16 点（FX0 到 FXF）*7				——	——	——
功能输出		16 点（FY0 到 FYF）*7				——	——	——
特殊继电器		1000 点	2048 点			256 点	256 点	256 点
功能寄存器		5 点（FD0 到 FD4）				——	——	——
特殊寄存器		1000 点	2048 点			256 点	256 点	256 点
链接直接软元件		由 J□\□ 指定				——	——	——
智能功能模块软元件		由 U□\G□ 指定				——	——	——
变址寄存器	Z	10 点（Z0 到 Z9）	16 点（Z0 到 Z15）			7 点（Z、Z1 到 Z6）	7 点（Z、Z1 到 Z6）	1 point（Z）
	V*2	——	——			7 点（V、V1 到 V6）	7 点（V、V1 到 V6）	1 point（V）
文件寄存器		32768 点 / 块*5 （R0 到 R32767）	32768 点 / 块 （R0 到 R32767）			8192 点 / 块 （R0 到 R8191）	8192 点 / 块 （R0 到 R8191）	8192 点 / 块 （R0 到 R8191）
累加器 r*3		——				2 点	2 点	2 点
嵌套		15 点				8 点	8 点	8 点
指针		300 点	4096 点			256 点	256 点	256 点
中断指针		128 点	256 点	48 点		32 点	32 点	32 点
SFC 块		126*6	320 点			——	——	——
SFC 转移软元件		——	512 点			——	——	——
十进制常数		K－2147483648 到 K2147483647				K－2147483648 到 K2147483647	K－2147483648 到 K2147483647	K－2147483648 到 K2147483647
十六进制常数		H0 到 HFFFFFFFF				H0 到 HFFFFFFFF	H0 到 HFFFFFFFF	H0 到 HFFFFFFFF
实数常数*6		E ± 1.17550-38 到 E ± 3.40282+38				——	——	——
字符串		“QnACPU”，“ABCD”*4				——	——	——

*1: 软元件的点数可以在参数中修改。
*2:QCPU/QnACPU 使用 V 作为变址继电器。
*3: 在 AnNCPU、AnACPU 和 AnUCPU 中使用了累加器的指令与 Q/QnACPU 中指令的格式不同。
*4: 对于 Q00JCPU、Q00CPU 和 Q01CPU，只能通过 $MOV 指令使用。
*5:Q00JCPU 中没有文件寄存器。
*6: 可用于序列号前五位为 04122 或以后的 Q00JCPU、Q00CPU 和 QCPU。
*7: 在程序中只能使用 FX0 到 FX4 和 FY0 到 FY4 的各 5 点。
*8: 可用于程序的点数。
*9: 对实际 I/O 模块的可访问点数。

附录 2.1.2 I/O 控制模式

附表 2.2 I/O 控制模式

I/O 控制模式			QCPU	QnACPU	AnUCPU	AnACPU	AnNCPU
刷新模式			○	○	○	○	○*2
	直接 I/O 模式	部分刷新指令	○	○	○	○	○
		专用指令*1	--	--	○	○	--
		直接访问输入	○	○	--	--	--
		直接访问输出	○	○	--	--	--
直接模式			--	--	--	--	○*2

表中的符号....○: 可用，--: 不可用

*1 : DOUT、DSET 和 SRST 指令是直接输出专用指令。
这里没有用于直接输出的专用指令。
*2 : 在刷新模式和直接模式之间的切换是由 AnNCPU DIP 开关完成的。

附录 2.1.3 可以被指令使用的数据

附表 2.3 可以被指令使用的数据

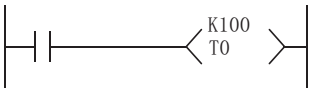
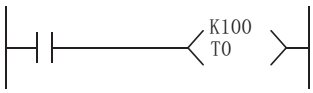
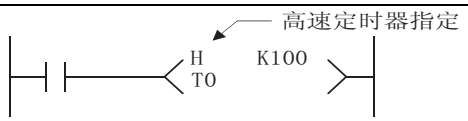
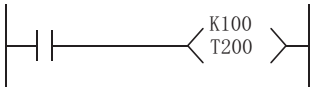
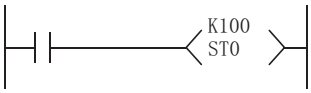
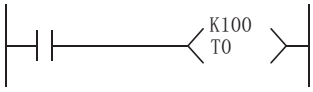
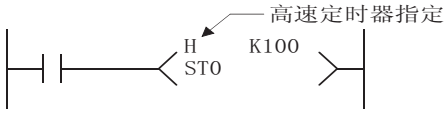
设定数据		QCPU	QnACPU	AnUCPU	AnACPU	AnNCPU
位数据	位软元件	○		○	○	○
	字软元件	○（需要位指定）		--	--	--
字数据	位软元件	○（需要位数指定）		○（需要位数指定）	○（需要位数指定）	○（需要位数指定）
	字软元件	○		○	○	○
双字数据	位软元件	○（需要位数指定）		○（需要位数指定）	○（需要位数指定）	○（需要位数指定）
	字软元件	○		○	○	○
实数数据		○*1	--	○	○	--
字符串数据		○*2	--	--	--	--

表中的符号 .... ○ : 可用，-- : 不可用

*1 : 可用于序列号前五位为 04122 或以后的 Q00JCPU、Q00CPU 和 Q01CPU。
*2 : 对于 Q00JCPU、Q00CPU 和 Q01CPU，只能通过 MOV 指令使用。

附录 2.1.4 定时器比较

附表 2.4 定时器比较

功能		QCPU/QnACPU	AnUCPU	AnACPU	AnNCPU
低速定时器	测量单位	100ms(默认值) 可以在参数中更改测量单位。 QCPU : 1 到 1000ms(1ms 单位) QnACPU: 10 到 1000ms(10ms 单位)	固定为 100ms		
	指定方法				
高速定时器	测量单位	10ms(默认值) 可以在参数中更改测量单位。 QnUCPU: 0.01 到 100ms(0.01ms 单位) QCPU(除 QnUCPU 以外) : 0.1 到 100ms(0.1ms 单位)) QnACPU: 1 到 1000ms(1ms 单位)	固定为 10ms		
	指定方法	 高速定时器设置：通过顺序程序进行。	 高速定时器设置：通过参数进行。		
累计定时器	测量单位	和低速定时器的测量单位相同。	固定为 100ms		
	指定方法				
高速累计定时器	测量单位	与高速定时器的测量单位相同。	None		
	指定方法	 高速定时器设置：通过顺控程序进行。			
设定值的设定范围		1 到 32767	1 到 32767		
对设定值 0 的处理		瞬时 ON	无限大(没有时间溢出)		
变址修饰	触点	允许(只有 Z0 和 Z1 可用)	可以		不可以
	线圈	允许(只有 Z0 和 Z1 可用)	不可以		不可以
	设定值	允许(Z0 到 Z15 可用)*1	不可以		不可以
	当前值	允许(Z0 到 Z15 可用)*1	可以		可以
当前值的更新处理		OUT Tn 指令执行时	END 处理时		
触点的 ON/OFF 处理					

*1 : Q00J/Q00/Q01CPU 可以使用 Z0 到 Z9。
Q02U/Q03UD/Q04UDH/Q06UDHCPU 可以使用 Z0 到 Z19。

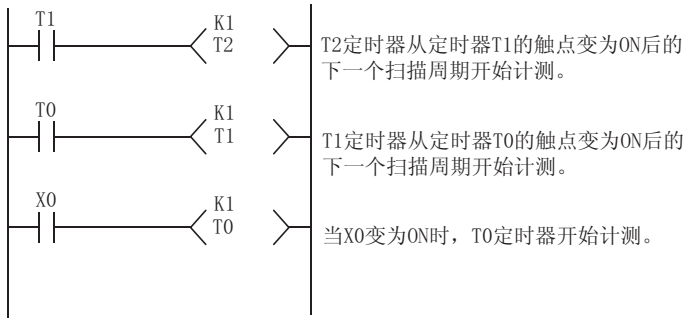
(1) 使用定时器时的注意事项

Q/QnACPU 更新定时器的当前值，并在 OUT T □ 指令被执行时将它们的触点切换为 ON/OFF。因此，如果当定时器线圈变为 ON 时 “当前值 ≥ 设定值”，此定时器的触点变为 ON。
当创建一个程序，在此程序中一个定时器触点的操作将触发其它定时器的操作时，按照定时器的操作顺序来创建程序 - 首先创建较后操作的定时器的程序。
在下列的情况中，如果程序按照定时器操作的顺序创建，则所有的定时器将在同一扫描时刻变为 ON。

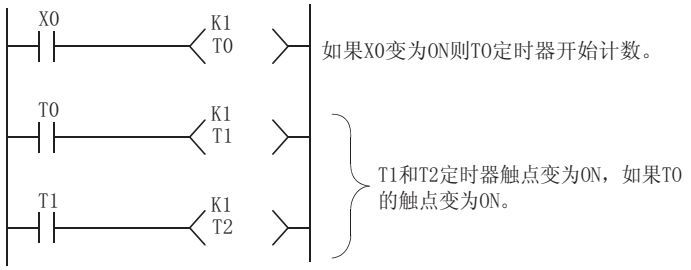
- 对于高速定时器，如果设定值小于一个扫描时间。
- 对于低速定时器，如果 “1” 被设定。

例子

• 对于定时器从 T0 到 T2，按照较后操作的定时器在先的顺序创建程序。

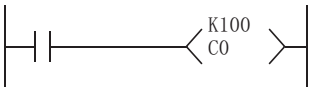
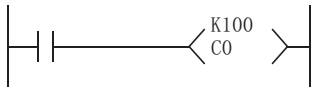


• 对于定时器从 T0 到 T2，按照定时器的执行顺序创建程序。



附录 2.1.5 计数器比较

附表 2.5 计数器比较

功能		QCPU/QnACPU	AnUCPU	AnACPU	AnNCPU
指定方法					
变址修饰	触点	• 可以（只有 Z0 和 Z1 可用）	• 可以		• 不可以
	线圈	• 可以（只有 Z0 和 Z1 可用）	• 不可以		• 不可以
	设定值	• 不可以	• 不可以		• 不可以
	当前值	• 可以 (Z0 到 Z15 可用)*1	• 可以		• 可以
当前值的更新处理		• 当 OUT Cn 指令被执行时	• 当 END 处理被执行时		
触点 ON/OFF 处理					

*1 : Q00J/Q00/Q01CPU 可以使用 Z0 到 Z9。
Q02U/Q03UD/Q04UDH/Q06UDHCPU 可以使用 Z0 到 Z19。

附录 2.1.6 显示指令比较

附表 2.6 显示指令比较

指令	QCPU/QnACPU	AnUCPU	AnACPU	AnNCPU
PR*1	• 当 SM701 变为 OFF：连续输出直到碰到 00H • 当 SM701 变为 ON： 16 字符输出	• 当 M9049 变为 OFF：连续输出直到碰到 00H • 当 M9049 变为 ON： 16 字符输出		
PRC*1	• 当 SM701 变为 OFF：32 字符注释输出 • 当 SM701 变为 ON： 高 16 字符输出	16 字符注释输出		

*1：不可以用于 Q00J/Q00/Q01CPU。

附录 2.1.7 改变定义格式指令（不包括用于 AnACPU 和 AnUCPU 的专用指令）

由于 Q/QnACPU 没有累加器 (A0、A1),AnUCPU、AnACPU 和 AnNCPU 这些使用累加器的指令的格式发生了改变。

附表 2.7 改变表达方式的指令

功能	QCPU/QnACPU		AnUCPU/AnACPU/AnNCPU	
	指令格式	备注	指令格式	备注
16 位循环右移	ROR D n	• D ： 循环移动的数据	ROR n	• 循环移动的数据在 A0 处设定。
	RCR D n	• D ： 循环移动的数据 • 进位标志使用 SM700	RCR n	• 循环移动的数据在 A0 处设定。 • 进位标志使用 M9012。
16 位循环左移	ROL D n	• D ： 循环移动的数据	ROL n	• 循环移动的数据在 A0 处设定。
	RCL D n	• D ： 循环移动的数据 • 进位标志使用 SM700	RCL n	• 循环移动的数据在 A0 处设定。 • 进位标志使用 M9012。
32 位循环右移	DROR D n	• D ： 循环移动的数据	DROR n	• 循环移动的数据在 A0 和 A1 处设定。
	DRCR D n	• D ： 循环移动的数据 • 进位标志使用 SM700	DRCR n	• 循环移动的数据在 A0 和 A1 处设定。 • 进位标志使用 M9012。
32 位循环左移	DROL D n	• D ： 循环移动的数据	DROL n	• 循环移动的数据在 A0 和 A1 处设定。
	DRCL D n	• D ： 循环移动的数据 • 进位标志使用 SM700	DRCL n	• 循环移动的数据在 A0 和 A1 处设定。 • 进位标志使用 M9012。
16 位数据查找	SER S1 S2 D n	• 搜索结果存储在 D 和 D+1 软元件中。	SER S1 S2 n	• 搜索结果存储在 A0 和 A1。
32 位数据查找	DSER S1 S2 D n	• 搜索结果存储在 D 和 D+1 软元件中。	DSER S1 S2 n	• 搜索结果存储在 A0 和 A1。
16 位数据位检查	SUM S D	• 检查结果存储在 D 软元件中。	SUM S	• 检查结果存储在 A0。
16 位数据位检查	DSUM S D	• 检查结果存储在 D 软元件中。	DSUM S	• 检查结果存储在 A0。
分区刷新	RFS D n	• 新增的专用指令 .	SEG D n	• 只有在 M9052 为 ON 时
8 字符 ASCII 转换	$MOV (字符串) D	--	ASC (字符串) D	--
进位标志置位	SET SM700	• 无专用指令	STC	--
进位标志复位	RST SM700	• 无专用指令	CLC	--
跳转到 END 指令	GOEND	• 新增的专用指令	CJ P255	• P255: END 指令指定
CHK 指令 *1	CHKST CHK	• 新增的专用指令	CJ Pn P254 CHK	--

*1: 不能用于 Q00J/Q00/Q01CPU。

附录 2.1.8 AnACPU 和 AnUCPU 专用指令

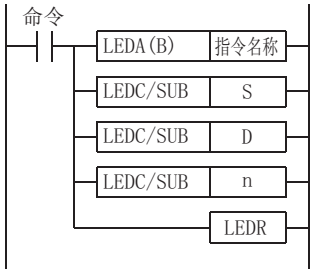
(1) 专用指令的表达方法

在 Q/QnACPU 中，带 AnACPU 或 AnUCPU 的 LEDA、LEDB、LEDC、SUB 和 LEDR 指令的专用指令已经被改变格式，以便和基本指令和应用指令保持一致。

那些由于在 QCPU/QnACPU 中没有相应指令而不能被转换的指令，被转换成 OUT SM1255/OUT SM999(用于 Q00J/Q00/Q01CPU)。

那些已经被转换成 OUT SM1255/OUT SM999 的指令应该被其它指令替换掉或者被删除。

附表 2.8 专用指令的表达方法

QCPU/QnACPU	AnUCPU/AnACPU
	
	*: S、D 和 n 表示被指令使用的数据。

(2) 被改名的的专用指令

用于 AnUCPU 或 AnACPU 的专用指令，它们和基本指令或应用指令使用同样的指令名字，它们在 Q/QnACPU 中会被改名。

附表 2.9 被改名的专用指令

功能	QCPU/QnACPU	AnUCPU/AnACPU
浮动小数点数据加	E+	ADD
浮动小数点数据减	E—	SUB
浮动小数点数据乘	E*	MUL
浮动小数点数据除	E/	DIV
数据分解	NDIS	DIS
数据组合	NUNI	UNI
更新检查模式	CHKCIR、CHKEND	CHK、 CHKEND

附录 2.1.9 只能在一般模式中编程的指令

以下指令只能用于在 QnACPU 的一般模式中编程：

(当程序创建后，程序被编译并被转换成实际的顺序程序。)

- CHKCIR 到 CHKEND(更新检查指令的检查模式)
- IX 到 IXEND(用于整个梯形图的变址修饰)
- IXDEV、 IXSET

附录 3　特殊继电器列表

特殊继电器 SM 是应用固化在可编程控制器中的内部继电器。

由于这个原因，它们不能和普通的内部继电器一样被顺控程序使用。然而，如果需要，它们可以变为 ON 或 OFF 以控制 CPU 模块和远程 I/O 模块。

列表的各项目的说明如附表 3.1 以下所示。.

附表 3.1 特殊继电器列表

条目	条目的功能
号码	• 表示特殊寄存器号
名称	• 表示特殊寄存器的名称
含义	• 表示特殊寄存器的内容
解释	• 更详细地讨论特殊寄存器的内容
设置方式 （设置时间）	• 表示继电器是由系统设定还是由用户设定；如果是由系统，当执行设置时。 〈设置方式〉 S　：由系统设定 U　：用户设定（顺控程序或者来自 GX Developer 的测试操作） S/U　：系统和用户都设定 〈当被设定时〉 只用于指示由系统设定寄存器 每个 END：在每个 END 处理过程中进行设定 初始化：只在初始化处理（当电源变为 ON，或者从 STOP 变为 RUN 时）过程中进行设定 状态改变：只在有状态变化时才进行设定 出错　：在出错发生时进行设定 指令执行：当指令被执行时设定 请求　：只在有用户请求（通过 SM，等软元件。）时才进行设定 系统切换：在执行系统切换时进行设定。
相应的 ACPU M9□□□	• 表示 ACPU 中相应的特殊继电器（M9□□□） （当内容改变时，特殊继电器代表 M9□□□格式改变。不兼容 Q00J/Q00/Q01 和 QnPRH。） • 新增表示特殊继电器是新近增加到 QnACPU 或者 Q 系列 CPU 模块中。
相应的 CPU	表示相应的 CPU 模块型号。 ○　：表示所有的 QnACPU 和 QCPU。 QCPU　：表示所有的 Q 系列 CPU 模块。 Q00J/Q00/Q01：表示基本型 QCPU。 Qn(H)　：表示高性能型 QCPU。 QnPH　：表示过程控制 CPU。 QnPRH　：表示冗余 CPU。 QnU　：表示通用型 QCPU。 QnA　：表示 QnA 系列和 Q2ASCPU 系列。 Rem　：表示 MELSECNET/H 远程 I/O 模块。 每个 CPU 模块的型号：表示相关的特殊 CPU 模块。（例如：Q4AR，Q3A）

对于下列条目的详细信息，请参考下面的手册：

- 网络　→相应网络模块的手册
- SFC　→ QCPU(Q 模式)/QnACPU 编程手册（SFC）

要 点

SD1200 到 SD1255 用于 QnACPU。
这些继电器在 QCPU 中是空闲的。

(1) 诊断信息

附表 3.2 特殊继电器

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM0	检测出错	OFF ：没有出错 ON ：出错	• 如果检测结果为出错则变为 ON。（包括当报警器为 ON 和 CHK 指令检测到出错时） • 如果条件自此以后恢复为正常，仍保持为 ON。	S（出错）	新增	QnA Qn(H) QnPH QnPRH Rem
			• 如果检测结果为出错则变为 ON。（包括报警器为 ON） • 如果自此以后条件恢复为正常，仍保持为 ON。	S（出错）	新增	Q00J/Q00/Q01 QnU
SM1	自检测出错	OFF ：没有自检测出错 ON ：自检测	• 如果检测结果为出错则变为 ON。（不包括当报警器为 ON 或者 CHK 指令检测到出错时） • 如果自此以后条件恢复为正常，则保持为 ON。	S（出错）	M9008	QnA Qn(H) QnPH QnPRH Rem
			• 如果检测结果为出错则变为 ON。（不包括当报警器为 ON 时） • 如果自此以后条件恢复为正常，则保持为 ON。	S 出错）	新增	Q00J/Q00/Q01 QnU
SM5	出错公共信息	OFF ：没有出错公共信息 ON ：出错公共信息	• 当 SM0 是 ON 时，如果有出错公共信息则为 ON	S（出错）	新增	○ Rem
SM16	出错个别信息	OFF ：没有出错个别信息 ON ：出错个别信息	• 当 SM0 是 ON 时，如果有出错个别信息则为 ON	S（出错）	新增	
SM50	出错复位	OFF→ON：出错复位	• 进行出错复位操作	U	新增	
SM51	电池电量不足锁存	OFF ：正常 ON ：出错	• 如果 CPU 模块或者存储卡的电池电压下降到低于额定值，则为 ON。 • 如果此后电池电压恢复为正常，保持为 ON。 • 和 BAT.ALARM/BAT.LED 同步。	S（出错）	M9007	QnA Qn(H) QnPH QnPRH QnU
			• 如果 CPU 模块或者存储卡上的电池电压下降到低于额定值，则为 ON。 • 如果此后电池电压恢复为正常，保持为 ON。 • 和 BAT.LED 同步	S（出错）)	新增	Q00J/Q00/Q01
SM52	电池电量不足	OFF ：正常 ON ：出错	• 和 SM51 相同，但是当电池电压恢复为正常后，接着变为 OFF。	S（出错）	M9006	
SM53	AC/DC DOWN 检测	OFF ：没有检测到 AC/DC DOWN ON ：检测到 AC/DC DOWN	• 如果在 AC 电源模块的过程中发生 20ms 以内的瞬间掉电，变为 ON。当电压被切换到 OFF，然后再切换为 ON 时复位。	S（出错）	M9005	○
			• 如果在使用 DC 电源模块的过程中发生 10ms 以内的瞬间掉电，变为 ON。当电源被切换到 OFF，然后再切换为 ON 时复位。			QCPU
			• 如果在使用 DC 电源模块的过程中发生 1ms 以内的瞬间掉电，变为 ON。当电源切换到 OFF，然后再切换为 ON 时复位。			QnA
SM54	MINI 链接出错	OFF ：正常 ON ：出错	• 即使只在安装的 AJ71PT32(S3) 模块中的一个上检测到 MINI(S3) 链接出错，也变为 ON。 • 如果自此以后条件恢复为正常，则保持为 ON。	S（出错）	M9004	

附表 3.2 特殊继电器

号码	名称	含义	解释		设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM56	操作出错	OFF ：正常 ON ：操作出错	• 当产生操作出错时，变为 ON。 • 如果自此以后情况恢复为正常，仍保持为 ON。		S（出错）	M9011	○
SM60	保险丝熔断检测	OFF ：正常 ON ：模块有熔断的保险丝	• 如果至少有一个保险丝熔断的输出模块存在，变为 ON。 • 如果自此以后情况恢复为正常，仍保持为 ON。 • 即使对远程 I/O 站输出模块，也进行保险丝熔断状态检查。		S（出错）	M9000	○ Rem
SM61	I/O 模块验证出错	OFF ：正常 ON ：出错	• 如果 I/O 模块和通电时注册的状态不同，变为 ON。 • 如果自此以后情况恢复为正常，则保持为 ON。 • 对远程 I/O 站模块也进行 I/O 模块校验。		S（出错）	M9002	
SM62	报警器检测	OFF ：没有检测到 ON ：检测到	• 如果即使只有一个报警器 F 变为 ON，也变为 ON。		S（电源接通或者）	M9009	○
SM80	CHK 检测	OFF ：没有检测到 ON ：检测到	• 如果 CHK 指令检测到出错，变为 ON。 • 即使自此以后情况恢复为正常，保持为 ON。		S（电源接通或者）	新增	
SM90	用于步转移的看门狗定时器的启动（只在存在 SFC 程序时才被激活）	OFF ：不启动（看门狗定时器复位） ON ：启动（看门狗定时器被启动）	对应 SD90	• 当步转移看门狗定时器的测量开始时，变为 ON。 • 当它变为 OFF 时，复位步转移看门狗定时器。	U	M9108	QnA Qn(H) QnPH QnPRH
SM91			对应 SD91			M9109	
SM92			对应 SD92			M9110	
SM93			对应 SD93			M9111	
SM94			对应 SD94			M9112	
SM95			对应 SD95			M9113	
SM96			对应 SD96			M9114	
SM97			对应 SD97			新增	
SM98			对应 SD98			新增	
SM99			对应 SD99			新增	
SM100	串行通讯功能使用标志	OFF ：没有使用串行通讯功能。 ON ：使用了串行通讯功能。	• 在串行通讯设置参数中存储是否使用串行通讯功能的设置		S（电源接通或者复位）	新增	Q00J/Q00/Q01
SM101	通讯协议状态标志	OFF ：GX Developer ON ：MC 协议通讯软元件	• 存储通过 RS-232 接口是通讯的软元件是 GXDeveloper 还是 MC 协议通讯软元件		S(RS232 通讯)		
SM110	协议出错	OFF ：已清除 ON ：异常	• 当在串行通讯功能中，使用了异常协议进行通讯时，变为 ON。 • 如果自此以后情况恢复为正常，仍保持为 ON。		S（出错）		
SM111	通讯状态	OFF ：正常 ON ：异常	• 当用于进行通讯的模式和在串行通讯功能中设置的模式不一样时，变为 ON。 • 如果自此以后情况恢复为正常，仍保持为 ON。		S（出错）		
SM112	出错信息清除	ON ：已清除	• 当存储在 SM110、SM111、SD110 和 SD111 中的出错代码被清除时，变为 ON。（当从 OFF 变为 ON 时触发）		U		
SM113	溢出出错	OFF ：正常 ON ：异常	• 当在串行通讯出错中发生溢出出错时，变为 ON。		S（出错）		
SM114	奇偶性出错	OFF ：正常 ON ：异常	• 当在串行通讯出错中发生奇偶性出错时，变为 ON。		S（出错）		
SM115	结构出错	OFF ：正常 ON ：异常	• 当在串行通讯出错中发生结构出错时，变为 ON。		S（出错）		

(2) 系统信息

附表 3.3 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 CPU ACPU M9□□□	相应的 CPU
SM202	LED OFF 命令	OFF→ON : LED OFF	• 当此继电器从 OFF 变为 ON 时，对应 SD202 上的单个位的 LED 关闭	U	新增	QnA Qn(H) QnPH QnPRH QnU
SM203	STOP 触点	STOP 状态	• 在 STOP 状态变为 ON	S（状态改变）	M9042	○
SM204	PAUSE 触点	PAUSE 状态	• 在 PAUSE 状态变为 ON	S（状态改变）	M9041	
SM205	STEP-RUN 触点	STEP-RUN 状态	• 在 STEP-RUN 状态变为 ON	S（状态改变）	M9054	QnA
SM206	PAUSE 许可线圈	OFF :PAUSE 禁止 ON :PAUSE 许可	• 当 PAUSE 触点变为 ON 时，如果此继电器是 ON，则 PAUSE 状态被输入	U	M9040	○
	软元件测试请求接收状态	OFF ：软元件测试还没有执行 ON ：软元件测试执行	• 当在 GX Developer 上执行软元件测试模式时，变为 ON。	S（请求）	新增	Rem
SM210	时钟数据设置请求	OFF ：忽略 ON ：设置请求	• 当此继电器从 OFF 变为 ON 时，在扫描周期有变化的 END 指令执行之后，正被从 SD210 存储到 SD213 的时钟数据将被写到 CPU 模块中。	U	M9025	○
SM211	时钟数据出错	OFF ：没有出错 ON ：出错	• 当时钟数据（SD210 到 SD213）值产生出错时为 ON，如果没有检测到出错则为 OFF。	S（请求）	M9026	
SM212	时钟数据显示	OFF ：忽略 ON ：显示	• 在 CPU 模块前面的 LED 显示器上，以月、日、小时、分、秒的格式显示时钟数据。（只对 Q3ACPU 和 Q4ACPU 有效）	U	M9027	Q3A Q4A Q4AR
SM213	时钟数据显示	OFF ：忽略 ON ：读取请求	• 当此继电器是 ON 时，时钟数据以 BCD 值读到 SD210 到 SD213 中。	U	M9028	○ Rem
SM220	1 号 CPU 准备完成	OFF :1 号 CPU 准备未完成 ON :1 号 CPU 准备完成	电源接通或复位操作时，可以从其它号 CPU 访问 1 号 CPU 时，变为 ON。SM220 在多 CPU 间同步设置中被设置为非同步时，作为 1 号 CPU 访问的互锁。	S（状态改变）	新增	QnU
SM221	2 号 CPU 准备完成	OFF :2 号 CPU 准备未完成 ON :2 号 CPU 准备完成	电源接通或复位操作时，可以从其它号 CPU 访问 2 号 CPU 时，变为 ON。SM221 在多 CPU 间同步设置中被设置为非同步时，作为 2 号 CPU 访问的互锁。			
SM222	3 号 CPU 准备完成	OFF :3 号 CPU 准备未完成 ON :3 号 CPU 准备完成	电源接通或复位操作时，可以从其它号 CPU 访问 3 号 CPU 时，变为 ON。SM222 在多 CPU 间同步设置中被设置为非同步时，作为 3 号 CPU 访问的互锁。			
SM223	4 号 CPU 准备完成	OFF :4 号 CPU 准备未完成 ON :4 号 CPU 准备完成	电源接通或复位操作时，可以从其它号 CPU 访问 4 号 CPU 时，变为 ON。SM223 在多 CPU 间同步设置中被设置为非同步时，作为 4 号 CPU 访问的互锁。			QnU*5
SM235	运行中模块更换标志	OFF ：当前没有进行运行中模块更换 ON ：运行中模块更换正在进行	在运行中模块更换正在进行时变为 ON。（用于本站 CPU）	S（当运行中模块更换执行时）	新增	QnPH
SM236	运行中模块更换后 ON 一个扫描的标志	OFF ：运行中模块更换未结束 ON ：运行中模块更换结束	• 运行中模块更换结束后变为 ON 一个扫描周期。 • 此触点只能被扫描程序所使用。（用于本站 CPU）	S（当运行中模块更换结束时）	新增	
SM240	1 号 CPU 复位标志	OFF :1 号 CPU 复位被取消 ON :1 号 CPU 复位	• 当 1 号 CPU 的复位被取消时，变为 OFF。 • 当 1 号 CPU 被复位时，变为 ON。（包括从基板上卸载可编程控制器的情况）。 其他可编程控制器也进入复位状态。	S（状态改变）	新增	
SM241	2 号 CPU 复位标志	OFF :2 号 CPU 复位被取消 ON :2 号 CPU 复位	• 当 2 号 CPU 的复位被取消时，变为 OFF。 • 当 2 号 CPU 被复位时，变为 ON。（包括从基板上卸载可编程控制器的情况）。 其他可编程控制器处于 “MULTI CPU DOWN” 状态（出错代码：7000）。			Q00/Q01*1 Qn(H)*1 QnPH QnU
SM242	3 号 CPU 复位标志	OFF :3 号 CPU 复位被取消 ON :3 号 CPU 复位	• 当 3 号 CPU 的复位被取消时，变为 OFF。 • 当 3 号 CPU 被复位时，变为 ON。（包括从基板上卸载可编程控制器的情况）。 其他可编程控制器处于 “MULTI CPU DOWN” 状态（出错代码 :7000）。			
SM243	4 号 CPU 复位标志	OFF :4 号 CPU 复位被取消 ON :4 号 CPU 复位	• 当 4 号 CPU 复位被取消时变为 OFF。 • 当 4 号 CPU 是复位时，变为 ON（包括从基板上卸载可编程控制器的情况）。 其他可编程控制器处于 “MULTI CPU DOWN” 状态（出错代码 :7000）。			Qn(H)*1 QnPH QnU*5
SM244	1 号 CPU 出错标志	OFF :1 号 CPU 正常 ON :1 号 CPU 处于停止出错状态	• 当 1 号 CPU 正常时，变为 OFF（包括连续出错）。 • 当 1 号 CPU 处于停止出错状态时，变为 ON。			Q00/Q01*1 Qn(H)*1 QnPH QnU
SM245	2 号 CPU 出错标志	OFF :2 号 CPU 正常 ON :2 号 CPU 处于停止出错状态	• 当 2 CPU 号正常时，变为 OFF（包括连续出错）。 • 当 2 号 CPU 处于停止出错状态，变为 ON。			
SM246	3 号 CPU 出错标志	OFF :3 号 CPU 正常 ON :3 号 CPU 处于停止出错状态	• 当 3 号 CPU 正常时，变为 OFF（包括连续出错）。 • 当 3 号 CPU 处于停止出错状态，变为 ON。			

*5: 除 Q02UCPU 外的通用型 QCPU。

附表 3.3 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM247	4 号 CPU 出错标志	OFF :4 号 CPU 正常 ON :4 号 CPU 处于停止出错状态	• 当 4 号 CPU 正常时，变为 OFF（包括连续出错）。 • 当 4 号 CPU 号处于停止出错状态，变为 ON	S（状态改变）	新增	Qn(H)*1 QnPH QnU*5
SM250	读有负载的最大 I/O 号	OFF ：忽略 ON ：读	• 当此继电器从 OFF 变为 ON 时，有负载的最大 I/O 号被读到 SD250。	U	新增	QnA Qn(H) QnPH QnPRH Rem
SM251	I/O 更换标志	OFF ：无更换 ON ：更换	• 在将要被更换的 I/O 模块的起始 I/O 号设定到 SD251 之后，通过打开此 继电器，可以在运行中更换 I/O 模块（电源接通）。（对每个设置只能更换一个模块。） • 在程序或者用于在 RUN 过程中进行 I/O 模块更换的外围设备的测试模式中，或者在用于在 STOP 时进行 I/O 更换的外围设备的测试模式，将此继电器变为 ON。 • 在 I/O 模块更换完成之前，不要执行 RUN/STOP 模式改变。	U	M9094	Q2A(S1) Q3A Q4A Q4AR
SM252	I/O 更换 OK	OFF ：更换禁止 ON ：更换允许	• 当 I/O 更换 OK 时，变为 ON。	S（结束）	新增	
SM254	所有站刷新命令	OFF ：刷新到达的站 ON ：刷新所有站	• 对批量刷新有效（也对低速循环有效） • 在 MELSECNET/H 中指定是只接收到达的站，还是接收所有的从站。	U	新增	Qn(H) QnPH QnPRH
			• 在 MELSECNET/G 中指定是只接收到达的站，还是接收所有的从站。			Qn(H)*2
			• 对批量刷新有效（也对低速循环有效） • 在 MELSECNET/H 中指定是只接收到达的站，还是接收所有的从站。			QnU
SM255	MELSECNET/10, MELSECNET/H 模块 1 信息	OFF ：运行网络 ON ：待机网络	• 为待机网络变为 ON（如果 没有指定是活动还是待机，则假设是活动。）	S（初始化）	新增	QnA Qn(H) QnPH QnPRH
SM256		OFF ：读 ON ：不读	• 用于从链接到 CPU 模块的刷新（B、W 等），表示是否从链接模块读。	U	新增	
SM257		OFF ：写 ON ：不写	• 用于从 CPU 模块到链接的刷新（B、W 等），指定是否写入链接模块。	U	新增	
SM260	MELSECNET/10, MELSECNET/H 模块 2 信息	OFF ：运行网络 ON ：待机网络	• 为待机网络变为 ON （如果没有是活动还是待机，假设是活动。）	S（初始化）	新增	
SM261		OFF ：读 ON ：不读	• 用于从链接到 CPU 模块的 刷新（B、W 等），表示是否从链接模块读。	U	新增	
SM262		OFF ：写 ON ：不写	• 用于从 CPU 模块到链接的刷新（B、W 等），指定是否写到链接模块。	U	新增	
SM265	MELSECNET/10, MELSECNET/H 模块 3 信息	OFF ：运行网络 ON ：待机网络	• 为待机网络变为 ON （如果没有指定是活动还是待机，假定为活动。）	S（初始化）	新增	
SM266		OFF ：读 ON ：不读	• 用于从链接到 CPU 模块的刷新（B、W 等），表示是否从链接模块读。	U	新增	
SM267		OFF ：写 ON ：不写	• 用于从 CPU 模块到链接的刷新（B、W 等），指定是否到写到链接模块。	U	新增	
SM270	MELSECNET/10, MELSECNET/H 模块 4 信息	OFF ：运行网络 ON ：待机网络	• 为待机网络变为 ON （如果没有指定是活动还是待机，假定为活动。）	S（初始化）	新增	
SM271		OFF ：读 ON ：不读	• 用于从链接到 CPU 模块的刷新（B、W 等），表示是否从链接模块读。	U	新增	
SM272		OFF ：正常 ON ：出错	• 用于从 CPU 模块到链接的刷新（B、W 等），指定是否写到链接模块。	U	新增	
SM280	CC-Link 出错	OFF ：正常 ON ：出错	• 当在任何一个安装的 CC-Link 模块中检测到 CC-Link 出错时，变为 ON。当正常操作恢复时，变为 OFF。	S（状态改变）	新增	Qn(H) QnPH QnPRH Rem
			• 当在任何一个安装的 CC-Link 模块中检测到 CC-Link 出错时，变为 ON。如果条件恢复为正常，仍保持为 ON。	S（出错）	新增	QnA
SM315	通讯预留时间延迟许可 / 禁止标志	OFF ：无延迟 ON ：有延迟	• 当为通讯处理预留的时间被设定到 SD315 中时，此标志被激活。 • 以将 END 处理延迟在 SD315 中为执行通讯处理设定的时间，变为 ON。 （扫描时间将增加在 SD315 中设定的时间。） • 当没有通讯处理时，变为 OFF，以便于不延迟在 SD315 中设定的时间的情况下执行 END 处理。（默认是 OFF）	U	新增	Q00J/Q00/Q01

*1: 适用于功能版本为 B 或者更高的 CPU。
*2: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。
*5: 除 Q02UCPU 外的通用型 QCPU。

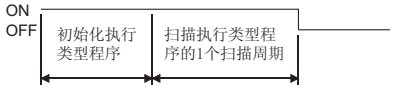
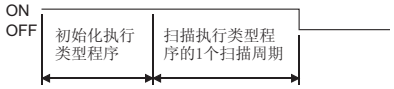
附表 3.3 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM320	存在 / 不存在 SFC 程序	OFF :SFC 程序不存在 ON :SFC 程序存在	• 当注册了 SFC 程序时，变为 ON。 • 当没有注册 SFC 程序时，变为 OFF。	S（初始化）	M9100	QnA Q00J/Q00/Q01*1 Qn(H) QnPH QnPRH QnU
SM321	启动 / 停止 SFC 程序	OFF :SFC 程序没有执行（停止） ON :SFC 程序执行（启动）	• 初始值被设定为和 SM320 相同的值。（自动变为 ON ，如果 SFC 程序存在。） • 将此继电器从 ON 变为 OFF 以停止程序执行。 • 将此继电器从 OFF 变为 ON 以恢复程序执行。	S（初始化）/U	M9101 形式在变化	
SM322	SFC 程序启动状态	OFF ：初始化启动 ON ：重启	• 在可编程控制器参数对话框的 SFC 设置中的 SFC 程序启动模式被设定为初始值。 在初始化启动时：OFF 在连续启动时 ： ON	S（初始化）/U	M9102 形式在变化	
SM323	SFC 程序启动状态	OFF ：连续转移无效 ON ：连续转移有效	在没有设定 SFC 数据软元件的 “Continuous transition bit” 地方设定存在 / 不存在块的连续转移。	U	M9103	
SM324	连续转移防止标志	OFF ：当转移被执行时 ON ：当没有转移时	• 在连续转移模式中运行或者在连续转移过程中为 OFF，而当连续转移没有执行时为 ON。 • 在没有转移模式中运行时为常 ON。	S（指令执行）	M9104	
				S（状态改变）	新增	
SM325	块停止时的输出模式	OFF ： OFF ON ：保持	选择在块停止时刻活动步的线圈输出是否被保持。 • 作为初始值，当线圈输出为 OFF 时参数中块停止时的输出模式是 OFF，当线圈输出被保持时为 ON。 • 当此继电器是 OFF 时，所有线圈输出变为 OFF。 • 当此继电器是 ON 时，线圈输出被保持。	S（初始化）/U	M9196	
SM326	SFC 软元件清除模式	OFF ：清除软元件 ON ：保持软元件	当 SFC 程序存在时，选择当处于停止状态的 CPU 在顺控程序或者 SFC 程序被修改之后重新运行时软元件的状态。	U	新增	
SM327	执行结束步时的输出	OFF ：保持步的输出为 OFF（清除） ON ：保持步的输出保持	• 该继电器 OFF 时，选择在从 STOP → 程序写入 →RUN 的切换时刻软元件的状态。（所有软元件，步继电器除外）	S（初始化）/U	新增	QnA Qn(H) QnPH QnPRH QnU
				U		Q00J/Q00/Q01*1
SM328	当到达结束步时清除处理的模式	OFF ：清除处理执行。 ON ：清除处理是不执行。	当结束步到达时，如果在块中除正被保持的步以外没有活动步存在，选择是否执行清除处理？ • 当此继电器变为 OFF 时，所有的活动步被强制结束，以结束此块。 • 当此继电器是 ON 时，执行的块继续执行。 • 当结束步到达时，如果除正被保持的步以外没有活动步存在，则正被保持的步被结束以结束此块。	U	新增	Q00J/Q00/Q01*1 QnU
SM330	低速执行类型程序的运行模式	OFF ：异步模式 ON ：同步模式	选择低速执行类型程序是以同步模式执行，还是以异步模式执行。 • 异步模式（此继电器变为 OFF。） 在此模式中，低速执行类型程序的操作在剩余的时间内继续执行。 • 同步模式（此继电器变为 ON。） 在此模式中，低速执行类型程序的操作不再继续执行，如果有剩余的时间，操作将从下一个扫描周期继续执行。	U	新增	QnA Qn(H) QnPH
SM331	正常 SFC 程序执行状态	OFF ：未执行 ON ：正在执行	• 表示是否有正常的 SFC 程序正在执行。 • 被用作 SFC 控制指令执行互锁。	S（状态改变）	新增	Qn(H)*3 QnPH*4 QnPRH
SM332	程序执行管理 SFC 程序执行状态	OFF ：未执行 ON ：正在执行	• 表示是否有程序执行管理 SFC 程序正在执行。 • 被用作 SFC 控制指令执行互锁。			
SM390	访问执行标志	ON 表示智能功能模块访问的结束	• 执行的智能功能模块 访问指令在存储之前的状态。（当重新执行智能功能模块访问指令时，此数据被重写。） • 被用户在程序中用作结束位。	S（状态改变）	新增	Qn(H) QnPH QnPRH
SM391	GINT 指令执行结束标志	OFF ：未执行 ON ：执行结束	表示 S(P),GINT 指令的执行状态。 • 指令执行前为 OFF。 • 指令执行结束后为 ON。	S（指令执行）	新增	QnU

*1: 适用于功能版本为 B 或者更高的 CPU。
*3: 以序列号的高 5 位为 “04122” 以后的 CPU 为对象。
*4: 以序列号的高 5 位为 “07032” 以后的 CPU 为对象。

(3) 系统时钟 / 计数器

附表 3.4 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM400	常 ON	ON OFF	• 正常是 ON	S（每个 END 处理）	M9036	○
SM401	常 OFF	ON OFF	• 正常是 OFF	S（每个 END 处理）	M9037	
SM402	在 RUN 之后，只 ON1 个扫描周期	ON OFF 1个扫描周期	• 在 RUN 之后，只 ON1 个扫描周期。 • 此连接只能用于扫描执行类型程序。 • 当使用初始化执行类型程序时，在 RUN 之后的第一个扫描周期中，此继电器在扫描执行类型程序的 END 处理中变为 OFF。 	S（每个 END 处理）	M9038	QnA Qn(H) QnPH QnPRH QnU
			• 在 RUN 之后，只 ON1 个扫描周期。	S（每个 END 处理）	新增	Q00J/Q00/Q01
SM403	在 RUN 之后，只 OFF1 个扫描周期	ON 1个扫描周期 OFF	• 在 RUN 之后，只 OFF1 个扫描周期。 • 此连接只能用于扫描周期执行类型程序。 • 当使用了初始化执行类型程序时，在 RUN 之后第一个扫描周期中，扫描执行类型程序的 END 处理时此继电器变为 OFF。 	S（每个 END 处理）	M9039	QnA Qn(H) QnPH QnPRH QnU
			• 在 RUN 之后，只 OFF1 个扫描周期。	S（每个 END 处理）	新增	Q00J/Q00/Q01
SM404	在 RUN 之后低速执行类型程序只 ON1 个扫描周期	ON 1个扫描周期 OFF	• 在 RUN 之后，只 ON1 个扫描周期。 • 此连接只能用于低速执行类型程序。	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SM405	在 RUN 之后，低速执行类型程序只 OFF1 个扫描周期	ON 1个扫描周期 OFF	• 在 RUN 之后，只 OFF1 个扫描周期。 • 此连接只能用于低速执行类型程序。	S（状态改变）	新增	
SM409	0.01 秒时钟	0.005s 0.005s	• 以 5 毫秒为周期，在 ON 和 OFF 之间重复改变。 • 当可编程控制器电源变为 OFF 或者执行 CPU 模块复位时，从 OFF 开始启动。 （注意，在程序执行过程中，当指定的时间结束时，ON-OFF 状态改变。）	S（状态改变）	新增	Qn(H) QnPH QnPRH QnU
SM410	0.1 秒时钟	0.05s 0.05s	• 以指定的时间间隔，在 ON 和 OFF 之间重复改变。 • 当可编程控制器电源变为 OFF 或者执行 CPU 模块复位时，从 OFF 开始启动。 （注意，在程序执行过程中，当指定的时间结束时，ON-OFF 状态改变。）	S（状态改变）	M9030	○
SM411	0.2 秒时钟	0.1s 0.1s			M9031	
SM412	1 秒时钟	0.5s 0.5s			M9032	
SM413	2 秒时钟	1s 1s			M9033	
SM414	2n 秒时钟	ns ns	• 此继电器按照在 SD414 中指定的时间间隔（单位：s），在 ON 和 OFF 两个状态交替改变。 • 当可编程控制器电源变为 OFF 或者执行 CPU 模块复位时，从 OFF 开始启动。 （注意，在程序执行过程中，当指定的时间结束时，ON-OFF 状态改变。）	S（状态改变）	M9034 形式	
SM415	2n(ms) 时钟	n(ms) n(ms)	• 此继电器按照 SD415 中指定的时间间隔（单位：ms），在 ON 和 OFF 两个状态交替改变。 • 当可编程控制器电源变为 OFF 或者执行 CPU 模块复位时，从 OFF 开始启动。 （注意，在程序执行过程中，当指定的时间结束时，ON-OFF 状态改变。）	S（状态改变）	新增	Qn(H) QnPH QnPRH QnU

附表 3.4 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM420	0 号用户定时时钟		• 继电器以固定扫描间隔重复 ON/OFF 切换。 • 当可编程控制器电源变为 ON 或者执行 CPU 模块复位时，从 OFF 开始启动。（对于冗余 CPU，此继电器在系统切换之后一直为 OFF。） • ON/OFF 间隔是用 DUTY 指令设定的 DUTY | n1 | n2 | SM420 n1：ON 扫描周期间隔 n2：OFF 扫描周期间隔	S（每个 END 处理）	M9020	○
SM421	1 号用户定时时钟				M9021	
SM422	2 号用户定时时钟				M9022	
SM423	3 号用户定时时钟				M9023	
SM424	4 号用户定时时钟	n2个扫描周期 n2个扫描周期 n1个扫描周期			M9024	
SM430	5 号用户定时时钟		• 用于和 SM420 到 SM424 低速程序一起使用	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SM431	6 号用户定时时钟					
SM432	7 号用户定时时钟					
SM433	8 号用户定时时钟					
SM434	9 号用户定时时钟					

(4) 扫描周期信息

附表 3.5 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM510	低速程序执行标志	OFF ：结束或者未执行 ON ：正在执行。	• 当执行低速执行类型程序时，变为 ON。	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SM551	读模块工作间隔	OFF ：忽略 ON ：读	• 当此继电器从 OFF 变为 ON 时，SD550 指定的模块工作间隔被读到 SD551 到 SD552。	U	新增	QnA Qn(H) QnPH QnPRH Rem

(5) I/O 刷新

附表 3.6 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM580	程序到程序 I/O 刷新	OFF ：未刷新 ON ：刷新	• 当此特殊继电器变为 ON 时，在第一个程序执行之后执行 I/O 刷新，然后执行第二个程序。当顺控程序和 SFC 程序将要被执行时，先执行顺控程序，再执行 I/O 刷新，最后执行 SFC 程序。	U	新增	Q00J/Q00/Q01*1

*1：适用于功能版本为 B 或者更高的 CPU。

(6) 存储卡

附表 3.7 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM600	存储卡 (A) 可用标志	OFF ：不可用 ON ：使用允许	• 当存储卡 (A) 准备好供用户使用时，变为 ON	S（状态改变）	新增	QnA Qn(H) QnPH QnPRH QnU
SM601	存储卡 (A) 保护标志	OFF ：无保护 ON ：保护	• 当存储卡 (A) 保护开关为 ON，变为 ON	S（状态改变）	新增	
SM602	驱动器 1 标志	OFF ：无驱动器 1 ON ：驱动器 1 存在	• 当安装的存储卡 (A) 是 RAM 时，变为 ON	S（状态改变）	新增	
SM603	驱动器 2 标志	OFF ：无驱动器 2 ON ：驱动器 2 存在	• 当安装的存储卡 (A) 是 ROM 时，变为 ON	S（状态改变）	新增	
SM604	存储卡 (A) 使用标志	OFF ：未使用 ON ：使用中	• 当安装的存储卡 (A) 是 ROM 时，变为 ON	S（状态改变）	新增	
SM605	存储卡 (A) 卸载 / 插入禁止标志	OFF ：卸载 / 插入允许 ON ：卸载 / 插入禁止	• 当存储卡 (A) 不能插入或者卸载时，变为 ON	U	新增	
SM609	存储卡卸载 / 插入允许标志	OFF ：卸载 / 插入禁止 ON ：卸载 / 插入允许	• 由用户设定为 ON，以允许卸载 / 插入存储卡。 • 在存储卡卸载之后，由系统设定为 OFF。 • 只有当 SM604 和 SM605 为 OFF 时，才能使用此触点。	S/U	新增	Qn(H) QnPH QnPRH QnU
SM620	驱动器 3/4 可用标志	OFF ：不可用 ON ：允许使用	• 常 ON	S（初始化）	新增	QCPU
	存储卡 B 可用标志	OFF ：不可用 ON ：允许使用	• 当存储卡 B 准备好供用户使用时，变为 ON	S（初始化）	新增	Q2A(S1) Q3A Q4A Q4AR
SM621	驱动器 3/4 保护标志	OFF ：无保护 ON ：保护	• 常 OFF	S（初始化）	新增	QCPU
	存储卡 B 保护标志	OFF ：无保护 ON ：保护	• 当存储卡 B 保护开关是 ON 时，变为 ON	S（初始化）	新增	Q2A(S1) Q3A Q4A Q4AR
SM622	驱动器 3 标志	OFF ：无驱动器 3 ON ：驱动器 3 存在	• 常 ON	S（初始化）	新增	QCPU
			• 当驱动器 3（卡 2RAM 区）为当前状态时，变为 ON	S（初始化）	新增	Q2A(S1) Q3A Q4A Q4AR
SM623	驱动器 4 标志	OFF ：无驱动器 4 ON ：驱动器 4 存在	• 常 ON	S（初始化）	新增	QCPU
			• 当驱动器 4（卡 2ROM 区）为当前状态时，变为 ON	S（初始化）	新增	Q2A(S1) Q3A Q4A Q4AR
SM624	存储卡 B 使用标志	OFF ：未使用 ON ：使用中	• 变为 ON，当驱动器 3（标准 RAM）或者驱动器 4（标准 ROM）内文件被使用时。	S（状态改变）	新增	Q00J/Q00/Q01*1 Qn(H) QnPH QnPRH
			• 当存储卡 B 被使用时，变为 ON	S（状态改变）	新增	Q2A(S1) Q3A Q4A Q4AR
	驱动器 3/4 使用标志	OFF ：未使用 ON ：使用中	• 变为 ON，当驱动器 3（标准 RAM）或者驱动器 4（标准 ROM）内文件被使用时。	S（状态改变）	新增	QnU
SM625	存储卡 B 卸载 / 插入禁止标志	OFF ：卸载 / 插入允许 ON ：卸载 / 插入禁止	• 当存储卡 B 不能插入或者卸载时，变为 ON	U	新增	Q2A(S1) Q3A Q4A Q4AR

附表 3.7 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 CPU ACPU M9□□□	相应的 CPU
SM640	文件寄存器使用	OFF ：文件寄存器未使用 ON ：文件寄存器使用中	• 当文件寄存器被使用时，变为 ON	S（状态改变）	新增	○
SM650	注释使用	OFF ：文件寄存器未使用 ON ：文件寄存器使用中	• 当注释文件被使用时，变为 ON	S（状态改变）	新增	QnA Qn(H) QnPH QnPRH QnU
SM660	引导操作	OFF ：内部内存执行 ON ：引导操作进行中	• 当引导操作正在进行时，变为 ON • 如果引导指示开关是 OFF 时，变为 OFF	S（状态改变）	新增	QnA Qn(H) QnPH QnPRH
		OFF ：程序内存执行 ON ：引导操作进行中	• 当引导操作正在进行时，变为 ON	S（状态改变）	新增	Q00J/Q00/Q01 QnU
SM671	锁存数据备份到标准 ROM 结束标志	OFF ：未结束 ON ：结束	• 锁存数据备份到标准 ROM 结束时，变为 ON。 • 锁存数据备份到标准 ROM 的执行时间，存储在 SD672 以后。	S（状态改变）	新增	QnU
SM672	存储卡 A 文件寄存器访问范围标志	OFF ：在访问范围内 ON ：超出访问范围	• 当访问是在存储卡 A 的文件寄存器 R 的范围以外的区域进行时（在 END 处理中设定。），变为 ON • 在用户程序中复位	S/U	新增	QnA Qn(H) QnPH QnPRH
SM673	存储卡 B 文件寄存器访问范围标志	OFF ：在访问范围内 ON ：超出访问范围	• 当访问是在存储卡 B 的文件寄存器 R 的范围以外的区域进行时（在 END 处理中设定。），变为 ON • 在用户程序中复位	S/U	新增	Q2A(S1) Q3A Q4A Q4AR
SM675	锁存数据备份到标准 ROM 结束出错	OFF ：未出错 ON ：出错	• 正常备份锁存数据时，不能备份数据到标准 ROM 中，变为 ON。 • 正常备份锁存数据时，数据备份到标准 ROM 中，变为 OFF。	S	新增	QnU
SM676	恢复反复执行指定	OFF ：未指定 ON ：指定	• SM676 为 ON 时，如果执行锁存数据备份，则每次下次电源 OFF → ON 时恢复数据。 • 删除备份的锁存数据，或每次电源 OFF → ON 时恢复数据，直到锁存数据备份再次执行。	U	新增	QnU
SM680	程序内存写入出错	OFF ：写入出错 ON ：未执行写入 / 正常	• 如果程序内存（闪存 ROM）写入时检测到出错，变为 ON。 写入指示时为 OFF。	S（写入时）	新增	QnU
SM681	程序内存写入标志	OFF ：写入中 ON ：未执行写入	• 正在进行程序内存（闪存 ROM）写入时，变为 ON。写入结束时变为 OFF。	S（写入时）	新增	QnU
SM682	程序内存改写次数出错标志	OFF ：改写次数为 100,000 以后 ON ：改写次数不到 100,000	• 当程序内存（闪存 ROM）改写次数达到 100,000 时，变为 ON。	S（写入时）	新增	QnU
SM685	标准 ROM 写入出错	OFF ：写入出错 ON ：未写入 / 正常	• 如果标准 ROM 写入时检测到出错，变为 ON。 • 写入指示时为 OFF。	S（写入时）	新增	QnU
SM686	标准 ROM 写入标志	OFF ：写入中 ON ：未执行写入	• 正在进行标准 ROM 写入时，变为 ON。写入结束时变为 OFF。	S（写入时）	新增	QnU
SM687	标准 ROM 改写次数出错标志	OFF ：改写次数为 100,000 以后 ON ：改写次数不到 100,000	• 当标准 ROM 改写次数达到 100,000 时，变为 ON。（必需更换 CPU 模块。）	S（写入时）	新增	QnU

(7) 指令相关特殊继电器

附表 3.8 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM700	进位标志	OFF ：进位 OFF ON ：进位 ON	• 用在应用指令中的进位标志	S（指令执行）	M9012	○
SM701	输出字符数切换	切换输出的字符数、输出模式	• 通过 PR、PRC、BINDA、DBINDA、BINHA、DBINHA、BCDDA、DBCDDA 或 COMRD 指令使用。 • 详细内容请参阅 QCPU(Q 模式)/QnACPU 编程手册（公共指令篇）。	U	M9049	QnA Qn(H) QnPH QnPRH QnU
SM702	查找方法	OFF ：查找下一个 ON ：二分法查找	• 指定查找指令使用的方法。 • 必须排列数据，以用于二分法查找。	U	新增	○
SM703	排序类型	OFF ：升序 ON ：降序	• 排序指令用于指定以升序存储数据还是以降序存储数据。	U	新增	
SM704	块比较	OFF ：没有找到匹配项 ON ：所有的匹配项	• 当 BKCMP 指令的所有的数据条件都满足时，变为 ON	S（指令执行）	新增	
SM707	实数指令处理类型选择	OFF ：速度优先 ON ：精度优先	• 当 SM707 是 OFF 时，实数指令以高速执行。 • 当它是 ON 时，实数指令以高精度执行。	U	新增	Q4AR

附表 3.8 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM710	CHK 指令优先等级标志	OFF ：条件优先级 ON ：模式优先级	• 当 OFF 时，保持初始设置。 • 当 ON 时，更新 CHK 优先级。	S（指令执行）	新增	QnA Qn(H) QnPH QnPRH
SM711	分开传送状态	OFF ：分开处理之外 ON ：在分开处理过程中	• 在 AD57(S1) 的处理中，当画布画面被拆分以用于传送时变为 ON，当拆分处理结束时变为 OFF	S（指令执行）	M9065	QnA
SM712	传送处理选择	OFF ：批量处理 ON ：分开处理	• 在 AD57(S1) 的处理中，当画布画面被分开以用于传送时变为 ON。	S（指令执行）	M9066	
SM714	通讯请求注册区域 BUSY 信号	OFF ：允许到远程端子模块的通讯请求 ON ：禁止到远程端子模块的通讯请求	• 用于确定是否可以执行连接到 AJ71PT32-S3 的远程端子模块的通讯请求。	S（指令执行）	M9081	
SM715	EI 标志	OFF ：在 DI 过程中 ON ：在 EI 过程中	• 当 EI 指令正被执行时。	S（指令执行）	新增	○
SM720	注释读结束标志	OFF ：注释读未结束 ON ：注释读结束	• 只在 COMRD 或者 PRC 指令的处理结束时，接通一个扫描周期。	S（状态改变）	新增	Qn(H) QnPH
			• 只在 COMRD 指令的处理结束时，接通一个扫描周期。			Qn(H) QnPH QnPRH
SM721	文件正被访问	OFF ：文件未被访问 ON ：文件正被访问	• 当文件正被 S.FWRITE、S.FREAD、COMRD、PRC 或者 LEDC 指令访问时，切换为 ON。	S（状态改变）	新增	Qn(H) QnPH
			• 当文件正被 S.FWRITE、S.FREAD、COMRD 或者 LEDC 指令访问时，切换为 ON。			Qn(H) QnPH QnPRH
			• 当文件正被 S.FWRITE、S.FREAD、COMRD 或者 SP.DEVST 指令访问时，切换为 ON。			QnU
SM722	BIN/DBIN 指令出错禁止标志	OFF ：执行出错检测 ON ：未执行出错检测	• 当为 BIN 或者 DBIN 指令抑制了 “OPERATION ERROR” 时，变为 ON。	U	新增	QCPU
SM730	用于 CC-Link 通讯请求登录区的 BUSY 信号	OFF ：与智能软元件站进行通讯的请求被允许 ON ：与智能软元件站进行通讯的请求被禁止	• 用于判断是允许还是禁止与连接了 CC-Link 模块的智能软元件站进行通讯的请求。	S（执行指令时）	新增	QnA
SM734	XCLL 指令执行条件指定	OFF ：执行条件的上升沿时不执行 ON ：执行条件的上升沿时执行	• OFF 的情况下执行条件的上升沿时不执行 XCLL 指令。 • ON 的情况下执行条件的上升沿时执行 XCLL 指令。	U	新增	Qn(H)*1
SM735	SFC 注释读取指令执行中标志	OFF ：未执行 SFC 注释读取指令 ON ：正在执行 SFC 注释读取指令	• 正在执行 SFC 步注释读取指令 (S(P).SFCSCOMR)、SFC 转移条件注释读取指令 (S(P).SFCTCOMR) 时 ON。	S（状态变化）	新增	Qn(H)*2 QnPH*3 QnPRH*3
SM736	PKEY 指令执行进行中标志	OFF ：指令未执行 ON ：指令执行	• 当 PKEY 指令正在被执行时，变为 ON。 当 CR 是输入或者当输入字符串达到 32 个字符时，变为 OFF。	S（指令执行）	新增	QnA
SM737	用于 PKEY 指令的键盘输入接收标志	OFF ：键盘输入接收允许 ON ：键盘输入接收禁止	• 当键盘输入正在进行时，变为 ON。 当键盘输入已经存储在 CPU 中时，变为 OFF。	S（指令执行）	新增	
SM738	MSG 指令接收标志	OFF ：指令未执行 ON ：指令执行	• 当执行 MSG 指令时，变为 ON。	S（指令执行）	新增	Qn(H) QnPRH

*1: 以序列号的高 5 位为 “06082” 或者更高的 CPU。

附表 3.8 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM774	PID 无冲击处理（用于全微分）	OFF ：匹配 ON ：不匹配	• 指定在手动模式中，设定值 (SV) 是否和过程值 (PV) 匹配。	U	新增	QnA Q00J/Q00/Q01*4 Qn(H) QnPRH QnU
SM775	在 COM 指令执行过程中，选择刷新处理	OFF ：执行链接刷新 ON ：不执行链接刷新	• 在执行 COM 指令时，在只与 CPU 模块进行通讯的情况下，选择是否执行链接刷新处理。	U	新增	QnA Q00J/Q00/Q01 Qn(H) QnPH
		OFF ：执行所有的刷新处理 ON ：执行在 SD778 中设定的刷新	• 当执行 COM 指令时，选择是执行所有的刷新处理还是执行 SD778 设定的刷新处理。	U	新增	Q00J/Q00/Q01*4 Qn(H)*5 QnPH*3 QnPRH
SM776	在 CALL 指令执行时允许 / 禁止局部软元件	OFF ：局部软元件禁止 ON ：局部软元件允许	• 设定 CALL 指令执行时调用的子程序的局部软元件是否有效。	U	新增	QnA Qn(H) QnPH QnPRH QnU
SM777	在中断程序中允许 / 禁止局部软元件	OFF ：局部软元件禁止 ON ：局部软元件允许	• 设定在执行中断程序时局部软元件是否有效。	U	新增	
SM780	CC-Link 专用指令可执行	OFF ：CC-Link 专用指令可执行 ON ：CC-Link 专用指令不可执行	• 当可以同时执行的 CC-Link 专用指令数达到 32 时。切换为 ON，当指令数小于 32 时，切换为 OFF。	U	新增	QnA
SM794	PID 无冲击处理（用于非全微分	OFF ：匹配 ON ：不匹配	• 指定在手动模式中，设定值 (SV) 是否和过程值 (PV) 相匹配。	U	新增	Q00J/Q00/Q01*4 Qn(H)*6 QnPRH QnU

*1: 以序列号的高 5 位为 “06082”以后的 CPU 为对象。
*2: 以序列号的高 5 位为 “07012”以后的 CPU 为对象。
*3: 以序列号的高 5 位为 “07032”以后的 CPU 为对象。
*4: 适用于功能版本 B 以后的 CPU。
*5: 以序列号的高 5 位为 “04012”以后的 CPU 为对象。
*6: 以序列号的高 5 位为 “05032”以后的 CPU 为对象。

附表 3.8 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM796	多 CPU 高速总线的专用指令使用块信息 (1 号 CPU)	OFF ：固定块 ON ：SD796 设置的块不能固定	• 多 CPU 高速总线（目标 CPU=1 号 CPU）的专用指令使用的专用指令传送区剩余块数少于 SD796 指定的块数时，变为 ON。 指令执行时为 ON。当 END 处理存在空余区时，为 OFF。	S（指令 /END 处理执行时）	新增	QnU*7
SM797	多 CPU 高速总线的专用指令使用块信息 (2 号 CPU)	OFF ：固定块 ON ：SD797 设置的块不能固定	• 多 CPU 高速总线（目标 CPU=2 号 CPU）的专用指令使用的专用指令传送区剩余块数少于 SD797 指定的块数时，变为 ON。 指令执行时为 ON。当 END 处理存在空余区时，为 OFF。	S（指令 /END 处理执行时）	新增	QnU*7
SM798	多 CPU 高速总线的专用指令使用块信息 (3 号 CPU)	OFF ：固定块 ON ：SD798 设置的块不能固定	• 多 CPU 高速总线（目标 CPU=3 号 CPU）的专用指令使用的专用指令传送区剩余块数少于 SD798 指定的块数时，变为 ON。 指令执行时为 ON。当 END 处理存在空余区时，为 OFF。	S（指令 /END 处理执行时）	新增	QnU*7
SM799	多 CPU 高速总线的专用指令使用块信息 (4 号 CPU)	OFF ：固定块 ON ：SD799 设置的块不能固定	• 多 CPU 高速总线（目标 CPU=4 号 CPU）的专用指令使用的专用指令传送区剩余块数少于 SD799 指定的块数时，变为 ON。 指令执行时为 ON。当 END 处理存在空余区时，为 OFF。	S（指令 /END 处理执行时）	新增	QnU*7

*7： 除 Q02UCPU 外的通用型 QCPU。

(8) 调试

附表 3.9 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM800	跟踪准备	OFF ：未准备好 ON ：已准备好	• 当跟踪准备完成时，切换为 ON	S（状态改变）	新增	Qn(H) QnPH QnPRH QnU
	采样跟踪准备		• 当采样跟踪准备好时，变为 ON	S（状态改变）	新增	QnA
SM801	跟踪启动	OFF ：中断 ON ：启动	• 当此继电器切换为 ON 时，跟踪被启动。 • 当此继电器切换为 OFF 时，跟踪中断。 （所有的相关特殊 M 切换为 OFF。）	U	M9047	Qn(H) QnPH QnPRH QnU
	采样跟踪启动		• 采样跟踪启动，当此继电器变为 ON 时 • 当此继电器变为 OFF 时（相关特殊 M 都为 OFF），中断	U	M9047	QnA
SM802	跟踪执行进行中	OFF ：中断 ON ：启动	• 在跟踪执行过程中，切换为 ON。	S（状态改变）	M9046	Qn(H) QnPH QnPRH QnU
	采样跟踪执行进行中		• 在采样跟踪执行过程中，变为 ON	S（状态改变）	M9046	QnA
SM803	跟踪触发器	OFF→ON：启动	• 当此继电器从 OFF 切换到 ON 时（等同于 TRACE 指令执行状态），跟踪被触发器。	U	M9044	Qn(H) QnPH QnPRH QnU
	采样跟踪触发器		• 当此从 OFF 变为 ON 时，（等同于 STRA 指令执行状态）采样跟踪触发器变为 ON。	U	M9044	QnA
SM804	在跟踪触发器之后	OFF ：不在触发器之后 ON ：在触发器之后	• 在跟踪被触发器之后，切换为 ON。	S（状态改变）	新增	Qn(H) QnPH QnPRH QnU
	在采样跟踪触发器之后		• 在采样跟踪触发器之后，变为 ON。	S（状态改变）	新增	QnA

附表 3.9 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM805	跟踪结束	OFF ：未结束 ON ：结束	• 跟踪结束时，切换为 ON	S（状态改变）	M9043	Qn(H) QnPH QnPRH QnU
	采样跟踪结束		• 采样跟踪结束时，变为 ON	S（状态改变）	M9043	QnA
SM806	状态锁存准备	OFF ：未准备好 ON ：准备好	• 当状态锁存是准备好，变为 ON	S（状态改变）	新增	
SM807	状态锁存命令	OFF→ON：锁存	• 运行状态锁存命令	U	新增	
SM808	状态锁存结束	OFF ：锁存未结束 ON ：锁存结束	• 当状态锁存结束时，变为 ON	S（状态改变）	M9055	
SM809	状态锁存清除	OFF→ON：清除	• 允许下一个状态锁存	U	新增	
SM810	程序跟踪准备	OFF ：未准备好 ON ：准备好	• 当程序跟踪是准备好，变为 ON	S（状态改变）	新增	
SM811	启动程序跟踪	OFF ：中断 ON ：启动	• 当此继电器变为 ON 时，程序跟踪启动 • 当此继电器变为 OFF 时（相关特殊 M 都为 OFF），中断	S（状态改变）	新增	
SM812	在程序跟执行进行中	OFF ：中断 ON ：启动	• 当程序跟踪执行正在进行时，变为 ON	U	新增	
SM813	程序跟踪触发器	OFF→ON：启动	• 当此继电器从 OFF 变为 ON 时（等同于 PTRA 指令执行状态），程序跟踪触发器变为 ON	S（状态改变）	新增	
SM814	程序跟踪触发器	OFF ：不在触发之后 ON ：在触发之后	• 在程序跟踪触发器之后，变为 ON	S（状态改变）	新增	
SM815	程序跟踪结束	OFF ：未结束 ON ：结束	• 在程序跟踪结束时，变为 ON	S（状态改变）	新增	
SM820	步跟踪准备	OFF ：未准备好 ON ：准备好	• 在程序跟踪注册准备好时，变为 ON	S（状态改变）	新增	QnA
SM821	步跟踪启动	OFF ：中断 ON ：启动	• 选择是启动还是中断步跟踪的执行。 • 当此继电器变为 ON，步跟踪被启动 • 当 OFF 时中断（相关特殊 M 都为 OFF）	U	M9182 格式改变	
SM822	步跟踪执行进行中	OFF ：中断 ON ：启动	• 当步跟踪执行正在进行时，变为 ON • 在结束或者挂起时，变为 OFF	S（状态改变）	M9181	
SM823	步跟踪触发之后	OFF ：不在触发之后 ON ：在第一个触发之后	• 只要在正被执行的步跟踪内有 1 块被触发时，变为 ON • 当步跟踪开始时，变为 OFF	S（状态改变）	新增	
SM824	步跟踪触发后	OFF ：有触发未成立的块 ON ：所有块均触发成立	• 如果正在执行步跟踪的所有的块都触发成立，则变为 ON。 • 当步跟踪开始时，变为 OFF。	S（状态改变）	新增	
SM825	步跟踪结束	OFF ：未结束 ON ：结束	• 在步跟踪结束时，变为 ON • 当步跟踪开始时，变为 OFF	S（状态改变）	M9180	
SM826	跟踪出错	OFF ：正常 ON ：出错	• 如果出错发生在跟踪执行过程中，切换为 ON	S（状态改变）	新增	Qn(H) QnPH QnPRH QnU
	采样跟踪出错		• 如果出错发生在采样跟踪执行过程中，变为 ON	S（状态改变）	新增	QnA
SM827	状态锁存出错	OFF ：正常 ON ：出错	• 如果出错发生在状态锁存的执行过程中，变为 ON	S（状态改变）	新增	
SM828	程序跟踪出错	OFF ：正常 ON ：出错	• 如果出错发生在程序跟踪的执行过程中，变为 ON	S（状态改变）	新增	
SM829	跟踪设置强制登录指定	OFF ：允许强制登录 ON ：不允许强制登录	• SM829 为 ON 后，通过 GX Developer 登录采样跟踪设置，即使跟踪条件或触发条件成立的情况下，也可以将采样跟踪设置设置到 QnUCPU 中。	U	新增	QnU

(9) 锁存区

附表 3.10 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM900	电源关断文件	OFF ：无电源关断文件 ON ：电源关断文件存在	• 如果电源关断在访问过程中有文件存在，变为 ON。	S（状态改变）/U	新增	QnA
SM910	RKEY 注册标志	OFF ：键盘输入未注册 ON ：键盘输入注册	• 注册键盘输入时，变为 ON。 • 如果键盘输入未注册时，OFF，	S（指令执行）	新增	

(10)A → Q/QnA 转换的对应

ACPU 的特殊继电器 M9000 ～ M9255 在通过 A → Q/QnA 转换进行了转换之后，所对应的特殊继电器为 SM1000 ～ SM1255。
（但是，基本型 QCPU、冗余 CPU 和通用型 QCPU 不支持 A → Q/QnA 转换。）
这些特殊继电器都是由系统设定的，用户程序不能设定它们。
要使用用户程序将它们变为 ON/OFF，请将程序中的特殊继电器更换为 QCPU/QnACPU 的相应继电器。
但是，SM1084 和 SM1200 到 SM1255（对应转换前的 M9084 和 M9200 到 M9255）中的一些继电器不能由用户程序变为 ON/OFF，即使在转换之前它们可以由用户程序变为 ON/OFF 的话。关于 ACPU 特殊继电器的详细信息，见单独 CPU 的用户手册，和 MELSECNET 或者 MELSECNET/B 数据链接系统参考手册。

要 点

当转换后的特殊继电器用于高性能型 QCPU 和过程控制 CPU 时，处理时间可能会变长。当转换后的特殊继电器未被使用时，在 GX Developer 参数的可编程控制器系统设置内，取消对 “A-series CPU compatibility setting”的选择。

备 注

下面是对 “用于改进的特殊继电器”栏中，特殊继电器的附加解释。

1) 当提供了用于改进的特殊继电器时，软元件号应该改为提供的 QCPU/QnACPU 特殊继电器。
2) 当提供了□时，转换后的特殊继电器可以用于此软元件号。
3) 当提供了☒时，此软元件号不能和 QCPU/QnACPU 一起工作。

在高性能模式 QCPU 以及过程 CPU 中使用转换后的特殊继电器时，需要耗费处理时间。在不使用转换后的特殊继电器时，应在 GX Developer 的可编程控制器参数的可编程控制器系统设置中将 “A 系列 CPU 兼容设置”的勾选取消。

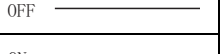
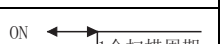
附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9000	SM1000	-	保险丝熔断	OFF : 正常 ON : 模块有熔断的保险丝	• 当有一个或者多个保险丝已熔断的输出模块时，接通。 • 如果自此以后情况恢复为正常，仍保持为 ON。 • 对远程 I/O 站，也检查输出模块的保险丝情况。	QnA Qn(H) QnPH
M9002	SM1002	-	I/O 模块验证出错	OFF : 正常 ON : 出错	• 如果 I/O 模块的状态和电源上电时输入的状态不同，接通。 • 如果自此以后情况恢复为正常，保持为 ON。 • 对远程 I/O 站模块，也进行 I/O 模块校验。 • 只有当特殊寄存器 SD1116 到 SD1123 被复位时，复位才被允许。	
M9004	SM1004	-	NIMI 链接出错	OFF : 正常 ON : 出错	• 即使只在安装的 AJ71PT32(S3) 模块中的一个检测到 MINI(S3) 链接出错，变为 ON。 • 如果自此以后情况恢复为正常，仍保持为 ON。	QnA

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9005	SM1005	-	AC DOWN 检测	OFF : AC DOWN 未检测到 ON : AC DOWN 检测到	• 如果在使用 AC 电源模块的过程中发生 20ms 以内的瞬间掉电，变为 ON。 • 当电源被切换到 OFF，然后再切换为 ON 时，复位。	QnA Qn(H) QnPH
					• 如果在使用 DC 电源模块的过程中发生 10ms 以内的瞬间掉电，变为 ON。 • 当电源是被切换到 OFF，然后再切换为 ON 时，复位。	Qn(H) QnPH
					• 如果在使用 DC 电源模块的过程中发生 1ms 以内的瞬间掉 电，变为 ON。 • 当电源是被切换到 OFF，然后再切换为 ON 时，复位。	QnA
M9006	SM1006	-	电池电量不足	OFF : 正常 ON : 电池电量不足	• 当电池电压下降到或者低于指定值时，变为 ON。 • 当电池电压恢复为正常，变为 OFF。	QnA Qn(H) QnPH
M9007	SM1007	-	电池电量不足锁存	OFF : 正常 ON : 电池电量不足	• 当电池电压下降到或者低于指定值时，变为 ON。 • 如果电池电压恢复为正常，仍保持为 ON。	
M9008	SM1008	SM1	自检测出错	OFF : 没有出错 ON : 出错	• 当作为自检测的结果发现出错时，接通。	
M9009	SM1009	SM62	报警器检测	OFF : 无 F 号检测到 ON : 有 F 号检测到	• 当 SETF 指令的的 OUTF 被执行时，接通。 • 当 SD1124 数据是零时，切换为关断。	
M9011	SM1011	SM56	操作出错标志	OFF : 没有出错 ON : 出错	• 当操作出错发生在应用指令的执行过程中时，接通。 • 如果自此以后情况恢复为正常，仍保持为 ON。	
M9012	SM1012	SM700	进位标志	OFF : 进位 OFF ON : 进位 ON	• 用在应用指令中的进位标志。	
M9016	SM1016	×	数据内存清除标志	OFF : 忽略 ON : 输出清除	• 当 SM1016 是 ON 时，从计算机等设备上以远程运行模式清除包括锁存范围（特殊继电器和特殊寄存器以外的其他软元件）在内的数据内存。	
M9017	SM1017	×	数据内存清除标志	OFF : 忽略 ON : 输出清除	• 当 SM1017 是 ON 时，从计算机等设备上以远程运行模式清除未锁存的数据内存（特殊继电器和特殊寄存器以外的其他软元件）。	
M9020	SM1020	-	0 号用户定时时钟	n2个扫描周期 n2个扫描周期 n1个扫描周期	• 在预先定义的扫描周期的间隔上重复 ON/OFF 的继电器。 • 当电源接通或者执行复位时，时钟从 OFF 启动。用 DUTY 指令设定 ON/OFF 的间隔。 DUTY n1 n2 SM1020 n1: ON 扫描周期间隔 n2: OFF 扫描周期间隔	
M9021	SM1021	-	1 号用户定时时钟			
M9022	SM1022	-	2 号用户定时时钟			
M9023	SM1023	-	3 号用户定时时钟			
M9024	SM1024	-	4 号用户定时时钟			
M9025	SM1025	-	时钟数据设置请求	OFF : 忽略 ON : 设置请求存在	• SM1025 从 OFF 变为 ON 的扫描周期中，在 END 指令执行之后，将存储在 SD1025 到 SD1028 中的时钟数据写到 CPU 模块。	
M9026	SM1026	-	时钟数据出错	OFF : 没有出错 ON : 出错	• 由时钟数据 (SD1025 到 SD1028) 出错接通	
M9027	SM1027	-	时钟数据显示	OFF : 忽略 ON : 显示	• 从 SD1025 到 SD1028 中读时钟数据，月、日、小时、分钟和秒显示在 CPU 模块前面的 LED 显示器上。	Q3A Q4A Q4AR
M9028	SM1028	-	时钟数据读请求	OFF : 忽略 ON : 读取请求	• 当 SD1028 是 ON 时，以 BCD 格式读时钟数据到 SD1025 到 SD1028 中。	QnA Qn(H) QnPH
M9029	SM1029	×	数据通讯请求的批量处理	OFF : 批量处理未执行 ON : 批量处理执行	• SM1029 继电器接通，在此扫描周期的 END 处理中，使用顺控程序去处理在此扫描周期中接收到的所有数据通讯请求。 • 数据通讯请求的批量处理可以在运行过程中接通和关断。 • 默认是 OFF（按照数据通讯请求的接收顺序，每个 END 处理一次执行一个）。	
M9030	SM1030	-	0.1 秒时钟	0.05s 0.05s	• 生成 0.1 秒，0.2 秒，1 秒和 2 秒时钟。 • 不是在每个扫描周期都接通或者关断，但是即使在扫描周期过程中，如果相应时间已经用完，也会接通或者关断。 • 当可编程控制器电源接通或者执行 CPU 模块复位时，从 OFF 启动。	
M9031	SM1031	-	0.1 秒时钟	0.1s 0.1s		
M9032	SM1032	-	1 秒时钟	0.5s 0.5s		
M9033	SM1033	-	2 秒时钟	1s 1s		

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9034	SM1034	-	2n 分钟时钟 (1 分钟时钟)*		• 按照在 SD414 钟指定的秒数，在 ON 和 OFF 之间交替。(默认 :n=30) • 不是在每个扫描周期都接通或者关断，但是即使在在扫描周期过程中，如果相应时间已经用完，也会接通或者关断。 • 当可编程控制器电源接通或者执行 CPU 模块复位时，以 OFF 启动。	QnA Qn(H) QnPH
M9036	SM1036	-	常 ON		• 被用作顺控程序中初始化和应用指令的虚拟触点。 • SM1038 和 SM1037 被接通和关断，而不管 CPU 模块前面的钥匙开关的位置如何。如果钥匙开关处于停止位置 ，则切换为关断。如果钥匙开关不在停止位置 S，M1038 只接通一个扫描周期，而 SM1039 只关断一个扫描周期。	
M9037	SM1037	-	常 OFF			
M9038	SM1038	-	只 ON1 个扫描周期在 RUN 之后			
M9039	SM1039	-	RUN 标志 (在 RUN 之后，只 OFF1 个扫描周期)			
M9040	SM1040	SM206	PAUSE 状态触点	OFF : PAUSE 禁止 ON : PAUSE 允许	• 当 RUN 钥匙开关在 PAUSE 位置或者 PAUSE 触点已接通，并且如果 SM204 是 ON，则 PAUSE 模式被设定，并且 SM206 接通。	
M9041	SM1041	SM204	USE 状态触点	OFF : PAUSE 无效 ON : PAUSE 有效		
M9042	SM1042	SM203	STOP 状态触点	OFF : STOP 无效 ON : STOP 有效	• 当 RUN 钥匙开关或者 RUN/STOP 开关位于停止位置 , 接通。	
M9043	SM1043	SM805	采样跟踪结束	OFF : 采样跟踪进行中 ON : 采样跟踪结束	• 在 STRA 指令执行之后，采样跟踪执行了参数预设的次数时接通。 复位当 STRAR 指令执行时。	
M9044	SM1044	SM803	采样跟踪	OFF→ON STRA 和执行相同 ON→OFF STRAR 和执行相同	• 接通 / 关断 SM803 可以执行 STRA STRA / STRAR 指令。 (SM803 是由外围设备强制接通 / 关断的。) 当从 OFF 切换到 ON: STRA 指令 当从 ON 切换到 OFF: STRAR 指令 存储在 SD1044 中的值被用作采样跟踪的条件。 在扫描时，时间→ 时间 (10ms 为单位)	
M9045	SM1045	×	看门狗定时器 (WDT) 复位	OFF : 不复位 WDT ON : 复位 WDT	• SM1015 继电器接通以复位 WDT，当 ZCOM 指令和数据通讯请求批量处理被执行 (当扫描时间超出 200ms 时被使用)。	
M9046	SM1046	SM802	采样跟踪	OFF : 跟踪未进行 ON : 跟踪进行中	• 在采样跟踪过程中切换为 ON。	
M9047	SM1047	SM801	采样跟踪准备	OFF : 采样跟踪中断 ON : 采样跟踪启动	• 采样跟踪未执行，除非 SM801 变为 ON。 当 SM801 变为 OFF 时 , 采样跟踪被中断。	
M9049	SM1049	SM701	切换字符输出数	OFF : 输出直到碰到 NULL 代码 ON : 16 字符输出	• 当 SM701 是 OFF 时，输出字符直到碰到 NULL(00H) 代码。 • 当 SM701 是 ON 时，输出 16 字符的 ASCII 代码。	
M9051	SM1051	×	CHG 指令执行禁止	OFF : 允许 ON : 禁止	• 切换为 ON 以禁止 CHG 指令。 • 切换为 ON，当程序发送被请求时。 当发送结束时自动切换到 OFF。	
M9052	SM1052	×	SEG 指令开关	OFF : 7SEG 段显示 ON : I/O 部分刷新	• 当 SM1052 是 ON 时，SEG 指令以 I/O 部分刷新指令执行。 当 SM1052 是 OFF 时，SEG 指令以 7-SEG 显示指令执行。	
M9054	SM1054	SM205	STEP RUN 标志	OFF : SETP RUN 无效 ON : SETP RUN 有效	• 当 RUN 钥匙开关处于 SETPRUN 位置 , 切换为 ON。	QnA
M9055	SM1055	SM808	状态锁存结束标志	OFF : 未结束 ON : 结束	• 当状态锁存结束时 , 接通。 • 由复位指令关闭。	

*: 1 分钟时钟表示 ACPU 的特殊继电器 (M9034) 的名称。

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9056	SM1056	×	主端 P、I 设置请求	OFF ：P、I 设置正被请求以外的情况 ON ：P、I 设置正被请求	• 在 RUN 过程中，当其他程序的发送（例如，当主程序正在运行时的子程序）结束之后提供 P、I 设置请求。当 P、I 设置结束后自动被切换到 OFF。	QnA Qn(H) QnPH
M9057	SM1057	×	从端 P、I 设置请求	OFF ：P、I 设置正被请求以外的情况 ON ：P、I 设置正被请求		
M9058	SM1058	×	主端 P、I 设置结束	在 P、I 设置结束时立即为 ON	• 当 P、I 设置结束时立即变为 ON，然后重新变为 OFF。	
M9059	SM1059	×	子程序 P、I 设置结束	在 P、I 设置结束时立即为 ON		
M9060	SM1060	×	子程序 2P、I 设置请求	OFF ：P、I 设置正被请求之外的情况 ON ：P、I 设置正被请求	• 在 RUN 过程中，当其他程序的发送（例如，当主程序正在运行时的子程序）结束之后提供 P、I 设置请求。当 P、I 设置结束后自动被切换到 OFF。	
M9061	SM1061	×	子程序 3P、I 设置请求	OFF ：P、I 设置正被请求之外的情况 ON ：P、I 设置正被请求		
M9065	SM1065	SM711	分开发送状态	OFF ：分开处理没有进行 ON ：在分开处理过程中	• 当通过分开处理执行到 AD57(S1)/AD58 的画布画面发送时接通，在分开处理结束时关断	QnA
M9066	SM1066	SM712	发送处理切换	OFF ：批量处理 ON ：分开处理	• 当到 AD57(S1)/AD58 的画布画面发送是通过分开处理执行时接通。	
M9070	SM1070	×	A8UPU/A8PUJ 需要的查找时间 *	OFF ：读取时间没有缩短 ON ：读取时间缩短	• 变为 ON 以缩短在 A8UPU/A8PUJ 中的查找时间。（在这种情形中，扫描时间增加 10 %.）	QnA Qn(H) QnPH
M9081	SM1081	SM714	通讯请求注册区域 BUSY 信号	OFF ：通讯请求注册区域中的空余空间 ON ：在通讯请求注册区域中无空余空间	• 与连接到 AJ71PT32-S3、A2C 或者 A52G 的远程端子模块通讯允许 / 禁止的指示。	QnA
M9084	SM1084	×	出错检查	OFF ：出错检查执行 ON ：没有出错检查	它设定当 END 指令被处理时是否执行下面的出错检查（设定 END 指令处理时间）。 • 检查保险丝是否断开。 • 检查电池 • I/O 模块的验证检查	QnA Qn(H) QnPH
M9091	SM1091	×	指令出错标志	OFF ：没有出错 ON ：出错	• 当操作出错的详细因素被存储到 SD1091 中时，变为 ON。 • 如果自此以后情况恢复为正常，保持为 ON。	
M9094	SM1094	SM251	I/O 更换标志	OFF ：更换 ON ：无更换	• 在将要被更换的 I/O 模块的起始 I/O 号码被设定到 SD251 之后，当 SM251 变为 ON 时，I/O 模块可以在运行（电源接通）中更换。（一个设置只允许更换一个模块。） • 在程序或者外围设备测试模式中切换为接通，以便在 CPU 为 RUN 过程中更换模块。在外围设备测试模式中切换到接通以便在 CPU 停止过程中更换模块。 • 在 I/O 模块更换结束之前，不能改变 RUN/STOP 模式。	QnA
M9100	SM1100	SM320	SFC 程序存在 / 不存在	OFF ：SFC 程序被使用 ON ：SFC 程序停止	• 如果 SFC 程序被注册，接通。 • 如果没有注册，关断。	QnA Qn(H) QnPH
M9101	SM1101	SM321	启动 / 停止 SFC 程序	OFF ：SFC 程序停止 ON ：SFC 程序启动	• SM320 中的值被设置为初始值。（当 SFC 程序存在时，继电器自动变为 ON。） • 当此继电器从 ON 变为 OFF 时，SFC 程序的执行停止。 • 当此继电器从 OFF 变为 ON 时，恢复 SFC 程序的执行。	
M9102	SM1102	SM322	SFC 程序启动程序	OFF ：初始化启动 ON ：继续运行	• 可编程控制器参数对话框的 SFC 设置中的 SFC 程序启动模式被设定为初始值。 在初始化启动时：OFF 在连续启动：ONx	

*：A8UPU/A8PUJ 不能用于 QCPU/QnACPU。

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9103	SM1103	SM323	存在 / 不存在连续转移	OFF : 连续转移无效 ON : 连续转移有效	• 设定在没有设定 SFC 信息软元件的 “continuous transition bit”的地方是否执行块的连续转移。	QnA Qn(H) QnPH
M9104	SM1104	SM324	连续转移挂起标志	OFF : 当转移结束时 ON : 当没有转移时	• 在运行在连续转移模式或者在连续转移过程中为 OFF，当连续转移没有执行时为 ON。 • 常 ON，当运行在无连续转移模式中时。	
M9108	SM1108	SM90	步转移警戒狗定时器启动（等同于 SD90）	OFF : 警戒狗定时器复位 ON : 警戒狗定时器复位启动	• 当步转移警戒狗定时器的测量启动时，变为 ON。将此继电器置为 OFF，以复位步转移警戒狗定时器。	
M9109	SM1109	SM91	步转移警戒狗定时器启动（等同于 SD91）			
M9110	SM1110	SM92	步转移警戒狗定时器启动（等同于 SD92）			
M9111	SM1111	SM93	步转移警戒狗定时器启动（等同于 SD93）			
M9112	SM1112	SM94	步转移警戒狗定时器启动（等同于 SD94）			
M9113	SM1113	SM95	步转移警戒狗定时器启动（等同于 SD95）			
M9114	SM1114	SM96	步转移警戒狗定时器启动（等同于 SD96）			
M9180	SM1180	SM825	活动步采样跟踪结束标志	OFF : 跟踪启动 ON : 跟踪结束	• 当所有指定块的采样跟踪结束时设置，当采样跟踪启动时复位。	QnA
M9181	SM1181	SM822	活动步采样跟踪执行标志	OFF : 没有正在执行跟踪 ON : 跟踪执行执行中	• 当采样跟踪正在执行时置位。当采样跟踪结束或者挂起时复位。	
M9182	SM1182	SM821	活动步采样跟踪允许	OFF : 跟踪禁止 / 挂起 ON : 跟踪允许	• 选择采样跟踪执行允许 / 禁止。 ON : 允许采样跟踪执行。 OFF : 禁止采样跟踪执行。如果在采样跟踪执行过程中关断，跟踪被挂起。	
M9196	SM1196	SM325	操作输出块停止	OFF : 线圈输出 OFF ON : 线圈输出 ON	• 选择执行块停止时的操作输出。 ON : 通过使用在块停止时正被执行的步的操作输出，保持正被使用的线圈的 ON/OFF 状态。 OFF : 所有的线圈输出关断。(SET 指令的操作输出被保留，而不管 M9196 的 ON/OFF 状态如何。)	QnA Qn(H) QnPH
M9197	SM1197	×	在保险丝熔断和 I/O 验证出错显示之间切换	SM1197 / SM1198 / 要显示的 I/O 号 OFF / OFF / X/Y0 到 7F0 ON / OFF / X/Y800 到 FF0 OFF / ON / X/Y1000 到 17F0 ON / ON / X/Y1800 到 1FF0	依据 SM1197 和 SM1198 的 ON/OFF 的组合，在保险丝熔断模块存储寄存器（SD1100 到 SD1107）和 I/O 模块验证出错存储寄存器（SD1116 到 SD1123）中的 I/O 号之间切换。	
M9198	SM1198	×				
M9199	SM1199	×	运行中采样跟踪 / 状态锁存的数据恢复	OFF : 数据恢复禁止 ON : 数据恢复允许	• 当执行采样跟踪 / 状态锁存时，在重启时恢复存储在 CPU 模块中的设置数据。 • SM1199 应该为 ON 以便重新执行。（当重新从外围设备写数据时不需要。）	

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9200	SM1200	-	ZNRD 指令 (LRDP 指令用于 ACPU) 接收 (用于主站)	OFF : 未接收 ON : 接收	• 取决于是否接收到 ZNRD(字软元件读)指令。 • 在程序用作 ZNRD 指令的互锁。 • 使用 RST 指令去复位。	
M9201	SM1201	-	ZNRD 指令 (LRDP 指令用于 ACPU) 结束 (用于主站)	OFF : 未结束 ON : 结束	• 取决于 ZNRD(字软元件读)指令执行是否结束。 • 在 ZNRD 指令结束之后，用作复位 M9200 和 M9201 的条件触点。 • 使用 RST 指令去复位。	
M9202	SM1202	-	ZNWR 指令 (LWTP 指令用于 ACPU) 接收 (用于主站)	OFF : 未接收 ON : 接收	• 取决于 ZNWR(字软元件写)指令是否已经被接收。 • 在程序中用作用于 ZNWR 指令的互锁。 • 使用 RST 指令去复位。	
M9203	SM1203	-	ZNWR 指令 (LWTP 指令用于 ACPU) 结束 (用于主站)	OFF : 未接收 ON : 结束	• 取决于 ZNWR(字软元件写)指令执行是否结束。 • 在 ZNWR 指令结束之后，用作复位 M9202 和 M9203 的条件触点。 • 使用 RST 指令去复位。	
M9204	SM1204	-	ZNRD 指令 (LRDP 指令用于 ACPU) 接收 (用于本地站)	OFF : 未接收 ON : 结束	• 接通表示 ZNRD 指令在本地站结束。	
M9205	SM1205	-	ZNWR 指令 (LWTP 指令用于 ACPU) 接收 (用于本地站)	OFF : 未结束 ON : 结束	• 接通表示 ZNWR 指令在本地站结束。	
M9206	SM1206	-	主机站链接参数出错	OFF : 正常 ON : 异常	• 取决于主机的链接参数设置是否有效。	
M9207	SM1207	-	链接参数检查结果	OFF : 匹配 ON : 不匹配	• 取决于两级系统中第二级主站的链接参数设置是否和三级系统中第三级主站的链接参数设置相匹配。(只对三级系统中的主站有效。)	
M9208	SM1208	-	设置主站 B 和 W 传送范围 (只用于较低级链接主站)	OFF : 发送到第 2 级和第 3 级 ON : 只发送到第 2 级	• 取决于高一级链接主站 (本站) 控制的 B 和 W 数据是否被发送到低一级链接本地站 (第三级站)。 当 SM1208 是 OFF..... B 和 W 的本站被发送到第三级站。 当 SM1208 是 ON..... 和 W 的本站不发送到第三级站。	QnA
M9209	SM1209	-	链接参数检查命令 (只用于低一级链接主站)	OFF : 执行检查功能 ON : 检查不执行	• 设为 ON，以便不匹配高一级和低一级链接的 B 和 W。 • 当 SM1209 是 ON，不检查高一级和低一级链接的链接参数。 • 当 SM1209 是 OFF，检查高一级和低一级链接的链接参数。	
M9210	SM1210	-	链接卡出错 (用于主站)	OFF : 正常 ON : 异常	控制是否执行取决于链接卡硬件是否有故障。	
M9211	SM1211	-	链接模块出错 (用于本地站使用)	OFF : 正常 ON : 异常	控制是否执行取决于链接卡硬件是否有故障。	
M9224	SM1224	-	链接状态	OFF : 运行中 ON : 离线，站到站的测试，或者自环路回送测试	取决于主站是运行中还是离线，或者是处于站到站的测试或自环路回送测试模式中。	
M9225	SM1225	-	正向环路出错	OFF : 正常 ON : 异常	取决于正向环路电缆的出错条件。	
M9226	SM1226	-	反向环路出错	OFF : 正常 ON : 异常	取决于反向环路电缆的出错条件。	
M9227	SM1227	-	环路测试状态	OFF : 未执行 ON : 正向或者反向环路测试执行中	取决于主站是执行正向环路测试，还是反向环路测试。	
M9232	SM1232	-	本地站操作状态	OFF : RUN 或者 STEP RUN 状态 ON : STOP 或者 PAUSE 状态	依据本地站处于 STOP 或者 PAUSE 模式，来执行控制。	
M9233	SM1233	-	本地站出错检测状态	OFF : 没有出错 ON : 出错检测	取决于本地站是否在其他站中检测到出错。	

附表 3.11 特殊继电器列表

ACPU 特殊继电器	转换后的特殊继电器	用于改进的特殊继电器	名称	含义	详细信息	相应的 CPU
M9235	SM1235	-	本地站，远程 I/O 站参数出错检测状态	OFF ：没有出错 ON ：出错检测	取决于本地或者远程 I/O 站是否在主站检测到任何链接参数出错。	QnA
M9236	SM1236	-	本地站，远程 I/O 站初始化通讯状态	OFF ：无通讯 ON ：通讯进行中	取决于本地或者远程 I/O 站和主站初始化通讯的结果。（参数通讯等。）	
M9237	SM1237	-	本地站，远程 I/O 站出错	OFF ：正常 ON ：异常	取决于本地或者远程 I/O 站的出错条件。	
M9238	SM1238	-	本地站，远程 I/O 站正向或者反向环路出错	OFF ：正常 ON ：异常	取决于本地或者远程 I/O 站的正向和反向环路电缆的出错条件。	
M9240	SM1240	-	链接状态	OFF ：运行中 ON ：离线，站到站测试，或者自环路回送测试	取决于本地站是运行中还是处于离线，或者是处于站到站测试或自环路回送测试模式中。	
M9241	SM1241	-	正向环路电缆出错	OFF ：正常 ON ：异常	取决于正向环路电缆的出错条件。	
M9242	SM1242	-	反向环路电缆出错	OFF ：正常 ON ：异常	取决于反向环路电缆的出错条件。	
M9243	SM1243	-	环路回送实现	OFF ：环路回送未执行 ON ：环路回送实现	取决于本地站上是否正在发生环路回送。	
M9246	SM1246	-	数据未接收	OFF ：接收 ON ：未接收	取决于数据是否已经从主站中接收到。	
M9247	SM1247	-	数据未接收	OFF ：接收 ON ：未接收	取决于在三级系统，第三级站是否已经从它的主站接收到数据。	
M9250	SM1250	-	参数未接收	OFF ：接收 ON ：未接收	取决于是否已经从主站接收到链接参数。	
M9251	SM1251	-	链接继电器	OFF ：正常 ON ：异常	取决于本地站的数据链接情况。	
M9252	SM1252	-	环路测试状态	OFF ：未执行 ON ：正向或者反向环路测试执行进行中	取决于本地站是否正在执行正向或者反向环路测试。	
M9253	SM1253	-	主站操作状态	OFF ：RUN 或者 STEP RUN 状态 ON ：STOP 或者 PAUSE 状态	控制是否执行，取决于主站是处于 STOP 模式还是处于 PAUSE 模式。	
M9254	SM1254	-	本站以外的本地站操作状态	OFF ：RUN 或者 STEP RUN 状态 ON ：STOP 或者 PAUSE 状态	控制是否执行，取决于本站以外的主站是处于 STOP 模式还是处于 PAUSE 模式。	
M9255	SM1255	-	本站以外的本地站出错	OFF ：正常 ON ：异常	取决于本站以外的本地站是否处于出错状态。	

(11)过程控制控制指令

附表 3.12 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1500	保持模式	OFF ：不保持 ON ：保持	• 指定当 S.IN 指令范围检查发生超限时是否保持输出值。	U	新增	Q4AR QnPH QnPRH
SM1501	保持模式	OFF ：不保持 ON ：保持	• 指定当 S.OUT 指令范围检查发生超限时是否保持输出值。	U	新增	

(12)对于冗余系统（本站系统 CPU 信息 *1）

SM1510 到 SM1599 只对冗余系统有效。

对单独系统，所有的继电器都为 OFF。.

附表 3.13 特殊继电器列表

号码	名称	含义	解释	设置方（当被设定时	相应的 ACPU M9□□□	相应的 CPU
SM1510	运行模式	OFF ：冗余系统备用模式，独立系统 ON ：冗余系统独立模式	• 当运行模式是冗余系统分开时，变为 ON。	S（每次 END）	新增	Q4AR QnPRH
SM1511	电源接通时的启动模式	OFF ：系统固定模式 ON ：先前控制系统锁存模式	• 当当电源变为 ON 时冗余系统的启动模式是先前控制系统锁存模式时，变为 ON。	S（初始化）	新增	Q4AR
SM1512	CPU 启动时的启动模式	OFF ：初始化启动 ON ：热启动	• 当冗余系统被启动时，CPU 模块的运行模式是热启动时，变为 ON。	S（初始化）	新增	
SM1511	系统 A 识别标志	• 区分系统 A 和系统 B。 • 如果热备电缆断开，系统识别被保持在先前的状态。		S（初始化）	新增	QnPRH
SM1512	系统 B 识别标志					
SM1513	CPU 启动时操作状态的	OFF ：初始化启动 ON ：热启动	• 当冗余系统实际启动时，CPU 模块的运行模式是热启动时，变为 ON。	S（初始化）	新增	Q4AR
	调试模式状态标志	OFF ：不处于调试模式 ON ：调试模式	• 当冗余系统运行模式被设定到调试模式，变为 ON。	S（初始化）	新增	QnPRH
SM1514	CPU 模块更换时的运行模式	OFF ：初始化启动 ON ：热启动	• 当为冗余系统进行 CPU 模块运行切换，操作是热启动时，变为 ON。	S（初始化）	新增	Q4AR
SM1515	输出保持模式	OFF ：输出复位 ON ：输出保持	• 当在停止出错过程中输出模式是输出保持时，变为 ON。	S（每次 END）	新增	
SM1516	操作系统状态	OFF ：控制系统 ON ：待机系统	• 当 CPU 模块运行系统状态是待机系统时，变为 ON。	S（状态改变）	新增	
SM1515	控制系统判断标志	• 表示运行系统状态。 • 如果在冗余系统运行过程中，跟踪电缆断开，此标志状态不变。		S（状态改变）	新增	QnPRH
SM1516	待机系统判断标志					

	系统A	系统B	当TRK.CABLEERR.发生时（出错代码：6210）发生(未知)
SM1511	ON	OFF	OFF
SM1512	OFF	ON	OFF

	控制系统	待机系统	当TRK.CABLE ERR.发生时（出错代码：6210）发生(未知)
SM1515	ON	OFF	OFF
SM1516	OFF	ON	OFF

*1: 本站 CPU 模块的信息被存储。

附

附录

附表 3.13 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1517	CPU 模块启动状态	OFF ：电源接通启动 ON ：操作系统开关启动	• 当 CPU 模块是操作系统开关启动时，变为 ON。 • 使用用户程序复位。	S（状态改变）/U	新增	Q4AR
			• 当 CPU 模块是由系统切换启动（从待机系统切换到控制系统），变为 ON。当待机系统是通过电源接通启动而被切换到控制系统，保持为 OFF。	S（状态改变）	新增	QnPRH
SM1518	跟踪执行模式	OFF ：批运行模式 ON ：进位模式	• 如果跟踪内存在 END 处正被使用，当此继电器变为 OFF，跟踪的启动被延迟，直到可执行。 • 如果跟踪内存在 END 处正被使用，当此继电器是变为 ON 时，启动的跟踪程序执行，直到下一次 END。	U	新增	Q4AR
	待机系统到控制系统切换状态标志	ON OFF 1个扫描周期	• 只要从待机系统切换到控制系统（只 ON1 个扫描周 期）发生，变为 ON。 • 此状态标志只能用于扫描周期执行类型程序。	S（每次 END）	新增	QnPRH
SM1519	先前控制系统识别标志	ON OFF 1个扫描周期	• 上次运行的控制系统是系统 B，如果电源上电，或者两个系统一起复位，则在 RUN 之后，只有系统 A 侧 ON1 个扫描周期。	S（每次 END）	新增	
SM1520	数据跟踪传送触发规格	OFF ：无触发器 ON ：由触发器	SM1520 块 1 SM1521 块 2 SM1522 块 3 SM1523 块 4 SM1524 块 5 SM1525 块 6 SM1526 块 7 SM1527 块 8 SM1528 块 9 SM1529 块 10 SM1530 块 11 SM1531 块 12 SM1532 块 13 SM1533 块 14 SM1534 块 15 SM1535 块 16 SM1536 块 17 SM1537 块 18 SM1538 块 19 SM1539 块 20 SM1540 块 21 SM1541 块 22 SM1542 块 23 SM1543 块 24 SM1544 块 25 SM1545 块 26 SM1520 块 27 SM1521 块 28 SM1522 块 29 <在 Q4AR 中> • 当用数据跟踪指令 S.TRUCK 发送数据时，目标块被指定为触发器。 <在 QnPRH 中> • 当数据发送是基于冗余参数对话框的跟踪设置时，目标块被指定为触发器。 • 当在跟踪设置中允许“Auto Tracking block No.1”时，SM1520 由系统在电源接通 /STOP 到 RUN 时将其变为 ON。在其他情况中，SM1520 到 SM1583 由用户变为 ON。	<在 Q4AR 中> U <在 QnPRH 中> S（初始化）/U	新增	Q4AR QnPRH
SM1521						
SM1522						
SM1523						
SM1524						
SM1525						
SM1526						
SM1527						
SM1528						
SM1529						
SM1530						
SM1531						
SM1532						
SM1533						
SM1534						
SM1535						
SM1536						
SM1537						
SM1538						
SM1539						
SM1540						
SM1541						
SM1542						
SM1543						
SM1544						
SM1545						
SM1546						
SM1547						
SM1548						

附表 3.13 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1549	数据跟踪传送链接规格	OFF ：无触发器 ON ：有触发器	SM1549 块 30 SM1550 块 31 SM1551 块 32 SM1552 块 33 SM1553 块 34 SM1554 块 35 SM1555 块 36 SM1556 块 37 SM1557 块 38 SM1558 块 39 SM1559 块 40 SM1560 块 41 SM1561 块 42 SM1562 块 43 SM1563 块 44 SM1564 块 45 SM1565 块 46 SM1566 块 47 SM1567 块 48 SM1568 块 49 SM1569 块 50 SM1570 块 51 SM1571 块 52 SM1572 块 53 SM1573 块 54 SM1574 块 55 SM1575 块 56 SM1576 块 57 SM1577 块 58 SM1578 块 59 SM1579 块 60 SM1580 块 61 SM1581 块 62 SM1582 块 63 SM1583 块 64 〈在 Q4AR 中〉 当数据是数据跟踪指令 S.TRUCK 发送时，目标块被指定为触发器。 〈在 QnPRH 中〉 • 当数据是基于冗余参数对话框的跟踪设置发送时，目标块被指定为触发器。 • 当在跟踪设置中允许“AutoTrackingblock No.1”时，SM1520 由系统在电源接通 /STOP 到 RUN 时将其变为 ON。在其他情况中，SM1520 到 SM1583 由用户变为 ON。	〈在 Q4AR 中〉 U 〈在 QnPRH 中〉 S（初始化）/U	新增	Q4AR QnPRH
SM1590	来自网络模块的切换状态	OFF ：正常 ON ：切换不成功	• 如果网络模块检测到网络故障并发送切换请求到本站 CPU 模块，当切换不能正常执行，变为 ON。	S（出错发生）	新增	Q4AR
	来自网络模块的系统切换允许 / 禁止标志	OFF ：系统切换请求发送模块不存在 ON ：存在有系统切换请求发送模块	• 当从网络模块上发送系统切换请求时，变为 ON。发送系统切换的模块号可以通过 SD1590 确认。 • SD1590 的各个位全部 OFF 时该继电器将 OFF。	S（每次 END）		QnPRH
SM1591	在系统切换时，待机系统出错检测禁止标志	ON ：在系统切换时新待机系统没有检测到出错 ON ：在系统切换时新待机系统检测到出错	当 SD1590 的所有位都为 OFF 时，变为 OFF。如果在系统切换过程中新待机站检测到 6210：STANDBY，此标志用于确认。 此可以应用到下列切换方法： • 从 GX Developer 上进行系统切换 • 使用专用指令进行系统切换 • 由智能功能模块进行系统切换	U	新增	
SM1592	允许 / 禁止用户系统切换	OFF ：禁止用户系统切换 ON ：允许用户系统切换	• 此标志允许用户从 GX Developer 或者通过专用指令 (SP.CONTSW) 执行系统切换。	U	新增	
SM1593	待机系统 CPU 的扩展基板的访问设置	OFF ：出错 ON ：无处理	独立模式中，待机系统 CPU 访问扩展基板上安装的智能功能模块缓冲存储区时的操作设置。 OFF ：待机系统 CPU 访问扩展基板上安装的智能功能模块缓冲存储区时，返回“OPERATION ERROR”（出错代码： 4112）。 ON ：待机系统 CPU 访问扩展基板上安装的智能功能模块缓冲存储区时，无处理。	U	新增	QnPRH*2
SM1595	复制内存到其他系统启动标志	OFF ：启动内存复制 ON ：无内存复制被启动	• 当 SM1595 从 OFF 变为 ON，从控制系统到待机系统的内存复制启动。注意，当 SM1595 从 OFF 变为 ON，内存复制并不启动，如果复制目标的 I/O 号（待机系统 CPU 模块：3D1H）没有存储到 SD1595。			QnPRH
SM1596	复制内存到其他系统状态标志	OFF ：内存复制未执行 ON ：内存复制执行	• 当内存被复制到其他系统，变为 ON。 • 当内存复制执行已经结束时，变为 OFF。	S（复制启动 / 完成）		

附表 3.13 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1597	内存复制到其他系统结束标志	OFF ：内存复制未结束 ON ：内存结束	• 到其他系统的内存复制一结束就变为 ON。	S（结束）/U	新增	QnPRH
SM1598	内存复制过程中标准 ROM 的复制内容	OFF ：复制标准 ROM 数据 ON ：标准 ROM 数据未被复制	• 如果由用户设定为 ON，当执行内存复制时，标准 ROM 的数据并不被复制到其他系统。	U		

*2: 以序列号的高 5 位为 “09012”以后的 CPU 为对象。

(13)用于冗余系统（其他系统 CPU 信息 *1)

SM1600 到 SM1650 只对 CPU 冗余系统备用模式有效，因此不能在独立模式中刷新它们。备用模式或者独立模式对。SM4651 到 SM1699 都有效。对单独系统，SM1600 到 SM1699 都为关断。

附表 3.14 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的本站 SM□□ *2	相应的 CPU
SM1600	检测出错	OFF ：无出错 ON ：有出错	• 如果在检测结果中发生出错，则接通。（包括外部检测） • 保持为接通，即使此后恢复为正常。	S（每次 END）	SM0	Q4AR
	其他系统出错标志	OFF ：无出错 ON ：有出错	• 当在冗余系统中发生出错时，接通。出错检查（接通 SD1600 的单个位。） • 当无出错存在时，关断。	S（每次 END）	-	QnPRH
SM1601	自检测出错	OFF ：无自检测出错 ON ：有自检测出错	• 当在自检测结果中发生出错时，接通。 • 即使此后恢复为正常，保持为接通。	S（每次 END）	SM1	Q4AR
SM1605	出错公共信息	OFF ：无出错公共信息 ON ：有出错公共信息	• 当有出错公共信息时，并且 SM1600 接通，接通。	S（每次 END）	SM5	
SM1610	其它系统的检测出错	OFF ：无出错 ON ：有出错	• 当发生检测出错时，接通。（包括报警器为 ON 时的出错检测，和 CHK 指令检测到的出错。） • 对应其他系统的 SM0 的状态。	S（每次 END）	SM0	QnPRH
SM1611	其它系统的自检测出错。	OFF ：无自检测出错发生 ON ：有自检测出错发生	• 当发生自检测出错时，接通。（不包括报警器为 ON 时的出错检测，和 CHK 指令检测到的出错。） • 对应其他系统的 SM1 的状态。	S（每次 END）	SM1	
SM1615	其它系统的公共出错信息	OFF ：无公共出错信息存在 ON ：有公共出错信息存在	• 当其他系统中有公共出错信息时，接通。 • 对应其他系统的 SM5 的状态。	S（每次 END）	SM5	
SM1616	出错个别信息	OFF ：无出错个别信息 ON ：有出错个别信息	• 当有出错个别信息时，并且 SM1600 接通，接通。	S（每次 END）	SM16	Q4AR
SM1626	其它系统的出错个别信息	OFF ：无个别出错信息存在 ON ：有个别出错信息存在	• 其他系统中有个别出错信息时，接通。 • 对应其他系统的 SM16 的状态 。	S（每次 END）	SM16	QnPRH
SM1649	待机系统取消出错标志	OFF 到 ON: 待机系统的取消出错	通过将此继电器从 OFF 变为 ON，可以将待机系统 CPU 模块中发生的连续出错取消掉。 使用 SD1649 指定要被取消的出错的出错代码。	U	-	
SM1653	STOP 触点	STOP 状态	• 当处于 STOP 状态时，接通。	S（每次 END）	SM203	Q4AR
SM1654	PAUSE 触点	PAUSE 状态	• 当处于 PAUSE 状态时，接通。	S（每次 END）	SM204	
SM1655	STEP-RUN 触点	STEP-RUN 状态	• 当处于 STEP-RUN 状态时，接通。	S（每次 END）	SM205	

*1 存储其他系统 CPU 检测信息和系统信息。

*2 这给出了用于本站系统 CPU 的特殊继电器 (SM□□)。

(14)用于冗余系统（跟踪）

备用模式和独立模式对 SM1700 到 SM1799 都有效。
对单独系统都为 OFF。

附表 3.15 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1700	跟踪执行标志	OFF ：不可以执行 ON ：可以执行	• 当跟踪可以正常执行时，变为 ON。	S（状态改变）	新增	Q4AR
	传送结束标志	OFF ：传送未结束 ON ：传送结束	• 一旦块 1 到块 64 的传送结束，变为 ON 一个扫描周期。	S（状态改变）	新增	QnPRH
SM1709	在运行中程序更改冗余跟踪过程中手动切换禁止 / 允许设置	ON ：手动切换（禁止被删除） OFF ：手动切换禁止	(1) 将此继电器从 OFF 变为 N，以允许在运行中程序更改冗余跟踪过程中进行手动切换。 在手动切换禁止状态被删除之后，系统自动关闭 SM1709。 (2) 在运行中程序更换冗余跟踪过程中，发生了下列情况中的一种，则执行系统切换，而不管此继电器的状态如何。 • 电源关断、复位、硬件故障、CPU 停止出错。 (3) 在下列状态种，系统切换禁止状态也可以被此继电器删除。 • 多个块运行中程序更改冗余跟踪执行状态 • 文件批量运行中程序更改冗余跟踪执行状态	S（当执行时）/U		
SM1710	在运行中程序更改过程中发送跟踪数据允许标志	OFF ：无软元件跟踪 ON ：发送软元件存储区	(1) 设定在运行中程序更改冗余跟踪过程中，是否执行下列数据的跟踪。 • 软元件存储区 （包括 SM/SD，它们将自动执行跟踪） • PIDINIT 信息、S.PIDINIT 信息、SFC 信息 (2) SM1710 也可以用于设定当多个程序块的运行中更改或者批量的文件正在被执行以确保两个系统的一致性时，是否执行跟踪。 (3) 此 SM 也被跟踪数据从控制系统 CPU 模块发送到待机系统 CPU 模块。	U		

附表 3.15 特殊继电器列表

号码	名称	含义	解释			设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1712	传送触发器结束标志	OFF ：传送未结束 ON ：传送结束	SM1712	块 1	〈在 Q4AR 中〉 只在相应的数据传送结束时的一个扫描周期中变为 ON。 〈在 QnPRH 中〉 只在相应的块传送结束时的一个扫描周期中变为 ON。	S（状态改变）	新增	Q4AR QnPRH
SM1713			SM1713	块 2				
SM1714			SM1714	块 3				
SM1715			SM1715	块 4				
SM1716			SM1716	块 5				
SM1717			SM1717	块 6				
SM1718			SM1718	块 7				
SM1719			SM1719	块 8				
SM1720			SM1720	块 9				
SM1721			SM1721	块 10				
SM1722			SM1722	块 11				
SM1723			SM1723	块 12				
SM1724			SM1724	块 13				
SM1725			SM1725	块 14				
SM1726			SM1726	块 15				
SM1727			SM1727	块 16				
SM1728			SM1728	块 17				
SM1729			SM1729	块 18				
SM1730			SM1730	块 19				
SM1731			SM1731	块 20				
SM1732			SM1732	块 21				
SM1733			SM1733	块 22				
SM1734			SM1734	块 23				
SM1735			SM1735	块 24				
SM1736			SM1736	块 25				
SM1737			SM1737	块 26				
SM1738			SM1738	块 27				
SM1739			SM1739	块 28				
SM1740			SM1740	块 29				
SM1741			SM1741	块 30				
SM1742			SM1742	块 31				
SM1743			SM1743	块 32				
SM1744			SM1744	块 33				
SM1745			SM1745	块 34				
SM1746			SM1746	块 35				

附表 3.15 特殊继电器列表

号码	名称	含义	解释			设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1747	传送触发器结束标志	OFF ：传送未结束 ON ：传送结束	SM1747	块 36	〈在 Q4AR 中〉 只在相应的数据传送结束时的一个扫描周期中变为 ON。 〈在 QnPRH 中〉 只在相应的块传送结束时的一个扫描周期中变为 ON。	S（状态改变）	新增	Q4AR QnPRH
SM1748			SM1748	块 37				
SM1749			SM1749	块 38				
SM1750			SM1750	块 39				
SM1751			SM1751	块 40				
SM1752			SM1752	块 41				
SM1753			SM1753	块 42				
SM1754			SM1754	块 43				
SM1755			SM1755	块 44				
SM1756			SM1756	块 45				
SM1757			SM1757	块 46				
SM1758			SM1758	块 47				
SM1759			SM1759	块 48				
SM1760			SM1760	块 49				
SM1761			SM1761	块 50				
SM1762			SM1762	块 51				
SM1763			SM1763	块 52				
SM1764			SM1764	块 53				
SM1765			SM1765	块 54				
SM1766			SM1766	块 55				
SM1767			SM1767	块 56				
SM1768			SM1768	块 57				
SM1769			SM1769	块 58				
SM1770			SM1770	块 59				
SM1771			SM1771	块 60				
SM1772			SM1772	块 61				
SM1773			SM1773	块 62				
SM1774			SM1774	块 63				
SM1775			SM1775	块 64				

(15)冗余电源模块信息

附表 3.16 特殊继电器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU M9□□□	相应的 CPU
SM1780	电源 OFF 检测标志	OFF ：无输入电源为 OFF 状态的冗余电源模块 ON ：有输入电源为 OFF 状态的冗余电源模块	• 当一个或以上输入电源为 OFF 状态的冗余电源模块被检测到时，变为 ON。 • 如果 SD1780 的某个位为 ON 时，变为 ON。 • 如果 SD1780 的所有位都为 OFF 时，本继电器也 OFF。 • 当主基板不是冗余主基板（Q38RB）时，本继电器将变为 OFF。 0：执行与外围设备的通信 1：不执行与外围设备的通信	S（每次 END）	新增	Qn(H)*2 QnPH*2 QnPRH Rem
SM1781	电源模块故障检测标志	OFF ：没有检测到有故障的冗余电源模块 ON ：检测到有故障的冗余电源模块	• 当检测到一个或者更多有故障的冗余电源模块，变为 ON。 • 如果 SD1781 的某个位为 ON，本继电器将 ON。 • 如果 SD1781 的所有位都为 OFF，本继电器也将 OFF。 • 当主基板不是冗余主基板（Q38RB），本继电器变为 OFF。 0：执行与外围设备的通信 1：不执行与外围设备的通信	S（每次 END）		
SM1782	电源模块 1*1 用瞬间掉电检测标志	OFF ：未检测出瞬间掉电故障 ON ：检测出瞬间掉电	• 当到电源 1 或者 2 的输入电源的瞬间掉电被检测到一次或多次，变为 ON。在变为 ON 之后，如果电源从瞬间掉电中恢复，也将保持为 ON 状态。 • 当 CPU 模块启动时，将电源 1、电源 2 的标志（SM1782、SM1783）变为 OFF。 • 在其中一个冗余电源模块的输入电源为 OFF 时，输入电源为 OFF 的冗余电源模块相对应的标志将变为 OFF。 • 当主基板不是冗余主基板（Q38RB）时，本继电器变为 OFF。 0：执行与外围设备的通信 1：不执行与外围设备的通信	S（每次 END）		
SM1783	电源模块 2*1 用瞬间掉电检测标志					

*1：“电源 1”表示安装在冗余基板（Q38RB/Q68RB/Q65WRB）的 POWER1 插槽上的冗余电源模块（Q64RP）。
“电源 2”表示安装在冗余基板（Q38RB/Q68RB/Q65WRB）的 POWER2 插槽上的冗余电源模块（Q64RP）。

*2：以序列号的高 5 位为“04012”以后的 CPU 为对象。
多 CPU 系统配置中，以序列号的高 5 位为“07032”以后的 CPU 为对象。

附录 4　特殊寄存器列表

特殊寄存器 SD 是应用固化在可编程控制器中的内部寄存器。

由于这个原因，它们不能以普通寄存器那样的方式被顺控程序使用。然而，数据可以按照需要被写入以控制 CPU 模块。

如果没有特殊指定，数据以 BIN 值的形式存储在特殊寄存器内。

列表的各项目的说明如以下附表 3.16 所示。

附表 4.1 特殊寄存器列表的说明

条目	条目的功能
号码	• 表示特殊寄存器号
名称	• 表示特殊寄存器的名称
含义	• 表示特殊寄存器的内容
解释	• 更详细地讨论特殊寄存器的内容
设置方式 （设置时间）	• 如果是由系统执行设置时，表示继电器是由系统设定还是由用户设定； 〈设置方式〉 S　　：由系统设定 U　　：用户设定（顺控程序或者来自 GX Developer 的测试操作） S/U　：系统和用户都可以设定 〈当被设定时〉 只用于指示由系统设定寄存器 每次 END　　：在每个 END 处理过程中进行设定 初始化　　：只在初始化处理（当电源变为 ON，或者从 STOP 变为 RUN 时）过程中进行设定 状态改变　　：只在有状态变化时才进行设定 出错　　：在出错发生时进行设定 指令执行　　：当指令被执行时设定 请求　　：只在有用户请求（通过 SM，等软元件。）时才进行设定 系统切换　　：在执行系统切换时进行设定。
相应的 ACPU M9□□□	• 表示 ACPU 中相应的特殊继电器 （当内容改变时，特殊继电器代表 D9□□□格式改变。不兼容 Q00J/Q00/Q01 和 QnPRH。） • 新增表示特殊继电器是新近增加到 QnACPU 或者 Q 系列 CPU 模块中。
相应的 CPU	表示相应的 CPU 模块型号。 ○　　：表示所有的 QnACPU 和 QCPU。 QCPU　　：表示所有的 Q 系列 CPU 模块。 Q00J/Q00/Q01　　：表示基本型 QCPU。 Qn(H)　　：表示高性能型 QCPU。 QnPH　　：表示过程控制 CPU。 QnPRH　　：表示冗余 CPU。 QnU　　：表示通用型 QCPU。 QnA　　：表示 QnA 系列和 Q2ASCPU 系列。 Rem　　：表示 MELSECNET/H 远程 I/O 模块。 每个 CPU 模块的型号　　：表示相关的特殊 CPU 模块。（例如：Q4AR、Q3A）

对于下列条目的详细信息，请参考下面的手册：

• 网络　→ 相应网络模块的手册

• SFC　→ QCPU(Q 模式)/QnACPU 编程手册 (SFC)

要 点

SD1200 ～ 1255 用于 QnACPU。
在 QCPU 中将成为 “空闲”。

(1) 检测信息

附表 4.2 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD0	检测出错	检测出错代码	• 检测发现的出错的出错代码，存储为 BIN 数据。 • 内容和最新的故障历史信息相同。	S（出错）	D9008 格式改变	
SD1	用于检测出错发生的时间	用于检测出错发生的时间	• SD0 数据被更新时的年（后两位数）和月被以 2 位 BCD 代码存储。 b15 到 b8 b7 到 b0 年(0到99) | 月(1到12) (实例)十月，1995 H9510	S（出错）	新增	
SD2			• SD0 被更新时的日期和小时以 2 位 BCD 代码存储。 b15 到 b8 b7 到 b0 日期(1到31) | 小时(0到23) (实例)25日，上午10点 H2510			
SD3			• SD0 数据被更新时的分钟和秒以 2 位 BCD 代码存储。 b15 到 b8 b7 到 b0 分钟(0到59) | 秒(0到59) (实例)35分48秒 H3548			
SD4	出错信息类别	出错信息类别代码	用于表示正被存储到公共信息区域（SD5 到 SD15）和个别信息区域（SD16 到 SD26）中的是存储哪种类型的信息的类别代码被存储在这里。 类别代码用于判断被存储的出错信息类型。 b15 到 b8 b7 到 b0 单个信息类别代码 | 公共信息类别代码 • 公共信息类别代码存储下列代码： 0：没有出错 1：基板 / 模块号 / 可编程控制器号 / 基板号 * 2：文件名称 / 驱动器名称 3：时间（设定值） 4：程序出错位置 5：系统切换原因（只用于 Q4ARCPU 和冗余 CPU） 6：跟踪容量超限出错的原因（冗余 CPU 特有） 7：电源号（冗余 CPU 特有） 8：跟踪传送数据分类（冗余 CPU 特有） *：对于包含基本型 QCPU、高性能型 QCPU、过程控制 CPU 和通用型 QCPU 的多 CPU 系统，模块号或者 CPU 号是根据发生出错存储。（参考相应号码被存储的出错代码。） 1 号可编程控制器：1，2 号可编程控制器：2，3 号可编程控制器：3，4 号可编程控制器：4 • 个别信息类别代码存储下列代码： 0 ：没有出错 1 ：（空置） 2 ：文件名称 / 驱动器名称 3 ：时间（实际测量值） 4 ：程序出错位置 5 ：参数号 6 ：报警器号 7 ：CHK 指令故障号（这不能用于基本型 QCPU 和通用型 QCPU） 8 ：系统切换故障的原因（冗余 CPU 特有） 12：文件诊断信息（通用型 QCPU 特有） 13：参数号 /CPU 号（通用型 QCPU 特有）	S（出错）	新增	○ Rem

附表 4.2 特殊寄存器列表

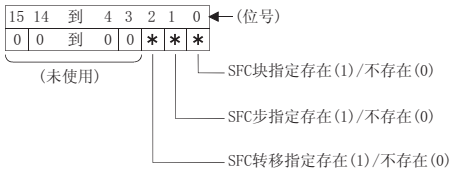
<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>出错公共信息</td><td>出错公共信息</td><td>• 对应出错代码（SD0）的公共信息存储在这里。
• 下列十种类型的信息被存储在这里：
• 出错公共信息 类型可以通过 SD4 中的 “公共信息类别代码”来判断。（存储在 SD4 中的 “公共信息 类别代码”的值对应下面的 1）到 8）、12）、13）。）
1) 槽号
<table><tr><th>号码</th><th>含义</th></tr><tr><td>SD5</td><td>槽号/PLC 号/基板 *1、*2、*3、*4</td></tr><tr><td>SD6</td><td>I/O号 *5</td></tr><tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>（空置）</td></tr></table>*1: 对于包含基本型 QCPU、高性能型 QCPU、过程控制 CPU 和通用型 QCPU 的多 CPU 系统，依据发生的出错存储槽号或者 CPU 号。
在多 CPU 系统中的插槽 0，是最右端的 CPU 模块右边的插槽。
（参考已存储的号码对应的出错代码。）
1 号 CPU：1，2 号 CPU：2，3 号 CPU：3，4 号 CPU：4
*2: 如果在 MELSECNET/H 远程 I/O 站中安装的模块上发生保险丝熔断或者 I/O 验证出错，网络号被存储到高 8 位，站号被存储到低 8 位。
使用 I/O 号去检查发生保险丝 熔断 或者 I/O 验证出错的模块。
*3: 当为可以安装模块的最后一个插槽以后的插槽中的模块执行指令时，255 被存储到基本型 QCPU 上的 SD5 中。
*4: 基板号和槽号的定义
〈基板号〉
此值用于确认安装了 CPU 模块的基板。 下列给出了基板号的的定义。
<table><tr><th>基板号</th><th>定义</th></tr><tr><td>0</td><td>表示安装了CPU模块的主基板。</td></tr><tr><td>1 到 7</td><td>表示扩展基板。通过扩展基板上的级号设置连接器进行的级号设置是基板号。
当级号设置是扩展1： 基板号 = 1
当级号设置是扩展7： 基板号 = 7</td></tr></table>冗余 CPU 的基板号一直是 “0”，因为它不能连接到扩展基板。
〈槽号〉
此值用于确认每个基板的插槽和安装在此插槽上的模块。
• 主基板的 I/O 槽 0（CPU 插槽右侧的插槽）被定义为 “槽号 = 0”的插槽。
• 按照主基板和扩展基板 1 到 7 的顺序，槽号码被连续分配到基板的槽号上。
• 当基板的插槽在可编程控制器 参数对话框的 I/O 分配设置中设定之后，槽号只分配到已设定的槽号上。
*5: 当 0FFFFH 被存储到 SD6（I/O 号）中时，由于可编程控制器参数对话框的 I/O 分配设置中有重叠的 I/O 号，所以 I/O 号不能被确认。 因此，使用 SD5 确认出错位置。
2) 文件名称 / 驱动器名称
<table><tr><th>号码</th><th colspan="2">含义</th></tr><tr><td>SD5</td><td colspan="2">驱动器</td></tr><tr><td>SD6
SD7
SD8
SD9</td><td colspan="2">文件名称
（ASCII代码：8字符</td></tr><tr><td>SD10</td><td>扩展 *6</td><td>2EH(.)</td></tr><tr><td>SD11</td><td colspan="2">（ASCII代码：3字符）</td></tr><tr><td>SD12
SD13
SD14
SD15</td><td colspan="2">（空置）</td></tr></table>（实例）文件名称
ABCDEFGH. IJK
b15 到 b8 b7 到 b0
<table><tr><td>42H(B)</td><td>41H(A)</td></tr><tr><td>44H(D)</td><td>43H(C)</td></tr><tr><td>46H(F)</td><td>45H(E)</td></tr><tr><td>48H(H)</td><td>47H(G)</td></tr><tr><td>49H(I)</td><td>2EH(.)</td></tr><tr><td>4BH(K)</td><td>4AH(J)</td></tr></table></td><td>S（出错）</td><td>新增</td><td>○
Rem</td></tr>
</table>

附表 4.2 特殊寄存器列表

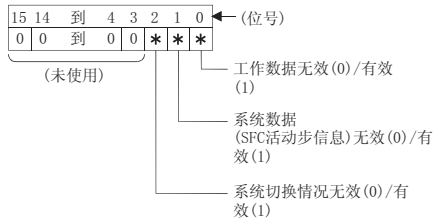
<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD5</td><td rowspan="11">出错公共信息</td><td rowspan="11">出错公共信息</td><td rowspan="6">3) 时间（设定值）
<table><tr><th>号码</th><th>含义</th></tr><tr><td>SD5</td><td>时间：1μs为单位(0到999μs)</td></tr><tr><td>SD6</td><td>时间：1ms为单位(0到65535ms)</td></tr><tr><td>SD7</td><td rowspan="9">(空置)</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table></td><td rowspan="6">S(出错)</td><td rowspan="6">新增</td><td rowspan="6">○
Rem</td></tr>
<tr><td>SD6</td></tr>
<tr><td>SD7</td></tr>
<tr><td>SD8</td></tr>
<tr><td>SD9</td></tr>
<tr><td>SD10</td></tr>
<tr><td>SD11</td><td rowspan="5">5) 系统切换的原因
<table><tr><th>号码</th><th>含义</th></tr><tr><td>SD5</td><td>系统切换情况
(0：自动切换/1：手动切换)</td></tr><tr><td>SD6</td><td>系统切换方向
(0：自动切换/1：手动切换
1：控制系统到待机系统)</td></tr><tr><td>SD7</td><td>跟踪标志 *8</td></tr><tr><td>SD8</td><td rowspan="9">(空置)</td></tr><tr><td>SD9</td></tr><tr><td>SD10</td></tr><tr><td>SD11</td></tr><tr><td>SD12</td></tr><tr><td>SD13</td></tr><tr><td>SD14</td></tr><tr><td>SD15</td></tr></table></td><td rowspan="5">S(出错)</td><td rowspan="5">新增</td><td rowspan="5">Q4AR</td></tr>
<tr><td>SD12</td></tr>
<tr><td>SD13</td></tr>
<tr><td>SD14</td></tr>
<tr><td>SD15</td></tr>
</table>

<table><tr><th>号码</th><th>含义</th></tr><tr><td>SD5</td><td rowspan="4">文件名称
(ASCII代码：8字符)</td></tr><tr><td>SD6</td></tr><tr><td>SD7</td></tr><tr><td>SD8</td></tr><tr><td>SD9</td><td>扩展 *6</td><td>2EH(.)</td></tr><tr><td>SD10</td><td colspan="2">(ASCII代码：3字符)</td></tr><tr><td>SD11</td><td colspan="2">模式 *7</td></tr><tr><td>SD12</td><td colspan="2">块号</td></tr><tr><td>SD13</td><td colspan="2">步号/转移号</td></tr><tr><td>SD14</td><td colspan="2">顺控步号(L)</td></tr><tr><td>SD15</td><td colspan="2">顺控步号(H)</td></tr></table>

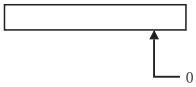
*7 ： 模式数据的内容



*8 ：跟踪标志内容
表示跟踪数据是否有效。



附表 4.2 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（当被设定时</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD5
SD6
SD7
SD8
SD9
SD10
SD11
SD12</td><td>出错公共信息</td><td>出错公共信息</td><td>5)　系统切换的原因
<table>
<tr><th>号码</th><th>含义</th></tr>
<tr><td>SD5</td><td>系统切换情况 *13</td></tr>
<tr><td>SD6</td><td>控制系统切换指令参数</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>（空置）</td></tr>
</table>
*13：系统切换原因的细节

0 ：无系统切换情况(默认)
1 ：电源OFF、复位、硬件故障、看门狗出错
2 ：停止出错(看门狗出错除外)
3 ：网络模块的系统切换请求
16 ：控制系统切换指令
17 ：来自GX Developer的控制系统切换请求

6)　跟踪容量超限出错的原因
当可以被跟踪的数据数（100k）被超过时，用相应特殊继电器的位模式来指示块号。
<table>
<tr><th></th><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr>
<tr><td>SD5</td><td>1
(SM1535)
(块16)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1
(SM1528)
(块9)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1
(SM1520)
(块1)</td></tr>
<tr><td>SD6</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD7</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD8</td><td>1
(SM1583)
(块64)</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1
(SM1568)
(块49)</td></tr>
<tr><td>SD9</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>SD15</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
</table></td><td>S（出错）</td><td>新增</td><td>QnPRH</td></tr>
<tr><td>SD13
SD14
SD15</td><td></td><td></td><td>7）电源号
<table>
<tr><th>编号</th><th>内容</th></tr>
<tr><td>SD5</td><td>基板号</td></tr>
<tr><td>SD6</td><td>电源号</td></tr>
<tr><td>SD7
SD8
SD9
SD10
SD11
SD12
SD13
SD14
SD15</td><td>（空置）</td></tr>
</table>
1：电源1异常
2：电源2异常
“电源模块1”：安装在冗余基板(Q38RB、68RB)的POWER1插槽上的冗余电源模块
“电源模块2”：安装在冗余基板(Q38RB、68RB)的POWER2插槽上的冗余电源模块</td><td>S（出错）</td><td>新增</td><td>Qn(H)*1
QnPH*1
QnPRH</td></tr>
</table>

*1：以序列号的高 5 位为 “07032”以后的 CPU 为对象。

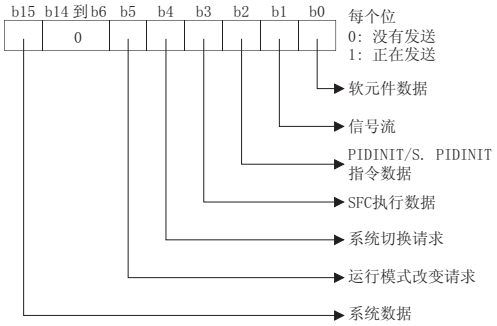
附表 4.2 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD5						
SD6						
SD7						
SD8						
SD9						
SD10	出错公共信息	出错公共信息		S（出错）	新增	QnPRH
SD11						
SD12						
SD13						
SD14						
SD15						

8) 跟踪传送数据块
在跟踪过程中存储数据块。

号码	含义
SD5	数据类型 *15
SD6	(空置)
SD7	
SD8	
SD9	
SD10	
SD11	
SD12	
SD13	
SD14	
SD15	

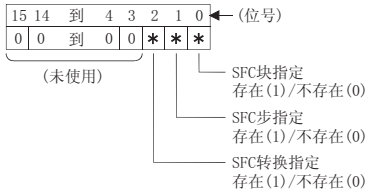
*15：数据分类的详细信息



附表 4.2 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD16
SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>出错公共信息</td><td>出错公共信息</td><td>
• 对应出错代码（SD0）的单个信息存储在这里。
• 存储了下列八种不同类型的信息。
• 出错个别信息类型可以通过 SD4 中的 “个别信息类别代码” 来判断。
（存储在 SD4 中的 “个别信息类别代码” 的值对应下列的 1）到 8）。）

1) （空置）

2) 文件名 / 驱动器名
<table>
<tr><th>号码</th><th colspan="2">含义</th></tr>
<tr><td>SD16</td><td colspan="2">驱动器</td></tr>
<tr><td>SD17</td><td colspan="2" rowspan="4">文件名
（ASCII码：8个字符）</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td></tr>
<tr><td>SD21</td><td>扩展 *6</td><td>2EH(.)</td></tr>
<tr><td>SD22</td><td colspan="2">（ASCII码：3个字符）</td></tr>
<tr><td>SD23</td><td colspan="2" rowspan="4">（空置）</td></tr>
<tr><td>SD24</td></tr>
<tr><td>SD25</td></tr>
<tr><td>SD26</td></tr>
</table>
(实例)文件名 = ABCDEFGH. IJK
<table>
<tr><th>b15 到 b8</th><th>b7 到 b0</th></tr>
<tr><td>42H(B)</td><td>41H(A)</td></tr>
<tr><td>44H(D)</td><td>43H(C)</td></tr>
<tr><td>46H(F)</td><td>45H(E)</td></tr>
<tr><td>48H(H)</td><td>47H(G)</td></tr>
<tr><td>49H(I)</td><td>2EH(.)</td></tr>
<tr><td>4BH(K)</td><td>4AH(J)</td></tr>
</table>
3) 时间（实际测量值）
<table>
<tr><th>号码</th><th>含义</th></tr>
<tr><td>SD16</td><td>时间：1 μs为单位(0到999 μs)</td></tr>
<tr><td>SD17</td><td>时间：1ms为单位(0到65535ms)</td></tr>
<tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>（空置）</td></tr>
</table>
4) 程序出错位置
<table>
<tr><th>号码</th><th colspan="2">含义</th></tr>
<tr><td>SD16</td><td colspan="2" rowspan="4">文件名
（ASCII码：8个字符）</td></tr>
<tr><td>SD17</td></tr>
<tr><td>SD18</td></tr>
<tr><td>SD19</td></tr>
<tr><td>SD20</td><td>扩展 *6</td><td>2EH(.)</td></tr>
<tr><td>SD21</td><td colspan="2">（ASCII码：3个字符）</td></tr>
<tr><td>SD22</td><td colspan="2">模式 *7</td></tr>
<tr><td>SD23</td><td colspan="2">块号</td></tr>
<tr><td>SD24</td><td colspan="2">步号/转换号</td></tr>
<tr><td>SD25</td><td colspan="2">顺控步号(L)</td></tr>
<tr><td>SD26</td><td colspan="2">顺控步号(H)</td></tr>
</table>
*7 : 模式数据的内容


5) 参数数目 6) 报警器数/ 7) CHK指令故障号
<table>
<tr><th>号码</th><th>含义</th></tr>
<tr><td>SD16</td><td>参数数目 *16</td></tr>
<tr><td>SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>（空置）</td></tr>
</table>
<table>
<tr><th>号码</th><th>含义</th></tr>
<tr><td>SD16</td><td>数目</td></tr>
<tr><td>SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>（空置）</td></tr>
</table>
*16：对于参数数目的详细信息，参考 QCPU 用户手册（功能解说 / 程序基础篇）或者使用的 QnACPU 的用户手册。
</td><td>S（出错）</td><td>新增</td><td>○
Rem</td></tr>
</table>

备 注

*6 ： 扩展名如下表所示。

附表 4.3 扩展名

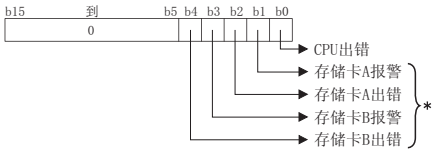
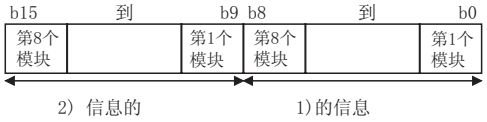
SDn	SDn+1		扩展名称	文件类型
高 8 位	低 8 位	高 8 位		
51H	50H	41H	QPA	参数
51H	50H	47H	QPG	• 顺控程序 • SFC 程序
51H	43H	44H	QCD	软元件注释
51H	44H	49H	QDI	软元件初始化值
51H	44H	52H	QDR	文件寄存器
51H	44H	53H	QDS	仿真数据 （用于 QnA）
51H	44H	4CH	QDL	局部软元件 （基本型 QCPU 以外的其他 CPU）
51H	54H	44H	QTD	采样跟踪数据 （基本型 QCPU 以外的其他 CPU）
51H	54H	4CH	QTL	状态锁存数据（用于 QnA）
51H	54H	50H	QTP	程序跟踪数据（用于 QnA）
51H	54H	52H	QTR	SFC 跟踪文件（用于 QnA）
51H	46H	44H	QFD	故障历史数据 （基本型 QCPU 和通用型 QCPU 以外的其他 CPU）
51H	46H	44H	QST	SP.DEVST/S.DEVLD 指令文件 （用于通用型 QCPU）

附 附录

附表 4.2 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td rowspan="2">SD26</td><td rowspan="2">出错公共信息</td><td rowspan="2">出错公共信息</td><td>8) 系统切换故障的原因
<table><tr><td>号码</td><td>含义</td></tr><tr><td>SD16</td><td>系统切换禁止情况 *14</td></tr><tr><td>SD17
SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>（空置）</td></tr></table>*14：系统切换故障详细的原因
0 ：正常切换结束(默认)
1 ：热备电缆故障(电缆移动，电缆故障，内部电路故障，硬件故障)
2 ：在待机系统中发生硬件故障，电源 OFF，复位或者看门狗出错
3 ：在控制系统中发生硬件故障，电源 OFF，复位或者看门狗出错
4 ：准备跟踪
5 ：时间超限
6 ：待机系统处于停止出错(看门狗出错除外)
7 ：两个系统的运行状态不同(只在备用模式中)
8 ：在过程中内存从控制系统到待机系统
9 ：程序在线更改
10 ：待机系统网络模块检测到出错
11 ：系统切换正在执行
12 ：运行中模块更换正在进行</td><td rowspan="2">S（出错）</td><td rowspan="2">新增</td><td>QnPRH</td></tr>
<tr><td>12) 文件诊断信息
<table><tr><td>SD16</td><td>故障信息(H)</td><td>驱动器号(L)</td></tr><tr><td>SD17
SD18
SD19
SD20</td><td colspan="2">文件名
(ASCII码：8个字符)</td></tr><tr><td>SD21</td><td>扩展 *6</td><td>2EH(.)</td></tr><tr><td>SD22</td><td colspan="2">(ASCII码：3个字符)</td></tr><tr><td>SD23
SD24</td><td colspan="2">故障信息2
(读取CRC值)</td></tr><tr><td>SD25
SD26</td><td colspan="2">故障信息3
(计算CRC值)</td></tr></table>13) 参数号 /CPU 号
<table><tr><td>号码</td><td>含义</td></tr><tr><td>SD16</td><td>参数号 *16</td></tr><tr><td>SD17</td><td>CPU号(1到4)</td></tr><tr><td>SD18
SD19
SD20
SD21
SD22
SD23
SD24
SD25
SD26</td><td>（空置）</td></tr></table></td><td>QnU</td></tr>
</table>

附表 4.2 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD50	出错复位	执行出错复位的出错号	• 存储执行出错复位的出错号	U	新增	○ Rem
SD51	电池电量不足锁存	位模式指示哪里发生电池电压下降	• 当电池电压下降时，所有的对应位变为 1(ON)。 • 随后，即使在电池电压恢复到正常之后，它们保持为 1(ON)。  *：这不能用于基本型QCPU。 • 在报警中，在为电池电量低指定的时间以内，数据可以被保持。 • 此出错表示电池放电的结束。 • 当使用高性能型 QCPU、过程控制 CPU、冗余 CPU 或者通用型 QCPU 时，存储卡 B 是标准的，因此相应位一直保持为 OFF。	S（出错）	新增	○
SD52	电池电量不足	位模式指示哪里发生电池电压下降	• 和上面 SD51 的配置相同 • 检到报警 (ON) 后，通过错误检测 (ON) 使报警 OFF。（仅用于通用型 QCPU） • 当电池电压恢复为正常，变为 0(OFF)。 • 当使用高性能型 QCPU、过程控制 CPU、冗余 CPU 或者通用型 QCPU 时，存储卡 B 是标准的，因此相应位一直保持为 OFF。	S（出错）	新增	
SD53	AC/DC DOWN 检测	AC/DC DOWN 检测次数	• 在 CPU 模块的运行过程中，每当输入电压下降到或者低于额定值的 85%(AC 电源)/65%(DC 电源）时，此值增加 1 并以 BIN 码存储。	S（出错）	D9005	○ Rem
SD54	MINI 链接出错	出错检测状态	1) 当安装的 MINI(-S3) 的 X(n+0)/X(n+20)，X(n+6)/X(n+26)，X(n+7)/X(n+27) 和 X(n+8)/X(n+28) 中的任何一个变为 ON 时，相应站的位变为 1(ON)。 2) 当安装的 MINI(-S3) 和 CPU 模块之间不能通讯时，变为 1(ON)。 	S（出错）	D9004 格式改变	QnA
SD60	保险丝熔断的模块号	保险丝熔断的模块号	• 存储在这里的值是最小的有保险丝熔断的模块的站 I/O 号。	S（出错）	D9000	○ Rem
SD61	I/O 模块验证出错号	I/O 模块验证出错号	• 发生 I/O 模块校验号出错的模块的最小 I/O 号。	S（出错）	D9002	

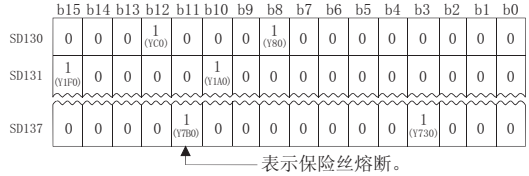
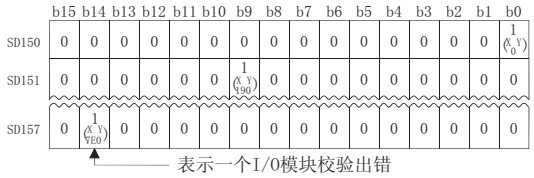
附表 4.2 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD62</td><td>报警器号</td><td>报警器号</td><td>• 检测到的第一个报警器 号（F 号）被存储在这里。</td><td>S（指令执行）</td><td>D9009</td><td rowspan="18">○</td></tr>
<tr><td>SD63</td><td>报警器数</td><td>报警器数</td><td>• 存储查找到的报警器数。</td><td>S（指令执行）</td><td>D9124</td></tr>
<tr><td>SD64</td><td rowspan="16">检测到的报警器号码表</td><td rowspan="16">报警器检测号</td><td rowspan="16">当 F 由于 [OUT F] 或 [SET F]，而变为 ON 时，依次变为 ON 的 F 号被注册到 SD64 到 SD79 中。
由 [RST F] 变为 OFF 的 F 号，被从 SD64-SD79 中删除，存储在删除的 F 号之后的 F 号被移位到前面的寄存器中。
执行 [LEDR] 指令 ，将 SD64 到 SD79 的 内容向前移动一位。
（这也可以通过使用 Q3A/Q4ACPU 的 INDICATOR RESET 开关来进行。）
在检测到 16 个报警器之后，检测到的第 17 个不会被存储到 SD64 到 SD79 中。
SET F50 SET F25 SET F99 RST F25 SET F15 SET F70 SET F65 SET F38 SET F110 SET F151 SET F210 LEDR
SD62 0 50 50 50 50 50 50 50 50 50 50 50 99 (检测到的号码)
SD63 0 1 2 3 2 3 4 5 6 7 8 9 8 (检测到的报警器数)
SD64 0 50 50 50 50 50 50 50 50 50 50 50 99
SD65 0 0 25 25 99 99 99 99 99 99 99 99 15
SD66 0 0 0 99 0 15 15 15 15 15 15 15 70
SD67 0 0 0 0 0 0 70 70 70 70 70 70 65
SD68 0 0 0 0 0 0 0 65 65 65 65 65 38
SD69 0 0 0 0 0 0 0 0 38 38 38 38 110
SD70 0 0 0 0 0 0 0 0 0 110 110 110 151
SD71 0 0 0 0 0 0 0 0 0 0 151 151 210
SD72 0 0 0 0 0 0 0 0 0 0 0 210 0
SD73 0 0 0 0 0 0 0 0 0 0 0 0 0
SD74 0 0 0 0 0 0 0 0 0 0 0 0 0
SD75 0 0 0 0 0 0 0 0 0 0 0 0 0
SD76 0 0 0 0 0 0 0 0 0 0 0 0 0
SD77 0 0 0 0 0 0 0 0 0 0 0 0 0
SD78 0 0 0 0 0 0 0 0 0 0 0 0 0
SD79 0 0 0 0 0 0 0 0 0 0 0 0 0
(检测到的号码)</td><td rowspan="16">S（指令执行）</td><td>D9125</td></tr>
<tr><td>SD65</td><td>D9126</td></tr>
<tr><td>SD66</td><td>D9127</td></tr>
<tr><td>SD67</td><td>D9128</td></tr>
<tr><td>SD68</td><td>D9129</td></tr>
<tr><td>SD69</td><td>D9130</td></tr>
<tr><td>SD70</td><td>D9131</td></tr>
<tr><td>SD71</td><td>D9132</td></tr>
<tr><td>SD72</td><td>新增</td></tr>
<tr><td>SD73</td><td>新增</td></tr>
<tr><td>SD74</td><td>新增</td></tr>
<tr><td>SD75</td><td>新增</td></tr>
<tr><td>SD76</td><td>新增</td></tr>
<tr><td>SD77</td><td>新增</td></tr>
<tr><td>SD78</td><td>新增</td></tr>
<tr><td>SD79</td><td>新增</td></tr>
<tr><td>SD80</td><td>CHK 号</td><td>CHK 号</td><td>• 由 CHK 指令检测到的出错代码以 BCD 代码存储。</td><td>S（指令执行）</td><td>新增</td><td>QnA
Qn(H)
QnPH
QnPRH</td></tr>
</table>

附表 4.2 特殊寄存器列表

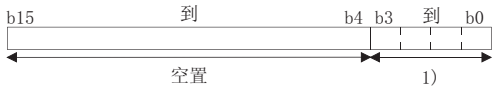
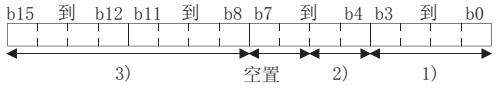
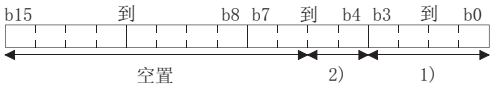
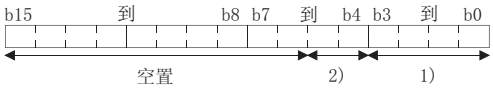
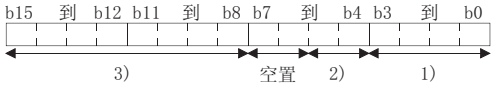
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD90	步转移看门狗定时器设定值（只有当 SFC(只有当 SFC	用于定时器设定值和时间溢出出错的 F 号	对应 SM90 对应 SM91 对应 SM92 对应 SM93 对应 SM94 对应 SM95 对应 SM96 对应 SM97 对应 SM98 对应 SM99	• 设定当步转移看门狗定时器设置或者看门狗定时器时限出错发生时，将变为 ON 的报警器号 (F 号)。 b15 到 b8 b7 到 b0 F号设置 (0 到 255)　定时器时间限设置 (1 到 255s: (1s 为单位)) • 在活动步中将 SM90 到 SM99 中的任何一个变为 ON 以启动定时器，并且如果在定时器时间限以内没有满足到下一个相应步的转移条件，则设定的报警器 (F) 变为 ON。	U	D9108 D9109 D9110 D9111 D9112 D9113 D9114 新增 新增 新增	QnA Qn(H) QnPH QnPRH
SD100	传送速度存储区域通讯	存储在串行通讯设置中指定的传送速度。	96 : 9.6kbps, 192 : 19.2kbps, 384 : 38.4kbps, 576 : 57.6kbps, 1152 : 115.2kbps		S(电源接通或者复位)	新增	
SD101	设置存储区域	存储在串行通讯设置中指定的通讯设置。	b15 到 b6 b5 b4 b3 到 b0 * 运行中程序修改设置 0: 否 1: 允许 总数检查是/否 0: 是 1: 否 * *：由于数据由系统使用，所以未定义。		S(电源接通或者复位)	新增	Q00/Q01
SD102	传送时间等待存储区域	存储在串行通讯设置中指定的传送等待时间。	0 : 无等待时间 1 到 FH : 等待时间（单位：10ms) 默认为 0。		S(电源接通或者复位)	新增	
SD105	CH1 传送速度设置 (RS-232)	存储当采用 GX Developer 时预设的传送速度。	96 : 9600bps, 192 : 19.2kbps, 384 : 38.4kbps, 576 : 57.6kbps, 1152 : 115.2kbps *: RS-232 以外的连接，保持 RS-232 连接时数据不变。 （未连接时，默认值为 1152。）	S	新增	Qn(H) QnPH QnPRH QnU Rem
SD110	数据发送结果存储区域	当使用串行通讯功能时，存储数据发送结果。	数据发送时的出错代码被存储。	S(出错)	新增	Q00/Q01
SD111	数据接收结果存储区域	当使用串行通讯功能时，存储数据接收结果。	存储数据接收时的出错代码。	S(出错)		
SD119	电池寿命延长原因	电池寿命延长原因	存储使电池寿命延长功能有效的原因。 SD119 不为 0 时，电池寿命延长功能有效。 b15 到 b2 b1 b0 0: 无原因 1: 原因 b0: CPU开关设置 b1: 正在通过锁存数据备份功能进行到标准ROM的备份	S(状态改变)	新增	QnU

附表 4.2 特殊寄存器列表

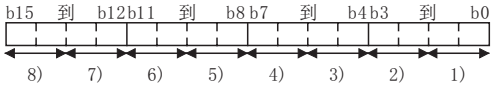
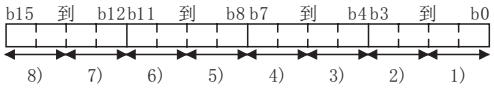
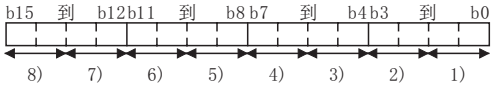
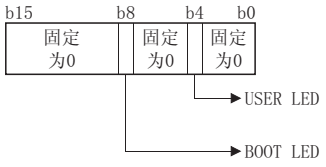
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD130 SD131 SD132 SD133 SD134 SD135 SD136 SD137	保险丝熔断的模块	以 16 点为单位的位模式表示保险丝熔断的模块 0：无保险丝熔断 1：有保险丝熔断存在	• 保险丝熔断的输出模块的号码以位模式（以 16 点为单位）输入。 （如果模块号是由参数设定的，则参数设定的号码被存储。）  • 不清除，即使将熔断的保险丝更换为一个新保险丝。 此标志由出错复位操作清除	S（出错）	新增	Q00J/Q00/Q01
SD150 SD151 SD152 SD153 SD154 SD155 SD156 SD157	I/O 模块验证出错	位模式，以 16 点为单位，表示验证出错的模块。 0：无 I/O 验证出错 1：I/O 验证出错存在	• 当 I/O 模块，其数据和电源接通时输入的数据不同，被检测到时，I/O 模块号（以 16 点为单位）被输入到位模式中。（当执行参数设置时，在参数中预设 I/O 模块号设置。）  • 不清除，即使将熔断的保险丝更换为新的保险丝。 此标志由出错复位操作清除。			

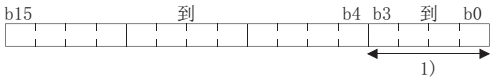
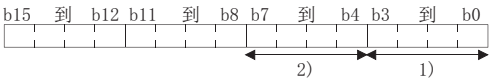
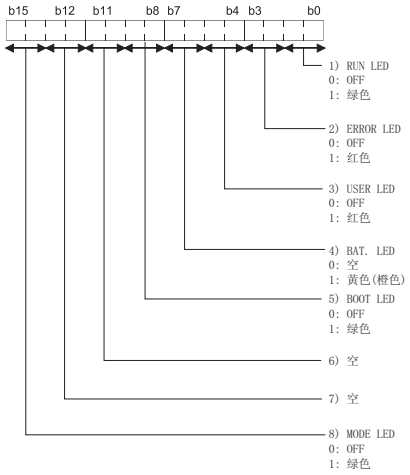
(2) 系统信息

附表 4.4 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD200	开关的状态	CPU 开关的状态	• 远程 I/O 模块的开关状态被以下列格式存储。  1) 远程 I/O 模块开关状态常 1：停止	S(经常)	新增	Rem
			• CPU 的开关状态被以下列格式存储。  1): CPU开关状态 0: RUN 1: 停止 2: L.CLR 2): 存储卡开关状态 常OFF 3): DIP开关 b8到b12对应系统设置开关1的SW1到SW5。0：OFF，1：ON。b13到b15空置。	S（每个 END 处理）	新增	Qn(H) QnPH QnPRH
			• CPU 的开关状态被以下列格式存储。  1): CPU开关状态 0: RUN 1: 停止 2): 内存卡开关常OFF 常OFF	S（每个 END 处理）	新增	Q00J/Q00/Q01
			• CPU 的开关状态被以下列格式存储。  1): CPU开关状态 0: RUN 1: 停止 2): 内存卡开关常OFF 常OFF	S (RUN/STOP/RESET 开关改变时)	新增	QnU
			• CPU 的开关状态被以下列格式存储。  1): CPU开关状态 0: RUN 1: 停止 2: L.CLR 2): 存储卡开关 b4对应到存储卡A，b5对应存储卡B 0：OFF，1：ON 3): DIP开关 b8到b12对应系统设置开关1的SW1到SW5。b14和b15分别对应系统设置开关2的SW1和SW2。0：OFF，1：ON	S（每个 END 处理）	新增	QnA

附表 4.4 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD201	LED 状态	CPU-LED 的状态	• 下列位模式用于存储 CPU 模块上 LED 的状态： • 0 是熄灭，1 是点亮，2 是闪烁。  1): RUN 5): BOOT 2): ERR. 6): 空 空模式位模式 3): USER 7): 空 0: OFF 1: 绿色 4): BAT. 8): MODE 2: 橙色 （基本型 QCPU 不包括 3）到 8）。）	S（状态改变）	新增	QCPU
			• 下列位模式用于存储 CPU 模块上 LED 的状态： • 0 是熄灭，1 是点亮，2 是闪烁。  1): RUN 5): BOOT 2): ERROR 6): 空 3): USER 7): 空 4): BAT. 8): MODE	S（状态改变）	新增	QnU
			• 下列位模式用于存储 CPU 模块上 LED 的状态： • 0 是熄灭，1 是点亮，2 是闪烁  1): RUN 5)BOOT 2): ERROR 6): CARD A（存储卡 A） 3): USER 7): CARD B（存储卡 B） 4): BAT.ALARM 8): 空	S（状态改变）	新增	QnA
SD202	LED 熄灭命令	熄灭的 LED 的位模式	• 存储熄灭的 LED 的位模式 （只允许 USER 和 BOOT） • 1：熄灭，0：不熄灭	U	新增	Qn(H) QnPH QnPRH QnU
			• 使用此寄存器指定要被熄灭的 LED，将 SM202 从 OFF 变为 ON 以熄灭指定的 LED。 可以指定 USER 和 BOOT 为将要熄灭的 LED。 • 以下列位模式指定将要熄灭的 LED。 （1：熄灭，0：不熄灭）  	U	新增	

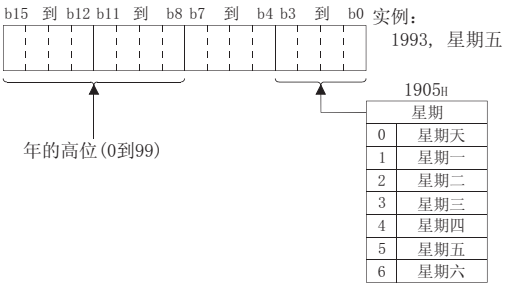
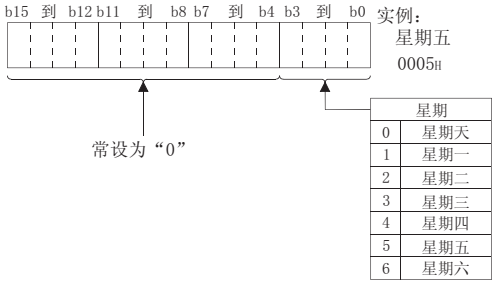
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
			• 远程 I/O 模块的运行状态被以下列格式存储。  1) 远程 I/O 模块 运行状态 2：停止	S（经常）	新增	Rem
SD203	CPU 的运行状态	CPU 的运行状态	• CPU 运行状态按照下图所示进行存储：  1)：CPU的运行状态 0：RUN 1：STEP-RUN(仅用于QnACPU) 2：STOP 3：PAUSE 2)：STOP/PAUSE的原因 0：在远程操作程序中来自RUN/STOP开关的指令(对于基本型QCPU，是“RUN/STOP/RESET开关”) 1：远程触点 2：GX Developer/串行通讯等 3：内部程序指令 4：出错 注释：优先级是先来先执行	S（每个 END 处理）	D9015 格式改变	○
SD204	LED 显示颜色	CPU-LED 显示颜色	• SD201 的 1) 到 8) 表示 LED 状态时，LED 显示颜色。  	S（状态改变）	新增	QnU

附表 4.4 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD206	软元件测试执行类型	0：测试还没有执行 1：在 X 软元件测试过程中 2：在 Y 软元件测试过程中 3：在 X/Y 软元件测试过程中	• 当软元件测试模式是在 GX Developer 上执行时，进行设置。	S（请求）	新增	Rem
SD207	LED 显示优先级	优先级 1 到 4	• 当产生出错时，LED 显示（闪烁）对应出错号设置优先级。（基本型 QCPU 只支持报警器（出错条目号 7）。 • 在通用型 QCPU 中，设置发生异常时的各优先顺序相应出错 LED 的执行 / 不执行。 • 用于优先级的设置区域如下所示： b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 SD207 优先级4 优先级3 优先级2 优先级1 SD208 优先级8 优先级7 优先级6 优先级5 SD209 优先级11 优先级10 优先级9 （当使用冗余CPU时，优先级11有效。） 默认值 SD207 = 4321H(0000H 对于基本型 QCPU) SD208 = 8765H(0700H 对于基本型 QCPU) (0765H 对于冗余 CPU) SD209 = 00A9H(0000H 对于基本型 QCPU) (0B09H 对于冗余 CPU) • 如果 “0” 被设定，则无显示。 • 在基本型 QCPU 中，如果 “7” 被设定到优先级 1 到 11 中的某一个，则当报警器变为 ON 时 ERR. LED 变为 ON。 • 在基本型 QCPU 中，如果 “7” 没有被设定到优先级 1 到 11 中的某一个，则当报警器变为 ON 时 ERR. LED 不变为 ON。 但是，即使设定了 “0”，信息有关的 CPU 模块操作停止（包括参数设置）出错将会被 LED 无条件地显示。	U	D9038	QnA Q00J/Q00/Q01*9 Qn(H) QnPH QnPRH QnU
SD208		优先级 5 到 8			D9039 格式改变	
SD209		优先级 9 到 11			新增	
SD210	时钟数据	时钟数据（年，月）	• 年（后两位数）和月按如下所示，以 BCD 码存储： b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 7月，1993 9307H 年 月	S（请求）/U	D9025	○ Rem
SD211	时钟数据	时钟数据（日期，小时）	• 日期和小时按如下所示，以 BCD 码存储： b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 31日，上午10 3110H 天 小时		D9026	
SD212	时钟数据	时钟数据（分钟，秒）	• 分钟和秒（在小时之后）按如下所示，以 BCD 码存储： b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 35分钟，48秒。 3548H 分钟 秒		D9027	

*9：适用于功能版本为 B 或者更高的 CPU。

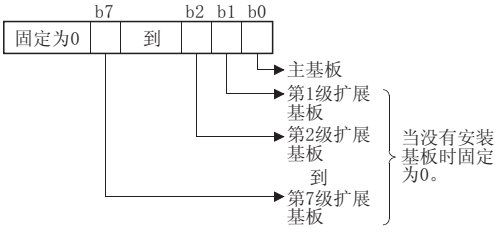
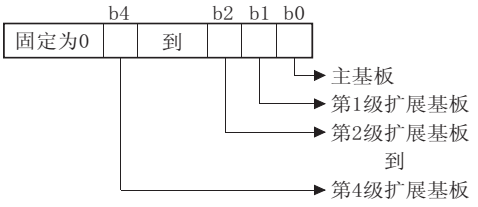
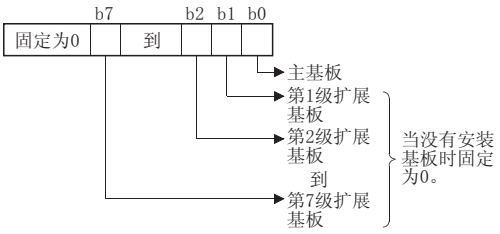
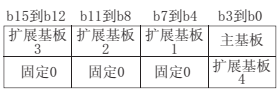
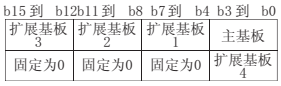
附表 4.4 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td rowspan="2">SD213</td><td rowspan="2">时钟数据</td><td>时钟数据（年的前位，星期）</td><td>• 按如下所示，以 BCD 码存储年（前两位）和星期。
 
</td><td rowspan="2">S（请求）/U</td><td rowspan="2">D9028</td><td>QCPU
Rem</td></tr>
<tr><td>时钟数据（星期）</td><td>• 按如下所示，星期是以 BCD 码存储：
 
</td><td>QnA</td></tr>
<tr><td>SD220</td><td rowspan="8">LED 显示数据</td><td rowspan="8">LED 显示数据</td><td rowspan="8">• LED 显示存储在这里的 ASCII 数据（16 字符）。
（在基本型 QCPU 中，寄存器存储出错发生（包括报警器 ON）时的消息（16 字符的 ASCII 数据）。

b15 到 b8 b7 到 b0
SD220 右起第15个字符 右起第16个字符
SD221 右起第13个字符 右起第14个字符
SD222 右起第11个字符 右起第12个字符
SD223 右起第9个字符 右起第10个字符
SD224 右起第7个字符 右起第8个字符
SD225 右起第5个字符 右起第6个字符
SD226 右起第3个字符 右起第4个字符
SD227 右起第1个字符 右起第2个字符

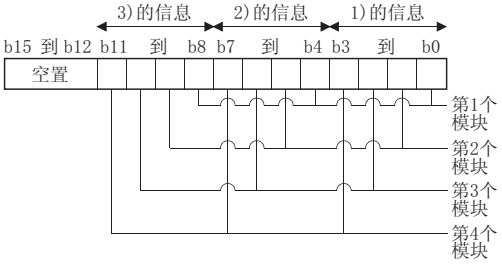
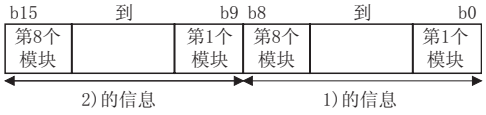
• 在基本型 QCPU 和通用型 QCPU 中，CHK 指令执行时的显示软元件数据不被存储。</td><td rowspan="8">S（当改变时）</td><td rowspan="8">新增</td><td rowspan="8">○</td></tr>
<tr><td>SD221</td></tr>
<tr><td>SD222</td></tr>
<tr><td>SD223</td></tr>
<tr><td>SD224</td></tr>
<tr><td>SD225</td></tr>
<tr><td>SD226</td></tr>
<tr><td>SD227</td></tr>
<tr><td>SD235</td><td>直到运行中模块更换正在执行的基板</td><td>运行中模块更换正在执行的基板的起始 I/O 号 /10H</td><td>• 10H 被加到运行中模块更换正在执行的模块的起始 I/O 号的值上。</td><td>S（在运行中模块更换过程中）</td><td>新增</td><td>QnPH
QnPRH</td></tr>
<tr><td>SD240</td><td>基板模式</td><td>0：自动模式
1：详细模式</td><td>• 存储基板模式。</td><td>S（初始化）</td><td>新增</td><td rowspan="2">QCPU
Rem</td></tr>
<tr><td>SD241</td><td>扩展基板数</td><td>0：只有主基板
1 到 7：扩展基板数</td><td>• 存储安装的最大扩展基板数。</td><td>S（初始化）</td><td>新增</td></tr>
</table>

附表 4.4 特殊寄存器列表

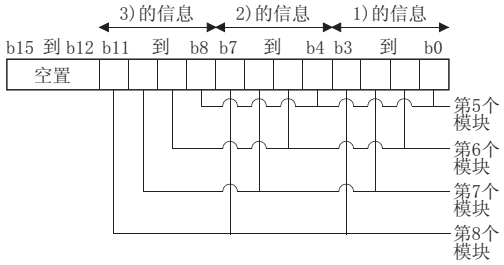
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD242	A/Q 基板差别	基板类型差别 0：安装了 QA**B（A 模式） 1：安装了 Q**B（Q 模式）		S（初始化）	新增	Qn(H) QnPH QnPRH Rem
	安装的 Q 基板存在 / 不存在	基板类型差别 0：没有安装基板 1：Q**B 安装了		S（初始化）	新增	Q00J/Q00/Q01 Rem
	安装的 Q 基板存在 / 不存在	基板类型差别 0：没有安装基板 1：Q**B 安装了		S（初始化）	新增	QnU
SD243 SD244	基板插槽数	基板插槽数	 • 如上所示，每个区域存储安装的插槽数。	S（初始化）	新增	Qn(H) QnPH QnPRH QnU Rem
SD243 SD244	基板插槽数（操作状态）	基板插槽数	 • 如上所示，每个区域存储安装的插槽数。 （当进行参数设置时设定插槽数）	S（初始化）	新增	Q00J/Q00/Q01 Rem
SD245 SD246	基板插槽数（安装状态）	基板插槽数	 • 如上所示，每个区域存储基板上安装的模块的插槽数（安装的基板的实际插槽数）。	S（初始化）	新增	Q00J/Q00/Q01*9
SD250	安装的最大 I/O	安装的最大 I/O 号	• 当 SM250 从 OFF 变为 ON 时，以 BIN 值存储安装的模块的最后 I/O 号的高 2 位数加 1 后的值。	S（请求 END）	新增	Qn(H) QnPH QnPRH Rem
			• 以 BIN 值存储安装的模块的最后 I/O 号的高 2 位数加 1 后的值。	S（初始化）	新增	Q00J/Q00/Q01 QnU Rem
SD251	用于更换的起始 I/O 号	用于模块更换的起始 I/O 号	• 存储在运行状态（电源接通）中被卸载 / 更换的 I/O 模块的起始 I/O 号的高两位数。（默认值：100H）。	U	D9094	Q2A(S1) Q3A Q4A Q4AR
SD253	RS422 传送速度	RS422 传送速度	• 存储 RS422 的传送速度。 0 ： 9600bps 1 ： 19.2kbps 2 ： 38.4kbps	S（当改变时）	新增	QnA

*9： 适用于功能版本为 B 或者更高的 CPU。

附表 4.4 特殊寄存器列表

号码	名称	含义		解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD254	MELSECNET/10. MELSECNET/H 信息	安装的模块数		• 表示安装的 MELSECNET/10 模块或者 MELSECNET/H 模块的数目。	S（初始化）	新增	○
SD255		来自第 1 个模块的信息	• I/O 号	• 表示安装的 MELSECNET/10 模块或者 MELSECNET/H 模块的 I/O 号			
SD256			• 网络号	• 表示安装的 MELSECNET/10 模块或者 MELSECNET/H 模块的网络号			
SD257			• 组号站号	• 表示安装的 MELSECNET/10 模块或者 MELSECNET/H 模块的组号			
SD258			• 站号	• 表示安装的 MELSECNET/10 模块或者 MELSECNET/H 模块的站号			
SD259			• 待机信息	• 在待机站中，待机站的模块数被存储。(1 到 4)			QnA Qn(H) QnPH QnPRH QnU
SD260 到 SD264		来自第 2 个模块的信息		• 配置和第一个模块的相同。			
SD265 到 SD269		来自第 3 个模块的信息		• 配置和第一个模块的相同			QnA Qn(H) QnPH QnPRH QnU*10
SD270 到 SD274		来自第 4 个模块的信息		• 配置和第一个模块的相同。			
SD280	CC-Link 出错	出错检测状态		1) 当安装的 CC-Link 模块的 Xn0 变为 ON 时，相应站的位信息变为 1(ON)。 2) 当安装的 CC-Link 模块的 Xn1 或者 XnF 变为 OFF，相应站的位信息变为 1(ON)。 3) 变为 1，(ON) 当安装的 CC-Link 模块和 CPU 模块之间不能通讯时。  上面的第 n 号模块是安装起始 I/O 号码排序的。 （但是，没有进行参数设置的 I/O 号没有考虑在内。）	S（出错）	新增	Qn(H) QnPH QnPRH Rem
				• 当安装的 CC-Link 模块的 Xn0 变为 ON 是，相应站的位信息变为 1(ON)。 • 当安装的 CC-Link 模块的 Xn1 或者 XnF 变为 OFF 时，相应站的位信息变为 1(ON)。 • 变为 1(ON)，当安装的 CC-Link 模块和 CPU 模块不能通讯时。 	S（出错）	新增	QnA

*10：除 Q02UCPU 外的通用型 QCPU。

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD281	CC-Link 出错	出错检测状态	1) 当安装的 CC-Link 模块的 Xn0 变为 ON 时，相应站的位信息变为 1(ON)。 2) 当安装的 CC-Link 模块的 Xn1 或者 XnF 变为 OFF，相应站的位信息变为 1(ON)。 3) 当安装的 CC-Link 模块和 CPU 模块之间不能通讯时变为 1(ON)。  上面的第 n 号模块是按照起始 I/O 号码排序的。 （但是，没有进行参数设置的 I/O 号没有考虑在内。）	S（出错）	新增	Qn(H)*14 QnPH*14
SD290	软元件分配（和参数内容相同）	分配给 X 的点数	• 存储当前为 X 软元件分配的点数	S（初始化）	新增	○ Rem
SD291		分配给 Y 的点数	• 存储当前为 Y 软元件分配的点数			
SD292		分配给 M 的点数	• 存储当前为 M 软元件分配的点数			
SD293		分配给 L 的点数	• 存储当前为 L 软元件分配的点数			○
SD294		分配给 B 的点数	• 存储当前为 B 软元件分配的点数			○ Rem
SD295		分配给 F 的点数	• 存储当前为 F 软元件分配的点数			○
SD296		分配给 SB 的点数	• 存储当前为 SB 软元件分配的点数			○ Rem

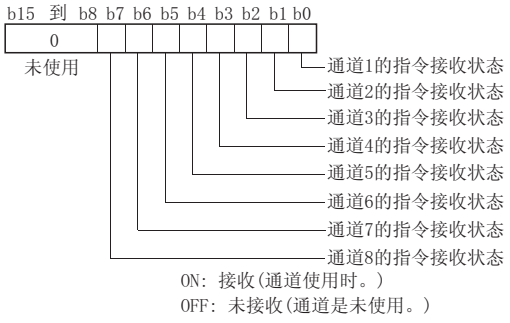
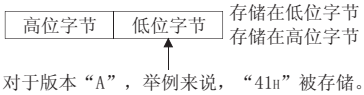
*14: 以序列号的高 5 位为 “08032”以后的 CPU 为对象。

附表 4.4 特殊寄存器列表

号码	名称	含义		解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD297	软元件分配（和参数内容相同）	分配给 V 的点数		• 存储当前为 V 软元件分配的点数	S（初始化）	新增	○
SD298		分配给 S 的点数		• 存储当前为 S 软元件分配的点数			
SD299		分配给 T 的点数		• 存储当前为 T 软元件分配的点数			
SD300		分配给 ST 的点数		• 存储当前为 ST 软元件分配的点数			
SD301		分配给 C 的点数		• 存储当前为 C 软元件分配的点数			
SD302		分配给 D 的点数		• 存储当前为 D 软元件分配的点数			○ Rem
SD303		分配给 W 的点数		• 存储当前为 W 软元件分配的点数			
SD304		分配给 SW 的点数		• 存储当前为 SW 软元件分配的点数			
SD305	软元件分配（变址寄存器）	16 位修饰 Z 分配点数		• 存储以 16 位范围修饰的变址寄存器（Z）的点数。（通过参数的 ZR 软元件的变址修饰设置）	S（初始化）	新增	QnU
SD315	为通讯处理预留的时间	为通讯处理预留的时间		• 为与 GX Developer 或者其他单元进行的通讯处理预留指定的时间。 • 指定的值越大，与其他软元件（GX Developer，串行通讯单元）通讯的响应时间就越短。 • 如果指定的值超出了下面的范围，则按没有设置处理。 • 设置范围：1 到 100ms • 扫描时间将延长指定的时间。	U	新增	QCPU Rem
SD340	以太网信息	安装的模块数		• 表示安装的以太网模块数。	S（初始化）	新增	
SD341		来自第 1 个模块的信息	I/O 号	• 表示安装的以太网模块的 I/O 号			
SD342			网络号	• 表示安装的以太网模块的网络号			
SD343			组号	• 表示安装的以太网模块的组号			
SD344			站号	• 表示安装的以太网模块的站号			
SD345 到 SD346			空置	• 空置 （对于 QCPU，第一个模块的以太网 IP 地址被存储在缓冲存储区内。）			Qn(H) QnPH QnPRH QnU Rem
SD347			空置	• 空置 （对于 QCPU，用 ERRORRD 指令读取第一个模块的以太网出错代码。）			
SD348 到 SD354	以太网信息	来自第 2 个模块的信息		• 配置和第一个模块的相同。	S（初始化）	新增	
SD355 到 SD361		来自第 3 个模块的信息		• 配置和第一个模块的相同。			Qn(H) QnPH QnPRH QnU*10 Rem
SD362 到 SD368		来自第 4 个模块的信息		• 配置和第一个模块的相同。			

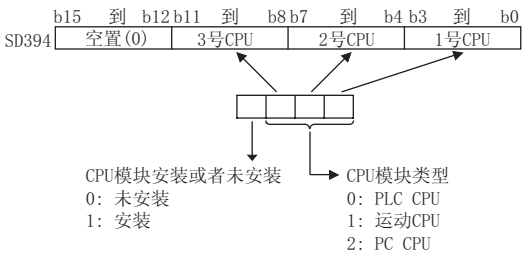
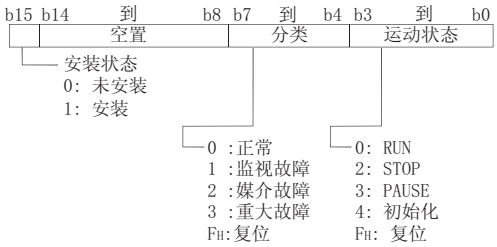
*10：除 Q02UCPU 外的通用型 QCPU。

附表 4.4 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th colspan="2">含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD340</td><td rowspan="10">以太网信息</td><td colspan="2">安装的模块数</td><td>• 表示安装的以太网模块数。</td><td rowspan="10">S（初始化）</td><td rowspan="10">新增</td><td rowspan="10">QnA</td></tr>
<tr><td>SD341</td><td rowspan="6">来自第 1 个模块的信息</td><td>I/O 号</td><td>• 表示安装的以太网模块的 I/O 号</td></tr>
<tr><td>SD342</td><td>网络号</td><td>• 表示安装的以太网模块的网络号</td></tr>
<tr><td>SD343</td><td>组号</td><td>• 表示安装的以太网模块的组号</td></tr>
<tr><td>SD344</td><td>站号</td><td>• 表示安装的以太网模块的站号</td></tr>
<tr><td>SD345
到
SD346</td><td>IP 地址</td><td>• 表示安装的以太网模块的 IP 地址</td></tr>
<tr><td>SD347</td><td>出错代码</td><td>• 表示安装的以太网模块的出错代码</td></tr>
<tr><td>SD348
到
SD354</td><td colspan="2">来自第 2 个模块的信息</td><td>• 配置和第一个模块的相同。</td></tr>
<tr><td>SD355
到
SD361</td><td colspan="2">来自第 3 个模块的信息</td><td>• 配置和第一个模块的相同。</td></tr>
<tr><td>SD362
到
SD368</td><td colspan="2">来自第 4 个模块的信息</td><td>• 配置和第一个模块的相同。</td></tr>
<tr><td rowspan="2">SD380</td><td rowspan="2">以太网指令接收状态</td><td colspan="2" rowspan="2">第 1 个模块的指令接收状态</td><td rowspan="2"> 
</td><td>S（初始化）</td><td rowspan="2">新增</td><td>QnA</td></tr>
<tr><td>S（指令执行）</td><td>QnPRH</td></tr>
<tr><td>SD381</td><td rowspan="3">以太网指令接收状态</td><td colspan="2">第 2 个模块的指令接收状态</td><td>• 配置和第一个模块的相同。</td><td rowspan="3">S（指令执行）</td><td rowspan="3">新增</td><td rowspan="3">QnPRH</td></tr>
<tr><td>SD382</td><td colspan="2">第 3 个模块的指令接收状态</td><td>• 配置和第一个模块的相同。</td></tr>
<tr><td>SD383</td><td colspan="2">第 4 个模块的指令接收状态</td><td>• 配置和第一个模块的相同。</td></tr>
<tr><td>SD392</td><td>软件版本</td><td colspan="2">内部系统软件版本</td><td>• 以 ASCII 码存储内部系统软件版本。
 

对于版本“A”，举例来说，“41H”被存储。
注释 :内部系统软件版本可能和打印在外壳上的版本符号所指示的版本不同。</td><td>S（初始化）</td><td>D9060</td><td>QnA</td></tr>
</table>

附表 4.4 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD393	多 CPU 系统信息	多 CPU 数	• 组成多可编程控制器系统的 CPU 模块数被存储。(1 到 3，空置的也包括）	S（初始化）	新增	Q00/Q01*9 QnU
SD394		CPU 安装信息	• 1 号 CPU 到 3 号 CPU 的 CPU 模块类型以及是否安装 CPU 模块，这些信息被存储。 			Q00/Q01*9
SD395		多 CPU 号	• 在多 CPU 系统配置中，主站 CPU 的 CPU 号被存储。 1 号 CPU：1，2 号 CPU：2，3 号 CPU：3，4 号 CPU：4	S（初始化）	新增	Q00/Q01*9 Qn(H)*9 QnPH QnU
SD396		1 号 CPU 运行状态	在多 CPU 系统配置中，本站 CPU 的 CPU 号被存储。 （在 SD393 中指示的多可编程控制器的号码信息被存储。） 	S（初始化 / END 处理出错发生时）	新增	Q00/Q01*9 QnU
SD397		2 号 CPU 运行状态				
SD398		3 号 CPU 运行状态				
SD399		4 号 CPU 运行状态				QnU

*9：适用于功能版本为 B 或者更高的 CPU。

(3) 系统时钟 / 计数器

附表 4.5 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD412	1 秒计数器	以 1 秒为单位计数	• 下列可编程控制器 CPU 模块 RUN 时，每秒增加 1 • 计数在 0 到 32767 到 -32768 到 0 之间重复	S（状态改变）	D9022	○
SD414	2n 秒时钟设置	2n 秒时钟单位	• 存储 2n 秒时钟的值 n（默认是 30） • 设置可以在 1 到 32767 之间进行	U	新增	
SD415	2nms 时钟设置	2nms 时钟单位	• 存储 2nms 时钟的值 n（默认是 30） • 设置可以在 1 到 32767 之间进行	U	新增	Qn(H) QnPH QnPRH QnU
SD420	扫描计数器	每个扫描周期的计数值	• 在 CPU 模块被设定为 RUN 之后，每个扫描执行增加 1。 （初始化执行类型程序中，扫描周期不计数。） • 计数在 0 到 32767 到 -32768 到 0 之间重复	S（每个 END 处理）	新增	QnA Qn(H) QnPH QnPRH QnU
			• 在 CPU 模块被设定为 RUN 之后，每个扫描执行增加 1。 • 计数在 0 到 32767 到 -32768 到 0 之间重复	S（每个 END 处理）	新增	Q00J/Q00/Q01
SD430	低速扫描计数器	在每个扫描周期中的计数值	• 在 CPU 模块被设定为 RUN 之后，每个扫描执行增加 1。 • 计数在 0 到 32767 到 -32768 到 0 之间重复 • 只用于低速执行类型程序	S（每个 END 处理）	新增	QnA Qn(H) QnPH

(4) 扫描周期信息

附表 4.6 特殊寄存器列表

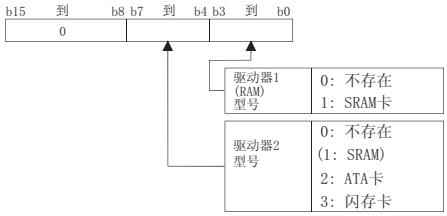
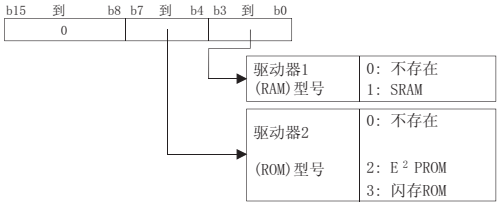
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD500	执行程序号	正在执行的程序号	• 当前正在执行的程序的程序号被以 BIN 值存储。	S（状态改变）	新增	QnA Qn(H) QnPH QnPRH QnU
SD510	低速执行类型程序号	正在执行的低速执行类型程序号	• 当前正在执行的低速执行类型程序的程序号被以 BIN 值存储。 • 只有当 SM510 是 ON 时才允许。	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SD520	当前扫描周期时间	当前扫描周期时间（以 1ms 为单位）	• 当前扫描时间被存储到 SD520 和 SD521 中。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD520：存储 ms 的值。（存储范围：0 到 65535） SD521：存储μs 的值。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999）） （实例）当当前扫描时间是 23.6ms 时，下列值被存储。 SD520 = 23 SD521 = 600	S（每个 END 处理）	D9017 格式改变	QnA Q00J/Q00/Q01 Qn(H) QnPH QnPRH QnU
SD521		当前扫描周期时间（以 100μs 为单位）		S（每个 END 处理）	新增	
SD522	初始化扫描周期时间	初始化扫描周期时间（以 1ms 为单位）	• 存储初始化执行类型程序的扫描时间到 SD522 和 SD523。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD522：存储 ms 的地方。（存储范围：0 到 65535） SD523：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999））	S（每个 END 处理）	新增	QnA Qn(H) QnPH QnPRH QnU
SD523		初始化扫描时间（以 100μs 为单位）				
SD524	最小扫描周期时间	最小时间（以 1ms 单位）	• 存储扫描时间的最小值到 SD524 和 SD525 中，初始化执行类型程序除外（测量是以 100μs 为单位。） （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD524：存储 ms 的地方。（存储范围：0 到 65535） SD525：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999））	S（每个 END 处理）	D9018 格式改变	
SD525		最小扫描周期时间（以 100μs 为单位）		S（每个 END 处理）	新增	
SD526	最大扫描周期时间	最大扫描周期时间（以 1ms 为单位）	• 存储扫描时间的最大值到 D526 和 SD527 中，初始化执行类型程序除外。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD526：存储 ms 的地方。（存储范围：0 到 65535） SD527：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999））	S（每个 END 处理）	D9019 格式改变	
SD527		最大扫描周期时间（以 100us 为单位）			新增	
SD528	低速执行类型程序的当前扫描周期时间	当前扫描时间（以 1ms 为为单位）	• 存储低速执行类型程序的当前扫描时间到 SD528 和 SD529。 （测量是以 100μs 为单位。） SD528：存储 ms 的地方。（存储范围：0 到 65535） SD529：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SD529		当前扫描时间（以 100μs 为单位）				
SD532	低速执行类型程序的最小扫描周期时间	最小扫描周期时间（以 1ms 为单位）	• 存储低速执行类型程序的扫描时间的最小值到 SD532 和 SD533。 （测量是以 100μs 为单位。） SD532：存储 ms 的地方。（存储范围：0 到 65535） SD533：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理	新增	
SD533		最小扫描周期时间（以 100μs 为单位）				
SD534	低速执行类型程序的最大扫描周期时间	最大扫描周期时间（以 1ms 为单位）	• 存储低速执行类型程序的扫描时间的最大值到 SD534 和 SD535，不包括第一个扫描周期的时间。 （测量是以 100μs 为单位。） SD534：存储 ms 的地方。（存储范围：0 到 65535） SD535：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理）	新增	
SD535		最大扫描周期时间（以 100μus 为单位）				
SD540	END 处理时间	END 处理时间（以 1ms 为单位）	• 存储从扫描周期执行类型程序结束到下一个扫描周期的开始之间的时间到 SD540 和 SD541。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD540：存储 ms 的地方。（存储范围：0 到 65535） SD541：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999））	S（每个 END 处理）	新增	QnA Qn(H) QnPH QnPRH QnU
SD541		END 处理时间（以 100μs 为单位）				
SD524	最小扫描周期时间	最小扫描周期时间（以 1ms 为单位）	• 存储扫描时间的最小值到 SD524 和 SD525。 （测量是以 100μs 为单位。） SD524：存储 ms 的地方。（存储范围：0 到 65535） SD525：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理）	新增	Q00J/Q00/Q01
SD525		最小扫描周期时间（以 100μs 为单位）				
SD526	最大扫描周期时间	最大扫描周期时间（以 1ms 为单位）	• 存储扫描时间的最大值到 SD526 和 SD527。 （测量是以 100μs 为单位。） SD526：存储 ms 的地方。（存储范围：0 到 65535） SD527：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理）		
SD527		最大扫描周期时间（以 100μs 为单位）				

附表 4.6 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD540	END 处理时间	END 处理时间（以 1ms 为单位）	• 存储从当前扫描周期程序结束到下一个扫描周期启动之间的时间到 SD540 和 SD541。 （测量是以 100μs 为单位。） SD540：存储 ms 的地方。（存储范围：0 到 65535） SD541：存储μs 的地方。（存储范围：0 到 900）	S（每个 END 处理）	新增	Q00J/Q00/Q01
SD541		END 处理时间（以 100μs 为单位）				
SD542	恒定扫描周期等待时间	恒定扫描周期等待时间（以 1ms 为单位）	• 存储恒定扫描设置的等待时间到 SD542 和 SD543。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD542：存储 ms 的地方。（存储范围：0 到 65535） SD543：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999））	S（每个 END 处理）	新增	○
SD543		恒定扫描周期等待时间（以 100μs 为单位）				
SD544	低速执行类型程序的累积执行时间	低速执行类型程序的累积执行时间（以 1ms 为单位）	• 存储低速执行类型程序的累积执行时间到 SD544 和 SD545。 （测量是以 100μs 为单位。） SD544：存储 ms 的地方。（存储范围：0 到 65535） SD545：存储μs 的地方。（存储范围：0 到 900） • 在一个低速扫描周期结束之后清除到 0。	S（每个 END 处理）	新增	QnA Qn(H) QnPH
SD545		低速执行类型程序的累积执行时间（以 100μs 为单位）				
SD546	低速执行类型程序的执行时间	低速执行类型程序的执行时间（以 1ms 为单位）	• 在 1 个扫描周期内低速执行类型程序存储执行时间的 SD546 和 SD547。 （测量是以 100μs 为单位。） SD546：存储 ms 的地方。（存储范围：0 到 65535） SD547：存储μs 的地方。（存储范围：0 到 900） • 在每个扫描周期存储。	S（每个 END 处理）	新增	
SD547		低速执行类型程序的执行时间（以 100μs 为单位）				
SD548	扫描执行类型程序执行时间	扫描执行类型程序执行时间（以 1ms 为单位）	• 在 1 个扫描周期内存储扫描执行类型程序执行时间到 SD548 和 SD549。 （测量是以 100μs 为单位。） SD548：存储 ms 的地方。（存储范围：0 到 65535） SD549：存储μs 的地方。（存储范围：0 到 900） • 在每个扫描周期存储。	S（每个 END 处理）	新增	QnA Qn(H) QnPH QnPRH
SD549		扫描执行类型程序执行时间（以 100μs 为单位）				
SD548	扫描程序执行时间	扫描程序执行时间（以 1ms 为单位）	• 在 1 个扫描内存储扫描程序执行时间到 SD548 和 SD549。 （测量是以 100μs 为单位。（对于通用型 QCPU，以 1μs 为单位。）） SD548：存储 ms 的地方。（存储范围：0 到 65535） SD549：存储μs 的地方。（存储范围：0 到 900（对于通用型 QCPU，存储范围：0 到 999）） • 在每个扫描周期存储。	S（每个 END 处理）	新增	Q00J/Q00/Q01 QnU
SD549		扫描程序执行时间（以 100μs 为单位）				
SD550	服务间隔测量模块	基板 / 模块号	• 为测量服务间隔的模块设定 I/O 号	U	新增	QnA Qn(H) QnPH QnPRH
SD551	服务间隔时间	模块服务间隔（以 1ms 为单位）	• 当 SM551 变为 ON 时，将在 SD550 中指定的模块的服务间隔存储到 SD551 和 SD552。 （测量是以 100μs 为单位。） SD551：存储 ms 的地方。（存储范围：0 到 65535） SD552：存储μs 的地方。（存储范围：0 到 900）	S（请求）	新增	
SD552		模块服务间隔（以 100μs 为单位）				

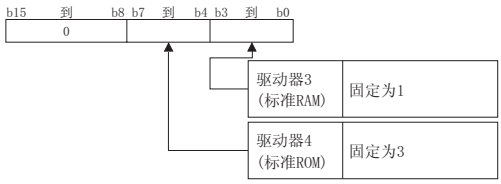
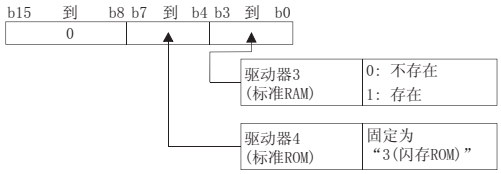
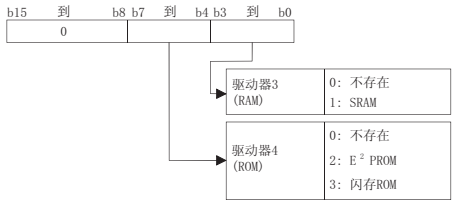
(5) 驱动器信息

附表 4.7 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD600	存储卡型号	存储卡型号	• 表示安装的存储卡型号 	S（初始化和卡卸载）	新增	Qn(H) QnPH QnPRH QnU
	存储卡 A 型号	存储卡 A 型号	• 表示安装的存储卡 A 型号 	S（初始化和卡卸载）	新增	QnA
SD602	驱动器 1(RAM 存储卡）容量	驱动器 1 容量	• 驱动器 1 容量被以 1k 字节为单位存储。 （存储格式化后的空余容量）	S（初始化和卡卸载）	新增	QnA Qn(H) QnPH QnPRH QnU
SD603	驱动器 2（ ROM 存储卡）容量	驱动器 2 的容量	• 驱动器 2 容量被以 1k 字节为单位存储。*1 （存储格式化后的空余容量）	S（初始化和卡卸载）	新增	Qn(H) QnPH QnPRH QnU
			• 驱动器 2 容量被以 1k 字节为单位存储。	S（初始化和卡卸载）	新增	QnA
SD604	存储卡使用情况	存储卡使用情况	• 存储卡 (A) 的使用情况以位模式存储。 （当 ON 时为使用中） • 这些位模式的意义如下所示： b0 ： 引导操作(QBT) b1 ： 参数(QPA) b2 ： 软元件注释(QCD) b3 ： 软元件初始化值(QDI) b4 ： 文件寄存器R(QDR) b5 ： 采样跟踪(QTD) b6 ： 未使用 b7 ： 未使用 b8 ： 未使用 b9 ： CPU故障历史(QFD) b10 ： 未使用 b11 ： 局部软元件(QDL) b12 ： 未使用 b13 ： 未使用 b14 ： 未使用 b15 ： 未使用	S（状态改变）	新增	Qn(H) QnPH QnPRH
	存储卡 A 使用情况	存储卡 A 使用情况	• 存储卡 A 的使用情况以位模式存储。 （当 ON 为使用中） • 这些位模式的意义如下所示： b0 ： 引导操作(QBT) b1 ： 参数(QPA) b2 ： 软元件注释(QCD) b3 ： 软元件初始化值(QDI) b4 ： 文件R(QDR) b5 ： 采样跟踪(QTD) b6 ： 状态锁存(QTL) b7 ： 程序跟踪(QTP) b8 ： 仿真数据(QDS) b9 ： CPU故障历史(QFD) b10 ： SFC跟踪(QTR) b11 ： 局部软元件(QDL) b12 ： 未使用 b13 ： 未使用 b14 ： 未使用 b15 ： 未使用	S（状态改变）	新增	QnA

*1：使用了 Q2MEM-8MBA 时，根据高性能型 QCPU 的序列号和 ATA 卡的生产管理编号的不同，特殊寄存器 SD603 中存储的值也不相同。
详细内容请参阅 7.1.2 项。

附表 4.7 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD604	存储卡使用情况	存储卡使用情况	• 存储卡 (A) 的使用情况以位模式存储。 （当 ON 时为使用中） • 这些位模式的意义如下所示： b0 : 引导操作(QBT)*1　b8 : 未使用 b1 : 参数(QPA)　b9 : 未使用 b2 : 软元件注释(QCD)　b10 : 未使用 b3 : 软元件初始化值(QDI)*2　b11 : 局部软元件(QDL) b4 : 文件寄存器R(QDR)　b12 : 未使用 b5 : 采样跟踪(QTD)　b13 : 未使用 b6 : 未使用　b14 : 未使用 b7 : 未使用　b15 : 未使用 *1: 引导开始时 ON，结束时 OFF。 *2: 软元件初始值反映开始时 ON，结束时 OFF。	S（状态改变）	新增	QnU
SD620	驱动器 3/4 型号	驱动器 3/4 型号	• 表示驱动器 3/4 型号。 	S（初始化）	新增	Qn(H) QnPH QnPRH QnU
			• 表示驱动器 3/4 型号。 	S（初始化）	新增	Q00J/Q00/Q01
	存储卡 B 型号	存储卡 B 型号	• 表示安装的存储卡 B 型号 	S（初始化 / 卡安装和卸载）	新增	Q2A(S1) Q3A Q4A Q4AR
SD622	驱动器 3（标准 RAM）容量	驱动器 3(RAM) 容量	• 驱动器 3 容量以 1k 字节为单位存储。 （存储格式化后的空余容量）	S（初始化）	新增	Qn(H) QnPH QnPRH QnU
				S（初始化 / 卡安装和卸载）	新增	Q2A(S1) Q3A Q4A Q4AR
SD623	驱动器 4（标准 ROM）容量	驱动器 4 容量	• 驱动器 4 容量以 1k 字节为单位存储。 （存储格式化后的空余容量）	S（初始化）	新增	Qn(H) QnPH QnPRH QnU
				S（初始化 / 卡安装和卸载）	新增	Q2A(S1) Q3A Q4A Q4AR

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
	驱动器 3/4 使用情况	驱动器 3/4 使用情况	• 驱动器 3/4 的使用情况以位模式存储。（当 ON 时为使用中） • 这些位模式的意义如下所示： b0 : 引导操作(QBT) b1 : 参数(QPA) b2 : 软元件注释(QCD) b3 : 软元件初始化值(QDI) b4 : 文件R(QDR) b5 : 采样跟踪(QTD) b6 : 未使用 b7 : 未使用 b8 : 未使用 b9 : CPU故障历史(QFD) b10 : SFC跟踪(QTS) b11 : 局部软元件(QDL) b12 : 未使用 b13 : 未使用 b14 : 未使用 b15 : 未使用	S（状态改变）	新增	Qn(H) QnPH QnPRH
SD624	存储卡 B 使用情况	存储卡 B 使用情况	• 存储卡 B 的使用情况以位模式存储。（当 ON 时为使用中） • 这些位模式的意义如下所示： b0 : 引导操作(QBT) b1 : 参数(QPA) b2 : 软元件注释(QCD) b3 : 软元件初始化值(QDI) b4 : 文件R(QDR) b5 : 采样跟踪(QTD) b6 : 状态锁存(QTL) b7 : 程序跟踪(QTP) b8 : 仿真数据(QDS) b9 : CPU故障历史(QFD) b10 : SFC跟踪(QTS) b11 : 局部软元件(QDL) b12 : 局部软元件(QDL) b13 : 未使用 b14 : 未使用 b15 : 未使用	S（状态改变）	新增	Q2A(S1) Q3A Q4A Q4AR
	驱动器 3/4 使用情况	驱动器 3/4 使用情况	• 驱动器 3/4 的使用情况以位模式存储。（当 ON 时为使用中） • 这些位模式的意义如下所示： b0 : 未使用 b1 : 参数(QPA) b2 : 软元件注释(QCD) b3 : 软元件初始化值(QDI)*1 b4 : 文件寄存器R(QDR) b5 : 采样跟踪(QTD) b6 : 未使用 b7 : 未使用 b8 : 未使用 b9 : 未使用 b10 : 未使用 b11 : 局部软元件(QDL) b12 : 未使用 b13 : 未使用 b14 : 未使用 b15 : 未使用 *1: 引导开始时 ON，结束时 OFF。	S（状态改变）	新增	QnU

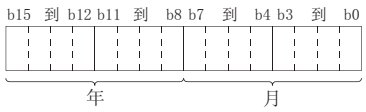
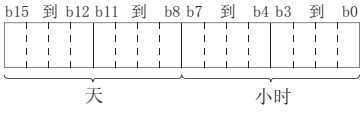
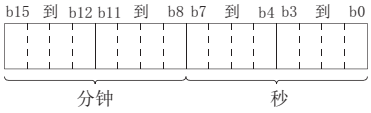
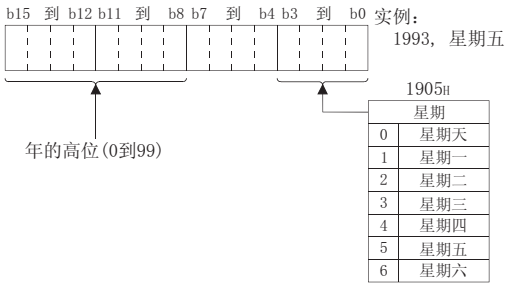
附表 4.7 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD622</td><td>驱动器 3（标准 RAM）容量</td><td>驱动器 3 容量</td><td>• 驱动器 3 容量以 1k 字节为单位存储。</td><td>S（初始化）</td><td rowspan="3">新增</td><td rowspan="3">Q00J/Q00/Q01</td></tr>
<tr><td>SD623</td><td>驱动器 4（标准 ROM）容量</td><td>驱动器 4 容量</td><td>• 驱动器 4 容量以 1k 字节为单位存储。</td><td>S（初始化）</td></tr>
<tr><td>SD624</td><td>驱动器 3 使用情况</td><td>驱动器 3 使用情况</td><td>• 驱动器 3 使用情况以位模式存储。
b15 到 b5 b4 到 b0
0 | 0 | 0 | 0 | 0
文件寄存器(R)
0：未使用
1：使用中</td><td>S（状态改变）</td></tr>
<tr><td>SD640</td><td>文件寄存器驱动器</td><td>驱动器号：</td><td>• 存储文件寄存器正在使用的驱动器号</td><td>S（状态改变）
*10</td><td>新增</td><td>○</td></tr>
<tr><td>SD641
SD642
SD643
SD644
SD645</td><td rowspan="2">文件寄存器文件名称</td><td rowspan="2">文件寄存器文件名称</td><td>• 以 ASCII 代码存储在参数上选择的文件寄存器文件名称（带扩展）或者 QDRSET 指令使用的文件寄存器文件名称。
b15 到 b8 | b7 到 b0
SD641：第2个字符 | 第1个字符
SD642：第4个字符 | 第3个字符
SD643：第6个字符 | 第5个字符
SD644：第8个字符 | 第7个字符
SD645：扩展的第1个字符 | 2EH(.)
SD646：扩展的第3个字符 | 扩展的第2个字符</td><td>S（状态改变）</td><td>新增</td><td>QnA
Qn(H)
QnPH
QnPRH
QnU</td></tr>
<tr><td>SD646</td><td>• 以 ASCII 代码存储在参数上选择的文件寄存器文件名称(MAIN.QDR)。
b15 到 b8 | b7 到 b0
SD641：第2个字符(A) | 第1个字符(M)
SD642：第4个字符(N) | 第3个字符(I)
SD643：第6个字符() | 第5个字符()
SD644：第8个字符() | 第7个字符()
SD645：扩展的第1个字符(Q) | 2EH(.)
SD646：扩展的第3个字符(R) | 扩展的第2个字符(D)</td><td>S（初始化）</td><td>新增</td><td>Q00J/Q00/Q01</td></tr>
<tr><td>SD647</td><td>文件寄存器容量</td><td>文件寄存器容量</td><td>• 以 1k 字为单位存储当前选择的文件寄存器的数据容量。</td><td>S（状态改变）</td><td rowspan="2">新增</td><td>QnA
Qn(H)
QnPH
QnPRH
QnU</td></tr>
<tr><td></td><td></td><td></td><td></td><td>S（初始化）</td><td>Q00J/Q00/Q01</td></tr>
<tr><td>SD648</td><td>文件寄存器块号</td><td>文件寄存器块号</td><td>• 存储当前选择的文件寄存器块号。</td><td>S（状态改变）
*10</td><td>D9035</td><td>○</td></tr>
<tr><td>SD650</td><td>注释驱动器号</td><td>注释驱动器号</td><td>• 存储在参数上选择的注释驱动器号或者 QCDSET 指令使用的注释驱动器号。</td><td>S（状态改变）</td><td>新增</td><td rowspan="7">QnA
Qn(H)
QnPH
QnPRH
QnU</td></tr>
<tr><td>SD651
SD652
SD653
SD654
SD655</td><td rowspan="2">注释文件名称</td><td rowspan="2">注释文件名称</td><td rowspan="2">• 以 ASCII 代码存储在参数上选择的文件寄存器文件名称（带扩展）或者 QCDSET 指令使用的文件寄存器文件名称。
b15 到 b8 | b7 到 b0
SD651：第2个字符 | 第1个字符
SD652：第4个字符 | 第3个字符
SD653：第6个字符 | 第5个字符
SD654：第8个字符 | 第7个字符
SD655：扩展的第1个字符 | 2EH(.)
SD656：扩展的第3个字符 | 扩展的第2个字符</td><td rowspan="2">S（状态改变）</td><td rowspan="2">新增</td></tr>
<tr><td>SD656</td></tr>
</table>

*10：在基本型 QCPU 中，在参数执行之后，在 SEOP 到 RUN 或者 RSET 指令执行时设定数据。

附表 4.7 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD660</td><td rowspan="7">引导操作指定文件</td><td>引导指定文件驱动器号</td><td>・存储正被存储的引导启动指定文件（*.QBT）所在的驱动器号。</td><td>S（初始化）</td><td>新增</td><td rowspan="7">QnA
Qn(H)
QnPH
QnPRH</td></tr>
<tr><td>SD661</td><td rowspan="6">引导指定文件的文件名称</td><td rowspan="6">・存储引导启动指定文件的文件名（*.QBT）。
b15 到 b8 b7 到 b0
SD661 第2个字符 第1个字符
SD662 第4个字符 第3个字符
SD663 第6个字符 第5个字符
SD664 第8个字符 第7个字符
SD665 扩展的第1个字符 2EH(.)
SD666 扩展的第3个字符 扩展的第2个字符</td><td rowspan="6">S（初始化）</td><td rowspan="6">新增</td></tr>
<tr><td>SD662</td></tr>
<tr><td>SD663</td></tr>
<tr><td>SD664</td></tr>
<tr><td>SD665</td></tr>
<tr><td>SD666</td></tr>
<tr><td>SD670</td><td>参数有效驱动器信息</td><td>参数有效驱动号</td><td>・存储有效的参数存储目标驱动器的信息。
0：驱动器 0（程序内存）
1：驱动器 1(SRAM 卡）
2：驱动器 2（闪存卡 /ATA 卡）
4：驱动器 4（标准 ROM）</td><td>S（初始化）</td><td>新增</td><td>QnU</td></tr>
<tr><td>SD671</td><td>锁存数据备份功能的状态</td><td>状态显示</td><td>显示锁存数据备份功能的状态。
状态 | 有无备份数据 | 下一次以后的电源OFF→ON时的还原动作
0 无备份数据 | 无 | 不进行还原。
1 还原准备完毕 | 有 | 仅在下一次的电源OFF→ON时进行还原。
2 还原执行完毕 | 有 | 不执行还原。
3 备份执行等待 | 有 | 不执行还原。
4 反复还原执行准备完毕 | 有 | 每次电源OFF→ON时执行还原。
・“2：还原执行完毕”为还原执行之后的状态。
“3：备份执行等待”为“2：还原执行完毕”时电源 OFF → ON 后的状态。</td><td>S（状态改变）</td><td>新增</td><td>QnU</td></tr>
<tr><td>SD672</td><td rowspan="4">备份信息</td><td>备份时间
（年，月）</td><td>・以 BCD 码的 2 位存储备份的年（后两位数）和月：
b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例：
7月，1993
9307H
年 月</td><td rowspan="4">S（写入时）</td><td rowspan="4">新增</td><td rowspan="4">QnU</td></tr>
<tr><td>SD673</td><td>备份时间
（日期，小时）</td><td>・以 BCD 码的 2 位存储备份的日期和小时：
b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例：
31日，上午10
3110H
天 小时</td></tr>
<tr><td>SD674</td><td>备份时间
（分钟，秒）</td><td>・以 BCD 码的 2 位存储备份的分钟和秒：
b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例：
35分钟，48秒。
3548H
分钟 秒</td></tr>
<tr><td>SD675</td><td>备份时间
（年的前位，星期）</td><td>・以 BCD 码的 2 位存储备份的年（前两位）和星期。
b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例：
1993，星期五
1905H
年的高位(0到99)
星期
0 星期天
1 星期一
2 星期二
3 星期三
4 星期四
5 星期五
6 星期六</td></tr>
</table>

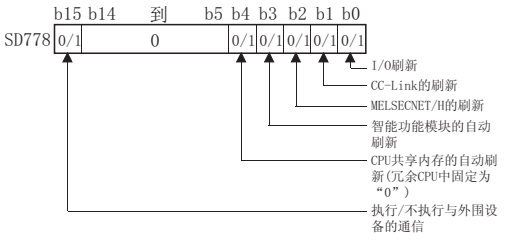
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD676	备份数据还原信息	还原时间（年，月）	• 以 BCD 码的 2 位存储还原的年（后两位数）和月：  实例： 7月，1993 9307H	S（初始化）	新增	QnU
SD677		还原时间（日期，小时）	• 以 BCD 码的 2 位存储还原的日期和小时：  实例： 31日，上午10 3110H			
SD678		还原时间（分钟，秒）	• 以 BCD 码的 2 位存储还原的分钟和秒：  实例： 35分钟，48秒。 3548H			
SD679		还原时间（年的前位，星期）	• 以 BCD 码的 2 位存储还原的年（前两位）和星期。  实例： 1993，星期五			
SD681	程序内存写入（传输）状况	写入（传输）状况显示 (%)	• 以百分比显示至程序内存（闪存 ROM）的写入（传输）状况。在有 (0 ～ 100%) 写入指示的时点设置为 “0”。	S（写入时）	新增	QnU
SD682	程序内存写入次数指标	至目前为止的写入次数指标	• 以 32 位的 BIN 值存储至目前为止的程序内存（闪存 ROM）写入操作次数的指标值。 指标值超过了 10 万次时将发生 “FLASH ROM ERROR”（出错代码：1610）。（指标值超过 10 万次后依然计数。） 注）写入次数并不等于指标值。（通过系统执行闪存 ROM 的写入寿命延长功能，使得约 2 次写入操作只增加指标值 1。）	S（写入时）	新增	QnU
SD683						
SD686	标准 ROM 写入（传输）状况	写入（传输）状况显示 (%)	• 以百分比显示至标准 ROM（闪存 ROM）的写入（传输）状况。在有 (0 ～ 100%) 写入指示的时点设置为 “0”。	S（写入时）	新增	QnU
SD687	标准 ROM 写入次数指标	至目前为止的写入次数指标	• 以 32 位的 BIN 值存储至目前为止的标准 ROM（闪存 ROM）写入操作次数的指标值。 指标值超过了 10 万次时将发生 “FLASH ROM ERROR”（出错代码：1610）。（指标值超过 10 万次后依然计数。） 注）写入次数并不等于指标值。（通过系统执行闪存 ROM 的写入寿命延长功能，使得上一次 “计数到” 后的合计写入容量达到约 1M 字节后增加指标值 1。）	S（写入时）	新增	QnU
SD695	至标准 ROM 的写入指令执行次数指定	指定至标准 ROM 的写入指令执行次数	• 指定 1 日内的标准 ROM 写入指令 (SP.DEVST) 的最多执行次数。 • 标准 ROM 写入指令的执行次数超过了 SD 中设置的次数时，将发生 “OPERATION ERROR”（出错代码：4113）。 • SD695 的设置范围为 0 ～ 32767。设置了 0 或者超出范围的值时，执行标准 ROM 写入指令时将发生 “OPERATION ERROR”（出错代码：4113）。	U	新增	QnU

(6) 指令相关寄存器

附表 4.8 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD705
SD706</td><td>掩码模式</td><td>掩码模式</td><td>• 在块操作过程中，将 SM705 变为 ON 以便可以使用存储在 SD705（如果正在使用双字，则是 SD705 和 SD706）上的掩码模式，用掩码的值在块中所有的数据上进行操作。</td><td>U</td><td>新增</td><td>Q00J/Q00/Q01
Qn(H)
QnPH
QnPRH
QnA</td></tr>
<tr><td>SD714</td><td>空通讯请求注册区域的号</td><td>0 到 32</td><td>• 存储用于连接到 MELSECNET/MINI-S3 的远程端子模块的通讯请求区域中空块的号。</td><td>S（在执行过程中）</td><td>D9081</td><td>QnA</td></tr>
<tr><td>SD715
SD716
SD717</td><td>IMASK 指令掩码模式</td><td>掩码模式</td><td>• 使用 IMASK 指令掩码的模式以下列方式存储：
<table><tr><td></td><td>b15</td><td></td><td>b1</td><td>b0</td></tr><tr><td>SD715</td><td>115</td><td>到</td><td>11</td><td>10</td></tr><tr><td>SD716</td><td>131</td><td>到</td><td>117</td><td>116</td></tr><tr><td>SD717</td><td>147</td><td>到</td><td>133</td><td>132</td></tr></table></td><td>S（在执行过程中）</td><td>新增</td><td rowspan="3">○</td></tr>
<tr><td>SD718
SD719</td><td>累加器</td><td>累加器</td><td>• 用作在 A 系列程序中使用的累加器的替代。</td><td>S/U</td><td>新增</td></tr>
<tr><td>SD720</td><td>PLOADP 指令的程序号指定</td><td>PLOADP 指令的程序号指定</td><td>当 PLOADP 指令被指定时，存储 PLOADP 指令装载的程序的程序号。
指定范围：1 到 124</td><td>U</td><td>新增</td><td>Qn(H)
QnPH</td></tr>
<tr><td>SD730</td><td>CC-Link 通讯请求的空注册区域号</td><td>0 到 32</td><td>• 存储和连接到 A(1S)J61QBT61 的智能软元件站进行通讯的请求的空注册区域号。</td><td>S（在执行过程中）</td><td>新增</td><td rowspan="2">QnA</td></tr>
<tr><td>SD736</td><td>PKEY 输入</td><td>PKEY 输入</td><td>• 用 PKEY 指令暂时存储键盘数据输入的特殊寄存器。</td><td>S（在执行过程中）</td><td>新增</td></tr>
</table>

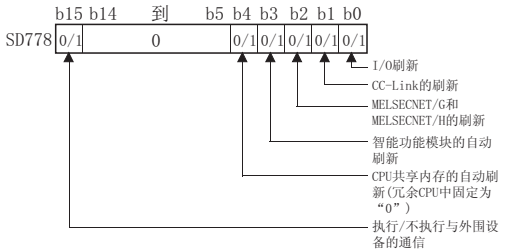
附表 4.8 特殊寄存器列表

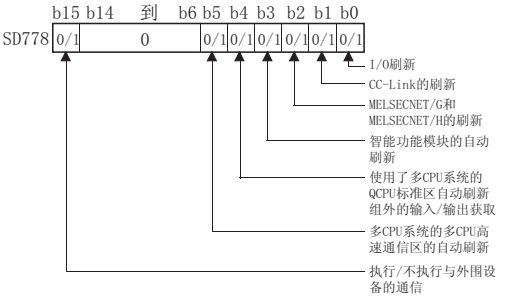
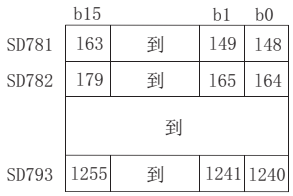
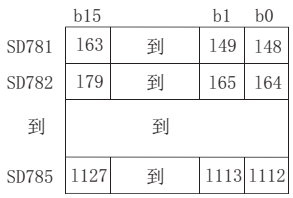
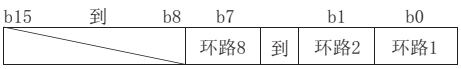
<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD738
SD739
SD740
SD741
SD742
SD743
SD744
SD745
SD746
SD747
SD748
SD749
SD750
SD751
SD752
SD753
SD754
SD755
SD756
SD757
SD758
SD759
SD760
SD761
SD762
SD763
SD764
SD765
SD766
SD767
SD768
SD769</td><td>消息存储</td><td>消息存储</td><td>• 存储 MSG 指令指定的消息。
<table><tr><th></th><th>b15 到 b8</th><th>b7 到 b0</th></tr><tr><td>SD738</td><td>第2个字符</td><td>第1个字符</td></tr><tr><td>SD739</td><td>第4个字符</td><td>第3个字符</td></tr><tr><td>SD740</td><td>第6个字符</td><td>第5个字符</td></tr><tr><td>SD741</td><td>第8个字符</td><td>第7个字符</td></tr><tr><td>SD742</td><td>第10个字符</td><td>第9个字符</td></tr><tr><td>SD743</td><td>第12个字符</td><td>第11个字符</td></tr><tr><td>SD744</td><td>第14个字符</td><td>第13个字符</td></tr><tr><td>SD745</td><td>第16个字符</td><td>第15个字符</td></tr><tr><td>SD746</td><td>第18个字符</td><td>第17个字符</td></tr><tr><td>SD747</td><td>第20个字符</td><td>第19个字符</td></tr><tr><td>SD748</td><td>第22个字符</td><td>第21个字符</td></tr><tr><td>SD749</td><td>第24个字符</td><td>第23个字符</td></tr><tr><td>SD750</td><td>第26个字符</td><td>第25个字符</td></tr><tr><td>SD751</td><td>第28个字符</td><td>第27个字符</td></tr><tr><td>SD752</td><td>第30个字符</td><td>第29个字符</td></tr><tr><td>SD753</td><td>第32个字符</td><td>第31个字符</td></tr><tr><td>SD754</td><td>第34个字符</td><td>第33个字符</td></tr><tr><td>SD755</td><td>第36个字符</td><td>第35个字符</td></tr><tr><td>SD756</td><td>第38个字符</td><td>第37个字符</td></tr><tr><td>SD757</td><td>第40个字符</td><td>第39个字符</td></tr><tr><td>SD758</td><td>第42个字符</td><td>第41个字符</td></tr><tr><td>SD759</td><td>第44个字符</td><td>第43个字符</td></tr><tr><td>SD760</td><td>第46个字符</td><td>第45个字符</td></tr><tr><td>SD761</td><td>第48个字符</td><td>第47个字符</td></tr><tr><td>SD762</td><td>第50个字符</td><td>第49个字符</td></tr><tr><td>SD763</td><td>第52个字符</td><td>第51个字符</td></tr><tr><td>SD764</td><td>第54个字符</td><td>第53个字符</td></tr><tr><td>SD765</td><td>第56个字符</td><td>第55个字符</td></tr><tr><td>SD766</td><td>第58个字符</td><td>第57个字符</td></tr><tr><td>SD767</td><td>第60个字符</td><td>第59个字符</td></tr><tr><td>SD768</td><td>第62个字符</td><td>第61个字符</td></tr><tr><td>SD769</td><td>第64个字符</td><td>第63个字符</td></tr></table></td><td>S（在执行过程中）</td><td>新增</td><td>QnA</td></tr>
<tr><td>SD774
到
SD775</td><td>PID 限制设置（用于完整微分）</td><td>0: 有限制
1: 无限制</td><td>• 按照如下所示，为每个 PID 环路指定限制：
<table><tr><th></th><th>b15</th><th></th><th>b1</th><th>b0</th></tr><tr><td>SD774</td><td>环路16</td><td>到</td><td>环路2</td><td>环路1</td></tr><tr><td>SD775</td><td>环路32</td><td>到</td><td>环路18</td><td>环路17</td></tr></table></td><td>U</td><td>新增</td><td>Qn(H)
QnPRH
QnU</td></tr>
<tr><td>SD774</td><td>PID 限制设置（用于完整微分）</td><td>0: 有限制
1: 无限制</td><td>• 按照如下所示，为每个 PID 环路指定限制：
<table><tr><th></th><th>b15 到 b8</th><th>b7</th><th>到</th><th>b1</th><th>b0</th></tr><tr><td>SD774</td><td></td><td>环路8</td><td>到</td><td>环路2</td><td>环路1</td></tr></table></td><td>U</td><td>新增</td><td>Q00J/Q00/Q01*9</td></tr>
<tr><td>SD778</td><td>执行 COM 指令时的刷新处理选择</td><td>b0 ～ b14:
0: 不刷新
1: 刷新
b15 位
0: 执行与 CPU 模块的通讯
1: 不执行与 CPU 模块的通讯</td><td>• 选择执行 COM 指令时，是否刷新数据。
• 当 SM775 变为 ON 时，SD778 的指定有效。
 

• 在下述情况下，根据 COM 指令在多 CPU 之间执行刷新。
来自于其它号 CPU 的接收动作 :SD778 的 b4(CPU 共享内存的自动刷新）为 1 时
来自于自站 CPU 的接收动作 :SD778 的 b15（执行 / 不执行与外围设备的通信）为 0 时</td><td>U</td><td>新增</td><td>Q00J/Q00/Q01*9
Qn(H)*11
QnPH*12
QnPRH</td></tr>
</table>

*9 : 以功能版本 B 以后的 CPU 为对象。
*11: 以序列号的高 5 位为 “04012”以后的 CPU 为对象
*12: 以序列号的高 5 位为 “07032”以后的 CPU 为对象

附表 4.8 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td rowspan="2">SD778</td><td rowspan="2">执行 COM 指令时的刷新处理选择</td><td rowspan="2">b0 ～ b14:
0: 不刷新
1: 刷新
b15 位
0: 执行与 CPU 模块的通讯
1: 不执行与 CPU 模块的通讯</td><td>• 选择执行 COM 指令时，是否刷新数据。
• 当 SM775 变为 ON 时，SD778 的指定有效。
 

• 在下述情况下，根据 COM 指令在多 CPU 之间执行刷新。
来自于其它号 CPU 的接收动作 :SD778 的 b4(CPU 共享内存的自动刷新）为 1 时
来自于自站 CPU 的接收动作 :SD778 的 b15（执行 / 不执行与外围设备的通信）为 0 时
• SD778 的 b2(MELSECNET/G 和 MELSECNET/H 的刷新）为 1 时，MELSECNET/G 和 MELSECNET/H 都将刷新。因此刷新点数较多时，COM 指令的处理时间将会延长。</td><td>U</td><td>新增</td><td>Qn(H)*13</td></tr>
<tr><td>• 选择执行 COM 指令时，是否刷新数据。
• 当 SM775 变为 ON 时，SD778 的指定有效。
 
</td><td>U</td><td>新增</td><td>QnU</td></tr>
<tr><td>SD780</td><td>同时执行的 CC-Link 专用指令的剩余数目</td><td>0 到 32</td><td>• 存储同时执行的 CC-Link 专用指令的剩余数目。</td><td>U</td><td>新增</td><td>QnA</td></tr>
<tr><td>SD781
到
SD793</td><td>IMASK 指令的掩码模式</td><td>掩码模式</td><td>• 按照如下所示，存储 IMASK 指令掩码的掩码模式：
 
</td><td>S（在执行过程中）</td><td>新增</td><td>Qn(H)
QnPH
QnPRH
QnU</td></tr>
<tr><td>SD781
到
SD785</td><td>IMASK 指令的掩码模式</td><td>掩码模式</td><td>• 如下所示，存储由 IMASK 指令掩码的掩码模式：
 
</td><td>S（在执行过程中）</td><td>新增</td><td>Q00J/Q00/Q01</td></tr>
<tr><td>SD794
到
SD795</td><td>PID 限制设置（用于不完整微分）</td><td>0: 限制设定
1: 限制未设定</td><td>• 按如下所示，指定每个 PID 环路的限制。
 
</td><td>U</td><td>新增</td><td>Qn(H)*13
QnPRH
QnU</td></tr>
<tr><td>SD794</td><td>PID 限制设置（用于不精确微分）</td><td>0: 有限制
1: 无限制</td><td>• 按如下所示，指定每个 PID 环路的限制。
 
</td><td>U</td><td>新增</td><td>Q00J/Q00/Q01*9</td></tr>
</table>

*9 : 适用于功能版本为 B 或者更高的 CPU。
*13: 以序列号的高 5 位为 “09012” 以后的 CPU 为对象

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD796	多 CPU 间高速总线对应专用指令最大使用块数设置（1 号 CPU 用）	专用指令最大使用块数范围 1～9（默认：2） *设置了 1～9 以外的值时按 9 执行动作。	・指定多 CPU 间高速总线对应专用指令（对象 CPU=1 号 CPU）的最多使用块数。对 1 号 CPU 执行多 CPU 间通信专用指令时，专用指令传输区的空余块数未达到本寄存器的设置值的情况下，使 SM796 为 ON。作为多 CPU 间通信专用指令的连续执行用互锁信号使用。	U(RUN 后 1 个扫描时）	新增	QnU*14
SD797	多 CPU 间高速总线对应专用指令最大使用块数设置（2 号 CPU 用）	专用指令最大使用块数范围 1～9（默认：2） *设置了 1～9 以外的值时按 9 执行动作。	・指定多 CPU 间高速总线对应专用指令（对象 CPU=2 号 CPU）的最多使用块数。对 2 号 CPU 执行多 CPU 间通信专用指令时，专用指令传输区的空余块数未达到本寄存器的设置值的情况下，使 SM797 为 ON。作为多 CPU 间通信专用指令的连续执行用互锁信号使用。	U(RUN 后 1 个扫描时）	新增	QnU*14
SD798	多 CPU 间高速总线对应专用指令最大使用块数设置（2 号 CPU 用）	专用指令最大使用块数范围 1～9（默认：2） *设置了 1～9 以外的值时按 9 执行动作。	・指定多 CPU 间高速总线对应专用指令（对象 CPU=3 号 CPU）的最多使用块数。对 3 号 CPU 执行多 CPU 间通信专用指令时，专用指令传输区的空余块数未达到本寄存器的设置值的情况下，使 SM798 为 ON。作为多 CPU 间通信专用指令的连续执行用互锁信号使用。	U(RUN 后 1 个扫描时）	新增	QnU*14
SD799	多 CPU 间高速总线对应专用指令最大使用块数设置（2 号 CPU 用）	专用指令最大使用块数范围 1～9（默认：2） *设置了 1～9 以外的值时按 9 执行动作。	・指定多 CPU 间高速总线对应专用指令（对象 CPU=4 号 CPU）的最多使用块数。对 4 号 CPU 执行多 CPU 间通信专用指令时，专用指令传输区的空余块数未达到本寄存器的设置值的情况下，使 SM799 为 ON。作为多 CPU 间通信专用指令的连续执行用互锁信号使用。	U(RUN 后 1 个扫描时）	新增	QnU*14

*14：除 Q02UCPU 外的通用型 QCPU。

(7) 调试

附表 4.9 特殊寄存器列表

<table>
<tr><th>号码</th><th>名称</th><th>含义</th><th>解释</th><th>设置方
（设置时间）</th><th>相应的
ACPU
D9□□□</th><th>相应的 CPU</th></tr>
<tr><td>SD806</td><td rowspan="6">状态锁存文件名称</td><td rowspan="6">状态锁存文件名称</td><td rowspan="6">• 以 ASCII 码存储从执行状态锁存的开始的文件名称（带扩展）。
<table><tr><th></th><th>b15 到 b8</th><th>b7 到 b0</th></tr><tr><td>SD806</td><td>第2个字符</td><td>第1个字符</td></tr><tr><td>SD807</td><td>第4个字符</td><td>第3个字符</td></tr><tr><td>SD808</td><td>第6个字符</td><td>第5个字符</td></tr><tr><td>SD809</td><td>第8个字符</td><td>第7个字符</td></tr><tr><td>SD810</td><td>扩展的第1个字符</td><td>2EH(.)</td></tr><tr><td>SD811</td><td>扩展的第3个字符</td><td>扩展的第2个字符</td></tr></table></td><td rowspan="6">S（在执行过程中）</td><td rowspan="6">新增</td><td rowspan="11">QnA</td></tr>
<tr><td>SD807</td></tr>
<tr><td>SD808</td></tr>
<tr><td>SD809</td></tr>
<tr><td>SD810</td></tr>
<tr><td>SD811</td></tr>
<tr><td>SD812</td><td rowspan="5">状态锁存步</td><td rowspan="5">状态锁存步</td><td rowspan="5">• 存储从执行状态锁存的点开始的步号。
<table><tr><td>SD812</td><td>模式*</td></tr><tr><td>SD813</td><td>块号</td></tr><tr><td>SD814</td><td>步号/转移号</td></tr><tr><td>SD815</td><td>顺控步号(L)</td></tr><tr><td>SD816</td><td>顺控步号(H)</td></tr></table>*: 模式数据的内容
<table><tr><td>15</td><td>14</td><td>到</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td><td>←(位号)</td></tr><tr><td>0</td><td>0</td><td>到</td><td>0</td><td>0</td><td>*</td><td>*</td><td>*</td><td></td></tr></table>(未使用)
SFC块指定存在(1)/不存在(0)
SFC步指定存在(1)/不存在(0)
SFC移指定存在(1)/不存在(0)</td><td rowspan="5">S（在执行过程中）</td><td rowspan="5">D9055 格式改变</td></tr>
<tr><td>SD813</td></tr>
<tr><td>SD814</td></tr>
<tr><td>SD815</td></tr>
<tr><td>SD816</td></tr>
</table>

(8) 锁存区域

附表 4.10 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD900	电源中断的驱动器	电源中断的驱动器	• 存储驱动器号，如果在电源掉电过程中有文件正被访问。	S（状态改变）	新增	QnA
SD901	电源掉电过程中的活动文件名	电源掉电过程中的活动文件名	• 以 ASCII 码存储文件名（带扩展），如果在电源掉电过程中有文件正被访问。 SD901：b15 到 b8 = 第2个字符；b7 到 b0 = 第1个字符 SD902：第4个字符；第3个字符 SD903：第6个字符；第5个字符 SD904：第8个字符；第7个字符 SD905：扩展的第1个字符；2EH(.) SD906：扩展的第3个字符；扩展的第2个字符	S（状态改变）	新增	
SD902						
SD903						
SD904						
SD905						
SD906						
SD910	RKEY 输入	RKEY 输入	• 按顺序存储输入的 PU 键码。 SD910：b15 到 b8 = 第2个字符；b7 到 b0 = 第1个字符 SD911：第4个字符；第3个字符 SD912：第6个字符；第5个字符 SD913：第8个字符；第7个字符 SD914：第10个字符；第9个字符 SD915：第12个字符；第11个字符 SD916：第14个字符；第13个字符 SD917：第16个字符；第15个字符 SD918：第18个字符；第17个字符 SD919：第20个字符；第19个字符 SD920：第22个字符；第21个字符 SD921：第24个字符；第23个字符 SD922：第26个字符；第25个字符 SD923：第28个字符；第27个字符 SD924：第30个字符；第29个字符 SD925：第32个字符；第31个字符	S（状态改变）	新增	
SD911						
SD912						
SD913						
SD914						
SD915						
SD916						
SD917						
SD918						
SD919						
SD920						
SD921						
SD922						
SD923						
SD924						
SD925						

(9) 冗余 CPU 信息（主站系统 CPU 信息 *1）

附表 4.11 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD952	从控制系统到待机系统内存复制的历史记录	从控制系统到待机系统内存复制的最新状态	存储最后一次执行的从控制系统到待机系统的内存复制的结束状态。 1) 在从控制系统到待机系统的内存复制正常结束 / 异常结束时，存储和存储到 SD1596 中的值一样的值。 2) 为电源故障做备份，此特殊寄存器保持最后一次执行从控制系统到待机系统内存复制的状态。 3) 由锁存清除操作清除到 0。	S（状态改变）	新增	QnPRH

*1：主站系统 CPU 信息被存储。

(10)A → Q/QnA 转换

在 A → Q/QnA 转换之后，ACPU 特殊寄存器 D9000 ～ D9255 对应于 Q/QnA 特殊寄存器 SD1000 ～ SD1255。

(但是，基本型 QCPU、冗余 CPU 和通用型 QCPU 不支持 A → Q/QnA 转换。)

这些特殊继电器都是由系统设定的，用户程序不能设定它们。

要由用户程序设定数据，修改程序以使用 Q/QnACPU 特殊寄存器。

但是，SD1200 到 SD1255(对应转换前的 D9200 到 9255) 中的一些寄存器不能由用户程序设定，如果在转换之前它们可以由用户程序设定的话。

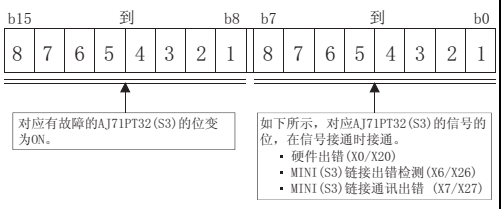
对于 ACPU 特殊寄存器的详细信息，请参考相应的 CPU 的用户手册，和 MELSECNET 或者 MELSECNET/B 数据链接系统参考手册。

备 注

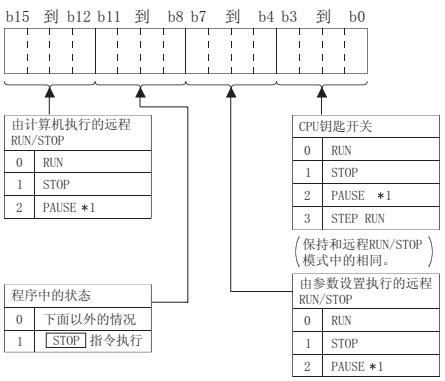
“用于改进的特殊寄存器”一栏的附加解释

1. 对于指定了用于改进的特殊寄存器的软元件号，将其修改为用于 QCPU/QnACPU 的特殊寄存器。
2. 对于指定了[－]的软元件号，可以使用转换之后的特殊寄存器。
3. 指定了[X]的软元件号不能用于 QCPU/QnACPU。

附表 4.11 特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9000</td><td>SD1000</td><td>－</td><td>保险丝熔断</td><td>保险丝熔断的模块的号码</td><td>• 当保险丝熔断的模块被检测到时，检测到的模块的最小号码的第一个 I/O 号被存储到十六进制数中。
（实例：当 Y50 到 6F 输出模块的保险丝熔断时，“50”被存储到十六进制数中）
要使用外围设备监视此号码，以十六进制执行给定的监视操作。
（当 SD1100 到 SD1107 的所有内容都被复位到 0 时，清除。）
• 对远程 I/O 站的输出模块，也执行保险丝熔断检查。</td><td rowspan="2">QnA
Qn(H)
QnPH</td></tr>
<tr><td>D9001</td><td>SD1001</td><td>－</td><td>保险丝熔断</td><td>保险丝熔断模块的数目</td><td>• 存储当发生保险丝熔断时对应设置开关号的模块号或者基板槽号。
<table>
<tr><th colspan="2">用于A0J2的I/O模块</th><th colspan="2">扩展基板</th></tr>
<tr><th>设置开关</th><th>存储的数据</th><th>基板槽号</th><th>存储的数据</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>5</td></tr>
<tr><td>2</td><td>2</td><td>2</td><td>6</td></tr>
<tr><td>3</td><td>3</td><td>3</td><td>7</td></tr>
<tr><td>4</td><td>4</td><td colspan="2" rowspan="4"></td></tr>
<tr><td>5</td><td>5</td></tr>
<tr><td>6</td><td>6</td></tr>
<tr><td>7</td><td>7</td></tr>
</table>
• 对于远程 I/O 站，（模块 I/O 号 /10H)+1 值被存储。</td></tr>
<tr><td>D9002</td><td>SD1002</td><td>－</td><td>I/O 模块验证出错</td><td>验证出错的 I/O 模块号</td><td>• 如果其数据不同于输入的数据的 I/O 模块，在电源变为 ON 时被检测到，在检测到的基板中，最低号码的基板的第一个 I/O 号被存储成十六进制数据。（存储方法和 SD1000 的相同。）
要用外围设备监视此号码，以十六进制执行给定的监视操作。
（清除，当 SD1116 到 SD1123 的所有内容都复位到 0 时。）
• 对远程 I/O 端子的模块也执行 I/O 模块验证检查。</td><td>QnA
Qn(H)
QnPH</td></tr>
<tr><td>D9004</td><td>SD1004</td><td>－</td><td>MINI 连接主站模块出错</td><td>在参数上设定的存储设置状态</td><td>• 在安装的 AJ71PT32(S3) 上检测到的 MINI(S3) 链接的出错状态被存储。

</td><td>QnA</td></tr>
<tr><td rowspan="3">D9005</td><td rowspan="3">SD1005</td><td rowspan="3">－</td><td rowspan="3">AC DOWN 计数器</td><td rowspan="3">AC DOWN 发生的次数</td><td>• 当使用 AC 电源模块时，在发生 20ms 以内的瞬间掉电时加 1。
（此值以 BIN 码存储。）当电源从 OFF 切换到 ON，它被复位。</td><td>QnA
Qn(H)
QnPH</td></tr>
<tr><td>• 当使用 DC 电源模块时，在发生 10ms 以内的瞬间掉电时加 1。
（此值以 BIN 码存储。）当电源从 OFF 切换到 ON，它被复位。</td><td>Qn(H)
QnPH</td></tr>
<tr><td>• 当使用 DC 电源模块时，在发生 1ms 以内的瞬间掉电时加 1。
（此值以 BIN 码存储。）当电源从 OFF 切换到 ON，它被复位。</td><td>QnA</td></tr>
<tr><td>D9008</td><td>SD1008</td><td>SD0</td><td>自检测出错</td><td>自检测出错号</td><td>• 当在自检测结果中发现出错时，出错号以 BIN 码存储。</td><td>QnA
Qn(H)
QnPH</td></tr>
</table>

附表 4.11 特殊寄存器列表

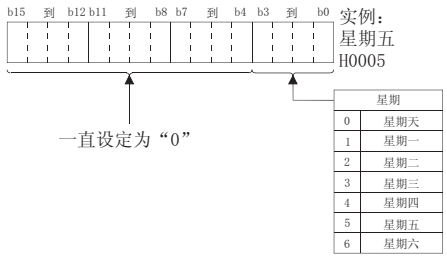
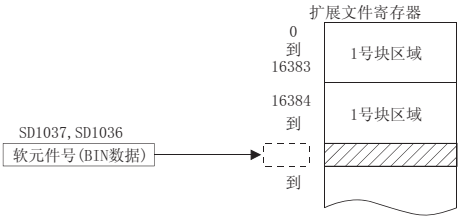
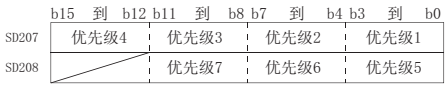
ACPU 特殊转换	转换后的特殊寄存器	用于改进的特殊寄存器	名称	含义	详细信息	相应的 CPU
					• 当 F0 到 2047 中的一个被 OUT F 或者 SET F 变为 ON 时，在变为 ON 的 F 号中最早被检测到的 F 号，被以 BIN 码存储。 • SD62 可以用 RST F 或者 LEDR 指令清除。 如果还其检测到他 F 号，则清除 SD62 将使下一个号码被存储到 SD62。	Qn(H) QnPH Q2AS Q2A
D9009	SD1009	SD62	报警器检测	发生外部故障的 F 号	• 当 F0 到 2047 中的一个由 OUT F 或者 SET F 变为 ON 时，在变为 ON 的 F 号中最早被检测到的 F 号，被以 BIN 码存储 。 • SD62 可以通过执行 RST F 或者 LEDR 指令来清除，或者通过将 CPU 模块前面的 INDICATORRESET 开关移动到 ON 位置来清除。 如果还其检测到他 F 号，则清除 SD62 将使下一个号码被存储到 SD62。	Q3A Q4A Q4AR
D9010	SD1010	×	出错步	发生运行出错的步号。	• 当在执行应用指令过程中发生操作出错时，发生出错的步号以 BIN 码存储。 因此，每当发生操作出错时，SD1010 的内容被更新。	Qn(H) QnPH
D9011	SD1011	×	出错步	发生操作出错的步号	• 当在执行的应用指令过程中发生操作出错时，发生出错的步号以 BIN 码存储。由于步号是在 SM1011 从 OFF 变到 ON 时被存储到 SD1011 中的，SD1011 的数据并不被更新，除非用户程序清除了 SM1011。	
D9014	SD1014	×	I/O 控制模式	I/O 控制模式号	• I/O 控制模式设置以下列号中的一个返回： 0：输入和输出都处于直接模式 1：输入是刷新模式，输出是直接模式 3：输入和输出都处于刷新模式	
D9015	SD1015	SD203	CPU 的运行状态	CPU 的运行状态	• 如下所示的 CPU 的运行状态被存储到 SD203。  *1：当 CPU 模块处于 RUN 模式，并且 SM1040 是关断时，CPU 模块保持在 RUN 模式，如果它被改变到 PAUSE 模式的话。	QnA Qn(H) QnPH

附表 4.11 特殊寄存器列表

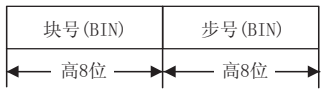
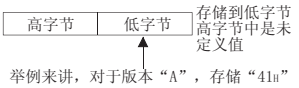
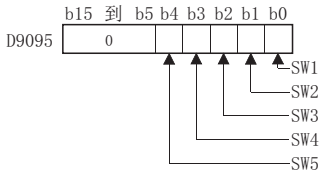
ACPU 特殊转换	转换后的特殊寄存器	用于改进的特殊寄存器	名称	含义	详细信息	相应的 CPU
D9016	SD1016	×	程序号	0：主程序 (ROM) 1：主程序 (RAM) 2：子程序 1(RAM) 3：子程序 2(RAM) 4：子程序 3(RAM) 5：子程序 1(ROM) 6：子程序 2(ROM) 7：子程序 3(ROM) 8：主程序 (E^2PROM) 9：子程序 1(E^2PROM) A：子程序 2(E^2PROM) B：子程序 3(E^2PROM)	• 表示当前哪个顺控程序正在运行。0 到 B 之间的一个值被存储在 BIN 码。	QnA Qn(H) QnPH
D9017	SD1017	SD520	扫描周期时间	最小扫描周期时间 (10ms 为单位)	• 如果扫描周期时间小于 SD520 中的值，则在每个 END 新值被存储。也就是说，扫描周期时间的最小值被以 BIN 码存储到 SD520。	
D9018	SD1018	SD524	扫描周期时间	扫描周期时间 (10ms 为单位)	• 在每个 END 中，扫描周期时间被以 BIN 码存储，并一直被重写。	
D9019	SD1019	SD526	扫描周期时间	最大扫描周期时间 (10ms 为单位)	• 如果扫描周期时间大于 SD526 中的值，则在每个 END 新值被存储。也就是说，扫描周期时间的最大值被以 BIN 码存储到 SD526。	
D9020	SD1020	×	恒定扫描周期	恒定扫描周期时间 (由用户以 10ms 为单位进行设定)	• 设置连续程序启动间隔为 10ms 的倍数。 0 ：无设置 1 到 200 ：设置。程序以 (设定值) × 10 ms 的间隔执行。	
D9021	SD1021	-	扫描周期时间	扫描周期时间 (1ms 为单位)	• 在每个 END 中，扫描周期时间被以 BIN 码存储，并一直被重写。	
D9022	SD1022	SD412	1 秒计数器	以 1s 为单位计数。	• 当 PCCPU 开始运行时，它开始每秒计数 1。 • 从 0 开始计数，一直加到 32767，然后到 -32768，再重新加到 0。计数一直重复此过程。	
D9025	SD1025	-	时钟数据	时钟数据 (年，月)	• 以 BCD 码存储年 (低 2 位数字) 和月。 b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 1987 七月 H8707 年 月	
D9026	SD1026	-	时钟数据	时钟数据 (日，小时)	• 以 BCD 码存储日期和小时。 b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 31日上午10点。 H3110 日期 小时	
D9027	SD1027	-	时钟数据	时钟数据 (分钟，秒)	• 以 BCD 码存储分钟和秒。 b15 到 b12 b11 到 b8 b7 到 b4 b3 到 b0 实例： 35分48秒。 H3548 分钟 秒	

附 附录

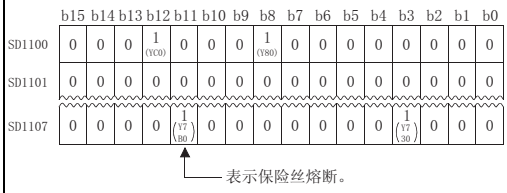
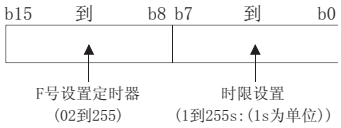
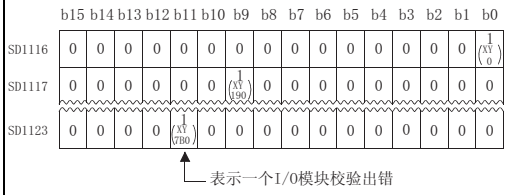
附表 4.11 特殊寄存器列表

ACPU 特殊转换	转换后的特殊寄存器	转换后的特殊寄存器	名称	含义	详细信息	相应的 CPU
D9028	SD1028	-	时钟数据	时钟数据（星期）	• 以 BCD 码存储星期。 	QnA Qn(H) QnPH
D9035	SD1035	SD648	扩展文件寄存器	使用的块号	• 以 BCD 码存储正被使用的扩展文件寄存器的块号。	
D9036	SD1036	×	用于软元件号的指定的扩展文件寄存器	来自扩展文件寄存器的单个软元件被直接访问时的软元件号	• 以 BIN 数据，用 SD1036 和 SD1037 中的 2 个字，为用于直接读和写的扩展文件寄存器指定软元件号。 从 1 号块的 R0 开始，使用连续的号码去指定软元件号。 	
D9037	SD1037	×				
D9038	SD1038	SD207	LED 显示优先级	优先级 1 到 4	• 设置 ERRORLED 的优先级，此 LED 点亮（或者闪烁），以出错代码号表示出错。 • 优先级设置区域的配置如下所示。  • 详细信息，请参考可用 CPU 的用户手册和编程手册（基础篇）。	
D9039	SD1039	SD208		优先级 5 到 7		
D9044	SD1044	×	用于采样跟踪	采样跟踪过程中的步或者时间	• 用外围设备变为 ON/OFF。 当执行 STRA 或者 STRAR 时，存储在 SD1044 中的值被用作采样跟踪条件。 在扫描过程中 --------0 在定时间 ---------- 时间（10ms 为单位） 此值以 BIN 码存储到 SD1044 中。	
D9049	SD1049	×	用于 SFC 的工作区域	扩展文件寄存器的块号	• 以二进制值存储扩展文件寄存器的块号，此寄存器用作用于执行 SFC 程序的工作区域。 • 如果使用了 16K 字节或者更小的空置区域（不能是 1 号扩展文件寄存器），或者如果 SM320 是 OFF，则存储 “0”。	
D9050	SD1050	×	SFC 程序出错号	SFC 程序产生的出错代码	• 以 BIN 码存储 SFC 程序中发生的出错的出错代码。 0 ： 没有出错 80：SFC 程序参数出错 81：SFC 代码出错 82：超过了同时执行的步数 83：块启动出错 84：SFC 程序运行出错	
D9051	SD1051	×	出错块	出错发生的块号	• 以 BIN 码存储 SFC 程序中发生出错的块的块号。 在出错 83 中，起始块号被存储。	
D9052	SD1052	×	出错步	出错发生的步号	• 以 BIN 值存储 SFC 程序中发生出错代码 84 的步号。 • 当出错代码 80、81 或者 82 发生时，存储 “0”。 • 当出错代码 83 发生时，存储块起始步号。	

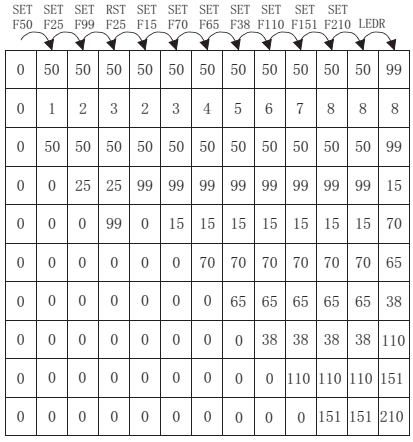
附表 4.11 特殊寄存器列表

ACPU 特殊转换	转换后的特殊寄存器	用于改进的特殊寄存器	名称	含义	详细信息	相应的 CPU
D9053	SD1053	×	出错转移	出错发生处的转移条件号	• 以 BIN 值存储 SFC 程序中发生出错代码 84 的地方的转移条件号。 当出错代码 80、81、82 或者 83 发生，存储 “0”。	QnA Qn(H) QnPH
D9054	SD1054	×	出错顺控程序	出错发生处的顺控程序步号	• 以 BIN 码存储 SFC 程序中发生出错 84 的发送条件和操作输出的顺控程序步号。	
D9055	SD1055	SD812	状态锁存执行步号	状态锁存步	• 当执行状态锁存时，存储步号。 • 如果在主顺控程序中执行状态锁存，以二进制值存储 • 步号。 • 如果在 SFC 程序执行状态锁存，存储块号和步号。 	
D9060	SD1060	SD392	软件版本	内部软件的软件版本	• 以 ASCII 码存储内部系统的软件版本。  注释：初始化系统的软件版本可能和印刷在外壳上的版本信息指示的版本不同。	QnA
D9072	SD1072	×	可编程控制器通讯检查	串行通讯模块数据检查	• 在串行通讯模块的自环路回送测试中，串行通讯模块自动写 / 读数据以进行通讯检查。	QnA Qn(H) QnPH
D9081	SD1081	SD714	通讯请求注册区域中的空余块数	通讯请求注册区域中的空余块数	• 存储到连接到 MELSECNET/MINI-S3 主基板、A2CCPU 或者 A52GCPU 的远程端子模块的通讯请求注册区域中的空余块数。	QnA
D9085	SD1085	×	用于设置时间检查值的寄存器	1 s 到 65535 s	• 为 MELSECNET/10 设置数据链接指令 (ZNRD，ZNWR) 的时间检查时间。 • 设置范围 ： 1s 到 65535s(1 到 65535) • 设置单位：1 s • 默认值：10s(如果设置了 0，则应用默认值 10s)	QnA Qn(H) QnPH
D9090	SD1090	×	特殊功能模块数	特殊功能模块数	• 对于详细信息，请参考每个微型计算机程序软件包的手册。	
D9091	SD1091	×	详细出错代码	自检测详细出错代码	• 存储指令出错原因的详细代码。	
D9094	SD1094	SD251	要更换的 I/O 模块的起始 I/O 号	要更换的 I/O 模块的起始 I/O 号	• 以 BIN 值存储将运行 (电源接通) 中卸载 / 安装的 I/O 模块起始 I/O 号码的头两位数。 实例) 输入模块 X2F0→H2F	
D9095	SD1095	SD200	DIP 开关信息	DIP 开关信息	• 以下列格式存储 CPU 模块的 DIP 开关信息。 0：OFF 1：ON 	Qn(H) QnPH

附表 4.11 特殊寄存器列表

ACPU 特殊转换	转换后的特殊寄存器	用于改进的特殊寄存器	名称	含义	详细信息	相应的 CPU
D9100	SD1100	–	保险丝熔断的模块	以 16 点为单位的位模式，表示保险丝熔断的模块	• 保险丝熔断的输出模块号（以 16 点为单位）以位模式输入。（当执行参数设置时，预设输出模块号。）  • 对远程 I/O 站的输出模块也执行保险丝熔断检查。（如果正常状态被存储，则不执行清除。因此，需要由用户程序执行清除。）	QnA Qn(H) QnPH
D9101	SD1101					
D9102	SD1102					
D9103	SD1103					
D9104	SD1104					
D9105	SD1105					
D9106	SD1106					
D9107	SD1107					
D9108	SD1108	–	步发送看门狗定时器设置	定时器设置值和时间用完时的报警器号	• 设定步转移看门狗定时器的设置值和当看门狗定时器时间用完时将接通的报警器号 (F 号)。  • (通过将 SM1108 到 SM1114 中的任何一个接通，看门狗定时器启动。如果对应定时器的步后面的发送条件在设定的时间内不能建立，则设置的报警器 (F) 将接通。）	
D9109	SD1109					
D9110	SD1110					
D9111	SD1111					
D9112	SD1112					
D9113	SD1113					
D9114	SD1114					
D9116	SD1116	–	I/O 模块验证出错	以 16 点为单位的位模式，表示验证出错的模块。	• 当其数据和电源接通时输入的数据不同的 I/O 模块被检测到时，I/O 模块号（以 16 点为单位）被以位模式输入。（当执行参数设置时，在参数中预设 I/O 模块号。）  • I/O 模块验证检查也可以对远程 I/O 站模块执行。（如果正常状态被存储，则不执行清除。因此，需要由用户程序执行清除。）	
D9117	SD1117					
D9118	SD1118					
D9119	SD1119					
D9120	SD1120					
D9121	SD1121					
D9122	SD1122					
D9123	SD1123					
D9124	SD1124	SD63	报警器检测数量	报警器检测数量	• 当 F0 到 255（对于 AuA 和 AnU 是 F0 到 2047）中的一个由 SET F 变为 ON 时，1 加到 SD63 中的值上。当 RST F 或者 LEDR 指令执行时，从 SD63 中的值减 1。（如果 CPU 模块有 INDICATORRESET 开关，则按下此开关可以执行相同的处理。） • 由 SET F 变为 ON 的数量被以 BIN 码存储到 SD63 中。SD63 的值最大是 8。	

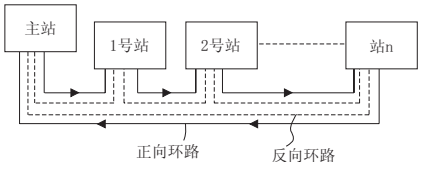
附表 4.11 特殊寄存器列表

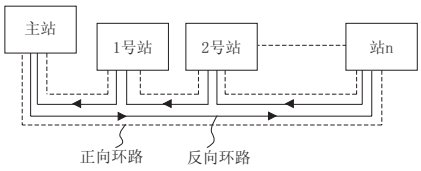
<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9125</td><td>SD1125</td><td>SD64</td><td rowspan="10">报警器检测号</td><td rowspan="10">报警器检测号</td><td rowspan="10">• 当 F0 到 2047 中的一个被 SET F 变为 ON 时，按顺序变为 ON 的报警器号（F 号）被注册到 D9125 到 D9132 中。
• 由 RST F 关断的 F 号被从 D9125 到 D9132 中的某一个中擦除，而存储在被擦除的 F 号之后的 F 号被移位到先前的寄存器中。
通过执行 LEDR 指令，SD64 到 SD71 的内容向前移动一位。（对于 A3N 和 A3HCPU，可以通过使用 CPU 模块前面的 INDICATORRESET 开关执行此操作。）
当检测到 8 个报警器时，即使检测到了第 9 个，也不会被存储到 SD64 到 SD71 中。
 
</td><td rowspan="10">QnA
Qn(H)
QnPH</td></tr>
<tr><td>D9126</td><td>SD1126</td><td>SD65</td></tr>
<tr><td>D9127</td><td>SD1127</td><td>SD66</td></tr>
<tr><td>D9128</td><td>SD1128</td><td>SD67</td></tr>
<tr><td>D9129</td><td>SD1129</td><td>SD68</td></tr>
<tr><td>D9130</td><td>SD1130</td><td>SD69</td></tr>
<tr><td>D9131</td><td>SD1131</td><td>SD70</td></tr>
<tr><td>D9132</td><td>SD1132</td><td>SD71</td></tr>
</table>

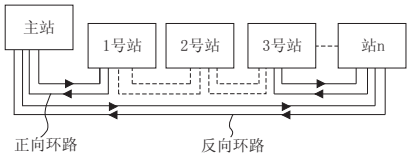
SD62	0	50	50	50	50	50	50	50	50	50	50	50	99
SD63	0	1	2	3	2	3	4	5	6	7	8	8	8
SD64	0	50	50	50	50	50	50	50	50	50	50	50	99
SD65	0	0	25	25	99	99	99	99	99	99	99	99	15
SD66	0	0	0	99	0	15	15	15	15	15	15	15	70
SD67	0	0	0	0	0	0	70	70	70	70	70	70	65
SD68	0	0	0	0	0	0	0	65	65	65	65	65	38
SD69	0	0	0	0	0	0	0	0	38	38	38	38	110
SD70	0	0	0	0	0	0	0	0	0	110	110	110	151
SD71	0	0	0	0	0	0	0	0	0	0	151	151	210

(11)QnA 专用特殊寄存器列表

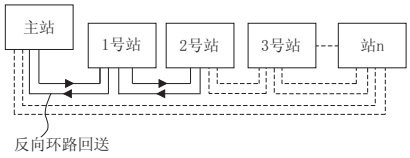
附表 4.12 特殊寄存器列表

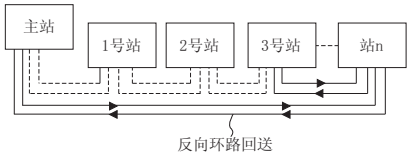
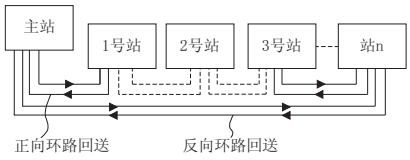
<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9200</td><td>SD1200</td><td>－</td><td>ZNRD(LRDP 用于 ACPU) 处理结果</td><td>0 ： 正常结束
2 ： ZNRD 指令设置故障
3 ： 相关站上的出错
4 ： 相关站 ZNRD 执行禁止</td><td>存储 ZNRD(字软元件读) 指令的执行结果
• ZNRD 指令设置故障 :ZNRD 指令常量，源和 / 或者目标的故障设置。
• 相应站出错 :其中的一个站没有通讯。
• ZNRD 不能在相应站中执行 :指定的站是远程 I/O 站。</td><td rowspan="4">QnA</td></tr>
<tr><td>D9201</td><td>SD1201</td><td>－</td><td>ZNWR(LWTP 用于 ACPU) 处理结果</td><td>0 ： 正常结束
2 ： ZNWR 指令设置故障
3 ： 相关站上的出错
4 ： 相关站 ZNWR 执行禁止</td><td>存储 ZNWR(字软元件读) 指令的执行结果
• ZNWR 指令设置故障 :ZNWR 指令常量，源和 / 或者目标的故障设置。
• 相应站出错 :其中的一个站设有通讯。
• ZNWR 不能在相应站中执行 :指定的站是远程 I/O 站。</td></tr>
<tr><td>D9202</td><td>SD1202</td><td>－</td><td rowspan="2">本地站链接类型</td><td>存储 1 号到 16 号的条件</td><td rowspan="2">存储从站是否对应 MELSECNET 或者 MELSECNETII。
• 对应 MELSECNETII 站的位变为 “1”。
• 对应 MELSECNET 站或者没有连接的位变为 “0”。
<table><tr><th rowspan="2">软元件号</th><th colspan="16">位</th></tr><tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD1202</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1203</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1241</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1242</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>• 如果本地站在运行过程中停止，则停止之前的内容被保持。
SD1224 到 SD1227 和 SD1228 到 SD1231 的内容被执行逻辑或操作。如果相应位是 “0”，则上面的特殊寄存器的相应位变为有效。
• 如果自身（主）站停止，则停止之前的内容也会被保持。</td></tr>
<tr><td>D9203</td><td>SD1203</td><td>－</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9204</td><td>SD1204</td><td>－</td><td>链接状态</td><td>0 ： 正向环路， 在数据链接中
1 ： 反向环路，在数据链接中
2 ： 正向 / 反向实现环路回送
3 ： 只在正向实现环路回送
4 ： 只在反向实现环路回送
5 ： 数据链接禁止</td><td>存储数据链接的当前路径状态。
• 正向环路中的数据链接
 

• 反向环路中的数据链接
 

• 正向 / 反向实现环路回送
 
</td><td></td></tr>
</table>

附表 4.12 特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9204</td><td>SD1204</td><td>－</td><td>链接状态</td><td>0 ： 正向环路， 在数据链接中
1 ： 反向环路， 在数据链接中
2 ： 正向 / 反向实现环路回送
3 ： 只在正向实现环路回送
4 ： 只在反向实现环路回送
5 ： 数据链接禁止</td><td>• 只在正向环路实现环路回送
 

• 只在反向环路实现环路回送
 
</td><td rowspan="11">QnA</td></tr>
<tr><td>D9205</td><td>SD1205</td><td>－</td><td>实现环路回送的站</td><td>实现正向环路回送的站</td><td rowspan="2">存储正在执行环路回送的本地或者远程 I/O 站号
 

在上面的实例中，1 被存储到 D9205，3 到 D9206。
如果数据链接恢复为正常状态（正向环路中的数据链接），D9205 和 D9206 中的值保持为 1 和 3。因此要将它们变为 “0”，使用顺控程序或者执行复位操作。</td></tr>
<tr><td>D9206</td><td>SD1206</td><td>－</td><td>实现环路回送的站</td><td>实现反向环路回送的站</td></tr>
<tr><td>D9210</td><td>SD1210</td><td>－</td><td>重试次数</td><td>存储 1 号到 16 号的条件</td><td>存储由于传送出错而引起的重试次数。
在最大值 “FFFFH” 处，计数停止。
要恢复到值 “0”，执行复位操作。</td></tr>
<tr><td>D9211</td><td>SD1211</td><td>－</td><td>环路选择次数</td><td>存储 1 号到 16 号的条件</td><td>存储环路电缆被切换到反向环路或者环路回送的次数。
在最大值 “FFFFH” 处，计数停止。
要恢复到值 “0”，执行复位操作。</td></tr>
<tr><td>D9212</td><td>SD1212</td><td>－</td><td>本地站运行状态</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储处于 STOP 或者 PAUSE 模式的本地站号。
<table><tr><th rowspan="2">软元件号</th><th colspan="16">位</th></tr><tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD1212</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1213</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1214</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1215</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>
当本地站被切换到 STOP 或者 PAUSE 模式时，寄存器中对应此站号的位变为 “1”。
实例：当站 7 切换到 STOP 模式，SD1212 中的 b6 变为 “1”，当 SD1212 被监视时，它的值是 “(40H)”。</td></tr>
<tr><td>D9213</td><td>SD1213</td><td>－</td><td>本地站运行状态</td><td>存储到 17 号 到 32 号的条件</td></tr>
<tr><td>D9214</td><td>SD1214</td><td>－</td><td>本地站运行状态</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9215</td><td>SD1215</td><td>－</td><td>本地站运行状态</td><td>存储 49 号到 64 号的条件</td></tr>
<tr><td>D9216</td><td>SD1216</td><td>－</td><td>本地站出错检测状态</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储出错的本地站号。
<table><tr><th rowspan="2">软元件号</th><th colspan="16">位</th></tr><tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD1216</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1217</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1218</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1219</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>
如果本地站检测到出错，对应站号的位变为 “1”。
实例：当站 6 和 12 检测到出错，SD1216 中的 b5 和 b11 变为 “1”，当 SD1216 被监视时，它的值是 “2080(820H)”。</td></tr>
<tr><td>D9217</td><td>SD1217</td><td>－</td><td>本地站出错检测状态</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9218</td><td>SD1218</td><td>－</td><td>本地站出错检测状态</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9219</td><td>SD1219</td><td>－</td><td>本地站出错检测状态</td><td>存储 49 号到 64 号的条件</td></tr>
</table>

附

附录

附表 4.12 特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9220</td><td>SD1220</td><td>-</td><td>本地站参数不一致；远程 I/0 站 I/0 分配出错</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储包含不匹配的参数的本地站号或者设置了不正确 I/0 分配的远程站号。
<table><tr><td>软元件号</td><td colspan="16">位</td></tr><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD1220</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1221</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1222</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1223</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>如果作为三级系统的主站的本地站检测参数出错或者远程 I/0 站 I/0 分配异常，对应本地站或者远程 I/0 站的站号的软元件号的位变为“1”。
实例：当本地站 5 和远程 I/0 站 14 检测到出错，SD1220 中的 b4 和 b13 变为“1”，当 SD1220 被监视时，它的值是“8208(2010H)”。</td><td rowspan="10">QnA</td></tr>
<tr><td>D9221</td><td>SD1221</td><td>-</td><td>本地站参数不一致；远程 I/0 站 I/0 分配出错</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9222</td><td>SD1222</td><td>-</td><td>本地站参数不一致；远程 I/0 站 I/0 分配出错</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9223</td><td>SD1223</td><td>-</td><td>本地站参数不一致；远程 I/0 站 I/0 分配出错</td><td>存储 49 号到 64 号的条件</td></tr>
<tr><td>D9224</td><td>SD1224</td><td>-</td><td>本地站和远程 I/0 站初始化通讯进行中</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储正在和它们的相关主站通讯初始化数据的本地或者远程站的站号。
<table><tr><td>软元件号</td><td colspan="16">位</td></tr><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD1224</td><td>L/R 16</td><td>L/R 15</td><td>L/R 14</td><td>L/R 13</td><td>L/R 12</td><td>L/R 11</td><td>L/R 10</td><td>L/R 9</td><td>L/R 8</td><td>L/R 7</td><td>L/R 6</td><td>L/R 5</td><td>L/R 4</td><td>L/R 3</td><td>L/R 2</td><td>L/R 1</td></tr><tr><td>SD1225</td><td>L/R 32</td><td>L/R 31</td><td>L/R 30</td><td>L/R 29</td><td>L/R 28</td><td>L/R 27</td><td>L/R 26</td><td>L/R 25</td><td>L/R 24</td><td>L/R 23</td><td>L/R 22</td><td>L/R 21</td><td>L/R 20</td><td>L/R 19</td><td>L/R 18</td><td>L/R 17</td></tr><tr><td>SD1226</td><td>L/R 48</td><td>L/R 47</td><td>L/R 46</td><td>L/R 45</td><td>L/R 44</td><td>L/R 43</td><td>L/R 42</td><td>L/R 41</td><td>L/R 40</td><td>L/R 39</td><td>L/R 38</td><td>L/R 37</td><td>L/R 36</td><td>L/R 35</td><td>L/R 34</td><td>L/R 33</td></tr><tr><td>SD1227</td><td>L/R 64</td><td>L/R 63</td><td>L/R 62</td><td>L/R 61</td><td>L/R 60</td><td>L/R 59</td><td>L/R 58</td><td>L/R 57</td><td>L/R 56</td><td>L/R 55</td><td>L/R 54</td><td>L/R 53</td><td>L/R 52</td><td>L/R 51</td><td>L/R 50</td><td>L/R 49</td></tr></table>和当前正在通讯初始化设置的站号对应的位变为“1”。
实例：当站 23 和 45 正在通讯时，SD1225 的 b6 和 SD1226 的 b12 变为“1”，当 SD1225 被监视时，它的值是“64(40H)”，当 SD1226 被监视时，它的值是“4096(1000H)”。</td></tr>
<tr><td>D9225</td><td>SD1225</td><td>-</td><td>本地站和远程 I/0 站初始化通讯进行中</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9226</td><td>SD1226</td><td>-</td><td>本地站和远程 I/0 站初始化通讯进行中</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9227</td><td>SD1227</td><td>-</td><td>本地站和远程 I/0 站初始化通讯进行中</td><td>存储 49 号到 64 号的条件</td></tr>
<tr><td>D9228</td><td>SD1228</td><td>-</td><td>本地站和远程 I/0 站出错</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储出错的本地或者远程站号。
<table><tr><td>软元件号</td><td colspan="16">位</td></tr><tr><td></td><td>b15</td><td>b14</td><td>b13</td><td>b12</td><td>b11</td><td>b10</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td></tr><tr><td>SD1228</td><td>L/R 16</td><td>L/R 15</td><td>L/R 14</td><td>L/R 13</td><td>L/R 12</td><td>L/R 11</td><td>L/R 10</td><td>L/R 9</td><td>L/R 8</td><td>L/R 7</td><td>L/R 6</td><td>L/R 5</td><td>L/R 4</td><td>L/R 3</td><td>L/R 2</td><td>L/R 1</td></tr><tr><td>SD1229</td><td>L/R 32</td><td>L/R 31</td><td>L/R 30</td><td>L/R 29</td><td>L/R 28</td><td>L/R 27</td><td>L/R 26</td><td>L/R 25</td><td>L/R 24</td><td>L/R 23</td><td>L/R 22</td><td>L/R 21</td><td>L/R 20</td><td>L/R 19</td><td>L/R 18</td><td>L/R 17</td></tr><tr><td>SD1230</td><td>L/R 48</td><td>L/R 47</td><td>L/R 46</td><td>L/R 45</td><td>L/R 44</td><td>L/R 43</td><td>L/R 42</td><td>L/R 41</td><td>L/R 40</td><td>L/R 39</td><td>L/R 38</td><td>L/R 37</td><td>L/R 36</td><td>L/R 35</td><td>L/R 34</td><td>L/R 33</td></tr><tr><td>SD1231</td><td>L/R 64</td><td>L/R 63</td><td>L/R 62</td><td>L/R 61</td><td>L/R 60</td><td>L/R 59</td><td>L/R 58</td><td>L/R 57</td><td>L/R 56</td><td>L/R 55</td><td>L/R 54</td><td>L/R 53</td><td>L/R 52</td><td>L/R 51</td><td>L/R 50</td><td>L/R 49</td></tr></table>对应出错站号的位变为“1”。
实例：当本地站 3 和远程 I/0 站 14 出错时，SD1228 的 b2 和 b13 变为“1”，当 SD1228 被监视时，它的值是“8196(2004H)”。</td></tr>
<tr><td>D9229</td><td>SD1229</td><td>-</td><td>本地站和远程 I/0 站出错</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9230</td><td>SD1230</td><td>-</td><td>本地站和远程 I/0 站出错</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9231</td><td>SD1231</td><td>-</td><td>本地站和远程 I/0 站出错</td><td>存储 49 号到 64 号的条件</td></tr>
</table>

附表 4.12 特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9232</td><td>SD1232</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 1 号到 8 号的条件</td><td rowspan="8">存储发生正向或者反向环路出错的本地或者远程站号
<table><tr><th>软元件号</th><th>位</th></tr><tr><td></td><td>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</td></tr><tr><td>SD1232</td><td>R F R F R F R F R F R F R F R F R F
L/R8 L/R7 L/R6 L/R5 L/R4 L/R3 L/R2 L/R1</td></tr><tr><td>SD1233</td><td>R F R F R F R F R F R F R F R F R F
L/R16 L/R15 L/R14 L/R13 L/R12 L/R11 L/R10 L/R9</td></tr><tr><td>SD1234</td><td>R F R F R F R F R F R F R F R F R F
L/R24 L/R23 L/R22 L/R21 L/R20 L/R19 L/R18 L/R17</td></tr><tr><td>SD1235</td><td>R F R F R F R F R F R F R F R F R F
L/R32 L/R31 L/R30 L/R29 L/R28 L/R27 L/R26 L/R25</td></tr><tr><td>SD1236</td><td>R F R F R F R F R F R F R F R F R F
L/R40 L/R39 L/R38 L/R37 L/R36 L/R35 L/R34 L/R33</td></tr><tr><td>SD1237</td><td>R F R F R F R F R F R F R F R F R F
L/R48 L/R47 L/R46 L/R45 L/R44 L/R43 L/R42 L/R41</td></tr><tr><td>SD1238</td><td>R F R F R F R F R F R F R F R F R F
L/R56 L/R55 L/R54 L/R53 L/R52 L/R51 L/R50 L/R49</td></tr><tr><td>SD1239</td><td>R F R F R F R F R F R F R F R F R F
L/R64 L/R63 L/R62 L/R61 L/R60 L/R59 L/R58 L/R57</td></tr></table>存储发生正向或者反向环路出错的本地或者远程站号环路电缆。软元件号的位对应有正向环路电缆或者反向环路电缆出错的本地站或者远程 I/O 站的站号。
实例：当站 5 的正向环路电缆出错时，SD1232 的 b8 变为 “1”，当 SD1232 被监视时，它的值是 “256(100H)”。</td><td rowspan="9">QnA</td></tr>
<tr><td>D9233</td><td>SD1233</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 9 号到 16 号的条件</td></tr>
<tr><td>D9234</td><td>SD1234</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 17 号到 24 号的条件</td></tr>
<tr><td>D9235</td><td>SD1235</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 25 号到 32 号的条件</td></tr>
<tr><td>D9236</td><td>SD1236</td><td>–</td><td>存储 25 号到 32 号的条件</td><td>存储 25 号到 32 号的条件</td></tr>
<tr><td>D9237</td><td>SD1237</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 41 号到 48 号的条件</td></tr>
<tr><td>D9238</td><td>SD1238</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 49 号到 56 号的条件</td></tr>
<tr><td>D9239</td><td>SD1239</td><td>–</td><td>本地站和远程 I/0 站环路出错</td><td>存储 57 号到 64 号的条件</td></tr>
<tr><td>D9240</td><td>SD1240</td><td>–</td><td>通讯出错检测到的次数</td><td>存储接收到的出错的累积值</td><td>存储下列传送出错检测到的次数：
CRC，OVER，AB. IF
计数最大到 FFFFH.
要恢复值到 “0”，执行复位操作。</td></tr>
<tr><td>D9241</td><td>SD1241</td><td rowspan="2">–</td><td rowspan="2">本地站链接类型</td><td>存储 33 号到 48 号的条件</td><td rowspan="2">存储从站是否对应 MELSECNET 或者 MELSECNET II。
• 对应 MELSECNETII 站的位变为 “1。”
• 对应 MELSECNET 站或者未连接的位变为 “0。”
<table><tr><th>软元件号</th><th>位</th></tr><tr><td></td><td>b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0</td></tr><tr><td>SD1202</td><td>L16 L15 L14 L13 L12 L11 L10 L9 L8 L7 L6 L5 L4 L3 L2 L1</td></tr><tr><td>SD1203</td><td>L32 L31 L30 L29 L28 L27 L26 L25 L24 L23 L22 L21 L20 L19 L18 L17</td></tr><tr><td>SD1241</td><td>L48 L47 L46 L45 L44 L43 L42 L41 L40 L39 L38 L37 L36 L35 L34 L33</td></tr><tr><td>SD1242</td><td>L64 L63 L62 L61 L60 L59 L58 L57 L56 L55 L54 L53 L52 L51 L50 L49</td></tr></table>• 如果本地站在运行过程中停止，停止前的内容不保持。SD1224 到 SD1227 和 SD1228 到 SD1231 的内容被执行逻辑或操作。
如果相应位是 “0”，上面的特殊寄存器的相应位变为有效。
• 如果自身（主）站停止，停止前的内容也被保持。</td></tr>
<tr><td>D9242</td><td>SD1242</td><td>存储站号 (0 到 64)</td></tr>
<tr><td>D9243</td><td>SD1243</td><td>–</td><td>用于本站的站号信息</td><td>存储站号 (0 到 64)</td><td>允许本地站去确认它的自身站号</td></tr>
<tr><td>D9244</td><td>SD1244</td><td>–</td><td>存储从站数</td><td>存储从站数</td><td>表示一个环路中的从站数。</td></tr>
<tr><td>D9245</td><td>SD1245</td><td>–</td><td>接受出错检测到的次数</td><td>接受出错检测到的次数</td><td>存储下列传送出错检测到的次数：CRC，OVER，AB. IF
计数最大到 FFFFH。要恢复此值到 “0”，执行复位操作。</td></tr>
</table>

附

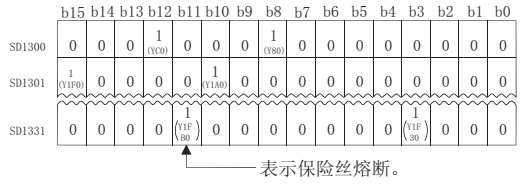
附录

附表 4.12 特殊寄存器列表

<table>
<tr><th>ACPU 特殊转换</th><th>转换后的特殊寄存器</th><th>用于改进的特殊寄存器</th><th>名称</th><th>含义</th><th>详细信息</th><th>相应的 CPU</th></tr>
<tr><td>D9248</td><td>SD1248</td><td>－</td><td>本地站运算状态</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储处于 STOP 或者 PAUSE 模式的本地站号。
<table><tr><th rowspan="2">软元件号</th><th colspan="16">位</th></tr><tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD1248</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1249</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1250</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1251</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>对应处于 STOP 或者 PAUSE 模式的站号的位变为 “1”。
实例：当本地站 7 和 15 处于 STOP 模式，SD1248 的 b6 和 b14 变为 “1”，当 SD1248 被监视时，它的值是 “16448(4040H)”。</td><td rowspan="4">QnA</td></tr>
<tr><td>D9249</td><td>SD1249</td><td>－</td><td>本地站运算状态</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9250</td><td>SD1250</td><td>－</td><td>本地站运算状态</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9251</td><td>SD1251</td><td>－</td><td>本地站运算状态</td><td>存储 49 号到 64 号的条件</td></tr>
<tr><td>D9252</td><td>SD1252</td><td>－</td><td>本地站运算状态</td><td>存储 1 号到 16 号的条件</td><td rowspan="4">存储出错的本地或者远程站号。
<table><tr><th rowspan="2">软元件号</th><th colspan="16">位</th></tr><tr><th>b15</th><th>b14</th><th>b13</th><th>b12</th><th>b11</th><th>b10</th><th>b9</th><th>b8</th><th>b7</th><th>b6</th><th>b5</th><th>b4</th><th>b3</th><th>b2</th><th>b1</th><th>b0</th></tr><tr><td>SD1252</td><td>L16</td><td>L15</td><td>L14</td><td>L13</td><td>L12</td><td>L11</td><td>L10</td><td>L9</td><td>L8</td><td>L7</td><td>L6</td><td>L5</td><td>L4</td><td>L3</td><td>L2</td><td>L1</td></tr><tr><td>SD1253</td><td>L32</td><td>L31</td><td>L30</td><td>L29</td><td>L28</td><td>L27</td><td>L26</td><td>L25</td><td>L24</td><td>L23</td><td>L22</td><td>L21</td><td>L20</td><td>L19</td><td>L18</td><td>L17</td></tr><tr><td>SD1254</td><td>L48</td><td>L47</td><td>L46</td><td>L45</td><td>L44</td><td>L43</td><td>L42</td><td>L41</td><td>L40</td><td>L39</td><td>L38</td><td>L37</td><td>L36</td><td>L35</td><td>L34</td><td>L33</td></tr><tr><td>SD1255</td><td>L64</td><td>L63</td><td>L62</td><td>L61</td><td>L60</td><td>L59</td><td>L58</td><td>L57</td><td>L56</td><td>L55</td><td>L54</td><td>L53</td><td>L52</td><td>L51</td><td>L50</td><td>L49</td></tr></table>对应出错站号的位变为 “1”。
实例：当本地站 12 出错时，SD1252 的 b11 变为 “1”，当 SD1252 被监视时，它的值是 “2048(800H)”。</td></tr>
<tr><td>D9253</td><td>SD1253</td><td>－</td><td>本地站异常状态</td><td>存储 17 号到 32 号的条件</td></tr>
<tr><td>D9254</td><td>SD1254</td><td>－</td><td>本地站异常状态</td><td>存储 33 号到 48 号的条件</td></tr>
<tr><td>D9255</td><td>SD1255</td><td>－</td><td>本地站</td><td>存储 49 号到 64 号的条件</td></tr>
</table>

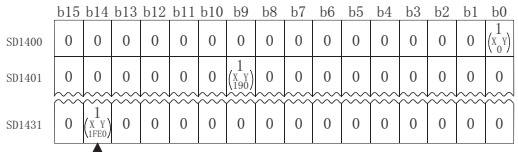
(12)保险丝熔断的模块

附表 4.13 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1300	保险丝熔断的模块	以 16 点为单位的位模式，表示保险丝熔断的模块 0 ： 无保险丝熔断 1 ： 保险丝熔断存在	• 保险丝熔断的输出模块的号码以位模式（以 16 点为单位）输入。 （如果已经用参数设置了模块号，参数设置号被存储。） • 也检测远程站的输出模块的保险丝熔断条件  • 即使保险丝熔断被更换为新的也不清除。 此标志由出错复位操作清除。	S（出错）	D9100	QnA Qn(H) QnPH QnPRH QnU Rem
SD1301					D9101	
SD1302					D9102	
SD1303					D9103	
SD1304					D9104	
SD1305					D9105	
SD1306					D9106	
SD1307					D9107	
SD1308					新增	
SD1309 到 SD1330					新增	
SD1331					新增	

(13)I/O 模块验证

附表 4.14 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1400	I/O 模块验证错误	以 16 点为单位的位模式，表示有验证出错的模块。 0 ： 无 I/O 验证出错 1 ： 有 I/O 验证出错	• 当 I/O 模块信息和系统上电时注册的信息不同的 I/O 模块被检测到时，这些 I/O 模块的号码被以位模式输入。 （如果 I/O 号码是由参数设定的，则参数设定的号码被存储。） • 还检测 I/O 模块信息。  • 即使熔断的保险丝被更换为一个新的，也不清除。 此标志由出错复位操作清除。	S（出错）	D9116	QnA Qn(H) QnPH QnPRH QnU Rem
SD1401					D9117	
SD1402					D9118	
SD1403					D9119	
SD1404					D9120	
SD1405					D9121	
SD1406					D9122	
SD1407					D9123	
SD1408					新增	
SD1409 到 SD1430					新增	
SD1431					新增	

(14)过程控制指令

附表 4.15 特殊寄存器列表

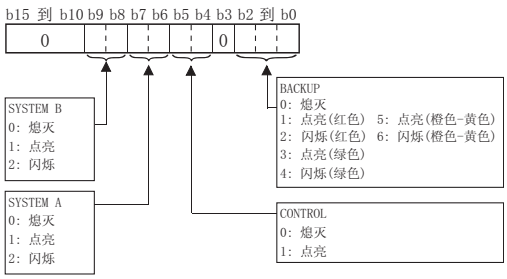
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1500 SD1501	基本周期	基本周期时间	• 以浮点数设置用于过程控制指令的基本周期（1 秒为单位） 浮点数 = SD1501 | SD1500	U	新增	Q4AR QnPH
SD1502	过程控制指令详细出错代码	过程控制指令详细出错代码	• 给出了发生在过程控制指令中的出错的详细出错内容	S（出错）	新增	
SD1503	过程控制指令产生出错的位置	过程控制指令产生出错的位置	• 给出了发生在过程控制指令中的出错处理块	S（出错）	新增	
SD1506 SD1507	虚拟软元件	虚拟软元件	• 用于使用过程控制指令去指定虚拟软元件	U	新增	QnPH QnPRH

(15)用于冗余系统（主站系统 CPU 信息 *1）

SD1510 到 SD1599 只对冗余系统有效。

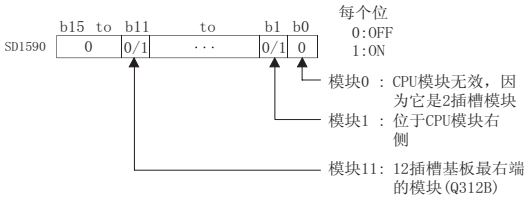
对于单独系统，它们都被设定为 0。.

附表 4.16 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1512	在 CPU 模块启动过程中的运行模式	热启动切换电源输出时间	• 给出了当 CPU 模块启动时，系统在运行模式中自动从热启动切换到初始化启动的电源输出时间 (S)。	S（初始化）	新增	Q4AR
SD1585	冗余系统 LED 状态	4 种 LED 状态 • BACKUP • CONTROL • SYSTEM A • SYSTEM B	用于 BACKUP、CONTRL、SYSTEMA、SYSTEMB 的 LED 状态以下列格式存储：  	S（状态改变）	新增	QnPRH
SD1588	系统切换的原因	发生在本站的系统切换的原因	存储本站系统上系统切换的原因。 下列对应系统切换方法的值被存储： 初始化为 0，当电源是被关断，然后接通或者 RESET 开关被设定到 RESET 位置，然后回到中间位置。 0：初始值（控制系统没有被切换） 1：电源关断、复位、H/W 故障，WDT 出错 2：CPU 停止出错（不包括 WDT） 3：来自网络模块的系统切换请求 16：系统切换专用指令 17：来自 GX Developer 的系统切换请求	S（当条件发生时）	○	

*1：主站 CPU 模块的信息被存储。

附表 4.16 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1589	系统切换故障条件的原因	系统切换故障的原因	• 存储系统切换故障的原因。 0：系统切换正常（默认） 1：热备电缆没有连接，热备电缆出错，FPGA 电路故障。 2：H/W 故障，电源关断，复位，待机系统的 WDT 出错 3：H/W 故障，电源关断，复位，待机系统的 WDT 出错 4：跟踪数据发送初始化 5：通讯超时 6：待机系统的严重出错（不包括 WDT 出错） 7：两个系统之间有差异 （只有后备模式能检测到） 8：在从控制系统到待机系统的内存复制过程中 9：在运行中程序更改过程中 10：待机系统上，处于智能功能模块故障的检测过程中 11：系统切换正在执行 • 当本站系统电源上电时，复位到 “0”。 • 只要系统切换成功，复位到 “0”。	S（当系统被切换时）	○	QnPRH
SD1590	切换请求网络号	请求源网络号	• 当 SM1590 变为 ON 时，存储请求源工作网络号。	S（出错）	新增	Q4AR
	请求系统切换的网络模块起始地址	请求系统切换的网络模块起始地址	• 存储发起系统切换请求的网络模块的起始地址。 • 在网络出错被用户复位之后，自动由系统关断。  • 请参考 SD1690，它存储了其他系统上相应网络模块的起始地址。	S（出错 / 状态改变）	新增	QnPRH
SD1595	内存复制目标 I/O 号码	内存复制目标 I/O 号码	• 在 SM1595 从 OFF → ON 之前存储内存复制目标 I/O 号（待机系统 CPU 模块：3D1H）。	U	新增	
SD1596	内存复制状态	内存复制状态	• 存储内存复制功能的执行结果。 0：内存复制成功结束 4241H：待机系统电源关断 4242H：热备电缆断开或者损坏 4247H：内存复制功能正在执行 4248H：不支持内存复制目标 I/O 号码	S（状态改变）	新增	

(16)用于冗余系统（其他系统 CPU 信息 *1）

SD1600 到 SD1659 只对冗余系统的备份模式有效，当处于独立模式时刷新不能执行。SD1651 到 SD1699 对备用模式和独立模式都有效。当处于单独系统中时，SD1600 到 SD1699 都为 0。

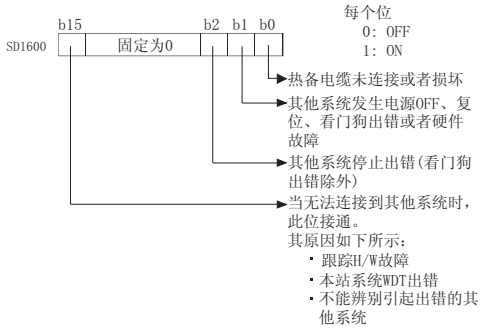
附表 4.17 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU SD□ □*2	相应的 CPU
SD1600	检测出错	检测出错号	• 以 BIN 码存储在其他系统 CPU 模块检测过程中发生的出错的出错号。 • 存储最新发生的出错。	S（每次 END）	SD0	Q4AR
SD1601 SD1602 SD1603	检测出错发生时间	检测出错发生时间	• SD1600 存储更新的日期和时间。 • 存储 BCD 码的两位数中的一个。 • 对于存储状态参考 SD1 到 SD3。 （SD1→SD1601，SD2→SD1602，SD3→SD1603）	S（每次 END）	SD1 到 SD3	
SD1604	出错信息分类	出错信息分类	• 存储出错注释信息 / 个别信息分类代码。 • 对于存储状态参考 SD4。	S（每次 END）	SD4	
SD1605 SD1606 SD1607 SD1608 SD1609 SD1610 SD1611 SD1612 SD1613 SD1614 SD1615	出错公共信息	出错公共信息	• 存储出错代码的公共信息。 • 对于存储状态参考 SD5 到 SD15。 （SD5→SD1605，SD6→SD1606，SD7→SD1607，SD8→SD1608，SD9→SD1609，SD10→SD1610，SD11→SD1611，SD12→SD1612，SD13→SD1613，SD14→SD1614，SD15→SD1615）	S（每次 END）	SD5 到 SD15	
SD1616 SD1617 SD1618 SD1619 SD1620 SD1621 SD1622 SD1623 SD1624 SD1625 SD1626	出错个别信息	出错个别信息	• 存储出错代码的个别信息。 • 对于存储状态参考 SD16 到 SD26。 （SD16→SD1616，SD17→SD1617，SD18→SD1618，SD19→SD1619，SD20→SD1620，SD21→SD1621，SD22→SD1622，SD23→SD1623，SD24→SD1624，SD25→SD1625，SD26→SD1626）	S（每次 END）	SD16 到 SD26	
SD1650	开关状态	CPU 模块开关状态	• 存储 CPU 模块的开关状态。 • 对于存储状态参考 SD200。 （SD1650→SD200）	S（每次 END）	SD200	
SD1651	LED 状态	CPU 模块的 LED 状态	• 存储 CPU 模块的 LED 状态。 • 熄灭为 0，点亮为 1，闪烁为 2。 • 对于存储状态参考 SD201。 （SD1651→SD201）	S（每次 END）	SD201	
SD1653	CPU 模块运行状态	CPU 模块运行状态	• 存储 CPU 模块的运行状态。 • 对于存储状态参考 SD203。 （SD1653→SD203）	S（每次 END）	SD203	

*1 ： 存储其他系统 CPU 模块的检测信息和系统信息。

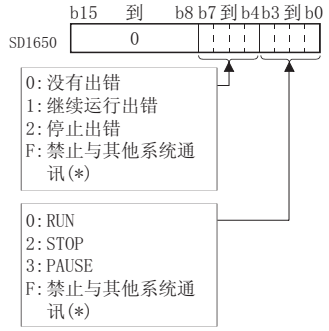
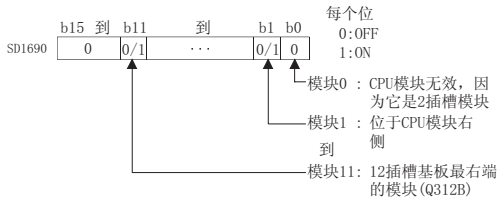
*2 ： 表示用于主站系统 CPU 模块的特殊寄存器 (SD□ □)。

附表 4.17 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU SD□□*2	相应的 CPU
SD1600	系统出错信息	系统出错信息	• 如果对冗余系统的出错检查检测到出错，下面所示的相应位变为 ON。当出错被清除之后，此位变为 OFF。  • 如果 b0、b1、b2 和 b15 中如何一个是 ON，则其他位都为 OFF。 • 在调试模式中，b0、b1、b2 和 b15 都为 OFF。	S（每个 END）	-	QnPRH
SD1601	系统切换结果	系统切换结果	存储引起系统切换的原因。 • 当系统切换发生时，将系统切换的原因存储到两个系统的 SD1601 中。 • 在电源从 OFF 到 ON/ 从复位到重新置位时，初始化到 0。 • 下列给出的值被存储到此寄存器中。 0：初始化值（系统切换还未发生） 1：电源关断，复位，H/W 故障，WDT 出错，(*) 2：CPU 停止出错 (WDT 除外） 3：网络模块发出的系统切换请求 16：系统切换专用指令 17：来自 GX Developer 的系统切换请求 *：当系统切换是由控制系统的电源 OFF/ 复位引起时，“1”不被存储到新待机系统的 SD1601。	S（当系统被切换时）		
SD1602	系统切换专用指令参数	系统切换专用指令参数	• 存储用于系统切换专用指令 SP.CONTSW 的参数。（用于 SP.CONTSW 的参数 (SD1602) 在两个系统 A 和 B 中都存储。） • 只有当 “16”被存储到 SD1601 中时，SD1602 才有效。 • 只要系统切换指令 SP.CONTSW 被激活，SD1602 就被更新。	S（当系统被切换时）		
SD1610	其他系统检测出错	检测出错代码	• 出错值以 BIN 码存储。 • 存储其他系统 CPU 模块的 SD0	S（每个 END）	SD0	
SD1611 SD1612 SD1613	其他系统检测出错发生时间	检测出错发生时间	• 存储和存储在 SD1610 中的出错代码相对应的检测出错发生时的日期和时间。 • 数据格式和 SD1 到 SD3 相同。 • 也存储此值到 SD1 到 SD3。	S（每个 END）	SD1 到 SD3	
SD1614	其他系统出错信息类别	出错信息类别代码	• 存储对应出错注释信息 / 个别信息代码的类别代码。 • 数据格式和 SD4 的相同。 • 也存储此值到 SD4 中。	S（每个 END）	SD4	
SD1615 到 SD1625	其他系统出错公共信息	出错公共信息	• 存储和存储在此系统 CPU 中的出错代码的相对应的公共信息。 • 数据格式和 SD5 到 SD15 的相同。 • 也存储此值到 SD5 到 SD15 中。	S（每个 END）	SD5 到 SD15	
SD1626 到 SD1636	其他系统出错个别信息	出错个别信息	• 存储和存储在此系统 CPU 中的出错代码相对应的个别信息。 • 数据格式和 SD16 到 SD26 的相同。 • 也存储此值到 SD16 到 SD26 中。	S（每个 END）	SD16 到 SD26	
SD1649	待机系统上出错清除	要被清除的出错的出错代码	• 存储通过清除待机系统出错而被清除的出错的出错代码。 • 存储要被清除的出错的出错代码到此寄存器，将 SM1649 从 OFF 变到 ON 以清除待机系统出错。 • 当存储到此寄存器时，出错代码的最低位 (1 位置）中的值被忽略。（通过存储 4100 到此寄存器以及复位出错，出错 4100 到 4109 可以被清除。）	S（每个 END）	-	

*2：表示用于主站系统 CPU 模块的特殊寄存器 (SD□□)。

附表 4.17 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU SD□ □*2	相应的 CPU
SD1650	其他系统运行信息	其他系统运行信息	以下列格式存储另一个系统 CPU 模块的运行信息。 当发生通讯出错，或者当处于调试模式时，存储 “00FFH”。  注释：通讯出错是由下列原因引起： • 当电源被关断，或者当其他系统被复位。 • 在系统或者 B 中发生 H/W 出错。 • WDT 出错发生。 • 热备电缆未连接。 • 热备电缆断开或者损坏。	S（每个 END）	–	QnPRH
SD1690	请求在主站（控制）系统上进行系统切换的网络模块的起始地址	请求在主站（控制）系统上进行系统切换的网络模块的起始地址	• 存储发出系统切换请求的网络模块的起始地址，使用下列格式。 • 在网络出错被用户复位之后自动由系统关断。  • 参考存储相应的主站系统上网络模块的起始地址的 SD1590。	S（每个 END）		

*2 : 表示用于主站系统 CPU 的特殊寄存器 (SD□ □)。

(17)用于冗余系统（跟踪信息）

SD1700 到 SD1779 只对冗余系统有效。
对于单独系统，这些寄存器都为 0。

附表 4.18 特殊寄存器列表

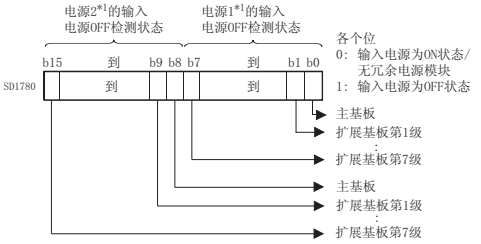
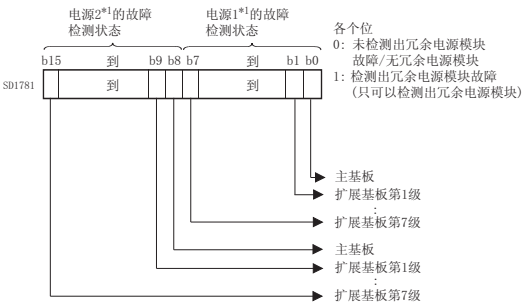
号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□	相应的 CPU
SD1700	跟踪出错检测计数	跟踪出错检测计数	• 将跟踪出错检测计数值 +1。	S（出错）		Q4AR QnPRH
SD1710	运行中程序改变的等待时间（待机系统）	运行中程序改变的等待时间（待机系统）	• 以秒为单位设定，从用于冗余功能的运行中程序更改完成控制系统 CPU 模块的运行中程序更改开始，到待机系统 CPU 模块的运行中程序更改启动为止，待机系统 CPU 模块的等待时间。 • 如果控制系统 CPU 模块的运行中程序更换结束之后，在预设的时间以内没有运行中程序更改请求发送到待机系统 CPU 模块，则两个系统 CPU 模块认为这是用于冗余的运行中程序更改故障。在这种情形中，系统 CPU 模块恢复在运行中程序改变过程中被挂起的，系统 A&B 之间的一致性检查。而且，控制系统 CPU 模块被设定为接收新的用于冗余的运行中程序更改请求。 • 当两个系统上电时，将 90 秒设定到 SD1710 中，作为默认值。 • 设定此值在 90 到 3600 秒范围以内。当设置是 0 到 89 秒时，认为 90 秒用于操作。如果设置超出允许的范围，则认为 0 到 3600 秒以外的时间用于操作。 • 在用于冗余的多个块的运行中写入冗余跟踪的和批量文件的运行中写入冗余跟踪的过程中，依照 SD1710 中的设置，检查用于待机系统 CPU 模块的运行中程序更改启动的等待时间。	U/S（初始化）	新增	QnPRH

*2 ： 表示用于本站系统 CPU 的特殊寄存器 (SD□□)。

(18)冗余电源模块信息

SD1780 到 SD1789 只对冗余电源系统有效。对于单个电源系统，这些位都为 0。

附表 4.19 特殊寄存器列表

号码	名称	含义	解释	设置方（设置时间）	相应的 ACPU D9□□□	相应的 CPU
SD1780	电源 OFF 检测状态	电源 OFF 检测状态	• 以下列位模式存储输入电源 OFF 状态时的冗余电源模块（Q64RP）的状态。 • 当主基板不是冗余电源主基板（Q38RB）时，本寄存器存储 0。  • 多 CPU 系统配置时，状态只被存储到 1 号机的 CPU 模块中。	S（每次 END）	新增	Qn(H)*2 QnPH*2 QnPRH Rem
SD1781	电源模块故障检测状态	电源模块故障检测状态	• 以下列位模式存储冗余电源模块（Q64RP）的故障检测状态。（当检测出冗余电源模块有故障之后，有故障的冗余电源模块的输入电源被切换到 OFF 时，相应位被清除为 0。） • 当主基板不是冗余电源主基板（Q38RB）时，此寄存器存储 0。  • 多 CPU 系统配置时，状态只被存储到 1 号机的 CPU 模块中。	S（每次 END）	新增	
SD1782	用于电源模块 1*1 的瞬间掉电检测计数器	用于电源 1 的瞬间掉电检测计数	• 对电源 1/ 电源 2 的瞬间掉电的次数进行计数。 • 监视安装在冗余电源主基板（Q38RB）上的电源 1/ 电源 2 的状态，并对瞬间掉电的次数进行计数。 • 不监视安装在冗余扩展基板上的电源 1/ 电源 2 的状态。 • 当 CPU 模块启动时，电源 1/ 电源 2 的计数器被清除为 0。 • 如果到两个冗余电源模块中的一个的输入电源被切换为 OFF，和输入电源被切换为 OFF 的冗余电源模块相对应的计数器被清除为 0。 • 每次检测到电源 1/ 电源 2 的瞬间掉电时，计数器加 1。（0 ～ 65535：当计数超出 65535 时，从 0 开始重新计数。） • 当主基板不是冗余电源主基板（Q38RB）时此寄存器存储 0。 • 多 CPU 系统配置时，状态只被存储到 1 号机的 CPU 模块中。	S（每次 END）	新增	
SD1783	用于电源模块 2*1 的瞬间掉电检测计数器	用于电源 1 的瞬间掉电检测计数		S（每次 END）	新增	

*1: “电源 1”表示安装在冗余基板（Q38RB/68RB/Q65WRB）的 POWER1 插槽中的冗余电源模块（Q64RP）。
“电源 2”表示安装在冗余基板（Q38RB/68RB/Q65WRB）的 POWER2 插槽中的冗余电源模块（Q64RP）。

*2: 以序列号的高 5 位为 “04012”以后的 CPU 为对象。
多 CPU 系统配置中，以序列号的高 5 位为 “07032”以后的 CPU 为对象。

附录 5　应用程序实例

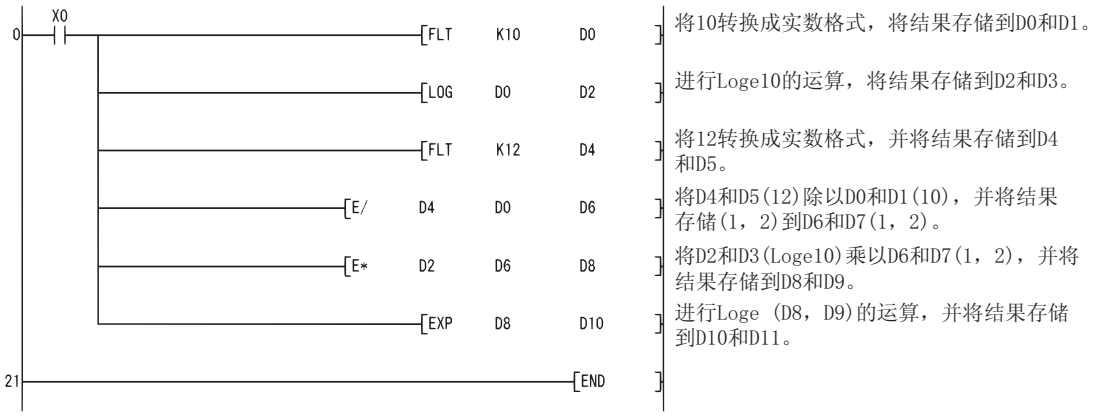
附录 5.1　执行 X^n、$\sqrt[n]{X}$ 操作的程序概念

虽然 QCPU 和 QnACPU 没有执行 X^n、$\sqrt[n]{X}$ 操作的指令，但是 LOG 和 EXP 可以用在一起去执行 X^n、$\sqrt[n]{X}$ 操作。

(1) 执行 X^n 操作的程序的概念可以通过使用 $e^{(n\log eX)}$ 来实现。

X^n 可以通过 $(n\log eX)$ 进行运算。

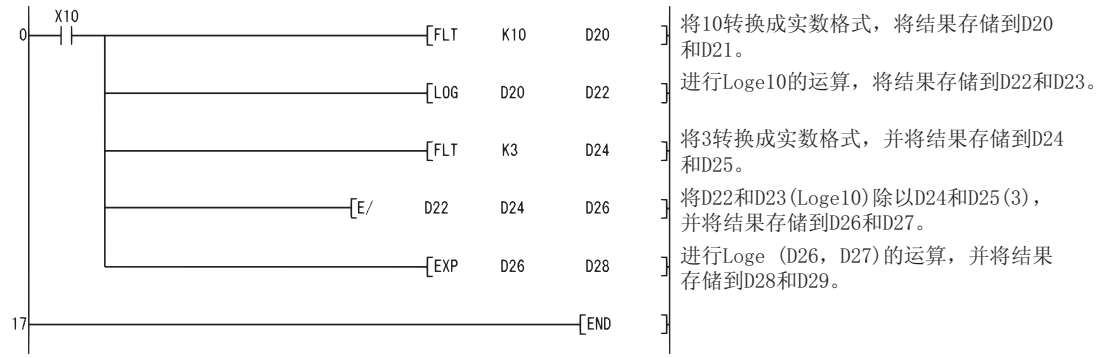
举例来说，$10^{1.2}$ 的运算就是 $e^{(1.2\ \times\ \log e10)}$，它在顺控程序中的表达方式如下所示。



(2) 执行 $\sqrt[n]{X}$

$\sqrt[n]{X}$ 操作的程序的概念可以通过使用 $e^{(\frac{1}{n}\log eX)}$ 来实现。

举例来说，$\sqrt[3]{10}$ 的操作就是 $e^{(\frac{1}{3}\times\log e10)}$，它在顺控程序中的表达方式如下所示。



附录

备忘录

索引

[符号]

-(BIN16 位减法运算) 6-19
$+(链接字符串) ..... 6-59、6-61
$=、$<>、$>、$<=、$<、$>=
(字符串数据比较) 6-11
$MOV(字符串转移) 6-106
*(BIN16 位乘法运算) 6-27
/(BIN16 位除法运算) 6-27
+(BIN16 位加法运算) 6-19
<(BIN16 位数据比较) 6-2
<=(BIN16 位数据比较) 6-2
<>(BIN16 位数据比较) 6-2
=(BIN16 位数据比较) 6-2
>(BIN16 位数据比较) 6-2
>=(BIN16 位数据比较) 6-2

[数字]

16 位否定移动 (CML) 6-108
16 位数据的 4 位分解 (DIS) 7-71
16 位数据的 4 位链接 (UNI) 7-73
16 位数据的 n 位右移 (SFR) 7-46
16 位数据的 n 位左移 (SFL) 7-46
16 位数据的逻辑和 (WOR) 7-11
16 位数据的逻辑积 (WAND) 7-3
16 位数据的右旋 (ROR、RCR) 7-35
16 位数据的总和计算 (WSUM) 7-93
16 位数据的最大值搜索 (MAX) 7-83
16 位数据的最小值搜索 (MIN) 7-86
16 位数据的左旋 (ROL、RCL) 7-38
16 位数据检查 (SUM) 7-63
16 位数据交换 (XCH) 6-116
16 位数据块转移 (FMOV) 6-114
16 位数据搜索 (SER) 7-60
16 位数据异或运算 (WXOR) 7-19
16 位数据与或运算 (WXNR) 7-27
16 位死区控制 (BAND) 7-314
16 位移动 (MOV) 6-100
1 相输入增大或减小 (UDCNT1) 6-138
256 到 8 位编码 (ENCO) 7-67
2 相输入增大或减小 (UDCNT2) 6-141
32 位否定移动 (DCML) 6-108
32 位数据的逻辑和 (DOR) 7-11
32 位数据的逻辑积 (DAND) 7-3
32 位数据的右旋 (DROR、DRCL) 7-41
32 位数据的总和计算 (DWSUM) 7-95
32 位数据的最大值搜索 (DMAX) 7-83
32 位数据的最小值搜索 (DMIN) 7-86
32 位数据的左旋 (DROL、DRCL) 7-44
32 位数据检查 (DSUM) 7-63
32 位数据交换 (DXCH) 6-116
32 位数据搜索 (DSER) 7-60
32 位数据异或运算 (DXOR) 7-19
32 位数据与或运算 (DXNR) 7-27
32 位死区控制 (DBAND) 7-314
32 位转移 (DMOV) 6-100
7 段解码 (SEG) 7-69
8 位到 256 位解码 (DECO) 7-65

[A]

ACOS(浮点数据的 COS-1 运算 (单精度)) 7-265
ACOSD(浮点数据的 COS-1 运算 (双精度)) 7-267
ADRSET(间接地址读取) 7-367
ANB(梯形图块系列) 5-7
AND(=、<>、>、<=、<、>=)
(BIN16 位数据比较) 6-2
AND(A 触点串行连接) 5-2
AND(D=、D<>、D>、D<=、D<、D>=)
(BIN32 位数据比较) 6-4
AND(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较 (单精度)) 6-6
AND(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较 (双精度)) 6-8
AND($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较) 6-11
ANDF(脉冲串行连接 / 上升沿) 5-5
ANDP(脉冲串行连接 / 下降沿) 5-5
ANI(B 触点串行连接) 5-2
ASC(十六进制 BIN 到 ASCII 的转换) 7-230
ASCII 到十六进制 BIN 的转换 (HEX) 7-232
ASCII 码 LED 显示指令 (LED) 7-159
ASIN(浮点数据的 SIN-1 运算 (单精度)) 7-260
ASIND(浮点数据的 SIN-1 运算 (双精度)) 7-263
ATAN(浮点数据的 TAN-1 运算 (单精度)) 7-269
ATAND(浮点数据的 TAN-1 运算 (双精度)) 7-271
A 触点并行连接 (OR) 5-2
A 触点串行连接 (AND) 5-2
A 触点运算开始 (LD) 5-2

[B]

B-(BCD4 位减法运算) 6-31
B*(BCD4 位乘法运算) 6-39
B/(BCD4 位除法运算) 6-39
B+(BCD4 位加法运算) 6-31
BACOS(BCD 类型 COS-1 运算) 7-307
BAND(16 位死区控制) 7-314
BASIN(BCD 类型 SIN-1 运算) 7-305
BATAN(BCD 类型 TAN-1 运算) 7-303
BCD(BIN 数据到 4 位数) 6-67
BCD4 位数据乘法和除法运算 (B*、B/) 6-39
BCD4 位数据到十进制 ASCII 的转换 (BCDDA) ... 7-190
BCD4 位数据加法和减法运算 (B+、B-) 6-31
BCD4 位数据平方根 (BSQR) 7-296
BCD8 位加法和减法运算 (DB+、DB-) 6-35
BCD8 位数据乘法和除法运算 (DB*、DB/) 6-41
BCD8 位数据到十进制 ASCII 的转换 (DBCDDA) .. 7-190
BCD8 位数据平方根 (BDSQR) 7-296
BCDDA(BCD4 位数据到十进制 ASCII 的转换) ... 7-190
BCD 的转换
  BIN 数据到 BCD4 位数据的转换 (BCD) 6-67
  BIN 数据到 BCD8 位数据的转换 (DBCD) 6-67
BCD 格式数据到浮点数 (EREXP) 7-246
BCD 类型 COS-1 运算 (BACOS) 7-307
BCD 类型 COS 运算 (BCOS) 7-301
BCD 类型 SIN-1 运算 (BASIN) 7-305

BCD 类型 SIN 运算 (BSIN) 7-299
BCD 类型 TAN-1 运算 (BATAN) 7-309
BCD 类型 TAN 运算 (BTAN) 7-303
BCOS(BCD 类型 COS 运算) 7-301
BDSQR(BCD8 位数据平方根) 7-296
BIN(BCD4 位数据到 BIN 数据) 6-69
BIN 16 位乘法和除法运算 (*、/) 6-27
BIN 16 位到 BIN32 位的转换 (DBL) 6-82
BIN 16 位到格雷码的转换 (GRY) 6-84
BIN 16 位加法和减法运算 (+、-) 6-19
BIN 16 位数据比较 (=、<>、>、<=、<、>=) 6-2
BIN 16 位数据排序 (SORT) 7-89
BIN 32 位乘法和除法运算 (D*、D/) 6-29
BIN 32 位到 BIN16 位的转换 (WORD) 6-83
BIN 32 位到格雷码的转换 (DGRY) 6-84
BIN 32 位加法和减法运算 (D+、D-) 6-23
BIN 32 位数据比较
(D=、D<>、D>、D<=、D<、D>=) 6-4
BIN 32 位数据排序 (DSORT) 7-89
BIN16 位到浮点数据的转换 (单精度)(FLT) 6-72
BIN16 位到浮点数据的转换 (双精度)(FLTD) ... 6-75
BIN16 位到十进制 ASCII 的转换 (BINDA) 7-184
BIN16 位到十六进制 ASCII 的转换 (BINHA) 7-187
BIN16 位到字符串的转换 (STR) 7-209
BIN16 位数据的 2 位补码 (NEG) 6-88
BIN16 位数据的死区控制 (ZONE) 7-317
BIN32 位到浮点数据的转换 (单精度)
(DFLT) 6-72
BIN32 位到浮点数据的转换 (双精度)
(DFLTD) 6-75
BIN32 位到十进制 ASCII 的转换 (DBINDA) 7-184
BIN32 位到十六进制 ASCII 的转换 (DBINHA) ... 7-187
BIN32 位到字符串的转换 (DSTR) 7-209
BIN32 位数据的 2 位补码 (DNEG) 6-88
BIN32 位数据的上限和下限控制 (DLIMIT) 7-311
BIN32 位数据的死区控制 (DZONE) 7-317
BINDA(BIN16 位到十进制 ASCII 的转换) 7-184
BINHA(BIN16 位到十六进制 ASCII 的转换) 7-187
BIN 的转换
  BCD4 位数据到 BIN16 位的转换 (BIN) 6-69
  BCD8 位数据到 BIN32 位的转换 (DBIN) 6-69
  浮点数据到 BIN16 位的转换 (单精度)
  (INT) 6-77
  浮点数据到 BIN16 位的转换 (双精度)
  INTD) 6-80
  浮点数据到 BIN32 位的转换 (单精度)
  (DINT) 6-77
  浮点数据到 BIN32 位的转换 (双精度)
  (DINTD) 6-80
BIN 块数据比较 (BKCMP □) 6-15
BK-(块减法运算) 6-56
BK+(块加法运算) 6-56
BKAND(块逻辑积) 7-9
BKBCD(块 BIN16 位数据到块 BCD4 位数据的转换) 6-92
BKBIN(块 BCD4 位数据到块 BIN16 位数据的转换) 6-94
BKCMP □ (BIN 块数据比较) 6-15
BKOR(块逻辑和运算) 7-17
BKRST(位软元件的批量复位) 7-58
BKXNR(块与或运算) 7-33
BKXOR(块异或运算) 7-25
BMOV(16 位数据块移动) 6-111
BREAK(FOR 到 NEXT 指令循环的强制结束) 7-100
BRST(字软元件的位复位) 7-53
BSET(字软元件的位置位) 7-53
BSFL(n 位数据的 1 位左移) 7-49
BSFR(n 位数据的 1 位右移) 7-49
BSIN(BCD 类型 SIN 运算) 7-299
BSQR(BCD4 位数据平方根) 7-296
BTAN(BCD 类型 TAN 运算) 7-303
BTOW(以字节为单位进行数据链接) 7-79
BXCH(块 16 位数据交换) 6-118
B 触点运算开始 (LDI) 5-2
报警器输出 (OUT F) 5-24
比较 (BIN16 位数据) 6-2
比较 (BIN32 位数据) 6-4
比较 (字符串数据) 6-11
比较操作指令 6-2
比较操作指令列表 2-10
编程注意事项 3-22
变址寄存器的批量存储 (ZPUSH) 7-372
变址寄存器的批量恢复 (ZPOP) 7-372
变址修饰 3-12
变址修饰中的修改值的指定 (IXDEV、IXSET) ... 7-133
并行连接 (OR、ORI) 5-2
并行连接 (ORB) 5-7

[C]

CALL(子程序调用) 7-102
CHKCIR(改变 CHK 指令的检查格式) 7-170
CHKEND(改变 CHK 指令的检查格式) 7-170
CHKST、CHK(特殊格式故障检查) 7-166
CJ(指针分支指令) 6-121
CML(16 位否定移动) 6-108
COM(刷新指令) ..... 7-126、9-63
COMRD(读软元件注释数据) 7-202
COS(浮点数据的 COS 运算 (单精度)) 7-252
COSD(浮点数据的 COS 运算 (双精度)) 7-254
CPU 启动时的动作模式设定指令 (S.STMODE) 10-2
CPU 切换时的动作模式设定指令 (S.CGMODE) 10-4
步数 3-28
乘法
  BCD4 位数据 (B*) 6-39
  BCD8 位数据 (DB*) 6-41
  BIN16 位 (*) 6-27
  BIN32 位 (D*) 6-29
  浮点数据乘法运算 (单精度)(E*) 6-51
  浮点数据乘法运算 (双精度)(ED*) 6-53
程序备用指令 (PSTOP) 7-349
程序低速执行登记 (PLOW) 7-354
程序分支指令表 2-25
程序跟踪的复位 (PTRAR) 7-182
程序跟踪的设置 (PTRA) 7-182
程序跟踪的执行 (PTRAEXE) 7-182
程序控制指令 2-49
程序扫描执行登记指令 (PSCAN) 7-352
程序输出 OFF 备用指令 (POFF) 7-350

索引

程序文件之间的输出 OFF 调用 (EFCALL)..... 7-117
程序执行控制指令表 2-25
程序执行状态检查指令 (PCHK)..... 7-356
出栈 (MPS)..... 5-9
除法运算
BCD4 位数据 (B/) 6-39
BCD8 位除法运算 (DB/) 6-41
BIN16 位 (/) 6-27
浮点数据的除法运算 (单精度)(E/)..... 6-51
浮点数据的除法运算 (双精度)(ED/)..... 6-53
触点指令 2-6
触点指令
并行连接 (OR、ORI)..... 5-2
串行连接 (AND、ANI)..... 5-2
脉冲并行连接 (ORF、ORP)..... 5-5
脉冲串行连接 (ANF、ANP)..... 5-5
脉冲运算开始 (LDF、LDP)..... 5-5
运算开始 (LD、LDI)..... 5-2
串行连接 (ANB) 5-7
串行连接 (AND、ANI)..... 5-2
从本地站读取软元件数据 (ZNRD)..... 8-109
从标准 ROM 读数据 (S.DEVLD)..... 9-33
从程序存储器中卸载程序 (PUNLOADP)..... 9-38
从存储器载入程序 (PLOADP)..... 9-35
从其它站的 CPU 共享存储器读取数据 (FROM)... 9-57
从其它站读取软元件数据 (ZNRD)..... 8-105
从数据表读取最旧数据 (FIFR)..... 7-138
从数据表读取最新数据 (FPOP)..... 7-140
从数据列表中删除数据 (FDEL)..... 7-142
从右边提取字符串数据 (RIGHT)..... 7-234
从指定文件中读取数据 (SP.FREAD)..... 9-19
从智能功能模块读取 2 字数据 (DFRO)..... 7-145
从智能功能模块中读取 1 字数据 (FROM)..... 7-145
从中断程序中恢复 (IRET)..... 6-134
从子程序的返回 (RET)..... 7-107
从左边提取字符串数据 (LEFT)..... 7-234
错误显示和报警器复位指令 (LEDR) 7-163

[D]

D-(BIN32 位数据减法运算) 6-23
D*(BIN32 位数据乘法运算) 6-29
D/(BIN32 位数据除法运算) 6-29
D+(BIN32 位数据加法运算) 6-23
D=、D<>、D>、D<=、D<、D>=
(BIN32 位数据比较) 6-4
DABCD(十进制 ASCII 到 BCD4 位数据的转换).. 7-199
DABIN(十进制 ASCII 到 BIN16 位的转换) 7-193
DAND(32 位数据的逻辑乘积) 7-3
DATE-(时钟数据减法运算) 7-337
DATE+(时钟数据加法运算) 7-335
DATERD(读时钟数据) 7-327
DATEWR(写时钟数据) 7-331
DB-(BCD8 位减法运算) 6-35
DB*(BCD8 位乘法运算) 6-41
DB/(BCD 8 位除法运算) 6-41
DB+(BCD8 位加法运算) 6-35
DBAND(32 位死区控制) 7-314
DBCD(BIN 到 BCD8 位数的转换) 6-67
DBCDDA(BCD8 位数据到十进制 ASCII 的转换).. 7-190
DBIN(BCD8 位数据到 BIN16 位的转换) 6-69
DBINDA(BIN32 位到十进制 ASCII 的转换) 7-184
DBINHA(BIN32 位到十六进制 ASCII 的转换) ... 7-187
DBL(BIN16 位到 BIN32 位的转换) 6-82
DCML(32 位否定传送) 6-108
DDABCD
(十进制 ASCII 到 BCD8 位数据的转换) 7-199
DDABIN(十进制 ASCII 到 BIN32 位的转换) 7-193
DDEC(32 位 BIN 数据递减) 6-65
DEC(16 位 BIN 数据递减) 6-63
DECO(8 位到 256 位解码) 7-65
DEG(浮点弧度到角度的转换 (单精度)) 7-277
DEGD(浮点弧度到角度的转换 (双精度)) 7-279
DELTA(直接输出的脉冲转换) 5-38
DFLT(BIN32 位到浮点数据的转换
(单精度)) 6-72
DFLTD(BIN32 位到浮点数据的转换
(双精度)) 6-75
DFRO(从智能功能模块读取 2 字数据) 7-145
DGBIN(格雷码到 BIN16 位的转换) 6-86
DGRY(BIN32 位到格雷码的转换) 6-84
DHABIN(十六进制 ASCII 到 BIN32 位的转换).. 7-196
DI(中断禁止) 6-125
DINC(递增 32 位 BIN) 6-65
DINT(浮点数据到 BIN32 位的转换
(单精度)) 6-77
DINTD(浮点数据到 BIN32 位的转换
(双精度)) 6-80
DIS(16 位数据的 4 位分组) 7-71
DLIMIT(BIN32 位数据的上限和下限控制) 7-311
DMAX(32 位数据的最大值搜索) 7-83
DMIN(32 位数据的最小值搜索) 7-86
DMOV(32 位转移) 6-100
DNEG(BIN32 位数据的 2 位补码) 6-88
DOR(32 位数据的逻辑和) 7-11
DRCL(32 位数据的左循环) 7-44
DRCR(32 位数据的右循环) 7-41
DROL(32 位数据的左循环) 7-44
DROR(32 位数据的右循环) 7-41
DSER(32 位数据搜索) 7-60
DSFL(n 字数据左移 1 字) 7-51
DSFR(n 字数据右移 1 字) 7-51
DSORT(BIN32 位数据排序) 7-89
DSTR(BIN32 位到字符串的转换) 7-209
DSUM(32 位数据检查) 7-63
DTEST(位测试) 7-55
DTO(智能功能模块 2 字数据写运算) 7-148
DUTY(定时脉冲的产生) 7-360
DVAL(字符串到 BIN32 位的转换) 7-215
DWSUM(32 位数据的总数计算) 7-95
DXCH(16 位数据交换) 6-116
DXNR(32 位数据否定排它逻辑和运算) 7-27
DXOR(32 位排它或运算) 7-19
DZONE(BIN32 位数据的死区控制) 7-317
打印 ASCII 码指令 (PR)..... 7-151
打印注释指令 (PRC)..... 7-154
单精度到双精度的转换 (ECON)..... 6-96

登记路由信息 (RTWRITE)..... 8-133
低速定时器 (OUT T)..... 5-18
低速累计定时器 (OUTH ST)..... 5-18
递减
16 位 BIN(DEC) 6-63
32 位 BIN(DDEC) 6-65
递增
BIN 16 位 (INC) 6-63
BIN 32 位 (DINC) 6-65
调试和故障诊断指令 2-39
定时脉冲的产生 (DUTY)..... 7-360
定时器 (OUT T)..... 5-18
读路径参数 (RTREAD)..... 8-129
读模块信息 (UNIRD)..... 9-2
读取 (MRD)..... 5-9
读取扩展时钟数据 (S.DATERD)..... 9-66
读软元件注释数据 (COMRD)..... 7-202
读时钟数据 (DATERD)..... 7-327

[E]

E-(浮点数减法运算)(单精度))..... 6-43、6-45
E*(浮点数乘法运算)(单精度))..... 6-51
E/(浮点数除法运算)(单精度))..... 6-51
E+(浮点数加法运算)(单精度))..... 6-43、6-45
E=、E<>、E>、E<=、E<、E>=
(浮点数据比较(单精度))..... 6-6
ECALL(在程序文件之间的子集调用)..... 7-112
ECON(单精度到双精度的转换)..... 6-96
ED-(浮点数减法运算)(双精度))..... 6-47、6-49
ED*(浮点数乘法运算)(双精度))..... 6-53
ED/(浮点数除法运算)(双精度))..... 6-53
ED+(浮点数加法运算)(双精度))..... 6-47、6-49
ED=、ED<>、ED>、ED<=、ED<、ED>=
(浮点数据比较(双精度))..... 6-8
EDCON(双精度到单精度的转换)..... 6-98
EDMOV(浮点数据转移(双精度))..... 6-104
EDNEG(浮点数的符号取反(双精度)) 6-91
EFCALL(在程序文件之间的输出 OFF 调用)..... 7-117
EGF(脉冲运算结果/上升沿)..... 5-14
EGP(脉冲运算结果/下降沿)..... 5-14
EI(中断允许)..... 6-125
EMOD(浮点数到 BCD)..... 7-243
EMOV(浮点数据转移(单精度))..... 6-102
ENCO(256 到 8 位编码)..... 7-67
END(结束顺控程序)..... 5-49
ENEG(浮点数符号取反(单精度))..... 6-90
EREXP(BCD 格式数据到浮点数)..... 7-246
EROMWR(批量写运算到 E2PROM 文件寄存器)... 7-374
ESTR(浮点数到字符串的转换)..... 7-220
EVAL(字符串到浮点数的转换)..... 7-220
EXP(浮点数据的指数运算(单精度))..... 7-285
EXPD(浮点数据的指数运算(双精度))..... 7-288

[F]

FCALL(子程序输出 OFF 调用)..... 7-108
FDEL(从数据表中删除数据)..... 7-142
FEND(结束主程序)..... 5-47
FF(位软元件输出取反)..... 5-36
FIFR(从数据表读取旧数据)..... 7-138
FIFW(写数据到数据表)..... 7-136
FINS(在数据表中插入数据)..... 7-142
FLT(BIN16 位到浮点数据的转换(单精度))..... 6-72
FLTD(BIN16 位到浮点数据的转换(双精度))... 6-75
FMOV(16 位数据块传送)..... 6-114
FOR(FOR 到 NEXT)..... 7-97
FOR 到 NEXT(FOR、NEXT)..... 7-97
FOR 到 NEXT 指令循环的强制结束 (BREAK)..... 7-100
FPOP(从数据表读取新数据)..... 7-140
FROM(从智能功能模块读取 1 字数据)..... 7-145
FROM(其它站点的 CPU 共享存储区读取数据)... 9-57
发送数据到其它站点 (SEND)..... 8-62
浮点弧度到角度的转换(单精度)(DEG)..... 7-277
浮点弧度到角度的转换(双精度)(DEGD)..... 7-279
浮点角度到弧度的转换(单精度)(RAD)..... 7-273
浮点角度到弧度的转换(双精度)(RADD)..... 7-275
浮点数到 BCD(EMOD)..... 7-243
浮点数据比较(单精度)
(E=、E<>、E>、E<=、E<、E>=) 6-6
浮点数据比较(双精度)
(ED=、ED<>、ED>、ED<=、ED<、ED>=) 6-8
浮点数据乘除法运算(单精度)(E*、E/)..... 6-51
浮点数据乘除法运算(双精度)(ED*、ED/)..... 6-53
浮点数据传送(单精度)(EMOV)..... 6-102
浮点数据传送(双精度)(EDMOV)..... 6-104
浮点数据到字符串的转换 (ESTR)..... 7-220
浮点数据的 COS^{-1} 运算(单精度)(ACOS) 7-265
浮点数据的 COS^{-1} 运算(双精度)(ACOSD) 7-267
浮点数据的 COS 运算(单精度)(COS)..... 7-252
浮点数据的 COS 运算(双精度)(COSD)..... 7-254
浮点数据的 SIN^{-1} 运算(单精度)(ASIN) 7-260
浮点数据的 SIN^{-1} 运算(双精度)(ASIND) 7-263
浮点数据的 SIN 运算(单精度))(SIN)..... 7-248
浮点数据的 SIN 运算(双精度))(SIND)..... 7-250
浮点数据的 TAN^{-1} 运算(单精度)(ATAN) 7-269
浮点数据的 TAN^{-1} 运算(双精度)(ATAND) 7-271
浮点数据的 TAN 运算(单精度)(TAN)..... 7-256
浮点数据的 TAN 运算(双精度)(TAND)..... 7-258
浮点数据的指数运算(单精度)(EXP)..... 7-285
浮点数据的指数运算(双精度)(EXPD)..... 7-288
浮点数据的转换(单精度)(FLT、DFLT)..... 6-72
浮点数据的转换(双精度)(FLTD、DFLTD)..... 6-75
浮点数据的自然对数运算(单精度)(LOG)..... 7-290
浮点数据的自然对数运算(双精度)(LOGD).... 7-292
浮点数据符号取反(单精度)(ENEG) 6-90
浮点数据符号取反(双精度)(EDNEG)..... 6-91
浮点数区域的平方根运算(单精度)(SQR)..... 7-281
浮点数区域的平方根运算(双精度)(SQRD).... 7-283
浮动点数据的加法和减法运算(单精度)
(E+、E-) ..... 6-43、6-45
浮动点数据的加法和减法运算(双精度)
(ED+、ED-) ..... 6-47、6-49
复位报警器 (RST F)..... 5-31
复位采样跟踪 (STRAR)..... 7-180
复位看门狗定时器 (WDT)..... 7-358
复位软元件 (RST)..... 5-28、5-31
复位主控制 (MCR)..... 5-43
复位状态闭锁 (SLTR)..... 7-178

索引

[G]

GBIN(格雷码到BIN16位的转换)..... 6-86
GOEND(跳转到END)..... 6-124
GRY(BIN16位到格雷码的转换)..... 6-84
改变CHK指令的检查格式(CHKCIR、CHKEND)... 7-170
高速定时器(OUTH T)..... 5-18
高速累计定时器(OUTH ST)..... 5-18
格雷码到BIN16位的转换(GBIN)..... 6-86
格雷码到BIN32位的转换(DGBIN)..... 6-86

[H]

HABIN(十六进制ASCII到BIN16位的转换)... 7-196
HEX(ASCII到十六进制BIN的转换)..... 7-232
HOUR(改变时间数据格式)..... 7-341
恒定周期脉冲输出(PLSY)..... 6-157
缓冲存储区访问指令..... 2-38
缓冲存储区批量刷新指令(S.SPREF)..... 10-10
换页(NOPLF)..... 5-53
换页(PAGE n)..... 5-53

[I]

I/O刷新(RFS)..... 6-136
I/O刷新指令列表..... 2-25
IMASK(中断程序掩码)..... 6-125
INC(递增16位BIN)..... 6-63
INSTR(字符串搜索)..... 7-241
INT(浮点数据到BIN16位(单精度))..... 6-77
INTD(浮点数据到BIN16位(双精度))..... 6-80
INV(运算结果取反)..... 5-12
IRET(从中断程序恢复)..... 6-134
IX、IXEND(整个梯形图的变址修饰)..... 7-129
IXDEV(变址修饰中修饰值的指定)..... 7-133
IXSET(变址修饰中修饰值的指定)..... 7-133

[J]

JMP(指针分支)..... 6-121
基本指令..... 2-10
计数器(OUT C)..... 5-22
加法
  BCD4位加法运算(B+)..... 6-31
  BCD8位加法运算(DB+)..... 6-35
  BIN16位加法运算(+)..... 6-19
  BIN32位加法运算(D+)..... 6-23
  浮点数据的加法运算
  (单精度)(E+)..... 6-43、6-45
  浮点数据的加法运算
  (双精度)(ED+)..... 6-47、6-49
  块加法运算(BK+)..... 6-56
间接地址读取(ADRSET)..... 7-367
间接指定..... 3-18
减法
  BCD4位减法运算(B-)..... 6-31
  BCD8位减法运算(DB-)..... 6-35
  BIN16位减法运算(-)..... 6-19
  BIN32位减法运算(D-)..... 6-23
  浮点数减法运算(单精度)(E-)..... 6-43、6-45
  浮点数减法运算(双精度)(ED-)..... 6-47、6-49
  块的减法运算(BK-)..... 6-56
将n位的数据右移1位(BSFR)..... 7-49
将n位的数据左移1位(BSFL)..... 7-49
将n字的数据右移1字(DSFR)..... 7-51
将n字的数据左移1字(DSFL)..... 7-51
教学定时器(TTMR)..... 6-144
接收
  浮点数的符号取反(单精度)(ENEG)..... 6-90
  浮点数的符号取反(双精度)(EDNEG)..... 6-91
  位软元件输出取反(FF)..... 5-36
  运算结果取反(INV)..... 5-12
结构体创建指令..... 2-35
结束顺控程序(END)..... 5-49
结束主程序(FEND)..... 5-47
矩阵输入(MTR)..... 6-161

[K]

KEY(通过键盘的数字键输入)..... 7-368
块16位交换(BXCH)..... 6-118
块16位转移(BMOV)..... 6-111
块BCD4位数数据到块BIN16位数据的转换
(BKBIN)..... 6-94
块BIN16位数据到BCD4位数据的转换
(BKBCD)..... 6-92
块加法运算(BK+)..... 6-56
块减法运算(BK-)..... 6-56
块逻辑乘(BKAND)..... 7-9
块逻辑和运算(BKOR)..... 7-17
块异或运算(BKXOR)..... 7-25
块与或运算(BKXNR)..... 7-33
扩展时钟数据的加法运算(S.DATE+)..... 9-66
扩展时钟数据的减法运算(S.DATE-)..... 9-66

[L]

LD($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较)..... 6-11
LD(=、<>、>、<=、<、>=)
(BIN16位数据比较)..... 6-2
LD(A触点运算开始)..... 5-2
LD(D=、D<>、D>、D<=、D<、D>=)
(BIN32位数据比较)..... 6-4
LD(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较(单精度))..... 6-6
LD(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较(双精度))..... 6-8
LDF(脉冲运算开始/下降沿)..... 5-5
LDI(B触点运算开始)..... 5-2
LDP(脉冲运算开始/上升沿)..... 5-5
LED(ASCII码LED显示指令)..... 7-159
LEDC(注释的LED显示指令)..... 7-161
LEDR(错误显示和报警器复位指令)..... 7-163
LEFT(从左边提取字符串数据)..... 7-234
LEN(字符串长度检测)..... 7-207
LIMIT(BIN16位的上下限控制)..... 7-311
LOG(浮点数据的自然对数运算(单精度))..... 7-290
LOGD(浮点数据的自然对数运算(双精度))... 7-292
来自其它站点的瞬时请求(REQ)..... 8-81
来自外围软元件的键盘输入(PKEY)..... 7-345
连接指令

连接指令列表 2-7
链接字符串 ($+) 6-59
梯形图块并行连接 (ORB) 5-7
梯形图块串行连接 (ANB) 5-7
链接刷新指令 2-51
链接字符串 ($+) ..... 6-59、6-61
逻辑和 7-2
逻辑积 7-2
逻辑运算指令 2-27

[M]

MAX(16 位数据的最大值搜索) 7-83
MC(主控制置位) 5-43
MCR(主控制复位) 5-43
MEF(脉冲运算结果 / 下降沿) 5-13
MEP(脉冲运算结果 / 上升沿) 5-13
MIDR(字符串中任意选择) 7-237
MIDW(字符串中任意替换) 7-237
MIN(16 位数据的最小值搜索) 7-86
MOV(16 位传送) 6-100
MPP(运算结果入栈) 5-9
MPS(运算结果出栈) 5-9
MRD(操作结果读) 5-9
MSG(显示信息到外围设备) 7-343
MTR(矩阵输入) 6-161
脉冲并行连接 (ORF、ORP) 5-5
脉冲串行连接 (ANDF、ANDP) 5-5
脉冲宽度调制 (PWM) 6-159
脉冲密度测量 (SPD) 6-155
脉冲运算结果
变址继电器运算结果的脉冲转换 (EGF、EGP) .. 5-14
运算结果转换 (MEF、MEP) 5-13
脉冲运算开始 (LDF、LDP) 5-5
脉冲转换
(EGF、EGP) 5-14
(MEF、MEP) 5-13
(DELTA) 5-38

[N]

NEG(BIN16 位数据的 2 位补码) 6-88
NEXT(FOR 到 NEXT) 7-97
NOP 5-53
NOP(无处理) 5-53
NOPLF(无处理换页) 5-53
NUNI(任意数据的链接) 7-75

[O]

OR($=、$<>、$>、$<=、$<、$>=)
(字符串数据比较) 6-11
OR(=、<>、>、<=、<、>=)
(BIN16 位数据比较) 6-2
OR(A 触点并行连接) 5-2
OR(D=、D<>、D>、D<=、D<、D>=)
(BIN32 位数据比较) 6-4
OR(E=、E<>、E>、E<=、E<、E>=)
(浮点数据比较 (单精度)) 6-6
OR(ED=、ED<>、ED>、ED<=、ED<、ED>=)
(浮点数据比较 (双精度)) 6-8
ORB(梯形图块并行连接) 5-7
ORF(脉冲并行连接 / 下降沿) 5-5
ORI(B 触点并行连接) 5-2
ORP(脉冲并行连接 / 上升沿) 5-5
Or 取反 (ORI) 5-2
OUT
报警器输出 (OUT F) 5-24
低速定时器 (OUT T) 5-18
低速累计定时器 (OUT ST) 5-18
高速定时器 (OUTH T) 5-18
高速累计定时器 (OUTH ST) 5-18
计数器 (OUT C) 5-22
输出 (OUT) 5-16

[P]

PAGE(无处理换页) 5-53
PCHK(程序执行状态检查指令) 7-356
PLF(下降沿输出) 5-33
PLOADP(从存储卡中装载程序) 9-35
PLOW(程序低速执行登记) 7-354
PLS(上升沿输出) 5-33
PLSY(固定周期脉冲输出) 6-157
POFF(程序输出 OFF 备用指令) 7-350
PR(打印 ASCII 码指令) 7-151
PRC(打印注释指令) 7-154
PSCAN(程序扫描执行登记指令) 7-352
PSTOP(程序备用指令) 7-349
PSWAPP(装载 + 卸载) 9-40
PTRA(程序跟踪的设置) 7-182
PTRAEXE(程序跟踪的执行) 7-182
PTRAR(程序跟踪的复位) 7-182
PUNLOADP(从程序存储器中卸载程序) 9-38
PWM(脉冲宽度调制) 6-159

[Q]

QCDSET(注释用文件的设置) 7-325
QCPU 专用指令 2-53
QDRSET(文件寄存器用文件的设置) 7-322
QnA 链接专用指令 8-18
其它方便指令 2-6
其它站点读字软元件数据 (READ) 8-18
其它站点读字软元件数据 (SREAD) 8-29
其它站点接收数据 (RECV) 8-72
其它指令 5-51
顺控指令 2-6
应用指令 2-27
切换文件寄存器号码 (RSET) 7-320
取反
位软元件输出取反 (FF) 5-36
运算结果取反 (INV) 5-12

[R]

RAD(浮点数角度到弧度的转换 (单精度)) ... 7-273
RADD(浮点数角度到弧度的转换 (双精度)) ... 7-275
RAMP(斜坡信号) 6-152
RBMOV(文件寄存器的高速块传送) 9-43
RCL(16 位数据的左旋) 7-38
RCR(16 位数据的右旋) 7-35

索引

READ(从其它站读取字软元件数据)..... 8-18
RECV(从其它站接收数据)..... 8-72
REQ(来自其它站的瞬时请求)..... 8-81
RET(从子程序的返回)..... 7-107
RFRP(从远程 I/O 站的特殊功能模块中
读数据)..... 8-121
RFS(I/O 刷新)..... 6-136
RIGHT(从右边提取字符串数据)..... 7-234
RND(随机数生成和序列更新)..... 7-294
ROL(16 位数据的左旋)..... 7-38
ROR(16 位数据的右旋)..... 7-35
ROTC(旋转台就近控制)..... 6-149
RSET(文件寄存器的块号切换)..... 7-320
RST
  复位报警器 (RST F)..... 5-31
  复位软元件 (RST)..... 5-28
RTOP(写数据到远程 I/O 站的
特殊功能模块)..... 8-125
RTREAD(读路由信息)..... 8-129
RTWRITE(登录路由信息)..... 8-133
任意数据的分解 (NDIS)..... 7-75
任意数据的链接 (NUNI)..... 7-75
冗余系统指令(对于 Q4ARCPU)..... 2-55
如何读指令..... 4-2
如何读指令表..... 2-4
入栈 (MPP)..... 5-9
软元件范围检查..... 3-22

[S]

S.CGMODE(CPU 切换时间动作模式设置)..... 10-4
S.DATE-(扩展时钟数据减法运算)..... 9-66
S.DATE+(扩展时钟数据加法运算)..... 9-66
S.DATERD(读扩展时钟数据)..... 9-66
S.DEVLD(从标准 ROM 读数据)..... 9-33
S.SPREF(缓冲存储器批量刷新)..... 10-10
S.STMODE(CPU 启动时的动作模式设置指令).... 10-2
S.TO(向 CPU 共享存储器写入)..... 9-49
S.TRUCK(数据跟踪指令)..... 10-6
SCJ(指针分支指令)..... 6-121
SECOND(改变时间数据格式)..... 7-339
SEG(7 段解码)..... 7-69
SEND(发送数据到其它站点)..... 8-62
SER(16 位数据搜索)..... 7-60
SET
  设置报警器 (SET F)..... 5-31
  设置软元件 (SET)..... 5-26
SFL(16 位数据的 n 位左移)..... 7-46
SFR(16 位数据的 n 位右移)..... 7-46
SFT(位软元件移位)..... 5-40
SIN(浮点数据的 SIN 运算)(单精度))..... 7-248
SIND(浮点数据的 SIN 运算(双精度))..... 7-250
SLT(设置状态闭锁)..... 7-178
SLTR(复位状态闭锁)..... 7-178
SORT(BIN16 位数据排序)..... 7-89
SP.CONTSW(系统转换指令)..... 11-2
SP.DEVST(写数据到标准 ROM)..... 9-31
SP.FREAD(从指定文件中读取数据)..... 9-19
SP.FWRITE(写数据到指定的文件)..... 9-8
SPD(脉冲密度测量)..... 6-155
SQR(浮点数区域的平方根运算(单精度))..... 7-281
SQRD(浮点数区域的平方根运算(双精度))... 7-283
SREAD(从其它站读取字软元件数据)..... 8-29
SRND(随机数生成和序列更新)..... 7-294
STMR(特殊功能定时器)..... 6-146
STOP(顺控程序停止)..... 5-51
STR(BIN16 位到字符串的转换)..... 7-209
STRA(设置采样跟踪)..... 7-180
STRAR(复位采样跟踪)..... 7-180
SUM(16 位数据检查)..... 7-63
SWAP(最高和最低字节交换)..... 6-120
SWRITE(写软元件数据到其它站)..... 8-51
上升沿输出 (PLS)..... 5-33
设置报警器 (SET F)..... 5-31
设置采样跟踪 (STRA)..... 7-180
设置软元件 (SET)..... 5-26、5-31
设置主控制 (MC)..... 5-43
设置状态闭锁 (SLT)..... 7-178
十进制 ASCII 到 BCD4 位的转换 (DABCD)..... 7-199
十进制 ASCII 到 BCD8 位的转换 (DDABCD)..... 7-199
十进制 ASCII 到 BIN16 位的转换 (DABIN)..... 7-193
十进制 ASCII 到 BIN32 位的转换 (DDABIN)..... 7-193
十六进制 ASCII 到 BIN16 位的转换 (HABIN).... 7-196
十六进制 ASCII 到 BIN32 位的转换 (DHABIN)... 7-196
十六进制 BIN 到 ASCII 的转换 (ASC)..... 7-230
时间检查指令 (TIMCHK)..... 7-362
时间数据转换 (HOUR)..... 7-341
时间数据转换 (SECOND)..... 7-339
时钟数据加法运算 (DATE+)..... 7-335
时钟数据减法运算 (DATE-)..... 7-337
时钟指令..... 2-48
实数数据..... 3-8
输出取反 (FF)..... 5-36
输出指令 (OUT)..... 5-16
输出指令 (OUT)..... 5-16
输出指令列表..... 2-8
数据处理指令..... 2-32
数据跟踪指令 (S.TRUCK)..... 10-6
数据控制指令..... 2-46
数据链接指令..... 2-51
数据链接指令..... 2-51
数据列表操作指令..... 2-37
数据转换指令..... 6-67
数据转换指令列表..... 2-21
数字键输入 (KEY)..... 7-368
刷新指令 (COM)..... 7-126
双精度到单精度的转换 (EDCON)..... 6-98
双字数据..... 3-6
顺控程序停止 (STOP)..... 5-51
顺控指令..... 2-6
算术运算指令..... 2-16
随机数生成 (RND/SRND)..... 7-294

[T]

TAN(浮点数据的 TAN 运算(单精度))..... 7-256
TAND(浮点数据的 TAN 运算(双精度))..... 7-258
TEST(位测试)..... 7-55
TIMCHK(时间检查指令)..... 7-362
TO(写 1 字数据到智能功能模块)..... 7-148

TRACE(跟踪设置) 9-6
TRACER(跟踪复位) 9-6
TTMR(教学定时器) 6-144
特殊定时器 (STMR) 6-146
特殊格式故障检查 (CHKST、CHK) 7-166
特殊功能指令 2-43
梯形图块并行连接 (ORB) 5-7
梯形图块串行连接 (ANB) 5-7
跳转到 END(GOEND) 6-124
通过键盘的数字键输入 (KEY) 7-368

[U]

UDCNT1(计数器 1 相输入增大或减小) 6-138
UDCNT2(计数器 2 相输入增大或减小) 6-141
UNI(16 位数据的 4 位链接) 7-73
UNIRD(读模块信息) 9-2

[V]

VAL(字符串到 BIN16 位的转换) 7-215

[W]

WAND(16 位数据的逻辑积) 7-3
WDT(复位看门狗定时器) 7-358
WOR(16 位数据的逻辑和) 7-11
WORD(BIN32 位到 BIN16 位的转换) 6-83
WRITE(向其它站写入数据) 8-40
WSUM(16 位数据的总数计算) 7-93
WTOB(以字节为单位的数据分解) 7-79
WXNR(16 位数据否定排它逻辑和运算) 7-30
WXNR(16 位数据与或运算) 7-27
WXOR(16 位数据排它或运算) ..... 7-19、7-22
外围设备指令 2-49
网络刷新指令 (ZCOM) 8-10
位测试 (TEST/DTEST) 7-55
位处理指令 2-31
位软元件的批量复位 (BKRST) 7-58
位软元件的位数指定 3-4
位软元件输出反转 (FF) 5-36
位软元件移动 (SET) 5-40
位数据 3-3
位数指定 3-4
文件寄存器 1 字节写 (ZRWRB) 7-365
文件寄存器的高速块转移 (RBMOV) 9-43
文件寄存器直接读取 1 字节数据 (ZRRDB) 7-363
无处理 (NOP、NOPLF、PAGE) 5-53

[X]

XCALL(子程序调用) 7-121
XCH(32 位数据交换) 6-116
系统转换 (SP.CONTSW) 11-2
显示信息到外围设备 (MSG) 7-343
显示指令 2-38
相关程序手册 1-2
向 E^2PROM 文件寄存器的批量写入操作
(EROMWR) 7-374
斜坡信号 (RAMP) 6-152
写 1 字数据到智能功能模块 (TO) 7-148
写 2 字数据到智能功能模块 (DTO) 7-148
写入自站 CPU 的 CPU 共享存储器 CPU- 9-47
写入自站 CPU 的 CPU 共享存储器 CPU-S.TO 9-49
写入自站 CPU 的 CPU 共享存储器 CPU-TO 9-52
写软元件数据到本地站 (ZNWR) 8-117
写软元件数据到其它站 (ZNWR) 8-113
写时钟数据 (DATEWR) 7-331
写数据到标准 ROM(SP.DEVST) 9-31
写数据到其它站 (WRITE) 8-40
写数据到数据表 (FIFW) 7-136
写数据到指定的文件 (SP.FWRITE) 9-8
写字软元件数据到其它站 (SWRITE) 8-51
循环表近路径循环移位控制 (ROTC) 6-149
循环指令 2-30

[Y]

移位指令 ..... 5-40、7-46
(应用指令) 2-27
移位指令表(顺控程序) 2-8
以字节为单位的数据分解 (WTOB) 7-79
以字节为单位的数据链接 (BTOW) 7-79
应用指令 2-27
用于文件寄存器的设置文件 (QDRSET) 7-322
用于注释的文件设置 (QCDSET) 7-325
与远程 I/O 站通讯
从远程 I/O 站的特殊功能模块中读数据
(RFRP) 8-121
从远程 I/O 站的特殊功能模块中读数据
(ZNFR) 8-95
向远程 I/O 站的特殊功能模块中写数据
(RTOP) 8-125
向远程 I/O 站的特殊功能模块中写数据
(ZNTO) 8-100
运算错误 3-22
运算结果出栈 (MPS) 5-9
运算结果读 (MRD) 5-9
运算结果取反 (INV) 5-12
运算结果入栈 (MPP) 5-9
运算开始 (LD、LDI) 5-2

[Z]

ZCOM(网络刷新指令) 8-10
ZNFR(从远程 I/O 站的特殊功能模块中
读取数据) 8-95
ZNRD(从本地站读取软元件数据) 8-109
ZNRD(从其它站读取软元件数据) 8-105
ZNTO(写数据到远程 I/O 站中的
特殊功能模块) 8-100
ZNWR(写软元件数据到本地站) 8-117
ZNWR(写软元件数据到其它站) 8-113
ZONE(BIN16 位数据的死区控制) 7-317
ZPOP(变址寄存器的批量恢复) 7-372
ZPUSH(变址寄存器的批量保存) 7-372
ZRRDB(从文件寄存器中读取 1 字节) 7-363
ZRWRB(文件寄存器中写入 1 字节) 7-365
在程序文件之间的子程序调用 (ECALL) 7-112
在数据表中插入数据 (FINS) 7-142
增大 / 减小计数器
计数器 1 相输入增大或减小 (UDCNT1) 6-138

索引

计数器 2 相输入增大或减小 (UDCNT2)..... 6-141
直接输出的脉冲转换 (DELTA)..... 5-38
指定数据 3-3
指令类型 2-2
指令执行的条件 3-27
指针分支指令 (CJ、SCJ、JMP)..... 6-121
中断程序掩码 (IMASK)..... 6-125
中断禁止 (DI)..... 6-125
中断允许 (EI)..... 6-125
终端指令表 2-9
主控制指令 5-43
注释的 LED 显示指令 (LEDC)..... 7-161
转换
BCD4 位数据到 BIN 数据的转换 (BIN) 6-69
BCD8 位数据到 BIN 数据的转换 (DBIN) 6-69
BIN16 位到浮点数据的转换 (单精度)
(FLT) 6-72
BIN16 位到浮点数据的转换 (双精度)
(FLTD) 6-75
BIN16 位到格雷码的转换 (GRY) 6-84
BIN16 位数据到 BIN32 位的转换 (DBL) 6-82
BIN32 位到 BIN16 位的转换 (WORD) 6-83
BIN32 位到浮点数据的转换 (单精度)
(DFLT) 6-72
BIN32 位到浮点数据的转换 (双精度)
(DFLTD) 6-75
BIN32 位到格雷码的转换 (DGRY) 6-84
BIN 到 BCD4 位数据的转换 (BCD) 6-67
BIN 到 BCD8 位数据的转换 (DBCD) 6-67
单精度到双精度的转换 (ECON)..... 6-96
浮点数据到 BIN16 位的转换 (单精度)
(INT) 6-77
浮点数据到 BIN16 位的转换 (双精度)
(INTD) 6-80
浮点数据到 BIN32 位的转换 (单精度)
(DINT) 6-77
浮点数据到 BIN32 位的转换 (双精度)
(DINTD) 6-80
格雷码到 BIN16 位的转换 (GBIN)..... 6-86
格雷码到 BIN32 位的转换 (DGBIN)..... 6-86
双精度到单精度的转换 (EDCON)..... 6-98
转换指令 2-47
装载 + 卸载 (PSWAPP)..... 9-40
子程序的输出 OFF 调用 (FCALL)..... 7-108
子程序调用 (CALL)..... 7-102
子程序调用 (XCALL)..... 7-121
子程序输出 OFF 调用 (FCALL)..... 7-108
子集处理 3-20
字符串长度检测 (LEN)..... 7-207
字符串处理指令 2-40
字符串到 BIN16 位的转换 (VAL)..... 7-215
字符串到 BIN32 位的转换 (DVAL)..... 7-215
字符串到浮点数据的转换 (EVAL)..... 7-226
字符串的任意选择和替换 (MIDR)..... 7-237
字符串的任意选择替换 (MIDW)..... 7-237
字符串数据 3-11
字符串数据比较 6-11
字符串搜索 (INSTR)..... 7-241
字符串转移 ($MOV)..... 6-106
字软元件的位复位 (BRST)..... 7-53
字软元件的位指定 3-3
字软元件的位置位 (BSET)..... 7-53
字数据 3-4
最高和最低字节交换 (SWAP)..... 6-120

质保

使用之前请确认以下产品质保的详细说明。

1. 免费质保期限和免费质保范围

在免费质保期内使用本产品时如果出现任何属于三菱责任的故障或缺陷（以下称“故障”），则经销商或三菱服务公司将负责免费维修。

注意如果需要在国内现场或海外维修时，则要收取派遣工程师的费用。对于涉及到更换故障模块后的任何再试运转、维护或现场测试，三菱将不负任何责任。

[免费质保期限]

免费质保期限为自购买日或货到目的地日的一年内。

注意产品从三菱生产并出货之后，最长分销时间为 6 个月，生产后最长的免费质保期为 18 个月。维修零部件的免费质保期不得超过修理前的免费质保期。

［ 免费质保范围 ］

(1) 范围局限于按照使用手册、用户手册及产品上的警示标签规定的使用状态、使用方法和使用环境正常使用的情况下。

(2) 以下情况下，即使在免费质保期内，也要收取维修费用。

1. 因不适当存储或搬运、用户粗心或疏忽而引起的故障。因用户的硬件或软件设计而导致的故障。
2. 因用户未经批准对产品进行改造而导致的故障等。
3. 对于装有三菱产品的用户设备，如果根据现有的法定安全措施或工业标准要求配备必需的功能或结构后本可以避免的故障。
4. 如果正确维护或更换了使用手册中指定的耗材（电池、背光灯、保险丝等）后本可以避免的故障。
5. 因火灾或异常电压等外部因素以及因地震、雷电、大风和水灾等不可抗力而导致的故障。
6. 根据从三菱出货时的科技标准还无法预知的原因而导致的故障。
7. 任何非三菱或用户责任而导致的故障。

2. 产品停产后的有偿维修期限

(1) 三菱在本产品停产后的 7 年内受理该产品的有偿维修。
停产的消息将以三菱技术公告等方式予以通告。

(2) 产品停产后，将不再提供产品（包括维修零件）。

3. 海外服务

在海外，维修由三菱在当地的海外 FA 中心受理。注意各个 FA 中心的维修条件可能会不同。

4. 意外损失和间接损失不在质保责任范围内

无论是否在免费质保期内，对于任何非三菱责任的原因而导致的损失、机会损失、因三菱产品故障而引起的用户利润损失、无论能否预测的特殊损失和间接损失、事故赔偿、除三菱以外产品的损失赔偿、用户更换设备、现场机械设备的再调试、运行测试及其它作业等，三菱将不承担责任。

5. 产品规格的改变

目录、手册或技术文档中的规格如有改变，恕不另行通知。

6. 产品应用

(1) 在使用三菱 MELSEC 通用可编程控制器时，应该符合以下条件：即使在可编程控制器设备出现问题或故障时也不会导致重大事故，并且应在设备外部系统地配备能应付任何问题或故障的备用设备及失效保险功能。

(2) 三菱通用可编程控制器是以一般工业用途等为对象设计和制造的。因此，可编程控制器的应用不包括那些会影响公共利益的应用，如核电厂和其它由独立供电公司经营的电厂以及需要特殊质量保证的应用如铁路公司或用于公用设施目的的应用。

另外，可编程控制器的应用不包括航空、医疗应用、焚化和燃烧设备、载人设备、娱乐及休闲设施、安全装置等与人的生命财产密切相关以及在安全和控制系统方面需要特别高的可靠性时的应用。

然而，对于这些应用，假如用户咨询当地三菱代表机构，提供有特殊要求方案的大纲并提供满足特殊环境的所有细节及用户自主要求，则可以进行一些应用。

Microsoft、Windows、Windows NT 是美国 Microsoft Corporation 公司在美国及其它国家的注册商标。
Adobe、Acrobat 是 Adobe Systems Incorporated 公司的注册商标。
Pentium、Celeron 是 Intel Corporation 公司在美国及其它国家的商标及注册商标。
Ethernet 是美国 Xerox.co.ltd 公司的注册商标。
PC-9800、PC98-NX 是日本电气公司的注册商标。
其它本文中的公司名、商品名是各公司的商标或注册商标。

QCPU(Q模式)/QnACPU

编程手册(公共指令篇) 2/2

技术服务热线:

800-828-9910

服务时间: **9:00~12:00**

13:00~17:00(节假日除外)

三菱电机自动化(上海)有限公司

地址：上海市黄浦区新昌路80号智富广场4楼
邮编：200003
电话：021-61200808　传真：021-61212444
网址：www.mitsubishielectric-automation.cn

书号	SH(NA)-080450CHN-B(0711)STC
印号	STC-Q/QnaCPU-CI-PM(0711)

内容如有更改
恕不另行通知